Answers to the exercises from Chapter 2

Sébastien Wieckowski

2021-08-31

Contents

Data exercises

Question 1

Create a character vector called my_names that contains all your first, middle and last names as elements. Calculate the length of my_names.

```
my_names <- c("Sébastien", "Wieckowski")
length(my_names)
```

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

Question 2

Create a second numeric vector called which which corresponds to my_names. The entries should be the position of each name in the order of your full name. Verify that it has the same length as my_names.

```
which <- c(1,2)
length(which)
```

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

We verify if the dimensions of the two vectors are identical: TRUE.

Question 3

Create a dataframe called names, which consists of the two vectors my_names and which as columns. Calculate the dimensions of names.

```
names <- data.frame(my_names, which)
```

The dimension of the vector `names` is 2, 2.

Question 4

Create a new dataframe new_names with the which column converted to character type. Verify that your command worked using str().

```
new_names <- data.frame(my_names, as.character(which))
str(new_names)
```

```
## 'data.frame':    2 obs. of  2 variables:
##  $ my_names           : chr  "Sébastien" "Wieckowski"
##  $ as.character.which.: chr  "1" "2"
```

Question 5

Load the ugtests data set via the peopleanalyticsdata package or download it from the internet7. Calculate the dimensions of ugtests and view the first three rows only.

```
ugtests <- read.csv("http://peopleanalytics-regression-book.org/data/ugtests.csv")
dim(ugtests)
```

```
## [1] 975   4
```

```
head(x=ugtests, n=3)
```

```
##   Yr1 Yr2 Yr3 Final
## 1  27  50  52    93
## 2  70 104 126   207
## 3  27  36 148   175
```

Question 6

View a statistical summary of all of the columns of ugtests. Determine if there are any missing values.

```
summary(ugtests)
```

```
##       Yr1             Yr2             Yr3            Final
##  Min.   : 3.00   Min.   :  6.0   Min.   :  8.0   Min.   :  8
##  1st Qu.:42.00   1st Qu.: 73.0   1st Qu.: 81.0   1st Qu.:118
##  Median :53.00   Median : 94.0   Median :105.0   Median :147
##  Mean   :52.15   Mean   : 92.4   Mean   :105.1   Mean   :149
##  3rd Qu.:62.00   3rd Qu.:112.0   3rd Qu.:130.0   3rd Qu.:175
##  Max.   :99.00   Max.   :188.0   Max.   :198.0   Max.   :295
```

There are 0 missing values.

Question 7

View the subset of ugtests for values of Yr1 greater than 50.

```
ugtests %>%
  subset(subset=(ugtests$Yr1 > 50)) %>%
  head(n = 10)
```

```
##    Yr1 Yr2 Yr3 Final
## 2   70 104 126   207
## 6   86 122 119   159
## 8   60  92  78    84
## 10  80 127  67    80
## 13  64 123 110   175
## 14  62  84 142   182
## 15  61  65 134   155
## 16  60 150 116   198
## 17  58  76 107   161
```

```
## 19  64  87 106   100
```

Question 8

Install and load the package dplyr. Look up the help for the filter() function in this package and try to use it to repeat the task in the previous question.

```
library(dplyr)
# help(filter) # look up at the help

ugtests %>%
  filter(Yr1 > 50) %>%
  head(n = 10)
```

```
##    Yr1 Yr2 Yr3 Final
## 1   70 104 126   207
## 2   86 122 119   159
## 3   60  92  78    84
## 4   80 127  67    80
## 5   64 123 110   175
## 6   62  84 142   182
## 7   61  65 134   155
## 8   60 150 116   198
## 9   58  76 107   161
## 10  64  87 106   100
```

Question 9

Write code to find the mean of the Yr1 test scores for all those who achieved Yr3 test scores greater than 100. Round this mean to the nearest integer.

R base-style

```
ugtests$Yr1 %>%
  subset(ugtests$Yr3 > 100) %>%
  mean() %>%
  round()
```

```
## [1] 52
```

Attempt using Dplyr

```
ugtests %>%
  filter(Yr3 > 100) %>%
  summarise(avg=mean(Yr1)) %>% # mean function doesn't work (see below)...
  round()
```

```
##   avg
## 1  52
```

The `mean` base function doesn't work on the output of the pipe because the upstream `filter` function generates a data frame structure and not a vector. One solution is to use the `colMeans` function:

```
ugtests %>%
  filter(Yr3 > 100) %>%
  select(Yr1) %>%
  colMeans() %>%
  round()
```

```
## Yr1
##  52
```

Question 10

Familiarize yourself with the two functions filter() and pull() from dplyr. Use these functions to try to do the same calculation in the previous question using a single unbroken piped command. Be sure to namespace where necessary.

```
ugtests %>%
  filter(Yr3 > 100) %>%
  pull(Yr1) %>%
# pull() creates a vector -- which, in this case, is numeric --
# whereas select() creates a data frame. Basically, pull() is the equivalent
# to writing ugtests$Yr1 or ugtests[, "Yr1"], whereas select() removes all of
# the columns except for Yr1 but maintains the data frame structure.
  mean() %>%
  round()
```

```
## [1] 52
```

Question 11

Create a scatter plot using the ugtests data with Final scores on the y axis and Yr3 scores on the x axis.

```
plot(
  x = ugtests$Yr3,
  y = ugtests$Final,
  xlab = "Yr3 scores",
  ylab = "Final scores",
  main = "ugtests scatter plot"
)
```

Question 12

Create your own 5-level grading logic and use it to create a new finalgrade column in the ugtests data set with grades 1-5 of increasing attainment based on the Final score in ugtests. Generate a histogram of this finalgrade column.

```
ugtests$final_grade <- ugtests$Final %>%
  cut(breaks = 5, ordered_result = T, labels = c(1:5)) %>%
  as.numeric()

head(ugtests)
```

```
##   Yr1 Yr2 Yr3 Final final_grade
## 1  27  50  52    93           2
## 2  70 104 126   207           4
## 3  27  36 148   175           3
## 4  26  75 115   125           3
## 5  46  77  75   114           2
## 6  86 122 119   159           3
```

```
hist(ugtests$final_grade, breaks = 0:5)
```

Question 13

```
boxplot(
  formula = Yr3 ~ final_grade,
  data = ugtests,
  xlab = "Final score grade",
  ylab = "Yr3 score",
  main = "box plot"
)
```

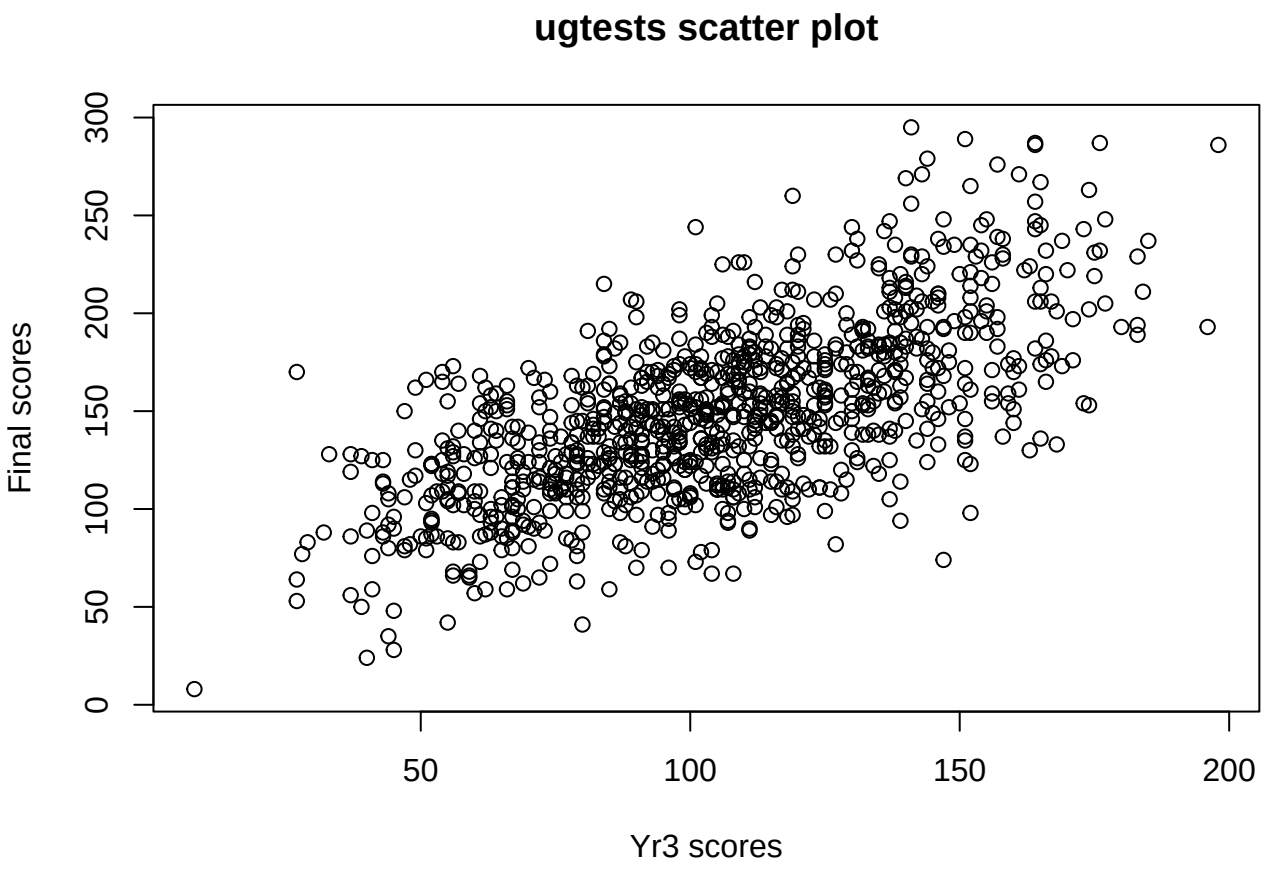

Figure 1: Scatter plot.

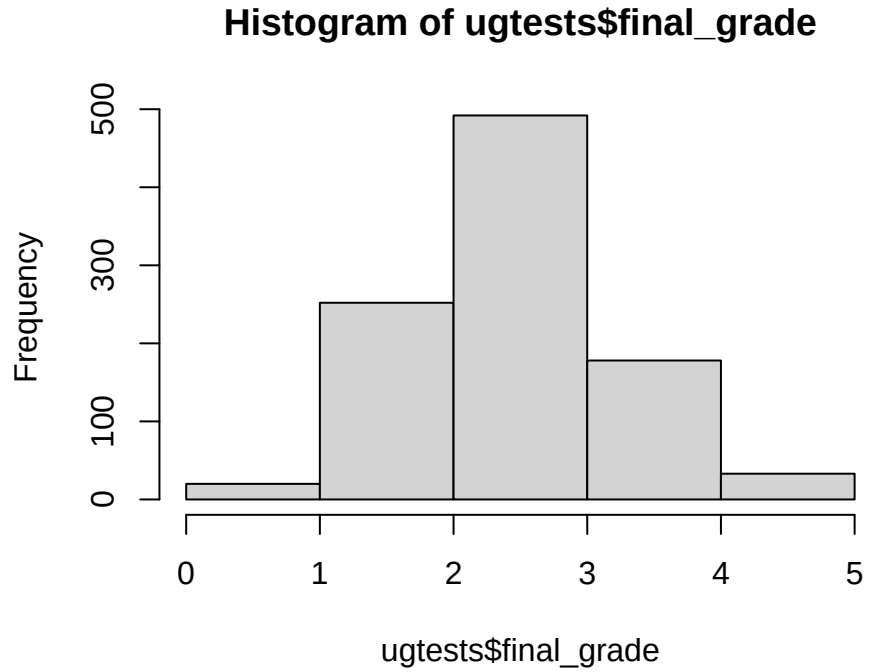

Figure 2: Histogram.

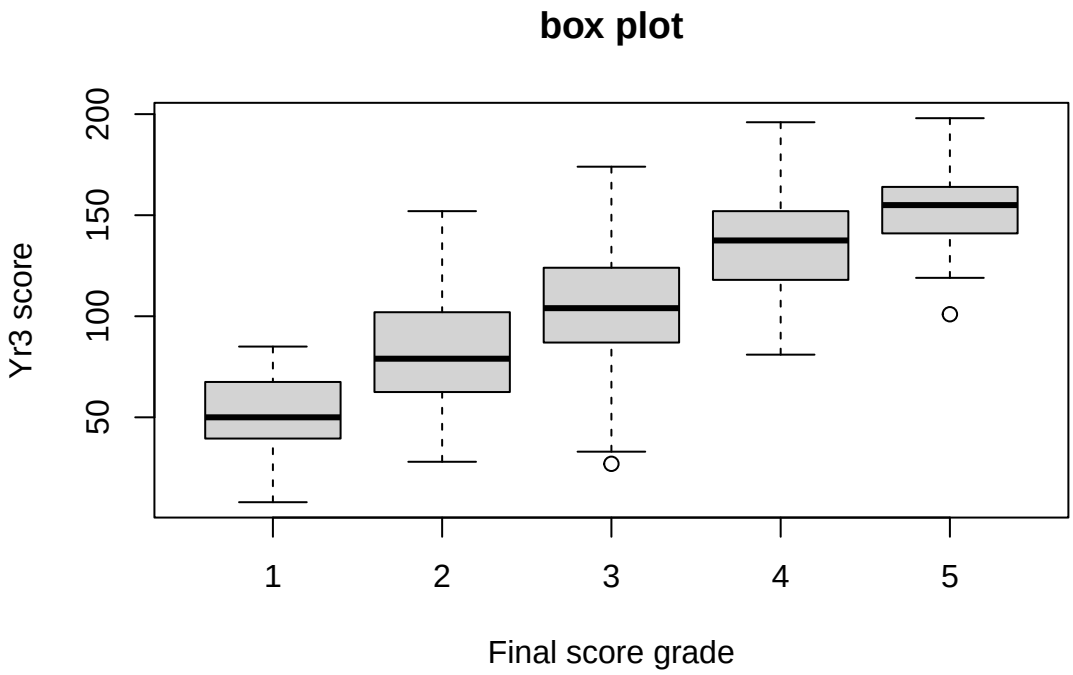

Figure 3: Boxplot.

Session info

The output from running `sessionInfo` is shown below and details all packages and version necessary to reproduce the results from this report.

```
sessionInfo()
```

```
## R version 4.1.1 (2021-08-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19043)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=French_France.1252  LC_CTYPE=French_France.1252
## [3] LC_MONETARY=French_France.1252 LC_NUMERIC=C
## [5] LC_TIME=French_France.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] dplyr_1.0.7    knitr_1.33     tinytex_0.33   rmarkdown_2.10
##
## loaded via a namespace (and not attached):
##  [1] magrittr_2.0.1   tidyselect_1.1.1 R6_2.5.1         rlang_0.4.11
##  [5] fastmap_1.1.0    fansi_0.5.0      highr_0.9        stringr_1.4.0
##  [9] tools_4.1.1      xfun_0.25        utf8_1.2.2       htmltools_0.5.2
## [13] ellipsis_0.3.2   yaml_2.2.1       digest_0.6.27    tibble_3.1.4
## [17] lifecycle_1.0.0  crayon_1.4.1     purrr_0.3.4      vctrs_0.3.8
```

```
## [21] glue_1.4.2       evaluate_0.14    stringi_1.7.4    compiler_4.1.1
## [25] pillar_1.6.2     generics_0.1.0   jsonlite_1.7.2   pkgconfig_2.0.3
```

```
# for compiling the Rmd document to HTML and PDF using radian:
# rmarkdown::render("chapter2.Rmd", 'html_document')
# rmarkdown::render("chapter2.Rmd", 'pdf_document')
```