



数据分析与处理技术——4数据探索与准备

商学院 徐宁

参考资料

阅读:

1. Wickham的《R数据科学》第3章 dplyr 第7章 tibble 和第8章 readr导入数据;
2. 《R语言—实用数据分析与可视化技术》第12章 高效的分组操作dplyr
3. 北京大学李东风《R语言教程》
[26 数据整理 | R语言教程](#)
[\(\[pku.edu.cn\]\(http://pku.edu.cn\)\)](http://pku.edu.cn)



数据探索与准备

数据存取

数据预览

数据编辑

1.数据导入与预览

常用数据文件格式

常用数据文件类型

- **csv**文件：标准文本数据文件，出现最早也是应用最为广泛的数据文件格式，因无冗余内容通常文件相对较小，大型数据库导出文件常使用**csv**文件
- **xls/xlsx**：excel文件，通过**readr**导入
- **dta**文件：**stata**数据文件，**read.dta()**函数导入

R自有的数据文件

- **rda**文件：**r**语言的数据文件，仅包含单个变量，**save**函数保存，在**Rstudio**的文件窗口中点击加载
- **Rdata**文件：装载多个变量的数据文件，默认保存全局环境全部数据

数据载入与存盘

R自有的数据文件主要是rda和Rdata文件

- rda数据文件的保存

```
a=c(1,3,7,-2,9,10)  
save(a,file="myvariable.rda")
```

需要注意的是，R中“\”是转义符，不能在文件路径中直接使用，而需要换成双斜线“\\”或反斜线方式“/”。

- rda数据文件的载入

```
load(file="~/OneDrive/0-备课/code/myvariable.rda")
```

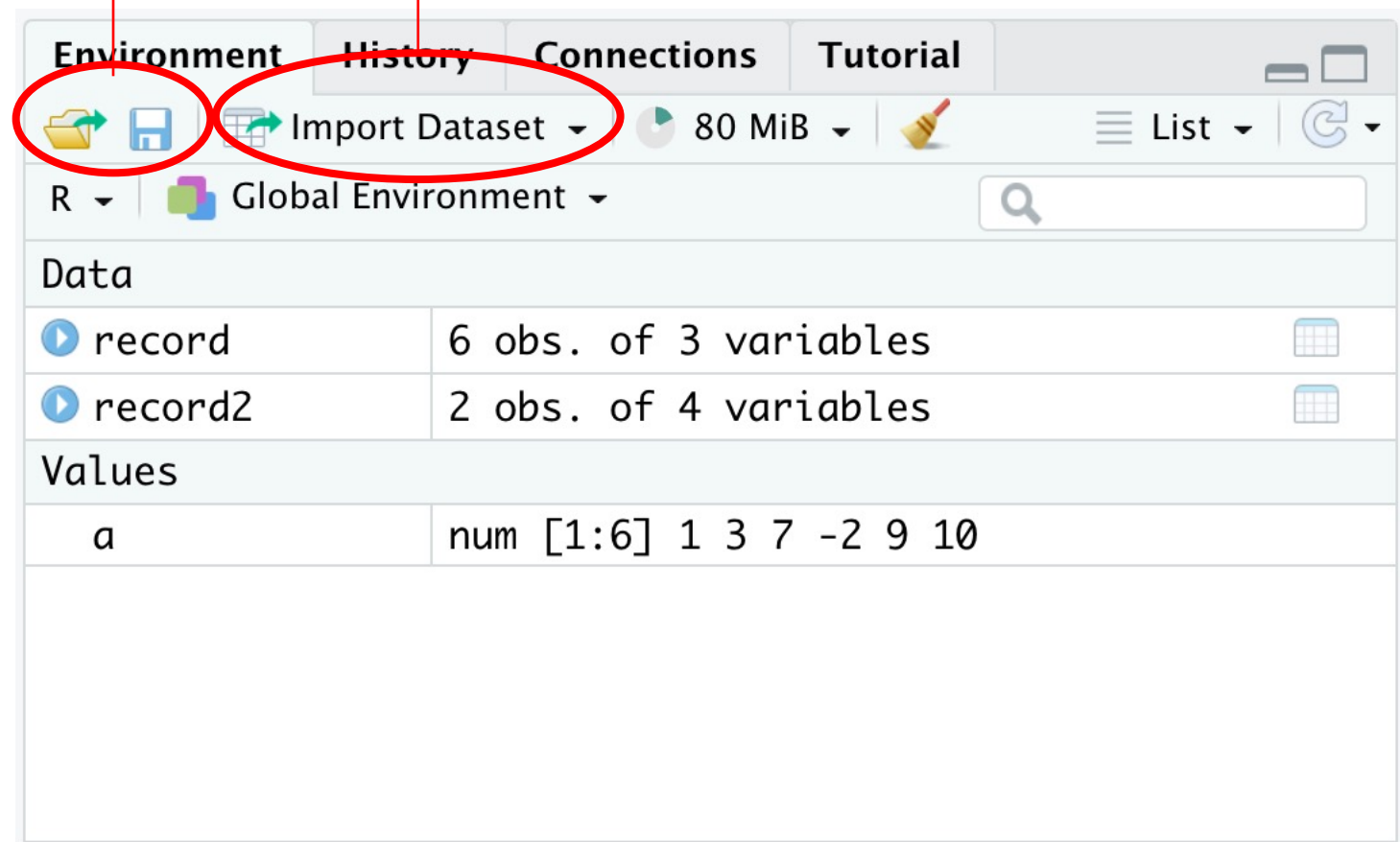
数据存取的图形界面

读取/保存：R自有文件格式

导入/导出：通过转换格式读取或存储其他文件格式，如 `xls`, `xlsx`, `csv`, `dta`, `txt` 等。需要相应的工具包支持。

读取/保存环境数据到Rdata文件

调用工具导入其他格式数据文件



数据导入工具

数据文件所在路径

Import Text Data

File/URL:
~/persons.csv

Browse...

Data Preview:

| X1 (character) | Sex (character) | Class (double) | Age (double) | Math (double) | English (character) | Computer (double) |
|-------------------|--------------------|-------------------|-----------------|------------------|------------------------|----------------------|
| 李雷 | Male | 1 | 19 | 95 | B | 72 |
| 韩梅梅 | Female | 1 | 19 | 88 | A | 67 |
| 张萌 | Female | 2 | 20 | 72 | B | 55 |
| 王珂 | Male | 2 | 19 | 85 | C | 89 |
| 刘红 | Female | 1 | 20 | 56 | B | 75 |
| 潘迎 | Female | 2 | 20 | 64 | B | 67 |
| 张亮 | Male | 1 | 18 | 77 | D | 90 |
| 卫天方 | Male | 2 | 21 | 34 | C | 81 |
| 熊萍萍 | Female | 2 | 20 | 87 | B | 45 |

Previewing first 50 entries.

Import Options:

Name: persons

Skip: 0

☒ First Row as Names
☒ Trim Spaces
☒ Open Data Viewer

Delimiter: Comma
 Quotes: Default
 Locale: Configure...

Escape: None
 Comment: Default
 NA: Default

Code Preview:

```
library(readr)
persons <- read_csv("persons.csv")
View(persons)
```

Reading rectangular data using readr

Import Cancel

调整导入数据类型

导入参数设置

自动代码预览

导入后的变量名

数据集简报

了解数据是入手分析的第一步，**summary**用于呈现各变量统计情况，**str**则继承自列表变量的观察变量结构

```
```{r}  
summary(persons)
```
```

呈现各元素变量基本统计量

```
```{r}  
str(persons)
```
```

呈现列表变量结构

数据集抽样

通常直接观察数据状况是了解数据最直观的方法

```
```{r}  
head(persons)
```
```

`head()`, `tail()`分别取数据前6行和后6行数据

随机抽取对象:

`slice_sample(persons, n=5)`

| X1 <chr> | Sex <chr> | Class <dbl> | Age <dbl> | Math <dbl> | English <fctr> |
|--------------------|---------------------|-----------------------|---------------------|----------------------|--------------------------|
| 李雷 | Male | 1 | 19 | 95 | B |
| 韩... | Female | 1 | 19 | 88 | A |
| 张萌 | Female | 2 | 20 | 72 | B |
| 王珂 | Male | 2 | 19 | 85 | C |
| 刘红 | Female | 1 | 20 | 56 | B |
| 潘迎 | Female | 2 | 20 | 64 | B |

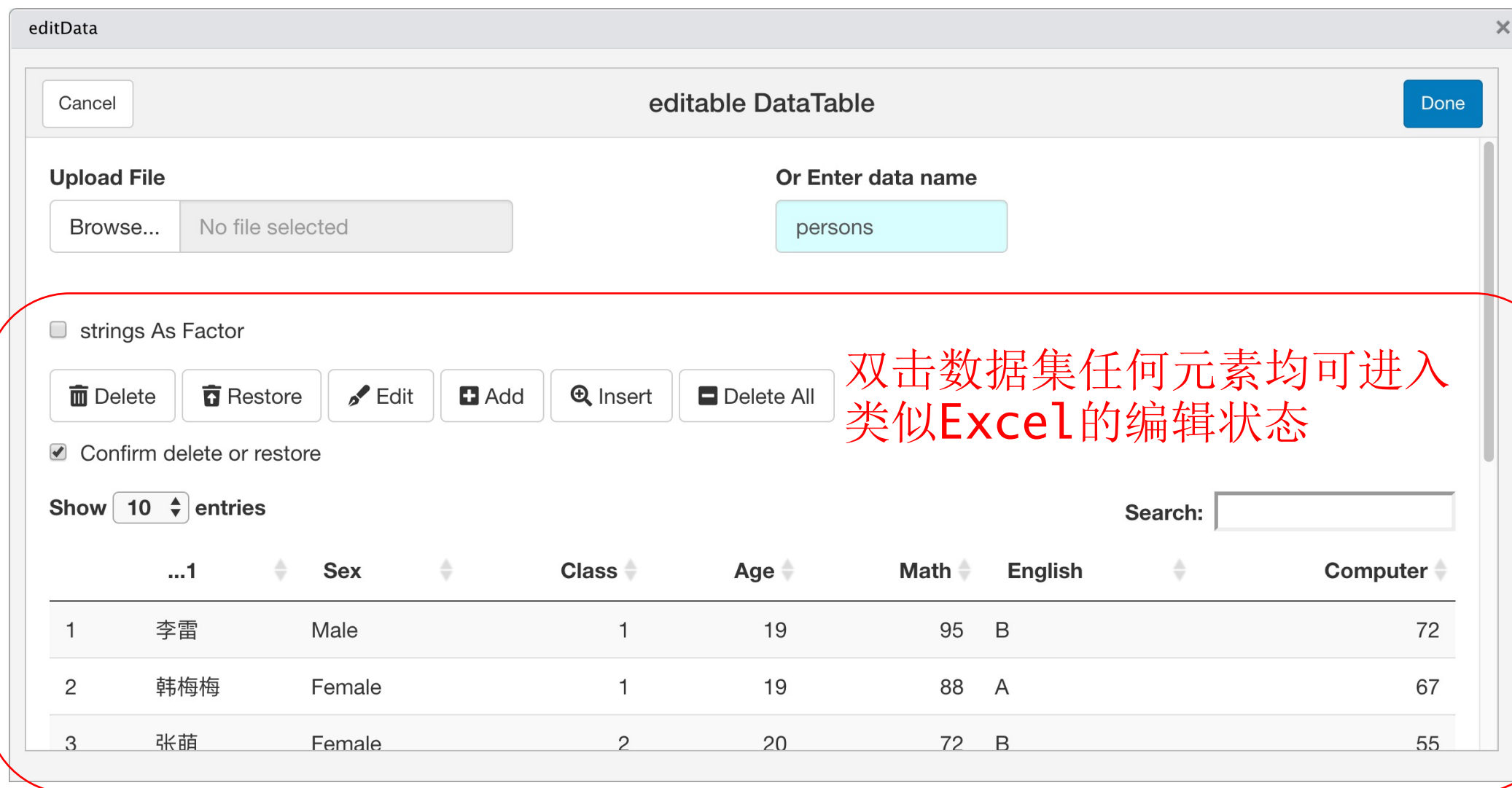
6 rows | 1-6 of 7 columns

`slice_sample`是dplyr包中的函数

数据编辑工具

`editData`以插件形式提供直接修改数据的方式，便于直观处理。

```
> library(editData)
> editData(persons)
```



双击数据集任何元素均可进入类似Excel的编辑状态

但当数据量较大时，手动工具便无法胜任数据处理。

数据探索与准备

重复值处理

空缺检验

补缺处理

2.数据质量与清洗

向量元素的重复

`unique()`函数提取向量中的不重复元素，可以将其理解为集合化。

```
> a=c(1,3,5,21,5,1,20,3,20)
> unique(a)
[1] 1 3 5 21 20
```

利用去重后的结果，自然可以获取每一个元素在数据中分布在哪里

```
> which(a==20)
[1] 7 9
```

对象的重复

数据表格中，通常关注对象是否被重复记录。创建如下表的数据框变量scorelist，duplicated函数检验对象(行数据)是否有重复出现

```
> duplicated(scorelist)
[1] FALSE FALSE FALSE FALSE TRUE TRUE
```

检测结果放入索引则去除重复对象

```
> scorelist[!duplicated(scorelist),]
> distinct(scorelist, Math, .keep_all = T) #dplyr中的函数
```

| Name <chr> | Sex <chr> | Class <int> | Age <int> | Math <int> | English <chr> | Computer <int> |
|---------------|--------------|----------------|--------------|---------------|------------------|-------------------|
| 李雷 | Male | 1 | 19 | 95 | A | 72 |
| 韩梅梅 | Female | 1 | 19 | 95 | A | 67 |
| 张萌 | Female | 2 | 20 | 72 | B | 55 |
| 王珂 | Male | 2 | 19 | 85 | C | 89 |
| 韩梅梅 | Female | 1 | 19 | 95 | A | 67 |
| 王珂 | Male | 2 | 19 | 85 | C | 89 |

变量scorelist, data.frame类型

空缺值的影响和计算

- 计算结果受到影响
- 空缺值符号: NA
- `na.rm`参数排除空缺影响

```
> y=c(1,2,4,6,NA,10)
> y
[1] 1 2 4 6 NA 10
> mean(y)
[1] NA
> log(y)
[1] 0.00 0.69 1.39 1.79 NA 2.30
```

```
> mean(y,na.rm = T)
[1] 4.6
> sum(y,na.rm = T)
[1] 23
> sort(y,na.last = T)
[1] 1 2 4 6 10 NA
```

常见统计函数均有`na.rm`参数，默认为关闭状态

空缺席检测

空缺席检测

- `is.na()` 函数
- 汇总空缺席数量

```
> is.na(y)
[1] FALSE FALSE FALSE FALSE TRUE
FALSE
> any(is.na(y)) #利用判断函数检测
[1] TRUE
> sum(is.na(y)) #逻辑值的整数性
[1] 1
> y[!is.na(y)] #访问非空元素
[1] 1 2 4 6 10
```

替换处理

- 用 `which()` 定位替空值
- 用索引访问和替换空值

```
> which(is.na(y))
[1] 5
> y[is.na(y)]=0
> y
[1] 1 2 4 6 0 10
```

空缺席检测

数据进行计算和分析之前需要检查是否存在空缺席

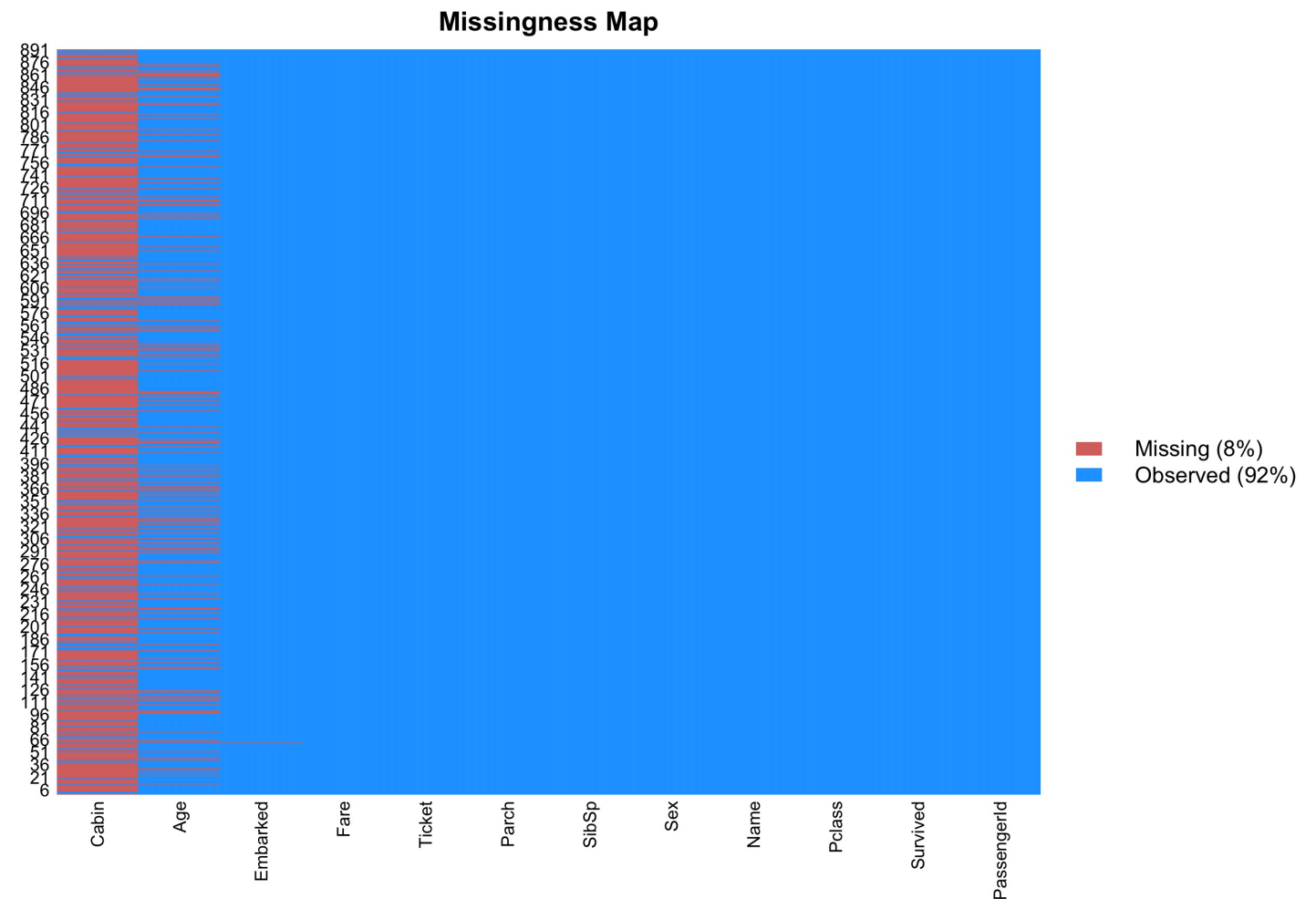
| Filter | | | | | | | | | | | | | |
|--------|-------------|----------|--------|---|--------|-------|-------|-------|------------------|---------|-------|----------|--|
| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked | |
| 1 | 1 | 0 | 3 | Braund, Mr. Owen Harris | male | 22.00 | 1 | 0 | A/5 21171 | 7.2500 | NA | S | |
| 2 | 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Thayer) | female | 38.00 | 1 | 0 | PC 17599 | 71.2833 | C85 | C | |
| 3 | 3 | 1 | 3 | Heikkinen, Miss. Laina | female | 26.00 | 0 | 0 | STON/O2. 3101282 | 7.9250 | NA | S | |
| 4 | 4 | 1 | 1 | Futrelle, Mrs. Jacques Heath (Lily May Peel) | female | 35.00 | 1 | 0 | 113803 | 53.1000 | C123 | S | |
| 5 | 5 | 0 | 3 | Allen, Mr. William Henry | male | 35.00 | 0 | 0 | 373450 | 8.0500 | NA | S | |
| 6 | 6 | 0 | 3 | Moran, Mr. James | male | NA | 0 | 0 | 330877 | 8.4583 | NA | Q | |
| 7 | 7 | 0 | 1 | McCarthy, Mr. Timothy J | male | 54.00 | 0 | 0 | 17463 | 51.8625 | E46 | S | |
| 8 | 8 | 0 | 3 | Palsson, Master. Gosta Leonard | male | 2.00 | 3 | 1 | 349909 | 21.0750 | NA | S | |
| 9 | 9 | 1 | 3 | Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg) | female | 27.00 | 0 | 2 | 347742 | 11.1333 | NA | S | |
| 10 | 10 | 1 | 2 | Nasser, Mrs. Nicholas (Adele Achem) | female | 14.00 | 1 | 0 | 237736 | 30.0708 | NA | C | |
| 11 | 11 | 1 | 3 | Sandstrom, Miss. Marguerite Rut | female | 4.00 | 1 | 1 | PP 9549 | 16.7000 | G6 | S | |
| 12 | 12 | 1 | 1 | Bonnell, Miss. Elizabeth | female | 58.00 | 0 | 0 | 113783 | 26.5500 | C103 | S | |
| 13 | 13 | 0 | 3 | Saunderscock, Mr. William Henry | male | 20.00 | 0 | 0 | A/5. 2151 | 8.0500 | NA | S | |
| 14 | 14 | 0 | 3 | Andersson, Mr. Anders Johan | male | 39.00 | 1 | 5 | 347082 | 31.2750 | NA | S | |
| 15 | 15 | 0 | 3 | Vestrom, Miss. Hulda Amanda Adolfina | female | 14.00 | 0 | 0 | 350406 | 7.8542 | NA | S | |

1912年Titanic号邮轮事件中登船者记录，数据来源Kaggle

大数据集空缺定位

- Amelia工具包
- 图形化标示空缺数据位置

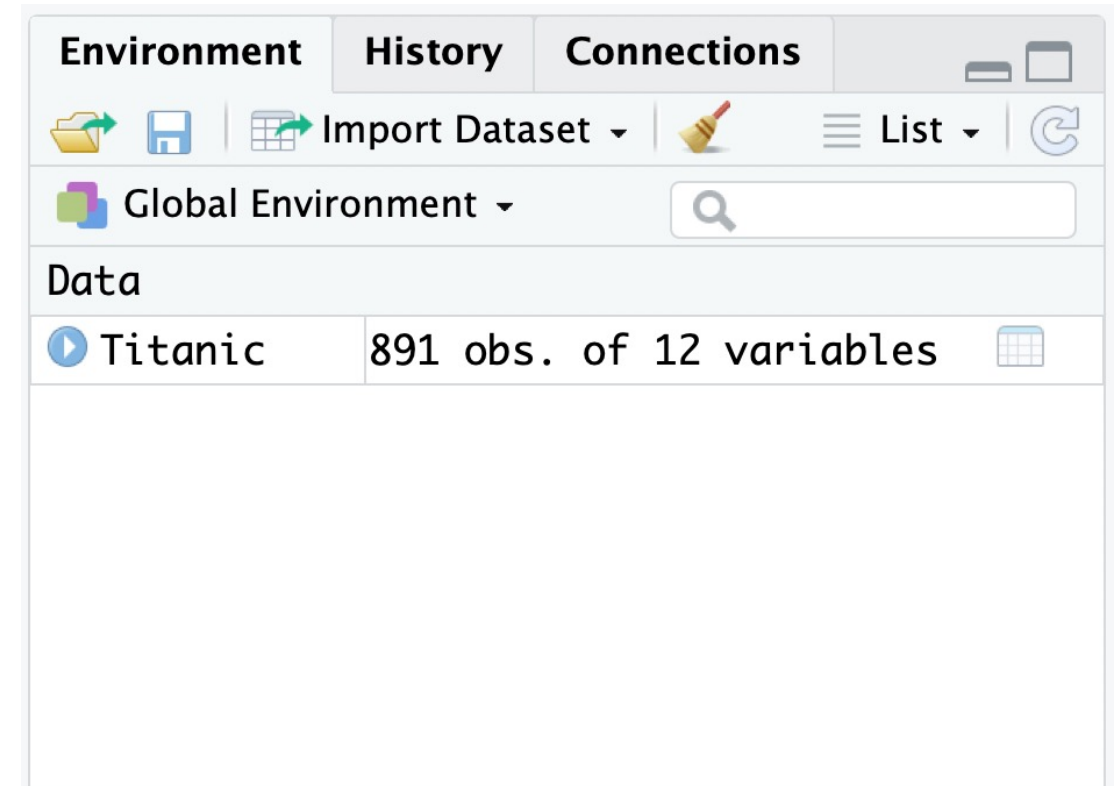
```
> library(Amelia)
> dev.new() #新建图形画板
> missmap(Titanic) #检测数据集
```



数据集空值检验

通常关心数据集空值如下

- 数据集是否完整
 - 哪些对象（行数据）中含空缺
- ```
> anyNA(Titanic) #是否存在空缺
[1] TRUE
```
- ```
> complete.cases(Titanic) #检验行数据是否完整
```
- ```
> sum(complete.cases(Titanic))
[1] 183
```
- ```
> na.omit(Titanic) #检测所有行，含空缺的行将被删除
```
- ```
> drop_na(Titanic, Age, Cabin) #tidyr工具包的函数
```



# 空缺值的删除

- 整列删除：数据严重残缺、对分析影响不大
- 对象删除：删除含空值的行数据

```
> Titanic=subset(Titanic,select = -Cabin)
```

```
> na.omit(Titanic)
```

```
A tibble: 712 x 11
```

|    | PassengerId | Survived | Pclass | Name        | Sex     | Age   | SibSp | Parch | Ticket   | Fare  | Embarked |
|----|-------------|----------|--------|-------------|---------|-------|-------|-------|----------|-------|----------|
|    | <dbl>       | <dbl>    | <dbl>  | <chr>       | <chr>   | <dbl> | <dbl> | <dbl> | <chr>    | <dbl> | <chr>    |
| 1  | 1           | 0        | 3      | Braund, ... | male    | 22    | 1     | 0     | A/5 2... | 7.25  | S        |
| 2  | 2           | 1        | 1      | Cumings,... | fema... | 38    | 1     | 0     | PC 17... | 71.3  | C        |
| 3  | 3           | 1        | 3      | Heikkine... | fema... | 26    | 0     | 0     | STON/... | 7.92  | S        |
| 4  | 4           | 1        | 1      | Futrelle... | fema... | 35    | 1     | 0     | 113803   | 53.1  | S        |
| 5  | 5           | 0        | 3      | Allen, M... | male    | 35    | 0     | 0     | 373450   | 8.05  | S        |
| 6  | 7           | 0        | 1      | McCarthy... | male    | 54    | 0     | 0     | 17463    | 51.9  | S        |
| 7  | 8           | 0        | 3      | Palsson,... | male    | 2     | 3     | 1     | 349909   | 21.1  | S        |
| 8  | 9           | 1        | 3      | Johnson,... | fema... | 27    | 0     | 2     | 347742   | 11.1  | S        |
| 9  | 10          | 1        | 2      | Nasser, ... | fema... | 14    | 1     | 0     | 237736   | 30.1  | C        |
| 10 | 11          | 1        | 3      | Sandstro... | fema... | 4     | 1     | 1     | PP 95... | 16.7  | S        |

```
... with 702 more rows
```

# 空缺值的补缺

在不产生较大误差情况下，补缺思路

- 指定补缺数值
- 以均值/中位数作为补缺

```
a=c(11,5,12,NA,18,9,20,NA,15,22)
impute(a,mean)
impute(a,median)
impute(a,5)
impute(a,c(-1,-1))
a[which(is.na(a))]=c(3,6)
```



# 规律性补缺

部分数据集在创建时由于格式或书写习惯造成规律性空缺值，这类补缺可以根据空值附近数值进行填补。

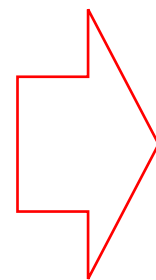
```
library(tidyr)
```

```
fill(sales, year, .direction = "down")
```

`.direction`参数指定补缺方向

此外，`fill`函数可以与`dplyr`工具包的分组`group_by`组合使用

| quarter<br><chr> | year<br><dbl> | sales<br><dbl> |
|------------------|---------------|----------------|
| Q1               | 2000          | 66013          |
| Q2               | NA            | 69182          |
| Q3               | NA            | 53175          |
| Q4               | NA            | 21001          |
| Q1               | 2001          | 46036          |
| Q2               | NA            | 58842          |
| Q3               | NA            | 44568          |
| Q4               | NA            | 50197          |



| quarter<br><chr> | year<br><dbl> | sales<br><dbl> |
|------------------|---------------|----------------|
| Q1               | 2000          | 66013          |
| Q2               | 2000          | 69182          |
| Q3               | 2000          | 53175          |
| Q4               | 2000          | 21001          |
| Q1               | 2001          | 46036          |
| Q2               | 2001          | 58842          |
| Q3               | 2001          | 44568          |
| Q4               | 2001          | 50197          |

数据探索与准备

---

长型与宽型数据

长宽数据转换

### 3.数据重铸

# 长型数据与款型数据

长型数据是一种堆叠型表格，某个变量可以作为标志值重复出现；

宽型数据则将标志值整合为一个属性。

长型数据

| Date<br><date> | Pay_Type<br><chr> | Amount<br><dbl> |
|----------------|-------------------|-----------------|
| 2018-06-10     | alipay            | 2783.6          |
| 2018-06-10     | wechat            | 1987.7          |
| 2018-06-10     | cash              | 688.9           |
| 2018-06-11     | alipay            | 2588.3          |
| 2018-06-11     | wechat            | 2189.4          |
| 2018-06-11     | cash              | 835.6           |

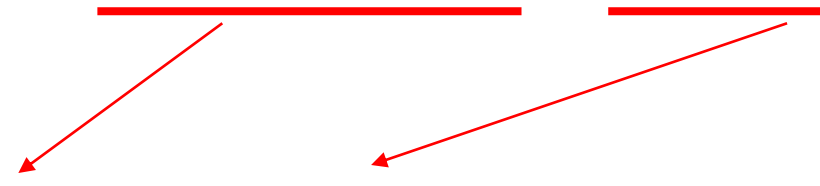
宽型数据

| Date<br><date> | alipay<br><dbl> | wechat<br><dbl> | cash<br><dbl> |
|----------------|-----------------|-----------------|---------------|
| 2018-06-10     | 2783.6          | 1987.7          | 688.9         |
| 2018-06-11     | 2588.3          | 2189.4          | 835.6         |

# 长转宽数据

长转宽：根据指定的标志变量拆分数据，**key**和**value**组成新的一列

`spread(record, key=Pay_Type, value=Amount)`



| Date<br><date> | Pay_Type<br><chr> | Amount<br><dbl> |
|----------------|-------------------|-----------------|
| 2018-06-10     | alipay            | 2783.6          |
| 2018-06-10     | wechat            | 1987.7          |
| 2018-06-10     | cash              | 688.9           |
| 2018-06-11     | alipay            | 2588.3          |
| 2018-06-11     | wechat            | 2189.4          |
| 2018-06-11     | cash              | 835.6           |

| Date<br><date> | alipay<br><dbl> | wechat<br><dbl> | cash<br><dbl> |
|----------------|-----------------|-----------------|---------------|
| 2018-06-10     | 2783.6          | 1987.7          | 688.9         |
| 2018-06-11     | 2588.3          | 2189.4          | 835.6         |

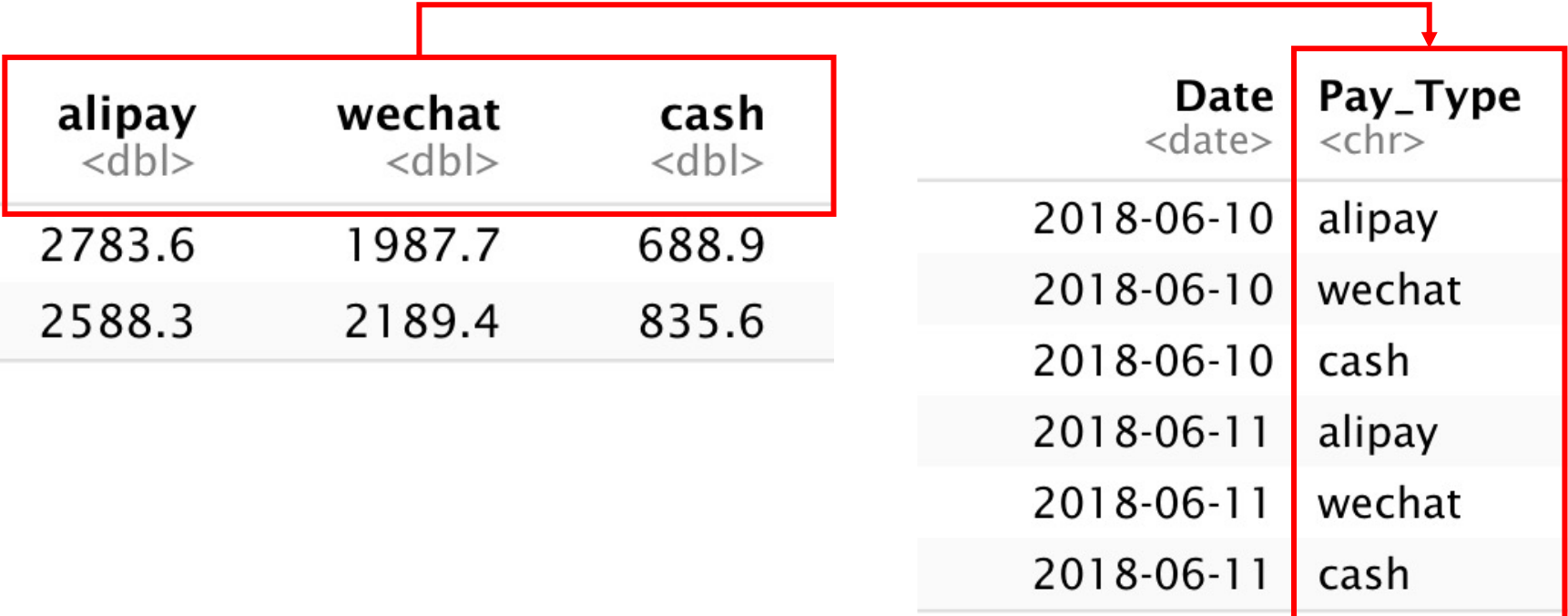
# 宽转长数据

宽转长：将宽数据里指定列作为标签值收集到同一个列(key)，对应数据放入新生成的value列中

与长转宽为互逆操作。

```
gather(record2, key="Pay_Type", value = "amount", -Date)
```

由于Date变量没有参与转换，以-Date方式排出在转换之外。



| Date<br><date> | alipay<br><dbl> | wechat<br><dbl> | cash<br><dbl> |
|----------------|-----------------|-----------------|---------------|
| 2018-06-10     | 2783.6          | 1987.7          | 688.9         |
| 2018-06-11     | 2588.3          | 2189.4          | 835.6         |

| Date<br><date> | Pay_Type<br><chr> | Amount<br><dbl> |
|----------------|-------------------|-----------------|
| 2018-06-10     | alipay            | 2783.6          |
| 2018-06-10     | wechat            | 1987.7          |
| 2018-06-10     | cash              | 688.9           |
| 2018-06-11     | alipay            | 2588.3          |
| 2018-06-11     | wechat            | 2189.4          |
| 2018-06-11     | cash              | 835.6           |

数据探索与准备

---

## 4.大数据集工具

dplyr

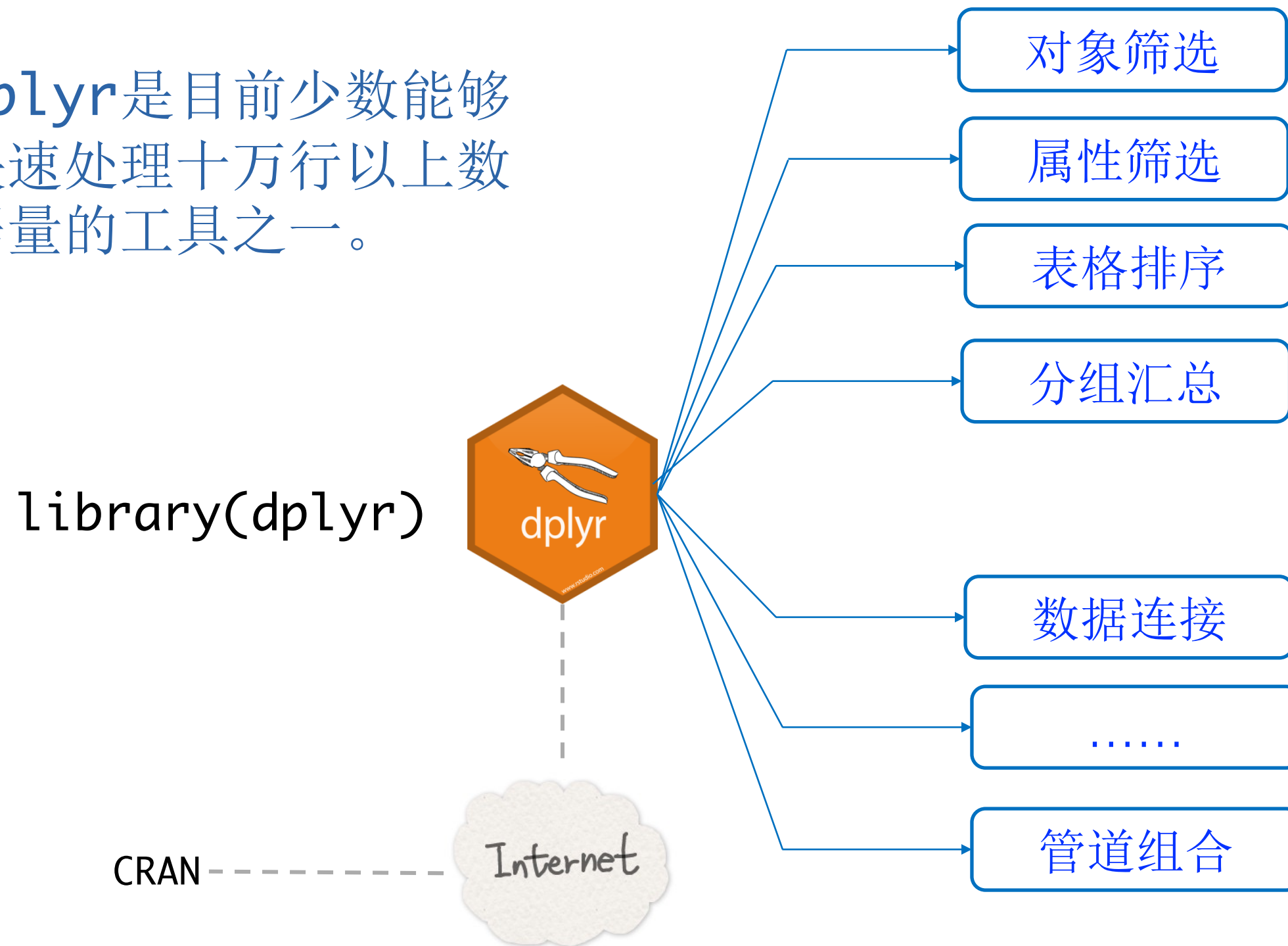
常用工具函数

分组汇总



# 数据框工具dplyr

- dplyr是目前少数能够快速处理十万行以上数据量的工具之一。



# data.frame的扩展类

tibble类型数据创建:

```
```{r}
persons=tibble(
  name=c("Li Lei", "Han Meimei", "Zhang Meng"),
  gender=c("Male", "Female", "Female"),
  age=c(20, 19, 21),
  major=c("Finance", "Statistics", "Economics")
)
```
```

tibble类型对data.frame类增加了显示上的优化, persons 变量同时是tibble类和data.frame类

```
```{r}
class(persons)
```
```

```
[1] "tbl_df" "tbl" "data.frame"
```

# 数据框工具

- 加载dplyr工具包
- filter() 筛选记录

```
> install.packages("dplyr") #首次使用需要安装工具包
> library(dplyr) #加载工具包
> filter(persons, Age>20)
```

|   | Name | Gender | Age | Major     |
|---|------|--------|-----|-----------|
| 1 | 张萌   | Female | 21  | Economics |

## filter ( 筛选对象 )

### • filter() 筛选

问题：筛选年龄大于20岁的人员信息

使用规律：  
功能(数据集, 条件)

```
> persons
```

|   | Name | Gender | Age | Major      |
|---|------|--------|-----|------------|
| 1 | 李雷   | Male   | 20  | Finance    |
| 2 | 韩梅梅  | Female | 19  | Statistics |
| 3 | 张萌   | Female | 21  | Economics  |

```
> filter(persons, Age>20)
```

|   | Name | Gender | Age | Major     |
|---|------|--------|-----|-----------|
| 1 | 张萌   | Female | 21  | Economics |

# filter ( 筛选对象 )

**filter**函数用于按条件筛选对象，类似**subset**函数，用法如下

```
```{r}
filter(persons,English=="B",Math>=80)
```
```

| <b>X1</b><br><chr> | <b>Sex</b><br><chr> | <b>Class</b><br><dbl> | <b>Age</b><br><dbl> | <b>Math</b><br><dbl> | <b>English</b><br><fctr> | <b>Computer</b><br><dbl> |
|--------------------|---------------------|-----------------------|---------------------|----------------------|--------------------------|--------------------------|
| 李雷                 | Male                | 1                     | 19                  | 95                   | B                        | 72                       |
| 熊...               | Female              | 2                     | 20                  | 87                   | B                        | 45                       |
| 王...               | Female              | 2                     | 20                  | 82                   | B                        | 83                       |
| 段程                 | Male                | 2                     | 18                  | 84                   | B                        | 71                       |
| 吴...               | Female              | 2                     | 19                  | 91                   | B                        | 80                       |

5 rows

注意：**filter**中的条件默认靠“与”运算连接，与逗号同义，如用“或”连接则需用逻辑运算 “|”

## select ( 筛选变量 )

### • select() 选择变量

如右侧示例数据

问题1: 取Gender到Major之间的所有属性数据

问题2: 除了姓名属性Name以外, 其他属性提取出来

```
> persons
```

|   | Name | Gender | Age | Major      |
|---|------|--------|-----|------------|
| 1 | 李雷   | Male   | 20  | Finance    |
| 2 | 韩梅梅  | Female | 19  | Statistics |
| 3 | 张萌   | Female | 21  | Economics  |

```
> select(persons, Gender:Major)
```

|   | Gender | Age | Major      |
|---|--------|-----|------------|
| 1 | Male   | 20  | Finance    |
| 2 | Female | 19  | Statistics |
| 3 | Female | 21  | Economics  |

```
> select(persons, -Name)
```

|   | Gender | Age | Major      |
|---|--------|-----|------------|
| 1 | Male   | 20  | Finance    |
| 2 | Female | 19  | Statistics |
| 3 | Female | 21  | Economics  |



## select ( 筛选变量 )

**select**功能类似**subset**函数的列参数，但更加灵活使用。

选择**Age**到**Computer**四列变量，右侧四种方式都是可行并且等效的

```
```{r}
select(persons, Age, Math, English, Computer)
```
```

```
```{r}
select(persons, Age:Computer)
```
```

```
```{r}
select(persons, -(name:Class))
```
```

```
```{r}
select(persons, -name, -Sex, -Class)
```
```

## select ( 筛选变量 )

当数据集变量特别多时，挑选变量称为一件困难事情，**select**支持变量名的字符匹配方式挑选变量，如下常用的3个辅助函数

- `starts_with("abc")`: 匹配以“abc”开头的名称。
- `ends_with("xyz")`: 匹配以“xyz”结尾的名称。
- `contains("ijk")`: 匹配包含“ijk”的名称。

```
```{r}  
select(persons, starts_with("C"))  
```
```

| Class<br><dbl> | Computer<br><dbl> |
|----------------|-------------------|
| 1              | 72                |
| 1              | 67                |
| 2              | 55                |
| 2              | 89                |
| 1              | 75                |
| 2              | 67                |
| 1              | 90                |
| 2              | 81                |
| 2              | 45                |
| 1              | 78                |

## arrange ( 对象排序 )

`arrange()`函数用来给对象排序，类似于`sort`函数。但它的排序条件组合方式是主次指标方式。

```
```{r}
arrange(persons, English, Math)
```
```

| <b>X1</b><br><chr> | <b>Sex</b><br><chr> | <b>Class</b><br><dbl> | <b>Age</b><br><dbl> | <b>Math</b><br><dbl> | <b>English</b><br><fctr> | <b>Computer</b><br><dbl> |
|--------------------|---------------------|-----------------------|---------------------|----------------------|--------------------------|--------------------------|
| 刘璐                 | Female              | 1                     | 18                  | 76                   | A                        | 72                       |
| 韩...               | Female              | 1                     | 19                  | 88                   | A                        | 67                       |
| 刘红                 | Female              | 1                     | 20                  | 56                   | B                        | 75                       |
| 潘迎                 | Female              | 2                     | 20                  | 64                   | B                        | 67                       |
| 张萌                 | Female              | 2                     | 20                  | 72                   | B                        | 55                       |
| 王...               | Female              | 2                     | 20                  | 82                   | B                        | 83                       |
| 段程                 | Male                | 2                     | 18                  | 84                   | B                        | 71                       |

如令排序变量逆序，则对变量使用`desc()`辅助函数

## arrange ( 对象排序 )

- `arrange()` 对象排序

问题：对学生数据集按照年龄`Age`属性排序

- `desc()`函数用于逆转排序方式，仅能用于参数内。

```
> arrange(persons, Age)
```

|   |     |        |    |            |
|---|-----|--------|----|------------|
| 1 | 韩梅梅 | Female | 19 | Statistics |
| 2 | 李雷  | Male   | 20 | Finance    |
| 3 | 张萌  | Female | 21 | Economics  |

```
> arrange(persons, desc(Age))
```

|   | Name | Gender | Age | Major      |
|---|------|--------|-----|------------|
| 1 | 张萌   | Female | 21  | Economics  |
| 2 | 李雷   | Male   | 20  | Finance    |
| 3 | 韩梅梅  | Female | 19  | Statistics |

# rename ( 变量改名 )

**rename**用于重新命名变量名，使用格式如右侧图示：

导入变量时第一列由于无列名称，被默认命名为**X1**，修改列名称为**name**

```
```{r}``
(persons <- rename(persons,name=X1))
```
```

| name<br><chr> | Sex<br><chr> | Class<br><dbl> | Age<br><dbl> | Math<br><dbl> | English<br><fctr> | Computer<br><dbl> |
|---------------|--------------|----------------|--------------|---------------|-------------------|-------------------|
| 李雷            | Male         | 1              | 19           | 95            | B                 | 72                |
| 韩...          | Female       | 1              | 19           | 88            | A                 | 67                |
| 张萌            | Female       | 2              | 20           | 72            | B                 | 55                |
| 王珂            | Male         | 2              | 19           | 85            | C                 | 89                |
| 刘红            | Female       | 1              | 20           | 56            | B                 | 75                |
| 潘迎            | Female       | 2              | 20           | 64            | B                 | 67                |
| 张亮            | Male         | 1              | 18           | 77            | D                 | 90                |
| 卫...          | Male         | 2              | 21           | 34            | C                 | 81                |
| 熊...          | Female       | 2              | 20           | 87            | B                 | 45                |
| 徐...          | Male         | 1              | 19           | 68            | C                 | 78                |

1-10 of 19 rows

Previous 1 2 Next

**tips:** 由于**rename**的结果被赋值回变量**persons**而无法在显示中出现结果，为达到显示效果可在首尾加括弧。

# mutate/transmute ( 变量计算 )

**mutate**将变量按照计算公式生成新变量，并将之置于最后

```
```{r}
mutate(persons, total=Math+Computer)
```
```

| <b>X1</b> | <b>Sex</b> | <b>Class</b> | <b>Age</b> | <b>Math</b> | <b>English</b> | <b>Computer</b> | <b>total</b> |
|-----------|------------|--------------|------------|-------------|----------------|-----------------|--------------|
| <chr>     | <chr>      | <dbl>        | <dbl>      | <dbl>       | <fctr>         | <dbl>           | <dbl>        |
| 李雷        | Male       | 1            | 19         | 95          | B              | 72              | 167          |
| 韩...      | Female     | 1            | 19         | 88          | A              | 67              | 155          |
| 张萌        | Female     | 2            | 20         | 72          | B              | 55              | 127          |
| 王珂        | Male       | 2            | 19         | 85          | C              | 89              | 174          |

若不希望加入原数据表，则用**transmute**函数作为替代

```
```{r}
transmute(persons, total=Math+Computer)
```
```

## summarise ( 汇总处理 )

`summarise`用于生成数据集中变量的汇总统计，类似于`transmute`，但将向量运算改为了统计运算

```
```{r}
summarise(persons, index1=n(), index2=mean(Math))
```
```

| <b>index1</b><br><int> | <b>index2</b><br><dbl> |
|------------------------|------------------------|
| 19                     | 73.15789               |

1 row

辅助函数`n()`返回对象个数

# group\_by ( 分组原理 )

假设数据框**tb**如下

|  | index | score |  |
|--|-------|-------|--|
|  | A     | 80    |  |
|  | B     | 90    |  |
|  | B     | 75    |  |
|  | A     | 83    |  |
|  | A     | 76    |  |
|  | B     | 91    |  |
|  | A     | 88    |  |
|  | B     | 79    |  |

group\_by(tb, index)

按index属性分组



|  | index | score |  |
|--|-------|-------|--|
|  | A     | 80    |  |
|  | B     | 90    |  |
|  | B     | 75    |  |
|  | A     | 83    |  |
|  | A     | 76    |  |
|  | B     | 91    |  |
|  | A     | 88    |  |
|  | B     | 79    |  |

group\_by处理后的数据直观上并无变化，但内在分组逻辑已经存在，之后的汇总运算即按组进行。



## group\_by ( 分组处理 )

**group\_by**用于对数据集插入分组观测变量，之后再汇总时将根据分组标志变量进行统计。如下

```
```{r}
persons2=group_by(persons,Class,Sex)
summarise(persons2,index1=n(),index2=mean(Math))
```
```

| <b>Class</b><br><dbl> | <b>Sex</b><br><chr> | <b>index1</b><br><int> | <b>index2</b><br><dbl> |
|-----------------------|---------------------|------------------------|------------------------|
| 1                     | Female              | 5                      | 69.80000               |
| 1                     | Male                | 6                      | 73.66667               |
| 2                     | Female              | 5                      | 79.20000               |
| 2                     | Male                | 3                      | 67.66667               |

4 rows

# 数据的聚合

数据聚合：分组汇总，整合成新的数据表格。

|  | index | score |  |
|--|-------|-------|--|
|  | A     | 80    |  |
|  | B     | 90    |  |
|  | B     | 75    |  |
|  | A     | 83    |  |
|  | A     | 76    |  |
|  | B     | 91    |  |
|  | A     | 88    |  |
|  | B     | 79    |  |

分组汇总后整合

|  | index | average     |  |
|--|-------|-------------|--|
|  | A     | mean(score) |  |
|  | B     | mean(score) |  |

```
```{r}
step1=filter(persons,Math>60)
step2=group_by(step1,Class,Sex)
result=summarise(step2,
                  number=n(),
                  average=mean(Computer))
```
```

# 管道符 %>%

**tidyverse**集合所有函数均有高度一致的格式，即：函数名(数据集，条件1，条件2.....)

管道符：%>% 将上一个运算结果导入下一个函数的首参数位。

筛选**filter**

然后

分组**group\_by**

然后

汇总**summarise**

```
```{r}
persons%>%
  filter(Math>=60)%>%
  group_by(Class,Sex)%>%
  summarise(number=n(),average=mean(Computer))
```
```

| <b>Class</b><br><dbl> | <b>Sex</b><br><chr> | <b>number</b><br><int> | <b>average</b><br><dbl> |
|-----------------------|---------------------|------------------------|-------------------------|
| 1                     | Female              | 3                      | 74.66667                |
| 1                     | Male                | 5                      | 79.00000                |
| 2                     | Female              | 5                      | 66.00000                |
| 2                     | Male                | 2                      | 80.00000                |

4 rows

%>%位于**magrittr**包中(**dplyr**自动载入)

# 引导管道传输

**aggregate**做聚合处理与**dplyr**非常类似，但数据集放在后边

```
```{r}
persons %>%
  aggregate(Math~Class,.,mean)
```
```

| <b>Class</b><br><fctr> | <b>Math</b><br><dbl> |
|------------------------|----------------------|
| 1                      | 71.90909             |
| 2                      | 74.87500             |

公式符~ 左侧为操作变量，右侧为分组控制变量，通过+连接多个控制变量，之后为计算函数（作为参数调用的函数不带括号）

**注意：**由于**aggregate**第一参数位并非数据集，点符号“.” 引导管道%>%将数据放入指定位置。

```
```{r}
persons %>%
  aggregate(Math~Class+Sex,.,mean)
```
```

# 练习数据集

- 数据集Baseball: vcd 工具包, 322名球员的数据, 哪类球员对比赛最重要?
- 数据集mtcars: 基础包datasets, 记录了多种车型的属性参数, 运用数据框操作工具快速了解不同车型的特点。
- 数据集diamonds: ggplot2工具包, 记录了钻石加工中切割方法cut和钻石长宽高以及价格等属性, 请根据cut属性做聚合, 统计各类钻石加工品的平均价格。
- 数据集sleep,mice工具包, 含空缺值数据集, 医学临床观察数据。
- 数据集BostonHousing,mlbench工具包, 记录美国波士顿市房价及相关数据。
- 数据集NHANES, NHANES工具包, 美国医疗系统的教学数据, 扣除住院病人意外的10000人健康调查数据。

## 练习：长宽转换

某成绩单记录两门课6名学生成绩，数据如下表，请对该表所有科目进行长宽数据转换。

| Semester<br><chr> | id<br><int> | Chinese<br><int> | Math<br><int> | Economics<br><int> |
|-------------------|-------------|------------------|---------------|--------------------|
| 2016学年            | 1           | 73               | 67            | 83                 |
| 2017学年            | 2           | 85               | 79            | 62                 |
| 2016学年            | 3           | 69               | 89            | 55                 |
| 2017学年            | 4           | 82               | 68            | 82                 |
| 2016学年            | 5           | 89               | 87            | 53                 |
| 2017学年            | 6           | 81               | 73            | 74                 |

### 数据创建的代码

```
score_width=data.frame(Semester = c("2016学年","2017学年","2016学
年","2017学年","2016学年","2017学年"),id = c(1L, 2L, 3L, 4L, 5L, 6L),
Chinese = c(73L, 85L, 69L, 82L, 89L, 81L), Math = c(67L, 79L, 89L,
68L, 87L, 73L), Economics = c(83L, 62L, 55L, 82L, 53L, 74L))
```

# 练习

创建如下数据框变量，完成如下练习要求

- 数据集中是否存在重复记录，请去除重复
- 该数据表格涉及多少位学生的记录，年龄为多大
- 若只关注Math和English两科，请去除这两个变量存在重复的行数据。

| Name<br><chr> | Sex<br><chr> | Class<br><int> | Age<br><int> | Math<br><int> | English<br><chr> | Computer<br><int> |
|---------------|--------------|----------------|--------------|---------------|------------------|-------------------|
| 李雷            | Male         | 1              | 19           | 95            | A                | 72                |
| 韩梅梅           | Female       | 1              | 19           | 95            | A                | 67                |
| 张萌            | Female       | 2              | 20           | 72            | B                | 55                |
| 王珂            | Male         | 2              | 19           | 85            | C                | 89                |
| 韩梅梅           | Female       | 1              | 19           | 95            | A                | 67                |
| 王珂            | Male         | 2              | 19           | 85            | C                | 89                |

# 练习

数据集 `squirrels` 记录了3组成员的数据，请观察数据集，采用合适的方法对数据进行补缺处理。




```
squirrels <- tibble::tribble(
 ~group, ~name, ~role, ~n_squirrels,
 1, "Sam", "Observer", NA,
 1, "Mara", "Scorekeeper", 8,
 1, "Jesse", "Observer", NA,
 1, "Tom", "Observer", NA,
 2, "Mike", "Observer", NA,
 2, "Rachael", "Observer", NA,
 2, "Sydekea", "Scorekeeper", 14,
 2, "Gabriela", "Observer", NA,
 3, "Derrick", "Observer", NA,
 3, "Kara", "Scorekeeper", 9,
 3, "Emily", "Observer", NA,
 3, "Danielle", "Observer", NA)
```



# 练习

**titanic**数据集记录了1912年泰坦尼克号沉船中的登船旅客信息，该数据集被拆分为三部分，分别存储在三个**csv**文件中。

- 请导入三个数据集，将三个数据集合并成完整记录表格，并分析数据的结构，数据集共多少变量和对象。
- 检验数据中空缺值分布情况，哪些变量空缺最多，对数据进行适当处理以降低空值的影响。
- **Pclass**是仓位等级，分析不同仓位人数分布情况。
- 若按性别分类，登船乘客男女平均年龄各是多少。

| 名称                                                                                                        | 修改日期              | 大小    | 种类     |
|-----------------------------------------------------------------------------------------------------------|-------------------|-------|--------|
|  gender_submission.csv | 2018年12月12日 12:00 | 3 KB  | CSV 文稿 |
|  Titanic.csv           | 2018年9月22日 23:45  | 61 KB | CSV 文稿 |
|  Titanictest.csv       | 2018年12月12日 11:56 | 29 KB | CSV 文稿 |