

数据管理与可视化

二、数据处理基础

毛益民 副教授

浙江工商大学公共管理学院

一些英文单词

- sequence
- replicate
- merge
- combine
- gather
- spread
- row
- column
- bind
- head
- tail

- mutate
- select
- filter
- subset
- plot
- figure
- graph
- layer
- summary
- character
- numeric
- factor

数据类型

- 数值型

```
a <- 1; is.numeric(a)
```

```
## [1] TRUE
```

- 字符型

```
b <- 'peter'; nchar(b)
```

```
## [1] 5
```

- 日期型

```
c <- as.Date('2019-09-16')  
class(c)
```

```
## [1] "Date"
```

- 逻辑型

```
e <- TRUE  
class(e)
```

```
## [1] "logical"
```

数据结构

向量(vector)

- 向量是用于存储数值型、字符型或逻辑型数据的一维数组。
- 执行组合功能的函数`c()`可用来创建向量。

```
c(2,4,6);seq(2,4,by=0.5);rep(1:3,time=3)
```

```
## [1] 2 4 6
```

```
## [1] 2.0 2.5 3.0 3.5 4.0
```

```
## [1] 1 2 3 1 2 3 1 2 3
```

向量需要定义，否则结果会很残忍

输入：lalala

输出：找不到对象 'lalala'

向量的处理

- 向量的排序

```
lalala <- c(1,3,5,2,4)
order <- sort(lalala,index.return=TRUE,decreasing = TRUE)
order
```

```
## $x
## [1] 5 4 3 2 1
##
## $ix
## [1] 3 5 2 4 1
```

- 向量的唯一值

```
xing <- c('zhang','li','wang','mao','zhang','wang')
unique(xing)
```

```
## [1] "zhang" "li"      "wang"   "mao"
```

- 向量的离散化

```
num <- c(10,5,4,7,6,1,4,3,8,9,2)
cut_num <- cut(num,breaks = c(0,3,6,9,11),
               labels = c('0-3','3-6','6-9','>9'),right=TRUE)
cut_num
```

```
## [1] >9 3-6 3-6 6-9 3-6 0-3 3-6 0-3 6-9 6-9 0-3
## Levels: 0-3 3-6 6-9 >9
```

- 向量的索引

```
x <- c(1,2,3,4,5)
x[3];x[x>3]
```

```
## [1] 3
```

```
## [1] 4 5
```

不难不难，甚至有点轻松



数据结构

因子(factor)

因子可以看成是包含了额外信息的向量，这额外的信息就是不同的类别，称之为说 (level)

```
cut <- c('差', '一般', '良好', '优秀')  
as.factor(cut)
```

```
## [1] 差    一般 良好 优秀  
## Levels: 一般 优秀 差 良好
```


数据结构

数据框(data.frame)

- 数据框是一种表格结构，类似于EXCEL中的数据表。
- 数据框是由多个向量构成的，每个向量的长度相同。

```
# 数据框的创建
df <- data.frame(
  x=c('a', 'b', 'c'),
  y=1:3,
  z=c(2,5,3)
)
df
```

```
##      x y z
## 1  a 1 2
## 2  b 2 5
## 3  c 3 3
```

记了这么多，先喘口气



然后，继续。。。

请创建如下数据框

```
##      name code score level
## 1  lily    1    80     B
## 2  jame    2    57     C
## 3  jone    3    66     C
## 4 lilei    4    78     B
```

数据属性

类别型：不同类型的数据

- 无序：比如性别（男或女）
- 有序：不太喜欢、喜欢、非常喜欢

数值型：比如成绩

成绩划分等级

- 优秀：90分及以上
- 良好：80-89
- 中等：70-79
- 合格：60-69
- 不合格：0-59

数据的导入与导出

csv文件

```
mydata <- read.csv('data.csv',sep=',',na.strings = 'NA',stringsAsFactors = FALSE)
```

```
write.csv(mydata,file='file.csv')
```

excel文件

```
mydata <- read.xlsx('data.xlsx',sheetindex=1)
```

```
write.xlsx(mydata,file='file.xlsx',sheetName='mysheetname')
```

当然，可以在窗口上操作

记了这么多，再喘口气



然后，再继续。。。

控制语句

理解 if...else...和函数ifelse的用法

if(条件){ 执行语句 } else{ 其它执行语句 }

#例子

```
i <- 5
if(i>3){
  print('yes')
} else {
  print('no')
}
```

```
## [1] "yes"
```

#例子

```
ifelse(i>3,
      'yes',
      'no')
```

```
## [1] "yes"
```

for 循环语句

for (变量 in 向量) { 执行语句 }

#例子

```
for(i in 1:4){  
  j <- i+10  
  print(j)  
}
```

```
## [1] 11  
## [1] 12  
## [1] 13  
## [1] 14
```


让我们再往前走一步

基于数据框的操作：表格的转换与整理

- 这部分挺好玩，也很有用哦！
- 先生成1个数据框文件

```
mydf <- data.frame(x=c('A','B','C'),  
                  '2010'=c(1,3,4),  
                  '2011'=c(3,5,2),  
                  check.names = FALSE)  
  
mydf
```

```
##      x 2010 2011  
## 1 A      1     3  
## 2 B      3     5  
## 3 C      4     2
```

tidyr包 两个非常有用的函数。

gather(): 将短数据变成长数据

spread(): 将长数据变成短数据

wide

id	x	y	z
1	a	c	e
2	b	d	f

long

id	key	val
1	x	a
2	x	b
1	y	c
2	y	d
1	z	e
2	z	f

动态过程

练习一下

```
mydf
```

```
##    x 2010 2011
## 1 A    1    3
## 2 B    3    5
## 3 C    4    2
```

```
df_gather <- tidyr::gather(mydf, year, value, -x)
df_gather
```

```
##    x year value
## 1 A 2010     1
## 2 B 2010     3
## 3 C 2010     4
## 4 A 2011     3
## 5 B 2011     5
## 6 C 2011     2
```

如何恢复原状

```
# 请用spread ()  
df_spread <- tidyr::spread(df_gather,year,value)  
df_spread
```

```
##      x 2010 2011  
## 1 A      1      3  
## 2 B      3      5  
## 3 C      4      2
```

请整理一下全国各省近10年的GDP数据

- 数据来源（点击）
- 由短数据变成长数据

变量的变换

- 着眼于df_gather
- 将里面的value都扩大两倍

```
df_gather2 <- dplyr::mutate(df_gather,  
                             value2=value*2)
```

```
df_gather2
```

```
##   x year value value2  
## 1 A 2010     1      2  
## 2 B 2010     3      6  
## 3 C 2010     4      8  
## 4 A 2011     3      6  
## 5 B 2011     5     10  
## 6 C 2011     2      4
```

练习一下

操作：对于df_gather，请将2011年的value扩大3倍，其它value保留原值

df_gather

##	x	year	value
## 1	A	2010	1
## 2	B	2010	3
## 3	C	2010	4
## 4	A	2011	3
## 5	B	2011	5
## 6	C	2011	2



代码与结果

```
df_gather$value2 <- ifelse(df_gather$year=='2011',  
                           df_gather$value*3,  
                           df_gather$value)  
df_gather
```

```
##   x year value value2  
## 1 A 2010     1      1  
## 2 B 2010     3      3  
## 3 C 2010     4      4  
## 4 A 2011     3      9  
## 5 B 2011     5     15  
## 6 C 2011     2      6
```


也可以用dplyr包

```
df_gather %>%  
  mutate(value3=ifelse(year=='2011',  
                        value*3,  
                        value))
```

```
##   x year value value2 value3  
## 1 A 2010     1      1      1  
## 2 B 2010     3      3      3  
## 3 C 2010     4      4      4  
## 4 A 2011     3      9      9  
## 5 B 2011     5     15     15  
## 6 C 2011     2      6      6
```

表格的拼接

rbind() & cbind()

```
df1 <- data.frame(x=c('a','b','c'),y=1:3)
df1
```

```
##    x y
## 1 a 1
## 2 b 2
## 3 c 3
```

```
df2 <- data.frame(m=c('A','B','C'),n=6:8)
df2
```

```
##    m n
## 1 A 6
## 2 B 7
## 3 C 8
```

横向拼接

```
df_cbind <- cbind(df1,df2)
df_cbind
```

```
##      x y m n
## 1  a 1 A 6
## 2  b 2 B 7
## 3  c 3 C 8
```

纵向拼接

```
df3 <- data.frame(x=c('lala','dada'),y=2:3)
df_rbind <- rbind(df1,df3)
df_rbind
```

```
##      x y
## 1  a 1
## 2  b 2
## 3  c 3
## 4 lala 2
## 5 dada 3
```

表格的融合

```
df1
```

```
##    x y  
##  1 a 1  
##  2 b 2  
##  3 c 3
```

```
df4 <- data.frame(x=c('a','c','b'),  
                  z=c('好','不好','中等'))
```

```
df4
```

```
##    x    z  
##  1 a   好  
##  2 c 不好  
##  3 b 中等
```

使用merge进行融合

```
df_merge <- merge(df1,df4,by='x')  
df_merge
```

```
##      x y      z  
## 1 a 1      好  
## 2 b 2 中等  
## 3 c 3 不好
```

当然，练习是少不了的

也说不出为什么，就是想喊上一嗓子



分组计算均值

dplyr包 group_by 函数

```
iris %>%  
  group_by(Species) %>%  
  summarise_all(mean)
```

```
## # A tibble: 3 x 5  
##   Species      Sepal.Length Sepal.Width Petal.Length Petal.Width  
##   <fct>          <dbl>         <dbl>         <dbl>         <dbl>  
## 1 setosa          5.01           3.43           1.46           0.246  
## 2 versicolor      5.94           2.77           4.26           1.33  
## 3 virginica       6.59           2.97           5.55           2.03
```

分组列出最大值

```
iris %>%  
  group_by(Species) %>%  
  summarise_all(max)
```

```
## # A tibble: 3 x 5  
##   Species    Sepal.Length Sepal.Width Petal.Length Petal.Width  
##   <fct>         <dbl>         <dbl>         <dbl>         <dbl>  
## 1 setosa         5.8           4.4           1.9           0.6  
## 2 versicolor     7             3.4           5.1           1.8  
## 3 virginica      7.9           3.8           6.9           2.5
```


综合练习（看看你真的会了没有？）

为了方便大家练习，且对数据有很好的代入感，我构造了两份数据：

练习数据，点击即可下载：

1. 学生名单，包括班级（class）、姓名（name）、以及性别(gender)。下载 [students.xls](#)
2. 考试成绩，包括姓名(name)以及各科成绩，即语文(chinese)、数学(math)、英语(english)、物理(physics)、化学(chemistry)、生物(biology)。下载 [scores.xls](#)

让我们先来浏览一下数据内容：

学生名单（为了逼真，我还起了396个名字，可见用心良苦!）

```
## # A tibble: 396 x 3
##   class name  gender
##   <dbl> <chr>  <chr>
## 1     1    柏寒    F
## 2     1    柏朗    M
## 3     1    陈依    F
## 4     1    陈宇    M
## 5     1    单天傲  F
## 6     1    单运旭  M
## 7     1    董冬烟  F
## 8     1    董鹏    M
## 9     1    范昆    M
## 10    1    范南    F
## # ... with 386 more rows
```

让我们先来浏览一下数据内容：

考试成绩

```
## # A tibble: 396 x 7
##   name      chinese  math english physics chemistry biology
##   <chr>      <dbl> <dbl>   <dbl>   <dbl>      <dbl>   <dbl>
## 1 安昆          93    69     56      30        63     62
## 2 安香          88    34     71      74        52     57
## 3 柏寒          75    38     59      36        80     54
## 4 柏朗          69   100     71      84        85     73
## 5 包松烟        88    95     43      37        94     66
## 6 包文          94    72     96      99        57     66
## 7 鲍达泽        50    48     88      37        63     79
## 8 鲍雁          83    99     99      48        91     87
## 9 贝和浦        97    56     64      50        87     57
## 10 贝小         65    95     83      41        96     81
## # ... with 386 more rows
```

操作任务

1. 请列出总分最高分和最低分（姓名和分数）
2. 请列出各科成绩最高分的学生（姓名和分数）
3. 请计算各科及格率（60分为及格线）
4. 请按班级分别计算各科平均成绩
5. 请列出各个班级总分最高和最低的学生（姓名和分数）
6. 请分班级比较男生和女生的成绩情况（总分与各科）

每完成一步，请举手示意。我要记录一下，以示表扬。

数据整理就介绍到这里

- 师傅领进门，修行靠个人。
- 其它函数请自己去学习吧
- 接下来我们开始学习绘图。。。



谢谢聆听!

记得回去以后多多练习哦

毛益民

浙江工商大学公共管理学院