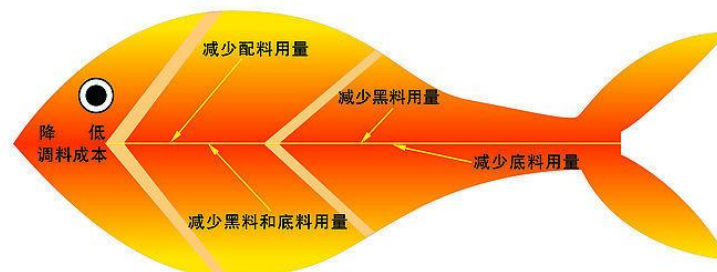


反鱼骨图



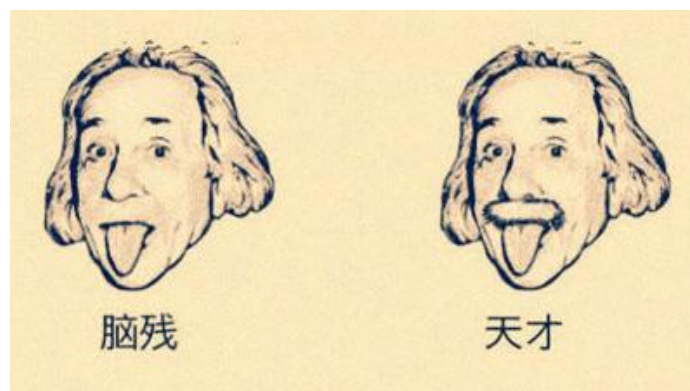
## 数据分析与R语言 第2周

2012.5.10

# 数据可视化的重要性

2006年资金预算收支执行情况表

单位: 万元														
月份	收 入							支 出						
	预算情况				实际情况			预算情况				实际情况		
	经营活动	投资活动	筹资活动	合 计	经营活动	投资活动	筹资活动	合 计	经营活动	投资活动	筹资活动	合 计	经营活动	投资活动
1月份	2700			2700	3610		0.17	3610.17	5476	2082	50	7608	4961	1175
2月份	3800			3800	2420		10.2	2430.2	3809	1244	50	5103	2887	108
3月份	4274			4274	9474		11	9485	4376	1496	50	6072	4529	6088
4月份	12396			12396	11121	88	2097	13286	5386	1514	50	7130	4246	1230
5月份	5311	132		5463	5784	98	94	5976	5841	2431	440	8712	4783	792
6月份	3801			3801	1217	15	103	1335	4332	2904	87	7323	4067	1903
7月份	5951			5951	4427	65	3593	8085	4085	2591	331	7007	5218	2187
8月份	5388			5388	1883		2021	3904	3375	3830	2120	9325	3133	3472
9月份	2830			2830	2459	2	914	3375	3955	2505	93	6933	2800	1469
10月份	3250			3250	2853		49	2904	4283	2209	40	6534	3526	1591
11月份	2870		700	4370	647		134	781	5873	6036	340	12449	810	3861
12月份	4105		2110	6255	7723		2576	10299	7631	3551	88	11270	7063	1838
合 计	57676	132	2850	60678	53620	248	11602.37	65470.37	58774	32793	3939	95506	48027	25714



2012.5.10

- 模拟产生统计专业同学的名单（学号区分），记录数学分析，线性代数，概率统计三科成绩，然后进行一些统计分析

```
> num=seq(10378001,10378100)
> num
 [1] 10378001 10378002 10378003 10378004 10378005 10378006 10378007 10378008
 [9] 10378009 10378010 10378011 10378012 10378013 10378014 10378015 10378016
[17] 10378017 10378018 10378019 10378020 10378021 10378022 10378023 10378024
[25] 10378025 10378026 10378027 10378028 10378029 10378030 10378031 10378032
[33] 10378033 10378034 10378035 10378036 10378037 10378038 10378039 10378040
[41] 10378041 10378042 10378043 10378044 10378045 10378046 10378047 10378048
[49] 10378049 10378050 10378051 10378052 10378053 10378054 10378055 10378056
[57] 10378057 10378058 10378059 10378060 10378061 10378062 10378063 10378064
[65] 10378065 10378066 10378067 10378068 10378069 10378070 10378071 10378072
[73] 10378073 10378074 10378075 10378076 10378077 10378078 10378079 10378080
[81] 10378081 10378082 10378083 10378084 10378085 10378086 10378087 10378088
[89] 10378089 10378090 10378091 10378092 10378093 10378094 10378095 10378096
[97] 10378097 10378098 10378099 10378100
```

## ■ 用runif和rnorm 均匀分布

```
> x1=round(runif(100,min=80,max=100))
```

```
> x1
```

```
[1] 95 97 88 82 95 85 81 81 91 99 84 95 89 92 89 93 96 87  
[19] 90 81 94 94 88 91 90 90 97 92 91 97 96 93 80 93 86 89  
[37] 81 87 86 85 89 92 84 91 92 86 91 85 96 96 83 99 80 97  
[55] 88 98 85 97 94 99 82 89 96 85 80 88 93 97 97 91 100 89  
[73] 98 86 97 88 88 95 99 83 96 85 95 88 88 91 90 85 84 86  
[91] 94 87 99 93 89 87 95 89 84 81
```

```
> |
```

```
> x2=round(rnorm(100,mean=80,sd=7))
```

正态分布

```
> x2
```

```
[1] 89 73 76 70 64 74 95 86 65 83 80 83 71 72 83 79 91 70 75 84 72 94 74 81  
[25] 81 74 85 96 69 84 73 93 72 76 73 81 76 92 73 81 88 80 87 81 81 66 76 85  
[49] 97 77 83 77 82 86 76 69 83 83 77 79 84 90 82 81 81 79 76 90 80 83 88 93  
[73] 75 83 89 93 69 97 85 85 93 73 73 79 75 64 81 81 55 63 81 80 84 78 88 75  
[97] 75 78 78 87
```

```
> |
```

```
> x3=round(rnorm(100,mean=83,sd=18))
> x3
 [1] 62 83 73 71 92 53 59 89 90 98 123 75 107 108 69 73 110 61
[19] 88 83 76 96 81 56 41 70 64 78 80 61 94 108 77 91 83 93
[37] 66 64 56 87 97 92 99 82 45 93 86 77 82 75 69 94 75 98
[55] 75 65 63 75 88 79 80 104 88 94 92 77 63 97 87 85 89 58
[73] 83 84 93 64 109 115 104 87 78 58 74 67 120 66 64 80 72 88
[91] 86 97 97 114 89 41 104 76 70 81
> x3[which(x3>100)]=100
> x3
 [1] 62 83 73 71 92 53 59 89 90 98 100 75 100 100 69 73 100 61
[19] 88 83 76 96 81 56 41 70 64 78 80 61 94 100 77 91 83 93
[37] 66 64 56 87 97 92 99 82 45 93 86 77 82 75 69 94 75 98
[55] 75 65 63 75 88 79 80 100 88 94 92 77 63 97 87 85 89 58
[73] 83 84 93 64 100 100 100 87 78 58 74 67 100 66 64 80 72 88
[91] 86 97 97 100 89 41 100 76 70 81
> |
```

# 合成数据框并保存到硬盘

- data.frame()
- write.table



```
> x=data.frame(num,x1,x2,x3)
> x
```

	num	x1	x2	x3
1	10378001	95	89	62
2	10378002	97	73	83
3	10378003	88	76	73
4	10378004	82	70	71
5	10378005	95	64	92
6	10378006	85	74	53
7	10378007	81	95	59
8	10378008	81	86	89
9	10378009	91	65	90
10	10378010	99	83	98
11	10378011	84	80	100
12	10378012	95	83	75

```
> write.table(x,file="d:\\mark.txt",col.names=F,row.names=F,sep=" ")
> |
```

## ■ 函数mean(), colMeans(), apply()

```
> mean(x)
      num      x1      x2      x3
10378050.50  90.19  80.00  80.47
警告信息:
mean(<data.frame>) is deprecated.
Use colMeans() or sapply(*, mean) instead.
> colMeans(x)
      num      x1      x2      x3
10378050.50  90.19  80.00  80.47
> colMeans(x)[c("x1", "x2", "x3")]
      x1      x2      x3
90.19 80.00 80.47
> apply(x, 2, mean)
      num      x1      x2      x3
10378050.50  90.19  80.00  80.47
> |
```

# 求各科最高最低分

## ■ 函数max( ),min( ),apply( )

2 是指,在列的方向上进行计算

```
> apply(x,2,max)
```

num	x1	x2	x3
10378100	100	97	100

```
> apply(x,2,min)
```

num	x1	x2	x3
10378001	80	55	41

```
> |
```



## 求出每人总分

这个l是指apply函数中,按照行处理,是1,按照列处理,是2

```
> apply(x[c("x1","x2","x3")],1,sum)  这里是按行的方向做,相当于对每个人求总分
[1] 246 253 237 223 251 212 235 256 246 280 264 253 260 264 241 245 287 218
[19] 253 248 242 284 243 228 212 234 246 266 240 242 263 286 229 260 242 263
[37] 223 243 215 253 274 264 270 254 218 245 253 247 275 248 235 270 237 281
[55] 239 232 231 255 259 257 246 279 266 260 253 244 232 284 264 259 277 240
[73] 256 253 279 245 257 292 284 255 267 216 242 234 263 221 235 246 211 237
[91] 261 264 280 271 266 203 270 243 232 249
```

## 总分最高的同学

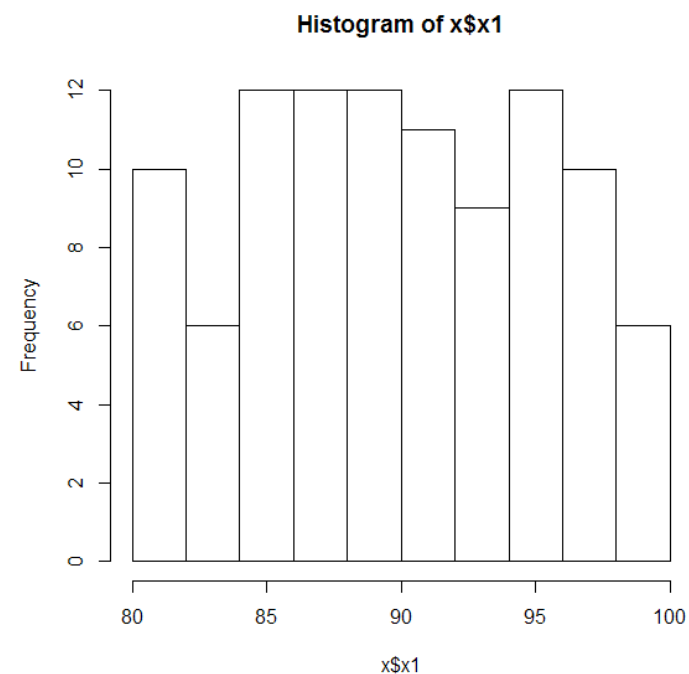
```
> apply(x[c("x1", "x2", "x3")], 1, sum)
 [1] 246 253 237 223 251 212 235 256 246 280 264 253 260 264 241 245 287 218
[19] 253 248 242 284 243 228 212 234 246 266 240 242 263 286 229 260 242 263
[37] 223 243 215 253 274 264 270 254 218 245 253 247 275 248 235 270 237 281
[55] 239 232 231 255 259 257 246 279 266 260 253 244 232 284 264 259 277 240
[73] 256 253 279 245 257 292 284 255 267 216 242 234 263 221 235 246 211 237
[91] 261 264 280 271 266 203 270 243 232 249
> which.max(apply(x[c("x1", "x2", "x3")], 1, sum))
[1] 78 求的是下标
> x$num[which.max(apply(x[c("x1", "x2", "x3")], 1, sum))]
[1] 10378078 把下标重新代回去,就是学号
> |
```

# 对x1进行直方图分析

## ■ 绘制直方图函数hist( )

x数据框里面的x1这个列  
也就是成绩

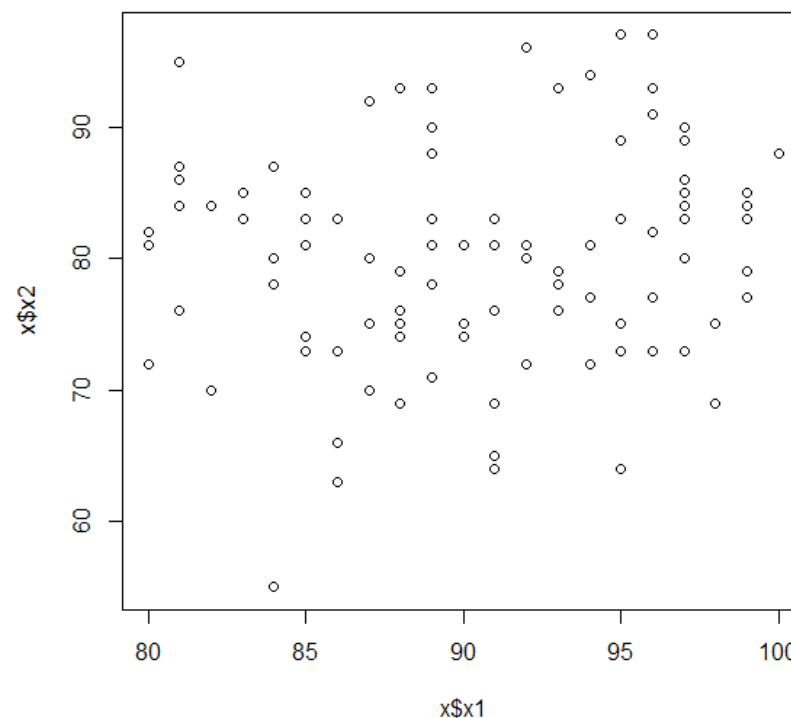
```
> hist(x$x1)  
> |
```



# 探索各科成绩的关联关系

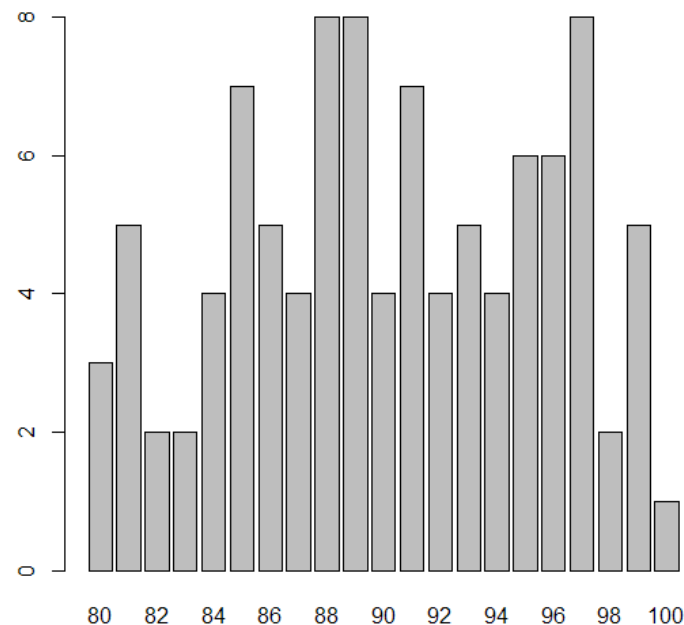
## ■ 散点图绘制函数plot()

```
> plot(x1,x2)  
> plot(x$x1,x$x2)  
> |
```



## ■ 列联函数table( )，柱状图绘制函数barplot( )

跟直方图其实差不多  
只是使用的函数和使用方法不同



> table(x\$x1) 统计每种分数分别有多少人

```
80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98
 3   5   2   2   4   7   5   4   8   8   4   7   4   5   4   6   6   8   2
99 100
 5   1
```

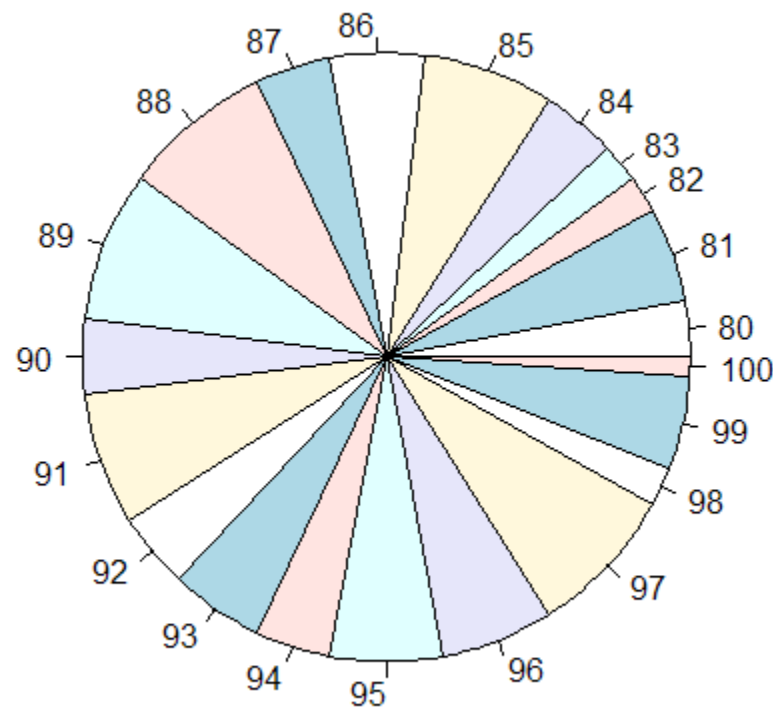
```
> barplot(table(x$x1))
~ |
```

2012.5.10

## ■ 饼图绘制函数pie( )

```
>  
> pie(table(x$x1))  
> |
```

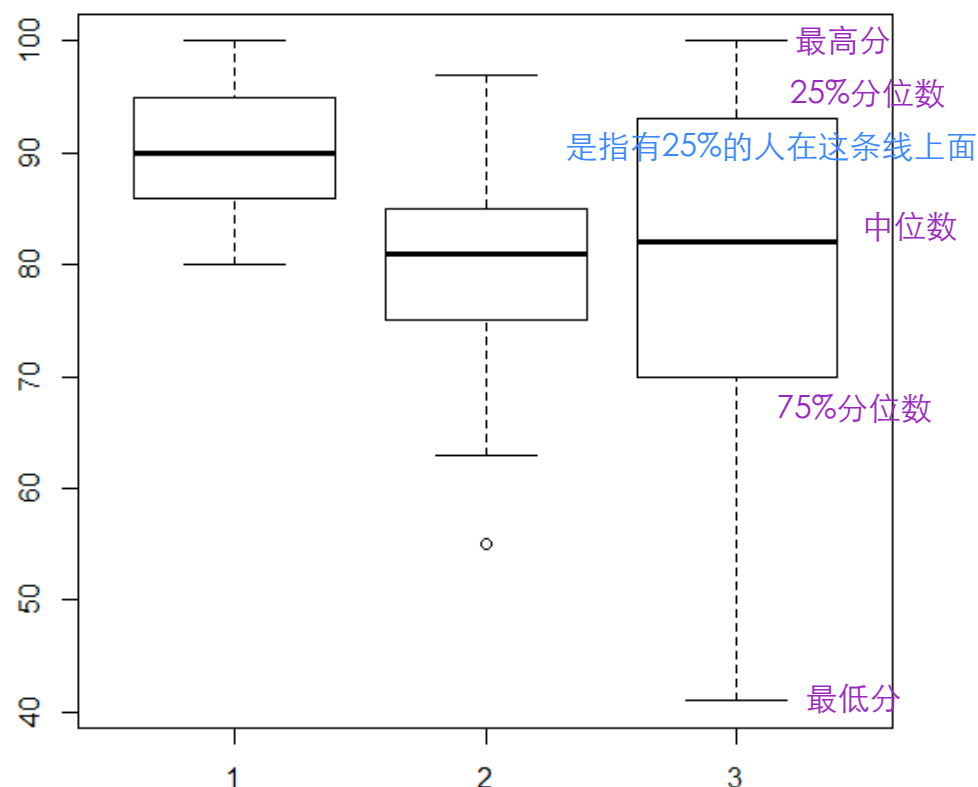
先用列联行数统计出来  
再把参数给pie  
才可以画图



# 箱尾图

- 箱子的上下横线为样本的25%和75%分位数
- 箱子中间的横线为样本的中位数
- 上下延伸的直线称为尾线，尾线的尽头为最高值和最低值
- 异常值

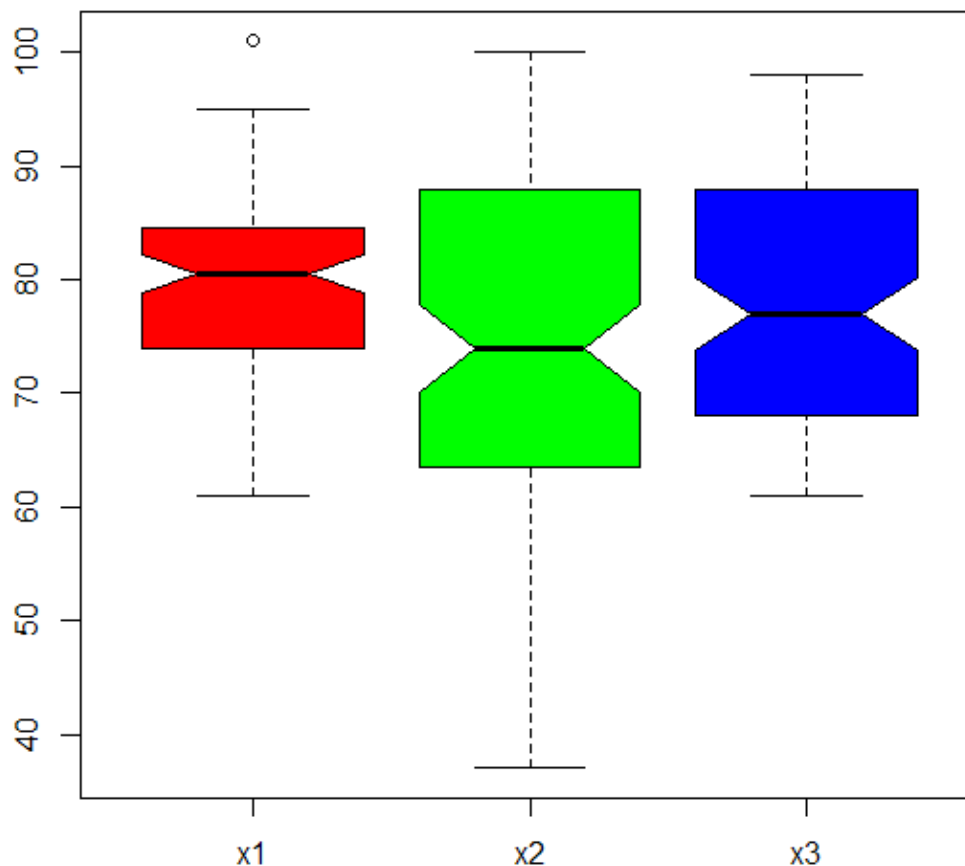
```
> boxplot(x$x1, x$x2, x$x3)  
> |
```



```
boxplot(x[2:4],col=c("red","green","blue"),notch=T)
```

指定了颜色和缺口  
颜色也是向量,用col控制

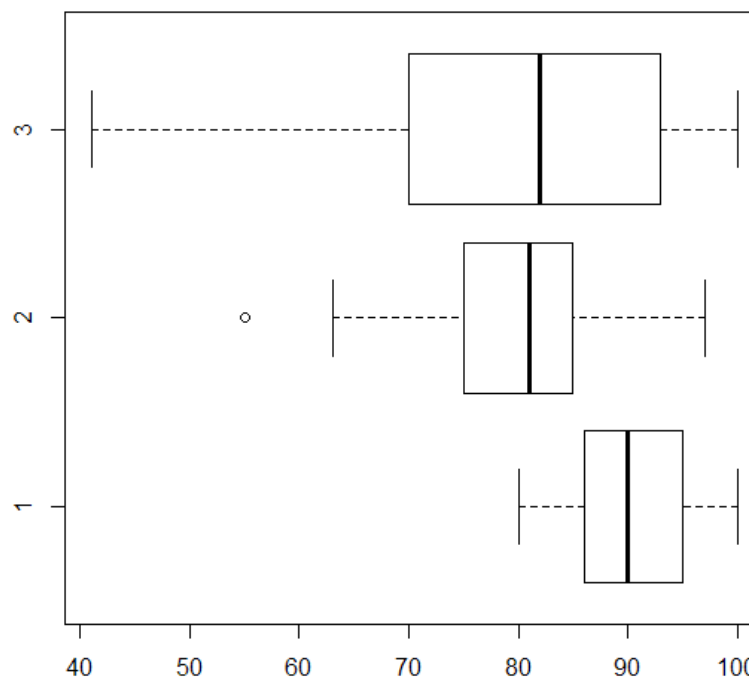
notch来控制缺口  
可以把中位线所在的位置看得更清楚  
T是指true





## ■ 水平放置的箱尾图

```
> boxplot(x$x1,x$x2,x$x3,horizontal=T)  
> |
```

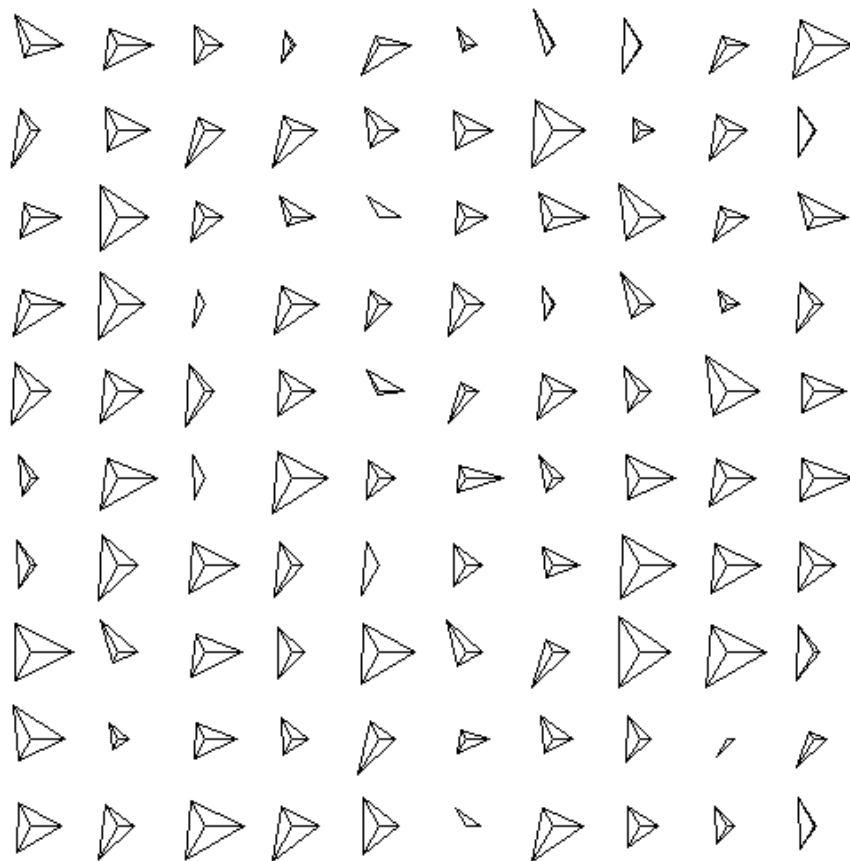


- 每个观测单位的数值表示为一个图形
- 每个图的每个角表示一个变量,字符串类型会标注在图的下方
- 角线的长度表达值的大小

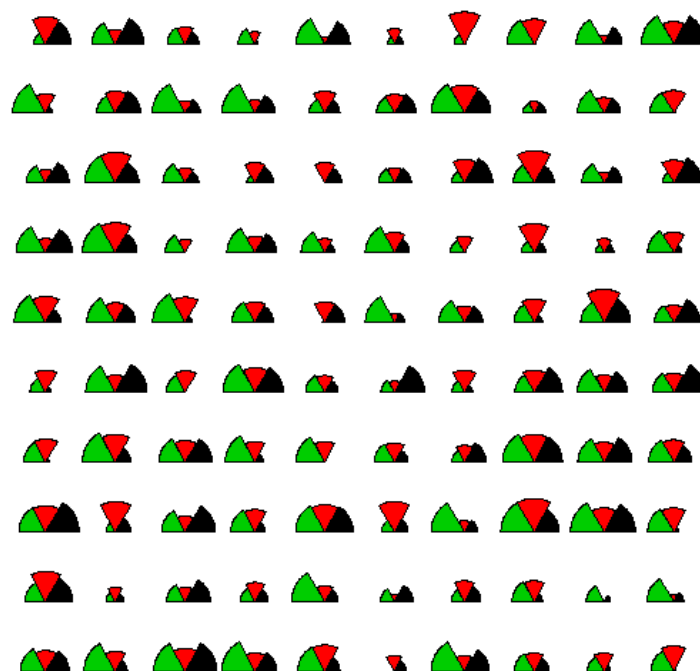
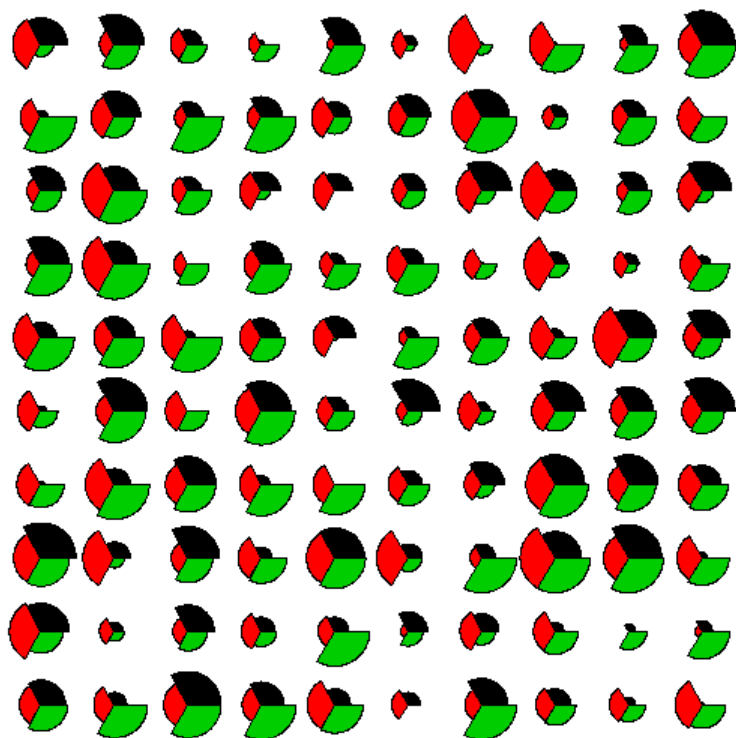
可以找三好学生,可以找偏科学生

```
> stars(x[c("x1", "x2", "x3")])
```

```
> |  
  c("x1", "x2", "x3")  
  这里是指取数据框里面的哪几个列  
  取了几个列就有几个量
```



```
> stars(x[c("x1", "x2", "x3")], full=T, draw.segment=T)
```



两幅图的区别是,是否full=true还是false

Full= true: 是指整个圆  
full= false:是指半个圆

```
> stars(x[c("x1", "x2", "x3")], full=F, draw.segment=T)
```

## ■ 安装aplpack包

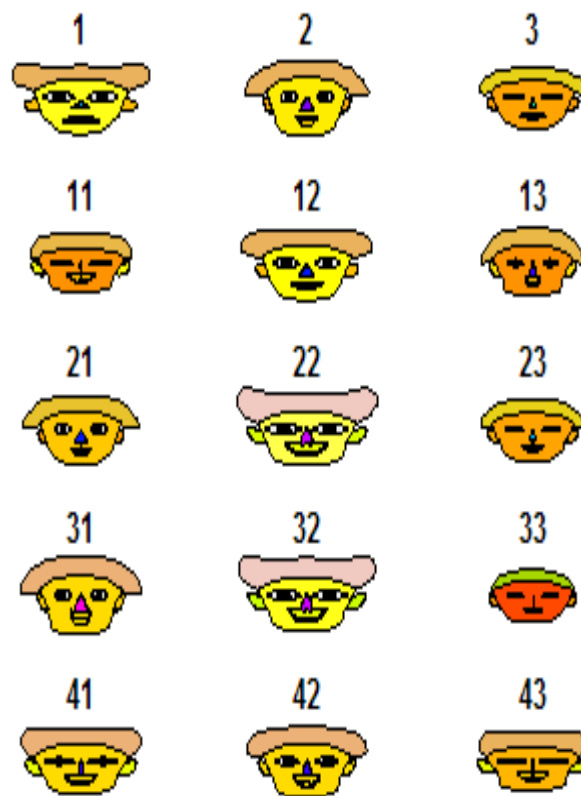
R基本软件里面原来没有,要靠安装额外的包  
如何安装这个包之前上过,  
在安装软件里面安装  
就是在library里面

```
> faces(x[c("x1", "x2", "x3")])
```



- 用五官的宽度和高度来描绘数值
- 人对脸谱高度敏感和强记忆
- 适合较少样本的情况

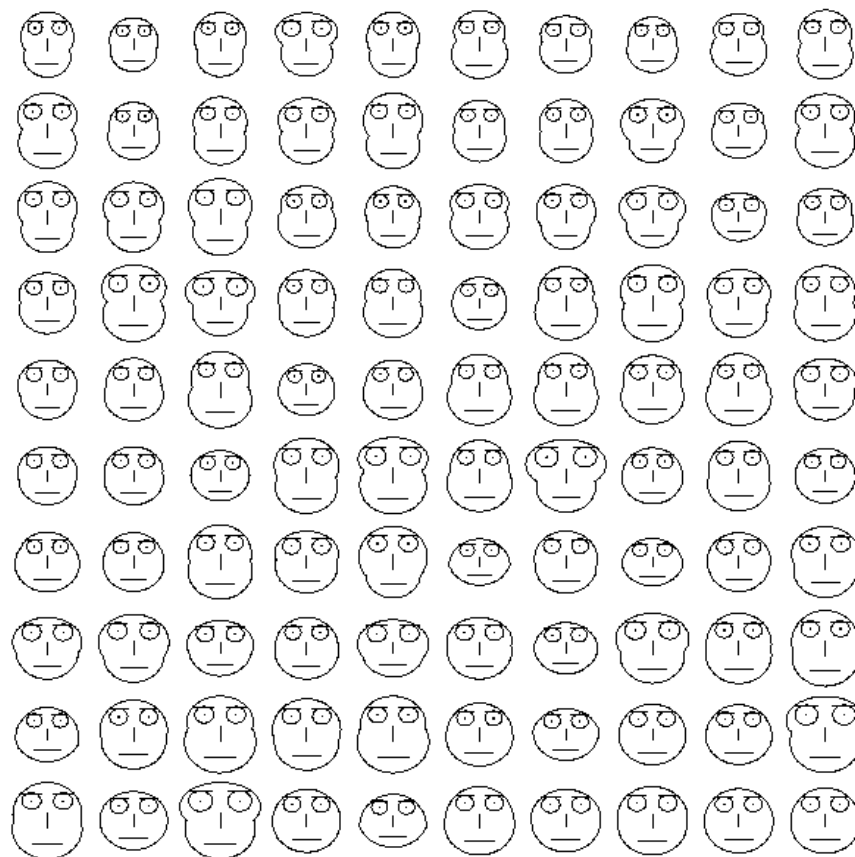
人对人脸比较敏感,脸谱图可以给人留下很深刻的印象



### ■ 安装TeachingDemos包

```
> library(TeachingDemos)
```

```
> faces2(x)
```



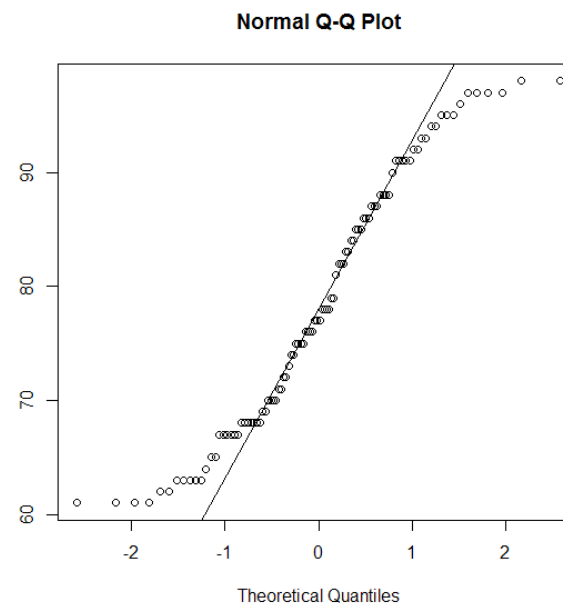
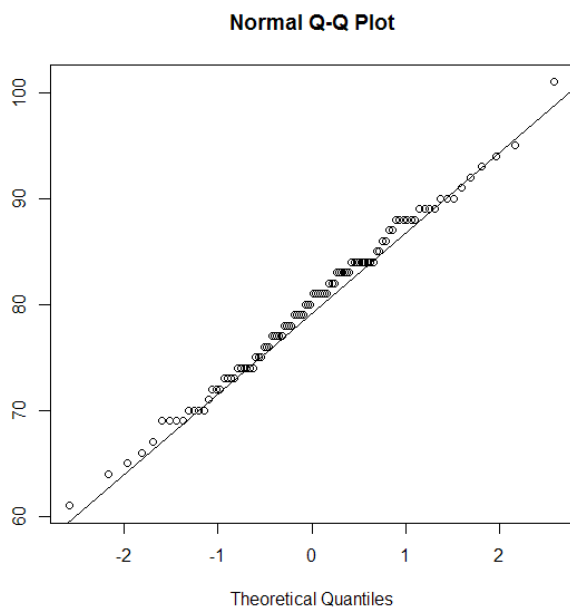
```
> stem(x$x1)
```

```
The decimal point is 1 digit(s) to the right of the |
```

```
6 | 14      茎叶图: 虽然看起来很简陋,但是是信息高度密集的统计图  
6 | 5679999 6开头的有两个,就是61分和64分  
7 | 000012223333444444  
7 | 55566677777888899999  
8 | 0001111111222333333444444444  
8 | 5566778888889999  
9 | 0001234  
9 | 5  
10 | 1
```

- 可用于判断是否正态分布
- 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(x1)  
> qqline(x1)  
> qqnorm(x3)  
> qqline(x3)
```



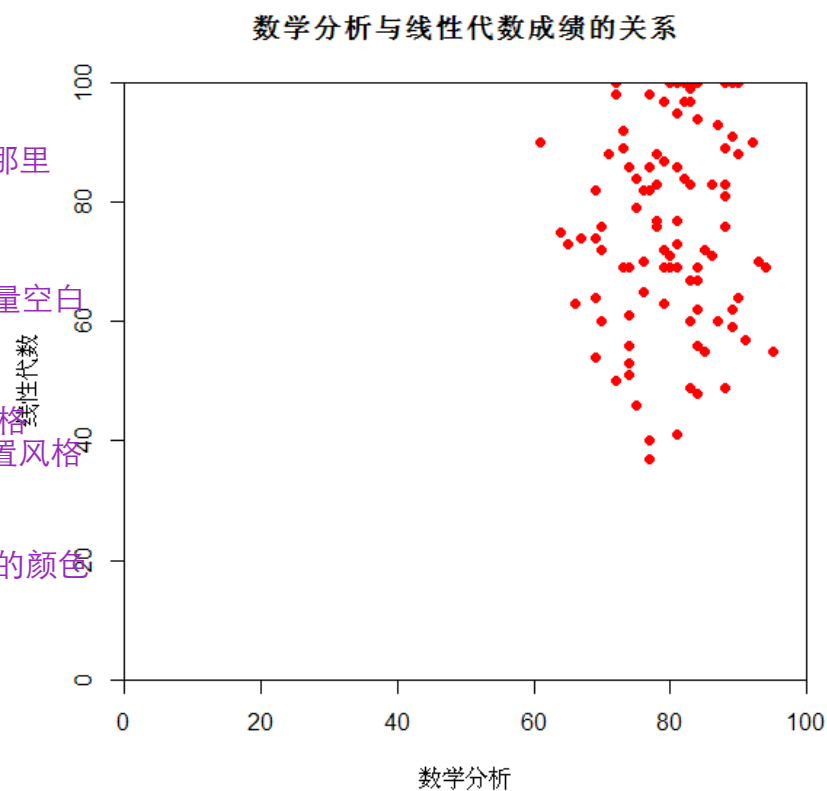
2012.5.10



## ■ 散点图的进一步设置

plot函数后面还可以加很多参数

```
plot(x$x1,x$x2,  
main="数学分析与线性代数成绩的关系",  
xlab="数学分析", 控制x轴的注解,否则就会放个x$x1在那里  
ylab="线性代数",  
xlim=c(0,100), x轴的坐标范围是0到100,所以左下角才大量空白  
ylim=c(0,100),  
xaxs="i", #Set x axis style as internal 设置x轴的显示风格  
yaxs="i", #Set y axis style as internal 这里的i指的是内置风格  
col="red", #Set the color of plotting symbol to red 点的颜色  
pch=19) #Set the plotting symbol to filled dots  
图形的形式,这里是19,是圆盘
```



## ■ 连线图

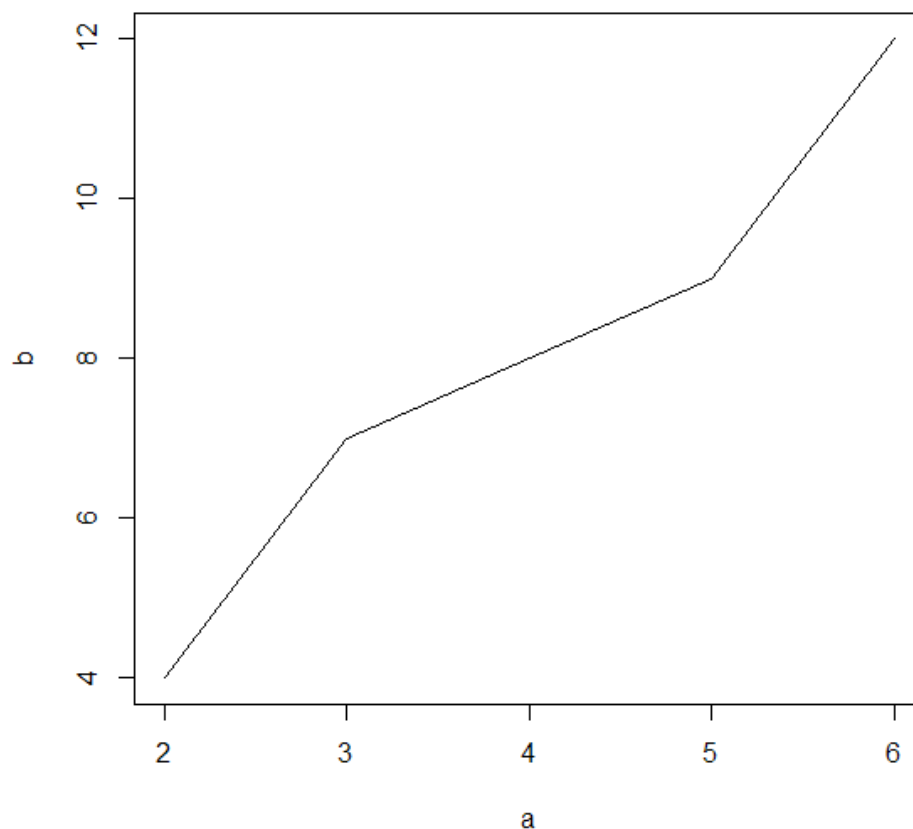
```
a=c(2,3,4,5,6)
```

```
b=c(4,7,8,9,12)
```

```
plot(a,b,type="l")
```

把(2,4)和(3,7),(4,7)这些点连起来

Type= "l" 是指连线



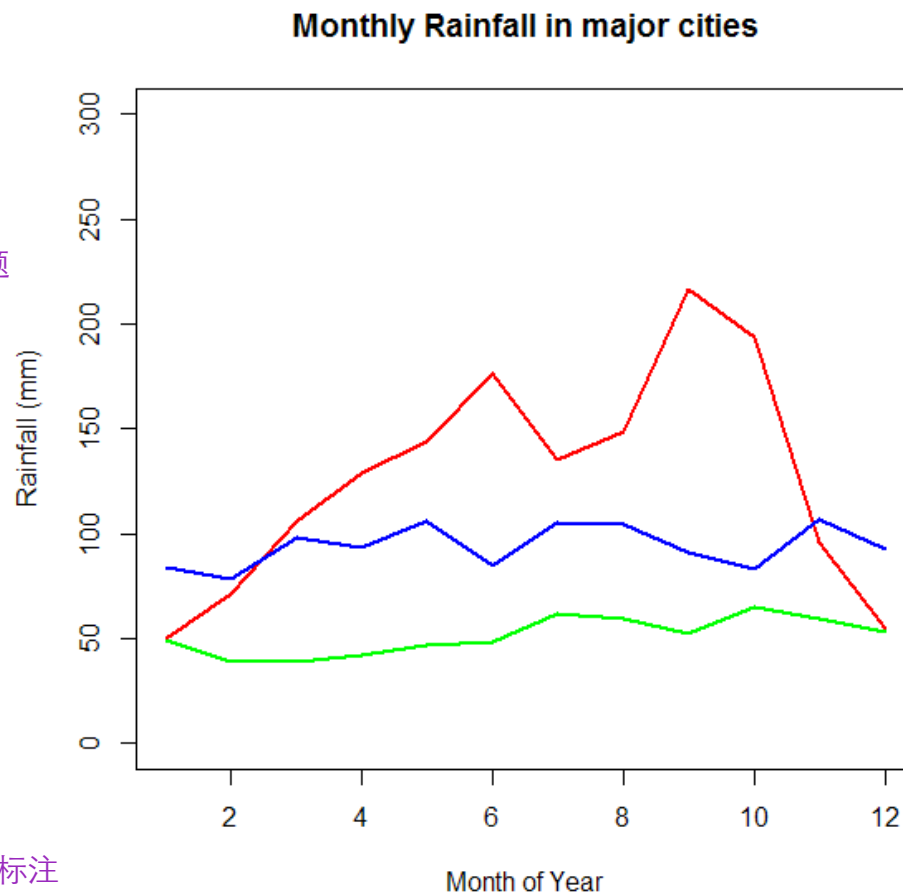
## ■ 多条曲线的效果

`rain$Tokyo` 是指 `rain` 这个数据框里面 `Tokyo` 这个列

```
plot(rain$Tokyo,type="l",col="red",  
ylim=c(0,300),  
main="Monthly Rainfall in major cities", 设置图表标题  
xlab="Month of Year",  
ylab="Rainfall (mm)",  
lwd=2)  lwd是line width,如果不说的话,默认为1  
lines(rain$NewYork,type="l",col="blue",lwd=2)  
lines(rain$London,type="l",col="green",lwd=2)  
lines(rain$Berlin,type="l",col="orange",lwd=2)
```

可以在原来的基础上继续进行画图

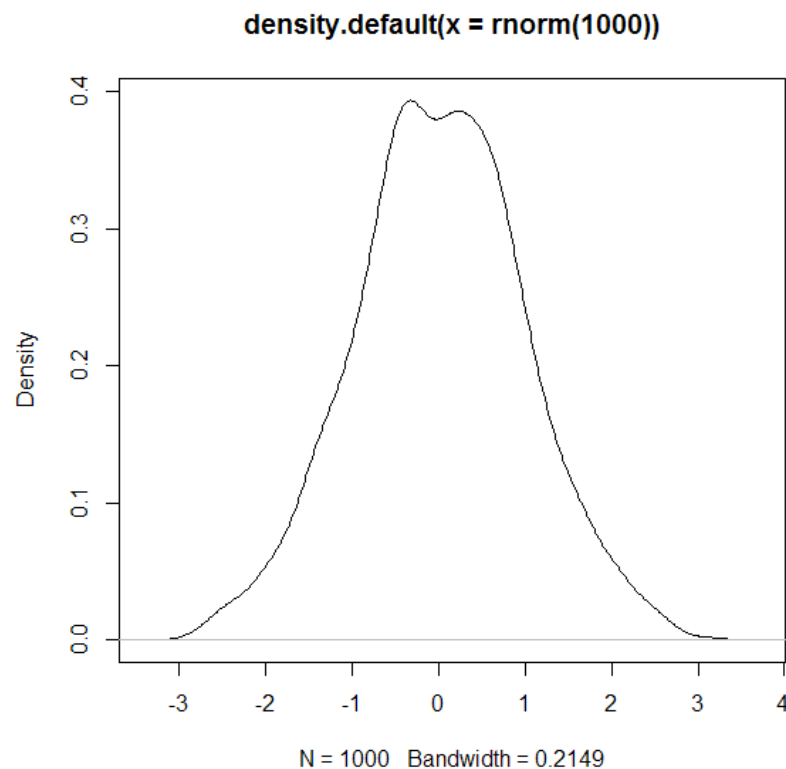
`plot` 是高水平画图命令,可以直接画出一张图  
`lines` 是低水平画图命令,必须要用 `plot` 先画出来,才能在上面标注



分布函数的密度图

## ■ 函数density( )

```
plot(density(rnorm(1000)))
```



## ■ 函数data()列出内置数据

不需要输入数据,只要安装好了R软件,可以找到内置的这些数据集

输入data()就会跳出来一个窗口,如果对其中的mtcars的数据有兴趣,就打入mtcars

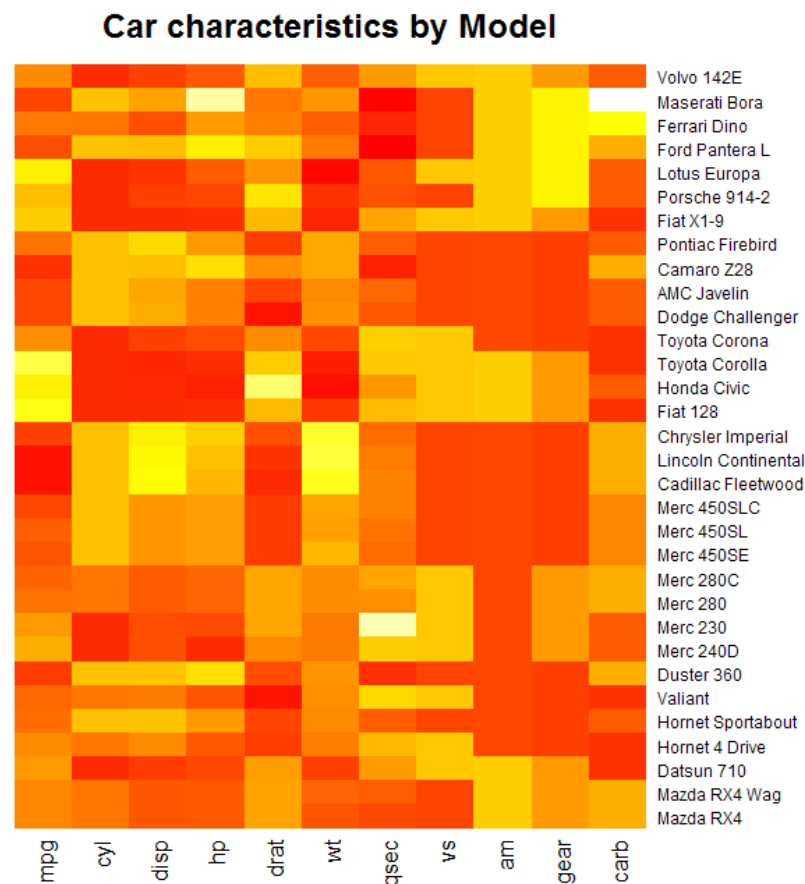
```
> mtcars
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4

## ■ 利用内置的mtcars数据集绘制

必须用数据框转换为矩阵,才能用heatmap

```
heatmap(as.matrix(mtcars),  
Rowv=NA,  
Colv=NA,  
col = heat.colors(256),  
scale="column",  
margins=c(2,8),  
main = "Car characteristics by  
Model")
```



# Iris ( 鸢尾花 ) 数据集

- Sepal 花萼
- Petal 花瓣
- Species 种属

打入iris,就会马上出现150行的数据



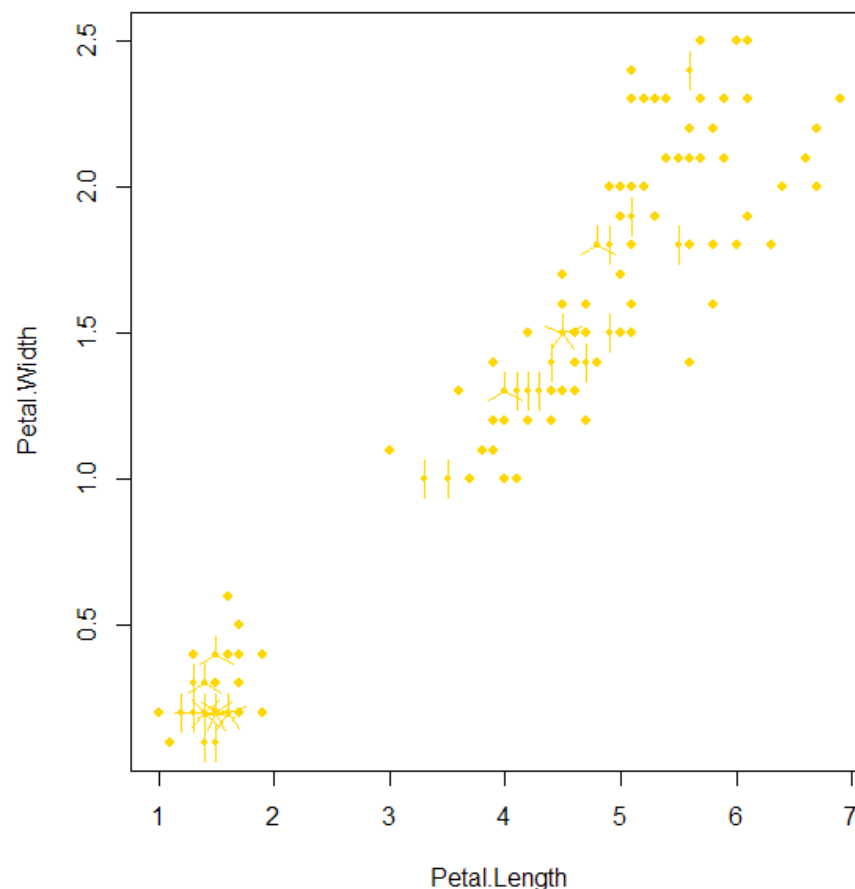
```
> iris
      Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
1           5.1         3.5         1.4         0.2     setosa
2           4.9         3.0         1.4         0.2     setosa
3           4.7         3.2         1.3         0.2     setosa
4           4.6         3.1         1.5         0.2     setosa
5           5.0         3.6         1.4         0.2     setosa
6           5.4         3.9         1.7         0.4     setosa
7           4.6         3.4         1.4         0.3     setosa
8           5.0         3.4         1.5         0.2     setosa
9           4.4         2.9         1.4         0.2     setosa
```

# 向日葵散点图

- 用来克服散点图中数据点重叠问题
- 在有重叠的地方用一朵“向日葵花”的花瓣数目来表示重叠数据的个数

有一些点,会放射出一些小射线出来,表示重复的点  
之前的plot,如果是散点有重叠的话,是看不出来的.  
如果是100个一样的点和另外一个点,那就只有看到两个点

```
sunflowerplot(iris[, 3:4], col =  
"gold", seg.col = "gold")
```



2012.5.10



# 散点图集

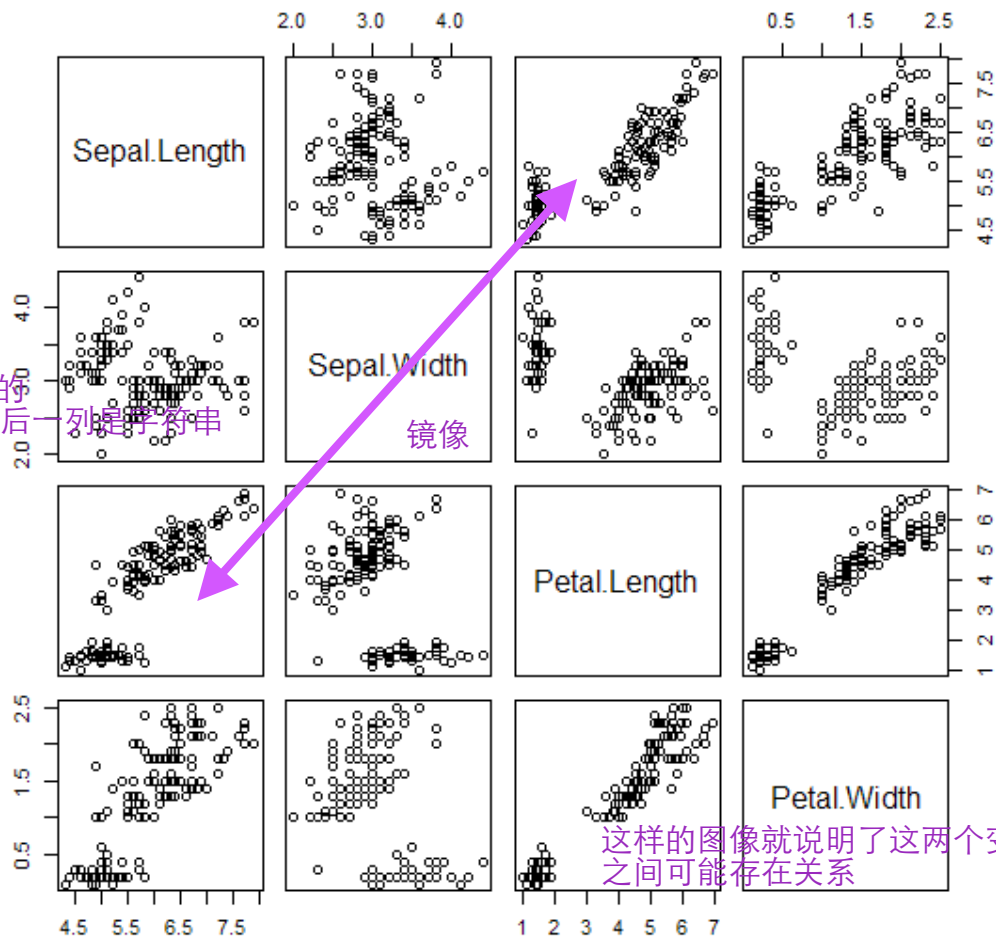
图看起来很复杂  
但是画图的命令很简单

第一列的数据对第二列数据的散点图

- 遍历样本中全部的变量配对  
画出二元图
- 直观地了解所有变量之间的  
关系

指出要画的数据框是什么,是iris数据框中的  
第一列到第四列,第五列就不要了,因为最后一列是字符串  
`pairs(iris[,1:4])`

对四个列进行两两配对,就有14种可能  
对角线上面,都是自己和自己,意义不大,所以是空白的

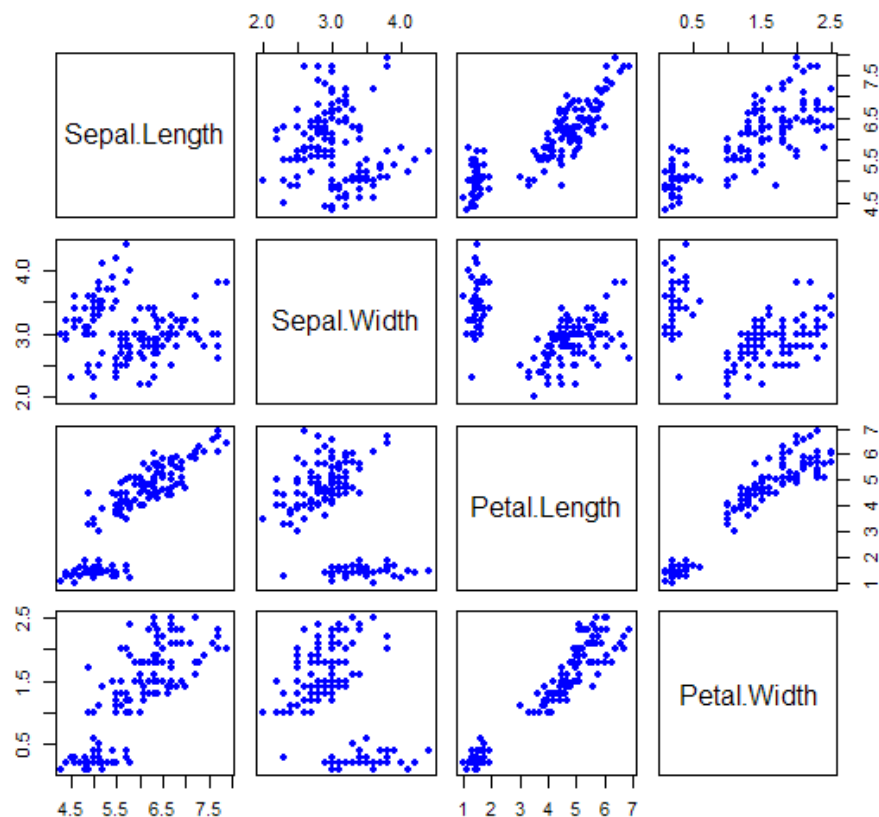


2012.5.10

## ■ 用plot也可以实现同样的效果

```
plot(iris[,1:4],  
     main="Relationships between  
           characteristics of iris flowers", 指定标题  
     pch=19, 指点点的类型是圆盘  
     col="blue", 指点颜色  
     cex=0.9)
```

Relationships between characteristics of iris flowers



- 利用par()在同一个device输出多个散点图

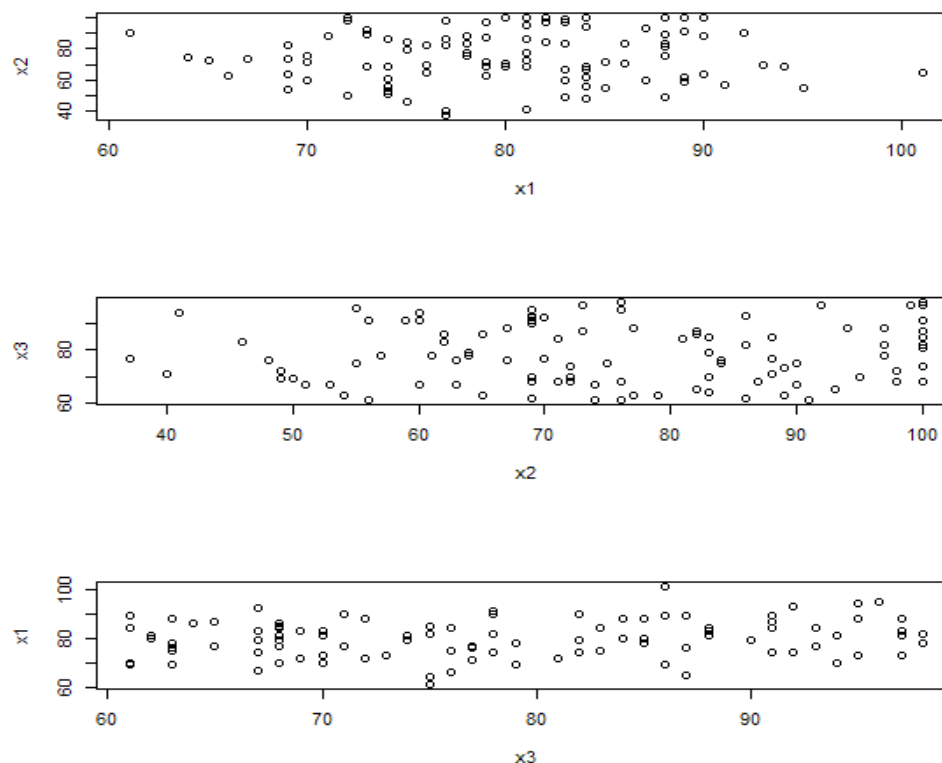
- Par命令博大精深,用于设置绘图参数, help(par)

很复杂,有很多高端用法

par(mfrow=c(3,1)) 把图形分成三行一列,  
三个不同的水平区域

plot(x1,x2);plot(x2,x3);plot(x3,x1)

第一个plot放在第一个水平区域  
第二个plot放在第二个水平区域



- help(par)
- 有哪些颜色？ colors()

```
> colors()
[1] "white"
[4] "antiquewhite1"
[7] "antiquewhite4"
[10] "aquamarine2"
[13] "azure"
[16] "azure3"
[19] "bisque"
[22] "bisque3"
[25] "blanchedalmond"
[28] "blue2"
[31] "blueviolet"
[34] "brown2"
[37] "burlywood"
[40] "burlywood3"
[43] "cadetblue1"
[46] "cadetblue4"
[49] "chartreuse2"

"aliceblue"
"antiquewhite2"
"aquamarine"
"aquamarine3"
"azure1"
"azure4"
"bisque1"
"bisque4"
"blue"
"blue3"
"brown"
"brown3"
"burlywood1"
"burlywood4"
"cadetblue2"
"chartreuse"
"chartreuse3"

"antiquewhite"
"antiquewhite3"
"aquamarine1"
"aquamarine4"
"azure2"
"beige"
"bisque2"
"black"
"blue1"
"blue4"
"brown1"
"brown4"
"burlywood2"
"cadetblue"
"cadetblue3"
"chartreuse1"
"chartreuse4"
```

2012.5.10

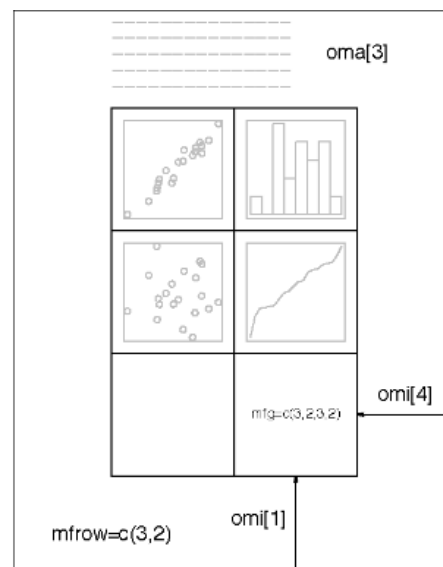
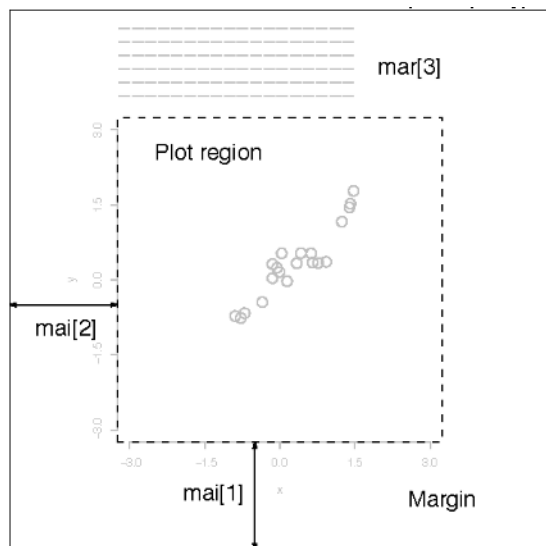
## 关于绘图参数

### ■ 绘图设备

```
dev.cur()    当前的图形窗
dev.list()
dev.next(which = dev.cur())
dev.prev(which = dev.cur())
dev.off(which = dev.cur())
dev.set(which = dev.next())
dev.new(...)
graphics.off()
```

关于画图:推荐一本书: <R Graphs Cookbook>,有对画图的极细致的入门教学.有很多范例

- 位置控制参数
- `mai`参数 : A numerical vector of the form `c(bottom, left, top, right)` which gives the margin size specified in inches.
- `oma`参数 : A vector of the form `c(bottom, left, top, right)` giving the size of the outer margins in lines of text.

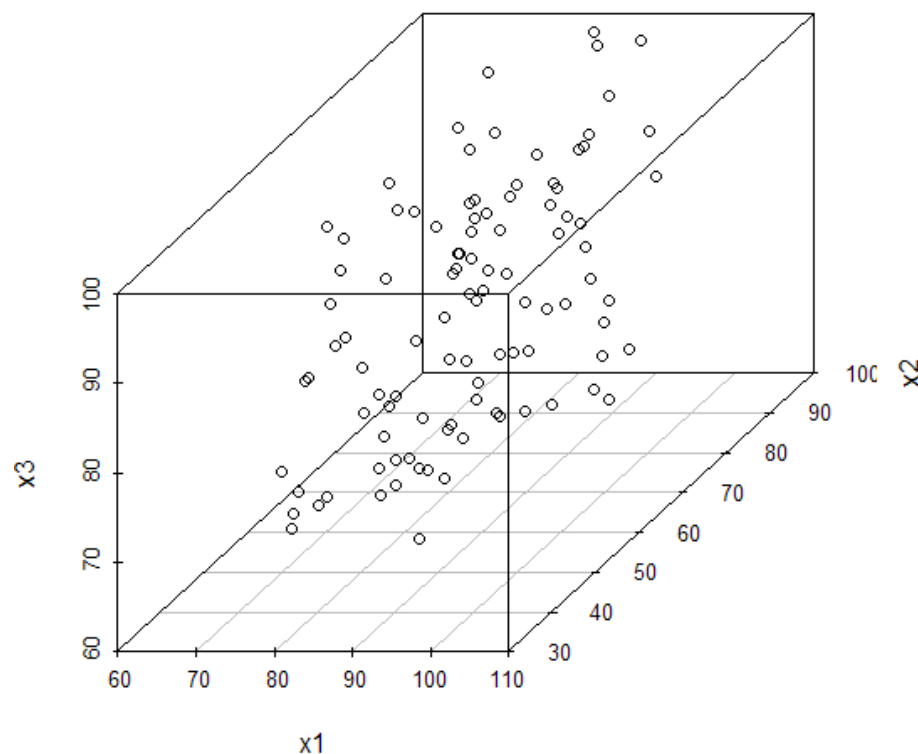


# 三维散点图

## ■ 安装scatterplot3d 包

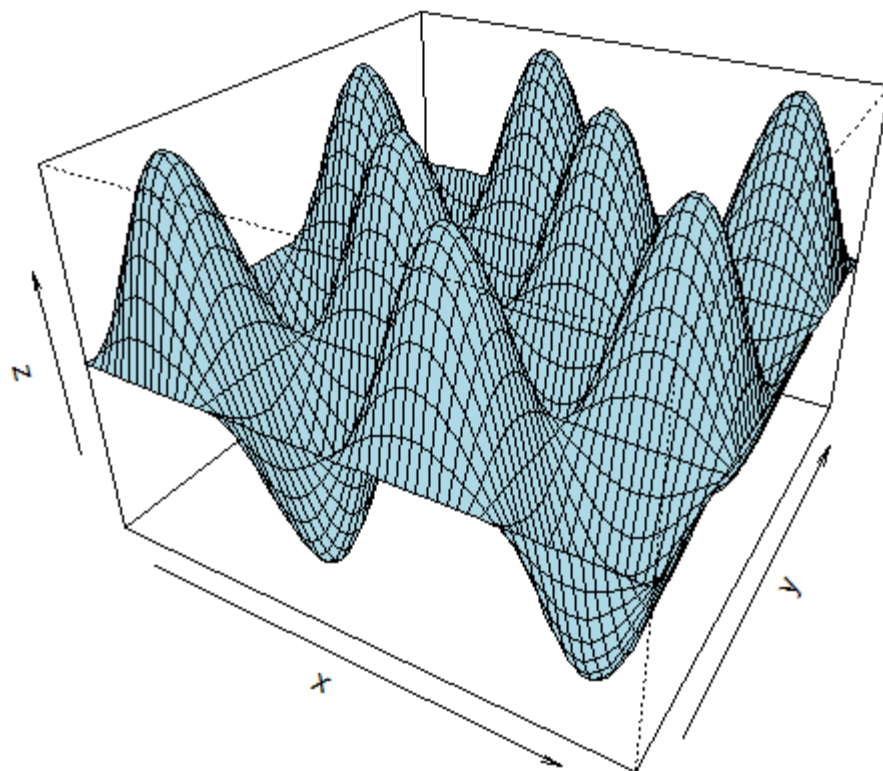
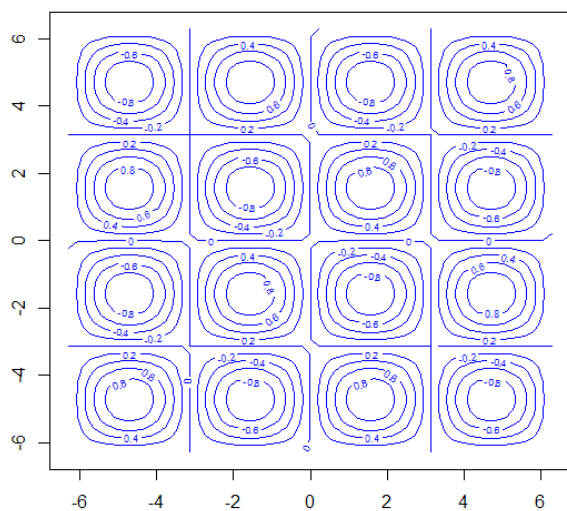
三维散点图

```
scatterplot3d(x[2:4])
```



2012.5.10

```
x<-y<-seq(-2*pi, 2*pi, pi/15)
f<-function(x,y) sin(x)*sin(y)
z<-outer(x, y, f)  outer是什么意思?
contour(x,y,z,col="blue")
persp(x,y,z,theta=30, phi=30,
      expand=0.7,col="lightblue")
```



2012.5.10



调和曲线图是 Andrews (安德鲁斯) 在 1972 年提出来的三角表示法, 其思想是将多维空间中的一个点对应于二维平面的一条曲线, 对于  $p$  维数据, 假设  $X_r$  是第  $r$  观测值, 即

$$X_r^T = (x_{r1}, x_{r2}, \dots, x_{rp}),$$

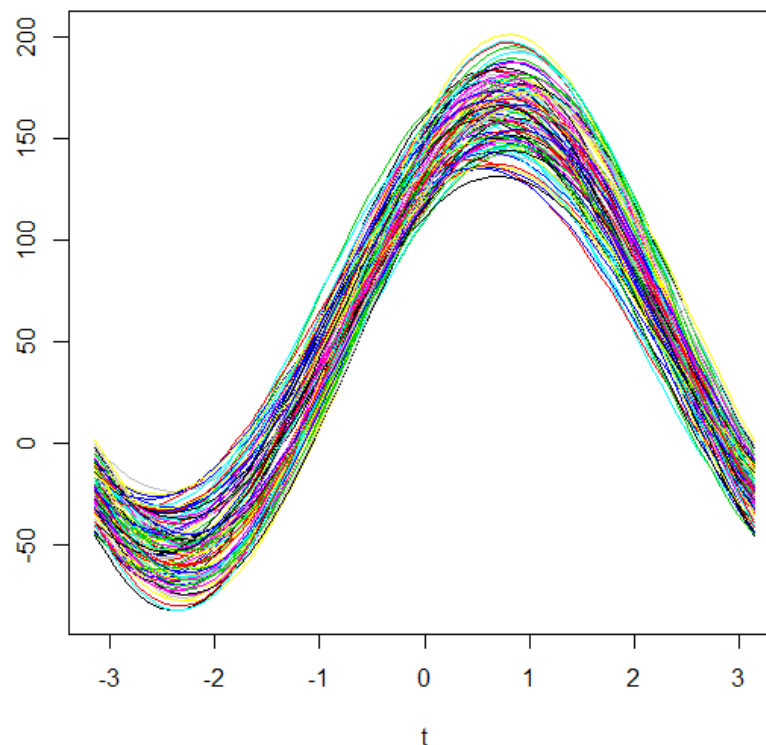
则对应的调和曲线是

$$\begin{aligned} f_r(t) = & \frac{x_{r1}}{\sqrt{2}} + x_{r2} \cdot \sin(t) + x_{r3} \cdot \cos(t) + x_{r4} \cdot \sin(2t) + x_{r5} \cdot \cos(2t) + \\ & + \dots +, \quad -\pi \leq t \leq \pi. \end{aligned} \quad (3.29)$$

- unison.r的代码
- 自定义函数
- 调和曲线用于聚类判断非常方便

```
> source("d:\\unison.R")  
> unison(x[2:4])  
> |
```

The Unison graph of Data



2012.5.10

## ■ 安装maps包

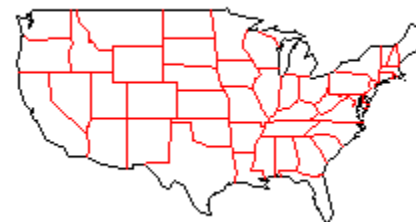
“state” 就是美国

```
map("state", interior = FALSE)
```

就是画边界,里面是没有的

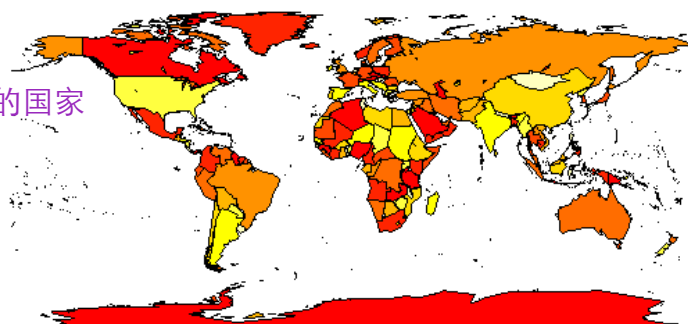


```
map("state", boundary = FALSE, col="red",  
    add = TRUE)
```



```
map('world', fill = TRUE,col=heat.colors(10))
```

热力图的颜色  
用十个不同的颜色来表示不同的国家



# R实验：社交数据可视化

- 先下载安装maps包和geosphere包并加载

```
library(maps)
```

```
library(geosphere)
```

- 画出美国地图

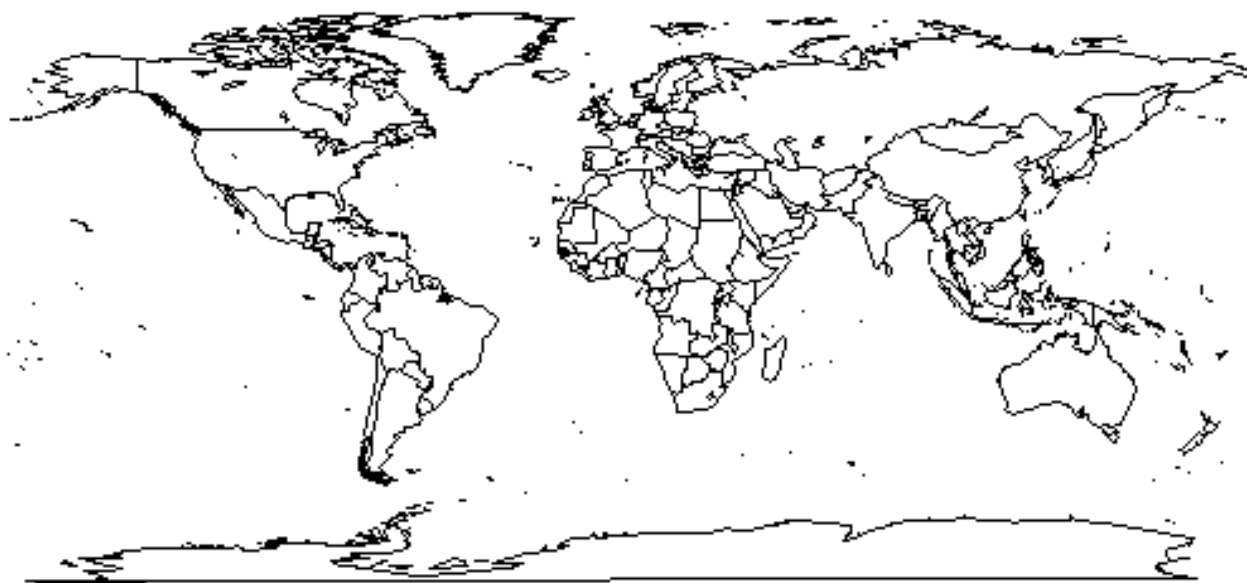
```
map("state")
```



# R实验：社交数据可视化

## ■ 画世界地图

`map("world")`



2012.5.10

# R实验：社交数据可视化

- 通过设置坐标范围使焦点集中在美国周边，并且设置一些有关颜色

```
xlim <- c(-171.738281, -  
          56.601563)      xlimited是纬度
```

```
ylim <- c(12.039321,  
          71.856229)
```

```
map("world", col="#f2f2f2",  
    fill=TRUE, bg="white",  
    lwd=0.05, xlim=xlim,  
    ylim=ylim)
```



# R实验：社交数据可视化

- 画一条弧线连线，表示社交关系

```
lat_ca <- 39.164141
```

```
lon_ca <- -121.64062
```

```
lat_me <- 45.21300
```

```
lon_me <- -68.906250
```

```
inter <-
```

```
  gcIntermediate(c(lon_c  
a, lat_ca), c(lon_me,  
lat_me), n=50,  
addStartEnd=TRUE)
```

```
lines(inter)
```

这条命令是一个弧线



# R实验：社交数据可视化

## ■ 继续画弧线

```
lat_tx <- 29.954935
```

```
lon_tx <- -98.701172
```

```
inter2 <-
```

```
  gcIntermediate(c(lon_ca  
    , lat_ca), c(lon_tx, lat_tx),  
    n=50,  
    addStartEnd=TRUE)
```

```
lines(inter2, col="red")
```





# R实验：社交数据可视化

## ■ 装载数据

和read.table是一个用途  
 不仅可以读本地文件,还可以直接到网上去拉一个csv文件

```
airports <- read.csv("http://datasets.flowingdata.com/tuts/maparcs/airports.csv",
  header=TRUE)
```

```
flights <- read.csv("http://datasets.flowingdata.com/tuts/maparcs/flights.csv",
  header=TRUE, as.is=TRUE)
```

# R实验：社交数据可视化

## ■ 画出多重联系

```
map("world", col="#f2f2f2", fill=TRUE, bg="white", lwd=0.05, xlim=xlim, ylim=ylim)
```

把刚才的语句通过循环再重复,把关系全部的弧线都画出来

```
fsub <- flights[flights$airline == "AA",]
```

```
for (j in 1:length(fsub$airline)) {
```

```
  air1 <- airports[airports$iata == fsub[j,]$airport1,]
```

```
  air2 <- airports[airports$iata == fsub[j,]$airport2,]
```

```
  inter <- gcIntermediate(c(air1[1,]$long, air1[1,]$lat), c(air2[1,]$long, air2[1,]$lat), n=100,  
    addStartEnd=TRUE)
```

```
  lines(inter, col="black", lwd=0.8)
```

```
}
```

## R实验：社交数据可视化



2012.5.10

## R实验：社交数据可视化



<http://flowingdata.com/2011/05/11/how-to-map-connections-with-great-circles/>

2012.5.10



# Thanks

## FAQ时间