数据计算

周世祥

2020 年 5 月 2 日

```
options(knitr.duplicate.label = 'allow')
```

baseR 计算工具

基本的数学运算函数，三角函数等。?S3groupGeneric 或?S4groupGeneric 查看。

```
x <- c(1:10,NA, 11:20)
cumsum(x)
```

```
##  [1]  1  3  6 10 15 21 28 36 45 55 NA NA NA NA NA NA NA NA NA NA NA
```

```
cumsum(x[-which(is.na(x))])#which 定位，-按位置移除对应的值
```

```
##  [1]   1   3   6  10  15  21  28  36  45  55  66  78  91 105 120 136 153 171 190
## [20] 210
```

计算差值函数 diff

```
b <- c(1:3,5,7:11,13)
a <- diff(b) # 向量的各个元素间差值
a
```

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

```
diff(b, lag = 2)
```

```
## [1] 2 3 4 3 2 2 2 3
```

```
length(b)
```

```
## [1] 10
```

```
length(a)
```

```
## [1] 9
```

```
a <- c(NA,diff(b))
a
```

```
##  [1] NA  1  1  2  2  1  1  1  1  2
```

过滤数据框

查询匹配运算符 “%in%”。

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.5
```

```
## Registered S3 methods overwritten by 'tibble':
##   method     from
##   format.tbl pillar
##   print.tbl  pillar
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

```
## Warning: package 'readr' was built under R version 4.0.5
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
## Warning: package 'forcats' was built under R version 4.0.5
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
df <- tibble(a = 1:6,
             b = letters[1:6])
df %>%
  filter( a %in% c(1,3,4))
```

```
## # A tibble: 3 x 2
##         a b
##     <int> <chr>
## 1       1 a
## 2       3 c
## 3       4 d
```

```
df <- tibble(a = 1:6,
             b = letters[1:6])
df %>%
  filter( !a %in% c(1,3,4))
```

```
## # A tibble: 3 x 2
##         a b
##     <int> <chr>
## 1       2 b
## 2       5 e
## 3       6 f
```

基本的统计函数

summary 函数，数据汇总，返回信息太多，干扰分析过程。

统计模型函数，anova 和 lm，公式 formula 的排列顺序，线性回归模型 lm 中的设置格式一般为 y~x，波浪线左侧多位因变量，而右侧为自变量。

dplyr 包

使用频率较高的函数：select，filter，mutate(对列进行增删改写)，arrage，group_by 和 summarise。

行的处理：

数据集 paper_titles 是一个含有 27 行观测值和 3 列变量的字符串型数据框。为新西兰农艺学报 2015-2017 年所发表的期刊名和作者，使用 rds 格式。

arrange 排序，group_by 分组。

dplyr 包中的 arrange 相当于 baseR 中的 order 函数的简化版，参数设置更加简单。默认为升序排列。

```
df <- readRDS("RawData/paper_titles.rds")
head(df,n=3)
```

```
## # A tibble: 3 x 3
##     year  titles                                                  authors
##   <chr> <chr>                                <chr>
## 1 2017  Nitrogen uptake and nitrate-nitroge~ E. Chakwizira, J.M. de Ruiter and ~
## 2 2017  Pasture brome (Bromus valdivianus) ~ I.P. Ordó<U+00F1>ez, I.F. López, P.D. Kem~
## 3 2017  A possible sustainable harvesting r~ W.T. Bussell and C.M. Triggs pp. 2~
```

数值型按从小到大排；字符型按字母表顺序排。

```
df %>%
  group_by(year) %>%  # 将数据集 df 由管道传递给 group_by，并以 year 这一变量对整个数据集进行分组，之后
  arrange(titles, .by_group = TRUE) %>% # 最后通过 str 函数显示分组排序后的数据集
  str()
```

```
## grouped_df [27 x 3] (S3: grouped_df/tbl_df/tbl/data.frame)
##  $ year   : chr [1:27] "2015" "2015" "2015" "2015" ...
##  $ titles : chr [1:27] "Automated measurement of crop water balances under a mobile rain-exclus
##  $ authors: chr [1:27] "A.J. Michel, H.E. Brown, R.N. Gillespie, M.J. George and E.D. Meenken p
##  - attr(*, "groups")= tbl_df [3 x 2] (S3: tbl_df/tbl/data.frame)
##   ..$ year : chr [1:3] "2015" "2016" "2017"
##   ..$ .rows: list<int> [1:3]
##   .. ..$ : int [1:6] 1 2 3 4 5 6
##   .. ..$ : int [1:12] 7 8 9 10 11 12 13 14 15 16 ...
##   .. ..$ : int [1:9] 19 20 21 22 23 24 25 26 27
##   .. ..@ ptype: int(0)
##   ..- attr(*, ".drop")= logi TRUE
```

```
df$year <- as.integer(df$year)
```

```
df %>%
  arrange_if(is.character) %>% # 分组排序前进行条件筛选
  head()
```

```
## # A tibble: 6 x 3
##    year titles                               authors
##   <int> <chr>                                <chr>
## 1  2017 A possible sustainable harvesting r~ W.T. Bussell and C.M. Triggs pp. 2~
## 2  2015 Automated measurement of crop water~ A.J. Michel, H.E. Brown, R.N. Gill~
## 3  2016 Carbohydrate degradation during sam~ C. Matthew, B.W. Howard, A.R. Drys~
## 4  2016 Catch crops after winter grazing fo~ B. Malcolm, E. Teixeira, P. Johnst~
## 5  2015 Comparison of continuous and spot m~ E. Chakwizira, E.D. Meenken, M.J. ~
## 6  2015 Determining sources of variation in~ S.J. Gibbs, S. Hodge, B. Saldias, ~
```

filter 按条件过滤行

```
df <- readRDS("RawData/paper_titles.rds")
head(df)
```

```
## # A tibble: 6 x 3
##   year  titles                               authors
##   <chr> <chr>                                <chr>
## 1 2017  Nitrogen uptake and nitrate-nitroge~ E. Chakwizira, J.M. de Ruiter and ~
## 2 2017  Pasture brome (Bromus valdivianus) ~ I.P. Ordó<U+00F1>ez, I.F. López, P.D. Kem~
## 3 2017  A possible sustainable harvesting r~ W.T. Bussell and C.M. Triggs pp. 2~
## 4 2017  Stem yield response of annual ryegr~ J.W.L. Heney, M.P. Rolston, R.J. C~
## 5 2017  Effect of variable rate lime applic~ A.W. Holmes and G. Jiang pp. 37-45
## 6 2017  Plant density effects on yield para~ L.H.J. Kerckhoffs, S. O’Neill, R. ~
```

```
df %>%
  filter(year == 2016 | year == 2017) %>% # 逻辑或
  head(df ,n=3)
```

```
## # A tibble: 3 x 3
##   year  titles                               authors
##   <chr> <chr>                                <chr>
## 1 2017  Nitrogen uptake and nitrate-nitroge~ E. Chakwizira, J.M. de Ruiter and ~
## 2 2017  Pasture brome (Bromus valdivianus) ~ I.P. Ordó<U+00F1>ez, I.F. López, P.D. Kem~
## 3 2017  A possible sustainable harvesting r~ W.T. Bussell and C.M. Triggs pp. 2~
```

```
df %>%
  filter(year == 2016 & year == 2017)
```

```
## # A tibble: 0 x 3
## # ... with 3 variables: year <chr>, titles <chr>, authors <chr>
```

```
df %>%
  filter(year == 2016 , year == 2017)
```

```
## # A tibble: 0 x 3
## # ... with 3 variables: year <chr>, titles <chr>, authors <chr>
```

```
df %>%
  filter(year < 2017 & year > 2015)
```

```
## # A tibble: 12 x 3
##     year  titles                              authors
##     <chr> <chr>                                <chr>
##   1 2016  Stem shortening plant growth regul~ M.P. Rolston, R.J. Chynoweth, J.A.~
##   2 2016  Time of cutting effects on seed yi~ J.M. Linton, R.J. Chynoweth, M.P. ~
##   3 2016  Carbohydrate degradation during sa~ C. Matthew, B.W. Howard, A.R. Drys~
##   4 2016  Improving yield and quality of pro~ S.J. Dellow, A. Hunt, R.N. Gillesp~
##   5 2016  The effects of maize seed treatmen~ P.S. Oliver, S.A. Harvey and D.M. ~
##   6 2016  Optimising sweet corn plant popula~ A.G. Hunt, J.B. Reid and P.R. John~
##   7 2016  Effect of sowing date on plant cou~ N. Stocker, B. Saldias, R. Brosnah~
##   8 2016  Effects of irrigation regime, bed ~ A.J. Michel, S.M. Sinton, S.J. Del~
##   9 2016  Growth and nitrogen partitioning o~ E. Chakwizira, J.M. de Ruiter and ~
##  10 2016  Catch crops after winter grazing f~ B. Malcolm, E. Teixeira, P. Johnst~
##  11 2016  Effectiveness of winter cover crop~ R.F. Zyskowski, E.I. Teixeira. B.J~
##  12 2016  Nitrogen fertilisation effects on ~ D.F. Guinto<U+00A0>pp. 121-132
```

```
df %>%
  filter(year != 2017) %>%
  head()
```

```
## # A tibble: 6 x 3
##    year  titles                               authors
##    <chr> <chr>                                <chr>
## 1 2016  Stem shortening plant growth regula~ M.P. Rolston, R.J. Chynoweth, J.A.~
## 2 2016  Time of cutting effects on seed yie~ J.M. Linton, R.J. Chynoweth, M.P. ~
## 3 2016  Carbohydrate degradation during sam~ C. Matthew, B.W. Howard, A.R. Drys~
## 4 2016  Improving yield and quality of proc~ S.J. Dellow, A. Hunt, R.N. Gillesp~
## 5 2016  The effects of maize seed treatment~ P.S. Oliver, S.A. Harvey and D.M. ~
## 6 2016  Optimising sweet corn plant populat~ A.G. Hunt, J.B. Reid and P.R. John~
```

```
df %>%
  filter(!year %in% c(2016,2017)) %>%
  head(df,n=3)
```

```
## # A tibble: 3 x 3
##    year  titles                               authors
##    <chr> <chr>                                <chr>
## 1 2015  Grain yield of winter feed wheat in~ R.A. Craigie, H.E. Brown and M. Ge~
## 2 2015  Comparison of continuous and spot m~ E. Chakwizira, E.D. Meenken, M.J. ~
## 3 2015  Managing whole crop cereal silage y~ M.E. Arnaudin, J.M. de Ruiter, S. ~
```

符号函数%in%与filter函数属于天生绝配，假定用户有一组数据集中包含了若干观测值需要排除或包括，那么可以将筛选条件向量化后置于%in%之右，而将需要筛选的列置于其左侧，简化代码，提高效率。若使用filter(year==2015)可能会丢失符合条件的项。

```
df %>%
  filter(titles %in% grep(pattern="^Nitrogen.+", x=.$titles, value = T))
```

```
## # A tibble: 2 x 3
##   year  titles                                     authors
##   <chr> <chr>                                      <chr>
## 1 2017  Nitrogen uptake and nitrate-nitrogen accu~ E. Chakwizira, J.M. de Ruite~
## 2 2016  Nitrogen fertilisation effects on the qua~ D.F. Guinto<U+00A0>pp. 121-132
```

```
# 模式参数: ^Nitrogen.+ 表示以单词 Nitrogen 开头，再加上任一字符
#.$titles, . 号表示管道函数之前的 df 数据框
#value 为真，代表需要函数返回其包含模式的真实字符串值
```

```
df %>%
  filter(is.na(titles))
```

```
## # A tibble: 0 x 3
## # ... with 3 variables: year <chr>, titles <chr>, authors <chr>
```

```
# 默认值 NA 和空白值 NULL 的处理, is.na
```

```
df$year <- as.integer(df$year) # 因为后面的%% 运算不接受字符串类型
df %>%
  filter_at(vars(year), any_vars((. %% 4) == 0))
```

```
## # A tibble: 12 x 3
##     year titles                              authors
##    <int> <chr>                               <chr>
##  1  2016 Stem shortening plant growth regul~ M.P. Rolston, R.J. Chynoweth, J.A.~
##  2  2016 Time of cutting effects on seed yi~ J.M. Linton, R.J. Chynoweth, M.P. ~
##  3  2016 Carbohydrate degradation during sa~ C. Matthew, B.W. Howard, A.R. Drys~
##  4  2016 Improving yield and quality of pro~ S.J. Dellow, A. Hunt, R.N. Gillesp~
##  5  2016 The effects of maize seed treatmen~ P.S. Oliver, S.A. Harvey and D.M. ~
##  6  2016 Optimising sweet corn plant popula~ A.G. Hunt, J.B. Reid and P.R. John~
##  7  2016 Effect of sowing date on plant cou~ N. Stocker, B. Saldias, R. Brosnah~
##  8  2016 Effects of irrigation regime, bed ~ A.J. Michel, S.M. Sinton, S.J. Del~
##  9  2016 Growth and nitrogen partitioning o~ E. Chakwizira, J.M. de Ruiter and ~
## 10  2016 Catch crops after winter grazing f~ B. Malcolm, E. Teixeira, P. Johnst~
## 11  2016 Effectiveness of winter cover crop~ R.F. Zyskowski, E.I. Teixeira. B.J~
## 12  2016 Nitrogen fertilisation effects on ~ D.F. Guinto<U+00A0>pp. 121-132
```

```r
# 闰年的一种表达
```

```r
set.seed(42) # 确保每次随机抽样的样本一致
df %>%
  sample_n(size = 5) # 按用户指定的个数随机抽取行数据，即观测值
```

```
## # A tibble: 5 x 3
##    year titles                               authors
##   <int> <chr>                                <chr>
## 1  2016 Effects of irrigation regime, bed a~ A.J. Michel, S.M. Sinton, S.J. Del~
## 2  2017 Effect of variable rate lime applic~ A.W. Holmes and G. Jiang pp. 37-45
## 3  2017 Nitrogen uptake and nitrate-nitroge~ E. Chakwizira, J.M. de Ruiter and ~
## 4  2016 Stem shortening plant growth regula~ M.P. Rolston, R.J. Chynoweth, J.A.~
## 5  2017 Stem yield response of annual ryegr~ J.W.L. Heney, M.P. Rolston, R.J. C~
```

```r
df %>%
  sample_frac(size = 0.3) # 按比例抽样
```

```
## # A tibble: 8 x 3
##    year titles                               authors
##   <int> <chr>                                <chr>
## 1  2016 Growth and nitrogen partitioning of~ E. Chakwizira, J.M. de Ruiter and ~
## 2  2015 Trends in New Zealand herbage seed ~ R.J. Chynoweth, N.B. Pyke, M.P. Ro~
## 3  2016 Effects of irrigation regime, bed a~ A.J. Michel, S.M. Sinton, S.J. Del~
## 4  2016 Optimising sweet corn plant populat~ A.G. Hunt, J.B. Reid and P.R. John~
## 5  2017 Effect of sowing date on forage rap~ M. Rashid, J.G. Hampton, J.A.K. Tr~
## 6  2017 Stem yield response of annual ryegr~ J.W.L. Heney, M.P. Rolston, R.J. C~
## 7  2017 Effect of variable rate lime applic~ A.W. Holmes and G. Jiang pp. 37-45
## 8  2016 The effects of maize seed treatment~ P.S. Oliver, S.A. Harvey and D.M. ~
```

```r
df %>%
  group_by(year) %>%
  sample_n(size = 2) # 对分组后的数据进行抽样
```

```
## # A tibble: 6 x 3
## # Groups:   year [3]
##    year titles                               authors
##   <int> <chr>                                <chr>
## 1  2015 Automated measurement of crop water~ A.J. Michel, H.E. Brown, R.N. Gill~
## 2  2015 Comparison of continuous and spot m~ E. Chakwizira, E.D. Meenken, M.J. ~
## 3  2016 Time of cutting effects on seed yie~ J.M. Linton, R.J. Chynoweth, M.P. ~
## 4  2016 Carbohydrate degradation during sam~ C. Matthew, B.W. Howard, A.R. Drys~
## 5  2017 Growing edible Taro in Waikato stre~ A. Parshotam, V. Parshotam and O.M~
## 6  2017 Nitrogen uptake and nitrate-nitroge~ E. Chakwizira, J.M. de Ruiter and ~
```

```
df %>%
  group_by(year) %>%
  arrange(titles, .by_group = TRUE) %>%
  filter( titles == first(titles))
```

```
## # A tibble: 3 x 3
## # Groups:   year [3]
##    year titles                                 authors
##   <int> <chr>                                  <chr>
## 1  2015 Automated measurement of crop water~ A.J. Michel, H.E. Brown, R.N. Gill~
## 2  2016 Carbohydrate degradation during sam~ C. Matthew, B.W. Howard, A.R. Drys~
## 3  2017 A possible sustainable harvesting r~ W.T. Bussell and C.M. Triggs pp. 2~
```

```
# 将数据集 df 按 year 进行分组，之后按 titles 进行升序排列，最后抽取每组的第一行观测值

# 配合 first 来抽取指定位置的观测值
```

bind family 强行合并数据集

在 baseR 中，rbind 和 cbind 函数分别用于按行将若干数据集上下对接，或者按列对若干数据集进行左右对接。在 dplyr 包中，相同功能的函数名称为 bind_rows 和 bind_cols。

```
df <- readRDS("RawData/paper_titles.rds")
one <- filter(df, year == 2015) # 按不同的年份拆分数据集
two <- filter(df, year == 2016)
three <- filter(df, year ==2017)
bind_rows(  one,   two, three, .id = "IDs") %>%
  glimpse() # 参数.id 用于在整合后的大数据集中标注每个数据集的来源，其默认为空
```

```
## Rows: 27
## Columns: 4
## $ IDs     <chr> "1", "1", "1", "1", "1", "1", "2", "2", "2", "2", "2", "2", "2~
## $ year    <chr> "2015", "2015", "2015", "2015", "2015", "2015", "2016", "2016"~
## $ titles  <chr> "Grain yield of winter feed wheat in response to sowing date a~
## $ authors <chr> "R.A. Craigie, H.E. Brown and M. George pp. 1-8", "E. Chakwizi~
```

```
df <- readRDS("RawData/paper_titles.rds")
one <- filter(df, year == 2015) # 按不同的年份拆分数据集
two <- filter(df, year == 2016)
```

```
three <- filter(df, year ==2017)
bind_rows(list(a = one, b = two, c = three), .id = "IDs") %>%
  glimpse() # 将需要整合的数据集放入一个或若干个 list 函数之内，相应的标注也改变了
```

```
## Rows: 27
## Columns: 4
## $ IDs     <chr> "a", "a", "a", "a", "a", "a", "b", "b", "b", "b", "b", "b", "b~
## $ year    <chr> "2015", "2015", "2015", "2015", "2015", "2015", "2016", "2016"~
## $ titles  <chr> "Grain yield of winter feed wheat in response to sowing date a~
## $ authors <chr> "R.A. Craigie, H.E. Brown and M. George pp. 1-8", "E. Chakwizi~
```

两个数据集无须具有相同的变量数也可以进行上下对接。不能对接的部分用 NA 填补。

若要按列整合若干数据集，则必须要求各个数据集都要具有相同的行数，否则会报错。

```
bind_cols(one[1:3, ], two[1:3, ], three[1:3, ]) %>% # 先做子集筛选处理，选前三行观测值
  glimpse()
```

```
## New names:
## * year -> year...1
## * titles -> titles...2
## * authors -> authors...3
## * year -> year...4
## * titles -> titles...5
## * ...

## Rows: 3
## Columns: 9
## $ year...1    <chr> "2015", "2015", "2015"
## $ titles...2  <chr> "Grain yield of winter feed wheat in response to sowing da~
## $ authors...3 <chr> "R.A. Craigie, H.E. Brown and M. George pp. 1-8", "E. Chak~
## $ year...4    <chr> "2016", "2016", "2016"
## $ titles...5  <chr> "Stem shortening plant growth regulators enhance seed yiel~
## $ authors...6 <chr> "M.P. Rolston, R.J. Chynoweth, J.A.K.Trethewey, A.J. Hildi~
## $ year...7    <chr> "2017", "2017", "2017"
## $ titles...8  <chr> "Nitrogen uptake and nitrate-nitrogen accumulation in fora~
## $ authors...9 <chr> "E. Chakwizira, J.M. de Ruiter and S. Maley pp. 1-12", "I.~
```

dplyr 对列 column 处理

```
df %>%
  rename("1" = year) %>%  # 重命名列
  glimpse()
```

```
## Rows: 27
## Columns: 3
## $ `1`     <chr> "2017", "2017", "2017", "2017", "2017", "2017", "2017", "2017"~
## $ titles  <chr> "Nitrogen uptake and nitrate-nitrogen accumulation in forage k~
## $ authors <chr> "E. Chakwizira, J.M. de Ruiter and S. Maley pp. 1-12", "I.P. O~
```

rename 和 select 都可以完成对变量列重命名的操作，两者的区别：rename 会将重命名列及其他列同时返回为结果，而 select 仅返回选择的指定列及新列名。select 的效率要高于 rename。

```
df %>%
  select("1" = year) %>%
  glimpse()
```

```
## Rows: 27
## Columns: 1
## $ `1` <chr> "2017", "2017", "2017", "2017", "2017", "2017", "2017", "2017", "2~
```

命名规则为：新列名在等号左侧，数据集原有列名在等号右侧。

```
df %>%
  select(starts_with(match = "y"))%>%
  #SELECT 函数有很多搭配使用的函数
  glimpse()
```

```
## Rows: 27
## Columns: 1
## $ year <chr> "2017", "2017", "2017", "2017", "2017", "2017", "2017", "2017", "~
```

```
# 匹配的字符串，y 开头
```

```
df %>%
  select(ends_with(match = "s"))%>%
  glimpse()
```

```
## Rows: 27
## Columns: 2
## $ titles  <chr> "Nitrogen uptake and nitrate-nitrogen accumulation in forage k~
## $ authors <chr> "E. Chakwizira, J.M. de Ruiter and S. Maley pp. 1-12", "I.P. O~
```

```
# 匹配结尾
```

```
df %>%
  select(matches(match = ".tle.")) %>%
  glimpse()
```

```
## Rows: 27
## Columns: 1
## $ titles <chr> "Nitrogen uptake and nitrate-nitrogen accumulation in forage ka~
```

```
df %>%
  select(contains(match = "ear")) %>%
  glimpse()
```

```
## Rows: 27
## Columns: 1
## $ year <chr> "2017", "2017", "2017", "2017", "2017", "2017", "2017", "2017", "~
```

```
df %>%
  select(one_of( c("year","titles","day"))) %>% # 其中之一
  glimpse()
```

```
## Warning: Unknown columns: `day`
```

```
## Rows: 27
## Columns: 2
## $ year   <chr> "2017", "2017", "2017", "2017", "2017", "2017", "2017", "2017",~
## $ titles <chr> "Nitrogen uptake and nitrate-nitrogen accumulation in forage ka~
```

https://blog.csdn.net/wltom1985/article/details/54973811

mutate：dplyr 包的灵魂函数之一

mutate 是变化的含义，代表着变化。如增删变量，更新变量的值或替换符合标准的值。

星球大战中主要角色名字及其相关信息。选取前 10 行观测值和 4 个比较有代表性的变量。这个数据集是 dplyr 自带的。

```
starwars_short <- starwars %>%
  slice(1:10) %>%
  select(name, height, mass, species)
```

研究这 10 个人物是否有肥胖倾向。否则完不成维护宇宙和平使命。

```
starwars_short %>%
  mutate(height = height/100, # 当等式左侧列名与数据集中已有列名一致时，旧列被新列替换
         BMI = mass/(height^2),
         cumprod(mass))
```

```
## # A tibble: 10 x 6
##    name               height  mass species   BMI `cumprod(mass)`
##    <chr>               <dbl> <dbl> <chr>   <dbl>           <dbl>
##  1 Luke Skywalker       1.72    77 Human    26.0         7.7 e 1
##  2 C-3PO                1.67    75 Droid    26.9         5.78e 3
##  3 R2-D2                0.96    32 Droid    34.7         1.85e 5
##  4 Darth Vader          2.02   136 Human    33.3         2.51e 7
##  5 Leia Organa          1.5     49 Human    21.8         1.23e 9
##  6 Owen Lars            1.78   120 Human    37.9         1.48e11
##  7 Beru Whitesun lars   1.65    75 Human    27.5         1.11e13
##  8 R5-D4                0.97    32 Droid    34.0         3.55e14
##  9 Biggs Darklighter    1.83    84 Human    25.1         2.98e16
## 10 Obi-Wan Kenobi       1.82    77 Human    23.2         2.29e18
```

BMI 为体重指数，国际标准超过 30 即为肥胖。

```
starwars_short %>%
  mutate(height = height/100,
         BMI = mass/(height^2),
         obese = if_else(BMI > 30, "YES", "NO")) #obese(肥胖)
```

```
## # A tibble: 10 x 6
##    name               height  mass species   BMI obese
##    <chr>               <dbl> <dbl> <chr>   <dbl> <chr>
##  1 Luke Skywalker       1.72    77 Human    26.0 NO
##  2 C-3PO                1.67    75 Droid    26.9 NO
##  3 R2-D2                0.96    32 Droid    34.7 YES
##  4 Darth Vader          2.02   136 Human    33.3 YES
##  5 Leia Organa          1.5     49 Human    21.8 NO
##  6 Owen Lars            1.78   120 Human    37.9 YES
##  7 Beru Whitesun lars   1.65    75 Human    27.5 NO
##  8 R5-D4                0.97    32 Droid    34.0 YES
##  9 Biggs Darklighter    1.83    84 Human    25.1 NO
## 10 Obi-Wan Kenobi       1.82    77 Human    23.2 NO
```

```
#if_else 与 baseR 中的 ifelse 功能类似
```

```
starwars_short %>%
  mutate(height = height/100,
         BMI = mass/(height^2),
         obese = if_else(BMI > 30, "YES", "NO")) %>%
  filter(species == "Human", obese == "YES")
```

```
## # A tibble: 2 x 6
##   name        height  mass species   BMI obese
##   <chr>        <dbl> <dbl> <chr>   <dbl> <chr>
## 1 Darth Vader   2.02   136 Human    33.3 YES
## 2 Owen Lars     1.78   120 Human    37.9 YES
```

上述代码过滤出人类来，只有人类才有肥胖的概念。当超过 3 种或 3 种以上的判别结果，if_else 需要嵌套，dplpr 推荐用 case_when 完成标记。

```
starwars_short %>%
  mutate(height = height/100,
         BMI = mass/(height^2),
         obese = case_when(BMI > 30 ~ "YES"))
```

```
## # A tibble: 10 x 6
##    name               height  mass species   BMI obese
##    <chr>               <dbl> <dbl> <chr>   <dbl> <chr>
##  1 Luke Skywalker       1.72    77 Human    26.0 <NA>
##  2 C-3PO                1.67    75 Droid    26.9 <NA>
##  3 R2-D2                0.96    32 Droid    34.7 YES
##  4 Darth Vader          2.02   136 Human    33.3 YES
##  5 Leia Organa          1.5     49 Human    21.8 <NA>
##  6 Owen Lars            1.78   120 Human    37.9 YES
##  7 Beru Whitesun lars   1.65    75 Human    27.5 <NA>
##  8 R5-D4                0.97    32 Droid    34.0 YES
##  9 Biggs Darklighter    1.83    84 Human    25.1 <NA>
## 10 Obi-Wan Kenobi       1.82    77 Human    23.2 <NA>
```

```
# obese = case_when(BMI > 30 ~ "YES",BMI <= 30 ~ "NO")
```

cumsum 计算角色的累积质量；

cummax 通过两两对比相邻的观测值，来求得两者之间的最大值并返回修改值；

cummean 会在累加观测值之后除以累加观测值的个数。

```
starwars_short %>%
  select(-height, -species) %>%
  mutate(cum_mass = cumsum(mass),
         max_mass = cummax(mass),
         mean_mass = cummean(mass))
```

```
## # A tibble: 10 x 5
##    name                mass cum_mass max_mass mean_mass
##    <chr>              <dbl>    <dbl>    <dbl>     <dbl>
##  1 Luke Skywalker        77       77       77      77
##  2 C-3PO                 75      152       77      76
##  3 R2-D2                 32      184       77      61.3
##  4 Darth Vader          136      320      136      80
##  5 Leia Organa           49      369      136      73.8
##  6 Owen Lars            120      489      136      81.5
##  7 Beru Whitesun lars    75      564      136      80.6
##  8 R5-D4                 32      596      136      74.5
##  9 Biggs Darklighter     84      680      136      75.6
## 10 Obi-Wan Kenobi        77      757      136      75.7
```

```
starwars_short %>%
  select(-species) %>%
  mutate(order = row_number(mass),
         ntile = ntile(height,n = 2),
         diff_heigt = c(NA,diff(height)))
```

```
## # A tibble: 10 x 6
##    name               height  mass order ntile diff_heigt
##    <chr>               <int> <dbl> <int> <int>      <int>
##  1 Luke Skywalker        172    77     6     2         NA
##  2 C-3PO                 167    75     4     1         -5
##  3 R2-D2                  96    32     1     1        -71
##  4 Darth Vader           202   136    10     2        106
##  5 Leia Organa           150    49     3     1        -52
##  6 Owen Lars             178   120     9     2         28
##  7 Beru Whitesun lars    165    75     5     1        -13
##  8 R5-D4                  97    32     2     1        -68
##  9 Biggs Darklighter     183    84     8     2         86
## 10 Obi-Wan Kenobi        182    77     7     2         -1
```

```
#diff 函数计算相邻角色之间身高差
```

summarise 函数

总结函数，提取集中性的指标和离散性的指标。前者包括均值，众数，中位数，后者包括标准差和区间等。

```
starwars_short %>%
  group_by(species) %>%
  summarise(avg_mass = mean(mass, na.rm = TRUE),
            avg_height = mean(height, na.rm = TRUE),
            n = n(),
            sd_mass = sd(mass, na.rm = TRUE),
            se = sd_mass/sqrt(n))
```

```
## # A tibble: 2 x 6
##   species avg_mass avg_height     n sd_mass    se
##   <chr>      <dbl>      <dbl> <int>   <dbl> <dbl>
## 1 Droid       46.3        120     3    24.8  14.3
## 2 Human       88.3        176     7    29.7  11.2
```

结论是人类平均体重要重些；

平均身高要比机器人高些；

人类体重分布范围比机器人更大些。

```
starwars_short %>%
  group_by(species) %>%
  summarise(max_mass = max(mass, na.rm = TRUE),
            min_mass = min(mass, na.rm = TRUE),
            median_mass = median(mass, na.rm = TRUE),
            range_mass = max_mass-min_mass,
            range = diff(range(mass))) #range 函数返回一个向量，diff 求差值
```

```
## # A tibble: 2 x 6
##   species max_mass min_mass median_mass range_mass range
##   <chr>      <dbl>    <dbl>       <dbl>      <dbl> <dbl>
## 1 Droid         75       32          32         43    43
## 2 Human        136       49          77         87    87
```

添加最大最小值，范围。

可视化

```r
starwars_short %>%
  group_by(species) %>%
  summarise(avg_mass = mean(mass, na.rm = TRUE),
            avg_height = mean(height, na.rm = TRUE),
            n = n(),
            sd_mass = sd(mass, na.rm = TRUE),
            se = sd_mass/sqrt(n)) %>%
  ggplot(aes(species))+ #species 置于 aes 中作为图标的 x 轴
  geom_point(aes(y = avg_mass), shape = 2)+  # 变量 mass 作为 y 轴
  geom_point(data = starwars_short, aes(y = mass))+
  geom_errorbar(aes(ymin = avg_mass - se, ymax = avg_mass + se))+  # 增加误差图层，有上限，下限
  theme_classic()
```

参考文献

刘健邬书豪，《R 数据科学实战工具详解与案例分析》，机械工业出版社，2019 年 7 月。