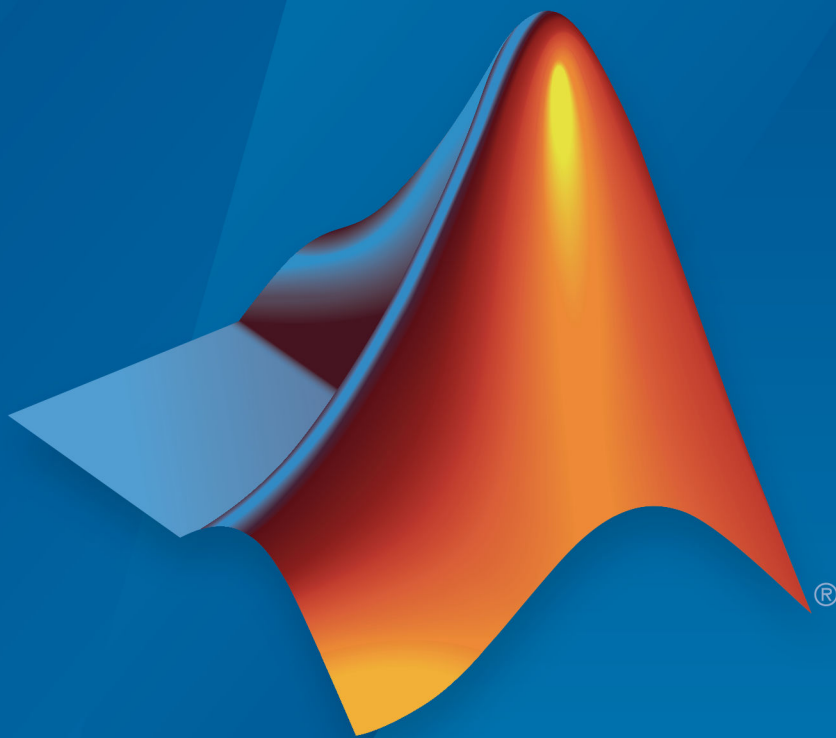


MATLAB®

数据分析



MATLAB®

R2019a

 **MathWorks®**

如何联系 MathWorks



最新动态：www.mathworks.com
销售和服务：www.mathworks.com/sales_and_services
用户社区：www.mathworks.com/matlabcentral
技术支持：www.mathworks.com/support/contact_us



电话：010-59827000



迈斯沃克软件 (北京) 有限公司
北京市朝阳区望京东园四区 6 号楼
北望金辉大厦 16 层 1604

MATLAB® 数据分析

© COPYRIGHT 2005–2019 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

商标

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

专利

MathWorks products are protected by one or more U.S. patents. Please see www.mathworks.com/patents for more information.

修订历史记录

2005 年 9 月	仅限在线版本	MATLAB 7.1 (版本 14SP3) 中的新增内容
2006 年 3 月	仅限在线版本	MATLAB 7.2 (版本 2006a) 中的修订内容
2006 年 9 月	仅限在线版本	MATLAB 7.3 (版本 2006b) 中的修订内容
2007 年 3 月	仅限在线版本	MATLAB 7.4 (版本 2007a) 中的修订内容
2007 年 9 月	仅限在线版本	MATLAB 7.5 (版本 2007b) 中的修订内容
2008 年 3 月	仅限在线版本	MATLAB 7.6 (版本 2008a) 中的修订内容
2008 年 10 月	仅限在线版本	MATLAB 7.7 (版本 2008b) 中的修订内容
2009 年 3 月	仅限在线版本	MATLAB 7.8 (版本 2009a) 中的修订内容
2009 年 9 月	仅限在线版本	MATLAB 7.9 (版本 2009b) 中的修订内容
2010 年 3 月	仅限在线版本	MATLAB 7.10 (版本 2010a) 中的修订内容
2010 年 9 月	仅限在线版本	MATLAB 7.11 (版本 2010b) 中的修订内容
2011 年 4 月	仅限在线版本	MATLAB 7.12 (版本 2011a) 中的修订内容
2011 年 9 月	仅限在线版本	MATLAB 7.13 (版本 2011b) 中的修订内容
2012 年 3 月	仅限在线版本	MATLAB 7.14 (版本 2012a) 中的修订内容
2012 年 9 月	仅限在线版本	MATLAB 8.0 (版本 2012b) 中的修订内容
2013 年 3 月	仅限在线版本	MATLAB 8.1 (版本 2013a) 中的修订内容
2013 年 9 月	仅限在线版本	MATLAB 8.2 (版本 2013b) 中的修订内容
2014 年 3 月	仅限在线版本	MATLAB 8.3 (版本 2014a) 中的修订内容
2014 年 10 月	仅限在线版本	MATLAB 8.4 (版本 2014b) 中的修订内容
2015 年 3 月	仅限在线版本	MATLAB 8.5 (版本 2015a) 中的修订内容
2015 年 9 月	仅限在线版本	MATLAB 8.6 (版本 2015b) 中的修订内容
2016 年 3 月	仅限在线版本	MATLAB 9.0 (版本 2016a) 中的修订内容
2016 年 9 月	仅限在线版本	MATLAB 9.1 (版本 2016b) 中的修订内容
2017 年 3 月	仅限在线版本	MATLAB 9.2 (版本 2017a) 中的修订内容
2017 年 9 月	仅限在线版本	MATLAB 9.3 (版本 2017b) 中的修订内容
2018 年 3 月	仅限在线版本	MATLAB 9.4 (版本 2018a) 中的修订内容
2018 年 9 月	仅限在线版本	MATLAB 9.5 (版本 2018b) 中的修订内容
2019 年 3 月	仅限在线版本	MATLAB 9.6 (版本 2019a) 中的修订内容

导入和导出数据	1-2
将数据导入工作区	1-2
从工作区导出数据	1-2
绘制数据	1-3
简介	1-3
从文本文件加载数据并进行绘图	1-3
MATLAB 中的缺失数据	1-6
数据平滑和离群值检测	1-11
不一致的数据	1-24
滤波数据	1-26
滤波器差分方程	1-26
交通流量数据的移动平均值滤波器	1-26
修改数据振幅	1-27
使用卷积对数据进行平滑处理	1-31
去除数据的线性趋势	1-35
简介	1-35
从数据中去除线性趋势	1-35
描述性统计量	1-39
用于计算描述性统计量的函数	1-39
示例：使用 MATLAB 数据统计信息	1-41

2

- 线性相关性 2-2
 - 简介 2-2
 - 协方差 2-2
 - 相关系数 2-3
- 线性回归 2-5
 - 简介 2-5
 - 简单线性回归 2-6
 - 残差与拟合优度 2-10
 - 用 Curve Fitting Toolbox 函数拟合数据 2-13
- 交互式拟合 2-15
 - 基本拟合用户界面 2-15
 - 基本拟合准备 2-15
 - 打开基本拟合用户界面 2-16
 - 示例：使用基本拟合用户界面 2-17
- 以编程方式拟合 2-32
 - 适用于多项式模型的 MATLAB 函数 2-32
 - 带非多项式项的线性模型 2-37
 - 多次回归 2-39
 - 以编程方式拟合 2-40

3

- 什么是时序？ 3-2
- 时序对象 3-3
 - 时序的类型及其用途 3-3
 - 时序数据样本 3-3
 - 示例：时序对象和方法 3-5
 - 时序构造函数 3-16
 - 时序集合构造函数 3-17

数据处理

- “导入和导出数据”（第 1-2 页）
- “绘制数据”（第 1-3 页）
- “MATLAB 中的缺失数据”（第 1-6 页）
- “数据平滑和离群值检测”（第 1-11 页）
- “不一致的数据”（第 1-24 页）
- “滤波数据”（第 1-26 页）
- “使用卷积对数据进行平滑处理”（第 1-31 页）
- “去除数据的线性趋势”（第 1-35 页）
- “描述性统计量”（第 1-39 页）

导入和导出数据

本节内容
“将数据导入工作区”（第 1-2 页）
“从工作区导出数据”（第 1-2 页）

将数据导入工作区

分析数据的第一步是将其导入 MATLAB 工作区。有关从特定文件格式导入数据的信息，请参阅“导入数据的方法”。

从工作区导出数据

分析数据时，您可能会创建新变量或修改导入的变量。您可以将变量从 MATLAB 工作区中导出为各种文件格式，包括基于字符的格式和二进制格式。例如，您可以创建包含您的数据的 HDF 和 Microsoft® Excel® 文件。有关详细信息，请参阅有关“支持的导入和导出文件格式”的文档。

绘制数据

本节内容
“简介”（第 1-3 页）
“从文本文件加载数据并进行绘图”（第 1-3 页）

简介

将数据导入 MATLAB 工作区后，最好绘制数据以便研究其特性。通过对数据进行探索性绘图，您可以识别不连续性和潜在离群值以及感兴趣的区域。

MATLAB 图窗窗口显示绘图。有关图窗窗口的完整说明，请参阅“MATLAB 绘图类型”。它还介绍了可用于编辑和自定义 MATLAB 图形的各种交互式工具。

从文本文件加载数据并进行绘图

此示例使用以空格分隔的文本文件 `count.dat` 中的样本数据。该文件包含三组以小时计的交通流量，分别记录三个不同城镇十字路口在 24 小时内的交通流量。文件中的每个数据列表示一个十字路口的数据。

加载 `count.dat` 数据

使用 `load` 函数将数据导入工作区中。

`load count.dat`

加载此数据将在 MATLAB 工作区中创建一个称为 `count` 的 24×3 矩阵。

获取数据矩阵的大小。

`[n,p] = size(count)`

`n = 24`

`p = 3`

`n` 表示行数，`p` 表示列数。

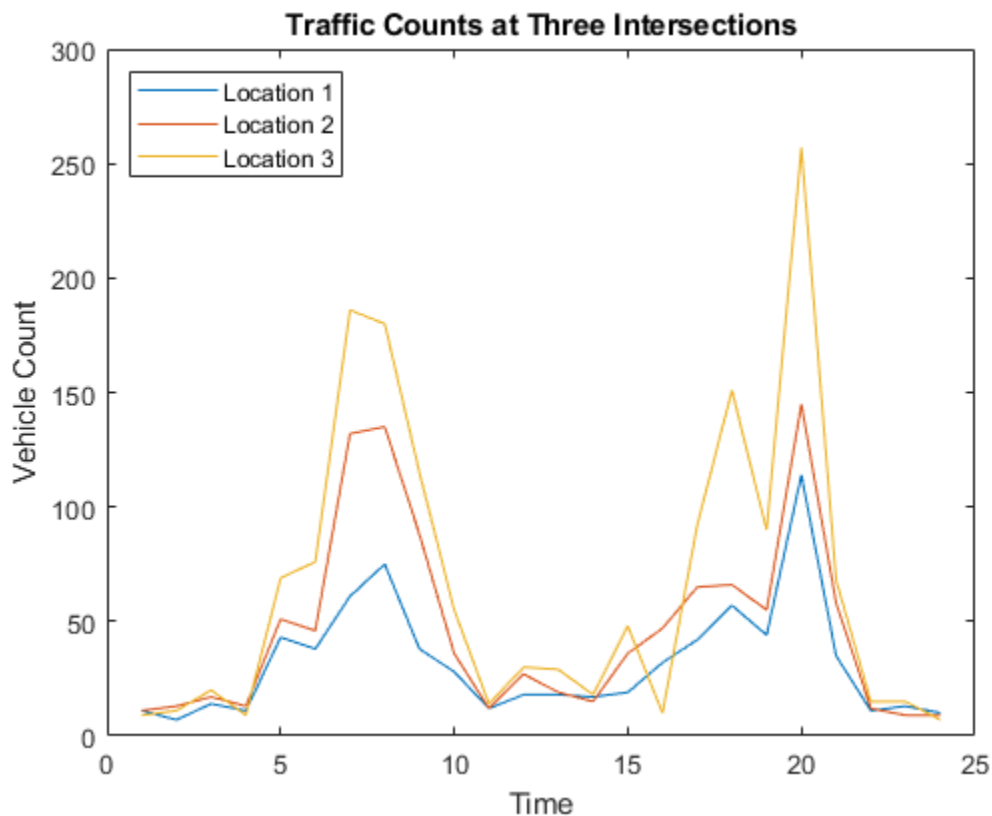
绘制 `count.dat` 数据

创建时间向量 `t`，其中包含从 1 到 `n` 的整数。

```
t = 1:n;
```

将数据绘制为时间的函数，并对图进行注释。

```
plot(t,count),  
legend('Location 1','Location 2','Location 3','Location','NorthWest')  
xlabel('Time'), ylabel('Vehicle Count')  
title('Traffic Counts at Three Intersections')
```



另请参阅

[legend](#) | [load](#) | [plot](#) | [size](#) | [title](#) | [xlabel](#) | [ylabel](#)

详细信息

- “MATLAB 绘图类型”

MATLAB 中的缺失数据

处理缺失数据是数据预处理中的常见任务。有时缺失值表示数据中有意义的事件，但它们通常表示不可靠或不可用的数据点。对于以上两种情况，MATLAB® 都提供了许多处理缺失数据的选项。

创建并组织缺失数据

MATLAB 中缺失值的形式取决于数据类型。例如，数值数据类型（例如 `double`）使用 `NaN`（非数值）表示缺失值。

```
x = [NaN 1 2 3 4];
```

您也可以使用 `missing` 值表示缺失数值数据或其他类型的数据，例如 `datetime`、`string` 和 `categorical`。MATLAB 自动将 `missing` 值转换为数据的原生类型。

```
xDouble = [missing 1 2 3 4]
```

```
xDouble = 1×5
```

```
NaN    1    2    3    4
```

```
xDatetime = [missing datetime(2014,1:4,1)]
```

```
xDatetime = 1×5 datetime array
```

```
Columns 1 through 3
```

```
NaT           01-Jan-2014 00:00:00  01-Feb-2014 00:00:00
```

```
Columns 4 through 5
```

```
01-Mar-2014 00:00:00  01-Apr-2014 00:00:00
```

```
xString = [missing "a" "b" "c" "d"]
```

```
xString = 1×5 string array
```

```
<missing>  "a"  "b"  "c"  "d"
```

```
xCategorical = [missing categorical({'cat1' 'cat2' 'cat3' 'cat4'})]
```

```
xCategorical = 1x5 categorical array
<undefined>   cat1   cat2   cat3   cat4
```

数据集可能包含要作为缺失数据处理的值，但这些值不是 MATLAB 中的标准 MATLAB 缺失值，例如 NaN。您可以使用 `standardizeMissing` 函数将这些值转换为该数据类型的标准缺失值。例如，除 NaN 之外，将 4 也处理为缺失的 `double` 值。

```
xStandard = standardizeMissing(xDouble,[4 NaN])
```

```
xStandard = 1x5
NaN   1   2   3 NaN
```

假设您要将缺失值保留为数据集的一部分，但将其与其余数据隔离。您可以使用几个 MATLAB 函数控制缺失值的位置，然后再进一步处理。例如，将 `'MissingPlacement'` 选项与 `sort` 函数结合使用，可将 NaN 移动到数据的末尾。

```
xSort = sort(xStandard,'MissingPlacement','last')
```

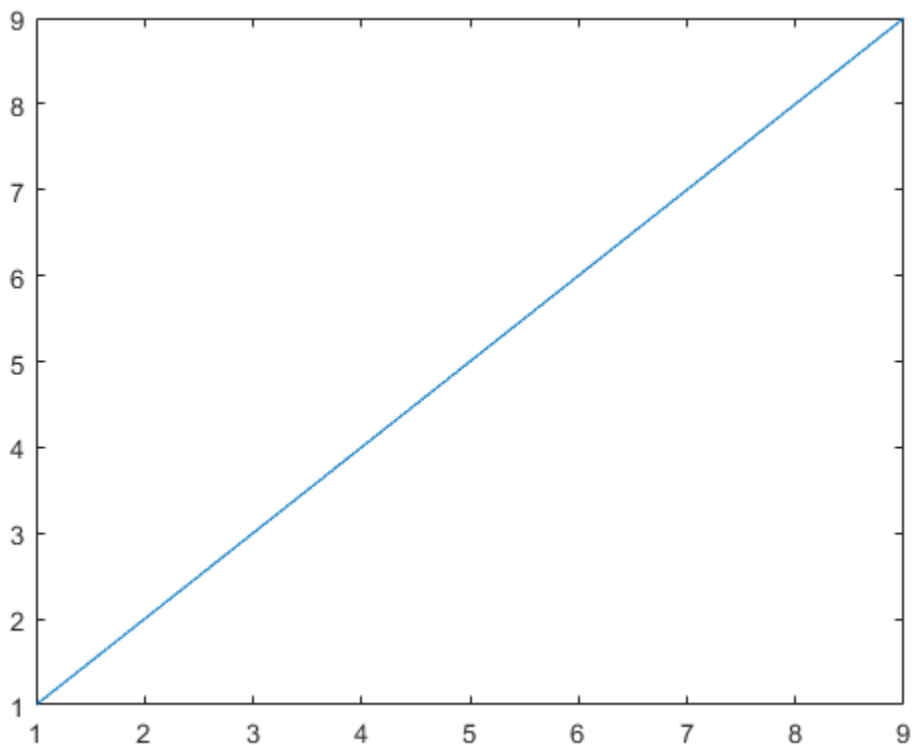
```
xSort = 1x5
1   2   3 NaN NaN
```

查找、替换和忽略缺失数据

即使您没有在 MATLAB 中显式创建缺失值，在导入现有数据或使用数据进行计算时也会显示缺失值。如果您不知道数据中有缺失值，后续计算或分析可能会产生误导。

例如，如果您对某向量绘图而不知道其中包含 NaN 值，则不会显示 NaN，因为 `plot` 函数会忽略它而正常绘制其余点。

```
nanData = [1:9 NaN];
plot(1:10,nanData)
```



但是，如果您计算数据的平均值，则结果为 **NaN**。在这种情况下，最好事先知道数据包含 **NaN**，然后在计算平均值之前选择忽略或删除它。

```
meanData = mean(nanData)
```

```
meanData = NaN
```

在数据中查找 **NaN** 的一种方法是使用 **isnan** 函数，该函数返回一个逻辑数组，指示 **NaN** 值的位置。

```
TF = isnan(nanData)
```

```
TF = 1x10 logical array
```

```
0 0 0 0 0 0 0 0 0 1
```

同样，`ismissing` 函数返回多个数据类型的数据中缺失值的位置。

```
TFdouble = ismissing(xDouble)
```

```
TFdouble = 1x5 logical array
```

```
1 0 0 0 0
```

```
TFdatetime = ismissing(xDatetime)
```

```
TFdatetime = 1x5 logical array
```

```
1 0 0 0 0
```

假设您正在处理由多个数据类型的变量组成的表或时间表。您可以通过调用一次 `ismissing` 找到所有缺失值，而不管其类型如何。

```
xTable = table(xDouble',xDatetime',xString',xCategorical')
```

```
xTable=5x4 table
```

	Var1	Var2	Var3	Var4
	NaN	NaT	<missing>	<undefined>
1	01-Jan-2014 00:00:00	"a"	cat1	
2	01-Feb-2014 00:00:00	"b"	cat2	
3	01-Mar-2014 00:00:00	"c"	cat3	
4	01-Apr-2014 00:00:00	"d"	cat4	

```
TF = ismissing(xTable)
```

```
TF = 5x4 logical array
```

```
1 1 1 1
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
```

缺失值可以表示不可用于处理或分析的数据。使用 `fillmissing` 将缺失值替换为另一个值，或者使用 `rmmissing` 删除全部缺失值。

```
xFill = fillmissing(xStandard,'constant',0)
```

```
xFill = 1×5
```

```
0    1    2    3    0
```

```
xRemove = rmmissing(xStandard)
```

```
xRemove = 1×3
```

```
1    2    3
```

许多 MATLAB 函数都可以忽略缺失值，您不必首先显式定位、填充或删除它们。例如，如果计算含有 NaN 值的向量的和，则结果为 NaN。但是，您可以结合使用 `sum` 函数和 `'omitnan'` 选项来直接忽略和中的 NaN。

```
sumNan = sum(xDouble)
```

```
sumNan = NaN
```

```
sumOmitnan = sum(xDouble,'omitnan')
```

```
sumOmitnan = 10
```

另请参阅

[ismissing](#) | [fillmissing](#) | [standardizeMissing](#) | [missing](#)

相关示例

- “清除表中的杂乱数据和缺失数据”

数据平滑和离群值检测

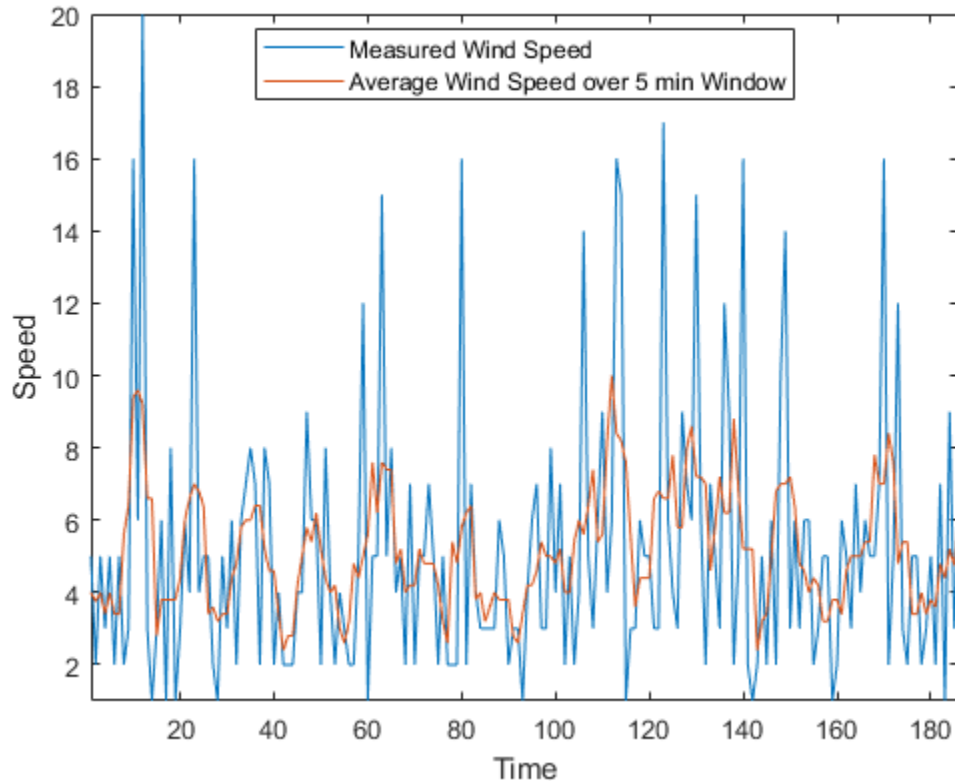
数据平滑指用于消除数据中不需要的噪声或行为的技术，而离群值检测用于标识与其余数据显著不同的数据点。

移动窗口方法

移动窗口方法是分批处理数据的方式，通常是为了从统计角度表示数据中的相邻点。移动平均值是一种常见的数据平滑技术，它沿着数据滑动窗口，同时计算每个窗口内点的均值。这可以帮助消除从一个数据点到下一个数据点的非显著变化。

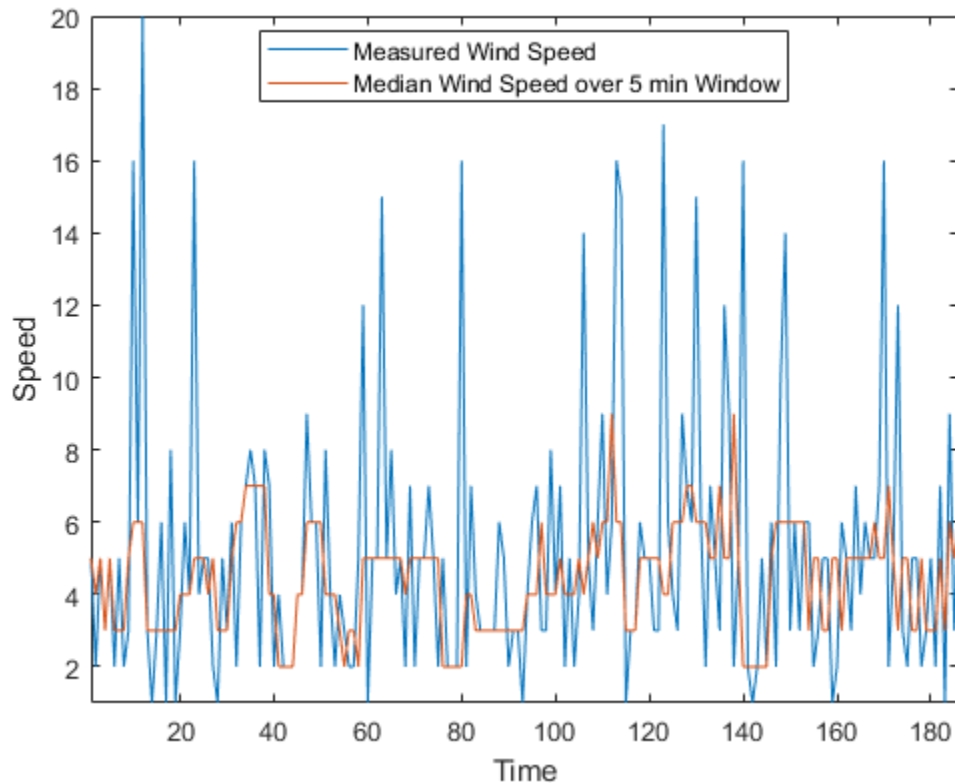
例如，假设每分钟测量一次风速，持续约 3 小时。使用 `movmean` 函数和 5 分钟的窗口大小可去除高速阵风。

```
load windData.mat
mins = 1:length(speed);
window = 5;
meanspeed = movmean(speed,window);
plot(mins,speed,mins,meanspeed)
axis tight
legend('Measured Wind Speed','Average Wind Speed over 5 min Window','location','best')
xlabel('Time')
ylabel('Speed')
```



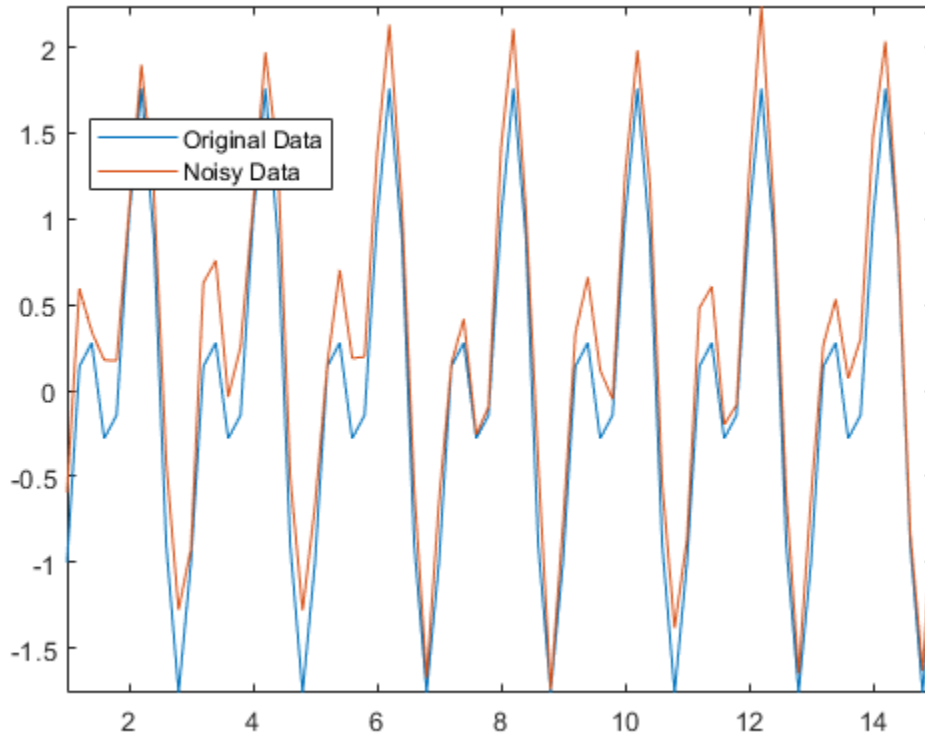
同样，您可以使用 `movmedian` 函数计算滑动窗口中的风速中位数。

```
medianspeed = movmedian(speed>window);  
plot(mins,speed,mins,medianspeed)  
axis tight  
legend('Measured Wind Speed','Median Wind Speed over 5 min Window','location','best')  
xlabel('Time')  
ylabel('Speed')
```



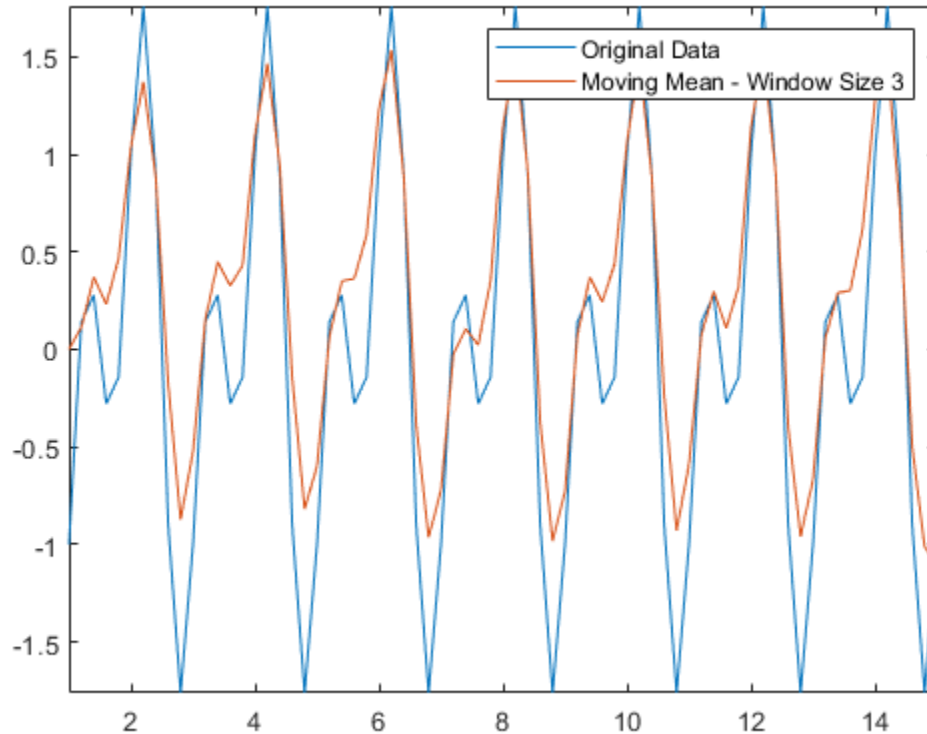
并非所有数据都适合用移动窗口方法进行平滑处理。例如，创建插入了随机噪声的正弦信号。

```
t = 1:0.2:15;
A = sin(2*pi*t) + cos(2*pi*0.5*t);
Anoise = A + 0.5*rand(1,length(t));
plot(t,A,t,Anoise)
axis tight
legend('Original Data','Noisy Data','location','best')
```



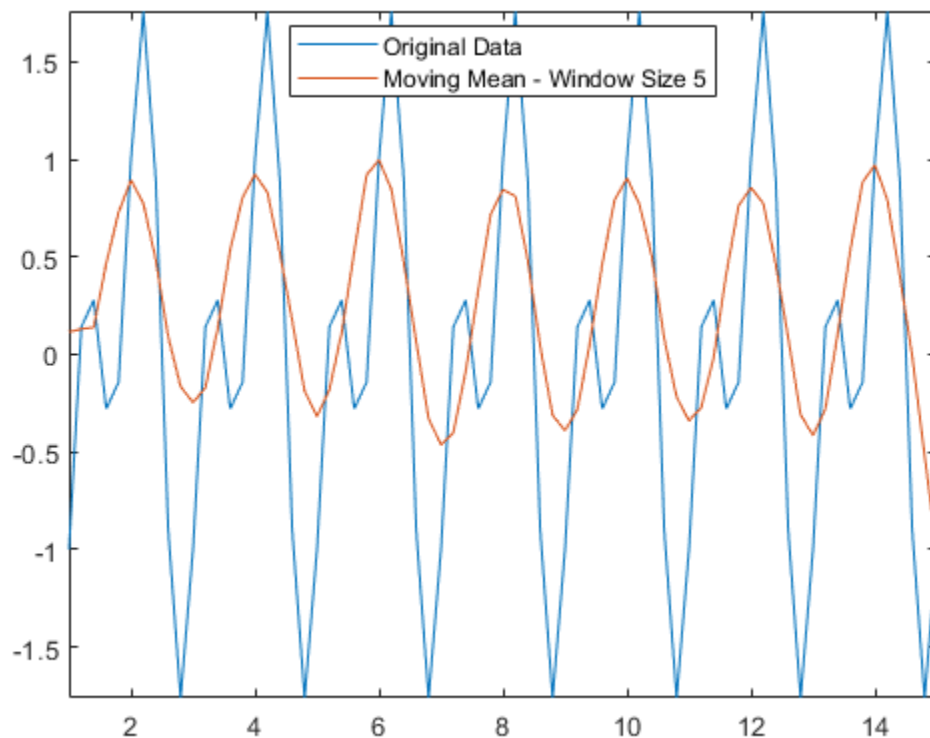
使用移动均值和大小为 3 的窗口对含噪数据进行平滑处理。

```
window = 3;  
Amean = movmean(Anoise,window);  
plot(t,A,t,Amean)  
axis tight  
legend('Original Data','Moving Mean - Window Size 3')
```



移动均值方法可获得数据的大致形状，但不能非常准确地捕获波谷（局部最小值）。由于波谷点在每个窗口中两个较大的邻点之间，因此均值不是那些点的理想近似值。如果使窗口大小变大，均值将完全消除较短的波峰。对于这种类型的数据，您可能需要考虑其他平滑技术。

```
Amean = movmean(Anoise,5);  
plot(t,A,t,Amean)  
axis tight  
legend('Original Data','Moving Mean - Window Size 5','location','best')
```

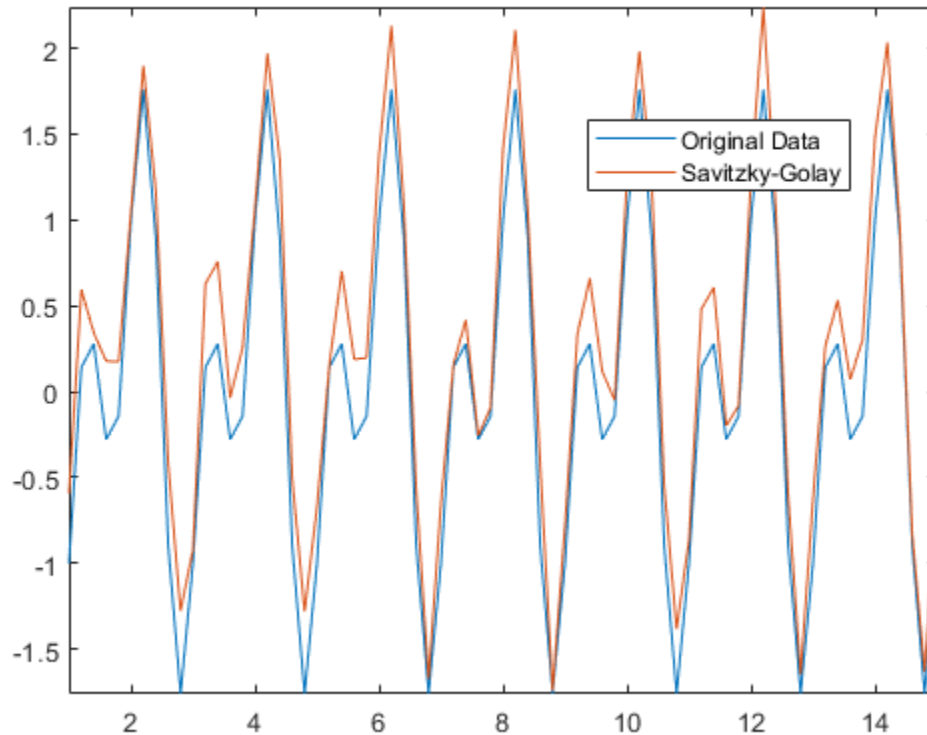


常见的平滑方法

`smoothdata` 函数提供几种平滑选项，如 Savitzky-Golay 方法，这是一种常用的信号处理平滑技术。默认情况下，`smoothdata` 根据数据为方法选择最佳估计窗口大小。

使用 Savitzky-Golay 方法可对噪声信号 `Anoise` 进行平滑处理，并输出它使用的窗口大小。与 `movmean` 相比，该方法可提供更好的波谷近似值。

```
[Asgolay,window] = smoothdata(Anoise,'sgolay');  
plot(t,A,t,Asgolay)  
axis tight  
legend('Original Data','Savitzky-Golay','location','best')
```

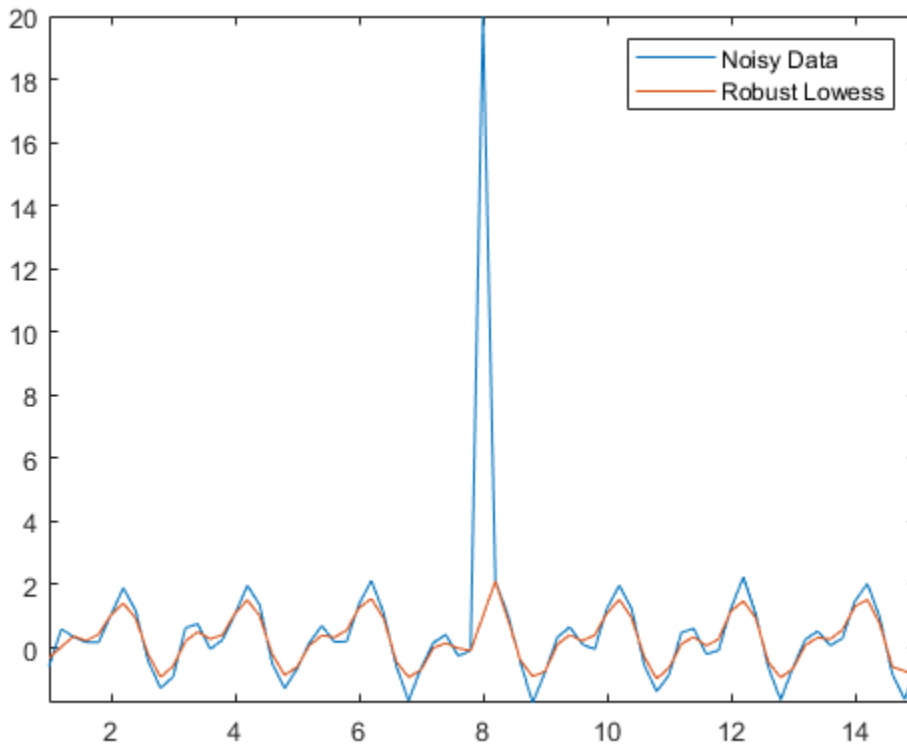


window

`window = 3`

稳健的 Lowess 方法是另一种平滑方法，尤其适用于同时包含噪声和离群值的数据。在含噪数据中插入离群值，并使用稳健的 Lowess 方法对数据进行平滑处理，从而消除离群值。

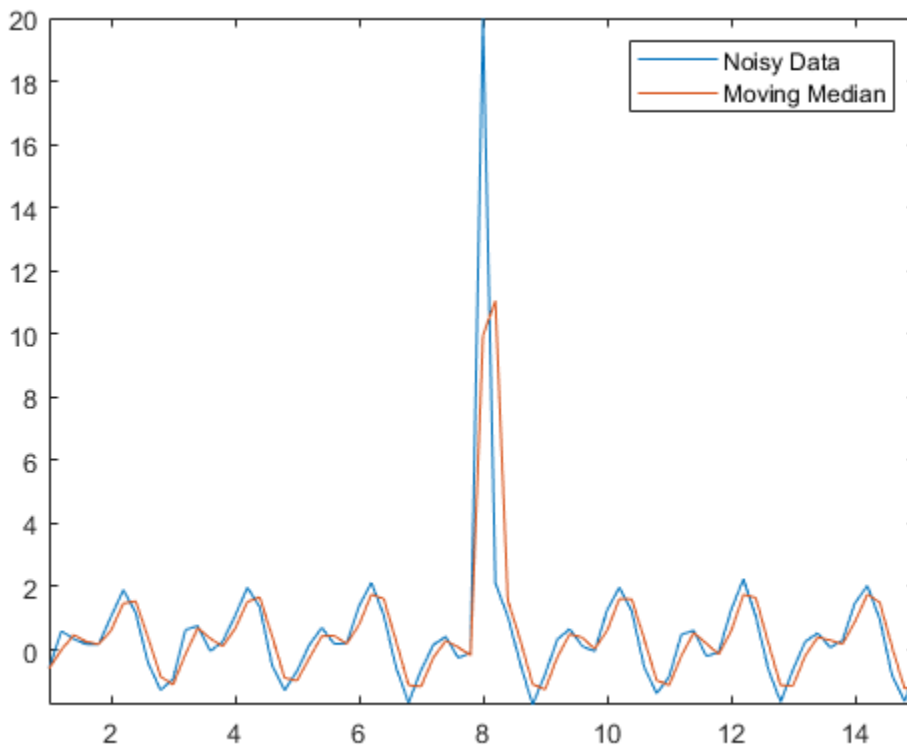
```
Anoise(36) = 20;
Arlowess = smoothdata(Anoise,'rlowess',5);
plot(t,Anoise,t,Arlowess)
axis tight
legend('Noisy Data','Robust Lowess')
```



检测离群值

数据中的离群值可能使数据处理结果和其他计算量严重失真。例如，如果您尝试用移动平均值方法对包含离群值的数据进行平滑处理，则可能得到误导性的波峰或波谷。

```
Amedian = smoothdata(Anoise,'movmedian');  
plot(t,Anoise,t,Amedian)  
axis tight  
legend('Noisy Data','Moving Median')
```

当检测到离群值时，`isoutlier` 函数返回逻辑值 1。验证 `Anoise` 中离群值的索引和值。

```
TF = isoutlier(Anoise);
ind = find(TF)
```

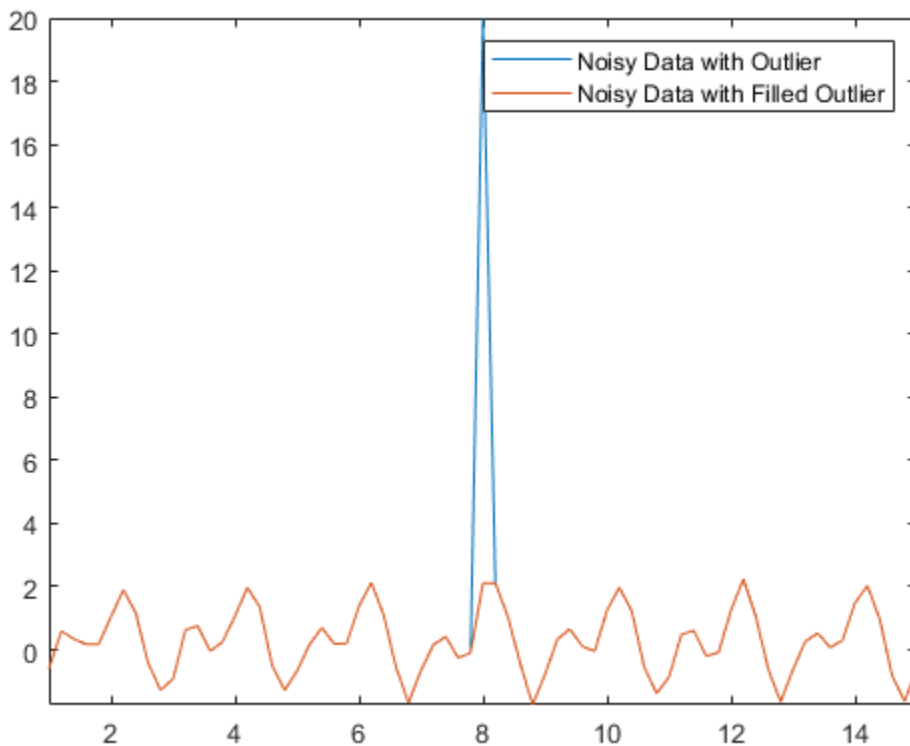
```
ind = 36
```

```
Aoutlier = Anoise(ind)
```

```
Aoutlier = 20
```

您可以使用 `filloutliers` 函数通过指定填充方法来替换数据中的离群值。例如，用紧挨 `Anoise` 中离群值右侧的邻点值填充该离群值。

```
Afill = filloutliers(Anoise,'next');  
plot(t,Anoise,t,Afill)  
axis tight  
legend('Noisy Data with Outlier','Noisy Data with Filled Outlier')
```

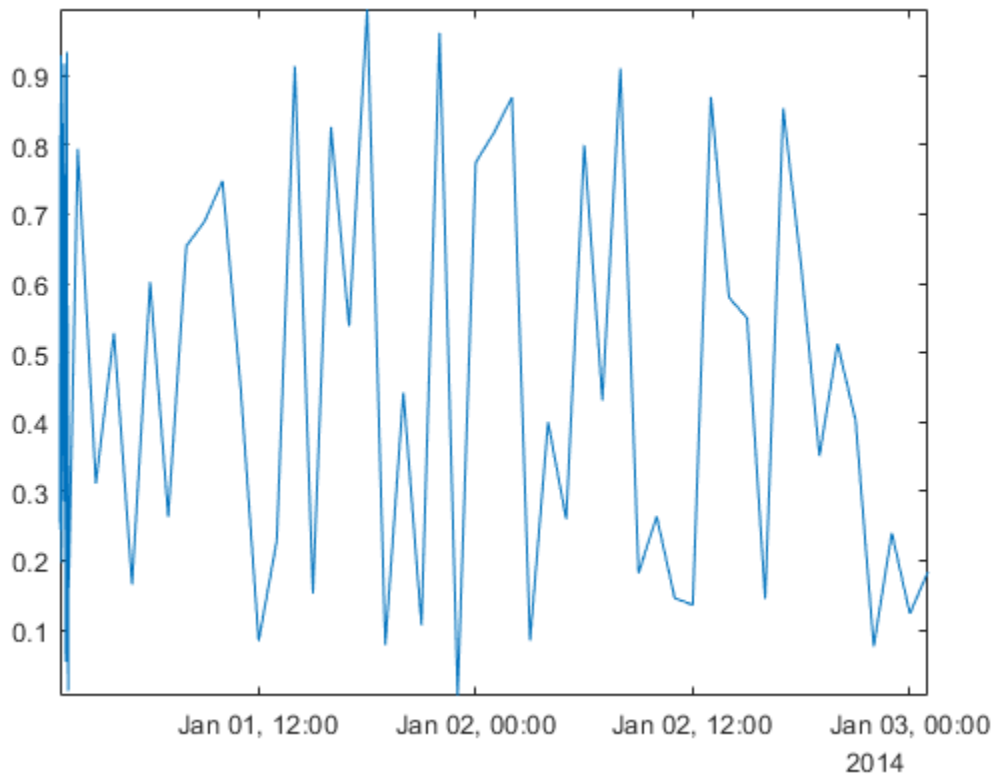


非均匀数据

并非所有数据都由等间隔的点组成，这会影响数据处理的方法。创建一个 **datetime** 向量，其中包含 **Airreg** 中数据的不规则采样时间。**time** 向量表示了前 30 分钟内每分钟采集一次的样本和两天内每小时采集一次的样本。

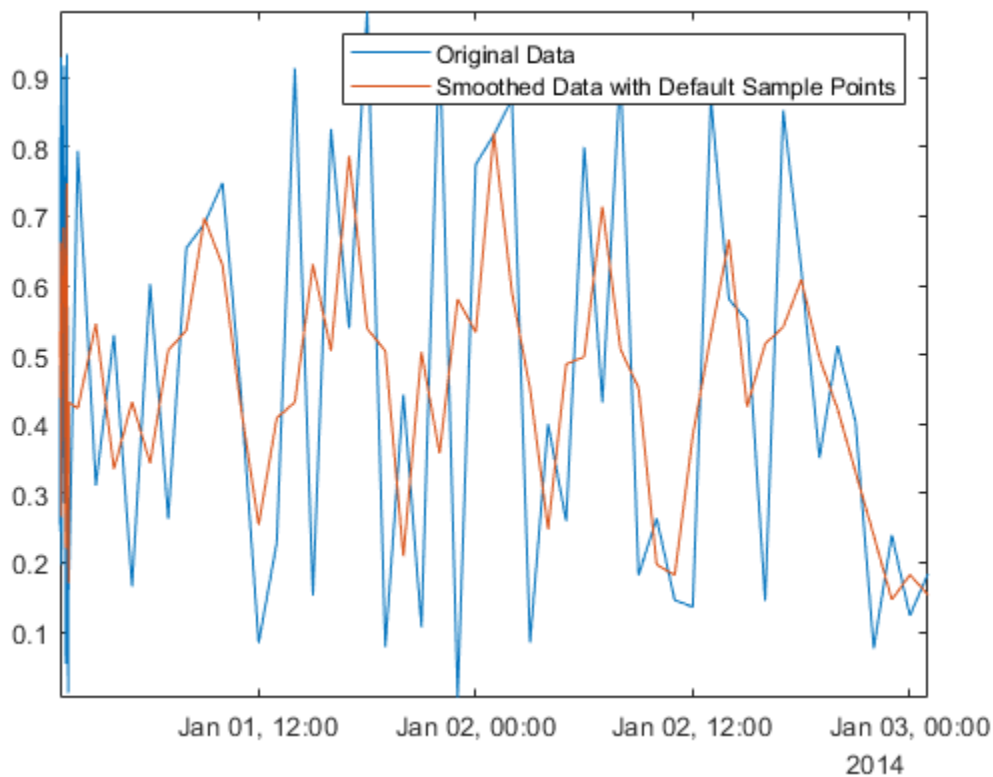
```
t0 = datetime(2014,1,1,1,1,1);  
timeminutes = sort(t0 + minutes(1:30));  
timehours = t0 + hours(1:48);
```

```
time = [timeminutes timehours];
Airreg = rand(1,length(time));
plot(time,Airreg)
axis tight
```



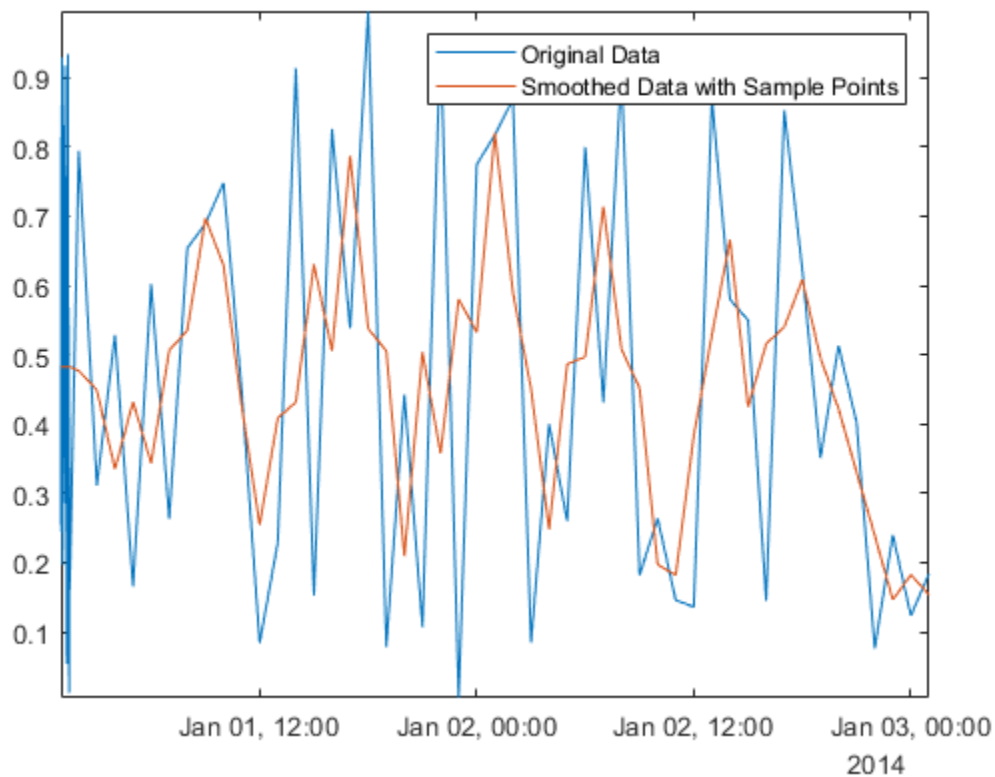
默认情况下，`smoothdata` 按照等间距整数进行平滑处理，在本例中为 1,2,...,78。由于整数时间戳与 `Airreg` 中各点的采样不协调，前半小时的数据在平滑后仍然出现噪声。

```
Adefault = smoothdata(Airreg,'movmean',3);
plot(time,Airreg,time,Adefault)
axis tight
legend('Original Data','Smoothed Data with Default Sample Points')
```



MATLAB® 中的许多数据处理函数（包括 `smoothdata`、`movmean` 和 `filloutliers`）允许您提供样本点，从而确保相对于其采样单位和频率处理数据。要消除 `Airreg` 中前半小时数据的高频变化，请将 `'SamplePoints'` 选项和 `time` 中的时间戳结合使用。

```
Asamplepoints = smoothdata(Airreg,'movmean',hours(3),'SamplePoints',time);  
plot(time,Airreg,time,Asamplepoints)  
axis tight  
legend('Original Data','Smoothed Data with Sample Points')
```



另请参阅

[smoothdata](#) | [isoutlier](#) | [filloutliers](#) | [movmean](#) | [movmedian](#)

相关示例

- “滤波数据”（第 1-26 页）

不一致的数据

当您检查数据绘图时，您可能会发现有些点显著偏离了其他数据。在某些情况下，可合理地将这些点视为离群值，即与其余数据不一致的数据值。

以下示例说明如何从 24×3 矩阵 `count` 中的三个数据集中移除离群值。在本例中，离群值定义为偏离均值超过三倍标准差的值。

小心 除非您确信了解要更正的问题的根源，否则请谨慎对待数据更改。去除离群值对标准差的影响大于对数据均值的影响。删除一个离群值点会导致新标准差变小，从而可能导致其余一些点似乎又成为离群值！

```
% Import the sample data
load count.dat;
% Calculate the mean and the standard deviation
% of each data column in the matrix
mu = mean(count)
sigma = std(count)
```

命令行窗口显示

```
mu =
    32.0000    46.5417    65.5833

sigma =
    25.3703    41.4057    68.0281
```

将偏离均值三倍标准差以上的值视为离群值时，请使用以下语法确定 `count` 矩阵的每列中的离群值数量：

```
[n,p] = size(count);
% Create a matrix of mean values by
% replicating the mu vector for n rows
MeanMat = repmat(mu,n,1);
% Create a matrix of standard deviation values by
% replicating the sigma vector for n rows
SigmaMat = repmat(sigma,n,1);
% Create a matrix of zeros and ones, where ones indicate
% the location of outliers
outliers = abs(count - MeanMat) > 3*SigmaMat;
% Calculate the number of outliers in each column
nout = sum(outliers)
```

该过程返回每列中的离群值数量，如下：

```
nout =  
    1  0  0
```

在 **count** 的第一个数据列中有一个离群值，其他两列中都没有。

要删除包含该离群值的整行数据，请键入

```
count(any(outliers,2),:) = [];
```

此处，当 **outliers** 向量中有任何非零元素时，**any(outliers,2)** 返回 1。参数 2 指定 **any** 继续处理 **count** 矩阵的第二个维度 - 列。

滤波数据

滤波器差分方程

滤波器是一种数据处理技术，可滤掉数据中的高频波动部分使之平滑或从数据中删除特定频率的周期趋势。在 MATLAB 中，`filter` 函数会根据以下差分方程对数据 x 的向量进行滤波，该差分方程描述一个抽头延迟线滤波器。

$$a(1)y(n) = b(1)x(n) + b(2)x(n-1) + \dots + b(N_b)x(n-N_b+1) \\ - a(2)y(n-1) - \dots - a(N_a)y(n-N_a+1)$$

在此方程中， a 和 b 是滤波器系数的向量， N_a 是反馈滤波器阶数， N_b 是前馈滤波器阶数。 n 是 x 的当前元素的索引。输出 $y(n)$ 是 x 和 y 的当前元素和前面元素的线性组合。

`filter` 函数使用指定的系数向量 a 和 b 对输入数据 x 进行滤波。有关描述滤波器的差分方程的详细信息，请参阅 [1]。

交通流量数据的移动平均值滤波器

`filter` 函数是实现移动平均值滤波器的一种方式，它是一种常见的数据平滑技术。

以下差分方程描述一个滤波器，它对关于当前小时和前三个小时的数据的时间相关数据求平均值。

$$y(n) = \frac{1}{4}x(n) + \frac{1}{4}x(n-1) + \frac{1}{4}x(n-2) + \frac{1}{4}x(n-3)$$

导入描述交通流量随时间变化的数据，并将第一列车辆计数分配给向量 x 。

```
load count.dat
x = count(:,1);
```

创建滤波器系数向量。

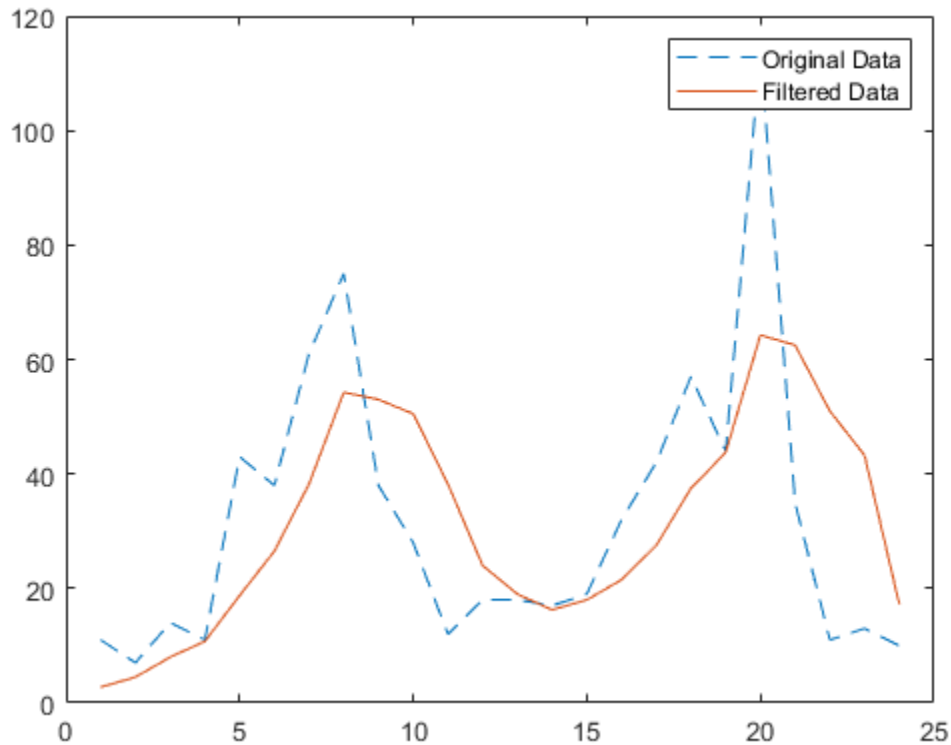
```
a = 1;
b = [1/4 1/4 1/4 1/4];
```

计算数据的 4 小时移动平均值，同时绘制原始数据和滤波后的数据。

```
y = filter(b,a,x);
```



```
t = 1:length(x);
plot(t,x,'-','t,y','-')
legend('Original Data','Filtered Data')
```



修改数据振幅

此示例显示如何通过应用传递函数来修改数据向量的振幅。

在数字信号处理中，滤波器通常由传递函数表示。以下差分方程的 Z 变换

$$a(1)y(n) = b(1)x(n) + b(2)x(n-1) + \dots + b(N_b)x(n-N_b+1) \\ - a(2)y(n-1) - \dots - a(N_a)y(n-N_a+1)$$

是以下传递函数。

$$Y(z) = H(z^{-1})X(z) = \frac{b(1) + b(2)z^{-1} + \dots + b(N_b)z^{-N_b+1}}{a(1) + a(2)z^{-1} + \dots + a(N_a)z^{-N_a+1}}X(z)$$

使用传递函数

$$H(z^{-1}) = \frac{b(z^{-1})}{a(z^{-1})} = \frac{2 + 3z^{-1}}{1 + 0.2z^{-1}}$$

修改 `count.dat` 中数据的振幅。

加载数据并将第一列分配到向量 `x`。

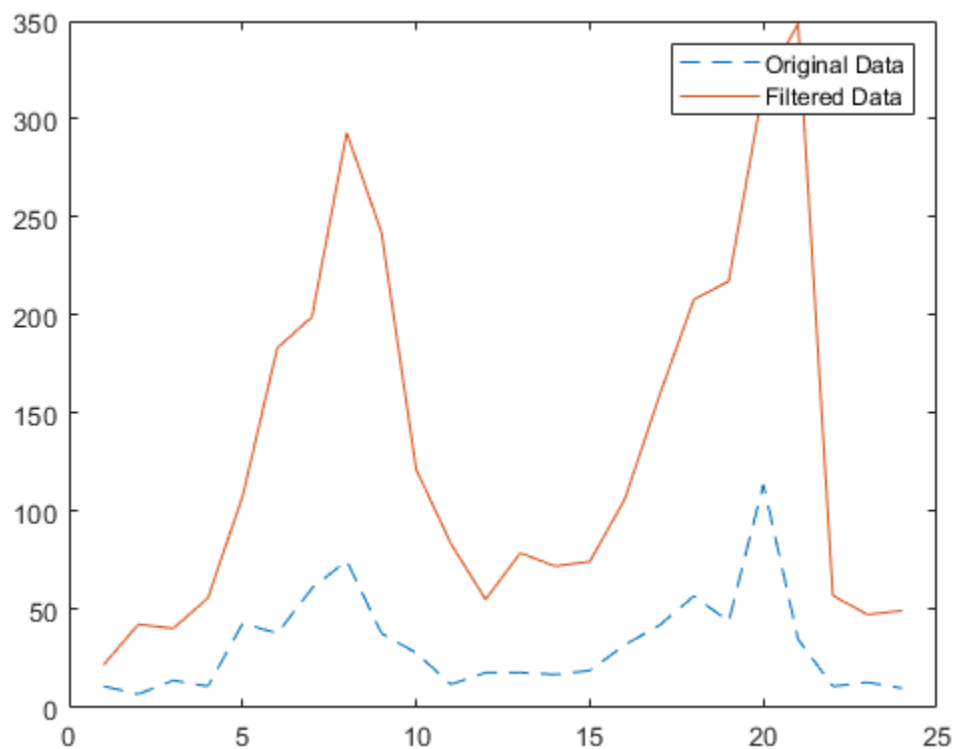
```
load count.dat  
x = count(:,1);
```

根据传递函数 $H(z^{-1})$ 创建滤波器系数向量。

```
a = [1 0.2];  
b = [2 3];
```

计算滤波后的数据，同时绘制原始数据和滤波后的数据。此滤波器主要修改原始数据的振幅。

```
y = filter(b,a,x);  
  
t = 1:length(x);  
plot(t,x,'-',t,y,'-')  
legend('Original Data','Filtered Data')
```



参考

- [1] Oppenheim, Alan V., Ronald W. Schafer, and John R. Buck. *Discrete-Time Signal Processing*. Upper Saddle River, NJ: Prentice-Hall, 1999.

另请参阅

`conv` | `filter` | `filter2` | `movmean` | `smoothdata`

相关示例

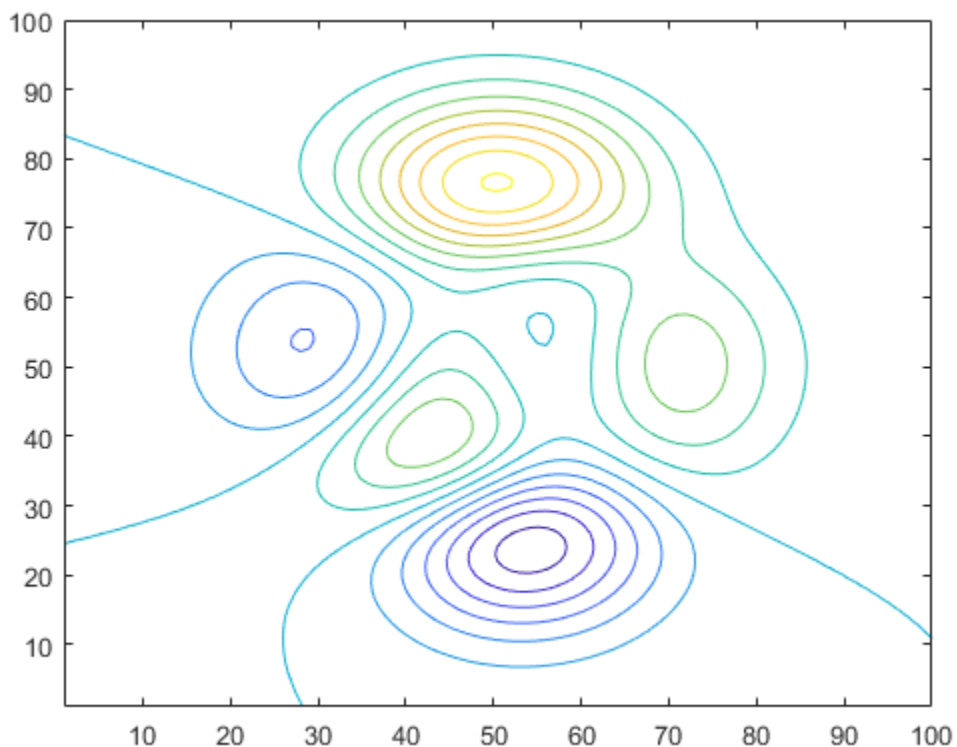
- “使用卷积对数据进行平滑处理”（第 1-31 页）

使用卷积对数据进行平滑处理

您可以使用卷积对包含高频分量的二维数据进行平滑处理。

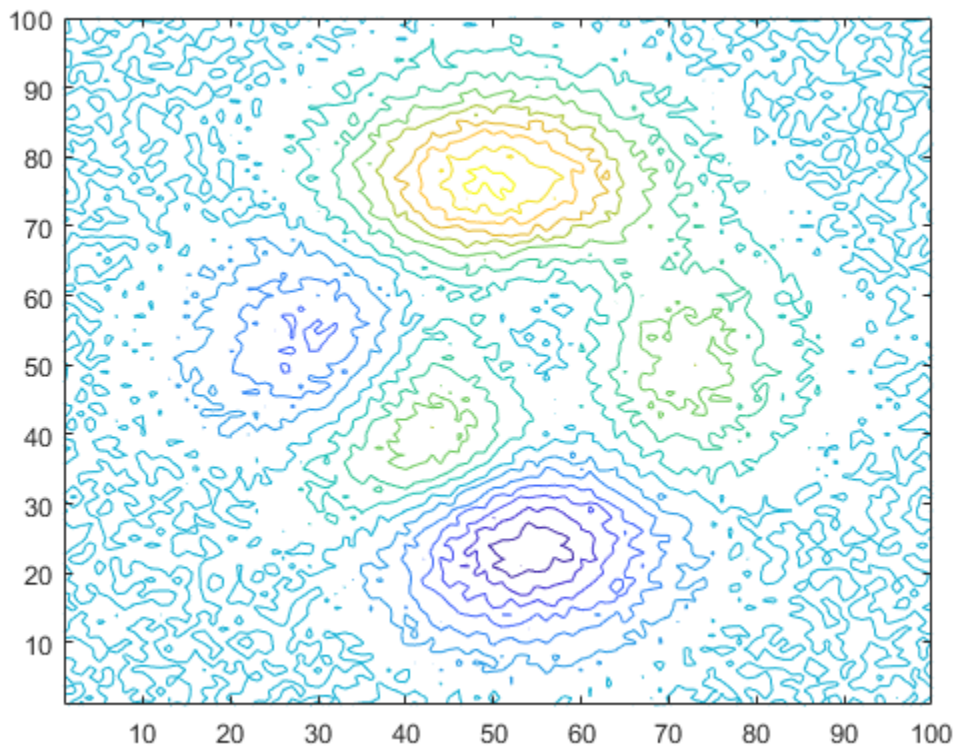
使用 `peaks` 函数创建二维数据，并在各个等高线层级对数据绘图。

```
Z = peaks(100);  
levels = -7:1:10;  
contour(Z,levels)
```



向数据中插入随机噪声并绘制带噪声的等高线。

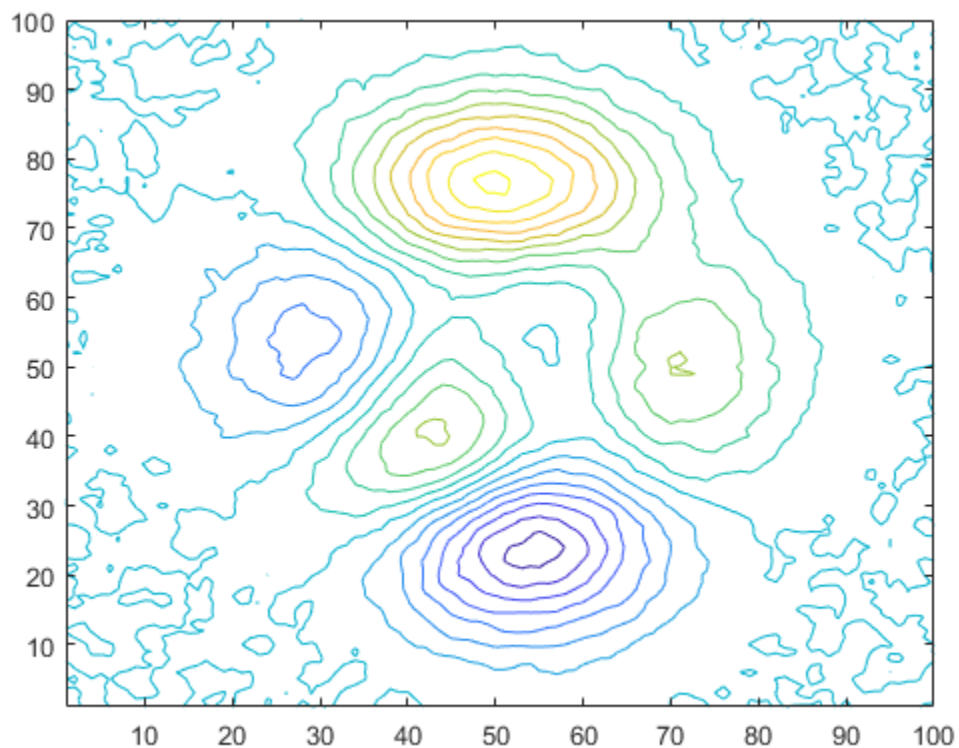
```
Znoise = Z + rand(100) - 0.5;  
contour(Znoise,levels)
```



MATLAB® 中的 `conv2` 函数使用指定的核求二维数据的卷积，该核的元素定义如何去除或增强原始数据的特征。核的大小不必与输入数据相同。小核足以对仅包含少数频率分量的数据进行平滑处理。较大的核可以更精确地对频率响应进行调整，从而得到更平滑的输出。

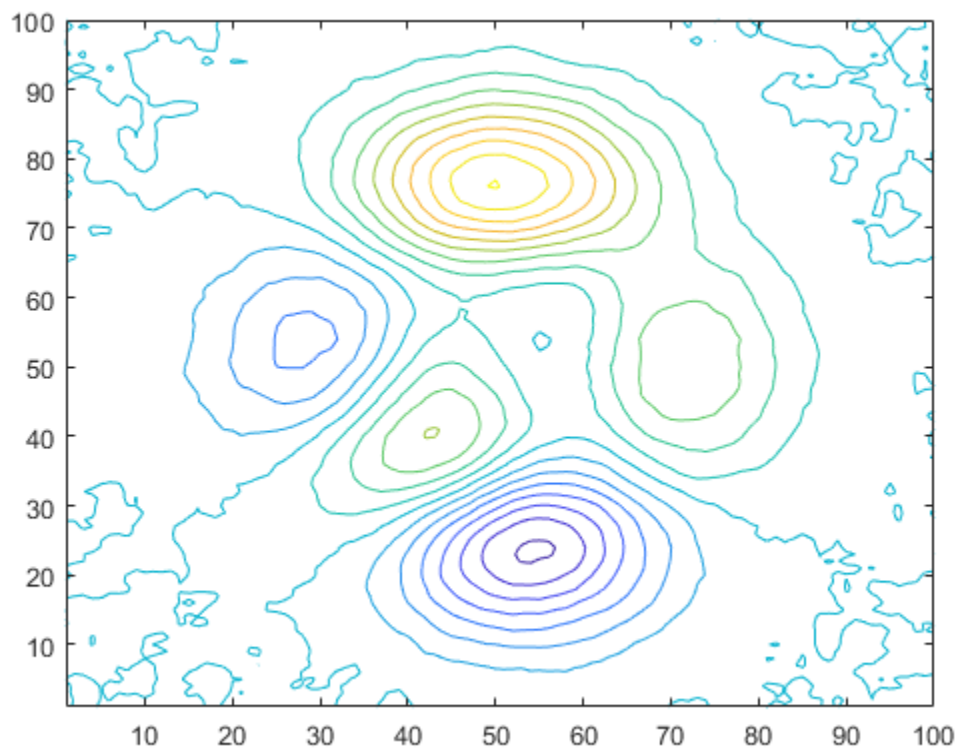
定义一个 3×3 核 `K` 并使用 `conv2` 对 `Znoise` 中的含噪数据进行平滑处理。绘制经过平滑处理的等高线。`conv2` 中的 `'same'` 选项使输出的大小与输入相同。

```
K = 0.125*ones(3);  
Zsmooth1 = conv2(Znoise,K,'same');  
contour(Zsmooth1, levels)
```



用 5×5 核对带噪声的数据进行平滑处理，并绘制新等高线。

```
K = 0.045*ones(5);  
Zsmooth2 = conv2(Znoise,K,'same');  
contour(Zsmooth2,levels)
```



另请参阅

[conv](#) | [conv2](#) | [filter](#) | [smoothdata](#)

相关示例

- “滤波数据”（第 1-26 页）

去除数据的线性趋势

本节内容
“简介” （第 1-35 页）
“从数据中去除线性趋势” （第 1-35 页）

简介

MATLAB 函数 `detrend` 从数据中减去均值或最佳拟合线（以最小二乘方式）。如果您的数据包含多个数据列，`detrend` 会分别处理每个数据列。

通过从数据中去除线性趋势，您能够将分析集中在趋势数据的波动上。线性趋势通常表示数据的系统性增加或减少。例如，传感器漂移可能导致系统性偏移。虽然趋势可能是有意义的，但在去除线性趋势后，某些类型的分析能展现更好的洞察力。

视分析目的不同，您可决定是否需要去除数据中的趋势效应。

从数据中去除线性趋势

此示例说明如何从股票每日收盘价中去除线性趋势，以重点观察整体涨幅的价格波动。如果数据确实有趋势，则去除线性趋势会强制其均值为零并减少总体变化。该示例使用从 `gallery` 函数获取的分布来模拟股价波动。

创建一个模拟数据集并计算其均值。`sdata` 表示股票的每日价格变动。

```
t = 0:300;
dailyFluct = gallery('normaldata',size(t),2);
sdata = cumsum(dailyFluct) + 20 + t/100;
```

求出数据的平均值。

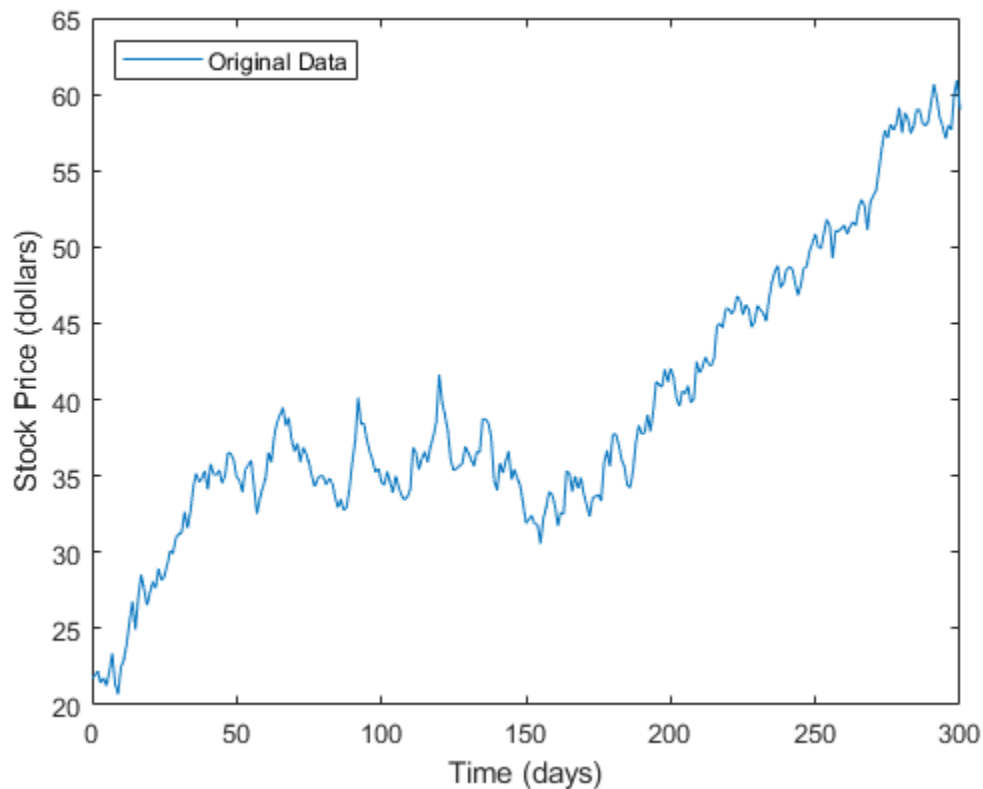
```
mean(sdata)
```

```
ans = 39.4851
```

绘制和标记数据。可以看到，数据显示股价呈系统性增长。

```
figure
plot(t,sdata);
legend('Original Data','Location','northwest');
```

```
xlabel('Time (days)');  
ylabel('Stock Price (dollars)');
```



应用 `detrend`，它对 `sdata` 执行线性拟合，然后对其进行去除线性趋势处理。从输入中减去输出，得出计算所得的趋势线。

```
detrend_sdata = detrend(sdata);  
trend = sdata - detrend_sdata;
```

求出去除线性趋势后的数据的平均值。

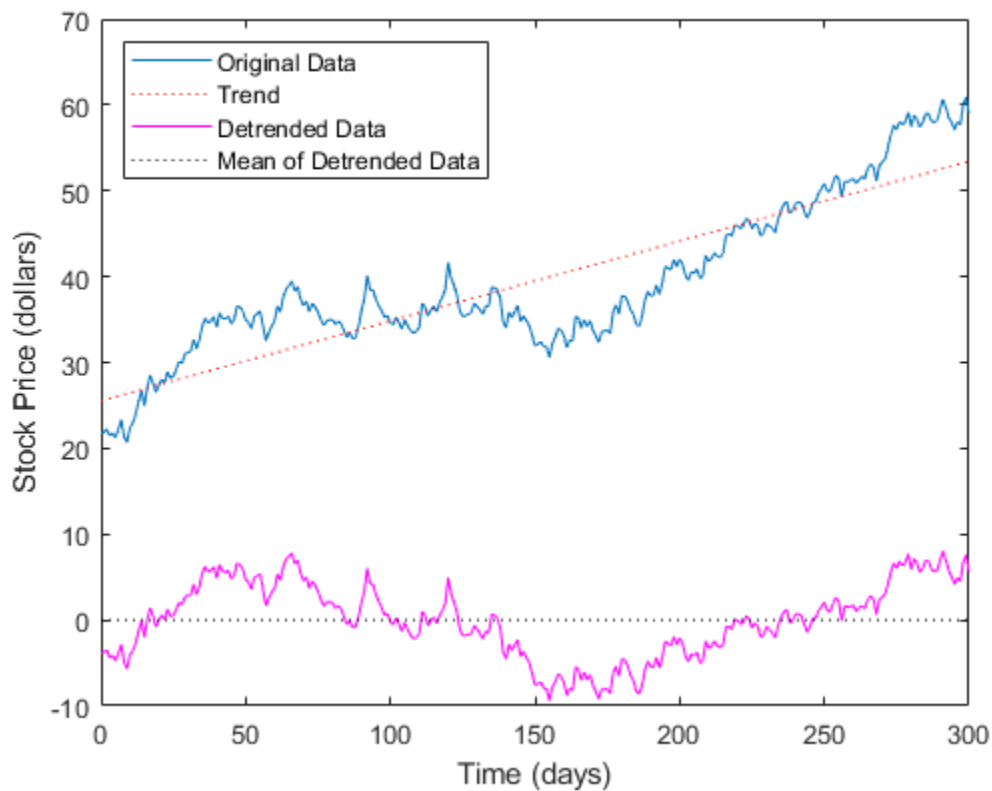
```
mean(detrend_sdata)
```

```
ans = 8.9703e-15
```

去除线性趋势后，数据均值非常接近 0，与预期相符。

将趋势线、去除线性趋势后的数据及其均值添加到图中，以显示结果。

```
hold on
plot(t,trend,'r')
plot(t,detrend_sdata,'m')
plot(t,zeros(size(t)),'k')
legend('Original Data','Trend','Detrended Data',...
       'Mean of Detrended Data','Location','northwest')
xlabel('Time (days)');
ylabel('Stock Price (dollars)');
```



另请参阅

`cumsum` | `detrend` | `gallery` | `plot`

描述性统计量

本节内容
“用于计算描述性统计量的函数”（第 1-39 页）
“示例：使用 MATLAB 数据统计信息”（第 1-41 页）

如果您需要更高级的统计功能，您可能要使用 Statistics and Machine Learning Toolbox™ 软件。

用于计算描述性统计量的函数

使用以下 MATLAB 函数对您的数据进行描述性统计量计算。

注意 对于矩阵数据，每列的描述性统计量是独立计算的。

统计函数汇总

函数	说明
<code>max</code>	最大值
<code>mean</code>	平均值或均值
<code>median</code>	中值
<code>min</code>	最小值
<code>mode</code>	出现次数最多的值
<code>std</code>	标准差
<code>var</code>	方差，用于度量值的分散程度

以下示例应用 MATLAB 函数来计算描述性统计量：

- “示例 1 - 计算最大值、均值和标准差”（第 1-39 页）
- “示例 2 - 减去均值”（第 1-41 页）

示例 1 - 计算最大值、均值和标准差

此示例说明如何使用 MATLAB 函数计算一个 24×3 矩阵（称为 `count`）的最大值、均值和标准差值。MATLAB 为矩阵中的每列独立计算这些统计信息。

```
% Load the sample data
load count.dat
% Find the maximum value in each column
mx = max(count)
% Calculate the mean of each column
mu = mean(count)
% Calculate the standard deviation of each column
sigma = std(count)
```

结果是

```
mx =
    114    145    257

mu =
    32.0000    46.5417    65.5833

sigma =
    25.3703    41.4057    68.0281
```

要获取每个数据列中最大数据值所在的行号，请指定另一个输出参数 **indx** 以返回行索引。例如：

```
[mx,indx] = max(count)
```

这些结果是

```
mx =
    114    145    257

indx =
    20    20    20
```

此处，变量 **mx** 是行向量，它包含三个数据列中每个列中的最大值。变量 **indx** 包含每列中对应于最大值的行索引。

要找到整个 **count** 矩阵中的最小值，请使用语法 **count(:)** 将 24×3 矩阵转换为 72×1 列向量。然后，要找到该单一列中的最小值，请使用以下语法：

```
min(count(:))

ans =
    7
```

示例 2 - 减去均值

使用以下语法从矩阵的每列中减去均值：

```
% Get the size of the count matrix
[n,p] = size(count)
% Compute the mean of each column
mu = mean(count)
% Create a matrix of mean values by
% replicating the mu vector for n rows
MeanMat = repmat(mu,n,1)
% Subtract the column mean from each element
% in that column
x = count - MeanMat
```

注意 从数据中减去均值也称为去除线性趋势。有关从数据中删除均值或最佳拟合线的详细信息，请参阅“去除数据的线性趋势”（第 1-35 页）。

示例：使用 MATLAB 数据统计信息

“数据统计信息”对话框可帮助您计算和绘制数据的描述性统计量。此示例说明如何使用 MATLAB 数据统计信息来计算并绘制 24×3 矩阵 **count** 的统计量。该数据表示有多少辆车经过了三条街道上的交通计数站。

本节包含以下主题：

- “计算和绘制描述性统计量”（第 1-41 页）
- “设置绘图上数据统计量的格式”（第 1-44 页）
- “将统计量保存到 MATLAB 工作区”（第 1-46 页）
- “生成代码文件”（第 1-46 页）

注意 MATLAB 数据统计仅可用于二维图。

计算和绘制描述性统计量

1 加载并绘制数据：

```
load count.dat
[n,p] = size(count);
```

```
% Define the x-values
```

```
t = 1:n;
```

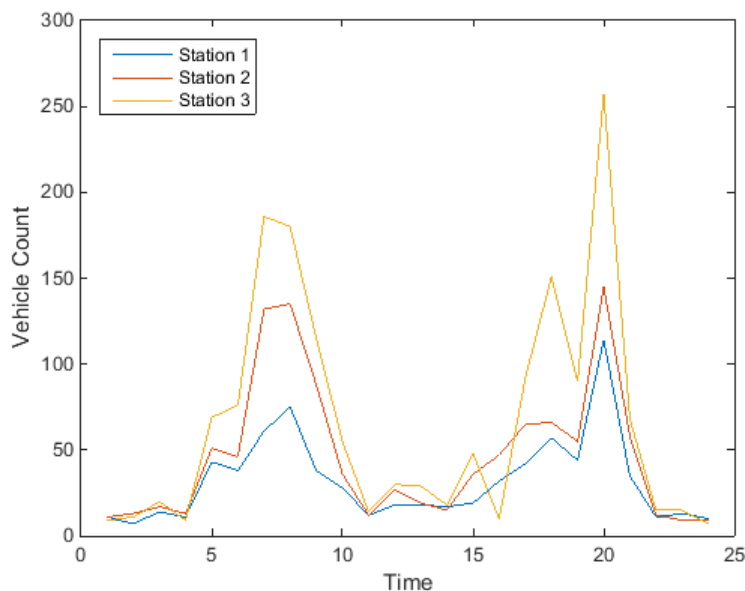
```
% Plot the data and annotate the graph
```

```
plot(t,count)
```

```
legend('Station 1','Station 2','Station 3','Location','northwest')
```

```
xlabel('Time')
```

```
ylabel('Vehicle Count')
```



注意 图例包含每个数据集的名称，由 `legend` 函数指定：Station 1、Station 2 和 Station 3。数据集指您绘制的数组中的每列数据。如果您未命名数据集，则会分配默认名称 `data1`、`data2`，依此类推。

- 2 在图窗窗口中，选择 **工具 > 数据统计信息**。

“数据统计信息”对话框将打开并显示 Station 1 数据集的 X 和 Y 数据的描述性统计量。

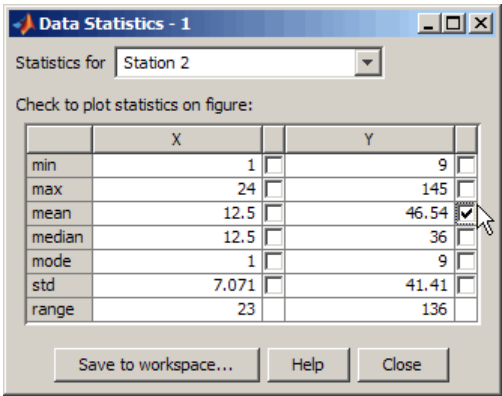
注意 “数据统计信息”对话框显示一个极差，它是所选数据集中最小值和最大值之间的差值。该对话框不在绘图上显示该范围。

- 3 在**以下项的统计信息**列表中选择另一个数据集 **Station 2**。

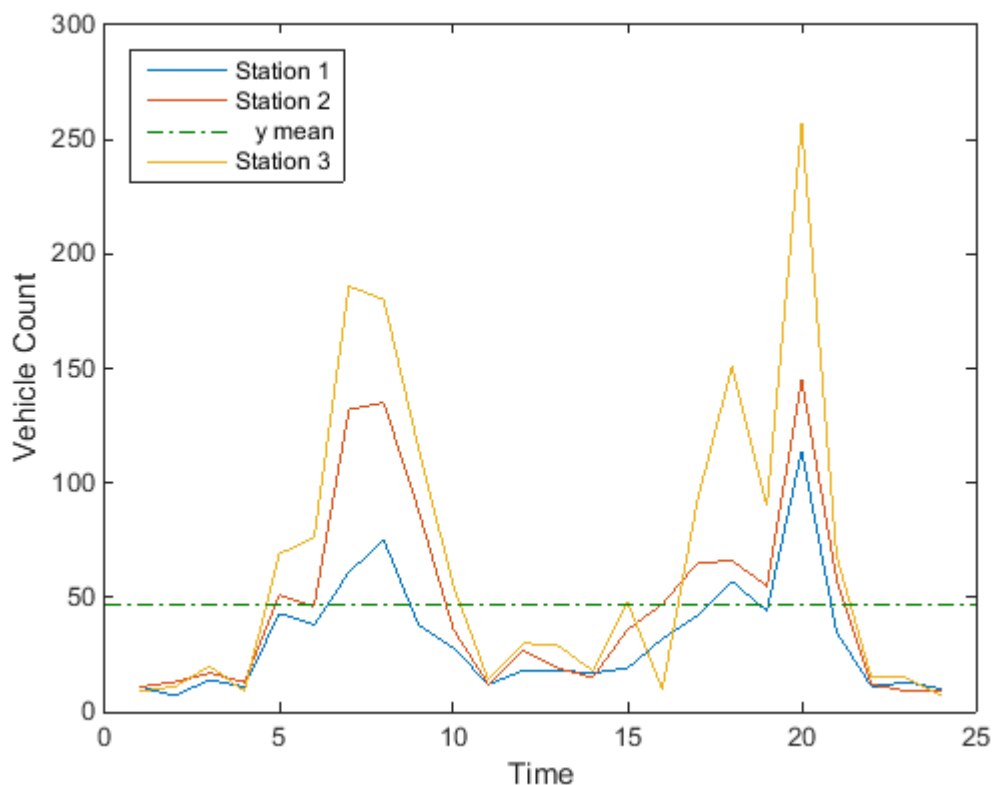
这将显示 **Station 2** 数据集的 X 和 Y 数据的统计量。

- 4 选中您要在绘图上显示的每个统计量的复选框，然后点击**保存到工作区**。

例如，要绘制 **Station 2** 的均值，请在 **Y** 列中选中**均值**复选框。



这将绘制一条水平线来表示 **Station 2** 的均值，并更新图例以包含此统计量。




设置绘图上数据统计量的格式

“数据统计信息”对话框使用颜色和线型将统计量与绘图上的数据区分开来。示例的此部分显示如何自定义绘图上描述性统计量的显示，例如颜色、线宽、线型或标记。

注意 在绘制完数据的所有统计量前，不要编辑统计量的显示属性。如果在编辑绘图属性后添加或删除统计量，则对绘图属性的更改将丢失。

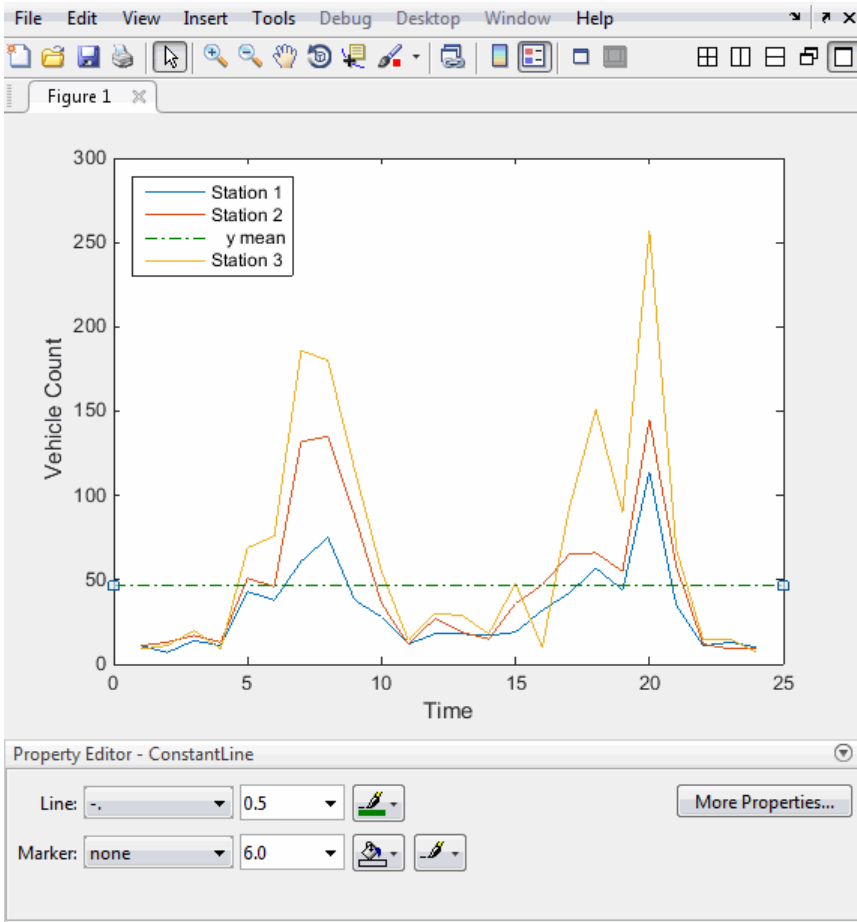
要修改绘图上数据统计量的显示，请执行下列操作：

- 1 在 MATLAB 图窗窗口中，点击工具栏中的  (编辑绘图) 按钮。

此步骤将启用绘图编辑。

- 2 双击要编辑其显示属性的绘图上的统计量。例如，双击表示 **Station 2** 均值的水平线。

此步骤将在 MATLAB 图窗窗口下方打开属性编辑器，您可以在其中修改用于表示此统计量的线条的外观。



- 3 在属性编辑器中，指定**线条**和**标记**的样式、大小和颜色。

提示 或者，右键点击绘图上的统计量，然后从快捷菜单中选择一个选项。

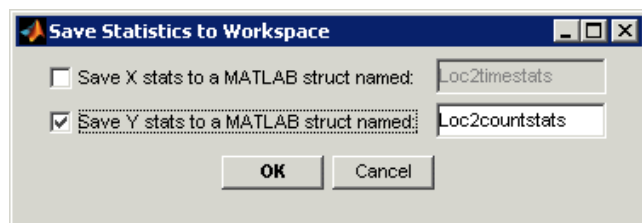
将统计量保存到 MATLAB 工作区

执行下列步骤可将统计量保存到 MATLAB 工作区。

注意 当您的绘图包含多个数据集时，请分别保存各数据集的统计量。要显示其他数据集的统计量，请从“数据统计信息”对话框的**以下项的统计信息**列表中选择该数据集。

- 1 在“数据统计信息”对话框中，点击**保存到工作区**按钮。
- 2 在“将统计信息保存到工作区”对话框中，选择用于保存 X 数据或 Y 数据或两者的统计量的选项。然后，输入相应的变量名称。

在本示例中，只保存 Y 数据。输入变量名称为 **Loc2countstats**。



- 3 点击**确定**。

此步骤将描述性统计量保存到一个结构体中。新变量将添加到 MATLAB 工作区中。

要查看新结构体变量，请在 MATLAB 提示符下键入变量名称：

Loc2countstats

Loc2countstats =

```
min: 9
max: 145
mean: 46.5417
median: 36
mode: 9
std: 41.4057
range: 136
```

生成代码文件

示例的此部分显示如何从图形生成 MATLAB 代码文件，再将代码应用至新数据以重新生成相同格式的绘图和统计量。在 MATLAB Online™ 中不能生成代码文件。

- 1 在图窗窗口中，选择**文件 > 生成代码**。

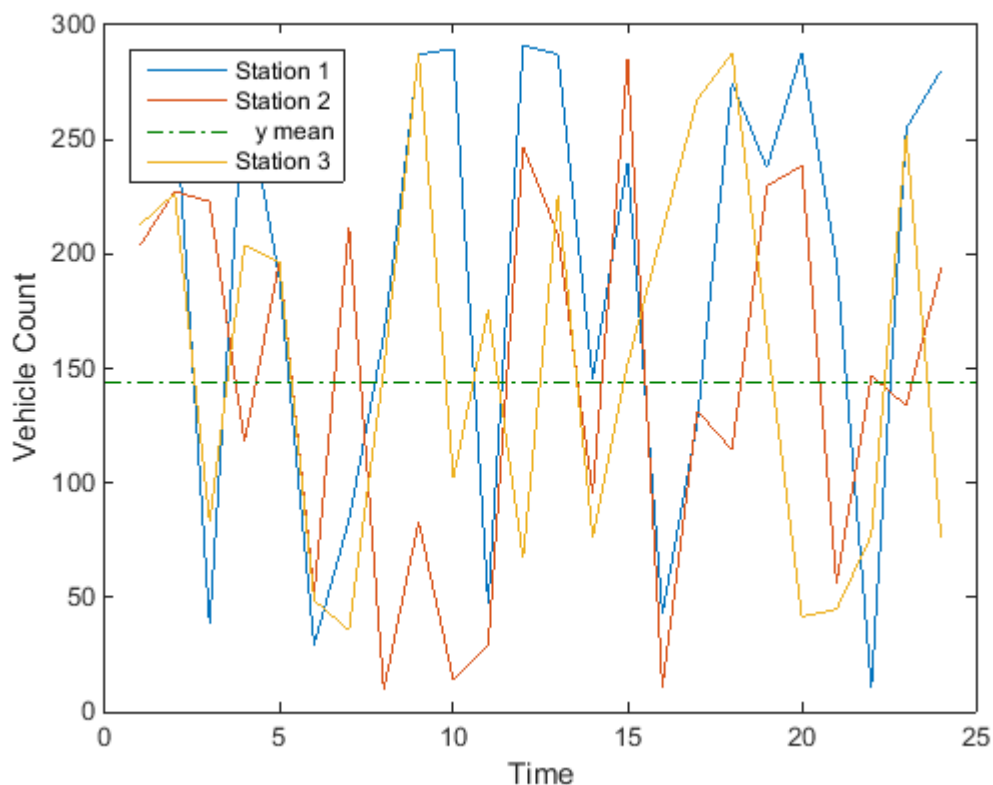
此步骤创建一个函数代码文件并将其显示在 MATLAB 编辑器中。

- 2 将文件的第一行上的函数名称从 `createfigure` 更改为更具体的名称，如 `countplot`。使用文件名 `countplot.m` 将文件保存到您的当前文件夹中。
- 3 生成一些新的随机计数数据：

```
randcount = 300*rand(24,3);
```

- 4 用新数据和重新计算的统计量重新生成绘图：

```
countplot(t,randcount)
```



回归分析

- “线性相关性”（第 2-2 页）
- “线性回归”（第 2-5 页）
- “交互式拟合”（第 2-15 页）
- “以编程方式拟合”（第 2-32 页）

线性相关性

本节内容
“简介” （第 2-2 页）
“协方差” （第 2-2 页）
“相关系数” （第 2-3 页）

简介

相关性用于量化两个变量之间的线性关系强度。若两个变量之间并无相关性，则不存在变量值发生关联式增减的趋势。但是，两个不相关的变量不一定彼此独立，因为它们可能具有非线性关系。

您可以使用线性相关研究变量之间是否在线性关系，而无须对您的数据假设或拟合具体的模型。线性相关较弱或并无线性相关的两个变量可能具有较强的非线性关系。但是，在拟合模型之前计算线性相关是识别具有简单关系的变量的有用方法。研究变量之间相关性的另一种方法是对数据绘制散点图。

协方差用相对于两个变量的方差的单位来量化它们之间的线性关系强度。相关性是归一化的协方差，可提供衡量线性关系程度的无量纲量，且不受任一变量的范围的影响。

以下 MATLAB 函数可计算样本相关系数和协方差。这些样本系数是中提取数据样本的总体数据的真实协方差及相关系数的估计值。

函数	说明
<code>corrcoef</code>	相关系数矩阵
<code>cov</code>	协方差矩阵
<code>xcorr</code>	随机过程的互相关性序列（包括自相关性）

协方差

使用 MATLAB `cov` 函数计算数据矩阵的样本协方差矩阵（每一列代表单独的数量）。

样本协方差矩阵具有以下特征：

- `cov(X)` 是对称的。

- **diag(cov(X))** 是每个数据列的方差的向量。方差代表数据在相应列中的分散程度指标。（**var** 函数计算方差。）
- **sqrt(diag(cov(X)))** 是标准差的向量。（**std** 函数计算标准差。）
- 协方差矩阵的非对角元素代表各个数据列之间的协方差。

在这里，**X** 可以是向量或矩阵。对于 $m \times n$ 矩阵，协方差矩阵为 $n \times n$ 。

若要查看协方差计算示例，请在包含 24×3 矩阵的 **count.dat** 中加载样本数据：

```
load count.dat
```

为此数据计算协方差矩阵：

```
cov(count)
```

MATLAB 得出以下结果：

```
ans =
    1.0e+003 *
    0.6437  0.9802  1.6567
    0.9802  1.7144  2.6908
    1.6567  2.6908  4.6278
```

此数据的协方差矩阵具有以下形式：

$$\begin{bmatrix} s_{11}^2 & s_{12}^2 & s_{13}^2 \\ s_{21}^2 & s_{22}^2 & s_{23}^2 \\ s_{31}^2 & s_{32}^2 & s_{33}^2 \end{bmatrix}$$

$$s_{ij}^2 = s_{ji}^2$$

在这里， s_{ij}^2 是数据的 i 列和 j 列之间的样本协方差。由于矩阵 **count** 含有三列，协方差矩阵为 3×3 。

注意 在向量是 **cov** 的参数的特殊情况下，函数返回方差。

相关系数

MATLAB 函数 **corrcoef** 为数据矩阵产生样本相关系数矩阵（每一列代表单独的数量）。相关系数范围为 -1 至 1，其中：

- 接近 1 的值表示数据列之间有正线性关系。
- 接近 -1 的值表示一个数据列与另一个数据列之间有负线性关系（反相关）。
- 接近或等于 0 的值表示数据列之间无线性关系。

对于 $m \times n$ 矩阵，相关系数矩阵为 $n \times n$ 。相关系数矩阵中的元素排列对应于协方差矩阵中元素的位置，如“协方差”（第 2-2 页）中所述。

若要查看相关系数计算示例，请在包含 24×3 矩阵的 `count.dat` 中加载样本数据：

```
load count.dat
```

输入以下语法以计算相关系数：

```
corrcoef(count)
```

这样会产生以下 3×3 相关系数矩阵：

```
ans =  
    1.0000    0.9331    0.9599  
    0.9331    1.0000    0.9553  
    0.9599    0.9553    1.0000
```

由于所有相关系数都接近 1，`count` 矩阵中每一对数据列之间都具有较强的正相关性。

线性回归

本节内容

“简介”（第 2-5 页）

“简单线性回归”（第 2-6 页）

“残差与拟合优度”（第 2-10 页）

“用 Curve Fitting Toolbox 函数拟合数据”（第 2-13 页）

简介

数据模型明确描述预测变量与响应变量之间的关系。线性回归拟合模型系数为线性的数据模型。最常见的线性回归类型是最小二乘拟合，它可用于拟合线和多项式以及其他线性模型。

在对各对数量之间的关系进行建模之前，最好进行相关性分析，以确定这些数量之间是否存在线性关系。请注意，变量可能具有非线性关系，相关性分析无法检测到这一点。有关详细信息，请参阅“线性相关性”（第 2-2 页）。

MATLAB 基本拟合用户界面可帮助您拟合数据，以便根据数据计算模型系数和绘制模型。有关示例，请参阅“示例：使用基本拟合用户界面”（第 2-17 页）。您还可以使用 MATLAB `polyfit` 和 `polyval` 函数将您的数据拟合至具有线性系数的模型。有关示例，请参阅“以编程方式拟合”（第 2-40 页）。

如果您需要使用非线性模型拟合数据，请转换变量以使关系变成线性关系。或者，尝试使用 Statistics and Machine Learning Toolbox `nlinfit` 函数、Optimization Toolbox™ `lsqcurvefit` 函数或应用 Curve Fitting Toolbox™ 中的函数，直接拟合非线性函数。

本主题解释如何：

- 使用 \ 运算符执行简单的线性回归。
- 使用相关性分析确定两个数量之间是否相关，从而确定其是否适合进行数据拟合。
- 对数据进行线性模型拟合。
- 通过绘制残差及探索模式，评估拟合优度。
- 计算拟合 R^2 和调整后的 R^2 的优度量。

简单线性回归

此示例说明如何使用 **accidents** 数据集执行简单线性回归。此示例还向您说明如何计算决定系数 R^2 以评估回归。**accidents** 数据集包含美国重大交通事故的数据。

线性回归对一个因变量（即响应变量） y 与一个或多个自变量（即预测变量） x_1, \dots, x_n 之间的关系进行建模。简单线性回归使用以下关系方程，仅考虑一个自变量：

$$y = \beta_0 + \beta_1 x + \epsilon,$$

其中， β_0 是 y 轴截距， β_1 是斜率（即回归系数）， ϵ 是误差项。

首先确定一组（ n 个） x 和 y 的观测值，以 $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ 形式给出。对这些值应用简单线性回归关系方程，构成一个线性方程组。这些方程以矩阵形式表示如下：

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}.$$

假设

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}, B = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}.$$

现在关系变为 $Y = XB$ 。

在 MATLAB 中，您可使用 **mldivide** 运算符求 B ，即 $B = X \backslash Y$ 。

从数据集 **accidents** 中将事故数据加载到 **y** 中，将州人口数据加载到 **x** 中。使用 \backslash 运算符求州事故数量与州人口之间的线性回归关系 $y = \beta_1 x$ 。 \backslash 运算符执行最小二乘回归。

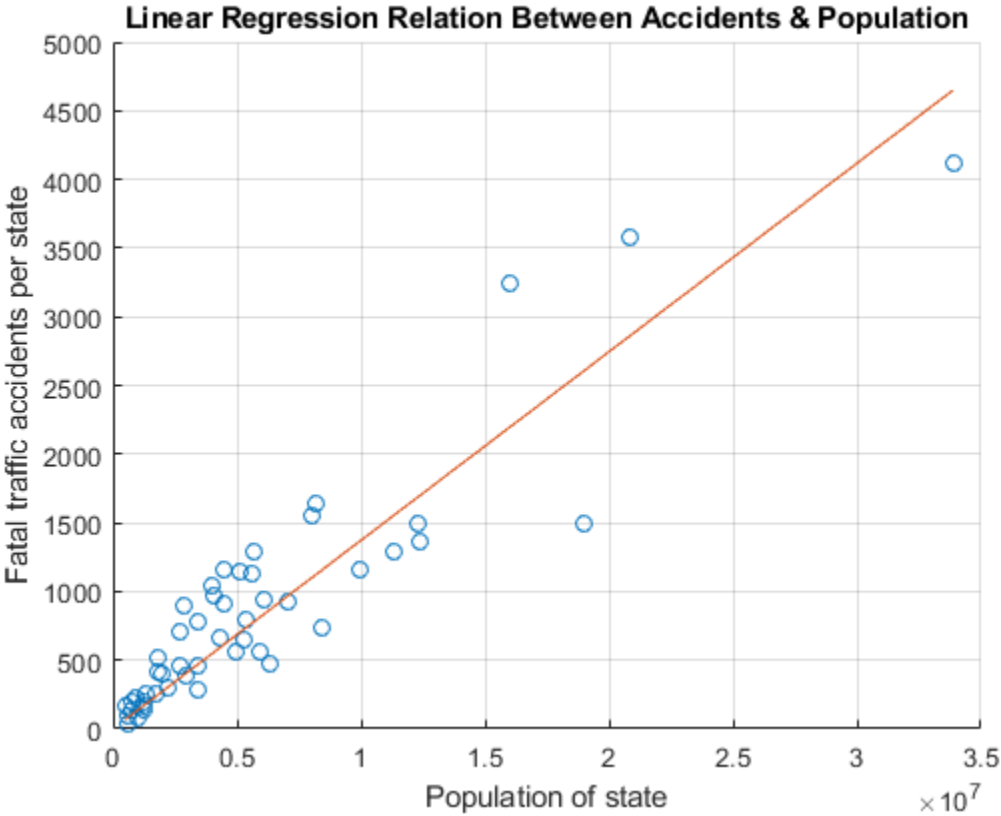
```
load accidents
x = hwydata(:,14); %Population of states
y = hwydata(:,4); %Accidents per state
format long
b1 = x \ y
```

```
b1 =  
    1.372716735564871e-04
```

b1 是斜率或回归系数。线性关系为 $y = \beta_1 x = 0.0001372x$ 。

使用该关系，根据 **x** 计算每州事故数量 **yCalc**。对实际值 **y** 与计算值 **yCalc** 进行绘图，显示回归情况。

```
yCalc1 = b1*x;  
scatter(x,y)  
hold on  
plot(x,yCalc1)  
xlabel('Population of state')  
ylabel('Fatal traffic accidents per state')  
title('Linear Regression Relation Between Accidents & Population')  
grid on
```



在您的模型中纳入 y 轴截距 β_0 以改进拟合，即 $y = \beta_0 + \beta_1 x$ 。用一系列 1 填补 x 并使用 \ 运算符计算 β_0 。

```
X = [ones(length(x),1) x];  
b = X\y
```

```
b = 2×1  
102 ×
```

```
1.427120171726537  
0.000001256394274
```

此结果表示关系 $y = \beta_0 + \beta_1 x = 142.7120 + 0.0001256x$ 。

在同一幅图上绘制该结果，以图窗方式显示该关系。

```
yCalc2 = X*b;  
plot(x,yCalc2,'-')  
legend('Data','Slope','Slope & Intercept','Location','best');
```



如图所示，两个拟合非常相似。探索更佳拟合的一种方法是计算决定系数 R^2 。 R^2 用于度量模型能够在多大程度上预测数据，其值介于 0 和 1 之间。 R^2 的值越高，模型预测数据的准确性越高。

其中， \hat{y} 表示 y 的计算值， \bar{y} 是 y 的均值， R^2 定义为

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}.$$

通过比较 R^2 的值，找出两个拟合中较好的一个。如 R^2 值所示，包含 y 轴截距的第二个拟合更好。

```
Rsq1 = 1 - sum((y - yCalc1).^2)/sum((y - mean(y)).^2)
```

```
Rsq1 =  
0.822235650485566
```

```
Rsq2 = 1 - sum((y - yCalc2).^2)/sum((y - mean(y)).^2)
```

```
Rsq2 =  
0.838210531103428
```

残差与拟合优度

残差是响应变量（因变量）的观测值与模型的预测值之间的差。当拟合的模型适合数据时，残差接近独立随机误差。即，残差分布不应该呈现出可辨识的模式。

利用线性模型产生拟合需要尽量减小残差平方和。该最小化的结果即为最小二乘拟合。您可通过直观地观察残差图，了解拟合的“优度”。如果残差图具有一定的模式（即残差数据点未呈现随机分布），该随机性表明该模型并未适当地拟合数据。

评估您对数据进行的每个拟合。例如，如果您拟合数据的目的是提取具有物理含义的系数，则必须确保您的模型能够反映数据的物理属性。了解您的数据代表着什么、如何度量以及如何建模在评估拟合优度时非常重要。

拟合优度的一个度量是决定系数或 R^2 （读作 R 的平方）。该统计量表明您通过拟合模型得到的值与模型可预测的因变量的匹配程度。统计人员通常利用拟合模型的残差方差定义 R^2 ：

$$R^2 = 1 - \frac{SS_{\text{resid}}}{SS_{\text{total}}}$$

SS_{resid} 是与回归的残差的平方和。 SS_{total} 是与因变量均值的差的平方和（总平方和）。两者都是正标量。

若要了解在使用基本拟合工具时如何计算 R^2 ，请参阅“计算决定系数 R^2 ”（第 2-21 页）。若要了解关于计算 R^2 统计量及其多元概化的更多信息，请继续阅读此处内容。

示例：通过多项式拟合计算 R^2

您可以从多项式回归的系数得出 R^2 ，以确定线性模型对 y 的方差的解释率，如以下示例所述：

- 1 从数据文件 `count.dat` 中 `count` 变量的前两列创建两个变量 x 和 y ：

```
load count.dat
x = count(:,1);
y = count(:,2);
```

- 2 利用 `polyfit` 计算从 x 预测 y 的线性回归：

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

```
p =
    1.5229   -2.1911
```

$p(1)$ 是斜率， $p(2)$ 是线性预测变量的截距。您还可以使用基本拟合用户界面（第 2-15 页）获得回归系数。

- 3 调用 `polyval` 以使用 p 预测 y ，调用结果 `yfit`：

```
yfit = polyval(p,x);
```

使用 `polyval`，您无需自行输入拟合方程，在本例中拟合方程为：

```
yfit = p(1) * x + p(2);
```

- 4 将残差值计算为有符号数的向量：

```
yresid = y - yfit;
```

- 5 计算残差的平方并相加，以获得残差平方和：

```
SSresid = sum(yresid.^2);
```

- 6 通过将观测次数减 1 再乘以 y 的方差，计算 y 的总平方和：

```
SStotal = (length(y)-1) * var(y);
```

- 7 利用本主题简介部分给出的公式计算 R^2 ：

```
rsq = 1 - SSresid/SStotal
```

```
rsq =
    0.8707
```

这表明，线性方程 $1.5229 * x - 2.1911$ 可预测变量 y 中方差的 87%。

计算多项式回归的调整后的 R²

您通常可通过拟合更高次多项式，减少模型中的残差。当您添加更多项时，会增加决定系数 R²。您可获得更接近数据的拟合，但代价是模型更为复杂，R² 无法解释。因而，作为对该统计量的改进，调整后的 R² 中包括了一项对模型中项数的罚值。因此，调整后的 R² 更适合比较不同的模型对同一数据的拟合程度。调整后的 R² 定义如下：

$$R^2_{\text{adjusted}} = 1 - \left(\frac{SS_{\text{resid}}}{SS_{\text{total}}} \right) * ((n-1)/(n-d-1))$$

其中 n 是数据中的观测值数量，d 是多项式的次数。（线性拟合的阶数为 1，二次拟合为 2，三次拟合为 3，依此类推。）

以下示例重复上一示例“示例：通过多项式拟合计算 R²”（第 2-11 页）的步骤，但进行三次（3 阶）拟合而非线性（1 阶）拟合。通过三次拟合，您可同时计算简单的和调整后的 R² 值，以评估额外的项是否可改善预测能力：

- 1 从数据文件 `count.dat` 中 `count` 变量的前两列创建两个变量 `x` 和 `y`：

```
load count.dat
x = count(:,1);
y = count(:,2);
```

- 2 调用 `polyfit` 生成三次拟合，以从 `x` 预测 `y`：

```
p = polyfit(x,y,3)

p =
    -0.0003    0.0390    0.2233    6.2779
```

`p(4)` 是三次预测变量的截距。您还可以使用基本拟合用户界面（第 2-15 页）获得回归系数。

- 3 调用 `polyval` 以使用 `p` 中的系数预测 `y`，将结果命名为 `yfit`：

```
yfit = polyval(p,x);

polyval 计算显式方程，手动输入则如下所示：
```

```
yfit = p(1) * x.^3 + p(2) * x.^2 + p(3) * x + p(4);
```

- 4 将残差值计算为有符号数的向量：

```
yresid = y - yfit;
```

- 5 计算残差的平方并相加，以获得残差平方和：

```
SSresid = sum(yresid.^2);
```

- 6 通过将观测次数减 1 再乘以 y 的方差，计算 y 的总平方和：

```
SStotal = (length(y)-1) * var(y);
```

- 7 利用本主题简介部分给出的公式计算三次拟合的简单 R^2 ：

```
rsq = 1 - SSresid/SStotal
```

```
rsq =  
    0.9083
```

- 8 最后，计算调整后的 R^2 以解释自由度：

```
rsq_adj = 1 - SSresid/SStotal * (length(y)-1)/(length(y)-length(p))
```

```
rsq_adj =  
    0.8945
```

调整后的 R^2 (0.8945) 小于简单 R^2 (0.9083)。后者可以更可靠地估计多项式模型的预测能力。

在许多多项式回归模型中，对方程添加项会使 R^2 和调整后的 R^2 都增加。在前面的示例中，与线性拟合相比，使用三次拟合使这两种统计量都有所增加。（您可自行计算线性拟合的调整后的 R^2 ，能够看到它具有较小的值。）但是，线性拟合并非始终差于更高阶拟合：更复杂拟合的调整后的 R^2 也有可能低于更简单的拟合，此时表明增加复杂度并不适当。此外，虽然基本拟合工具生成的多项式回归模型的 R^2 值始终在 0 和 1 之间变动，但某些模型的调整后的 R^2 可能为负值，这表明该模型的项太多。

相关并不意味着因果性。因此，应始终谨慎解释相关性和确定系数。这些系数仅用来量化拟合模型对因变量方差的消除率。此类度量并不说明您的模型（或您选择的自变量）有多适合用于解释模型预测的变量行为。

用 Curve Fitting Toolbox 函数拟合数据

Curve Fitting Toolbox 软件通过启用以下数据拟合功能扩展 MATLAB 核心功能：

- 线性及非线性参数拟合，包括标准线性最小二乘、非线性最小二乘、加权最小二乘、约束最小二乘以及稳健拟合程序
- 非参数拟合
- 确定拟合优度的统计量
- 外插、微分和积分
- 有助于数据分段及平滑处理的对话框

- 以不同的格式保存拟合结果，包括 MATLAB 代码文件、MAT 文件及工作区变量。

有关详细信息，请参阅 Curve Fitting Toolbox 文档。

交互式拟合

本节内容
“基本拟合用户界面” （第 2-15 页）
“基本拟合准备” （第 2-15 页）
“打开基本拟合用户界面” （第 2-16 页）
“示例：使用基本拟合用户界面” （第 2-17 页）

基本拟合用户界面

MATLAB 基本拟合用户界面能让您以交互方式：

- 利用样条插值、保形插值或最高 10 次的多项式进行数据建模
- 绘制一个或多个数据拟合图
- 绘制拟合的残差图
- 计算模型系数
- 计算残差范数（可用于分析模型与数据的拟合度的一种统计量）
- 使用模型进行内插或在数据外部进行外插
- 将系数及计算出的值保存到 MATLAB 工作区，以便在对话框外部使用
- 生成 MATLAB 代码，以使用新数据重新计算拟合和重新生成绘图

注意 基本拟合用户界面仅可用于二维绘图。若要了解更先进的拟合及回归分析，请参阅 Curve Fitting Toolbox 文档和 Statistics and Machine Learning Toolbox 文档。

基本拟合准备

基本拟合用户界面在拟合前对数据进行升序排列。如果您的数据集较大且值未按升序排列，基本拟合用户界面在拟合前可能需要较长时间来预处理数据。


您可通过提前对数据进行排序，加快基本拟合用户界面的处理速度。若要通过数据向量 **x** 和 **y** 创建排序向量 **x_sorted** 和 **y_sorted**，请使用 MATLAB **sort** 函数：

```
[x_sorted, i] = sort(x);  
y_sorted = y(i);
```

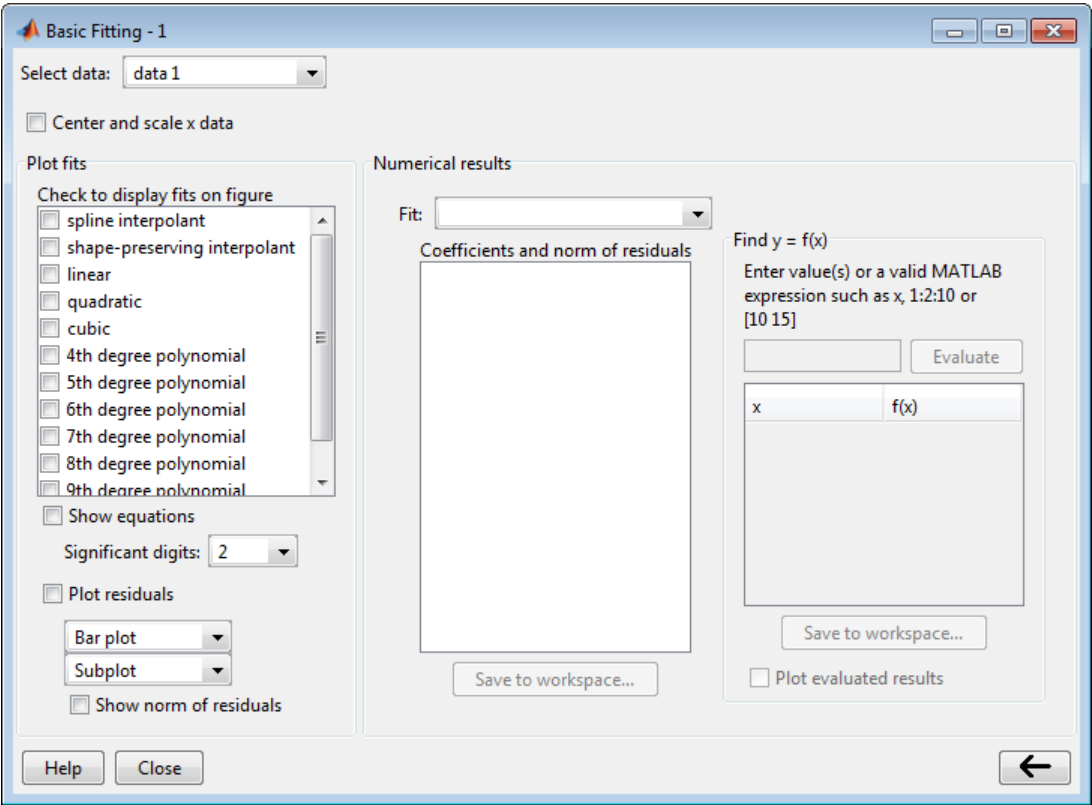
打开基本拟合用户界面

若要使用基本拟合用户界面，您必须首先利用（仅）产生 x 和 y 数据的任何 MATLAB 绘图命令在图窗窗口中绘制数据。

若要打开基本拟合用户界面，请从图窗窗口顶部的菜单中选择**工具 > 基本拟合**。

当您通过双击右下角的箭头按钮  完全展开它时，窗口显示三个面板。使用这些面板：

- 选择模型及绘图选项
- 检查及导出模型系数及残差范数
- 检查及导出内插值及外插值。



若要逐一展开或收起面板，请点击界面右下角的箭头按钮。

示例：使用基本拟合用户界面

此示例说明如何使用基本拟合用户界面拟合、可视化、分析、保存多项式回归以及为其生成代码。

- “加载和绘制人口普查数据”（第 2-17 页）
- “通过三次多项式拟合预测人口普查数据”（第 2-17 页）
- “查看和保存三次拟合参数”（第 2-20 页）
- “计算决定系数 R^2 ”（第 2-21 页）
- “内插和外插人口值”（第 2-25 页）
- “生成代码文件以重新生成结果”（第 2-28 页）
- “了解基本拟合工具如何计算拟合”（第 2-29 页）

加载和绘制人口普查数据

文件 `census.mat` 包含 1790 年至 1990 年的美国人口数据，以 10 年为间隔。

若要加载及绘制数据，请在 MATLAB 提示符下键入以下命令：

```
load census  
plot(cdate,pop,'ro')
```

`load` 命令将以下变量添加至 MATLAB 工作区：

- `cdate` - 包含从 1790 年到 1990 年（以 10 为增量）的年份列向量。它是预测变量。
- `pop` - `cdate` 中每一年的美国人口列向量。它是响应变量。

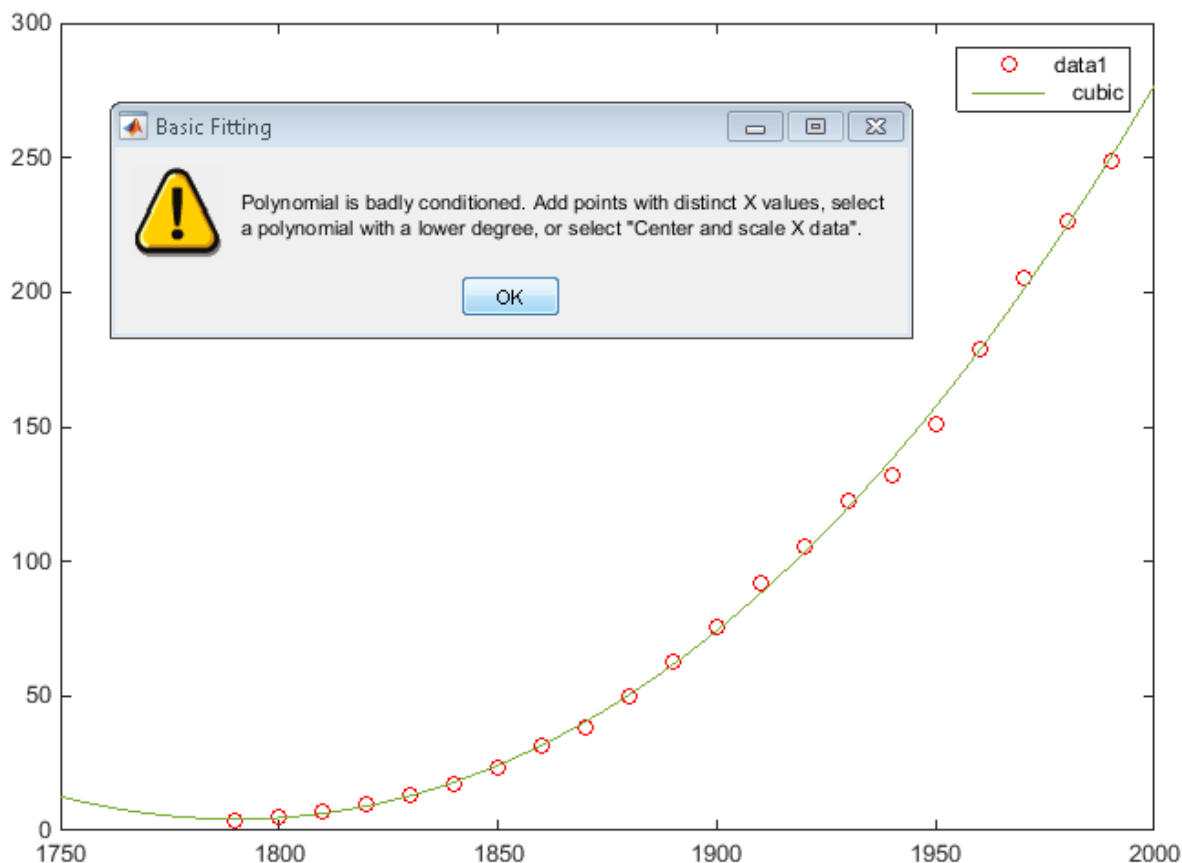
数据向量按年以升序排列。该图将人口显示为年份的函数。

现在，您已准备好对数据进行方程拟合，以建立人口随时间增长的模型。

通过三次多项式拟合预测人口普查数据

- 1 在图窗窗口中选择 **工具 > 基本拟合**，打开“基本拟合”对话框。
- 2 在“基本拟合”对话框的**绘制拟合图**区域中，选中**三次方**复选框以将三次多项式拟合至数据。

MATLAB 根据您的选项拟合数据，并将三次回归线添加至图中，如下所示。



在计算拟合时，MATLAB 遇到问题并发出以下警告：

多项式未正确设置条件。请添加具有不同 **X** 值的点，选择次数较低的多项式或者选择“中心化并缩放 **X** 数据”。

此警告指示计算出的该模型的系数对响应（测得的人口）中的随机误差敏感。它同时会提出一些建议，以帮助改进拟合。

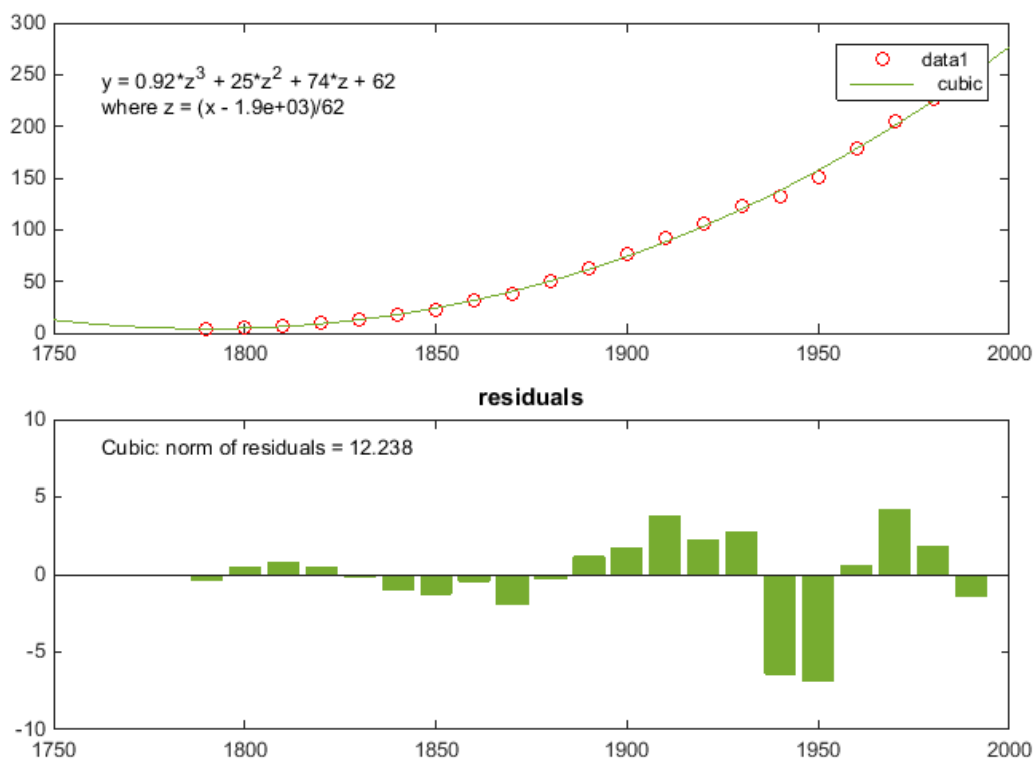
- 3 继续使用三次拟合。由于您无法对人口普查数据添加新观测值，您可在重新计算拟合前将您拥有的值转换为 z 值以改进拟合。选中对话框中的**中心化并缩放 X 数据**复选框，使基本拟合工具执行转换。

若要了解数据中心化和缩放的工作方式，请参阅“了解基本拟合工具如何计算拟合”（第 2-29 页）。

- 4 现在查看方程和显示残差。除了选中**中心化并缩放 X 数据**和**三次方**复选框以外，再选中以下选项：

- **显示方程**
- **绘制残差图**
- **显示残差范数**


选择**绘制残差图**会将其子图创建为条形图。下图显示您所选的基本拟合用户界面选项的结果。

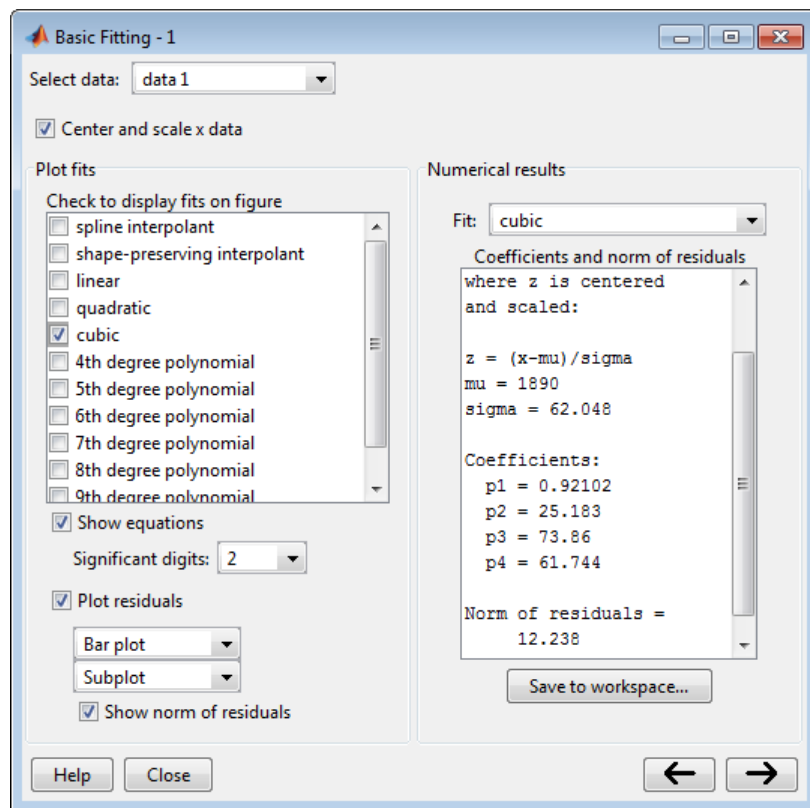


对于 1790 年之前，三次拟合是较差的预测变量，因为它显示在此期间人口呈递减趋势。该模型似乎更适合 1790 年之后的数据。但是，残差的模式显示该模型并不符合正态误差假设，而这正是最小二乘拟合的基础。图例中标识的 **data 1** 行是观测到的 x (**cdate**) 和 y (**pop**) 数据值。**三次方** 回归线显示对数据值进行中心化并缩放后的拟合。请注意，即使工具使用变换后的 Z 值计算拟合，该图仍以原始数据单位显示图窗。

为进行比较，请尝试在**绘制拟合图**区域中选择另一个多项式方程来拟合人口普查数据。

查看和保存三次拟合参数

在“基本拟合”对话框中，点击箭头按钮 ，在**数值结果**面板中显示估算的系数及残差范数。



若要查看特定拟合，请从**拟合**列表中选择它。这将在“基本拟合”对话框中显示系数，但不会在图窗窗口中绘制拟合图。

注意 如果您还希望在绘图中显示拟合，必须选中相应的**绘制拟合图**复选框。

通过点击“数值结果”面板上的**保存到工作区**按钮，将拟合数据保存至 MATLAB 工作区。
“将拟合保存到工作区”对话框随即打开。

选中所有复选框后，点击**确定**以将拟合参数保存为 MATLAB 结构体：

```
fit
fit =
    type: 'polynomial degree 3'
    coeff: [0.9210 25.1834 73.8598 61.7444]
```

现在，您可以在基本拟合用户界面以外将拟合结果用于 MATLAB 编程。

计算决定系数 R²

您可通过计算决定系数或 R 平方 (R^2)，确定多项式回归预测您的观测数据的能力。 R^2 统计量的范围介于 0 至 1 之间，它用来度量自变量在预测因变量的值方面的可用度：

- 接近 0 的 R^2 表示该拟合并不明显优于模型 $y = \text{constant}$ 。
- 接近 1 的 R^2 表示自变量能够解释因变量的大部分变化。


若要计算 R^2 ，请首先计算拟合，然后通过它获得残差。残差是因变量的观测值与其拟合预测值之间的有符号差值。

$$\text{residuals} = y_{\text{observed}} - y_{\text{fitted}}$$

基本拟合工具可为它计算的任何拟合生成残差。若要查看残差图，请选中**绘制残差图**复选框。您可以条形图、线图或散点图形式查看残差。

在您获得残差值后，您可将它们保存到工作区，在那里您可以计算 R^2 。完成本示例的前面部分，对人口普查数据进行三次多项式拟合，然后执行以下步骤：

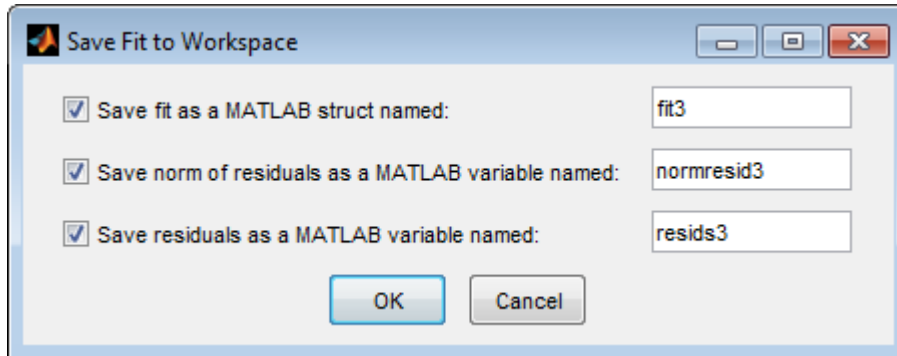
计算三次拟合的残差数据及 R²

- 1 点击右下角的箭头按钮 ，打开“数值结果”选项卡（若尚未显示）。
- 2 从**拟合**下拉菜单中选择“三次方”（若尚未显示）。
- 3 点击**保存到工作区**，保存拟合系数、残差范数及残差。

“将拟合保存到工作区”对话框随即打开，其中有三个复选框和三个文本字段。

- 4 选中所有三个复选框，以保存拟合系数、残差范数及残差值。

- 5 将保存的变量标识为属于三次拟合。通过将 3 添加至各个默认名称（如 `fit3`、`normresid3` 和 `resids3`），更改变量名称。该对话框应如下图所示。



- 6 点击**确定**。基本拟合将残差保存为数字的列向量，将拟合系数保存为结构体，将残差范数保存为标量。

请注意，基本拟合为残差范数计算的值为 **12.2380**。此数字是三次拟合的残差平方和的平方根。

- 7 您也可以验证基本拟合工具提供的残差范数值。从您刚刚保存的 `resids3` 数组自行计算残差范数：

```
mynormresid3 = sum(resids3.^2)^(1/2)
```

```
mynormresid3 =  
12.2380
```

- 8 计算因变量 `pop` 的总平方和以计算 R^2 。总平方和是变量的各值与均值之差的平方和。例如，使用以下代码：

```
SSpop = (length(pop)-1) * var(pop)
```

```
SSpop =  
1.2356e+005
```

`var(pop)` 计算人口向量的方差。将它与观测值个数减去 1 之后的差值相乘，以将自由度考虑在内。总平方和与残差范数均为正标量。

- 9 现在，使用 `normresid3` 的平方与 `SSpop` 一起计算 R^2 ：

```
rsqcubic = 1 - normresid3^2 / SSPop
```

```
rsqcubic =
0.9988
```

- 10 最后，计算线性拟合的 R^2 ，并将它与您刚才得出的三次 R^2 值进行比较。基本拟合用户界面还为您提供线性拟合结果。若要获得线性结果，请重复步骤 2-6，将您的操作修改如下：

- 若要计算最小二乘线性回归系数和统计，请在“数值结果”面板的**拟合**下拉菜单中选择“线性”而非“三次方”。
- 在“保存到工作区”对话框中，将 1 附加至每个变量名称以将其标识为取自线性拟合，然后点击**确定**。变量 `fit1`、`normresid1` 和 `resids1` 现在存在于工作区中。
- 使用变量 `normresid1` (98.778) 计算线性拟合的 R^2 ，操作与您为三次拟合执行的步骤 9 相同：

```
rsqlinear = 1 - normresid1^2 / SSpop
```


```
rsqlinear =
0.9210
```

此结果表明，人口数据的线性最小二乘拟合可解释其方差的 92.1%。由于此数据的三次拟合可解释该方差的 99.9%，该拟合似乎是更佳的预测变量。但是，由于三次拟合使用三个变量 (x 、 x^2 和 x^3) 进行预测，基本的 R^2 值并不能完全反映该拟合的稳健性。评估多变量拟合优度的更合适的度量是调整后的 R^2 。若要了解关于计算和使用调整后的 R^2 的信息，请参阅“残差与拟合优度”（第 2-10 页）。

小心 R^2 度量多项式方程预测因变量的能力，而不是多项式模型对数据的适合程度。当您分析在实质上不可预测的数据时，较小的 R^2 值表示自变量不能准确预测因变量。但是，它并不一定表示该拟合有问题。

为线性拟合计算残差数据及 R^2

在接下来的示例中，使用基本拟合用户界面执行线性拟合，将结果保存到工作区，并且为线性拟合计算 R^2 。然后，您可以将线性 R^2 与您在示例“计算三次拟合的残差数据及 R^2 ”（第 2-21 页）中得出的三次 R^2 值进行比较。

- 1 点击右下角的箭头按钮 ，打开“数值结果”选项卡（若尚未显示）。
- 2 选中**绘制拟合图**区域中的**线性**复选框。
- 3 从**拟合**下拉菜单中选择“线性”（若尚未显示）。“残差的系数和残差范数”区域显示线性拟合的统计。

- 4 点击**保存到工作区**，保存拟合系数、残差范数及残差。

“将拟合保存到工作区”对话框随即打开，其中有三个复选框和三个文本字段。

- 5 选中所有三个复选框，以保存拟合系数、残差范数及残差值。
- 6 将保存的变量标识为属于线性拟合。通过将 1 添加至各个默认名称（如 **fit1**、**normresid1** 和 **resids1**），更改变量名称。
- 7 点击**确定**。基本拟合将残差保存为数字的列向量，将拟合系数保存为结构体，将残差范数保存为标量。

请注意，基本拟合为残差范数计算的值为 **98.778**。此值是线性拟合的残差平方和的平方根。

- 8 您也可以验证基本拟合工具提供的残差范数值。从您刚刚保存的 **resids1** 数组自行计算残差范数：

```
mynormresid1 = sum(resids1.^2)^(1/2)
```

```
mynormresid1 =  
98.7783
```

- 9 计算因变量 **pop** 的总平方和以计算 R^2 。总平方和是变量的各值与均值之差的平方和。例如，使用以下代码：

```
SSpop = (length(pop)-1) * var(pop)
```

```
SSpop =  
1.2356e+005
```

var(pop) 计算人口向量的方差。将它与观测值个数减去 1 之后的差值相乘，以将自由度考虑在内。总平方和与残差范数均为正标量。

- 10 现在，使用 **normresid1** 的平方与 **SSpop** 一起计算 R^2 ：

```
rsqlinear = 1 - normresid1^2 / SSPop
```


```
rsqcubic =  
0.9210
```

此结果表明，人口数据的线性最小二乘拟合可解释其方差的 92.1%。由于此数据的三次拟合可解释该方差的 99.9%，该拟合似乎是更佳的预测变量。但是，三次拟合有四个系数（ x 、 x^2 、 x^3 和一个常量），而线性拟合有两个系数（ x 和一个常量）。简单的 R^2 统计不能解释不同的自由度。评估多项式拟合的更合适度量是调整后的 R^2 。若要了解关于计算和使用调整后的 R^2 的信息，请参阅“残差与拟合优度”（第 2-10 页）。

小心 R^2 度量多项式方程预测因变量的能力，而不是多项式模型对数据的适合程度。当您分析在实质上不可预测的数据时，较小的 R^2 值表示自变量不能准确预测因变量。但是，它并不一定表示该拟合有问题。

内插和外插人口值

假设您希望使用三次模型对 1965 年（原始数据中并未提供的日期）的美国人口数据进行插值。

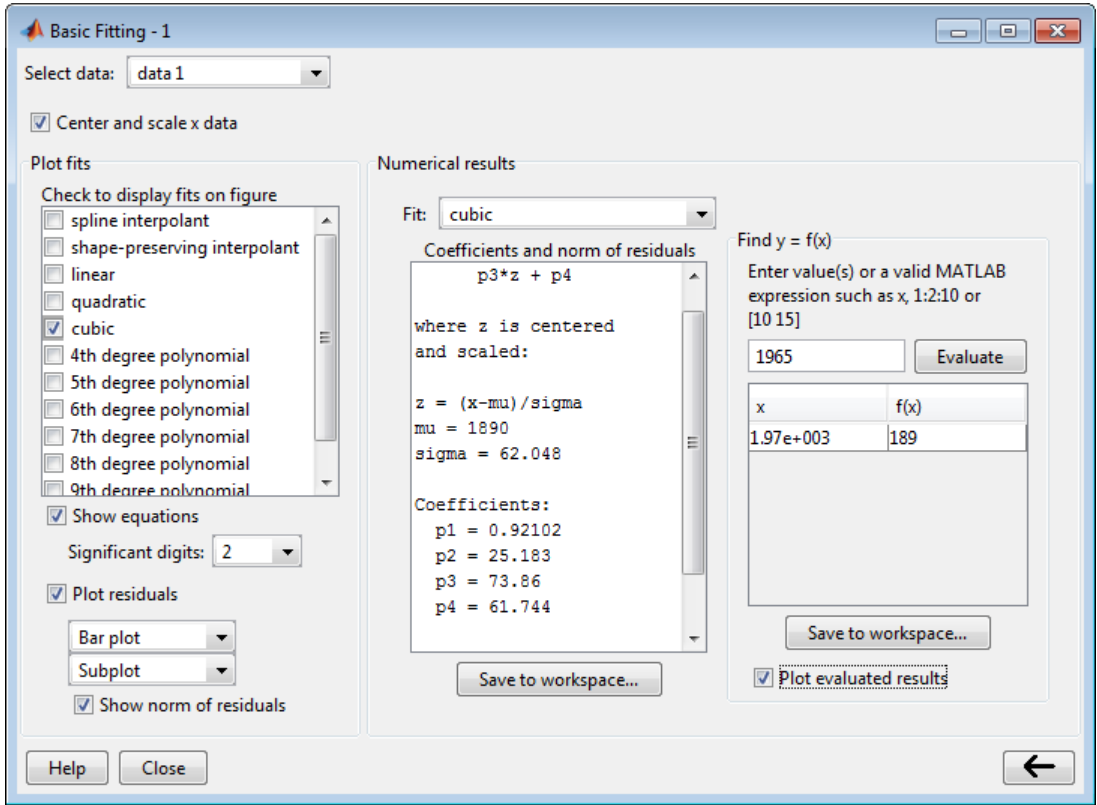
- 1 在“基本拟合”对话框中，点击  按钮以指定评估当前拟合的 x 值的向量。
- 2 在**输入值...** 字段中，键入以下值：

1965

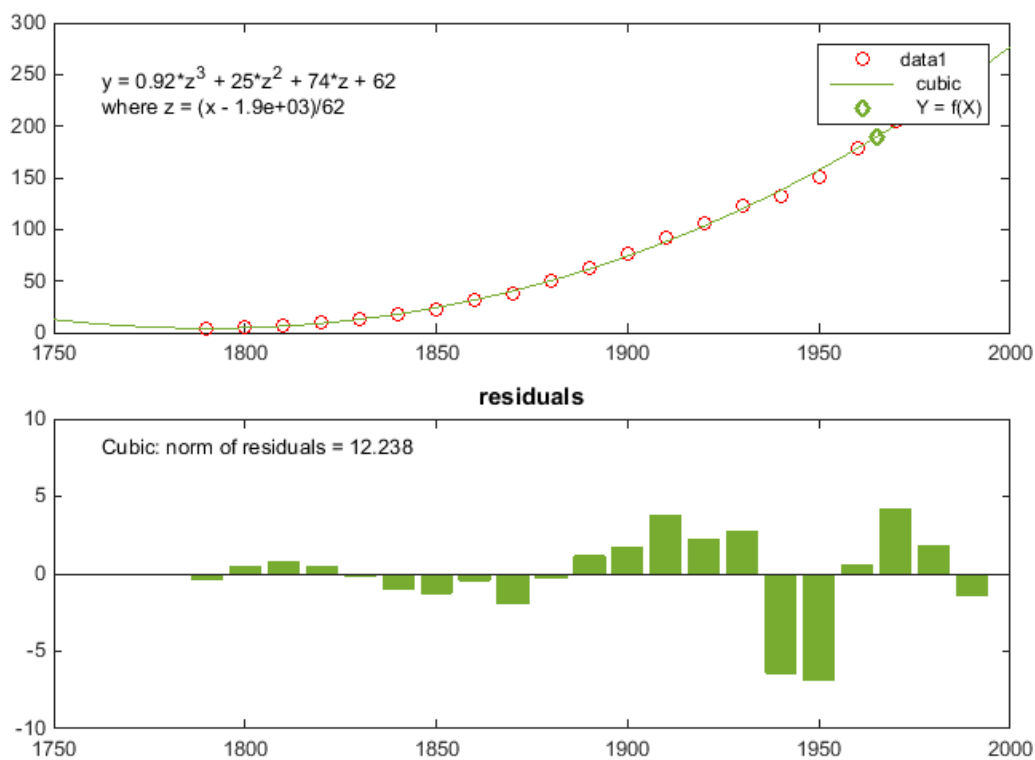
注意 使用未经缩放和中心化的 x 值。您不需要先进行中心化和缩放，即使您在“通过三次多项式拟合预测人口普查数据”（第 2-17 页）中选择了缩放 x 值以获得系数。“基本拟合”工具会在后台进行必要的调整。

- 3 点击**计算**。

x 值和 $f(x)$ 的对应值根据拟合进行计算并显示在表中，如下所示：

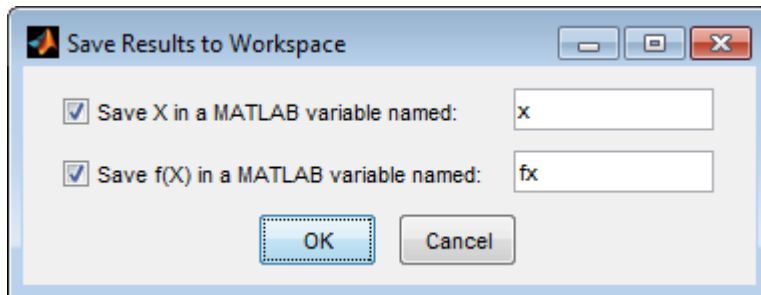


4 选中绘制计算结果图复选框以将插值显示为菱形标记：



- 5 点击**保存到工作区**，将 1965 年的人口插值保存到 MATLAB 工作区。

这将打开以下对话框，您可以在其中指定变量名称：



- 6 点击**确定**，但如果您希望执行下一节“生成代码文件以重新生成结果”（第 2-28 页）中的步骤，则要使图窗窗口保持打开状态。

生成代码文件以重新生成结果

完成基本拟合会话后，您可生成 MATLAB 代码，该代码使用新数据重新计算拟合并重新生成绘图。

- 1 在图窗窗口中，选择**文件 > 生成代码**。

这将创建一个函数并在 MATLAB 编辑器中显示它。该代码向您说明如何以编程方式重现您与“基本拟合”对话框的交互操作。

- 2 将第一行上的函数名称从 `createfigure` 更改为更具体的名称，如 `censusplot`。将代码文件保存至当前文件夹，文件名为 `censusplot.m`。该函数的开头为：

```
function censusplot(X1, Y1, valuesToEvaluate1)
```

- 3 生成一些新的随机扰动的人口普查数据：

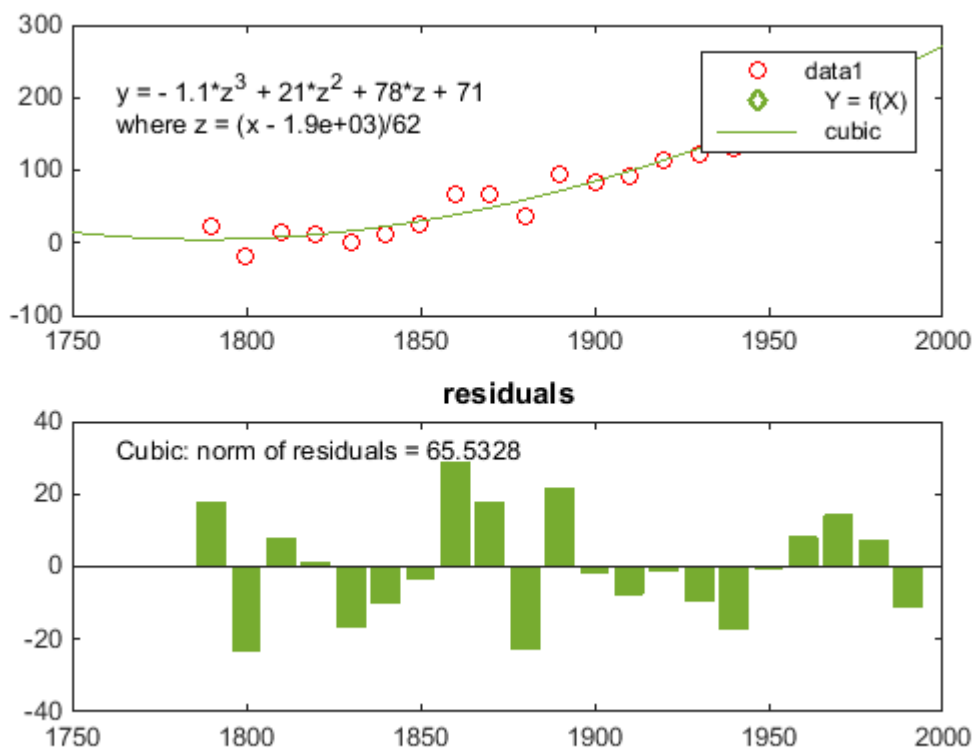
```
randpop = pop + 10*randn(size(pop));
```

- 4 用新数据重新生成绘图并重新计算拟合：

```
censusplot(cdate,randpop,1965)
```

您需要三个输入参数：在原始图形中绘制的 x, y 值 (**data 1**)，以及用于标记的 x 值。

下图显示所生成的代码产生的绘图。新绘图与您生成的代码所产生的图窗外观相匹配，但 y 数据值、三次拟合的方程以及条形图中的残差值不同，这在意料之中。



了解基本拟合工具如何计算拟合

基本拟合工具调用 **polyfit** 函数计算多项式拟合。它调用 **polyval** 函数评估拟合。**polyfit** 分析它的输入，以确定数据的条件是否适合要求的拟合度。

若发现病态数据，**polyfit** 将尽其所能计算回归，但还会返回警告，表示拟合可以改善。基本拟合示例一节“通过三次多项式拟合预测人口普查数据”（第 2-17 页）显示了该警告。

改善模型可靠性的一种方法是增加数据点。但是，向数据集添加观测值并非始终可行。替代策略是转换预测变量，对其中心化和缩放进行归一化。（在示例中，预测变量是人口普查日期的向量。）

通过计算 z 值，**polyfit** 函数进行归一化：

$$z = \frac{x - \mu}{\sigma}$$

其中， x 是预测变量数据， μ 是 x 的均值， σ 是 x 的标准差。 z 值为数据提供均值 0 和标准差 1。在基本拟合用户界面中，您通过选中**中心化并缩放 x 数据**复选框，将预测变量数据转换为 z 值。

中心化并缩放后，计算 y 数据作为 z 的函数时的模型系数。这些数据与 y 作为 x 的函数时计算所得的系数不同，且更为稳定。模型的形式与残差范数不变。基本拟合用户界面会自动重新缩放 z 值，以便按与原始 x 数据相同的比例绘制拟合图。

若要了解如何将中心化并缩放后的数据作为中间数据创建最终绘图，请在命令行窗口中运行以下代码：

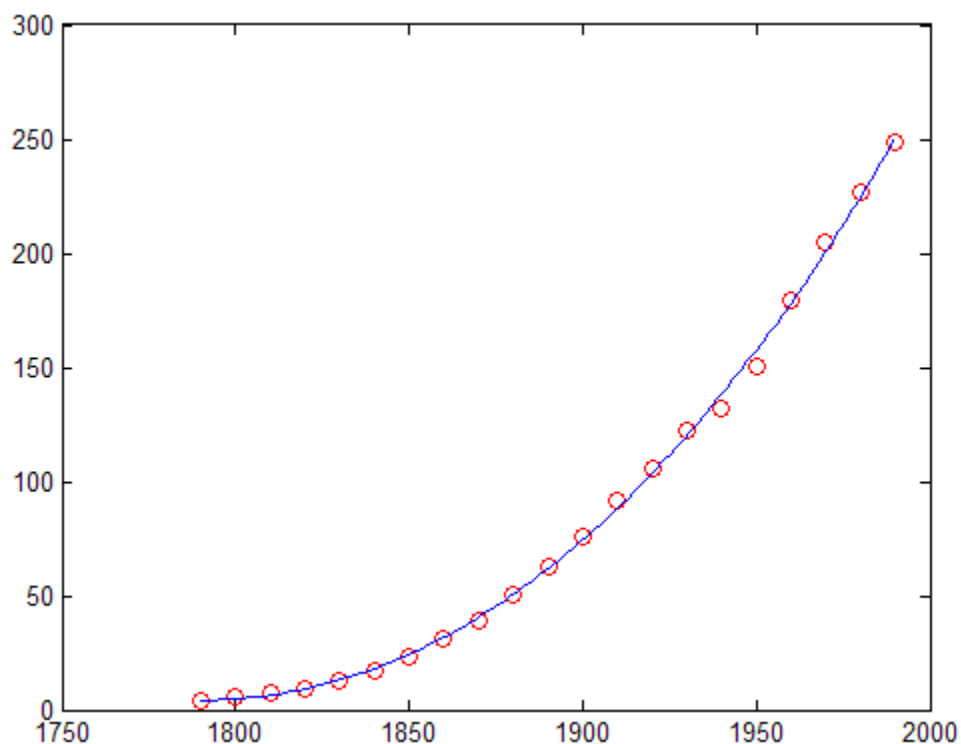
```
close
load census
x = cdate;
y = pop;
z = (x-mean(x))/std(x); % Compute z-scores of x data

plot(x,y,'ro') % Plot data as red markers
hold on % Prepare axes to accept new graph on top

zfit = linspace(z(1),z(end),100);
pz = polyfit(z,y,3); % Compute conditioned fit
yfit = polyval(pz,zfit);

xfit = linspace(x(1),x(end),100);
plot(xfit,yfit,'b-') % Plot conditioned fit vs. x data
```

中心化并缩放后的三次多项式绘制为一条蓝线，如下所示：



在代码中， z 的计算说明了如何对数据进行归一化。如果您在调用 `polyfit` 函数时提供三个返回变量，该函数将自行执行转换：

```
[p,S,mu] = polyfit(x,y,n)
```

返回的回归参数 p 现在基于归一化的 x 。返回的向量 mu 包含 x 的均值和标准差。有关详细信息，请参阅 `polyfit` 参考页。

以编程方式拟合

本节内容
“适用于多项式模型的 MATLAB 函数”（第 2-32 页）
“带非多项式项的线性模型”（第 2-37 页）
“多次回归”（第 2-39 页）
“以编程方式拟合”（第 2-40 页）

适用于多项式模型的 MATLAB 函数

有两个 MATLAB 函数可通过多项式对您的数据建模。

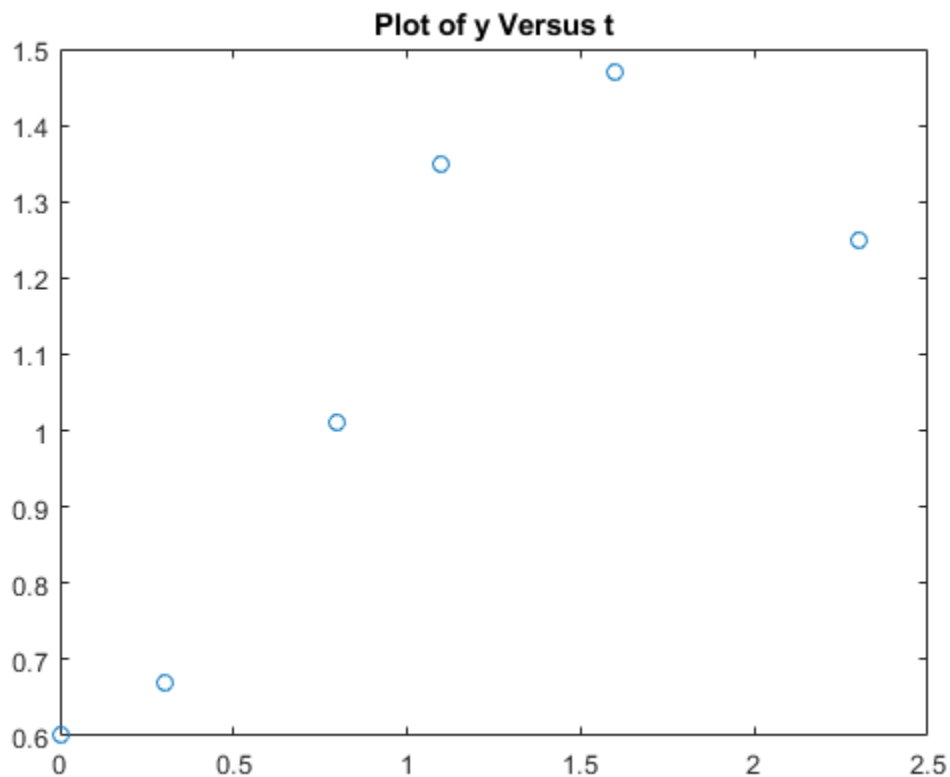
多项式拟合函数

函数	说明
<code>polyfit</code>	<code>polyfit(x,y,n)</code> 通过最大限度地减小数据与模型偏差的平方和（最小二乘拟合），求拟合 <code>y</code> 数据的 <code>n</code> 次多项式 <code>p(x)</code> 的系数。
<code>polyval</code>	<code>polyval(p,x)</code> 返回基于 <code>x</code> 进行计算，且由 <code>polyfit</code> 确定的 <code>n</code> 次多项式的值。

此示例说明如何使用多项式对数据建模。

在时间 `t` 的多个值处测量数量 `y`。

```
t = [0 0.3 0.8 1.1 1.6 2.3];  
y = [0.6 0.67 1.01 1.35 1.47 1.25];  
plot(t,y,'o')  
title('Plot of y Versus t')
```



您可尝试使用以下二次多项式函数对此数据进行建模：

$$y = a_2 t^2 + a_1 t + a_0 .$$

通过最大限度地减小数据与模型偏差的平方和（最小二乘拟合），计算未知系数 a_0 、 a_1 和 a_2 。

使用 `polyfit` 求多项式系数。

```
p = polyfit(t,y,2)
```

```
p = 1×3
```

-0.2942 1.0231 0.4981

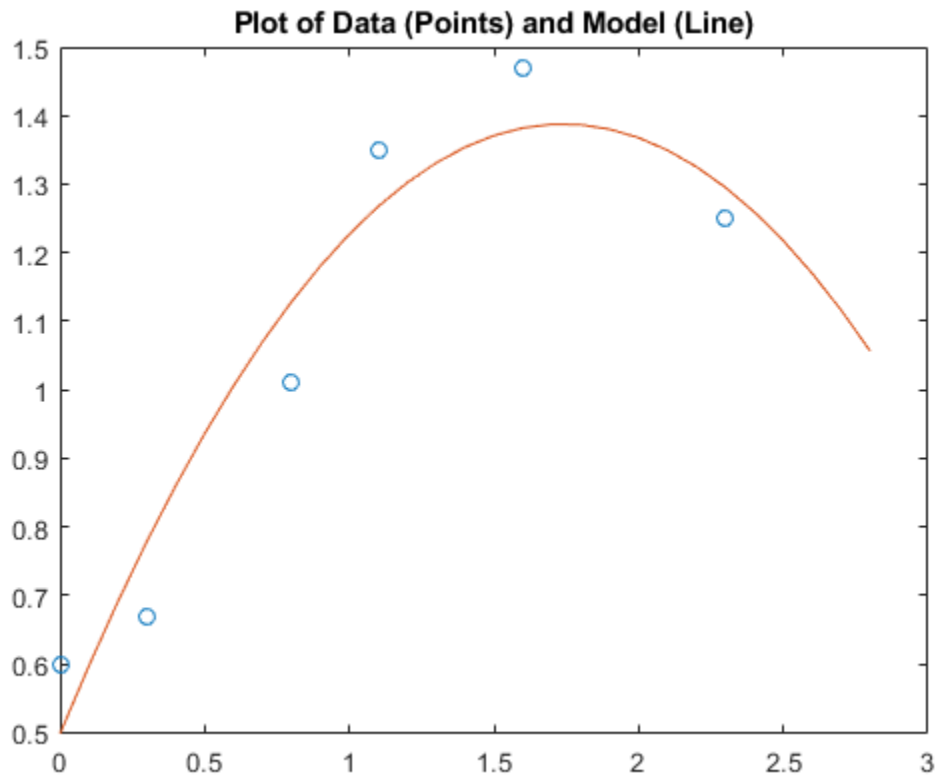
MATLAB 以降幂计算多项式系数。

数据的二次多项式模型由以下方程给出：

$$y = -0.2942t^2 + 1.0231t + 0.4981.$$

按统一的时间间隔 t_2 计算多项式。然后，在同一个图中绘制原始数据和模型。

```
t2 = 0:0.1:2.8;  
y2 = polyval(p,t2);  
figure  
plot(t,y,'o',t2,y2)  
title('Plot of Data (Points) and Model (Line)')
```

按数据时间向量计算模型

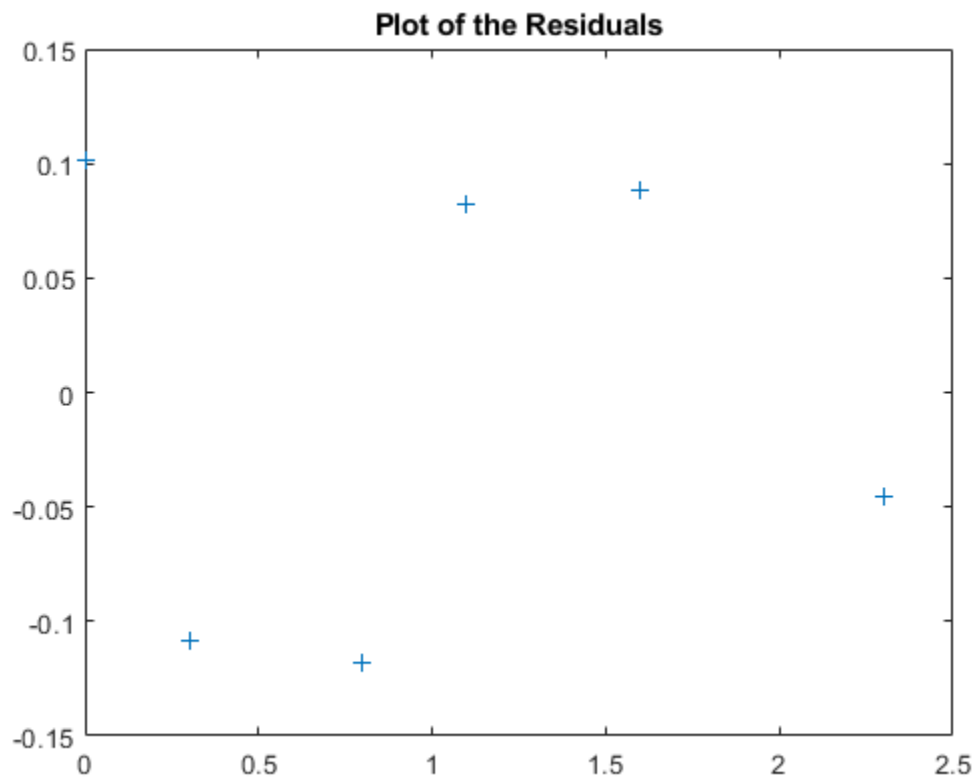
```
y2 = polyval(p,t);
```

计算残差。

```
res = y - y2;
```

绘制残差图。

```
figure, plot(t,res,'+')  
title('Plot of the Residuals')
```



请注意，二阶拟合大致遵循数据的基本形状，但并不能捕获数据似乎具备的平滑曲线。残差似乎存在一个模式，意味着可能需要不同的模型。如下所示，五次多项式在遵循数据波动方面表现更佳。

重复该练习，不过这次使用来自 `polyfit` 的五次多项式。

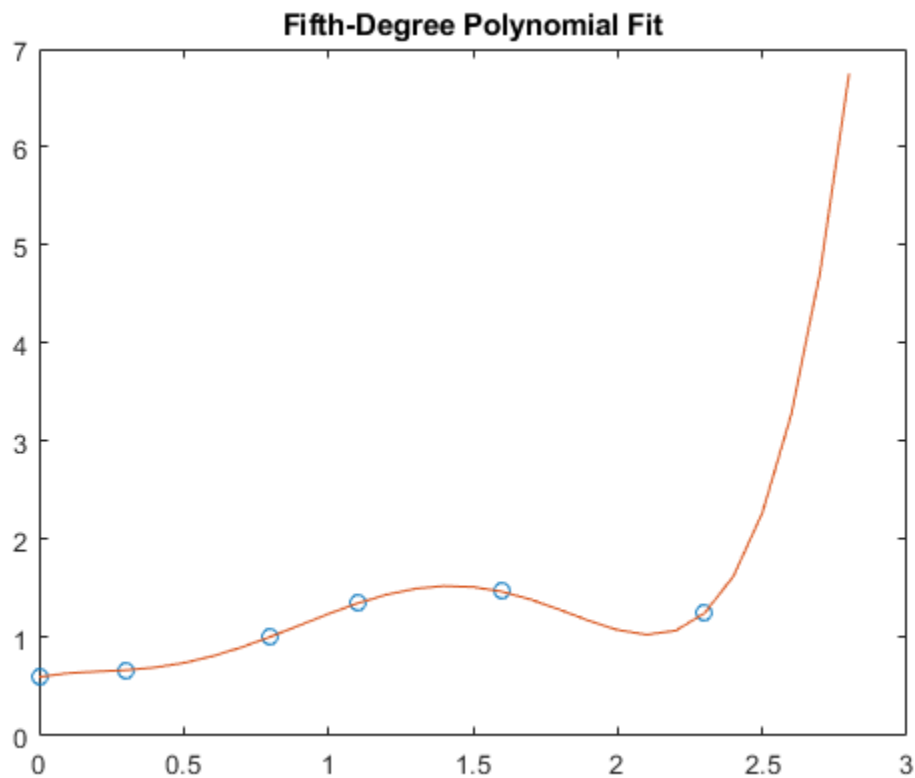
```
p5 = polyfit(t,y,5)
```

```
p5 = 1×6
```

```
0.7303 -3.5892 5.4281 -2.5175 0.5910 0.6000
```

在 `t2` 上计算多项式，并在新的图窗窗口中基于数据绘制拟合图。

```
y3 = polyval(p5,t2);  
figure  
plot(t,y,'o',t2,y3)  
title('Fifth-Degree Polynomial Fit')
```



注意 如果您尝试对物理情况建模，务必考虑特定阶次的模型是否对您的情况有意义。

带非多项式项的线性模型

此示例说明如何使用含有非多项式项的线性模型拟合数据。

若多项式函数并未得出适合您数据的满意模型，您可尝试使用带非多项式项的线性模型。以如下函数为例，它在参数 a_0 、 a_1 和 a_2 中为线性，而在 t 数据中为非线性：

$$y = a_0 + a_1 e^{-t} + a_2 t e^{-t}.$$

您可通过构建及求解一组联立方程并为参数求解，计算未知系数 a_0 、 a_1 和 a_2 。以下语法通过构建一个设计矩阵实现此目的，该矩阵中的每一列代表用于预测响应（模型中的项）的变量，每一行对应于这些变量的一个观测值。

输入 t 和 y 作为列向量。

```
t = [0 0.3 0.8 1.1 1.6 2.3]';  
y = [0.6 0.67 1.01 1.35 1.47 1.25]';
```

构建设计矩阵。

```
X = [ones(size(t)) exp(-t) t.*exp(-t)];
```

计算模型系数。

```
a = X\y
```

```
a = 3×1
```

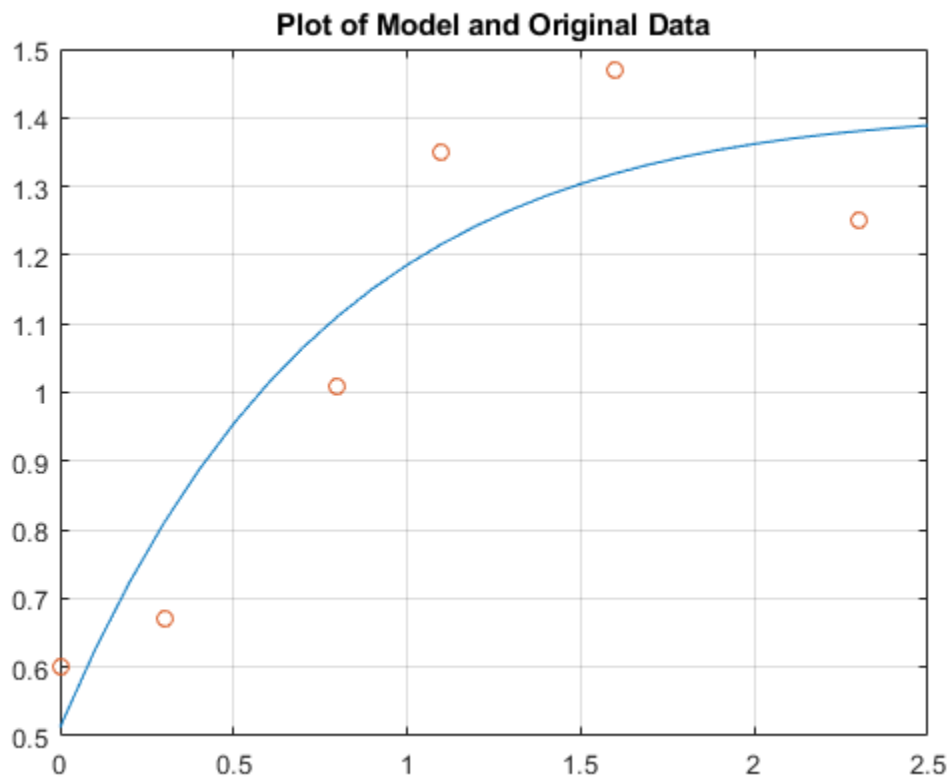
```
1.3983  
-0.8860  
0.3085
```

因此，该数据的模型由以下公式提供：

$$y = 1.3983 - 0.8860e^{-t} + 0.3085te^{-t}.$$

现在以等间距的点评估模型，并以原始数据绘制模型。

```
T = (0:0.1:2.5)';  
Y = [ones(size(T)) exp(-T) T.*exp(-T)]*a;  
plot(T,Y,'-',t,y,'o'), grid on  
title('Plot of Model and Original Data')
```



多次回归

此示例说明如何使用多次回归对具有多个预测变量的函数进行数据建模。

若 y 是具有多个预测变量的函数，则必须对表示各变量之间关系的矩阵方程进行扩展，以容纳额外的数据。这称为多次回归。

为多个 x_1 和 x_2 值测量对应的量 y 。将这些值分别存储在向量 $\mathbf{x1}$ 、 $\mathbf{x2}$ 和 \mathbf{y} 中。

```

$$\begin{aligned}\mathbf{x1} &= [.2 \ .5 \ .6 \ .8 \ 1.0 \ 1.1]'; \\ \mathbf{x2} &= [.1 \ .3 \ .4 \ .9 \ 1.1 \ 1.4]'; \\ \mathbf{y} &= [.17 \ .26 \ .28 \ .23 \ .27 \ .24]';\end{aligned}$$

```

此数据的模型采用以下形式：

$$y = a_0 + a_1x_1 + a_2x_2.$$

多次回归可通过最大限度地减小数据与模型偏差的平方和（最小二乘拟合），对未知系数 a_0 、 a_1 和 a_2 求解。

通过构建设计矩阵 \mathbf{X} ，构建和求解一组联立方程。

```
X = [ones(size(x1)) x1 x2];
```

使用反斜杠运算符对参数求解。

```
a = X\y
```

```
a = 3×1
```

```
0.1018
```

```
0.4844
```

```
-0.2847
```

数据的最小二乘拟合模型为

$$y = 0.1018 + 0.4844x_1 - 0.2847x_2.$$

为了验证该模型，请求出数据与模型偏差绝对值的最大值。

```
Y = X*a;
```

```
MaxErr = max(abs(Y - y))
```

```
MaxErr = 0.0038
```

该值远小于任何数据值，表明该模型能够准确贴合数据。

以编程方式拟合

此示例说明如何使用 MATLAB 函数执行以下操作：

- “计算相关系数”（第 2-42 页）
- “对数据进行多项式拟合”（第 2-42 页）

- “绘制和计算置信边界”（第 2-44 页）

从 `census.mat` 加载样本人口普查数据，它含有 1790 年至 1990 年的美国人口数据。

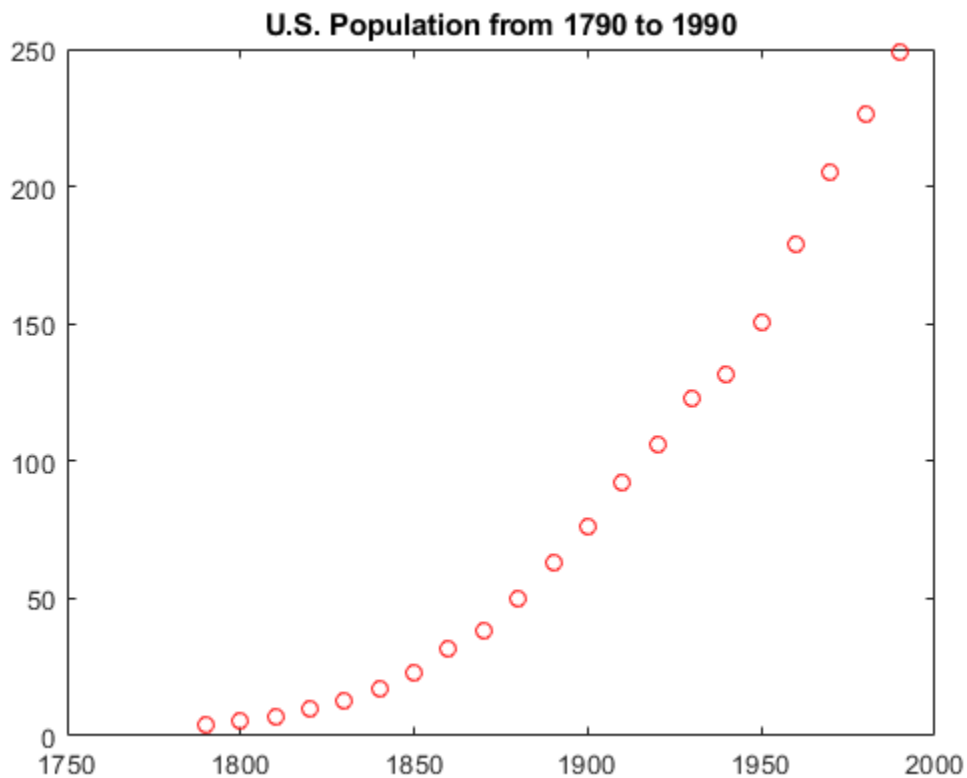
```
load census
```

这会向 MATLAB 工作区添加以下两个变量。

- `cdate` 是包含从 1790 年到 1990 年（以 10 为增量）的年份的列向量。
- `pop` 是对应于 `cdate` 中每一年的美国人口数字的列向量。

绘制数据图。

```
plot(cdate,pop,'ro')  
title('U.S. Population from 1790 to 1990')
```



该图呈现出一个明显的模式，表明各变量之间存在高相关性。

计算相关系数

在示例的这一部分，您可确定变量 **cdate** 和 **pop** 之间的统计相关性，以证明数据建模的合理性。若要了解关于相关系数的详细信息，请参阅“线性相关性”（第 2-2 页）。

计算相关系数矩阵。

```
corrcoef(cdate,pop)
```

```
ans = 2×2
```

```
1.0000  0.9597  
0.9597  1.0000
```

对角矩阵元素等于 1，代表每个变量与其自身具有完美相关性。非对角矩阵非常接近 1，表明变量 **cdate** 和 **pop** 之间存在较强的统计相关性。

对数据进行多项式拟合

示例的这一部分应用 **polyfit** 和 **polyval** MATLAB 函数对数据建模。

计算拟合参数。

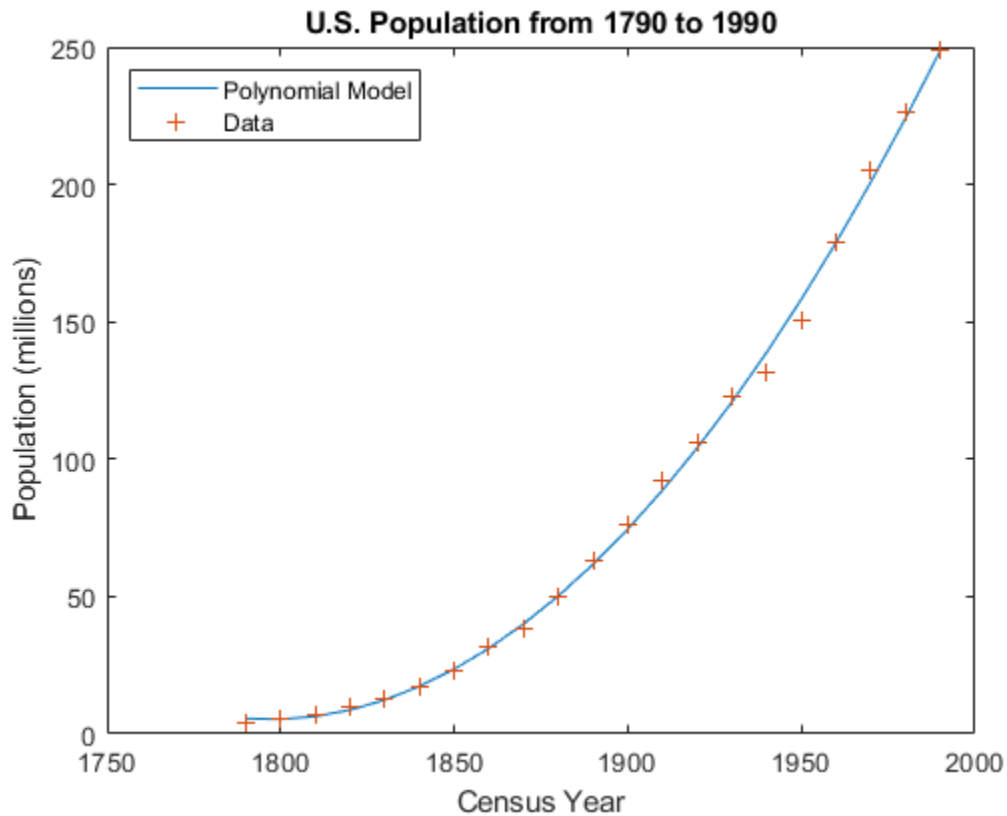
```
[p>ErrorEst] = polyfit(cdate,pop,2);
```

评估拟合。

```
pop_fit = polyval(p,cdate>ErrorEst);
```

绘制数据图及拟合图。

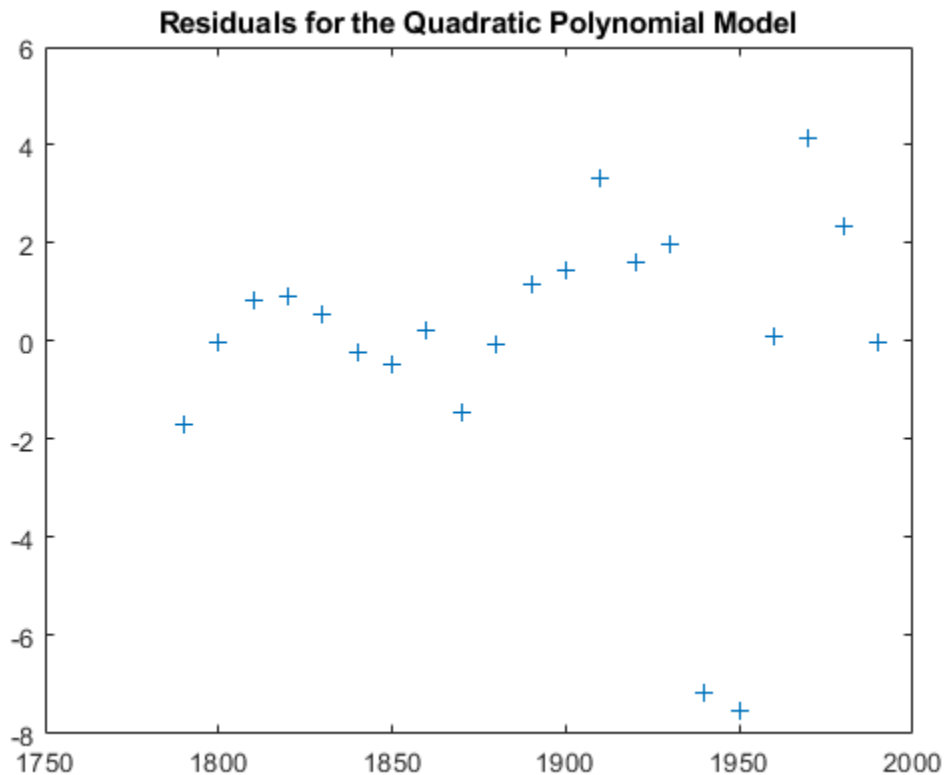
```
plot(cdate,pop_fit,'-',cdate,pop,'+');  
title('U.S. Population from 1790 to 1990')  
legend('Polynomial Model','Data','Location','NorthWest');  
xlabel('Census Year');  
ylabel('Population (millions)');
```

该图显示，二次多项式拟合可提供良好的数据近似度。

计算此拟合的残差。

```
res = pop - pop_fit;  
figure, plot(cdate,res,'+')  
title('Residuals for the Quadratic Polynomial Model')
```



请注意，残差图显示出一种模式，该模式表示二次多项式可能不适合用于此数据的建模。

绘制和计算置信边界

置信边界是预测响应的置信区间。该区间的宽度表示拟合的确定度。

示例的这一部分将 `polyfit` 和 `polyval` 应用于 `census` 样本数据，为二次多项式模型产生置信边界。

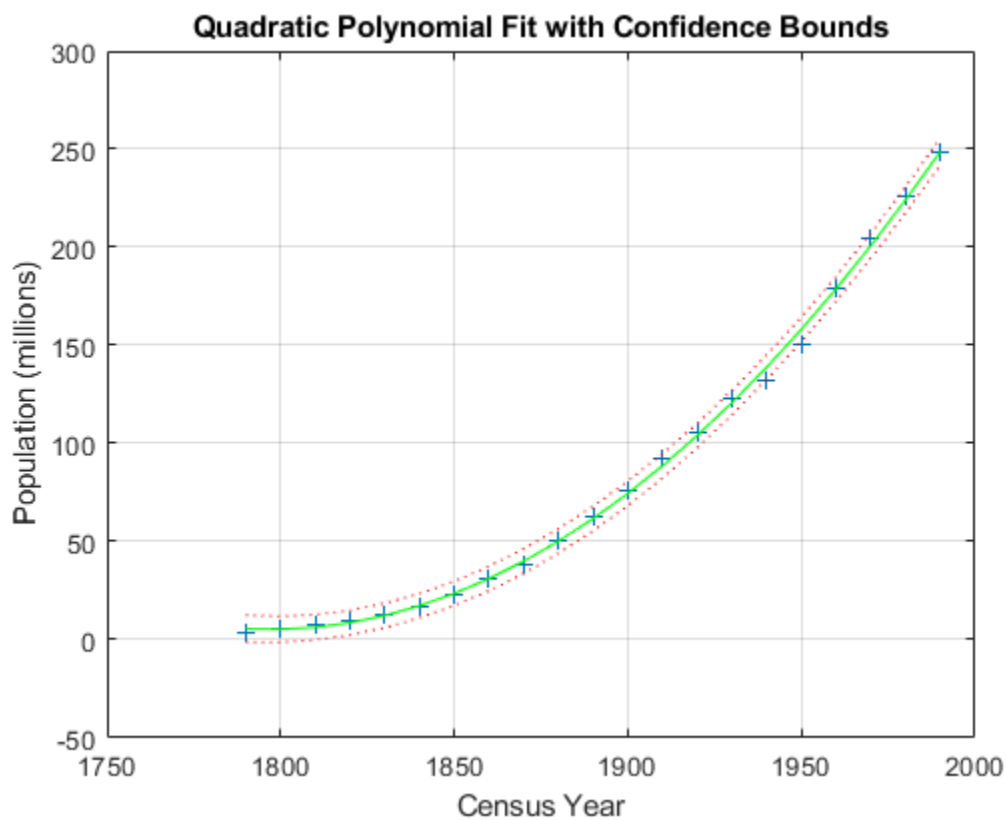
以下代码使用 $\pm 2\Delta$ 的区间，对应于大型样本的 95% 置信区间。

评估拟合和预测误差估计 (Δ)。

```
[pop_fit,delta] = polyval(p,cdate>ErrorEst);
```

绘制数据图、拟合图和置信边界。

```
plot(cdate,pop,'+',...
      cdate,pop_fit,'g-',...
      cdate,pop_fit+2*delta,'r:',...
      cdate,pop_fit-2*delta,'r:');
xlabel('Census Year');
ylabel('Population (millions)');
title('Quadratic Polynomial Fit with Confidence Bounds')
grid on
```



95% 区间表示新观测值有 95% 的可能性会落在范围内。

时序分析

- “什么是时序？”（第 3-2 页）
- “时序对象”（第 3-3 页）

什么是时序？

时序是按照顺序（通常是按固定间隔）在一段时间中采样的数据向量。它们与构成许多其他数据分析基础的随机采样数据有所不同。时序表示动态规模或过程的时间演化。时序线性排序通过一组专门的方法为时序提供一个独特的数据分析位置。

时序分析与以下项目有关：

- 识别模式
- 建模模式
- 预测值

有几个专用的 MATLAB 函数执行时序分析。本节介绍用于时序分析的对象和交互式工具。

时序对象

本节内容
“时序的类型及其用途”（第 3-3 页）
“时序数据样本”（第 3-3 页）
“示例：时序对象和方法”（第 3-5 页）
“时序构造函数”（第 3-16 页）
“时序集合构造函数”（第 3-17 页）

时序的类型及其用途

MATLAB 时序对象有两种类型：

- **timeseries** - 存储数据和时间值，以及包含单位、事件、数据质量和插值方法的元数据信息
- **tscollection** - 存储共享公共时间向量的 **timeseries** 对象的集合，便于对具有不同单位的同步时序执行操作

本节讨论以下主题：

- 使用时序构造函数来实例化时序类
- 使用 **set** 方法或圆点表示法修改对象属性
- 调用时序函数和方法

要快速了解有关使用 **timeseries** 和 **tscollection** 对象进行编程的信息，请按照“示例：时序对象和方法”（第 3-5 页）中的步骤进行操作。

时序数据样本

要正确理解本文档中有关 **timeseries** 对象属性和方法的说明，必须了解与在 **timeseries** 对象中存储数据相关的一些术语 - 尤其是数据值和数据样本的区别。

数据值是在特定时间记录的单个标量值。数据样本由 **timeseries** 对象中与特定时间关联的一个或多个值组成。一个时序中的数据样本数量与时间向量的长度相同。

以由三个传感器信号组成的数据为例：其中两个信号表示对象的位置，以米为单位；第三个信号表示其速度，以米/秒为单位。

要输入数据矩阵，请在 MATLAB 提示符下键入以下内容：

```
x = [-0.2 -0.3 13;  
     -0.1 -0.4 15;  
      NaN  2.8 17;  
      0.5  0.3 NaN;  
     -0.3 -0.1 15]
```

NaN 值代表缺失的数据值。MATLAB 将显示以下 5×3 矩阵：

```
x=  
-0.2000 -0.3000 13.0000  
-0.1000 -0.4000 15.0000  
NaN      2.8000 17.0000  
0.5000   0.3000   NaN  
-0.3000 -0.1000 15.0000
```

x 的前两列包含具有相同单位的数量，您可以创建多变量 **timeseries** 对象来存储这两个时序。有关创建 **timeseries** 对象的详细信息，请参阅“时序构造函数”（第 3-16 页）。以下命令创建 **timeseries** 对象 **ts_pos** 来存储位置值：

```
ts_pos = timeseries(x(:,1:2), 1:5, 'name', 'Position')
```

MATLAB 通过显示 **ts_pos** 的以下属性作出响应：

timeseries

Common Properties:

 Name: 'Position'

 Time: [5x1 double]

 TimeInfo: [1x1 tsdata.timemetadata]

 Data: [5x2 double]

 DataInfo: [1x1 tsdata.datametadata]

More properties, Methods

时间向量的 **Length**（在本示例中为 5）等于 **timeseries** 对象中的数据样本数。通过在 MATLAB 提示符下键入以下内容，在 **ts_pos** 中找到数据样本的大小：

```
getdatasamplesize(ts_pos)
```

```
ans =
```

```
1    2
```


同样，您可以创建另一个 **timeseries** 对象来存储速度数据：

```
ts_vel = timeseries(x(:,3), 1:5, 'name', 'Velocity');
```

通过键入以下内容在 **ts_vel** 中找到每个数据样本的大小：

```
getdatasamplesize(ts_vel)
```

```
ans =
```

```
1    1
```

请注意，**ts_vel** 在每个数据样本中有一个数据值，而 **ts_pos** 在每个数据样本中有两个数据值。

注意 通常，当时序数据是具有 M 个样本的 $M \times N \times P \times \dots$ 多维数组时，每个数据样本的大小为 $N \times P \times \dots$ 。

如果您要对 **ts_pos** 和 **ts_vel** **timeseries** 对象执行操作且同时使它们保持同步，请将它们组合到一个时序集合中。有关详细信息，请参阅“时序集合构造函数语法”（第 3-17 页）。

示例：时序对象和方法

- “创建时序对象”（第 3-6 页）
- “查看时序对象”（第 3-7 页）
- “修改时序单位和插值方法”（第 3-9 页）
- “定义事件”（第 3-9 页）
- “创建时序集合对象”（第 3-10 页）
- “对时序集合对象重新采样”（第 3-11 页）
- “将数据样本添加到时序集合对象”（第 3-12 页）
- “删除缺失数据和对其进行插值”（第 3-13 页）
- “从时序集合中删除时序”（第 3-14 页）
- “将时间向量值显示为日期字符串”（第 3-14 页）
- “绘制时序集合成员”（第 3-15 页）

创建时序对象

示例的此部分说明如何从数组中创建若干个 `timeseries` 对象。有关 `timeseries` 对象的详细信息，请参阅“时序构造函数”（第 3-16 页）。

将样本数据从 `count.dat` 导入 MATLAB 工作区。

```
load count.dat
```

这会将 24×3 矩阵 `count` 添加到工作区中。`count` 中的每一列分别代表三个城镇十字路口中每个十字路口每小时经过的车辆计数。

查看 `count` 矩阵。

```
count
```

创建三个 `timeseries` 对象来存储在每个十字路口收集的数据。

```
count1 = timeseries(count(:,1), 1:24,'name','intersection1');  
count2 = timeseries(count(:,2), 1:24,'name','intersection2');  
count3 = timeseries(count(:,3), 1:24,'name','intersection3');
```

注意 在上述构造中，`timeseries` 对象同时具有变量名称（例如 `count1`）和内部对象名称（例如 `intersection1`）。变量名称与 MATLAB 函数结合使用。对象名称是对象的属性，可以使用对象方法进行访问。有关 `timeseries` 对象属性和方法的详细信息，请参阅“时序属性”（第 3-16 页）和“时序方法”（第 3-16 页）。

默认情况下，一个时序具有一个时间向量，该时间向量的单位为秒，开始时间为 0 秒。此示例构造开始时间为 1 秒、结束时间为 24 秒、增量为 1 秒的 `count1`、`count2` 和 `count3` 时序对象。您将在“修改时序单位和插值方法”（第 3-9 页）中将时间单位更改为小时。

注意 如果要创建一个用于组合 `count` 中的三个数据列的 `timeseries` 对象，请使用以下语法：

```
count_ts = timeseries(count, 1:24,'name','traffic_counts')
```

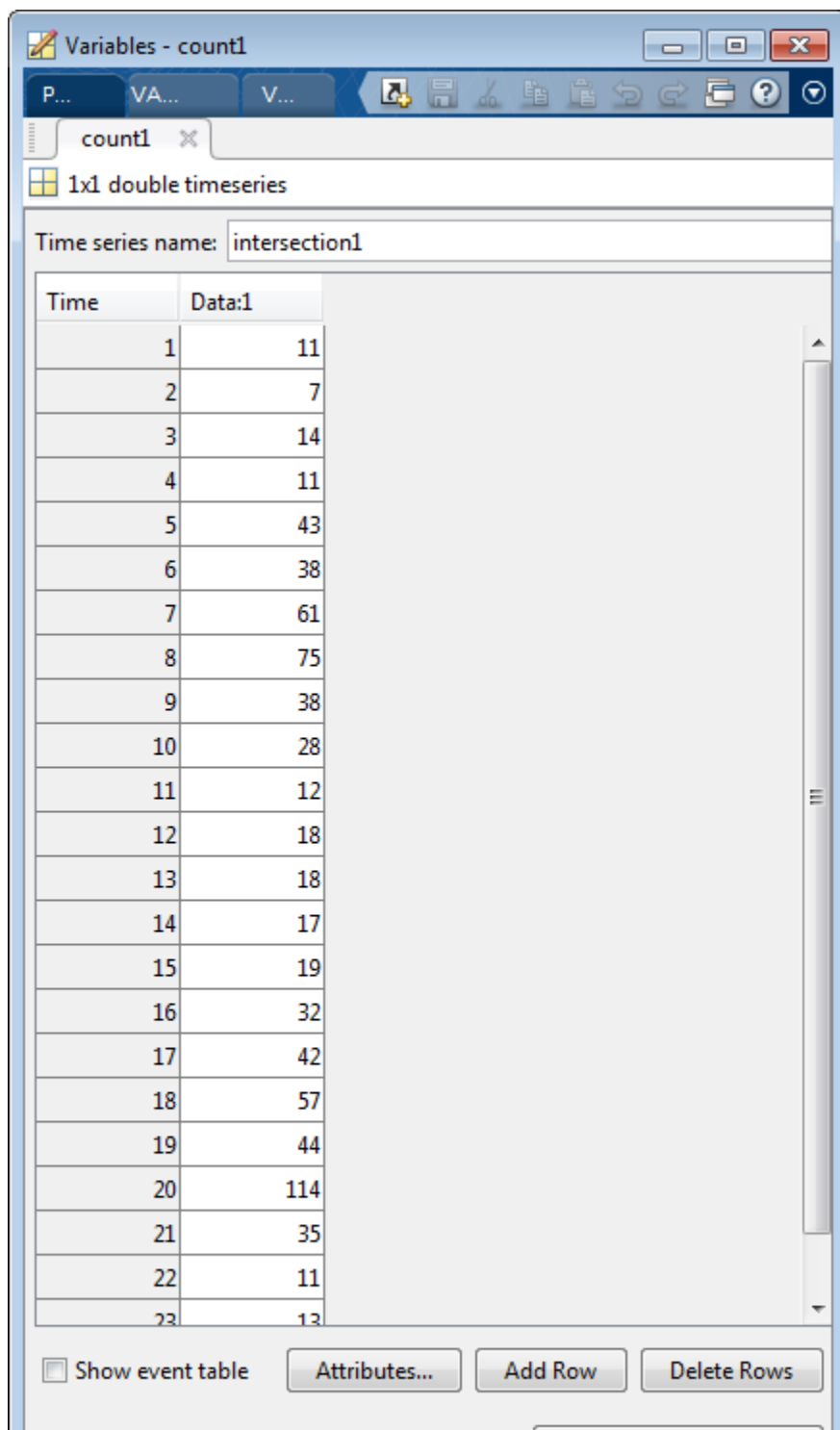
当所有时序具有相同的单位并且您要在计算过程中使它们保持同步时，这很有用。

查看时序对象

按照“创建时序对象”（第 3-6 页）中的描述创建 **timeseries** 对象后，您可以在变量编辑器中查看它。

要在变量编辑器中查看 **timeseries** 对象，如 **count1**，请使用以下任一方法：

- 在命令提示符下键入 **open('count1')**。
- 在**主页**选项卡上的**变量**部分中，点击**打开变量**并选择 **count1**。此方法在 MATLAB Online 中不可用。



修改时序单位和插值方法

如“创建时序对象”（第 3-6 页）中所述创建 `timeseries` 对象后，可以使用圆点表示法修改其单位和插值方法。

查看 `count1` 的当前属性。

```
get(count1)
```

MATLAB 显示 `count1 timeseries` 对象的当前属性值。

使用圆点表示法查看当前 `DataInfo` 属性。

```
count1.DataInfo
```

将 `count1` 的数据单位更改为 `'cars'`。

```
count1.DataInfo.Units = 'cars';
```

将 `count1` 的插值方法设置为零阶保持。

```
count1.DataInfo.Interpolation = tsdata.interpolation('zoh');
```

确认 `DataInfo` 属性已修改。

```
count1.DataInfo
```

将三个时序的时间单位修改为 `'hours'`。

```
count1.TimeInfo.Units = 'hours';
```

```
count2.TimeInfo.Units = 'hours';
```

```
count3.TimeInfo.Units = 'hours';
```

定义事件

示例的此部分说明如何使用 `tsdata.event` 辅助对象为 `timeseries` 对象定义事件。事件在特定时间标记数据。绘制数据时，事件标记会显示在绘图上。事件还是同步多个时序的一种便捷方式。

在数据中添加两个事件，分别标记早上和下午的通勤时间。

构造第一个事件并将其添加到所有时序。第一个事件发生在早上 8 点。

```
e1 = tsdata.event('AMCommute',8);
```

```
e1.Units = 'hours';           % Specify the units for time
```

```
count1 = addevent(count1,e1); % Add the event to count1
count2 = addevent(count2,e1); % Add the event to count2
count3 = addevent(count3,e1); % Add the event to count3
```

构造第二个事件并将其添加到所有时序。第二个事件发生在下午 6 点。

```
e2 = tsdata.event('PMCommute',18);
e2.Units = 'hours'; % Specify the units for time
count1 = addevent(count1,e2); % Add the event to count1
count2 = addevent(count2,e2); % Add the event to count2
count3 = addevent(count3,e2); % Add the event to count3
```

对时序 count1 绘图。

```
figure
plot(count1)
```

对任何时序绘图时，为时序对象定义的绘图方法都会将事件显示为标记。默认标记是红色实心圆。

绘图反映 count1 使用零阶保持插值。

绘制 count2。

```
plot(count2)
```

如果绘制时序 count2，它会替换 count1 显示。您会看到它的事件，并且它使用线性插值。

通过设置 hold on 覆盖时序绘图。

```
hold on
plot(count3)
```

创建时序集合对象

示例的此部分说明如何创建 **tscollection** 对象。集合中的每个时序称为一个成员。有关 **tscollection** 对象的详细信息，请参阅“时序集合构造函数”（第 3-17 页）。

注意 通常，您可以使用 **tscollection** 对象对具有不同单位的同步时序进行组合。在此简单示例中，所有时序具有相同的单位，因此相对于将三个时序组合到单一 **timeseries** 对象而言，**tscollection** 对象没有优势。有关如何在一个 **timeseries** 对象中组合多个时序的示例，请参阅“创建时序对象”（第 3-6 页）。

创建一个名为 `count_coll` 的 `tscollection` 对象，并使用构造函数语法立即添加当前在 MATLAB 工作区中的三个时序中的两个（稍后将添加第三个时序）。

```
tsc = tscollection({count1 count2}, 'name', 'count_coll')
```

注意 添加到 `tscollection` 的 `timeseries` 对象的时间向量必须匹配。

请注意，`timeseries` 对象的 `Name` 属性用于将集合成员命名为 `intersection1` 和 `intersection2`。

将工作区中的第三个 `timeseries` 对象添加到 `tscollection`。

```
tsc = addts(tsc, count3)
```

集合中的所有三个成员都已列出。

对时序集合对象重新采样

示例的此部分说明如何使用新时间向量对 `tscollection` 中的每个成员进行重新采样。重新采样操作用于选择在特定时间值的现有数据，或以更精细的时间间隔进行数据插值。如果新时间向量包含在前一时间向量中不存在的时间值，则使用与时序关联的默认插值方法计算新数据值。

对时序进行重新采样以包含每 2 小时（而不是每小时）的数据值，并将其保存为新 `tscollection` 对象。

```
tsc1 = resample(tsc, 1:2:24)
```

在某些情况下，您可能需要比当前更精细的信息采样，通过对数据值进行插值可实现此目的。

在每个半小时标记处进行插值。

```
tsc1 = resample(tsc, 1:0.5:24)
```

在每个半小时标记处添加值时，将使用时序的默认插值方法。例如，`intersection1` 中的新数据点是使用零阶保持插值方法计算的，该方法会保存上一个样本常量的值。按照“修改时序单位和插值方法”（第 3-9 页）中所述设置 `intersection1` 的插值方法。

使用线性插值（这是默认方法）计算 `intersection2` 和 `intersection3` 中的新数据点。

对具有标记的 `tsc1` 成员进行绘图以查看插值结果。

```
hold off          % Allow axes to clear before plotting
plot(tsc1.intersection1,'-xb','Displayname','Intersection 1')
```

您可以看到以半小时为间隔对数据点进行了插值，并且 Intersection 1 使用零阶保持插值，而其他两个成员使用线性插值。

在您将其他两个成员添加到绘图时，请保持图形位于图窗中。由于 `plot` 方法不显示轴标签，而 `hold` 为 `on`，因此还要添加一个图例来描述这三个序列。

```
hold on
plot(tsc1.intersection2,'-xm','Displayname','Intersection 2')
plot(tsc1.intersection3,'-xr','Displayname','Intersection 3')
legend('show','Location','NorthWest')
```

将数据样本添加到时序集合对象

示例的此部分说明如何将数据样本添加到 `tscollection`。

在 3.25 小时（即该整点时间后 15 分钟）处向 `intersection1` 集合成员添加一个数据样本。

```
tsc1 = addsampletocollection(tsc1,'time',3.25,...
    'intersection1',5);
```

`tsc1` 集合中有三个成员，在 3.25 小时处向一个成员添加数据样本时，也会向其他两个成员添加数据样本。但是，由于您未在新样本中指定 `intersection2` 和 `intersection3` 的数据值，因此对于这些成员，缺失值由 `NaN` 表示。要了解如何删除缺失数据值或对其进行插值，请参阅“删除缺失数据”（第 3-13 页）和“对缺失数据进行插值”（第 3-13 页）。

从 2.0 到 3.5 小时的 `tsc1` 数据

小时	Intersection 1	Intersection 2	Intersection 3
2.0	7	13	11
2.5	7	15	15.5
3.0	14	17	20
3.25	5	NaN	NaN
3.5	14	15	14.5

要查看所有 `intersection1` 数据（包括 3.25 小时处的新样本），请键入

```
tsc1.intersection1
```


同样，要查看所有 `intersection2` 数据（包括 3.25 小时处的新样本，包含 NaN 值），请键入

```
tsc1.intersection2
```

删除缺失数据和对其进行插值

时序对象使用 NaN 来表示缺失数据。示例的此部分说明如何删除缺失数据或通过使用该成员指定的插值方法来对缺失数据进行插值。在“将数据样本添加到时序集合对象”（第 3-12 页）中，您在 3.25 小时处向 `tsc1` 集合中添加了一个新数据样本。

由于 `tsc1` 集合有三个成员，因此在 3.25 小时处向一个成员添加数据样本时，也会向其他两个成员添加数据样本。但是，由于您未在 3.25 小时处为 `intersection2` 和 `intersection3` 成员指定数据值，因此它们当前包含由 NaN 表示的缺失值。

删除缺失数据

查找并删除 `tsc1` 集合中包含 NaN 值的数据样本。

```
tsc1 = delsamplefromcollection(tsc1,'index',...
    find(isnan(tsc1.intersection2.Data)));
```

该命令一次搜索一个 `tscollection` 成员 - 在本例中为 `intersection2`。如果在 `intersection2` 中找到缺失值，将从 `tscollection` 的全部成员中删除在该时间处的数据。

注意 请使用圆点表示法语法访问 `tsc1` 集合中 `intersection2` 成员的 `Data` 属性：

```
tsc1.intersection2.Data
```

有关 `timeseries` 属性的完整列表，请参阅“时序属性”（第 3-16 页）。

对缺失数据进行插值

为了演示此示例，请在 `intersection2` 和 `intersection3` 中重新引入 NaN 值。

```
tsc1 = addsampletocollection(tsc1,'time',3.25,...
    'intersection1',5);
```

使用当前时间向量 (`tsc1.Time`) 对 `tsc1` 中的缺失值进行插值。

```
tsc1 = resample(tsc1,tsc1.Time);
```

这会通过使用线性插值替换 `intersection2` 和 `intersection3` 中的 NaN 值（线性插值是这些时序的默认插值方法）。

注意 圆点表示法 `tsc1.Time` 用于访问 `tsc1` 集合的 `Time` 属性。有关 `tscollection` 属性的完整列表，请参阅“时序集合属性”（第 3-18 页）。

例如，要查看插值后的 `intersection2` 数据，请键入

```
tsc1.intersection2
```

从 2.0 小时到 3.5 小时的新 tsc1 数据

小时	Intersection 1	Intersection 2	Intersection 3
2.0	7	13	11
2.5	7	15	15.5
3.0	14	17	20
3.25	5	16	17.3
3.5	14	15	14.5

从时序集合中删除时序

从 `tscollection` 对象 `tsc1` 中删除 `intersection3` 时序。

```
tsc1 = removets(tsc1,'intersection3')
```

现在列出了作为集合成员的两个时序。

将时间向量值显示为日期字符串

示例的此部分说明如何使用 MATLAB 日期字符串来控制数值时间向量的显示格式。有关 `timeseries` 和 `tscollection` 对象支持的 MATLAB 日期字符串格式的完整列表，请参阅 `timeseries` 参考页中的时间向量定义。

要使用日期字符串，您必须设置 `TimeInfo` 属性的 `StartDate` 字段。时间向量中的所有值均使用 `StartDate` 作为参考日期转换为日期字符串。

假设参考日期发生在 2009 年 12 月 25 日。

```
tsc1.TimeInfo.Units = 'hours';  
tsc1.TimeInfo.StartDate = '25-DEC-2009 00:00:00';
```

与您对 `count1`、`count2` 和 `count3` 时序对象的处理类似，将 `tsc1` 成员的数据单位设置为字符串 `'car count'`。

```
tsc1.intersection1.DataInfo.Units = 'car count';
tsc1.intersection2.DataInfo.Units = 'car count';
```

绘制时序集成员

要绘制时序集合中的数据，请逐个绘制其成员。

首先绘制 `tsc1` 成员 `intersection1` 的图形。

```
hold off
plot(tsc1.intersection1);
```

绘制时序集合的成员时，其时间单位显示在 `x` 轴上，其数据单位显示在 `y` 轴上。绘图标题显示为 `'Time Series Plot:<member name>'`。

如果您使用相同的图窗绘制该集合中的其他成员，则不会显示注释。当 `hold` 为 `on` 时，时序 `plot` 方法不会尝试更新标签和标题，因为序列的描述符可能不同。

在同一图窗中绘制 `intersection1` 和 `intersection2`。防止覆盖绘图，但删除轴标签和标题。添加图例并设置线条序列的 `DisplayName` 属性以作为每个成员的标签。

```
plot(tsc1.intersection1,'-xb','DisplayName','Intersection 1')
hold on
plot(tsc1.intersection2,'-xm','DisplayName','Intersection 2')
legend('show','Location','NorthWest')
```

绘图现在包括集合中的两个时序：`intersection1` 和 `intesection2`。绘制第二个图会擦除第一个图上的标签。

最后，将 `x` 轴上的日期字符串更改为 `hours`，再次绘制这两个时序集成员并包含图例。

指定集合的时间单位为 `'hours'`。

```
tsc1.TimeInfo.Units = 'hours';
```

指定时间显示格式。

```
tsc1.TimeInfo.Format = 'HH:MM';
```

用新时间单位重新创建上一个绘图。

```
hold off
plot(tsc1.intersection1,'-xb','Displayname','Intersection 1')

% Prevent overwriting plot, but remove axis labels and title.
hold on
plot(tsc1.intersection2,'-xm','Displayname','Intersection 2')
legend('show','Location','NorthWest')

% Restore the labels with the |xlabel| and |ylabel| commands and overlay a
% data grid.
xlabel('Time (hours)')
ylabel('car count')
grid on
```

有关时序的绘图选项的详细信息，请参阅 `timeseries`。

时序构造函数

在实现专门设计用于处理时序数据的各种 MATLAB 函数和方法之前，您必须创建一个 `timeseries` 对象来存储数据。有关 `timeseries` 对象构造函数语法，请参阅 `timeseries`。

有关使用构造函数的示例，请参阅“创建时序对象”（第 3-6 页）。

时序属性

有关所有 `timeseries` 对象属性的说明，请参阅 `timeseries`。可以将 `Data`、`IsTimeFirst`、`Name`、`Quality` 和 `Time` 属性指定为构造函数中的输入参数。要分配其他属性，请使用 `set` 函数或圆点表示法。

注意 要从命令行获取属性信息，请在 MATLAB 提示符下键入 `help timeseries/tsprops`。

有关编辑 `timeseries` 对象属性的示例，请参阅“修改时序单位和插值方法”（第 3-9 页）。

时序方法

有关所有时序方法的说明，请参阅 `timeseries`。

时序集合构造函数

- “简介”（第 3-17 页）
- “时序集合构造函数语法”（第 3-17 页）
- “时序集合属性”（第 3-18 页）
- “时序集合方法”（第 3-19 页）

简介

MATLAB 对象 **tscollection** 是一个将使用相同时间向量的多个时序组合在一起的 MATLAB 变量。您包含在 **tscollection** 对象中的 **timeseries** 对象称为此集合的成员，这些成员具有若干方法，可用于轻松地分析和处理 **timeseries**。

时序集合构造函数语法

在实现专用于对 **timeseries** 对象的集合进行操作的 MATLAB 方法之前，您必须创建一个 **tscollection** 对象来存储数据。

下表总结了使用 **tscollection** 构造函数的语法。有关使用此构造函数的示例，请参阅“创建时序集合对象”（第 3-10 页）。

时序集合语法说明

语法	说明
<code>tsc = tscollection(ts)</code>	<p>创建一个包含一个或多个 <code>timeseries</code> 对象的 <code>tscollection</code> 对象 <code>tsc</code>。</p> <p><code>ts</code> 参数可以是以下项之一：</p> <ul style="list-style-type: none">• MATLAB 工作区中的单一 <code>timeseries</code> 对象• MATLAB 工作区中 <code>timeseries</code> 对象的元胞数组 <p><code>timeseries</code> 对象在 <code>tscollection</code> 中共享相同的时间向量。</p>
<code>tsc = tscollection(Time)</code>	<p>创建一个具有时间向量 <code>Time</code> 的空 <code>tscollection</code> 对象。</p> <p>当时间值是日期字符串时，必须将 <code>Time</code> 指定为日期字符串的元胞数组。</p>
<code>tsc = tscollection(Time, TimeSeries, 'Parameter', Value, ...)</code>	<p>(可选) 在 <code>Time</code> 和 <code>TimeSeries</code> 参数后输入以下参数-值对组：</p> <ul style="list-style-type: none">• <code>Name</code> (请参阅 “时序集合属性” (第 3-18 页))

时序集合属性

下表列出了 `tscollection` 对象的属性。可以将 `Name`、`Time` 和 `TimeInfo` 属性指定为 `tscollection` 构造函数中的输入参数。

时序集合属性说明

属性	说明
Name	tscollection 对象名称，以字符串形式输入。该名称可以与 MATLAB 工作区中的 tscollection 变量名称不同。
Time	<p>时间值向量。</p> <p>TimeInfo.StartDate 为空时，使用指定单位相对于 0 测量 Time 的数值。如果定义了 TimeInfo.StartDate，则时间值表示相对于 StartDate 的以指定单位显示的日期字符串。</p> <p>Time 的长度必须与每个 tscollection 成员的 Data 属性的第一个或最后一个维度匹配。</p>
TimeInfo	<p>使用以下字段存储有关 Time 的上下文信息：</p> <ul style="list-style-type: none">• Units - 时间单位，具有以下值：'weeks'、'days'、'hours'、'minutes'、'seconds'、'milliseconds'、'microseconds' 和 'nanoseconds'• Start - 开始时间• End - 结束时间（只读）• Increment - 两个后续时间值之间的间隔。当时间为非均匀采样时，增量为 NaN。• Length - 时间向量的长度（只读）• Format - 定义日期字符串显示格式的字符串。有关详细信息，请参阅 MATLAB datestr 函数参考页。• StartDate - 定义参照日期的日期字符串。有关详细信息，请参阅 MATLAB setabstime 函数参考页。• UserData - 存储任何其他的用户定义信息

时序集合方法

- “常规时序集合方法”（第 3-19 页）
- “数据和时间操作方法”（第 3-20 页）

常规时序集合方法

使用以下方法来查询和设置对象属性以及绘制数据。

属性查询方法

方法	说明
get	查询 <code>tscollection</code> 对象属性值。
isempty	对于空 <code>tscollection</code> 对象，计算结果为 <code>true</code> 。
length	返回时间向量的长度。
plot	绘制集合中的时序。
set	设置 <code>tscollection</code> 属性值。
size	返回 <code>tscollection</code> 对象的大小。

数据和时间操作方法

使用以下方法来添加或删除数据样本，以及操作 `tscollection` 对象。

操作数据和时间的方法

方法	说明
addts	将 timeseries 对象添加到 tscollection 对象。
addsampletocollection	将数据样本添加到 tscollection 对象。
delsamplefromcollection	从 tscollection 对象中删除一个或多个数据样本。
getabstime	将日期字符串时间向量从 tscollection 对象中提取到元胞数组。
getsamplusingtime	将现有 tscollection 对象中的数据样本提取到新 tscollection 对象中。
gettimeseriesnames	返回 tscollection 对象中时序名称的元胞数组。
horzcat	水平串联 tscollection 对象。将具有相同时间向量的若干 timeseries 对象合并到一个时序集合中。
removets	从 tscollection 对象中删除一个或多个 timeseries 对象。
resample	使用新时间向量选择 tscollection 对象中的数据或对其进行插值。
setabstime	将 tscollection 对象的时间向量中的时间值设置为日期字符串。
settimeseriesnames	更改 tscollection 对象中所选 timeseries 对象的名称。
vertcat	垂直串联 tscollection 对象。沿时间维度联接若干 tscollection 对象。

