

R 学习

目录

第一章 准备	5
1.1 R 语言介绍	5
1.2 R 命令介绍	5
1.3 关于包的安装与使用	5
1.3.1 安装对应的包	5
1.3.2 包的使用	6
1.4 查看帮助	6
第二章 基础语法	7
2.1 算术操作和向量运算	7
2.2 向量的组合	8
2.3 字符和字符向量	8
2.4 关于正则序列	8
2.5 一些奇怪的函数	9
2.6 矩阵的操作	9

第一章 准备

1.1 R 语言介绍

R 语言和其他的语言之间提供了非常好的接口。

- R 语言对大小写敏感。
- 基本的命令是表达式或者赋值。
- 命令可以被 ; 隔开。
- 注释符号用 #

R 的缺点:

- 耗内存, 所以要用 rm 命令来删除对象, 以释放内存, 如:rm(x,y,z)。
- 精度有问题

1.2 R 命令介绍

在 linux 下面使用 R 的时候, 我们一般用到的命令是:

```
R --vanilla <plot.R > a.out
```

这里的 -vanilla 是参数, 当然还有别的很多参数, 具体有兴趣可以用 man 命令查看。

<plot.R 的意思是将 plot-R 这个文件作为一个输入。既然有输入, 那自然有输出, >a.out 的意思就是将显示的内容输出到 a.out 这个文件中。

1.3 关于包的安装与使用

1.3.1 安装对应的包

R 语言中, 有多种方式可以导入数据包, 如下, 一目了然:

```
library(Hmisc)
source("plotter.R")
```

如果没有对应的包，就要安装，其中 `source("plotter.R")`，这个 `plotter.R` 文件是在当前的目录下的。在 linux 下，进入 R 之后，用如下命令：

```
install.packages("ggplot2")  
install.packages("Hmisc")
```

1.3.2 包的使用

用 `source` 命令导入包，如：`>source("plot.R")`

1.4 查看帮助

如查看 `solve` 的帮助：

- `>help(solve)`
- `>?solve`

对于一些特殊的字符串可以加上双引号，如：`>help("[[")`

第二章 基础语法

`mode(X)` # 可以查看变量的类型

2.1 算术操作和向量运算

创建含有 5 个值的向量 `x`:

- `x<-c(10,2.5,3.4,2,6,1)`
- `x=c(10,2.5,3.4,2,6,1)`
- `assign("x",c(10,2.5,3.4,2,6,1))`
- `c(10,2.5,3.4,2,6,1)->x`

下面是一些简单的应用:

- `1/x` # 显示 `x` 的倒数。
- `y<-c(x,0,x)` # 创建 `y` 向量。
- `v<-2*x+y+1` #

接下来是一些常用的数学函数:

`log(X)` `log` 函数

`exp(X)` 以 `e` 为底的指数函数

`sin(X)` `sin`

`cos(X)` `cos`

`tan(X)` `tan`

`sqrt(X)` 对里面的数开根号

`max(X)`

`min(X)`

`length(X)`

`sum(X)`

`prod(X)` 得到向量中所有数的乘积

`mean(X)` 得到均值

`var(X)` 得到方差

`sort(X)` 对 X 进行排序

`rev(X)` 颠倒向量

`sd(X)` 得到标准差

2.2 向量的组合

`X1=c(1,2,3,4)`

`X2=c(5,6,7,8)`

`rbind(X1,X2)` # 得到一个排列的矩阵

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}$$

`cbind(X1,X2)` # 另一种排列方式

$$\begin{bmatrix} 1 & 5 \\ 2 & 6 \\ 3 & 7 \\ 4 & 8 \end{bmatrix}$$

2.3 字符和字符向量

字符向量既可以用双引号也可以使用单引号。

`\n` # 换行 `\t` # 制表符 `\b` # 退格

`c()` 可以将几个字符向量链接成一个字符向量。

`paste()` 可以进行任意的链接。

`letters`, 这是一个特殊的向量, 里面包含 26 个字母。例如 `letters[2]='b'`

2.4 关于正则序列

`1:30` 等价于 `c(1,2,...,29,30)` # 请注意, 冒号: 的优先级别是最高的。

`30:1` 也是同样的道理, 可以产生逆向序列。

`seq(2,100,by=2)`# 指定公差, 表示 (2,4,...,98,100), `by` 就公差的意思。
`seq(5,121,length=10)`# 指定长度。
`a[i]` 表示 `a` 向量中的第 `i` 个元素。
`a[2,3,4]` 无法显示, 报错如下:`Error in a[2, 4] : incorrect number of dimensions`
`a[2:4]` 表示 `a` 向量中第二和到第四个元素, 返回的是一个 3 个数值的向量。
`a[-1]` 表示第一个不显示。
`a[-(1:3)]` 表示第一个到第三个不显示。

2.5 一些奇怪的函数

函数 `is.na(X1)` 表示返回一个和 `X1` 长度相同的向量, 里面的值为 `FALSE`。
`which.max(a)`# 显示 `a` 向量中最大值的下标, 不可以用对字符向量进行该操作。
`which.min(a)`

2.6 矩阵的操作

`a1=c(1:12)`
`matrix(a1,nrow=3,ncol=4),`
`matrix(a1,nrow=3,ncol=4,byrow=T)` 显示如下:

```
> a1<-c(1:12)
> a1
[1] 1 2 3 4 5 6 7 8 9 10 11 12
> matrix(a1,nrow=3,ncol=4)
[,1] [,2] [,3] [,4]
[1,] 1 4 7 10
[2,] 2 5 8 11
[3,] 3 6 9 12
> matrix(a1,nrow=3,ncol=4,byrow=T)
[,1] [,2] [,3] [,4]
[1,] 1 2 3 4
[2,] 5 6 7 8
[3,] 9 10 11 12
```

2.6.1 矩阵乘法

`a%*%b` 注意, 中间没有空格。

```
> a1%*%a1
[,1]
```

```
[1,] 650
```

```
> a1%%t(a1)
```

[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]	[,11]	[,12]	
[1,]	1	2	3	4	5	6	7	8	9	10	11	12
[2,]	2	4	6	8	10	12	14	16	18	20	22	24
[3,]	3	6	9	12	15	18	21	24	27	30	33	36
[4,]	4	8	12	16	20	24	28	32	36	40	44	48
[5,]	5	10	15	20	25	30	35	40	45	50	55	60
[6,]	6	12	18	24	30	36	42	48	54	60	66	72
[7,]	7	14	21	28	35	42	49	56	63	70	77	84
[8,]	8	16	24	32	40	48	56	64	72	80	88	96
[9,]	9	18	27	36	45	54	63	72	81	90	99	108
[10,]	10	20	30	40	50	60	70	80	90	100	110	120
[11,]	11	22	33	44	55	66	77	88	99	110	121	132
[12,]	12	24	36	48	60	72	84	96	108	120	132	144