RWorksheet#5_group_Delgado_Sobusa_Tamonan.Rmd

Nexon Sobusa

2024-11-22

```
library(polite)
```

```
## Warning: package 'polite' was built under R version 4.4.2
```

```
library(httr)
```

```
## Warning: package 'httr' was built under R version 4.4.2
```

```
library(rvest)
```

```
## Warning: package 'rvest' was built under R version 4.4.2
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.4.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(stringr)
```

```
## Warning: package 'stringr' was built under R version 4.4.2
```

```
library(magrittr)
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.2
```

```
library(tidyverse)
```

```
## Warning: package 'purrr' was built under R version 4.4.2
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.5
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::extract()        masks magrittr::extract()
## x dplyr::filter()         masks stats::filter()
## x readr::guess_encoding() masks rvest::guess_encoding()
```

```
## x dplyr::lag()            masks stats::lag()
## x purrr::set_names()      masks magrittr::set_names()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
site_url <- "https://www.imdb.com/chart/toptv/?ref_=nv_tvv_250"

# 1. Extracting TV Shows
page <- read_html(site_url)
show_titles <- page %>%
  html_nodes("a h3.ipc-title__text") %>%
  html_text()

show_titles
```

```
##  [1] "1. Breaking Bad"                    "2. Planet Earth II"
##  [3] "3. Planet Earth"                    "4. Band of Brothers"
##  [5] "5. Chernobyl"                       "6. The Wire"
##  [7] "7. Avatar: The Last Airbender"      "8. Blue Planet II"
##  [9] "9. The Sopranos"                    "10. Cosmos: A Spacetime Odyssey"
## [11] "11. Cosmos"                         "12. Our Planet"
## [13] "13. Game of Thrones"                "14. Bluey"
## [15] "15. The World at War"               "16. Fullmetal Alchemist Brotherhood"
## [17] "17. Rick and Morty"                 "18. Life"
## [19] "19. The Last Dance"                 "20. The Twilight Zone"
## [21] "21. The Vietnam War"                "22. Sherlock"
## [23] "23. Attack on Titan"                "24. Batman: The Animated Series"
## [25] "25. Arcane"
```

```
titles_frame <- as.data.frame(show_titles[3:52], stringsAsFactors = FALSE)
colnames(titles_frame) <- "rank_and_title"
split_titles <- strsplit(as.character(titles_frame$rank_and_title), "\\.", fixed = FALSE)
split_titles <- data.frame(do.call(rbind, split_titles), stringsAsFactors = FALSE)

colnames(split_titles) <- c("rank", "title")
split_titles <- split_titles %>% dplyr::select(rank, title)

split_titles$title <- trimws(split_titles$title)

ranked_titles <- split_titles
```

```
ratings_list <- read_html(site_url) %>%
  html_nodes('.ipc-rating-star--rating') %>%
  html_text()

votes_list <- read_html(site_url) %>%
  html_nodes('.ipc-rating-star--voteCount') %>%
  html_text()
cleaned_votes <- gsub('[()]', '', votes_list)
```

```
episodes_list <- read_html(site_url) %>%
  html_nodes('span.sc-5bc66c50-6.OOdsw.cli-title-metadata-item:nth-of-type(2)') %>%
  html_text()
cleaned_episodes <- gsub('[eps]', '', episodes_list)
total_episodes <- as.numeric(cleaned_episodes)
```

```r
# Extracting years from the IMDb page
years_list <- read_html(site_url) %>%
  html_nodes(".secondaryInfo") %>% # CSS selector for the year
  html_text()

# Clean the extracted years (removing parentheses)
cleaned_years <- gsub("[()]", "", years_list)
```

```r
# Ensure the length matches the other columns
max_length <- max(length(ranked_titles$title), length(ratings_list), length(cleaned_votes), length(tota

ranked_titles$title <- c(ranked_titles$title, rep(NA, max_length - length(ranked_titles$title)))
ratings_list <- c(ratings_list, rep(NA, max_length - length(ratings_list)))
cleaned_votes <- c(cleaned_votes, rep(NA, max_length - length(cleaned_votes)))
total_episodes <- c(total_episodes, rep(NA, max_length - length(total_episodes)))
cleaned_years <- c(cleaned_years, rep(NA, max_length - length(cleaned_years)))

# Create the final dataframe
tv_shows_data <- data.frame(
  Rank = ranked_titles$rank,
  Title = ranked_titles$title,
  Rating = ratings_list,
  Voters = cleaned_votes,
  Episodes = total_episodes,
  Year = cleaned_years,
  stringsAsFactors = FALSE
)

# View the final dataframe
print(tv_shows_data)
```

```
##    Rank                          Title Rating Voters Episodes Year
## 1     3                    Planet Earth    9.5   2.2M       NA <NA>
## 2     4                Band of Brothers    9.5   162K       NA <NA>
## 3     5                       Chernobyl    9.4   224K       NA <NA>
## 4     6                       The Wire    9.4   546K       NA <NA>
## 5     7     Avatar: The Last Airbender    9.3   908K       NA <NA>
## 6     8                   Blue Planet II    9.3   391K       NA <NA>
## 7     9                     The Sopranos    9.3   390K       NA <NA>
## 8    10    Cosmos: A Spacetime Odyssey    9.3    49K       NA <NA>
## 9    11                          Cosmos    9.2   499K       NA <NA>
## 10   12                      Our Planet    9.2   131K       NA <NA>
## 11   13                 Game of Thrones    9.3    46K       NA <NA>
## 12   14                           Bluey    9.2    54K       NA <NA>
## 13   15               The World at War    9.2   2.4M       NA <NA>
## 14   16 Fullmetal Alchemist Brotherhood    9.3    33K       NA <NA>
## 15   17                  Rick and Morty    9.2    31K       NA <NA>
## 16   18                            Life    9.1   209K       NA <NA>
## 17   19                The Last Dance    9.1   628K       NA <NA>
## 18   20              The Twilight Zone    9.1    44K       NA <NA>
## 19   21                The Vietnam War    9.0   160K       NA <NA>
## 20   22                       Sherlock    9.0    97K       NA <NA>
## 21   23                Attack on Titan    9.1    29K       NA <NA>
## 22   24    Batman: The Animated Series    9.1     1M       NA <NA>
```

```
## 23   25                          Arcane    9.1   562K       NA <NA>
## 24 <NA>                            <NA>    9.0   122K       NA <NA>
## 25 <NA>                            <NA>    9.0   309K       NA <NA>
## 26 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 27 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 28 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 29 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 30 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 31 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 32 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 33 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 34 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 35 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 36 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 37 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 38 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 39 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 40 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 41 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 42 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 43 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 44 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 45 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 46 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 47 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 48 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 49 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 50 <NA>                            <NA>   <NA>   <NA>       NA <NA>
```

```r
base_link <- 'https://www.imdb.com/chart/toptv/'
main_html <- read_html(base_link)

review_links <- main_html %>%
  html_nodes("a.ipc-title-link-wrapper") %>%
  html_attr("href")
```

```r
review_data <- lapply(review_links, function(url_segment) {
  full_url <- paste0("https://imdb.com", url_segment)

  review_page <- read_html(full_url)
  individual_review_page <- review_page %>%
    html_nodes('a.isReview') %>%
    html_attr("href")

  critics_data <- review_page %>%
    html_nodes("span.score") %>%
    html_text()
  critics_frame <- data.frame(Critic_Reviews = critics_data[2], stringsAsFactors = FALSE)

  popularity_score <- review_page %>%
    html_nodes('[data-testid="hero-rating-bar__popularity__score"]') %>%
    html_text()

  detailed_review <- read_html(paste0("https://imdb.com", individual_review_page[1]))
  user_review_count <- detailed_review %>%
```

```
    html_nodes('[data-testid="tturv-total-reviews"]') %>%
    html_text()

  return(data.frame(User_Reviews = user_review_count, Critic = critics_frame, Popularity_Rating = popula
})

final_critics_data <- do.call(rbind, review_data)

tv_show_details <- cbind(tv_shows_data, final_critics_data)
tv_show_details
```

```
##    Rank                          Title Rating Voters Episodes Year
## 1     3                    Planet Earth    9.5   2.2M       NA <NA>
## 2     4                Band of Brothers    9.5   162K       NA <NA>
## 3     5                       Chernobyl    9.4   224K       NA <NA>
## 4     6                        The Wire    9.4   546K       NA <NA>
## 5     7     Avatar: The Last Airbender    9.3   908K       NA <NA>
## 6     8                   Blue Planet II    9.3   391K       NA <NA>
## 7     9                     The Sopranos    9.3   390K       NA <NA>
## 8    10    Cosmos: A Spacetime Odyssey    9.3    49K       NA <NA>
## 9    11                          Cosmos    9.2   499K       NA <NA>
## 10   12                      Our Planet    9.2   131K       NA <NA>
## 11   13                 Game of Thrones    9.3    46K       NA <NA>
## 12   14                           Bluey    9.2    54K       NA <NA>
## 13   15                The World at War    9.2   2.4M       NA <NA>
## 14   16 Fullmetal Alchemist Brotherhood    9.3    33K       NA <NA>
## 15   17                  Rick and Morty    9.2    31K       NA <NA>
## 16   18                            Life    9.1   209K       NA <NA>
## 17   19                The Last Dance    9.1   628K       NA <NA>
## 18   20               The Twilight Zone    9.1    44K       NA <NA>
## 19   21                The Vietnam War    9.0   160K       NA <NA>
## 20   22                       Sherlock    9.0    97K       NA <NA>
## 21   23                Attack on Titan    9.1    29K       NA <NA>
## 22   24    Batman: The Animated Series    9.1     1M       NA <NA>
## 23   25                          Arcane    9.1   562K       NA <NA>
## 24 <NA>                            <NA>    9.0   122K       NA <NA>
## 25 <NA>                            <NA>    9.0   309K       NA <NA>
## 26 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 27 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 28 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 29 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 30 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 31 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 32 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 33 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 34 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 35 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 36 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 37 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 38 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 39 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 40 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 41 <NA>                            <NA>   <NA>   <NA>       NA <NA>
## 42 <NA>                            <NA>   <NA>   <NA>       NA <NA>
```

```
## 43 <NA>                          <NA>   <NA>   <NA>       NA <NA>
## 44 <NA>                          <NA>   <NA>   <NA>       NA <NA>
## 45 <NA>                          <NA>   <NA>   <NA>       NA <NA>
## 46 <NA>                          <NA>   <NA>   <NA>       NA <NA>
## 47 <NA>                          <NA>   <NA>   <NA>       NA <NA>
## 48 <NA>                          <NA>   <NA>   <NA>       NA <NA>
## 49 <NA>                          <NA>   <NA>   <NA>       NA <NA>
## 50 <NA>                          <NA>   <NA>   <NA>       NA <NA>
##     User_Reviews Critic_Reviews Popularity_Rating
## 1  5,102 reviews            175                22
## 2  5,102 reviews            175                22
## 3    158 reviews              6             1,050
## 4    158 reviews              6             1,050
## 5    111 reviews             10             1,946
## 6    111 reviews             10             1,946
## 7  1,056 reviews             34               137
## 8  1,056 reviews             34               137
## 9  3,533 reviews             88               168
## 10 3,533 reviews             88               168
## 11   787 reviews             77               112
## 12   787 reviews             77               112
## 13 1,001 reviews             57               354
## 14 1,001 reviews             57               354
## 15    53 reviews              9             4,265
## 16    53 reviews              9             4,265
## 17   964 reviews             93                30
## 18   964 reviews             93                30
## 19   205 reviews             12             1,476
## 20   205 reviews             12             1,476
## 21    80 reviews              8             3,394
## 22    80 reviews              8             3,394
## 23   245 reviews             15             2,594
## 24   245 reviews             15             2,594
## 25 5,904 reviews            368                14
## 26 5,904 reviews            368                14
## 27   368 reviews              4               380
## 28   368 reviews              4               380
## 29   126 reviews              5             2,427
## 30   126 reviews              5             2,427
## 31   466 reviews             16               490
## 32   466 reviews             16               490
## 33   910 reviews             94               127
## 34   910 reviews             94               127
## 35    12 reviews              9             3,311
## 36    12 reviews              9             3,311
## 37   541 reviews             28             1,497
## 38   541 reviews             28             1,497
## 39   213 reviews             85               355
## 40   213 reviews             85               355
## 41   175 reviews             13             1,864
## 42   175 reviews             13             1,864
## 43 1,098 reviews            121               160
## 44 1,098 reviews            121               160
## 45 2,363 reviews             64                45
```

```
## 46 2,363 reviews             64                45
## 47   219 reviews             25               453
## 48   219 reviews             25               453
## 49 1,991 reviews             52                 2
## 50 1,991 reviews             52                 2
```

```r
# Convert 'Year' column to numeric for processing
tv_show_details$Year <- as.numeric(tv_show_details$Year)
if (any(is.na(tv_show_details$Year))) {
  warning("Some years could not be converted to numeric.")
}
```

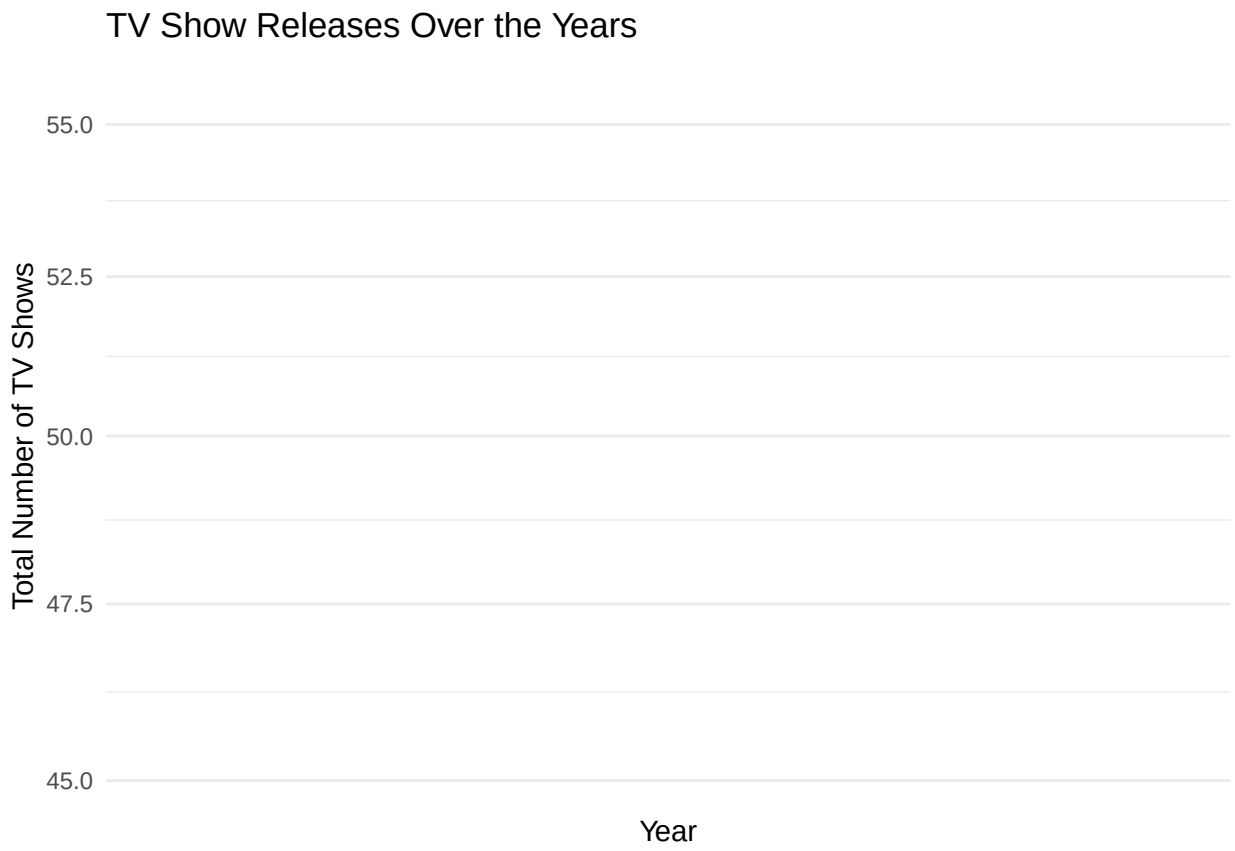
```
## Warning: Some years could not be converted to numeric.
```

```r
# Group shows by release year and calculate the count
shows_per_year <- tv_show_details %>%
  group_by(Year) %>%
  summarise(Total_Shows = n())

# Visualize the trend of TV show releases over time
ggplot(shows_per_year, aes(x = Year, y = Total_Shows)) +
  geom_line(color = "red", size = 1.2) +
  geom_point(color = "green", size = 2.5) +
  labs(title = "TV Show Releases Over the Years",
       x = "Year",
       y = "Total Number of TV Shows") +
  scale_y_log10() +
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```




```
# Identify the year with the highest number of releases
peak_release_year <- shows_per_year %>%
  filter(Total_Shows == max(Total_Shows))

print(peak_release_year)
```

```
## # A tibble: 1 x 2
##    Year Total_Shows
##   <dbl>       <int>
## 1    NA          50
```

```
# Breaking Bad
BreakingBad_urls <- "https://www.imdb.com/title/tt0903747/reviews/?ref_=tt_ov_urv"

df <- list()
df_names <- "Breaking_Bad"

session <- read_html(BreakingBad_urls)

# Extracting reviewer names
reviewer_names <- session %>%
  html_nodes(".ipc-link.ipc-link--base") %>%
  html_text() %>%
  head(20)

# Extracting review dates
review_dates <- session %>%
```

```
  html_nodes(".ipc-inline-list__item.review-date") %>%
  html_text() %>%
  head(20)

# Extracting user ratings
user_ratings <- session %>%
  html_nodes(".ipc-rating-star--rating") %>%  # Example selector, verify it in the HTML
  html_text() %>%
  head(20)

# Extracting review titles
review_titles <- session %>%
  html_nodes(".ipc-title__text") %>%
  html_text() %>%
  head(20)

# Extracting helpful reviews count
helpful_counts <- session %>%
  html_nodes(".ipc-voting__label__count.ipc-voting__label__count--up") %>%
  html_text() %>%
  head(20)

# Extracting not helpful reviews count
not_helpful_counts <- session %>%
  html_nodes(".ipc-voting__label__count.ipc-voting__label__count--down") %>%
  html_text() %>%
  head(20)

# Extracting text reviews
text_reviews_content <- session %>%
  html_nodes(".ipc-html-content-inner-div") %>%
  html_text() %>%
  head(20)

# Ensuring all lists have the same length (20 reviews)
reviewer_names <- c(reviewer_names, rep(NA, 20 - length(reviewer_names)))[1:20]
review_dates <- c(review_dates, rep(NA, 20 - length(review_dates)))[1:20]
user_ratings <- c(user_ratings, rep(NA, 20 - length(user_ratings)))[1:20]
review_titles <- c(review_titles, rep(NA, 20 - length(review_titles)))[1:20]
helpful_counts <- c(helpful_counts, rep(NA, 20 - length(helpful_counts)))[1:20]
not_helpful_counts <- c(not_helpful_counts, rep(NA, 20 - length(not_helpful_counts)))[1:20]
text_reviews_content <- c(text_reviews_content, rep(NA, 20 - length(text_reviews_content)))[1:20]

# Creating a temporary dataframe
dfTemp <- data.frame(
  reviewer_name = reviewer_names,
  review_date = review_dates,
  user_rating = user_ratings,
  review_title = review_titles,
  helpful_reviews = helpful_counts,
  not_helpful_reviews = not_helpful_counts,
  text_reviews = text_reviews_content,
  stringsAsFactors = FALSE
```

```
)

# Storing the dataframe in the list
df[[df_names]] <- dfTemp

# Print the results for Breaking Bad
print(df$Breaking_Bad)
```

```
##         reviewer_name  review_date user_rating
## 1             FiRE010  Jul 3, 2021          10
## 2           Permalink  Mar 6, 2019          10
## 3          bruhperson Jul 29, 2021          10
## 4           Permalink Feb 18, 2020          10
## 5        KinoKoopaKid  Nov 8, 2021          10
## 6           Permalink May 30, 2019          10
## 7         jehuschultz  Dec 8, 2022          10
## 8           Permalink Nov 15, 2019          10
## 9      Supermanfan-13 Jul 17, 2021          10
## 10          Permalink Nov 12, 2017          10
## 11  manishsingh-03299  Aug 5, 2022           7
## 12          Permalink Feb 14, 2021           4
## 13            Rob1331 Sep 12, 2015          10
## 14          Permalink  Dec 8, 2022          10
## 15           xpinerhd Jan 11, 2014          10
## 16          Permalink  Nov 8, 2021          10
## 17 dhanushreddy-14919 Aug 11, 2021          10
## 18          Permalink May 19, 2019          10
## 19  TheLittleSongbird  May 4, 2021          10
## 20          Permalink Jun 23, 2021          10
##
## 1
## 2
## 3
## 4
## 5
## 6
## 7                                                                                                  Those days a
## 8
## 9                                                                                                            I
## 10                                                                                                           A
## 11                                                                                  Among the best and most ac
## 12                                                                                                           A
## 13
## 14                                                                                              The Most Over-Rated
## 15                                                                                                        Once
## 16 If you mix Scarface, Robin Hood and maybe Tyler Durden with enough meth - you'll get a mean cocktai
## 17                                                                                          By far the greatest
## 18                                                                                                  in a ca
## 19                                                                  Since GOT is over, this is Officially the G
## 20                                                                                                Every bit a
##    helpful_reviews not_helpful_reviews
## 1             <NA>                <NA>
## 2             <NA>                <NA>
## 3             <NA>                <NA>
```

```
## 4             <NA>                <NA>
## 5             <NA>                <NA>
## 6             <NA>                <NA>
## 7             <NA>                <NA>
## 8             <NA>                <NA>
## 9             <NA>                <NA>
## 10            <NA>                <NA>
## 11            <NA>                <NA>
## 12            <NA>                <NA>
## 13            <NA>                <NA>
## 14            <NA>                <NA>
## 15            <NA>                <NA>
## 16            <NA>                <NA>
## 17            <NA>                <NA>
## 18            <NA>                <NA>
## 19            <NA>                <NA>
## 20            <NA>                <NA>
##
## 1
## 2
## 3
## 4
## 5
## 6
## 7
## 8
## 9
## 10 'Breaking Bad' is one of the most popular rated shows on IMDb, is one of those rarities where eve
## 11
## 12
## 13
## 14
## 15
## 16
## 17
## 18
## 19
## 20
```

```r
# Game of Thrones
game_of_thrones_url <- "https://www.imdb.com/title/tt0944947/reviews/?ref_=tt_ov_urv"

df_name <- "Game_of_Thrones"

session <- read_html(game_of_thrones_url)

# Extracting reviewer names
reviewer_names <- session %>%
  html_nodes(".ipc-link.ipc-link--base") %>%
  html_text() %>%
  head(20)

# Extracting review dates
review_dates <- session %>%
```

```
  html_nodes(".ipc-inline-list__item.review-date") %>%
  html_text() %>%
  head(20)

# Extracting user ratings
user_ratings <- session %>%
  html_nodes(".ipc-rating-star--rating") %>%
  html_text() %>%
  head(20)

# Extracting review titles
review_titles <- session %>%
  html_nodes(".ipc-title__text") %>%
  html_text() %>%
  head(20)

# Extracting helpful reviews count
helpful_review_counts <- session %>%
  html_nodes(".ipc-voting__label__count.ipc-voting__label__count--up") %>%
  html_text() %>%
  head(20)

# Extracting not helpful reviews count
not_helpful_review_counts <- session %>%
  html_nodes(".ipc-voting__label__count.ipc-voting__label__count--down") %>%
  html_text() %>%
  head(20)

# Extracting text reviews
text_review_contents <- session %>%
  html_nodes(".ipc-html-content-inner-div") %>%
  html_text() %>%
  head(20)

# Ensure all lists have the same length (20 reviews)
reviewer_names <- c(reviewer_names, rep(NA, 20 - length(reviewer_names)))[1:20]
review_dates <- c(review_dates, rep(NA, 20 - length(review_dates)))[1:20]
user_ratings <- c(user_ratings, rep(NA, 20 - length(user_ratings)))[1:20]
review_titles <- c(review_titles, rep(NA, 20 - length(review_titles)))[1:20]
helpful_review_counts <- c(helpful_review_counts, rep(NA, 20 - length(helpful_review_counts)))[1:20]
not_helpful_review_counts <- c(not_helpful_review_counts, rep(NA, 20 - length(not_helpful_review_counts
text_review_contents <- c(text_review_contents, rep(NA, 20 - length(text_review_contents)))[1:20]

# Creating a temporary dataframe
df_temp <- data.frame(
  reviewer_names = reviewer_names,
  review_dates = review_dates,
  user_ratings = user_ratings,
  review_titles = review_titles,
  helpful_review_counts = helpful_review_counts,
  not_helpful_review_counts = not_helpful_review_counts,
  text_review_contents = text_review_contents,
  stringsAsFactors = FALSE
```

```
)

# Storing the dataframe in the list
df[[df_name]] <- df_temp

# Print the results for Game of Thrones
print(df$Game_of_Thrones)
```

```
##       reviewer_names review_dates user_ratings
## 1         danielkpkp May 11, 2020            9
## 2          Permalink May 24, 2019            8
## 3  samxxx-671-826221 May 20, 2019            8
## 4          Permalink  Apr 8, 2020           10
## 5          slowcando  May 9, 2019            9
## 6          Permalink Aug 22, 2022           10
## 7           SaifOVGU  Feb 5, 2023            9
## 8          Permalink  Dec 9, 2023           10
## 9   jacobnoble-02524 May 25, 2019            8
## 10         Permalink  Nov 8, 2017           10
## 11    heavenacceptme  Nov 1, 2019            6
## 12         Permalink May 20, 2019            1
## 13     Dan_W_Reviews Nov 10, 2023           10
## 14         Permalink May 18, 2020            9
## 15    Supermanfan-13  Nov 8, 2022            9
## 16         Permalink  May 5, 2019            9
## 17         tweaknhoe  Jun 2, 2020           10
## 18         Permalink  May 9, 2019            8
## 19 TheLittleSongbird Apr 16, 2011           10
## 20         Permalink May 19, 2019            7
##                                                                 review_titles
## 1                                                                User reviews
## 2                        It could have been the best TV series ever made...
## 3                    A perfect example of: Falling in Love with the Wrong Guy
## 4              Seasons 1-6: outstanding. 7: daft but good. 8: disappointing
## 5                           Can you just make the remake the season finale?
## 6                                                           Game of Thrones
## 7  Despite the final season, Game of Thrones remains an all time classic
## 8                 Captivating and Gripping but a Disappointing Final Season
## 9                                                 One of the best shows ever
## 10                  Why just why? This show could have been the best ever.
## 11                                                This is a television show?
## 12                                                 Imagine an Ice Cream Shop
## 13                                                 A Message to Dan and Dave
## 14                                                                   Amazing
## 15                                          Extraordinary untill season 8
## 16                                           one the best shows ever made
## 17  Can we please just restart season 8, perhaps 7 as well, but mainly 8
## 18                                                           almost perfect
## 19                                       This was an 10/10 until S08E03
## 20                                                   Excellent adaptation.
##    helpful_review_counts not_helpful_review_counts
## 1                   <NA>                       <NA>
## 2                   <NA>                       <NA>
## 3                   <NA>                       <NA>
```

```
## 4                   <NA>                       <NA>
## 5                   <NA>                       <NA>
## 6                   <NA>                       <NA>
## 7                   <NA>                       <NA>
## 8                   <NA>                       <NA>
## 9                   <NA>                       <NA>
## 10                  <NA>                       <NA>
## 11                  <NA>                       <NA>
## 12                  <NA>                       <NA>
## 13                  <NA>                       <NA>
## 14                  <NA>                       <NA>
## 15                  <NA>                       <NA>
## 16                  <NA>                       <NA>
## 17                  <NA>                       <NA>
## 18                  <NA>                       <NA>
## 19                  <NA>                       <NA>
## 20                  <NA>                       <NA>
##
## 1
## 2
## 3
## 4
## 5
## 6
## 7
## 8
## 9  Was over-time on a gradual binge of watching 'Game of Thrones' from the first episode (gradual be
## 10
## 11
## 12
## 13
## 14
## 15
## 16
## 17
## 18
## 19
## 20
```

```r
# Stranger Things
stranger_things_url <- "https://www.imdb.com/title/tt4574334/reviews/?ref_=tt_ov_urv"

df_name <- "Stranger_Things"

session <- read_html(stranger_things_url)

# Extracting reviewer names
reviewer_names <- session %>%
  html_nodes(".ipc-link.ipc-link--base") %>%
  html_text() %>%
  head(20)

# Extracting review dates
review_dates <- session %>%
```

```
  html_nodes(".ipc-inline-list__item.review-date") %>%
  html_text() %>%
  head(20)

# Extracting user ratings
user_ratings <- session %>%
  html_nodes(".ipc-rating-star--rating") %>%
  html_text() %>%
  head(20)

# Extracting review titles
review_titles <- session %>%
  html_nodes(".ipc-title__text") %>%
  html_text() %>%
  head(20)

# Extracting helpful reviews count
helpful_review_counts <- session %>%
  html_nodes(".ipc-voting__label__count.ipc-voting__label__count--up") %>%
  html_text() %>%
  head(20)

# Extracting not helpful reviews count
not_helpful_review_counts <- session %>%
  html_nodes(".ipc-voting__label__count.ipc-voting__label__count--down") %>%
  html_text() %>%
  head(20)

# Extracting text reviews
text_review_contents <- session %>%
  html_nodes(".ipc-html-content-inner-div") %>%
  html_text() %>%
  head(20)

# Ensure all lists have the same length (20 reviews)
reviewer_names <- c(reviewer_names, rep(NA, 20 - length(reviewer_names)))[1:20]
review_dates <- c(review_dates, rep(NA, 20 - length(review_dates)))[1:20]
user_ratings <- c(user_ratings, rep(NA, 20 - length(user_ratings)))[1:20]
review_titles <- c(review_titles, rep(NA, 20 - length(review_titles)))[1:20]
helpful_review_counts <- c(helpful_review_counts, rep(NA, 20 - length(helpful_review_counts)))[1:20]
not_helpful_review_counts <- c(not_helpful_review_counts, rep(NA, 20 - length(not_helpful_review_counts
text_review_contents <- c(text_review_contents, rep(NA, 20 - length(text_review_contents)))[1:20]

# Creating a temporary dataframe
df_temp <- data.frame(
  reviewer_names = reviewer_names,
  review_dates = review_dates,
  user_ratings = user_ratings,
  review_titles = review_titles,
  helpful_review_counts = helpful_review_counts,
  not_helpful_review_counts = not_helpful_review_counts,
  text_review_contents = text_review_contents,
  stringsAsFactors = FALSE
```

```
)

# Storing the dataframe in the list
df[[df_name]] <- df_temp

# Print the results for Stranger Things
print(df$Stranger_Things)
```

```
##    reviewer_names review_dates user_ratings
## 1  Supermanfan-13 Jan 22, 2023            9
## 2       Permalink  Jan 8, 2018            8
## 3  Sleepin_Dragon Jul 21, 2016            9
## 4       Permalink Nov 22, 2016           10
## 5         cherold Mar 10, 2023            9
## 6       Permalink Mar 12, 2023            9
## 7   planktonrules Jan 22, 2018           10
## 8       Permalink Jun 21, 2022            9
## 9         Rob1331 Sep 22, 2020           10
## 10      Permalink  Jul 3, 2022           10
## 11        EVON1TY Jan 22, 2023           10
## 12      Permalink  Jul 9, 2022            7
## 13    gogoschka-1  Jul 3, 2019            5
## 14      Permalink Jul 16, 2019           10
## 15  magnoliacream Jul 23, 2021           10
## 16      Permalink Jul 14, 2016           10
## 17  personalhonor May 30, 2022           10
## 18      Permalink Jul 14, 2016            9
## 19     pnxrxvgcfg Jul 14, 2016            8
## 20      Permalink Jul 14, 2016           10
##
## 1
## 2
## 3
## 4
## 5
## 6
## 7
## 8  A nostalgic callback to the stories from my youth - but also a beautifully shot supernatural tale
## 9
## 10
## 11
## 12
## 13
## 14
## 15                                                                                  I rated it 9 
## 16
## 17
## 18
## 19
## 20
##    helpful_review_counts not_helpful_review_counts
## 1                   <NA>                       <NA>
## 2                   <NA>                       <NA>
## 3                   <NA>                       <NA>
```

```
## 4                   <NA>                    <NA>
## 5                   <NA>                    <NA>
## 6                   <NA>                    <NA>
## 7                   <NA>                    <NA>
## 8                   <NA>                    <NA>
## 9                   <NA>                    <NA>
## 10                  <NA>                    <NA>
## 11                  <NA>                    <NA>
## 12                  <NA>                    <NA>
## 13                  <NA>                    <NA>
## 14                  <NA>                    <NA>
## 15                  <NA>                    <NA>
## 16                  <NA>                    <NA>
## 17                  <NA>                    <NA>
## 18                  <NA>                    <NA>
## 19                  <NA>                    <NA>
## 20                  <NA>                    <NA>
##
## 1
## 2
## 3
## 4
## 5
## 6  There were two dominating forces in the eighties that had a lasting effect on my cinematic taste
## 7
## 8
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16
## 17
## 18
## 19
## 20
```

```r
# Band Of Brothers
BoB_url <- "https://www.imdb.com/title/tt0185906/reviews/?ref_=tt_ov_urv"

df_name <- "Band_of_Brothers"

html_session <- read_html(BoB_url)

# Extracting reviewer names
reviewer_names <- html_session %>%
  html_nodes(".ipc-link.ipc-link--base") %>%
  html_text() %>%
  head(20)

# Extracting review dates
review_dates <- html_session %>%
```

```r
  html_nodes(".ipc-inline-list__item.review-date") %>%
  html_text() %>%
  head(20)

# Extracting user ratings
user_ratings <- html_session %>%
  html_nodes(".ipc-rating-star--rating") %>%
  html_text() %>%
  head(20)

# Extracting review titles
review_titles <- html_session %>%
  html_nodes(".ipc-title__text") %>%
  html_text() %>%
  head(20)

# Extracting helpful reviews count
helpful_counts <- html_session %>%
  html_nodes(".ipc-voting__label__count.ipc-voting__label__count--up") %>%
  html_text() %>%
  head(20)

# Extracting not helpful reviews count
not_helpful_counts <- html_session %>%
  html_nodes(".ipc-voting__label__count.ipc-voting__label__count--down") %>%
  html_text() %>%
  head(20)

# Extracting text reviews
text_reviews_content <- html_session %>%
  html_nodes(".ipc-html-content-inner-div") %>%
  html_text() %>%
  head(20)

# Fill missing values
reviewer_names <- c(reviewer_names, rep(NA, 20 - length(reviewer_names)))[1:20]
review_dates <- c(review_dates, rep(NA, 20 - length(review_dates)))[1:20]
user_ratings <- c(user_ratings, rep(NA, 20 - length(user_ratings)))[1:20]
review_titles <- c(review_titles, rep(NA, 20 - length(review_titles)))[1:20]
helpful_counts <- c(helpful_counts, rep(NA, 20 - length(helpful_counts)))[1:20]
not_helpful_counts <- c(not_helpful_counts, rep(NA, 20 - length(not_helpful_counts)))[1:20]
text_reviews_content <- c(text_reviews_content, rep(NA, 20 - length(text_reviews_content)))[1:20]

# Creating a temporary dataframe
dfTemp <- data.frame(
  reviewer_names = reviewer_names,
  review_dates = review_dates,
  user_ratings = user_ratings,
  review_titles = review_titles,
  helpful_review_counts = helpful_counts,
  not_helpful_review_counts = not_helpful_counts,
  text_review_contents = text_reviews_content,
  stringsAsFactors = FALSE
```

```
)

# Storing the dataframe in the list
df[[df_name]] <- dfTemp

# Print the results for Band of Brothers
print(df$Band_of_Brothers)
```

```
##        reviewer_names review_dates user_ratings
## 1             Rob1331 Sep 27, 2022           10
## 2           Permalink Oct 14, 2001           10
## 3         sanderson777 Jan 18, 2002           10
## 4           Permalink Apr 18, 2004           10
## 5          wildcatt268 Feb 13, 2003           10
## 6           Permalink Jan 23, 2005           10
## 7             arjay24 Sep 16, 2004           10
## 8           Permalink  May 6, 2022           10
## 9           rbverhoef  Nov 4, 2019           10
## 10          Permalink  Nov 5, 2001           10
## 11         yodaschoda Aug 25, 2004           10
## 12          Permalink May 30, 2015            7
## 13 philip_vanderveken Apr 10, 2021            5
## 14          Permalink  May 2, 2006           10
## 15     Supermanfan-13  Jun 3, 2019           10
## 16          Permalink Jan 26, 2005           10
## 17          thiagoutp  May 3, 2022           10
## 18          Permalink Oct 24, 2018            9
## 19         bsmith5552  Dec 7, 2002           10
## 20          Permalink Nov 25, 2002           10
##                                                        review_titles
## 1                                                       User reviews
## 2                                                       Incredible!!
## 3                           Possibly the finest 10 hours ever created
## 4                              One of the best war movies/series ever
## 5                                                           Realistic
## 6                                                           Excellent
## 7                    One of, if not the best, mini series' ever made
## 8             This series is so unbelievably realistic, so authentic.
## 9                           One of the best mini-series ever created!
## 10                                           Probably the best ever
## 11                           Realistic WWII Drama With Warts Included
## 12                                                    war, no frills
## 13                                          You can't beat this....
## 14                                                      Overrated??
## 15                                       Not very realistic at all
## 16                 Without Doubt, the Best Mini-Series Ever Recorded
## 17                                                   Great Miniseries
## 18 A series like this won't be made again (see below), so treasure it
## 19                                         Share With Your Children
## 20                                            Best Mini series ever
##    helpful_review_counts not_helpful_review_counts
## 1                   <NA>                       <NA>
## 2                   <NA>                       <NA>
## 3                   <NA>                       <NA>
```

```
## 4                   <NA>                    <NA>
## 5                   <NA>                    <NA>
## 6                   <NA>                    <NA>
## 7                   <NA>                    <NA>
## 8                   <NA>                    <NA>
## 9                   <NA>                    <NA>
## 10                  <NA>                    <NA>
## 11                  <NA>                    <NA>
## 12                  <NA>                    <NA>
## 13                  <NA>                    <NA>
## 14                  <NA>                    <NA>
## 15                  <NA>                    <NA>
## 16                  <NA>                    <NA>
## 17                  <NA>                    <NA>
## 18                  <NA>                    <NA>
## 19                  <NA>                    <NA>
## 20                  <NA>                    <NA>
##
## 1
## 2
## 3
## 4
## 5
## 6
## 7
## 8
## 9
## 10
## 11
## 12
## 13
## 14 Lots of people applaud this series for its realism, but I can't really agree. I think there is st
## 15
## 16
## 17
## 18
## 19
## 20
```

```r
# Chernobyl
Chernobyl_url <- "https://www.imdb.com/title/tt7366338/reviews/?ref_=tt_ov_urv"

df_name <- "Chernobyl"

html_session <- read_html(Chernobyl_url)

# Extracting reviewer names
reviewer_names <- html_session %>%
  html_nodes(".ipc-link.ipc-link--base") %>%
  html_text() %>%
  head(20)

# Extracting review dates
review_dates <- html_session %>%
```

```r
  html_nodes(".ipc-inline-list__item.review-date") %>%
  html_text() %>%
  head(20)

# Extracting user ratings
user_ratings <- html_session %>%
  html_nodes(".ipc-rating-star--rating") %>%
  html_text() %>%
  head(20)

# Extracting review titles
review_titles <- html_session %>%
  html_nodes(".ipc-title__text") %>%
  html_text() %>%
  head(20)

# Extracting helpful reviews count
helpful_counts <- html_session %>%
  html_nodes(".ipc-voting__label__count.ipc-voting__label__count--up") %>%
  html_text() %>%
  head(20)

# Extracting not helpful reviews count
not_helpful_counts <- html_session %>%
  html_nodes(".ipc-voting__label__count.ipc-voting__label__count--down") %>%
  html_text() %>%
  head(20)

# Extracting text reviews
text_reviews_content <- html_session %>%
  html_nodes(".ipc-html-content-inner-div") %>%
  html_text() %>%
  head(20)

# Fill missing values
reviewer_names <- c(reviewer_names, rep(NA, 20 - length(reviewer_names)))[1:20]
review_dates <- c(review_dates, rep(NA, 20 - length(review_dates)))[1:20]
user_ratings <- c(user_ratings, rep(NA, 20 - length(user_ratings)))[1:20]
review_titles <- c(review_titles, rep(NA, 20 - length(review_titles)))[1:20]
helpful_counts <- c(helpful_counts, rep(NA, 20 - length(helpful_counts)))[1:20]
not_helpful_counts <- c(not_helpful_counts, rep(NA, 20 - length(not_helpful_counts)))[1:20]
text_reviews_content <- c(text_reviews_content, rep(NA, 20 - length(text_reviews_content)))[1:20]

# Creating a temporary dataframe
dfTemp <- data.frame(
  reviewer_names = reviewer_names,
  review_dates = review_dates,
  user_ratings = user_ratings,
  review_titles = review_titles,
  helpful_review_counts = helpful_counts,
  not_helpful_review_counts = not_helpful_counts,
  text_review_contents = text_reviews_content,
  stringsAsFactors = FALSE
```

```
)

# Storing the dataframe in the list
df[[df_name]] <- dfTemp

# Print the results for Chernobyl
print(df$Chernobyl)
```

```
##       reviewer_names review_dates user_ratings
## 1    curiosityonmars May 23, 2019           10
## 2          Permalink May 10, 2019           10
## 3           stelmakh  May 9, 2019           10
## 4          Permalink May 14, 2019           10
## 5       natashapekar  May 7, 2019           10
## 6          Permalink May 20, 2019           10
## 7         m-porpaczi  May 6, 2019           10
## 8          Permalink May 13, 2019           10
## 9           Lladerat  May 6, 2019           10
## 10         Permalink Nov 27, 2019           10
## 11          jfirebug May 23, 2019            5
## 12         Permalink Jun 15, 2019            8
## 13           thegldt May 20, 2019           10
## 14         Permalink May 30, 2019           10
## 15 alexander-phoenix  Jun 7, 2019            9
## 16         Permalink Sep 27, 2022           10
## 17       wmeduardowm  May 6, 2019            9
## 18         Permalink Jul 10, 2022            9
## 19    Leofwine_draca May 26, 2019           10
## 20         Permalink May 15, 2019            7
##                                                                review_titles
## 1                                                               User reviews
## 2                                                          They got it right
## 3                                                      Goosebumps and tears
## 4                                           I highly recommend this film!
## 5                                         No hero wakes up wanting to die
## 6                                                  So far looks excellent
## 7                                                              Incredible
## 8                                 Bleak, Unsettling, Haunting All Throughout
## 9                                                            Unbelievable
## 10                                                     HBO did it again!
## 11                                                             Exemplary
## 12                                                              Amazing!
## 13 The movie is far from thuth. A lot of fake info to create a drama...
## 14                                                   Emotionally drained...
## 15                                                        Just watch it (!)
## 16                                  Now you look like the minister of coal!
## 17                                                             Must Watch!
## 18                                                               Cracking.
## 19                                                              Brilliant!
## 20                  It is hard to overestimate the importance of this show.
##    helpful_review_counts not_helpful_review_counts
## 1                   <NA>                       <NA>
## 2                   <NA>                       <NA>
## 3                   <NA>                       <NA>
```

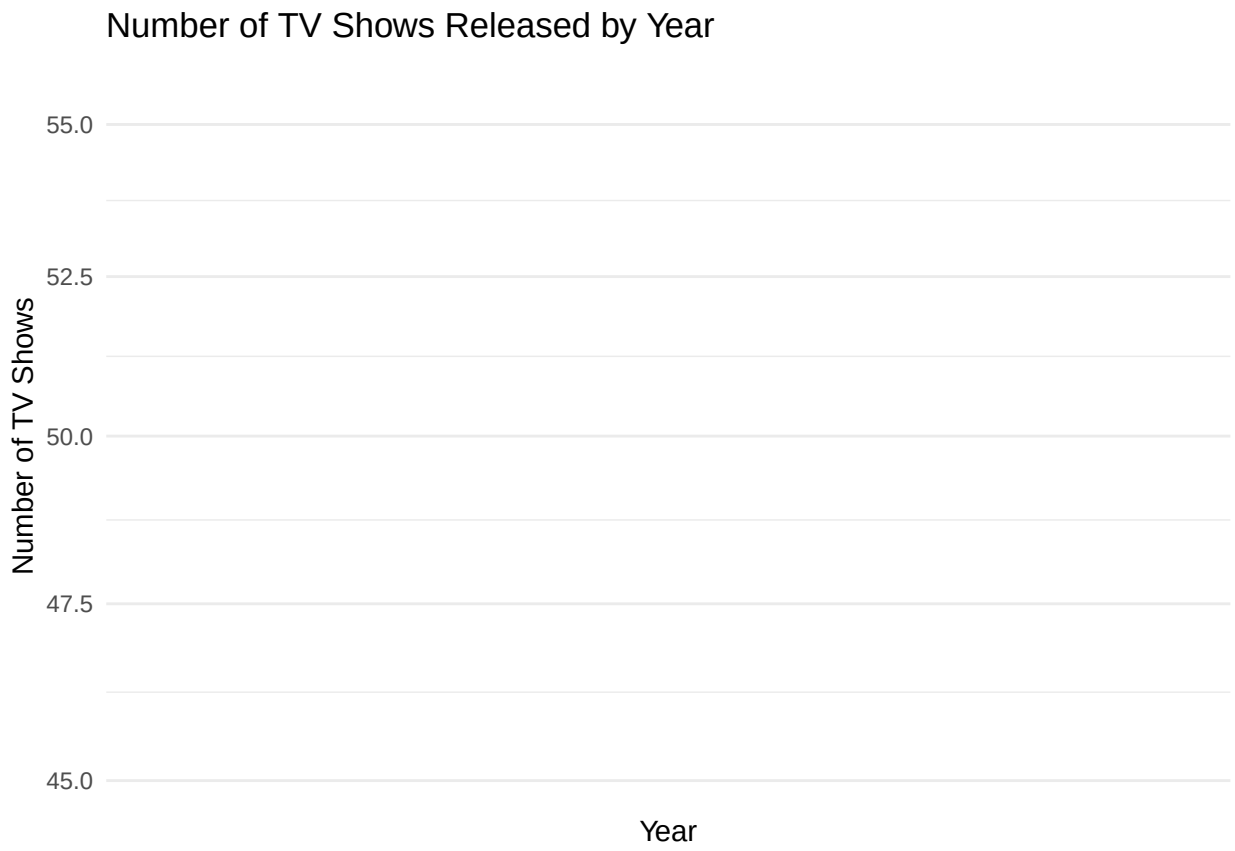
```
## 4                   <NA>                       <NA>
## 5                   <NA>                       <NA>
## 6                   <NA>                       <NA>
## 7                   <NA>                       <NA>
## 8                   <NA>                       <NA>
## 9                   <NA>                       <NA>
## 10                  <NA>                       <NA>
## 11                  <NA>                       <NA>
## 12                  <NA>                       <NA>
## 13                  <NA>                       <NA>
## 14                  <NA>                       <NA>
## 15                  <NA>                       <NA>
## 16                  <NA>                       <NA>
## 17                  <NA>                       <NA>
## 18                  <NA>                       <NA>
## 19                  <NA>                       <NA>
## 20                  <NA>                       <NA>
##
## 1
## 2
## 3
## 4  As my mother tells it, the weather was quite nice, the sky was clear without any sign of clouds i
## 5
## 6
## 7
## 8
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16
## 17
## 18
## 19
## 20
```

```r
tv_show_details$Year <- as.numeric(tv_show_details$Year)

shows_by_year <- tv_show_details %>%
  group_by(Year) %>%
  summarise(Count = n())

ggplot(shows_by_year, aes(x = Year, y = Count)) +
  geom_line(color = "yellow", size = 1) +
  geom_point(color = "green", size = 2) +
  labs(title = "Number of TV Shows Released by Year",
       x = "Year",
       y = "Number of TV Shows") +
  scale_y_log10() +
  theme_minimal()
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
```

```
## (`geom_line()`).
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```




```
most_shows_year <- shows_by_year %>%
  filter(Count == max(Count))

print(most_shows_year)
```

```
## # A tibble: 1 x 2
##    Year Count
##   <dbl> <int>
## 1    NA    50
```

```
# 4. Select 5 categories from Amazon and select 30 products from each category.

amazon_urls <- c('https://www.amazon.com/s?k=PC&crid=3O0HDEISP3NAZ&sprefix=pc%2Caps%2C610&ref=nb_sb_noss
                 'https://www.amazon.com/s?k=graphics+card&crid=4BJILUAEFT5I&sprefix=graphics%2Caps%2C3
                 'https://www.amazon.com/s?k=keyboard&crid=3647UWDU9H0TF&sprefix=keyboa%2Caps%2C331&ref=
                 'https://www.amazon.com/s?k=mouse&crid=QMZNC639VEYB&sprefix=mouse%2Caps%2C336&ref=nb_sb
                 'https://www.amazon.com/s?k=motherboard&crid=1ZHTORVTCHK1A&sprefix=motherb%2Caps%2C346
```

```
# 5. Extract the price, description, ratings and reviews of each product.

product_data <- list()

for (i in seq_along(amazon_urls)) {
```

```
  session <- bow(amazon_urls[i], user_agent = "Educational")

  product_name <- scrape(session) %>% html_nodes('h2.a-size-mini') %>% html_text() %>% head(30)

  product_description <- scrape(session) %>% html_nodes('div.productDescription') %>% html_text() %>% he

  product_rating <- scrape(session) %>% html_nodes('span.a-icon-alt') %>% html_text() %>% head(30)
  ratings <- as.numeric(str_extract(product_rating, "\\d+\\.\\d"))

  product_price <- scrape(session) %>% html_nodes('span.a-price') %>%  html_text() %>% head(30)
  price <- as.numeric(str_extract(product_price, "\\d+\\.\\d+"))

  product_review <- scrape(session) %>% html_nodes('div.review-text-content') %>% html_text() %>% head(3

  temp_dataframe <- data.frame(product_name = product_name[1:30],
                               product_description = product_description[1:30],
                               rating = ratings[1:30],
                               price = price[1:30],
                               stringsAsFactors = FALSE)

  product_data[[i]] <- temp_dataframe
}

print(product_data[[1]])
```

```
##
## 1                                      CyberPowerPC Gamer Xtreme VR Gaming PC, Intel Core i5-13400F 1
## 2           HP Elite Mini 800 G9 Business Desktop Computer, 14th Gen Intel 20-Core i7-14700 up to 5.4
## 3                                    CyberPowerPC Gamer Xtreme VR Gaming PC, Intel Core i7-14700F 2.1G
## 4      iBUYPOWER Slate 8 MESH Gaming PC Computer Desktop SMI7N47S01 (Intel Core i7 14700F CPU, NVIDIA
## 5          iBUYPOWER Y60 Black Gaming PC Computer Desktop Y60BA9N47TS03 (AMD Ryzen 9 7900X CPU, NVIDI
## 6                                                  Beelink AMD Ryzen 7 5800H Mini PC Up to 4
## 7                          HP Pro Mini 400 G9 Business Mini Desktop Computer, 12th Gen Intel Hexa-Core
## 8       DreamQuest Mini PC Windows 11 Home Preinstalled, Intel N100 (up to 3.4GHz) Mini Desktop Comput
## 9                            Beelink Mini PC AMD Ryzen 7 5800H Up to 4.4GHz 8C/16T, SER5 16GB RAM 1TB
## 10                                                                MXZ Desktop Gaming Comp
## 11 All-in-One PC All-in-one Desktop Computer PC 23.8inch All-in-One PC with Core i7 16GB RAM 512GB SS
## 12                                                          AOC AM16 Mini Gaming PC, Mini Computer
## 13                      ACEMAGICIAN Mini PC Computer, 12th Gen N100 (up to 3.4GHz) 16GB LPDDR5 512GB M
## 14                  Mini PC with Intel Alder Lake N97 CPU (up to 3.6GHz), Mini Desktop Computer PC Wi
## 15                                              CyberpowerPC Gamer Master Gaming PC, AMD Ryzen 5
## 16   SAAV X2 Prebuilt Gaming PC Desktop - Intel Core i5 3.4GHz, 16GB RAM, AMD RX580 8GB GDDR5, 512GB
## 17                            Skytech Archangel Gaming PC Desktop, Ryzen 5 5500 3.6 GHz (4.2GHz Turbo M
## 18              Dell Optiplex Small Desktop Computer (SFF) PC | Quad Core Intel i5 (3.2GHz) | 16GB DD
## 19                                             Thermaltake LCGS Quartz i460 R4 Gaming Desktop (Int
## 20   STGAubron Gaming Desktop PC Computer, Intel Core I7 3.4 GHz up to 3.9 GHz, Radeon RX 580 8G GDDR
## 21   GEEKOM GT1 Mega AI Mini PC,14th Gen Intel Core Ultra U9-185H Processor (16C/22T,up to 5.1 GHz),3
## 22                         STGAubron Prebuilt Gaming PC Desktop, AMD Radeon RX 550 4G GDDR5, Intel Co
## 23                                  Dell OptiPlex Computer Desktop PC, Intel Core i5 3rd Gen 3.2 G
## 24                          Skytech Gaming Nebula Gaming PC Desktop - AMD Ryzen 5 3600 3.6 GHz, NVIDIA
## 25                                                Gaming PC AMD Ryzen5 5600G 6core 3.9GHz 16GB DD
## 26                                                     Beelink AMD Ryzen 7 5800H Mini PC Up to 4
## 27 Blackout Computers Gaming Desktop PC Computer, Intel Core i7 3.6 GHz up to 4.0 GHz,AMD Radeon RX 5
```

```
## 28                                          Dell Optiplex 7050 SFF Desktop PC Intel i7-7700 4-Cores 3
## 29        Beelink EQR6 Mini PC AMD Ryzen 5 6600H(Up to 4.5GHz), 16GB DDR5 RAM 500GB PCIE4.0 SSD Dual
## 30             Cat 6 Outdoor Ethernet Cable 100 ft, 24AWG 10Gbps Cat6 Cable Cord Waterproof Direct Bu
##    product_description rating  price
## 1                 <NA>    4.1 759.99
## 2                 <NA>    4.6 899.99
## 3                 <NA>    4.4 832.15
## 4                 <NA>    4.4 979.00
## 5                 <NA>    4.3  59.99
## 6                 <NA>    4.6 249.99
## 7                 <NA>    4.8 274.99
## 8                 <NA>    4.7 499.99
## 9                 <NA>    4.6 899.99
## 10                <NA>    4.2 339.00
## 11                <NA>    4.3 459.00
## 12                <NA>    4.1 169.00
## 13                <NA>    4.4 229.99
## 14                <NA>    4.4 299.00
## 15                <NA>    4.6 389.00
## 16                <NA>    4.8 468.00
## 17                <NA>    4.7 410.00
## 18                <NA>    3.7 299.00
## 19                <NA>    4.5 499.00
## 20                <NA>    3.7 169.00
## 21                <NA>    4.4 219.00
## 22                <NA>    4.0 129.99
## 23                <NA>    3.6 170.99
## 24                <NA>    4.5 529.99
## 25                <NA>    4.2 779.99
## 26                <NA>    4.6 207.89
## 27                <NA>    4.4 799.99
## 28                <NA>    3.9 899.99
## 29                <NA>    4.5 449.99
## 30                <NA>    4.4 989.00
```

```
print(product_data[[2]])
```

```
##
## 1                                          maxsun AMD Radeon RX 550 4GB GDDR5 ITX Computer PC Gaming
## 2                                               KAER RX580 8GB Graphics Card GDDR5 256bit Computer G
## 3                                                          GIGABYTE Radeon RX 7600 XT Ga
## 4              MSI GeForce RTX 4070 Ti Super 16G Ventus 3X Black OC Graphics Card (NVIDIA RTX 4070 T
## 5                                                                                  MSI Gamin
## 6                                                 QTHREE GT 730 4GB 64Bit DDR3 Graphics Card,2X HD
## 7                                                       QTHREE GT 210 1024 MB DDR3 Graphics Card
## 8                                 AISURIX RX 580 Graphics Card, 2048SP, Real 8GB, GDDR5, 256 Bit, Pc G
## 9       Radeon RX 580 8GB GDDR5 Graphics Card White PCB for Gaming PC Video Card 2048SP 256-Bit PCIe
## 10                                                                    GIGABYTE GeForce RTX 4060
## 11                                                                 ASUS ProArt GeForce RTX  4060
## 12                                                                 GIGABYTE GeForce RTX 3050 WIND
## 13  ASUS Dual NVIDIA GeForce RTX 3050 6GB OC Edition Gaming Graphics Card - PCIe 4.0, 6GB GDDR6 Memor
## 14                                                          ASUS TUF Gaming GeForce RTX  4070 Ti Sup
## 15          ASUS Dual GeForce RTX  4070 Super EVO OC Edition 12GB GDDR6X (PCIe 4.0, 12GB GDDR6X, DLSS
## 16                                              RX 580 8GB Graphics Card, 2048SP,GDDR5,256 Bit Graph
## 17                                                          GIGABYTE GeForce RTX 3060 Gaming OC 12
```

```
## 18 ASUS Dual GeForce RTX  4060 EVO OC Edition 8GB GDDR6 (PCIe 4.0, 8GB GDDR6, DLSS 3, HDMI 2.1a, Disp
## 19                              MSI Gaming GeForce RTX 4070 Ventus 3X E1 12G OC Graphics Card (Ada L
## 20                                                                                  ASUS ProArt GeForce RT
## 21                                       RX 580 Graphics Card 8GB 2048SP, GDDR5, 256 Bit, Pc Gaming
## 22
## 23                                                                                        XFX Speedster QI
## 24                                                                                GIGABYTE Radeon RX 7800 XT G
## 25                                                                              RX 470 4GB GDDR5 256bit Gaming Gr
## 26                      RX 5500 XT 8gb GDDR6 Graphics Card,128 Bit, 2XDP, HDMI, PCI Express 4.0X8, 8pi
## 27
## 28
## 29
## 30
##    product_description rating  price
## 1                 <NA>    4.3  83.99
## 2                 <NA>    4.3 109.98
## 3                 <NA>    4.5 279.97
## 4                 <NA>    4.4 329.99
## 5                 <NA>    4.7 739.99
## 6                 <NA>    4.0 839.99
## 7                 <NA>    4.5 279.99
## 8                 <NA>    4.2  89.99
## 9                 <NA>    4.2  35.99
## 10                <NA>    4.7  89.99
## 11                <NA>    4.7  95.95
## 12                <NA>    4.4 284.99
## 13                <NA>    4.6 319.99
## 14                <NA>    4.7 449.99
## 15                <NA>    4.8 529.99
## 16                <NA>    4.0 152.97
## 17                <NA>    4.7 179.99
## 18                <NA>    4.7 159.99
## 19                <NA>    5.0 179.99
## 20                <NA>    4.6 799.99
## 21                <NA>    4.8 889.99
## 22                <NA>    4.5 599.99
## 23                <NA>    4.6  94.99
## 24                <NA>    3.0 279.99
## 25                <NA>    4.2 319.99
## 26                <NA>     NA 304.99
## 27                <NA>     NA 499.99
## 28                <NA>     NA 564.99
## 29                <NA>     NA 149.97
## 30                <NA>     NA  99.95
```

```
print(product_data[[3]])
```

```
##
## 1
## 2                                        DIERYA T68SE 60% Gaming Mechanical Keyboard,Ultra Compact M
## 3              Logitech MX Keys S Wireless Keyboard, Low Profile, Fluid Precise Quiet Typing, Progra
## 4                                         SteelSeries Apex Pro TKL HyperMagnetic Gaming Keyboard - A
## 5                               Rii RK100+ Multiple Color Rainbow LED Backlit Large Size USB Wired Mecha
## 6      Razer Huntsman V3 Pro TKL Esports Gaming Keyboard: Analog Optical Switches - Razer Snap Tap -
## 7                       ASUS ROG Strix Scope II 96 Wireless Gaming Keyboard, Tri-Mode Connection, Dampe
```

```
## 8  AULA Keyboard, T102 104 Keys Gaming Keyboard and Mouse Combo with RGB Backlit Quiet Computer Keybo
## 9              AULA F75 75% Wireless Mechanical Keyboard,Gasket Hot Swappable Custom Keyboard,Pre-lub
## 10
## 11
## 12                                                                    Redragon S101 Gaming Keyboard, M60
## 13
## 14                                                                                                  Ri
## 15          AULA F75 Pro Wireless Mechanical Keyboard,75% Gasket Hot Swappable Custom Keyboard,RGB Ba
## 16             AULA F75 75% Wireless Mechanical Keyboard,Gasket Hot Swappable Custom Keyboard,Pre-lub
## 17                                                 65% Gaming Keyboard, Wired Backlit Mini Keyboard,
## 18                    AULA F2088 Typewriter Style Mechanical Gaming Keyboard Blue Switches,Rainbow LE
## 19                       SABLUTE Large Print Backlit Computer Keyboards, Wired Lighted USB Keyboards
## 20                     RK ROYAL KLUDGE S98 Mechanical Keyboard w/Smart Display & Knob, Top Mount 96%
## 21                                                  MageGee Mechanical Gaming Keyboard with Blue Sw
## 22                                                                       Amazon Basics Wireless K
## 23
## 24     iClever BK10 Bluetooth Keyboard, Multi Device Keyboard Rechargeable Bluetooth 5.1 with Number
## 25                                    MageGee Portable 60% Mechanical Gaming Keyboard, MK-Box LED Bac
## 26                                                                                                  An
## 27                                                                  Verbatim Wired USB Computer
## 28                                      FOPETT 2.4GHz Wireless Keyboard and Mouse Set with Switch But
## 29                          FOPETT Wireless Keyboard and Mouse Combo, 104 Keys Full-Sized 2.4 GHz Rou
## 30                                                                       Gaming Keyboard and M
##    product_description rating  price
## 1                 <NA>    4.5  11.29
## 2                 <NA>    4.6  13.30
## 3                 <NA>    4.6  23.69
## 4                 <NA>    4.5  27.88
## 5                 <NA>    4.4 108.19
## 6                 <NA>    4.2 199.99
## 7                 <NA>    4.6 219.99
## 8                 <NA>    4.4 149.99
## 9                 <NA>    4.7 179.99
## 10                <NA>    4.5  25.49
## 11                <NA>    4.5  39.99
## 12                <NA>    4.6  63.11
## 13                <NA>    4.5  79.99
## 14                <NA>    4.4  11.29
## 15                <NA>    4.7  13.30
## 16                <NA>    4.7  93.15
## 17                <NA>    4.2   9.32
## 18                <NA>    4.6  31.99
## 19                <NA>    4.3  56.26
## 20                <NA>    4.6  13.99
## 21                <NA>    4.4  14.99
## 22                <NA>    4.2   9.49
## 23                <NA>    4.4  14.99
## 24                <NA>    4.3  66.31
## 25                <NA>    4.3  82.89
## 26                <NA>    4.1  63.11
## 27                <NA>    4.5  79.99
## 28                <NA>    4.2  13.97
## 29                <NA>    4.3  15.97
## 30                <NA>    4.4  39.19
```

```
print(product_data[[4]])
```

```
##
## 1                TMKB Falcon M1SE Ultralight Honeycomb Gaming Mouse, High-Precision 12800DPI Optical
## 2   E-YOOSO Wireless Mouse, Computer Mouse 18 Months Battery Life Cordless Mouse, 5 Level 4800 DPI, G
## 3                                                  Logitech M185 Wireless Mouse, 2.4GHz with USB Mi
## 4                                                     Razer DeathAdder Essential Gaming Mouse: 6400
## 5     Razer Basilisk V3 Customizable Ergonomic Gaming Mouse: Fastest Gaming Mouse Switch - Chroma RGB
## 6                                  Logitech G502 X Lightspeed Wireless Gaming Mouse - LIGHTFORCE H
## 7                                                      Lenovo 300 Wireless Mouse -
Computer Mouse for PC, Laptop with Windows - Ambidextrous Design - 2.4 GHz Nano USB Receiver -
12 Month Battery Life
## 8                                                  Logitech MX Ergo Wireless Trackball Mouse, Ergo
## 9                                                  Lenovo 100 Wired USB Computer Mouse for PC, Lap
Black
## 10                                                                                      Amazon Bas
## 11                                                                                    Amazon Basic
## 12                                                                          VssoPlor Wireless Mouse, 2
## 13                TECKNET Wireless Mouse, 2.4G Ergonomic Optical Mouse, Computer Mouse for Laptop
## 14                        Redragon M612 Predator RGB Gaming Mouse, 8000 DPI Wired Optical Gamer M
## 15                                                 EVGA X12 Gaming Mouse, 8k, Wired, White, Customi
## 16                     SM600 White Wireless Gaming Mouse,8000 DPI Tri-Modes BT5.1/Type-C Wired/2.4
## 17                                Razer Basilisk V3 X HyperSpeed Wireless Gaming Mouse: Up to 285
## 18
## 19                                                                                    Amazon Basic
## 20          Razer Viper V3 HyperSpeed Wireless Esports Gaming Mouse: 82g Lightweight - Up to 280 Hr H
## 21                                             Glorious Model O- (Minus) Compact Wired Gaming Mouse -
## 22                                                Logitech MX Vertical Wireless Mouse -
Ergonomic Design Reduces Muscle Strain, Move Content Between 3 Windows and Apple Computers, Rechargeable
## 23        Wireless Bluetooth Mouse Rechargeable LED Silent Slim Laptop Mouse Portable(Bluetooth5.2 an
## 24
## 25 TECKNET Bluetooth Mouse, 4000DPI Wireless Mouse Bluetooth for Laptop 2-in-1(BT 5.0/3.0+2.4Ghz) Com
## 26                                                                                      Amazon Bas
## 27
## 28
## 29
## 30
##    product_description rating price
## 1                 <NA>    4.5 15.99
## 2                 <NA>    4.4 19.99
## 3                 <NA>    4.5  9.49
## 4                 <NA>    4.4 19.99
## 5                 <NA>    4.6 13.97
## 6                 <NA>    4.6 14.99
## 7                 <NA>    4.6 20.98
## 8                 <NA>    4.5 29.99
## 9                 <NA>    4.5 39.98
## 10                <NA>    4.3 69.99
## 11                <NA>    4.6 82.99
## 12                <NA>    4.6  2.36
## 13                <NA>    4.5 99.99
## 14                <NA>    4.6  7.99
## 15                <NA>    4.2  8.99
## 16                <NA>    4.2  8.09
```

```
## 17                <NA>    4.3 12.99
## 18                <NA>    4.6  7.19
## 19                <NA>    4.6  9.98
## 20                <NA>    4.3 12.99
## 21                <NA>    4.6  8.49
## 22                <NA>    4.4 19.99
## 23                <NA>    4.2 14.99
## 24                <NA>    4.6 24.99
## 25                <NA>    4.4  8.99
## 26                <NA>    4.3 29.99
## 27                <NA>     NA 21.50
## 28                <NA>     NA 29.89
## 29                <NA>     NA 49.00
## 30                <NA>     NA 69.99
```

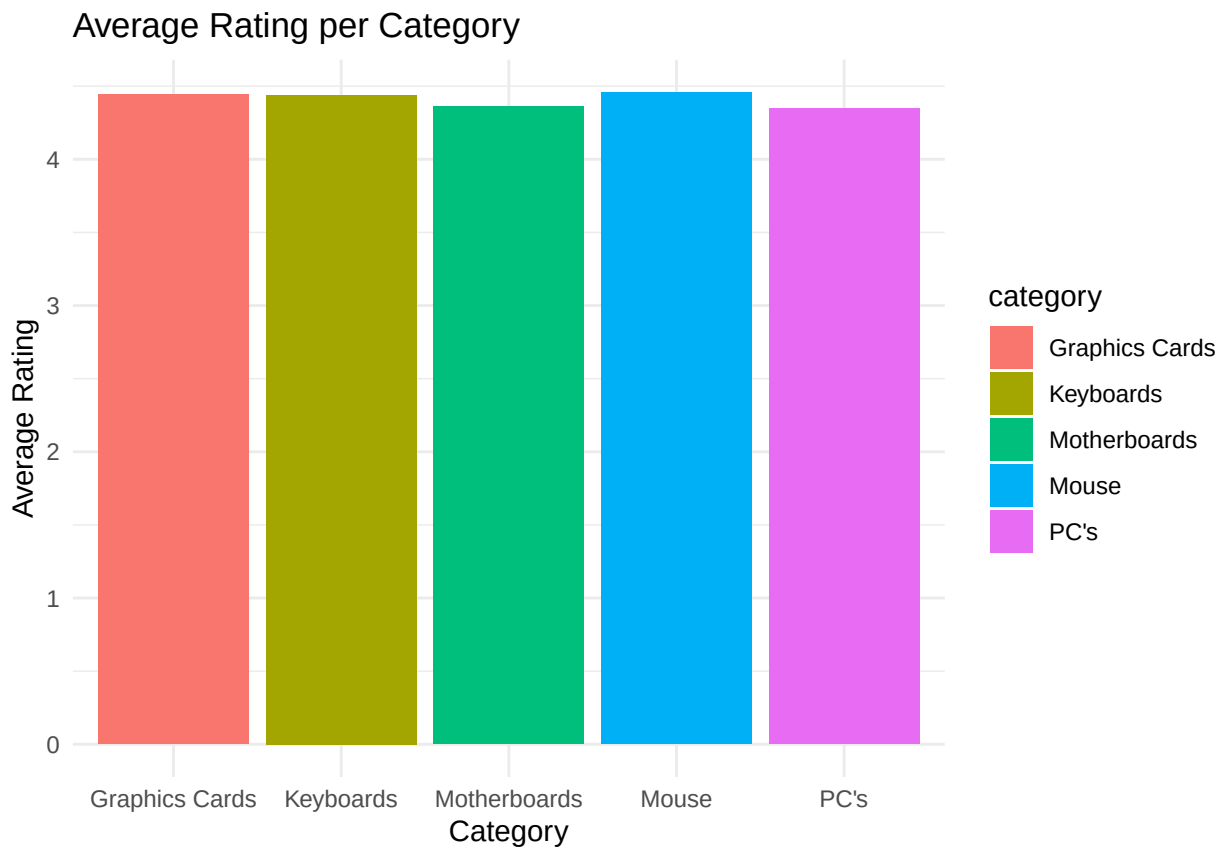
```
print(product_data[[5]])
```

```
##
## 1                                    ASUS ROG Strix B650-A Gaming WiFi 6E AM5 (LGA1718) Ryzen 7000 M
## 2   ASUS TUF Gaming X870-PLUS WiFi AMD AM5 X870 ATX Motherboard, 16+2+1, 80A SPS Power Stages, DDR5,
## 3                          Asus ROG Strix B550-F Gaming WiFi II AMD AM4 (3rd Gen Ryzen) ATX Motherboa
## 4                                                MSI MAG B550 TOMAHAWK Gaming Motherboard (AM
## 5  ASUS TUF Gaming Z790-Plus WiFi LGA 1700(Intel 14th,12th &13th Gen) ATX Gaming Motherboard(PCIe 5.0
## 6                           GIGABYTE B760M Gaming Plus WiFi DDR4 LGA 1700 Intel B760 M-ATX Motherboar
## 7                               GIGABYTE Z790 Gaming Plus AX LGA 1700 Intel Z790 ATX Motherboard with
## 8                                    ASUS TUF Gaming B550-PLUS WiFi II AMD AM4 (3rd Gen Ryzen ) ATX M
## 9                                                MSI MAG B550 TOMAHAWK Gaming Motherboard (AM
## 10        ASUS Prime B550-PLUS AC-HES AMD AM4 (3rd Gen Ryzen) ATX Motherboard (Dual M.2, PCIe4.0, WI
## 11                                   ASUS ROG Strix B650-A Gaming WiFi 6E AM5 (LGA1718) Ryzen 7000 M
## 12 ASUS TUF Gaming Z790-Plus WiFi LGA 1700(Intel 14th,12th &13th Gen) ATX Gaming Motherboard(PCIe 5.0
## 13  ASUS TUF Gaming X870-PLUS WiFi AMD AM5 X870 ATX Motherboard, 16+2+1, 80A SPS Power Stages, DDR5,
## 14                                                                GIGABYTE B650 Eagle AX AM5
## 15
## 16                                                            MSI B550 Gaming GEN3 Gaming Mother
## 17            MSI B760 Gaming Plus WiFi Gaming Motherboard (Supports 12th/13th/14th Gen Intel Proces
## 18          ASUS Prime B550-PLUS AMD AM4 Zen 3 Ryzen 5000 & 3rd Gen Ryzen ATX Motherboard (PCIe 4.0,
## 19                          GIGABYTE B760M Gaming Plus WiFi DDR4 LGA 1700 Intel B760 M-ATX Motherboar
## 20        ASUS Prime B550-PLUS AC-HES AMD AM4 (3rd Gen Ryzen) ATX Motherboard (Dual M.2, PCIe4.0, WI
## 21     MSI MAG X670E Tomahawk WiFi Gaming Motherboard (AMD Ryzen 9000/8000/7000 Series Processors, AM
## 22      MSI MAG B650 Tomahawk WiFi Gaming Motherboard (AMD Ryzen 9000/8000/7000 Series Processors, AM
## 23   ASUS ROG Strix X870-A Gaming WiFi AMD AM5 X870 ATX Motherboard 16+2+2 Power Stages, Dynamic OC S
## 24              ASUS ROG Strix Z790-A Gaming WiFi II (WiFI 7) LGA 1700(Intel 14th & 13th & 12th Gen)
## 25                                                                                      ASUS Z
## 26                                                                          MSI B550-A PRO Pro
## 27                                                                               Asrock B760M
## 28
## 29                                                                                     SAMA 24
## 30                          MEIRIYFA Motherboard USB3.0 19PIN Splitter 1 to 2 Extension Cable,USB 3.0 1
##    product_description rating  price
## 1                 <NA>    4.2 209.99
## 2                 <NA>    4.6 249.99
## 3                 <NA>    4.6 274.99
## 4                 <NA>    4.6 309.99
## 5                 <NA>    4.4 159.99
## 6                 <NA>    5.0 189.99
```

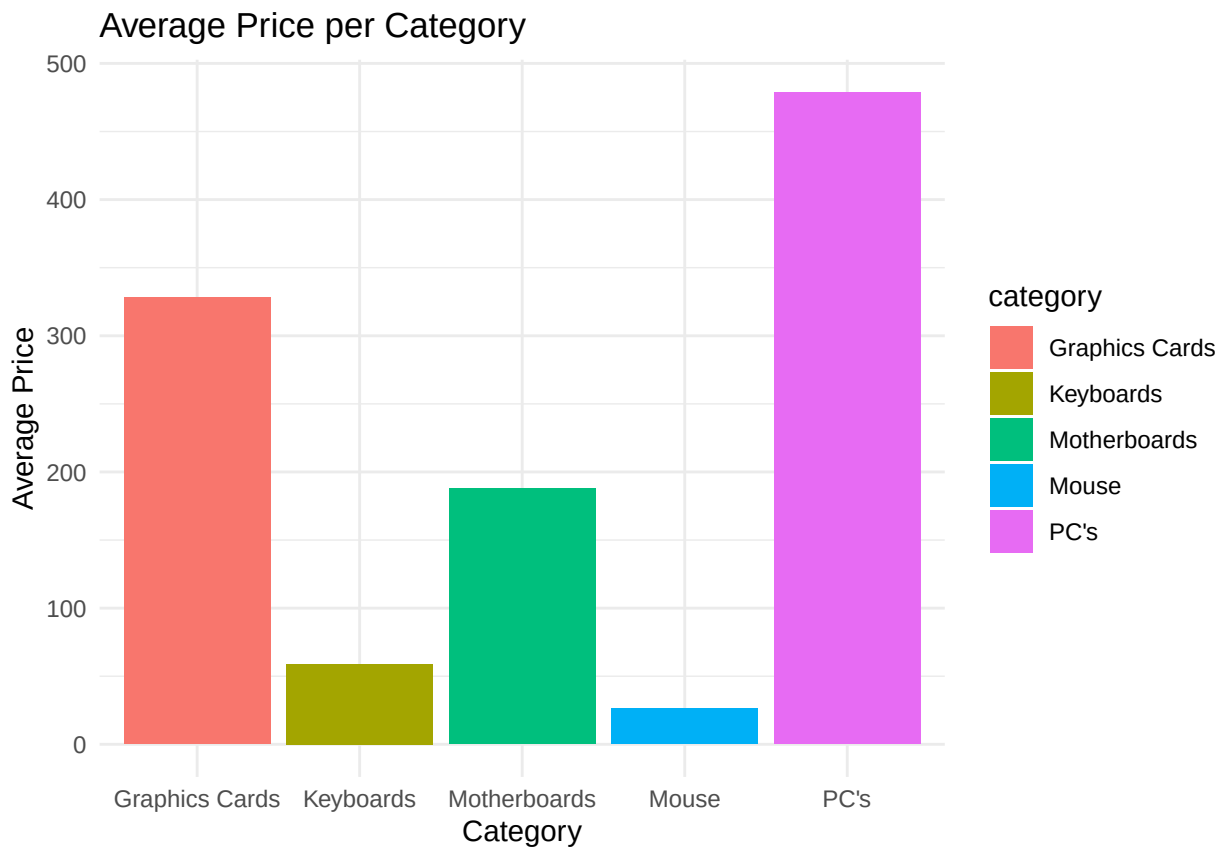
```
## 7                 <NA>    5.0 109.99
## 8                 <NA>    4.6 159.99
## 9                 <NA>    4.6 194.99
## 10                <NA>    4.5 249.99
## 11                <NA>    4.2  99.99
## 12                <NA>    4.4 149.99
## 13                <NA>    4.6 139.99
## 14                <NA>    4.3 169.99
## 15                <NA>    4.1 119.99
## 16                <NA>    4.5 149.99
## 17                <NA>    4.4 109.99
## 18                <NA>    4.5 159.99
## 19                <NA>    5.0  99.99
## 20                <NA>    4.5 119.99
## 21                <NA>    4.2 209.99
## 22                <NA>    4.2 249.99
## 23                <NA>    4.2 194.99
## 24                <NA>    4.4 249.99
## 25                <NA>    4.3 274.99
## 26                <NA>    4.5 309.99
## 27                <NA>    5.0 139.99
## 28                <NA>    4.6 179.99
## 29                <NA>    3.8 174.99
## 30                <NA>    1.0 219.99
```

```r
# 6. Describe the data you have extracted.

# The code collects data from Amazon product listings across several categories, such as "PCs," "Graphi

# 7. What will be your use case for the data you have extracted?

# This data can be used to compare the popularity of different products, track pricing trends, analyze
```

```r
# 8. Create graphs regarding the use case. And briefly explain it.

combined_product_data <- do.call(rbind, product_data)
combined_product_data$category <- rep(c("PC's", "Graphics Cards", "Keyboards", "Mouse", "Motherboards")

avg_rating <- combined_product_data %>%
  group_by(category) %>%
  summarize(average_rating = mean(rating, na.rm = TRUE))

ggplot(avg_rating, aes(x = category, y = average_rating, fill = category)) +
  geom_bar(stat = "identity") +
  labs(title = "Average Rating per Category", x = "Category", y = "Average Rating") +
  theme_minimal()
```

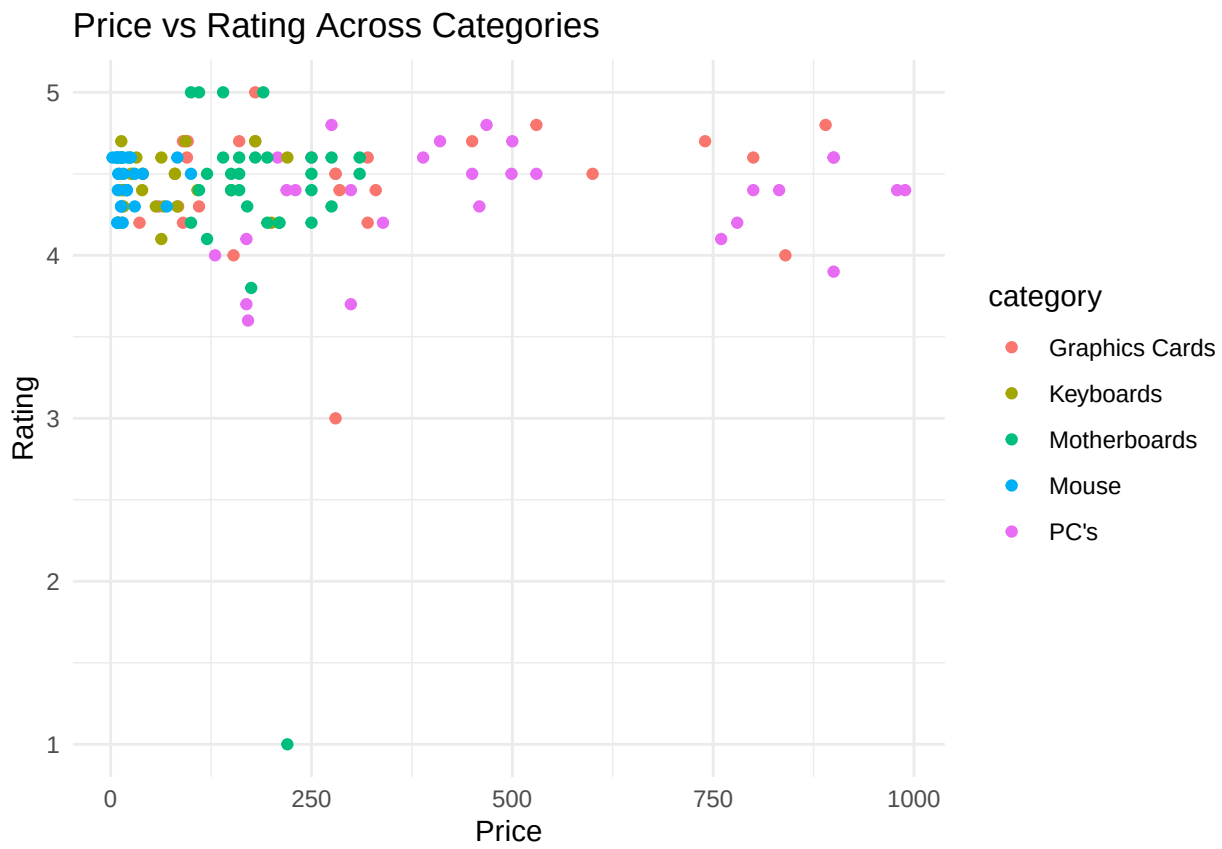



```r
avg_price <- combined_product_data %>%
  group_by(category) %>%
  summarize(average_price = mean(price, na.rm = TRUE))

ggplot(avg_price, aes(x = category, y = average_price, fill = category)) +
  geom_bar(stat = "identity") +
  labs(title = "Average Price per Category", x = "Category", y = "Average Price") +
  theme_minimal()
```



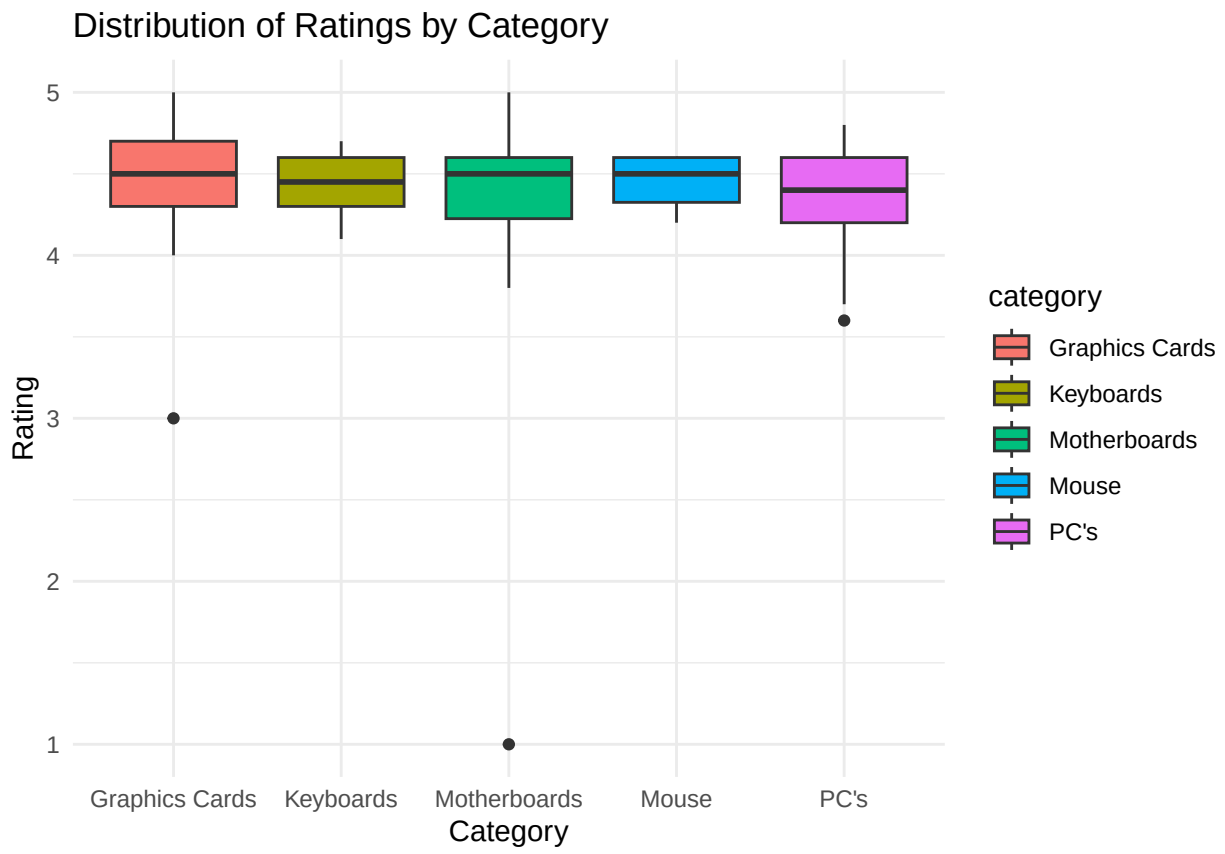

```
ggplot(combined_product_data, aes(x = price, y = rating, color = category)) +
  geom_point() +
  labs(title = "Price vs Rating Across Categories", x = "Price", y = "Rating") +
  theme_minimal()
```

```
## Warning: Removed 9 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

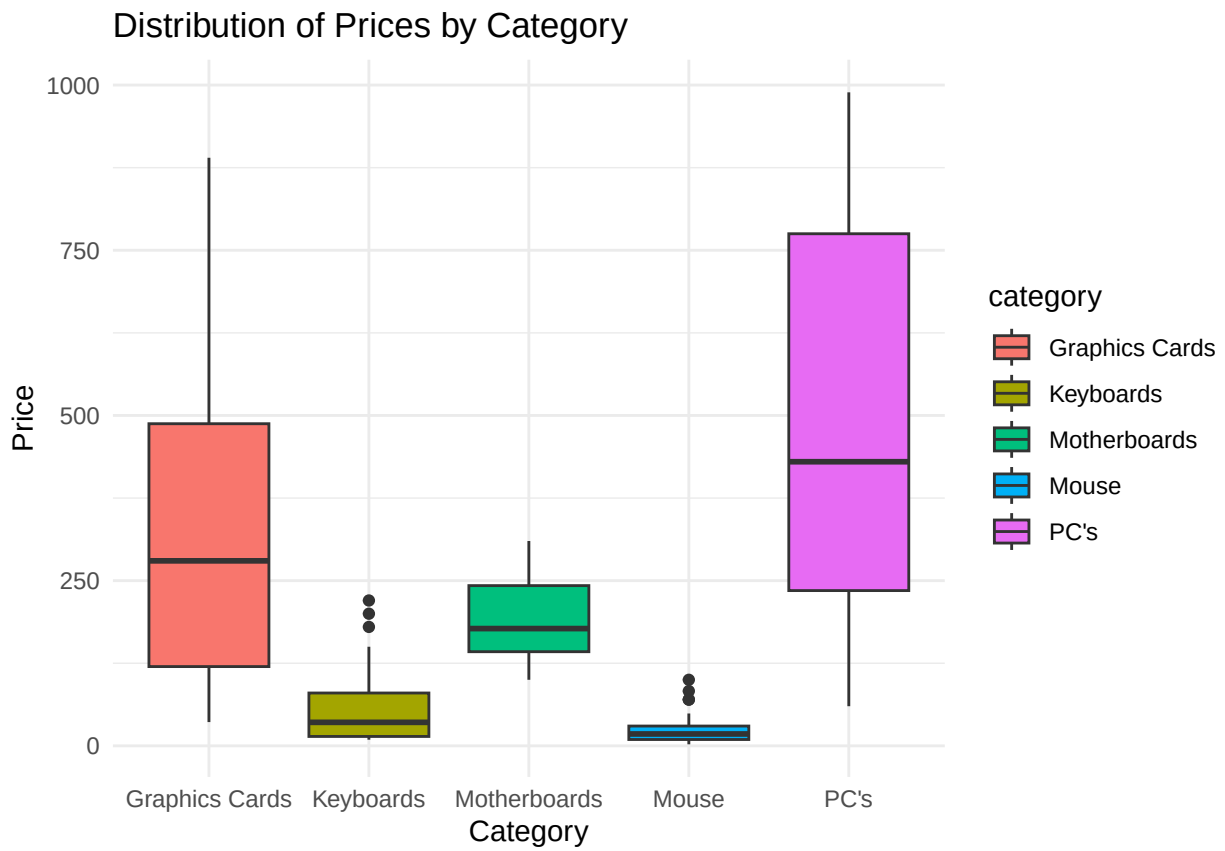



```
# 9. Graph the price and the ratings for each category. Use basic plotting functions and ggplot2 packag

ggplot(combined_product_data, aes(x = category, y = rating, fill = category)) +
  geom_boxplot() +
  labs(title = "Distribution of Ratings by Category", x = "Category", y = "Rating") +
  theme_minimal()
```

```
## Warning: Removed 9 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```




```r
ggplot(combined_product_data, aes(x = category, y = price, fill = category)) +
  geom_boxplot() +
  labs(title = "Distribution of Prices by Category", x = "Category", y = "Price") +
  theme_minimal()
```




```r
# 10. Rank the products of each category by price and ratings. Explain briefly.

ranked_product_data <- lapply(product_data, function(df_category) {
  df_category %>%
    arrange(desc(rating), price) %>%
    mutate(rank = row_number()) %>%
    select(rank, everything())
})

categories <- c("PC's", "Graphics Cards", "Keyboards", "Mouse", "Motherboards")
for (i in seq_along(ranked_product_data)) {
  ranked_product_data[[i]]$category <- categories[i]
}

ranked_combined_product_data <- do.call(rbind, ranked_product_data)
ranked_combined_product_data <- ranked_combined_product_data %>%
  arrange(category, rank) %>%
  group_by(category) %>%
  slice(1:5)

print(ranked_combined_product_data)
```

```
## # A tibble: 25 x 6
## # Groups:   category [5]
##     rank product_name                product_description rating price category
##    <int> <chr>                       <chr>                <dbl> <dbl> <chr>
```

```
##  1     1 "MSI Gaming GeForce RTX 4070~ <NA>                5   180.  Graphic~
##  2     2 "ASUS Dual GeForce RTX  4070~ <NA>              4.8 530.  Graphic~
##  3     3 "RX 580 Graphics Card 8GB 20~ <NA>              4.8 890.  Graphic~
##  4     4 "GIGABYTE GeForce RTX 4060 G~ <NA>              4.7  90.0 Graphic~
##  5     5 "ASUS ProArt GeForce RTX  40~ <NA>              4.7  96.0 Graphic~
##  6     1 "AULA F75 Pro Wireless Mecha~ <NA>              4.7  13.3 Keyboar~
##  7     2 "AULA F75 75% Wireless Mecha~ <NA>              4.7  93.2 Keyboar~
##  8     3 "AULA F75 75% Wireless Mecha~ <NA>              4.7 180.  Keyboar~
##  9     4 "DIERYA T68SE 60% Gaming Mec~ <NA>              4.6  13.3 Keyboar~
## 10     5 "RK ROYAL KLUDGE S98 Mechani~ <NA>              4.6  14.0 Keyboar~
## # i 15 more rows
```