Practical 1: Data Wrangling

Nina van Gerwen

13-09-2018

First, load the packages:

```
library(ISLR)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(haven)
library(readxl)
```

1. Objects and classes

```
object_1 <- 1:5 ## integers
object_2 <- 1L:5L ## integers
object_3 <- "-123.456" ## character
object_4 <- as.numeric(object_2) ## numeric
object_5 <- letters[object_1] ## characters
object_6 <- as.factor(rep(object_5, 2)) ## factors
object_7 <- c(1, 2, 3, "4", "5", "6") ## characters
```

2. Converting

```
object_7 <- as.numeric(object_7)
```

3. Making a list

```
objects <- list(object_1, object_2, object_3, object_4, object_5,
                  object_6, object_7)
```

4. Making a data frame

```
object_frame <- data.frame(object_1, object_2, object_5)
```

5. Determining the size of a data frame

```
ncol(object_frame) ## 3
```

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

```
nrow(object_frame) ## 5
```

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

Loading, viewing and summarising data

6. Importing google data

```
apps <- read_csv("Data/googleplaystore.csv")
```

```
## Rows: 10841 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (11): App, Category, Size, Installs, Type, Price, Content Rating, Genres...
## dbl  (2): Rating, Reviews
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

7. Unexpected results

```
str(apps)
```

```
## spec_tbl_df [10,841 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ App           : chr [1:10841] "Photo Editor & Candy Camera & Grid & ScrapBook" "Co
##  $ Category      : chr [1:10841] "ART_AND_DESIGN" "ART_AND_DESIGN" "ART_AND_DESIGN" "
##  $ Rating        : num [1:10841] 4.1 3.9 4.7 4.5 4.3 4.4 3.8 4.1 4.4 4.7 ...
##  $ Reviews       : num [1:10841] 159 967 87510 215644 967 ...
##  $ Size          : chr [1:10841] "19M" "14M" "8.7M" "25M" ...
##  $ Installs      : chr [1:10841] "10,000+" "500,000+" "5,000,000+" "50,000,000+" ...
##  $ Type          : chr [1:10841] "Free" "Free" "Free" "Free" ...
##  $ Price         : chr [1:10841] "0" "0" "0" "0" ...
##  $ Content Rating: chr [1:10841] "Everyone" "Everyone" "Everyone" "Teen" ...
##  $ Genres        : chr [1:10841] "Art & Design" "Art & Design;Pretend Play" "Art & De
##  $ Last Updated  : chr [1:10841] "January 7, 2018" "January 15, 2018" "August 1, 2018
##  $ Current Ver   : chr [1:10841] "1.0.0" "2.0.0" "1.2.4" "Varies with device" ...
##  $ Android Ver   : chr [1:10841] "4.0.3 and up" "4.0.3 and up" "4.0.3 and up" "4.2 an
##  - attr(*, "spec")=
##   .. cols(
##   ..   App = col_character(),
##   ..   Category = col_character(),
##   ..   Rating = col_double(),
##   ..   Reviews = col_double(),
##   ..   Size = col_character(),
##   ..   Installs = col_character(),
##   ..   Type = col_character(),
##   ..   Price = col_character(),
##   ..   `Content Rating` = col_character(),
##   ..   Genres = col_character(),
##   ..   `Last Updated` = col_character(),
##   ..   `Current Ver` = col_character(),
##   ..   `Android Ver` = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

The variables ‘Category’, ‘Size’, ‘Installs’, ‘Price’, ‘Content Rating’, ‘Genres’ and ‘Android Ver’ should probably be a Factor instead of a Character.

8. The head

```
head(apps)
```

```
## # A tibble: 6 x 13
##   App        Category Rating Reviews Size  Installs Type  Price `Content Rating`
##   <chr>      <chr>     <dbl>   <dbl> <chr> <chr>    <chr> <chr> <chr>
## 1 "Photo Ed~ ART_AND~    4.1     159 19M   10,000+  Free  0     Everyone
## 2 "Coloring~ ART_AND~    3.9     967 14M   500,000+ Free  0     Everyone
## 3 "U Launch~ ART_AND~    4.7   87510 8.7M  5,000,0~ Free  0     Everyone
## 4 "Sketch -~ ART_AND~    4.5  215644 25M   50,000,~ Free  0     Teen
## 5 "Pixel Dr~ ART_AND~    4.3     967 2.8M  100,000+ Free  0     Everyone
## 6 "Paper fl~ ART_AND~    4.4     167 5.6M  50,000+  Free  0     Everyone
## # ... with 4 more variables: Genres <chr>, `Last Updated` <chr>,
## #   `Current Ver` <chr>, `Android Ver` <chr>
```

9. Students data

```
## Loading data
students <- read_xlsx("Data/students.xlsx")

## Seeing the class of every variable
str(students)
```

```
## tibble [37 x 3] (S3: tbl_df/tbl/data.frame)
##  $ student_number: num [1:37] 5117250 6562582 6000241 4862862 6561723 ...
##  $ grade         : num [1:37] 6.54 7.57 6.08 7.71 6.57 ...
##  $ programme     : chr [1:37] "A" "A" "B" "A" ...
```

```
head(students)
```

```
## # A tibble: 6 x 3
##   student_number grade programme
##            <dbl> <dbl> <chr>
## 1        5117250  6.54 A
## 2        6562582  7.57 A
## 3        6000241  6.08 B
## 4        4862862  7.71 A
## 5        6561723  6.57 B
## 6        5625916  7.90 B
```

Again, the variable ‘Programme’ should probably be a factor.

10. Summarising

```
summary(students)
```

```
##  student_number        grade         programme
##  Min.   :4011659   Min.   :4.844   Length:37
##  1st Qu.:4862862   1st Qu.:6.390   Class :character
##  Median :6000241   Median :7.151   Mode  :character
##  Mean   :5686729   Mean   :6.991
##  3rd Qu.:6553913   3rd Qu.:7.573
##  Max.   :6997130   Max.   :9.291
```

The grades achieved by students ranges from 4.84 to 9.29 with an average grade of 7.15.

Data transformation

11. Filtering

```
students %>% filter(., grade < 5.5)
```

```
## # A tibble: 3 x 3
##   student_number grade programme
##            <dbl> <dbl> <chr>
## 1        6114656  5.16 A
## 2        5265402  5.49 B
## 3        4639846  4.84 A
```

12. More filtering

```
students %>% filter(., grade > 8 & programme == "A")
```

```
## # A tibble: 5 x 3
##   student_number grade programme
##            <dbl> <dbl> <chr>
## 1        6352581  8.09 A
## 2        6165611  8.02 A
## 3        4133949  8.40 A
## 4        4011659  8.94 A
## 5        6553913  8.24 A
```

13. Arranging

```
students %>% arrange(., programme, desc(grade))
```

```
## # A tibble: 37 x 3
##    student_number grade programme
##             <dbl> <dbl> <chr>
##  1        4011659  8.94 A
##  2        4133949  8.40 A
##  3        6553913  8.24 A
##  4        6352581  8.09 A
##  5        6165611  8.02 A
##  6        6997130  7.75 A
##  7        4862862  7.71 A
##  8        6562582  7.57 A
##  9        4483974  7.46 A
## 10        5128923  7.26 A
## # ... with 27 more rows
```

We use desc(grade) because we want the order to be descending and not ascending.

14. Selecting

```
students %>% select(., student_number, programme)
```

```
## # A tibble: 37 x 2
##    student_number programme
##             <dbl> <chr>
##  1        5117250 A
##  2        6562582 A
##  3        6000241 B
##  4        4862862 A
##  5        6561723 B
##  6        5625916 B
##  7        4096023 A
##  8        6114656 A
##  9        5265402 B
## 10        5977188 B
## # ... with 27 more rows
```

15. Mutating

```
students_recoded <- students %>% mutate(., programme = recode(programme,
                                        A = "Science", B = "Social Science"))
```

So the above code states: students_recoded will be students, where we mutated the column programme such that A = Sciencce, B = Social Science.

Data processing pipelines

16. Pipelining

```
popular_apps <-
  read_csv("Data/googleplaystore.csv") %>%
  mutate(Downloads = parse_number(Installs)) %>%
  filter(Downloads > 500000000) %>%
  arrange(desc(Rating)) %>%
  select(Rating, Category, Price, App) %>%
  distinct(App, .keep_all = TRUE)
```

```
## Rows: 10841 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (11): App, Category, Size, Installs, Type, Price, Content Rating, Genres...
## dbl  (2): Rating, Reviews
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
popular_apps
```

```
## # A tibble: 20 x 4
##    Rating Category            Price App
##     <dbl> <chr>               <chr> <chr>
##  1    4.5 GAME                0     "Subway Surfers"
##  2    4.5 SOCIAL              0     "Instagram"
##  3    4.5 PHOTOGRAPHY         0     "Google Photos"
##  4    4.4 COMMUNICATION       0     "WhatsApp Messenger"
##  5    4.4 TOOLS               0     "Google"
##  6    4.4 PRODUCTIVITY        0     "Google Drive"
##  7    4.3 COMMUNICATION       0     "Google Chrome: Fast & Secure"
##  8    4.3 COMMUNICATION       0     "Gmail"
##  9    4.3 ENTERTAINMENT       0     "Google Play Games"
## 10    4.3 TRAVEL_AND_LOCAL    0     "Maps - Navigate & Explore"
## 11    4.3 VIDEO_PLAYERS       0     "YouTube"
## 12    4.2 SOCIAL              0     "Google+"
## 13    4.2 TRAVEL_AND_LOCAL    0     "Google Street View"
## 14    4.1 COMMUNICATION       0     "Skype - free IM & video calls"
## 15    4.1 SOCIAL              0     "Facebook"
## 16    4   COMMUNICATION       0     "Messenger \x96 Text and Video Chat for Fre~
## 17    4   COMMUNICATION       0     "Hangouts"
## 18    3.9 BOOKS_AND_REFERENCE 0     "Google Play Books"
## 19    3.9 NEWS_AND_MAGAZINES  0     "Google News"
## 20    3.7 VIDEO_PLAYERS       0     "Google Play Movies & TV"
```

Grouping and summarisation

17 & 18. Summarising

The median of which variable exactly? I'm guessing Rating.

```
mad <- function(x){
  median(abs(x - median(x)))
}
popular_apps %>%
  summarise(
    median = median(Rating),
    min = min(Rating),
    max = max(Rating),
    mad = mad(Rating)
  )
```

```
## # A tibble: 1 x 4
##   median   min   max   mad
##    <dbl> <dbl> <dbl> <dbl>
## 1    4.3   3.7   4.5 0.150
```

19. Grouped summary

```
popular_apps %>%
  group_by(Category) %>%
  summarise(
    median = median(Rating),
    min = min(Rating),
    max = max(Rating),
    mad = mad(Rating)
  )
```

```
## # A tibble: 11 x 5
##    Category            median   min   max    mad
##    <chr>                <dbl> <dbl> <dbl>  <dbl>
##  1 BOOKS_AND_REFERENCE   3.9    3.9   3.9 0
##  2 COMMUNICATION         4.2    4     4.4 0.150
##  3 ENTERTAINMENT         4.3    4.3   4.3 0
##  4 GAME                  4.5    4.5   4.5 0
##  5 NEWS_AND_MAGAZINES    3.9    3.9   3.9 0
##  6 PHOTOGRAPHY           4.5    4.5   4.5 0
##  7 PRODUCTIVITY          4.4    4.4   4.4 0
##  8 SOCIAL                4.2    4.1   4.5 0.100
##  9 TOOLS                 4.4    4.4   4.4 0
## 10 TRAVEL_AND_LOCAL      4.25   4.2   4.3 0.0500
## 11 VIDEO_PLAYERS         4      3.7   4.3 0.300
```

Final Exercise

Summary of how many Downloads per category:

```
own_summary <-
  read_csv("Data/googleplaystore.csv") %>%
  mutate(Downloads = parse_number(Installs)) %>%
  group_by(Category) %>%
  summarise(
    mean = mean(Downloads),
    min = min(Downloads),
    max = max(Downloads),
    var = var(Downloads)
  ) %>%
  arrange(desc(mean))
```

```
## Rows: 10841 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (11): App, Category, Size, Installs, Type, Price, Content Rating, Genres...
## dbl  (2): Rating, Reviews
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
own_summary
```

```
## # A tibble: 34 x 5
##    Category                mean   min        max     var
##    <chr>                  <dbl> <dbl>      <dbl>   <dbl>
##  1 COMMUNICATION      84359887.     1 1000000000 5.42e16
##  2 SOCIAL             47694467.     0 1000000000 3.07e16
##  3 VIDEO_PLAYERS      35554301.    10 1000000000 2.09e16
##  4 PRODUCTIVITY       33434178.     0 1000000000 1.52e16
##  5 GAME               30669602.     1 1000000000 9.40e15
##  6 PHOTOGRAPHY        30114172.     5 1000000000 1.24e16
##  7 TRAVEL_AND_LOCAL   26623594.     0 1000000000 1.92e16
##  8 NEWS_AND_MAGAZINES 26488755.     0 1000000000 1.88e16
##  9 ENTERTAINMENT      19256107. 10000 1000000000 7.18e15
## 10 TOOLS              13585732.     1 1000000000 5.25e15
## # ... with 24 more rows
```

From this summary, we can see that on average ‘Communication’ apps get the most downloads. However, we do not know whether the difference between the other apps categories and Communication is significant (though I would bet good money on it).