Final Exam

Adi Sarid / adi@sarid-ins.co.il

April 2019

Exam instructions

The following exam contains 10 questions which spans across the different topics we learned during our course. Each question has four options but only one answer. Each correct answer will provide you with 10 points.

You have one hour to answer this exam. You can use any materials you want including but not limited to: cheat sheets, our course materials, stack overflow, package documentation, running code and seeing what it does.

You have 60 minutes to complete the exam.

Question 1 (10 points):

When would you use an R markdown file (.Rmd) versus a script file (.R) to save your work?

a. If I want the relative position of the file retained (so that it is easier to load files from the same directory), I will use an .Rmd file, otherwise I will use a .R file.
b. When I want a complete documentation of my work in a report I will use a .Rmd. I will use a .R file for debugging and sourcing functions.
c. There is no significant difference between the two formats, and they can be used for the same things interchangably.
d. There is no benefit to using .R script files, the .Rmd format is always superior.

Question 2 (10 points):

Look at the following segments of code.

```
# segment 1:
new_data <- read.csv("myfilename.csv")

# segment 2:
new_data %>%
   group_by(some_cool_suff) %>%
   summarize(average = mean(avg_me, na.rm = T)) -> updated_df

# segment 3:
avg_var <- mean(new_data$avg_me[!is.na(some_cool_stuff)], na.rm = T)

# segment 4:
data.frame(a = 1:10, b = letters[1:10]) %>%
   sample_n(3)
```

Which segments would you classify as `tidyverse` syntax? (`tidyverse` syntax = code which uses functions from tidyverse packages, in which there is no function that you can replace to a `tidyverse` equivalent)

a. Segment 1 and segment 3.
b. Segment 2 and segment 4.
c. Segment 4.

d. Segment 2.

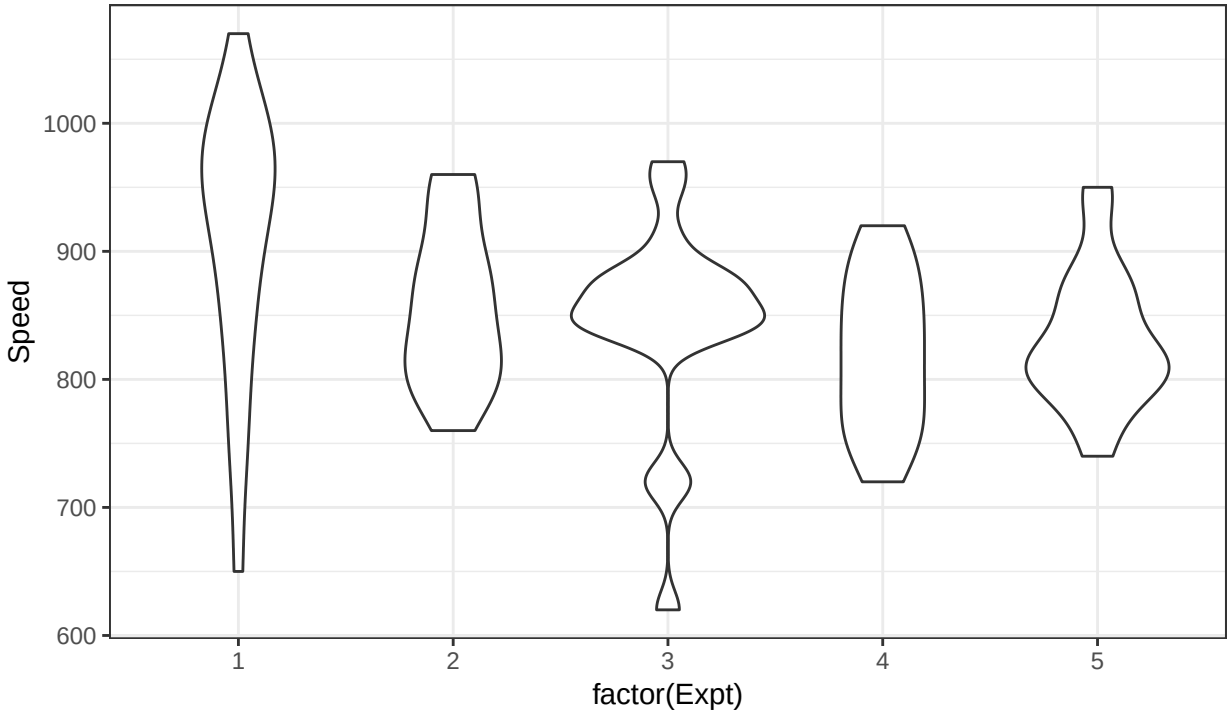
Question 3 (10 points):

What `ggplot2` geoms would you use to generate the following charts?

a. Figure 1: not generated with `ggplot2`, Figure 2: `geom_point`.
b. Figure 1: `geom_boxplot`, Figure 2: `geom_line`.
c. Figure 1: `geom_violin`, Figure 2: `geom_point`.
d. Figure 1: `geom_boxplot`, Figure 2: `geom_point` + `geom_line`.

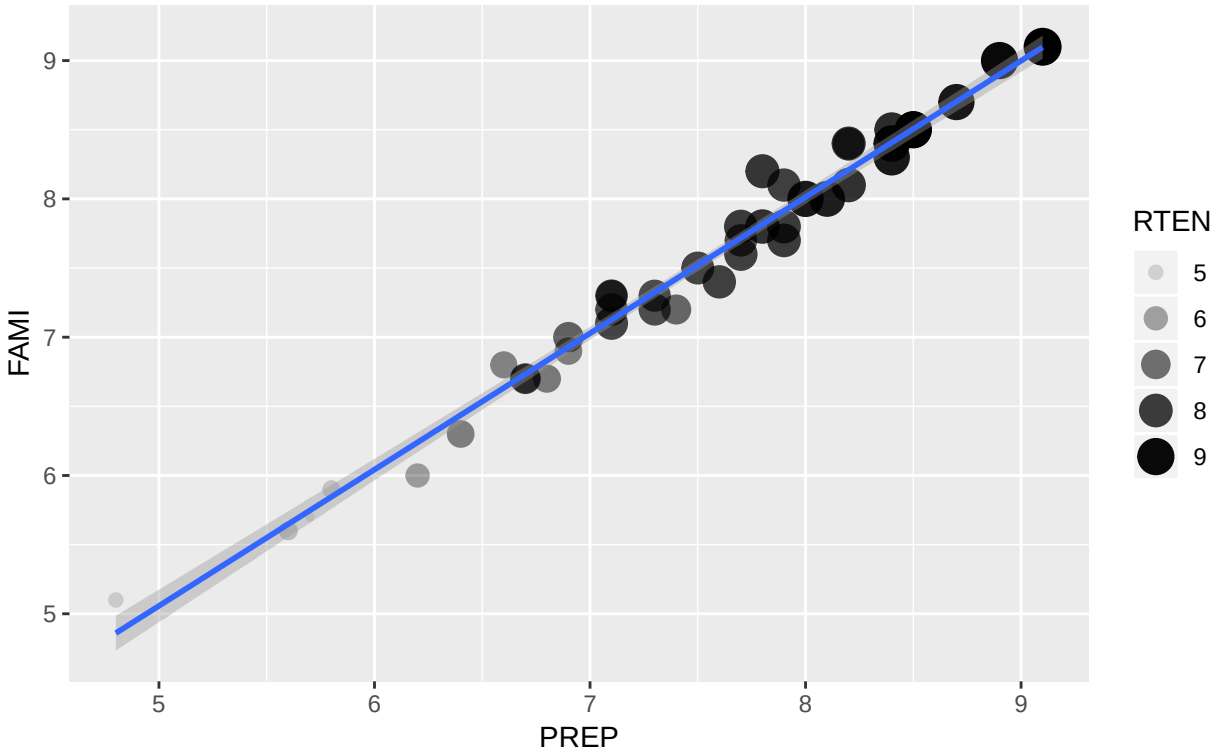
Figure 1: Michelson Speed of Light Data

Speed of light measurement experiment [km/sec], 299000 subtracted



Source: moerley from package datasets, see ?morley for information

Figure 2: Lawyer's rating of state judgets in the US superior court



Source: USJusdgeRatings from package datasets, see ?USJudgeRatings for information

Question 4 (10 points):

What is the difference between the `matrix` and the `tibble` in the following?

```
matrix(cbind(1:10, letters[1:10], LETTERS[1:10]), ncol = 3)
tibble(num = 1:10, sl = letters[1:10], cl = LETTERS[1:10])
```

a. The tibble has named variables (columns) and the matrix does not name the columns.
b. The tibble retains the original data type and the matrix converts the data types.
c. `matrix` is a base R function and `tibble` is a tidyverse function.
d. All of the above.

Question 5 (10 points):

What `stringr` function would you use to simplify the following code?

```
some_string <- c("How are you today?", "Is this test ok?", "You're already half way in!")
map_chr(some_string, ~paste0(stringi::stri_wrap(., width = 5), collapse = "\n"))
```

a. `str_count`.
b. `str_wrap`.
c. `str_sub`.
d. No such function: must use a combination of a `stringr` and a loop (or a `map` function).

Question 6 (10 points):

What is the difference between `contains` and `one_of`?

a. Both are "select helpers", `one_of` is used to specify strings which starts with one of the specified expressions, and `contains` lets you specify the variable names in "non standard evaluation" (unquoted) style.
b. `contains` selects variables based on the regular expression you feed as an argument. `one_of` needs you to specify the variable names as strings.
c. `contains` selects variables which contain the literal string you feed into it. `one_of` needs you to specify the variables names as strings.
d. Both functions do the same thing with the same arguments.

Question 7 (10 points):

When reshaping data with the `gather` function, what is the meaning of the `...` argument?

a. Specify which variables to gather by.
b. Specify which variables **not** to gather by (using the "-" sign).
c. Specify either *a* or *b*.
d. Provide variable by which to group the resulting tibble.

Question 8 (10 points):

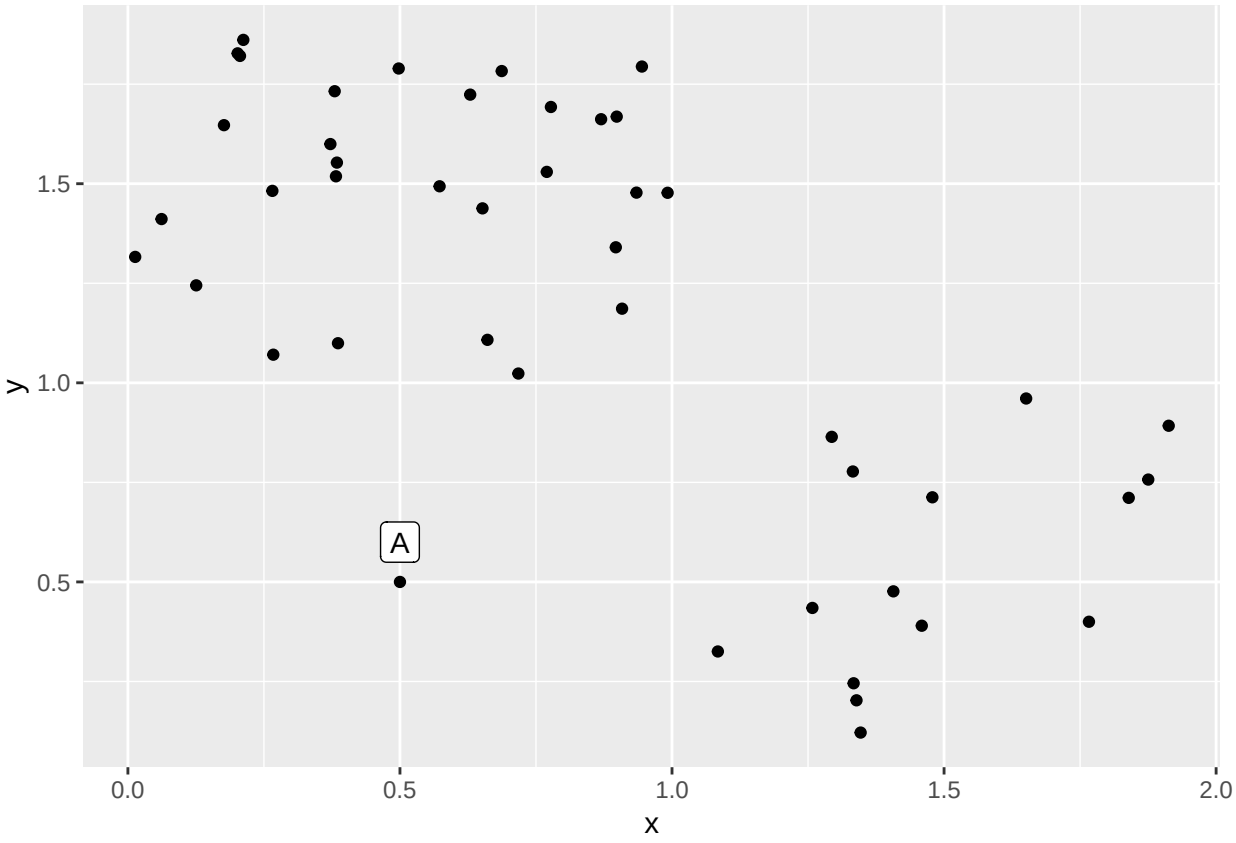
What function would you use to get all the rows in `tibble1` which are not in `tibble2`?

a. `setdiff(tibble1, tibble2)`
b. `setdiff(tibble2, tibble1)`
c. `intersect(tibble1, tibble2)`
d. `semi_join(tibble1, tibble2)`

Question 9 (10 points):

Assume you examine the data which appears in the following scatter plot using per-axis boxplots. Would classify point A as an outlier?

a. Yes, only accoring to the y-axis.
b. Yes, only according to the x-axis.
c. Yes, according to either x-axis or y-axis.
d. No, it will not be classified as an outlier.



Question 10 (10 points):

You encountered a data set in which all variables are normally distributed with an unequal variance and unequal expectancy (mean). You wish to run a KMeans clustering to cluster the data. What would you do as a preprocessing step?

a. Scale and center the data using the function `scale`.
b. Scale and center the data using min-max scaling and centering.
c. Either a or b.
d. Nothing - since the data is already normally distributed, no scaling or centering is required.

Bonus question (5 points bonus):

Did you sign up for R-Bloggers updates? (feed to receive R related news and updates)

a. Yes (5 points bonus).

b. No, but I'm doing it now (2.5 points bonus).
c. No, and I don't intend to.