

《数据分析与处理技术》附录4
南京审计大学2015级物流管理专业

作业中高频问题汇总

版本：2017.10.21

- 作业一问题汇总

区分变量的赋值与调用(访问)

- 区分清楚赋值与调用(访问)的作用

赋值: 赋值过程改变变量内记录情况, 直接修改内存中的变量区域

赋值符号包括:

<- 左赋值

-> 右赋值

= 等同于左赋值, 但是: 建议赋值时用上边两种, 而函数内参数设置用=

调用: 等同于拷贝一份变量记录的副本, 并不改变变量实际记录情况

调用的基本方式是直接写出变量名和元素定位

如

```
> a<-c(1,3,6,8)  创建一个向量, 通过赋值符号存入一个名为a的变量中
> a              直接调用整个a变量
[1] 1 3 6 8
> a[3]           调用a变量的第三个元素
[1] 6
> a[3]<-7         通过赋值符号修改变量a中第三个元素的数值
> a[3]           再次调用a[3], 发现记录的数值变为7而非原来的6
[1] 7
> a              调用a变量, 清楚看到原本a中记录了一个向量1, 3, 6, 8, 现在第三个元素数值被改变为7了
[1] 1 3 7 8
```

数据生成：部分等效但多余的操作

- 冒号生成连续数列是目前所学范围内唯一不加组合函数c()的方式

```
> a<-c(1,2,3,4,5,6,7,8,9,10)
> b<-1:10
> a
[1] 1 2 3 4 5 6 7 8 9 10
> b
[1] 1 2 3 4 5 6 7 8 9 10
```

- 简单矩阵的生成中cbind可以达到等同于matrix()的效果. 但矩阵生成不建议使用cbind，因为你无法生成高维矩阵，矩阵生成使用函数matrix(……)，而cbind通常用来添加行

- 创建了一个矩阵，但是没有赋值给某变量，这个矩阵去哪了

答案：内存中有一块临时区域，每次创建出来的数据都放在那里，如果赋值给变量a，便在创造出来时就被a领走了；如果没有赋值，那么在下次任何命令执行时就被覆盖掉了，因此没有赋值的数据只能存在于一瞬间用于显示

```
> b<-1:10
> matrix(b,nrow=2,byrow=TRUE)
      [,1] [,2] [,3] [,4] [,5]
[1,]    1    2    3    4    5
[2,]    6    7    8    9   10
```

矩阵的运算

- 大多数同学都没注意到附录3的存在，附录3中列有具体的运算符号含义和用法。矩阵运算不同于单个数字的运算，因此矩阵的运算符号也是比较复杂。例如：

假如 $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, $B = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ ，加法 $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}$ ，R中直接写A+B。注意：这个过程是点对点加法

乘法 $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 3 \\ 7 & 7 \end{bmatrix}$ ，R中写入 $\begin{matrix} & & > \text{a*b} \\ & & & \begin{matrix} [,1] & [,2] \\ [1,] & 1 & 2 \\ [2,] & 3 & 4 \end{matrix} \end{matrix}$ ，有没有发现其实是点对点相乘，跟加法一样的规则，

而不是我们理解的矩阵左乘关系。关键在于符号用错了，正确的写法是 $\begin{matrix} & & > \text{a\%\%b} \\ & & & \begin{matrix} [,1] & [,2] \\ [1,] & 3 & 3 \\ [2,] & 7 & 7 \end{matrix} \end{matrix}$

分析

- plot用法为什么那么多等效方式。

plot(……)是基础画图函数，你送给它一堆参数，它还给你一副美图。

基本格式是plot(x,y)，即告诉plot函数你要画的横纵坐标数据（分别是一列数据）。作业中cars是一个数据集，本身包含两个变量cars\$speed和cars\$dist，所以一个cars就相当于两个变量的数据，plot(cars)就可以了，只是你要注意谁被画在横坐标上谁被画在纵坐标上了。

- 注意：数据做出来是为了分析，切不可当数学题那样算出结果就结束，算出结果后应当再根据结果给出结论（观点），这才是数据分析的标准范儿。
- 为什么cars不需要创建就存在？

cars是R自带的案例数据，其实是存在基础package之一的datasets中，当你打开R的同时就被自动加载进来了。如果想在变量列表里看到它，请打data(cars) 加载数据命令，而非head(……)。

- with是什么命令？

with()这个函数能实现的功能跟attach比较类似，用于简化数据集调用的方式，实验2的PPT中有一个attach() ……detach() 即加载……卸除的用法。

也就是说我们正常调用数据集cars下的变量speed必须写出它的 数据集名\$变量个体名，用上这两种方式后只需要写变量个体名就可以了。

简单程序根本不需要这东西，复杂程序里会非常有用。

- 圆括号 () 用于跟在函数后边表示参数输入，单独的()是什么意思？

`a <- (5+2)*3` 用于优先级划分

- 变量 `a <- c(1,3,5,6,8,9)`, 想同时调用第3和第5个元素怎么做？

`a[c(3,5)]`

`c(……)`的作用就是把多个单位组合在一起

- 作业二问题汇总

- install.package过程不需要写在作业中，实际操作时一次安装以后再用时只需要library()加载就可以了。
- Rglpk包有一个关联包，也就是它正常运行需要内存中同时已经加载了一个叫slam的包，二者加载先后顺序可以调换，但推荐先加载前置包slam。
- 当你生成一个数据集时，例如先做出x1和x2变量，然后生成数据集a

```
> x1<-1:10
> x2<-c(2,3,5,3,6,2,1,5,3,2)
> a<-data.frame(x1,x2)
```

注意此时x1和a\$x1是两个完全独立的变量，不能简单用x1当成数据集a中的x1变量

- R语言中变量名虽然可以自由命名，但需要符合一定的基本规则，常见的情况大致总结如下：

哪些不可以：

数字开头的不可以，如2f

符号开头的不可以，如*f

纯数字的不可以，如123 不能做变量名

已有的函数名不能占用，你可以创建变量a 变量b 但不能创建变量c，因为c()是基础包中的组合函数，同理，不要创建变量sum mean等（Rstudio中会自动提示函数名）

哪些可以呢：

可以用汉字，但是严重建议不要使用汉字做变量名，会引起很多混乱。

可以用点.放在中间隔开，如xu.ning是一个变量名

- 数据集的创建用data.frame(),但是创建之后的修改无需再用data.frame。假如数据集tes中包含两个变量Q1和Q2, 如果想再加一个Q3, 可以 tes<-cbind(tes,Q3) , 或者更加直接一点: tes\$n2<-Q3,强行添加新变量n2。
- 外部数据的导入: R语言自己的数据文件格式有两种.Rdata文件, .Rda文件, 这两种文件直接用load("文件名")导入到R中; 如果导入其他格式的数据文件则需要对应的专门函数来导入, 没有任何一种方式能够通吃所有类型数据文件的导入。
 - read.table() 导入.txt文件或者复制在内存剪贴板上的数据
 - read.csv() 导入.csv文件
 - read.xlsx() 导入excel文件 (即.xlsx文件)
 - read.dta() 导入.dta文件 (.dta文件是STATA软件的数据保存文件)

其实规律已经非常明显了

- read.dta() 导入文件时总是失败。外部数据导入到R中时都是存储为data.frame变量结构，在data.frame中符号、文字等非数字是直接转换为factor类型，但是R无法正确识别存在空缺时的非数字变量，因此在导入时如果可能有空缺值存在，则关闭转换为factor类型的参数，即convert.factors=FALSE

```
> tes5<-read.dta("F:\\NauCloud\\CSDPS2.dta",convert.factors = FALSE)
```

- 日期类型的格式不同能否相减。如存入变量的两个日期2016-07-12和10/07/24，只要赋值时类型正确，即存为日期型格式，均可以正确参与计算，显示格式并不影响存在内存中的真实数值。

- 作业三问题汇总

