用 R 语言做数据分析

用 R 做数据分析, siqin.hou@gmail.com 整理，资料来自互联网：http://www.itongji.cn/

目录

用 R 语言做数据分析……1

用 R 语言做数据分析——1.R 简介……4

一、R 语言介绍……4

二、 R 软件安装下载……4

三、 R 语言的特点……5

四、 如何学学习 R 语言……5

五、 我对 R 语言的理解和看法……6

用 R 语言做数据分析——2.R 包介绍……7

一、 R 语言包的安装……7

二、 加载包……8

三、 查看包的相关信息……8

四、 R 包的分类……9

用 R 语言做数据分析——3.向量……11

一、向量基本概念……11

二、几种特殊向量的生成……12

三、向量运算……13

vectorized operation（向量化运算）……14

用 R 语言做数据分析——4.矩阵……15

1. 矩阵的创建……15

2. 矩阵的运算 15

用 R 语言做数据分析——5.数据框 18

数据框的创建 18

数据框基本操作 19

用 R 语言做数据分析——6.列表 22

列表特点 23

列表基本操作 23

用 R 语言做数据分析——7.数据输入与输出之 Scan 函数 26

scan() 函数 26

一、主要的参数说明如下： 26

二、SCAN 几点注意： 27

三、案例说明 27

用 R 语言做数据分析——8.数据输入与输出之 READ 函数 29

read.table() 函数 29

read.fwf ( ) 函数 30

w <- readline()函数 31

Readlines() 函数 31

R 语言入门基础教程 33

R 语言入门基础教程-01.常用运算函数 34

R 语言入门基础教程-02.数据框 36

数据框(data frame) 36

R 语言入门基础教程-03.因子和有序因子 38

因子(factor)和有序因子(ordered factor) 38

R 语言入门基础教程-04.数组和列表……40

数组(array)……40

列表(list)……40

R 语言入门基础教程-05.矩阵……42

矩阵(matrix)……42

R 语言入门基础教程-06.向量……44

向量（vector）……44

R 语言学习由浅入深路线图……46

1.初级入门……47

2.高级入门……47

3.绘图与可视化……47

4.计量经济学……48

5.时间序列分析……48

6.金融……49

7.数据挖掘……49

8.附注……49

用 R 语言求置信区间……50

用 R 语言做数据分析——1.R 简介

发表于 2013-04-18 14:25 来源：http://www.itongji.cn/

R 语言是由 Ross Ihaka、Robert Gentleman 二位创建的，这也许可以解释为什么叫 R 语言。现在由“R 开发核心团队”负责开发。R 是基于 S 语言的一个 GNU 项目。

一、R 语言介绍

R 是为统计计算和作图的一门语言和环境。是一个 GNU 项目，和 S 语言和环境很相似，S 语言是由 BELL 实验室的 John Chambers 和他的同事开发的。R 语言可以认为是从 S 语言衍生而来的，他们之前有很重要的不同，但是大多数用 S 语言写的代码也可以在 R 中运行。

目前 R 在高校非常流行，特别是随着这几年互联网的发展，（R 在一些大公司的运用得到的实践，例如：国外的 google、linkdin、facebook 等，国内一些大型互联网公司也在开始使用 R），及随着互联网版权的意识增强，也促使了 R 在互联网的发展。当然 R 在很多领域都有很广泛的运用。

R 语言是开源的，同时可以运行在各种平台上（Linux、Windows、MacOS 等）。R 的许多软件包是由 R 语言、 LaTeX、Java 及最常用 C 语言和 Fortran 撰写。

可以说现在 R 包含各种各样的功能，可以说目前你能想到的功能，都可以找到一个或者多个 R 包来实现。几千个 R 包，哪个才最适合你呢？“最适合你自己的 R 包，也许就是你自己写的那个包”。

二、 R 软件安装下载

CRAN 地址：http://www.r-project.org/，什么是 CRAN：

CRAN 为 Comprehensive R Archive Network（R 综合典藏网）的简称。它除了收藏了 R 的执行档下载版、源代码和说明文件，也收录了各种用户撰写的软件包。现时，全球有超过一百个 CRAN 镜像站。（来源 http://baike.baidu.com/view/942569.htm）

根据你的操作系统，下载相应的 R 语言安装文件。

下载地址：http://ftp.ctex.org/mirrors/CRAN/

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- Download R for Linux
- Download R for (Mac) OS X
- Download R for Windows

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

三、 R 语言的特点

1、变量不需要申明即可引用。

2、R 语言的核心是：向量。

3、R 语言是一个函数语言。

4、向量中的下标是从1开始引用的。

5、R 是脚本语言、面像对象；

四、 如何学学习 R 语言

《R 语言学习由浅入深路线图》 这篇文章大家可以参考，这篇文章简单介绍了一下 R 学习的资料，大家可以根据需要进行参考。那么如何才能学好 R，个人理解有以下几点：

1、不要期望你能学会 R 中所有的包。

2、关键能理解 R 语言的内涵。

多看看 CRAN 上的相关文档，例如：季刊、R 语言相关新闻，特别是每次版本更新的一些内容。

3、运用

如果你是做数据分析相关的工作的，一定要把学习到的 R 语言知识运用到你的工作中，不管你把 R 语言当用一门编程语言还是统计工具，用的多了，自然你就有感觉，很多东西你就记住了。

4、持续

每天花点时间写几条 R 代码，实现一些小功能。如果你工作上就用 R，那是最完美的。

5、多看

多看别人的代码，R 运用案例。可以 google 一下 R 有很多好博客，文章。很多人都是 R 的 GREEK。

6、开放

一定要开放、分享的心态。多与别人交流，不要总是需求，一定要学习给予。（我是我个人观点，如果要真好用到 R，让 R 发挥价值就是必须的。）

7、总结

学到的东西，及时做好总结，可以总结成案例或者笔记，如果可以欢迎分享给大家（http://www.itongji.cn/member/article_add.php 到这投稿是一个不错的选择）

五、 我对 R 语言的理解和看法

随时互联网的发展，特别是互联网对于版权、成本的因素考虑，因为免费、开源使越来越多的公司开始用 R 语言来处理数据、分析数据、完成模型等，当然这其中也伴随着对于数据价值挖掘的，特别是在大数据的背景下，想通过对数据挖掘&分析建立自己的竞争优势。

R 不仅 免费还有各种各样的的功能包资源。从某种程度上讲，任何你想要的功能应该都可以找到对应的包，只是说是否完全满足，对于一些算法研究人员来说，可以在原来的代码的基础进行借鉴。这也许就是为什么 R 最开始主要用到高校或者学术领域（当然和国外学者、专家这种自由、开放的环境或者意识有很关系，这也许就是为什么许多开源软件都是国外出来，很少看到国内大公司有什么好东西开源）。

很多行业人士都说 R 是未来的“王道”，就像 unix 的发展过程一样。我觉得未来一定有属于 R 的一片天空，而且这种天空可以说是接近无限。所以，对于有志于从事数据挖掘、数据分析这个行业的朋友来说，掌握 R 是也许会成为未来的必备技能（就像现在数据分析师大多要求会：SQL）。

最后，我对 R 语言的理解与总结可以概括为一句话：“开源、二次加工、分享精神”。

(责任编辑：黑阳)

用 R 语言做数据分析——2.R 包介绍

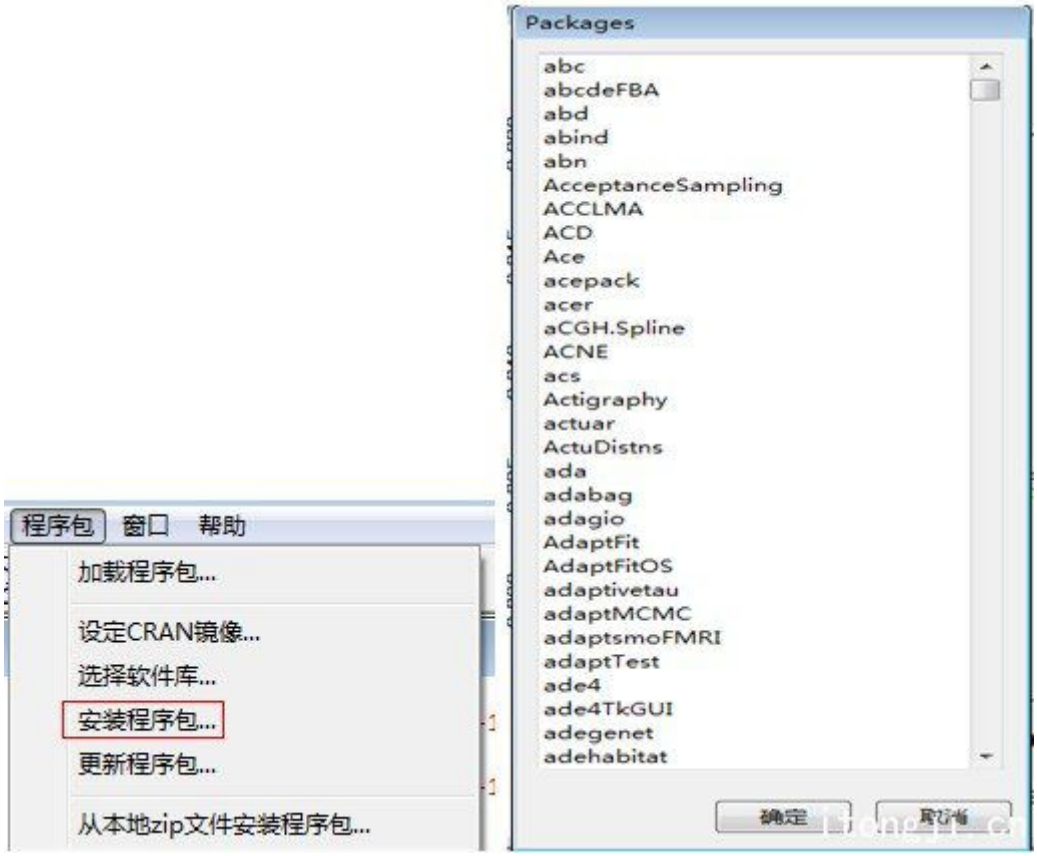
发表于 2013-04-18 14:26 来源：http://www.itongji.cn/

R 语言的使用，很大程度上是借助各种各样的 R 包的辅助，从某种程度上讲，R 包就是针对于 R 的插件，不同的插件满足不同的需求，截至2013年3月6日，CRAN 已经收录了各类包4338个。

一、 R 语言包的安装

1、通过选择菜单：

程序包->安装程序包->在弹出的对话框中，选择你要安装的包，然后确定。



2、使用命令

install.packages("package_name","dir")

package_name:是指定要安装的包名，请注意大小写。

dir:包安装的路径。默认情况下是安装在..\library 文件夹中的。可以通过本参数来进行修改，来选择安装的文件夹。

例如：mvtnorm 安装到 D:/DM/r/R-2.15.2/library/目标下

install.packages("mvtnorm","D:/DM/r/R-2.15.2/library/")

3、本地来安装

如果你已经下载的相应的包的压缩文件，则可以在本地来进行安装。请注意在 windows、unix、macOS 操作系统下安装文件的后缀名是不一样的：

1）linux 环境编译运行：tar.gz 文件

2）windows 环境编译运行 ：.zip 文件

3）MacOSg 环境编译运行:.tgz 文件

注：包安装好后，并不可以直接使用，如果在使用包中相关的函数，必须每次使用前包加载到内存中。通过 library(package_name)来完成。

二、 加载包

包安装后，如果要使用包的功能。必须先把包加载到内存中（默认情况下，R 启动后默认加载基本包），加载包命令：

Library(“包名”)

Require(“包名”)

三、 查看包的相关信息

1、查看包帮忙

library(help=package_name)

主要内容包括：例如：包名、作者、版本、更新时间、功能描述、开源协议、存储位置、主要的函数，例如：library(help=igraph)

2、查看当前环境哪些包加载

find.package() 或者.path.package()

例如：默认情况下安装7个包（本系列文章使用的 R 的版：2.15.2）

```
> .path.package()
[1] "D:/DM/r/R-2.15.2/library/stats"     "D:/DM/r/R-2.15.2/library/graphics"
[3] "D:/DM/r/R-2.15.2/library/grDevices" "D:/DM/r/R-2.15.2/library/utils"
[5] "D:/DM/r/R-2.15.2/library/datasets"  "D:/DM/r/R-2.15.2/library/methods"
[7] "D:/DM/r/R-2.15.2/library/base"
```

3、移除包出内存

detach()

4、把其它包的数据加载到内存中

data(dsname, package="pkgname")

5、查看这个包里的包有数据

data(package="包名")

6、列出所有安装的包

library()

四、 R 包的分类

1、根据包的功能，分成不同的类：

http://cran.r-project.org/web/views/

2、截止到2013年3月6号，CRAN 上一共收录的：4338个包。

3、不同开源协议的包的分布如下：

```
       GPL        GPL-2        GPL-3         LGPL          BSD
      2532         1016          304           85           82
    LGPL-3          MIT    Unlimited         file Artistic-2.0
        48           44           42           39           24
    Apache          CC0     LGPL-2.1          X11       AGPL-3
        18           14           14           13           10
        CC       LGPL-2     CeCILL-2          GNU         AGPL
         9            9            7            6            3
    CeCILL         EUPL          MPL       Common      FreeBSD
         3            3            3            2            2
  Artistic Artistic-1.0      BSL-1.0          EPL      Mozilla
         1            1            1            1            1
   MPL-1.1
         1
```

（相关查看代码：

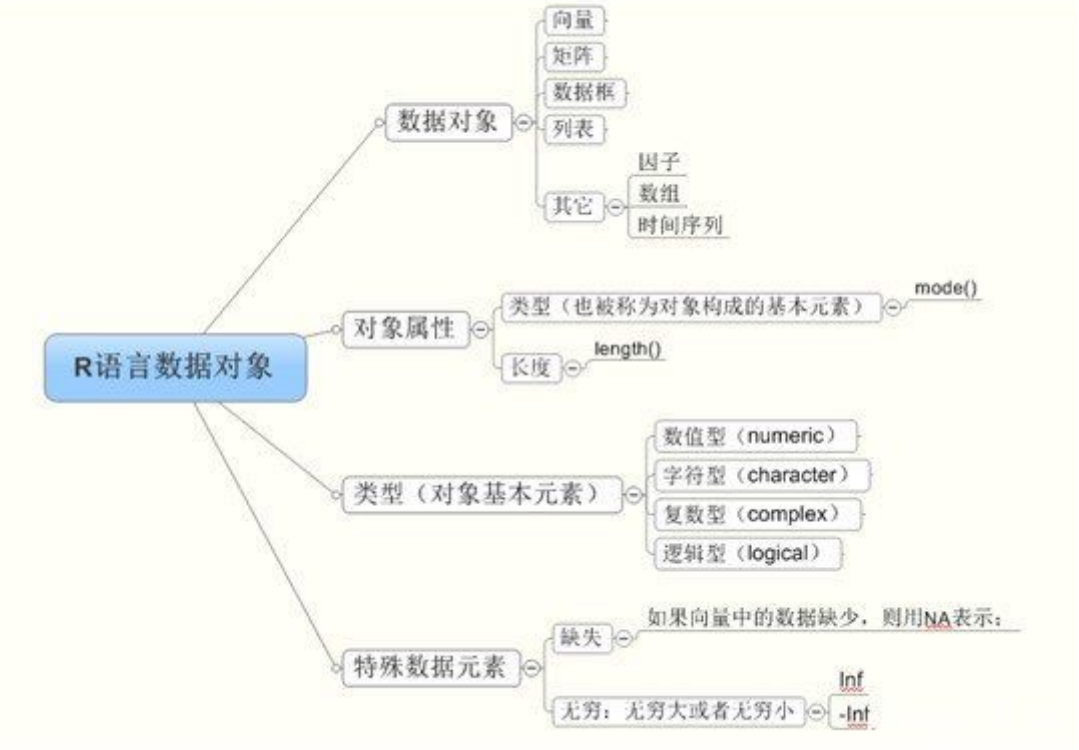
#读取 CRAN 上包的相关信息

```
a <- available.packages(contrib.url("http://ftp.ctex.org/mirrors/CRAN", "source"))

#查看 CRAN 上当前包的个数

nrow(a)

#各种开源协议的包的分布个数

sort(table(gsub(" .*", "", a[,"License"])),decreasing = T)
```

(责任编辑：黑阳)

用 R 语言做数据分析——3.向量

发表于 2013-04-18 14:26 来源：http://www.itongji.cn/



图一 R 语言的数据对象类型简介图

在本系列文章中，接下将分别介绍 R 语言的几种数据对象：向量、矩阵、数据框与列表、其它特殊对象；

请大家在学习 R 语言的时候永远记住一个概念，R 语言所有内容都是对象；

一、向量基本概念

R 语言中最为核心的构成之一

1、向量构成的基本元素为：数值（numeric）、字符（character）、逻辑值（logical）、复数型（complex）；

2、向量不需要定义类型，可直接赋值：

1）生成一个空向量；x<-c();

2）给向量赋值。x<-c(0,1,2,3);

3、向量的元素下标取值是以1开始，这点请刚开始接触 R 的同学注意。例如：要前例中取 x 向量中值为1,则为 x[2]

4、如果一个向量中有一个字符，则该向量的类型会变成字符；

```
> demo_1<-c(1,2,'a')
> mode(demo_1)
[1] "character"
```

如果逻辑变量与数值在一起，则为转换成数值。TRUE 转变成1 and FALSE 转变成 0.

```
> demo_2<-c(1,2,TRUE)
> mode(demo_2)
[1] "numeric"
> demo_2
[1] 1 2 1
```

2、在 R 语言中没有单一的整数、单一字符的概念

例如：X<-2；X<-'a'；R 都是当作向量来处理，只是这个向量只包括单一值。

3、给向量各元素命名: names(x)

Demo_3<-1:3

names(Demo_3) <- c("a","b","ab")

```
>  Demo_3<-1:3
> names(Demo_3) <- c("a","b","ab")
>  Demo_3
 a  b ab
 1  2  3
```

二、几种特殊向量的生成

1、生成系列 seq()

1）seq(length=, from=, to=)

Length:指定生成个数

From:是指开始生成的点

To:截止点

2）如果不指定，则默认条件下：seq(N1,N2,BY=)

N1：开始位置

N2：截止位置

BY=指定间隔

```
> seq(length=10,from=10,to=100)
 [1]  10  20  30  40  50  60  70  80  90 100
> seq(10,100,10)
 [1]  10  20  30  40  50  60  70  80  90 100
```

2、rep(P，N) 重复生成 P 值 N 次

```
> rep(1,10)
 [1] 1 1 1 1 1 1 1 1 1 1
```

三、向量运算

1、取子向量

通过下标:

取某个元素：x[2];(如果 X 包括名称，注意:x[2]与 x[[2]]的区别；

取某中几个：x[c(1,2,0)]

取某个/几个元素，利用-：x[-n]

2、向量排序；

sort(); 输出排序后的结果；

order()；输出排序后的各个向量位置；

a<-c(3,9,0,12,19)

sort(a)；order();

```
> a<-c(3,9,0,12,19)
> order(a)
[1] 3 1 2 4 5
> sort(a)
[1]  0  3  9 12 19
```

3、循环。如果二个向量进行运算过程中，二个向量不相同，则 R 对长度小的向量自动补充值，直到长度等于大的向量。

例如：

Demo_5<-1:3;

Demo_6<-2:5;

Demo_7= Demo_6+ Demo_5;

```
警告信息:
In Demo_6 + Demo_5 : 长的对象长度不是短的对象长度的整倍数
```

Demo_5<-1:3;

Demo_6<-2:7;

Demo_7= Demo_6+ Demo_5;

```
> Demo_5<-1:3;
> Demo_6<-2:7;
> Demo_7= Demo_6+ Demo_5;
> Demo_7
[1]  3  5  7  6  8 10
```

(这也是很多人学习 R 语言的时候很难理解的地方，为什么会有这种机制）

vectorized operation（向量化运算）

1）、如果有一个是向量，是则结果都是向量形式；

2）、进行向量运算；判断向量是否长度相等，如果长度不相等，则把短向量长度从头开始循环补充值，到与长向量的长度相等（所以二个长度的相关必须是整数倍）；

(责任编辑：黑阳)

用 R 语言做数据分析——4.矩阵

发表于 2013-04-18 14:27 来源：http://www.itongji.cn/

1. 矩阵的创建

在 R 中用函数 matrix()来创建一个矩阵，使用该函数时需要输入必要的参数值。matrix(data = NA, nrow = 1, ncol = 1, byrow = FALSE,dimnames = NULL)

1.data 向量元素列表

2.nrow 行数

3.ncol 列数

4.byrow 矩阵是按列优先的方式进行排序，先列后行。byrow 项控制排列元素时优先级按行。例如：

1)matrix(c(1,2,3,4,5,6),nrow=2,byrow=T)

5.Dimnames(Row_name,Col_name)给定行和列的名称,如果不需要给行或者列命名，则以 NULL 代替。例如：给下面的矩阵列命令

2)Demo_1<-matrix(rnorm(15)*10,5,3,dimnames=list(NULL,c('A','B','C')))

2. 矩阵的运算

1、R 矩阵查看矩阵的列/行相关信息

colnames(Demo_1) #查看矩阵列名

rownames(Demo_1) #查看矩阵行名

rownames(Demo_1)<-c('r1','r2','r3','r4','r5') #给矩阵的行命名

dim(Demo_1)　#矩阵的维度

#返回与矩阵相同的列与行

row()/col()函数将返回一个与某矩阵有相同维数的矩阵

#返回矩阵行数与列数

nrow()#返回行数

ncol()#返回列数

2、R 的子矩阵

#取矩阵中某个元素值，第二行第三列的值：

Demo_1[2,3]

#取矩阵中的某列，取矩阵的第一列

Demo_1[,1]

#取矩阵中的某行，取矩阵的第一行

Demo_1[1,]

#取某列大于某个值，取第二列大于 3

Demo_1[Demo_1[,2]>3,]

```
> Demo_1[Demo_1[,2]>3,]
             A         B        C
[1,]  9.069551  9.771795 2.884977
[2,] -3.226222 16.158484 2.487996
[3,]  6.309653 27.086694 4.101088
```

3、R 矩阵的基本运算

#矩阵加&减

Demo_2=Demo_3=matrix(1:20,nrow=5,ncol=4)

```
> Demo_2-Demo_3
     [,1] [,2] [,3] [,4]
[1,]    0    0    0    0
[2,]    0    0    0    0
[3,]    0    0    0    0
[4,]    0    0    0    0
[5,]    0    0    0    0
> Demo_2+Demo_3
     [,1] [,2] [,3] [,4]
[1,]    2   12   22   32
[2,]    4   14   24   34
[3,]    6   16   26   36
[4,]    8   18   28   38
[5,]   10   20   30   40
```

#矩阵相乘

Demo_4= matrix(1:20,nrow=4,ncol=5)

Demo_5=matrix(1:20,nrow=5,ncol=4)

Demo_4%*%Demo_5

```
> Demo_4%*%Demo_5
     [,1] [,2] [,3] [,4]
[1,]  175  400  625  850
[2,]  190  440  690  940
[3,]  205  480  755 1030
[4,]  220  520  820 1120
```

4、增加行与列

Demo_6<- matrix(,4,2)

Demo_6[c(1,3),] <- matrix(c(1,2,3,4))

Demo_6

```
> Demo_6<- matrix(,4,2)
> Demo_6[c(1,3),] <- matrix(c(1,2,3,4))
> Demo_6
     [,1] [,2]
[1,]    1    3
[2,]   NA   NA
[3,]    2    4
[4,]   NA   NA
```

5、R 矩阵的转置

t(Demo_1)

```
> t(Demo_1)
          r1         r2          r3         r4         r5
A  0.8514152 0.6646625  0.6759027 -1.3863551  2.6685210
B -1.5976567 1.5143233 -0.9762156  0.8732116 -0.2629948
C -0.5942041 0.9693076  1.0168403 -1.4827939 -0.6952818
```

6、矩阵其它运算

diag() #取对角元素

rowSums() #各行汇总值

rowMeans() #各行的平均值

colSums() #各位的汇总值

colMeans() #各列的平均值

(责任编辑：黑阳)

用 R 语言做数据分析——5.数据框

发表于 2013-04-18 14:27 来源：http://www.itongji.cn/

前面几篇文章对 R 语言、R 包以及 R 向量与矩阵等对象做了简单介绍，本篇继续介绍 R 数据框的相关内容。

数据框的创建

通过 data.frame 函数来创建数据框，该函数包含的主要参数为:

data.frame(..., row.names = NULL, check.rows = FALSE, check.names = TRUE,

stringsAsFactors = default.stringsAsFactors())

创建一个数据框

```
Demo_1<-data.frame(name=c('lucy','alice','lei'),
heigh=c(178,156,188),
sex=c('F','F','M'),
score=c(200,210,198))
#包括一个缺少值的数据框
Demo_2<-data.frame(name=c('lucy','alice','lei'),
heigh=c(178,156,188),
sex=c('F','F',NA),
score=c(200,210,198),
stringsAsFactors =F)
```

默认情况下，name 与 sex 都是 factor 类型，如果要将某转化成字符类型，则使用选项：

stringsAsFactors =F

数据框特别点

1、可以包含不同类型的变量；而在矩阵中只能是同一种类型，例如：不能出现字符与数值二种类型；所以可以把数据框理解为各列可为不同类型的向量组合。

```
> data.frame(a=c(1,2,3),b=c('a','b','c'))
  a b
1 1 a
2 2 b
3 3 c
```

2、各列的长度要求一致。如果长度不一，则会报错。例如：

```
> data.frame(a=c(1,2,3),b=c('a','b'))
错误于data.frame(a = c(1, 2, 3), b = c("a", "b")) :
  参数值意味着不同的行数 3, 2
```

3、如果你对数据库比较熟悉，你可以把数据框理解为数据库中的表。

数据框基本操作

1、查看数据框信息

class(Demo_1) #查看类型

class(Demo_1$sex) #查看某列的类型

length(Demo_1) #查看数据框长度

names(Demo_1) #数据框各项名称

#数据框信息总结,对字符类/因子类数据，给出相应的频数统计；给数值型数据给出 5 个主要的指标。

summary(Demo_1)

```
   name        heigh      sex        score
 alice:1   Min.   :156   F:2   Min.   :198.0
 lei  :1   1st Qu.:167   M:1   1st Qu.:199.0
 lucy :1   Median :178         Median :200.0
           Mean   :174         Mean   :202.7
           3rd Qu.:183         3rd Qu.:205.0
           Max.   :188         Max.   :210.0
```

unique(Demo_1) #对数据框进行去重

Demo_1[duplicated(Demo_1),] #取数据框中重复的行

2、取数据框中的某个元素

Demo_1[i,j]指第 i 行第 j 列的数据；

3、取数据框中一列与多列

#取某列，可以通过列标或者名称

Demo_1$sex

Demo_1[,3]

```
> Demo_1$sex
[1] "F" "F" "M"
> Demo_1[,3]
[1] "F" "F" "M"
```

#取多行，例如取 1，2，4 列

Demo_1[,-3]

Demo_1[,c(1,2,4)]

```
> Demo_1[,c(1,2,4)]
   name heigh score
1  lucy   178   200
2 alice   156   210
3   lei   188   198
```

4、取数据框中的一行与多行

#取某行，可以通过行标或者名称,例如：取数据框的第一行

Demo_1[1,]

```
> Demo_1[1,]
  name heigh sex score
1 lucy   178   F   200
```

#取多行，例如：取数据框的第一与第二行

```
> Demo_1[c(1,2),]
   name heigh sex score
1  lucy   178   F   200
2 alice   156   F   210
```

5、判断数据框各行是否完整

complete.cases(Demo_2)

```
> complete.cases(Demo_2)
[1]  TRUE  TRUE FALSE
```

#选择非缺失值的数据

Demo_1[complete.cases(Demo_2),]

```
> Demo_1[complete.cases(Demo_2),]
   name heigh sex score
1  lucy   178   F   200
2 alice   156   F   210
```

6、增加列

#给 Demo_1 增加一列

Demo_1$score2<-c(12,13,23)

Demo_1<-cbind(Demo_1,score2=c(12,13,23))

```
> Demo_1<-cbind(Demo_1,score2=c(12,13,23))
>  Demo_1
   name heigh sex score score2
1  lucy   178   F   200     12
2 alice   156   F   210     13
3   lei   188   M   198     23
```

7、增加行

#合并某二个数据框的二列

Demo_3<-rbind(Demo_1[,c(1,2)], Demo_2[,c(1,2)])

```
> Demo_3<-rbind(Demo_1[,c(1,2)], Demo_2[,c(1,2)])
> Demo_3
   name heigh
1  lucy   178
2 alice   156
3   lei   188
4  lucy   178
5 alice   156
6   lei   188
```

(责任编辑：黑阳)

用 R 语言做数据分析——6.列表

发表于 2013-04-18 14:28 来源：http://www.itongji.cn/

通过 list 函数来创建列表,例如，创建一个包含三个列的列表，请注意与 R 语言中的列表显示有什么不一样：

1、创建列表 demo_1

```
> demo_1<-list(name="nathan", salary=55000, sex="M")
> demo_1
$name
[1] "nathan"

$salary
[1] 55000

$sex
[1] "M"
```

2、创建数据框 demo_1_1

```
> demo_1_1<-data.frame(name="nathan", salary=55000, sex="M")
> demo_1_1
    name salary sex
1 nathan  55000   M
```

·列表对每个列显示都会单独以一行的形式存在；

·数据框则是以表（或者是矩阵）的形式显示；所以根据数据输出的格式可以判断是列表还是数据框；

创建一个示例列表

Demo_2<-list(name=c('lucy','alice','lei'),

heigh=c(178,156,188),

sex=c('F','F','M'),

score=c(200,210,198))

#包括一个缺少值的列表

Demo_3<-list.frame(name=c('lucy','alice','lei'),

heigh=c(178,156,188),

sex=c('F','F',NA),

score=c(200,210,198))

列表特点

列表包含所有列表的特点。列表与列表的最大区别在于，列表的各列的长度是可以不一致的。例如：

```
Demo_3<-list(name=c('lucy','alice','lei'),
             heigh=c(178,156),
             sex= 'F')
```

```
> Demo_3
$name
[1] "lucy"  "alice" "lei"

$heigh
[1] 178 156

$sex
[1] "F"
```

列表基本操作

1.查看列表信息

class(Demo_1) #查看类型

class(Demo_1$sex) #查看某列的类型

length(Demo_1) #查看列表长度 即就是列的个数；

names(Demo_1) #列表各项名称

2.取列表中的某列或者某列元素

Demo_2[2] #取列表的第二列

Demo_2$heigh #取列表的第二列

```
> Demo_2[2]
$heigh
[1] 178 156 188
```

Demo_2[[2]] #取列表的某二列的元素

```
> Demo_2[[2]]
[1] 178 156 188
```

Demo_2[[2]]与 Demo_2[2]二者输出的区别在于，前者是输出向量，后者输出是列表；

我们详细来看下面的内容：

```
> test<-Demo_2[[2]]
> class(test)
[1] "numeric"
> test1<-Demo_2[2]
> class(test1)
[1] "list"
```

3.取向量中的某个元素

Demo_2[[N]][M] 取：N 列中的第 M 个元素；

4.判断列表各行是否完整

complete.cases(Demo_2)

```
> complete.cases(Demo_2)
[1]  TRUE  TRUE FALSE
```

#选择非缺失值的数据

Demo_3[complete.cases(Demo_3)

```
> Demo_3[complete.cases(Demo_3)]
   name heigh score
1  lucy   178   200
2 alice   156   210
3   lei   188   198
```

5.增加列

#给 Demo_1 增加一列

Demo_1$score2<-c(12,13,23)

Demo_1<-cbind(Demo_1,score2=c(12,13,23)

```
> Demo_1<-cbind(Demo_1,score2=c(12,13,23))
> Demo_1
       Demo_1   score2
name   "nathan" 12
salary "55000"  13
sex    "M"      23
```

6.其它操作

#删除一个值

Demo_1$score2<-NULL

```
>  Demo_1<-list(name="nathan",salary="55000",sex="M")
> Demo_1$score2<-c(12,13,23)
> Demo_1$score2<-NULL
> Demo_1
$name
[1] "nathan"

$salary
[1] "55000"

$sex
[1] "M"
```

#把列表转为向量：

unlist()

#移除列表中的元素：

list[[N]] <- NULL

(责任编辑：黑阳)

用 R 语言做数据分析——7.数据输入与输出之 Scan 函数

发表于 2013-04-18 14:28 来源：http://www.itongji.cn/

前面系列文章，介绍了 R 语言中的几种数据格式；其它特殊的数据格式将在后续的文章中介绍，接下来的几篇文章，将讨论如何从外部中读入数据或者将其它格式的数据读入到 R 环境中；

scan() 函数

```
scan(file = "", what = double(0), nmax = -1, n = -1, sep = "",
quote = if(identical(sep, "\n")) "" else "\"", dec = ".",
skip = 0, nlines = 0, na.strings = "NA",
flush = FALSE, fill = FALSE, strip.white = FALSE,
quiet = FALSE, blank.lines.skip = TRUE, multi.line = TRUE,
comment.char = "", allowEscapes = FALSE, encoding = "unknown")
```

一、主要的参数说明如下：

1)what:声明读入为字符类型数据，可能指定读入的精度/类型，例如：what=integer(0)；what=numeric(0)；what=character(0)；

如果 SCAN()读入有字符与数字，用 what=""来进行声明，则会把读入的数字隐式的都转变成字符；

2)SEP：指定各个读入的数据之间的分隔符；默认情况下分隔符：空格、tab；如果不是其它分隔符，例如“：/”通过 SEP 来指定；

3)可以通过 list 指定读入变量的变量名，同时生成的对象为列表，则可以同时读入字符与数字；

4)Skip 从第几行开始读入数据；

5)Nlines 指定最大读入行数；

6)如果通过键盘输入的时候，不希望出现下标提示，则可以使用：quiet=TRUE；

7)encoding =””指定的编码格式，有时候读入的中文可能会出现乱码的时候，可能通过这个参数来指定：Latin-1 或者 UTF-8；

二、SCAN 几点注意：

1) 用于读入纯字符或者数字，没有表头；

2) 如果输入的单一类型的变量，例如均是：数值或者均是字符，用 scan 效率更高。但其不能读入混合类型的数据，也就是在 scan()读入的必须同为字符或者同为数值；

3) 默认情况下用 scan 读入的数据生成向量类型（这也就是为什么读入的数据必须是同为字符或者同为数字）。

三、案例说明

1、手工输入数据

1)从键盘中输入数字

```
> demo_1<-scan("")
1: 1
2: 2
3: 3
4: 4
5: 5
```

2)从键盘中输入字符

```
> demo_2<-scan("",what="")
1: a
2: b
3: c
4: d
5: e
```

2、从外部读入

1)从 D 盘读入 TXT 文件，例如：

```
> demo_3<-scan("d:/demo_3.txt")
Read 8 items
> demo_3
[1] 1 2 3 4 6 7 5 8
```

3、混合字符或者数字的列表格式

如果读入的数据中有字符或者数字，可以通过 list 来进行指定，则生成的对象是列表格式，如下：

```
> demo_4<-scan("d:/demo_4.txt",list(name="",name2=0,name3=0))
Read 3 records
> demo_4
$name
[1] "er" "we" "ww"

$name2
[1] 2 6 5

$name3
[1] 3 7 6
```

4、指定输入字符的编码类型

```
> demo_5
[1] "我要进行测试一下encoding参数："
> demo_5<-scan("d:/demo_5.txt",what="",encoding="Latin-1")
Read 1 item
> demo_5
[1] "我要进行测试一下encoding参数："
```

(责任编辑：黑阳)

用 R 语言做数据分析——8.数据输入与输出之 READ 函数

发表于 2013-04-18 14:29 来源：http://www.itongji.cn/

read.table() 函数

1、用于读入表格（表）类型的数据，同时生成数据框对象。

2、读入的数据要求有规则的分隔符，默认有：空格、TAB、换行符、回车符；其它的分隔符，通过 sep=来进行指定。

```
read.table(file, header = FALSE, sep = "", quote = "\"",
dec = ".", row.names, col.names,
as.is = !stringsAsFactors,
na.strings = "NA", colClasses = NA, nrows = -1,
skip = 0, check.names = TRUE, fill = !blank.lines.skip,
strip.white = FALSE, blank.lines.skip = TRUE,
comment.char = "#",
allowEscapes = FALSE, flush = FALSE,
stringsAsFactors = default.stringsAsFactors(),
fileEncoding = "", encoding = "unknown")
```

file：指定读入的文件，或者文件所在地址；

header：是否读入列名，默认是不读入；

sep：来进行指定分隔符：读入的数据要求有规则的分隔符，可以是：空格、TAB、换行符、回车符；

as.is：读入的字符是否转换成因子，默认所有读入的字符都转换成因子；

colClasses：指定列的数据类型格式

header=TRUE 第一行是否是列的名称，默认是 TRUE

stringsAsFactors= 是否字符转化成因子，默认是 true

row.names=c()指定各行的名称

col.names=C（ ）指定列的名称，如果读入是文件没有头，可以指定

skip=N 从文件第几行开始读入数据

nrows=N 读入的最大行数

na.strings=c()指定什么样的字符表示值缺少

comment.char=‘’指定评论的开始字符，默认是#

dec= 指定小数点数

encoding=指定非 non-ASCII 的编码规则

例如：

demo_3<-read.table('e:/demo_3.txt',header=T)

```
> demo_3<-read.table('e:/demo_3.txt',header=T)
> demo_3
    name age
1   张三  20
2   李四  30
3 王麻子  34
4   李雷  30
5 韩梅梅  33
```

read.fwf（）函数

1、适用用于读入数据相应没有相应的分隔符，但是读入的数据字段长度是固定长度。

2、数据导入 R 后，生成列表对象。

读入固定分隔长度的数据；

read.fwf(file, widths, header = FALSE, sep = "\t",

skip = 0, row.names, col.names, n = -1,

buffersize = 2000)

file:指定读入的文件，或者文件所在地址；

widths：指定分隔的长度，可以等于向量指定不同的分隔；

buffersize：一次最大的读入行数；

n:读入数据的行数，默认为无数；

例如：在这个数据中，前面的 3 个字符与接下来的 3 个数字表示名称、得分，因为二个字段之间没有分隔符号，但其长度是固定的，所以适合用本函数。

ABC123%$12

TEX124@#12

y o14 @@#

read.fwf('e:/demo_1.txt',widths=c(3,3),col.names=c('name','score'));

```
> read.fwf('e:/demo_1.txt',widths=c(3,3),col.names=c('name','score'));
  name score
1  ABC   123
2  TEX   124
3  y o    14
```

w <- readline()函数

1、用于程序的交互，根据输入的条件来判断下一步执行的方向；

2、通过键盘读入一行数据；

例如：根据输入的来判断后续程序的执行流程

```
Demo_2<-function()
{
  input<-readline("DO you think R is hard to learn,Please give your choice:Y or N ")
  if(input=="Y")
cat("Come on; Spent more time.\n")
  else
cat("Good!")
}
Demo_2()
```

```
> Demo_2()
DO you think R is hard to learn,Please give your choice:Y or N Y
come on; Spent more time.
```

Readlines() 函数

1、控制读入的数据行数，非批处理，有点类似于数据库中的指标操作，可对文件中的数据逐行操作。

2、这个对于读入日志类的数据很有用。例如：通过对读入数据的每行来判断是否有需要的数据，有再对数据进行处理；tips:该数据配合 R 中的正则表达式相关函数，对于处理不规则的数据很强大。

例如：

1、与文件 demo_1 建立连接

con<- file("demo_3","r")

2、指定每次执行只读入一行；

RC<-readLines(con,n=1)

3、关闭联接

close(con)

```
> con<- file("e:/demo_3.txt","r")
> RC<-readLines(con,n=1)
> RC
[1] "name age"
> RC<-readLines(con,n=1)
> RC
[1] "张三 20"
> close(con)
```

说明：

1、如果读到文件的最后，则 length(RC)=0；EOF 文件最后返回的空值。

2、N 控制每次读入几行；

3、当读到最后要重新开始的时间：seek(con=c,where=0)，返回当前指标所有的位置

(责任编辑：黑阳)

R 语言入门基础教程

目录

R 语言入门基础教程……33

R 语言入门基础教程-01.常用运算函数……34

R 语言入门基础教程-02.数据框……36

数据框(data frame)……36

R 语言入门基础教程-03.因子和有序因子……38

因子(factor)和有序因子(ordered factor)……38

R 语言入门基础教程-04.数组和列表……40

数组(array)……40

列表(list)……40

R 语言入门基础教程-05.矩阵……42

矩阵(matrix)……42

R 语言入门基础教程-06.向量……44

向量（vector）……44

R 语言入门基础教程-01.常用运算函数

发表于 2012-12-14 01:40 来源：http://blog.sina.com.cn/s/blog_59f8748e01011iw6.html

R 语言入门基础教程：常用运算函数。对一般数据进行运算的常用函数：

1、round() #四舍五入

例：x <- c(3.1416, 15.377, 269.7)

round(x, 0) #保留整数位

round(x, 2) #保留两位小数

round(x, -1) #保留到十位

2、signif() #取有效数字(跟学过的有效数字不是一个意思)

例：略

3、trunc() #取整

floor() #向下取整

ceiling() #向上取整

例：xx <- c(3.60, 12.47, -3.60, -12.47)

trunc(xx)

floor(xx)

ceiling(xx)

4、logb(a, b) #以 b 为底的对数，省略 b 表示自然对数

log() #自然对数

log10() #以 10 为底的常用对数

例：logb(8, 2)

log(8); logb(8)

log10(100); logb(100, 10)

5、sqrt() #平方根

exp() #指数

6、sin() #正弦

cos() #余弦

tan() #正切

asin() #反正弦

acos() #反余弦

atan() #反正切

sinh() #双曲正弦

tanh() #双曲正切

7、nchar() #字符长度

例：xx <- 'China is a great country'

nchar(xx)

8、substring() #取子字符串

例：substring(xx, 1, 5)

9、paste() #连接字符

语法是：paste(..., sep = " ", collapse = NULL)

例 1：x <- 'I'; y <- 'am'; z <- 'a'; d <- 'student'

paste(x, y, z, d)

例 2：paste(c('x', 'y'), 1:4, sep = '')

例 3：paste('x', 1:4, sep = '', collapse = '+')

R 语言入门基础教程-02.数据框

发表于 2012-12-14 01:23 来源：http://blog.sina.com.cn/s/blog_59f8748e01011ivz.html

数据框(data frame)

数据框是一种矩阵形式的数据，但数据框中各列可以是不同类型的数据。数据框每列是一个变量，每行是一个观测。数据框可以看成是矩阵的推广，也可看作一种特殊的列表对象，很多高级统计函数都会用到数据框。

数据框用函数 data.frame()生成，语法是：data.frame(data1, data2, ...)

1、生成一个数据框

例 1：name <- c('Mr A', 'Mr B', 'Mr C'); group <- rep(1, 3); score <- c(69, 71, 92)

dd <- data.frame(name, group, score)

2、合并数据框

例 1：name <- c('Ms C', 'Ms D'); group <- c(2, 2); score <- c(93, 99)

dd1 <- data.frame(name, score, group) #注意这里排列顺序与 dd 中不同

dd2 <- rbind(dd, dd1) #行合并结果与 dd 排列顺序一致，说明其中有一个匹配过程。

dd3 <- rbind(dd1, dd)

例 2：age <- c(14, 15, 14, 16, 13)

dd4 <- cbind(dd2, age) #列合并

dd4[2, 3]; dd4$score[2]

3、"连接"函数

attach()和 detach()函数是应用数据框时很有用的工具。attach()函数将数据框连接入当前工作空间，detach()取消连接。

如果不用 attach()，需要用$提取数据框内某一列数据。

1、attach()和 detach()函数的应用

例 1：girl1 <- read.table('d:/girl1.txt', head = T) #读取数据

```
WT2

attach(girl1) #连接入当期工作空间

WT2 <- 12:13

mode(WT2) #结果时数值型

rm(WT2)

detach(girl1) #取消连接

WT2

girl1$WT2
```

R 语言入门基础教程-03.因子和有序因子

发表于 2012-12-14 01:23 来源：http://blog.sina.com.cn/s/blog_59f8748e01011in6.html

因子(factor)和有序因子(ordered factor)

因子用来存储类别变量(categorical variables)和有序变量，这类变量不能用来计算而只能用来分类或者计数。因子表示分类变量，有序因子表示有序变量。生成因子数据对象的函数是 factor(),语法是 factor(data, levels, labels, ...)，其中 data 是数据，levels 是因子水平向量，labels 是因子的标签向量。

1、创建一个因子。

例 1：colour <- c('G', 'G', 'R', 'Y', 'G', 'Y', 'Y', 'R', 'Y')

col <- factor(colour)

col1 <- factor(colour, levels = c('G', 'R', 'Y'), labels = c('Green', 'Red', 'Yellow')) #labels 的内容替换 colour 相应位置对应 levels 的内容

col2 <- factor(colour, levels = c('G', 'R', 'Y'), labels = c('1', '2', '3'))

col_vec <- as.vector(col2) #转换成字符向量

col_num <- as.numeric(col2) #转换成数字向量

col3 <- factor(colour, levels = c('G', 'R'))

2、创建一个有序因子。

例 1：score <- c('A', 'B', 'A', 'C', 'B')

score1 <- ordered(score, levels = c('C', 'B', 'A')); score1

3、用 cut()函数将一般的数据转换成因子或有序因子。

例 1：exam <- c(98, 97, 52, 88, 85, 75, 97, 92, 77, 74, 70, 63, 97, 71, 98, 65, 79, 74, 58, 59, 60, 63, 87, 82, 95, 75, 79, 96, 50, 88)

exam1 <- cut(exam, breaks = 3) #切分成 3 组

exam2 <- cut(exam, breaks = c(0, 59, 69, 79, 89, 100)) #切分成自己设置的组

attr(exam1, 'levels'); attr(exam2, 'levels'); attr(exam2, 'class')

ordered(exam2, labels = c('bad', 'ok', 'average', 'good', 'excellent')) #一个有序因子

R 语言入门基础教程-04.数组和列表

发表于 2012-12-14 01:21 来源：http://blog.sina.com.cn/s/blog_59f8748e01011imt.html

数组(array)

一维数据是向量，二维数据是矩阵，数组是向量和矩阵的直接推广，是由三维或三维以上的数据构成的。数组函数是 array(),语法是：array(dadta, dim)，其中 data 必须是同一类型的数据，dim 是各维的长度组成的向量。

1、产生一个三维和四维数组。

例 1：xx <- array(1:24, c(3, 4, 2)) #一个三维数组

例 2：yy <- array(1:36, c(2, 3, 3, 2)) #一个四维数组

2、dim()函数可将向量转化成数组或矩阵。

例 1：xx <- 1:24; dim(xx) <- c(3, 4, 2); xx #效果同 array(1:24, c(3, 4, 2))

例 2：zz <- 1:10; dim(zz) <- c(2, 5); zz #效果同 matrix(1:10, 2, 5)

列表(list)

向量、矩阵和数组的元素必须是同一类型的数据。一个数据对象需要包含不同的数据类型，它可以采用列表这种形式。

创建列表可用 list()函数，语法是：list(name1 = component1, name2 = component2, ...)。

1、创建一个列表

例 1： xx <- rep(1:2, 3:4)

yy <- c('Mr A', 'Mr B', 'Mr C', 'Mr D', 'Mr E', 'Mr D', 'Mr F')

zz <- 'discussion group'

name.list <- list(group = xx, name = yy, decription = zz) #创建了一个名为"name.list"的列表

name.list$n[name.list$g == 2]

length(name.list)

mode(name.list)

names(name.list)

R 语言入门基础教程-05.矩阵

发表于 2012-12-14 01:18 来源：http://blog.sina.com.cn/s/blog_59f8748e01011hwx.html

矩阵(matrix)

矩阵生成函数 matrix():matrix(data, nrow = , ncol = , byrow = F)，其中，数据 data 是必须的，其他都是选择参数，可以不选。byrow = F 默认为按列来排列数据，如果想要按行排列，令 byrow = T。

1、对角矩阵和单位阵。

例 1：x <- 1:6; diag(x) #对角矩阵

例 2：y <- rep(1, 5); diag(y) #单位阵

2、矩阵下标

例 1：xx <- matrix(1:20, 4, 5)

xx[2, 2]; xx[2, 3:5]; xx[3:4, 3:4]

xx[2,]; xx[, 2]

3、代数意义下的矩阵乘法"%*%"

例 1：yy <- matrix(1:6, 3, 2); zz <- matrix(1:6, 2, 3)

yy %*% zz; zz %*% yy

4、矩阵行和列的维数

例 1：xx <- matrix(1:20, 4, 5)

dim(xx) #行和列的维数

nrow(xx); ncol(xx) #行数和列数

5、矩阵的主要运算函数

例 1：x <- 1:6; y <- as.matrix(x) #转换成矩阵

is.matrix(x); is.matrix(y) #判断是否矩阵

例 2：

diag() #方阵对角线元素或者生成对角矩阵　　apply() #对矩阵应用函数

eigen() #求特征值和特征向量

qr() #QR 分解

solve() #求逆矩阵

det() #求行列式

chol() #Choleski 分解

dim() #给出行列数

svd() #奇异值分解

t() #矩阵转置

6、矩阵合并

例 1：aa <- matrix(1:6, 3, 2); bb <- matrix(7:12, 3, 2)

cbind(aa, bb) #按列合并

rbind(aa, bb) #按行合并

7、矩阵 apply()运算函数

语法是 apply(data, dim, function),dim 取 1 表示对行运用函数，取 2 表示对列运用函数。

例 1：xx <- matrix(1:20, 4, 5)

colMeans(xx) #列均值

colSums(xx) #列和

其余大部分都要用到 apply()函数

例 2：xx <- matrix(1:20, 4, 5)

apply(xx, 2, mean) #列均值，等同于 colMeans(xx)

apply(xx, 2, sum) #列和，等同于 colMeans(xx)，所以矩阵行和列的运算推荐用 apply()。

apply(xx, 1, var) #行方差

apply(xx, 2, max) #每列最大值

apply(xx, 2, rev) #每列的数反排列

R 语言入门基础教程-06.向量

发表于 2012-12-14 01:18 来源：http://blog.sina.com.cn/s/blog_59f8748e01011ht7.html

向量（vector）

1. seq():产生有规律的数列

间距省略时默认值为 1。 例 1：seq(10, 20, 0.5) 例 2：seq(0, by = 0.03, length = 15)

2. rep()：重复产生有规律的数列

重复第一个变量若干次。例 1：rep(1:3, 1:3) 例 2：rep(1:3, rep(2, 3)) 例 3：rep(1:3, length = 10)

3. 向量运算

一般是对应元素之间的运算，所以两个或多个向量运算时，要求它们包含的元素个素相同（或一个是另一个的整数倍）。

例 1：a <- 1:3; b <- 4:6; a * b; b^a

例 2：a <- 1:3; b <- 4:9; a * b; b^a

4. 获取向量某一个或多个子集

向量前的负号"-"表示去除相应内容。

例 1：x <- c(3, 4, 5, 2, 6); x[1:2]; x[-(1:2)]

例 2：x <- c(3, 4, 5, 2, 6); x[c(1, 2, 4, 1)]; x[-c(1, 2, 4, 1)]

例 3：xx <- seq(1, by = 3, length = 10); xx[xx > 13]

例 4：x <- 1:20; y <- -9:11; x[y > (1)] #注意最后一个是"NA"

5. 主要向量运算函数

例 1：xx <- c(2, 6, 10, 8, 4)

sum(xx) #和

max(xx) #最大值

min(xx) #最小值

range(xx) #取值范围

mean(xx) #平均值

var(xx) #方差

sort(xx) #从小到大排序

rev(xx) #反排列， 所以从大到小排序应该是 rev(sort(xx))

rank(xx) #单元值大小顺序

prod(xx) #乘积，所以阶乘是 prod(1:n)

例 2：x <- seq(1, 15, 2)

append(x, 20:30, after = 5) #插入数据

append(x, 20:30) #参数 after 缺省默认从向量的最后插入值

replace(x, c(2, 4, 6), -1) #替换函数

例 3：state.name

match(c('Ohio', 'Wyoming'), state.name) #完全匹配函数

pmatch(c('Oh', 'Wy'), state.name) #部分匹配函数

state.name[pmatch(c('Oh', 'Wy'), state.name)]

例 4：yy <- -9:10

all(yy > 0) #判断所有

all(yy > -10)

any(yy == 0) #判断部分

any(yy > 0)

any(yy < -10)

R 语言学习由浅入深路线图

发表于 2012-12-19 01:19 来源：格物堂 责任编辑：中国统计网

目录

R 语言学习由浅入深路线图……46

1.初级入门……47

2.高级入门……47

3.绘图与可视化……47

4.计量经济学……48

5.时间序列分析……48

6.金融……49

7.数据挖掘……49

8.附注……49

现在对 R 感兴趣的人越来越多，很多人都想快速的掌握 R 语言，然而，由于目前大部分高校都没有开设 R 语言课程，这就导致很多人不知道如何着手学习 R 语言。

对于初学 R 语言的人，最常见的方式是：遇到不会的地方，就跑到论坛上吼一嗓子，然后欣然 or 悲伤的离去，一直到遇到下一个问题再回来。当然，这不是最好的学习方式，最好的方式是——看书。目前，市面上介绍 R 语言的书籍很多，中文英文都有。那么，众多书籍中，一个生手应该从哪一本着手呢？入门之后如何才能把自己练就成某个方面的高手呢？相信这是很多人心中的疑问。有这种疑问的人有福了，因为笔者将根据自己的经历总结一下 R 语言书籍的学习路线图以使 Ruser 少走些弯路。

本文分为 6 个部分，分别介绍初级入门，高级入门，绘图与可视化，计量经济学，时间序列分析，

金融等。

1.初级入门

《An Introduction to R》，这是官方的入门小册子。其有中文版，由丁国徽翻译，译名为《R 导论》。《R4Beginners》，这本小册子有中文版应该叫《R 入门》。除此之外，还可以去读刘思喆的《153 分钟学会 R》。这本书收集了 R 初学者提问频率最高的 153 个问题。为什么叫 153 分钟呢？因为最初作者写了 153 个问题，阅读一个问题花费 1 分钟时间，全局下来也就是 153 分钟了。有了这些基础之后，要去读一些经典书籍比较全面的入门书籍，比如《统计建模与 R 软件》，国外还有《R Cookbook》和《R in action》，本人没有看过，因此不便评论。

最后推荐，《R in a Nutshell》。对，“果壳里面的 R”！当然，是开玩笑的，in a Nutshell 是俚语，意思大致是“简单的说”。目前，我们正在翻译这本书的中文版，大概明年三月份交稿！这本书很不错，大家可以从现在开始期待，并广而告知一下！

2.高级入门

读了上述书籍之后，你就可以去高级入门阶段了。这时候要读的书有两本很经典的。《Statistics with R》和《The R book》。之所以说这两本书高级，是因为这两本书已经不再限于 R 基础了，而是结合了数据分析的各种常见方法来写就的，比较系统的介绍了 R 在线性回归、方差分析、多元统计、R 绘图、时间序列分析、数据挖掘等各方面的内容，看完之后你会发现，哇，原来 R 能做的事情这么多，而且做起来是那么简洁。读到这里已经差不多了，剩下的估计就是你要专门攻读的某个方面内容了。下面大致说一说。

3.绘图与可视化

亚里斯多德说，“较其他感觉而言，人类更喜欢观看”。因此，绘图和可视化得到很多人的关注和重视。那么，如何学习 R 画图和数据可视化呢？再简单些，如何画直方图？如何往直方图上添加密度曲线呢？我想读完下面这几本书你就大致会明白了。

首先，画图入门可以读《R Graphics》，个人认为这本是比较经典的，全面介绍了 R 中绘图系统。该

书对应的有一个网站，google 之就可以了。更深入的可以读《Lattice：Multivariate Data Visualization with R》。上面这些都是比较普通的。当然，有比较文艺和优雅的——ggplot2 系统，看《ggplot2 :Elegant Graphics for Data Analysis》。还有数据挖掘方面的书：《Data Mining with Rattle and R》，主要是用 Rattle 软件，个人比较喜欢 Rattle!当然，Rattle 不是最好的，Rweka 也很棒！再有就是交互图形的书了，著名的交互系统是 ggobi，这个我已经喜欢两年多了，关于 ggobi 的书有《Interactive and Dynamic Graphics for Data Analysis With R and GGobi》，不过，也只是适宜入门，更多更全面的还是去 ggobi 的主页吧，上面有各种资料以及包的更新信息！

特别推荐一下，中文版绘图书籍有谢益辉的《现代统计图形》。

4.计量经济学

关于计量经济学，首先推荐一本很薄的小册子:《Econometrics In R》，做入门用。然后，是《Applied Econometrics with R》，该书对应的 R 包是 AER，可以安装之后配合使用，效果甚佳。计量经济学中很大一部分是关于时间序列分析的，这一块内容在下面的地方说。

5.时间序列分析

时间序列书籍的书籍分两类，一种是比较普适的书籍，典型的代表是：《Time Series Analysis and Its Applications ：with R examples》。该书介绍了各种时间序列分析的经典方法及实现各种经典方法的 R 代码，该书有中文版。如果不想买的话，建议去作者主页直接下载，英文版其实读起来很简单。时间序列分析中有一大块儿是关于金融时间序列分析的。这方面比较流行的书有两本《Analysis of financial time series》，这本书的最初是用的 S-plus 代码，不过新版已经以 R 代码为主了。这本书适合有时间序列分析基础和金融基础的人来看，因为书中关于时间序列分析的理论以及各种金融知识讲解的不是特别清楚，将极值理论计算 VaR 的部分就比较难看懂。另外一个比较有意思的是 Rmetrics 推出的《TimeSeriesFAQ》，这本书是金融时间序列入门的东西，讲的很基础，但是很难懂。对应的中文版有《金融时间序列分析常见问题集》，当然，目前还没有发出来。经济领域的时间序列有一种特殊的情况叫协整，很多人很关注这方面的理论，关心这个的可以看《Analysis of Integrated and Cointegrated Time Series with R》。最后，比较

高级的一本书是关于小波分析的，看《Wavelet Methods in Statistics with R》。附加一点，关于时间序列聚类的书籍目前比较少见，是一个处女地，有志之士可以开垦之！

6.金融

金融的领域很广泛，如果是大金融的话，保险也要被纳入此间。用 R 做金融更多地需要掌握的是金融知识，只会数据分析技术意义寥寥。我觉得这些书对于懂金融、不同数据分析技术的人比较有用，只懂数据分析技术而不动金融知识的人看起来肯定如雾里看花，甚至有人会觉得金融分析比较低级。这方面比较经典的书籍有：《Advanced Topics in Analysis of Economic and Financial Data Using R》以及《Modelling Financial Time Series With S-plus》。金融产品定价之类的常常要用到随机微分方程，有一本叫《Simulation Inference Stochastic Differential Equations：with R examples》的书是关于这方面的内容的，有实例，内容还算详实!此外，是风险度量与管理类。比较经典的有《Simulation Techniques in Financial Risk Management》、《Modern Actuarial Risk Theory Using R》和《Quantitative Risk Management：Concepts, Techniques and Tools》。投资组合分析类和期权定价类可以分别看《Portfolio Optimization with R》和《Option Pricing and Estimation of Financial Models with R》。

7.数据挖掘

这方面的书不多，只有《Data Mining with R:learing with case studies》。不过，R 中数据挖掘方面的包已经足够多了，参考包中的帮助文档就足够了。

8.附注

出于版权等事宜的考虑，我无法告知你说在“新浪爱问”等地方可以直接免费下载到上面提到的这些书，但是，我想你可以发挥自己的聪明才智去体悟！

用 R 语言求置信区间

发表于 2012-12-14 01:11 来源：新浪博客 via：炼数成金(责任编辑：中国统计网)

用 R 语言求置信区间是很方便的，而且很灵活，至少我觉得比 spss 好多了。

如果你要求的只是 95%的置信度的话，那么用一个很简单的命令就可以实现了

首先，输入 da=c(你的数据，用英文逗号分割)，然后 t.test(da)，运行就能得到结果了。

我的数据是 newbomb <- c(28,26,33,24,34,-44,27,16,40,-2,29,22,24,21,25,30,23,29,31,19)

t.test(newbomb)得到的结果如下

```
        One Sample t-test

data:  newbomb
t = 5.5181, df = 19, p-value = 2.533e-05
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 13.50014 29.99986
sample estimates:
mean of x
    21.75
```

如果要求任意置信度下的置信区间的话，就需要自己编一个函数了。当然，有两点要记住的，置信区间的计算在知道方差和不知道方差的情况下，计算公式是不一样的。下面做一个两种情况下都可以用的函数。

```
confint<-function(x,sigma=-1,alpha=0.05)
  {
      n<-length(x)
      xb<-mean(x)
      if(sigma>=0)
          {
              tmp<-sigma/sqrt(n)*qnorm(1-alpha/2);df<-n
```

```
        }
    else{
        tmp<-sd(x)/sqrt(n)*qt(1-alpha/2,n-1);df<- n-1
        }
    data.frame(mean=xb,df=df,a=xb-tmp,b=xb+tmp)
}
```

这个函数的使用：如果不知道方差，则 confint（x,alpha）；知道方差，则 confint（x,sigma,alpha）,这样就能计算出结果了。