

# 《数据统计分析与R语言》

## 基本数据管理



# 内容回顾

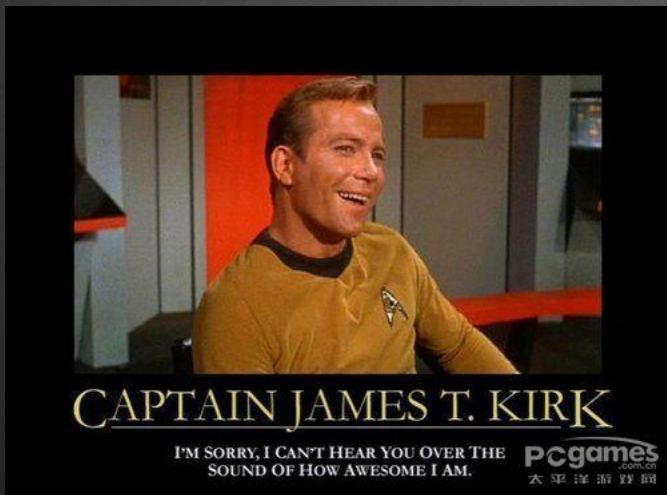
- 🦋 数据集的概念
- 🦋 数据结构
- 🦋 数据类型
- 🦋 数据的输入





# 引入

🌀 Kirk船长：“数据是一件麻烦事，一件非常非常麻烦的事。”





# 引入

🌀 研究主题：男性和女性在领导各自企业方式上的不同

典型问题：处于管理岗位的男性和女性在听从上级的程度上是否有所不同？这种情况是否依国家的不同而有所不同，或者说这些由性别导致的不同是否普遍存在？





# 引入

每位经理人的上司根据与服从程度的五项陈述（q1到q5）对经理人进行评分

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

这名经理在做出人事决策之前会询问我的意见。

1	2	3	4	5
非常不同意	不同意	既不同意也不反对	同意	非常同意



# 创建新变量

🔗 语句：变量名←表达式

“表达式”可以包含多种运算符和函数。下表列出R中的算术运算符

运 算 符	描 述
+	加
-	减
*	乘
/	除
^或**	求幂
$x \% y$	求余 ( $x \bmod y$ )。5%2的结果为1
$x \%/\% y$	整数除法。5%/%2的结果为2



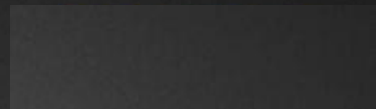
# 变量重编码

使用R中的一个或多个逻辑运算符

运 算 符	描 述
<	小于
<=	小于或等于
>	大于
>=	大于或等于
==	严格等于*
!=	不等于
!x	非x
x   y	x或y
x & y	x和y
isTRUE(x)	测试x是否为TRUE



# 变量的重编码



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





# 变量的重命名

❧ 若对现有的变量名称不满意，可以交互地或者以编程的方式修改

交互式 `fix()`

编程方式 `rename()` 在plyr包下面

```
rename(dataframe, c(olddname="newname",  
olddname="newname",...))
```

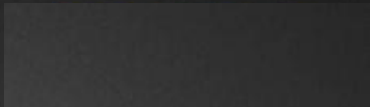
The screenshot shows the R environment with the following components:

- R Console (64-bit):** Displays the command `> fix(leadership)`.
- Data Editor:** A window showing a dataset with 10 rows and 10 columns. The columns are: managerID, date, country, gender, age, q1, q2, q3, q4. The data is as follows:

	managerID	date	country	gender	age	q1	q2	q3	q4
1	1	10/24/2008	US	M	32	5	4	5	5
2	2	10/28/2008	US	F	45	3	5	2	5
3	3	10/1/2008	UK	F	25	3	5	5	5
4	4	10/12/2008	UK	M	39	3	3	4	NA
5	5	5/1/2009	UK	F	99	2	2	1	2
6									
7									
8									
9									
10									
- Variable editor:** A small dialog box for editing the 'managerID' variable. It shows the variable name 'managerID' and the type 'numeric' selected.



# 缺失值

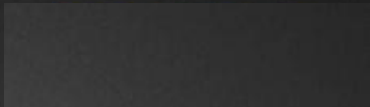


- ❧ 缺失值用符号NA表示
- ❧ 不可能出现的值用符号NaN ( Not a Number , 非数值 ) 表示





# 缺失值



🌀 重编码某些值为缺失值

```
leadership$age[leadership$age == 99] <- NA
```





# 缺失值

🔗 在分析中排除缺失值

```
x <- c(1, 2, NA, 3)
```

```
y <- x[1] + x[2] + x[3] + x[4]
```

```
z <- sum(x)
```

#通过na.rm=TRUE选项移除缺失值

```
x <- c(1, 2, NA, 3)
```

```
y <- sum(x, na.rm=TRUE)
```

#通过数na.omit()移除所有含有缺失值

```
leadership
```

```
newdata <- na.omit(leadership)
```

```
newdata
```





# 日期值

将默认格式的字符型数据转换为日期值

为 `as.Date(x, "input_format")`

示例：

```
strDates <- c("01/05/1965", "08/10/1975")
```

```
dates <- as.Date(strDates, "%m/%d/%Y")
```

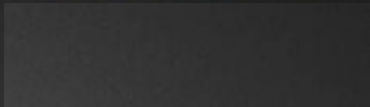
Table 4.4. Date formats

Symbol	Meaning	Example
%d	Day as a number (0-31)	01-31
%a	Abbreviated weekday	Mon
%A	Unabbreviated weekday	Monday
%m	Month (00-12)	00-12
%b	Abbreviated month	Jan
%B	Unabbreviated month	January
%y	2-digit year	07
%Y	4-digit year	2007





# 日期值



- ❧ Sys.Date()可以返回当天的日期
- ❧ date() 返回当前的日期和时间
- ❧ format(x, format="output\_format") 输出指定格式的日期值





# 日期值

## 🕒 日期时间间隔的计算

方法一：

```
startdate <- as.Date("2004-02-13")
```

```
enddate <- as.Date("2011-01-22")
```

```
days <- enddate - startdate
```

Time difference of 2535 days

方法二：使用函数difftime()来计算时间间隔

```
today <- Sys.Date()
```

```
dob <- as.Date("1988-10-12")
```

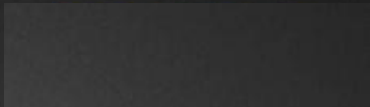
```
difftime(today, dob, units="weeks")
```

Time difference of 2825 weeks





# 日期值



将日期转换为字符型变量

```
strDates <- as.character(dates)
```





# 类型转换

🔗 `is.datatype()` 返回TRUE或FALSE

🔗 `as.datatype()` 将其参数转换为对应的类型

Test	Convert
<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>



# 数据排序

使用order()函数对一个数据框排序

```
newdata <- leadership[order(leadership$age),]
```

示例:

```
attach(leadership)
```

```
newdata <- leadership[order(gender, age),]
```

```
detach(leadership)
```

```
attach(leadership)
```

```
newdata <- leadership[order(gender, age),]
```

```
detach(leadership)
```







# 数据集的合并

## 添加列

使用 merge()函数

```
total <- merge(dataframeA, dataframeB, by="ID")
```

## #添加列

```
x <- matrix(c(1,3,4,5,6,7,8,9),nrow = 3,ncol = 3)
```

```
y <- x
```

```
z <- cbind(x,y)
```

```
manager <- c(1,2,3,4,5)
date <- c("10/24/08","10/28/08","10/1/08","10/12/08","5/1/09")
country <- c("US","US","UK","UK","UK")
gender <- c("M","F","F","M","F")
age <- c(32,45,25,39,99)
dataframeA <- data.frame(manager,date,country,gender,age,
                           stringsAsFactors=FALSE)

manager <- c(1,2,3,4,5)
q1 <- c(5,3,3,3,2)
q2 <- c(4,5,5,3,2)
q3 <- c(5,2,5,4,1)
q4 <- c(5,5,5,NA,2)
q5 <- c(5,5,2,NA,1)

dataframeB <- data.frame(manager,q1,q2,q3,q4,q5,
                           stringsAsFactors=FALSE)

total <- merge(dataframeA, dataframeB, by="manager")
```

	manager	date	country	gender	age	q1	q2	q3	q4	q5
1	1	10/24/08	US	M	32	5	4	5	5	5
2	2	10/28/08	US	F	45	3	5	2	5	5
3	3	10/1/08	UK	F	25	3	5	5	5	2
4	4	10/12/08	UK	M	39	3	3	4	NA	NA
5	5	5/1/09	UK	F	99	2	2	1	2	1



# 数据集的合并

添加行 ( rbind()函数 )

```
total <- rbind(dataframeA, dataframeB)
```

注意：两个数据框必须拥有相同的变量，顺序不必一定相同

```
x <- matrix(c(1,3,4,5,6,7,8,9),nrow = 3,ncol = 3)
```

```
y <- x
```

```
r <- rbind(x,y)
```





# 数据集取子集

🦌 选入（保留）变量

方法一：

```
newdata <- leadership[, c(6:10)]
```

方法二：

```
myvars <- c("q1", "q2", "q3", "q4", "q5")
```

```
newdata <- leadership[myvars]
```





# 数据集取子集

🦋 剔除（丢弃）变量

```
myvars <- names(leadership) %in% c("q3", "q4")  
newdata <- leadership[!myvars]
```

知道列序号的情况下：

```
newdata <- leadership[c(-8,-9)]
```

相同的变量删除：

```
leadership$q3 <- leadership$q4 <- NULL
```







# 数据集取子集

🔗 选入观测

```
newdata <- leadership[1:3,]
```

```
newdata <- leadership[which(leadership$gender=="M" &  
                             leadership$age > 30),]
```

```
attach(leadership)
```

```
newdata <- leadership[which(gender=='M' & age > 30),]
```

```
detach(leadership)
```







# 数据集取子集

将研究范围限定在2009年1月1日到2009年12 月31日之间

```
leadership$date <- as.Date(leadership$date, "%m/%d/%y")  
startdate <- as.Date("2009-01-01")  
enddate <- as.Date("2009-10-31")  
newdata <- leadership[which(leadership$date >= startdate &  
leadership$date <= enddate),]
```





# 数据集取子集

🔗 subset()函数

```
newdata <- subset(leadership, age >= 35 | age < 24,  
  select=c(q1, q2, q3, q4))
```

```
newdata <- subset(leadership, gender=="M" & age > 25,  
  select=gender:q4)
```





# 数据集取子集

🌀 随机抽样 sample()函数

```
mysample <- leadership[sample(1:nrow(leadership), 3,  
replace=FALSE),]
```



# 数据集取子集

🔗 使用SQL语句操作数据框

```
library(sqldf)
newdf <- sqldf("select * from mtcars where carb=1 order by mpg",
               row.names=TRUE)
newdf
```



# 数据集取子集

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Valiant	18.1	6	225.0	105	2.76	3.46	20.2	1	0	3	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.21	19.4	1	0	3	1
Toyota Corona	21.5	4	120.1	97	3.70	2.46	20.0	1	0	3	1
Datsun 710	22.8	4	108.0	93	3.85	2.32	18.6	1	1	4	1
Fiat X1-9	27.3	4	79.0	66	4.08	1.94	18.9	1	1	4	1
Fiat 128	32.4	4	78.7	66	4.08	2.20	19.5	1	1	4	1
Toyota Corolla	33.9	4	71.1	65	4.22	1.83	19.9	1	1	4	1





# 数据集取子集

```
sqldf("select avg(mpg) as avg_mpg, avg(displacement) as avg_disp, gear  
      from mtcars where cyl in (4, 6) group by gear")
```

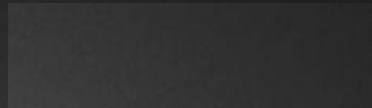
	avg_mpg	avg_disp	gear
1	20.3	201	3
2	24.5	123	4
3	25.4	120	5



# 小结

- 🌀 操作日期和缺失值
- 🌀 熟悉数据类型的转换
- 🌀 变量的创建和重编码
- 🌀 数据集的排序、合并和取子集
- 🌀 选入和丢弃变量





**Thankyou !**

