

《生物计算机程序设计语言》

“Biology Computer Programming Language”

杨建华、郑凌伶

生命科学学院

2019年10月8日

授课教师

杨建华教授



电话: 13512733538

邮箱: yangjh7@mail.sysu.edu.cn

办公室: 曾宪梓北院317房

研究方向: 非编码RNA基因和RNA修饰及其互作蛋白的结构、功能和作用机制、生物信息学

软件和平台: starBase、snoSeeker、StarScan、deepBase、RMBase、circScan、tRF2Cancer和ChIPBase等。

论文: Nature、Nature Cell Biology、European Urology、Cell Research、Nucleic Acids Res.、Cell Reports, EMBO Reports, 超过15篇IF>10.0的研究论文, 8篇ESI高引用论文,

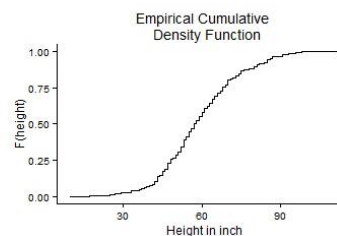
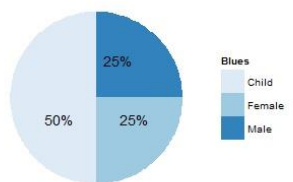
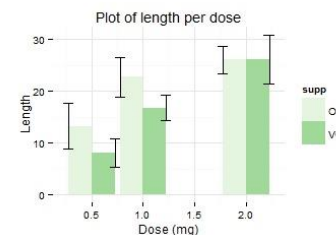
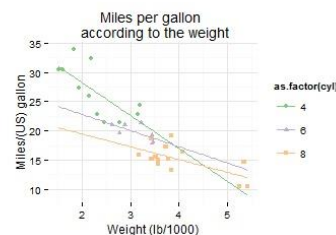
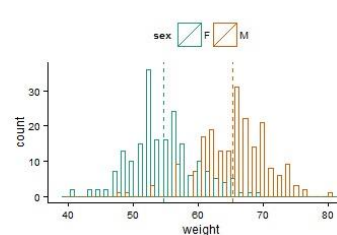
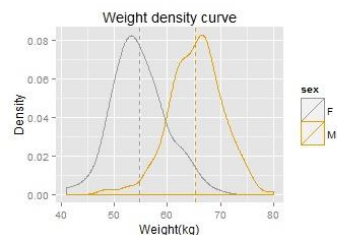
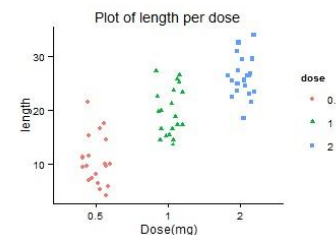
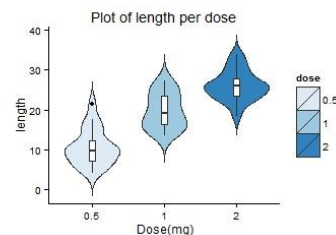
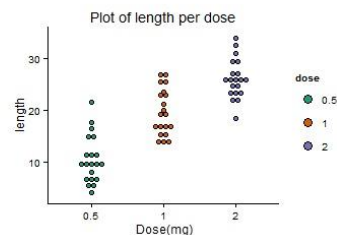
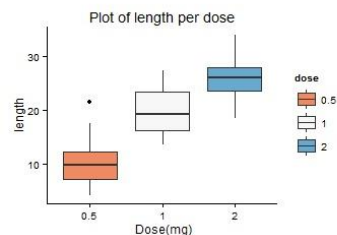
1篇论文入选“2014年中国百篇最具影响国际学术论文”,

5篇被Nature Reviews Genetics、Nature Chemical Biology、Nature Cell Biology, European Urology、Cell Research、Cancer Research、EMBO Reprots等亮点文章。

国际期刊Non-coding RNA杂志的编委, 是*Mol. Biol. Evol.*, *Nucleic Acids Res.*, *Bioinformatics*等杂志审稿人。

教学: 指导中大iGEM软件队获得6次金牌, 4年冠军。

Chapter 6: R-Basic Plots

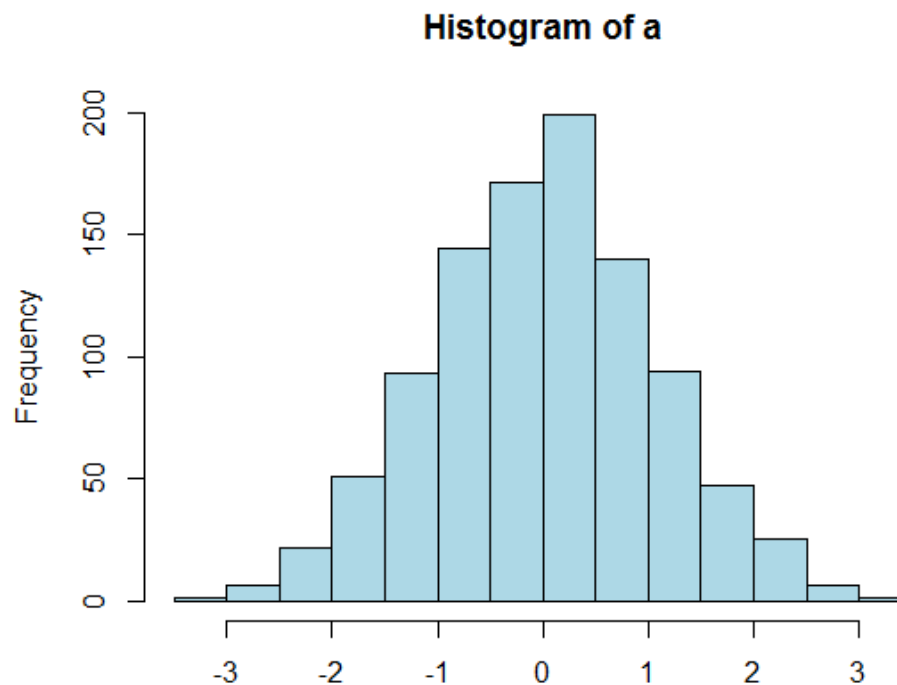


直方图

基本样式: `hist(x, breaks, freq, col, border, ...)`, 其中x为一维数值, breaks为组距 (**bins**) (“Sturges” (default), "Scott" and "FD") 。

```
a=rnorm(1000)
hist(a, col="lightblue")
```

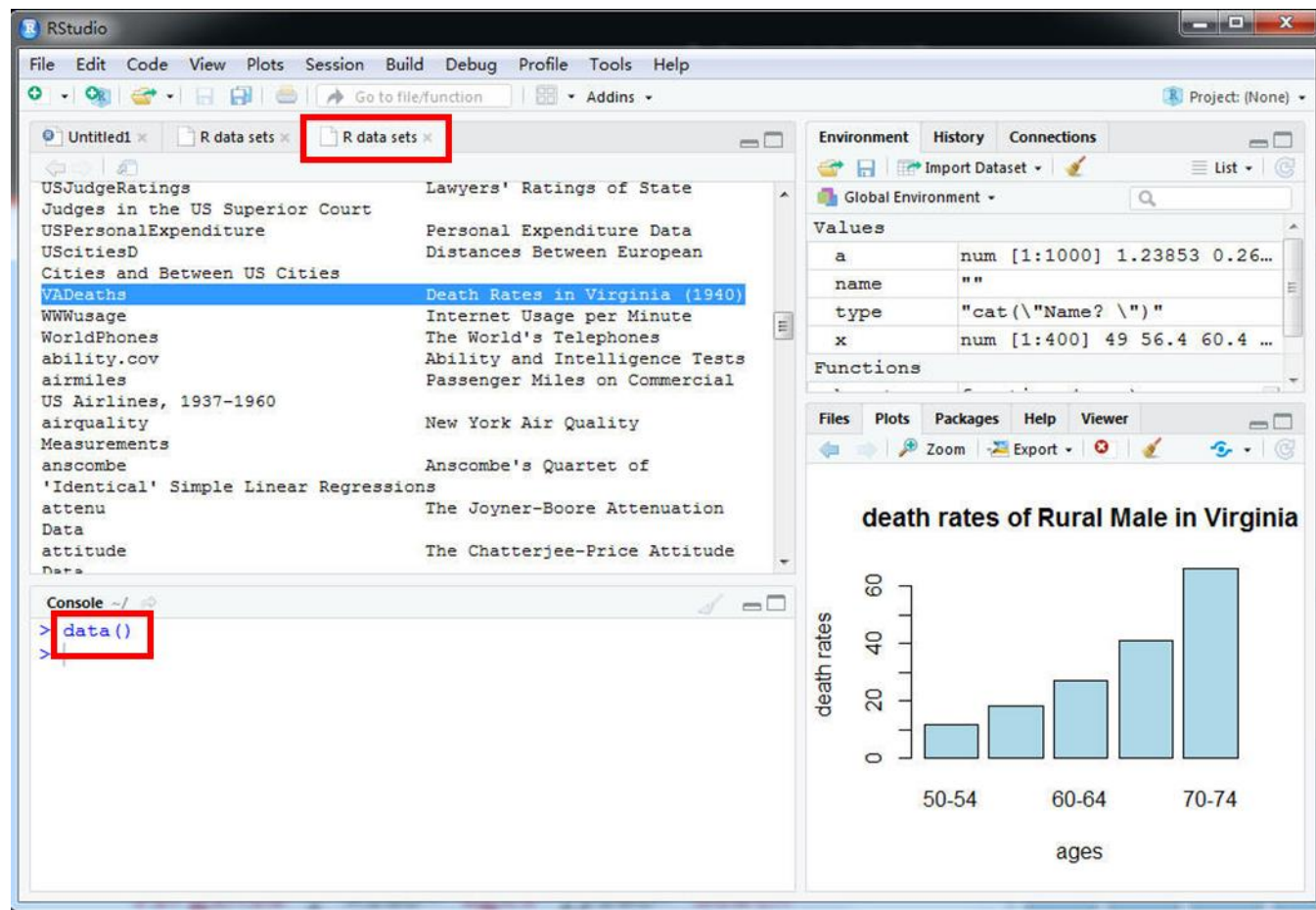
`rnorm(n, mean = 0, sd = 1)`
随机产生具有mean和sd值的正态分布的n个数值
注意: 可以设定随机seed, `set.seed(1234)`



如何调整直方图的数据分布?

直方图

查看R的内嵌的数据(R Built-in Data Sets)用: `data()`命令



1. Loading

`data("iris")`

2. Print

`head(iris)`

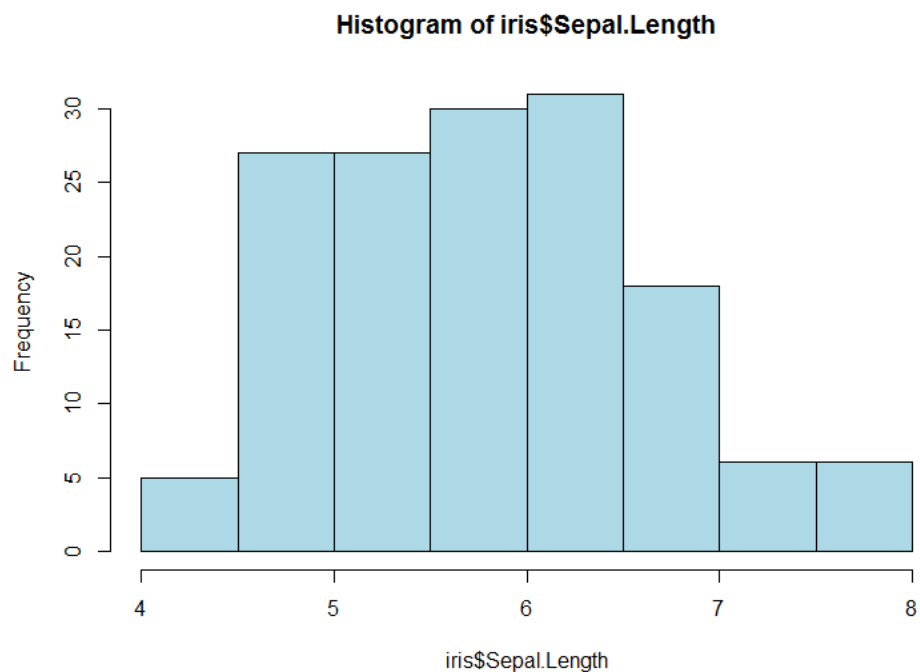
本周将利用R中自带的iris的数据集进行分析, iris中文名是鸢尾花, 有四个属性, 分别是 Sepal.Length (花萼长度), Sepal.Width (花萼宽度), Petal.Length (花瓣长度), Petal.Width (花瓣宽度), 以及一个亚种标签Species



直方图

查看Sepal.Length (花萼长度)的分布

```
hist(iris$Sepal.Length, col="lightblue")
```



本周将利用R中自带的iris的数据集进行分析，iris中文名是鸢尾花，有四个属性，分别是 Sepal.Length（花萼长度），Sepal.Width（花萼宽度），Petal.Length（花瓣长度），Petal.Width（花瓣宽度），以及一个亚种标签Species

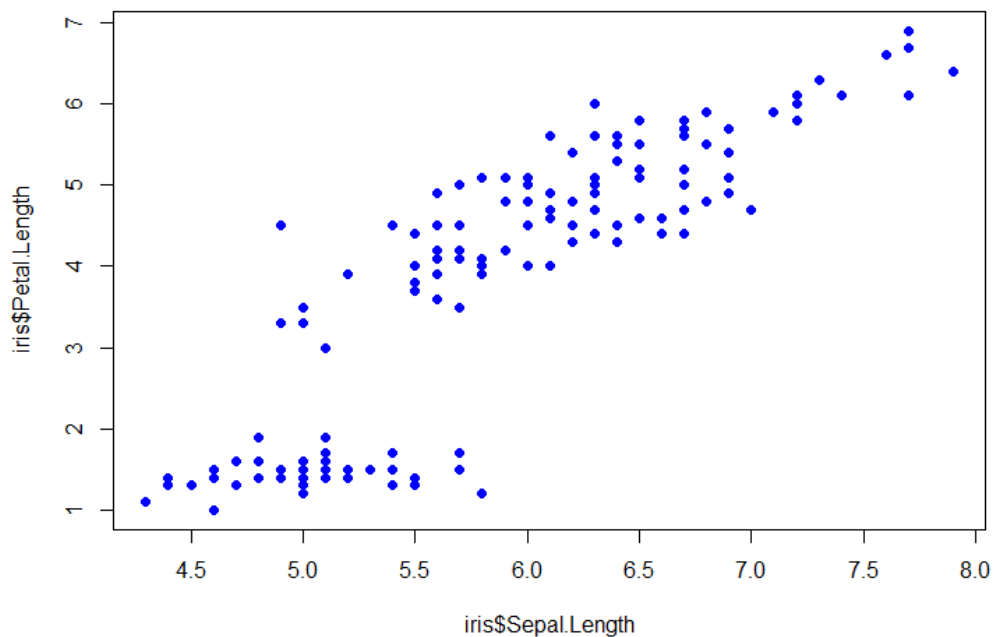


散点图

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

其中： **x**, **y**一维数值。

```
plot(iris$Sepal.Length, iris$Petal.Length, pch=19, col="blue")
```



本周将利用R中自带的iris的数据集进行分析，iris中文名是鸢尾花，有四个属性，分别是 Sepal.Length（花萼长度），Sepal.Width（花萼宽度），Petal.Length（花瓣长度），Petal.Width（花瓣宽度），以及一个亚种标签Species



- 1.对鸢尾花的三个亚种（setosa、versicolor、virginica）分别统计它们的花萼长度（Sepal.Length）、花萼宽度（Sepal.Width）、花瓣长度（Petal.Length）、花瓣宽度（Petal.Width）的最大值、最小值、均值、中位数等特征，并绘制箱线图进行展示
2. 请问鸢尾花的四种属性花萼长度（Sepal.Length）、花萼宽度（Sepal.Width）、花瓣长度（Petal.Length）、花瓣宽度（Petal.Width）之间是否存在相关性？如何展示？##
3. 作为一个生物学家，你能从这些数据中得到什么结论吗？

散点图

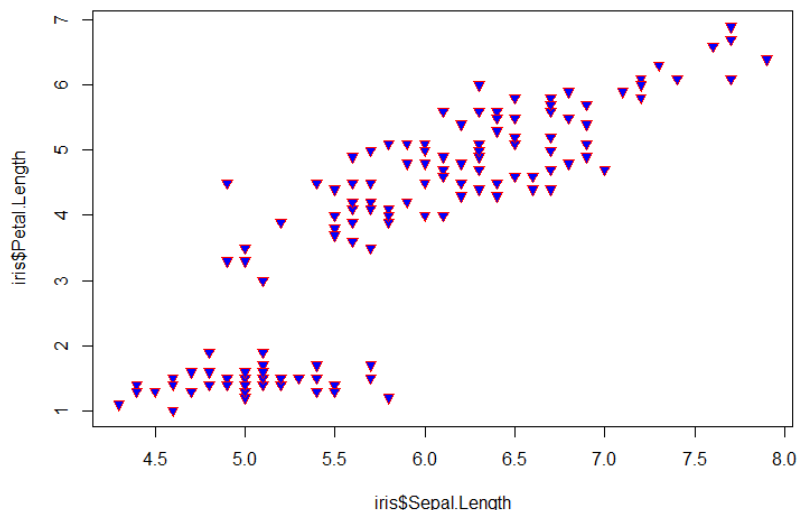
`plot(x, y, ...)`

其中： `x`, `y`一维数值。

`plot`(iris\$Sepal.Length, iris\$Petal.Length, `pch`=19, `col`="blue")

当`pch`取0~14时，其点为空心点，可以用`col`(颜色)参数设置其边框的颜色；
当`pch`取15~20时，其点是实心点，可以用`col`参数设置其填充的颜色；
当`pch`取21~25时，其点也是实心点，既可以用`col`参数设置边框的颜色，也可以用`bg`参数设置其内部的填充颜色。

`plot`(iris\$Sepal.Length, iris\$Petal.Length, `pch`=25, `col`="red", `bg`="blue")



R语言里的点样式`pch`

Change plotting symbols

The following points symbols can be used in R :

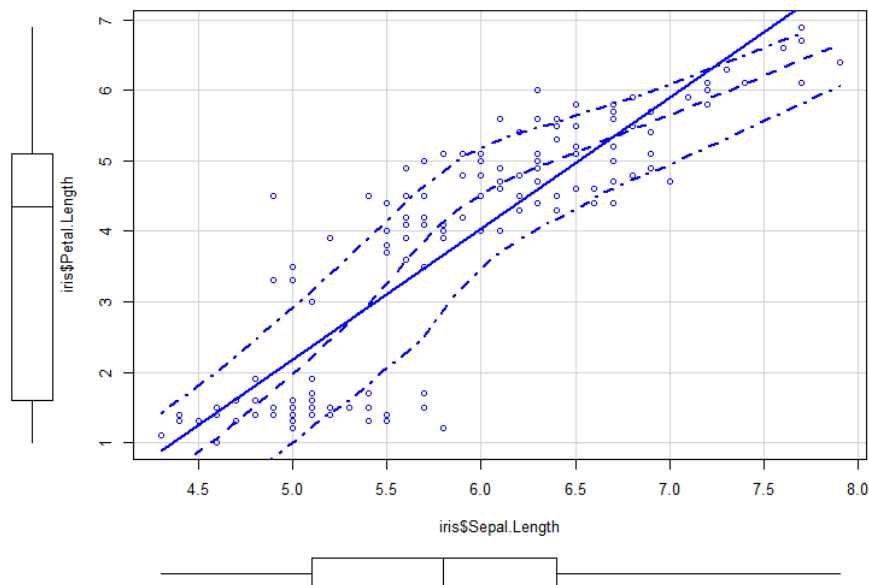
0 □	1 ○	2 △	3 +	4 ×	
5 ◇	6 ▽	7 ⊠	8 ✱	9 ⊕	
10 ⊕	11 ⊗	12 ⊞	13 ⊠	14 ⊞	
15 ■	16 ●	17 ▲	18 ◆	19 ●	
20 ●	21 ●	22 ■	23 ◆	24 ▲	25 ▼

Point symbols can be changed using the argument `pch`.

散点图

高级版的散点图`scatterplot(x, y,...)`
其中：`x`，`y`一维数值。

1. `install.packages("car")`
2. `library("car")`
3. `scatterplot(iris$Sepal.Length, iris$Petal.Length)`



本周将利用R中自带的iris的数据集进行分析，iris中文名是鸢尾花，有四个属性，分别是Sepal.Length（花萼长度），Sepal.Width（花萼宽度），Petal.Length（花瓣长度），Petal.Width（花瓣宽度），以及一个亚种标签Species



- 1.对鸢尾花的三个亚种（setosa、versicolor、virginica）分别统计它们的花萼长度（Sepal.Length）、花萼宽度（Sepal.Width）、花瓣长度（Petal.Length）、花瓣宽度（Petal.Width）的最大值、最小值、均值、中位数等特征，并绘制箱线图进行展示
2. 请问鸢尾花的四种属性花萼长度（Sepal.Length）、花萼宽度（Sepal.Width）、花瓣长度（Petal.Length）、花瓣宽度（Petal.Width）之间是否存在相关性？如何展示？##
3. 作为一个生物学家，你能从这些数据中得到什么结论吗？

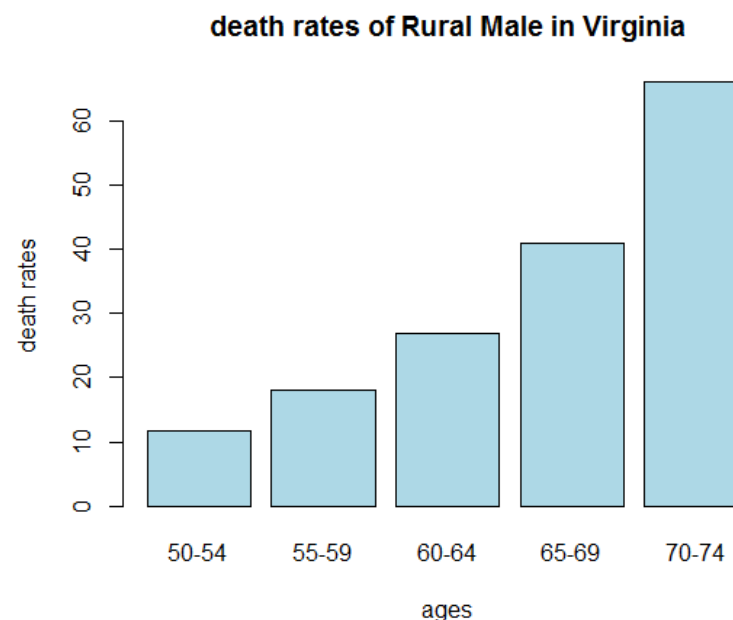
柱形图

基本样式: `barplot(height, width, beside, col, border, ...)`, 其中`height`为y轴数值, `width`为柱形宽度, `beside`为是否堆叠柱形, T为是, F为否 (default)。

VADeaths (**Death Rates in Virginia (1940)**)

	Rural Male	Rural Female	Urban Male	Urban Female
50-54	11.7	8.7	15.4	8.4
55-59	18.1	11.7	24.3	13.6
60-64	26.9	20.3	37.0	19.3
65-69	41.0	30.9	54.6	35.1
70-74	66.0	54.3	71.1	50.0

```
barplot(VADeaths[,1], col="lightblue",  
main="death rates of Rural Male in  
Virginia", xlab="ages", ylab="death  
rates")
```



如何调整柱形图的样式?

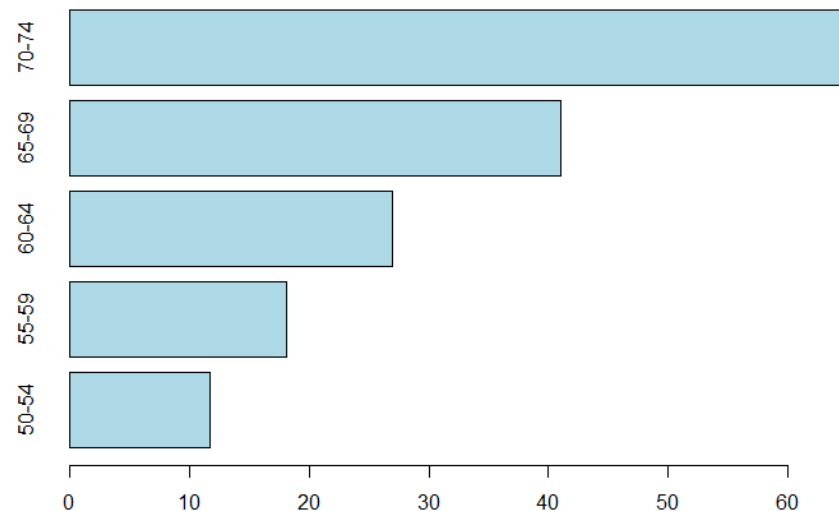
柱形图

基本样式: `barplot(height, width, beside, col, border, ...)`, 其中`height`为y轴数值, `width`为柱形宽度, `beside`为是否堆叠柱形, T为是, F为否 (default)。

VADeaths (**Death Rates in Virginia (1940)**)

水平布局柱形图: `barplot(x, horiz = TRUE)`

```
barplot(VADeaths[,1], col="lightblue",  
horiz=TRUE)
```



如何调整柱形图的样式?

柱形图

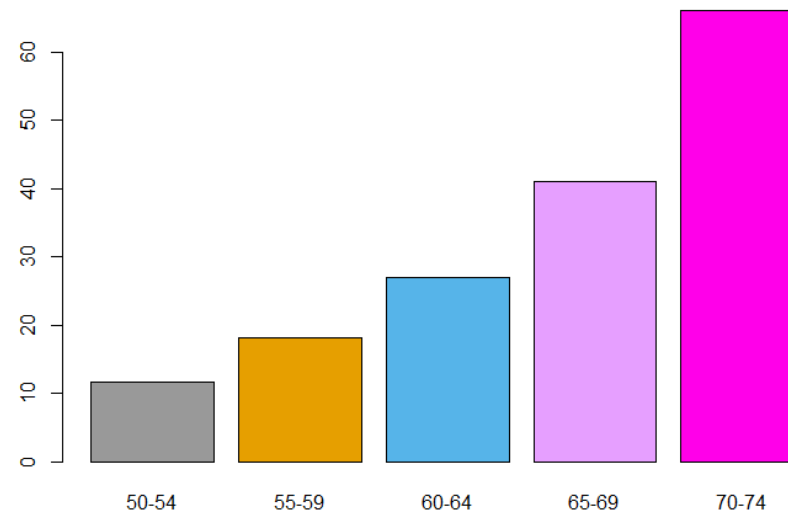
基本样式: `barplot(height, width, beside, col, border, ...)`, 其中`height`为y轴数值, `width`为柱形宽度, `beside`为是否堆叠柱形, T为是, F为否 (default)。

VADeaths (**Death Rates in Virginia (1940)**)

改变柱形图的颜色: `barplot(x, col = "white", border = "steelblue")`

`barplot(VADeaths[,1], col = "white", border = "steelblue")`

`barplot(VADeaths[,1], col=c("#999999", "#E69F00", "#56B4E9", "#E69FFF", "#FF00E9"))`



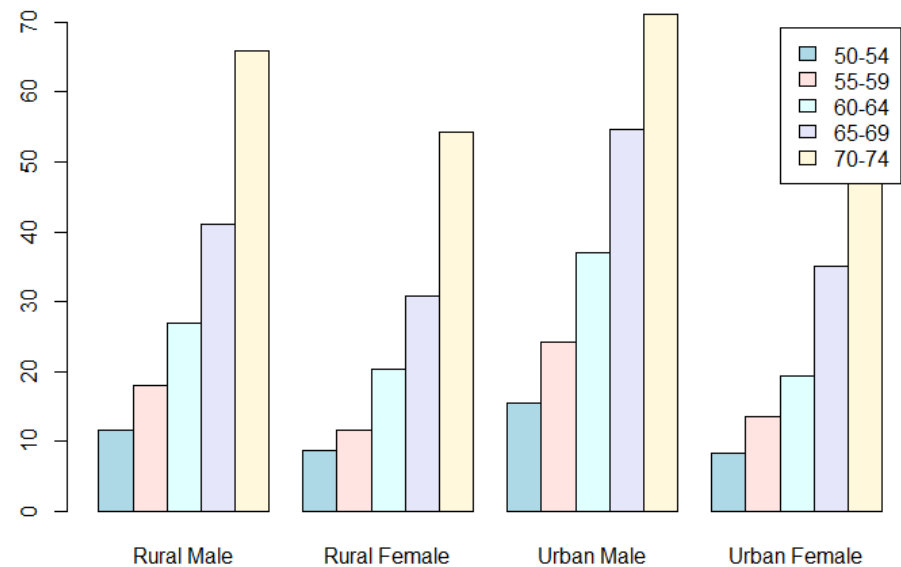
如何调整柱形图的样式?

柱形图

基本样式: `barplot(height, width, beside, col, border, ...)`, 其中`height`为y轴数值, `width`为柱形宽度, **`beside`**为是否堆叠柱形, T为是, F为否 (default)。

VADeaths (**Death Rates in Virginia (1940)**)

改变柱形图分组样式: `barplot(VADeaths, col = c("lightblue", "mistyrose", "lightcyan", "lavender", "cornsilk"), legend = rownames(VADeaths), beside = TRUE)`



饼图

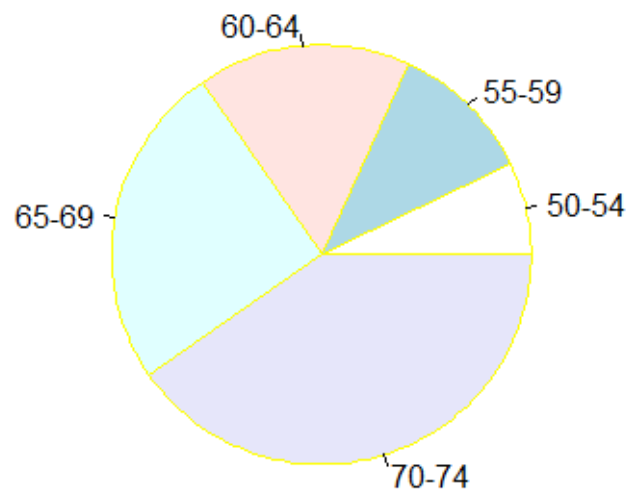
基本样式: `pie(x, labels, clockwise, col, border, ...)`, 其中x为非负数值, labels为切片名称, clockwise为是否顺时针画图, T为是, F为否 (default)。

VADeaths

	Rural Male	Rural Female	Urban Male	Urban Female
50-54	11.7	8.7	15.4	8.4
55-59	18.1	11.7	24.3	13.6
60-64	26.9	20.3	37.0	19.3
65-69	41.0	30.9	54.6	35.1
70-74	66.0	54.3	71.1	50.0

```
pie(VADeaths[,1],main="death rates of Rural Male in Virginia",border="yellow")
```

death rates of Rural Male in Virginia



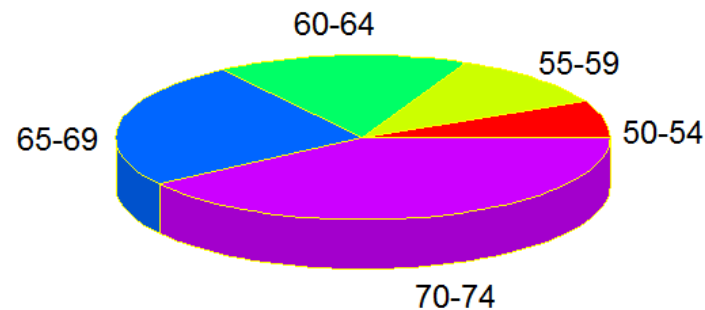
饼图

画3D的饼图：基本样式：`pie3D(x,radius=1,height=0.1, ...)`，其中x为非负数值，radius为半径长度，height为饼图的高度。

VADeaths

	Rural Male	Rural Female	Urban Male	Urban Female
50-54	11.7	8.7	15.4	8.4
55-59	18.1	11.7	24.3	13.6
60-64	26.9	20.3	37.0	19.3
65-69	41.0	30.9	54.6	35.1
70-74	66.0	54.3	71.1	50.0

death rates of Rural Male in Virginia



1. 安装R包:

```
install.packages("plotrix")
```

2. 导入安装的R包

```
library("plotrix")
```

3. 调用R函数

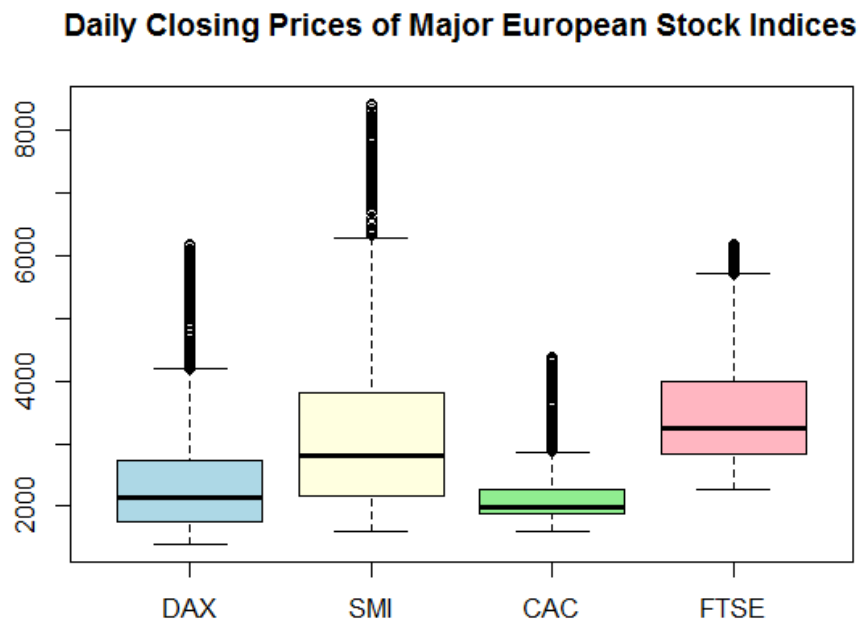
```
pie3D(VADeaths[,1],main="death rates of Rural Male in  
Virginia",border="yellow", labels=names(VADeaths[,1]))
```

箱线图

基本样式: `boxplot(x, names, outline, col, border, ...)`, 其中x为数据集, names为组名, outline为是否显示异常值, T为是 (default), F为否。

```
> head(EuStockMarkets)
```

	DAX	SMI	CAC	FTSE
[1,]	1628.75	1678.1	1772.8	2443.6
[2,]	1613.63	1688.5	1750.5	2460.2
[3,]	1606.51	1678.6	1718.0	2448.2
[4,]	1621.04	1684.1	1708.1	2470.4
[5,]	1618.16	1686.6	1723.1	2484.7
[6,]	1610.61	1671.6	1714.3	2466.8



```
boxplot(EuStockMarkets, col=c("lightblue", "lightyellow", "lightgreen", "lightpink"),  
main="Daily Closing Prices of Major European Stock Indices")
```


箱线图

基本样式: `boxplot(x, names, outline, col, border, ...)`, 其中x为数据集, names为组名, outline为是否显示异常值, T为是 (default), F为否。

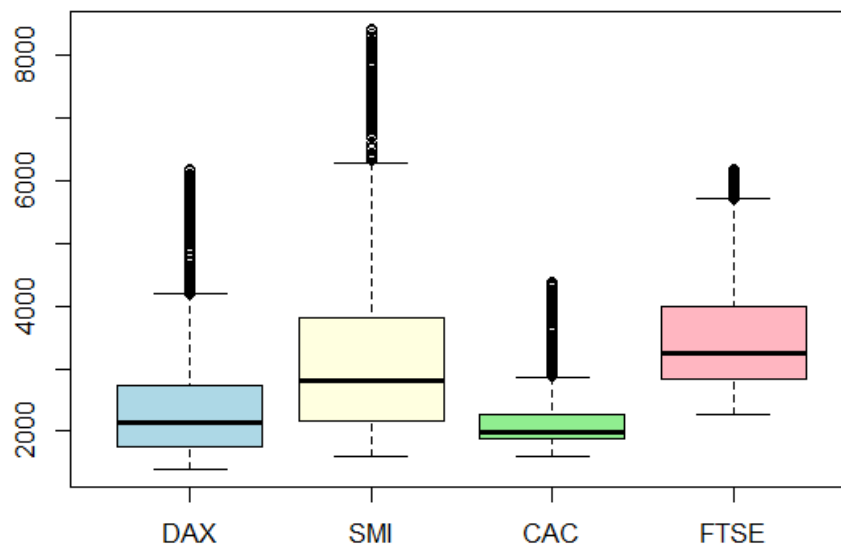
Daily Closing Prices of Major European Stock Indices, 1991-1998

欧洲主要股指日收盘价

```
> head(EuStockMarkets)
```

	DAX	SMI	CAC	FTSE
[1,]	1628.75	1678.1	1772.8	2443.6
[2,]	1613.63	1688.5	1750.5	2460.2
[3,]	1606.51	1678.6	1718.0	2448.2
[4,]	1621.04	1684.1	1708.1	2470.4
[5,]	1618.16	1686.6	1723.1	2484.7
[6,]	1610.61	1671.6	1714.3	2466.8

Daily Closing Prices of Major European Stock Indices



```
boxplot(EuStockMarkets, col=c("lightblue", "lightyellow", "lightgreen", "lightpink"),  
main="Daily Closing Prices of Major European Stock Indices")
```

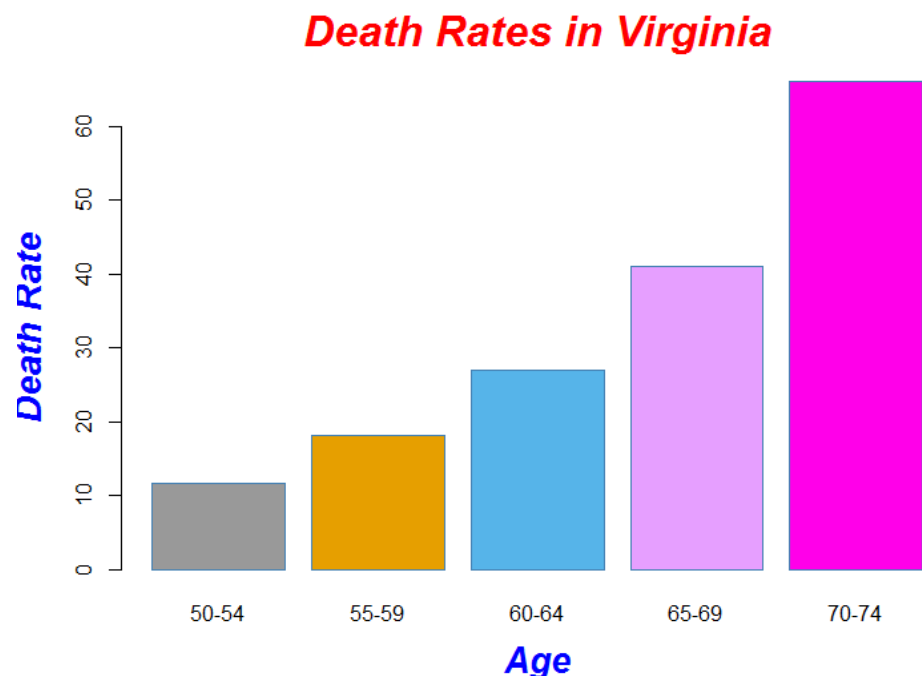
R的图形参数 (**Graphical parameters**)

如何改变或自定义R的图形参数/样式?

R的图形参数 (Graphical parameters)

2. 自定义main title, axis labels, title colors, font size, font style

```
barplot(VADeaths[,1], col=c("#999999", "#E69F00", "#56B4E9", "#E69FFF", "#FF00E9"), border = "steelblue",  
main="Death Rates in Virginia", xlab="Age", ylab="Death Rate", col.main="red", col.lab="blue", cex.main=2,  
cex.lab=1.7, font.main=4, font.lab=4)
```



cex: 指定符号的大小。cex是一个数值,表示绘图符号相对于默认大小的缩放倍数。默认大小为1, 1.5表示放大为默认值的1.5倍, 0.5表示缩小为默认值的50%等。

cex.lab: 坐标轴标签(名称)的缩放倍数,类似于cex

cex.main: 标题的缩放倍数,类似于cex

col: 默认的绘图颜色。

col.lab: 坐标轴标签(名字)的颜色

col.main: 标题颜色

font: 整数。用于指定绘图所用的字体样式. 1=常规. 2=粗体. 3=斜体. 4=粗斜体. 5=符号字体(以Adobe符号编码表示)

font.lab: 坐标轴标签(名字)的字体样式

font.main: 标题字体样式

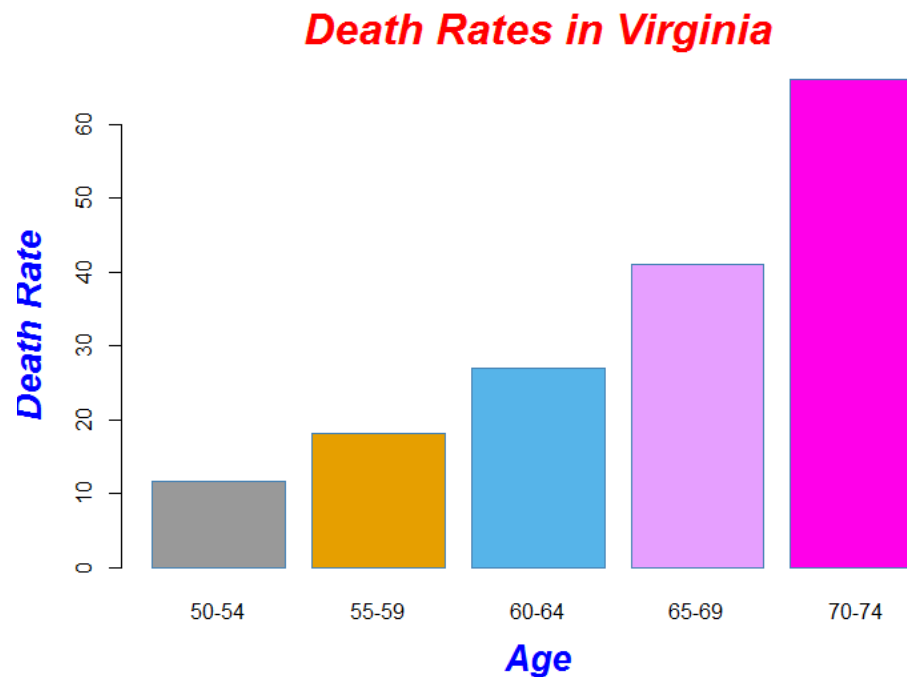
R的图形参数 (Graphical parameters)

1. 自定义main title, axis labels, title colors, font size

使用**title()**函数

title(main = NULL, sub = NULL, xlab = NULL, ylab = NULL, ...)

1. **barplot**(VADeaths[,1], col=c("#999999", "#E69F00", "#56B4E9", "#E69FFF", "#FF00E9"), border = "steelblue")
2. **title**(**main**="Death Rates in Virginia", **xlab**="Age", **ylab**="Death Rate", **col.main**="red", **col.lab**="blue", **cex.main**=2, **cex.lab**=1.7, **font.main**=4, **font.lab**=4)



R的图形参数 (Graphical parameters)

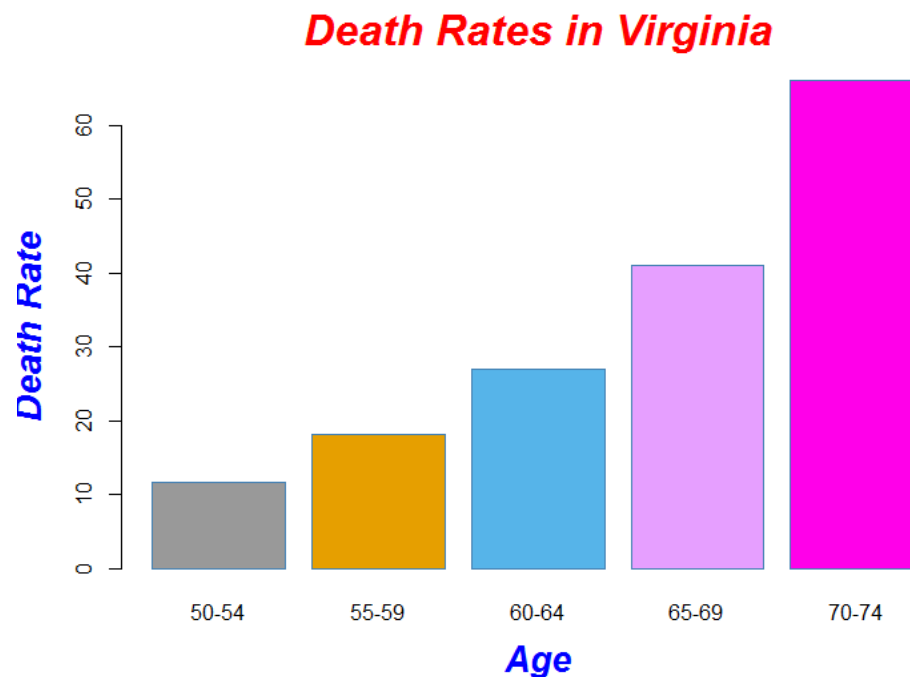
1. 自定义main title, axis labels, title colors, font size

使用par()函数，par()函数被称为“永久修改”，对整个R进程都有效

par(main = NULL, sub = NULL, xlab = NULL, ylab = NULL, ...)

1. par(col.main="red", col.lab="blue", col.sub="black", cex.main=2, cex.lab=1.7, font.main=4, font.lab=4)

2. barplot(VADeaths[,1], col=c("#999999", "#E69F00", "#56B4E9", "#E69FFF", "#FF00E9"), border = "steelblue", main="Death Rates in Virginia", xlab="Age", ylab="Death Rate")



R的图形参数 (Graphical parameters)

2. 添加图例

`legend(x, y=NULL, legend, col)`

x and y: 图例的坐标, x 可以为 "bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright", "right" and "center".

legend: 图例的文本

col: 线或点的颜色

画第一条线

```
x<-1:9;
```

```
y1=x*x;
```

```
y2=2*y1 ;
```

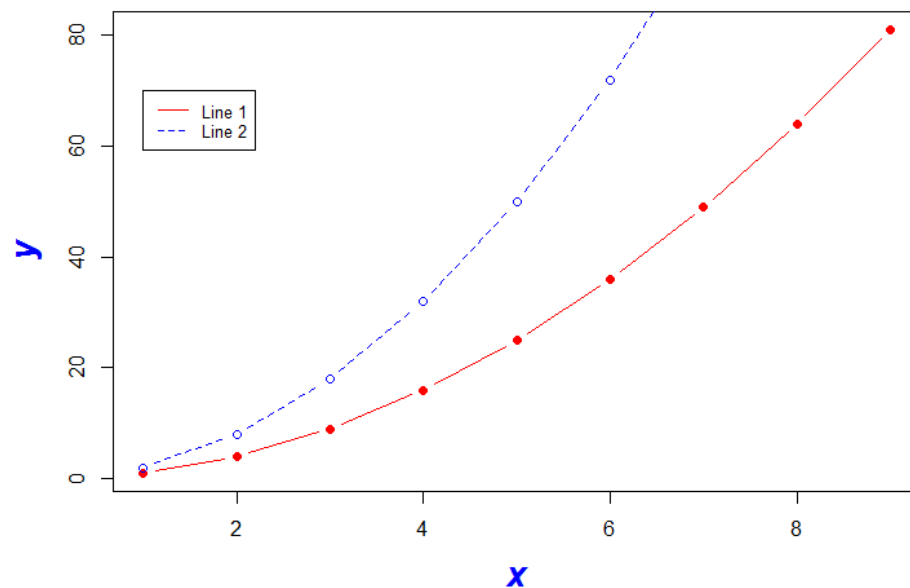
```
plot(x, y1, type="b", pch=19, col="red", xlab="x", ylab="y")
```

增加一条线

```
lines(x, y2, pch=21, col="blue", type="b", lty=2)
```

#增加图例

```
legend(1, 70, legend=c("Line 1", "Line 2"), col=c("red", "blue"), lty=1:2, cex=0.8)
```



R的图形参数 (Graphical parameters)

2. 添加图例

legend(x, y=NULL, legend, col)

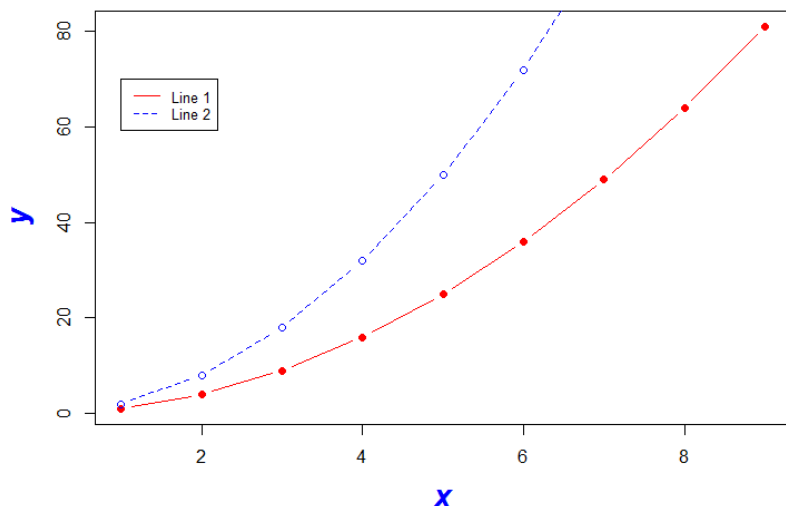
x and y: 图例的坐标, x 可以为 "bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright", "right" and "center".

legend: 图例的文本

col: 线或点的颜色

plot(x, y1, **type**="b", **pch**=19, **col**="red", **xlab**="x", **ylab**="y")

legend(1, 70, **legend**=c("Line 1", "Line 2"), **col**=c("red", "blue"), **lty**=1:2, **cex**=0.8)



type: 指定所绘图形类型

type

what type of plot should be drawn. Possible types are

- "p" for points,
- "l" for lines,
- "b" for both,
- "c" for the lines part alone of "b",
- "o" for both 'overplotted',
- "h" for 'histogram' like (or 'high-density') vertical lines,
- "s" for stair steps,
- "S" for other steps, see 'Details' below,
- "n" for no plotting.

Line types: lty.

- | | |
|--------------|---------------------|
| 6.'twodash' | ---- |
| 5.'longdash' | — — — — — |
| 4.'dotdash' | - . - . - . - . - . |
| 3.'dotted' | |
| 2.'dashed' | ----- |
| 1.'solid' | ————— |
| 0.'blank' | |

R的图形参数 (Graphical parameters)

3. 添加文本 (Add texts)

`text(x, y, labels)`

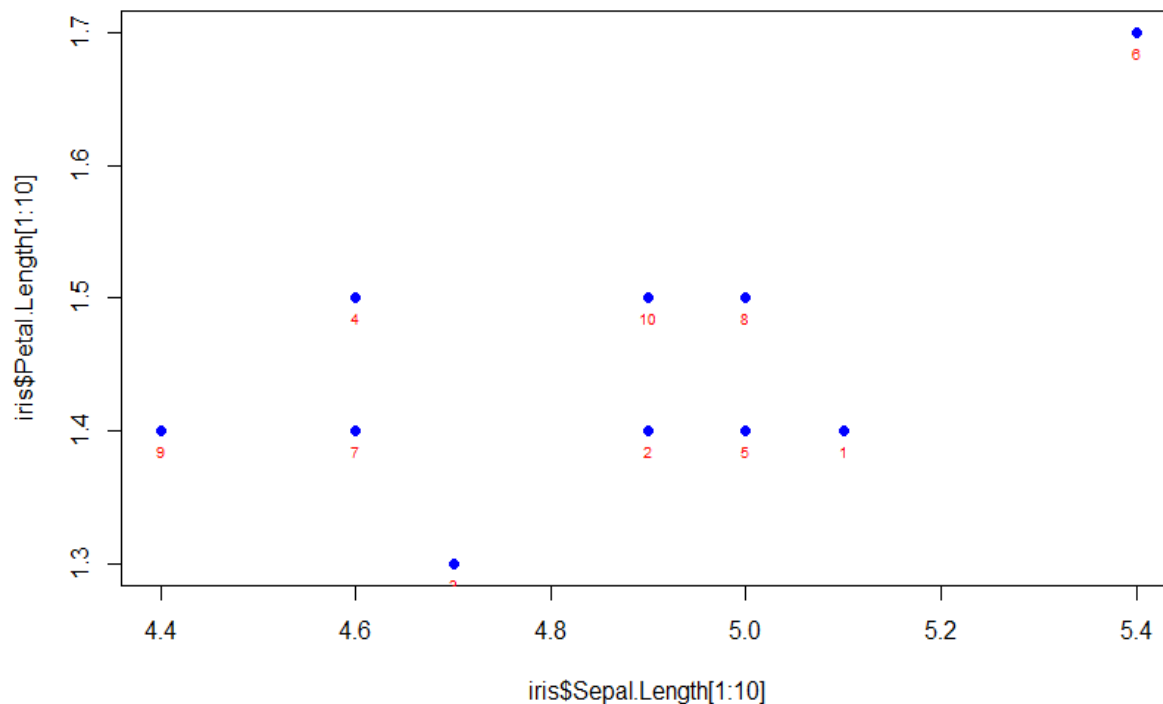
x and y: 文本的坐标

labels: 一个字符向量: 包含添加的文本

```
plot(iris$Sepal.Length[1:10], iris$Petal.Length[1:10], pch=19, col="blue")
```

```
text(iris$Sepal.Length[1:10], iris$Petal.Length[1:10],  
     row.names(iris[1:10,]), cex=0.65, pos=1, col="red")
```

pos a position specifier for the text. If specified this overrides any `adj` value given. Values of 1, 2, 3 and 4, respectively indicate positions below, to the left of, above and to the right of the specified (x, y) coordinates.



R的图形参数 (Graphical parameters)

4. 添加直线 (Add straight lines)

abline(a=NULL, b=NULL, h=NULL, v=NULL, ...)

a, b : 线的截距 (**intercept**) 和斜度 (**slope**)

h : **y-value(s)** 水平线 (horizontal line)

v : **x-value(s)** 垂直线 (vertical line)

plot(iris\$Sepal.Length, iris\$Petal.Length, **pch=19**)

Add vertical line

abline(v=6, col="blue")

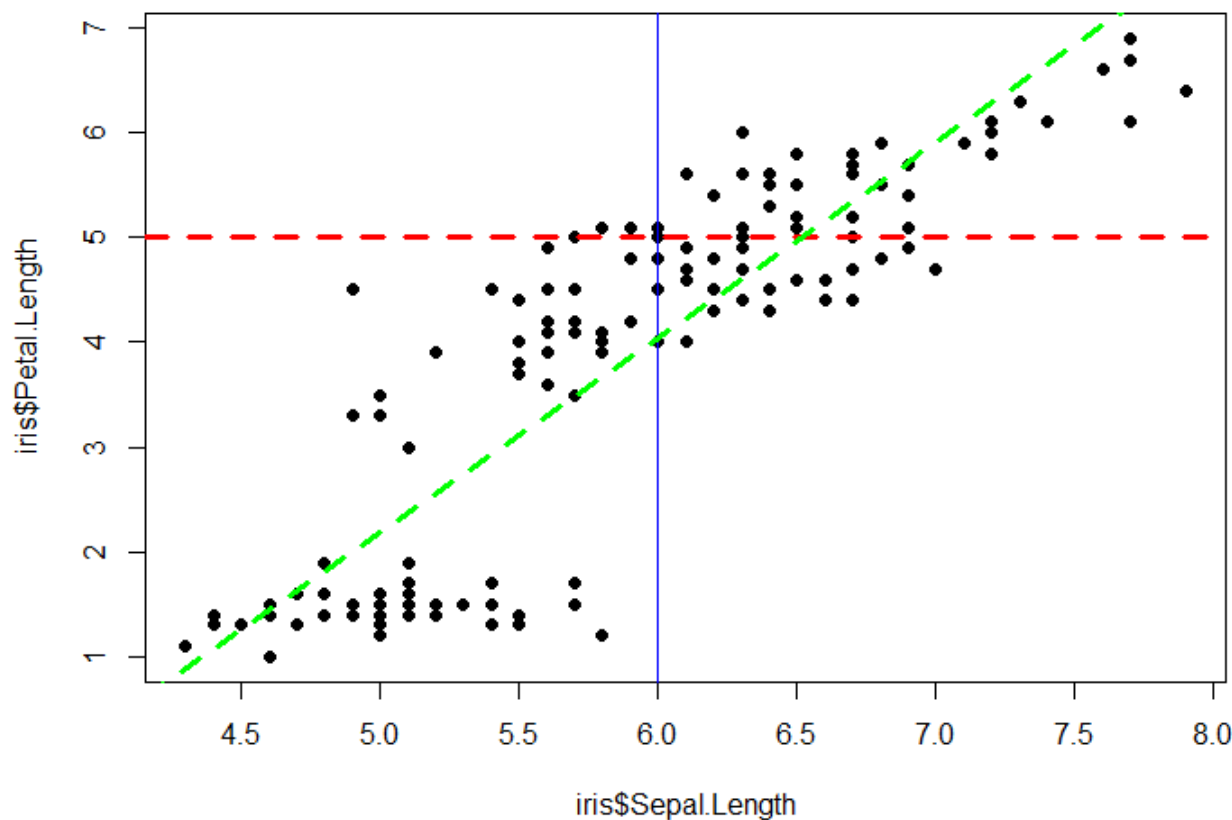
Add horizontal line, change line color, size and type

abline(h=5, col="red", lty=2, lwd=3)

Add regression line

reg<-**lm**(iris\$Petal.Length~iris\$Sepal.Length, **data** = iris)

abline(reg, col="green", lty=2, lwd=3)



R的图形参数 (Graphical parameters)

5. 添加坐标轴 (Add an axis)

`axis(side, at=NULL, labels=TRUE)`

side: 坐标放置的位置, 值可为:

1(below), 2(left), 3(above) and 4(right).

at: the **points** at which **tick-marks** are to be drawn.

labels: **vector of texts** for the labels of **tick-marks**.

```
x<-1:5;
```

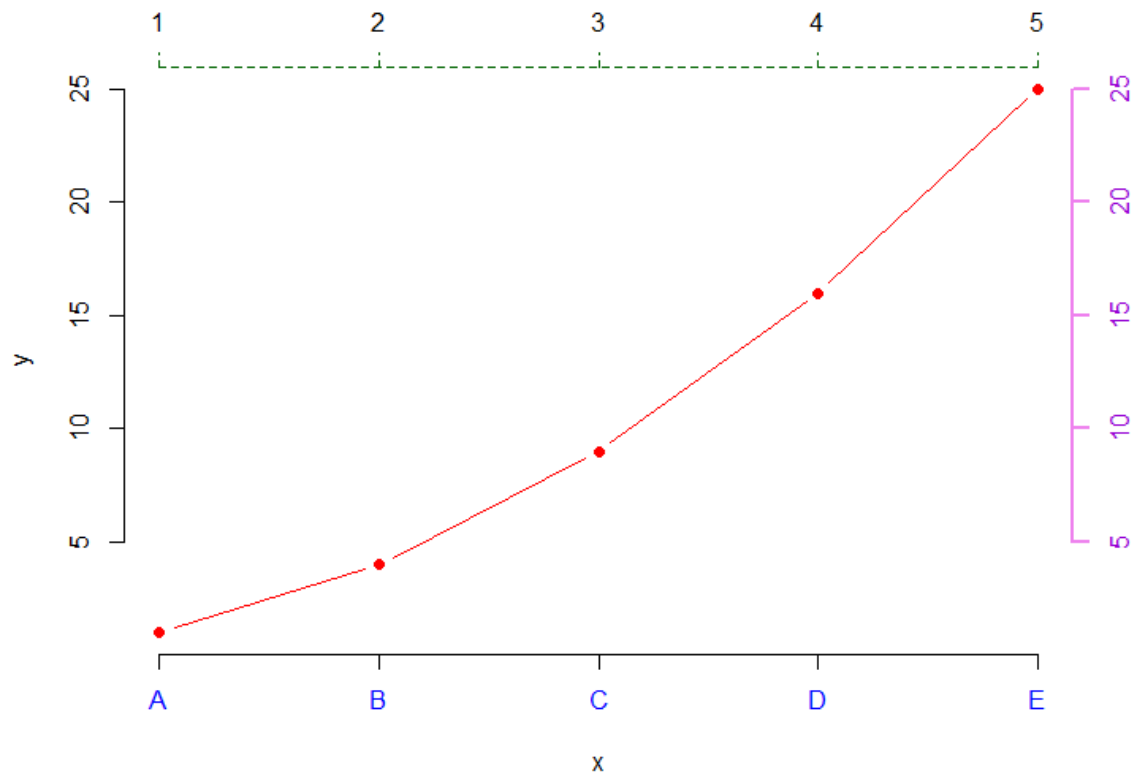
```
y=x*x
```

```
plot(x, y, pch=19, col="red", type="b", frame=FALSE, xaxt="n")
```

```
axis(1, 1:5, LETTERS[1:5], col.axis="blue")
```

```
axis(3, col = "darkgreen", lty = 2, lwd = 0.5)
```

```
axis(4, col = "violet", col.axis = "dark violet", lwd = 2)
```



R的图形参数 (Graphical parameters)

6. 改变坐标轴的范围 (Change axis scale)

Xlim和**ylim** 参数去限定x轴和y轴的范围

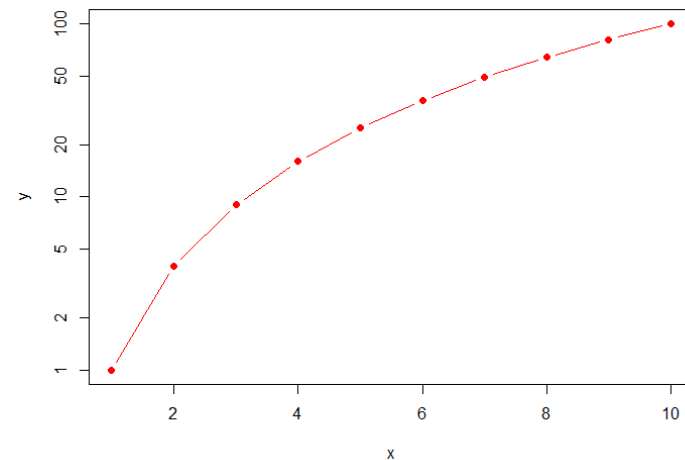
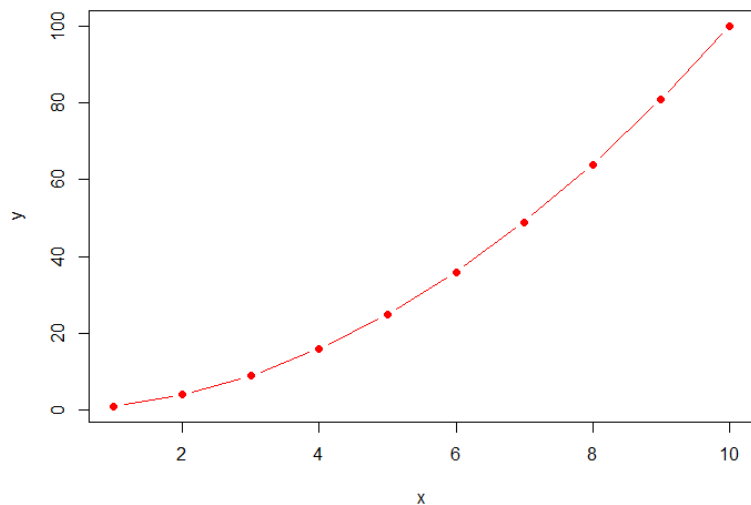
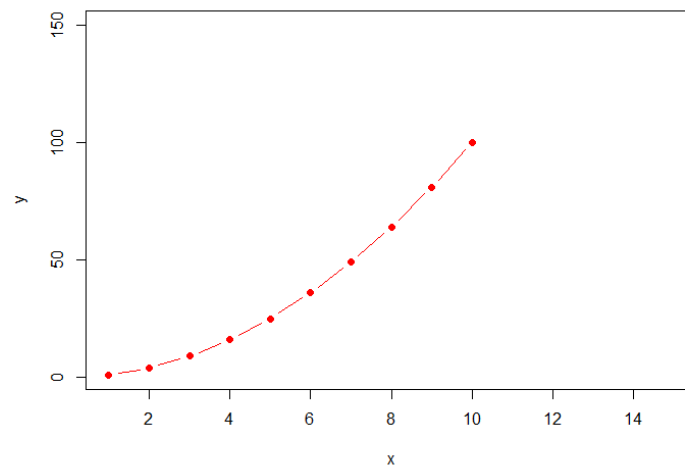
格式Format : **xlim = c(min, max); ylim = c(min, max)**.

log 参数进行log转换: **log="x", log="y" or log="xy"**.

plot(x, y, pch=19, col="red", type="b")

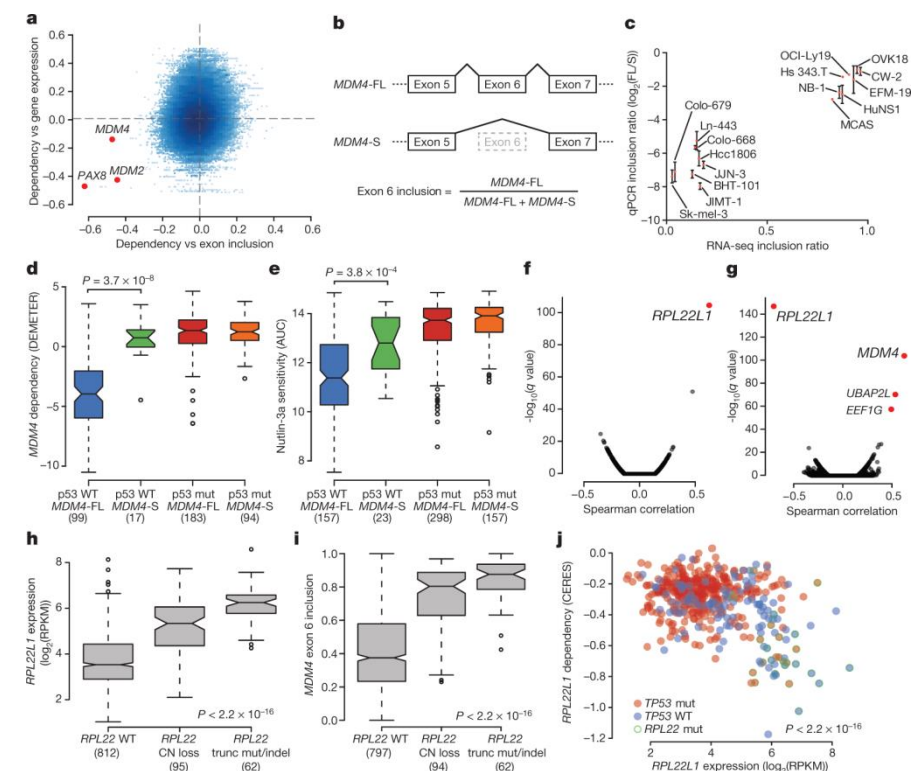
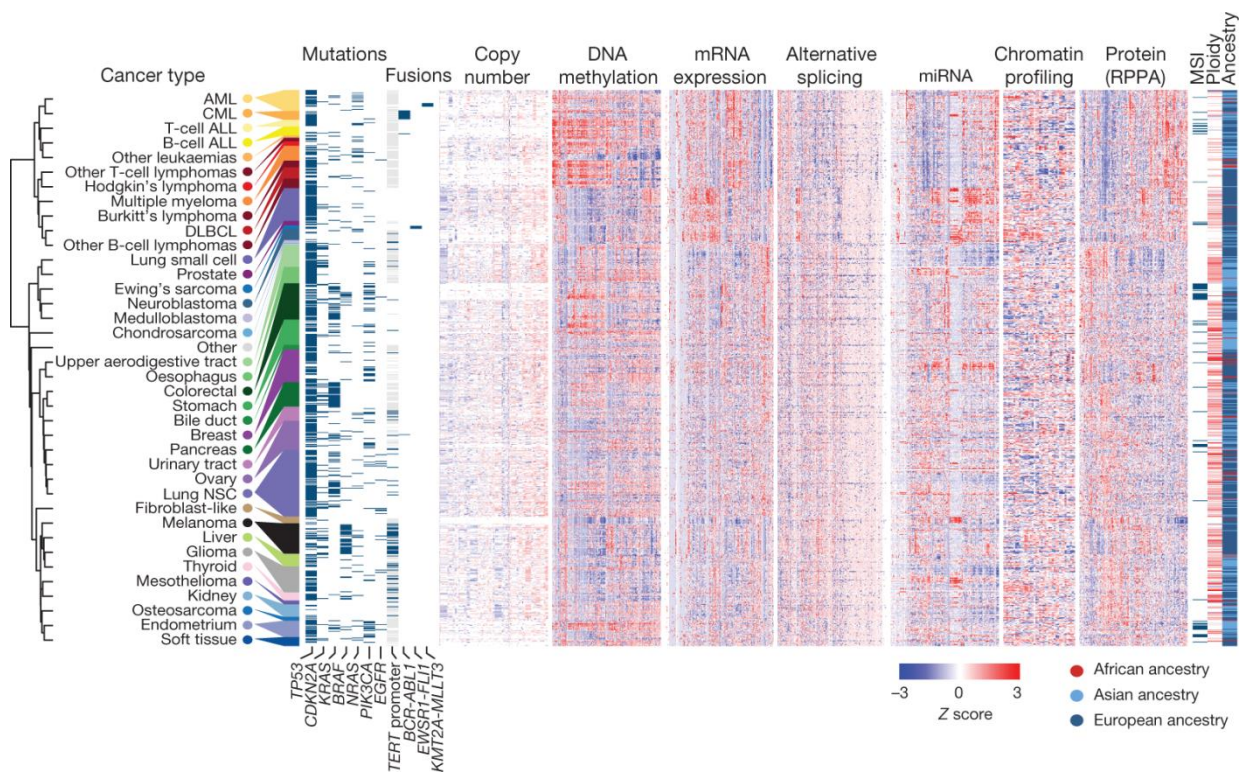
plot(x, y, pch=19, col="red", type="b", xlim=c(1,15), ylim=c(1,150))

plot(x, y, pch=19, col="red", type="b", log="y")



R图形在生物学的应用

大数据的论文，R做出的各种自定义的图形。



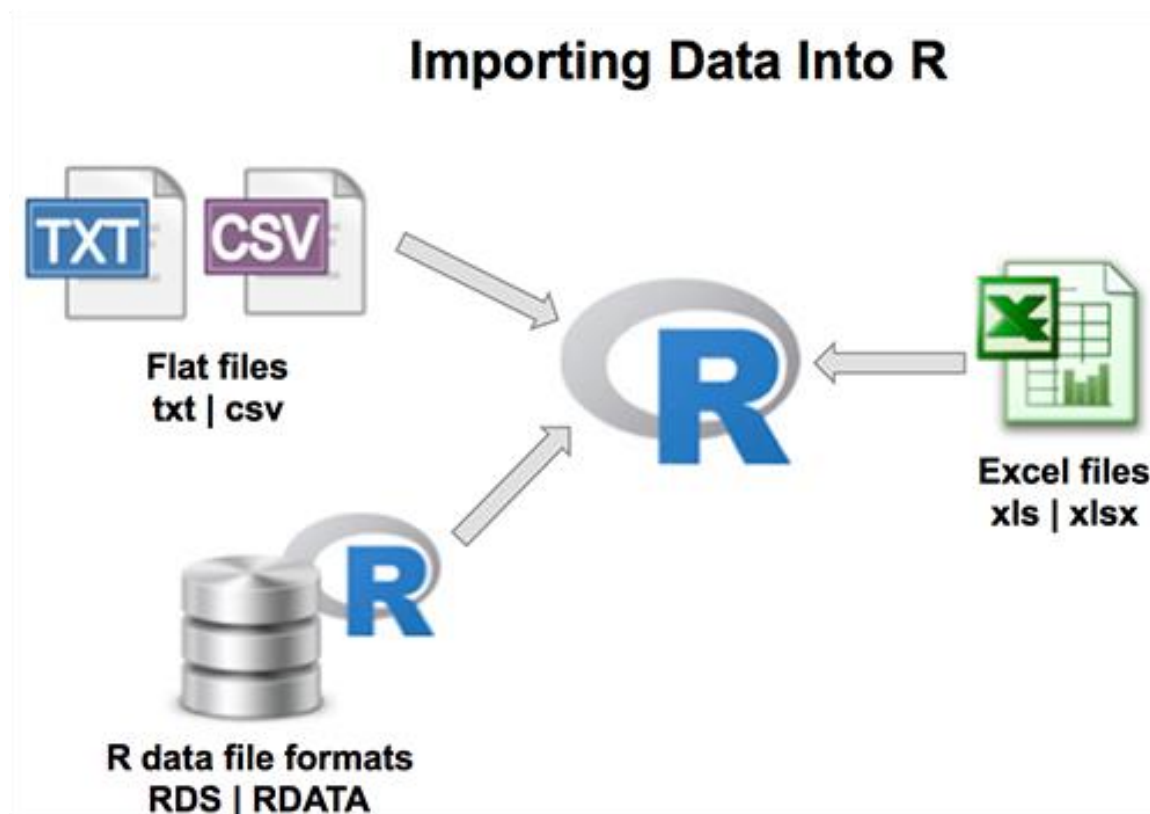
Cancer Cell Line Encyclopedia: <https://www.nature.com/articles/s41586-019-1186-3>

如何对自己的数据进行作图分析

如何读取数据进行作图分析？

如何对自己的数据进行作图分析

R可以读取多种数据格式的文件的内容



如何对自己的数据进行作图分析

准备你自己的数据

注意事项:

1. 用第一行作为列表头（列名，**column names**）
列代表变量（**variables**）。
2. 用第一列作为行名（**row names**）。行代表观测值（**observations**）。
3. 列名和行名都不重复（**unique**）。
4. 避免列名或行名有空格（**blank spaces**）。
5. 避免列名或行名以数字开头。
6. 用NA(not available)替换缺失值（**missing values**）。

The screenshot shows an Excel spreadsheet with a dataset. The first row (row 1) contains column names: 'miRNA', '816038GSM', 'GSM 816039', 'GSM_816040', 'GSM816041', and 'GSM816042'. The first column (column A) contains row names: 'miRNA', 'hsa-let-7a', 'hsa-let-7b', 'hsa-let-7d', 'hsa-let-7e', 'hsa-miR-100', 'hsa-miR-101', 'hsa-miR-103', 'hsa-miR-105', 'hsa-let-7f', 'hsa-let-7i', 'hsa-miR-1', 'hsa-miR-127-3p', 'hsa-miR-127-5p', 'hsa-miR-128a', and 'hsa-miR-129-3p'. Annotations include: a red box around '816038GSM' and 'GSM 816039' with the text 'Bad names'; a green box around 'GSM_816040', 'GSM816041', and 'GSM816042' with the text 'Good names'; and a yellow box around a cell in row 10, column F with the text 'Missing data'.

	A	B	C	D	E	F
1	miRNA	816038GSM	GSM 816039	GSM_816040	GSM816041	GSM816042
2	hsa-let-7a	11.3723211	9.25255394	10.3553349	9.31817627	9.730364799
3	hsa-let-7b	11.2871121	10.36092377	10.63897	10.03401184	11.2725935
4	hsa-let-7d		21			9.896810532
5	hsa-let-7e	10.00422001	9.331817627	9.145921707	8.743097305	10.4216938
6	hsa-miR-100	10.2312212	9.627078056	9.327999115		9.403511047
7	hsa-miR-101	9.702172279	8.431631088	8.793889046	9.026667595	8.413924217
8	hsa-miR-103		9.316625595	10.18142414	10.81638336	10.55797958
9	hsa-miR-105	9.074141502	8.748454094		10.70946121	9.316625595
10	hsa-let-7f	11.15860939	9.19231987	9.668526649	9.647512436	
11	hsa-let-7i	11.5684309				10.73631954
12	hsa-miR-1	8.592456818	8.35747242	8.559271049	8.847042899	9.173212051
13	hsa-miR-127-3p	8.816983223	8.500180244	8.969026566	8.608224869	8.978338242
14	hsa-miR-127-5p	9.579316139	9.342801094	9.441287994	9.053165436	8.285809517
15	hsa-miR-128a	9.03617382	8.406779289	8.460775375	8.695280075	8.97591114
16	hsa-miR-129-3p	9.802515984	9.583559036	9.423906326	8.998573303	8.455832481


如何对自己的数据进行作图分析

准备你自己的数据

注意事项:

1. 用第一行作为列表头（列名，**column names**）
列代表变量（**variables**）。
2. 用第一列作为行名（**row names**）。行代表观测值（**observations**）。
3. 列名和行名都不重复（**unique**）。
4. 避免列名或行名有空格（**blank spaces**）。
5. 避免列名或行名以数字开头。
6. 用NA(not available)替换缺失值（**missing values**）。


修正后的数据



	A	B	C	D	E	F
1	miRNA	G816038GSM	GSM_816039	GSM_816040	GSM816041	GSM816042
2	hsa-let-7a	11.37232113	9.25255394	10.35053349	9.331817627	9.730364799
3	hsa-let-7b	11.2877121	10.36092377	10.657897	10.03401184	11.2725935
4	hsa-let-7d	10.97799492	9.044920921	9.931402206	9.550902367	9.896810532
5	hsa-let-7e	10.00422001	9.331817627	9.145921707	8.743097305	10.4216938
6	hsa-miR-100	10.2312212	9.627078056	9.327999115	NA	9.403511047
7	hsa-miR-101	9.702172279	8.431631088	8.793889046	9.026667595	8.413924217
8	hsa-miR-103	NA	9.316625595	10.18142414	10.81638336	10.55797958
9	hsa-miR-105	9.074141502	8.748454094	NA	10.70946121	9.316625595
10	hsa-let-7f	11.15860939	9.19231987	9.668526649	9.647512436	NA
11	hsa-let-7i	11.5684309	NA	10.14689064	10.27403927	10.73631954
12	hsa-miR-1	8.592456818	8.35747242	8.359271049	8.847642899	9.173212051
13	hsa-miR-127-3p	8.816983223	8.500180244	8.969026566	8.608224869	8.978338242
14	hsa-miR-127-5p	9.579316139	9.342801094	9.441287994	9.053165436	8.285809517
15	hsa-miR-128a	9.03617382	8.406779289	8.460775375	8.695280075	8.97591114
16	hsa-miR-129-3p	9.802515984	9.583559036	9.423906326	8.998573303	8.455832481
17						

从外部读取数据

最为常用的数据读取方式是用`read.table()`，`read.delim()`和`read.csv()`函数。

`txt`文件，制表符（`"\t"`）或逗号（`","`）分隔 

`csv`文件，逗号（comma，`","`）分隔

第1步 将Excel中的数据另存为`.txt`格式（制表符间隔）或`.csv`格式。

第2步 用`read.table()`，`read.delim()`，`read.csv()`函数将数据读入R工作空间，并赋值给一个对象。

read.table() 等的使用

基本样式: `read.table(file, header, sep, row.names, ...)`, 其中`file`表示文件名, 需用双引号; `header`表示是否把第一行作为表头, T为是, F为否(默认); `sep`表示分隔符, `"\t"`表示文件以制表符分隔, `","`表示以逗号分隔; `row.names`为可选, 表示指定哪一列(一般都是第一列)做为行名。

例:

```
1. chip<-read.table("chipplot.txt", header=T, sep="\t", row.names="miRNA")
```

```
2. irisData<-read.csv("iris.csv", header=T)
```

```
3. irisData<-read.delim("iris.txt", header=T, sep = "\t", dec = ".")
```

提示: `read.table(file.choose(), header=T, sep="\t")` 可以弹出对话框, 选择文件。

深度：数据框or矩阵

`read.table()` 从外部读取数据后，会转换成数据框，但有时候一些函数需要矩阵，如 `barplot()`，这时候需要把数据框转换为矩阵，即 `data.matrix(chip)`。

数据框和矩阵是两种不同的数据格式。

数据框同行可以包含不同类型的数据，如字符串或数字，而矩阵同行必须是相同类型的，且往往是数字。

读入学生名单

```
students<-read.csv("sysu_student.csv")
```

0	10	20	30	40
1	Id, Name, Major			
2	16336001, 阿比达·阿布来提, 生物技术			
3	16336007, 蔡静, 生物技术			
4	16336008, 蔡奇, 生物技术			
5	16336010, 蔡响, 生物科学			
6	16336014, 曾思琳, 生物科学			
7	16336019, 陈嘉杰, 生物科学			
8	16336025, 陈瑞琪, 生物技术			
9	16336029, 程海涛, 生物技术			
10	16336030, 程凯平, 生物技术			
11	16336031, 迟可欣, 生物技术			
12	16336033, 代智允, 生物技术			
13	16336035, 邓柯, 生物技术			
14	16336053, 郝远浩, 生物技术			
15	16336054, 何东印, 生物技术			
16	16336055, 何天龙, 生物技术			
17	16336056, 何欣桐, 生物技术			
18	16336057, 何永胜, 生物技术			
19	16336059, 胡芬芳, 生物技术			
20	16336063, 黄名海, 生物科学			
21	16336068, 黄芷杰, 生物技术 (生物技术及应用基地班)			
22	16336073, 金杰皓, 生物科学			
23	16336081, 李博今, 生物技术			
24	16336083, 李钧洋, 生物技术			
25	16336088, 李星碧, 生物技术			
26	16336097, 梁诗怡, 生物技术			
27	16336098, 梁新源, 生物技术 (生物技术及应用基地班)			
28	16336102, 林嘉煌, 生物技术			
29	16336106, 林逸轩, 生物科学			
30	16336110, 林子璇, 生物科学			
31	16336124, 刘玥楼, 生物科学			
32	16336136, 敏逸晖, 生物技术 (生物技术及应用基地班)			
33	16336141, 潘子晴, 生物技术 (生物技术及应用基地班)			

生成各科成绩

```
stu_num<-nrow(students) #nrow获取行数， 什么函数获取列数？
```

```
English<-sample(80:100,stu_num,replace=TRUE) # sample抽样
```

```
Math<-round(runif(stu_num,min=80,max=100)) #runif生成均匀随机数
```

```
Computer<-round(rnorm(stu_num,mean=90,sd=5)) #rnorm生成正态分布的随机数
```

```
Computer[which(Computer>100)]<- 100 #超过100分的， 设为100分。 which函数的用法
```

```
Student_score <- data.frame(students, English=English, Math=Math,
```

```
Computer=Computer) # 组合成新的数据框架
```

```
write.table(Student_score,file= "Student_score.csv" ) #输出结果到新的文件
```


计算各科平均分

```
mean(Student_score$Math)
```

```
## [1] 90.26812
```

```
mean(Student_score$English)
```

```
## [1] 90.18841
```

```
mean(Student_score$Computer)
```

```
## [1] 89.89855
```

Apply函数

apply(X, MARGIN, FUN, ...)

x: 一个数组或者矩阵

MARGIN: 两种数值1或者2决定对哪一个维度进行函数计算

MARGIN=1: 操作基于行

MARGIN=2: 操作基于列

MARGIN=c(1,2): 对行和列都进行操作

FUN: 使用哪种操作，内置的函数有**mean**（平均值）、**medium**（中位数）、**sum**（求和）、**min**（最小值）、**max**（最大值），当然还包括广大的用户自定义函数

```
Scores<-Student_score[,4:6]
```

```
apply(Scores,2,mean)
```

```
## English Math Computer
```

```
## 90.18841 90.26812 89.89855
```

计算各科最高分及最低分

```
apply(Scores,2,max)
```

```
## English Math Computer
```

```
## 100 100 100
```

```
apply(Scores,2,min)
```

```
## English Math Computer
```

```
## 80 80 77
```

求出每人总分及最高分同学

```
total_score<-apply(Scores,1,sum)

which.max(total_score) #决定最大、最小值等的索引，求最小值的索引函数是？

## [1] 16

Student_score[which.max(total_score),]

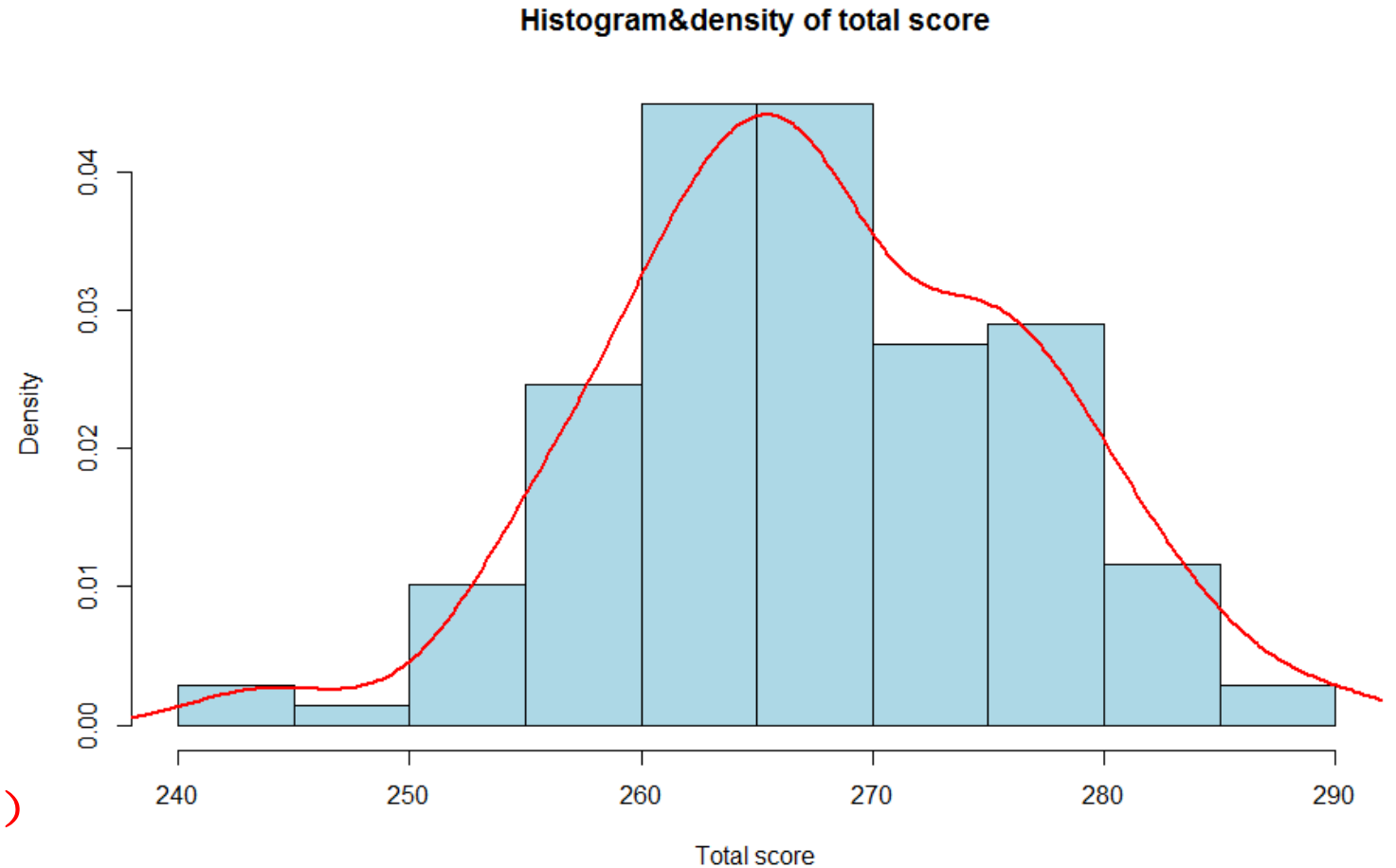
## Id Name Major English Math Computer

## 16 16336056 何欣桐 生物技术 93 99 100
```

画出成绩分布图(直方图)

```
hist(total_score,xlab = "Total  
score",breaks = 10,main =  
"Histogram&density of total  
score",col = "lightblue",freq =  
FALSE)  
lines(density(total_score),col="red",lwd=2)
```

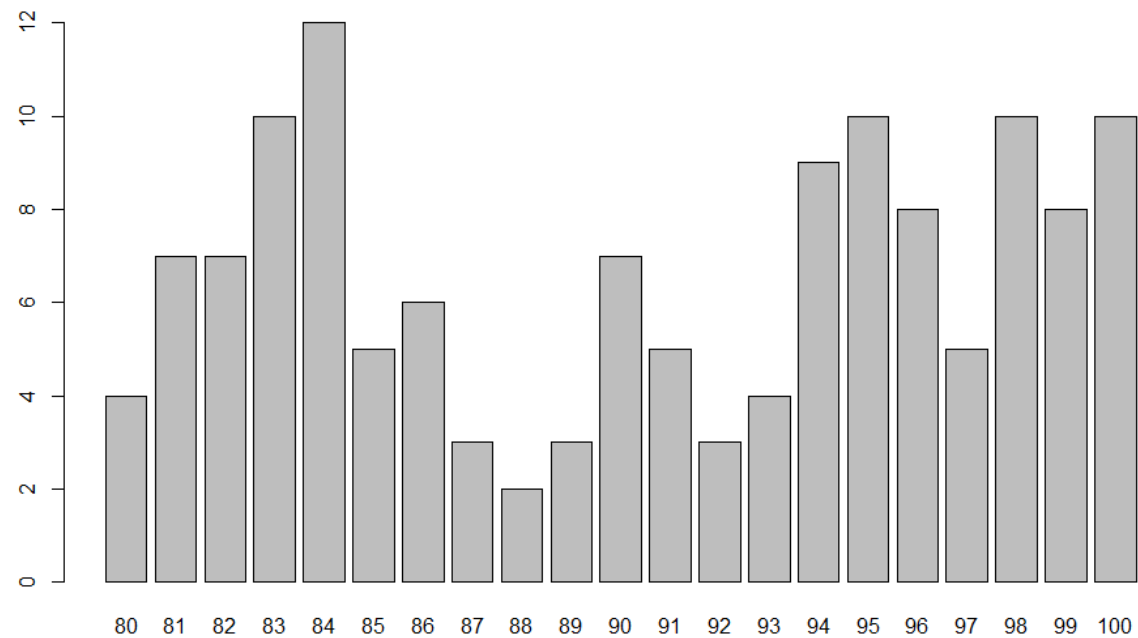
Density: Kernel Density Estimation (拟合曲线)



画出成绩分布图(条形图)

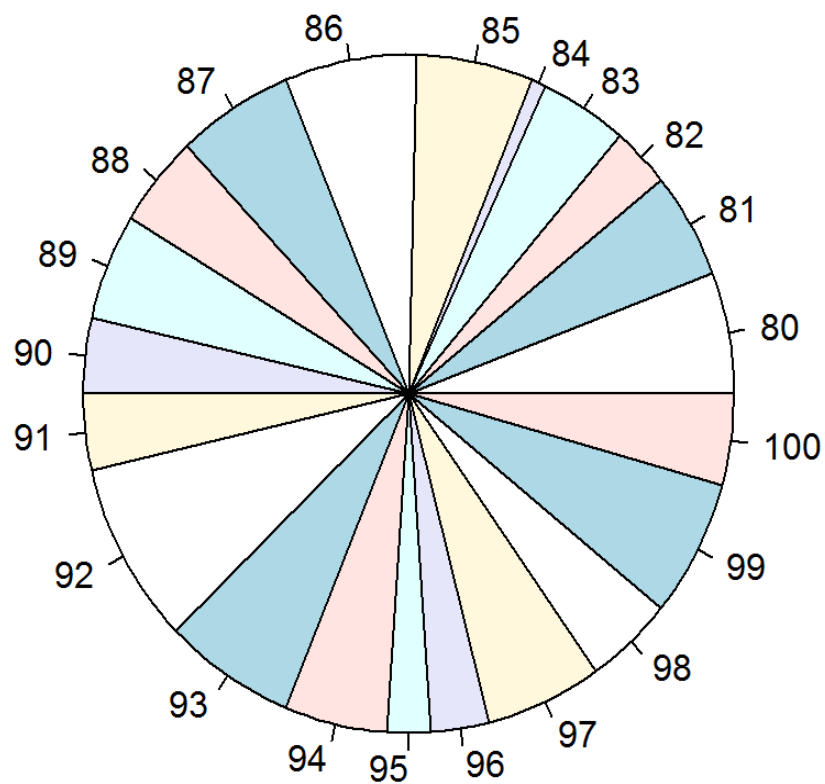
```
table(Scores$English)
##
## 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97
## 8 7 4 6 1 8 9 8 6 7 5 5 12 9 7 3 4 8
## 98 99 100
## 6 9 6
barplot(table(Scores$English))
```

table函数： 可以实现各数据频次的统计



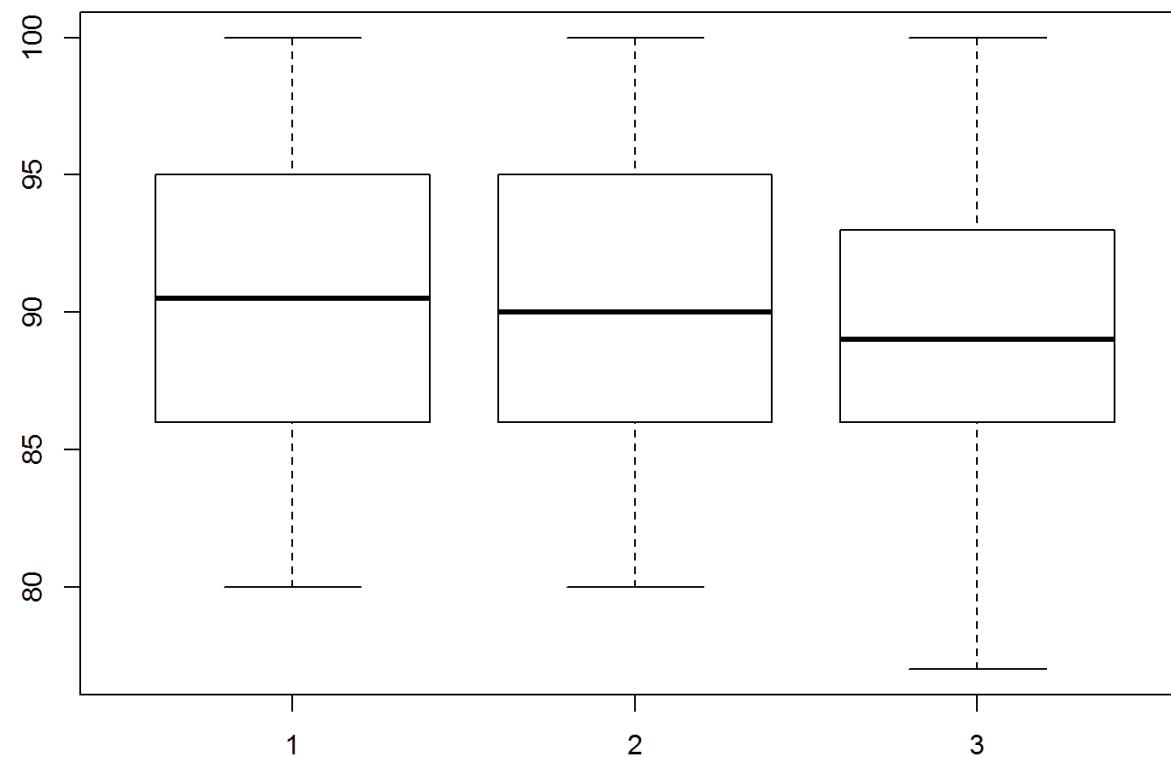
饼图

```
pie(table(Scores$English))
```



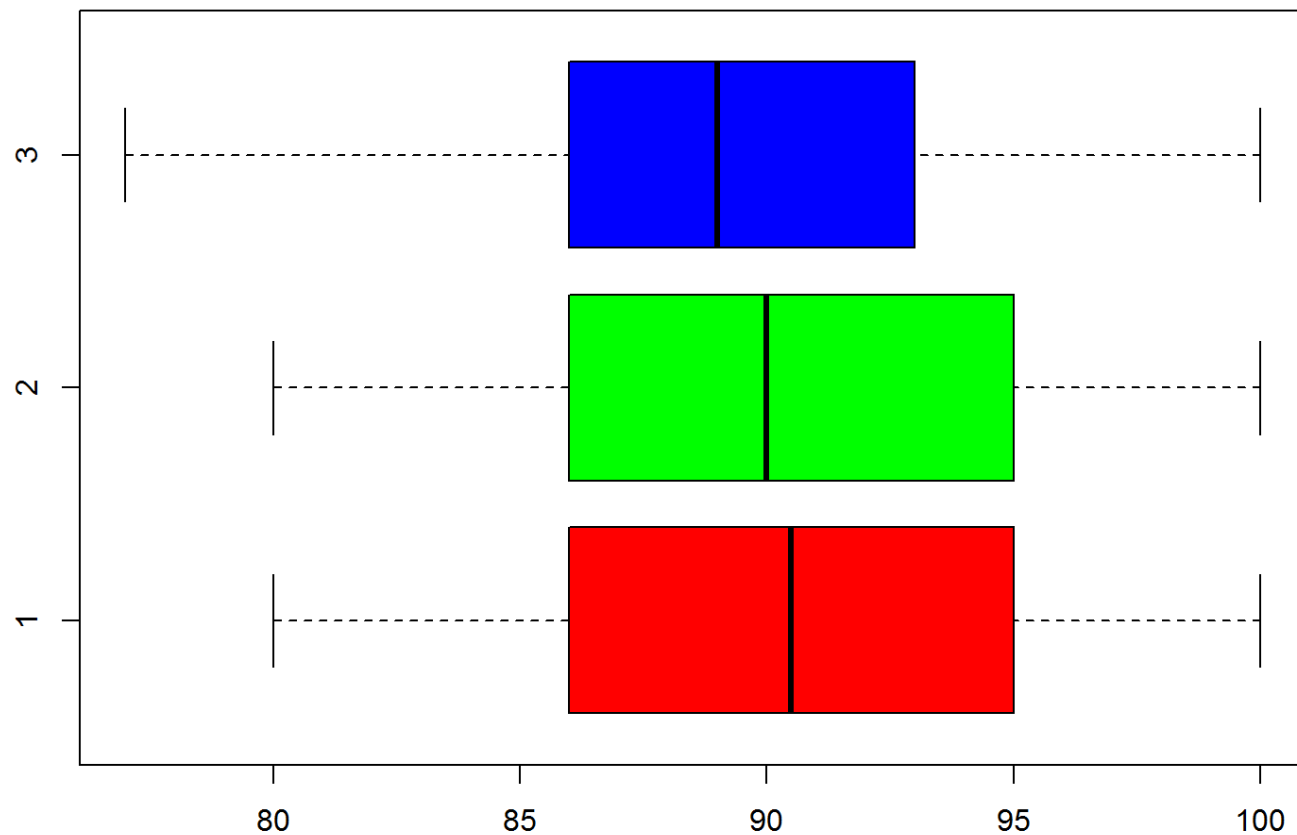
箱线图

```
boxplot(Scores$English, Scores$Math, Scores$Computer)
```



水平箱线图

```
boxplot(Scores$English, Scores$Math, Scores$Computer,  
col=c("red", "green", "blue"), horizontal = TRUE)
```



星相图

星相图（Star Plot）是用线段离中心的长度来表示变量值的大小。

`stars(x, full = TRUE, scale = TRUE, radius = TRUE,...)`

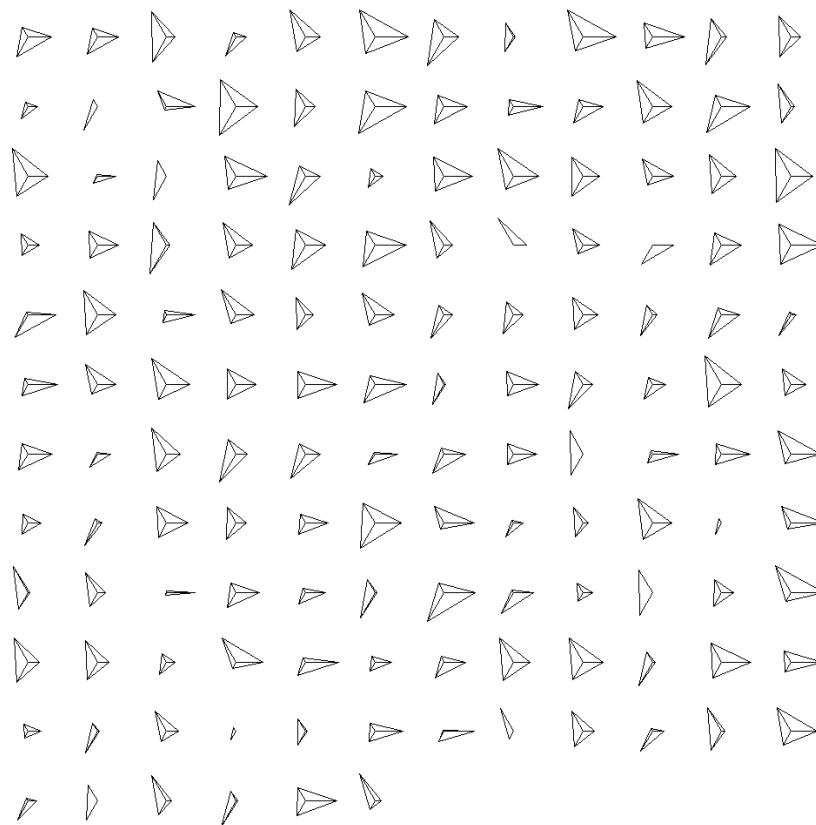
参数x 为一个多维数据矩阵或数据框，

每一行数据将生成一个星形；

full 为逻辑值，决定了是否使用整圆（或半圆）；

```
stars(Scores)
```

练习：给每一部分加上标签



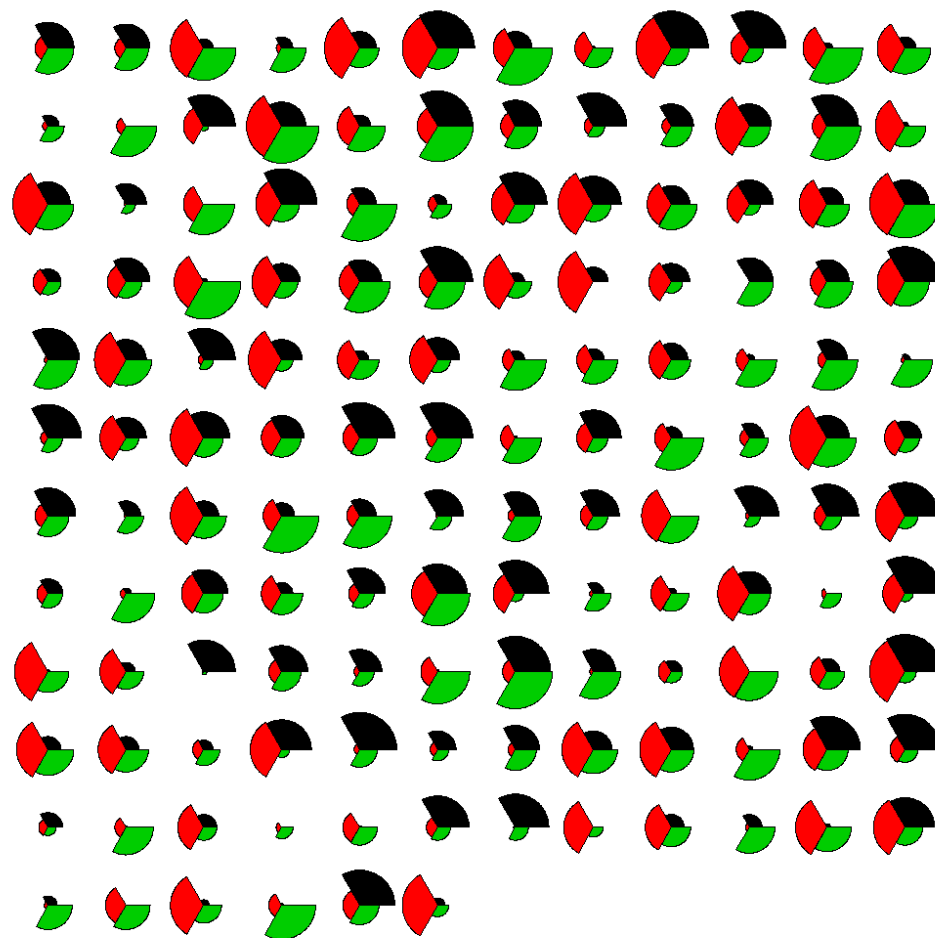
星相图

```
stars(Scores, full = TRUE, draw.segments = TRUE)
```

使用draw.segments=TRUE画扇形。

练习：给每一部分加上自定义的颜色

`col.segments=c("blue", "green", "red")`



茎叶图

`stem()`函数绘制茎叶图

`stem(x, scale=1, width=80, atom=1e-08)`

`x`是数据向量.

`scale`控制绘出茎叶图的长度.

`width`绘图的宽度.

`atom`是容差

选择`scale=2`, 即将10个个位数俞成两段,
0~4为一段, 5~9为另一段

```
stem(Scores$English)
```

				树 茎	树叶
88	80	75	97	9	7 6 5
57	69	74	96	9	4 4 3 2 0
86	79	68	56	8	8 6 5 5
95	85	79	74	8	3 2 1 1 0
54	68	94	67	7	9 9 8 7 7 6 5 5 5
73	85	78	53	7	4 4 3 1 1 0
83	65	77	48	6	9 8 8 7 5
94	71	93	82	5	7 6
77	71	64	81	5	4 3
76	70	92	64	4	8
89	81	75	63	4	
75					

表(1) 某校 98 级概率统计成绩

图(1) 成绩茎叶图表

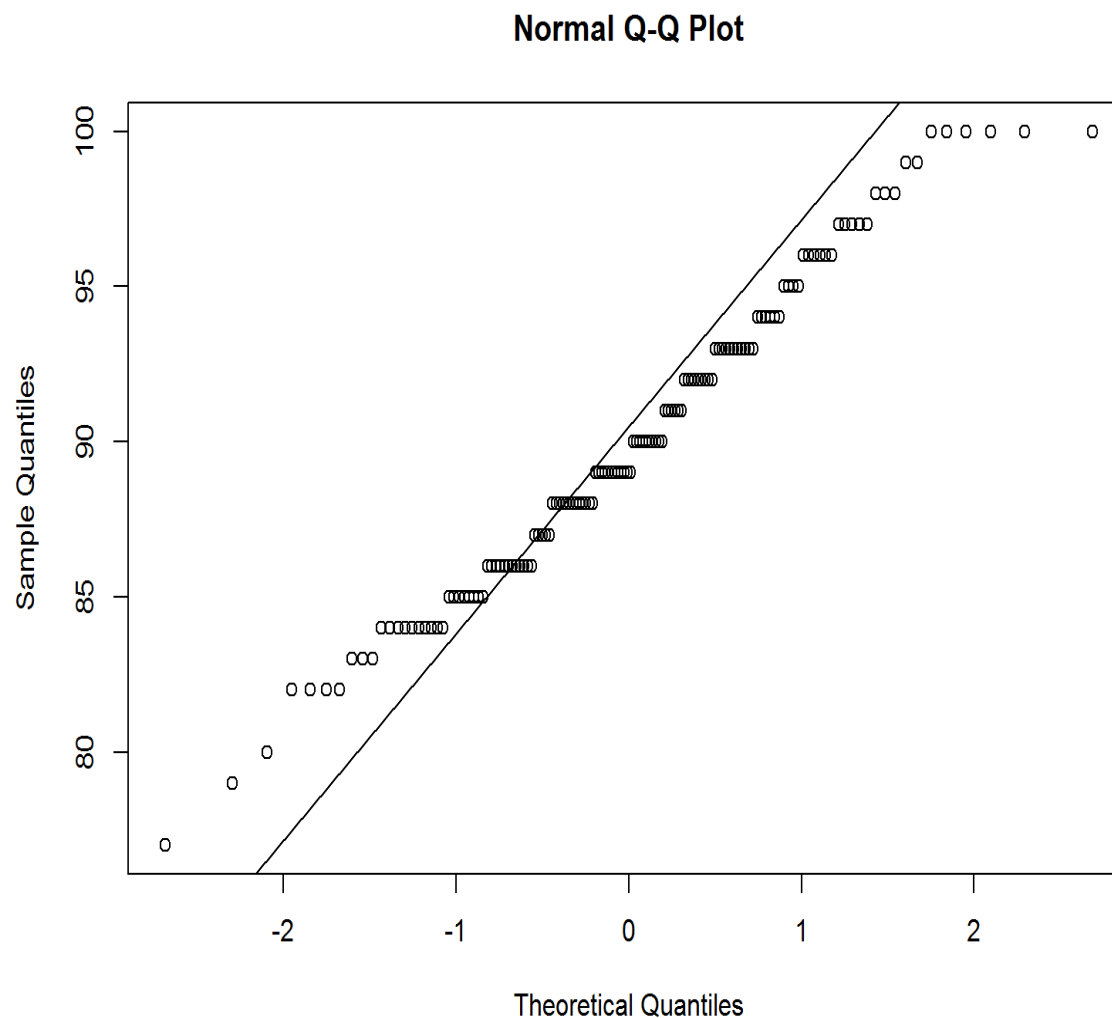
QQ图

QQ图：正态分位数图

- 用于判断数据是否呈正态分布
- 直线的斜率是标准差，截距是均值
- 点的散布越接近直线，则越接近正态分布

```
qqnorm(Scores$English)
```

```
qqline(Scores$English)
```



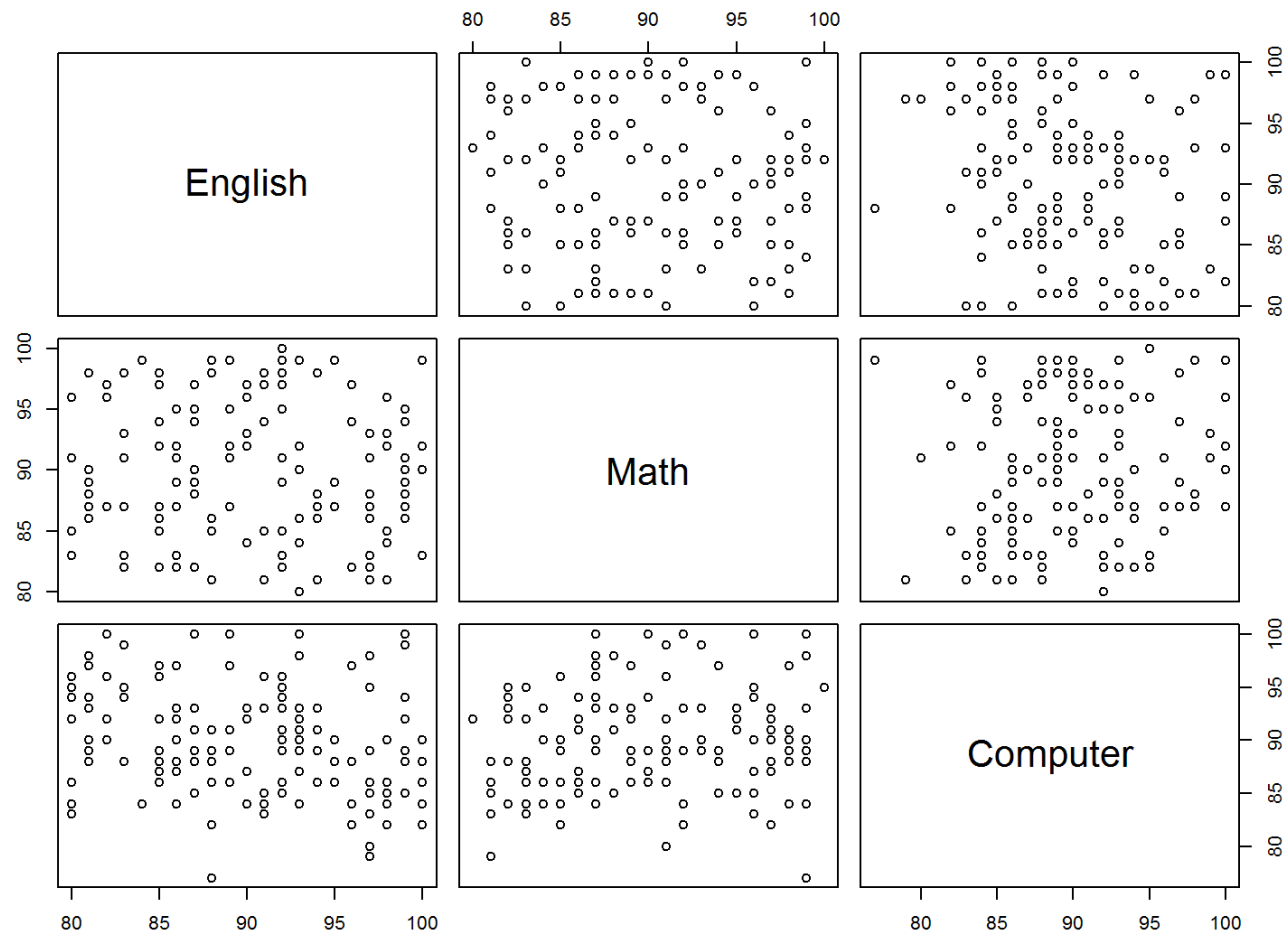
散点图矩阵

`pairs(x, ...)` Scatterplot Matrices

矩阵由x中的每列的列变量

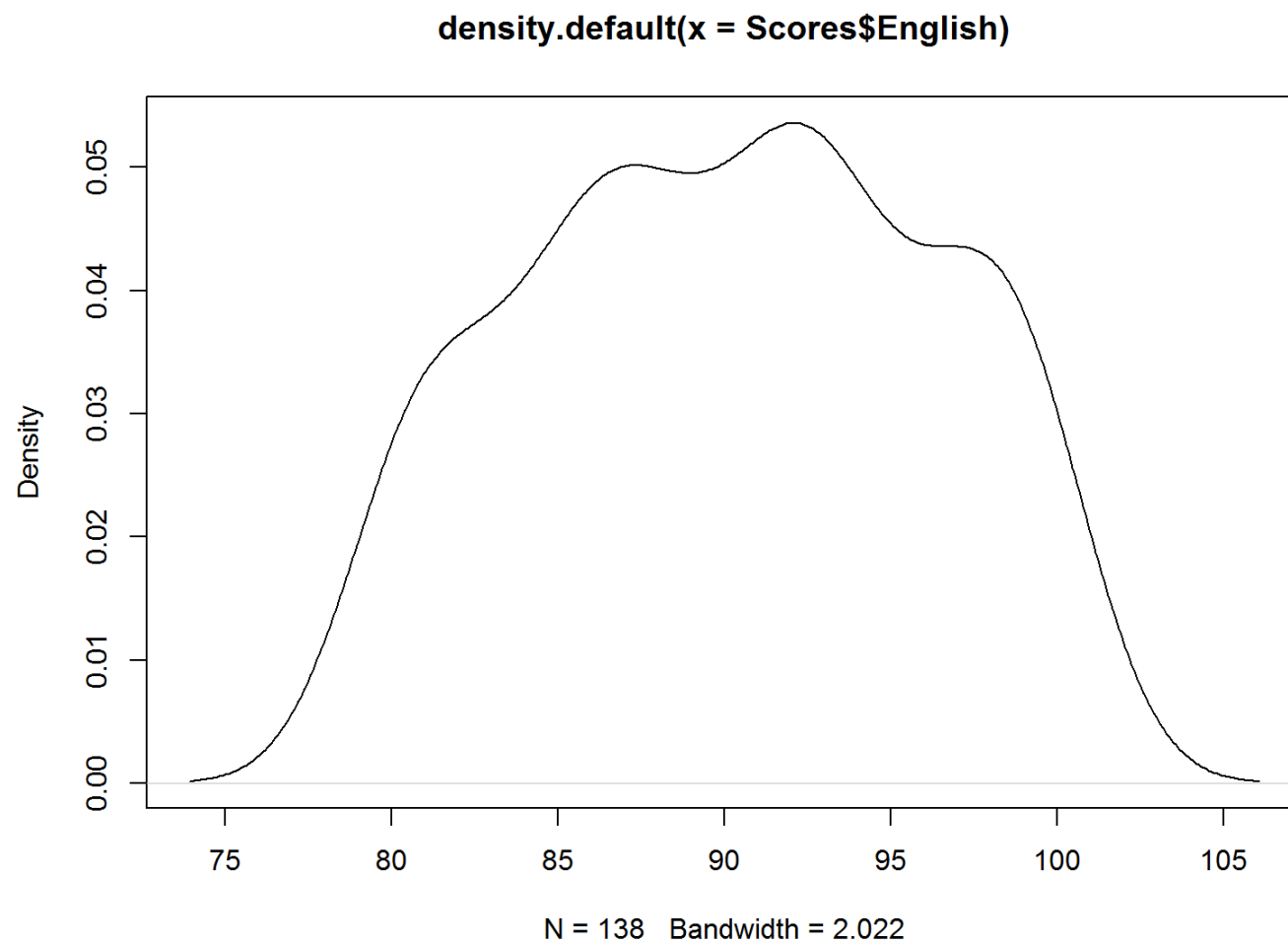
对其他各列列变量的散点图组成

`pairs(Scores)`



密度图

```
plot(density(Scores$English))
```

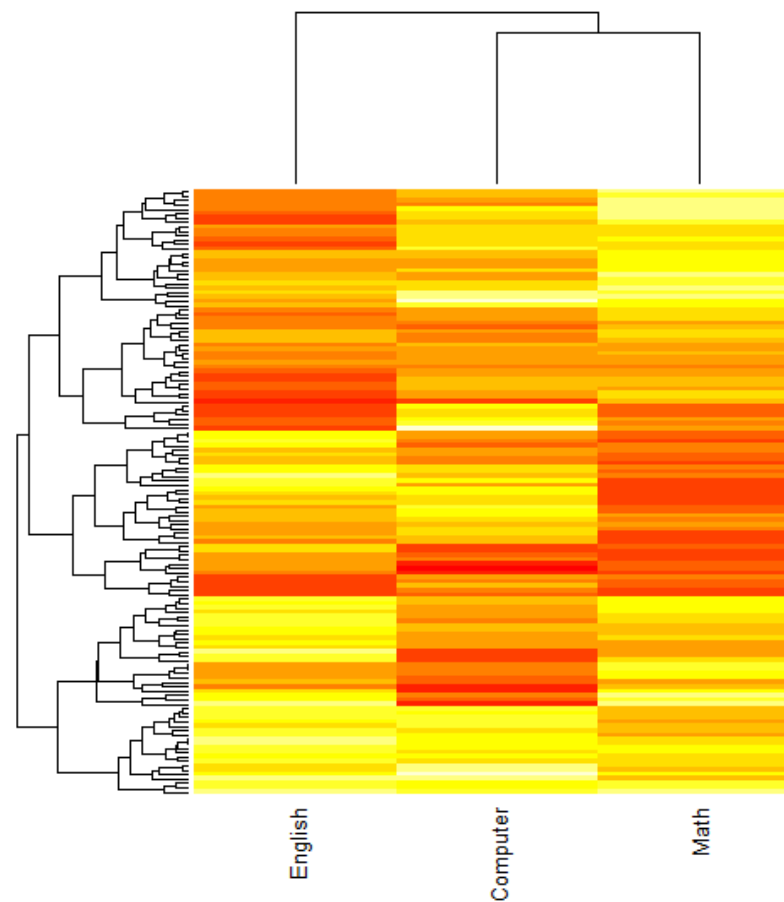


热力图

```
heatmap(data.matrix(Student_score[,4:6]), labRow =  
NA, cexCol =1, scale = "col")
```

scale

character indicating if the values should be centered and scaled in either the row direction or the column direction, or none.



数据中心化和标准化

- 1.数据的中心化

- 所谓数据的中心化是指数据集中的各项数据减去数据集的均值。

例如有数据集1, 2, 3, 6, 3，其均值为3,那么中心化之后的数据集为1-3,2-3,3-3,6-3,3-3,即：-2,-1,0,3,0

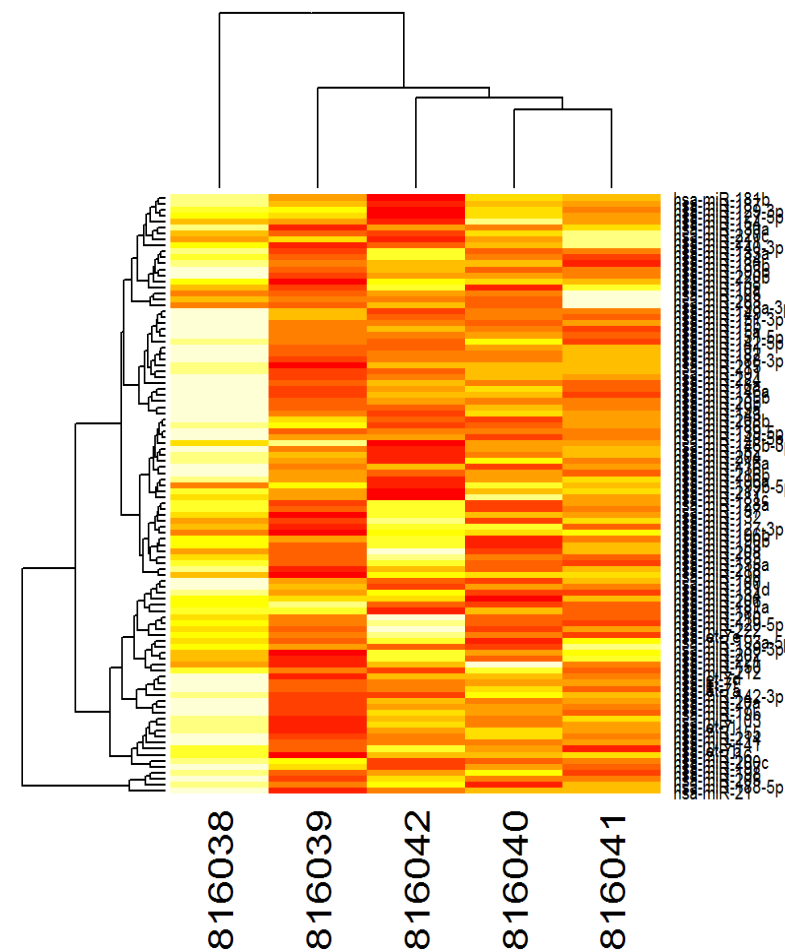
- 2.数据的标准化

所谓数据的标准化是指中心化之后的数据在除以数据集的标准差，即数据集中的各项数据减去数据集的均值再除以数据集的标准差。

热力图

```
chip<-read.table("chipplot.txt", header=T,  
sep="\t", row.names="miRNA")
```

```
heatmap(data.matrix(chip))
```



homework

将chipplot.txt芯片数据按下列要求完成绘图：

1. 绘制以样品GSM816041为x轴，GSM816042为y轴的散点图，点为蓝色实心原点，在坐标(12,10)处添加形状任意的红色点，添加必要的x轴，y轴，图例和标题名称；
2. 绘制has-miR-21在这5个样品的表达值柱形图，添加必要的x轴，y轴和标题名称，柱形需用lightyellow颜色表示；
3. 绘制样品GSM816038，GSM816040，GSM816041和GSM816042的箱线图，添加必要的x轴，y轴和标题名称箱线需用不同的颜色表示；

R Packages

- Many of R's most useful functions do not come preloaded when you start R, but reside in packages that can be installed on top of R
- When you tell R to install a package, it will automatically install any other packages that the first package depends on

What Are Repositories?

- A repository is a place where packages are located so you can install them from it. Three of the most popular repositories for R packages are:
- [CRAN](#): Comprehensive R Archive Network. The official repository, it is a network of ftp and web servers maintained by the R community around the world.
- [Bioconductor](#): this is a topic specific repository, intended for open source software for bioinformatics.
- [Github](#) : although this is not R specific, github is probably the most popular repository for open source projects.

Bioconductor

- Bioconductor is a suite of additional functions and some 200 packages dedicated to analysis, visualization, and management of genetic data

<https://www.bioconductor.org/>

The screenshot shows the Bioconductor website homepage. At the top, there is a teal navigation bar with the Bioconductor logo (a stylized 'B' with a DNA helix) and the text 'Bioconductor OPEN SOURCE SOFTWARE FOR BIOINFORMATICS'. To the right of the logo is a search bar labeled 'Search:'. The navigation bar also contains links for 'Home', 'Install', 'Help', 'Developers', and 'About'. Below the navigation bar, the main content area is divided into several sections. On the left, there is an 'About Bioconductor' section with a paragraph describing the project and a 'News' section with a list of recent updates. To the right, there are four boxes: 'Install' (with links for getting started), 'Learn' (with links for master tools), 'Use' (with links for creating solutions), and 'Develop' (with links for contributing). Each section contains a list of links to various resources and documentation.

About Bioconductor

Bioconductor provides tools for the analysis and comprehension of high-throughput genomic data. Bioconductor uses the R statistical programming language, and is open source and open development. It has two releases each year, [1473 software packages](#), and an active user community. Bioconductor is also available as an [AMI](#) (Amazon Machine Image) and a series of [Docker](#) images.

News

- Bioconductor [3.6](#) is available.
- Bioconductor [F1000 Research Channel](#) available.
- Orchestrating high-throughput genomic analysis with *Bioconductor* ([abstract](#)) and other [recent literature](#).
- View recent [course material](#).
- Use the [support site](#) to get help installing, learning and using Bioconductor.

Install »

Get started with *Bioconductor*

- [Install Bioconductor](#)
- [Explore packages](#)
- [Get support](#)
- [Latest newsletter](#)
- [Follow us on twitter](#)
- [Install R](#)

Learn »

Master *Bioconductor* tools

- [Courses](#)
- [Support site](#)
- [Package vignettes](#)
- [Literature citations](#)
- [Common work flows](#)
- [FAQ](#)
- [Community resources](#)
- [Videos](#)

Use »

Create bioinformatic solutions with *Bioconductor*

- [Software](#), [Annotation](#), and [Experiment](#) packages
- [Amazon Machine Image](#)
- [Latest release announcement](#)
- [Support site](#)

Develop »

Contribute to *Bioconductor*

- [Developer resources](#)
- [Use Bioc 'devel'](#)
- ['Devel' Software](#), [Annotation](#) and [Experiment](#) packages
- [Package guidelines](#)
- [New package submission](#)
- [Git source control](#)
- [Build reports](#)

Installing Packages From CRAN

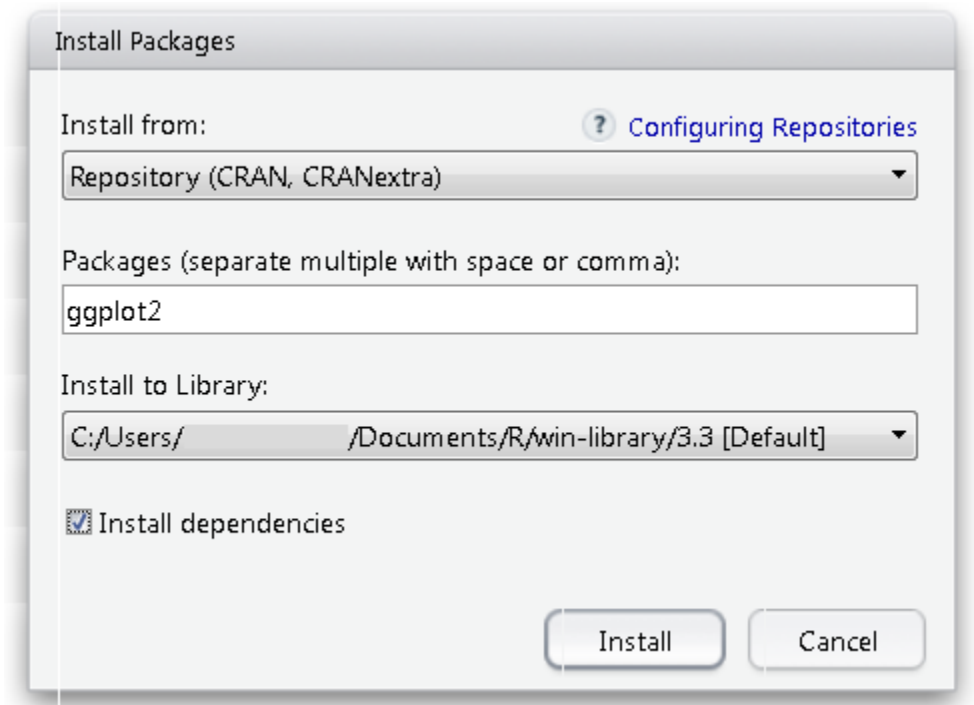
- **By Command Line**

```
install.packages("ggplot2")
```

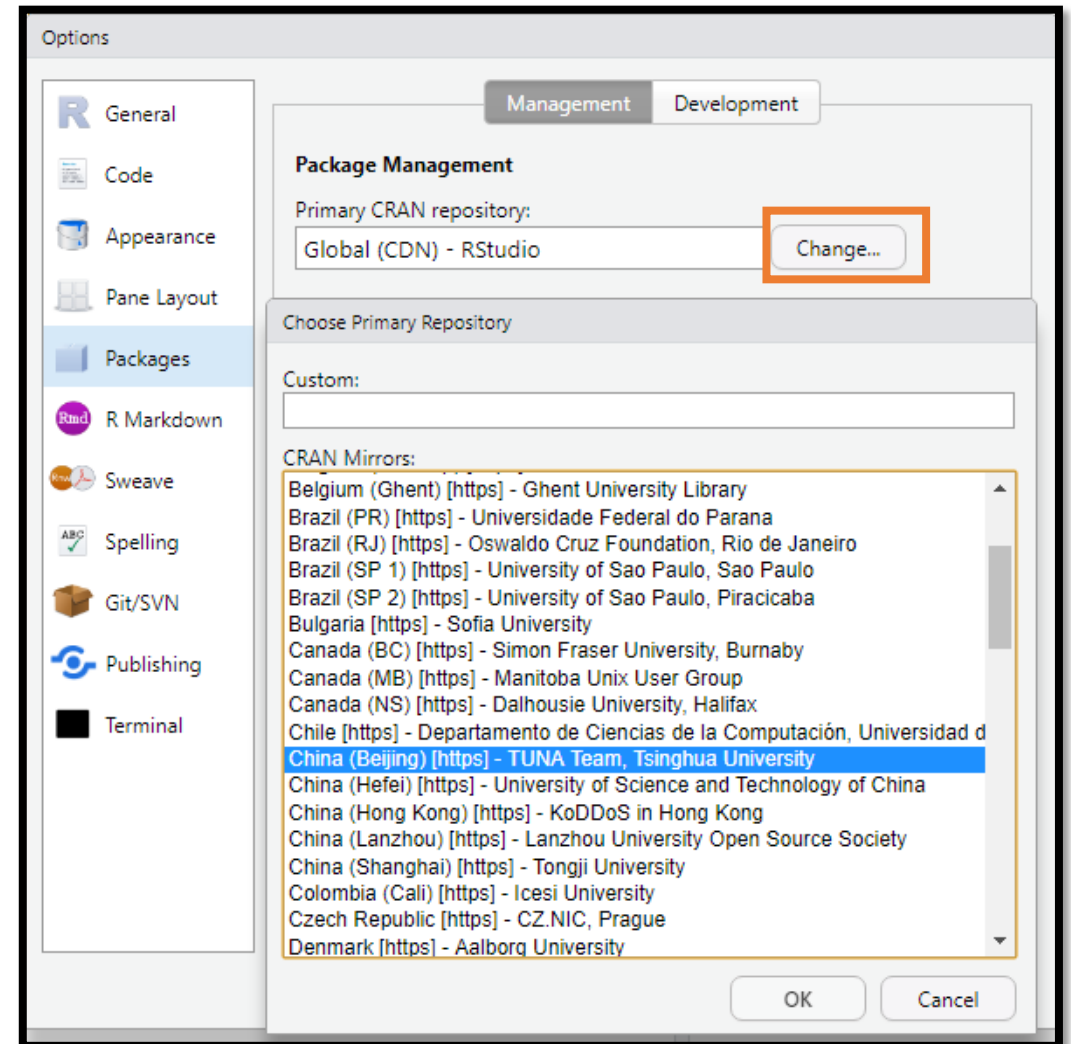
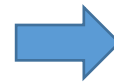
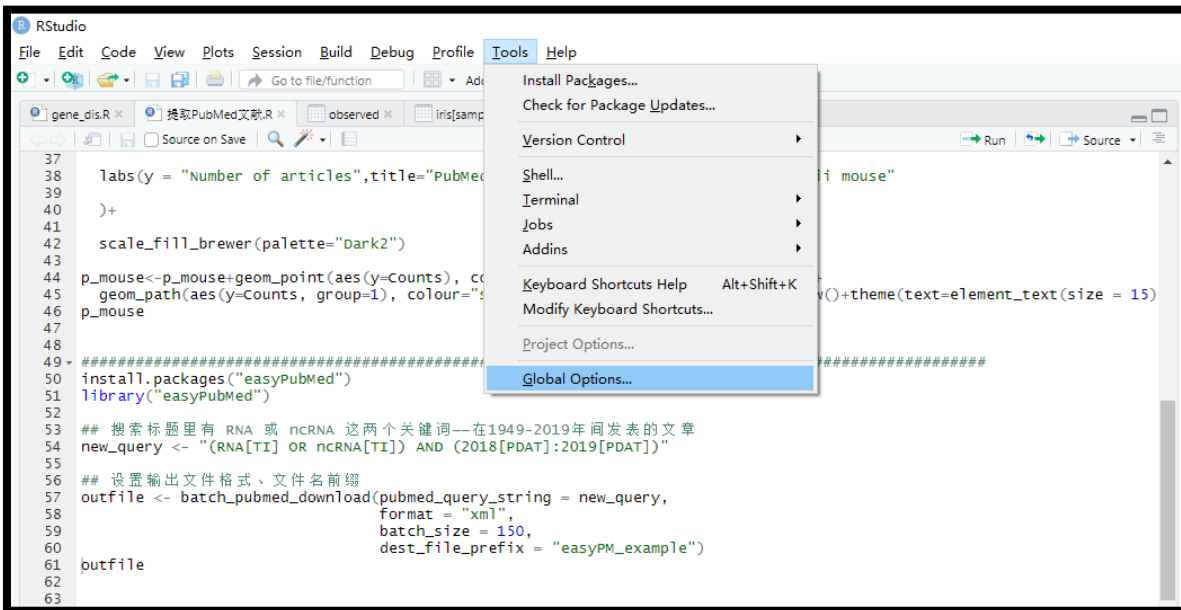
- **or**

- **By User Interface**

- In RStudio you will find it at Tools -> Install Package, and there you will get a pop-up window to type the package you want to install:



Change Install Mirror



```
options(repos=structure(c(CRAN="https://mirrors.tuna.tsinghua.edu.cn/CRAN/")))
```

How To Load Packages

- Installing a package doesn't immediately place its functions at your fingertips. It just places them on your computer.
- To use an R package, you next have to load it in your R session with the command:

```
library(ggplot2)
```

- To see an overview of what functions and data are contained in a package with the command:

```
help(package = "ggplot2")
```

Installing Bioconductor Packages

- In the case of Bioconductor, the standard way of installing a package is by first executing the following script:

```
source("https://bioconductor.org/biocLite.R")  
biocLite("Biostrings")
```

- You can run the command to test the package

```
library("Biostrings")  
myDNASeq <- DNAString("CTGATTT-GATGGTC-NAT")
```

HOMEWORK

- **Download and install ggplot2 and survminer**
- **Open RStudio.**
 1. **Make sure you are connected to the Internet.**
 2. **Run `install.packages(c("ggplot2", "survminer"))` at the command line**
 3. **Test with the command: `library("ggplot2"), library("survminer")`**



Thank You