

统计编程的框架 与 R 语言统计分析基础

(北京地区第一次COS沙龙活动讲稿)

中国人民大学统计学院

谢益辉

2007年5月13日

目 录

一、道冲，而用之或不盈.....	4
1. 为什么需要统计编程?	5
2. 哪里需要编程?	6
3. 从一般计算机编程说起.....	10
4. 数值计算与统计编程.....	12
5. 统计编程的流程.....	16
二、欲善其事，先利其器.....	17
1. 什么是利器?	18

2. 为什么用R?	19
3. 谁不适合用R?	20
三、一箪食，一瓢饮，在陋巷.....	21
1. 获取与安装.....	23
2. R数据结构.....	26
3. R统计分析.....	55
4. R图形.....	76

一、道冲，而用之或不盈

道冲，而用之或不盈。渊兮，似万物之宗。挫其锐，解其纷，和其光，同其尘。湛兮，似或存。吾不知谁之子，象帝之先。

——老子

1. 为什么需要统计编程？

□ 统计人的责任：了解算法的黑箱（我不入黑箱谁入黑箱？）

◆ 黑箱：忽悠的源泉（你快回来，我一人忽悠不来！）

□ 统计人的自由：灵活计算与输出（我的地盘我做主！）

◆ 我的地盘儿，我也不知道听谁的。

◆ 我的地盘儿，听你的。

□ 统计人的愿景：走在时代的前列（直挂云帆济沧海！）

◆ 吃罢早饭吃中饭，吃罢中饭吃晚饭，晚饭吃过困觉哉，困觉起来吃早饭。

2. 哪里需要编程？

- 没错，很多情况下确实不需要（即便是用R也不需要）
- 但是，无论如何你的GUI都不可能把所有的方法都囊括到菜单和按钮中的
 - ◆ 正态性的检验有多少种方法？Jarque-Bera 检验
 - ◆ SPSS 做一个 t 分布的 KS 检验？（只有正态、泊松、指数等）
 - ◆ 前沿：Copulas? Machine Learning? ...
 - ◆ 等等...

Jarque-Bera 检验

$$JB = \frac{n}{6} \left(S^2 + \frac{(K - 3)^2}{4} \right),$$

$$S = \frac{\mu_3}{\sigma^3} = \frac{\mu_3}{(\sigma^2)^{3/2}} = \frac{\frac{1}{n} \sum_{i=1}^n (x - \bar{x})^3}{\left(\frac{1}{n} \sum_{i=1}^n (x - \bar{x})^2 \right)^{3/2}}$$

$$K = \frac{\mu_4}{\sigma^4} = \frac{\mu_4}{(\sigma^2)^2} = \frac{\frac{1}{n} \sum_{i=1}^n (x - \bar{x})^4}{\left(\frac{1}{n} \sum_{i=1}^n (x - \bar{x})^2 \right)^2}$$

JB 统计量渐近服从 $\chi^2(2)$ 分布

一个随意的初级 R 程序:

```
n=length(x)
S=mean((x-mean(x))^3)/((n-1)/n*var(x))^1.5
K=mean((x-mean(x))^4)/((n-1)/n*var(x))^2
chi=n/6*(S^2+(K-3)^2/4)
pvalue=1-pchisq(chi,2)
```

R 源代码 (tseries 包; *jarque.bera.test()* 函数)

```
> jarque.bera.test
function (x)
{
  if (NCOL(x) > 1)
    stop("x is not a vector or univariate time series")
  if (any(is.na(x)))
    stop("NAs in x")
  DNAME <- deparse(substitute(x))
```



```

n <- length(x)
m1 <- sum(x)/n
m2 <- sum((x - m1)^2)/n
m3 <- sum((x - m1)^3)/n
m4 <- sum((x - m1)^4)/n
b1 <- (m3/m2^(3/2))^2
b2 <- (m4/m2^2)
STATISTIC <- n * b1/6 + n * (b2 - 3)^2/24
names(STATISTIC) <- "X-squared"
PARAMETER <- 2
names(PARAMETER) <- "df"
PVAL <- 1 - pchisq(STATISTIC, df = 2)
METHOD <- "Jarque Bera Test"
structure(list(statistic = STATISTIC, parameter = PARAMETER,
  p.value = PVAL, method = METHOD, data.name = DNAME),
  class = "htest")
}

```

3. 从一般计算机编程说起

程序设计三要素:

- 数据结构

 - ◆ 数值与字符

 - ◆ 结构型数据（数组等）

- 算法

 - ◆ 控制流

- 语言环境

```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Yihui Xie>目录
'目录'不是内部或外部命令，也不是可运行的程序
或批处理文件。

C:\Documents and Settings\Yihui Xie>给我列出目录！
'给我列出目录！'不是内部或外部命令，也不是可运行的程序
或批处理文件。

C:\Documents and Settings\Yihui Xie>丫给我列出目录听见没有？
'丫给我列出目录听见没有？'不是内部或外部命令，也不是可运行的程序
或批处理文件。

C:\Documents and Settings\Yihui Xie>dir
驱动器 C 中的卷没有标签。
卷的序列号是 8095-6676

C:\Documents and Settings\Yihui Xie 的目录

2007-05-09 13:29 <DIR>      .
2007-05-09 13:29 <DIR>      ..
2007-04-28 13:04           160 .appletviewer
2007-05-07 11:57 <DIR>      .autosave
谷歌拼音 半:
```

4. 数值计算与统计编程

数值计算的基本内容:

- 拟合与插值
- 数值积分
- 线性方程组求解
- 矩阵的计算（特征根、矩阵分解等）
- 非线性方程（组）求解

数值计算的学习有什么好处？

□ 训练编程思维（学完之后会“忆苦思甜”）

□ 统计中常遇到的问题

◆ 求根、优化问题（如：GLM、SEM的极大似然估计）

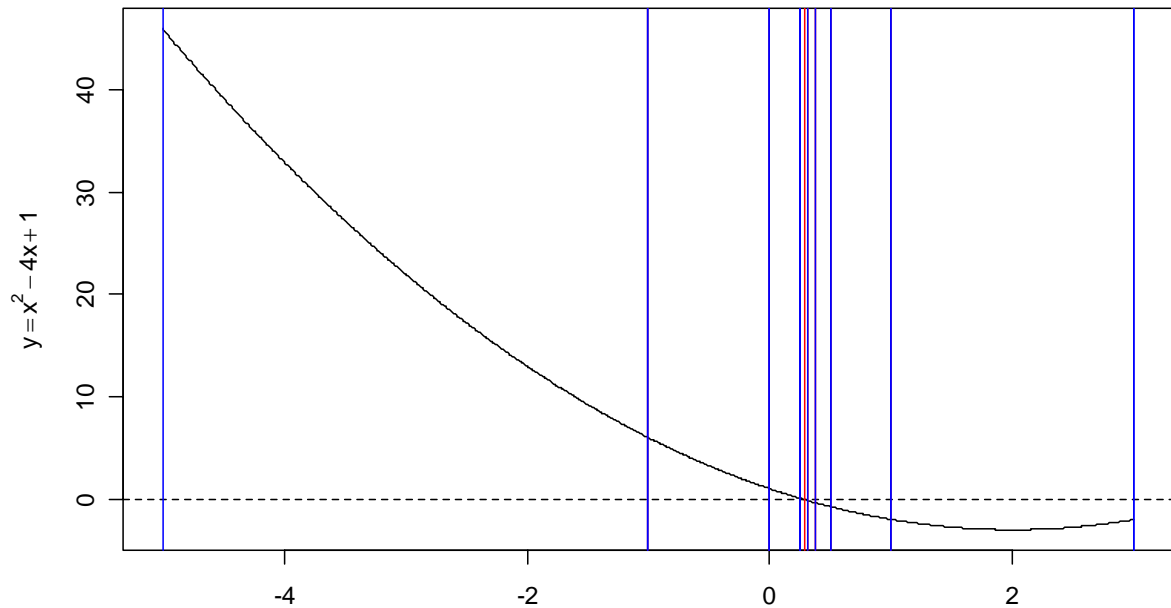
◆ 矩阵求逆、行列式（如：回归 $\hat{\beta} = (X'X)^{-1}X'y$ ）

◆ 特征根、特征向量（如：因子分析）

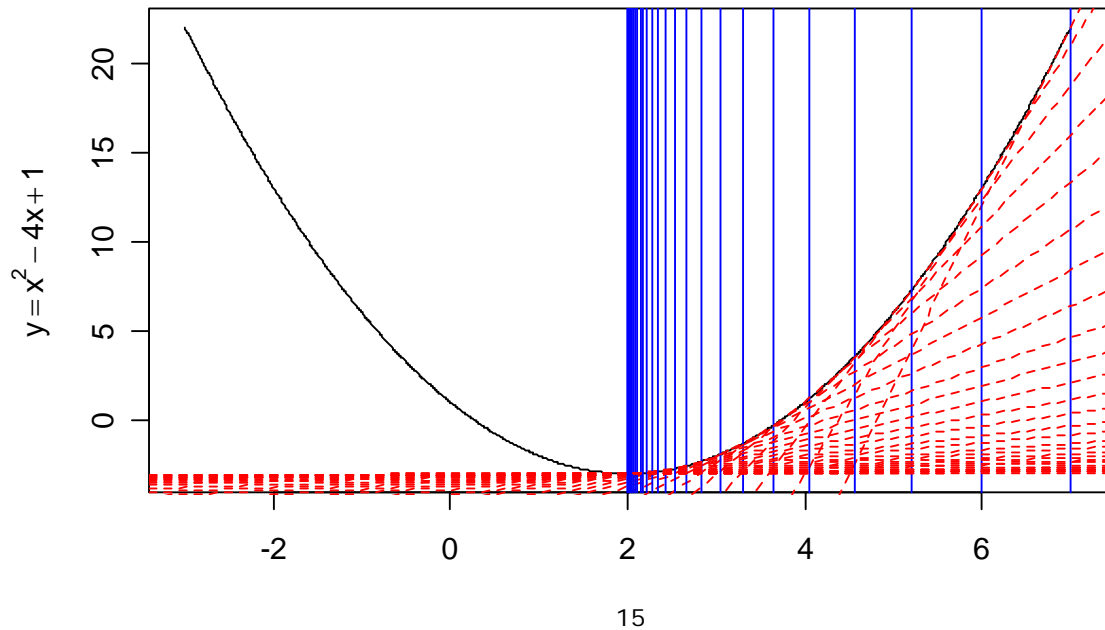
◆ 复杂的（非线性）积分

数值计算的地位？统计人应该在了解它的基础上追求效率！

一元方程求根的演示



一元函数寻找最小值的演示



5. 统计编程的流程

- 统计理论是怎样的？（问题的起源）
- 数学的计算步骤是？（这是求解的关键，必须清楚告诉计算机怎么走）
- 选择怎样的工具？（速度、便利性等考虑）

二、欲善其事，先利其器



.....

1. 什么是利器？

优秀的软件

- 功能齐全
- 编程方便
- 高度模块化
- 技术支持力量强
- 小巧、速度快
- 透明

2. 为什么用 R?

- 以上提到的标准基本都满足
- R 作为解释性语言，在速度上自然会逊于 C、Fortran 等“底层”语言，但是 R 可以与其它语言相互调用
- 对于统计编程来说，R 具有其它软件无法比拟的便捷性（统计的数据结构、统计的函数、统计的图形）
- 当然，我们希望软件能便宜一些，而 R 的价格是 \$0（就凭这点，其它软件都不好意思出来混）

3. 谁不适合用 R?

我从来不提倡所有人都干同一件事，因为人各有别。

- 英语不好，或对英语有着天生的痛恨，热爱中文

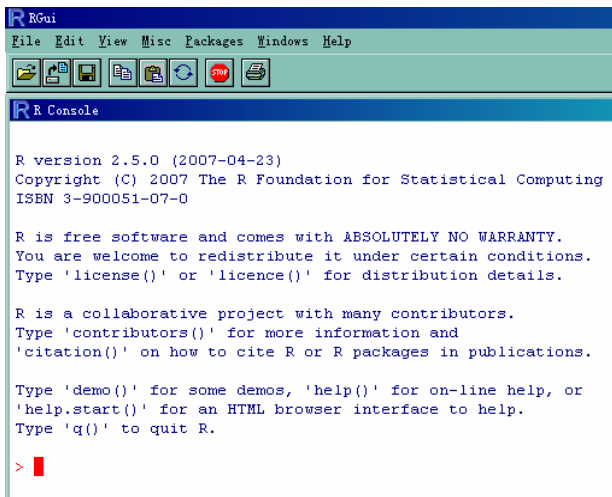
- 对计算机代码有着天生的痛恨、热爱鼠标

 - ◆ 该知足了，我们这个时代不用打孔机和纸带跑程序了

- 统计知识一窍不通

 - ◆ 不知道什么是分布、什么是 P 值、什么是假设检验

三、一箪食，一瓢饮，在陋巷



```
RGui
File Edit View Misc Packages Windows Help
R Console

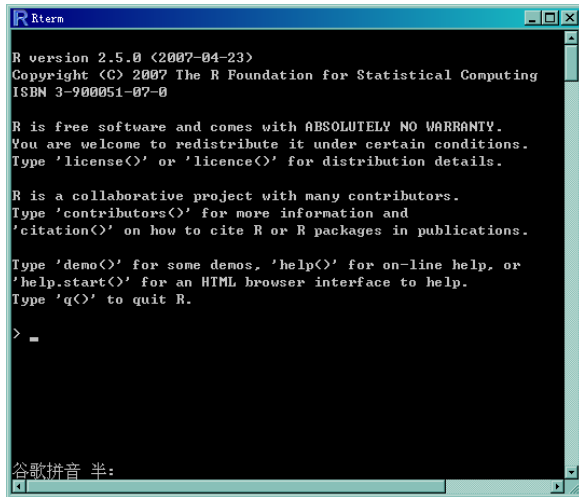
R version 2.5.0 (2007-04-23)
Copyright (C) 2007 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> █
```



```
Rterm
R version 2.5.0 (2007-04-23)
Copyright (C) 2007 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> █

谷歌拼音 半:
```

子曰：“贤哉，回也！一簞食，一瓢饮，在陋巷。人不堪其忧，回也不改其乐。贤哉，回也！”

《论语·雍也篇第六·第十一章》

1. 获取与安装

先看两位 R 的创始者



Ross Ihaka



Robert Gentleman

R 诞生于 the University of Auckland 的统计系

官方网站: <http://www.r-project.org>

下载: CRAN → 选择镜像 (如: <http://cran.cnr.berkeley.edu/>)

→ 选择操作系统 (Linux、Windows 或 MacOS)

以 Windows 为例: 选择 base (基础系统), 进去之后点击

R-*.*.*-win32.exe 这样的链接 (*.*. *表示版本) 下载安装程序

注意: 安装过程中尽量不要选择 message translation 选项 (它表示安装“翻译版”(包含中文等语言)的 R, 但某些 Windows

系统会因这个选项而无法启动 R), **更不要安装在中文目录下!**

附加包的安装

`install.packages(packagename, dependencies = TRUE)`

Windows 下可以用菜单 Packages → Install package(s) 安装

版本的更新

- ❑ 主程序: Windows 下面只能卸载再安装
- ❑ 包: `update.packages()`

2. R 数据结构

在 R 中，我们一直都在与对象（object）打交道。

基础类型（mode）

- ❑ 实数型（real）：整数（integer）、单精度（single）、双精度（double）
- ❑ 虚数型（complex）：如 $9 + 11i$
- ❑ 字符型（character, string）：如 "hello"（单双引号都行）

- ❑ 逻辑型 (logical): TRUE, FALSE (简写 T, F)
- ❑ 函数 (function)
- ❑ 表达式 (expression)

结构化数据

- ❑ 向量 (vector): 一系列数值或字符
- ❑ 矩阵 (matrix): m 行 \times n 列 (各列之间类型都相同)
- ❑ 数据框 (data frame): 类似矩阵, 但每一列的数据

类型可以不同

- ❑ 数组 (array): 多维度 (不是多变量)
- ❑ 列表 (list): 有诸多成员杂合在一起, 这些成员可以是任意类型, 甚至是 list 本身 (及其灵活的数据类型)
- ❑ 因子 (factor): 分类变量
- ❑ 时间序列 (ts): 时间序列数据

使用变量的时候要特别注意, R 对大小写敏感!

产生数据

简单的规则序列

```
> 1:10 # 井号是 R 的注释符号
```

```
[1] 1 2 3 4 5 6 7 8 9 10
```

```
> 10:1
```

```
[1] 10 9 8 7 6 5 4 3 2 1
```

```
> seq(1, 10, 0.5) # 等差数列
```

```
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5
```

```
[11] 6.0 6.5 7.0 7.5 8.0 8.5 9.0 9.5 10.0
```

```
> rep(2, 5) # 重复相同的对象
```

```
[1] 2 2 2 2 2
```

```
> rep(1:3, times = 3) # 观察与下例的不同
```

```
[1] 1 2 3 1 2 3 1 2 3
```

```
> rep(1:3, each = 3)
```

```
[1] 1 1 1 2 2 2 3 3 3
```

```
> rep(1:3, 1:3)
```

```
[1] 1 2 2 3 3 3
```

产生结构化数据

向量很容易用函数 `c()` 产生:

```
> x = c(9, 1, 1)
```

```
> x
```

```
[1] 9 1 1
```

```
> (x = c('Xie', 'Yi', 'Hui')) # 为什么用括号😊
```

```
[1] "Xie" "Yi" "Hui"
```

矩阵用 `matrix()` 产生:

```
> matrix(1:10, 2)           # 注意: 默认按列排列
```

```
      [,1] [,2] [,3] [,4] [,5]
```

```
[1,]     1     3     5     7     9
```

```
[2,]     2     4     6     8    10
```

```
> matrix(1:10, nrow = 2, ncol = 5, byrow = T)
```

```
      [,1] [,2] [,3] [,4] [,5]
```



```
[1,] 1 2 3 4 5
[2,] 6 7 8 9 10
```

数据框用 `data.frame()` 产生

```
> x = data.frame(1:5, 4:8) # 把若干个向量合成数据框
```

```
> x
```

```
  X1.5 X4.8
```

```
1     1     4
```

```
2     2     5
```

3 3 6

4 4 7

5 5 8

```
> x = cbind(x, c('A', 'B', 'C', 'D', 'E')) # 绑上一列字符
```

```
> x
```

```
 X1.5 X4.8 c("A", "B", "C", "D", "E")
```

```
1 1 4 A
```

```
2 2 5 B
```

3 3 6

C

4 4 7

D

5 5 8

E

```
> dimnames(x)
```

看一下 x 的行列名

```
[[1]]
```

```
[1] "1" "2" "3" "4" "5"
```

```
[[2]]
```

```
[1] "X1.5"
```

```
[2] "X4.8"
```

```
[3] "c(\"A\", \"B\", \"C\", \"D\", \"E\")"
```

```
> colnames(x)           # 只看列名
```

```
[1] "X1.5"
```

```
[2] "X4.8"
```

```
[3] "c(\"A\", \"B\", \"C\", \"D\", \"E\")"
```

```
> colnames(x) = c('X1', 'X2', 'X3') # 改列名
```

```
> x # 这样看起来就舒服多了
```

```
  X1 X2 X3
```

```
1  1  4  A
```

```
2  2  5  B
```

```
3  3  6  C
```

```
4  4  7  D
```

```
5  5  8  E
```

因子用 `factor()` 产生

列表用 `list()` 产生

时间序列用 `ts()` 产生

数据还可以从外部文件读入，甚至是剪贴板（clipboard），
或者 SQL、Access 数据库（RODBC 包）

例：D:\x.txt 文件

```
"V1" "V2" "V3" "V4" "V5"
```

1 5 9 13 17

2 6 10 14 18

3 7 11 15 19

4 8 12 16 20

```
> x = read.table('d:\\x.txt', header = T)
```

```
> x
```

```
  V1 V2 V3 V4 V5
```

```
1  1  5  9 13 17
```

2 2 6 10 14 18

3 3 7 11 15 19

4 4 8 12 16 20

运算

算术运算: $+$, $-$, $*$, $/$, $\% \%$ (余数), $\% / \%$ (整数商), $^$
(乘方)

$> 5 \% \% 2$

[1] 1

> 5%/2

[1] 2

> 2^5

[1] 32

逻辑运算: &, |, ! (且、或、非); >, <, >=, <=, == (“小于一个负数”如何表示? 注意“<-”是赋值符号, x<-9 与

$x=9$ 等价! 正确写法是加上空格 $x < -9$ 或者 $x < (-9)$

[http://statist.spaces.live.com/blog/cns!64B6368534A3BF2C!257.](http://statist.spaces.live.com/blog/cns!64B6368534A3BF2C!257)

entry

```
> T & F
```

```
[1] FALSE
```

```
> T | F
```

```
[1] TRUE
```

```
> !T
```

```
[1] FALSE
```

```
> 1 == T
```

```
[1] TRUE
```

```
> 2 == T
```

```
[1] FALSE
```

```
> 0 == F
```

```
[1] TRUE
```

```
> 1:10 > 5
```

```
[1] FALSE FALSE FALSE FALSE FALSE TRUE TRUE  
TRUE TRUE TRUE
```

下标的使用（获取元素）

向量、因子、时间序列 $x[i]$ ；矩阵 $x[i, j]$ $x[i,]$ $x[, j]$ ，数据框同矩阵；数组就是根据维度多打几个逗号而已 $x[i, j, k, \dots]$ ；列表要用双重中括号 $x[[i]]$

特殊的使用：在中括号中用**逻辑表达式**提取元素或给特定元素赋值

```
> x[x > 5]      # 提出大于5的元素
```

```
[1] 6 7 8 9 10
```

```
> x[x > 5 & x < 9]    # 双重条件
```

```
[1] 6 7 8
```

```
> x = matrix(1:20, 4, 5)
```

```
> x[x >= 2 & x < 16]
```

```
[1]  2  3  4  5  6  7  8  9 10 11 12 13 14 15
```

```
> x
```

```
     [,1] [,2] [,3] [,4] [,5]
```

```
[1,]    1    5    9   13   17
```

```
[2,] 2 6 10 14 18
```

```
[3,] 3 7 11 15 19
```

```
[4,] 4 8 12 16 20
```

```
> x[x >= 2 & x < 16] = NA
```

```
> x
```

```
      [,1] [,2] [,3] [,4] [,5]
```

```
[1,] 1 NA NA NA 17
```

```
[2,] NA NA NA NA 18
```

[3,] NA NA NA NA 19

[4,] NA NA NA 16 20

一些数学和统计函数

- 最大值 `max()`, 最小值 `min()`, 均值 `mean()`, 标准差 `sd()`, 方差 `var()`, 相关系数 `cor()`, 求和 `sum()`, 积 `prod()`, 中位数 `median()`, 分位数 `quantile()`, 对数 `log()`, 指数 `exp()`, 排列 `factorial()`, 组合 `choose()`, 四舍五入 `round()`, 向下

取整 `floor()`, 向上取整 `ceiling()`, 总结 `summary()`, ...

- ❑ 累加 `cumsum()`, 秩 `rank()`, 排序 `sort()`, 倒序 `rev()`, 矩阵转置 `t()`, 逆矩阵 `solve()`, 特征根 `eigen()`, ...
- ❑ 关于统计分布的四大金刚: `pnorm()`, `dnorm()`, `qnorm()`, `rnorm()`, ... (p, d, q, r+分布名称分别构成: 分布函数值、密度函数值、分位数、随机数, 如 `pf()`表示 F 分布函数值, `runif()`表示产生均匀分布的随机数); 抽样 `sample()`, ...
- ❑ 线性模型 `lm()`, 广义线性模型 `glm()`, t 检验 `t.test()`...

程序控制流：告诉计算机怎么走

□ 选择控制

◆ if (条件) {怎样怎样}

◆ if (条件) {
 怎样怎样} else {
 又怎样}

◆ ifelse()函数：ifelse(条件, 满足则取我, 不满足则取我)

◆ switch()函数：多条件选择（至少我很少用它）

□ 循环

◆ for (循环变量 in 一个序列中) {怎样怎样}

◆ while (某条件满足则) {怎样怎样}

例:

```
> cond = T
```

```
> if (cond) winDialog('ok', 'Hello Kitty!')
```

```
> x = 0
```

```
> for (i in 10:1) {
```

```
+     x = x + i
```

```
+ }
```

```
> x
```

```
[1] 55
```

```
> while (x >= 1 & winDialog("yesno", "I'm a little  
pig...") != 'YES') {
```

```
+     x = x - 1
```

```
+ }
```

这个例子比较整人.....

Tips: 括号怎么打？小括号一般用于函数，中括号获取元素，大括号把一段完整语句“保护起来”（如果只有一句话，通常大括号可用可不用）

至此，我们有了数据结构，知道了怎么控制我们的程序，那么，编程的问题基本已经解决。剩下的工作就是适应 R 的语言环境（这是一个积累的过程，勿急躁）

国有国法，家有家规。不要对你的计算机想当然。

几个小例子：

[http://statist.spaces.live.com/blog/cns!64B6368534A3BF2C!175.
entry](http://statist.spaces.live.com/blog/cns!64B6368534A3BF2C!175.entry)

$x = 9$ 不是逻辑运算； $0 < x < 9$ 不合语法； $9x$ 不是计算乘法！

先掌握基础，再在实践中摸打滚爬。

最后一点建议

□ 用帮助！ 用帮助！ 用帮助！

◆ ？

◆ `help.search()`

◆ 看文档

3. R 统计分析

在开始“编程”之前

- 问题：R 需要编程吗？答曰：大部分情况下对大部分人来说，答案是“不”
- R 中的函数太多了，包太多了（仅靠 base 和 stats 已经可以做很多事情了）
- “编程”和“写代码”完全是两码事

(1) 线性回归

```
x = runif(100); y = 0.2*x + 0.1*rnorm(100)
```

```
fit = lm(y ~ x)
```

```
summary(fit)
```

Call:

```
lm(formula = y ~ x)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.230127	-0.067896	0.007706	0.054087	0.249110

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.009639	0.019638	0.491	0.625
x	0.193976	0.032711	5.930	4.55e-08 ***

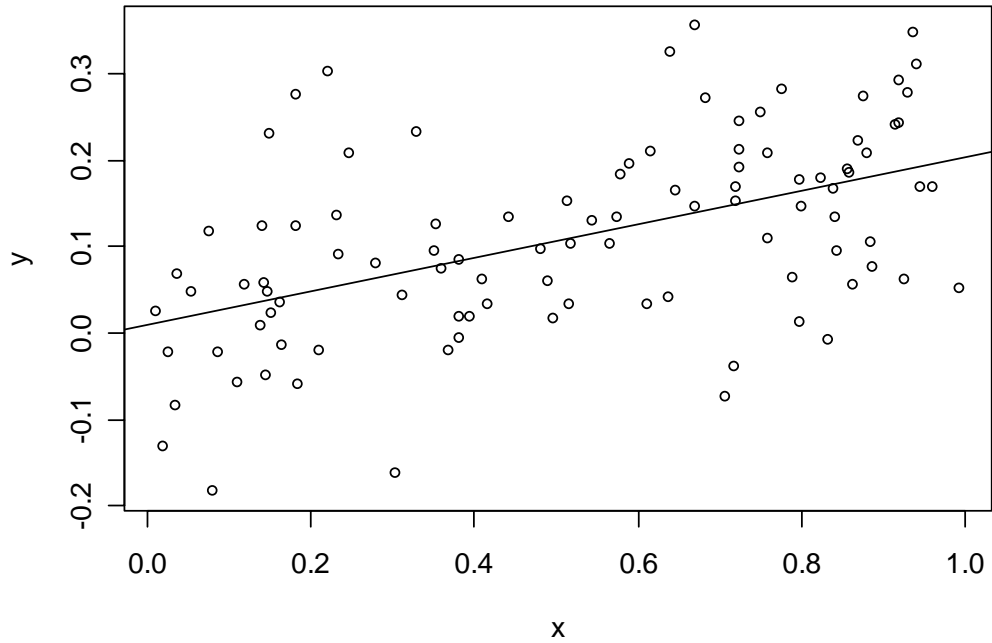
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0979 on 98 degrees of freedom

Multiple R-Squared: 0.2641, Adjusted R-squared: 0.2566

F-statistic: 35.17 on 1 and 98 DF, p-value: 4.548e-08

`plot(x, y); abline(fit)`



```
fit = lm(y ~ x + I(x^2))    # 高次的回归，注意 I()  
summary(fit)
```

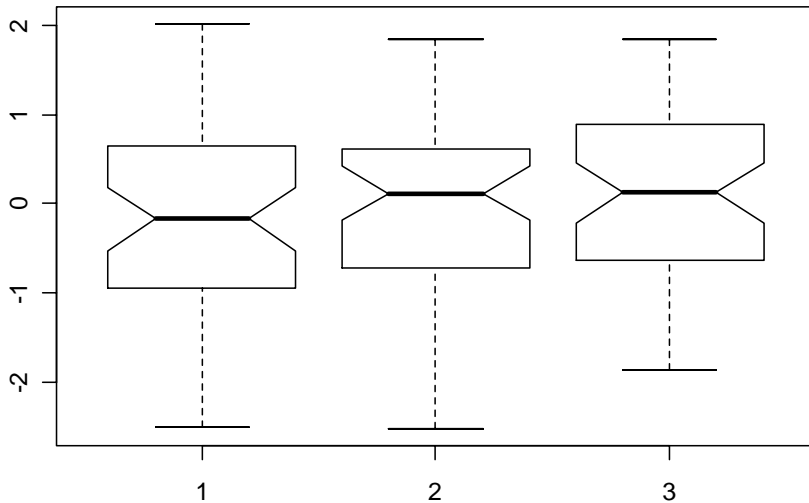
(2) 方差分析

```
x = gl(3, 50); y = rnorm(150)  
summary(aov(y ~ x))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
x	2	1.743	0.872	0.8053	0.4489

Residuals 147 159.092 1.082

boxplot(y ~ x, notch = T)



(3) 位置参数的检验

参数方法 t 检验: `t.test()`

非参数方法: `wilcox.test()`, `kruskal.test()`, 等等

条件检验: `coin` 包 (Conditional Inference)

(4) 广义线性模型

三个核心概念:

□ 因变量: 指数分布族

$$f(y_i; \theta_i, \varphi) = \exp[A_i\{y_i\theta_i - \gamma(\theta_i)\} / \varphi + \tau(y_i, \varphi / A_i)]$$

□ 自变量的线性组合 $\eta = \beta_1 x_1 + \cdots + \beta_p x_p$

□ 连接函数 (link function): 用来建立因变量的分布参数与 η 之间的连接

用得很广泛的所谓 logistic 回归只不过是 GLM 的特例: 分布族为二项分布、连接函数取 logit 函数。普通的回归也是 GLM 的特例: 分布族为正态分布、连接函数为 $f(x) = x$

R 函数: `glm(formula, family = gaussian, data, ...)`

Dobson (1990) Page 93: Randomized Controlled Trial :

```
counts <- c(18,17,15,20,10,20,25,13,12)
```

```
outcome <- gl(3,1,9)
```

```
treatment <- gl(3,3)
```

```
print(d.AD <- data.frame(treatment, outcome, counts))
```

```
  treatment outcome counts
```

```
1          1         1    18
```

```
2          1         2    17
```


3	1	3	15
4	2	1	20
5	2	2	10
6	2	3	20
7	3	1	25
8	3	2	13
9	3	3	12

summary(glm.D93)

Call:

```
glm(formula = counts ~ outcome + treatment, family = poisson())
```

Deviance Residuals:

1	2	3	4	5	6
-0.67125	0.96272	-0.16965	-0.21999	-0.95552	1.04939
7	8	9			

0.84715 -0.09167 -0.96656

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	3.045e+00	1.709e-01	17.815	<2e-16 ***
outcome2	-4.543e-01	2.022e-01	-2.247	0.0246 *
outcome3	-2.930e-01	1.927e-01	-1.520	0.1285
treatment2	8.717e-16	2.000e-01	4.36e-15	1.0000

treatment3 4.557e-16 2.000e-01 2.28e-15 1.0000

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 10.5814 on 8 degrees of freedom

Residual deviance: 5.1291 on 4 degrees of freedom

AIC: 56.761

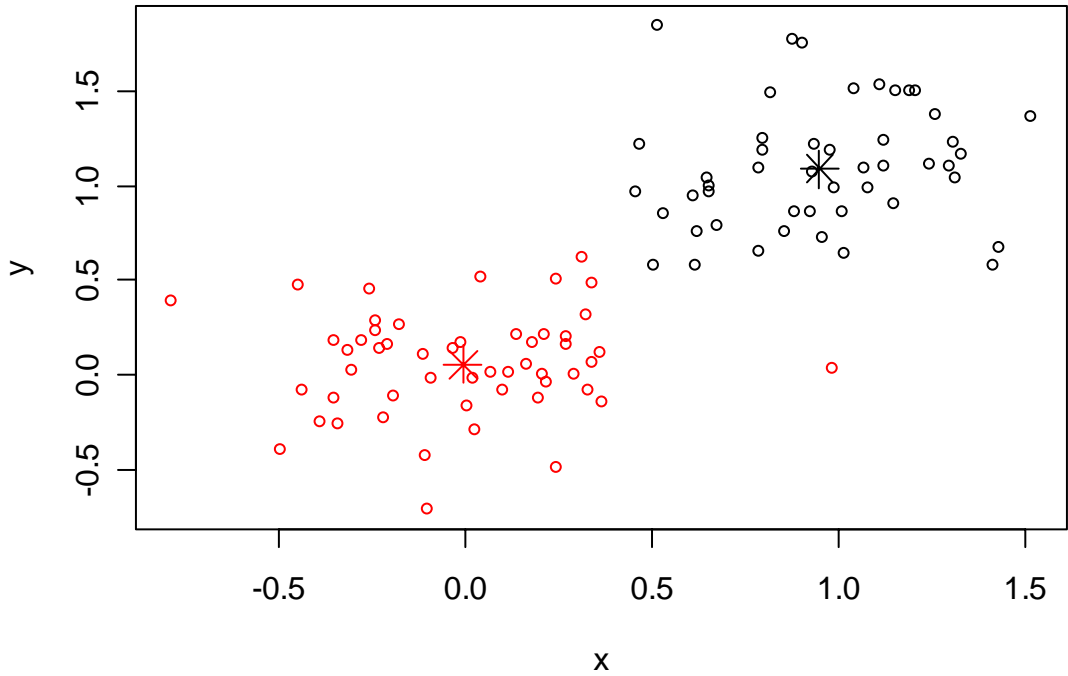
Number of Fisher Scoring iterations: 4

(5) 聚类分析

K-Means 聚类: `kmeans()`

层次聚类: `hclust()`

专做聚类的包: `cluster` (里面还有很多种聚类方法)



(6) 列联表的独立性检验

一个有用的函数: `table()`

```
> table(rpois(100, 5), rbinom(100, 1, .5))
```

```
      0  1
0     0  2
1     1  2
2     4  4
3    10  5
```

4 5 10

5 7 8

6 8 7

7 3 7

8 7 0

9 6 0

10 2 1

11 0 1

Fisher 精确检验: `fisher.test()`

卡方检验: `chisq.test()`

```
x = matrix(c(12, 5, 7, 7), nc = 2)
```

```
chisq.test(x)
```

Pearson's Chi-squared test with Yates' continuity correction

data: x

X-squared = 0.6411, df = 1, p-value = 0.4233

(7) 混合效应模型

nlme 包

lme4 包

(8) 分位数回归

quantreg 包

线性分位数回归函数 `rq()`

(9) 等等等等……

□ 根据自己的研究需要，可以自己写一点代码，或者用别人的包。

□ 怎么知道该用什么包呢？——读书！如：

A Handbook of Statistical Analyses Using R

Mixed-Effects Models in S and S-Plus

…

4. R 图形

- R 是枯燥的吗？当然不是，你可以用它做很多千奇百怪、匪夷所思的东西——看你的创造力如何。
- 图形由什么构成？点、线、形状、颜色、文本、坐标轴、图例。这一切，在 R 中都有相应的底层函数可以完成。
- 那么，还有什么作不出来的图？当然没有了。

本学期我正极其头大地写一本关于 R 作图的书，困难在哪里？在于统计理论。

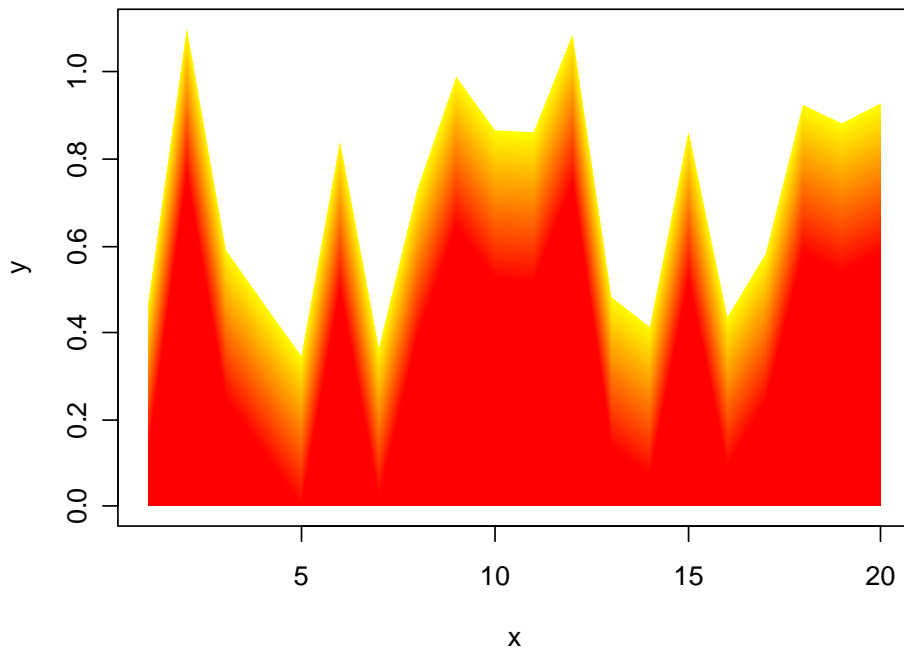
自己回去花两天时间把 graphics 包的所有函数都看一遍，也就到了境界了。

演示几个我自己编的小例子：

火海刀山？

```
y=.1+runif(20,.2,1)
```

```
xx=c(1,1:20,20)
yy=c(0,y,0)
plot(xx, yy,type='n',xlab='x',ylab='y')
for (i in 255:0){
  yy=c(0,y-(1-i/255)*min(y),0)
  polygon(xx,yy,col=rgb(1,i/255,0),border=NA)
  Sys.sleep(.05)
}
```



时钟?

```
x=seq(0,2*pi,.01)
```

```
par(pty='s')
```

```
plot(sin(x),cos(x),type='l')
```

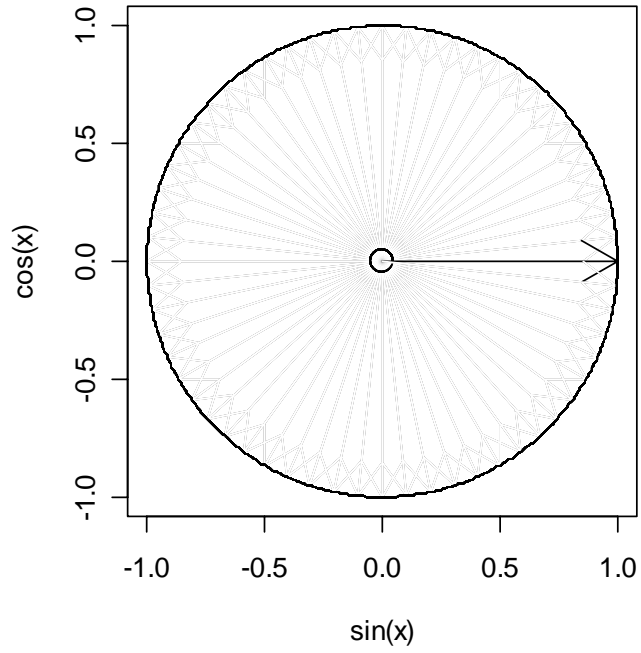
```
n=60
```

```
for (i in 1:n){
```

```
  points(0,0,cex=2)
```



```
    arrows(0,0,cos(2*pi/n*(i-1)),sin(2*pi/n*(i-1)),col='white')
arrows(0,0,cos(2*pi/n*i),sin(2*pi/n*i))
Sys.sleep(1)
arrows(0,0,cos(2*pi/n*(i-1)),sin(2*pi/n*(i-1)),col='white')
lines(sin(x),cos(x))
}
winDialog('到点了，吃饭去啦，冲啊! \n\n (旁白：我晕，这个钟走反
了吧? -_-//)')
```



置信区间:

```
f = function(n = 1000, alpha = 0.95, rn = 50) {  
  d = replicate(n, rnorm(rn))  
  m = colMeans(d)  
  #s = apply(d, 2, sd)  
  z = qnorm(1 - (1 - alpha) / 2)  
  y0 = m - z * 1 / sqrt(50)  
  y1 = m + z * 1 / sqrt(50)  
}
```

```
plot(1, xlim = c(0.5, n + 0.5), ylim = c(min(y0), max(y1)),
     type = "n", xlab = "", ylab = "")
abline(h = 0, lty = 2)
for (i in 1:n) {
  arrows(i, y0[i], i, y1[i], length = 0.05, angle = 90,
         code = 3, col = ifelse(0 > y0[i] & 0 < y1[i], "blue",
                                "red"))
  points(i, m[i])
}
```

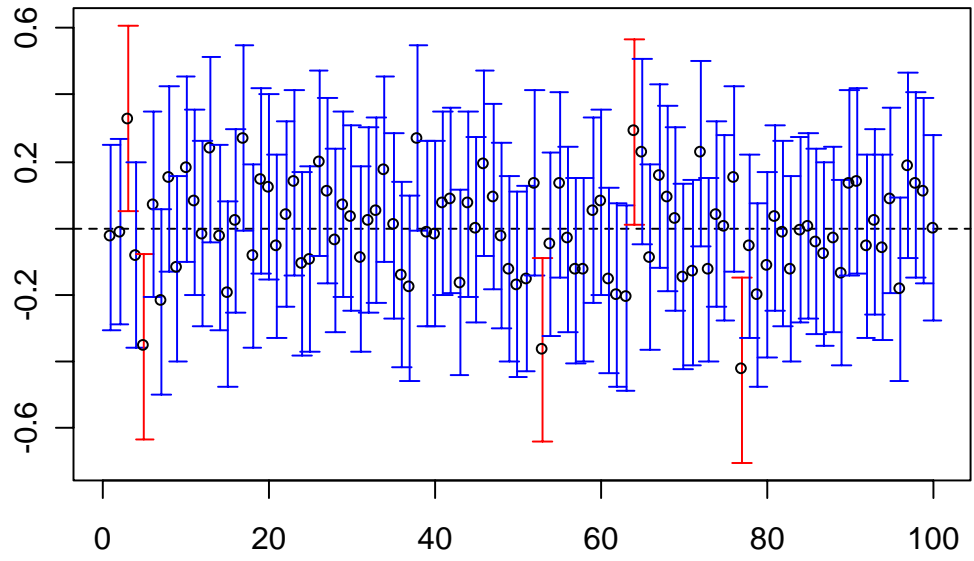
```
    Sys.sleep(.1)
```

```
}
```

```
  Sys.sleep(3)
```

```
}
```

```
replicate(2, f(100,.95))
```



重要的不在于图形的花哨，而是表达的信息，还有对于图形原理的深刻理解！（你确信你理解直方图吗？）

一句话：在 R 面前，我时常感觉到我渺小得如同蚂蚁的左脚小指的指甲盖儿。

四、写在最后（给师弟师妹们）

总听师长说要打好基础，我们究竟需要什么基础？

我是统计学方向的学生，在统计学院已经呆了近五年，到我这种“老龄”阶段，“打基础”只能幻想了（“我想早恋，但是已经晚了……”）

那么，想早恋的都趁早吧！

这是我的个人晚恋总结:

□ 高等代数和数学分析: 矩阵的乘法、矩阵特征根/特征向量(优化问题中常遇到)、迹的性质 $\text{tr}(AB)=\text{tr}(BA)$ 、行列式的乘积 $|AB|=|A|*|B|$ 、正定 (positive definite) 与正交 (orthogonal) 矩阵的定义、各种分布/密度函数的微积分、泰勒展开 (Talor expansion)、凸函数、偏导数 (太重要了)、极值和条件极值 (记好拉格朗日乘数法)

- 概率论：贝叶斯公式、分布的变换（一个随机变量/随机向量的函数的分布）、随机变量的定义、期望和方差的定义（你一辈子都会和它们打交道）、Chebyshev 不等式、柯西-施瓦兹不等式、熵和母函数、大数定律和中心极限定理（注意条件！）
- 数理统计：统计量及其分布（统计量从正态到 t、卡方、F 的一连串构造、经验分布函数）、赵选民那本数理统计书上 P58 公式 2.13: $(\bar{X} - \mu)/(S_n / \sqrt{n}) \xrightarrow{L} N(0,1)$ 、

点估计（矩估计与极大似然估计：发生的都是大概率事件）与区间估计、假设检验（小概率事件不太可能发生）P165 之后的非参数假设检验很重要！

□ 回归： $\hat{\beta} = (X'X)^{-1}X'y$ 、帽子矩阵 H 等幂且迹为 $p+1$ 、

$$\hat{\sigma}^2 = \frac{1}{n-p-1} \sum_{i=1}^n e_i^2, \quad D(\hat{\beta}) = \sigma^2(X'X)^{-1}, \quad \text{系数检验的 } t$$

统计量构造、F 统计量构造、回归诊断的数学本质、选

择子集的传统原则： R_a^2 ；AIC； C_p 、岭回归、非

线性回归（迭代求解的方法）、广义线性模型

- ❑ 多元统计分析：玩弄矩阵
- ❑ 非参数统计：玩弄秩，到时候编程序就死命地用 `rank()` 函数（有秩走遍天下）
- ❑ 时间序列：平稳的概念、自相关、偏自相关、关于 ARIMA：初学的时候盯着一件事——AR(1)模型的系数怎样用最小二乘法求解出来，不然学完将一无所获。
- ❑ 抽样：实际上是统计的基础，但是这个地位很少被强

调。关于这门课程，思考一个问题就达到境界了：为什么样本均值近似服从正态分布？悟透了这一点，便打通了从概率论到数理统计到抽样的经脉。（我相信有很多人在这里有着根深蒂固的误解，参见某些教科书）

现在我期待着弄清楚什么：

因子分析的计算过程以及数学表达式的现实意义、IV（工具变量）与 GMM（广义矩估计）、PLS（偏最小二乘）的现实意义、分位数回归的系数估计方差、对数线性模型、Zero-Inflated 系列模型（ZIP、ZIB 等）、机器学习算法模型的推广性能、SEM（结构方程模型）的收敛性能与估计的稳定性、线性混合效应模型（Linear Mixed Effects Models）的随机效应的现实意义、罚似然、伪似然……

统计之都的正式建立时间是 2006 年 5 月 19 日,也就是说,过两天就是COS一周年“诞辰”了。一年时间中,我们迅速积累了近 17000 名会员,发表近 3500 篇主题、26000 篇帖子(比率说明什么?),每月网站独立访问IP近 30000,每天论坛访问IP数超过 1500。

Last but not least,
(排名不分先后)

感谢勤劳的师弟王剑（无痕），通过和他的谈话，我才得以了解到师弟师妹们的学习情况，同时他作为 COS 论坛的另一位管理员付出了辛勤的劳动。

感谢热情的 COS 管家杨俏（shashouzhong），她为本次活动以及 COS 论坛的发展都做了大量的工作。

感谢李静萍老师、王燕老师、谢邦昌老师、薛薇老师、吴喜之老师，以及 amzon007、netcow、zouwu 等各位老师一直以来学术上的鼎力支持，还有统计学院各位领导老师

（如王晓军老师、宋大我老师等）长期的关怀！

感谢 xifan、colinisstudent、ypchen、abel、jyma、涤尘、cran、nurseshark、shoeda、areg、蟋蟀、woodcutter、lucky、gracefeng 等各位版主在各自岗位上的认真负责。

感谢工作在 COS 网站幕后的几位功臣：鲁明、叶舟、刘秋艳、郝明月、陈喆、雷斯帏！

当然，更要感谢广大会员的支持！我不敢列举出一个名单，

因为对本论坛做出贡献的会员实在太多。若仅从我个人的印象来说，我们要感谢 rtist、micro@、liutyy、hcg930、eshanzi、ilikemath、anning189、drewlee、bjt、sociology、anita_jiu、statax、.....

感谢"select username from COS 会员数据库".....

青春年少的人们，让我们再接再厉，用我们的努力打造一个统计学的精品平台，继而推广、普及统计学知识，谢谢！

IN THE NAME OF STATISTICS, UNITE!

WWW.COS.NAME