R Lab Graded Assignment 1

Stavros Nikolakopoulos*
Department of Statistics, AUEB

Introduction

This assignment is to be done on an individual basis. The scoring will be 0-10 (irrespective of the total sum of the points), and it will have a weight of 40% of the overall R Labs grade (which counts for 10% of the overall course grade).

The data to be used for this assignment will be scraped from `https://www.metacritic.com`. You will have to manipulate data from film reviews (both critics- and public-based). Critics reviews (from *Washington Post*) are described in the `critic` variable while public reviews in the `metascore` variable.

Please answer the questions below. You may submit the script as a solution, with filename `YOURSURNAME_P/FT.R`, where `P/F` refers to whether you are a Part or Full time student. Assignment is due Friday 23/10/2020 at 23:59.

Questions

1. After installing the required packages, run the following code. Provide a short description of what each line of code does, **only for the lines that have a #? at the end, and not for the ones mentioning `#OK`**. Not everything has been taught in the class, you will have to figure out what each line is doing by the outcome produced and by using internet search. Be short and concise in your answers. Put the answers in your .R file as comments, before the respective code line. Do not bother with figuring out `html` technicalities or the $\%>\%$ operators, just refer to R-related outcome, as in, what is the product of each line? Some examples are provided in the first lines. You may ignore the warnings produced. (3 points)

```
library(robotstxt)
library(rvest)

# ADD HERE A SHORT DESCRIPTION OF WHAT THE WHOLE CODE DOES

# Check whether scraping is allowed from this webpage (returns TRUE)
# ATTENTION: PUT THE WHOLE URL IN ONE LINE WHEN RUNNING THE CODE
paths_allowed("https://www.metacritic.com/publication/washington-post?filter=
              movies&num_items=100&sort_options=date&page=0")

# Define character element "main.page", to be used recursively for defining
# multiple pages from metacritic.com
# ATTENTION: PUT THE WHOLE URL IN ONE LINE WHEN RUNNING THE CODE
main.page <- "https://www.metacritic.com/publication/washington-post?filter=
              movies&num_items=100&sort_options=date&page="

for (i in 0:27){  # This is a "for" loop.
                  # This means that all the lines until the closure of }
                  # will be repeated for different values of object i
                  # thus, on the first run i=0, second run i=1,... last run i=27
```

*e:sknikolak@aueb.gr

```
# for each step, define...
  step.page <- paste(main.page,i,sep="") # ?

  webdata <-read_html(step.page) # OK

# Vector ... is created which includes .....
  title <-c(webdata %>% html_nodes("div.review_product") %>% html_nodes("a") %>%
              html_text())  #?

  metascore <- c(webdata %>% html_nodes("li.review_product_score.brief_metascore") %>%
                   html_nodes("span.metascore_w") %>% html_text()) #?

  critic <- c(webdata %>% html_nodes("li.review_product_score.brief_critscore") %>%
                html_nodes("span.metascore_w") %>% html_text()) #?

  date <- c(webdata %>% html_nodes("li.review_action.post_date") %>% html_text()) #?

  if (length(date)<100 ){for (j in length(date):100){ date[j] <- date[length(date)]}} #OK

  a <- substr(date,12,13) #?
  b <- substr(date,8,10)  #?
  d <- substr(date,16,19) #?

  date2 <- apply(cbind(a,b,d),1,paste,collapse="/") #?
  date3 <- as.Date(date2,"%d/%b/%Y")  #?

  df = data.frame(title,metascore,critic,date3) #?

  colnames(df) <- c("title", "metascore", "critic","date") #?

  df$metascore <- as.numeric(as.character(df$metascore)) #?
  df$critic <- as.numeric(as.character(df$critic)) #?

  df <- df[complete.cases(df), ] #?

  if (i==0){       #OK
    df.tot <- df}  #OK

  if (i>0){        #?
    df.tot <- rbind(df.tot,df) } #?

}

df.tot$title <- as.character(df.tot$title) #?
```

NOTE: FOR THE SOLUTIONS TO THE QUESTIONS BELOW, NO USE OF ADDITIONAL PACKAGE IS ALLOWED

2. You will now work with the `df.tot` data frame. Provide a short description of the data (hint: `str`). (0.5 points)

3. Create three new variables that are directly included in the data frame **(assign them directly as variables of the `df.tot` data frame)**. These variables should describe (where `(xxxx)` the name of the created variable): (2 points)

   - The ratio of public score / critics score for each movie `(ratio)`
   - The percentile of each `metascore` value `(perc.meta)` (hint: `rank()`)
   - The percentile of each `critic` value `(perc.critic)`

- The year each film was reviewed `(year)`

4. Which film has the highest `metascore` score? (0.5 points)

5. Produce a boxplot of the `(perc.meta)` variable, for (faceted by) each year observed in the dataset (in the same plot window). Draw a vertical line at `y=0.5` and discuss the result. (2 points)

6. Some of the `ratio` values are infinity. Explain why this is happening and create a new data frame, named `df.tot2`, which does not include these observations (0.5 points).

7. Work with the `df.tot2` data frame. Create a matrix with two columns, one with the `metascore` and one with the `critic`. Calculate a vector that includes the average of the two, by using the `apply()` function. (1 point)

8. Work with the `df.tot2` data frame. Create a scatterplot with `date` on the x-axis and `perc.meta` on the y-axis. Main title should be "Metascores percentiles" and the axes named accordingly. Colour the dots according to whether the observation has a `metascore`>50 or not. Add a vertical dashed line for `metascore`=50. Make the y-axis labels to be perpendicular to the axis. (2 points)

9. Comment on the above graph, taking into account the possible range of values for `metascore` (0-100) (1 point).