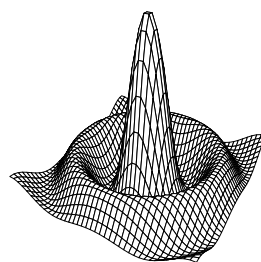


R 常见问题解答

R frequently asked questions
<http://www.r-project.org>



153 分钟学会 **R**



This document is generated from L^AT_EXsources compiled with x_eL^AT_EX in a Windows platform. The used packages are amsmath, listings, makeidx and so on.

序:

这篇文档内容的来源多样,既有来自于 R 官方文档(包括 `R_intro`, `R_data`, `R_admin`),也有来自于互联网的 contributed documents; 还有若干来自于 Capital Of Statistics 论坛的讨论问题。

本文档的目的是为具有一定统计(数学)背景的 R 软件初学者提供一个快速认识 R 软件的平台,如果你无此背景,可能会对其中的若干表达存在疑问。这篇文档重点不在统计方法上,因此所列问题不可能详尽到统计学的每个知识点。

R 是一个很庞大的体系,在 CRAN 的 Task Views 上可以清楚地看到贝叶斯推断、聚类分析、机器学习、空间统计、稳健统计等方法的介绍。而这些方法又通过相应的 R Packages 扩展,可以说学习 R 是一件没有尽头的事情。

如果你的英文阅读没问题,那么精读一本关于 R 的原版书籍也是一个不错的选择,但这个开头常常让人很头痛。希望这份 44 页的文档,对你认识、学习 R 是个不错的帮助。

刘思喆^{1 2}

February 4, 2013

致谢:

abel,cran,rtist,Xie Yihui,zhangv ...

¹<http://www.bjt.name/todo>

²sunbjt<at>gmail.com

A 前言

1. R 是做什么的？

R 是一个有着统计分析功能及强大作图功能的软件系统，是由新西兰奥克兰大学统计学系的 Ross Ihaka 和 Robert Gentleman 共同创立。由于 R 受 Becker, Chambers & Wilks 创立的 S 和 Sussman 的 Scheme 两种语言的影响，所以 R 看起来和 S 语言非常相似。

2. 在哪里可以下载 R 的安装程序？

在 R 的官方网址上，选择网站镜像 <http://cran.r-project.org/mirrors.html>，比如 UC Berkeley 下载软件副本。R 拥有在 Linux, MacOS X, Windows 平台下的各个版本，如果是 Windows 用户，进入镜像网站，选择 Windows (95 and later)，进入 base，下载 R-x.x.x-win32.exe。

3. 为什么 R 叫做 R？

部分是因为两位 R 的作者 (Robert Gentleman 和 Ross Ihaka) 的姓名，部分是受到了贝尔实验室 S 语言的影响（称其为 S 语言的方言或 GNU S）。

4. CRAN 是什么意思？

CRAN 是 *Comprehensive R Archive Network* 的简写，顾名思义：是拥有同一资料，包括 R 的发布版本，包，文档和源代码的网络集合。

5. 我是新手，我如何开始学习 R？

由于 R 的创立者和维护人员大部分都是统计学家，所以如果你有相关统计背景，将是一个极好的帮助。如果没有相关背景，R 的官方网站拥有大量的英文学习资源可供选择。当然还可以参考 <http://cran.r-project.org/other-docs.html> 上的中文翻译文档。

统计之都 BBS 是一个非常不错的学习 R 的中文网站。在论坛上，你可以搜索到大量学习资料或直接提出问题同大家探讨。

6. 初学者阅读 R 自行安装的 R-intro 就可以了么？

R-intro 确实是官方文档中最基础的，但它不是从 R 软件应用角度讲的，故并不适合于 R 初学者。可以参考 R-intro 中数据类型、结构部分，作为基础学习。

7. 能列举一些 R 的经典书目么？

正如这份文档序言中所述，《153 分钟学会 R》更侧重于快速认识 R 或者为拥有一定统计背景的初学者提供快速掌握 R 语言提供一个有效途径。更多技术细节或统计知识仍需通过阅读相应文献来掌握。

中文资料的话，推荐华东师范大学汤银才老师的《R 语言与统计分析》和北京工业大学薛毅老师《统计建模与 R 软件》。

当然还有一些原版资料是必须要读的，例如：

Modern Applied Statistics with S (Venables and Ripley)
The New S Language: A Programming Environment for Data Analysis and Graphics
— (Richard A. Becker, John M. Chambers, Allan R. Wilks)
A Handbook of Statistical Analysis Using R (Brian S. Everitt, Torsten Hothorn)
Data Analysis and Graphics using R (Maindonald and Braun)
Introductory Statistics with R (Dalgaard)
R in a nutshell

8. R 需要编程么？

不！大多数时候不需要，因为 R 有很多函数和包，而且每天都在增加，你用的一般方法和函数都可以在 R 自带包中找到。

9. 能否简单举一个 R 的例子？

生成 100 个高斯（正态）分布随机数，并对这 100 个数进行特征描述。

```
1 | x <- rnorm(100, mean = 5, sd = 0.1)
   | mean(x)
3 | sd(x)
   | summary(x)
```

当然你还可以使用 `demo()` 函数，比如 R 漂亮的图形演示：

```
demo(graphics)
```

10. R 需要注册费用么？

不需要！R 是一款在 GNU General Public License (GPL) 下发布的开源软件，只是很少一部分包不能用于商业用途。不知道为什么有些费时、费力且价格不菲的商业统计软件，居然还在生存？

11. 为什么 R 不能使用超过 50% 的 CPU？

这是 Windows 下任务管理器的误导，它将多个 CPU 看作是单个 CPU，同时计算使用比例。而 R 是单线程计算软件，它不能同时使用 2 个以上的 CPU。当你的计算机应用的是双核技术，你会发现 CPU 应用会定格在 50% 上。

12. 如何在发行出版物里引用 R

如果你是 \LaTeX 用户，可以在 R 中使用命令 `citation()` 得到可供 \BibTeX 使用的内容；或者是某一个包的引用

```
1 | citation(package = 'package')
```

B 基础知识

13. 如何获得帮助?

R 的帮助系统非常强大，可以直接使用 “?topic” 或 `help(topic)` 来获取 topic 的帮助信息；也可使用 `help.search("topic")` 来搜索帮助系统。

如果你只知道函数的部分名称，那么可以使用 `apropos("tab")` 来搜索得到载入内存所有包含 tab 字段的函数。

如果还没有得到需要的资料，还有 R Site Search: <http://finzi.psych.upenn.edu/search.html>，等价于在 R 平台上使用 `RSiteSearch()` 函数。

14. R 可使用的最大内存是多少?

R 经常因为过分消耗内存而受到指责，这主要是因为 R 的设计理念造成。不过还好，我们使用的数据量通常不是很大，通常都可以处理，而且 R 也逐步地在做优化。如果在特定条件下需要更大的内存来做运算，提供两种途径来设定（增大）内存：

- 启动 R 进程前，增加 R 启动参数。在 CMD 环境下，运行增加参数的 Rterm:

```
1 | r --max-mem-size=1Gb
```

或通过添加 `RHOME/bin` 至系统环境中，直接在“运行”中运行³

```
1 | rgui --max-mem-size=1Gb
```

- 启动 R 进程后，通过 `memory.limit` 函数增大 R 进程的内存限制。

32 位 R 的工作内存大小的设定值为 32Mb 到 3Gb 间的任意数值，而 64 位则没有任何限制。但需要提示的是：32 位 Windows 平台可用最大有效内存为 2Gb，也就是说，实际上在 32 位 Windows 上 R 的可用工作内存区间为 32Mb 至 2Gb。

15. 为什么 `help.search()` 搜索不能使用?

基于浏览器的搜索引擎要求正确安装完整版 Java，且 Java 和 Javascript 须嵌入浏览器。

16. R 支持中文么?

支持，但不好！在 R 中，大部分包的作者都是以英文为母语的，不会对中文字符考虑太多，故建议使用全英文环境。

但在绘图时，可能会遇到中文支持问题，这时候需要设置 `par` 中的 `family` 参数，比如在 2.14.0 版本以后 R 可以直接使用 `cairo` 模式，这时就需要使用

```
1 | png('foo.png', type = "cairo", ,family="SimSun")
   | plot(1,type = 'n')
3 | text(1,1, '这是 cairo 模式')
```

³同样支持 Rterm

```
dev.off()
```

同时支持黑体 (SimHei)、楷体 (KaiTi_GB2312)、幼圆 (YouYuan)、隶书 (LiSu)，或者更一般的图形设备 (非 cairo)，设置 `family = 'GB1'`，即可。

17. R 支持自动补全 (Tab completion) 么?

支持! 在 2.5.0 版本中, R 引入了命令自动补全功能, 使用 Tab 键能自动补全 R 命令; 或使用第二次 Tab 后, 返回所有可能的补全命令列表。

18. 如何清除变量?

清除单个变量使用 `rm()` 函数, 清除内存中所有的变量:

```
rm(list = ls(all = TRUE))
```

19. 如何更改小数点后显示数字位数?

`options(digits =)`, `digits` 后面的参数为 1 至 22 的数字, 默认为 7。 `options` 函数还可以改变很多全局选项, 如更改提示符 (`prompt`), 是否显示错误信息 (`show.error.messages`) 等。

20. 如何调用系统内的程序?

使用 `system()` 函数或用 `shell.exec()` 调用相应程序来打开文件:

```
1 # go to the cran
2 system(paste('C:/Program Files/Internet Explorer/iexplore.exe',
3 'cran.r-project.org'), wait = FALSE)
4 # invoke the notepad
5 system("notepad")
6 shell.exec("C:/WINDOWS/clock")
```

21. Windows 下升级 R , 但不想重装 packages ?

鉴于 R 的更新速度 (大概每 3 个月会有一次升级), 频繁的更新主程序, 可能是我们在 Windows 环境下最常遇到的问题, 再加之本地 Contributed Packages 数量很大, 这个问题会更加突出。

一般来说, R 本地的目录结构相对固定, 可以通过在其他目录下安装 R , 再将旧版本保留的 `library` 目录下的文件拷贝至新版本 `library` 目录下或卸载 R , 把 R 装到旧的目录下, 最后使用如下命令更新:

```
1 update.packages(checkBuilt = TRUE, ask = FALSE)
2 # or install.packages(.packages(all = TRUE))
```

22. 如何卸载已安装的 packages?

参考

```
2 | remove.packages(c("pkg1", "pkg2"),  
lib = file.path("path", "to", "library"))
```

23. R 的工作目录在哪里?

一般的, Windows XP 下的 R 工作目录在

```
C:\Documents and Settings\username
```

或者使用 `getwd()` 命令获得 R 的工作目录 (Working Directory), 使用 `setwd()` 设置工作目录位置。

24. 我怎样保存自己的工作?

使用 `save.image()` 函数。它将在 R 的起始目录保存记忆区 (working space) 至.RData 文件; 或者使用 `save(..., file =)` 保存需要保存的 R 对象。

25. R 如何安装包?

通过选择下载镜像, R 可以自动安装未安装在本地的包, 当然也可以从镜像网站下载可用的包, 直接本地安装⁴。

26. library() 的逆向操作是什么?

当加载包后, 需要分离 R 同包时, 可以使用

```
1 | detach("package:pkg")
```

27. Library 和 Package 有什么区别?

这两个概念的确容易混淆, 因为 R 中加载 Package 的命令是 Library! Library 是一个目录, 可能包含一个或多个 Package; 而 Package 是包含函数、数据、手册的一个集合, 属于某个 Library, 即 (Windows 下) 的 "*.zip" 文件。

28. 如何得到加载 Package 的列表?

`search()` 函数返回当前加载的包的情况, 使用

```
1 | .packages(all.available = TRUE)
```

命令获得本地安装的包列表。

当 R 启动后, R 在内存中会自动加载若干 Package:

⁴R 有 Unix、Mac、Windows 三个版本, 注意包也分别对应三个版本

R 初始状态载入包列表

包	描述
stats	常用统计函数
graphics	基础绘图函数
grDevices	基础或 grid 图形设备
utils	R 工具函数
datasets	基础数据集
methods	用于 R 对象和编程工具的方法和类的定义
base	基础函数

29. 如何使用 R 内置的数据集?

R 在 datasets 包中共提供了 102 个可以使用的数据集，这些数据集都可以通过 data() 函数加载入内存。

```
1 dim(data()$results)
  data()$results[,4]
```

30. R 的数据类型有几种?

R (S 语言) 没有标量，它通过使用各种类型的向量来存储数据。常用的数据类型 (class) 有：

常用数据类型

	类型	说明
1	字符 (character)	它们常常被引号包围
2	数字 (numeric)	实数向量
3	整数 (integer)	整数向量
4	逻辑 (logical)	逻辑向量 (TRUE=T、FALSE=F)
5	复数 (complex)	复数 ^a
6	列表 (list)	S 对象的向量
7	因子 (factor)	常用于标记样本

^a参考第 18 页“复数计算”

在 R (S) 语言中，有一点要牢记：

Everything in S is an object;
Every object in S has a class.

31. data frame 是什么?

data frame (数据框) 可以理解是一个松散的数据集。它可以是由不同类型的列 (数字、因子、字符等) 组成的类矩阵 (matrix-like)。

32. 如何得到函数的代码?

通常情况你只需要在 R 平台下写出你需要查看的函数名，回车即可。比如：

```
dist
```

但有时候这个函数可能是一个泛型函数 (Generic Function)，上面的方法就需要稍稍改进一下：先使用 `methods()` 函数来查看这个类函数的列表，找到具体需要的函数⁵，写出来，回车 — 问题解决。

```
1 summary # It is a generic funciton
2 methods(summary) # list of the S3 methods
3 summary.lm # maybe you want to know the linear models's summary
```

如果要究根问底，可以去下载源代码压缩包 (*.tar.gz，比如 R-2.5.1.tar.gz)

33. 我想查看一个矩阵的前（后）几行，怎么办？

可以使用 `head()` 或 `tail()` 函数。

```
1 head(CO2)
```

这两个函数是类函数，它们可以应用于向量、矩阵、数据框、表格或函数。如果只想随机看看对象中的一些内容，还可以使用 `car` 包中的 `some` 函数。

34. 在 R 中公式的符号都是什么意义？

拿常见的 `lm`, `glm` 模型来说，`y ~ model` 是一种特定的格式，表示以 `y` 为响应变量，模型为 `model`。其中 `model` 中的变量由 `+` 来连接，或者由 `:` 来表示变量间的“交互作用”。

除了 `+` 和 `:`，我们使用 `*` 来表示 '`a + b + a : b`'。`(a + b + c)^2` 表示 `(a + b + c) * (a + b + c)`，即主因素 `a`、`b`、`c` 和各个因素的交互作用。`-` 表示去掉之意。`(a + b + c)^2 - a : b` 表示 '`a + b + c + b : c + a : c`'。

在公式表达中除了变量和因子名外，运算符号也是可以存在的。如 '`log(y) a + log(x)`' 是合法的。

符号. 在 `update` 函数中有特殊的意义，它表示“已经存在”之意。

```
1 fm <- aov(Speed ~ Run + Expt)
  fm0 <- update(fm, . ~ . - Run)
```

在第 H 节中的网格 (`lattice`) 绘图，我们还会看到 `|` 符号，它可以用来标示“条件变量”。

35. R 里面可以使用科学计数法么？

可以。

⁵标注星号的函数可以使用 `getAnywhere()` 函数获得代码

```
1 1e10 == 10000000000
2 1.2e-4 == 0.00012
```

C 输入输出

36. R 可以读取其他统计软件录入的数据么？

可以，使用 `foreign` 包，它可以读取 Minitab, S, SAS, SPSS, Stata, Systat, dBase 保存的数据。或者使用 `Hmisc` 包中 `sas.get`, `spss.get` 函数来获取 SAS 和 SPSS 软件格式数据。

但随着这些商业软件版本更新，依旧会出现一些文件不能通过 `foreign` 包读取的情况，建议在交换数据时尽量使用 `.csv` 这种通用格式。

37. R 可以读 Excel 的数据么？

可以，但不推荐直接读取 Excel 文件，或许只有微软知道 Excel 里面有什么东西。通常有三种方法读取 Excel：

1. 将 Excel 另存为 `csv`(Comma Separated Values) 文件，使用 `read.csv()` 函数读取（推荐）；
2. 加载 `RODBC` 包，使用 `odbcConnectExcel()` 函数读取 `xls` 文件，

```
1 library(RODBC)
2 z <- odbcConnectExcel("rexceltest.xls")
3 dd <- sqlFetch(z, "Sheet1")
4 close(z)
```

详细请参考 R Data Import/Export;

3. `xlsReadWrite` 包中的 `read.xls` 函数。

38. 可以将 R 中显示的结果输出到文件么？

可以。使用 `sink()` 函数。

```
1 data(CO2)
2 sink("CO2.txt")
3 CO2
4 sink() # go to your work directory, you will get CO2.txt
```

39. 如何调用 R 的输出信息？

R 提供了 `capture.output()` 函数，这个函数可以将 R 的输出信息转化为字符或文件。

```
1 glmout <- capture.output(example(glm))
2 glmout[1:5]
```

当然，如果你想得到漂亮的输出，Go to L^AT_EX!

40. R 可以从内存直接读写数据么？

可以。拷贝需要读取的内容，使用

```
1 data <- read.table("clipboard")
2 write.table("clipboard")
```

这个技巧对于临时性的分析工作非常有益。

41. 怎样将因子 (factor) 转换为数字

这个问题时有发生，假设 `f` 是一个这样的因子对象，我们可以使用

```
1 as.numeric(as.character(f))
2 # or
as.numeric(levels(f))[as.integer(f)]
```

这个点一定要小心，因为对于顺序型 factor 数据，如果强制转化为数值型，会返回的是 factor 的顺序信息，而非你看到的 character 信息。

42. R 可以使用电子表格输入数据么？

可以使用 `edit()`和 `fix()`函数。

```
1 data <- data.frame()
edit(x) ; fix(x)
```

但，这非常 naive!

43. 为什么当我使用 `source()` 时，不能显示输出结果？

对需要显示输出的对象使用 `print()`，或者使用 `source(file, echo = TRUE)`。如果 R 代码里面包含 `sink()` 之类的函数，必须使用 `source(file, echo = TRUE)` 才能得到正确的输出结果，否则 `sink` 的对象将为空。

44. R 可以输出可供 T_EX 使用的文本么？

可以，参考 `Hmisc` 包中的 `latex()` 函数和 `xtable` 包中的 `xtable()`函数。

```
1 a <- matrix(1:6, nr=1) # require(xtable)
2 colnames(a) <- paste("col", 1:6)
xtable(a)
```

`xtable()` 函数可以用于产生 HTML 格式的原码，这样 R 生成的表格就可以非常方便、漂亮地插入到 word、powerpoint 这类文字处理软件。

输出 L^AT_EX 格式的表格还可以 `quantreg` 包中的 `latex.table()`函数。

45. 找不到文件，但我知道它在哪！

在 R 里面使用必须使用双反斜杠或单斜杠表示文件路径，比如：

```
1 | d:\\R-2.4.1\\library\\xgobi\\scripts\\xgobi.bat
   | d:/R-2.4.1/library/xgobi/scripts/xgobi.bat
```

当然还可以使用 `file.choose()` 函数打开一个 Windows 标准文件选择对话框，手动选择文件。当然还有可以使用 `choose.dir()` 打开 Windows 标准目录选择对话框 ☺。

46. R 可以直接从数据库读取数据么？

可以，并且还可以通过 SQL 语句对数据库进行操作。R 对于基于 SQL 语言的关系型数据库有良好的支持，这些数据库既有商业数据库 Oracle、Microsoft SQL Server、IBM DB2 等，也包含在 GNU General Public License (GPL) 下发布的 MySQL 等开源数据库。

RMySQL⁶ 包中提供了到 MySQL 数据库的接口；RODBC 包提供了更为广泛数据库接口的解决方案 — 支持所有标准 ODBC 接口的数据库。通过这种方式，相同的 R 代码可以方便地应用于不同类型的数据库。

```
2 | library(RODBC)
   | ch <- odbcConnect("stocksDSN",uid = "myuser",pwd = "mypassword")
   | stocks <- sqlQuery(ch,"select * from quotes")
4 | odbcClose(ch)
```

经测试，Windows 平台上的 Microsoft SQL Server、Access、Oracle、MySQL、PostgreSQL，和 Linux 平台上的 MySQL、Oracle、PostgreSQL、SQLite 都有良好的应用案例（详细参考 [R-data](#)）。

D 数据处理

47. 如何删掉缺失值？

在 R 中使用 NA (not available) 表示缺失值，要注意 R (S) 语言中 NA 同样是一个逻辑值，⁷

```
2 | x <- NA
   | x > 3
   | class(x)
```

故当判断是否相等时不能使用

```
1 | x == NA
```

⁶需要包 DBI 的支持

⁷R 共有三个逻辑值 TRUE、FALSE、NA

来判断缺失值。而是使用函数 `is.na()` 来判断是否为缺失值，使用

```
1 x[!is.na(x)]
```

删除缺失值。

48. 如何将字符串转变为命令执行？

这里用到 `eval()` 和 `parse()` 函数。首先使用 `parse()` 函数将字符串转化为表达式 (expression)，而后使用 `eval()` 函数对表达式求解。

```
1 x <- 1:10
  a <- "print(x)"
3 class(a)
  eval(parse(text = a))
```

但在字符串转化为 expression 过程中可能会有一些小麻烦，比如下面这种情况：

```
1 a <- 3
2 x <- 4
  str2expr <- function(x){eval(parse(text = x))}
4 str2expr('a + a*sqrt(x)')
```

但这时候会报错，原因是参数名为 `x`，且外部的变量名也为 `x`，而实际上 `eval` 在对 expression 做 `eval` 的时候优先找最近的上一层的 `x`，也就是“参数 `x`”，而不是“数字 `x`”，所以 `sqrt` 要报错。那么上面的代码需要变通一下：

```
1 a <- 3
2 x <- 4
  str2expr <- function(y){eval(parse(text = y))}
4 str2expr('a + a*sqrt(x)')
```

或者设置函数表达式中的所有变量都是全局变量，即在 `eval` 能够在 `.GlobalEnv` 中找到：

```
str2expr <- function(x){eval(parse(text = x), .GlobalEnv)}
```

49. 如何向一个向量追加元素？

参考 `append()` 函数。

```
1 x <- 1:5
  (foo <- c(x[1],0,x[2:5])) # expected result
3 append(x , 0 , after = 1)
```

50. 如何移除某行 (列) 数据

可以使用函数 `subset(select =)`；或者使用下标：

```

1 x <- data.frame(matrix(1:30, nrow = 5, byrow = T))
  dim(x)
3 print(x)
  new.x1 <- x[-c(1,4),] #row
5 new.x2 <- x[, -c(2,3)] #col
  new.x1 ; new.x2

```

事实上，关于选取特定条件下的数据框数据，subset 函数同使用下标效果相同：

```

1 iS <- iris$Species == "setosa"
2 iris[iS, c(1,3)]
  subset(iris, select = c(Sepal.Length, Petal.Length),
4         Species == "setosa")

```

51. 如何比较两个数据框是否相同？

比较每个元素是否相同，如果每个元素都相同，那么这两个数据框也相同

```

1 a1 <- data.frame(num = 1:8, lib = letters[1:8])
2 a2 <- a1
  a2[[3,1]] <- 2 -> a2[[8,2]]
4 any(a1!=a2) # all(a1 == a2)

```

any() 函数可以返回是值是否至少有一个为真的逻辑值。而数据框中的元素有不等的情况，则

```
a1!=a2
```

将返回至少一个 TRUE，那么 any() 函数将判断为 TRUE。同样也可以使用 identical() 函数。

```
1 identical(a1, a2)
```

如果需要返回两个数据框不相同的位置，可以使用

```
1 which(a1!=a2, arr.ind = TRUE)
```

arr.ind 参量是 **array indices** 之意，返回数据框的行列位置。

52. 我的数据框有相同的行，如何去掉这些行？

参考 unique 函数。unique 函数可以去掉向量、数据框或类似数列的数据中重复的元素。

```

1 x <- c(9:20, 1:5, 3:7, 0:8)
  (xu <- x[!duplicated(x)])
3 unique(x) # is more efficient

```

这里 duplicated 函数返回了元素是否重复的逻辑值。

53. 如何对数列 (array) 进行维度变换?

使用函数 `aperm`

```
1 x <- array(1:24, 2:4)
  xt <- aperm(x, c(2,1,3))
3 dim(x) ; dim(xt)
```

54. 如何删除 list 中的元素?

R 中使用 NULL 表示无效的对象。

```
1 lst <- list("a"=list("b"=1,"c"=2),"b"=list("d"=3,"e"=4))
  lst[["a"]][["b"]] <- NULL # or lst$a$b <- NULL
3 lst
```

55. 如何对矩阵按行 (列) 作计算?

使用函数 `apply()`

```
1 vec=1:20
  mat=matrix(vec, ncol=4)
3 vec
  cumsum(vec)
5 mat
  apply(mat, 2, cumsum)
7 apply(mat, 1, cumsum)
```

56. 如何注掉大段的 R 脚本

如果你使用支持正则表达式的文本编辑器的话, 可以考虑用正则表达式 (Regular Expression); 或者将大段的代码写入一个 *.R 文件, 如果需要注掉的话, 在 `source(*.R)` 前加入 `#` 即可; 还可以使用

```
1 if (FALSE){
  something passby
3 }
```

57. 如何对数据框 (data frame) 的某列作数学变换?

使用 `transform()` 函数对其操作, 具体参考?transform

58. 如何求解两组平行向量的极值?

`pmax()` 和 `pmin()`, 如:

```

1 x <- 1:10 ; y <- rev(x)
  pmax(x,y) ; pmin(x,y)

```

59. 如何对不规则数组进行统计分析?

参考 `tapply()` :

```

2 n <- 17; fac <- factor(rep(1:3, len = n), levels = 1:5)
  table(fac)
  tapply(1:n, fac, sum)
4 tapply(1:n, fac, mean)
  ## or reverse a list
6 to <- list(a = 1, b = 1, c = 2, d = 1)
  tapply(to, unlist(to), names)

```

`tapply()` 的常见于方差分析中对各个组别进行 `mean`、`var` (`sd`) 的计算。说到概要统计，不得不说另外一个函数 `aggregate()`，它将 `tapply()` 函数对象为向量的限制扩展到了数据框。⁸

```

1 attach(warpbreaks)
  tapply(breaks, list(wool, tension), mean)
3 aggregate(breaks, list(wool, tension), mean)
  ## from the help
5 aggregate(state.x77,
             list(Region = state.region,
7             Cold = state.x77[, "Frost"] > 130),
             mean)

```

60. 判断数据框的列是否为数字?

使用 `sapply` 函数

```
sapply(dataframe, is.numeric)
```

`sapply` 有循环之意，即返回了 `dataframe` 的每一列的信息，同样还可以显示内存中所有对象的占用大小：

```
sapply(ls(), function(x) round(object.size(get(x))/1024/1024))
```

61. 一组数中随机抽取数据?

函数 `sample()`

⁸当然同样概要统计的表现形式不一样

	sample(n)	随机组合 $1, \dots, n$
	sample(x)	随机组合向量 $x, length(x) > 1$
	sample(x, replace = T)	解靴带法
	sample(x,n)	非放回的从 x 中抽取 n 项
	sample(x,n, replace = T)	放回的从 x 中抽取 n 项
	sample(x,n, replace = T ,prob = p)	以概率 p , 放回的从 x 中抽取 n 项

```

1  n <- 1000
   x <- sample(c(-1,1), n, replace=T)
3  plot(cumsum(x), type="l",
      main="Cumulated sums of Bernoulli variables")

```

还可以参考第 20 页中关于模拟已知分布的随机数据函数，如：

```
rnorm(100, mean=0, sd=1)
```

62. 如何根据共有的列将两个数据框合并？

我们经常会遇到两个数据框拥有相同的时间或观测值，但这些列却不尽相同。处理的办法就是使用 `merge(x, y ,by.x = ,by.y = ,all =)` 函数。

如果你对 SQL 语言非常熟悉，还可以使用 `sqldf` 包来进行数据框合并，这个包可以让用户在 R 环境下使用标准 SQL 语言进行数据的预处理：

```

1  example(merge)
   sqldf("select * from authors a1, books a2
3      where a1.surname = a2.name", method = 'character')
   (m1 <- merge(authors, books, by.x = "surname", by.y = "name"))

```

当然 `sqldf` 的作用远远不止于此，一般性 SQL 数据操作均可以实现。

63. 如何将数据标准化？

参考 `scale` 函数。

```

1  x <- c(rnorm(100), 2*rnorm(30))
2  m <- scale(x, scale = F) # only centering
   n <- scale(x, center = F) # only scaling

```

64. 为什么 `fivenum` 和 `summary` 两个函数返回的结果不同？

因为他们对数据描述机理一致，所以有些教材将二者等同，但他们确实有细微差别。

```

1  > fivenum(c(1,4,6,17,50,51,70,100))
   [1] 1.0 5.0 33.5 60.5 100.0

```

```

3 > quantile(c(1,4,6,17,50,51,70,100))
5 0%      25%      50%      75%      100%
   1.00    5.50   33.50   55.75  100.00

```

我们看下他们的的定义：分位数是指有百分之多少的数据小于的数值⁹，我们可以看到关于 $\frac{1}{4}, \frac{3}{4}$ 分位数位置的定义：

$$1 + \frac{1}{4}(\text{length}(x) - 1), \frac{1}{4} \text{分位数位置}$$

$$1 + \frac{3}{4}(\text{length}(x) - 1), \frac{3}{4} \text{分位数位置}$$

那么数据

```

1 c(1,4,6,17,50,51,70,100)

```

的两个四分位数的位置分别为

$$1 + \frac{7}{4} = 2.75, 1 + \frac{21}{4} = 6.25$$

故对应分位数为

$$4 + (6 - 4) \times 0.75 = 5.5, 51 + (70 - 51) \times 0.25 = 55.75$$

而 `fiveum()` 函数中 N_L (下) 和 N_U (上) 两个数，是两次利用中位数概念：先取中位数将数据分为上下两部分当然，如果 $\text{length}(x)$ 为偶数，那么数据刚好被分为两部分，如果 $\text{lenght}(x)$ 为奇数，那么中位数同属上下两部分，然后再取各部分的中位数，即为 N_L, N_U 。

E 数学运算

65. 如何做出曲线积分？

R 语言使用 `integrate` 函数来得到积分结果，如

```

1 integrate(dnorm, -1.96, 1.96)
2 integrate(dnorm, -Inf, Inf)
3 ## a slowly-convergent integral
4 integrand <- function(x) {1/((x+1)*sqrt(x))}
5 integrate(integrand, lower = 0, upper = Inf)

```

66. 如何得到一个列向量？

矩阵转置可以使用函数 `t()`，R 中默认 x 为 “integer” 类型数据，这时可以用 `t(t(x))` 得到列向量：

⁹`summary()` 函数，即使用分位数概念

```

1 x <- 1:10 ; class(x)
  t(x)      ; class(t(x))
3 t(t(x))  ; class(t(t(x)))

```

行向量、列向量常常会有一个比较容易让人迷糊的地方：

```

1 x%*%x

```

计算的是 $x^T x$ （计算 $x x^T$ 使用 `%o%` 或 `outer()` 函数）。`crossprod()` 函数能避免这种情况：

```

1 XT.y <- crossprod(X,y)

```

它直接计算 $X^T Y$ ，可以看作前者的另一种表达方式，当然 `crossprod()` 更为有效¹⁰。由于 `outer()` 函数的矩阵意义，它常用于三维绘图数据，比如我们计算

$$10 \times \frac{\sin \sqrt{x^2 + y^2}}{\sqrt{x^2 + y^2}}$$

那么对应的 R 函数计算为：

```

1 f <- function(x,y) { r <- sqrt(x^2+y^2); 10 * sin(r)/r }
  z <- outer(x,y,f)

```

67. R 如何进行复数计算？

参考 `complex()` 函数的帮助。

```

1 x <- 1 + 1i # x <- complex(1,1)
2 Mod(x) ; Conj(x)

```

68. 如何生成对角矩阵？

对一个向量使用 `diag()` 函数，得到对角线元素为向量的对角矩阵；对整数 Z 使用此函数得到 Z 维的单位矩阵。

69. 求矩阵的特征值和特征向量的函数是什么？

参考 `eigen` 函数。已知 $A = \begin{bmatrix} -1 & 2 & 2 \\ 2 & -1 & -2 \\ 2 & -2 & -1 \end{bmatrix}$ 试求 $B = (\frac{1}{2}A^{-1}) + E$ 的特征值。

```

1 A <- matrix(c(-1,2,2,2,-1,-2,2,-2,-1),3,3)
2 m <- solve(0.5*A) + diag(c(1,1,1))
  eigen(m)

```

这里还使用了函数 `solve()`，这个函数用于运算

¹⁰当矩阵很大时，会非常明显☺

```
1 a%%*%x = b
```

而得到 x，当然也可以用来求矩阵的逆。

70. 如何构造上（下）三角矩阵？

参考函数 lower.tri() 和 upper.tri()。

```
1 Rmat <- matrix(1:16,4,4)
  Rmat[lower.tri(Rmat)] <- 0
3 Rmat
```

71. 求立方根如何运算？

$x^{1/3}$ 。在 R 里面 sqrt() 函数可以计算开平方，故新手容易推测开立方也有函数。事实上 R 里面使用 ^ 来作幂函数运算。^ 不但是运算符号，还可以看作是函数：

```
1 "^(x , 1/3)
```

在 R 中的运算符号包括：

R 中的运算符号

数学运算	+, -, *, /, ^, %%, %/%	加、减、乘、除、乘方、余数、整除
逻辑运算	>, <, >=, <=, ==, !=	大于，小于，大于等于，小于等于，等于，不等于

72. 如何求矩阵各行（列）的均值？

如果运算量不是很大，当然可以使用 apply() 函数。rowMeans() 和 colMeans() 函数可以更快地得到你要的结果。

```
1 m <- 1000 ; n <- 3000
  A <- matrix(1:m*n ,m ,n)
3 system.time(B1 <- matrix(apply(A, 2, mean), m, n ,by = TRUE))
  system.time(B2 <- matrix(colMeans(A), m, n , by = TRUE))
```

或者使用 sqldf 包来处理。

73. 如何计算组合数或得到所有组合？

choose() 用于计算组合数 $\binom{n}{k}$ ，函数 combn() 可以得到所有元素的组合。使用 factorial() 计算阶乘。希望大家还记得组合公式：

$$C_n^m = \frac{n!}{m!(n-m)!}$$

74. 如何在 R 里面求（偏）导数？

使用函数 D()

```

f1 <- expression(sin(x)*x)
f2 <- expression(x^2*y + y^2)
D(f,"x")

```

75. 如何模拟高斯（正态）分布数据？

使用 `rnorm(n, mean, sd)` 来产生 n 个来自于均值为 `mean`，标准差为 `sd` 的高斯（正态）分布的数据。在 R 里面通过分布前增加字母 ‘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’ 表示产生该分布的随机数。这些分布具体可以参考第 24 页中“如何做密度曲线”，或 R-intro 中的 [Probability distributions](#) 章节，或

```

help.search("distribution")

```

如已知 $X \sim N(3,1)$ ，计算

$$P(2 \leq X \leq 5)$$

利用正态分布的累积分布函数 `pnorm`

```

pnorm(5,3,1) - pnorm(2,3,1)

```

计算结果为 0.8185946，即图 1 中阴影的面积。

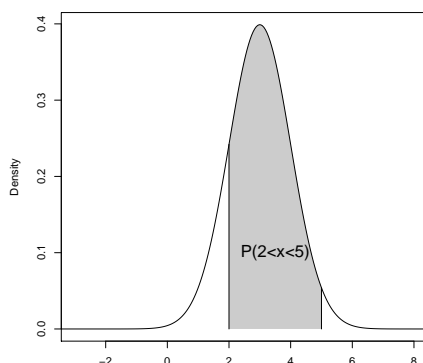


Figure 1: 正态分布累积分布函数的意义

76. 如何求一元方程的根？

使用 `uniroot()` 函数，不过 `uniroot` 是基于二分法来计算方程根，当初始区间不能满足要求时，会返回错误信息。

```

f <-function(x)x^3 - 2*x - 1
uniroot(f,c(0,2))

```

如果一元方程的根恰恰是其极值，那么还可以使用 `optimize()` 函数来求极值。

```
1 f <- function(x)x^2 + 2*x + 1
2 optimize(f, c(-2,2))
```

F 字符操作

77. R 对大小写敏感么？

R 中有很多基于 Unix 的包，故 R 对大小写是敏感的。可以使用 `tolower()`、`toupper()`、`casefold()` 这类的函数对字符进行转化。

```
1 x <- "MiXeD cAsE 123"
2 chartr("iXs", "why", x)
3 chartr("a-cX", "D-Fw", x)
4 tolower(x)
5 toupper(x)
```

78. R 运行结果输出到文件中时，文件名中可以用变量代替吗？

可以，通过使用 `paste()` 函数。

```
1 for(var in letters[1:6]){
2   x <- var
3   write.table(x, paste("FOO_", var, ".txt", sep = ""))
4 } # You will get "FOO_a.txt" ...
```

79. 在 R 中如何使用正则表达式 (Regular Expressions)

在 R 中，有三种类型的正则表达式：*extended* regular expressions，使用函数 `grep(extended = TRUE)`（默认）；*basic* regular expressions，使用 `grep(extended = FALSE)`；*Perl-like* regular expressions，使用 `grep(perl = TRUE)`。比如 “.” 用来匹配任意字符（使用 “\.” 来匹配 “.”）：

```
grep("J.", month.abb)
```

详细可以参考 `help("regex")`。

80. 如何在字符串中选取特定位置的字符？

参考 `substr()` 函数。

```
1 substr("abcdef", 2, 4)
2 substring("abcdef", 1:6, 1:6)
```

这个函数同时支持中文，用她来处理“简称”和“全称”还是一个不错的选择。

对于处理一般性网页文件（xml 或 html 格式），可以使用 XML 包来直接进行解析处理。使用 XML 包对网页进行数据和内容的提取非常方便，这里不做说明，请参考 XML 帮助文件。

81. 如何返回字符个数？

这里要区别一下 length，length 函数是返回向量里元素的个数。比如“你好吗”是长度为一的向量，但这个元素的字符长度为三，这里就需要使用 nchar 函数：

```
1 nchar('你好吗')
2 [1] 3
3 length('你好吗')
4 [1] 1
```

G 日期时间

82. 日期可以做算术运算么？

可以。一般我们需要使用 as.Date()，as.POSIXct() 函数将读取的日期（字符串）转化为“Date”类型数据，“Date”类型数据可以进行算术运算。

```
1 d1 <- c("06/29/07") ; d2 <- c("07/02/07")
2 D1 <- as.Date(d1, "%m/%d/%y")
3 D2 <- as.Date(d2, "%m/%d/%y")
4 D1 + 2 ; D1 - D2
5 difftime(D1, D2, units = "days")
```

这里需要注意一下 R 环境可能会对结果有影响，比如我的机器是中文环境，对于

```
1 as.Date('22JUN01', format='%d%b%y')
```

并不能识别，这时需要增加设置参数 Sys.setlocale("LC_TIME", "C") 来告知系统采用英文（北美）环境用法。

83. 如何将日期表示为“星期日, 22 七月 2007”这种格式？

使用 format() 函数。

```
1 format(Sys.Date(), format="%A, %d %B %Y")
```

具体 format 参数可以参考 help(strptime) 的 details 部分。

H 绘图相关

84. 如何在同一画面画出多张图？

这里提供三种解决方案:

- 修改绘图参数, 如 `par(mfrow = c(2,2))` 或 `par(mfcol = c(2,2))`;
- 更为强大功能的 `layout` 函数, 它可以设置图形绘制顺序和图形大小;
- `split.screen()` 函数。

推荐使用 `layout()` 函数, *Statistics with R* 的一个例子:

```
1 layout(matrix(c(1, 1, 1,
3                 2, 3, 4,
5                 2, 3, 4), nr = 3, byrow = T))
6 hist(rnorm(25), col = "VioletRed")
7 hist(rnorm(25), col = "VioletRed")
8 hist(rnorm(25), col = "VioletRed")
9 hist(rnorm(25), col = "VioletRed")
```

85. 如何设置图形边缘大小

修改绘图参数 `par(mar = c(bottom, left, top, right))`, `bottom`, `left`, `top`, `right` 四个参数分别是距离 `bottom`, `left`, `top`, `right` 的长度, 默认距离是 `c(5, 4, 4, 2) + 0.1`。或者修改绘图参数 `par(mai = c(bottom, left, top, right))`, 以英寸为单位来指定边缘大小。

86. 常用的 `pch` 符号都有哪些?

`pch` 是 `plotting character` 的缩写。`pch` 符号可以使用 “0 : 25” 来表示 26 个标识 (参看下图 “`pch` 符号”)。当然符号也可以使用 `#`, `%`, `*`, `|`, `+`, `-`, `.`, `o`, `O`。值得注意的是, 21 : 25 这几个符号可以在 `points` 函数使用不同的颜色填充 (`bg = 参数`)。

```
1  3  5  7  9 11 13 15 17 19 21 23 25
   ○ + ◇ ⊠ ⊡ ⊢ ⊣ ⊤ ⊥ ⊦ ⊧ ⊨ ⊩ ⊪ ⊫ ⊬ ⊭ ⊮ ⊯ ⊰ ⊱ ⊲ ⊳ ⊴ ⊵ ⊶ ⊷ ⊸ ⊹ ⊺ ⊻ ⊼ ⊽ ⊾ ⊿ ⊿
0  2  4  6  8 10 12 14 16 18 20 22 24
   □ △ × ▽ * ⊕ ⊗ ⊘ ⊙ ⊚ ⊛ ⊜ ⊝ ⊞ ⊟ ⊠ ⊡ ⊢ ⊣ ⊤ ⊥ ⊦ ⊧ ⊨ ⊩ ⊪ ⊫ ⊬ ⊭ ⊮ ⊯ ⊰ ⊱ ⊲ ⊳ ⊴ ⊵ ⊶ ⊷ ⊸ ⊹ ⊺ ⊻ ⊼ ⊽ ⊾ ⊿ ⊿
```

比如以下关于符号的颜色设置的示例:

```
1 op <- par(bg = "light blue")
2 x <- seq(0, 2*pi, len = 51)
3 plot(x, sin(x), type = "o", bg = par("bg"))
4 points(x, sin(x), pch = 21, cex = 1.5, bg = "red")
```

87. 如何在已有图形上加一条水平线

使用低水平绘图命令 `abline()`，它可以作出水平线（y 值 $h=$ ）、垂线（x 值 $v=$ ）和斜线（截距 $a=$ ，斜率 $b=$ ）。

R 中的绘图命令可以分为“高级”（High_level）、“低级（Low_level）”和“交互式”（Interactive）三种绘图命令。

简要地说，“高级”绘图命令可以在图形设备上绘制新图；“低级”绘图命令将在已经存在图形上添加更多的绘图信息，如点、线、多边形等；使用“交互式”绘图命令创建的绘图，可以使用如鼠标这类的定点装置来添加或提取绘图信息。在已有图形上添加信息当然要使用“低级”绘图命令。

88. 如何做密度曲线？

常用的办法是：做出 x 的一个序列，然后做出 `dfunction(x)`，比如：

```

2 x=seq(-3, 3, .05)
  plot(x, dnorm(x), type="l",
4       ylab = '', main = 'The normal and t distribution')
  lines(x, dt(x,1), col = "red")
  legend('topright', c('normal', 't'),
6       col = c('black', 'red'), lty = 1)

```

`dfunction(x)` 中的 `function` 是指分布族，可以参考 R-intro 中的 [Probability distributions](#) 章节，或 `help.search("distribution")`。关于构造相关分布函数参考第 20 页中“如何模拟高斯分布数据”。

R 中的分布函数

分布	R 函数	附加参数	默认参数
beta	beta	shape1(α),shape2(β)	
二项	binom	size(n),prob(p)	
χ^2	chisq	df	
均匀	unif	min(a),max(b)	$min = 0, max = 1$
指数	exp	rate	$rate = 1$
F	f	df1(r_1),df2(r_2)	
伽玛	gamma	shape(α),scale(θ)	$scale = 1$
超几何	hyper	$m = N_1, n = N_2, k = n$	
正态	norm	mean(μ),sd(σ)	$mean = 0, sd = 1$
泊松	pois	lamda(λ)	
t	t	df	
威布尔	weibull	shape(α),scale(θ)	$scale = 1$

89. 如何加图例？

绘制图形后，使用 legend函数，help("legend")

```
with(iris , plot(Sepal.Length , Sepal.Width ,  
                pch=as.numeric(Species) , cex=1.2))  
legend(6.1 , 4.4 , c("setosa" , "versicolor" , "virginica") ,  
       cex=1.5 , pch=1:3)
```

90. 怎么做饼图?

参考 pie()函数。饼图展示数据的能力较差，因为我们的眼睛对长度单位比较敏感，而对关联区域和角度感觉较差。建议使用条形图 (bar chart) 和点图 (dot chart)。

91. 如何做茎叶图?

参考 stem 函数。

```
stem(faithful$eruptions)
```

92. R 如何做双坐标图?

在 R 中可以通过绘图参数 par(new = TRUE)使得绘制第二个绘图 (high-level plot) 时保留第一个绘图区域，这样两张绘图会重叠在一起，看起来就是双坐标图。下面的例子是在同一张图上绘制 GDP 和失业率 (UR)：

```
1 year <- 1995:2005  
2 x1 <- data.frame(year , GDP = sort(rnorm(11 , 1000 , 100)))  
3 x2 <- data.frame(year , UR = rnorm(11 , 5 , 1))  
4 par(mar = c(5 , 4 , 4 , 6) + 0.1)  
5 plot(x1 , axes = FALSE , type = "l")  
6 axis(1 , at = year , label = year)  
7 axis(2)  
8 par(new = TRUE , mar = c(10 , 4 , 10 , 6) + 0.1)  
9 plot(x2 , axes = FALSE , xlab = "" , ylab = "" ,  
10       col = "red" , type = "b")  
11 mtext("UR(%)", 4 , 3 , col = "red")  
axis(4 , col = "red" , col.axis = "red")
```

或者使用 plotrix 包中，twoord.plot() 函数

```
twoord.plot(2:10 , seq(3 , 7 , by = 0.5) + rnorm(9) ,  
            1:15 , rev(60:74) + rnorm(15) , xlab="Sequence" ,  
            ylab="Ascending values" , rylab="Descending values" ,  
            main="Test of twoord.plot")
```

但不推荐使用双坐标图来进行数据描述，这样很容易造成误解。并且在 R 中做出并排图形作对比很容易，没有必要绘制双坐标图。

93. 常用的绘图设备都有哪些？

R 支持的图形设备有如下几种（参考?Devices）：

R 图形设备		
	名称	描述
屏幕 显示	x11	X 窗口
	windows	Windows 窗口
文件 设备	postscript	ps 格式文件
	pdf	pdf 格式文件
	pictex	供 L ^A T _E X 使用的文件
	png	png 格式文件
	jpeg	jpeg 格式文件
	bmp	bmp 格式文件
	xfig	供 XFIG 使用的图形格式
win.metafile ^a	emf 格式的文件	

^a仅在 Windows 下有效

这里推荐使用 pdf() 函数，因为 pdf 图形格式为矢量绘图格式，且通用性较强。在 2.14.0 版本后，R 提供了 cairo 这种高质量绘图方式，除了矢量绘图以外，cairo 方式下的 png、jpg 格式也可考虑。

94. 如何为绘图加入网格？

使用 grid() 函数，

```
plot(1:3)
2 grid(NA, 5, lwd = 2) # grid only in y-direction
```

95. 如果绘图时标题太长，如何换行？

可以使用 strwrap 函数，这个函数可以将定义段落格式。

```
plot(0, main = paste(strwrap("This is a really long title that
2 i can not type it properly", width = 50 ),
collapse = "\n"))
```

96. 可以打开多个图形设备么？

可以。当打开多个图形设备后，使用 `dev.list()` 察看图形设备的数目（除了设备一），使用 `dev.cur()` 察看当前使用的图形设备，`dev.set()` 改变激活指定的图形设备，`dev.off()` 关闭图形设备。

97. 坐标 y 上的数字如何水平放置？

仍然是绘图参数问题：

```
1 ?par # see las
  plot(0, 0, xaxt="n", type="n", ylim=c(0, 100), las = 1)
3 mtext("35", side = 2, at = 35, line = 1, las = 1)
```

98. 如何做雷达图？

R 里面使用 `stars` 函数来做雷达图。

```
1 stars(state.x77[, c(7, 4, 6, 2, 5, 3)], full = FALSE,
      key.loc = c(10, 2))
```

这里的 `full = FALSE` 参数表示只绘制雷达图的上半部分（反之，绘制整个雷达图）；`key.loc` 参数表示基准图例的位置。

99. 为什么 R 不能显示 8 种以上的颜色？

当绘图参数 `col` 使用数字来代替颜色名时会有这种情形，这是因为 R 内置调色板默认为 8 种颜色：

```
palette()
2 barplot(rnorm(15, 10, 3), col = 1:15)
  palette(rainbow(15))
4 barplot(rnorm(15, 10, 3), col = 1:15)
  palette("default")
```

在 R 中共有 657 种颜色名称可以使用，它们的名称可以通过

```
1 colors()
```

来得到，但事实上有些颜色名称代表的颜色重复，R 中颜色名称只能显示 502 种颜色。当然可以使用函数 `rgb()` 来指定任意色彩。

100. 如何用不同的颜色来代表数据？

高级绘图函数一般都有 `col` 参数可以设置。对于像 `barplot()` 这类图形，可以使用“颜色组”(color sets) 来设置颜色，颜色组包括如下几类：

R 颜色组函数

名称	描述
rainbow()	彩虹色 ()
heat.colors()	红色至黄色 ()
terrain.colors()	绿色、棕色至白色 ()
topo.colors()	深蓝色至浅棕色 ()
cm.colors()	浅蓝到白色, 浅紫色 ()
gray(), grey()	灰色 ()

```
1 x <- 1:10 ; names(x) <- letters[1:10]
2 barplot(x, col = rev(heat.colors(10)))
3 barplot(x, col = gray((1:10)/10));
```

R 中还有一些专门处理“色谱”的包，比如 RColorBrewer 和 colorRamps，尤其是 RColorBrewer 提供了连续型 (sequential)、离散型 (diverging)、定性型 (qualitative) 三种配色方案。

101. 怎样将 R 的颜色同 RGB 对应起来?

参考函数 col2rgb()

```
1 write.table(t(col2rgb(rainbow(7))/255),sep = ",")
```

102. 如何调整所绘图形的大小?

Windows 平台下，正常情况打开绘图窗口，调整窗口大小，点击菜单直接保存，或使用 savePlot() 函数保存；当然也可以事先用

```
1 windows(width = , height = )
```

打开一个定义好大小的窗口，然后绘图；还可以使用 pdf() ,postscript() , png() ,jpeg() ,pictex() 等“后台生成”函数，

```
1 ## start a PDF file
2 pdf("picture.pdf",height=4,width=6)
3 ## your drawing commands here
4 dev.off() ##### close the PDF file
```

这些函数都有设置图形大小的参数；还可以使用

```
dev.copy(device , file="", height , width)
```

命令。

103. 如何模拟布朗运动?

布朗运动可以用标准正态的随机模拟值的累积和来模拟:

```
1 # two dimensions
  n <- 100
3 x <- cumsum(rnorm(n))
  y <- cumsum(rnorm(n))
5 plot(x, y, type = 'l')
```

104. 如何获得连接若干点的平滑曲线?

如果已知做出这些点的函数可以使用 `curve(expr, from, to, add = T)` 函数。反之, 使用样条差值函数 `spline(x, y, n=)`, 如:

```
1 x <- 1:5
  y <- c(1,3,4, 2.5,2)
3 plot(x, y)
  sp <- spline(x, y, n = 50)
5 lines(sp)
```

105. 网格 (lattice) 绘图和普通绘图有什么区别?

网格 (lattice) 绘图实际上是 S-plus 中 Trellis 绘图在 R 中的实现, 是多元数据可视化的方法。网格绘图相对于普通绘图来说, 是一种拥有“固定格式”的绘图方式, 当然它相对来说较难修改。如果数据分属不同的类别, 需要将这些类别下的数据进行比较, 网格绘图是很不错的选择:

```
1 library(lattice)
  histogram(~height | voice.part, data = singer)
```

常用的 lattice 绘图函数有:

常用 lattice 绘图函数

函数	说明
<code>xyplot(y~x)</code>	双变量散点图
<code>dotplot(y~x)</code>	Cleveland 点图 (逐行逐列累加图)
<code>barchart(y~x)</code>	y 对 x 的条形图
<code>stripplot(y~x)</code>	一维图, x 必须是数值型, y 可以是因子
<code>bwplot(y~x)</code>	箱线图
<code>histogram(~x)</code>	直方图

除了 lattice 绘图外, Hadley Wickham 还通过 `ggplot2` 包实现了 Grammar of Graphics, 这个包综合了 base 和 lattice 两个包的优点。详细的例子可以参考作者的主页 <http://had.co.nz/>

ggplot2/

106. 如何绘制三维图？

参考 `persp()` , `contour()` 函数。这里需要注意的三维绘图中第三维坐标的形式。参考第 18 页中的 `outer()` 函数。

107. 想把一个数值的矩阵映射为一个颜色方格的矩阵，什么函数？

参考 `image()` 和 `filled.contour()` 函数：

```
1 x <- y <- seq(-10, 10, length=50)
2 f <- function(x,y){
3     r <- sqrt(x^2 + y^2)
4     10*sin(r)/r
5 }
6 z <- outer(x , y ,f )
7 image(x , y , z )
8 filled.contour(x ,y ,z )
```

108. 散点图中散点大小同因变量值成比例如何画？

在 R 中做这类图很简单，因为 R 的很多绘图参数可以使用变量：

```
1 x <- 1:10
2 y <- runif(10)
3 symbols(x,y,circles = y/2 ,inches = F,bg = x)
```

109. 我想为一个数据框的每一列都做 Q-Q 图？

使用 `apply()` 函数作用于矩阵的行或列，且能避免 R 中的显式循环

```
1 table <- data.frame(x1 = rnorm(100) ,x2 = rnorm(100, 1, 1))
2 par(ask = TRUE) # wait for changing
3 results = apply(table , 2, qqnorm)
4 par(ask = FALSE)
```

110. 如何在一个直方图上添加一个小的箱线图？

在直方图的空白位置添加另外的小图（像图例一样），仍然使用参数 `par()`：

```
1 x <- rnorm(100)
2 hist(x)
3 op <- par(fig=c(.02,.5,.5,.98), new=TRUE)
4 boxplot(x)
```

111. 如何在 R 的绘图中加入数学公式或希腊字符?

参考?plotmath，熟悉 L^AT_EX 的用户，会发现二者语法非常类似。

```
1 x <- 1:10 ; plot(x, type = "n")
2 text(3,2,expression(paste("Temperature (", degree, "C) in 2003")))
3 text(4,4,expression(bar(x) == sum(frac(x[i], n), i==1, n)))
4 text(6,6,expression(hat(beta) == (X^t * X)^{.1} * X^t * y))
5 text(8,8,expression(z[i] == sqrt(x[i]^2 + y[i]^2)))
```

112. 如何在条形图上显示每个 bar 的数值?

如果明白 barplot() 函数其实是由低级绘图命令 rect() 函数构造的，那下面的例子也就不难理解了：

```
1 x <- 1:10 ; names(x) <- letters[1:10]
2 b <- barplot(x, col = rev(heat.colors(10)))
3 text(b, x, labels = x, pos = 3)
```

113. 如何绘制椭圆或双曲线?

根据函数式的基本绘图。直角坐标系下可使用参数方程：

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 = 1 \implies x = a \sin \theta, y = b \cos \theta, 0 < \theta < 2\pi$$

```
1 t <- seq(0,2*pi,length = 100)
2 x <- sin(t) # a=1
3 y <- 2*cos(t) # b=2
4 plot(x,y, type = 'l')
```

114. 在 word 里如何使用 R 生成的高质量绘图?

矢量绘图的效果是最好的，比如 eps、pdf，而不是位图（png、jpg、tiff 等）。在 word 里面，可以使用 eps，虽然在屏幕上显示不是很好，但打印效果却不错。

I 统计模型

115. 有没有直接计算峰度和偏度的函数?

当然自己写一个也费不了太多时间。FBasics 包中提供了

```
1 skewness()
2 kurtosis()
```


可以直接计算偏度和峰度。

116. 如何做交叉列联表?

table() 函数。table(x) 为 x 的频数表；table(x,y) 为交叉列联表。

```
2 x <- with(airquality, table(cut(Temp, quantile(Temp)), Month))
  prop.table(x,1)
```

117. 如何做线性回归模型?

线性模型是最核心的经典统计方法，且至今仍然有广泛应用；很多现代统计方法都是在此基础上发展起来的。最简单的线性回归模型为：

$$y_i = \alpha + \beta x_i + \epsilon_i$$

其中 α 为截距项， β 为模型的斜率， ϵ 为误差项。

lm() 函数提供了线性回归的计算方法。

```
lm.swiss <- lm(Fertility ~ ., data = swiss)
```

lm() 的结果是一个包含回归信息的列表，它包含以下信息：

- coefficients: 回归系数（矩阵）
- residuals: 返回模型残差（矩阵）
- fitted.values: 模型拟合值

...: ...

可以使用如下命令得到列表名称：

```
1 names(lm.swiss)
```

summary() 和 anova() 分别返回回归模型的概要信息和方差分析表。

```
1 # the same as summary.lm()
  summary(lm.swiss)
3 anova(lm.swiss)
```

提取模型信息的类函数有很多，其他可以参考 R-intro 中 Statistical models in R 一节。

如果处理数据的量很大，可以使用 biglm 包中的 biglm() 函数。这个函数可以用于“海量”数据的回归模拟。

118. 如何更新模型?

参考 update() 函数：

```
1 summary(f0 <- lm(Fertility ~ ., data = swiss))
  f1 <- update(f0, .~. - Examination)
3 summary(f1)
```

119. 如何使用逐步回归?

在 R 里, 可以使用计算逐步回归的 `step()` 函数。它以计算 AIC 信息统计量为准则, 选取最小的 AIC 信息统计量来达到逐步回归的目的。

```
1  utils::example(lm)
   step(lm.D9)
```

`step` 函数可使用 “both,forward,backward” 三种方法, 其默认为 “backward”。当然你还可以参考 `add1`, `drop1` 函数。

120. R 中如何实现分位数回归 (Quantile Regression)

参考 `quantreg` 和 `quantregForest` 包

```
2  data(engel)
   plot(log10(foodexp) ~ log10(income), data = engel,
        main = "'engel' data (log10 - transformed)")
4  taus <- c(.15, .25, .50, .75, .95, .99)
   rqs <- as.list(taus)
6  for(i in seq(along = taus)) {
   rqs[[i]] <- rq(log10(foodexp) ~ log10(income),
8         tau = taus[i], data = engel)
   lines(log10(engel$income), fitted(rqs[[i]]), col = i+1) }
10 legend("bottomright", paste("tau = ", taus), inset = .04,
        col = 2:(length(taus)+1), lty=1)
```

121. 如何得到一个正态总体均值 μ 的区间估计?

很简单, `t.test()` 函数

```
1  x <- rnorm(100)
   t.test(x)
```

122. 如何做聚类分析?

K 均值聚类 (`kmeans()`):

```
2  x <- rbind(matrix(rnorm(100, sd = 0.3), ncol = 2),
   matrix(rnorm(100, mean = 1, sd = 0.3), ncol = 2))
   cl <- kmeans(x, 2, 20)
4  plot(x, col = cl$cluster, pch=3, lwd=1)
   points(cl$centers, col = 1:2, pch = 7, lwd=3)
6  segments(x[cl$cluster==1,][,1], x[cl$cluster==1,][,2],
```

```

      cl$centers[1,1], cl$centers[1,2])
8  segments( x[cl$cluster==2,][,1], x[cl$cluster==2,][,2],
      cl$centers[2,1], cl$centers[2,2],
10  col=2)

```

层次聚类 (hclust()):

```

n <- seq(1,50,by = 4)
2  (x <- USArrests[n,]) # print()
hcl <- hclust(dist(x), method = "complete")
4  hc2 <- hclust(dist(scale(x)), method = "complete")
hc3 <- hclust(dist(x), method = "ave")
6  layout(matrix(c(1,1,2,3),nrow = 2,byrow = T))
plot(hcl); plot(hc2); plot(hc3)

```

聚类过程中我们可能只需要对象的分类信息，那么使用 cutree() 函数也是不错的选择：

```

1  cutree(hc, k = 1:3)

```

当然还有专做聚类的包：cluster

```

1  library(cluster)
clusplot(x, pam(x, 2)$clustering )

```

123. 如何做主成分分析？

stats 包中的 princomp 函数（特征根求解），以及 prcomp 函数（奇异值分解）

```

(pc.cr <- princomp(USArrests, cor = TRUE))
2  plot(pc.cr , type = "lines" # or "barplot"
      ) # or screeplot
4  loadings(pc.cr)

```

princomp() 中的参数 cor = TRUE 表示使用样本相关矩阵作主成分分析，反之使用样本协方差矩阵。loadings() 返回因子荷载。screeplot() 绘制碎石图。

124. 怎样做因子分析？

在 R 中，使用 factanal() 函数对矩阵进行极大似然因子分析。

```

example(factanal)

```

125. 如何对样本数据进行正态检验？

比较常见的方法：shapiro.test()，ks.test()(Kolmogorov-Smirnov 检验)，jarque.bera.test()

(需要 tseries 包)。或者参考专门用作正态检验的 normtest 包，fBasics 包中的相关函数。这几个包（包括基础包）大概提供了十几种检验函数。

126. 如何做配对 t 检验？

参考 t.test() 中的 paired 参数。

```
1 require(stats)
  ## Student's paired t-test
3 t.test(extra ~ group, data = sleep, paired = TRUE)
```

这里需要注意的是数据的录入形式（主要区别于 SPSS）：

extra	group
0.7	1
-0.6	1
...	...
4.6	2
3.4	2

事实上如果你熟悉统计检验的话，你完全可以使用

```
1 apropos("test")
```

来返回所有关于“检验”的信息。比如一些常用的检验，见下表：

bartlett.test	方差齐次性检验	binom.test	二项检验
chisq.test	χ^2 检验	cor.test	相关性检验
fisher.test	Fisher 精确检验	friedman.test	Friedman 秩和检验
kruskal.test	Kruskal-Wallis 秩和检验	mcnemar.test	McNemar 检验
pairwise.t.test	均值的多重比较	PP.test	Phillips-Perron 检验
var.test	方差比检验	wilcox.test	Wilcoxon 秩和检验

尽情享受吧！

127. R 如何做结构方程模型？

参考 sem 包。

128. 多项式回归应该使用什么函数？

使用 I()，例如：

```
1 lm(y ~ x + I(x^2) + I(x^3))
```

129. 如何使用方差分析 (ANOVA)?

方差分析同线性回归模型很类似，毕竟它们都是线性模型。最简单实现方差分析的函数为 `aov()`，通过规定函数内公式形式来指定方差分析类型：

方差分析	
<code>aov(x ~ a)</code>	单因素方差分析
<code>aov(x ~ a + b)</code>	没有交互作用的双因素方差分析
<code>aov(x ~ a + b + a:b)</code>	有交互作用的双因素方差分析
<code>aov(x ~ a*b)</code>	同上

130. 如何求解没有常数项的线性回归模型?

只需在公式中引入 0 即可：

```
1 result <- lm(smokes ~ 0 + male + female , data = smokerdata)
```

131. 如何计算回归模型参数的置信区间?

参考 `confint` 函数，`glm` 模型和 `nls` 模型可参考 MASS 包中的 `confint.glm` 和 `confint.nls` 函数。

```
1 fit <- lm(100/mpg ~ disp + hp + wt + am, data = mtcars)
2 confint(fit)
3 confint(fit, "wt")
```

132. 岭回归的命令是?

参考 MASS 包中的 `lm.ridge()` 函数。

```
1 data(longley) # not the same as the S-PLUS dataset
2 names(longley)[1] <- "y"
3 lm.ridge(y ~ ., longley)
4 plot(lm.ridge(y ~ ., longley ,
5         lambda = seq(0,0.1,0.001)))
6 select(lm.ridge(y ~ ., longley ,
7         lambda = seq(0,0.1,0.0001)))
```

当然，其他类型的 Regularized and Shrinkage Methods 在 R 里也有对应扩展包，请读者自行搜索。

133. logistic 回归相关函数是?

logistic 回归是关于响应变量为 0-1 定性变量的广义线性回归问题，这里需要使用广义线性模型 `glm()` 函数，且广义线性模型的分布族为二项分布。

广义线性模型中的常用分布族

分布	函数	模型
高斯 (Gaussian) ^a	$E(y) = x^T \beta$	普通线性模型
二项 (Binomial)	$E(y) = \frac{\exp(x^T \beta)}{1 + \exp(x^T \beta)}$	Logistic 模型和概率单位 (probit) 模型
泊松 (Poisson)	$E(y) = \exp(x^T \beta)$	对数线性模型

^a正态 (Normal)

高斯 (正态) 分布族的广义线性模型事实上同线性模型是相同的, 即

```
1 | fit1 <- glm(formula, family = gaussian, data)
```

同线性模型

```
1 | fit1 <- lm(formula, data)
```

得到的结论是一致的, 当然效率会差很多。

134. 如何使用正交多项式回归?

我们考虑回归方程:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \dots + \beta_k x_i^k, i = 1, 2, \dots, n,$$

当多项式的次数 k 比较大时, x, x^2, \dots, x^k 会出现线性相关问题。故需要使用正交多项式回归来克服这方面的缺点。在 R 中, 使用 `poly()` 函数:

```
1 | (z <- poly(1:10, 3))
```

135. 如何求帽子矩阵?

参考 `hat()`, `hatvalues()` 函数。

136. D-W 检验在哪里?

`car` 包中的 `durbin.watson` 函数, `lmtest` 包中的 `dwtest` 函数。

```
1 | help.search("Durbin-Watson")
```

137. 如何求 Spearman 等级 (或 kendall) 相关系数

`cor()` 函数默认为求出 Person 相关系数, 修改其 `method` 参数即可求得 Kendall τ 和 Spearman 秩相关系数。

```
1 | cor(longley, method = "spearman")
```

名称	方法	用途（条件）
Pearson	线性	正态总体假定
Kendall τ	协同	非参数检验
Spearman	样本秩	非参数检验

138. 如何做 Decision Tree?

基于树型方法的模型（Tree-based model）并不被统计学背景的研究者所熟悉，但在其他领域却时常被广泛应用。下面是 Modern Applied Statistics With S 中的例子，需要加载 rpart 包。

```

1 library(rpart)
  set.seed(123)
3  cpus.rp <- rpart(log10(perf) ~ ., cpus[, 2:8], cp = 1e-3)
  plot(cpus.rp, uniform = T)
5  text(cpus.rp, digits = 3)

```

更多关于 R 做数据挖掘的方法请参考 中国第一届 R 语言会议 的报告。

139. 如何使用时间序列相关模型?

假设 ϵ_t 是一组均值为 0，方差为 σ^2 的不相关的序列，那么我们定义 q 阶滑动平均模型为

$$X_t = \sum_0^q \beta_j \epsilon_{t-j}$$

p 阶自回归模型：

$$X_t = \sum_1^p \alpha_i X_{t-i} + \epsilon_t$$

定义 $ARMA(p, q)$ 过程为

$$X_t = \sum_1^p \alpha_i X_{t-i} + \sum_0^q \beta_j \epsilon_{t-j}$$

我们将差分后的 $arma$ 模型称为 $arima$ 模型，甚至更一般的季节性 $arima$ 模型。在 R 中使用

```
arima(x, order = c(0,0,0), seasonal = list(order = c(0, 0, 0)))
```

对该模型进行拟合：

```

1 require(graphics)
  (fit1 <- arima(presidents, c(1, 0, 0)))
3  tsdiag(fit1)

```

140. box-cox 变换?

MASS 包中的

```
1 | boxcox()
```

函数。

141. 检验异方差的 Breusch-Pagan 检验?

lmtest 包中的 bptest() 函数，或者利用 car 包中的 ncv.test() 函数

142. 如何做判别分析?

参考 MASS 包中的 lda() 函数 (Fisher Linear Discriminant Analysis) 和 qda() 函数。

143. 计算 OLS 有没有简便方法?

有，可以使用函数 qr.solve()，

```
1 | qr.solve(X, y)
```

等价于 $(X'X)^{-1}X'y$

144. 如何进行典型相关分析?

典型相关分析是用于研究两组随机变量之间的相关性的一种统计方法。R 提供了 cancor() 函数进行相关计算。

```
1 | pop <- LifeCycleSavings[, 2:3]
   | oec <- LifeCycleSavings[, -(2:3)]
3 | cancor(pop, oec)
```

145. 如何使用 R 做生存分析?

需要加载 survival 包，Kaplan-Meier 估计：

```
1 | # fit a Kaplan-Meier and plot it
   | fit <- survfit(Surv(time, status) ~ x, data=aml)
3 | plot(fit)
   | # life table
5 | cbind(fit$time, fit$n.risk, fit$n.event, fit$surv)
```

注意 survfit 函数中分析方法 type 中有 “kaplan-meier”, “fleming-harrington”, “fh2” 三种方法可以选择。

以及生存分析的核心 -COX 比例风险模型：

```
1 | ## Fit a time transform model using current age
   | coxph(Surv(time, status) ~ ph.ecog + tt(age), data=lung,
```



```
3 tt = function(x,t,...) pspline(x + t/365.25))
```

J 其他

146. 如何加速 R 的运行速度

这个问题的范围非常大，针对于不同的目标会有不同的解决方案，这里仅仅介绍一种较为简单的并行化 R 的方式 – 使用 parallel 包：

```
1 library(parallel)
  doit <- function(x)(x)^2 + 2*x
3 system.time(res <- lapply(1:5000000, doit))
  rm(res)
5 gc()
  cl <- makeCluster(getOption("cl.cores", 3))
7 system.time(res <- parLapply(cl, 1:5000000, doit))
  stopCluster(cl)
```

147. R 有类似于 SPSS 的界面么？

有！安装包 Rcmdr，加载包后，使用命令

```
Commander()
```

调出可供使用的图形使用界面。由于这个图形使用界面需要若干基础包外的其他函数，故还需要包 car、effects、abind、lmtest、multcomp、relimp、RODBC、rgl 的支持。

还有用于数据挖掘的界面环境 rattle，它提供了整套的图形界面的数据挖掘环境。有兴趣的读者可自行安装尝试。

148. 怎样来计算函数运行使用时间？

使用 system.time()。proc.time() 可以获得 R 进程存在的时间，system.time() 通过调用两次 proc.time() 来计算函数运行的时间。

一般来说秒级以上的时间计算可以使用 system.time() 函数，但更短的则需要使用两次 Sys.time()，来精确显示程序所花的时间了。

149. 在 R 中如何处理地图数据？

R 提供了 maps 和 mapdata 两个包来绘制地图，其中 mapdata 提供了中国地图的相关信息：

```
1 library(mapdata)
  map("china")
```

不过可惜，这种方法得到的中国地图没有重庆的行政区划，且各省的名称都是用数字拼装而成，不能用 map 包中的函数像对

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

一样进行进一步加工。

不过如果你熟悉地理数据，那么 maptools 包将是一个不错的选择。她可以读取、处理空间对象，且提供了同 PBSmapping, spatstat, maps, RArcInfo, Stata tmap, WinBUGS, Mondrian 这类包的封装接口。

150. 如何释放 R 运行后占用的内存？

使用函数

```
gc()
```

因为 R 是在内存中运算，所以当 R 读入了体积比较大的数据后，即使删除了相关对象，内存空间仍不能释放。gc() 函数虽然主要用来报告内存使用情况，但是一个重要的用途便是释放内存。

151. 用什么 IDE(集成开发环境) 比较好？

工欲善其事，必先利其器。方便快捷使用的集成代码编辑环境能能给使用者良好的体验。一般来说，任意一款文本编辑器都可以，比如 Windows 桌面环境下的记事本，Linux 下面的 Vi 等编辑器。但这些编辑器比较简单，并不支持诸如项目管理、语法高亮、自动补全等功能。下面介绍一下常用的 R 集成开发环境。

R 环境下常用的 IDE

函数	运行环境	说明	推荐
ESS	All	Emacs Speaks Statistics, 号称神的编辑器	no
RStudio	All	四栏显示, 支持语法高亮, 自动补全	yes
UltraEdit	All	商用软件, 语法高亮和自动补全功能, 功能相对单一	no
Eclipse	All	实际使用StatET的插件, 习惯使用 Eclipse 环境的人士可以考虑使用	no
Notepad++	Windows	使用 NpptoR 扩展同 R 交互	no
Tinn-R	Windows	代码编辑和 R Console 分离, 支持语法高亮等功能	no
RWinEdt	Windows	依附于 WinEdt 的插件, 功能较为简单	no

RStudio 和 R 语言一样是开源项目，而且无论从安装还是使用上都非常简单。除了桌面版本外，RStudio 还提供服务器版本。

152. Sweave 是用来做什么的？

Sweave 提供了一种为“混排 $\text{T}_{\text{E}}\text{X}$ 文本和 S 编码”生成文档的机制。单个的 Sweave 文档中既包含 $\text{T}_{\text{E}}\text{X}$ 文本又包含 S 编码，通过编译最终形成的文档包含：

- T_EX 文档的编译输出;
- S 编码和 (或);
- S 编码的代码输出 (文本、图形)。

如果想了解更多, 请参考Sweave User Manual, 或参考附录A: Sweave 的实例。
它的文档形成过程:

Sweave 文档 $\xrightarrow{\text{Sweave(in R)}}$ T_EX 文档 $\xrightarrow[\text{dvi2pdf}]{\text{L^AT_EX}}$ 最终 pdf 文档

现在还可以使用谢益辉博士开发的 knitr 包来处理这种类似的混排文档, 当然 knitr 支持的类型更多。

A Sweave 例

```
\documentclass[CJK]{cctart}
\usepackage{verbatim}
\title{Sweave 实例}
\author{}
\date{}
\SweaveOpts{echo=FALSE}

\begin{document}

\maketitle
```

使用 Sweave 可以很容易地将 \LaTeX 同 R 的代码混排文档转化为可编译的 \LaTeX 文档。

在这种混排的文档里，基本结构仍然是 \LaTeX 形式的，唯一的区别是，R 代码需要放置在以 $\langle\langle\rangle\rangle=$ 为开头， $\$0\$$ 为结尾的段落里面。开头部分有两个常用的参数：`echo` 和 `fig`，使用逻辑值分别表示是否将 R 代码输入作为 \LaTeX 文本输出；是否在 \LaTeX 文档中绘制图形。这篇文档只需要在 R 中编译一遍，即可形成 \LaTeX 需要的输出（文件）。

下面是一个配对 t 检验的一个例子：

```
<<echo = TRUE>>=
require(stats)
## Student's paired t-test
m <- t.test(extra ~ group, data = sleep, paired = TRUE)
print(m)
@
```

R 在计算过程中生成的中间结果很容易插入到标准文档，比如 `\texttt{sleep}` 数据的双样本的配对 t 检验结果中的 `$p-value` 是 `\Sexpr{format.pval(m$p.value)}`；或者是直接运算

```
<<echo=TRUE,results=hide>>=
choose(49,6)
@
```

美国威力球（类似于福彩双色球）的理论组合数等于 `\Sexpr{choose(49,6)}`。

通过这种方法处理有大量计算的文档，比 word 不知方便多少倍。

R 代码中可以随意写注释，但这些注释默认不会被输出。如果要求输出注释，抱歉，现在还没有更好的解决办法。

使用 Sweave 还可以将 R 生成的图形加入到 \LaTeX 文档中，而不必事先做出 \LaTeX 需要的图形文件 $\footnote{\text{Sweave 会自动生成 ps 和 pdf 图形}}$ 。

下图是根据 Titanic 号海难中人员的经济状况、性别、年龄和是否存活四个变量绘制的马赛克图：

```
<<fig=TRUE,echo=FALSE>>=
require(graphics)
mosaicplot(Titanic, main = "Survival on the Titanic")
@

\end{document}
```

Copyright ©2012 R and all the Contributors to R FAQ. All rights reserved.

R 以及 R FAQ 的作者拥有版权 ©2012。保留所有权利。

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with the Invariant Sections being Contributors, no Front-Cover Texts, and no Back-Cover Texts.

你可以拷贝、发布或者修改这份文档，但必须遵守自由软件组织颁布的 GNU 自由文档许可证 1.2 或者以后版本的条款。Invariant Sections 包括 Contributors，没有 Front-Cover Texts 和 Back-Cover Texts。

Index

Symbols	
\\	11
.....	21
.GlobalEnv	12
.packages	6
/	11
%*%	18
^	19, 35
{}	14
A	
abline	24
aggregate	15
any	13
aov	36
aperm	14
append	12
apply	14, 19, 30
as.Date	22
as.numeric	25
as.POSIXct	22
axes	25
axis	25
B	
barplot	27, 28
boxcox	39
bptest	39
Breusch-Pagan	39
C	
cancor	39
capture.output	9
car	8, 39
casefold	21
choose	19
citation	3
clipboard	10
cm.colors	28
col	27
col2rgb	28
colMeans	19
colors	27
combn	19
Commander	40
complex	18
confint	36
confint.glm	36
confint.nls	36
contour	30
crossprod	18
cumsum	14, 29
curve	29
cutree	34
D	
D	19
data	7
data frame	7, 14
demo	3
detach	6
dev.copy	28
dev.cur	27
dev.list	27
dev.off	27
dev.set	27
Devices	26
diag	18
difftime	22
duplicate	13
durbin.watson	37
dwtest	37
E	
edit	10
eigen	18
ESS	41
eval	12
F	
factanal	34
factorial	19
FALSE	14
file.choose	11
filled.contour	30
fivnum	17
fix	10
format	22
G	
gc	41
getAnywhere	8
getwd	6
glm	36
gray	28
grep	21
grey	28
grid	26
H	
hat,hatvalues	37
hclust	34
head	8
heat.colors	28
help	4
help.search	4

I		mar	23	png	28
I	35	matrix	14, 19	points	23
identical	13	memory.limit	4	poly	37
if	14	merge	16	postscript	28
image	30	methods	8	princomp	34
integer	17	N		print	10
integrate	17	nchar	22	proc.time	40
iris	25	ncv.test	39	prompt	5
is.na	12	NULL	14	Q	
J		O		qda	39
jarque.bera.test	35	optimize	20	qqnorm	30
jpeg	28	options	5	qr.solve	39
K		outer	18	Quantile Regression	33
kmeans	33	P		R	
knitr	42	Package	6	rainbow	28
ks.test	34	cluster	34	rattle	40
L		plotrix	25	read.table	10
latex	10	Rcmdr	40	read.xls	9
latex.table	10	rpart	38	rect	31
layout	23	sem	35	Regular Expressions	14, 21
lda	39	stats	34	rev	14
legend	25	tseries	35	rgb	27
letters	28	par	23, 25, 30	rm	5
Library	6	parallel	40	RMySQL	11
library	5, 6	parplot	31	rnorm	20, 25, 29
lines	29	parse	12	RODBC	11
list	5	paste	5, 21	rowMeans	19
lm	36	pch	25	RSiteSearch	4
lm.ridge	36	pdf	28	S	
lmttest	39	persp	30	sample	15
loadings	34	pictex	28	sapply	15
lower.tri	19	pie	25	save	6
ls	5	plotmath	31	save.image	6
M		pmax	14	savePlot	28
mai	23	pmin	14	scale	16

screepplot	34	step	33	toupper	21
search	6	strwrap	26	transform	14
setwd	6	subset	12	twoord.plot	25
shapiro.test	34	substr	21		
shell.exec	5	survfit	39	U	
show.error.messages	5	Sweave	41	unique	13
sink	9, 10	system	5	uniroot	20
solve	18	system.time	19, 40	update	32
some	8	T		update.packages	5
sort	25	t	17	upper.tri	19
source	10	t.test	33	W	
spline	29	table	32	windows	28
split.screen	23	tail	8	with	25
sqldf	19	tapply	15		
sqrt	19	terrain.colors	28	X	
stars	27	tolower	21	xlsReadWrite	9
stem	25	topo.colors	28	xtable	10