



信阳师范学院  
数学与统计学院  
SCHOOL OF MATHEMATICS AND STATISTICS

# 第8章 MATLAB绘图与可视化

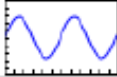
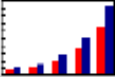

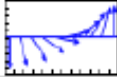


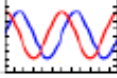
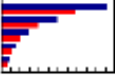

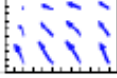


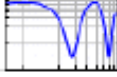
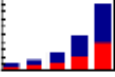



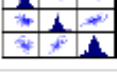

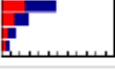
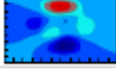

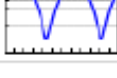

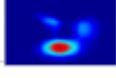


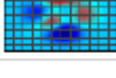

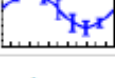
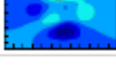
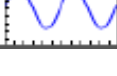



讲授人：牛言涛



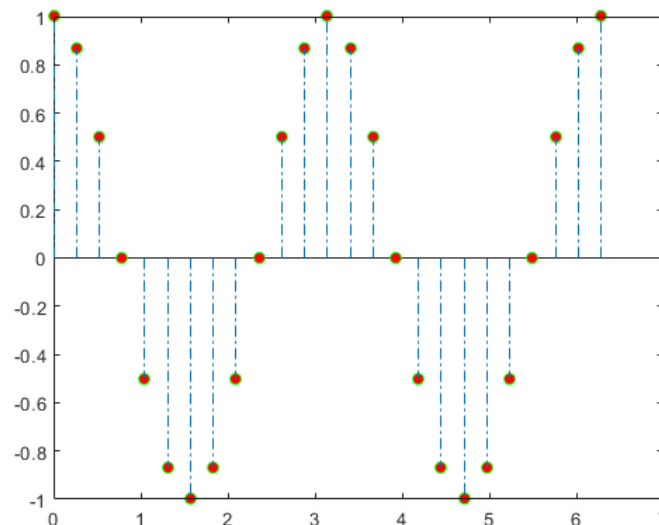
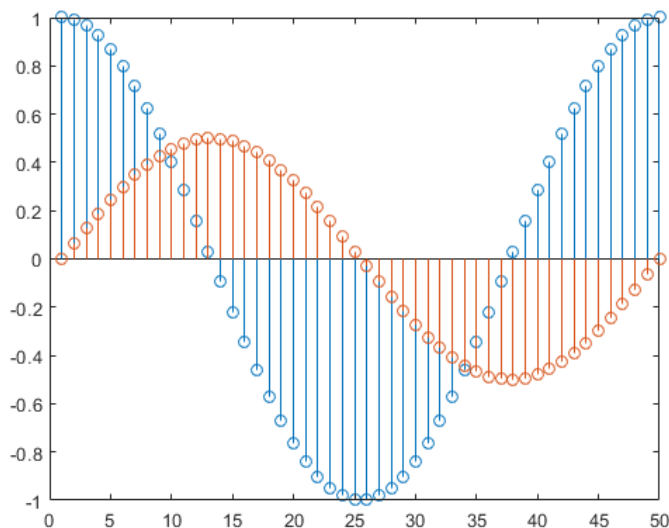
日期：2020年3月27日

# 二维绘图函数汇总

Line Graphs	Bar Graphs	Area Graphs	Direction Graphs	Radial Graphs	Scatter Graphs
<a href="#">plot</a> 	<a href="#">bar</a> (grouped) 	<a href="#">area</a> 	<a href="#">feather</a> 	<a href="#">polar</a> 	<a href="#">scatter</a> 
<a href="#">plotyy</a> 	<a href="#">barh</a> (grouped) 	<a href="#">pie</a> 	<a href="#">quiver</a> 	<a href="#">rose</a> 	<a href="#">spy</a> 
<a href="#">loglog</a> 	<a href="#">bar</a> (stacked) 	<a href="#">fill</a> 	<a href="#">comet</a> 	<a href="#">compass</a> 	<a href="#">plotmatrix</a> 
<a href="#">semilogx</a> 	<a href="#">barh</a> (stacked) 	<a href="#">contourf</a> 		<a href="#">ezpolar</a> 	
<a href="#">semilogy</a> 	<a href="#">hist</a> 	<a href="#">image</a> 			
<a href="#">stairs</a> 	<a href="#">pareto</a> 	<a href="#">pcolor</a> 			
<a href="#">contour</a> 	<a href="#">errorbar</a> 	<a href="#">ezcontourf</a> 			
<a href="#">ezplot</a> 	<a href="#">stem</a> 				

# 1. stem绘制离散序列数据

- `stem(Y)` 将数据序列  $Y$  绘制为从沿  $x$  轴的基线延伸的针状图。各个数据值由终止每个针状图的圆指示。
- `stem(X,Y, 'filled', LineSpec)` 在  $X$  指定的值的位置绘制数据序列  $Y$ 。
- `stem(___,Name,Value)` 使用一个或多个  $Name,Value$  对组参数修改针状图。
- `stem(ax,___)` 将图形绘制到  $ax$  指定的坐标区中，而不是当前坐标区 ( $gca$ ) 中。
- `h = stem(___)` 在  $h$  中返回由 Stem 对象构成的向量。使用  $h$  可在创建针状图后对其进行修改。



# 1. stem绘制离散序列数据



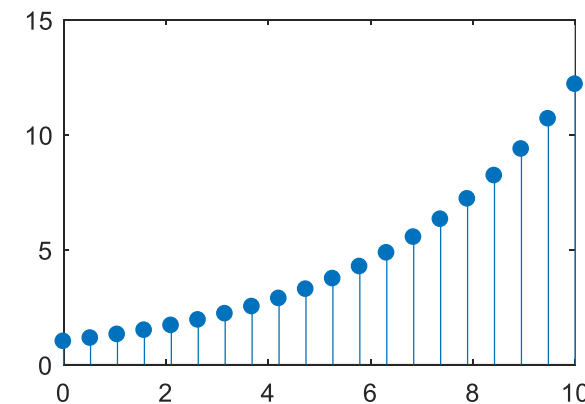
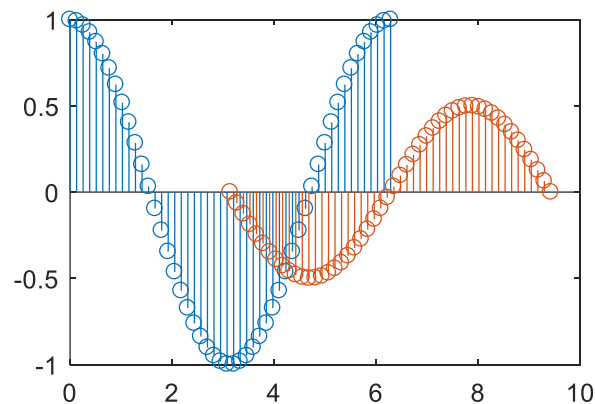
```
subplot(2,2,1)
```

```
x1 = linspace(0,2*pi,50)';
```

```
x2 = linspace(pi,3*pi,50)';
```

```
X = [x1, x2]; Y = [cos(x1), 0.5*sin(x2)];
```

```
stem(X,Y)
```



```
subplot(2,2,2)
```

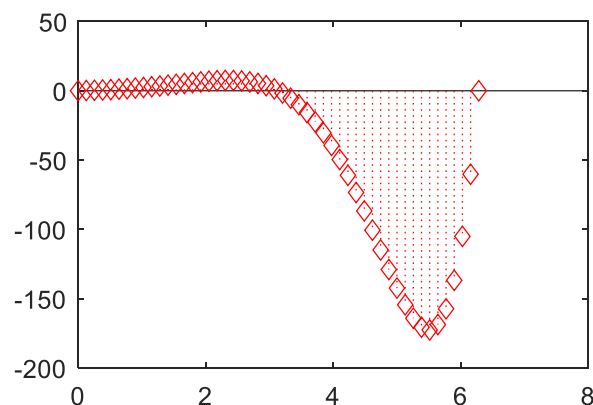
```
X = linspace(0,10,20)'; Y = (exp(0.25*X));
```

```
stem(X,Y,'filled')
```

```
subplot(2,2,3)
```

```
X = linspace(0,2*pi,50)'; Y = (exp(X).*sin(X));
```

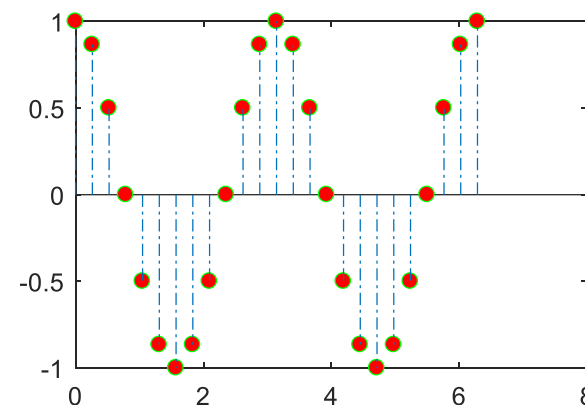
```
stem(X,Y,':diamondr')
```



```
subplot(2,2,4)
```

```
X = linspace(0,2*pi,25)'; Y = (cos(2*X));
```

```
stem(X,Y,'LineStyle','-','MarkerFaceColor','red','MarkerEdgeColor','green')
```



## 2. 条形图bar和barh

bar创建一个水平条形图，barh创建一个水平条形图，其语法结构同bar，这里以bar为例说明其函数结构。

`bar(y)` 创建一个条形图，`y` 中的每个元素对应一个条形。如果 `y` 是  $m \times n$  矩阵，则 `bar` 创建每组包含 `n` 个条形的 `m` 个组。

`bar(x,y)` 在 `x` 指定的位置绘制条形。

`bar( __,width)` 设置条形的相对宽度以控制组中各个条形的间隔。将 `width` 指定为标量值。可以将此选项与前面语法中的任何输入参数组合一起使用。

`bar( __,style)` 指定条形组的样式。例如，使用 `'stacked'` 将每个组显示为一个多种颜色的条形。

`bar( __,color)` 设置所有条形的颜色。例如，使用 `'r'` 表示红色条形。

`bar( __,Name,Value)` 使用一个或多个名称-值对组参数指定条形图的属性。仅使用默认 `'grouped'` 或 `'stacked'` 样式的条形图支持设置条形属性。在所有其他输入参数之后指定名称-值对组参数。有关属性列表，请参阅 [Bar 属性](#)。

`bar(ax, __)` 将图形绘制到 `ax` 指定的坐标区中，而不是当前坐标区 (`gca`) 中。选项 `ax` 可以位于前面的语法中的任何输入参数组合之前。

`b = bar( __)` 返回一个或多个 `Bar` 对象。如果 `y` 是向量，则 `bar` 将创建一个 `Bar` 对象。如果 `y` 是矩阵，则 `bar` 为每个序列返回一个 `Bar` 对象。显示条形图后，使用 `b` 设置条形的属性。

**条形宽度**，指定为可用于每个条形的总空间的一部分。默认值 0.8 表示条形宽度是从上一条形到下一条形之间的空间的 80%，两端各占该空间的 10%。如果宽度为 1，则组中的条形紧挨在一起。

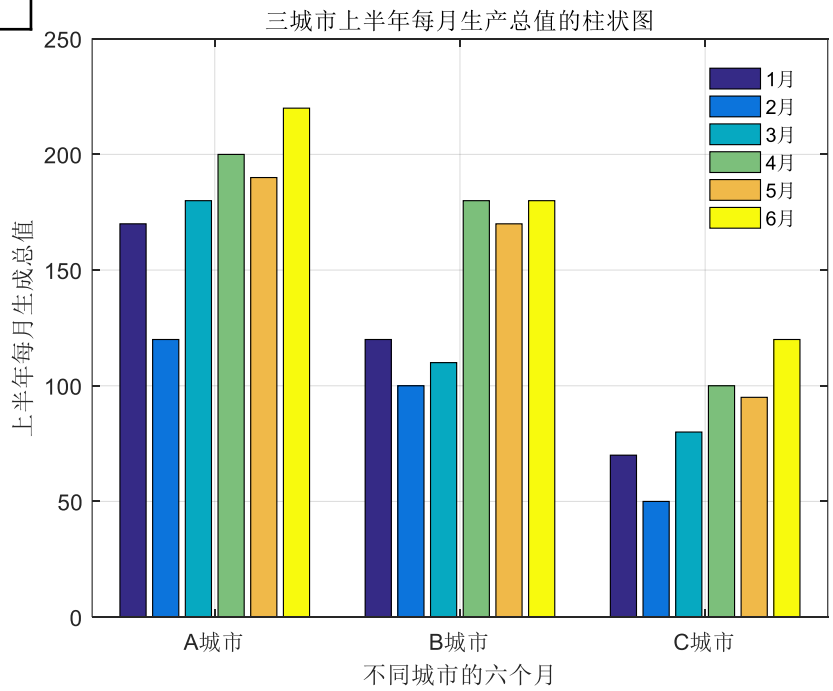
示例： `bar([1 2 3],0.5)` 创建使用 50% 可用空间的条形。

## 2. 条形图bar和barh

例1： A,B,C 三个城市上半年每个月的国民生产总值见表 ， 试绘制三城市每月生产总值的条形图。

城市	1月	2月	3月	4月	5月	6月
A	170	120	180	200	190	220
B	120	100	110	180	170	180
C	70	50	80	100	95	120

```
>> XX = [170 120 180 200 190 220;120 100 110 180 170 180;70 50 80 100  
95 120];  
>> bar(XX)  
>> set(gca,'xticklabel',{'A城市','B城市','C城市'})  
>> legend('1月','2月','3月','4月','5月','6月')  
>> xlabel('不同城市的六个月'); ylabel('上半年每月生成总值');  
>> legend('boxoff')  
>> title('三城市上半年每月生产总值的柱状图')
```



## 2. 条形图bar和barh

```
XX = [170 120 180 200 190 220;120 100 110 180 170 180;70 50 80 100 95 120];
```

```
subplot(1,2,1)
```

```
bar(XX,'stack')
```

```
set(gca,'xticklabel',{'A城市','B城市','C城市'})
```

```
legend('1月','2月','3月','4月','5月','6月')
```

```
xlabel('不同城市的六个月');
```

```
ylabel('上半年每月生成总值');
```

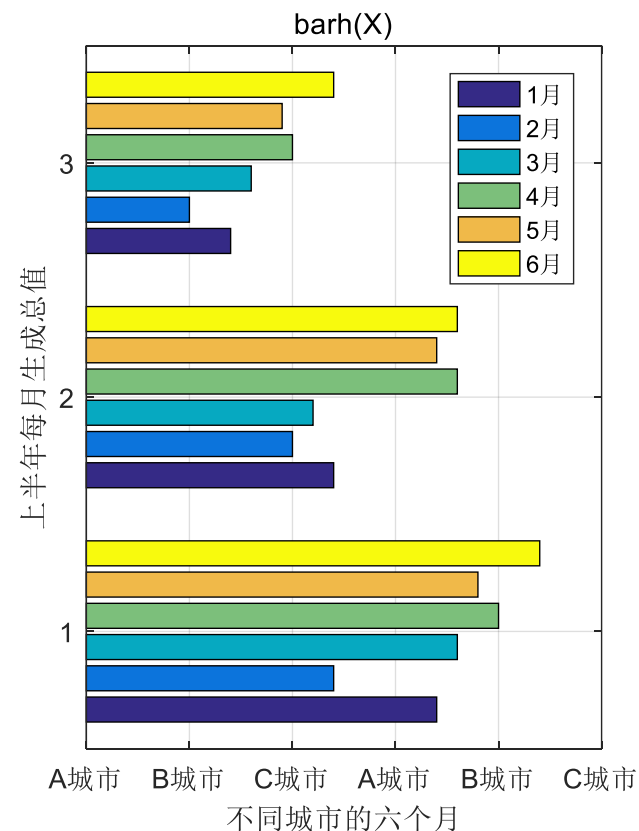
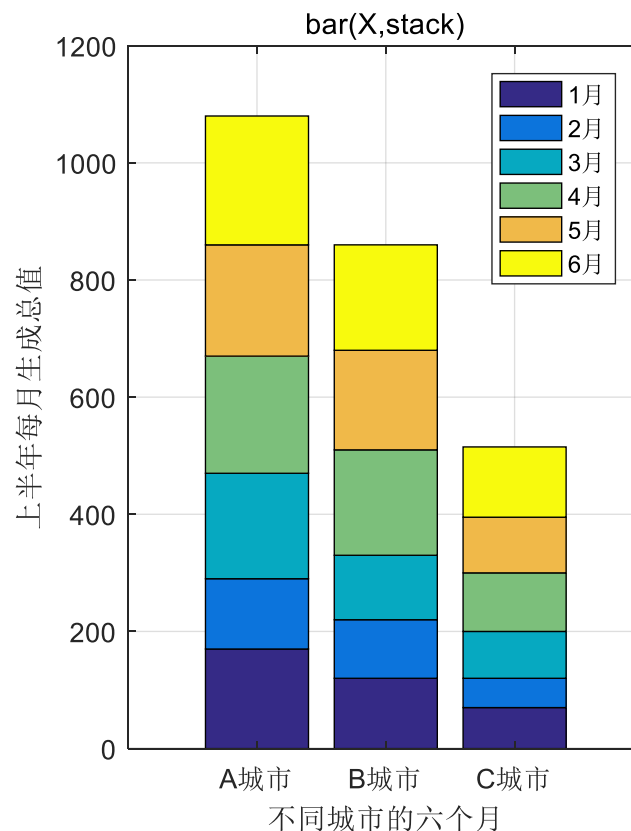
```
grid on
```

```
title('bar(X,stack)')
```

```
subplot(1,2,2)
```

```
barh(XX) %创建水平条形图
```

```
title('barh(X)')
```



### 3. 直方图histogram

直方图可以直观地显示数据的分布情况。注：不在推荐使用hist函数！

`histogram(X)` 基于 X 创建直方图。`histogram` 函数使用自动 bin 划分算法，然后返回均匀宽度的 bin，这些 bin 可涵盖 x 中的元素范围并显示分布的基本形状。`histogram` 将 bin 显示为矩形，这样每个矩形的高度就表示 bin 中的元素数量。

`histogram(X,nbins)` 使用标量 `nbins` 指定的 bin 数量。

`histogram(X,edges)` 将 X 划分到由向量 `edges` 来指定 bin 边界的 bin 内。每个 bin 都包含左边界，但不包含右边界，除了同时包含两个边界的最后一个 bin 外。

`histogram('BinEdges',edges,'BinCounts',counts)` 手动指定 bin 边界和关联的 bin 计数。`histogram` 绘制指定的 bin 计数，而不执行任何数据的 bin 划分。

`histogram(C)` (其中 C 为分类数组) 通过为 C 中的每个类别绘制一个条形来绘制直方图。

`histogram(C,Categories)` 仅绘制 `Categories` 指定的类别的子集。

`histogram('Categories',Categories,'BinCounts',counts)` 手动指定类别和关联的 bin 计数。`histogram` 绘制指定的 bin 计数，而不执行任何数据的 bin 划分。

`histogram( ___,Name,Value)` 使用前面的任何语法指定具有一个或多个 Name,Value 对组参数的其他选项。例如，可以指定 'BinWidth' 和一个标量以调整 bin 的宽度，或指定 'Normalization' 和一个有效选项 ('count'、'probability'、'countdensity'、'pdf'、'cumcount' 或 'cdf') 以使用不同类型的归一化。有关属性列表，请参阅 [Histogram 属性](#)。

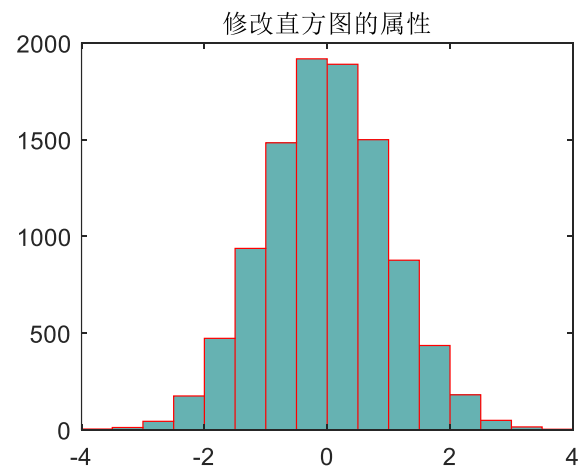
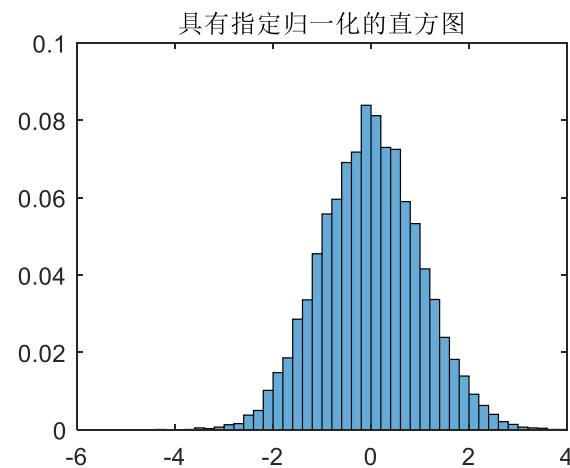
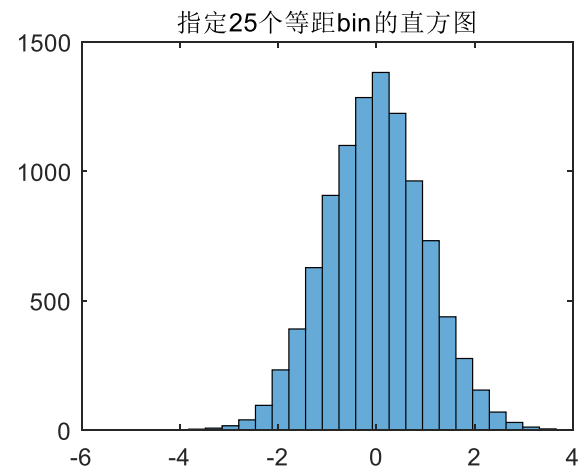
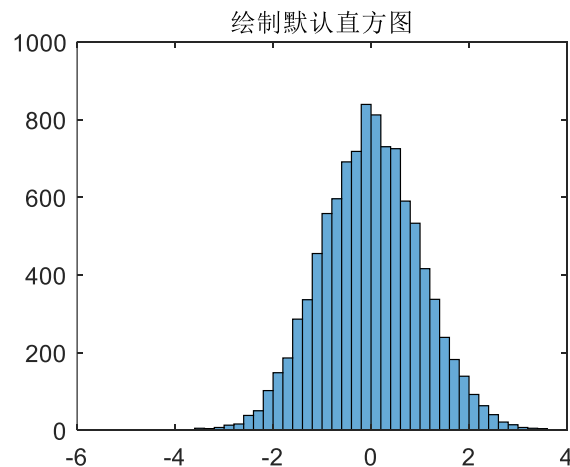
`histogram(ax, ___)` 将图形绘制到 ax 指定的坐标区中，而不是当前坐标区 (gca) 中。选项 ax 可以位于前面的语法中的任何输入参数组合之前。

`h = histogram( ___)` 返回 Histogram 对象。使用此语法可检查并调整直方图的属性。有关属性列表，请参阅 [Histogram 属性](#)。



### 3. 直方图histogram

```
x = randn(10000,1);  
subplot(2,2,1); h = histogram(x);  
subplot(2,2,2)  
nbins = 25; %指定分类为25个等距bin  
h = histogram(x,nbins);  
subplot(2,2,3)  
%使用 'probability' 归一化创建直方图。  
h = histogram(x,'Normalization','probability');  
S = sum(h.Values); % S = 1  
subplot(2,2,4)  
h = histogram(x);  
h.NumBins = 30; %准确指定要使用的 bin 数量。  
h.BinEdges = [-4:0.5:4]; %通过向量指定 bin 边界。  
h.FaceColor = [0 0.5 0.5];  
h.EdgeColor = 'r';
```



### 3. 直方图histogram

```
subplot(2,1,1)
```

%生成均值为5、标准差为2的正态分布随机数。在Normalization设为'pdf'的情况下绘制直方图可生成概率密度函数的估计值。

```
x = 2*randn(5000,1) + 5;
```

```
histogram(x,'Normalization','pdf')
```

```
subplot(2,1,2)
```

%对于均值为 5、标准差为 2 的正态分布，叠加一个概率密度函数图。

```
histogram(x,'Normalization','pdf')
```

```
hold on
```

```
y = -5:0.1:15;
```

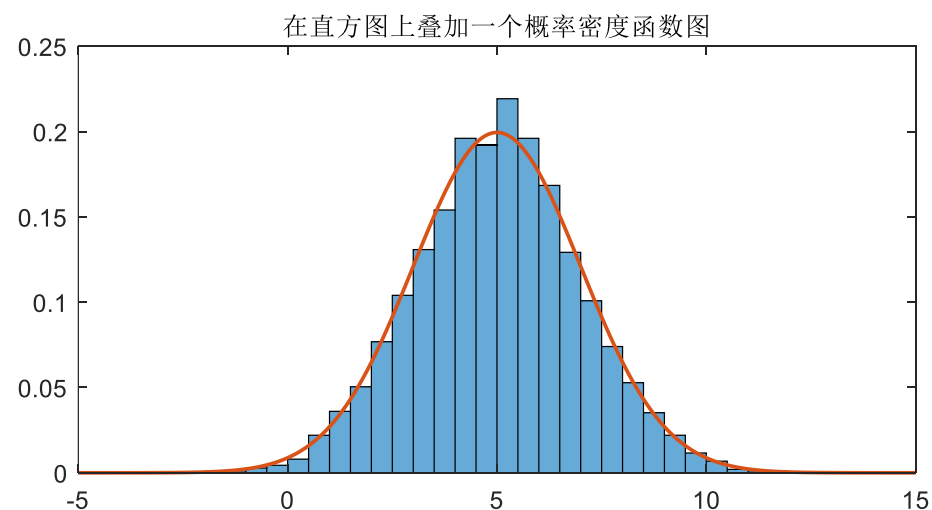
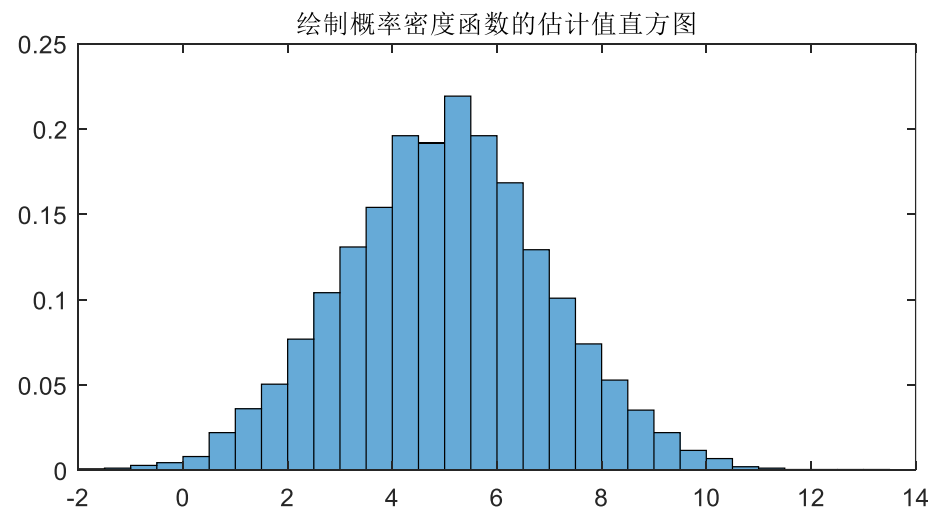
```
mu = 5;
```

```
sigma = 2;
```

```
%正态分布密度函数
```

```
f = exp(-(y-mu).^2./(2*sigma^2))./(sigma*sqrt(2*pi));
```

```
plot(y,f,'LineWidth',1.5)
```



### 3. 直方图histogram

```
subplot(2,1,1)
```

```
x = randn(2000,1);
```

```
y = 1 + randn(5000,1);
```

```
h1 = histogram(x); hold on; h2 = histogram(y);
```

```
title('绘制多个直方图')
```

```
subplot(2,1,2)
```

%由于直方图的示例大小和 bin 宽度不同，很难将它们进行比较。

对这些直方图进行归一化，这样所有的条形高度相加的结果为 1 并

使用统一的 bin 宽度。

```
h1 = histogram(x); hold on; h2 = histogram(y);
```

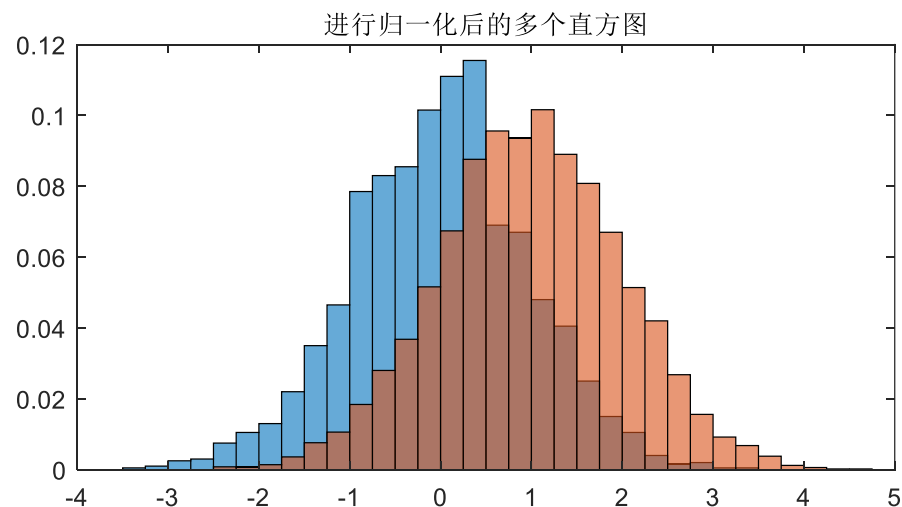
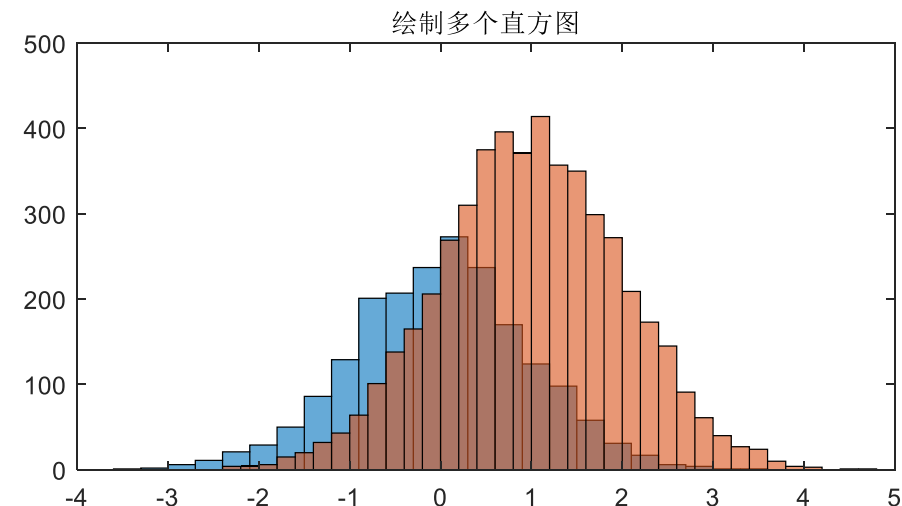
```
h1.Normalization = 'probability';
```

```
h1.BinWidth = 0.25;
```

```
h2.Normalization = 'probability';
```

```
h2.BinWidth = 0.25;
```

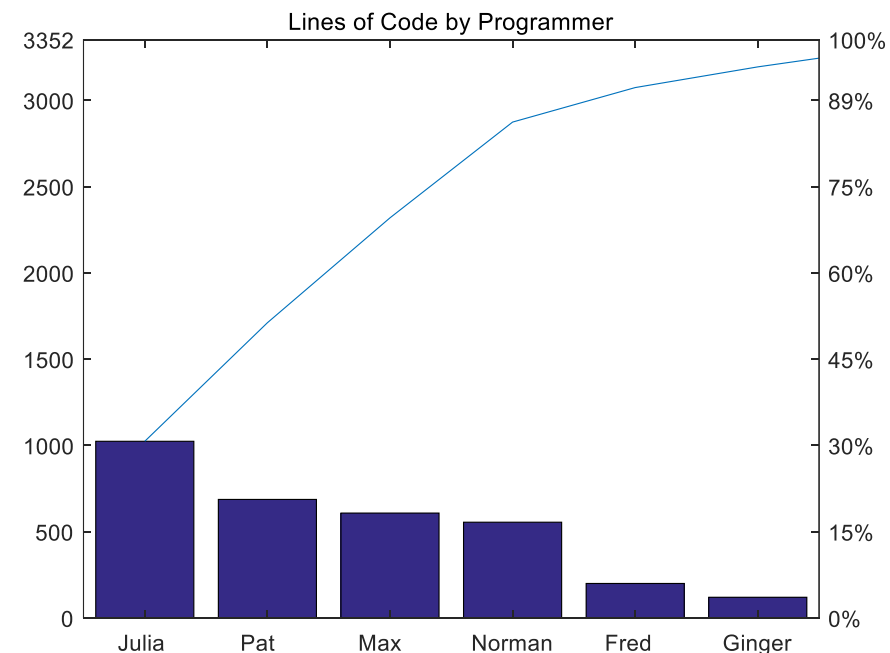
```
title('进行归一化后的多个直方图')
```



## 4. 帕累托图pareto

- 帕累托图将向量  $Y$  中的值显示为降序排列的条形图。 $Y$  中的值必须是非负的且不能包含 NaN。  
仅显示前 95% 的累积分布。
- `pareto(Y, names)` 用  $Y$  中的元素索引标识每个条形，并绘制线条以显示  $Y$  的累积和。用矩阵或元胞数组 `names` 中的关联文本标识每个条形。
- `[H, ax] = pareto(ax, ...)` 在坐标区 `ax` 而不是当前坐标区 `gca` 中绘图。`H` 返回创建的原始 `Line` 和 `Bar` 对象。`ax` 返回创建的两个坐标区对象。

```
codelines = [200 120 555 608 1024 101 57 687];  
coders =  
{'Fred', 'Ginger', 'Norman', 'Max', 'Julia', 'Wally', 'Heidi', 'Pat'};  
pareto(codelines, coders)  
title('Lines of Code by Programmer')
```

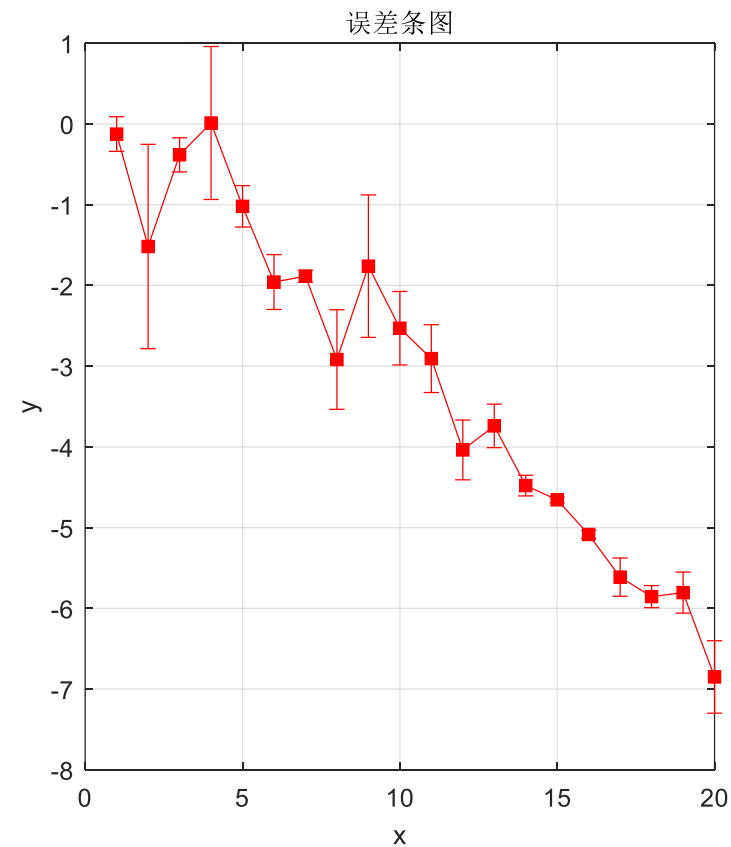
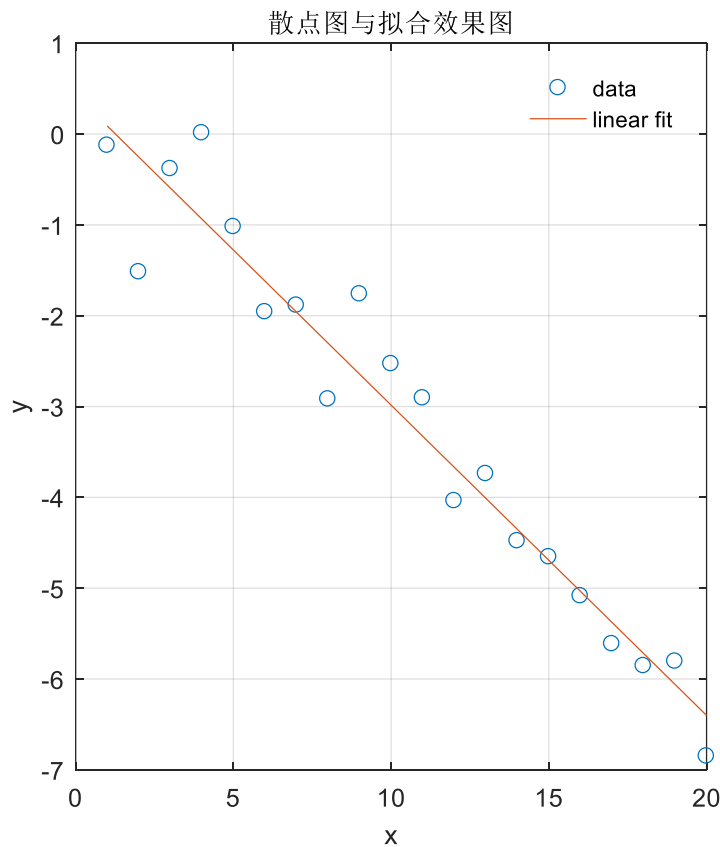


## 5. 误差条图errorbar

- `errorbar(x,y,err)` 绘制  $y$  对  $x$  的图，并在每个数据点处绘制一个垂直误差条。`err` 中的值确定数据点上方和下方的每个误差条的长度，因此，总误差条长度是 `err` 值的两倍。
- `errorbar(x,y,neg,pos,ornt)` `neg` 确定数据点下方的长度，`pos` 确定数据点上方的长度。`ornt` 设置误差条的方向：水平误差条 'horizontal'，水平和垂直误差条 'both'。默认值为 'vertical'。
- `errorbar(x,y,yneg,ypos,xneg,xpos)` 绘制  $y$  对  $x$  的图，并同时绘制水平和垂直误差条。`yneg` 和 `ypos` 输入分别设置垂直误差条下部和上部的长度。`xneg` 和 `xpos` 输入分别设置水平误差条左侧和右侧的长度。
- `errorbar(____,linespec)` 设置线型、标记符号和颜色。线型仅影响线，而不影响误差条。
- `errorbar(____,Name,Value)` 使用一个或多个名称-值对组参数修改线和误差条的外观。例如，'CapSize',10 将每个误差条末端的端盖长度设置为 10 磅。
- `errorbar(ax,____)` 在由 `ax` 指定的坐标区（而不是当前坐标区）中创建绘图。指定坐标区作为第一个输入参数。
- 当  $y$  为向量时，`e = errorbar(____)` 返回一个 `ErrorBar` 对象。如果  $y$  是矩阵，它将为  $y$  中的每一列返回一个 `ErrorBar` 对象。可在创建特定的 `ErrorBar` 对象后使用 `e` 修改其属性。有关属性列表，请参阅 `ErrorBar` 属性。

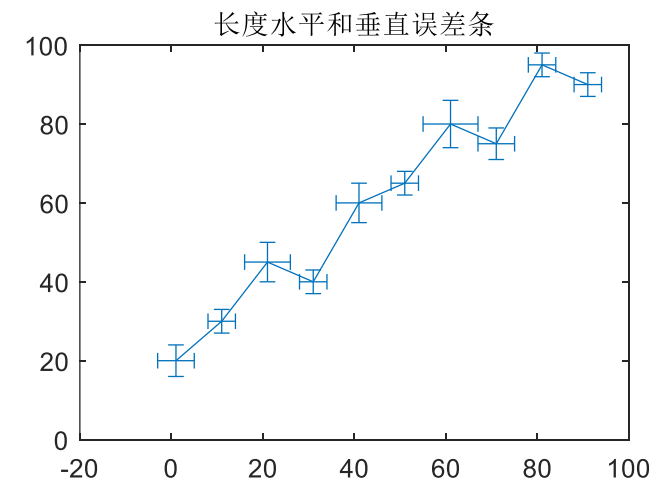
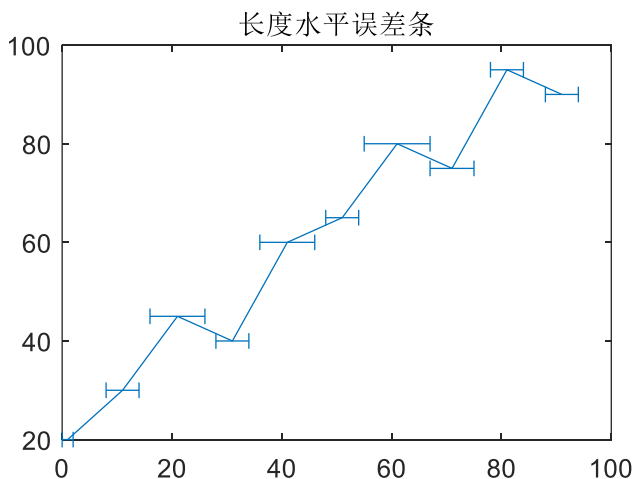
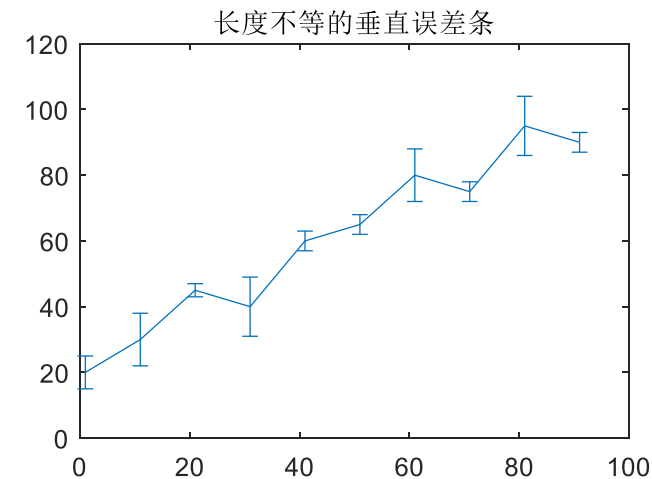
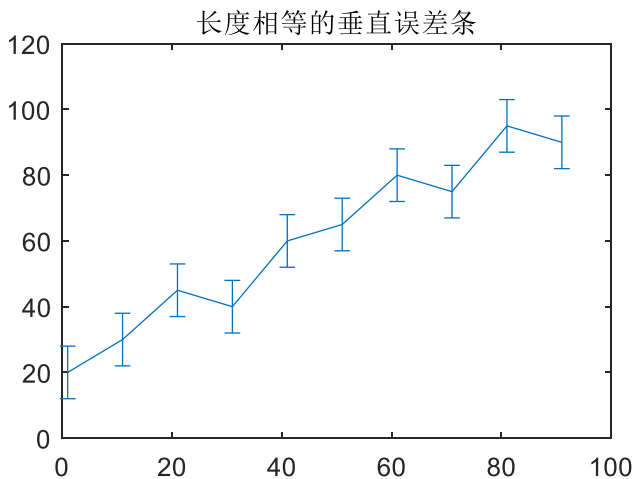
## 5. 误差条图errorbar

```
x = 1:20;  
y = -0.3*x + 0.5*randn(1,20);  
p = polyfit(x,y,1);  
fvar = polyval(p,x);  
subplot(1,2,1)  
plot(x,y,'o',x,fvar,'-')  
title('散点图与拟合效果图')  
legend('data','linear fit')  
legend('boxoff')  
grid on  
subplot(1,2,2)  
err = abs(y-fvar);  
errorbar(x,y,err,'-rs','MarkerFaceColor','r')  
grid on  
title('误差条图')
```



## 5. 误差条图errorbar

```
>> x = 1:10:100;  
>> y = [20 30 45 40 60 65 80 75 95 90];  
>> subplot(2,2,1)  
>> err = 8*ones(size(y));  
>> errorbar(x,y,err)  
>> subplot(2,2,2)  
>> err = [5 8 2 9 3 3 8 3 9 3];  
>> errorbar(x,y,err)  
>> subplot(2,2,3)  
>> err = [1 3 5 3 5 3 6 4 3 3];  
>> errorbar(x,y,err,'horizontal')  
>> subplot(2,2,4)  
>> err = [4 3 5 3 5 3 6 4 3 3];  
>> errorbar(x,y,err,'both')
```



## 5. 误差条图errorbar

```
subplot(2,2,1)
```

```
err = [4 3 5 3 5 3 6 4 3 3];
```

```
errorbar(x,y,err,'both','o')
```

```
subplot(2,2,2)
```

```
yneg = [1 3 5 3 5 3 6 4 3 3];
```

```
ypos = [2 5 3 5 2 5 2 2 5 5];
```

```
xneg = [1 3 5 3 5 3 6 4 3 3];
```

```
xpos = [2 5 3 5 2 5 2 2 5 5];
```

```
errorbar(x,y,yneg,ypos,xneg,xpos,'o')
```

```
subplot(2,2,3)
```

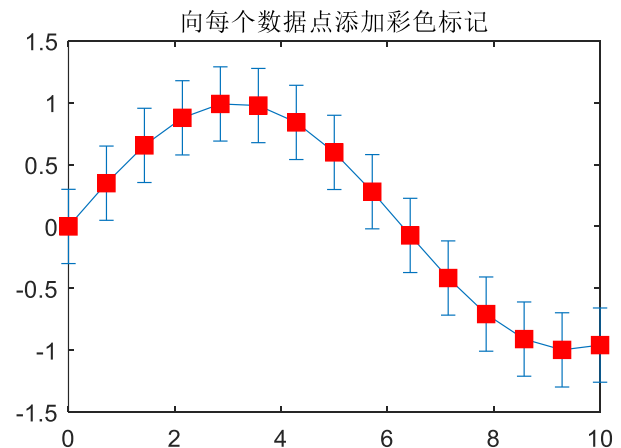
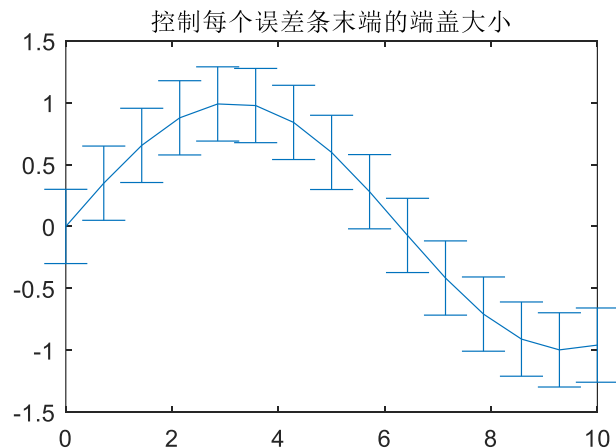
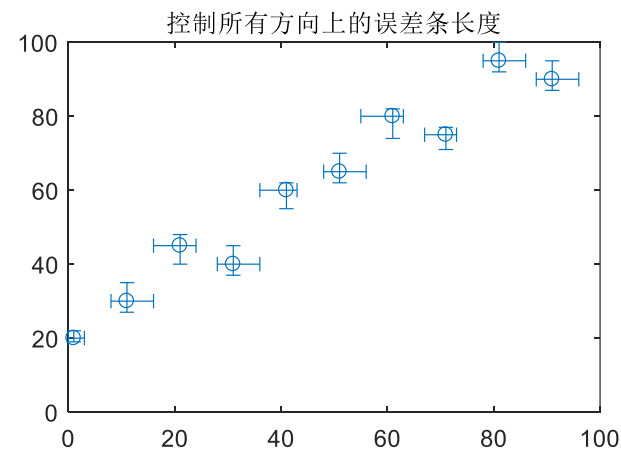
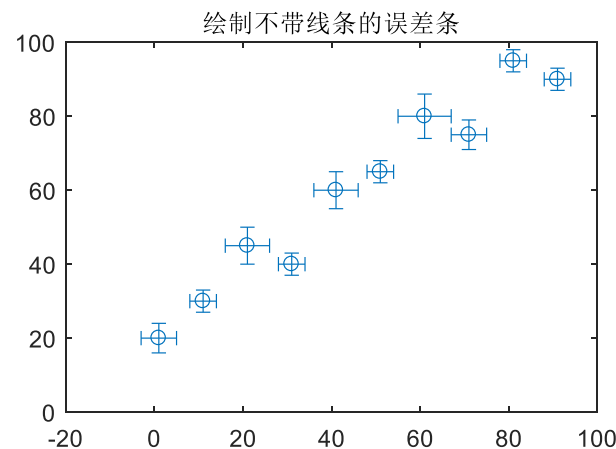
```
x = linspace(0,10,15); y = sin(x/2);
```

```
err = 0.3*ones(size(y));
```

```
errorbar(x,y,err,'CapSize',18)
```

```
subplot(2,2,4)
```

```
errorbar(x,y,err,'-s','MarkerSize',10,'MarkerEdgeColor','red','MarkerFaceColor','red')
```







---

# 感谢聆听

---