#R语言Data Frame数据框常用操作

推荐使用dplyr进行数据预处理；

Data Frame一般被翻译为数据框，感觉就像是R中的表，由行和列组成，与Matrix不同的是，每个列可以是不同的数据类型，而Matrix是必须相同的。

Data Frame每一列有列名，每一行也可以指定行名。如果不指定行名，那么就是从1开始自增的Sequence来标识每一行。
初始化

使用data.frame函数就可以初始化一个Data Frame。比如我们要初始化一个student的Data Frame其中包含ID和Name还有Gender以及Birthdate，那么代码为：

```
student <- data.frame(ID = c(11,12,13),Name =c('Devin','Edward','Wenli'),
            Gender=c('M','M','F'),Birthdate=c('1984-12-29','1983-5-6','1986-8-8'))
```

另外也可以使用read.table() read.csv()读取一个文本文件，返回的也是一个Data Frame对象。读取数据库也是返回Data Frame对象。
查看student的内容为：

```
> student
  ID   Name Gender  Birthdate
1 11  Devin      M 1984-12-29
2 12 Edward      M   1983-5-6
3 13  Wenli      F   1986-8-8
```

这里只指定了列名为ID，Name，Gender和Birthdate，使用names函数可以查看列名，如果要查看行名，需要用到row.names函数。这里我们希望将ID作为行名，那么可以这样写

```
## 查看列名、行名
> names(student) # 查看列名
[1] "ID"        "Name"      "Gender"    "Birthdate"
> colnames(student) # 查看列名
[1] "ID"        "Name"      "Gender"    "Birthdate"
> row.names(student) #查看行名
[1] "1" "2" "3"
```

修改列名，变量的重命名

- 方法1 ： 推荐使用reshape

```
> #install.packages('reshape')
> library(reshape)
> rename(student,c(ID ='reID',Name = 'reName')) #重命名
  reID reName Gender  Birthdate
1   11  Devin      M 1984-12-29
2   12 Edward      M   1983-5-6
3   13  Wenli      F   1986-8-8
```

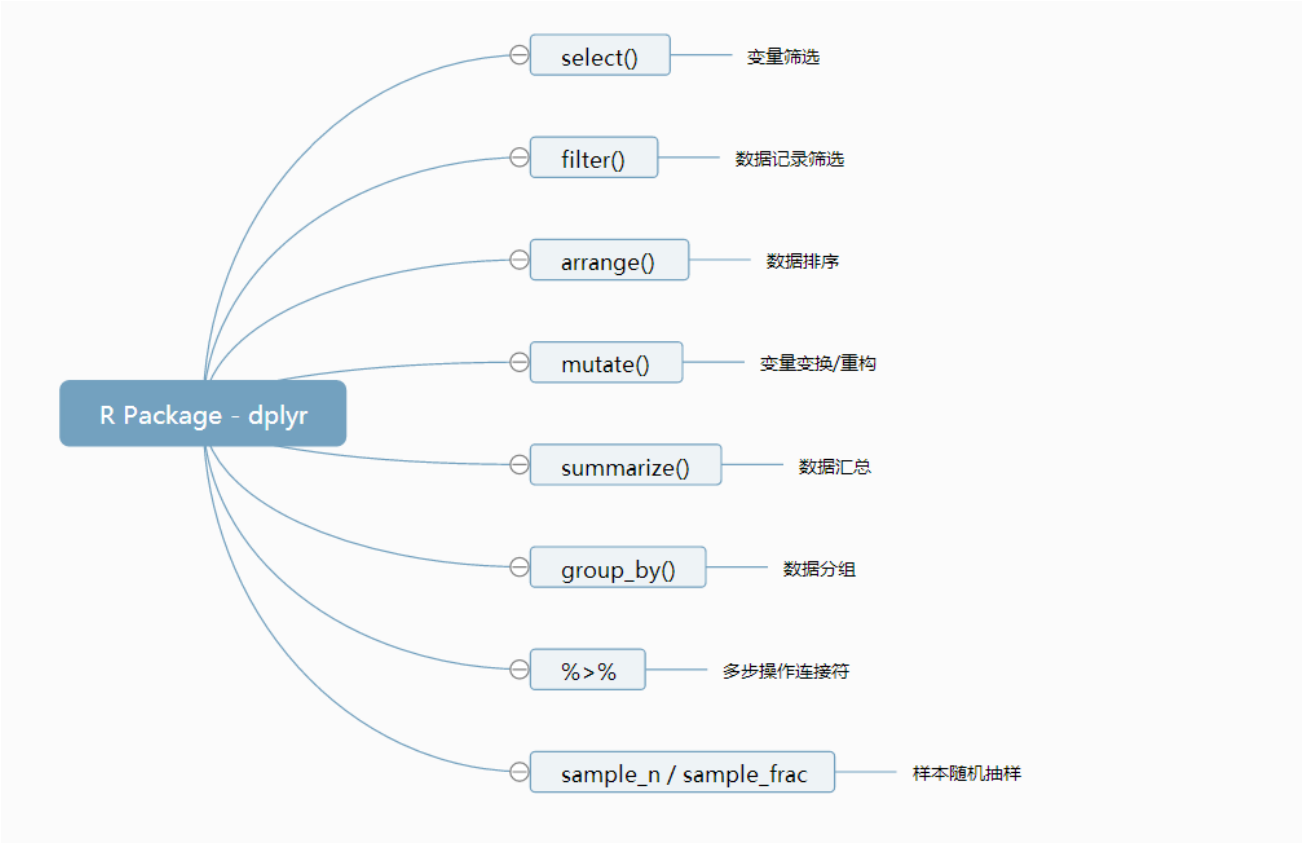
注意：1.修改的参数要放到一个向量中，格式为 (ID ='reID',Name = 'reName')

- 方法二：names取列名索引，修改

```
names[student][1] <- 'textID'
```

变量的筛选、选择、保留、删除(元素的选择)

推荐使用**dplyr**包，更易数据操作与清洗



单表操作函数（one table verbs）如下：
filter: 保留满足条件的行
select: 使用列名选出列
arrange: 对数据的所有行排序
mutate: 添加新的变量
summarise: 分类汇总

dplyr参考教程

原生态数据选择，元素选择

- 方法一：索引选择

```
> student #查看数据集
  ID   Name Gender  Birthdate
1 11  Devin      M 1984-12-29
2 12 Edward      M   1983-5-6
3 13  Wenli      F   1986-8-8
> student[1,] #访问第一行
  ID  Name Gender  Birthdate
1 11 Devin      M 1984-12-29
> student[,1] #访问第一列
[1] 11 12 13
```

数据集取子集、选入(保留、丢弃、删除)变量

选择一列数据

> #如果只访问一列，返回的是Vector类型的，那么可以使用[[或者$来访问。

推荐使用$,直观更容易理解

```
> #比如我们要所有student的Name，代码为：
** 方法一：索引式选择
> student[[2]] #选择student的Name
[1] Devin  Edward Wenli
Levels: Devin Edward Wenli
**方法二：索引式选择
> student[['Name']] #选择student的Name
[1] Devin  Edward Wenli
Levels: Devin Edward Wenli
**方法三：变量名式选择
> student$Name  #选择student的Name
[1] Devin  Edward Wenli
Levels: Devin Edward Wenli
```

选择多列数据、变量

推荐使用student[,c('ID','Name')]，将行下标留空，表示默认选择所有行,直观易理解

```
#选取student表中列名是ID和Name所有数据
**方法一：索引式选择
> student[1:2] ##选择第一列和第二列
  ID   Name
1 11  Devin
2 12 Edward
3 13  Wenli
**
> student[,1:2] ##选择第一列和第二列,将行下标留空，表示默认选择所有行
  ID   Name
1 11  Devin
2 12 Edward
3 13  Wenli
> #选择多列，逗号添加前后效果一样，建议添加逗号
**方法二：  变量名式选择
> student[c('ID','Name')] ##选择第一列和第二列
  ID   Name
1 11  Devin
2 12 Edward
3 13  Wenli
**
> student[,c('ID','Name')] ##选择第一列和第二列,将行下标留空，表示默认选择所有行
  ID   Name
1 11  Devin
2 12 Edward
3 13  Wenli
> #选择多列，逗号添加前后效果一样，建议添加逗号
```

- 推荐使用student[,c('ID','Name')] ##选择第一列和第二列,将行下标留空，表示默认选择所有行

剔除(删除)丢失变量

```
#方法一：  赋值为NULL
student$Gender <- NULL
#方法二:
myvars <- names(student) %in% c('Gender','Birthdate')
newdata <- student[!myvars]
```

丢弃变量是保留变量的逆向操作，选择哪一种方式进行变量的筛选 依赖于两种方式编码难易程度。
如果需要很多变量丢弃，那么直接保留需要留下的变量可能更简单，反之亦然；

数据记录、观测的筛选

```
> student[1:2,] #选择前两行
  ID   Name Gender  Birthdate
1 11  Devin      M 1984-12-29
2 12 Edward      M   1983-5-6
> #选择姓名是F的数据
> student[which(student$Gender == 'F'),]
  ID  Name Gender Birthdate
3 13 Wenli      F  1986-8-8
> #在以上示例中，选择了行下标，并将列下标留空，故默认选择所有的列；
```

修改列数据类型

```
> str(student) #查看每列数据类型
'data.frame':   3 obs. of  4 variables:
 $ ID       : num  11 12 13
 $ Name     : Factor w/ 3 levels "Devin","Edward",..: 1 2 3
 $ Gender   : Factor w/ 2 levels "F","M": 2 2 1
 $ Birthdate: Factor w/ 3 levels "1983-5-6","1984-12-29",..: 2 1 3
```

默认情况下，字符串向量都会被自动识别成Factor，也就是说，ID是数字类型，其他的3个列都被定义为Factor类型了。显然这里Name应该是字符串类型，Birthdate应该是Date类型，我们需要对列的数据类型进行更改：

```
student$Name<-as.character(student$Name)
student$Birthdate<-as.Date(student$Birthdate)
```

添加新列

对于以及存在的student对象，我们希望增加Age列，该列是根据Birthdate算出来的。首先需要知道怎么算年龄。我们可以使用日期函数Sys.Date()获得当前的日期，然后使用format函数获得年份，然后用两个年份相减就是年龄。好像R并没有提供几个能用的日期函数，我们只能使用format函数取出年份部分，然后转换为int类型相减

```
student$Age<-as.integer(format(Sys.Date(),'%Y'))-as.integer(format(student$Birthda
te,'%Y'))
```

缺失值处理

数据排序

数据集合并

参考资料：

- R语言Data Frame数据框常用操作
- R学习笔记—4 基本数据管理
- R-base