# R软件在CGED-Q JSL中的运用

## （六）数据集的内外链接

作者：陈俊（硕士研究生，华中师范大学）；康文林（教授，香港科技大学；华中师范大学）

01

数据集内链接
(PersonID)

PART one

02

数据集外链接
(模糊匹配)

PART two

ps

总结

PART ps

01

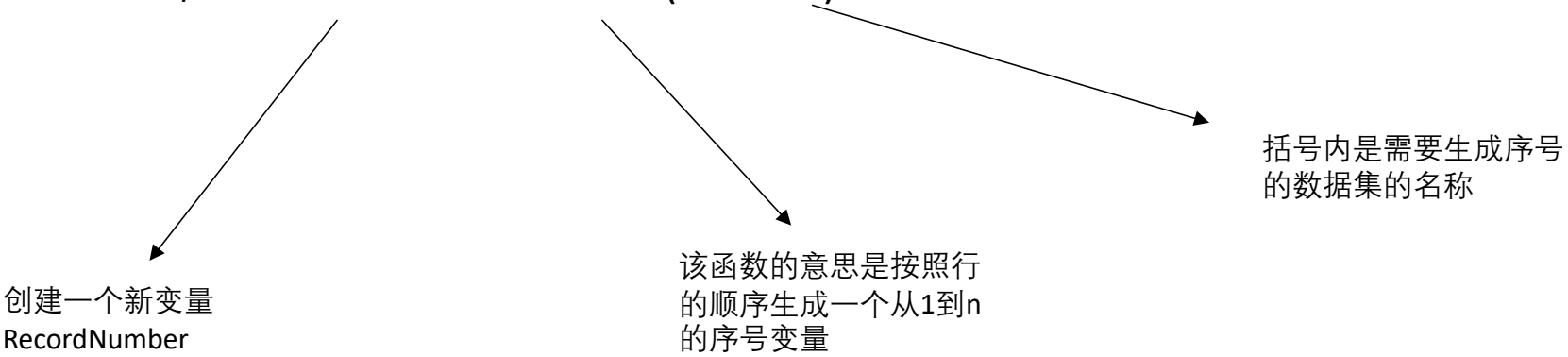
# 数据集内链接
# (PersonID)

# 第一步：创建分析所用数据集、创建一个升序变量

JSL1906 <- subset(JSL1900_1912_delete_kongbaiming,阳历年份 == "1906")

利用subset()函数创建一个数据集，只提取1906年的数据

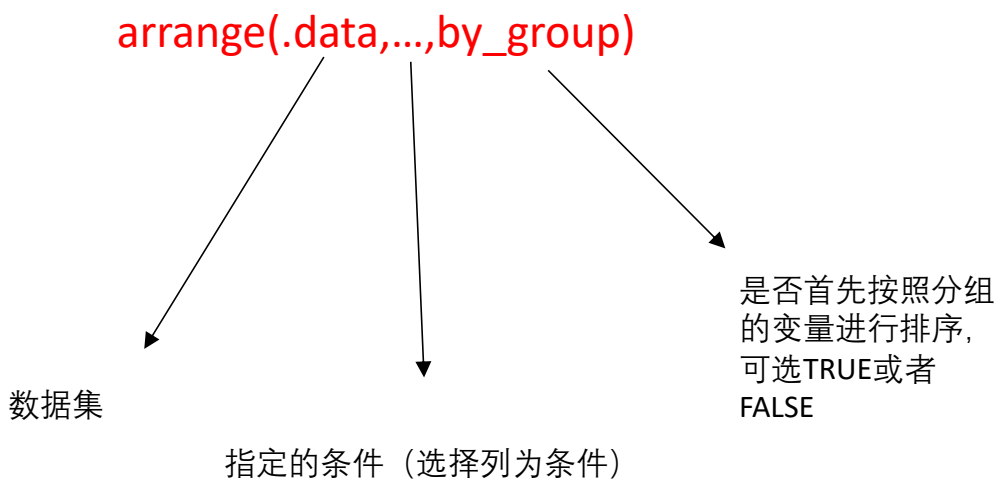
JSL1906$RecordNumber <- 1:nrow(JSL1906)



第二步：按特定条件进行排序

## 排序函数：arrange()函数

arrange()函数，可以将行按指定列的顺序来排序，格式为：



第二步

不考虑by_group，令其等于默认，即FALSE

JSL1906 <- arrange(JSL1906,xinming,身份二,旗分,出身一,年份季节,RecordNumber)

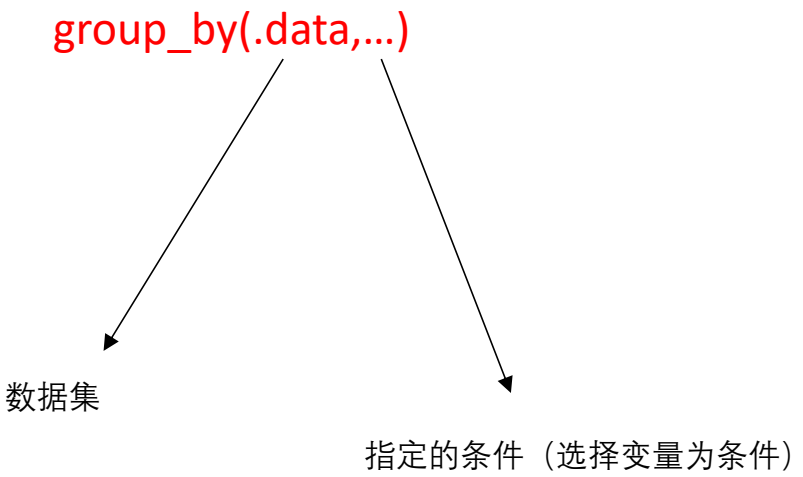
数据集的名称

排序的条件，这里我们选择的首要条件是xingming，然后R会根据姓名进行排序，如果有多条件，那么R会根据条件的先后进行排序，即先考虑xingming，再考虑身份二。把身份二和旗分放在第二和第三是因为有很多旗人是同名的，所以我们利用身份二和旗分变量进行二次甄别。

# 第三步：按特定条件进行分组

## 分组函数：group_by()函数

group_by()函数，可以将行按指定条件进行分组，格式为：



第三步

利用管道符建立组合代码函数

```
JSL1906 <- JSL1906 %>%
    group_by(xinming,旗分) %>%
    mutate(PersonID = ifelse(row_number() == 1, 1, 0))
```

分组函数，将数据集按照xingming和旗分两个变量的顺序进行分组，分组是确定不同官员编号的关键

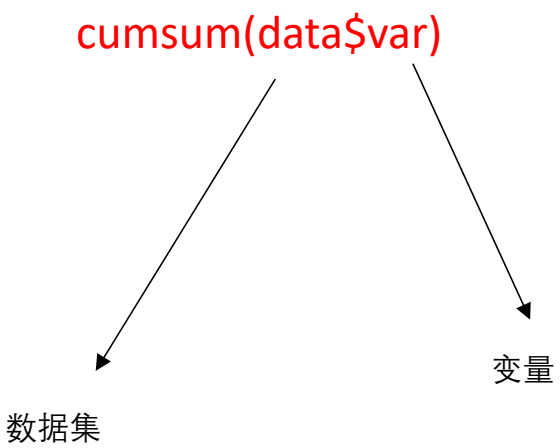
mutate()，是dplyr包下面的一个创建变量的函数。现在利用mutate() 函数创建一个叫PersonID的变量。mutate()里用到了嵌套函数ifelse(),如果某一行它是该分组的第一行，那么R给它赋值1，否则为0。现在数据集中就有了一个只有逻辑值的PersonID，但还没有结束，我们需要继续用到累加函数cumsum()给每个官员一个独一无二的编号。

第四步：对升序变量进行累加

## 累加

## cumsum()函数

cumsum()函数，可以对指定行、分组求累加值，格式为：



第四步

新变量，取名为PersonID_cumsum

JSL1906$PersonID_cumsum <- cumsum(JSL1906$PersonID)

Cumsum()函数，对PersonID求累加值。

数据集的名称

现在，每个官员都拥有了一个独一无二的编号。
利用官员编号可以得到该官员的任职年限

# 第五步：利用累加好的变量创建一个官员任职年限的新变量

利用管道符建立组合代码函数

```
JSL1906 <- JSL1906 %>%
    group_by(PersonID_cumsum) %>%
    mutate(YearsServed = abs((年份季节-年份季节[1])+0.25))
```

分组函数，这次分组采用的是PersonID_cumsum

现在利用mutate() 函数创建一个叫YearsServed的变量。mutate()里用到了嵌套函数abs(),即取绝对值，以防出现有负数的情况。然后用当前行的年份季节减去每个分组的第一个年份季节，再加上0.25，即可求得每个官员的任职年限。为什么要加0.25？因为我们在创造年份季节这个变量的时候为了避免值进1引起混淆减去了一个0.25，这个时候要加回来。

我们还可以利用PersonID将统计模式从记载数转变为人数。用人数来统计有什么好处？假如一个数据集有500000条记录，但可能同一个人就有10条记录，所以整个数据集可能只有50000人。当我们用记载数来统计整个数据库的时候，有很多项目都会出现重复记录的情况，因为一个官员出现了多次。当我们用人数来做统计对象时，满汉比例、出身构成、官员社会和地理来源都会变得非常精确，不会出现重复计算的情况。

```
JSL1906_人数 <- JSL1906 %>% filter( PersonID == 1 )
```

新数据集的名称

filter()函数，筛选出PersonID == 1的行。

现在，我们的数据库里面，每个官员都只有一条记录了。用人数来统计是分析大数据集整体情况的最佳方式。

02

# 数据集外链接
# （模糊匹配）

第一步：创建两个数据集

```
JSL1900_1912_delete_kongbaiming %>%
  filter((阳历年份 == "1906" & 地区 == "四川省" & zhixian == 1)) %>%
  select(阳历年份,地区,机构一,官职一,xinming,字号,籍贯省,籍贯县,旗分,出身一,年份季节) -> JSL1906_sichuan_zhixian
```

```
JSL1900_1912_delete_kongbaiming %>%
  filter((阳历年份 == "1910" & 地区 == "四川省" & zhixian == 1)) %>%
  select(阳历年份,地区,机构一,官职一,xinming,字号,籍贯省,籍贯县,旗分,出身一,年份季节) -> JSL1910_sichuan_zhixian
```

# 第二步：利用函数进行模糊匹配

## 模糊匹配函数：fuzzy_join()函数

fuzzy_join()函数，可对两个数据集进行模糊匹配，格式为：

```
fuzzy_join( x,
            y,
            by = NULL,
            match_fun = NULL,

            multi_by = NULL,
            multi_match_fun = NULL,
            index_match_fun = NULL,
            mode = "inner",
            ...)
```

- x → 数据集
- y → 数据集2
- by = NULL → 根据某个变量进行模糊匹配
- match_fun = NULL → 给定两列的向量化函数，返回true或FALSE以判断它们是否匹配。可以是中指定的每对列的函数列表（如果是命名列表，则使用x中的名称）。如果只给定一个函数，则在所有列对上使用它。
- multi_by = NULL → 要联接的列，其中所有列将用于一起测试匹配项
- multi_match_fun = NULL → 用于测试匹配项的参数，同时对每个数据帧中的所有列执行
- index_match_fun = NULL → 用于匹配表的参数
- mode = "inner" → 匹配的模式，可以选左连接，右连接，外连接，内连接，全连接，需输入对应名称

参考网址：www.idata8.com/rpackage/fuzzyjoin/fuzzy_join.html

第二步

新建数据集的名称

stringdist_inner_join()函数是模糊匹配系列函数中一个较为直接、简单的函数，其用法和merge()函数相似

```
JSL1906_1910_zhixian <- stringdist_inner_join(JSL1906_sichuan_zhixian,
JSL1910_sichuan_zhixian,
by = "xinming",
method = "dl",
max_dist = 1,
distance_col = NULL )
```

数据集1和数据集2

根据某一列进行模糊匹配

计算距离的算法，这里选择的是“dl”算法，可选择的算法包括详见help

链接的最大距离

是否创建包含两个数据集不同点的一个新变量

可以看出，模糊匹配所生成的数据集可能比原始两个数据集都大，因为模糊匹配选择的公共列只有一项，R会将两个数据集中任何满足这一模糊匹配项的行全都找出来。R在寻找这些满足条件的行的时候，采用内置的距离算法，计算出与公共列距离最近的样本量。距离算法只考虑计算出的距离，而不考虑其它因素，其计算过程中只有以给定的少量信息为基准，在最大程度上找到与之相关的样本，因此称为模糊匹配。模糊匹配与精准匹配（我们之前学习的merge函数）的匹配结果会有所不同。但是，我们也发现，模糊匹配和精准匹配其逻辑是相似的，甚至可以说是相同的。那么，在哪些情况下可以使用模糊匹配呢？当两个数据集存在较大的结构差异、或者某个数据集信息量大而另一个信息量少、匹配时缺乏足够的精准匹配信息时，就可以采用模糊匹配的方法。

ps

# 总结

很荣幸能和大家一起学习R语言。到这里，基础数据课已经全部结束了。相信大家已经学到运用R语言处理数据集的一些技巧。

初学R语言，大家肯定会觉得非常难。的确，相对于其它分析软件，R 语言对初学者的数学基础和计算机基础要求更高（R甚至称不上一个分析软件，它实际上是一个语言编程的环境）。利用R，是在创造一种联系，而用分析软件，只是在分析数据。这就是计算机编程语言和计算机软件的本质区别之一。

我们从0开始一起学习R，从一些最基本的操作，逐步延伸到比较高级的操作，我们到目前学习了：

**1.导入和读取文件**-导入数据集（read_excel()、read_dta () 等）、保存文件（save () ）；
**2.创建新变量**-转换变量类型（as.numeric()）、创建新变量（$符号的运用）、逻辑表达式（ifelse()）、数值与字符变量互换（[ ]符号的运用）、串联字符（ paste () ）、提取 判断字符（ str_extract () 、 str_detect () ）、替换字符（ gsub () ）；
**3.制表**-简单制表（table () ）、整理变量（factor () ）、制作可以导出的表格（table1 () ）、指定条件制表（subset ()以及&和|符号的运用）
**4.制图**-简单直方图（ ggplot () 、 geom_bar () ）、进阶直方图（ labs () 、 guides () 、 scale_fill_manual () 、 theme () 、 scale_x_continuous () 、 scale_y_continuous () 、 geom_text () ）、散点图（ geom_point () ）、折线图（ geom_point () +geom_line () ）
**5.dplyr包的简单运用**-转换数据集类型（ tibble::as_tibble () ）、筛选列（ %>% 符号的运用、select () ）、筛选行（filter () ）、匹配数据集以整理变量（merge () ）、连接两个数据集（inner_join () ）
**6.Record linkage**-排序（arrange () ）、分组（group_by () ）、生成新变量（mutate () ）、累加（cumsum () ）、模糊匹配（ stringdist_inner_join () ）、Probabilistic Record Linkages

以上就是基础数据课内容的总结。希望大家在之后的学习生涯中，不断精进自己的数据分析技巧，能够独自处理数据库的诸多问题。

# 基础数据课程到此结束，谢谢大家