推荐书籍：

Modern Applied Statistics with S

--------Venables and Ripley

The New S Language: A Programming Environment for Data Analysis and Graphics

---------Richard A. BeckerJohn M. ChambersAllan R.Wilks

A Handbook of Statistical Analysis Using R

----------Brian S. EverittTorsten Hothorn

Data Analysis and Graphics using R

----------Maindonald and Braun

Introductory Statistics with R

----------Dalgaard

数据操作

等差数列 seq()

seq()有两个常用的参数，by 和 length。by 指定数列的公差，length 指定数列的长度。两者一般不同时使用。

```
seq(length=10,from=1,to=11)      #从 1 到 11 的 10 个数据
seq(1,10,by=2)          [1] 1 3 5 7 9
seq(1,10,length=6)      [1]  1.0  2.8  4.6  6.4  8.2 10.0
sequence(2:3)           #产生以 2 和 3 结尾的序列数据
    [1] 1 2 1 2 3
```

注意 seq() 和 sequence()的区别。

重复数列 rep()

rep(a, each=b, times=c)函数有两个常用参数，each 和 times,其中 a，b，c 可以是数值和向量。

```
rep(1:5,each=2, times=2)                #重复 1 到 5，每个元素重复二次，整个数列重复两次
    [1] 1 1 2 2 3 3 4 4 5 5 1 1 2 2 3 3 4 4 5 5
rep(a1:a2,a1:a2)          #重复 a1 到 a2，按 a1 产生 a1 次，按 a2 产生 a2 次，
    例：rep(1:3,1:3)      1 2 2 3 3 3
```

因子的产生

gl(a,b,c,labels=c())，a 表示因子有几个水平，b 表示这几个水平重复的次数，c 表示因子总的长度，labels 用来给因子标识

```
gl(2, 1, 20)
    [1] 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2
    Levels: 1 2

gl(2, 2, 20)
    [1] 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2
    Levels: 1 2
```

数值与概率计算

数值计算函数：

log(x)，exp(x)	以 e 为底的自然对数与指数函数，　log(x,10) 以 10 为底的对数
sqrt(x)	x 的平方根
abs(x)	绝对值
factorial(x),choose(n,x)	排列与组合
gamma(x)　lgamma(x)	
floor(x)，ceiling(x)，round()	向下取整，向上取整，四舍五入
round(x, digits=0)	将 x 的元素四舍五入，digits 参数表示四舍五入到小数点后第几位
trunc(x)	

signif(x, digits=6)	
cos(x),sin(x),tan(x) acos(x), asin(x), atan(x) acosh(x), asinh(x), tanh(x)	常用三角函数

常用统计函数

max(x)　min(x)　range(x)	求向量 x 最大值，最小值，同时求出最大和最小值
sum(x)　prod(x)	求向量 x 的所有元素的和与积
cumsum(x) cumprod(x)	对向量 x 的元素逐个累加求和；累乘求积，生成一个新的向量
cummax(x) cummin(x)	同上，取最大值和最小值
mean(x) median(x)	求出向量 x 的均值，中位数
quantile(x,n)	求出向量 x 的任意 n 分位数，当 n 忽略时求出 five number
var(x)　　sd(x)	求出向量 x 的方差与标准差，若 x 是矩阵或数据框则计算协方差阵
var(x,y)	求向量 x 和 y 协方差，若 x 和 y 是矩阵或数据框则计算对应列的协方差
cor(x,y)	求向量 x 和 y 线性相关系数，若 x 和 y 是矩阵或数据框则计算相关系数矩阵
rank(x)	求出向量 x 的秩统计量
sort()　rev()	排序，倒序
pmin(x,y…)　pmax(x,y)	返回一个向量，第 i 个元素是 x[i] y[i]中的最小、大值
scale(x)	若 x 是矩阵，则中心化和标准化数据，有两个逻辑型参数 scale 和 center，默认值为 TRUE，
match(x,y)	返回一个和 x 的长度相同的向量，表示 x 中与 y 中元素相同的元素在 y 中的位置
subset(x, subset, select,)	返回 x 中满足特定条件的子集，注意后面参数的用法。
table(x) table(x,y)	返回一个表格，给出 x 中重复元素的个数列表， x 于 y 的列联表

注意：exp(mean(log(x)))　求几何平均数；1/mean(1/x)　求调和平均数

colMeans(x)　colSums(x)　rowMeans(x)　rowSums(x)

常用运算符

数学运算	+ - * / ^	加 减 乘 除 乘方
	%% %/%	模 整除
比较运算	< >	小于　大于
	<= >=	小于等于　大于等于
	== !=	等于　不等于
逻辑运算	& | !	与　或　非

求偏导与积分：

```
f=expression(x^3+y^4)          #定义表达式
D(f,"x")                       #求 x 的偏导
    3 * x^2
D(D(f,"x"),"x")                #求 x 的二阶偏导
    3 * (2 * x)
f=function(x) {sqrt(x)}              #定义函数
integrate(f,1,10)                    #求 f 在 1 到 10 的积分
```

统计分析

对于同一个检验的单侧临界值，根据不同的备择假设形式使用 qfunc(0.05)或 1-qfunc(0.95)

求 p 值： 1-pchisq(3.84,1) #自由度为 1 的 chisq=3.84 的 p 值(实际是用分布函数来求 p 值) [1] 0.05004352

常用分布函数

不同的名字前缀表示不同的含义，d 表示概率密度函数，p 表示累积分布函数，q 表示分位数函数，r 表示随机模拟或随机数发生器。

贝塔分布	beta	α β ncp
二项分布	binom	n, p
Cauchy 分布	cauchy	location,scale
卡方分布	chisq	df, ncp
指数分布	exp	rate
F 分布	f	df1, df2, ncp
伽马分布	gamma	shape, scale
几何分布	geom	p
超几何分布	hyper	m, n, k
对数正态分布	lnorm	meanlog, sdlog
logistic 分布	logis	location,scale
负二项分布	nbinom	size, p
正态分布	norm	mean, sd
Poisson 分布	pois	lambda
t 分布	t	df, ncp
均匀分布	unif	min, max
Weibull 分布	weibull	shape scale
Wilcoxon 分布	wilcox	m, n

抽样 sample()

sample(x, size, replace = FALSE,prob=y) ， x 表示数据来源，size 表示抽取的多少，replace 表示是否有放回抽样。prob 表示样本单位的入样概率，y 是一与 x 等长的向量。y[i]/sum(y)表示 x[i]被抽中的概率。

```
sample(5:20,3)        #从 5 到 20 里抽取 3 个数值
[1] 12  7  14
```

特别的，当该函数只有一个参数 x 时，相当与把原数据重新随机排序。

```
sample(1:10)
[1]  3  1  9  5  4 10  7  6  8  2
sample(1:100,10,prob=1:100)        #数列 1：100 中越后面的数据越容易被抽中，是一不等概率抽样。
```

假设检验

假设检验的两个常用参数：

alternative = "two.sided"，表示双边检验，另外取值为："less", "greater"

conf.level = 0.95 数值型参数，表示置信水平。

QQ 图

qqnorm(x); qqline(x)是观察数据是否服从正态分布

```
w <- c(75.0, 64.0, 47.4, 66.9, 62.2, 62.2, 58.7, 63.5,66.6, 64.0, 57.0, 69.0, 56.9, 50.0, 72.0)
qqnorm(w); qqline(w)
```

若散点大致都在一条直线上，便可认为数据是服从正态分布的。该方法直观易懂，但不便与量化，若需要精

确结果，还应用其他方法检验。

正态性 W 检验（Shapiro－Wilk 检验）

H_0 :数据服从正态分布，H_1 :数据不服从正态分布

```
w <- c(75.0, 64.0, 47.4, 66.9, 62.2, 62.2, 58.7, 63.5,+ 66.6, 64.0, 57.0, 69.0, 56.9, 50.0, 72.0)
shapiro.test(w)
```

```
        Shapiro-Wilk normality test
data:  w
W = 0.9686, p-value = 0.8371
```

结论：p-value = 0.8371>0.05,则不能拒绝原假设，认为数据是正态分布的

T 检验（均值检验）

常见的包括单样本 t 检验，独立样本 t 检验，配对样本 t 检验。

单样本 t 检验：

```
x<-c(36,32,31,25,28,36,40,32,41,26,35,35,32,87,33,35)
t.test(x,mu=mean(x))
```

```
        One Sample t-test
data:  x
t = 0, df = 15, p-value = 1
alternative hypothesis: true mean is not equal to 36.5
95 percent confidence interval:  28.95415 44.04585
sample estimates:
mean of x      36.5
```

或者采用 t.test(x-n)　x 是一数值向量，n 是待检验的均值

```
x<-c(36,32,31,25,28,36,40,32,41,26,35,35,32,87,33,35)
t.test(x-36.5)                    #检验均值是否为 36.5
```

```
        One Sample t-test
data:  x - 36.5
t = 0, df = 15, p-value = 1
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:  -7.545853  7.545853
sample estimates:
mean of x         0
```

两者结果相同，但略有差别。

结果说明：p-value = 1>0.5 接受原假设，认为数据的均值为 36.5

注意：t 检验要求数据是服从正态分布的大样本，此例数据有异常且为小样本，t 检验失败，符号检验效果更好一些。

独立双样本 t 检验：

t.test(x,y)　x，y 是待检验的数值向量。

配对样本 t 检验

t.test(x,y,paired=TRUE),

T 检验的参数：

mu=　数值参数，表示待检验的数值。

paired = FALSE　逻辑型参数，表示是否配对。

方差检验

该检验是检验两个总体的方差是否等于 1

```
x<-c(78.1,72.4,76.2,74.3,77.4,78.4,76.0,75.5,76.7,77.3)
y<-c(79.1,81.0,77.3,79.1,80.0,79.1,79.1,77.3,80.2,82.1)
var.test(x,y)
```

```
        F test to compare two variances
data:  x and y
F = 1.4945, num df = 9, denom df = 9, p-value = 0.559
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.3712079 6.0167710
sample estimates:
ratio of variances
          1.494481
```

注意：该检验也要求数据是服从正态分布的。

符号检验

binom.test(x,n,p)　x 是成功的次数，n 是实验的次数，p 是成功的概率。在有些检验中 p 可以忽略，默认 0.5。

如在 50 次实验中成功的次数为 35 次，检验成功的概率是否为 0.5。

```
binom.test(x=35,n=50,p=0.5,alternative="greater",conf.level = 0.99)
        Exact binomial test
data:  35 and 50
number of successes = 35, number of trials = 50, p-value = 0.0033
alternative hypothesis: true probability of success is greater than 0.5
99 percent confidence interval: 0.5278131 1.0000000
sample estimates:  probability of success        0.7
```

也可用于配对样本的检验，但归根结底还是化为上述的问题，这时需要作好数据的处理工作。

参数：x 成功的次数，n 实验的总次数，p 待检验的成功概率。

Wilcoxon 符号秩检验

对中位数的检验

用于检验一组数据的中位数是否为特定的数值。

wilcox.test(x,mu=n,……) x 是数值向量，mu 是待检验的中位数，注意其他几个参数的重要性。

```
x<- c(144.3,139.1,141.7,137.3,133.5,138.2,141.1,139.2,136.5,136.5,135.6,138.0,140.9,140.6,136.3,134.1)
wilcox.test(x, mu=140, alternative="less",exact=FALSE, conf.int=TRUE)
        Wilcoxon signed rank test with continuity correction
data:  x
V = 28.5, p-value = 0.02183
alternative hypothesis: true location is less than 140
95 percent confidence interval:      -Inf 139.5500
sample estimates:(pseudo)median
          138.25
```

配对样本检验

格式：wilcox.test(x,y,alternative="",paired=TRUE) 或者 wilcox.test(x-y, alternative = "")效果相同

注意：“two-sample Wilcoxon test” 在有些书中也叫 “Mann－Whitney test” 。

卡方检验

拟合优度检验

chisp.test(x,p)x 是数值向量，一般是实际的观察次数。p 是理论频次向量，注意 p 的和应为 1。

参数 p 缺省时是默认为检验均匀分布。

```
x<-c(210,312,170,85,223)
chisq.test(x)
        Chi-squaredtestforgivenprobabilities
data:x
X-squared=136.49,df=4,p-value<2.2e-16
```

结论：p 值小于 0.05，拒绝原假设，认为数据不是来自均匀分布。

在检验数据是否服从其他形式的分布时，主要是计算理论频次的问题，同时还要注意每个理论频数应大于 5。

独立性检验

独立性检验是针对如右的列联表进行检验的。其数据形式是矩阵。

60	32	92
3	11	14
63	43	106

```
x<-c(60,3,32,11)
dim(x)<-c(2,2)
chisq.test(x)
Pearson'sChi-squaredtestwithYates'continuitycorrection
data:x
X-squared=7.9327,df=1,p-value=0.004855
```

K-S 检验

单样本 K-S 检验：

单样本 K-S 检验主要检验数据是否来自某一特定的总体。

格式：ks.test(x,"",n1,n2……)，x 是待检验的数据，""是待检验的分布，后面是该分布的参数。

```
x<-c(420,500,920,1380,1510,1650,1760,2100,2300,2350)
ks.test(x,"pexp",1/1500)#检验 x 是否服从参数为 1/1500 的指数分布。
    One-sampleKolmogorov-Smirnovtest
    data:x
    D=0.3015,p-value=0.2654
    alternativehypothesis:two-sided
```

双样本 K-S 检验

双样本 K-S 检验是检验两组数据是否来自相同分布。

格式：ks.test(x,y)x,y 均是数值型向量

Fisher 精确性独立检验

Fisher 精确性独立检验是相对卡方独立性检验而言的，并常用与 2x2 列联表的检验，尤其是数据不满足卡方检验时（单元的期望频数小于 4），精确检验比较有用。其数据形式是矩阵。具体看帮助。

方差分析

方差分析是用来判断观测到的几个样本的的均值的差异是否大到足以拒绝 H_0 的统计方法，其数据应满足正态性、独立性和方差齐性的要求，所以在进行方差分析前应先检验数据是否满足条件。

单因素方差分析

aov(x~f)，x 是数值向量，来自与个水平实验结果，f 是因子，对应各种水平。

```
x=c(1600, 1610, 1650, 1680, 1700, 1700, 1780, 1500, 1640,1400, 1700, 1750, 1640, 1550, 1600, 1620, 1640, 1600,1740, 1800, 1510, 1520, 1530, 1570, 1640, 1600)
A=factor(c(rep(1,7),rep(2,5), rep(3,8), rep(4,6)))
lamp.aov<-aov(x~A)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
A	3	49212	16404	2.1659	0.1208
Residuals	22	166622	7574		

如果数据来自数据框，则用参数 data 指明数据来源即可。如下：

```
lamp<-data.frame(x,A)
lamp.aov<-aov(x ~ A, data=lamp)
summary(lamp.aov)
```

双因素方差分析

如右表，对其做方差分析，判断 A 和 B 因素对实验的结果影响。

	B1	B2	B3

x=c(325, 292, 316, 317, 310, 318,310, 320, 318, 330, 370, 365)

A=gl(4,3)

B=gl(3,1,12)

result.aov<-aov(x~A+B)

summary(result.aov)

A1	325	292	316
A2	317	310	318
A3	310	320	318
A4	330	370	365

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
A	3	3824.2	1274.7	5.2262	0.04126 *
B	2	162.5	81.2	0.3331	0.72915
Residuals	6	1463.5	243.9		

如果数据来自数据框，如下：

y<-data.frame(x,A,B)

summary(aov(x~A+B,data=y))

有交互作用的方差分析

如右表所示，检验是否有交互作用的存在。

	B1	B2	B3	B4
A1	23 25 21 14 15	20 17 11 26 21	16 19 13 16 24	20 21 18 27 24
A2	28 30 19 17 22	26 24 21 25 26	19 18 19 20 25	26 26 28 29 23
A3	18 15 23 18 10	21 25 12 12 22	19 23 22 14 13	22 13 12 22 19

x=c(23, 25, 21, 14, 15, 20, 17, 11, 26, 21,16, 19, 13, 16, 24, 20, 21, 18, 27, 24,28, 30, 19, 17, 22, 26, 24, 21, 25, 26,19, 18, 19, 20, 25, 26, 26, 28, 29, 23,18, 15, 23, 18, 10, 21, 25, 12, 12, 22,19, 23, 22, 14, 13, 22, 13, 12, 22, 19)

A=gl(3,20,60)

B=gl(4,5,60)

result.aov<-aov(x~A+B+A:B)

summary(result.aov)

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
A	2	352.53	176.27	8.9589	0.000494 ***
B	3	87.52	29.17	1.4827	0.231077
A:B	6	71.73	11.96	0.6077	0.722890
Residuals	48	944.40	19.67		

如果数据来自数据框，如下：

y<-data.frame(x,A,B)

summary(aov(x~A+B+A*B,data=y))

注意：方差分析一定要产生正确的因子 A 和 B，把数据转化成数据框的形式，然后输出可以粗略的判断因子是否正确。

问题：若拒绝了原假设，如何进一步了解哪些均值是相等的，哪些是不等的。

回归分析

一元线形回归

1）lm()函数的应用

例如：建立以下数据的回归模型

x<-c(0.1,0.11,0.12,0.13,0.14,0.15,0.16,0.17,0.18,0.20,0.21,0.23)

y<-c(42,43.5,45,45.5,45,47.5,49,53,50,55,55,60)

lm.sol<-lm(y~x)

Call:

lm(formula = y ~ x)

Coefficients:

```
(Intercept)                x
      28.49           130.83
```

```
summary(lm.sol)
```

```
Call:    lm(formula = y ~ x)
Residuals:
     Min       1Q   Median       3Q      Max
 -2.0431  -0.7056   0.1694   0.6633   2.2653
Coefficients:
             Estimate   Std. Error   t value   Pr(>|t|)
(Intercept)   28.493      1.580       18.04    5.88e-09 ***
x            130.835      9.683       13.51    9.50e-08 ***
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 1.319 on 10 degrees of freedom
Multiple R-Squared: 0.9481,     Adjusted R-squared: 0.9429
F-statistic: 182.6 on 1 and 10 DF,  p-value: 9.505e-08
```

结果分析：建立的为一元线形回归，最终结果为

$$y = 28.49 + 130.83x$$
$$\text{std}\quad (1.58)\qquad (9.683)$$
$$\text{t.value}\quad (18.04)\qquad (13.51)$$
$$\text{Adjusted R-squared: } 0.9429 \qquad F=182.6$$

从结果可以看出，系数和 F 的 p 值均小于 0.05，说明都通过了检验。注意：输出结果中的 Std. Error 是标准差，不是标准误。

预测：模型的预测使用 predict()函数，如下：

```
new=data.frame(x<-0.16)
pred<-predict(lm.sol,new,interval="prediction",level=0.95)
pred
```

```
          fit       lwr      upr
[1,] 49.42639  46.36621 52.48657
```

该函数的第一个参数表示模型的名称，第二个函数表示用于预测的自变量的值，注意：即使只有一个数，也要写成数据框的形式，第三个参数表示给出预测区间，第四个参数表示置信水平为 0.95,该参数可不写，默认值即为 0.95。

最终结果为 Y(0.16) = 49.43, [46.37, 52.49].

残差分析：residuals(lm.sol)函数可以得到模型的残差，参数是模型的名称。

```
y.res<-residuals(lm.sol);plot(x,y.res)        # 得到残差并作出残差和自变量的散点图
```

lm()参数解释：该函数是线形回归模型基本函数，格式为：lm.sol<- lm(formula, data = data.frame)，表示模型结果存放在 lm.sol 中。

formula 为模型的表达式，

y~x 或 y~1+x 表示一元有截距项的回归模型；

y~0 + x 表示没有常数项

y~x1+x2+… 表示多元回归；

y~x1 +I(x1^n) 表示高次回归

y ~ x1+x2 + x1:x2 表示有交叉项，或者 y~x1*x2 注意：（a*b=a+b+a:b）

data.frame 为数据框，表示表达式中变量数据的来源，

其他参数：如 weights 表示数据拟合的权重；method 表示？？？

2）lm()结果信息提取函数

lm()返回的结果过于简单，需要用其他函数提取模型的信息：

anova(lm.name)；返回模型的方差分析表，lm.name 表示 lm 方法得到模型的名称；下同。

coef(lm.name)：返回模型的系数；

deviance(lm.name)：返回模型的残差平方和；

fitted(lm.name)：返回每个自变量的估计值；

formula(lm.name)：返回模型的表达式；

plot(lm.name)：返回模型的几种诊断图形；但其结果是一张一张出现，最好把它画到一张图上。

predict()：用于模型预测，见上；

residuals(lm.name)；用于得到模型的残差；

summary(lm.name)；得到模型的概述信息；

step(lm.name)：返回模型的逐步回归结果；

kappa(lm.name)：返回一个数值，判断模型是否存在多重共线性。

其他相关函数：effects　alias　family　labels　proj

多元线形回归

和一元回归类似，表达式的差别，同时注意数据的输入，

例：

```
blood<-data.frame(x1=c(76.0, 91.5, 85.5, 82.5, 79.0, 80.5, 74.5,79.0, 85.0, 76.5, 82.0, 95.0, 92.5),
    x2=c(50, 20, 20,30, 30, 50, 60, 50, 40, 55,40, 40, 20),
    y= c(120, 141, 124, 126, 117, 125, 123, 125,132, 123, 132, 155, 147))
lm.sol<-lm(y~x1+x2,data=blood);sol
```

```
Call:  lm(formula = y ~ x1 + x2, data = blood)
Coefficients:
(Intercept)          x1          x2
   -62.9634     2.1366     0.4002
```

逐步回归

step()函数:　lm.step<-step(lm.sol)

将 lm.sol 回归后的模型进行逐步回归，并将结果赋值与 lm.step。

```
cement<-data.frame(x1=c( 7, 1, 11, 11, 7, 11, 3, 1, 2, 21, 1, 11, 10),
    x2=c(26, 29, 56, 31, 52, 55, 71, 31, 54, 47, 40, 66, 68),
    x3=c( 6, 15, 8, 8, 6, 9, 17, 22, 18, 4, 23, 9, 8),
    x4=c(60, 52, 20, 47, 33, 22, 6, 44, 22, 26, 34, 12, 12),
    y =c(78.5, 74.3, 104.3, 87.6, 95.9, 109.2, 102.7, 72.5,93.1,115.9, 83.8, 113.3, 109.4))
lm.sol<-lm(y~x1+x2+x3+x4,data=cement)
lm.step<-step(lm.sol)
```

其他逐步函数：add1()　drop1()　　使用方法类似，具体看帮助

回归诊断

常用函数：influence.measures　rstandard　rstudent　dffits　cooks.distance　dfbeta　dfbetas　covratio　hatvalues　hat

多重共线性

主要有两个问题：判断模型是否存在多重共线性，若有，则是哪几个变量造成的。

一）kappa()函数计算 X^TX 的条件数来判断是否存在多重共线性。

若 k<100，　　认为多重共线性程度较小；

若 100<k<1000，认为多重共线性程度中等；

若 k>1000，　　认为多重共线性程度严重

```
collinear<-data.frame(x1=rep(c(8, 0, 2, 0), c(3, 3, 3, 3)),
    x2=rep(c(1, 0, 7, 0), c(3, 3, 3, 3)),
    x3=rep(c(1, 9, 0), c(3, 3, 6)),
```

```
    x4=rep(c(1, 0, 1, 10), c(1, 2, 6, 3)),
    x5=c(0.541, 0.130, 2.116, -2.397, -0.046, 0.365,1.996, 0.228, 1.38, -0.798, 0.257, 0.440),
    x6=c(-0.099, 0.070, 0.115, 0.252, 0.017, 1.504,-0.865, -0.055, 0.502, -0.399, 0.101, 0.432),
    y=c(10.006, 9.737, 15.087, 8.422, 8.625, 16.289,5.958, 9.313, 12.960, 5.541, 8.756, 10.937))
XX<-cor(collinear[2:7])                    #计算协方差矩阵
kappa(XX,exact=TRUE)                       #计算条件数，exact 表示精确计算，否则为近似计算。
        [1] 2195.908
```

k=2195>1000， 认为存在严重的多重共线性。

二）eigen()函数计算协方差矩阵的特征根和特征向量，并以此判断多重共线性是由哪几个变量引起的。

```
    eigen(XX)
```

输出一个向量和一个矩阵，从中可以看到 $\lambda_{min}=0.001106,\varphi = (0.4476,0.4211,0.5417,0.5734,0.006052,0.002167)^T$ 特征值选取的是最后一列，（为什么？）0.4476x1 + 0.4211x2 + 0.5417x3 + 0.5734x4+ 0.006052x5 + 0.002167x6 ≈ 0。x5 和 x6 的系数近似为零，忽略，则可以看出多重共线性是有前四个变量引起的。

绘图

绘图函数的常用参数

add= F　　逻辑型参数，若为 T，表示在原图的添加图形。

type=""　p 点，l 线，b 点线，h 竖直线，n 空白，o 将点覆盖在线上，s 和 S 步阶图。

xlab=""　ylab=""　字符型参数，用于显示特定的 x 和 y 轴的标签。

xlim=c(0,10), ylim=c( ) 指定轴的上下限。

main=""　字符型参数，图形标题，以大字体显示于图形顶部。

sub=""　字符型参数，子标题，以小字体显示于 x 轴下部。

col=" "：指定图形颜色。col.axis=" " col.lab=" " col.main=" "col.sub=" "指定相关元素的颜色。其取值是颜色的代码，可由 colors()得到，共 657 种取值。

色彩的处理

1）colors()函数

colors()：即可得到该函数的 657 个取值，这些值常被给 bg, col 等属性赋值。

```
    par(bg="mistyrose")          #指定作图的背景
    plot(1:10,1:10,type="l",col="mediumblue")      #绘出彩色的线
```

2）颜色渐变函数

```
    x=runif(10000,-250,250)       #后面要使用的数据
```

rainbow()函数指定七彩虹颜色：红橙黄绿蓝靛紫，

```
    hist(x,breaks=seq(-250,250,5),col=rainbow(100))
```

heat.colors()函数用来指定颜色由红变到橙再到白，

```
    hist(x,breaks=seq(-250,250,5),col=heat.colors(100))
```

terrain.colors()函数用来指定颜色由绿色变到棕色再到白色，

```
    hist(x,breaks=seq(-250,250,5),col=terrain.colors(100))
```

topo.colors()函数用来指定颜色由蓝色变到绿色再到白色，

```
    hist(x,breaks=seq(-250,250,5),col=topo.colors(100))
```

cm.colors()函数，

```
     hist(x,breaks=seq(-250,250,5),col=cm.colors(100))
```

3）hcl()与 hsv()函数

hcl(h, c, l , alpha, fixup = TRUE)：其中 h 指定颜色，为 0～360 间的整数，0 为红色，120 为绿色，240 为蓝色；c 指定浓度；l 指定亮度；alpha 指定透明度。

```
    hist(x,breaks=seq(-250,250,5),col=hcl(1:100))
```

hsv()

hist(x,breaks=seq(-250,250,5),col=hsv(seq(0,1,0.01)))

4) 灰度函数 grey() 或者 gray()

grey(level)：level 指定灰度，为 0～1 间的数，0 为黑，1 为白。

hist(x,breaks=seq(-250,250,5),col=grey(seq(0,1,0.01)))

5）rgb()函数

该函数按三元色，红，绿，蓝原理指定颜色，rgb(r=1,g=0,b=0)为红色，rgb(r=0,g=1,b=0)为绿色，rgb(r=0,g=0,b=1)为蓝色。

```
hist(x,breaks=seq(-250,250,5),col=rgb(r=1,g=0,b=0))          #该方法生成的图象色彩单调
hist(x,breaks=seq(-250,250,5),col=rgb((0:100)/100, g=0,b=0))  #此方法的图象有渐变效果
```

plot()函数

plot()函数的用法

1>绘制散点图和时间序列图

plot(x,y)　plot(x)

2>plot(f)f 是因子，生成 f 的直方图，

plot(f,y)f 是因子，y 是数值向量，生成 y 关于 f 水平的箱线图。

```
y<-
c(1600,1610,1650,1680,1700,1700,1780,1500,1640,1400,1700,1750,1640,1550,1600,1620,1640,1600,1740,1800,1510,1520,1530,1570,1640,1600)
f<-factor(c(rep(1,7),rep(2,5),rep(3,8),rep(4,6)))
plot(f,y)
```

解释：y 是一数值向量，f 把此数值向量分为四个水平。结果绘出四个箱线图，直观反映各水平的对比。

如果数据来自数据框，采用以下方式可以达到同样的效果。如下：

```
A=data.frame(y,f)
plot(y~f,data=A,col=heat.colors(4))
```

barplot()条形图

简单条形图

```
x<-rpois(100,5)
y<-table(x)          #将 x 表格化
r<-barplot(y,col=rainbow(20))
lines(r, y, type='h', col='red', lwd=2)          #在条形图的中间加上一条直线
```

注意：必须把产生的向量 x 表格化才可以做条形图，可以比较 barplot(x,col=rainbow(20))两者的效果是不同的。而表格化后的条形图才是我们需要的。

数据表的产生：如要产生如下表格：

1	2	3	4	5	6	7	8	9	10
3	13	18	15	24	12	6	4	4	1

```
x<-rep(1:10,c(3,13,18,15,24,12,6,4,4,1))
y<-table(x)  ; y
```

```
1  2  3  4  5  6  7  8  9  10
3  13 18 15  24 12  6  4  4  1
```

如果 rep 的第一个参数是字符向量，那作出的条形图下方是该类别的文字表述，如下例：

```
b<-table(rep(c("A","B","C","D"),c(4,9,13,7)))
barplot(b,col=c("tomato1","turquoise", "violetred4" , "yellow1"))
```

累计条形图与对比条形图

如右表是 A、 B、 C 三个公司前四个月的销售额，

	A	B	C
一月	9	4	6
二月	6	9	9
三月	7	13	14
四月	15	7	8

```
a<-c(9,6,7,15,4,9,13,7,6,9,14,8)
b<-matrix(a,4,3,dimnames=list(c("一月","二月","三月","四月"),c("A","B","C")))          #产生如图的矩阵
```

```
barplot(b)                #画出各公司的累计条形图
barplot(b,beside=T,ylim=c(0,50),legend=c("一月","二月","三月","四月"),names.arg=c("A 公司","B 公司","C 公司"))
                          #分类画出各公司逐月条形图
```

总之，如果 barplot 的画图对象是一矩阵，那图形是以列为分类，行为各条高度的条形图，

beside 是一逻辑值，缺省为 F，画出累计条形图，取 T 时画出分类对比条形图

legend 和 names.arg 的取值和结果如上例

使用如下方法可以在条形图每个 bar 上方显示相应的数值：

```
x<-1:10;names(x)<-letters[1:10]          #产生向量，并对向量赋名
b<-barplot(x,col=rev(heat.colors(10)))   #得到每个 bar 的横坐标并绘出条形图
text(b,x,labels=x,pos=3)
```

pie()饼图

```
pie(y,col = rainbow(10),labels="")
```

该例沿用条形图数据，注意第一个参数取值不同时图的意义也不同，若是数据表，则表示每一类数据所占比例，若为数值向量，则按照 y[i]/sum(y)的比例绘出饼图。labels 参数为空值时表示没有说明标签，缺省时默认用第一个参数来标识。

stem()茎叶图

```
stem(y,scale=2)
```

y 是一数值向量，scale 用于确定茎的宽度，该参数可省略，若要详细的茎叶图，可将数值调大。

boxplot()箱线图

```
boxplot(x)          #x 是一数值向量，绘出 x 的简单箱线图
boxplot(x,y)        # x，y 均为数值向量，在一个图中绘出 x 和 y 的箱线图。
boxplot(x~f)        #x 是数值向量，f 是相应的因子，绘出各因子下的对应箱线图
```

hist()直方图

```
x=c(29.6,28.2,19.6,13.7,13.0,7.8,3.4,2.0,1.9,1.0,0.7,0.4,0.4,0.3,0.3,0.3,0.3,0.3,0.3,0.2,0.2,0.2,0.1,0.1,0.1,0.1,0.1)
hist(x)    #绘出频数图
```

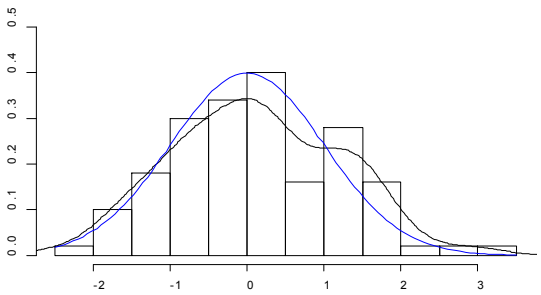
常用参数：

probability：逻辑变量，默认为 F，表示绘频数图。取值为 T 时表示绘出频率图。

breaks：设置断点，该参数可以调整直方图的宽度，看帮助。

在同一个图中画出频率直方图和概率密度图：如右图

```
x <- rnorm(100,0,1)
hist(x,freq=F,main="直方图和密度曲线")
lines(density(x))          # 绘出根据样本得到的密度曲线
curve(dnorm(x),add=T,col='blue')      # ？？？什么意思
```



persp()三维图象的绘制

绘制 $\sqrt{x^2+y^2}$ 的函数图象：程序如下

```
y<-x<-seq(-10,10,length=30)
f<-function(x,y){sqrt(x^2+y^2)}
z<-outer(x,y,f)
persp(x,y,z,theta=30,phi=30,expand=0.7,col="lightblue")
```

注意：z 并不是 x 和 y 的简单运算，而是要在函数 f 关系下作外积运算(outer(x,y,f))，形成网格，才能绘出三维图形。

par 函数

该函数对全局设置更改，采用以下方法可以恢复设置的更改：

```
opar <- par()          #在设置前把所有默认的设置赋值到 opar 变量中。
```

par(opar)　　#用刚才的设置恢复到默认水平。

ask=TRUE：在绘制多图时一个一个显示，回车键控制

add=TRUE　表示原图不变，在原图添加图形，缺省为 add=FALSE。

bg=""　fg=""：指定图片的背景色、前景色

new=TRUE：这个函数可以在绘第二个图时保留第一个图的绘图区域，效果就是两张图合并，用这种方法也可以做双坐标图。

线条，点的相关参数

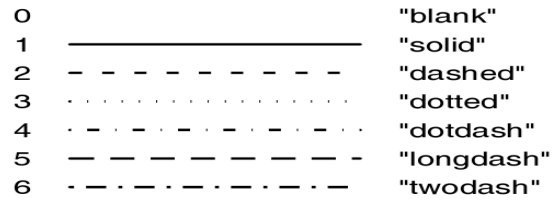
bty="o" 表示在绘图区域显示边框，（边框和字符的形状很像）

l（字母）、7（数字）表示不显示上、右；下、左边框。

]，[表示不显示左、右边边框。

u 表示不显示上边框

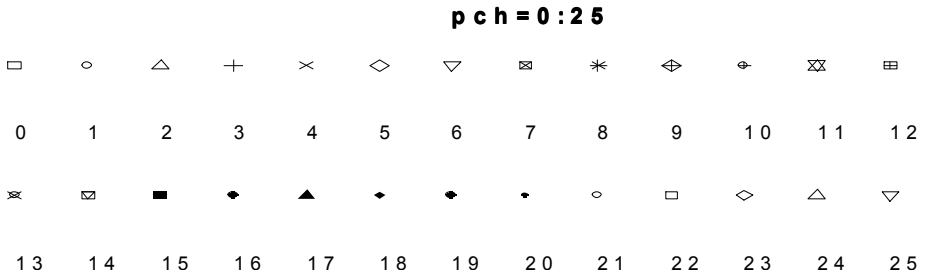
lty=　：设置线的形式，可以用数值或字符赋值，0=blank, 1=solid, 2=dashed, 3=dotted, 4=dotdash, 5=longdash, 6=twodash，但要注意两种数据赋值的区别。感觉用 type 更方便一些。



lwd=1：设置线的宽度，数值型参数。默认为 1，数字越大，线条越宽。

pch　该参数可以用数值赋值，如下。也可以用字符赋值，如 pch="%"

legend(locator(1), as.character(0:25), pch = 0:25)　#显示 pch 所有取值的图示



坐标轴：

ann=TRUE,　表示画出坐标轴的标签。若为 F，表示不画坐标轴的标签。和 xlab，ylab 赋空值效果相同。

xaxs="r" yaxs="r"　表示两个坐标轴的交点不是 0，设置为 i 时交点为 0。

xaxt="s" yaxt="s"　表示显示坐标轴刻度，设置为 n 时表示不显示坐标轴刻度。

axes=T　逻辑型参数，表示有坐标轴边框，若为 F，表示无坐标轴和边框。

axis() 是坐标轴控制函数。可以用这个函数绘制双坐标图。

文字：

adj=0.5，控制文字的对齐方式，0 是左对齐，1 是右对齐。

cex=1　控制符号和文字大小的值，cex.axis,cex.lab,cex.main,cex.sub 是对应坐标轴刻度数字，坐标轴标签文字，标题，副标题文字大小。

font=1，控制文字字体的整数，1 正常，2 斜体，3 粗体，4 粗斜体，font.axis font.lab font.main font.sub 同上。

ps=12，控制文字大小的整数，单位为磅。

绘图区域分割

mfrow=c(3, 2) 将绘图区域分割成 3 行 2 列，每块显示一个图形。按行显示。

mfcol=c(3, 2)　将绘图区域分割成 3 行 2 列，每块显示一个图形。按列显示。

以上两个是 par 函数的参数，缺点是只能将绘图区域分割成规则的区域，以下两个函数可以将绘图区域分割成不规则的形状。

layout()　　#能产生什么样的矩阵，就可以把绘图区域随意分割。

layout(matrix(1:4, 2, 2))　　#该函数的参数是一个矩阵，该语句把绘图区域分成两行两列四个小块。

layout(matrix(c(1:3, 3), 2, 2))　　#该语句生成三个不规则的区域，可以用以下语句查看分割情况。

layout.show(3)

widths=c(1, 3),heights=c(3, 1)，函数默认是等距分割，可以使用这两个参数调节长宽比例。

split.screen()

低水平作图函数

points(x,y)　x 和 y 为数值向量，添加(x,y)……等这样的点。

lines(x,y)　x 和 y 为数值向量，该函数连接 x 和 y 对应的点所成的直线，特别 x 和 y 为一维向量时，是线段 。

segments(x1, y1,x2, y2)　绘出(x1,y2)到(x2,y2)的线段。

arrows(x1, y1,x2, y2, angle= 30,code=2)

绘出(x1,y2)到(x2,y2)的箭头，angle 表示箭头的角度。code=c(1,2,3)表示起点、终点、两端绘箭头。

text(x,y,"")　x 和 y 为数值向量，text(3,4,"?")，表示在(3,4)处添加一个问号。

abline()

1)abline(a,b)　表示绘出一条 y=a+bx 的直线

2)abline(h=x)　表示画出过向量 x 所有点的水平线

3)abline(v=x)　表示画出过向量 x 所有点的竖直线

4)abline(lm.obj)　表示画出线形模型的拟合直线

polygon(x,y)　x 和 y 为数值向量，该函数逐次连接（x,y），绘出一多边形

rect(x1,y1,x2,y2)　以(x1,y1)为左下点，以(x2,y2)为右上点绘制矩形

locator(n,type="n", ...)　返回用户 n 次点击的坐标，type 参数的取值不同可以直接在图上画出相应图形

图例注解函数：

mtext()　This function draws text at any location in any of the margins.

legend(x1,y1,legend=c("","","""……),col=c(",","……),lty=c(2,-1,1),pch=c(-1,3,4),merge=TRUE)

```
x<-1:100
z<-rnorm(100)
y<-rnorm(100)
plot(x,y,xlab='',ylab='',pch=1,col='red',main='图例注解函数示意图')
points(x,z,pch=16,col='blue')
legend(locator(1),legend=c('Y','Z'),col=c('red','blue'),pch=c(1,16))
```

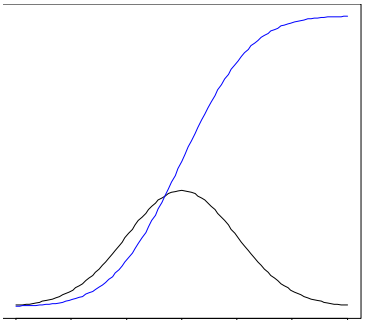
x1,y1　设置显示的位置的坐标，最好是用 locator(1)，然后在图形上点击

legend　设置图解图形解释的文字

bty = "o"设置图例边框，默认为 o，有边框，取值为"n"，表示为无边框。

merge=TRUE　合并点和线，如果两者都有，默认为 T

curve 函数



该函数用于绘制曲线图形，一个简单的例子，curve(x^2,-3,3)

在概率与统计中的应用：

在一张图中作出的密度与分布函数图：如右

curve(dnorm(x), xlim=c(-3,3),ylim=c(0,1),main="正态分布密度函数和分布函数图",xlab="",ylab="") # 密度函数

curve(pnorm(x), xlim=c(-3,3),ylim=c(0,1), col='blue',xlab="",ylab="",add=T)　# 分布函数

分位数函数图：curve(qnorm(x), xlim=c(0,1))

卡方分布自由度为 1：5 的分布函数图：

```
color<-c('black','red','green','blue','orange')
```

```
adds<-c(FALSE,TRUE,TRUE,TRUE,TRUE)
for (i in 1:5)      {
   curve(dchisq(x,i), xlim=c(0,10), ylim=c(0,0.6), add=adds[i],col=color[i], lwd=2)}
```

绘制密度函数与分布函数图

```
x<-c(1,2,3,4,4,5,6,7,10,3,4,5)                #定义数据
plot(density(x) )                             #密度函数
plot(ecdf(x) )                                #经验分布函数
```

条件和循环

条件语句：

```
if(.....) {.....}
   else {.....}
```

在赋值中的应用：x <- if(.....) 3.14 else 2.71

或者使用 x <- ifelse(....., 3.14, 2.71)

for 循环

```
for (i in 1:10) {    .....
   if(.....) { next }    .....
   if(.....) { break }    .....
 }
```

while 循环

```
while(.....) {
 .....
 }
```

repeat 循环

```
repeat {
   .....
   if(.....) { break }
   .....
   }
```

Option()函数

options(digits=n)，　　digits 参数设置输出结果的小数位数。

options(prompt="R> ")，prompt 参数更改系统提示符为 R> ，

杂项

对象

两个属性，类型和长度

```
mode(x)                    #返回 x 的类型
length(x)                  #返回 x 的长度
```

对工作目录的操作

```
getwd()                    #得到 r 的工作目录
    [1]"D:/ProgramFiles/R/R-2.6.1"
setwd("F:/")               #重新设置 r 的工作目录
```

或者右键点击 R 的桌面快捷方式，在出现的菜单选择‘属性'，弹出对话框，点击“快捷方式"，可以看到“起始位置"，填上你的目标文件夹就可以了。

清除工作空间所有对象

```
rm(list=ls(all=T))      #清除全部对象
rm(list=ls())
```

对包的操作

```
library()                  #查看本机所有的包
    library(package)       #加载特定的包
search()                   #查看目前加载的包
detach(package:base)       #卸载 base 包
library(help=package)      #显示某个包中所有可用的函数
```

获取帮助

```
?qt
help(qt)
help.start()
help.search("covariance")
```

得到函数的代码

直接在 R 的工作平台输入函数的名字，不要带括号，可以得到大部分函数的代码。

问题

如何在向量的中间插入一个数据？

```
x<-c(1:4,6:10);x           #定义数据
    [1]1 2 3 4 6 7 8 9 10
xbefore<-x[1:4]            #把前面的数据提取出来
xbehing<-x[5:9]            #把后面的数据提取出来
xnew<-c(xbefore,5,xbehing) #添加数据，合并
xnew
    [1]1 2 3 4 5 6 7 8 9 10
```

如何剔除向量中特定的元素？

```
x<-c(1:10);x               #定义数据
    [1]1 2 3 4 5 6 7 8 9 10
xnew<-x[-5]                #剔除特定数据并赋值给新变量
xnew
    [1]1 2 3 4 6 7 8 9 10
```

如何更改向量中特定的数据？实质是将数据赋值给特定的数据。

```
x
    [1]1 2 3 4 5 6 7 8 9 10
x[5]<-50
x
    [1]1 2 3 4 50 6 7 8 9 10
```

显示特定条件的元素？

```
x[c(1,3,5,7,9)]#显示特定的元素
x[x>3&x<5]#显示某范围的元素
```

无穷大的处理

Inf 表示无穷大，注意：第一个字母大写。

```
(0:3)^Inf
    [1]0 1 Inf Inf
```

连接若干点得到平滑曲线

如果已知做出这些点的函数可以用 curve(expr, from, to, add = T)函数，反之，使用立方曲线差值函数 spline(x, y, n=)，如：

```
x<-1:5;y<-c(1,3,4,2.5,2)
plot(x,y,main='立方曲线差值函数 spline()示意图')
sp<-spline(x,y,n=50)
lines(sp)
```