Challenge-2

Soh Li Ying

2023-08-21

Welcome! Hope you have watched the lecture videos and followed the instructions in code-along. Go through the steps described below, *carefully*. It is totally fine to get stuck - ASK FOR HELP; reach out to your friends, TAs, or the discussion forum on Canvas.

Here is what you have to do,

1. Pair with a neighbor and work
2. Download the Challenge-2.Rmd and playlist\_data.csv files from Canvas
3. Move the downloaded files to the folder, “Week-2”
4. Set it as the working directory
5. Edit content wherever indicated
6. Remember to set eval=TRUE after completing the code to generate the output
7. Ensure that echo=TRUE so that the code is rendered in the final document
8. Inform the tutor/instructor upon completion
9. Submit the document on Canvas after they approve
10. Attendance will be marked only after submission
11. Once again, do not hesitate to reach out to the tutors/instructor, if you are stuck

# I. Exploring music preferences

### A. Background

Imagine that you have been hired as a data analyst by a radio station to analyze music preferences of their DJs. They have provided you with a dataset, playlist\_data.csv, containing information about DJs, their preferred music genres, song titles, and ratings.

Using the data-set you are required to complete some tasks that are listed subsequently. All these tasks are based on the concepts taught in the video lectures. The questions may not be entirely covered in the lectures; To complete them, you are encouraged to use Google and the resources therein.

### B.Tasks

#### Task-1

In the lecture, we used two data-sets, starwars and anscombe's quartet that were readily available with the packages, tidyverse and Tmisc, respectively. When we have to use custom-made data-sets or the ones like we downloaded from Canvas, we have to import it using the R commands before using them. All the questions below are related to this task.

**Question 1.1:** What does the term “CSV” in playlist\_data.csv stand for, and why is it a popular format for storing tabular data?

**Solution:** CSV stands for Comma-Separated Values. It is a popular format for storing tabular data due to its simplicity in editing and creating using basic text editors. It is also easy to use since it is compatible with most spreadsheet programs, such as Microsoft Excel or Google Spreadsheets. Since CSV files are plain-text files, they are easier to be imported into a spreadsheet or another data storage database. CSV files are also human readable (data is not encoded or converted to binary before storing), hence increasing the readability and ease of manipulation. **Question 1.2:** load the tidyverse package to work with .csv files in R.

**Solution:**

# Load the necessary package to work with CSV files in R.  
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.2 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ lubridate 1.9.2 ✔ tibble 3.2.1  
## ✔ purrr 1.0.2 ✔ tidyr 1.3.0  
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

**Question 1.3:** Import the data-set, playlist\_data.csv

**Solution:**

# Import the "playlist\_data.csv" dataset into R   
  
read\_csv("playlist\_data.csv")

## Rows: 26 Columns: 7  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (4): DJ\_Name, Music\_Genre, Experience, Location  
## dbl (3): Rating, Age, Plays\_Per\_Week  
##   
## 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.

## # A tibble: 26 × 7  
## DJ\_Name Music\_Genre Rating Experience Age Location Plays\_Per\_Week  
## <chr> <chr> <dbl> <chr> <dbl> <chr> <dbl>  
## 1 DJ A Pop 4.2 Advanced 28 City X 80  
## 2 DJ B Rock 3.8 Intermediate 24 City Y 60  
## 3 DJ C Electronic 4.5 Advanced 30 City Z 100  
## 4 DJ D Pop 4 Intermediate 22 City X 70  
## 5 DJ E Electronic 4.8 Advanced 27 City Y 90  
## 6 DJ F Rock 3.6 Intermediate 25 City Z 55  
## 7 DJ G Pop 4.3 Advanced 29 City X 85  
## 8 DJ H Electronic 4.1 Intermediate 23 City Y 75  
## 9 DJ I Rock 3.9 Advanced 31 City Z 70  
## 10 DJ J Pop 4.4 Intermediate 26 City X 95  
## # ℹ 16 more rows

**Question 1.4:** Assign the data-set to a variable, playlist\_data

**Solution:**

# Assign the variable to a dataset   
  
playlist\_data <- read\_csv("playlist\_data.csv")

## Rows: 26 Columns: 7  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (4): DJ\_Name, Music\_Genre, Experience, Location  
## dbl (3): Rating, Age, Plays\_Per\_Week  
##   
## 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.

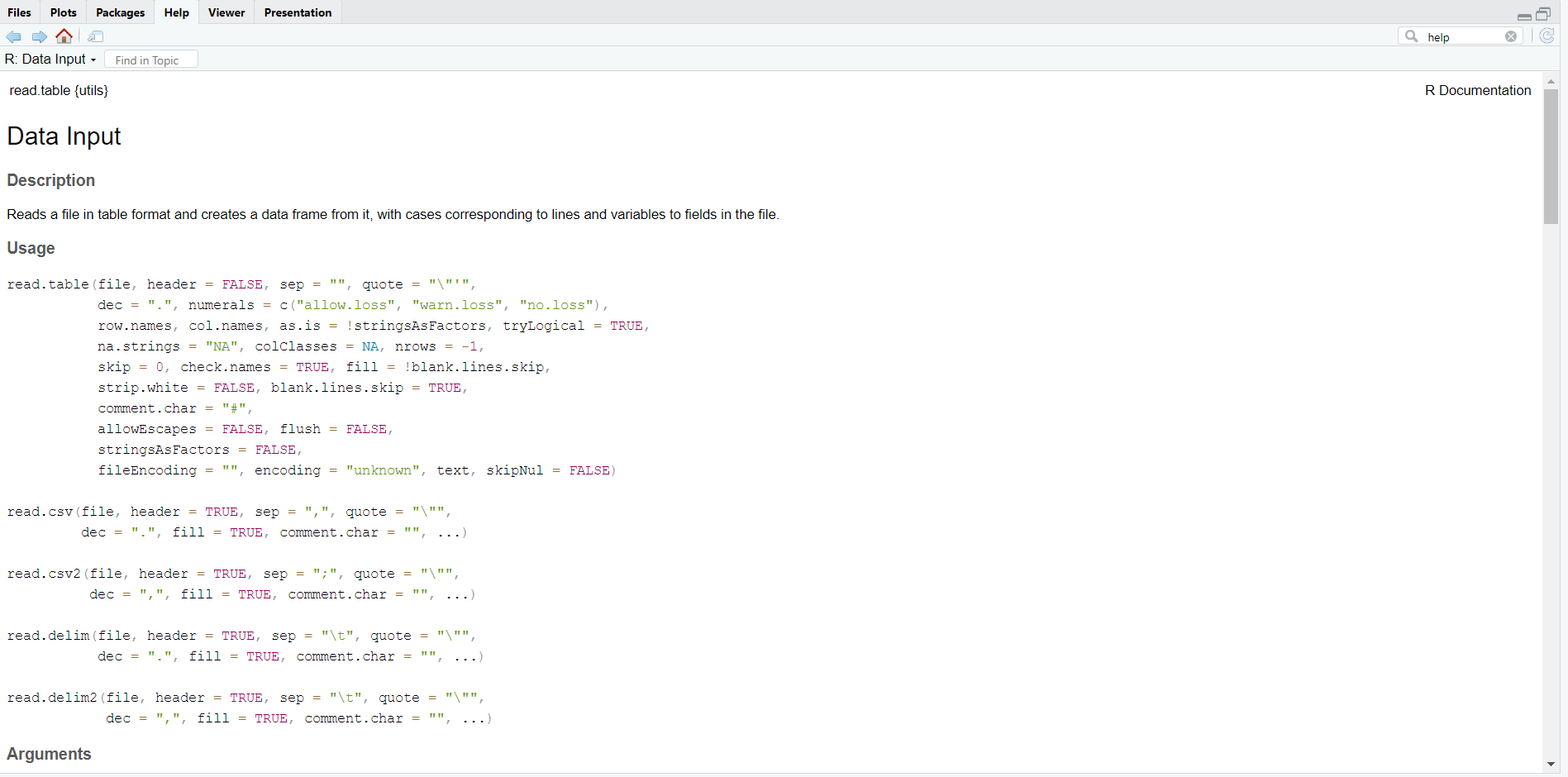
*From now on, you can use the name of the variable to view the contents of the data-set*

**Question 1.5:** Get more information about read\_csv() command and provide a screenshot of the information displayed in the “Help” tab of the “Files” pane

**Solution:**

# More information about the R command  
?read\_csv()

## starting httpd help server ... done



**Question 1.6:** What does the skip argument in the read\_csv() function do?

**Solution:** The skip argument in the read\_csv function ignores a specific number of rows inside the data. For example, to skip the first 15 rows of the data set, the code is as follows:

read\_csv("playlist\_data", skip = 15)

**Question 1.7:** Display the contents of the data-set

**Solution:**

# Type the name of the variable, to see what it contains  
playlist\_data

## # A tibble: 26 × 7  
## DJ\_Name Music\_Genre Rating Experience Age Location Plays\_Per\_Week  
## <chr> <chr> <dbl> <chr> <dbl> <chr> <dbl>  
## 1 DJ A Pop 4.2 Advanced 28 City X 80  
## 2 DJ B Rock 3.8 Intermediate 24 City Y 60  
## 3 DJ C Electronic 4.5 Advanced 30 City Z 100  
## 4 DJ D Pop 4 Intermediate 22 City X 70  
## 5 DJ E Electronic 4.8 Advanced 27 City Y 90  
## 6 DJ F Rock 3.6 Intermediate 25 City Z 55  
## 7 DJ G Pop 4.3 Advanced 29 City X 85  
## 8 DJ H Electronic 4.1 Intermediate 23 City Y 75  
## 9 DJ I Rock 3.9 Advanced 31 City Z 70  
## 10 DJ J Pop 4.4 Intermediate 26 City X 95  
## # ℹ 16 more rows

**Question 1.8:** Assume you have a CSV file named sales\_data.csv containing information about sales transactions. How would you use the read\_csv() function to import this file into R and store it in a variable named sales\_data?

**Solution:**

# No output is required for this code  
# Only the list of commands that execute the task mentioned in the question are required  
  
read\_csv("C:/Users/65946/Downloads/NM2207/Week-2/sales\_data.csv")  
sales\_data <- read\_csv("C:/Users/65946/Downloads/NM2207/Week-2/sales\_data.csv")

#### Task-2

After learning to import a data-set, let us explore the contents of the data-set through the following questions

**Question 2.1:** Display the first few rows of the data-set to get an overview of its structure

**Solution:**

# Type the name of the variable we assigned the data-set to  
head(playlist\_data)

## # A tibble: 6 × 7  
## DJ\_Name Music\_Genre Rating Experience Age Location Plays\_Per\_Week  
## <chr> <chr> <dbl> <chr> <dbl> <chr> <dbl>  
## 1 DJ A Pop 4.2 Advanced 28 City X 80  
## 2 DJ B Rock 3.8 Intermediate 24 City Y 60  
## 3 DJ C Electronic 4.5 Advanced 30 City Z 100  
## 4 DJ D Pop 4 Intermediate 22 City X 70  
## 5 DJ E Electronic 4.8 Advanced 27 City Y 90  
## 6 DJ F Rock 3.6 Intermediate 25 City Z 55

**Question 2.2:** Display all the columns of the variable stacked one below another

**Solution:**

# Stack columns of playlist\_data  
glimpse(playlist\_data)

## Rows: 26  
## Columns: 7  
## $ DJ\_Name <chr> "DJ A", "DJ B", "DJ C", "DJ D", "DJ E", "DJ F", "DJ G",…  
## $ Music\_Genre <chr> "Pop", "Rock", "Electronic", "Pop", "Electronic", "Rock…  
## $ Rating <dbl> 4.2, 3.8, 4.5, 4.0, 4.8, 3.6, 4.3, 4.1, 3.9, 4.4, 4.6, …  
## $ Experience <chr> "Advanced", "Intermediate", "Advanced", "Intermediate",…  
## $ Age <dbl> 28, 24, 30, 22, 27, 25, 29, 23, 31, 26, 32, 28, 29, 25,…  
## $ Location <chr> "City X", "City Y", "City Z", "City X", "City Y", "City…  
## $ Plays\_Per\_Week <dbl> 80, 60, 100, 70, 90, 55, 85, 75, 70, 95, 110, 75, 60, 8…

**Question 2.3:** How many columns are there in the dataset?

**Solution:**

# Number of columns  
ncol(playlist\_data)

## [1] 7

**Question 2.4:** What is the total count of DJs?

**Solution:**

# Number of DJs  
num\_DJs <- nrow(playlist\_data)  
num\_DJs

## [1] 26

**Question 2.5:** Display all the location of all the DJs

**Solution:**

# Location of DJs  
playlist\_data %>% select(DJ\_Name,Location)

## # A tibble: 26 × 2  
## DJ\_Name Location  
## <chr> <chr>   
## 1 DJ A City X   
## 2 DJ B City Y   
## 3 DJ C City Z   
## 4 DJ D City X   
## 5 DJ E City Y   
## 6 DJ F City Z   
## 7 DJ G City X   
## 8 DJ H City Y   
## 9 DJ I City Z   
## 10 DJ J City X   
## # ℹ 16 more rows

**Question 2.6:** Display the age of the DJs

**Solution:**

# Age of DJs  
playlist\_data %>% select(DJ\_Name, Age)

## # A tibble: 26 × 2  
## DJ\_Name Age  
## <chr> <dbl>  
## 1 DJ A 28  
## 2 DJ B 24  
## 3 DJ C 30  
## 4 DJ D 22  
## 5 DJ E 27  
## 6 DJ F 25  
## 7 DJ G 29  
## 8 DJ H 23  
## 9 DJ I 31  
## 10 DJ J 26  
## # ℹ 16 more rows

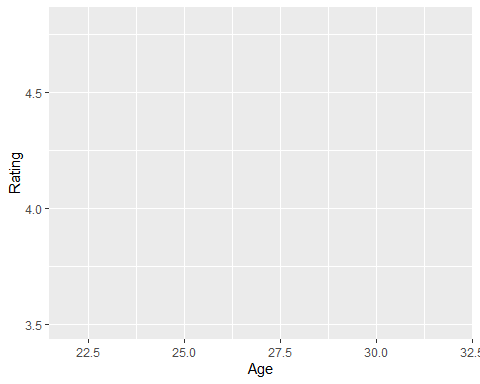
#### Task-3

Let us plot the data to get more insights about the DJs.

**Question 3.1:** Create a plot to visualize the relationship between DJs’ ages and their ratings.

**Solution:**

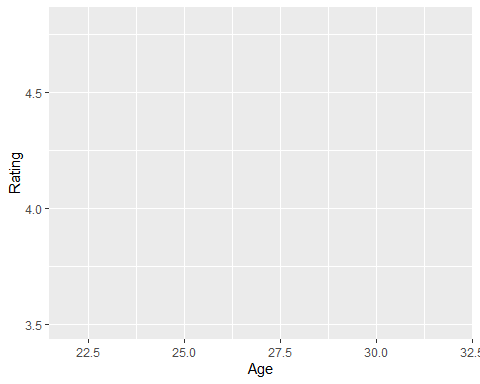
# complete the code to generate the plot  
  
ggplot(playlist\_data) + aes(x=Age,y=Rating)



**Question 3.2:** Label the x-axis as “Age” and the y-axis as “Rating.”

**Solution:** d

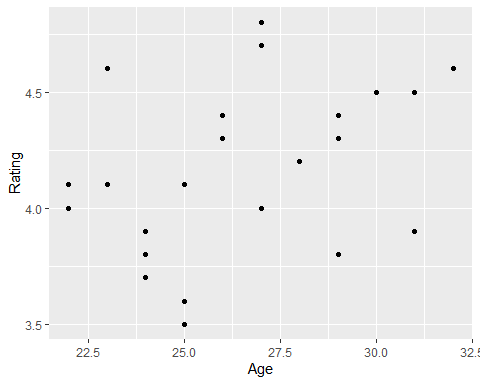
# complete the code to generate the plot  
  
ggplot(playlist\_data) + aes(x=Age,y=Rating) +   
 labs(x="Age", y="Rating")



**Question 3.3:** Represent data using points

**Solution:**

# complete the code to generate the plot  
  
ggplot(playlist\_data) + aes(x=Age,y=Rating) + geom\_point() +  
 labs(x="Age", y="Rating")



**Question 3.4:** Can you change the points represented by dots/small circles to any other shape of your liking?

**Solution:**

# complete the code to generate the plot  
  
# <-- Hint: Use ? to learn more about geom\_point and use appropriate values for shape  
  
# ggplot(playlist\_data) + aes(x=Age,y=Rating) + geom\_point() +  
# labs(x="Age", y="Rating")  
  
  
#ggplot(playlist\_data) + aes(x=Age,y=Rating) + geom\_point(shape=21) + labs(x="Age", y="Rating")

**Question 3.5:** Insert a suitable title and briefly provide your insights in the caption

**Solution:**

# complete the code to generate the plot  
  
#?geom\_point()  
  
ggplot(playlist\_data) + aes(x=Age,y=Rating) + geom\_point(shape = 2, size = 3, colour = "black"  
, fill = "white")+  
 labs(x="Age", y="Rating",   
 title = "Age versus Rating",  
 caption = "There is no direct relationship between the age and rating of a DJ")

