# Statistical explorations, data preparation, and correlations

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becor
```

```
library(haven)
library(labelled)
library(rvest)
```

```
Attaching package: 'rvest'

The following object is masked from 'package:readr':

    guess_encoding
```

### Data Preparation

```r
# Import data
hw2_data <- read_sav("11002_2022_9626_MOESM1_ESM.sav") |>
  mutate(
    across(everything(), ~ na_if(., 99))
  ) |>
  mutate(
    time_music_last_week = Q6ax1_1 + Q6ax2_1 / 60,
    Q19_avg = rowMeans(across(starts_with("Q19_1_")), na.rm = TRUE)
  )

# Data of time spend on music and rating
time_rating_data <- hw2_data |>
  select(time_music_last_week, Q19_avg) |>
  rename(rating_avg = Q19_avg) |>
  drop_na()

# Data of age and song_age
age_data <- hw2_data |>
  select(Q1, starts_with("Q19_1_")) |>
  rename(birth_year = Q1) |>
  pivot_longer(
    cols = starts_with("Q19_1_"),
    names_to = "release",
    values_to = "rating"
  ) |>
  mutate(
    release = as.numeric(gsub("Q19_1_", "", release)) * 2 + 1948,
    song_age = release - birth_year
  )

# Data of song_age vs rating average
age_rating_data <- age_data |>
  select(song_age, rating) |>
  group_by(song_age) |>
  summarise(
    rating_avg = mean(rating, na.rm = TRUE),
    .groups = "drop"
  ) |>
  drop_na()

# Make a table of variable names and labels
variable_names <- names(hw2_data)
```

```r
variable_labels <- sapply(hw2_data, var_label)
variable_table <- tibble(
  variable = variable_names,
  label = variable_labels
)
```

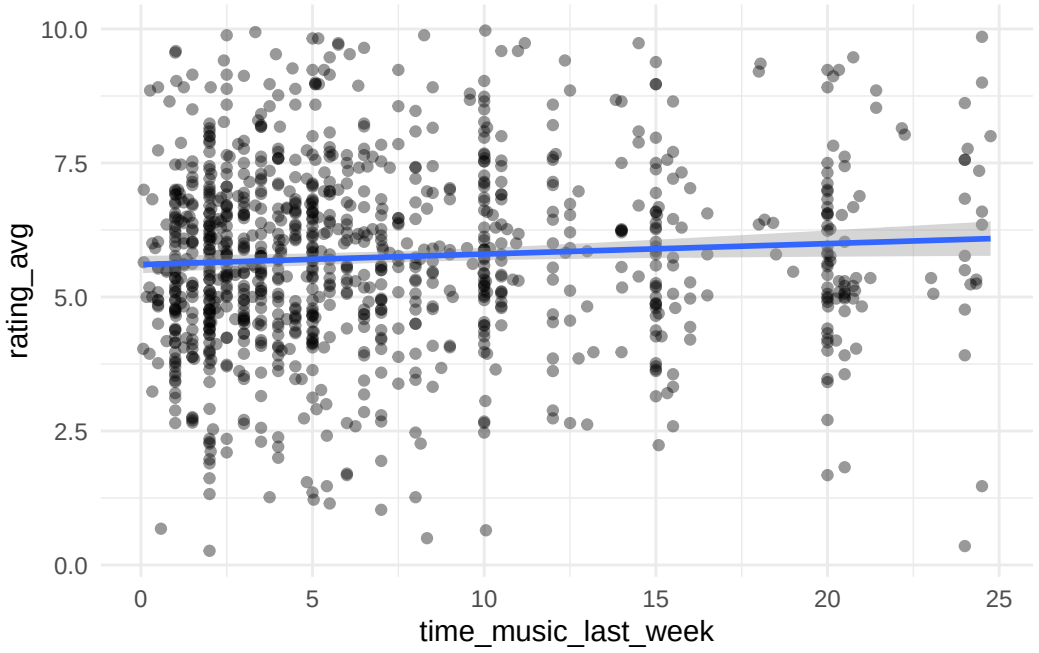
```r
# Read the webpage
url <- "https://link.springer.com/article/10.1007/s11002-022-09626-7/tables/2"
webpage <- read_html(url)

# Extract the table
music_table <- webpage |>
  html_node("table") |>
  html_table() |>
  rename(
    song_year = 'Song Year',
    song_title = 'Song Title',
    performers = 'Performer/s'
  )
print(music_table)
```

```
# A tibble: 34 x 3
   song_year song_title                    performers
       <int> <chr>                        <chr>
 1      1950 Play a Simple Melody         Bing and Gary Crosby
 2      1952 You Belong to Me             Jo Stafford
 3      1954 Sh Boom Sh Boom              The Crew Cuts
 4      1956 My Prayer                    The Platters
 5      1958 Patricia                     Perez Prado
 6      1960 Running Bear                 Johnny Preston
 7      1962 Roses are Red                Bobby Vinton
 8      1964 I Get Around                 Beach Boys
 9      1966 The Last Train to Clarksville The Monkees
10      1968 People Got to be Free        The Rascals
# i 24 more rows
```

## Analysis

### Time spend on music vs. rating

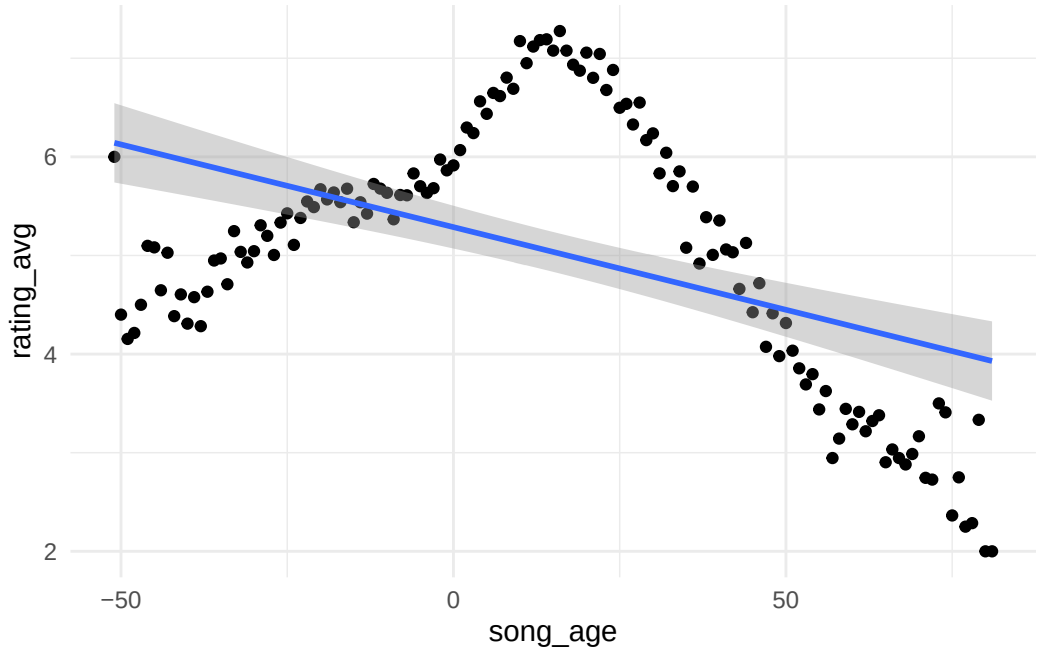
```
time_rating_data |>
  ggplot(aes(x = time_music_last_week, y = rating_avg)) +
  geom_point(
    alpha = .4
  ) +
  geom_smooth(
    method = "lm",
    formula = y ~ x
  ) +
  theme_minimal()
```




### Song age vs. rating average

```
age_rating_data |>
  ggplot(aes(x = song_age, rating_avg)) +
  geom_point() +
```

```
  geom_smooth(
    method = "lm",
    formula = y ~ x
  ) +
  theme_minimal()
```




```
cor.test(x = age_rating_data$song_age, y = age_rating_data$rating_avg, method = "pearson")
```

```
    Pearson's product-moment correlation

data:  age_rating_data$song_age and age_rating_data$rating_avg
t = -6.2962, df = 131, p-value = 4.254e-09
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.6027608 -0.3396278
sample estimates:
      cor
-0.481989
```