Challenge-2

Tang Ching Xian

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 simple and widely used file format for storing tubular data such as spreadsheet and databases. In each CSV file, each line represents a row do data within each line, individual values for each column are separated by commas.

Question 1.2: load the `tidyverse` package to work with `.csv` files in R commands.

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
## ✓ ggplot2   3.4.3     ✓ tibble    3.2.1
## ✓ lubridate 1.9.2     ✓ tidyr     1.3.0
## ✓ purrr     1.0.2
## ── 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")
```

```
##    DJ_Name Music_Genre Rating   Experience Age Location Plays_Per_Week
## 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    4.0 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
## 11    DJ K     Hip-Hop    4.6     Advanced  32   City Y            110
## 12    DJ L  Electronic    4.2 Intermediate  28   City Z             75
## 13    DJ M         Pop    3.8     Advanced  29   City X             60
## 14    DJ N        Rock    4.1 Intermediate  25   City Y             80
## 15    DJ O  Electronic    4.5     Advanced  31   City Z             95
## 16    DJ P     Hip-Hop    4.3 Intermediate  26   City X            105
## 17    DJ Q         Pop    4.0     Advanced  27   City Y             70
## 18    DJ R        Rock    3.7 Intermediate  24   City Z             50
## 19    DJ S  Electronic    4.4     Advanced  29   City X             85
## 20    DJ T     Hip-Hop    4.6 Intermediate  23   City Y            100
## 21    DJ U         Pop    4.2     Advanced  28   City Z             80
## 22    DJ V        Rock    3.9 Intermediate  24   City X             60
## 23    DJ W  Electronic    4.5     Advanced  30   City Y            100
## 24    DJ X         Pop    4.1 Intermediate  22   City Z             70
## 25    DJ Y  Electronic    4.7     Advanced  27   City X             90
## 26    DJ Z        Rock    3.5 Intermediate  25   City Y             55
```

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")
```

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

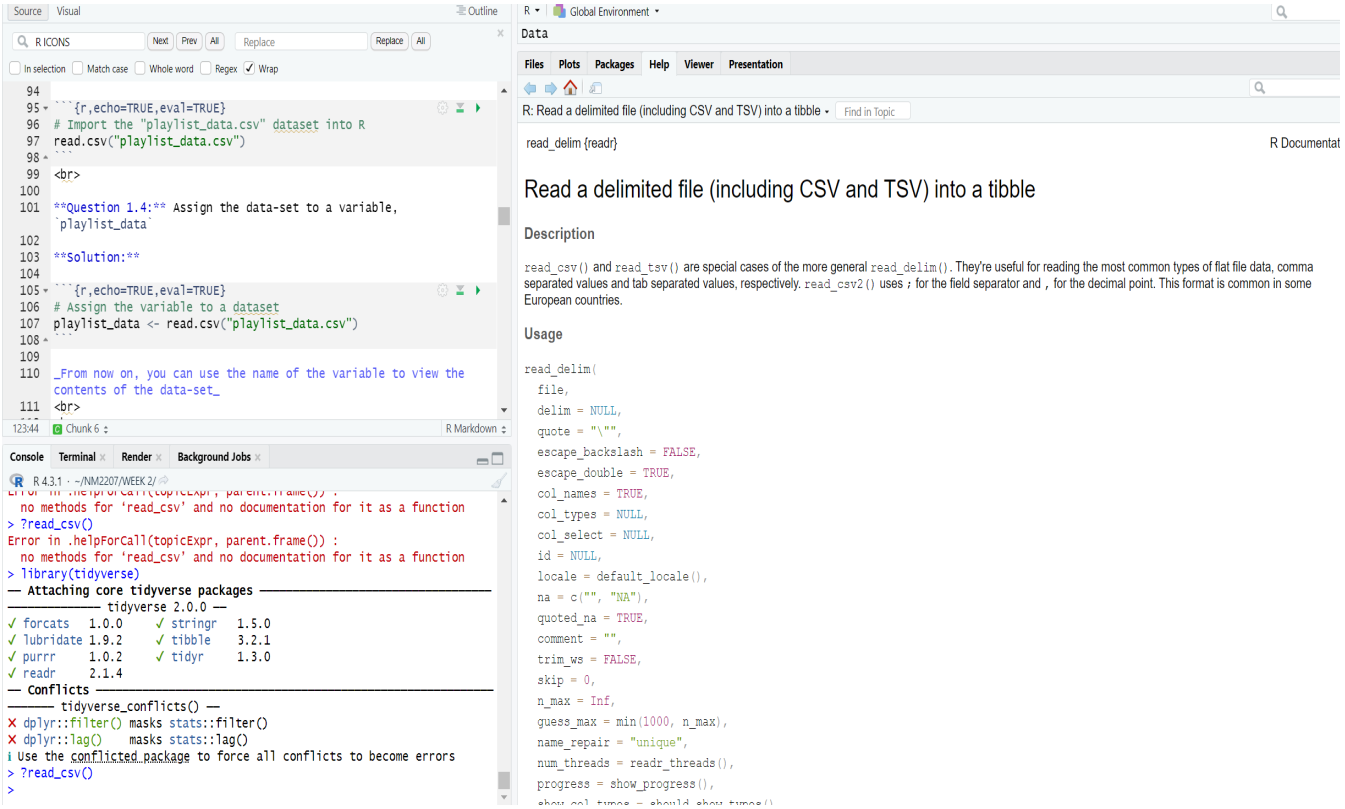
Question 1.5: Get more information about `read_csv()` command and provide a screenshot of the information displayed in the “Help” tab of the “Files” pane

Solution:

```
# More information about the R command, complete the code
?read.csv()
```

```
## starting httpd help server ... done
```

```
knitr::include_graphics("wk2p1.png")
```



View Help

Question 1.6: What does the `skip` argument in the read_csv() function do?

Solution: The skip argument specifies the number of lines to skip before starting to read data from the file. This is useful when you have header information, comments, or other irrelevant lines at the beginning of your file that you want to skip over. Use “data <- read.table(”data.txt”, skip = 10)” in R.

Question 1.7: Display the contents of the data-set

Solution:

```
# Type the name of the variable, to see what it contains
playlist_data
```

```
##    DJ_Name Music_Genre Rating   Experience Age Location Plays_Per_Week
## 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    4.0 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
## 11    DJ K     Hip-Hop    4.6     Advanced  32   City Y            110
## 12    DJ L  Electronic    4.2 Intermediate  28   City Z             75
## 13    DJ M         Pop    3.8     Advanced  29   City X             60
## 14    DJ N        Rock    4.1 Intermediate  25   City Y             80
## 15    DJ O  Electronic    4.5     Advanced  31   City Z             95
## 16    DJ P     Hip-Hop    4.3 Intermediate  26   City X            105
## 17    DJ Q         Pop    4.0     Advanced  27   City Y             70
## 18    DJ R        Rock    3.7 Intermediate  24   City Z             50
## 19    DJ S  Electronic    4.4     Advanced  29   City X             85
## 20    DJ T     Hip-Hop    4.6 Intermediate  23   City Y            100
## 21    DJ U         Pop    4.2     Advanced  28   City Z             80
## 22    DJ V        Rock    3.9 Intermediate  24   City X             60
## 23    DJ W  Electronic    4.5     Advanced  30   City Y            100
## 24    DJ X         Pop    4.1 Intermediate  22   City Z             70
## 25    DJ Y  Electronic    4.7     Advanced  27   City X             90
## 26    DJ Z        Rock    3.5 Intermediate  25   City Y             55
```

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 command 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
Sales_data <- read_csv("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)
```

```
##   DJ_Name Music_Genre Rating   Experience Age Location Plays_Per_Week
## 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    4.0 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            <int> 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 <int> 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:

```
ncol(playlist_data)
```

```
## [1] 7
```

Question 2.4: What is the total count of DJs?

Solution:

```
playlist_data$DJ_Name
```

```
##  [1] "DJ A" "DJ B" "DJ C" "DJ D" "DJ E" "DJ F" "DJ G" "DJ H" "DJ I" "DJ J"
## [11] "DJ K" "DJ L" "DJ M" "DJ N" "DJ O" "DJ P" "DJ Q" "DJ R" "DJ S" "DJ T"
## [21] "DJ U" "DJ V" "DJ W" "DJ X" "DJ Y" "DJ Z"
```

```
length(playlist_data$DJ_Name)
```

```
## [1] 26
```

Question 2.5: Display all the location of all the DJs

Solution:

```
filter_columns <- c(playlist_data$DJ_Name)
playlist_data %>% filter(DJ_Name%in%filter_columns) %>% select(DJ_Name,Location)
```

```
##    DJ_Name Location
## 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
## 11    DJ K   City Y
## 12    DJ L   City Z
## 13    DJ M   City X
## 14    DJ N   City Y
## 15    DJ O   City Z
## 16    DJ P   City X
## 17    DJ Q   City Y
## 18    DJ R   City Z
## 19    DJ S   City X
## 20    DJ T   City Y
## 21    DJ U   City Z
## 22    DJ V   City X
## 23    DJ W   City Y
## 24    DJ X   City Z
## 25    DJ Y   City X
## 26    DJ Z   City Y
```

Question 2.6: Display the age of the DJs

Solution:

```
playlist_data %>% filter(DJ_Name%in%filter_columns) %>% select(Age)
```

```
##    Age
## 1   28
## 2   24
## 3   30
## 4   22
## 5   27
## 6   25
## 7   29
## 8   23
## 9   31
## 10  26
## 11  32
## 12  28
## 13  29
## 14  25
## 15  31
## 16  26
## 17  27
## 18  24
## 19  29
## 20  23
## 21  28
## 22  24
## 23  30
## 24  22
## 25  27
## 26  25
```

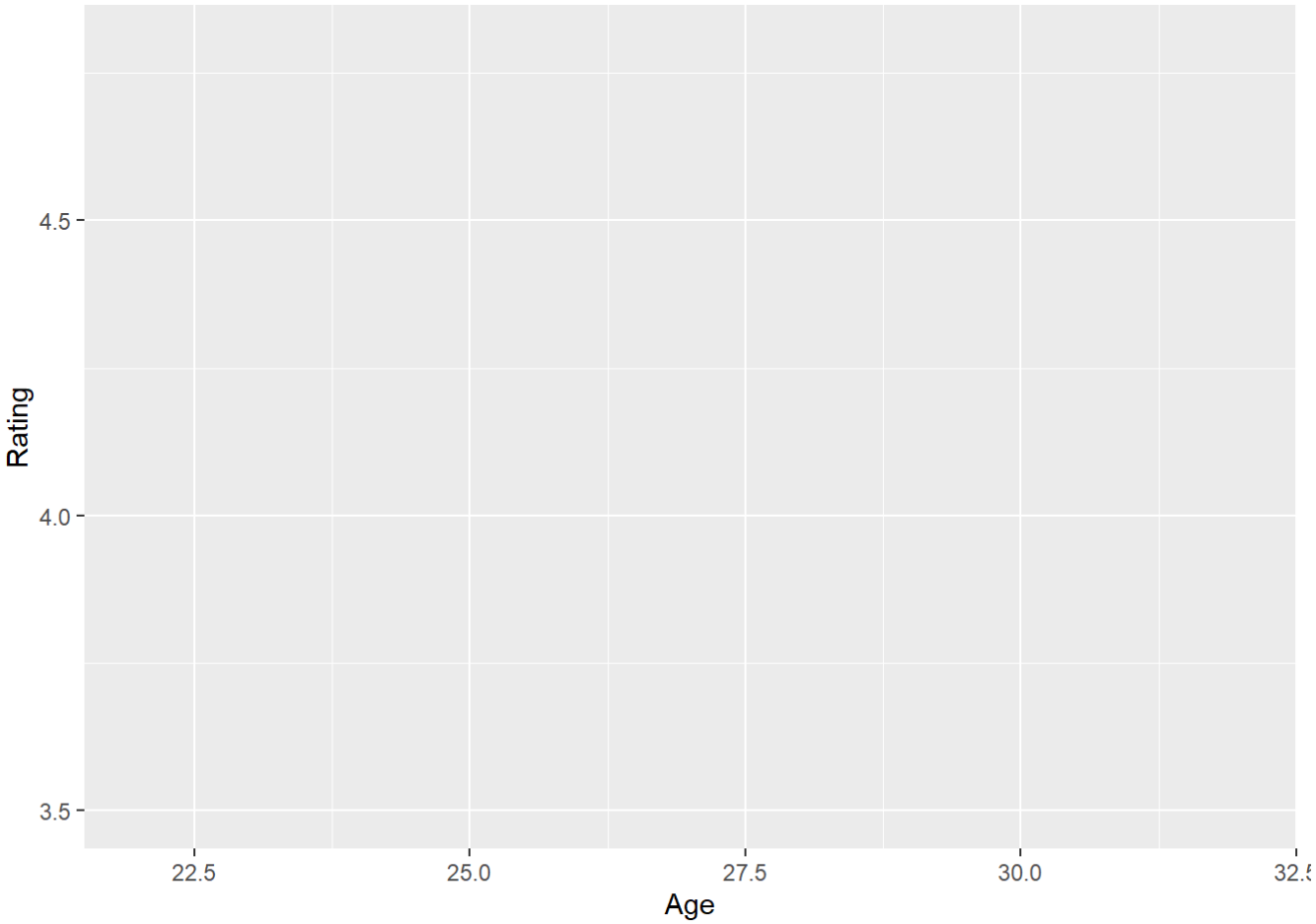
Task-3

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:

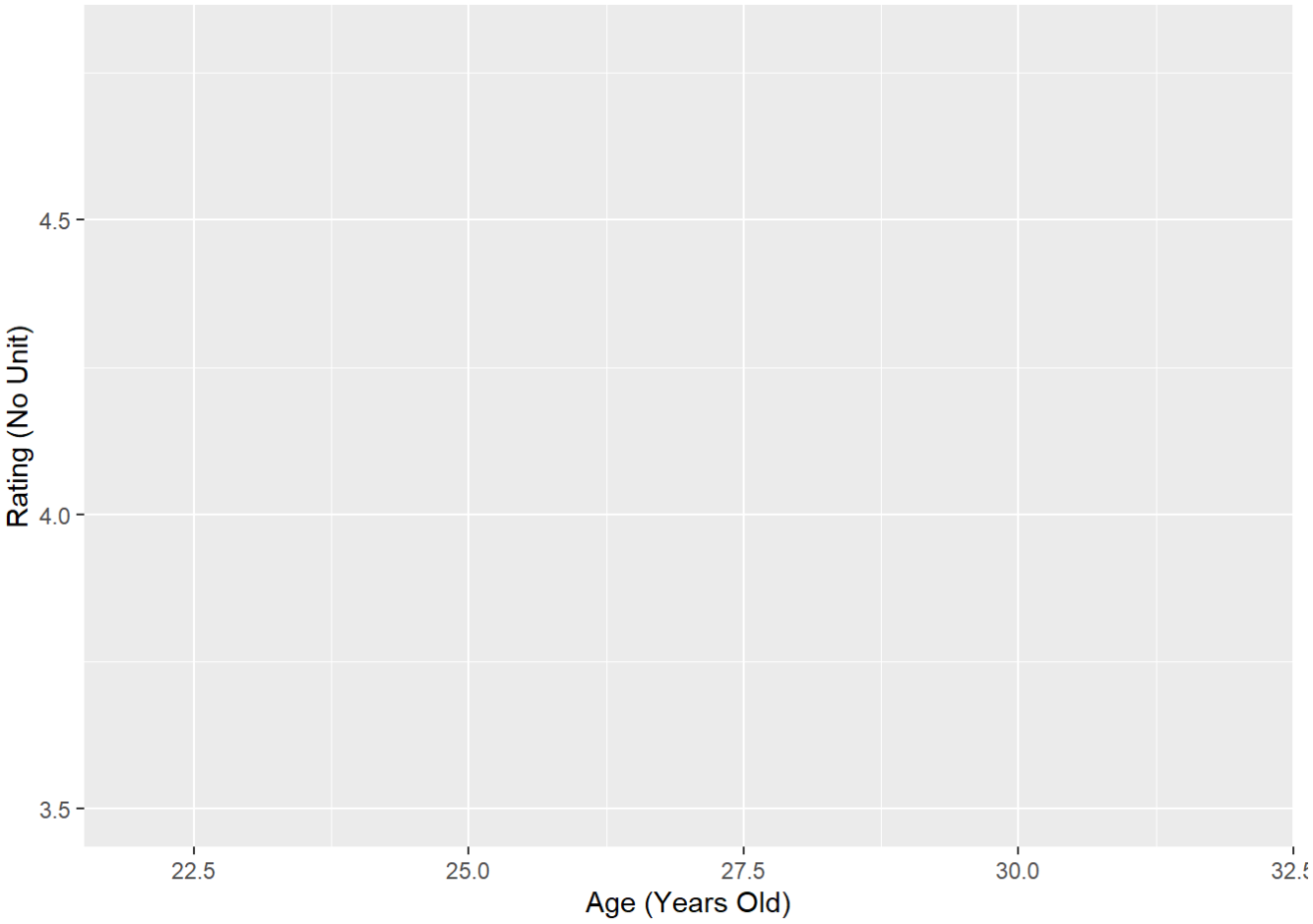
```
# complete the code to generate the plot

ggplot(data=playlist_data) + aes(x=Age,y=Rating)
```



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

Solution:

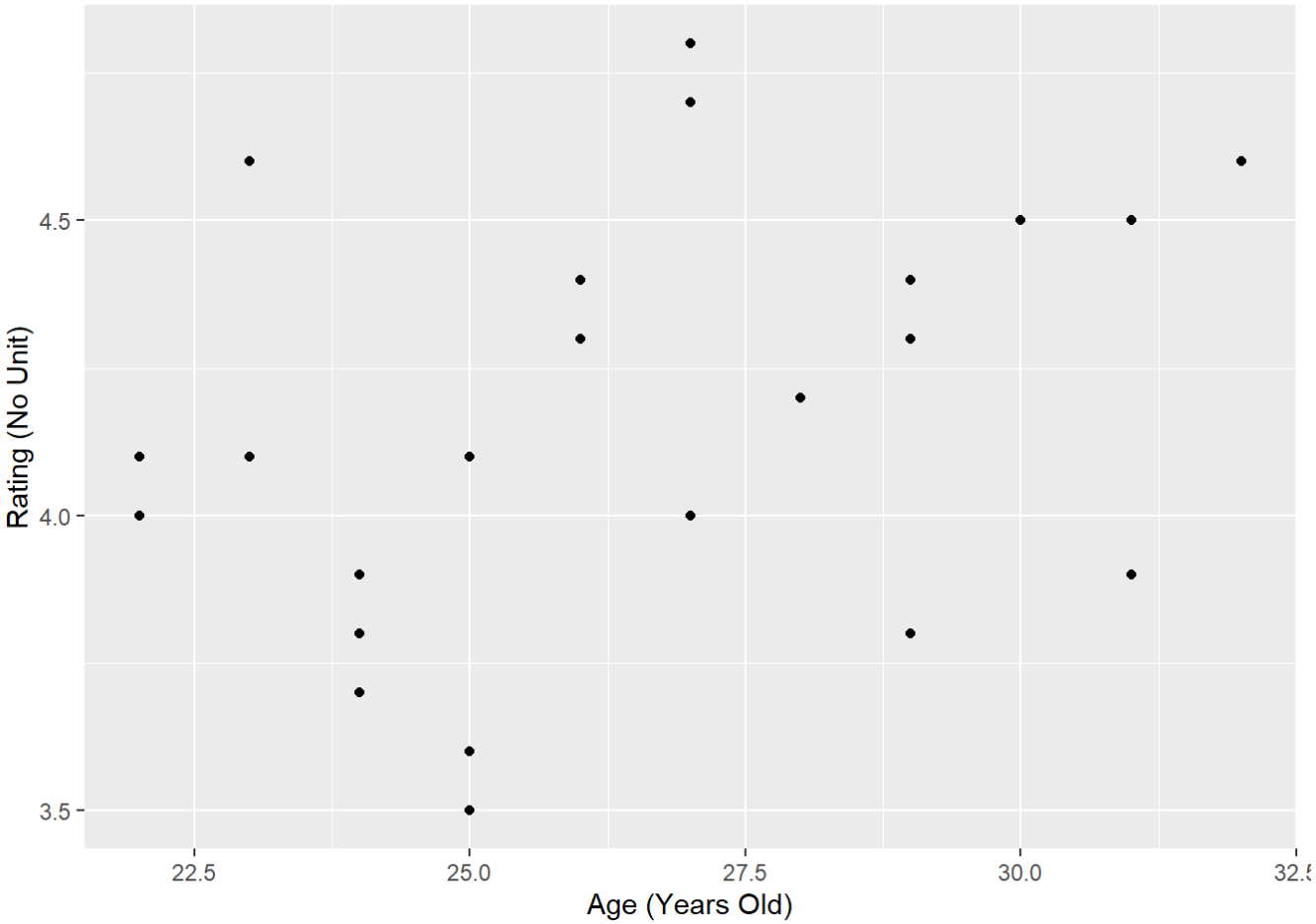
```
# complete the code to generate the plot

ggplot(data=playlist_data) + aes(x=Age,y=Rating) + labs(x="Age (Years Old)",y="Rating (No Unit)")
```



Question 3.3: Represent data using points

Solution:

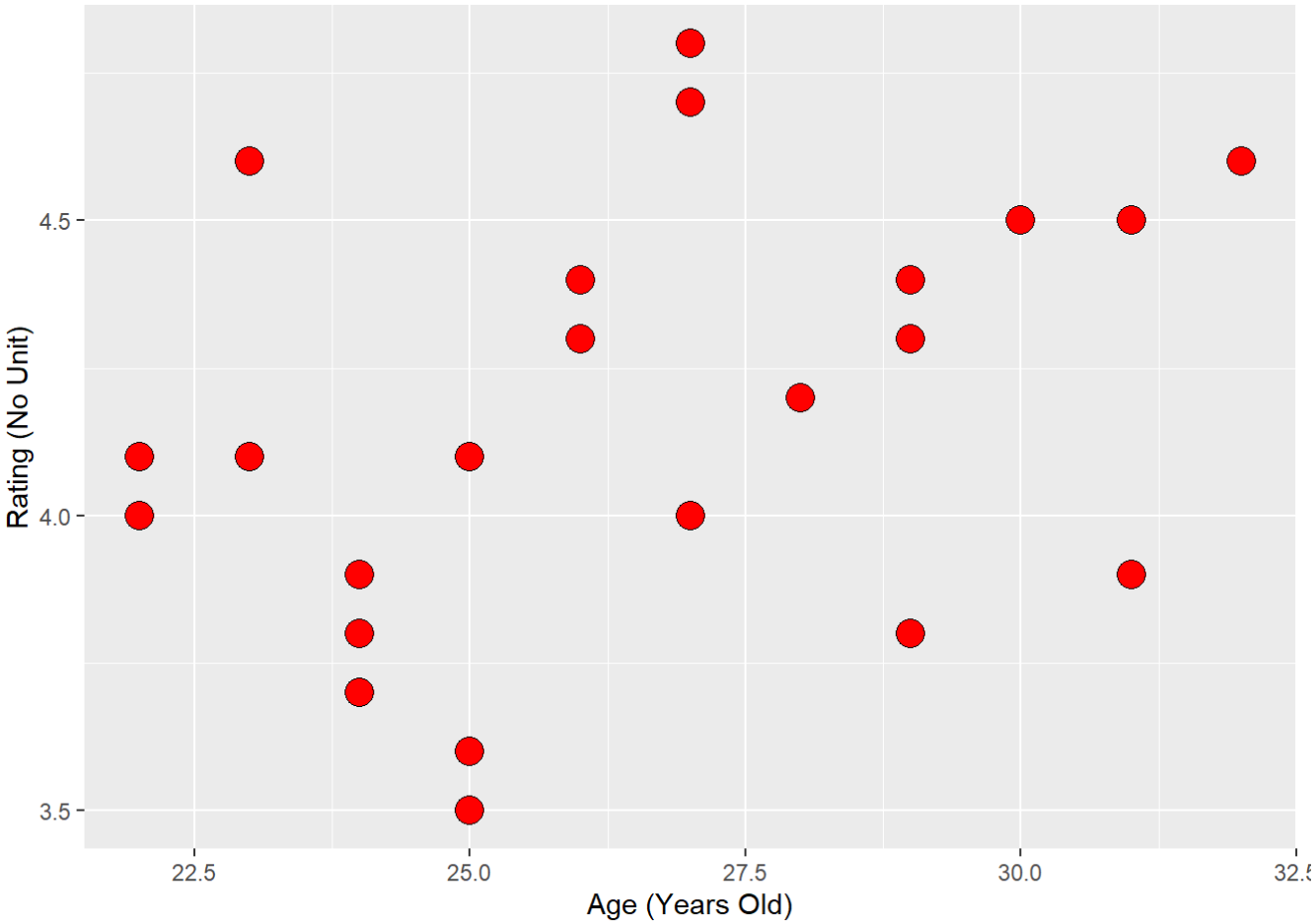
```
# complete the code to generate the plot

ggplot(data=playlist_data) + aes(x=Age,y=Rating) + labs(x="Age (Years Old)",y="Rating (No Unit)") + geom_point()
```



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

Solution:

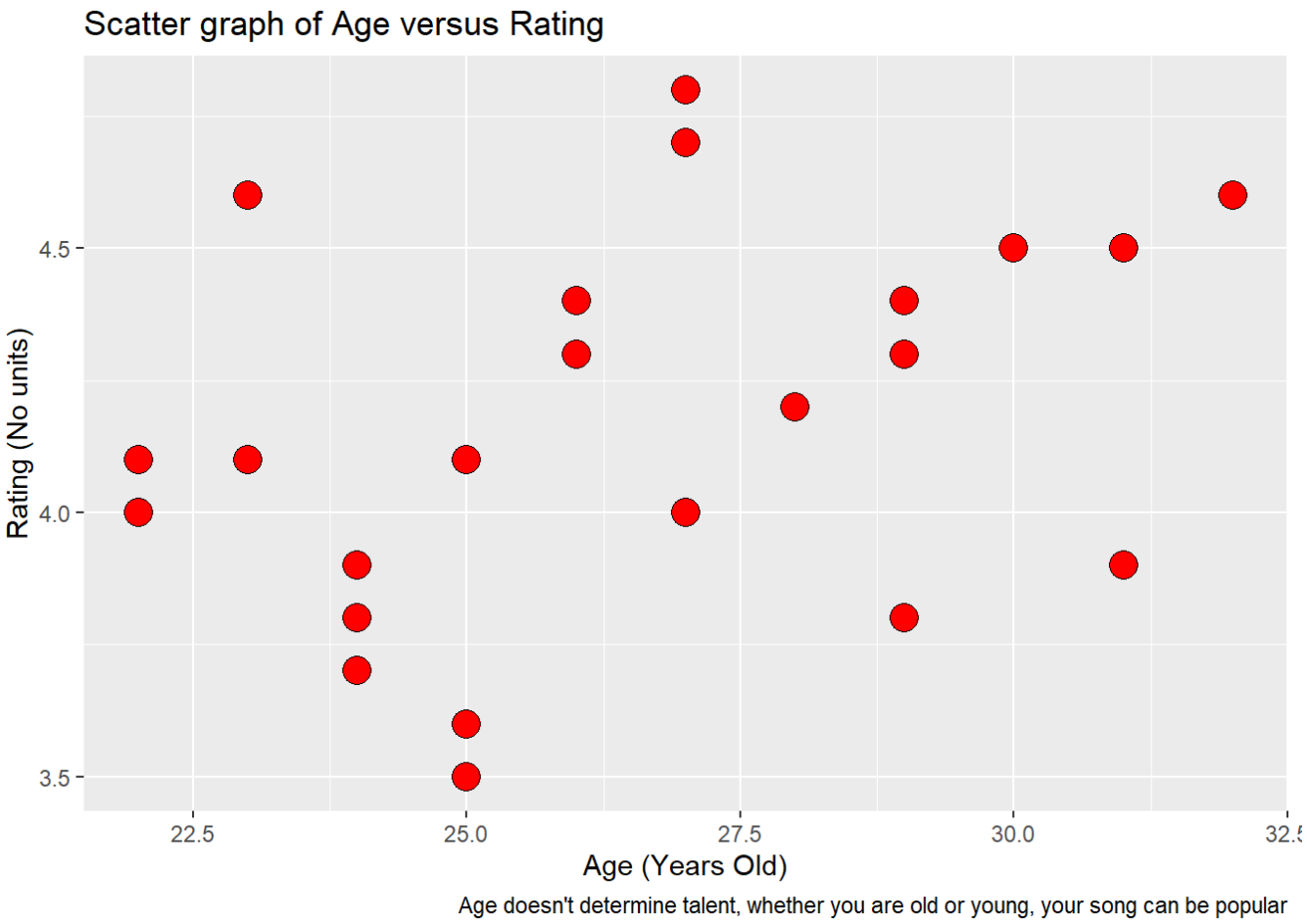
```
# complete the code to generate the plot
ggplot(data=playlist_data) + aes(x=Age,y=Rating) + labs(x="Age (Years Old)",y="Rating (No Unit)") + geom_point(shape=21, fill="red", size=5)
```



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

Solution:

```
# complete the code to generate the plot

ggplot(data=playlist_data) + aes(x=Age,y=Rating) + labs(x="Age (Years Old)",y="Rating (No Unit)") + geom_point(shape=21, fill="red", size=5)+ labs(x="Age (Years Old)",y="Rating (No units)",title="Scatter graph of Age versus Rating",caption="Age doesn't determine talent, whether you are old or young, your song can be popular")
```


Scatter graph of Age versus Rating
Rating (No units)
4.5
4.0
3.5
22.5
25.0
27.5
30.0
Age (Years Old)
Age doesn't determine talent, whether you are old or young, your song can be popular