

R数据对象II



课堂测试时间

- 1、填写下表：

功能	函数	功能	函数
1:4-1		sep(1.3, 5, 1)	
rep(1:2, 3)		x <- c(1,4,6,7); x[-2]; x*x;	
z <- c(1,5,3,8,1) z[z>3], order(z)		y <- matrix(1:10, nrow=2) y[2, c(4,5)]	
	sort/sort.list()/order		which.min()/which.max()

- 2、请列举R中常见的数据类型数据结构，描述矩阵和数据框的区别？
- 3、使用rep()函数构建一个向量x。向量是由5个1，3个2，4个3和2个4组成的。
- 4、写出命令：1) 创建一个矩阵mat,矩阵的值为1-20之间的整数，四行五列，按行排列
2) 获取矩阵的第三行第四列的元素值 3) 获取矩阵第一行的所有值。
- 5、写出命令：1) 创建字符串向量names,元素值为"zhang san"、"li si"、"wang wu",创建数值型向量scores,元素为70、80、90，创建字符串向量levels,元素值为"C""B""A"; 2)将levels转换成因子类型 3) 创建数据框exam,其列向量为names、scoes、levels; 4) 根据列名称，提出第一列和第三列。

- 数据结构定义: `c()`; `matrix()`; `array()`; `data.frame()`; `factor()`; `list()`;
- 数据结构访问: 下标; 下标向量; 逻辑向量; 负下标;
- 向量: `::`; `seq()`; `rep()`;
- 算术运算符: `+`; `-`; `*`; `/`; `**`; `^`; `%%`; `%/`;
- 逻辑运算: `>`; `<`; `>=`; `<=`; `==`; `!=`; `!`; `|`; `&`; `isTRUE()`; `identical()`; `any()`; `all()`;
- 属性函数: `length()`; `dim()`; `class()`; `names()`; `head()`; `tail()`;
- 排序函数: `order()`; `sort()`; `sort.list()`; `which()`; `which.max()`; `which.min()`;
- 运算函数: `max()`; `min()`; `range()`; `sum()`; `prod()`; `sqrt()`; `abs()`;
- 类型函数: `is.numeric()`; `is.integer()`; `is.logical()`; `is.character()`; `as.xxxx()`;
- 其余函数: `attach()`; `detach()`; `with()`; `$`; `t()`; `diag()`; `solve()`; `eigen()`;

- 矩阵运算
- 缺失值处理
- 类型转换
- 数据集合并
- 字符处理
- 日期和时间
- apply函数
- 统计函数

<code>t()</code>	矩阵转置
<code>det()</code>	求方阵行列式的值
<code>crossprod(x,y)</code>	x和y的内积(<code>%*%</code>)
<code>tcrossprod(x,y)</code>	x和y的外积(<code>%o%</code>), <code>outer()</code>
<code>diag()</code>	生成对角阵和矩阵取对角运算
<code>solve()</code>	解线性方程组, 求矩阵的逆
<code>eigen()</code>	求矩阵的特征值和特征向量

```
> A <- matrix(1:6, nrow = 2)
```

```
> A
```

```
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

```
> t(A)
```

```
      [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
```

```
> x <- 1:5
```

```
> y <- 2*1:5
```

```
> x %*% y
```

```
      [,1]
[1,]  110
```

```
> crossprod(x,y)
```

```
      [,1]
[1,]  110
```

```
> x %o% y
```

```
      [,1] [,2] [,3] [,4] [,5]
[1,]    2    4    6    8   10
[2,]    4    8   12   16   20
[3,]    6   12   18   24   30
[4,]    8   16   24   32   40
[5,]   10   20   30   40   50
```

```
> tcrossprod(x,y)
```

```
      [,1] [,2] [,3] [,4] [,5]
[1,]    2    4    6    8   10
[2,]    4    8   12   16   20
[3,]    6   12   18   24   30
[4,]    8   16   24   32   40
[5,]   10   20   30   40   50
```

```
> outer(x,y,FUN = "*")
```

```
      [,1] [,2] [,3] [,4] [,5]
[1,]    2    4    6    8   10
[2,]    4    8   12   16   20
[3,]    6   12   18   24   30
[4,]    8   16   24   32   40
[5,]   10   20   30   40   50
```

```
> det(matrix(1:4,ncol = 2))
```

```
[1] -2
```

```
> A <- array(1:9,dim=c(3,3))
> B <- array(1:9,dim=c(3,3))
> C <- A * B
> C
```

```
      [,1] [,2] [,3]
[1,]    1   16   49
[2,]    4   25   64
[3,]    9   36   81
```

```
> D <- A %*% B
> D
```

```
      [,1] [,2] [,3]
[1,]   30   66  102
[2,]   36   81  126
[3,]   42   96  150
```

```
> M <- array(1:9, dim=c(3,3))
> M
```

```
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

```
> diag(M)
[1] 1 5 9
```

```
> A <- t(array(c(1:8,10), dim = c(3,3)))
```

```
> b <- c(1,1,1)
> x <- solve(A,b)
```

```
> x
[1] -1.000000e+00  1.000000e+00  3.330669e-16
```

```
> B <- solve(A)
```

```
> B
      [,1]      [,2] [,3]
[1,] -0.6666667 -1.333333  1
[2,] -0.6666667  3.666667 -2
[3,]  1.0000000 -2.000000  1
```

```
> Sm <- crossprod(A,A)
```

```
> ev <- eigen(Sm)
```

```
> ev
```

```
$values
```

```
[1] 303.19533618  0.76590739  0.03875643
```

```
$vectors
```

```
      [,1]      [,2]      [,3]
[1,] -0.4646675  0.833286355  0.2995295
[2,] -0.5537546 -0.009499485 -0.8326258
[3,] -0.6909703 -0.552759994  0.4658502
```

```
> A
```

```
      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8   10
```


表4-1 领导行为的性别差异

经理人	日期	国籍	性别	年龄	q1	q2	q3	q4	q5
1	10/24/08	US	M	32	5	4	5	5	5
2	10/28/08	US	F	45	3	5	2	5	5
3	10/01/08	UK	F	25	3	5	5	5	2
4	10/12/08	UK	M	39	3	3	4		
5	05/01/09	UK	F	99	2	2	1	2	1

- 例子见教材74页
- **NA**
- **is.na()**
- **na.rm = TRUE**
- **na.omit()**

```
> y <- c(1,2,3,NA)
> is.na(y)
[1] FALSE FALSE FALSE TRUE
```

```
> sum(1:5, NA)
[1] NA
> sum(1:5, NA, na.rm = TRUE)
[1] 15
```

看：例子4-3和4-4

表4-5 类型转换函数

判 断	转 换
<code>is.numeric()</code>	<code>as.numeric()</code>
<code>is.character()</code>	<code>as.character()</code>
<code>is.vector()</code>	<code>as.vector()</code>
<code>is.matrix()</code>	<code>as.matrix()</code>
<code>is.data.frame()</code>	<code>as.data.frame()</code>
<code>is.factor()</code>	<code>as.factor()</code>
<code>is.logical()</code>	<code>as.logical()</code>

- 见教材78页

看：例子4-5

```
> x1 <- rbind(c(1,2),c(3,4))
> x1
      [,1] [,2]
[1,]    1    2
[2,]    3    4
> x2 <- 10 + x1
> x3 <- cbind(x1, x2)
> x3
      [,1] [,2] [,3] [,4]
[1,]    1    2   11   12
[2,]    3    4   13   14
> x4 <- rbind(x1,x2)
> x4
      [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]   11   12
[4,]   13   14
> cbind(1, x1)
      [,1] [,2] [,3]
[1,]    1    1    2
[2,]    1    3    4
```

- 见教材93页

<code>nchar()</code>	计算x中的字符数
<code>substr(s,start,stop)</code>	提取或替换一个字符向量中的子串
<code>strsplit(x,split)</code>	在split处分割字符向量x中的元素
<code>toupper(x), tolower()</code>	大小写转换
<code>paste(..., sep="")</code>	连接字符串
<code>grep(pattern,x,ignore.case=FALSE,fixed=FLASE)</code>	搜索 
<code>sub(pattern,replacement,x,ignore.case=FALSE,fixed=FLASE)</code>	搜索替换 

```
> paste("My", "Job")
[1] "My Job"
>
> labs <- paste("X", 1:6, sep="")
> labs
[1] "X1" "X2" "X3" "X4" "X5" "X6"
>
> paste("Today is", date())
[1] "Today is Wed Mar  2 12:41:21 2016"
>
> paste(c("a", "b"), collapse=".")
[1] "a.b"
```

- 见教材76页

日期函数	<i>as.Date(x, "input_format")</i>
<i>%d</i>	数字表示的日期 (0-31)
<i>%a, %A</i>	星期名 (缩写, 非缩写)
<i>%m</i>	月份 (0-12)
<i>%b, %B</i>	月份 (缩写, 非缩写)
<i>%y, %Y</i>	年份 (两位, 四位)
<i>Sys.Date(), date(), difftime(), format()</i>	

```
> mydates <- as.Date(c("2007-06-22"))
> mydates
[1] "2007-06-22"
> mydates <- as.Date(c("2007-06-22"))
>
> strDates <- c("01/05/1965")
> dates <- as.Date(strDates, "%m/%d/%Y")
>
> Sys.Date()
[1] "2016-03-02"
> date()
[1] "Wed Mar  2 12:48:52 2016"
```

```
> today <- Sys.Date()
> format(today, format = "%B %d %Y")
[1] "March 02 2016"
> format(today, format = "%A")
[1] "Wednesday"
>
> startdate <- as.Date("2004-02-13")
> enddate <- as.Date("2009-06-22")
> days <- enddate - startdate
>
> today <- Sys.Date()
> format(today, format = "%B %d %Y")
[1] "March 02 2016"
> dob <- as.Date("1956-10-10")
> format(dob, format = "%A")
[1] "Wednesday"
> difftime(today, dob, units="weeks")
Time difference of 3099 weeks
```


apply(x, MARGIN, FUN, ...)

```
> a <- matrix(1:6,nrow=2)
> a
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> apply(a, 1,sum)
[1]  9 12
> apply(a,2,sum)
[1]  3  7 11
```

- 见教材95页

看：例子5-5和5-6

```
> mydata <- matrix(rnorm(30), nrow=6)
> mydata
      [,1]      [,2]      [,3]      [,4]      [,5]
[1,]  1.1039131 -0.6779796  0.09072753  0.6943354 -0.68360455
[2,] -1.2876154  0.1540778  1.41431948 -0.9622685  2.07486216
[3,] -0.4221483  0.3073955  0.36975022 -0.4124088  0.54614267
[4,] -0.5283792 -0.7510899 -1.58224514  1.1124982  0.24044145
[5,] -1.1322217  1.0616374  0.37744029  0.1879165 -0.03192165
[6,]  0.7633084  0.6153539  0.58158158  0.3485943  0.17747101
> apply(mydata, 1, mean)
[1]  0.10547839  0.27867512  0.07774626 -0.30175492  0.09257018  0.49726184
> apply(mydata, 2, mean)
[1] -0.2505238  0.1182325  0.2085957  0.1614445  0.3872318
> apply(mydata, 2, mean, trim=.4)
[1] -0.4752638  0.2307367  0.3735953  0.2682554  0.2089562
```

- 见教材87页

<i>mean(x), median(x)</i>		平均数, 中位数
<i>sd(x), var(x)</i>		标准差, 方差
<i>max(x), min(x)</i>		最大值, 最小值
<i>range(x), sum(x)</i>	★	值域, 求和
<i>quantile(x, prob)</i>	★	求分位数
<i>diff(x, lag=n)</i>	★	滞后差分
<i>scale(x, center=TRUE, scale=TRUE)</i>	★	为数据对象x按列进行中心化和标准化
<i>str(), summary()</i>		

看: 例子5-1

提问时间!

孙惠平

sunhp@ss.pku.edu.cn

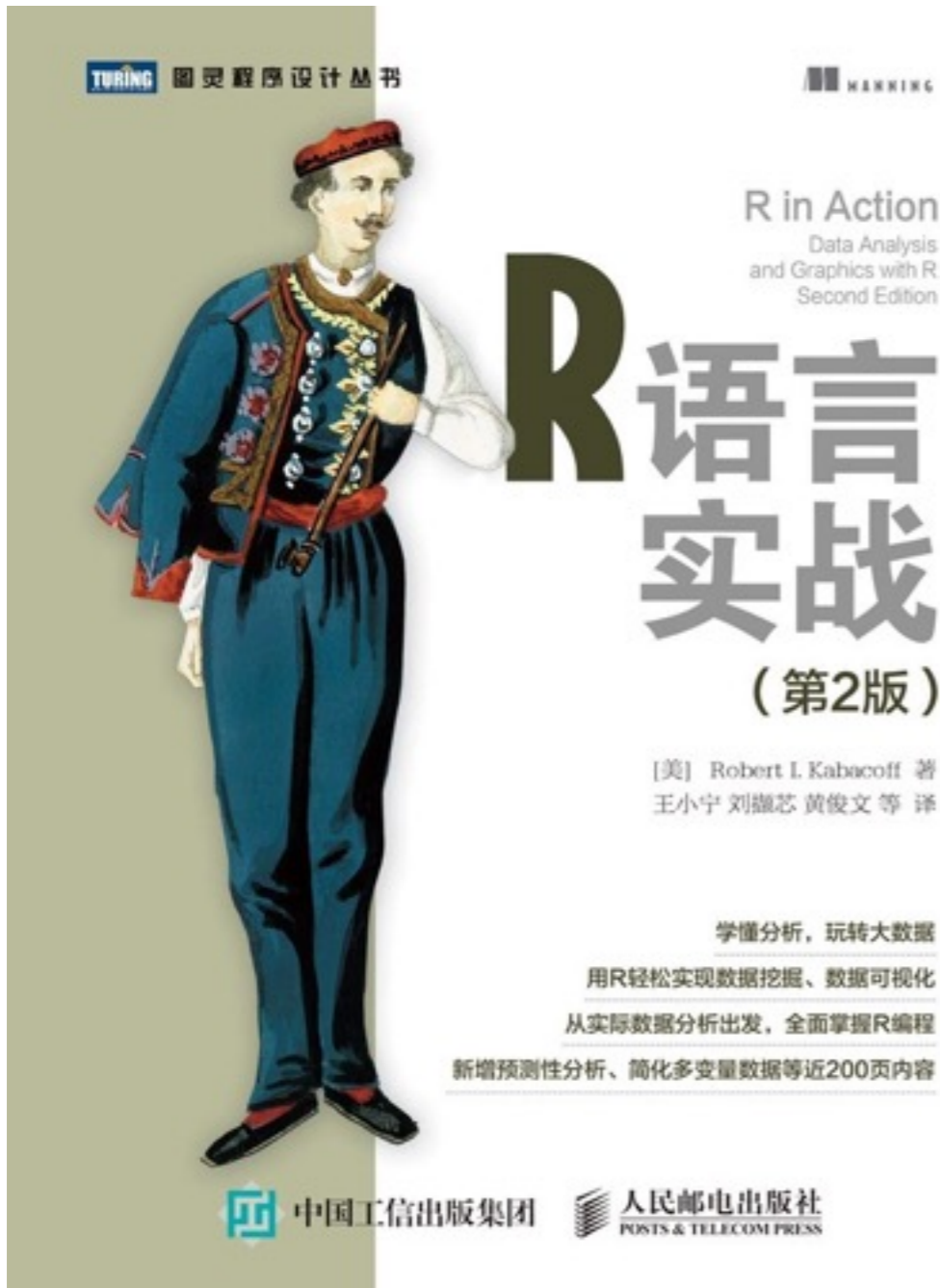
练习



Learn R, in R.

swirl teaches you R programming and data science
interactively, at your own pace, and right in the R console!

`install_course_github("pkusdatanalysis", "C3_Data_Structure_02")`



第四章和第五章



第五章、第六章、第七章

- 完成课后大作业0001
- 按照要求提交

谢谢!

孙惠平

sunhp@ss.pku.edu.cn