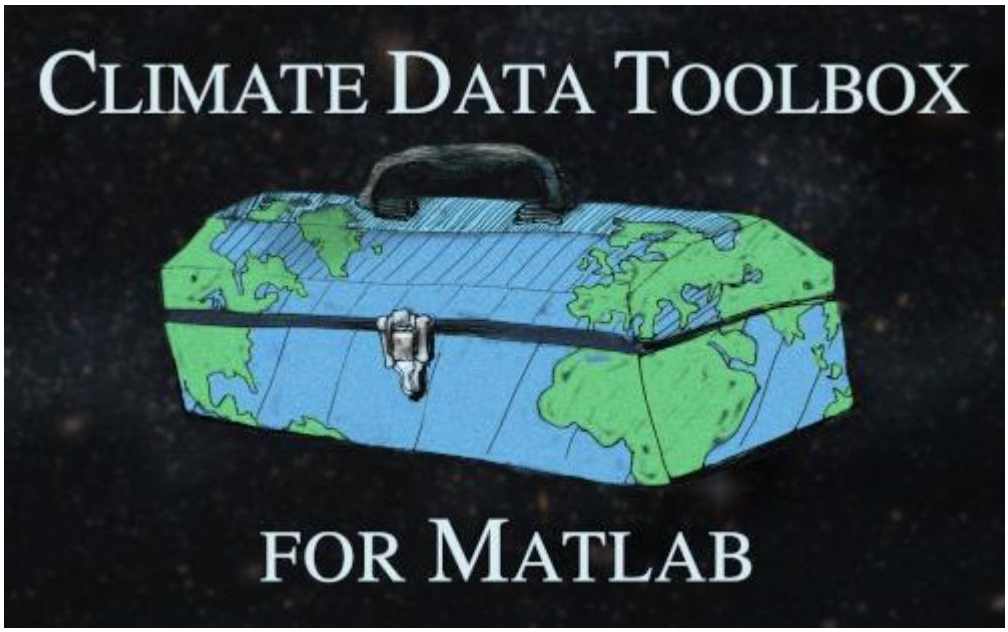
目 录

Climate Data Toolbox for Matlab



欢迎使用 Matlab 气候数据工具箱！ 以下是 CDT 入门的一些提示：

CDT 入门

CDT 目录

有关 CDT 功能、数据集和教程的完整列表，请查看 CDT 目录页面。

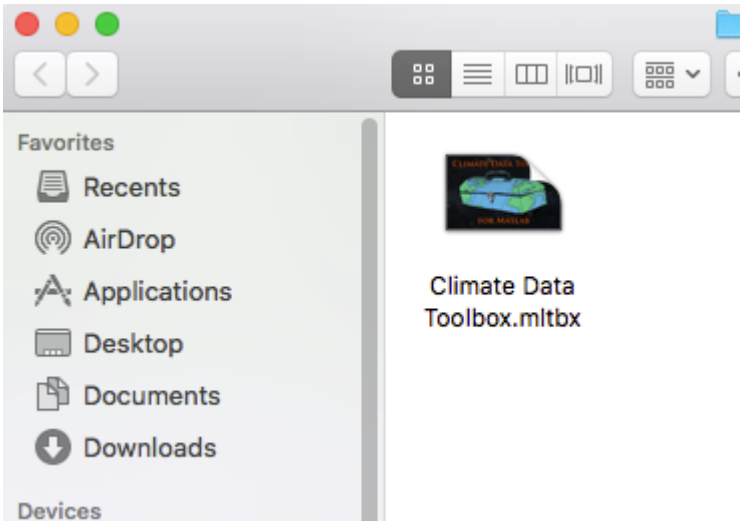
下载 CDT

您有一些下载 CDT 的选项。您可以在此处获得该工具箱的最新版本（大约 100MB），只需双击该文件以进行安装（如下所述），或者可以通过“附加功能”菜单找到所有最新的单个文件（方法 2 所述下方），在 Mathworks File Exhange 文件交换或 GitHub 上。

在 Matlab 中安装 CDT

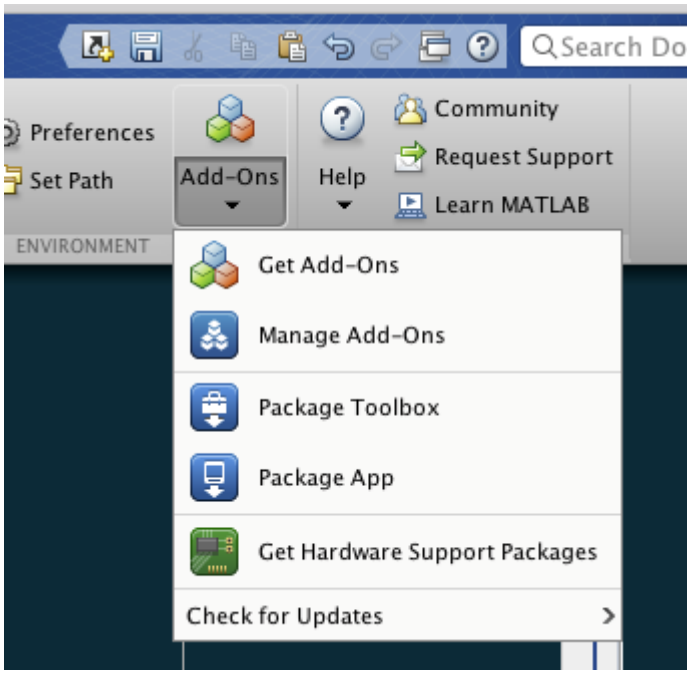
对于如何安装 CDT，您有两个选择：

1.如果您已经下载了.mltbx 文件，请在文件系统中导航至该文件，该文件可能类似于以下内容：



只需双击该文件，然后在弹出菜单时，单击“Install”。安装应该花费不到一秒钟的时间。

2.或者，如果尚未下载 mltbx 文件，则可以直接从 Matlab 中通过“加载项”按钮获取它。在 HOME 菜单中看起来像这样：



单击“获取附加功能”，查找 *Climate Data Toolbox*，然后按照提示自动下载并安装 CDT。

访问帮助

需要帮助？只需输入

```
cdt
```

来访问 TMD 文档。要获得有关特定函数的帮助，请键入 cdt，然后输入函数名称。例如，

```
cdt eof
```

访问有关 eof 函数的帮助。

还需要帮助吗？

- 获得帮助的最快方法是在 Matlab Answers forum. 上提问。
- 如果发现 bug 或遇到论坛无法解决的其他技术问题，请通过 CDT Issues page 进行报告。
- 您也可以尝试直接通过 chad@chadagreene.com 向我发送电子邮件。我不能保证会回答，但是我确实阅读了我收到的所有消息。为了最大限度地提高您收到回复的机会，请描述您遇到的问题，描述您要完成的工作，包括所有错误消息，并**提供一个最小的可行示例**（如果可以）。

引用 CDT

请引用我们的论文！

Chad A. Greene, Kaustubh Thirumalai, Kelly A. Kearney, Jose Miguel Delgado, Wolfgang Schwanghart, Natalie S. Wolfenbarger, Kristen M. Thyng, David E. Gwyther, Alex S. Gardner, and Donald D. Blankenship. The Climate Data Toolbox for MATLAB. *Geochemistry, Geophysics, Geosystems* 2019. doi:10.1029/2019GC008392

描述性统计

- scatstat1 返回每个值在给定一维半径内所有点的统计值。这类似于采用移动平均，但不必将点等距分布，也不必使 x 值单调递增。
- scatstat2 返回每个值在给定半径内所有点的统计值。这类似于采用二维移动平均，但是不需要将点等距分布。
- wmean 计算加权平均值或其他加权平均值。
- standardize 可删除变量的均值并对其进行缩放，以使其标准偏差为 1。
- ensemble2bnd 计算并绘制整体数据的百分比范围。
- trend 使用最小二乘法计算数据序列的线性趋势。
- polyfitw 计算加权多项式拟合。
- detrend3 沿矩阵的第三维执行线性最小二乘法去趋势化。
- monthly 计算一年中指定月份的变量统计信息。
- season 估算与年度周期或时间序列相关的异常。
- deseason 从时间序列中删除了变异性的季节性（也称为年度）成分。
- climatology 给出了变量的典型值，因为该变量一年四季都在变化。
- sinefit 将正弦曲线的最小二乘估计与周期为 1 年的时间序列数据拟合。
- sineval 产生指定振幅和相位的正弦波，频率为 1/yr。

scatstat1 文档

scatstat1 返回每个值在给定一维半径内所有点的统计值。这类似于采用移动平均，但不必将点等距分布，也不必使 x 值单调递增。这类似于采用移动平均值，但不必将点等距分布，也不必使 x 值单调递增。另请参见 scatstat2。

语法

```
ybar = scatstat1(x,y,radius)
ybar = scatstat1(x,y,radius,fun)
ybar = scatstat1(...,options)
```

说明

ybar = scatstat1(x,y,radius) 返回每个点 x 在指定半径内所有 y 值的平均值。

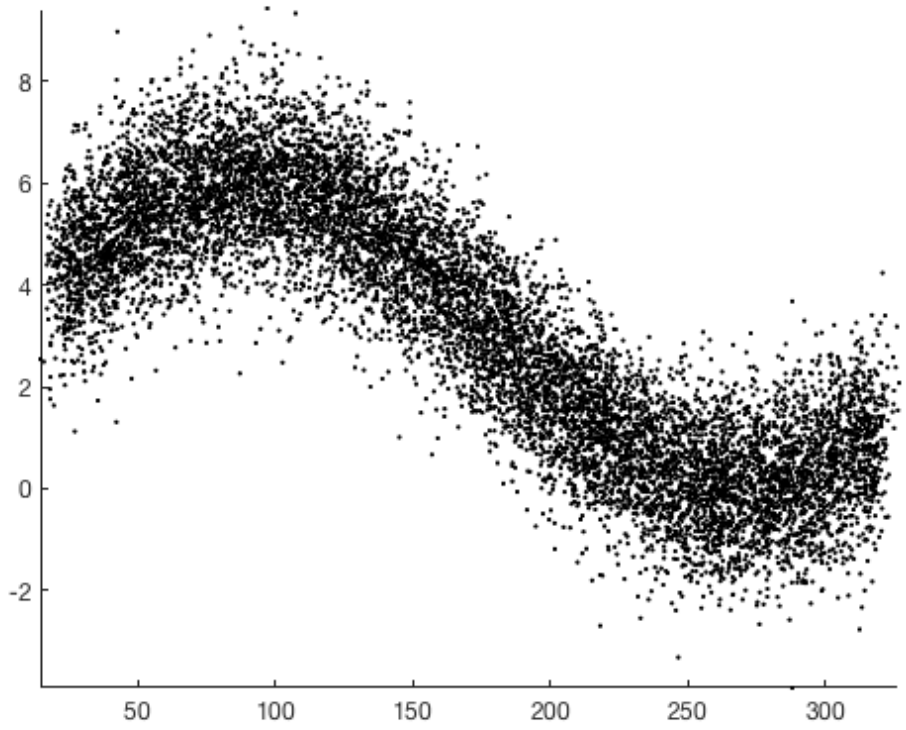
ybar = scatstat1(x,y,radius,fun) 将任何函数 fun 应用于 y 值，默认 fun 为@mean，但可以是 @ median，@ nanstd 等。

ybar = scatstat1(...,options) 允许函数接受的任何选项。例如，'omitnan'。

示例 1

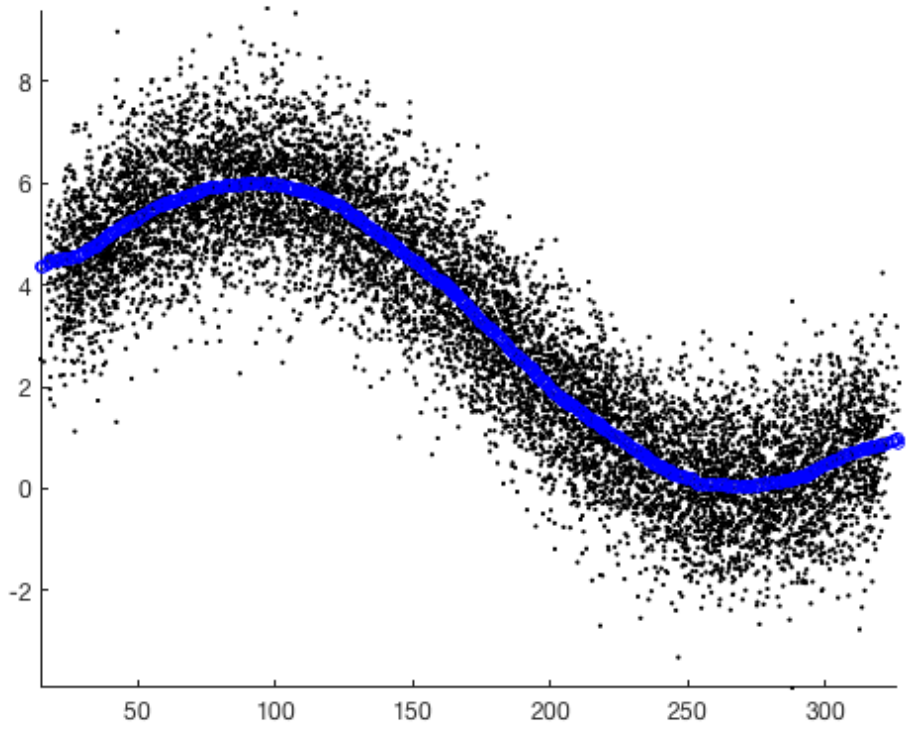
获取每个 x 点 10 个单位内所有点的局部中位数。 首先创建一些 N = 10,000 点的随机未排序数据：

```
N = 10000;

x = randi(300,N,1) + 20+3*randn(N,1) ;

y = 3*sind(x) + randn(size(x)) + 3;

plot(x,y,'k.')
axis tight
box off
```



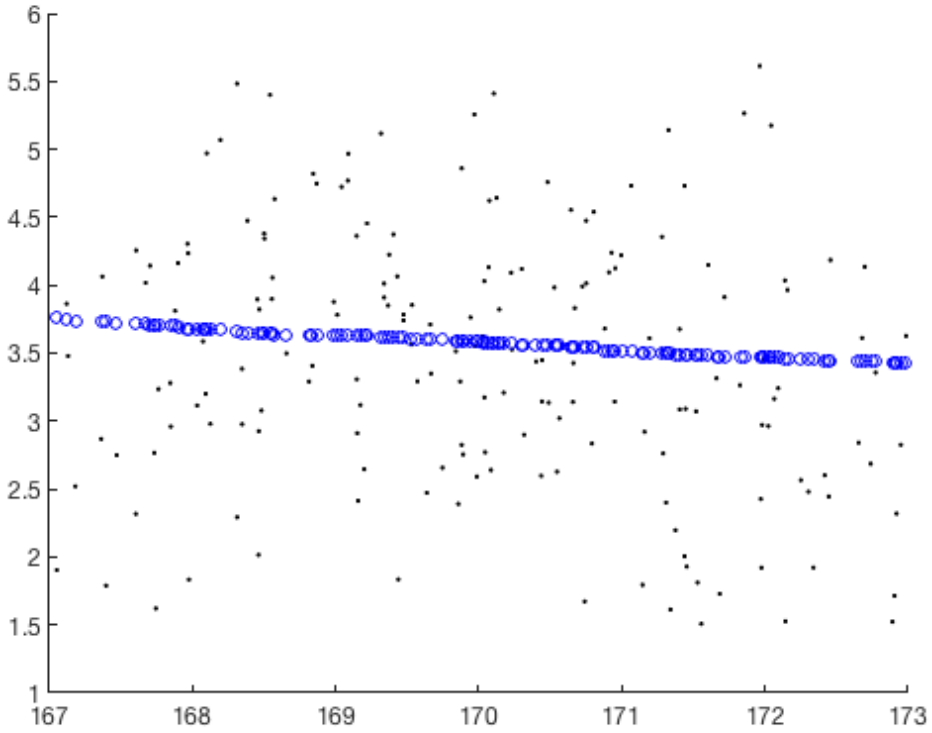
现在获得每个值在 15x 个单位内所有点的移动中位数：

```
yb = scatstat1(x,y,15,@median,'omitnan');

hold on
plot(x,yb,'bo')
```



请注意，此函数不会插值或强制执行任何相等的间距。最终值与每个 x 点相对应，而不必等间隔或排序。这是一个放大，因此您可以看到 x 值的间距不相等：

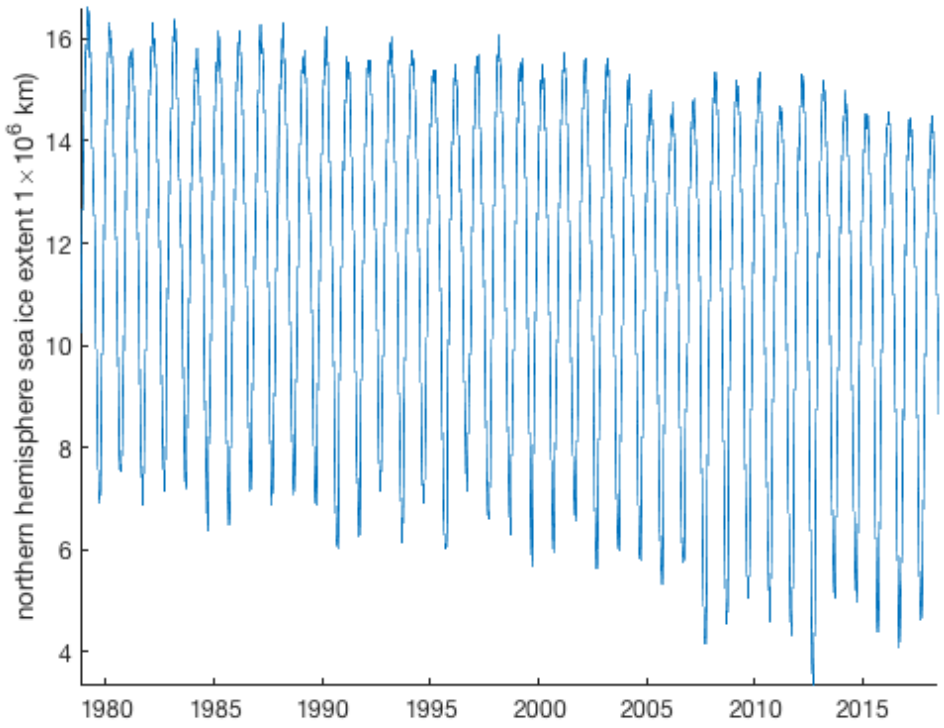
```
axis([167 173 1 6])
```



示例 2:海冰数据

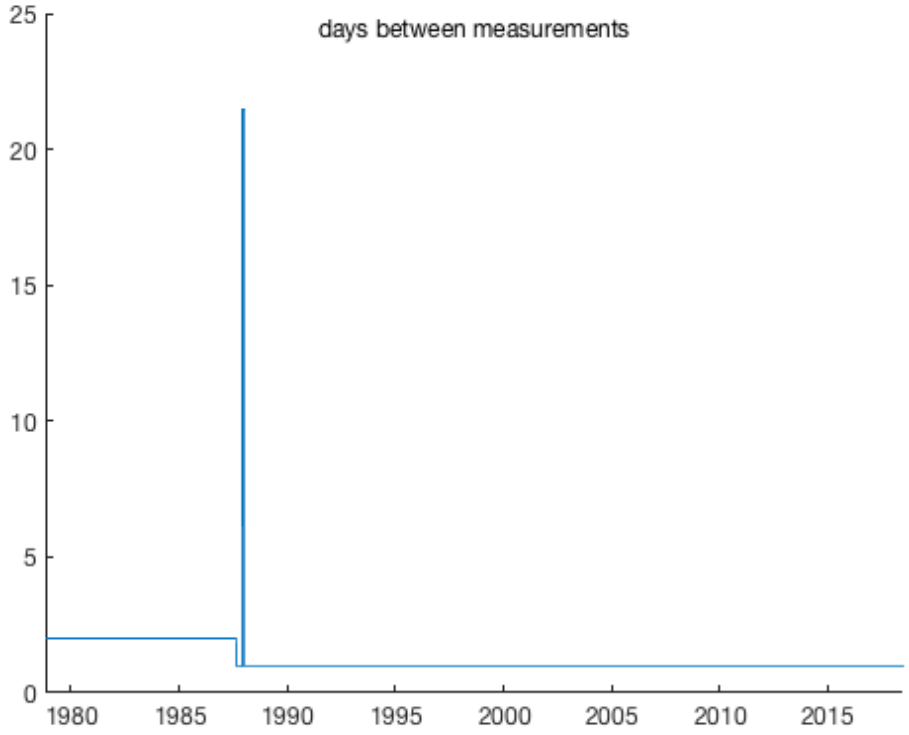
从 1978 年到 1988 年，NSIDC 海冰时间序列包含大约每两天一次测量。1988 年之后，是每日时间序列。如果您要在这个没有恒定时间分辨率的数据集上获得 2 年海冰时间序列的移动平均值，该咋办？
情况是酱婶的：

```
load seaice_extent.mat %载入样本数据

figure
plot(t,extent_N)
axis tight
box off
ylabel 'northern hemisphere sea ice extent 1\times10^6 km)'
```



在 1988 年之前，该数据每隔一天可用一次，然后数据中出现空白，然后是每日数据。绘制两次测量之间的时间（以天为单位）为 datenum 值的梯度：

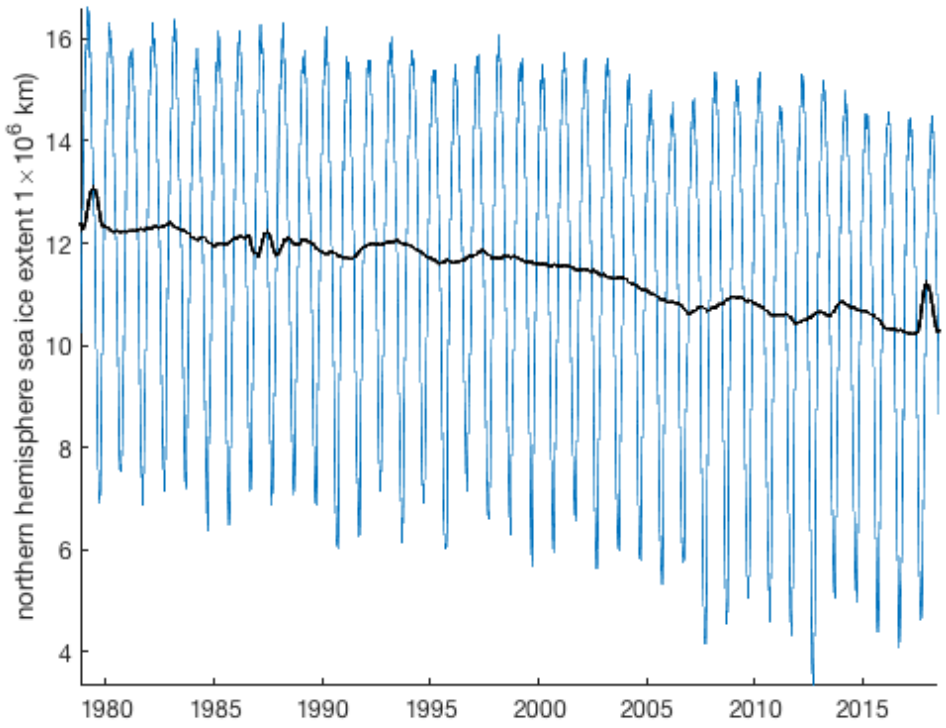
```
figure

plot(t,gradient(datenum(t)))

ntitle 'days between measurements'
box off
```



你看，这有个空。

对于两年的移动平均值（半径为 1 年），请将 datetime 格式 t 转换为 datenum 格式，并使用半径参数为 365.25 天的 scatstat1 函数：

```
extent_N_2yr = scatstat1(datenum(t),extent_N,365.25);



figure(1) % 回望海冰时间序列图

hold on
plot(t,extent_N_2yr,'k','linewidth',2)
```



当然，您可能不希望信任在端点 1 个半径内（2 年移动平均值为 1 年）的任何事物，这就是为什么在时间序列开始和结束处的出现奇怪摆动的原因。

作者简介:

这个函数是来自德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 于 2016 年 6 月写的。

scatstat2 文档

scatstat2 函数返回每个值在给定半径内所有点的统计值。这类似于采用二维移动平均，但是不需要将点等距分布。

另请参见 scatstat1.

语法

```
zbar = scatstat2(x,y,z,radius)
zbar = scatstat2(x,y,z,radius,fun)
ybar = scatstat1(...,options)
```

说明

zbar = scatstat2(x,y,z,radius) 返回每个点（x，y）内指定半径内所有 z 值的平均值。

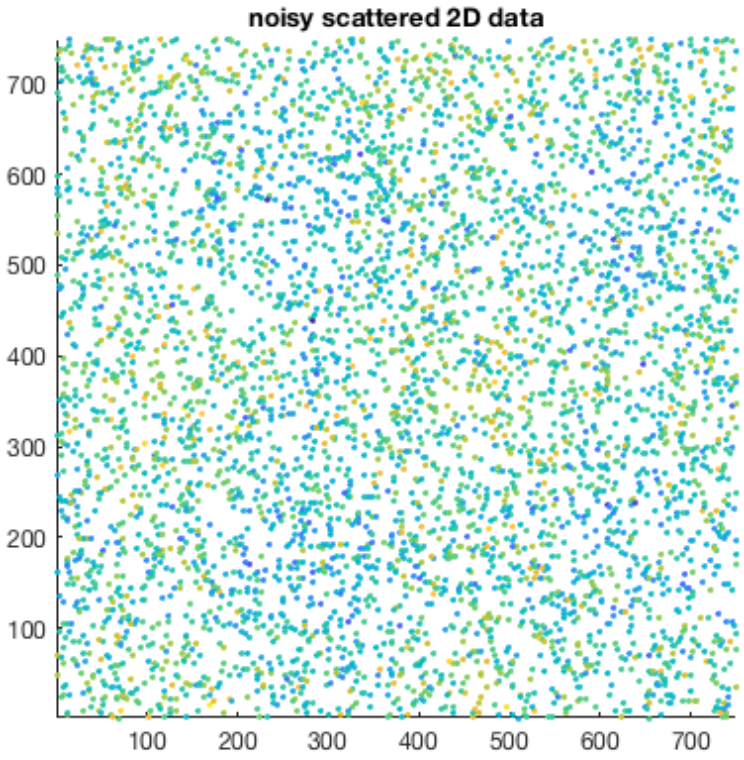
zbar = scatstat2(x,y,z,radius,fun) 将任何函数 fun 应用于 z 值，默认 fun 为@mean，但这可能是您想要的任何函数，甚至是您自己的匿名函数。

zbar = scatstat2(...,options) 允许函数接受的任何选项。例如，'omitnan'。

示例：局部中值

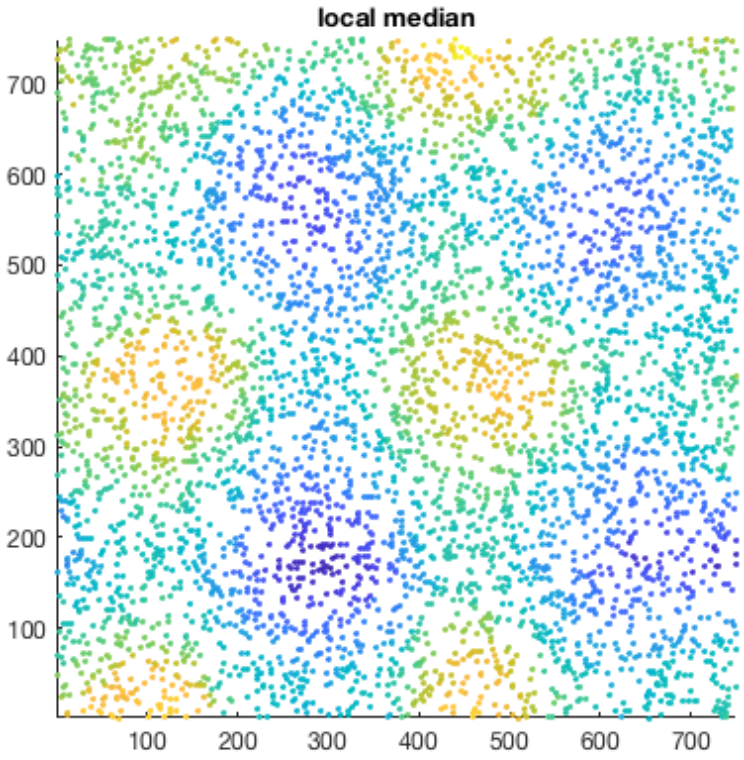
考虑以下这个包含 5000 个分散点的噪声数据集：

```
N = 5000;

x = randi(750,N,1);

y = randi(750,N,1);



%z 包含*真实的*结构，加了很多噪声:
z = sind(x) + cosd(y) + 3*randn(size(x));

scatter(x,y,10,z,'filled')
axis image
title 'noisy scattered 2D data'
```



看起来几乎只有噪音。通过取 75 个单位半径内的局部中值来滤除高频噪声：

```
z_median = scatstat2(x,y,z,75,@median);



scatter(x,y,10,z_median,'filled')
axis image
title 'local median'
```



作者简介:

这个函数是来自德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 于 2016 年 6 月写的。

wmean 文档

wmean 计算加权平均值或其他加权平均值。

语法

```
M = wmean(A,weights)
M = wmean(...,'all')
M = wmean(...,'dim',dim)
M = wmean(...,'nanflag')
```

说明

M = wmean(A,weights) 返回 A 的元素沿大小不等于 1 的第一个数组维的加权平均值。A 的维必须与权重的维匹配。

- 如果 A 是向量，则 wmean（A，weights）返回元素的加权平均值。
- 如果 A 是矩阵，则 wmean（A，weights）返回包含每一列加权平均值的行向量。
- 如果 A 是多维数组，则 wmean（A，weights）沿大小不等于 1 的第一个数组维操作，将元素视为向量。 该尺寸变为 1，而所有其他尺寸的尺寸保持不变。.

M = wmean(...,'all') 计算所有元素的加权平均值。（需要 Matlab R2018b 或更高版本）

M = wmean(...,'dim',dim) 返回沿维度 dim 的平均值。例如，如果 A 是矩阵，则 wmean（A，weights，'dim'，2）是包含每行加权平均值的列向量。

M = wmean(...,'nanflag') 指定是否为任何先前语法从计算中包括或省略 NaN 值。 wmean（A，weights，'includenan'）包括计算中的所有 NaN 值，而 wmean（A，weights，'omitnan'）则会忽略它们。

示例 1：一维数组

这是值 y 及其平均值的数组，没有任何特殊的权重：

```
y = 0:2:10

mean(y)
```

```
y =

  Columns 1 through 5

         0    2.00    4.00    6.00    8.00

  Column 6

        10.00

ans =

         5.00
```

上面的答案等同于对 y 中的每个值使用权重相等的 wmean：

```
weight = [1 1 1 1 1 1];



wmean(y,weight)
```

```
ans =

          5.00
```

但是也许您正在进行计算，其中将对接近零的值进行最重的加权，如下所示：

```
weight = logspace(1,0,6)



wmean(y,weight)
```

```
weight =

  Columns 1 through 5

       10.00          6.31          3.98          2.51          1.58

  Column 6

          1.00

ans =

          2.61
```

示例 2：二维矩阵

这是一个二维矩阵 A，就像示例 1 一样，我们将首先在整个上使用相等的权重：

```
% A 数据矩阵:
A = [0 1 2
     2 3 4
     4 5 6
     6 7 8]

%都是等权重:
w = ones(size(A))
```

```
wmean(A,w)
```

```
A =

     0     1.00     2.00

2.00     3.00     4.00

4.00     5.00     6.00

6.00     7.00     8.00

w =

1.00     1.00     1.00

1.00     1.00     1.00

1.00     1.00     1.00

1.00     1.00     1.00

ans =

3.00     4.00     5.00
```

如果您需要沿行操作而不是在 A 的各列下进行操作，请按以下方式指定维度 2：

```
wmean(A,w,'dim',2)
```

```
ans =

1.00

3.00

5.00

7.00
```

如果您跟随这之前的主页，您可以向自己证明上面的答案（在所有权重均相等的情况下）与 mean（A，2）相同。

问：如何使 A 的加权均值结果是等于 A 的第二行？

答：我可以想到两种方式。一种方法是将除第 2 行之外的所有权重都设置为零。或者，我们可以使这些行保持原样，而将第二行权重设置为一个非常大的数字：

```
w(2,:) = 1e100



wmean(A,w)
```

```
w =

          1.00          1.00          1.00


1000000000000000001590289110975991804683608085639452813897813275577478387721703810608134699858568151
04.00
1000000000000000001590289110975991804683608085639452813897813275577478387721703810608134699858568151
04.00
1000000000000000001590289110975991804683608085639452813897813275577478387721703810608134699858568151
04.00

          1.00          1.00          1.00

          1.00          1.00          1.00

ans =

          2.00          3.00          4.00
```

如果 A 中一个元素是 NaN 咋办？

```
A(2,2) = NaN



wmean(A,w)
```

```
A =

            0          1.00          2.00

          2.00           NaN          4.00

          4.00          5.00          6.00

          6.00          7.00          8.00
```

```
ans =

    2.00     NaN     4.00
```

我们可以忽略该缺失的部分，而不必将该列的整个解决方案转换为 NaN：

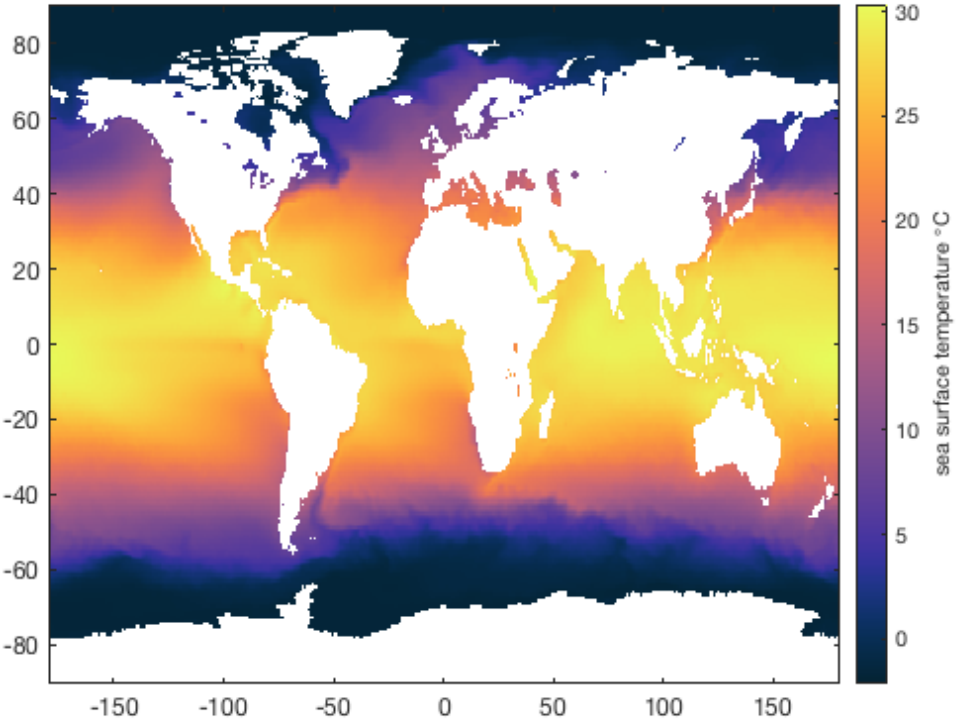
```
wmean(A,w,'omitnan')
```

```
ans =

    2.00    4.33    4.00
```

示例 3：面积权重 SST

这是一张使用 CDT 附带的示例数据绘制的海面温度图：

```
% 载入示例数据:
load global_sst

% 从开氏温度转为摄氏度:
sst = sst-273.15;

% 绘制 SST:
imagescn(lon,lat,sst)
cmocean thermal % colormap
cb = colorbar;
ylabel(cb,'sea surface temperature \circC')
```



初学者可能会尝试通过获取所有 sst 值的未加权平均值来获得全球平均海表温度：

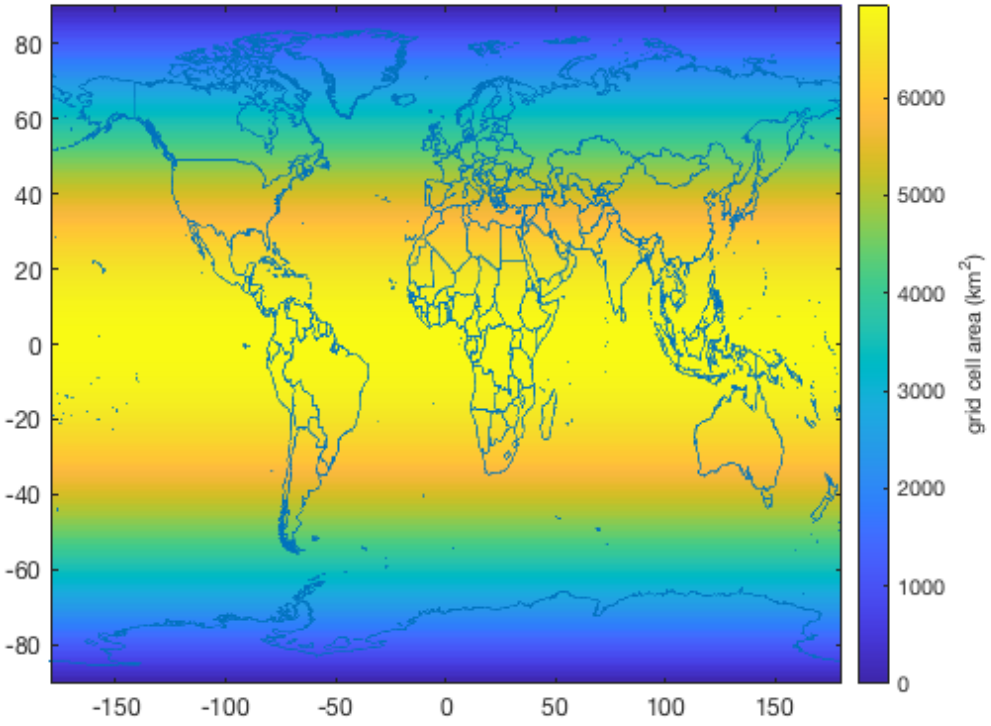
```
mean(sst,'all','omitnan')
```

```
ans =

    13.84
```

但是，我们必须记得，sst 网格对应于等距的纬度和经度，它们的面积并不都相等。经度线在极点会聚，因此 SST 的面积平均量度需要加权每个网格单元的均值。
为此，请将 lat 和 lon 数组转换为网格，然后使用 cdtarea 获取每个网格单元的面积。以下是每个网格单元的区域：（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
% 将经纬度数据转为网格:
[Lon,Lat] = meshgrid(lon,lat);

% 获取每个网格单元的面积:
A = cdtarea(Lat,Lon,'km2');

imagescn(lon,lat,A)
hold on
borders % 国界线

cb = colorbar;
```

```
ylabel(cb,'grid cell area (km^2)')
```



有了网格单元面积 A，我们现在可以使用 wmean 来获得区域平均海面温度：

```
wmean(sst,A,'all','omitnan')
```

```
ans =

    18.45
```

这与我们使用未加权平均值获得的 13.8 度值有很大差异！这并不奇怪：在这种情况下，“未加权”平均值实际上是用词不当，因为通过在平均 SST 计算中对每个网格单元赋予相等的权重，我们实际上对极点附近的微小网格，冷网格单元给予了不适当的权重。因此，“未加权”平均 SST 值低于面积平均 SST 值也就不足为奇了。

作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

standardize 文档

standardize 可删除变量的均值并对其进行缩放，以使其标准偏差为 1。该操作有时被称为“居中和缩放”。

语法

```
Xs = standardize(X)
Xs = standardize(...,dim)
Xs = standardize(...,nanflag)
Xs = standardize(...,'weighting',w)
[Xs,mu] = standardize(...)
```

说明

Xs = standardize(X) 从 X 减去 X 的平均值，然后除以 X 的标准偏差。

Xs = standardize(...,dim) 指定要沿其进行操作的维度。默认情况下，标准化是沿第一个非单维度进行的。

Xs = standardize(...,nanflag) 指定是否从任何先前语法的计算中包括或省略 NaN 值。standardize(X,'includenan') 在计算中包括所有 NaN 值，而 standardize(X,'omitnan') 忽略它们。默认行为是'includenan'。

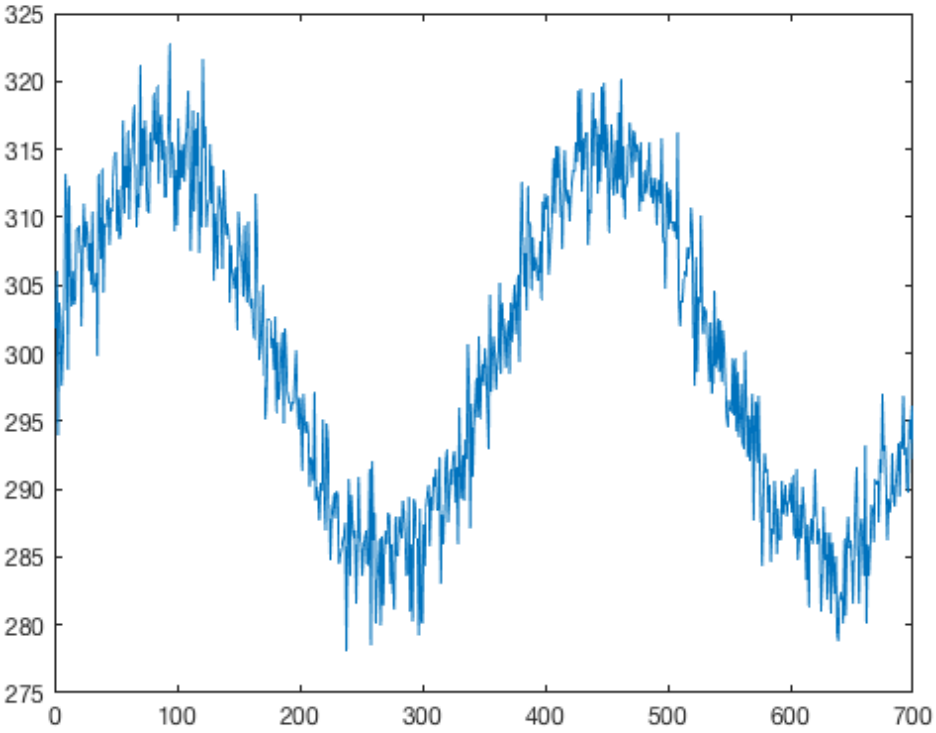
Xs = standardize(...,'weighting',w) 指定用于计算标准偏差的加权方案 w。当 w=0（默认值）时，S 用 N-1 归一化。当 w=1 时，S 通过观测数据量 N 归一化。w 也可以是包含非负元素的权重向量。在这种情况下，w 的长度必须等于 std 在其上进行操作的尺寸的长度。

[Xs,mu] = standardize(...) 返回 mu 平均值和 X 的标准差。如果 X 是向量，则 mu(1) 是 X 的平均值，而 mu(2) 是 X 的标准偏差。如果 X 是多维的，平均值是操作方向上的第一个元素，而 std 是第二个方向。换句话说，如果 X 是三维，并且沿第三维标准化，则 mu(:,:,1) 就是平均值，而 mu(:,:,2) 就是 X 的标准差。

示例 1：一维时间序列

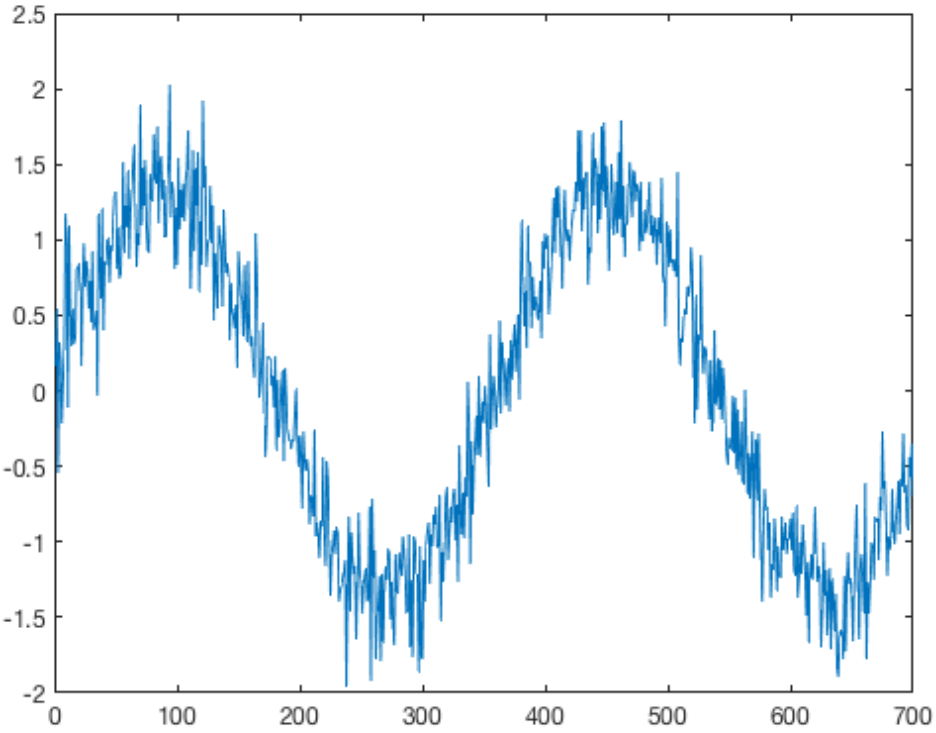
介是个时间序列：

```
x = 1:700;

y = 15*sind(x)+3*randn(size(x))+300;



plot(x,y)
```



在上图中，您可以看到 y 的平均值约为 300，标准差约为 10。Y 标准化之后大约是酱婶的：

```
ys = standardize(y);



plot(x,ys)
```

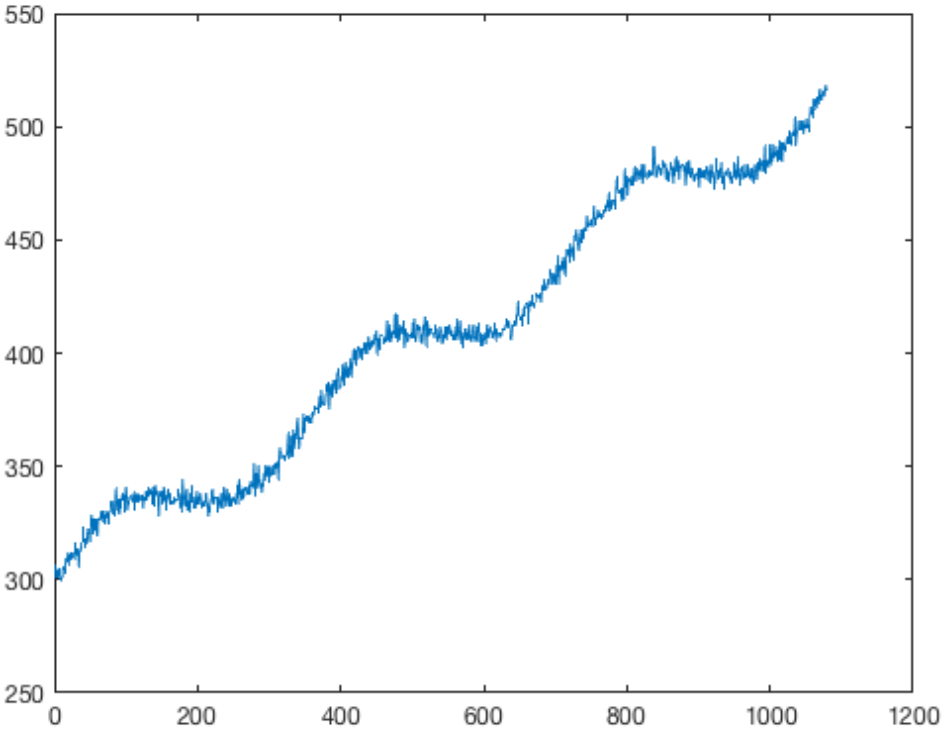


现在，信号在 0 附近振荡，其标准差为 1。

示例 2：带有趋势的一维时间序列

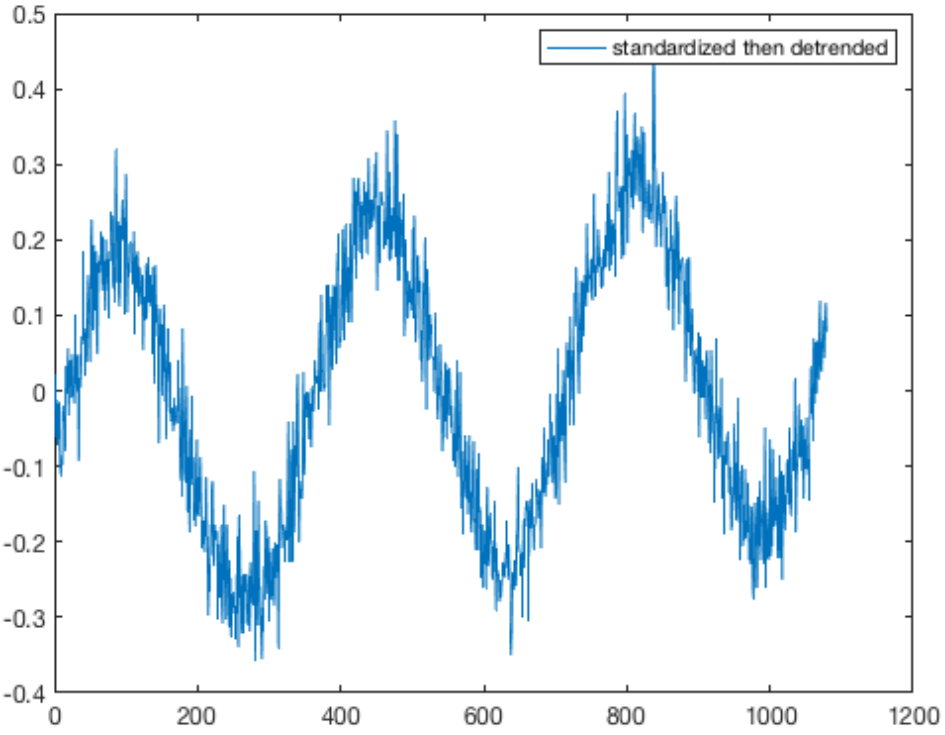
如果你的时间序列有趋势该咋办？像酱紫：

```
x = 1:1080;

y = 15*sind(x)+3*randn(size(x))+300 +x/5;



plot(x,y)
```

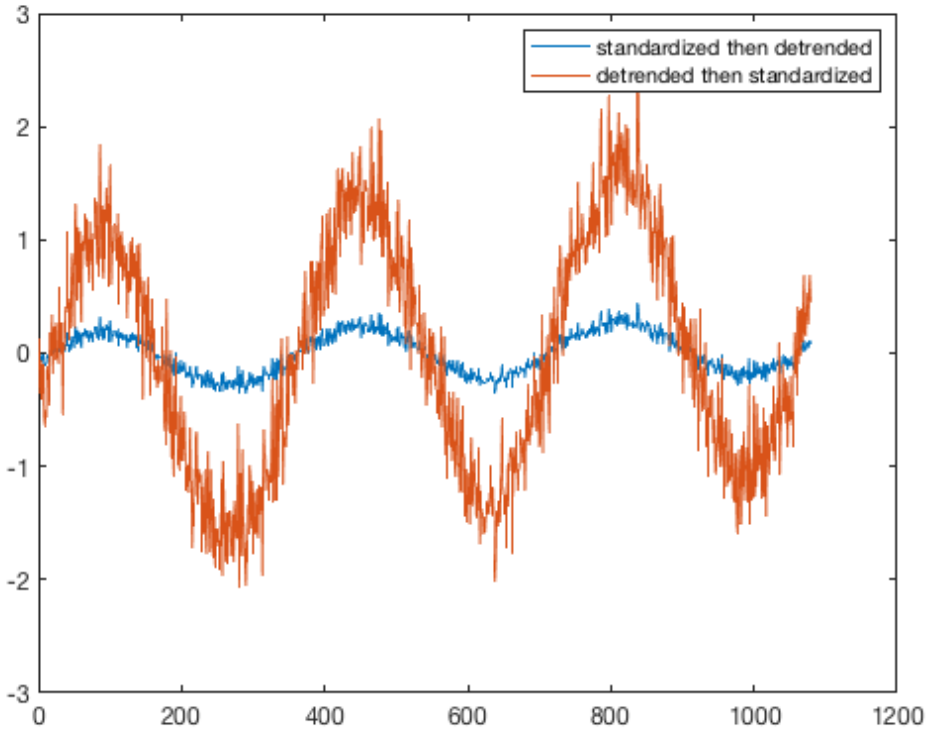


重要的是要注意，先进行标准化再进行去趋势化处理所产生的结果与先进行反趋势化然后再进行标准化处理的结果不同。区别在于：

```
%标准化 y:
ys = standardize(y);

%把标准化后的 y 去去趋势化:
ysd = detrend(ys);

plot(x,ysd)
legend('standardized then detrended')
```




```
yd = detrend(y);



yds = standardize(yd);



hold on
plot(x,yds)
legend('standardized then detrended',...
   'detrended then standardized')
```

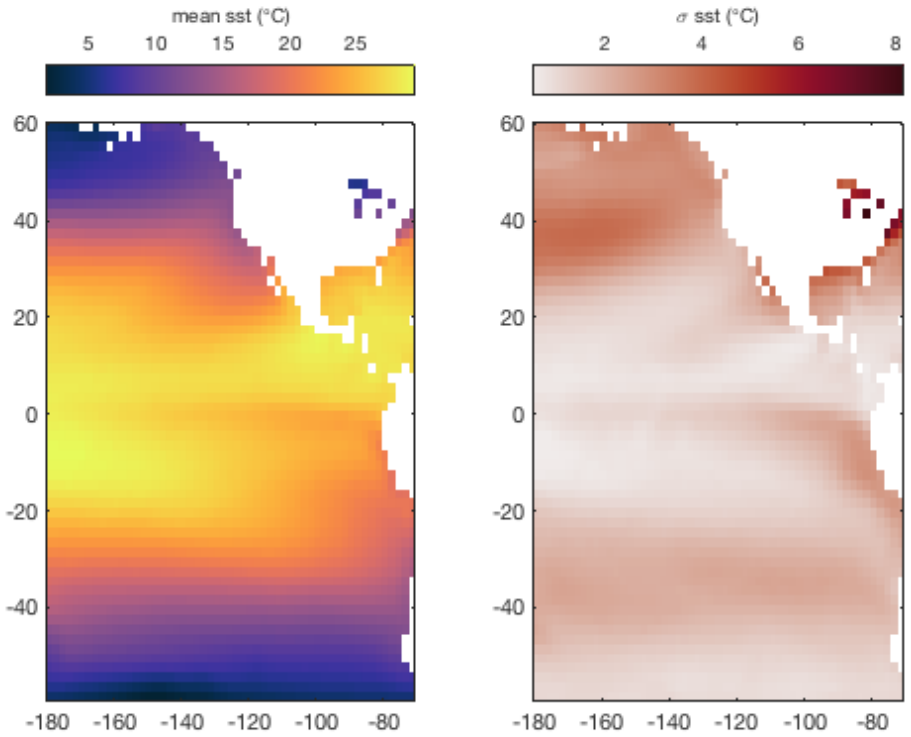


最终结果 ysd 和 yds 的平均值均为 0，但是 ysd 的标准偏差小于 1，因为它在标准化后被去趋势化了。

示例 3:

考虑该样本海面温度数据集的大小为 60x55x802。首先绘制原始 sst 数据集的均值和标准差：

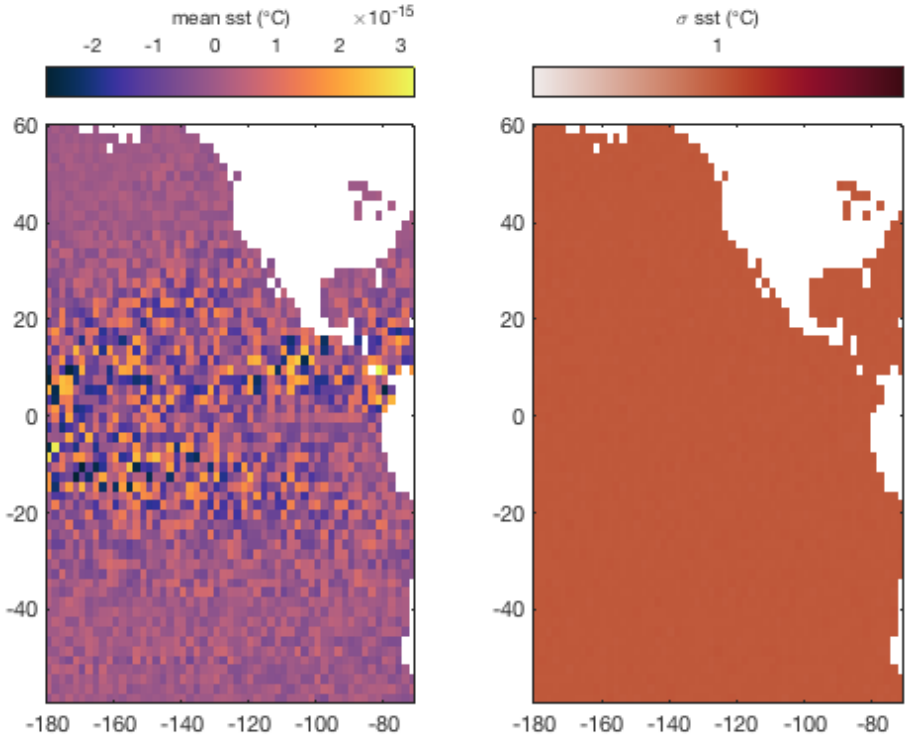
```
load pacific_sst



figure
subplot(1,2,1)
imagescn(lon,lat,mean(sst,3))
cmocean thermal
cb(1) = colorbar('location','northoutside');
xlabel(cb(1),'mean sst (\circC)')

subplot(1,2,2)
imagescn(lon,lat,std(sst,[],3))
cmocean amp
cb(2) = colorbar('location','northoutside');
xlabel(cb(2),'\sigma sst (\circC)')
```



上面的地图展示了我们的期望：平均而言，靠近赤道的温度较高，而靠近两极的温度较低。北美五大湖平均寒冷，具有最大的海面温度变化性，因为它们在冬天结冰，而在夏天则变暖。

如果您想居中并缩放此 SST 数据集，

```
% 沿着第三维标准化 sst:
sst_s = standardize(sst,3);

figure
subplot(1,2,1)
imagescn(lon,lat,mean(sst_s,3))
cmocean thermal
cb(1) = colorbar('location','northoutside');
xlabel(cb(1),'mean sst (\circC)')

subplot(1,2,2)
imagescn(lon,lat,std(sst_s,[],3))
cmocean amp
cb(2) = colorbar('location','northoutside');
xlabel(cb(2),'\sigma sst (\circC)')
```



左上方的图显示，在对 sst 进行居中和缩放后，sst 的平均值仅为数值噪声。这是典型的噪声模式，始终鼓励您检查色标轴上的科学计数法。
在右边，标准偏差到处都是 1。是因为这正是我们将其设置为标准的条件。

作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

ensemble2bnd 文档

ensemble2bnd 函数可计算整个数据集的统计信息，并提供多个选项来绘制这些集合统计信息。
在这种情况下，我们使用“集合”来指代使用相同自变量坐标的任何矢量数据集。集合数据在气候科学中很常见；模型研究通常使用模型内集合来包含初始条件或参数中的不确定性；模型间集合更包括模型过程中的结构不确定性。时间序列数据也是此类分析的主要候选者，因为我们经常希望查看年际变化。
ensemble2bnd 函数最初是为了将集合数据输入到 boundedline 函数中而创建的，并且已扩展为允许使用多个选项来绘制这些类型的数据集的平均值、中位数、四分位数、上限/下限等。

语法

```
A = ensemble2bnd(x,y)
[A, h] = ensemble2bnd(x,y, 'plot', plottype)
[...] = ensemble2bnd(..., 'dims', dims)
[...] = ensemble2bnd(..., 'center', center)
[...] = ensemble2bnd(..., 'prc', prc)
[...] = ensemble2bnd(..., 'alpha', alphaval)
[...] = ensemble2bnd(..., 'tlim', tlim)
[...] = ensemble2bnd(..., 'cmap', colors)
[...] = ensemble2bnd(..., 'axis', hax)
[...] = ensemble2bnd(..., 'whisker', w)
```

说明

[A, h] = ensemble2bnd(x,y) 计算每个 x 坐标处 y 的平均值和上下限。x 数据应为向量数组，y 为 nx x ny x nens 数组，其中 nx 对应于 x 维点的数量（即 x 输入的长度），ny 对应于不同数据集的数量，nens 是每个数据集中集合成员的数量。（请参阅'dims'参数以排列这些维度）
[A, h] = ensemble2bnd(x,y, 'plot', plottype) 将数据集绘制为带阴影的线表示百分位数（'boundedline'）、带误差线的行（'errorbar'）、带误差线的未堆叠条形图（'bar'）或箱线图（'boxplot'）。默认值为'none'（无），表示将不创建图。
ensemble2bnd(..., 'dims', dims) 允许通过包含字母 x、y 和 e 的 3 个字母的字符串对输入矩阵 y 进行置换，这些字母表示输入矩阵中的哪些维度对应于独立坐标，不同的数据集，以及每个数据集的集合。默认值为“ xye”。
ensemble2bnd(..., 'center', center) 在每个数据集的“平均值”（默认）和“中位数”之间切换中心线统计量。
ensemble2bnd(..., 'prc', prc) 指定与上下限相对应的百分位值。这些应该成对出现，从最低到最高百分位。默认值为[0 100]。
ensemble2bnd(..., 'alpha', true) 指示应使用透明色块而不是不透明色块来绘制边界色块。
ensemble2bnd(..., 'tlim', tlim) 更改与百分位数相关的填色或错误条相关的颜色限制。默认情况下，以指示的颜色（饱和度=1）绘制中心线/条/点，边界对象的颜色（色块、误差线等）使用较浅的颜色，并且随着边界的扩大而变浅（默认限制为[0.1 0.5]）。
ensemble2bnd(..., 'cmap', colors) 为绘制的数据集提供了替代的颜色顺序。
ensemble2bnd(..., 'axis', hax) 将任何图添加到指示的轴，而不是当前轴。
ensemble2bnd(..., 'whisker', w) 更改用于计算箱图离群值的晶须长度因数（仅适用于 boxplot 选项）。

示例：海冰年际变化

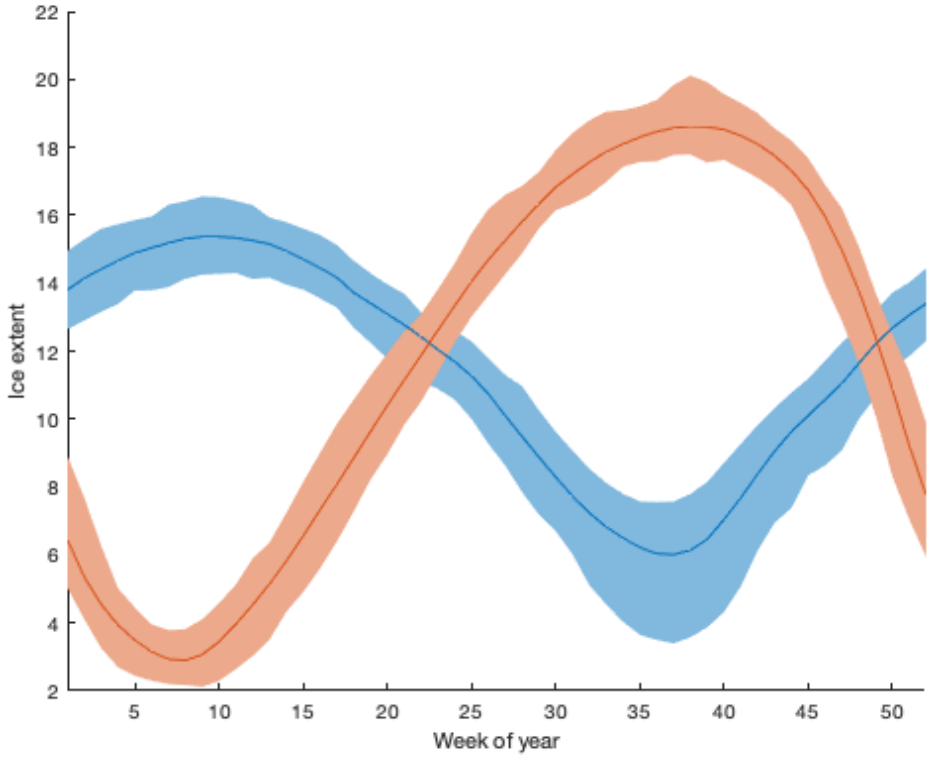
对于我们的示例，我们将使用海冰范围时间序列作为集合数据。我们将首先使用 reshapetimeseries 将两个时间序列整形为年×年的时间矩阵，然后将它们串联起来。早期的数据使用两天的时间间隔，而后面的数据则是每天的时间间隔；我们将按周对数据进行分割，以避免此更改产生任何人为因素：

```
load seaice_extent.mat

iceN = reshapetimeseries(t, extent_N, 'bin', 52);
iceS = reshapetimeseries(t, extent_S, 'bin', 52);

ice = cat(3, iceN, iceS);
```

默认情况下，该函数在数据中每个 x 位置计算数据的均值和上/下限。我们可以绘制任何一个 boundedline：

```
[A, h] = ensemble2bnd(1:52,ice, 'dims', 'xey', 'plot', 'boundedline')
set(gca,'xlim', [1 52]);
xlabel('Week of year');
ylabel('Ice extent');
```

```
A =

  struct with fields:



     cent: [52×2 double]

    bndlo: [52×2 double]

    bndhi: [52×2 double]

    errlo: [52×2 double]

    errhi: [52×2 double]

h =

  struct with fields:



       ln: [1×2 Line]

    patch: [1×2 Patch]
```

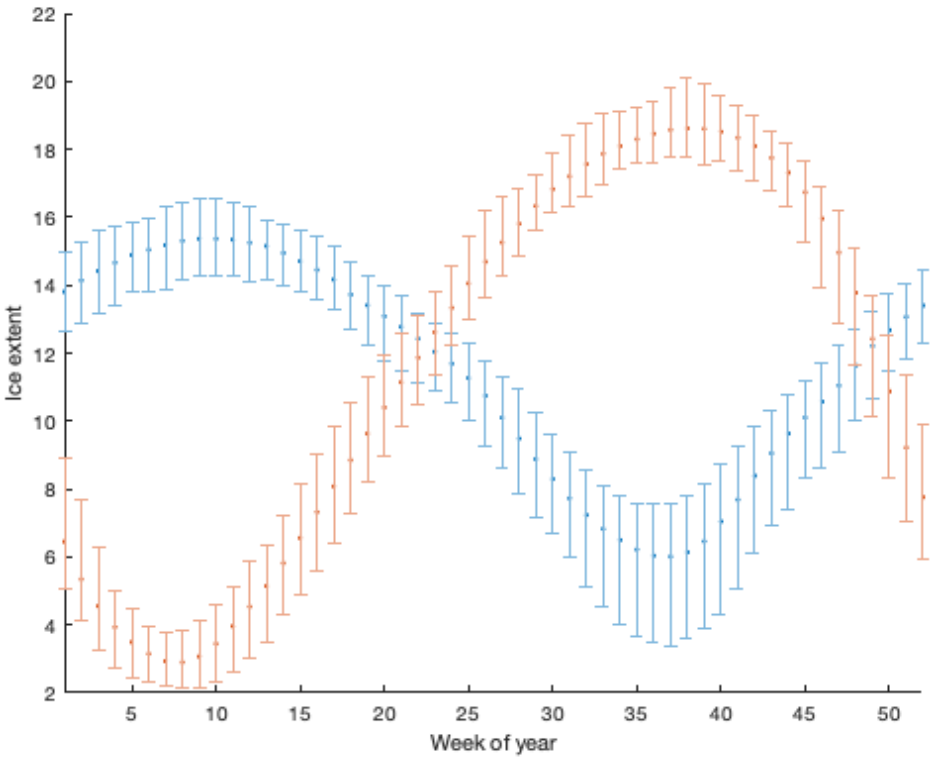


带有误差条的线图：

```
cla

[A, h] = ensemble2bnd(1:52,ice, 'dims', 'xey', 'plot', 'errorbar')
```

```
A =

  struct with fields:



     cent: [52×2 double]

    bndlo: [52×2 double]

    bndhi: [52×2 double]

    errlo: [52×2 double]

    errhi: [52×2 double]

h =

  struct with fields:
```

```
      ln: [2×1 Line]

  errbar: [1×2 ErrorBar]
```

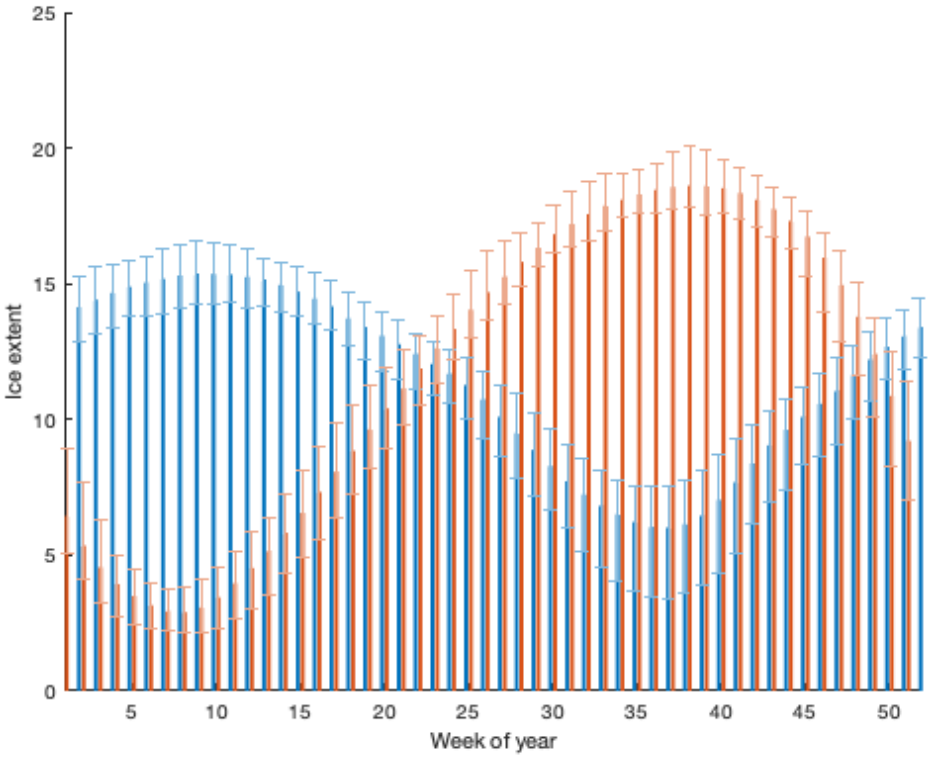


或者带有误差条的柱状图：

```
cla

[A,h] = ensemble2bnd(1:52, ice, 'dims', 'xey', 'plot', 'bar')
set(h.bar, 'edgecolor', 'none');
```

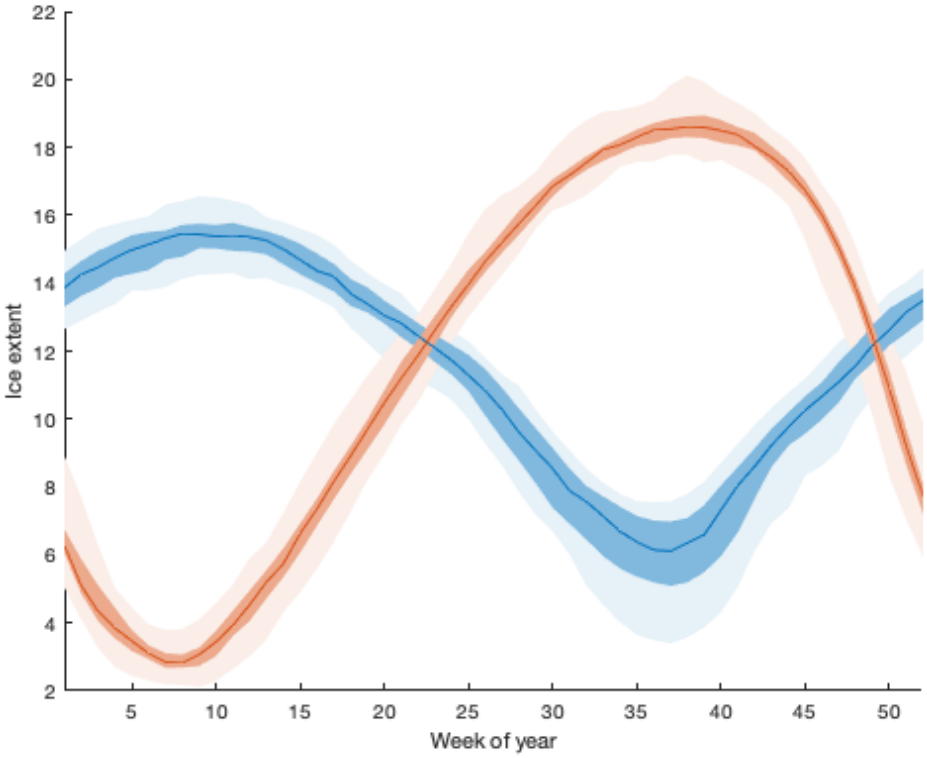
```
A =

  struct with fields:


     cent: [52×2 double]

    bndlo: [52×2 double]

    bndhi: [52×2 double]
```

```
    errlo: [52×2 double]
    errhi: [52×2 double]
h =
  struct with fields:
      bar: [1×2 Bar]
    errbar: [1×2 ErrorBar]
```



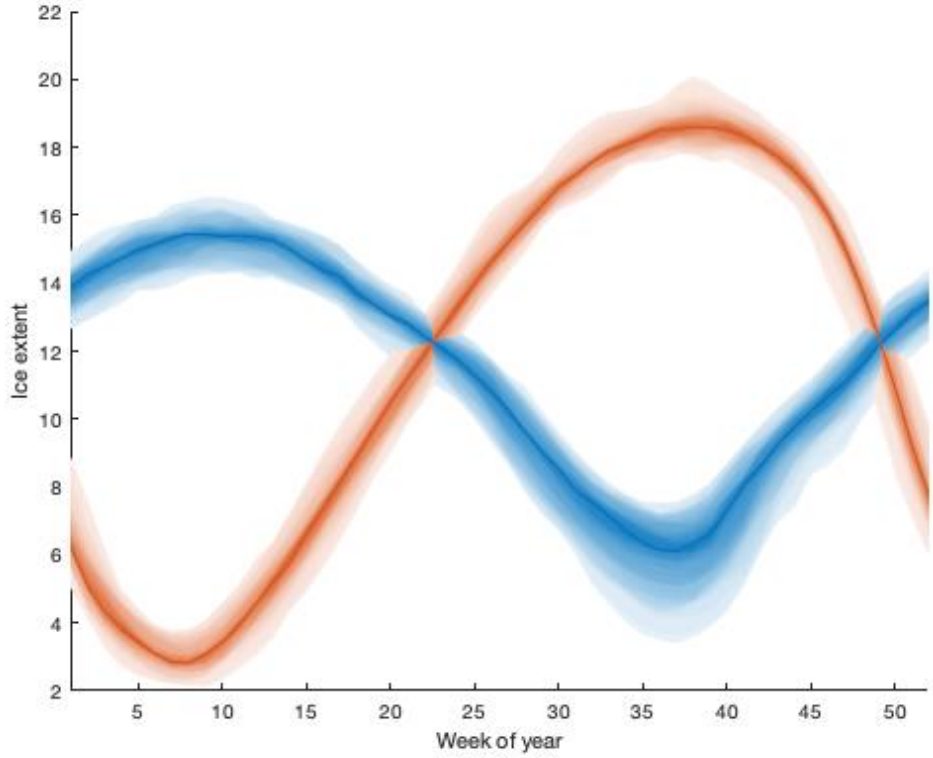
请注意，句柄输出结构体中的字段名称会根据您选择的绘图选项而有所不同。
我们还可以增加显示的间隔数。例如，也许您想查看四分位间距以及最小值和最大值。在这种情况下，将中位数作为中心线而不是均值可能更有意义：

```
cla

ensemble2bnd(1:52,ice, 'dims', 'xey', 'plot', 'boundedline', ...
    'cent', 'median', 'prc', [0 25 75 100]);
```



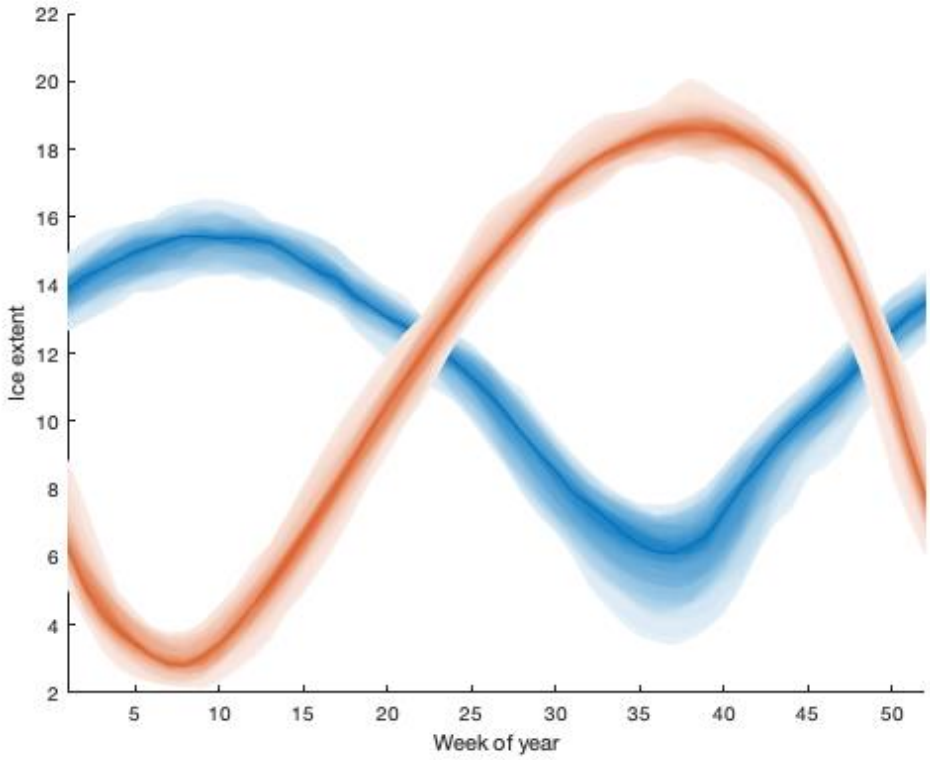
如果添加大量均匀分布的百分位数，则可以模拟阴影分布图。

```
cla

[~,h] = ensemble2bnd(1:52,ice, 'dims', 'xey', 'plot', 'boundedline', ...
    'cent', 'median', 'prc', [0:2:48 52:2:100], 'tlim', [0.1 0.9]);
```



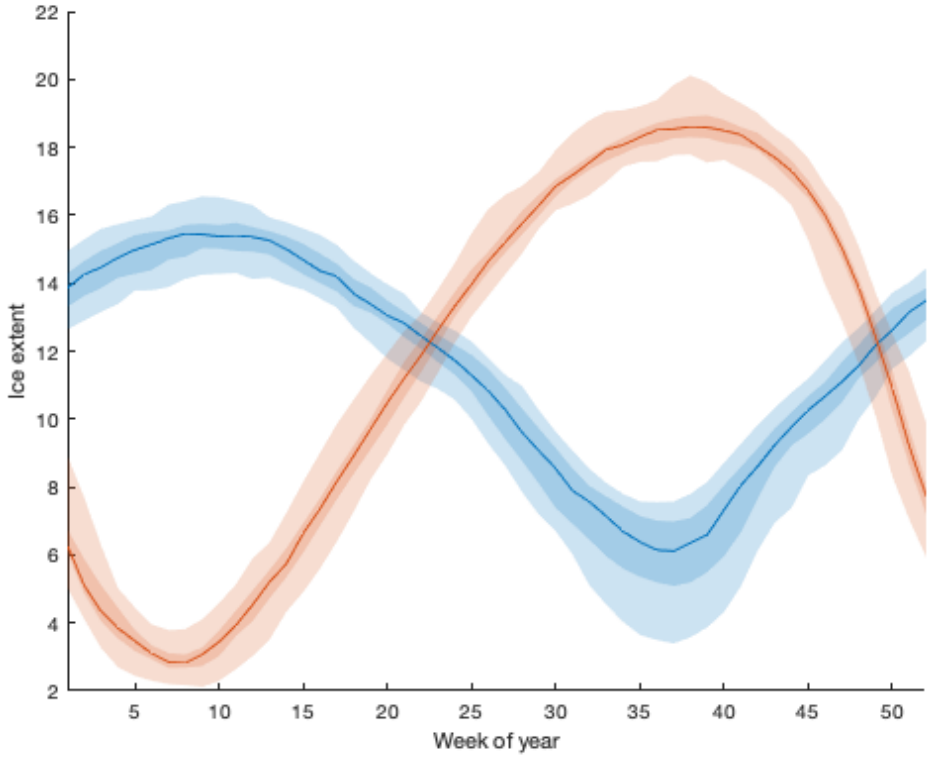
填充对象的绘制顺序从外边界移到内边界，这导致线条重叠交错的外观，但是您可以通过一些重新堆叠来解决此问题：

```
uistack(h.patch(end:-1:1,2), 'top');
```



作为具有不同颜色不透明边界的替代方法，您可以使用透明色块来绘制边界。使用透明色块时，将透明性限制设置为单个值通常很有意义，因为重叠的色块会自动显示附加颜色：

```
cla

[~,h] = ensemble2bnd(1:52,ice, 'dims', 'xey', 'plot', 'boundedline', ...
    'cent', 'median', 'prc', [0 25 75 100], 'tlim', [0.2 0.2], ...
    'alpha', true);
```

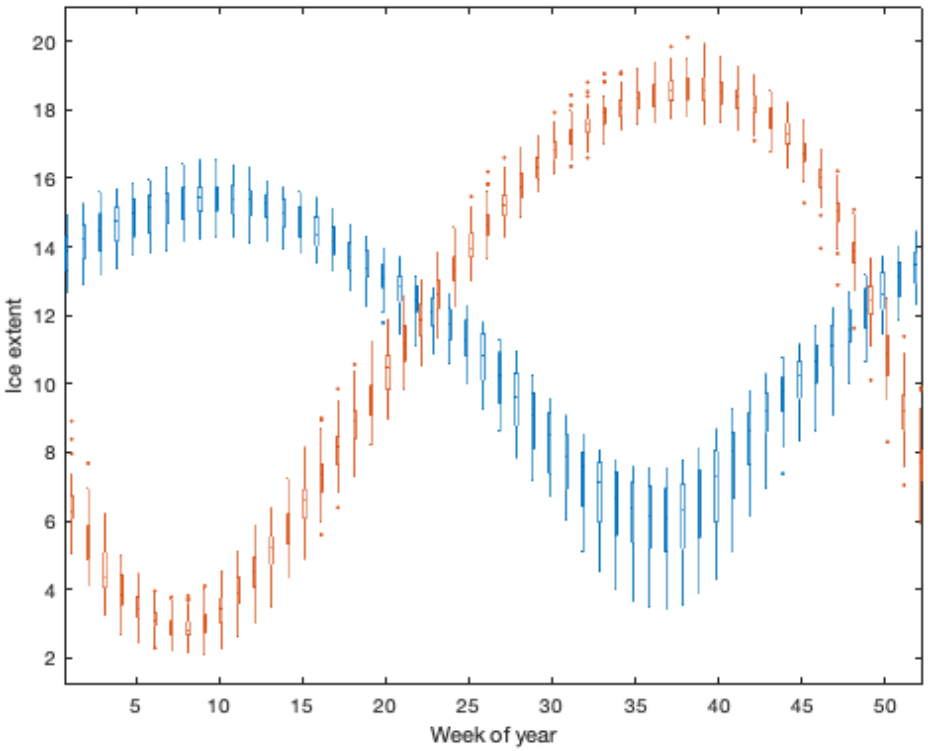


boxplot plotting 选项的工作方式与其他选项略有不同，因为您无法指定要绘制的确切百分位范围。相反，此选项使用常规的箱线图统计信息（请注意，此选项需要统计工具箱（ Statistics Toolbox,），因为它使用了该工具箱中的箱线图函数）。像条形图选项一样，此版本并排放置数据集，以避免过多的重叠。
至少可以说，boxplot 函数返回的图形句柄是…不直观的。如果要更改或编辑属性，建议使用标签：

```
cla

[A,h] = ensemble2bnd(1:52,ice, 'dims', 'xey', 'plot', 'boxplot')

set(findall(h.box, 'tag', 'Upper Whisker'), 'linestyle', '-');
set(findall(h.box, 'tag', 'Lower Whisker'), 'linestyle', '-');
```

```
A =

  struct with fields:


    cent: [52×2 double]

   bndlo: [52×2×2 double]

   bndhi: [52×2×2 double]

   errlo: [52×2×2 double]
```

```
    errhi: [52×2×2 double]

h =

  struct with fields:


    box: [7×2×52 double]
```



作者简介

ensemble2bnd 函数和支持文档是华盛顿大学的 Kelly A. Kearney 写的。 ensemble2bnd 函数是 Climate Data Toolbox for Matlab 的一部分。

trend 文档

trend 使用最小二乘法计算数据序列的线性趋势。数据不需要时间等间隔。
另请参见 detrend3。

语法

```
tr = trend(y)
tr = trend(y,Fs)
tr = trend(y,t)
tr = trend(...,'dim',dim)
tr = trend(...,'omitnan')
[tr,p] = trend(...)
[tr,p] = trend(...,corrOptions)
```

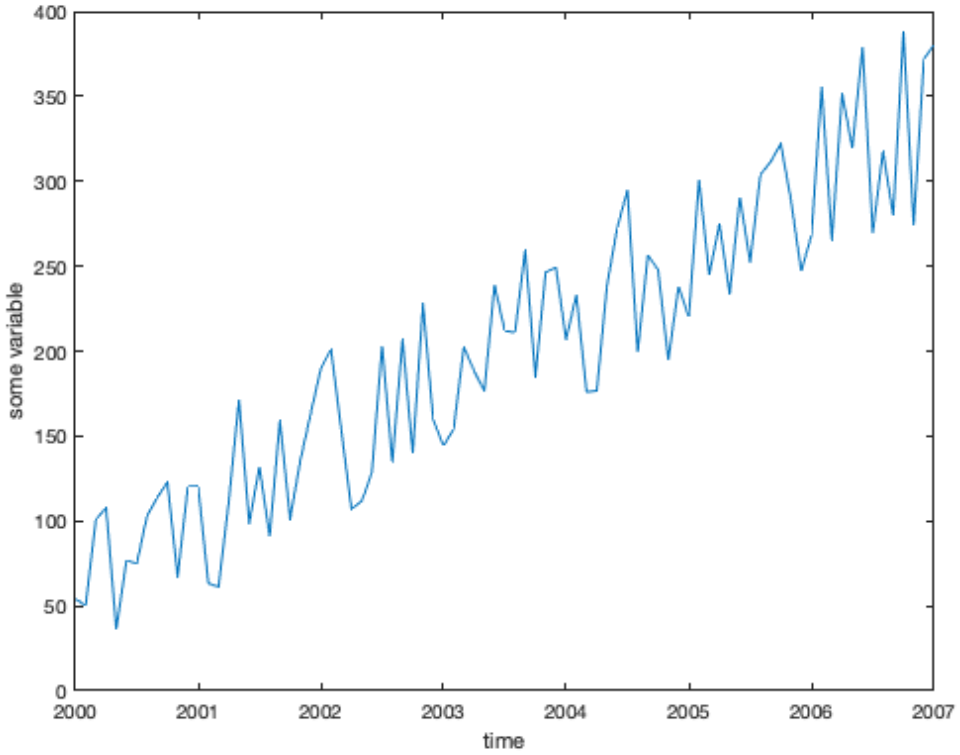
说明

tr = trend(y) 计算每个 y 样本的线性趋势。
tr = trend(y,Fs) 指定采样率 Fs。例如，要从以月分辨率收集的数据获取每年的趋势，请将 Fs 设置为等于 12。此语法假定 y 中的所有值在时间等间隔。
tr = trend(y,t) 指定相对于向量 t 的计算趋势。t 的每个元素对应于 y 中的度量，当使用此语法时，时间不必等距间隔。趋势的单位是（y 单位）/单位（t），因此，如果 t 的单位是天（例如 datenum），请乘以 365.25，以获取每年的趋势。
tr = trend(...,'dim',dim) 指定用于计算趋势的维度。默认情况下，如果 y 是一维数组，则沿 y 的第一个非单维度计算趋势；否则，沿 y 的第一个非单维度进行计算。如果 y 是二维矩阵，则沿 y 的行（维度 1）向下计算趋势；如果 y 是三维矩阵，则趋势将沿第三维度进行计算。
tr = trend(...,'omitnan') 求解最小二乘趋势，即使不是 y 的所有值都是有限的也是如此。如果许多网格单元包含一些（但不是全部）NaN，则此选项可能会比较慢。使用'omitnan'选项时要小心：仅在存在有限数据的时间范围内计算趋势。因此，例如，如果某些网格单元仅包含 10 年记录中的一年的有限数据，则趋势函数所报告的表观“10 年”趋势可能实际上可能是混叠信号。因此，仅当 NaN 在整个时间记录中散布得相当均匀时，才应使用'omitnan'选项。
[tr,p] = trend(...) 返回趋势的统计显著性的 p 值。（需要统计工具箱）
[tr,p] = trend(...,corrOptions) 指定 corr 函数附带的任何可选的“名称/值”对自变量。例如，'Type','Kendall'指定计算 Kendall tau 相关系数。

示例 1：一维数组

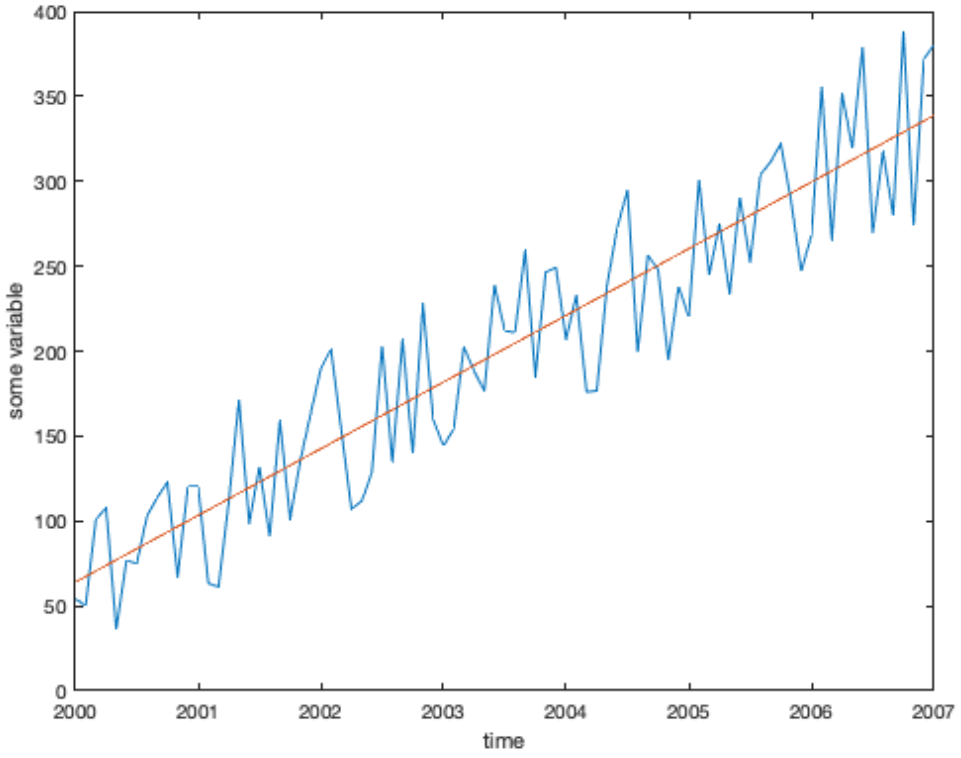
这是每年 12 次采样的时间序列：

```
Fs = 12;                    %采样率（每年 12 个）
t = (2000:1/Fs:2007)';      %每单位时间以 Fs 采样的时间向量
y = 40*t + 123*rand(size(t)) - 8e4; %y 每秒 40 个单位的强制趋势

plot(t,y)
xlabel 'time'
ylabel 'some variable'
```



像这样用 polyplot 在趋势图上叠加趋势线很容易：

```
hold on
polyplot(t,y,1)
```



...但是那个趋势的数值是多少？ 换句话说，坡度有多陡？ 当我们定义 y 时，我们指定它应为 40*t（加上一些随机噪声和偏移量），因此趋势应约为 40：

```
trend(y)
```

```
ans =

    3.2710
```

这个趋势甚至没有接近 40！这是因为，如果不指定采样频率或时间向量，则趋势函数会以每个样本的 y 表示趋势。指定每年 12 个样本的采样率，如下所示：

```
trend(y,Fs)
```

```
ans =

   39.2522
```

...这就是我们期望的接近 40 的值。由于我们有意添加了噪声，所以它不完全是 40，但正如预期的那样，它大约是 40。
您现在可能已经注意到，可以通过三种方式中的任何一种来计算趋势，并且每种方式都将给出相同的答案。您可以将趋势乘以采样率，或者可以在趋势函数中指定采样率，或者可以指定相应的时间向量 t，它们都将给出相同的答案：

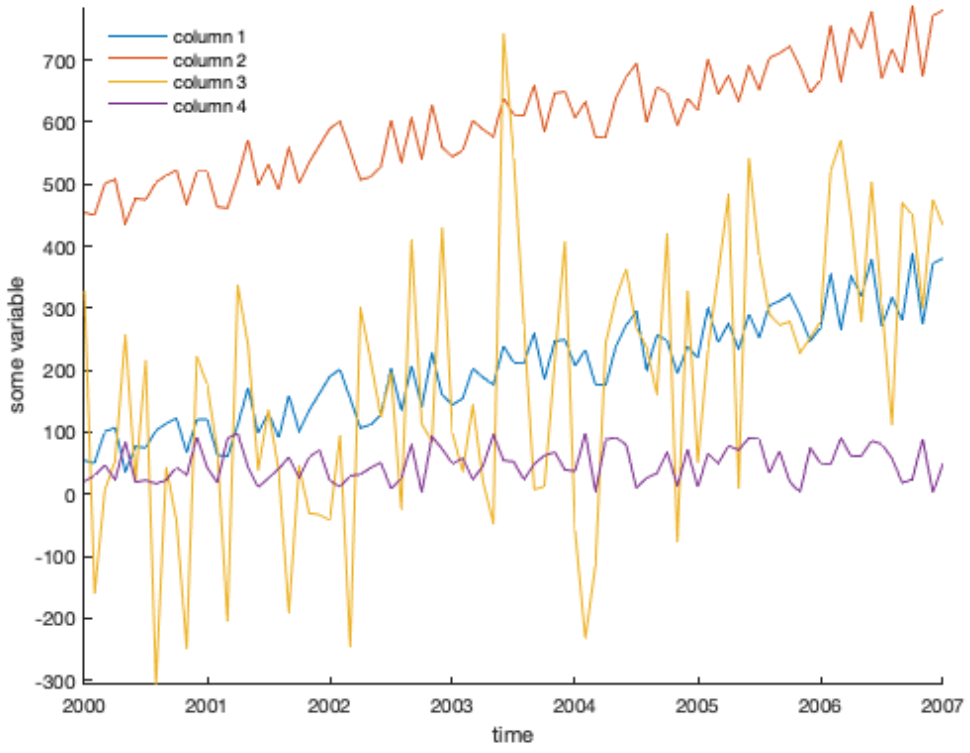
```
[trend(y)*Fs trend(y,Fs) trend(y,t)]
```

```
ans =

   39.2522   39.2522   39.2522
```

示例 2：二维数据

如果您有一堆都排在同一时间的测量值，则趋势函数可以以计算高效的方式一次计算所有趋势。考虑从我们在示例 1 中定义的 y 数组构建的这四个时间序列：

```
%   [y     y+400          噪声很强 y              只有噪声      ];
A = [y     y+400      y+200*randn(size(y))     100*rand(size(y))];

figure
plot(t,A)
legend('column 1','column 2','column 3','column 4',...
   'location','northwest')
xlabel 'time'
ylabel 'some variable'
axis tight
box off       % 移除坐标轴的丑边框
legend boxoff % 移除图例的丑边框
```



每列的趋势可以这样计算：

```
trend(A,t)
```

```
ans =

   39.2522   39.2522   56.5252    2.6557
```

这就是我们的预期：第 1 列和第 2 列每年大约 40 个单位，除了偏移量之外，这些数据都是相同的；第 3 列有一些趋势，而仅噪声的第 4 列则没有太大趋势。
这些趋势在统计上是否有意义？使用可选的第二个函数输出来查看：

```
[tr,p] = trend(A,t)
```

```
tr =

   39.2522   39.2522   56.5252    2.6557

p =

    0.0000    0.0000    0.0000    0.0732
```

前三列的 p 值表示统计显着性，但是第四列告诉我们，在那里测得的任何趋势都可能是垃圾。
如果您的数据跨时间跨列而不是向下跨行怎么办？ 只需指定尺寸：

```
[tr,p] = trend(A',t,'dim',2)
```

```
tr =

   39.2522

   39.2522

   56.5252

    2.6557

p =

    0.0000

    0.0000

    0.0000

    0.0732
```

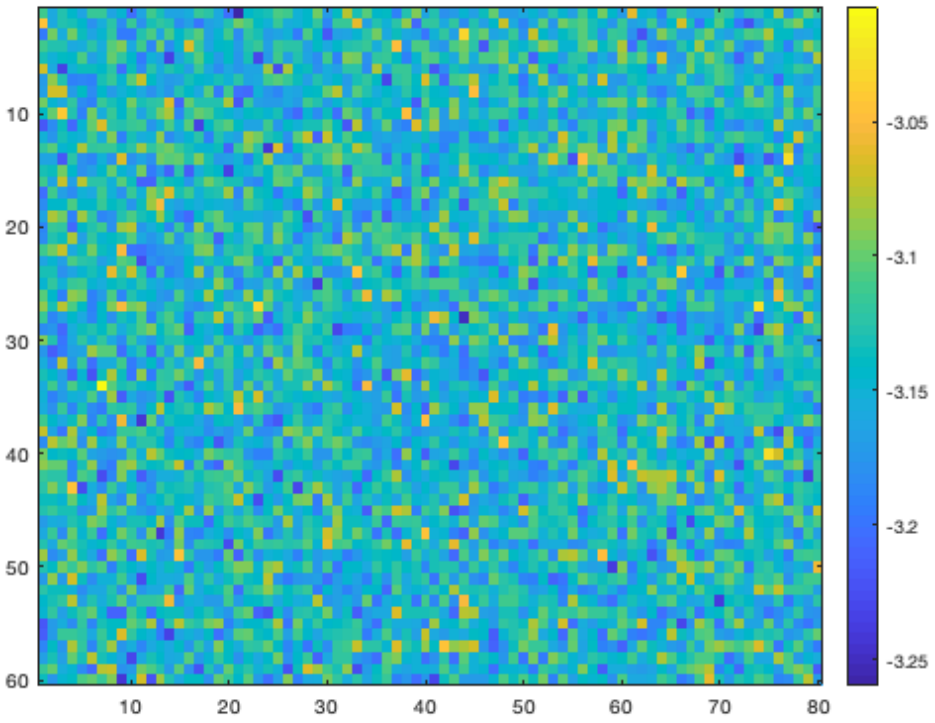
示例 3：一个三维数据集

这是一个示例数据集，每个时间片的下降速率为 pi。使用 expand3 创建示例数据集，然后添加一堆噪点以使事情变得有趣：

```
% 60x80 的网格，每个格子以按照-pi 下降:
Y = expand3(ones(60,80),-pi*(1:100));

%增加一堆噪点和偏移量:
Y = Y + 10*randn(size(Y)) + 900;
```

这是 Y 的趋势:

```
imagesc(trend(Y))

colorbar
```



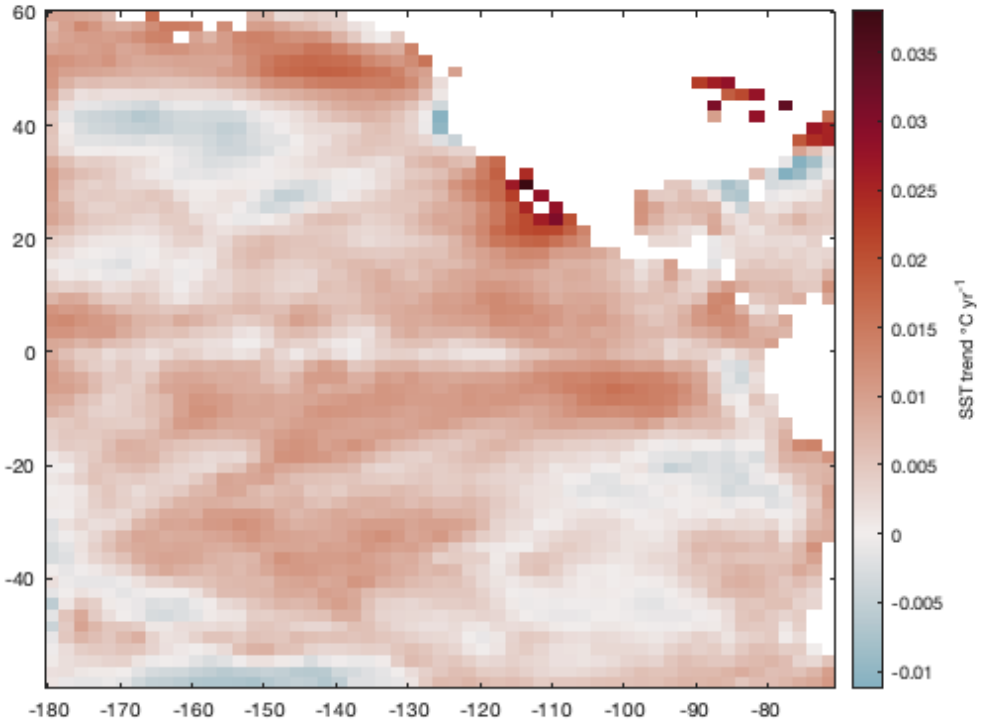
乍一看可能像是噪点，但这只是我们有意放入的噪点。请注意颜色条比例尺-所有值均以负 pi 为中心，完全符合预期。

示例 4a：海表温度

加载样本 pacific_sst.mat 数据集，该数据集包含每月栅格化的海面温度数据，并计算趋势。

```
load pacific_sst
```

```
[tr,p] = trend(sst,12);

figure
imagescn(lon,lat,tr)
cb = colorbar;
ylabel(cb,'SST trend \circC yr^{-1}')
cmocean('balance','pivot') %将颜色图设置为 0 在中间
```

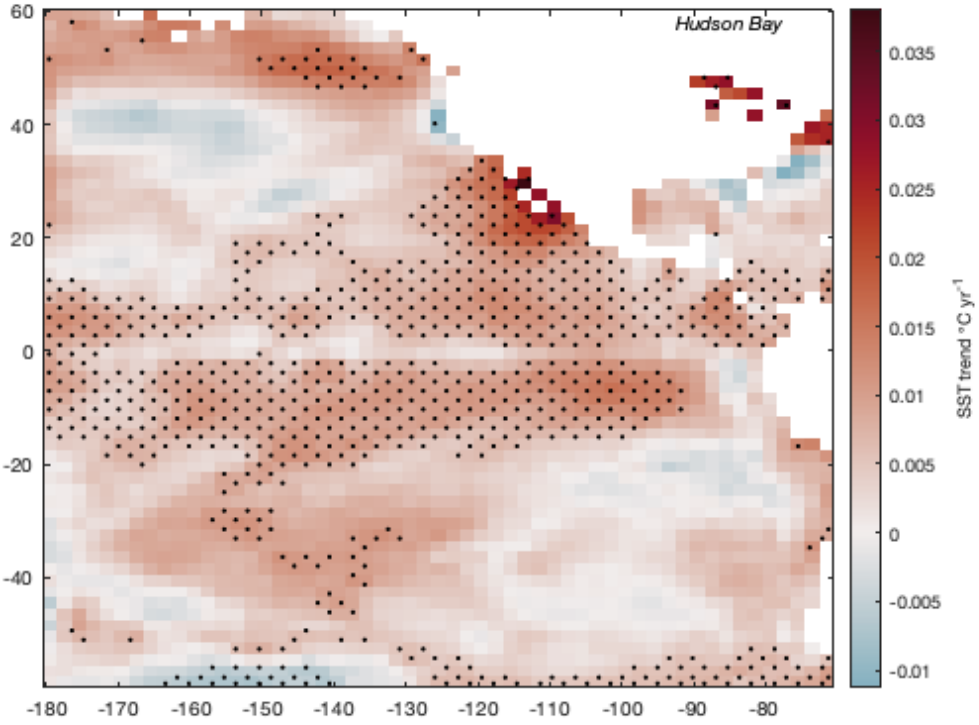


点画标记具有统计意义的区域。首先，这意味着定义统计意义。假设 p 值小于 0.01 的所有数据在统计上都是有意义的：

```
StatisticallySignificant = p<0.01;
```

用 stipple 函数识别重要区域。stipple 只能输入网格，这是一个小问题，因此我们必须将那些经纬度数组转换为带有 meshgrid 的网格：

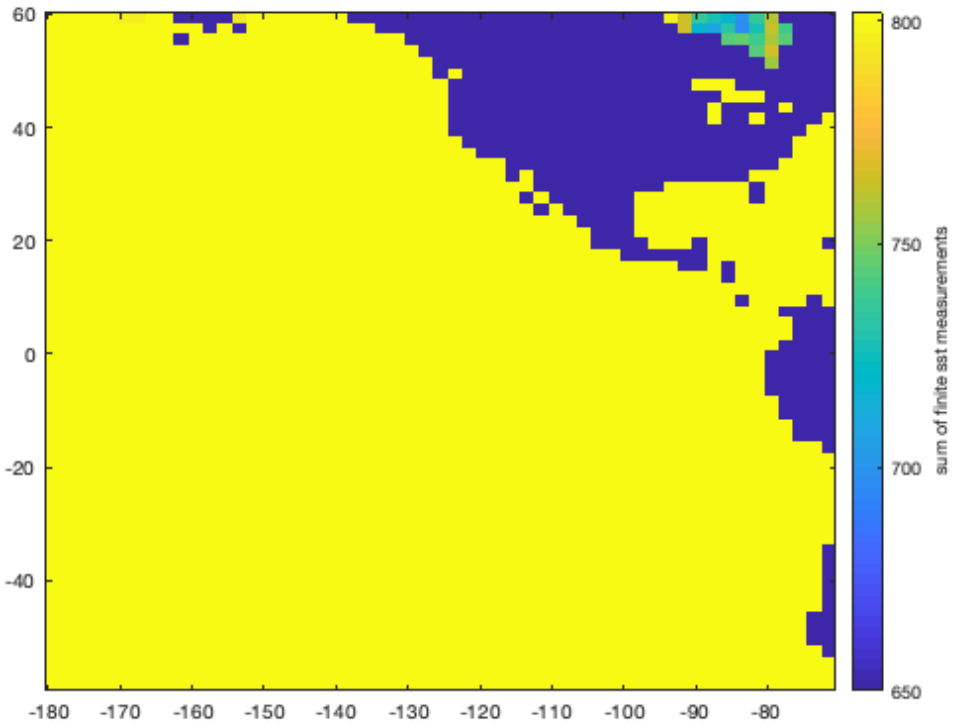
```
[Lon,Lat] = meshgrid(lon,lat);



hold on
stipple(Lon,Lat,StatisticallySignificant)
text(-85,60,'Hudson Bay','vert','top','horiz','center','fontangle','italic')
```



示例 4b：'omitnan'选项

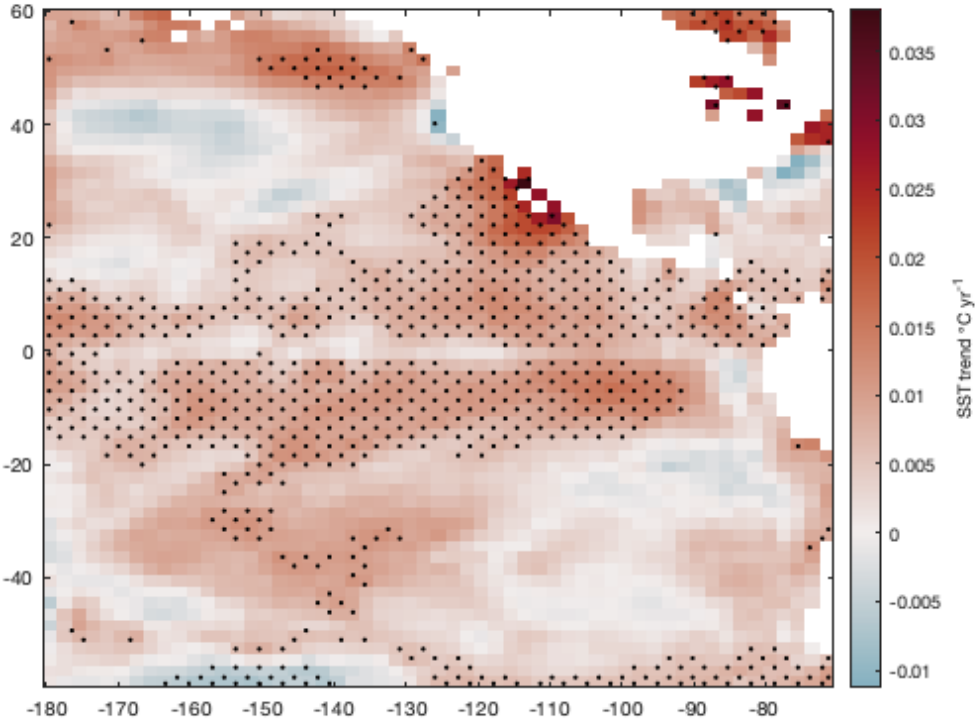
敏锐的观察者可能已经注意到，在上图中，哈德逊湾的趋势是不确定的。但是，那里确实有很多好的数据！查看一下 SST 数据集中的有限测量总数：

```
figure

imagescn(lon,lat,sum(isfinite(sst),3))

cb = colorbar;

ylabel(cb,'sum of finite sst measurements')
caxis([650 802])
```



在上图中，我们看到哈德逊湾有 600 多个良好的 SST 测量值，即使这还不是 802 个月的完整记录，但计算最小二乘趋势仍应足够。在这种情况下，我们可以使用'omitnan'选项：I

```
[tr,p] = trend(sst,12,'omitnan');

figure
imagescn(lon,lat,tr)
cb = colorbar;
ylabel(cb,'SST trend \circC yr^{-1}')
cmocean('balance','pivot') %将颜色图设置为 0 在中间
hold on
stipple(Lon,Lat,p<0.01) %标记统计显著区域
```



从上方我们可以看到哈德逊湾有变暖的趋势，并且具有统计意义。

作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

polyfitw 文档

polyfitw 计算加权多项式拟合。

语法

```
p = polyfitw(x,y,n)
p = polyfitw(x,y,n,w)
[p,S,mu] = polyfitw(...)
```

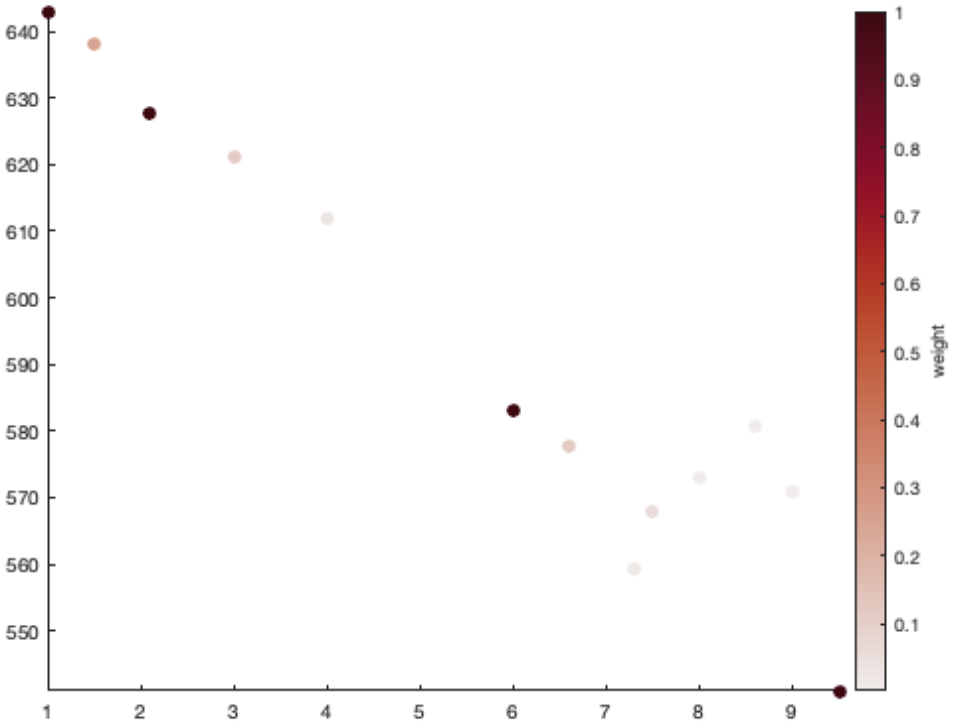
说明

p = polyfitw(x,y,n) 计算 x 与 y 到 n 阶的未加权多项式拟合，与标准 Matlab polyfit 函数完全一样。
p = polyfitw(x,y,n,w) 指定要应用于每个 y 值的权重。对于由 err 给出形式误差估计的测量，请尝试使用权重 w = 1/err.^2。
[p,S,mu] = polyfitw(...) 返回用于 polyval 的 S 结构和居中/缩放值 mu。别忘了，如果您返回一个以上的输出（意味着[p，S，mu]而不只是 p），则 p 中的值将根据 mu 中的值进行缩放。

示例：第一顺序

以下是一些已知斜率为-12 的分散数据，以及一些与每次测量相关的规定误差：

```
x = [1 1.5 2.1 3 4 6 6.6 7.3 7.5 8 8.6 9 9.5];

err = [1 2 -1 3 6 1 3 -7 4 15 30 25 1];



y = 654 - 12*x + err;



% 由 err 给出的权重:
w = 1./err.^2;

figure
scatter(x,y,50,w,'filled')
hold on
axis tight
cb = colorbar;
ylabel(cb,'weight')
cmocean amp
```



这是使用标准 Matlab 函数 polyfit 查找线的未加权斜率的方法

```
p = polyfit(x,y,1)
```

```
p =

        -10.35          650.99
```

尽管我们施加的斜率是-12，这告诉我们未加权的斜率是-10.35。差异是由于测量误差引起的。值得注意的是，您可以使用 CDT 的 trend 函数获得相同的答案：

```
trend(y,x)
```

```
ans =

        -10.35
```

-10.35 斜率值和施加的-12 斜率之间的不匹配是由于测量误差引起的。幸运的是，我们知道每个测量要赋予多少权重，我们可以使用 polyfitw 相应地权重：

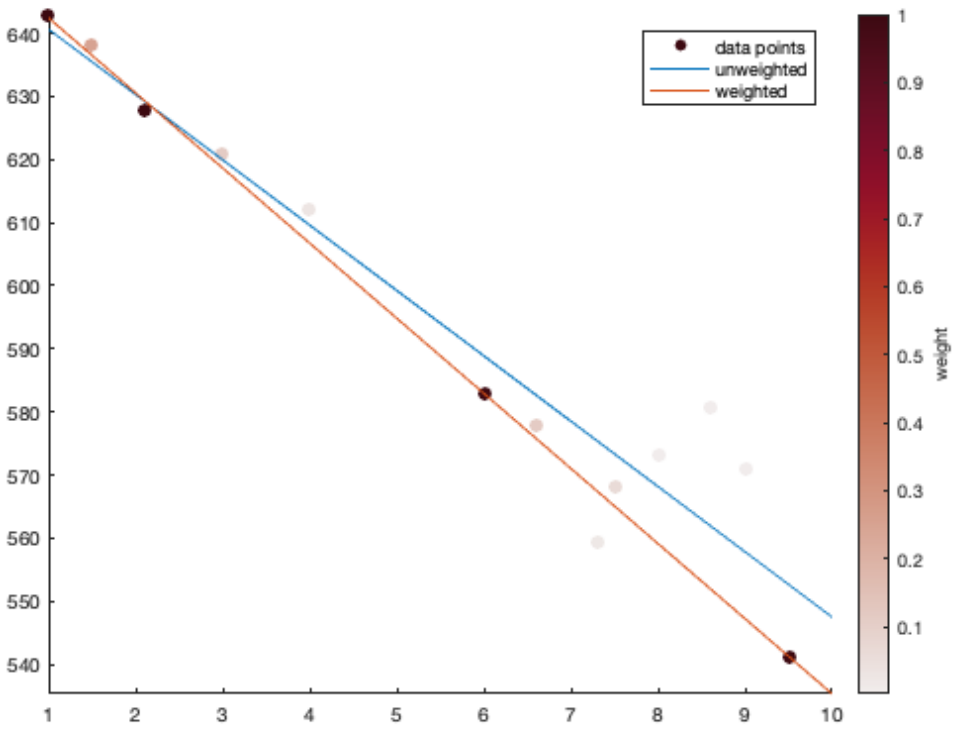
```
pw = polyfitw(x,y,1,w)
```

```
pw =
```

```
   -11.90          654.34
```

现在-11.9 并不完全是我们施加的-12 斜率，但这比未加权趋势的估计值更接近。区别见此：

```
xi = 1:10;



hold on
plot(xi,polyval(p,xi))
plot(xi,polyval(pw,xi))

legend('data points','unweighted','weighted')
```



作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

detrend3 文档

detrend3 沿矩阵的第三维执行线性最小二乘法去趋势化。
另请参见 trend。

语法

```
Ad = detrend3(A)
Ad = detrend3(A,t)
Ad = detrend3(...,'omitnan')
```

说明

Ad = detrend3(A) 从 A 的第三维中去掉线性趋势，假设 A 的切片以恒定的间隔进行采样。
Ad = detrend3(A,t) 指定与 A 的每个切片相关联的时间 t。时间 t 不必以规则的时间间隔出现。
Ad = detrend3(...,'omitnan') 即使在包含 NaN 值的网格单元中，也会应用去趋势。如果许多网格单元包含虚假的 NaN，则您可能会发现此选项比默认选项慢。

示例 1：一个三维网格化数据集

此示例数据集的趋势为 3.2 个单位/时间步：

```
t = 50:50:1000;

y = 3.2*t + 0.1*randn(size(t)); % 趋势是 3.2*t (增加了一些随机噪声)

% 创建 5x5 网格/由 y 决定的趋势:
Z = expand3(ones(5),y);
```

这是通过 trend 趋势函数证明的：

```
trend(Z,t)
```

```
ans =

    3.2000    3.2000    3.2000    3.2000    3.2000

    3.2000    3.2000    3.2000    3.2000    3.2000

    3.2000    3.2000    3.2000    3.2000    3.2000

    3.2000    3.2000    3.2000    3.2000    3.2000

    3.2000    3.2000    3.2000    3.2000    3.2000
```

现在来去掉趋势：

```
Z_detrend = detrend3(Z,t);
```

现在趋势是什么呢？

```
trend(Z_detrend,t)
```

```
ans =

   1.0e-14 *

   -0.2212   -0.2212   -0.2212   -0.2212   -0.2212

   -0.2212   -0.2212   -0.2212   -0.2212   -0.2212

   -0.2212   -0.2212   -0.2212   -0.2212   -0.2212

   -0.2212   -0.2212   -0.2212   -0.2212   -0.2212

   -0.2212   -0.2212   -0.2212   -0.2212   -0.2212
```

只是数值噪声（请注意，它在 1.0e-14 级别为零）。

示例 2：海面温度

本示例根据网格化时间序列计算面积平均海面温度异常，然后对网格化的 sst 时间序列进行去趋势化，并绘制去趋势化的时间序列。

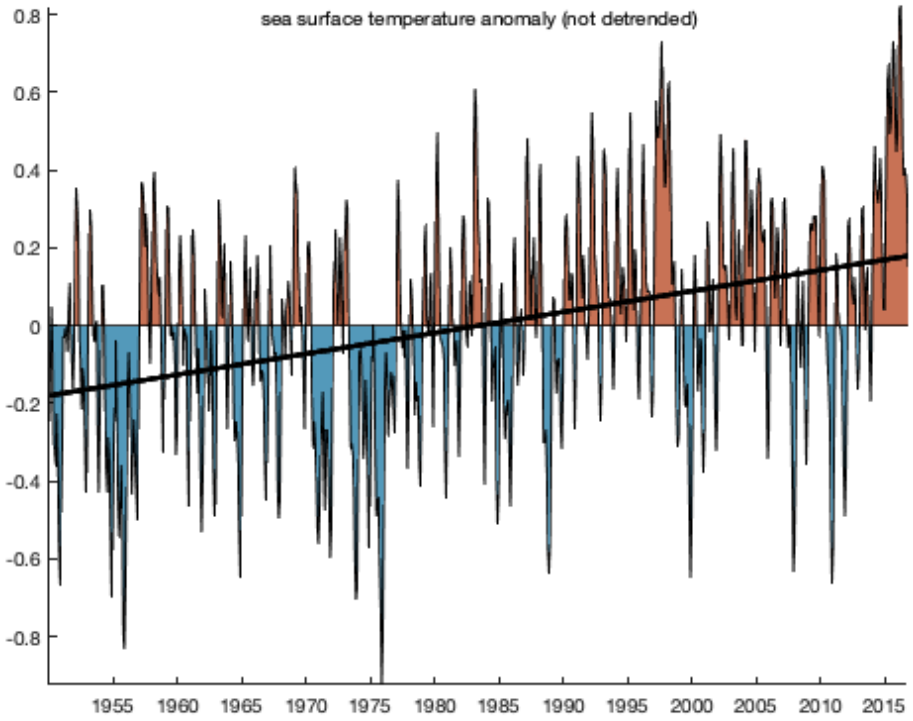
首先加载每月 SST 的样本 pacific_sst.mat 数据集。使用 cdtarea 获取每个网格单元的面积，然后使用 local 获取所有海洋网格单元的一维时间序列的面积加权 sst。

```
%载入示例数据:
load pacific_sst

%将经纬度数组转为网格:
[Lon,Lat] = meshgrid(lon,lat);

%获取每个网格单元的面积（平方米）:
A = cdtarea(Lat,Lon); %网格单元的面积（平方米）

%只屏蔽有限的数据（以有效地忽略土地）
mask = all(isfinite(sst),3);

%获取面积权重 sst 的一维时间序列:
sst_1 = local(sst,mask,'weight',A);
```

现在使用 anomaly 函数绘制面积加权的 sst 异常。对于背景，使用 polyplot 绘制线性趋势：

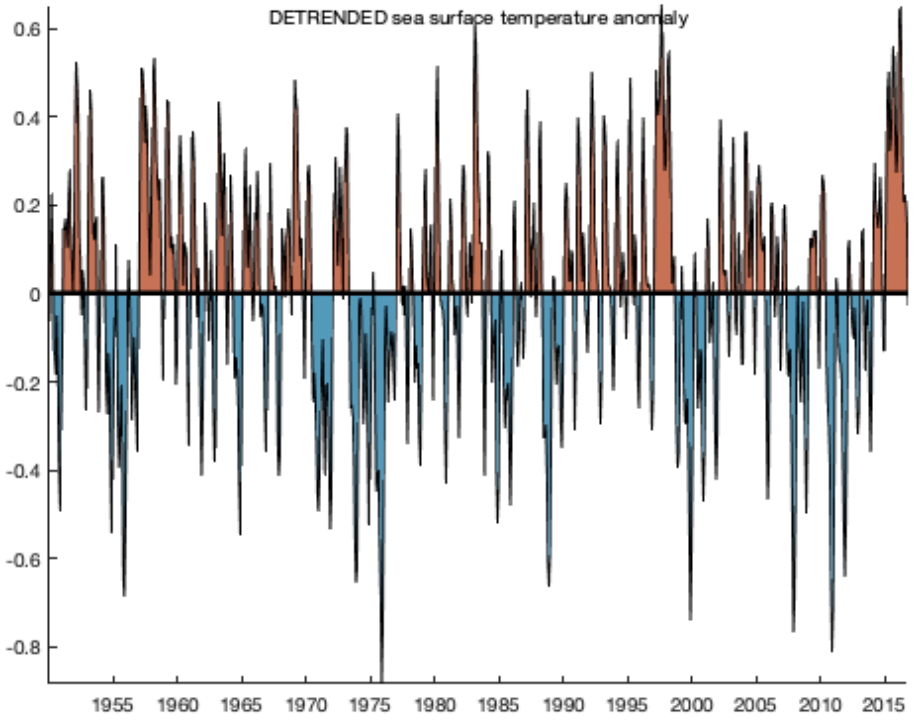
```
figure

anomaly(t,sst_1-mean(sst_1))
```

```
axis tight
datetick('x','keeplimits')
ntitle('sea surface temperature anomaly (not detrended)')

hold on
polyplot(t,sst_1-mean(sst_1),1,'k','linewidth',3)
```



现在对整个三维 sst 数据集进行去趋势处理，并绘制相同的面积平均时间序列，但是这次使用去趋势处理的数据：

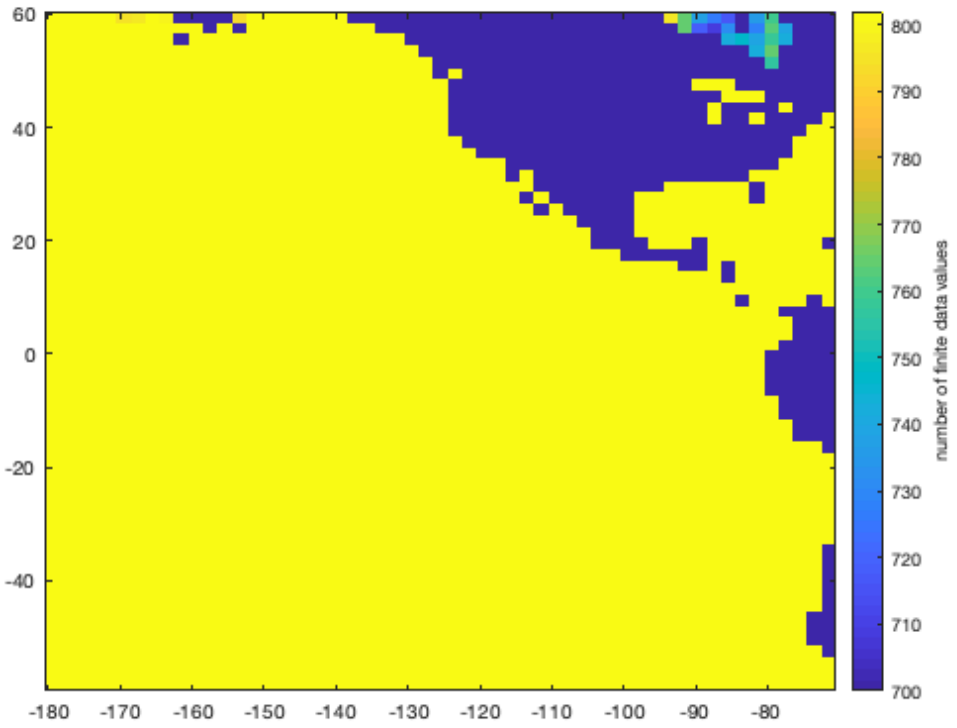
```
%对整个三维 sst 数据集进行去趋势:
sst_dt = detrend3(sst);

%获取面积平均去趋势的 SST 序列:
sst_dt_1 = local(sst_dt,mask,'weight',A);

figure
anomaly(t,sst_dt_1-mean(sst_dt_1))
axis tight
datetick('x','keeplimits')
ntitle('DETRENDED sea surface temperature anomaly')

hold on
polyplot(t,sst_dt_1-mean(sst_dt_1),1,'k','linewidth',3)
```



示例 3：'omitnan'选项

在某些情况下，网格化数据集中可能包含虚假的 NaN。我们在上面的示例中分析的 sst 数据集就是这种情况。对于 802 个月的 SST 数据，请查看有多少个有限值：

```
figure

imagescn(lon,lat,sum(isfinite(sst),3))

cb = colorbar;

ylabel(cb,'number of finite data values')
caxis([700 802]) %设置颜色坐标轴范围
```

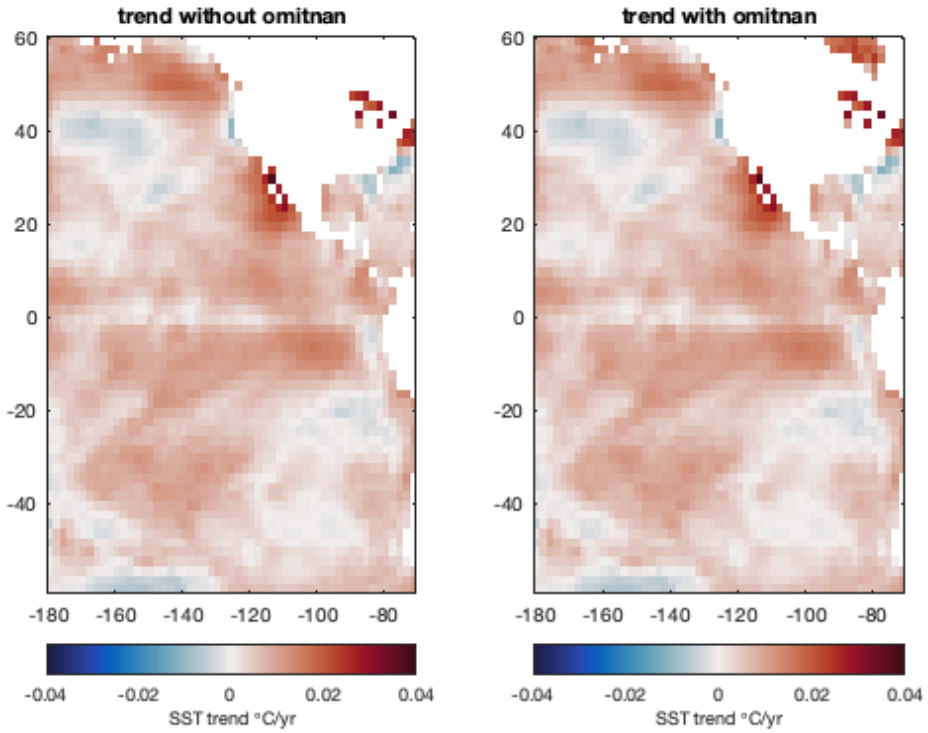


在上图中，看看加拿大的哈德逊湾。您会注意到，并非所有 802 个月中都有有效数据。丢失了多达 100 个月的数据，因为在冬季，如果冰弄坏了测量结果，可能很难获得良好的 SST 读数。不过，您可能仍需要从可用数据中删除长期趋势。

这是对不使用 sst 数据集和使用'omitnan'选项的趋势进行比较的比较。首先使用 trend 函数，只需评估趋势的大小即可：

```
figure

subplot(1,2,1)

imagescn(lon,lat,trend(sst,12))

cb = colorbar('location','southoutside');
xlabel(cb,'SST trend \circC/yr')
title 'trend without omitnan'
caxis([-0.04 0.04])
cmocean('balance')

subplot(1,2,2)
imagescn(lon,lat,trend(sst,12,'omitnan'))
cb = colorbar('location','southoutside');
xlabel(cb,'SST trend \circC/yr')
title 'trend with omitnan'
caxis([-0.04 0.04])
```

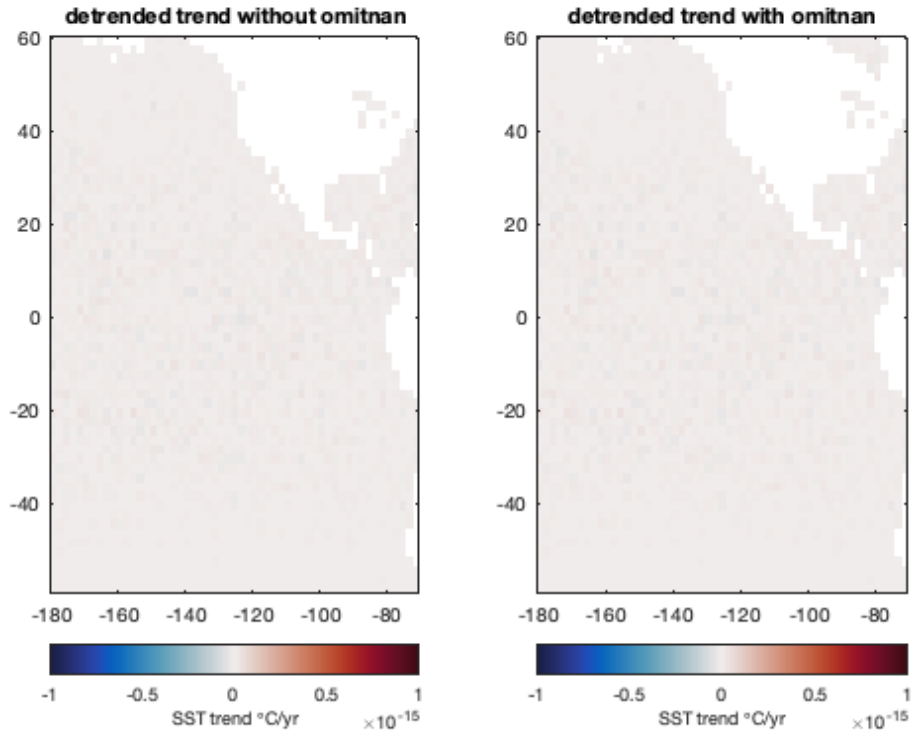
```
cmocean('balance')
```



在上图中，请注意加拿大哈德逊湾的区别。
现在，去趋势的工作方式相同。我们将对每个数据集进行去趋势处理，然后如上所述绘制出被去趋势处理的数据集的趋势图：

```
sst_dt = detrend3(sst);

sst_dt_o = detrend3(sst,'omitnan');

figure
subplot(1,2,1)
imagescn(lon,lat,trend(sst_dt,12))
cb = colorbar('location','southoutside');
xlabel(cb,'SST trend \circC/yr')
caxis([-0.000000000000001 0.000000000000001])
cmocean('balance')
title 'detrended trend without omitnan'

subplot(1,2,2)
imagescn(lon,lat,trend(sst_dt_o,12,'omitnan'))
cb = colorbar('location','southoutside');
xlabel(cb,'SST trend \circC/yr')
caxis([-0.000000000000001 0.000000000000001])
```

```
cmocean('balance')
title 'detrended trend with omitnan'
```



现在，保留在去趋势数据集中的唯一“趋势”是数字噪声（请注意色标的 1x10 ^ -15 限制。）

作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

monthly 文档

monthly 计算一年中指定月份的变量统计信息。

另请参见: season, climatology, sinefit, sineval , 和 sinefit_bootstrap.

语法

```
Xm = monthly(X,t,months)
Xm = monthly(...,'dim',dimension)
Xm = monthly(...,@fn)
Xm = monthly(...,'omitnan')
Xm = monthly(...,options)
```

说明

Xm = monthly(X,t,months) 返回数字 1 到 12 指定的所有月份的 X 的平均值。例如，如果将月份指定为 1，则 Xm 将是一月份所有 X 值的平均值。如果 month 是[12 1 2]，则 Xm 将是所有 DJF 的平均值。时间 t 对应于 X，并且可以采用 datenum，datetime 或 datestr 格式

Xm = monthly(...,'dim',dimension) 指定操作的维度。默认情况下，如果 X 是一维数组，则假定 t 对应于 X 的第一个非单维度；如果 X 是二维矩阵，则假定 t 对应于 X 的行；如果 X 是三维矩阵，则将时间假定为 X 的第三维度。

Xm = monthly(...,@fn) 指定要应用于 X 的任何函数，例如@max，@std 或您自己的匿名函数。默认函数为@mean。

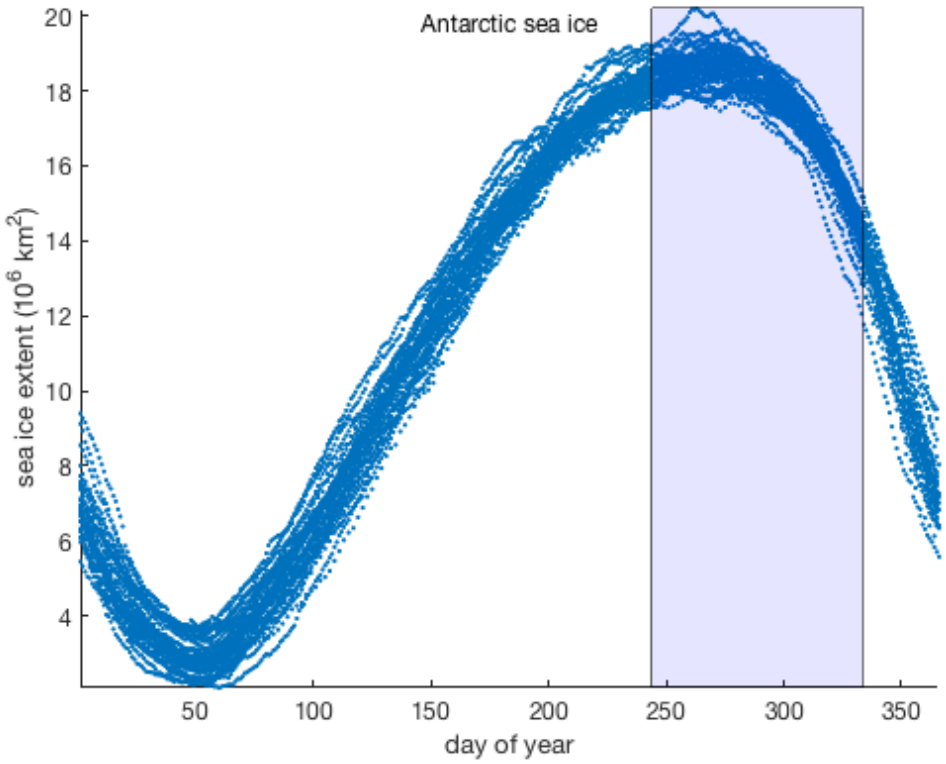
Xm = monthly(...,'omitnan') 在计算中忽略 NaN 值。

Xm = monthly(...,options) 指定任何可选输入，这取决于应用于数据的@fn。

示例 1：春季海冰范围

南极春季平均海冰范围是多少？ 让我们加载示例时间序列并对其进行绘制，以大致了解我们所问的问题：

```
% 加载样本数据:
load seaice_extent

%使用函数绘制一年中一天的海冰范围:
plot(doy(t),extent_S,'.')

hold on    % 允许添加到图上
box off    % 移除外框
axis tight % 移除多余空间
xlabel 'day of year'
ylabel 'sea ice extent (10^6 km^2)'
ntitle 'Antarctic sea ice'

%高亮 9-11 月:
vfill(doy('sept 1'),doy('nov 30'),'b','facealpha',0.1)
```



上面是从 1978 年到现在的每日海冰范围绘制的，它是一年中某天的函数。doy 函数可将日期转换为一年中的某天，ntitle 会创建一个很好地隐藏在轴内的标题，而 hfill 则会创建从 9 月 1 日到 11 月 30 日的阴影区域。

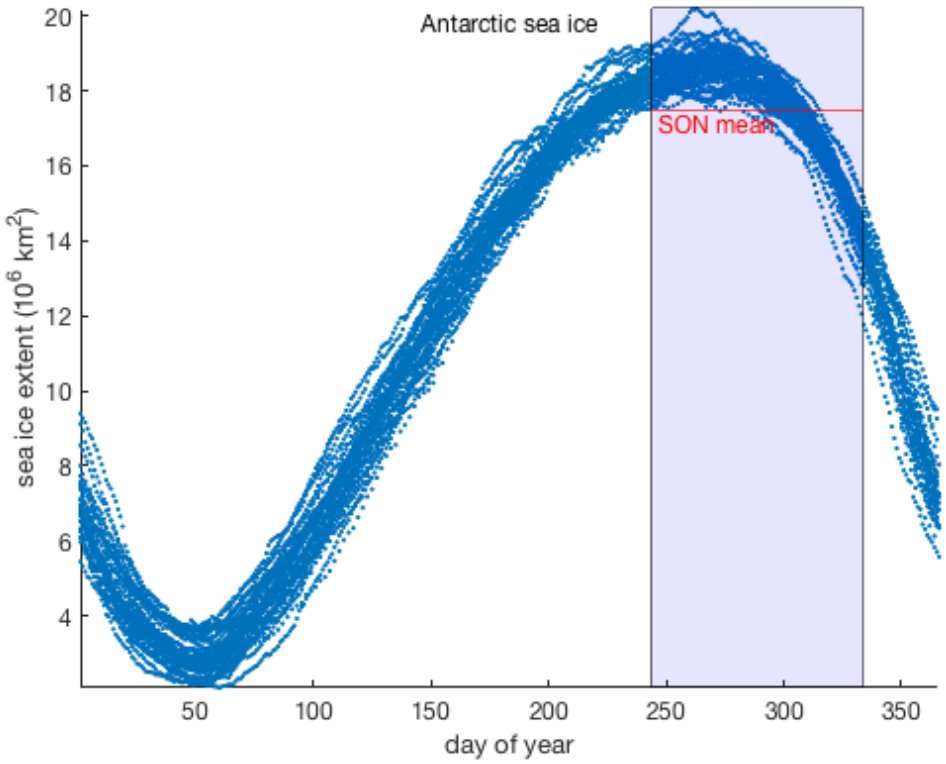
在上图中，我们通过将该区域阴影化，将春季隐式定义为 9 月至 11 月（SON）。现在，我们可以使用月度函数获得这些月份的平均海冰范围，将第 9、10 和 11 月指定为我们感兴趣的月份：

```
monthly(extent_S,t,[9 10 11])
```

```
ans =

        17.5
```

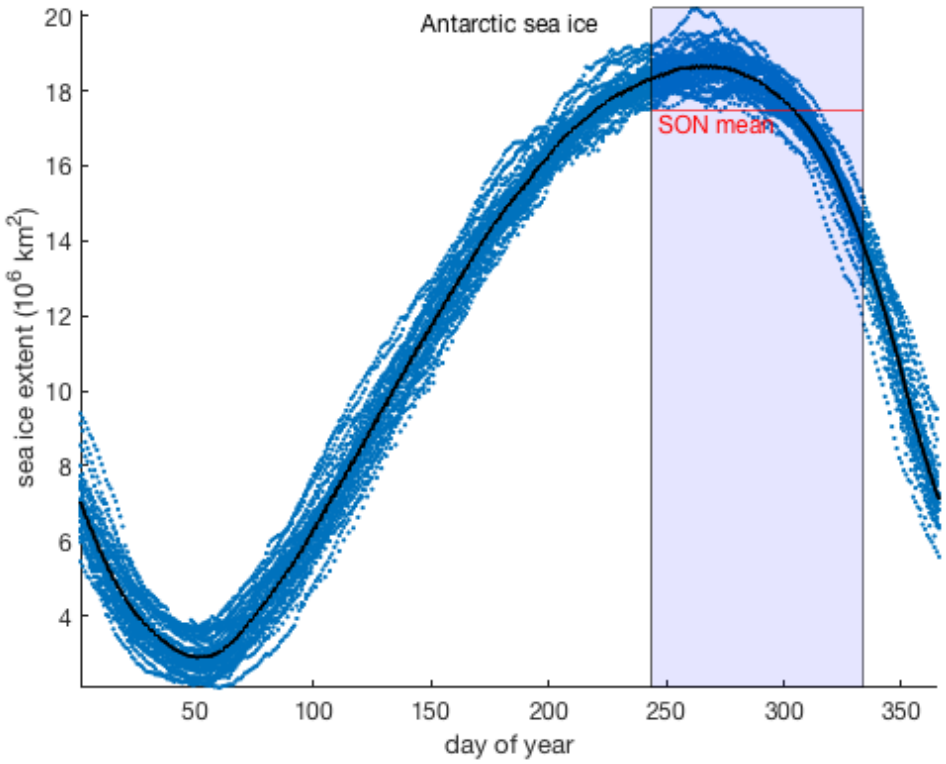
南极春季春季平均海冰范围约为 1750 万平方公里。如果您愿意，我们可以进行绘图：

```
plot([doy('sept 1') doy('nov 30')],[17.5 17.5],'r')

text(doy('sept 1'),17.5,' SON mean','color','r','vert','top')
```



在这里值得注意的是，我们还有另一种方式可以得出类似的答案。我们本可以使用 climatology 来获取海冰时间序列的每日气候，例如：

```
[extent_S_clim,t_clim] = climatology(extent_S,t);



% 使用黑线绘制日均气候态:
plot(t_clim,extent_S_clim,'k','linewidth',2)
```



然后，可以从 doy('sept 1') to doy('nov 30')将平均 SON 值作为所有每日气候值的平均值：

```
mean(extent_S_clim(doy('sept 1'):doy('nov 30')))
```

```
ans =

        17.56
```

示例 2：总降水量:

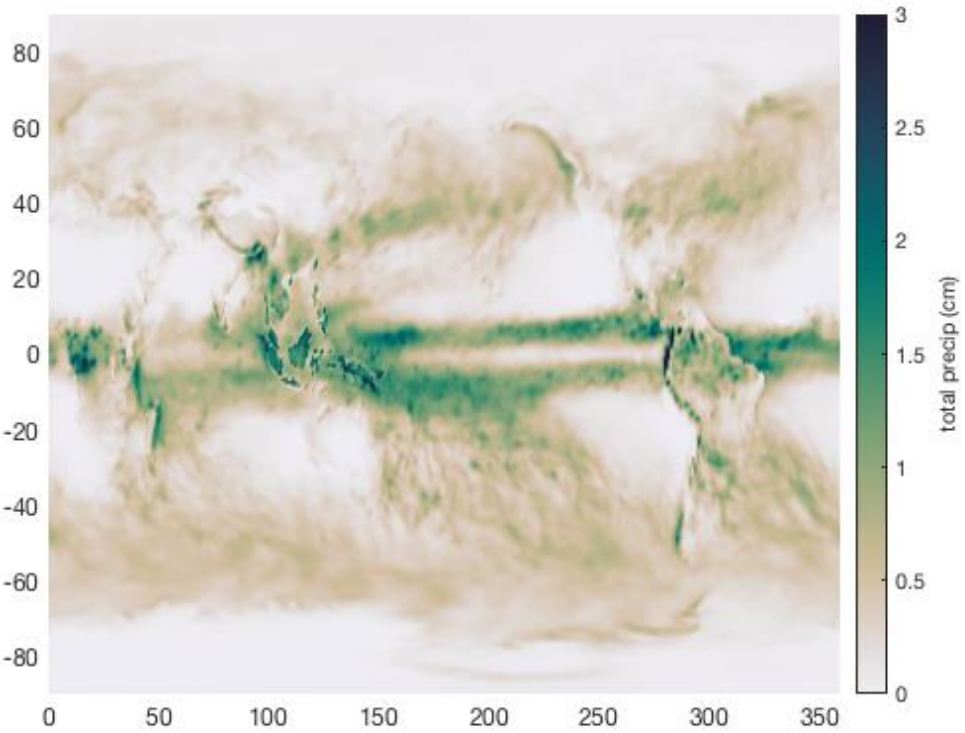
在此示例中，加载一些 ERA-Interim 再分析数据，其中包含 2017 年的每月降水总量：

```
filename = 'ERA_Interim_2017.nc';
lat = ncread(filename,'latitude');
lon = ncread(filename,'longitude');
t = datenum(1900,1,1,double(ncread(filename,'time')),0,0);
tp = ncread(filename,'tp');
```

2017 年 3 月至 2017 年 5 月的降水总量是多少？

```
MAM_sum = monthly(tp,t,3:5,@sum);



figure
```

```
pcolor(lon,lat,MAM_sum'*100)

shading interp
cmocean rain
cb = colorbar;
ylabel(cb,'total precip (cm)')
caxis([0 3])
```



由于 2017 年数据集仅包含 1 年数据，因此将 3 月到 5 月的总和与我们手动将这些月度网格加在一起的情况完全相同：

```
% 比较 monthly 的结果和手动相加的结果:
isequal(MAM_sum, tp(:,:,3)+tp(:,:,4)+tp(:,:,5))
```

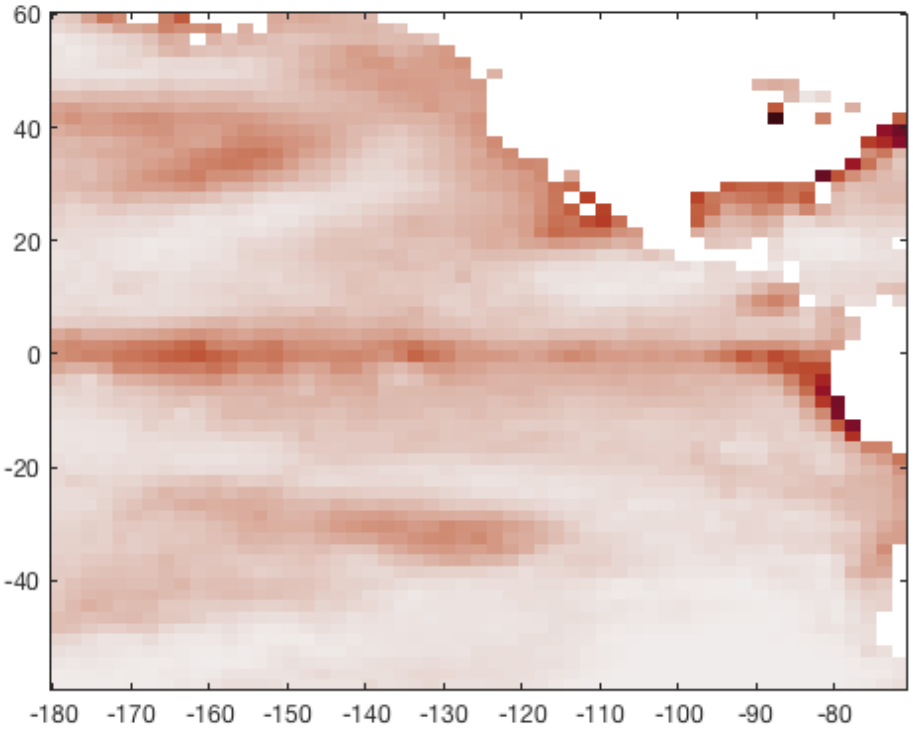
```
ans =

  logical

   1
```

示例 3：年际变化

太平洋三月的可预测性如何？ 我的意思是说，明年三月的海表温度会与去年三月的海表温度大致相同，还是不同三月之间的变化超级大？

要回答此问题，请将 monthly 函数应用于样本 pacific_sst 数据集，该数据集包含 67 年的数据：

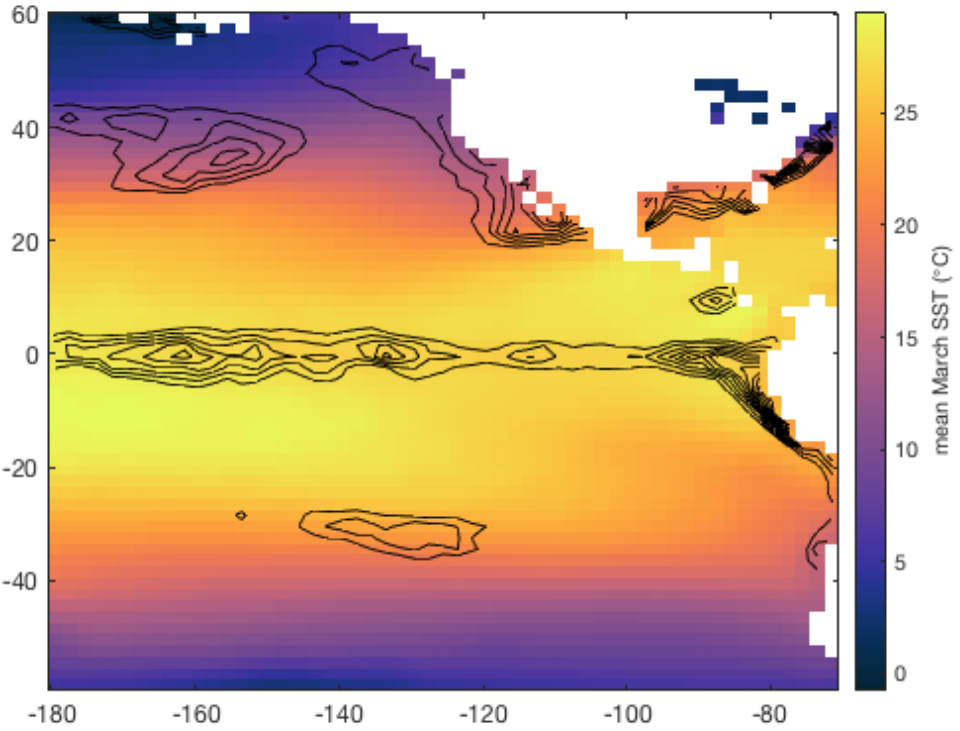
```
load pacific_sst

mar_var = monthly(sst,t,3,@var);

figure
imagescn(lon,lat,mar_var)
caxis([0.05 2])
cmocean amp
```



我们在上图中看到的是，沿海地区，3 月海表温度存在一定的变化，因为每年之间可能会发生显着变化，但是在东南太平洋，3 月海表温度的变化非常低，这意味着 3 月的海表温度中气候学上更容易预测。
让我们将平均三月 SST 与三月 SST 的方差放到地图上：

```
% 计算所有 3 月的平均 SST:
mar_mean = monthly(sst,t,3);

figure
imagescn(lon,lat,monthly(sst,t,3))
cmocean thermal
cb = colorbar;
ylabel(cb,'mean March SST (\circC)')

%用填色图绘制方差:
hold on
```

```
contour(lon,lat,mar_var,0.5:0.1:8,'k')
```



作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

season 文档

Season 估算与年度周期或时间序列相关的异常。

另请参见: deseason, climatology, sinefit, sineval , 和 sinefit_bootstrap.

语法

```
[As,ts] = season(A,t)
[As,ts] = season(...,'daily')
[As,ts] = season(...,'monthly')
[As,ts] = season(...,'detrend',DetrendOption)
[As,ts] = season(...,'dim',dimension)
As = season(...,'full')
```

说明

[As,ts] = season(A,t) 给出时间序列 A 的典型季节（也称为年度）周期，该周期对应于 datenum 或 datetime 格式的时间 t。如果 t 为每天，则输出 ts 为 1 到 366，并且 As 将包含一年中 366 天中每一天的平均值。如果输入为每月，则 ts 为 1:12，而 As 将包含一年中 12 个月中每个月的平均值。

[As,ts] = season(...,'daily') 直接指定输入为每日分辨率。season 函数通常会自动解决此问题，但是如果您的数据中缺少大量空白，您可能希望通过指定每天来确保正确的结果。

[As,ts] = season(...,'monthly') 同上，但强制采用每月的解决方案。

[As,ts] = season(...,'detrend',DetrendOption) 指定相对于其确定季节性异常的基准。选项为“线性”（'linear'），“二次”（'quadratic'）或“无”（'none'）。默认情况下，在删除线性最小二乘趋势后会计算异常，但是如果，例如，加热是强烈非线性的，则您可能更喜欢使用'quadratic'选项。默认为'linear'。

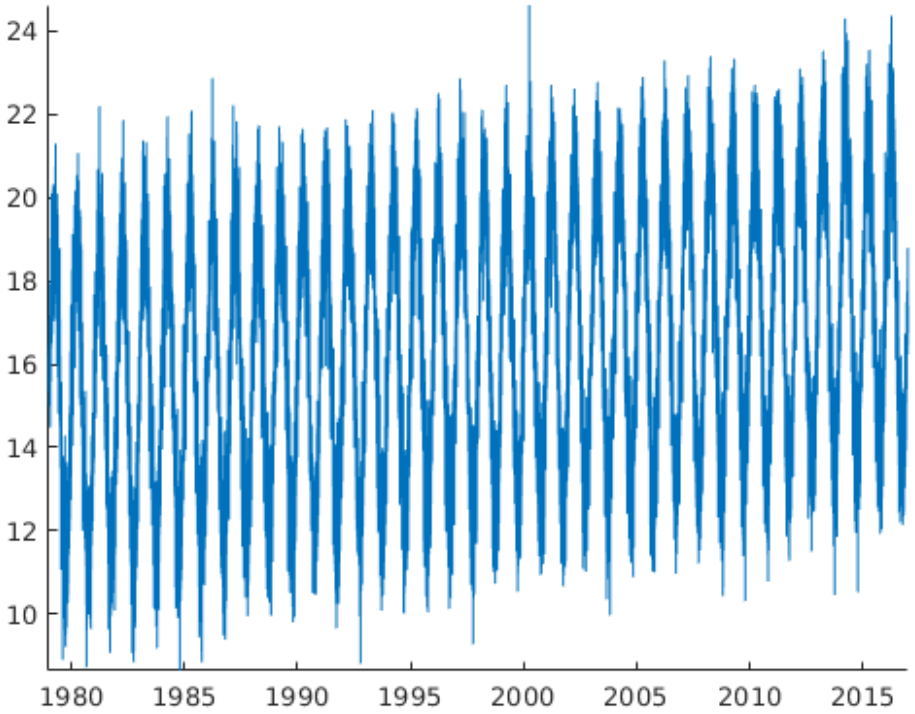
[As,ts] = season(...,'dim',dimension) 指定用于评估季节的维度。默认情况下，如果 A 为一维，则沿非单维度返回季节周期；否则，返回 0。如果 A 是二维，则沿第一维度执行（时间沿着行行进）；如果 A 是三维，则沿第三维度执行。

As = season(...,'full') 返回有关整个时间序列 A 的信息。这是一个方便的选项，可以单独查看长时间序列的组成部分。

示例 1：一个单独的时间序列

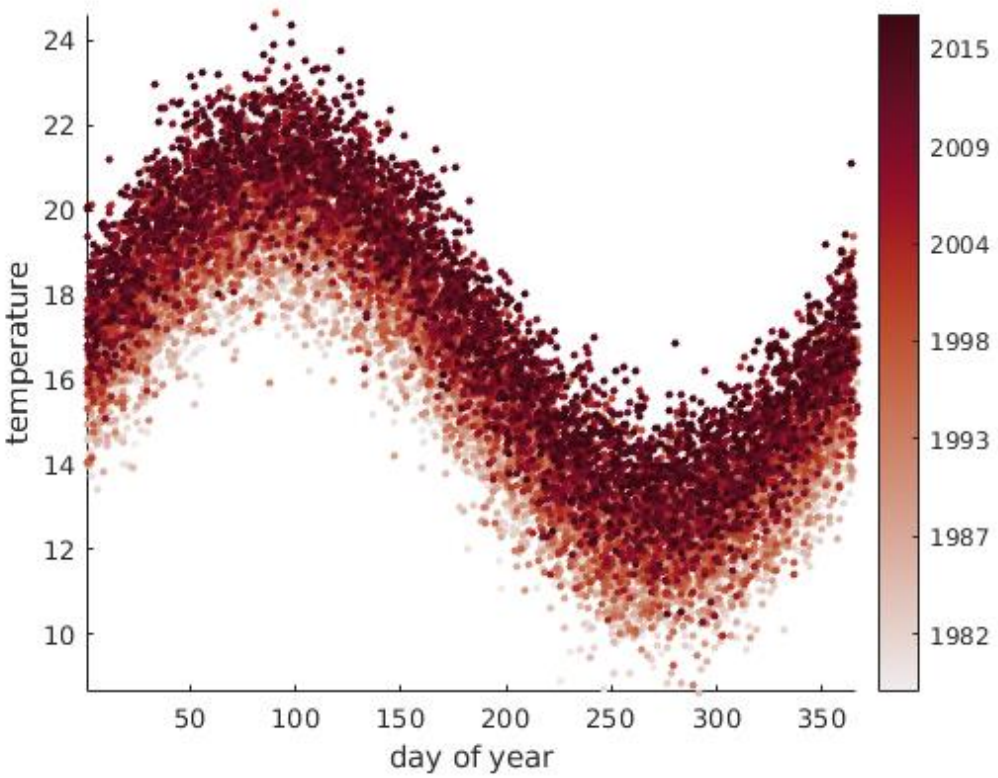
考虑这个时间序列。 我们称它为温度：

```
t = datenum('jan 1, 1979'):datenum('dec 31, 2016');

% 正弦季节信号+变暖趋势+噪声+平均值:
T = 4*sin(doy(t,'decimalyear')*2*pi) + (t-min(t))/5e3 + randn(size(t)) + 15;

plot(t,T)
axis tight
box off
datetick('x','keeplimits')
```



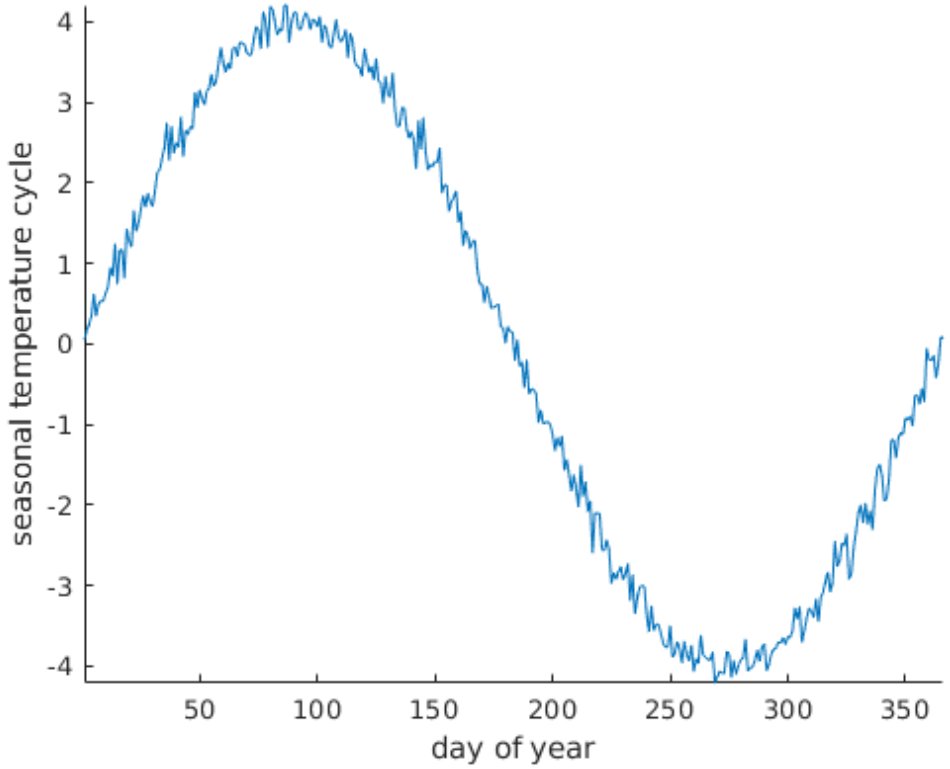
您可以看到温度是一个季节性周期，但是随着时间的推移会逐渐变暖，并且那里还存在一些随机噪音。为了更好地了解季节性周期的结构，我们可以使用 doy 函数将相同的时间序列作为一年中某天的函数。我们还将使用 cbdate 函数设置颜色栏日期的格式。

```
figure

scatter(doy(t),T,10,t,'filled')
axis tight

colorbar
cbdate('yyyy')
xlabel 'day of year'
ylabel 'temperature'
cmocean amp
```



同样，您可以看到温度随着时间的推移而上升，但是通过绘制 doy 的函数，我们可以更好地看到季节性周期的结构。现在，如果我们想将长期变暖的影响与季节周期分开，该怎么办？

```
Ts = season(T,t);



figure

plot(1:366,Ts)

box off
axis tight
xlabel 'day of year'
ylabel 'seasonal temperature cycle'
```

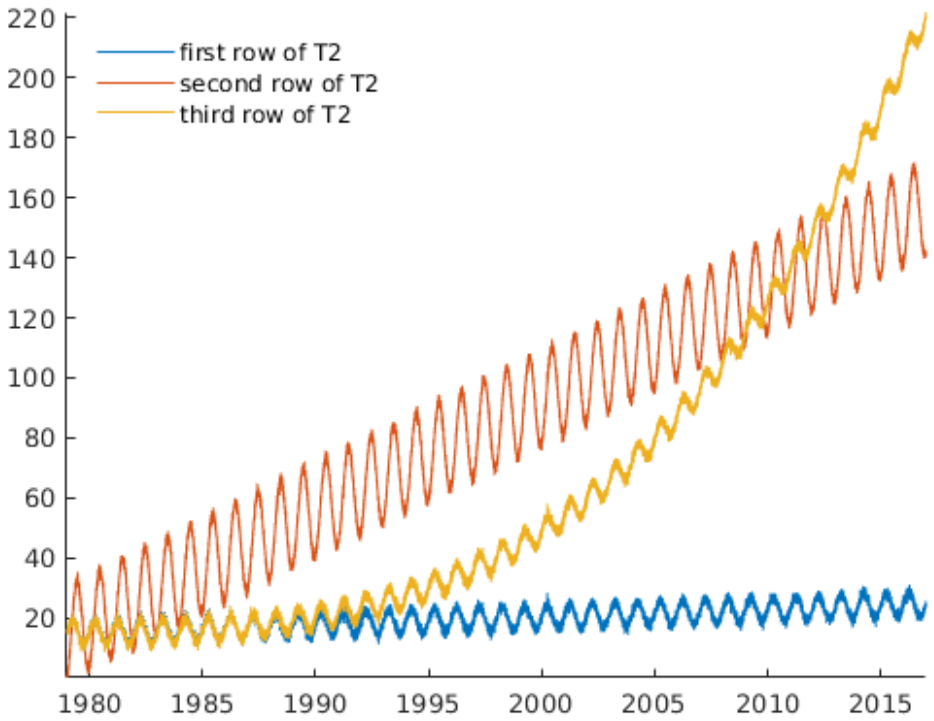


就像我们定义的那样，在上方我们可以看到振幅为 4 度的季节性周期。由于一年中的每一天代表 38 次嘈杂测量的平均值，因此仍然存在一些噪声。但是，这是我们最接近季节周期的近似值，并且非常接近。

示例 2：一次多时间序列

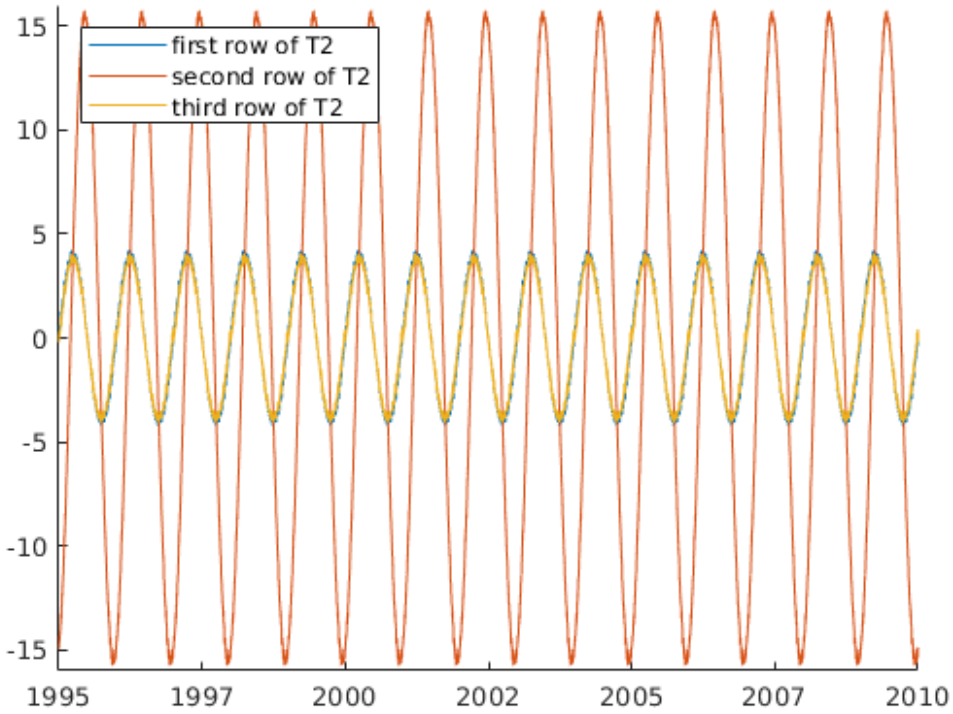
假设您有几行数据，每行包含一个不同的时间序列。数据可能看起来像这样（我们在上面的时间序列 T 上构建）

```
T2 = [T + (t-min(t))/2e3;

      T - 15*cos(doy(t,'decimalyear')*2*pi) + (t-min(t))/1e2;
      T + ((t-min(t))/2.5e3).^3.1];

figure
plot(t,T2)
axis tight
box off
datetick('x','keeplimits')
legend('first row of T2','second row of T2','third row of T2','location','northwest')
legend boxoff
```



每个时间序列的年度周期如何？使用 season，但请指定第二维度，因为 T2 的每一行都包含一个时间序列。如果 T2 的每一列都包含一个时间序列，则默认第一维度将是我们想要的。
我们还要使用'full'选项而不是默认的 1 到 366 天来获取整个时间序列。

```
T2_full = season(T2,t,'dimension',2,'full');

figure
plot(t,T2_full)
axis([datenum('jan 1, 1995') datenum('jan 1, 2010') -16 16])
box off
datetick('x','keeplimits')

legend('first row of T2','second row of T2','third row of T2','location','northwest')
```

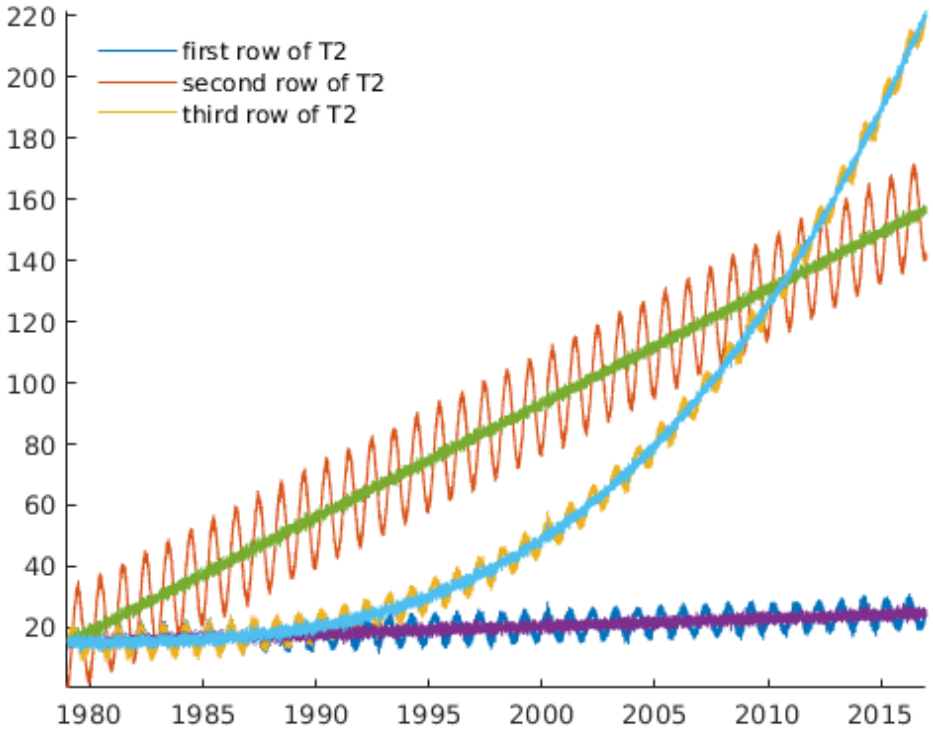


由此我们可以看到，season 提取了 T2 中每个时间序列的季节周期。即使使用长期线性趋势，也可以充分分析 T2 的第 3 行，该行具有长期趋势，可扩展到 3.1 的幂。但是，如果线性去趋势不足以适合您的应用程序，请使用'detrend','quadratic'选项将其提升一个档次。

移除季节性周期

为了方便起见，Climate Data Toolbox 具有一个 deseason 函数，但是在这里值得注意的是，deseason 函数仅通过从原始信号中减去季节分量来消除季节周期，如下所示：

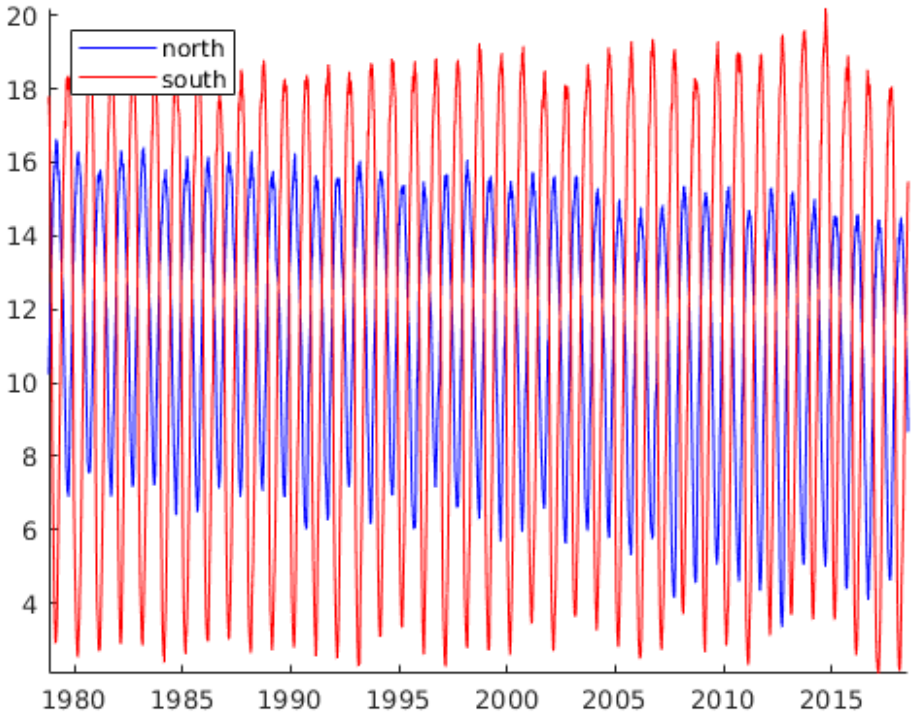
```
T_deseasoned = T2 - T2_full;



figure

plot(t,T2)

hold on

plot(t,T_deseasoned)
axis tight
box off
datetick('x','keeplimits')
legend('first row of T2','second row of T2','third row of T2','location','northwest')
legend boxoff
```



示例 3：真实海冰数据

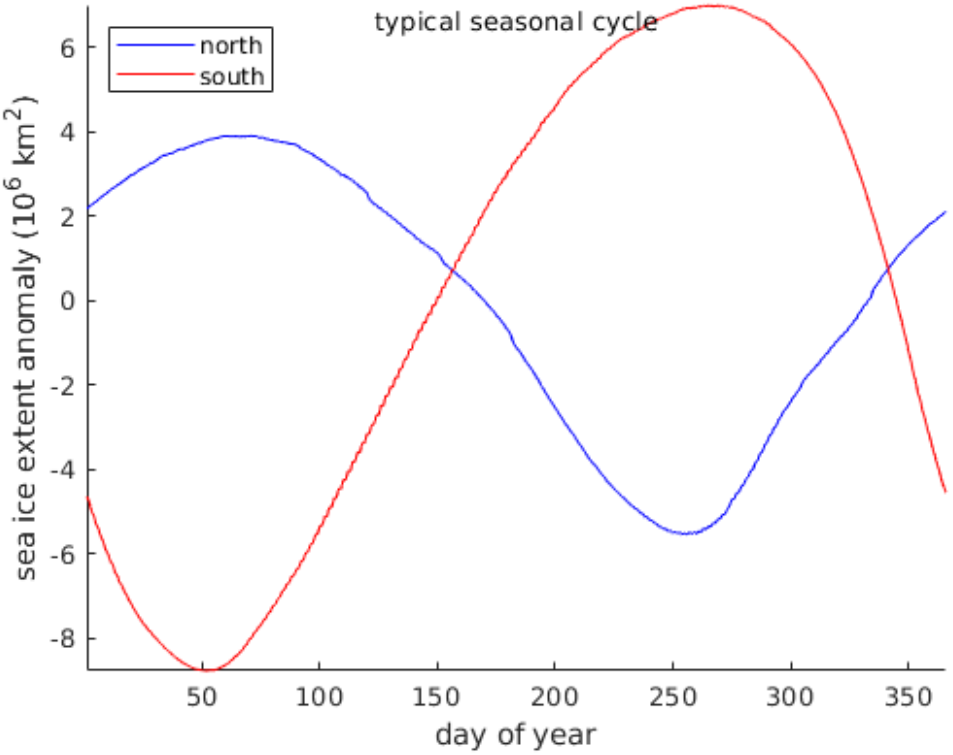
上面的示例使用了我们创建的一些人工数据。 但是此功能如何应用于真实数据？让我们比较一下北半球和南半球的海冰范围的季节性周期。这是整个时间序列的样子：

```
load seaice_extent

figure
plot(t,extent_N,'b')
hold on
plot(t,extent_S,'r')

axis tight
box off
legend('north','south','location','northwest')
```



所以，典型的海冰年长啥样呢？

```
figure

plot(season(extent_N,t),'b')
hold on
plot(season(extent_S,t),'r')

axis tight
box off
xlabel 'day of year'
ylabel('sea ice extent anomaly (10^6 km^2)')
ntitle(' typical seasonal cycle ')
legend('north','south','location','northwest')
```



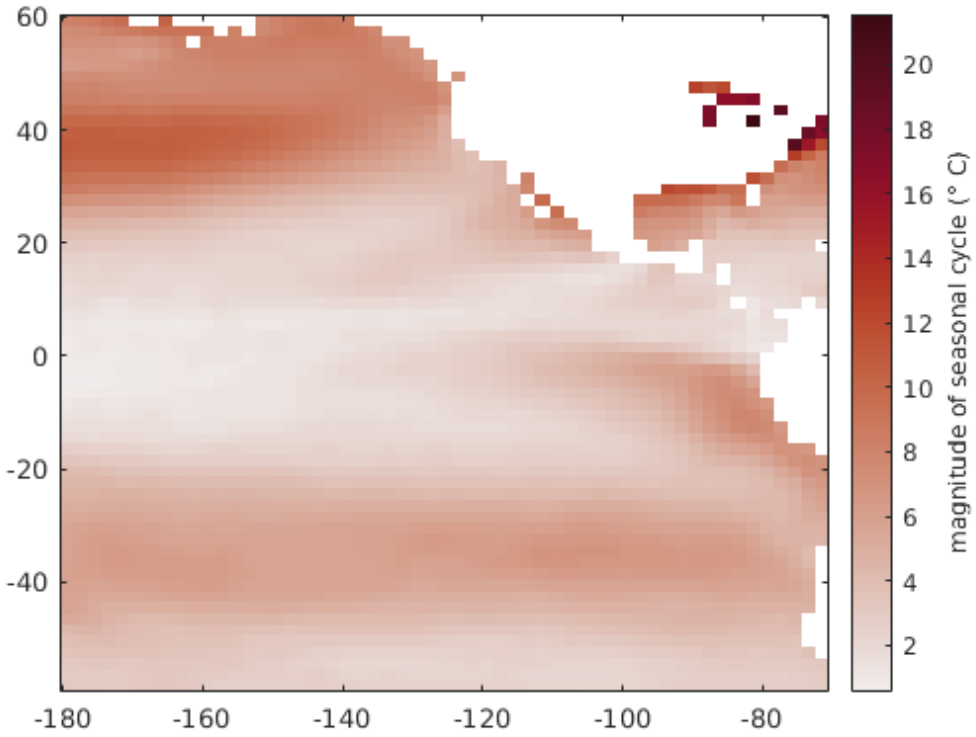
示例 4：格点再分析数据

让我们探索海面温度的典型季节性周期。这是尺寸为 60x55x802 的每月数据集：

```
load pacific_sst

sst_season = season(sst,t);
```

这样就得出了 sst_season，即 60x55x12，代表一年中每个月一个 60x55 的网格。利用典型最高和最低温度之间的差值，可以了解太平洋不同地区的季节强度：

```
T_range = max(sst_season,[],3)-min(sst_season,[],3);



figure

imagescn(lon,lat,T_range)

cb = colorbar;

ylabel(cb,'magnitude of seasonal cycle (\circ C)')
cmocean amp
```



这不足为奇——靠近两极和浅水区的季节性最强。

这个函数是怎么工作的

默认情况下，季节函数会从时间序列中删除线性趋势，然后通过对一年中每一天（或每月）的所有趋势数据进行平均来确定时间序列的气候态。

对于每日数据，闰年提出了挑战。此功能通过平均与一年中第 59 天相对应的所有数据来计算 2 月 28 日的平均温度。但是 3 月 1 日该怎么办？大多数年份，3 月 1 日是一年中的第 60 天，但每 4 年 3 月 1 日是一年中的第 61 天。

此函数将所有日期转换为一年中的某一天，并根据一年中的某天计算季节周期。由于 366 天的年数要少得多，因此应用对 366 天起所有数据进行平均的相同做法会使 366 天对异常值敏感。因此，对于第 366 天，季节函数对所有年份的第 365 天，第 366 天和第 1 天的所有数据取平均值。

季节与气候态

CDT 具有称为 season 的函数和称为 climatology 的另一函数。唯一的区别是 climatology 输出包含变量的平均值，而 season 输出始终为 0 平均值。因此，deseason 函数在保留总体均值和趋势的同时，消除了变异性的季节性成分。

一般而言，CDT 假定以亚年度分辨率采样的变量的多年记录可以通过以下方式描述

```
y = y_0 + y_tr + y_season + y_var + y_noise
```

这里

- y_0 是长期平均值，
- y_trend 是长期 trend，
- y_season 是典型的季节异常，在 season 函数去趋势并去除均值后获得
- y_var 表示年际变化，并且
- y_noise 就是其他一切

在这个模型中，

y_climatology = y_0 + y_season

定义季节性的其他方法

有关定义季节性的其他方法，参考 sinefit, sineval, 和 sinefit_bootstrap.

作者简介

这个函数是来自德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 于 2017 年 7 月写的。

deseason 文档

deseason 从时间序列中删除了变异性的季节性（也称为年度）成分。
另请参见: season, climatology, sinefit, sineval , 和 sinefit_bootstrap.

语法

```
Ads = deseason(A,t)
Ads = deseason(...,'daily')
Ads = deseason(...,'monthly')
Ads = deseason(...,'detrend',DetrendOption)
Ads = deseason(...,'dim',dimension)
```

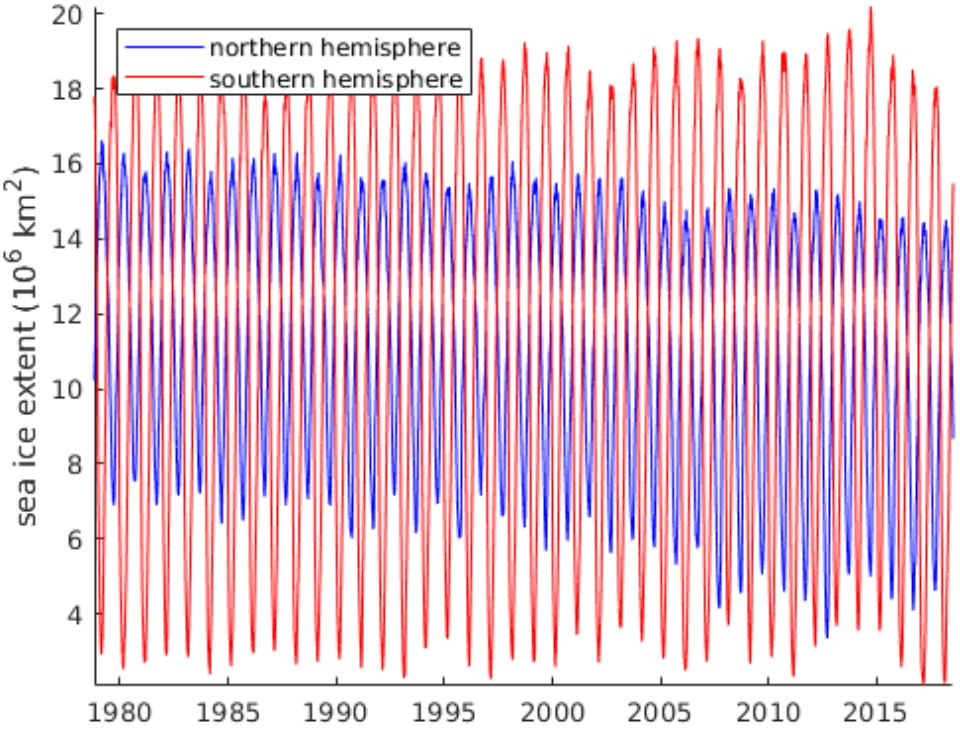
说明

Ads = deseason(A,t) 从对应于 datenum 格式中时间 t 的时间序列 A 中删除典型的季节性（也称为年度）周期。如果 t 为每天，则输出 ts 为 1 到 366，并且 As 将包含一年中 366 天中每一天的平均值。如果输入为每月，则 ts 为 1:12，而 As 将包含一年中 12 个月中每个月的平均值。
Ads = deseason(...,'daily') 直接指定输入为每日分辨率。Deseason 函数通常会自动解决此问题，但是如果您的数据中缺少大量空白，您可能希望通过指定每天来确保正确的结果。
Ads = deseason(...,'monthly') 同上，但强制采用每月的解决方案。
Ads = deseason(...,'detrend',DetrendOption) 指定相对于确定季节性异常的基准。选项为“线性”（'linear'），“二次”（'quadratic'）或“无”（'none'）。默认情况下，在删除线性最小二乘趋势后会计算异常，但是如果，例如，加热是强烈非线性的，则您可能更喜欢使用'quadratic'选项。注意：deseason 函数不会返回被去趋势的数据。相反，仅进行趋势消除以确定季节性周期。默认为'linear'。
Ads = deseason(...,'dim',dimension) 指定用于评估季节的维度。默认情况下，如果 A 为一维，则沿非单维度返回季节周期；否则，返回 0。如果 A 是二维，则沿第一维度执行（时间沿着行行进）；如果 A 是三维，则沿第三维度执行。

示例：海冰范围

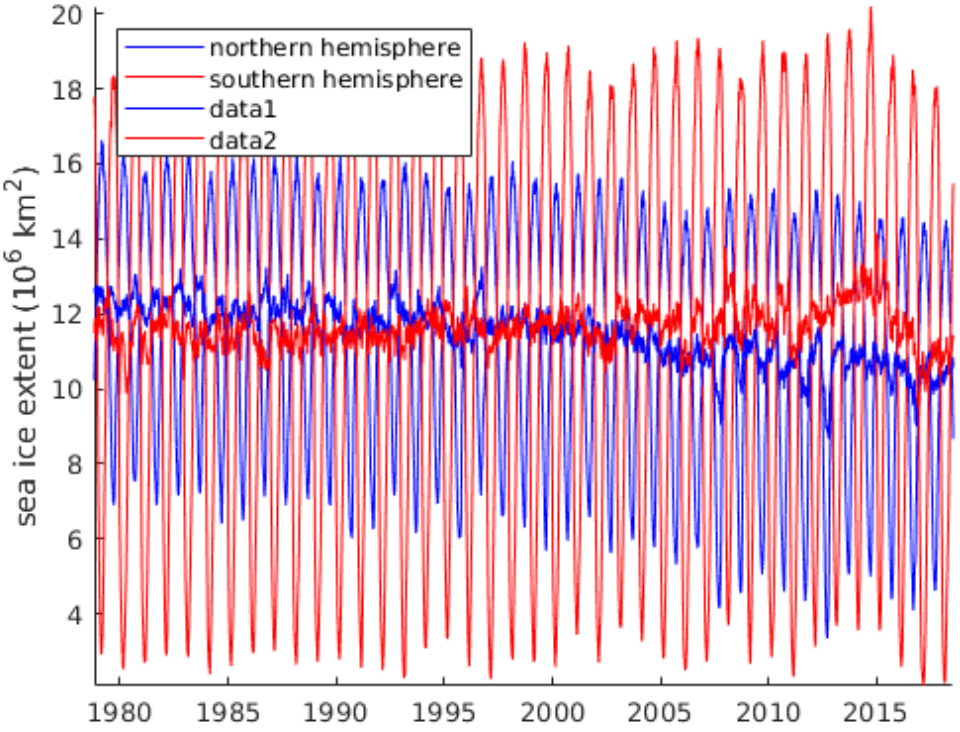
考虑以下样本海冰范围数据：

```
load seaice_extent

figure
plot(t,extent_N,'b')
hold on
plot(t,extent_S,'r')

axis tight
box off
legend('northern hemisphere','southern hemisphere','location','northwest')
ylabel ' sea ice extent (10^6 km^2) '
```



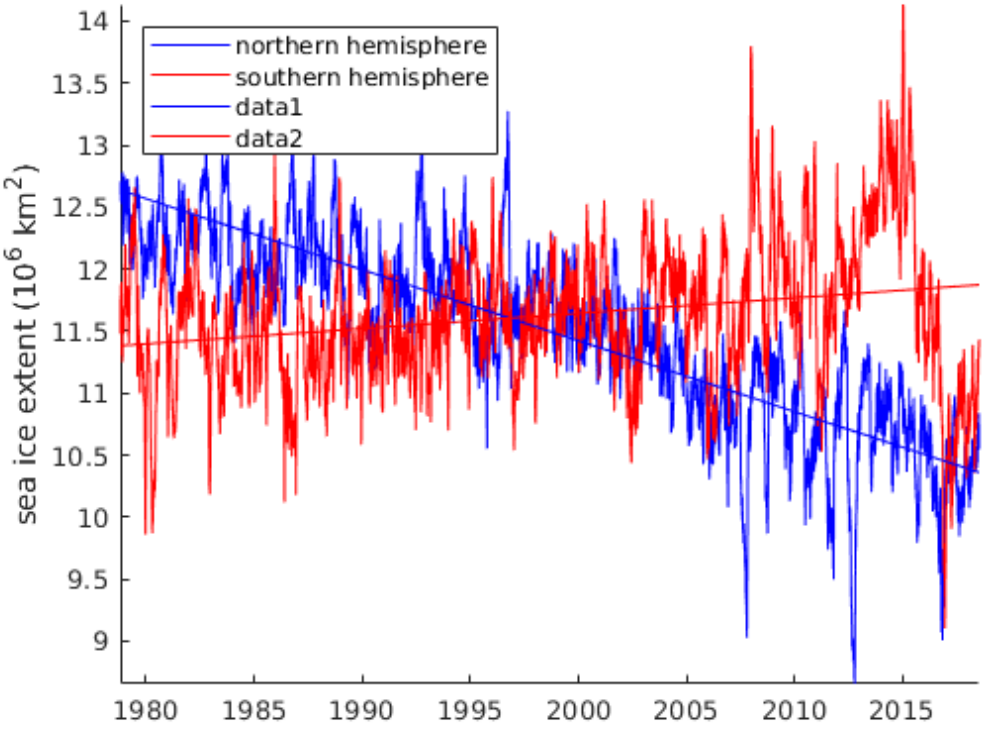
如您所见，海冰范围的变化主要受季节周期的支配。让我们对每个时间序列进行淡入淡出，以更好地了解每隔一年发生的事情：

```
north_ds = deseason(extent_N,t);

south_ds = deseason(extent_S,t);



plot(t,north_ds,'b')
plot(t,south_ds,'r')
```



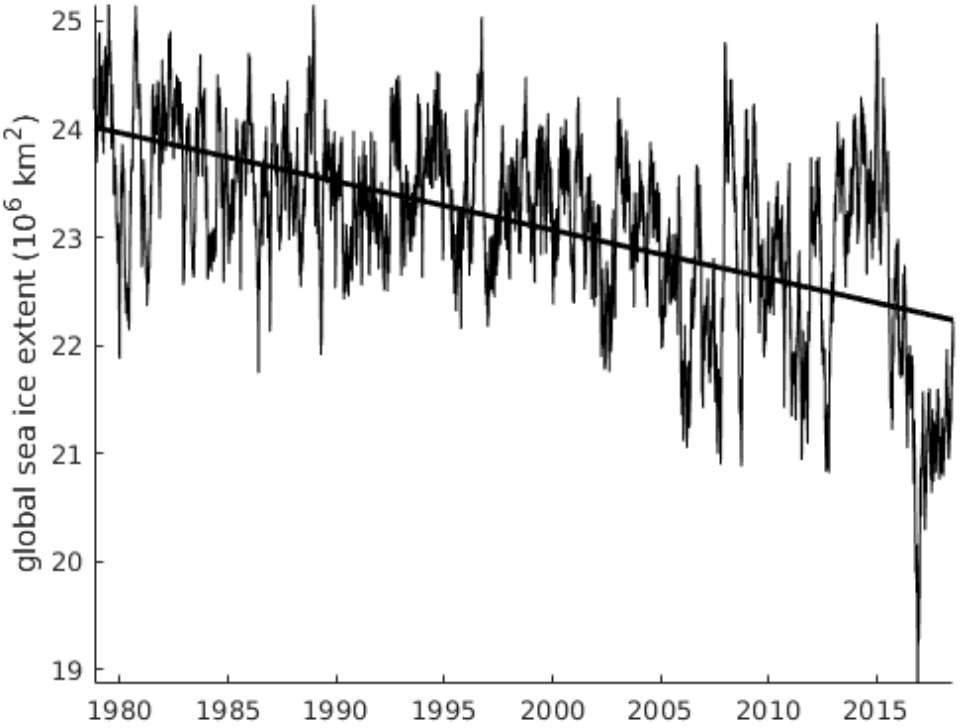
让我们仅使用去趋势数据创建一个新图，然后使用 polyplot 添加趋势线：

```
figure

plot(t,north_ds,'b')
hold on
plot(t,south_ds,'r')
axis tight
box off
legend('northern hemisphere','southern hemisphere','location','northwest')
ylabel ' sea ice extent (10^6 km^2) '

% 覆盖一条趋势线:
polyplot(t,north_ds,1,'b')
polyplot(t,south_ds,1,'r')
```



上图显示，在整个卫星时代，北半球的海冰面积下降，而南半球的海冰面积增加。但是，很难看出哪个半球获胜。了解全球情况的一种简单方法是简单地添加北半球和南半球的海冰范围：

```
figure

plot(t,north_ds+south_ds,'k')
hold on
polyplot(t,north_ds+south_ds,1,'k','linewidth',2)
axis tight
box off
ylabel ' global sea ice extent (10^6 km^2) '
```



上面的图清楚地表明，南极海冰的增加与北极海冰的损失没有同步。
我们可以使用趋势函数来量化全球海冰趋势。trend 函数计算相对于您给定单位的线性趋势，并且我们使用的是 datenum 单位为天，因此我们将乘以 365.25 * 10 以“每十年”单位获得海冰趋势，并且乘以 1e6 得到以 km ^ 2 为单位的单位：

```
trend(extent_S+extent_N,datenum(t))*365.25*10*1e6
```

```
ans =

   -4.5833e+05
```

通过卫星记录，这告诉我们，全球范围内，我们一直在损失每十年近 50 万平方公里的海冰。

这个函数是怎么工作的

deseason 函数就是简单的这么工作的：

```
Ads = A - season(A,t);
```

您将在此处 here 找到有关季节功能如何计算季节信号的说明。

季节与气候态

CDT 具有称为 season 的函数和称为 climatology 的另一函数。唯一的区别是 climatology 输出包含变量的平均值，而 season 输出始终为 0 平均值。因此，deseason 函数在保留总体均值和趋势的同时，消除了变异性的季节性成分。
一般而言，CDT 假定以亚年度分辨率采样的变量的多年记录可以通过以下方式描述

```
y = y_0 + y_tr + y_season + y_var + y_noise
```

这里

- y_0 是长期平均值,
- y_trend 是长期 trend,
- y_season 是典型的季节异常，在 season 函数去趋势并去除均值后获得
- y_var 表示年际变化，并且
- y_noise 就是其他一切

在这个模型中，

```
y_climatology = y_0 + y_season
```

定义季节性的其他方法

有关定义季节性的其他方法，参考 sinefit, sineval, 和 sinefit_bootstrap.

作者简介

这个函数是来自德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 于 2017 年 7 月写的。

climatology 文档

climatology 函数给出了一个变量的典型值，因为它全年都在变化。该函数的输出包括总平均值（而 season 函数的输出不包括）

另请参见: season, deseason, sinefit, sineval, 和 sinefit_bootstrap.

语法

```
[Ac,tc] = climatology(A,t)
[Ac,tc] = climatology(...,'daily')
[Ac,tc] = climatology(...,'monthly')
[Ac,tc] = climatology(...,'detrend',DetrendOption)
[Ac,tc] = climatology(...,'dim',dimension)
Ac = climatology(...,'full')
```

说明

[Ac,tc] = climatology(A,t) 给出了变量 A 全年变化时的典型值。时间 t 采用 datenum 或 datetime 格式。如果 t 为每日，则输出 tc 为 1 至 366，Ac 将包含一年中 366 天的平均值。如果输入为每月，tc 为 1:12，Ac 将包含一年中 12 个月的平均值。

[Ac,tc] = climatology(...,'daily') 直接指定输入为每日分辨率。Climatology 函数通常会自动解决这一问题，但如果您的数据中缺少大量空白，您可能希望通过指定'daily'来确保正确的结果。

[Ac,tc] = climatology(...,'monthly') 同上，但强制每月解决。

[Ac,tc] = climatology(...,'detrend',DetrendOption) 指定了与季节异常确定相关的基线。选项有'linear', 'quadratic'或'none'。默认情况下，在删除线性最小二乘趋势后计算异常，但是，例如，如果变暖是强非线性的，您可能更喜欢'quadratic'选项。默认值为'linear'。

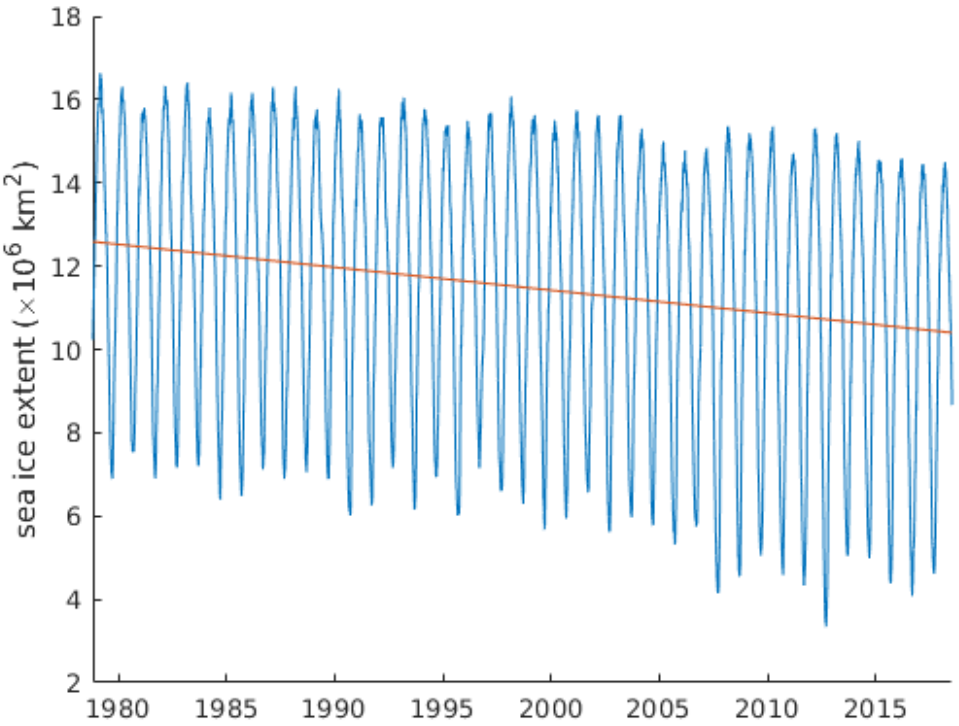
[Ac,tc] = climatology(...,'dim',dimension) 指定了评估季节的维度。默认情况下，如果 A 为一维，则沿非一维度返回季节循环；如果 A 是二维，则沿维度 1 执行 climatology（时间沿行向下移动）；如果 A 是三维，则沿维度 3 执行 climatology。

Ac = climatology(...,'full') 返回整个时间序列 A 的 Ac。这是一个方便的选项，用于单独查看长时间序列的组件。

示例 1: 每日海冰范围

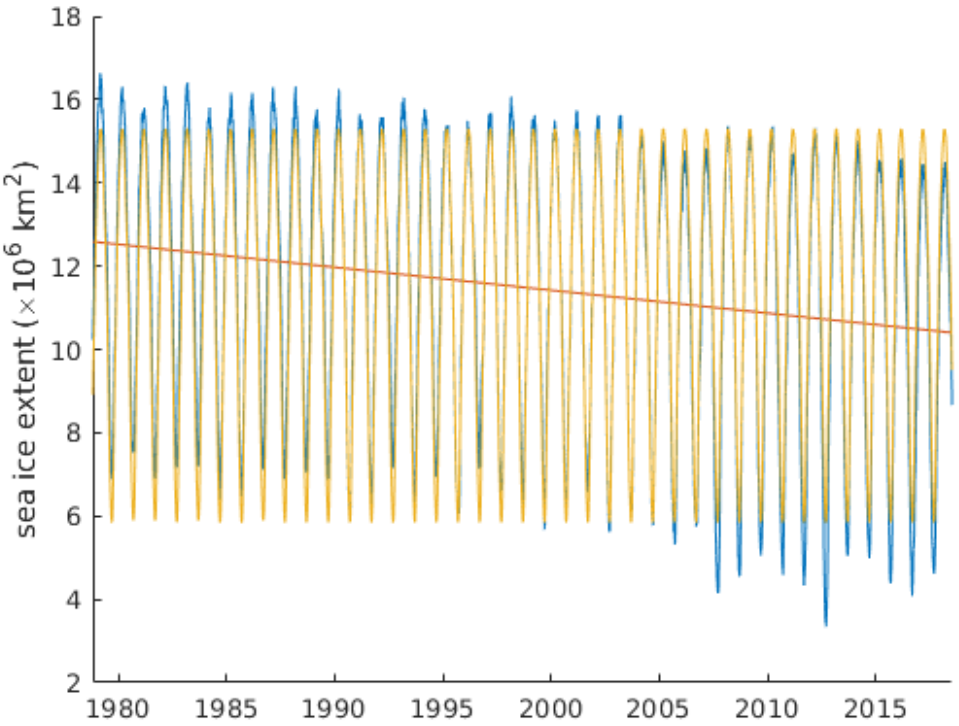
这里有一些海冰数据，*大部分*是每天的。1988 年之前，只有每隔一天的数据，但这很好——climatology 函数会计算出来。从加载和绘制数据开始。还可以使用 polyplot 添加最小二乘趋势线。

```
load seaice_extent

plot(t,extent_N)
box off
ylabel 'sea ice extent (\times10^6 km^2)'

hold on
polyplot(t,extent_N)
```



然后 climatology:

```
extent_N_clim = climatology(extent_N,t,'full');

plot(t,extent_N_clim)
```



在上面的图中，您会注意到 climatology 与原始海冰范围时间序列具有相同的平均值，但它没有任何随时间变化的趋势。它唯一的变化是季节的变化。

从原始信号中去除气候学产生的结果与去除平均值和季节周期产生的结果相同——只剩下长期趋势、年际变化和噪声。

示例 2: 格点数据

对于本示例，加载 pacific_sst 示例数据集，其中包含 802 个月网格化海面温度的网格化数据。使用这些 66.8 年的数据来了解 SST 在一年中的变化趋势，如下所示：

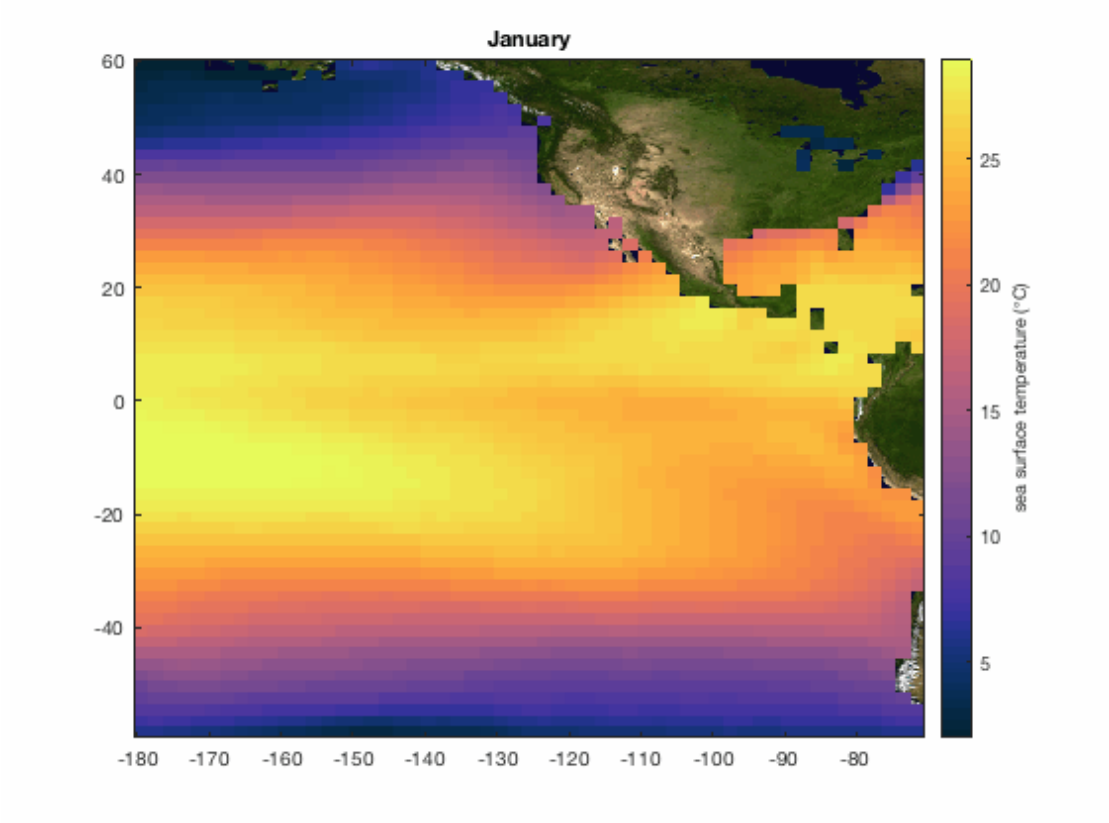
```
load pacific_sst


sst_c = climatology(sst,t);
```

现在，sst_c 是一个 60x55x12 的海面温度网格。第三个维度表示一年中每个月的海面温度。设置它们的动画并制作如下所示的 gif ，首先绘制第一帧：

```
figure
h = imagescn(lon,lat,sst_c(:,:,1));
cb = colorbar;
ylabel(cb,'sea surface temperature (\circC)')
cmocean thermal % 设置颜色图
title(datestr(datenum(0,1,1),'mmmm')) % 将 "1" 转化为 "January"
caxis([2 29])
% 添加傻傻的地球图像:
hold on
he = earthimage;
```

```
uistack(he,'bottom') % 把地球图像置于 SST 数据之下
% 写入第一帧:
gif('pacific_sst_climatology.gif','frame',gcf,'delaytime',1/12,'nodither')
% 写入其它月份的帧:
for k = 2:12
   h.CData = sst_c(:,:,k);                % 更新月度值
   title(datestr(datenum(0,k,1),'mmmm')) % 更新标题
   gif                                    % 在 gif 中添加此帧
end
```

…这就产生了一个漂亮的动画：



季节与气候态

CDT 具有称为 season 的函数和称为 climatology 的另一函数。唯一的区别是 climatology 输出包含变量的平均值，而 season 输出始终为 0 平均值。因此，deseason 函数在保留总体均值和趋势的同时，消除了变异性的季节性成分。

一般而言，CDT 假定以亚年度分辨率采样的变量的多年记录可以通过以下方式描述

```
y = y_0 + y_tr + y_season + y_var + y_noise
```

这里

- y_0 是长期平均值，
- y_trend 是长期 trend，
- y_season 是典型的季节异常，在 season 函数去趋势并去除均值后获得
- y_var 表示年际变化，并且
- y_noise 就是其他一切

在这个模型中，

```
y_climatology = y_0 + y_season
```

定义季节性的其他方法

有关定义季节性的其他方法，参考 sinefit，sineval，和 sinefit_bootstrap.

作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

sinefit 文档

sinefit 将正弦曲线的最小二乘估计与周期为 1 年的时间序列数据拟合。
对于有关的函数，参考 sineval，和 sinefit_bootstrap 文档。

语法

```
ft = sinefit(t,y)
ft = sinefit(...,'weight',weights)
ft = sinefit(...,'terms',TermOption)
[ft,rmse] = sinefit(...)
```

说明

ft = sinefit(t,y) 输入在时间 t 收集的数据 y，拟合周期为 1 年的正弦曲线。输入时间可以是 datenum，datetime 或 datestr 格式，不需要定期采样。输出 ft 包含下述计算中各项的系数。
ft = sinefit(...,'weight',w) 对每个观测值 y 加权。例如，如果形式误差 err 与 y 相关联，则可以让 w = 1./err.^2。默认 w = ones(size(y)).
ft = sinefit(...,'terms',TermOption) 指定在正弦拟合中计算哪些项目。TermOption 可以是 2、3、4 或 5:

- 2: ft = [A doy_max] 其中 A 是正弦波的振幅，doy_max 是一年中对应于正弦波最大值的中的那一天。缺省 TermOption 为 2。
- 3: ft = [A doy_max C] 也估计 C，即恒定偏移量。求解会增加处理时间，因此您可能更希望自己估计 C 作为输入 y 的平均值。但是，如果您不能假设 C = mean (y)，则可能更喜欢这种三项解决方案。
- 4: ft = [A doy_max C trend] 还会估计整个时间序列中的线性趋势，以每年 y 为单位。同样，同时求解四个项将比求解两个项的计算量大得多，因此您可能更愿意使用 polyfit 自己估算趋势，然后对趋势数据进行二项正弦拟合。
- 5: ft = [A doy_max C trend quadratic_term] 在解决方案中还包含一个二次项，但这目前仍是实验性的，因为将多项式拟合到参考没有趋势的年中日期可能会难以缩放。

[ft,rmse] = sinefit(...)返回去除正弦拟合后 y 残差的均方根误差。这是对正弦曲线拟合数据的程度的一种度量，但是要更深入地了解不确定性(包括时序中的不确定性)，请参见 sinefit_bootstrap.

示例 1a:正弦拟合一个好玩的数据

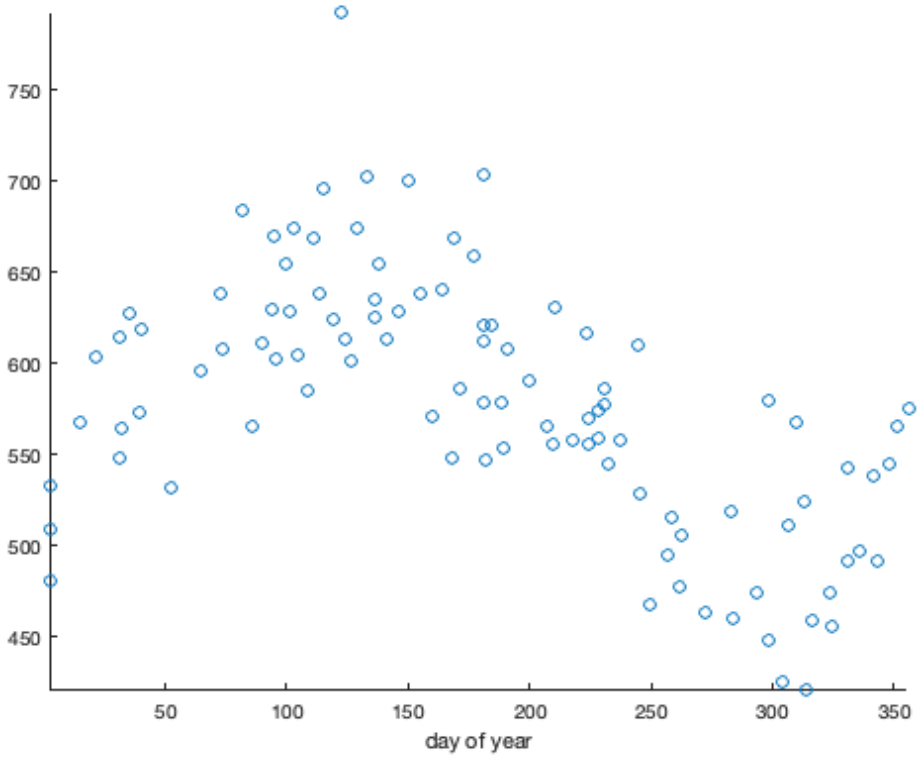
在此示例中，我们将用已知参数组成一些数据，然后使用 sinefit 将正弦曲线拟合到数据中。假设您在 2005 年 1 月 1 日到今天之间进行了 100 次测量。使用 sineval 可生成振幅为 75 单位的正弦曲线，在每年的 123 天（7 月 5 日）具有最大值，并且每年的线性趋势为-2.2 单位 请注意，在 sineval 中，我们还必须输入一个有点奇怪的值 5e3，它只是零年的 y 截距，但对 21 世纪的我们而言意义不大。
为了使其对 sinefit 更具挑战性，我们还将添加标准偏差为 36 个单位的高斯噪声。（相对于正弦波的 75 单位振幅，这是相当大的噪声。）

```
% 在 2005 年 1 月 1 日和今天直接随机选 100 天:
t = datenum('jan 1, 2005') + (now-datenum('jan 1, 2005'))*rand(100,1);

% 振幅为 75 的正弦波；在第 123 天最大（7 月 5 日）；趋势为-2.2 单位/年:
sine_parameters = [75 123 5e3 -2.2];
err = 36*randn(size(t)); % 随机误差
y = sineval([75 123 5e3 -2.2],t) + err;  % 正弦波+噪声
```

```
%画上 100 个数据点:
figure(1)
plot(t,y,'o')
axis tight %移除空白
box off    %移除外框
datetick('x','keeplimits') %格式化一下 x 轴的日期标签
```

信不信由你，这个数据集是正弦曲线。如果我们使用 doy 将所有数据绘制为一年中某一天的函数，则更容易看到正弦曲线的性质：

```
figure(2)

plot(doy(t),y,'o')

axis tight
box off
xlabel 'day of year'
```



在这个好玩的数据集中，我们知道应该有一个正弦曲线，其振幅为 75 个单位，在第 123 天为最大值，并且每年的长期趋势为-2.2 个单位。而且，由于我们有意添加了 36 个单位的高斯噪声，因此我们知道正弦曲线应使“测量值”与 36 个单位的均方根误差相匹配。

```
[ft,ft_error] = sinefit(t,y,'terms',4)
```

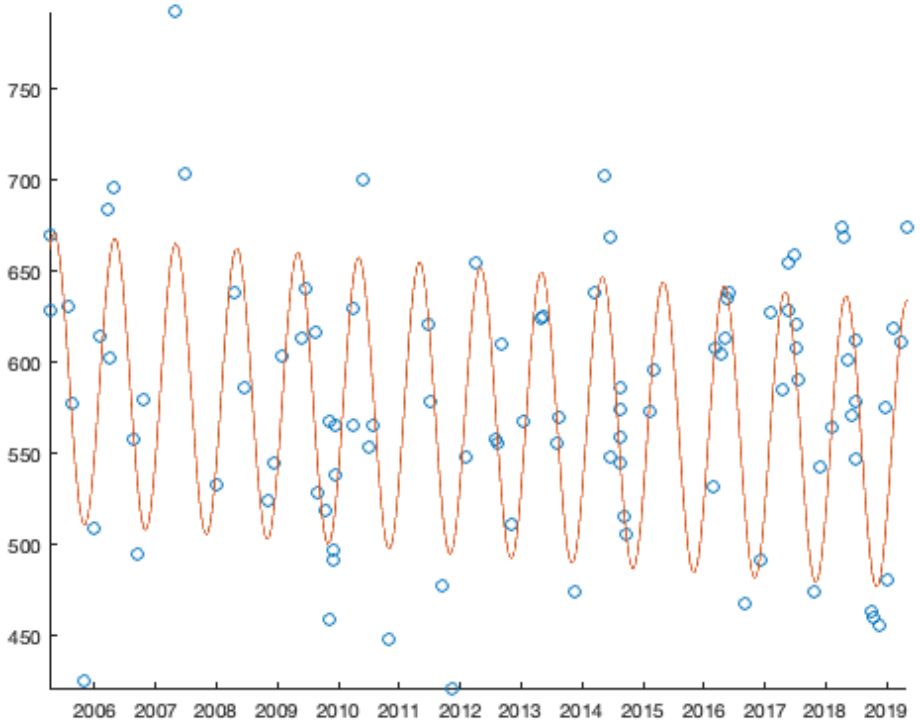
```
ft =

          79.24          122.69        5867.81          -2.63

ft_error =

          39.54
```

上面的数字告诉我们，sinefit 确定了我们的 100 个嘈杂数据点的特征是正弦曲线，其幅度为 ft(1)单位，我们看到它接近规定值 75，最大值接近一年的第 123 天，并且每年线性趋势约为-2.2 个单位。sinefit 函数的第二个可选输出告诉我们，正弦曲线将“测量值”匹配到 ft_error 给定的 1σ 不确定度，这与高斯噪声的规定值一致。

这是适合数据的正弦曲线：

```
%从第一个数据点到最后一个的日时间数组:
t_daily = min(t):max(t);

%日分辨率正弦拟合:
y_daily = sineval(ft,t_daily);
```

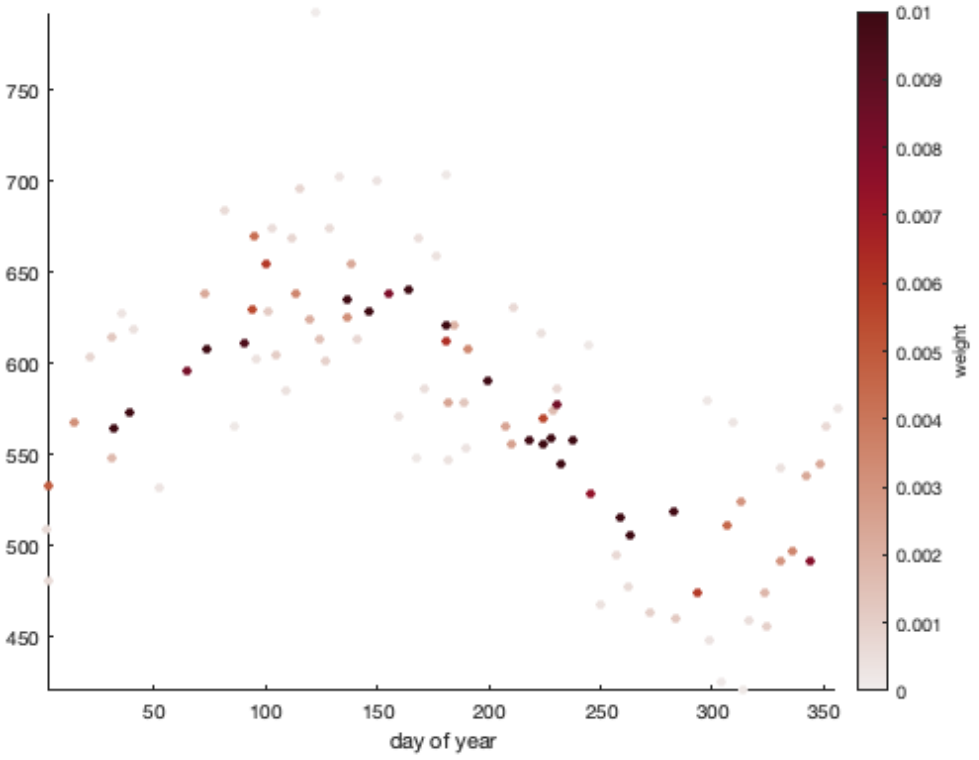
```
figure(1) %回到第一张图
hold on
plot(t_daily,y_daily)
```



示例 1b: 指定权重

假设您知道与每个观测值 y 相关的形式误差。一些测量比其他测量具有更高的不确定性。从上面我们用来指定噪声的错误向量中，我们将对每个观察值 y 进行加权，如下所示：

```
w = 1./err.^2;



figure

scatter(doy(t),y,25,w,'filled')
axis tight
xlabel 'day of year'
cb = colorbar;
ylabel(cb,'weight')
cmocean('amp')
caxis([0 0.01])
```



上图显示，我们将赋予这些值更多的权重，并降低与它们相关的误差。现在使用具有指定权重的 sinefit：

```
ft = sinefit(t,y,'weight',w,'terms',4)
```

```
ft =

          74.54          123.31         5627.04           -2.51
```

与未加权的解决方案相比，您会看到此版本产生的系数稍微更准确。

示例 1c:更多的不确定量

要对正弦曲线拟合的不确定性进行更深入的分析，请查看 sinefit_bootstrap，它可以为每个参数提供 1-sigma 的不确定性级别，如下所示（需要几秒钟）：

```
ftb = sinefit_bootstrap(t,y,'terms',4);

%每个参数 1 个 sigma 的不确定性:
std(ftb)
```

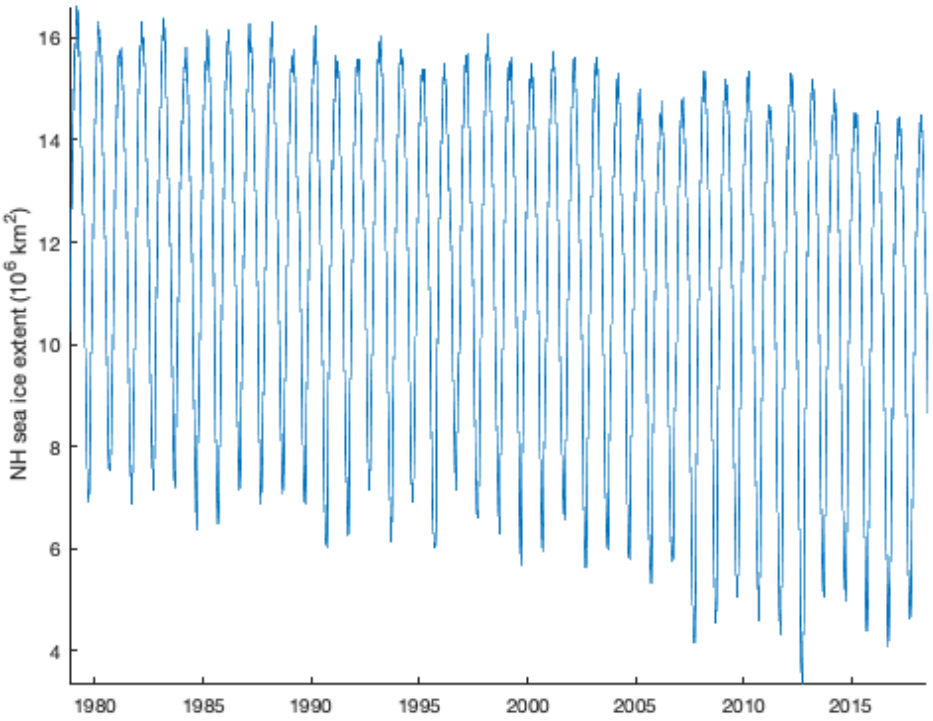
```
ans =

          6.09           3.84        2075.83           1.03
```

这就告诉我们，sinefit 的幅度应该精确到 1-sigma 水平，大约为 5（将 ft（1）与规定的幅度进行比较以向自己证明这一点）；时间应在约 4 天内准确；并且趋势应该在每年大约 1 个单位之内准确。

示例 2: 正弦拟合真实的海冰数据

让我们看一下来自 NSIDC 的一些真实海冰数据。 已经打包了.mat 格式的海冰范围的时间序列，并包含在此 File Exchange 上传中，因此您可以按照以下步骤进行操作：

```
% 载入示例数据:
load seaice_extent

% 绘制原始数据:
figure
plot(t,extent_N)
hold on
axis tight
box off
ylabel 'NH sea ice extent (10^6 km^2)'
```



很显然，数据在 1/yr 频率上具有一定的周期性。在半年的时间尺度上，北半球海冰面积变化多少，通常何时达到最大值，并且就长期趋势而言，海冰变化了多少？

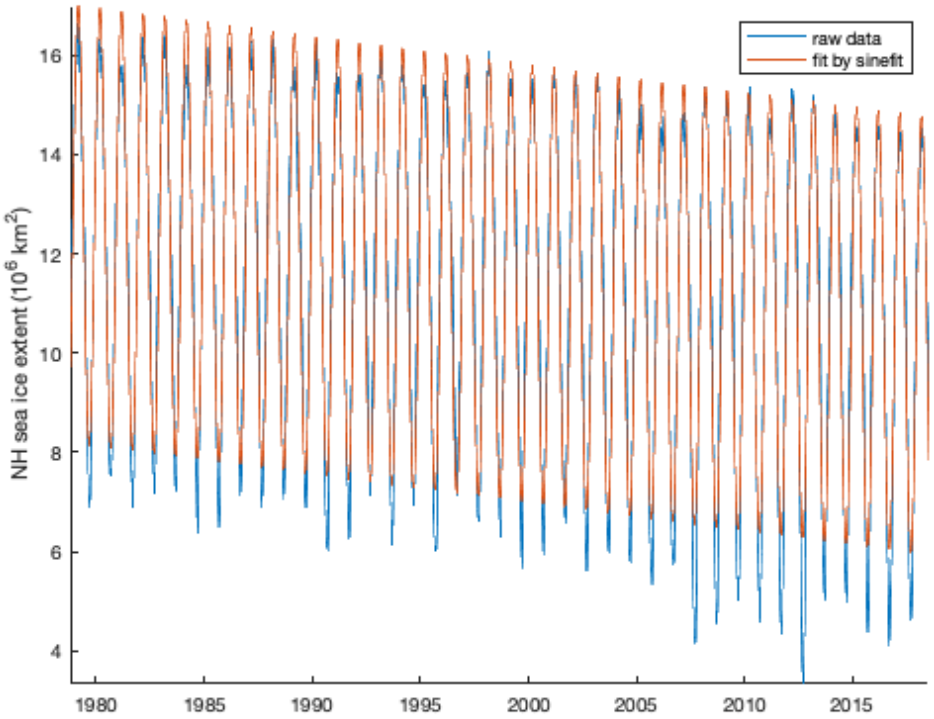
```
% 正弦拟合:
ft = sinefit(t,extent_N,'terms',4)
```

```
ft =
```

```
    4.41          66.84         125.30          -0.06
```

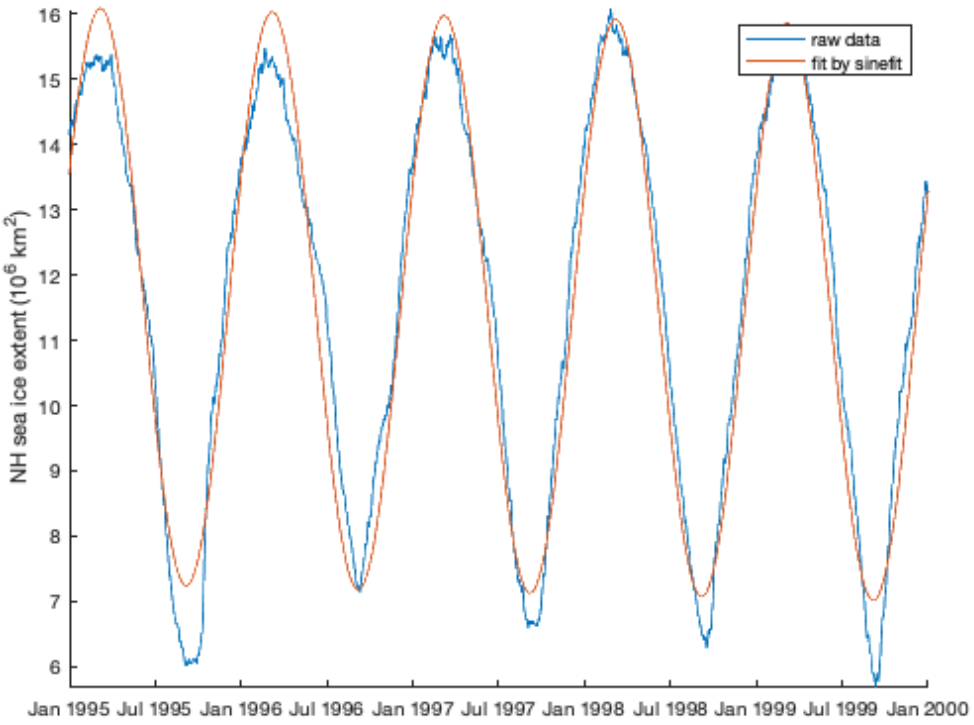
与示例 1 相似，sinefit 的输出告诉我们，北极海冰每年趋于变化约 441 万平方公里，在第 66 天（3 月 7 日）左右达到最大值（见下面的注释），自 1978 年以来每年北半球海冰减少了 60,000 平方公里。
这是在原始数据上方绘制的正弦曲线拟合：

```
hold on
plot(t,sineval(ft,t))
legend('raw data','fit by sinefit')
```



放大亿点点看看结构:

```
xlim([datetime(1995,1,1) datetime(2000,1,1)])
```



给用户的注释

一个简短与 sinefit 估计的所有参数有关的注释：这些参数描述了最佳拟合正弦曲线，但这并不一定意味着它们完全描述了基础数据本身的行为。例如，就气候平均而言，北半球的海冰范围实际上通常在第 71 天（3 月 12 日）左右达到最大值，而 sinefit 说，最合适的正弦曲线的最大值出现在第 66 天（3 月 7 日）。 这是因为海冰范围的真实行为比简单的正弦曲线更为复杂。在您的工作中，请务必考虑真实行为与真实行为的 1/yr 频率分量之间的差异。

作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。sinefit、sineval 和 sinefit_bootstrap 函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

sineval 文档

sineval 产生指定振幅和相位的正弦波，频率为 1/yr。
相关函数请参考 sinefit 和 sinefit_bootstrap 文档。

语法

```
y = sineval(ft,t)
```

说明

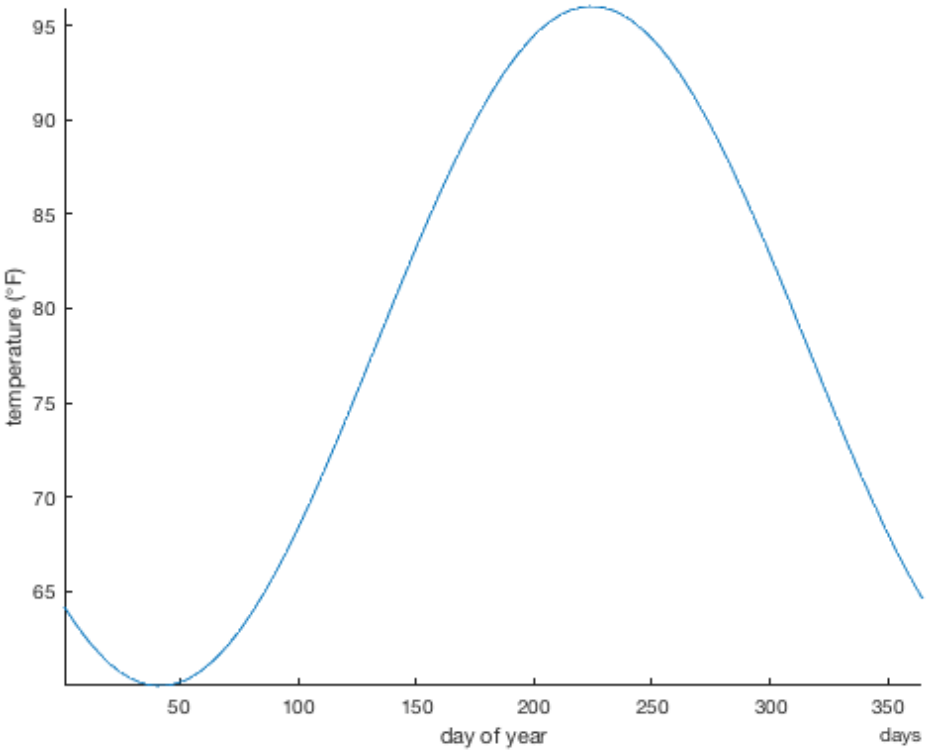
y = sineval(ft,t) 估算在时间 t 给定拟合参数 ft 的正弦曲线。时间 t 可以采用 datenum，datetime 或 datestr 格式，参数 ft 对应于 sinefit 函数的输出，可以包含 2 到 5 个元素，它们描述了以下内容：

- 2: ft = [A doy_max] 其中 A 是正弦波的振幅，doy_max 是一年中对应于正弦波最大值的中的那一天。默认 TermOption 为 2。
- 3: ft = [A doy_max C] 也估计 C，即恒定偏移量。求解会增加处理时间，因此您可能更希望自己估计 C 作为输入 y 的平均值。但是，如果您不能假设 C = mean (y)，则可能更喜欢这种三项解决方案。
- 4: ft = [A doy_max C trend] 还会估计整个时间序列中的线性趋势，以每年 y 为单位。同样，同时求解四个项将比求解两个项的计算量大得多，因此您可能更愿意使用 polyfit 自己估算趋势，然后对趋势数据进行二项正弦拟合。
- 5: ft = [A doy_max C trend quadratic_term] 在解决方案中还包含一个二次项，但这目前仍是实验性的，因为将多项式拟合到参考没有趋势的年中日期可能会难以缩放。

示例

我住在德克萨斯州的奥斯汀，一年中最热的一天发生在 8 月 12 日左右，大约在华氏 96 度，冬天的气温大约要低 36 度。换句话说，季节性正弦波的振幅可能为 18 度，其中最大值出现在儒略日 224，平均值为 78 度。使用这些原始数据，这是奥斯汀一年的每日高温气候：

```
%t 可以适应各种日期格式，啥都行:
t = days(1:365);

% 最高温的正弦曲线:
HighTemp = sineval([18 224 78],t);

plot(t,HighTemp)
axis tight
box off
xlabel 'day of year'
ylabel 'temperature ({\circ}F)'
```



作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。sinefit、sineval 和 sinefit_bootstrap 函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

矩阵运算

- cube2rect 整形三维矩阵以与 Matlab 标准函数一起使用。
- rect2cube 是 cube2rect 的补充。它将二维矩阵整形和置换为三维立方体。
- mask3 将掩膜应用于与二维掩膜相对应的三维矩阵所有层次。
- expand3 根据二维网格和一维向量的乘积创建三维矩阵。
- local 返回从三维矩阵中的感兴趣区域计算出的一维数组。例如，如果您有一个庞大的全球三维海面温度数据集，则使用此函数可以轻松获取感兴趣区域内平均 sst 的时间序列。
- reshapetimeseries 将时间序列数据整形为年*时间的网格。
- near1 查找最接近指定坐标的数组中点的线性索引。
- near2 查找最接近指定位置的网格中点的下标索引。
- cell2nancat 将元胞数组的元素连接到 NaN 分离的载体中。
- xyz2grid 将规则间隔的列化 x、y、z 数据转换为网格数据。
- C2xyz 将等值图矩阵（由 contour 函数返回）转换为 x、y 和相应的 z 坐标。
- xyzread 只导入.xyz 文件的 x、y、z 列。

cube2rect 文档

cube2rect 函数整形三维矩阵以与 Matlab 标准函数一起使用。

cube2rect 使您能够轻松高效地进行矢量化分析，而不是使用嵌套循环对矩阵的每一行和每一列进行操作。它的补充版是 rect2cube。

语法

```
A2 = cube2rect(A3)
A2 = cube2rect(A3,mask)
```

说明

A2 = cube2rect(A3) 对于一个尺寸对应于空间×空间×时间的三维矩阵 A3，cube2rect 将 A3 整形为尺寸对应于时间×空间的二维矩阵 A2。

A2 = cube2rect(A3,mask) 仅使用与二维 mask 中的真值对应的网格单元。对于在某些区域（也许是整个陆地或所有海洋网格单元）可以忽略的大型数据集，此选项可以减少其内存需求。

为什么会有这个函数?

好问题！简短的答案是它可以实现快速简便的矢量化，这意味着不再需要嵌套循环。可以在本教程中的 3D 数据分析中找到带有许多示例的更细致的解释。

示例 1a:一个简单的 2×2 网格

考虑这个 2x2 网格。在时间 1，值刚好超过 100。到时间 2，所有值都增加了 100，除了左下网格单元格增加了 120。趋势在时间 3 继续，所有值都在阶上的 300。以下是数据：

```
% 第一个时间切片:
A(:,:,1) = [101 103;
            102 104];

% 第二个时间切片:
A(:,:,2) = [201 203;
            222 204];

% 第三个时间切片:
A(:,:,3) = [301 303;
            342 304];
```

许多标准的 Matlab 函数在二维矩阵的列上向下操作，并且如果我们对 A 进行整形以使一个维对应于时间而另一维对应于空间，则可以轻松使用这些函数。以下是使用 cube2rect 整形 A 的方法：

```
cube2rect(A)
```

```
ans =

   101   102   103   104
```

```
   201    222    203    204

   301    342    303    304
```

现在，每一列对应于 A 中的一个网格单元，而每一行对应于时间。就像这样：

```
    s p a c e

   ---------->

t | 101    102    103    104

i | 201    222    203    204

m | 301    342    303    304

e v
```

示例 1b:数据掩膜区域

如果要对大型数据集执行计算量大的操作，有时可以帮助掩膜所有不需要分析的网格单元。例如，如果您有一个高分辨率的全局数据集，该数据集随时间进行了大量测量，则您可能希望使用 island 来掩盖陆地或海洋区域，而仅对感兴趣的区域进行地形分析。使用上面的简单 2×2 数据集，如果要在整形 A 之前屏蔽掉右上方的网格单元，则如下所示：

```
mask = [true false;

        true true]



cube2rect(A,mask)
```

```
mask =

  2×2 logical array

   1   0

   1   1

ans =

   101    102    104

   201    222    204
```

```
    301    342    304
```

示例 2:海表温度:

伪数据可能有点抽象，因此让我们看一下如何在实际气候数据中使用 cube2rect 。

```
load pacific_sst
```

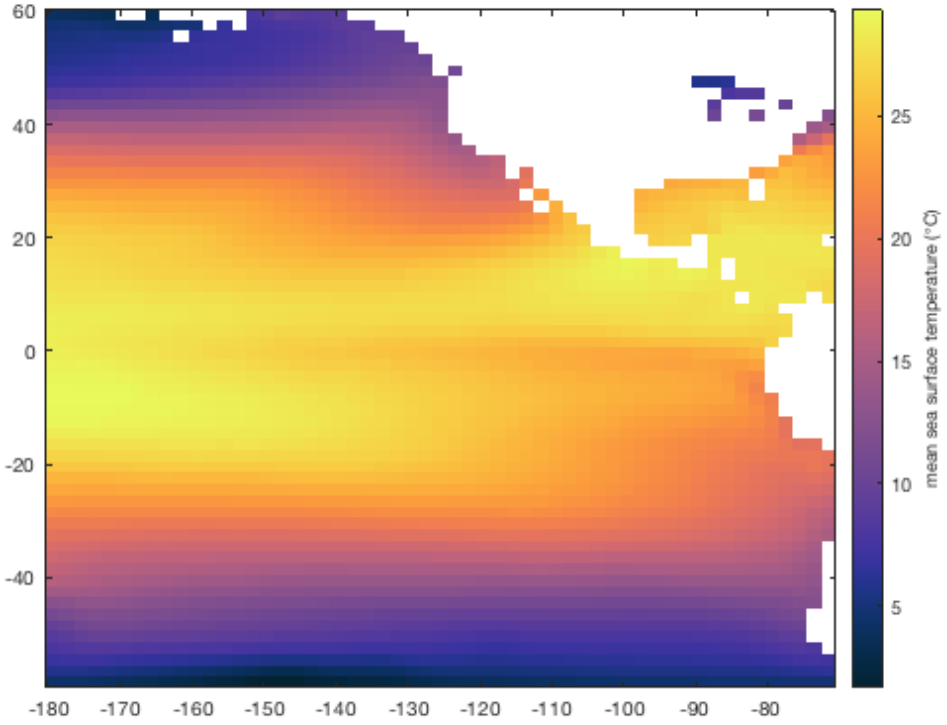
在 pacific_sst 样本数据集中，我们有一个 sst 的变量

```
size(sst)
```

```
ans =

    60    55   802
```

对应于 60 个纬度网格单元、55 个经度网格单元和 802 个月。为了计算每个网格单元的平均海面温度，我们可以在调用均值时指定 sst 的第三维，如下所示：

```
meanSST = mean(sst,3);



imagescn(lon,lat,meanSST)

cmocean thermal
cb = colorbar;
ylabel(cb,'mean sea surface temperature (\circC)')
```



是的，您当然可以轻松获得平均海面温度，但是 cube2rect 旨在允许使用更多的应用程序，而不仅仅是内置的 mean、std 等。因此，让我们使用 cube2rect 获得平均 SST 图，就像上面那个。首先，重塑三维 sst 矩阵：

```
sstr = cube2rect(sst);



size(sstr)
```

```
ans =

         802        3300
```

现在，经过整形的 sstr 矩阵包含对应于 802 个时间步长的 802 行和对应于 60x55 栅格单元中的每个的 3300 列。现在很容易计算每个网格单元的平均海面温度：

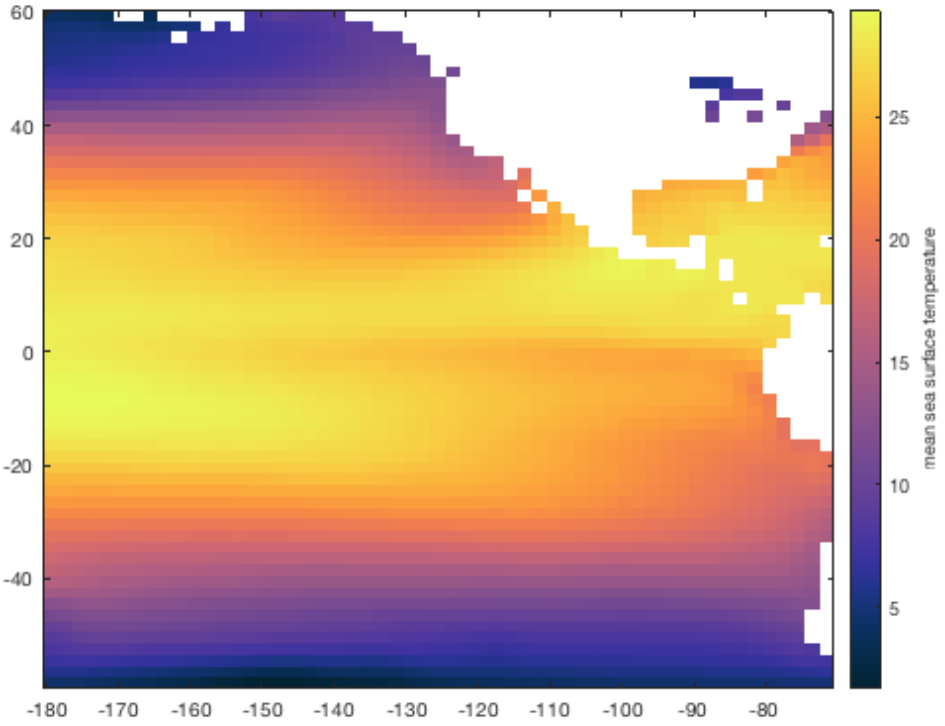
```
my_meanSST_2d = mean(sstr);



size(my_meanSST_2d)
```

```
ans =
```

1 3300

现在，my_meanSST 仅包含一行-这是 3300 个网格单元中每个单元的平均值。要使这些网格单元恢复其原始形状，请使用 rect2cube。对于 cube2rect 的阳，这是阴（译者注：这里的阴阳指的是阴阳互补）。

```
my_meanSST_3d = rect2cube(my_meanSST_2d,[60 55 802]);



imagescn(lon,lat,my_meanSST_3d)

cmocean thermal
cb = colorbar;
ylabel(cb,'mean sea surface temperature')
```



作者简介

这个函数和支持文档是来自德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

rect2cube 文档

rect2cube 函数 是 cube2rect 的补充。它将二维矩阵整形和置换为三维立方体。

语法

```
A3 = rect2cube(A2,gridsize)
A3 = rect2cube(A2,mask)
```

说明

A3 = rect2cube(A2,gridsize) 将二维矩阵 A2 整形为三维矩阵，该矩阵的前两个维度是空间（例如纬度×经度或经度×纬度），而第三个维度是时间或海洋深度或一些沿其进行操作的变量。A3 的最终尺寸由 gridsize 指定，它可以是描述 A3 的完整 3 元素数组，或者 gridsize 可以只是包含 A3 的前两个尺寸的 2 元素数组。
A3 = rect2cube(A2,mask) 将 A2 的元素整形为二维矩阵掩膜中的真实网格单元。

为什么会有这个函数?

好问题！简短的答案是它可以实现快速简便的矢量化，这意味着不再需要嵌套循环。可以在本教程中的 3D 数据分析中找到带有许多示例的更细致的解释。

示例 1: 简单的情况

该示例以一个随机的 4x3x2 矩阵开始，这意味着它是一个空间尺寸为 4x3 的网格，并且网格中的每个点都有两个测量值-两个“时间”切片-:

```
% 一个随机的 4x3x2 矩阵:
A3 = randi(50,[4 3 2])
```

```
A3(:,:,1) =

    33    45    15

    42    36    14

    16    13     1

    21    38    19

A3(:,:,2) =

    22    47    29

    16    44    25

    15    20    14

    13    24    50
```

使用 cube2rect 将 A3 转为二维矩阵:

```
A2 = cube2rect(A3)
```

```
A2 =

    33    42    16    21    45    36    13    38    15    14     1    19

    22    16    15    13    47    44    20    24    29    25    14    50
```

在上面，我们看到 A2 的第一行包含 A3 的第一个“时间片”中每个网格单元的值，而 A2 的第二行对应于第二个时间片。
要将 A3 恢复到其原始三维排列，请使用 rect2cube 并指定您希望 A3 为的尺寸：

```
A3 = rect2cube(A2,[4 3 2])
```

```
A3(:,:,1) =

    33    45    15

    42    36    14

    16    13     1

    21    38    19

A3(:,:,2) =

    22    47    29

    16    44    25

    15    20    14

    13    24    50
```

这使我们回到了起点。
注意：使用上述语法时，实际上不必指定 A3 的第三维。您可以只指定前两个维度，rect2cube 将找出第三个维度，如下所示：

```
A3 = rect2cube(A2,[4 3]);
```

示例 2:掩膜

如果您正在使用非常大的气候网格数据，则可能会发现在执行计算之前删除不感兴趣的区域在计算上更为有效。为此，请创建一个包含要保留的任何网格单元格真实值的掩膜。
在此示例中，我们还将使用具有 7 个 true 值的掩膜指示我们要分析的网格单元，并包含 5 个我们不希望在分析中包含的 false 值：

```
% 创建一个掩膜:
mask = true([4 3]);
mask([3 4 6 10 11]) = false
```

```
mask =

  4×3 logical array

   1   1   1

   1   0   0

   0   1   0

   0   1   1
```

使用 cube2rect 将 A3 转为二维矩阵:

```
A2 = cube2rect(A3,mask)
```

```
A2 =

    33    42    45    13    38    15    19

    22    16    47    20    24    29    50
```

在上面，我们看到 mask 中的每个 true 网格单元在 A2 中都有一列，而 A2 的行对应于 A3 的时间片。
现在，如果我们要将 A2 重新放入 3D 网格中，请使用相同的掩膜：

```
A3b = rect2cube(A2,mask)
```

```
A3b(:,:,1) =

    33    45    15

    42   NaN   NaN

   NaN    13   NaN

   NaN    38    19
```

```
A3b(:,:,2) =

    22     47     29
    16    NaN    NaN
   NaN     20    NaN
   NaN     24     50
```

以上，您会注意到 A3b 与我们开始使用的 A3 矩阵不完全匹配，因为我们丢弃了与 mask 中对应于 0 的网格单元中的所有信息，然后我们不得不用 NaN 值填充它们。这是进行分析的一种有损方式，但是如果这些网格单元格始终与分析无关，那么真的丢失了什么吗？

作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

mask3 文档

mask3 将掩膜应用于与二维掩膜相对应的三维矩阵所有层次。

语法

```
Am = mask3(A,mask)
Am = mask3(A,mask,repval)
```

说明

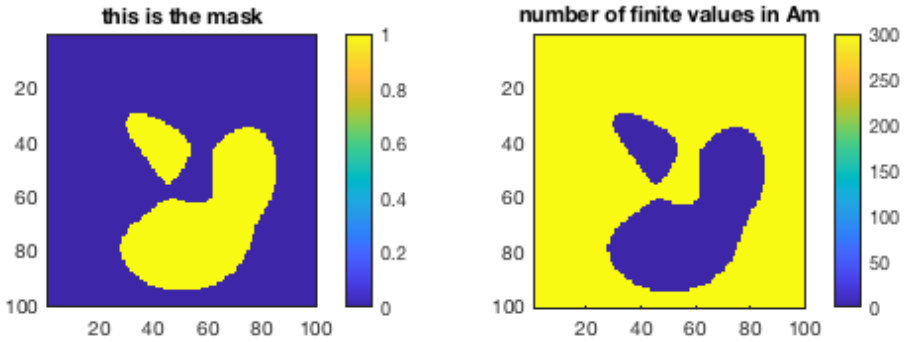
Am = mask3(A,mask) 将三维矩阵 A 的第 3 维上的所有元素设置为 NaN，只要在相应的二维逻辑掩膜中存在真实元素。

Am = mask3(A,mask,repval) ）用指定的值填充掩膜元素。默认 repval 为 NaN。.

示例 1:将掩膜下的值设为 NaN、

这是一些随机样本数据 A，其中包含 100x100 网格的 300 个时间步。制作一个掩膜，并将掩膜中的所有 true 值设置为 NaN：

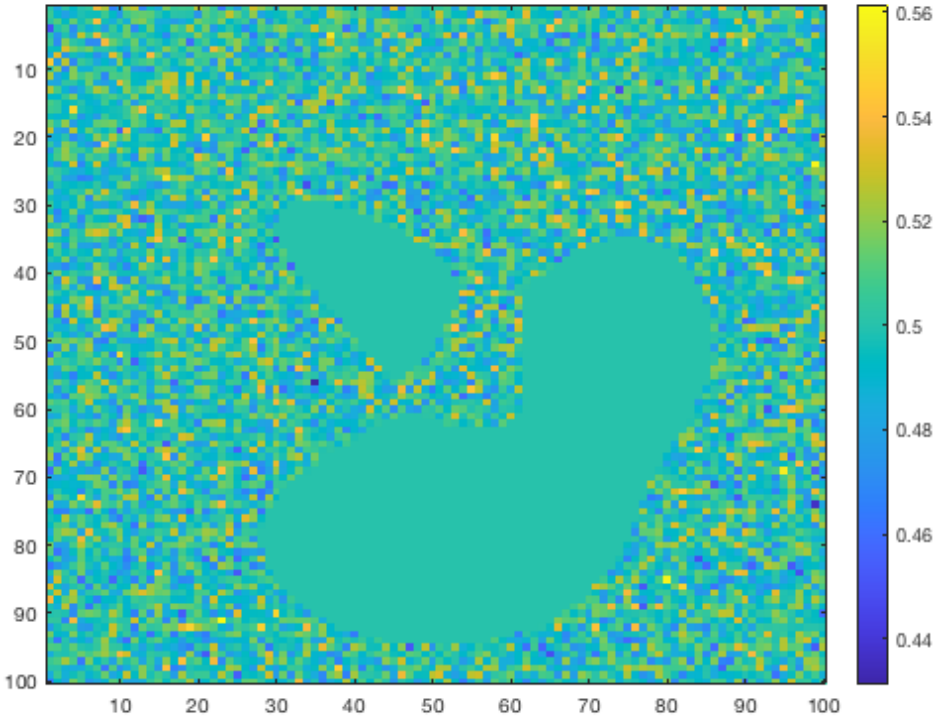
```
% 网格化时间序列:
A = rand(100,100,300);

% 当样品峰数据超过 1 时为 true:
mask = peaks(100)>1;

% 在掩膜为 true 的地方设为 NaN:
Am = mask3(A,mask);

% 画图:
figure
subplot(1,2,1)
imagesc(mask)
colorbar
title 'this is the mask'
axis image

subplot(1,2,2)
imagesc(sum(isfinite(Am),3))
colorbar
title 'number of finite values in Am'
axis image
```



示例 2:将掩膜下的值设为标量

假设您想将所有掩膜下的网格单元设置为特定值，例如 0.5。这是您的处理方式：

```
Am = mask3(A,mask,0.5);



figure

imagesc(mean(Am,3))

colorbar
```



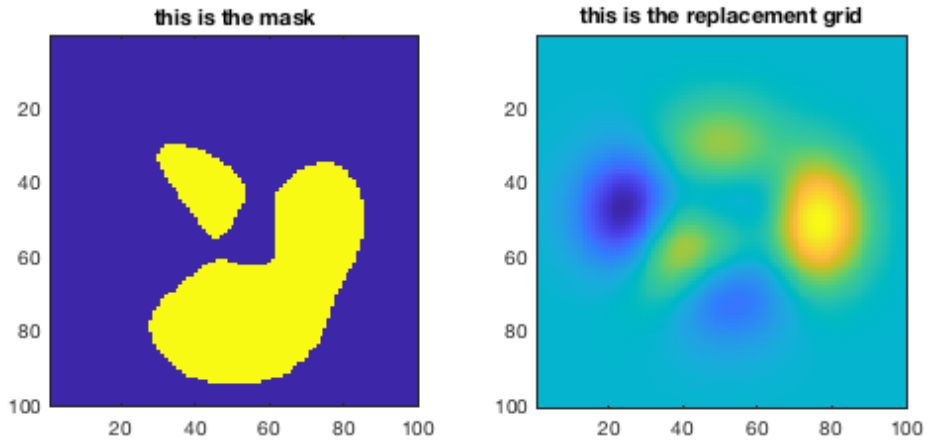
在上图中，我们看到掩码中的所有 true 值都正好为 0.5。其他所有内容都看起来像噪声，因为所有其他网格单元均显示 300 个随机值的平均值。

示例 3:用网格填充掩膜区域

有时，当您具有网格化的时间序列时，您不希望使用单个标量值而是使用相应的值网格来填充区域。例如，对于与上述相同的掩膜，您可能会有一个替换值 repgrid 来显示，如下所示：

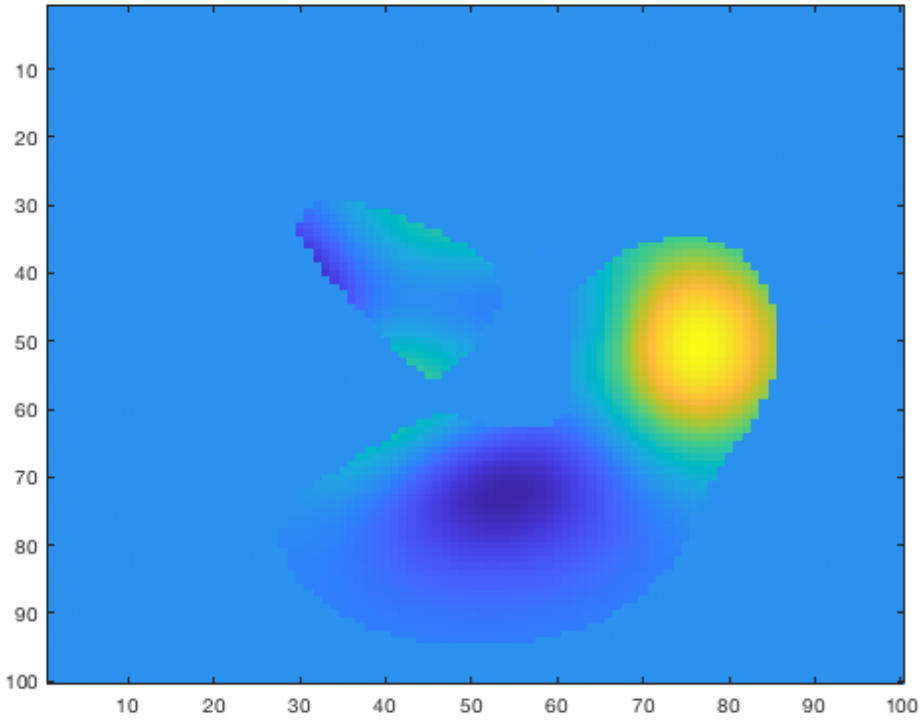
```
repgrid = rot90(peaks(100),1);



figure

subplot(1,2,1)

imagesc(mask)

axis image
title 'this is the mask'

subplot(1,2,2)
imagesc(repgrid)
axis image
```

```
title 'this is the replacement grid'
```



用 repgrid 中的相应值替换 A 中的所有掩膜网格单元：

```
Am = mask3(A,mask,repgrid);



figure

imagesc(mean(Am,3))
```



示例 4:海平面气压时间序列

本示例使用来自 ERA_Interim_2017.nc 数据集的数据。（有关使用.nc 文件的更多信息，请参见 NetCDF 教程。）首先加载数据：

```
filename = 'ERA_Interim_2017.nc';
lon = ncread(filename,'longitude');
lat = ncread(filename,'latitude');
SP = ncread(filename,'sp');
```

海平面气压时间序列 SP 具有以下维度

```
size(SP)
```

```
ans =

   480   241    12
```

这些对应于纬度、经度和时间。

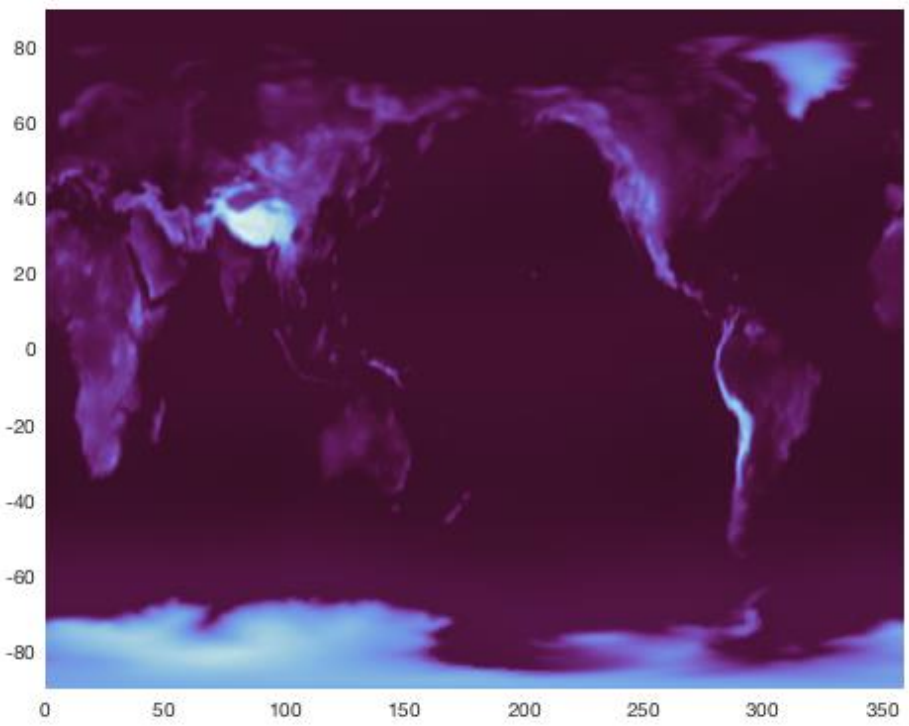
让我们屏蔽掉 SP 中与海洋相对应的所有点。为此，请从一维纬度、经度数组中创建纬度、经度网格，并使用 `island` 找出哪些网格单元是陆地，哪些网格单元是海洋。

```
[Lat,Lon] = meshgrid(lat,lon);
```

```
land = island(Lat,Lon);



% 海洋*不是*陆地:
ocean = ~land;
```

这是完整数据集的时间平均：

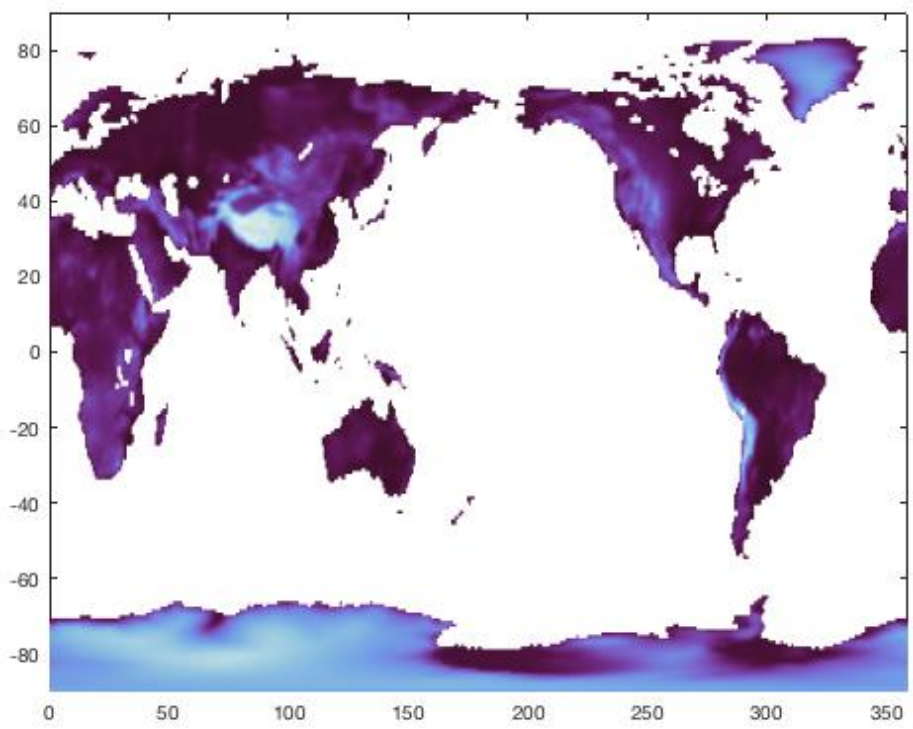
```
figure

pcolor(Lon,Lat,mean(SP,3))

shading interp
axis tight
cmocean dense % 选择颜色图
```



现在把所有海洋网格设为 NaN：

```
SPm = mask3(SP,ocean);



figure
```

```
pcolor(Lon,Lat,mean(SPm,3))

shading interp
axis tight
cmocean dense % 选择颜色图
```



作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

expand3 文档

expand3 根据二维网格和一维向量的乘积创建三维矩阵。

语法

```
Z3 = expand3(Z,y)
```

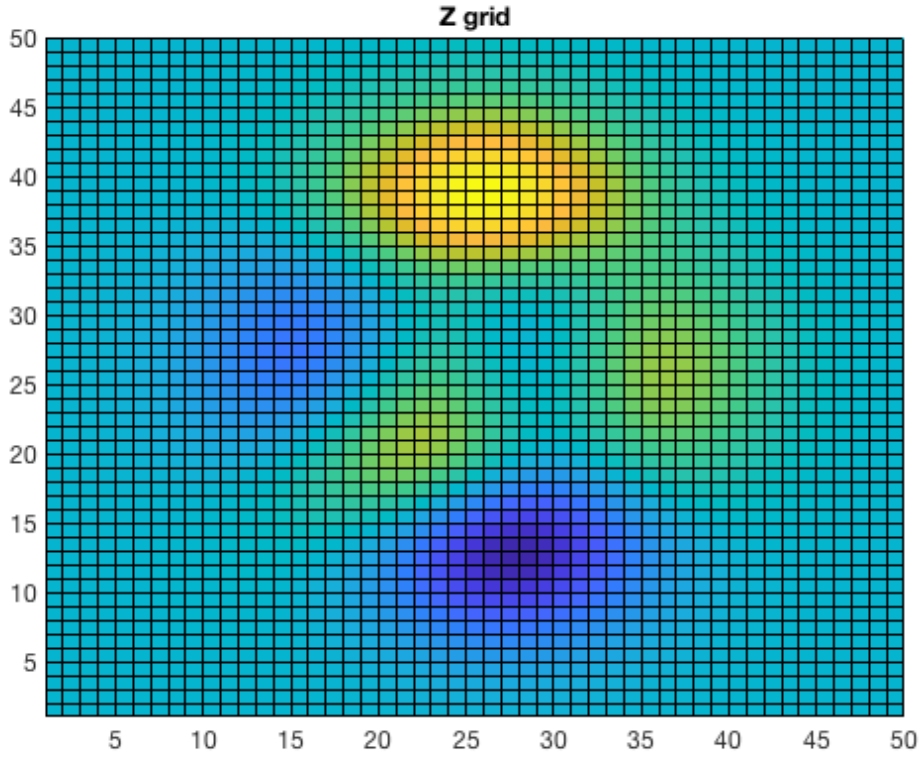
说明

Z3 = expand3(Z,y) 通过二维网格 Z 和一维数组 y 的元素相乘来创建三维矩阵 Z3。

示例

想象你有一个这样的 50×50 的网格：

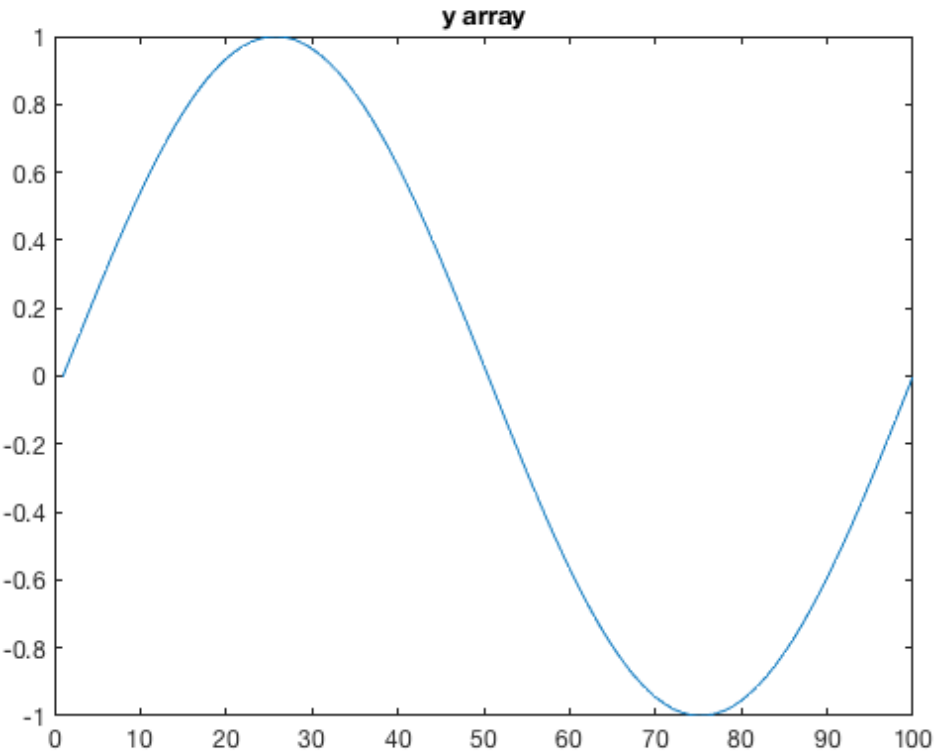
```
Z = peaks(50);



pcolor(Z);

title 'Z grid'
```



而且您知道整个网格随时间呈正弦变化，如下所示：

```
% 一个等距间隔 100 的正弦曲线:
y = sin(linspace(0,2*pi,100));
```

```
plot(y)
title 'y array'
```



这些是 Z 和 y 的维度：：

```
whos Z y
```

```
  Name        Size              Bytes  Class     Attributes

  Z          50x50             20000  double
  y           1x100              800  double
```

我们要将 50x50 的 Z 网格和 1x100 的 y 数组转换为 50x50x100 的矩阵，其中 Z3 的第一个 50x50 切片等于 Z*y(1)，第二个切片为 Z*y(2)，依此类推。这是这样的：

```
Z3 = expand3(Z,y);
```

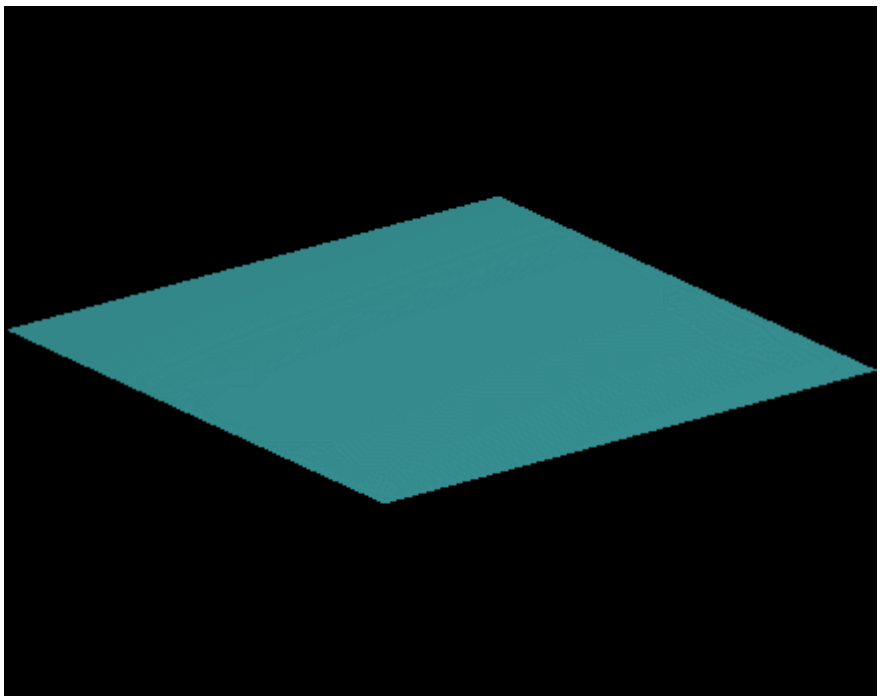
```
whos Z3
```

```
  Name        Size                 Bytes  Class     Attributes

  Z3          50x50x100            2000000  double
```

就酱紫。
然后，您可以为绘图设置动画，并将其另存为 gif，如下所示：

```
% 绘制第一帧:

h = surf(Z(:,:,1));

shading interp

axis([-3 3 -3 3 -9 9])

% 让他好看:

camlight

set(gca,'color','k')

set(gcf,'color','k')

caxis([min(Z(:)) max(Z(:))])

% 写入第一帧:

gif('myfile.gif')

% 循环每一帧:

for k = 2:100

   set(h,'Zdata',Z(:,:,k))

   gif % 保存这一帧

end
```

就是这样。 最终产品的外观如下：



作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

local 文档

local 函数返回从三维矩阵中的感兴趣区域计算出的一维数组。例如，如果您有一个庞大的全球三维海面温度数据集，则使用此函数可以轻松获取感兴趣区域内平均 sst 的时间序列。

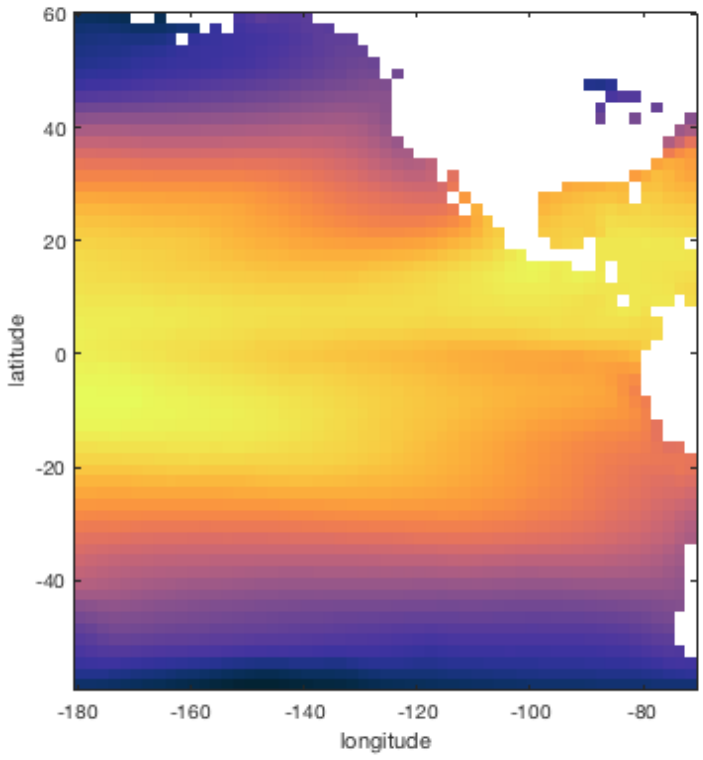
语法

```
y = local(A)
y = local(A,mask)
y = local(A,mask,'weight',weight)
y = local(A,mask,@function)
y = local(...,FunctionInputs,...)
y = local(...,'omitnan')
```

示例

加载此样本数据，并在绘制平均海面温度：

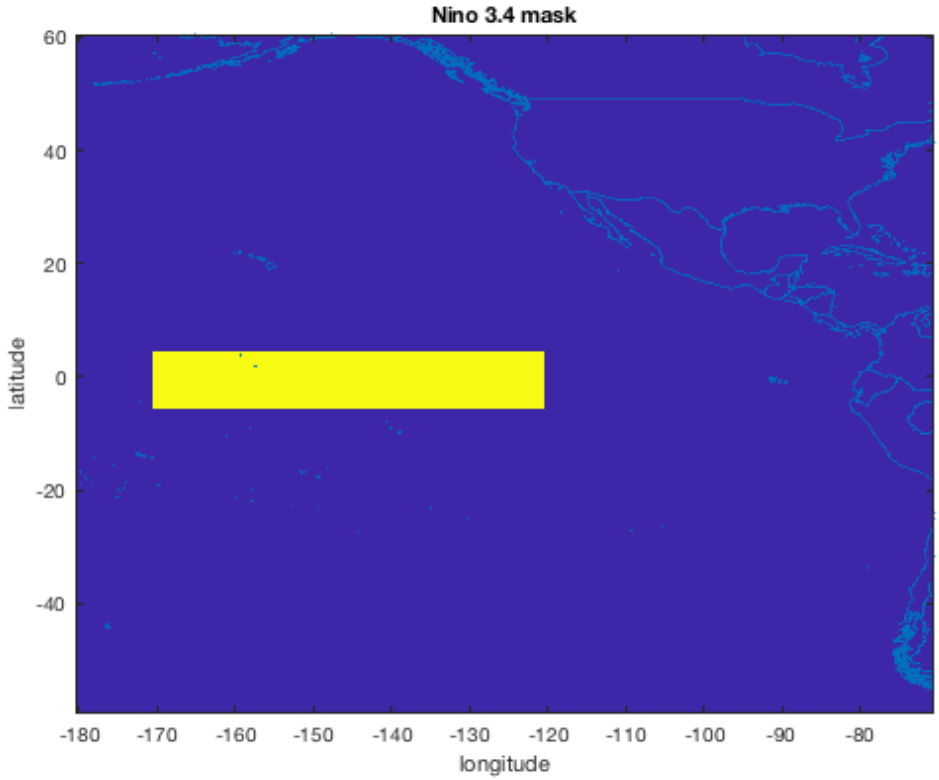
```
load pacific_sst.mat

imagescn(lon,lat,mean(sst,3))
axis xy image
cmocean thermal

xlabel 'longitude'
ylabel 'latitude'
```



定义掩膜

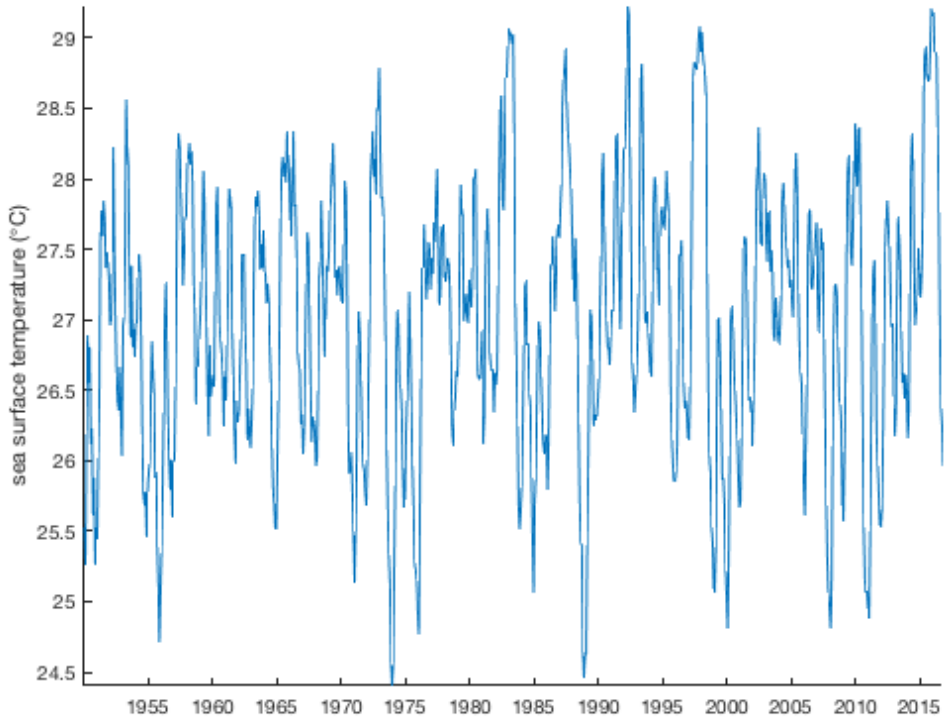
有些人通过 Nino 3.4 框内的平均海面温度来表征 ENSO，Nino 3.4 框由 5° N 至 5° S 纬度和 170° W 至 120° W 经度之间的区域定义。让我们获取该框中的平均海面温度的时间序列。首先创建与 Nino 3.4 框相对应的掩膜：

```
%将经纬度编程网格:
[Lon,Lat] = meshgrid(lon,lat);

% 在多边形内做个掩膜:
mask = geomask(Lat,Lon,[-5 5],[-170 -120]);

figure
imagescn(lon,lat,mask)
axis xy tight

title 'Nino 3.4 mask'
xlabel 'longitude'
ylabel 'latitude'

ax = axis;
borders('countries')
axis(ax)
```



黄色矩形包含网格单元，我们将在网格单元上计算平均 SST 时间序列。要在 Nino 3.4 框中获得平均 SST 的时间序列，只需给出 local 函数，sst 数据集和 mask：T

```
y = local(sst,mask);



figure

plot(t,y)

axis tight
box off
datetick('x','keeplimits')
ylabel(' sea surface temperature (\circC) ')
```

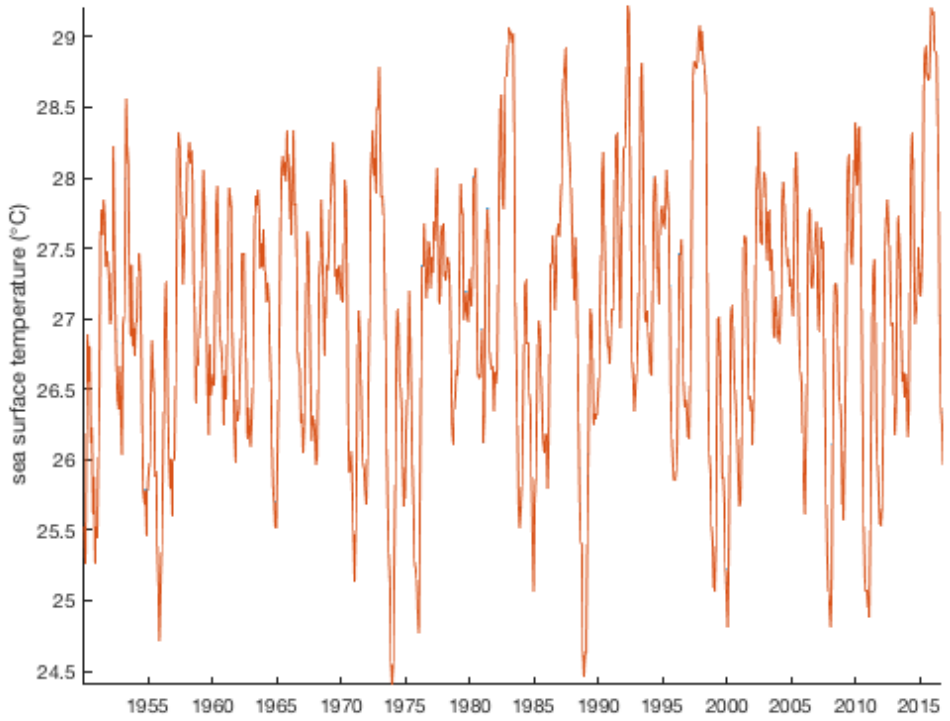


区域平均值

假设您不是“地平协会”的成员，我们将不得不面对一个不便的事实，即纬度和经度的网格单元的面积不相等。如果需要面积加权平均值，请使用 cdtarea 函数获取网格单元的面积，并在调用 local 时将这些面积用作权重：

```
Area = cdtarea(Lat,Lon);
```

```
yw = local(sst,mask,'weight',Area);

hold on
plot(t,yw)
```



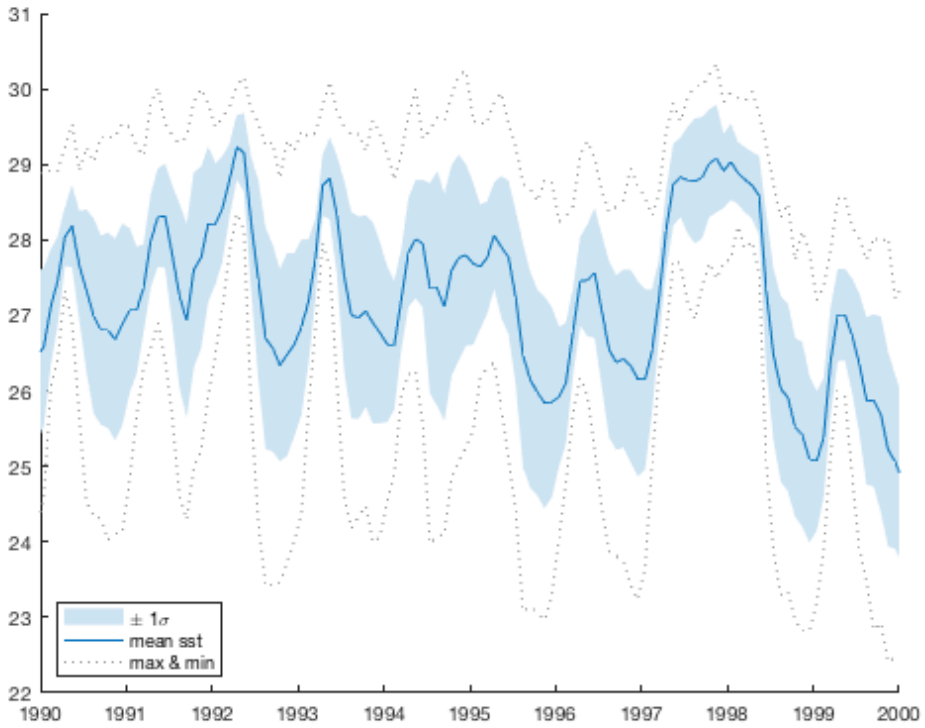
现在您可能已经注意到，面积平均值非常接近未加权平均值。这是因为 Nino 3.4 框靠近赤道，那里所有网格单元的大小都相同。Nino 3.4 框内最大和最小的网格单元的面积差异仅约 0.2%，因此在此处计算面积加权平均值可能是不必要的。
想自己看看吗？使用完整的 sst 数据集尝试上面的示例，让 `mask = isfinite(mean(sst,3));`.然后，当您考虑网格单元的尺寸时，您会发现平均值之间存在很大差异。

不只是平均值

也许您不希望 Nino 3.4 框中的平均值-也许您需要最大值和最小值。让我们根据时间来计算 Nino 3.4 框中的最大值和最小值。还要获得标准偏差并使用 boundedline 进行绘制：

```
% 在 Nino 3.4 框内计算最大最小值:
ymax = local(sst,mask,@max);
ymin = local(sst,mask,@min);
ystd = local(sst,mask,@std);

figure
boundedline(t,y,ystd);

hold on
```

```
plot(t,ymin,':','color',rgb('gray'))
plot(t,ymax,':','color',rgb('gray'))

% 放大看看发生了什么:
ylim([22 31])
xlim([datenum('jan 1, 1990') datenum('jan 1, 2000')])
datetick('x','keeplimits')

legend('\pm 1\sigma','mean sst','max & min',...
   'location','southwest')
```



更多示例

```
% 海洋掩膜:
ocean = all(isfinite(sst),3);

% 南北半球的掩膜:
north = ocean & Lat>0;
south = ocean & Lat<0;
```

这些掩膜是酱婶的：黄色是 true，蓝色是 false:

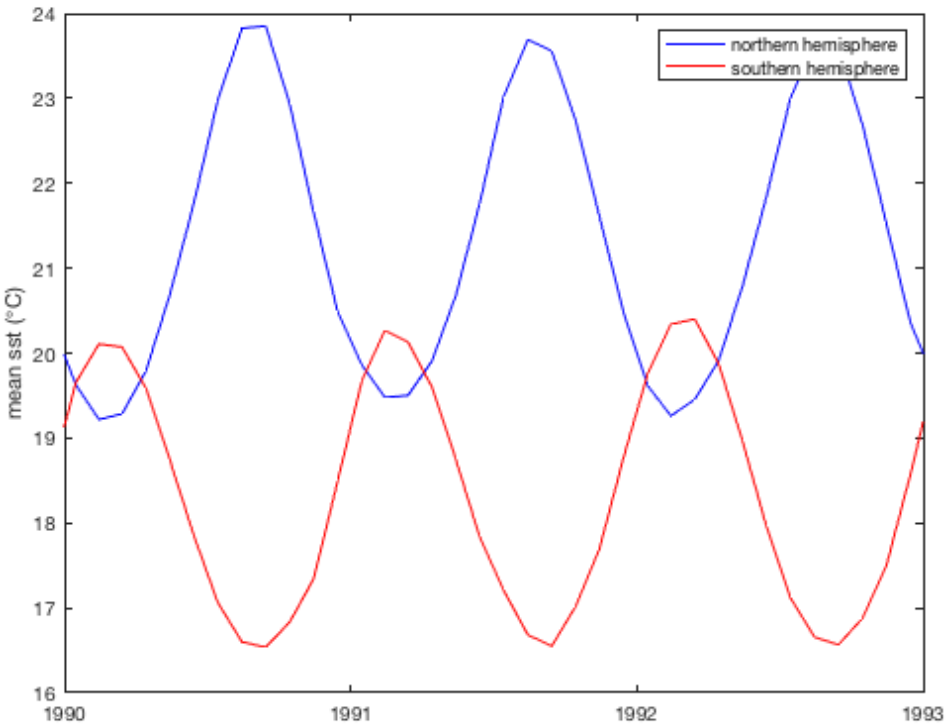
```
figure
```

```
subplot(1,3,1)

imagesc(ocean)

axis image off
title 'ocean'

subplot(1,3,2)
imagesc(north)
axis image off
title 'northern hemisphere'

subplot(1,3,3)
imagesc(south)
axis image off
title 'southern hemisphere'
```

通过定义掩膜，可以轻松比较北半球和南半球的海表温度：

```
% 计算 SST 时间序列:
ynorth = local(sst,north);
ysouth = local(sst,south);

% 绘制时间序列:
```

```
figure
plot(t,ynorth,'b')
hold on
plot(t,ysouth,'r')
ylabel(' mean sst (\circC) ')
legend('northern hemisphere','southern hemisphere')

% 放大到 3 年周期:
xlim([datenum('jan 1, 1990') datenum('jan 1, 1993')])
datetick('x','keeplimits')
```



说明

y = local(A) 从每个二维切片的平均值沿着三维矩阵 A 的维度 3 生成时间序列或轮廓。

y = local(A,mask) 给出在大网格时间序列矩阵 A 内 mask 定义的感兴趣区域内的时间序列值。对于尺寸为 MxNxP 的矩阵 A，mask 的尺寸必须为 MxN，输出 y 的尺寸是 Px1。

y = local(...,'weight',weight) 任何给定的加权网格（例如 cdtarea 生成的网格）的局部权重平均值，如果您需要区域加权平均值。

y = local(A,mask,@function) 将任何函数应用于构成 y 的每个元素的数据值。默认函数是@mean，但是您可以使用@max，@std 或几乎可以想到的任何其他函数。您甚至可以定义自己的匿名函数。

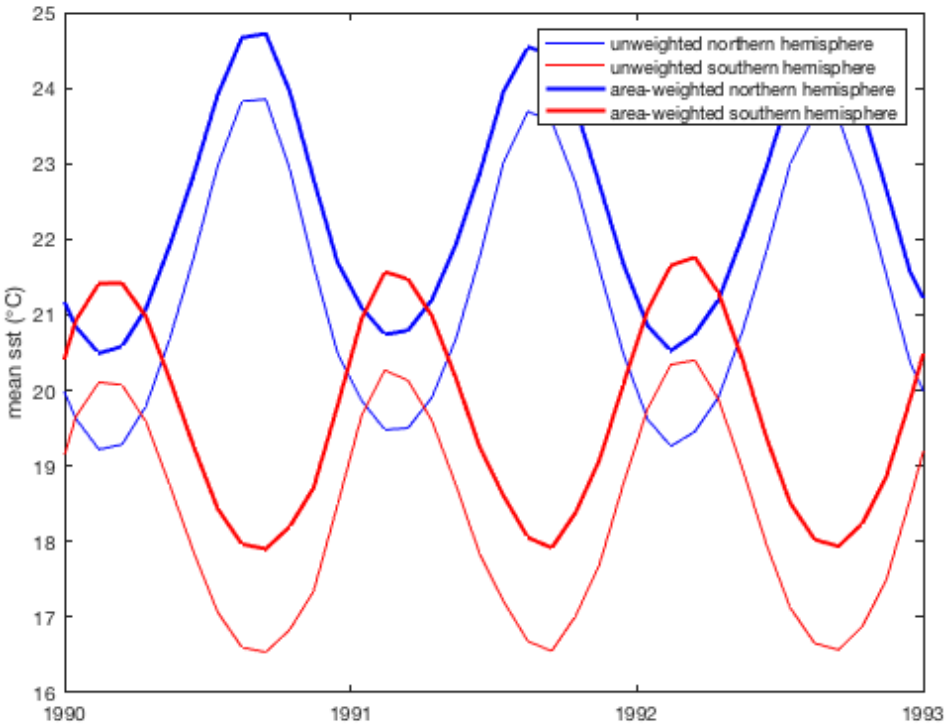
y = local(...,FunctionInputs,...) 允许对指定函数进行额外输入。例如，std 函数允许标准偏差加权方案为 0 或 1；要指定前者，请使用 y = local(A,mask,@std,0)。

y = local(...,'omitnan') 忽略 NaN 值。

面积加权平均

上面的时间序列表明，由于季节性因素，北半球和南半球的海表温度往往彼此不一致。但是在上图中，即使极向（冷）网格单元的面积小于赤道（较暖）网格单元的面积，在计算均值时每个网格单元的权重也相同。未加权意味着将全局平均值偏向极点附近的值，而不是代表真实的空间平均值。为了更好地衡量平均海面温度，请使用“面积加权”（'areaweighted'）选项：

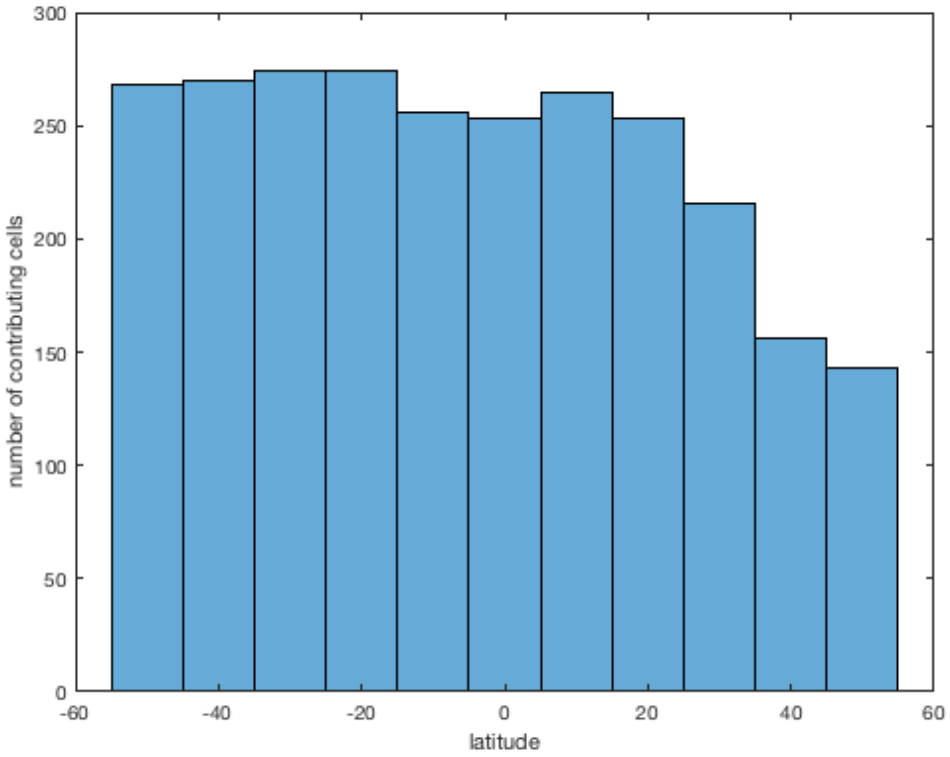
```
% 计算 SST 时间序列:
ynorthw = local(sst,north,'weight',Area);
ysouthw = local(sst,south,'weight',Area);

% 绘制时间序列:
plot(t,ynorthw,'b','linewidth',2)
plot(t,ysouthw,'r','linewidth',2)
legend('unweighted northern hemisphere','unweighted southern hemisphere',...
  'area-weighted northern hemisphere','area-weighted southern hemisphere')

% 放大到 3 年周期:
xlim([datenum('jan 1, 1990') datenum('jan 1, 1993')])
datetick('x','keeplimits')
```



面积加权的海表温度高于未加权的平均值，因为赤道附近的温暖网格单元比两极附近的冷小网格单元大得多，因此它们对真实空间平均值的贡献更大。北半球和南半球之间仍然存在几度的偏移，但这并不意味着

北半球比南半球的温度高几度。相反，这种偏见更有可能是加拿大和美国进入海洋的结果，从而减少了最北端的网格单元的总体贡献。通过纬度看一下有多少个网格单元对平均值有贡献：

```
figure

histogram(Lat(ocean),-55:10:55)

xlabel('latitude')
ylabel('number of contributing cells')
xlim([-60 60])
```

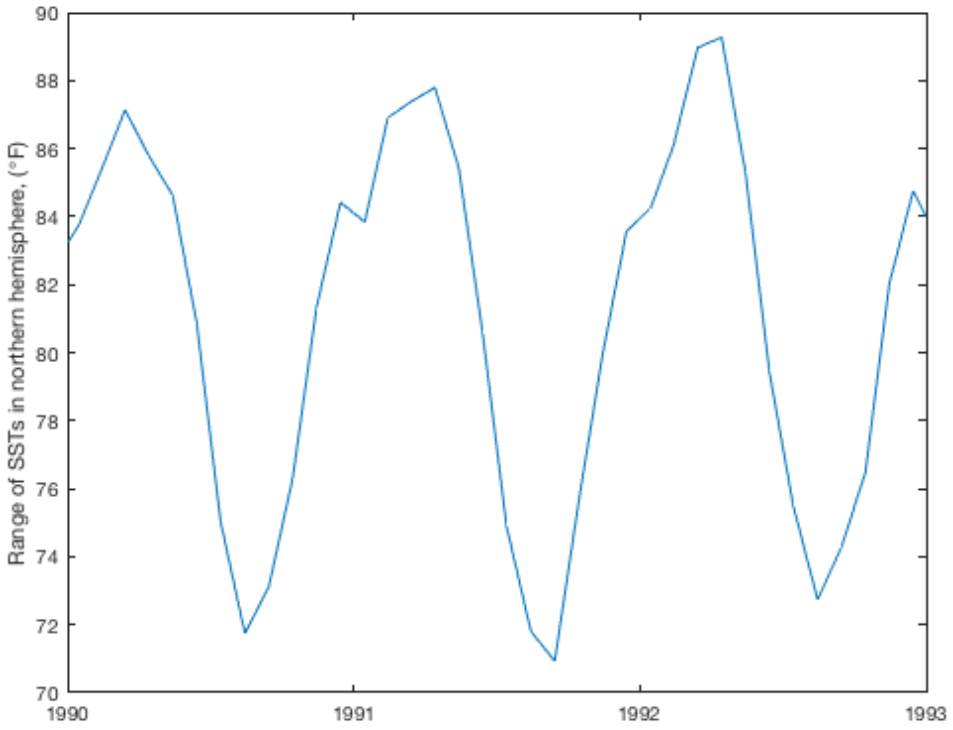


定义自己的匿名函数

也许您觉得 Matlab 提供的简单功能（如@mean，@max，@min 等）太过局限。如果需要对感兴趣区域内发生的事情进行更复杂的度量，只需定义自己的函数即可。
假设您要在一个感兴趣的区域内以华氏度为单位的总温度范围。对于任意一组温度值 x，我们需要 max(x) 减去 min(x)，然后乘以 9/5 并加 32 来转换为华氏温度：

```
% 定义温度范围（以华氏度为单位）：
fn = @(x) (max(x) - min(x))*9/5+32;

% 计算
sst_range_F = local(sst,north,fn);

figure
```

```
plot(t,sst_range_F)

% 放大到 3 年周期:
xlim([datenum('jan 1, 1990') datenum('jan 1, 1993')])
datetick('x','keeplimits')
ylabel(' Range of SSTs in northern hemisphere, (\circF) ')
```



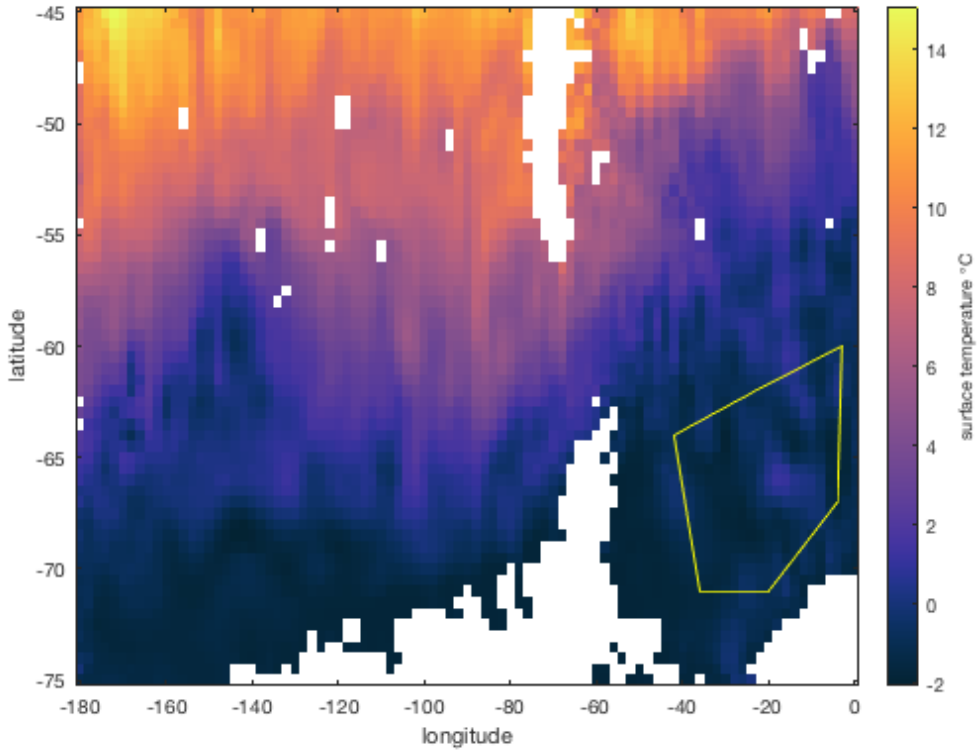
上图显示，在冬季，北极和赤道之间的温度差约为 90 度。在夏季，北极变暖，而赤道保持在相同的温度，因此温度差下降到约 72 度。

海洋剖面

local 函数不仅用于时间序列数据！任何三维网格（例如大气或海洋变量）都可以通过 local 轻松获得。您是否想要地中海的平均盐度分布？只需定义您感兴趣的区域的范围，就能获得您梦想的剖面。这是来自 Drake Passage 的 SODB 数据的示例。绘制表面温度（ptm 的第 1 层）

```
load sodb_example

figure
imagescn(lon,lat,ptm(:,:,1))
cmocean thermal
cb = colorbar;
ylabel(cb,'surface temperature \circC')
xlabel longitude
ylabel latitude
```

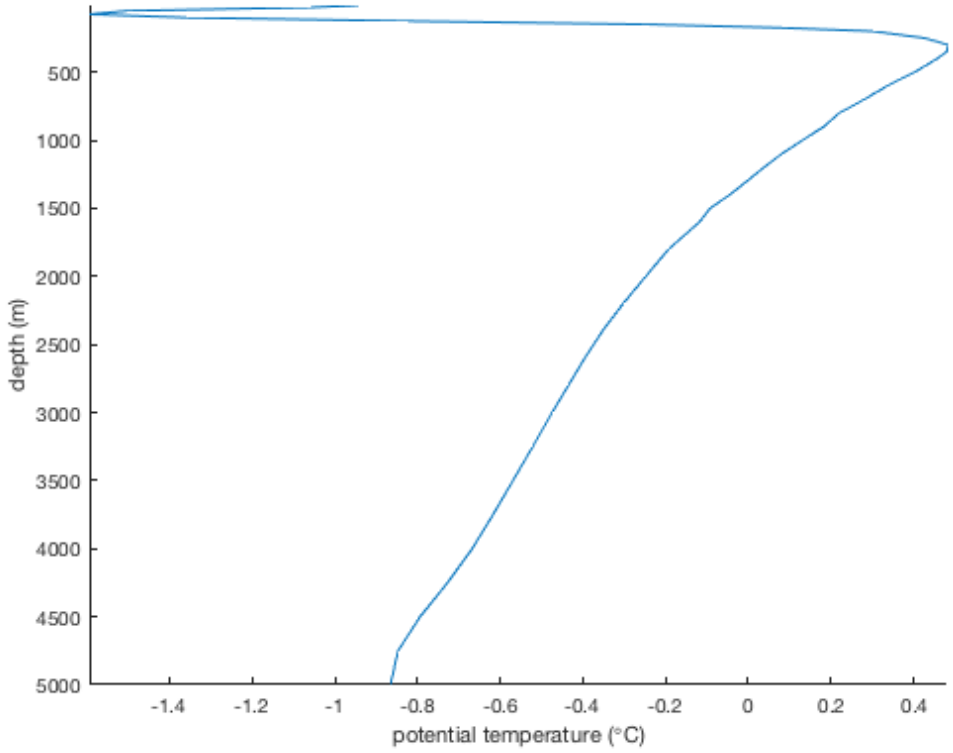
```
% 威德尔海坐标:
wlon = [-23   -42   -36   -20    -4    -3   -23];
wlat = [-62   -64   -71   -71   -67   -60   -62];

% 威德尔海外扩标黄:
hold on
plot(wlon,wlat,'y')
```



要获取威德尔海的温度剖面，将经纬度数组转为网格，将威德尔海的坐标转换为带有 geomask 的相应掩膜，然后使用 local 来获取威德尔海的平均温度剖面：

```
% 将数组转为网格:
[Lon,Lat] = meshgrid(lon,lat);

% 创建威德尔海掩膜:
mask = geomask(Lat,Lon,wlat,wlon);

% 获取温度剖面:
ptm1 = local(ptm,mask,'omitnan');

figure
plot(ptm1,d)
```

```
axis ij tight % 翻转 y 轴并消除空白
box off
xlabel 'potential temperature (\circC)'
ylabel 'depth (m)'
```



作者简介

local 函数是来自德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 于 2017 年 2 月写的。

reshapetimeseries 文档

reshapetimeseries 函数将时间序列数据的向量整形到时间 x 年的网格上，以解决与闰年或数据空间不均匀相关的混乱情况。

语法

```
[xg, yr] = reshapetimeseries(t, x)
[xg, yr] = reshapetimeseries(t, x, 'bin', nbin)
[xg, yr] = reshapetimeseries(t, x, 'bin', 'date')
[xg, yr] = reshapetimeseries(t, x, 'bin', 'month')
[xg, yr] = reshapetimeseries(t, x, 'func', @func)
[xg, yr] = reshapetimeseries(t, x, 'yrlim', yrlim)
[xg, yr] = reshapetimeseries(t, x, 'pivotdate, [mon day])
```

说明

`[xg, yr] = reshapetimeseries(t, x)` 将按时间 t 定义的时间序列 x 整形为按年的日期网格。如果原始时间序列的分辨率高于每日分辨率，则将每个日期的值取平均值。对于原始时间序列不包含任何数据的日期，将使用 NaN。最终的网格 xg 将是 365 x n 的网格，其中 n 是原始时间序列所跨越的唯一年数；yr 是保存这些年值的向量。

`[xg, yr] = reshapetimeseries(t, x, 'bin', nbin)` 每年使用 nbin 时间段重整时间序列。每个 bin 在非闰年年将保存 365 / nbin 天的数据，在闰年将保存 366 / nbin 天的数据。

`[xg, yr] = reshapetimeseries(t, x, 'bin', 'date')` `[xg, yr] = reshapetimeseries(t, x, 'bin', 'month')` 这两个'bin'属性的特殊选项允许与日历日期对齐的分类，而不是每年均分。'date'选项（未指定'bin'值时的默认选项）将 2 月 28 日至 29 日之间的数据合并在一起，从而导致每年 365 个与日历日期一致的分类，即使在闰年也是如此。'month'按日历月份对数据进行分类，因此每年产生 12 个分类。

`[xg, yr] = reshapetimeseries(t, x, 'func', @func)` 将函数句柄@func 应用于数据。@func 应该接受值的向量并返回标量。默认设置为平均数据（即@nanmean）

`[xg, yr] = reshapetimeseries(t, x, 'yrlim', yrlim)` 返回跨越年限的 1 x 2 矢量 yrlim 数据整形网格。这可用于创建比输入更长或更少年的输出数据集。

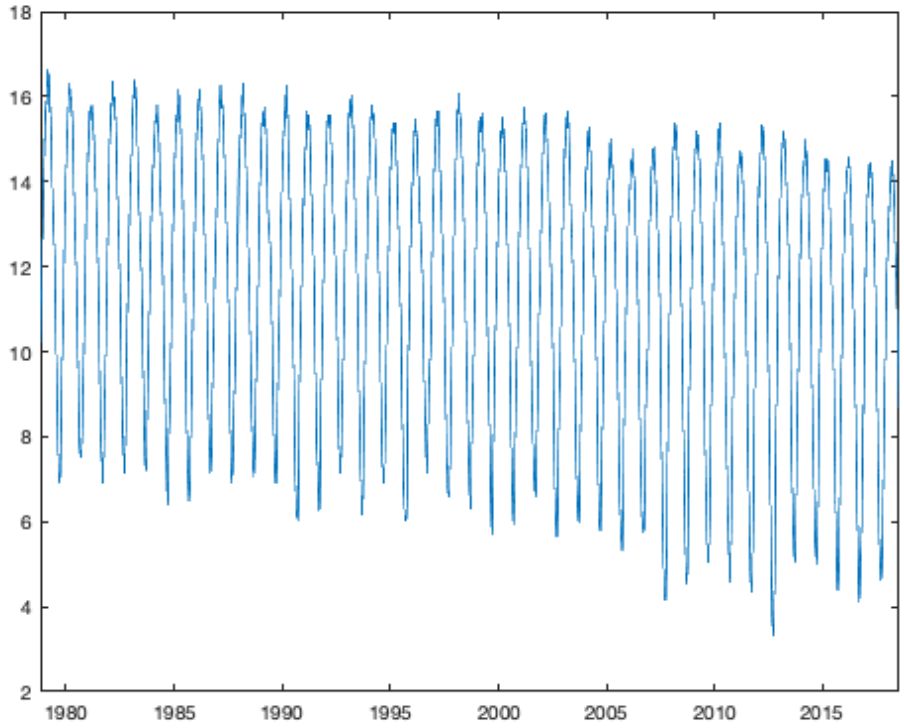
`[xg, yr] = reshapetimeseries(t, x, 'pivotdate, [mon day])` 设置数据集从一年到下一年的转折日期。默认情况下，此值为[1 1]，表示输出 xg 的每个新列都从向量 yr 中的相应年份的 1 月 1 日开始。

`[xg, yr, tmid] = ...` 返回附加输出，tmid 保存与 xg 的行相对应的日期时间。此向量中的年份设置为输入数据集中的第一个非闰年；年份的选择相对任意。此输出主要用于绘图目的，并且特定的年份可以由用户根据需要进行调整。

示例

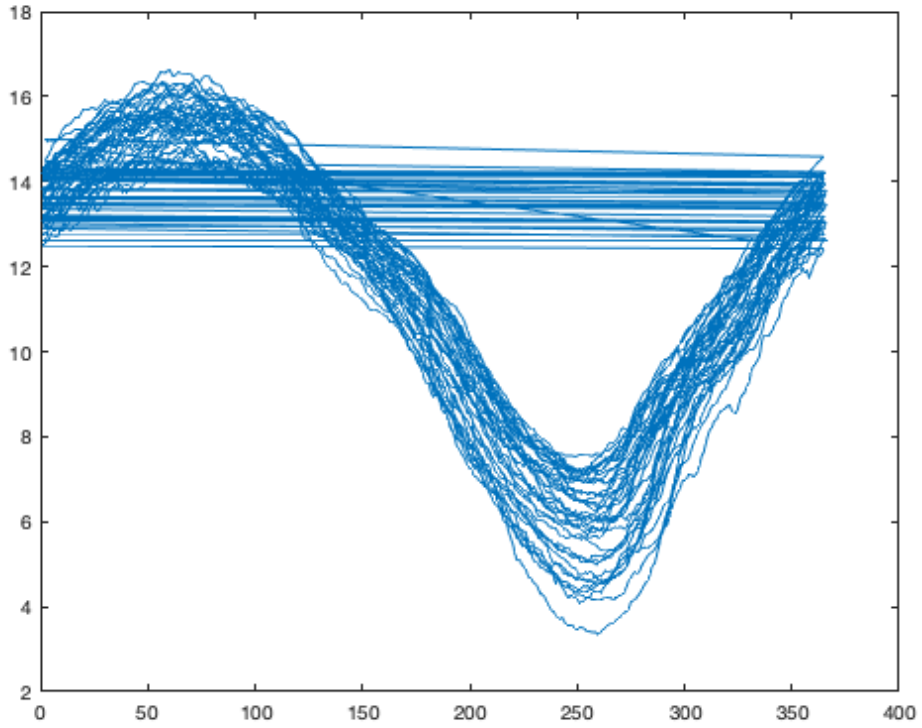
当处理时间序列数据时，通常需要按一年中的时间分析或绘制。例如，让我们看一下海冰范围示例：

```
load seaice_extent;

plot(t, extent_N);
```



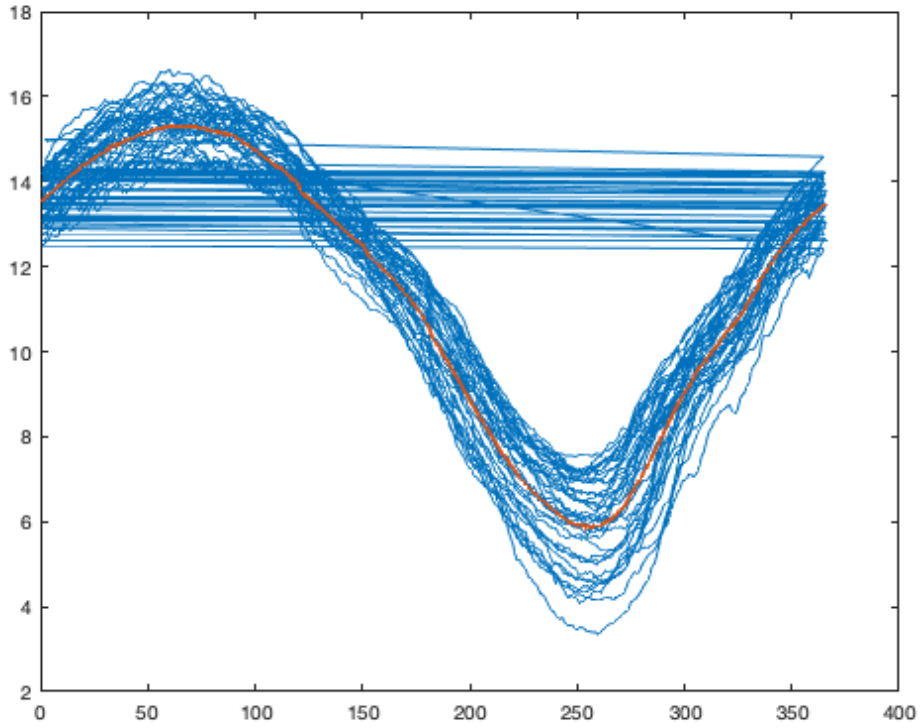
我们可以使用 doy 将这些数据放在一年中的日期轴上，而不是日期时间上：

```
cla;

plot(doy(t), extent_N);
```



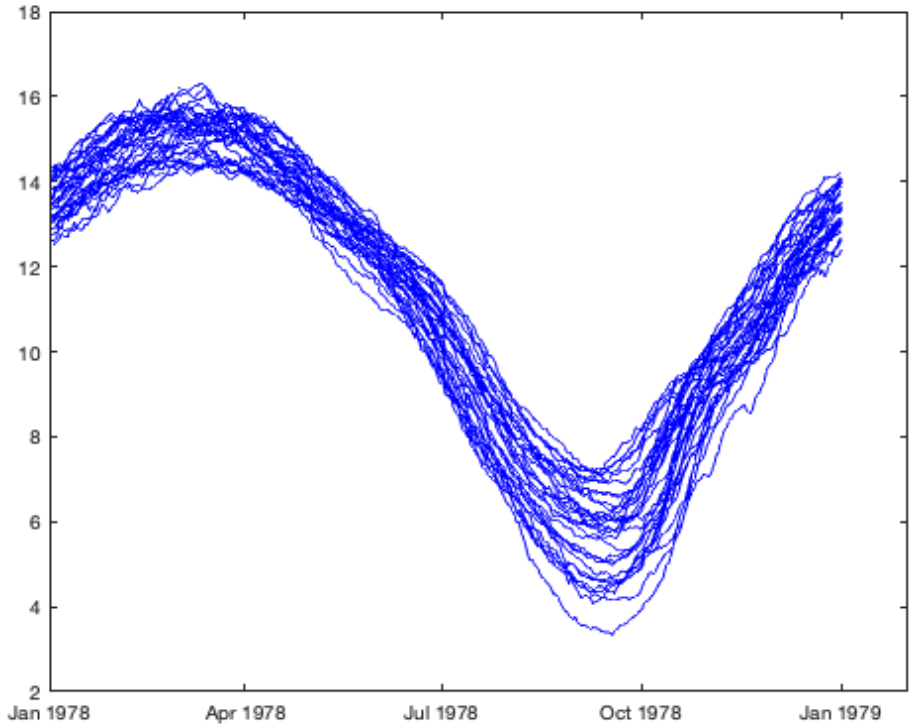
但是，从一年到下一年的转换会阻止任何快速简便的绘图。通过使用 splitapply 或 accumarray，我们可以计算多年的统计数据：

```
[g, tdoy] = findgroups(floor(doy(t)));

iceavg = splitapply(@nanmean, extent_N, g);



hold on;
plot(tdoy, iceavg, 'linewidth', 2);
```



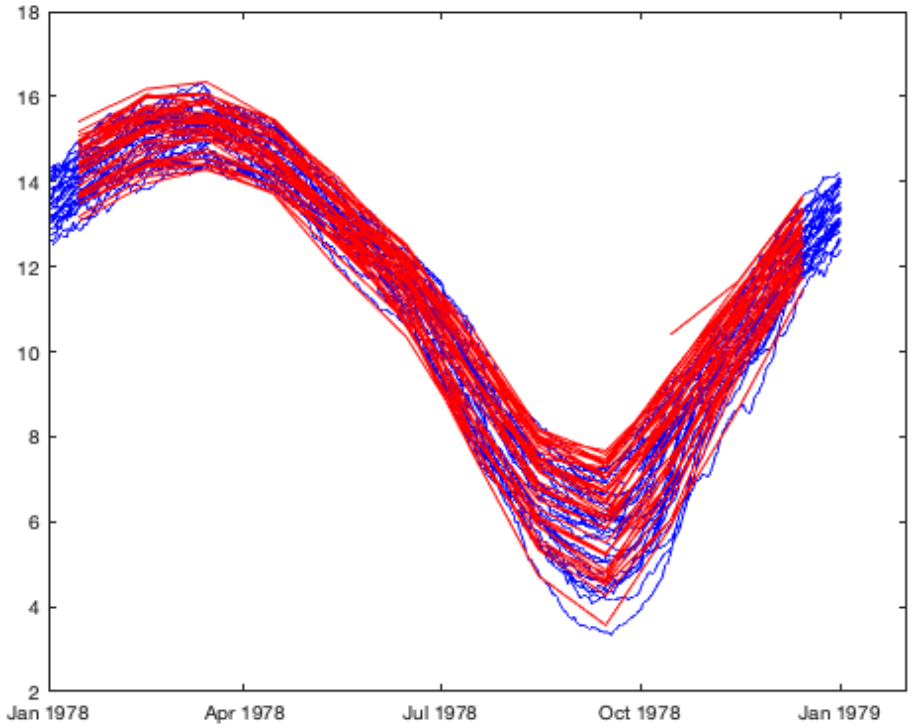
但是，当您想进行更深入的分析时，该方法会变得很笨拙。如果我们只需将数据重塑为矩阵，这两个任务将简单得多。当然，问题在于那些烦人的闰日，总是导致不同年份的数据点数量不均衡，并且引发了重塑的尝试。 eshapetimeseries 函数可以清除混乱情况：

```
[ice, yr, tmid] = reshapetimeseries(t, extent_N);



cla;

plot(tmid, ice, 'b')
hold on
```



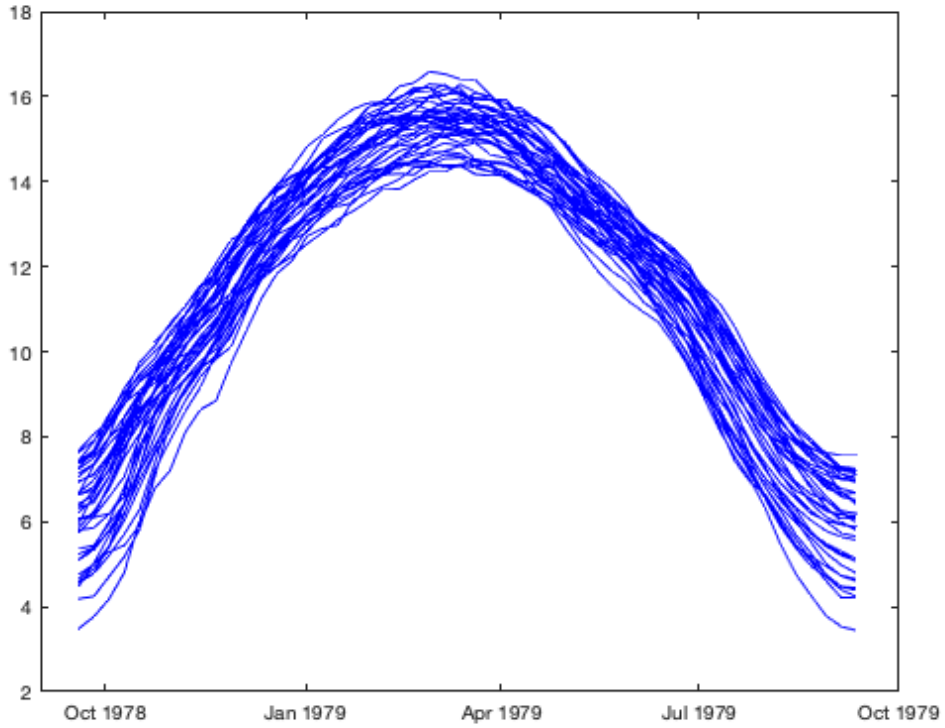
默认情况下，它使用日期分组，以确保日历日期从一年到下一年保持一致，并以闰年的平均值计算 2 月 28 日至 29 日的数据。也可以通过适当设置'bin'的输入来实现每月合并。（请注意，此数据集中的第一年始于 10 月中旬，这导致月平均值出现异常值）。

```
[icem, yr, tmidm] = reshapetimeseries(t, extent_N, 'bin', 'month');

plot(tmidm, icem, 'r')
```



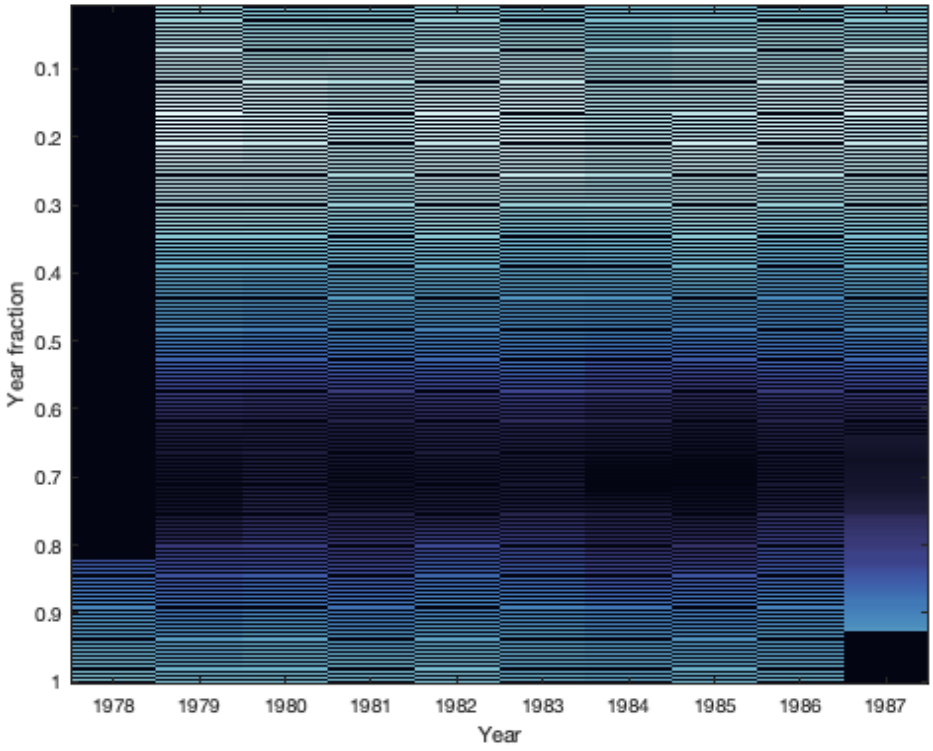
虽然通常 1 月 1 日是放置逐年换行的好地方，但有时您的数据集可能需要不同的放置。例如，我们可以在 9 月中旬将环绕效果定位在该时间段的最低点附近。在此示例中，我们每年选择 52 个均匀分布的分组，大约是每周的平均值：

```
[ice, yr, tmid] = reshapetimeseries(t, extent_N, ...
    'pivotdate', [9 15], 'bin', 52);

cla
plot(tmid, ice, 'b')
hold on
```



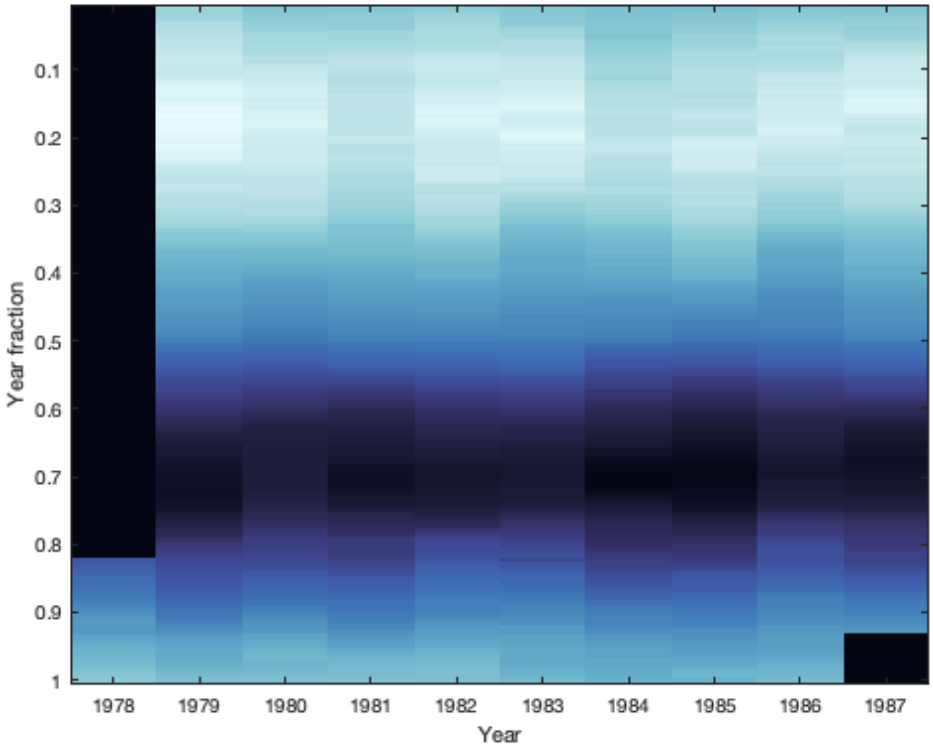
使用此功能要注意的一件事是，它将为原始时间序列中不存在任何数据的任何分组添加一个 NaN 占位符。在大多数情况下，这将与您的时间序列中较大的差距相对应，例如如果时间序列不是在同一日历日开始和结束，则在开始或结束时显示。但是，如果选择的合并间隔比原始数据要小，也会发生这种情况。例如，在此数据集中，每隔 2 天采样一次早期数据。如果我们每天尝试对此进行分类，则最终会得到稀疏的数据，每隔一年会交错一次：

```
isearly = t < datetime(1988,1,1);

[ice, yr, tmid] = reshapetimeseries(t(isearly), extent_N(isearly));



tlbl = doy(tmid, 'decimalyear')- year(tmid(1));

figure;
imagesc(yr, tlbl, ice);
shading flat;
cmocean('ice');
xlabel('Year');
ylabel('Year fraction');
```



在这种情况下，我们要确保在进行任何实际分析时选择一个较大的间隔：

```
[ice, yr, tmid] = reshapetimeseries(t(isearly), extent_N(isearly), ...
    'bin', floor(365/2));

figure;
imagesc(yr, tlbl, ice);
shading flat;
cmocean('ice');
xlabel('Year');
ylabel('Year fraction');
```



作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。函数和支持文档是由华盛顿大学的 Kelly A. Kearney 写的。

near1 文档

near1 查找最接近指定坐标的数组中点的线性索引。
另请参见 near2, geomask 和 local。

语法

```
ind = near1(x,xi)
[ind,dst] = near1(x,xi)
```

说明

ind = near1(x,xi) 返回 x 中最接近 xi 的点的索引 ind。如果 xi 在两个 x 值之间等距，则 ind 对应于两个值中的第一个。
[ind,dst] = near1(x,xi) 也返回 x(ind) 与 xi 之间的距离 dst。

示例 1

对于此 x 值数组，数组中的哪个点最接近 xi = 51？

```
x = 10:10:100

xi = 51;
```

```
x =

    10    20    30    40    50    60    70    80    90   100
```

最接近 51 的 x 值的索引是：

```
ind = near1(x,xi)
```

```
ind =

     5
```

如果您还想知道与 x(ind) 和 xi 的距离，请从 near1 获取第二个输出：

```
[ind,dst] = near1(x,xi);

dst
```

```
dst =

    -1
```

即，值 x(ind) = 50 比查询值 xi =51 低一个单位。

如果查询点位于两个 x 值之间的一半怎么办？55 介于 50 和 60 之间？

```
ind = near1(x,55)
```

```
ind =

     5
```

在有多个同等有效答案的情况下，near1 仅返回第一个有效索引。

示例：气候数据应用

near1 函数对于查找网格化气候时间序列中最靠近感兴趣位置的网格点的行和列很有用。此示例与文档 near2 和 geomask 中的示例相似，但应用方式略有不同。

```
load pacific_sst   % 载入示例数据

whos lat lon sst t %显示这些变量的尺寸
```

```
  Name        Size                    Bytes  Class     Attributes

  lat        60x1                      480  double
  lon        55x1                      440  double
  sst        60x55x802              21172800  double
  t         802x1                      6416  double
```

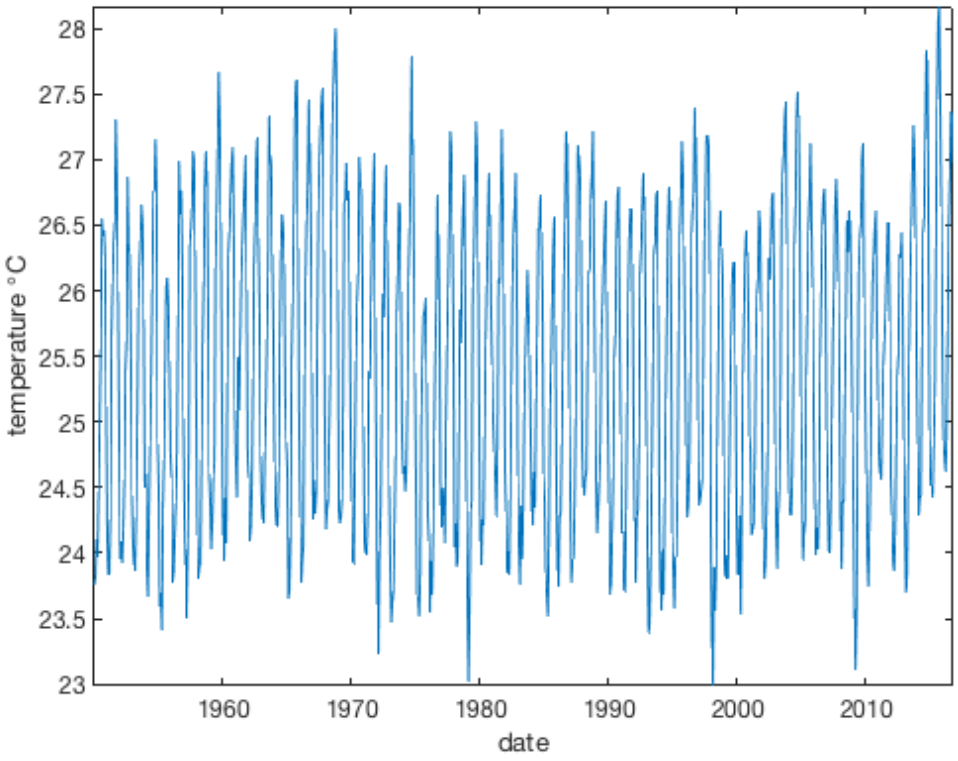
这些线性数据 lat 和 lon 对应于三维 SST 数组的前两个维度。我们可以使用 near1 来获取接近夏威夷檀香山（21.3 N，157.8 W）的 SST 的时间序列。T

```
row = near1(lat,21.3);

col = near1(lon,-157.8);
```

SST 时间序列可以这样绘制 只要记住要使用 squeeze 将得到的 1x1x802 数组转换为 802x1 形状，这样绘图就可以处理它：

```
sst1 = sst(row,col,:);
```

```
plot(t,squeeze(sst1))

axis tight
datetick('x','keeplimits')
xlabel date
ylabel 'temperature \circC'
```



为确保确定，请确认行和列实际上与靠近檀香山的经度和纬度相对应：

```
[lat(row) lon(col)]
```

```
ans =

   21.5000 -157.5000
```

是的，这差不多等于我们所能达到的（21.3 N，157.8 W）。、

作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

near2 文档

near2 查找最接近指定位置的网格点的下标索引。
另请参见 near1, geomask, 和 local.

语法

```
[row,col] = near2(X,Y,xi,yi)
[row,col] = near2(X,Y,xi,yi,mask)
[row,col,dst] = near2(...)
```

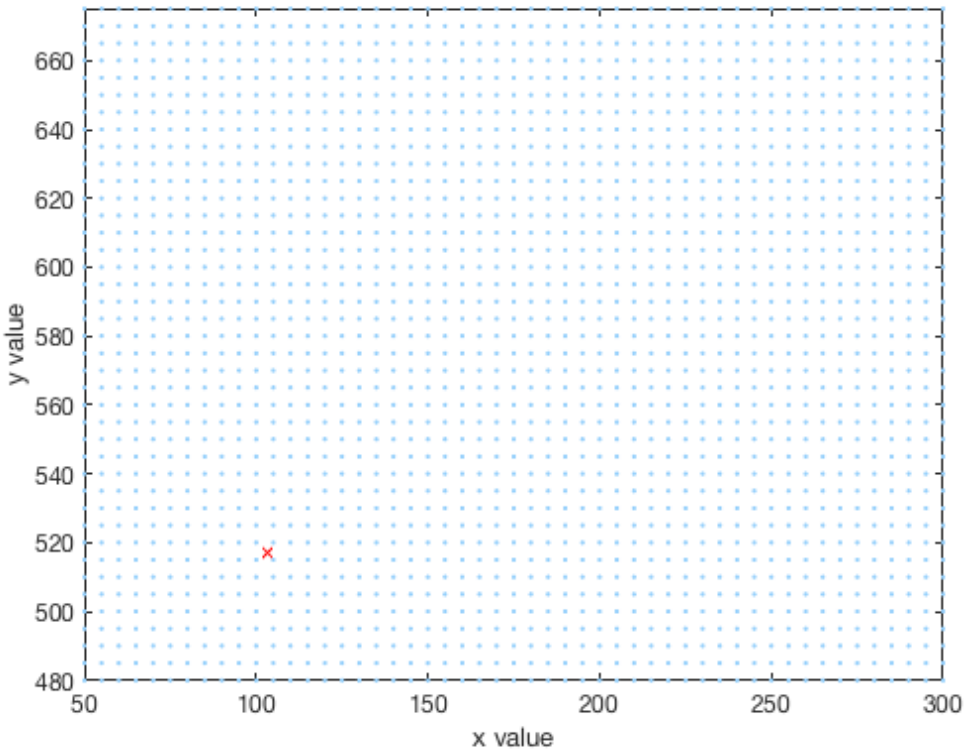
说明

[row,col] = near2(X,Y,xi,yi) 返回与点 xi，yi 的欧式距离最短的 X，Y 网格点对应的行和列。如果 xi 或 yi 在两个 X 或 Y 网格点之间等距，则仅返回两个同等有效的行或列中的第一个
[row,col] = near2(X,Y,xi,yi,mask) 忽略与错误 mask 值相对应的 X，Y 网格点。在数据边界附近，此选项可能很有用，其中数据网格中的最近点可能对应于时间序列中的 NaN 数据。
[row,col,dst] = near2(...) 还返回从点(X(row,col),Y(row,col))到点(xi,yi)的欧几里德距离。

示例 1

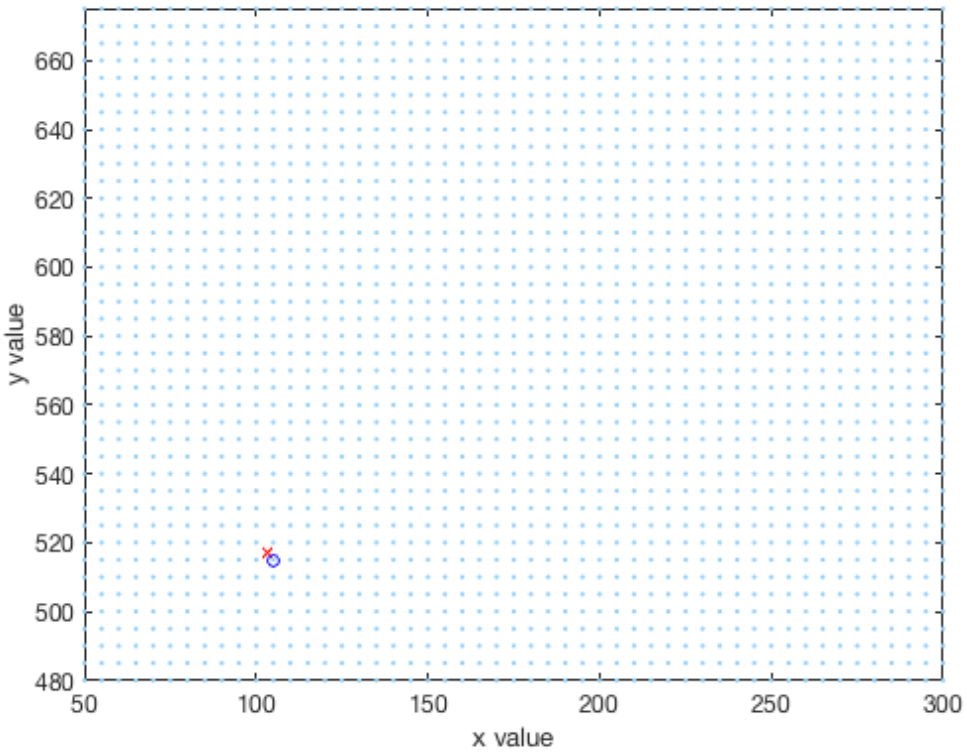
考虑这个由 X 和 Y 值组成的网格，以及一个特殊的感兴趣位置 xi，yi，我们将其标记为红色 x：

```
[X,Y] = meshgrid(50:5:300,480:5:675);



xi = 103.2;

yi = 517;



plot(X,Y,'.','color',rgb('light blue'))

hold on
plot(xi,yi,'rx')
axis equal tight

xlabel 'x value'
ylabel 'y value'
```



使用 near2 来确定最接近 xi，yi 的 X，Y 网格点的下标索引，并用蓝色圆圈标记该网格点：

```
[row,col] = near2(X,Y,xi,yi);



plot(X(row,col),Y(row,col),'bo')
```



如果您还想知道(xi,yi)到(X(row,col),Y(row,col))的距离，请请求第三个函数输出：

```
[row,col,dst] = near2(X,Y,xi,yi);

dst
```

```
dst =

    2.6907
```

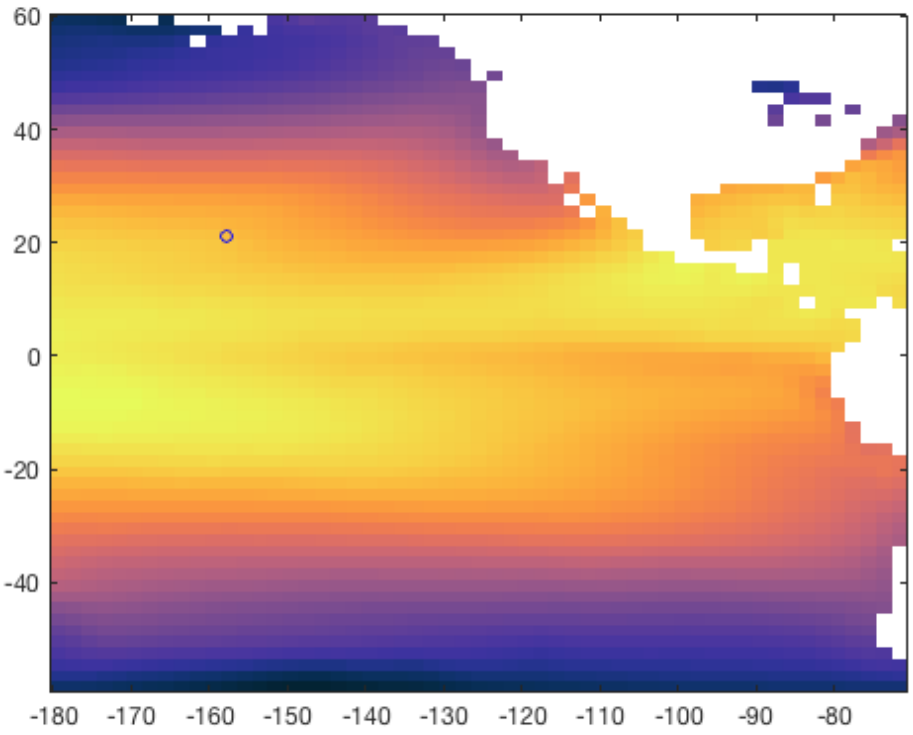
...我们看到距离感兴趣的点大约是 2.69 个单位。

示例 2: 气候数据应用

假设您想要一个夏威夷檀香山附近海面温度的时间序列（21.3 N，157.8 W）。您可以使用 geomask 和 local 进行此操作，也可以使用 near2 查找最靠近火奴鲁鲁的 SST 网格单元的行和列。情况如下：

```
load pacific_sst

[Lon,Lat] = meshgrid(lon,lat);

figure
imagescn(lon,lat,mean(sst,3))
cmocean thermal
hold on
```

```
plot(-157.8,21.3,'bo')
```



最接近檀香山的 SST 网格单元可以这样找到：

```
[row,col] = near2(Lat,Lon,21.3,-157.8)
```

```
row =

    20

col =

    12
```

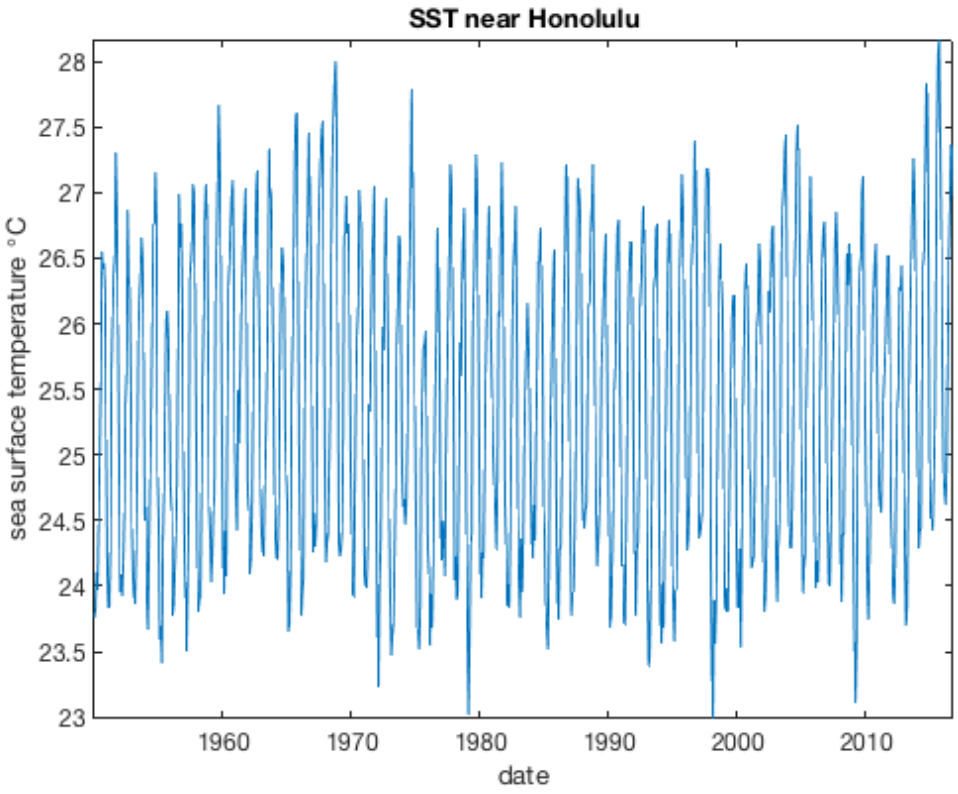
对于 row 和 col，则网格单元的地理坐标由下面给出：

```
[Lat(row,col) Lon(row,col)]
```

```
ans =

   21.5000 -157.5000
```

然后，可以轻松绘制由 row，col 给出的网格点处的 SST 时间序列。唯一的技巧是您需要使用 squeeze 命令将 1x1x802 时间序列压缩到 802x1 数组中，以便 plot 函数可以理解它：

```
sst_honolulu = squeeze(sst(row,col,:));



figure(22) % 给这个图一个号，我们以后还能回来看
plot(t,sst_honolulu)
axis tight
datetick('x','keeplimits')
xlabel 'date'
ylabel 'sea surface temperature \circC'
title 'SST near Honolulu'
```

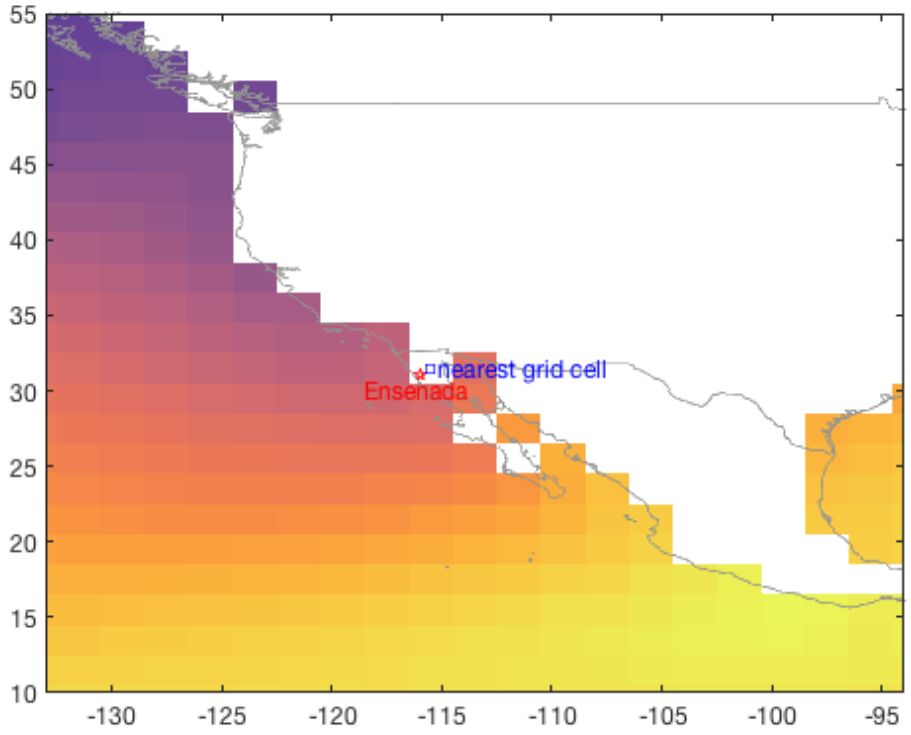


示例 3:NaN 掩膜

有时，感兴趣的位置位于边界附近，因此您尝试在最接近您感兴趣的位置的网格单元上获取时间序列，但您得到的仅仅是 NaN。例如，您可能对靠近城市的海表温度感兴趣，但是由于大多数城市都在陆地上，并且大多数海表温度都在海上，因此很有可能最接近您喜欢的城市的 SST 的时间序列是只是一堆 NaN。
例如，考虑墨西哥恩塞纳达附近的这一点：、

```
figure(33) % 给这个图一个号，我们以后还能回来看
imagescn(lon,lat,mean(sst,3))
cmocean thermal
hold on
borders('countries','color',rgb('gray'))
```

```
plot(-116,31,'rp')
text(-116,31,'Ensenada ','color','r',...
   'horiz','center','vert','top')
axis([-133 -94 10 55])

% 获得最近的网格索引并标记:
[row,col] = near2(Lat,Lon,31,-116);
plot(Lon(row,col),Lat(row,col),'bs')
text(Lon(row,col),Lat(row,col),' nearest grid cell',...
   'color','b','vert','middle')
```



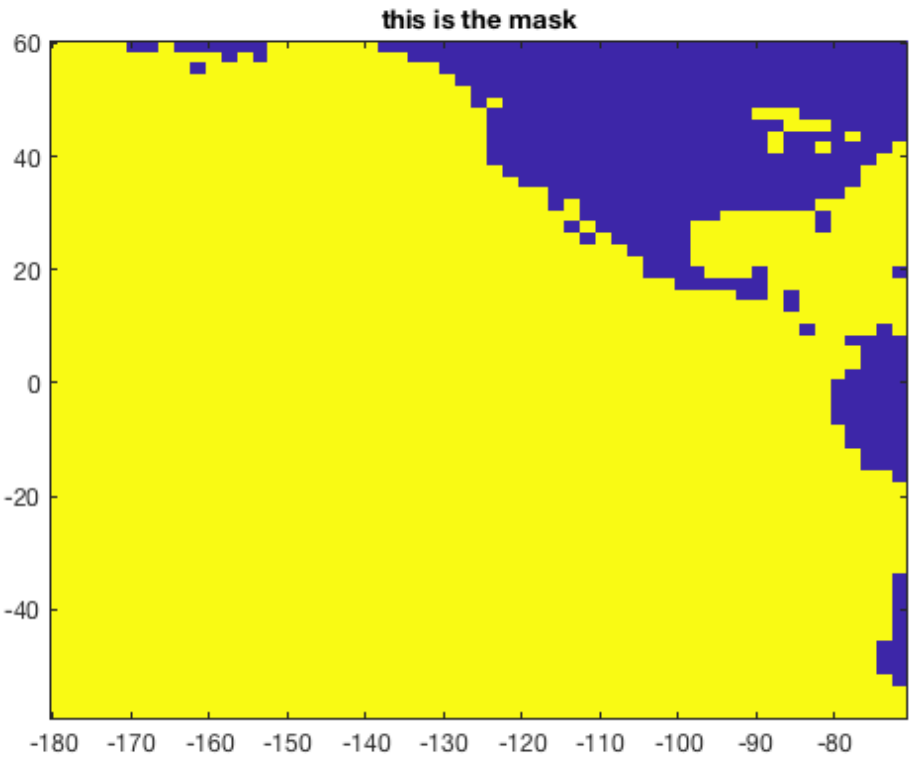
可以肯定的是，最接近恩塞纳达的网格单元中的所有 SST 数据都是 NaN：

```
% 获取恩塞纳达附近的 SST 时间序列:
sst_ensenada = squeeze(sst(row,col,:));

% 有多少恩塞纳达 SST 是有限的?
sum(isfinite(sst_ensenada))
```

```
ans =

     0
```

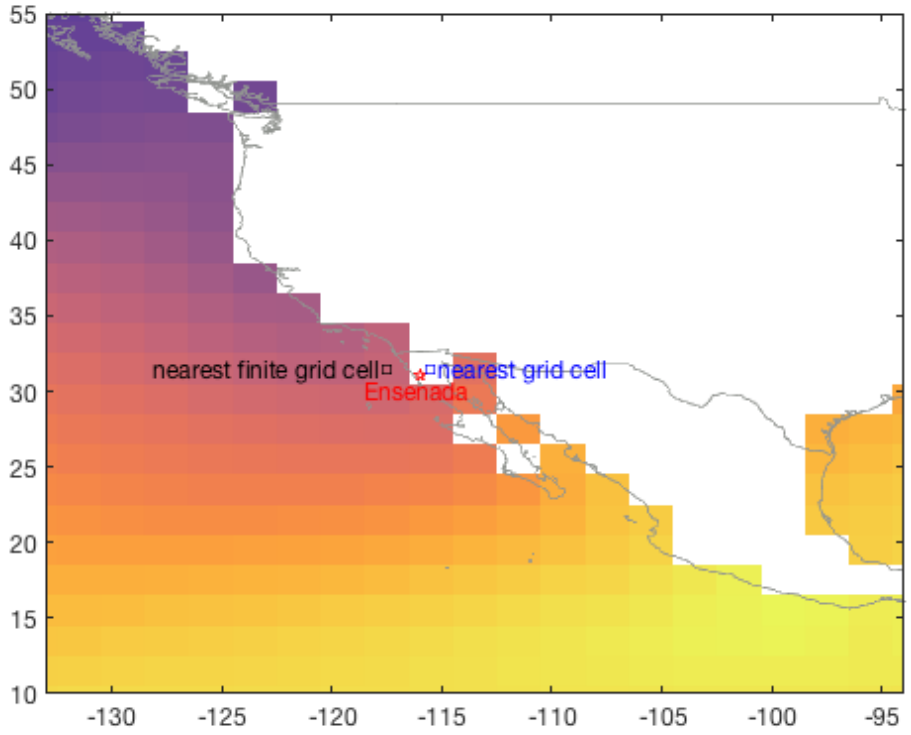
要查找数据不是 NaN 的最近的 SST 网格单元，请先定义一个良好的网格单元的掩膜，如下所示：

```
% 确定哪些网格单元格具有有限数据:
mask = all(isfinite(sst),3);

figure
imagescn(Lon,Lat,mask)
title 'this is the mask'
```



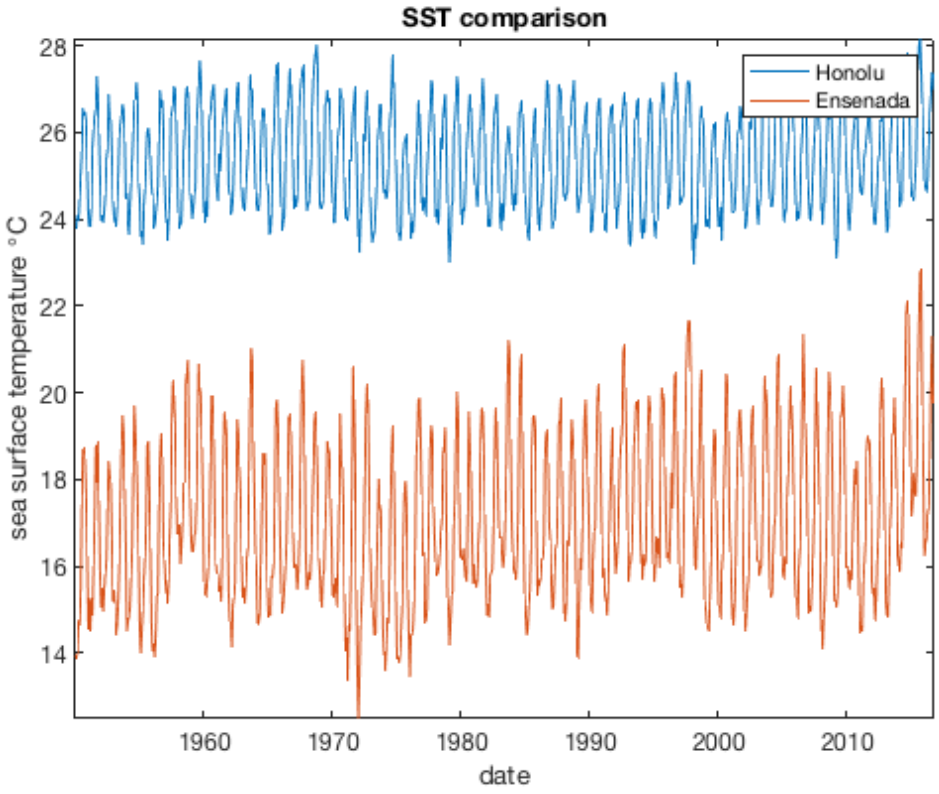
上面的掩膜以黄色显示好（真）网格单元，并以蓝色显示坏（假）网格单元。现在，我们可以将掩膜输入到 near2 中，以查找最接近恩塞纳达的网格单元：

```
[row,col] = near2(Lat,Lon,31,-116,mask);



figure(33) % 回到地图
plot(Lon(row,col),Lat(row,col),'ks')
text(Lon(row,col),Lat(row,col),'nearest finite grid cell ',...
   'horiz','right','vert','middle')
```



现在可以将类似于恩塞纳达的有限 SST 的时间序列与檀香山 SST 进行如下比较：

```
sst_ensenada = squeeze(sst(row,col,:));



figure(22) % 回到夏威夷的图
hold on
plot(t,sst_ensenada)
legend('Honolu','Ensenada')
title 'SST comparison'
```



注意地理参考网格的顺序

通常，分别跟踪纬度和经度非常重要。一些网格的构建方式

[Lat,Lon] = meshgrid(lat,lon);

而其他网格的构建方式如下

[Lon,Lat] = meshgrid(lon,lat);

这么绘制坐标

plot(lon,lat)

不会产生与以下相同的结果

plot(lat,lon)

相反，无论您将经度视为 x 还是 y 值，near2 都将始终产生相同的结果。也就是说，以下两种语法都会产生最接近檀香山的 SST 网格单元的正确行和列：

```
[row,col] = near2(Lat,Lon,21.3,-157.8)



[row,col] = near2(Lon,Lat,-157.8,21.3)
```

```
row =

    20

col =
```

```
    12

row =

    20

col =

    12
```

唯一需要注意的是，对于地理参考网格，您可能不想使用可选的第三个输出 dst。那是因为以经度表示的欧几里得距离没有任何意义，因为经度表示的距离与纬度表示的距离并不相同。

作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

cell2nancat 文档

The cell2nancat 将元胞数组的元素连接到 NaN 分离的载体中。

语法

```
xv = cell2nancat(xc)
[xv,yv,...,zv] = cell2nancat(xc,yc,...,zc)
```

说明

xv = cell2nancat(xc) 将元胞数组 xc 的数字数组连接到一个由 NaN 分隔的单个向量中。
[xv,yv,...,zv] = cell2nancat(xc,yc,...,zc) 如上所述，但适用于任意数量的单元格数组。

示例

CDT 附带了美国和美国所有州的边界数据。这是它的格式：

```
load borderdata
whos % 在工作区显示每个变量的名称和尺寸
```

```
  Name          Size              Bytes  Class     Attributes


  GEOM          1x4                  32  double
  clat        302x1                36240  cell
  clon        302x1                36240  cell
  lat         302x1              4629240  cell
  lon         302x1              4629240  cell
  places      302x1                40192  cell
```

这些都是元胞数组。这 302 个数组中的每个阵列都包含不同州或国家/地区的边界。要查看其外观，以下是前三个国家/地区的名称和纬度：

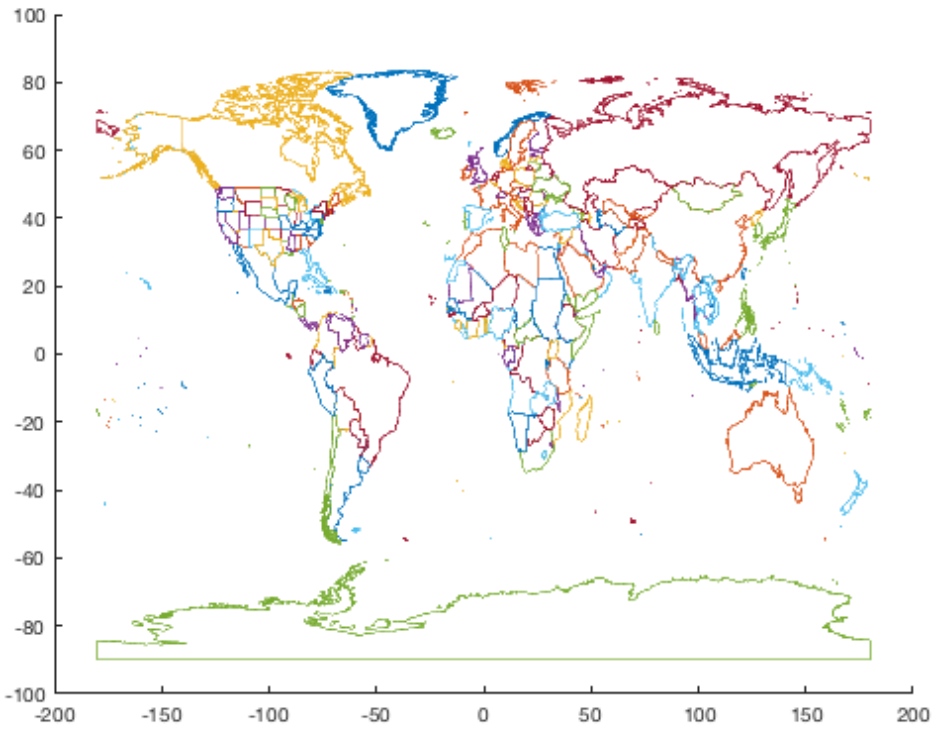
```
[places(1:3) lat(1:3)]
```

```
ans =

  3×2 cell array

    {'Antigua and Barbuda'}    {1×50  double}
```

```
{'Algeria'          }    {1×1242 double}

{'Azerbaijan'       }    {1×876  double}
```

如果要绘制所有边界线，您的第一个倾向可能是遍历 302 个条目中的每一个并分别绘制它们，如下所示：（译者注：此图存在政治问题。台湾是中华人民共和国不可分割的一部分！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
hold on
for k = 1:302
   plot(lon{k},lat{k})
end
```



但是，循环在计算上很慢，您可能不希望在图中绘制 300+个不同的对象。因此，您可以将每个单元格数组变成一个向量，然后将所有内容绘制在一起。

这是 cell2nancat 的工作方式：给它一个单元格数组，并得到一个向量返回：

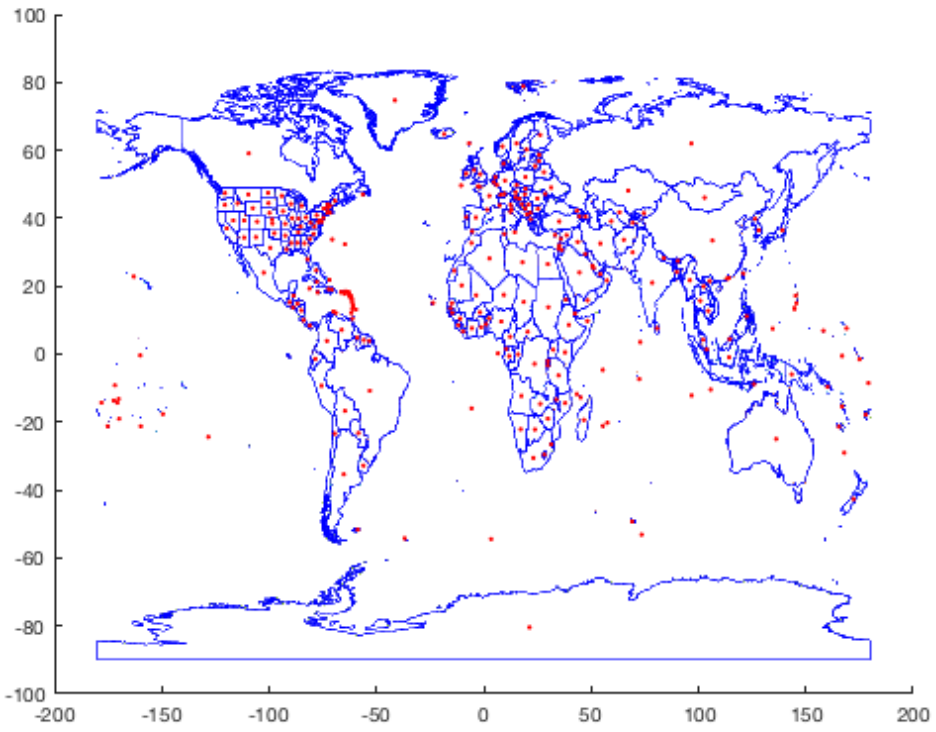
```
latv = cell2nancat(lat);



whos lat latv
```

```
  Name          Size                Bytes  Class     Attributes
```

```
lat          302x1              4629240  cell

latv       574729x1             4597832  double
```

现在，将 302x1 单元格数组变成了一个 574729x1 矢量数组，其中包含来自 lat 的所有边界线。要在多个元胞数组上使用该功能，只需将它们全部输入到 cell2nancat 中：（译者注：此图存在政治问题。藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
[latv,lonv,clatv,clonv] = cell2nancat(lat,lon,clat,clon);



plot(lonv,latv,'b') % 蓝色边界
hold on
plot(clonv,clatv,'r.') % 陆地质心为红点
```



作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

xyz2grid 文档

xyz2grid 将规则间隔的列化 x、y、z 数据转换为网格数据。另请参见 xyzread。

语法

```
Z = xyz2grid(x,y,z)
Z = xyz2grid(filename)
Z = xyz2grid(filename,Name,Value)
[X,Y,Z] = xyz2grid(...)
```

说明

Z = xyz2grid(x,y,z) 假设 x 和 y 具有某种形式的规则性，并将 z 的对应值放入规则的二维 MxN 网格矩阵 Z 中。

Z = xyz2grid(filename) 从三列（x，y 和 z）的.xyz 文件中加载数据，然后将数据放入网格中。 此函数假定输入的 x，y，z 数据具有一定的网格规则性，但是可能缺少一些数据点。

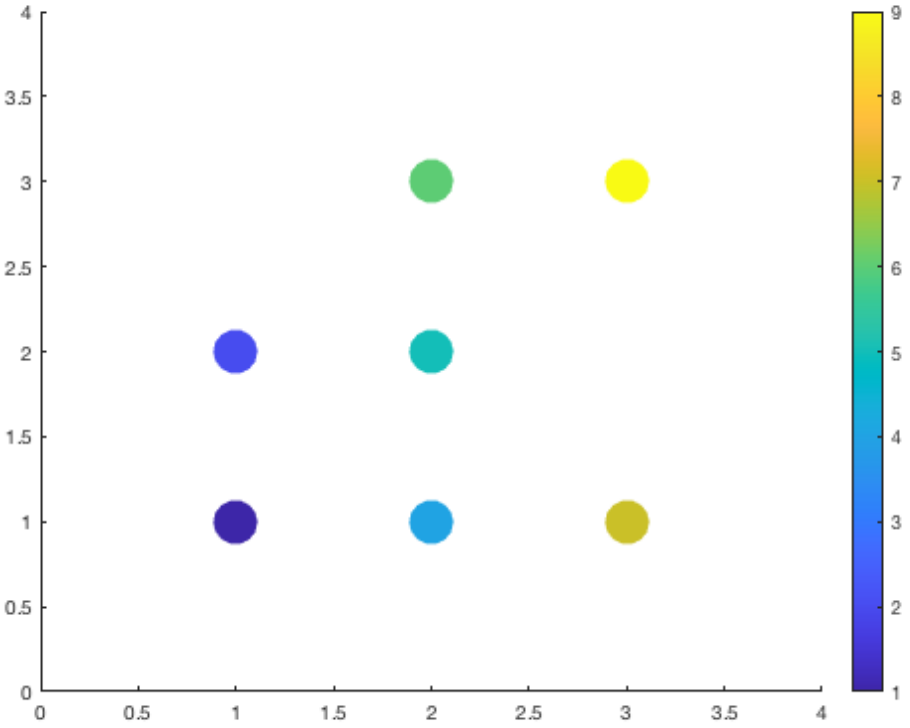
Z = xyz2grid(filename,Name,Value) 将打开一个带有 textscan 选项 Name，Value 的.xyz 文件，例如'headerlines',1。

[X,Y,Z] = xyz2grid(...) 返回与 Z 中的值相对应的二维网格划分的 X 和 Y 矩阵。

示例 1

您可能会有一些 x，y 和 z 值的列：

```
x = [1 1 2 2 2 3 3];

y = [1 2 1 2 3 1 3];

z = [1 2 4 5 6 7 9];



scatter(x,y,500,z,'filled')
axis([0 4 0 4])
colorbar
```



从上面的散点图中，您可以看到数据存在一定的网格规律性，即使网格中缺少几个点。没关系。让我们整理一下：

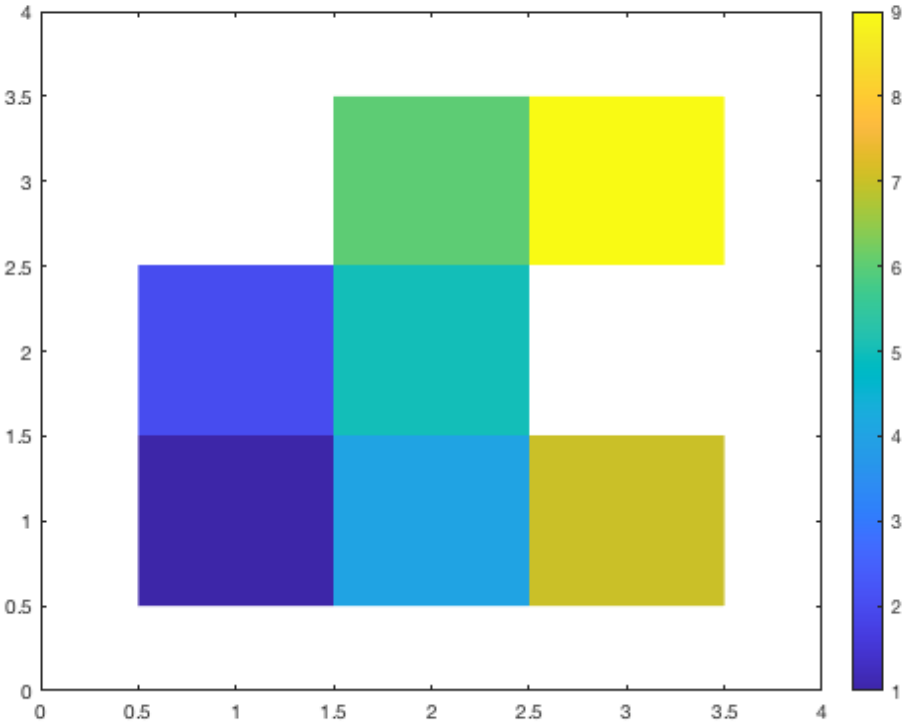
```
[X,Y,Z] = xyz2grid(x,y,z)
```

```
X =

     1     2     3

     1     2     3

     1     2     3

Y =

     3     3     3

     2     2     2

     1     1     1

Z =

   NaN     6     9
```

```
     2     5   NaN

     1     4     7
```

您会看到两个缺失值由 NaN 填充。您可以使用 surf(X,Y,Z)或 pcolor(X,Y,Z)绘制栅格化数据，但让我们使用 imagescn 代替：

```
imagescn(X,Y,Z)

axis([0 4 0 4])

colorbar
```

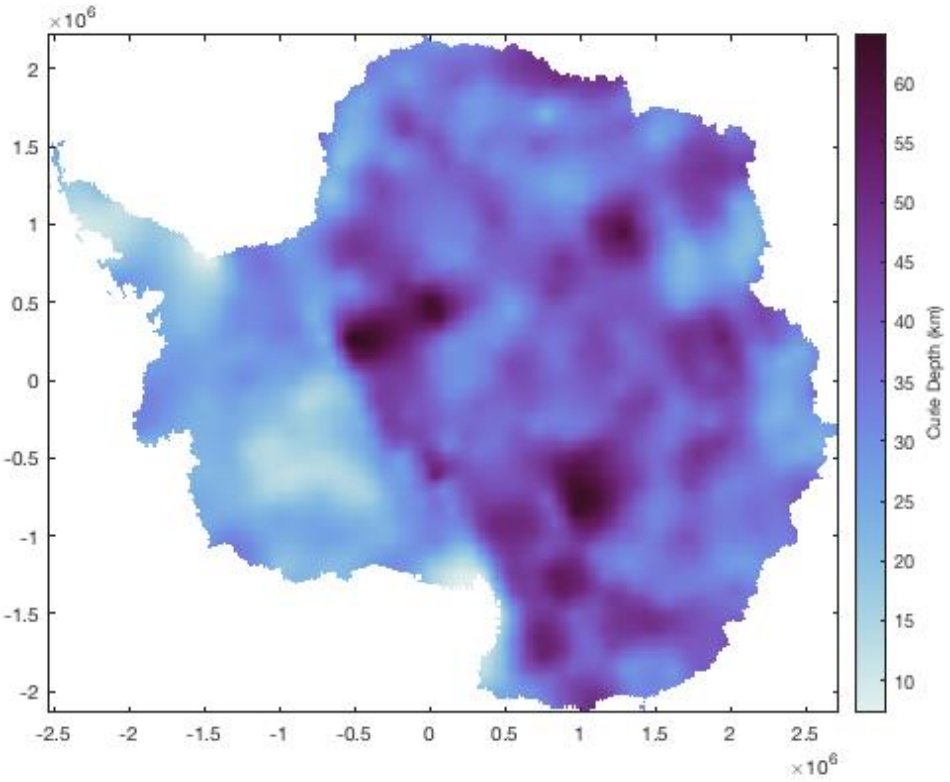


示例 2：南极居里深度

对于此示例，载入 Martos 2017 的数据并绘制南极居里深度：

```
[X,Y,CD] = xyz2grid('Curie_Depth.xyz');

figure
imagescn(X,Y,CD)
cmocean dense

cb = colorbar;
```

```
ylabel(cb,'Curie Depth (km)')
```



作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

C2xyz 文档

C2xyz 将等值图矩阵（由 contour 函数返回）转换为 x、y 和相应的 z 坐标。

语法

```
[x,y] = C2xyz(C)
[x,y,z] = C2xyz(C)
```

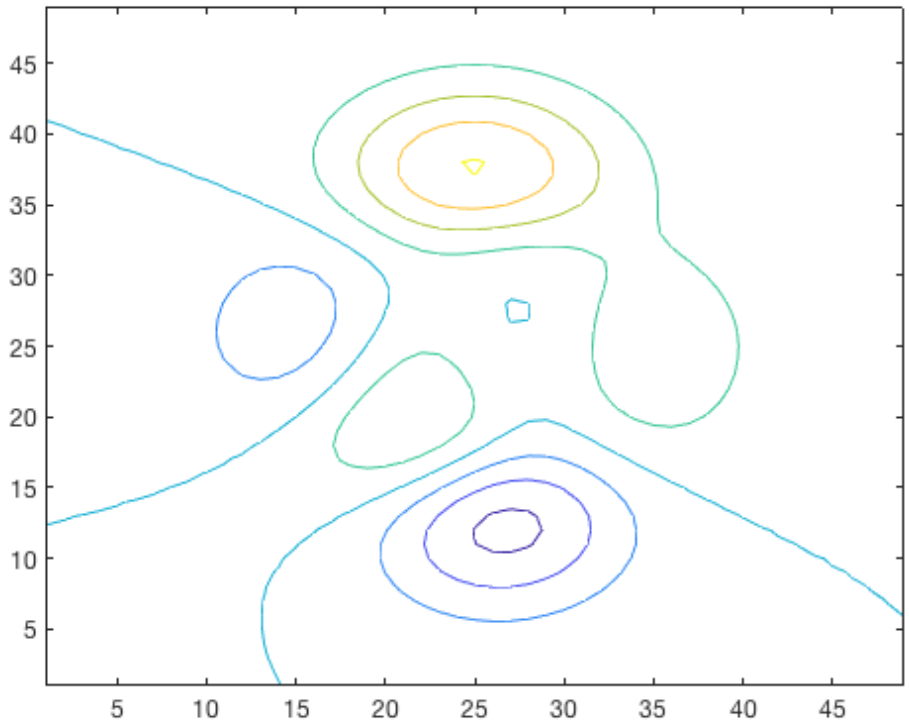
说明

[x,y] = C2xyz(C) 返回轮廓矩阵 C 中轮廓坐标的 x 和 y 元胞数组。
[x,y,z] = C2xyz(C) 还以双精度返回相应的 z 值。

示例

给定等值图，您想知道轮廓的(x,y)坐标以及与每个轮廓线相对应的 z 值。

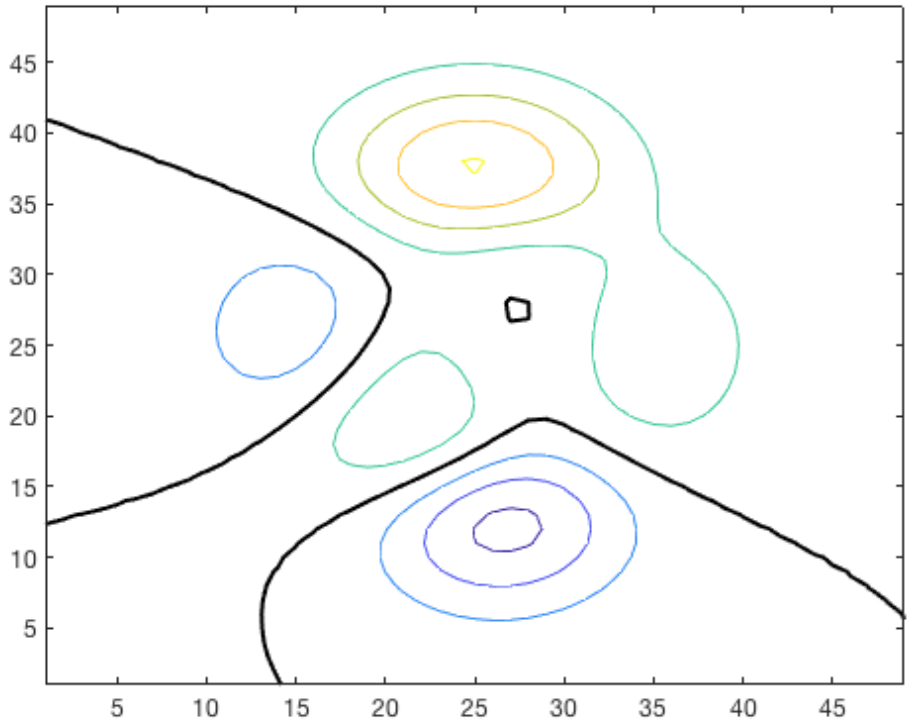
```
C = contour(peaks);

[x,y,z] = C2xyz(C);
```



这将返回 1 个 x 值和 y 值的 1 x numberOfContourLines 元胞数组，并且它们相应的 z 值在 1 x numberOfContourLines 数组中给出。让我们挑选出 z = 0 的所有 x，y 位置，并将该轮廓线变成粗黑线：

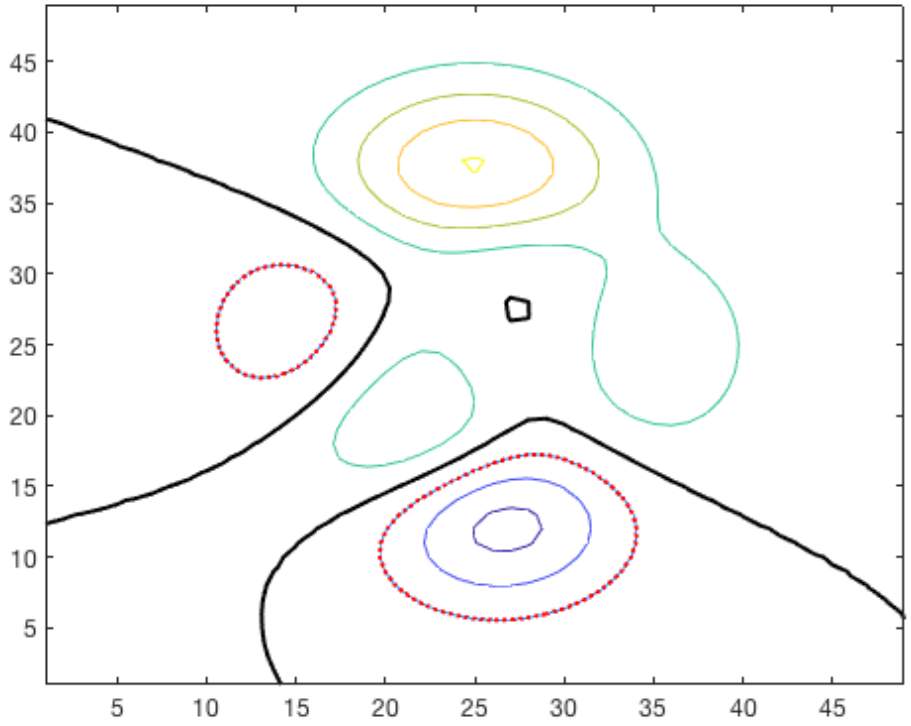
```
hold on; %允许在预先存在的峰值图上方绘制图。
```

```
for n = find(z==0) % 仅允许中 z = 0 循环。
    plot(x{n},y{n},'k','linewidth',2)
end
```



让我们将所有 z = -2 线设为红色并加点：

```
for n = find(z==-2) % z = -2 新循环。
    plot(x{n},y{n},'r:','linewidth',2)
end
```



作者简介

由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 于 2013 年 8 月创建，于 2014 年 8 月更新以包含在 Matlab 的南极绘图工具中(Greene et al., 2017)，于 2019 年再次包含在 Climate Data Toolbox for Matlab 更新。

xyzread 文档

xyzread 函数导入.xyz 文件的 x、y、z 列。注意：.xyz 文件没有真正的标准，因此您的.xyz 文件可能与我为此编写的.xyz 文件不同。我是为 GMT/GIS 文件编写的。另请参见 xyz2grid。

语法

```
[x,y,z] = xyzread(filename)
[x,y,z] = xyzread(filename,Name,Value)
```

说明

[x,y,z] = xyzread(filename) 导入纯.xyz 文件的列。

[x,y,z] = xyzread(filename,Name,Value) 接受任何 textscan 参数，例如'headerlines'等。

示例

对于此示例，请载入 Martos 2017 的南极居里深度数据：

```
[x,y,z] = xyzread('Curie_Depth.xyz');
```

上面的语法等效于

```
[x,y,z] = xyzread('Curie_Depth.xyz','headerlines',0);
```

因为此特定的 xyz 文件没有任何标题行。

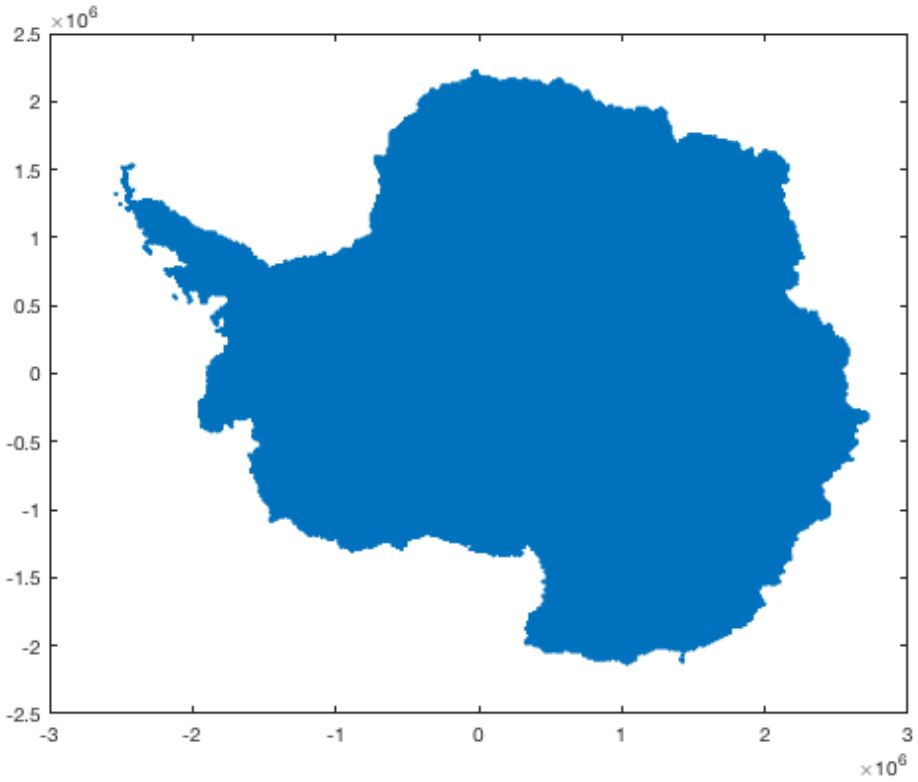
现在，我们已将数据加载到 Matlab 中，如下所示：

```
whos
```

```
  Name          Size              Bytes  Class     Attributes

  x         59328x1            474624  double
  y         59328x1            474624  double
  z         59328x1            474624  double
```

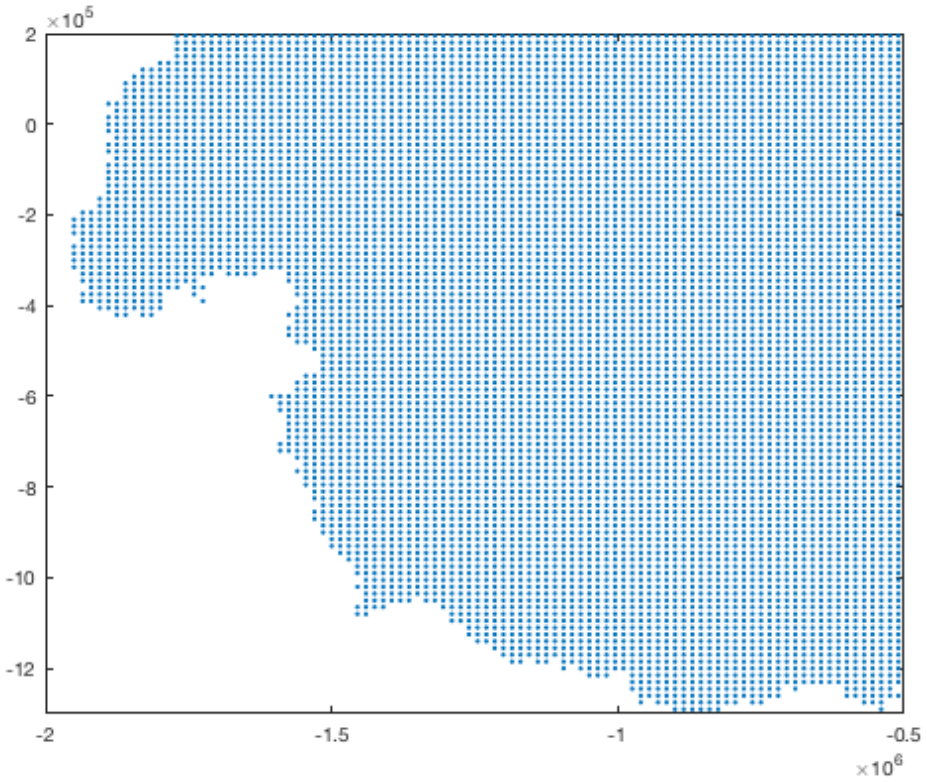
我们有三个变量，每个变量的大小为 59328x1。以下是绘制为点的 x，y 坐标：

```
plot(x,y,'.')
```



在此缩放级别上，这看起来像一块坚实的大陆，但这仅是因为∽60,000 个点都被挤压在一起了。放大一点以查看网格的规则性：

```
axis([-2 -0.5 -1.3 .2]*1e6)
```



这些点以 x 和 y 规则地间隔开，但是在 Matlab 中将它们放入规则的网格中有点像面条刮擦。幸运的是，xyz2grid 使操作变得简单！

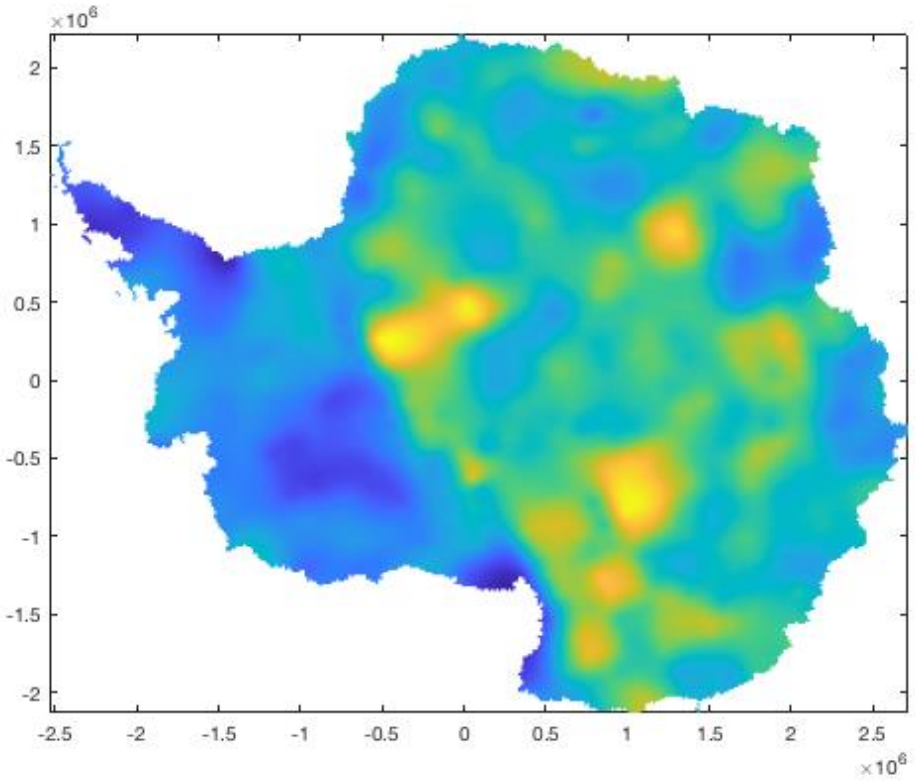
```
[X,Y,Z] = xyz2grid(x,y,z);



whos X Y Z
```

```
  Name        Size              Bytes  Class     Attributes

  X         291x350            814800  double
  Y         291x350            814800  double
  Z         291x350            814800  double
```

现在，这些 59328x1 数组已经整齐地打包到 291x350 网格中，可以使用 pcolor 进行绘制：

```
pcolor(X,Y,Z)
```

```
shading flat
```



作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

地理参考网格

- recenter 重新整理栅格化的数据集，使其以指定的经度为中心。
- cdtgrid 使用 meshgrid 轻松创建纬度和经度的全球网格。
- cdtdim 假定球形地球半径为 6371000 米，从而给出了经纬度网格中每个单元的近似尺寸。
- cdtarea 假定球形地球半径为 6371000 米，给出了经纬度网格中每个单元的近似面积。该功能旨在简化大型栅格气候数据集的面积平均加权。
- cdtgradient 计算在地理坐标中等间隔网格数据的空间梯度。
- cdtdivergence 计算椭圆形地球表面上的网格矢量的散度。
- cdtcurl 计算椭圆形地球上网格矢量的曲率的 z 分量。
- geomask 确定地理位置是否在给定的地理区域内。
- island 确定地理位置是对应于陆地还是水域
- binind2latlon 将正弦网格的合并索引值转换为地理坐标。

recenter 文档

recenter 函数重新整理栅格化的数据集，使其以指定的经度为中心。

语法

```
[lat,lon] = recenter(lat,lon)
[lat,lon,Z1,Z2,...,Zn] = recenter(lat,lon,Z1,Z2,...,Zn)
[...] = recenter(...,'center',centerLon)
```

说明

`[lat,lon] = recenter(lat,lon)` 重新整理一个全球的 lat，lon 网格，使其以本初子午线（零经度）为中心。常见的应用是将范围从 0 到 360 的网格转换为范围从-180 到 180 的网格。

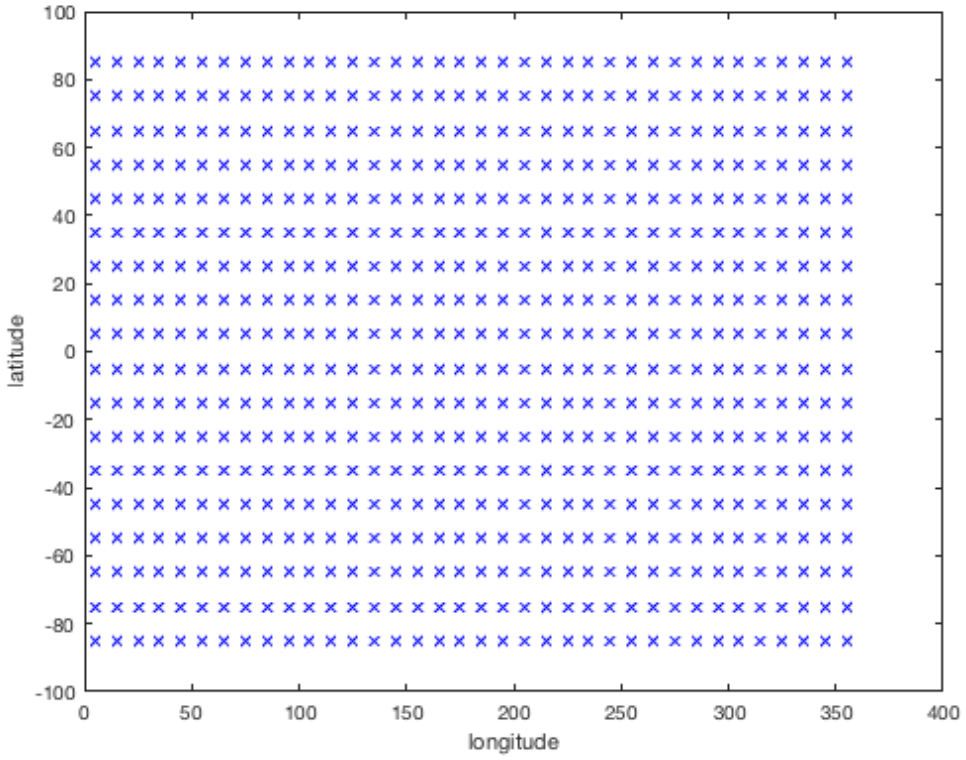
`[lat,lon,Z1,Z2,...,Zn] = recenter(lat,lon,Z1,Z2,...,Zn)` 重新整理输入网格 lat，lon 以及对应尺寸的任何其他数据集 Zn。Z 个数据集中的任何一个都可以是与纬度和经度完全相同的二维维度，也可以是三维，其前两个维度对应于纬度，经度网格，而第三个维度则可能对应于时间或深度。

`[...] = recenter(...,'center',centerLon)` 将网格化数据以任何指定的经度为中心。默认 centerlon 为 0 度。

示例 1:理论数据点

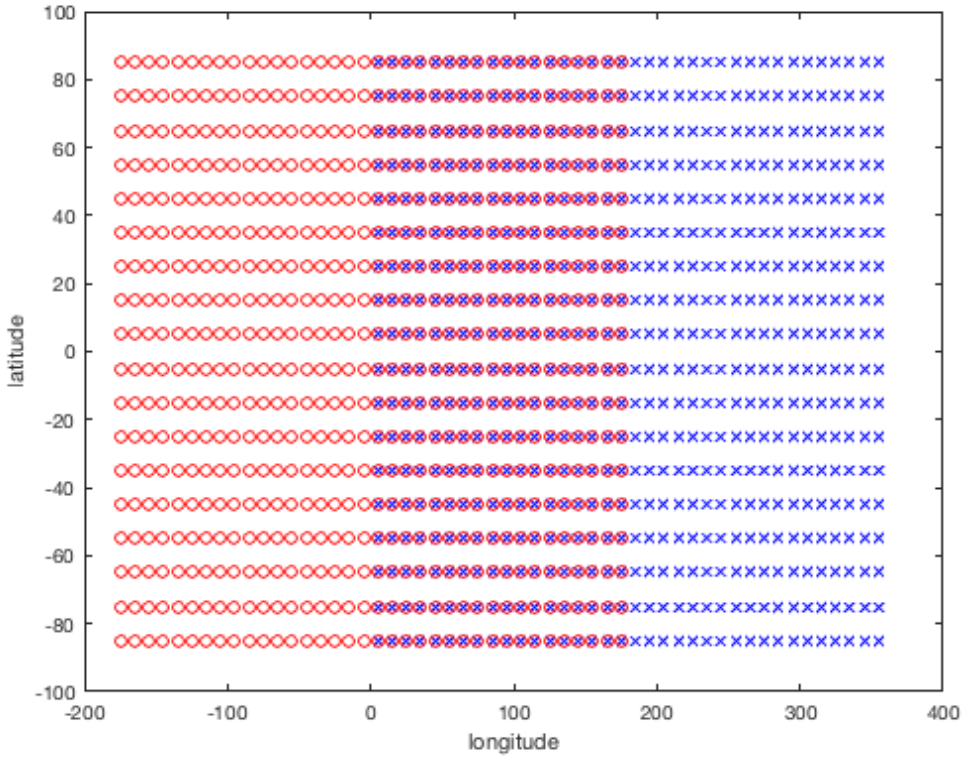
这是一个 10 度分辨率的网格，其经度值从 0 到 360，通过在 cdtgrid 来指定中心经度设置为 180：

```
[lat,lon] = cdtgrid(10,180);



plot(lon,lat,'bx')
xlabel 'longitude'
ylabel 'latitude'
```



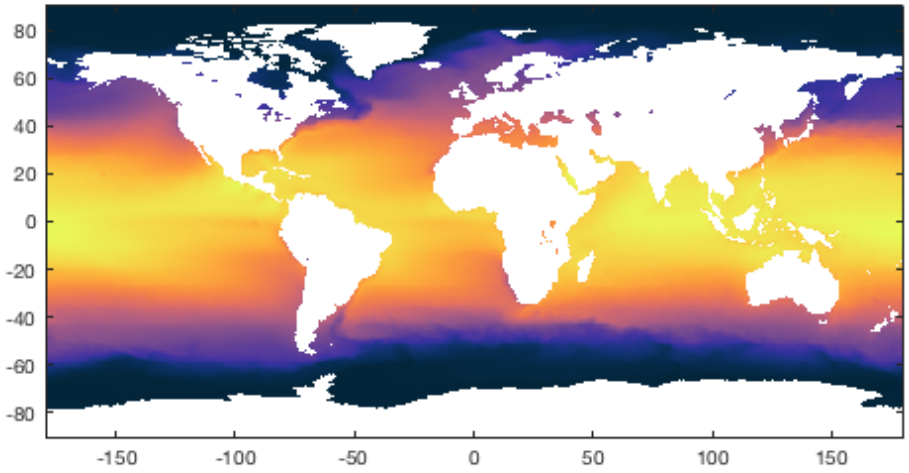
更新网格，使其中点位于本初子午线：

```
[lat2,lon2] = recenter(lat,lon);



hold on
plot(lon2,lat2,'ro')
```



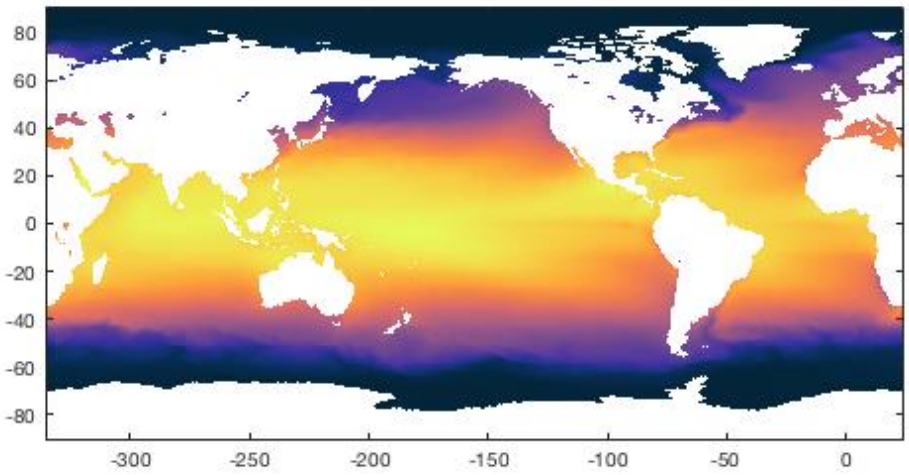
示例 2:应用于海面温度数据

```
load global_sst.mat

% 将经纬度数据转为网格格式:
[lon,lat] = meshgrid(lon,lat);

figure
imagescn(lon,lat,sst)
axis image
cmocean thermal
```



网格已经以本初子午线为中心。但是，如果您的研究重点是夏威夷，该怎么办？然后，也许您想将地图的中心放在 156 W：

```
[lat,lon,sst] = recenter(lat,lon,sst,'center',-156);

figure
imagescn(lon,lat,sst)
axis image
cmocean thermal
```



作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

cdtgrid 文档

cdtgrid 函数使用 meshgrid 轻松创建纬度和经度的全球网格。

语法

```
[lat,lon] = cdtgrid
[lat,lon] = cdtgrid(res)
[lat,lon] = cdtgrid([latres lonres])
[lat,lon] = cdtgrid(...,centerLon)
```

说明

[lat,lon] = cdtgrid 以 1 度的分辨率生成经度和纬度值的网状网格样式的全局网格。发布位于网格单元的中心，因此分辨率为 1 度的网格将具有 89.5、88.5、87.5 等纬度值。

[lat,lon] = cdtgrid(res) 指定网格分辨率 res，其中 res 是标量并指定度数。默认分辨率为 1。

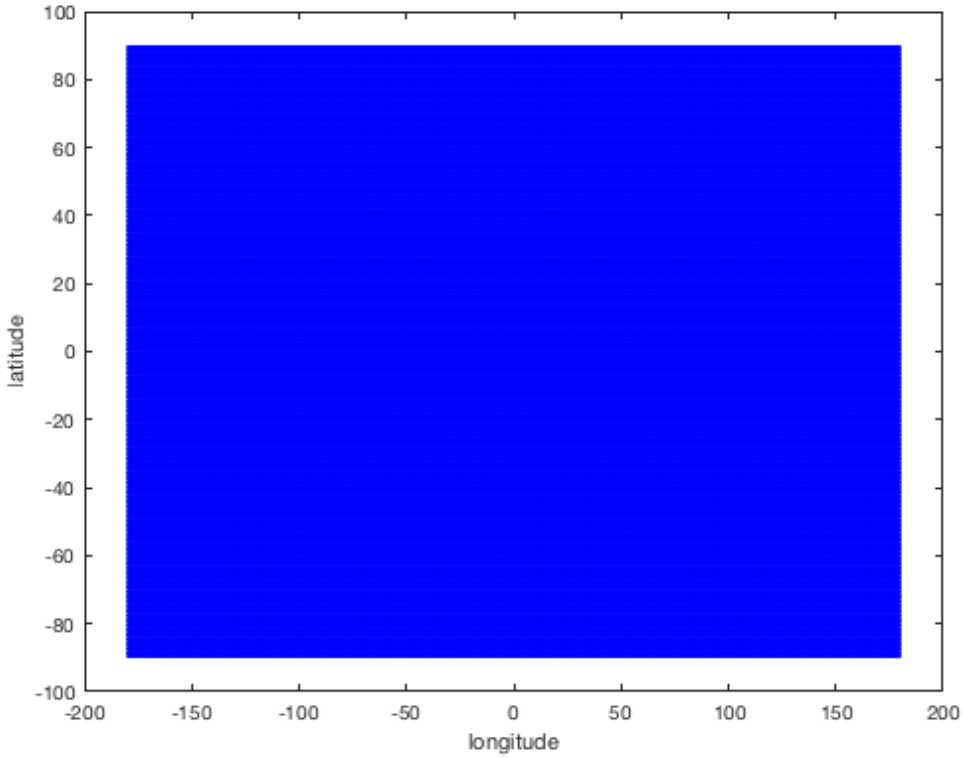
[lat,lon] = cdtgrid([latres lonres]) 如果 res 是一个由两个元素组成的数组，则第一个元素指定纬度分辨率，第二个元素指定经度分辨率。

[lat,lon] = cdtgrid(...,centerLon) 使网格在经度值 centerLon 上居中。默认的 centerLon 是本初子午线（0 度）。

示例 1:真实样本

这是一个 1 度全球网格:

```
[lat,lon] = cdtgrid;



plot(lon,lat,'b.')
xlabel('longitude')
ylabel('latitude')
```

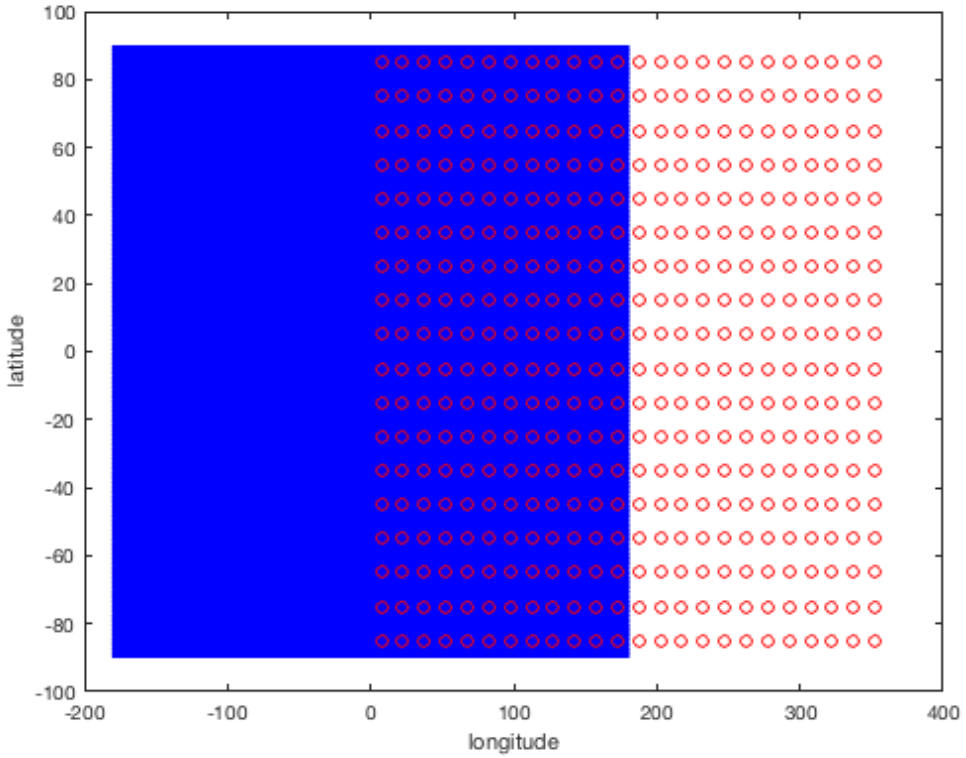


以上。它看起来像一个大的蓝色矩形，但放大后您会发现它实际上是 180 * 360 = 64,800 个蓝色点。

示例 2:更复杂

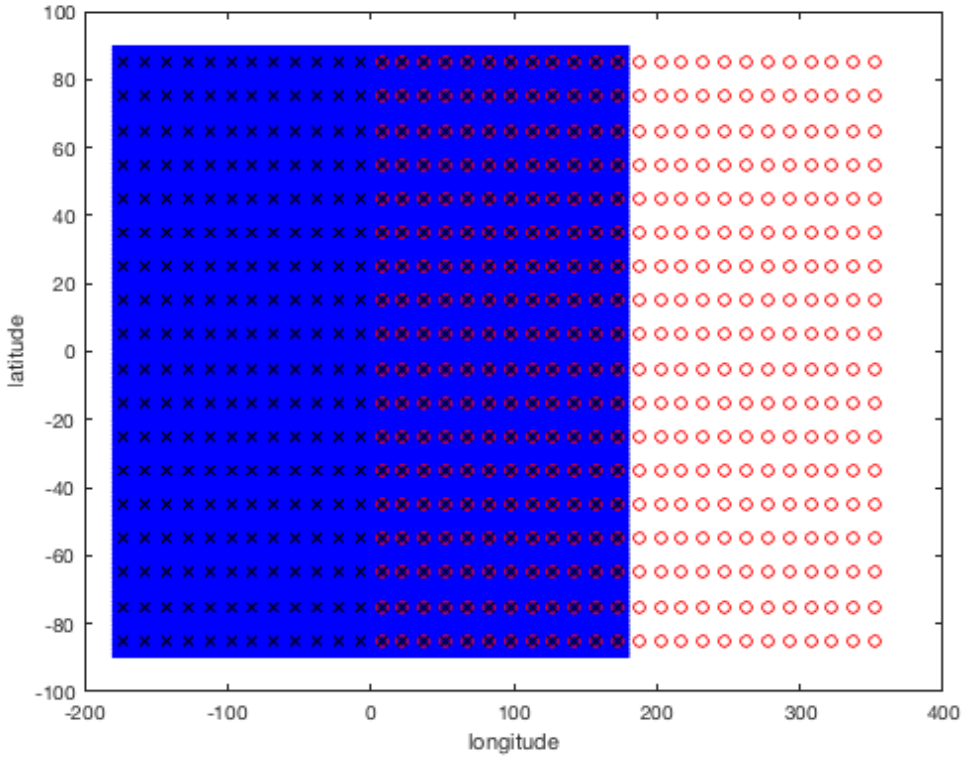
现在，以 180°E 为中心覆盖一个具有 10 度纬度分辨率和 15 度经度分辨率的网格。将新网格绘制为红色圆圈，以将其与蓝色圆点区分开：

```
[lat2,lon2] = cdtgrid([10 15],180);



hold on
plot(lon2,lat2,'ro')
```



两个栅格彼此不对齐，因为我们有意确保第二个栅格的中心位于 180°E。如果需要，可使用 `recenter` 函数将它们移回原处：

```
[lat2_rc,lon2_rc] = recenter(lat2,lon2);



plot(lon2_rc,lat2_rc,'kx')
```



作者简介

cdtgrid 函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 于 2017 年 2 月写的。

cdtdim 文档

cdtdim 给出了假设为椭圆形地球的经纬网格中每个单元的近似尺寸。此功能类似于 cdtarea。

语法

```
[dx,dy] = cdtdim(lat,lon)
[dx,dy] = cdtdim(lat,lon,'km')
```

说明

[dx,dy] = cdtdim(lat,lon) 给出了由地理坐标网格 lat，lon 给定的每个网格单元的近似尺寸（以米为单位）。输入 lat 和 lon 必须具有匹配的尺寸，就像它们是由 meshgrid 创建的一样。
[dx,dy] = cdtdim(lat,lon,'km') 给出以千米为单位的网格单元大小，而不是默认的米。

示例 1:10 度网格的单元尺寸

给定 cdtgrid 制作的 10 度全球网格。

```
[lat,lon] = cdtgrid(10);
```

每个网格单元都有以下尺寸：

```
[dx,dy] = cdtdim(lat,lon,'km');
```

如果您查看 dy 的值，您会发现它们都是相同的。这是因为纬度线始终等距分布（一个纬度约为 111 公里）。因此，对于我们的 10 度网格，所有网格单元都不足为奇

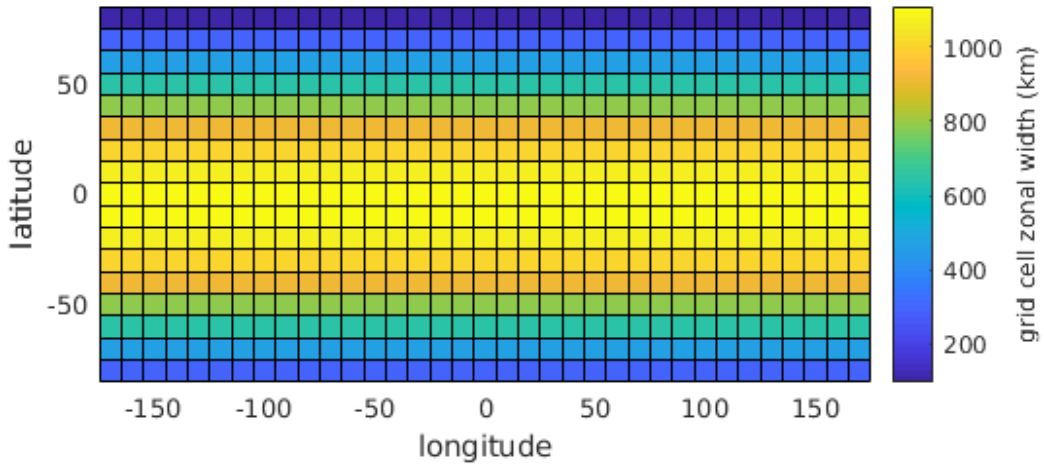
```
unique(dy(:))
```

```
ans =

   1.0e+03 *

   -1.1132

   -1.1129

   -1.1125

   -1.1120

   -1.1113

   -1.1107

   -1.1101
```

```
-1.1097

-1.1095
```

相距约 1111 公里。但是，在 x 方向上，每个网格单元的大小取决于纬度。让我们看一下网格单元的 x 尺寸在全球范围内如何变化：

```
p = pcolor(lon,lat,dx);

axis image
ylabel('latitude')
xlabel('longitude')
cb = colorbar;
ylabel(cb,'grid cell zonal width (km)')
```



如果您仔细查看上面的地图，您可能会注意到世界顶部的值似乎与世界底部的值不匹配。 这是由于 pcolor 的不幸行为所致，它会丢弃一行数据和一列数据。可以通过使用插值阴影或通过使用 imagesc 而不是 pcolor 来解决该问题，但是我在上面使用了 pcolor，因为这是包括网格线的简便方法。

示例 2:全球数据可视化网格单元大小

这是查看 cdtdim 如何计算网格单元大小的另一种方法。首先加载栅格分辨率为 0.75 度的海面温度数据集。

```
load global_sst
```

目前 lat 和 lon 是向量，因此请使用 meshgrid 将它们转换为与 sst 大小相同的矩阵：

```
[lon,lat] = meshgrid(lon,lat);
```

现在就像示例 1 中一样，计算每个网格单元的大小 dx 和 dy：

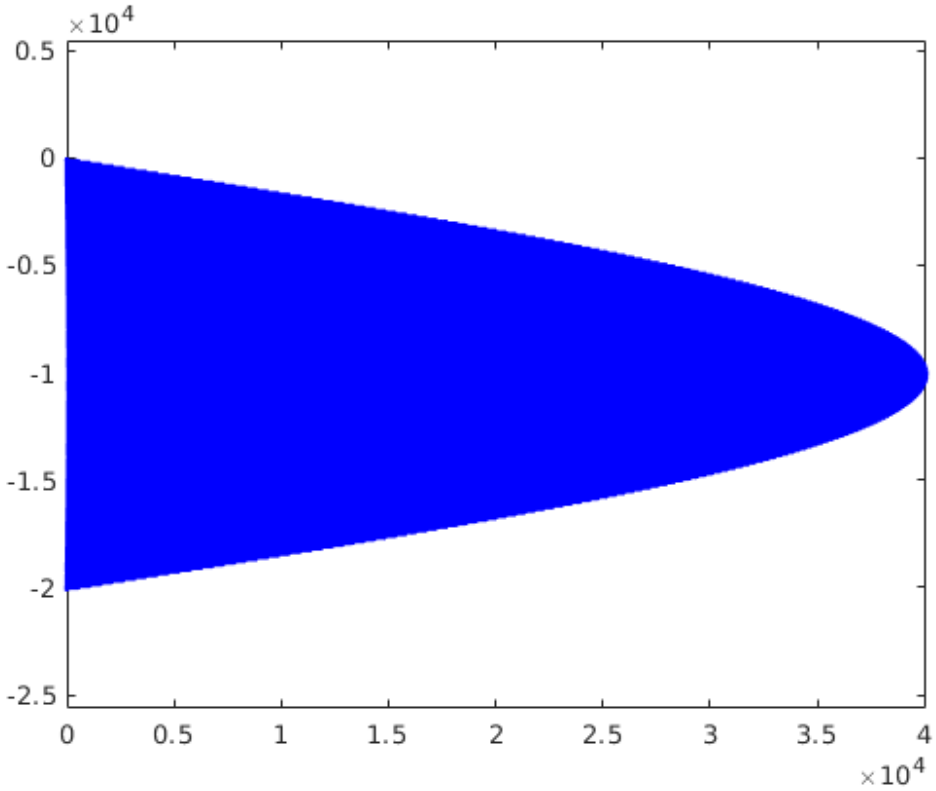
```
[dx,dy] = cdtdim(lat,lon,'km');
```

这次，我们不用在未投影的经度和纬度网格上显示网格像元大小，而是通过获取 dx 和 dy 的累加总和将 dx 和 dy 转换为有效的 x 和 y 值：

```
x = cumsum(dx,2);

y = cumsum(dy,1);
```

我们可以将 x 和 y 位置绘制为简单的从零开始的累计和，但它不会很漂亮：

```
plot(x,y,'b.')
axis equal
```

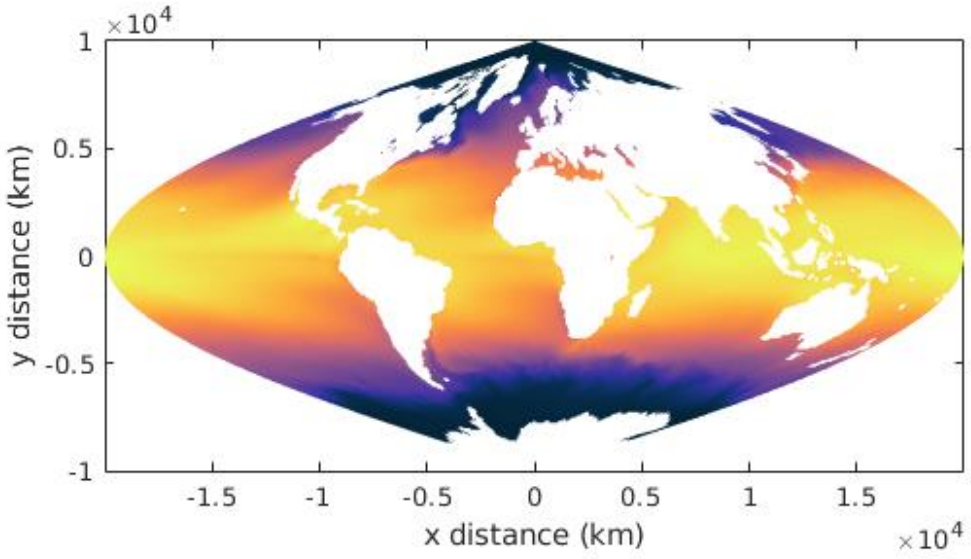


将这些值以原点为中心更有意义，这可以通过除去平均值 x 和 y 值来完成。如果您使用的是 2016b 后的 Matlab 版本，则可以简单地减去 x-mean(x,2) 来执行隐式扩展，但是使用旧版本 Matlab 的用户必须使用 bsxfun 减去：

```
x = bsxfun(@minus,x,mean(x,2));

y = bsxfun(@minus,y,mean(y,1));
```

除去均值后，我们看到 cdtdim 执行的网格单元计算和 cdtarea 类似于正弦图投影：

```
pcolor(x,y,sst)

shading interp
axis image
xlabel('x distance (km)')
ylabel('y distance (km)')
cmocean 'thermal'
```



作者简介

这个函数是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

cdtarea 文档

cdtarea 函数在假定为椭圆形的地球上，给出了经纬网格中每个单元的近似面积。该函数旨在简化大型栅格气候数据集的面积平均加权。此功能类似于 cdtdim。

语法

```
A = cdtarea(lat,lon)
A = cdtarea(lat,lon,'km2')
```

说明

`A = cdtarea(lat,lon)` 给出由 lat，lon 给出的每个网格单元的近似面积。输入 lat 和 lon 必须具有匹配的尺寸，就像它们是由 meshgrid 创建的一样。

`A = cdtarea(lat,lon,'km2')` 给出以平方公里为单位的网格单元面积。

示例

给定 cdtgrid 制作的 10 度全球网格。

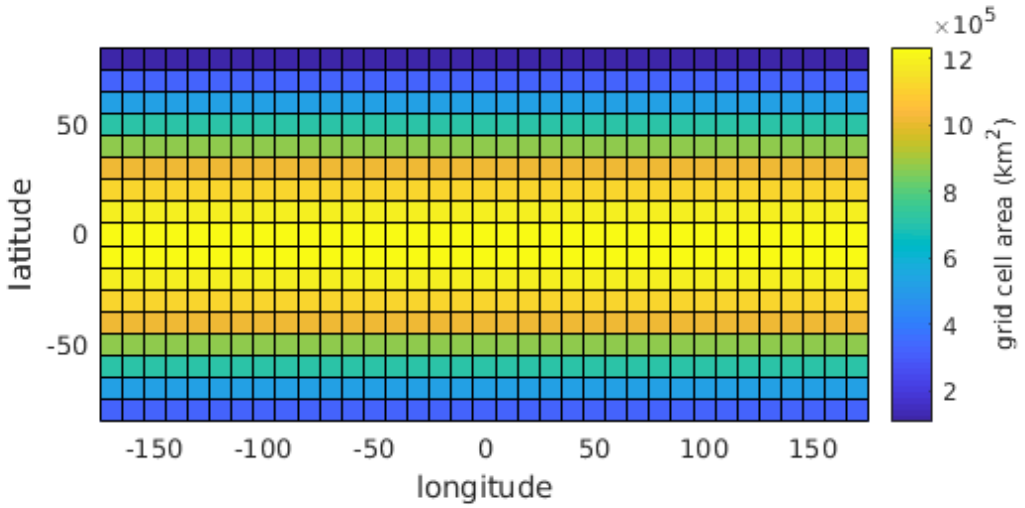
```
[lat,lon] = cdtgrid(10);
```

每个网格单元都有以下尺寸:

```
A = cdtarea(lat,lon,'km^2');
```

绘制网格单元区域:

```
p = pcolor(lon,lat,A);

axis image
ylabel('latitude')
xlabel('longitude')
cb = colorbar;
ylabel(cb,'grid cell area (km^2)')
```



正如我们所期望的，靠近赤道的网格单元较大，而靠近两极的网格单元较小。如果您仔细查看上面的地图，您可能会注意到世界顶部的值似乎与世界底部的值不匹配。这是由于 pcolor 的不幸行为所致，它会丢弃一行数据和一列数据。可以通过使用插值阴影或通过使用 imagesc 而不是 pcolor 来解决该问题，但是我在上面使用了 pcolor，因为这是包括网格线的简便方法。

作者简介

cdtarea 这个函数是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

cdtgradient 文档

cdtgradient 计算在地理坐标中等间隔网格数据的空间梯度。

语法

```
[FX,FY] = cdtgradient(lat,lon,F)
[FX,FY] = cdtgradient(lat,lon,F,'km')
```

说明

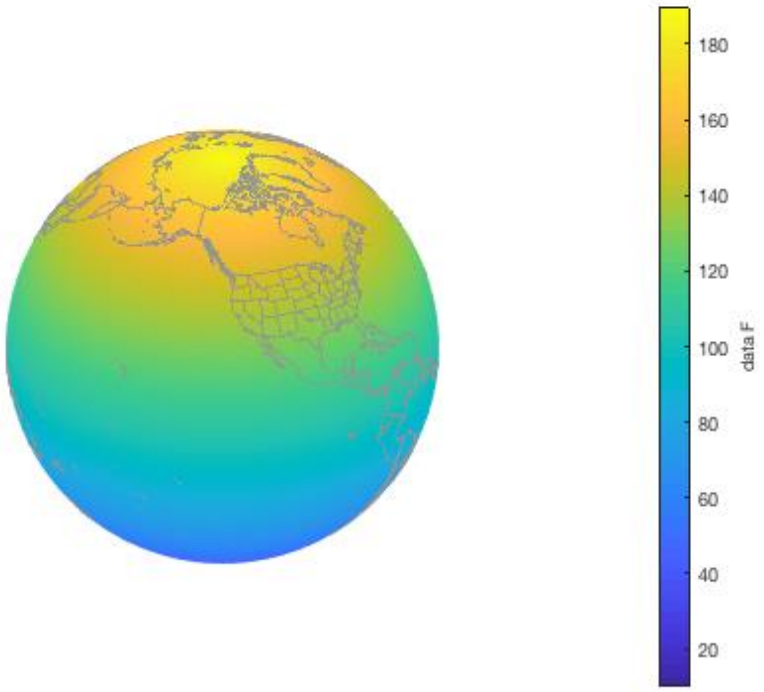
[FX,FY] = cdtgradient(lat,lon,F) 对于网格变量 F 以及相应地理坐标 lat 和 lon，cdtgradient 计算 FX，即沿地球表面每米 F 的东西向变化的空间速率和 FY，从南到北的每米 F 变化。此函数假定由 cdtdim 和 earth_radius 建模为椭圆形的地球。FX 的正值指示 F 在该网格单元中从西向东增加，而 FY 的正值指示 F 从南向北增加。F 可以是二维或三维矩阵，其前两个维度必须对应于 lat 和 lon。如果 F 为三维，则输出 FX 和 FY 也将为三维，沿着第三维的每个网格将单独计算。

[FX,FY] = cdtgradient(lat,lon,F,'km') 返回每公里而非默认米的坡度。

示例 1: 理论

这是一个包含一些变量 F 的全局网格，该变量仅是纬度的函数，并且以每纬度 1 个单位 F 的速率从南向北递增。使用 cdtgrid 创建网格，然后将 F 定义为纬度加 100 的值：

```
% 创建一个 0.25° 的网格:
[lat,lon] = cdtgrid(0.25);

% 定义 F:
F = lat + 100;

% 在地球上画 F:
figure
globepcolor(lat,lon,F)
globeborders('color',rgb('gray'))
axis tight
cb = colorbar;
ylabel(cb,'data F')
```



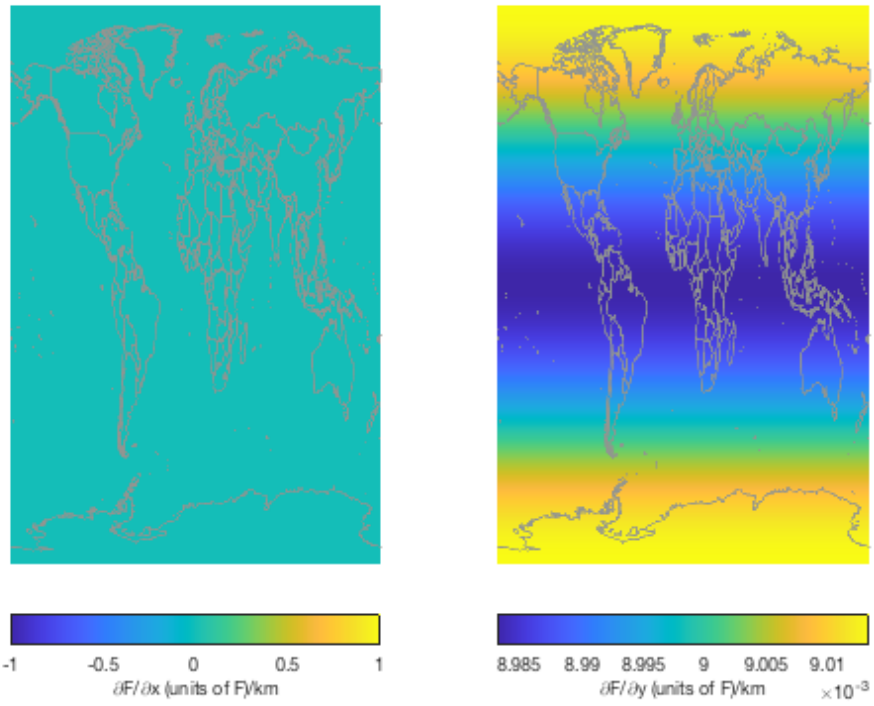
在上图中，我们看到 F 值的范围从南极附近的大约 10 到北极附近的大约 190。这是因为 F 只是每个网格单元的纬度加 100。

有了这个数据集，我们知道在任何给定的纬度下，所有经度的 F 都是相同的。这意味着 Fx 在地球上的任何地方都应该为零，因为 F 永远不会从西向东变化。

但是，F 确实会从南向北变化，每度纬度的变化率为 1 单位 F。知道纬度相距约 111 公里（从米的原始定义（即从赤道到北极的距离为 1000 万米））是很容易的。因此，如果每个纬度之间的距离为 111 km，并且每个纬度上的 F 增加 1 个单位，则地球上的任何地方 Fy 都应为 1/111 = 0.009。让我们来看看：

```
[Fx,Fy] = cdtgradient(lat,lon,F,'km');

figure
subplot(1,2,1)
pcolor(lon,lat,Fx)
borders('countries','color',rgb('gray'))
shading interp
axis off % 移除刻度
cb1=colorbar('location','southoutside');
xlabel(cb1,'\partialF/\partialx (units of F)/km')

subplot(1,2,2)
pcolor(lon,lat,Fy)
shading interp
borders('countries','color',rgb('gray'))
```

```
axis off % 移除刻度
cb2=colorbar('location','southoutside');
xlabel(cb2,'\partialF/\partialy (units of F)/km')
```



以上，我们看到我们的预测进展顺利（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）。正如我们所预测的，Fx 在世界各地都为零，而 Fy 的值集中在我们的预测值 9e-3 上。但是，Fy 会有一些纬度变化，因为地球不是一个完美的球体，而是一个椭圆形，其纬度并非完美地隔开 111,111 m，正如您最初认为米的定义那样。

示例 2: 真实

对于此示例，我们将使用 CDT 随附的示例表面气压数据。首先加载它：

```
filename = 'ERA_Interim_2017.nc';
sp = ncread(filename,'sp'); %表面压强
lat = double(ncread(filename,'latitude'));
lon = double(ncread(filename,'longitude'));
[Lat,Lon] = meshgrid(lat,lon);
```

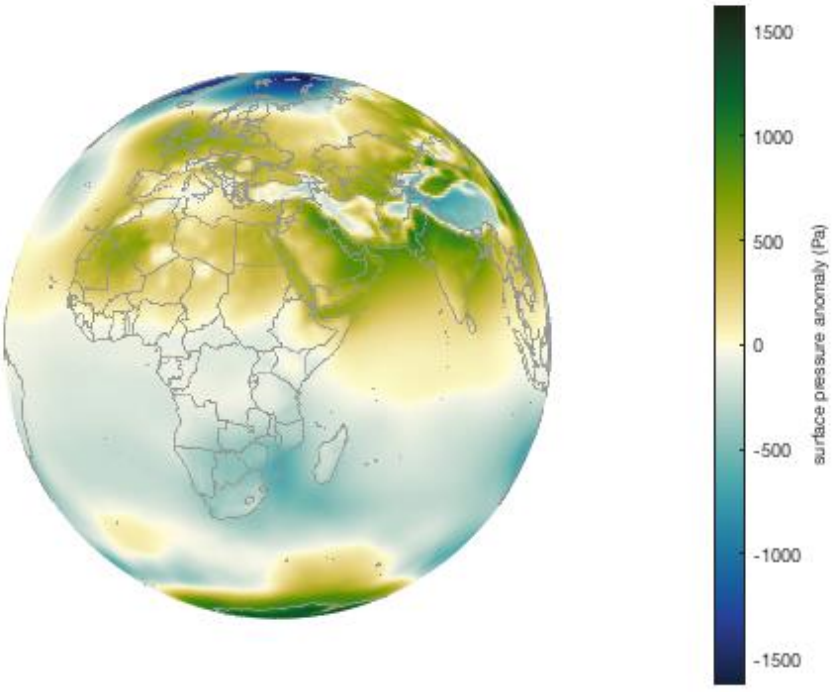
绝对表面压强并不是特别有趣，因为它主要跟踪表面高度。但是表面气压*异常*要有趣得多，因为它们使我们了解了系统在给定时间点的失衡程度。

为了方便起见，让我们将 2017 年 1 月的表面气压距平计算为该月的表面气压场，减去 2017 年全年的平均表面气压

```
% 表面气压“距平”：
spa = sp(:,:,1)-mean(sp,3);
```

这是我们要分析的表面气压距平：

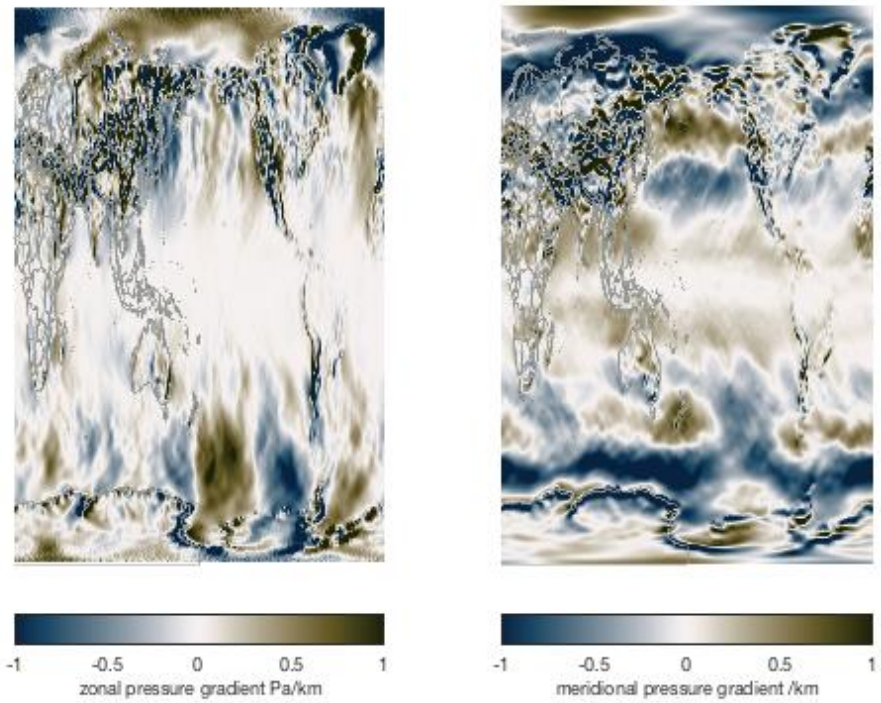
```
figure

globepcolor(Lat,Lon,spa);

globeborders('color',rgb('gray'))
axis tight
cmocean('delta','pivot') % 设置颜色图
cb = colorbar;
ylabel(cb,'surface pressure anomaly (Pa)')
view(125,5) % 设置视角
```



像这样计算 Pa/km 的纬向和经向表面气压梯度（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
[Sx,Sy] = cdtgradient(Lat,Lon,spa,'km');

figure
subplot(1,2,1)
pcolor(Lon,Lat,Sx)
borders('countries','color',rgb('gray'))
shading interp
axis off % 移除刻度
cb1=colorbar('location','southoutside');
```

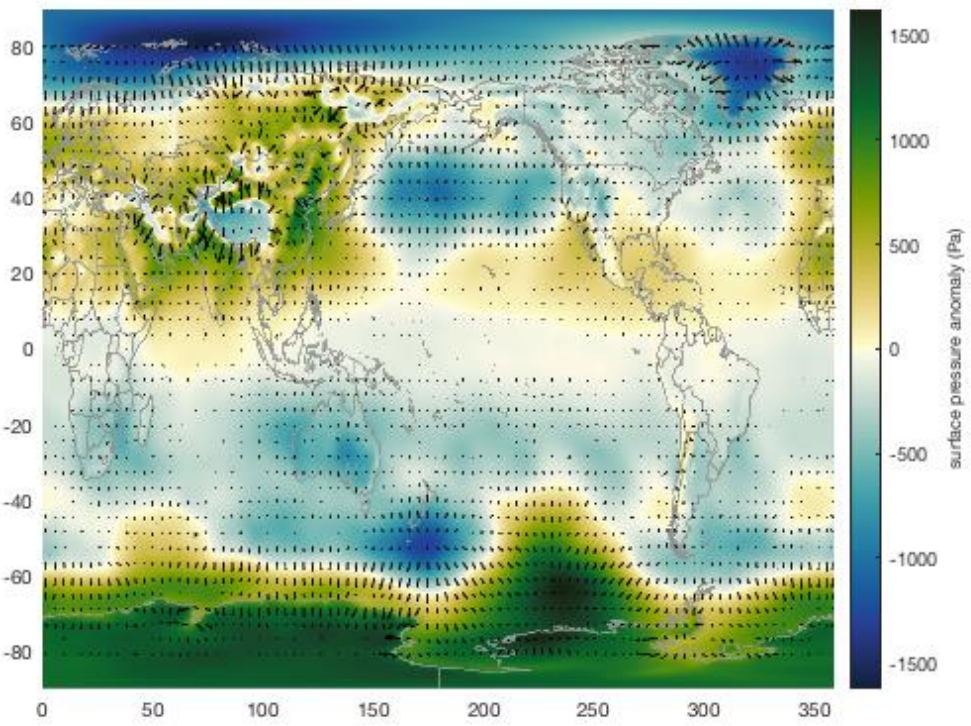
```
xlabel(cb1,'zonal pressure gradient Pa/km')
caxis([-1 1])
cmocean diff

subplot(1,2,2)
pcolor(Lon,Lat,Sy)
shading interp
borders('countries','color',rgb('gray'))
axis off %移除刻度
cb2=colorbar('location','southoutside');
xlabel(cb2,'meridional pressure gradient /km')
caxis([-1 1])
cmocean diff
```



上图可能难以解释，因此我们将采用另一种方法。不用将 Sx 和 Sy 绘制为颜色，而是将它们作为矢量绘制在表面气压距平贴图的顶部：（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
figure

pcolor(Lon,Lat,spa)

shading interp
```

```
hold on
borders('countries','color',rgb('gray'),'center',180)
cmocean('delta','pivot') % 设置颜色图
cb = colorbar;
ylabel(cb,'surface pressure anomaly (Pa)')

quiversc(Lon,Lat,Sx,Sy,'k','density',100)
```



在上面的地图中（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！），我们看到渐变矢量始终从低值指向高值。自然，那是气压梯度力的相反方向。

作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

cdtdivergence 文档

cdtdivergence 计算椭圆形地球表面上的网格矢量的散度。也可参见:cdtgradient,cdtcurl 和 ekman。

语法

```
D = cdtdivergence(lat,lon,U,V)
```

说明

D = cdtdivergence(lat,lon,U,V) 使用 cdtdim 估计 lat，lon 网格中每个网格单元的尺寸，然后计算网格矢量 U，V 的散度。D 的单位是 U 和 V 的单位除以米。

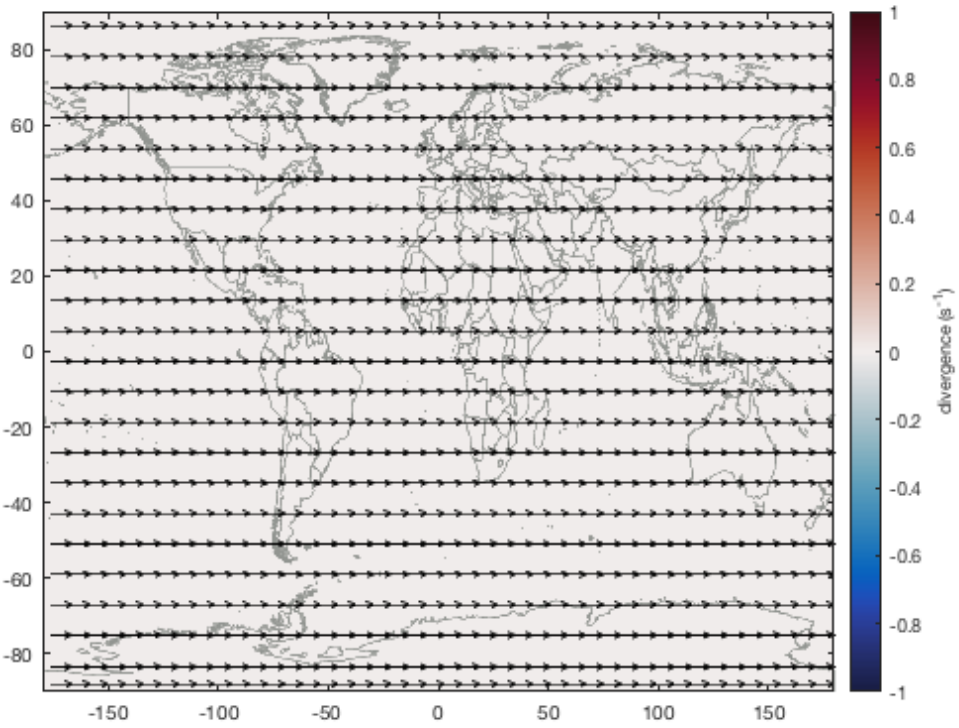
示例 1：理论

这是一个简单的纯纬向风场，全球各地的均匀风速为 1 m/s。使用 cdtgrid 制作一个四分之一度的网格，然后定义风场：

```
[lat,lon] = cdtgrid(0.25); % 0.25° 网格
u = ones(size(lat));       %纯纬向风
v = zeros(size(lat));      %经向为 0.
```

这是均匀速度的纯纬向风的风向矢量和散度图：（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
D = cdtdivergence(lat,lon,u,v);



figure

imagescn(lon,lat,D)

hold on
borders('countries','color',rgb('gray'))
quiversc(lon,lat,u,v,'k')
cb = colorbar;
ylabel(cb,'divergence (s^{-1})')
cmocean('balance','pivot')
```

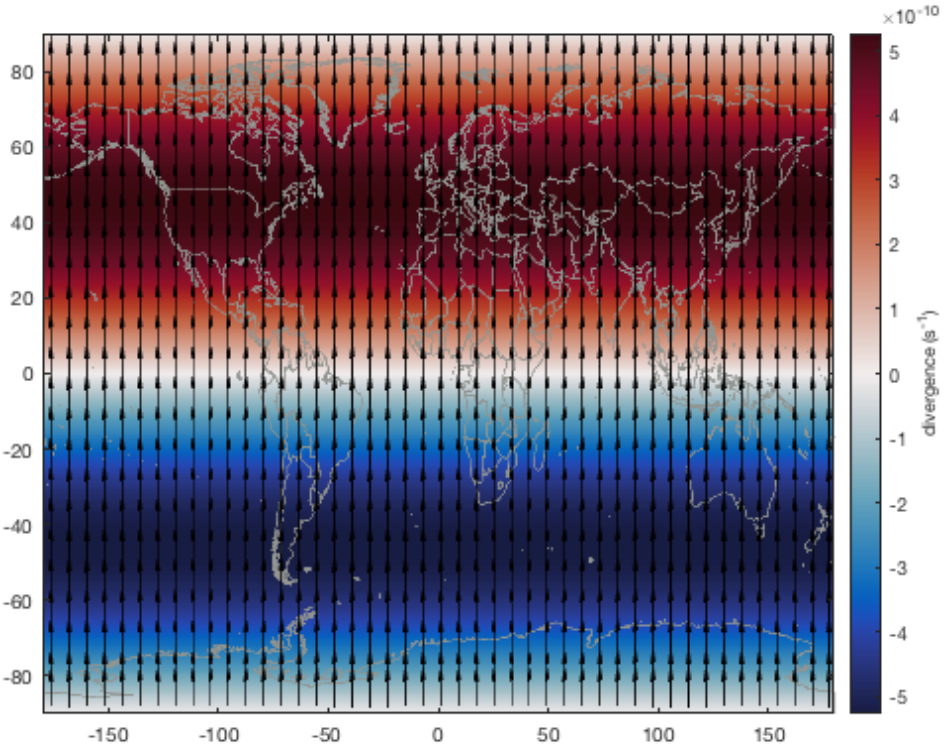


在上图中，我们看到各处的散度为零。那是因为当一束空气从一个栅格单元向东移动到下一个栅格单元时，风不会改变速度或方向，并且相邻栅格单元的大小与它刚来自的栅格单元的大小完全相同。一切保持完全相同，因此没有散度。

纯粹的子午风呢？我们可以期望全球各地出现相同的零散度吗？

从上方使用相同的四分之一纬度经度网格，在各处定义均匀速度的纯子午风：（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
u = zeros(size(lat)); %纬向没风
v = ones(size(lat));  % 到处都是 1m/s 北风

% 计算散度:
D = cdtdivergence(lat,lon,u,v);

figure
imagescn(lon,lat,D)
hold on
borders('countries','color',rgb('gray'))
quiversc(lon,lat,u,v,'k')
cb = colorbar;
ylabel(cb,'divergence (s^{-1})')
cmocean('balance','pivot')
```

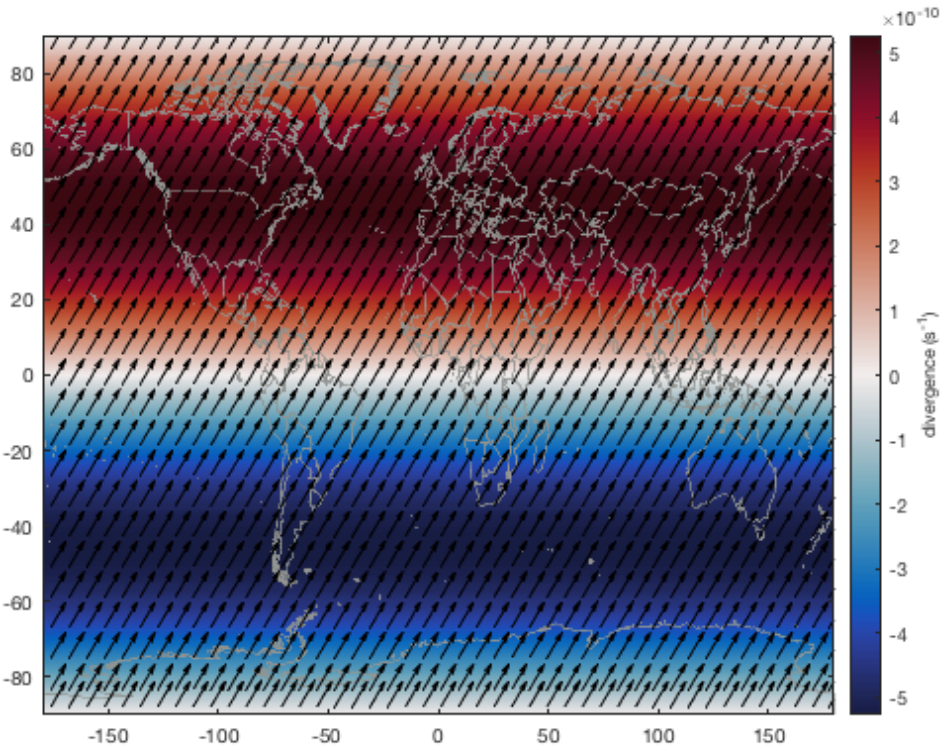


以上，我们得到的并不是到处都是零散度。相反，北半球似乎存在辐散，而南半球则存在辐合。那是因为地球不是一个完美的球体，而是一个椭球体，其纬线彼此之间的距离几乎相等，但并非相等。玩转 earth_radius 函数，您会发现在从南极到赤道的旅行中，纬度线越来越靠近。继续前进，从赤道到北极，纬线将再次散开。检查上方颜色栏上的比例，您会看到，尽管纬度间距确实会影响散度，但它在这里的作用仍然很小。

综合以上两个示例，如果各地的风在纬向和经纬方向均匀，该怎么办？（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
u = ones(size(lat));

v = ones(size(lat));



D = cdtdivergence(lat,lon,u,v);



figure

imagescn(lon,lat,D)

hold on
borders('countries','color',rgb('gray'))
```

```
quiversc(lon,lat,u,v,'k')
cb = colorbar;
ylabel(cb,'divergence (s^{-1})')
cmocean('balance','pivot')
```

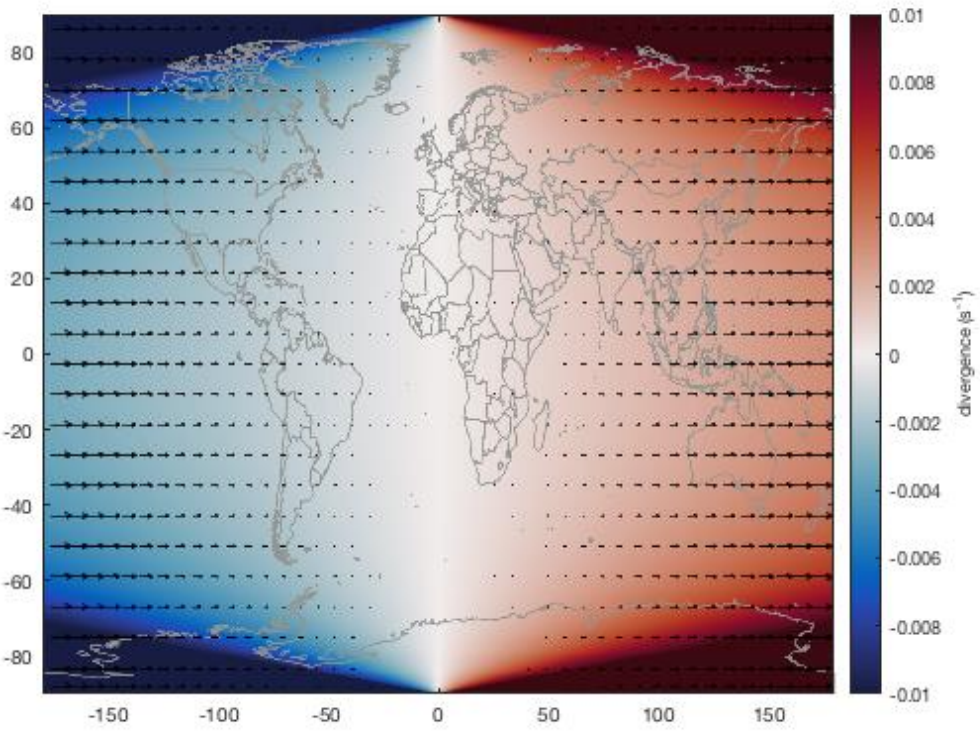


从上面我们可以看到，将零纬向散度添加到少量子午向散度上，所产生的结果与仅查看子午向散度相同。我想这并不奇怪。

现在考虑风速随经度变化的情况。盘他（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
u = lon.^2;

v = zeros(size(lat));



D = cdtdivergence(lat,lon,u,v);



figure

imagescn(lon,lat,D)

hold on
```

```
borders('countries','color',rgb('gray'))
quiversc(lon,lat,u,v,'k')
cb = colorbar;
ylabel(cb,'divergence (s^{-1})')
cmocean('balance','pivot')

caxis([-1 1]*1e-2)
```



以上，在西半球，风从国际日期变更线上的非常大的风辐合到本初子午线的零速度。在本初子午线，风又开始加速，随着风开始有效地将自身拉开而使其辐散。
还要注意，在上面的图中，会聚点和趋近因子的强度在极点附近增强，因为在高纬度下，从一个网格单元到下一个网格单元的风加速度发生的距离较小。如果您对网格单元间距在世界范围内如何变化感到好奇，请使用 cdtdim 函数探索效果。

示例 2：真实情况和 ITCZ

对于此示例，加载 CDT 附带的一些全球地面风数据。加载 10 米风速 u10 和 v10，并通过简单获取 2017 年的平均地面风来使其保持简单：

```
filename = 'ERA_Interim_2017.nc';
u10 = mean(ncread(filename,'u10'),3);
v10 = mean(ncread(filename,'v10'),3);
lat = double(ncread(filename,'latitude'));
lon = double(ncread(filename,'longitude'));
[Lat,Lon] = meshgrid(lat,lon);
```

原始数据位于从 0 到 360 经度的网格上。我宁愿将本初子午线放在地图的中间，因此让我们用 recenter 重新整理网格。此步骤不是必需的，但它是首选项：

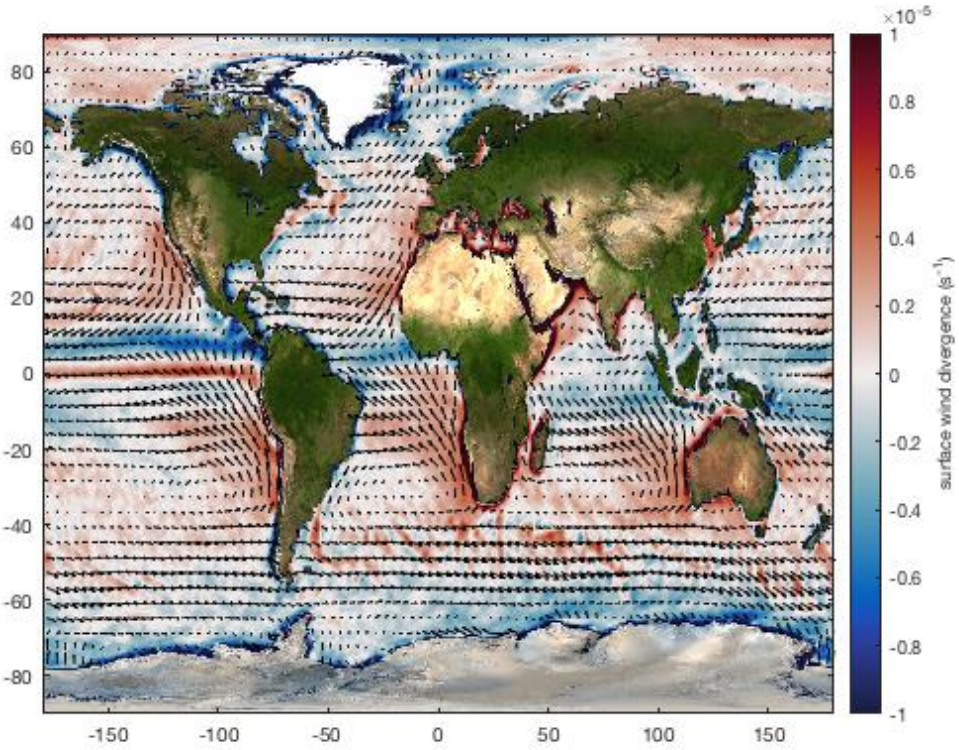
```
[Lat,Lon,u10,v10] = recenter(Lat,Lon,u10,v10);
```

使用真实数据计算风散度与使用示例 1 中创建的合成数据一样简单。出于美学目的，我们在这里所做的唯一调整是使用 island 函数掩盖陆地网格单元：

```
% 计算风的散度:
D = cdtdivergence(Lat,Lon,u10,v10);

% 屏蔽陆地:
land = island(Lat,Lon);
D(land) = NaN;
u10(land) = NaN;
v10(land) = NaN;
```

现在，我们可以像示例 1 中一样绘制风矢量及其散度，但是这次我们 earthimage 底图开始：

```
figure

earthimage;

hold on
pcolor(Lon,Lat,D)
shading interp
hold on
quiversc(Lon,Lat,u10,v10,'k','density',100)
cb = colorbar;
ylabel(cb,'surface wind divergence (s^{-1})')
caxis([-1 1]*1e-5)
cmocean balance
```



上面的地图显示了 10 米的风向矢量如何在世界范围内汇聚和散布。它显示了东太平洋赤道以北的蓝色会聚区域，与热带辐合带（ITCZ）吻合，但是 ITCZ 是季节性迁徙的，因此，这一全年的平均风场可能会产生在任何给定时间降低 ITCZ 的真实强度。

作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

cdtcurl 文档

cdtcurl 计算椭圆形地球上网格矢量的曲率的 z 分量。
也可参见: cdtgradient, cdtdivergence 和 ekman。

语法

```
Cz = cdtcurl(lat,lon,U,V)
```

说明

Cz = cdtcurl(lat,lon,U,V) 使用 cdtdim 估计 lat，lon 网格中每个网格单元的尺寸，然后计算网格化矢量 U，V 的曲率。.

示例

加载 10 米风 u 和 v 的样本。请注意，当加载风数据时，本例仅取平均风。

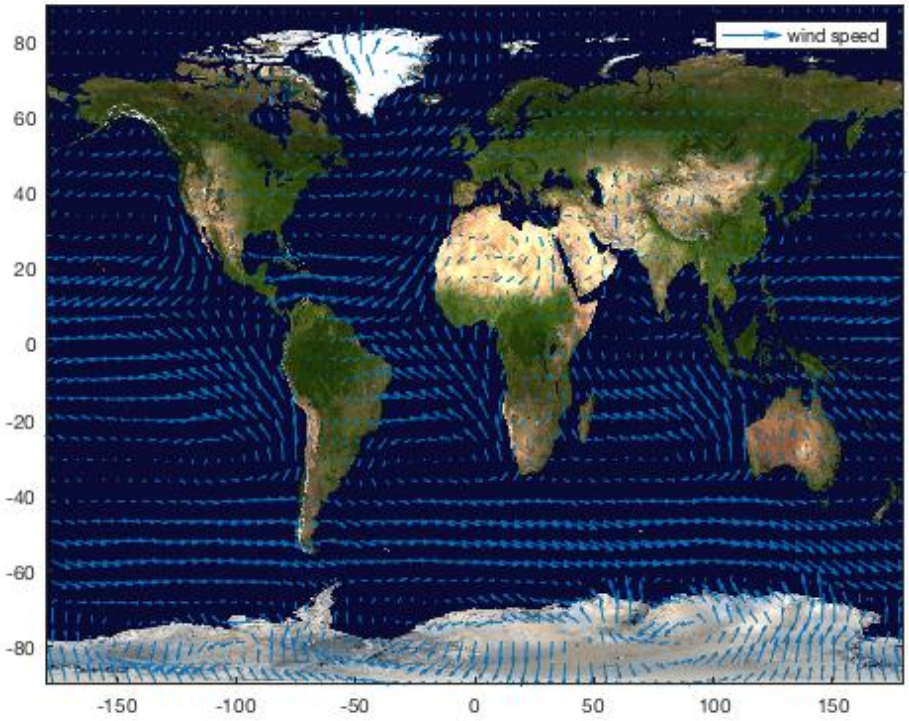
```
filename = 'ERA_Interim_2017.nc';
u = mean(ncread(filename,'u10'),3);
v = mean(ncread(filename,'v10'),3);
lat = double(ncread(filename,'latitude'));
lon = double(ncread(filename,'longitude'));
[Lat,Lon] = meshgrid(lat,lon);
```

原始数据位于从 0 到 360 经度的网格上。我宁愿将本初子午线放在地图的中间，因此让我们用 recenter 重新整理网格。此步骤不是必需的，但它是首选项：

```
[Lat,Lon,u,v] = recenter(Lat,Lon,u,v);
```

这是全球地图上的风：

```
figure

earthimage

hold on

q = quiversc(Lon,Lat,u,v,'density',75);

legend(q,'wind speed')
```



为了说明 cdtcurl 函数的用法，我们可以重新创建 Talley 等人的著作《描述性物理海洋学》 Descriptive Physical Oceanography 第六版的图 Figure S11.03a 。

首先使用 windstress 函数将 10 米风速矢量转换为 N / m ^ 2 的风应力：

```
[Taux,Tauy] = windstress(u,v);
```

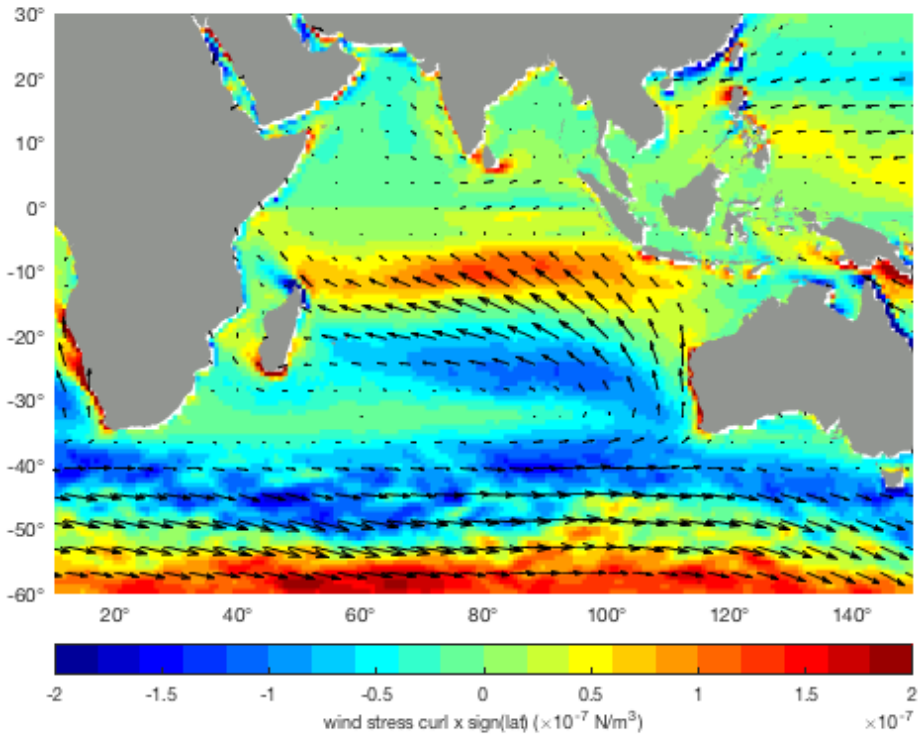
现在使用 cdtcurl 获取全局风应力曲率：

```
C = cdtcurl(Lat,Lon,Taux,Tauy);

island % 掩盖陆地:
land = island(Lat,Lon);
C(land) = NaN;
Taux(land) = NaN;
Tauy(land) = NaN;

%将南半球的曲率乘以负 1:
C = C.*sign(Lat);
```

现在我们准备重新创建 Talley 等人的地图，将风应力与风应力曲率相关联：

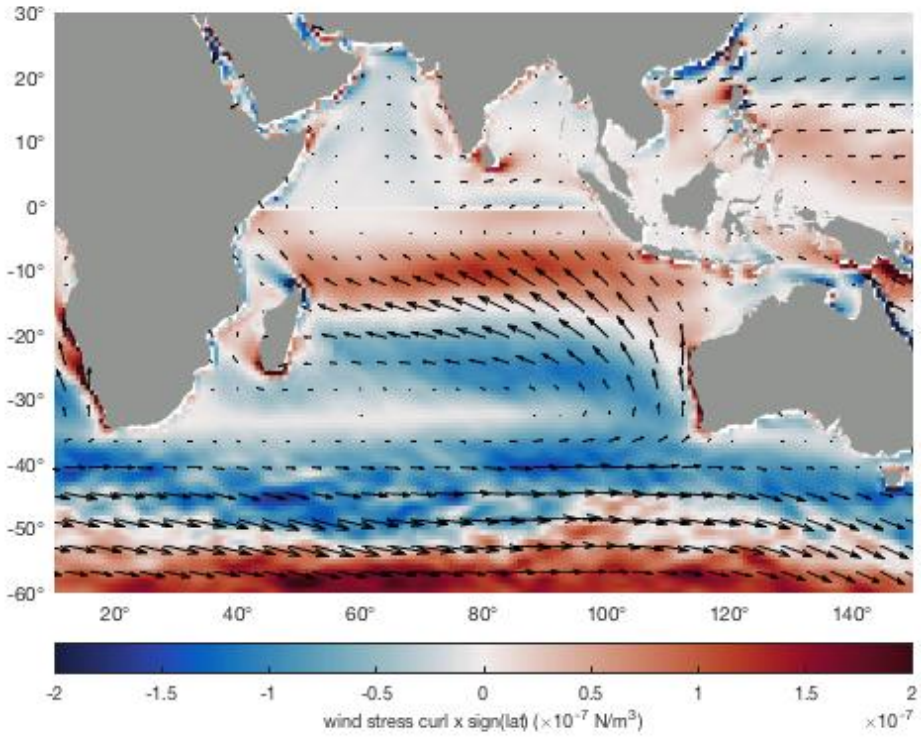
```
figure

pcolor(Lon,Lat,C)
```

```
shading flat
hold on

% 覆盖大陆:
borders('countries','facecolor',rgb('gray'),'edgecolor','none');

% 覆盖风应力矩阵:
quiversc(Lon,Lat,Taux,Tauy,'k','density',100)

% Jet 绝不是一个好主意，但我们会这样做来模仿成图:
colormap(jet(20))

% 设置色轴限制并制作一个色条:
caxis([-2 2]*1e-7)
cb = colorbar('location','southoutside');
xlabel(cb,'wind stress curl x sign(lat) (\times10^{-7} N/m^3)')

% 设置地图范围:
axis([10 150 -60 30])

% 将标签单位设为度:
xtickformat('degrees')
ytickformat('degrees')
```



如果您好奇的话，这是同一张图，带有来自 cmocean (Thyng et al., 2016)的更合适的，发散的颜色图（Thyng 等人，2016 年）

```
cmocean bal
```



作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

geomask 文档

geomask 确定地理位置是否在给定的地理区域内。

语法

```
mask = geomask(lat,lon,latv,lonv)
mask = geomask(lat,lon,latv,lonv,'inclusive')
[mask,coords] = geomask(...)
```

说明

mask = geomask(lat,lon,latv,lonv) 返回 lat 和 lon 大小的掩膜，对于 latv，lonv 给定范围内的所有点，该掩膜都是 true。

- 标量 latv，lonv：如果 latv，lonv 是标量值，则仅对最接近 latv，lonv 的像素，输出 mask 为 true。
- 两元素数组：如果 latv，lonv 是两元素数组（例如[40 50]，[110 120]），则对于以 latv，lonv 为边界的地理四边形中的所有 lat，lon 值，输出 mask 均为 true。
- 由 latv，lonv 定义的多边形：如果 latv，lonv 包含两个以上的元素，则在由 latv，lonv 定义的多边形内，所有 lat，lon 的元素的输出 mask 将为 true。
- 单元格格式的多边形：如果 latv，lonv 是元胞数组（对于 shapefile 中的多个区域很常见），则输出 mask 对于 latv，lonv 中任何多边形内的所有元素都是 true。

mask = geomask(lat,lon,latv,lonv,'inclusive') 包括位于 latv，lonv 定义的边界上的 lat，lon 点。

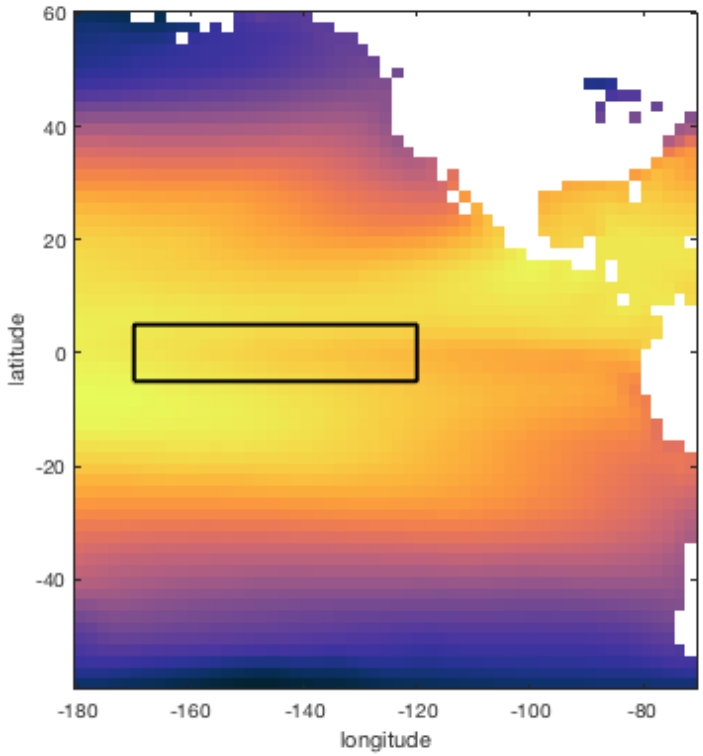
[mask,coords] = geomask(...) 如果 latv，lonv 是标量，则可选的 coords 输出是一个包含掩膜中像素坐标的结构。coords 结构包括为 lat，lon 网格的行和列的 coords.row 和 coords.col，以及为输出网格像元的地理位置的 coords.lat 和 coords.lon。

示例 1: 地理四边形

有时，您对具有地理四边形的值感兴趣。Nino 3.4 框就是这样一个四边形，定义为（5°S 至 5°N，170°W 至 120°W）。这是一些海面温度数据样本

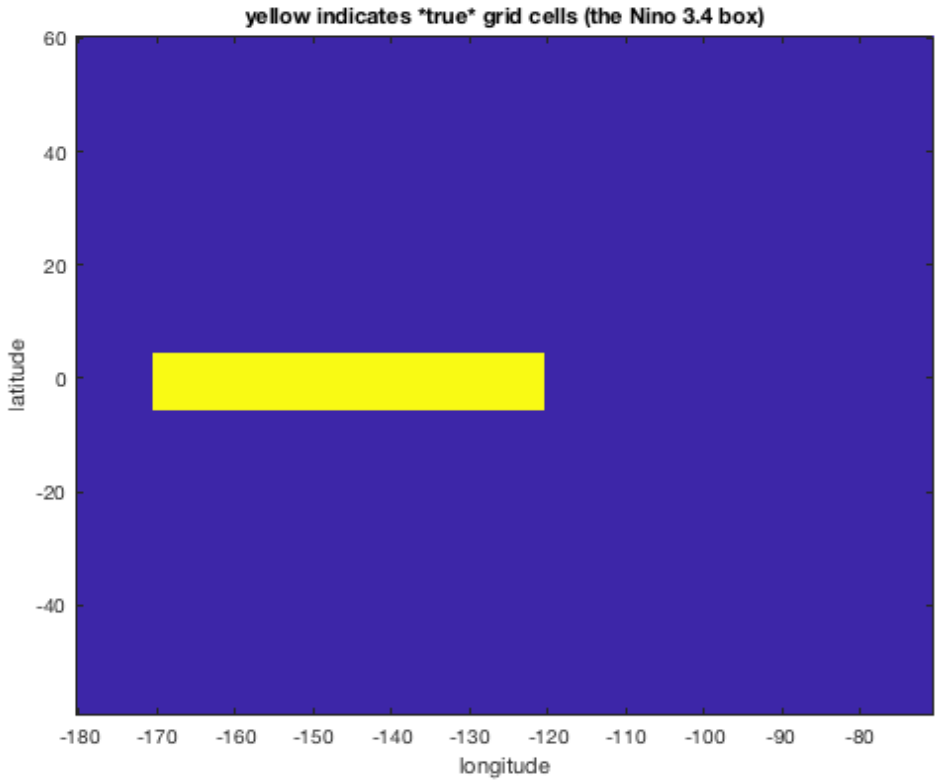
```
% 载入样本数据:
load pacific_sst.mat

% 画出样本数据:
imagescn(lon,lat,mean(sst,3))
axis xy image
cmocean thermal
xlabel 'longitude'
ylabel 'latitude'

% 定义 Nino 3.4 框:
latv = [-5 -5 5 5 -5];
lonv = [-170 -120 -120 -170 -170];

% 画出 Nino 3.4 框:
```

```
hold on
plot(lonv,latv,'k-','linewidth',2)
```



对于简单的地理四边形，获得掩膜的最简单方法是定义边界框的边界（如果您有 Matlab 的 Mapping Toolbox，则类似于 ingeoquad ）。
首先将纬度、经度数组转换为网格，然后通过指定限制感兴趣区域来获得掩膜：

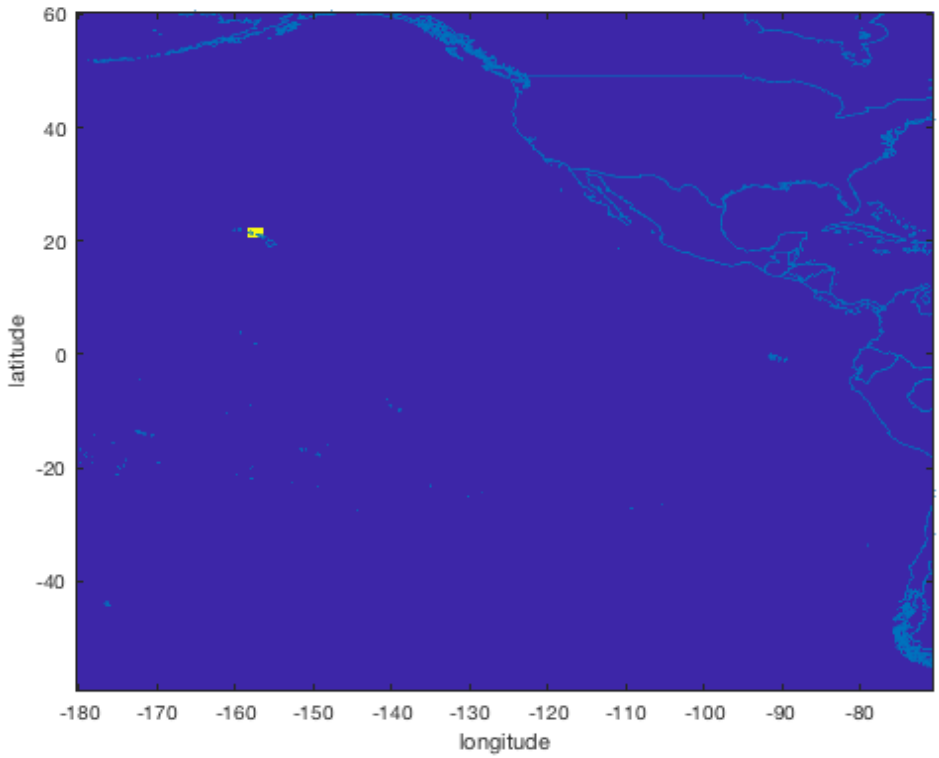
```
% 将经纬度转化为网格:
[Lon,Lat] = meshgrid(lon,lat);

% 在地理四边形内获取一个元胞数组掩膜:
mask = geomask(Lat,Lon,[-5 5],[-170 -120]);

% 画掩膜:
figure
imagescn(lon,lat,mask)
xlabel 'longitude'
ylabel 'latitude'
title 'yellow indicates *true* grid cells (the Nino 3.4 box)'
```



示例 2:最近的单元网格

有时您需要单个兴趣点附近的时间序列。例如，您可能想绘制檀香山（21.3 N，157.8 W）附近的 SST 时间序列。让我们找到最靠近檀香山的单元网格：

```
[honolulu,coords] = geomask(Lat,Lon,21.3,-157.8);



figure

imagescn(lon,lat,honolulu)

xlabel 'longitude'
ylabel 'latitude'
hold on
borders %为背景绘制国界
```



单像素蒙版可能用途有限，但是对于此类蒙版，geomask 还提供了最近像素的行、列和地理位置，请看：

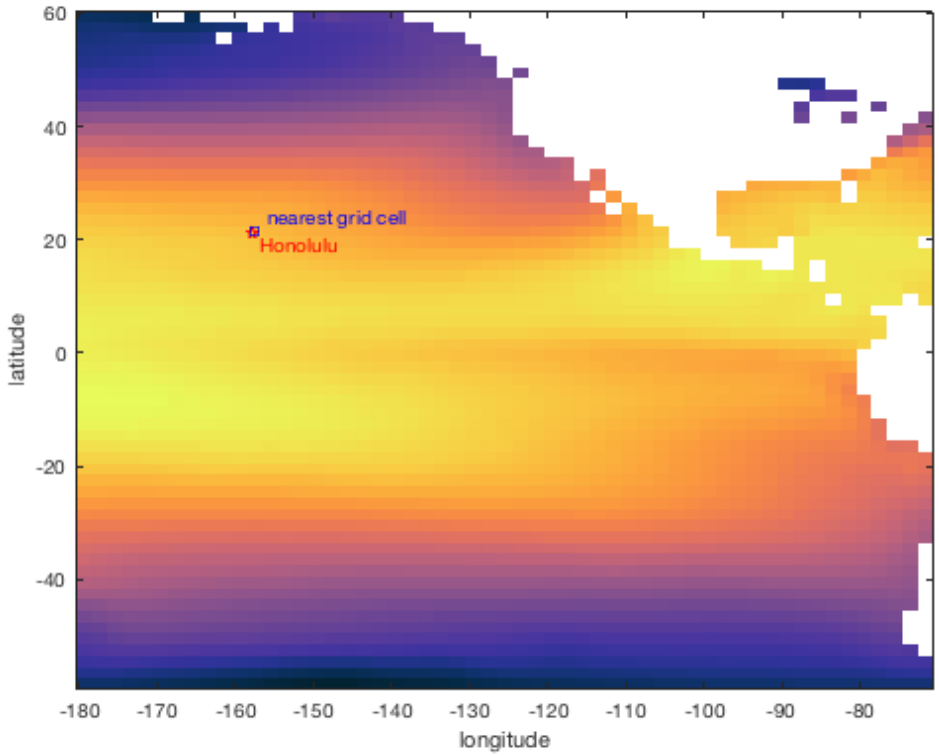
```
coords
```

```
coords =

  struct with fields:


    row: 20

    col: 12

    lat: 21.5000

    lon: -157.5000
```

为了证明这一点，下面我们绘制了平均 SST 贴图，在我们所需的檀香山位置上有一个红色的星星，在最靠近檀香山的网格上有一个蓝色正方形：

```
% 绘制平均 SST 背景:
figure
imagescn(lon,lat,mean(sst,3))
```

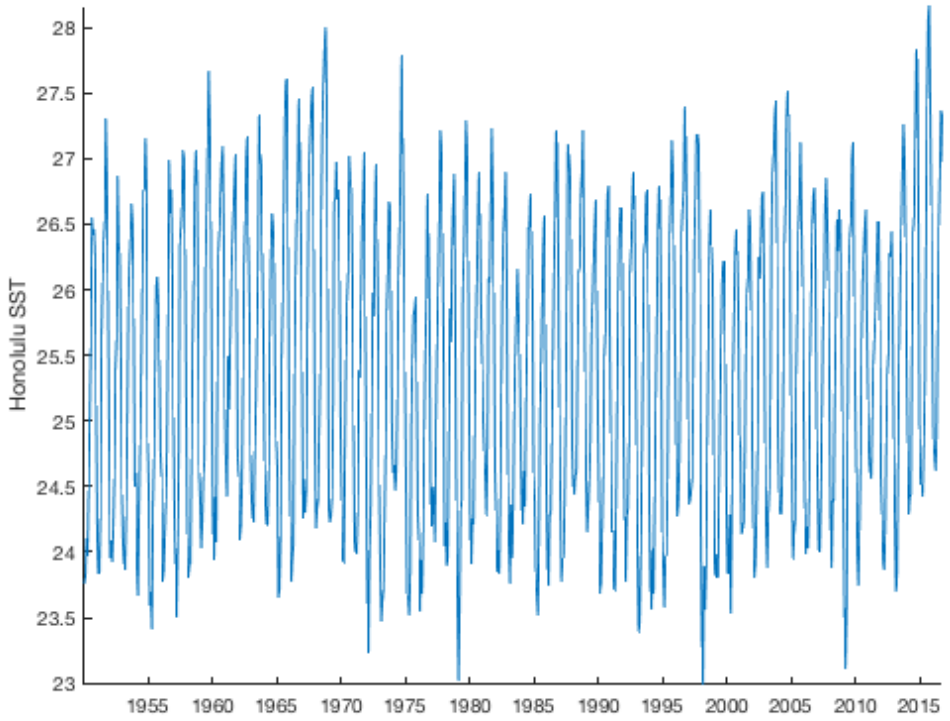
```
axis xy
cmocean thermal
xlabel 'longitude'
ylabel 'latitude'

%将檀香山绘制为红色星星，将其最近的单元网格绘制为蓝色正方形:
hold on
plot(-157.8,21.3,'rp')
plot(coords.lon,coords.lat,'bs')
text(-157.8,21.3,' Honolulu','color','r','vert','top')
text(coords.lon,coords.lat,'  nearest grid cell','color','b','vert','bottom')
```



借助坐标输出中的行和列信息，可以轻松获取檀香山单元网格的时间序列。请注意，sst 是三维矩阵，因此我们必须使用 squeeze 命令删除单维。为了清楚起见，我

```
sst_honolulu = sst(coords.row,coords.col,:);



% 移除单维:
sst_honolulu = squeeze(sst_honolulu);

figure
plot(t,sst_honolulu)
```

```
axis tight
box off
ylabel('Honolulu SST')
datetick('x','keeplimits')
```



示例 3:任意形状的多边形

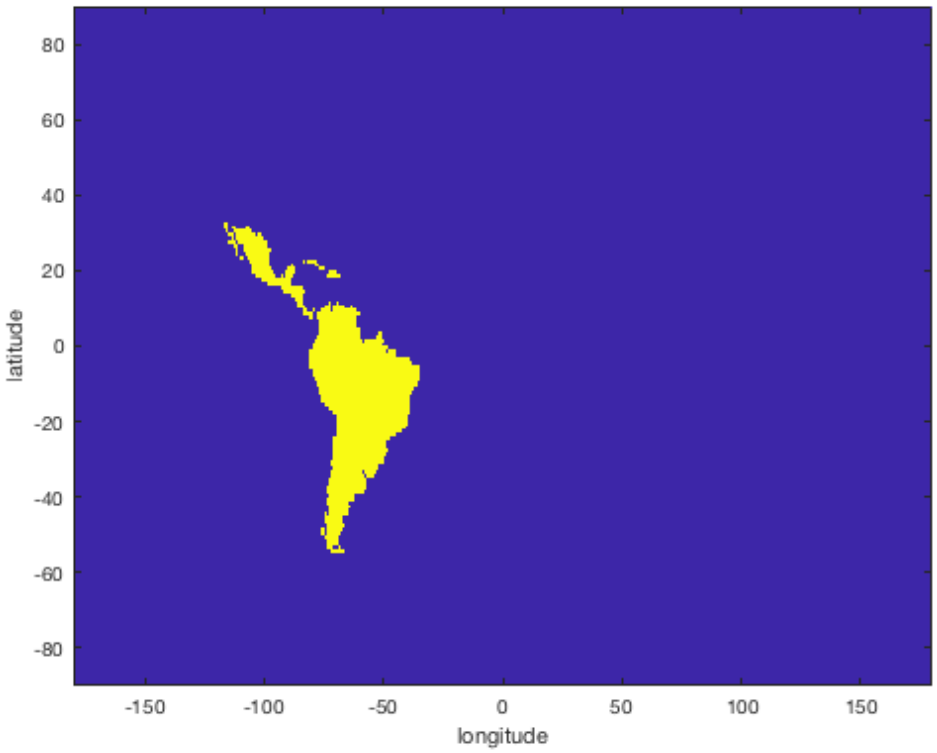
如果要制作一个任意多边形或多个多边形的掩膜，则可以使用标准的 Matlab 函数 inpolygon，也可以让 geomask 为您完成。使用 geomask，就像使用 inpolygon 一样，latv 和 lonv 可以是数字数组，或者 latv 和 lonv 可以是每个单元格中具有多边形的单元格数组。

在此示例中，我们有一个全球网格，我们想知道哪些单元网格位于拉丁美洲国家的边界内。CDT 附带有示例边界数据，并且通过浏览国家/地区名称，您可以手动选择与拉丁美洲相对应的国家/地区。下面，我选择了 20 个拉丁美洲国家/地区，我们将用它们来制作掩膜。

此示例可能要花费几秒钟的时间，因为 borderdata 中的国家轮廓线具有较高的分辨率，并且 Geomask 将单元网格与 41,500 个边界数据顶点的轮廓线进行比较需要花费时间。

```
% 这个 1 度分辨率的网格样本:
[Lat,Lon] = geogrid;

%载入一些国家边界数据:
B = load('borderdata.mat');

% 只用拉丁美洲国家:
latv = B.lat([8 17 21 33 38 39 41 48 49 55 75 78 79 120 159 161 162 165 211 214]);
```

```
lonv = B.lon([8 17 21 33 38 39 41 48 49 55 75 78 79 120 159 161 162 165 211 214]);

%查找与拉丁美洲国家相对应的单元网格:
mask = geomask(Lat,Lon,latv,lonv);

%绘制掩膜:
figure
imagescn(Lon,Lat,mask)
xlabel 'longitude'
ylabel 'latitude'
```



有了面向拉丁美洲国家的掩膜，现在就可以使用本地方法来获得一个时间序列，例如拉丁美洲国家的平均地表温度。

作者简介

这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

island 文档

island 是一个确定地理位置是对应于陆地还是水域的逻辑函数(这个是 ***is land***,不是 island!)

语法

```
tf = island(lat,lon)
```

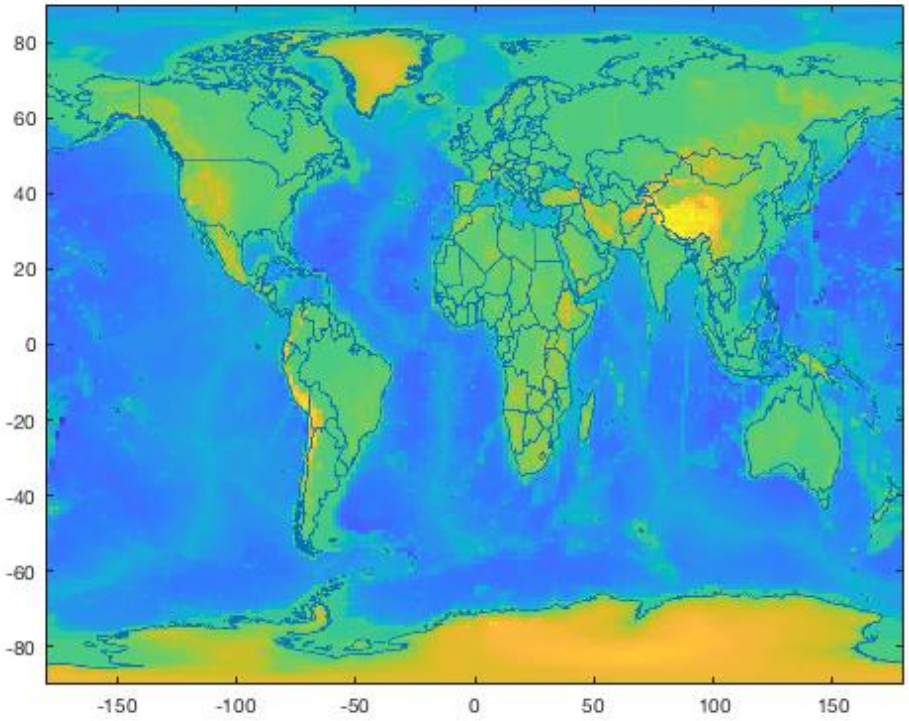
说明

tf = island(lat,lon) 使用 1/8 度分辨率的全球陆地掩模来确定 lat、lon 给定的地理位置是对应于陆地还是水域。对于陆地位置，输出为 true，否则为 false。

示例

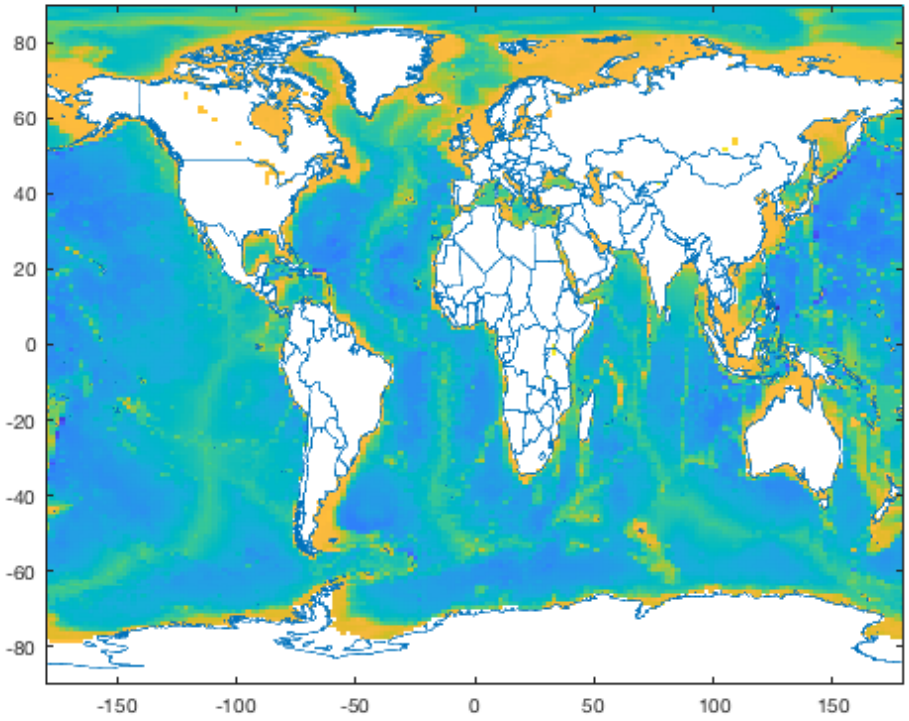
本示例使用 cdtgrid 创建一个 1x2 度的分辨率网格，我们将从 Matlab 的内置示例 peaks 函数中获得一些 z 值：（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
% 创建一个示例网格，分辨率为 1 度（经度）乘以 2 度（经度）：
[lat,lon] = cdtgrid([1 2]);

% 一些 z 样本数据:
z = topo_interp(lat,lon);

% 绘制完整网格:
imagescn(lon,lat,z)
hold on
borders
```



将陆地的值设为 NaN 掩掉: （译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
% 确定哪些栅格点对应于陆地:
land = island(lat,lon);

% 将陆地值设为 NaN:
z(land) = NaN;

% 绘制掩膜后的数据集:
figure
imagescn(lon,lat,z)
hold on
borders
```


80
60
40
20
0
-20
-40
-60
-80
-150
-100
-50
0
50
100
150

binind2latlon 文档

binind2latlon 将正弦网格的合并索引值转换为地理坐标。
更多正弦网格的信息请参考 此处。

语法

```
[lat,lon] = binind2latlon(binind)
[lat,lon] = binind2latlon(...,'rows',NumberOfRows)
```

说明

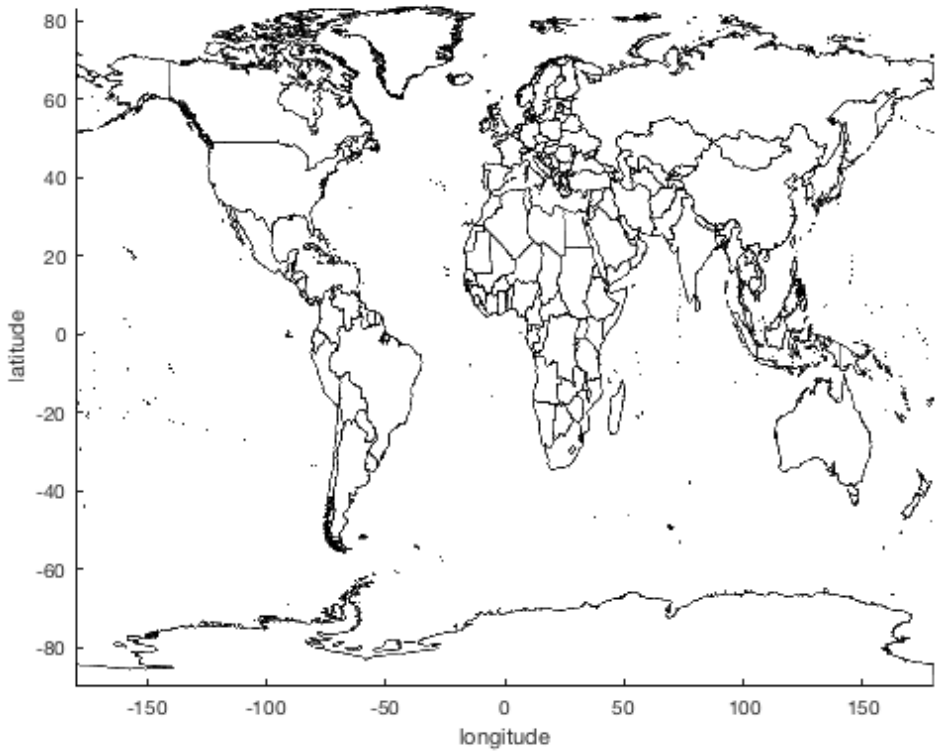
`[lat,lon] = binind2latlon(binind)` 给出 bin 索引 binind 的地理坐标。
`[lat,lon] = binind2latlon(...,'rows',NumberOfRows)` 指定网格中的行数。默认情况下，binind2latlon 会尝试自动计算出行数，但不能保证它会起作用。如果知道行数，请确保指定行数。根据此处的“分辨率”表，常见的行数如下：

- 180 行 1 度网格
- 2160 行 5 分网格
- 4320 行 1.5 分网格

示例 1: 小网格

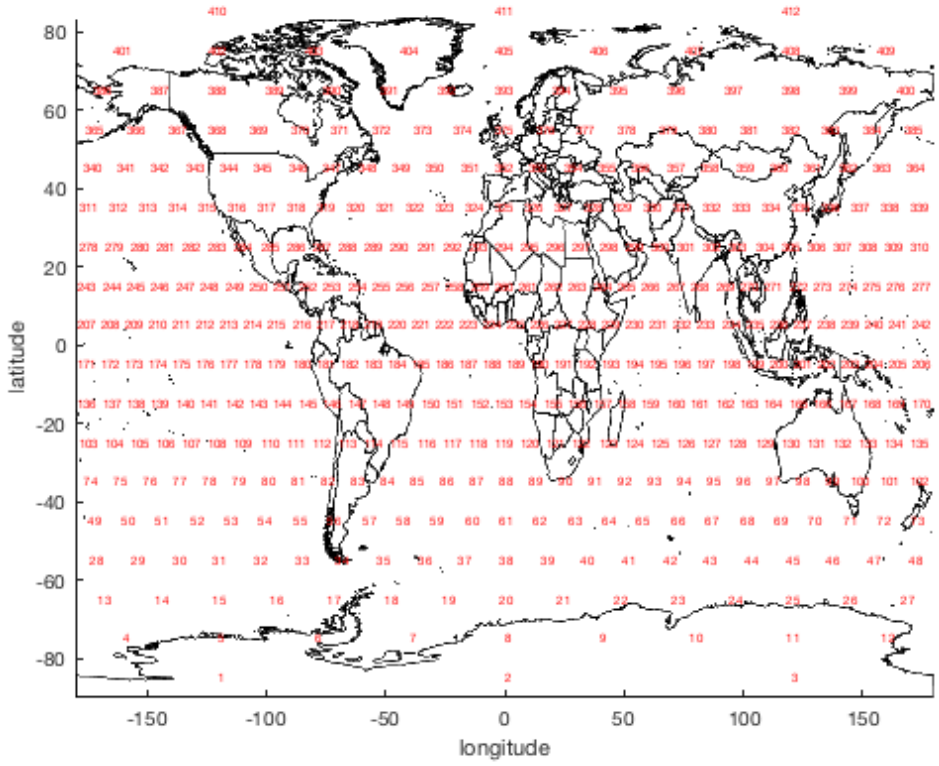
让我们重新创建这里 here 看到简单的 18 行示例。首先使用我的 borders 功能绘制国界线：（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
borders('countries','k-')
axis tight
xlabel 'longitude'
ylabel 'latitude'
```



NASA 站点上的示例包含跨越 18 个纬度行的 412 个间隔：（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
% 间隔数列:
bins = (1:412)';

% 每个间隔号对应什么坐标?
[lat,lon] = binind2latlon(bins,'rows',18);

%以其地理坐标为中心标记每个间隔:
text(lon,lat,num2str(bins),'color','r','horiz','center','fontsize',6)
```



示例 2: 实际数据

本示例使用 此处找到的一些每月 CHL 数据。首先读取数据。尽管这些文件以.nc 结尾，但是 ncread 函数无法识别我们要绘制的变量的数据类型。因此，我们必须改用 h5read:

```
A = h5read('A20021822002212.L3b_MO_CHL.nc','/level-3_binned_data/chlor_a');
B = h5read('A20021822002212.L3b_MO_CHL.nc','/level-3_binned_data/BinList');

% 为方便起见，将 z 和 bin 分配为变量:
z = double(A.sum_squared);
bins = B.bin_num;
```

可以很容易地找到每个 z 值的坐标:

```
[lat,lon] = binind2latlon(bins);
```

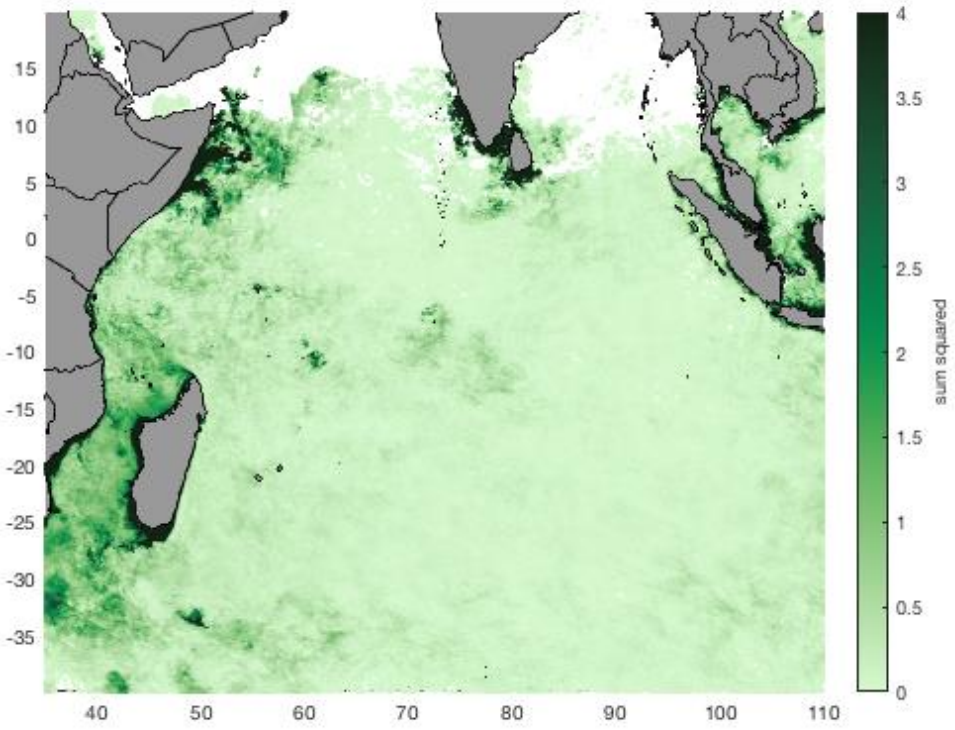
注意这里有超过一千一百万的测量值，所以我们可能只想看一下印度洋这个区域（40S-20N，35E-110E）的一小部分测量值。使用 geomask 获取与这些坐标对应的索引:

```
ind = geomask(lat,lon,[-40 20],[35 110]);
```

该地理区域中有 1625033 个数据点，但不幸的是，它们不能构成一个完美的网格，因此，如果要使用网格化数据，则必须使用 griddata 或 scatteredInterpolant。或者，您可以对数据进行散点图绘制。像我

正在用的旧版本 Matlab，当您给分散大量数据点时会挂，因此下面我使用 Aslak Grinsted 的 fastscatter 函数。我也在使用 cmocean 颜色图 (Thyng et al., 2016)。

```
figure

fastscatter(lon(ind),lat(ind),z(ind),'markersize',.1)
axis tight
caxis([0 4])
cb = colorbar;
ylabel(cb,'sum squared')
cmocean algae
borders('countries','facecolor',0.6*[1 1 1])
```



作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

空间分布

- filt2 对栅格数据执行高通、低通、带通或带阻 2D 高斯滤波器。
- scatstat2 返回每个值的给定半径内的所有点的统计值。这类似于采用二维移动平均值，但是不需要将点等距分布。
- eof 为时空数据分析提供了变化的特征模态图和相应的主成分时间序列。
- reof 根据指定的 EOF 模态重建 eof 异常的时间序列。
- corr3 计算时间序列和三维数据集的线性或秩相关。
- xcorr3 给出了三维时空数据集的单元网格与参考时间序列之间的相关系数映射。
- xcov3 给出了三维时空数据集的单元网格与参考时间序列之间的协方差映射。

filt2 文档

filt2 函数对栅格数据（例如地形、大气、海洋或任何种类的地理空间数据）执行高通、低通、带通或带阻 2D 高斯滤波器。此功能旨在轻松删除比给定特征波长更长或更短的特征。输入网格*可以*包含 NaN！
另请参见: filt1。

语法

```
Zf = filt2(Z,res,lambda,filtertype)
```

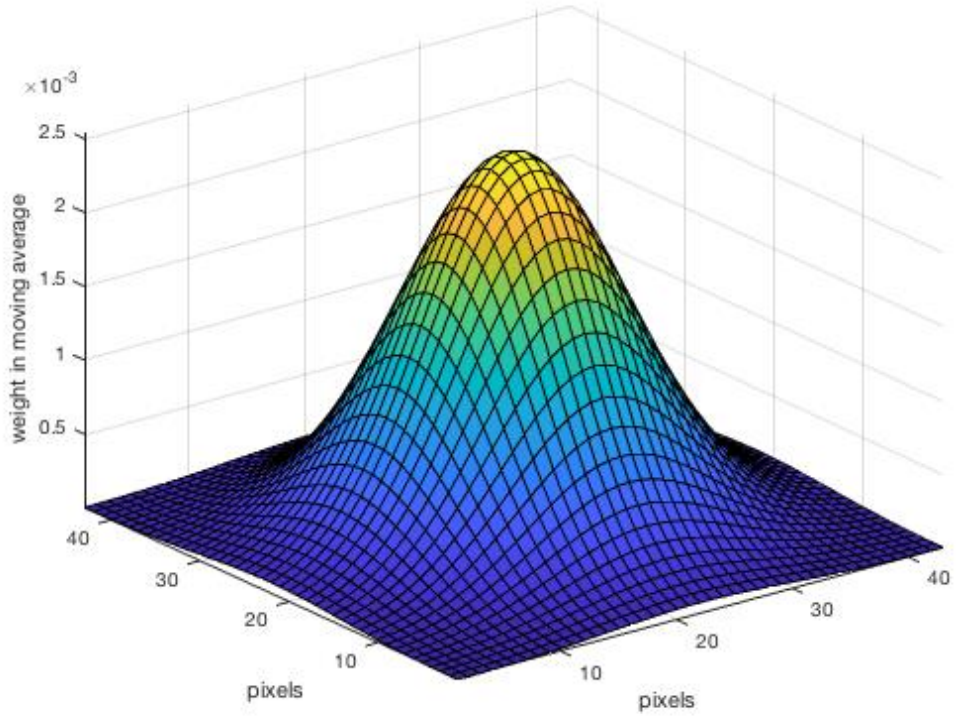
说明

Zf = filt2(Z,res,lambda,filtertype) 将分辨率为 res 的 2D 数据集 Z 过滤到近似波长 λ。如果低通或高通的过滤器类型为“ lp”或“ hp”，则 lambda 必须为标量值。 果对于带通或带阻，滤波器类型为“ bp”或“ bs”，则 λ 必须是两个截止波长的二元素数组

此类过滤器的说明

有很多方法可以过滤波。由 filt2 实施的方法比二维 FFT 复杂，但比起简单的二维移动平均略有细微差别。如果您有数据矩阵 Z，并且要创建 2D 低通滤波的矩阵 Zlow，则通常的做法是用 Z 中所有附近点的平均值填充 Zlow 中的每个点。您可以指定一个框大小为 25 点乘 25 点，然后沿着 Zlow 的每一行和每一列向下填充，并用 Z 中所有 25 乘 25 点的平均值填充。这种方法有效地消除了高频噪声，但是这有点不雅致，因为距离框中心 12 点远的点对滤波后的中心点值的贡献与中心点本身一样多。此外，移动的正方形具有来自角的贡献，角的贡献是距顶部或侧面点 1.4 倍的距离。使用相对于中心点对称的加权系统通常更有意义，在该系统中，靠近中心的点比远处的点贡献更多。
由 filt2 实现的二维移动高斯窗口类似于移动二维正方形窗口，但是每个点的贡献由关于中心点对称的高斯曲线加权。因此，在方形移动窗口的权重看起来像是三维空间中的盒子的情况下，高斯窗口的权重看起来更像是圆锥形。如果给定 filt2 的截止波长 10 km 和 200 m 分辨率数据集的 a，则这是 filt2 生成的加权曲线：

```
lambda = 10; % 以公里为单位的截止“波长”
res = 0.2;   %数据集分辨率 200m

% 定义高斯曲线的标准差:
sigma = (lambda/res) /(2*pi);

% 创建 2D 高斯曲线:
f = fspecial('gaussian',2*ceil(2.6*sigma)+1,sigma);

surf(f)
axis tight
xlabel 'pixels'
ylabel 'pixels'
zlabel 'weight in moving average'
```



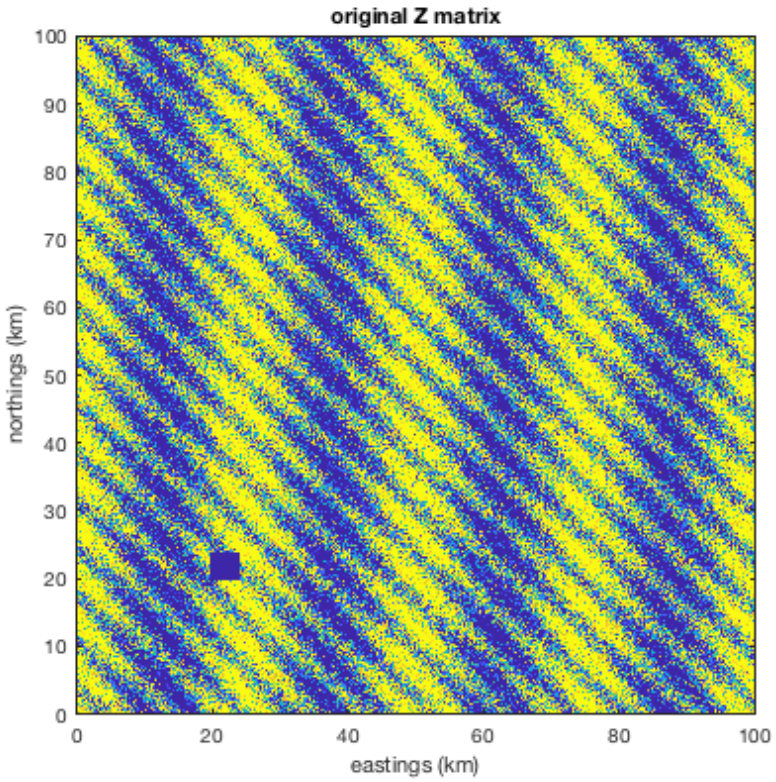
此处的 sigma 值约为 8 个点，此时的权重约为中心幅度的 0.6 倍。

示例

考虑一个分辨率为 200 m 的 100 km x 100 km 高程数据集。它具有与南北方向对齐的一些 25 km 长的波长特征，对角线定向的一些短于∽5 km 特征，以及相当数量的随机噪声。还有一些丢失的数据。这是您的数据集的样子：

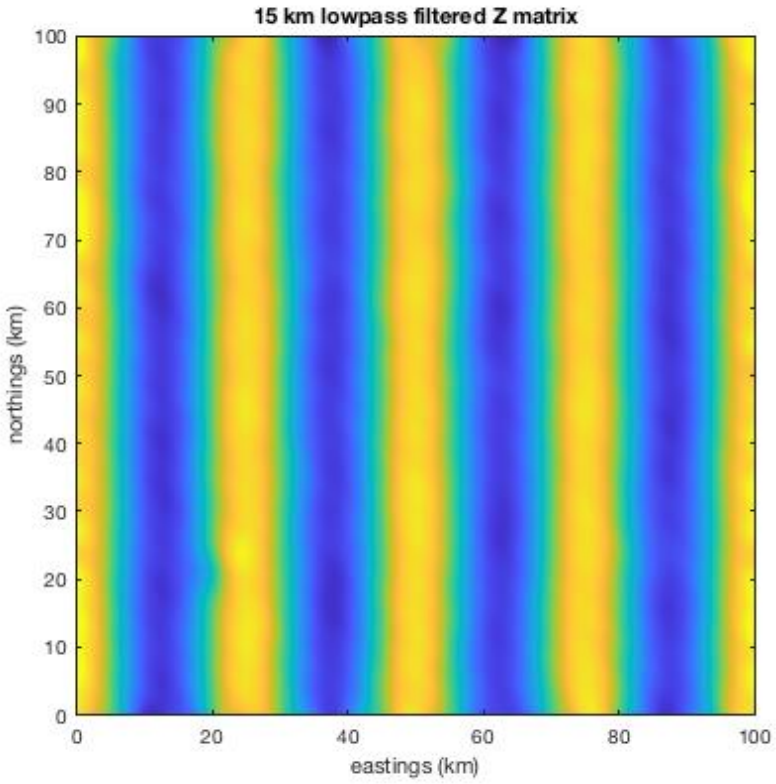
```
res = 0.2; % 200 m 分辨率

x = 0:res:100; % 打东边 0 to 100 km
y = 0:res:100; % 打北边 from 0 to 100 km
[X,Y] = meshgrid(x,y);

% Z 包含 25km 特征，~5 km 对角特征和噪声:
Z = cos(2*pi*X/25)+cos(2*pi*(X+Y)/7)+randn(size(X));

% Z 还有一些缺失数据:
Z(100:120,100:120) = nan;

imagesc(x,y,Z);
axis xy image
caxis([-1 1])
title ' original Z matrix '
xlabel ' eastings (km) '
```

```
ylabel ' northings (km) '
```



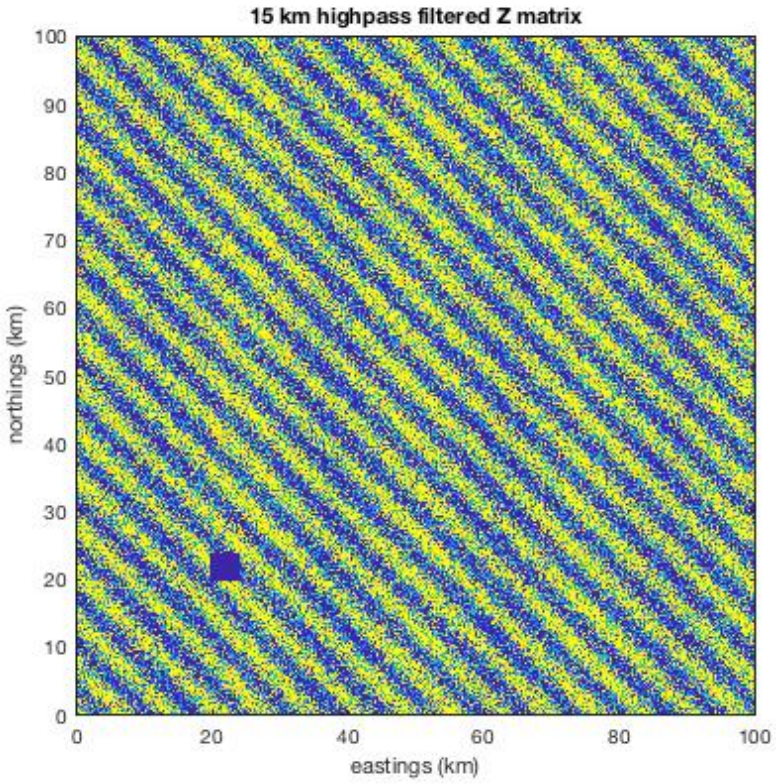
过滤掉所有短于 15 km 的特征：

```
Zlow = filt2(Z,res,15,'lp');

imagesc(x,y,Zlow);
axis xy image
caxis([-1 1])
title ' 15 km lowpass filtered Z matrix '
xlabel ' eastings (km) '
ylabel ' northings (km) '
```



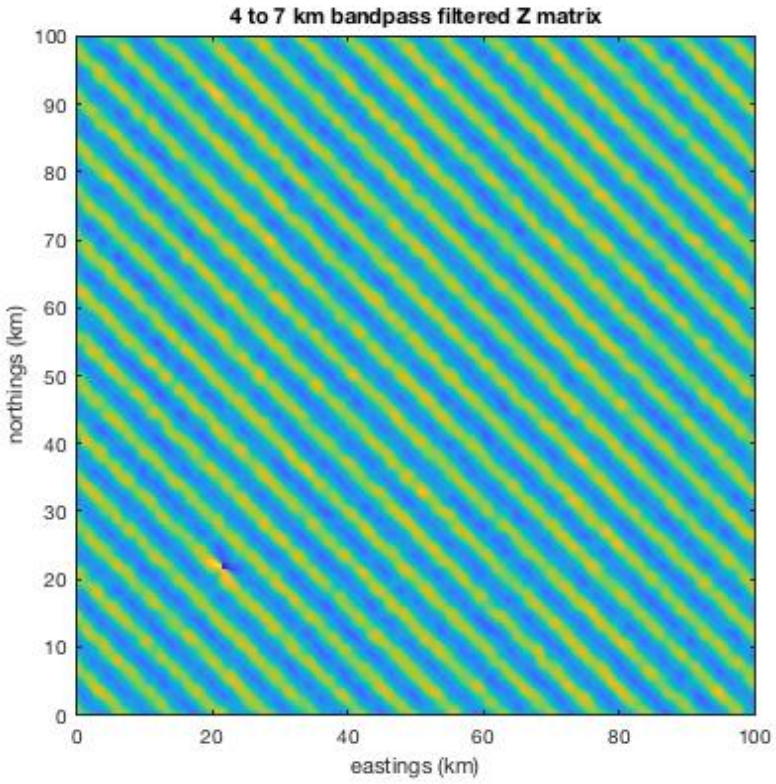
同样，过滤掉所有*超过* 15 公里的特征：

```
Zhi = filt2(Z,res,15,'hp');
imagesc(x,y,Zhi);
axis xy image
caxis([-1 1])
title ' 15 km highpass filtered Z matrix '
xlabel ' eastings (km) '
ylabel ' northings (km) '
```



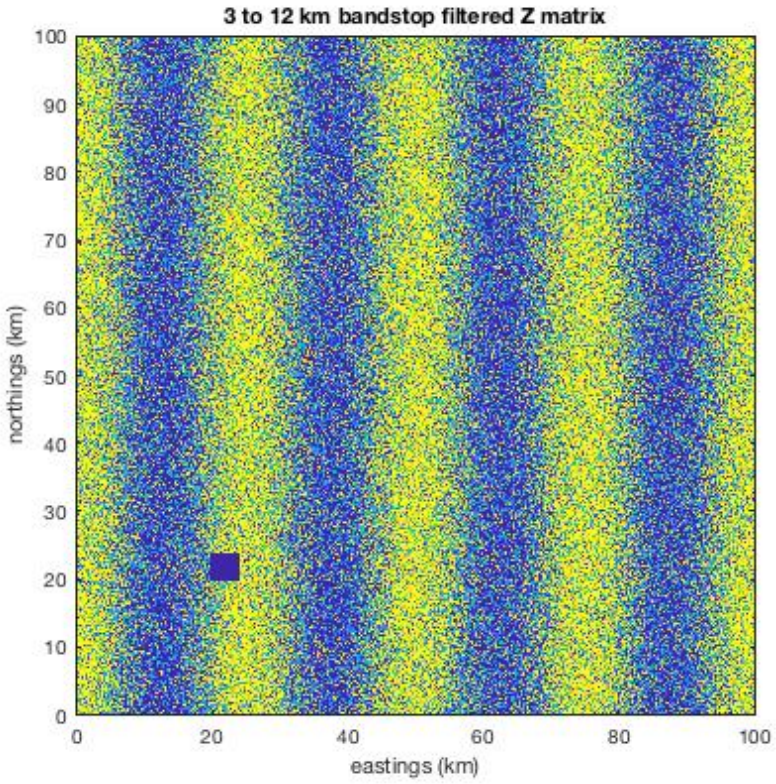
或者，我们可以通过去除 25 km 长的波长特征并去除短波长的斑点噪声来保持对角线：在这里，您将看到它成为一种平衡行为-随着波长的变化，您将发现更宽的波长范围。在带通区域，更多的噪音通过，但更多的信号通过。收紧带通区域，您还将滤除我们想要保留的 5 km 波长中的某些波长，因为该滤波器的截止频率不高：

```
Zbp = filt2(Z,res,[4 7],'bp');
imagesc(x,y,Zbp);
axis xy image
caxis([-1 1])
title ' 4 to 7 km bandpass filtered Z matrix '
xlabel ' eastings (km) '
ylabel ' northings (km) '
```



与带通滤波器类似，我们可以使用带阻滤波器来消除 5 km 波长对角线特征：

```
Zbs = filt2(Z,res,[3 12],'bs');
imagesc(x,y,Zbs);
axis xy image
caxis([-1 1])
title ' 3 to 12 km bandstop filtered Z matrix '
xlabel ' eastings (km) '
ylabel ' northings (km) '
```



有关地理参考数据的特别说明

此功能非常适合过滤 2D 空间数据，但不要在经度和纬度均等的数据集上使用它！那是因为过滤到 N 度没有任何意义。相反，此功能可用于投影坐标系中的数据，其中网格尺寸为米或千米，甚至为英尺。但是不要在 `cdtgrid` 创建的网格类型上使用它。

作者简介

这个函数是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 在 2016 年 11 月写的；但是，我所做的只是重新包装了 Carlos Adrian Vargas Aguilera 的出色 ndnanfilter 函数，以使其更容易用于此特定应用程序。非常感谢 Carlos 精心设计的代码和清晰的文档。

scatstat2　文档（参见前文）

eof 文档

eof 函数为时空数据分析提供了变化的特征模式图和相应的主成分时间序列。此函数是专为 3D 数据矩阵设计的，例如海表温度，其中维 1 和 2 是空间维（例如 lat 和 lon；lon 和 lat； x 和 y 等），而第 3 维代表不同的切片或及时的数据快照。

另请参见: reof.

语法

```
eof_maps = eof(A)
eof_maps = eof(A,n)
eof_maps = eof(...,'mask',mask)
[eof_maps,pc,expvar] = eof(...)
```

说明

eof_maps = eof(A) 计算 A 中的所有可变模式，其中 A 是一个 3D 矩阵，其前两个维是空间的，第三个维是时间的，并且假定数据在时间上等距。输出 eof_maps 具有与 A 相同的维度，其中沿第三维维度的每个映射都表示重要性的可变性顺序的模式。

eof_maps = eof(A,n) 仅计算变化的前 n 个模式。对于大型数据集，仅计算所需模式的数量在计算上会更快。如果未指定 n，则将计算所有 EOF（每个时间片一个）。.

eof_maps = eof(...,'mask',mask) 仅对由逻辑掩膜中的网格所代表的网格单元执行 EOF 分析，该逻辑掩膜的尺寸对应于 A 的维度 1 和 2。提供此选项是为了避免求解那些不需要解决，也不必让您对一个区域与另一个区域进行分析。默认情况下，A 中任何包含 NaN 的网格单元会被掩膜掉。

[eof_maps,pc,expvar] = eof(...) 返回主成分时间序列 pc，其行分别表示从 1 到 n 的不同模式，列对应于时间步长。例如，pc(1,:)是可变性的第一个（主要）模式的时间序列。第三个输出 expvar 是每种模式所解释的方差百分比。请参阅以下有关解释 expvar 的注释。

一个简单的示例

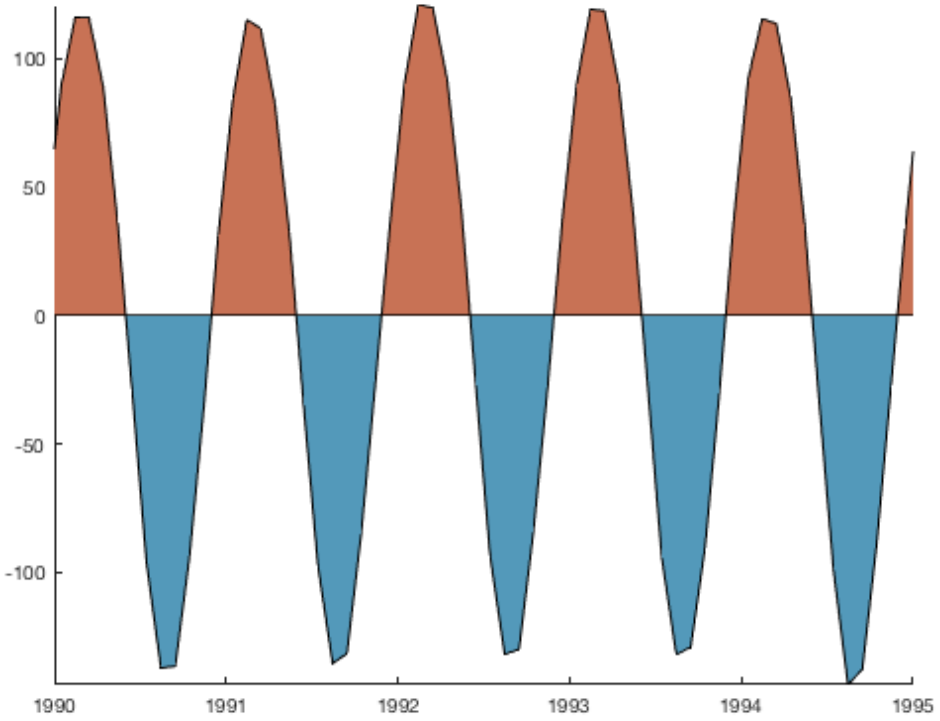
这是有关如何使用 eof 函数的快速示例。正确的 EOF 分析需要在计算 EOF 之前对数据进行去趋势处理和去季节趋势处理，这些步骤在下面的教程中进行了描述，但是现在让我们假设这个样本数据集已准备好进行分析。加载样本数据，然后计算第一 EOF。

在这里，我使用 cmocean delta 颜色图(Thyng et al., 2016)绘制了第一个 EOF 图，并带有'pivot'参数以确保其居中于零。

```
% 载入样本数据:
load pacific_sst.mat

% 计算海面温度的第一个 EOF 及其主要成分时间序列:
[eofmap,pc] = eof(sst,1);

% 绘制第一 EOF 图:
imagescn(lon,lat,eofmap);
axis xy image off

% 选项：使用 cmocean 颜色图:
cmocean('delta','pivot',0)
```

这是 SST 数据集的第一 EOF，但是由于我们还没有删除季节性周期，因此第一 EOF 主要表示季节性变化。为了证明上述模式与季节性周期有关，请查看相应的主成分时间序列。

```
figure

anomaly(t,pc)

axis tight
xlim([datenum('jan 1, 1990') datenum('jan 1, 1995')])
datetick('x','keeplimits')
```



对我来说，这看起来很季节性。

教程：从原始的气候再分析数据到 ENSO、PDO 等

CDT 附带有一个名为 pacific_sst.mat 的样本数据集，它是 Hadley Center 的 HadISST 海面温度数据集的缩采样子集。在本教程的最后，有一节描述了如何将原始 NetCDF 数据导入 Matlab 以及用于对其进行子集化的过程。如果您从上到下遵循本教程，则应该能够将 EOF 分析应用于任何类似的数据集。
如果尚未加载示例数据集，请立即加载它，并通过检查变量的名称和大小来了解其内容：

```
load pacific_sst.mat
whos
```

```
  Name          Size                 Bytes  Class     Attributes

  eofmap        60x55                26400  double
  lat           60x1                   480  double
  lon           55x1                   440  double
  pc             1x802                6416  double
  sst           60x55x802          21172800  double
```

```
  t           802x1                    6416  double
```

因此，我们有一个三维 sst 矩阵，其尺寸对应于 lat × lon × time。您问什么时间范围，什么时间步长？让我们看一下第一个和最后一个日期，以及平均时间步长：

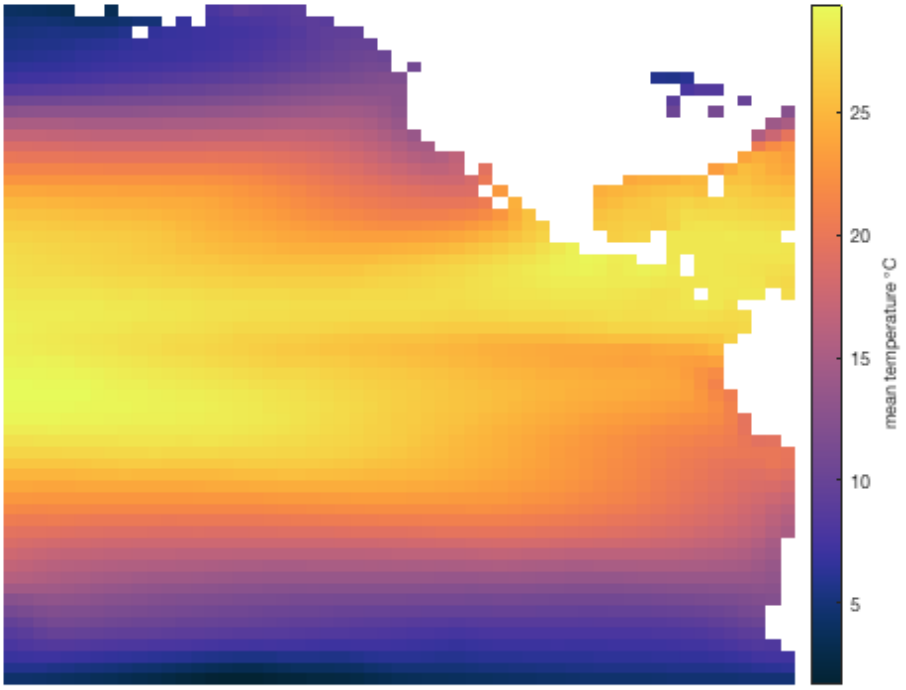
```
datestr(t([1 end]))

mean(diff(t))
```

```
ans =

  2×20 char array

    '15-Jan-1950 12:00:00'

    '15-Oct-2016 12:00:00'

ans =

   30.4370
```

平均海面温度

好的，所以这是每月数据，集中在 1950 年至 2016 年每个月的 15 日左右。要了解数据集的样子，请显示该时间段内的平均温度。我正在使用 `imagescn`，它会自动使 NaN 值透明，但是您可以根据需要使用 imagesc、pcolor 或任何等效的映射工具箱函数。我也在使用 `cmocean` *热*色图(Thyng et al., 2016):

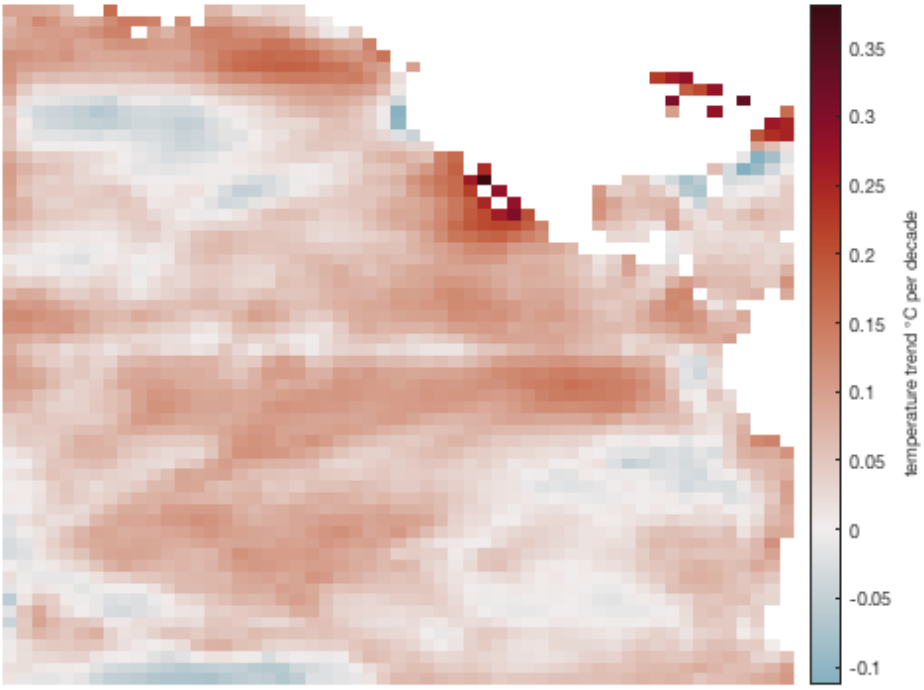
```
figure

imagescn(lon,lat,mean(sst,3));

axis xy off
cb = colorbar;
ylabel(cb,' mean temperature {\circ}C ')
cmocean thermal
```



全球变暖

全球变暖真的存在吗？使用趋势函数，我们可以轻松获得 1950 年至 2016 年的线性温度趋势。确保将趋势乘以 10 * 365.25，以将每天的度数转换为每十年的度数：

```
imagescn(lon,lat,10*365.25*trend(sst,t,3))

axis xy off
cb = colorbar;
ylabel(cb,' temperature trend {\circ}C per decade ')
cmocean('balance','pivot')
```



移除全球变暖信号

全球变暖趋势很有趣，但是 EOF 分析只涉及可变性，而不是长期趋势，因此我们必须通过 detrend3 消除趋势：

```
sst = detrend3(sst,t);
```

移除季节性循环

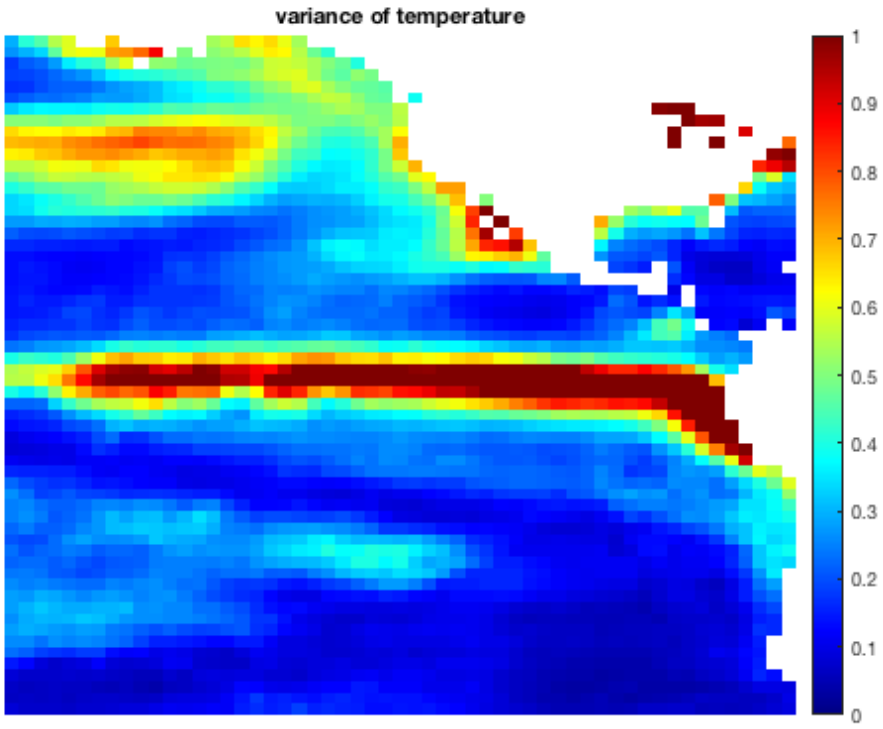
如果再次绘制温度趋势图，您会发现它已全部减小到零，可能会有几 eps 的数字噪声。现在，这是一个甚至 Anthony Watts 都会赞成的 SST 数据集。
现在，我们对 SST 数据集进行了去趋势处理（这也删除了均值），但是它仍然包含相当多的季节性变化，应在进行 EOF 分析之前将其删除，因为我们对季节性信号不感兴趣。

```
sst = deseason(sst,t);
```

因此，现在我们的 sst 数据集已被去趋势化、去除了均值，并且去除了季节性周期。剩下的只是异常-发生变化的事情，但不是长期趋势或短期年度周期。这是我们的 sst 异常数据集的剩余方差：

```
figure

imagescn(lon,lat,var(sst,[],3));

axis xy off
```

```
colorbar
title('variance of temperature')
colormap(jet) % 除了重画老图，否则 jet 不可原谅
caxis([0 1])
```

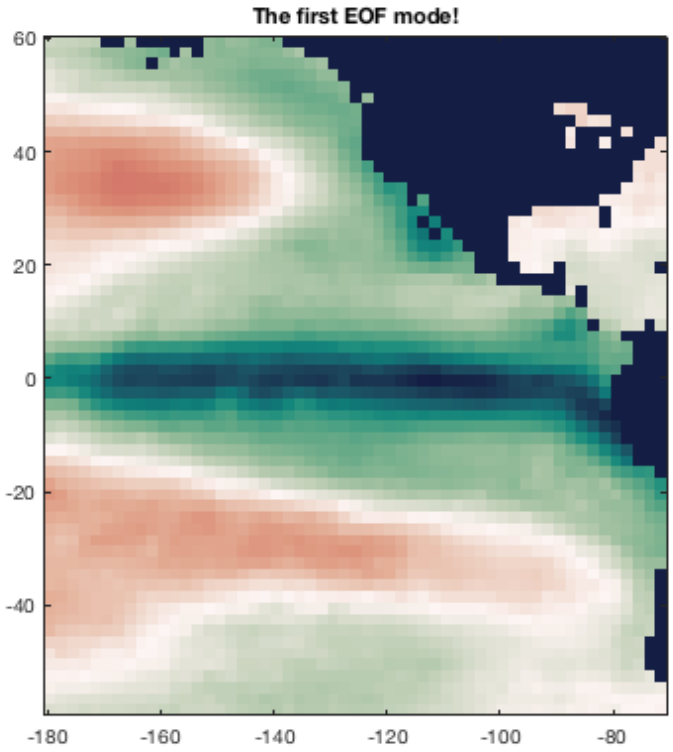


上面的地图与 Messie and Chavez (2011)的图 2a 很好地吻合，这说明我们处在正确的轨道上。

计算 EOFs

EOF 分析使我们不仅了解事物发生变化的地方，而且了解变化的频率以及哪些区域倾向于一起变化或彼此异相。使用我们经过去趋势处理，已去季节化的 sst 数据集，使用 eof 函数可以很简单地进行 EOF 分析：

```
[eof_maps,pc,expv] = eof(sst);



% 绘制第一模:
figure
imagesc(lon,lat,eof_maps(:,:,1))
axis xy image
cmocean('curl','pivot')
title 'The first EOF mode!'
```



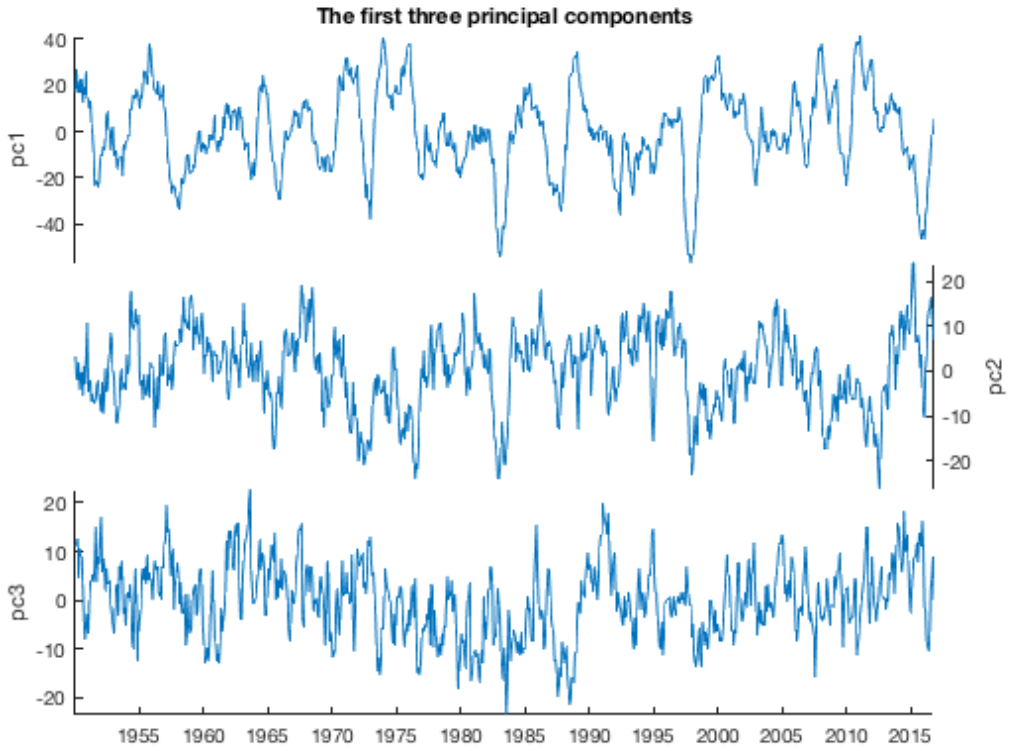
特征向量分析的行为很有趣，可以产生正或负的 EOF 映射，并且每次使用相同的精确输入时，解决方案可能会有所不同。正解和负解都同样有效-考虑一下鼓头的振动模式，其中鼓头的某些区域上升而其他区域下降，然后它们切换-同样，SST 变异性的特征值解可能是正面或负面。唯一重要的是，当我们从 EOF 解决方案重建时间序列时，我们将每个 EOF 映射乘以其对应的主分量（pc）。

每次使用相同的数据运行 eof 函数时，都会产生一致的结果，但是不要担心解决方案的符号与其他人的结果的符号不匹配，如果他们使用其他程序来计算 EOF 的话，那只是意味着他们的程序选择了相反符号的解决方案，这很好。

正如 EOF 贴图可以具有正解或负解，并且两者都同样有效时一样，在显示 EOF 贴图的幅度方面也有一定的灵活性。您可以将 EOF 映射的大小乘以所需的任何值，只要将相应的主成分时间序列除以相同的值即可。让我们看一下前三个可变模式的时间序列，并用 subsubplot 绘制：

```
figure

subsubplot(3,1,1)

plot(t,pc(1,:))

box off
axis tight
ylabel 'pc1'
title 'The first three principal components'

subsubplot(3,1,2)
plot(t,pc(2,:))
```

```
box off
axis tight
set(gca,'yaxislocation','right')
ylabel 'pc2'

subsubplot(3,1,3)
plot(t,pc(3,:))
box off
axis tight
ylabel 'pc3'
datetick('x','keeplimits')
```



可选缩放主成分和 EOF 映射

那些主成分时间序列按原样很好，但是有些人更喜欢按比例缩放每个时间序列以跨越所需的范围。查看 Messie and Chavez (2011)的图 5，似乎他们选择缩放每个主成分时间序列，使其跨过-1 到 1 的范围。让我们做同样的事情，将每个主成分时间序列除以其最大值。并且不要忘记将相应的 EOF 映射乘以相同的值：

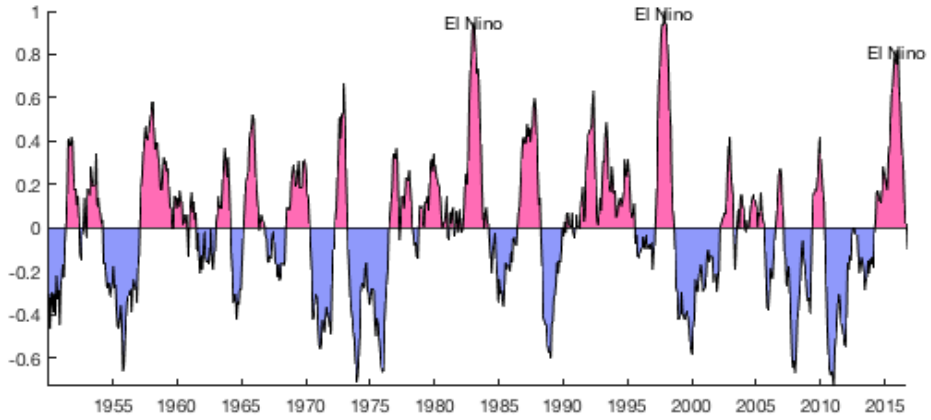
```
for k = 1:size(pc,1)

   % 在每个主成分的时间序列中找到最大值：
   maxval = max(abs(pc(k,:)));
```

```
    % 将时间序列除以其最大值:
    pc(k,:) = pc(k,:)/maxval;

    % 乘以对应的 EOF 映射:
    eof_maps(:,:,k) = eof_maps(:,:,k)*maxval;
end
```

厄尔尼诺南方涛动（ENSO）时间序列

去趋势的，去季节化的 SST 的第一种模式与 ENSO 相关联。我们可以再次将时间序列绘制为简单的线图，但是经常会填充异常图。让我们使用 `anomaly` 来绘制第一个模式，并乘以-1 以匹配 Messie and Chavez (2011)的图 5 的符号。

```
figure('pos',[100 100 600 250])
anomaly(t,-pc(1,:),'topcolor',rgb('bubblegum'),...
    'bottomcolor',rgb('periwinkle blue')) % 第一主成份是 enso
box off
axis tight
datetick('x','keeplimits')
text([724316 729713 736290],[.95 .99 .81],'El Nino','horiz','center')
```

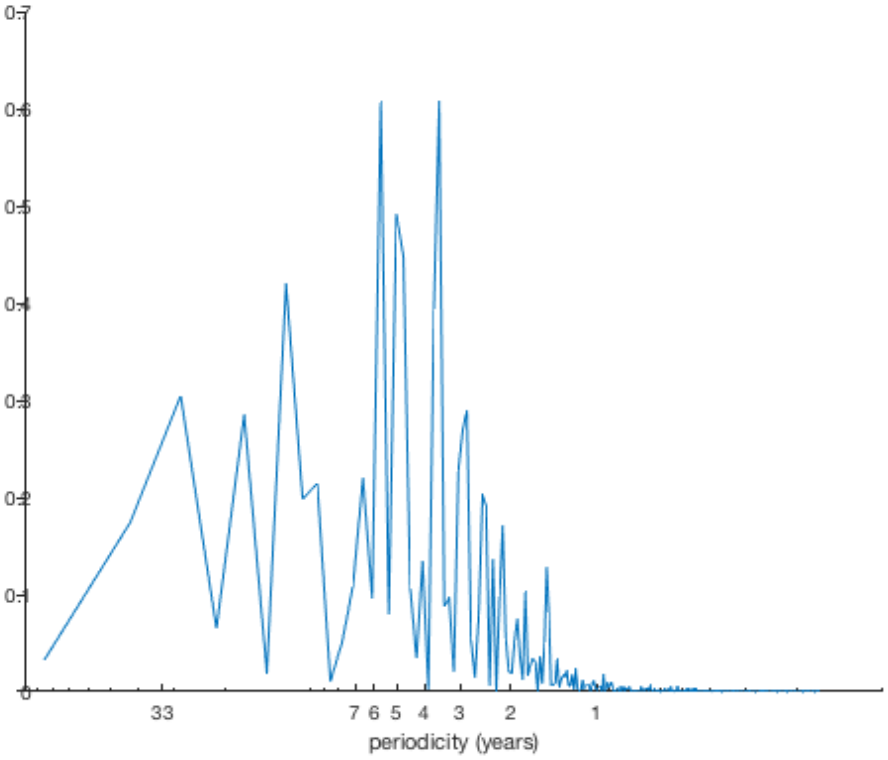


果然， 有记录以来最强劲的厄尔尼诺事件发生在 1982-1983 年，1997-1998 年和 2014-2016 年。有关调查 ENSO 的更多方法，请查看 enso 函数。

ENSO 在频域

有时我们听到厄尔尼诺现象的特征频率是每五年一次或五到七年一次，或者有时您听到的频率是每两到七年一次。很难在时间序列中看到这一点，因此我们使用 plotpsd 绘制了频域中的第一个主成分，指定了每年 12 个样本的采样频率，在对数 x 轴上绘制，x 值以 λ（年）为单位，而不是频率：

```
figure

plotpsd(pc(1,:),12,'logx','lambda')
xlabel 'periodicity (years)'
set(gca,'xtick',[1:7 33])
```

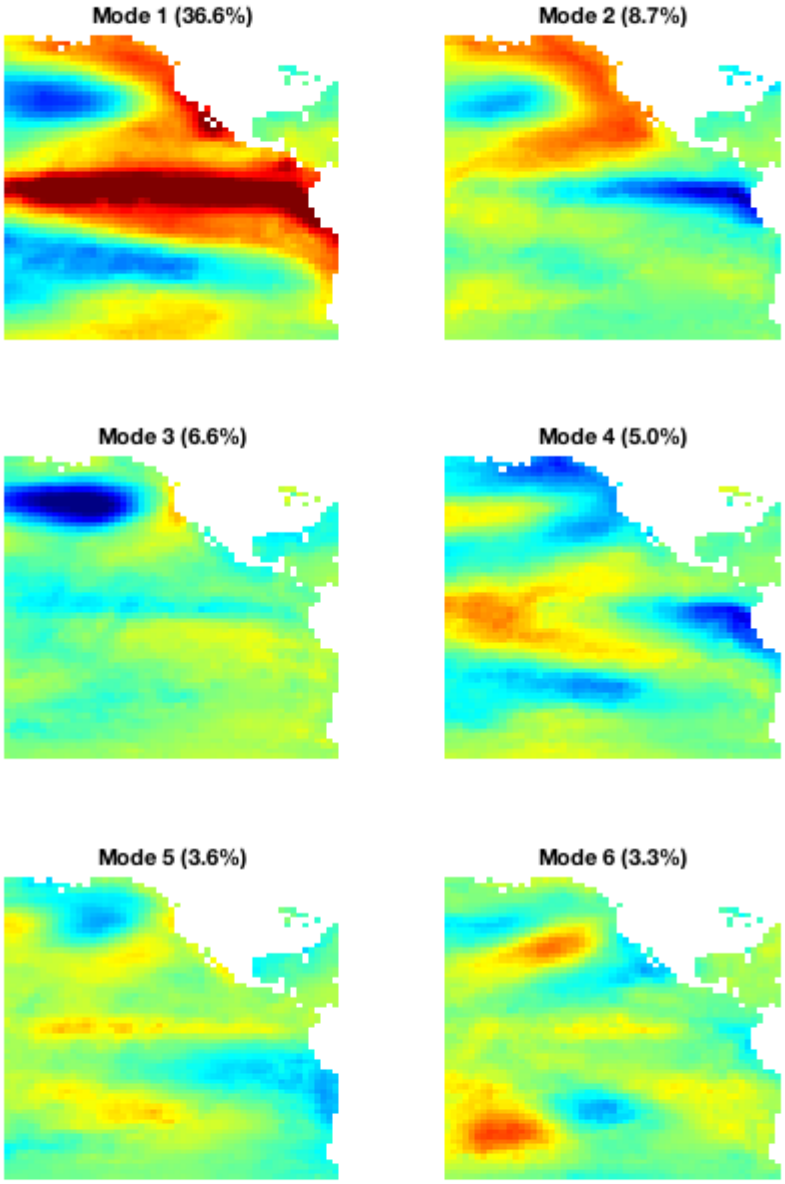


如您所见，ENSO 信号没有明确定义的谐振频率，但是在整个二到七年的范围内都有能量。我还标记了 33 年的周期性，因为对于该特定数据集而言，这就是奈奎斯特-任何周期比奈奎斯特（或其附近的任何地方）更长的能量都应该被视为垃圾。

变量图

EOF 不仅与时间序列有关，还与随时间变化的空间模式有关。每种模式都有可变的特征模式，就像鼓头的不同振动模式一样。在任何给定时间，可以对不同模式进行求和，以创建当时温度异常的总图。经验正交函数的正交部分意味着每种模式都倾向于做自己的事情，而与其他模式无关。让我们通过重新创建 Messie and Chavez (2011)的图 4 来查看前六个模式。我将某些模式乘以负数，因为我想匹配它们的符号，请记住，我们可以这样做。

```
s = [-1 1 -1 1 -1 1]; % (乘以符号来匹配 Messie and Chavez 2011)

figure('pos',[100 100 500 700])
for k = 1:6
   subplot(3,2,k)
   imagescn(lon,lat,eof_maps(:,:,k)*s(k));
   axis xy off
   title(['Mode ',num2str(k),' (',num2str(expv(k),'%0.1f'),'%)'])
   caxis([-2 2])
end

colormap jet
```

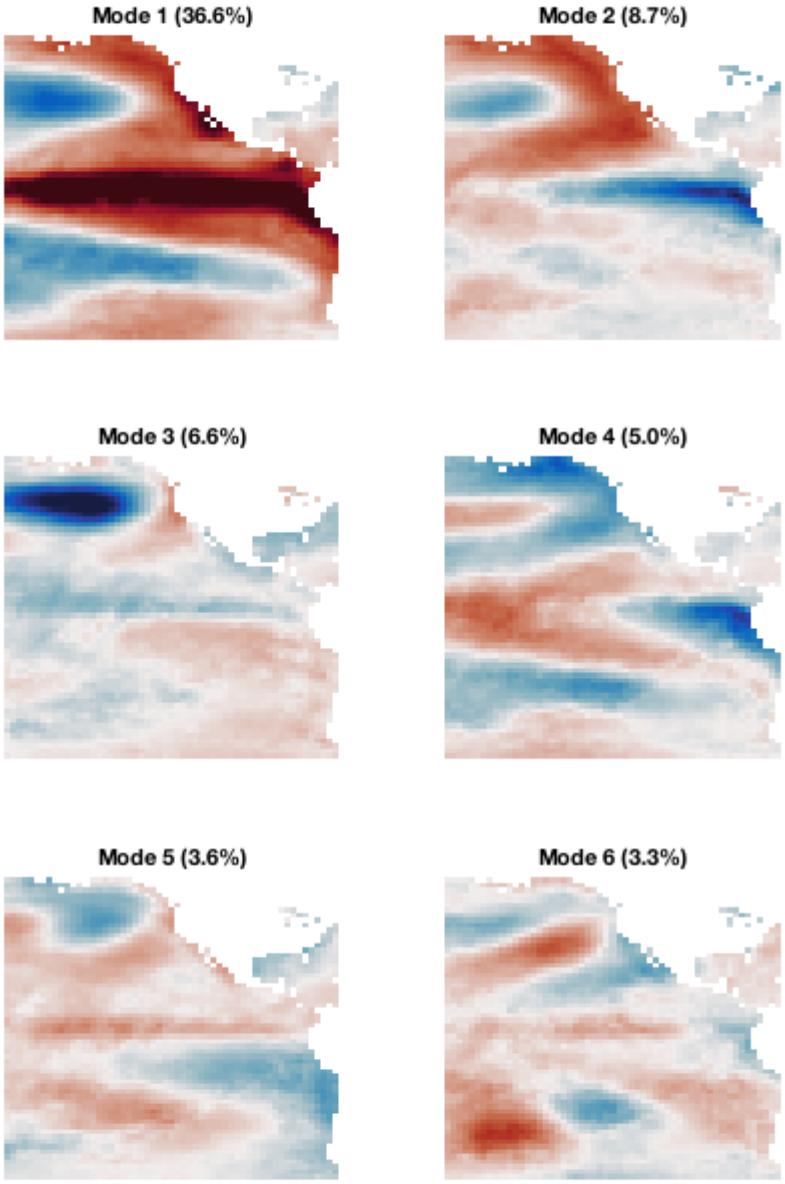


每种模式所解释的百分比差异与 Messie 和 Chavez 不匹配，因为我们使用的时间序列比它们使用的时间序列短得多，并且我们还使用了世界数据的空间子集。尽管如此，模式通常是一致的。

jet 色图与 Messie 和 Chavez 使用的颜色图不完全相同，这解释了为什么上面的某些图案可能看起来与 Messie 和 Chavez 略有不同。但是，由于我们谈论的是颜色图，所以彩虹实际上在表示数值数据方面非常糟糕(Thyng et al, 2016)。 这也是科学的遗憾，我们无法在不确切知道发行版本中使用了什么颜色的情况下完全复制该图，但是我跑偏了……

假设这些图表示异常，则应该用发散的颜色图表示，这些图赋予零的每一侧相等的权重。让我们将此图中所有子图的颜色图设置为更加平衡一些：

```
colormap(gcf,cmocean('balance'))
```



从 EOF 制作 SST 变量动图

在任何给定时间，可以通过绘制上面显示的模态 1 的地图，再乘以当时对应的主分量（矢量 pc(1，:)），来获得与 ENSO 相关的海面温度异常的快照。同样，您可以通过对所有 EOF 图进行求和，然后在每个时间上乘以其相应的主要成分，从而获得给定时间的全球海表温度异常情况的图片。这样，随着我们包含越来越多的可变性模式，我们可以构建越来越完整的 SST 异常影片。

例如，可以通过将每个模式的 eof 映射图相乘，再乘以该时间对应的 pc 值，来获得与指定时间的前三个变异模式相关的 SST 异常图。您可以像这样在 1990 年代手动进行求和：

```
% 影片的起止日期索引：
```

```
startind = find(t>=datenum('jan 1, 1990'),1,'first');

endind = find(t<=datenum('dec 31, 1999'),1,'last');

% 开始时前三种模态的 SST 异常图:
map = eof_maps(:,:,1)*pc(1,startind) + ... % 模态 1, 1990 年 1 月

      eof_maps(:,:,2)*pc(2,startind) + ... % 模态 2, 1990 年 1 月

      eof_maps(:,:,3)*pc(3,startind);      % 模态 3, 1990 年 1 月
```

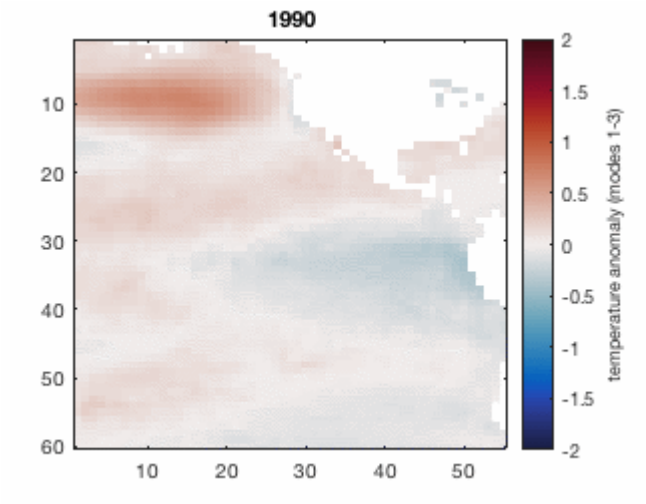
但是，这比必要的更为复杂，因为 reof 函数旨在为我们进行求和。为前三个模态制作一个 sst 异常时间序列，如下所示：

```
sst_f = reof(eof_maps,pc,1:3);
```

现在将 1990 年 1 月的地图绘制为影片的第一帧：

```
ind_1990s = 481:3:600; % (减少 1/3 的值以缩小 gif 大小)

figure

h = imagescn(sst_f(:,:,ind_1990s(1)));

caxis([-2 2])

cmocean balance

cb = colorbar;

ylabel(cb,'temperature anomaly (modes 1-3)')

title(datestr(t(ind_1990s(1)),'yyyy'))

gif('SSTs_1990s.gif','frame',gcf) % 写入第一帧

for k = 2:length(ind_1990s)

   h.CData = sst_f(:,:,ind_1990s(k));

   title(datestr(t(ind_1990s(k)),'yyyy'))

   gif % 在 gif 添加这一帧
```

```
end
```



您可能注意到的第一件事是 1990 年代 SST 异常时间序列受 ENSO 支配，并检查 1997-1998 信号！难怪那一年的新闻中有如此热门的话题。尽管如此，重要的是要记住，上面的影片不是对 SST 异常的完整重建，而只是前三种模态，它们共同说明了

```
sum(expv(1:3))
```

```
ans =

   51.8850
```

...刚好超过 SST 数据集总方差的一半。要重建绝对温度场而不仅仅是前三个模式中的异常，您需要包括所有 EOF 图，还必须重新添加平均 SST 图，趋势和季节周期

所解释的方差的注释。

以上，我们解决了包含 802 个时间步长的时间序列的所有 802 个模态。 这些模态一起解释了 sst 数据集中 100%的总方差。这意味着我们可以将所有这 802 个模态相加，再乘以它们的主成分，从而完全重建时间序列，而不会丢失任何信息。来看，所解释的方差之和...

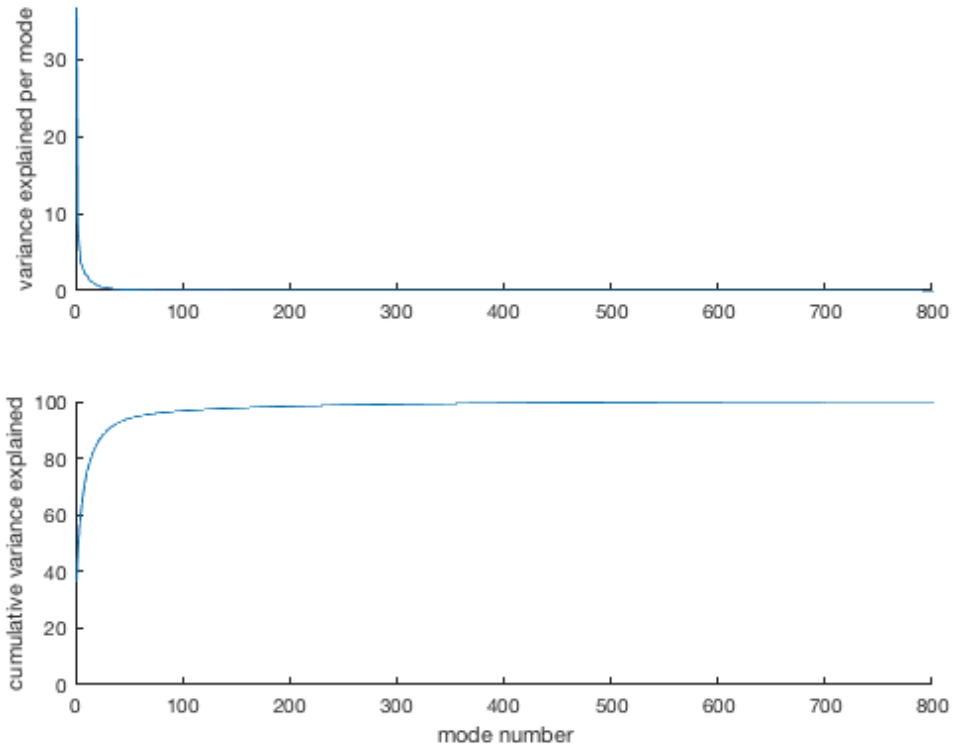
```
sum(expv)
```

```
ans =

  100.0000
```

...是 100%。 对于此数据集以及具有自然可变性模态的大多数其他现实世界数据集，前几个模态将携带大部分信息，其他所有内容都可能会被忽略。有时，将解释的方差视为模态编号的函数，或者也通常是累积的解释方差，是很有见地的。让我们一起绘制：.

```
figure
```

```
subplot(2,1,1)

plot(expv)

axis([0 802 0 37])

box off
ylabel 'variance explained per mode'

subplot(2,1,2)
plot(cumsum(expv))
axis([0 802 0 100])
ylabel 'cumulative variance explained'
xlabel 'mode number'
box off
```



上面的图表明，前三个模态占去趋势化，去季节化的 SST 数据集中方差的一半以上，其他所有内容都仅添加了微小的增量改进，使我们离重构完整的原始数据集有点近。

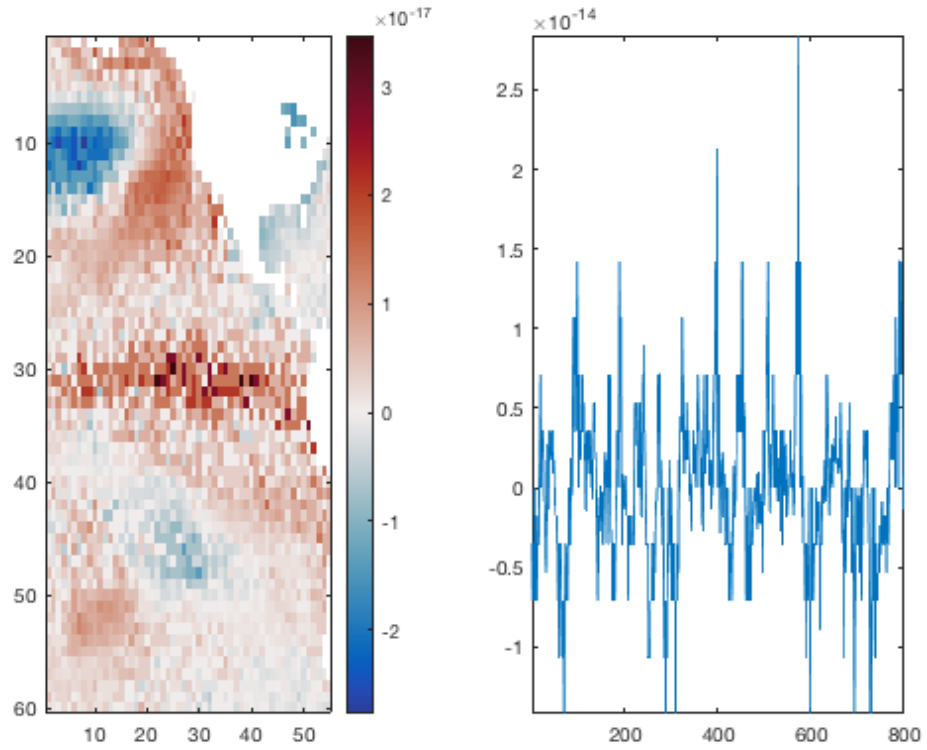
作为一般的经验法则，我倾向于认为超出第三或第五种模态的任何内容都是某种 Rorschach 测试，可能会诱使您相信存在有意义的信号，但是在尝试解释这些 N 阶模态时要格外小心-即使在这个简单的数据集中，也有 802 种模态，这意味着其中可能至少有一种模态似乎可以解释您可能拥有的任何疯狂理论。换句话说，在解释 EOF 时要谨防 p-hacking 的可能性。

在上面的示例中，我们解决的大多数 802 模态都对整体 SST 的变化贡献很小，并且在物理上可能毫无意义，因此使用这么多的计算机功能来解决所有这些模态没有任何意义。如果您的数据集非常大，它将更快并且将使用更少的内存来解决您所需的模态。只有一个小陷阱...

```
% 解出所有模态:
[eof_maps_all,pc_all,expv_all] = eof(sst);

% 只解出前 10 个模态:
[eof_maps_10,pc_10,expv_10] = eof(sst,10);
```

我们刚刚解决了前 10 个模态，我们看到 EOF 映射和主成分时间序列与通过解决所有 802 个模态获得的解决方案一致：

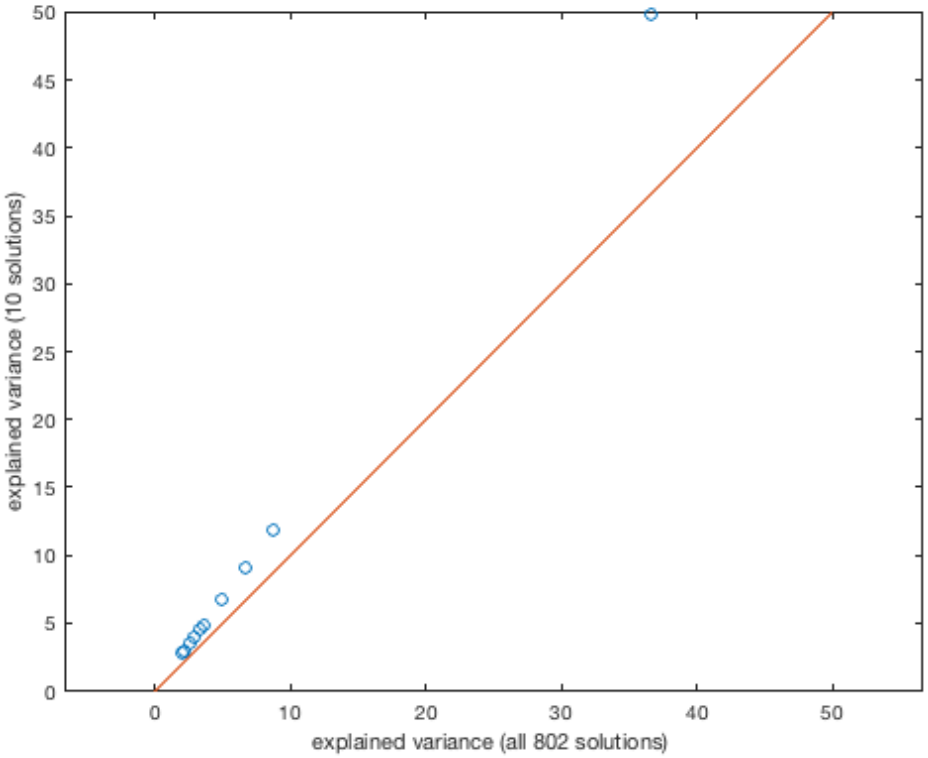
```
figure

subplot(1,2,1)

imagescn(eof_maps_all(:,:,1)-eof_maps_10(:,:,1))

colorbar

cmocean('balance','pivot')

subplot(1,2,2)
plot(pc_all(1,:)-pc_10(1,:))
axis tight
```



上面的两个图均显示纯数字噪声。它们有一些明显的结构，但请检查左侧的色轴（10^-17）和右侧的 y 轴（10^-14）的大小。舍入误差是由 Matlab 将其处理的数字数字化的事实造成的，舍入误差的大小跟踪要舍入的数字的大小是很常见的。

因此，尽管存在一些数字噪声，但求解所有 802 个模态时的解决方案与仅求解 10 个模态时的解决方案相同。牛 X！直到在我们研究解释的差异之前：

```
figure


plot(expv_all(1:10),expv_10,'o')
hold on
plot([0 50],[0 50]) % 直线（两个值相等）
axis equal
xlabel 'explained variance (all 802 solutions')
ylabel 'explained variance (10 solutions')
```



解释的方差不匹配，因为 expv 提供了在给定运行中求解的模态方差百分比的度量。 也就是说，如果我们针对所有 802 个模态进行求解，则前 10 个模态所解释的方差之和将不等于 100%：

```
sum(expv_all(1:10))
```

```
ans =

   73.4559
```

但是，由于我们只解决了前 10 个模态才能获得 expv_10，因此它们的和等于 100%

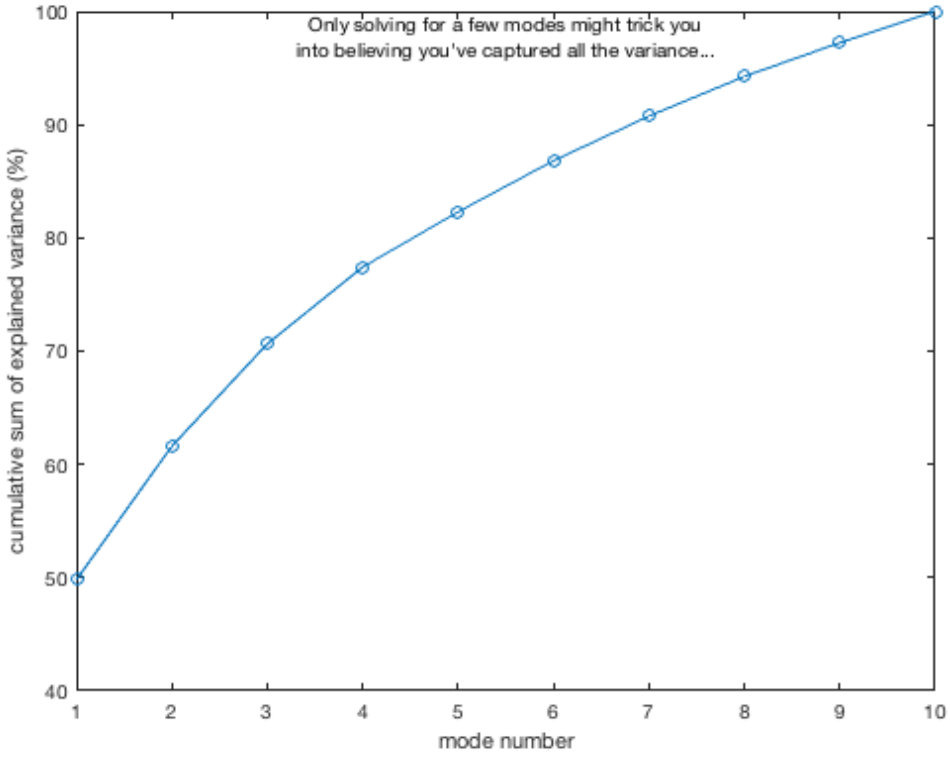
```
sum(expv_10(1:10))
```

```
ans =

  100.0000
```

因为这 10 个模态解释了与这 10 个模态相关的 100%的方差。我知道那是某种循环逻辑，所以如果您只求解几种模态，那么解释方差的值有什么用？也许他们根本没有那么好。最好的办法是使用较长的时间序列，求解所有模态，然后查看解释的方差以查看它们的渐近线。
如果由于计算上的限制只能求解几种模态，则仍然可以通过绘制所解释的各种模态的解释方差的累积和来了解一下是否获得了全貌：

```
figure
```

```
plot(cumsum(expv_10),'o-')
xlabel 'mode number'
ylabel 'cumulative sum of explained variance (%)'
ntitle({'Only solving for a few modes might trick you';...
   'into believing you''ve captured all the variance...'})
```



当然，该图的最终值是 100%（对于这个数据集，我们知道这实际上意味着约 73.5%的 100%），因此尽管我们可以清楚地看到它，但100%的值并不能告诉我们太多 在达到100%之前没有达到渐近线的大部分，这说明故事还有更多...

抱歉，我目前没有针对该问题的更整洁的解决方案，但是希望这个小小的解释可以更好地了解 eof 函数可以告诉您什么以及不能告诉您什么。

我是怎么获得样本数据的

上面显示的示例数据集来自 Hadley Center HadISST，该文件位于 here (Rayner et al., 2003)，总大小超过 200 MB。如果要在世界的不同区域执行相同类型的分析，则可以下载 HadISST_sst.nc 数据集，并将其导入到 Matlab 中，如下所示。对数据集进行下采样或子集设置取决于您：

```
% 载入完整的 SST 数据集:

lat = double(ncread('HadISST_sst.nc','latitude'));

lon = double(ncread('HadISST_sst.nc','longitude'));

t = double(ncread('HadISST_sst.nc','time')+datenum(1870,1,0));
```

```
sst = ncread('HadISST_sst.nc','sst');

% 为了将样本数据集的大小缩小到四分之一，我将粗略地缩减采样到每个其他网格点:

sst = sst(1:2:end,1:2:end,:);

lat = lat(1:2:end);

lon = lon(1:2:end);

% 为了进一步减小尺寸，我剪裁了一些经度和纬度，并仅保留了 1950 年以后的数据:

rows = lon<-70;

lon = lon(rows);

cols = lat>=-60 & lat<=60;

lat = lat(cols);

times = t>=datenum('jan 1, 1950');

t = t(times);

sst = sst(rows,cols,times);

sst(sst<-50) = NaN;

% 我发现按纬度×经度×时间重新排列更容易:

sst = permute(sst,[2 1 3]);

%保存样本数据:

save('PacOcean.mat','lat','lon','t','sst')
```

参考文献

Messié, Monique, and Francisco Chavez. "Global modes of sea surface temperature variability in relation to regional climate indices." Journal of Climate 24.16 (2011): 4314-4331. doi:10.1175/2011JCLI3941.1.

Rayner, N. A., Parker, D. E., Horton, E. B., Folland, C. K., Alexander, L. V., Rowell, D. P., Kent, E. C., Kaplan, A. (2003). Global analyses of sea surface temperature, sea ice, and night marine air

temperature since the late nineteenth century. J. Geophys. Res.Vol. 108, No. D14, 4407 doi:10.1029/2002JD002670.

Thyng, K.M., C.A. Greene, R.D. Hetland, H.M. Zimmerle, and S.F. DiMarco. 2016. True colors of oceanography: Guidelines for effective and accurate colormap selection. Oceanography 29(3):9-13, doi:10.5670/oceanog.2016.66.

作者简介

---

eof 函数是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 2017 年 1 月写的，但是非常依赖 Guillame MAZE 的 PCATool 中的 caleof 函数贡献。本教程由 Chad Greene 在 Kaustubh Thirumalai 的帮助下编写。

reof 文档

reof 根据指定的 EOF 模态重建 eof 异常的时间序列。
另请参见: eof.

语法

```
A = reof(eof_maps,pc,modes)
```

说明

A = reof(eof_maps,pc,modes) 从特征模态映射 eof_maps 和主成分时间序列 pc 重建指定模态的网格化时间序列，它们都是 eof 函数的输出。

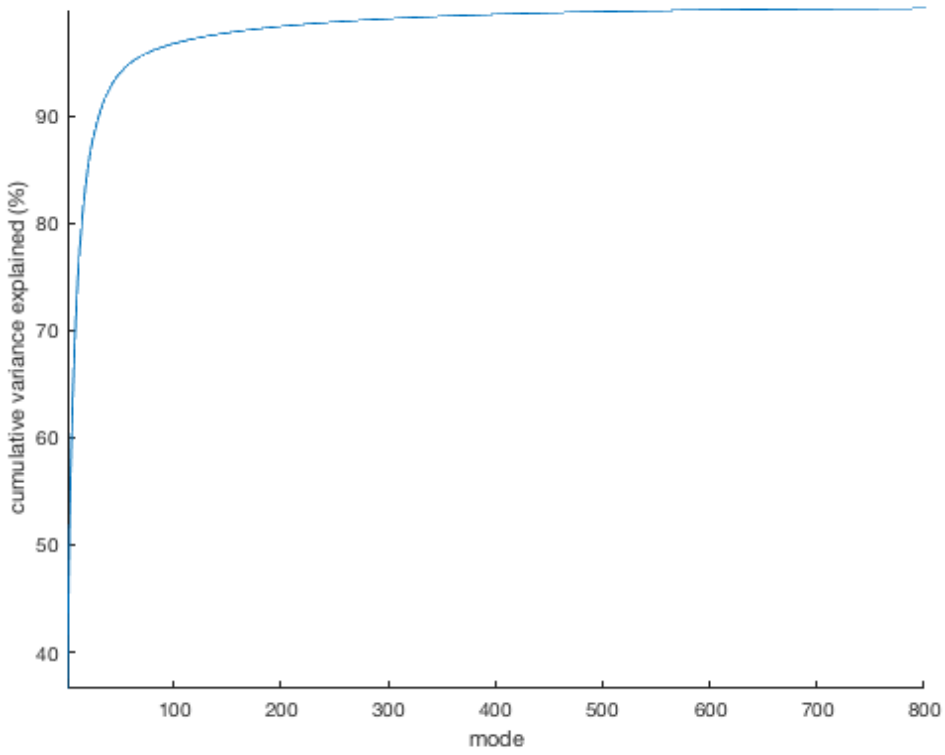
示例

假设我们有一些大的网格数据集，并且与前 75%的方差相关联的模态是可取的，但是我们假设（在此示例中）构成所有 25%的方差的模态都只是噪声。我们想要过滤掉所有的噪声，并且可以使用 reof 进行这种过滤。

```
load pacific_sst

sst = deseason(detrend3(sst,t),t);

[eof_maps,pc,expv] = eof(sst);
```

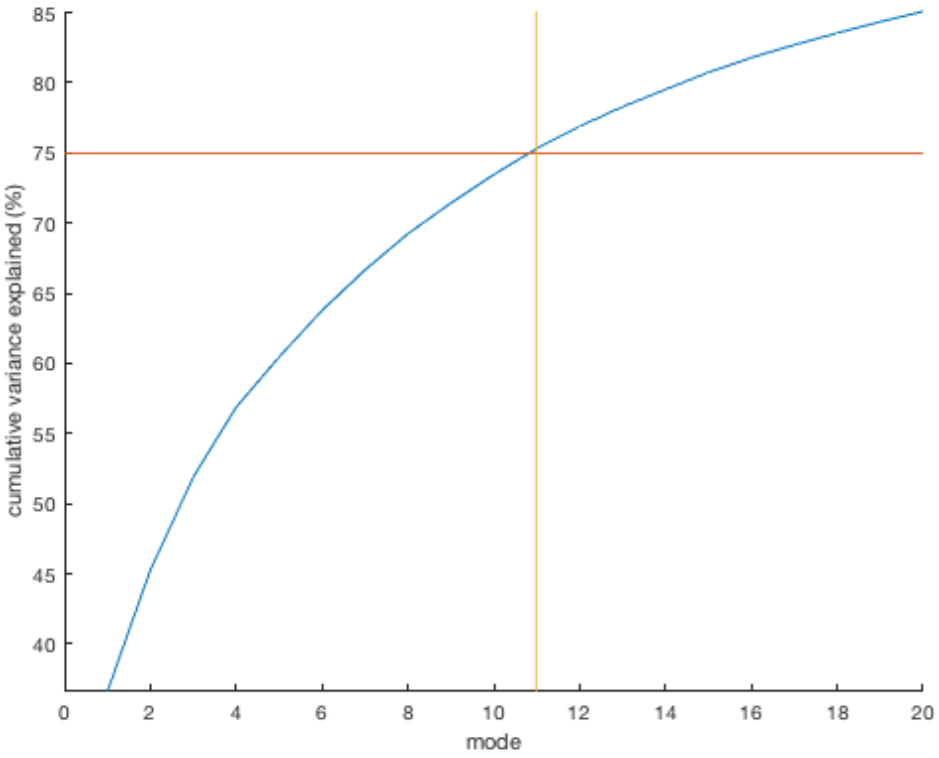
我们说过要保留第一个模态，因为它们是最重要的，但是要解释 75%的数据集方差需要多少个模态？为此，我们必须根据模态编号来绘制解释的累积方差：

```
figure

plot(cumsum(expv))

axis tight
box off
xlabel mode
ylabel 'cumulative variance explained (%)'
```



如您所见，仅需 802 个模态中的几个即可解释 sst 数据集的 75%，并且我们假设这意味着所有其他模态都只是噪声。放大，看起来只需 11 种模态即可解释 75%的方差：

```
xlim([0 20])

hline(75) % 75% 方差解释
vline(11) % 对应于大约模态 11
```



让我们检查。前 11 个模态解释的方差之和为

```
sum(expv(1:11))
```

```
ans =

   75.3165
```

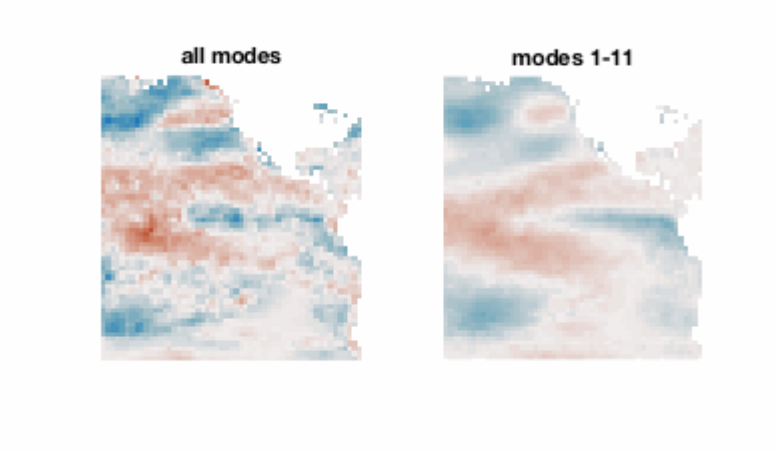
完美，75%。因此，让我们“过滤掉”带有 reof 的最后 25%的噪声方差。只需为其提供我们先前计算的 eof_maps 和 pc 数组，然后指定模态 1:11，如下所示：

```
sst_f = reof(eof_maps,pc,1:11);
```

现在，我们可以制作一个并排的动画，在过滤后的动画旁边显示原始（去趋势和去季节 detrended and deseasoned）时间序列。首先绘制第一个帧，然后使用 gif 函数保存，然后遍历其余帧，更新数据并使用 gif 保存每个帧。
为了保持较小的文件大小，我只绘制 200 至 300 帧（对应于 1966 年至 1974 年），并且每隔一个月绘制一次。

```
figure

subplot(1,2,1)

h = imagescn(lon,lat,sst(:,:,200));
```

```
caxis([-3 3])

cmocean bal

title 'all modes'

axis image off

subplot(1,2,2)

h_f = imagescn(lon,lat,sst_f(:,:,200));

caxis([-3 3])

cmocean bal

title 'modes 1-11'

axis image off

%保存第一帧:

gif('eof_filtering.gif','frame',gcf,'nodither')

for k = 202:2:300

   h.CData = sst(:,:,k);     % 更新原始数据

   h_f.CData = sst_f(:,:,k); % 更新过滤数据

   gif                       %保存这一帧

end
```



您可以看到，在上面的动画中，仅保留前 11 个模式可以保留大规模、大幅度的可变性，但可以消除较小规模的噪声。

作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

corr3 文档

corr3 计算时间序列和三维数据集的线性或秩相关。
另请参见: xcorr3 和 corrcoef.

语法

```
r = corr3(X,y)
r = corr3(...,'detrend')
r = corr3(...,'Name',Value)
[r,p] = corr3(...)
```

说明

r = corr3(X,y) 返回输入矩阵 X 中每对列之间的成对线性相关系数矩阵。
r = corr3(...,'detrend') 从 X 和 y 去除线性趋势。、
r = corr3(...,'Name',Value) 除了使用先前语法中的输入自变量之外，还使用一个或多个名称-值对自变量指定选项。 例如，'Type','Kendall'指定计算 Kendall 的 tau 相关系数。
[r,p] = corr3(...) 还返回 pval，这是一个 p 值矩阵，用于相对于非零相关性的另一假设检验无相关性的假设。

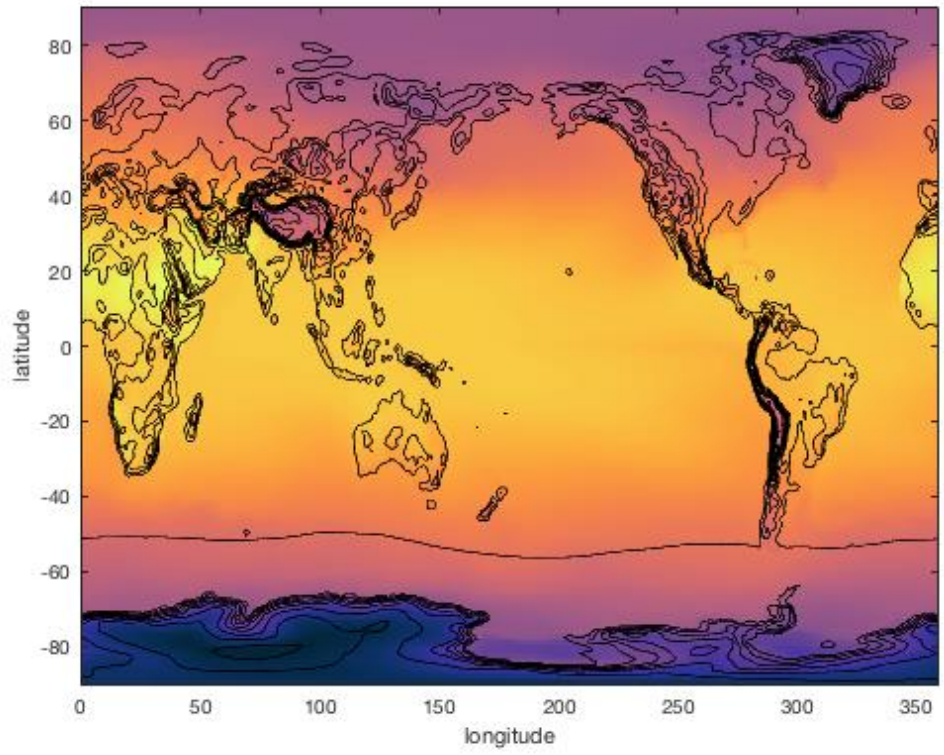
示例:

在 1850 年代，英国皇家海军上将罗伯特·菲茨罗伊（Admiral Robert FitzRoy）决定，他将毕生致力于挽救海上水手和渔民的生命。他选择的武器：气压计（ His weapon of choice: the barometer. ）。因此，开始了天气预报的研究以及气压变化与天气系统之间的关系。
那么，地表气压和温度到底有什么关系呢？ 我们可以使用 2017 年以来的每月网格数据来探索与 corr3 的关系：

```
filename = 'ERA_Interim_2017.nc';
P = ncread(filename,'sp'); % 表面气压
T = ncread(filename,'t2m'); % 2m 温度
precip = ncread(filename,'tp'); % 总降水量
lat = double(ncread(filename,'latitude'));
lon = double(ncread(filename,'longitude'));
```

首先，通过绘制 2017 年的平均地表气压和温度场来快速获得。请注意，在绘制时，我们使用'来对网格进行转置，以使行与纬度相对应，而列与经度相对应：

```
figure

imagescn(lon,lat,mean(T,3)')

cmocean thermal
hold on
caxis([221 311]) % 开尔文颜色轴限制
contour(lon,lat,mean(P,3)','k')
xlabel 'longitude'
ylabel 'latitude'
```



上面表明，地表气压似乎主要与地表高度有关，这在 air_pressure 函数的文档中有更充分的关系。但是现在我们不在乎平均温度或平均气压。相反，我们想看看每个变量的时间变异性如何相互变化。
让我们这样做，但是首先更新网格以将本初子午线置于中间：

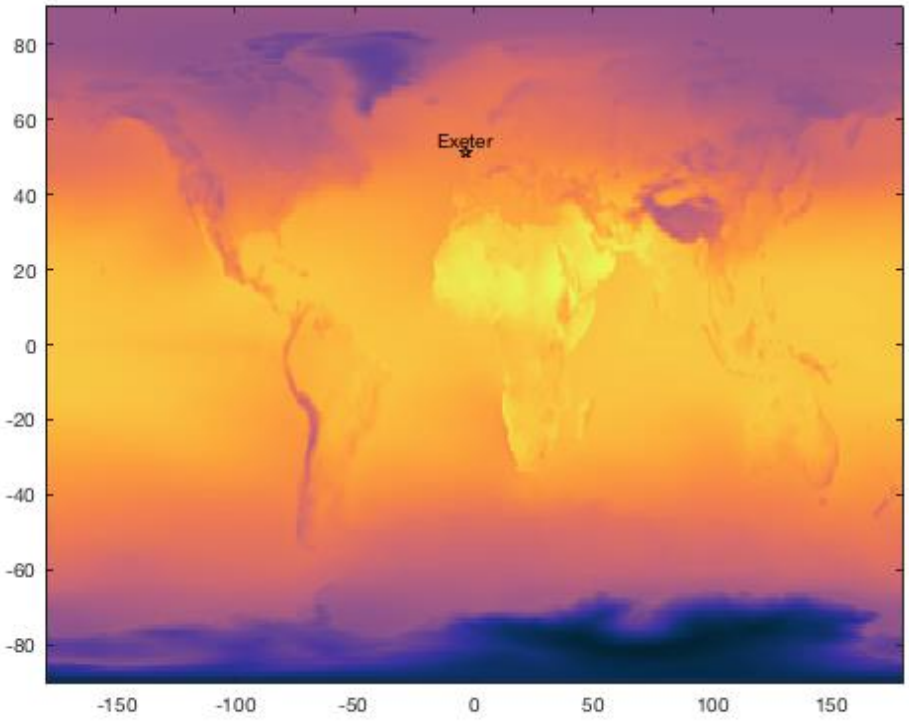
```
[lat,lon,T,P,precip] = recenter(lat,lon,T,P,precip);
```

FitzRoy 海军上将在英国埃克塞特（Exeter）成立了气象办公室，下面让我们看一下气象办公室的地表气压测量值与世界各地的地表温度之间的关系。

首先，使用 Near1 获取最接近埃克塞特（50.7N，3.5W）的网格单元的行和列索引：

```
row = near1(lon,-3.5);

col = near1(lat,50.7);



figure

imagescn(lon,lat,mean(T,3)')

cmocean thermal
hold on
```

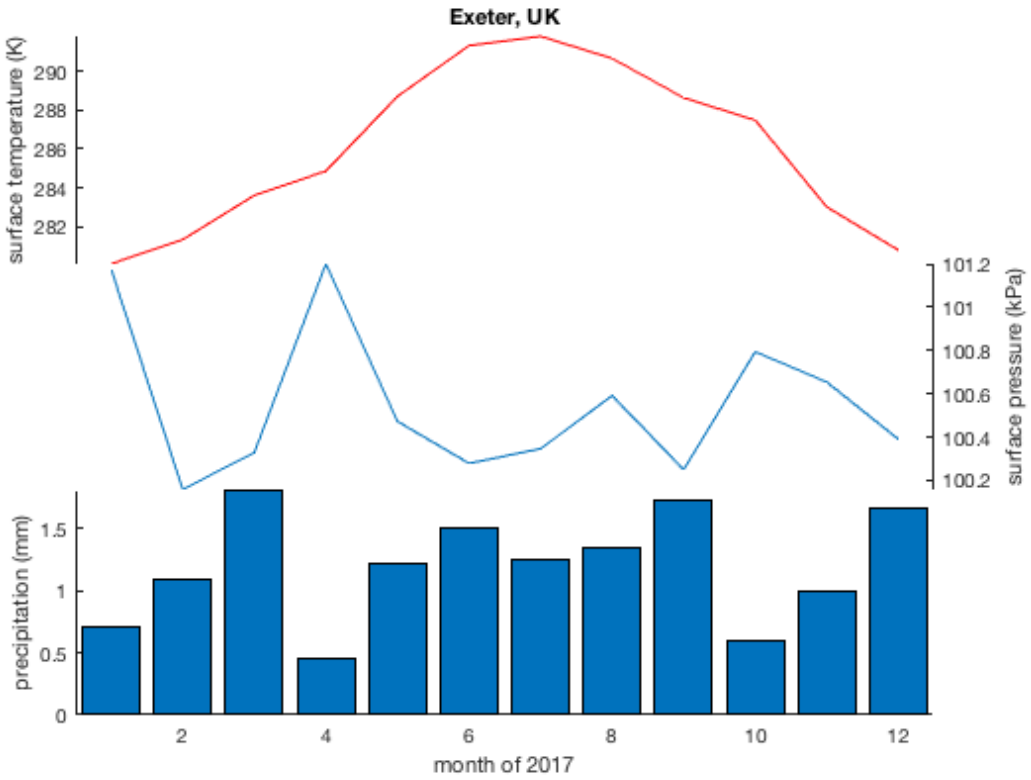
```
plot(lon(row),lat(col),'kp')
text(lon(row),lat(col),'Exeter','vert','bot','horiz','center')
```



现在，可以像这样获得最接近埃克塞特的网格单元的温度和气压的一维数组：

```
T_exeter = squeeze(T(row,col,:));

P_exeter = squeeze(P(row,col,:));

precip_exeter = squeeze(precip(row,col,:));



figure

subsubplot(3,1,1)

plot(1:12,T_exeter,'r')
axis tight
box off
xlim([0.5 12.5])
ylabel 'surface temperature (K)'
title 'Exeter, UK'
```

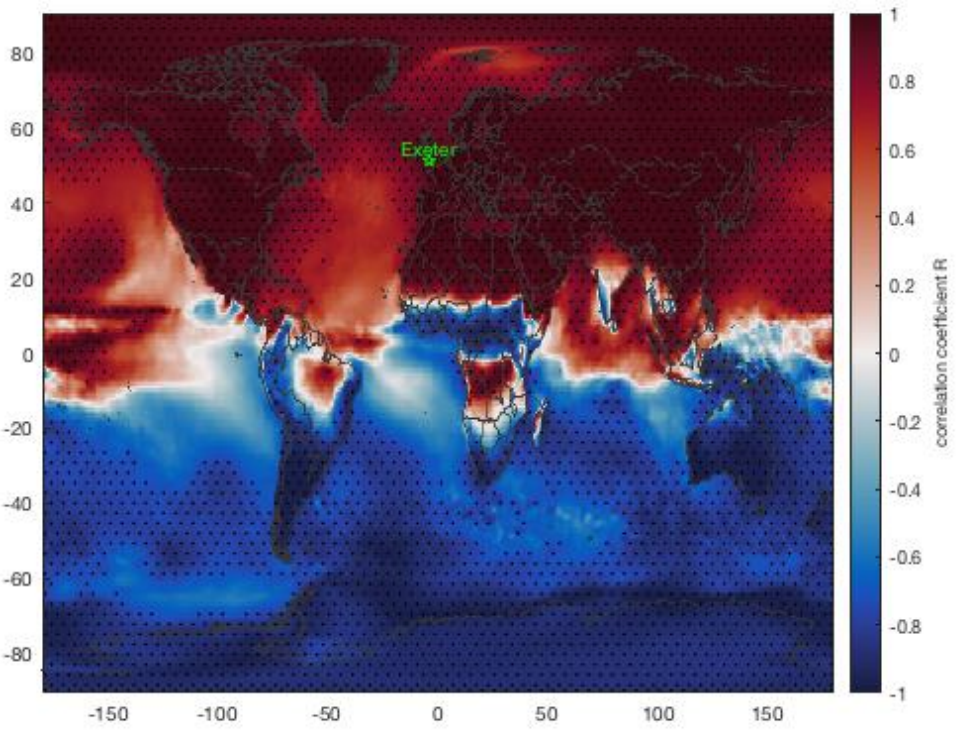
```
subsubplot(3,1,2)
plot(1:12,P_exeter/1000)
axis tight
box off
xlim([0.5 12.5])
ylabel 'surface pressure (kPa)'
set(gca,'yaxislocation','right')

subsubplot(3,1,3)
bar(1:12,precip_exeter*1000)
axis tight
box off
xlim([0.5 12.5])
ylabel 'precipitation (mm)'

xlabel 'month of 2017'
```



首先，埃克塞特的表面温度与世界各地的温度变化相比如何？（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
% 温度网格与埃克塞特温度之间的相关性:
[R,P] = corr3(T,T_exeter);

figure
```

```
imagescn(lon,lat,R')
caxis([-1 1])
cmocean balance
cb = colorbar;
ylabel(cb,'correlation coefficient R')

hold on
borders('countries','color',rgb('dark gray'),'center',180)
stipple(lon,lat,(P<0.01)','markersize',4,'density',150) % 重要区域
plot(lon(row),lat(col),'gp')
text(lon(row),lat(col),'Exeter','color','g',...
   'vert','bot','horiz','center')
```

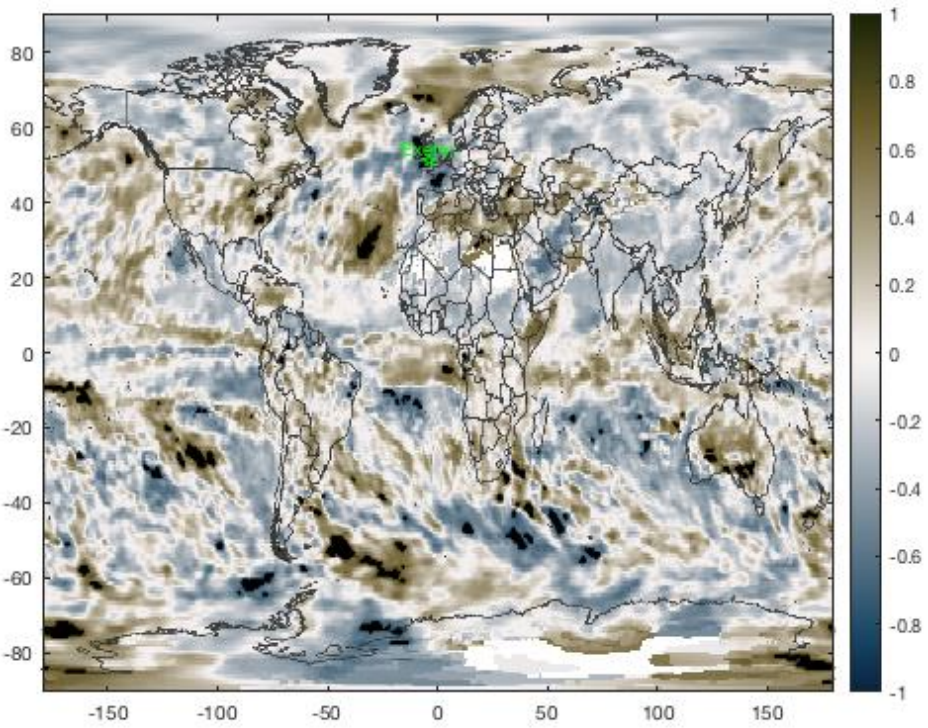


毫不奇怪，当埃克塞特变暖时，北半球其余大部分地区也会变暖，反之亦然。在赤道附近以及季节性变化较低的地方，这种关系不太明显。这是因为埃克塞特（Exeter）自 2017 年以来的月度数据中的温度变化主要受季节周期的影响。

埃克塞特的全球降雨量与地表气压有何关系？（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

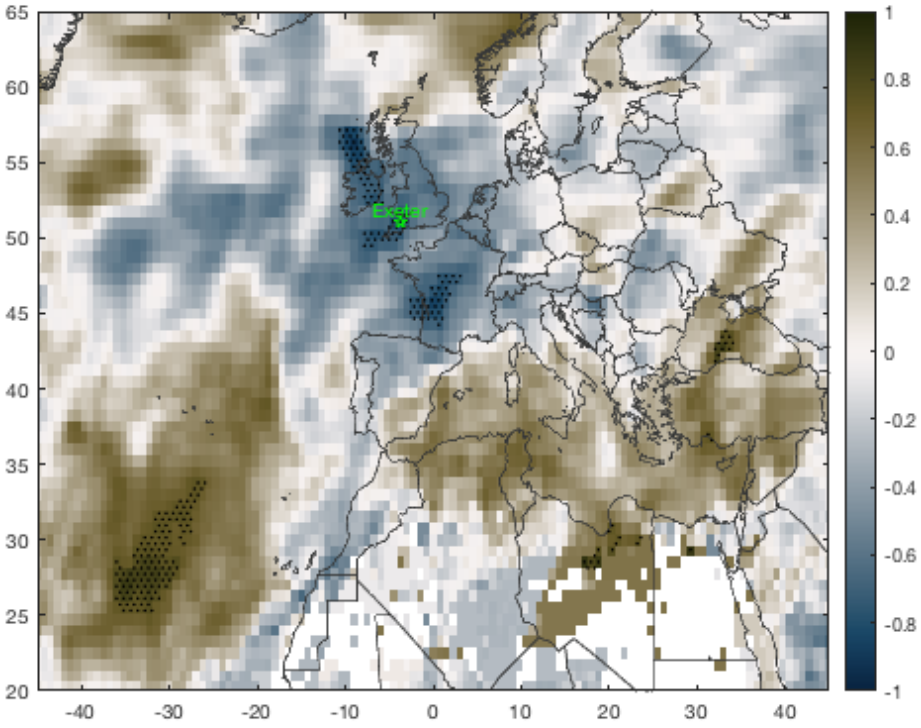
```
[R,P] = corr3(precip,P_exeter,'detrend');

figure
imagescn(lon,lat,R')
caxis([-1 1])
```

```
cmocean diff
colorbar

hold on
borders('countries','color',rgb('dark gray'),'center',180)
stipple(lon,lat,(P<0.01)','color','k',...
   'markersize',4,'density',1000)
plot(lon(row),lat(col),'gp')
text(lon(row),lat(col),'Exeter','color','g',...
   'vert','bot','horiz','center')
```



上图中没有点状，表明埃克塞特的表面气压与世界上大多数地方的降水没有显着关系。但是，如果我们放大欧洲，就会发现埃克塞特的地面气压低时，与整个英格兰和法国的高降水量有关。

```
axis([-45 45 20 65])
```

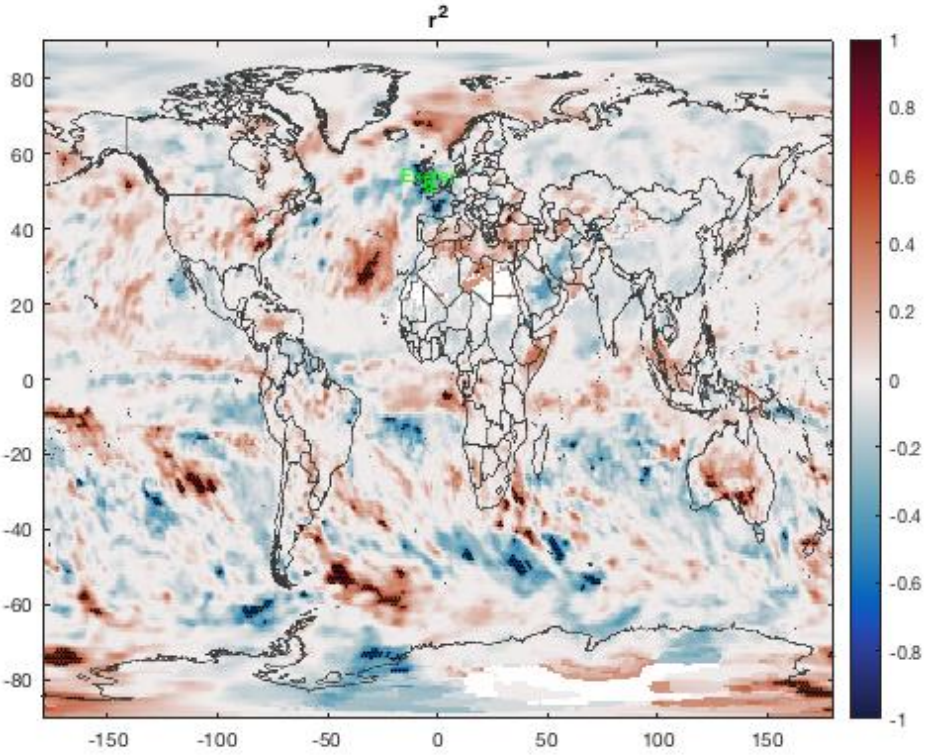


显示 R 平方

有时人们更喜欢使用 R 平方而不是相关系数 R，因为 R 平方描述了 X 的方差百分比，由与 y 的关系解释。但是，当您计算 R 平方时，您将丢失符号，因此请确保再次将符号乘以如下所示：（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
Rsq = R.^2 .* sign(R);



figure

imagescn(lon,lat,Rsq')

caxis([-1 1])

cmocean balance
colorbar

hold on
borders('countries','color',rgb('dark gray'),'center',180)
stipple(lon,lat,(P<0.01)','color','k',...
   'markersize',4,'density',500)
```

```
plot(lon(row),lat(col),'gp')
text(lon(row),lat(col),'Exeter','color','g',...
   'vert','bot','horiz','center')
title 'r^2'
```



作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

xcorr3 文档

xcorr3 函数给出了三维时空数据集的单元网格与参考时间序列之间的相关系数映射。
另请参见: corr3 和 xcov3.

语法

```
r = xcorr3(A,ref)
r = xcorr3(...,'detrend')
r = xcorr3(...,'maxlag',maxlag)
r = xcorr3(...,'mask',mask)
[r,rmax,lags] = xcorr3(...)
```

说明

r = xcorr3(A,ref) 给出二维相关映射 r，其维数对应于 A 的维 1 和 2。三维矩阵 A 假定具有空间维 1 和 3，可以对应于 x 和 y，lat 和 lon，lon 和 lat 等。假定 A 的第三维对应于时间。数组 ref 是一个时间序列参考信号，您正在与之比较 A 的每个网格单元。ref 的长度必须与 A 的三维尺寸匹配。
r = xcorr3(...,'detrend') 在计算相关性之前从每个时间序列中除去均值和线性趋势。建议将其用于 A 中的数据值或 ref 中的值范围不以零为中心的任何类型的分析。
r = xcorr3(...,'maxlag',maxlag) 将最大滞后指定为时间步长的整数标量。如果指定 maxlag，则返回的互相关序列的范围是-maxlag 到 maxlag。默认 maxlag 为 N-1 个时间步长。
r = xcorr3(...,'mask',mask) 仅对尺寸与 A 的前两个（空间）尺寸相对应的掩膜中的真实网格单元执行分析。如果 A 大，应用掩膜的选项旨在最大程度地减少处理时间。默认情况下，A 中的任何 NaN 值都会将默认掩膜中的相应网格单元设置为 false。
[r,rmax,lags] = xcorr3(...) 返回零相位相关系数 r，最大相关系数 rmax 和对应于最大相关的时滞。负滞后值表示本地时间序列在参考信号*之后*发生。正滞后表示局部现象领先于参考信号。

示例 1：四种人工信号的比较

考虑具有 10,000 个时间步长的简单 2x2 网格。您想查看哪些网格单元随某个参考信号 y 随时间变化:

```
% 一个时间矩阵:
t = 1:10000;

% 加点相关信号:
y = sind(t);

% 构建 A:
A = nan(2,2,10000);
A(1,1,:) = 2*y;              % 2x 相关信号
A(1,2,:) = sind(t-43);       % 有 43 个时间步长的滞后
A(2,1,:) = randn(size(t)); % 高斯噪声
A(2,2,:) = -y;               % 完全异相
```

可视化 A 的这四个网格单元的工作并不容易，因此希望这会有所帮助：这是一种查看 A 中内容的空间方法：

```
A = [ 2x 相关信号          有 43 个时间步长的滞后;
      高斯噪声             完全异相    ]
```

现在我们可以看到 A 的每个网格单元格如何随 y（或不随 y）变化：

```
[r0,rmax,lags] = xcorr3(A,y);
```

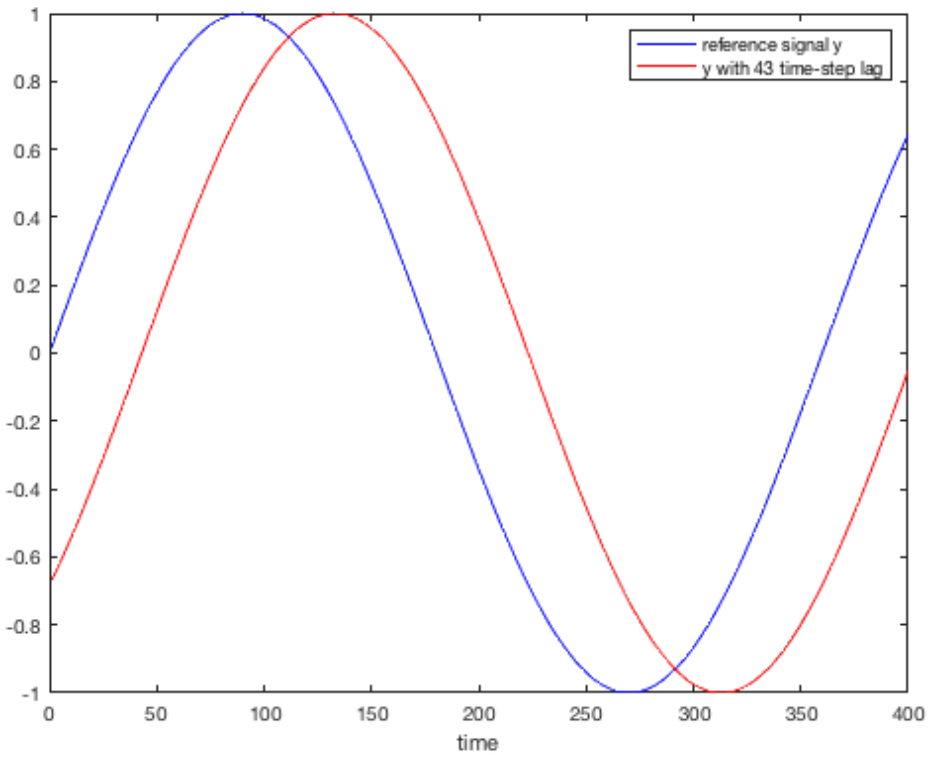
零相位相关性如下所示：

```
r0
```

```
r0 =

    1.00    0.73

   -0.01   -1.00
```

这准确地告诉了我们已经知道的。A 的左上方网格单元的值是 y 的两倍，因此它与 y 的相关性是完美的 1.0000。同样，A 的右下方网格单元具有一个我们定义为-y 的时间序列，因此它与 y 的相关性是-1.0000。A 的左下角仅包含噪声，因此它与 y 根本没有很好的相关性。棘手的是 A 的右上方网格单元，它具有与 y 相同的信号，但在时间上偏移了 43 个时间步长。以下是 y 和 sind（t-43）的摘要：

```
plot(t(1:400),y(1:400),'b')
hold on
plot(t(1:400),sind(t(1:400)-43),'r')
xlabel('time')
legend('reference signal y','y with 43 time-step lag')
```



在上面的图中，很明显 y 和 y 的时滞形式经常一起上升或下降，但是有时时间滞后意味着它们没有很好地相关，因此，如果我们不相关，考虑时间偏移，则相关系数为 0.73。但是，您还可以看到，如果将信号之一偏移 43 个时间步长，则两个信号将完全相关。实际上，如果我们看一下最大相关系数，那就很清楚了：

```
rmax
```

```
rmax =

          1.00          1.00

          0.02          0.98
```

我们看到，如果在时间上偏移 A 值，则 A 的右上方网格单元将与相关系数为 1（或接近 1）的参考时间序列匹配，这意味着完美匹配。信号最适合以下延迟

```
lags
```

```
lags =

             0        -43.00

       -617.00       -180.00
```

再一次，这准确地告诉了我们已经知道的。A 的左上方网格像元与参考时间序列完全匹配，且延迟为零；我们有意对 A 的右上方网格单元应用了 43 个时间步长滞后；左下方网格单元永远不会与参考信号很好地关联，因此其滞后值毫无意义，A 的右下方网格单元会以 180 个时间步长的偏移量匹配参考信号（因为在此示例中，偏移量方便地对应于度）。

示例 2：海表温度

上面的示例旨在提供 xcorr3 的工作原理背后的理论知识，但并未捕获您将 xcorr3 应用于实际数据时可获得的时空模式的见解。因此，让我们看一下海表温度的再分析样本数据集：

```
load pacific_sst
whos lat lon t
```

```
  Name        Size            Bytes  Class     Attributes


  lat        60x1              480  double
  lon        55x1              440  double
  t         802x1             6416  double
```

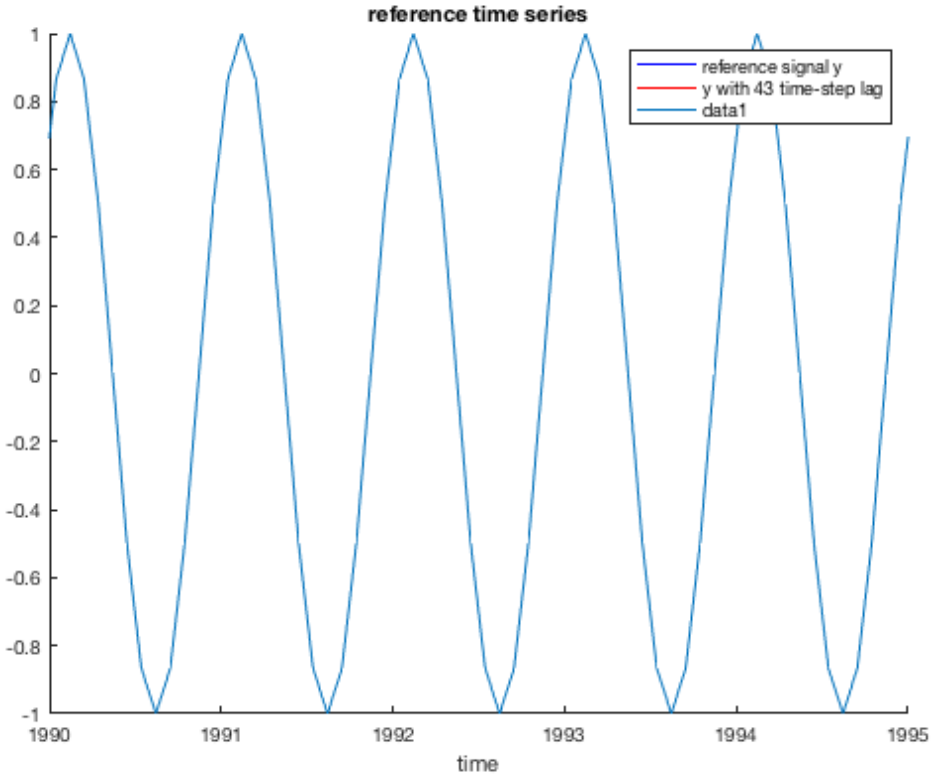
示例数据集包含一个具有以下分辨率的 sst 矩阵：

```
mean(diff(lat))

mean(diff(lon))

mean(diff(t))
```

```
ans =

         -2.00

ans =

          2.00

ans =

         30.44
```

也就是说，sst 是具有每月时间分辨率的 2x2 度网格。
我有一个假设，即海面温度随季节呈正弦变化。要将海面温度与正弦曲线进行比较，首先创建与一年中的月份相对应的参考正弦曲线：

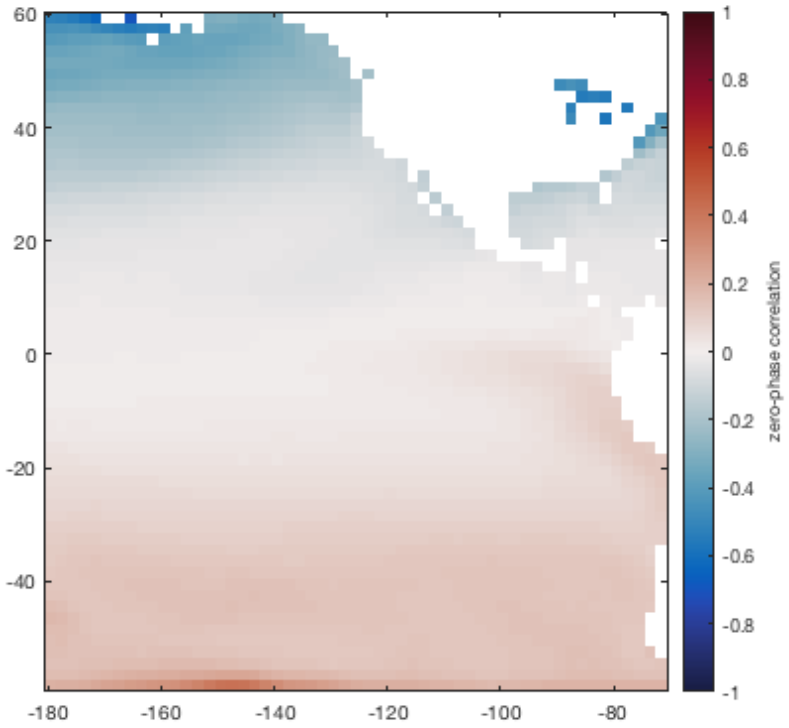
```
[~,month,~] = datevec(t);

ref = sin((month+1)*pi/6);



plot(t,ref)

xlim([datenum('jan 1, 1990') datenum('jan 1, 1995')])
box off
datetick('x','keeplimits')
title ' reference time series '
```



参考信号在每年的 2 月具有最大值。我的假设是，南半球的海面温度应与参考信号成正相关，这意味着南半球的海表温度应在 2 月达到最高，而在 8 月达到最低。北半球应显示完全相反的模式，每年八月最高，每年二月最低。

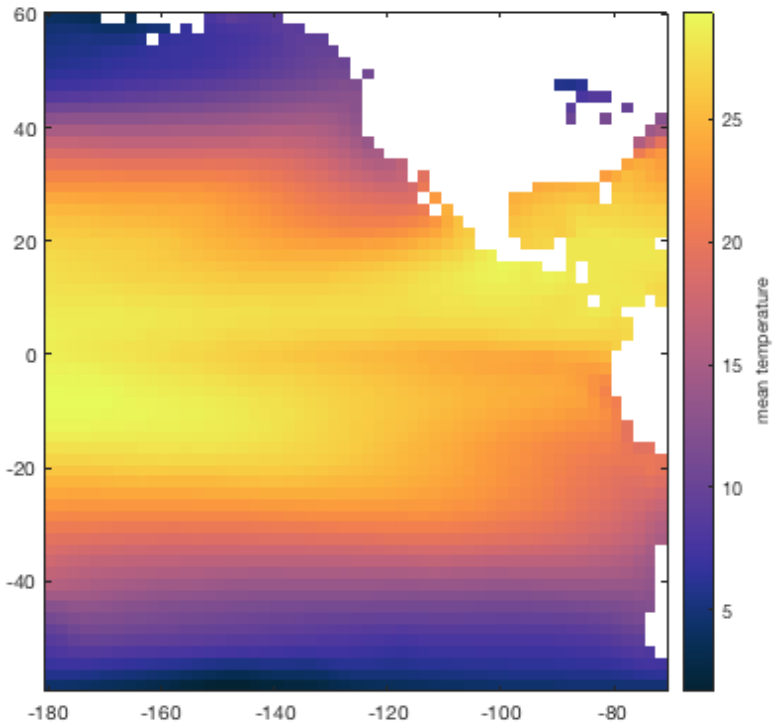
让我们看一下原始海面温度数据与参考正弦曲线之间的相关性。（下面我使用 imagesscn 来创建 NaC 值透明的 imagec 图，但是如果愿意，可以使用 imagesc 或 pcolor。我还使用了 cmocean 色图(Thyng et al., 2016)，这是发散的和感性上的统一。）

```
r0 = xcorr3(sst,ref);



figure

imagescn(lon,lat,r0)

axis xy image
cb = colorbar;
ylabel(cb,'zero-phase correlation')
caxis([-1 1])
cmocean balance
```



在上面的地图中，我们看到海面温度似乎与参考正弦曲线仅*略微*相关。但是我们已经疏忽了！我们忘记从 sst 数据集中删除平均值，因此与正弦曲线的相关性当然很弱。这是平均海面温度的地图，它污染了上面的分析：

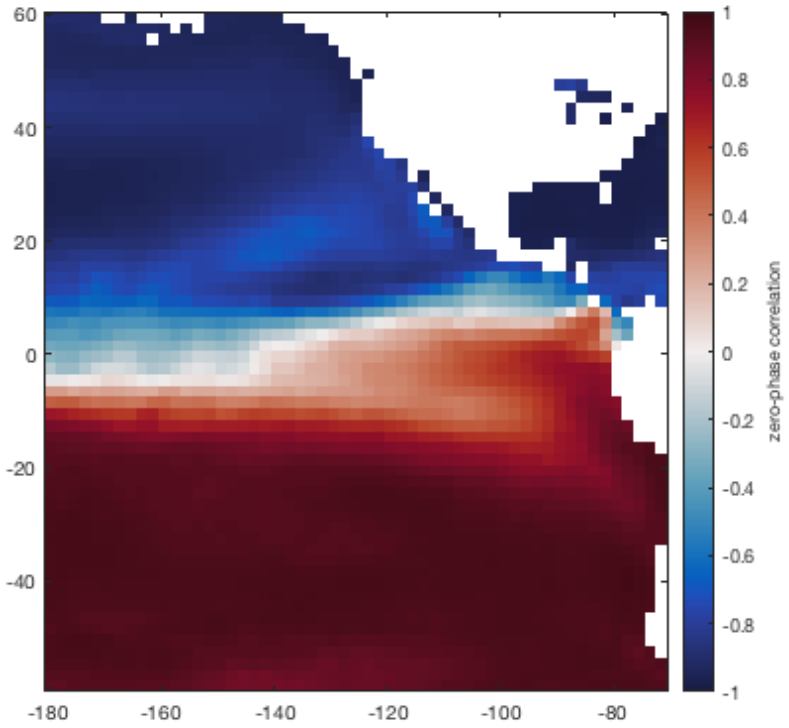
```
figure

imagescn(lon,lat,mean(sst,3))

axis xy image
cb = colorbar;
ylabel(cb,'mean temperature')
cmocean thermal
```



在使用 xcorr3 之前，您可以根据需要手动删除均值，也可以简单地使用'detrend'选项，这将为您删除均值。'detrend'选项还删除了长期（全球变暖）趋势，这很方便，因为当去除均值和长期趋势时，剩下的只是季节周期和年际变化。

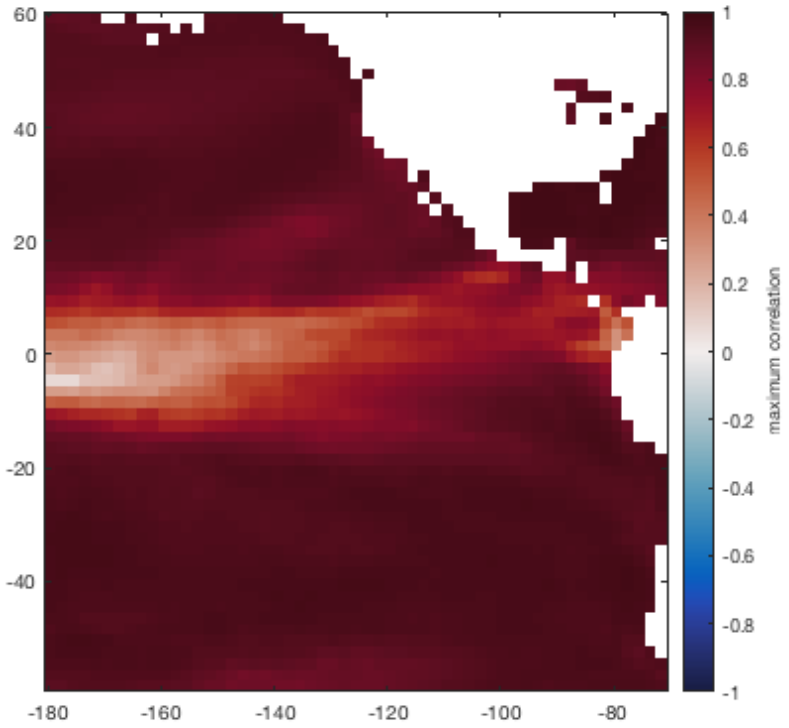
因此，让我们看一下海面温度*异常*（而不是绝对值）与参考正弦波的比较：

```
[r0,rmax,lags] = xcorr3(sst,ref,'detrend');

figure
imagescn(lon,lat,r0)
axis xy image
cb = colorbar;
ylabel(cb,'zero-phase correlation')
caxis([-1 1])
cmocean balance
```



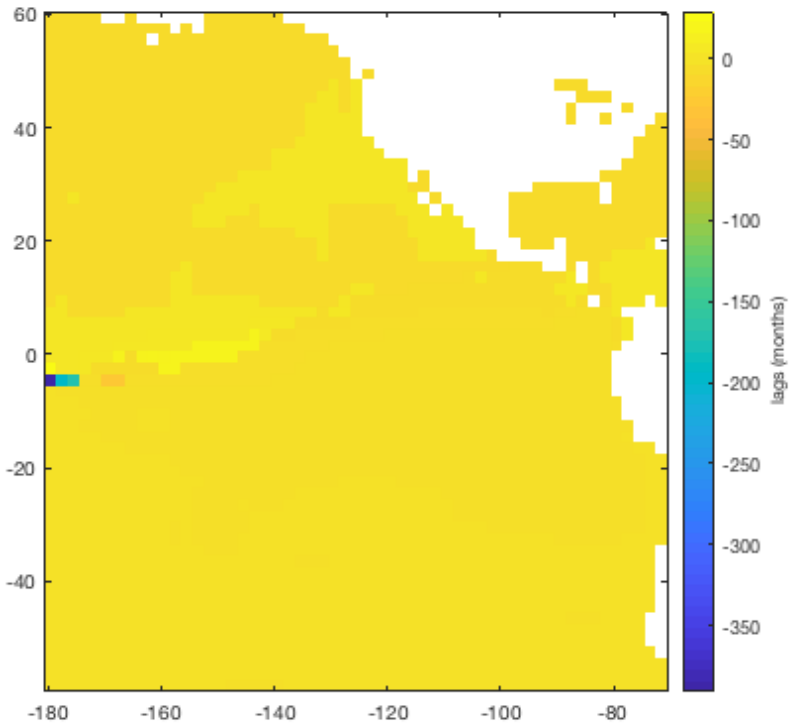
哇！现在，有一些明确的证据表明海表温度似乎随季节变化。正如预期的那样，在南半球与参考正弦波呈正相关，这意味着南半球的海面温度在 2 月前后达到最高，在 8 月左右达到最低。北半球与南半球几乎完全反相关。在赤道附近，与参考正弦曲线的相关性很小。但是也许我们为参考信号选择的相位与数据不完全匹配。如果我们随时间移动参考信号，那么每个网格单元与参考正弦波可达到的最佳匹配是什么？

```
figure

imagescn(lon,lat,rmax)

axis xy image
cb = colorbar;
ylabel(cb,'maximum correlation')
caxis([-1 1])
cmocean balance
```



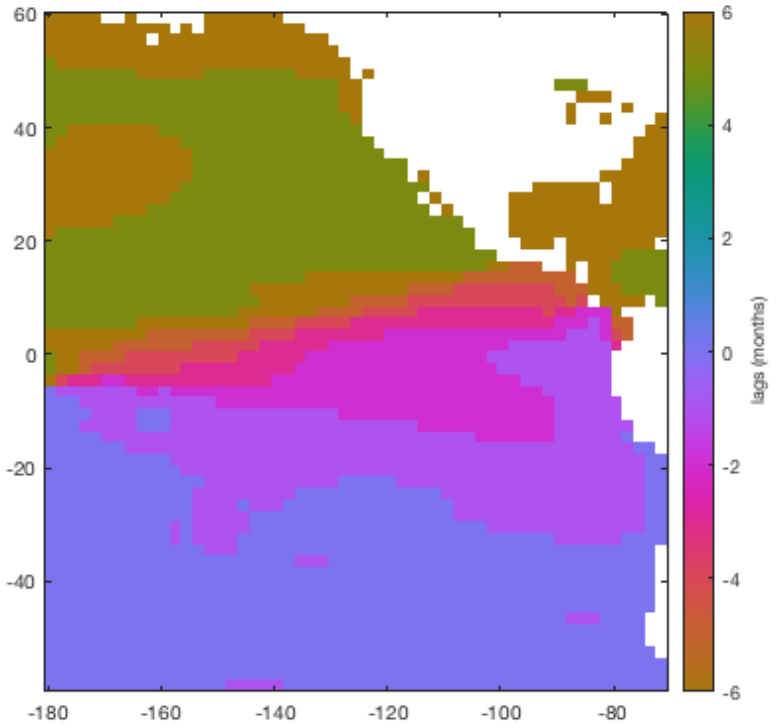
上面的地图向我们展示了几乎整个太平洋都有一些正弦变化。通过查看滞后，我们可以找到最匹配的正弦波的相位：

```
figure

imagescn(lon,lat,lags)

axis xy image
cb = colorbar;
ylabel(cb,'lags (months)')
```



上面的地图似乎是垃圾。（180W，4S）附近的网格单元需要大约 391 个月（32 年）的偏移量才能获得与正弦曲线的最佳匹配？再看一下最大相关图，您会发现（180W，4S）从未与正弦曲线很好地相关-它的海表温度似乎没有季节性周期。鉴于 6 个月的负偏移量与参考信号的 6 个月的正偏移量在数学上是相同的，因此我们实际上应该将最大滞后时间限制为 6 个时间步长，这将使滞后时间保持在+/- 6 个月内:

```
[r0,rmax,lags] = xcorr3(sst,ref,'maxlags',6,'detrend');

figure
imagescn(lon,lat,lags)
axis xy image
cb = colorbar;
ylabel(cb,'lags (months)')

caxis([-6 6])
cmocean phase
```

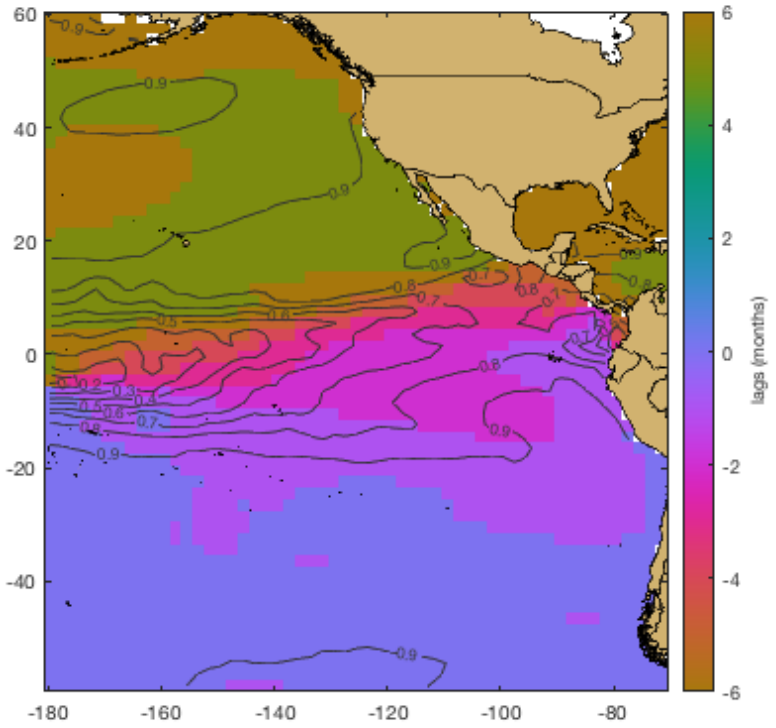


上面的地图向我们显示，阿拉斯加海岸附近的最高温度似乎发生在距智利海岸附近的最高温度大约六个月的地方。

上图使用 cmocean *相位*颜色图，因为在这种情况下，负 6 个月的延迟与正 6 个月的延迟是相同的，因此我们需要在顶部和底部使用相同的色图。

只是为了寻找刺激，让我们将最大相关性 rmax 叠加为等值线，并在上下文中覆盖一些国界：

```
hold on
[C,h] = contour(lon,lat,rmax,'color',0.2*[1 1 1]);
clabel(C,h,'color',rgb('dark gray'),'fontsize',8,'labelspacing',300);

borders('countries','facecolor',rgb('tan'))
```



作者简介

这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 于 2017 年 2 月写的。

xcov3 文档

xcov3 函数给出了三维时空数据集的单元网格与参考时间序列之间的协方差映射。

另请参见: xcorr3 和 cov.

（译者注：本部分文档应该是作者忘记修改了，几乎和 xcorr3 一模一样。我也是按照自己的理解翻译了部分文档，还请谅解）

语法

```
r = xcov3(A,ref)
r = xcov3(...,'detrend')
r = xcov3(...,'maxlag',maxlag)
r = xcov3(...,'mask',mask)
[r,rmax,lags] = xcov3(...)
```

说明

r = xcov3(A,ref) 给出二维协方差映射 r，其维数对应于 A 的维 1 和 2。三维矩阵 A 假定具有空间维 1 和 3，可以对应于 x 和 y，lat 和 lon，lon 和 lat 等。假定 A 的第三维对应于时间。数组 ref 是一个时间序列参考信号，您正在与之比较 A 的每个网格单元。ref 的长度必须与 A 的三维尺寸匹配。

r = xcov3(...,'detrend') 在计算协方差之前从每个时间序列中除去均值和线性趋势。建议将其用于 A 中的数据值或 ref 中的值范围不以零为中心的任何类型的分析。

r = xcov3(...,'maxlag',maxlag) 将最大滞后指定为时间步长的整数标量。如果指定 maxlag，则返回的互相关序列的范围是-maxlag 到 maxlag。默认 maxlag 为 N-1 个时间步长。

r = xcov3(...,'mask',mask) 仅对尺寸与 A 的前两个（空间）尺寸相对应的掩膜中的真实网格单元执行分析。如果 A 大，应用掩膜的选项旨在最大程度地减少处理时间。默认情况下，A 中的任何 NaN 值都会将默认掩膜中的相应网格单元设置为 false。

[r,rmax,lags] = xcov3(...) r 返回零相位相关系数 r，最大相关系数 rmax 和对应于最大相关的时滞。负滞后值表示本地时间序列在参考信号*之后*发生。正滞后表示局部现象领先于参考信号。

示例 1：四种人工信号的比较

考虑具有 10,000 个时间步长的简单 2x2 网格。您想查看哪些网格单元随某个参考信号 y 随时间变化:

```
% 一个时间矩阵:
t = 1:10000;

% 加点相关信号:
y = sind(t);

% 构建 A:
A = nan(2,2,10000);
A(1,1,:) = 2*y;              % 2x 相关信号
A(1,2,:) = sind(t-43);       % 有 43 个时间步长的滞后
A(2,1,:) = randn(size(t)); % 高斯噪声
A(2,2,:) = -y;               % 完全异相
```

可视化 A 的这四个网格单元的工作并不容易，因此希望这会有所帮助：这是一种查看 A 中内容的空间方法：

```
A = [ 2x 相关信号          有 43 个时间步长的滞后;
```

高斯噪声　　完全异相　]

现在我们可以看到 A 的每个网格单元格如何随 y（或不随 y）变化：

```
[r0,rmax,lags] = xcov3(A,y);
```

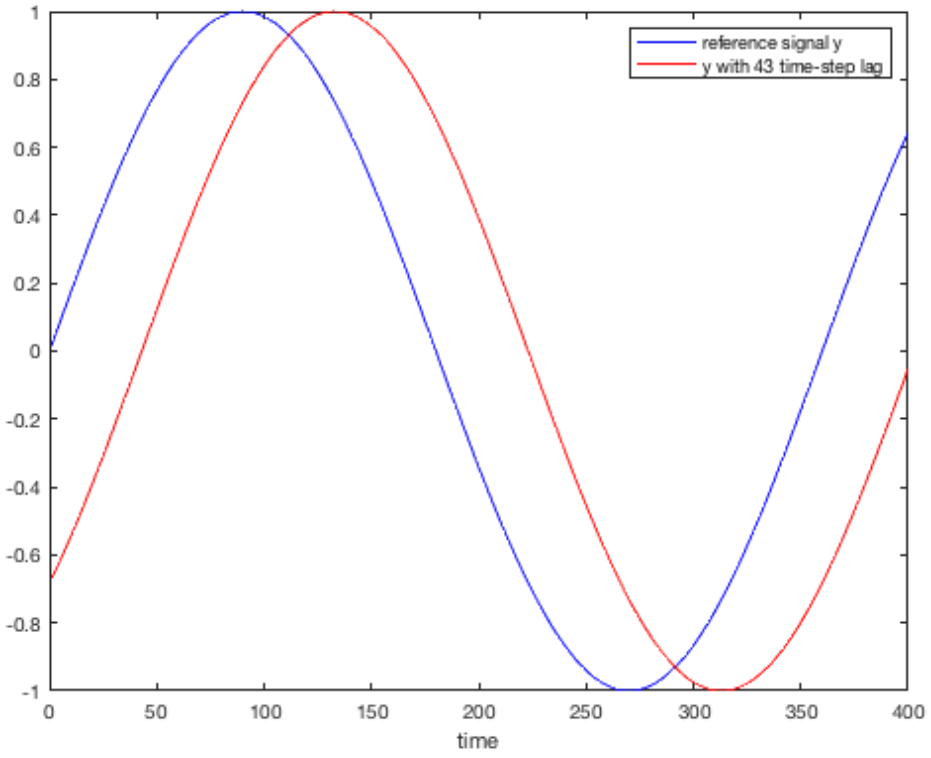
零相位协方差如下所示：

```
r0
```

```
r0 =

    1.00    0.36

   -0.01   -0.50
```

这准确地告诉了我们已经知道的。A 的左上方网格单元的值是 y 的两倍，因此它与 y 的协方差是完美的 1.0000。同样，A 的右下方网格单元具有一个我们定义为-y 的时间序列，因此它与 y 的协方差是-0.50。A 的左下角仅包含噪声，因此它与 y 根本没有很好的相关性。棘手的是 A 的右上方网格单元，它具有与 y 相同的信号，但在时间上偏移了 43 个时间步长。以下是 y 和 sind（t-43）的摘要：

```
plot(t(1:400),y(1:400),'b')
hold on
plot(t(1:400),sind(t(1:400)-43),'r')
xlabel('time')
legend('reference signal y','y with 43 time-step lag')
```



在上面的图中，很明显 y 和 y 的时滞形式经常一起上升或下降，但是有时时间滞后意味着它们没有很好地相关，因此，如果我们不相关，考虑时间偏移，则协方差为 0.36。但是，您还可以看到，如果将信号之一偏移 43 个时间步长，则两个信号将完全相关。实际上，如果我们看一下最大协方差，那就很清楚了：

```
rmax
```

```
rmax =

    1.04    0.67

    0.22    0.56
```

我们看到，如果在时间上偏移 A 值，则 A 的右上方网格单元将与相关系数为 1（或接近 1）的参考时间序列匹配，这意味着完美匹配。信号最适合以下延迟:

```
lags
```

```
lags =

   -9721.00    9999.00

    9976.00    9904.00
```

再一次，这准确地告诉了我们已经知道的。A 的左上方网格像元与参考时间序列完全匹配，且延迟为零；我们有意对 A 的右上方网格单元应用了 43 个时间步长滞后；左下方网格单元永远不会与参考信号很好地关联，因此其滞后值毫无意义，A 的右下方网格单元会以 180 个时间步长的偏移量匹配参考信号（因为在此示例中，偏移量方便地对应于度）

示例 2：海表温度

上面的示例旨在提供 xcov3 的工作原理背后的理论知识，但并未捕获您将 xcov3 应用于实际数据时可获得的时空模式的见解。因此，让我们看一下海表温度的再分析样本数据集：

```
load pacific_sst
whos lat lon t
```

```
  Name        Size              Bytes  Class     Attributes

  lat        60x1                480  double
  lon        55x1                440  double
  t         802x1               6416  double
```

示例数据集包含一个具有以下分辨率的 sst 矩阵：

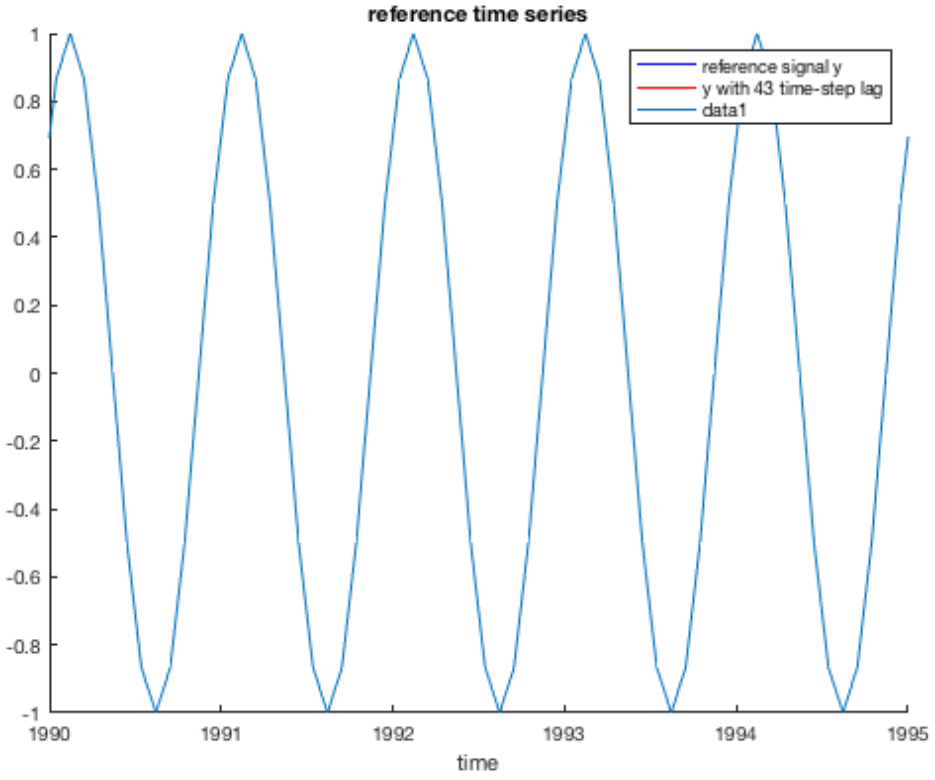
```
mean(diff(lat))

mean(diff(lon))

mean(diff(t))
```

```
ans =

        -2.00

ans =

         2.00

ans =

        30.44
```

也就是说，sst 是具有每月时间分辨率的 2x2 度网格。

我有一个假设，即海面温度随季节呈正弦变化。要将海面温度与正弦曲线进行比较，首先创建与一年中的月份相对应的参考正弦曲线：

```
[~,month,~] = datevec(t);

ref = sin((month+1)*pi/6);



plot(t,ref)

xlim([datenum('jan 1, 1990') datenum('jan 1, 1995')])
box off
datetick('x','keeplimits')
title ' reference time series '
```

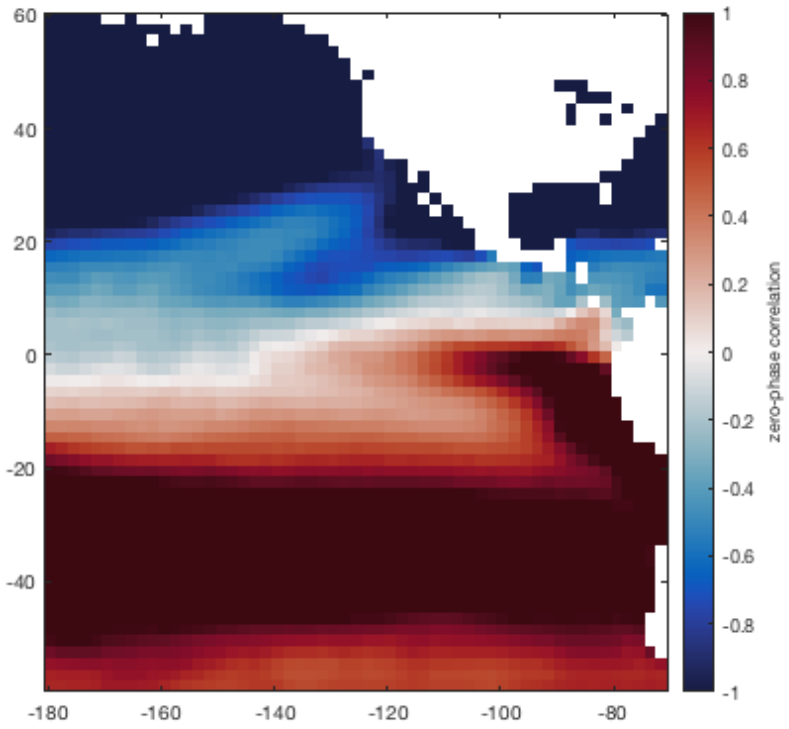


参考信号在每年的 2 月具有最大值。我的假设是，南半球的海面温度应与参考信号成正相关，这意味着南半球的海表温度应在 2 月达到最高，而在 8 月达到最低。北半球应显示完全相反的模式，每年八月最高，每年二月最低。

让我们看一下原始海面温度数据与参考正弦曲线之间的相关性。（下面我使用 imagesscn 来创建 NaC 值透明的 imagec 图，但是如果愿意，可以使用 imagesc 或 pcolor。我还使用了 cmocean 色图(Thyng et al., 2016)，这是发散的和感性上的统一。）

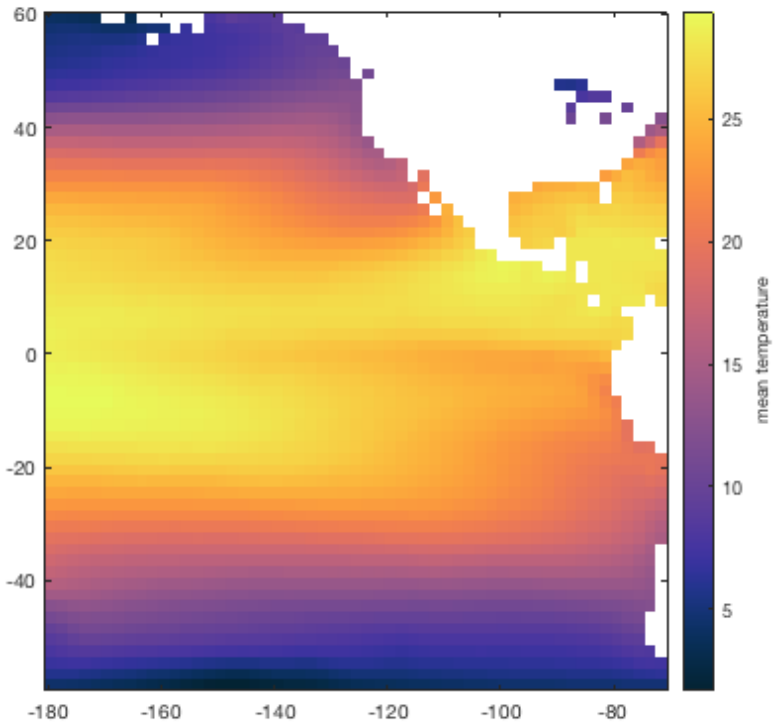
```
r0 = xcov3(sst,ref);
```

```
figure

imagescn(lon,lat,r0)

axis xy image
cb = colorbar;
ylabel(cb,'zero-phase correlation')
caxis([-1 1])
cmocean balance
```



在上面的地图中，我们看到海面温度似乎与参考正弦曲线仅*略微*相关。但是我们已经疏忽了！我们忘记从 sst 数据集中删除平均值，因此与正弦曲线的相关性当然很弱。这是平均海面温度的地图，它污染了上面的分析：

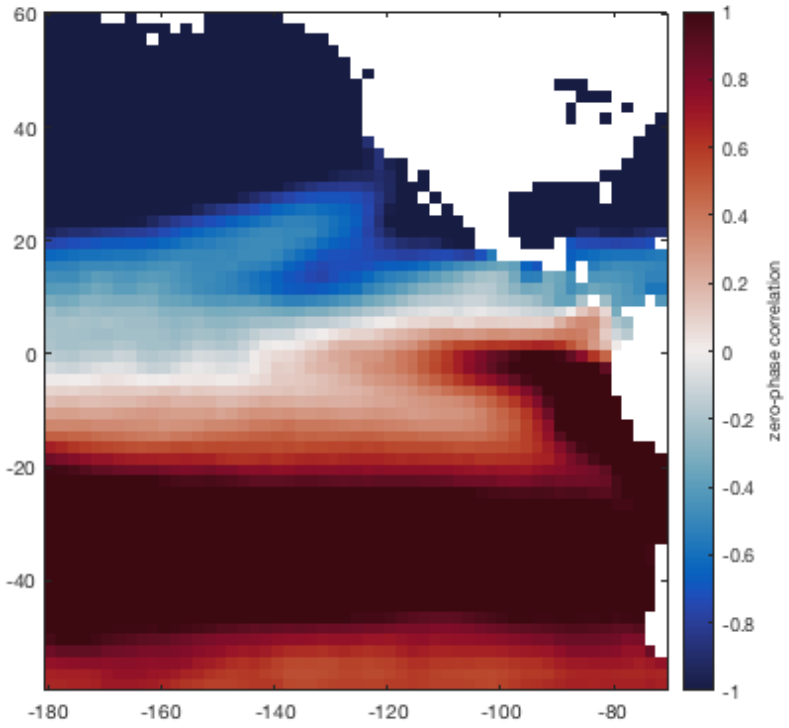
```
figure

imagescn(lon,lat,mean(sst,3))

axis xy image
cb = colorbar;
ylabel(cb,'mean temperature')
cmocean thermal
```



在使用 xcov3 之前，您可以根据需要手动删除均值，也可以简单地使用'detrend'选项，这将为您删除均值。'detrend'选项还删除了长期（全球变暖）趋势，这很方便，因为当去除均值和长期趋势时，剩下的只是季节周期和年际变化。

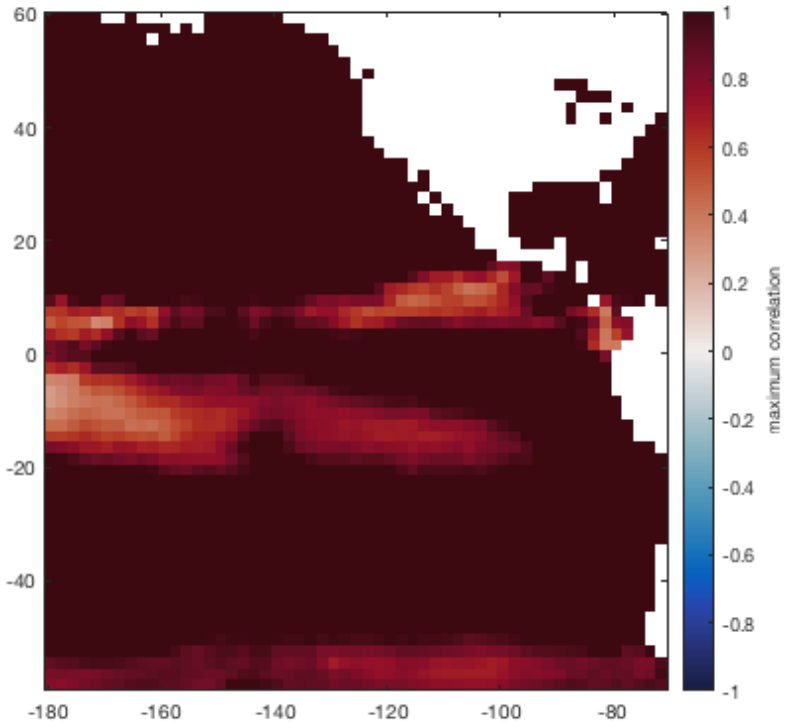
因此，让我们看一下海面温度*异常*（而不是绝对值）与参考正弦波的比较：

```
[r0,rmax,lags] = xcov3(sst,ref,'detrend');

figure
imagescn(lon,lat,r0)
axis xy image
cb = colorbar;
ylabel(cb,'zero-phase correlation')
caxis([-1 1])
cmocean balance
```



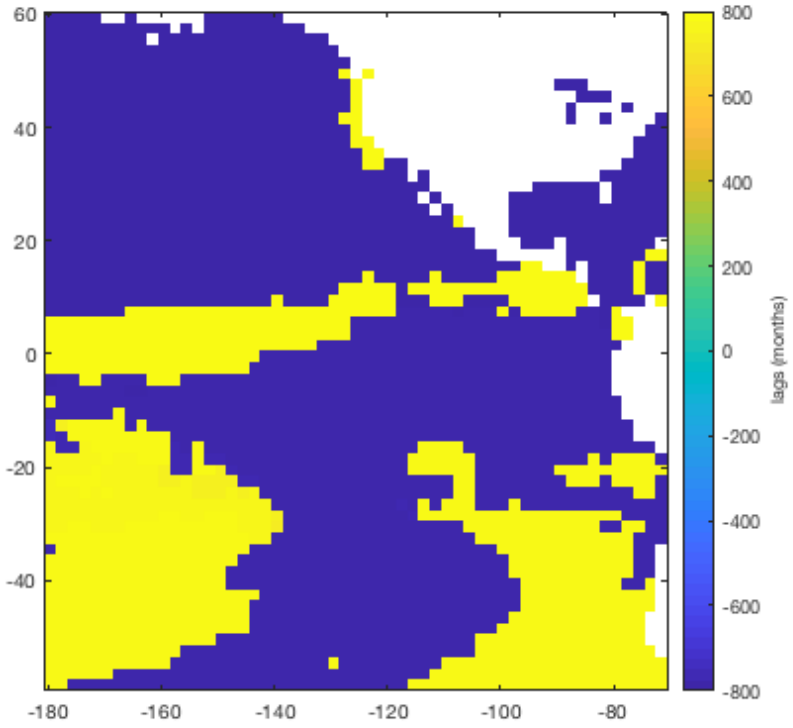
哇！现在，有一些明确的证据表明海表温度似乎随季节变化。正如预期的那样，在南半球与参考正弦波呈正相关，这意味着南半球的海面温度在 2 月前后达到最高，在 8 月左右达到最低。北半球与南半球几乎完全反相关。在赤道附近，与参考正弦曲线的相关性很小。但是也许我们为参考信号选择的相位与数据不完全匹配。如果我们随时间移动参考信号，那么每个网格单元与参考正弦波可达到的最佳匹配是什么？

```
figure

imagescn(lon,lat,rmax)

axis xy image
cb = colorbar;
ylabel(cb,'maximum correlation')
caxis([-1 1])
cmocean balance
```



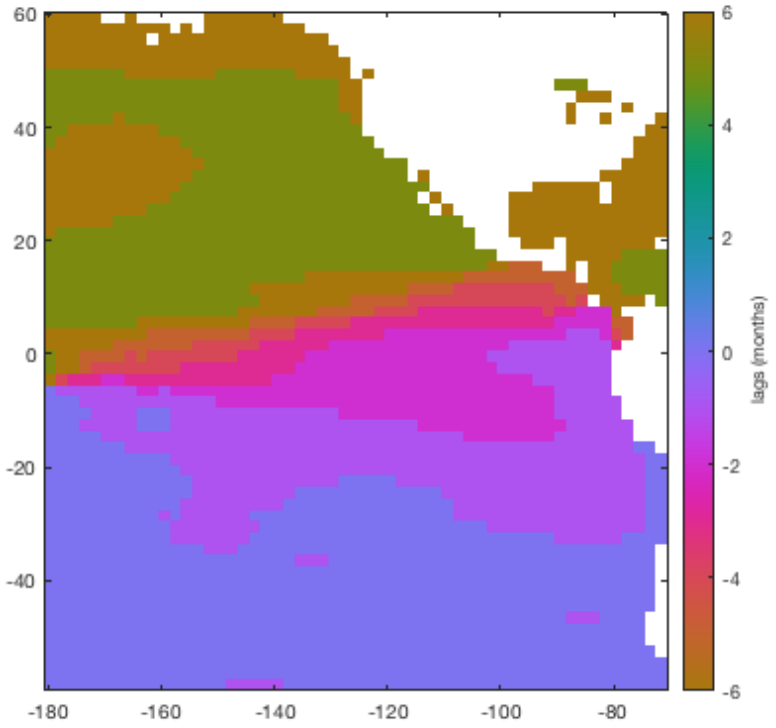
上面的地图向我们展示了几乎整个太平洋都有一些正弦变化。通过查看滞后，我们可以找到最匹配的正弦波的相位：

```
figure

imagescn(lon,lat,lags)

axis xy image
cb = colorbar;
ylabel(cb,'lags (months)')
```



上面的地图似乎是垃圾。（180W，4S）附近的网格单元需要大约 391 个月（32 年）的偏移量才能获得与正弦曲线的最佳匹配？再看一下最大相关图，您会发现（180W，4S）从未与正弦曲线很好地相关-它的海表温度似乎没有季节性周期。鉴于 6 个月的负偏移量与参考信号的 6 个月的正偏移量在数学上是相同的，因此我们实际上应该将最大滞后时间限制为 6 个时间步长，这将使滞后时间保持在+/- 6 个月内:

```
[r0,rmax,lags] = xcov3(sst,ref,'maxlags',6,'detrend');

figure
imagescn(lon,lat,lags)
axis xy image
cb = colorbar;
ylabel(cb,'lags (months)')

caxis([-6 6])
cmocean phase
```

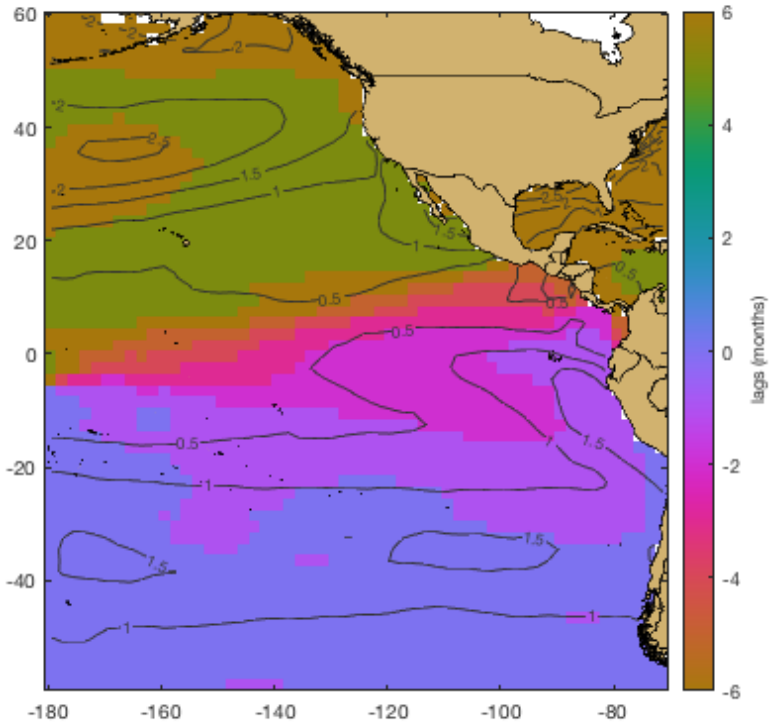


上面的地图向我们显示，阿拉斯加海岸附近的最高温度似乎发生在距智利海岸附近的最高温度大约六个月的地方。

上图使用 cmocean *相位*颜色图，因为在这种情况下，负 6 个月的延迟与正 6 个月的延迟是相同的，因此我们需要在顶部和底部使用相同的色图。

只是为了寻找刺激，让我们将最大相关性 rmax 叠加为等值线，并在上下文中覆盖一些国界：

```
hold on
[C,h] = contour(lon,lat,rmax,'color',0.2*[1 1 1]);
clabel(C,h,'color',rgb('dark gray'),'fontsize',8,'labelspacing',300);

borders('countries','facecolor',rgb('tan'))
```



作者简介

这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 于 2017 年 2 月写的。

时间序列

- filt1 将零相位巴特沃斯滤波器应用于时间序列.
- scatstat1 返回每个值在给定一维半径内所有点的统计值。这类似于采用移动平均，但不必将点等距分布，也不必使 x 值单调递增.
- doy 返回一年中的某天.
- reshapetimeseries 将时间序列数据整形为年*时间的网格。

filtl 文档

filtl 函数将零相位巴特沃斯滤波器应用于时间序列. **需要 Matlab 的信号处理工具箱（Signal Processing Toolbo x）.**

语法

```
yf = filtl(filtertype,y,'fc',Fc)
yf = filtl(filtertype,y,'Tc',Tc)
yf = filtl(filtertype,y,'lambdac',lambdac)
yf = filtl(...,'fs',Fs)
yf = filtl(...,'x',x)
yf = filtl(...,'Ts',Ts)
yf = filtl(...,'order',FilterOrder)
yf = filtl(...,'dim',dim)
[yf,filtb,filta] = filtl(...)
```

说明

yf = filtl(filtertype,y,'fc',Fc) 使用指定的 filtertype 和截止频率 fc 对一维信号 y 进行滤波。对于高通或低通滤波器，Fc 必须是标量。对于带通和带阻滤波器，Fc 必须是一个二维数组。filtertype 可以是

- 'hp' 带有截止频率标量 Fc 的高通
- 'lp' 带有截止频率标量 Fc 的低通
- 'bp' 带有两个元素截止频率 Fc 的带通
- 'bs' 带有两个元素截止频率 Fc 的带阻

yf = filtl(filtertype,y,'Tc',Tc) 指定截止周期而不是截止频率。这个语法假设 T=1/f，与'lambdac'选项完全相同，但对于处理时间序列来说可能更直观一些。

yf = filtl(filtertype,y,'lambdac',lambdac) 指定截止波长而不是截止频率。这个语法假设 λ=1/f。

yf = filtl(...,'fs',Fs) 指定采样频率 fs。如果既没有指定'fs'、'x'也没有指定'Ts'，则假定 fs=1。

yf = filtl(...,'x',x) 指定单调递增的等距采样时间向量或对应于 y 的 x 位置,用于确定采样频率。如果既没有指定'fs'、'x'也没有指定'Ts'，则假定 fs=1。

yf = filtl(...,'Ts',Ts) 指定采样周期或采样距离，使 Fs=1/Ts。如果既没有指定'fs'、'x'也没有指定'Ts'，则假定 fs=1。

yf = filtl(...,'dim',dim) 指定要沿其操作的维度。默认情况下，对于一维或二维数组，filt1 沿第一个非单精度维度操作，而对于三维网格数据集，filt1 沿第三维度操作。

yf = filtl(...,'order',FilterOrder) 指定 Butterworth 过滤器的顺序（有时称为 *rolloff*）。如果未指定，则假定 FilterOrder=1。

[yf,filtb,filta] = filtl(...) 还返回过滤器分子 filta 和分母 filtb。

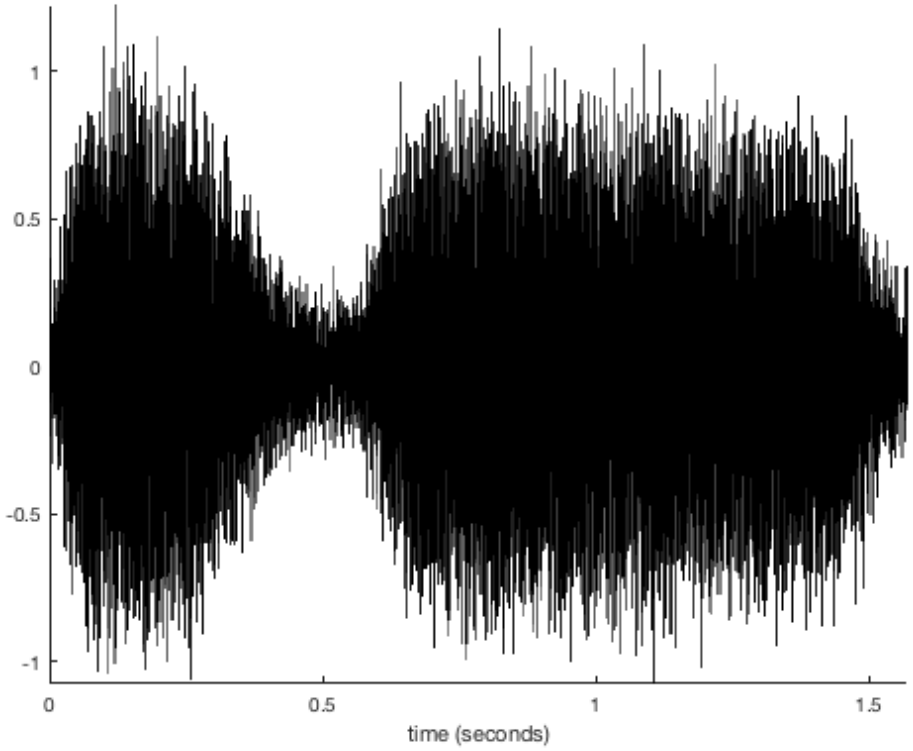
示例 1: 火车鸣笛

对于这个例子，我们将使用内置的火车鸣笛示例文件，并添加一点高斯随机噪声，使事情变得有趣。

```
load train
y = y+0.1*randn(size(y));
```

原始信号可以这样用黑色绘制。首先，我们必须根据采样频率 Fs 构建时间向量：

```
t = (0:length(y)-1)/Fs;



plot(t,y,'k-','linewidth',1)
box off;
axis tight
xlabel 'time (seconds)'
hold on
```

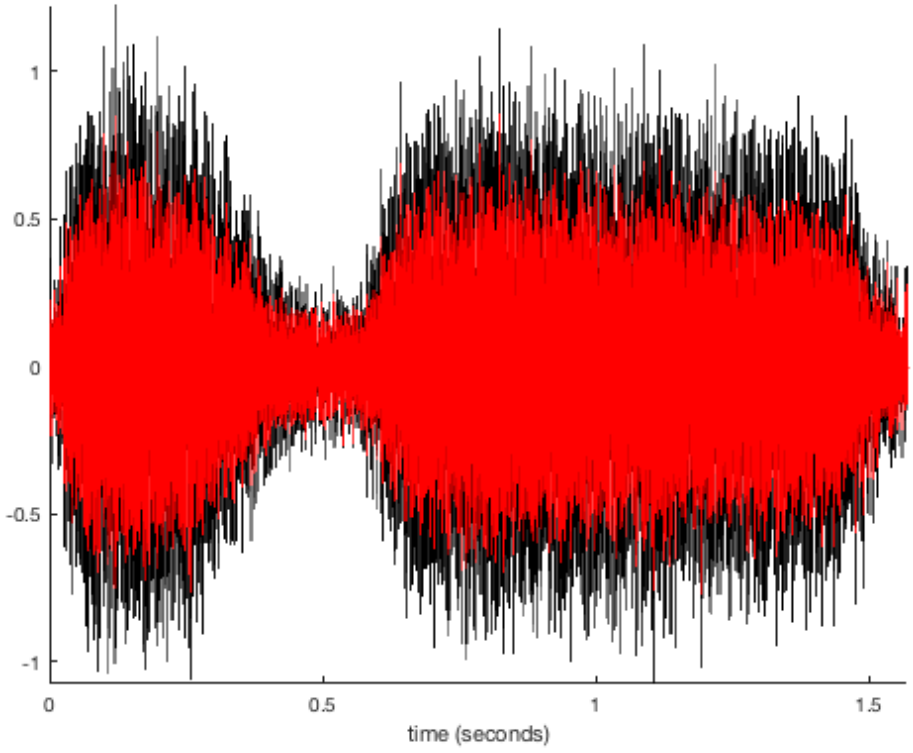


如果你有扬声器，你可以像这样听火车鸣笛：

```
soundsc(y,Fs)
```

高通滤波器列车鸣笛，保持频率仅高于 750Hz，并将其绘制在原始黑色信号的顶部：

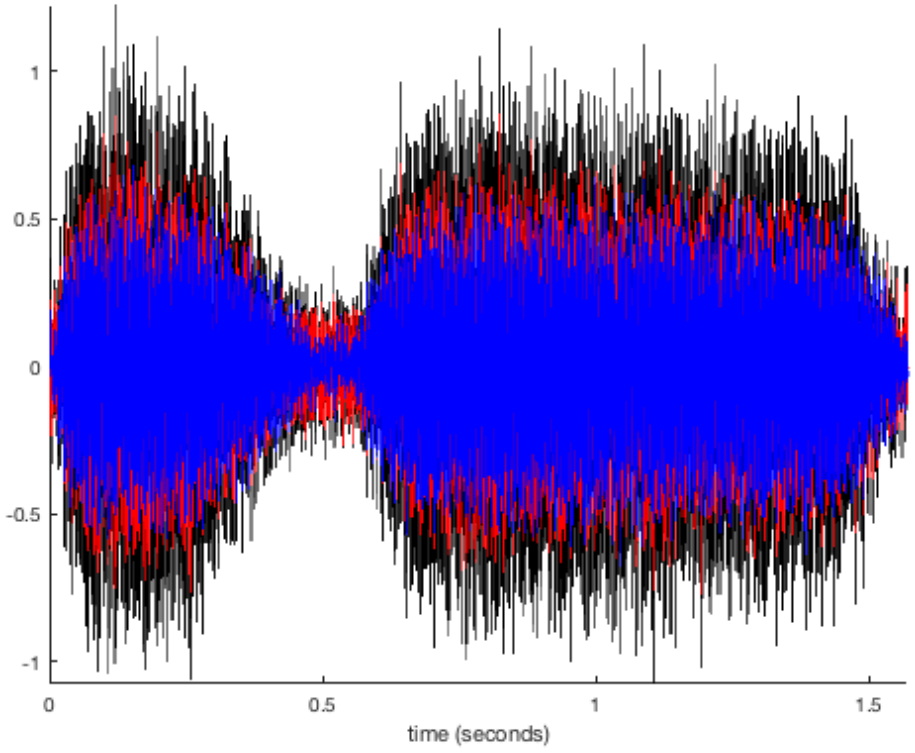
```
yhp = filt1('hp',y,'fs',Fs,'fc',750);

hold on
plot(t,yhp,'r')
```



或者你想对原始信号进行低通滤波以消除 1100Hz 以下的频率。注：上面我们通过设置'fs'值来指定采样频率。也可以将向量定义为自变量'x'。在这种情况下，自变量是时间，但对于空间滤波，它可能是沿某条路径的累积距离。

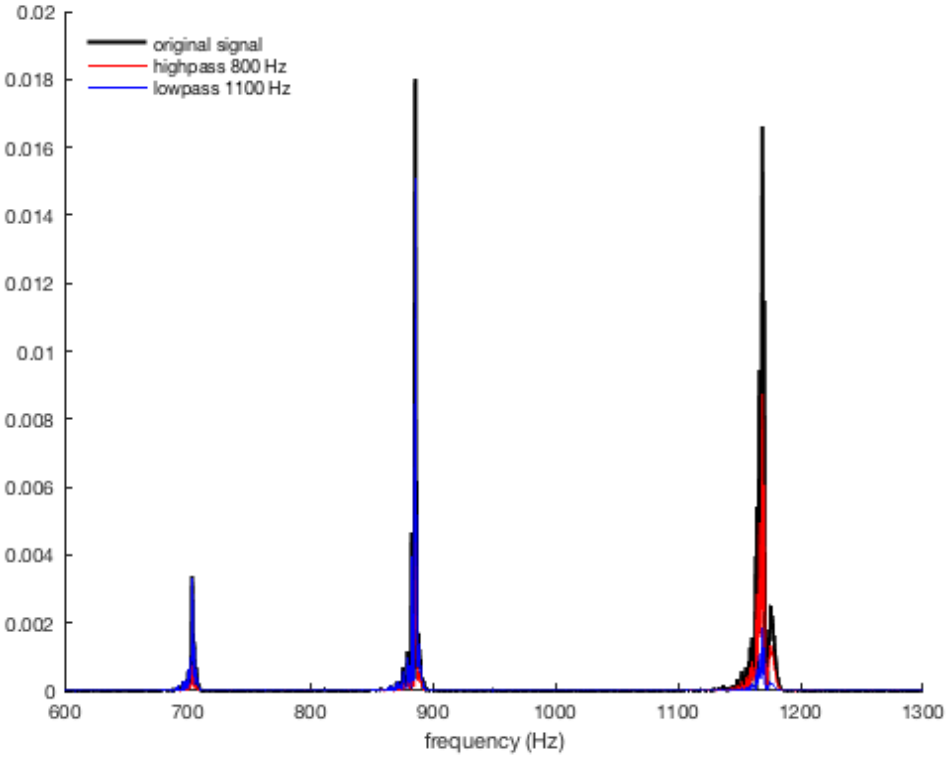
火车鸣笛的三个主频在频率空间中的间隔有些接近，因此我们上面使用的默认一阶巴特沃斯滤波器不会消除 750Hz 以下的所有能量。您可能希望通过指定'order'，5 来使用更陡的翻滚。我们将用蓝色绘制低通滤波火车鸣笛。

```
ylp = filt1('lp',y,'x',t,'fc',1100,'order',5);
plot(t,ylp,'b')
```



使用 plotpsd 比较：

```
figure

plotpsd(y,Fs,'k','linewidth',2)
hold on
plotpsd(yhp,Fs,'r')
plotpsd(ylp,Fs,'b')

xlabel 'frequency (Hz)'
axis([600 1300 0 0.02])
legend('original signal','highpass 800 Hz',...
    'lowpass 1100 Hz','location','northwest')
legend boxoff
```



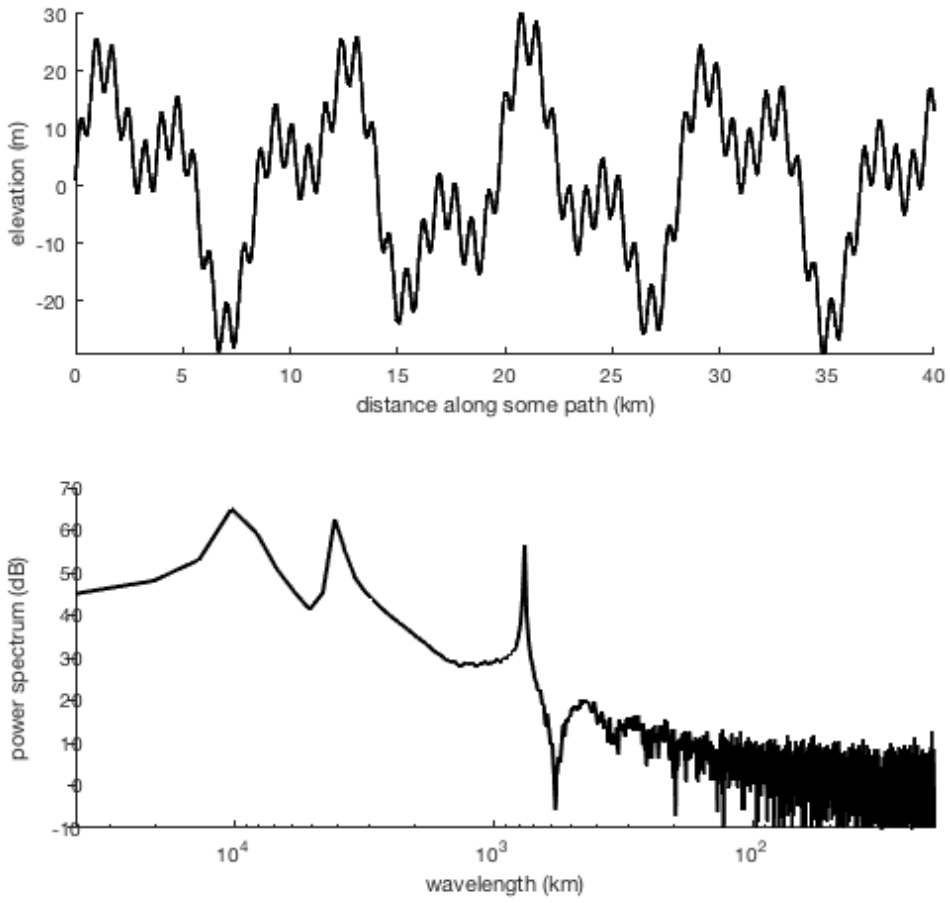

```
clear variables
close all
```

示例 2: 地形剖面

假设你有一个地形剖面，在 40 公里内每 10 米测量一次高程。假设这个剖面有三个主波长——761 米，4 公里，和 9.4 米。配置文件可能如下所示。在示例 1 中，我使用 plotpsd 来绘制周期图。

```
SpatialRes = 10;        % 每 10m 一个样本 Samples every 10 m
x = 0:SpatialRes:40e3; % 0 到 40km 的域
lambda1 = 761;          % 761 m
lambda2 = 4000;         % 4 km
lambda3 = 3000*pi;      % ~9.4 km

% 生成剖面:
y = rand(size(x)) + 5*sin(2*pi*x/lambda1) + ...
    11*sin(2*pi*x/lambda2) + 15*sin(2*pi*x/lambda3) ;

% 绘制剖面:
figure('position',[100 100 560 506])
subplot(211)
plot(x/1000,y,'k','linewidth',2)
hold on
xlabel 'distance along some path (km)'
```

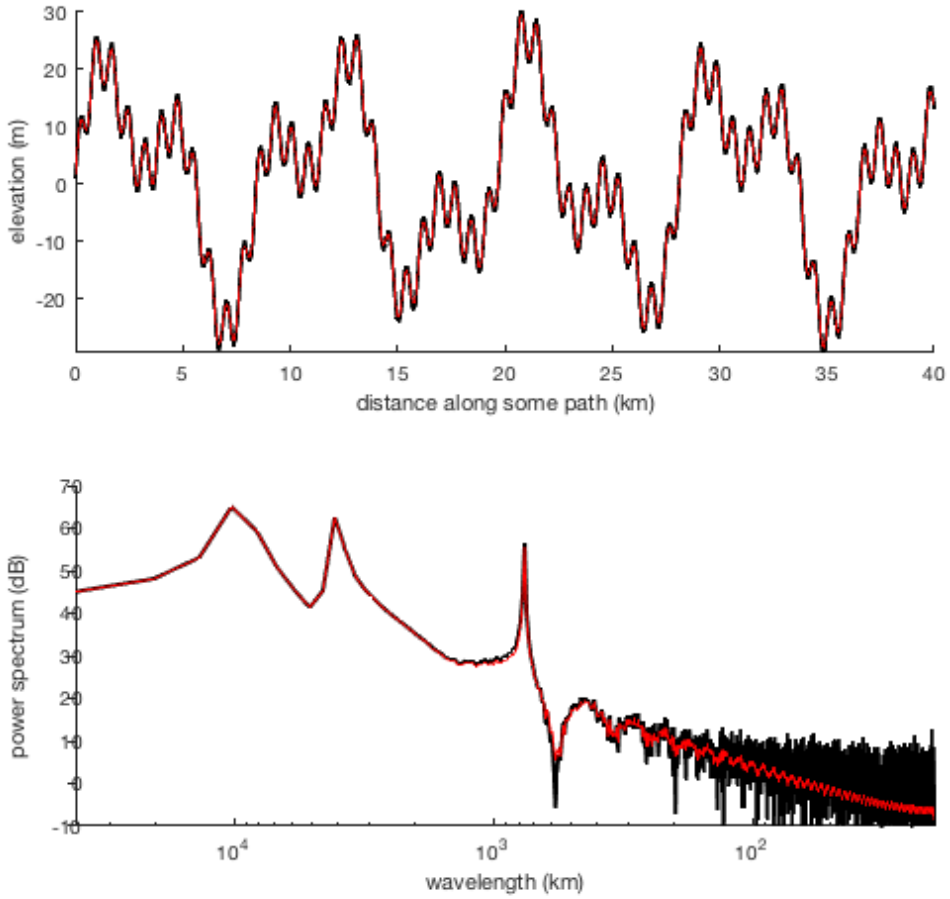
```
ylabel 'elevation (m)'
box off
axis tight

% 绘制功率谱:
subplot(212)
plotpsd(y,x,'k','linewidth',2,'db','log','lambda')
hold on
xlabel 'wavelength (km)'
ylabel 'power spectrum (dB)'
axis tight
ylim([-10 70])
```



在上面，你可以看到周期图中的三个主波长作为三个峰值。
也许你想消除高频随机噪声，我们加入了随机地形。为此，可以低通滤除所有小于 300 m 的波长：

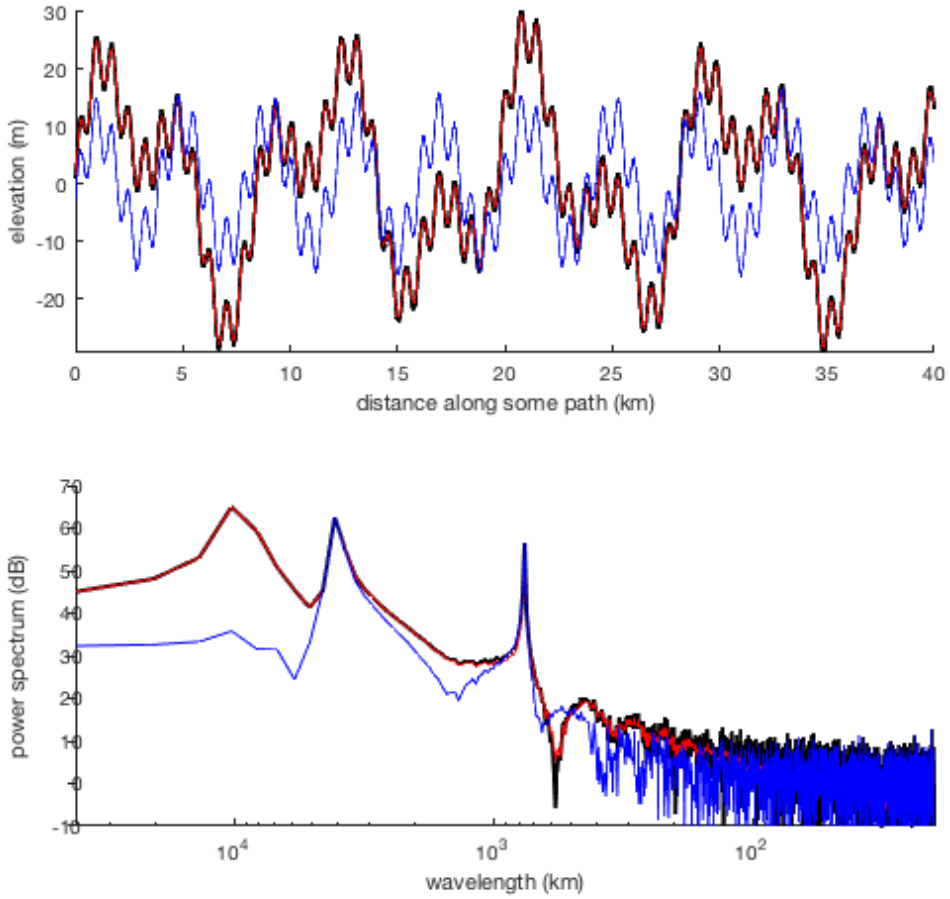
```
ylo = filt1('lp',y,'x',x,'lambdac',300);

subplot(211)
plot(x/1000,ylo,'r')
```

```
subplot(212)
plotpsd(ylo,x,'r','db','log','lambda')
```



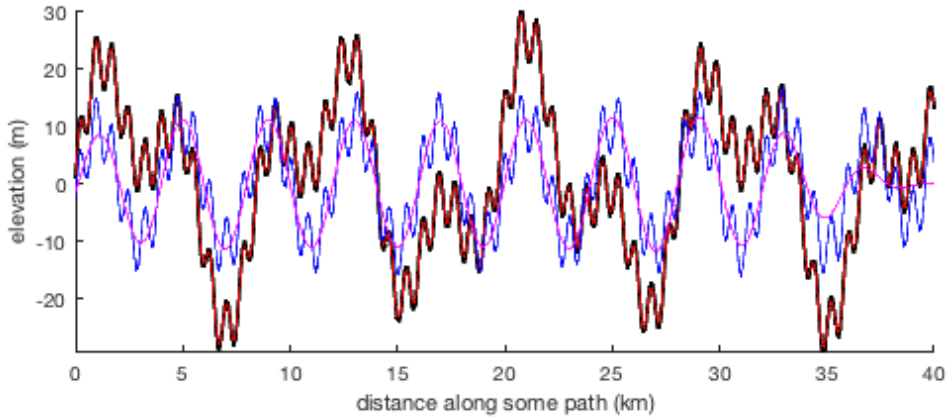
上面，当我们对地形进行低通滤波时，我们指定了一个数组 x 作为对应于 y 的路径距离。或者，我们可以指定空间分辨率，即采样距离'Ts'，以获得相同的结果。下面我们对原始地形进行高通滤波，去除波长超过 6 公里的波长。通过指定 5 阶巴特沃斯过滤器使用紧密滚动。

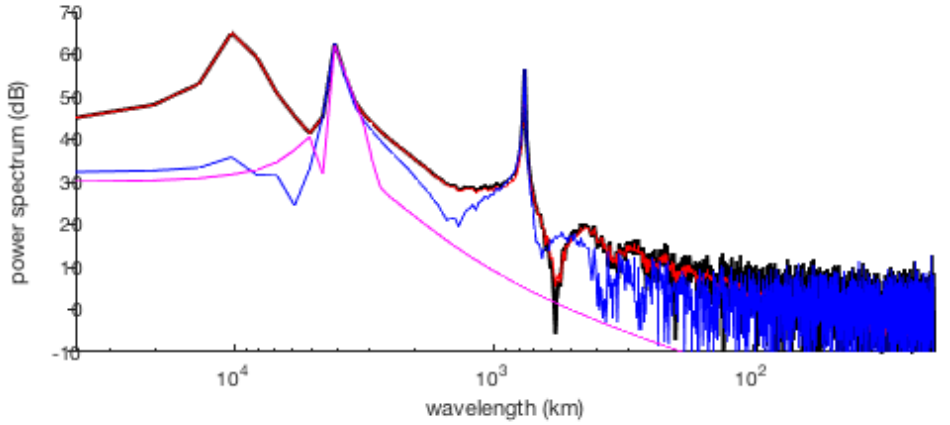
```
yhi = filt1('hp',y,'Ts',SpatialRes,'lambdac',6000,'order',5);

subplot(211)
plot(x/1000,yhi,'b')

subplot(212)
plotpsd(yhi,x,'b','db','log','lambda')
```



也许你想消除高频噪音和低频。您可以通过两次滤波信号或带通滤波器来实现这一点。我们只能通过带通滤波来保留功率谱中的中峰，所有波长小于 3000 米或超过 3000 米

```
ybp = filt1('bp',y,'x',x,'lambdac',[3000 5000],'order',3);

subplot(211)
plot(x/1000,ybp,'m')

subplot(212)
plotpsd(ybp,x,'m','db','log','lambda')
```



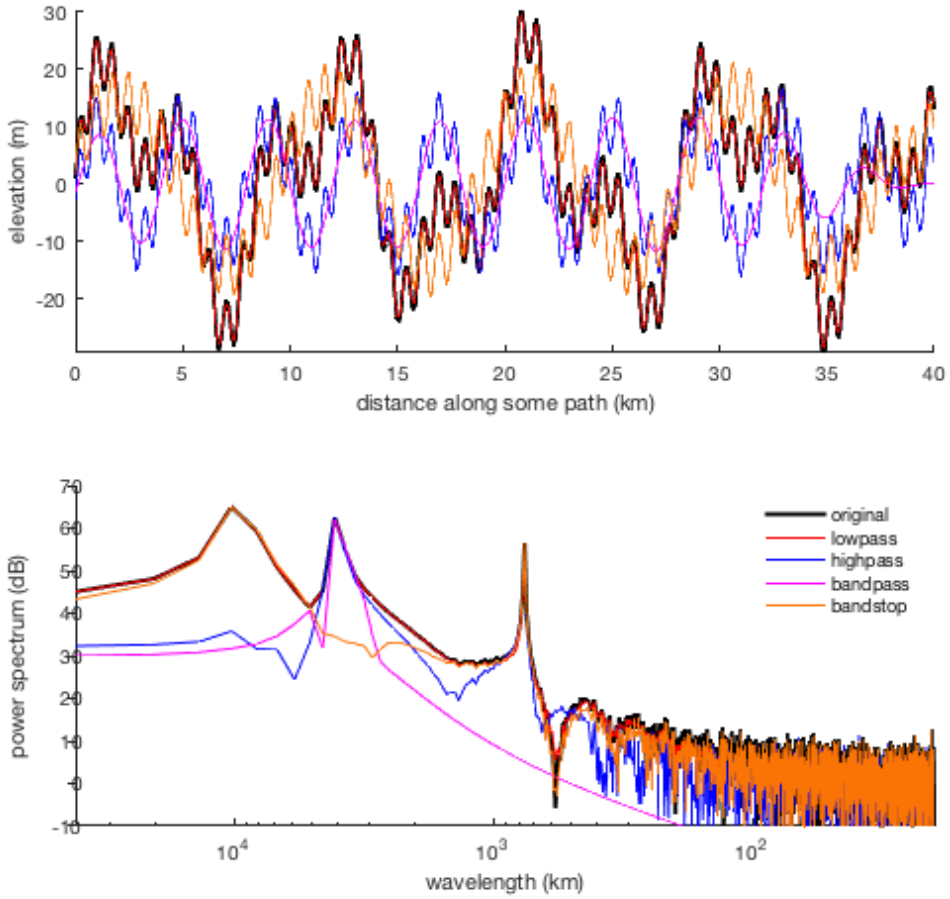


也许您只想删除一系列频率。你可以从原始信号中减去一个带通信号 la：

```
ybs = y - ybp;
```

或者您可以使用相同的方法直接创建带阻滤波器，语法与带通滤波器一样：

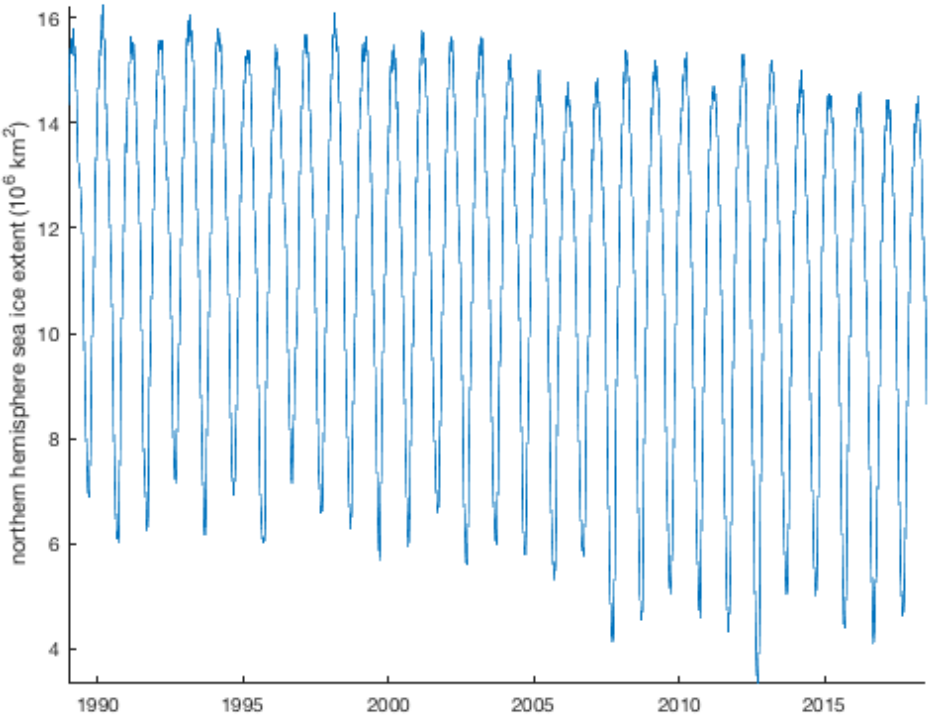
```
ybs = filt1('bs',y,'x',x,'lambdac',[3000 5000],'order',3);

subplot(211)
plot(x/1000,ybs,'color',[.98 .45 .02])

subplot(212)
plotpsd(ybs,x,'color',[.98 .45 .02],'db','log','lambda')

legend('original','lowpass','highpass',...
    'bandpass','bandstop','location','northeast')
legend boxoff
```



示例 3: 海冰范围

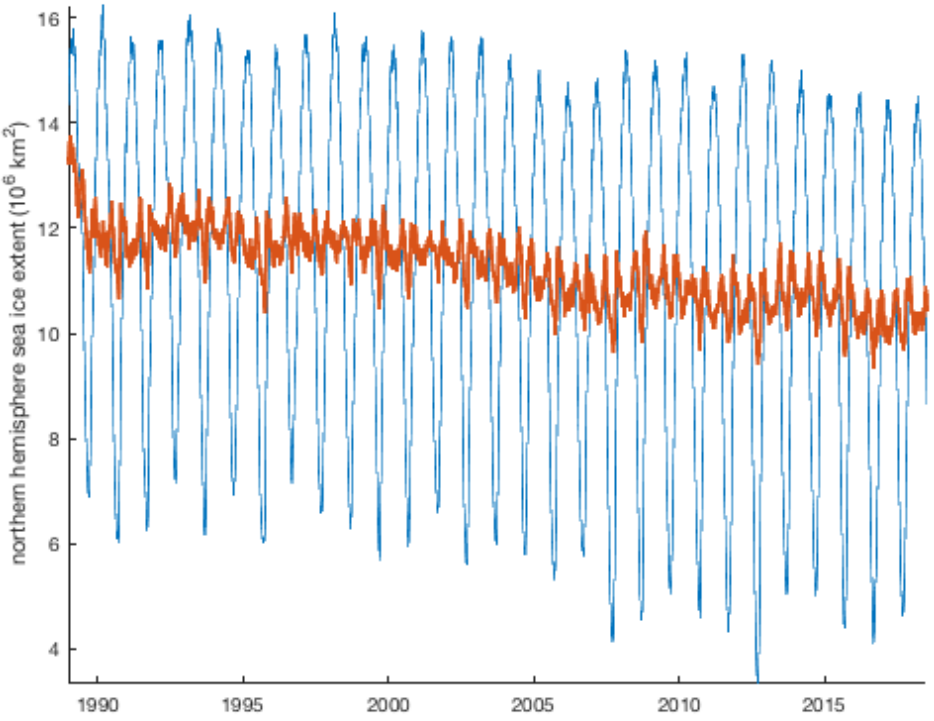
加载海冰范围的示例时间序列，仅使用 1989 年至今的数据，因为旧数据的采样分辨率低于每日分辨率。

```
load seaice_extent.mat

% 获取 1989 年后的指数:
ind = t>datetime(1989,1,1);

% 将数据集调整到 1989-2018 年：
t = t(ind);
extent_N = extent_N(ind);

figure
plot(t,extent_N)
axis tight
box off
ylabel 'northern hemisphere sea ice extent (10^6 km^2)'
```



我们能试着过滤掉季节循环吗？当然 deseason 函数是一种方法，但是如果我们用这个 butterworth 过滤器呢？季节周期上的带阻滤波器要求我们定义转角频率，那么 6 个月周期和 2 年周期之间的一切又如何呢？

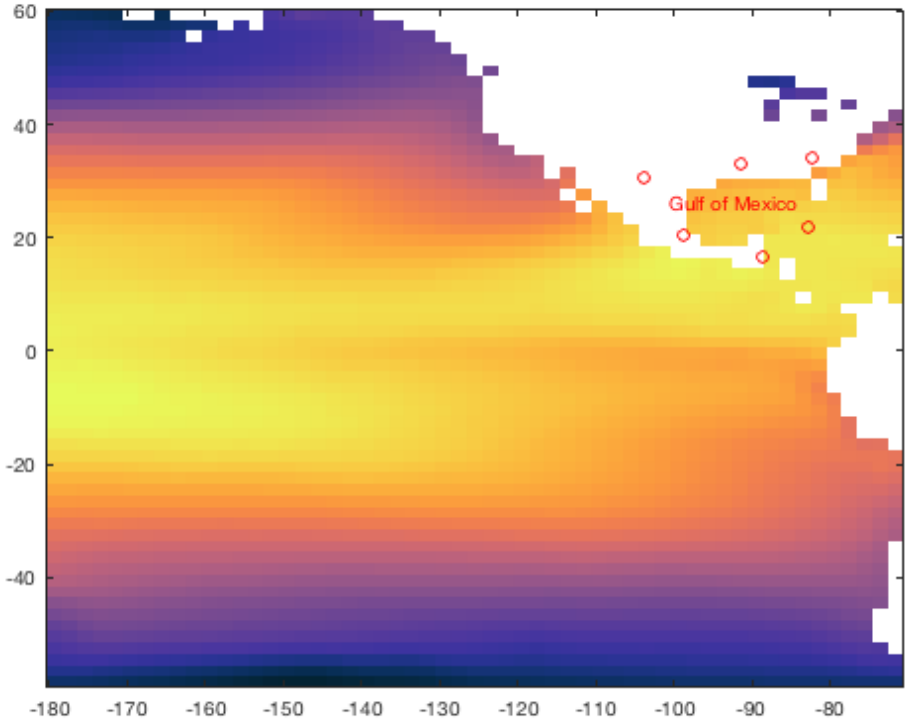
```
extent_N_filt = filt1('bs',extent_N,'fs',365.25,'Tc',[0.5 2]);

hold on
plot(t,extent_N_filt,'linewidth',2)
```



示例 4: 网格化三维时间序列

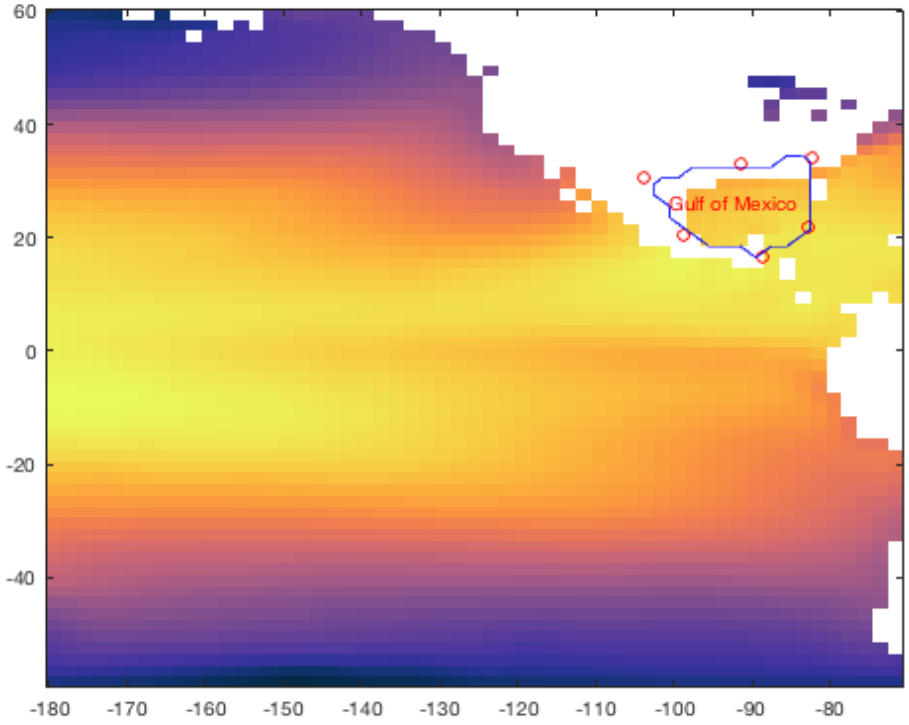
假设您对墨西哥湾（GoM）的 SST 感兴趣

```
load pacific_sst

figure
imagescn(lon,lat,mean(sst,3))
cmocean thermal % 颜色图
hold on

% 墨西哥湾的粗略轮廓：
gomlon = [-91.4 -103.8  -98.8  -88.6  -82.8  -82.3];
gomlat = [33.0   30.6   20.4   16.6   21.8   34.1];
plot(gomlon,gomlat,'ro')

text(-92.4,23.9,'Gulf of Mexico','color','red',...
   'horiz','center','vert','bot') % 设置文本对齐方式
```



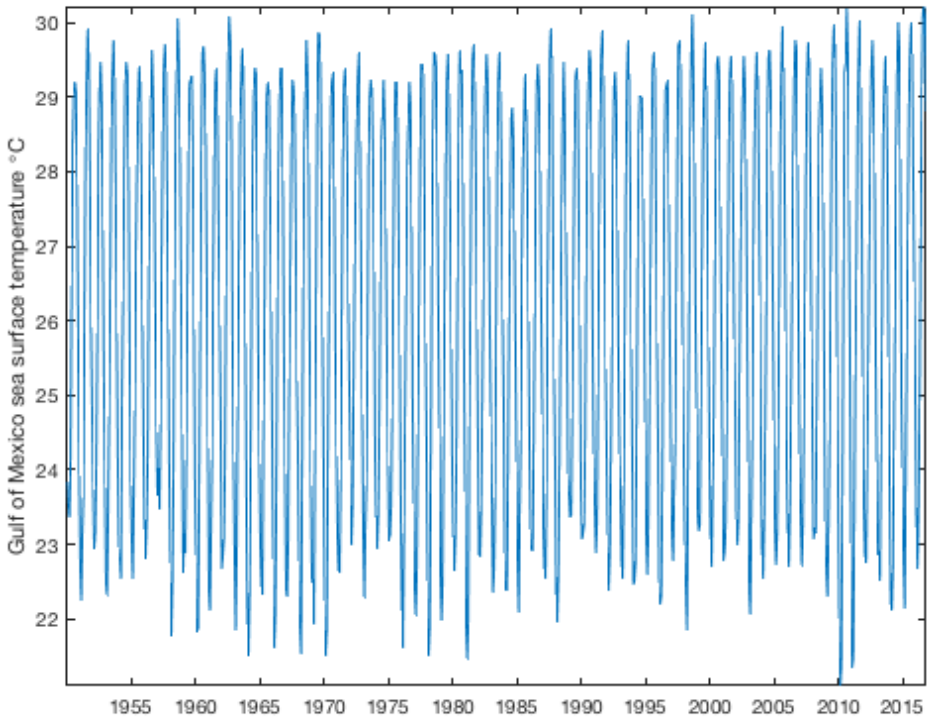
我们可以使用 geomask 和 local 得到墨西哥湾 SST 的时间序列。首先，制作掩膜并将其绘制为蓝色等值线，以确保掩膜位于正确的位置：

```
[Lon,Lat] = meshgrid(lon,lat);

mask = geomask(Lat,Lon,gomlat,gomlon);



contour(Lon,Lat,double(mask),[0.5 0.5],'b')
```



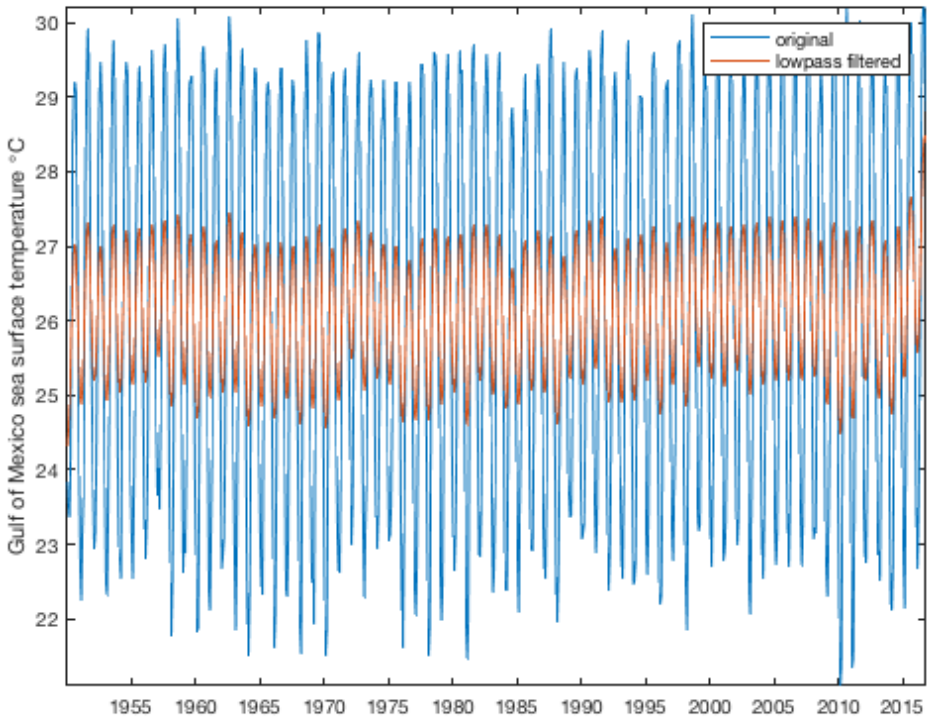
通过定义 mask，很容易获得墨西哥湾平均 SST 的一维时间序列：

```
sst_gom = local(sst,mask,'omitnan');

figure
plot(t,sst_gom)
axis tight
datetick('x','keeplimits')
ylabel 'Gulf of Mexico sea surface temperature \circC'
```



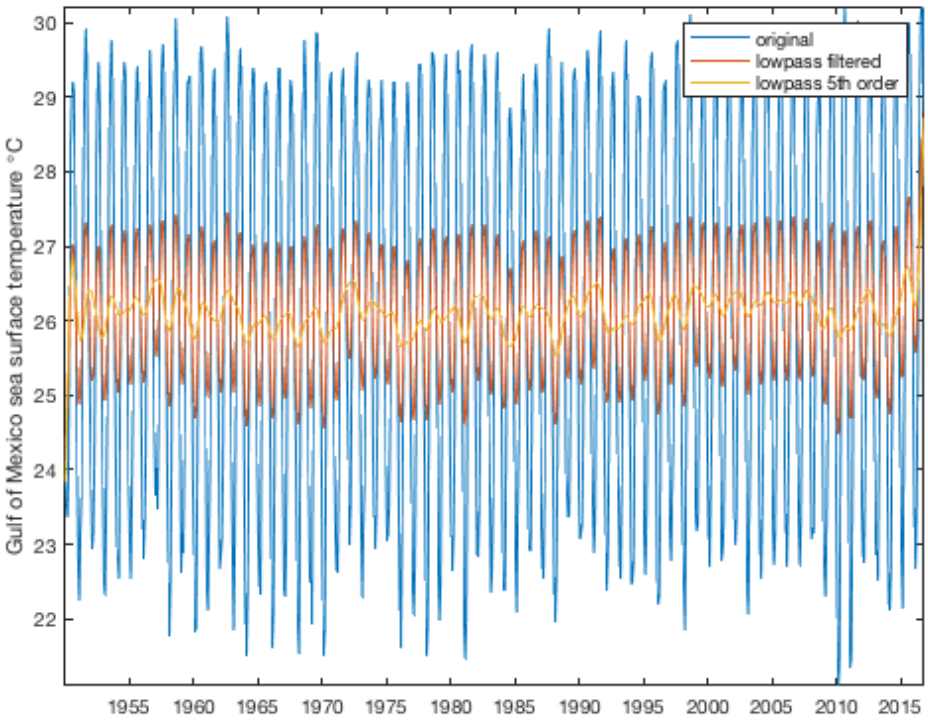
这个数据集具有每月的时间分辨率，因此如果我们想对其进行低通滤波，只保留周期超过 18 个月的频率，我们可以对一维 sst1 数据集进行如下操作：

```
% 一维 sst1 低通滤波:
sst_gom_lp = filt1('lp',sst_gom,'Tc',18);

hold on
plot(t,sst_gom_lp)

legend('original','lowpass filtered')
```



你会注意到在 1 年的频率下仍然有大量的能量。这是因为巴特沃斯滤波器 Butterworth filter 并不理想——在频率空间，它的肩很宽。若要在短于 18 个月的时间内使其更锋利并更有效地减少能量，请增加过滤器的阶数：

```
% 一维 sst1 低通滤波:
sst_gom_lp5 = filt1('lp',sst_gom,'Tc',18,'order',5);

plot(t,sst_gom_lp5)
legend('original',...
 'lowpass filtered',...
 'lowpass 5th order')
```



现在你可以看到一些年际变化，而不是被季节周期所控制。过滤掉任何应用中的大部分可变性可能不是一个好主意，但这里我们只是在玩弄函数的机制，所以我们只能说它很好，然后继续。

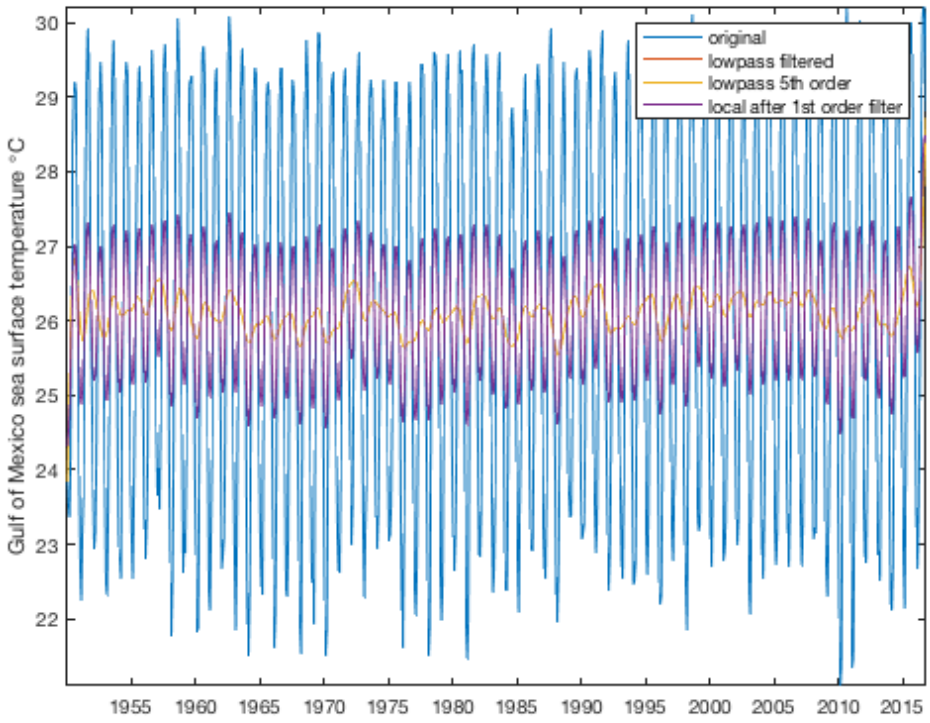
但是，如果不只是过滤一维数组，而是要过滤每个网格单元的时间序列呢？计算效率低下的方法是嵌套几个循环，遍历网格的每一行和每一列，所有的 65x55 网格单元。但这意味着要做同样的操作超过 3000 次，即使是对于这个粗糙的网格！幸运的是，filt1 可以更有效地做到这一点。如果因为 sst 是一个三维矩阵，filt1 已经知道在第三维度下操作，尽管您可以指定'dim'，3 如果您想额外确定的话。

下面是如何使用 filt1 过滤整个三维网格 SST 时间序列：

```
sst_lp = filt1('lp',sst,'Tc',18);
```

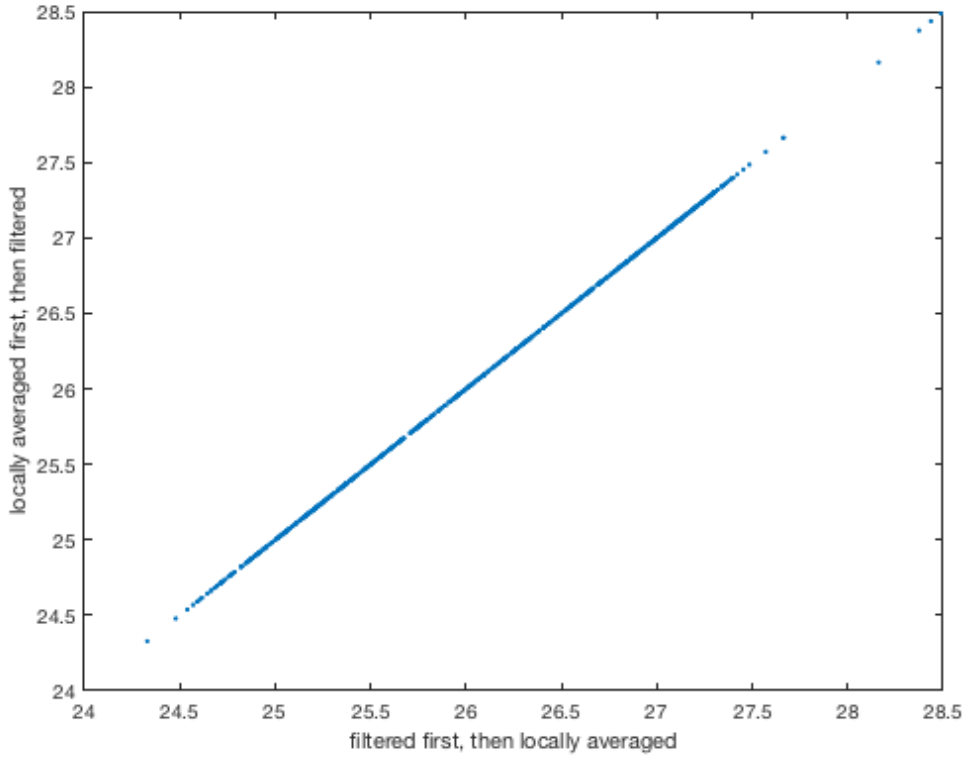
现在我们可以看看*过滤后*的三维 sst 数据的本地墨西哥湾时间序列，它应该完全匹配：

```
sst_lp_gom = local(sst_lp,mask,'omitnan');

plot(t,sst_lp_gom)
legend('original',...
 'lowpass filtered',...
 'lowpass 5th order',...
 'local after 1st order filter')
```



事实上，取过滤后的 SST 网格的局部平均时间序列得到的结果与过滤本地时间序列的结果相同。这是一个直接的比较，因为上面的图有点忙：

```
figure

plot(sst_lp_gom,sst_gom_lp,'.')
xlabel 'filtered first, then locally averaged'
ylabel 'locally averaged first, then filtered'
```



两者之间的区别只是数字噪音。

作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

scatstat1　文档（参见前文）

doy 文档

doy 函数返回每年的儒略日。
另请参见 day.

语法

```
n = doy(t)
n = doy(t,'decimalyear')
n = doy(t,'remdecimalyear')
```

说明

n = doy(t) 给出了与 t 给出的日期相对应的一年中的某一天（从 1 到 366.999）。输入日期可以是 datenum、datetime 或字符串格式。
n = doy(t,'decimalyear') 以输入日期 t 的十进制形式给出年份。它代表闰年，所以给定日期的十进制值将取决于它是否是闰年。例如，2016 年 7 月 4 日（闰年）为 2016.5082，而 2017 年 7 月 4 日（非闰年）为 2017.5068。
n = doy(t,'remdecimalyear') 只返回十进制年份的剩余部分，并且始终在 0 到 1 的范围内。、

示例 1: datestr 格式

每年情人节是哪一儒略日？

```
doy('february 14')
```

```
ans =

        45.00
```

45.那是因为一月份有 31 天，情人节是 2 月 14 日。

示例 2: datenum 格式

当我编写这个示例文件时，它是
>>now

```
ans =

    737427.95
```

也就是说，从今年 1 月 1 日零时起，已经有 737427 天了。那么现在是一年中的哪一天呢？

```
doy(737427.95)
```

```
ans =

         2.95
```

如你所见，现在是 1 月 2 日深夜。一天只剩下 5%

示例 3: datetime 格式

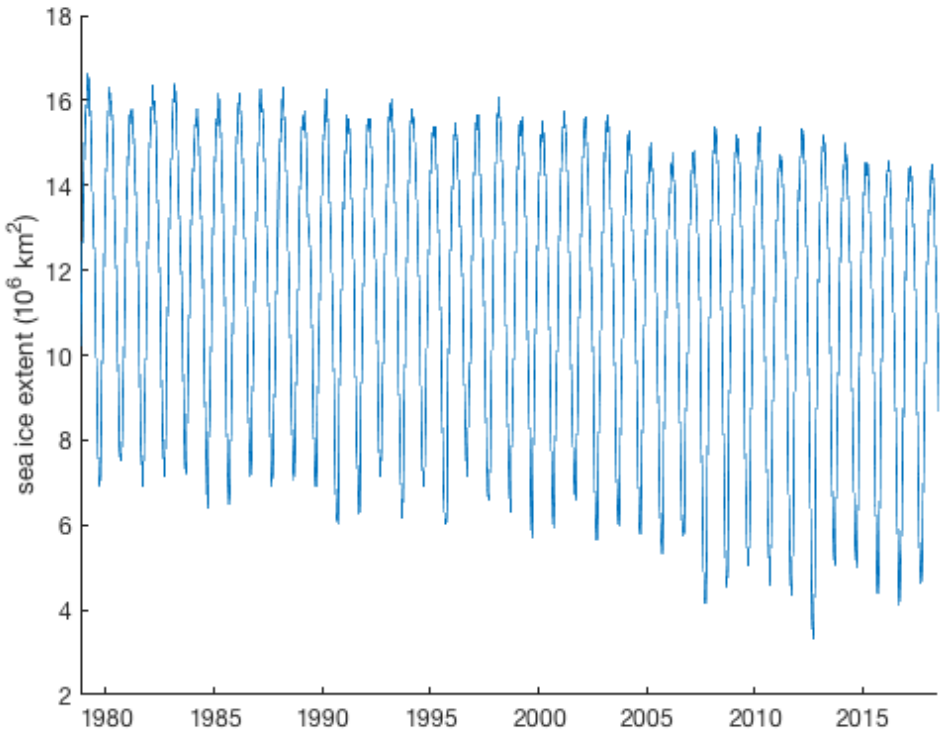
考虑海冰范围数据：

```
load seaice_extent.mat

whos extent_N t % 显示这些变量的尺寸
```

```
  Name              Size              Bytes  Class       Attributes


  extent_N      12854x1             102832  double

  t             12854x1             205665  datetime
```

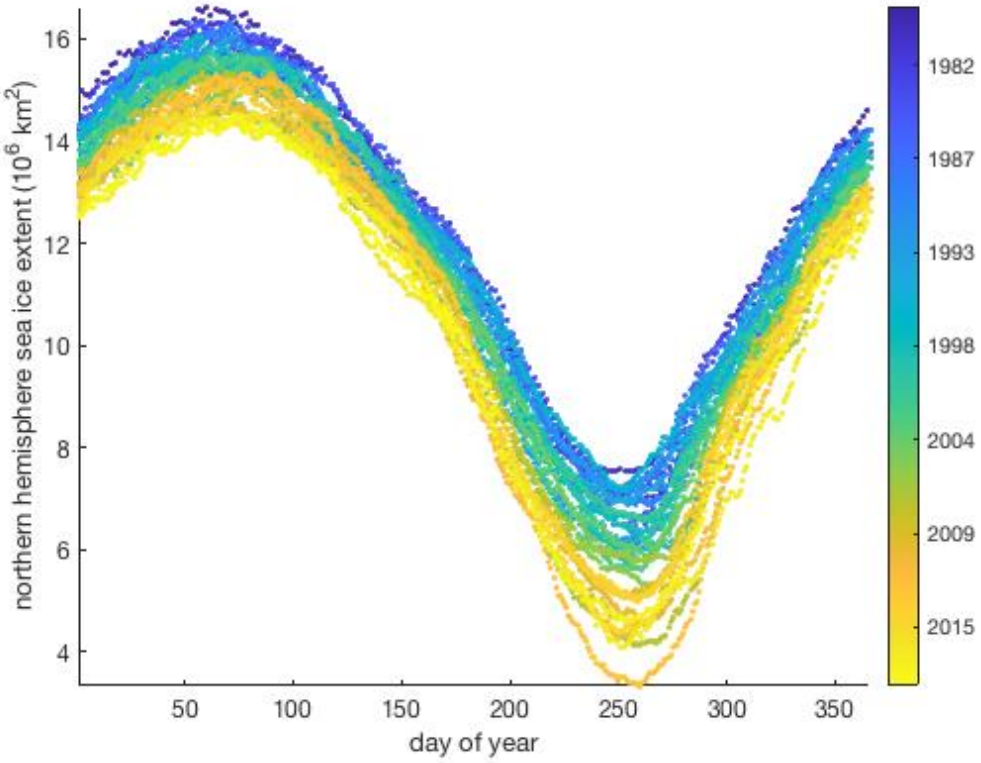
变量 t 是 datetime 格式的，它包含一个日期，从 1978 年到 2018 年，几乎每一天。以下是时间序列：

```
plot(t,extent_N)

ylabel 'sea ice extent (10^6 km^2)'
box off % 移除丑丑的外框
```



像这样绘制数据作为儒略日的函数

```
jday = doy(t);



scatter(jday,extent_N,10,datenum(t),'filled')
cb = cbdate('yyyy');     % 颜色条定为日数
set(cb,'ydir','reverse') % 翻转颜色条
axis tight
ylabel 'northern hemisphere sea ice extent (10^6 km^2)'
xlabel 'day of year'
```



作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

reshapetimeseries 文档（参见前文）

不确定度量化

- mann_kendall 执行标准简单的 Mann-Kendall 检验以确定是否存在显着趋势。
- ts_normstrap 对时间序列执行自举不确定性分析，假设每个步骤的不确定性值为正态概率分布。
- sinefit_bootstrap 对由 sinefit 估计的参数执行引导分析。

mann_kendall 文档

mann_kendall 执行标准简单的 Mann-Kendall 检验以确定是否存在显着趋势。(需要 Statistics Toolbox)

语法

```
h = mann_kendall(y)
h = mann_kendall(y,alpha)
h = mann_kendall(...,'dim',dim)
[h,p] = mann_kendall(...)
```

说明

h = mann_kendall(y) 对时间序列 y 执行标准的简单 Mann-Kendall 检验以确定是否存在显着趋势。如果 h 为真，则趋势存在；如果 h 为假，您可以拒绝趋势假设。此函数假设 y 的采样时间相同。

h = mann_kendall(y,alpha) 指定 0 到 1 范围内的 alpha 显著性水平。默认 alpha 为 0.05，对应于 5%的显著性水平。

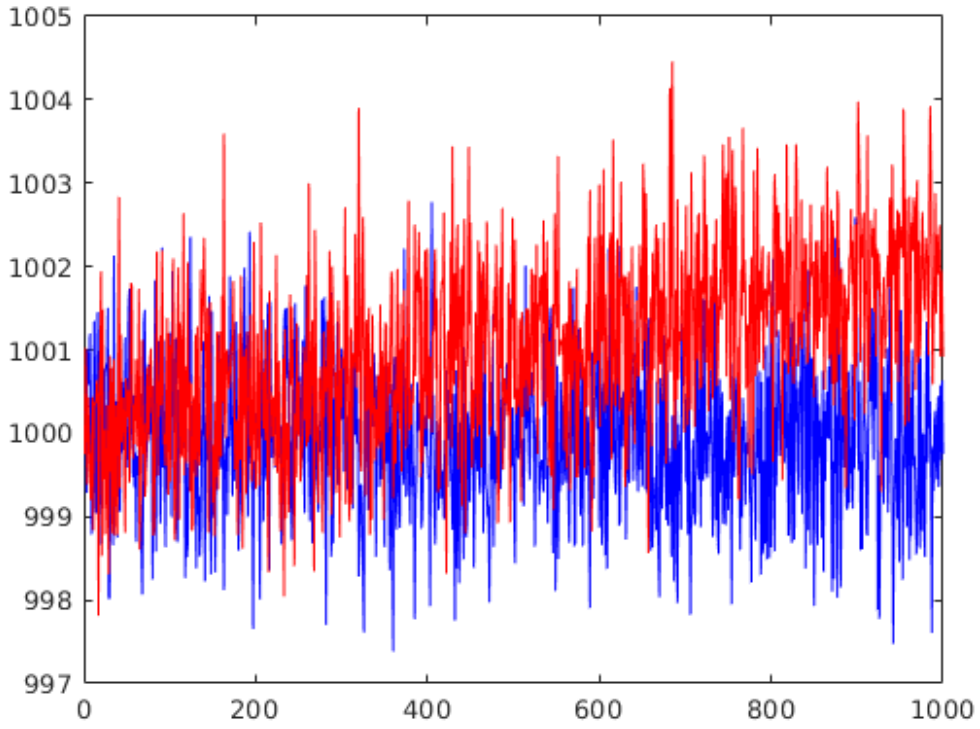
h = mann_kendall(...,'dim',dim) 指定计算趋势的维度。默认情况下，如果 y 是一维数组，则沿 y 的第一个非单一维度计算趋势；如果 y 是二维矩阵，则趋势向下计算 y 的行（维度 1）；如果 y 是三维矩阵，则向下计算第 3 维的趋势。

[h,p] = mann_kendall(...) 还返回趋势的 p 值。

示例 1: 一维数组

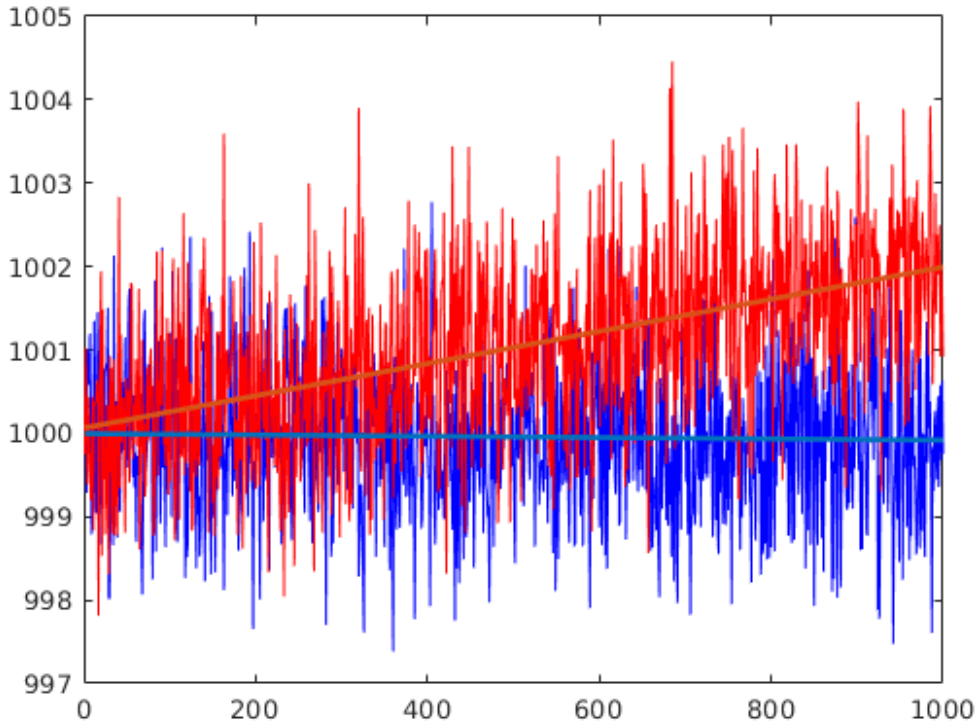
考虑这两个数组，一个有趋势，一个没有：

```
x = (1:1000)';

y0 = randn(size(x)) + 1000;          % 没有趋势的随机数据
y1 = randn(size(x)) + 1000 + x/500; % 带有 1/500 趋势的随机数据

plot(x,y0,'b');
hold on
plot(x,y1,'r');
```



为了帮助显示趋势，使用 polyplot，它将绘制 y0 和 y1 的一阶多项式趋势线。

```
polyplot(x,y0,1,'linewidth',2)
polyplot(x,y1,1,'linewidth',2)
```



为了验证 y0 和 y1 中趋势的大小，请使用 trend 趋势函数：

```
trend(y0)
```

```
ans =

  -7.7958e-05
```

...正如预期的那样，y0 的趋势大约为零。现在检查 y1：

```
trend(y1)
```

```
ans =

    0.0019
```

...正如预期的那样，y1 的趋势是我们有意添加的 1/500。这两个趋势幅度似乎都是接近于零的小数字，但并不完全为零。那么这些趋势中的任何一个都重要吗？使用 mann_kendall 找出，从 y0 开始：

```
mann_kendall(y0)
```

```
ans =

  logical

0
```

false（逻辑 0）证实了这一点：y0 时间序列只包含噪音。现在检查 y1，回想一下，我们在其中添加了 1/500 的趋势：

```
mann_kendall(y1)
```

```
ans =

  logical

   1
```

逻辑 1 或 true 答案证实，尽管 y1 包含噪声并且其趋势在幅度上很小，但趋势仍然存在于默认的 alpha=5% 显著性水平。5%的重要性对你来说还不够严格吗？像这样将其拧紧至 0.1%：

```
mann_kendall(y1,0.001)
```

```
ans =
```

```
  logical

   1
```

这证实了 y1 中的趋势以 0.1%的显著性存在。

示例 2: 多个时间序列

如果您在二维矩阵中有多个时间序列，则 mann_kendall 函数可以一次对它们进行操作。例如，我们将使用示例 1 中的两个一维数组时间序列创建数据集 D：

```
D = [y0 y1];
```

默认情况下，如果 mann_kendall 的输入是二维矩阵，则该函数将操作行，因此测试 y0 和 y1 趋势的显着性很容易：

```
mann_kendall(D)
```

```
ans =

  1×2 logical array

   0   1
```

0 和 1 答案表示 D 的第一列没有显著趋势，但 D 的第二列有显著趋势。
如果每个时间序列都在自己的行中而不是在自己的列中，请像这样指定操作的维度：

```
Dt = D'; % 转置 D 以使时间穿过 Dt 的列.

mann_kendall(Dt,'dim',2)
```

```
ans =

  2×1 logical array

   0

   1
```

也可以指定显著性水平 alpha。让我们*真正*放宽我们的标准，并将其设置为 99.999%：

```
mann_kendall(Dt,0.99999,'dim',2)
```

```
ans =

  2×1 logical array

   1
```

```
1
```

哇，如果我们足够放松我们的标准，即使是 y0 噪声也似乎包含一个重要趋势！

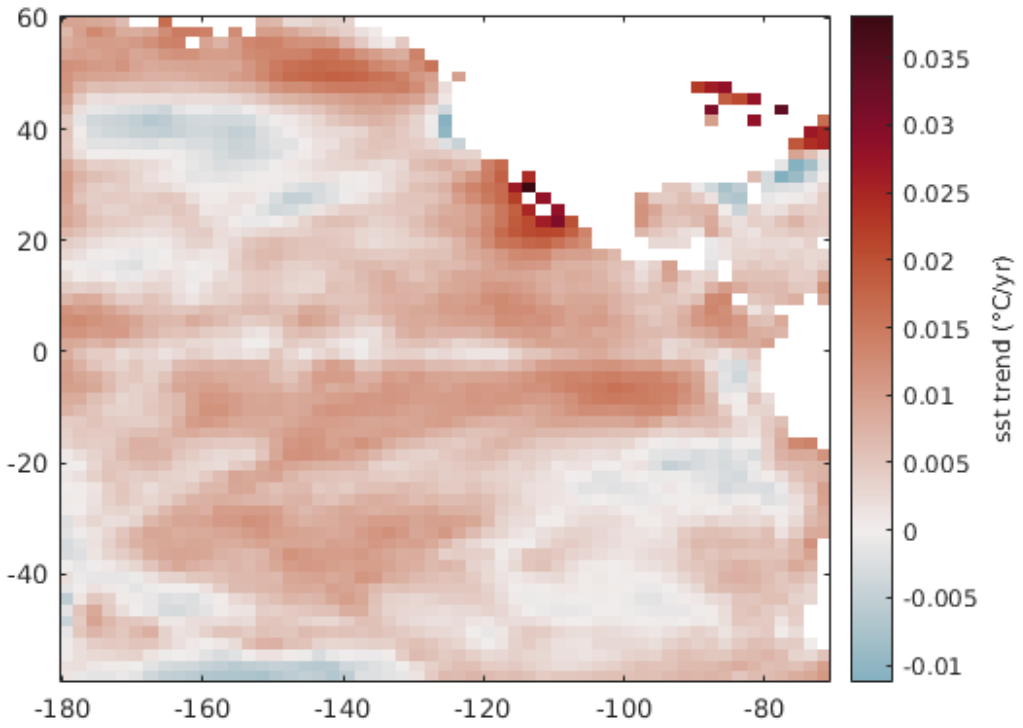
示例 3:三维数据

在过去的几十年里，海面温度是否发生了显著变化？要回答这个问题，请加载 60x55x802pacific_sst 数据集，其中包含从 1950 年到 2016 年的 802 个每月海面温度的网格：

```
load pacific_sst
```

以每年 12 次（每月数据）的采样率，使用 trend 函数计算每年度数的海温趋势，并使用 imagescn 制作地图。 使用 cmocean 设置颜色图：

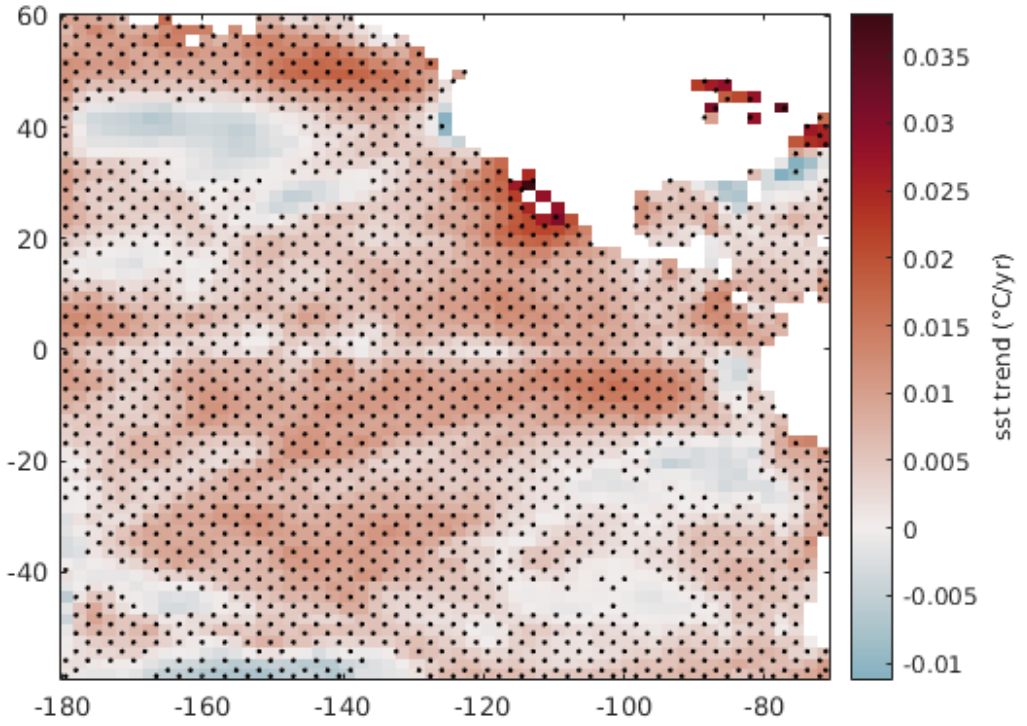
```
% 计算 SST 趋势:
tr = trend(sst,12);

% 绘制趋势:
figure
imagescn(lon,lat,tr)
cb = colorbar;
ylabel(cb,'sst trend (\circC/yr)')
cmocean('balance','pivot')
```



上面的地图显示，在大多数地方，海洋似乎正在变暖。但这种趋势重要吗？使用 mann_kendall 函数找出并使用 stipple 函数绘制重要区域：

```
significant = mann_kendall(sst); % (可能会等几秒)
```

```
hold on
stipple(lon,lat,significant)
```



上图显示 SST 趋势在大多数地方都很显著，在 alpha=5%的水平上。感觉可以随意尝试更严格的标准。想知道季节可能对计算产生什么影响？尝试使用 deseason 从 sst 数据集中删除季节性周期并重新计算趋势和重要性。

参考文献

Mann, H. B. (1945), Nonparametric tests against trend, Econometrica, 13, 245-259.

Kendall, M. G. (1975), Rank Correlation Methods, Griffin, London.

作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的，改编自 Simone Fatichi 的 Mann_Kendall 函数。

ts_normstrap 文档

ts_normstrap 在假设正态概率分布的每个步骤中给定不确定值的时间序列上执行独立引导不确定性分析。独立引导意味着在给定概率分布的特定不确定性范围内估计每个时间点的值。最终，目标是生成时间序列的多个实现，并在每个时间步提供置信区间。

语法

```
tsb = ts_normstrap(ts)
tsb = ts_normstrap(ts,e)
tsb = ts_normstrap(ts,E)
tsb = ts_normstrap(ts,'nboot',nboot)
[tsb,Nts] = ts_normstrap(...)
```

说明

`tsb = ts_normstrap(ts)` 计算给定时间序列的置信区间，假设整个时间序列 ts 的不确定性为 1 个标准偏差，则在每个点以正态概率对向量 ts 随机子采样 1000 次后计算置信区间。输出 tbs 是一个 `length(ts)`x2 大小的矩阵，其中包含时间序列 ts 的+/-1 标准偏差范围。请注意，ts 是一个没有时间维度的向量，因为边界是在查询点返回的。

`tsb = ts_normstrap(ts,e)` 指定不确定性值 e，从该值计算向量 ts 中每个步骤的不确定性分布，从而覆盖 ts 的 1 个标准偏差的默认值。

`tsb = ts_normstrap(ts,E)` 指定一个向量 E，其中包含计算向量 ts 中的不确定性分布的每个步骤的不确定性值。

`tsb = ts_normstrap(...,'nboot',nboot)` 指定独立引导样本的数量。默认值为 1000，表示计算 1000 个随机时间序列。

`[tsb,Nts] = ts_normstrap(...)` 还返回具有给定不确定性的 1000 个（或指定数量）随机生成的时间序列。

`[tsb,Nts] = ts_normstrap(...)` 还返回 1000 个（或指定数量的）随机生成的具有指定不确定性子采样的时间序列。.

示例

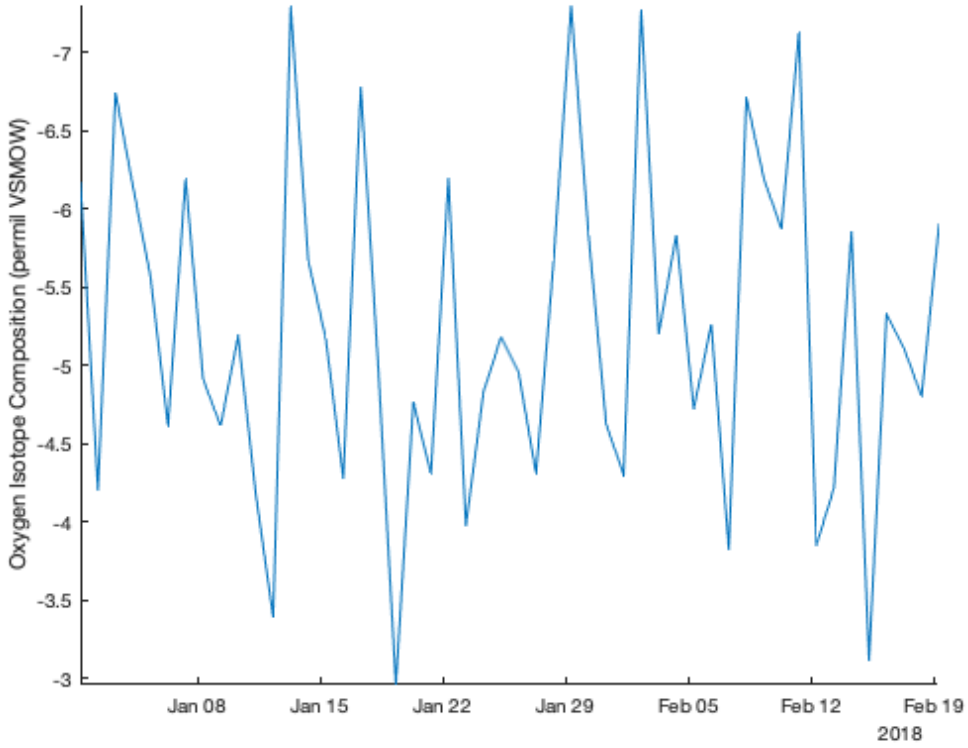
此示例对随机生成的 50 个点的时间序列执行自举分析，我们将假设这是水中氧同位素的 50 个测量值，平均值约为 -5‰VSMOW。

```
iso_ts = -5 + randn(50,1);



% 假设他们在 2018 年连续采样超过 50 天
t1 = datetime(2018,1,1,8,0,0);
t = t1:t1+49;
```

让我们绘制数据

```
figure

plot(t,iso_ts)
```

```
box off
axis tight
ylabel 'Oxygen Isotope Composition (permil VSMOW)'
set(gca,'ydir','reverse') % 翻转 Y 轴的方向
```

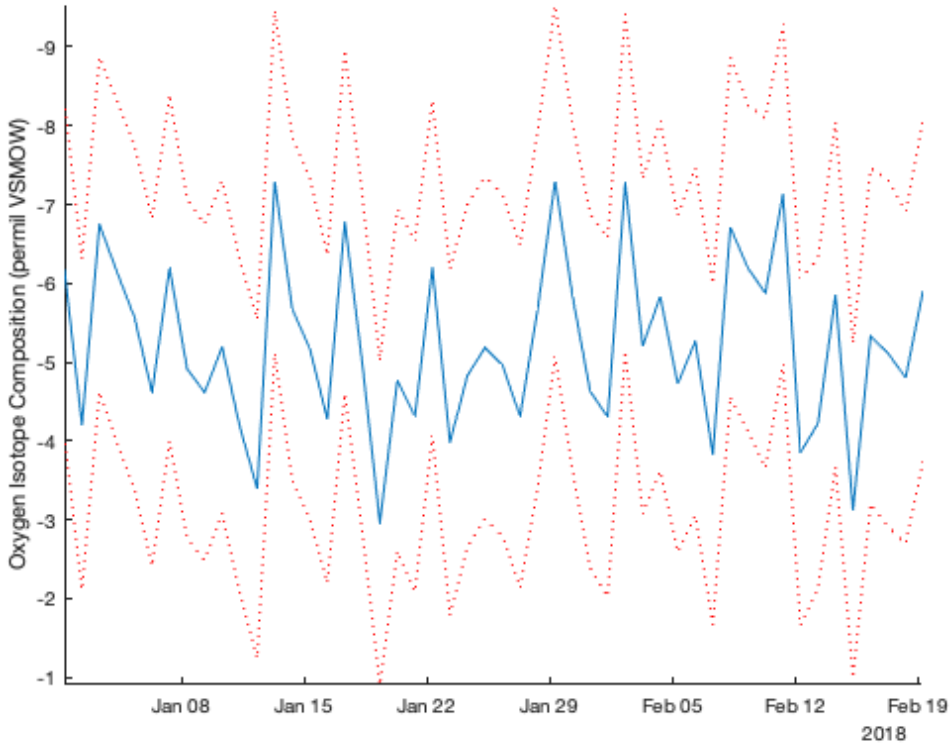


现在我们想要时间序列的不确定性界限

```
tsb = ts_normstrap(iso_ts);



% 默认情况下，这将给出接近整体 |iso_ts|时间序列标准偏差的不确定性界限。
overall_sd = std(iso_ts)
default_bootstrap_uncertainty = mean(tsb)

% 让我们在原始图上将其绘制为 2-sigma 界限（乘以 2）、
hold on;
plot(t,iso_ts+2.*tsb,':r')
plot(t,iso_ts-2.*tsb,':r')
```

```
overall_sd =

    1.0830

default_bootstrap_uncertainty =
```

1.0817



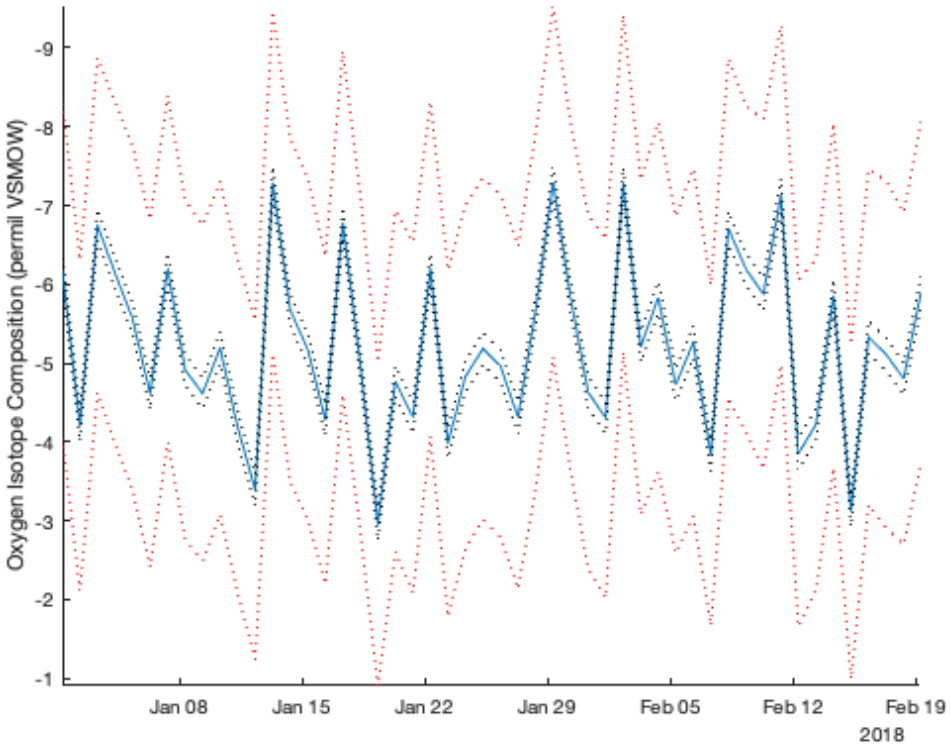
现在让我们更具体一点，将氧同位素测量的分析不确定度指定为 0.1‰

```
tsb = ts_normstrap(iso_ts,0.1);



specified_bootstrap_uncertainty = mean(tsb)
```

```
specified_bootstrap_uncertainty =

    0.1002
```

让我们绘制新的更好的不确定性，也作为 2-sigma 界限

```
hold on;
plot(t,iso_ts+2.*tsb,':k')
plot(t,iso_ts-2.*tsb,':k')
```



现在让我们将少量独立引导样本与大量样本进行比较、

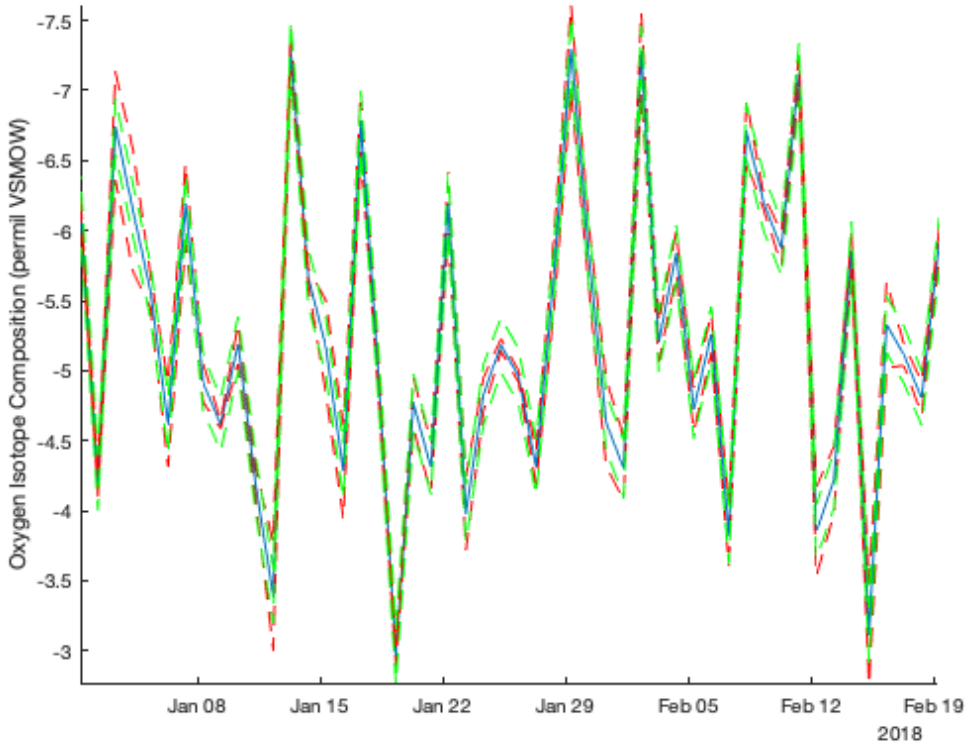
```
tsb_low = ts_normstrap(iso_ts,0.1,'nboot',3);
tsb_high = ts_normstrap(iso_ts,0.1,'nboot',500);

low_bootstrap_uncertainty = mean(tsb_low)
high_bootstrap_uncertainty = mean(tsb_high)
```

```
low_bootstrap_uncertainty =

    0.0956

high_bootstrap_uncertainty =

    0.0997
```

两个一起画：

```
figure; hold on;
plot(t,iso_ts)
box off
axis tight
ylabel 'Oxygen Isotope Composition (permil VSMOW)'
set(gca,'ydir','reverse') % 翻转 Y 轴方向
```

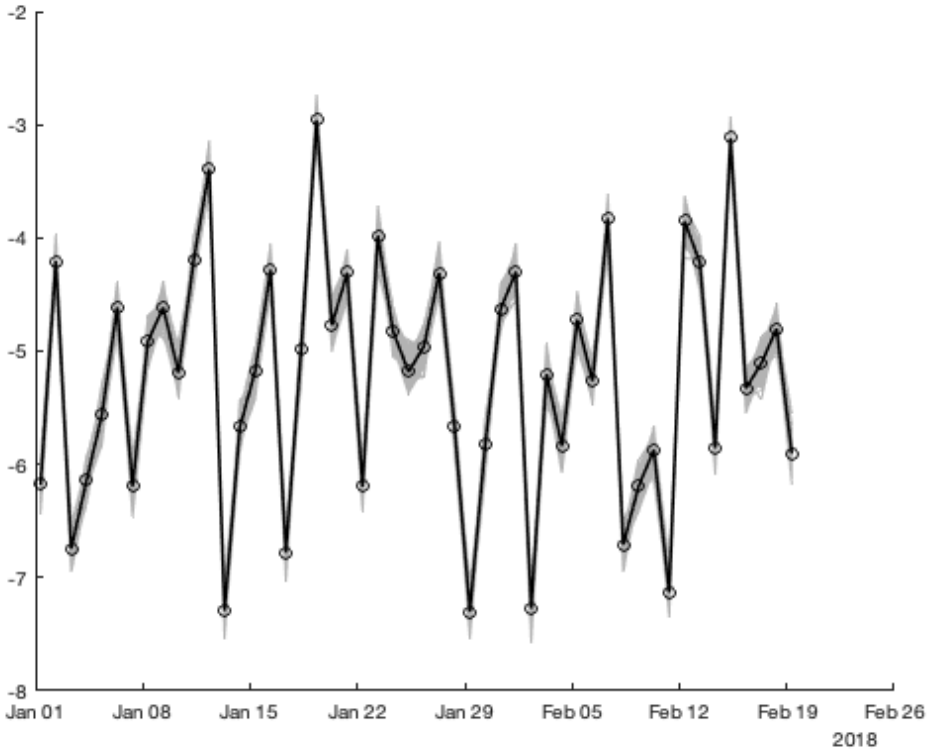
```
plot(t,iso_ts+2.*tsb_low,'--r')
plot(t,iso_ts-2.*tsb_low,'--r')

plot(t,iso_ts+2.*tsb_high,'--g')
plot(t,iso_ts-2.*tsb_high,'--g')
```



请注意，有时较低的数字表示较高的不确定性，有时则表示较低的不确定性。 请记住，大量的独立引导样本通常会收敛到某个值。

为了好玩，让我们绘制我们生成的所有时间序列！

```
[tsb,Nts] = ts_normstrap(iso_ts,0.08,'nboot',500);

figure;hold on;
plot(t,Nts,'color',[0.7 0.7 0.7]) % 灰色
plot(t,iso_ts,'ko-','linewidth',1.5); % 黑色
```



参考文献

原始独立引导程序被正式引入文献： Eforon in 1979: Efron, B., 1979: Bootstrap methods: another look at the jackknife. Ann. Stat. 7, 1-26. doi:10.1007/978-1-4612-4380-9_41.

虽然这是一个相当密集的阅读！ 以下是一些（古气候）应用：

Thirumalai, K., T. M. Quinn, and G. Marino, 2016: Constraining past seawater delta-18-O and temperature records developed from foraminiferal geochemistry, Paleoceanography doi:10.1002/2016PA002970.

Carré, M., J. P. Sachs, J. M. Wallace, and C. Favier, 2012: Exploring errors in paleoclimate proxy reconstructions using Monte Carlo simulations: paleotemperature from mollusk and coral geochemistry, Clim. Past, 8(2), 433-450. doi:10.5194/cp-8-433-2012.

这两篇论文中的第一幅图提供了一个流程图，用于理解年龄不确定性中引导程序的基本原理图。

作者简介

ts_normstrap 函数和支持文档是亚利桑那大学的 Kaustubh Thirumalai 为 Climate Data Toolbox for Matlab 中 2019 年 1 月写的。

sinefit_bootstrap 文档

sinefit_bootstrap 对函数 sinefit 估计的参数执行引导分析。这意味着将正弦拟合函数应用于一组数据子样本，然后分析每个参数的解的分布，以查看解的鲁棒性。
相关函数，参阅 sineval 和 sinefit 文档。

语法

```
ft = sinefit_bootstrap(t,y)
ft = sinefit_bootstrap(...,'weight',weights)
ft = sinefit_bootstrap(...,'terms',TermOption)
ft = sinefit_bootstrap(...,'nboot',nboot)
[ft,rmse,Nsamp] = sinefit_bootstrap(...)
```

说明

ft = sinefit_bootstrap(t,y) 将 2 项（幅度和相位）正弦曲线拟合到时间序列 t,y 的 1000 个随机子样本。 输出 ft 是一个 1000x2 矩阵，分别包含幅度和相位的所有 1000 个解。 有关输入和输出的完整说明，请参阅 sinefit。
ft = sinefit_bootstrap(...,'weight',w) 对每个观测值 y 应用加权。 例如，如果形式误差 err 与 y 相关联，您可以让 w = 1./err.^2。 默认情况下 w = ones(size(y))。
ft = sinefit_bootstrap(...,'terms',TermOption) 指定在正弦拟合中计算哪些项。 默认为 2，因为更多的项在计算上可能会很慢！ TermOption 可以是 2、3、4 或 5：

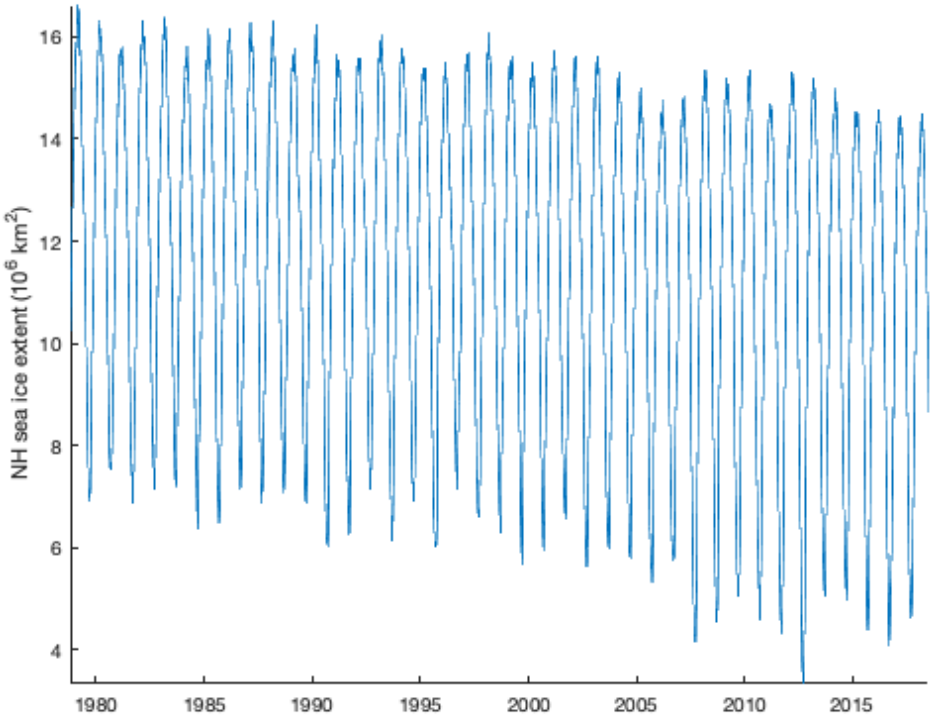
- 2: ft = [A doy_max] 其中 A 是正弦波的振幅，doy_max 是正弦波最大值对应的年份。默认的 TermOption 是 2。
- 3: ft = [A doy_max C] 也估计 C，一个常数偏移量。求解会增加处理时间，因此您可能更喜欢将 C 简单地估计为输入 y 的平均值。但是，如果您不能假设 C=mean(y)，您可能更喜欢这个三项解决方案。
- 4: ft = [A doy_max C trend] 还以每年 y 为单位估计整个时间序列的线性趋势。同样，同时求解四个项比求解两个项在计算上要昂贵得多，因此您可能更喜欢使用 polyfit 自行估计趋势，然后计算去趋势数据的两项正弦拟合。.
- 5: ft = [A doy_max C trend quadratic_term] 在解决方案中还包括一个二次项，但目前这是实验性的，因为将多项式拟合到参考零年的日期往往缩放效果不佳。

ft = sinefit_bootstrap(...,'nboot',nboot) 指定引导样本的数量。 默认值为 1000，这意味着正弦曲线适合数据的 1000 个随机子样本。
[ft,rmse,Nsamp] = sinefit_bootstrap(...) 还返回残差的均方根误差分布（正弦曲线拟合数据的程度）和 Nsamp，即对数据的每个子样本有贡献的数据点数。

示例

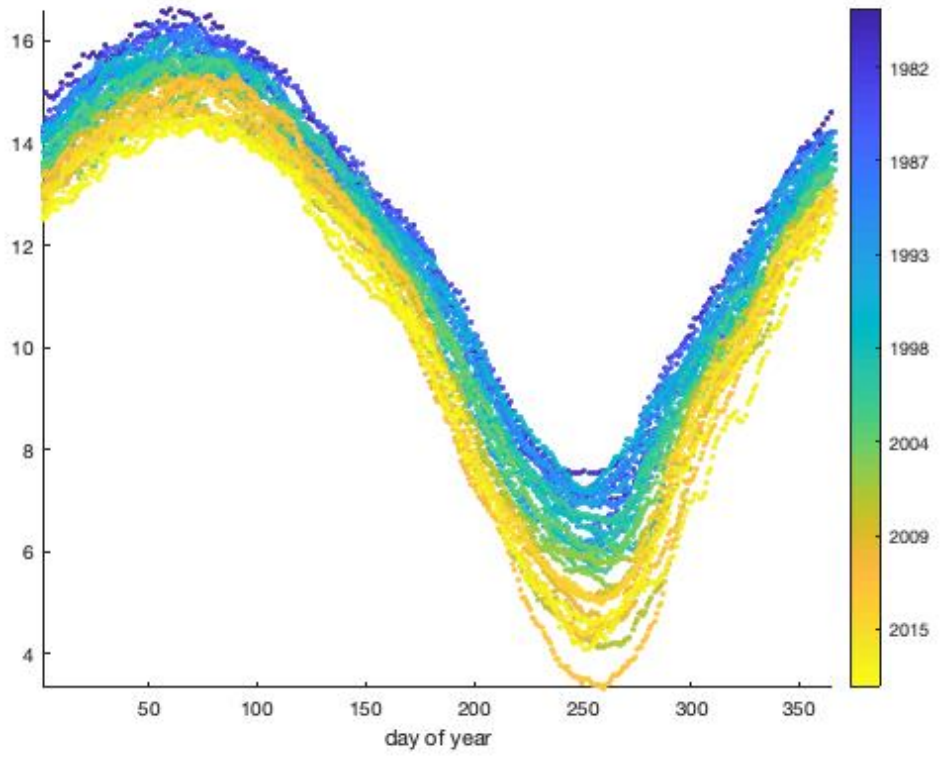
此示例对海冰范围数据集的相位和幅度执行自举分析。载入并绘制北半球海冰范围时间序列：

```
load seaice_extent

plot(t,extent_N)
box off
axis tight
ylabel 'NH sea ice extent (10^6 km^2)'
```



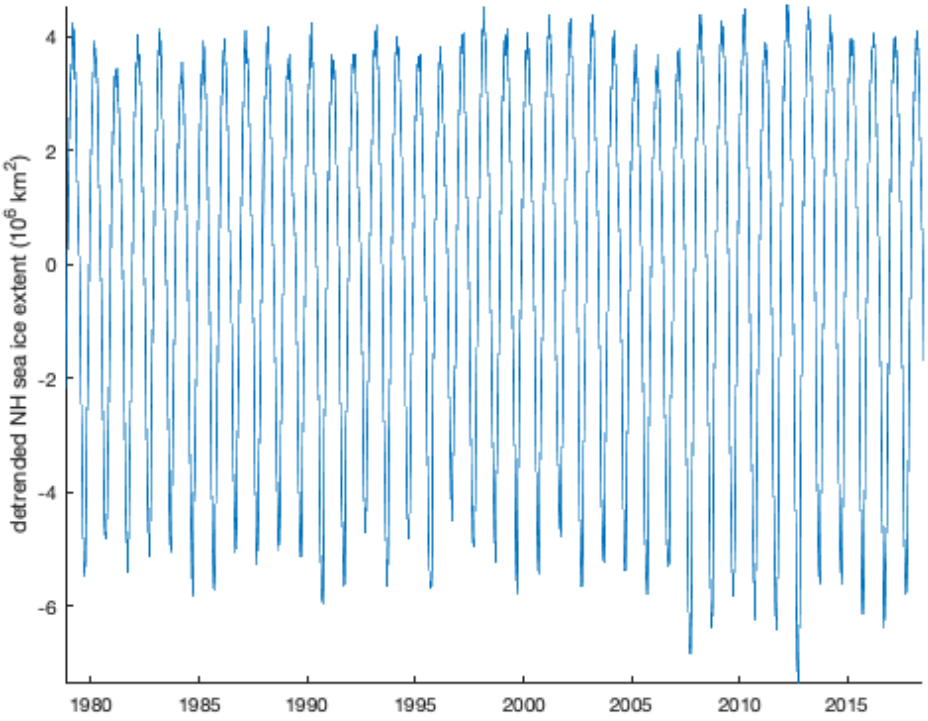
嗯，这就像解释条形码一样。尝试将绘图作为一年中的某一天的函数，以便更好地描述季节性周期。我将 cbdate 函数用于日期格式的颜色条。

```
figure

scatter(day(t,'dayofyear'),extent_N,10,datenum(t),'filled')
axis tight
box off
xlabel 'day of year'
cb = cbdate('yyyy');
set(cb,'ydir','reverse') % 翻转色条方向
```



现在 sinefit 函数可以一次估计该时间序列的幅度、相位、y 轴截距和斜率；然而，每增加一项，计算就会变得相当慢，并且 sinefit_bootstrap 将不得不进行一千次计算，所以为了简单起见，现在让我们对数据进行去趋势，只求解相位和幅度。去趋势化的时间序列如下所示：

```
y = detrend(extent_N);



figure

plot(t,y)

box off
axis tight
ylabel 'detrended NH sea ice extent (10^6 km^2)'
```



肉眼可见正弦波幅值约为 400 万平方公里，这是北半球海冰的范围，因此我们可以猜测其最大值将出现在北半球冬季或春季末。我们可以将 2 项正弦曲线拟合到完整的时间序列中，以获得比眼球估计更好的结果

```
sinefit(t,y)
```

```
ans =

           4.41          66.83
```

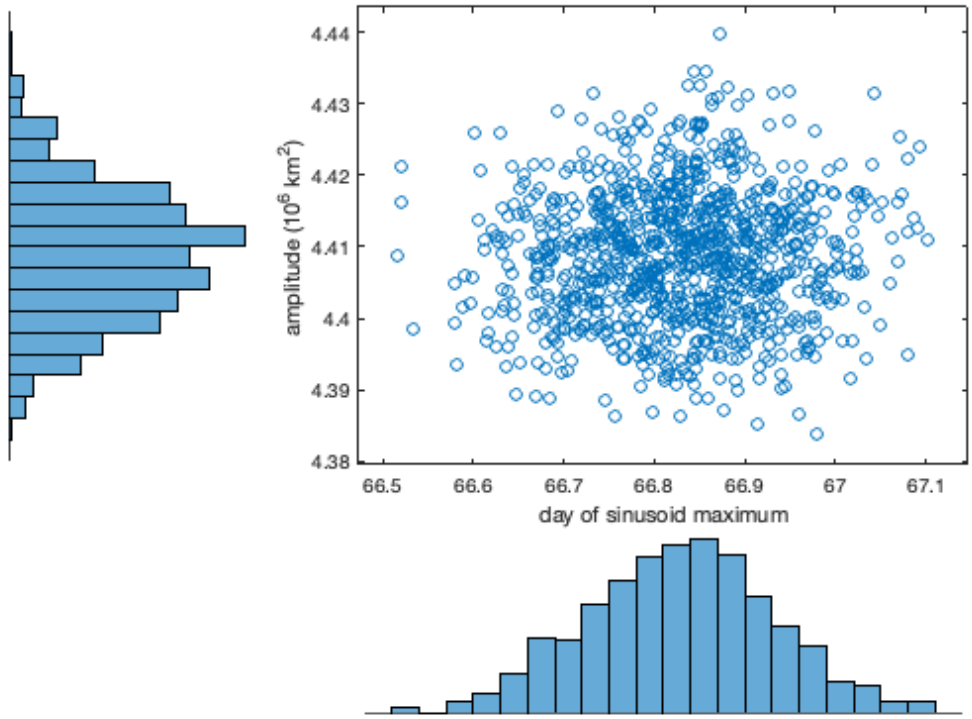
这两个数字 4.4 和 66.8 是正弦曲线的振幅和正弦曲线达到最大值的日期。也就是说，440 万平方公里和第 66 天（3 月 7 日）。但该解决方案的稳健性如何？使用 sinefit_bootstrap 将正弦曲线拟合到海冰范围时间序列的 1000 个随机样本（此解决方案大约需要 20 秒）

```
ft = sinefit_bootstrap(t,y);



amp = ft(:,1);  % 振幅是 ft 的第一列
phase = ft(:,2);% 相位是 ft 的第二列
```

如果您有统计工具箱，则可以使用 scatterhist 来显示幅度和相位解的分布。否则，您可以将它们绘制为二维散点图和/或绘制数据各个分布的直方图：

```
scatterhist(phase,amp)
```

```
xlabel 'day of sinusoid maximum'
ylabel 'amplitude (10^6 km^2)'
```



这些是非常紧密的集群。他们说正弦拟合的幅度和相位应该精确到大约

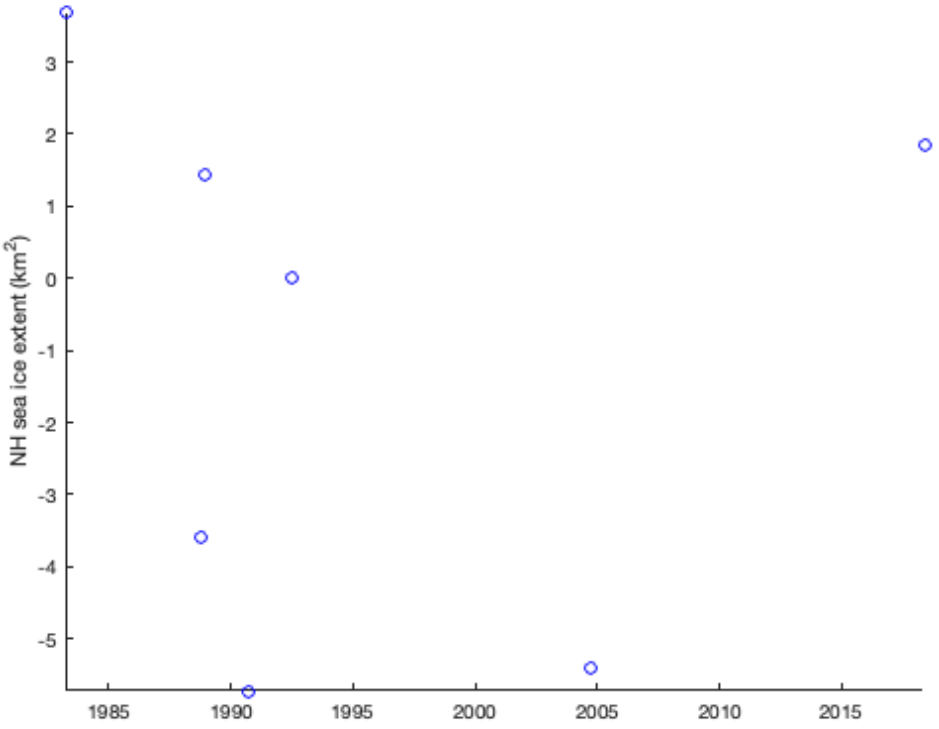
```
std(amp)

std(phase)
```

```
ans =

    0.01

ans =

    0.10
```

0.009 平方公里和 0.1 天。这并不奇怪，因为我们有 40 年（40 个周期）的非常好的采样（365/周期）数据集，我们期望正弦振幅和相位的高精度。但是，如果您只有几个随机收集的数据点呢？你还能估计正弦曲线吗？这就是我的意思：让我们将数据集缩减为仅 7 个随机点：

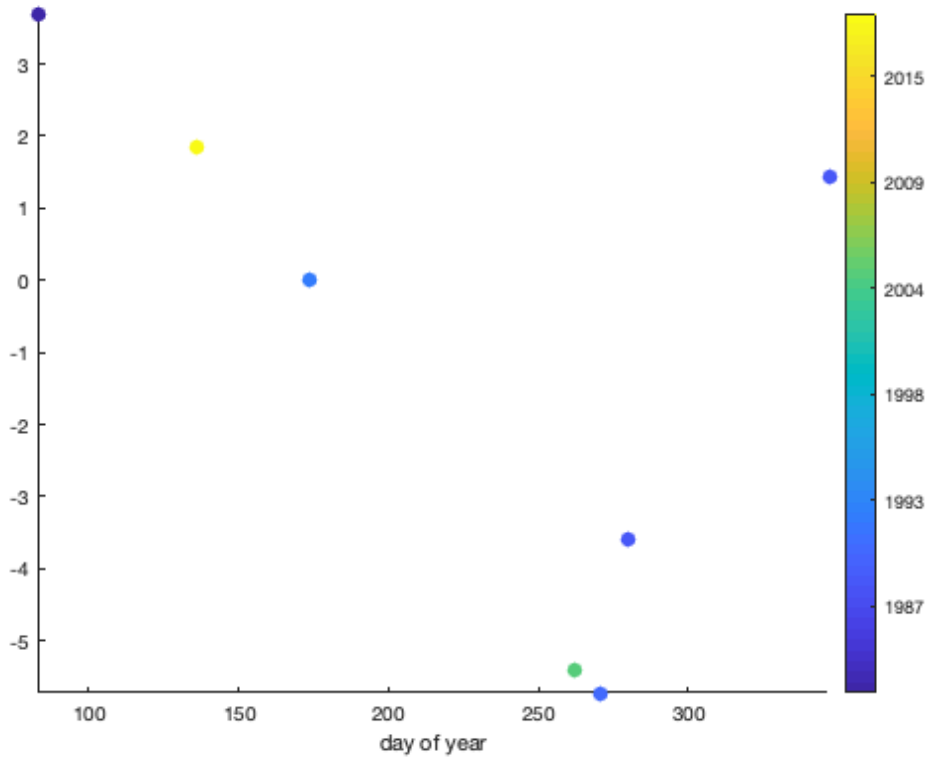
```
% 7 个随机数据点的索引:
ind = randi(length(y),[7 1]);

% 将 y 和 t 修剪为仅这 7 个点:
```

```
y = y(ind);
t = t(ind);

figure
plot(t,y,'bo')
axis tight
box off
ylabel 'NH sea ice extent (km^2)'
```



那看起来只是一些随机点。但再一次，绘制为一年中的某一天的函数可以帮助我们了解 sinefit 函数的作用：

```
figure

scatter(day(t,'dayofyear'),y,60,datenum(t),'filled')
axis tight
box off
xlabel 'day of year'
colorbar
cbdate('yyyy');
```



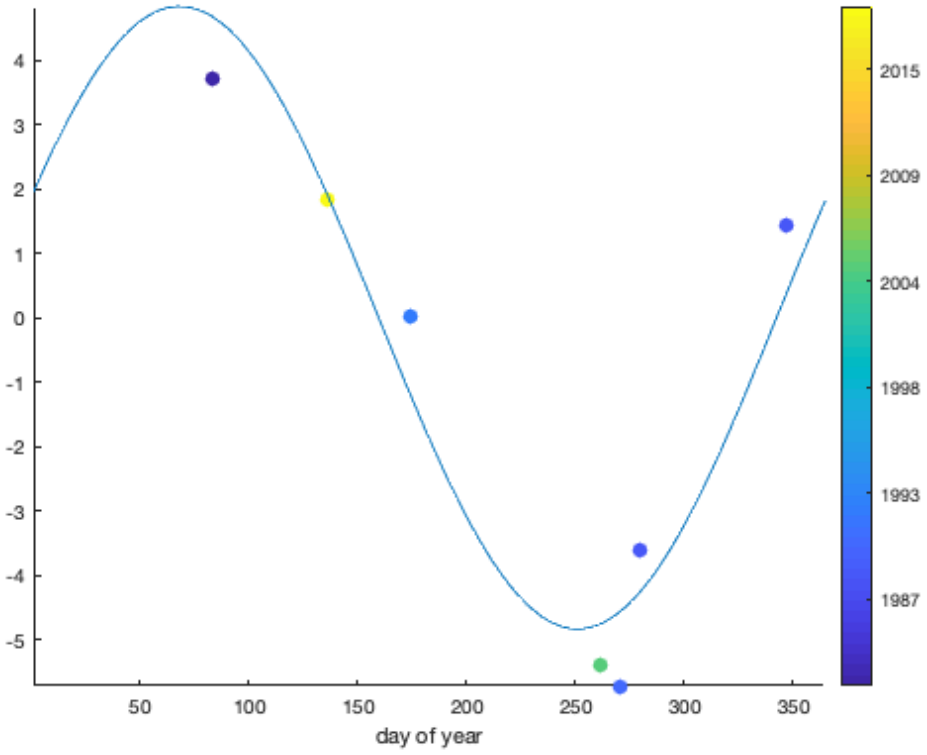
所以我们看到 sinefit 有类似正弦曲线的东西可以近似，即使只有 7 个点。仅凭这 7 个点，sinefit 的值仍然非常接近约束良好的 4.4 振幅，最大值出现在第 66.8 天也就不足为奇了

```
ft = sinefit(t,y)
```

```
ft =

            4.84            68.09
```

以下是这 7 个点之上的正弦曲线：

```
hold on
plot(1:365,sineval(ft,1:365))
```

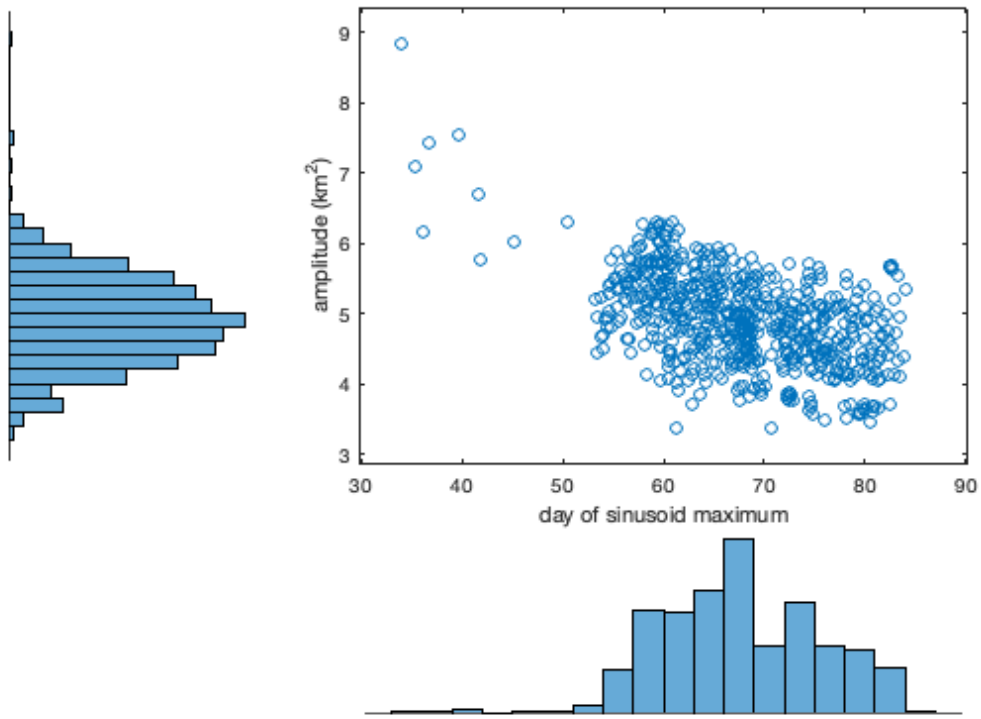


使用 sinefit_bootstrap 量化不确定性。

```
ft = sinefit_bootstrap(t,y);



figure

scatterhist(ft(:,2),ft(:,1))

xlabel 'day of sinusoid maximum'
ylabel 'amplitude (km^2)'

amp_uncertainty = std(ft(:,1))
phase_uncertainty = std(ft(:,2))
```

```
Warning: Fitting a sinusoid to less than one year of data. This might not be

what you want.

amp_uncertainty =

          0.62

phase_uncertainty =
```

7.87



事实上，即使只收集了 40 年的 7 个数据点，sinefit 也可以估计大约 40 万平方公里内的振幅和大约 7 天内的相位。

友情提示

与 sinefit 估计的所有参数相关的一个简短说明：这些参数描述了最佳拟合正弦曲线，但这并不一定意味着它们完全描述了基础数据本身的行为。例如，就气候平均值而言，北半球海冰范围实际上通常在第 71 天（3 月 12 日）左右达到最大值，而 sinefit 表示最适合正弦曲线的最大值出现在第 66 天（3 月 7 日）。这是因为海冰范围的真实行为比简单的正弦曲线更复杂。在您的工作中，一定要考虑真实行为与真实行为的 1/yr 频率分量之间的差异。

作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。`sinefit`,`sineval`,和 `sinefit_bootstrap` 函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

气候指数

- enso 计算厄尔尼诺南方涛动指数。
- sam 根据海平面气压计算南方环模指数。
- nao 根据 Hurrell, 1995 提出的定义，根据海平面气压计算北大西洋涛动指数。（相当于北极涛动或北方环形模式）。
- amo 根据 Enfield 等人 2001 年提出的定义，根据海面温度计算大西洋多年代际振荡指数。
- pet 根据 Hargreaves-Samani 给出了潜在的参考蒸发量。
- spei 计算标准化降水蒸散指数。

enso 文档

enso 计算 Nino 3.4 SST 指数。

语法

```
idx = enso(sst,t)
idx = enso(sst,t,lat,lon)
idx = enso(sst,t,lat,lon,'region',NinoRegion)
idx = enso(sst,t,mask)
idx = enso(...,'smoothing',months)
```

说明

idx = enso(sst,t) 根据海面温度 sst 及其相应时间 t 的时间序列计算厄尔尼诺南方涛动指数。 sst 可以是已在感兴趣区域上平均的海面温度向量，或者 sst 可以是一三维矩阵，其第三维对应于时间 t。 如果 sst 是三维矩阵，则通过对每个时间步长 sst 中的所有网格单元进行平均来自动生成时间序列。

idx = enso(sst,t,lat,lon) 计算三维 sst 时间序列的 Nino 3.4 指数和相应的网格坐标 lat,lon。使用此语法，Nino 3.4 区域内的网格单元将自动确定，并且 Nino 指数是根据该区域内 ssts 的面积平均时间序列计算的。

idx = enso(sst,t,lat,lon,'region',NinoRegion) 允许以下任何 Nino 区域，以字符串形式输入（在'单引号中'）

- '1+2'
- '3'
- '3.4' (指定 lat 和 lon 时的默认值)
- '4'
- 'ONI'

idx = enso(sst,t,mask) 使用对应于二维逻辑掩膜中的真实值的 sst 网格单元的未加权平均值计算 sst 索引。

idx = enso(...,'smoothing',months) 以月为单位定义移动平均窗口。根据 Trenberth 1997 ，默认值为 5。唯一的例外是 ONI 区域，它的平滑窗口为 3 个月。要关闭平均，设置'smoothing',false。

示例 1: 自动的 Nino 3.4

对于此示例，使用 CDT 附带的每月 pacific_sst.mat 数据集计算 Nino3.4 指数。首先加载数据，然后 enso 函数将想要确切知道 sst 数据集中每个网格单元对应的地理坐标，因此使用 meshgrid 将 lat,lon 数组放入我们称为 Lat,Lon 的二维网格中：

```
load pacific_sst.mat

% 从经纬度数组中获取二维网格:
[Lon,Lat] = meshgrid(lon,lat);
```

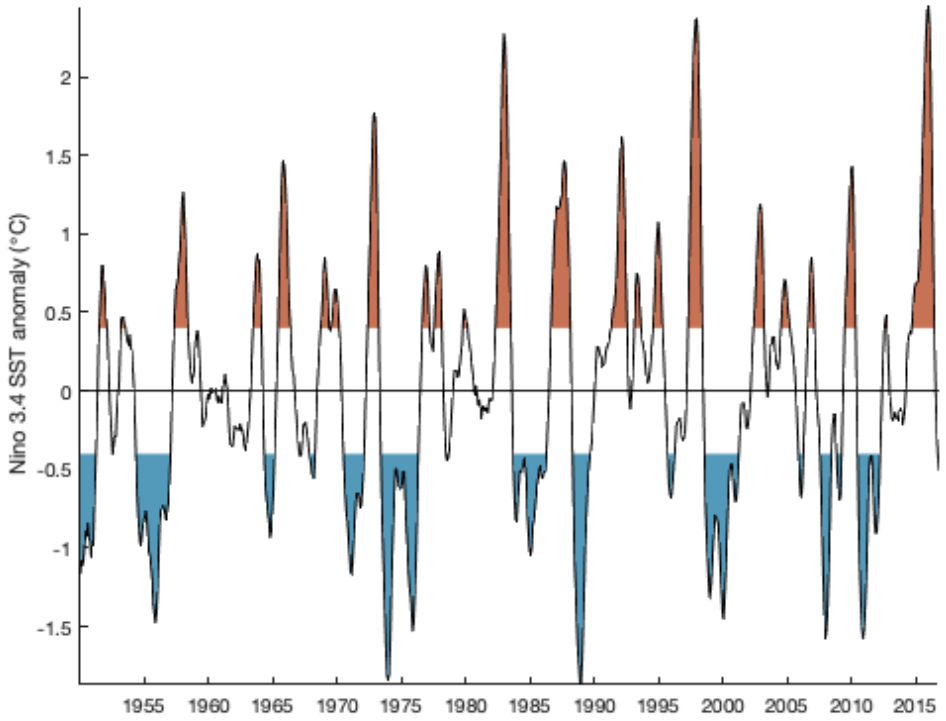
使用 enso 函数最简单的方法是输入三维 sst 数据，对应的时间 t，以及网格单元坐标，像这样：

```
idx = enso(sst,t,Lat,Lon);
```

绘制它真的很好

有了这个，我们现在可以绘制 Nino 3.4 异常。我们将使用阈值为 +/- 0.4 度的 anomaly 函数来分别识别厄尔尼诺和拉尼娜时期：

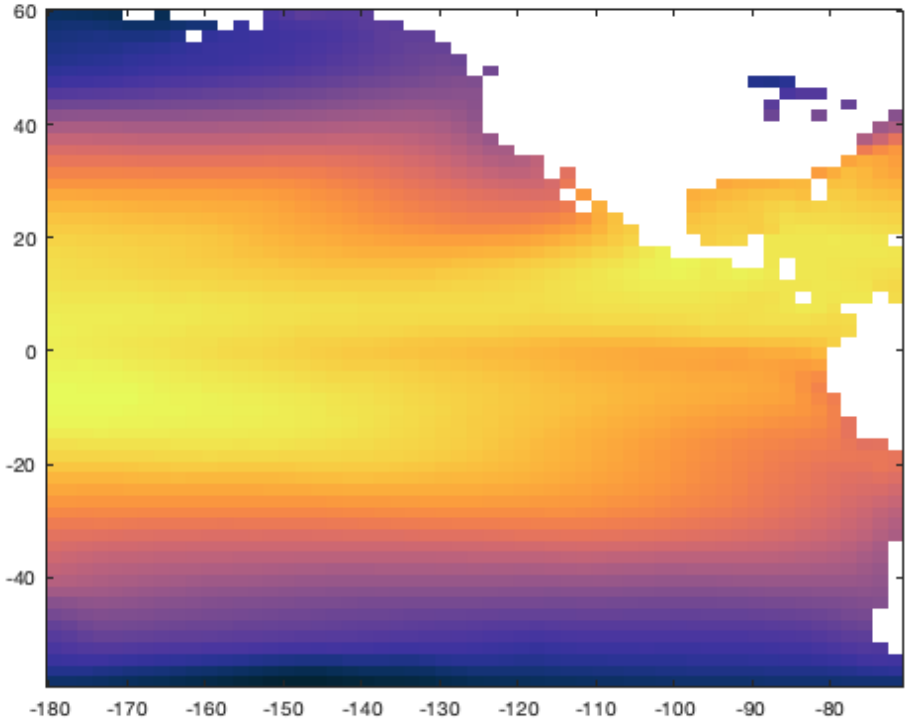
```
figure

anomaly(t,idx,'thresh',[-0.4 0.4]);
axis tight
hline(0,'k') % 在 0 处放置一条水平线
datetick('x','keeplimits')
ylabel 'Nino 3.4 SST anomaly (\circC)'
```



一些关联内容

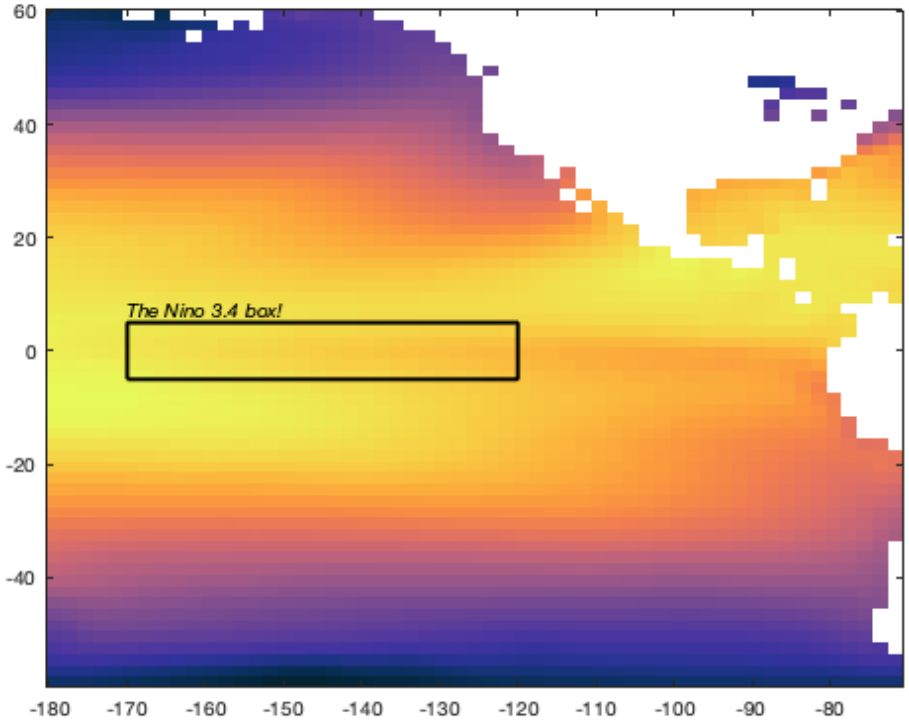
有关更多关联内容并了解 enso 函数正在做什么，这里是来自 pacific_sst 数据集的平均海面温度，使用 imagescn 绘制并使用 cmocean 热色图：

```
figure

imagescn(Lon,Lat,mean(sst,3))

cmocean thermal
```



默认情况下，enso 函数使用 Nino 3.4 框（5N-5S；170W-120W）的经纬度边界计算 Nino 指数。 这是地图上的 Nino 3.4 区域：

```
% 定义 Nino 3.4 框:
latv = [-5 -5 5 5 -5];
lonv = [-170 -120 -120 -170 -170];

% 绘制 Nino 3.4 框:
hold on
plot(lonv,latv,'k-','linewidth',2)
text(-170,5,'The Nino 3.4 box!','vert','bottom','fontangle','italic')
```



如果您想在默认 Nino 3.4 区域以外的区域探索 Nino 指数，请继续阅读..

示例 2: 手动方法

默认情况下，enso 函数计算 Nino3.4 框内的面积加权平均海面温度，然后使用 deseason 去除季节性周期。从淡化时间序列中去除平均值以获得 ENSO 异常，然后使用 scatstatl 作为移动平均滤波器进行平滑。enso 函数允许您覆盖 Nino 计算的大部分步骤或根据您的喜好设置它们。下面是一些如何自定义使用 enso 函数的示例。

定义 Nino 区域

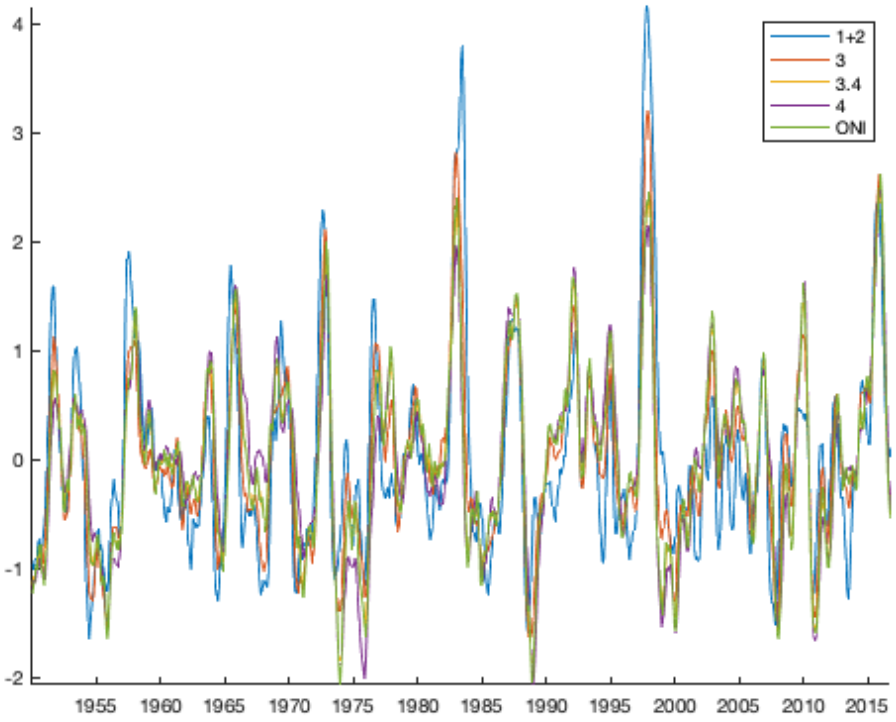
要使用默认的'3.4'以外的预定义 Nino 区域之一，只需像这样指定它：

```
idx3 = enso(sst,t,Lat,Lon,'region','3');
```

循环遍历所有预定义的 Nino 区域以进行比较：

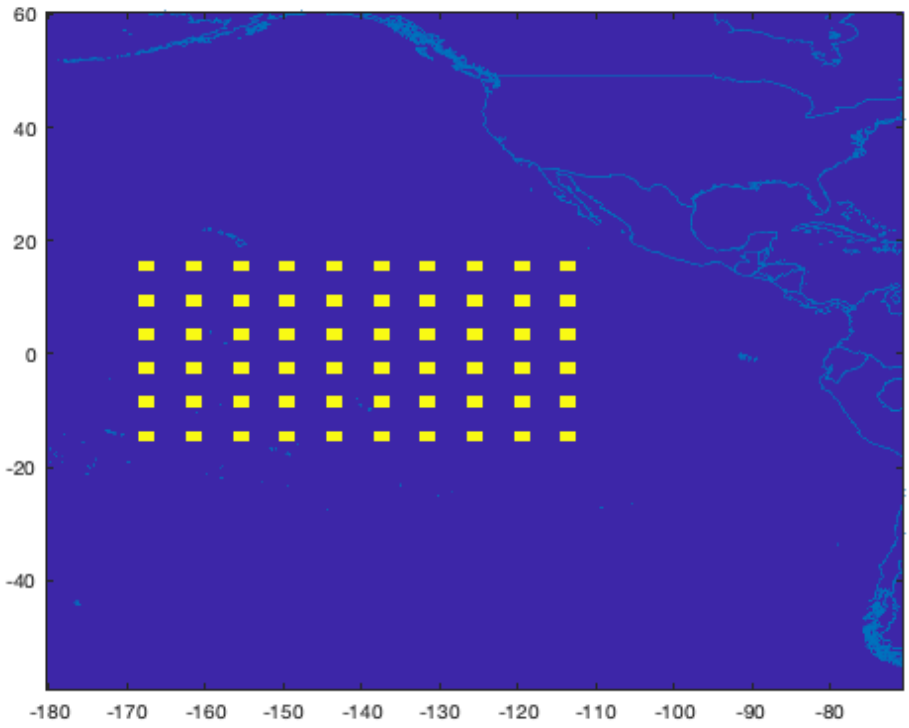
```
% 列出预定义区域:
regions = {'1+2','3','3.4','4','ONI'};

figure
hold on

% 遍历每个区域，计算其 Nino 指数，并绘制:
for k = 1:length(regions)
    tmp = enso(sst,t,Lat,Lon,'region',regions{k});
    plot(t,tmp)
```

```
end
axis tight
datetick('x','keeplimits')
box off
legend(regions)
```



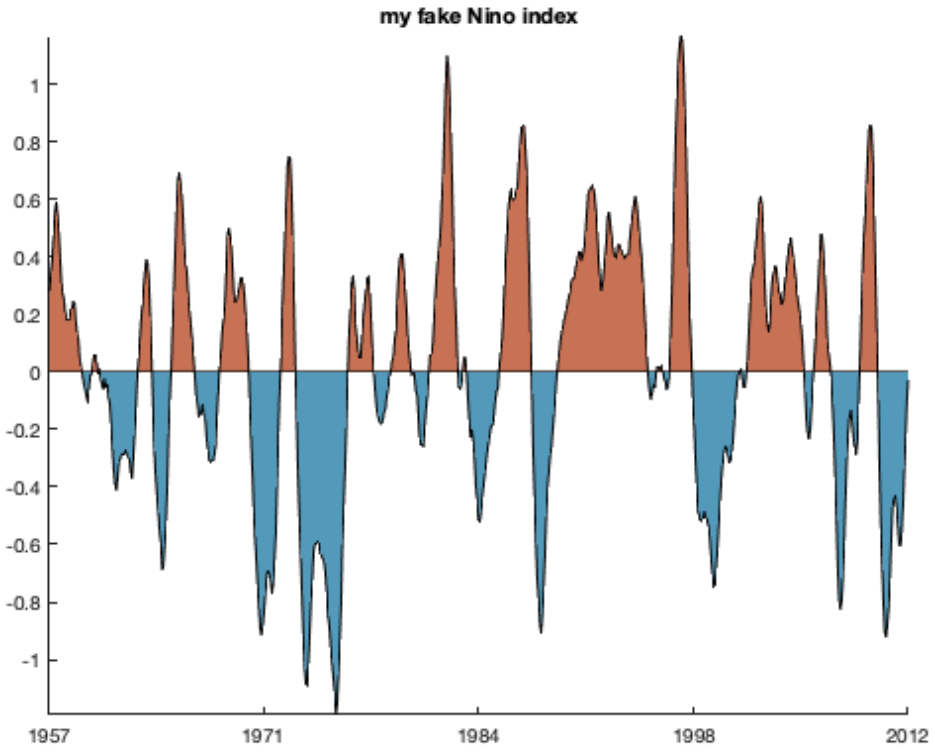
假设您想出一个更好的区域来表征厄尔尼诺和拉尼娜现象。它是从 20 S 到 20 N 以及从 170 W 到 110 W 的整个区域，但只有网格单元纬度和经度减去半度才能被 3 整除。使用 geomask 确定哪些网格单元位于该区域内并使用 mod 找出哪些网格单元（减去 0.5 度）可以被 3 整除。 这是掩膜的样子：

```
mask = geomask(Lat,Lon,[-20 20],[-170 -110]) ...
   & mod(Lat-0.5,3)==0 & mod(Lon-0.5,3)==0;

figure
imagescn(Lon,Lat,mask)
borders
```



如果您想获得上面掩膜中显示为黄色单元格的区域的*未加权* Nino 指数，请执行以下操作：

```
idx = enso(sst,t,mask);



figure

anomaly(t,idx)

axis tight
datetick('x','keepticks')
title 'my fake Nino index'
```



为了说明掩膜中的网格单元在地球表面面积上并不完全相等，请使用 cdtarea 来获取每个网格单元的面积，然后使用 local 来获取您的区域内海面温度的面积加权时间序列掩膜：

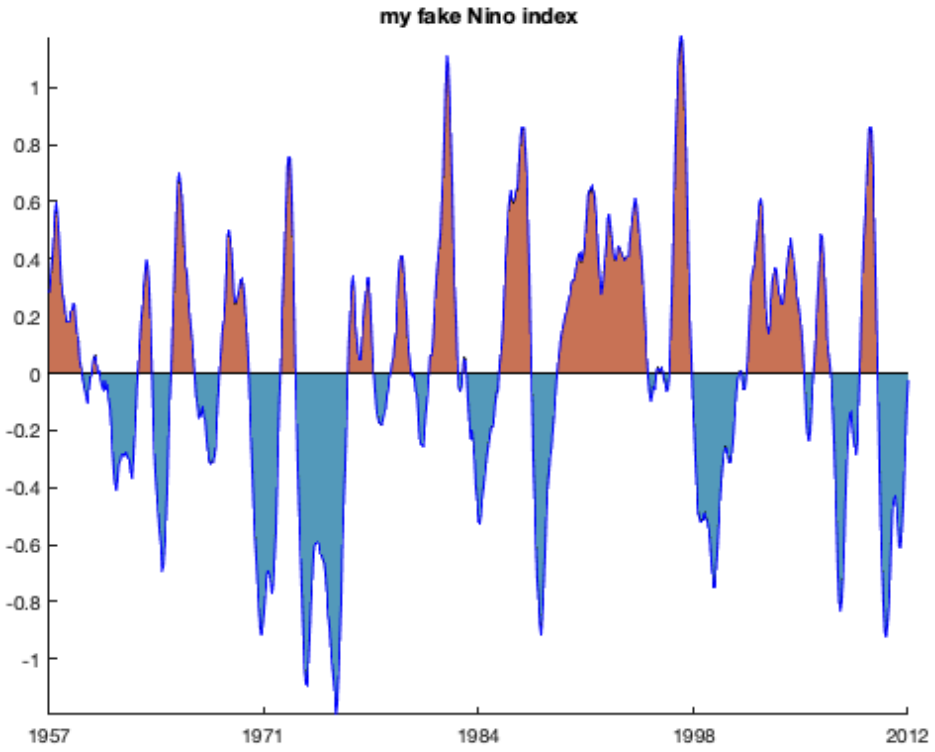
```
%获取每个网格单元的面积:
A = cdtarea(Lat,Lon);

%获取 Nino 3.4 框中的面积加权时间序列:
sst_myregion = local(sst,mask,'weight',A);
```

现在 sst_myregion 是掩膜中面积平均海面温度的 802x1 数组，并且可以直接将其输入 enso 函数以将其转换为索引：

```
idx_myregion = enso(sst_myregion,t);
```

在未加权的异常图上绘制面积加权指数，您会发现它们非常相似。 这是在我们上面创建的异常图顶部绘制为蓝线的加权版本：

```
hold on
plot(t,idx_myregion,'b')
```



在计算掩膜内的平均 sst 时，我们是否考虑了网格单元尺寸，在这里似乎并不重要。这是因为这些网格单元都在靠近赤道的相对较窄的纬度范围内，这些网格单元的大小都差不多。

示例 3: 重建 Trenberth 1997

在此示例中，我们将重建 Trenberth 1997 年的经典论文“厄尔尼诺现象的定义”(Trenberth, 1997)中的图 1，该论文描述了根据太平洋海面温度计算的 ENSO 指数。

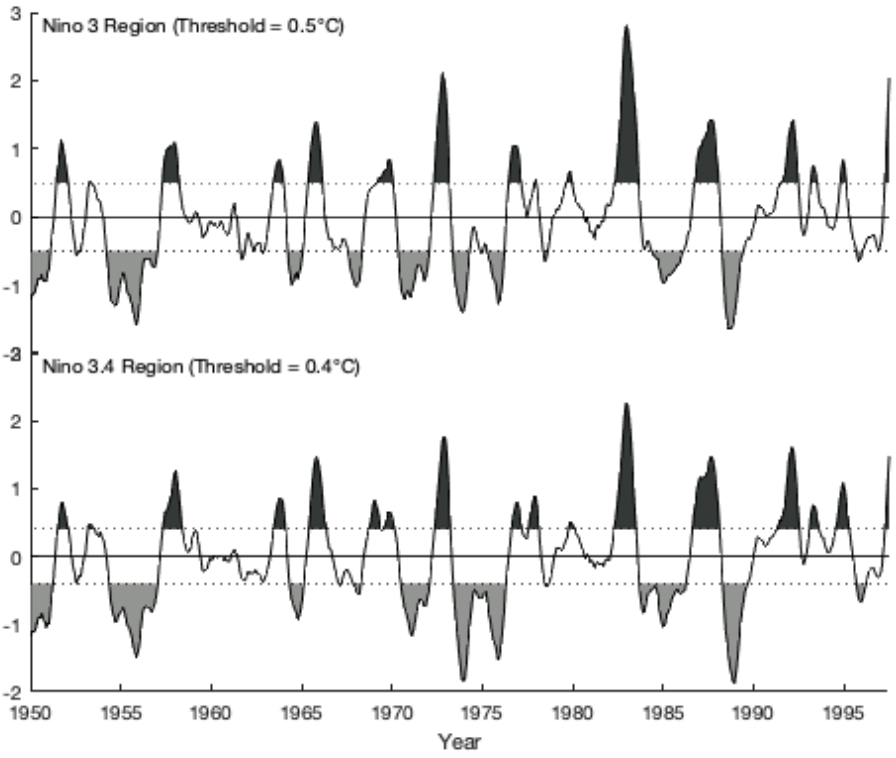
我们将使用与上述示例相同的 pacific_sst 数据集。 首先，计算 Nino 3 和 Nino 3.4 指数：

```
idx_3 = enso(sst,t,Lat,Lon,'region','3');
idx_34 = enso(sst,t,Lat,Lon,'region','3.4');
```

现在使用 subsubplot 重建 Trenberth 的图 1 的两个子图：

```
figure

subsubplot(2,1,1)

anomaly(t,idx_3,'top',rgb('charcoal'),...
   'bottom',rgb('gray'),'thresh',[-0.5 0.5]);

hline([-0.5 0.5],'k:')
hline(0,'k','linewidth',1)

xlim(datenum([1950 1997.5],1,1))
ylim([-2 3])
```

```
ntitle(' Nino 3 Region (Threshold = 0.5\circC) ','location','nw')

subsubplot(2,1,2)
anomaly(t,idx_34,'top',rgb('charcoal'),...
   'bottom',rgb('gray'),'thresh',[-0.4 0.4]);

hline([-0.4 0.4],'k:')
hline(0,'k','linewidth',1)

xlim(datenum([1950 1997.5],1,1))
ylim([-2 3])
ntitle(' Nino 3.4 Region (Threshold = 0.4\circC) ','location','nw')

datetick('x','keeplimits')
xlabel('Year')
```



参考文献

Trenberth, Kevin E. "The Definition of El Nino." Bulletin of the American Meteorological Society 78.12 (1997): 2771-2778. https://doi.org/10.1175/1520-0477(1997)078%3C2771:TDOENO%3E2.0.CO;2

作者简介

ts_enso 函数和支持文档是 Thirumalai <http://www.kaustubh.info> 和 Chad A. Greene 为 Climate Data Toolbox for Matlab 在 2019 年写的。

sam 文档

sam 计算南方环模指数的一个版本，也称为南极涛动。

语法

```
idx = sam(slp40,slp65,t)
```

说明

idx = sam(slp40,slp65,t) 根据两个纬度（40S 和 65S）的海平面气压的两个时间序列及其对应的时间 t 计算南方环模指数。

示例

在这里，我们将根据 Marshall 2003 年的经典论文《来自观察和再分析的南方环形模式趋势》（Marshall, 2003）重建其中的图 7，其中描绘了每月的 SAM 指数。

载入数据

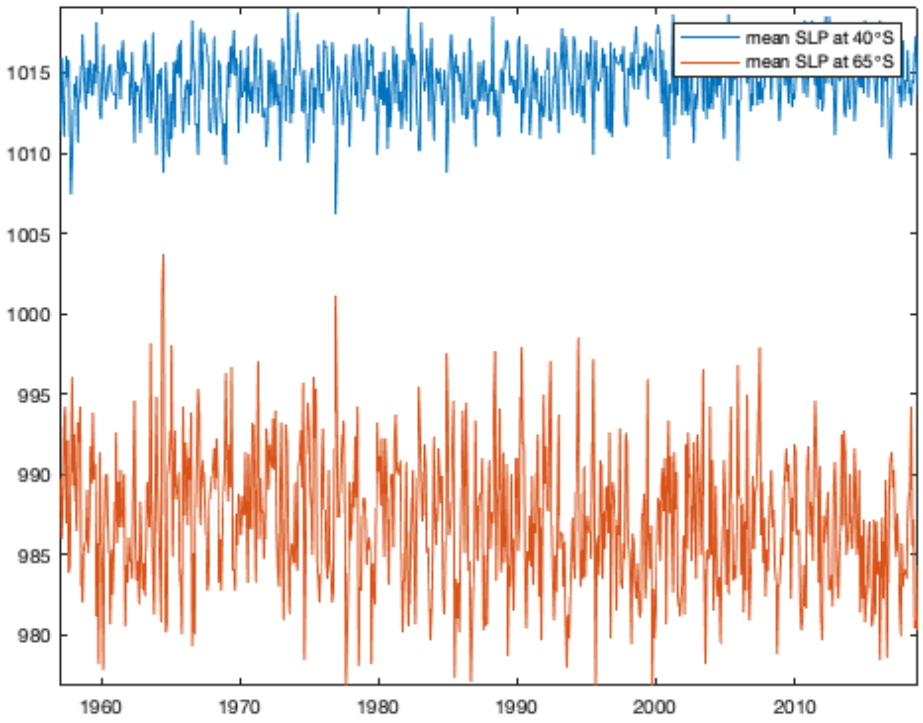
我们可以加载包含从 1957 年 1 月到 2018 年 12 月的观测值计算的 40S 和 65S 的纬向平均 SLP 的数据。

```
load sam_slp_data.mat
```

绘制气压数据

sam_slp_data.mat 文件包含从 1957 年到 2018 年的月纬向平均气压。（有关如何从网格数据中获取这些类型的纬向平均气压的注释出现在本节底部。）这是我们将使用计算 SAM 的气压数据：

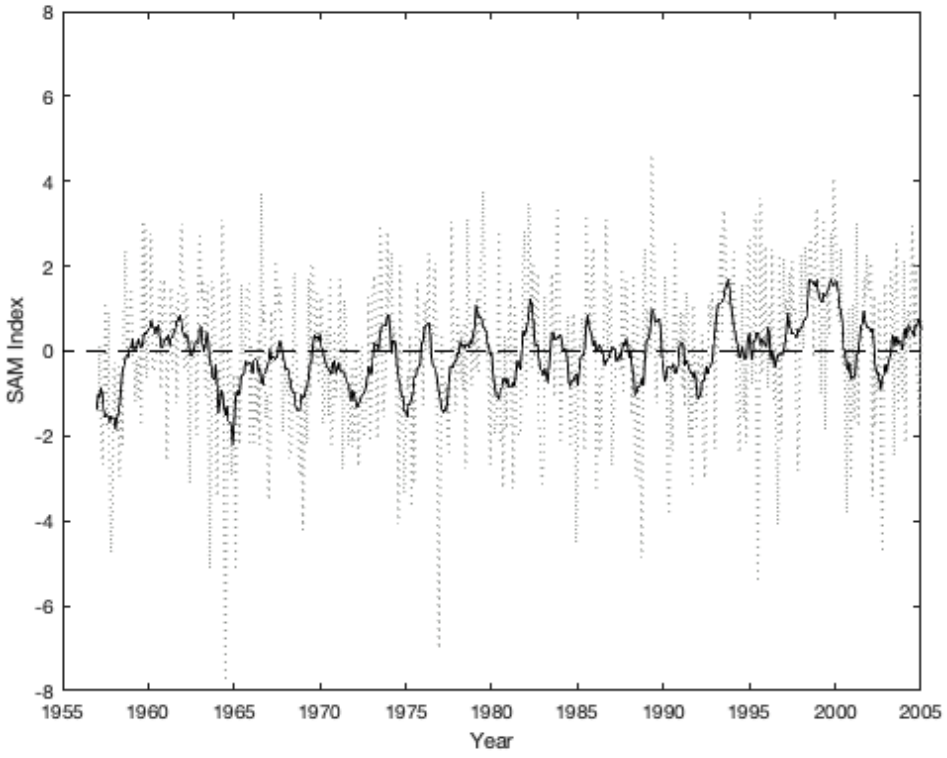
```
figure

plot(t,slp40);

hold on
plot(t,slp65)
axis tight
legend('mean SLP at 40\circS','mean SLP at 65\circS')
```



计算 SAM 指数

sam 函数相对于 1971-2000 年基线对每个时间序列进行归一化，并将两个气压异常时间序列进行差异化以产生 SAM 指数。

```
% 计算 SAM 指数:
sam_idx = sam(slp40,slp65,t);

figure
plot(t,sam_idx,':','color',rgb('gray'));
hold on
plot(t,movmean(sam_idx,12),'k','linewidth',1)
ylabel('SAM Index')
xlabel('Year')
xlim([datetime(1955,1,1) datetime(2005,1,1)]) % 设置日期范围
ylim([-8 8])                                  % 设置垂直范围
hline(0,'k--')                                % 绘制水平线
```



以上，用 Matlab 内置函数 movmean 计算 12 个月的移动平均值，用 hline 画一条水平虚线与 Marshall 的图 7 相匹配。

获取区域平均气压

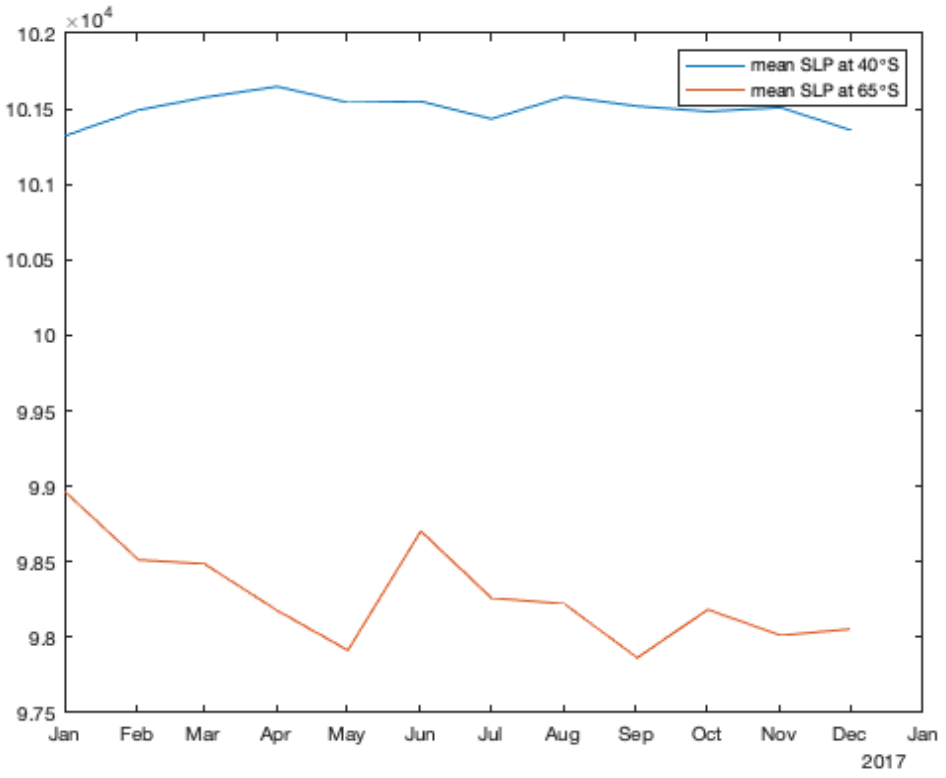
上面的示例首先加载 40S 和 65S 的区域平均气压。如果您正在处理网格数据，则需要计算 40S 和 65S 的区域均值。这样做。首先加载一些示例数据：（使用 ncdateread 读取日期时间）。

```
lat = double(ncread('ERA_Interim_2017.nc','latitude'));
lon = double(ncread('ERA_Interim_2017.nc','longitude'));
t = ncdateread('ERA_Interim_2017.nc','time');
sp = ncread('ERA_Interim_2017.nc','sp');
```

获得 40S 和 65S 的纬向均值的一种蛮力方法是在网格中的每个唯一经度处对 40S 和 65S 进行插值，然后对每个时间步的结果进行平均。 首先预分配输出，然后循环遍历每个时间步，插入 40S 和 65S

```
% 预分配纬向平均气压时间序列:
slp40 = NaN(size(t));
slp65 = NaN(size(t));

% 循环遍历每个时间步:
for k = 1:length(t)

   % 取所有经线 SLP 的平均值，插值到 40S:
   slp40(k) = mean(interp2(lat,lon,sp(:,:,k),-40*ones(size(lon)),lon));
```

```
    % 取所有经线 SLP 的平均值，插值到 65S:
    slp65(k) = mean(interp2(lat,lon,sp(:,:,k),-65*ones(size(lon)),lon));
end

figure
plot(t,slp40)
hold on
plot(t,slp65)

legend('mean SLP at 40\circS','mean SLP at 65\circS')
```



参考文献

Marshall, G. J., 2003: Trends in the Southern Annular Mode from observations and reanalyses. J. Clim., 16, 4134-4143. https://journals.ametsoc.org/doi/pdf/10.1175/1520-0442%282003%29016%3C4134%3ATITSAM%3E2.0.CO%3B2

作者简介

sam 函数和支持文档是 Kaustubh Thirumalai 为 Climate Data Toolbox for Matlab 在 2019 年写的。

nao 文档

nao 根据 Hurrell, 1995 提出的定义，根据海平面气压计算北大西洋涛动指数。就所有实际目的而言，NAO 指数等同于北极涛动和北环形模式。有关更多信息，请参阅 NCAR 的 David Thompson 对环形模式的介绍。

语法

```
idx = nao(slpA,slpB,t)
```

说明

idx = nao(slp40,slp65,t) 根据两个站点（A 和 B）的两个海平面气压时间序列及其对应的时间 t 计算北大西洋涛动指数。在这里，A 站（亚速尔群岛或里斯本或直布罗陀）通常位于 B 站（冰岛）以南。

示例

在这里，我们将根据 Jones 等人 1997 年的经典论文“使用直布罗陀和冰岛西南部的早期仪器气压观测扩展到北大西洋涛动”（Jones et al., 1997）重新创建图 4，该论文描绘了每月 NAO 指数。 该数据集建立在 Hurrell 1995 年的论文之上，站点数据可以在 https://crudata.uea.ac.uk/cru/data/nao/index.htm。找到。我们在这里使用的数据集从 1865 年 1 月到 2017 年 12 月。

载入数据

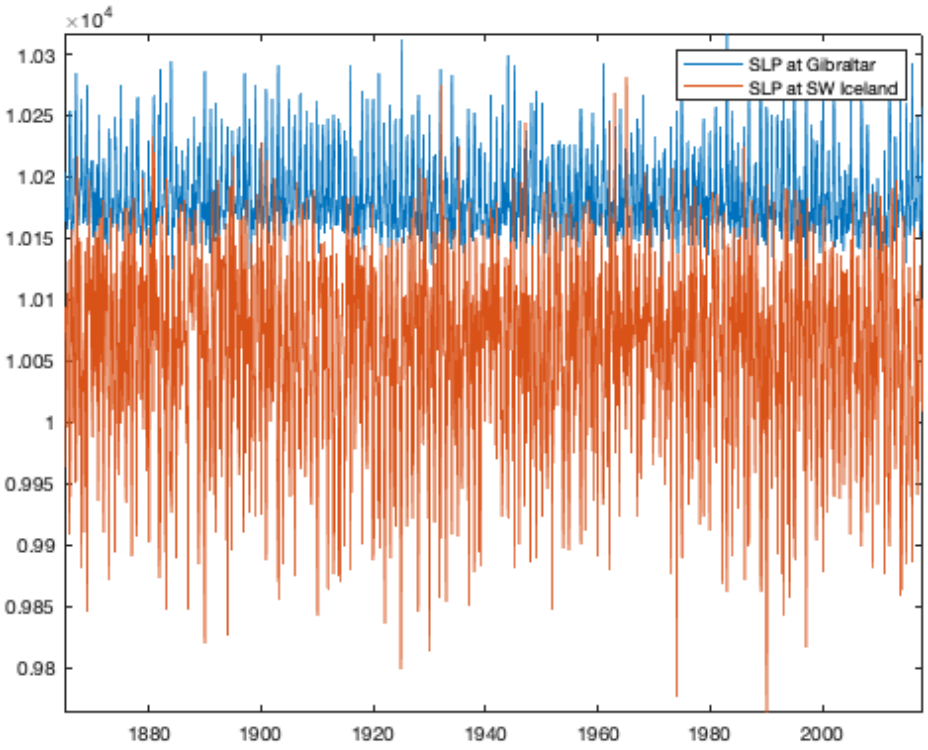
我们可以加载包含从 1957 年 1 月到 2018 年 12 月的观测值计算的 40S 和 65S 的纬向平均 SLP 的数据。

```
load nao_slp_data.mat
```

绘制气压数据

nao_slp_data.mat 文件包含从 1865 年到 2018 年直布罗陀 (A) 和冰岛 (B) 的月平均 SLP 数据。请参阅上文以查找可以从何处获得此数据集。 以下是我们将用于计算 NAO 的气压数据：

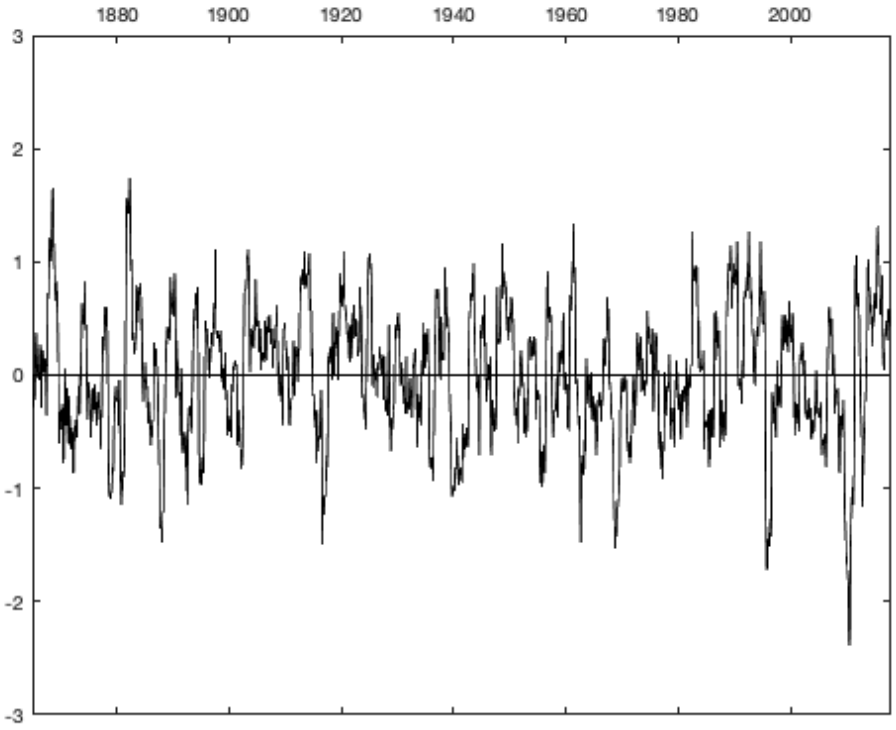
```
figure

plot(t,slpA);

hold on
plot(t,slpB)
axis tight
legend('SLP at Gibraltar','SLP at SW Iceland')
```



计算 NAO 指数

nao 函数相对于完整的基线数据集对每个时间序列进行归一化，并区分两个压强异常时间序列以产生 NAO 指数。我们将计算 NAO，然后使用 12 个月的移动平均线对其进行平滑处理（请参阅下面关于过滤的注释）：

```
% 计算 NAO 指数:
nao_idx = nao(slpA,slpB,t);

% 应用移动均值滤波:
nao_idx_f = movmean(nao_idx,12);

figure
plot(t,nao_idx_f,'k','linewidth',1)

ylim([-3 3])                                % 设置垂直范围
hline(0,'k-')                               % 绘制水平线
set(gca,'xaxislocation','top')              % 对上 Jones 1997
```



以上，用 Matlab 内置函数 movmean 计算 12 个月的移动平均值，用 hline 画一条水平虚线与 Marshall 的图 7 相匹配。

关于滤波的注释

Jones.et al,1997 中图 4 的标题表明在绘图之前使用了 12 个月的高斯滤波来平滑时间序列。高斯滤波器是一种移动平均滤波，它的加权使得移动窗口边缘附近的值对平均值的贡献不大。高斯滤波窗口的加权采用高斯曲线的形状，这与普通移动平均滤波的加权窗口实际上是一个矩形，其中所有值对平均值的贡献相等。不幸的是，Jones 等人“12 个月高斯滤波器”的说明有些模棱两可，因为不清楚高斯窗口的总宽度是什么意思，1-sigma 宽度，或 1-sigma 宽度的两倍，或者其他完全不同的东西。您如何准确定义应该经过无限距离逐渐减少到零的东西的宽度？有时人们会根据 2*pi*sigma 谈论高斯滤波器，因为这是权重因子等于 exp(-0.5)的距离，但目前还不清楚 Jones 等人是否就是故意的这样。

对于高斯窗口的确切宽度或其确切形状，我们实际上并没有明确的答案。然而，在这种特殊情况下，滤波器设计的细节可能并不重要。我们在上面使用了一个简单的移动平均线，但仍然得到了一个与 Jones 等人的图 4 几乎相同的图。这告诉我们，如果我们能复制 Jones 等人的方法就好了。确切地说，他们的方法和解释足以普遍复制他们的结果。

参考文献

Hurrell, J.W., 1995: Decadal Trends in the North Atlantic Oscillation: Regional Temperatures and Precipitation. *Science* Vol. 269, pp.676-679 doi:10.1126/science.269.5224.676

Jones, P. D. et al., 1997: Extension to the North Atlantic oscillation using early instrumental pressure observations from Gibraltar and south-west Iceland. Int. *J. Climatol.*, 17: 1433-1450. https://doi.org/10.1002/(SICI)1097-0088(19971115)17:13%3C1433::AID-JOC203%3E3.0.CO;2-P

作者简介

nao 函数和支持文档是 Kaustubh Thirumalai 为 Climate Data Toolbox for Matlab 在 2019 年写的。

amo 文档

amo 计算大西洋多年振荡指数的一个版本。

语法

```
idx = amo(sst,t)
idx = amo(sst,t,lat,lon)
```

说明

idx = amo(sst,t) 根据海面温度 sst 及其对应时间 t 的时间序列计算 AMO 指数。 sst 可以是已在感兴趣区域上平均的海面温度向量，或者 sst 可以是一个三维矩阵，其第三维对应于时间 t。 如果 sst 是三维矩阵，则通过对每个时间步长 sst 中的所有网格单元进行平均来自动生成时间序列。 通常，AMO 是根据北大西洋 0-70ºN 海温的面积加权平均值计算的。
idx = amo(sst,t,lat,lon) 计算三维 sst 时间序列的 AMO 指数和相应的网格坐标 lat,|lon|。 使用此语法，自动确定 AMO 区域内的网格单元，并根据该区域内 ssts 的面积平均时间序列计算 AMO 指数。.

示例

在此示例中，我们将使用 CDT 附带的每月 north_atlantic_sst.mat 数据集计算 AMO 指数。首先加载 CDT 附带的数据。它包含 1870-2017 年北大西洋地区(0-70N)的每月海面温度

```
load north_atlantic_sst.mat
```

以下是我们正在使用的变量:

```
whos % 展示变量名称和尺寸
```

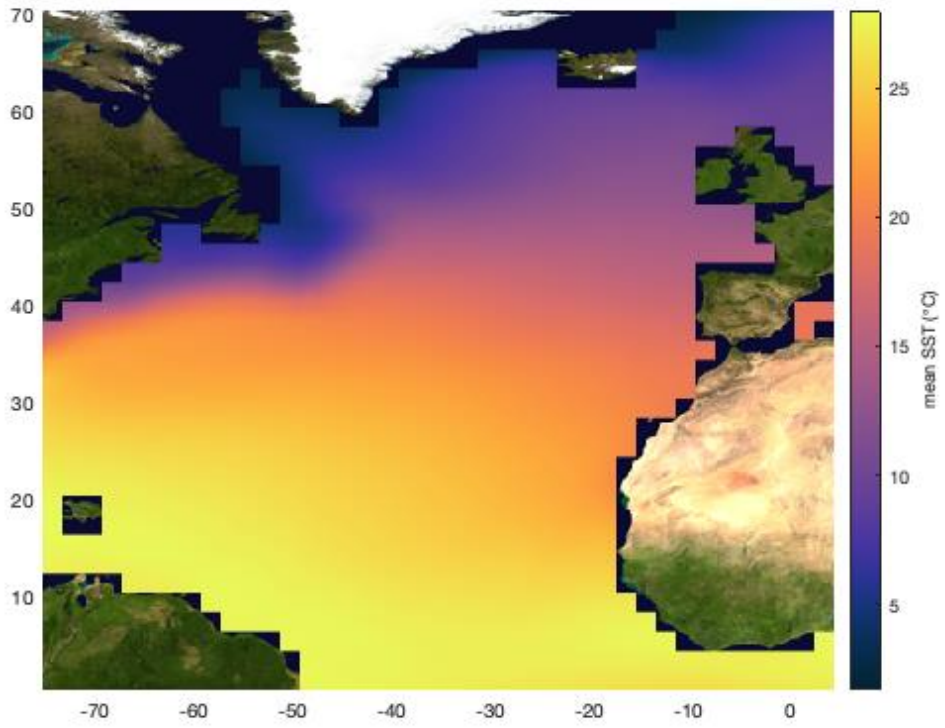
```
  Name          Size                Bytes  Class     Attributes

  lat           36x1                  144  single
  lon           41x1                  164  single
  sst_na        36x41x1776          10485504  single
  t           1776x1                  28417  datetime
```

上表告诉我们，我们有一个 lat 数组、一个 lon 数组和一个 t 数组，它们对应于北大西洋 SST 的 sst_na 数据立方体的维度。
我们需要准确地告诉 amo 函数对应于 sst_na 数据中的每个网格单元格的纬度和经度，因此将 lat 和 lon 数组转换为这样的网格：

```
[Lon,Lat] = meshgrid(lon,lat);
```

为了获得一点相关内容，让我们来看看我们将使用的数据。 制作时间平均平均 SST 的 pcolor 图（sst_na 数据立方体的第三向下维度）。通过 cmocean 函数将颜色图设置为 *thermal* ，并使用 earthimage 作为底图，以便更好地了解我们正在查看的位置：

```
pcolor(Lon,Lat,mean(sst_na,3))

shading interp
cmocean thermal % 设置颜色图
cb = colorbar;
ylabel(cb,'mean SST (\circC)')
hold on
earthimage('bottom') % 以地图卫星影像作为底图
```



上面的地图只显示了从 0N 到 70N 和 75W 到 5 E 的区域。 这是 amo 函数用来计算 AMO 指数的地理四边形，但如果您正在处理覆盖更大区域的 SST 数据集，请不要担心 -- amo 函数使用您提供的经度、纬度网格来仅根据北大西洋中的值进行计算。

计算 AMO 指数

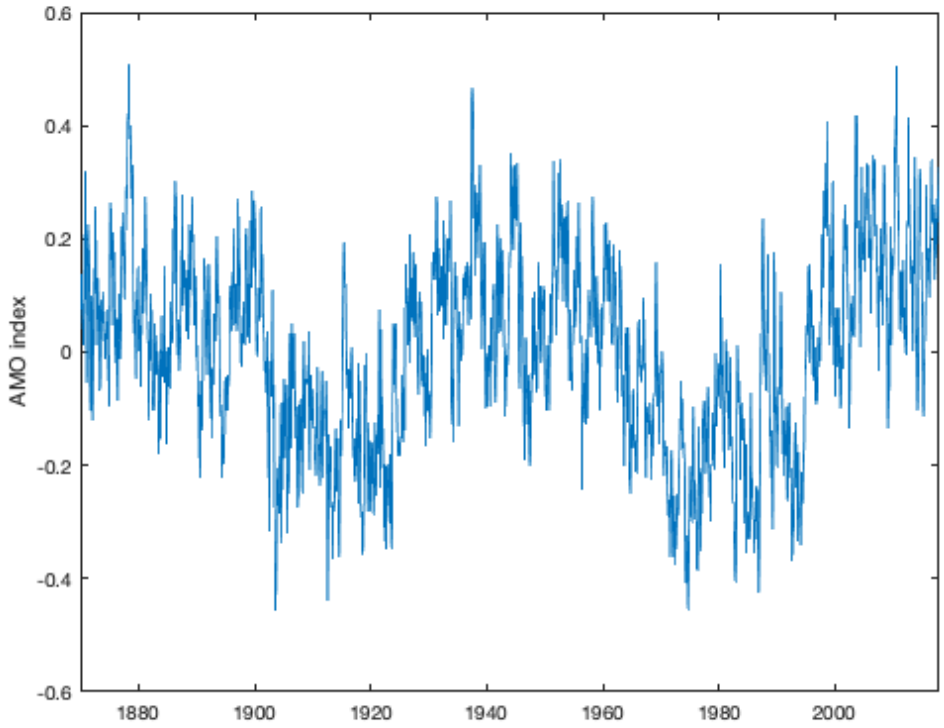
amo 函数通过输入 sst 数据立方体以及相应的时间和纬度、经度网格来工作：T

```
idx = amo(sst_na,t,Lat,Lon);
```

绘制原始的 AMO 指数:

```
figure
```

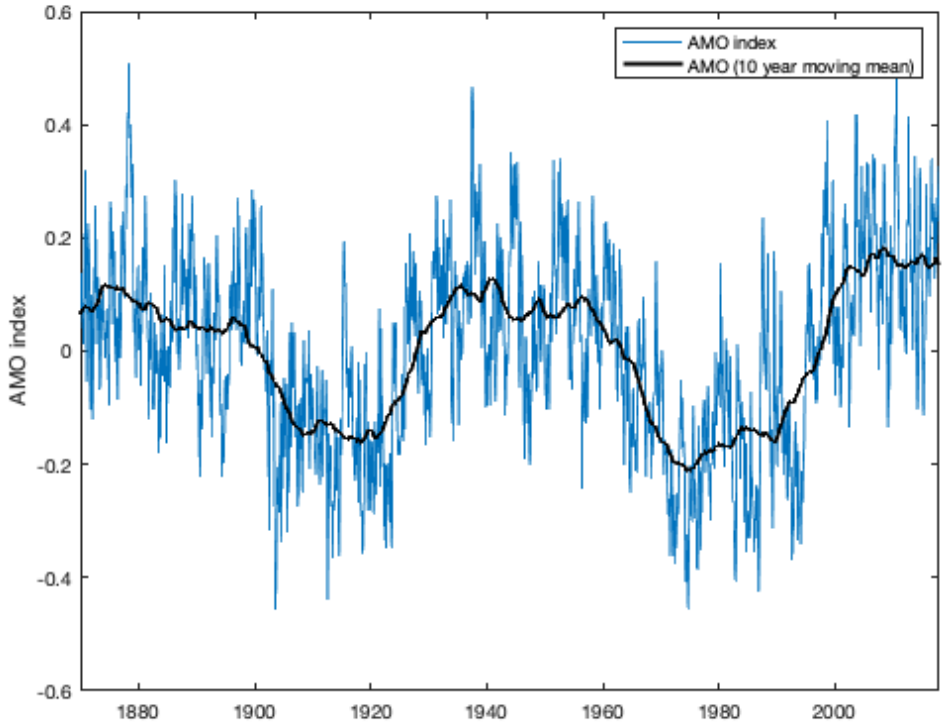
```
plot(t,idx)

ylabel 'AMO index'
```



上图显示了北大西洋海面温度的变化，它似乎有一种振荡行为，振荡周期似乎有几十年之久。 你可能会说这是某种大西洋多年代际振荡。

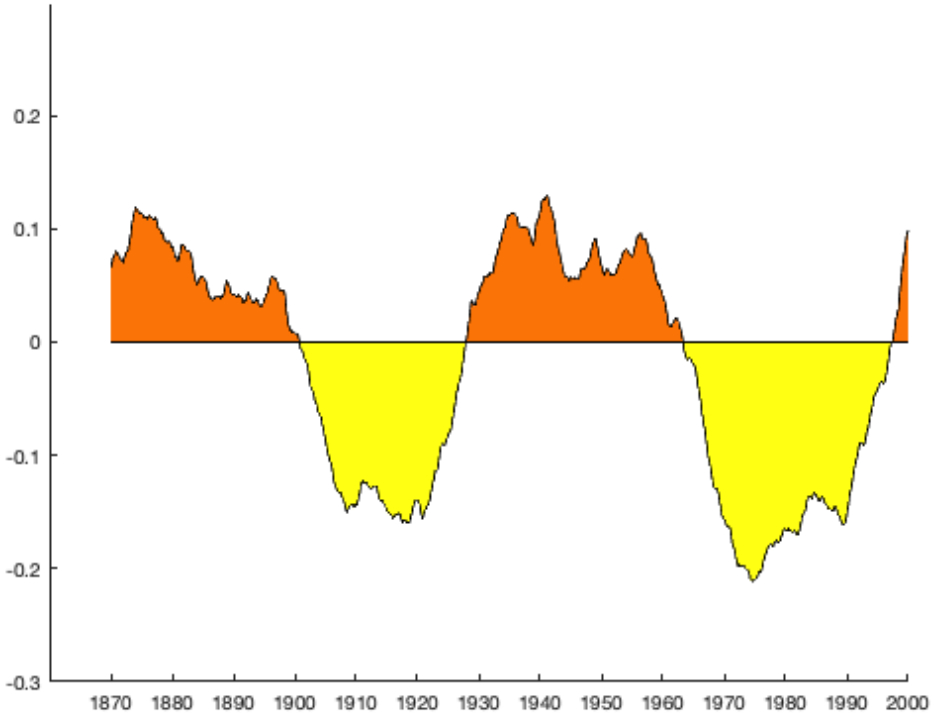
NOAA 建议采用 10 年（121 个月）移动平均来平滑时间序列：

```
% 具有 121 个月移动平均值的低通滤波器:
idx_f = movmean(idx,121);

hold on
plot(t,idx_f,'k-','linewidth',2)
legend('AMO index','AMO (10 year moving mean)')
```



为了表明我们在正确的轨道上，我们可以使用 anomaly 函数来模拟 Enfield et al., 2001 的图 1a：

```
figure

anomaly(datenum(t),idx_f,...
   'topcolor',rgb('orange'),'bottomcolor',rgb('yellow'))
axis tight
datetick('x')
xlim(datenum([1860 2000],1,1)) % 设置 x 轴日期范围
ylim([-0.3 0.3])               % 设置 y 轴范围
```



Enfield et al.的图 1b 显示了 AMO 指数与全球海面温度之间的相关性。 不幸的是，我们的示例数据集仅涵盖北大西洋，因此为了与全球其他地方进行比较，我们必须使用 pacific_sst 示例数据集：

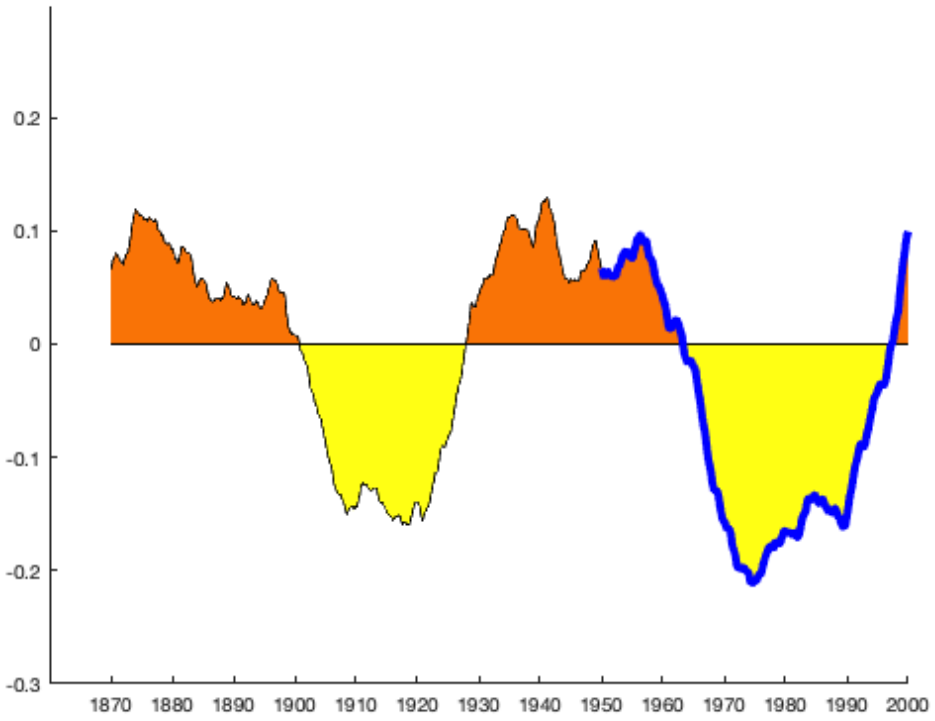
```
P = load('pacific_sst.mat');
```

恩 Enfield et al.将低通滤波的 AMO（本质上是去季节化的）与去季节的、低通滤波的全局 SST 相关联。因此，让我们对太平洋 SST 数据集进行去季节化和低通滤波：

```
% 对 SST 去季节和低通滤波:
sstd = deseason(P.sst,P.t);
sstdf = movmean(sstd,121,3);
```

在我们将低通滤波的 AMO 指数与去季节的、低通滤波的太平洋海温相关联之前，我们必须同步两个时间序列。 一种简单的方法是将 idx_f 数组插入太平洋 SST 时间，如下所示：

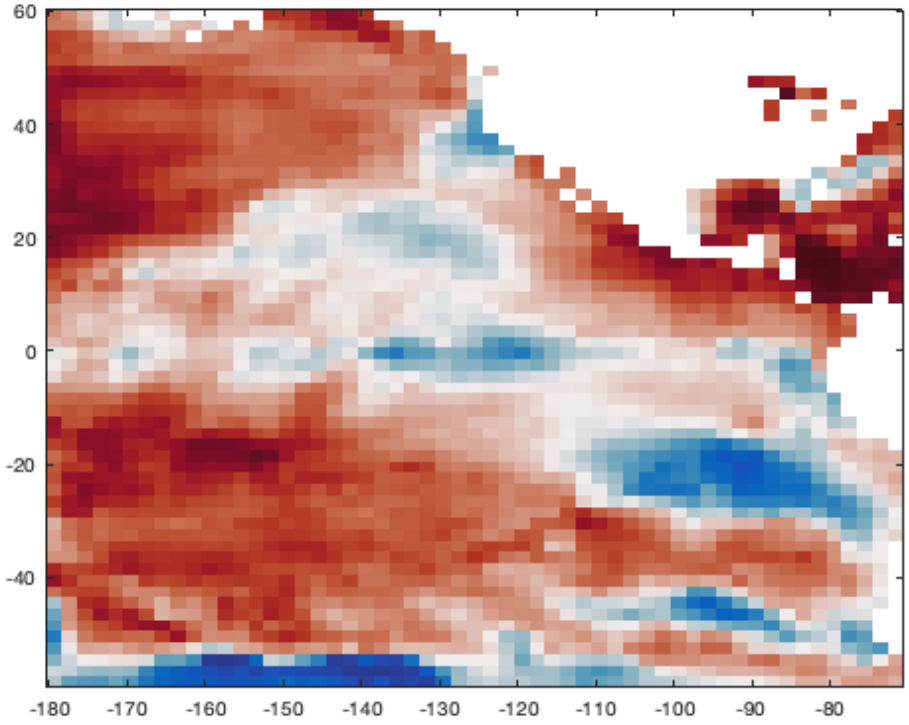
```
idx_fi = interp1(datenum(t),idx_f,P.t);



% 绘制内插值以确保一切都对齐:
hold on
plot(P.t,idx_fi,'b-','linewidth',4)
```



上面的蓝色粗线表示我们已将 idx_f 值正确地插入到时间 P.t，这对应于太平洋 SST 时间序列的时间。

现在我们能够将经过低通滤波的内插 AMO 指数 idx_fi 与太平洋 SST 相关联：

```
% 将 ssts 与 AMO 相关联:
C = corr3(sstdf,idx_fi);

figure
imagescn(P.lon,P.lat,C)
caxis([-1 1])
cmocean balance
```



上面的地图与 Enfield et al.的图 1b 并不完全匹配，这可能是因为我们只是将 AMO 与一个多年代际现象的 802 个月（66 年）进行了比较。 适当的相关性将查看振荡的几个完整周期。

参考文献

Enfield, D.B., A.M. Mestas-Nunez, and P.J. Trimble, 2001: The Atlantic Multidecadal Oscillation and its relationship to rainfall and river flows in the continental U.S., Geophys. Res. Lett., 28: 2077-2080. doi:0.1029/2000GL012745

作者简介

这个函数是亚利桑那大学的 Kaustubh Thirumalai 为 Climate Data Toolbox for Matlab 中 2019 年 3 月写的。 http://www.kaustubh.info

pet 文档

pet 依据 Hargreaves-Samani 给出了潜在的参考蒸发。
另请参见: spei.

语法

```
pevap = pet(Ra,tmax,tmin,tmean)
```

说明

pevap = pet(Ra,tmax,tmin,tmean) 计算外星辐射 Ra（例如来自 solar_radiation 函数）以及温度最大值、最小值和平均值的潜在蒸散量 (PET) 的 Hargreaves 方程。 温度 tmax、tmin 和 tmean 可以是向量，也可以是三维数组，其中前两个维度是空间，第三个维度对应于时间。 输入 tmax、tmin 和 tmean 表示给定时间段内的最大值、最小值和平均温度。

示例

此示例是 spei 函数文档中给出的示例的缩写形式。 您可能希望查看该页面以获取更多上下文和深入解释。
首先加载数据和转换单位。 降水单位为 kg^2/m^2/s，我们将这些单位更改为 mm/day。 温度以 K 为单位，我们将它们更改为摄氏度。

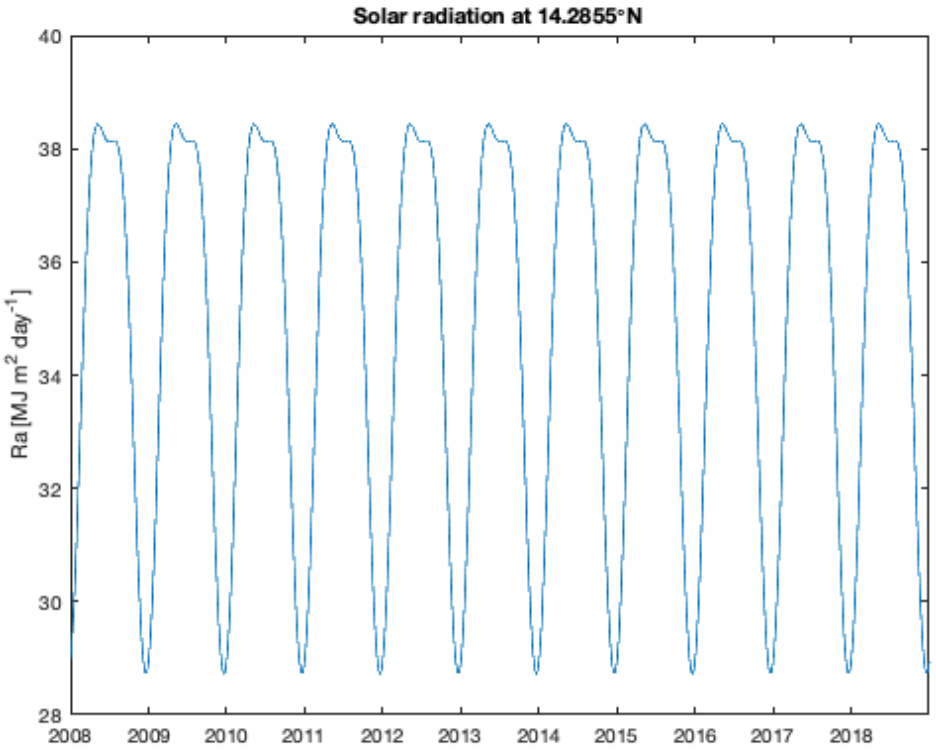
```
load ncep-ncar

P    = P*3600*24;
T    = T-273.15;
TMAX = TMAX-273.15;
TMIN = TMIN-273.15;
```

计算天文辐射

Hargreaves 和 Samani 的 PET 方程需要地外辐射(Ra)作为输入。Ra 是地球大气层顶部水平表面上的太阳辐射，根据与纬度相关的地球轨道参数计算得出。我们希望在每天的时间间隔内对太阳辐射的时间变化进行积分，这可以使用函数 solar_radiation 函数进行计算。使用 meshgrid 将 lat,lon 向量转换为网格，从而为每个网格单元提供不同的纬度值：：

```
[Lon,Lat] = meshgrid(lon,lat);



Ra  = solar_radiation(t,Lat);



% 绘制网格单元 5,5:
plot(t,squeeze(Ra(5,5,:)))
ylabel('Ra [MJ m^2 day^{-1}]')
```

```
title(['Solar radiation at ' num2str(lat(5)) '\circN'])
```

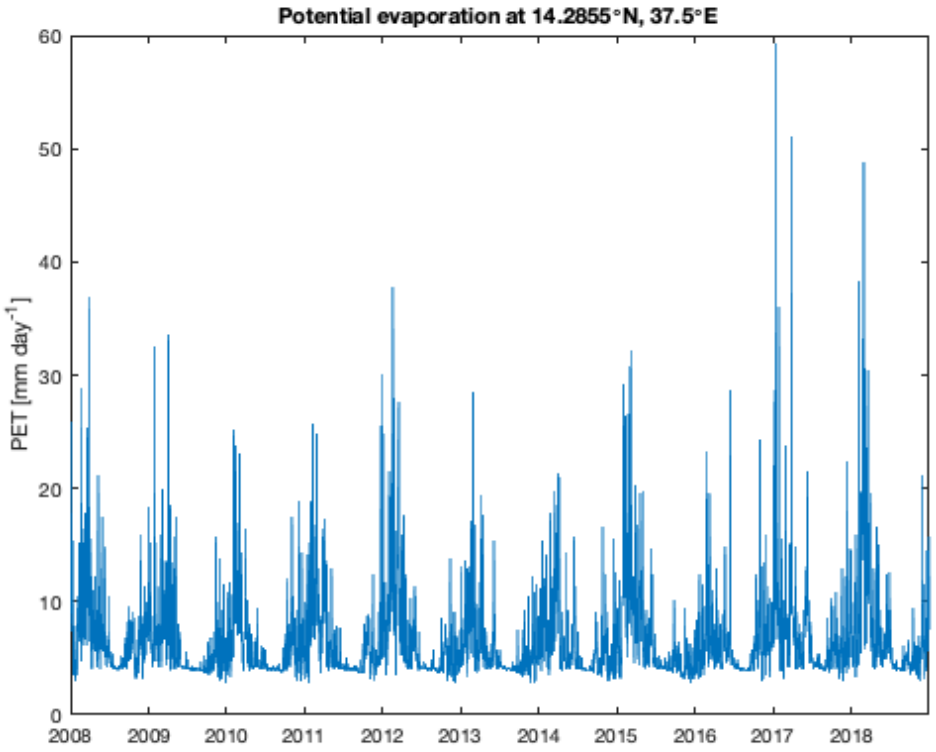


计算潜在蒸散量

SPEI 需要降水和潜在蒸散量(PET)。PET 是根据 Hargreaves 和 Samani(1985)的公式计算的，该公式根据温度和地外辐射估算参考作物的蒸散量。

有多种计算潜在蒸发量的方法。这里有一个汇编（见补充）。在这里，我们使用 Hargreaves-Samani(1985)的公式，这是一种基于温度的计算潜在蒸散量(PET)的方法，并在函数 pet 中实现。使用 Hargreaves-Samani 方程的主要优点在于它的简单性和对输入参数的低要求。

```
pevap = pet(Ra,TMAX,TMIN,T);



plot(t,squeeze(pevap(5,5,:)))

ylabel('PET [mm day^{-1}]')
title(['Potential evaporation at ' num2str(lat(5)) '\circN, ' num2str(lon(5)) '\circE' ])
%
```



参考文献

- Hargreaves, George H., and Zohrab A. Samani. "Reference crop evapotranspiration from temperature." Applied Engineering in Agriculture 1.2 (1985): 96-99. doi:10.13031/2013.26773.
- Samani, Zohrab. "Estimating solar radiation and evapotranspiration using minimum climatological data." Journal of Irrigation and Drainage Engineering 126.4 (2000): 265-267. 10.1061/(ASCE)0733-9437(2000)126:4(265).

作者简介

这个函数和支持文档是 José Delgado 和 Wolfgang Schwanghart（波茨坦大学）为 Climate Data Toolbox for Matlab 在 2019 年 2 月写的。

spei 文档

spei 函数根据基于特定位置气候的简化水平衡标准化计算标准化降水-蒸发蒸腾指数。
另请参见: pet.

语法

```
s = spei(t,prec,pevap)
[s,tint] = spei(t,prec,pevap,'integrationtime',months)
```

说明

s = spei(t,prec,pevap) 计算标准化降水-蒸发蒸腾指数 s，给定降水 prec 和对应于时间 t 的潜在蒸发 pevap。 prec 和 pevap 可以是一维向量或三维立方体，它们的前两个维度是空间维度，第三个维度对应于时间 t。 prec 和 pevap 的尺寸必须一致。 时间 t 可以是 datetime 或 datenum 格式。
[s,tint] = spei(t,prec,pevap,'integrationtime',months) 对 1、2、3、4、6 或 12 个月进行积分。 默认积分时间为 1 个月。 输出 tint 包含缩减时间序列的日期。

示例

此示例演示了如何计算和显示标准化降水蒸散指数 SPEI 的时空模式。 我们将使用从哥伦比亚大学 IRI/LDEO 气候数据库下载的 NCEP/NCAR 再分析数据（Climate Data Assimiliation System I Daily）。 文件 ncep-ncar.mat 包含许多平均温度 (T)、最高温度 (TMAX)、最低温度 (TMIN) 和降水 (P) 的日值的三维数组，以及时间、经度、纬度和降水量的向量。本页底部解释了如何从气候数据库中检索这些变量。

```
load ncep-ncar
```

我们的数据跨越以下时间范围

```
disp([min(t) max(t)])

disp(years(max(t)-min(t)))
```

```
   2008-01-01    2018-12-31

   10.9982
```

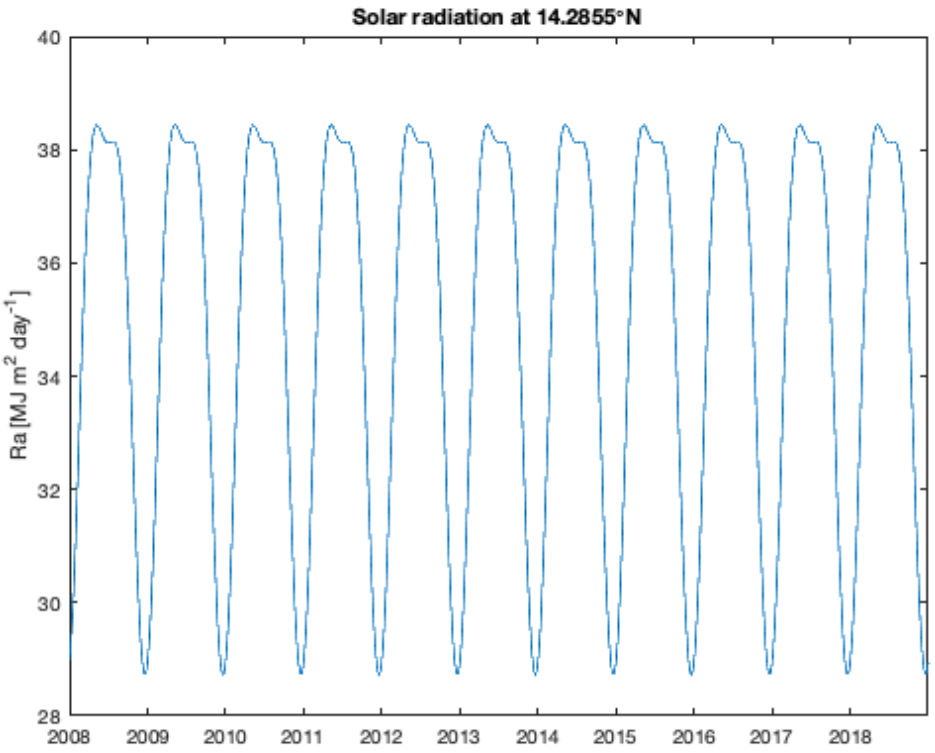
降水单位为 kg^2/m^2/s，我们将这些单位更改为 mm/day。 温度以 K 为单位，我们将它们更改为摄氏度。

```
P    = P*3600*24;

T    = T-273.15;

TMAX = TMAX-273.15;

TMIN = TMIN-273.15;
```

计算天文辐射

Hargreaves 和 Samani 的 PET 方程需要地外辐射 (Ra) 作为输入。 Ra 是地球大气层顶部水平表面上的太阳辐射，根据与纬度相关的地球轨道参数计算得出。 我们希望在每天的时间间隔内对太阳辐射的时间变化进行积分，这可以使用函数 `solar_radiation` 函数进行计算。 使用 `meshgrid` 将 `lat,lon` 向量转换为网格，从而为每个网格单元提供不同的纬度值:

```
[Lon,Lat] = meshgrid(lon,lat);



Ra  = solar_radiation(t,Lat);



% 绘制网格单元 5,5:
plot(t,squeeze(Ra(5,5,:)))
ylabel('Ra [MJ m^2 day^{-1}]')
title(['Solar radiation at ' num2str(lat(5)) '\circN'])
```

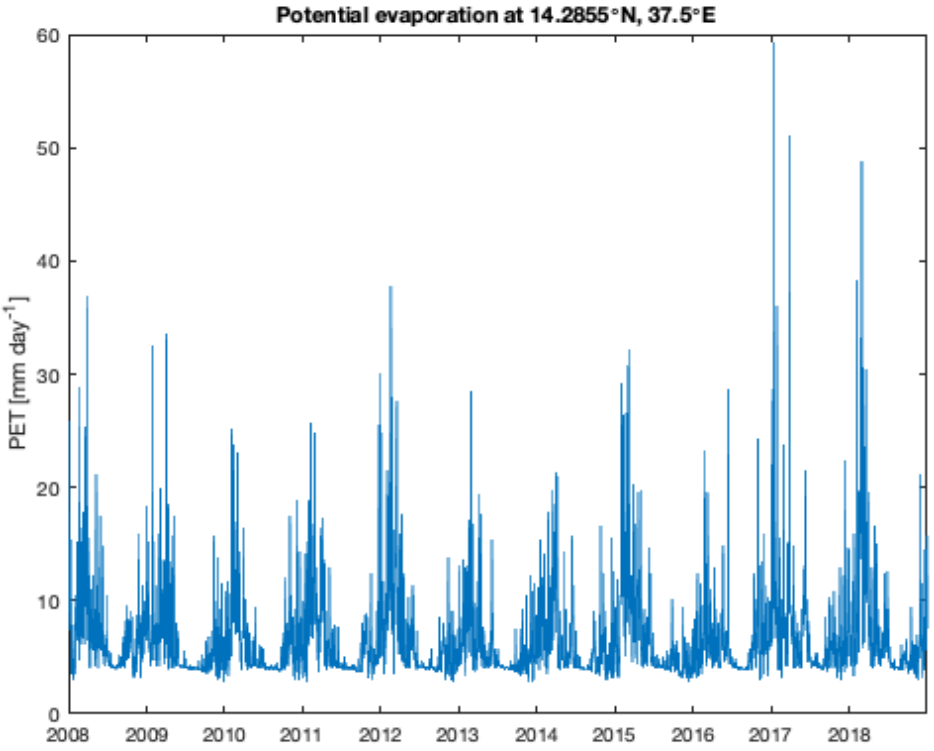


计算潜在蒸散量

SPEI 需要降水和潜在蒸散量 (PET)。 PET 是根据 Hargreaves 和 Samani (1985) 的公式计算的，该公式根据温度和地外辐射估算参考作物的蒸散量。

有多种计算潜在蒸发量的方法。这里有一个汇编（见补充）。在这里，我们使用 Hargreaves-Samani (1985) 的公式，这是一种基于温度的计算潜在蒸散量 (PET) 的方法，并在函数 pet 中实现。 使用 Hargreaves-Samani 方程的主要优点在于它的简单性和对输入参数的低要求。

```
pevap = pet(Ra,TMAX,TMIN,T);



plot(t,squeeze(pevap(5,5,:)))

ylabel('PET [mm day^{-1}]')
title(['Potential evaporation at ' num2str(lat(5)) '\circN, ' num2str(lon(5)) '\circE' ])
```

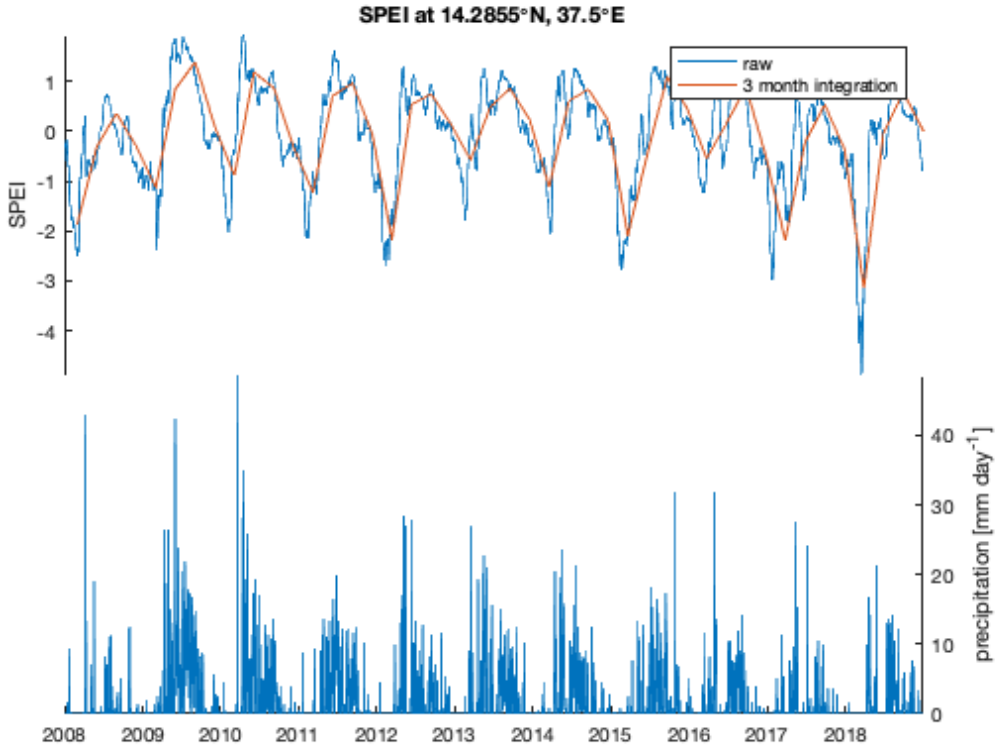


计算 SPEI

SPEI 是一种干旱指数，可供不同科学学科用于检测、监测和分析干旱。 SPEI 允许通过时间和空间比较干旱的严重程度，因为它可以在广泛的气候范围内进行计算。 函数 spei 根据降水和 PET 之间的差异计算指数，并允许设置不同大小的积分时间。

```
% 两种计算 SPEI 的方法:
s = spei(t,P,pevap);
[s3,t3] = spei(t,P,pevap,'integrationtime',3);

figure
subsubplot(2,1,1)
```

```
plot(t,squeeze(s(5,5,:)))
hold on
plot(t3,squeeze(s3(5,5,:)))
ylabel('SPEI')
axis tight
title(['SPEI at ' num2str(lat(5)) '\circN, ' num2str(lon(5)) '\circE' ])
legend('raw','3 month integration')

subsubplot(2,1,2)
plot(t,squeeze(P(5,5,:)))
ylabel('precipitation [mm day^{-1}]')
set(gca,'yaxislocation','right')
axis tight
```



检索 NCEP/NCAR 再分析数据

这是显示如何接收我们在此文件中使用的 ncep-ncar.mat 文件的脚本。、

```
whitespace = '%20';

bopen      = '%28';

bclose     = '%29';

datestart = '1 Jan 2008';
```

```
dateend   = '31 Dec 2018';

lonlims   = '30.0/50.0'; %'27.1875/45.9375';

latlims   = '20.0/5.0';

vars      = {'temp','maximum','minimum'};

for r = 1:numel(vars)

    url = ['https://iridl.ldeo.columbia.edu/SOURCES/.NOAA/.NCEP-NCAR/.CDAS-1/.DAILY/' ...

        '.Diagnostic/.above_ground/',...

        '.' vars{r} '/T/'...

        '(' datestart ')(' dateend ')RANGEEDGES/' ...

        'X/' lonlims '/RANGEEDGES/'...

        'Y/' latlims '/RANGEEDGES/dods'];

    url = replace(url,' ',whitespace);

    url = replace(url,'(',bopen);

    url = replace(url,')',bclose);

    if r == 1

       lon   = ncread(url,'X');

       lat   = ncread(url,'Y');

       t     = ncdateread(url,'T');

    end

    if r == 1

        T  = ncread(url,'temp');

    elseif r == 2

        TMAX  = ncread(url,'temp');
```

```
    else
        TMIN  = ncread(url,'temp');
    end
end
T = squeeze(T);
TMAX = squeeze(TMAX);
TMIN = squeeze(TMIN);
vars = 'prate'; % in kg/m2/s
url = ['https://iridl.ldeo.columbia.edu/SOURCES/.NOAA/.NCEP-NCAR/.CDAS-1/.DAILY/' ...
        '.Diagnostic/.surface/',...
        '.' vars '/T/'...
        '(' datestart ')(' dateend ')RANGEEDGES/' ...
        'X/' lonlims '/RANGEEDGES/'...
        'Y/' latlims '/RANGEEDGES/dods'];
url = replace(url,' ',whitespace);
url = replace(url,'(',bopen);
url = replace(url,')',bclose);
lonp   = ncread(url,'X');
latp   = ncread(url,'Y');
P  = ncread(url,'prate');
P  = squeeze(P);
% 三个维数组的第一维和第二维交换了。
% 这里我们使用函数 permute 来交换维度：
```

```
lon = lon';

T    = permute(T,[2 1 3]);

TMAX = permute(TMAX,[2 1 3]);

TMIN = permute(TMIN,[2 1 3]);

P    = permute(P,[2 1 3]);

save ncep-nacar.mat
```

参考文献

- Hargreaves, George H., and Zohrab A. Samani. "Reference crop evapotranspiration from temperature." Applied Engineering in Agriculture 1.2 (1985): 96-99. doi:10.13031/2013.26773.
- McMahon, T. A., et al. "Estimating actual, potential, reference crop and pan evaporation using standard meteorological data: a pragmatic synthesis." Hydrology and Earth System Sciences 17.4 (2013): 1331-1363. doi:10.5194/hess-17-1331-2013.

作者简介

这个函数和支持文档是 José Delgado 和 Wolfgang Schwanghart（波茨坦大学）为 Climate Data Toolbox for Matlab

海洋&大气

- bottom 查找三维矩阵的最低有限值，例如从三维网格数据集确定海底温度。
- windstress 根据风速估算海洋上的风应力。
- ekman 估计经典的 Ekman 传输和 10 m 风的上升流/下降流。
- coriolisf 返回任何给定纬度的科里奥利频率（也称为科里奥利参数或科里奥利系数）。
- rossby_radius 给出了正压海洋的罗斯比变形半径。
- mld 根据 Holte and Talley, 2009 计算混合层深度。
- binind2latlon 将正弦网格的分箱索引值转换为地理坐标。
- transect 根据在不同位置和/或时间收集的 CTD 剖面生成海洋学数据的彩色比例横断面图。
- transectc 根据在不同位置和/或时间收集的 CTD 剖面生成海洋学数据的等高线剖面图。

bottom 文档

bottom 返回三维矩阵中最低的有限值。此功能可用于在存在地形的情况下获取海底温度或大气表面温度。

语法

```
Zb = bottom(Z)

[Zb,ind] = bottom(Z)
```

说明

Zb =bottom(Z) 返回包含三维矩阵 Z 中最低有限值的二维矩阵。

[Zb,ind] =bottom(Z) 还返回二维矩阵 ind，其中包含沿 Z 的第三维对应于 Z 中最低有限值的索引。

示例:德雷克海峡

此示例使用来自南大洋数据库（如下所述）的一些网格化海洋温度数据。 首先加载示例数据：

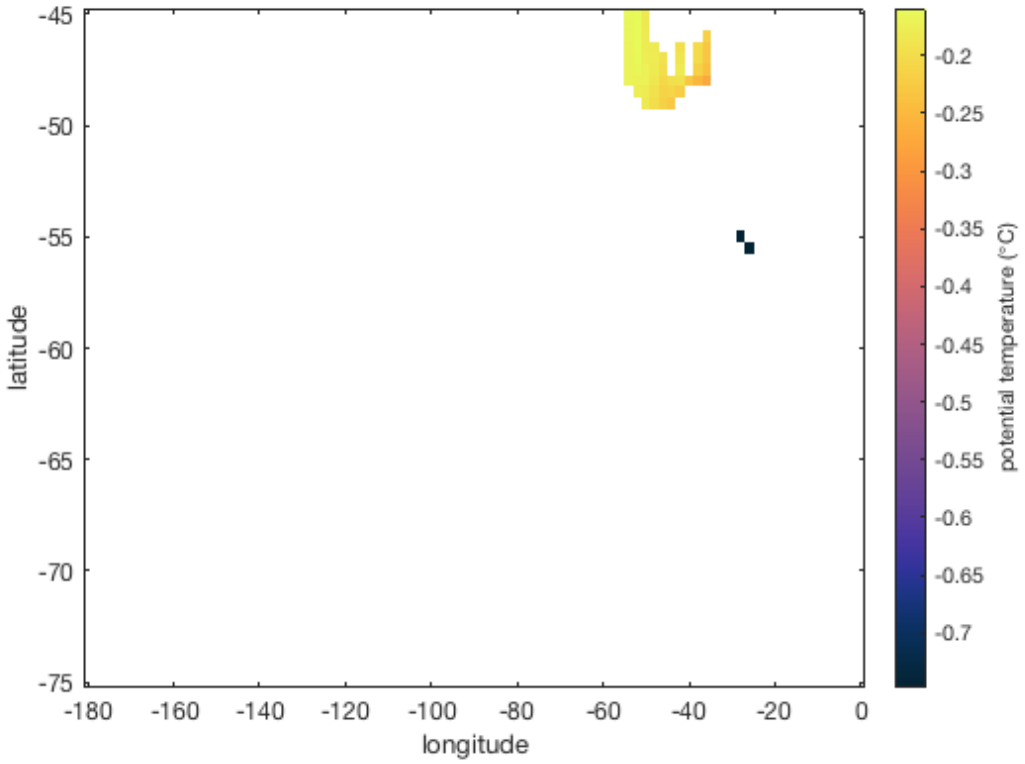
```
load sodb_example

whos % 展示变量的名称和尺寸
```

```
  Name          Size                Bytes  Class     Attributes

  d             44x1                  352  double

  lat           1x61                  488  double

  lon           1x91                  728  double

  ptm           61x91x44           1953952  double

  readme        1x79                  158  char
```

61x91x44 矩阵 ptm 包含纬度 lat、经度 lon 和深度 d 的位势温度。这是最深处的位势温度,d(44)

```
imagescn(lon,lat,ptm(:,:,44))



cmocean thermal
cb = colorbar;
ylabel(cb,'potential temperature (\circC)')
xlabel 'longitude'
```

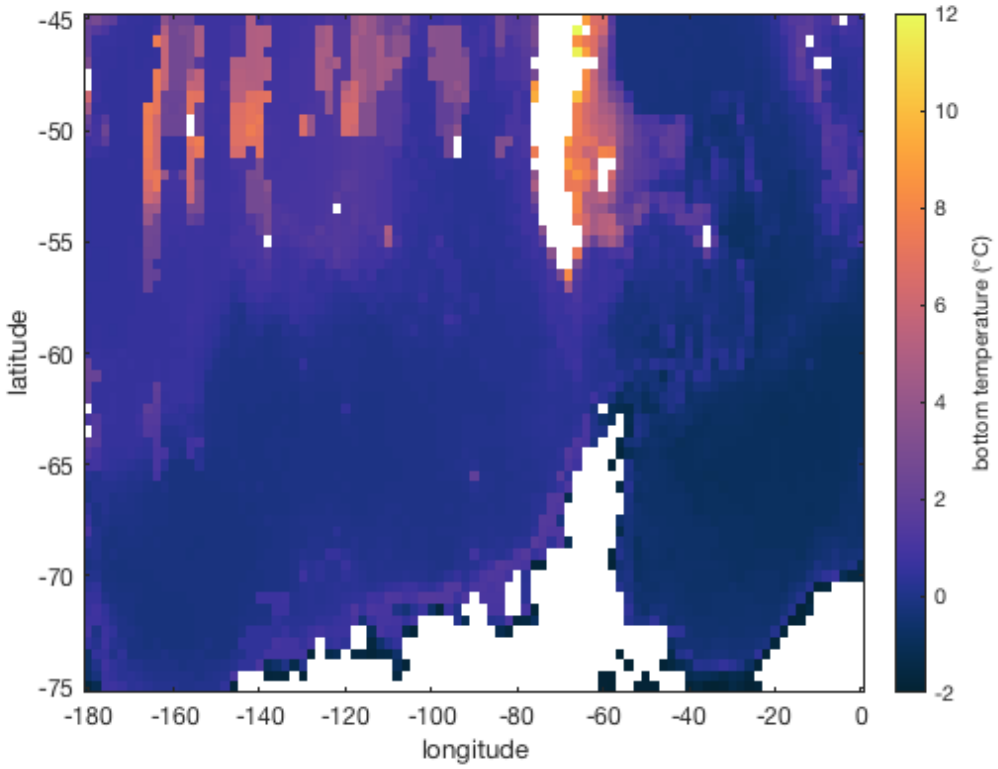
```
ylabel 'latitude'
```



这不是很多信息！ 这是因为在整个域的大部分区域，海底比 d(44) = 5500 m 深，所以我们只在 5500 m 层比水深测量更深的地方得到 NaN 值。

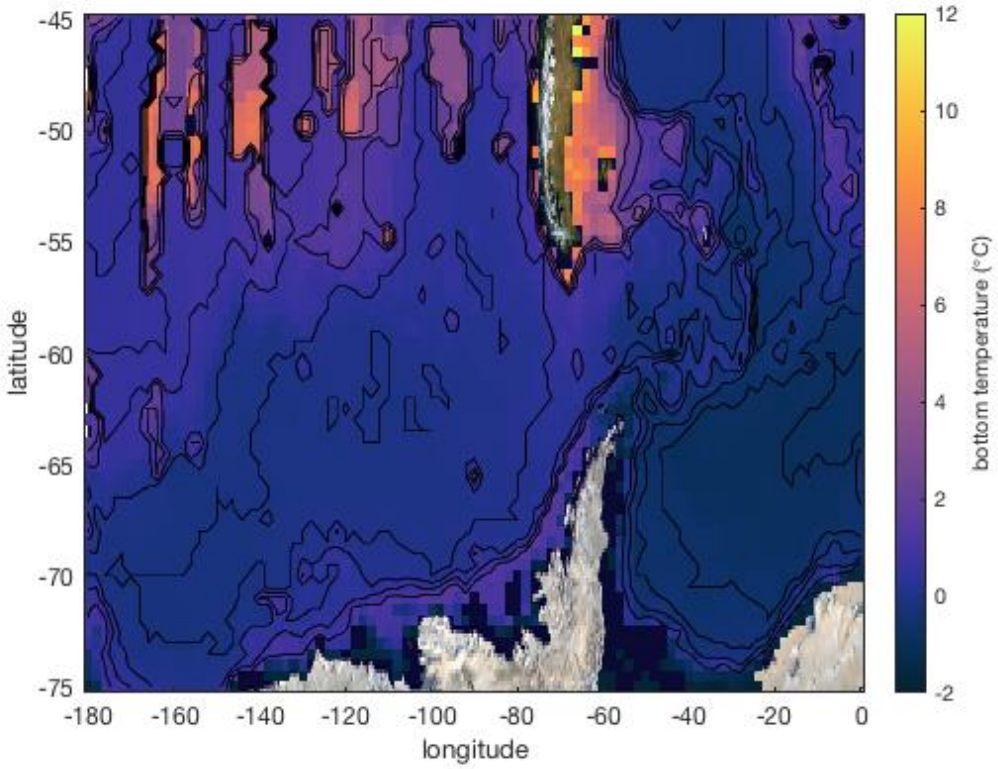
因此，相反，使用 bottom 函数绘制最低的有限值：

```
figure

imagescn(lon,lat,bottom(ptm))



cmocean thermal
caxis([-2 12])
cb = colorbar;
ylabel(cb,'bottom temperature (\circC)')
xlabel 'longitude'
ylabel 'latitude'
```



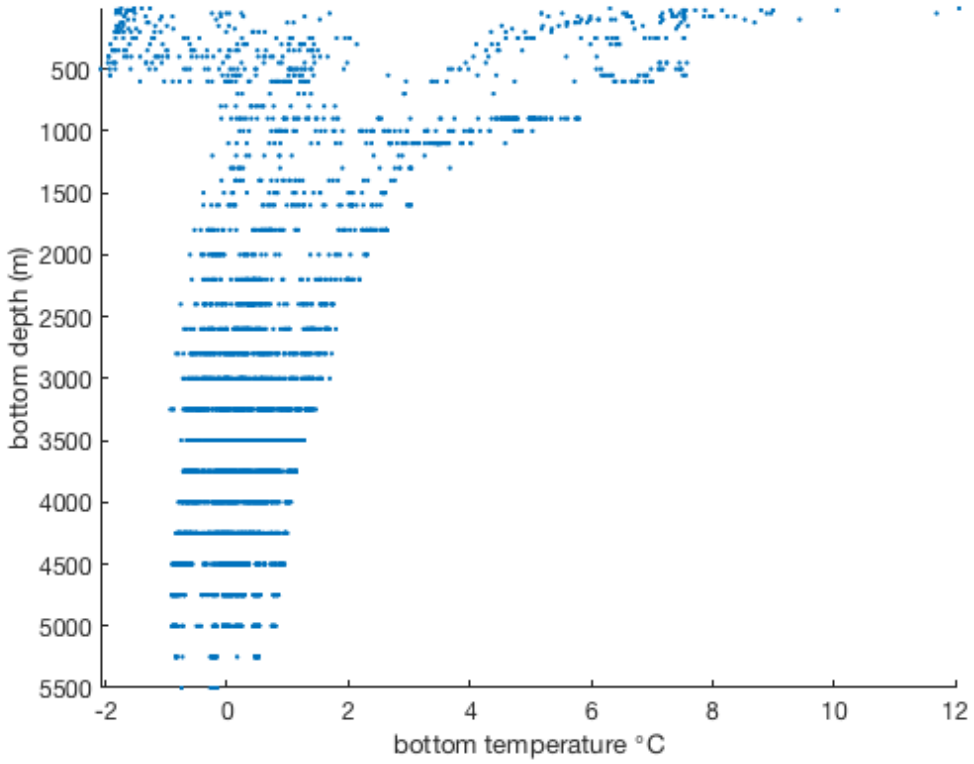
由于我们有一个深度数组 d 和对应于每个网格单元的最深有限值的索引 ind，我们可以简单地从 d(ind) 获得底部深度的二维网格。 将它们作为等值线添加到地图中：

```
% 获取底部温度和相应的索引:
[T_bot,ind] = bottom(ptm);

% 制作二维底部深度矩阵:
D = d(ind);

% 叠加测深等值线:
hold on
contour(lon,lat,D,0:1000:5500,'k')

% 在一个看起来很傻的地球图像下面:
h = earthimage;
uistack(h,'bottom');
```



稍微有趣的是，底部温度似乎与底部深度有某种关系。 像这样比较配置文件中的两者：

```
figure

plot(T_bot(:),D(:),'.')
axis tight ij % 翻转 y 轴方向
box off
xlabel 'bottom temperature \circC'
ylabel 'bottom depth (m)'
```



sodb_example 数据的获取方式

sodb_example.mat 数据来自南大洋数据库，使用 Matlab 南极制图工具(Greene et al., 2017)通过以下代码保存：

```
lon = -180:2:0;

lat = -75:0.5:-45;

[Lon,Lat] = meshgrid(lon,lat);

[ptm,d] = sodb_interp('ptm',Lat,Lon);

readme = 'potential temperature data from the Southern Ocean Database (Orsi and Whitorth)';

save('sodb_example.mat','lat','lon','ptm','d','readme');
```

作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

windstress 文档

windstress 根据风速估计海洋上的风应力。

语法

```
Tau = windstress(u10)
[Taux,Tauy] = windstress(u10,v10)
[...] = windstress(...,'Cd',Cd)
[...] = windstress(...,'ci',seaIce)
[...] = windstress(...,'rho',airDensity)
```

说明

`Tau = windstress(u10)` 估算了海面上方 10 m 处的风速（以米/秒为单位）施加在海洋上的风应力（以帕斯卡为单位）。输出 `Tau` 与输入 `u10` 的大小相同。

`[Taux,Tauy] = windstress(u10,v10)` 同时计算风应力的纬向和经向分量。

`[...] = windstress(...,'Cd',Cd)` 指定摩擦系数 `Cd`。默认 `Cd` 为 1.25e-3，这是全球平均值 (Kara et al., 2007)，但实际上 `Cd` 在空间和时间上可能会发生很大变化。 `Cd` 可以是与 `u10` 大小相同的标量或向量、二维矩阵或三维矩阵（如果包含 `v10`，则为 `v10`）。

`[...] = windstress(...,'ci',seaIce)` 指定用于估算 `Cd` 的海冰浓度，如 Lüpkes and Birnbaum, 2005 给出的。输入 `seaIce` 是海冰覆盖的一小部分，必须在范围内 0 到 1 包括在内。 `seaIce` 可以是与 `u10` 大小相同的标量或向量、二维矩阵或三维矩阵（如果包含 `v10`，则为 `v10`）。

`[...] = windstress(...,'rho',airDensity)` 指定空气密度，可以是标量或向量、二维矩阵或三维矩阵，其大小与 `u10` 相同（如果包含 `v10`，则为 `v10`）。

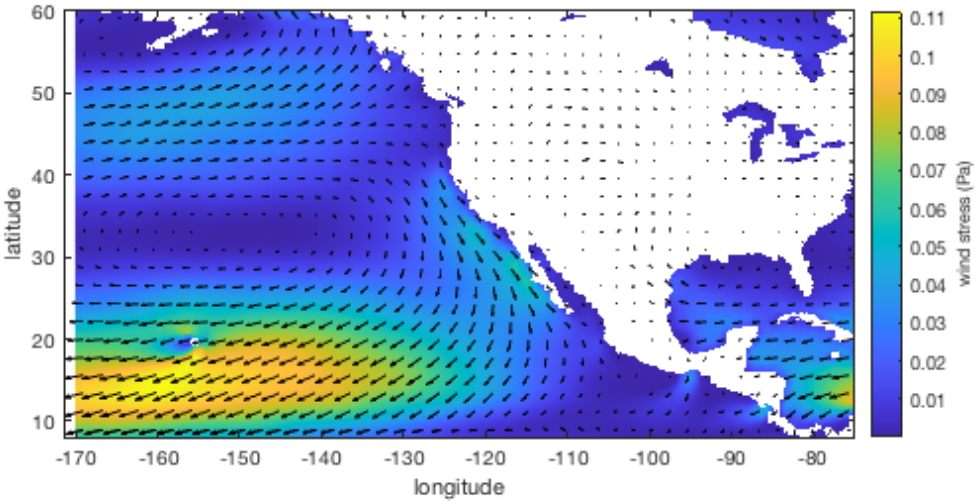
示例 1a: 根据再分析数据估算风应力

加载样本 `pacific_wind.mat`，然后绘制由风引起的海洋上的风应力。 下面我使用 `sst` 数据集来屏蔽陆地值，但如果数据集中还没有这种区分陆地和海洋的简单方法，您也可以轻松地使用 `island` 。

我们将使用 `imagescn` 绘制风应力和 `quiversc` 绘制风向量：

```
load pacific_wind

% 根据风速计算风应力的大小
Tau = windstress(hypot(u10,v10));

% 通过将值设定为 NaN 屏蔽陆地:
Tau(isnan(sst)) = NaN;

% 使用 imagesn, imagesc,或 pcolor 绘制:
figure
imagescn(lon,lat,Tau)
axis xy image
cb = colorbar;
ylabel(cb,' wind stress (Pa) ')
xlabel('longitude')
ylabel('latitude')
```

```
% 添加风速矢量:
hold on
quiversc(lon,lat,u10,v10,'k')
```



示例 1b: 注意分量

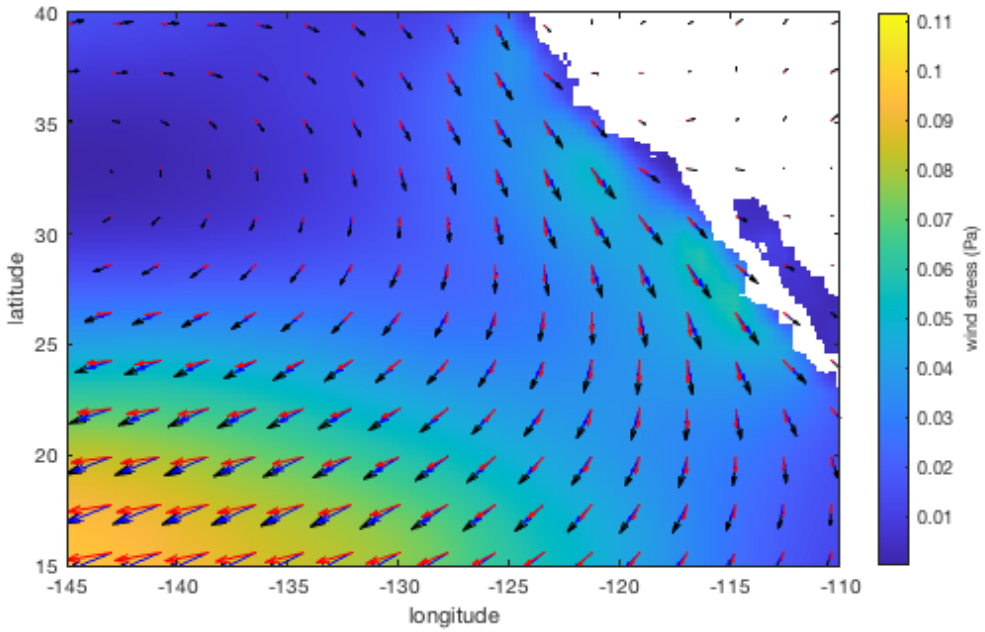
风应力有一个有趣的地方，它的大小与风速的平方成正比。如果根据风速的每个分量（u10 和 v10）分别计算风应力的每个分量（纬向和经向），则得到的风应力矢量的大小和方向可能不正确！

考虑一个速度矢量，其分量为 u10＝2 m/s，V10＝1 m/s。由此可知，速度矢量的角为 atand(1/2) = 26.6 度。风应力与风速的平方成正比，但是如果我们将风速的每个分量平方得到一个风应力矢量，得到的风应力矢量甚至不会指向正确的方向！在本例中，对风速的每个分量进行平方处理，将得到一个角度为 atand(1/4) = 14.0 度的风应力矢量。它的大小也不正确。

将正确计算的风应力矢量与错误计算的风应力矢量进行比较，方法是将正确矢量（蓝色）和错误矢量（红色）放置在示例 1a 中黑色风速矢量的顶部：

```
% 计算正确的风应力分量:
[Taux,Tauy] = windstress(u10,v10);

% 在黑色风速矢量上绘制正确的蓝色风应力矢量:
quiversc(lon,lat,Taux,Tauy,'b')
% Incorrectly calculate wind stress as components:
Taux_wrong = windstress(u10);
Tauy_wrong = windstress(v10);
```

```
% 绘制错误的红风应力矢量:
quiversc(lon,lat,Taux_wrong,Tauy_wrong,'r')

% 缩放:
axis([-145 -110 15 40])
```



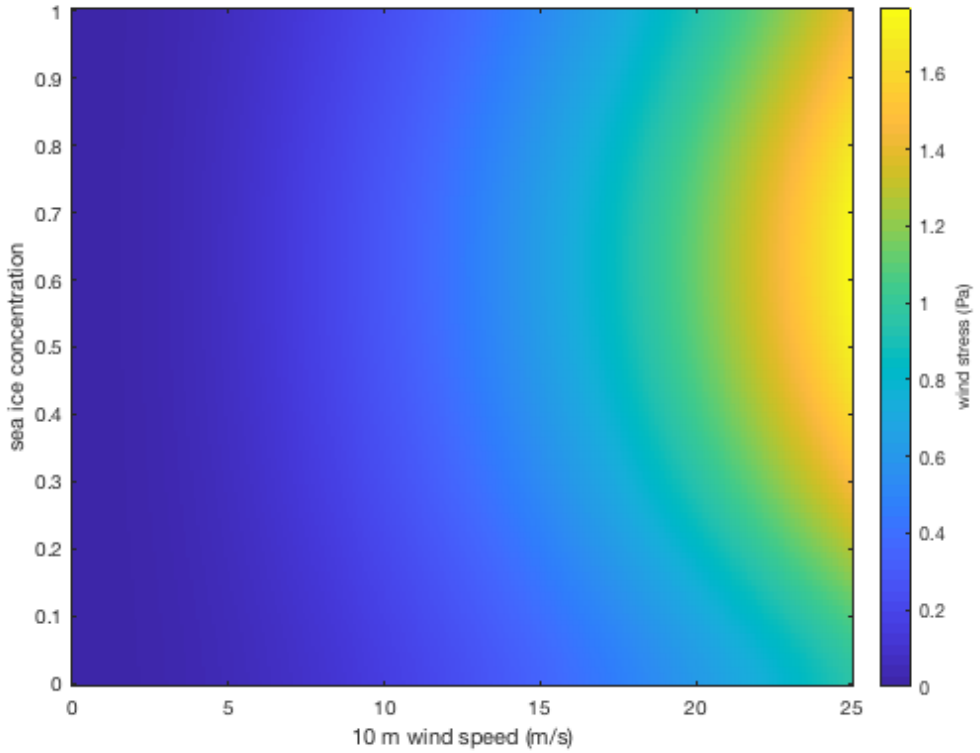
以上，正确的（蓝色）风应力矢量与黑色风速矢量对齐，而从风速分量单独计算风应力分量会产生不正确大小和方向的红色风应力矢量。

示例 2: （理论）海冰效应

探索风速、海冰浓度和风应力之间的关系。首先定义风速从 0 到 25 m/s，海冰浓度从 0 到 1 的值，然后计算风应力并将其绘制为风速和海冰浓度的函数：

```
% 定义风速和海冰浓度的一些值
[U,ci] = meshgrid(0:0.1:25,0:0.01:1);

% 计算风应力:
Tau = windstress(U,'ci',ci);

% 将风应力绘制为风速和海冰浓度的函数:
figure
imagescn(U,ci,Tau)
xlabel('10 m wind speed (m/s)')
```

```
ylabel('sea ice concentration')
cb = colorbar;
ylabel(cb,' wind stress (Pa) ')
```



注意到一开始风应力是如何随着海冰浓度的增加而增加的，因为海冰锯齿状的边缘给了风一些可以抓住的东西，但是当海冰浓度超过 63%时，阻塞的冰场开始阻止风将能量转移到水里。

参考文献

Kara, A. Birol, et al. "Wind stress drag coefficient over the global ocean." Journal of Climate 20.23 (2007): 5856-5864. doi:10.1175/2007JCLI1825.1.

Lüpkes, Christof, and Gerit Birnbaum. "Surface drag in the Arctic marginal sea-ice zone: a comparison of different parameterisation concepts." Boundary-layer meteorology 117.2 (2005): 179-211. doi:10.1007/s10546-005-1445-8.

作者简介

windstress 函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 于 2017 年 2 月写的。

ekman 文档

ekman 函数估计了 10 m 风的经典 Ekman 输送和上升流/下降流。
另请参见: windstress 和 cdtcurl.

语法

```
[UE,VE,wE] = ekman(lat,lon,u10,v10)
[UE,VE,wE] = ekman(...,'Cd',Cd)
[UE,VE,wE] = ekman(...,'rho',waterDensity)
[UE,VE,wE,dE] = ekman(...)
```

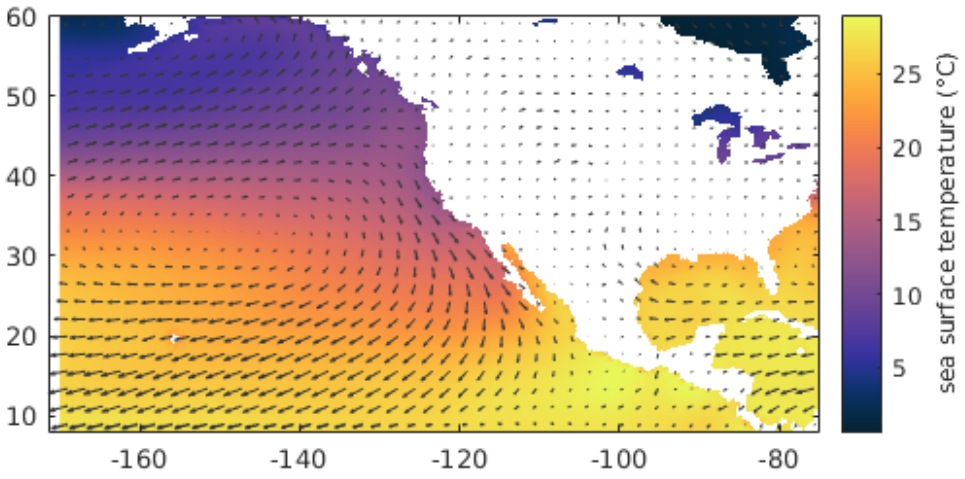
说明

[UE,VE,wE] = ekman(lat,lon,u10,v10) 估计了纬向(UE, m^2/s)和经向(VE, m^2/s) Ekman 层传输以及与 Ekman 泵送相关的垂直速度（wE，m/s）。我们的正值表示上升流。输入 lat 和 lon 必须是二维网格，其尺寸为纬向(u10, m/s)和经向(v0, m/s)风速取地面以上 10m。在执行 Ekman 传输计算之前，风速通过 windstress 自动转换为风应力。输入 u10 和 v10 可以是与 lat 和 lon 大小相同的二维网格，也可以是三维矩阵，其前两个维度对应于 lat 和 lon，第三个维度对应于时间。
[UE,VE,wE] = ekman(...,'Cd',Cd) ）指定风应力计算的阻力系数。Cd 可以是标量，也可以是维数与 u10 和 v10 匹配的矩阵。默认 Cd 为 1.25e-3。
[UE,VE,wE] = ekman(...,'rho',waterDensity) 指定水密度。默认值为 1025 kg/m³。
[UE,VE,wE,dE] = ekman(...) 也给出了 Ekman 层深度 dE 的近似值。

示例

这是一些风和海面温度数据。加载并绘制海面温度和风。下面我使用 imagescn 使 NaN 值透明，但是如果您愿意，可以使用普通的旧 imagesc 。颜色图由 cmocean 函数设置(Thyng et al., 2016)。为了得到深灰色的 RGB 值，我使用了 rgb。

```
load pacific_wind.mat

figure
imagescn(lon,lat,sst)
axis image      % (修剪掉多余的空白并将纵横比设置为 1:1)
cmocean thermal % 设置颜色图
cb = colorbar;
ylabel(cb,' sea surface temperature ({\circ}C) ')

hold on
quiversc(lon,lat,u10,v10,'color',rgb('dark gray'))
```



为了避免任何可能的尺寸混淆，ekman 函数不接受 lat 和 lon 作为向量，因此在计算 ekman 传输之前，我们必须使用 meshgrid 将 lat 和 lon 转换为二维网格。

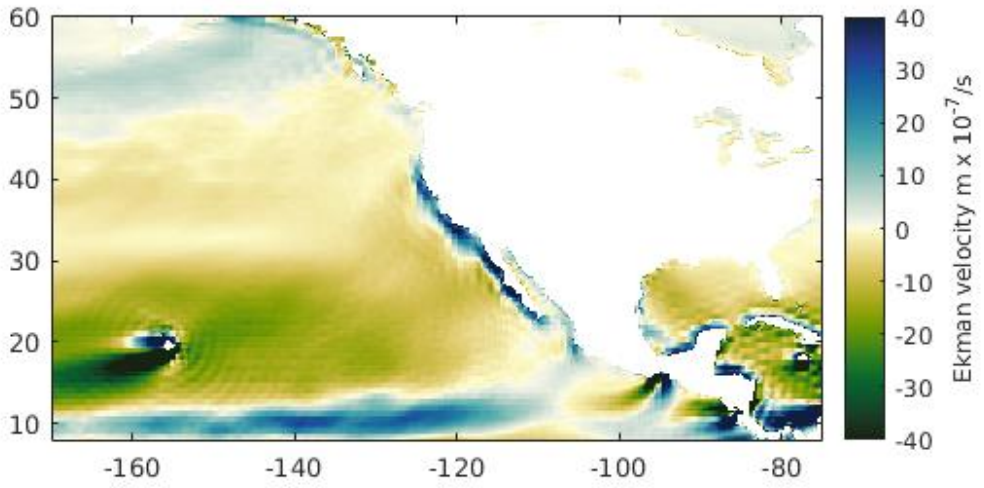
```
[Lon,Lat] = meshgrid(lon,lat);

[UE,VE,wE] = ekman(Lat,Lon,u10,v10);
```

```
Warning: You have included some data points within 10 degrees of the equator.

Some folks say Ekman is invalid within 10 degrees of the equator, others say

Ekman's formulas work as close as two or three degrees from the equator. There

is no clearly defined cutoff latitude, but the issue is that Ekman divides by

the Coriolis frequency, which approaches zero at the equator.
```

（注意纬度在赤道 10 度以内时出现的警告。）
在掩盖陆地的值后绘制垂直速度 wE。方便的是，我们的 sst 数据集在任何有陆地的地方都是 NaN，所以我们可以用 sst 制作一个掩膜(如果数据集不包含区分陆地和海洋的清晰方法，我们可以使用 island 代替。)

```
% 掩盖陆地值:
wE(isnan(sst)) = nan;
```

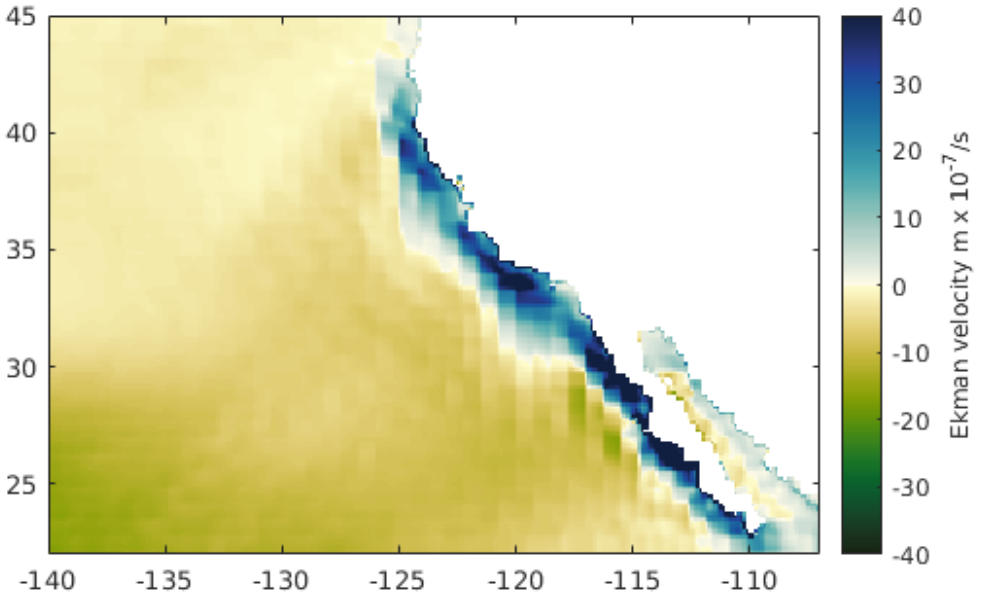
```
figure
imagescn(lon,lat,wE*1e7);
axis image
caxis([-1 1]*40)
cmocean -delta
cb = colorbar;
ylabel(cb,'Ekman velocity m x 10^{-7}/s')
```



加利福尼亚海岸上升流

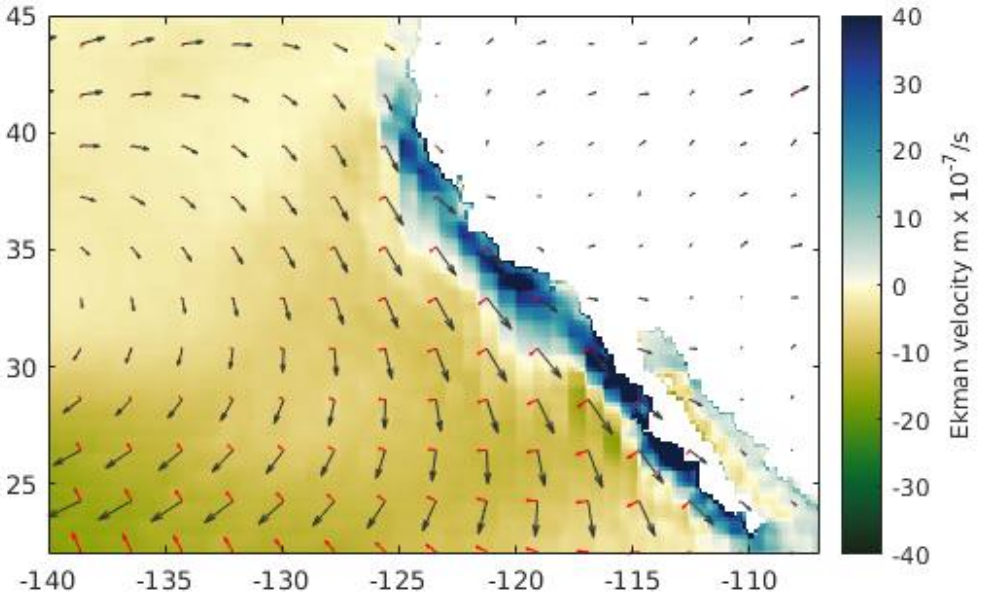
加利福尼亚海岸附近的上升流是气候和生物学的一个重要过程。让我们仔细看一看：

```
axis([-140 -107 22 45])
```



蓝色区域对应于上升流，是地表水扩散输送的结果。让我们像以前一样用深灰色绘制风矢量，但现在我们也将 Ekman 传输绘制为红色矢量：

```
hold on

% 绘制风矢:
quiversc(lon,lat,u10,v10,'color',rgb('dark gray'));

% 绘制 Ekman 传输矢量:
quiversc(lon,lat,UE,VE,'color',rgb('bright red'));
```



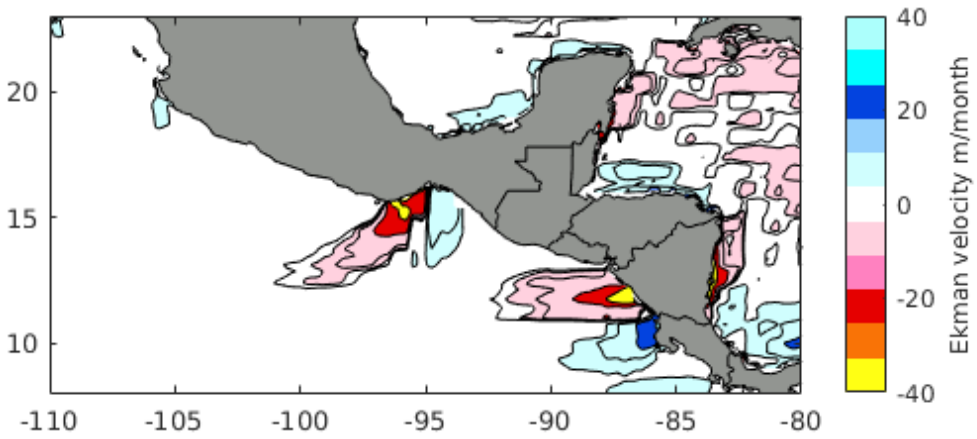
正如我们所料，在北半球，表层输送与风向成 90 度角，这就是 Ekman 上升流的驱动力。

模仿 Kessler

同样，让我们来看看 Kessler 2002 所分析的区域。我们将重新创建 Kessler 的图 6b，它以米为单位绘制每月的上升流。一个月有 2629800 秒，所以我们必须在绘制 wE 之前乘以这个数字。另外，Kessler 的颜色轴是非线性的，所以我们将使用它们的等值线，但我不担心如何匹配和压缩颜色条来匹配 Kessler。

```
cvals = [-40 -20 -10 -5 -2.5 -1 - 2.5 5 10 20 40];



figure

contourf(lon,lat,wE*2629800,cvals);

axis xy image
caxis([-1 1]*40)
cb = colorbar;
ylabel(cb,'Ekman velocity m/month')
axis([-110 -80 8 23])

%设置颜色图:
colormap(rgb('yellow','orange','red','pink','light pink','white',...
   'pale blue','light blue','blue','cyan','light cyan'));
```

```
borders('countries','facecolor',rgb('gray'))
```



注：因为上升流和下降流是同一个硬币的两面，所以最好使用一个真正发散的、感知上一致的彩色地图，比如 `cmocean` 函数提供的那些。尽管如此，我们试图模仿已发布的绘图，因此用 `rgb` 定义颜色是最简单的解决方案。

参考文献

Kessler, William S. "Mean three-dimensional circulation in the northeast tropical Pacific." Journal of Physical Oceanography 32.9 (2002): 2457-2471. doi:10.1175/1520-0485-32.9.2457.

Thyng, K.M., C.A. Greene, R.D. Hetland, H.M. Zimmerle, and S.F. DiMarco. 2016. True colors of oceanography: Guidelines for effective and accurate colormap selection. Oceanography 29(3):9-13, doi:10.5670/oceanog.2016.66.

作者简介

`ekman` 函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 于 2017 年 2 月写的。

coriolisf 文档

coriolisf 函数返回任意给定纬度的科里奥利频率。科里奥利频率有时被称为科里奥利参数或科里奥利系数。

语法

```
f = coriolisf(lat)
f = coriolisf(lat,rot)
```

说明

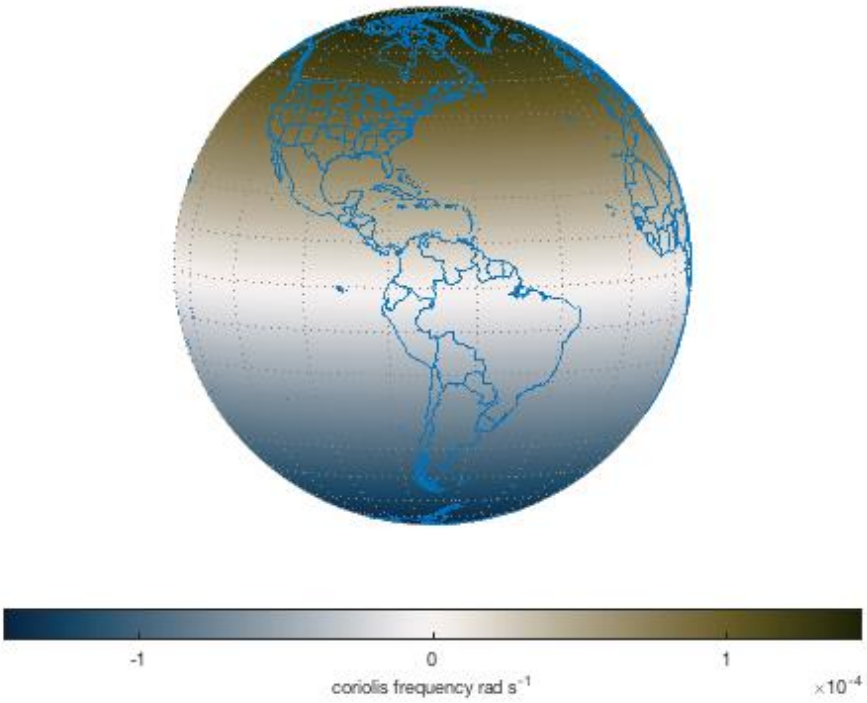
f = coriolisf(lat) 返回纬度 lat 上任何位置的地球科里奥利频率。默认情况下，科里奥利频率以 rad/s 为单位。

f = coriolisf(lat,rot) 指定旋转速率 rot 。默认情况下，rot 是地球当前的旋转速率 7.2921 x 10^-5 rad/s，但可以指定不同的速率来模拟不同时间的地球、其他天体或不同频率单位的 Coriolis 参数。

示例

用 cdtgrid 做一个 1°x1°的网格，得到每个网格单元对应的科里奥利参数：

```
[Lat,Lon] = cdtgrid;



f = coriolisf(Lat);

globepcolor(Lat,Lon,f)          % pcolor 科里奥利频率
globeborders                    % 国界线
globegraticule('linestyle',':') % 网格线

view(20,5)                      % 设置全球视角 sets the globe viewing angle

caxis([-1 1]*0.146e-3)          % 设置颜色坐标轴限制
cb = colorbar('location','southoutside');
xlabel(cb,'coriolis frequency rad s^{-1}')
cmocean diff                    % 设置颜色图
axis tight                      % 最小化空闲空间
```



作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

rossby_radius 文档

rossby_radius 计算正压海洋的 Rossby 变形半径,

$$L_R = \frac{\sqrt{gD}}{|f|}$$

式中，L_R是罗斯比变形半径，g是重力加速度，D是水深，f是 coriolisf 给出的科里奥利斯参数。

语法

```
Lr = rossby_radius(lat,lon)
Lr = rossby_radius(lat,'depth',D)
Lr = rossby_radius(...,'rot',rot)
Lr = rossby_radius(...,'g',gravity)
```

说明

Lr = rossby_radius(lat,lon) 计算地理坐标 lat，lon 处海洋正压 Rossby 变形半径。Rossby 半径的计算取决于海洋深度，海洋深度通过 topo_interp 函数自动确定。

Lr = rossby_radius(lat,'depth',D) 通过输入深度 D 覆盖海洋深度的自动计算，深度 D 可以是标量或与 lat 相同的大小。深度应为正。

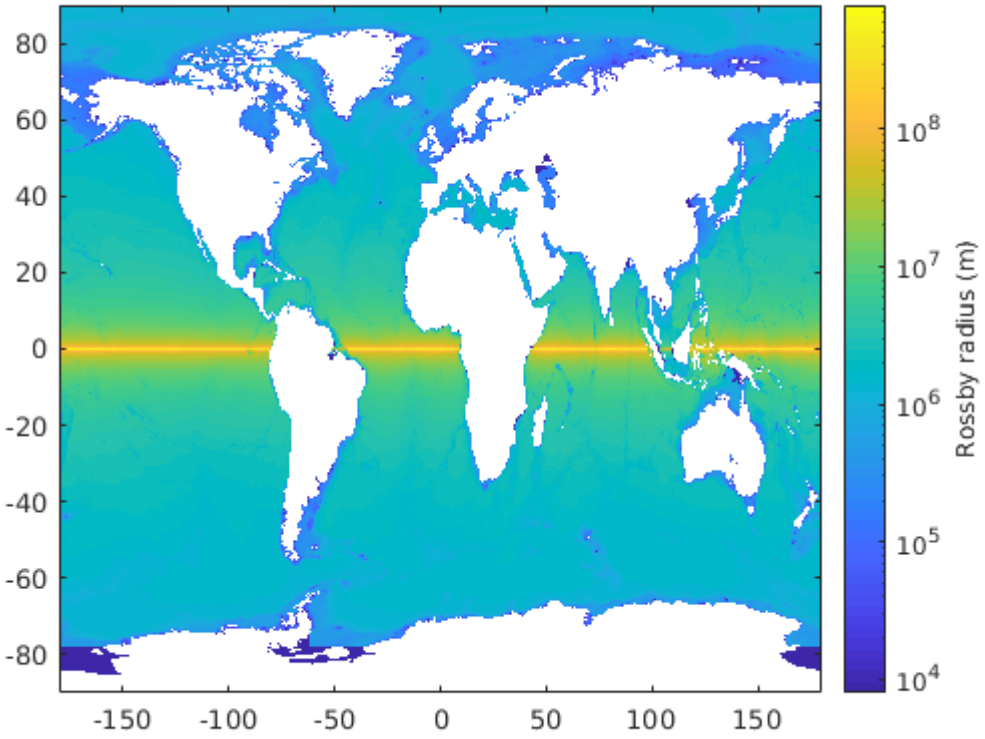
Lr = rossby_radius(...,'rot',rot) 指定旋转速率。默认情况下，rot 是地球当前的自转速率 7.2921 x 10^-5 rad/s，但是可以指定不同的速率来模拟不同时间的地球、其他天体。

Lr = rossby_radius(...,'g',gravity) 指定重力加速度。默认值为 9.81.

示例: 全球海洋

得到地球海洋中任何地方的正压 Rossby 半径。首先用 cdtgrid 生成覆盖世界的四分之一度网格，然后计算每个网格点的 Rossby 半径：

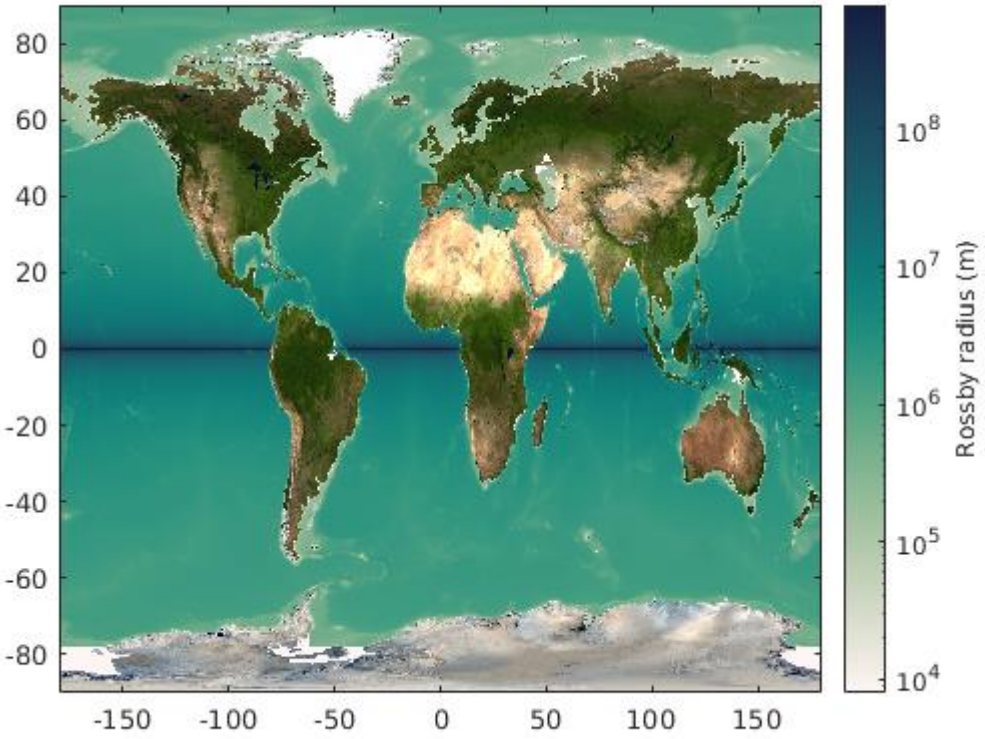
```
[lat,lon] = cdtgrid(0.25);



r = rossby_radius(lat,lon);



% 画起来:
imagescn(lon,lat,r)

% 设置颜色轴属性:
set(gca,'colorscale','log')
%caxis([1e5 1e7]) % 设置颜色轴范围
cb = colorbar;
ylabel(cb,'Rossby radius (m)')
```



我们看到的是，在两极附近和任何浅水区，Rossby 半径都很小。在赤道深水区，Rossby 半径相当大！

通过将 colormap 更改为 `cmocean` *tempo* 颜色图和带有 `earthimage` 的参考底图使其变得奇怪：

```
cmocean tempo


hold on
h = earthimage;
uistack(h,'bottom') % 将地球图片放在 Rossby 半径上面
```



作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

binind2latlon 文档（参见前文）

mld 文档

mld 函数根据温度、盐度和/或潜在密度的分布计算水柱的混合层深度。

海洋混合层是指海洋上层混合良好，湍流混合，具有恒定的温度和盐度值。虽然混合层的概念相对简单，但对于如何具体定义它缺乏统一的定量定义。存在许多算法，从简单的阈值算法到复杂的形状拟合工具，并且有大量文献对每种算法的相对优点进行了辩论。

对于这个特殊的工具箱函数，我们选择了 Holte and Talley, 2009 的算法。本文详细介绍了一种计算海洋 Argo 剖面混合层的多算法方法。这些算法包含了最常用的指标，因此是该功能的良好起点，该功能还致力于提供多种混合层深度方法。下面的示例详细介绍了一些最常用的计算方法。请始终记住，混合层深度计算不是一刀切的计算，您可能需要尝试许多不同的选项，以确定哪个选项最适用于您的数据。

请注意，该功能目前包括一些特定于 Holte& Talley 论文的习惯（例如，在某些算法下，盐度剖面与温度和密度剖面的处理方式不同），这些习惯是为开阔海洋、浮标测量剖面量身定制的。这些可能适用于也可能不适用于所有数据集。为了与该函数所基于的原始 Holte& Talley 代码保持一致，我们在这里保留了它们以及默认参数选择。

语法

```
pmld = mld(pres, temp)
pmld = mld(pres, temp, salt)
pmld = mld(..., Name, Value, ...)
[pmld, mldtbl, pth, h] = mld(...)
```

说明

pmld = mld(pres, temp) 计算与水柱混合层深度相关的压强，水柱混合层深度由压强向量 pres （单位：db）和温度向量 temp （单位：摄氏度）定义。在仅温度计算中，压强值仅用作垂直轴，因此可以替换深度值（单位：m）。

pmld = mld(pres, temp) 计算水柱的混合层深度压强，该压强由压强向量 pres （db）、温度向量 temp （摄氏度）和盐度向量 salt 定义。请注意，在这种情况下，压强值将用于计算潜在密度，因此用户应谨慎使用深度值（单位：m）代替压强。

pmld = mld(..., Name, Value, ...) 允许用户通过下一节中列出的 Name-Value 参数修改算法参数。

[pmld, mldtbl, pth, h] = mld(...) 返回有关所用内部算法的附加信息。如果指定了'best'算法，mldtbl 表将保存使用的所有混合层深度压力候选值。pth 值返回使用'best'算法时使用的算法路径；这主要用于调试目的，旨在匹配原始 Holte&Talley 2009 补充在线代码中的分析变量、分析变量和分析变量（请注意，此代码与在线补充代码不同，但在大多数情况下复制了相同的路径）。最后，如果指定了 plot 输入选项，图形句柄将在句柄结构 h 中返回。

Name-Value 参数

metric'threshold','gradient','fit','extrema','subsurface','best' **（默认）**

此功能允许使用 6 种不同的方法定义混合层深度：

- threshold:混合层深度是从参考深度向下测量的第一个深度，其中轮廓值与参考深度值的差异超过阈值。仅适用于温度和密度分布。
- gradient:从参考深度向下测量的第一个深度，其中剖面梯度超过指定的阈值量。如果梯度从未超过阈值，则使用最高绝对梯度的深度。对于盐度剖面，只需寻找最大梯度。
- fit:一条线拟合到轮廓的顶部、垂直部分和轮廓的最高渐变、水平部分。混合层深度设置为这些线相交处的压强，如果它们不相交，则设置为 0。

- extrema:压强极值。温度最大，盐度和密度最小。请注意，虽然这些值可能适用于大多数海洋剖面，但在某些情况下（例如，冰环境下以盐度为主的温度反演），这种简单的度量肯定会失败。
- subsurface:该方法试图在混合层底部发现地下异常，这是地表冷却和混合事件的结果。剖面的特征是温度梯度最大值接近温度最大值（或盐度最小值）。如果发现，mld 值反映两个值中较浅的值；如果未找到，则将其设置为 0。仅适用于温度和盐度剖面。
- 'best':根据上述选项计算所有适用的潜在 mld 值，并使用 Holte&Talley 2009 算法选择给定剖面的最佳估计值。

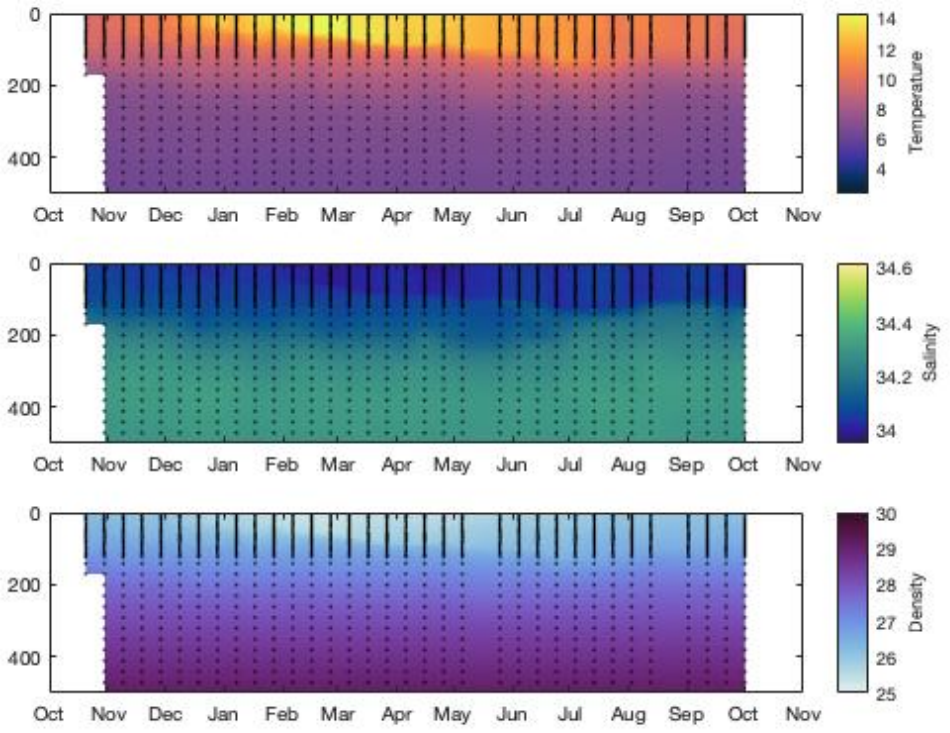
refpres 数值标量，默认值=10
参考压强（db）。计算基于剖面向下的这一点；忽略上面的点，目的是消除表面上的任何日间加热瑕疵。
tthresh 数值标量，默认值为 0.2
用于温度阈值计算的阈值（摄氏度）。
tgradthresh 数值标量，默认值=0.005
用于温度梯度计算的阈值（摄氏度 m^-1）。
dthresh 数值标量，默认值=0.03
用于密度阈值计算的阈值（kg m^-3）
dgradthresh 数值标量，默认值=0.0005
用于密度梯度计算的阈值（kg m^-4）
errortol 数值标量，默认值=1e-10
用于选择拟合算法中要包含在曲面垂直线中点的公差。截止设置为拟合的标准化平方和误差小于该公差的最深点。
range 标量，默认值=25
在选择最佳算法期间，用于识别 mld 值簇的范围参数，即 Holte&Talley，2009，第 3b 节（db）。
deltad 数值标量，默认值=100
在地下温度[盐度]计算中，最大[最小]值和梯度之间允许的最大距离 (db)。
tcutoffu 数值标量，默认值=0.5
在选择 best 算法期间，将温度剖面分类为类似冬季时的温度变化截止上限值(摄氏度）
tcutoffl 数值标量，默认值=-0.25
在选择 best 算法期间，将温度剖面分类为类似冬季时的温度变化截止下限值 (摄氏度）
dcutoff 数值标量，默认值=-0.06
在选择 best 算法期间，潜在密度变化截止值高于此值的剖面被分类为类似冬季时。(kg m^-3)
plot 逻辑标量，默认值=False
若为 true，则创建具有轮廓、混合层深度点和一些参考细节的图。
tblformat 'table' (默认),'array'
mldtbl 输出由一个 3 x 5 数组组成，其中行对应于 5 种算法（不包括 best），列分别对应于温度、盐度和密度。默认的表格式更具自描述性，因为列和行都有明确的标签，但创建表数组会给函数执行增加大量时间；如果多次调用此函数，建议使用数组输出(请注意，如果仅使用一个输出调用此函数，则这将变得不相关）。

示例 1: 阈值法

阈值定义是迄今为止最常用的混合层深度度量，并且是几种全球混合层深度产品的基础，如 Monterey and Levitus (1997)和 de Boyer Montegut et al. (2004)。
为了复制这种类型的计算，我们从 Argo 浮点的示例数据开始。我们使用 transect 函数将该数据可视化，将 34 个剖面转换为样条图，并使用 cmocean 设置颜色图。

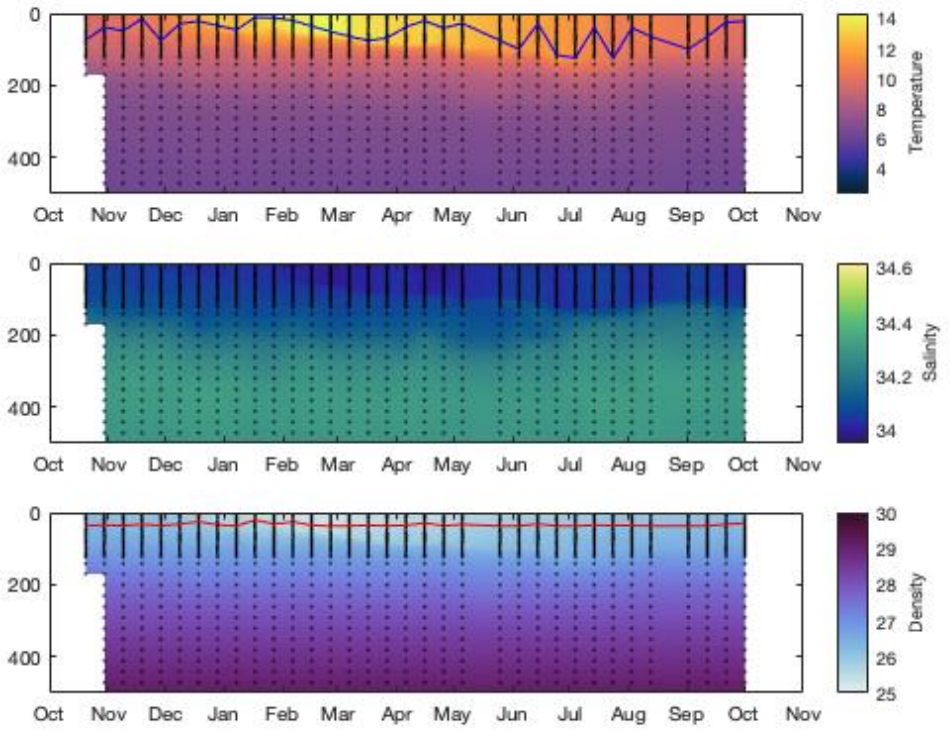
```
A = load('example_argo.mat');
```

```
nprof = length(A.date);
ax(1) = subplot(3,1,1);
transect(A.date, A.PRES, A.TEMP,'markersize',2);
hold on;
cb(1) = colorbar('eastoutside');
ylabel(cb(1), 'Temperature');
datetick;
cmocean('thermal');

ax(2) = subplot(3,1,2);
transect(A.date, A.PRES, A.PSAL,'markersize',2);
hold on;
cb(2) = colorbar('eastoutside');
ylabel(cb(2), 'Salinity');
datetick;
cmocean('haline');

ax(3) = subplot(3,1,3);
transect(A.date, A.PRES, A.pden,'markersize',2);
hold on;
cb(3) = colorbar('eastoutside');
ylabel(cb(3), 'Density');
datetick;
cmocean('dense');

set(ax, 'YLim', [0 500]);
set(ax(3), 'clim', [25 30]);
```



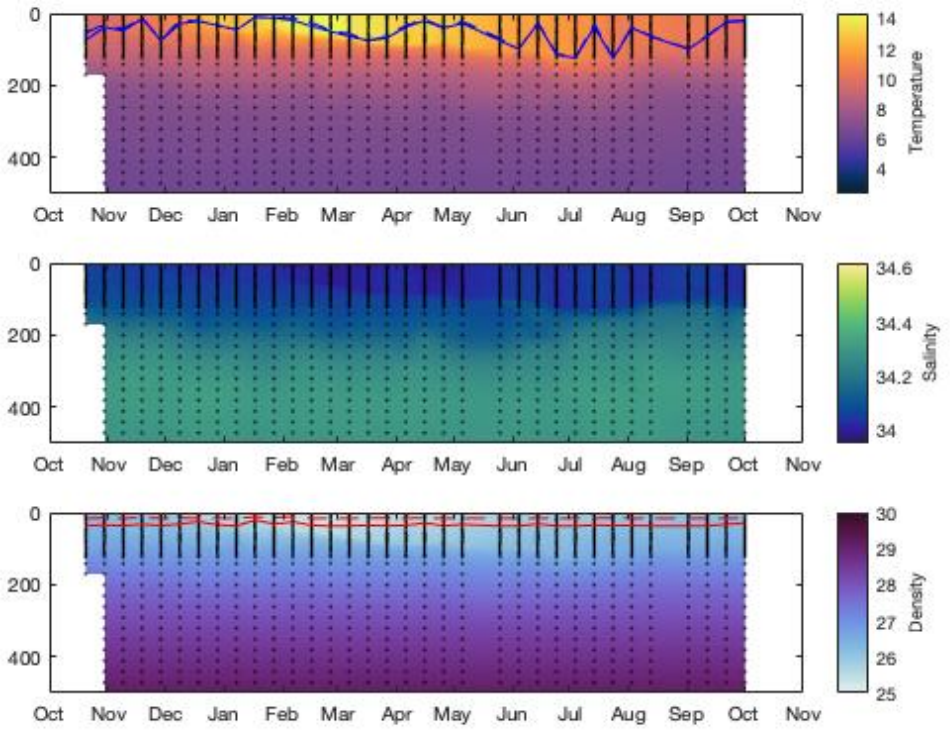
Monterey and Levitus (1997)使用 0.5℃的温度阈值和 0.125 kg/m^3 的密度阈值。这是 MLD 计算的典型默认选择。

```
pmld = zeros(nprof,3);

for ix = 1:nprof
    pmld(ix,:) = mld(A.PRES{ix}, A.TEMP{ix}, A.PSAL{ix}, 'metric', 'threshold', ...
        'tthresh', 0.05, 'dthresh', 0.125);
end

plot(ax(1), A.date, pmld(:,1), 'b');
plot(ax(3), A.date, pmld(:,3), 'r');
```



de Boyer Montegut et al. (2004)认为 Monterey and Levitus (1997)的阈值适用于网格和平均剖面，但不太适合直接剖面测量。它们分别使用 0.2℃和 0.03 kg/m^3 的温度和密度阈值。让我们看看这与之前对这组剖面文件的估计相比如何：

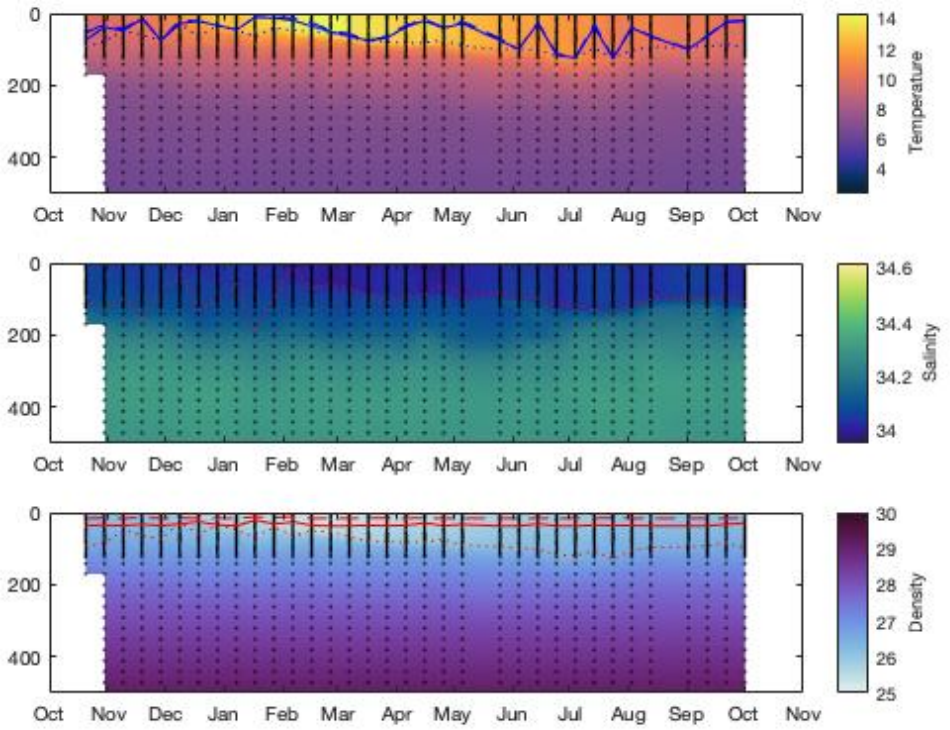
```
pmld = zeros(nprof,3);

for ix = 1:nprof
    pmld(ix,:) = mld(A.PRES{ix}, A.TEMP{ix}, A.PSAL{ix}, 'metric', 'threshold', ...
        'tthresh', 0.02, 'dthresh', 0.03);
end

plot(ax(1), A.date, pmld(:,1), '--b');
plot(ax(3), A.date, pmld(:,3), '--r');
```



示例 2: 拟合方法

Holte& Talley，2009 年的拟合方法通常在识别深冬混合层深度方面更为稳健，并且可以应用于所有三种剖面类型。此功能通过将两条线拟合到轮廓来工作，一条接近垂直的线表示轮廓的混合部分，另一条对角线表示混合层底部的过渡部分。请注意，这是五个指标中最耗时的选项。

```
pmld = zeros(nprof,3);

for ix = 1:nprof
    pmld(ix,:) = mld(A.PRES{ix}, A.TEMP{ix}, A.PSAL{ix}, 'metric', 'fit');
end

plot(ax(1), A.date, pmld(:,1), ':b');
plot(ax(2), A.date, pmld(:,2), ':r');
plot(ax(3), A.date, pmld(:,3), ':r');
```

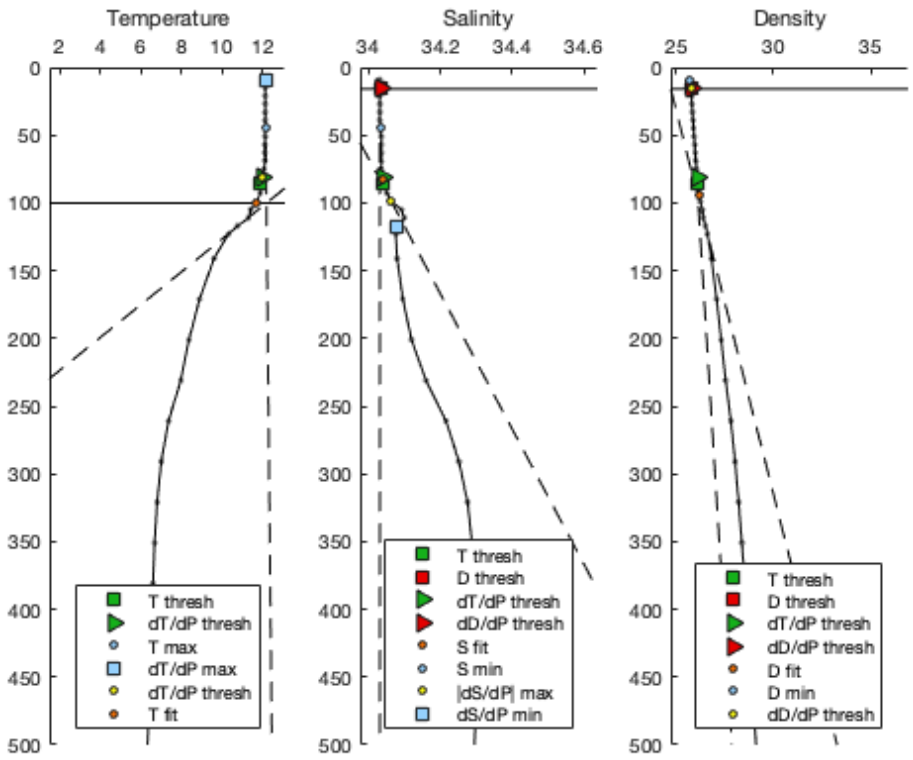


示例 3: 替代方法

此函数中介绍的 MLD 计算的其余方法不太常见，并且大部分已开发用于处理简单阈值计算无法识别所需 MLD 值的情况。请参阅 Holte& Talley，2009 年的完整论文，以了解这些算法的详细描述及其发展背后的动机。

此功能附带的自动绘图功能可快速总结各种可能性。在该图中，T=温度，S=盐度，D=密度（σθ），P=压力。虚线对应拟合方法最佳拟合线，实线是最终的"best" MLD。所有这些单个候选 MLD 值都可以在 mldtbl 输出中找到：

```
[pmld, mldtbl, pth, h] = mld(A.PRES{22}, A.TEMP{22}, A.PSAL{22}, 'plot', true);
```




```
mldtbl
```

```
mldtbl =
```

3×5 table

	threshold	gradient	fit	extrema	subsurface
temp	85.875	81	100.18	45	9
salt	NaN	99	83	45	45
pdens	15.885	15	94.6	9	NaN

最合适的算法取决于给定水团的特性。请记住，"best"方法是专门为 Argo 浮标剖面设计的（主要在南大洋剖面上测试）；由于该函数的历史记录，我们将其作为默认值提供（旨在模仿 Holte& Talley，2009 补充代码，但具有更大的灵活性）。

参考文献

de Boyer Montégut C, Madec G, Fischer AS, Lazar A, Iudicone D (2004) Mixed layer depth over the global ocean: An examination of profile data and a profile-based climatology. J Geophys Res C Ocean 109:1–20

Holte J, Talley L (2009) A new algorithm for finding mixed layer depths with applications to argo data and subantarctic mode water formation. J Atmos Ocean Technol 26:1920–1939 doi:10.1175/2009JTECHO543.1

Monterey, G. and Levitus, S., 1997: Seasonal Variability of Mixed Layer Depth for the World Ocean. NOAA Atlas NESDIS 14, U.S. Gov. Printing Office, Wash., D.C., 96 pp.

作者简介

这个函数和支持文档是华盛顿大学的 Kelly A. Kearney 在 2019 年为 Climate Data Toolbox for Matlab 写的。

transect 文档

transect 生成了一个从不同地点和/或时间收集的 CTD 剖面的海洋学数据的彩色标度样带图。
另请参见: transectc.

语法

```
transect(x,d,v)
transect(...,'LineProperty',Value)
transect(...,'bed',BedDepth)
transect(...,'bedcolor',ColorSpec)
transect(...,'extrap')
transect(...,'interp','InterpMethod)
transect(...,'di',di)
transect(...,'xi',xi)
[h_trans,h_marker,h_bed] = transect(...)
```

说明

transect(x,d,v) 在水平位置（或时间）x 和深度 d 处创建变量 v 的彩色标度样条图。x 必须是一个数值数组，表示每个 CTD 配置文件的位置或时间。输入 d 和 v 必须是包含每个 CTD 位置的深度和测量值的单元阵列(请参见下面有关将数据转换为单元阵列的部分)。
transect(...,'LineProperty',Value) 为每个数据点设置线或标记样式。默认情况下，每个数据点都由一个黑点表示，但是配置文件也可以很容易地用不同的标记类型或指定颜色、样式、厚度等的线来表示。
transect(...,'bed',BedDepth) 指定海床深度，以创建指示横断面沿线河床海床纵断面的面片对象。
transect(...,'bedcolor',ColorSpec) 指定海床的颜色。默认 bedcolor 为黑色。
transect(...,'extrap') 启用了超出数据范围的外推选项。默认情况下，横断面不进行外推，但该选项可用于填充间隙，例如最低点和海床之间的间隙。
transect(...,'interp','InterpMethod) 指定插值方法（请参见下面的方法部分）。默认值为 'linear'。
transect(...,'di',di) 指定插值深度 di。默认情况下，transect 通过在最浅和最深测量之间插值到 1000 个等距深度来创建栅格。
transect(...,'xi',xi) 指定插值位置（或时间）xi。默认情况下，transect 插值到 2000 个等距点 xi 在 x 的最小值和最大值之间。
[h_trans,h_marker,h_bed] = transect(...) 返回横断面图、标记和海床填充对象的句柄。

方法

此函数插值两次以创建变量 v 的网格化等值线。第一轮将每个 CTD 剖面的数据插值到等间距深度 di。第二轮水平等距的位置或时间 xi。默认情况下，两个插值都是线性的，不执行外推。

示例

下面的示例使用一些 Argo 数据，您可以这样加载这些数据：

```
load example_ctd

whos % 显示每个变量的名称和尺寸
```

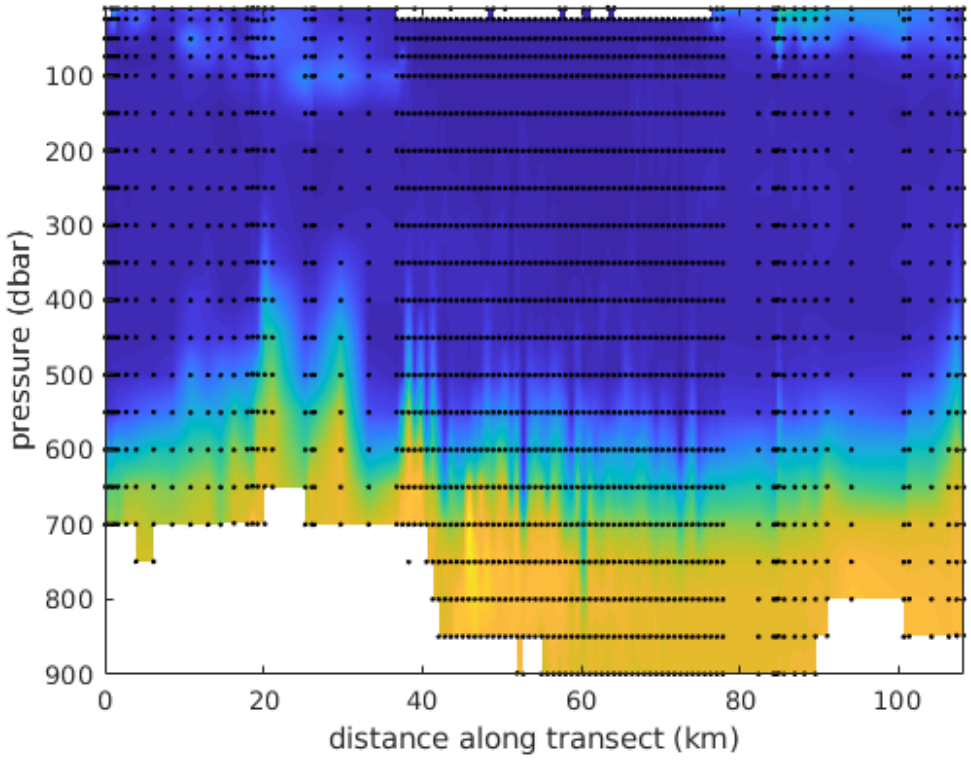
```
  Name      Size            Bytes  Class     Attributes

  P         1x95            25152  cell
  S         1x95            25152  cell
  T         1x95            25152  cell
  bed       1x95              760  double
  x         1x95              760  double
```

Argo 数据包含 95 个浮标漂移时的剖面。每个剖面都是在 x 的某个位置采集的，相应的海床深度由 bed 层给出，温度 T 和盐度 S 在相应压强 P 下的元胞数组中。

示例 1: 最简单的形式

以下是在位置 x 和深度 P 采集的温度数据 T 的简单样带：

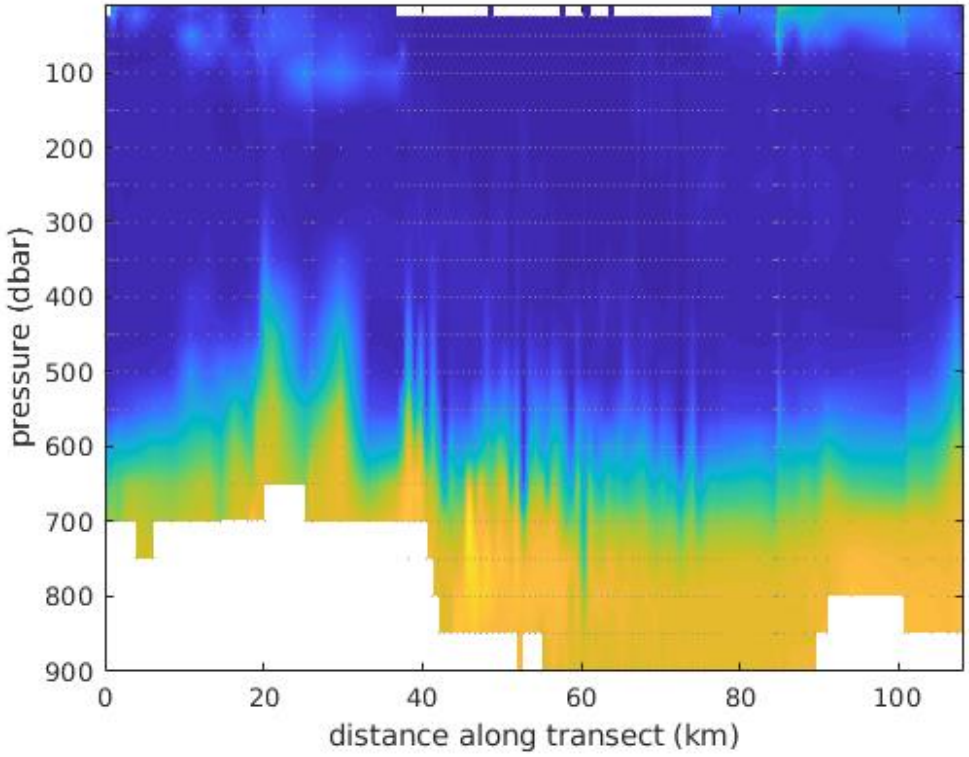
```
transect(x,P,T)



xlabel 'distance along transect (km)'
ylabel 'pressure (dbar)'
```



示例 2: 指定测量标记:

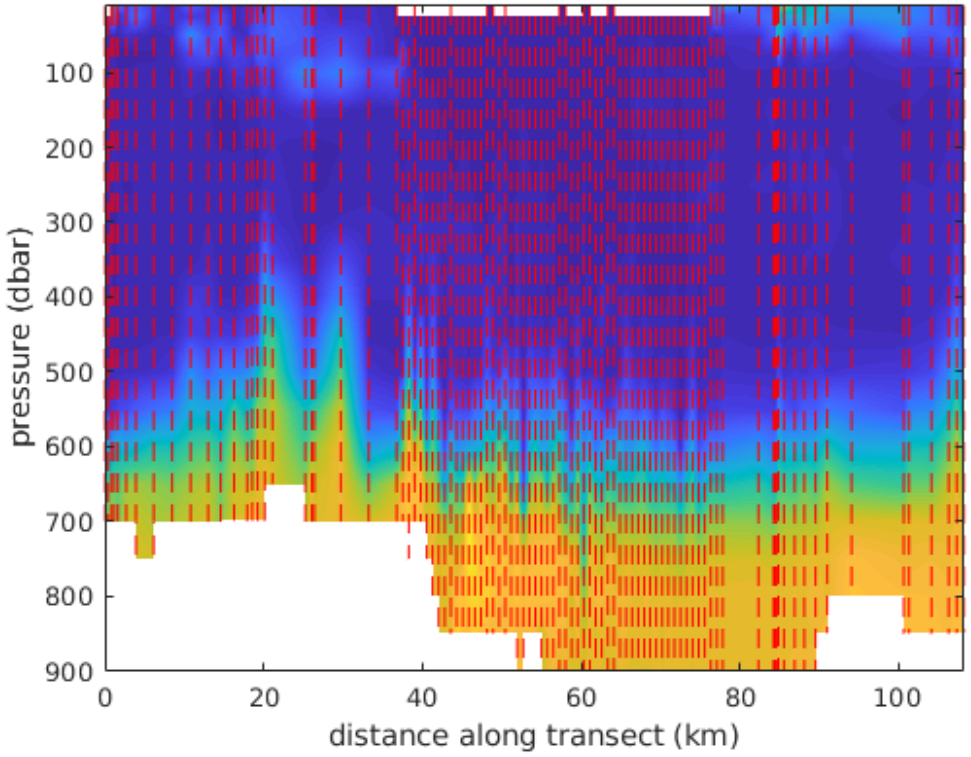
与上面相同，但这一次标记为小灰点：

```
transect(x,P,T,'color',rgb('gray'),'markersize',2)

xlabel 'distance along transect (km)'
ylabel 'pressure (dbar)'
```



或者，您可以使每个纵断面都有自己的线，而不是标记。剖面下方用红色虚线表示：

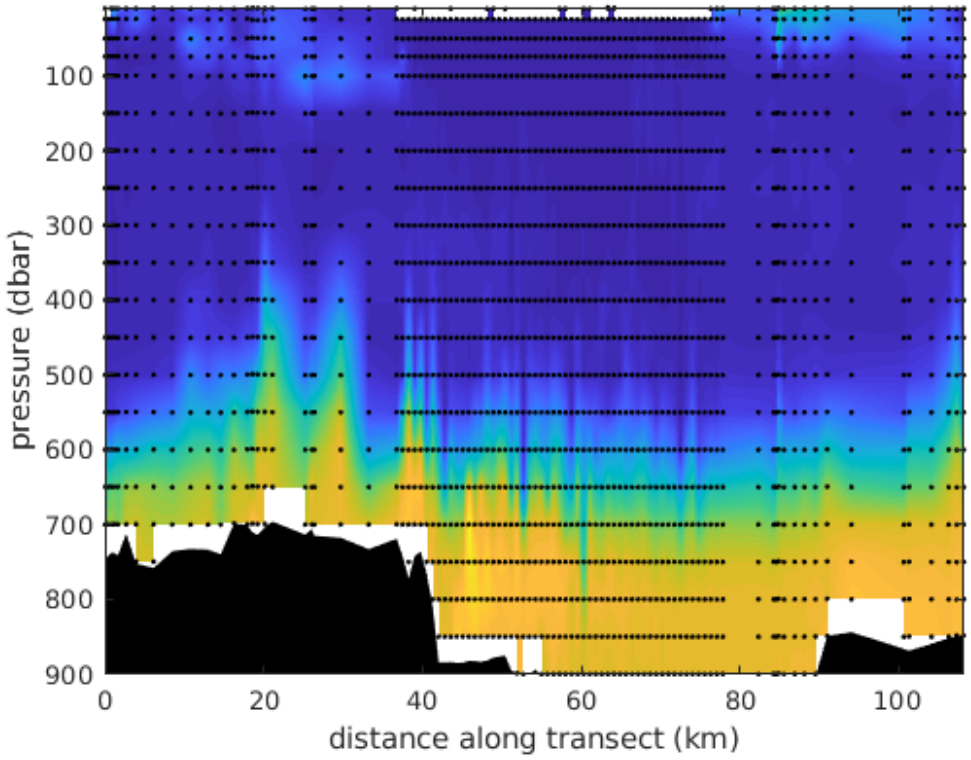
```
transect(x,P,T,'marker','none','linestyle','--','color','r')

xlabel 'distance along transect (km)'
ylabel 'pressure (dbar)'
```



示例 4:显示海床

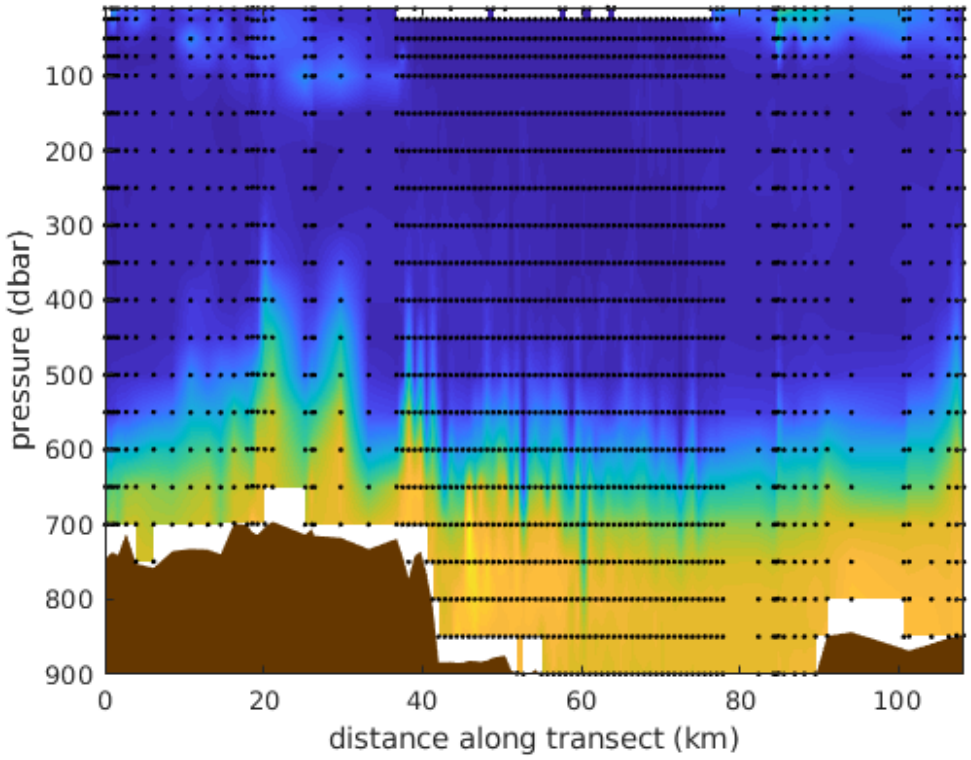
显示海床剖面，只需输入每个 x 值对应的海床深度：

```
transect(x,P,T,'bed',bed)


xlabel 'distance along transect (km)'
ylabel 'pressure (dbar)'
```



通过指定棕色的 RGB 值，将海床涂成棕色。如果您不知道准确的 RGB 值，只需使用 rgb 函数：

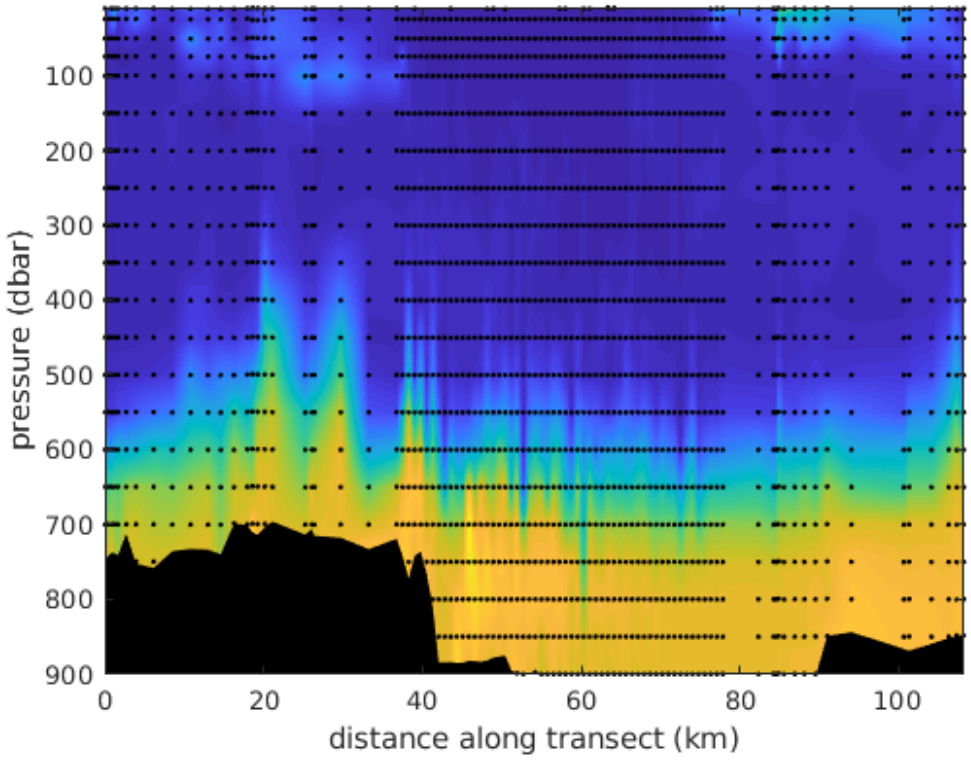
```
transect(x,P,T,'bed',bed,'bedcolor',rgb('brown'))

xlabel 'distance along transect (km)'
ylabel 'pressure (dbar)'
```



示例 5: 填补空白

不喜欢海床和最下面的数据之间的空白吗？或者在靠近等值线的中间之间的表面附近的空白？此函数允许您通过指定'extrap'在基础数据值之外进行推断（如果需要）：

```
transect(x,P,T,'bed',bed,'extrap')

xlabel 'distance along transect (km)'
ylabel 'pressure (dbar)'

caxis([-1.9 0.9]) % 设置颜色坐标轴范围
```

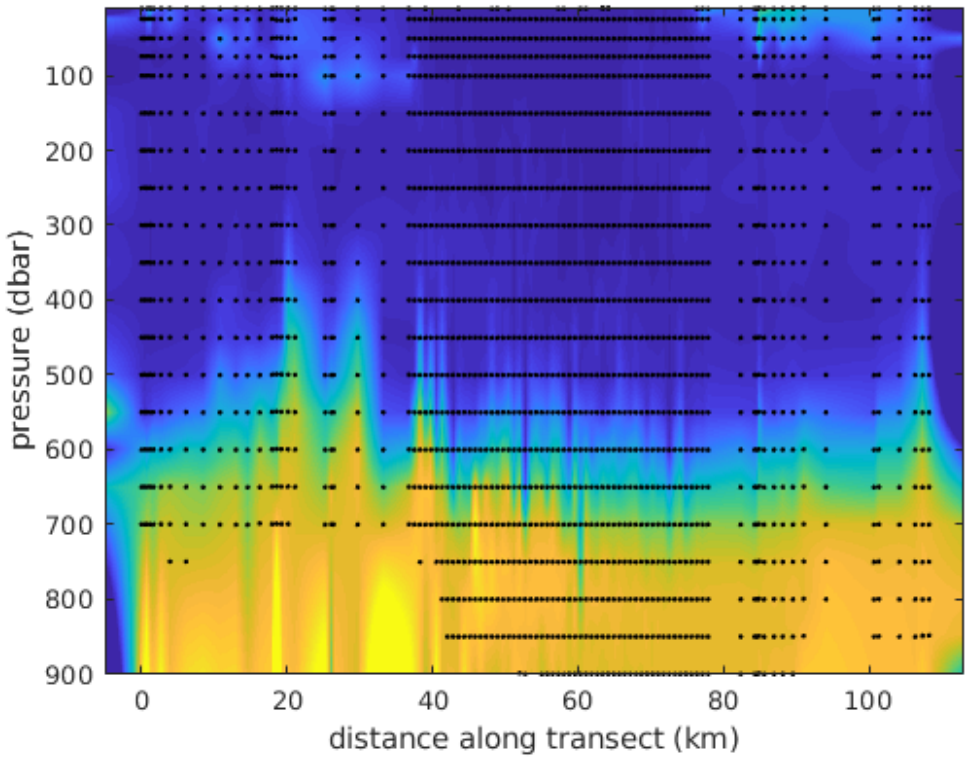


请记住，根据您的应用程序，推断可能不合适。还要注意，外推到左下角海底以下的地方会产生一些相当高的温度值，因此我们必须设置颜色轴以匹配实际的海洋学值。

示例 6: 扩展域

是否要将域扩展到基础数据的范围之外？数据从 0 到 108 公里。通过相应地指定'xi'值，在每侧增加 5 km：

```
% 在-5 和 113 公里之间每 0.1 公里内插一次:
xi = -5:0.1:113;

transect(x,P,T,'extrap','xi',xi)

xlabel 'distance along transect (km)'
ylabel 'pressure (dbar)'

caxis([-1.9 0.9]) % 设置颜色轴范围
```

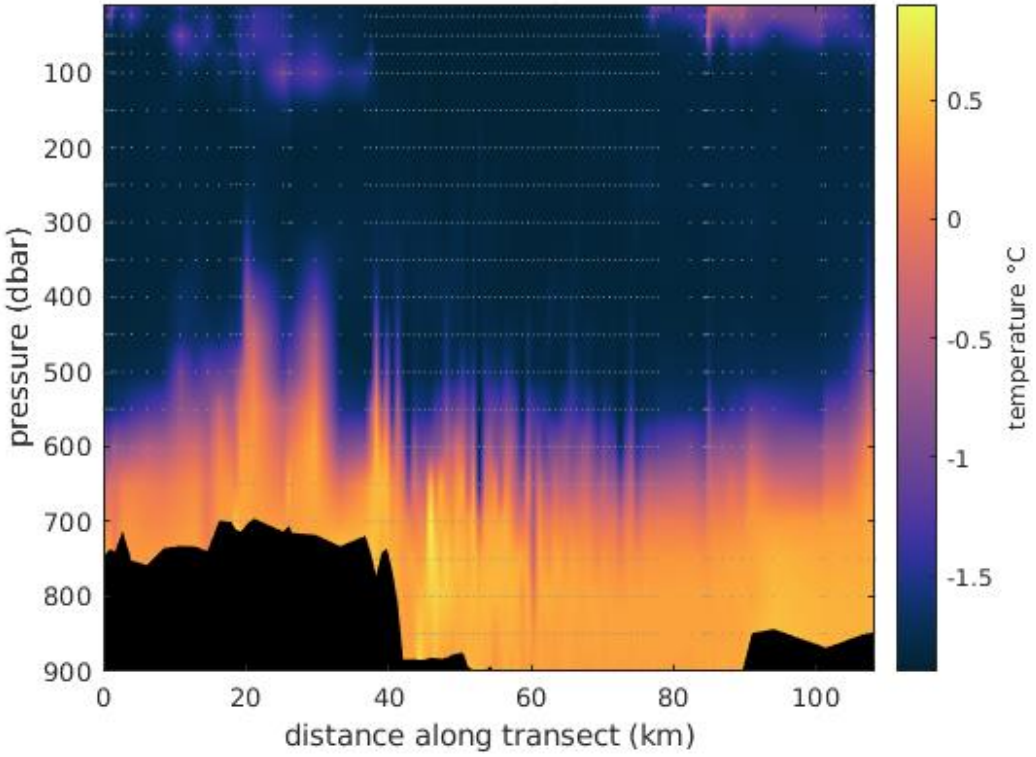


在将域扩展到超出潜在数据太远时要小心。我们只对上面例子中的数据进行了一点外推，边缘已经有点不稳定了。试着走 5 公里以上，你会看到一些古怪的东西。

示例 7: 让它好看

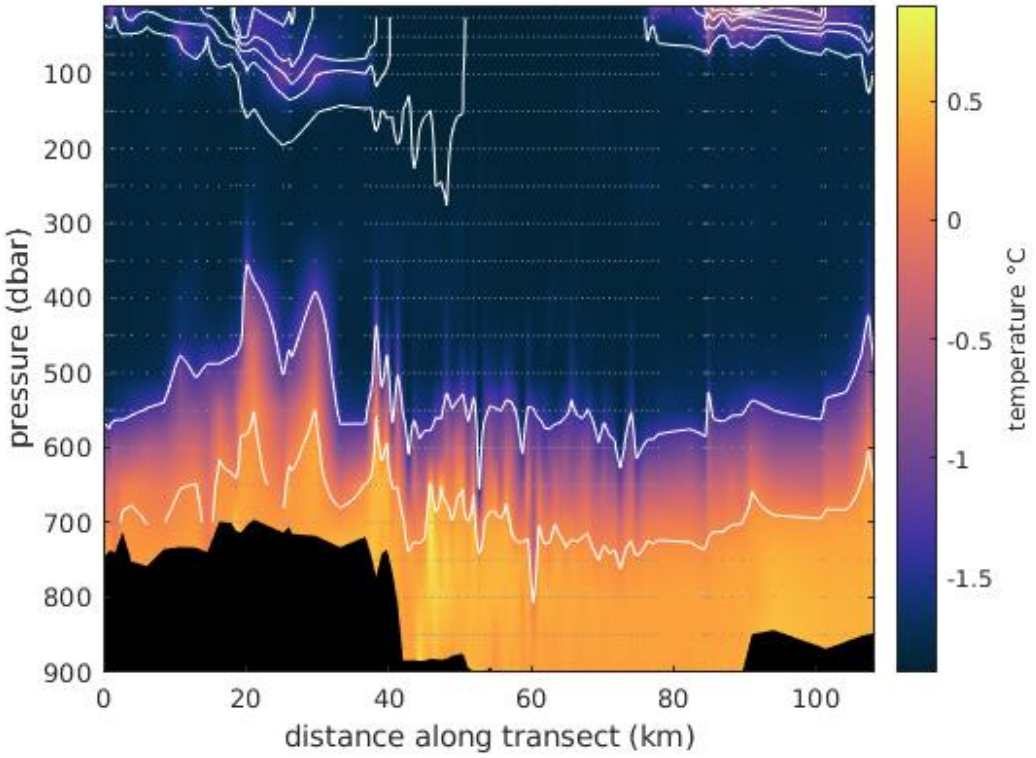
将上述选项组合到每个测量处的小灰点标记，包括海床剖面，并使用 cmocean 设置颜色图：

```
transect(x,P,T,...
   'color',rgb('gray'),... % 灰色标记
   'markersize',2,...      % 小点
   'bed',bed,...           % 海床海拔
   'extrap')

caxis([-1.9 0.9])          % 设置颜色坐标轴范围

xlabel 'distance along transect (km)'
ylabel 'pressure (dbar)'
cmocean thermal            % 设置颜色图
cb = colorbar;
ylabel(cb,'temperature \circC')
```



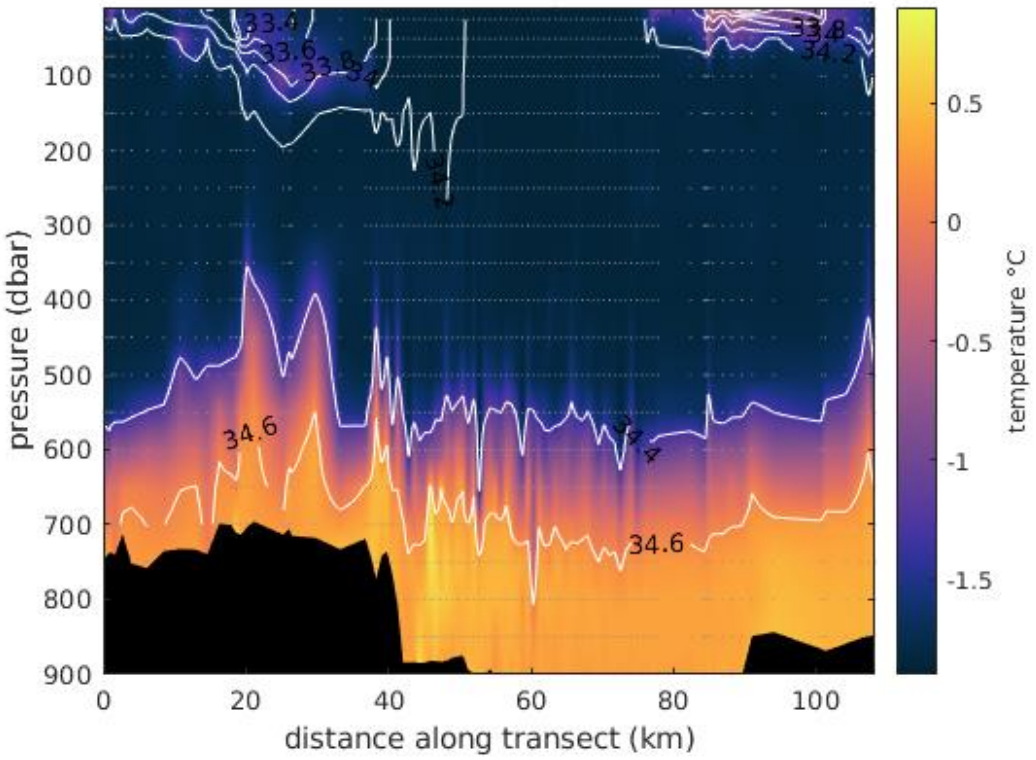
使用 transectc 函数为盐度添加白色等值线：

```
hold on
[C,h] = transectc(x,P,S,'w');
```



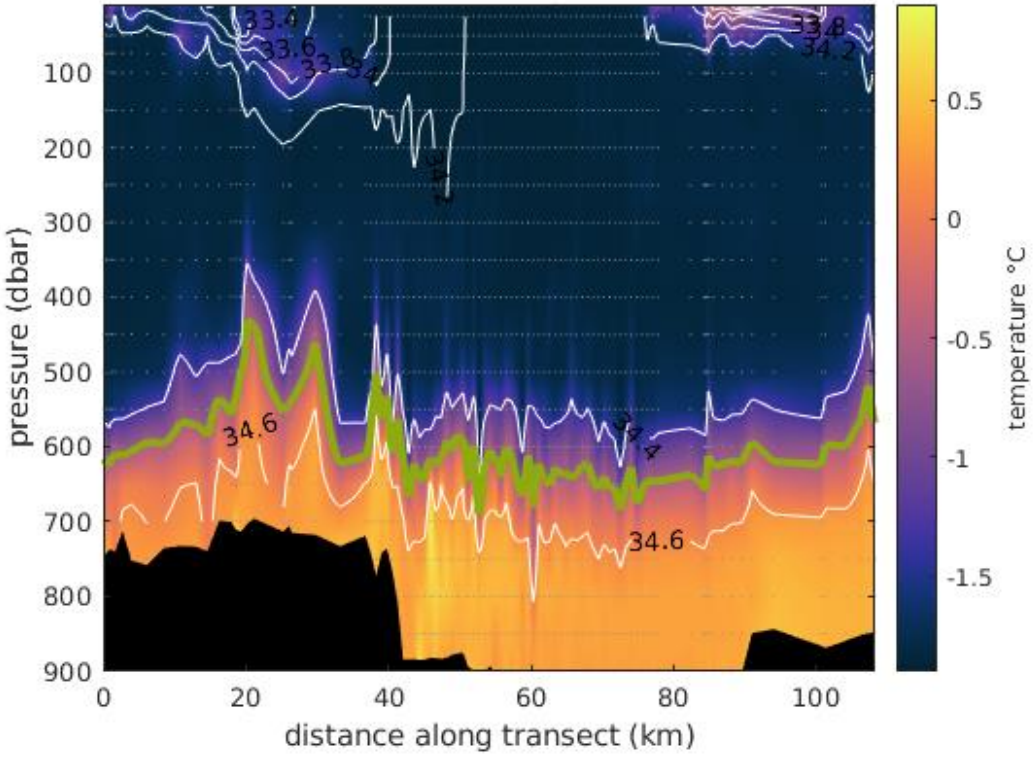
或给它们贴上标签：

```
delete(h) % 从上面删除等值线

[C,h] = transectc(x,P,S,'w','ShowText','on','LabelSpacing',1000);
```



添加您喜爱的盐度等值线，使其呈豌豆绿：

```
[C,h] = transectc(x,P,S,[34.5 34.5]);



h.Color = rgb('pea green');
h.LineWidth = 3;
```




```
clear % 在以下示例之前清除以前的数据
```

将 ODV 文本数据转换为元胞数组

在本例中，我们将使用一些由 Ocean data Viewer 保存为.txt 文件的数据。CDT 提供的数据是 2005/2005 年在南极洲海岸外航行的海洋学剖面数据的一个子集。完整的数据集在澳大利亚南极数据中心的网站上有描述，但我们将使用 CDT 附带的子集作为示例文件。

从加载示例数据开始：

```
D = importdata('BROKE_cruise_odv.txt')
```

```
D =

  struct with fields:

          data: [9220×7 double]
      textdata: {'Stn'  'Lat'  'Lon'  'Pr'  'T'  'S'  'O2'}
    colheaders: {'Stn'  'Lat'  'Lon'  'Pr'  'T'  'S'  'O2'}
```

data 字段包含七列，对应于:

1. 站号

2. 纬度
3. 经度
4. 压强
5. 温度
6. 盐度
7. 氧气

如果打开 BROKE_cruise_odv.txt 文件，您将看到第一列包含数字 103，重复 668 次。这是因为 103 号站包含 668 个温度、盐度和氧气测量值，每个测量值的压强（深度）不同。在对应于站 103 的 668 行数据下面，有对应于站 104 的 584 行，依此类推。总共有 16 个独特的站点，每个站点都有不同数量的数据行。创建一个 16x1 stn 数组，其中包含 16 个唯一的站号以及行索引，这将允许我们获得这些站的位置：

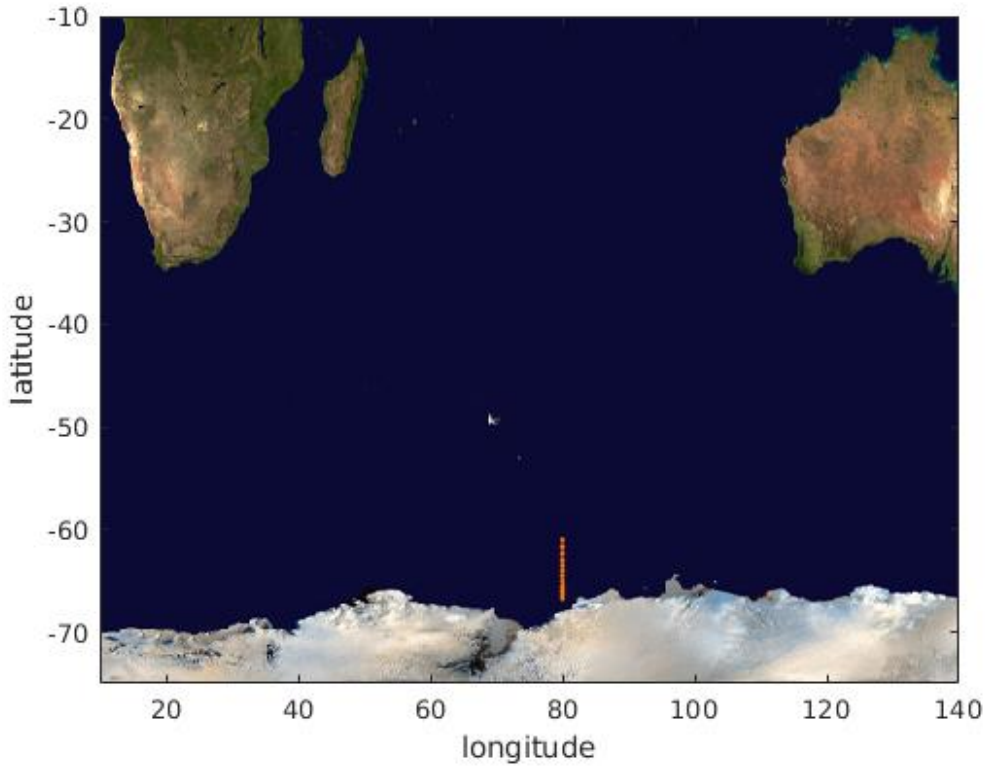
```
% 唯一的站号:
[stn,ind] = unique(D.data(:,1));
```

使用每个站点对应的索引 ind，得到每个站点的 16 个位置，如下所示，因为纬度和经度是 D.data 的第二列和第三列：

```
lat = D.data(ind,2);

lon = D.data(ind,3);
```

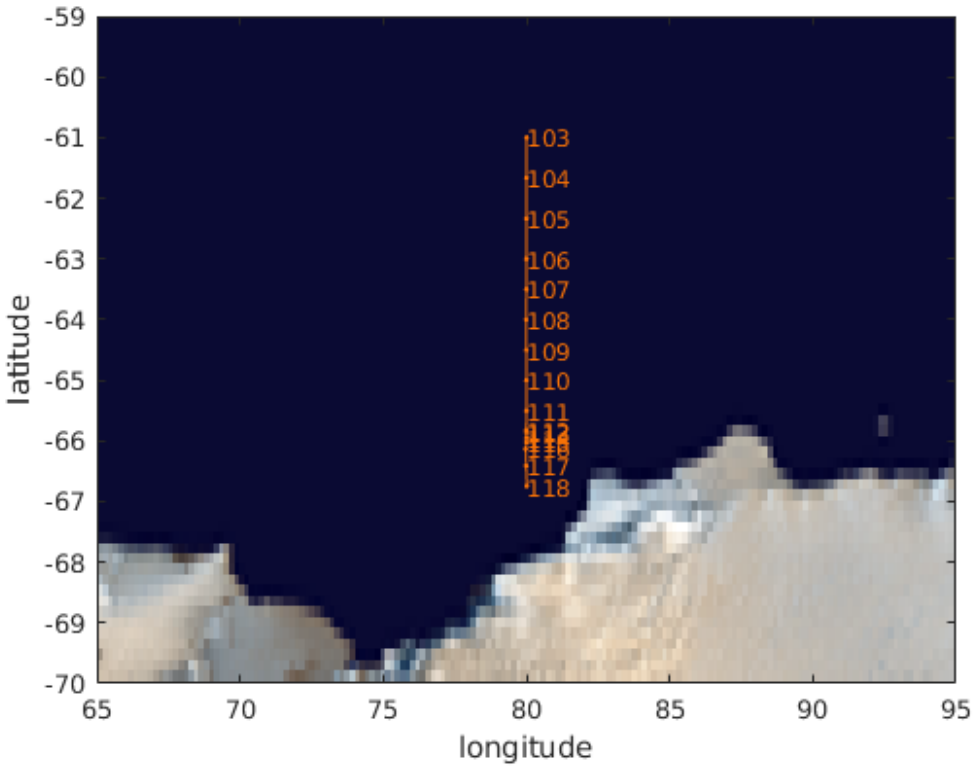
这是船的地理轨迹：

```
figure

axis([10 140 -75 -10]) % 设置坐标轴范围
earthimage
hold on
plot(lon,lat,'.-','color',rgb('orange'))
xlabel 'longitude'
ylabel 'latitude'
```



放大图可以看到每个 CTD 站的编号为 103 到 118。、

```
text(lon,lat,num2str(stn),... % 标记每个站点
   'fontsize',9,...
   'color',rgb('orange'))

axis([65 95 -70 -59]) % 设置坐标轴范围
```



要将所有测量值放入元胞数组中，请遍历每个唯一的站，找到与每个站编号相关联的所有行索引，如下所示：

```
for k = 1:16

   % 获取 stn（k）对应站的索引:
   ind = D.data(:,1)==stn(k);

   % 把这个站的压强、温度等放入元胞数组:
   Pr{k} = D.data(ind,4);
   T{k} = D.data(ind,5);
   S{k} = D.data(ind,6);
   O2{k} = D.data(ind,7);
end
```

现在 T{1}包含了所有的温度测量值、

```
stn(1)
```

```
ans =

   103
```

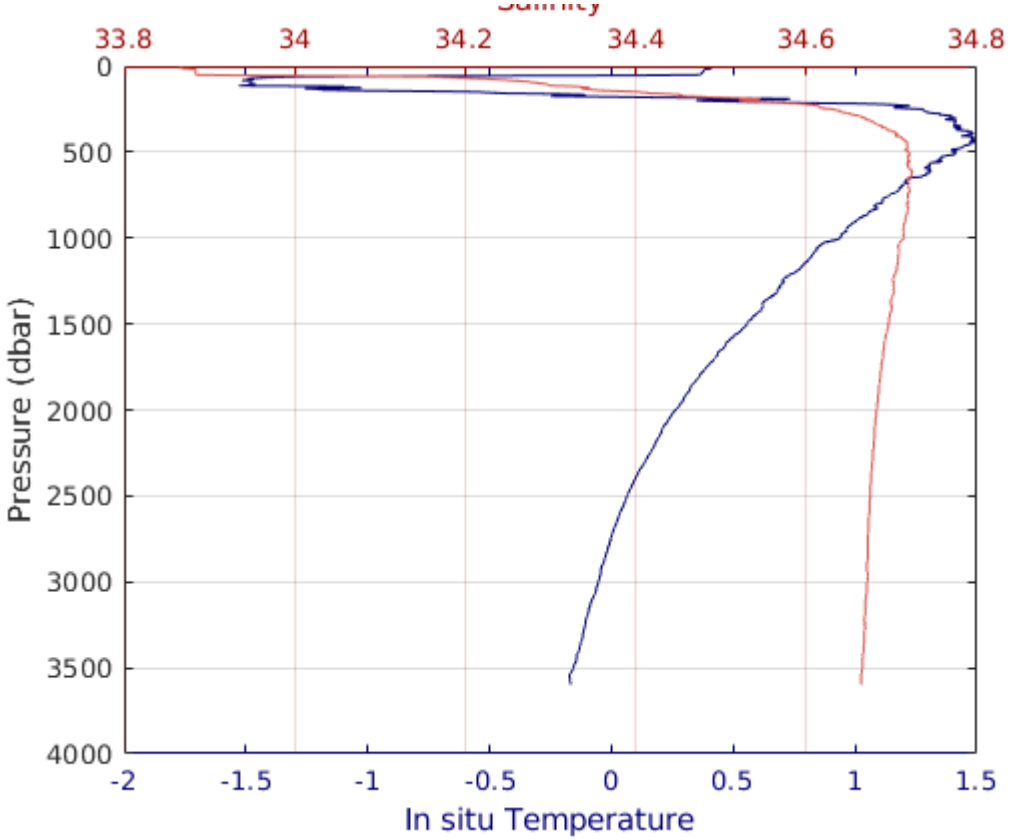
…103 号站，Pr{1}包含相应的压强。.

绘制单个特征

元胞数组允许不同大小的数据存储在一个单一的结构中，当每个 CTD 特征在不同的深度有不同数量的测量时，这是很有用的。

考虑到每个单元格都包含来自唯一 CTD 特征的数据，绘制单个特征相对简单。这里我们绘制了 110 号站(stn(8))的数据。温度绘制在第一（底部）轴上，盐度绘制在第二（顶部）轴上。

```
figure

plot(T{8},Pr{8},'color',rgb('deep blue'))
ax1 = gca;
set(ax1,'XColor',rgb('deep blue'),...
    'YDir','reverse');
xlabel('In situ Temperature')
ylabel('Pressure (dbar)')

ax1_pos = ax1.Position; % 存储第一个坐标轴的位置.
ax2 = axes('Position',ax1_pos,...
    'XAxisLocation','top',...
    'YAxisLocation','right',...
    'Color','none',...
    'XColor',rgb('deep red'),...
    'YTickLabel',[],...
    'YDir','reverse');
hold on
plot(S{8},Pr{8},'Parent',ax2,'color',rgb('pale red'))
grid on
xlabel('Salinity')
```

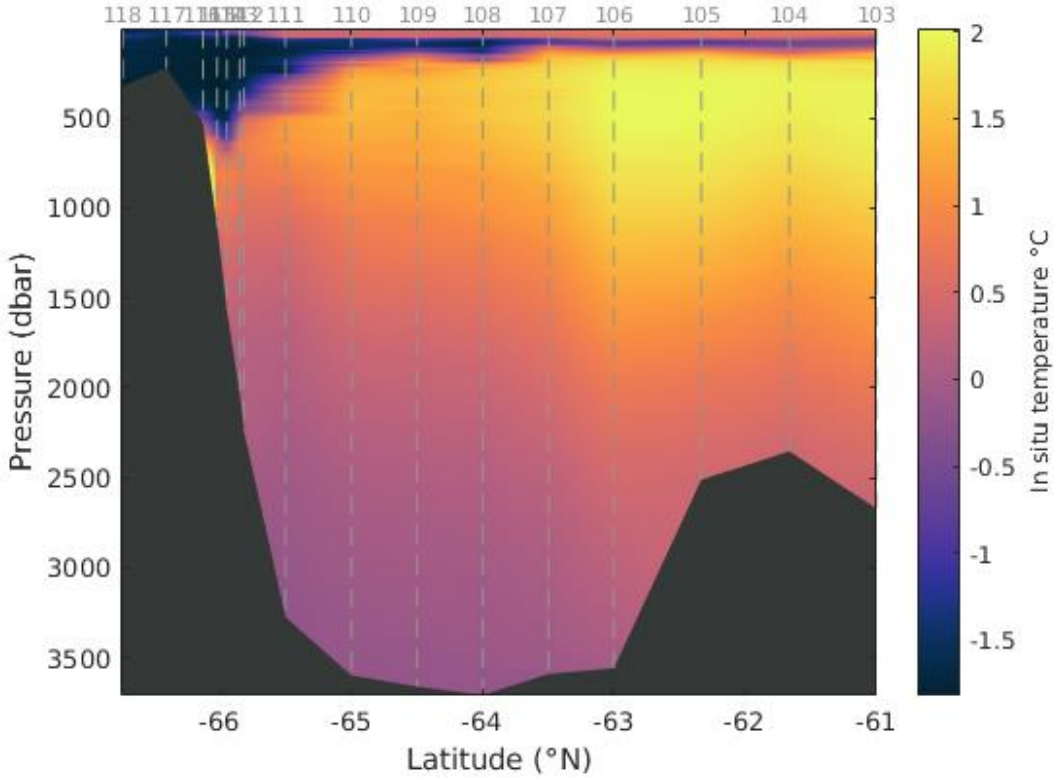


使用 transect 函数绘制多个特征

提取现在存储在元胞数组中的数据，绘制温度和盐度数据的横断面图。假设每个 CTD 特征都到达底部，我们可以使用每个站点的最大压强作为海床深度：

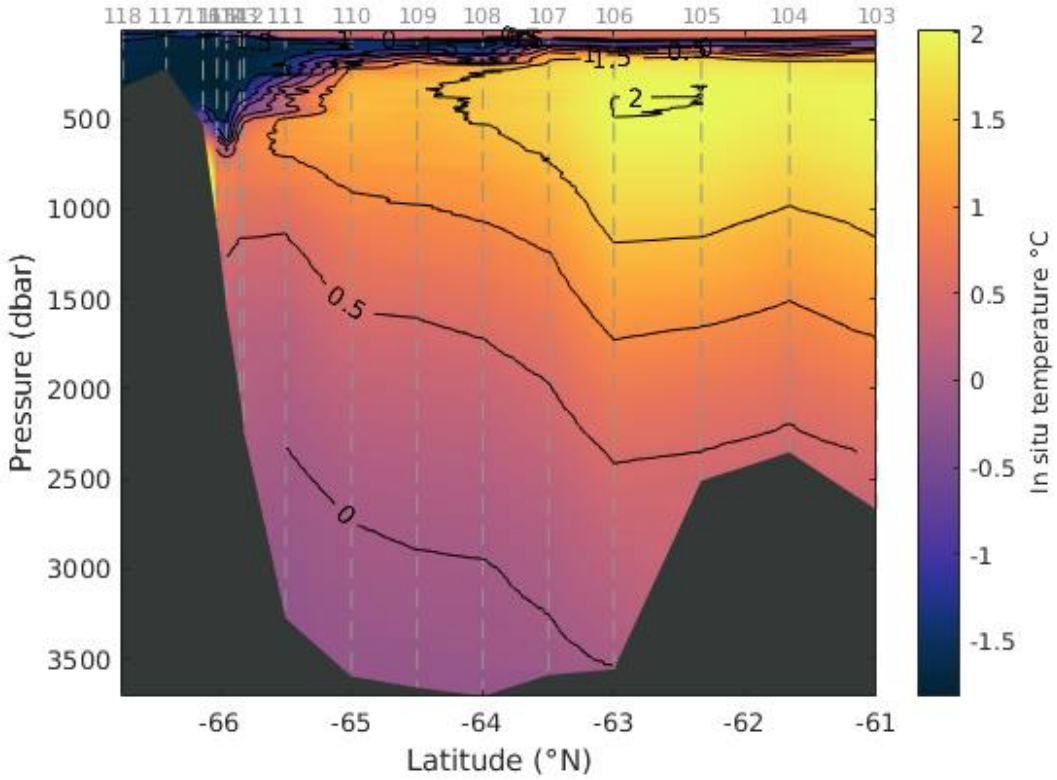
```
% 假设最大压强在海床深度:
maxPress = cellfun(@max,Pr);

figure
transect(lat,Pr,T,...
    'color',rgb('gray'),...
    'marker','none',...
    'linestyle','--',...
    'bed',maxPress,...
    'extrap',...
    'bedcolor',rgb('charcoal'));

caxis([-1.82 2.02])

cmocean('thermal')
cb = colorbar;
ylabel(cb,'In situ temperature \circC')
xlabel 'Latitude (\circN)'
```

```
ylabel 'Pressure (dbar)'

% 标注站点:
text(lat,zeros(size(lat)),num2str(stn),...
   'vert','bottom',... % 从它们的底部开始贴文字标签
   'horiz','center',...% 标签居中
   'fontsize',8,...
   'color',rgb('gray'))
```



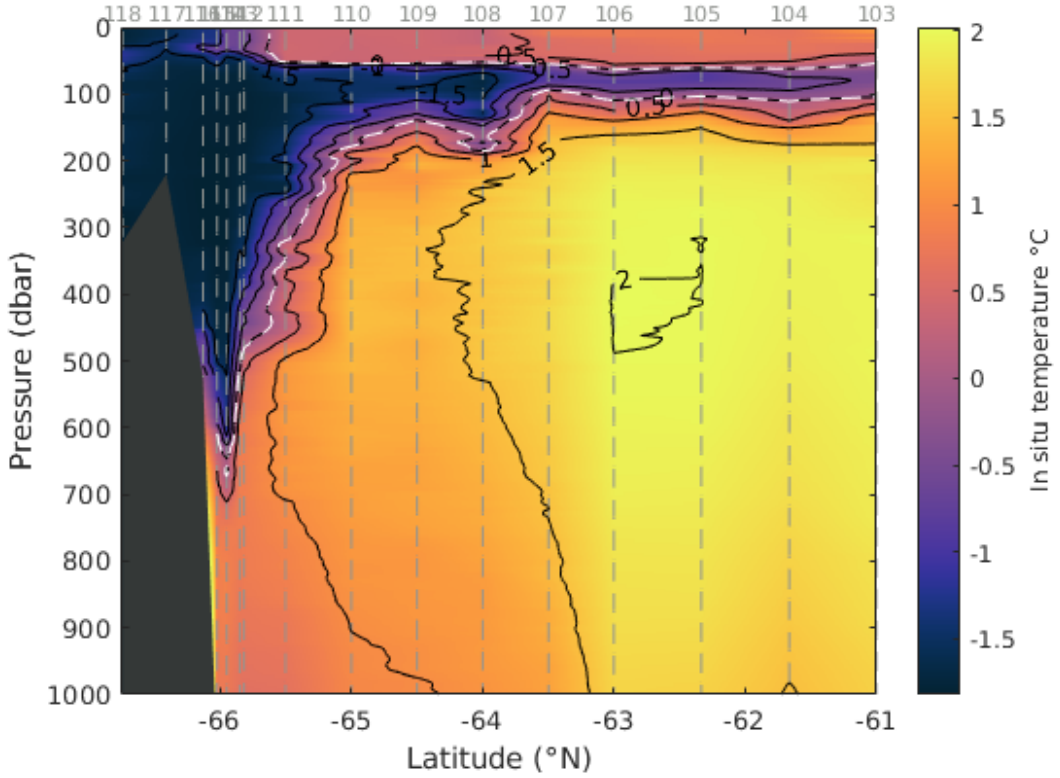
加上温度的等值线图:

```
hold on
transectc(lat,Pr,T,'k','ShowText','on','LabelSpacing',1000);
```



突出显示 0 度等温线并放大顶部 1000 dbar:

```
transectc(lat,Pr,T,[0 0],'w--');

ylim([0 1000])
```



作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分.这个函数和支持文档是由塔斯马尼亚大学的 David E. Gwyther 和德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

transectc 文档

transectc 根据在不同地点和/或时间收集的 CTD 剖面生成等值线图海洋学数据。
另请参见: transect 帮助将数据转化为元胞数组。

语法

```
transectc(x,d,v)
transectc(...,levels)
transectc(...,LineSpec)
transectc(...,'Name',Value)
transectc(...,'extrap')
transectc(...,'interp','InterpMethod)
transectc(...,'di',di)
transectc(...,'xi',xi)
[C,h] = transectc(...)
```

说明

transectc(x,d,v) 在水平位置（或时间）x 和深度 d 处创建变量 v 的等值线图。x 必须是一个数值数组，表示每个 CTD 配置文件的位置或时间。输入 d 和 v 必须是包含每个 CTD 位置的深度和测量值的元胞数组。
transectc(...,levels) 指定要绘制为等高线的 v 值。如果 levels 是一个标量值，那么 transectc 将绘制该数量的等高线。如果 levels 是一个数组，则为 levels 中的每个值绘制一条等高线。若要以一个值（k）绘制等高线，请指定级别作为两元素行向量[k k]。
transectc(...,LineSpec) 指定等值线的样式和颜色。
transectc(...,'Name',Value) 使用一个或多个名称-值对参数为等值线图指定其他选项。在所有其他输入参数之后指定选项。有关属性列表，请参见等值线属性。
transectc(...,'extrap') 打开选项以外插超出范围的数据。默认情况下，transatectc 不进行外插，但可以使用该选项来填补空白，例如最低点和海底之间的空白。
transectc(...,'interp','InterpMethod) 指定用于等值线的内插方法。默认值为 s 'linear'.
transectc(...,'di',di) 指定插值深度 di。默认情况下，transectc 通过在最浅和最深测量之间内插到 1000 个等距深度来创建栅格。
transectc(...,'xi',xi) 指定插值位置（或时间）xi。默认情况下，transectc 内插到 2000 个等间距点 xi ，在 x 的最小值和最大值之间。
[C,h] = transectc(...) 返回绘制对象的等值线矩阵 C 和句柄 h 。r

方法

该函数使用 transect 函数文档中描述的方法对间距不均匀的数据进行网格化，然后使用 Matlab 的内置 contour 函数绘制网格化数据。

示例

下面的示例使用一些 Argo 数据，您可以这样加载这些数据：

```
load example_ctd

whos % 展示每个变量的名称和尺寸
```

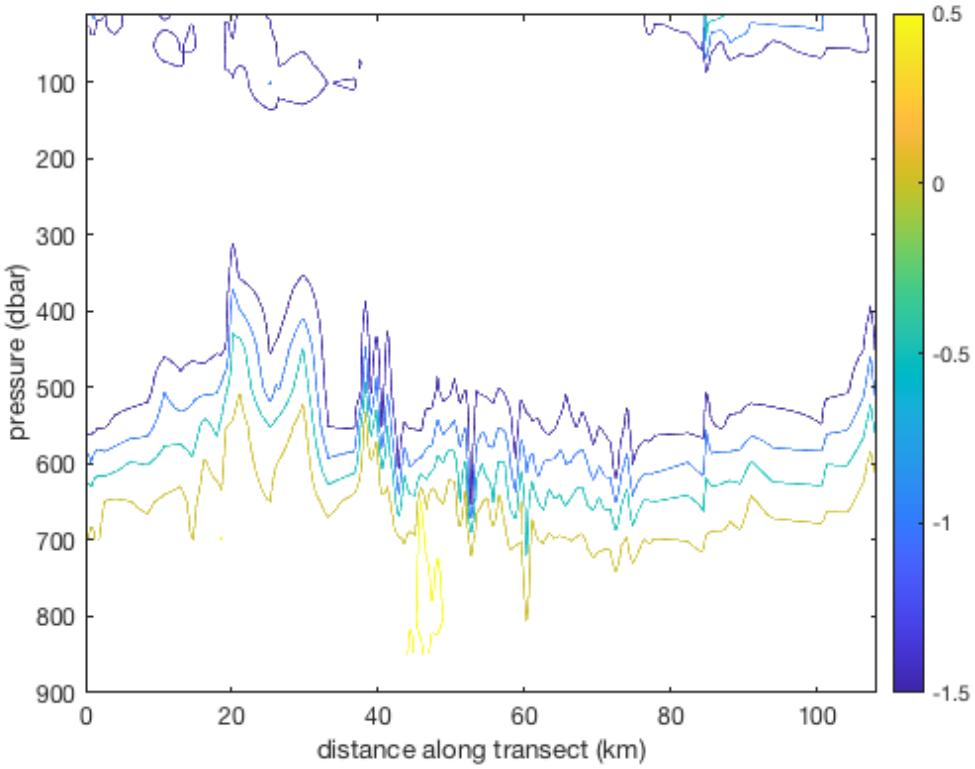
```
  Name      Size            Bytes  Class     Attributes

  P         1x95            25152  cell
  S         1x95            25152  cell
  T         1x95            25152  cell
  bed       1x95              760  double
  x         1x95              760  double
```

Argo 数据包含 95 个浮标漂移时的剖面。每个剖面都是在 x 的某个位置采集的，相应的海床深度由 bed 给出，温度 T 和盐度 S 在相应压强 P 下的元胞数组中。有关将数据转换为元胞数组的帮助，请参见 transect 。

示例 1:简单形式

以下是在 x 和 P 位置采集的温度数据 T 的简单样带：

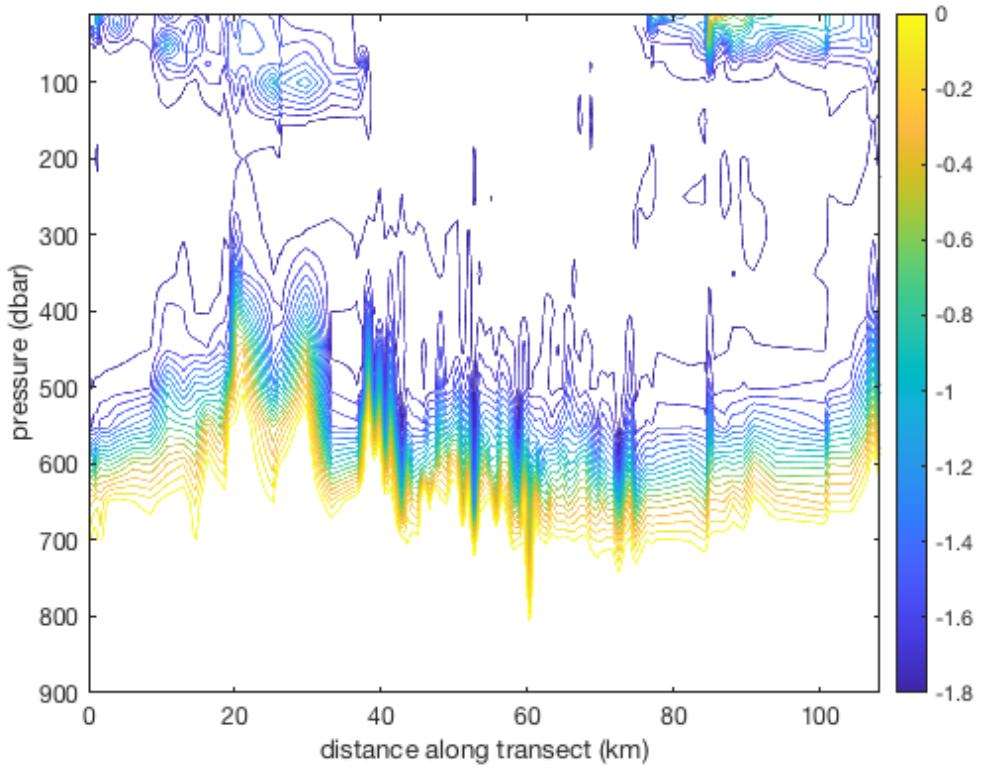
```
transectc(x,P,T)



colorbar

xlabel 'distance along transect (km)'
ylabel 'pressure (dbar)'
```



示例 2: 使用不同的线样式指定等值线层次 S

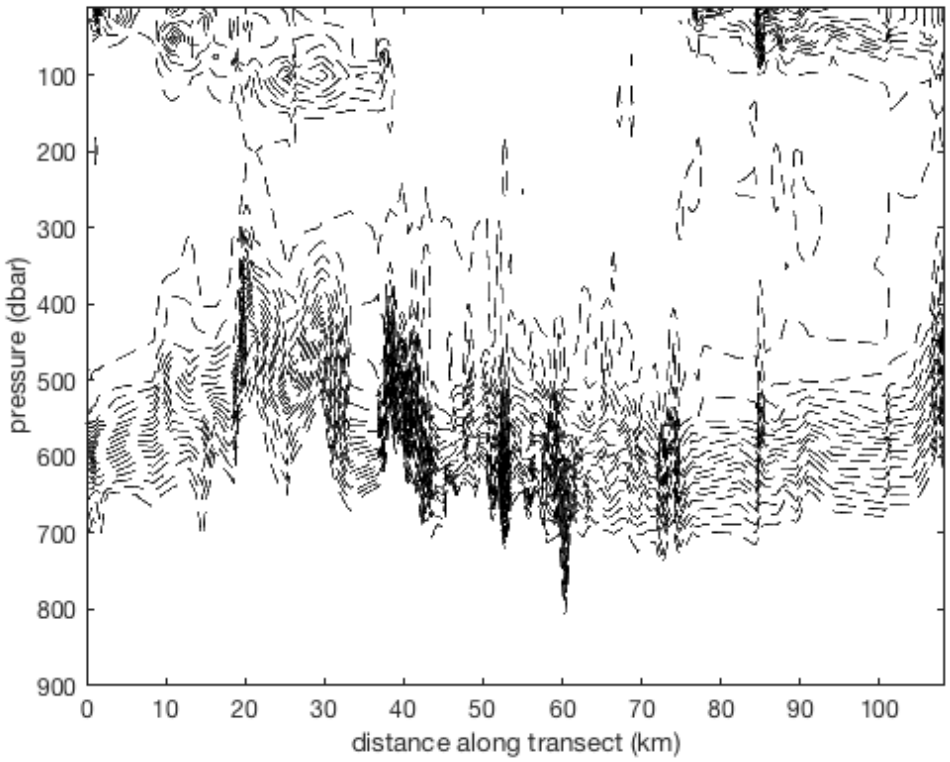
如上所述，但现在只绘制负值，每 0.1 度绘制一行：

```
vals = -2:0.1:0;



figure

transectc(x,P,T,vals)



colorbar

xlabel 'distance along transect (km)'
ylabel 'pressure (dbar)'
```



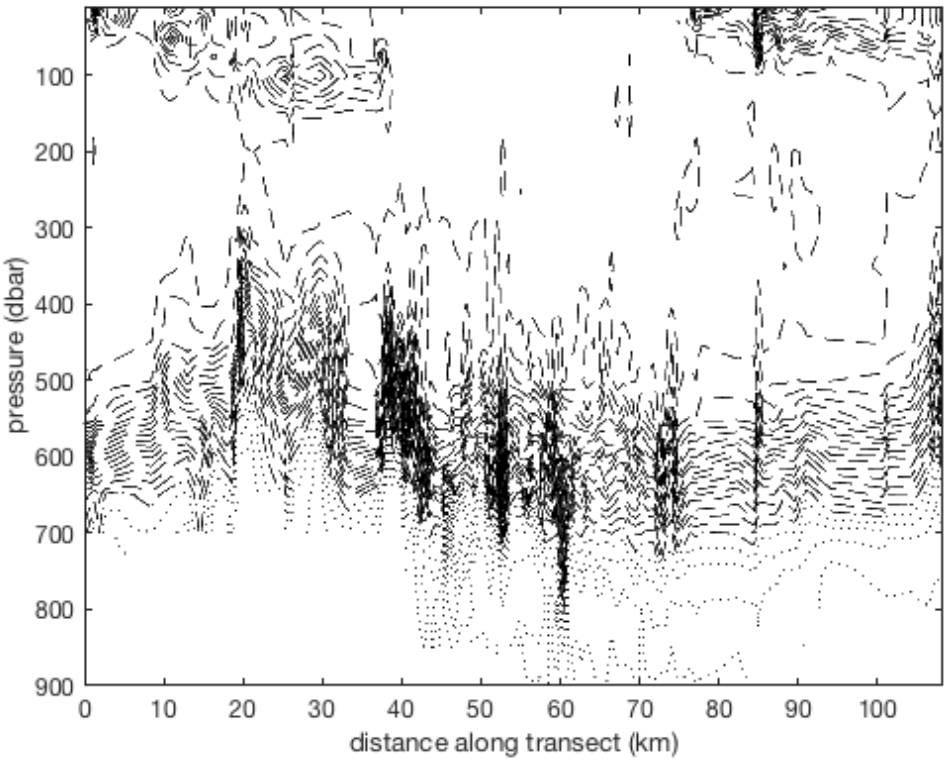
现在只绘制负值并使其成为黑色虚线：

```
figure



transectc(x,P,T,vals,'k--')

xlabel 'distance along transect (km)'
ylabel 'pressure (dbar)'
```



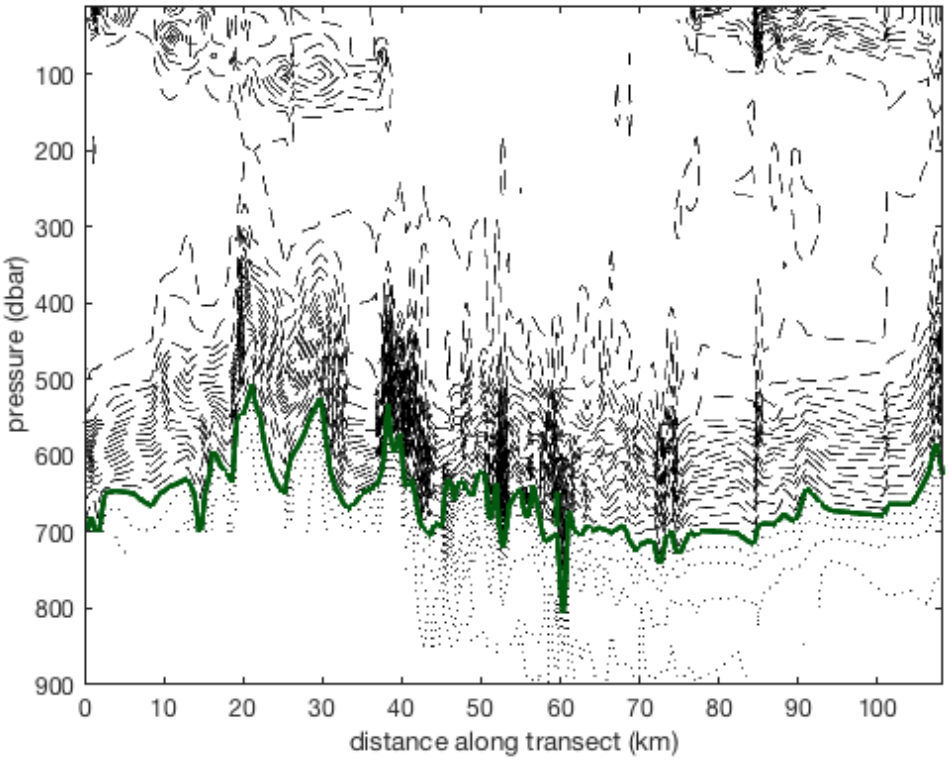
正值是点线:

```
hold on
transectc(x,P,T,0:0.1:2,'k:')
```



0 的等值线更粗:

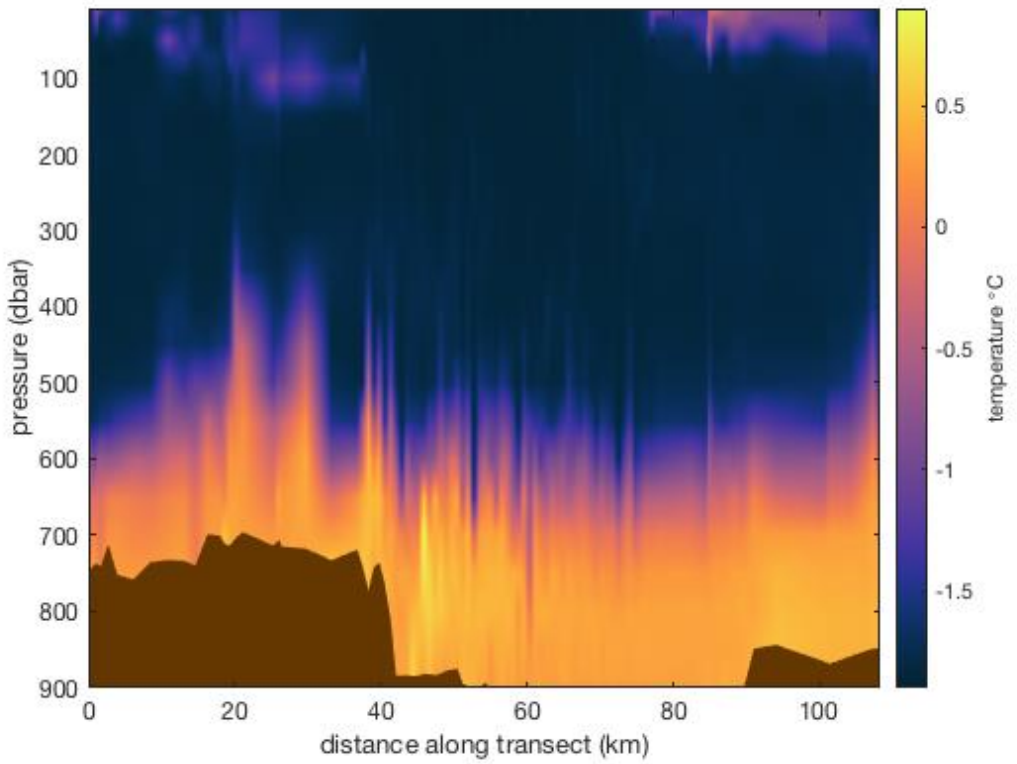
```
transectc(x,P,T,[0 0],'color',rgb('deep green'),'linewidth',3)
```



示例 3: 与 transect 联合使用

等值线图本身很难解释。当与 transect 函数结合使用时，它可以说明特性的重要变化。有关选项的说明，请参见 transect 文档。

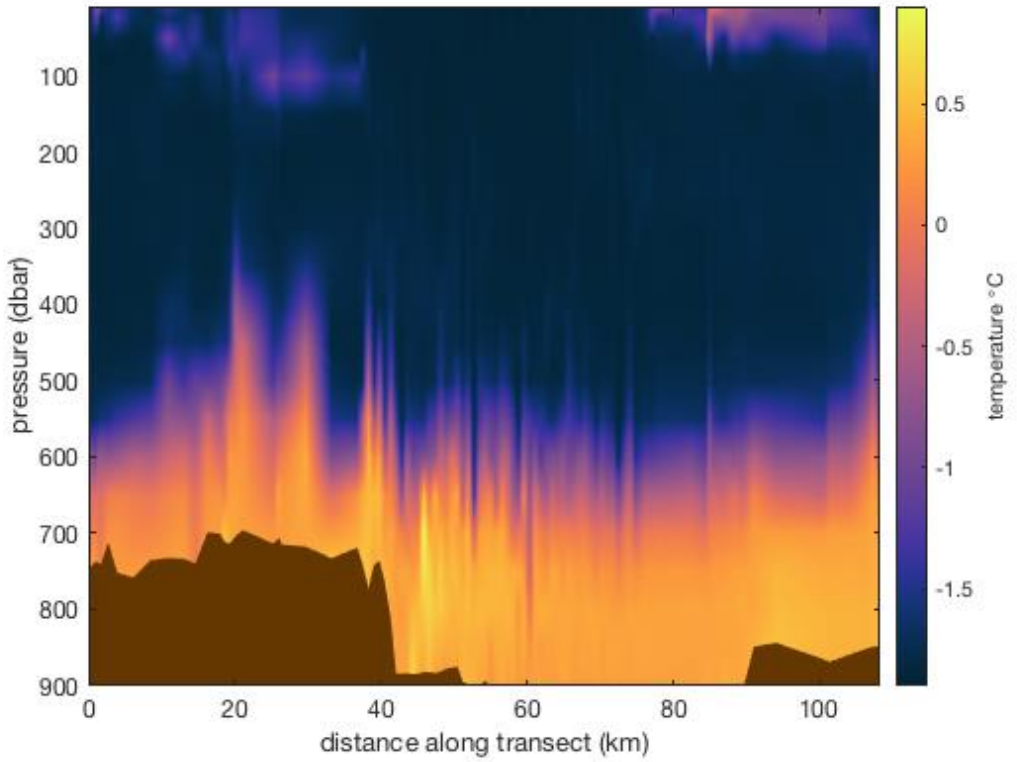
```
figure

transect(x,P,T,...
   'bed',bed,...          % 加上基岩数据
   'bed',bed,'bedcolor',rgb('brown'),... % 基岩棕色
   'marker','none',...       % 关掉标注点
   'extrap')

caxis([-1.9 0.9])          % 设置颜色轴范围

xlabel 'distance along transect (km)'
ylabel 'pressure (dbar)'
cmocean thermal            % 设置颜色图
cb = colorbar;
ylabel(cb,'temperature \circC')
```



现在添加等值线，确保抑制打印到屏幕不会溢出：

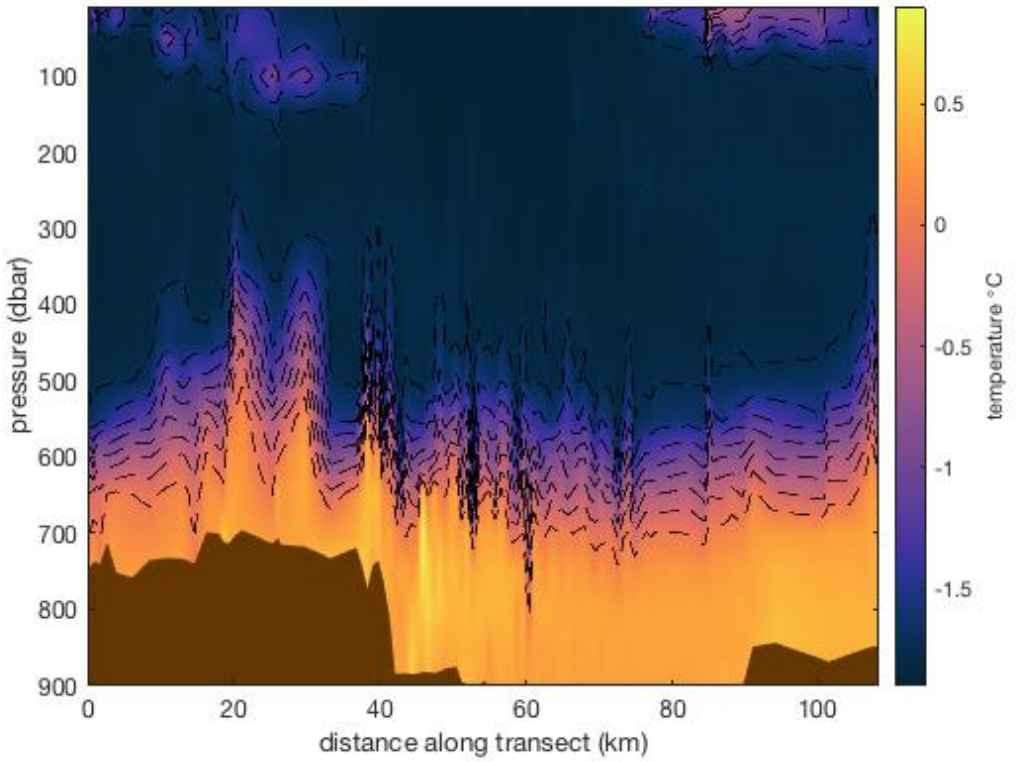
```
hold on
```

```
[C,h] = transectc(x,P,T,-2:0.1:0);
```



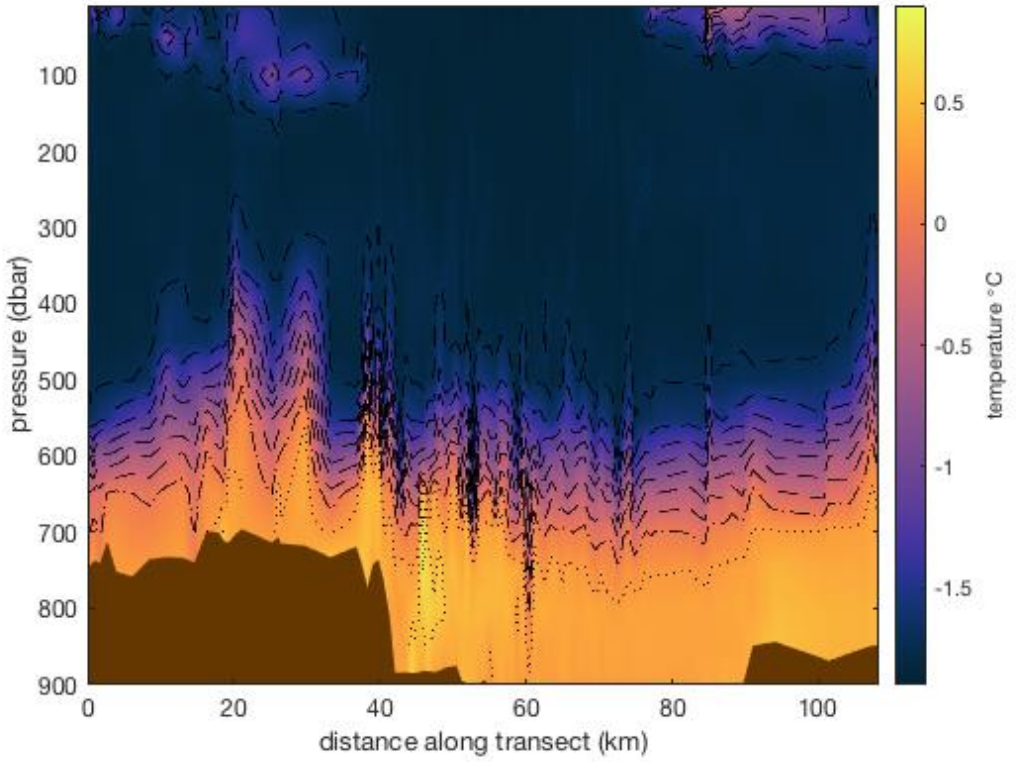
请注意，他们是不可见的，因为他们采取了相同的颜色图作为 transect 图如下。为了使它们可见，我们需要给它们一个恒定的颜色。将正等值线添加为点线，负等值线为虚线，零温度等值线为粗白线：

```
delete(h) % 删除上述等值线

transectc(x,P,T,-2:0.25:0,'k--')
```



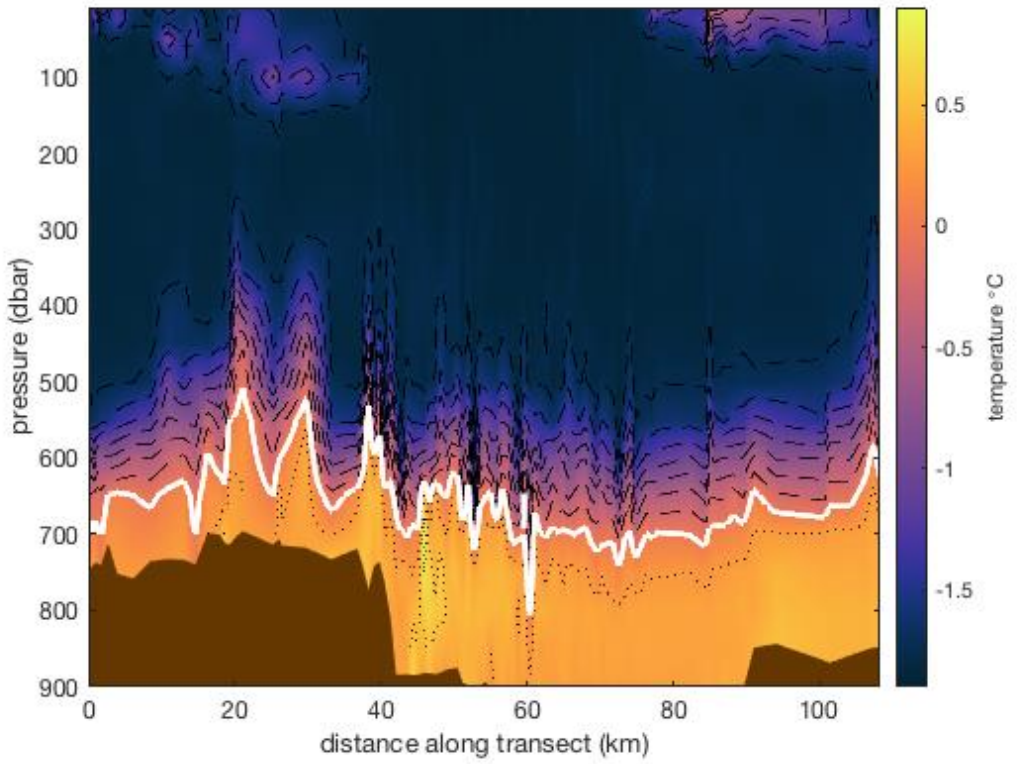
正值点线:

```
transectc(x,P,T,0:0.25:2,'k:')
```



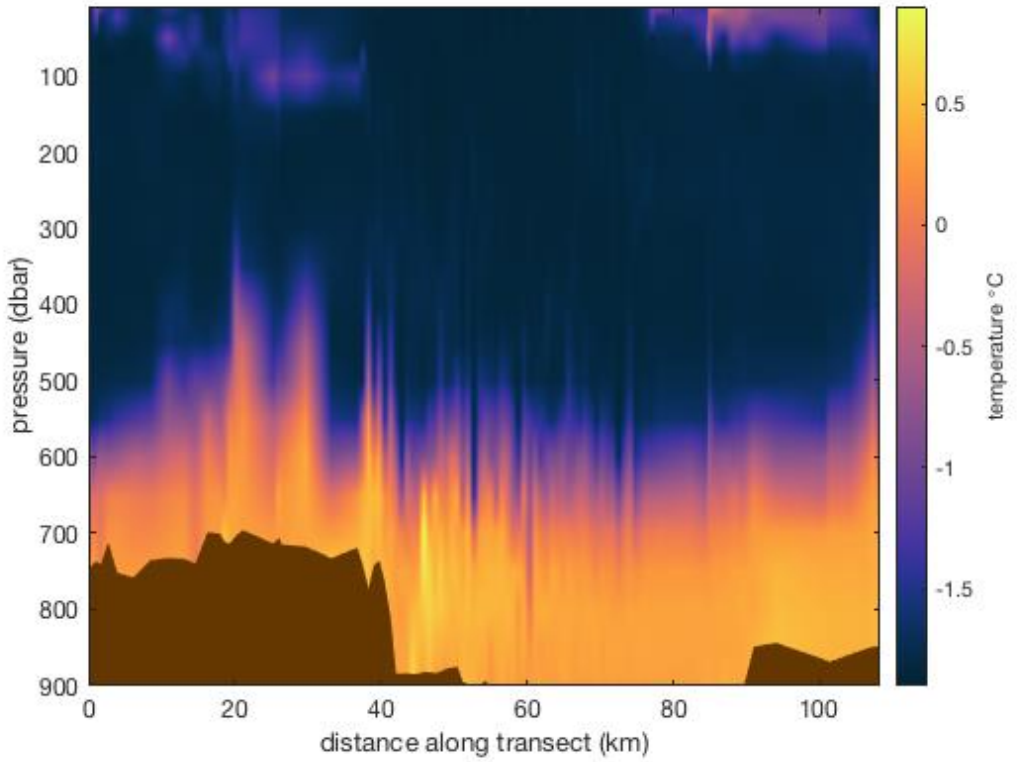
零等值线粗一点:

```
transectc(x,P,T,[0 0],'w','linewidth',3)
```



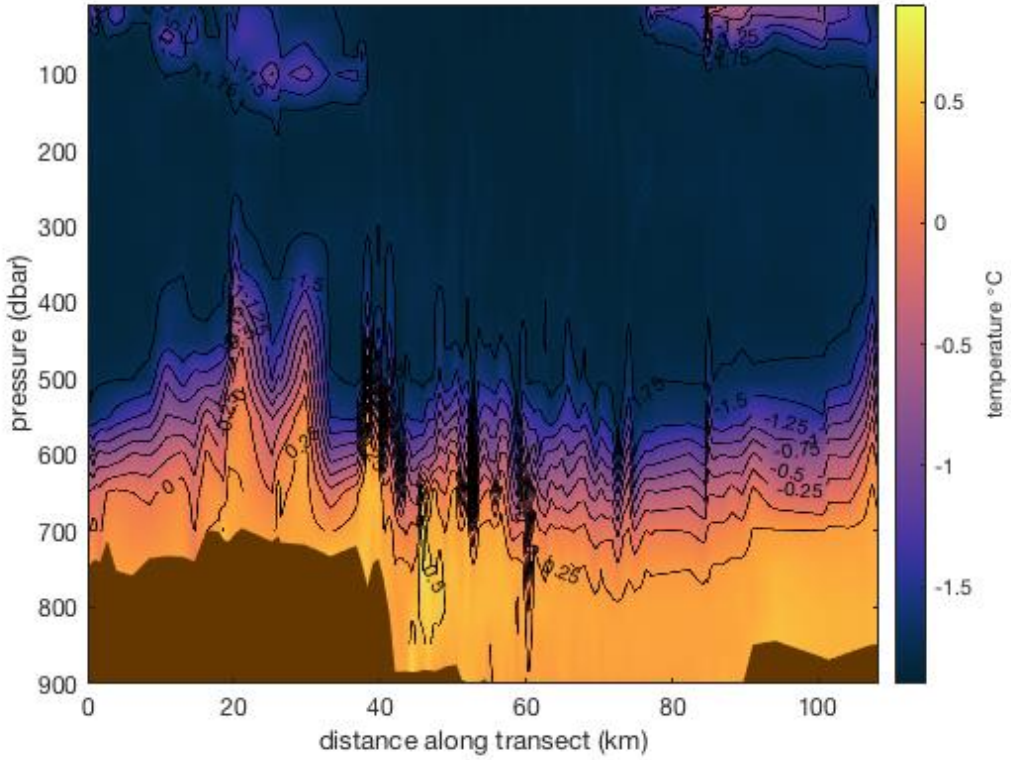
再次执行相同的操作，但这次将向每个等值线添加文本标签。

```
figure

transect(x,P,T,...
   'bed',bed,...         % 增加基岩数据
   'bed',bed,'bedcolor',rgb('brown'),... % 基岩棕色
   'marker','none',...       % 关掉标记
   'extrap')

caxis([-1.9 0.9])          % 设置颜色轴范围

xlabel 'distance along transect (km)'
ylabel 'pressure (dbar)'
cb = colorbar;
ylabel(cb,'temperature \circC')
cmocean thermal % 设置颜色图
```



现在通过设置'ShowText'选项添加文本标签，标签之间具有适当的间距：

```
hold on
transectc(x,P,T,-2:.25:2,'k-',...
    'ShowText','on',...
    'LabelSpacing',1000);
```



作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分.这个函数和支持文档是由塔斯马尼亚大学的 David E. Gwyther 和德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

地球物理属性

- earth_radius 给出了地球的标准半径或与纬度有关的半径。
- air_pressure 根据美国标准大气的气压场计算压力。
- air_density 计算美国标准大气的气压场密度。
- sun_angle 给出了地球上任何地点任何时间的太阳方位角和仰角。
- solar_radiation 计算了地球大气层顶部每天接收到的天文辐射总量。
- daily_insolation 计算的是在过去 500 万年中，任何一点日均日射能量作为日期和纬度的函数。
- topo_interp 插值 ETOPO5 中任何地理位置的高程。
- island 决定了地理位置是与陆地相对应还是与水相对应。
- dist2coast 确定从任何地理位置到最近海岸线的距离。

earth_radius 文档

earth_radius 给出了地球的半径。

语法

```
r = earth_radius
r = earth_radius(lat)
r = earth_radius(...,'km')
```

说明

r = earth_radius 返回 6371000,以米为单位的标准地球半径。
r = earth_radius(lat) 给出以纬度为函数的地球半径。
r = earth_radius(...,'km') 以公里为单位返回值。

示例 1: 标准半径

以米为单位给出标准地球半径:

```
earth_radius
```

```
ans =

     6371000
```

...或者以公里为单位:

```
earth_radius('km')
```

```
ans =

        6371
```

示例 2: 依赖纬度的地球半径

地球与其说是球体，不如说是椭球体，这意味着它的半径取决于纬度。按百分比计算，赤道的标准地球半径有多远？

```
100*(earth_radius-earth_radius(0))/earth_radius(0)
```

```
ans =

   -0.1119
```

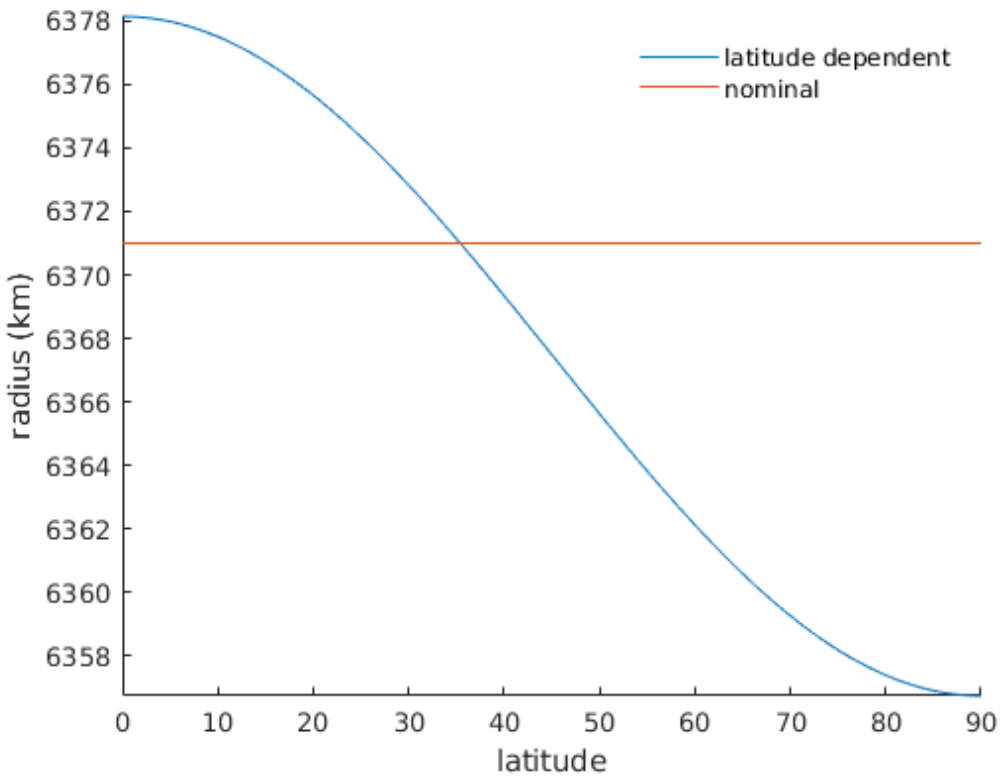
也就是说，地球的标准半径 6371 公里，比赤道的真实半径小约十分之一。我们应该期待在北极出现同样的错误吗？

```
100*(earth_radius-earth_radius(90))/earth_radius(90)
```

```
ans =

    0.2241
```

事实上，标准半径比两极的真实半径大 *0.2%*。这是地球半径与纬度的函数关系：

```
lat = 0:90;

r = earth_radius(lat,'km');

plot(lat,r)
axis tight
box off

hline(earth_radius('km')) % 标准地球半径
xlabel latitude
ylabel 'radius (km)'

legend('latitude dependent','nominal')
legend boxoff
```



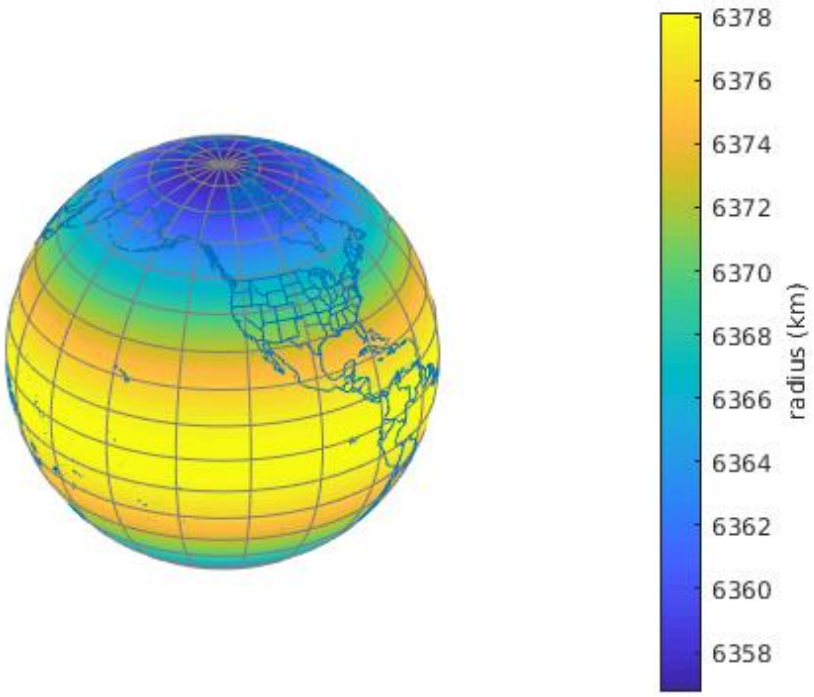
示例 3:一个网格

使用 cdtgrid 生成全局网格，使用 earth_radius 获取网格上每个点的地球半径：

```
[lat,lon] = cdtgrid;

r = earth_radius(lat,'km');
```

用 globepcolor 在地球仪上绘制它们：

```
figure

globepcolor(lat,lon,r)

globeborders   % 绘制政区边界 plots political boundaries
globegraticule % 绘制网格线
axis tight     % 移除空白
cb = colorbar;
ylabel(cb,'radius (km)')
```



作者简介

这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Natalie S. Wolfenbarger 和 Chad A. Greene 于 2019 年写的。

air_pressure 文档

air_pressure 根据美国标准大气的气压场计算气压。
另请参见 air_density.

语法

P = air_pressure(h)

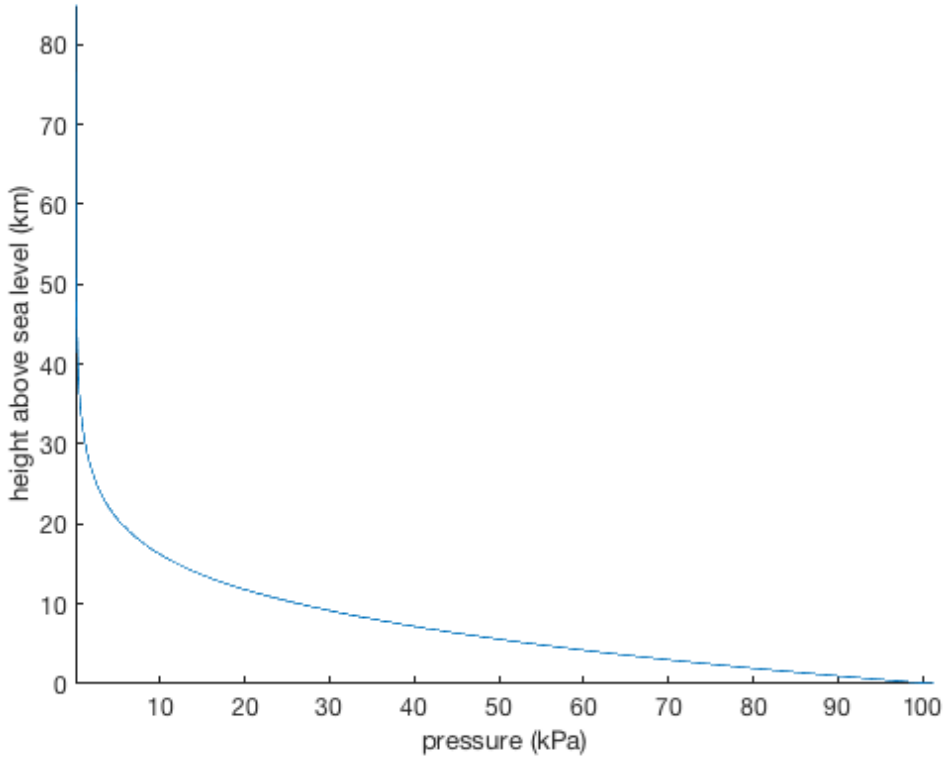
说明

P = air_pressure(h) 返回压力 P，单位为帕斯卡，对应于海拔高度 h，单位为几何米。

示例 1: 气压廓线

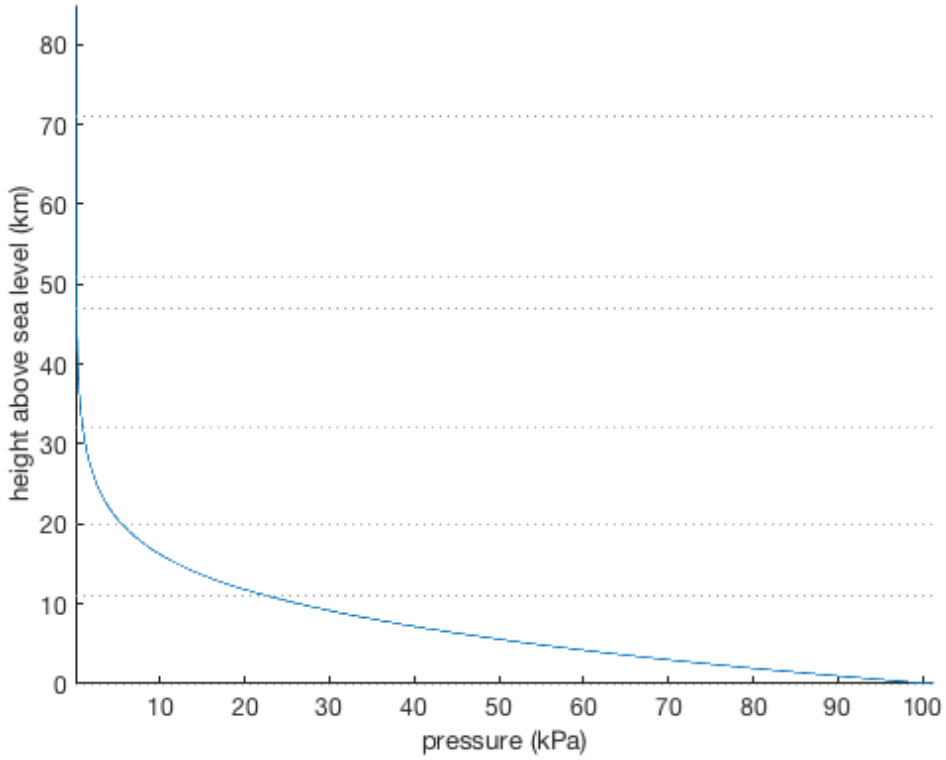
美国标准大气公式适用于海平面 86 公里处，所以让我们看看从海平面到 85 公里处的气压：

```
h = 0:85e3;



p = air_pressure(h);



figure

plot(p/1000,h/1000)

axis tight
box off
ylabel 'height above sea level (km)'
xlabel 'pressure (kPa)'
```



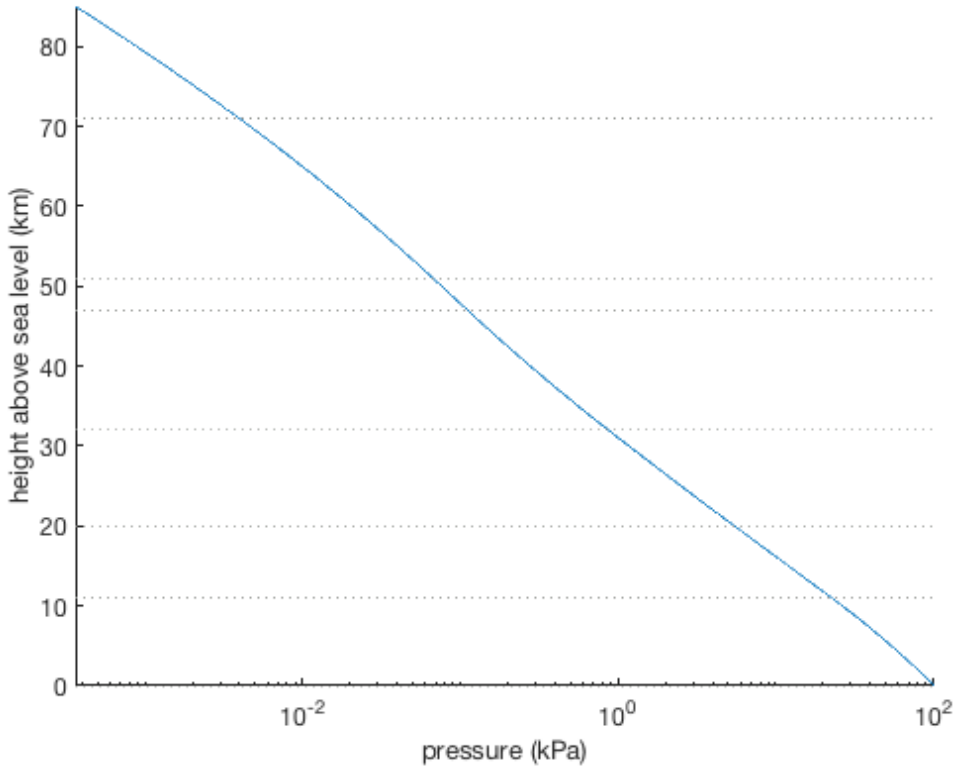
该公式使用了一个分段函数，根据不同大气层的不同常数计算出一个方程。这些地层的基于海平面以上 0、11、20、32、47、51 和 71 公里处。对于背景，我们可以使用 hline 将这些层基准绘制为水平线：

```
LayerBases = [0 11 20 32 47 51 71];



hline(LayerBases,':','color',rgb('gray'))
```



提示:对于某些应用程序,在对数刻度上显示 x 轴可能会有所帮助。要做到这一点,我们可以用 semilogx 来绘制数据，而不是使用上面的 plot 函数，或者我们可以将 x 标度改为 log，如下所示：

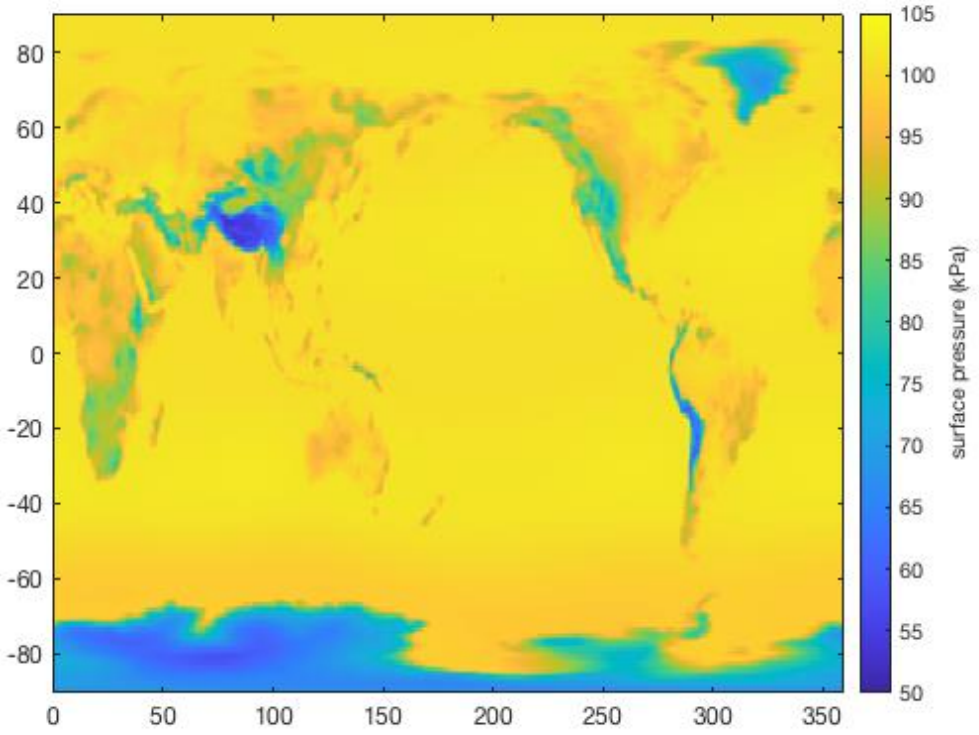
```
set(gca,'xscale','log')
```



示例 2: 与再分析地面气压的比较

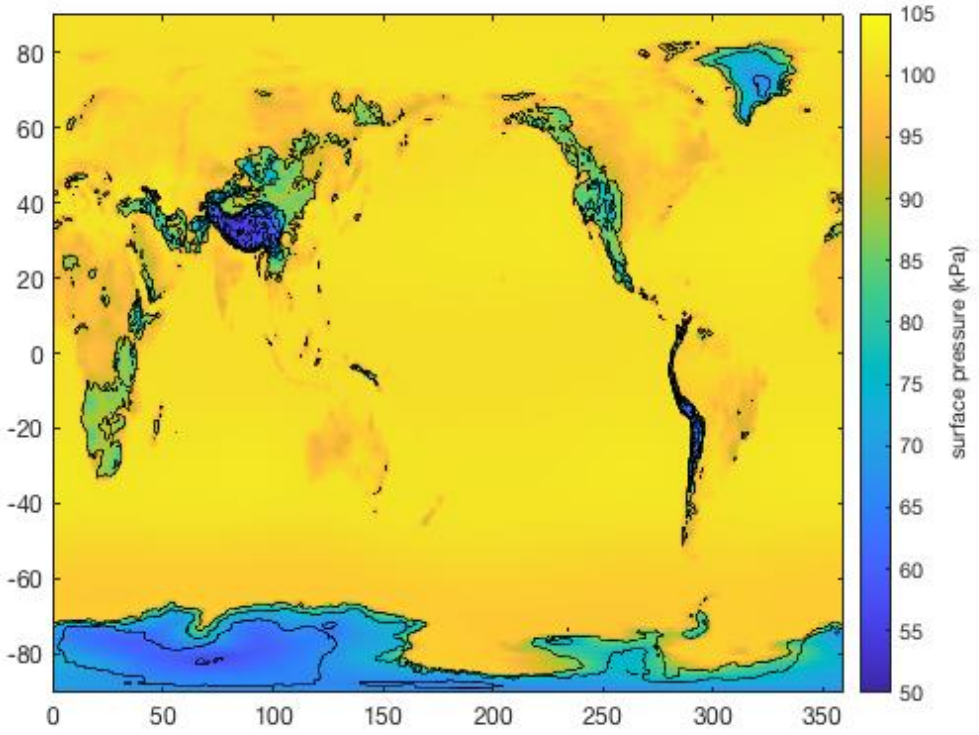
考虑这个全球网格化的地面气压数据，我们将只看 2017 年以来的地面气压：

```
lat = ncread('ERA_Interim_2017.nc','latitude');
lon = ncread('ERA_Interim_2017.nc','longitude');
t = ncread('ERA_Interim_2017.nc','time');
sp = ncread('ERA_Interim_2017.nc','sp');

% 计算平均地面气压:
spm = mean(sp,3);

% 绘制地面气压:
figure
imagescn(lon,lat,spm'/1000)

cb = colorbar;
ylabel(cb,'surface pressure (kPa)')
caxis([50 105])
```



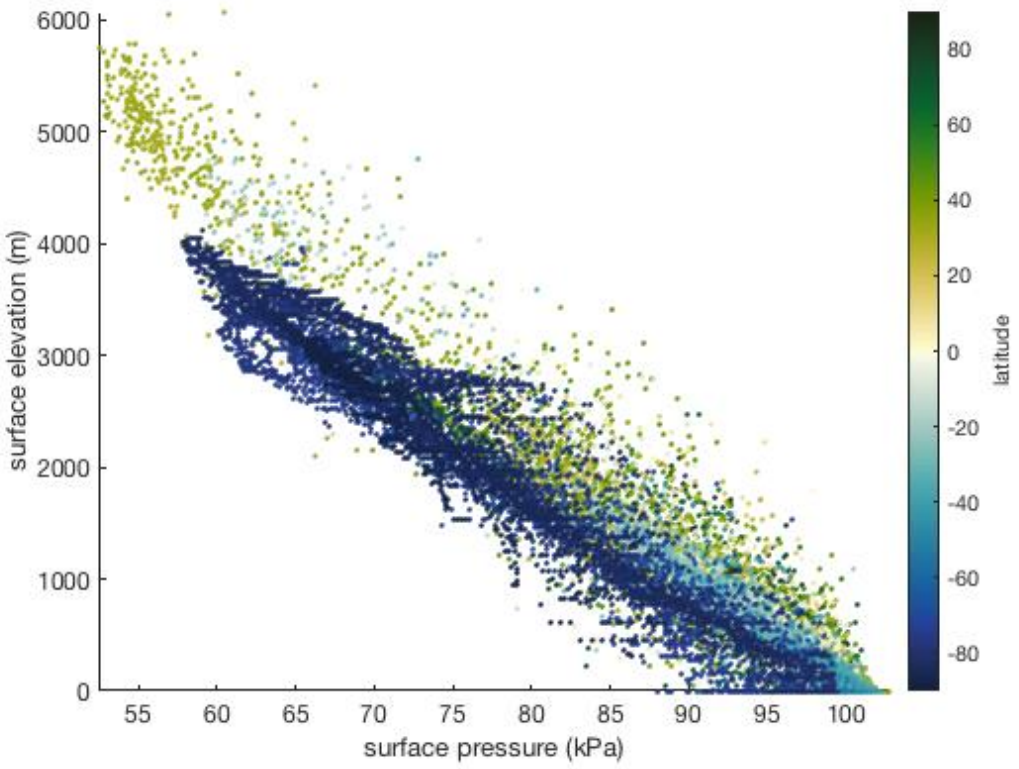
在上图中，你可能会注意到的第一件事是，地面气压大致对应于地貌。让我们通过叠加地形等值线来说明这一点。使用 topo_interp 获取海平面和海底地形，将水下地形设置为零，并覆盖等值线：

```
% 从经纬度数组获取 Lat，Lon 网格:
[Lat,Lon] = meshgrid(lat,lon);

% 获取地貌:
z = topo_interp(Lat,Lon);

% 将水下地形的值设为 0:
z(z<0) = 0;

hold on
contour(Lon,Lat,z,'k');
```



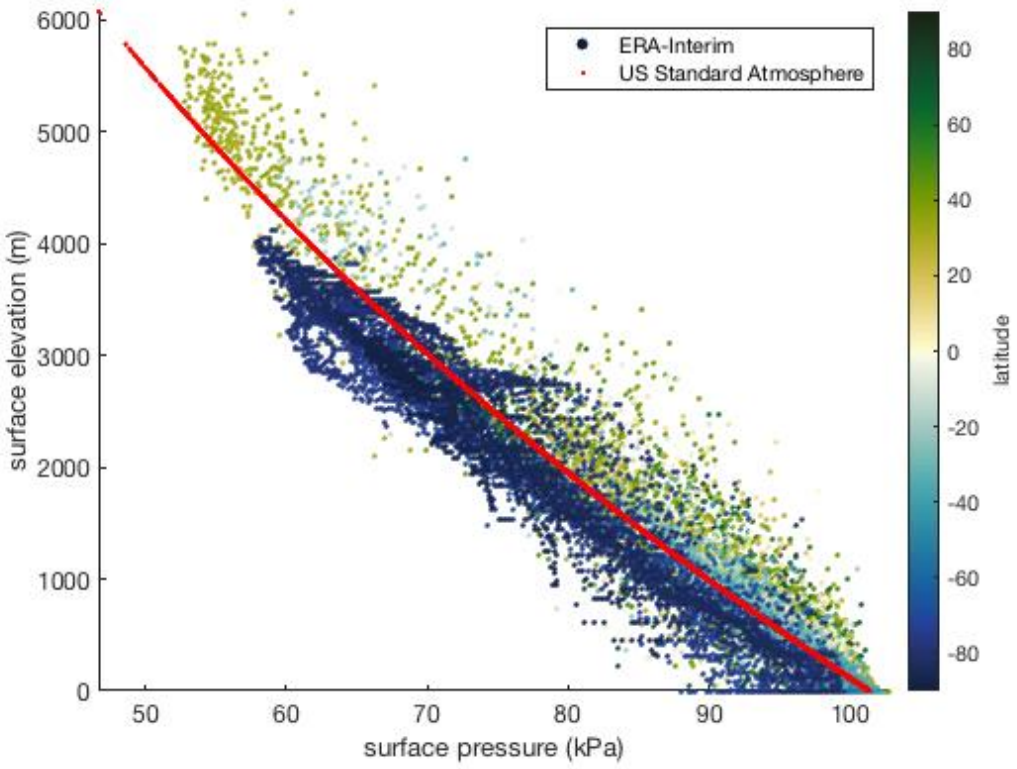
很明显，地面气压和地貌之间有关系。将关系显示为散点图

```
figure

scatter(spm(:)/1000,z(:),8,Lat(:),'filled')
cb = colorbar;
ylabel(cb,'latitude')
cmocean delta
xlabel 'surface pressure (kPa)'
ylabel 'surface elevation (m)'
axis tight
```



整洁的散点图，但它与美国标准大气中仅依赖于气压的理论高度相比如何？

```
p = air_pressure(z);



hold on
plot(p(:)/1000,z(:),'r.')
legend('ERA-Interim','US Standard Atmosphere')
```



想要一个定量的方法来衡量理论与再分析数据的吻合程度吗？有很多方法可以量化这种关系，比如相关系数和 p 值。不匹配的另一个度量是两者之间的均方根差：

```
rms([spm(:) - p(:)],'omitnan')/1000
```

```
ans =

    1.8792
```

均方根误差大约 1.88 kPa.

作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

air_density 文档

air_density 根据美国标准大气的气压场计算密度。

另请参见 air_pressure.

语法

```
rho = air_density(h)
```

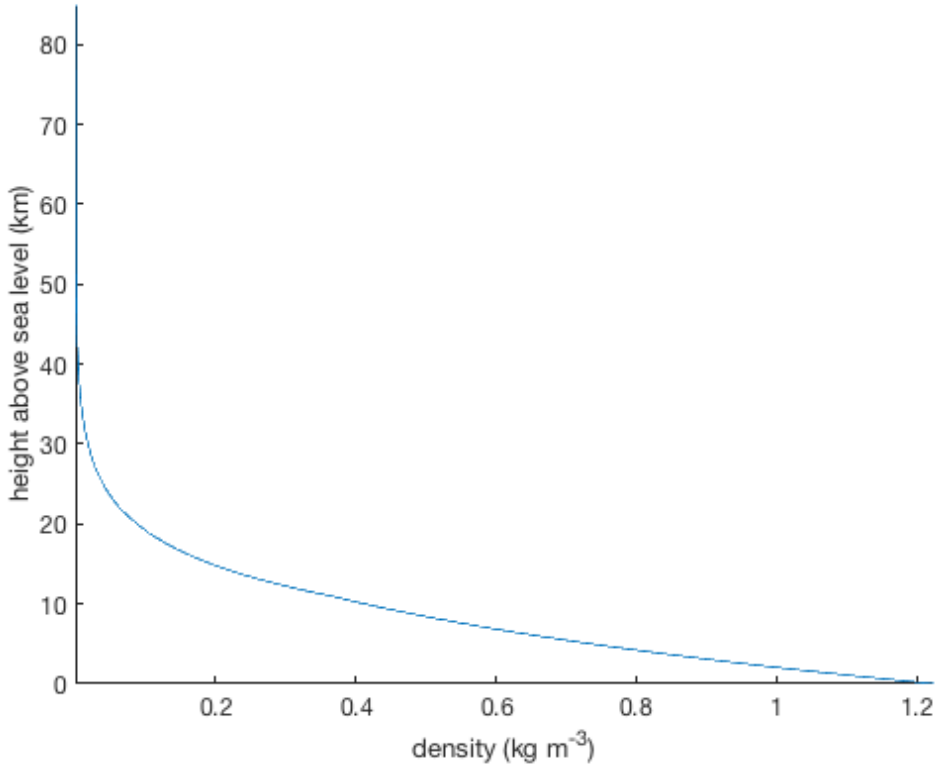
说明

rho = air_density(h) 返回大气密度 rho，单位为 kg/m^3，对应于海拔高度 h，单位为几何米。

示例

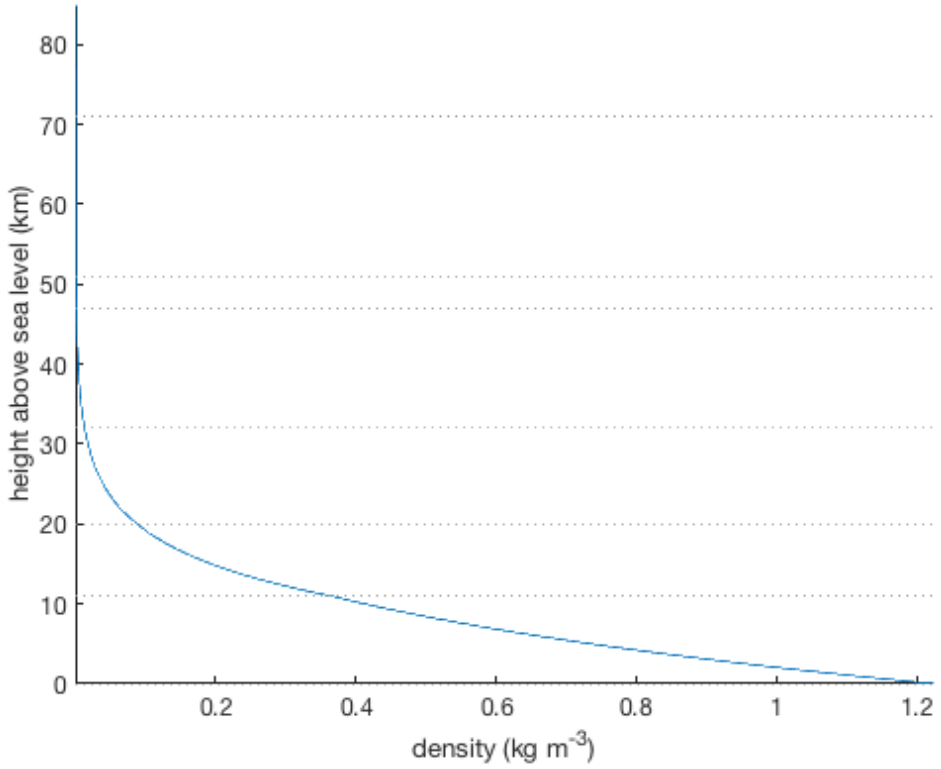
美国标准大气公式适用于海拔 86 公里，因此让我们看看从海拔到 85 公里的空气密度：

```
h = 0:85e3;



rho = air_density(h);



figure

plot(rho,h/1000)

axis tight
box off
ylabel 'height above sea level (km)'
xlabel 'density (kg m^{-3})'
```



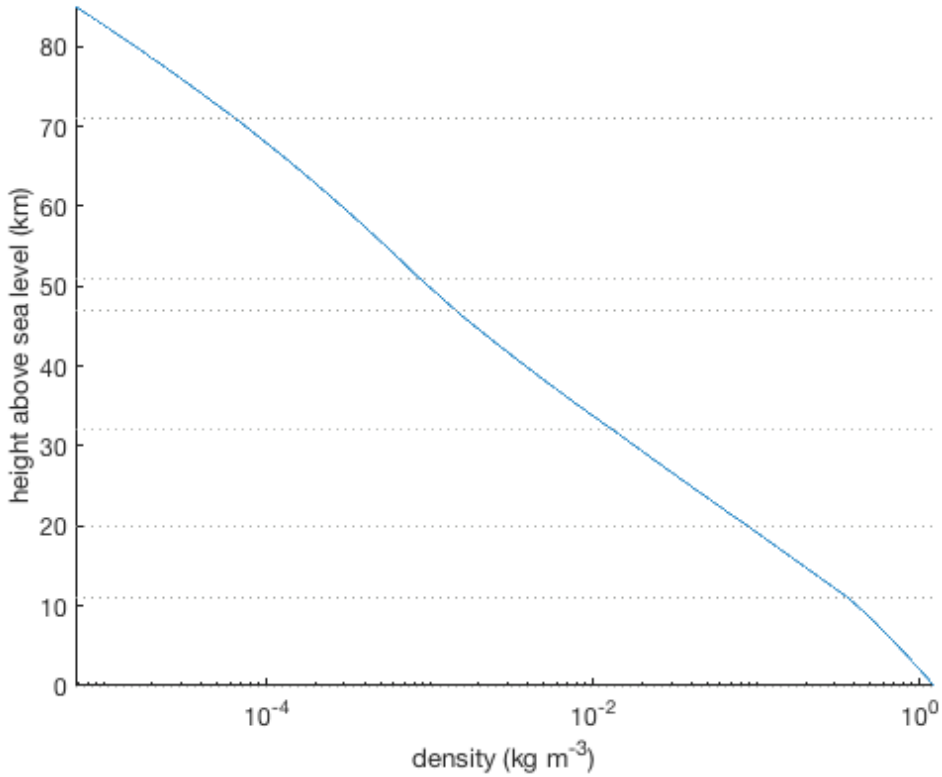
该公式使用了一个分段函数，根据不同大气层的不同常数计算出一个方程。这些地层的基于海平面以上 0、11、20、32、47、51 和 71 公里处。对于背景，我们可以使用 hline 将这些层基准绘制为水平线：

```
LayerBases = [0 11 20 32 47 51 71];



hline(LayerBases,':','color',rgb('gray'))
```



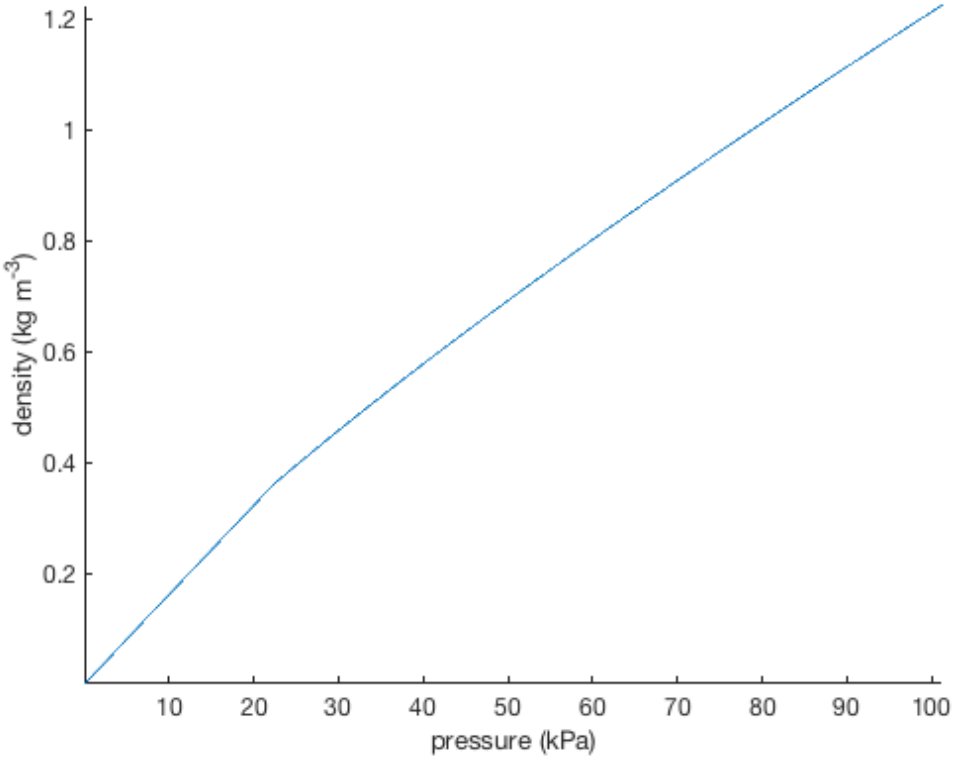
提示:对于某些应用程序,在对数刻度上显示 x 轴可能会有所帮助。要做到这一点,我们可以用 semilogx 来绘制数据，而不是使用上面的 plot 函数，或者我们可以将 x 标度改为 log，如下所示：

```
set(gca,'xscale','log')
```



好奇大气密度和大气压力之间的关系吗？与 air_pressure 比较：

```
p = air_pressure(h);



figure

plot(p/1000,rho)

axis tight
box off
xlabel 'pressure (kPa)'
ylabel 'density (kg m^{-3})'
```



作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

sun_angle 文档

sun_angle 给出了地球上任何地点任何时间的太阳方位角和仰角。此函数由 Darin C. Koblick 的 SolarAzEl 函数改编而来。

语法

```
[az,el] = sun_angle(t,lat,lon)
[az,el] = sun_angle(t,lat,lon,h)
```

说明

[az,el] = sun_angle(t,lat,lon) 给出指定地理位置的太阳方位角和仰角(度),以及时间 t(UTC)。输入 t 可以是 datenum、datestr 或 datetime 格式。

[az,el] = sun_angle(t,lat,lon,h) 指定海拔高度，单位为米。如果未指定高度，则通过 CDT 函数 topo_interp 自动确定高程。

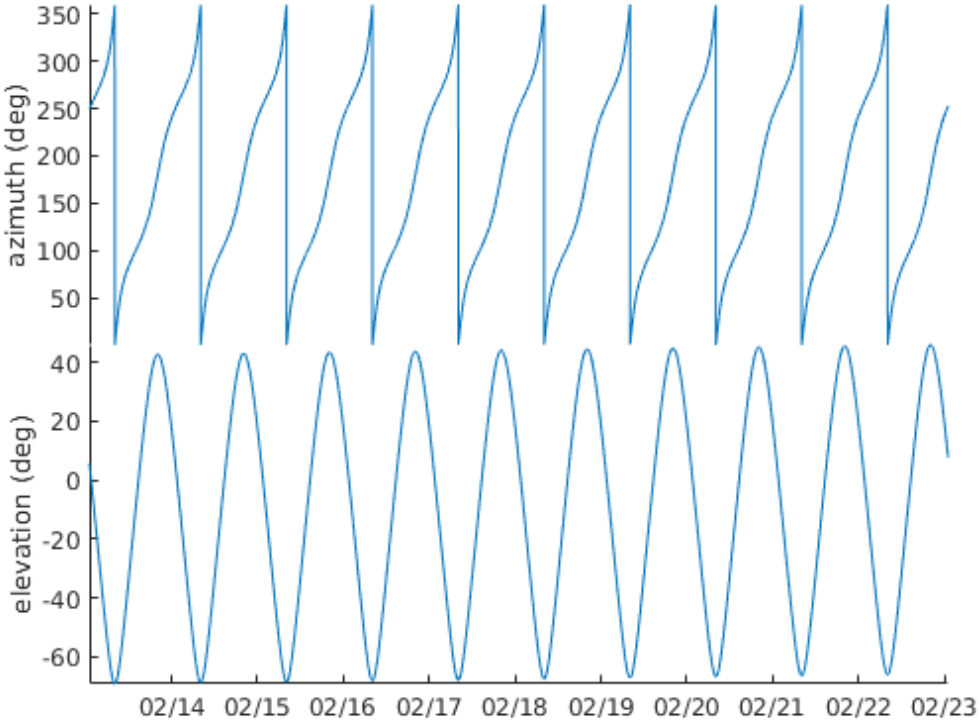
示例 1: 单点时间序列

当我写这篇文章时，我正坐在加利福尼亚州帕萨迪纳的一家咖啡店（北纬 34.16 度，西经 118.13 度）。确定现在和未来 10 天太阳的方位角和仰角。

```
t = linspace(now,now+10,10000); % 一个 10 天长的时间序列
t = t + 8/24; % 将太平洋标准时间转为 UTC

[az,el] = sun_angle(t,34.16,-118.13);
```

使用 subsubplot 绘制太阳角：

```
subsubplot(2,1,1)

plot(t,az)

axis tight
ylabel 'azimuth (deg)'
datetick('x','keeplimits')

subsubplot(2,1,2)
plot(t,el)
axis tight
ylabel 'elevation (deg)'
datetick('x','keeplimits')
```



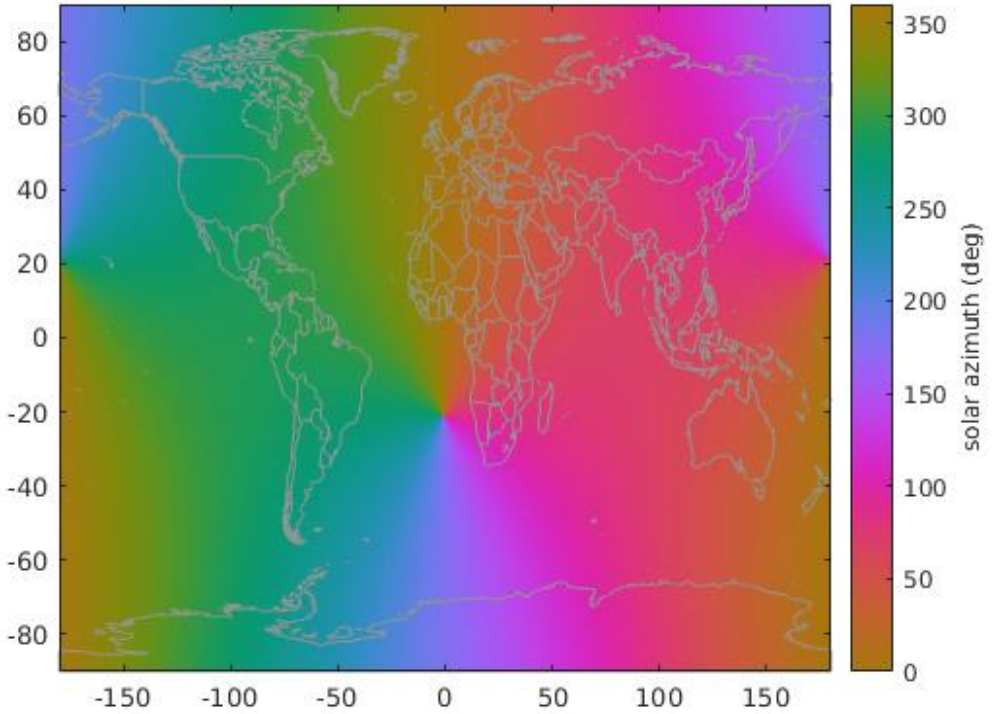
示例 2: 全球网格

使用 cdtgrid 创建四分之一度全球网格，并在 2019 年 5 月 27 日午夜 UTC 时获取全球各地的太阳角度：

```
[lat,lon] = cdtgrid(1/4);



[az,el] = sun_angle('may 27, 1984 00:00:00',lat,lon);
```

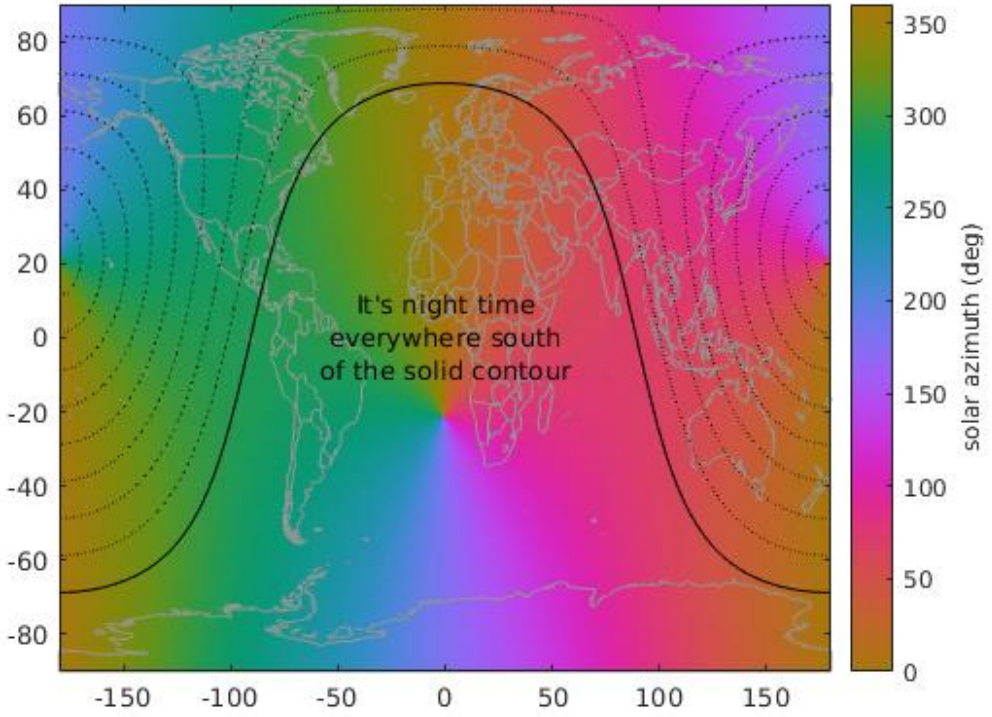
使用 imagescn 绘制太阳方位角，并将颜色图设置为 cmocean *phase*。使用 borders 添加政区边界。（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
figure

imagescn(lon,lat,az)

cb = colorbar;

ylabel(cb,'solar azimuth (deg)')
caxis([0 360])
cmocean phase
borders('countries','color',rgb('gray'))
```



将太阳高程叠加为等值线。当然，只有当太阳高度高于 0 时，太阳高度才是存在的，所以只包括正等值线：
译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

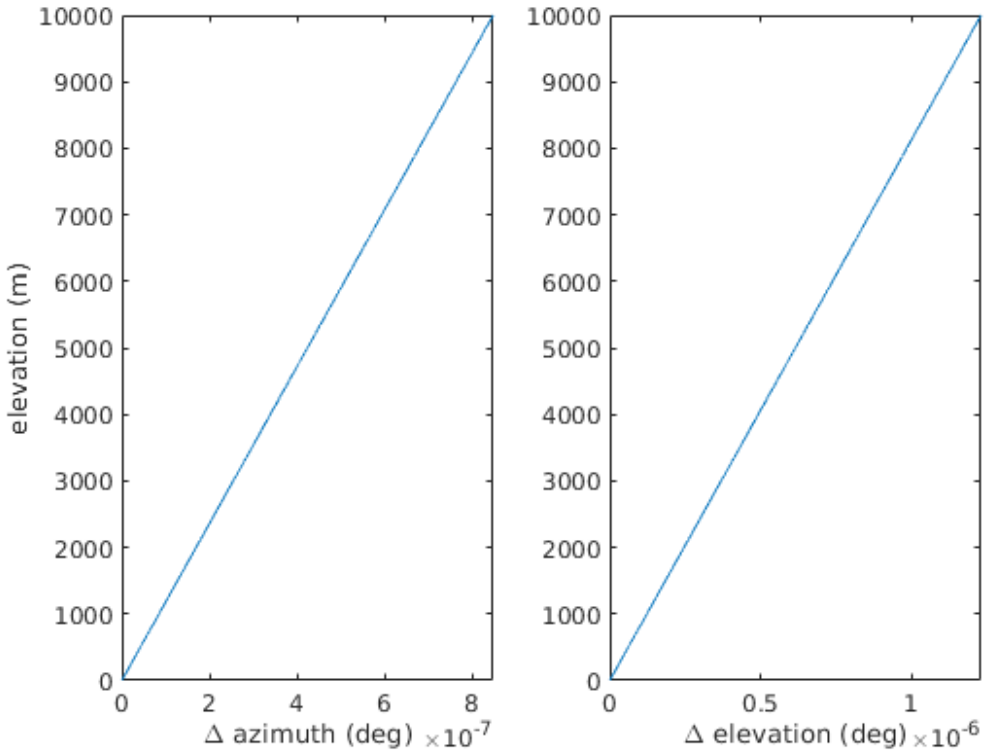
```
hold on
contour(lon,lat,el,[0 0],'k') % 0 等值线为实线
contour(lon,lat,el,0:10:90,'k:')

text(0,0,{'It''s night time';'everywhere south';'of the solid contour'},...
   'horiz','center')
```



示例 3: 海拔的微小影响

太阳的角度受海拔高度的影响。默认情况下，如果未指定高程，sun_angle 将使用 topo_interp 确定高于海平面的地面高程。但在这里，我们考虑一个假设塔，在伦敦中心 10000 米高（51.5N，0.123W）。2019 年 7 月 14 日晚上 7 点 45 分，这座塔的太阳方位角和仰角是多少？

```
t = 'july 14, 2019 7:45 pm';
lat = 51.51;
lon = -0.123;
z = 0:10000; % 沿着塔的高度 (m)

[az,el] = sun_angle(t,lat,lon,z);

figure
subplot(1,2,1)
plot(az-az(1),z) % 相对于基准绘图
axis tight
xlabel '\Delta azimuth (deg)'
ylabel 'elevation (m)'

subplot(1,2,2)
plot(el-el(1),z) % 相对于基准绘图
axis tight
xlabel '\Delta elevation (deg)'
```



注意 x 轴刻度上的科学符号。这告诉我们海拔的影响在几千米的范围内是微乎其微的。

作者简介

该函数原来是 Darin C. Koblick 使用 http://stjarnhimlen.se/comp/tutorial.html#5.描述的方程编制的。它在 2019 年被 Chad A. Greene.改写以适应于 Climate Data Toolbox for Matlab.

solar_radiation 文档

solar_radiation 函数计算在地球大气层顶部每天接收到的天文总辐射。
这个功能和 daily_insolation 功能非常相似，一个比另一个更适合你的需要。daily_insolation 函数最适合于研究数千至数百万年的轨道变化，而 solar_radiation 可能更容易应用于当今的降水/干旱研究。

语法

```
Ra = solar_radiation(t,lat)
```

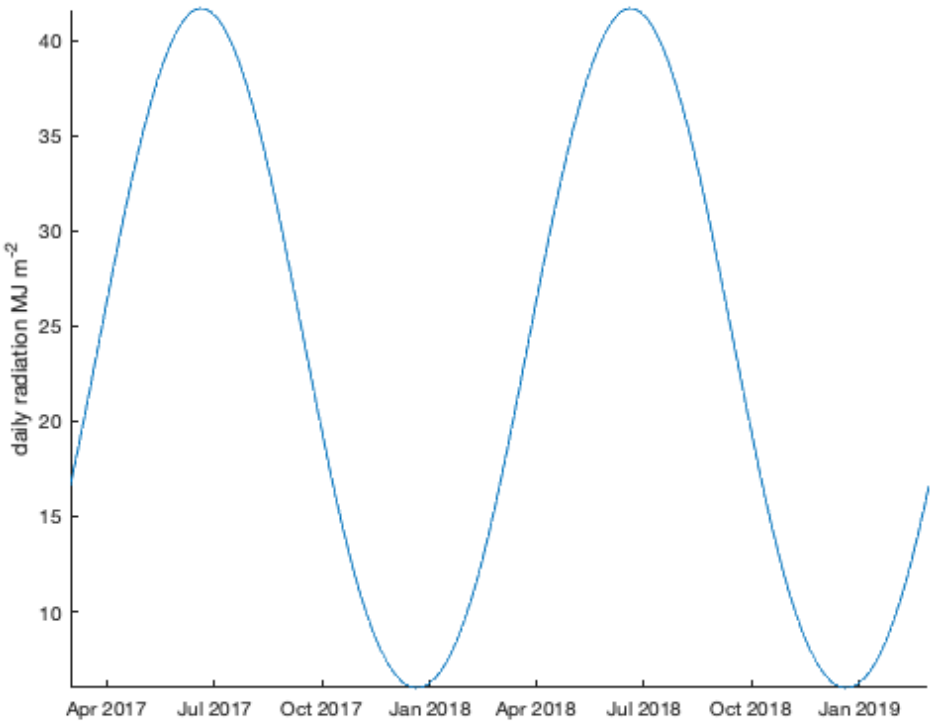
说明

Ra = solar_radiation(t,lat) 基于日期 t 和纬度 lat 计算地外辐射(MJ m^(-2) day^(-1))。日期 t 可以是 datetime、datenum 或 datestr 格式，但必须是一维数组或标量。lat 可以是标量、向量或网格。如果 lat 是具有大小 nrows 和 ncols 的向量或数组，则 Ra 具有大小[nrows,ncols,length(t)]。

示例 1: 柏林的太阳能电池板

假设你在德国柏林有一块 1 平方米的太阳能电池板（52.5N，13.4E）。如果没有大气层，从 2017 年 3 月 1 日到 2019 年 3 月 1 日，它每天能接收多少焦耳的能量？

```
% 定义日期数组:
t = datetime('march 1, 2017'):datetime('march 1, 2019');

% 定义柏林经度:
lat = 52.5;

% 计算日总辐射:
Ra = solar_radiation(t,lat);

% 绘制时间序列:
figure
plot(t,Ra)
axis tight
box off % 移除边框
ylabel 'daily radiation MJ m^{-2}'
```



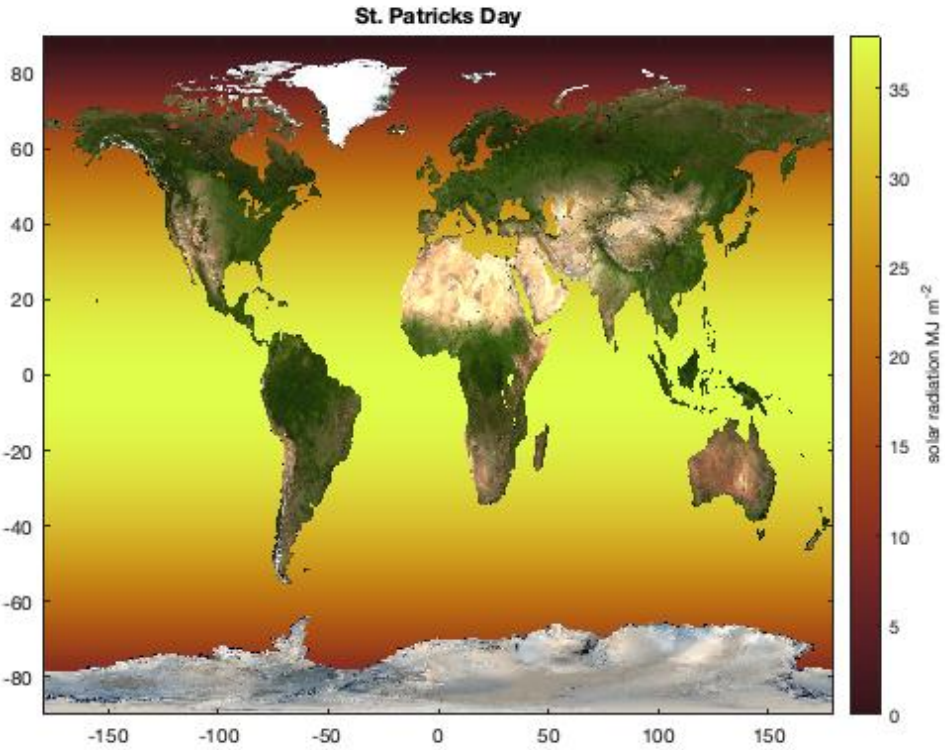
示例 2: 世界海洋中的太阳能

在我们开始这个例子之前，需要注意的是 solar_radiation 函数估计了大气顶部的辐射，所以下面的讨论忽略了所有的大气效应。考虑到这一点，我们来考虑一下在给定的一天里，多少太阳能撞击地球的海洋。

首先，选择一天。3 月 17 日的圣帕特里克节怎么样？估计圣帕特里克节有多少能量撞击地球的海洋，意味着我们必须制作一个全球网格，并计算网格中每个点的太阳辐射。非常密集的网格会更精确，但是非常粗糙的网格需要更少的内存。让我们使用 cdtgrid 创建一个半度全球网格，并计算网格上每个点的辐射：

```
% 做一个半度的网格:
[Lat,Lon] = cdtgrid(1/2);

% 计算圣帕特里克节的辐射:
Ra = solar_radiation('march 17',Lat);

% 绘制
figure
imagescn(Lon,Lat,Ra)
hold on
earthimage('watercolor','none')
cmocean solar
cb = colorbar;
ylabel(cb,'solar radiation MJ m^{-2}')
```

```
title 'St. Patricks Day'
```



以上，用 earthimage 绘制了陆地表面，用 cmocean 设置了彩色地图。
现在你可能想知道，与陆地相比，海洋能接收多少能量？回答这个问题需要我们知道每个网格单元的面积，以计算每个网格单元接收到的总能量。使用 cdtarea 得到每个网格单元的面积，并将 Ra 乘以面积，得到圣帕特里克节每个网格单元收到的总能量：

```
% 获取每个单元网格的面积:
A = cdtarea(Lat,Lon);  % (m^2)

% 计算每个单元网格吸收的总能量:
E = Ra.*A;
```

那么，陆地和海洋哪一个能接收到更多的太阳能呢？使用 island 来确定对应的网格单元，然后总结撞击陆地和海洋的太阳能：

```
% 获取每个陆地单元网格的面积 Get a mask of grid cells that are land:
land = island(Lat,Lon);

% 圣帕特里克节撞击陆地的总太阳能:
sum(E(land)) % MJ
```

```
ans =

    4.0392e+15
```

这是 4*10^15MJ 的太阳能，将在圣帕特里克节到达地表。对海洋：

```
sum(E(~land)) % MJ
```

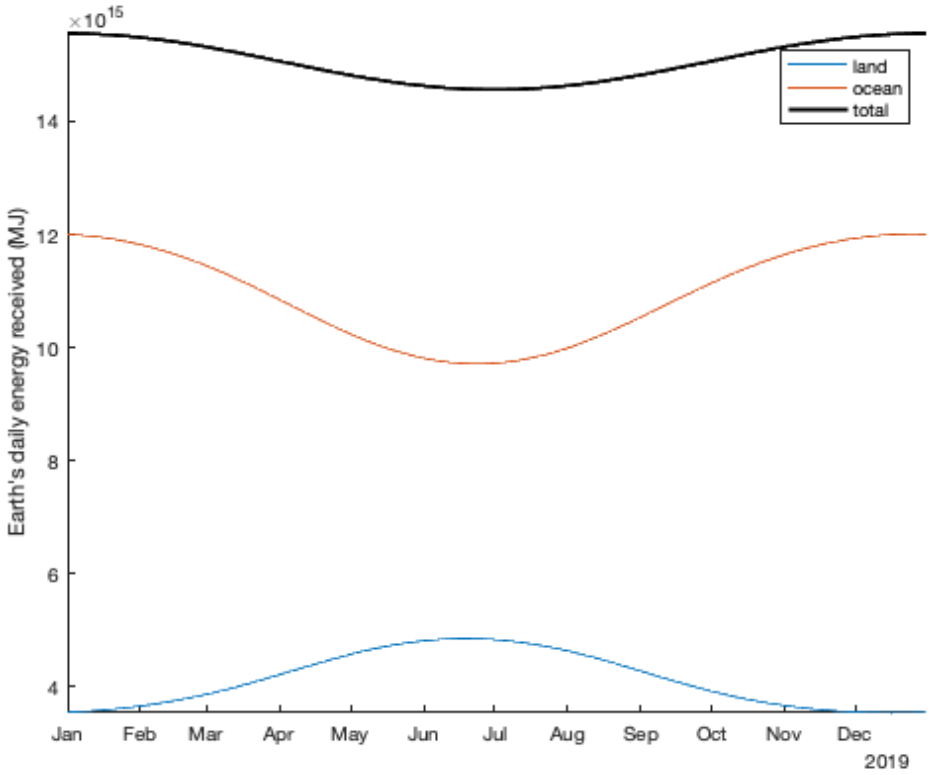
```
ans =

   1.1155e+16
```

撞击海洋的是 1*10^16MJ。毫不奇怪，海洋比陆地有更大的表面积，所以它接收更多的太阳能。
我们可以通过输入 t 作为时间数组，将此分析扩展到时间变化，如下所示，我们计算 2019 年每天的太阳辐射：

```
t = datetime('jan 1, 2019'):datetime('dec 31, 2019');

Ra = solar_radiation(t,Lat);
```

以上，我们输入了 1x365 日期时间数组 t 和 360x720 的网格经度 Lat。得到的 Ra 是 360x720x365，对应于一年中每一天的网格化解决方案。
将 Ra 乘以网格单元面积 A 得到每个网格单元接收到的能量的网格化时间序列，并使用 local 函数得到陆地与海洋的能量事件时间序列：

```
% 每日能量是 Ra 乘以网格单元面积:
E = Ra.*A;

% 陆地和海洋能量总和:
E_land = local(E,land,@sum);
E_ocean = local(E,~land,@sum);

figure
plot(t,E_land)
hold on
plot(t,E_ocean)
plot(t,E_land+E_ocean,'k','linewidth',2)
axis tight
box off
legend('land','ocean','total')
ylabel 'Earth''s daily energy received (MJ)'
```



上图显示，在一年中的任何时候，海洋接收的太阳能都比陆地表面多。这并不奇怪，因为海洋的表面比地球的陆地表面大。但是那里也发生了一些有趣的事情。

对于地球上居住在赤道以北的大多数人口来说，陆地的曲线可能并不奇怪。如果你生活在北半球，你可能会看到这一点，然后想，“是的，地球在夏天更暖和。”但是当然，当北半球是夏天的时候，南半球是冬天，所以事情不应该是一样的吗？

不，因为地球的大部分陆地面积都在北半球，所以从总量上讲，地球的陆地表面在 6 月份接收的能量比 12 月份多。

当我们观察海洋表面太阳能一年四季的变化时，我们看到的和陆地上的正好相反。这部分是因为海洋在南半球的表面积比北半球大，但这并不能完全解释 6 月和 7 月海洋能量的下降。

看看所接收的太阳能总量，我们发现它并不是全年不变的。事实上，地球在每年的 7 月 4 日左右离太阳最远，所以总的来说，那是地球接收到的太阳能最少的时候。

参考文献

---

solar_radiation 函数计算公式引用如下：

- Allen, R.G., Pereira, L.S., Raes, D., and Smith, M.: Crop evapotranspiration Guidelines for computing crop water requirements FAO Irrigation and Drainage Paper 56, Food and Agriculture Organization of the United Nations, 1998
- McCullough (1968) McCullough, E.C.: Total daily radiant energy available extraterrestrially as a harmonic series in the day of the year, Arch. Met. Geoph. Biokl., Ser. B, 16, 129-143, 1968.
- McCullough and Porter (1971) McCullough, E.C. and Porter, W.P.: Computing clear day solar radiation spectra for the terrestrial ecological environment, Ecology, 52, 1008-1015, 1971.

- McMahon, T. A., et al. "Estimating actual, potential, reference crop and pan evaporation using standard meteorological data: a pragmatic synthesis." Hydrology and Earth System Sciences 17.4 (2013): 1331-1363 https://doi.org/10.5194/hess-17-1331-2013.

另请参见 the supplement to McMahon et al 2013 here.

作者简介

---

solar_radiation 函数是由 José Delgado 和 Wolfgang Schwanghart（波茨坦大学）于 2019 年 2 月写的，Climate Data Toolbox for Matlab。

daily_insolation 文档

daily_insolation 计算的是在过去 500 万年中，任何一点日均日射能量作为日期和纬度的函数。这个函数来自 Ian Eisenman 和 Peter Huybers（参见下面的参考文献）。daily_insolation 函数与 solar_radiation 函数非常相似，但一种可能比另一种更适合你的需要。daily_insolation 函数最适合于涉及数千至数百万年轨道变化的研究，而太阳辐射可能更容易用于诸如当今降水/干旱研究等应用。

语法

```
Fsw = daily_insolation(kyear,lat,day)
Fsw = daily_insolation(kyear,lat,day,day_type)
Fsw = daily_insolation(kyear,lat,day,day_type,'constant',So)
Fsw = daily_insolation(kyear,lat,day,day_type,'mjmd')
[Fsw, ecc, obliquity, long_perh] = daily_insolation(...)
```

说明

Fsw = daily_insolation(kyear,lat,day) 给出了一年中指定日期（由 doy 函数给出）纬度上千年前的日平均日照（W/m^2）。例如，使用 kyear =+3000 表示距今 300 万年。kyear 的最大允许值为 5000。

Fsw = daily_insolation(kyear,lat,day,day_type) 将可选的日类型指定为 1（默认值）或 2。默认选项 1 指定的日范围为 1 到 365.24，其中第 1 天是 1 月 1 日，春分总是出现在第 80 天。选项 2 将日输入指定为 0 到 360 度范围内的太阳经度。太阳经度是从春分（3 月 21 日）开始测量的地球轨道的角度。请注意，日历日和太阳经度并不是线性相关的，因为根据开普勒第二定律，地球的角速度随其与太阳的距离而变化。如果 day_type 为负数，则将 kyear 视为包含[偏心率、倾角和近日点经度]的 3 元素数组。

Fsw = daily_insolation(kyear,lat,day,day_type,'constant',So) 表示太阳常数 So。默认 So 为 1365 W/m^2。

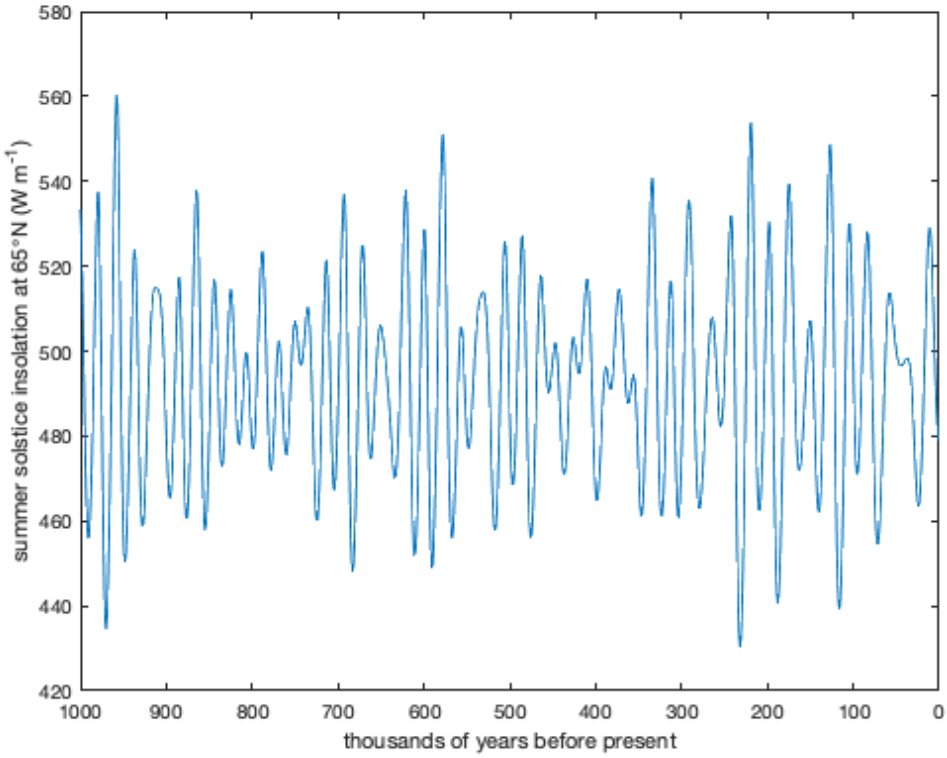
Fsw = daily_insolation(kyear,lat,day,day_type,'mjmd') 以(MJ/m^2)/day 为单位返回 Fsw，而不是默认的 W/m^2。

[Fsw, ecc, obliquity, long_perh] = daily_insolation(...) 还返回轨道偏心率、倾角和近日点经度（进动角）

示例 1

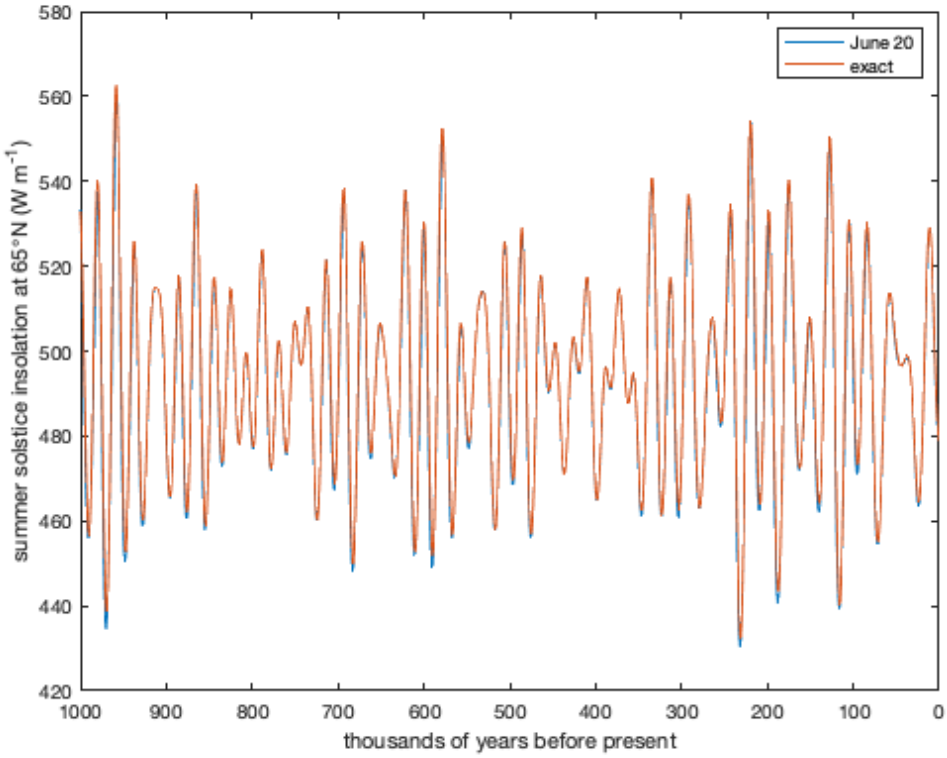
对于这个例子，计算在 65 N 的夏至日照。我们用两种不同的方法来做。首先，使用 doy 得到一年中与至点相对应的一天，假设它发生在 6 月 20 日：

```
kyr = 0:1000;



Fsw_jun20 = daily_insolation(kyr,65,doy('june 20'));

plot(kyr,Fsw_jun20)
set(gca,'xdir','reverse') % 翻转 x 轴方向
ylabel('summer solstice insolation at 65\circN (W m^{-1})')
xlabel 'thousands of years before present'
```



与其将 6 月 20 日近似为至点，更精确的方法是使用 day_type=2 选项指定对应于至点（90 度）的确切轨道角，如下所示：

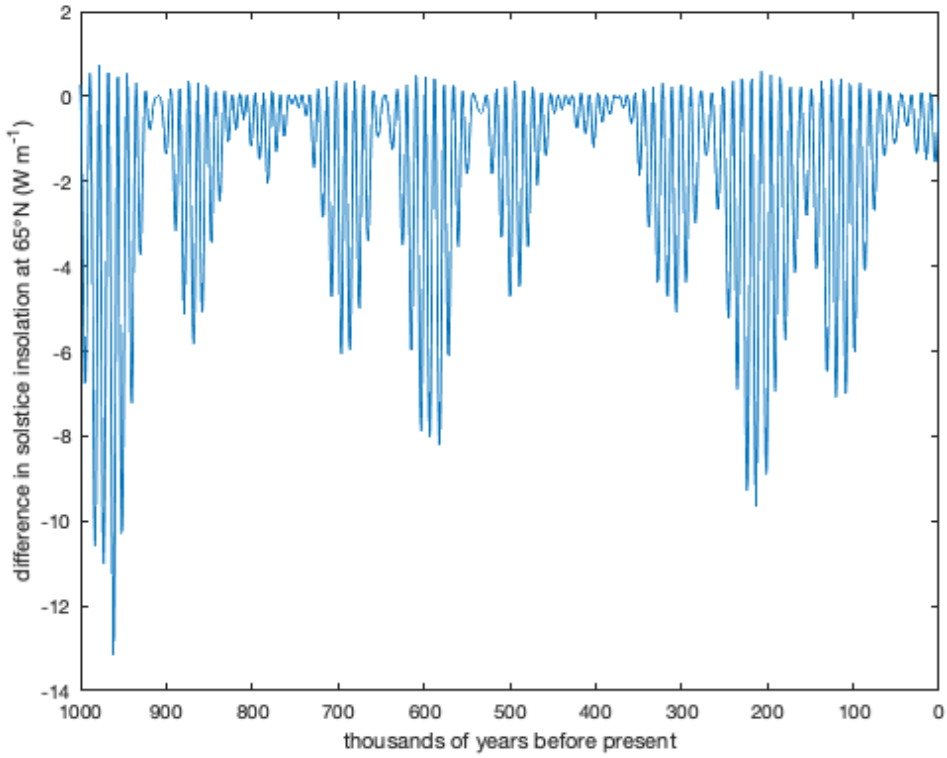
```
Fsw_exact = daily_insolation(kyr,65,90,2);



hold on
plot(kyr,Fsw_exact)

legend('June 20','exact')
```



示例 2

我们可以画出 65 N 在 6 月 20 日和准确的夏至日照之间的差异。

```
Fsw_jun20 = daily_insolation(kyr,65,doy('june 20'));
Fsw_exact = daily_insolation(kyr,65,90,2);

figure
plot(kyr,Fsw_jun20-Fsw_exact)
set(gca,'xdir','reverse') % 翻转 x 轴方向
ylabel('difference in solstice insolation at 65\circN (W m^{-1})')
xlabel 'thousands of years before present'
```



示例 3

当前轨道结构的日射，作为日期和纬度的函数。由于这是两个变量的函数，请使用 meshgrid 创建日期和纬度变量的网格：

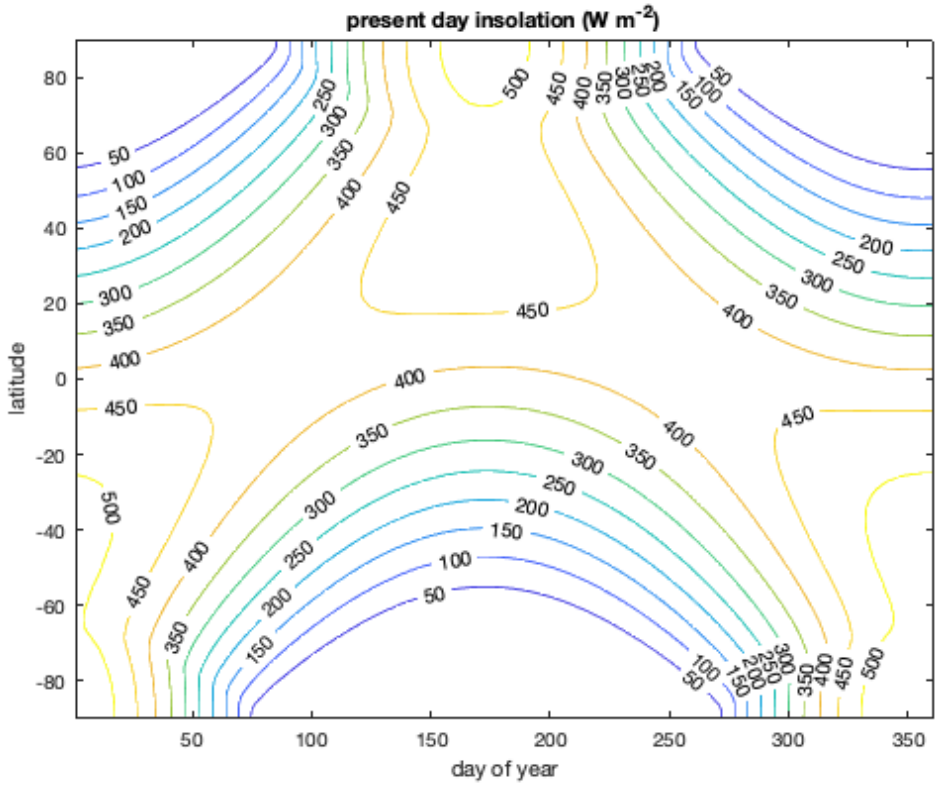
```
[day,lat]=meshgrid(1:5:365, -90:90);

[Fsw,ecc,obl,omega]=daily_insolation(0,lat,day);



disp([ecc,obl,omega]) % 显示可选输出
```

```
    0.0172   23.4460  281.3700
```

我们现在可以将日照作为日期和纬度的函数绘制如下：

```
figure

[c,h]=contour(day,lat,Fsw,[0:50:500]);

clabel(c,h)

xlabel 'day of year'
ylabel 'latitude'
```

```
title 'present day insolation (W m^{-2})'
```

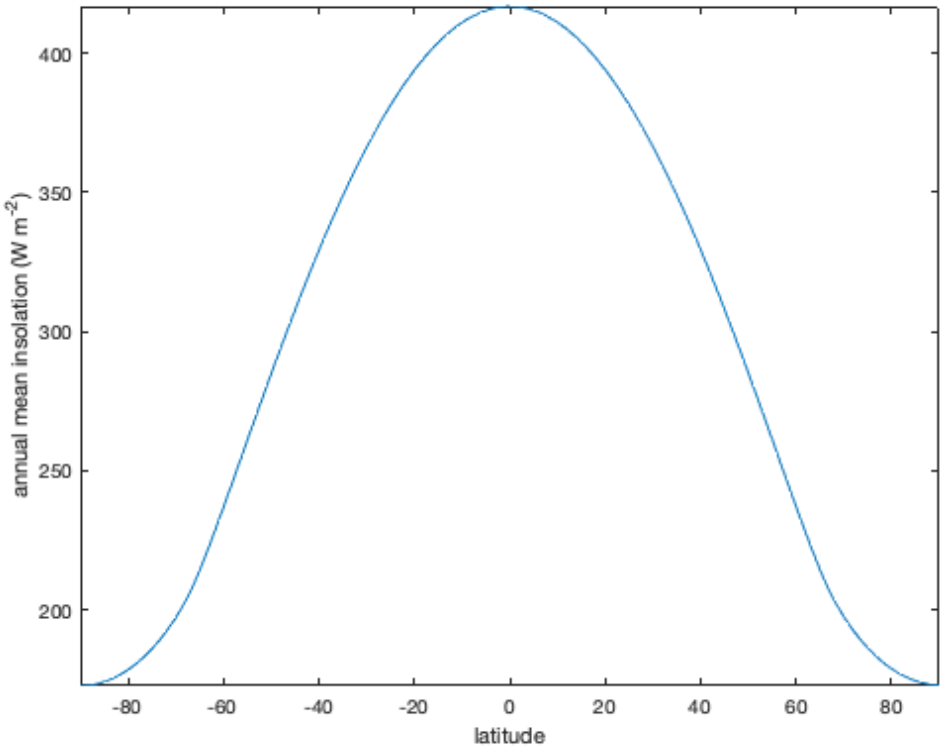


示例 4

通过明确指定轨道参数来计算年（而不是日）平均日照。

```
ecc=0.017236;

obl=23.446;

omega=101.37+180;

[day,lat]=meshgrid(1:5:365, -90:90);



Fsw=daily_insolation([ecc,obl,omega],lat,day,-1);



Fsw_annual = mean(Fsw,2);
```

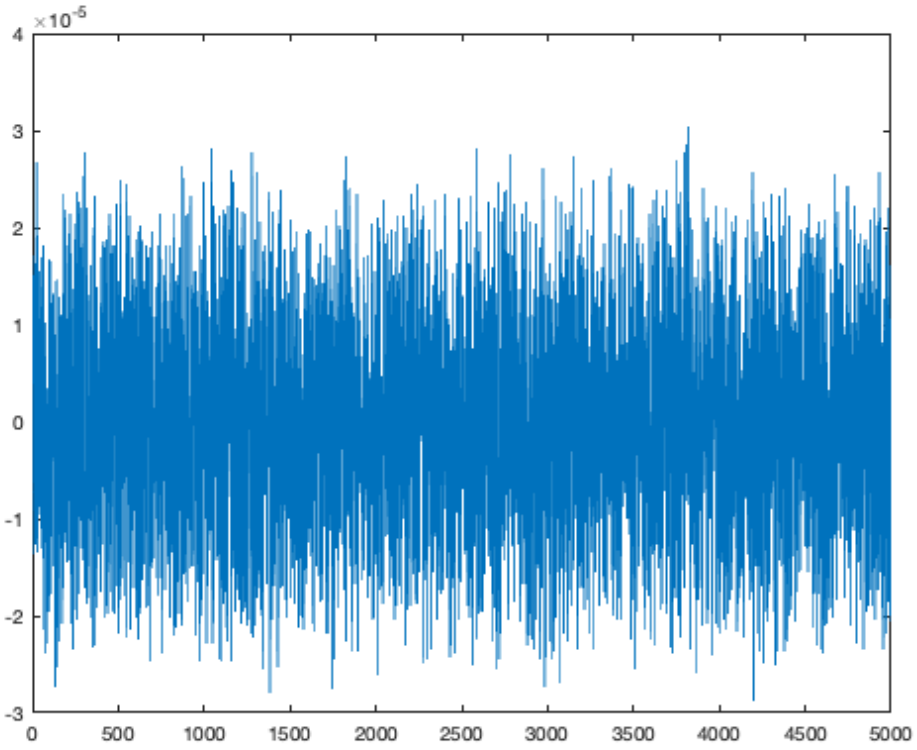
```
figure

plot(-90:90,Fsw_annual)

axis tight
xlabel 'latitude'
ylabel 'annual mean insolation (W m^{-2})'
```



示例 5

将计算的日照与 Berger (1991)给出的示例值进行比较。从 Berger A. and Loutre M.F., 1991 加载地球轨道参数开始，作为 CDT 中的样本数据集。请注意，Berger 和 Loutre 在千年中使用了负值，因此我们必须将它们乘以-1。此外，他们使用了 1360 W/m^2 的太阳常数，因此我们也将指定该值。

```
D = importdata('orbit91.txt'); %

kyear = -D.data(:,1);
insol_65NJul = D.data(:,6);

Fsw=daily_insolation(kyear,65,(7-3)*30,2,'constant',1360);

figure
plot(kyear,1-insol_65NJul./Fsw)
```

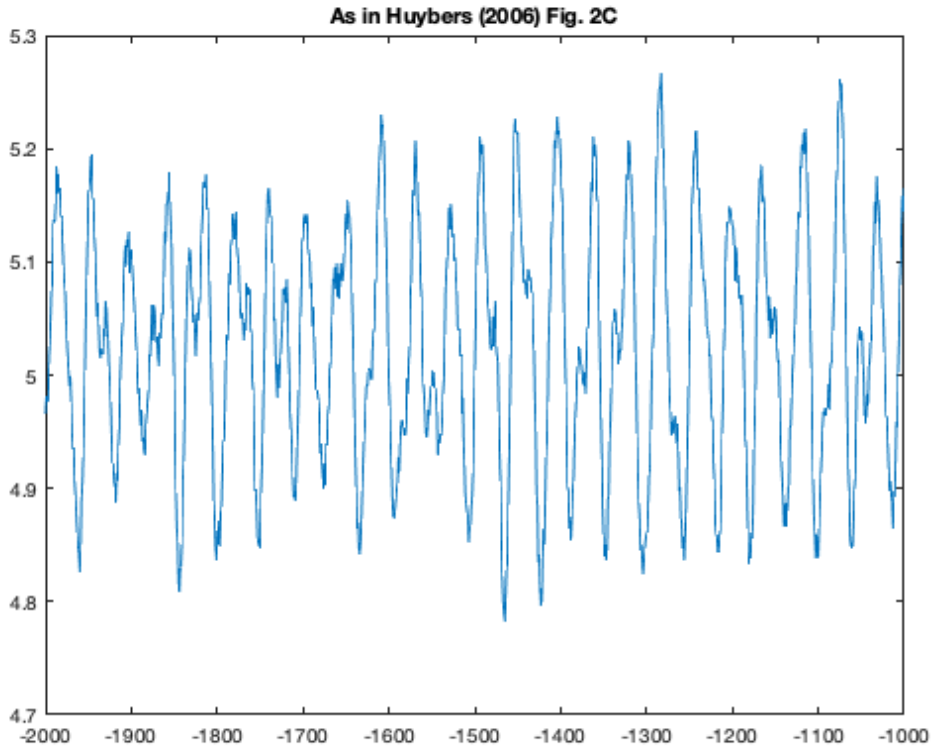


注意，上述数值在 3e-5 范围内一致。

示例 6

绘制 65N 的夏季综合日照 Huybers (2006), Science 313 508-511:

```
[kyear,day]=meshgrid(1000:1:2000,1:1:365);

Fsw = daily_insolation(kyear,65,day);

Fsw(Fsw<275)=0;



figure

plot(-(1000:1:2000),sum(Fsw,1)*86400*10^-9)

title('As in Huybers (2006) Fig. 2C')
```



详细计算方法的说明：

过去 5 百万年的偏心率、倾角和近日点经度值取自 Berger 和 Loutre 1991（数据来自 ncdc.noaa.gov）。如果使用日历日，则太阳经度是使用表示角动量守恒（开普勒第二定律）的微分方程的近似解来计算的。在给定轨道参数和太阳经度的情况下，日平均日照量的计算完全遵循 Berger 1978。

参考文献:

- Berger A. and Loutre M.F. (1991). Insolation values for the climate of the last 10 million years. Quaternary Science Reviews, 10(4), 297-317.
- Berger A. (1978). Long-term variations of daily insolation and Quaternary climatic changes. Journal of Atmospheric Science, 35(12), 2362-2367.
- Huybers, Peter. "Early Pleistocene glacial cycles and the integrated summer insolation forcing." science 313.5786 (2006): 508-511.

作者:

Ian Eisenman 和 Peter Huybers,哈佛大学, 2006 年 8 月 eisenman@fas.harvard.edu. 它在 2019 年被 Chad A. Greene 轻微改写以适应于 Climate Data Toolbox for Matlab.

topo_interp 文档

topo_interp 函数插值相对于任何地理点的海平面高程。数据来自 ETOPO5 世界数字高程模型，该模型以 5 分（或 1/12 度）网格分辨率提供。

语法

```
zi = topo_interp(lati,loni)
zi = topo_interp(lati,loni,'method',InterpMethod)
```

说明

zi = topo_interp(lati,loni) 返回相对于地理点 lati,loni 处海平面的高程。
zi = topo_interp(lati,loni,'method',InterpMethod) 指定插值方法，可以是 interp2 接受的任何方法。默认方法为'linear'。

示例 1: 一个网格

这是一个分辨率为 0.25 度的全球网格：

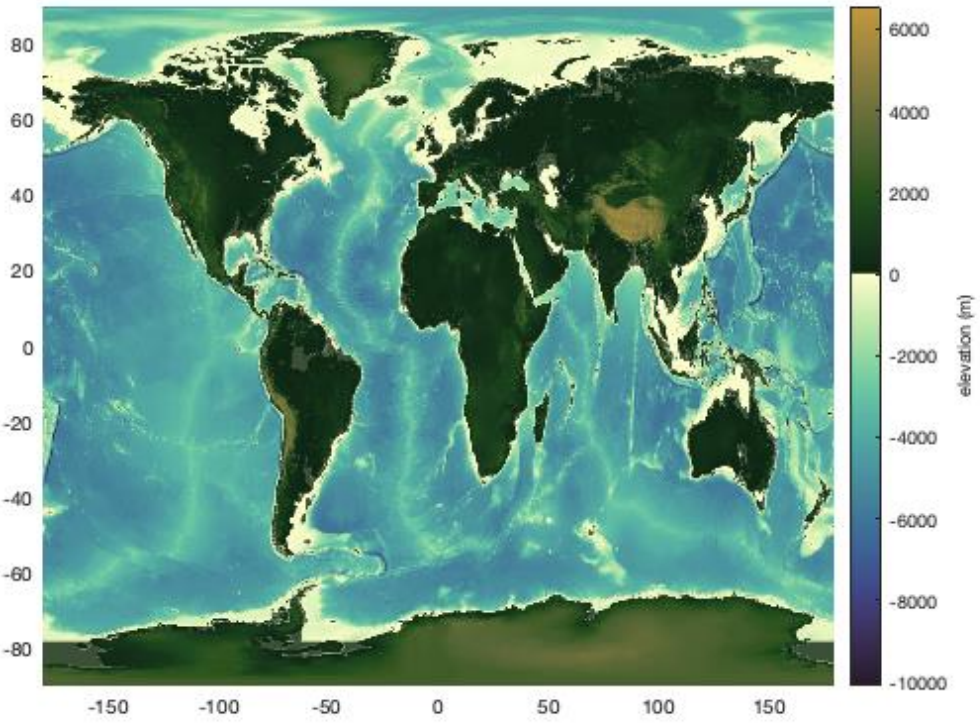
```
[lat,lon] = cdtgrid(0.25);
```

每个网格点的高程如下所示：

```
Z = topo_interp(lat,lon);
```

像这样绘制立面网格，下面我使用 cmocean 作为地形颜色图：

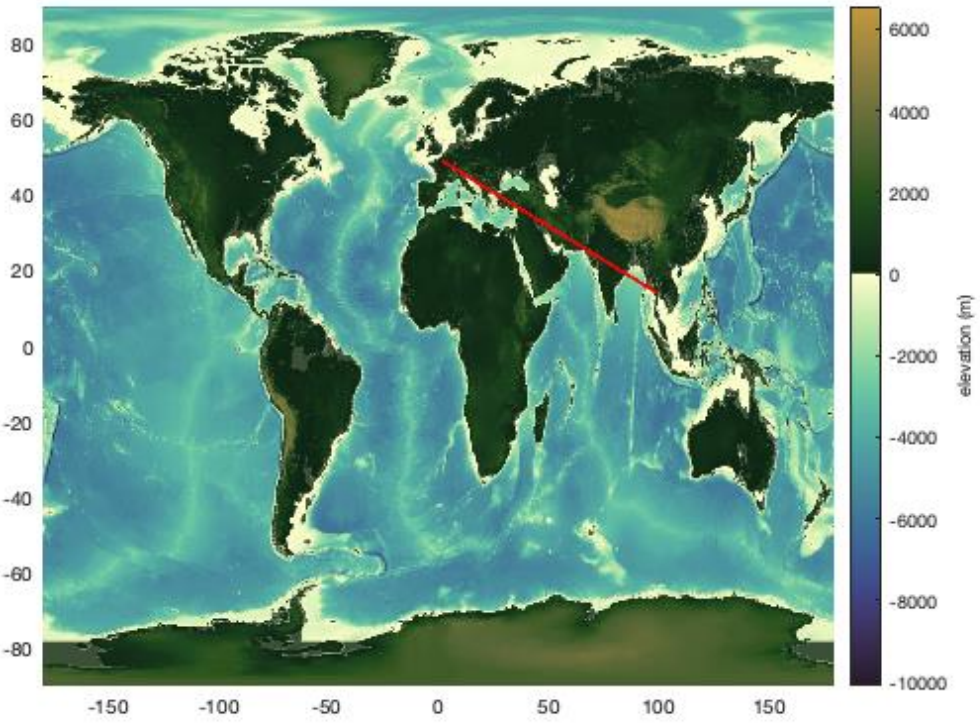
```
pcolor(lon,lat,Z)

shading interp

cb = colorbar;
ylabel(cb,'elevation (m)')
cmocean('topo','pivot') % 设置颜色图
```



示例 2: 一个高程剖面

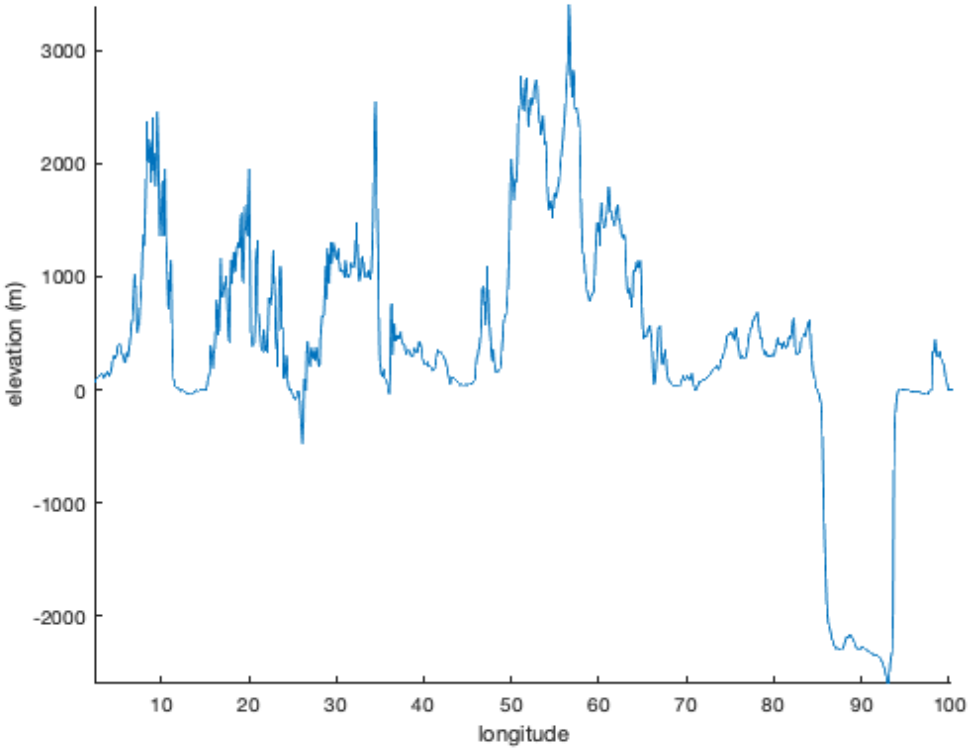
假设您想从巴黎（48.8567N，2.3508E）到曼谷（13.7525N，100.494167E）。这是两市之间由 1000 个点组成的粗线：

```
lat = linspace(48.8567,13.7525,1000);

lon = linspace( 2.3508,100.494167,1000);



hold on
plot(lon,lat,'r-','linewidth',2)
```



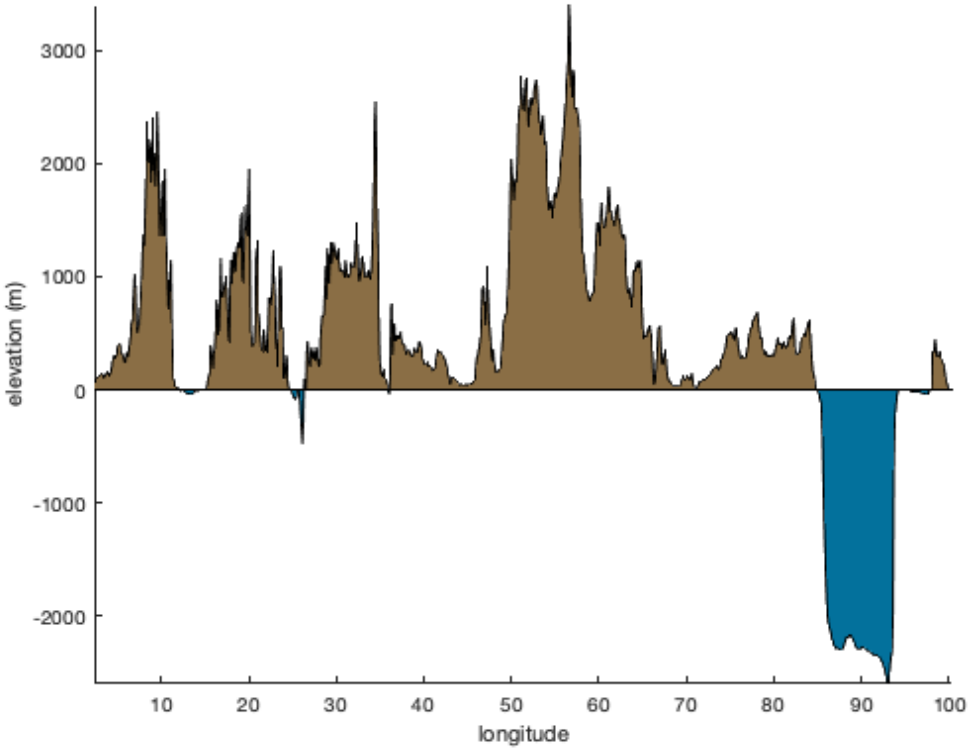
沿着该路线的高程剖面非常简单：

```
z = topo_interp(lat,lon);



figure

plot(lon,z)

axis tight
box off
xlabel 'longitude'
ylabel 'elevation (m)'
```



或者，为了更直观地描绘陆地和水域，请使用 anomaly 函数并使用 rgb 指定颜色：

```
figure

anomaly(lon,z,'topcolor',rgb('dirt'),...
   'bottomcolor',rgb('ocean blue'))
axis tight
box off
xlabel 'longitude'
ylabel 'elevation (m)'
```



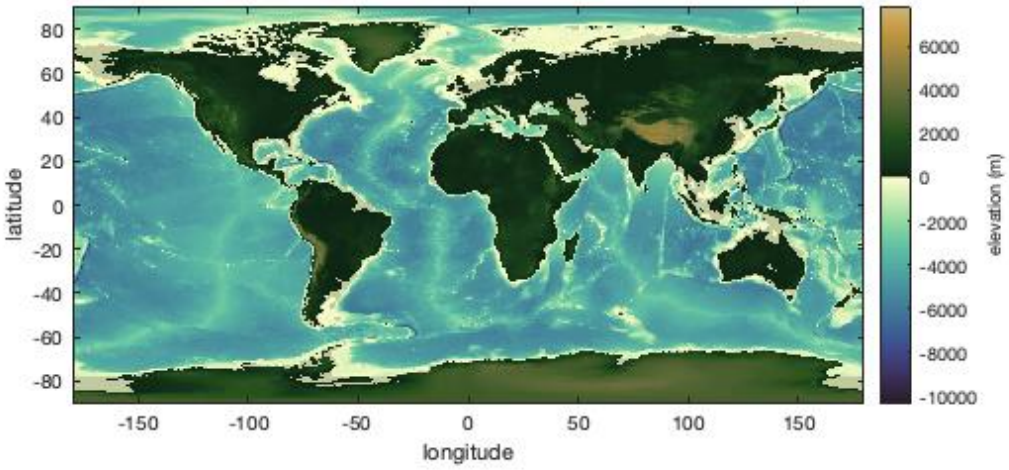
示例 3: 载入原始数据

要访问原始数据，只需键入

```
load('global_topography.mat');
```

其中包含变量 lat, lon, 和 Z。把它们画成这样：

```
figure

imagesc(lon,lat,Z)

axis xy image
xlabel 'longitude'
ylabel 'latitude'
cb = colorbar;
ylabel(cb,'elevation (m)')
cmocean('topo','pivot')
```



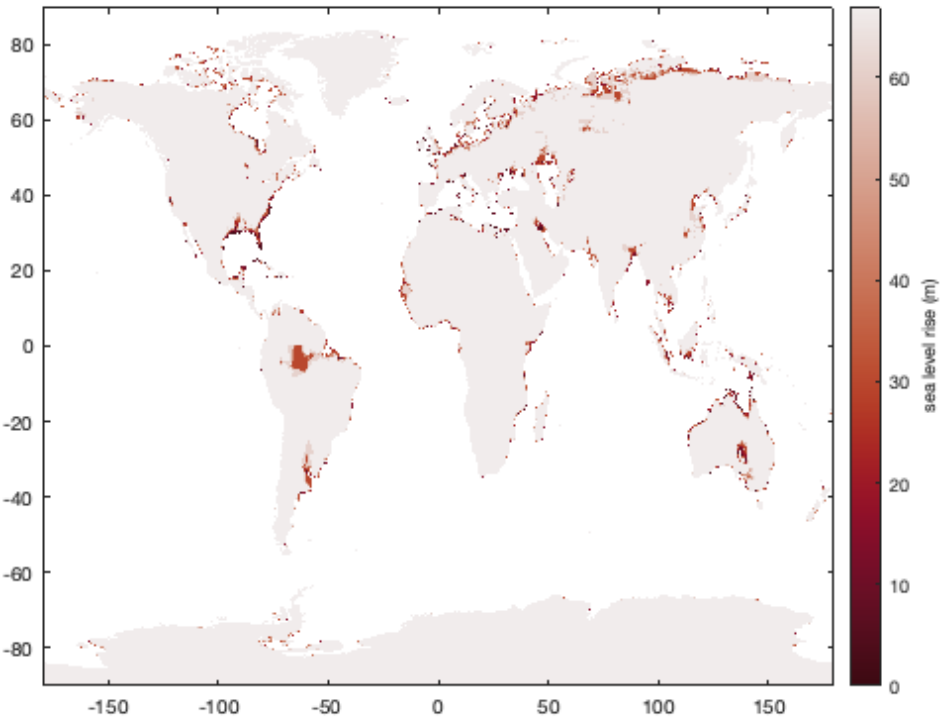
示例 4: 海平面上升的影响

让我们快速看一下可能受到海平面上升影响的地方。因为我们已经在示例 3 中加载了全分辨率数据集，所以我们将只使用该数据集，除非我们首先将所有海洋值设置为 NaN。所以把所有小于或等于零的都变成 NaN：

```
Z(Z<=0) = NaN;
```

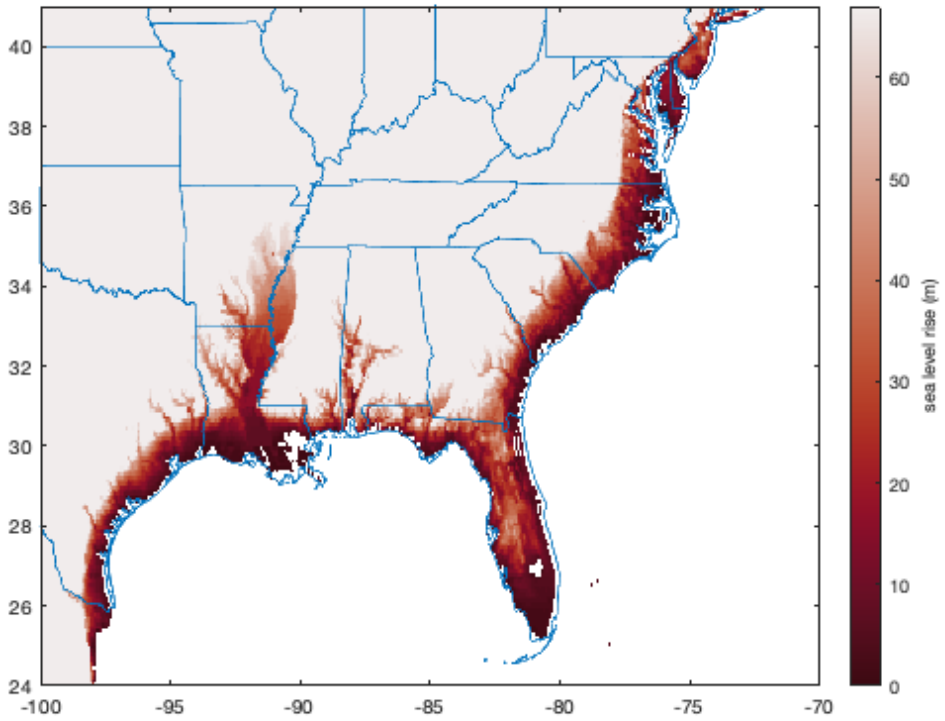
格陵兰岛和南极洲的所有冰都有可能使全球海平面上升约 67 米。当然，这种情况短期内不太可能出现，但让我们看看这样一个事件的影响。我们将不得不做出一个粗略的（不正确的）假设，即海平面将均匀分布在当今的海洋周围，但这只是一个开始。
为了研究全球海平面增加约 67 米的影响，绘制掩膜过的地形图，并将颜色轴限制从 0 到 67 米：

```
figure

imagescn(lon,lat,Z)

caxis([0 67])

cb = colorbar;

ylabel(cb,'sea level rise (m)')
cmocean -amp %设置颜色图
```



按照一阶近似，如果冰原完全崩塌，任何红色的地方都有可能被淹没。深红色意味着更脆弱。让我们放大东海岸。使用 borders 函数绘制状态边界。

```
axis([-100 -70 24 41])

borders('states')
```



关于锯齿的注记

ETOPO5 数据集以 5 分（或 1/12 度）的分辨率提供，但 `topo_interp` 函数在插值之前*不*进行任何消除锯齿。为了满足 Nyquist 要求以防止锯齿，您应该在技术上对网格点进行插值，间距至少为基础数据集 5 分分辨率的 2 倍。对于全局数据集来说，这将是一个非常密集的网格，因此您可能希望接受一点潜在的锯齿。或者可以加载原始数据，使用 `imresize` 减少网格（执行抗锯齿时）并进行插值。这取决于你。

作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

island 文档（参见前文）

dist2coast 文档

dist2coast 确定从任何地理位置到最近海岸线的距离。
另请参见: island 和 topo_interp.

语法

```
di = dist2coast(lati,loni)
```

说明

di = dist2coast(lati,loni) 给出从地理定位 lati,loni 到最近的海岸线的大圆距离 di（以公里为单位）。 对于陆地上的点，di 是到最近海洋的距离。 对于海洋位置，di 是到最近陆地的距离。

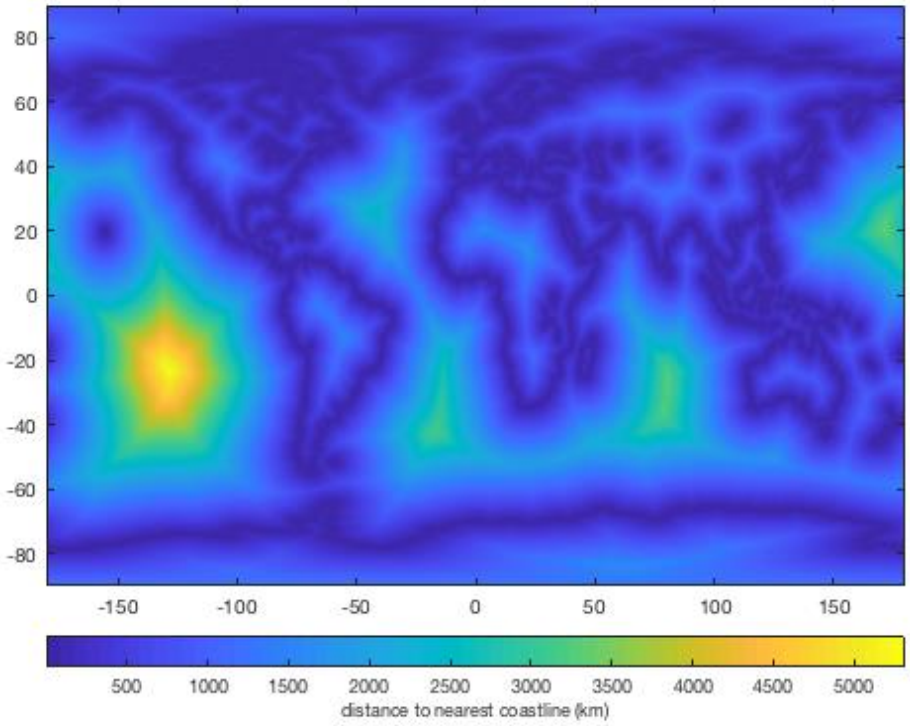
示例 1:全球网格

举个简单的例子，使用 cdtgrid 创建一个由纬度、经度点组成的四分之一度全球网格，并确定每个点离最近的海岸线有多远：

```
% 0.25 度网格:
[Lat,Lon] = cdtgrid(0.25);

% 到最近海岸线的距离:
D = dist2coast(Lat,Lon);
```

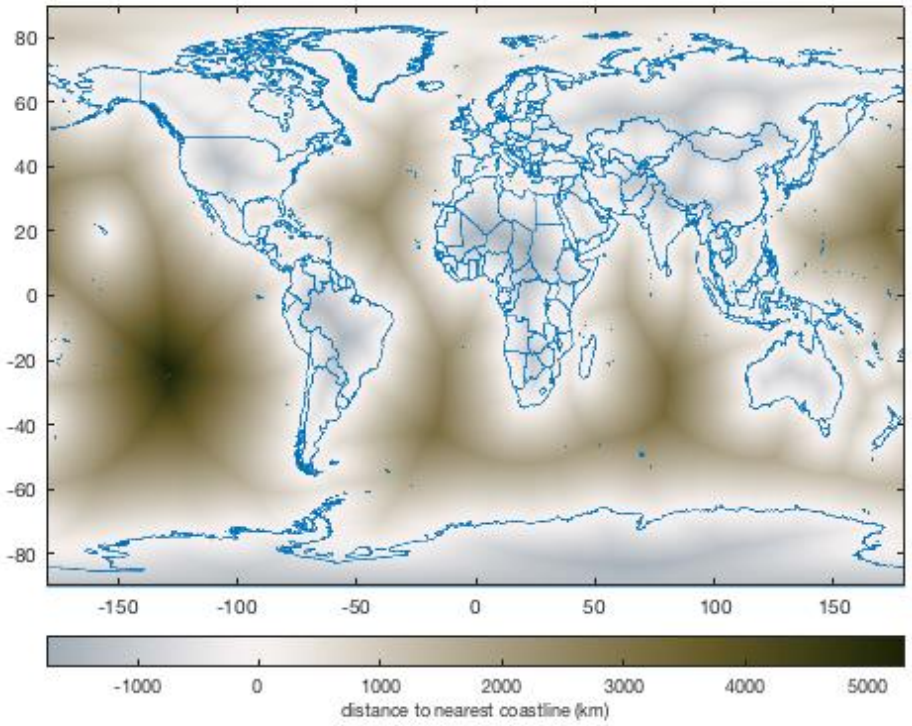
使用 imagescn 绘制到最近海岸的距离：:

```
imagescn(Lon,Lat,D)

cb = colorbar('location','southoutside');
xlabel(cb,'distance to nearest coastline (km)')
```



在上图中，您可能会注意到的第一件事是太平洋中的某个地方似乎距离最近的陆地有 5000 多公里。 那其实是假的。 （有关这方面的更多详细信息，请参阅下面的**限制**部分。）
您会注意到的第二件事是，无论是在陆地上还是在海上，与海岸的距离都是正数。 如果您愿意，可以使用 `island` 函数来确定哪些单元网格是陆地，哪些是海洋，然后将陆地单元网格设置为负值，如下所示：
（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
land = island(Lat,Lon);



% 让陆地距离是负值:
D(land) = -D(land);

figure
imagescn(Lon,Lat,D)
cb = colorbar('location','southoutside');
xlabel(cb,'distance to nearest coastline (km)')
cmocean('diff','pivot')
borders
```

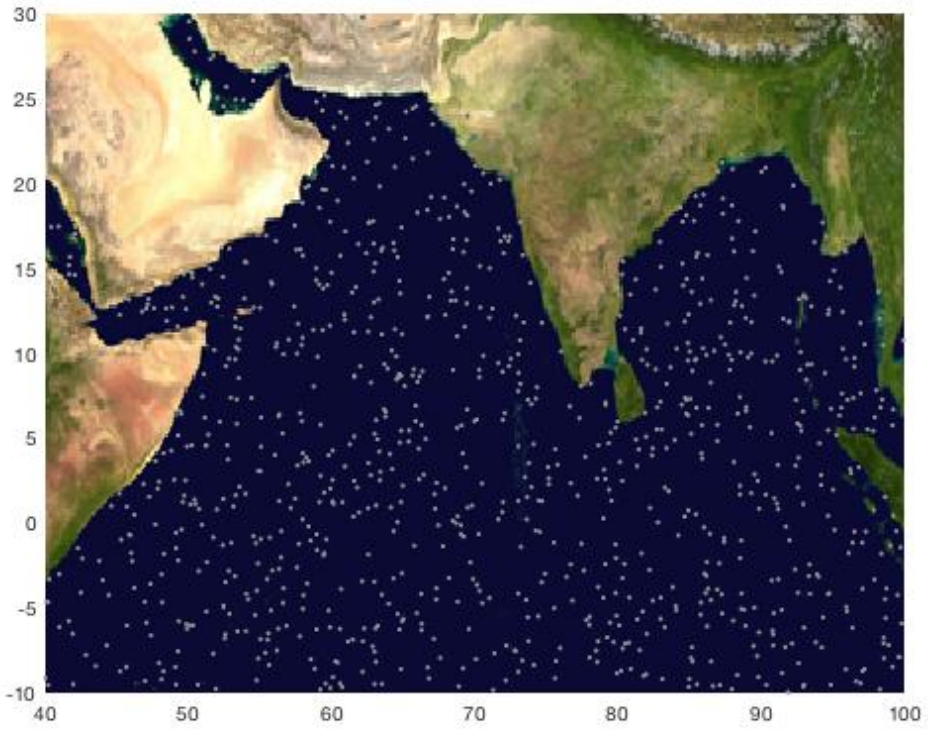


上面我们将颜色图设置为 `cmocean` *tarn* (Thyng et al., 2016) 并使其围绕零值“旋转”。还使用 `borders` 函数添加了政区边界。

示例 2: 海洋浮标

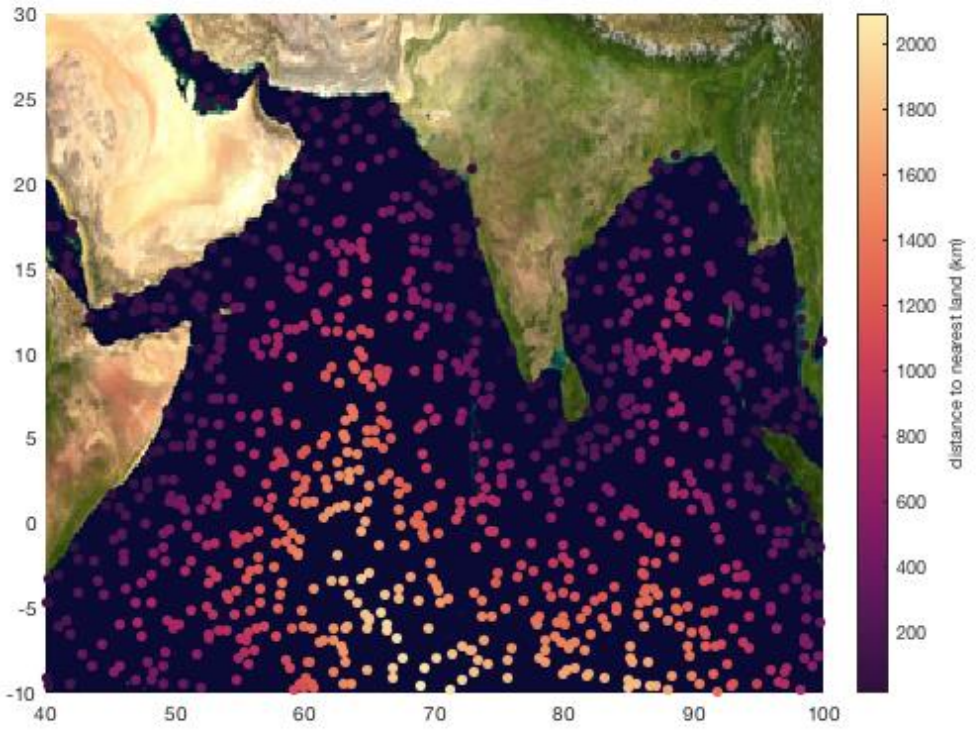
示例 1 着眼于全球，但现在让我们考虑分散数据的情况。 例如，Argo 浮标在北印度洋。 以下是一些可能是 Argo 浮标的随机数据点：

```
% 印度洋周围 1500 个随机点:
lat = 40*rand(1500,1)-10;
lon = 60*rand(1500,1)+40;

% 将此随机数据集限制为海洋点:
land = island(lat,lon);
lat(land) = [];
lon(land) = [];

% 绘制随机数据点:
figure
plot(lon,lat,'.','color',rgb('gray'))
hold on
he = earthimage;      % 加上地球图像
uistack(he,'bottom') % 将地球图像置于点下方
```



找出每个浮标离陆地有多远，并将距离绘制为分散的数据点：

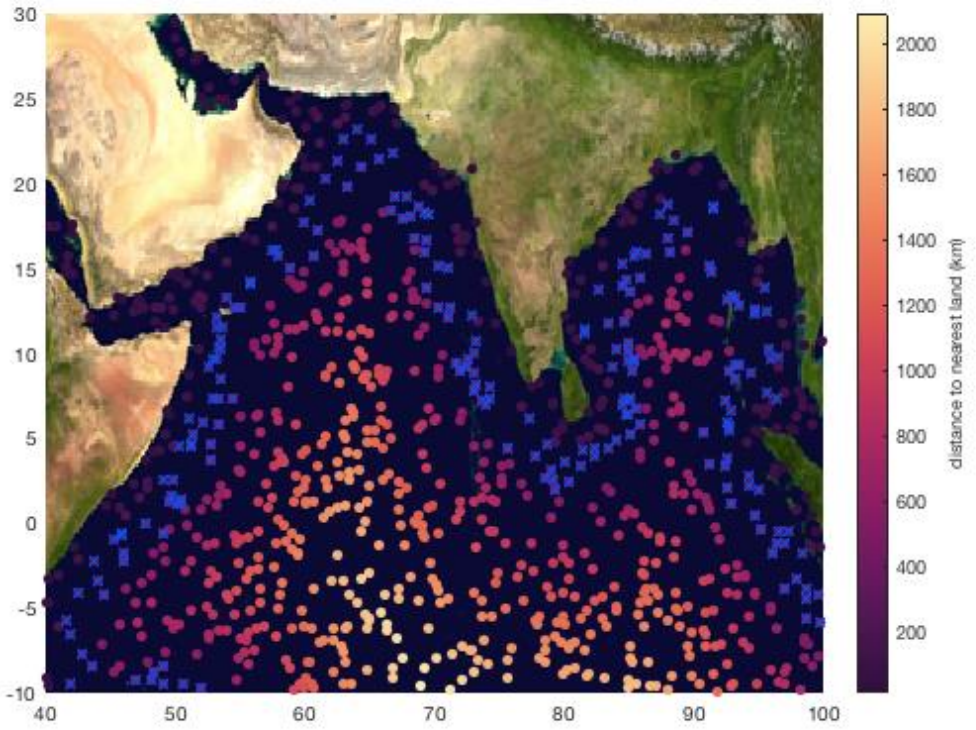
```
d = dist2coast(lat,lon);



scatter(lon,lat,30,d,'filled')

cb = colorbar;
ylabel(cb,'distance to nearest land (km)')

cmocean -matter % 设置颜色图
```



基于到陆地距离的掩膜

假设您只对距离海岸 200 到 500 公里的数据感兴趣。 获取与这些点对应的索引并使用 rgb 将它们绘制为电蓝色的 x 标记：

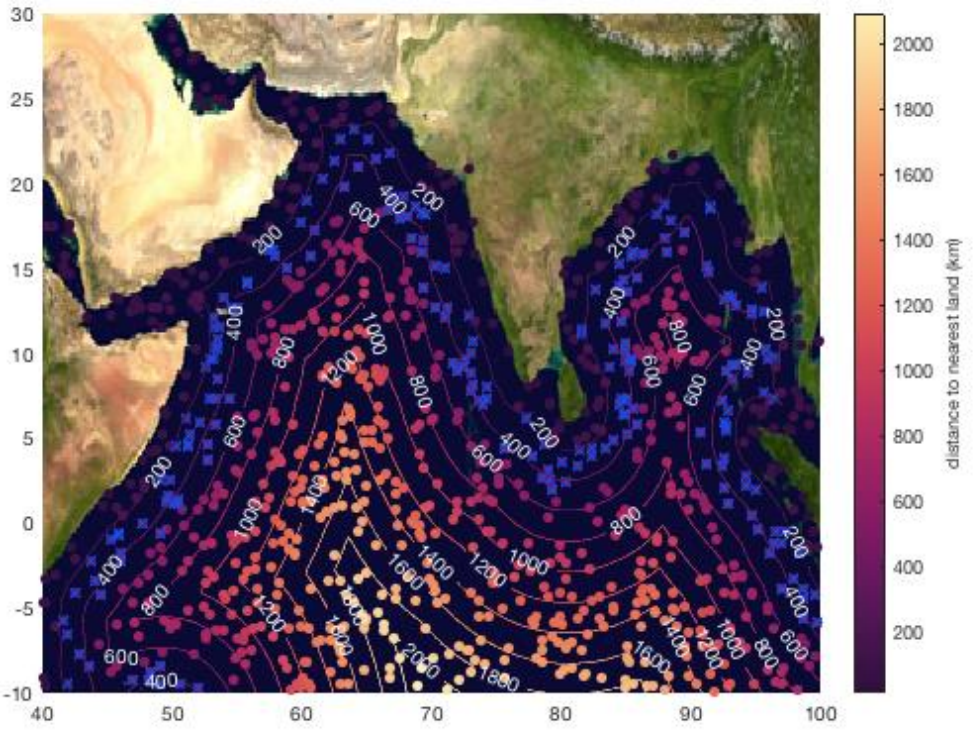
```
ind = d>200 & d<500;



plot(lon(ind),lat(ind),'x','color',rgb('electric blue'))
```



等值线

要制作离海岸距离的等高线，使用 meshgrid 或 cdtgrid 制作网格，获取每个网格点到最近海岸线的距离，并使用 island 屏蔽陆地网格单元：

```
% Make a grid 做个网格:
[Lon,Lat] = meshgrid(40:0.1:100,-10:0.1:30);

% 获取从网格点到海岸的距离:
D = dist2coast(Lat,Lon);

% 屏蔽陆地:
D(island(Lat,Lon)) = NaN;

% 绘制等值线和标签:
[C,h] = contour(Lon,Lat,D);
clabel(C,h,'color','w')
```



限制

在示例 1 中，我们首先看到的是太平洋中某个距离陆地 5000 多公里的地方。实际上，离陆地最远的点是 Point Nemo，位于 (37.58S,139.39W)，距离最近的陆地仅 2688 公里。dist2coast 与此相去甚远的原因是因为 dist2coast 使用与 island 函数相同的 1/8 度掩膜来确定陆地的存在。任何不足以出现在 1/8 度数据集中的土地都不会用 island 或 dist2coast 表示。因此，该函数可能最适合不受小岛影响的中尺度过程。

同样值得注意的是：dist2coast 函数认为北美五大湖和其他大型淡水体不是陆地，而是水。因此，北美离水最远的地方是怀俄明州，而不是加拿大有很多大湖的地方。

作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

图形

- rgb 按名称提供常见和不常见颜色的 RGB 值。
- cmocean 提供了 Thyng et al., 2016 提供的感知统一的颜色图。
- newcolorbar 允许在同一个坐标轴上有多个颜色图和颜色条。
- cbarrow 在颜色条上放置三角形端元以指示数据值存在于颜色条中显示的值的范围之外。
- cbdate 将颜色条刻度格式化为日期字符串。
- hline 在绘图上创建水平线。
- vline 在绘图上创建垂直线。
- hfill 在绘图上创建水平填充区域。
- vfill 在绘图上创建垂直填充区域。
- ntitle 将标题放置在图内而不是顶部。
- gif 可轻松创建 .gif 动画。

rgb 文档

根据 XKCD 的 Randall Munroe 令人印象深刻的彻底调查的结果，该函数返回您能想到的几乎任何颜色名称的 RGB 颜色三元组。与 Matlab 语法保持一致，RGB 值 0 缩放到 1。如果您拼错了一种颜色或数据库中没有您想要的颜色，rgb 将提供其他类似拼写颜色的建议。

语法

```
RGB = rgb('Color Name')
RGB = rgb('Color Name 1','Color Name 2',...,'Color Name N')
RGB = rgb({'Color Name 1','Color Name 2',...,'Color Name N'})
```

说明

`RGB = rgb('Color Name')` 返回由 'Color Name' 描述的颜色的 RGB 三元组。
`RGB = rgb('Color Name 1','Color Name 2',...,'Color Name N')` 返回一个 *N* x 3 矩阵，其中包含每个颜色名称的 RGB 三元组。
`RGB = rgb({'Color Name 1','Color Name 2',...,'Color Name N'})` 接受颜色名称列表作为字符数组。

颜色参考图

要在绘图前查看颜色选项，您可以参考此处的 RGB 图表，但如果您正在考虑特定颜色，请尝试 rgb 函数，它可能具有您想要的 RGB 值。

示例 1: 单色

获取黄绿色的 RGB 三元组:

```
rgb('chartreuse')
```

```
ans =

    0.7569    0.9725    0.0392
```

示例 2:多色

获取多种颜色的 RGB 三元组:

```
rgb('wintergreen','sunflower yellow','sapphire')
```

```
ans =

    0.1255    0.9765    0.5255

    1.0000    0.8549    0.0118

    0.1294    0.2196    0.6706
```

在计算机显示器上感知颜色的方式不一定是官方定义颜色的方式。 如果我们将 [1 0 0] 的 RGB 值视为“红色”，将 [0 1 0] 视为“绿色”，将 [0 0 1] 视为“蓝色”，则 rgb('red','green','blue') 会看起来像一个单位矩阵，而不是我们有这个：

```
rgb('red','green','blue')
```

```
ans =

    0.8980         0         0

    0.0824    0.6902    0.1020

    0.0118    0.2627    0.8745
```

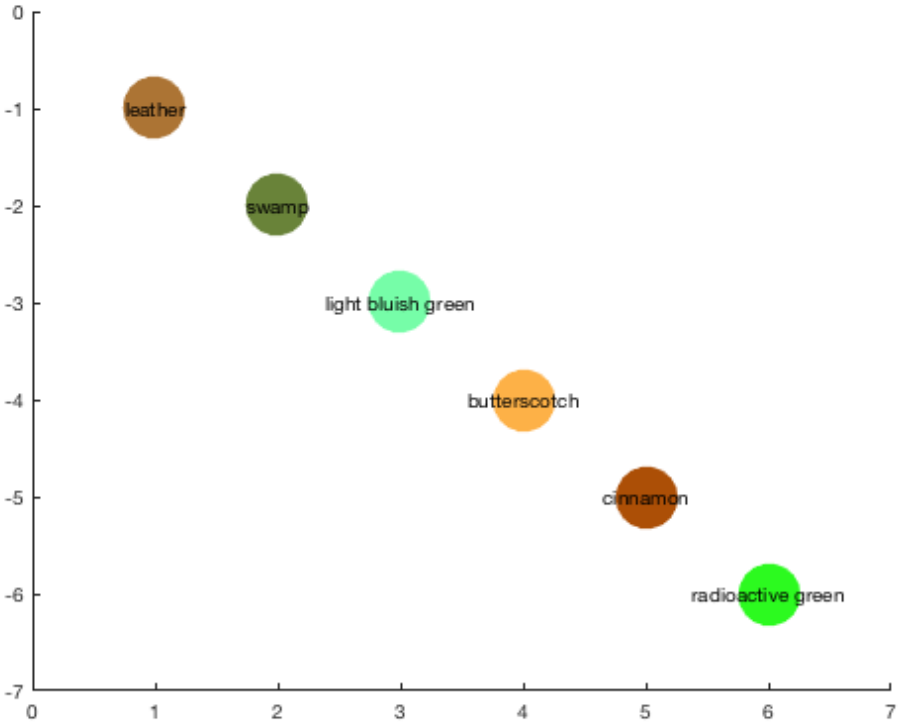
您还可以将颜色名称作为元胞数组输入：

```
myColors = {'leather','swamp','light bluish green','butterscotch','cinnamon','radioactive green'};
rgb_vals = rgb(myColors)
```

```
rgb_vals =

    0.6745    0.4549    0.2039

    0.4118    0.5137    0.2235

    0.4627    0.9922    0.6588

    0.9922    0.6941    0.2784

    0.6745    0.3098    0.0235

    0.1725    0.9804    0.1216
```

然后可以轻松地绘制我们的 rgb_vals:

```
x = 1:length(myColors);

y = -x;

scatter(x,y,1e3,rgb_vals,'filled')
text(x,y,myColors,'horizontalalignment','center')
axis([min(x)-1 max(x)+1 min(y)-1 max(y)+1])
```



作者简介

该函数由德克萨斯大学奥斯汀分校地球物理研究所的 Chad A. Greene 编写。我不对颜色数据承担任何责任；感谢 Randall Munroe 对调查的分析和设计。

cmocean 文档

cmocean 函数返回由 Kristen Thyng 生成的感知统一的颜色图。有关 cmocean 项目的详细说明，请访问 http://matplotlib.org/cmocean。

如果您发现有机会出于任何原因引用这些颜色图，或者您只是想要一些不错的海滩读物，请查看《*Oceanography*》期刊以下论文：

Kristen M. Thyng, Chad A. Greene, Robert D. Hetland, Heather M. Zimmerle, and Steven F. DiMarco (2016). True colors of oceanography: Guidelines for effective and accurate colormap selection. *Oceanography*, 29(3), 10. doi:10.5670/oceanog.2016.66

另请参见: rgb.

语法

```
cmocean
cmap = cmocean('ColormapName')
cmap = cmocean('-ColormapName')
cmap = cmocean(...,NLevels)
cmap = cmocean(...,'pivot',PivotValue)
cmap = cmocean(...,'negative')
cmocean(...)
```

说明

cmocean 没有任何输入时就显示颜色图的选项。

cmap = cmocean('ColormapName') 返回 256x3 颜色图。 ColormapName 可以是以下任何一项：

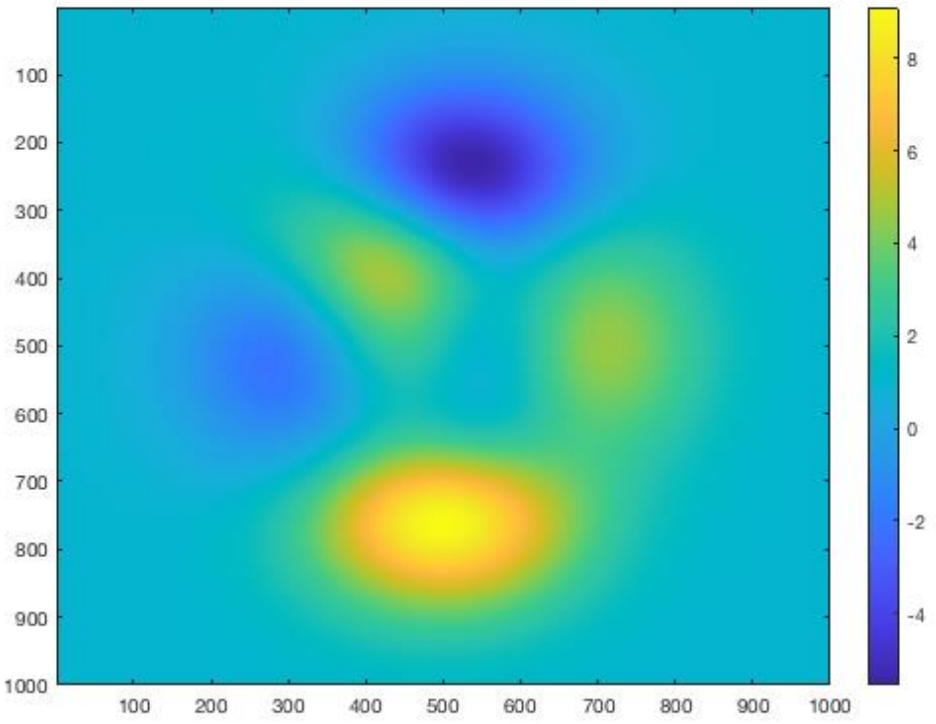
cmap = cmocean('-ColormapName') 在任何 ColormapName 之前的减号会翻转颜色图的顺序。
cmap = cmocean(...,NLevels) 指定颜色图的级别。 默认值为 256。
cmap = cmocean(...,'pivot',PivotValue) 将发散颜色图居中，以便颜色发散点对应于指定值，并使用当前 caxis 限制设置最大范围。 如果未设置 PivotValue，则假定为 0。 此函数的早期版本使用'zero' 作为'pivot'的语法，0 和旧的语法并且仍然支持。
cmap = cmocean(...,'negative') 反转颜色图的亮度配置文件。如果默认的发散白点在白色背景中丢失，那这对于发散颜色图尤其有用。
cmocean(...) 没有任何输出的时为当前坐标轴设置当前颜色图。

示例

使用此示例图：

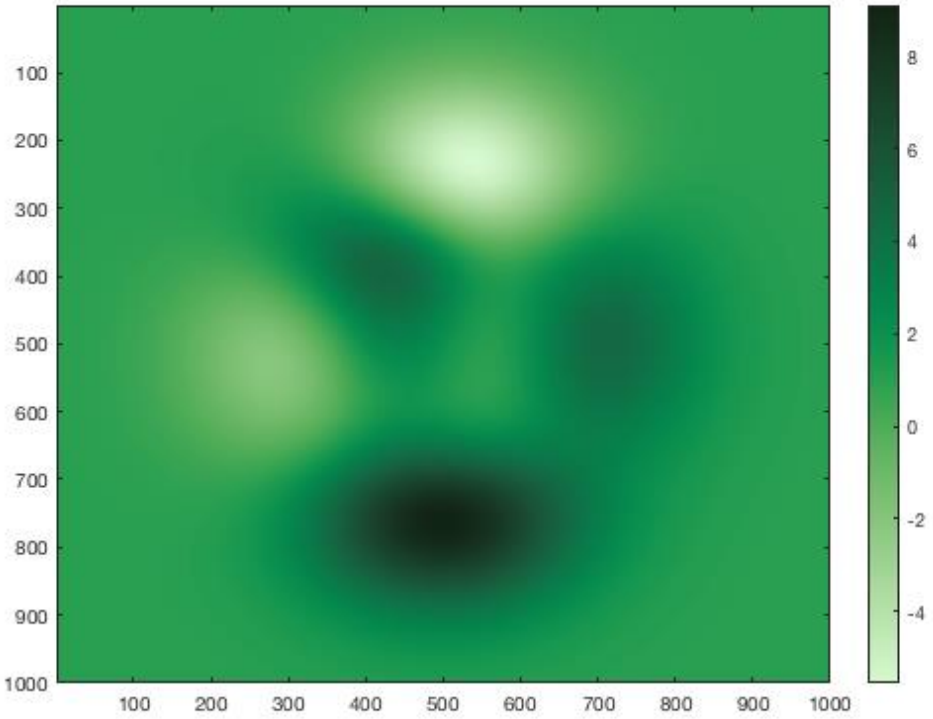
```
imagesc(peaks(1000)+1)
```

```
colorbar
```



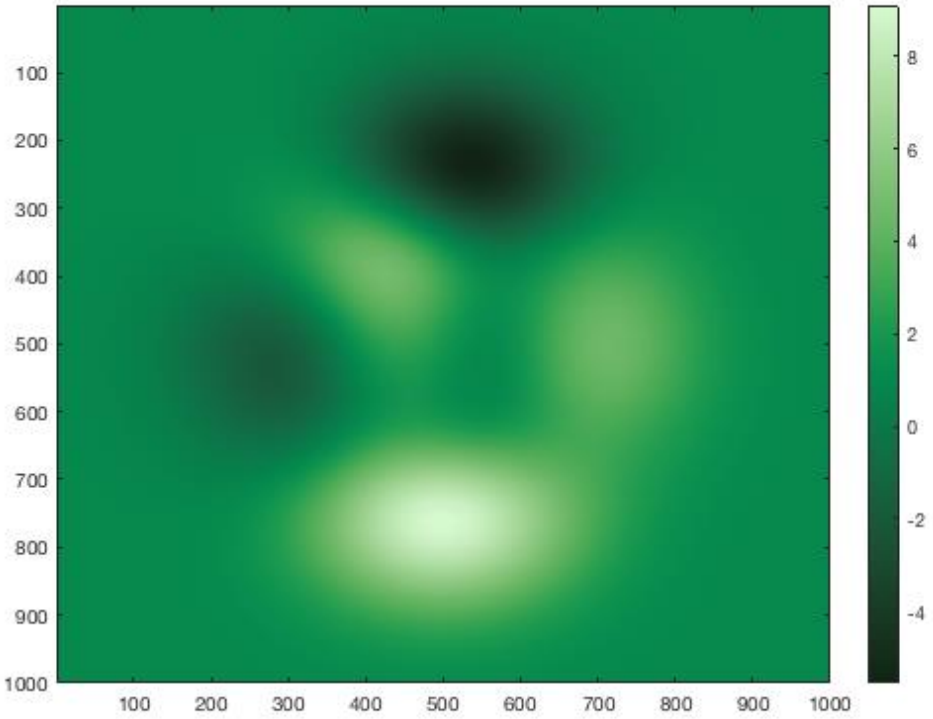
设置颜色图为'algae':

```
cmocean('algae')
```



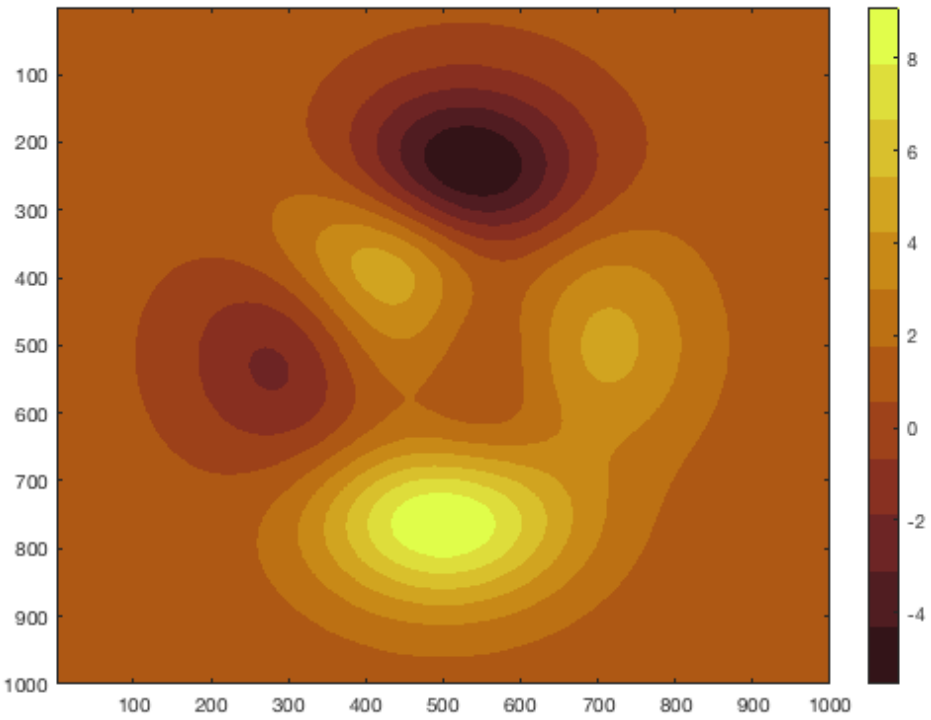
与以上相同，但是是倒置的 algae 颜色图:

```
cmocean('-algae')
```



将颜色图设置为 12 级 'solar':

```
cmocean('solar',12)
```



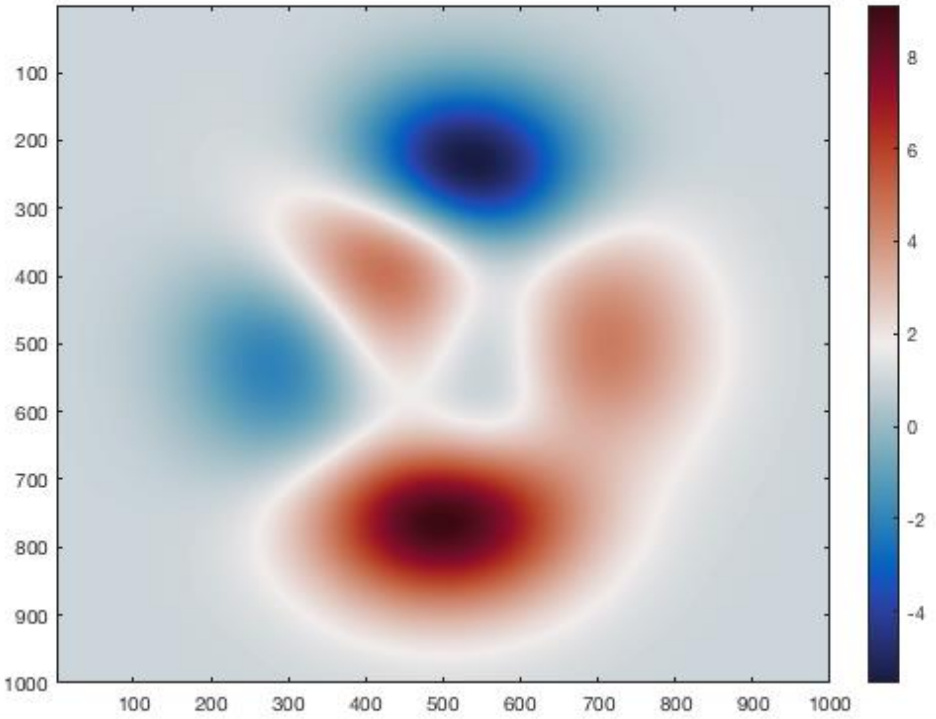
获取 5 级 thermal 色图的 RGB 值：

```
RGB = cmocean('thermal',5)
```

```
RGB =

    0.0156    0.1382    0.2018

    0.3366    0.2317    0.6123

    0.6893    0.3727    0.5097

    0.9772    0.5740    0.2578

    0.9090    0.9822    0.3555
```

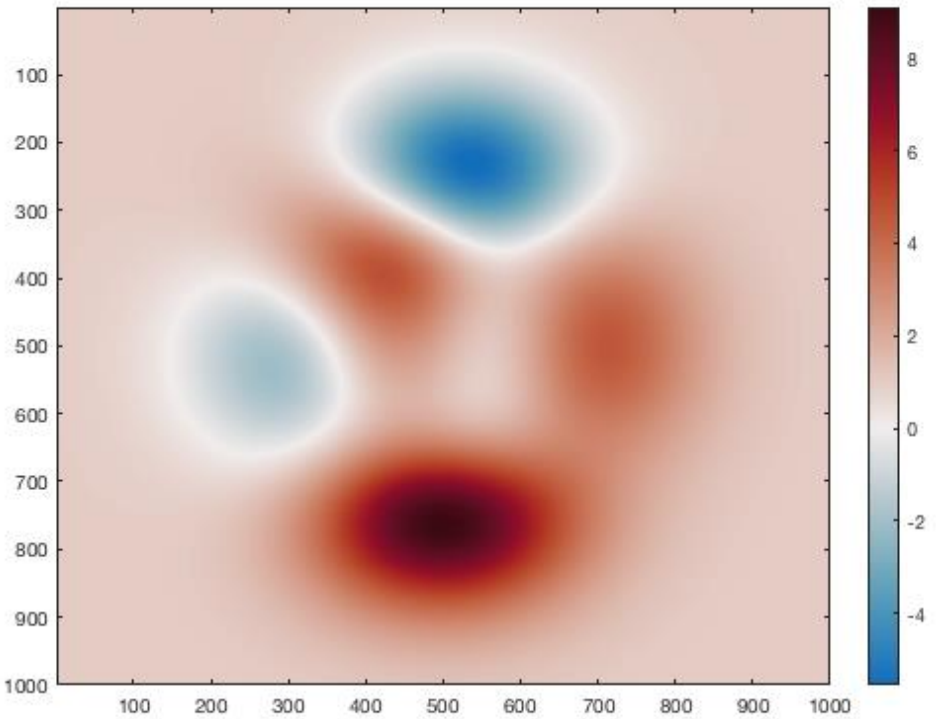
其中一些值低于零，其他值高于零。 如果此数据集表示异常，则发散颜色图可能更合适：

```
cmocean('balance')
```



颜色轴 1.7776 的中心值不太可能是数据偏离的感兴趣值。如果您想使用当前颜色轴限制将颜色图居中于零，只需包含 'pivot' 选项：

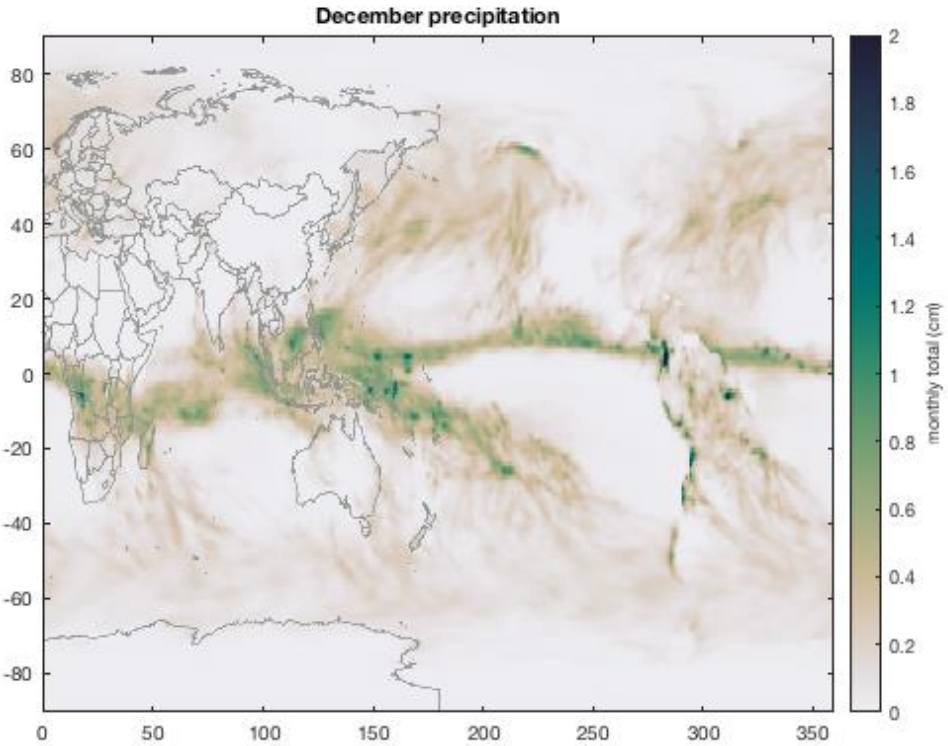
```
cmocean('balance','pivot',0)
```



绝对量与异常

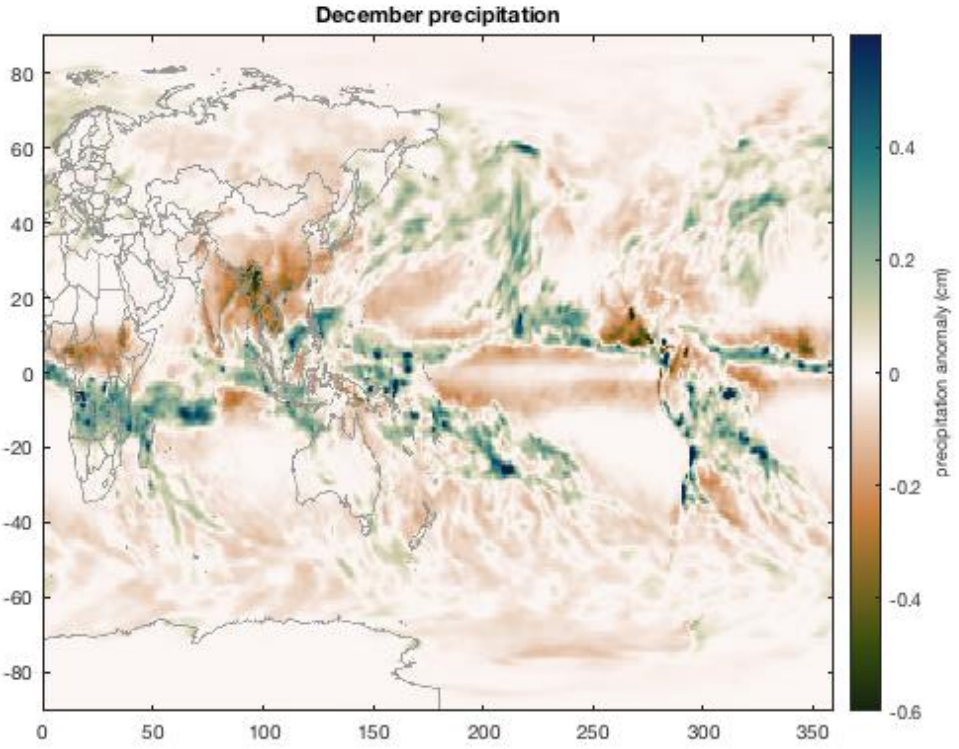
当绘制线性量（例如一个月的总降水量）时，使用从亮到暗或从暗到亮的线性颜色图。以下是 2017 年 12 月的降水量，用 *rain* 颜色图绘制：（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
filename = 'ERA_Interim_2017.nc';
lat = ncread(filename,'latitude');
lon = ncread(filename,'longitude');
tp = ncread(filename,'tp'); % 总降水量

figure
h = imagescn(lon,lat,tp(:,:,12)'*100);
title 'December precipitation'
cb = colorbar;
ylabel(cb,'monthly total (cm)')
cmocean rain

hold on
borders('countries','color',rgb('gray'))
caxis([0 2])
```



然而，异常情况需要一种不同的方法。使用发散颜色图显示异常的空间模式，在值异常高、低或接近平衡的区域之间提供清晰的划分。这是使用发散的 *tarn* 颜色图，显示了 12 月降水量与年平均值的关系：（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
Precip_anomaly = tp(:,:,12) - mean(tp,3);



figure

imagescn(lon,lat,Precip_anomaly'*100)

title 'December precipitation'
cb = colorbar;
ylabel(cb,'precipitation anomaly (cm)')
cmocean tarn
caxis([-1 1]*0.6)

hold on
borders('countries','color',rgb('gray'))
```

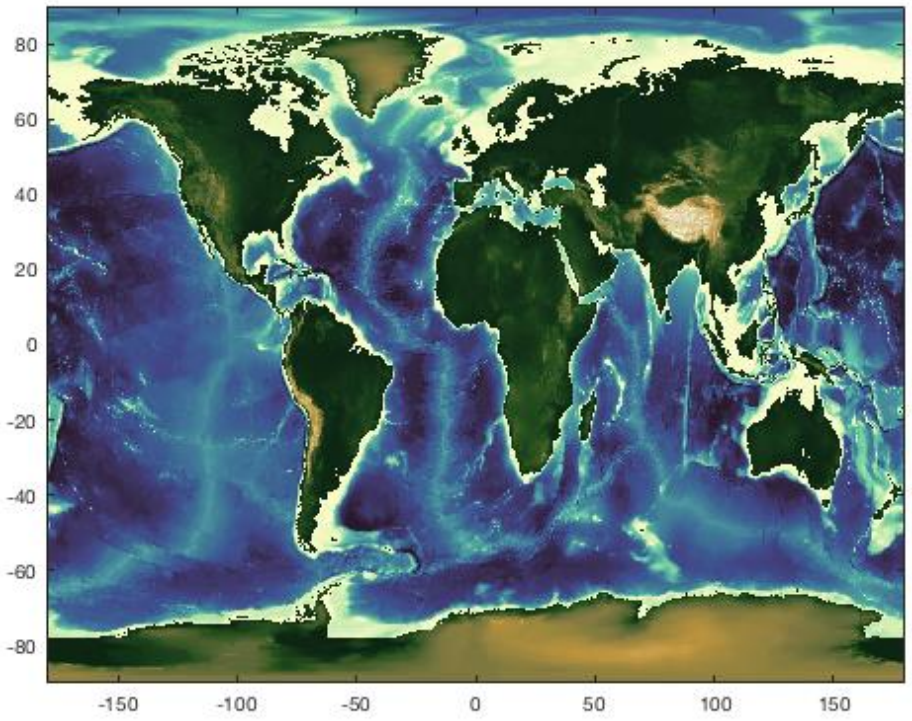


地形

地形是一个特例，因为清楚地区分陆地和海洋通常很重要，但我们仍然希望清楚地感知到的颜色和海拔之间存在线性关系。典型的发散彩色地图（如 *balance* 或 *curl* ）可能会将视线引向海岸线的大致方向，但无法提供陆地和海洋之间的清晰定义，因此 cmocean 颜色图包括专门为地形设计的“topo”。

```
[Lat,Lon] = cdtgrid(0.25);
Z = topo_interp(Lat,Lon);
```

```
figure
imagescn(Lon,Lat,Z)

caxis([-1 1]*6000)
cmocean topo
```

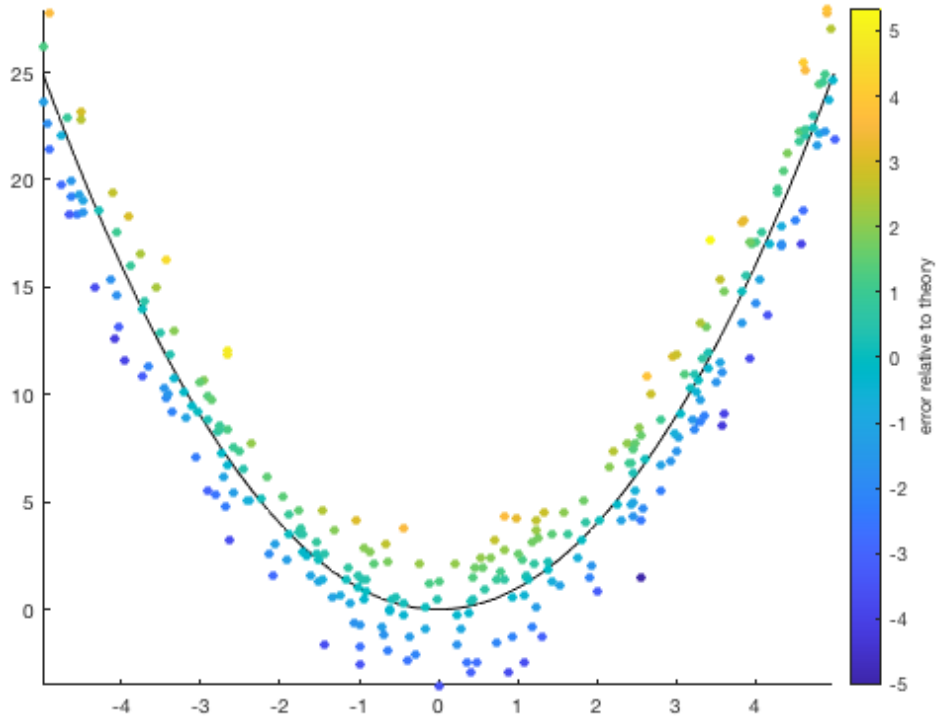


反向颜色图

我最近遇到的情况是，我在白色背景上绘制了分散数据的散点图，但 cmocean('balance')将最重要的数据点变成了白色，几乎看不见。情况如下：

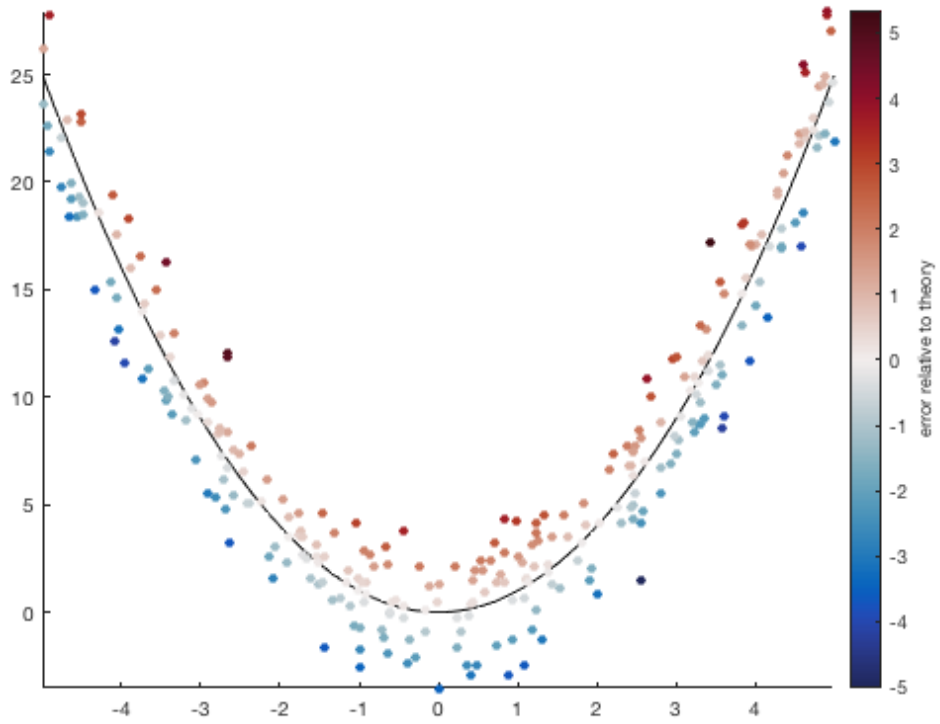
```
% 一些添加早点的样本数据:
x = 10*rand(300,1)-5;
noise = 2*randn(size(x));
y = x.^2+noise;

% 一根理论上完美的 x^2 线:
x_theoretical = linspace(min(x),max(x),50);
y_theoretical = x_theoretical.^2;

% 绘制数据:
figure
plot(x_theoretical,y_theoretical,'k-')
hold on
```

```
scatter(x,y,25,noise,'filled')
cb = colorbar;
ylabel(cb,'error relative to theory')
box off
axis tight
```



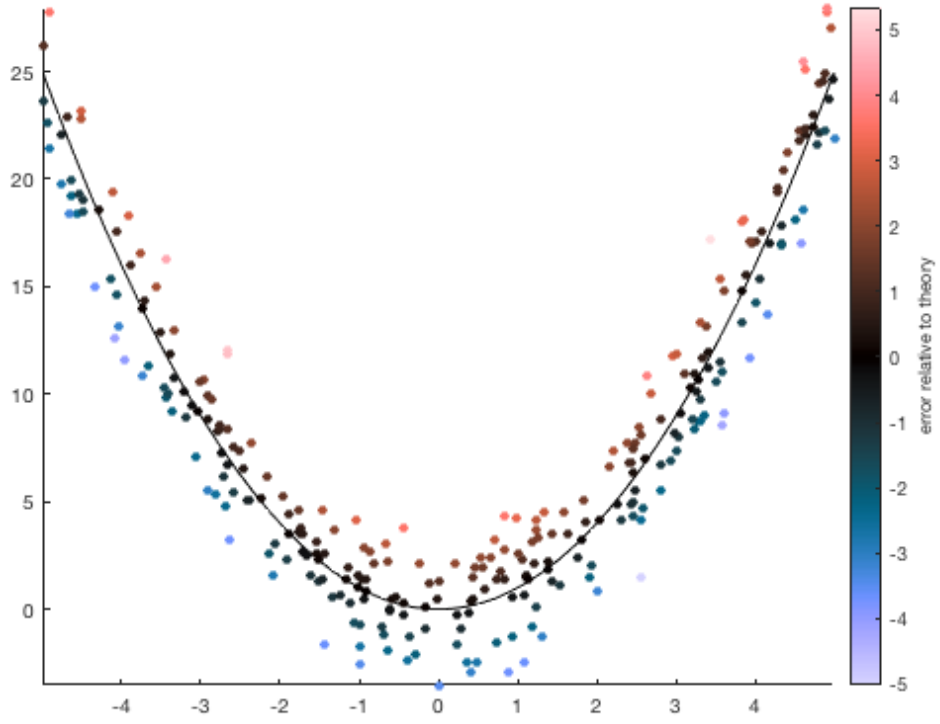
我想展示我的早点数据离完美的理论 x 平方线有多远，因此发散的 cmocean('balance') 图似乎是合适的：

```
cmocean('balance','pivot',0)
```



但在上图中，注意力从理论线转移到了深红色和深蓝色异常值。在这种情况下，可以优选反向颜色图：

```
cmocean('balance','pivot',0,'negative')
```



作者简介

该函数由德克萨斯大学奥斯汀分校地球物理研究所的 Chad A. Greene 于 2016 年 6 月编写，使用德克萨斯 A&M 大学海洋系 Kristen Thyng 创建的颜色图。有关 cmocean 项目的更多信息，请访问 http://matplotlib.org/cmocean. 2019 年 1 月更新了新的颜色图 *rain*, *topo*, *diff* 和 *tarn*。

newcolorbar 文档

newcolorbar 允许在同一子图中使用多个颜色图和颜色条。
此功能通过创建一组新的不可见坐标轴来工作，这些坐标轴与当前坐标轴的大小和范围相匹配。这允许以这样的方式使用第二个颜色贴图，即它被视为同一轴上的两个单独的颜色图和两个颜色条。

语法

```
colorbar
colorbar(Location)
colorbar(...,PropertyName,PropertyValue)
[cb2,ax2,ax1] = colorbar(...)
```

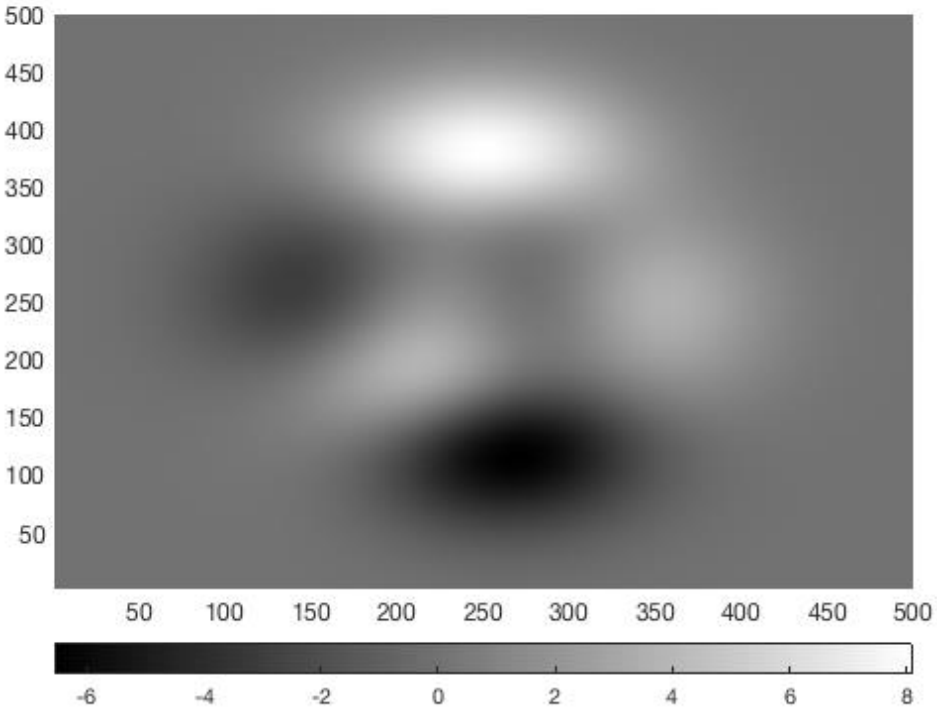
描述

colorbar 在默认（右侧）位置创建一组新的轴和一个新的颜色条。
colorbar(Location) 将新颜色条的位置指定为'North'在打印框内靠近顶部，'South'在底部，'East'在右侧，'West'在左侧，'NorthOutside'在打印框外靠近顶部，'SouthOutside'在底部，'EastOutside'在右侧外部（默认），'WestOutside'在左侧外部。
colorbar(...,PropertyName,PropertyValue) 为 colorbar 指定其他名称/值对。
[cb2,ax2,ax1] = colorbar(...) 返回新颜色条 cb2、新轴 ax2 和以前的当前轴 ax1 的句柄。

示例 1

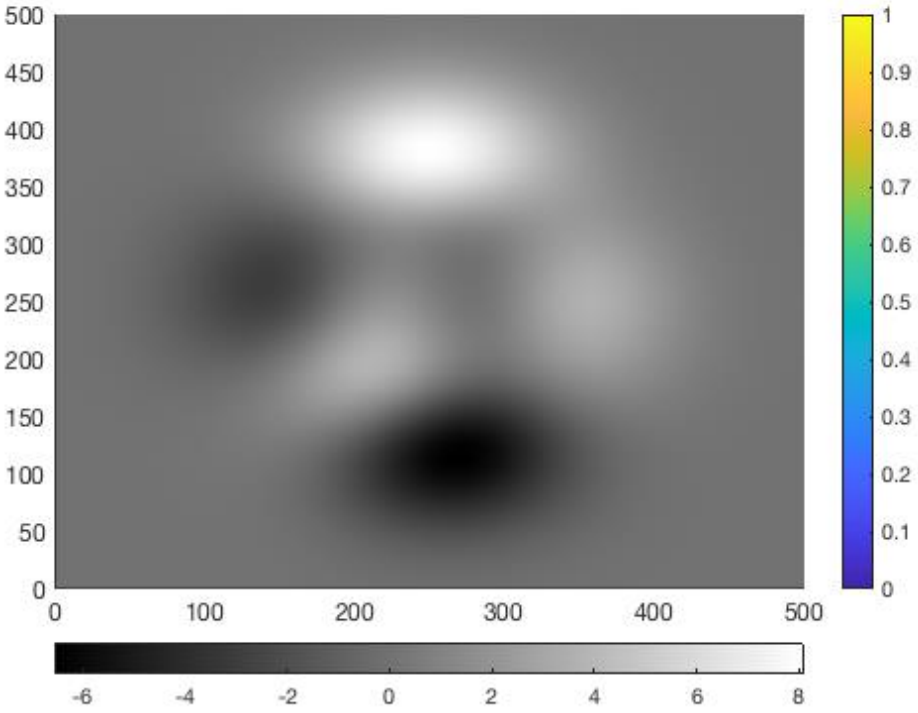
让我们在灰度网格数据集上绘制一些 parula 颜色的分散数据。从绘制网格化数据开始。在本例中，我们将使用内置的 peaks 数据集：

```
pcolor(peaks(500))

shading interp
colormap(gca,gray(256))
colorbar('southoutside')
```



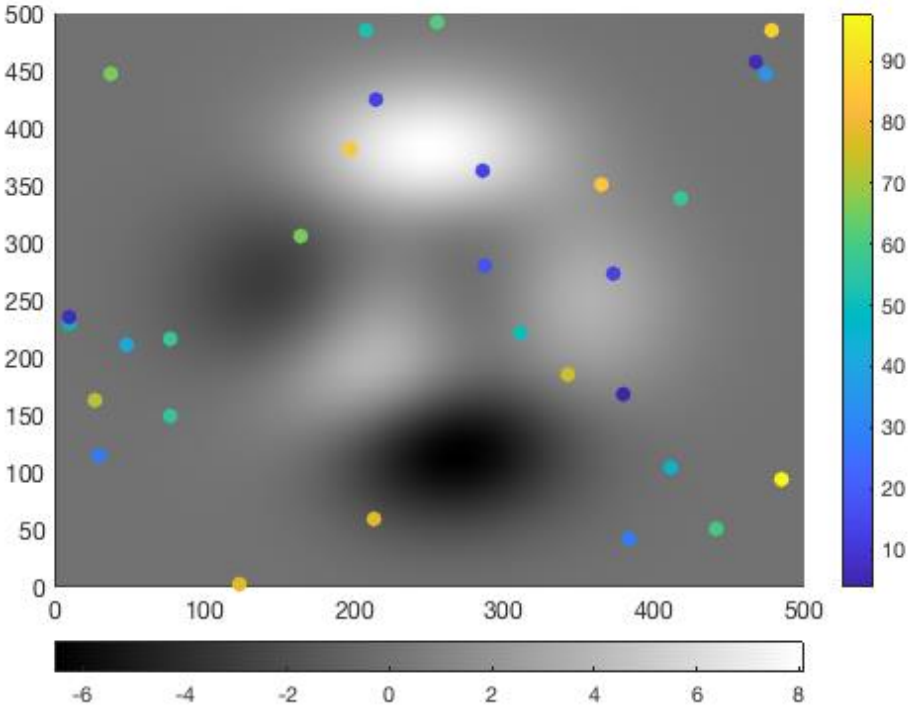
newcolorbar 函数与 Matlab 的 colorbar 函数的不同之处在于，我们必须在绘制新的颜色缩放数据集*之前*创建一个 newcolorbar 。我们可以非常简单地创建一个新的颜色条：

```
newcolorbar
```



然后绘制一些随机颜色缩放的散点数据：使用 parula 颜色图绘制散点数据集：

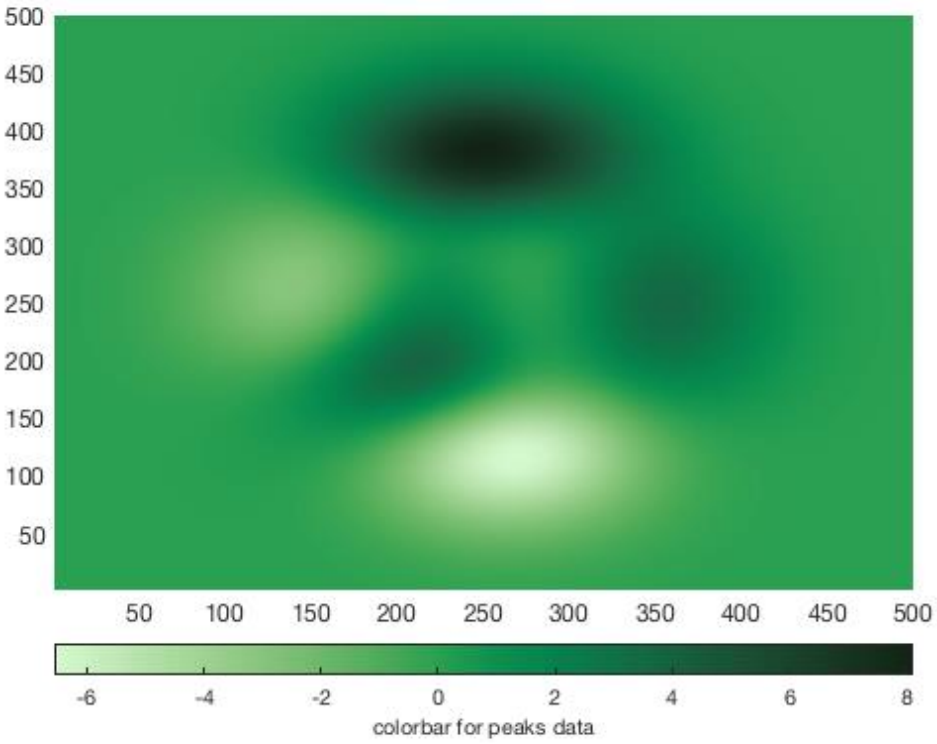
```
scatter(500*rand(30,1),500*rand(30,1),60,100*rand(30,1),'filled')
```



示例 2

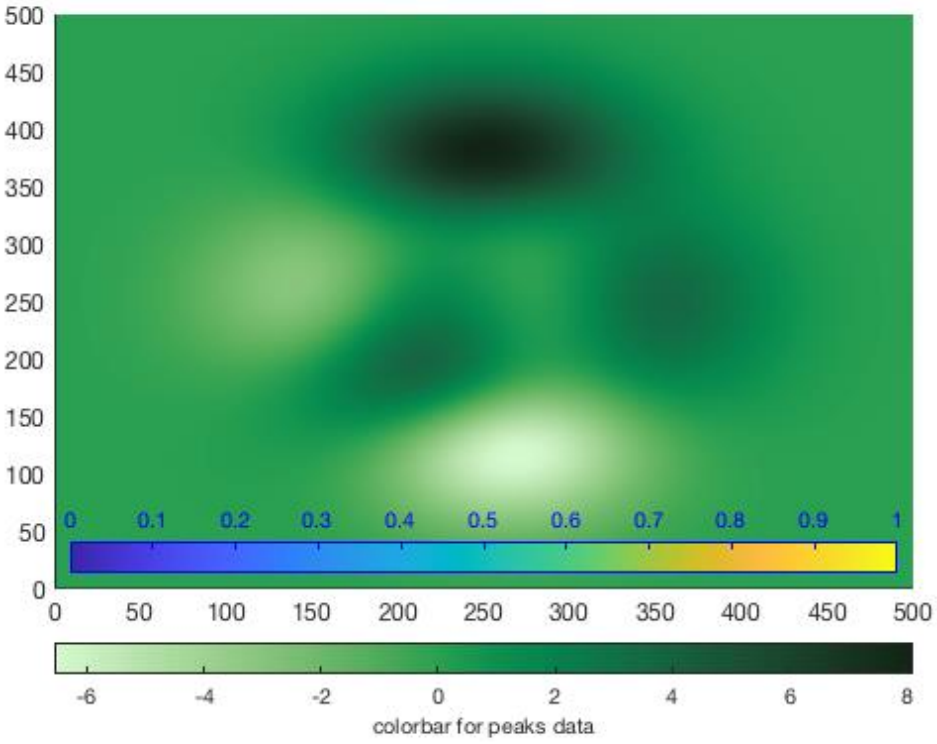
现在让我们重复示例 1，但添加一些格式。使用 cmocean 创建可爱的颜色贴图：

```
figure

pcolor(peaks(500))

shading interp
cmocean algae
cb1 = colorbar('southoutside');
xlabel(cb1,'colorbar for peaks data')
```



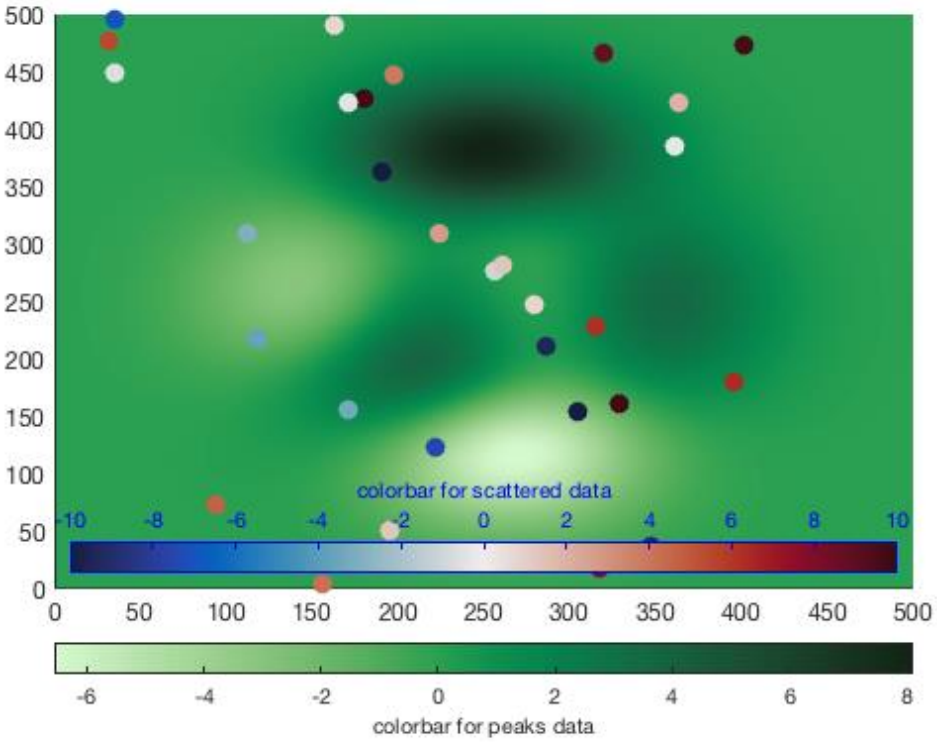
向当前轴内侧底部添加新的颜色条，并指定蓝色文本：

```
cb2 = newcolorbar('south','color','blue');
```



打印分散的数据并标记新的颜色条：

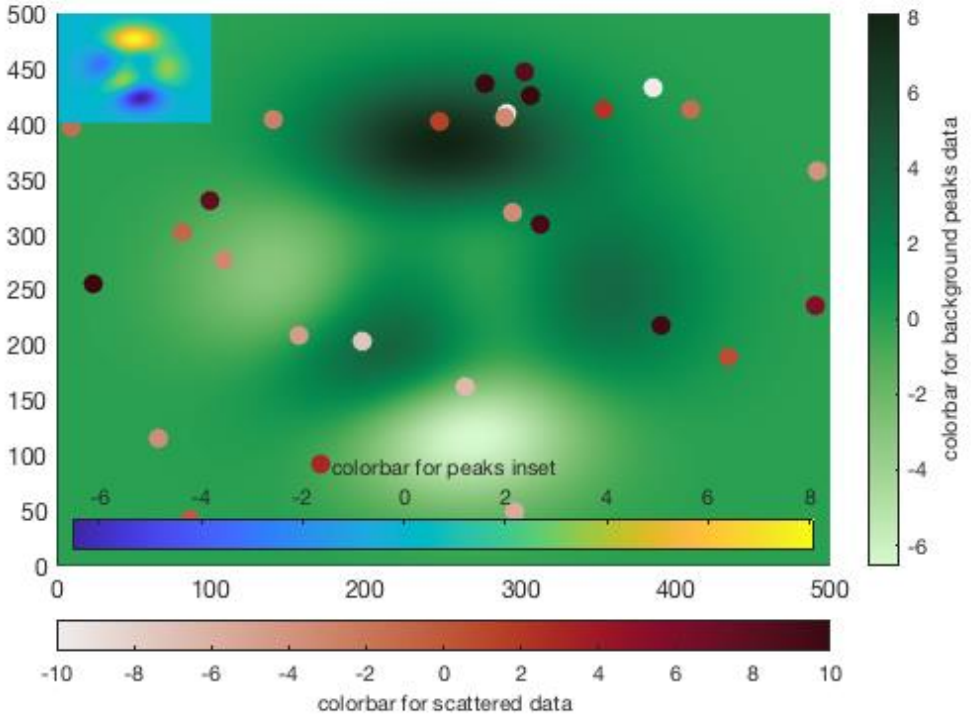
```
scatter(500*rand(30,1),500*rand(30,1),100,8*randn(30,1),'filled')
cmocean balance
caxis([-10 10]) % 设置散点颜色条的轴
xlabel(cb2,'colorbar for scattered data','color','blue')
```



示例 3: 三个颜色条

这是一个三个颜色条的示例：

```
figure

pcolor(peaks(500))

shading interp
cmocean algae
cb1 = colorbar;
ylabel(cb1,'colorbar for background peaks data')

cb2 = newcolorbar('southoutside') ;
scatter(500*rand(30,1),500*rand(30,1),100,8*randn(30,1),'filled')
cmocean amp
caxis([-10 10]) % 设置散点颜色条的轴
xlabel(cb2,'colorbar for scattered data')

cb3 = newcolorbar('south');
pcolor(1:100,401:500,peaks(100))
shading interp
xlabel(cb3,'colorbar for peaks inset')
```



作者简介

newcolorbar 函数是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 于 2015 年 8 月写的，2019 年为 Climate Data Toolbox for Matlab 而更新。

cbarrow 文档

cbarrow 函数在颜色栏上放置三角形末端，以指示数据值超出了颜色条中显示值的范围。
此功能通过在当前图形顶部创建一组轴并将填充对象放置在新轴上来工作。为此，调用 cbarrow 后编辑图形可能会导致一些小故障。因此，建议在创建绘图时最后调用 cbarrow 。

语法

```
cbarrow
cbarrow(Direction)
cbarrow('delete')
h = cbarrow(...)
```

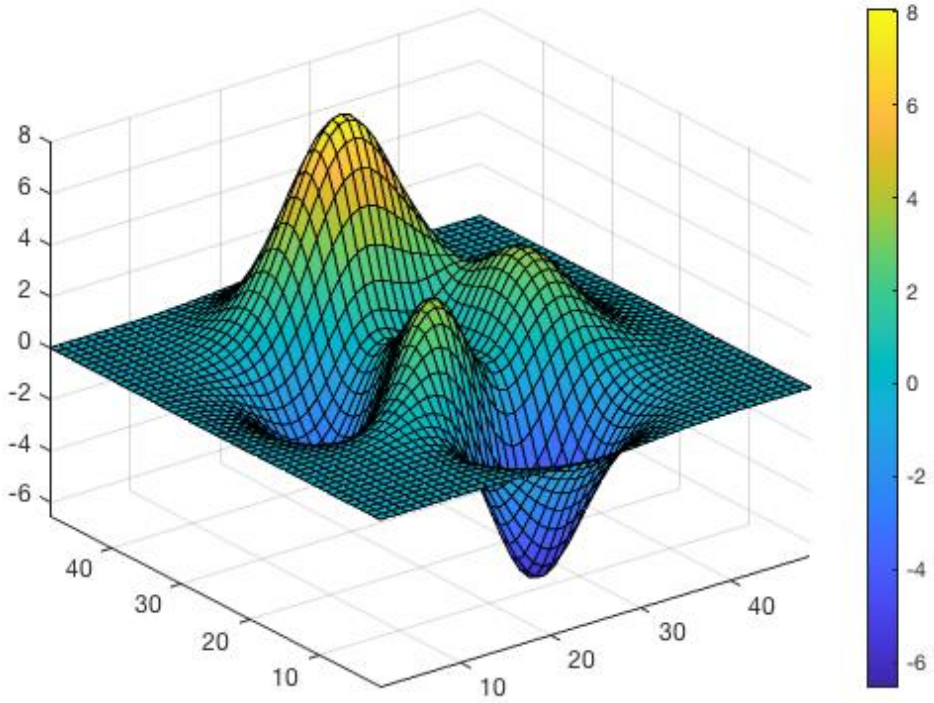
说明

cbarrow 将三角形端部构件放置在当前颜色栏的两端。
cbarrow(Direction) 指定放置颜色条结束箭头的单个方向。Direction 可以是 'up','down','right', 或'left'。
cbarrow('delete') 删除以前创建的 cbarrow 对象。
h = cbarrow(...) 返回创建 cbarrow 对象的轴的句柄。

示例 1: 两个方向

考虑样本数据的这个图：

```
surf(peaks)


axis tight
colorbar
```



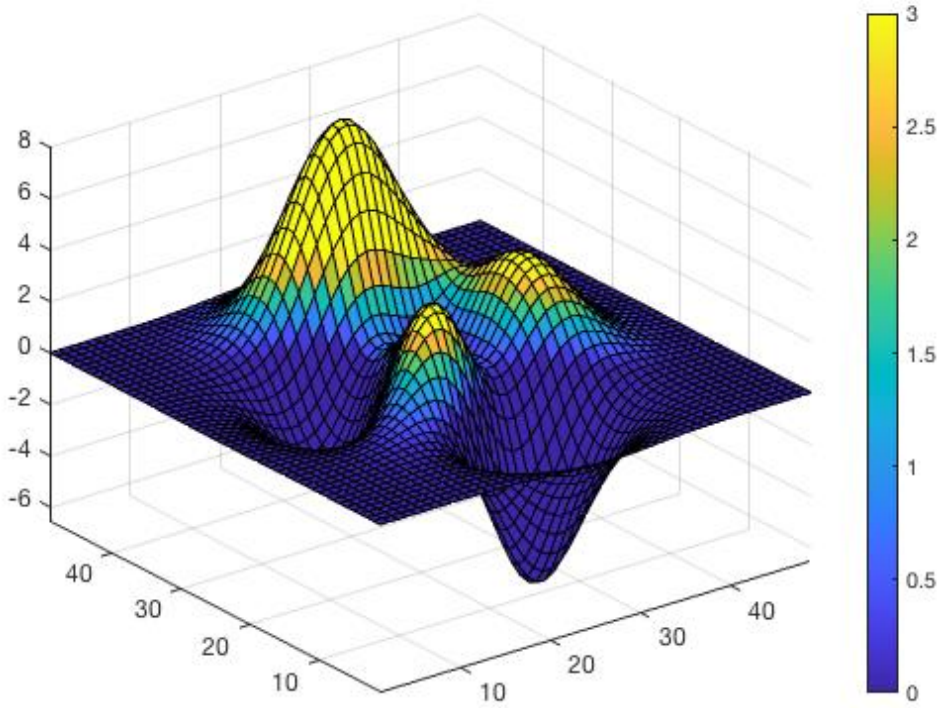
peaks 数据的 z 范围约为-6.55 至 8.08。默认情况下，颜色栏的范围设置为 z 数据的范围。我们可以在输入下列命令时看到这一点

```
caxis
```

```
ans =

   -6.5466    8.0752
```

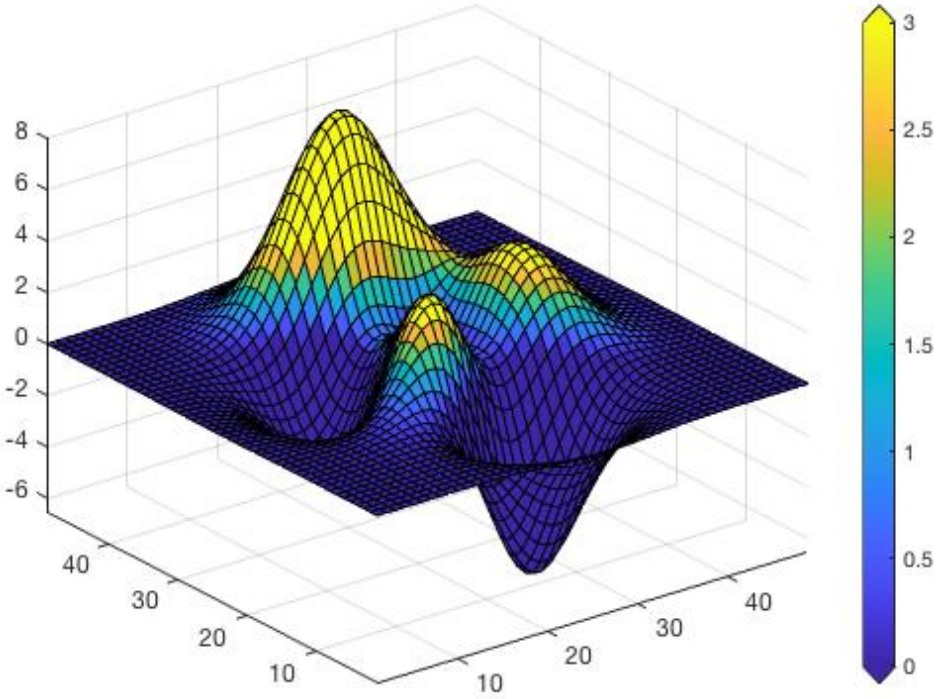
也许您希望看到在 0 到 3 的范围内发生的过程中的细微差别。为此，我们通常会相应地设置颜色轴的范围，如下所示：

```
caxis([0 3])
```



但是，现在颜色条指示所有亮黄色数据点的值正好为 3，所有深蓝色数据点的值正好为 0，并且没有数据超出这些限制。要向查看者表明某些数据确实超出了 0 到 3 的限制，请使用以下内容在颜色条的末端放置的小箭头：

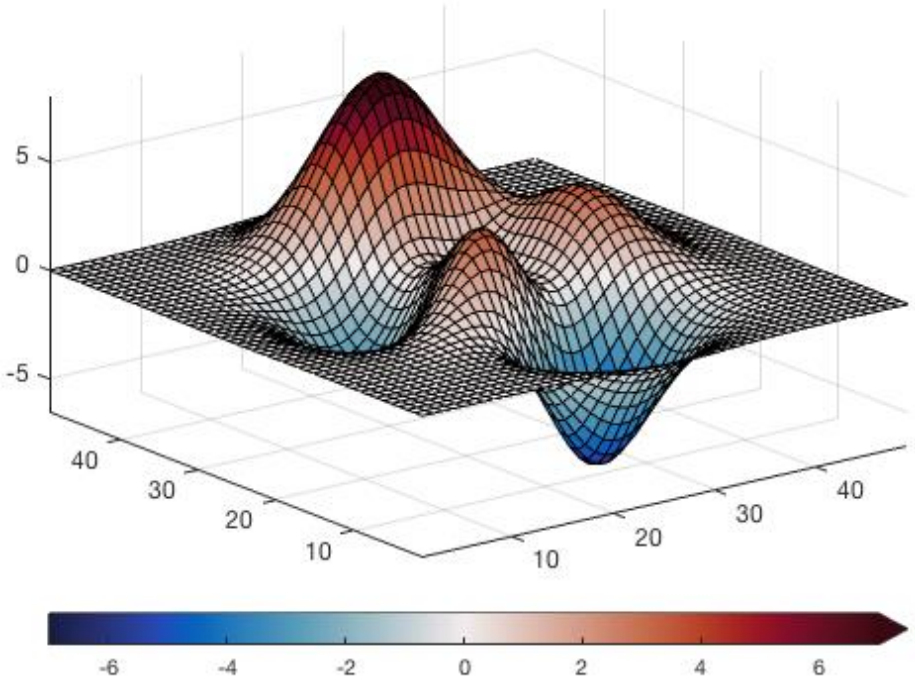
```
cbarrow
```



示例 2: 单一方向

需要只指向一个方向的颜色条吗？从 peaks 数据开始，用 cmocean 设置颜色图。下面的颜色图中零附近发散，但当我们将颜色轴从-7 设置为 7 时，仅剪裁颜色条正端的值。因此，在颜色条的右侧放置箭头才有意义：

```
figure

surf(peaks)

axis tight
colorbar('southoutside')
colormap(cmocean('balance'))
caxis([-7 7])
cbarrow('right')
```



已知问题

此函数在每个图中只工作一次。如果有多个子图，则只能使用一次，并且必须最后调用 cbarrow。此外，调用 cbarrow 后编辑绘图有时可能会有点小故障。

作者简介

newcolorbar 函数是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 于 2015 年 8 月写的。

cbdate 文档

cbdate 函数将颜色条刻度格式化为日期字符串。

语法

```
cbdate
cbdate(h)
cbdate(datenums)
cbdate(datenums,dateformat)
cbdate(datestrs)
cbdate('horiz')
h = cbdate(...)
```

说明

cbdate 将当前颜色条的刻度记号转换为日期字符串。

cbdate(h) 指定要在其上执行 cbdate 的颜色条句柄 h。如果未指定句柄，cbdate 将尝试查找当前颜色条。

cbdate(datenums) 指定要设置为刻度记号的 datenums。

cbdate(datenums,dateformat) 指定日期格式，如 datestring 文档中所述。例如，'mmm yyyy'将日期字符串打印为 Nov 2014。

cbdate(datestrs) 指定用作颜色条标记的日期字符串。

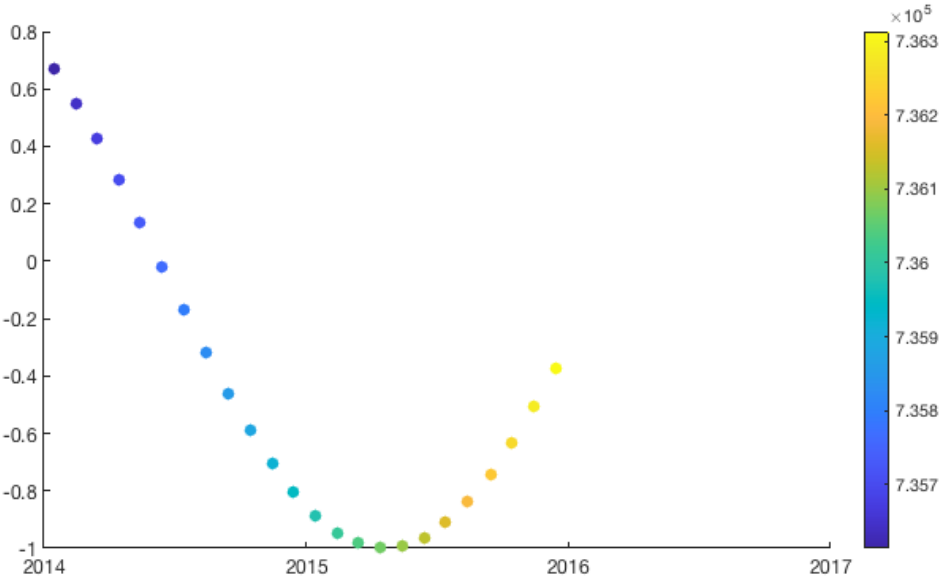
cbdate('horiz') 如果颜色条是水平的（R2014b 之前），则必须声明

h = cbdate(...) 返回颜色条句柄 h。

示例

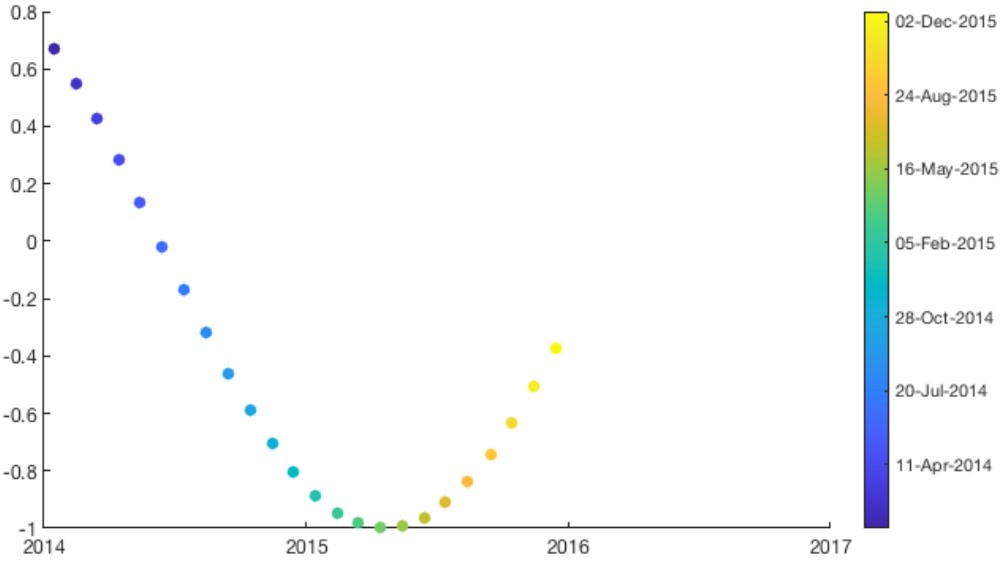
您已经绘制了一些数据，其颜色根据日期进行缩放。如下所示：

```
t = datenum(2014,1:24,15);

y = sin(t/200);



figure('pos',[100 100 700 400])
scatter(t,y,60,t,'filled');
datetick('x','yyyy')

colorbar
```



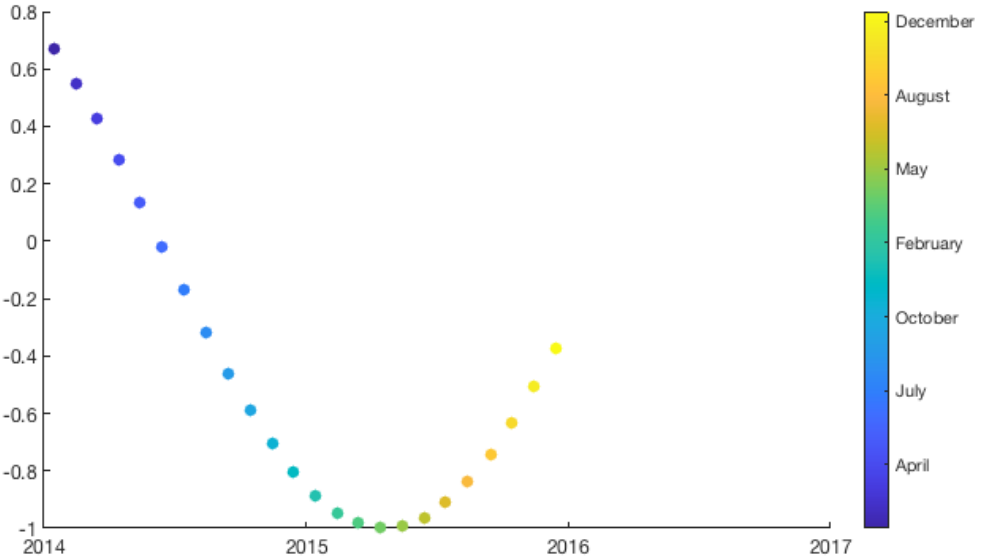
7.361 x 10^5 对大多数人来说意义不大。让我们将颜色栏上的 datenum 标记更改为日期字符串：

```
cbdate
```



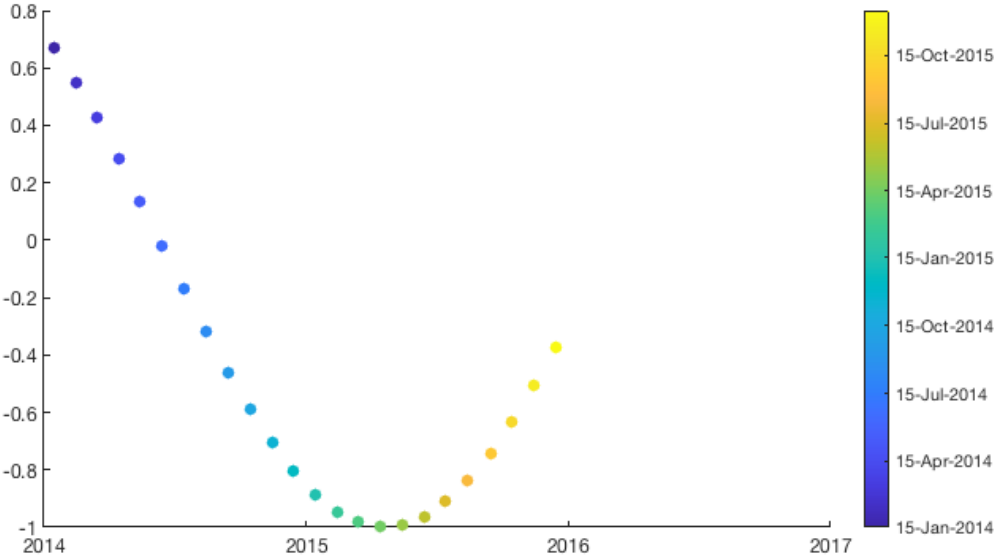
您的原始数据是每月的数据，集中在每个月的 15 日，因此使用颜色条刻度标签标记日期没有多大意义。每月的解决方案是我们所能最好的描述。因此，让我们格式化这些标签，以仅显示月份：

```
cbdate('mmmm')
```



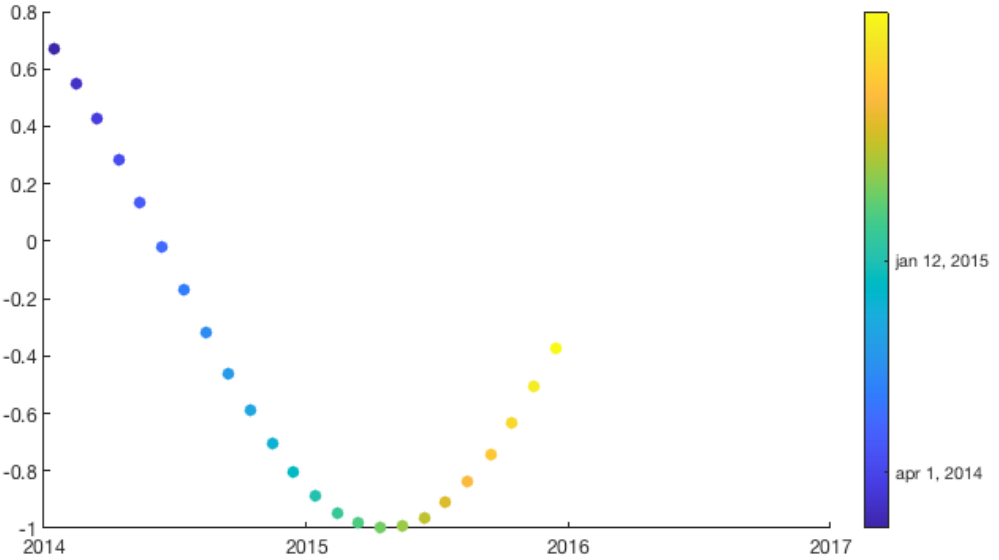
或者，您可能希望您的记号标签引用到特定的输入数据点。变量 t 是 datenum 格式的，因此它可以直接用作 cbdate 的输入。这里我们每三个月贴一次标签：

```
cbdate(t(1:3:end))
```



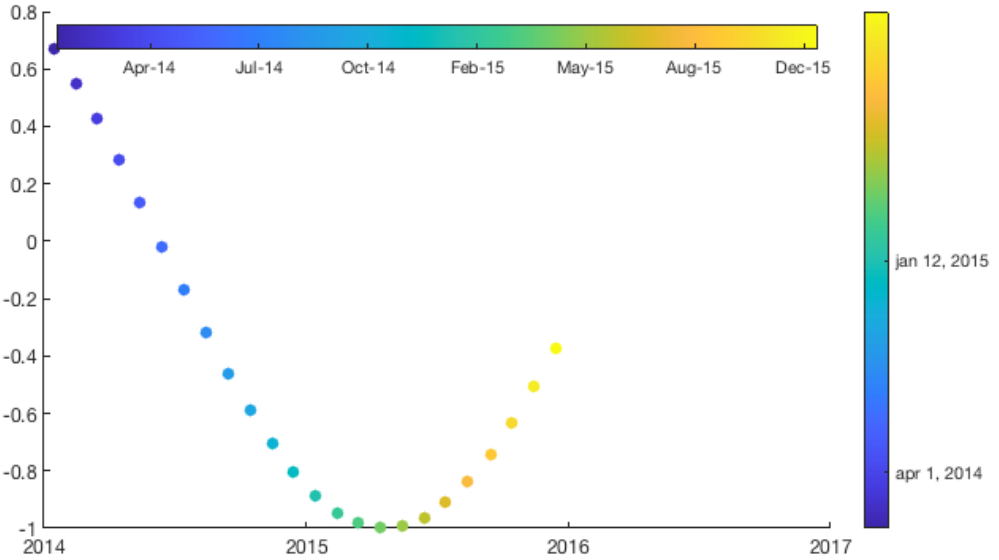
或者，您可能希望标记两个特定日期：

```
cbdate({'apr 1, 2014';'jan 12, 2015'})
```



也许您的颜色条是水平的：

```
colorbar('location','north');
cbdate('horiz','mmm-yy')
```



作者简介:

这个函数是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 于 2014 年 11 月写的，2019 年为 Climate Data Toolbox for Matlab 而更新。

hline 文档

hline 在绘图上创建水平线。
另请参见 vline 和 hfill.

语法

```
hline(y)
hline(y,LineSpec)
hline(...,'Name',Value,...)
h = hline(...)
```

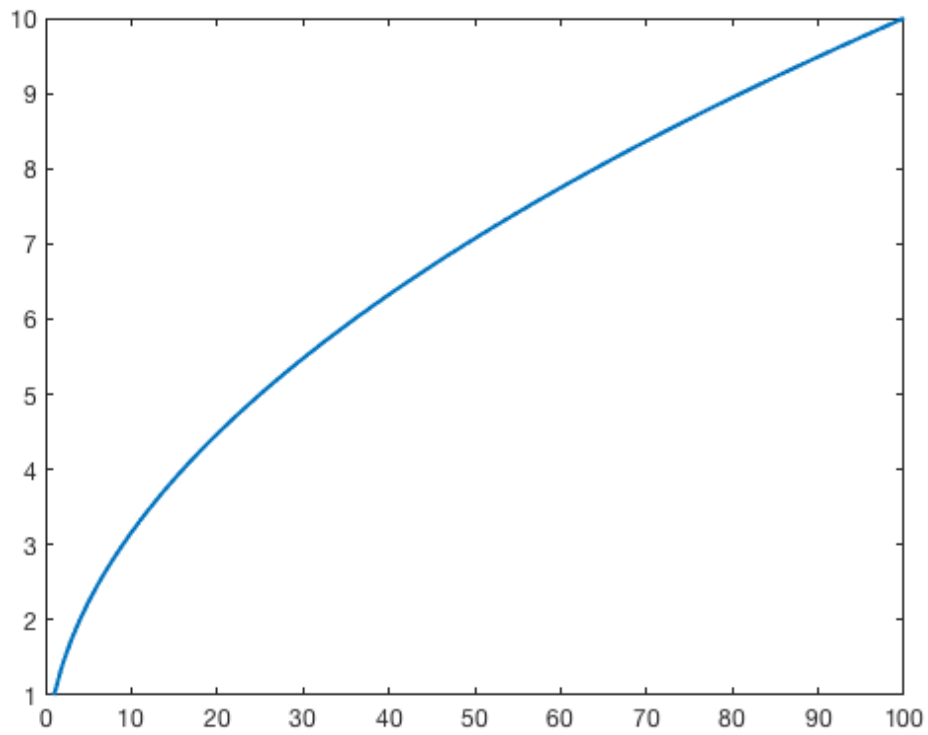
说明

hline(y) 在位置 y 处绘制水平线。
hline(y,LineSpec) 设置线条样式、标记符号和颜色。
hline(...,'Name',Value,...) 使用一个或多个名称、值对参数指定线的属性。有关特性列表，请参见线属性（Line Properties）。将此选项与上一节中的任何输入参数组合一起使用语法。
h = hline(...) 返回绘制线的句柄 h。

示例

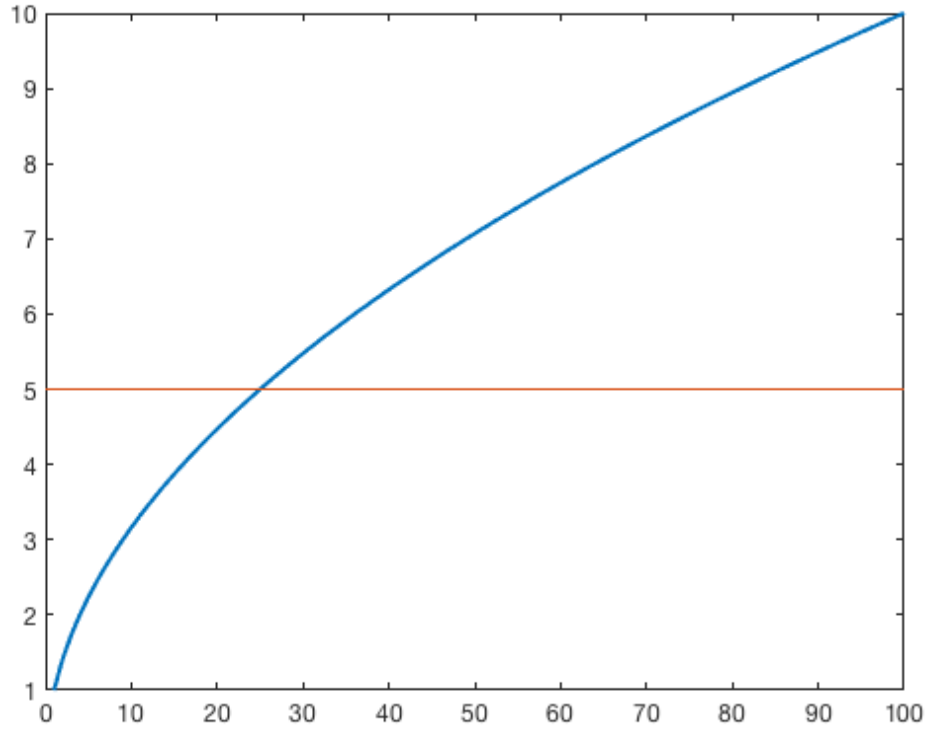
从这个图开始：

```
plot((1:100).^0.5,'linewidth',2)
```



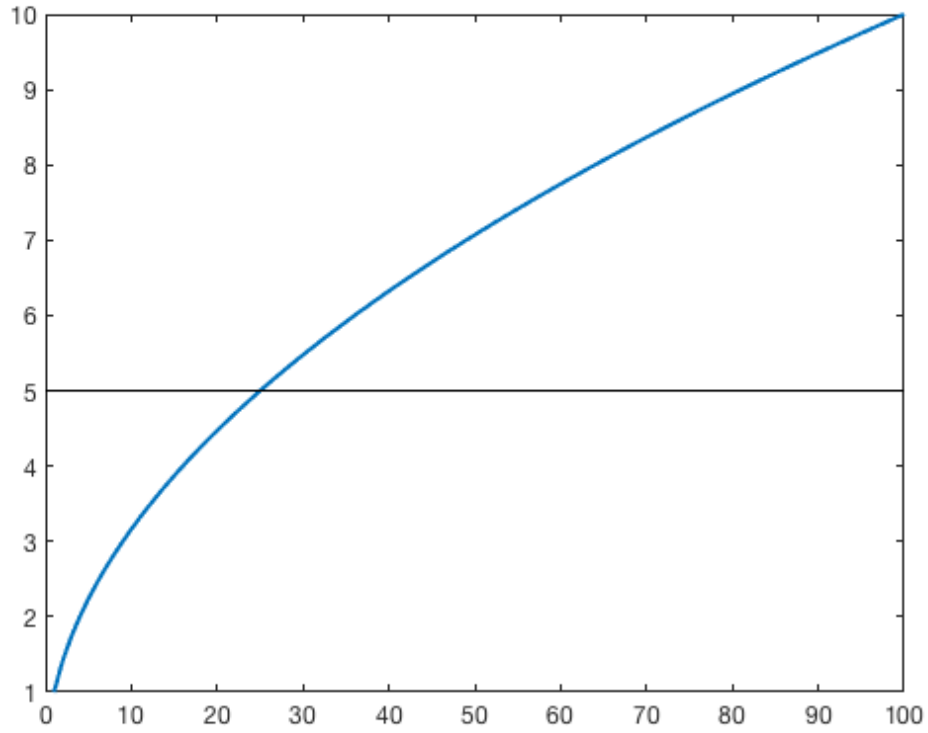
在 y=5 处添加一条水平线：

```
hline(5)
```



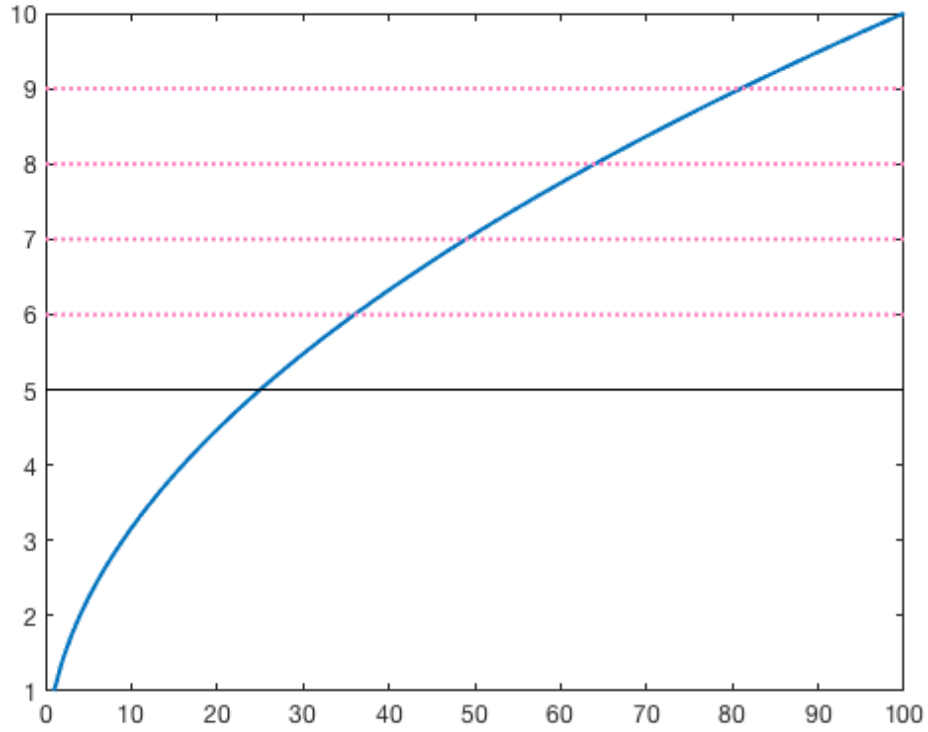
将其改为黑线：

```
hline(5,'k')
```



在 y=6、7、8 和 9 处绘制粗粉色点线。要指定粉色，可以输入 RGB 值[1 0.51 0.75]，也可以使用 rgb 函数：

```
hline(6:9,':','color',rgb('pink'),'linewidth',2)
```



作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

vline 文档

vline 在绘图上创建垂直线。
另请参见 hline 和 vfill.

语法

```
vline(y)
vline(y,LineSpec)
vline(...,'Name',Value,...)
h = vline(...)
```

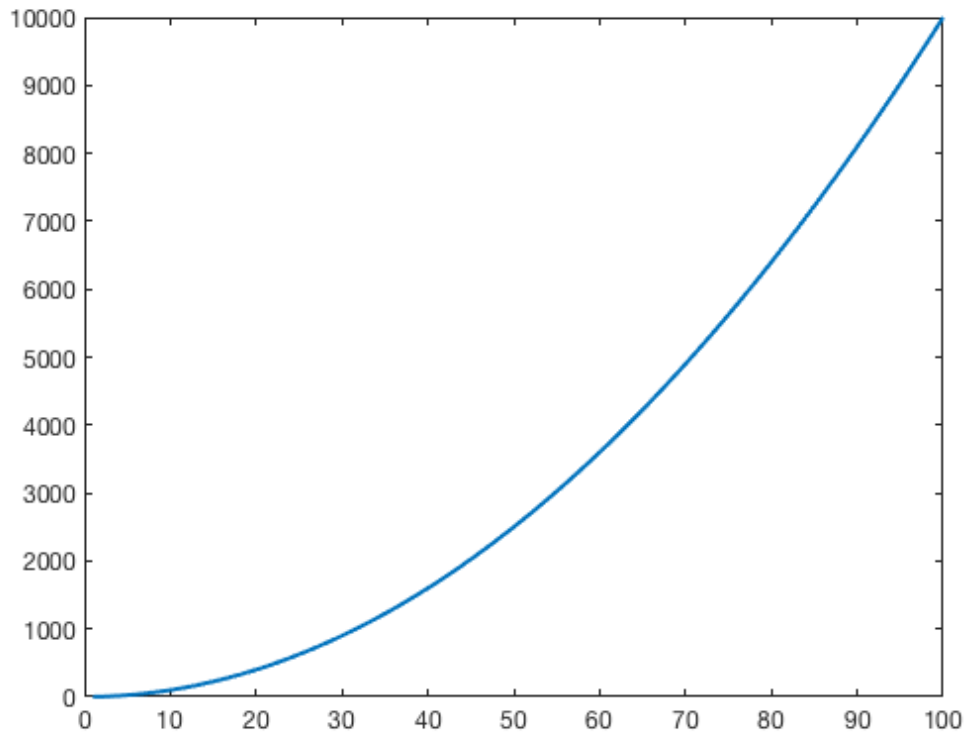
说明

vline(y) 在位置 y 处绘制垂直线
vline(y,LineSpec) 设置线条样式、标记符号和颜色。
vline(...,'Name',Value,...) 使用一个或多个名称、值对参数指定线的属性。有关特性列表，请参见线属性（Line Properties）。将此选项与上一节中的任何输入参数组合一起使用语法。
h = vline(...) 返回绘制线的句柄 h。

示例

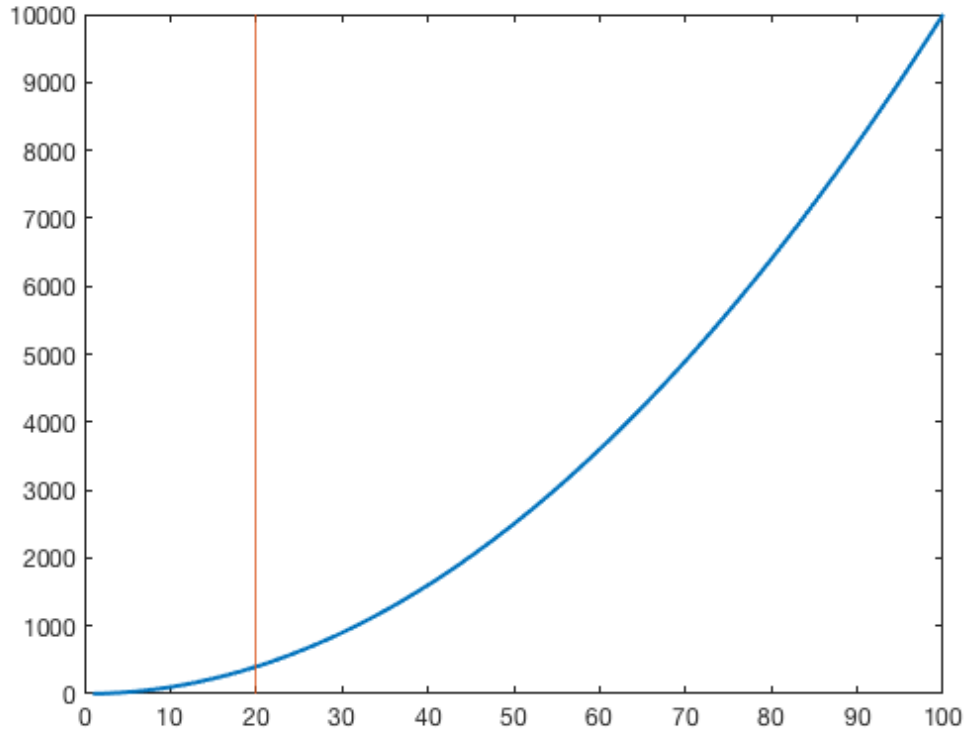
从这个图开始：

```
plot((1:100).^2,'linewidth',2)
```



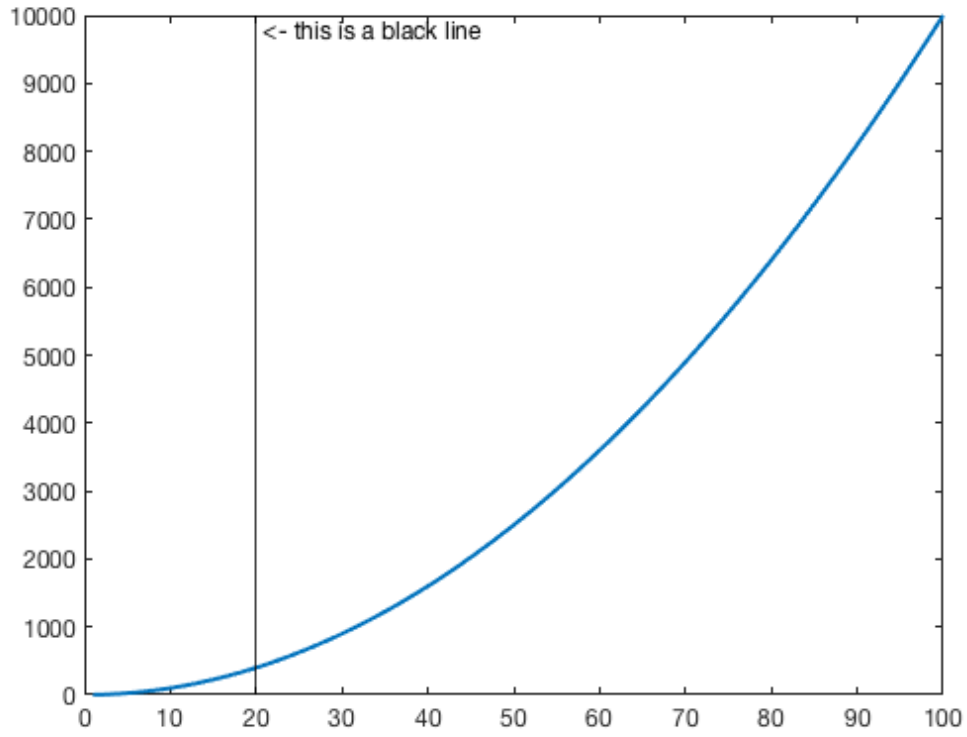
在 y=5 处添加一条垂直线：

```
vline(20)
```



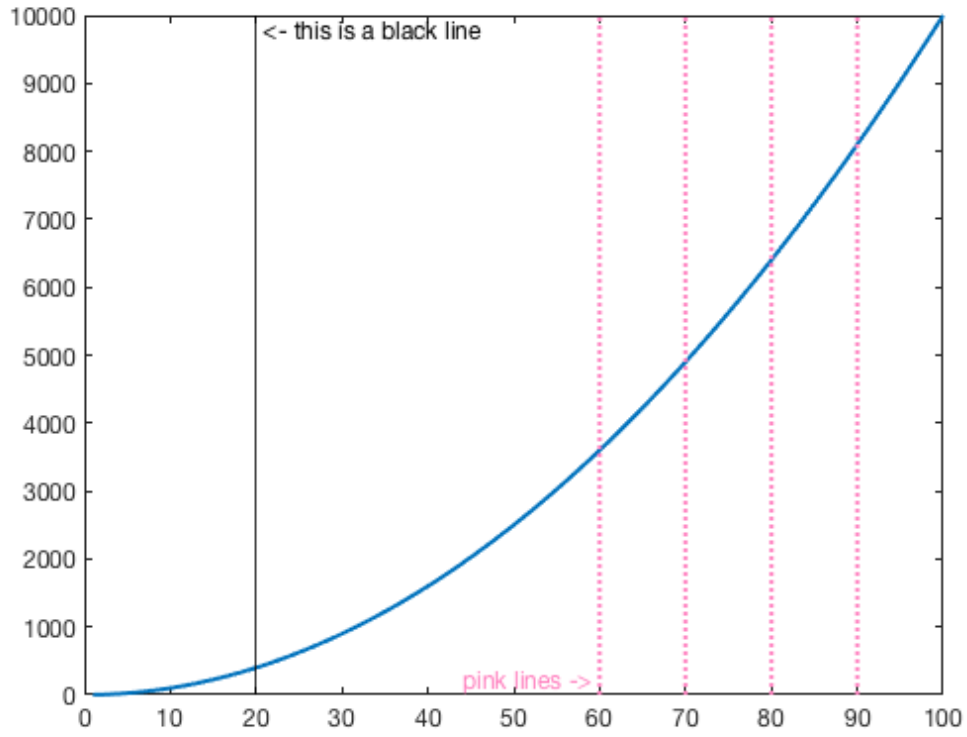
将其改为黑线：

```
vline(20,'k')

% 可选：标记线：
text(20,10000,' <- this is a black line',...
   'vert','top') % 将文本固定到左上角
```



在 y=60、70、80 和 90 处绘制粗粉色点线。要指定粉色，可以输入 RGB 值[1 0.51 0.75]，也可以使用 rgb 函数：

```
vline(60:10:90,':','color',rgb('pink'),'linewidth',2)

text(60,0,'pink lines -> ',...
   'color',rgb('pink'),...
   'horiz','right',... % 将文本固定到右侧
   'vert','bottom')       % 将文本固定到底部
```



作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

hfill 文档

hfill 在绘图上创建水平填充区域。
另请参见 hline 和 vfill.

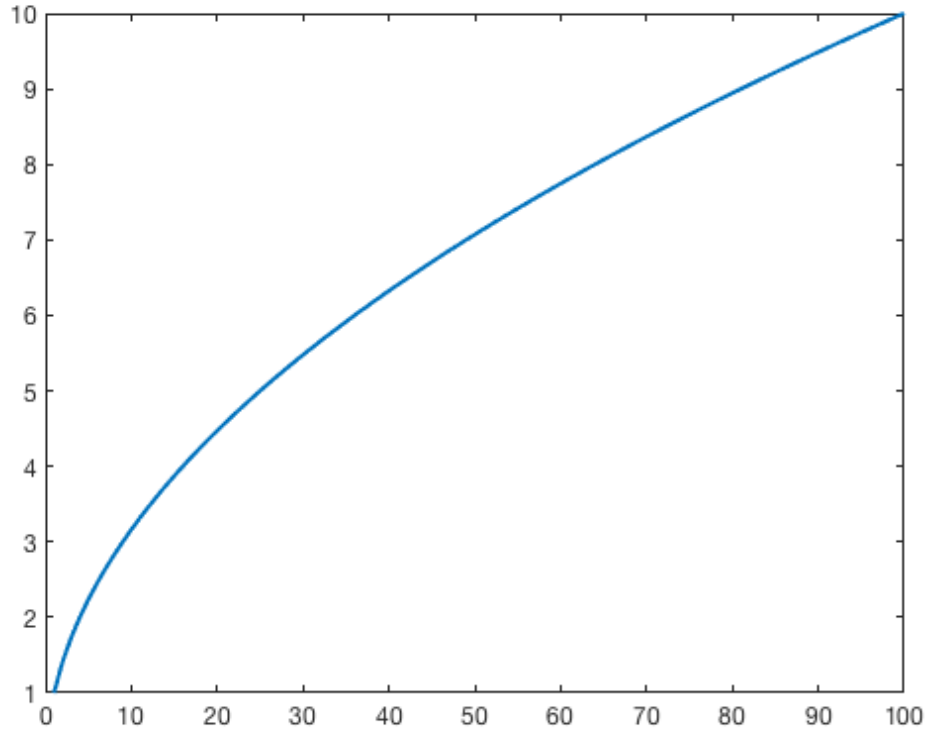
语法

```
hfill(yl,yu)
hfill(...,ColorSpec)
hfill(...,ColorSpec,'PatchProperty','PatchValue')
hfill(...,'bottom')
h = hfill(...)
```

说明

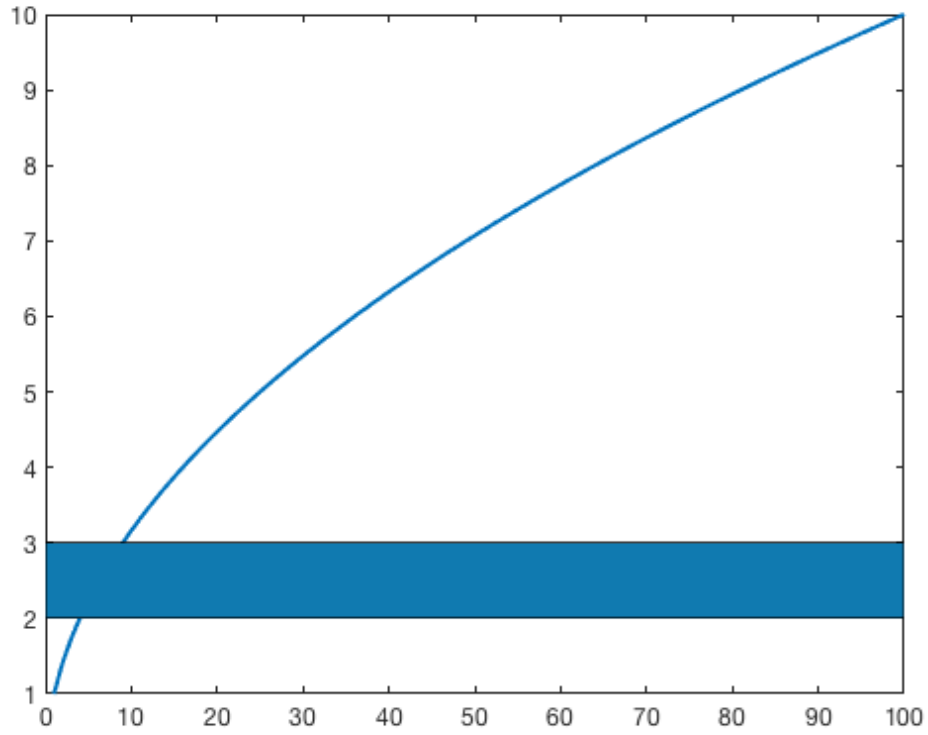
hfill(yl,yu) 在下限 yl 和上限 yu 之间创建水平填充区域。输入值 yl 和 yu 在每个阴影区域必须包含 1 个元素。
hfill(...,ColorSpec) 定义由 hfill 创建的填充的颜色。ColorSpec 可以是 Matlab 颜色名称（例如'red'）、缩写（例如'r'或 rgb 三元组（例如[1 0 0]）之一。也可以使用 rgb 函数。
hfill(...,ColorSpec,'PatchProperty','PatchValue') 定义填充属性，如'EdgeColor'或'FaceAlpha'。
hfill(...,'bottom') 将新创建的填充放在 uistack 的底部。
h = hfill(...) 返回新创建填充对象的句柄。
从这个图开始：

```
plot((1:100).^0.5,'linewidth',2)
```



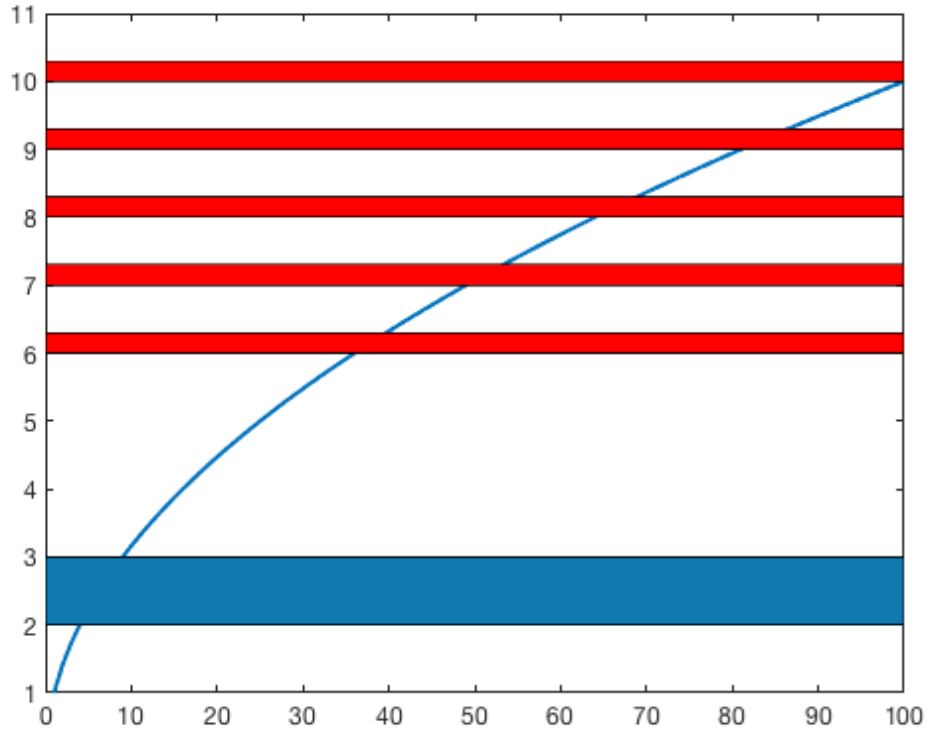
填充 y 轴上 2 和 3 之间的区域：

```
hfill(2,3)
```



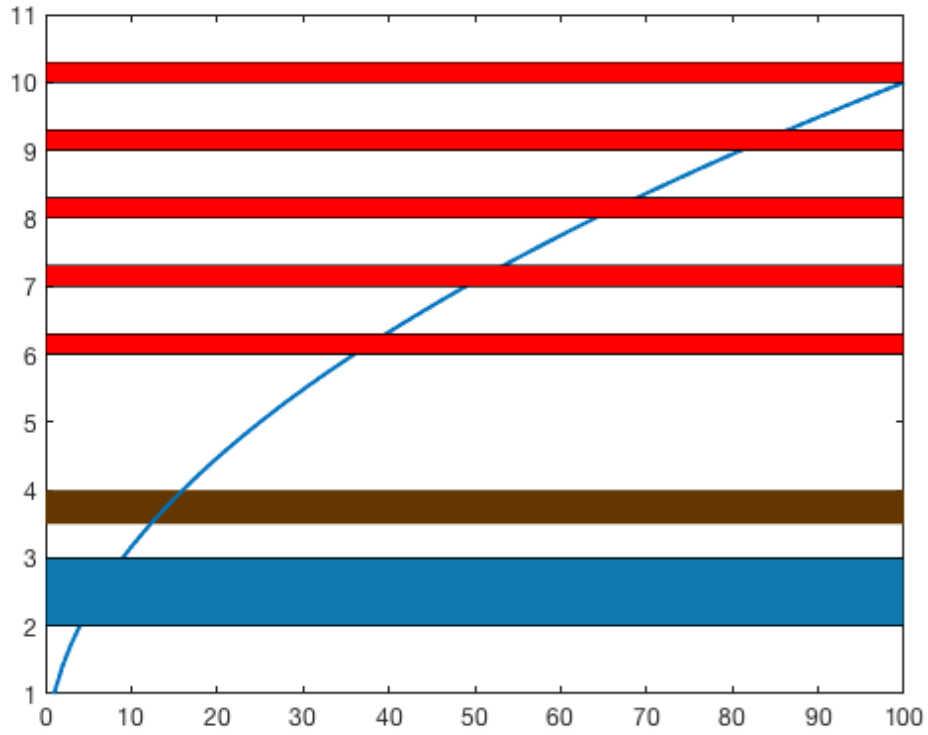
使红色区域在 6 到 10 之间的高度为 0.3 个单位

```
yl = 6:10;

hfill(yl,yl+0.3,'r')
```



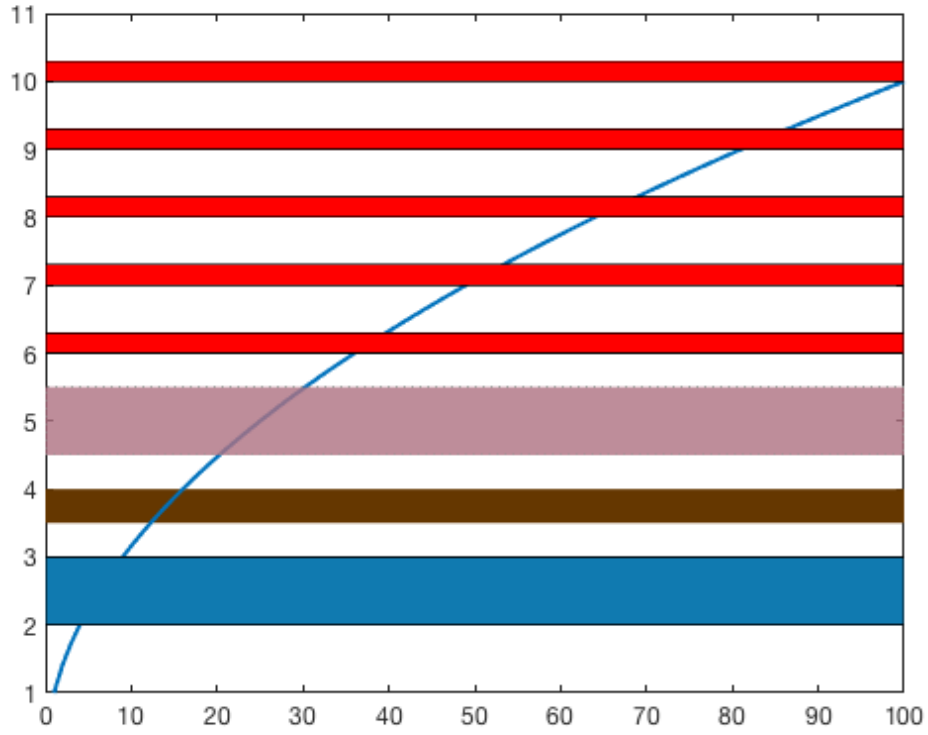
在 3.5 到 4 之间制作一个没有边缘颜色的棕色填充，并将其放在所有东西下面：

```
hfill(3.5,4,rgb('brown'),'edgecolor','none','bottom')
```



半透明的淡紫色区域，从 4.5 到 5.5，带点状灰色边缘：

```
hfill(4.5,5.5,rgb('mauve'),...
   'edgecolor',rgb('gray'),...
   'linestyle',':',...
   'facealpha',0.8);
```



作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

vfill 文档

vfill 在绘图上创建垂直填充区域。
另请参见 vline 和 hfill.

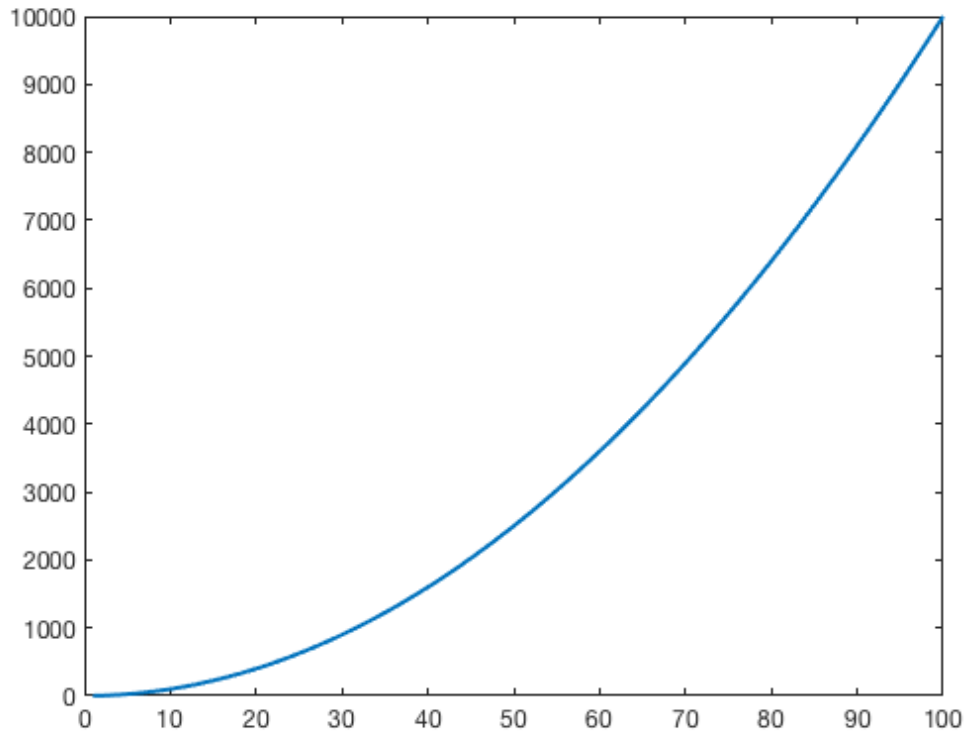
语法

```
vfill(xl,xu)
vfill(...,ColorSpec)
vfill(...,ColorSpec,'PatchProperty','PatchValue')
vfill(...,'bottom')
h = vfill(...)
```

说明

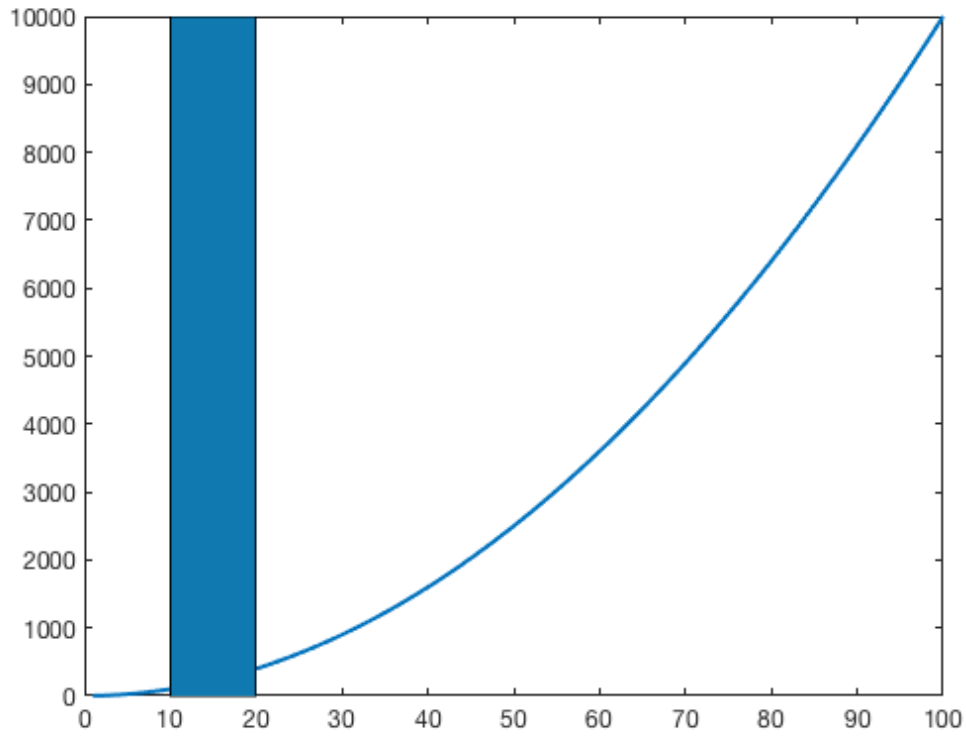
vfill(xl,xu) 在下限 xl 和上限 xu 之间创建垂直填充区域。输入值 xl 和 xu 在每个阴影区域必须包含 1 个元素。
vfill(...,ColorSpec) 定义由 vfill 创建的填充的颜色。ColorSpec 可以是 Matlab 颜色名称（例如'red'）、缩写（例如'r'或 rgb 三元组（例如[1 0 0]）之一。也可以使用 rgb 函数。
vfill(...,ColorSpec,'PatchProperty','PatchValue') 定义填充属性，如'EdgeColor'或'FaceAlpha'。
vfill(...,'bottom') 将新创建的填充放在 uistack 的底部。
h = vfill(...) 返回新创建填充对象的句柄。
从这个图开始：

```
plot((1:100).^2,'linewidth',2)
```



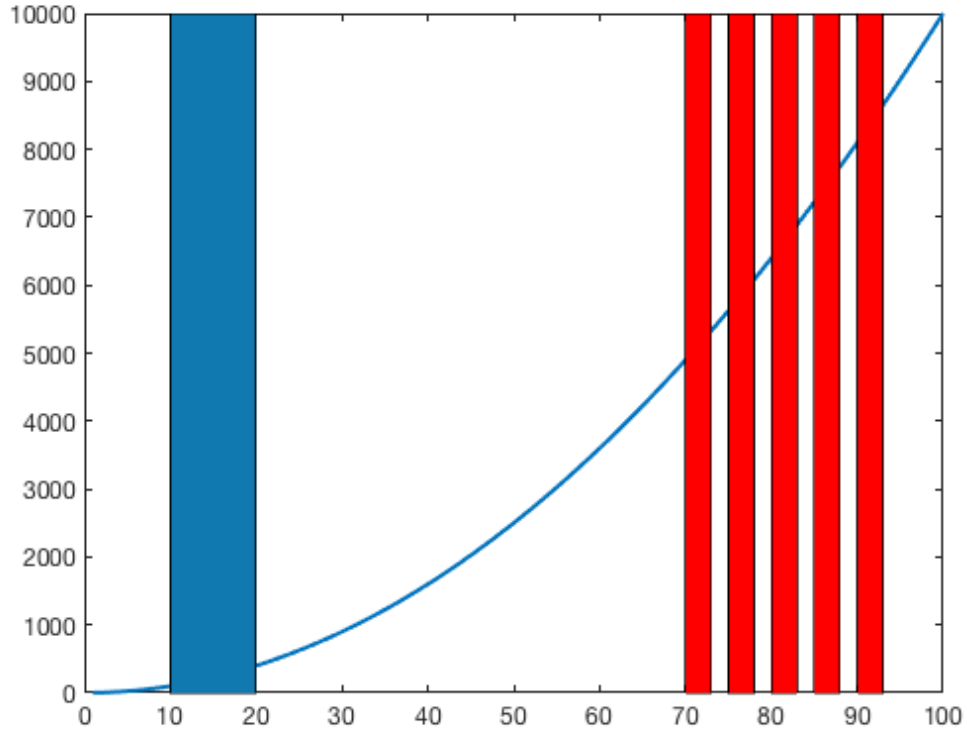
填充 10 和 20 之间的区域：

```
vfill(10,20)
```



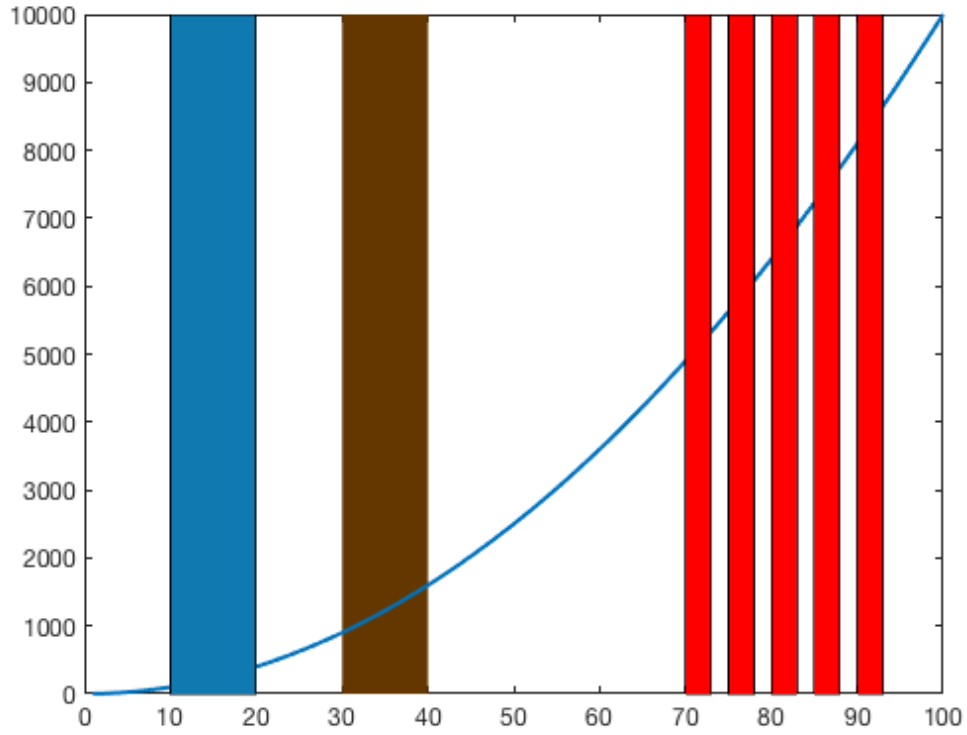
使红色区域在 70 到 90 之间的高度为 3 个单位

```
yl = 70:5:90; % 下限

vfill(yl,yl+3,'r')
```



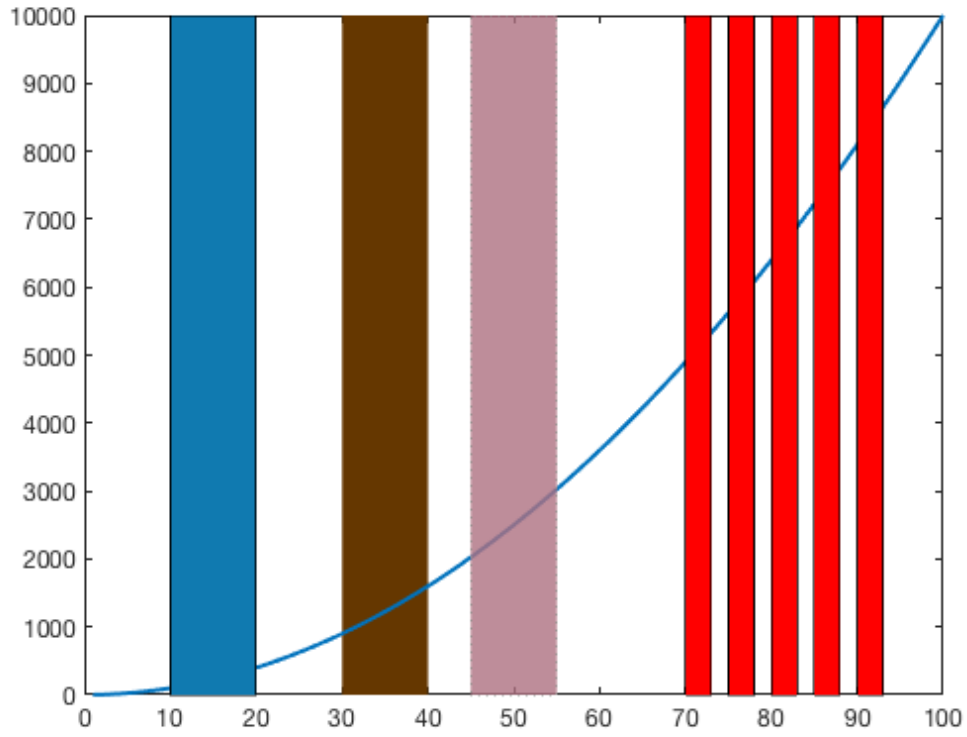
在 30 到 40 之间制作一个没有边缘颜色的棕色填充，并将其放在所有东西下面：

```
vfill(30,40,rgb('brown'),'edgecolor','none','bottom')
```



半透明的淡紫色区域，从 45 到 55，带点状灰色边缘：

```
vfill(45,55,rgb('mauve'),...
   'edgecolor',rgb('gray'),...
   'linestyle',':',...
   'facealpha',0.8);
```



作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

ntitle 文档

ntitle 将标题放置在图内而不是顶部。

语法

```
ntitle(txt)
ntitle(...,'location',InsetLocation)
ntitle(...,Name,Value)
ntitle(...,'pad',false)
h = ntitle(...)
```

说明

ntitle(txt) 将指定的标题 txt 添加到当前坐标轴。

ntitle(...,'location',InsetLocation) 将标题位置设置为

- 'north'(默认)
- 'northwest' or 'nw' 左上
- 'northeast' or 'ne' 右上
- 'east' or 'e' 左
- 'center' or 'c' 中
- 'west' or 'w' 右
- 'southwest' or 'sw' 左下
- 'south' or 's' 中下
- 'southeast' or 'sw' 右下

ntitle(...,Name,Value) 指定任何文本属性，如颜色、字体大小等。

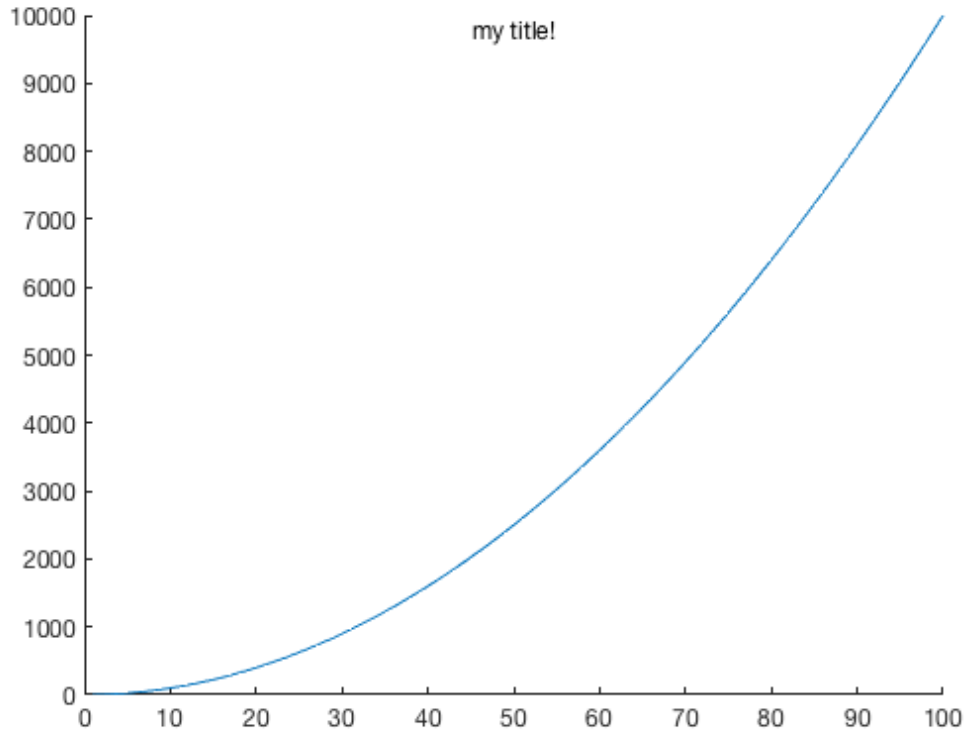
ntitle(...,'pad',false) 关闭将文本从每一侧偏移一个空格的默认行为。默认情况下，在轴边缘和文本之间放置一个空格。

h = ntitle(...) 返回标题文本的句柄 h。

示例 1

作一个简单的绘图并做个标题：

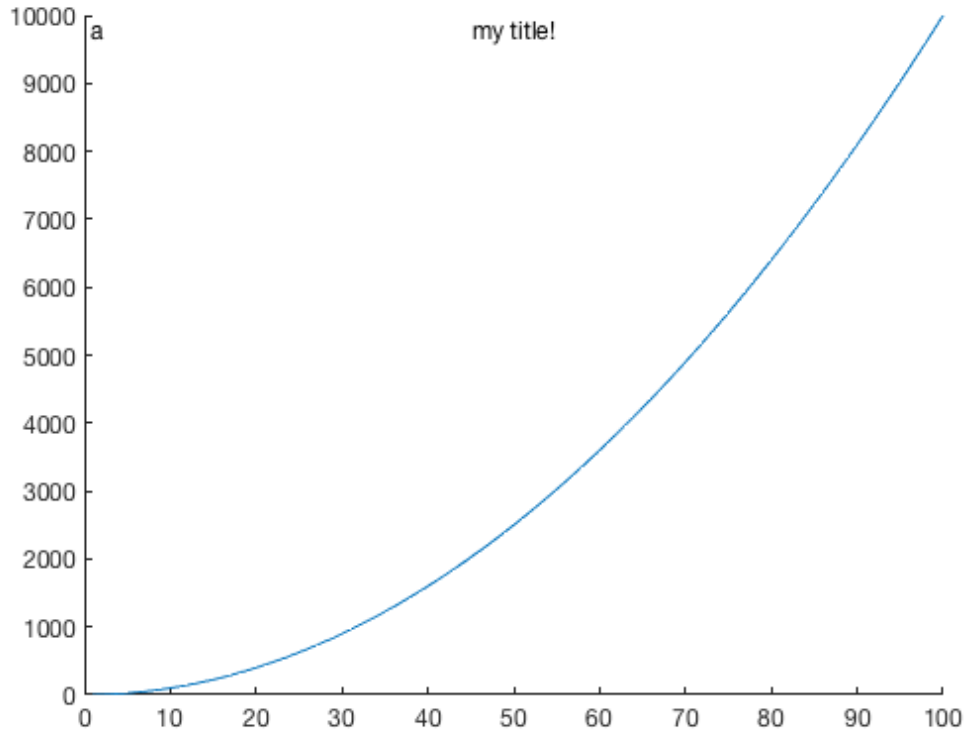
```
plot((1:100).^2)

box off

ntitle 'my title!'
```



示例 2

这是一个字母 a。正如您在发布的图形中看到的多个控件都标有字母：

```
ntitle('a','location','nw')
```



示例 3:

在右下角放置大的粗体红色文本：

```
ntitle('this text is big, bold, and red!',...
   'location','se',...
   'color','r',...
   'fontweight','bold',...
   'fontsize',20)
```

作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

gif 文档

gif 函数是制作 gif 的最简单方法。简单调用

gif('myfile.gif')

先显示第一帧，然后调用

gif

写入每个后续帧。就这样。

语法

gif('filename.gif')

gif(...,'DelayTime',DelayTimeValue,...)

gif(...,'LoopCount',LoopCountValue,...)

gif(...,'frame',handle,...)

gif(...,'nodither')

gif

gif('clear')

说明

gif('filename.gif') 以 filename.gif 的名称写入新 gif 文件的第一帧。

gif(...,'DelayTime',DelayTimeValue,...) 指定帧之间的延迟时间（以秒为单位）。默认延迟时间为 1/15。

gif(...,'LoopCount',LoopCountValue,...) 指定 gif 动画将播放的次数。默认循环计数为 Inf。

gif(...,'frame',handle,...) 使用给定图形或轴集的帧。默认帧控制句柄为 gca，表示当前轴。要将整个图形窗口转换为 gif，请使用'frame'，gcf 使用当前图形。

gif(...,'nodither') 将原始图像中的每种颜色映射到新图像中最近的颜色，而不抖动。默认情况下，执行抖动是为了获得更好的颜色分辨率，尽管这是以牺牲空间分辨率为代价的。

gif 在当前 gif 文件添加一个帧。

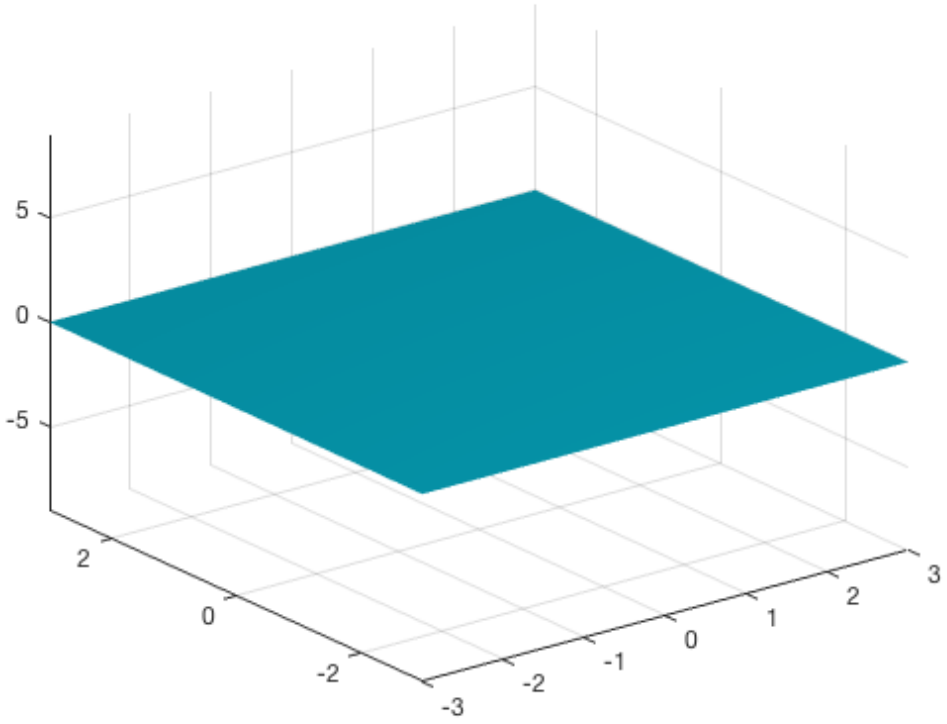
gif('clear') 清除与最新 gif 关联的持久变量。

示例

考虑这个随时间变化的表面样品：

```
% 一些样本数据:
t = sin(linspace(0,2*pi,30));
[X,Y,Z] = peaks(500);

% 绘制第一帧:
h = surf(X,Y,Z*t(1));
shading interp
axis([-3 3 -3 3 -9 9])

% 让她好看:
camlight
set(gca,'color','k')
set(gcf,'color','k')
```

```
caxis([min(Z(:)) max(Z(:))])
```



写第一帧:

当您按照您希望 gif 第一帧显示的方式来绘图时，请创建一个新的 gif 文件并按如下方式写入第一帧：

```
gif('myfile.gif')
```

如果要指定某些选项，请在第一次调用 `gif` 时包含它们。例如，如果希望每个帧之间有 0.2 秒的延迟，希望循环运行五次，并且希望使用整个体形窗口而不是当前轴，指定所有这些选项将如下所示：

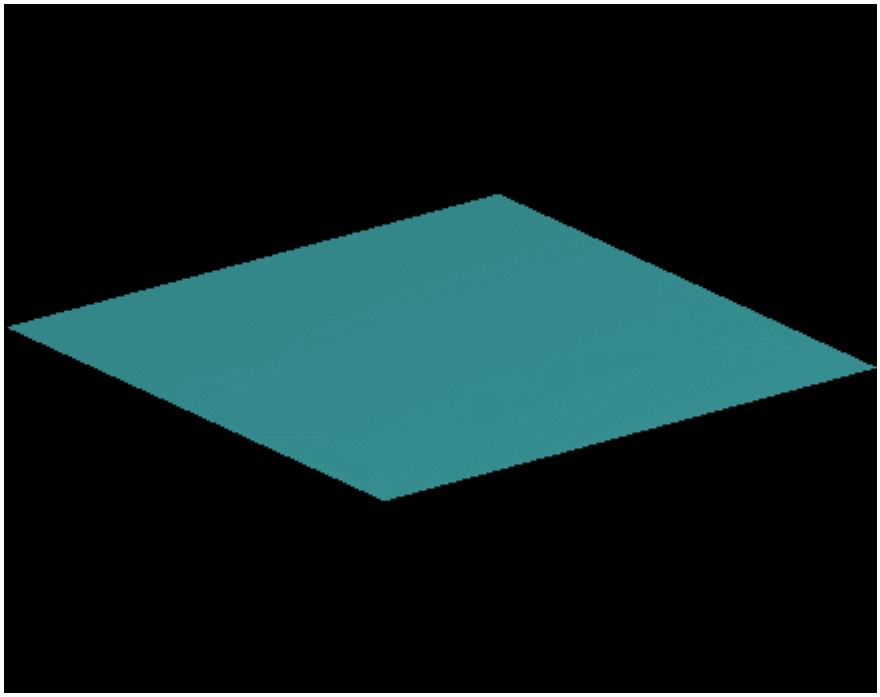
```
gif('myfile.gif','DelayTime',0.2,'LoopCount',5,'frame',gcf)
```

写其余的帧

在写入第一帧后，只需调用 `gif` 即可写入后续帧，无需任何选项。在这里，我们循环其余 29 帧：

```
for k = 2:30

   set(h,'Zdata',Z*t(k))

   gif

end
```

就这样。以下是最终产品的外观：



在 Matlab 看 gif

您知道可以在 Matlab 中查看 gif 吗？以下是方法：

```
web('myfile.gif')
```

加速动画创作

您会注意到，在上面的循环中，我们只更改了绘图中的最小数量。我们没有在每次循环中清除绘图并重新生成绘图，因为您做的事情越多，绘制的事情越多，每次迭代所需的时间就越长。与任何循环一样，尽量减少循环内的操作数。

作者简介

这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 于 2017 年 6 月写的。

线形图

- anomaly 用线绘制数据，然后用不同颜色的阴影填充曲线和参考值之间的区域。这是显示异常时间序列（如海面温度或气候指数）的常用方法。
- boundedline 使用误差着色/置信边界绘制线。
- subsubplot 在并列位置创建子轴。
- spiralplot 绘制时间序列的 Ed Hawkins 样式的螺旋图。
- plotpsd 使用 Matlab 内置的周期图函数绘制时间序列的功率谱密度。
- polyplot 绘制了一个多项式拟合散点的 x，y 数据。

anomaly 文档

anomaly 函数用线绘制数据，然后用不同颜色的阴影填充曲线和参考值之间的区域。这是显示距平时间序列（如海面温度或气候指数）的常用方法。

语法

```
anomaly(x,y)
anomaly(...,'thresh',thresholdValue)
anomaly(...,'top',ColorSpec)
anomaly(...,'bottom',ColorSpec)
anomaly(...,'LineProperty',LineValue)
[hlin,htop,hbot] = anomaly(...)
```

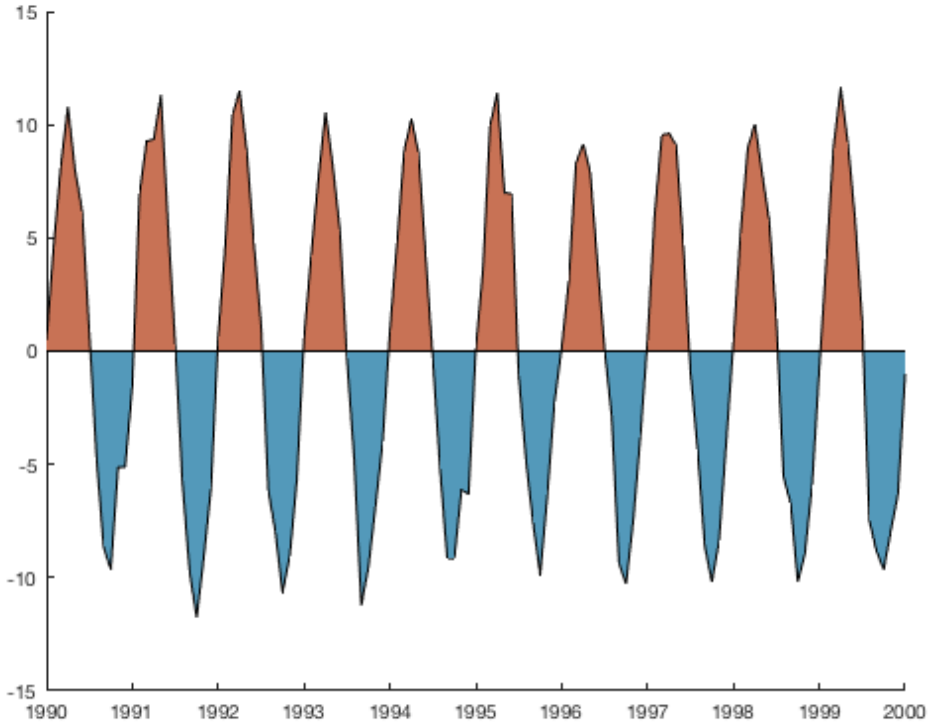
说明

anomaly(x,y) 绘制一条黑线，其中红色填充零线和零线以上的任何线值之间的区域；蓝色填充零和零以下任何值之间的区域。
anomaly(...,'thresh',thresholdValue) 指定超过便进行着色的值。默认情况下，阈值为零，这意味着高于或低于零的所有内容都将着色。如果需要相对于零以外的某个值进行着色，请将该值指定为标量阈值。如果只需要在某个较低阈值下方和较高阈值上方进行着色，请将 thresholdValue 指定为两元素数组（例如，将 thresholdValue 设为[-0.4 0.5]，以着色所有小于-0.4 或大于 0.5 的值）。
anomaly(...,'topcolor',ColorSpec) 指定上限以上着色颜色，可以用 RGB 值或任何 Matlab 速记颜色名称（例如，'r'或'red'）来描述。
anomaly(...,'bottomcolor',ColorSpec) 指定下限以下着色颜色。
anomaly(...,'LineProperty',LineValue)设置任何线条属性，如'color'或'linewidth'。
[hlin,htop,hbot] = anomaly(...) 分别返回直线图、上限图和下限图的图形句柄。

示例 1: 简单版

绘制这个样本数据:

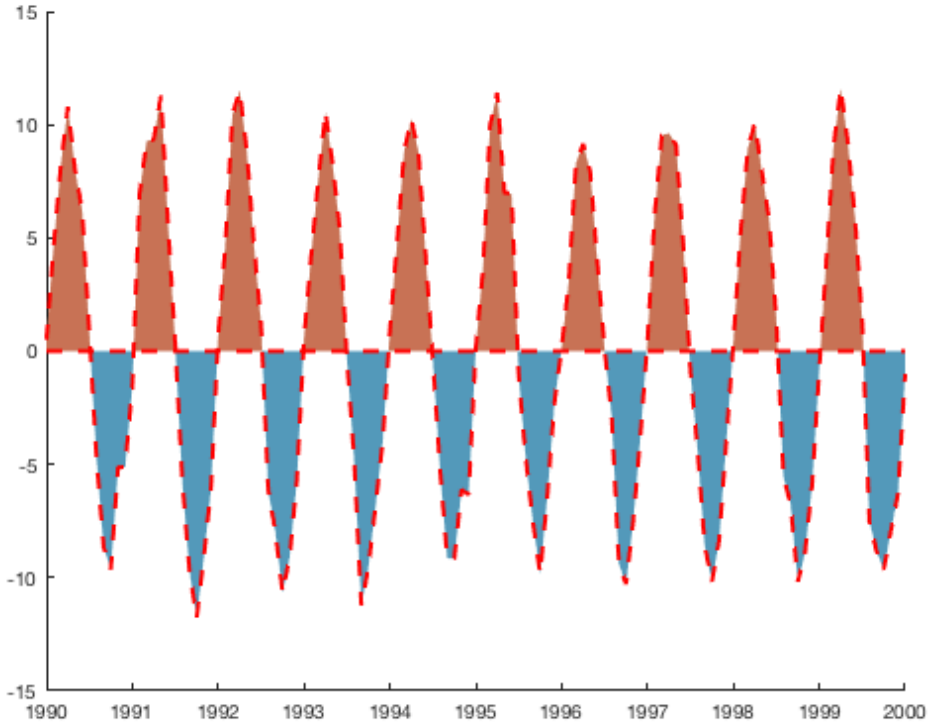
```
x = 1990:1/12:2000;

y = 10*sin(2*pi*x) + randn(size(x));



anomaly(x,y)
```



示例 2: 指定着色颜色和线条属性

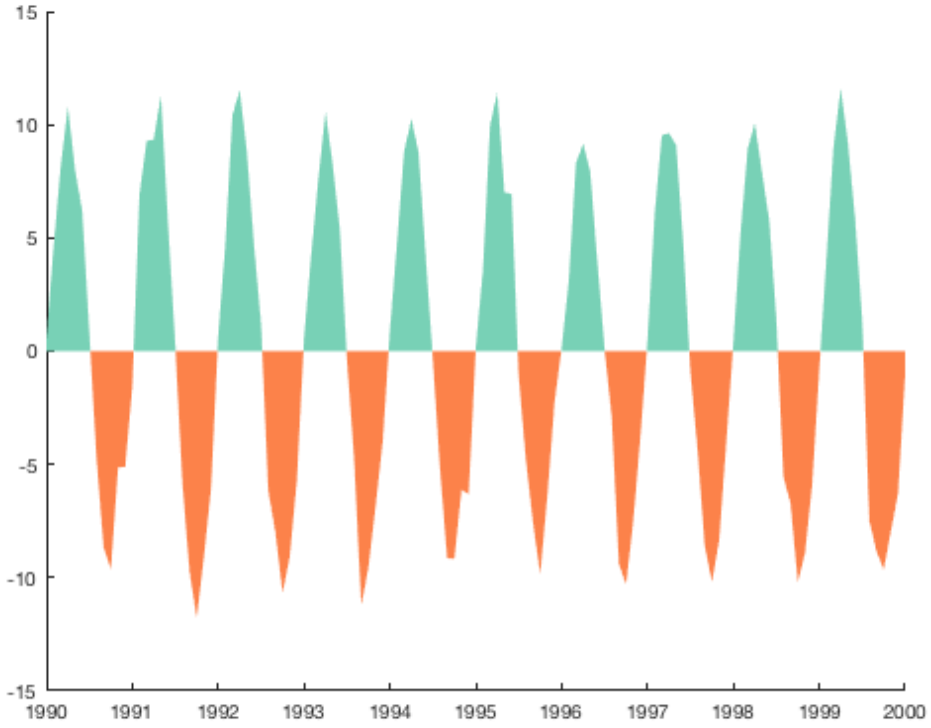
使用与上述相同的 x 和 y 数据，使线变粗、变红和变虚线：

```
figure

anomaly(x,y,'color','r','linewidth',2,'linestyle','--')
```



或者完全没有线条，而上限的是海蓝色的，下限是橙色的。使用 rgb 函数获取此类俗语颜色名称的 rgb 值：

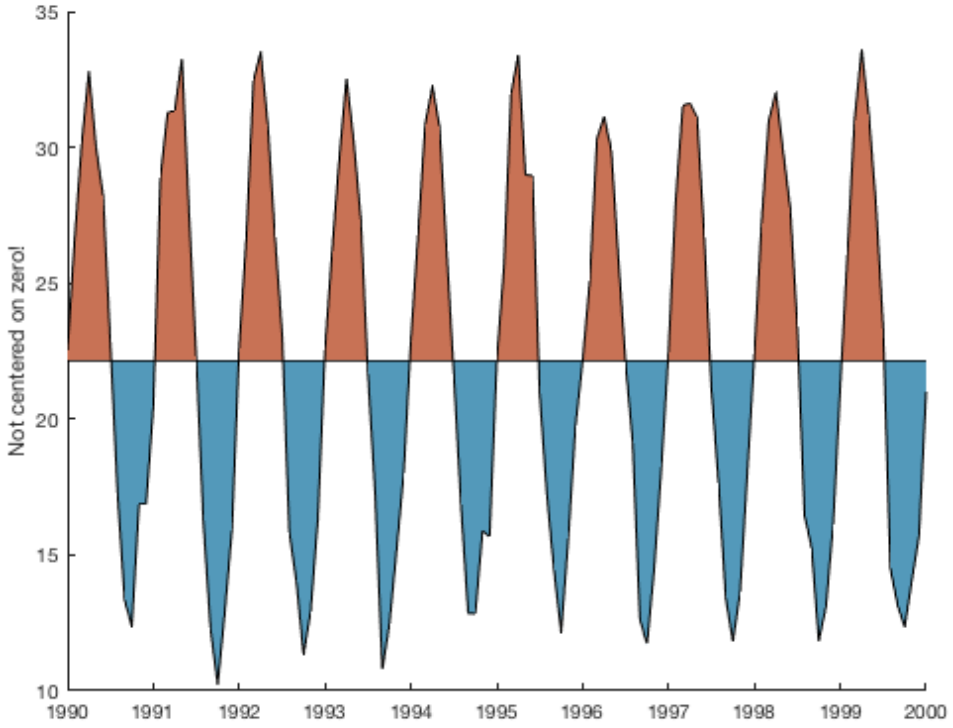
```
figure

anomaly(x,y,'color','none','top',rgb('seafoam blue'),...
   'bottom',rgb('orangish'))
```



示例 3: 关于非零值的距平

有时距平不是相对于零，而是相对于它们的平均值。也许有些温度值在 22 度左右波动：

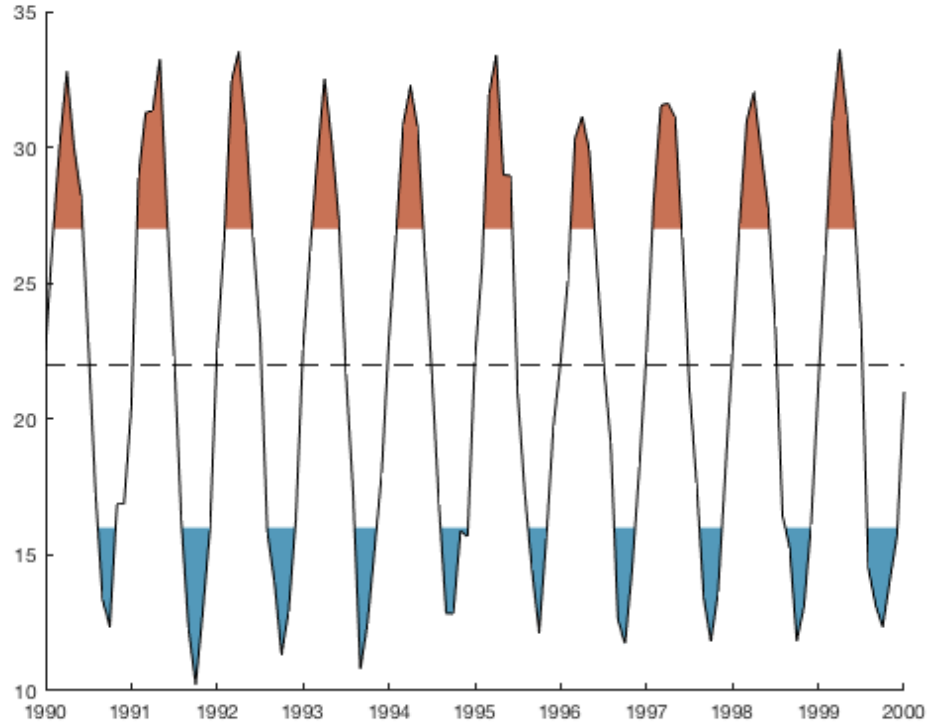
```
figure



anomaly(x,y+22,'thresh',mean(y+22))

ylabel 'Not centered on zero!'
```



示例 4: 阈值

仅在小于 16 或大于 27 的区域进行着色。如果需要，请使用 hline 在 y 值 22 处添加一条水平黑线：

```
figure



anomaly(x,y+22,'thresh',[16 27])

hline(22,'k--') % 在 y=22 处添加水平线.
```



作者简介

这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 于 2017 年 1 月写的。

boundedline 文档

boundedline 函数使用误差着色/置信边界绘制线。

语法

```
boundedline(x, y, b)
boundedline(x, y, b, linespec)
boundedline(x1, y1, b1, linespec1,  x2, y2, b2, linespec2)
boundedline(..., 'alpha')
boundedline(..., ax)
boundedline(..., 'transparency', trans)
boundedline(..., 'orientation', orient)
boundedline(..., 'nan', nanflag)
boundedline(..., 'cmap', cmap)
[hl, hp] = boundedline(...)
```

说明

boundedline(x, y, b) 用 x 和 y 给出的坐标绘制一条线，周围有一个填充，填充在该线上方/下方延伸一定距离 b。x、y 和 b 数组的尺寸可以变化，以允许同时绘制多条线，并且填充边界可以是恒定的，也可以沿线的长度变化。x 和 y 的尺寸必须满足与 plot 函数相同的要求，并将产生与具有相同输入的 plot 相同数量的线。b 数组应为 npoint x nside x nline，其尺寸对应于直线上的每个点（维度 1）、直线的每一侧（下/上或左/右，取决于方向）（维度 2）以及前面的 x-y 值（维度 3）描述的每个绘制线。如果 size(b,1) == 1，则直线上所有点的边界都相同。如果 size(b,2) == 1，则边界将在线的两侧对称。如果 size(b,3) == 1，则相同的边界将应用于前面 x-y 数组描述的所有行（仅当 x 或 y 是数组时适用）。边界不能包括 Inf、-Inf 或 NaN。

boundedline(..., 'alpha') 使用与对应线相同颜色的部分透明填充渲染有界区域。如果未包括在内，则边界区域将使用一个完全不透明的填充，其颜色为相应线颜色的较浅阴影。

boundedline(..., 'transparency', transp) 使用介于 0 和 1 之间的标量指示边界填充的透明度或强度。默认值为 0.2。

boundedline(..., 'orientation', orient) 表示边界的方向。方向可以是垂直（y 方向）边界的“'vert'，也可以是水平（x 方向）边界的'horiz'。默认值为'vert'。

boundedline(..., 'nan', nanflag) 指示边界填充应如何处理线坐标或边界值中的 nan。选项为 'fill'，使用相邻值平滑间隙，'gap' 在填充中这些点留下空白，或'remove' 完全删除 NaN 点，从而导致填充中间隙的线性插值。有关这些选项的更多详细信息，请参见下面的示例。

boundedline(..., 'color', col) 指定线条颜色。

boundedline(..., 'cmap', cmap) 根据此 n x 3 颜色图数组中的颜色为线着色（按打印顺序），覆盖任何 LineSpec 或默认颜色。

boundedline(..., ax) 将有界线绘制到控制柄 ax 指示的轴。如果未包括，则使用当前轴。

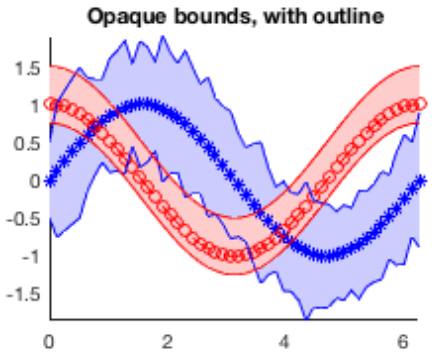
[hl, hp] = boundedline(...) 将结果线和填充对象的句柄分别返回给 hl 和 hp。

示例 1: 使用各种语法选项绘制线

此示例构建 MatlabCentral File Exchange 上使用的 4- panel 示例图像，其中显示了提供线坐标、边界坐标和着色选项的几种不同方法。

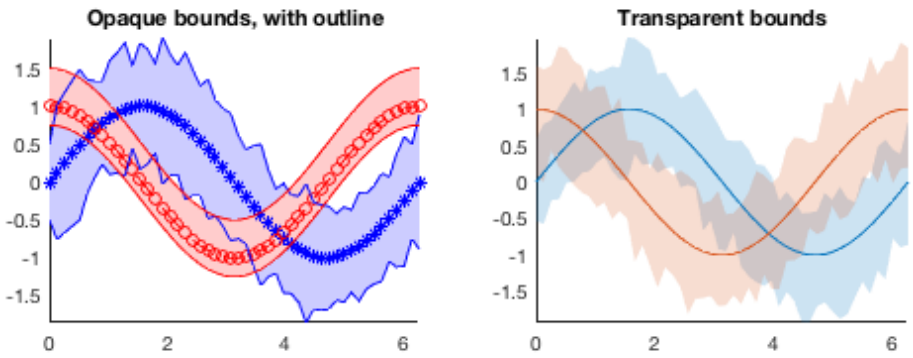
第一个轴使用 LineSpec 选项打印两条线作为输入，该选项允许 yoy 为每条线设置线颜色、线颜色和标记类型。第一条线的边界在 x 上变化，而第二条线的边界对于所有 x 都是常数。边界中添加了一个轮廓，以便可以更清楚地看到重叠区域。

```
x = linspace(0, 2*pi, 50);

y1 = sin(x);

y2 = cos(x);

e1 = rand(size(y1))*.5+.5;

e2 = [.25 .5];



ax(1) = subplot(2,2,1);

[l,p] = boundedline(x, y1, e1, '-b*', x, y2, e2, '--ro');
outlinebounds(l,p);
title('Opaque bounds, with outline');
axis tight;
```



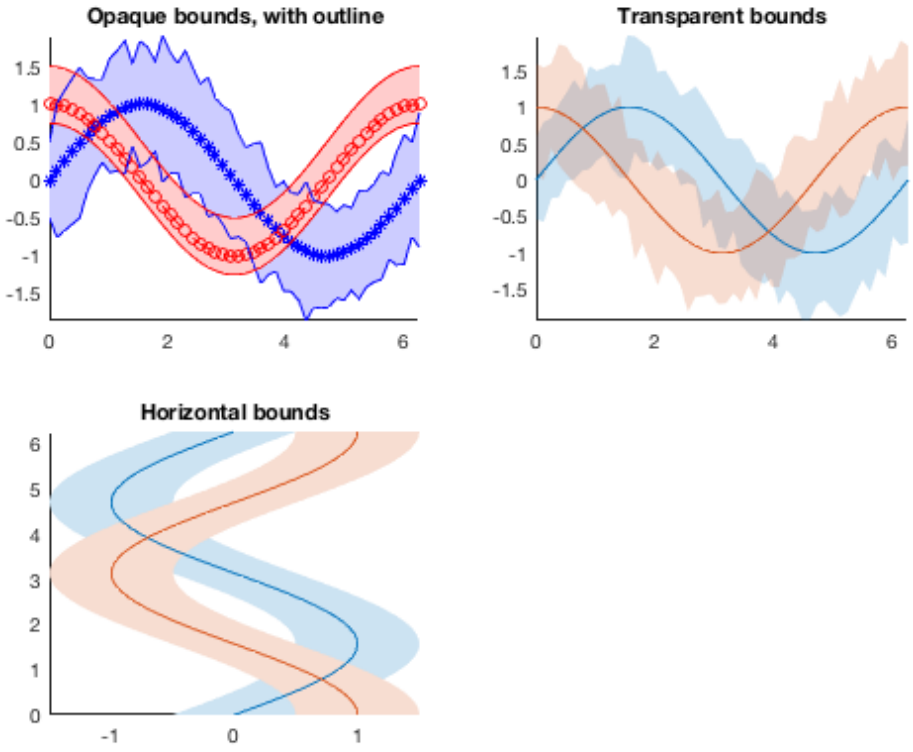
对于我们的第二个轴，我们使用相同的两条线，这次为这两条线指定 x 变化的边界。而不是使用 LineSpec 语法，此示例使用默认的颜色顺序来指定线和填充的颜色。我还启用了“'alpha'”选项，该选项可以使填充部分透明。

```
ax(2) = subplot(2,2,2);

boundedline(x, [y1;y2], rand(length(y1),2,2)*.5+.5, 'alpha');
title('Transparent bounds');
axis tight;
```



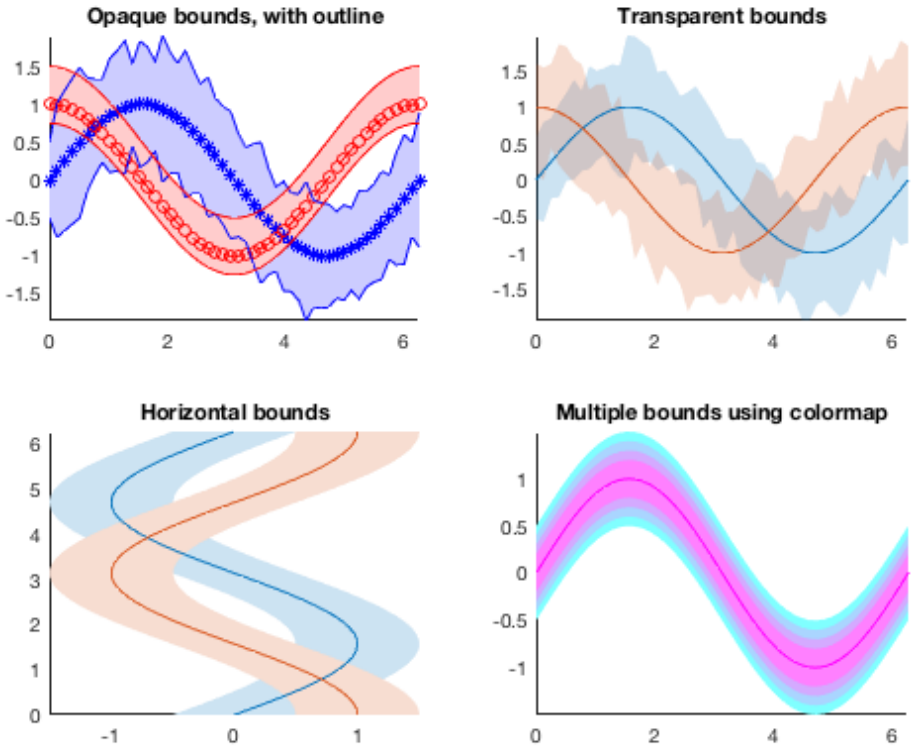
对于 x 轴表示因变量的情况，边界也可以指定为水平方向。在这种情况下，标量误差界限值应用于两条线和线的两侧。

```
ax(3) = subplot(2,2,3);

boundedline([y1;y2], x, e1(1), 'orientation', 'horiz')
title('Horizontal bounds');
axis tight;
```



可以使用颜色图数组来指定颜色，而不是使用 LineSpec 或默认颜色顺序。在这种情况下，将在同一行的顶部添加越来越窄的边界。

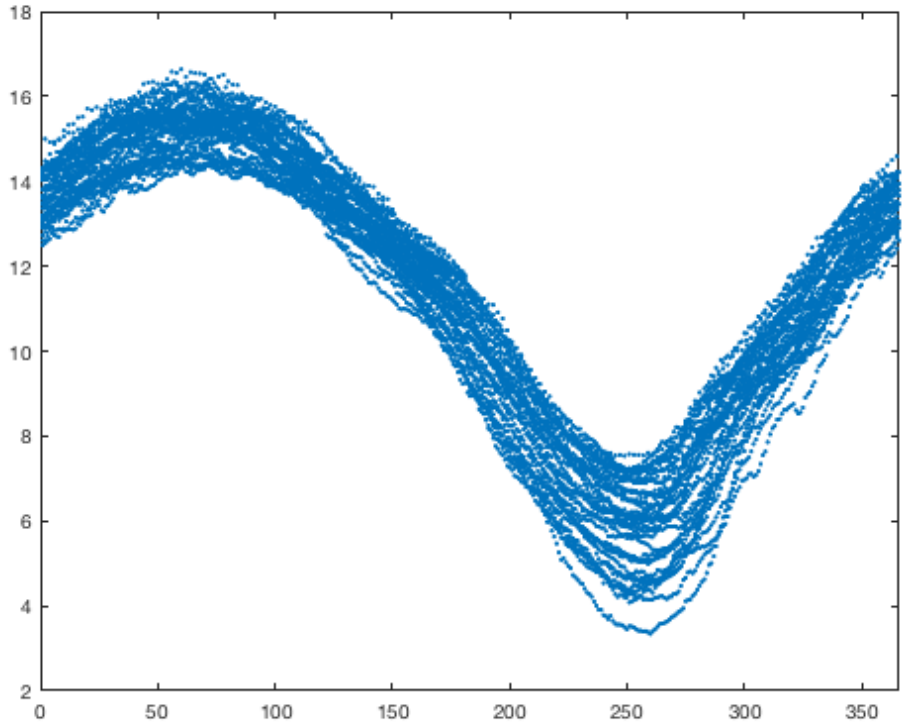
```
ax(4) = subplot(2,2,4);

boundedline(x, repmat(y1, 4,1), permute(0.5:-0.1:0.2, [3 1 2]), ...
    'cmap', cool(4), ...
    'transparency', 0.5);
title('Multiple bounds using colormap');

set(ax([1 2 4]), 'xlim', [0 2*pi]);
set(ax(3), 'ylim', [0 2*pi]);
axis tight;
```



示例 2: 通用气候数据用途

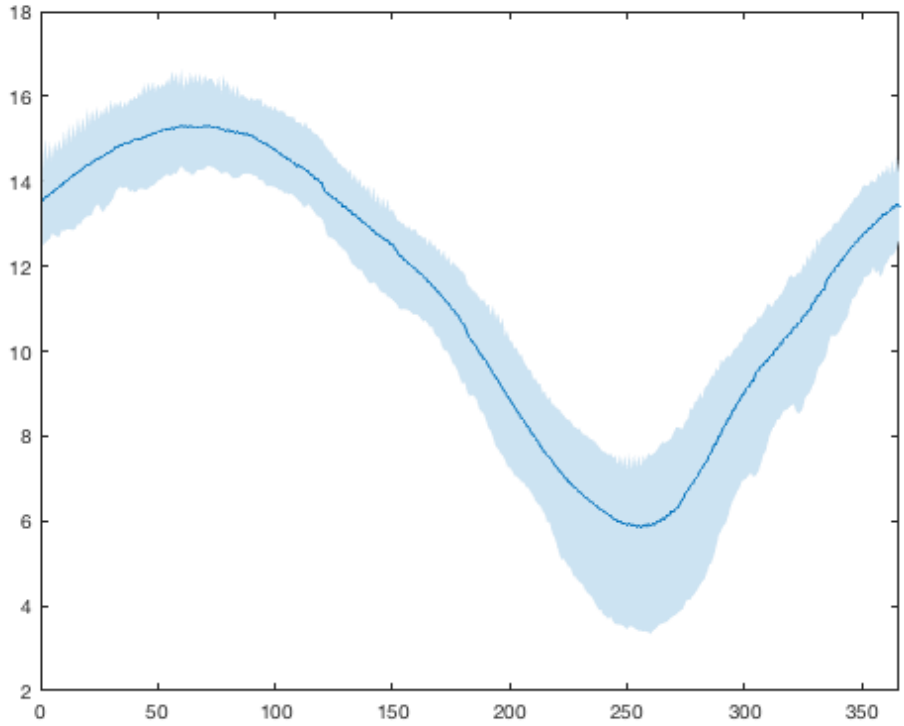
这种类型的绘图的典型用途是显示与时间序列关联的值的范围。例如，让我们看看海冰范围数据的季节变化：

```
S = load('seaice_extent.mat');
figure;
plot(doy(S.t), S.extent_N, '.');
set(gca, 'xlim', [0 366]);
```



我们可以计算所有年份的日变化范围，并将这些范围与平均值一起绘制

```
[g, dy] = findgroups(doy(S.t));

icemin = splitapply(@min, S.extent_N, g);

icemax = splitapply(@max, S.extent_N, g);

icemean = splitapply(@mean, S.extent_N, g);

bnd = [icemean - icemin, icemax - icemean];



cla;

boundedline(dy, icemean, bnd);
```



绘制时间序列时需要注意的一点是，填充对象当前不支持日期时间输入，因此此时也不能使用 boundedline；必须首先将 datetime 输入数据转换为 DateNumber，才能使用此函数进行打印，并且不能以 datetime 轴为目标。

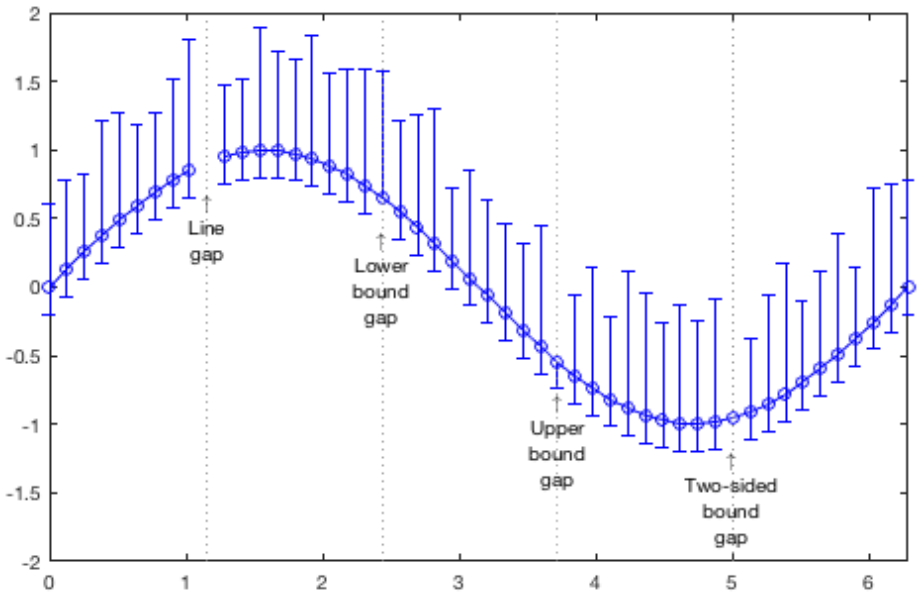
示例 3: 填充间隙

如果绘制一条在 x 或 y 向量中具有一个或多个 NaN 的线，则 NaN 位置将渲染为线中带有间隙的缺失标记。但是，patch 命令不能很好地处理 NAN；如果任何坐标包含 NAN，它根本无法显示填充。
因此，boundedline 的填充部分在边界数组（b）或线的 x/y 坐标（用于计算填充坐标）中遇到 NaN 时的预期行为不明确。我提供了一些选择。
在演示这些选项之前，我将创建一个具有几种不同间隙类型的数据集：

```
x = linspace(0, 2*pi, 50);

y = sin(x);

b = [ones(size(y))*0.2; rand(size(y))*.5+.5]';



y(10)    = NaN;  % NaN 在线上但不是边界
b(20,1) = NaN;  % NaN 在下边界但不是线上
b(30,2) = NaN;  % NaN 在上边界但不是线上
b(40,:) = NaN;  % NaN 在双边界但不是线上
```

下面是 errorbar 图中的情况。

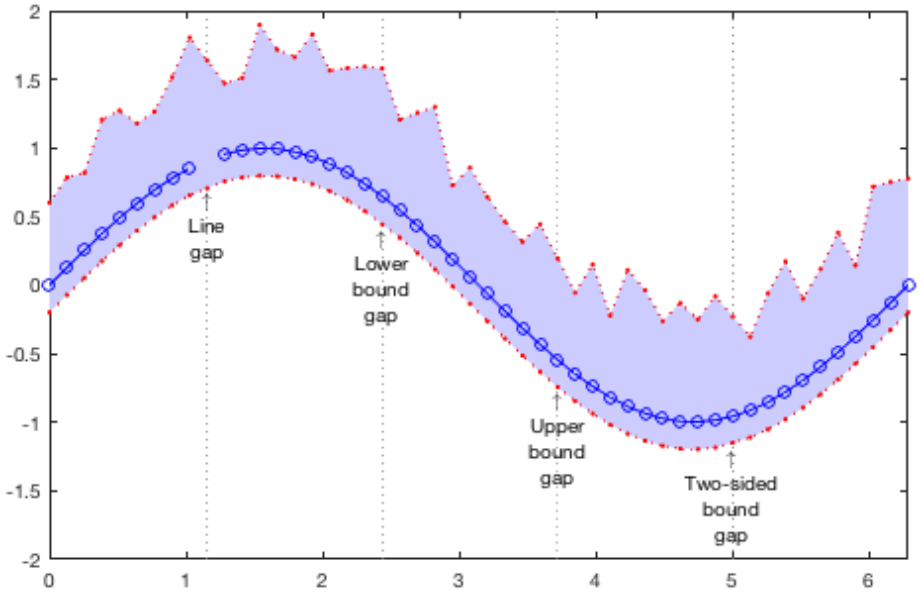
```
figure;

he = errorbar(x,y,b(:,1), b(:,2), '-bo');


line([x([10 20 30 40]); x([10 20 30 40])], [ones(1,4)*-2;ones(1,4)*2], ...
    'color', ones(1,3)*0.5, 'linestyle', ':');
text(x(10), sin(x(10))-0.2, {'\uparrow','Line','gap'}, 'vert', 'top', 'horiz', 'center');
text(x(20), sin(x(20))-0.2, {'\uparrow','Lower','bound','gap'}, 'vert', 'top', 'horiz',
'center');
text(x(30), sin(x(30))-0.2, {'\uparrow','Upper','bound','gap'}, 'vert', 'top', 'horiz',
'center');
text(x(40), sin(x(40))-0.2, {'\uparrow','Two-sided','bound','gap'}, 'vert', 'top', 'horiz',
'center');

axis tight equal;
```



处理 boundedline 中的 NaN 的默认方法是保留线中的间隙，但基于相邻点平滑边界中的间隙。如果您只缺少一个或两个点，并且您不想强调绘图中的这些间隙，则此选项会很好：

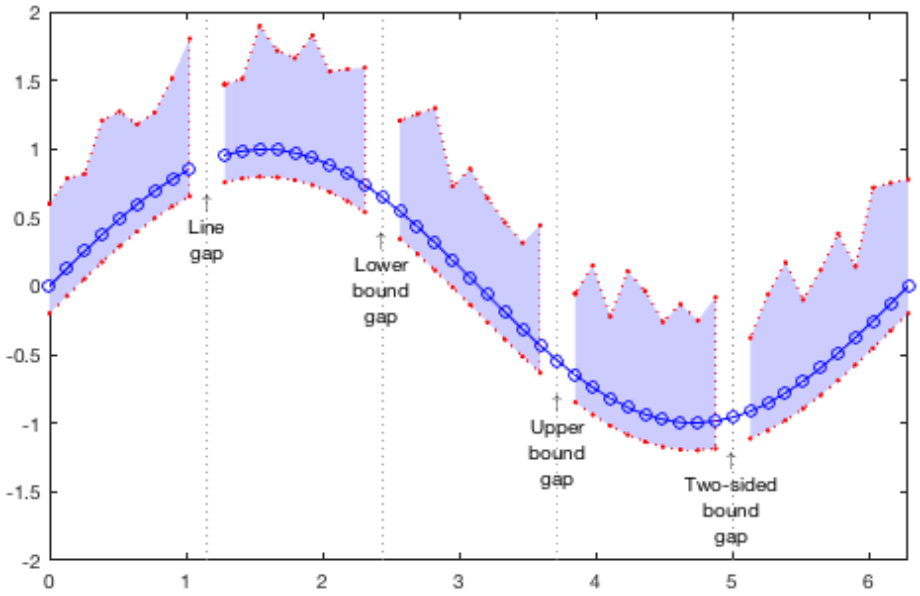
```
delete(he);

[hl,hp] = boundedline(x,y,b,'-bo', 'nan', 'fill');
ho = outlinebounds(hl,hp);
```

```
set(ho, 'linestyle', ':', 'color', 'r', 'marker', '.');
```



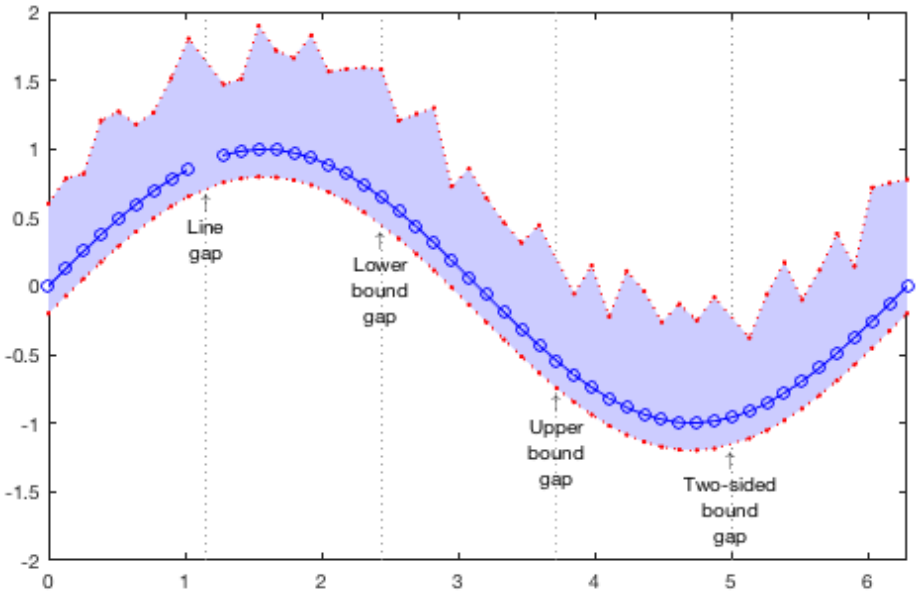
我添加了对比色的边界轮廓，以便您可以看到我如何处理各个点。
第二个选项为任何 NaN 在修补程序中留下完整的间隙。我考虑过允许单边间隙，但想不出区分间隙和零值界的好方法。如果你有任何建议，我愿意接受（给我发电子邮件）。

```
delete([hl hp ho]);
[hl,hp] = boundedline(x,y,b,'-bo', 'nan', 'gap');
ho = outlinebounds(hl,hp);
set(ho, 'linestyle', ':', 'color', 'r', 'marker', '.');
```



最后一个选项将从填充中删除属于 NAN 的点。视觉效果与 fill 选项非常相似，但如果绘制边界轮廓，则缺少的点很明显。

```
delete([hl hp ho]);

[hl,hp] = boundedline(x,y,b,'-bo', 'nan', 'remove');
ho = outlinebounds(hl,hp);
set(ho, 'linestyle', ':', 'color', 'r', 'marker', '.');
```



作者简介

这个函数和支持文档是华盛顿大学的 Kelly A. Kearney 在 2019 年为 Climate Data Toolbox for Matlab 写的。它是这个工具箱的一部分，也可以从 GitHub 单独下载。

subsubplot 文档

subsubplot 在并列位置创建子轴。

语法

```
subsubplot(mm,nn,pp)
subsubplot(m,n,p,mm,nn,pp)
subsubplot(...,'vpad',vspace)
subsubplot(...,'hpad',hspace)
h = subsubplot(...)
```

说明

subsubplot(mm,nn,pp) 将当前图形划分为 mm×nn 网格，并在 pp 指定的位置创建轴。第一个子图是第一行的第一列，第二个子图是第一行的第二列，依此类推。这种语法有效填充 subplot(1,1,1)将占用的区域。

subsubplot(m,n,p,mm,nn,pp) 如上所述，但在这种情况下， subplot(m,n,p)进一步划分为 mm×nn 网格，subsubplot 在 pp 指定的位置创建一组新的轴

subsubplot(...,'vpad',vspace) 指定每个子图轴之间的垂直间距量，单位为 0 到 1 的标准化图形单位。默认值为 0。例如，将其设置为 0.05，将轴分隔为图形高度的 5%。

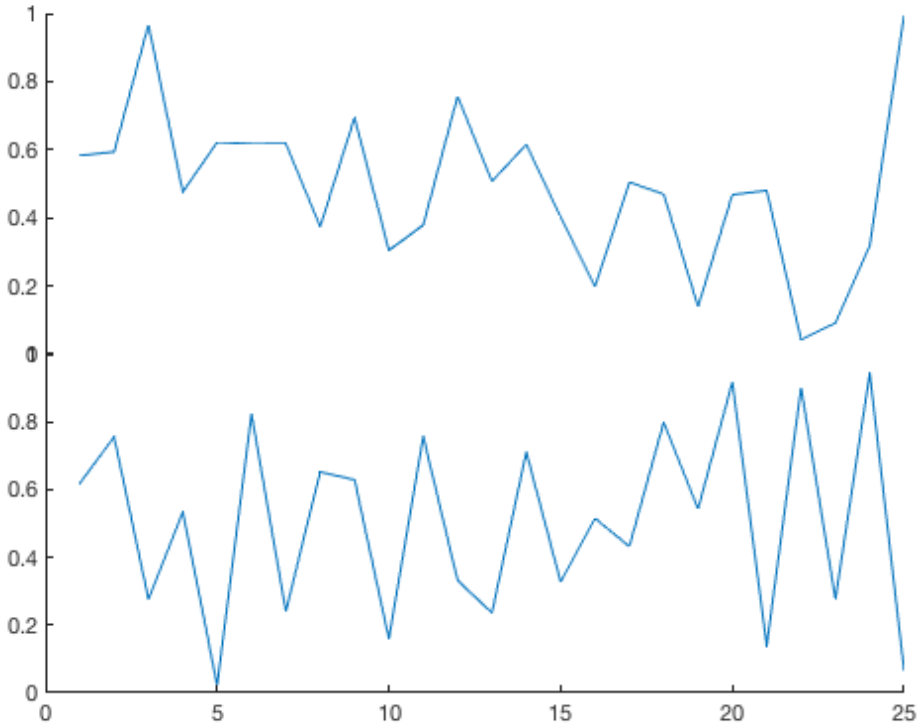
subsubplot(...,'hpad',hspace) 指定每个子图轴之间的水平间距量，单位为 0 到 1 的标准化图形单位。默认值为 0。例如，将其设置为 0.05，将轴分隔为图形宽度的 5%。

h = subsubplot(...) 返回由 subsubplot 创建的轴的句柄 h。

示例 1: 两个子图

以下是两个子图，它们相互叠置：

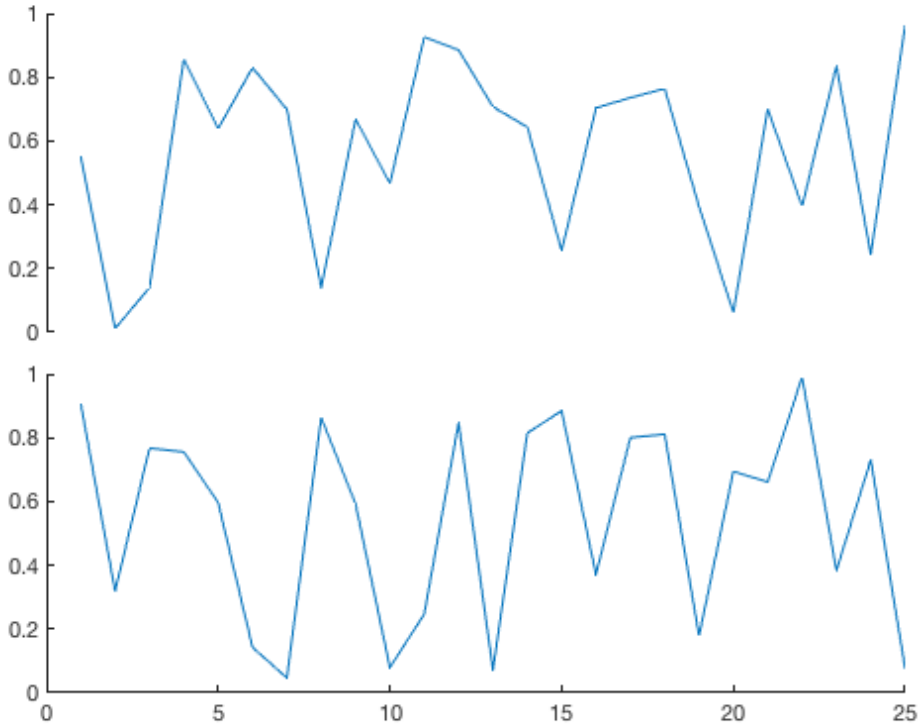
```
figure

subsubplot(2,1,1)

plot(rand(25,1))



subsubplot(2,1,2)

plot(rand(25,1))
```



示例 2: 垂直间隔：

默认情况下，subsubplot 将轴直接相邻放置。要在两个轴之间添加一点填充，请指定'vpad'。下面我们指定轴之间的间距应为图形高度的 5%：

```
figure

subsubplot(2,1,1,'vpad',0.05)
plot(rand(25,1))

subsubplot(2,1,2,'vpad',0.05)
plot(rand(25,1))
```

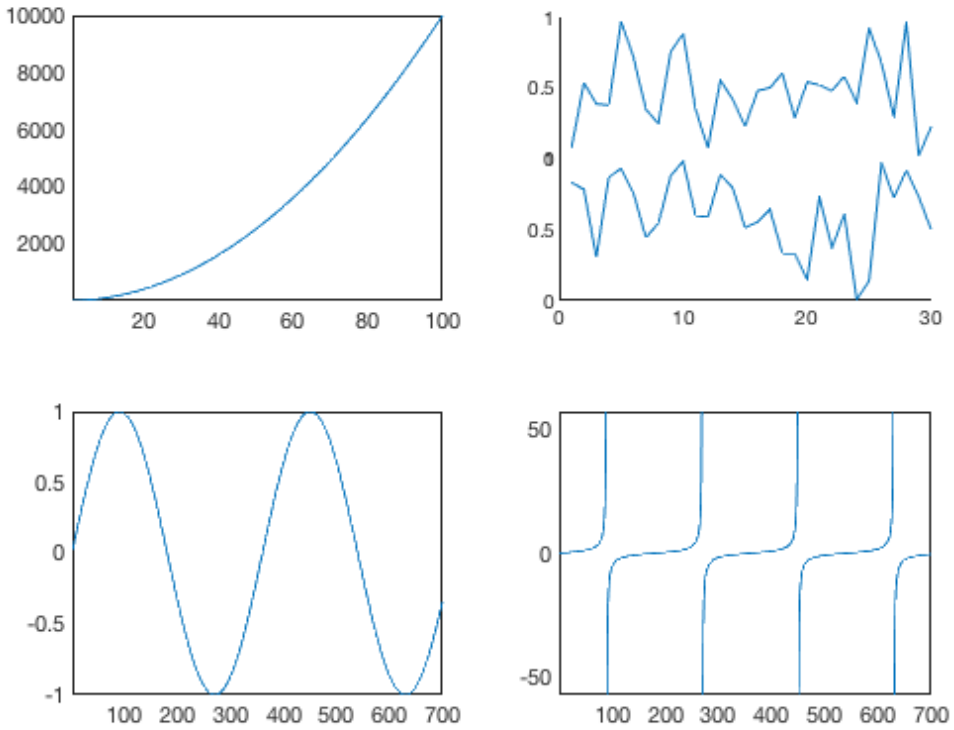


示例 3: 子图套子图

subsubplot 函数设计为在 Matlab 的内置 subplot 结构中工作。下面，我们使用 subplot(2, 2, 2) 将占用的空间创建两个子地块，由 subsubplot(2, 2, 2, 2, 1, 1) 和 subsubplot(2, 2, 2, 2, 2, 2) 指定：

```
figure



% 左上子图:
subplot(2,2,1)
plot((1:100).^2)
axis tight

% * * * 右上子图: * * *
% 上 subsublot:
subsubplot(2,2,2,2,1,1)
plot(rand(30,1))

% 底 subsublot:
subsubplot(2,2,2,2,1,2)
plot(rand(30,1))

% 左下 subplot:
```

```
subplot(2,2,3)
plot(sind(1:700))
axis tight

% 右下 subplot:
subplot(2,2,4)
plot(tand(1:700))
axis tight
```

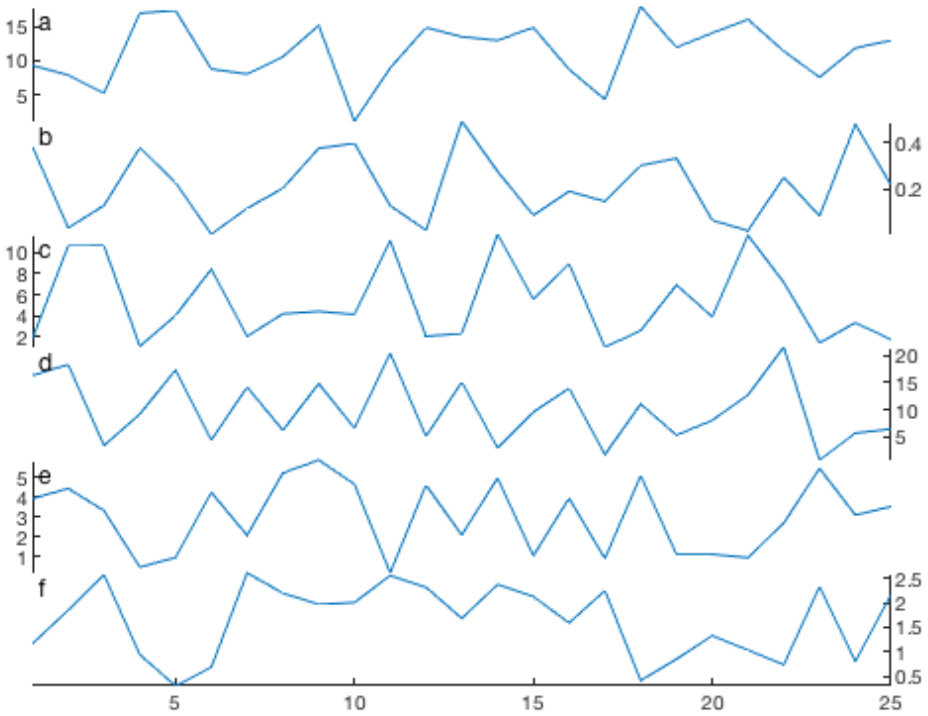


示例 4: 右手轴:

为了防止 y 轴重叠，您可以使用’vpad’选项在轴之间添加一些空间，也可以将每隔一个 y 轴放置在右侧，如下所示：

```
% 每个组件添加标签:
panel = {'a','b','c','d','e','f'};

figure
for k = 1:6
   ax(k) = subsubplot(6,1,k);
   plot(rand*30*rand(25,1)) % 绘制随机数据
   axis tight               % 移除空白

   ntitle(panel{k},'location','nw')
end
```

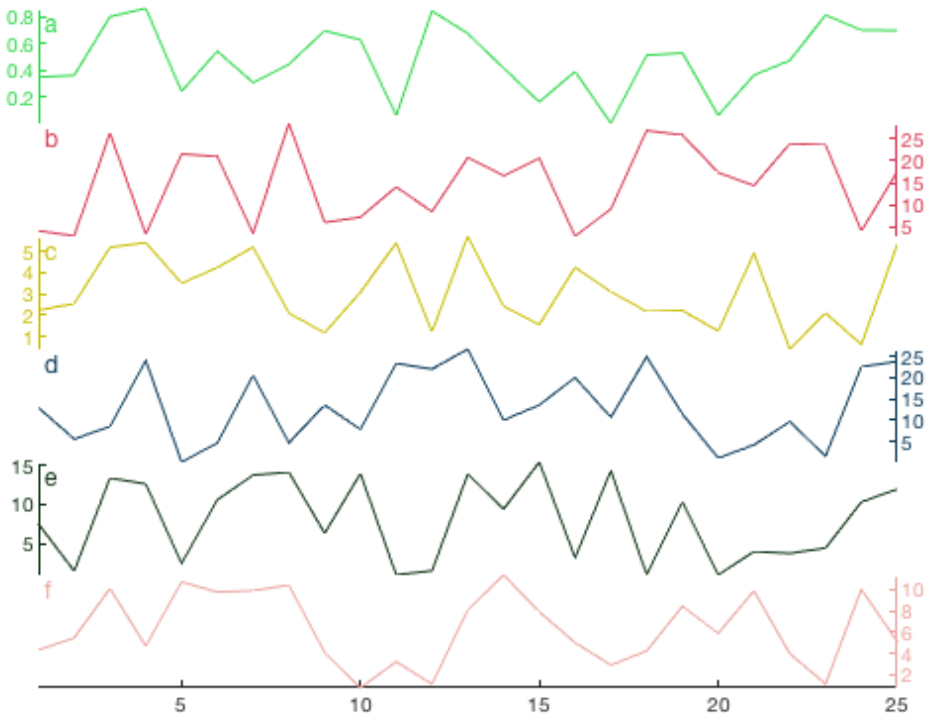
```
% 将其他轴置于右侧：
set(ax(2:2:end),'YAxisLocation', 'right')
```



示例 5: 彩色编码轴

同上，但这一次使轴颜色与线颜色匹配：

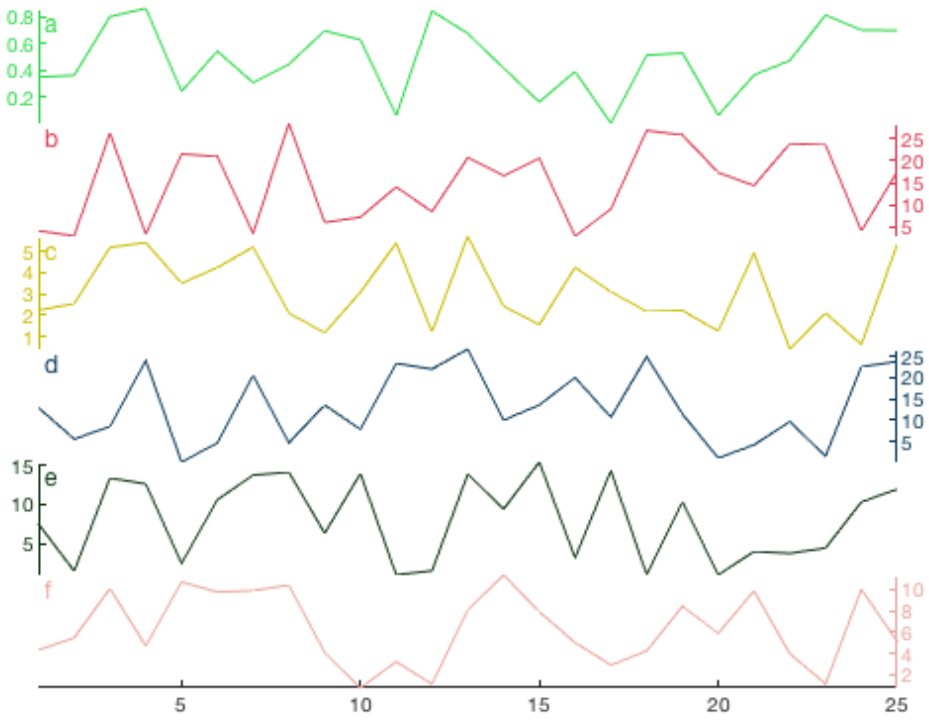
```
figure

for k = 1:6
   ax(k) = subsubplot(6,1,k);

   col = rand(1,3);
   plot(rand*30*rand(25,1),'color',col)
   set(gca,'ycolor',col)
   axis tight
   ntitle(panel{k},'location','nw','color',col)
end

set(ax(2:2:end),'YAxisLocation', 'right')
```



提示：linkaxes

有时，使用鼠标（或通过编程设置轴限制）放大 subsubplot 时，您希望其他 subsubplot 的范围自动缩放到相同的范围。要实现这种行为，请获取每个 subsubplot 的句柄并使用 linkaxes。对于上图，链接所有 x 轴范围很容易：

```
linkaxes(ax,'x')
```

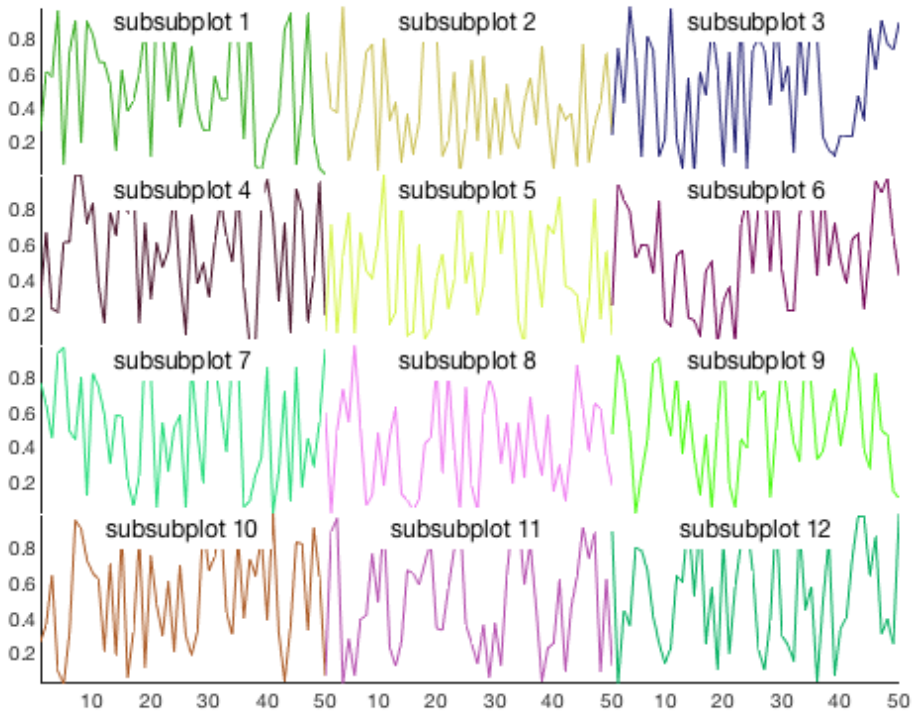


现在您可以缩放上面的任何子图，图形中的所有其他轴将缩放到相同的范围。

示例 6:一个 subsubplot 网格

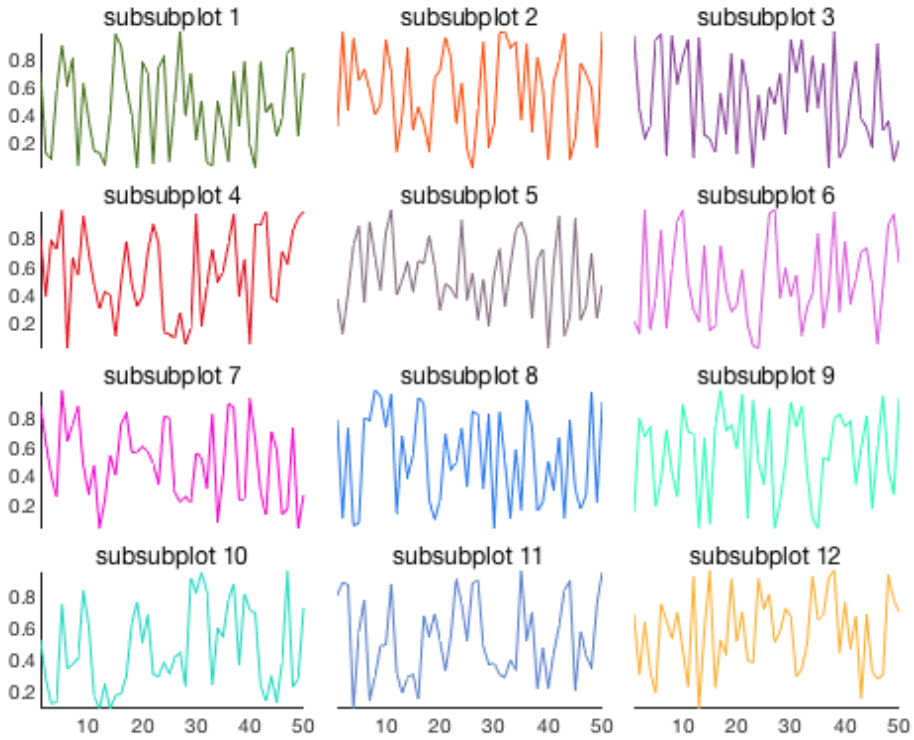
这是一个 4×3 的随机数据网格，用 subsubplot 绘制。我们还将使用 ntitle 在每个 subsubplot 中包含小标题。

```
figure



for k = 1:12
   % 初始化 subsubplots:
   subsubplot(4,3,k)

   % 使用随机颜色绘制一些随机数据:
   plot(rand(50,1),'color',rand(1,3))
   axis tight

   ntitle(['subsubplot ',num2str(k)],'backgroundcolor','w')
end
```



每个 subsubplot 之间没有足够的空白？添加一点水平和垂直填充，如下所示：

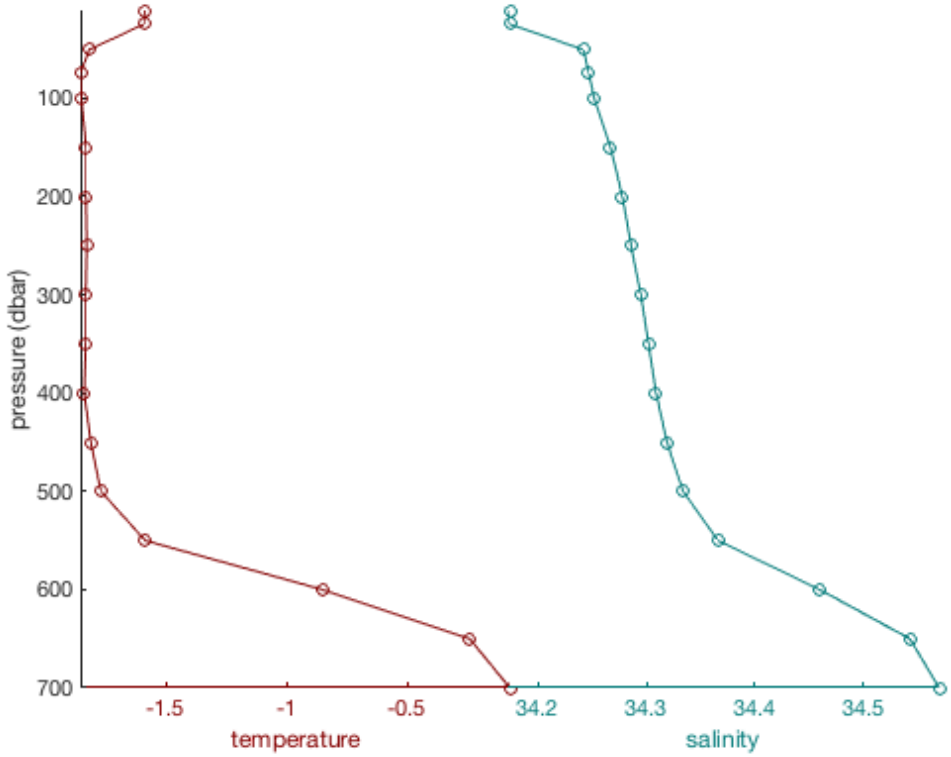
```
figure



for k = 1:12

   % 初始化 subsubplots:
   subsubplot(4,3,k,'hpad',0.03,'vpad',0.05)

   % 使用随机颜色绘制一些随机数据:
   plot(rand(50,1),'color',rand(1,3))
   axis tight



   ntitle(['subsubplot ',num2str(k)],'vert','bottom')
end
```



示例 7: 海洋剖面

加载一些示例 CTD 数据并绘制温度和盐度图:

```
load example_ctd

% 初始化图像:
figure

% 绘制温度:
subsubplot(1,2,1)
plot(T{1},P{1},'o-','color',rgb('dark red'))
axis tight ij % 移除空白并翻转坐标轴
ylabel 'pressure (dbar)'
xlabel 'temperature'
set(gca,'xcolor',rgb('dark red'))

% 绘制盐度:
subsubplot(1,2,2)
plot(S{1},P{1},'o-','color',rgb('bluegreen'))
axis tight ij % 移除空白并翻转坐标轴
xlabel 'salinity'
set(gca,'xcolor',rgb('bluegreen'))
```

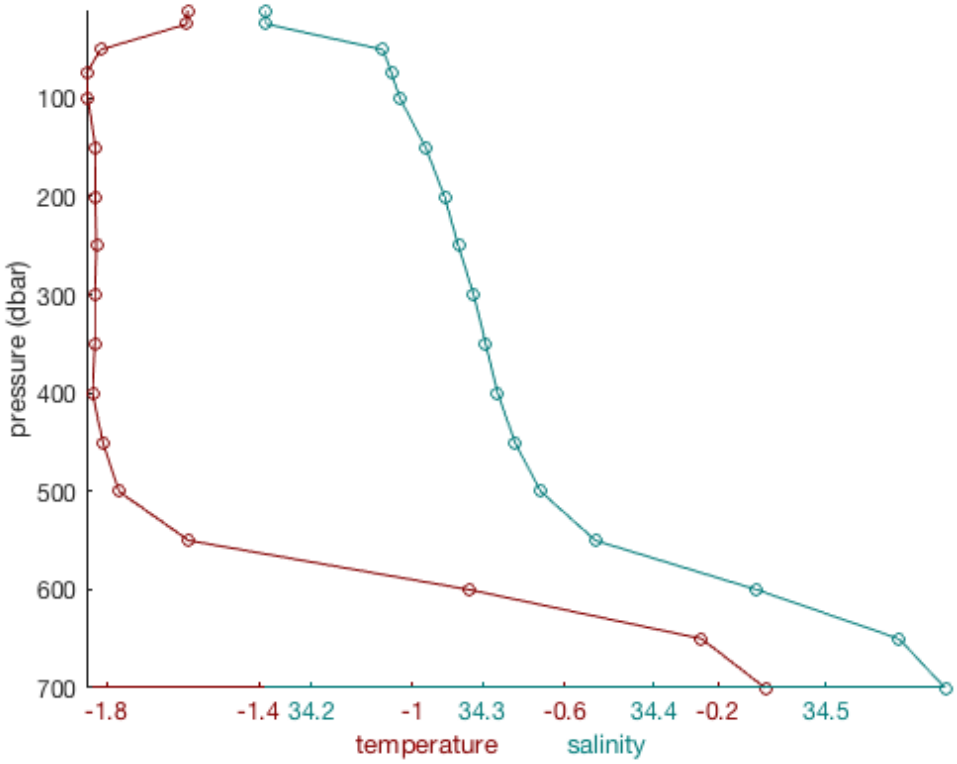


示例 8: 重叠轴

在上面的图中，两个子图直接邻接，但两个剖面之间仍有大量空白。要将它们组合在一起，请通过将'hpad'设置为负值使两个子图重叠。请注意，如果执行此操作，则需要使每个轴的背景透明（而不是默认的白色）。使用 set(gca,'color','none')使轴背景透明：

```
% 初始化图像:
figure

% 绘制温度:
subsubplot(1,2,1,'hpad',-0.45)
plot(T{1},P{1},'o-','color',rgb('dark red'))
axis tight ij % 移除空白并翻转坐标轴
ylabel 'pressure (dbar)'
xlabel 'temperature'
set(gca,'xcolor',rgb('dark red'),'color','none',...
   'xtick',-1.8:0.4:0)

% 绘制盐度:
subsubplot(1,2,2,'hpad',-0.45)
plot(S{1},P{1},'o-','color',rgb('bluegreen'))
axis tight ij % 移除空白并翻转坐标轴
xlabel 'salinity'
set(gca,'xcolor',rgb('bluegreen'),'color','none',...
```

```
    'xtick',34.1:0.1:34.6)
```



作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

spiralplot 文档

spiralplot 函数生成一个螺旋图，显示时间序列的年度性质和长期趋势。
这个函数受雷丁大学的 Ed Hawkins 的作品启发。更多关于螺旋图的信息可以在这里找到。

语法

```
spiralplot(t,z)
spiralplot(...,'LineWidth',LineWidth)
spiralplot(...,'zmin',zmin)
spiralplot(...,'ztick',Zticks)
spiralplot(...,'format',MonthFormat)
spiralplot(...,'fontsize',LabelFontsize)
spiralplot(...,'nospokes')
[h,hax,htxt] = spiralplot(...)
```

说明

`spiralplot(t,z)` 将 z 的时间序列绘制为螺旋图，其中时间为 datetime 或 datenum 格式。
`spiralplot(...,'LineWidth',LineWidth)` 指定螺旋打印的线宽（LineWidth）。默认线宽为 2。
`spiralplot(...,'zlim',zlim)` 指定 z 轴上的最小值和最大值。默认 zlim 对应于 z 的范围；但是，根据应用的不同，将 0 或其他一些临界值放在图形的中心可能是有意义的。
`spiralplot(...,'ztick',Zticks)` 指定 z 轴线和标签的值。
`spiralplot(...,'format',MonthFormat)` 使用 datestr 格式选项设置月份标记方式的格式。月份格式的选项包括：

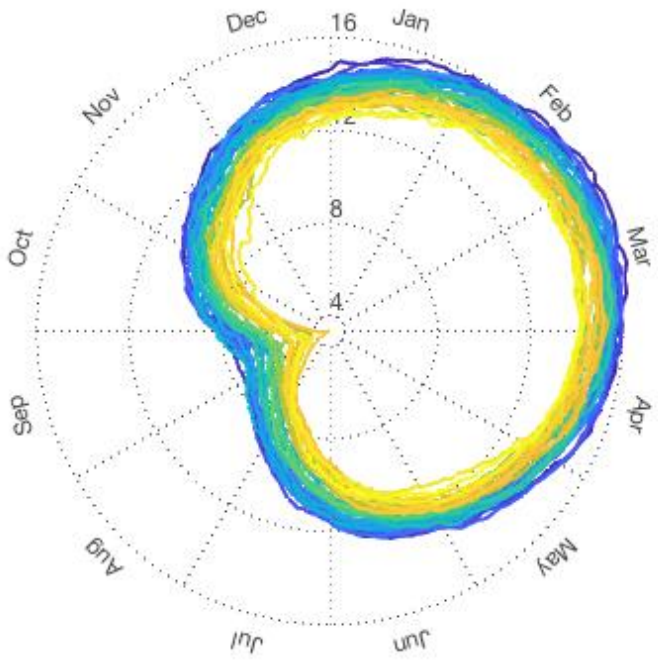
- 'mmmm' 全名, (例, March, December)
- 'mmm' 前三个字母, (例, Mar, Dec)
- 'mm' 两位数字, (例, 03, 12)
- 'm' 首字母大写, (例, M, D)

`spiralplot(...,'fontsize',LabelFontsize)` 指定月份名称和 z 轴标签的字体大小。
`spiralplot(...,'nospokes')` 从轴上删除辐条。
`[h,hax,htxt] = spiralplot(...)` 返回颜色尺度线、轴对象和文本对象的句柄

示例 1: 简单的

本例使用 NSIDC 提供的海冰范围数据（见下文参考资料）。
加载示例数据并绘制北半球海冰范围：

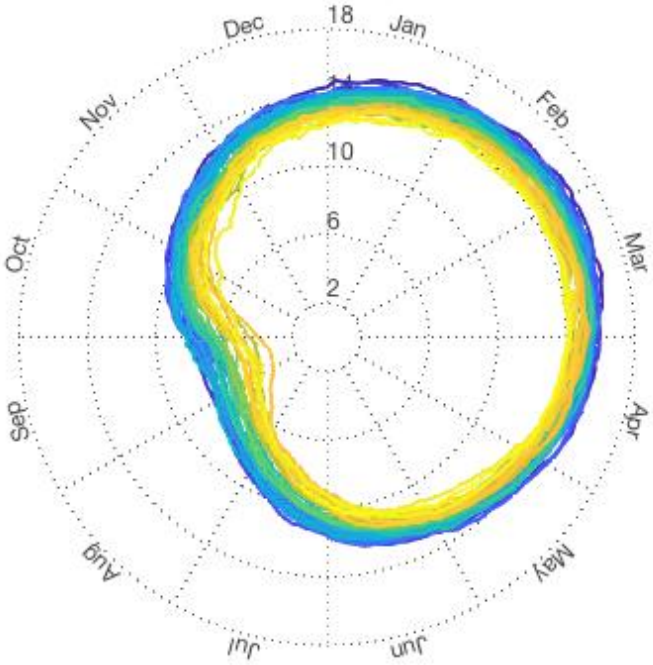
```
load seaice_extent


spiralplot(t,extent_N)
```



示例 2: 指定 z 轴范围

上面的图并没有显示海冰范围有多接近零。因此，最好将零放在圆的中心：

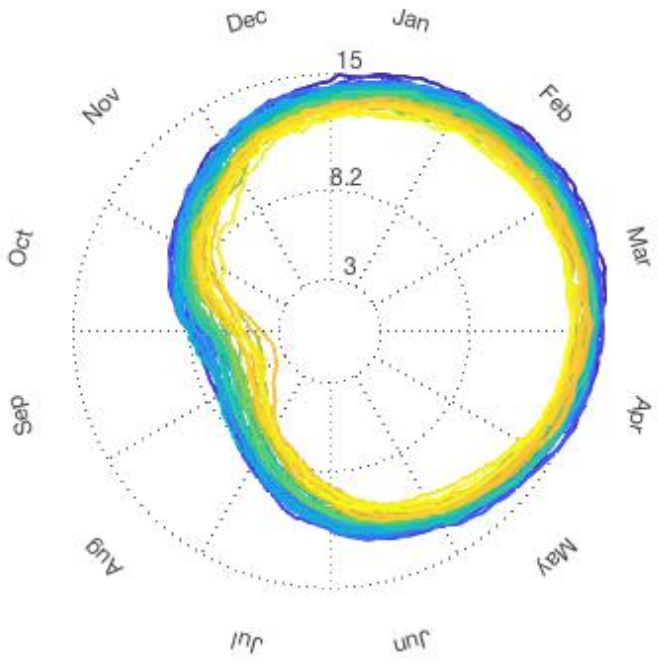
```
figure

spiralplot(t,extent_N,'zlim',[0 18])
```



示例 3: 指定 z 刻度:

也许您希望标记值 3、8.2 和 15，而不是使用默认的记号标签计算。以下是方法：

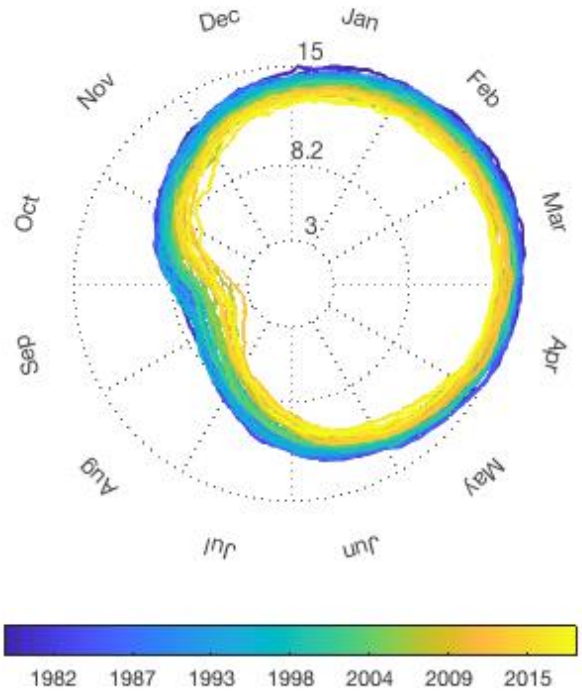
```
figure

spiralplot(t,extent_N,'zlim',[0 18],'ztick',[3 8.2 15])
```



示例 4: 加一个时间颜色条

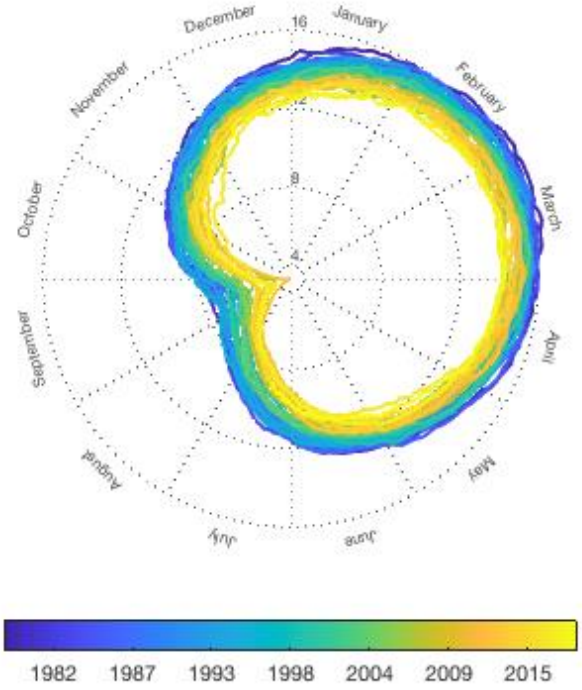
要包含显示时间的颜色条，请使用 cbdate 函数。

```
cb = colorbar('location','southoutside');
cbdate('yyyy','horiz')
```



示例 5: 设置轴标签文本的格式

```
cla % 清除上面的螺旋图，但保留颜色条
spiralplot(t,extent_N,'fontsize',8,'format','mmmm')
```



示例 6: 做个动图

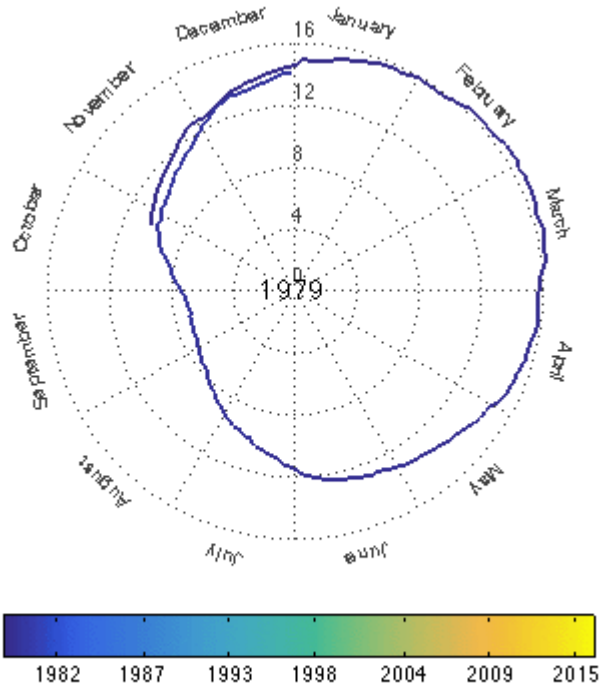
使用 spiralplot 和 gif 函数制作动图非常简单。只需像这样初始化第一帧：

```
years = 1980:2017;
figure('position',[100 100 377 420])
% 放置一个颜色条:
cb = colorbar('southoutside');
set(cb,'fontsize',10)
% 将颜色轴设置为完整日期范围：
caxis(datenum([min(t) max(t)]))
% 将颜色条轴设置为类似日期的格式:
cbdate('yyyy','horiz')
% 获取 1980 年 1 月 1 日之前所有日期的索引:
ind = t<datetime(years(1),1,1);
% 画螺旋图
spiralplot(t(ind),extent_N(ind),'zlim',[0 17],...
    'fontsize',10,'format','mmmm')
% 将文本标签放置在中心:
text(0,0,'1979','vert','middle','horiz','center')
% 编写新的.gif 动画的第一帧:
gif('seaice_extent.gif','frame',gcf)
```

设置第一帧后，循环剩余年份并保存每个帧：

```
%循环随后的每一年:
for k = 2:length(years)
    cla % 清除旧图
    % 获取第 k 年之前所有日期的索引:
    ind = t<datenum(years(k),1,1);
    % 画螺旋图
    spiralplot(t(ind),extent_N(ind),'zlim',[0 17],...
        'fontsize',10,'format','mmmm')
    % 将文本标签放置在中心:
    text(0,0,num2str(years(k)-1),...
        'vert','middle','horiz','center')
    % 保存这一帧:
    gif
end
```

结果如下:



参考资料

Fetterer 等人的海冰数据可在 NSIDC 获得。

Fetterer, F., K. Knowles, W. Meier, and M. Savoie. 2016, updated daily. Sea Ice Index, Version 2. Boulder, Colorado USA. NSIDC: National Snow and Ice Data Center. doi:10.7265/N5736NV7.

作者简介

这个函数是来自德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 于 2017 年 6 月写的。

plotpsd 文档

该函数使用周期图函数绘制时间序列的功率谱密度。**需要 Matlab 的信号处理工具箱（Signal Processing Toolbox）**。

语法

```
plotpsd(y,Fs)
plotpsd(y,x)
plotpsd(...,LineProperty,LineValue)
plotpsd(...,'logx')
plotpsd(...,'db')
plotpsd(...,'lambda')
h = plotpsd(...)
```

说明

`plotpsd(y,Fs)` 使用 `periodogram` 函数在采样频率 `Fs` 处绘制一维数组 `y` 的功率谱。采样频率 `Fs` 必须是标量。

`plotpsd(y,x)` 绘制参考自变量 `x` 的 `y` 功率谱。这语法要求 `x` 和 `y` 的长度相等，且 `x` 的必须等间距且单调递增。对于时间序列，`x` 可能有时间单位；对于空间分析，`x` 可以具有长度单位。

`plotpsd(...,LineProperty,LineValue)` 使用 LineSpec 属性的任意组合（例如，`'color','r','linewidth',2` 等）指定打印的线样式。

`plotpsd(...,'logx')` 指定了 `semilogx` 绘图。

`plotpsd(...,'db')` 以分贝为单位绘制功率谱。

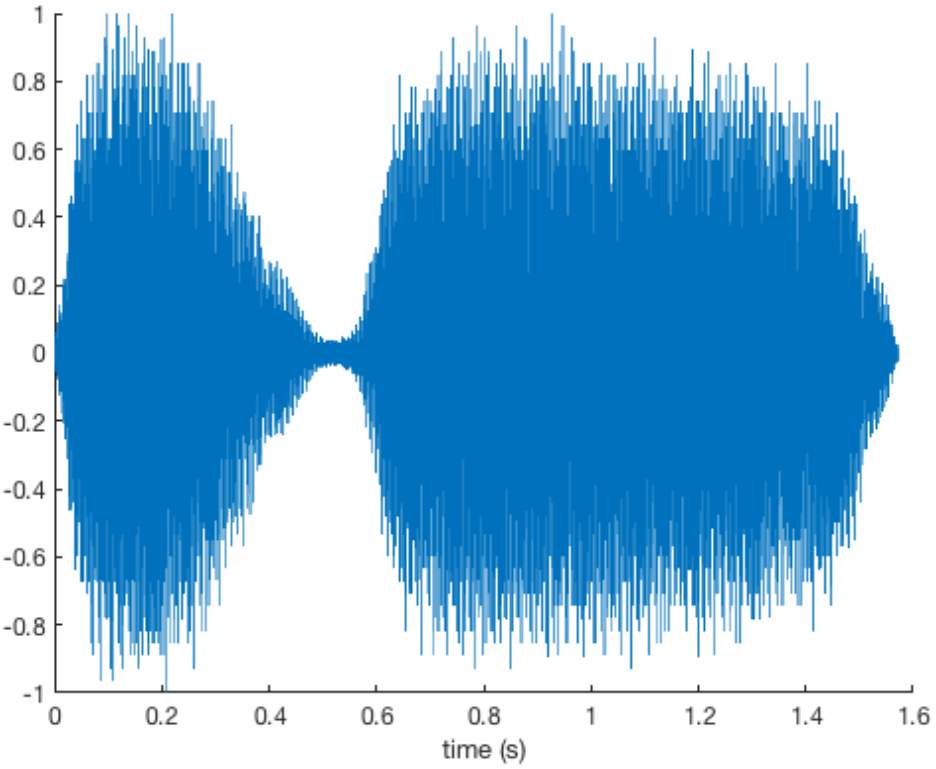
`plotpsd(...,'lambda')` 将水平轴标记为波长，而不是默认频率。注意，这个语法假设 **λ=1/f**。

`h = plotpsd(...)` 返回打印图形对象的句柄 `h`。

示例 1: 火车汽笛

使用内置列车汽笛（`train`）示例，通过绘制时间序列开始绘制：

```
load train
t = (0:length(y)-1)/Fs;

plot(t,y)
box off
xlabel 'time (s)'
```

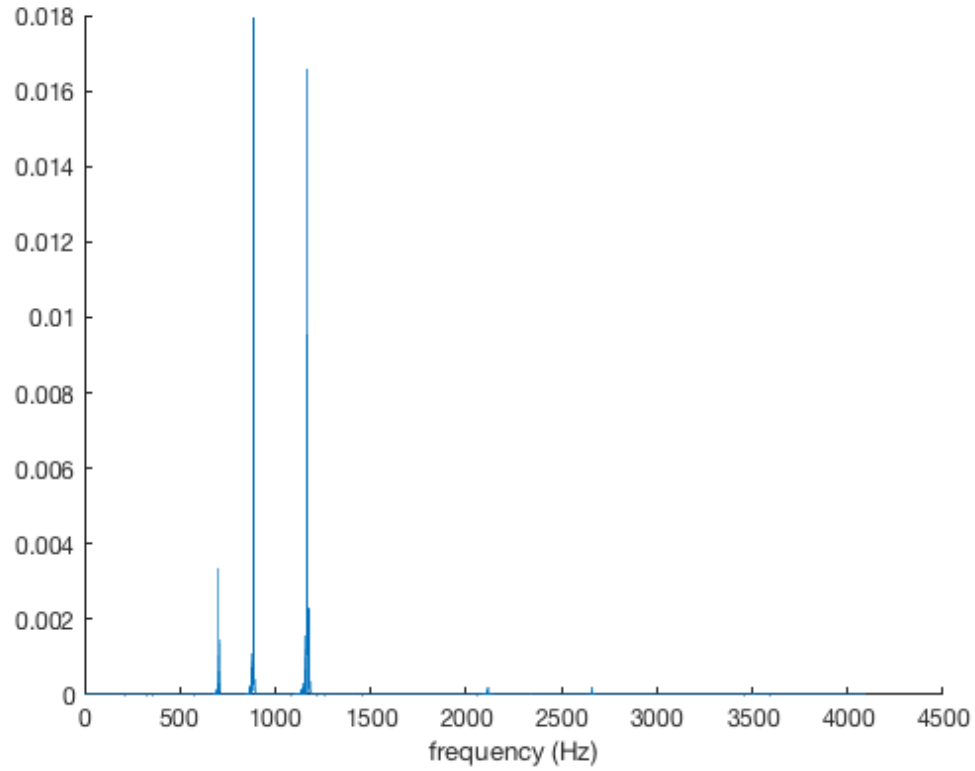


如果你戴着耳机，你可以像这样播放火车汽笛：

```
soundsc(y,Fs)
```

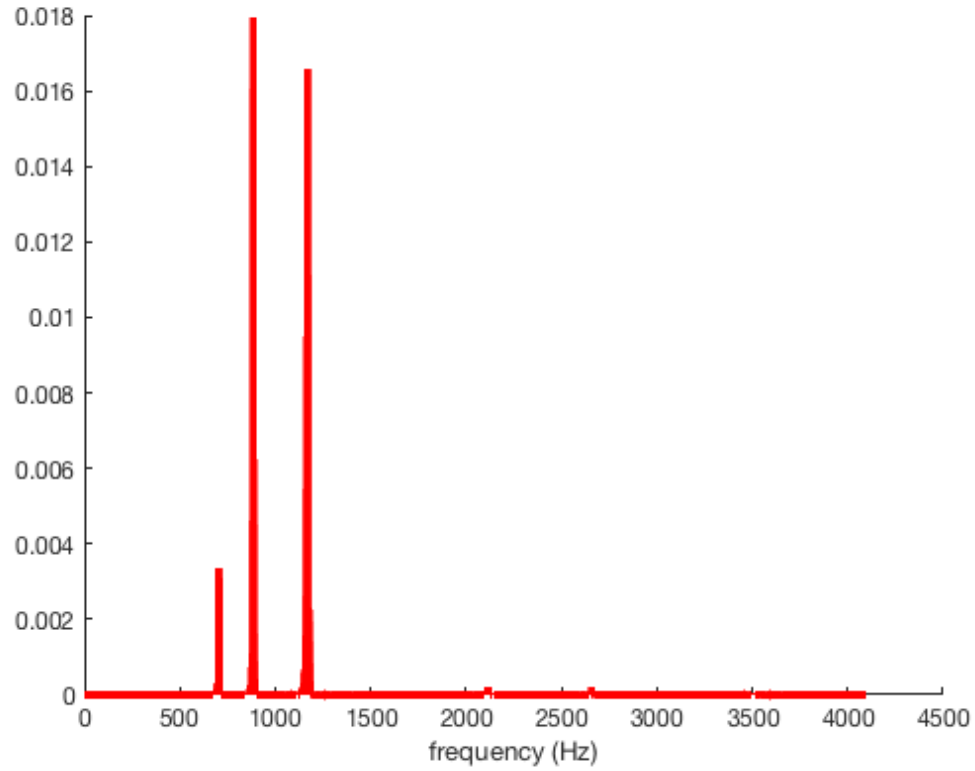
列车信号的功率谱如下所示：

```
plotpsd(y,Fs)

xlabel 'frequency (Hz)'
```



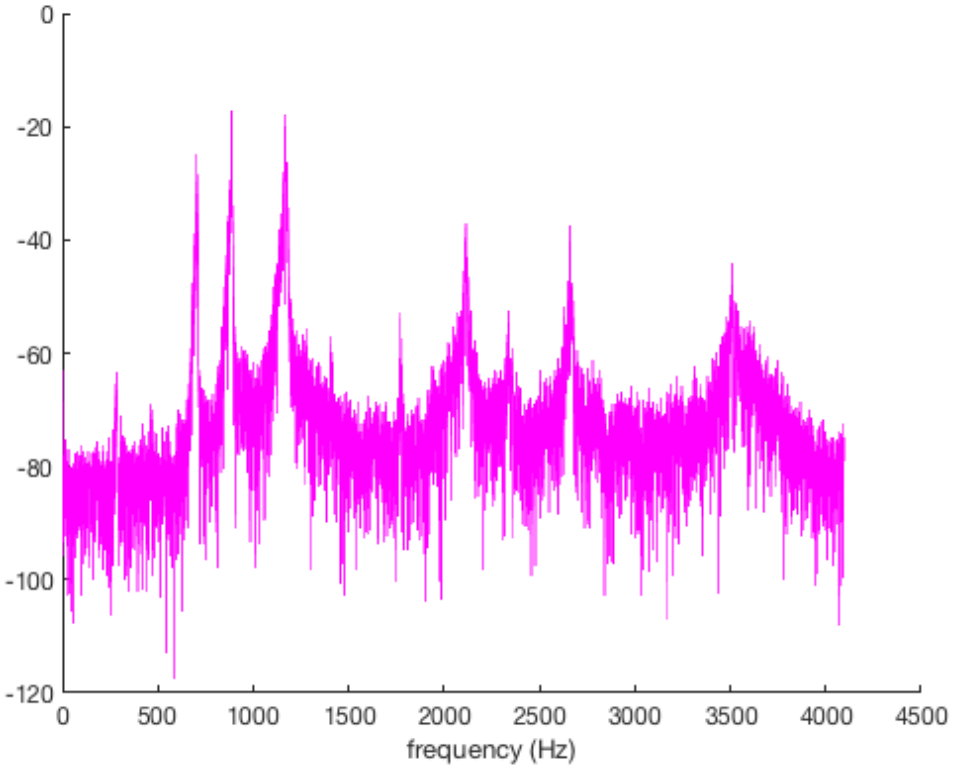
这使得火车的三个喇叭非常清晰！不喜欢默认的细蓝线吗？画一条粗红线：

```
plotpsd(y,Fs,'color','red','linewidth',4)
xlabel 'frequency (Hz)'
```



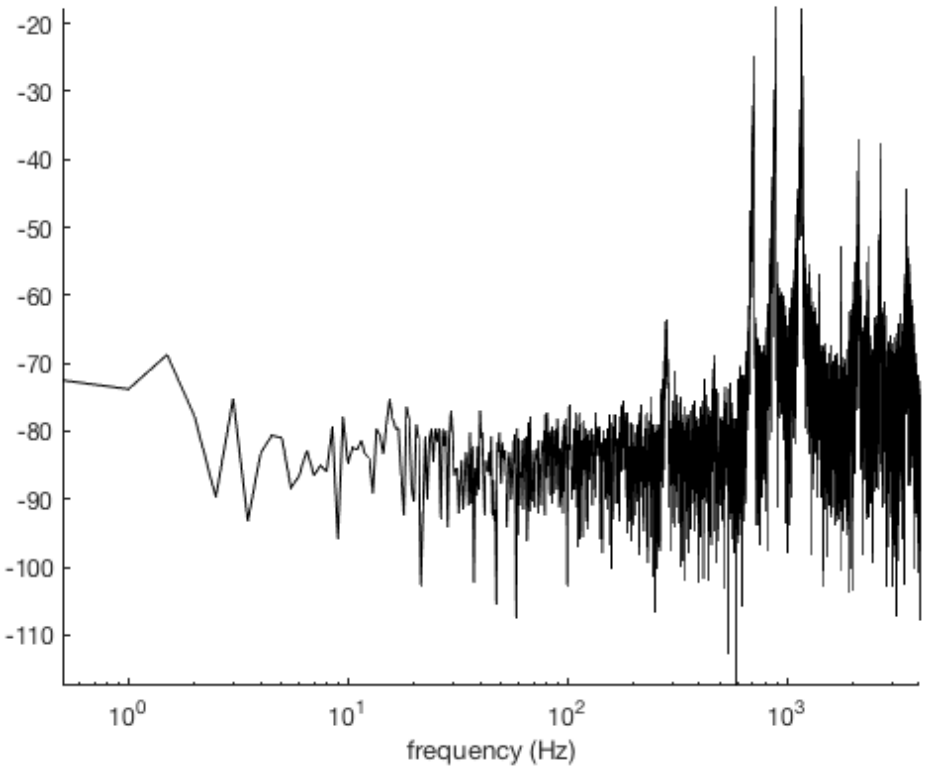
想把它看作是以分贝为单位绘制的洋红色线条吗？、

```
plotpsd(y,Fs,'m','db')
xlabel 'frequency (Hz)'
```



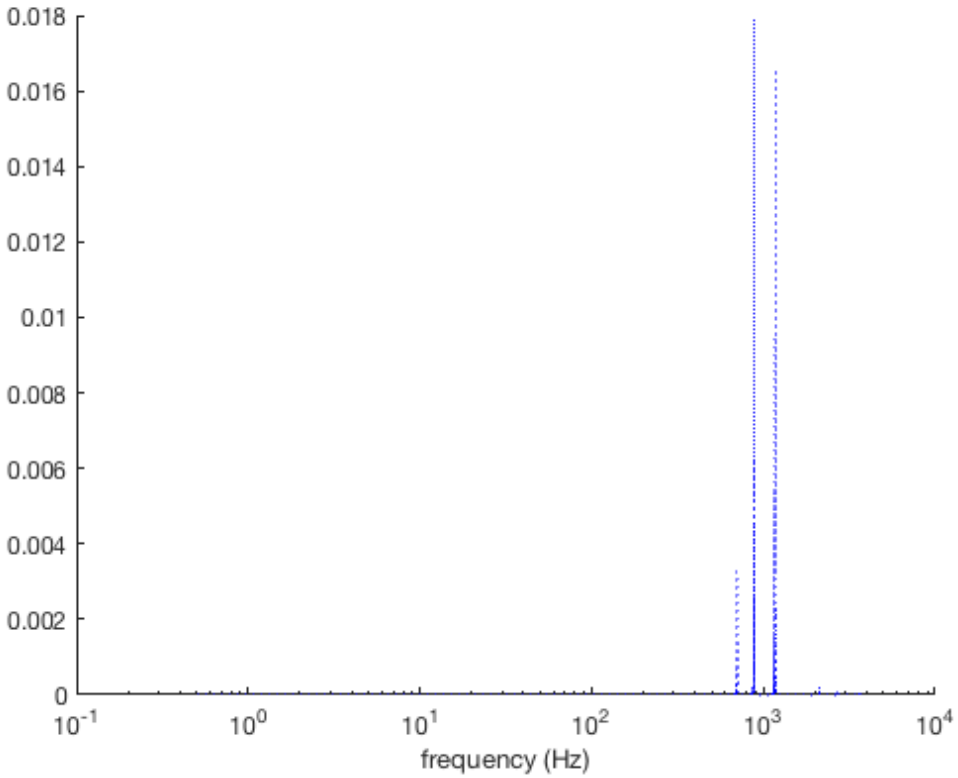
这是一条黑线，垂直方向上有分贝，水平方向上有对数刻度。

```
plotpsd(y,Fs,'k','db','logx')
axis tight
xlabel 'frequency (Hz)'
```



假设你有一些测量 y 与某个时间向量 t 相关联，并且你不想计算采样率。如果是这种情况，只需输入 t 而不是 Fs：

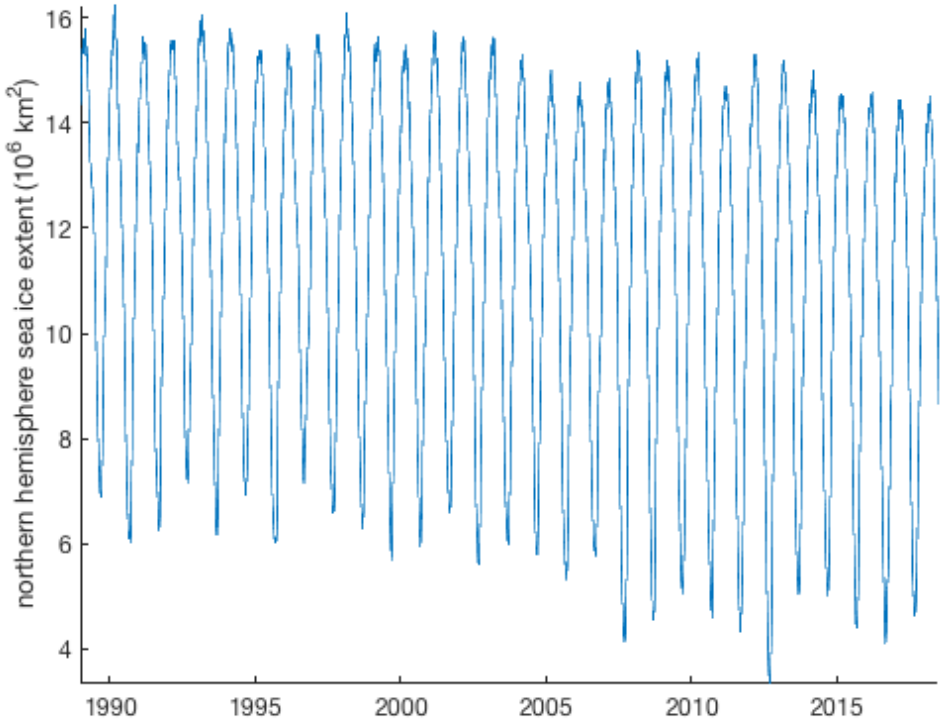
```
plotpsd(y,t,'b:','logx')
xlabel 'frequency (Hz)'
```



示例 2: 海冰范围

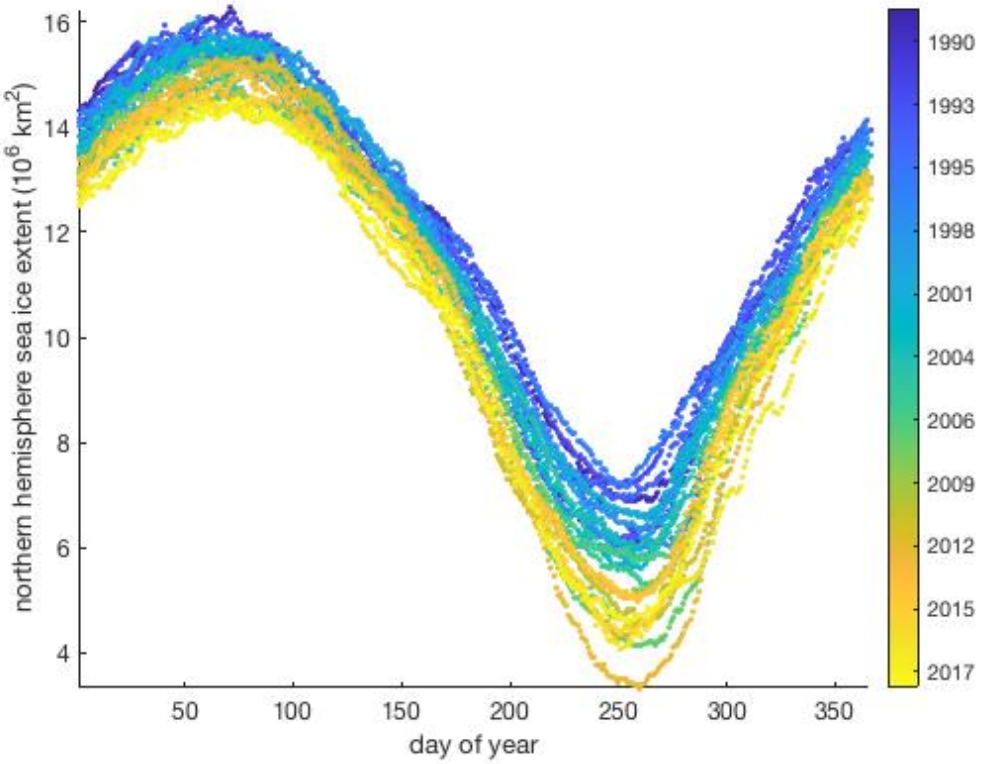
加载 CDT 附带的海冰范围数据，仅使用 1988 年之后的数据，因为之前的数据不是每日分辨率：

```
load seaice_extent.mat % CDT 附带的一些示例数据

% 1989 年以来的日期索引:
ind = t>datetime(1989,1,1);

figure
plot(t(ind),extent_N(ind))
ylabel 'northern hemisphere sea ice extent (10^6 km^2)'
box off % 移除丑陋的对位框架
axis tight
```



这显然有一定的周期性。我们可以使用 doy 绘制一个所有与儒略日相关数据的散点图：

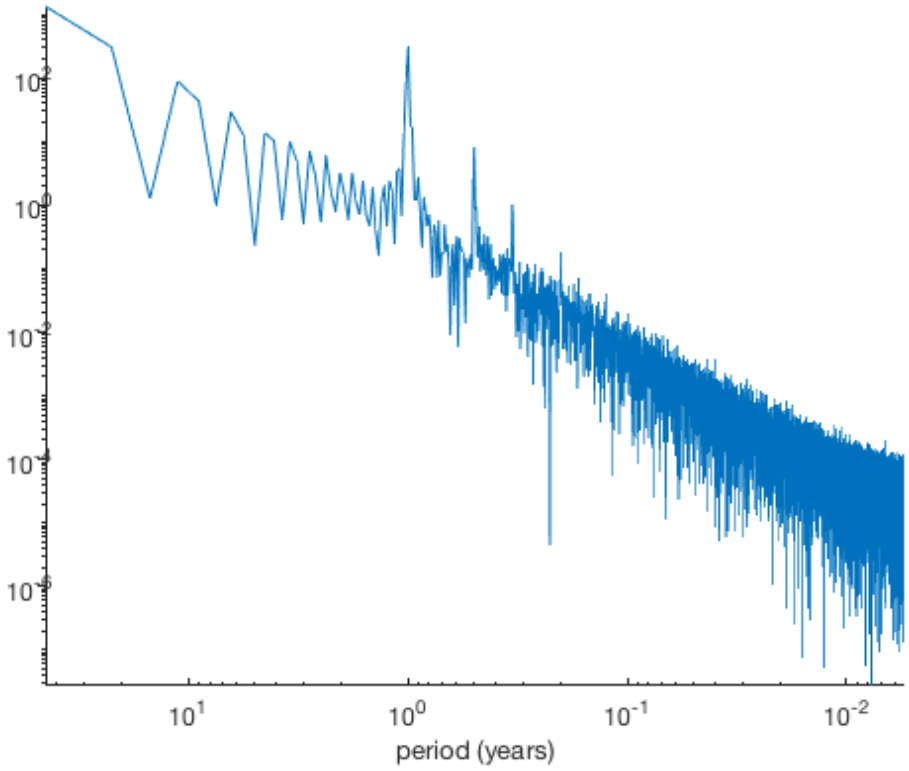
```
jday = doy(t);



scatter(jday(ind),extent_N(ind),10,datenum(t(ind)),'filled')
cb = cbdate('yyyy');     % 将颜色条格式化为日期
set(cb,'ydir','reverse') % 翻转颜色条轴
axis tight
ylabel 'northern hemisphere sea ice extent (10^6 km^2)'
xlabel 'day of year'
```



从上面的两幅图中，我们可以预期海冰范围时间序列在 1/yr 频率上具有能量，此外还有一些长期趋势，这些趋势应表示为低频宽带能量。

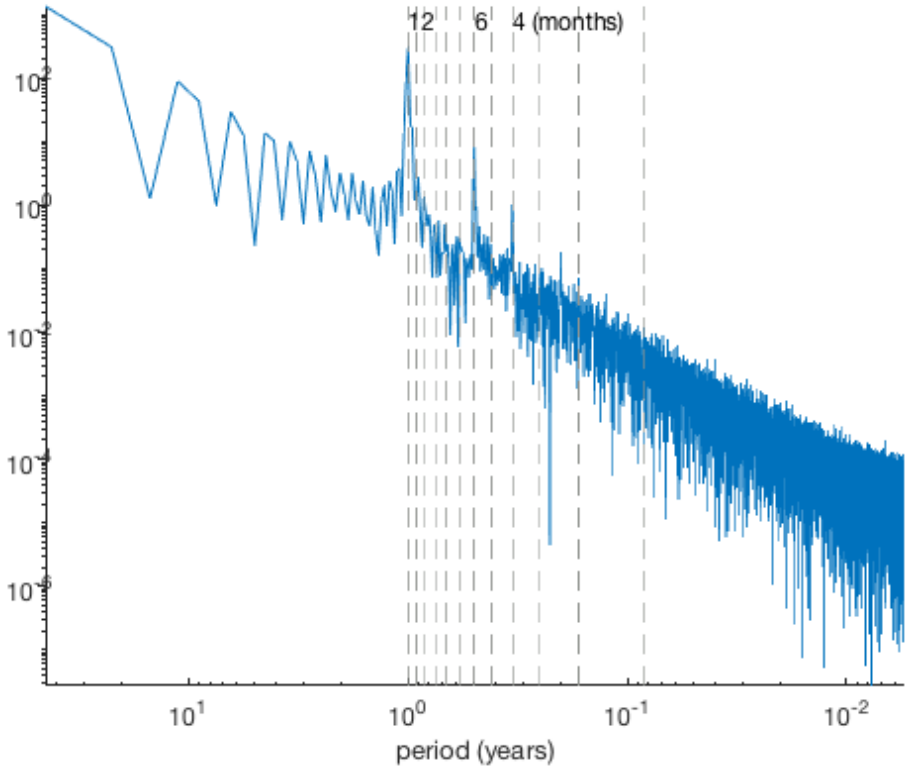
假设采样频率为每年 365.25 个等距采样，将功率谱密度绘制为周期的函数（此处为 lambda）：

```
figure

plotpsd(extent_N(ind),365.25,'lambda')
set(gca,'xscale','log','yscale','log')
axis tight
xlabel 'period (years)'
```



正如所料，1 年期的峰值尤为突出。但也有其他一些小高峰，特别是在 6 个月和 4 个月期间。使用 vline 显示他们：

```
vline((1:12)/12,'--','color',rgb('gray'))

yl = ylim; % 图中 y 的限制
text(12/12,yl(2),'12','vert','top')
text(6/12,yl(2),'6','vert','top')
text(4/12,yl(2),'4 (months)','vert','top')
```



作者简介

这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 于 2015 年 10 月写的。2019 年为 Climate Data Toolbox 改编。

polyplot 文档

该函数绘制一个拟合散乱的 *x*,*y* 数据的多项式。此函数可用于轻松地将线性趋势线或其他多项式拟合添加到数据图中。

语法

```
polyplot(x,y)
polyplot(x,y,n)
polyplot(...,'Name',Value,...)
polyplot(...,'error')
[h,p,delta] = polyplot(...)
```

说明

`polyplot(x,y)` 在散乱的 *x*,*y* 数据中放置一条最小二乘线性趋势线。
`polyplot(x,y,n)` 指定与 *x*,*y* 数据拟合的多项式的阶数 `n`。默认值 `n` 为 1。
`polyplot(...,'Name',Value,...)` 使用 LineSpec 属性名称-值对设置线型格式（例如，`'linewidth',3`）。如果打印'error'边界，则只接受 `boundedline` 属性。
`polyplot(...,'error')` 包括与误差增量的大约+/-1 标准偏差 `delta` 相对应的线条。至少 50%的数据应位于误差线范围内。错误边界用 `boundedline` 绘制。
`h = polyplot(...)` 返回绘制对象的句柄 `h`。

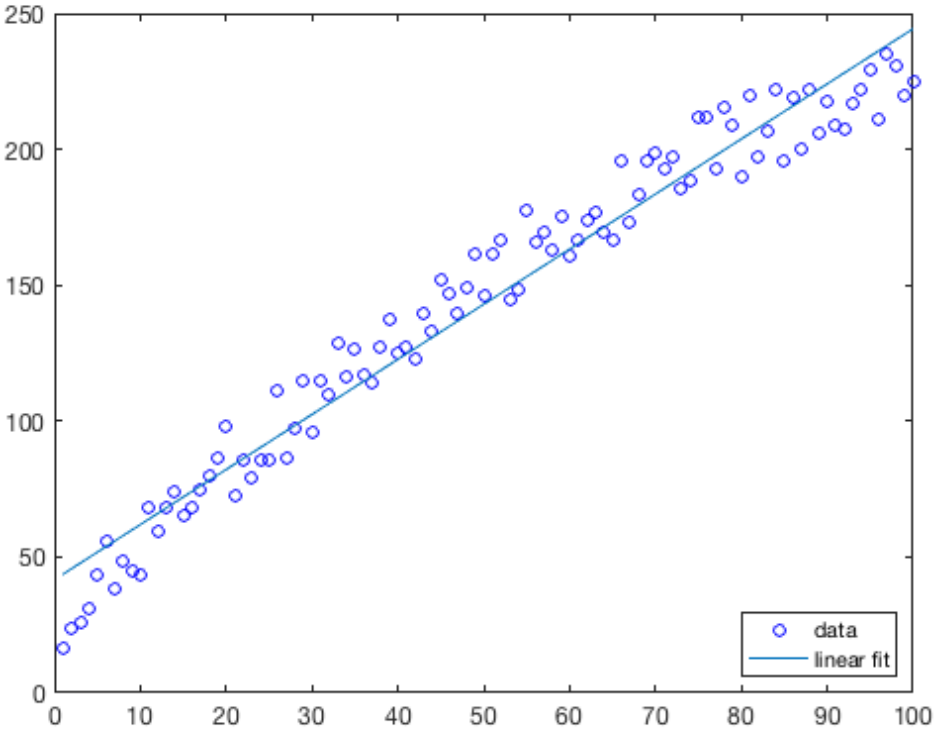
示例

给定一些数据：

```
x = 1:100;

y = 12 - 0.01*x.^2 + 3*x + sind(x) + 30*rand(size(x));
```

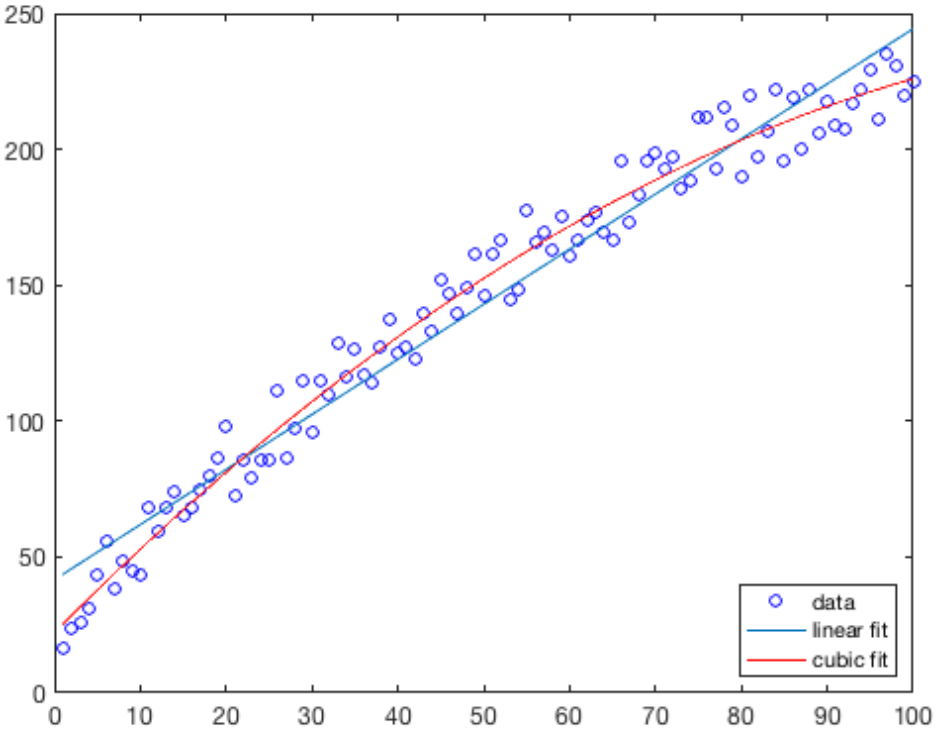
绘制数据并添加简单的线性趋势线：

```
plot(x,y,'bo')
hold on
polyplot(x,y);

legend('data','linear fit','location','southeast')
```



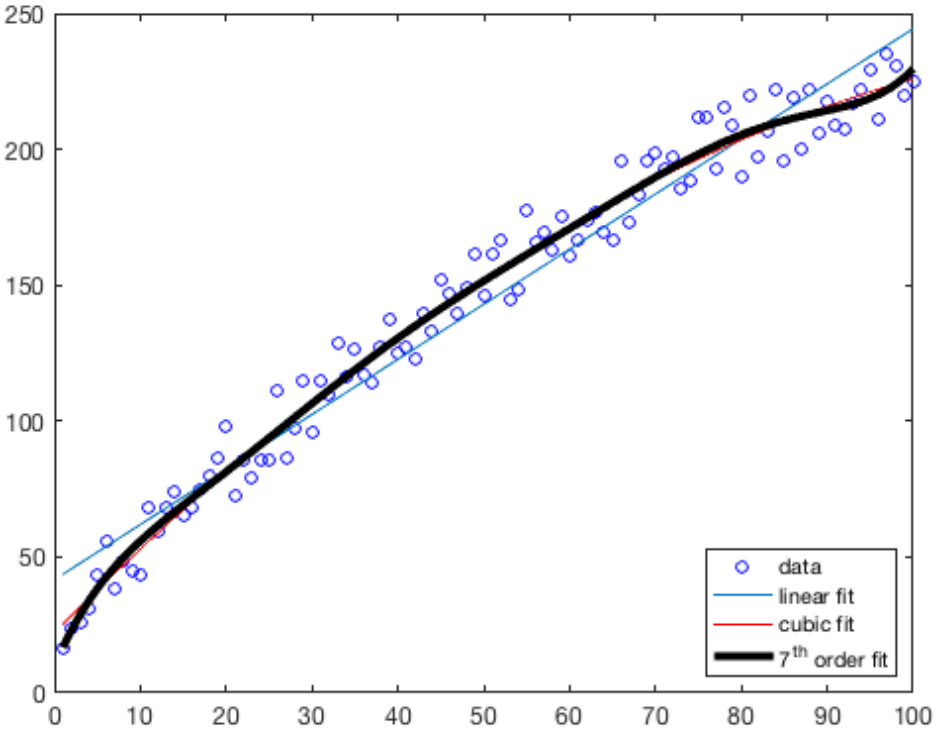
将其设置为立方拟合，而不是线性趋势：

```
polyplot(x,y,3,'r');
legend('data','linear fit','cubic fit','location','southeast')
```



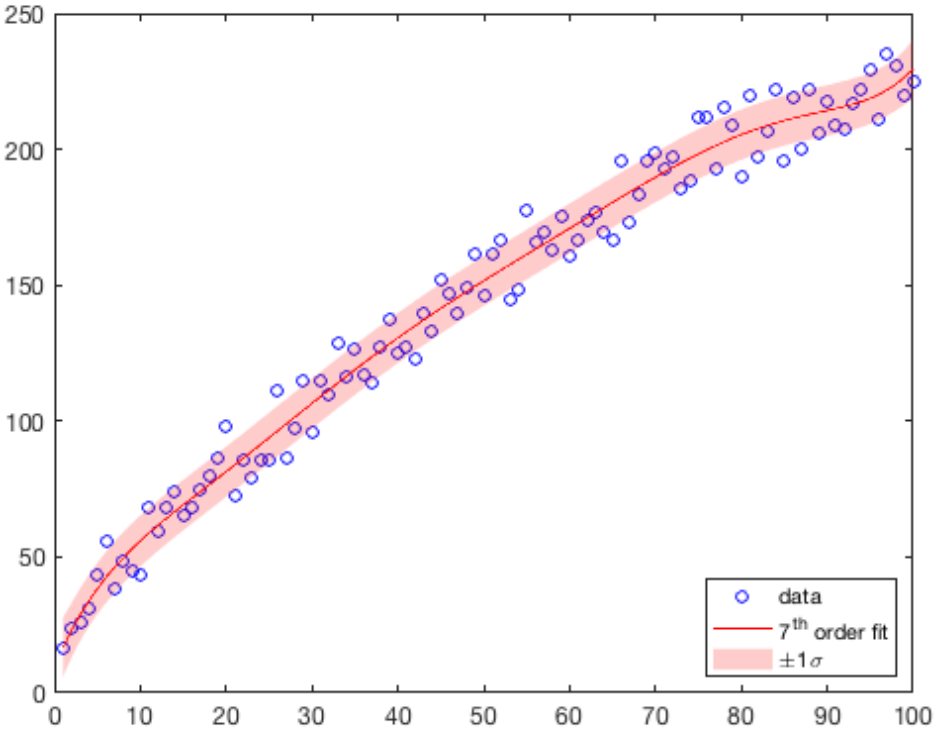
添加一个粗粗的、黑色的 7 阶多项式拟合：

```
polyplot(x,y,7,'k','linewidth',4);
legend('data','linear fit','cubic fit',...
   '7^{th} order fit','location','southeast')
```



同上，但添加标准偏差为+/-1 的误差线：

```
figure

h(1) = plot(x,y,'bo');
hold on
h(2:3) = polyplot(x,y,7,'r','error','alpha');
legend(h,'data','7^{th} order fit','\pm1\sigma','location','southeast')
```



作者简介

polyplot 函数和支持文档由来自德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 于 2015 年 1 月写的。2019 年为 Climate Data Toolbox 改编。

地图

- earthimage 绘制未投影的地球图像底图。
- imagescn 比 pcolor 快，打印您提供给它的*所有*数据（而 pcolor 删除边缘附近的数据和 NaN 值），使 NaN 值透明（而 imagesc 指定的颜色与颜色轴中的最低值相同），并且比 imagesc 稍微容易使用。
- borders 绘制国家或美国州边界，无需 Matlab 的地图工具箱（Mapping Toolbox）。
- bordersm 在使用 Matlab 地图工具箱（Mapping Toolbox）生成的地图上绘制国家或美国州边界。
- labelborders 标注国家或美国各州的边界。
- labelbordersm 在使用 Matlab 地图工具箱（Mapping Toolbox）生成的地图上标记国家或美国州边界。
- stipple 在网格中创建图案或点画填充。
- stipplem 在网格中创建图案或点画填充，用于使用 Matlab 的地图工具箱（Mapping Toolbox）创建的地图。
- quiversc 缩放矢量箭头的密集网格，以便在绘制轴之前舒适地适应轴。
- patchsc 使用按数值缩放的面颜色绘制填充对象。

earthimage 文档

earthimage 绘制未投影的地球图像底图。

语法

```
earthimage
earthimage('gray')
earthimage('watercolor',rgbValues)
earthimage('center',centerLon)
earthimage(...,'bottom')
h = earthimage(...)
```

说明

earthimage 以未投影坐标绘制地球的图像底图。

earthimage('gray') 以灰度绘制图像。

earthimage('watercolor',rgbValues) 使用三元素[R G B]向量（例如，[1 0 0]表示红色）指定海洋的颜色。

earthimage('center',centerLon) 指定中心经度，可以是-180 到 360 之间的任何值。默认的 centerLon 为 0。

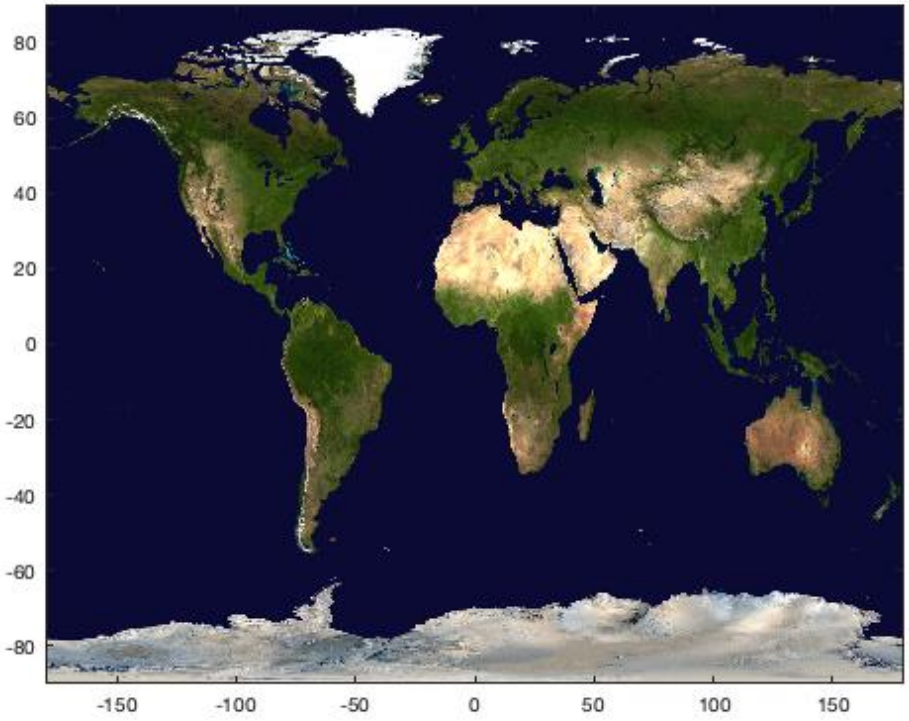
earthimage(...,'bottom') 将地球图像放置在图形堆栈的底部（位于已绘制的其他对象下方）。

h = earthimage(...) 返回打印图像的句柄 h。

示例 1: 简版

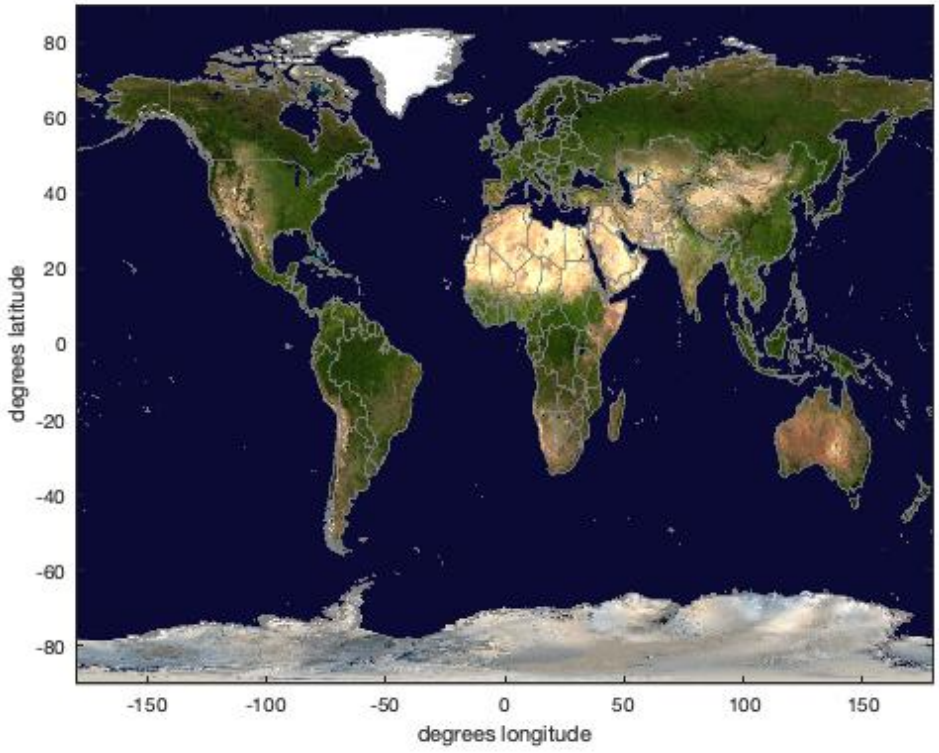
对于简单的底图图像，只需键入 earthimage:

```
earthimage
```



然后，您可以使用正常的 Matlab 绘图功能将其他图层添加到底图中，其中 x 用于经度，y 用于纬度：

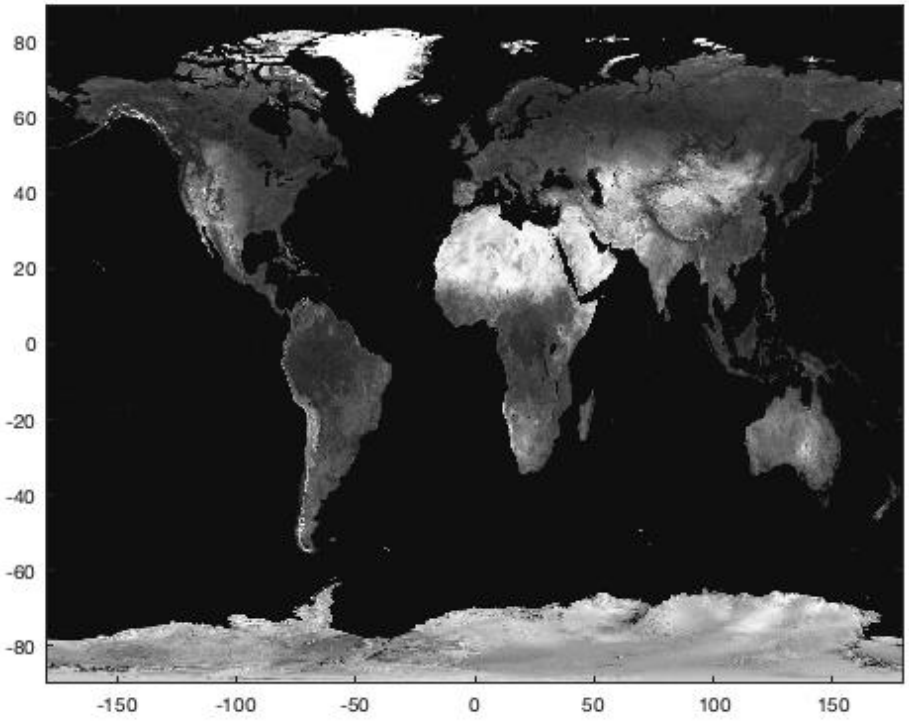
```
hold on
borders('countries','color',0.5*[1 1 1])
xlabel 'degrees longitude'
ylabel 'degrees latitude'
```



示例 2: 灰度图

对于灰度图像，请指定 'gray':

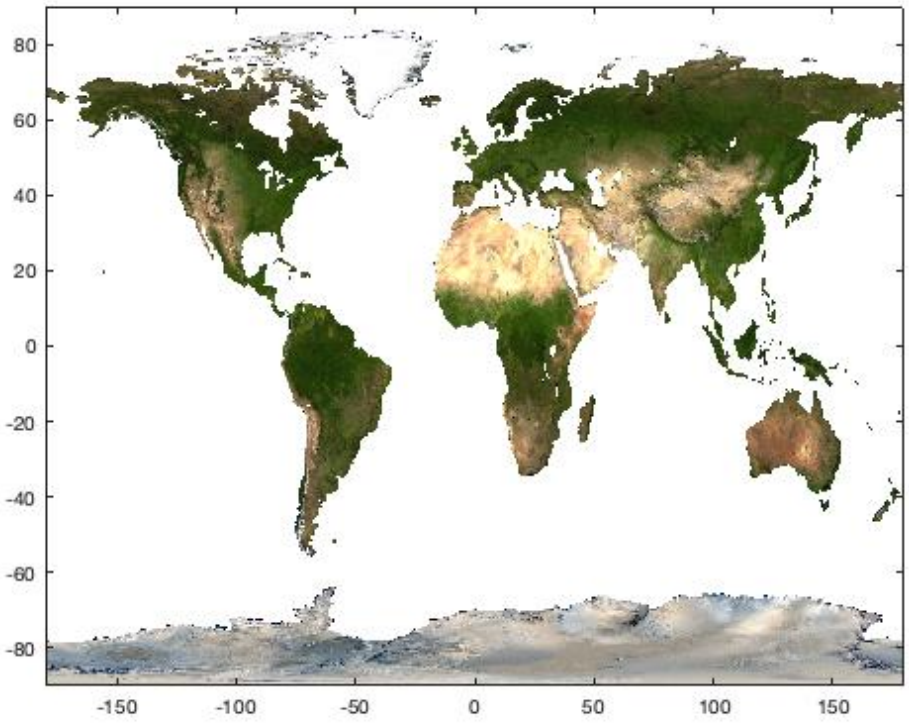
```
figure

earthimage gray
```



示例 3: 透明的海洋

要使海洋透明，请指定'watercolor','none':

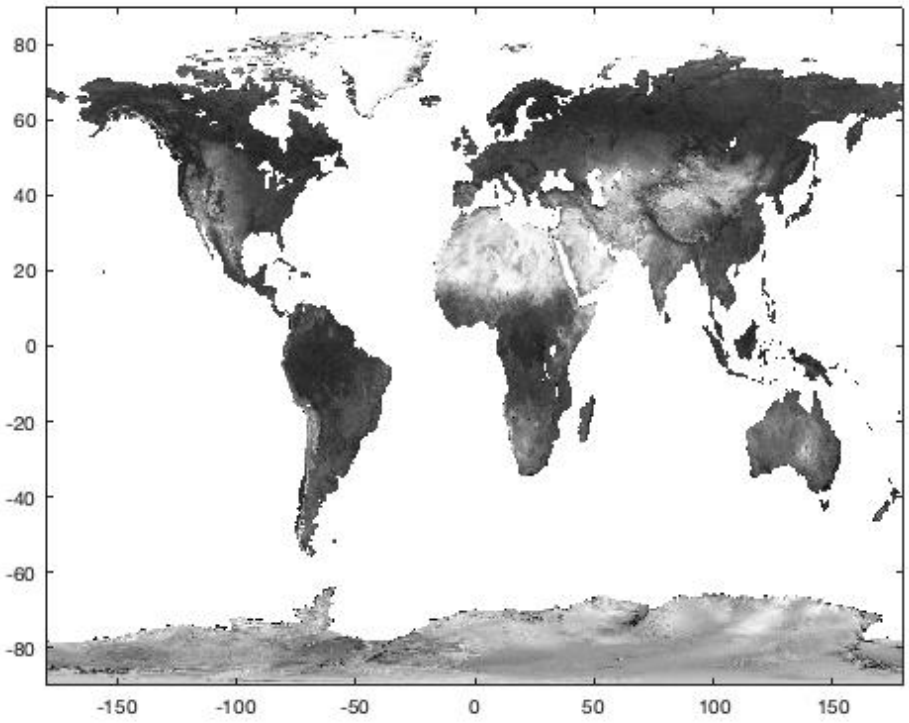
```
earthimage('watercolor','none')
```



示例 4: 灰度陆地和透明海洋

获取灰度陆地和透明海洋，如下所示：

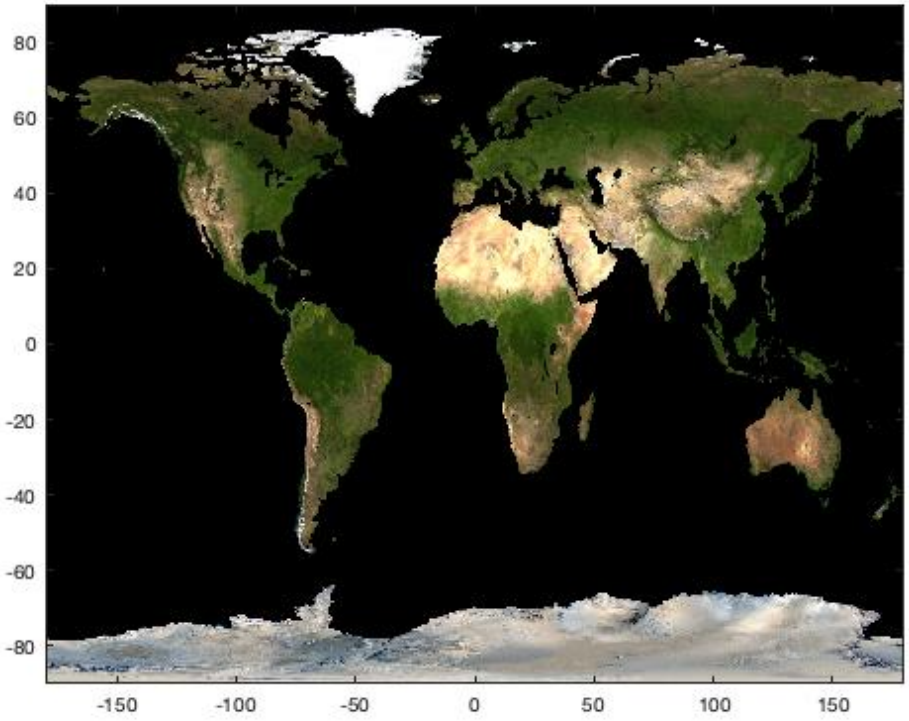
```
earthimage('gray','watercolor','none')
```



示例 4: 色彩斑斓的土地和黑色的海洋

要指定特定的海洋颜色，请输入希望海洋显示的颜色的 RGB 值。例如，plot 通过为'watercolor'指定值 [0 0 0]，使海洋变黑:

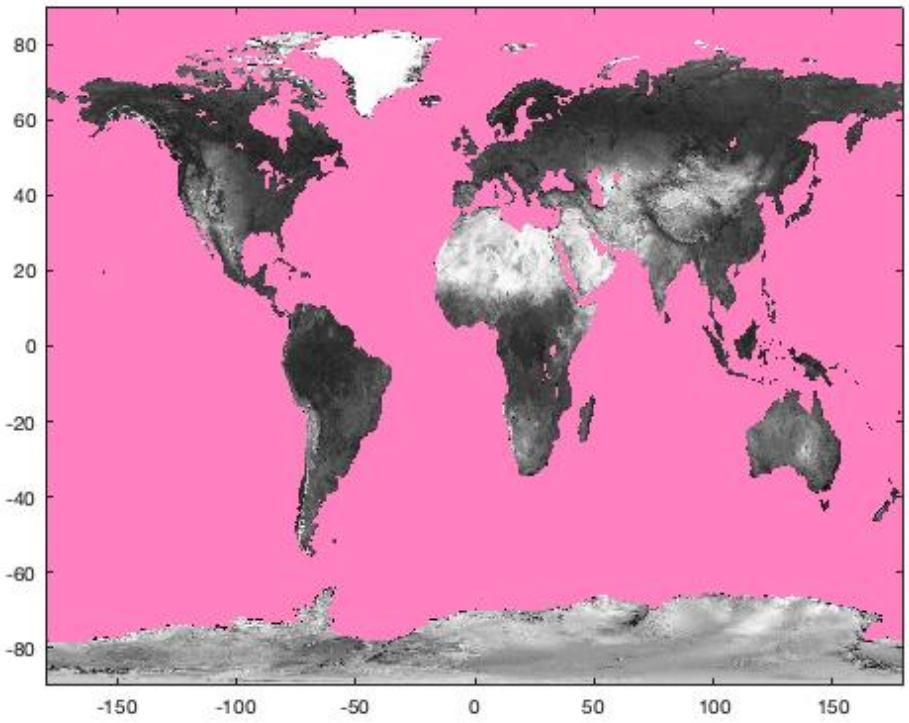
```
earthimage('watercolor',[0 0 0])
```



示例 5: 灰色的陆地和粉红色的海洋

如果您不知道最喜爱颜色的 RGB 值，请改用 rgb 函数：

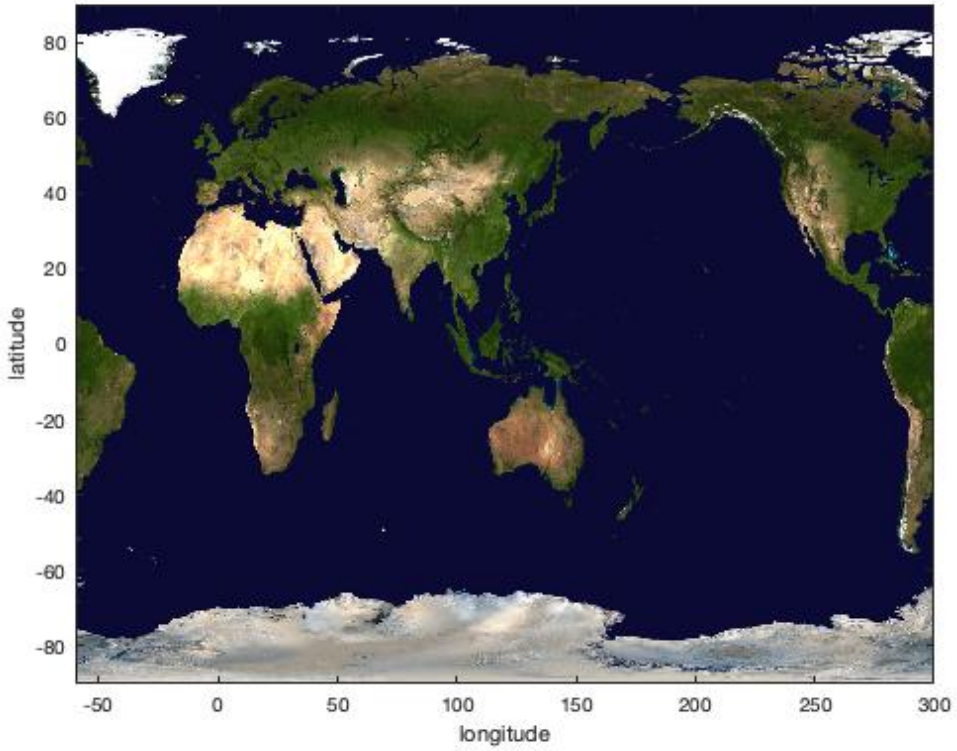
```
earthimage('gray','watercolor',rgb('pink'))
```



示例 6: 以太平洋为中心

默认情况下，中心经度为 0 度，即本初子午线。但是，如果希望地图以不同的经度为中心，只需如下指定中心经度：

```
figure
earthimage('center',120)
xlabel 'longitude'
ylabel 'latitude'
```



作者简介

这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene2018 年为 Climate Data Toolbox for Matlab 写的。

imagescn 文档

imagescn 与 imagesc 类似，但会使 NaN 透明，如果包含 xdata 和 ydata，则将 axis 设置为 xy，并且比 imagesc 具有更多的错误检查。

语法

```
imagescn(C)
imagescn(x,y,C)
imagescn(x,y,C,clims)
imagescn('PropertyName',PropertyValue,...)
h = imagescn(...)
```

说明

imagescn(C) 将数组 C 中的数据显示为使用颜色图中全部颜色范围的图像。C 的每个元素指定图像 1 个像素的颜色。生成的图像是一个 m×n 的像素网格，其中 m 是列数，n 是 C 中的行数。元素的行和列索引确定相应像素的中心。C 中的 NaN 值显示为透明。

imagescn(x,y,C) 指定 C 中像素中心的 x 和 y 位置。如果 x 和 y 是双元素向量，则 C 的外部行和列以 x 和 y 中的值为中心。与 imagesc 类似，如果 x 或 y 是包含两个以上元素的向量，则只考虑向量的第一个和最后一个元素，并自动缩放间距，就像输入两个元素数组一样。imagescn 函数将做得更进一步，允许您输入 x 和 y 作为与 C 大小相同的二维网格。如果包含 x 和 y，imagescn 函数会自动将轴设置为笛卡尔 xy，而不是（反向）ij 轴。

imagescn(x,y,C,clims) 指定映射到颜色图的第一个和最后一个元素的数据值。将 clims 指定为 [cmin cmax]形式的两元素向量，其中小于或等于 cmin 的值映射到颜色图中的第一种颜色，大于或等于 cmax 的值映射到颜色图中的最后一种颜色。

imagescn('PropertyName',PropertyValue,...) 将图像属性指定为名称-值对。

h = imagescn(...) 返回所创建对象的句柄。

真实世界示例

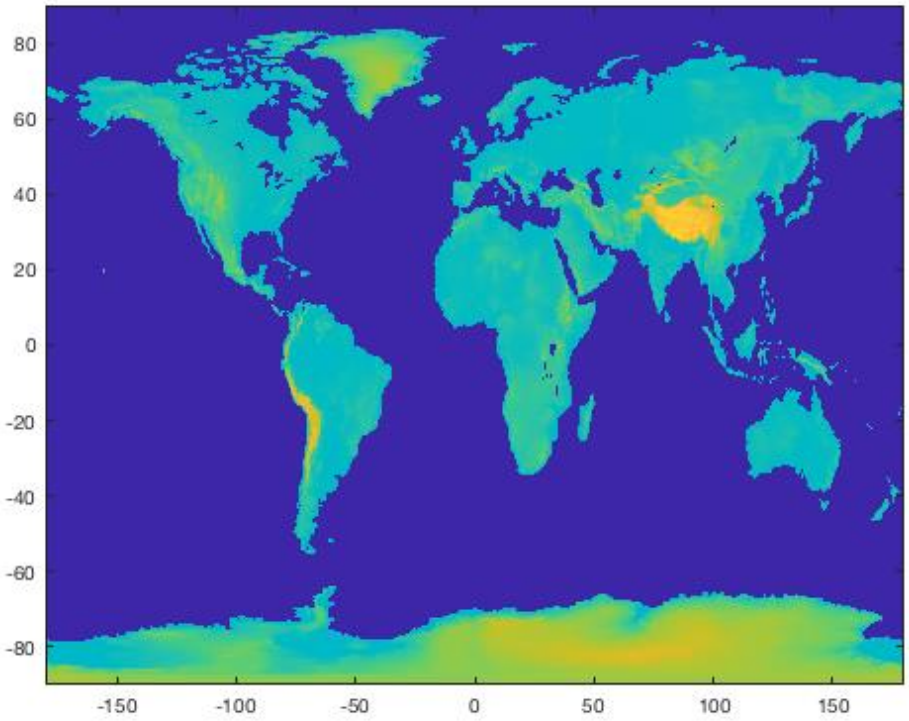
假设您有一个全球地形栅格，并且希望将所有海洋值设置为 NaN 并使其透明。让我们制作一个 1/8 度分辨率的采样栅格，并使用 topo_interp 获取地形，以及使用 island 生成陆地掩膜：

```
[lat,lon] = cdtgrid(1/8);

z = topo_interp(lat,lon);

land = island(lat,lon);



% 将海洋单元转化为 NaN:
z(~land) = NaN;
```

使用标准 imagesc 函数绘制数据的方式如下所示：

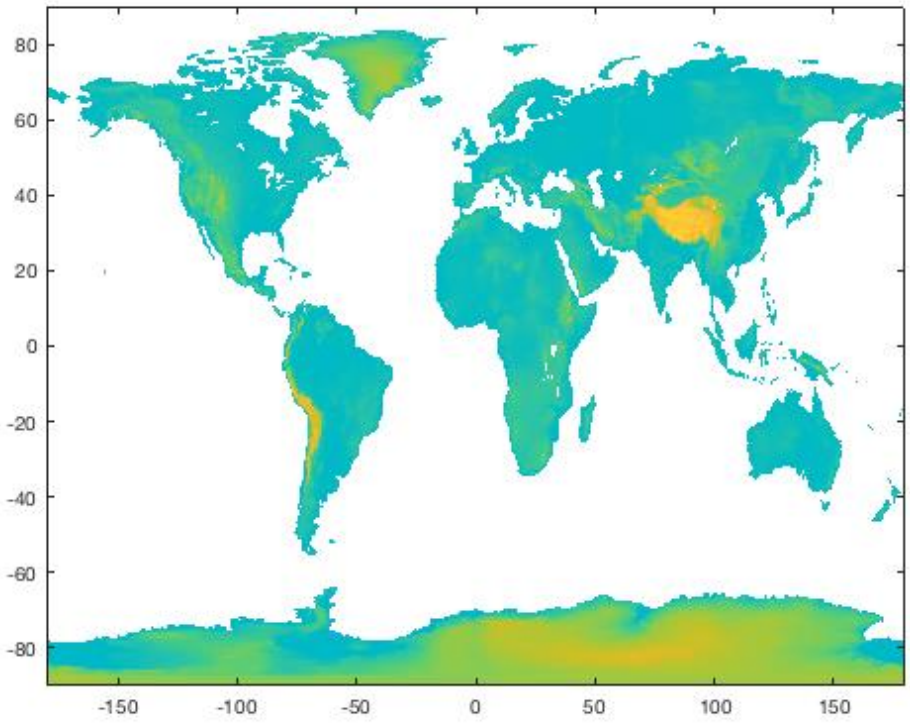
```
figure
```

```
imagesc(lon(1,:),lat(:,1),z)

axis xy
```



以上，我们必须指定经纬度网格的行和列，最后我们甚至没有得到海洋设置为 NaN 的清晰像素。使用 imagescn 更容易，它允许您输入完整的经纬度网格，自动将轴翻转为正常 xy 格式，并使 NaN 单元透明：

```
figure

imagescn(lon,lat,z)
```



imagesc, imagescn, 和 pcolor 的区别

imagescn 函数使用 imagesc 绘制数据，但在绘制后，将 NaN 像素设置为 0 的透明值。imagescn 函数允许输入坐标 x 和 y 为网格,假设其间隔均匀且单调,就像由 meshgrid 创建一样。如果调用 imagescn 时包含 x 和 y 数据，则 y 轴方向设置为正常，而不是 imagesc 的默认行为，将 y 轴方向设置为反转。imagescn 函数比 pcolor 快，这可能有利于大型数据集，pcolor 函数（无意义地）删除外部数据行和数据列（如下所示），pcolor 也拒绝绘制距离任何 NaN 最近的数据点。imagescn 函数不会删除任何数据。但是，如果 x、y 坐标的间隔不均匀，或者需要插值着色，则有时仍可能希望使用 pcolor。

示例和与其他函数的比较

此示例比较 imagescn, imagesc, 和 pcolor。从一个 5x5 样本数据集开始，将一个 NaN 放在矩阵的中心。

```
% 一个 5x5 样本数据集:
[X,Y,Z] = peaks(5);

% 中间有一个 NaN 值:
Z(3,3) = NaN;
```

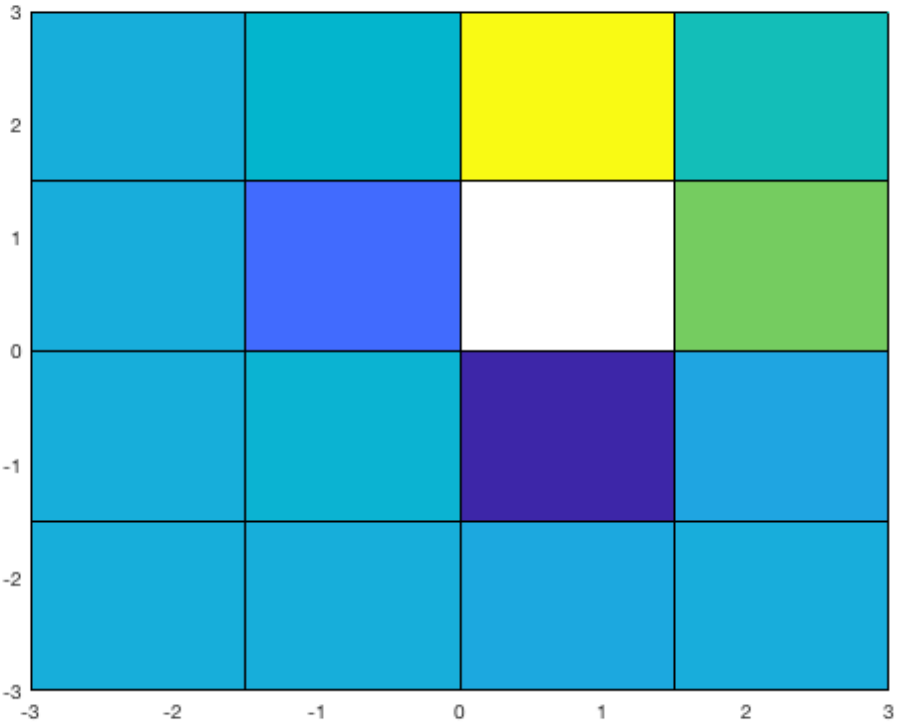
要清楚的是，我们放在 Z 中间的 NaN 值应该在 x，y 网格上的位置(0,0)显示。看，第 3 行第 3 列的位置为(0,0)：

```
[X(3,3) Y(3,3)]
```

```
ans =

     0     0
```

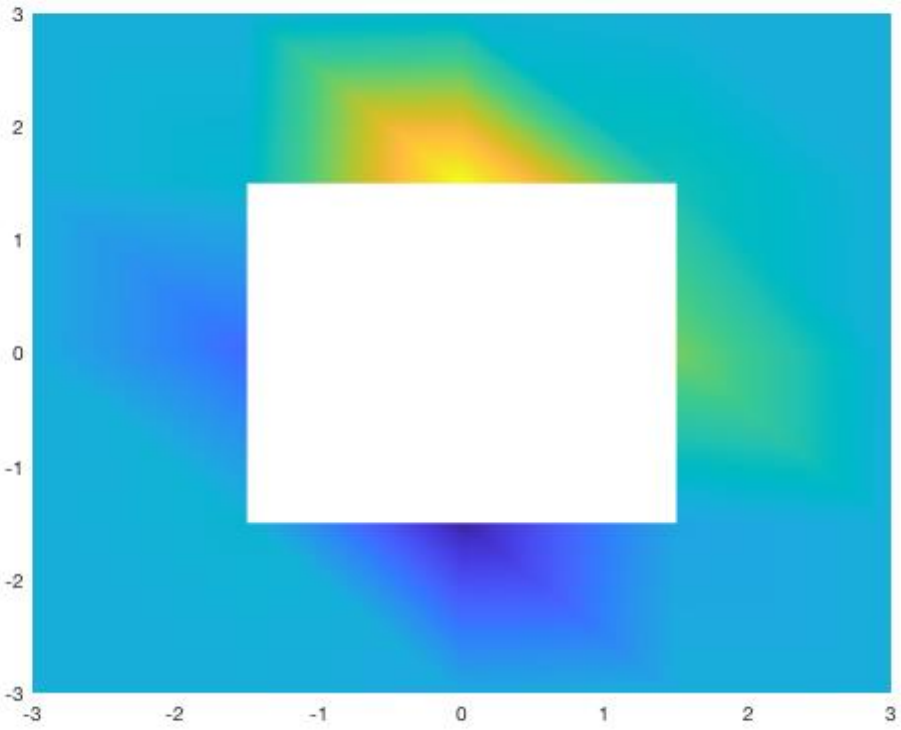
使用 pcolor 绘制这样的网格很容易:

```
figure
pcolor(X,Y,Z)
```



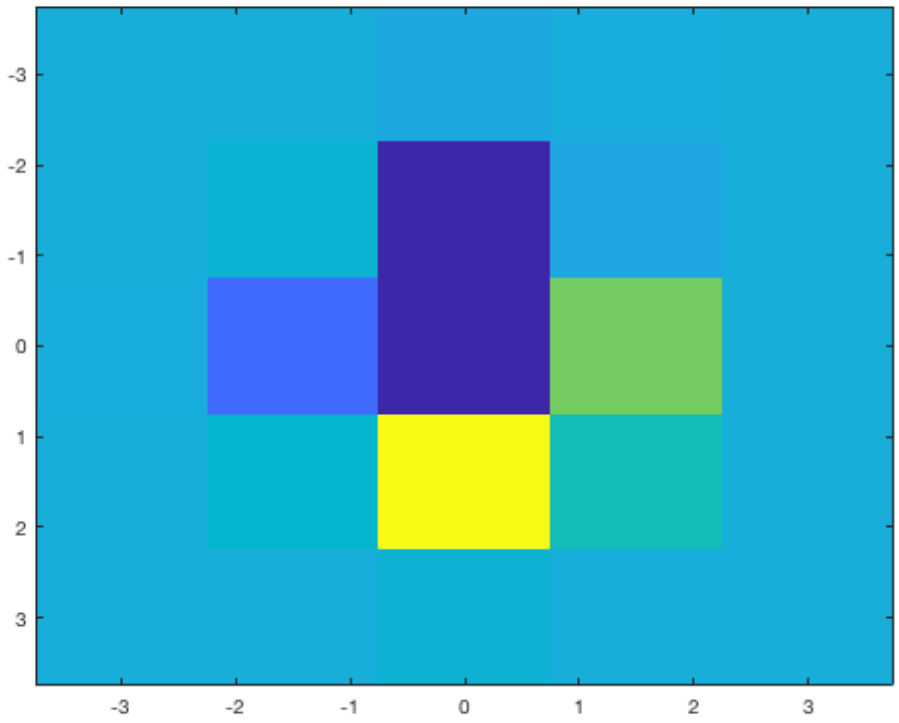
但是嘿，等等！Z 不是 5x5 矩阵吗？我只看到 4x4！中间的 NaN 孔不应该以中心 X 和 Y 值为中心，是 NaN 吗？通过插值着色可以修复偏心问题，但如您所见，插值着色会导致更多数据丢失：

```
shading interp
```



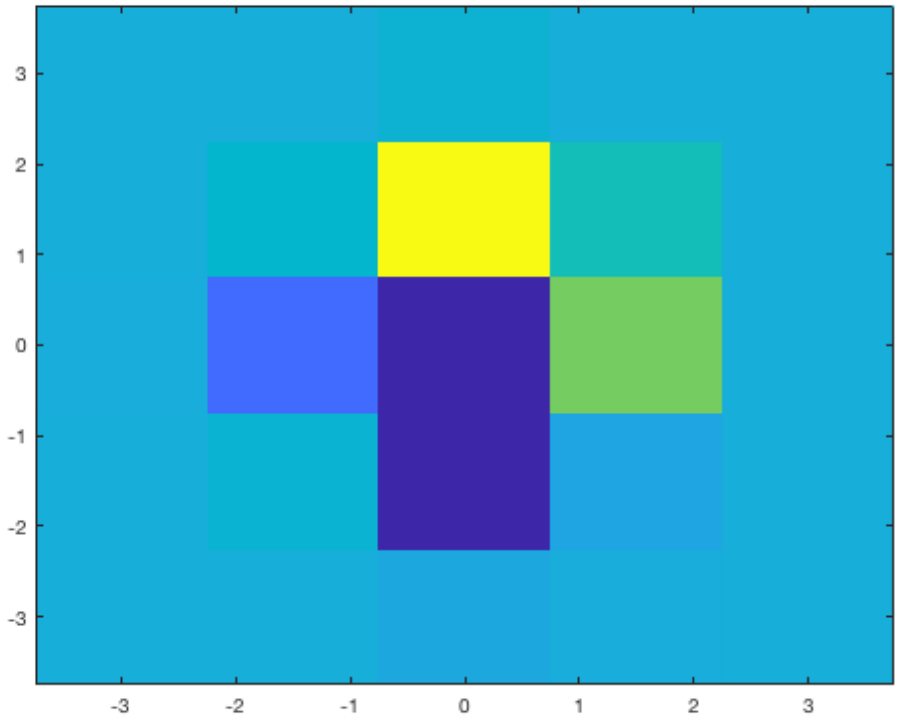
我们可以使用内置函数 imagesc 代替 pcolor，但它不允许我们输入 X 和 Y 网格，因为它们是 5x5。要使用 imagesc，必须作为一维矢量输入 x 和 y 坐标：

```
imagesc(X(1,:),Y(:,1),Z)
```



首先，您可能会注意到 y 轴已切换方向，因此您需要使用 imagesc 配合：

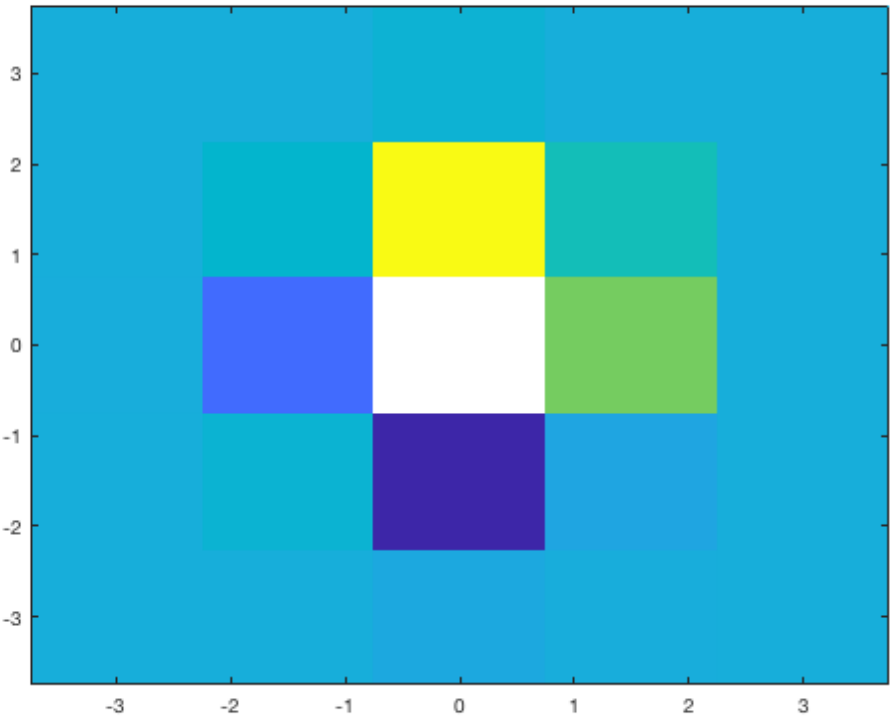
```
axis xy
```



现在我们有了一个适当的 5x5 矩阵，每个像素正确地集中在其适当的 x，y 位置上，但是中心像素，NaN 值与图像中的最低值像素不明确，因为默认情况下，imagesc 使 NaN 值看起来像具有真实值的真实数据，就像数据集中的其他数据一样。那很危险。

imagescn 允许您输入 x 和 y 值作为矢量或二维网格，正确对齐每个数据点（假设网格的间距相等），如果给定 x 和 y 值，则以 xy 坐标打印，并使 NaN 值透明。

```
imagescn(X,Y,Z)
```



作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene2017 年 2 月写的。

borders 文档

borders 绘制国家或美国州边界，无需 Matlab 的地图工具箱（Mapping Toolbox）。如果要在由 Matlab 的地图工具箱（Mapping Toolbox）生成的地图上绘制边界，请改用 bordersm 。
数据来自 2013 US Census Bureau 500k data（2013 年美国人口普查局 50 万数据）和 thematicmapping.org TM World Borders 0.3 数据集。

语法

```
borders
borders(place)
borders(...,LineProperty,LineValue)
borders(...,PatchProperty,PatchValue)
borders(...,'center',centerLon)
h = borders(...)
[lat,lon] = borders(place)
```

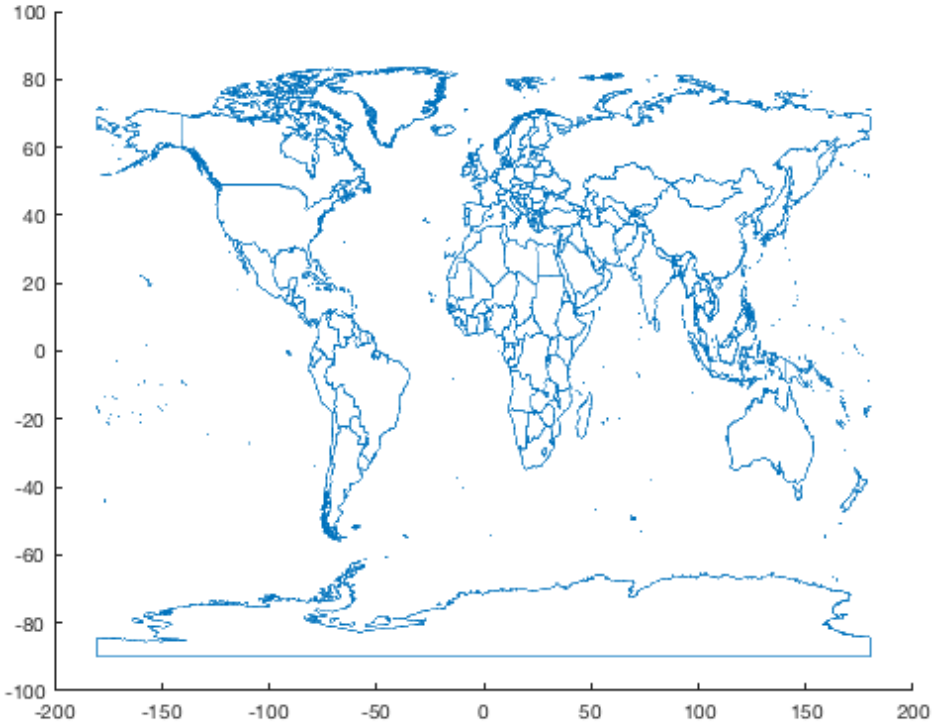
说明

borders 绘制国界线.
borders(place) 绘制一个地方（place）的边界，可以是任何国家或美国州。地点也可能是“国家”（'countries'）以绘制所有国界线，“州”（'states'）绘制所有美国州边界，或“美国大陆”（'Continental US'）仅绘制美国大陆（对不起，关岛）。注：要绘制格鲁吉亚国家，请使用'Georgia'。要绘制美国佐治亚州，请指定'Georgia.'，并加上句号。t
borders(...,LineProperty,LineValue) 指定线型或标记样式。
borders(...,PatchProperty,PatchValue) 在任何属性以'face'（例如，'facecolor','red'）开头时，将州或国家概括为填充对象。请注意，将所有国家绘制为填充可能有点慢。
borders(...,'center',centerLon) 指定中心经度。默认值为 0。
h = borders(...) 返回打印对象的句柄 h。
[lat,lon] = borders(place) 不绘制任何边界，但返回其地理坐标的数组。

示例 1: 非常简单

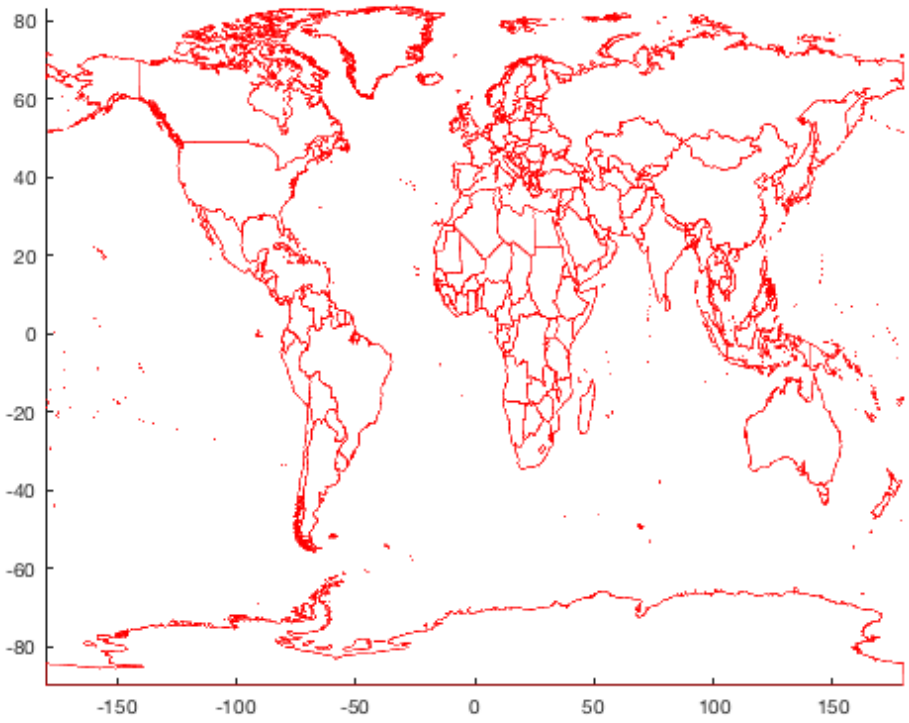
绘制所有的国界线，只需要键入 borders: （译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
borders
```



或按如下方式指定线属性：（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

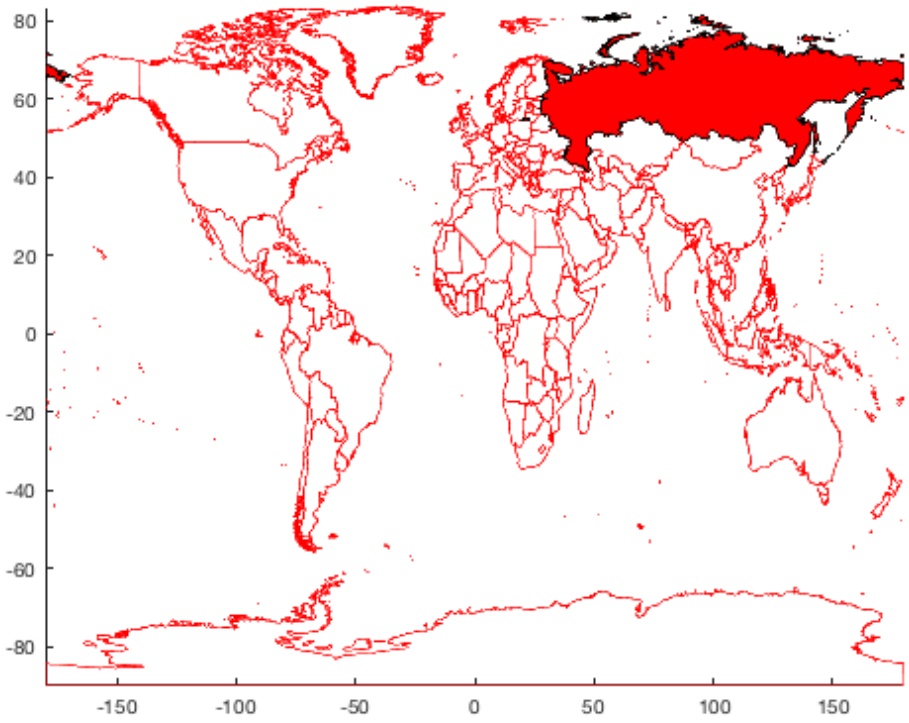
```
borders('countries','r')
axis tight
```



示例 2: 红色俄罗斯

将俄罗斯添加到地图中，作为一个红色大的填充：（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

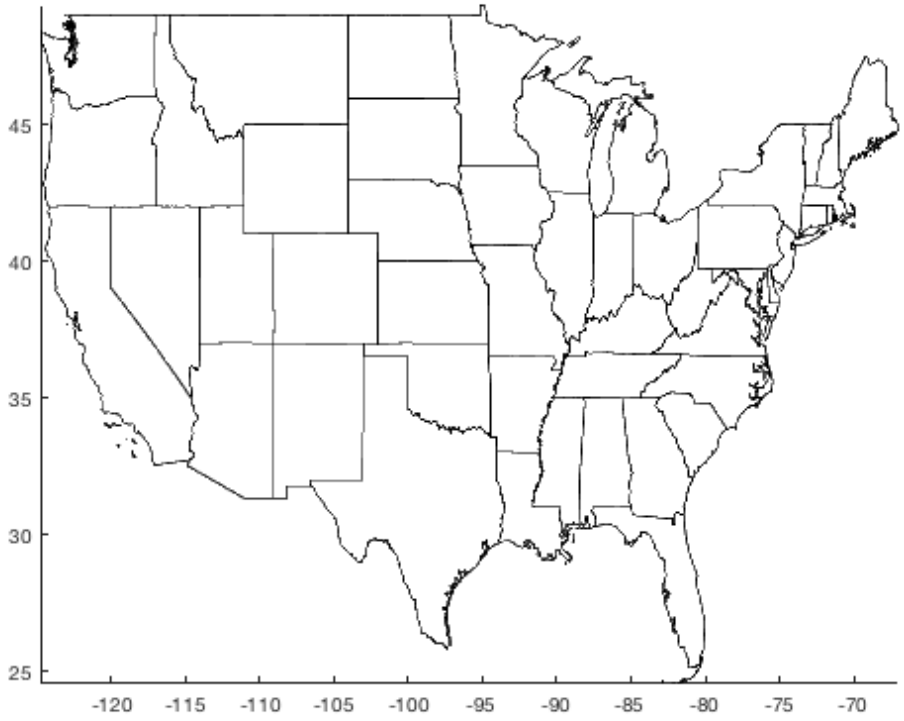
```
borders('russia','facecolor','red')
```



示例 3: 美国大陆

打开一个新图形，用黑色外线绘制美国大陆：

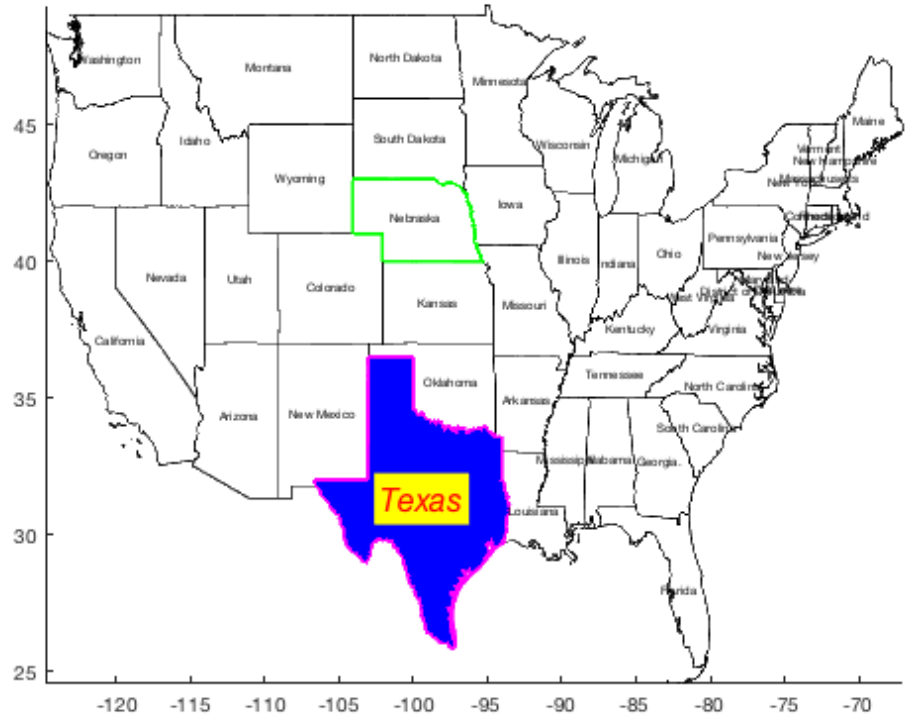
```
figure

borders('continental us','k')
axis tight
```



示例 4: 花哨的格式

让我们把德克萨斯变成蓝色（呵呵哒），给它一个浓密的洋红色轮廓，给内布拉斯加州一个浓密的绿色轮廓。labelborders 函数的工作原理与 borders 函数类似。

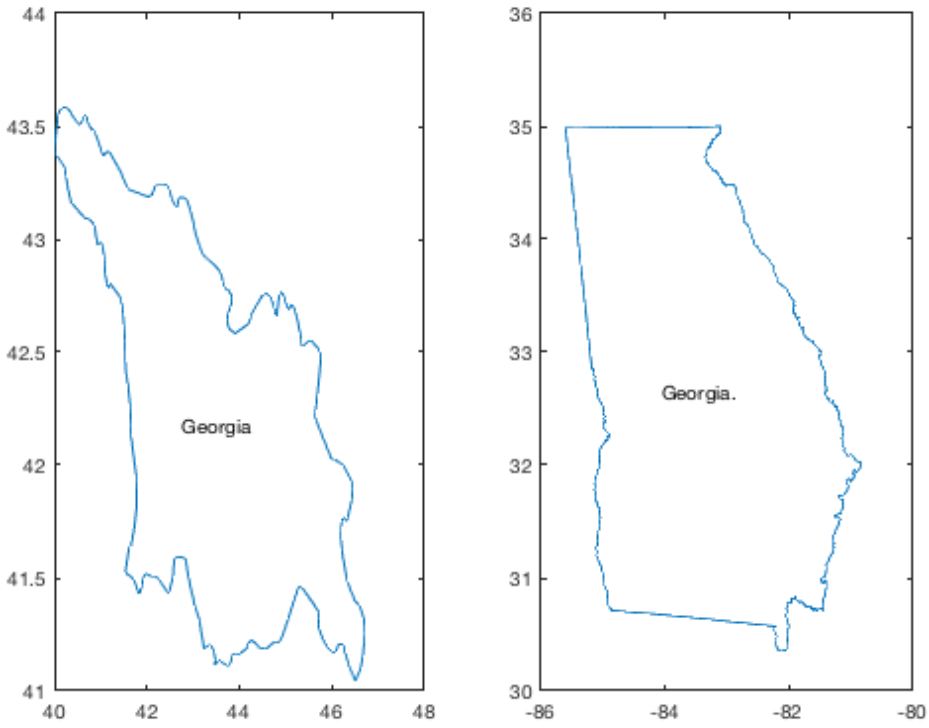
```
borders('texas','facecolor','blue','edgecolor','m','linewidth',2)
borders('nebraska','g','linewidth',2)
labelborders('continental us','fontsize',6);
labelborders('Texas','color','r','backgroundcolor','y',...
    'fontangle','italic','fontsize',16)
```



示例 5: Georgia 和 Georgia.

有两个 Georgia。为了区分它们，我在美国乔治亚州的末尾加了一个句号。让我们比较一下：

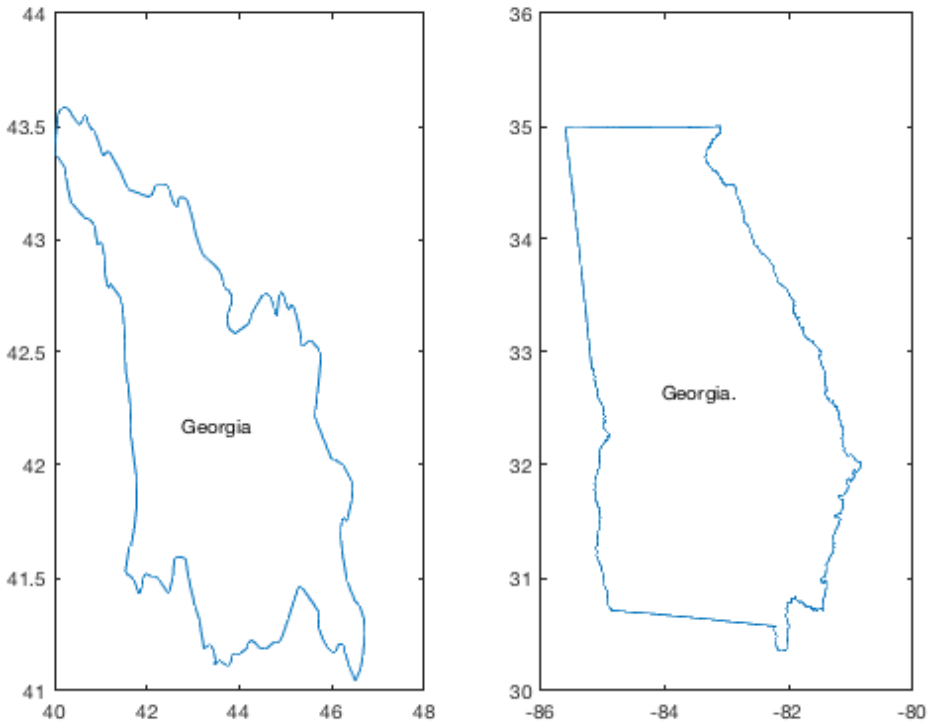
```
figure

subplot(121)

borders 'georgia'
labelborders 'Georgia'
subplot(122)
borders 'georgia.'
labelborders 'Georgia.'
```



示例 6: 只有数据

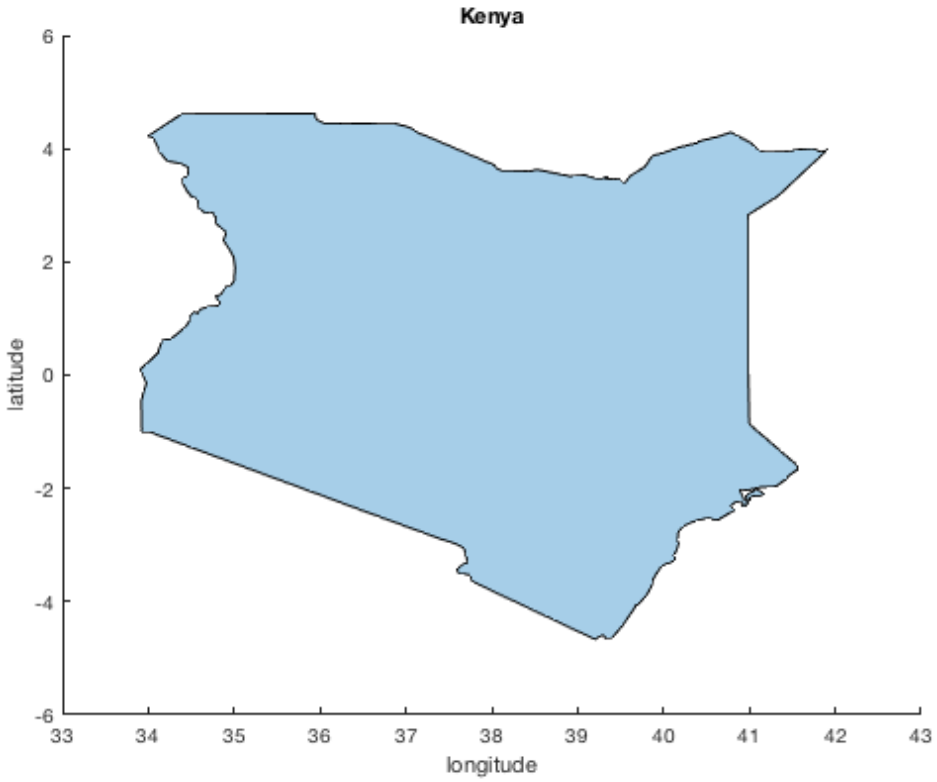
只想要一个国家或州的轮廓而不去绘制它？使用 borders 具有两个输出可以返回 lat、lon 阵列，而无需绘制。

```
[lat,lon] = borders('kenya');
```



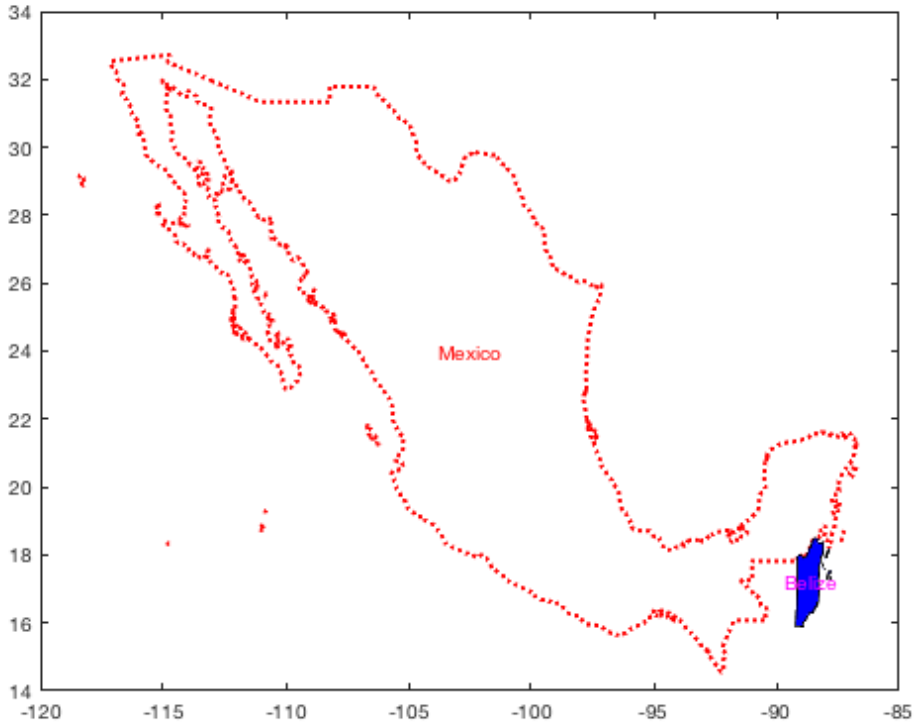
有了地理坐标数据，您可以随心所欲地绘制它。新的 polyshape 函数是实现此功能的一种方法：

```
pgon = polyshape(lon,lat,'simplify',false);

figure
plot(pgon)
xlabel longitude
ylabel latitude
title Kenya
```



示例 8: 更多格式

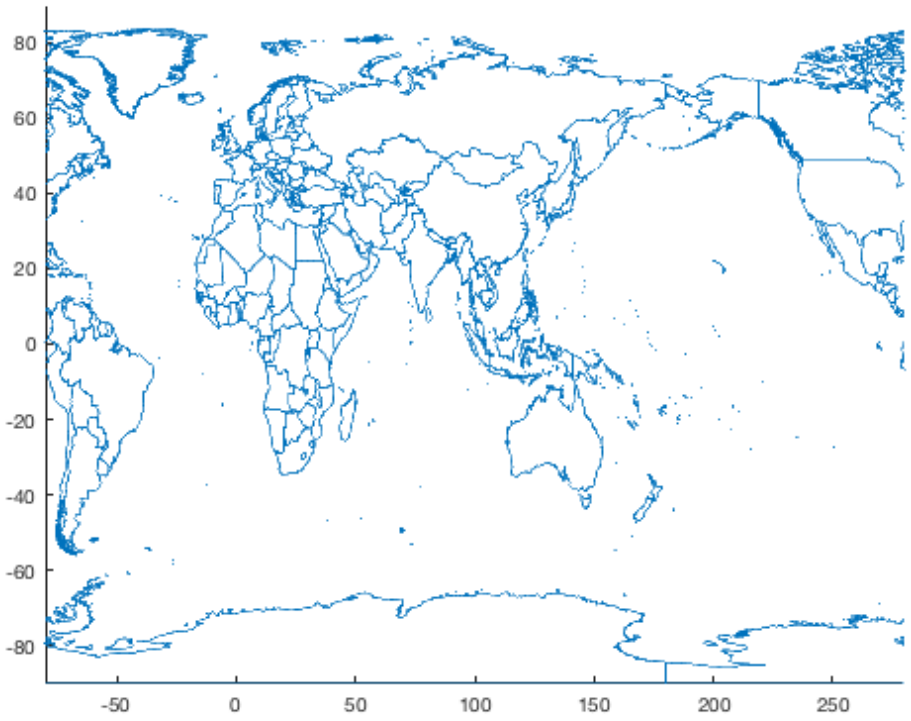
```
figure

borders('mexico','r:','linewidth',2)
hold on
borders('belize','facecolor','b','linestyle','-','linewidth',1)
labelborders('Mexico','color','r')
labelborders('Belize','color','m')
```



示例 9: 指定中间经度

不想在地图中间的本初子午线吗？指定您自己喜欢的中心经度，如下所示：（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
figure

borders('countries','center',100)
```



作者简介

borders 和 labelborders 函数是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene2015 年 4 月写的。这个函数是 Climate Data Toolbox for Matlab 的一部分。

bordersm 文档

bordersm 在由 Matlab 的地图工具箱（Mapping Toolbox）生成的地图上绘制边界绘制国家或美国州边界。如果没有 Matlab 地图工具箱（Mapping Toolbox），请改用 borders。

数据来自 2013 US Census Bureau 500k data（2013 年美国人口普查局 50 万数据）和 thematicmapping.org TM World Borders 0.3 数据集。

语法

```
bordersm
bordersm(place)
bordersm(...,LineProperty,LineValue)
bordersm(...,PatchProperty,PatchValue)
h = bordersm(...)
[lat,lon] = bordersm(place)
```

说明

bordersm 中地图上绘制国界线。

bordersm(place) 绘制一个地方（place）的边界，可以是任何国家或美国州。地点也可能是“国家”（'countries'）以绘制所有国界线，“州”（'states'）绘制所有美国州边界，或“美国大陆”（'Continental US'）仅绘制美国大陆（对不起，关岛）。注：要绘制格鲁吉亚国家，请使用'Georgia'。要绘制美国佐治亚州，请指定'Georgia.'，并加上句号。

bordersm(...,LineProperty,LineValue) 指定线型或标记样式。

bordersm(...,PatchProperty,PatchValue) 在任何属性以'face'（例如，'facecolor','red'）开头时，将州或国家概括为填充对象。请注意，将所有国家绘制为填充可能有点慢。

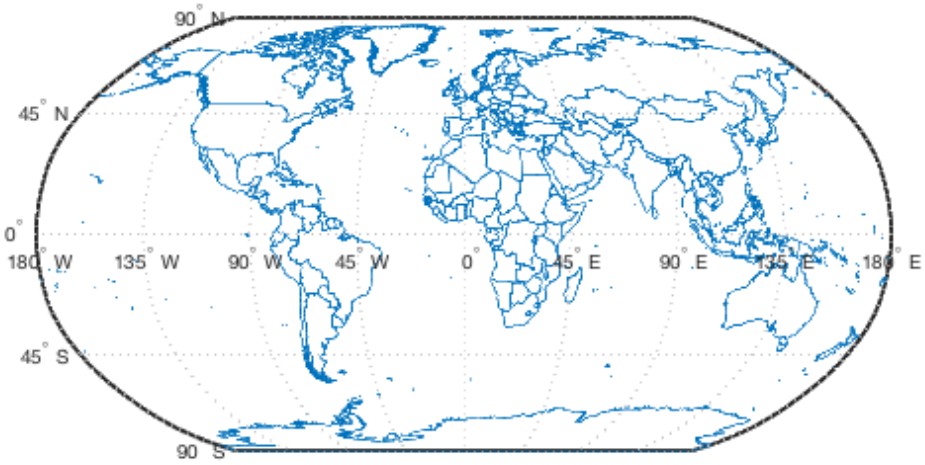
h = bordersm(...) 返回绘制对象的句柄 h。

[lat,lon] = bordersm(place) 不绘制任何边界，但返回其地理坐标的数组。borders 绘制国界线。

示例 1: 非常简单

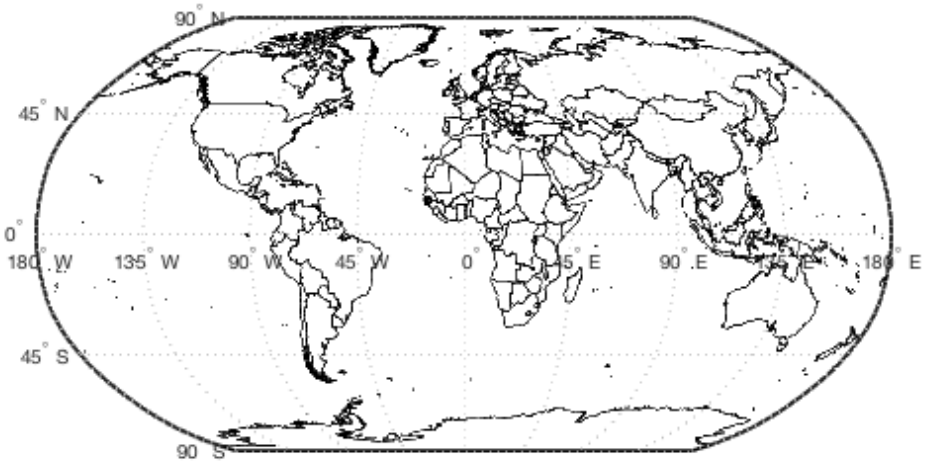
绘制所有的国界线，只需要键入 bordersm:（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
bordersm
```



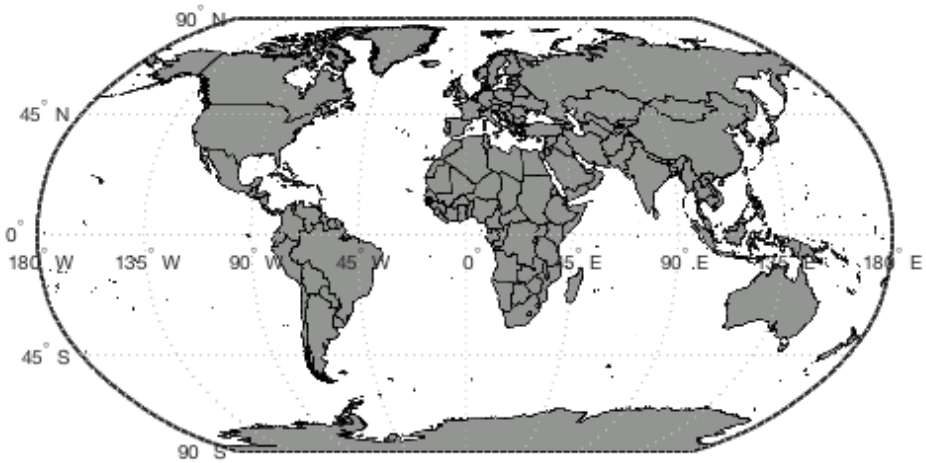
或者像这样指定线特性（这会使国家轮廓变为黑色）：（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
bordersm('countries','k')
```



指定面颜色或其他填充特性将从线绘制切换到填充对象绘制。（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

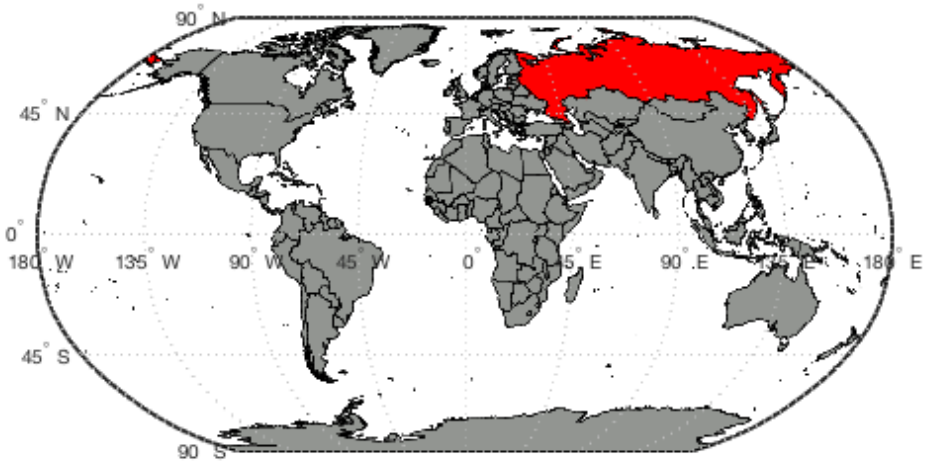
```
bordersm('countries','facecolor',rgb('gray'))
```



示例 2: 红色俄罗斯

要仅绘制一个国家，请按如下方式指定其名称。将俄罗斯添加到地图中，作为一个大的红色填充：（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

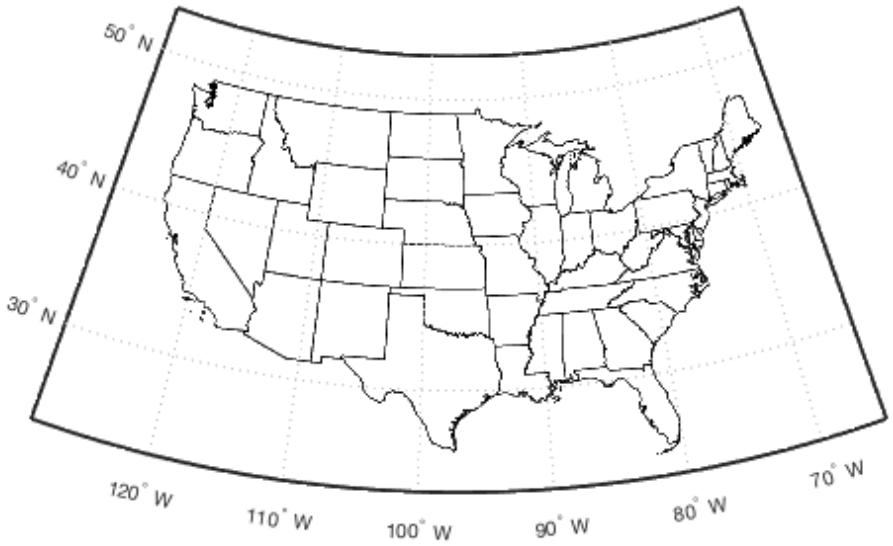
```
bordersm('russia','facecolor','red')
```



示例 3: 美国大陆

打开一个新图形，用黑色外线绘制美国大陆：

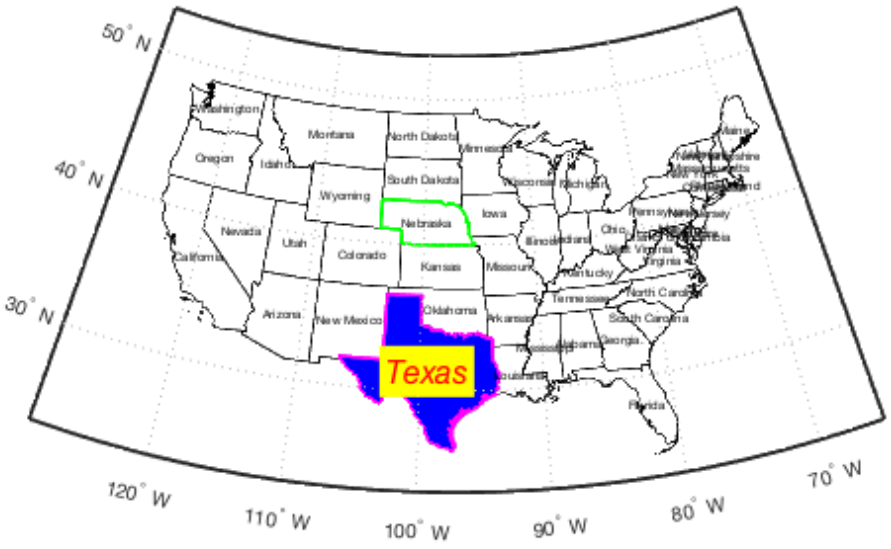
```
figure

bordersm('continental us','k')
```



示例 4: 花哨的格式

让我们把德克萨斯变成蓝色（贝托做不到，但我们肯定能做到），给它一个浓密的洋红色轮廓，给内布拉斯加州一个浓密的绿色轮廓。labelbordersm 函数的工作原理与 bordersm 函数类似。

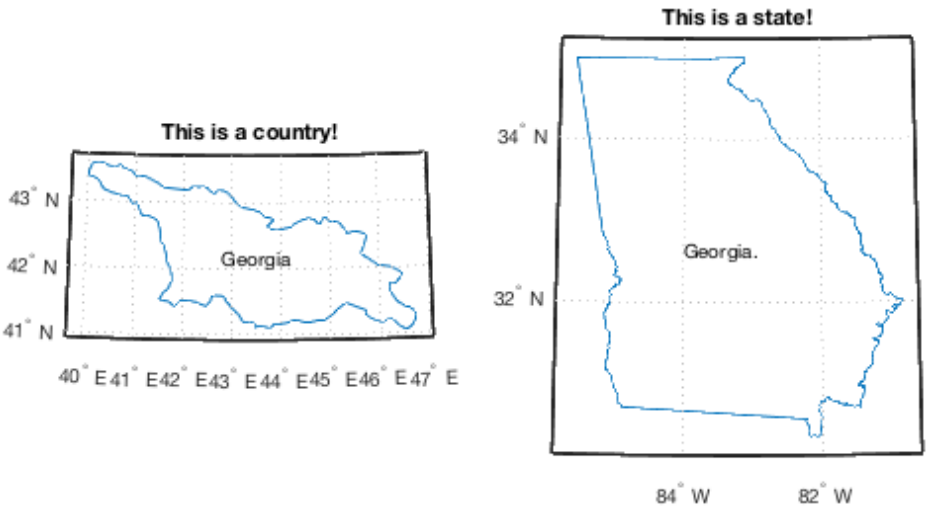
```
bordersm('texas','facecolor','blue','edgecolor','m','linewidth',2)
bordersm('nebraska','g','linewidth',2)
labelbordersm('continental us','fontsize',6);
labelbordersm('Texas','color','r','backgroundcolor','y',...
    'fontangle','italic','fontsize',16)
```



示例 5: Georgia 和 Georgia.

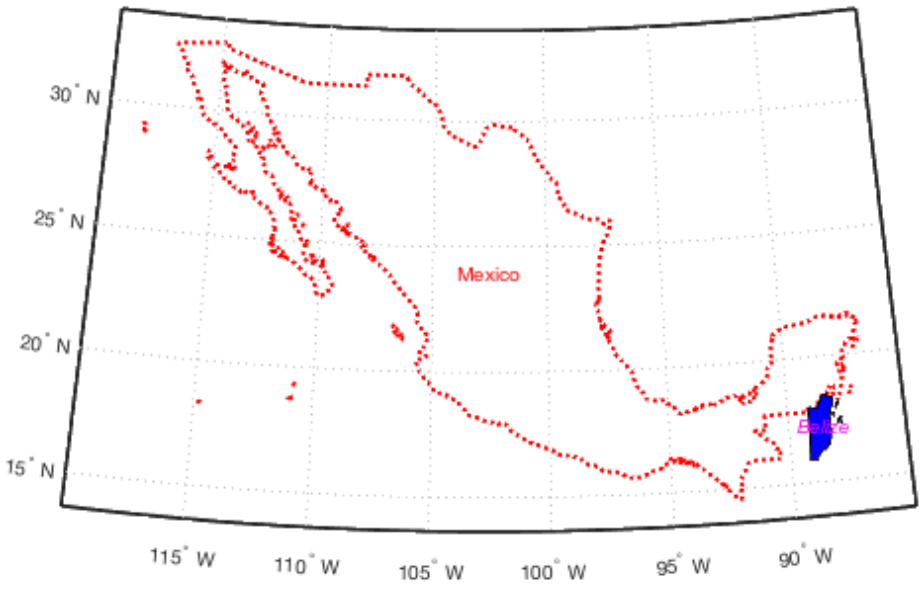
有两个 Georgia。为了区分它们，我在美国乔治亚州的末尾加了一个句号。让我们比较一下：

```
figure

subplot(121)

bordersm 'georgia'
labelbordersm 'Georgia'
title 'This is a country!'

subplot(122)
bordersm 'georgia.'
labelbordersm 'Georgia.'
title 'This is a state!'
```



示例 6: 更多格式

```
figure

bordersm('mexico','r:','linewidth',2)
hold on
bordersm('belize','facecolor','b','linestyle','-','linewidth',1)
labelbordersm('Mexico','color','r')
labelbordersm('Belize','color','m','fontangle','italic')
```



作者简介

bordersm 和 labelbordersm 函数是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene2015 年 4 月写的。这个函数是 Climate Data Toolbox for Matlab 的一部分。

labelborders 文档

labelborders 函数在未投影的地图上标记国家或美国各州的边界，并且不需要 Matlab 地图工具箱（Mapping Toolbox）。

语法

```
labelborders(place)
labelborders(...,TextProperty,TextValue)
h = labelborders(...)
```

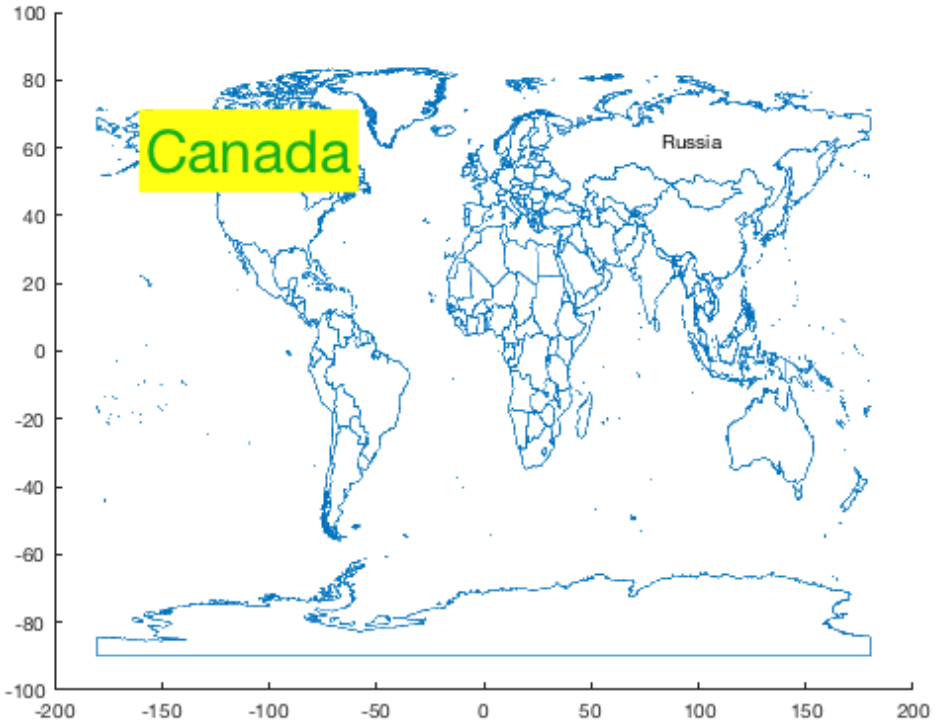
说明

labelborders(place) 绘制一个地方（place）的边界，可以是任何国家或美国州。地点也可能是“国家”（'countries'）以绘制所有国界线,“州”（'states'）绘制所有美国州边界，或“美国大陆”（'Continental US'）仅绘制美国大陆（对不起，关岛）。注：要绘制格鲁吉亚国家，请使用'Georgia'。要绘制美国佐治亚州，请指定'Georgia.'，并加上句号。
labelborders(...,TextProperty,TextValue) 指定文本格式。
h = labelborders(...) 返回绘制文本对象的句柄 h 。

示例 1

以下是使用 borders 函数绘制的一些边界。用简单的黑色文本标记俄罗斯，用黄色背景的大绿色文本标记加拿大：（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
borders

labelborders 'Russia'

labelborders('Canada','fontsize',30,...
   'color',rgb('green'),'backgroundcolor',rgb('yellow'))
```



示例 2:

绘制 earthimage ，放大欧洲，并在地图上放置灰色斜体文本标签：

```
figure

earthimage

axis([-10 30 35 60]) % 放大欧洲
labelborders('countries','color',rgb('gray'),'fontangle','italic')
```

作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

labelbordersm 文档

labelbordersm 函数在使用 Matlab 地图工具箱（Mapping Toolbox）生成的地图上标记国家或美国州边界。对不需要 Matlab 地图工具箱（Mapping Toolbox）的版本，使用 labelborders。

语法

```
labelbordersm(place)
labelbordersm(...,TextProperty,TextValue)
h = labelbordersm(...)
```

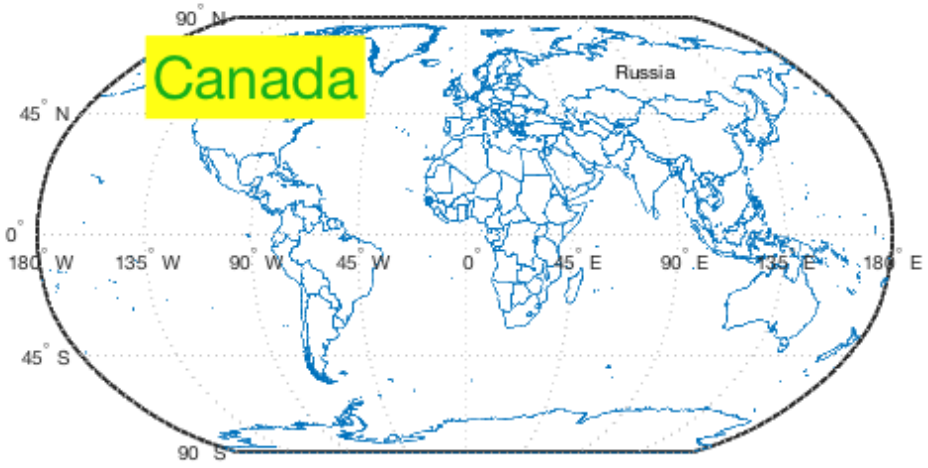
说明

labelbordersm(place) 绘制一个地方（place）的边界，可以是任何国家或美国州。地点也可能是“国家”（'countries'）以绘制所有国界线，“州”（'states'）绘制所有美国州边界，或“美国大陆”（'Continental US'）仅绘制美国大陆（对不起，关岛）。注：要绘制格鲁吉亚国家，请使用'Georgia'。要绘制美国佐治亚州，请指定'Georgia.'，并加上句号。
labelbordersm(...,TextProperty,TextValue) 指定文本格式。
h = labelbordersm(...) 返回绘制文本对象的句柄 h 。

示例

以下是使用 bordersm 函数绘制的一些边界。用简单的黑色文本标记俄罗斯，用黄色背景的大绿色文本标记加拿大：（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

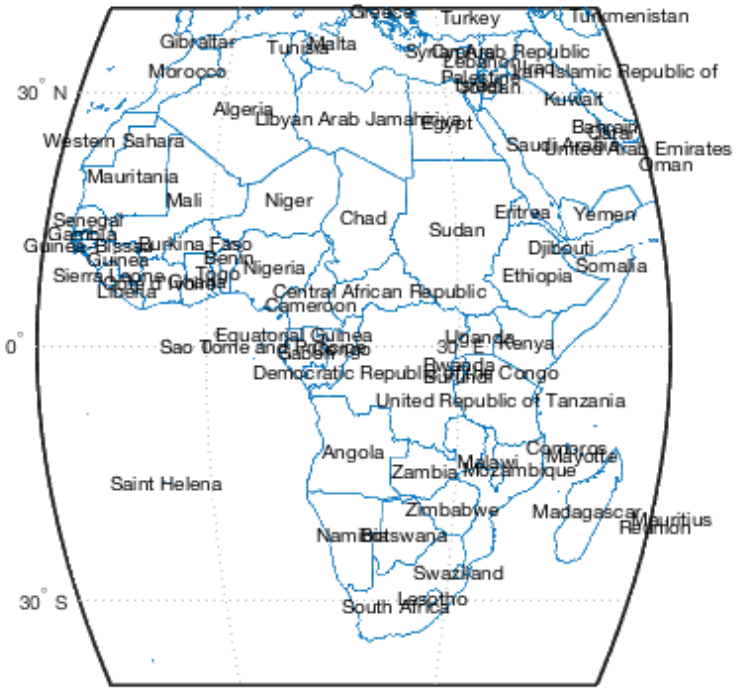
```
bordersm

labelbordersm 'Russia'

labelbordersm('Canada','fontsize',30,...
   'color',rgb('green'),'backgroundcolor',rgb('yellow'))
```



示例 2:

标记所有非洲国家:

```
worldmap('Africa')

bordersm
labelbordersm
```



作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

stipple 文档

stipple 在网格中创建图案或点画填充。这个函数主要用于显示空间地图中具有统计意义的区域。

语法

```
stipple(x,y,mask)
stipple(...,MarkerProperty,MarkerValue,...)
stipple(...,'density',DensityValue)
stipple(...,'resize',false)
h = stipple(...)
```

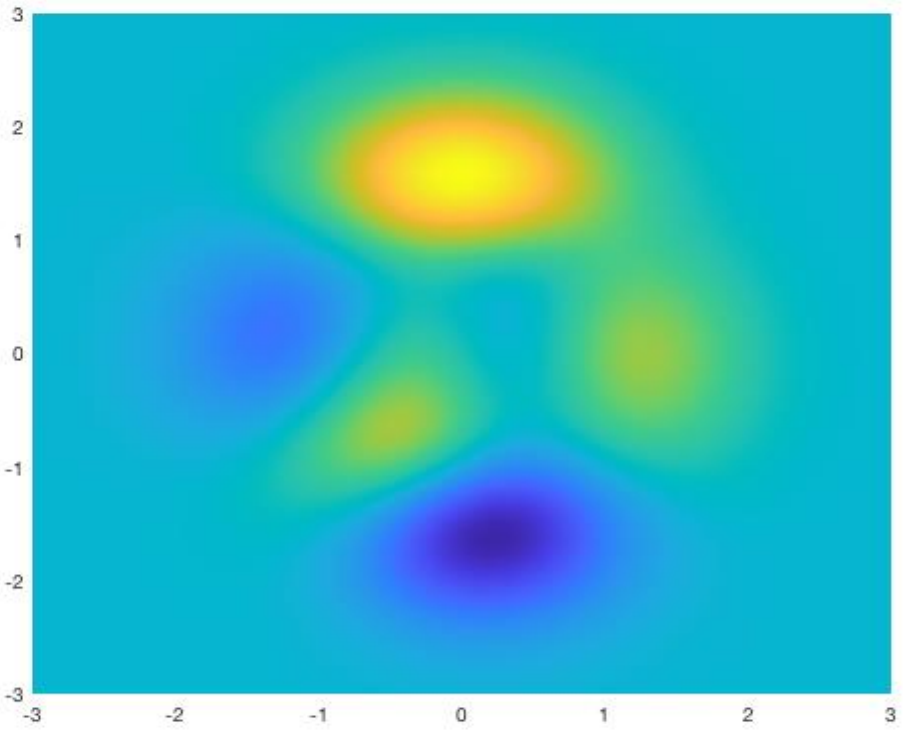
说明

stipple(x,y,mask) 在 mask 包含 true 值的 x,y 位置绘制黑点。x,y 和 mask 的尺寸必须全部匹配。
stipple(...,MarkerProperty,MarkerValue,...) 指定绘图函数接受的任何标记特性(例如,'color','marker','markersize',等)。
stipple(...,'density',DensityValue) 指定点画标记的密度。 默认密度为 100，但如果您的绘图过于拥挤，您可以指定较低的密度值（和/或调整标记大小）。
stipple(...,'resize',false) 覆盖 'density' 选项并以输入网格的精确分辨率绘制点画。 默认情况下，网格会被调整大小，因为任何大于约 100x100 的网格都会产生如此多的点画点，以至于它们下面的任何东西都会变黑。
h = stipple(...) 返回绘制的点画对象的句柄。

示例 1

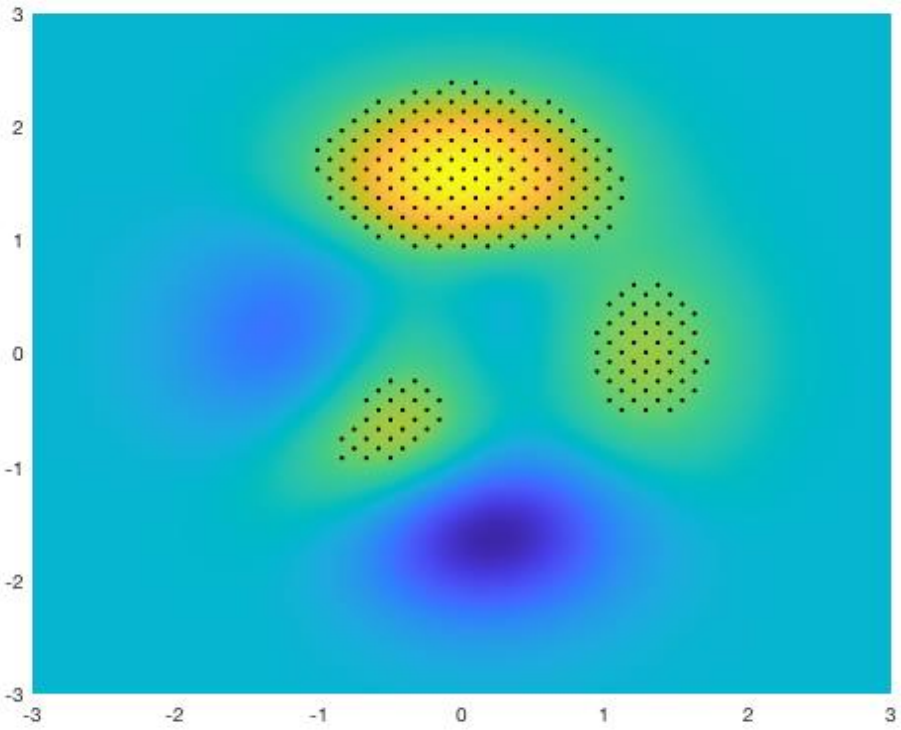
这是您可以在家中尝试的示例，该示例使用 peaks 创建 1000x1000 数据网格：

```
% 载入一些样本数据:
[X,Y,Z] = peaks(1000);

pcolor(X,Y,Z)
shading interp
hold on
```



假设 Z 超过 2.5 的任何地方都应该有点画：

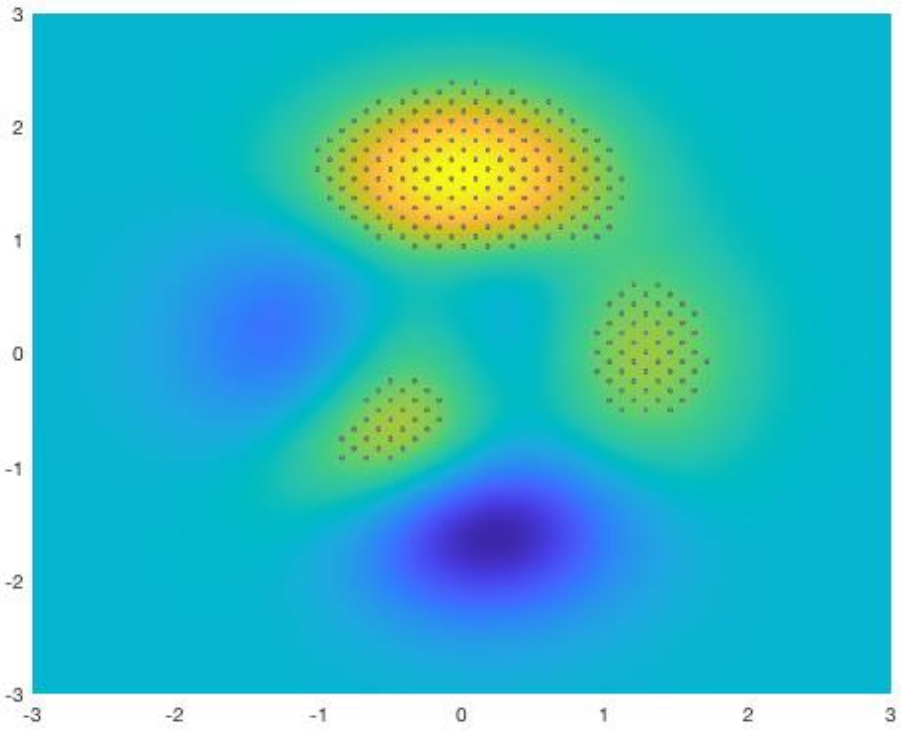
```
mask = Z>2.5;

stipple(X,Y,mask)
```



示例 2: 指定颜色

如果您更喜欢灰点而不是默认的黑点，请执行以下操作：

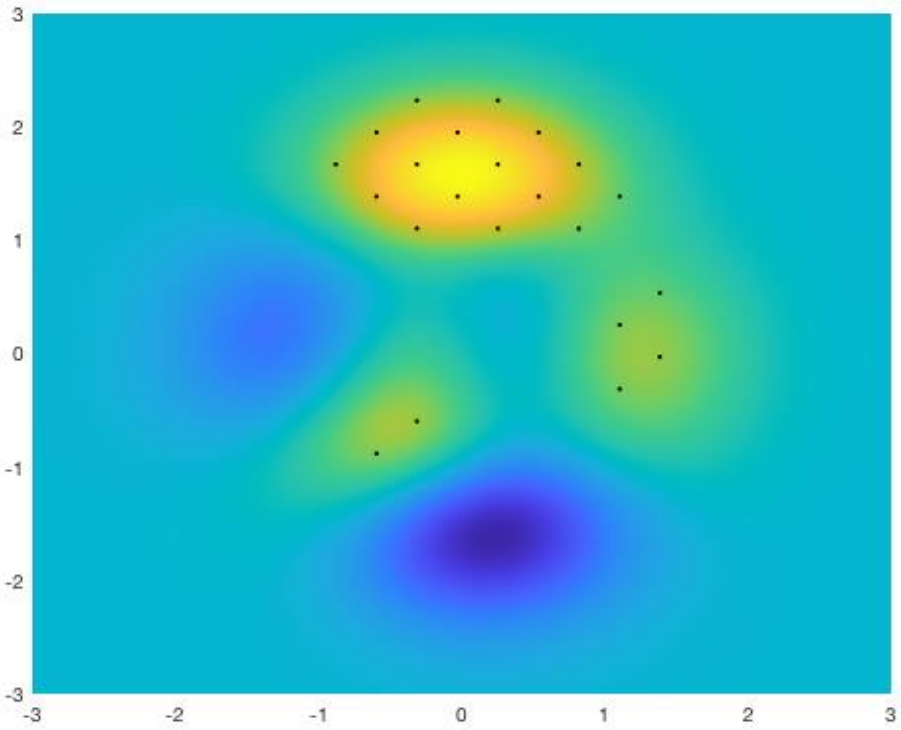
```
stipple(X,Y,mask,'color',0.5*[1 1 1])
```



示例 3: 指定密度

点太多？ 指定您想要的任何密度。 默认密度为 100，因此低于此值的值会产生较少的点，而较高的值会产生更多的点。 （如果您希望密度与输入网格完全匹配，请指定'resize',false 。）。

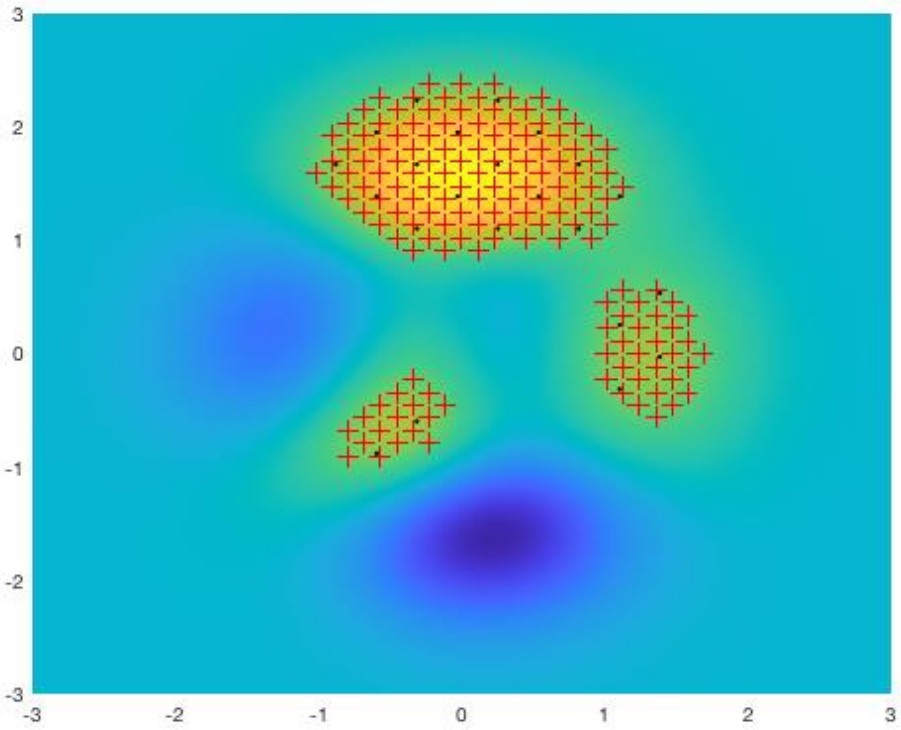
```
figure

pcolor(X,Y,Z)

shading interp
hold on
stipple(X,Y,mask,'density',30)
```



示例 4: 一次指定多个选项:

要按照您喜欢的方式设置所有内容，请指定任意数量的选项。 在这里，我们将绘制大的红色加号：

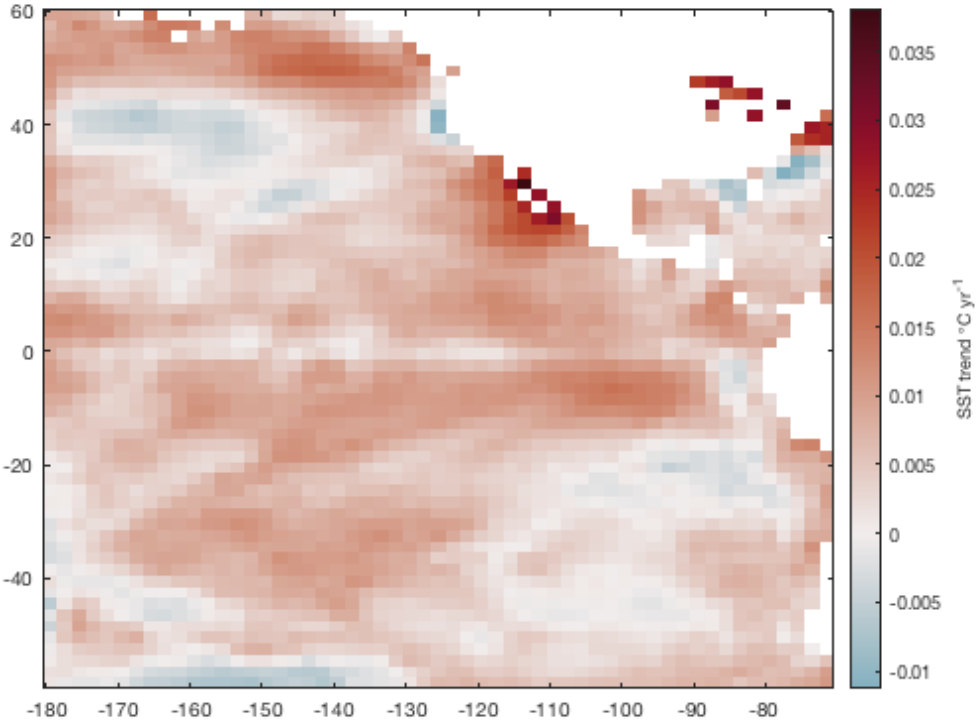
```
stipple(X,Y,mask,'density',75,'color','r','marker','+','markersize',9)
```



示例 5: 具有统计意义的区域

这是点画的实际应用：用点画表示具有统计意义的海面温度趋势区域。 首先，加载样本 pacific_sst.mat 数据集，其中包含每月网格化的海面温度数据，并使用 trend 计算趋势和对应的 p 值：

```
load pacific_sst

[tr,p] = trend(sst,12);

figure
imagescn(lon,lat,tr)
cb = colorbar;
ylabel(cb,'SST trend \circC yr^{-1}')
cmocean('balance','pivot') % 将颜色图设置为零在中间
```

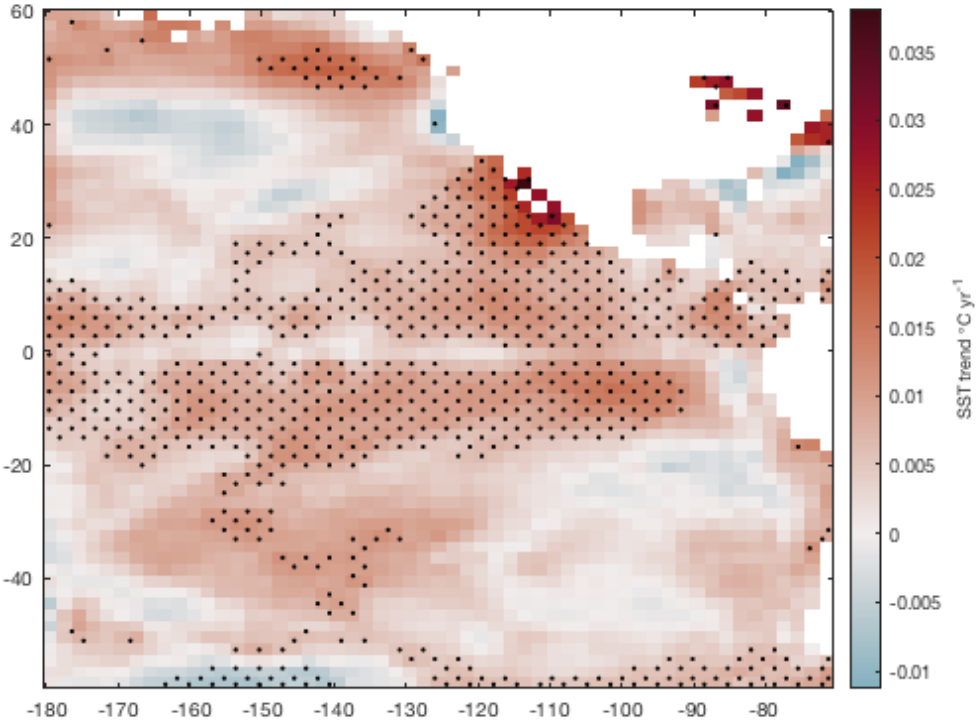


比方说，通过 p 值小于 0.01 的任何值来定义统计显着性：

```
StatisticallySignificant = p<0.01;
```

现在使用 stipple 来绘制:

```
hold on
stipple(lon,lat,StatisticallySignificant)
```



作者简介

这个函数是 Chad A. Greene2018 年 8 月写的。

stipplem 文档

stipplem 在地图坐标中在网格中创建图案或点画填充。这个函数主要用于显示空间地图中具有统计意义的区域。

这个函数需要 Matlab 地图工具箱（Mapping Toolbox）。 对于不需要地图工具箱（Mapping Toolbox）的版本，请参阅 stipple 或 globestipple。

语法

```
stipplem(lat,lon,mask)
stipplem(...,MarkerProperty,MarkerValue,...)
stipplem(...,'density',DensityValue)
stipplem(...,'resize',false)
h = stipplem(...)
```

说明

stipplem(lat,lon,mask) 在 mask 包含 true 值的 x,y 位置绘制黑点。x,y 和 mask 的尺寸必须全部匹配。

stipplem(...,MarkerProperty,MarkerValue,...) 指定绘图函数接受的任何标记特性(例如,'color','marker','markersize',等)。

stipplem(...,'density',DensityValue) 指定点画标记的密度。 默认密度为 100，但如果您的绘图过于拥挤，您可以指定较低的密度值(和/或调整标记大小)。stipplem(...,'resize',false) 覆盖 'density' 选项并以输入网格的精确分辨率绘制点画。 默认情况下，网格会被调整大小，因为任何大于约 100x100 的网格都会产生如此多的点画点，以至于它们下面的任何东西都会变黑。

h = stipplem(...) 返回绘制的点画对象的句柄。

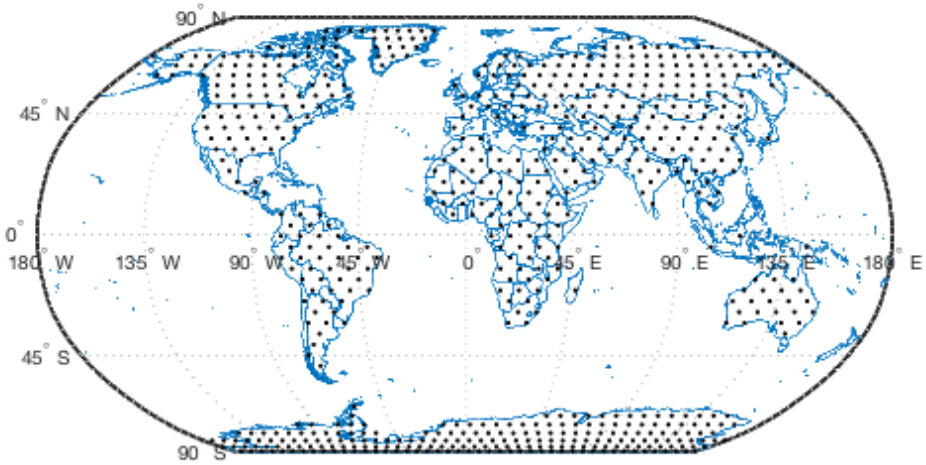
示例

在世界地图上,用点画填充所有陆地区域。要确定土地区域,请使用 cdtgrid 制作网格并使用 island 确定相应的土地网格单元。

```
[lat,lon] = cdtgrid(0.25);

land = island(lat,lon);
```

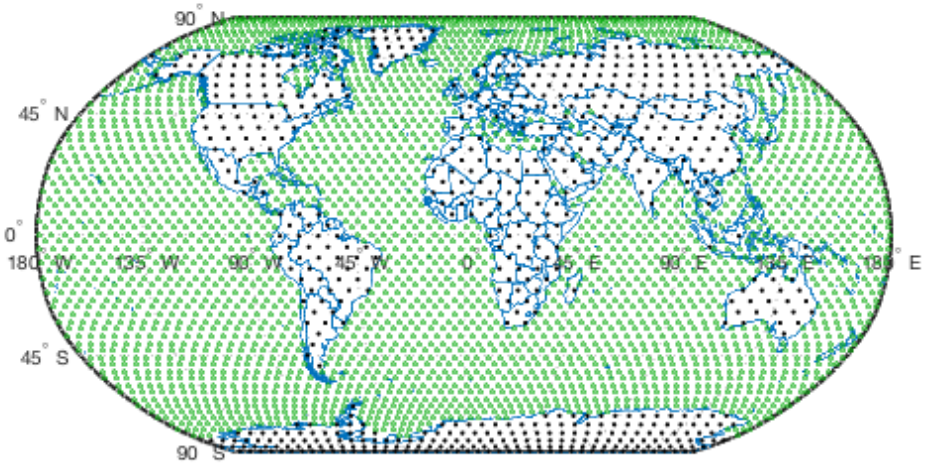
现在用 bordersm 初始化地图，并在有陆地的地方放置默认点画：（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
bordersm

stipplem(lat,lon,land)
```



现在用一个密集绿色小三角形图案的网格填充海洋：（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
stipplem(lat,lon,~land,'color',rgb('green'),...
   'marker','^','markersize',2,'density',150)
```



作者简介

这个函数和支持文档是 Chad A. Greene2018 年 8 月写的。

quiversc 文档

quiversc 缩放矢量箭头的密集网格，以便在绘制轴之前舒适地适应轴。

语法

```
quiversc(x,y,u,v)
quiversc(...,'density',DensityValue)
quiversc(...,scale)
quiversc(...,LineSpec)
quiversc(...,LineSpec,'filled')
quiversc(...,'Name',Value)
h = quiversc(...)
```

说明

quiversc(x,y,u,v) 在 x 和 y 中每个对应元素对中指定的坐标处将向量绘制为箭头。矩阵 x、y、u 和 v 都必须具有相同的大小并包含相应的位置和速度分量。默认情况下，箭头被缩放为不重叠，但您可以根据需要将它们缩放得更长或更短。

quiversc(...,'density',DensityFactor) 指定箭头的密度。DensityFactor 定义了绘制的箭头数量。默认 DensityFactor 为 50，这意味着 hypot(Nrows,Ncols)=50，但如果您的图太拥挤，您可以指定较低的 DensityFactor （和/或调整标记大小）。

quiversc(...,scale) 自动缩放箭头的长度以适应网格，然后按因子比例拉伸它们。 scale = 2 将它们的相对长度加倍，scale = 0.5 将长度减半。使用 scale = 0 绘制没有自动缩放的速度矢量。您还可以通过选择“绘图编辑”工具、选择箭头对象、打开“属性编辑器”并调整“长度”滑块来调整绘制箭头后的长度。

quiversc(...,LineSpec) 使用任何有效的 LineSpec 指定线型、标记符号和颜色。 quiversc 在向量的原点绘制标记。、

quiversc(...,LineSpec,'filled') 填充 LineSpec 指定的标记。

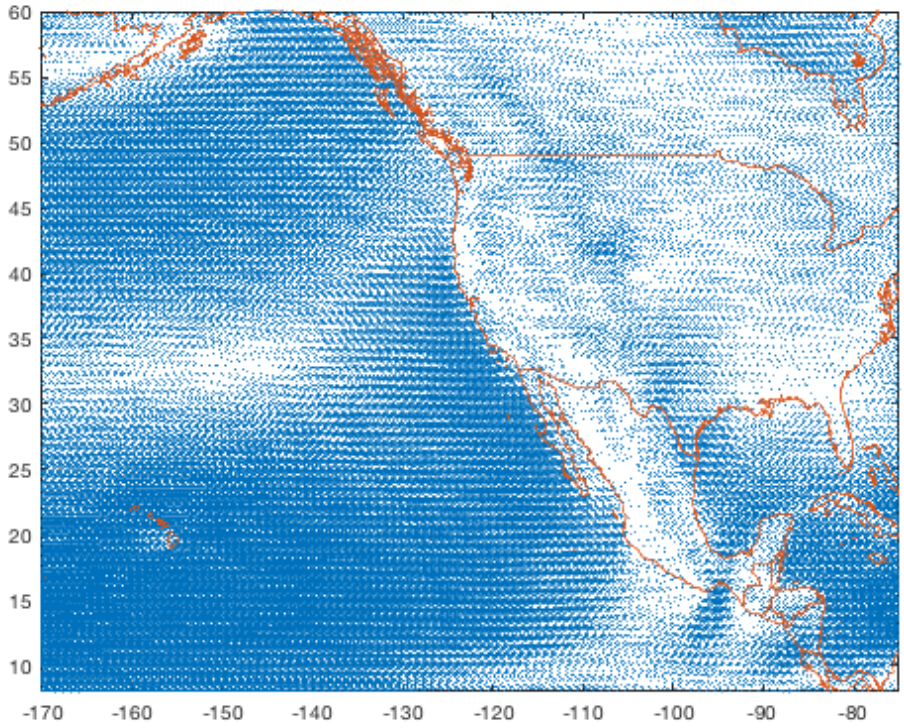
quiversc(...,'Name',Value) 为函数创建的 quiver 对象指定属性名称和属性值对。

h = quiversc(...) 返回 quiver 对象句柄 h。、

问题

问题是：当您尝试使用 Matlab 的内置 quiver 函数绘制密集的风模式网格时，事情可能会变得过于拥挤而无法理解正在发生的事情：

```
load pacific_wind

figure
quiver(lon,lat,u10,v10)
axis tight % 消除空白
borders    % 国界线
```

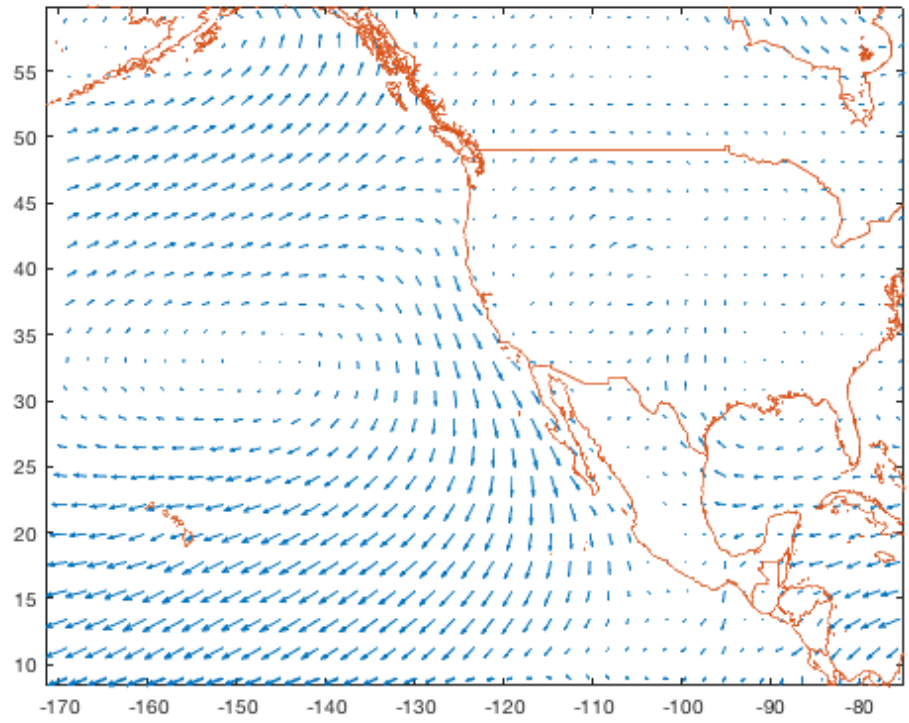


示例 1

以上，417x761 的风矢量网格意味着该图中有超过 300,000 个箭头。 将其置于上下文中，有 761 个箭头跨越绘图的宽度，这可能与绘图在屏幕上占据的像素数有关。 换句话说，如果每个箭头大约是一个像素的大小，那么它们就太小了，无法从中获得任何理解。

所以尝试同样的事情，但这次使用 quiversc：

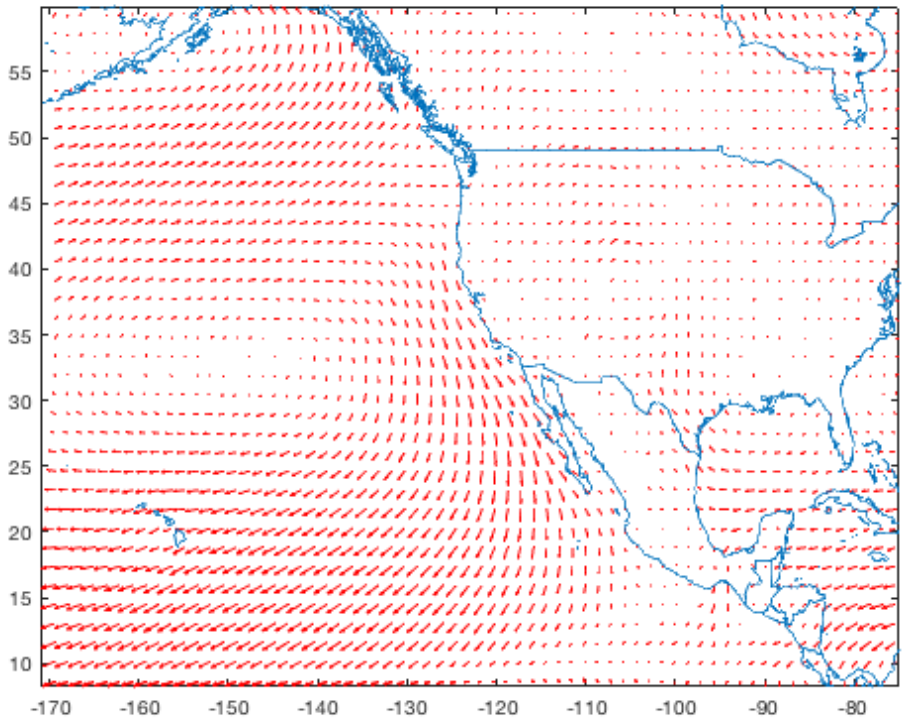
```
figure

quiversc(lon,lat,u10,v10)

axis tight
borders
```



示例 2:定义箭头属性

使箭头变为红色并通过将密度设置为 30 而不是默认的 50，绘制更少的箭头：

```
figure

quiversc(lon,lat,u10,v10,'r','density',75)
axis tight
borders
```



作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

patchsc 文档

patchsc 使用按数值缩放的面颜色绘制填充对象。

语法

```
patchsc(x,y,z)
patchsc(...,'colormap',cmap)
patchsc(...,'caxis',ColorAxisLimits)
patchsc(...,'PatchProperty',Value,...)
h = patchsc(...)
```

说明

patchsc(x,y,z) 按 z 中的数值进行颜色缩放来绘制元胞数组 x,y。 z 的维度必须与元胞数组 x 和 y 的维度匹配。 x 和 y 可以包含由 NaN 分隔的多个部分。p
patchsc(...,'colormap',cmap) 指定面颜色将映射到的颜色图。 如果未指定颜色图，则将使用您的默认颜色图。
patchsc(...,'caxis',ColorAxisLimits) 设置颜色轴限制。 这与其他功能（如 imagesc 或 surf 允许在绘图后设置颜色限制）不同。 patchsc 不允许在绘图后更改颜色轴限制。 默认限制为 [min(z) max(z)]。
patchsc(...,'PatchProperty',Value,...) 指定任何填充属性。
h = patchsc(...) 返回所有填充对象的句柄。对应于每个填充对象的数据元素包含在句柄'tag'属性中。

示例

对于此示例，制作拉丁美洲国家的地图，按平均海拔进行颜色缩放。 首先加载 borders 函数用于绘制国家轮廓的数据，然后将数据集缩减为仅包括拉丁美洲：

```
load('borderdata.mat');

% 拉美国家索引
ind = [8 17 21 33 38 39 41 48 49 55 59 75 77 78 79 109 120 158,...
   159 161 162 165 174 211 214 226 241 242];

% 将数据集修剪为拉丁美洲：
lat = lat(ind);
lon = lon(ind);
z = z(ind);
```

现在让我们看看我们正在绘制的数据：

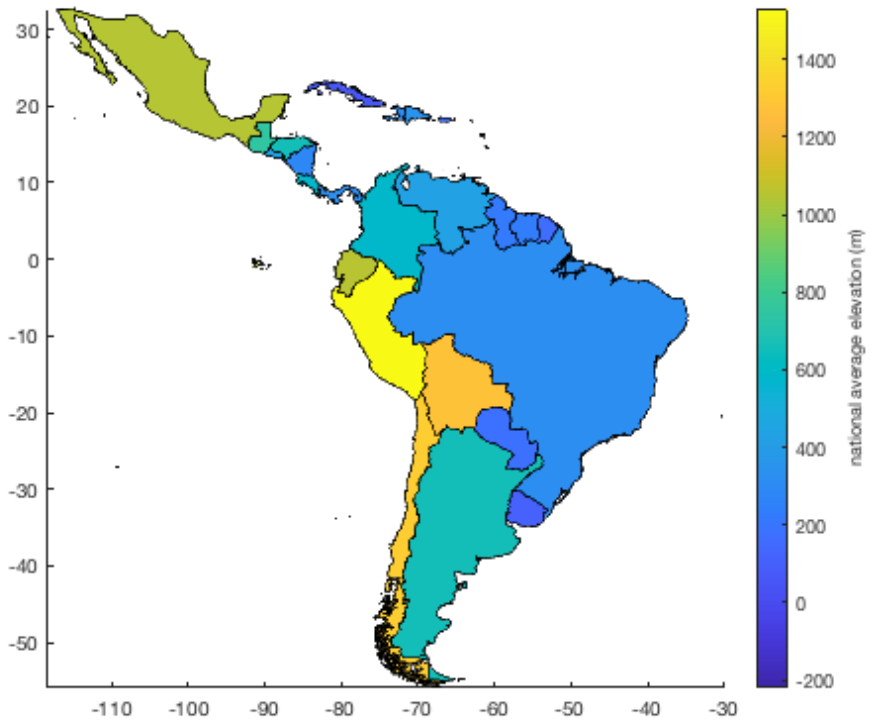
```
whos lat lon z
```

```
  Name        Size              Bytes  Class     Attributes


  lat        28x1              355776  cell
```

```
  lon        28x1              355776  cell

  z          28x1                 224  double
```

请注意，lat 和 lon 是元胞数组，每个数组都包含不同国家/地区的轮廓，而 z 是一个数值数组，仅包含每个国家/地区的一个数字（平均海拔）。这正是 patchsc 喜欢它的数据的方式，所以让我们绘制它：

```
% 绘制按平均海拔颜色缩放的国家/地区：
patchsc(lon,lat,z)

axis equal tight
cb = colorbar;
ylabel(cb,'national average elevation (m)')
```



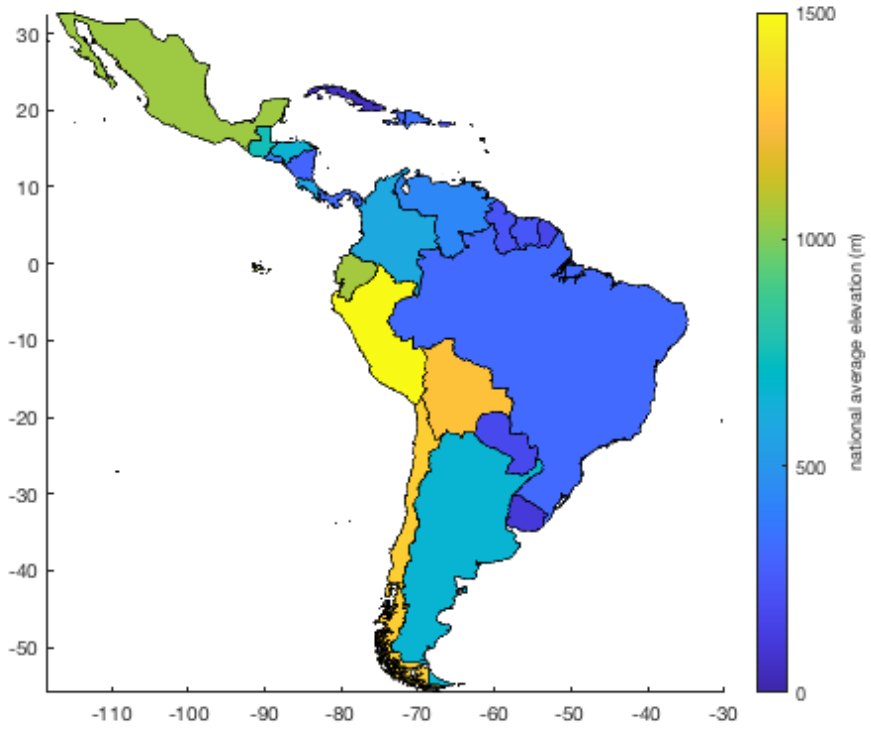
在上图中，颜色条轴下降到 -220 m。 这是因为全国平均高程数据集是根据粗分辨率网格创建的，该网格在瓜德罗普岛进行插值赋予其平均负值。 如果我们可以通过简单地输入以下语句来修复它，那就太好了

```
caxis([0 1500])
```

重置颜色轴限制，但不幸的是，这不适用于 patchsc 对象，因为它们没有动态绑定到颜色图。 因此，我们必须再次绘制它，在调用 patchsc 时指定 caxis 限制：

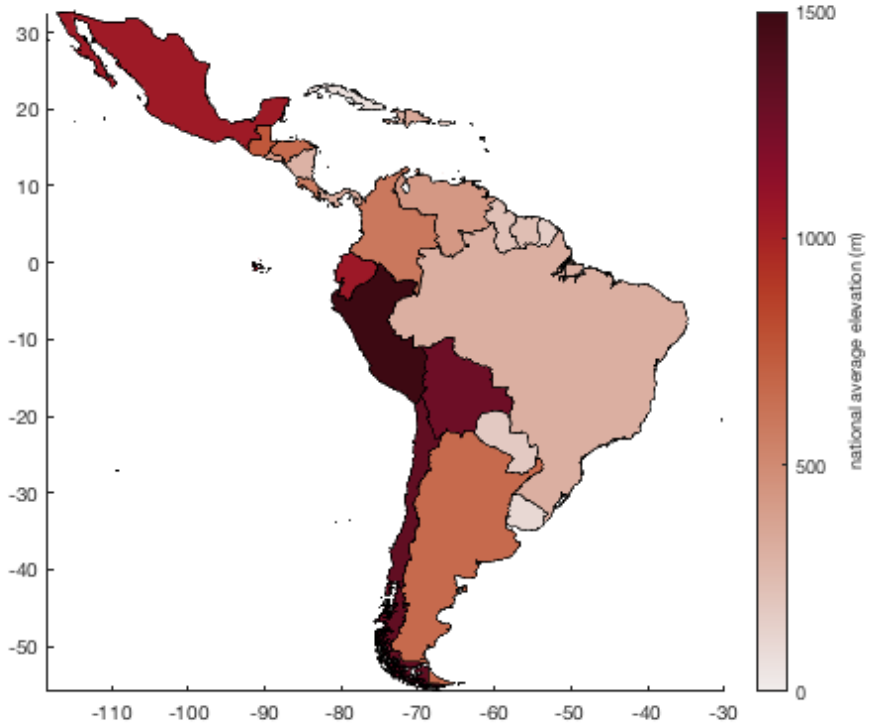
```
figure

patchsc(lon,lat,z,'caxis',[0 1500])
```

```
axis equal tight
cb = colorbar;
ylabel(cb,'national average elevation (m)')
```



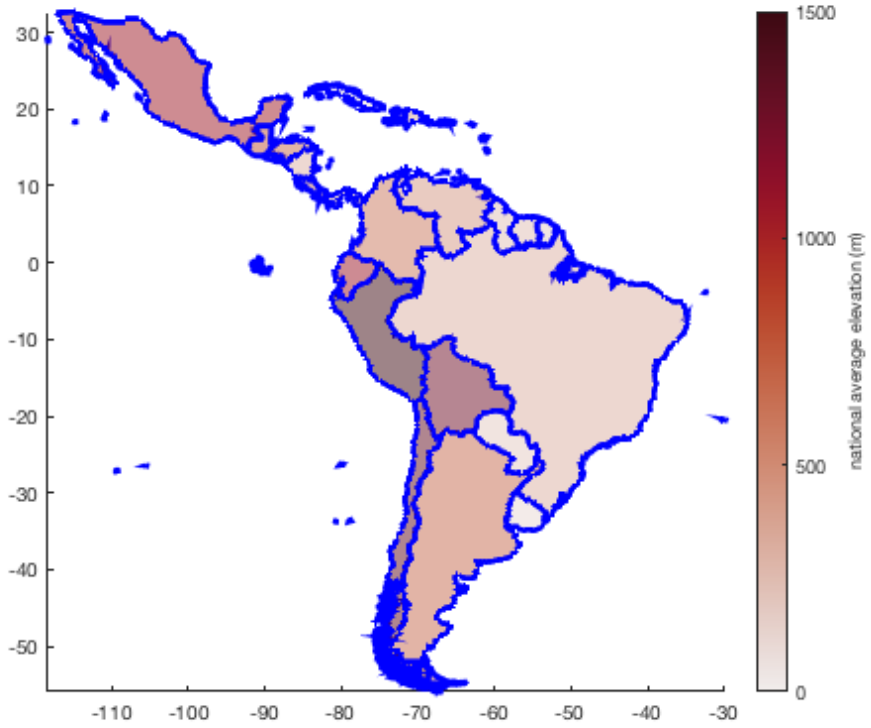
再做一次，这次指定 cmocean *amp* 颜色图：

```
figure

patchsc(lon,lat,z,'caxis',[0 1500],'colormap',cmocean('amp'))

axis equal tight
cb = colorbar;
ylabel(cb,'national average elevation (m)')
```



您还可以指定任何填充属性，包括 facealpha、linewidth、edgecolor 等。 所以这次设置 caxis 限制、颜色图、边缘颜色、边缘线宽和透明度：

```
figure

patchsc(lon,lat,z,'caxis',[0 1500],'caxis',[2 24],...
   'colormap',cmocean('amp'),...
   'edgecolor','blue',...
   'linewidth',3,...
   'facealpha',0.5)

axis equal tight
cb = colorbar;
ylabel(cb,'national average elevation (m)')
```



作者简介

这个函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene2017 年 5 月写的。这个函数是 Climate Data Toolbox for Matlab 的一部分。

地球

- globeimage 创建一个“蓝色大理石”3D 地球图像。
- globeplot 函数在地球上绘制地理参考数据。
- globepcolor 在地球上绘制地理参考数据，其中颜色按数据值缩放。
- globesurf 在地球上绘制地理参考数据，其中矩阵 Z 中的值绘制为地球上方的高度。
- globecontour 根据网格数据在地球上绘制等值线。
- globescatter 将地理参考数据绘制为地球上的颜色标度标记。
- globeborders 在地球上绘制政区边界。
- globequiver 在地球上绘制带有分量 (u,v) 的地理参考向量。
- globestipple 在地球的某个区域上创建填充或点画。
- globegraticule 绘制一个格线地球仪 可选输入控制经纬网的外观和行为。、
- globefill 绘制一个填充的地球。

globeimage 文档

globeimage 函数创建一个“蓝色大理石”3D 地球图像

语法

```
globeimage
globeimage('gray')
globeimage(I)
globeimage('radius',GlobeRadius)
h = globeimage(...)
```

说明

globeimage 在三维地球上绘制蓝色大理石地球图像。.
globeimage('gray') 以灰度绘制地球图像。
globeimage(I) 拉伸图像 I，使其顶部和底部延伸到两极，其边缘在国际日期变更线处相交。s
globeimage(...,'radius',GlobeRadius) 指定地球的半径。 默认的 GlobeRadius 为 6371，以公里为单位的地球标准半径。
h = globeimage(...) 返回绘制对象的句柄 h 。

示例 1

最简单的例子，只需要键入 globeimage:

```
globeimage
```

示例 2

制作灰度地球图像并设置视角：

```
figure

globeimage('gray')
view(45,20)
```

示例 3:你自己的图像

如果您正在处理地球以外的行星，并且想要使用该行星的图像，则此选项适合您。 但它真的可以是*任何*图像。

下面，加载 CDT 徽标并将其绘制在地球上。 使用 globeborders 添加政区边界：

```
I = imread('cdt.jpg');

figure
globeimage(I)
axis tight
globeborders('color',rgb('orange'))
view(10,30)
```

作者简介

这个函数是 Climate Data Toolbox for Matlab 的一部分。函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 写的。

globeplot 文档

globeplot 函数在地球上绘制地理参考数据。

语法

```
globeplot(lat,lon)
globeplot(lat,lon,LineSpec)
globeplot(...,PropertyName,PropertyValue,...)
globeplot(...,'radius',GlobeRadius)
h = globeplot(...)
```

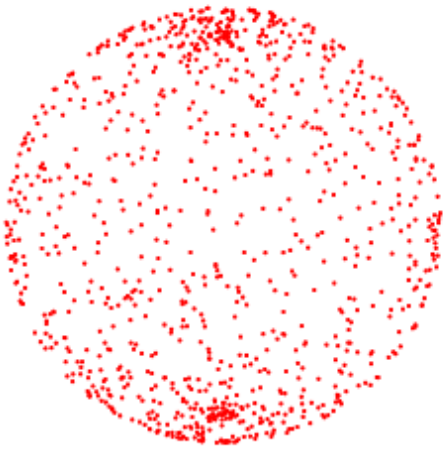
说明

globeplot(lat,lon) 创建由纬度和经度指定的地理参考数据点的二维线图。
globeplot(lat,lon,LineSpec) 指定二维线图中的线型、标记符号和/或颜色。
globeplot(...,PropertyName,PropertyValue,...) 指定二维线图中的线或标记属性。
globeplot(...,'radius',GlobeRadius) 指定地球的半径为 GlobeRadius。
h = globeplot(...) 返回绘制对象的句柄 h 。

示例 1

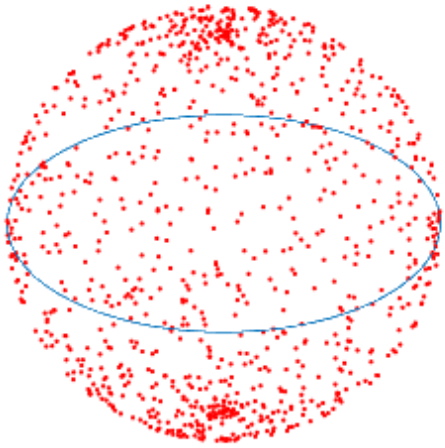
在地球仪上绘制随机分布的红点：

```
N = 1000;

lat = 180*rand(N,1)-90;

lon = 360*rand(N,1)-180;



figure

globeplot(lat,lon,'r.')
axis tight
```



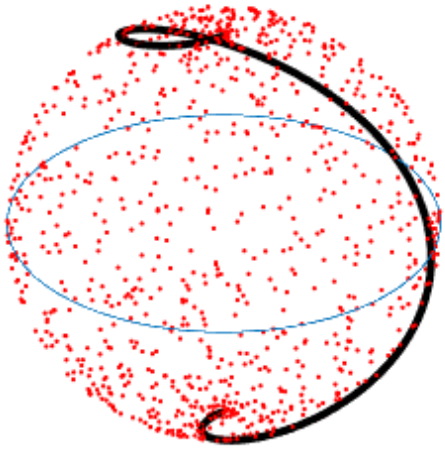
通过定义一个 1000 个元素的零纬度数组向地球添加一个赤道（这可以类似地使用 globegraticule 完成。）

```
lon = linspace(-180,180,1000);

lat = zeros(1,1000);



globeplot(lat,lon)
```



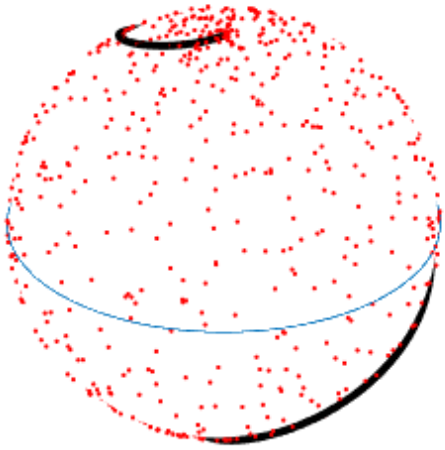
绘制一条从南极到北极环绕世界一次的粗黑线：

```
lat = linspace(-90,90,1000);

lon = linspace(-180,180,1000);



globeplot(lat,lon,'linewidth',4,'color','k')
```



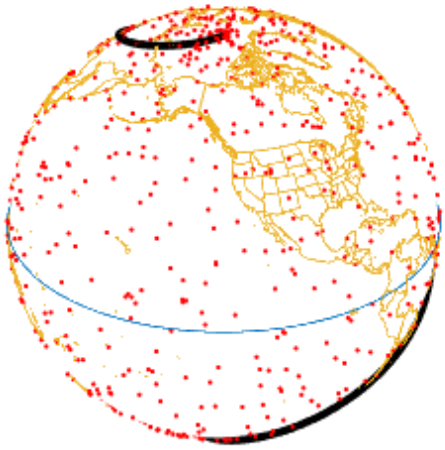
地球目前是透明的。 要填充它，请使用 globefill：

```
globefill
```



需要更多内容？ 使用 globeborders 添加政区边界：

```
globeborders
```



作者简介

函数和支持文档是由德克萨斯大学奥斯汀地球物理研究所（UTIG）的 Chad A. Greene 和 Natalie S. Wolfenbarger 2019 年为 Climate Data Toolbox for Matlab 写的。

globepcolor 文档

globepcolor 函数在地球上绘制地理参考数据，其中颜色按数据值缩放。

语法

```
globepcolor(lat,lon,C)
globepcolor(...,'radius',GlobeRadius)
h = globepcolor(...)
```

说明

globepcolor(lat,lon,C) 创建由 C 给定的网格值的伪彩色地球图。输入 lat 和 lon 与 C 的大小相同，可以使用 meshgrid 函数为任意域定义。

globepcolor(...,'radius',GlobeRadius) 指定地球的半径。 默认的 GlobeRadius 为 6371，以公里为单位的地球标准半径。

h = globepcolor(...) 返回绘制对象的句柄 h。

示例 1:全球地形

对于此示例，绘制颜色缩放的全局地形。使用 cdtgrid 创建四分之一度网格，并使用 topo_interp 获取相应的地形。 使用 cmocean 设置颜色图：

```
[Lat,Lon] = cdtgrid(0.25);

Z = topo_interp(Lat,Lon);



globepcolor(Lat,Lon,Z);

cmocean('topo','pivot')
axis tight % 移除空白
```

示例 2: 地球半径

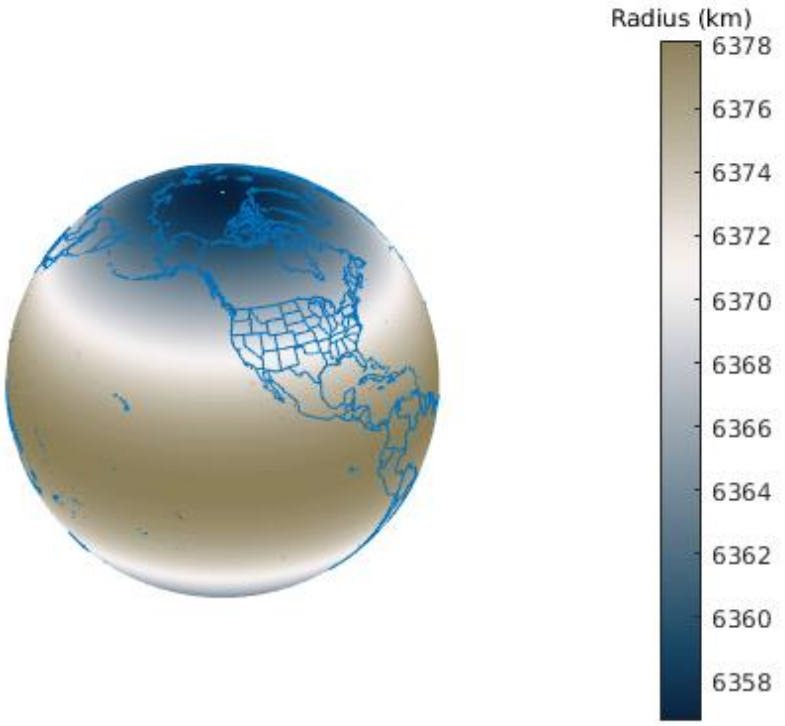
CDT 中的 globe* 函数将地球绘制为一个球体，但实际上它更像是一个扁球体（又名椭球体），或者半径是纬度函数的球体。 我们可以使用 globepcolor 来描绘球形地球和椭球形地球之间的区别，如下所示。 使用 cdtgrid 创建全局网格并使用 earth_radius 获取每个网格单元的纬度相关半径：

```
[Lat,Lon] = cdtgrid;



R = earth_radius(Lat,'km');
```

由于地球半径是纬度的函数，因此请显示 6371 公里的标准地球半径准确高估与低估地球真实半径的位置。 使用 cmocean 中的'pivot'选项使颜色围绕 6371 公里值周围：

```
figure

globepcolor(Lat,Lon,R);

axis tight

c = colorbar;
set(get(c,'title'),'string','Radius (km)');
cmocean('diff','pivot',6371) % 设置颜色图
```

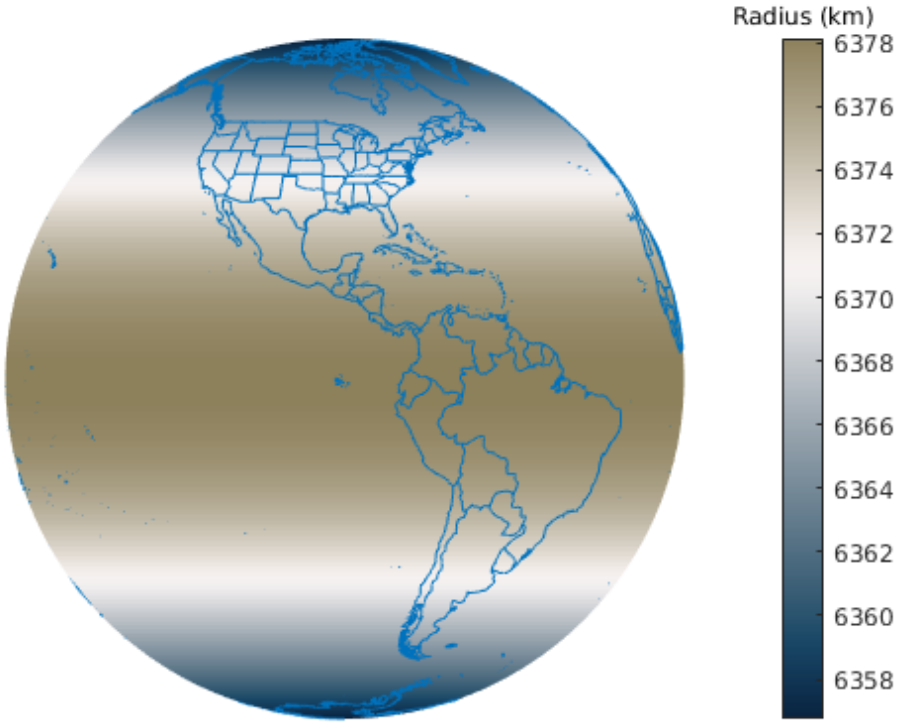
```
globeborders % 增加政区边界
```



上图中，地表为棕色的地方，地球半径大于标准的 6371 公里，地表为蓝色的地方，地表到地心的距离小于 6371 公里。

将视图从极地调整为赤道：

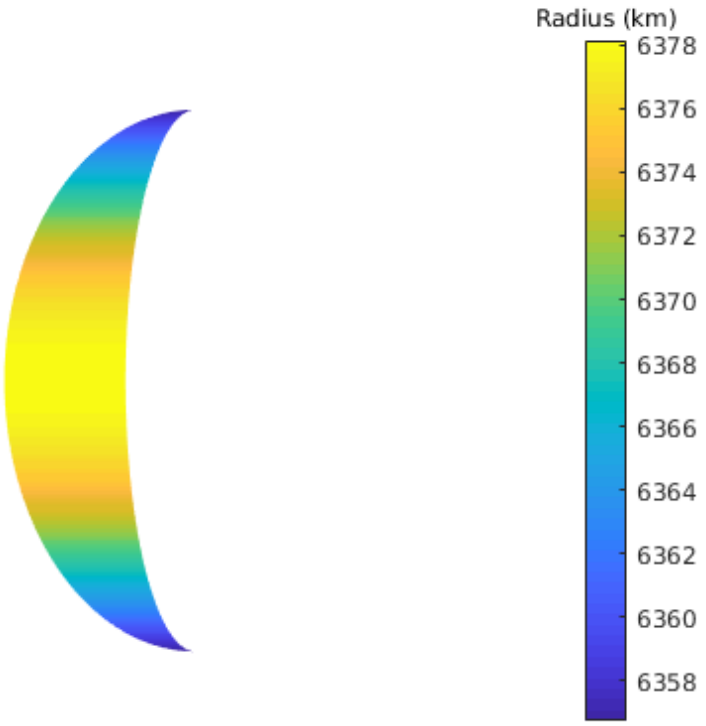
```
view([0 0])
```



示例 3: 只是一片地球

绘制地球半径，仅显示 0 到 30 度之间的经度：

```
figure

[Lon,Lat] = meshgrid(0:30,-90:90);

R = earth_radius(Lat,'km');

globepcolor(Lat,Lon,R);
c = colorbar;
set(get(c,'title'),'string','Radius (km)');
view([135 0])
```



作者简介

这个函数和支持文档是由 Natalie S. Wolfenbarger 2019 年为 Climate Data Toolbox for Matlab 写的。

globesurf 文档

globesurf 函数在地球上绘制地理参考数据，其中矩阵 Z 中的值绘制为地球上方的高度。

语法

```
globesurf(lat,lon,Z)
globesurf(lat,lon,Z,C)
globesurf(...,'exaggeration',exaggerationFactor)
globesurf(...,'radius',GlobeRadius)
h = globesurf(...)
```

说明

globesurf(lat,lon,Z) 在半径为 6371 的地球上方的高度处绘制由 Z 给出的地理参考值，其中 6371 对应于以公里为单位的地球平均半径。 输入 lat 和 lon 与 Z 的大小相同，可以使用 meshgrid 函数为任意域定义。

globesurf(lat,lon,Z,C) 通过与 Z 大小相同的矩阵或作为 RGB 三元组的 m×n×3 数组指定地理参考值的颜色，其中 Z 是 m× n。

globesurf(...,'exaggeration',exaggerationFactor) 按 exaggerationFactor 指定的因子缩放地理参考值的绘制高度。

globesurf(...,'radius',GlobeRadius) 将地理参考值绘制为高于由 GlobeRadius 指定的半径的地球的高度。

h = globesurf(...) 返回绘制对象的句柄 h。

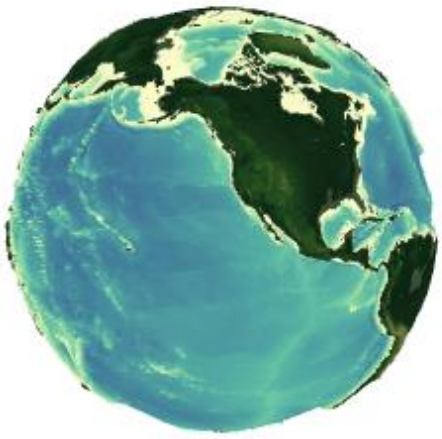
示例 1

对于此示例，绘制颜色缩放的全球地形。 使用 cdtgrid 创建四分之一度网格，并使用 topo_interp 获取相应的地形。 这是我们要绘制的数据：

```
% 创建 0.25 度网格:
[Lat,Lon] = cdtgrid(1/4);

% 获取对应的地形:
Z = topo_interp(Lat,Lon);
```

绘制表面地形，放大 50 倍。使用 'pivot' 选项设置带有 cmocean 的颜色图，将零放在颜色图的中间。

```
figure

globesurf(Lat,Lon,Z,'exag',50)
axis tight
cmocean('topo','pivot')
```



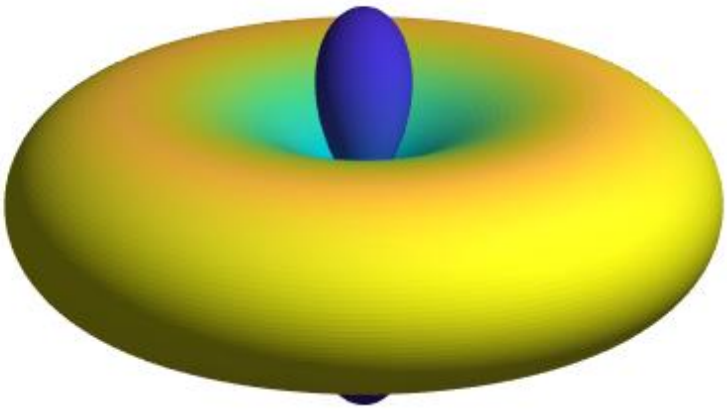
设置视角并调整光照位置和材料反射率：

```
view(60,20)

camlight

material dull
```

示例 2

地球半径与平均地球半径的绘图偏差被夸大了数百万倍。 使用 earth_radius 获得地球的椭球半径：

```
[lat,lon] = cdtgrid;



R = earth_radius(lat,'km');
dR = R - 6371;

figure
globesurf(lat,lon,dR,'exag',1e6)
axis tight

% 调整视角:
view(10,20)
camlight
material dull
```



作者简介

这个函数和支持文档是由 Natalie S. Wolfenbarger 2019 年为 Climate Data Toolbox for Matlab 写的。

globecontour 文档

globecontour 函数根据网格数据在地球上绘制等值线。 注意：这些等值线不是等值线图形对象，也没有链接到当前的颜色图。

语法

```
globecontour(lat,lon,Z)
globecontour(lat,lon,Z,n)
globecontour(lat,lon,Z,v)
globecontour(...,PropertyName,PropertyValue)
globecontour(...,'radius',GlobeRadius)
h = globecontour(...)
```

说明

globecontour(lat,lon,Z) 在半径为 6371 的地球上绘制 Z 中地理参考数据的等值线，其中 6371 对应于以公里为单位的地球平均半径。 输入 lat 和 lon 与 Z 的大小相同，可以使用 meshgrid 函数为任意域定义。
globecontour(lat,lon,Z,n)) 绘制与 Z 中的地理参考数据相对应的 n 条等距等值线。
globecontour(lat,lon,Z,v) 在向量 v 指定的高度绘制等值线。
globecontour(...,PropertyName,PropertyValue) 指定用于控制等值线外观和行为的线属性。
globecontour(...,'radius',GlobeRadius) 将地球的半径指定为 GlobeRadius。 默认 GlobeRadius 为 6371。
h = globecontour(...) 返回绘制对象的句柄 h。

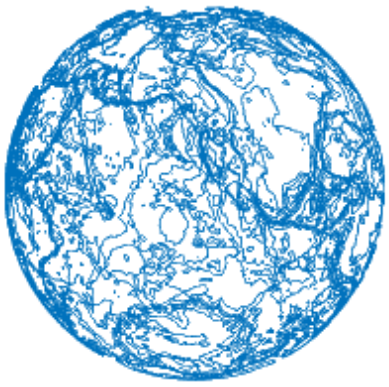
示例 1: 地形等值线

首先使用 cdtgrid 和 topo_interp 来获取全球地形：

```
[lat,lon] = cdtgrid;

topo = topo_interp(lat,lon);
```

现在绘制 10 条描绘地球地形的等值线：

```
figure

globecontour(lat,lon,topo,10)
```

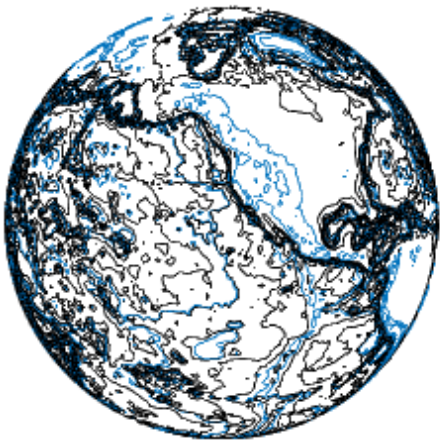


使用 globefill 将地球颜色设置为白色：

```
hold on
globefill
axis tight
```

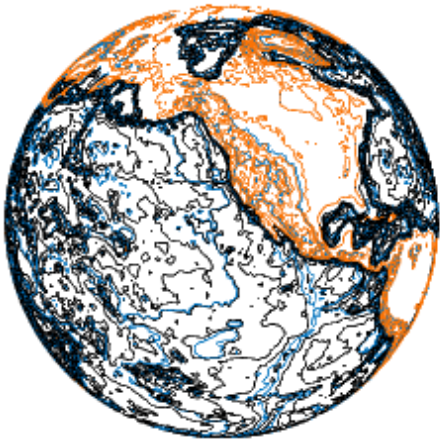
添加黑色等值线，以 500 m 的间距描绘地球海洋从海平面以下 7000 m 到海平面的地形：

```
globecontour(lat,lon,topo,-7000:500:0,'color','k')
```



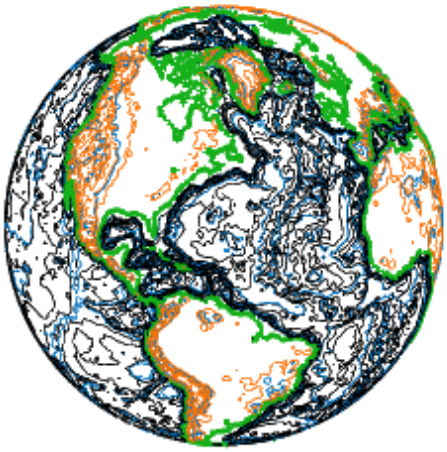
添加橙色等值线，以 500 m 的间距描绘地球陆地从海平面到海拔 5500 m 的地形：

```
globecontour(lat,lon,topo,0:500:5500,'color',rgb('orange'))
```



将海平面（海岸线）绘制为一条粗绿线：

```
globecontour(lat,lon,topo,[0 0],'color',rgb('green'),'linewidth',3)
view([30 30])
```



示例 2: 表面气压

对于此示例，绘制 2017 年 5 月的全球表面气压异常。首先，加载数据：

```
filename = 'ERA_Interim_2017.nc';
sp = ncread(filename,'sp');
lat = double(ncread(filename,'latitude'));
lon = double(ncread(filename,'longitude'));

% 网格化 lat,lon 数组：
[Lat,Lon] = meshgrid(lat,lon);

% 计算五月份表面气压异常：
spa = sp(:,:,5) - mean(sp,3);
```

现在将表面气压异常绘制为蓝色大理石地球图顶部的 30 条等值线：

```
figure

globeimage

globecontour(Lat,Lon,spa,30)

view(45,20)
```

```
axis tight
```

作者简介

这个函数和支持文档是由 Natalie S. Wolfenbarger 2019 年为 Climate Data Toolbox for Matlab 写的。

globescatter 文档

globescatter 函数将地理参考数据绘制为地球上的颜色标度标记。

语法

```
globescatter(lat,lon)
globescatter(lat,lon,sz)
globescatter(lat,lon,sz,c)
globescatter(...,'filled')
globescatter(...,PropertyName,PropertyValue,...)
globescatter(...,'radius',GlobeRadius)
h = globescatter(...)
```

说明

globescatter(lat,lon) 在地球上创建由 lat,lon 指定的地理参考数据点的散点图。
globescatter(lat,lon,sz) 指定圆圈的大小。
globescatter(lat,lon,sz,c) 用 C 指定的颜色绘制每个圆。

- 如果 c 是包含颜色名称的 RGB 三元组或字符向量或字符串，则所有圆都用指定的颜色绘制。
- 如果 c 是一个三列矩阵，其中 c 中的行数等于 lat 和 lon 的长度，那么 c 的每一行都指定了相应圆的 RGB 颜色值。
- 如果 c 是长度等于 lat 和 lon 长度的向量，则 c 中的值线性映射到当前颜色图中的颜色。

globescatter(...,'filled') 使用前面语法中的任何输入参数组合来填充圆圈。
globescatter(...,PropertyName,PropertyValue,...) 使用一个或多个名称-值对参数修改散点图。
globescatter(...,'radius',GlobeRadius) 将地球的半径指定为 GlobeRadius。 默认 GlobeRadius 为 6371。
h = globescatter(...) 返回绘制对象的句柄 h。

示例 1

绘制美国地质调查局记录的世界上 20 次最大地震的位置：

```
lat = [-38.14, 60.91, 3.3, 38.3, 52.62, -36.12, 0.96, 51.25, 28.36,...
    2.33, 2.09, 51.5, 53.49, -5.05, -28.29, 44.87, 54.49, -4.44, -16.27, 39.21];
lon = [-73.41, -147.34, 95.98, 142.37, 159.78, -72.9, -79.37, 178.72,...
    96.45, 93.06, 97.11, -175.63, -162.83, 131.61, -69.85, 149.48,...
    160.47, 101.37, -73.64, 144.59];

globescatter(lat,lon)
globefill
globeborders
camlight % 加亮
material dull % 使它不那么闪亮
axis tight
```

以上，我们用 globefill 填充地球内部，用 globeborders 绘制政区边界，并通过 camlight 添加了一种立体感。

现在执行与上述相同的操作，但根据地震震级缩放标记大小：

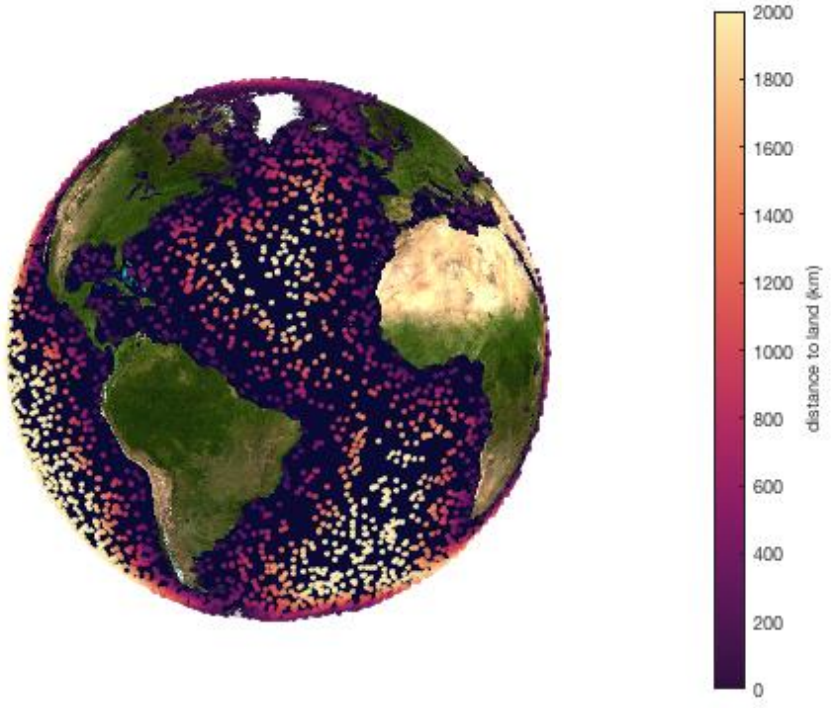
```
% 地震震级:
m = [9.5, 9.2, 9.1, 9.1, 9, 8.8, 8.8, 8.7, 8.6, 8.6, 8.6, 8.6, 8.6,...
    8.5, 8.5, 8.5, 8.4, 8.4, 8.4, 8.4];

globescatter(lat,lon,10.^m/1e6)
```

示例 2

想象一下，一群漂流者散落在海洋中，就像 Argo 浮标一样。 使用 dist2coast 获取到最近陆地的距离并将它们绘制为缩放颜色：

```
% 制作 10000 个随机分布的数据点:
N = 10000;
lat = 180*rand(N,1)-90;
lon = 360*rand(N,1)-180;

% 去掉陆地:
land = island(lat,lon);
lat(land) = [];
lon(land) = [];

% 获取到陆地的距离:
d = dist2coast(lat,lon);

figure
globescatter(lat,lon,10,d,'filled')
globeimage
cmocean -matter
axis tight
view(50,10)
cb = colorbar;
```

```
ylabel(cb,'distance to land (km)')
caxis([0 2000])
```



作者简介

这个函数和支持文档是由 Natalie S. Wolfenbarger 2019 年为 Climate Data Toolbox for Matlab 写的。

globeborders 文档

globeborders 函数在地球上绘制政区边界。

数据来自 2013 US Census Bureau 500k data（2013 年美国人口普查局 50 万数据）和 thematicmapping.org TM World Borders 0.3 数据集。

语法

```
globeborders
globeborders(LineSpec)
globeborders(Property,Value,...)
globeborders(...,'radius',GlobeRadius)
h = globeborders(...)
```

说明

globeborders 在定义为 6371 的地球上绘制文件 borderdata.mat 中定义的国界线，其中 6371 对应地球的平均半径（以公里为单位）。

globeborders(LineSpec) 指定边框的线条样式、标记符号和/或颜色。

globeborders(Property,Value,...) 指定用于控制边框外观和行为的线条属性。

globeborders(...,'radius',GlobeRadius) 将地球的半径指定为 GlobeRadius。

h = globeborders(...) 返回绘制对象的句柄 h。

示例 1

在地球上绘制国界线:

```
figure

globeborders

axis tight
```

用 globefill 填充空的地球：

```
hold on
globefill
axis tight
```

示例 2

在白色地球上的国界线绘制红色虚线：

```
figure

globefill

hold on
globeborders('--r')
axis tight
```

示例 3

将国界线绘制为半径 500 万的地球上的深蓝色圆点：

```
figure

globeborders('color',[0.25 0.25 0.65],'linewidth',2,'linestyle',':','radius',5e6);
axis tight
```

示例 4

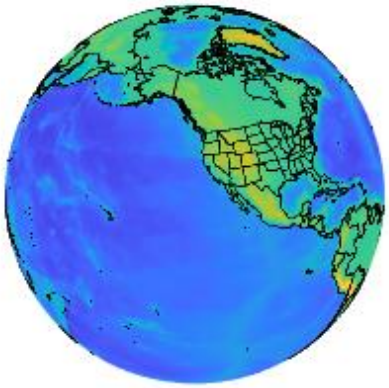
在蓝色大理石地球图上绘制政区边界：

```
figure

globeimage

globeborders

view(45,20)

axis tight
```

示例 5

将国界线绘制为描绘在地球地形上的细黑线：

```
load topo
[lon,lat] = meshgrid(0:359,-89:90);
figure
globepcolor(lat,lon,topo);
hold on
globeborders('color','k','linewidth',0.5,'linestyle','-')
```



作者简介

这个函数和支持文档是由 Natalie S. Wolfenbarger 2019 年为 Climate Data Toolbox for Matlab 写的。

globequiver 文档

globequiver 函数在地球上的东北坐标系中绘制带有分量(u,v) 的地理参考向量。

语法

```
globequiver(lat,lon,u,v)
globequiver(...,scale)
globequiver(...,LineSpec)
globequiver(...,LineSpec,'filled')
globequiver(...,'PropertyName',PropertyValue,...)
globequiver(...,'radius',GlobeRadius)
globequiver(...,'density',DensityValue)
h = globequiver(...)
```

说明

globequiver(lat,lon,u,v) 在地球上绘制地理参考矢量值 (u,v)。 输入 lat 和 lon 的大小与 u 和 v 相同，可以使用 meshgrid 或 cdtgrid 定义。

globequiver(...,scale) 自动缩放代表向量值 (u,v) 的箭头大小以适应地球，然后按照 S 将它们拉伸。

globequiver(...,LineSpec) 指定线型、标记符号和颜色。

globequiver(...,LineSpec,'filled') 填充 LineSpec 指定的标记。

globequiver(...,'PropertyName',PropertyValue,...) 为创建的对象指定属性名称和属性值对。

globequiver(...,'radius',GlobeRadius) 将地球的半径指定为 GlobeRadius。 默认 GlobeRadius 为 6371。

globequiver(...,'density',DensityValue) 指定地球上显示的矢量的密度。默认密度只是原始数据的密度。

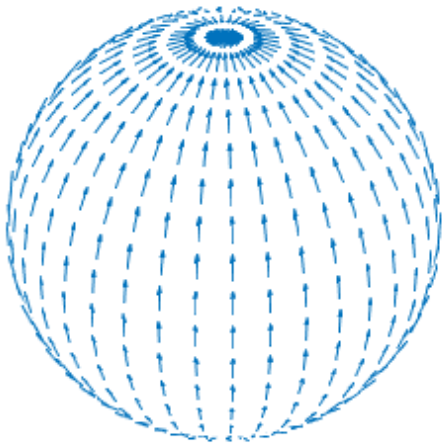
h = globequiver(...) 返回绘制对象的句柄 h 。

示例 1

绘制向北的向量，并使用 globefill 填充地球内部：

```
[lat,lon] = cdtgrid(10);

u = zeros(size(lat));

v = ones(size(lat));



figure

globequiver(lat,lon,u,v)

globefill
```

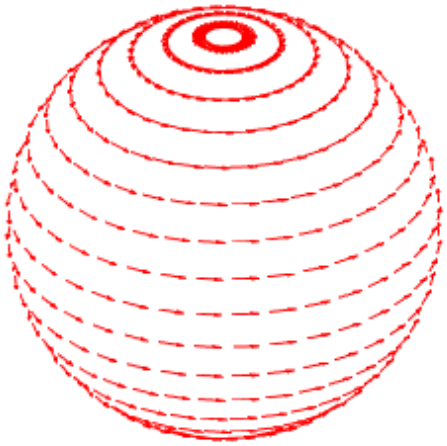
```
axis tight
```



示例 2

在地球上绘制红色向东向量：

```
[lat,lon] = cdtgrid(10);



u = ones(size(lat));

v = zeros(size(lat));



figure

globequiver(lat,lon,u,v,'r')
globefill
axis tight
```



示例 3

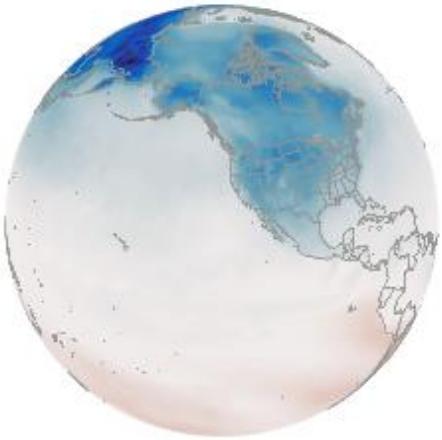
来自 ERA-Interim 产品的 2017 年 1 月的距平绘图：

```
filename = 'ERA_Interim_2017.nc';
u10 = ncread(filename,'u10'); % 10 米风东西分量
v10 = ncread(filename,'v10'); % 10 米风南北分量
sp = ncread(filename,'sp'); % 平面气压
T = ncread(filename,'t2m'); % 2 米温度
lat = double(ncread(filename,'latitude'));
lon = double(ncread(filename,'longitude'));

% 计算 1 月距平:
mo = 1;
Ta = T(:,:,mo)-mean(T,3);
spa = sp(:,:,mo)-mean(sp,3);
ua = u10(:,:,mo) - mean(u10,3);
va = v10(:,:,mo) - mean(v10,3);

[lat,lon] = meshgrid(lat,lon);
```

首先使用 globepcolor 绘制温度距平并使用 cmocean 调整颜色图；使用 globeborders 添加国界线：

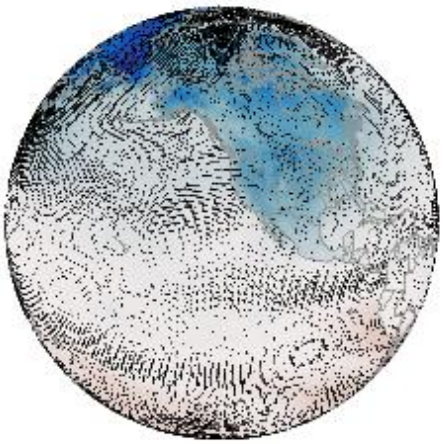
```
figure

globepcolor(lat,lon,Ta)
```

```
globeborders('color',rgb('gray'))
axis tight
cmocean('balance','pivot')
```



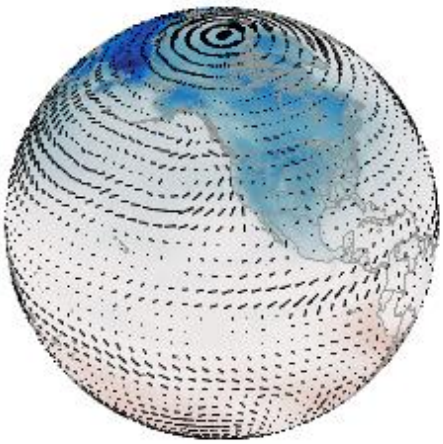
现在绘制风距平：

```
q = globequiver(lat,lon,ua,va,'k');
```



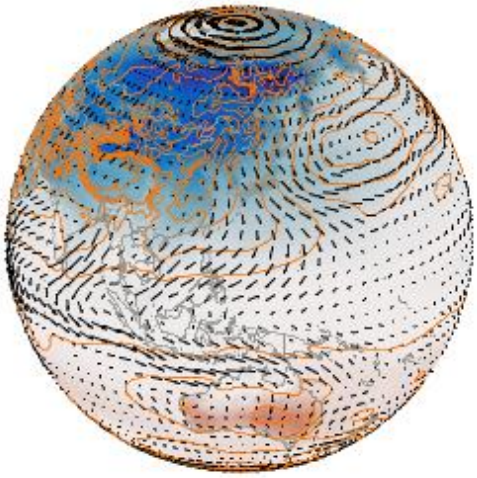
那个风矢量网格太密集了吗？ 删除它并重试，指定矢量密度和比例尺长度：

```
delete(q) % 删除我们刚刚在上面绘制的箭头

globequiver(lat,lon,ua,va,2,'density',100,'k')
```



为了增加内容，使用 globecontour 将表面气压距平绘制为等值线。还要改变视角：（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
globecontour(lat,lon,spa,10,'color',rgb('orange'))
view(-140,22)
```



作者简介

这个函数和支持文档是由 Natalie S. Wolfenbarger 和 Chad A. Greene 2019 年为 Climate Data Toolbox for Matlab 写的。

globestipple 文档

globestipple 函数在地球的一个区域上创建一个填充或点画。 此函数主要用于显示空间地图中具有统计意义的区域。

语法

```
globestipple(lat,lon,mask)
globestipple(...,MarkerProperty,MarkerValue,...)
globestipple(...,'radius',GlobeRadius)
globestipple(...,'density',DensityValue)
globestipple(...,'resize',false)
h = globestipple(...)
```

说明

globestipple(lat,lon,mask) 在地球上任何 mask 包含真值由 lat 和 lon 定义的位置绘制黑点。地球的半径为 6371，其中 6371 对应于地球的平均半径（以公里为单位）。lat,lon 和 mask 的尺寸必须全部匹配。

globestipple(...,MarkerProperty,MarkerValue,...) 指定绘图函数接受的任何标记属性(例如,'Color','Marker','MarkerSize', 等).

globestipple(...,'radius',GlobeRadius) 将地球的半径指定为 GlobeRadius。

globestipple(...,'density',DensityValue) 指定点画标记的密度。默认密度为 100，但如果您的绘图过于拥挤，您可以指定较低的密度值（和/或调整'MarkerSize'）。

globestipple(...,'resize',false) 覆盖 'density' 选项并以输入网格的精确分辨率绘制点画。默认情况下，网格会被调整大小，因为任何大于约 100x100 的网格都会产生如此多的点画点，以至于它们下面的任何东西都会变黑。

h = globestipple(...) 返回绘制的点画对象的句柄。

示例 1

在地形高于海平面的地方绘制高密度点画。为此，我们将使用 cdtgrid 创建一个 1 度全球网格，并使用 island 确定哪些网格单元对应于陆地：

```
[lat,lon] = cdtgrid;

mask = island(lat,lon);



figure

globestipple(lat,lon,mask,'density',300)
axis tight

% 将地球颜色设置为天蓝色:
globefill('color',rgb('azure'))
```

示例 2

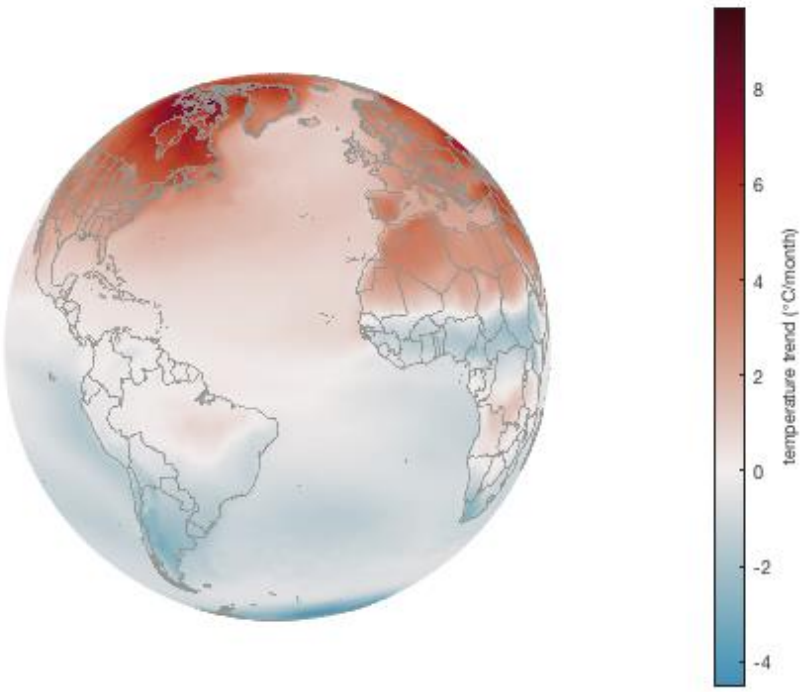
地球上的大多数地方都倾向于在 2 月和 8 月之间改变温度。 加载 2017 年的每月全球温度网格数据示例，我们将找出这些温度趋势的显著位置。

```
filename = 'ERA_Interim_2017.nc';
T = ncread(filename,'t2m');
lat = double(ncread(filename,'latitude'));
lon = double(ncread(filename,'longitude'));

% 获取经纬度数组:
[Lat,Lon] = meshgrid(lat,lon);

% 计算 2 月到 8 月的温度趋势:
[tr,p] = trend(T(:,:,2:8));
```

首先，使用 globepcolor 绘制温度趋势并使用 cmocean 设置颜色图：

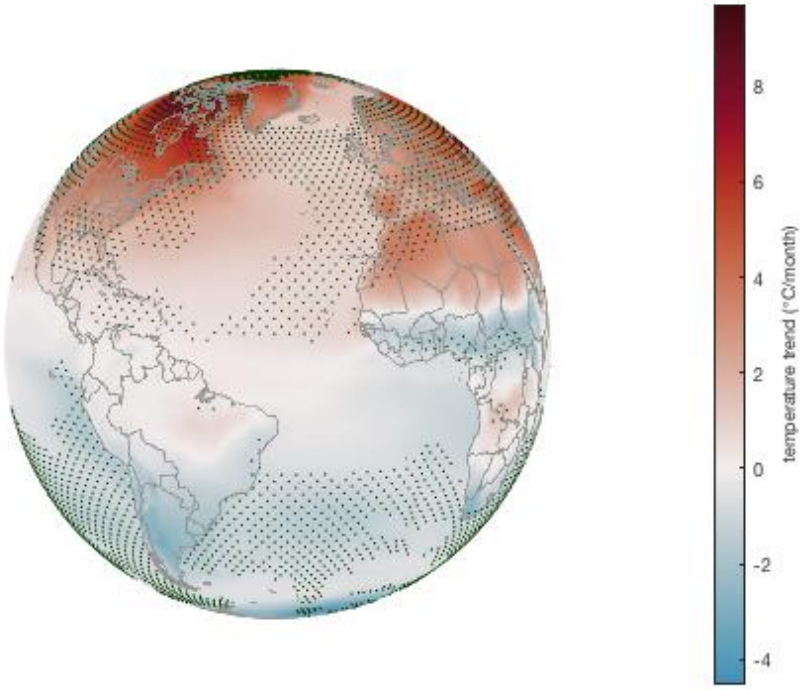
```
figure

globepcolor(Lat,Lon,tr)

axis tight
cmocean('balance','pivot')
cb = colorbar;
```

```
ylabel(cb,'temperature trend (\circC/month)')
globeborders('color',rgb('gray')) % 绘制政区边界
view(55,10) % 设置视角
```



不出所料，北半球从 2 月到 8 月趋于升温，而南半球则相反。但趋势在哪里显著？
在趋势显著至 p<0.001 的地方添加点画

```
globestipple(Lat,Lon,p<0.001,...
   'density',250,...
   'color',rgb('dark green'),...
   'markersize',2)
```



作者简介

这个函数和支持文档是由 Natalie S. Wolfenbarger 和 Chad A. Greene 2019 年为 Climate Data Toolbox for Matlab 写的。

globegraticule 文档

globegraticule 函数绘制一个经纬网球体。可选输入控制经纬网的外观和行为。

语法

```
globegraticule
globegraticule(lat,lon)
globegraticule(...,LineProperty,LineValue)
globegraticule(...,'radius',GlobeRadius)
h = globegraticule(...)
```

说明

globegraticule 在空地球上绘制一个经纬网或网格。

globegraticule(lat,lon) 指定经纬网描绘的纬度和经度。

globegraticule(...,LineProperty,LineValue) 使用用户指定的线属性绘制经纬网，以控制外观和行为。

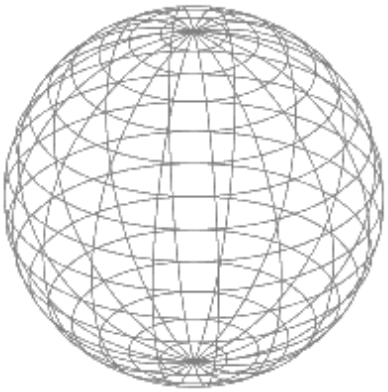
globegraticule(...,'radius',GlobeRadius) 将经纬网的半径指定为 GlobeRadius。 默认 GlobeRadius 为 6371。

h = globegraticule(...) 返回绘制对象的句柄 h。

示例 1

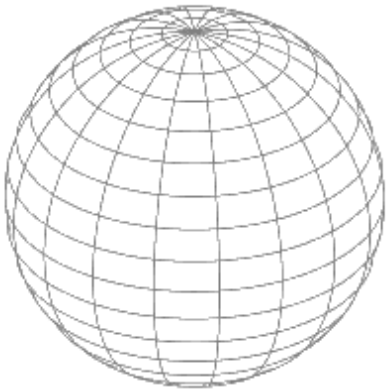
在地球上绘制一个简单的灰色经纬网：

```
figure


globegraticule
```



上面的地球是一个空的线框。用 globefill 填充它：

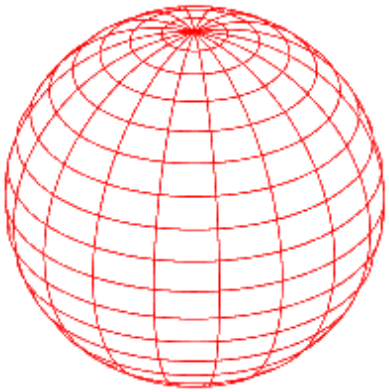
```
globefill
```



示例 2

绘制红色经纬网：

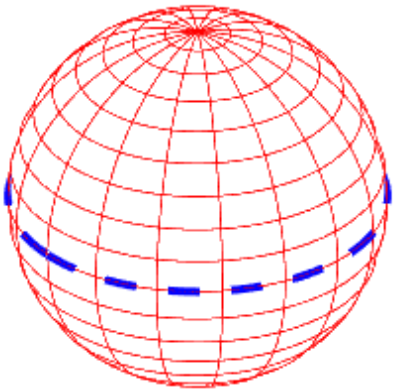
```
figure

globegraticule('color','r')
globefill
```



示例 3

沿赤道添加一条粗蓝色虚线：

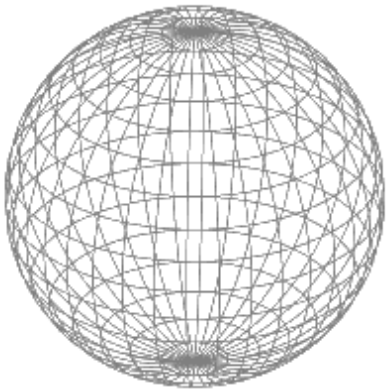
```
globegraticule(0,[],'color','b','linestyle','--','linewidth',4)
```



示例 4

用 10° ×10° 纬度和经度绘制经纬网：

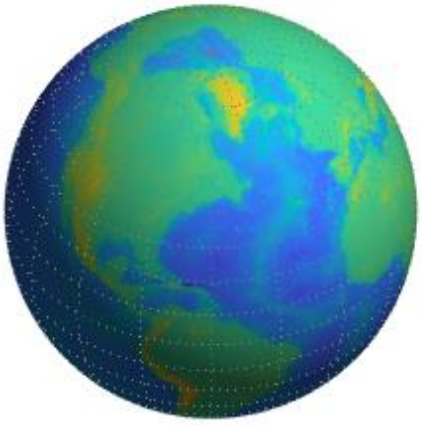
```
figure

globegraticule(-90:10:90,0:10:360)
```



示例 5

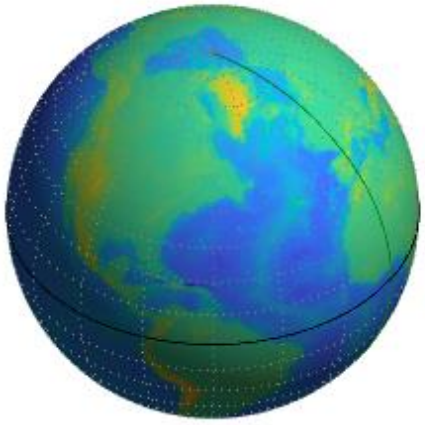
在其他数据之上绘制经纬网并指定查看几何参数：P

```
[lat,lon] = cdtgrid;

z = topo_interp(lat,lon);



figure

globepcolor(lat,lon,z)

hold on
axis tight
view(30,40) % 定义视角（方位角，仰角）

globegraticule('linestyle',':')

camlight % 调整亮度
material dull % 调整反射率
```



绘制赤道和本初子午线：

```
globegraticule(0,[],'color','k','linewidth',1) % 赤道
globegraticule([],0,'color',[0 0.5 0],'linewidth',1) % 本初子午线
```



作者简介

这个函数和支持文档是由 Natalie S. Wolfenbarger 2019 年为 Climate Data Toolbox for Matlab 写的。

globefill 文档

globefill 函数绘制一个填充的地球。可选输入控制地球的外观和行为。

语法

```
globefill
globefill(...,'color',ColorSpec)
globefill(...,'radius',GlobeRadius)
h = globefill(...)
```

说明

globefill 创建一个白色地球，可用于填充诸如 globeplot 之类的图的空白空间。

globefill(...,'color',ColorSpec) 指定填充地球的颜色。

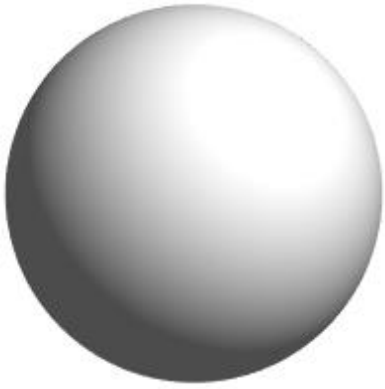
globefill(...,'radius',Radius) 将球体的半径指定为 Radius。 默认情况下，半径为 6307.3，这是标准地球 6371 公里地球半径的 99%。 地球填充半径略小于其他 globe* 函数使用的标准半径，以防止不必要的交互。

h = globefill(...) 返回绘制对象的句柄 h。

示例 1

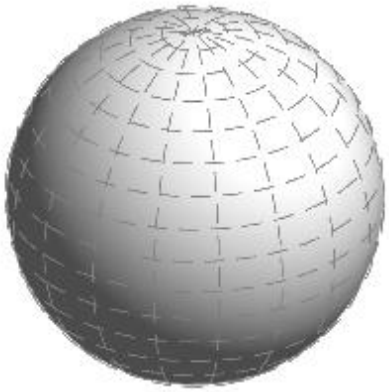
绘制一个简单的白色地球：

```
figure

globefill



camlight

material dull
```



（对于上图，我们必须打开 camlight，因为如果没有华丽的照明，白色背景上的白色地球将不可见。）
添加刻度：

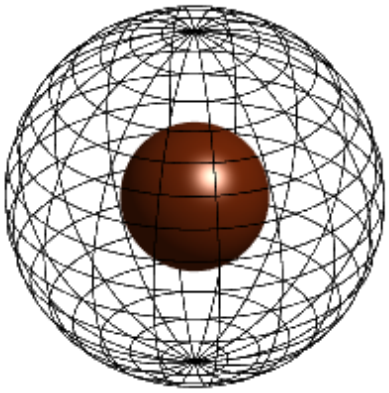
```
globegraticule('linestyle','--')
```



示例 2

绘制半径为 2500 公里的红棕色地球，然后在默认地球半径处添加黑色 globegraticule：

```
figure

globefill('color',rgb('reddish brown'),'radius',2500)
globegraticule('color','k')
camlight
```



作者简介

这个函数和支持文档是由 Natalie S. Wolfenbarger 2019 年为 Climate Data Toolbox for Matlab 写的。

NetCDF 和 HDF5

查看 NetCDF 教程以获取 NetCDF 数据入门帮助。

- ncstruct 从 netcdf 文件中读取多个变量。
- ncdateread 从 netcdf 时间变量中读取日期时间。
- ncdatelim 从 netcdf 文件中读取第一个和最后一个日期。
- h5struct 将 HDF5 数据加载到与原始文件具有相同层次结构的 Matlab 结构体中。

ncstruct 文档

ncstruct 函数提供了一种简单的语法，可以从 netCDF 文件中读取一个或多个变量，无论是完整的还是较小的 Hyperslab。

语法

```
Data = ncstruct(file)
Data = ncstruct(file, var1, var2, ...)
Data = ncstruct(file, Scs, ...)
```

说明

Data = ncstruct(file) 从指定的 netCDF 文件中读取所有变量数据到一个结构体 Data 中；Data 的字段名称对应于文件中的变量名称。

Data = ncstruct(file, var1, var2, ...) 只读取指定的变量数据。特殊字符串 'dimensions' 表示应该读入所有维度变量；维度变量是名称和大小对应于文件维度之一的任何变量。

Data = ncstruct(file, Scs, ...) 使用结构体 Scs 中的维度 hyperslabs 读取数据。 Scs 中的字段名要与文件中的维度名匹配，每个字段保存一个 1×3 的[start count stride],数组，表示数据应该从索引 start 开始沿指定维度读取，读取 count 个元素，用 stride 元素间间距。

示例 1: 从单个文件中读取变量

可以完整读取一个小的简单文件：

```
A = ncstruct('example.nc')
```

```
A =

  struct with fields:

    avagadros_number: 6.0221e+23
         temperature: [50×1 double]
               peaks: [50×50 int16]
```

对于较大的文件，您可能不想或不需要读入所有变量。 在这里，我们只读取风速变量以及维度变量：

```
B = ncstruct('ERA_Interim_2017.nc', 'u10', 'v10', 'dimensions')
```

```
B =
```

```
  struct with fields:

     latitude: [241×1 single]
    longitude: [480×1 single]
         time: [12×1 int32]
          u10: [480×241×12 double]
          v10: [480×241×12 double]
```

示例 2: 子集维度

ncread 函数允许您读取变量的 hyperslab； 但是，它的语法要求您以正确的顺序为该变量的所有维度提供 start、count 和 stride 参数； 为要读取的每个变量确定维度名称和维度顺序可能会变得乏味。 ncstruct 函数通过为您确定哪些变量包括特定维度以及按什么顺序来简化。

回到 ERA Interim 2017 数据集，我们可以从所有变量中读取数据，但只能在一个位置读取数据，并且每隔一次：

```
Scs = struct('latitude', [1 1 1], 'longitude', [1 1 1], 'time', [1 Inf 2]);
C = ncstruct('ERA_Interim_2017.nc', Scs)
```

```
C =

  struct with fields:

    longitude: 0
     latitude: 90
         time: [6×1 int32]
           sp: [1×1×6 double]
          u10: [1×1×6 double]
          v10: [1×1×6 double]
```

```
      t2m: [1×1×6 double]

       tp: [1×1×6 double]
```

示例 3: 从多个文件中读取

多文件数据集在气候模型输出或其他长期运行的数据集中很常见，其中数据被分成每天/每月/每年/等的单独文件。 保持文件大小可管理，并允许数据集随着时间的推移而增长，而无需编辑现有文件。 以下一组文件各自包含来自前面示例中使用的 ERA Interim 数据集的一个时间片：

```
files = {...
    'ERA_Interim_2017a.nc'
    'ERA_Interim_2017b.nc'
    'ERA_Interim_2017c.nc'};
```

我们可以使用与单个文件相同的语法从这组文件中读取数据。 跨文件重复的变量，例如维度变量，只会被读取一次，而包含时间维度的变量将从每个文件中读取并连接。

```
D = ncstruct(files, 'u10', 'v10', 'dimensions')
```

```
D =

  struct with fields:

     latitude: [241×1 single]
    longitude: [480×1 single]
         time: [3×1 int32]
          u10: [480×241×3 double]
          v10: [480×241×3 double]
```

作者简介

这个函数和支持文档是华盛顿大学的 Kelly A. Kearney 在 2019 年为 Climate Data Toolbox for Matlab 写的。它是这个工具箱的一部分，也可以从 GitHub 单独下载。

ncdateread 文档

此函数从 netCDF 文件中将时间变量读入日期时间数组，假设该变量符合 CF 标准并具有“*自参考时间以来的时间单位*”（'*time units* since *reference time*'）单位属性。T

语法

```
tdt = ncdateread(file, var)
[tdt, tnum, unit, refdate] = ncdateread(file, var)
```

说明

tdt = ncdateread(file, var)将 netCDF 文件（或多个文件）file 中的时间变量 var 中的值读入日期时间数组 tdt，假设变量使用标准（即公历）日历并包含一个单位属性，形式为'自参考时间以来的时间单位'（'*time units* since *reference time*'）。支持的时间单位是微秒、毫秒、秒、小时、分钟和天（或天）。 file 可以是包含单个文件路径的字符串/字符数组，也可以是包含共享单个无限时间维度的一组文件的字符串元胞数组。

[tdt, tnum, unit, refdate] = ncdateread(file, var) 还返回未转换形式的原始文件时间值 tnum 、unit 字符数组以及与单位属性中的 refdate 匹配的日期时间引用日期。

[...] = ncdateread(file, var, tnum) 提供了一个选项，可以使用文件和变量名称指示的相同单位属性转换变量中的数字时间值。

注意：此处有意不包括年或月单位的选项，因为与这些单位相关的转换可能不正确。 CF 标准将一年定义为 365.242198781 天，而不是一个日历年；同样，一个月是这个值的 1/12。这很少是一种有用的计算时间的方法，因此我们将其留给用户来确定使用这些单位的文件是打算使用日历年解释还是严格解释，并自己手动解析日期。

示例

示例 ERA 临时文件的时间属性遵循典型的气候数据标准：

```
ncdisp('ERA_Interim_2017.nc', 'time');
```

```
Source:

           /Users/kakearney/Documents/MATLAB/Add-Ons/Toolboxes/Climate Data
Toolbox/code/cdt_data/ERA_Interim_2017.nc

Format:

           64bit

Dimensions:

           time = 12     (UNLIMITED)

Variables:

    time
```

```
          Size:       12x1

          Dimensions: time

          Datatype:   int32

          Attributes:

                      units     = 'hours since 1900-01-01 00:00:00.0'

                      long_name = 'time'

                      calendar  = 'gregorian'
```

在这里，我们可以直接从文件中读取时间数据，而无需提前知道单位或参考日期是什么：

```
[tdt, tnum, unit, refdate] = ncdateread('ERA_Interim_2017.nc', 'time')
```

```
tdt =


  12×1 datetime array

   2017-01-01

   2017-02-01

   2017-03-01

   2017-04-01

   2017-05-01

   2017-06-01

   2017-07-01

   2017-08-01

   2017-09-01

   2017-10-01
```

2017-11-01

2017-12-01

tnum =

12×1 int32 column vector

1025628

1026372

1027044

1027788

1028508

1029252

1029972

1030716

1031460

1032180

1032924

1033644

unit =

```
    'hours'
```

```
refdate =
```

```
  datetime
```

```
   1900-01-01
```

如果时间变量已经在我们的工作区中（例如，如果使用 ncreads（译者注此函数已找不到相关说明）中提供的一些更灵活的 hyperslab 子集选项读取数据，我们可以通过指向具有必要转换的文件来转换这些单位数据：

```
A = ncreads('ERA_Interim_2017.nc', struct('time', [1 5 2]));
A.time
```

```
ans =
```

```
  5×1 int32 column vector
```

```
   1025628
   1027044
   1028508
   1029972
   1031460
```

```
A.time = ncdateread('ERA_Interim_2017.nc', 'time', A.time);
A.time
```

```
ans =


  5×1 datetime array

   2017-01-01
   2017-03-01
   2017-05-01
   2017-07-01
   2017-09-01
```

作者简介

这个函数和支持文档是华盛顿大学的 Kelly A. Kearney 在 2019 年为 Climate Data Toolbox for Matlab 写的。它是这个工具箱的一部分，也可以从 GitHub 单独下载。

ncdatelim 文档

此函数从一个或多个 netCDF 文件中的时间变量中读取第一个和最后一个时间值，并将这些值转换为日期时间。

语法

```
tlim = ncdatelim(files, var)
```

说明

`tlim = ncdatelim(files, var)` 从字符串/字符数组或字符串 `files` 元胞数组指示的一个或多个文件中读取数据。 它读取时间变量 `var` 的第一个和最后一个值，假设该变量使用标准（即公历）日历并包含 CF 标准单位属性（有关更多详细信息，请参阅 `ncdateread`）。 返回的 `tlim` 变量是一个 `nx2` 日期时间数组，其中 `n` 是查询的文件数。

示例

读取此工具箱中包含的每个示例 netCDF 文件中的日期限制：

```
cdtpath = fileparts(which('cdt'));

files = {...
    fullfile(cdtpath, 'cdt_data', 'ERA_Interim_2017.nc')
    fullfile(cdtpath, 'cdt_data', 'ERA_Interim_2017a.nc')
    fullfile(cdtpath, 'cdt_data', 'ERA_Interim_2017b.nc')
    fullfile(cdtpath, 'cdt_data', 'ERA_Interim_2017c.nc')};

tlim = ncdatelim(files, 'time')
```

```
tlim =

  4×2 datetime array

   2017-01-01   2017-12-01

   2017-02-01   2017-02-01

   2017-03-01   2017-03-01

   2017-04-01   2017-04-01
```

作者简介

这个函数和支持文档是华盛顿大学的 Kelly A. Kearney 在 2019 年为 Climate Data Toolbox for Matlab 写的。它是这个工具箱的一部分，也可以从 GitHub 单独下载。

hfive2struct 文档（h5struct）

h5struct 将 HDF5 数据加载到与原始文件具有相同层次结构的 Matlab 结构体中。

语法

```
data = hfive2struct(filename)
data = hfive2struct(filename, dataset)
data = hfive2struct(filename, dataset, true)
data = hfive2struct(filename, [], true)
```

说明

data = hfive2struct(filename) 将 h5 文件名的所有属性、数据集值和数据集属性加载到与原始文件具有相同层次结构的 Matlab 结构体中。 [当前最多可处理 11 个组级别]
data = hfive2struct(filename, dataset) 仅将请求的数据集和文件名的关联属性加载到与原始文件名具有相同层次结构的 Matlab 结构体中。dataset 是所需数据集的完整 h5 路径。 数据集可以指定为字符串或字符串的元胞数组，例如 /path/to/dataset 或 {path/to/dataset1, path/to/dataset2}。
data = hfive2struct(filename, dataset, fill_with_nan)
仅将请求的数据集和文件名的关联属性加载到 Matlab 结构体中。 如果 fill_with_nan = true，则所有具有 _FillValue 的单数据集和双数据集都将 _FillValue 替换为 NaN。

用户输入

- filename hdf5 文件的完整路径
- dataset hdf5 结构中的数据集完整路径（可选）。 可以指定为字符串或字符串元胞数组，例如'/orbit_info/rgt' 或 {'/orbit_info/rgt',/ancillary_data/control'}。
- fill_with_nan true 或 false，如果 true，带有 _FillValue 的单双数据集将用 NaN 填充 _FillValue 。

示例 1

从 h5 文件加载所有数据：

```
D = h5struct('altimetry_example.h5');
```

现在这里是 D 的结构体内容

```
D
```

```
D =

  struct with fields:

             elevation: [1×1 struct]
```

```
                latitude: [1×1 struct]
               longitude: [1×1 struct]
              Attributes: [1×1 struct]
          ancillary_data: [1×1 struct]
    instrument_parameters: [1×1 struct]
```

要访问 D 的字段之一，请在 D 后面加上 . 然后是字段名称。 例如，要访问高程数据，请执行以下操作：

```
D.elevation
```

```
ans =

  struct with fields:


        Value: [236379×1 single]
    Attributes: [1×1 struct]
```

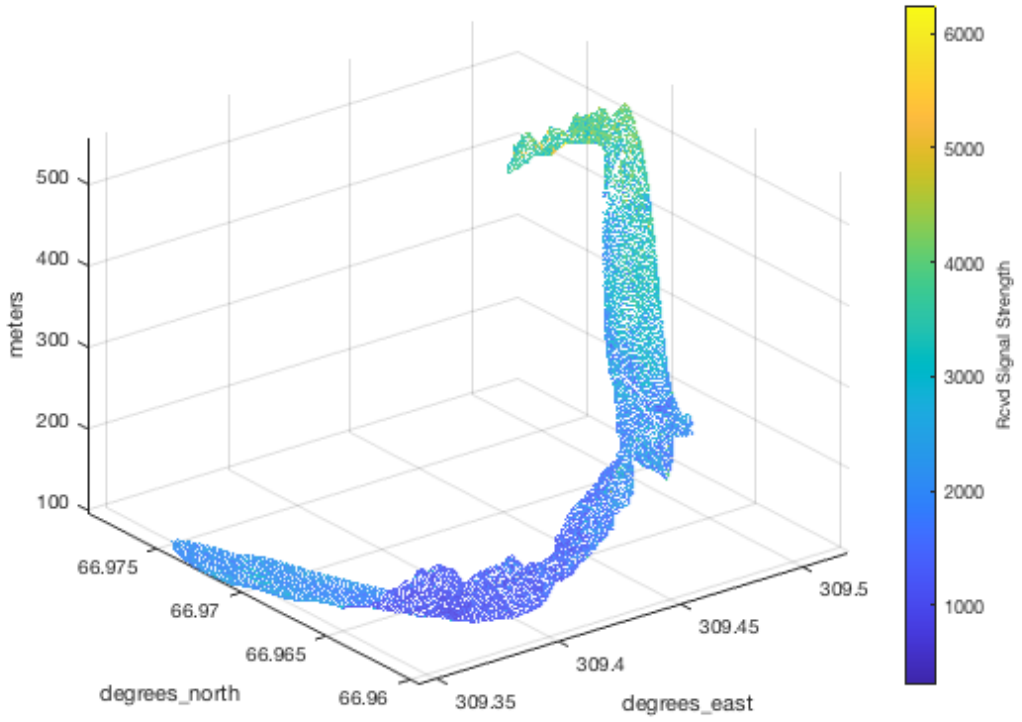
高程测量值位于 D.elevation.Value 字段中，有关高程数据的信息可以在 D.elevation.Attributes 中找到。 以下是高程数据的属性：

```
D.elevation.Attributes
```

```
ans =

  struct with fields:


         h5es_id: 4
           units: 'meters'
       long_name: 'elevation'
    standard_name: 'elevation'
      description: 'Elevation of the laser spot above ellipsoid'
```

将海拔和信号强度绘制三维散点图：

```
figure

scatter3(D.longitude.Value, D.latitude.Value, D.elevation.Value,...
   2, D.instrument_parameters.rcv_sigstr.Value,'filled')

% 增加网格和一些标签:
grid on
xlabel(D.longitude.Attributes.units, 'interpreter', 'none');
ylabel(D.latitude.Attributes.units, 'interpreter', 'none');
zlabel(D.elevation.Attributes.units, 'interpreter', 'none');

cb = colorbar;
ylabel(cb,D.instrument_parameters.rcv_sigstr.Attributes.long_name)
axis tight
```



示例 2

如果您只需要 HDF5 文件中的部分数据，您可能更愿意只加载数据，方法如下：

```
dataset = {'/instrument_parameters/rel_time', '/elevation'};
D = h5struct('altimetry_example.h5', dataset);
```

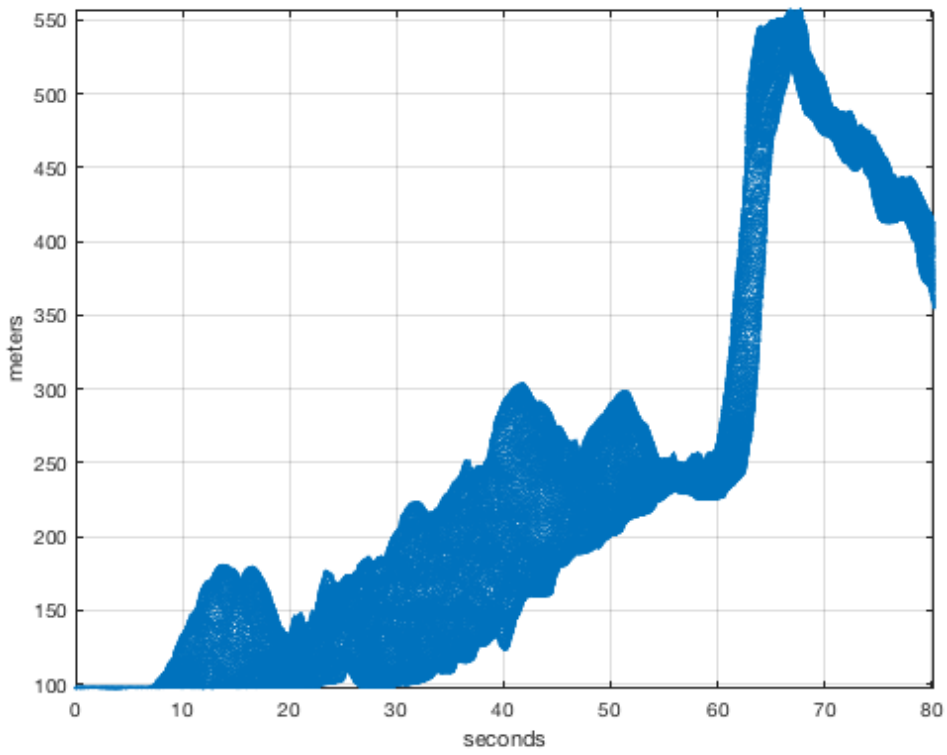
这次我们只加载了高程数据和相对时间。 这是 D 的样子：

```
D
```

```
D =

  struct with fields:


    instrument_parameters: [1×1 struct]

                elevation: [1×1 struct]
```

与示例 1 类似，绘制海拔与相对时间的关系图

```
figure

plot(D.instrument_parameters.rel_time.Value, D.elevation.Value,'.','markersize',0.5)

% 增加网格和一些标签
grid on
xlabel(D.instrument_parameters.rel_time.Attributes.units, 'interpreter', 'none');
ylabel(D.elevation.Attributes.units, 'interpreter', 'none');
axis tight
```



作者简介

这个函数和支持文档是由 JPL-Caltech 的 Alex S. Gardner 于 2018 年 10 月写的。

教程

CDT 包括一些教程，用于解决我们在 Matlab 中分析气候数据时遇到的一些常见问题。

- **NetCDF data** 描述了如何加载和分析 NetCDF 数据。
- **Dates and times** 描述了各种 Matlab 日期格式，以及如何将它们与地球科学数据一起使用。
- **Linear trends** 解释了如何将最小二乘回归应用于气候时间序列。
- **Mapmaking** 描述了在 Matlab 中制作气候数据地图的几种不同方法。

使用 NetCDF 数据

NetCDF 是气候科学中最常见的数据格式之一，在 Matlab 中使用它非常简单，但了解一些技巧会有所帮助。下面是一个分步指南，用于在 Matlab 中读取和分析 NetCDF 数据。

下载数据

本教程中使用的示例数据是作为 CDT 的一部分提供的，因此您无需下载任何内容。但是获取数据通常是第一步，所以为了后续，这里是我下载示例数据的方式：

1.登录 ECMWF 网站，找到 ERA-Interim Synoptic Monthly Means。

2.选择时间 00:00:00，12 步（00:00:00,12step），以及 2m 温度、10mU 和 V 风分量、表面气压和总降水量的字段。

3.选择 NetCDF 数据格式并以默认 0.75x0.75 度分辨率下载。

4.等待约 15 分钟数据准备就绪并下载。

5.从 ECMWF 下载的数据的默认文件名通常很长而且很神秘，因此在本教程中，我将其重命名为 ERA_Interim_2017.nc。

下载数据后，将其放在当前文件夹或 Matlab 可以找到的其他位置。考虑为数据创建一个文件夹，并通过 addpath 将该文件夹添加到 Matlab 的搜索路径。

检查 NetCDF 文件的内容

加载 NetCDF 数据的第一步是找出文件中的变量名称。为此，请像这样使用 ncdisp：

```
ncdisp 'ERA_Interim_2017.nc'
```

```
Source:

           /home/chad/Documents/MATLAB/github/cdt/cdt_data/ERA_Interim_2017.nc

Format:

           64bit

Global Attributes:

           Conventions = 'CF-1.6'

           history     = '2018-12-06 18:53:30 GMT by grib_to_netcdf-2.9.2: grib_to_netcdf
/data/data01/scratch/78/bb/_mars-atls17-98f536083ae965b31b0d04811be6f4c6-L3jQQq.grib -o
/data/data03/scratch/77/57/_grib2netcdf-atls19-98f536083ae965b31b0d04811be6f4c6-5zbphu.nc -
utime'

Dimensions:

           longitude = 480

           latitude  = 241
```

```
            time      = 12    (UNLIMITED)

Variables:

    longitude

            Size:       480x1

            Dimensions: longitude

            Datatype:   single

            Attributes:

                        units     = 'degrees_east'

                        long_name = 'longitude'

    latitude

            Size:       241x1

            Dimensions: latitude

            Datatype:   single

            Attributes:

                        units     = 'degrees_north'

                        long_name = 'latitude'

    time

            Size:       12x1

            Dimensions: time

            Datatype:   int32

            Attributes:

                        units     = 'hours since 1900-01-01 00:00:00.0'

                        long_name = 'time'
```

calendar = 'gregorian'

sp

Size: 480x241x12

Dimensions: longitude,latitude,time

Datatype: int16

Attributes:

scale_factor = 0.79238

add_offset   = 77621.389

_FillValue   = -32767

missing_value = -32767

units   = 'Pa'

long_name   = 'Surface pressure'

standard_name = 'surface_air_pressure'

u10

Size: 480x241x12

Dimensions: longitude,latitude,time

Datatype: int16

Attributes:

scale_factor = 0.00041182

add_offset   = -0.58524

_FillValue   = -32767

missing_value = -32767

units   = 'm s**-1'

long_name = '10 metre U wind component'

v10

Size: 480x241x12

Dimensions: longitude,latitude,time

Datatype: int16

Attributes:

scale_factor = 0.00041589

add_offset = 1.7302

_FillValue = -32767

missing_value = -32767

units = 'm s**-1'

long_name = '10 metre V wind component'

t2m

Size: 480x241x12

Dimensions: longitude,latitude,time

Datatype: int16

Attributes:

scale_factor = 0.0017972

add_offset = 263.9411

_FillValue = -32767

missing_value = -32767

units = 'K'

long_name = '2 metre temperature'

```
tp

        Size:       480x241x12

        Dimensions: longitude,latitude,time

        Datatype:   int16

        Attributes:

                    scale_factor  = 6.832e-07

                    add_offset    = 0.022386

                    _FillValue    = -32767

                    missing_value = -32767

                    units         = 'm'

                    long_name     = 'Total precipitation'
```

ncdisp 的输出告诉我们我们需要知道的一切。 最重要的是，它告诉我们 NetCDF 文件中变量的名称，还包括关于每个变量单位的注释。
注意每个变量的 Size 。 这通常是了解数据矩阵如何组织的第一个线索。 经度变量为 480x1，纬度为 241x1，时间为 12x1。 其他变量为 480x241x12 并非巧合：这些维度对应于经度、纬度和时间。
一些变量包括 scale_factor,add_offset,和_FillValue。 我们不需要关注这些值，因为 Matlab 的 ncread 函数会自动进行缩放和偏移，但如果您好奇，这些值就是算法用来将数据有效地打包（和解包）为 NetCDF 格式的值。

读取数据

现在我们知道 ERA_Interim_2017.nc 中变量的名称，我们可以加载我们感兴趣的变量。 让我们看一下在 2 m ('t2m') 处所取的气温，并得到其对应的地理坐标：

```
lat = ncread('ERA_Interim_2017.nc','latitude');
lon = ncread('ERA_Interim_2017.nc','longitude');
T = ncread('ERA_Interim_2017.nc','t2m');
```

读取时间数组

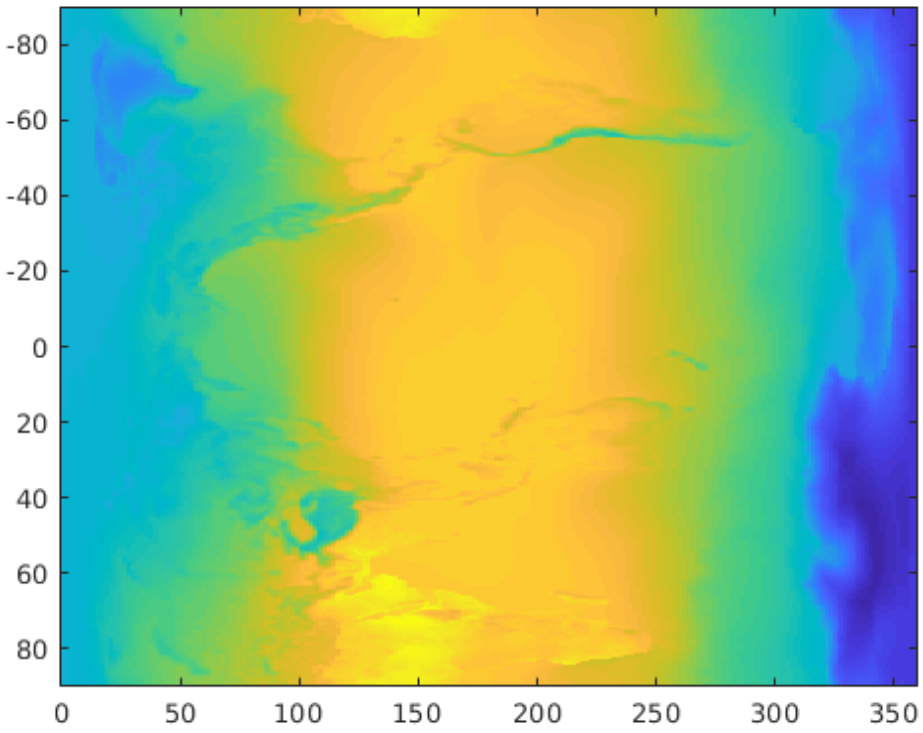
从 NetCDF 文件读取时间数组的最简单方法是使用 CDT 函数 ncdateread。 它的工作方式与 ncread 一样，但会自动将时间数组转换为日期时间格式（在此处阅读有关时间格式的更多信息）。 以下是如何使用 ncdateread 加载时间：

```
t = ncdateread('ERA_Interim_2017.nc','time');
```

绘制数据:

本教程描述了制作精美地图的几个技巧，但现在让我们只使用 imagesc 函数来检查网格方向。这意味着我们让 lon 是 x 变量，让 lat 是 y 变量。
对于 T 变量，我们知道第三维对应于时间，因此沿第三维取 T 的平均值即可得到 2017 年的平均温度图：

```
% 计算 2017 平均温度:
Tm = mean(T,3);

% 绘制平均温度:
imagesc(lon,lat,Tm)
```

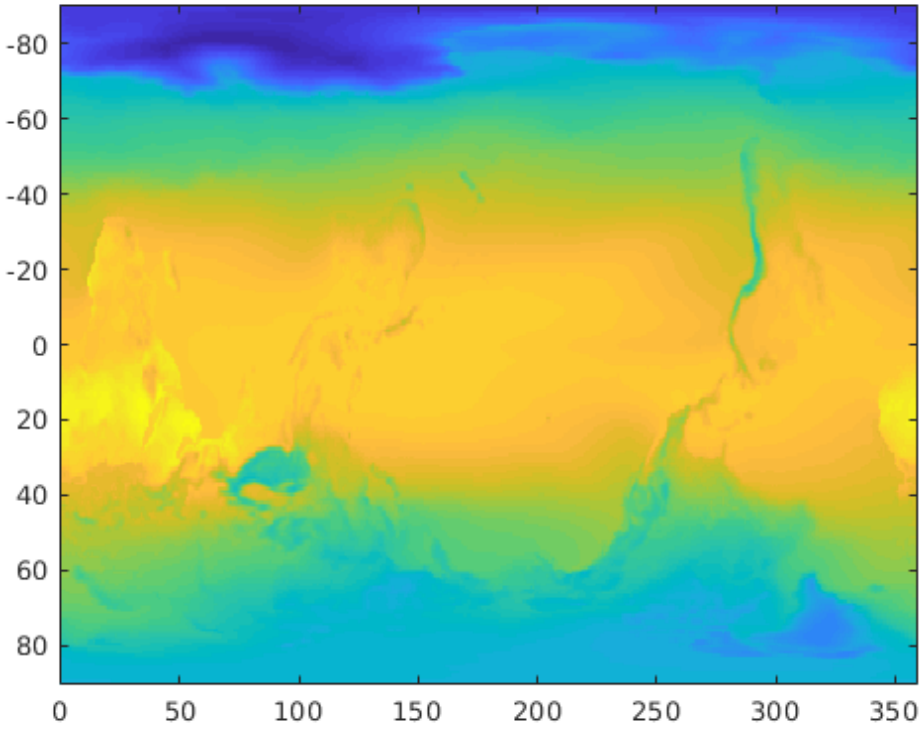


旋转和颠倒网格

上图中的网格已旋转！ 这在导入 NetCDF 和其他数据格式时并不少见，它们通常使用第一个维度作为“x”维度，第二个维度作为“y”维度。 解决这个问题的一种方法是用 rot90 旋转矩阵，然后用 flipud 垂直翻转它：

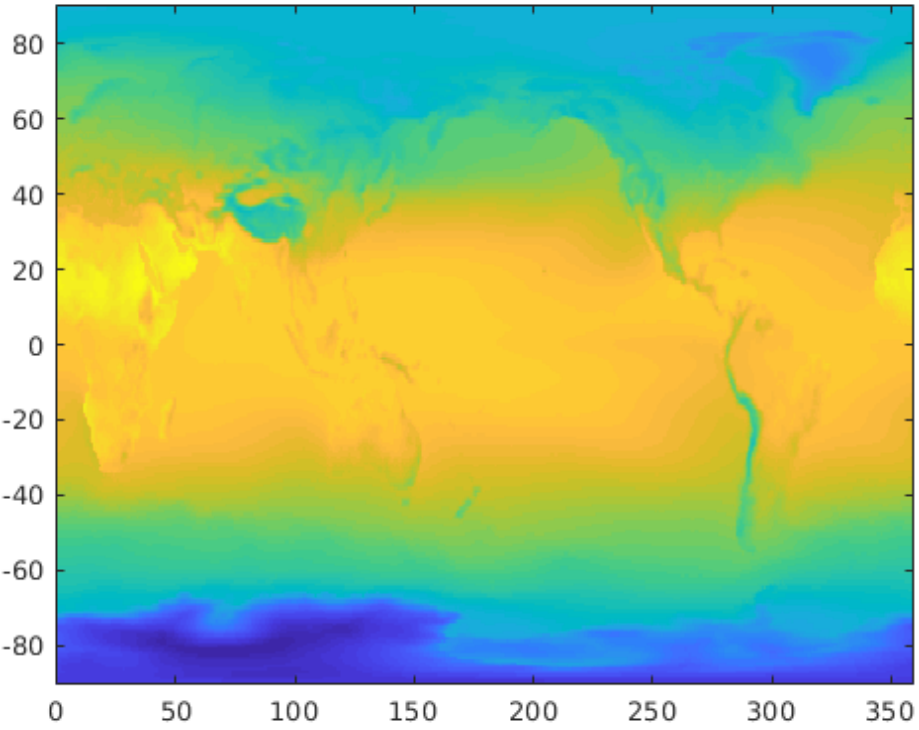
```
% 旋转并颠倒 T:
T = flipud(rot90(T));

% 使用旋转、翻转的数据矩阵重新计算 Tm:
Tm = mean(T,3);

% 重新绘制平均温度:
```

```
imagesc(lon,lat,Tm);
```



现在它旋转了，但它仍然是颠倒的。 那我们为什么要翻转它？ 由于 imagesc 的特殊性，每当您指定 x,y 坐标时，都需要使用 axis xy 命令：

```
axis xy
```



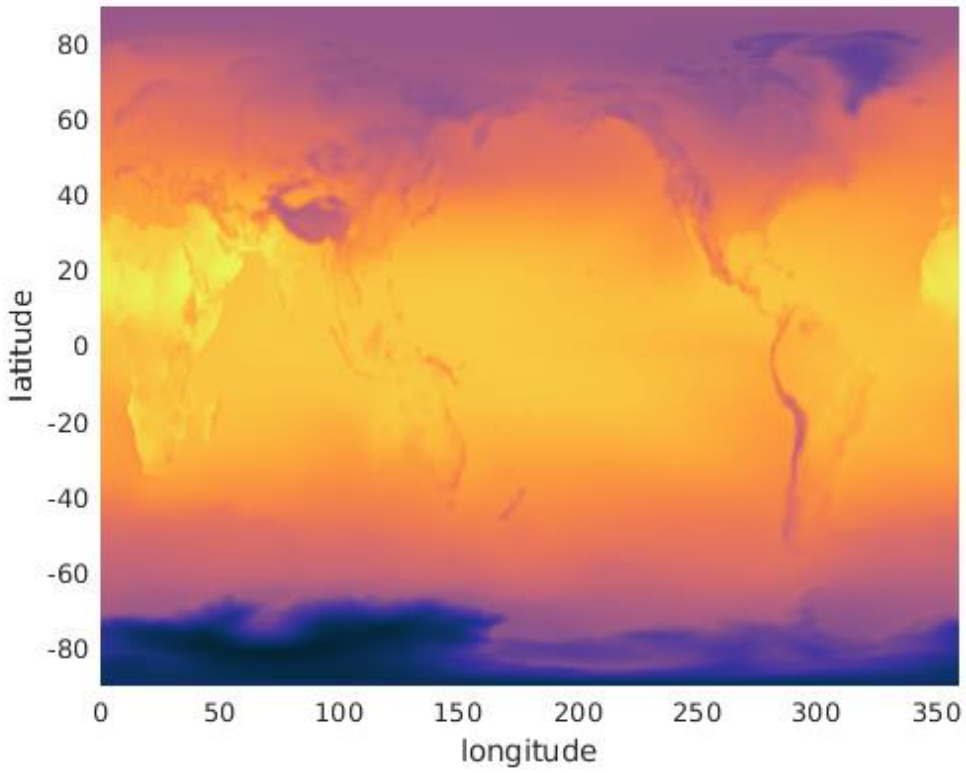
获取每个网格单元的坐标

到目前为止，我们一直使用 lat 和 lon 作为向量，简单的假设是 lat 中的每个值对应于 T 的行，而 lon 中的每个值对应于 T 的列。但有时您需要 lat 和 lon 完整的网格。 为此，请使用 meshgrid。

```
[Lon,Lat] = meshgrid(lon,lat);
```

使用 meshgrid 时请注意 lat 和 lon 的顺序。如果我们之前没有旋转网格的尺寸，我们将使用 [Lat,Lon]=meshgrid(lat,lon) 代替。
对于坐标的完整网格而不是一维数组，请使用 pcolor 而不是 imagesc。此外，pcolor 很挑剔，并希望坐标为双精度，因此请按照您在此处看到的进行转换：

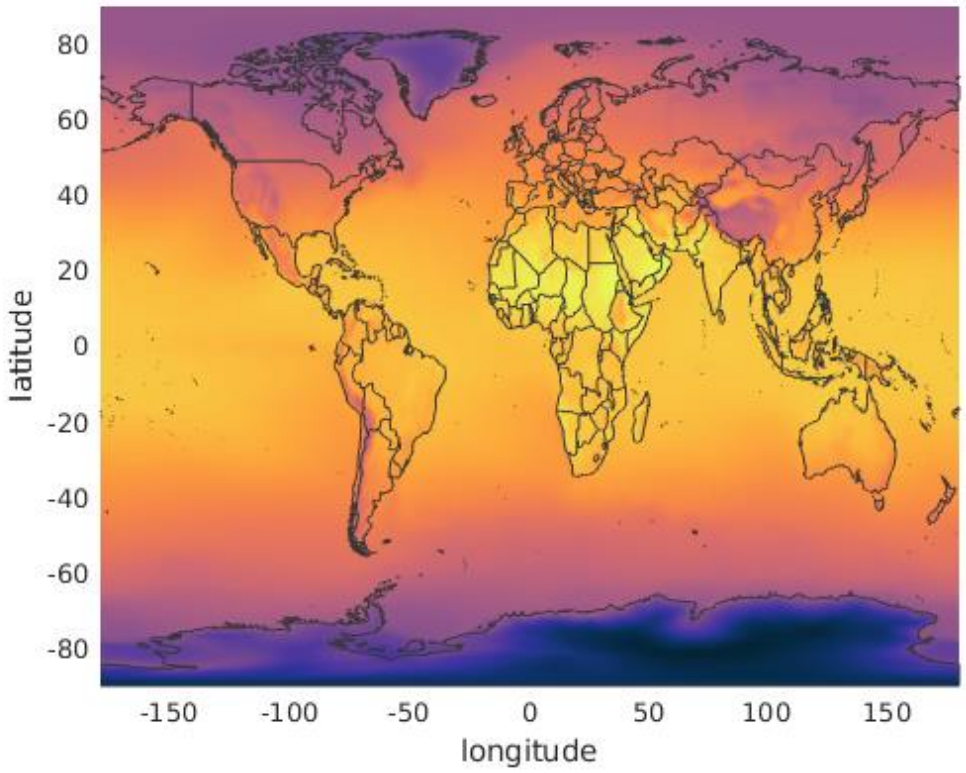
```
pcolor(double(Lon),double(Lat),Tm)

shading interp % 消除网格单元之间的黑线
cmocean thermal % 从冷到热颜色图
xlabel longitude
ylabel latitude
```



重新居中网格

您可能还注意到经度从 0 到 360，这将太平洋置于地图的中间。 使用 recenter 移动网格，使本初子午线位于中间。（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
[Lat,Lon,T,Tm] = recenter(Lat,Lon,T,Tm);




pcolor(double(Lon),double(Lat),Tm)

shading interp % 消除网格单元之间的黑线
hold on
borders('countries','color',rgb('dark gray'))
cmocean thermal % 从冷到热颜色图
xlabel longitude
ylabel latitude
```



（高阶）节省内存：仅读取所需数据

有时，NetCDF 数据集确实很大——比您研究所需的要大得多。也许它是一个密集的全球网格并且您的研究集中在地中海，或者它是一个很长的时间序列而您只需要一个时间步长。如果您只需要大型数据集的一部分，则需要找出与您感兴趣的数据相对应的索引。

地中海位于东经 5 至 45 度，北纬 28 至 45。lat 和 lon 数组比完整数据网格小得多，因此请完整加载它们以找出哪些索引对应于出于兴趣的地理范围：

```
% 加载经度数组:
lon = double(ncread('ERA_Interim_2017.nc','longitude'));

% 确定哪些索引对应于维度 1:
ind1 = find(lon>=5 & lon<=45);

% 对纬度做同样的事情:
lat = double(ncread('ERA_Interim_2017.nc','latitude'));
ind2 = find(lat>=28 & lat<=45);

% 将 lat 和 lon 剪辑到指定的范围:
lat = lat(ind2);
lon = lon(ind1);

% 做个网格:
```

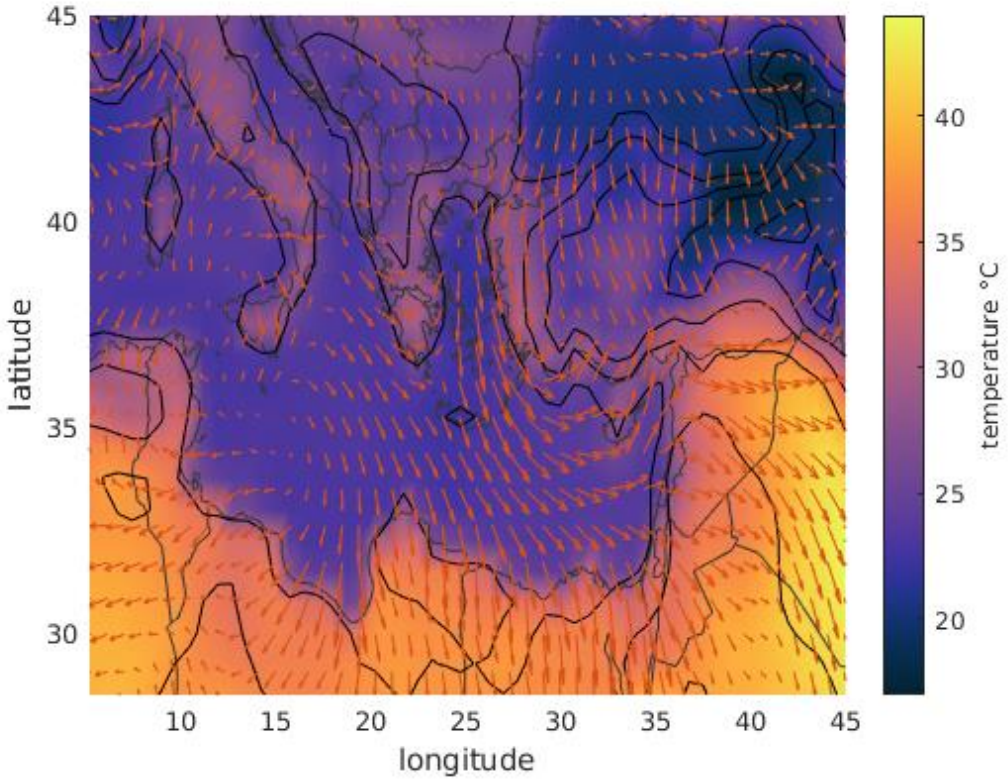
```
[Lat,Lon] = meshgrid(lat,lon);
```

ncread 函数让我们指定要加载哪些数据索引，它想要的格式是 start 和 stride，这意味着每个维度的起始索引，以及从 start 索引开始加载多少行或列数据。
网格数据集是 3 维的（经度 x 纬度 x 时间），因此 start 和 stride 数组将分别具有三个值。 我们只看 6 月份的数据，所以对于时间维度，起始索引为 6，步幅为 1。

```
start = [ind1(1) ind2(1) 6];

stride = [length(ind1) length(ind2) 1];



% 加载地中海 6 月表面气压:
sp = ncread('ERA_Interim_2017.nc','sp',start,stride);

% 还加载温度和风:
T = ncread('ERA_Interim_2017.nc','t2m',start,stride);
u10 = ncread('ERA_Interim_2017.nc','u10',start,stride);
v10 = ncread('ERA_Interim_2017.nc','v10',start,stride);
```

内存带来压力，但指定步幅和长度会更有效率。
这是我们刚刚加载的地中海数据：

```
figure

pcolor(Lon,Lat,T-273.15) % 摄氏温度
shading interp
cmocean thermal
caxis([17 44]) % 色轴限制
cb = colorbar;
ylabel(cb,'temperature \circC')
hold on

borders('countries','color',rgb('dark gray'))
contour(Lon,Lat,sp,'k') % 气压等值线
quiversc(Lon,Lat,u10,v10) % 风矢量
xlabel 'longitude'
ylabel 'latitude'
```



多个变量或多个文件

本教程从多次调用 ncread 开始——对于我们想要读取的每个变量一次。 但是，当您想要从 NetCDF 文件中读取多个变量时，这种方法可能会有点笨拙。 为了解决这个问题，CDT 提供了 ncstruct 函数，它允许您从 NetCDF 文件中读取多个变量并将结果放入 Matlab 结构体中。 就是这样：

```
S = ncstruct('ERA_Interim_2017.nc')
```

```
S =

  struct with fields:


    longitude: [480×1 single]
     latitude: [241×1 single]
         time: [12×1 int32]
           sp: [480×241×12 double]
          u10: [480×241×12 double]
          v10: [480×241×12 double]
```

```
          t2m: [480×241×12 double]

           tp: [480×241×12 double]
```

不想从 .nc 文件中读取*所有*变量？ 指定您想要阅读的内容，如下所示：

```
S = ncstruct('ERA_Interim_2017.nc','longitude','latitude','t2m')
```

```
S =

  struct with fields:


    longitude: [480×1 single]

     latitude: [241×1 single]

          t2m: [480×241×12 double]
```

有关更多选项和进一步说明，请参阅 ncstruct 。

HDF5

上述大部分原则也适用于 HDF5，这是一种与 NetCDF 非常相似的分层数据格式。使用 HDF5 的语法可能与 NetCDF 略有不同，但总体思路通常是相同的。使用 h5disp 代替 ncdisp 来显示文件的内容。使用 h5read 代替 ncread。而不是使用 CDT 函数 ncstruct 将数据读入结构体形式，而是使用 CDT 函数 h5struct。

致谢

这篇教程由 Chad A. Greene 于 2018 年 12 月创作。

日期和时间

在 Matlab 中有几种不同的处理日期和时间的方法。本教程回顾了不同的日期格式，并讨论了地球科学数据背景下的每种格式相关的相对优势和劣势。

介绍和概述

例如，我们将考虑以 NetCDF 格式打包时间的方式。当然，从 NetCDF 文件加载时间的最简单方法是使用 CDT 函数 ncdateread，但本教程关注 ncdateread 幕后发生的事情，因此我们将从示例.nc 文件中打包的原始时间格式开始。

首先读取时间数组：

```
t = ncread('ERA_Interim_2017.nc','time')
```

```
t =

  12×1 int32 column vector

   1025628

   1026372

   1027044

   1027788

   1028508

   1029252

   1029972

   1030716

   1031460

   1032180

   1032924

   1033644
```

在这一点上，您可能会感到有些不知所措。 所有这些数字是多少？ 使用 ncdisp 获得一些见解：

```
ncdisp('ERA_Interim_2017.nc','time')
```

```
Source:
```

```
            /home/chad/Documents/MATLAB/github/cdt/cdt_data/ERA_Interim_2017.nc

Format:

            64bit

Dimensions:

            time = 12     (UNLIMITED)

Variables:

    time

            Size:       12x1

            Dimensions: time

            Datatype:   int32

            Attributes:

                        units     = 'hours since 1900-01-01 00:00:00.0'

                        long_name = 'time'

                        calendar  = 'gregorian'
```

这告诉我们 time 单位是
units = 'hours since 1900-01-01 00:00:00.0'

有时单位可能是自 1971 年以来的天数或自大爆炸以来的数年或自我出生以来的秒数，因此请务必检查单位。
自 1900 年以来以小时为单位的时间不太容易理解，因此我建议养成一致使用 datenum 或 datetime 格式的习惯。 Matlab 可以很容易地在 datenum 和 datetime 格式之间切换，但它们各有优缺点。这是一个细分：

- datenum: 单位只是自公元 0 年 1 月 1 日以来的天数,这是几十年来 Matlab 中的标准日期格式。如果您现在在命令窗口中键入，它会显示类似 ans = 7.3740e+5 的内容，因为距耶稣出生已经超过 737,000 天了。使用 datestr、datevec 或 datetime 将 datenum 转换为日历日期。
- datetime: 这种格式是在 Matlab 2014b 中引入的。因为 datetime 是一个对象，它能够在很多方面变得“智能”，因为 Matlab 总是知道它代表时间（而不是 datenum，它只是 Matlab 的一个数字）。但由于 datetime 是“智能”的，它有时会出现意外行为或不兼容，在这种情况下 datenum 可能只是更好的选择

对于我们的示例文件，请记住 t 表示自 1900 年元旦午夜以来的小时数。这就是 t 中的所有数字。

datenum

要将这些小时转换为 datenum 格式，请使用 1900,1,1 作为日期并将 t 放在小时位置。
此外，NetCDF 文件中的 time 变量采用 int32 格式，datenum 不接受该格式。所以在调用 datenum 时将 t 转换为 double：

```
t = datenum(1900,1,1,double(t),0,0)
```

```
t =

      736696.50

      736727.50

      736755.50

      736786.50

      736816.50

      736847.50

      736877.50

      736908.50

      736939.50

      736969.50

      737000.50

      737030.50
```

datestr

这些是日期数字，它们对我们人类没有多大意义。 但是我们已经可以看到一些有趣的东西：datenum 不是整数——它们都以 0.5 结尾，这意味着时间对应于每天的中午。 要了解这些 datenum 的含义，只需将它们放入 datestr 函数中即可：

```
datestr(t)
```

```
ans =

  12×20 char array
```

```
    '01-Jan-2017 12:00:00'

    '01-Feb-2017 12:00:00'

    '01-Mar-2017 12:00:00'

    '01-Apr-2017 12:00:00'

    '01-May-2017 12:00:00'

    '01-Jun-2017 12:00:00'

    '01-Jul-2017 12:00:00'

    '01-Aug-2017 12:00:00'

    '01-Sep-2017 12:00:00'

    '01-Oct-2017 12:00:00'

    '01-Nov-2017 12:00:00'

    '01-Dec-2017 12:00:00'
```

这正是我们所期望的：这个月度数据对应于 2017 年每个月第一天的中午。

datevec

另一种有用的格式称为 datevec。日期向量非常有用，尤其是当您只想分析与九月份相关的数据时。日期向量为您提供与任何给定时间 t 相关联的年、月和日（以及小时、分钟和秒，如果您愿意）。以下是从 datenum 数组中获取日期向量的方法：

```
[year, month, day] = datevec(t);
```

有了这个，我们现在知道每个时间 t 的年、月和日。它们在这里，一起显示：

```
[year month day]
```

```
ans =

       2017.00          1.00          1.00

       2017.00          2.00          1.00

       2017.00          3.00          1.00
```

```
    2017.00    4.00    1.00
    2017.00    5.00    1.00
    2017.00    6.00    1.00
    2017.00    7.00    1.00
    2017.00    8.00    1.00
    2017.00    9.00    1.00
    2017.00   10.00    1.00
    2017.00   11.00    1.00
    2017.00   12.00    1.00
```

这很有用的原因是现在很容易获得与给定月份相对应的所有日期索引。 例如，以下是与任何 9 月数据相对应的所有索引：

```
ind = month==9
```

```
ind =

  12×1 logical array

   0
   0
   0
   0
   0
   0
   0
   0
   0
   1
```

```
   0

   0

   0
```

datetime

如果您想将 datenum 转换为 datetime 格式，只需将 t 放入 datetime 函数，并告诉它您要从 datenum 转换：

```
t = datetime(t,'ConvertFrom','datenum')
```

```
t =

  12×1 datetime array

   01-Jan-2017 12:00:00

   01-Feb-2017 12:00:00

   01-Mar-2017 12:00:00

   01-Apr-2017 12:00:00

   01-May-2017 12:00:00

   01-Jun-2017 12:00:00

   01-Jul-2017 12:00:00

   01-Aug-2017 12:00:00

   01-Sep-2017 12:00:00

   01-Oct-2017 12:00:00

   01-Nov-2017 12:00:00

   01-Dec-2017 12:00:00
```

转换回 datenum 就是这么简单：

```
t = datenum(t)
```

```
t =
```

736696.50

736727.50

736755.50

736786.50

736816.50

736847.50

736877.50

736908.50

736939.50

736969.50

737000.50

737030.50

作者简介

这篇教程由 Chad A. Greene 于 2019 年 2 月为 Climate Data Toolbox for Matlab 创作。

线性趋势

最小二乘线性回归是地球科学中最常见的分析类型之一，而 Matlab 的左除算子可以实现有效的最小二乘计算，这是 Matlab 最初在地球科学家中流行的一个重要原因。 最小二乘和矩阵乘法的数学在网上和许多教科书中都有很好的介绍，因此本教程仅介绍如何使用 Matlab 气候数据工具箱（Climate Data Toolbox for Matlab）计算最小二乘回归的机制。

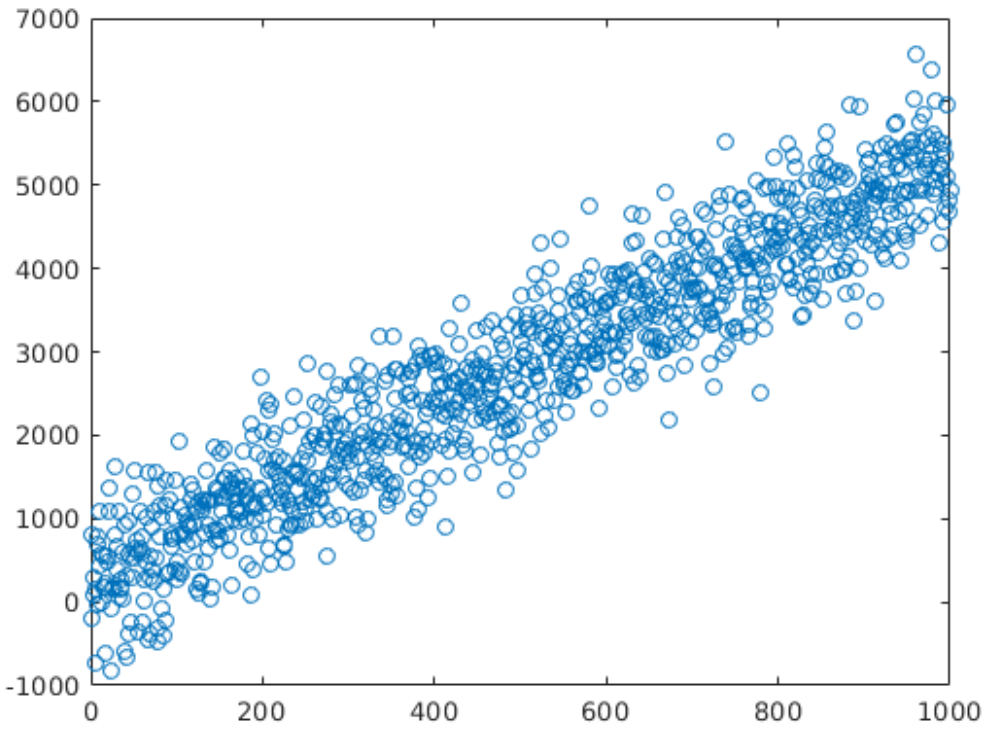
trend 函数

在本例中，我们将创建一些具有已知趋势的随机数据，然后我们将使用 trend 函数对数据进行线性最小二乘回归拟合。 首先，生成数据：

```
x = 1:1000;

y = 5*x + 500*randn(size(x)) + 300;
```

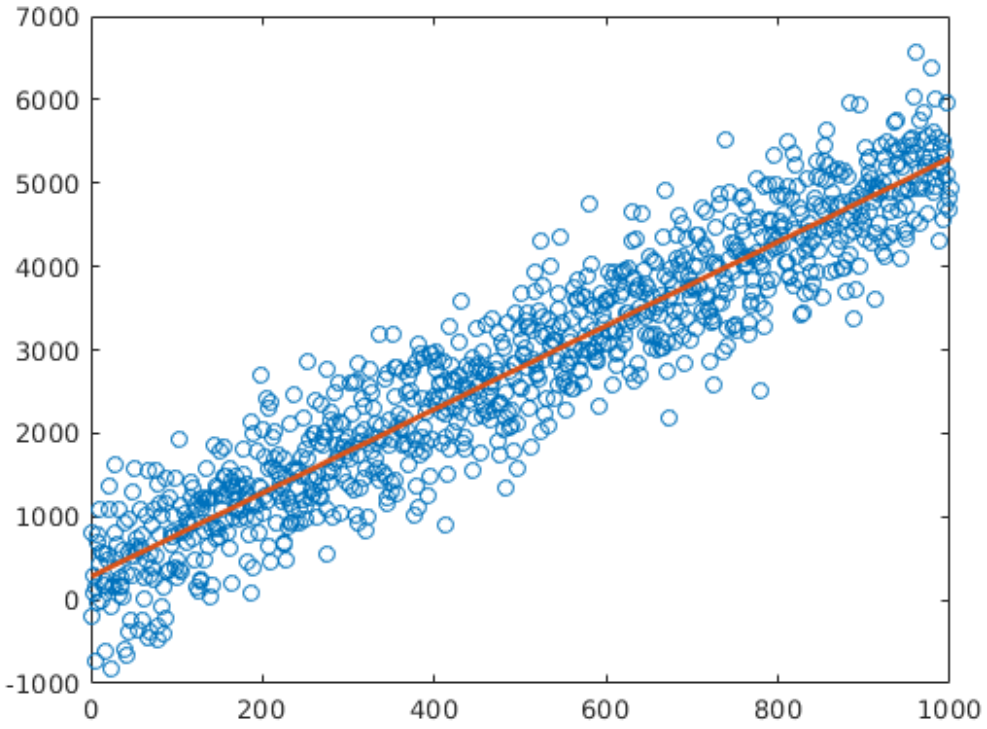
以上，我们强加了 y = 5*x 的趋势，然后添加了一些噪声和 300 的 y 截距。这是数据：

```
plot(x,y,'o')
```



使用 polyplot 快速绘制一阶最小二乘趋势：

```
hold on
polyplot(x,y,1,'linewidth',2)
```



在地球科学中最常见的问题是，这条线的斜率是多少？ y 如何随 x 变化？ 我们知道斜率应该约为 y=5*x，但是最小二乘法是什么意思呢？ 使用 trend 函数找出：

```
trend(y,x)
```

```
ans =

    5.02
```

大约是 5。这正是我们所期望的答案。

polyfit 和 polyval 函数

trend 函数因其简单而方便，但您可能希望获得更多信息，而不仅仅是趋势线的斜率。 例如，您可能想知道 y 截距或一些高阶最小二乘拟合。对于这种情况，标准的 Matlab 函数 polyfit 可能会派上用场。 polyfit 函数通过最小二乘法将任何阶多项式拟合到数据集。 使用 polyfit 将一阶多项式拟合到 x,y 数据如下所示：

```
P = polyfit(x,y,1)
```

```
P =

    5.02    279.80
```

以上，P 包含多项式的系数，从最高阶开始。 由于这是一阶最小二乘拟合，因此 P 的内容分别为 [P_1 P_0]，即斜率和截距。 因此，斜率 P_1 约为 5，截距 P_0 约为 300 也就不足为奇了。这些是我们在创建 y 时强加的值。 如果我们想得到二阶拟合的系数，我们会使用

```
P = polyfit(x,y,2)
```

```
P =

          -0.00            5.22          246.27
```

然而，将二阶或更高阶拟合到这个特定的数据集是不合适的，我们知道因为当我们定义 y 时，我们说它有一个斜率、一个 y 截距和噪声，没有别的。 这意味着即使拟合二阶多项式 y 也会使模型适应噪声。 这有时被称为过度拟合。
为了说明过度拟合，让我们将 25 阶多项式拟合到 y：

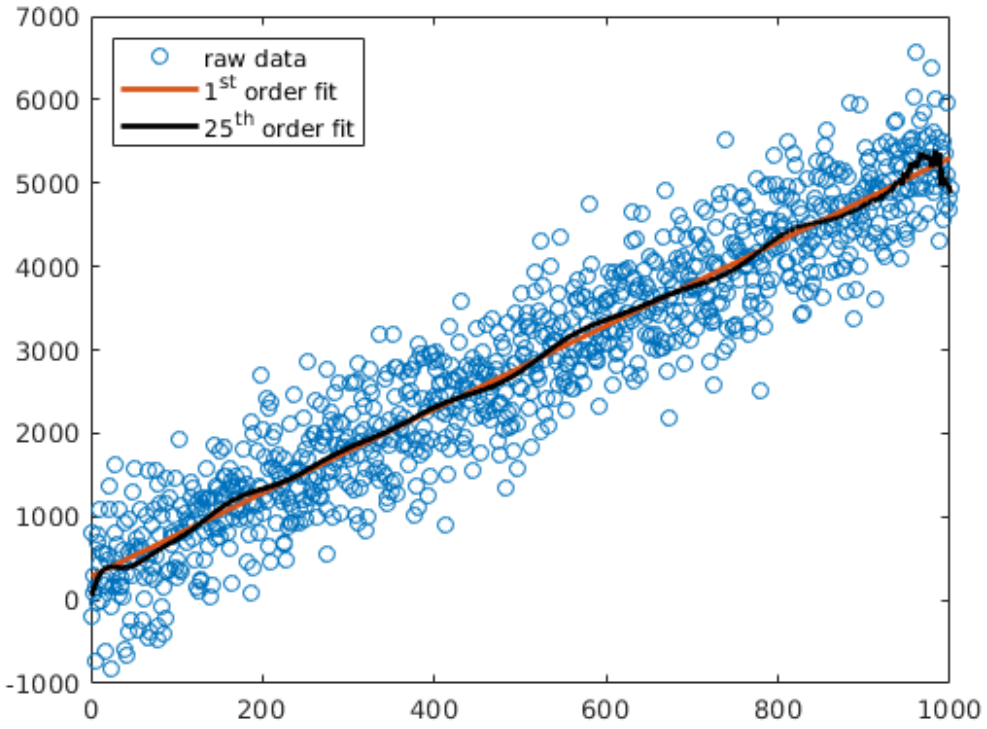
```
P = polyfit(x,y,25);
```

```
Warning: Polynomial is badly conditioned. Add points with distinct X values,

reduce the degree of the polynomial, or try centering and scaling as described

in HELP POLYFIT.
```

（警告：多项式条件不佳。 添加具有不同 X 值的点，减少多项式的次数，或尝试按照 HELP POLYFIT 中的说明进行居中和缩放。）

现在使用 polyval 函数计算每个 x 的 25 阶 P 并将其绘制为一条粗黑线：

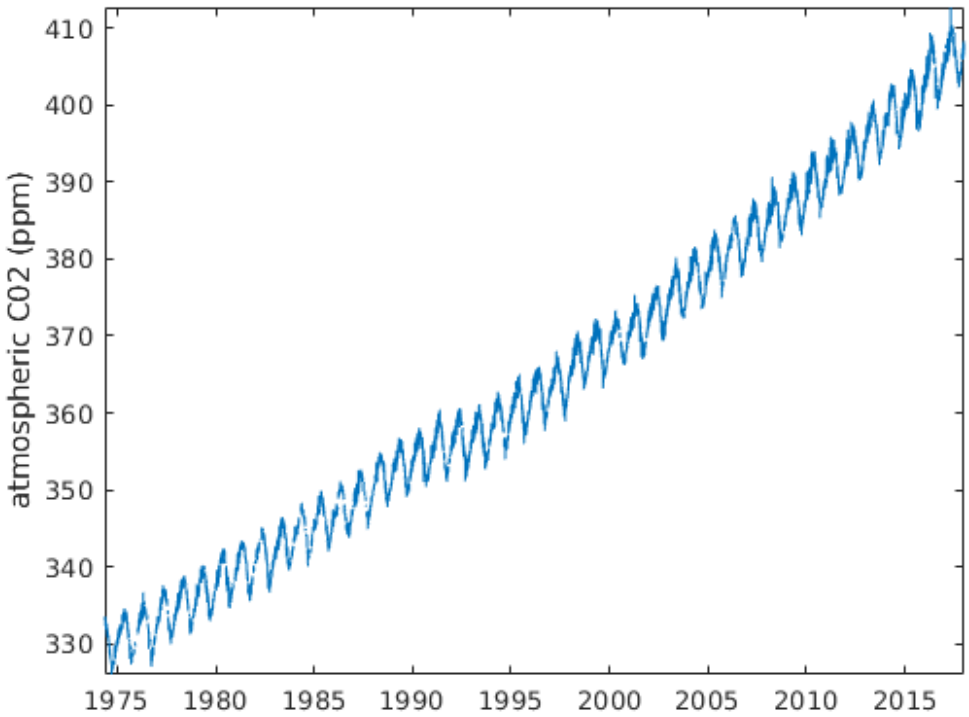
```
y_overfit = polyval(P,x);



plot(x,y_overfit,'k','linewidth',2)

legend('raw data','1^{st} order fit','25^{th} order fit',...
   'location','northwest')
```



大气 CO2 的趋势

现在让我们应用上述方法来评估在夏威夷莫纳罗亚 (Mauna Loa) 测量的大气 CO2 的变化。 首先加载数据并绘图：

```
load mlo_daily_c02.mat

figure
plot(t,C02)
axis tight
datetick('x','keeplimits') % 将 x 刻度格式化为日期
ylabel('atmospheric C02 (ppm)')
```



在整个测量期间，大气中的 CO2 以什么速率增加？ 使用 trend 函数找出：

```
trend(CO2,t)
```

```
ans =

           NaN
```

NaN 表示非数字。 这是因为 CO2 数据集包含一些 NaN 值。 这在真实数据集中很常见，有时会出现空白或缺失数据，但有一种简单的方法来处理它。 只需确定 CO2 数据集的哪些元素是有限的，然后只分析这些元素。

```
% 确定哪些 CO2 指数不是 NaN：
isf = isfinite(CO2);

% 计算有限值之间的趋势：
trend(CO2(isf),t(isf))
```

```
ans =

           0.00
```

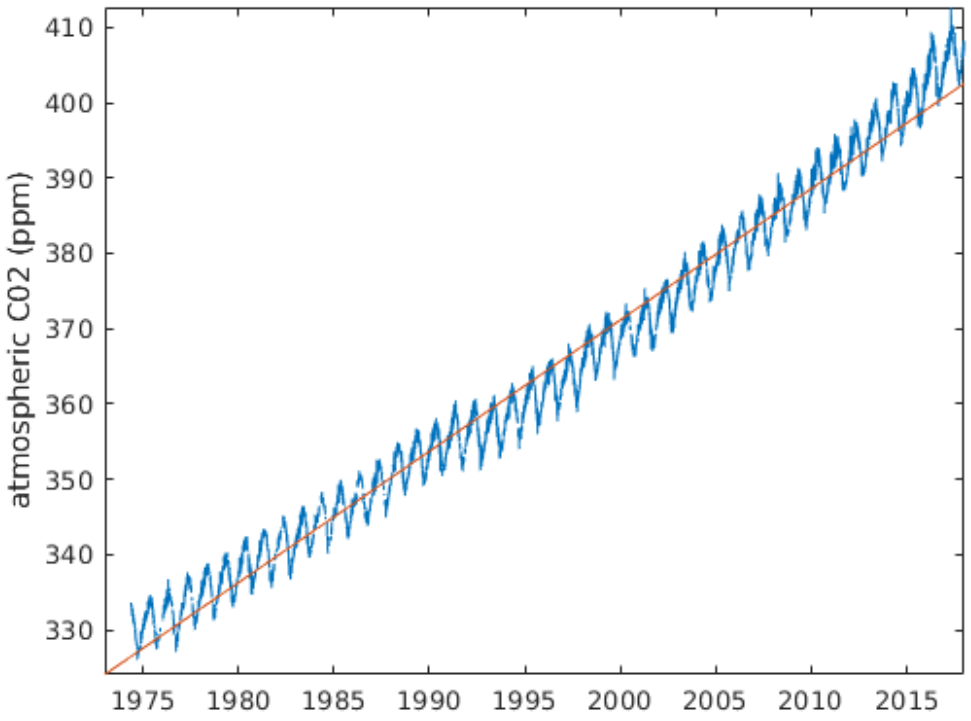
大约为零的趋势似乎根本不是趋势。但是注意，trend 函数计算的是单位时间的 CO2 变化，时间单位是 datenum，也就是天。（在本教程中阅读有关日期格式的更多信息。）因此，每天二氧化碳(ppm)的趋势接近于零也就不足为奇了。更有意义的衡量标准可能是每十年的趋势。
要将 CO2 趋势从 ppm/天转换为 ppm/十年，只需乘以每年 365.25 天，然后再乘以每十年 10 年：

```
trend(CO2(isf),t(isf))*365.25*10
```

```
ans =

    17.43
```

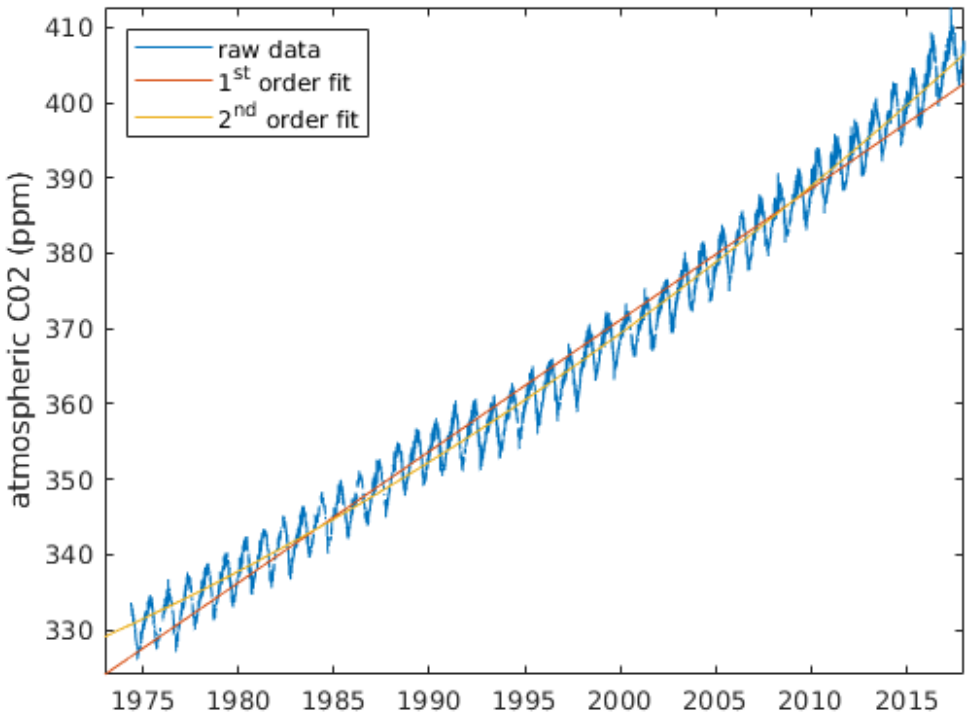
在该数据集中，大气 CO2 的趋势是每十年约 17 ppm。 使用 polyplot 在图上显示趋势线：

```
hold on
polyplot(t,CO2,1)
```



在上图中，我们看到每十年约 17 ppm 的一阶拟合当然符合整体趋势，但它并没有完全捕捉曲线的长期形状。 这里可能适合使用高阶拟合。 让我们展示一个二阶拟合：

```
polyplot(t,CO2,2)

legend('raw data','1^{st} order fit','2^{nd} order fit',...
   'location','northwest')
```



可以使用 polyfit 函数找到相应的多项式常数

```
P = polyfit(t(isf),CO2(isf),2);
```

```
Warning: Polynomial is badly conditioned. Add points with distinct X values,

reduce the degree of the polynomial, or try centering and scaling as described

in HELP POLYFIT.
```

（警告：多项式条件不佳。 添加具有不同 X 值的点，减少多项式的次数，或尝试按照 HELP POLYFIT 中所述进行居中和缩放。）

出现警告消息是因为 t (datenum) 的单位与 CO2 (ppm) 的单位非常不同。您会看到，与 CO2 的值相比，t 的值非常大。 为了说明这一点，请看 t 的第一个元素：

```
t(1)
```

```
ans =

      720626.00
```

这是一个非常大的数字。 这是自 0 年元旦以来的天数。 处理这个大数字的一种快速简便的方法是使用 doy 函数将其转换为十进制年份：

```
yr = doy(t,'decimalyear');
```

现在第一个日期看起来像这样：

```
yr(1)
```

```
ans =

        1973.01
```

这个数字足够小，我们可以用多项式拟合它和二氧化碳数据。 将 `yr` 除以 10 以获得相对于几十年而不是年的多项式：

```
P = polyfit(yr(isf)/10,CO2(isf),2)
```

```
P =

           1.26        -486.24       47162.38
```

第一个元素 `P` 中的正值告诉我们一些我们已经知道的事情：大气 CO2 不仅在增加，而且还在*加速*。

3D 数据集

上面的例子是一维数组，每个数组代表一个时间序列。 在气候科学中，我们经常以网格数据的形式同时处理数千个这样的时间序列。 在这些类型的三维网格中，前两个维度通常是空间维度，如经度和纬度，第三个维度对应于时间。 每个网格单元都包含自己的时间序列，处理这些时间序列的一种方法是循环遍历每个网格单元，对每个网格单元的时间序列进行一维分析。 我们将在本教程中更深入地介绍该方法，但首先我们将使用 `trend` 函数来计算三维数据集的趋势。

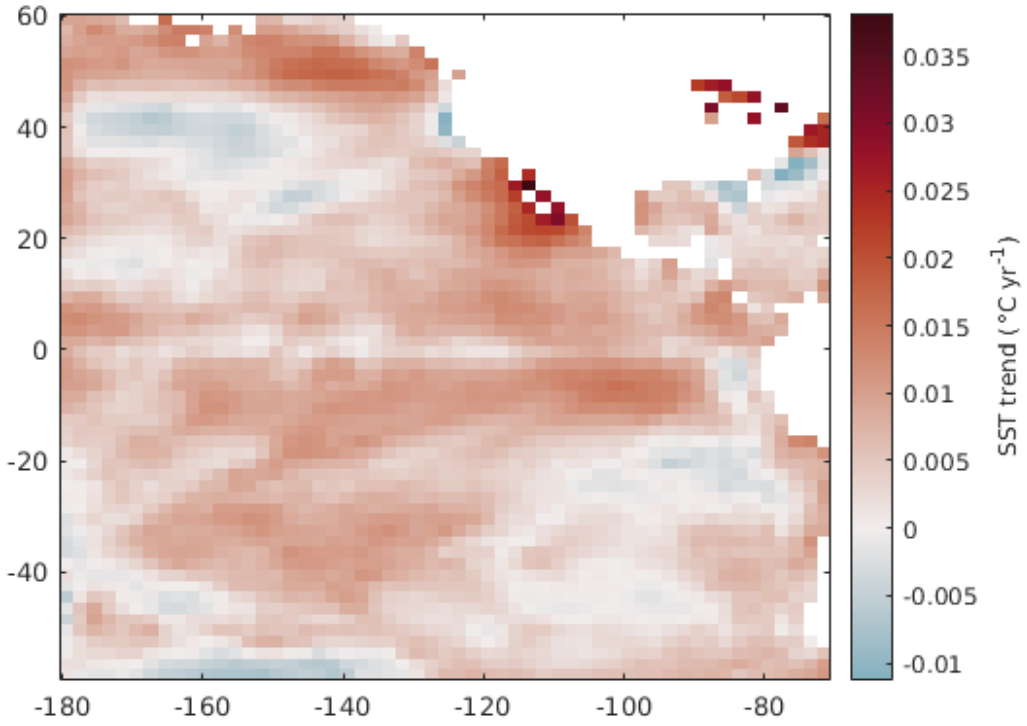
从加载数据开始

```
load pacific_sst

whos lat lon sst t % 显示变量大小
```

```
  Name        Size                    Bytes  Class     Attributes

  lat        60x1                      480  double
  lon        55x1                      440  double
  sst        60x55x802             21172800  double
  t         802x1                     6416  double
```

pacific_sst 样本数据集包含 60x55 网格上的 802 个每月海面温度。 网格分辨率相当粗糙，只覆盖了太平洋的一部分，但值得注意的是，即使是这个小数据集也包含 60x55=3300 个单独的时间序列。 虽然遍历每个网格单元可能很诱人，单独计算每个网格单元的时间序列的趋势，但我们应该将这种方法视为最后的手段，因为这是一种非常缓慢的处理方式。 所以只要有可能，尽量避免循环，只操作一次。
trend 函数一次完成整个操作。 以下是如何使用它：

```
% 计算每年的趋势，每年 12 个样本（每月数据）：
tr = trend(sst,12);

% 绘制线性趋势:
figure
imagescn(lon,lat,tr)
cb = colorbar;
ylabel(cb,'SST trend (\circC yr^{-1})')
cmocean('balance','pivot') % 将颜色图设置为零在中间
```



上面的 trend 计算速度很快，因为它只需要计算一次，而不是 3300 次。 （它使用 cube2rect 对 sst 数据集进行整形，然后执行最小二乘拟合，然后使用 rect2cube 对结果进行“取消整形”。）

处理三维数据集中的 NaN

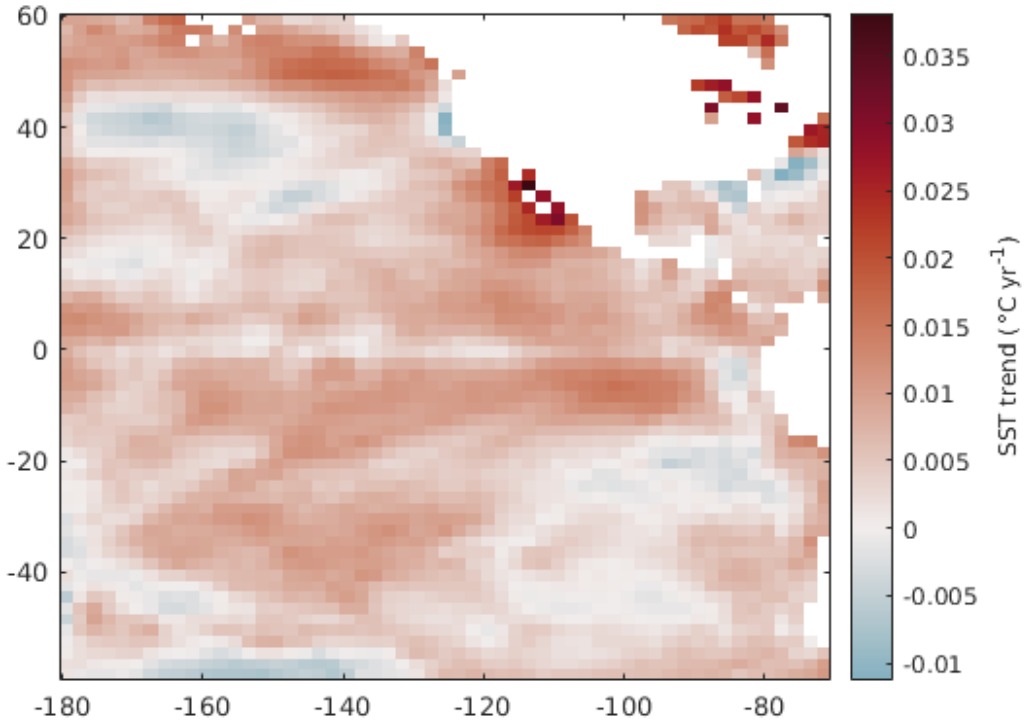
在某些情况下，您可能别无选择，只能使用循环来计算三维网格数据集中的趋势。 例如，如果数据中有一些分散的 NaN 值。 在这种情况下，循环遍历三维数据集的每一行和每一列，确定每个网格单元中哪些时间索引是有限的，并相应地计算趋势。 在开始循环之前不要忘记 preallocate 。

```
% 预分配:
sst_trend = nan(60,55);

% 每行循环:
for row = 1:60

   % 每列循环:
   for col = 1:55

      % 获取此网格单元中有限数据的时间索引:
      ind = isfinite(sst(row,col,:));

      % 仅在至少有两个有限索引时才计算趋势:
      if sum(ind)>=2
         sst_trend(row,col) = trend(squeeze(sst(row,col,ind)),t(ind))*365.25;
      end
   end
end
```

请注意上面的 squeeze 命令，它将 1x1xN 数组转换为 Nx1 数组。

现在绘制我们刚刚基于每个网格单元计算的 sst_trend ，只需确认它一次在整个数据集上使用 trend 的结果匹配：

```
figure

imagescn(lon,lat,sst_trend)

cb = colorbar;

ylabel(cb,'SST trend (\circC yr^{-1})')
cmocean('balance','pivot') % 将颜色图设置为零在中间
```



作者简介

这篇教程由 Chad A. Greene 于 2019 年 2 月为 Climate Data Toolbox for Matlab 创作。

Matlab 画地图

在 Matlab 中有多种制作地图的方法。本教程回顾了每种方法的优缺点。

载入数据

在下面的示例中，我们将使用几种不同的方法映射来自示例 ERA_Interim_2017.nc 的表面气压、温度和风数据。 从加载数据开始。 有关加载 NetCDF 数据的更多信息，请查看这个教程。

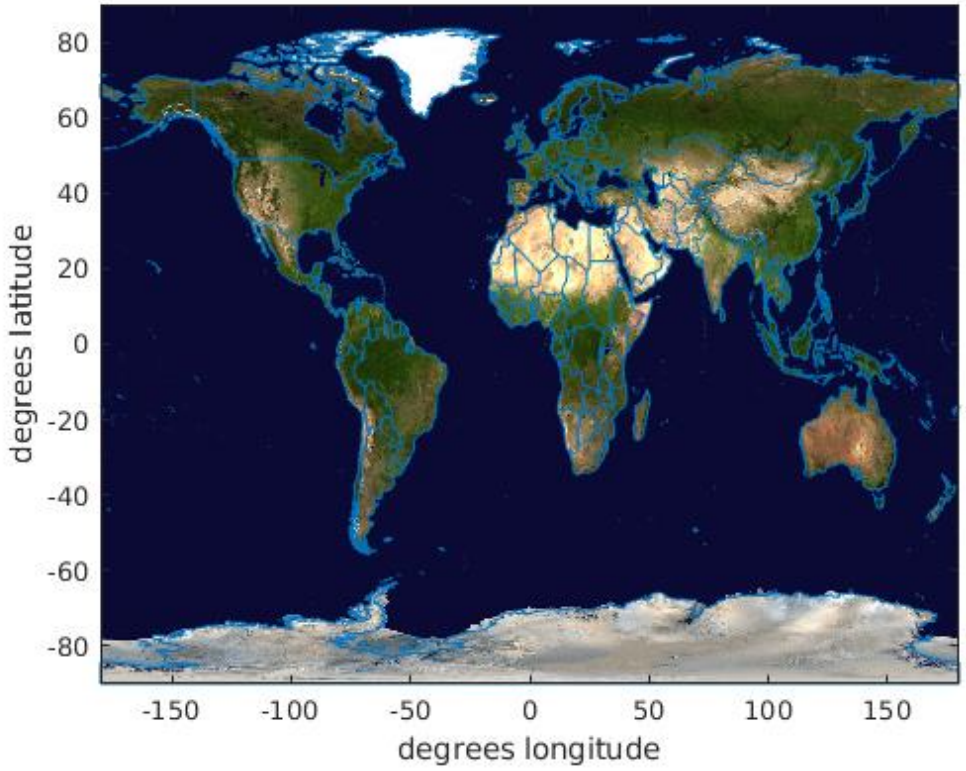
```
filename = 'ERA_Interim_2017.nc';

sp = ncread(filename,'sp');
T = ncread(filename,'t2m');
u10 = ncread(filename,'u10');
v10 = ncread(filename,'v10');

lon = double(ncread(filename,'longitude'));
lat = double(ncread(filename,'latitude'));
```

未投影坐标系

绘制网格地球数据集的最简单且计算效率最高的方法是在未投影坐标中。 这意味着我们不会试图解释这样一个事实，即经纬度网格中的网格单元在极点附近比在赤道处小。

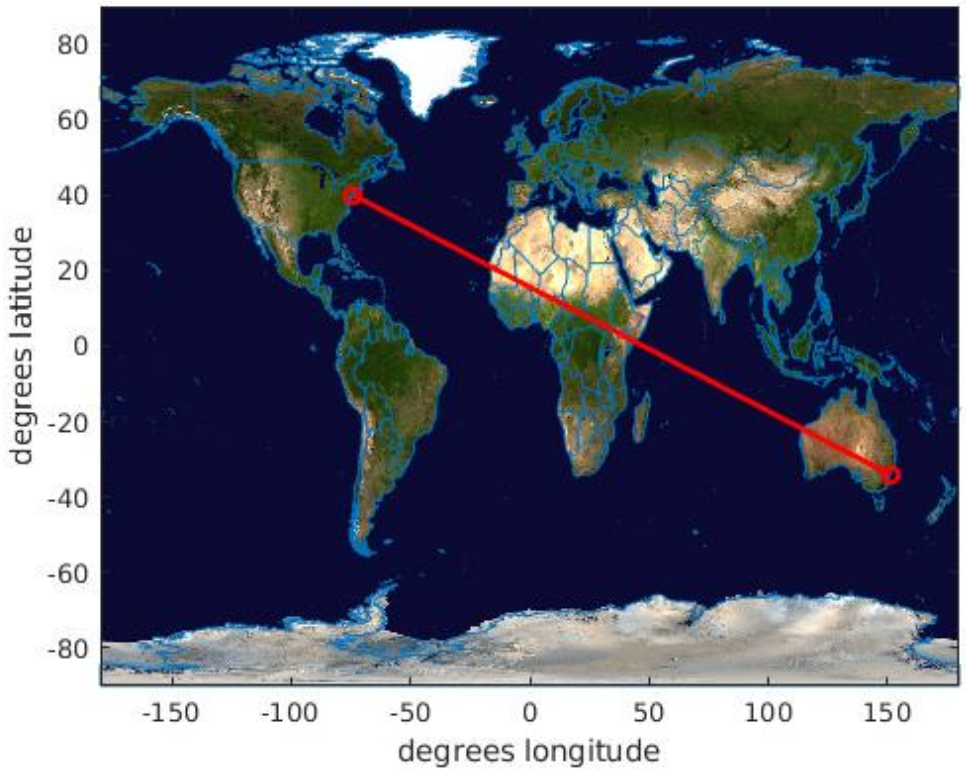
这是使用 earthimage 函数制作的简单未投影地球地图，使用 borders 函数覆盖了政区边界：（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
earthimage % 创建背景图
hold on    % 允许绘制更多对象
borders    % 绘制政区边界

xlabel 'degrees longitude'
ylabel 'degrees latitude'
```



在上面的地图中，您可以看到经度相当于 x 轴，而纬度实际上相当于 y 轴。 这意味着我们可以使用常规绘图函数，将经度视为 x，将纬度视为 y。 例如，要绘制一条从纽约 (40N,74W) 到悉尼 (34S,151E) 的线，只需使用标准 plot 函数，如下所示：（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

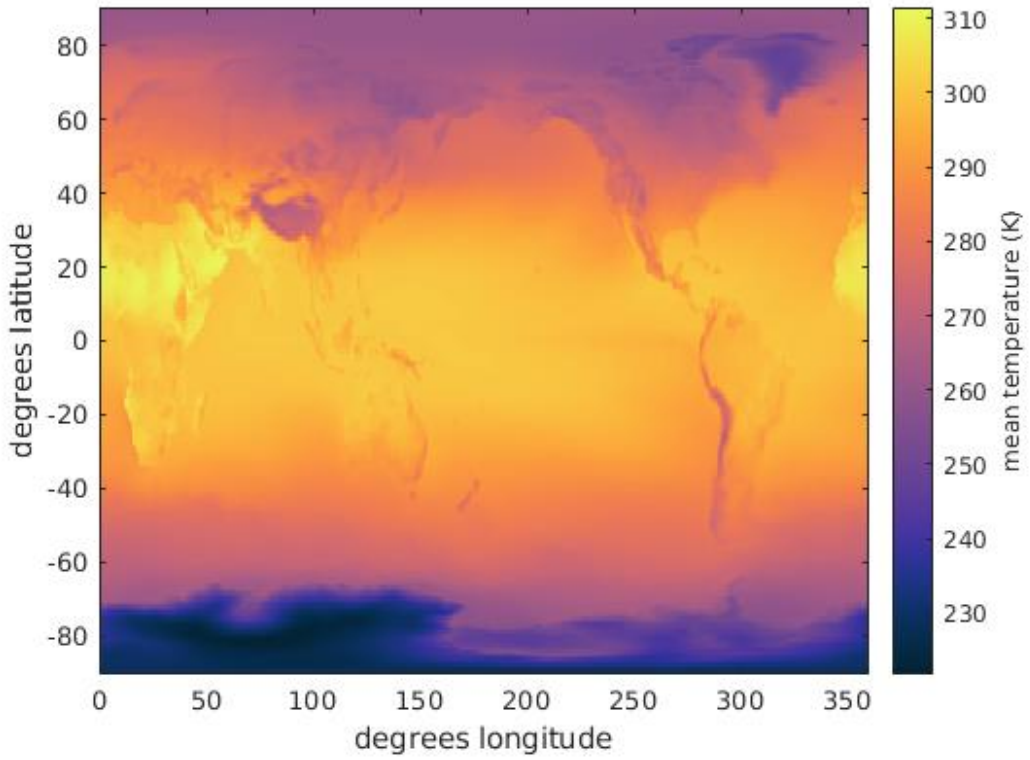
```
% 一条从纽约到悉尼的线:
plot([-74 151],[40 -34],'ro-','linewidth',2)
```



imagesc 和 imagescn

在制作地图时使用标准绘图函数方便、简单、有些直观，并且在某些情况下是有意义的。这也很好，因为不需要特殊的工具箱。也许最大的优势是速度。当网格数据集变大时，将网格扭曲为投影坐标的特殊映射函数可能会很慢并占用大量内存。因此，如果您遇到绘图速度问题，您的第一道防线应该是 imagesc。或者更好的是，使用 CDT 函数 imagescn，它会自动将 y 轴翻转到正确的方向并使 NaN 网格单元透明。这是使用 imagescn 绘制的平均表面温度，颜色图由 cmocean 设置。 我们还将使用 quiversc 来显示平均风向量：

```
figure

imagescn(lon,lat,mean(T,3)')

cmocean thermal
cb = colorbar;
ylabel(cb,'mean temperature (K)')
xlabel 'degrees longitude'
ylabel 'degrees latitude'
```

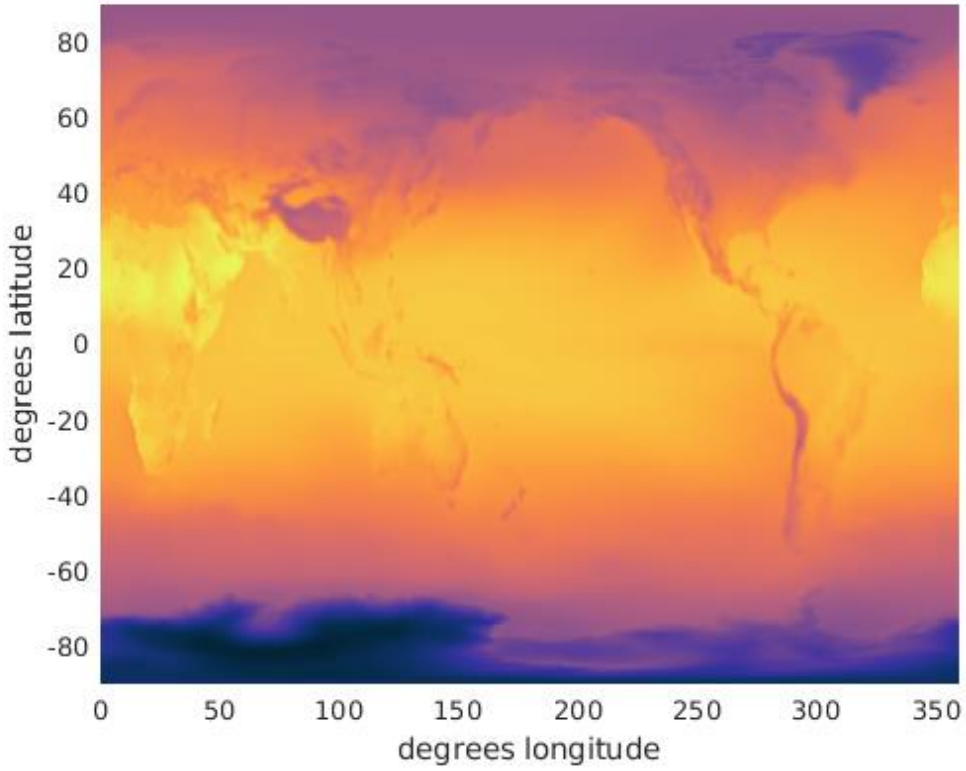


pcolor

在上面的地图中，我们必须转置（使用 ' 运算符旋转 90 度）平均温度网格以正确定位。原因在 NetCDF 教程中进行了解释。

如果您不想弄乱旋转网格，可以使用 pcolor 而不是 imagesc 或 imagescn，但您必须先将 lat,lon 数组转换为网格，如下所示：

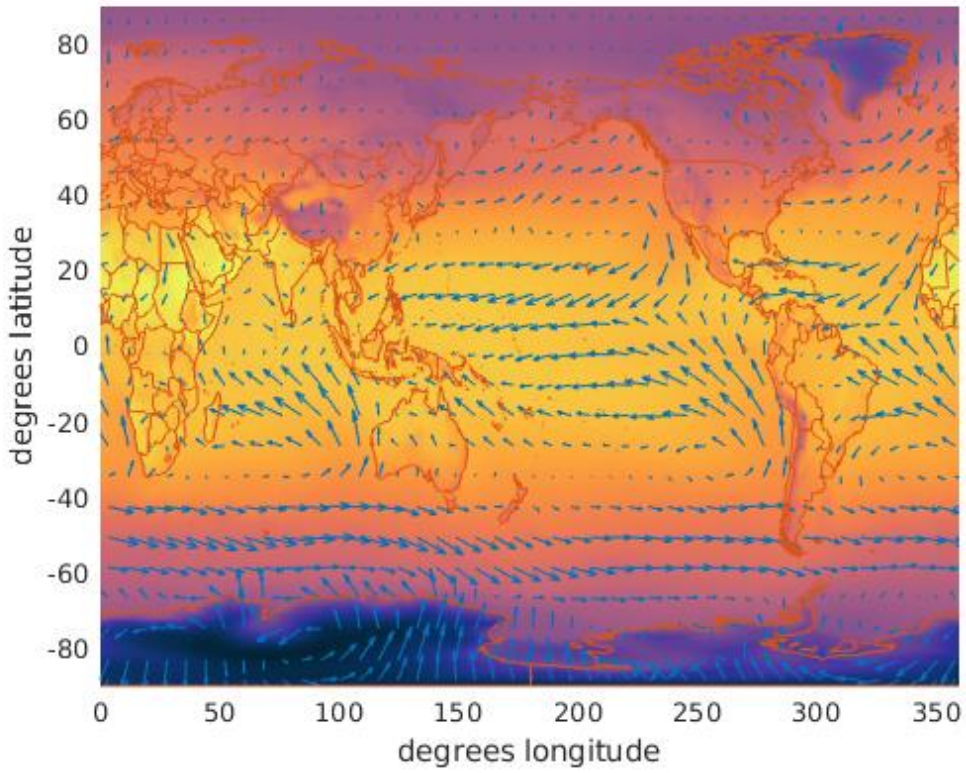
```
% 将 lat,lon 数组转换为网格:
[Lat,Lon] = meshgrid(lat,lon);

figure
pcolor(Lon,Lat,mean(T,3))
shading flat
cmocean thermal
xlabel 'degrees longitude'
ylabel 'degrees latitude'
```



请注意，pcolor 后面总是需要跟有 shading flat 或 shading interp 以摆脱通常使整个地图看起来像纯黑色矩形的黑线网格。pcolor 还存在一些性能问题，在 imagescn 中讨论过，但它通常仍然是一个有用的函数。

由于我们现在有了从 lat,lon 数组生成的 Lat,Lon 网格，我们现在也可以使用其他函数，例如 quiversc 来绘制平均全球风场：（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

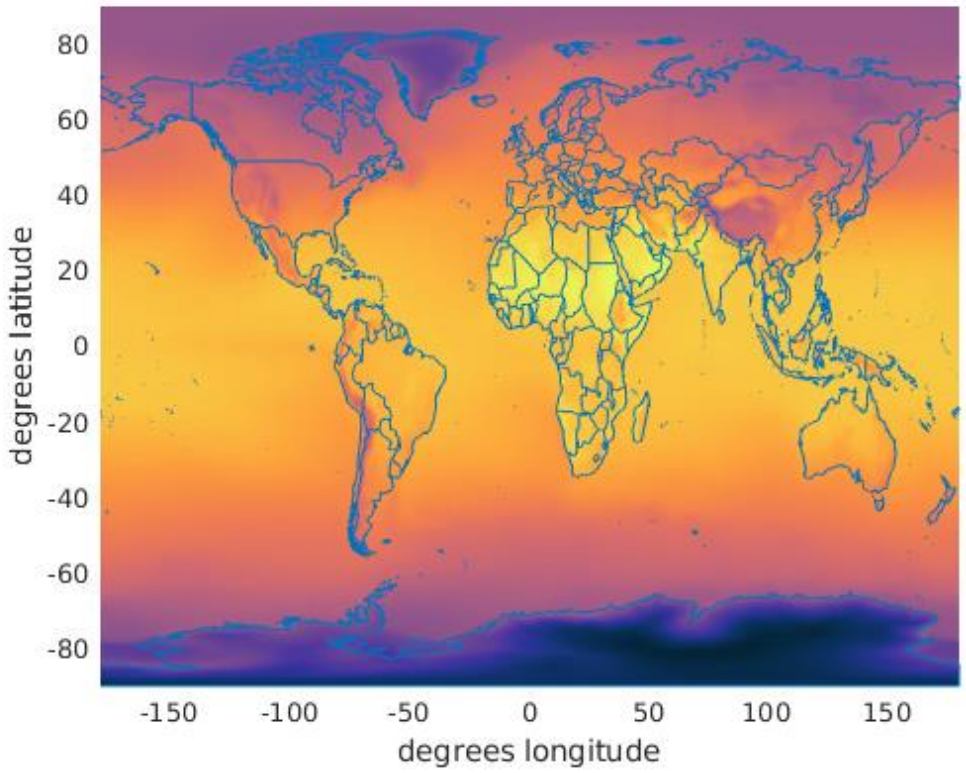
```
hold on % 允许在令 pcolor 绘图的顶部绘制更多的东西
quiversc(Lon,Lat,mean(u10,3),mean(v10,3))
borders('countries','center',180)
```



以上，我们使用 borders 函数来绘制政治边界，但我们必须指定中心经度为 180 度，因为此网格从 0 到 360，而不是从 -180 到 180。

如果您想重新定位基础数据，使本初子午线（或任何任意经线）位于网格的中心，只需使用 recenter 函数：（译者注：此图有政治问题！藏南地区是中华人民共和国西藏自治区不可分割的一部分！）

```
[Lat,Lon,T,u10,v10,sp] = recenter(Lat,Lon,T,u10,v10,sp);



figure

pcolor(Lon,Lat,mean(T,3))

shading interp
cmocean thermal
borders
xlabel 'degrees longitude'
ylabel 'degrees latitude'
```



在上面的地图中，您可以看到，现在我们已经将网格数据重新居中，0 度经度位于地图的中间，并且在调用 borders 函数时我们不必指定中心经度。

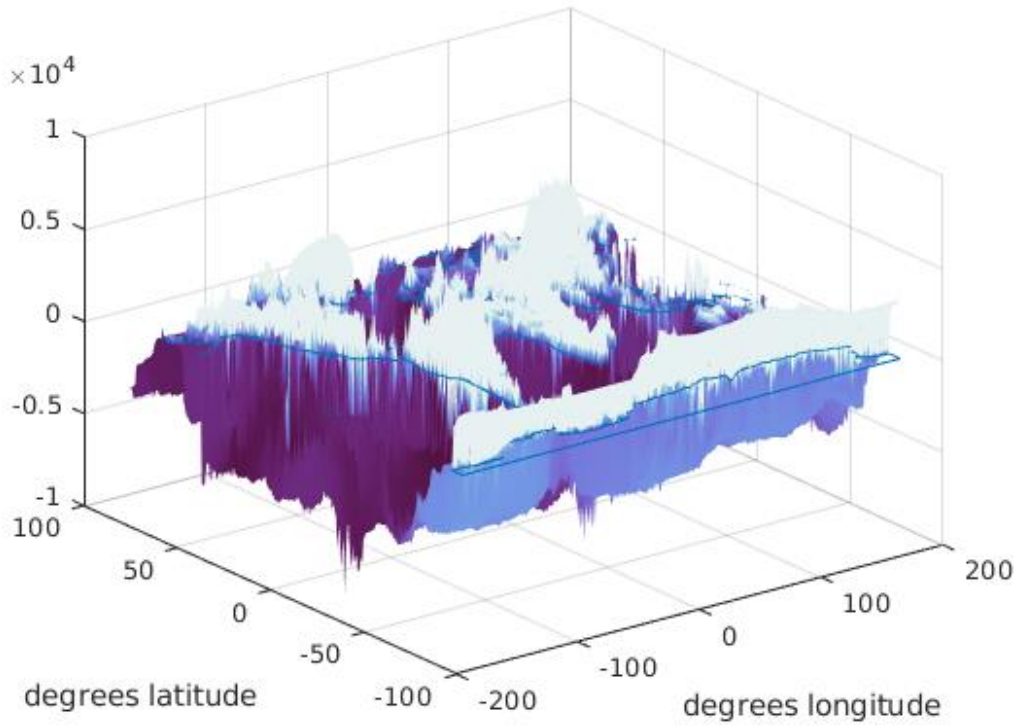
地形

有时在表面形貌中显示数据很好。要获得 Lat,Lon 网格上每个点的高程，请使用 topo_interp 函数：

```
Z = topo_interp(Lat,Lon);
```

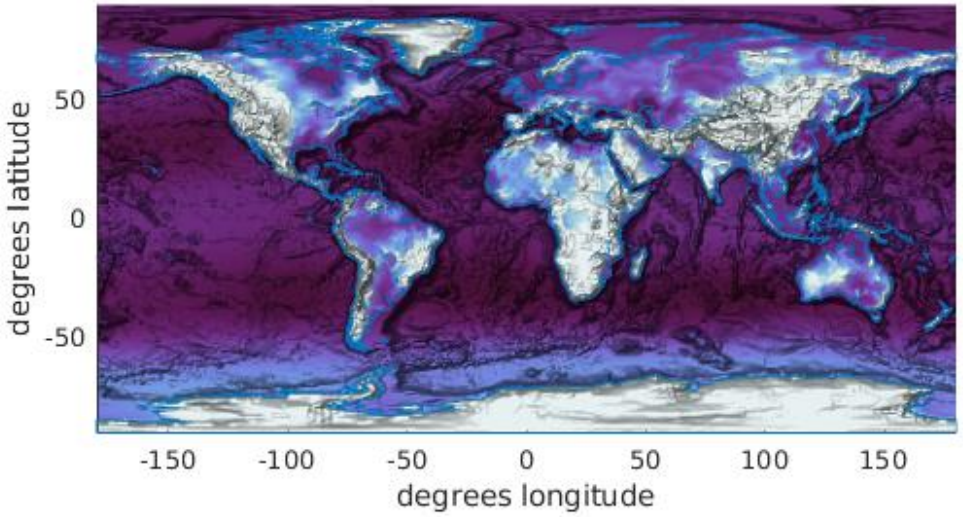
现在使用 surf 与我们在上面使用 pcolor 相同，但 Z 表示表面高程，sp 表示表面气压：

```
figure

surf(Lon,Lat,Z,mean(sp,3))

shading interp
cmocean dense
borders
xlabel 'degrees longitude'
ylabel 'degrees latitude'
caxis([950 1028]*100) % 设置颜色轴限制
```



三维视图不是很方便看，因此使用 view(2) 更改为二维视图并使用 shadem 添加一些浮雕阴影：（这个函数应该已经被淘汰了，目前的官网无此链接，但是可以访问）

```
view(2)              % 设置 2D 视角
axis image           % 去除空白并设置 1:1 纵横比
shadem(-7,[225 80] )% 添加山体阴影
```



投影坐标

尽管使用非投影坐标有一些优点，但也有一些缺点。即，未投影的坐标不是很漂亮，并且它们倾向于扩大两极附近的空间，同时使赤道附近的网格单元的相对大小最小。这可能会给人一种对重要物理现象正在发生的地方的错误印象。未投影的坐标还会在区域地图中引入不必要的失真，尤其是当您离两极距离更近时。如果要在投影坐标中制作地图，则有两个选择：

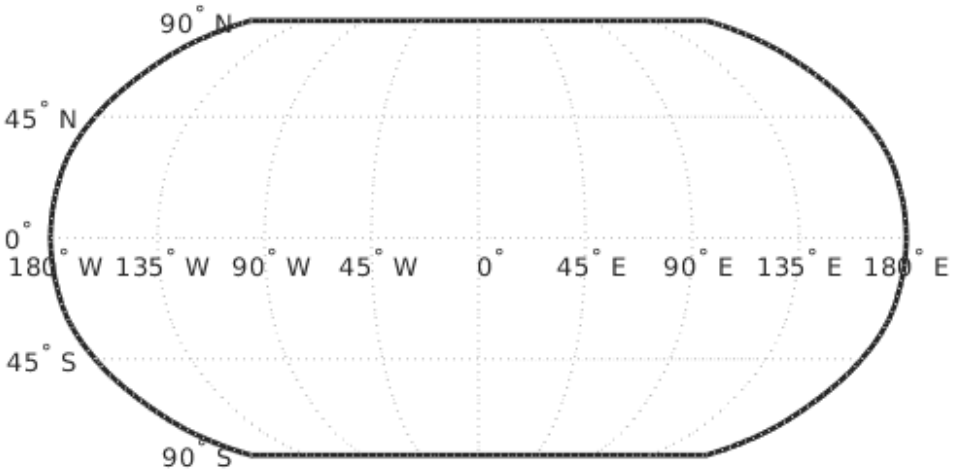
1. * Matlab 的地图工具箱（Mapping Toolbox）：* Matlab 的官方地图工具箱提供了广泛的地图制作功能，还提供了许多其他功能，可用于地理空间数据分析。使用“制图工具箱”，您可以制作出几乎所有您梦寐以求的地图，并且即使您可以使它看起来非常好。但是，确切地找出如何制作自己的梦想的地图可能并不容易。此外，地图工具箱功能有时也会非常慢。反对使用地图工具箱的最大争议也许是昂贵，而且它不是 Matlab 随附的标准工具箱之一。因此，即使您的机构拥有地图工具箱的许可证，您也无法确定合作者是否可以访问它，也很难在线寻求帮助，并且您的未来雇主很可能没有许可证。由于这些原因，最好不要对地图工具箱建立任何依赖关系。
2. * M_Map:* Rich Pawlowicz 的 M_M_Map 是 Matlab 地图工具箱的免费替代品。M_Map 的编写良好，文档齐全，维护良好，并且与 Octave 兼容。它可以生成具有出版质量的地图，唯一的真正缺点是要下载的东西还多。

地图工具箱（Mapping Toolbox）

要使用 Matlab 的地图工具箱（Mapping Toolbox） 制作地图，首先要初始化地图投影，然后以与上述未投影示例非常相似的方式使用绘图函数。这里唯一的区别是，不是使用 `plot(lon,lat)`、`pcolor(lon,lat,z)`等，而是使用 `plotm(lat,lon)`、`pcolorm(lat,lon,z)`等. 换句话说，将字母 `m` 放在您将在未投影坐标中使用的函数名称的末尾，并确保顺序是 `lat,lon` 而不是 `lon,lat`。

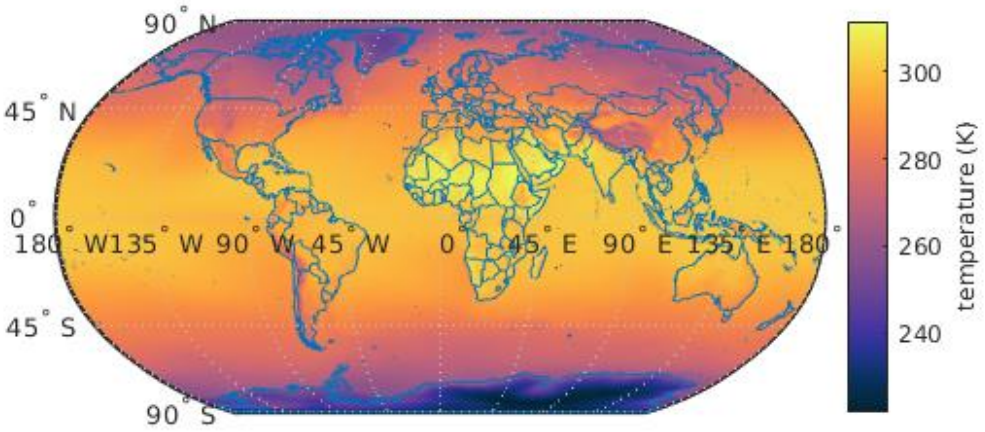
如果您的工作专注于特定区域，则在调用 worldmap 时按名称指定该区域或提供您要初始化的地图的地理范围，worldmap 函数将自动计算出合适的投影。对于这个例子，绘制整个世界：

```
figure

worldmap('world') % 初始化世界地图
```



现在按照我们对上面 pcolor 映射所做的相同步骤，但使用 pcolorm 和 borderm：

```
pcolorm(Lat,Lon,mean(T,3))

cmocean thermal
bordersm
xlabel 'degrees longitude'
ylabel 'degrees latitude'
cb = colorbar;
ylabel(cb,'temperature (K)')
```



不幸的是，我们不能像在未投影的示例中那样使用 quiverm 来显示风向量，因为 quiverm 有一些奇怪的缩放问题。（注意：quiverm 还混合了 u 和 v 向量分量！）因此，我们已经遇到了许多使 Matlab 的地图工具箱难以使用的小特性之一。

M_Map

M_Map 网站上有许多示例，说明如何使用其程序在 Matlab 中创建地图。这是一个很棒的程序包，值得一试。

地球

CDT 提供了一套用于在地球上绘图的功能。地球可能不是显示某些类型数据的最佳选择，因为世界上有一半的数据总是在地球的另一端，而且您能看到的大部分内容在视野范围内都失真了。地球在计算上也很昂贵，因为一切都需要在三维空间中绘制。然而，消除地球仪的缺点，那么对于探索数据集之间的关系仍然非常有用，因为您可以在闲暇时手动旋转地球，并对数据集之间的真实全球空间关系产生直觉。地球还可以用于制作可以吸引观众的动态、引人注目的图像。

制作地球仪的最快方法是使用 globeimage 函数。这是在行动：

```
figure

globeimage
```

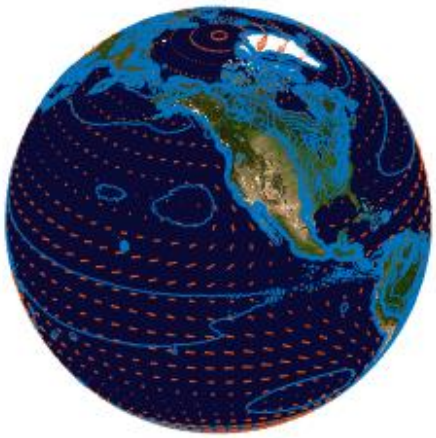
如果您正在跟进，您可以单击地球并将其移动到您希望的任何视角。您还可以开始使用地球绘图功能向地球添加更多数据层。地球绘图函数主要遵循 globalplot、globepcolor、globeborders 等形式。有关地球系列绘图函数的完整列表，请参阅 CDT 内容页面。

随着 globeimage 初始化，您现在可以添加其他数据，如表面等压线，就像这样：

```
globecontour(Lat,Lon,mean(sp,3),(950:10:1030)*100)
```

现在用 globequiver 添加平均风，如下所示：

```
globequiver(Lat,Lon,mean(u10,3),mean(v10,3),'density',75)
```



致谢

该教程由 Chad A. Greene 于 2018 年 12 月完成。

样本数据集

CDT 附带了几个数据集，可用于测试脚本或创建示例。它们如下：

- altimetry_example.h5: 来自 NASA 机载地形测绘仪的表面标高，用作以下文档的示例数据集： h5struct.
- bluemarble.png: 来自 NASA 的地球真彩图像。该图片由 earthimage 绘制。
- borderdata.mat: 用 borders,bordersm, labelborders, labelbordersm, 和 globeborders 绘制的各国和美国的边界。
- BROKE_cruise_odv.txt: 包含来自南极洲海岸的 CTD 模型的海洋温度、盐度和氧气。
- Curie_Depth.xyz: 来自 Martos, 2017 的网格化南极居里深度。
- distance2coast.mat: 使用 land_mask.mat 数据集上的大圆距离计算的到海岸线的全球距离网格。由 dist2coast 调用。
- ERA_interim_2017.nc: ECMWF 天气月平均温度、风、地表气压和降水。有关如何加载和绘制此数据集的说明，请参阅 NetCDF 教程。
- example_ctd.mat: transect 文档中使用的示例海洋剖面数据。
- global_sst.mat: 海面温度的 0.75 度全球网格。
- global_topography.mat: 来自由 topo_interp 调用的世界数字高程模型 (ETOPO5) 的 5 分钟（1/12 度）全球地形网格。
- land_mask.mat: 指示陆地或海洋区域的 1/8 度二进制掩膜。该数据集由 island 函数调用。
- mlo_daily_CO2.mat: 来自夏威夷莫纳罗亚山准连续测量的大气二氧化碳干空气摩尔分数。来自 NOAA。
- nao_slp_data.mat: 来自气候研究单位的带状平均海平面压力的观测，用作 nao 函数的示例。
- ncep-ncar.mat: spei 函数文档中描述的温度和降水再分析数据。
- north_atlantic_sst.mat: 从 1870 年到 2017 年，横跨北大西洋的 2 度每月网格化海面温度。
- orbit91.txt: 来自 Berger A. and Loutre M.F., 1991 的地球轨道参数，用作 daily_insolation 示例。
- orbital_parameter_data.mat:由伊恩·埃森曼和彼得·哈伯斯，哈佛大学于 2006 年 8 月编写，由 daily_insolation 函数所使用。
- pacific_sst.mat: 覆盖太平洋部分地区的 67 年时间序列网格化的每月海面温度。该数据集作为 eof 函数文档的一部分进行分析。
- pacific_wind.mat: 覆盖太平洋一部分的 1/8 度网格海面温度和风。该数据集作为 ekman 函数文档的一部分进行分析。
- sam_slp_data.mat G. Marshall 对 12 个台站的平均纬向平均表面压力，用作 sam 函数文档中的示例数据。
- seaice_extent.mat:来自 NSIDC 的北半球和南半球海冰范围的 38 年每日时间序列。该数据集被绘制为 spiralplot 和函数文档的一部分。
- sodb_example.mat: 来自南大洋数据库的三维位势温度网格。数据集在 bottom 文档的末尾有描述。
- xkcd_rgb_data.mat: 出现在计算机屏幕上的 950 种颜色的名称和 RGB 值。该数据来自 XKCD 的 Randall Munroe 进行的一项令人印象深刻的彻底调查，并由 CDT 函数 rgb 调用。

引用 CDT

请引用我们的论文！（二爷只要一个致谢就行了）

Chad A. Greene, Kaustubh Thirumalai, Kelly A. Kearney, Jose Miguel Delgado, Wolfgang Schwanghart, Natalie S. Wolfenbarger, Kristen M. Thyng, David E. Gwyther, Alex S. Gardner, and Donald D. Blankenship. The Climate Data Toolbox for MATLAB. *Geochemistry, Geophysics, Geosystems* 2019. doi:10.1029/2019GC008392