R语言记录3:数据的导入及复杂操作

Mar 9th, 2014

“文章原创，转载请注明出处”

上一节介绍了R语言中常用的数据类型，并且介绍了R语言中对数据的一些简单操作，如创建查看向量或者矩阵等等。今天就介绍一下如何从文件中导入数据，以及对数据进行一些较为复杂的操作。

一. 读取数据

R语言读取数据有很多方式，如scan函数，read.table函数等等。我这里还想介绍另外两个函数，read.csv和read.delim函数，因为处理数据，大部分时候遇到的文件是csv文件或者txt文件，而这两个函数可以很好的读取此类型的文件。

1. scan函数

首先简单说说scan函数，这个函数在默认的情况下，希望所有输入的数据都是数值型的。scan函数读取的数据必须是同一模式的，如果没有指定参数，scan将从R控制台读取数据，直到遇到一个空白行停止。

loadDatalink

```
1 > x <- scan()
2 1: 1 2 3
3 4: 2 3 4
4 7:
5 Read 6 items
6 > x
7 [1] 1 2 3 2 3 4
```

也可以通过设定参数，设定读取的数据类型：

loadDatalink

```
1 > y <- scan(what="") # 读取字符型数据
2 1: 1 2 3
3 4: 2 3 4
4 7:
5 Read 6 items
6 > y
7 [1] "1" "2" "3" "2" "3" "4"
```

也可以直接读取一个同一模式的数据文件：

loadDatalink

```
1 > z <- scan("example.txt")
2 Read 9 items
3 > z
4 [1] 1 2 3 4 5 6 7 8 9
```

其中example.txt的内容如下：

```
1    2    3
4    5    6
7    8    9
```

注意最后留一个空白行，以便R读取数据时可以找到终止位置(否则会出现warning)。

2. read.table函数

顾名思义，read.table函数即是以数据框的格式在R中读取数据。这里就涉及到一个新的概念，数据框。前一节我们提到过，可以使用列表和数据框去存储具有不同模式的数据。数据框可以存储具有相同长度的变量，并且在统计中，其每一行就是样本的一个观测值，有多少行就有多少次观测。我们可以使用data.frame函数来生成一个数据框：

loadDatalink

```
1 > name <- c("Li", "Wang", "Sun")
2 > age <- c(21, 22, 23)
3 > Datafr <- data.frame(name = name, age = age)
4 > Datafr
5   name age
6 1   Li  21
7 2 Wang  22
8 3  Sun  23
```

此外，数据框还有一个优点：可以在不影响原始数据的情况下，改变数据。比如我想知道，数据里面每个人的年龄比18大多少：

loadDatalink

```
1 > Datafr2 <- data.frame(name = name, age = age, age.u = age-18)
2 > Datafr2
3   name age age.u
4 1   Li  21     3
5 2 Wang  22     4
6 3  Sun  23     5
```

那么如何使用read.table函数呢，看看如下示例：

loadDatalink

```
1 > example <- read.table("example.txt", header=FALSE)
2 > example
3   V1 V2 V3
4 1  1  2  3
5 2  4  5  6
6 3  7  8  9
```

其中header参数指定数据文件的第一行是否为变量名。当没有变量名时，可以使用参数col.names来指定：

loadDatalink

```
1 > example <- read.table("example.txt", header=FALSE, col.names=c("x1", "x2", "x3"))
2 > example
3   x1 x2 x3
4 1  1  2  3
5 2  4  5  6
6 3  7  8  9
```

当然read.table函数读取数据时，可以使用sep=参数指定分隔符。这些可以使用?read.table查看。

3. read.csv和read.delim函数

使用方法与read.table函数类似，可以直接读取csv文件和txt文件。

loadDatalink

```
1 dataCsv <- read.csv("example.csv", header=FALSE)
2 dataTxt <- read.delim("example.txt", header=FALSE)
```

二. 数据操作

上一节介绍了一些简单的数据操作，这次讲一讲较为复杂的数据操作。

1. 首先从-5到5中随机取出不重复的10个数据(sample函数)：

dataManipulationlink

```
1 > x <- sample(-5:5, 10)
2 > y <- sample(-5:5, 10)
3 > x
4  [1] -2  3 -3  0 -5  5  1  4 -1  2
5 > y
6  [1] -1  3 -3  1 -4  5  0  2 -5 -2
```

2. 判断x中的数据是否大于0(all和any函数)：

dataManipulationlink

```
1 > x>0
2 [1] FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE
3 > all(x>0)
4 [1] FALSE
5 > any(x>0)
6 [1] TRUE
```

3. 试试更多的逻辑运算(&, |, !)：

dataManipulationlink

```
1 > x > 0 & y > 0
2       [1] FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE
3 > x > 0 | y > 0
4  [1] FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE
5 > !(x>0)
6       [1]  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE
```

4. 找出x与y相等的元素(which函数):

dataManipulationlink

```
1 > x == y
2  [1] FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
3 > which(x == y) # 取出满足条件的位置
4  [1] 2 3 6
```

5. 从-5到5中随机取出允许重复的10个数据(sample函数)：

dataManipulationlink

```
1 > z <- sample(-5:5, 10, replace=TRUE)
2 > z
3  [1]  4 -4 -1 -2  4  5 -3 -3  1  2
```

6. 取出z中不同的元素(unique函数)：

dataManipulationlink

```
1 > z.unique <- unique(z)
2 > z.unique
3 [1]  4 -4 -1 -2  5 -3  1  2
```

7. 生成存在缺失数据的一列数据(NA)：

dataManipulationlink

```
1 > x.na <- c(1:5, NA, 2, NA, 3:5)
2 > x.na
3  [1]  1  2  3  4  5 NA  2 NA  3  4  5
```

8. 对x求和(is.na函数):

dataManipulationlink

```
1 > sum(x.na) # 带着缺失数据计算
2 [1] NA
3 > sum(is.na(x.na)) # 统计缺失数据的个数
4 [1] 2
5 > sum(x.na, na.rm=TRUE) # 剔除缺失数据后求和
6 [1] 29
```

9. 向变量x中添加数据(append函数)：

dataManipulationlink

```
> x <- append(x, sample(-5:5, 2))
> x
[1] -2  3 -3  0 -5  5  1  4 -1  2 -1 -5
```

10. 更改x的数据类型(as.character,as.factor,as.numeric函数)：

dataManipulationlink

```
> x <- as.character(x)
> x
 [1] "2"  "8"  "3"  "5"  "4"  "10" "6"  "9"  "1"  "7"  "1"  "4"
> x <- as.factor(x)
> x
 [1] 2  8  3  5  4  10 6  9  1  7  1  4
Levels: 1 10 2 3 4 5 6 7 8 9
> x <- as.numeric(as.character(x)) # 如果一组因子变量是用数字表示，一般先将数字因子转换成字符后再转换成数值。
> x
 [1]  2  8  3  5  4 10  6  9  1  7  1  4
```

11. 数据类型更改(unlist,as.data.frame)：

dataManipulationlink

```
> myList <- list()
> myList$x1 <- c(1, 2, 3)
> myList$x2 <- c(2, 4, 6)
> myList
$x1
[1] 1 2 3

$x2
[1] 2 4 6

# unlist
> uList <- unlist(myList)
> uList
x11 x12 x13 x21 x22 x23
  1   2   3   2   4   6

# matrix
> mList <- matrix(uList, 2, 3, byrow=TRUE)
> mList
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    2    4    6

# dataframe
> dfList <- as.data.frame(mList)
```

```
> dfList
  V1 V2 V3
1  1  2  3
2  2  4  6
> colnames(dfList) <- c("x1", "x2", "x3")
> dfList
  x1 x2 x3
1  1  2  3
2  2  4  6
> rownames(dfList) <- c("o1", "o2")
> dfList
   x1 x2 x3
o1  1  2  3
o2  2  4  6
```

12. 按条件整理计算数据(table,tapply,by,aggregate函数)：

dataManipulationlink

```
> x <- list()
> x$a <- sample(-10:10, 20, replace=TRUE)
> x$b <- as.factor(sample(1:4, 20, replace=TRUE))
> x$c <- as.factor(sample(-3:-1, 20, replace=TRUE))
> x

# 统计因子出现的频数
> table(x$b)

 1  2  3  4
 5  3  1 11
> table(x$c)

-3 -2 -1
 9  7  4
> table(x$b, x$c)

   -3 -2 -1
 1  1  3  1
 2  1  2  0
 3  0  1  0
 4  7  1  3

# 使用tapply，by以及aggregate函数根据不同因子对数据进行计算
> tapply(x$a, x$b, sum)
 1  2  3  4
 8 11  2 -5
> tapply(x$a, x$c, sum)
-3 -2 -1
-7 16  7
> tapply(x$a, list(x$b, x$c), sum) # 可以看到，由于b=2且c=-1的数据不存在，所有计算结果为NA(其它类似)
  -3 -2 -1
1 -3  6  5
```

```
34 2  5  6 NA
35 3 NA  2 NA
36 4 -9  2  2
37
38 > by(x$a, list(x$b, x$c), sum) # 输出结果太长就不给出了，可以观察一下与tapply输出的区别
39
40 > aggregate(a ~ b+c, x, sum) # 第一个参数为要计算的量，中间的参数为条件，第三个参数为数据集名称，最后一个为函数
41   b  c  a
42 1 1 -3 -3
43 2 2 -3  5
44 3 4 -3 -9
45 4 1 -2  6
46 5 2 -2  6
47 6 3 -2  2
48 7 4 -2  2
49 8 1 -1  5
50 9 4 -1  2
```

三. 小节

这次我们介绍了R语言如何从外部读取数据，以及对得到的数据如何进行一些复杂点的操作。当然这还不是全部，比如我这里还没有讲到如何将数据集按行按列去进行合并(rbind,cbind函数)等等，这些都可以通过help文档得到。我觉得学习一门语言一定要学会使用help，还有善用Google很重要。

Posted by Jacky Code Mar 9th, 2014 R, RSeries

- « R语言记录2 ：R语言中的数据
- Blog Archives
- R语言记录4：自定义函数的深入 »

- 关于我

  概率统计专业的在读研究生

  喜欢看书，思考，然后写点什么

  能折腾，崇尚效率，喜爱精美

  喜欢Apple，离不了Google

  新浪微博: Jacky130

  邮箱: jackycodemsc@gmail.com

- 文章分类
-
- AssociationAnalysis (2)