

# Practical 1

Jim Regtien

12-9-2022

## Exercise 1

```
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.2 --
## v ggplot2 3.3.6    v purrr  0.3.4
## v tibble  3.1.8    v dplyr  1.0.7
## v tidyr   1.2.1    v stringr 1.4.0
## v readr   2.1.2    v forcats 0.5.1

## Warning: package 'readr' was built under R version 4.0.5
## Warning: package 'purrr' was built under R version 4.0.5
## Warning: package 'dplyr' was built under R version 4.0.5
## Warning: package 'stringr' 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()

library(readxl)

## Warning: package 'readxl' was built under R version 4.0.5

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

The first object will be a list of ints, same for the second and fourth objects. The third object will be a string/character. Object 5 will be a vector/list of strings, like the seventh object. The 6th object is a factor.

```
print(class(object_1))

## [1] "integer"

print(class(object_2))

## [1] "integer"

print(class(object_3))
```

```
## [1] "character"
print(class(object_4))
```

```
## [1] "numeric"
print(class(object_5))
```

```
## [1] "character"
print(class(object_6))
```

```
## [1] "factor"
print(class(object_7))
```

```
## [1] "character"
```

## Exercise 2

```
as.numeric(object_7 )
```

```
## [1] 1 2 3 4 5 6
```

## Exercise 3

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

```
## [[1]]
## [1] 1 2 3 4 5
##
## [[2]]
## [1] 1 2 3 4 5
##
## [[3]]
## [1] "-123.456"
##
## [[4]]
## [1] 1 2 3 4 5
##
## [[5]]
## [1] "a" "b" "c" "d" "e"
##
## [[6]]
## [1] a b c d e a b c d e
## Levels: a b c d e
##
## [[7]]
## [1] "1" "2" "3" "4" "5" "6"
```

## Exercise 4

```
obj_df <- data.frame(var_1 = object_1, var_2 = object_2, var_3 = object_5)
obj_df
```

```
##   var_1 var_2 var_3
```

```
## 1      1      1      a
## 2      2      2      b
## 3      3      3      c
## 4      4      4      d
## 5      5      5      e
```

### Exercise 5

```
print(nrow(obj_df))
```

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

```
print(ncol(obj_df))
```

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

### Exercise 9

```
apps <- read_csv('googleplaystore.csv')
print(head(apps))
```

```
## # A tibble: 6 x 13
##   App      Categ~1 Rating Reviews Size  Insta~2 Type  Price Conte~3 Genres Last ~4
##   <chr>   <chr>      <dbl>   <dbl> <chr> <chr>   <chr> <chr> <chr>   <chr> <chr>
## 1 "Phot~ ART_AN~    4.1     159 19M   10,000+ Free  0    Everyo~ Art &~ Januar~
## 2 "Colo~ ART_AN~    3.9     967 14M   500,00~ Free  0    Everyo~ Art &~ Januar~
## 3 "U La~ ART_AN~    4.7   87510 8.7M   5,000,~ Free  0    Everyo~ Art &~ August~
## 4 "Sket~ ART_AN~    4.5  215644 25M   50,000~ Free  0    Teen    Art &~ June 8~
## 5 "Pixe~ ART_AN~    4.3     967 2.8M   100,00~ Free  0    Everyo~ Art &~ June 2~
## 6 "Pape~ ART_AN~    4.4     167 5.6M   50,000+ Free  0    Everyo~ Art &~ March ~
## # ... with 2 more variables: `Current Ver` <chr>, `Android Ver` <chr>, and
## # abbreviated variable names 1: Category, 2: Installs, 3: `Content Rating`,
## # 4: `Last Updated`
```

The number of reviews is a double, while I would find it more natural for it to be an integer. The price is a chr, while it might be better if it was a double.

### Exercise 9

```
students <- read_xlsx('students.xlsx')
print(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
```

The student number should be an integer, as that type of objects requires less memory.

## Exercise 10

```
summarise(students, mean = round(mean(grade), 2), median = round(median(grade), 2),
          variance = round(var(grade), 2), min = round(min(grade), 2),
          max = round(max(grade), 2))
```

```
## # A tibble: 1 x 5
##   mean median variance   min   max
##   <dbl> <dbl>   <dbl> <dbl> <dbl>
## 1  6.99   7.15     1.06  4.84  9.29
```

The grades range from a 4.8 to a 9.3.

## Exercise 11

```
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
```

## Exercise 12

```
students %>%
  filter(grade >= 8) %>%
  filter(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
```

## Exercise 13

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

```
## # A tibble: 37 x 3
##   student_number grade programme
##           <dbl> <dbl> <chr>
## 1         4639846  4.84 A
## 2         6114656  5.16 A
## 3         4096023  5.92 A
## 4         6207923  6.00 A
## 5         5117250  6.54 A
## 6         6120285  6.71 A
## 7         6580486  6.73 A
```

```
## 8      6040650 6.75 A
## 9      6827756 6.80 A
## 10     5128923 7.26 A
## # ... with 27 more rows
```

#### Exercise 14

```
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
```

#### Exercise 15

```
students_recoded <- mutate(students, prog = recode(as.vector(students$programme),
  A = "Science",
  B = "Social Science"))
```

#### Exercise 16

```
popular_apps <- (read_csv('googleplaystore.csv') %>%
  mutate(downloads = parse_number(Installs)) %>%
  filter(downloads >= 5e7) %>%
  arrange(desc(Rating)))
```

#### Exercise 17

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

```
## # A tibble: 1 x 3
##   median  min  max
##   <dbl> <dbl> <dbl>
## 1    4.4    3.1    4.8
```

#### Exercise 18

```
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    4.4  3.1   4.8 0.100
```

## Exercise 19

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

## # A tibble: 23 x 5
##   Category      median    min    max    mad
##   <chr>         <dbl> <dbl> <dbl> <dbl>
## 1 ART_AND_DESIGN      4.5  4.5  4.5  0
## 2 BOOKS_AND_REFERENCE 4.5  3.9  4.7 0.200
## 3 BUSINESS            4.2  3.8  4.5 0.100
## 4 COMMUNICATION       4.3  4    4.6 0.100
## 5 EDUCATION           4.7  4.7  4.7  0
## 6 ENTERTAINMENT       4.3  3.7  4.6 0.100
## 7 FAMILY              4.4  3.7  4.7 0.100
## 8 FINANCE             4.2  4.2  4.3  0
## 9 GAME                4.4  3.7  4.7 0.100
## 10 HEALTH_AND_FITNESS 4.6  4.3  4.8  0
## # ... with 13 more rows
```

## Exercise 20

```
popular_apps %>%
  filter(Category == "GAME" | Category == 'EDUCATION') %>%
  group_by(Category) %>%
  summarise(median = median(Rating), max = max(Rating), min = min(Rating), var = var(Rating))

## # A tibble: 2 x 5
##   Category median    max    min    var
##   <chr>      <dbl> <dbl> <dbl> <dbl>
## 1 EDUCATION  4.7    4.7  4.7  0
## 2 GAME       4.4    4.7  3.7 0.0218

apps %>%
  group_by(Type) %>%
  na.omit() %>% #Ommiting the NaNs
  summarise(mean = mean(Rating), median = median(Rating), sd = sd(Rating))

## # A tibble: 2 x 4
##   Type    mean median    sd
##   <chr> <dbl>  <dbl> <dbl>
## 1 Free  4.19   4.3 0.512
## 2 Paid  4.27   4.4 0.548
```