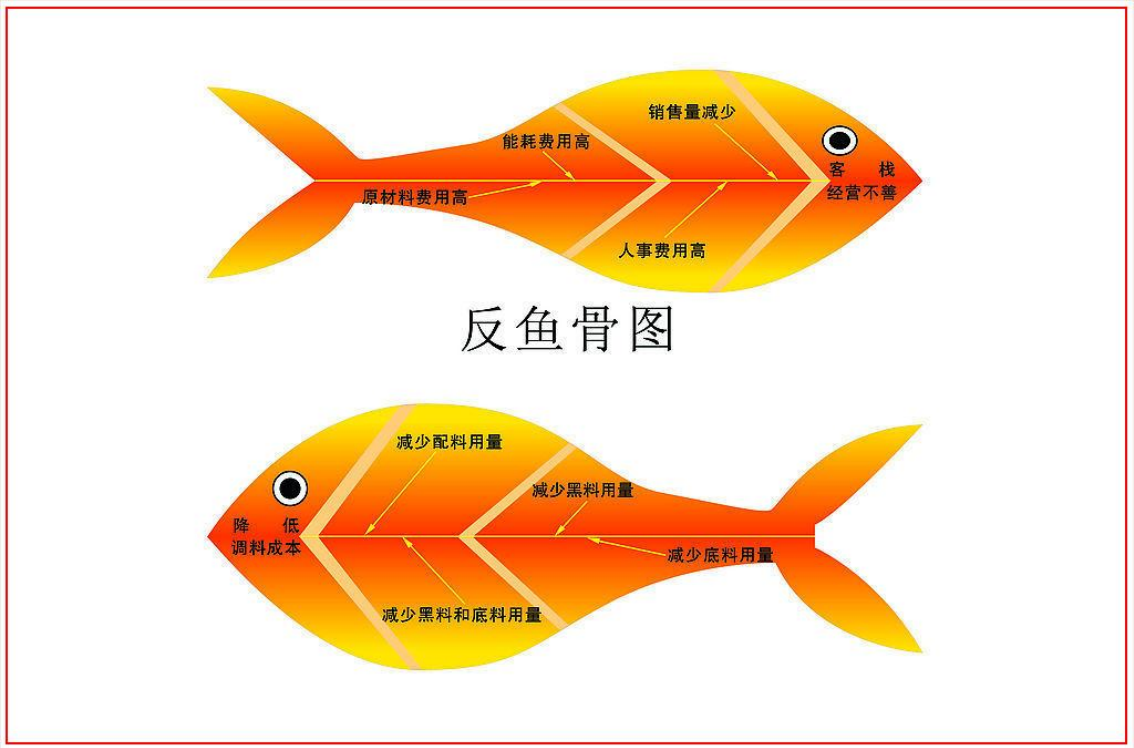


数据分析、展现与R语言 第2周

2013.04.20

法律声明

【声明】本视频和幻灯片为炼数成金网络课程的教学资料，所有资料只能在课程内使用，不得在课程以外范围散播，违者将可能被追究法律和经济责任。

课程详情访问炼数成金培训网站

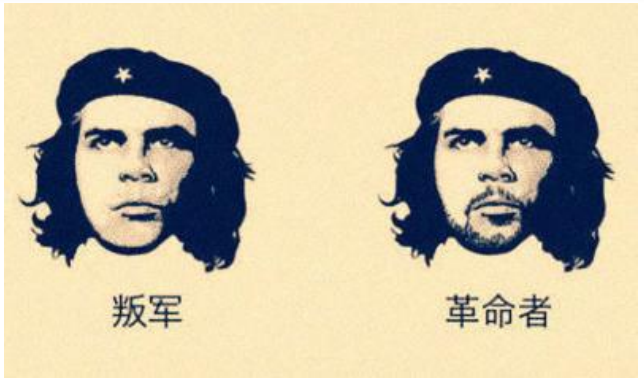
http://edu.dataguru.cn

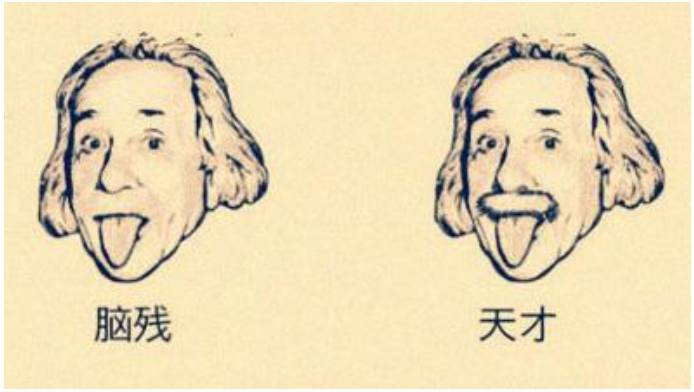
数据可视化的重要性

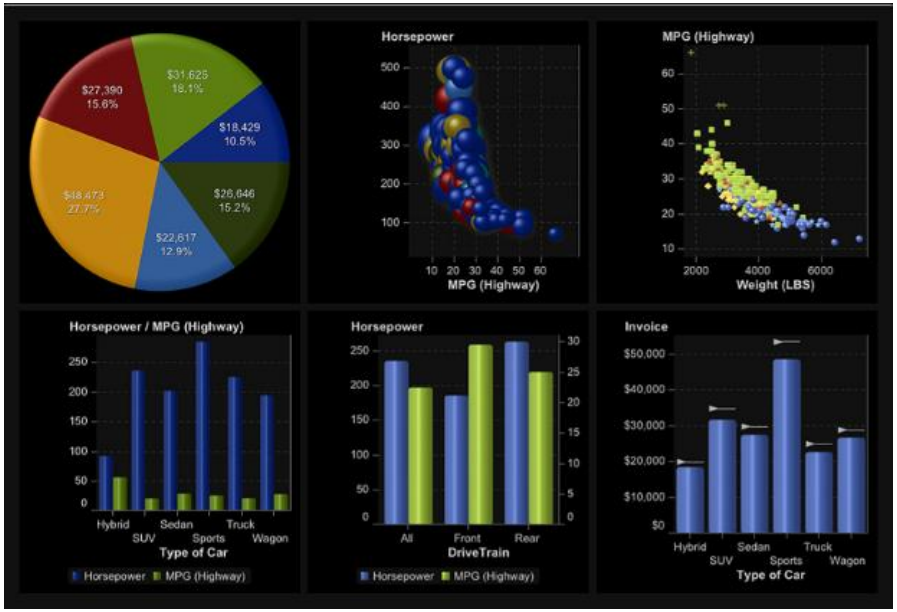
2006年资金预算收支执行情况表

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	筹资活动	合 计
1月份	2700			2700	3610		0.17	3610.17	5476	2082	50	7608	4961	1175	35	6171
2月份	3800			3800	2420		10.2	2430.2	3809	1244	50	5103	2887	108	54	3049
3月份	4274			4274	9474		11	9485	4326	1496	50	6072	4529	6088	30	10647
4月份	12396			12396	11121	68	2097	13286	5386	1314	50	7150	4246	1230	33	5509
5月份	5311	132		5463	5784	98	94	5976	5841	2431	440	8712	4785	792	432	6009
6月份	3801			3801	1217	15	103	1335	4332	2904	87	7323	4067	1903	33	6003
7月份	5951			5951	4427	65	3593	8085	4083	2593	331	7007	5238	2187	332	7737
8月份	5388			5388	1883		2021	3904	3375	3830	2120	9325	3133	3472	2120	8725
9月份	2830			2830	2459	2	914	3375	3953	2505	93	6553	2800	1469	85	4354
10月份	3250			3250	2853		49	2904	4283	2209	40	6534	3526	1591	39	5156
11月份	3870		700	4570	647		134	781	5873	6036	540	12449	810	3861	540	5211
12月份	4105		2150	6255	7723		2576	10299	7631	3551	88	11270	7065	1838	86	8989
合 计	57676	132	2850	60678	53620	248	11602.37	65470.37	58774	32793	3939	95506	48027	25714	3819	77560







2013.04.20

综合性例子

- 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
```

模拟成绩

- 用runif和rnorm

```
> 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
> 
```

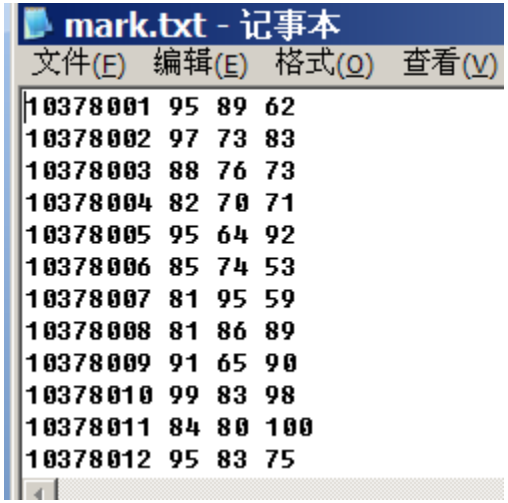
```
> x2=round(rnorm(100,mean=80,sd=7))
> x2
  [1] 89 73 76 70 64 74 95 86 65 83 80 83 71 72 83 79 91 70 75 84 72 94 74 81
 [25] 81 74 85 96 69 84 73 93 72 76 73 81 76 92 73 81 88 80 87 81 81 66 76 85
 [49] 97 77 83 77 82 86 76 69 83 83 77 79 84 90 82 81 81 79 76 90 80 83 88 93
 [73] 75 83 89 93 69 97 85 85 93 73 73 79 75 64 81 81 55 63 81 80 84 78 88 75
 [97] 75 78 78 87
> 
```

模拟成绩

```
> 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
13   10378013  89 71 100
```

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

计算各科平均分

- 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( )

```
> 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

```
> 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
```

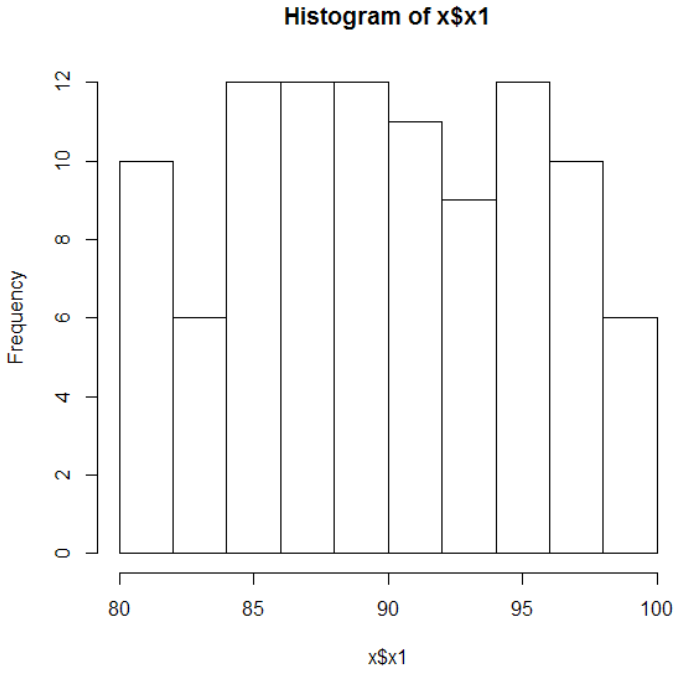
总分最高的同学

```
> 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
> x$num[which.max(apply(x[c("x1","x2","x3")],1,sum))]
[1] 10378078
>
```

对x1进行直方图分析

- 绘制直方图函数hist( )

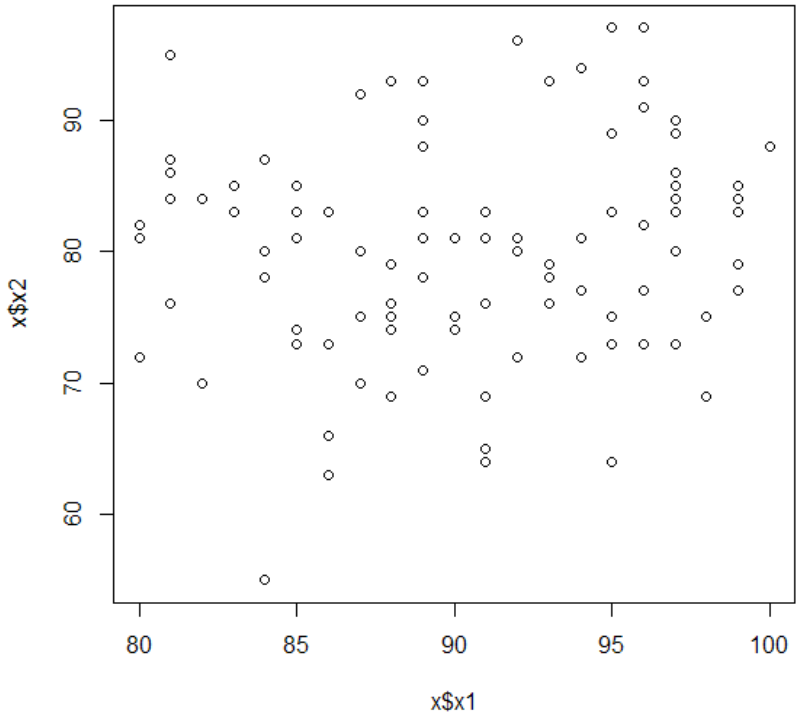
```
> hist(x$x1)
>
```



探索各科成绩的关联关系

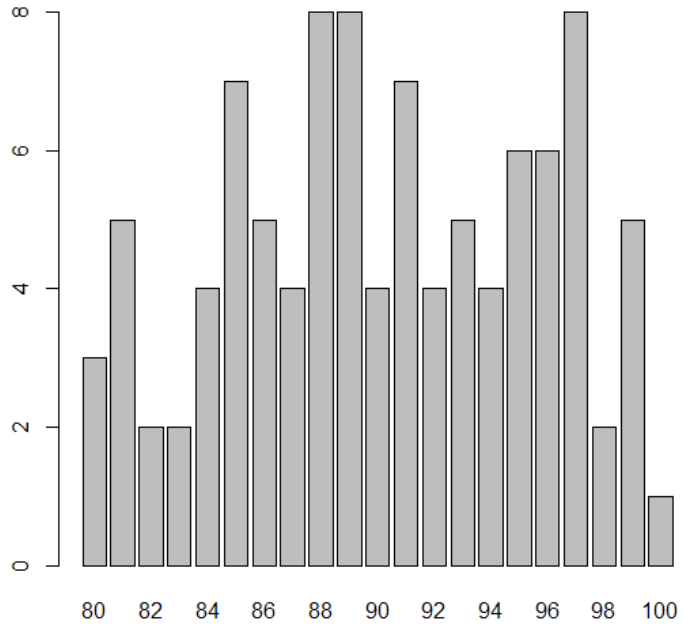
- 散点图绘制函数plot( )

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



列联表分析

- 列联函数table( )，柱状图绘制函数barplot( )

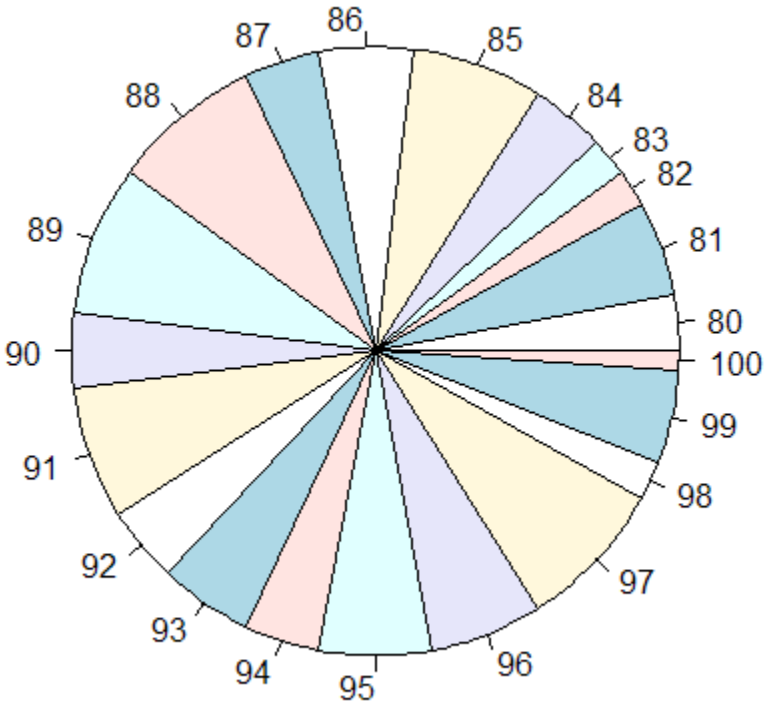



```
> table(x$x1)

 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))
```

饼图

- 饼图绘制函数pie( )

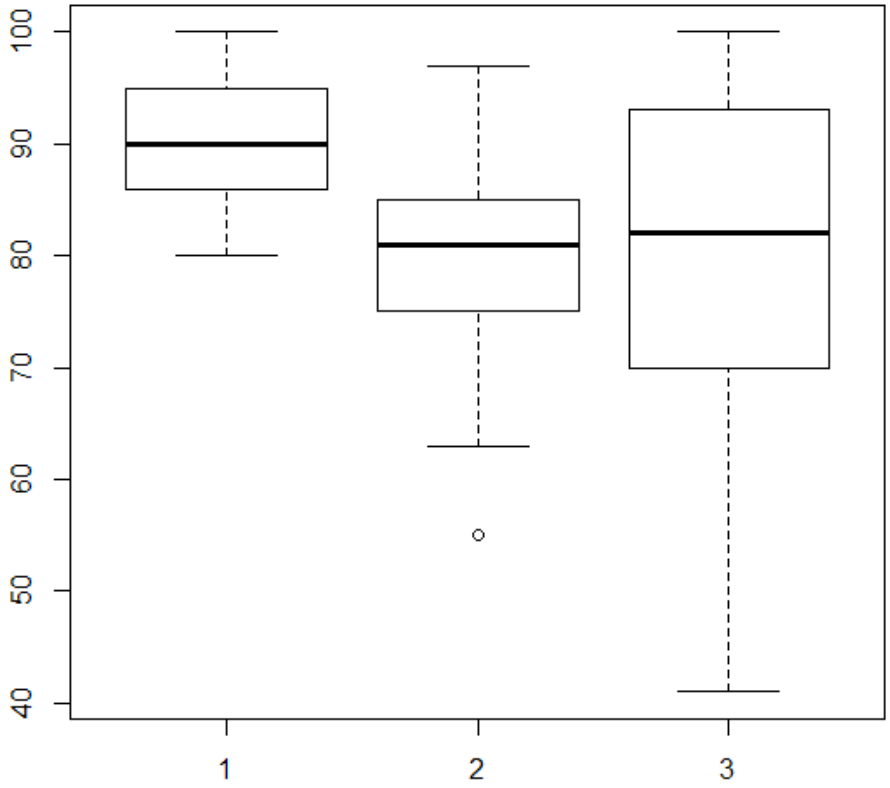
```
>
> pie(table(x$x1))
>
```



箱尾图

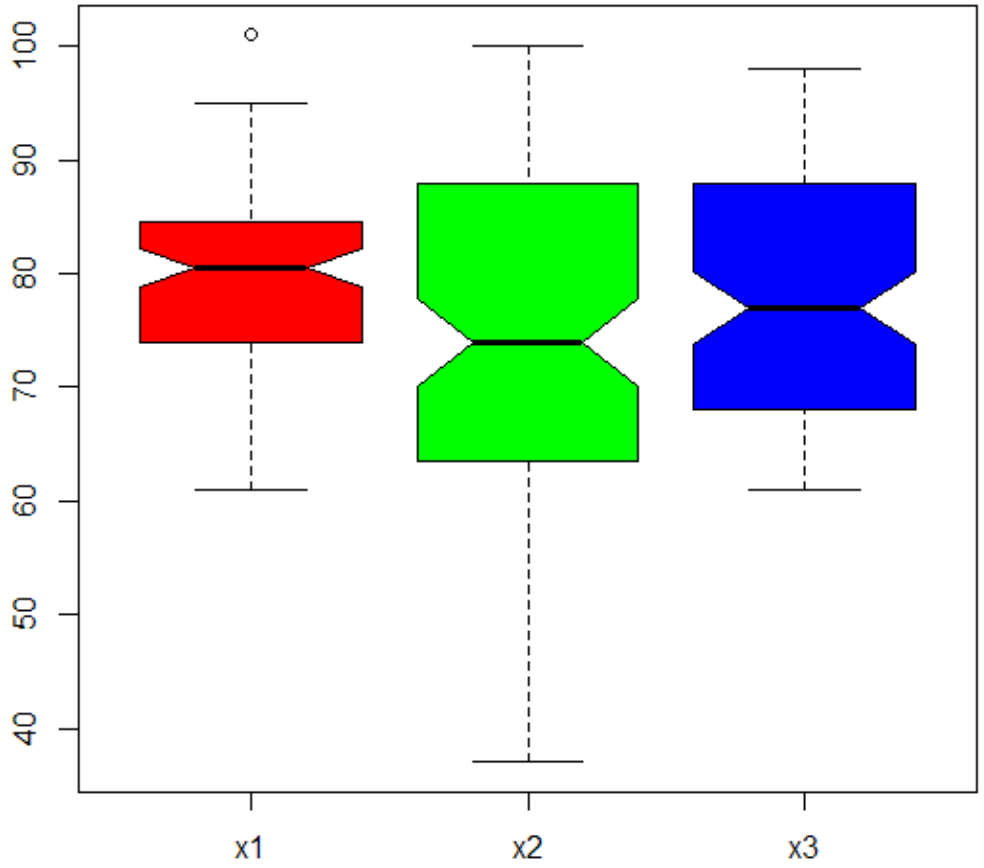
- 箱子的上下横线为样本的25%和75%分位数
- 箱子中间的横线为样本的中位数
- 上下延伸的直线称为尾线，尾线的尽头为最高值和最低值
- 异常值

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



箱线图

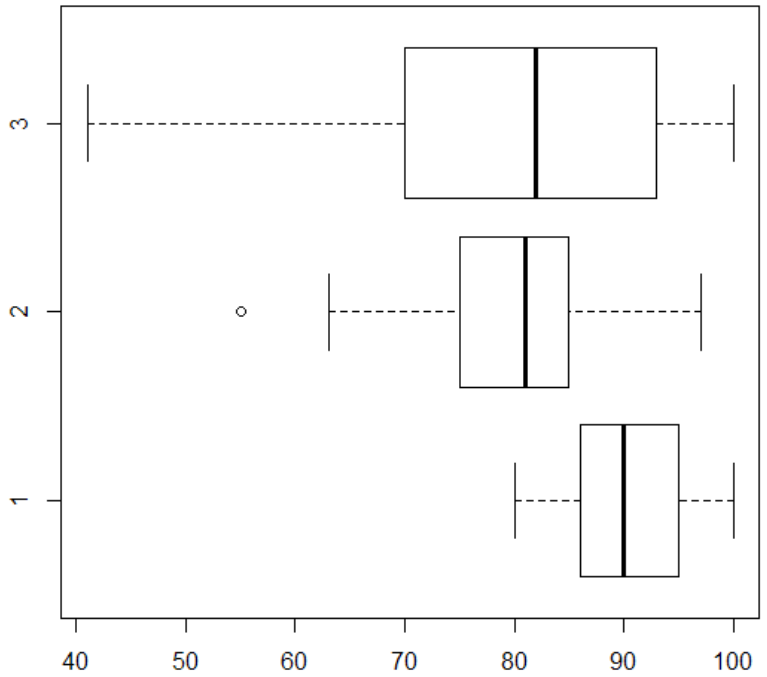
```
boxplot(x[2:4],col=c("red","gre
    en","blue"),notch=T)
```



箱尾图

- 水平放置的箱尾图

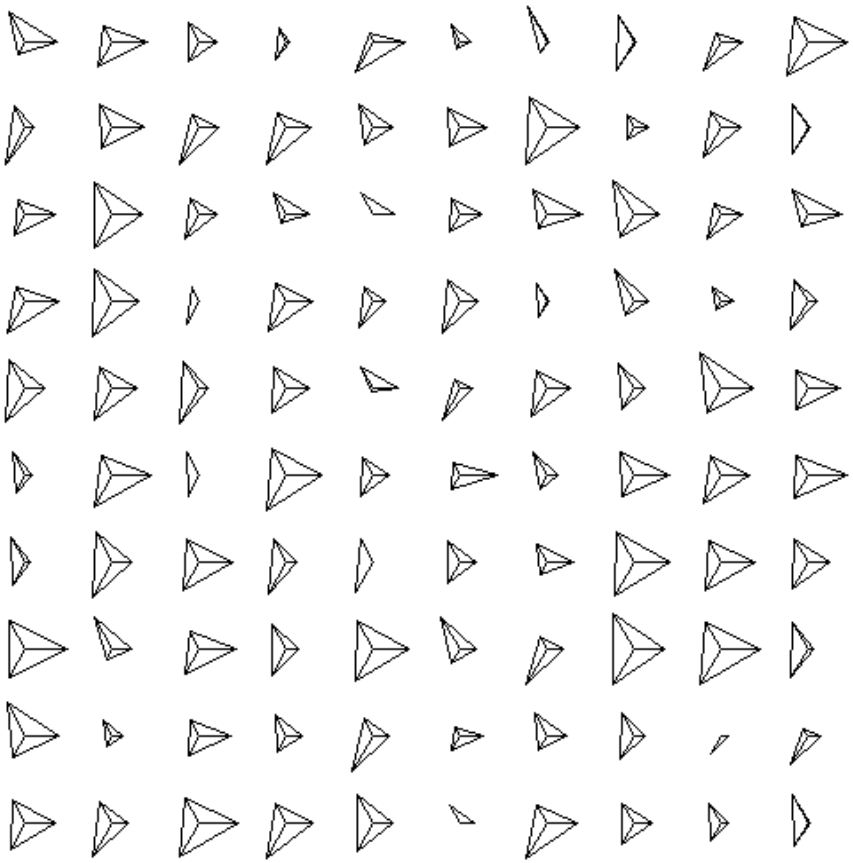
```
> boxplot(x$x1,x$x2,x$x3,horizontal=T)
> 
```



星相图

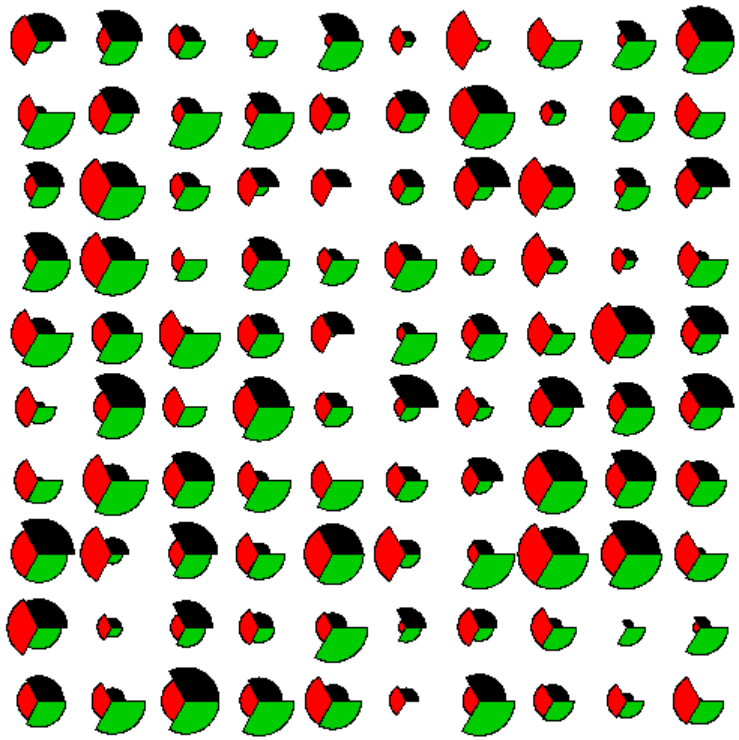
- 每个观测单位的数值表示为一个图形
- 每个图的每个角表示一个变量，字符串类型会标注在图的下方
- 角线的长度表达值的大小

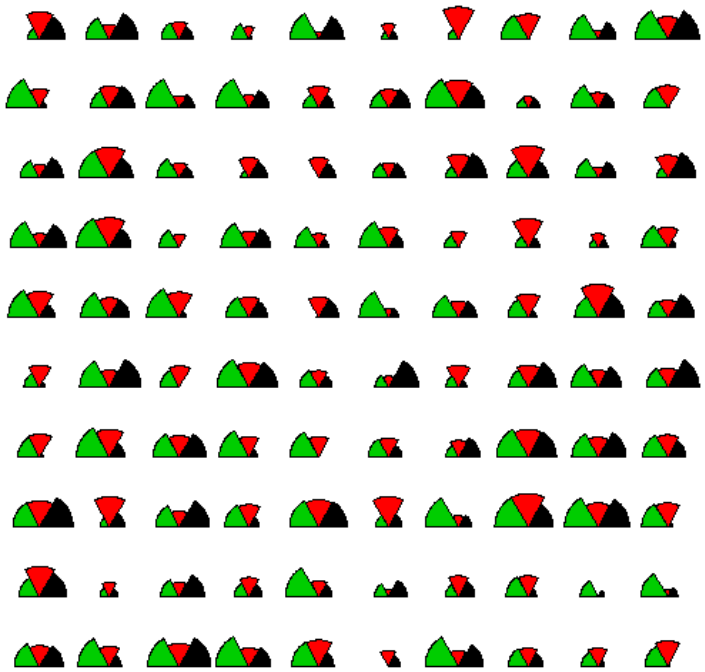
```
> stars(x[c("x1","x2","x3")])
>
```



星相图

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





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

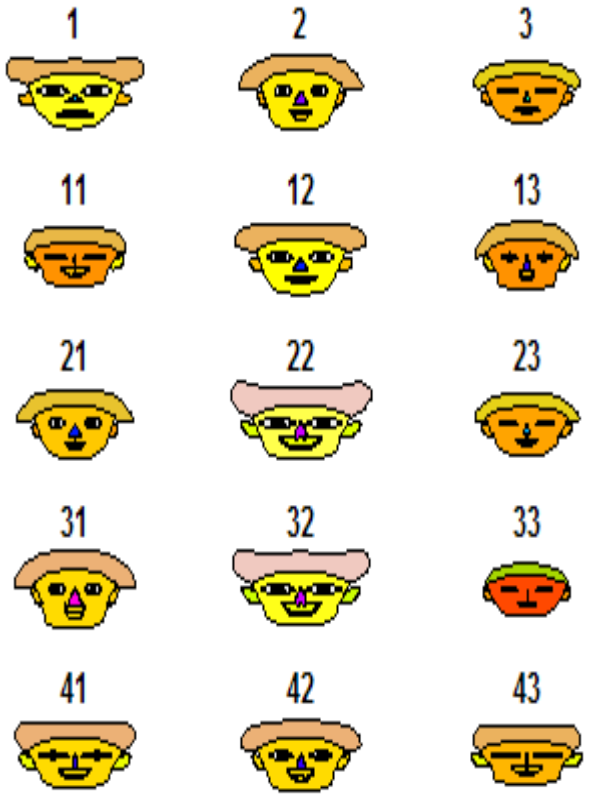
脸谱图

- 安装aplpack包

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

脸谱图

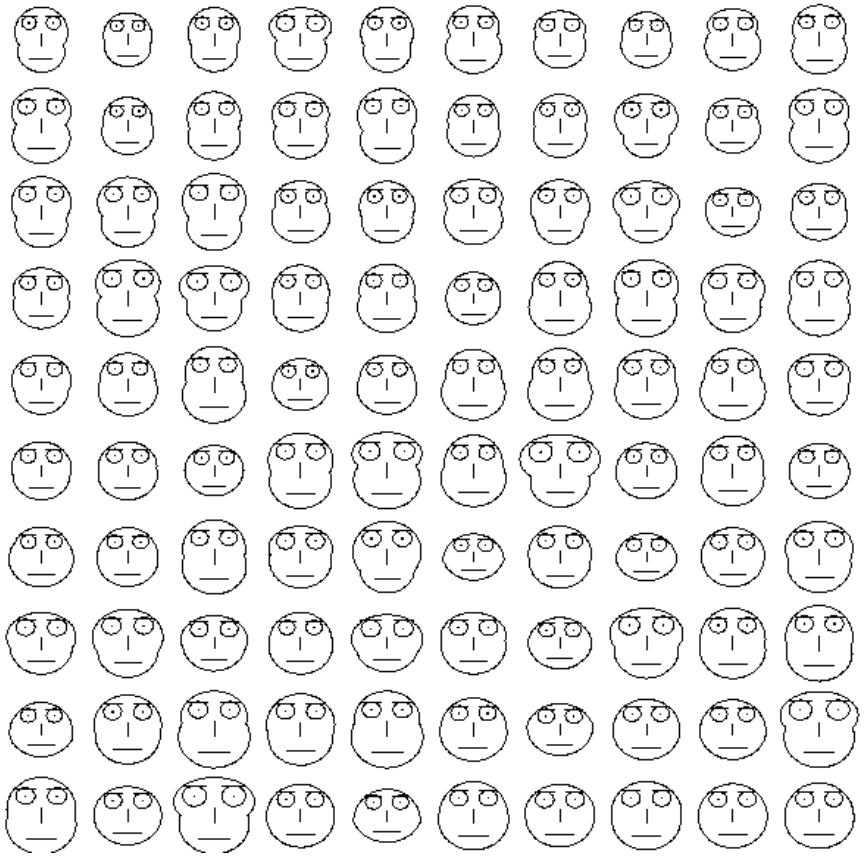
- 用五官的宽度和高度来描绘数值
- 人对脸谱高度敏感和强记忆
- 适合较少样本的情况



其它脸谱图

- 安装TeachingDemos包

```
> library(TeachingDemos)
> faces2(x)
```



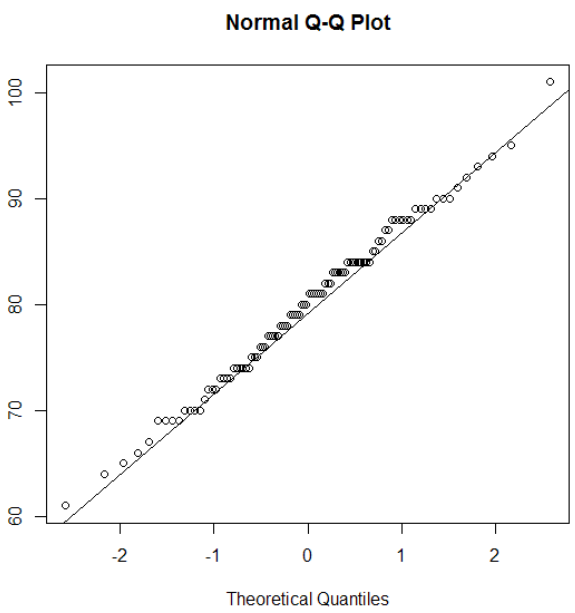
茎叶图

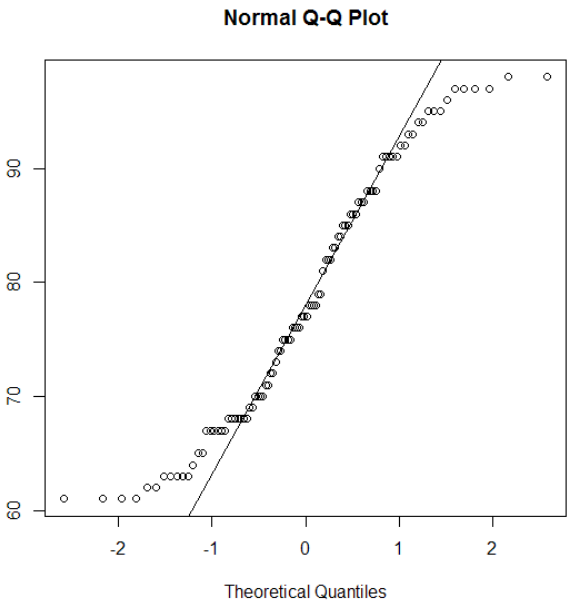
```
> stem(x$x1)

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

   6 | 14
   6 | 5679999
   7 | 00001222333444444
   7 | 5556667777888899999
   8 | 000111111122233333444444444
   8 | 5566778888889999
   9 | 0001234
   9 | 5
  10 | 1
```

QQ图

- 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)
```

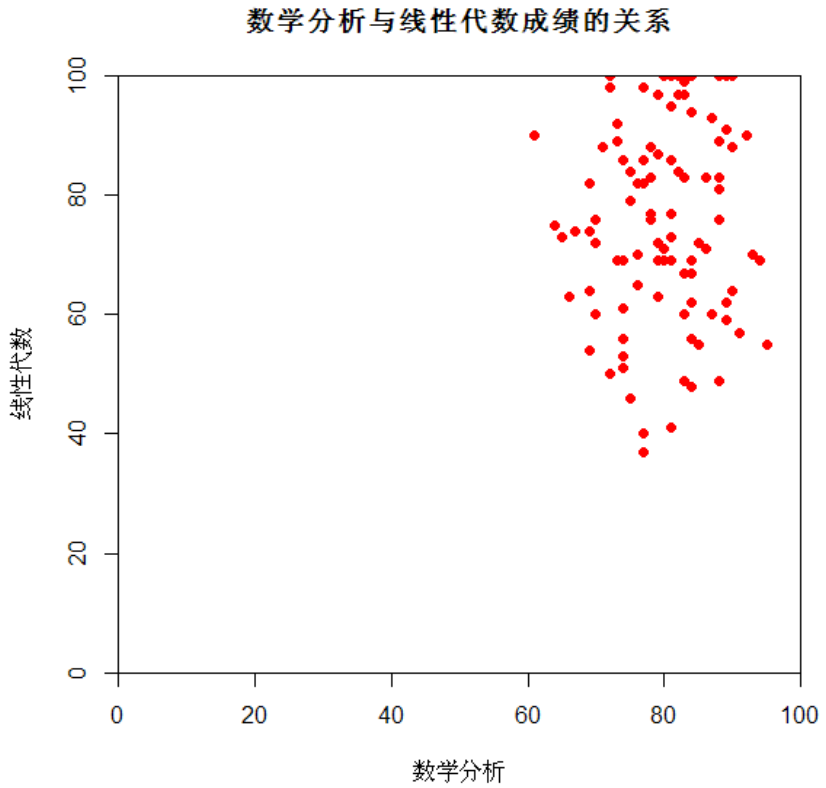




散点图

- 散点图的进一步设置

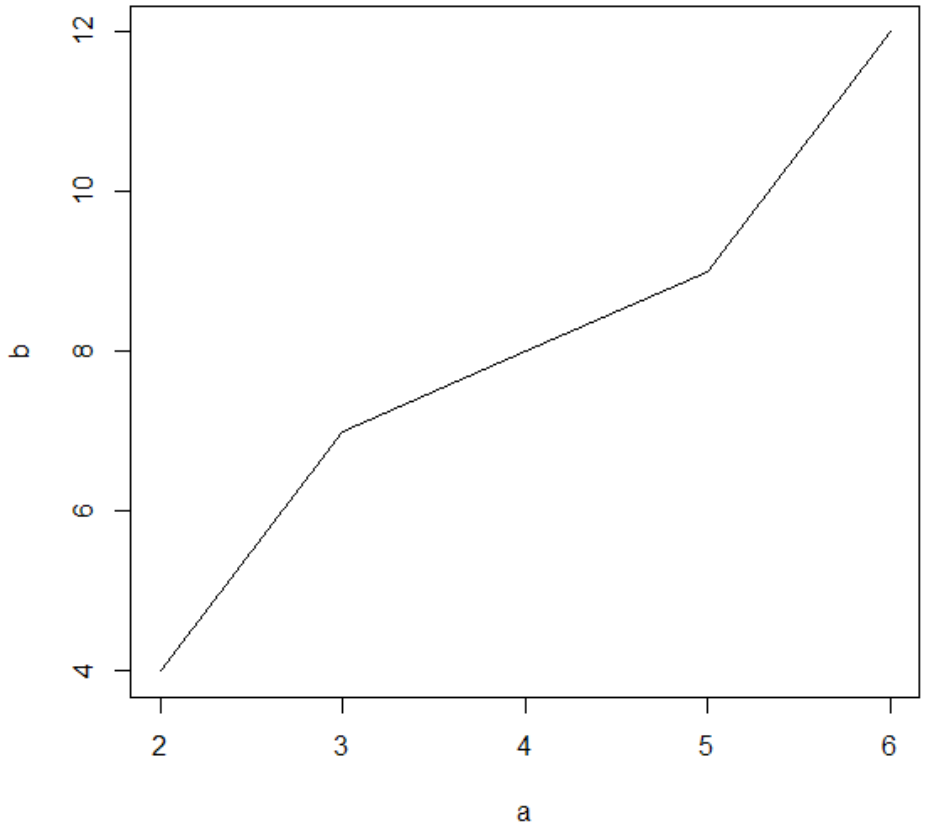
```
plot(x$x1,x$x2,
main="数学分析与线性代数成绩的关系",
xlab="数学分析",
ylab="线性代数",
xlim=c(0,100),
ylim=c(0,100),
xaxs="i", #Set x axis style as internal
yaxs="i", #Set y axis style as internal
col="red", #Set the color of plotting symbol to red
pch=19) #Set the plotting symbol to filled dots
```



散点图

- 连线图

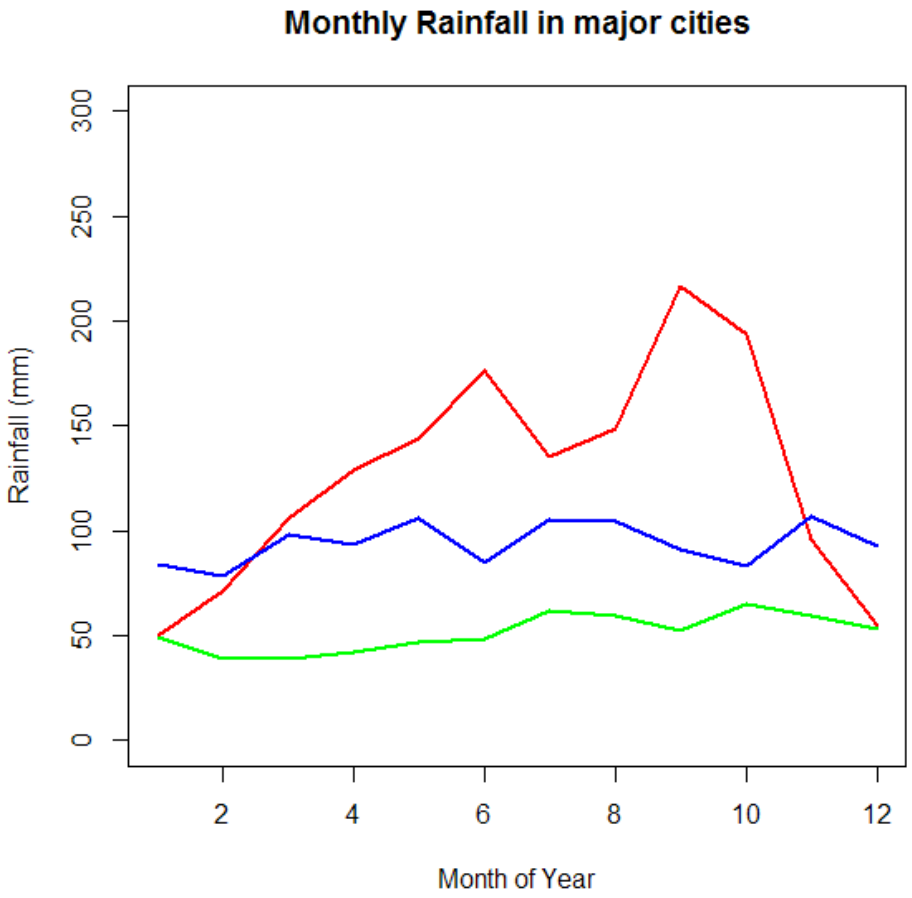
```
a=c(2,3,4,5,6)
b=c(4,7,8,9,12)
plot(a,b,type="l")
```



散点图

- 多条曲线的效果

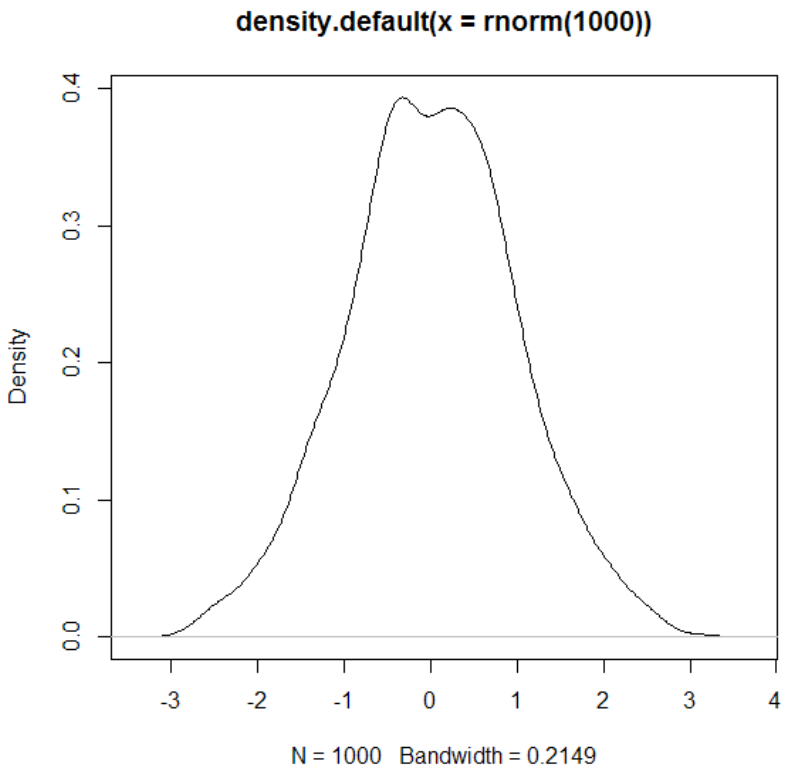
```
plot(rain$Tokyo,type="l",col="red",
ylim=c(0,300),
main="Monthly Rainfall in major cities",
xlab="Month of Year",
ylab="Rainfall (mm)",
lwd=2)
lines(rain$NewYork,type="l",col="blue",lwd=2)
lines(rain$London,type="l",col="green",lwd=2)
lines(rain$Berlin,type="l",col="orange",lwd=2)
```



密度图

- 函数density( )

plot(density(rnorm(1000)))



R内置数据集

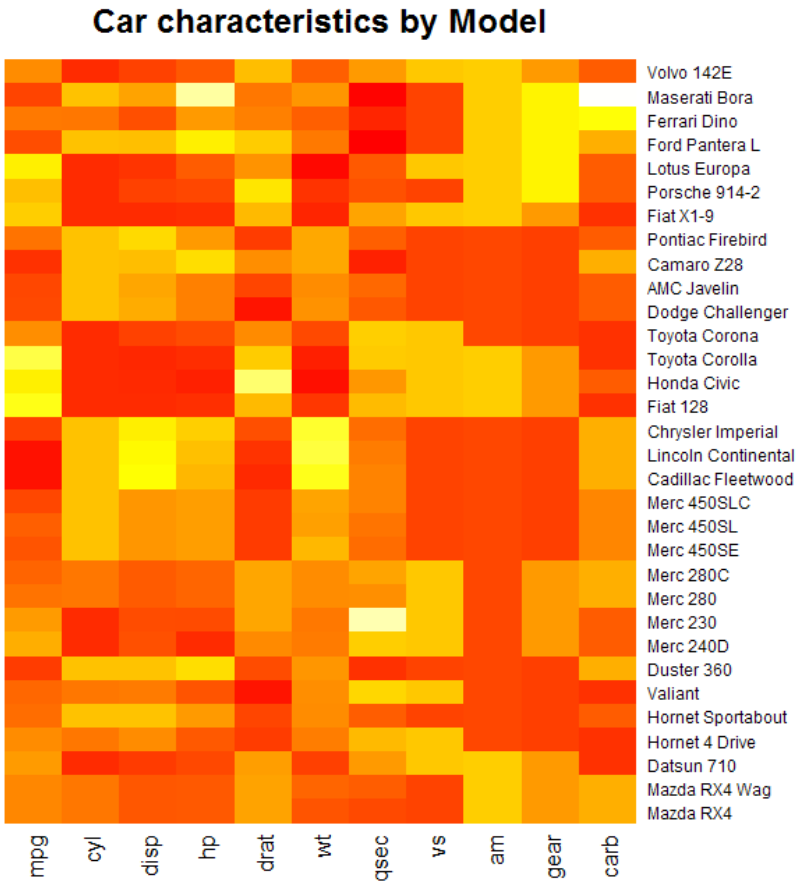
- 函数data( )列出内置数据

```
> 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
```

热力图

- 利用内置的mtcars数据集绘制

```
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

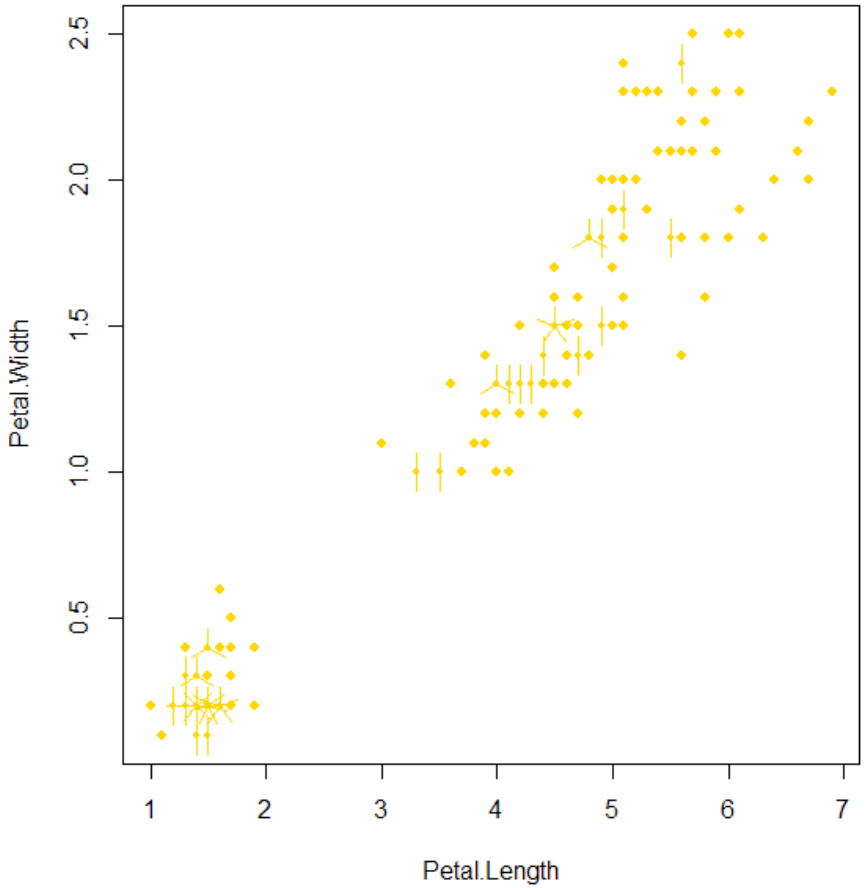
- Sepal 花萼
- Petal 花瓣
- Species 种属

```
> 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

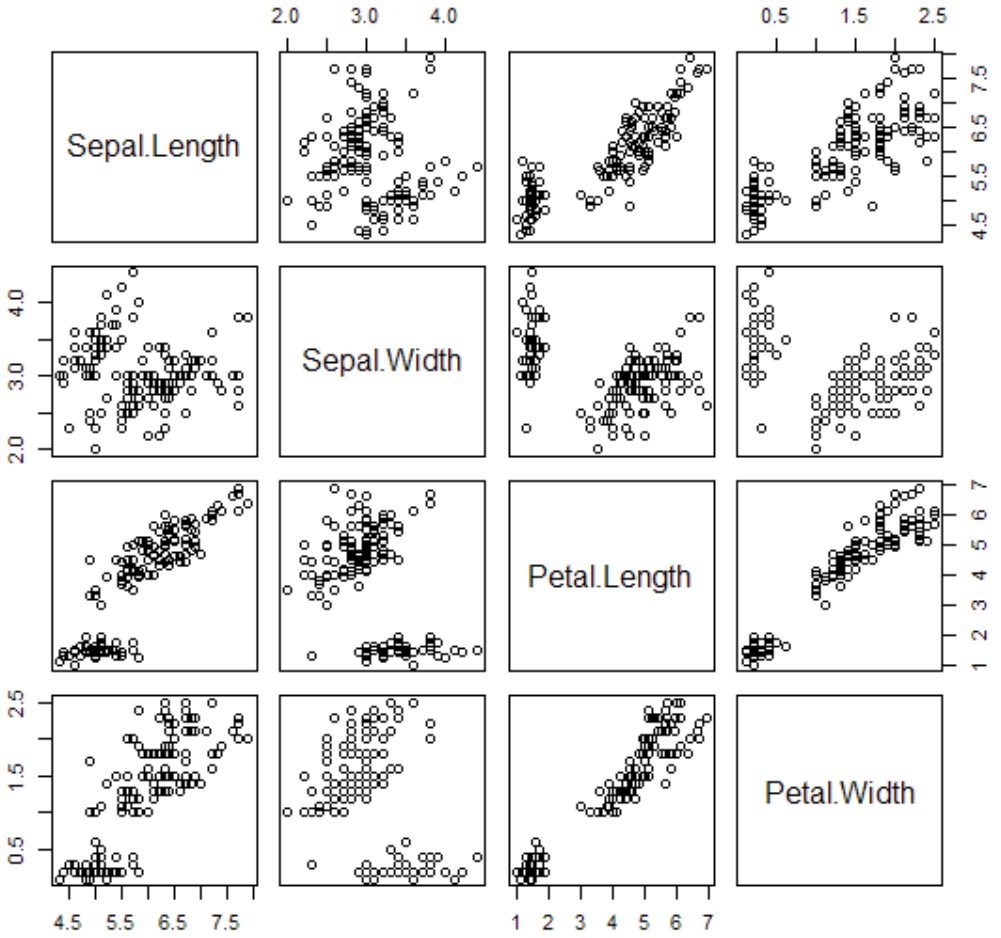
```
sunflowerplot(iris[, 3:4], col =
    "gold", seg.col = "gold")
```



散点图集

- 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

pairs(iris[,1:4])

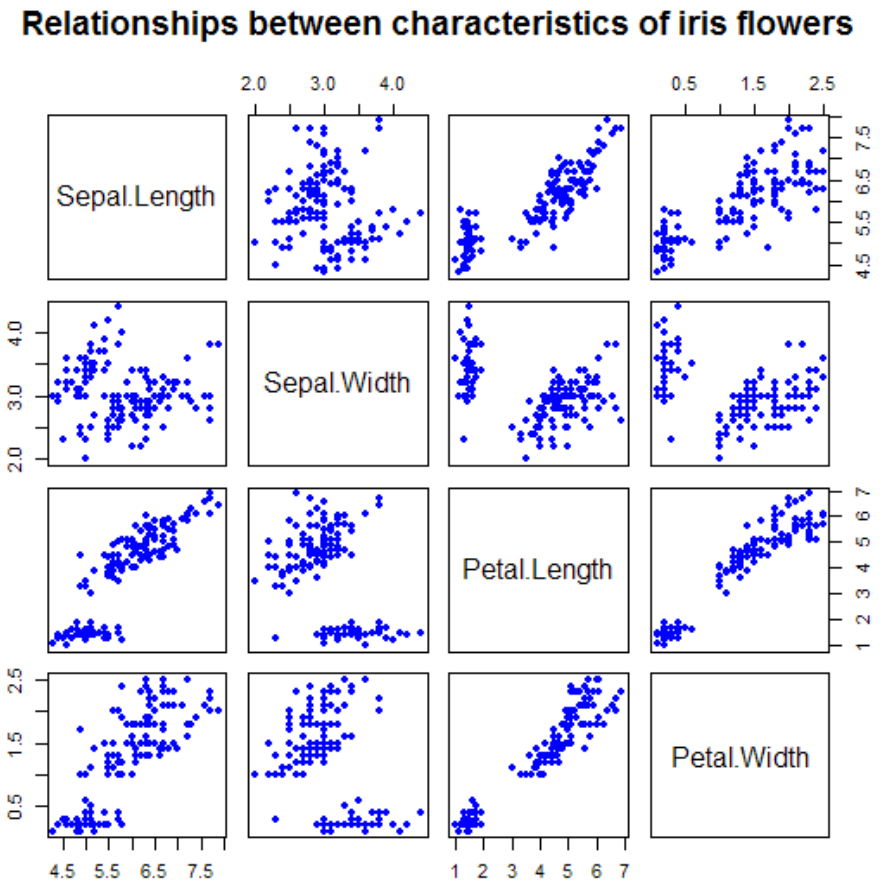


2013.04.20

散点图集

- 用plot也可以实现同样的效果

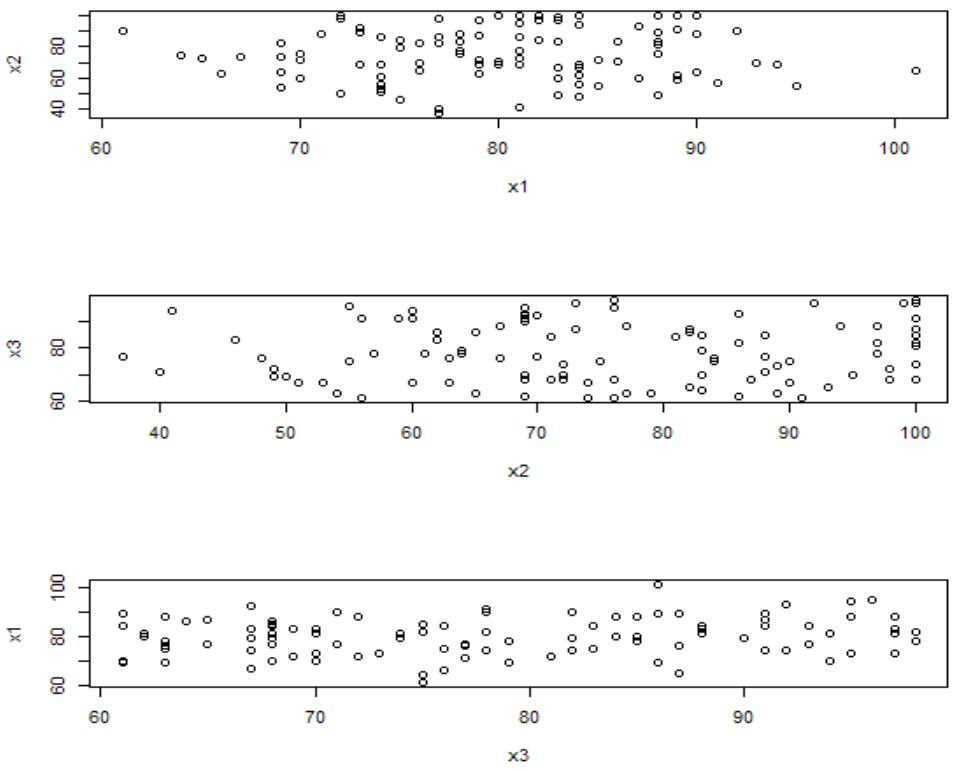
```
plot(iris[,1:4],
main="Relationships between
      characteristics of iris flowers",
pch=19,
col="blue",
cex=0.9)
```



散点图集

- 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)

```
par(mfrow=c(3,1))
plot(x1,x2);plot(x2,x3);plot(x3,x1)
```



关于绘图参数

- help(par)
- 有哪些颜色？ colors( )

```
> colors()
  [1] "white"           "aliceblue"       "antiquewhite"
  [4] "antiquewhite1"   "antiquewhite2"   "antiquewhite3"
  [7] "antiquewhite4"   "aquamarine"      "aquamarine1"
 [10] "aquamarine2"     "aquamarine3"     "aquamarine4"
 [13] "azure"           "azure1"          "azure2"
 [16] "azure3"          "azure4"          "beige"
 [19] "bisque"          "bisque1"         "bisque2"
 [22] "bisque3"         "bisque4"         "black"
 [25] "blanchedalmond"  "blue"            "blue1"
 [28] "blue2"           "blue3"           "blue4"
 [31] "blueviolet"      "brown"           "brown1"
 [34] "brown2"          "brown3"          "brown4"
 [37] "burlywood"       "burlywood1"      "burlywood2"
 [40] "burlywood3"      "burlywood4"      "cadetblue"
 [43] "cadetblue1"      "cadetblue2"      "cadetblue3"
 [46] "cadetblue4"      "chartreuse"      "chartreuse1"
 [49] "chartreuse2"     "chartreuse3"     "chartreuse4"
```

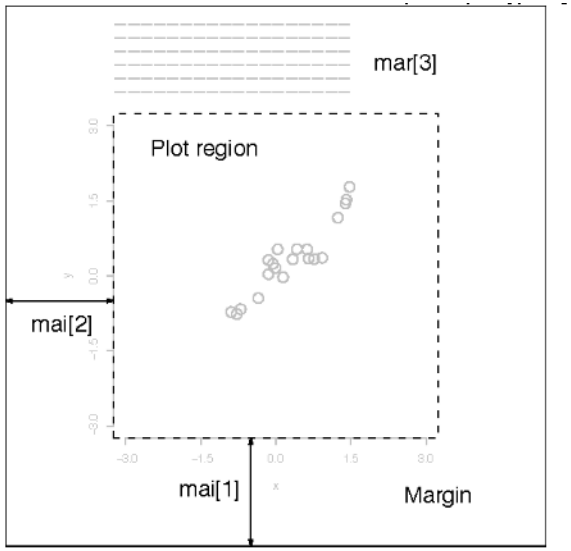
关于绘图参数

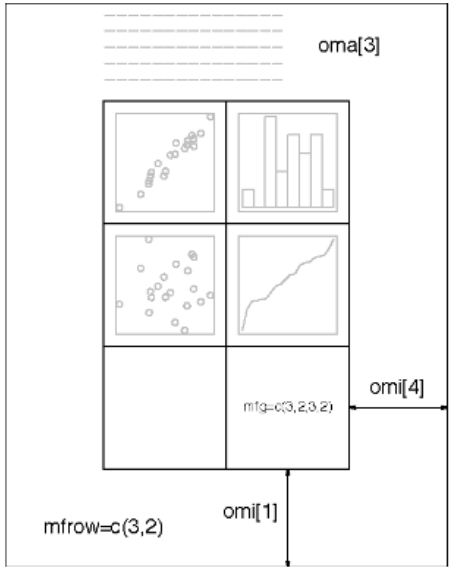
- 绘图设备

```
dev.cur()
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()
```

关于绘图参数

- 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.

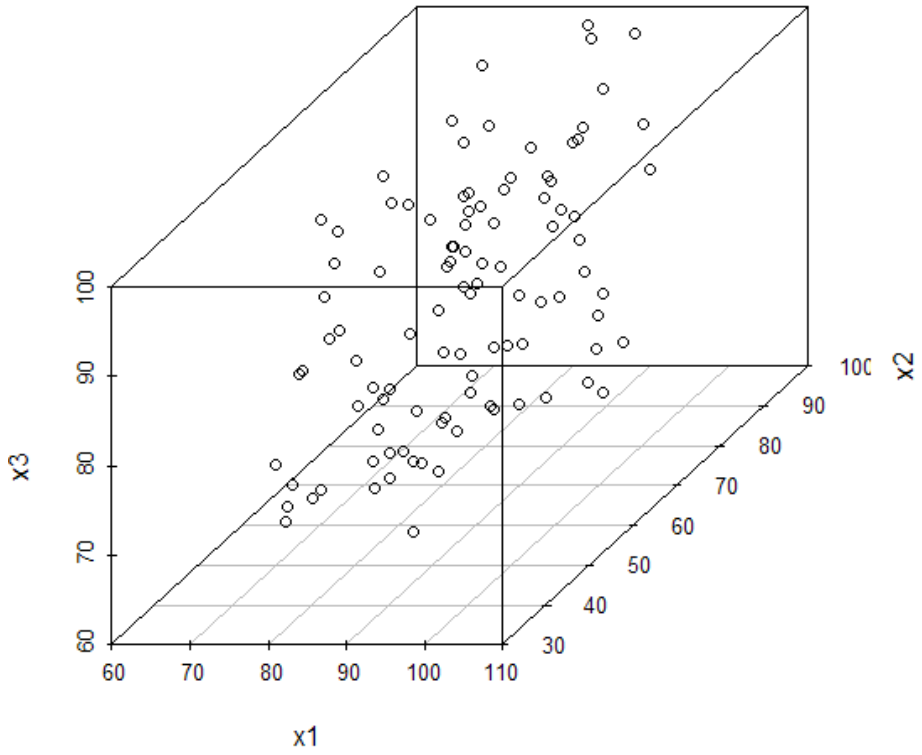




三维散点图

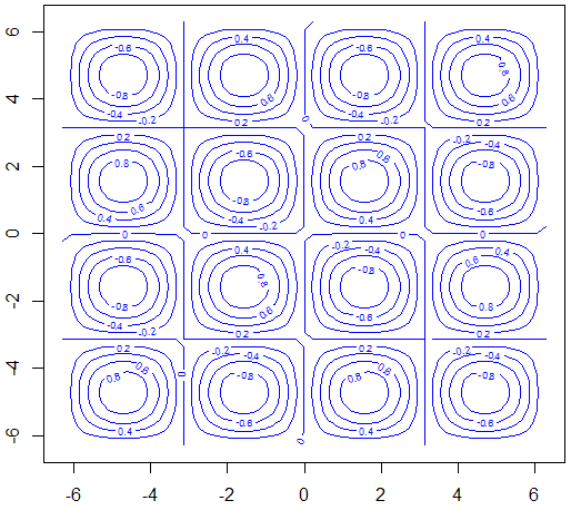
- 安装scatterplot3d 包

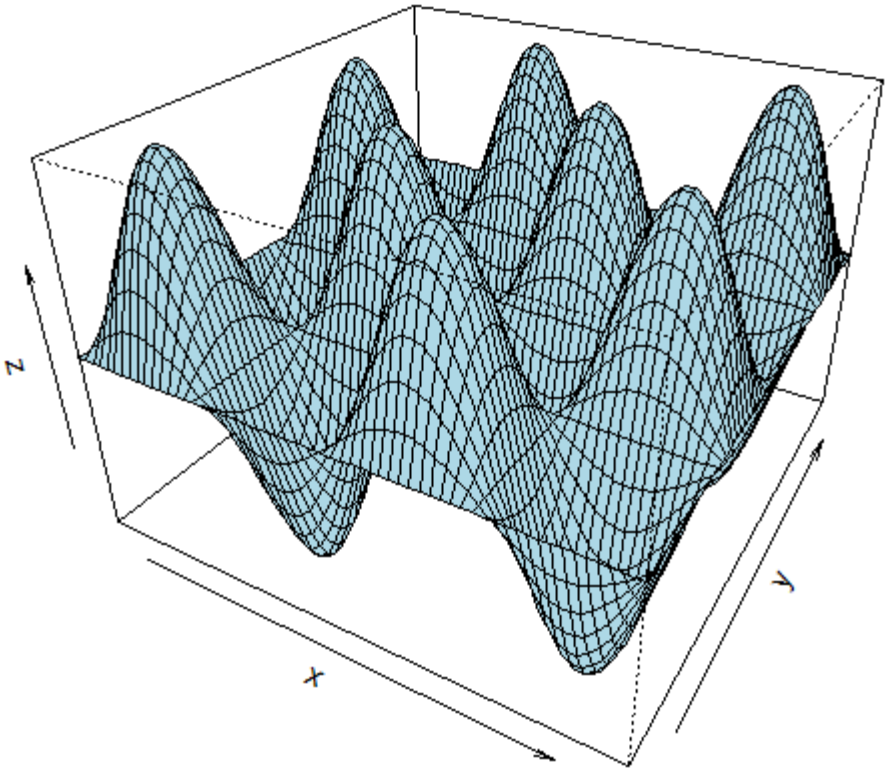
scatterplot3d(x[2:4])



三维作图

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





调和曲线图

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

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

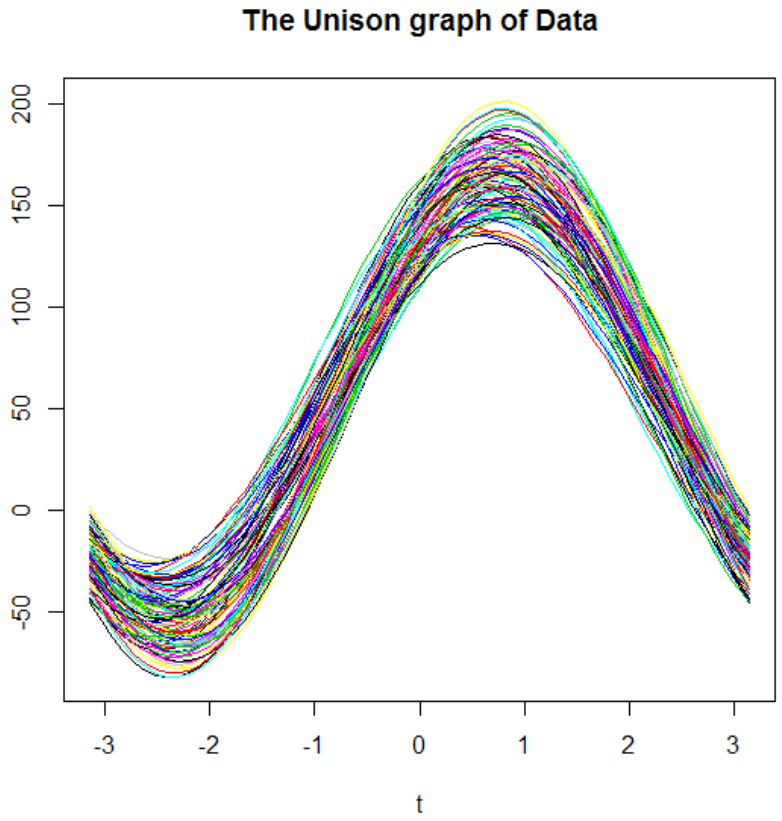
则对应的调和曲线是

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

调和曲线图

- 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])
>
```

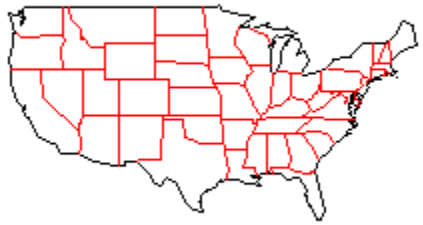


地图

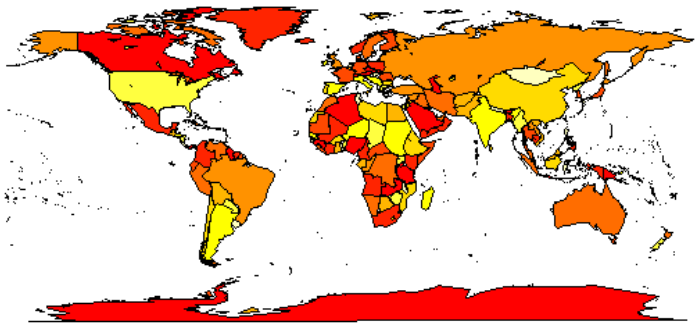
- 安装maps包

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

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



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



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)
```

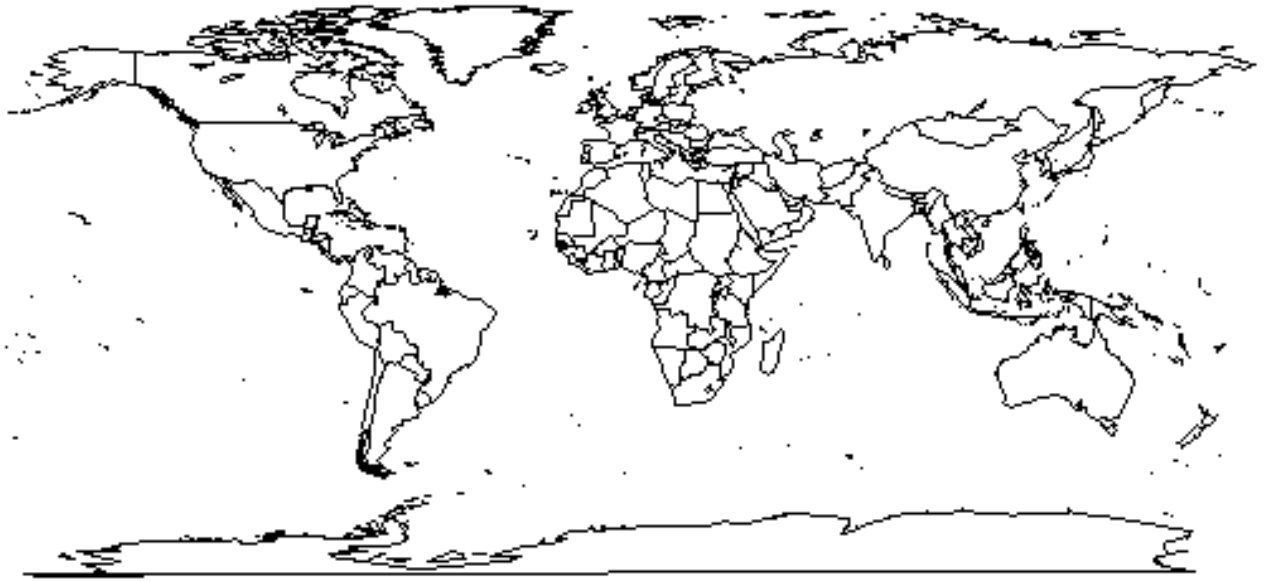
- 画出美国地图

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

R实验：社交数据可视化

- 画世界地图

map("world")



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)
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
5lat_me <- 45.21300
4lon_me <- -68.906250
inter <-
    gcIntermediate(c(lon_c
    a, lat_ca), c(lon_me,
    lat_me), n=50,
    addStartEnd=TRUE)
lines(inter)
```

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

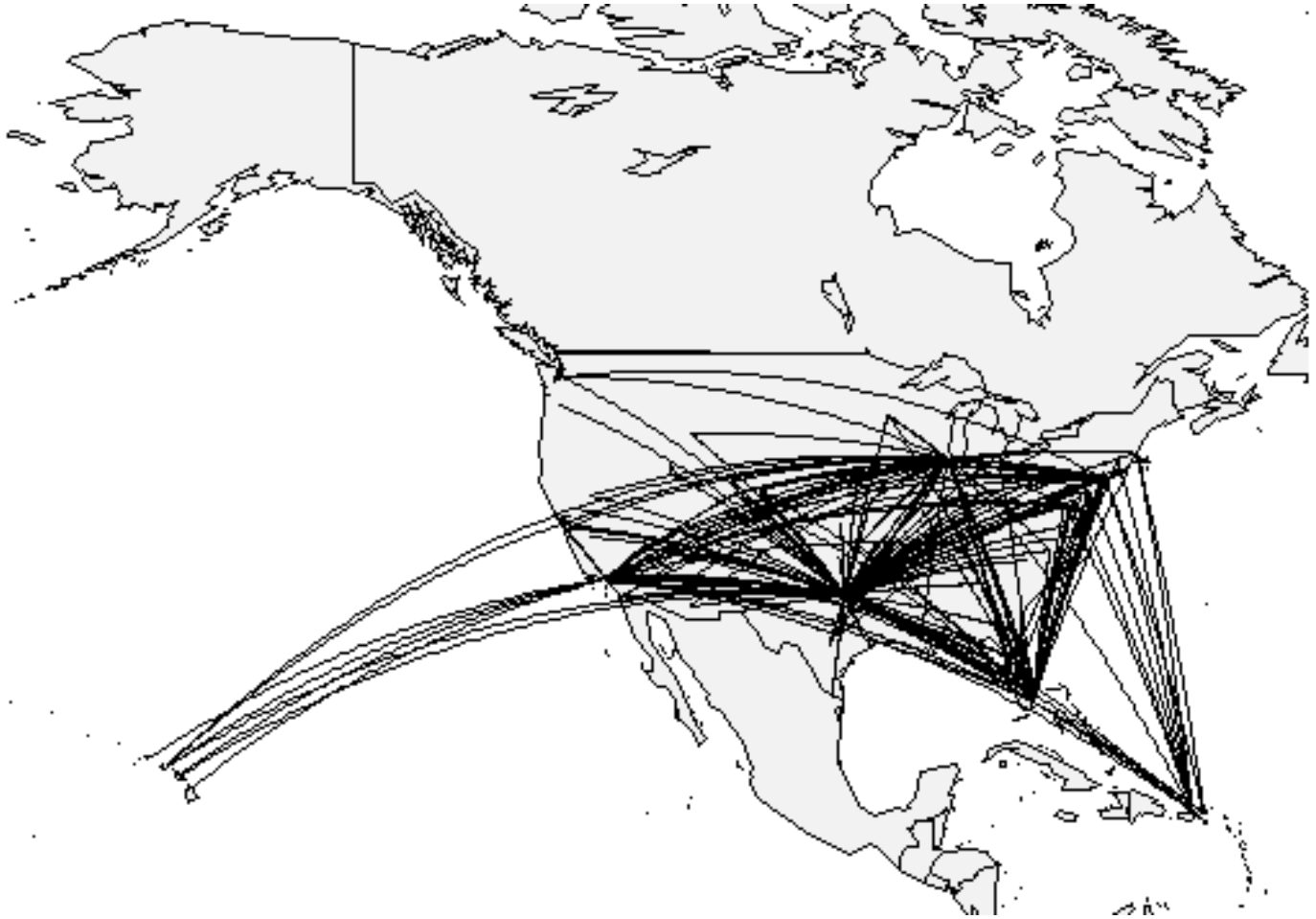
```
airports <- read.csv("http://datasets.flowingdata.com/tuts/maparcs/airports.csv",
    header=TRUE)
flights <- read.csv("http://datasets.flowingdata.com/tuts/maparcs/flights.csv",
    header=TRUE, as.is=TRUE)
```

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)

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



2013.04.20

R实验：社交数据可视化

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

炼数成金逆向收费式网络课程

- Dataguru（炼数成金）是专业数据分析网站，提供教育，媒体，内容，社区，出版，数据分析业务等服务。我们的课程采用新兴的互联网教育形式，独创地发展了逆向收费式网络培训课程模式。既继承传统教育重学习氛围，重竞争压力的特点，同时又发挥互联网的威力打破时空限制，把天南地北志同道合的朋友组织在一起交流学习，使到原先孤立的学习个体组合成有组织的探索力量。并且把原先动辄成千上万的学习成本，直线下降至百元范围，造福大众。我们的目标是：低成本传播高价值知识，构架中国第一的网上知识流转阵地。
- 关于逆向收费式网络的详情，请看我们的培训网站 http://edu.dataguru.cn

Thanks

FAQ时间