RWorksheet#5_group(Corvera, Paclibar, Sabarillo)

Rotciv Corvera, Jhon Albert Paclibar, Kirk Axl Dend Sabarillo

2024-11-11

1. Extracting TV Shows

```
library(polite)
library(httr)
library(rvest)
library(dplyr)
```

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

url <- "https://www.imdb.com/chart/toptv/?sort=rank%2Casc"
#1
#get the ranks and titles
title_list <- read_html(url) %>%
  html_nodes('.ipc-title__text') %>%
  html_text()
```

```
#Clean extracted text
title_list_sub <- as.data.frame(title_list[3:27], stringsAsFactors = FALSE)
colnames(title_list_sub) <- "ranks"
split_df <- strsplit(as.character(title_list_sub$ranks), "\\.", fixed = FALSE)
split_df <- data.frame(do.call(rbind, split_df), stringsAsFactors = FALSE)

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

split_df$title <- trimws(split_df$title)

rank_title <- split_df
```

```
#get tv rating, the number of people who voted, the number of episodes, and the year it was released.
rating_ls <- read_html(url) %>%
  html_nodes('.ipc-rating-star--rating') %>%
```

```
  html_text()

voter_ls <- read_html(url) %>%
  html_nodes('.ipc-rating-star--voteCount') %>%
  html_text()
clean_votes <- gsub('[()]', '', voter_ls)
```

```
#get the number of episodes
eps_ls <- read_html(url) %>%
  html_nodes('span.sc-5bc66c50-6.OOdsw.cli-title-metadata-item:nth-of-type(2)') %>%
  html_text()
clean_eps <- gsub('[eps]', '', eps_ls)
num_eps <- as.numeric(clean_eps)
```

```
#get year released
years <- read_html(url) %>%
  html_nodes('span.sc-5bc66c50-6.OOdsw.cli-title-metadata-item:nth-of-type(1)') %>%
  html_text()

top_tv_shows <- data.frame(
  Rank = rank_title[1],
  Title = rank_title[2],
  Rating = rating_ls,
  Voters = clean_votes,
  Episodes = num_eps,
  Year = years,
  stringsAsFactors = FALSE
)
```

```
#Number of user reviews
home_link <- 'https://www.imdb.com/chart/toptv/'
main_page <- read_html(home_link)

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

```
#get link of each show's page
show_data <- lapply(links, function(link) {
  complete_link <- paste0("https://imdb.com", link)

  #get the link for user review page
  usrv_link <- read_html(complete_link)
  usrv_link_page <- usrv_link %>%
    html_nodes('a.isReview') %>%
    html_attr("href")

  #get critic reviews
  critic <- usrv_link %>%
    html_nodes("span.score") %>%
    html_text()
  critic_df <- data.frame(Critic_Reviews = critic[2], stringsAsFactors = FALSE)

  #get pop rating
  pop_rating <- usrv_link %>%
```

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

  #get user reviews of each shows
  usrv <- read_html(paste0("https://imdb.com", usrv_link_page[1]))
  usrv_count <- usrv %>%
    html_nodes('[data-testid="tturv-total-reviews"]') %>%
    html_text()

  return(data.frame( User_Reviews = usrv_count, Critic = critic_df, Popularity_Rating = pop_rating))
})

 critics_df <- do.call(rbind, show_data)

shows <- cbind(top_tv_shows, critics_df)
shows
```

```
##    rank                           title Rating Voters Episodes      Year
## 1     1                     Breaking Bad    9.5   2.2M       62 2008-2013
## 2     2                  Planet Earth II    9.5   162K        6      2016
## 3     3                     Planet Earth    9.4   224K       11      2006
## 4     4                 Band of Brothers    9.4   546K       10      2001
## 5     5                        Chernobyl    9.3   907K        5      2019
## 6     6                         The Wire    9.3   391K       60 2002-2008
## 7     7       Avatar: The Last Airbender    9.3   390K       62 2005-2008
## 8     8                   Blue Planet II    9.3    49K        7      2017
## 9     9                     The Sopranos    9.2   498K       86 1999-2007
## 10   10      Cosmos: A Spacetime Odyssey    9.2   132K       13      2014
## 11   11                           Cosmos    9.3    46K       13      1980
## 12   12                       Our Planet    9.2    54K       12 2019-2023
## 13   13                  Game of Thrones    9.2   2.4M       74 2011-2019
## 14   14                            Bluey    9.3    34K      194    2018-
## 15   15                The World at War    9.2    31K       26 1973-1974
## 16   16 Fullmetal Alchemist: Brotherhood    9.1   209K       68 2009-2010
## 17   17                   Rick and Morty    9.1   627K       78    2013-
## 18   18                             Life    9.1    44K       11      2009
## 19   19                  The Last Dance    9.0   160K       10      2020
## 20   20               The Twilight Zone    9.0    97K      156 1959-1964
## 21   21                 The Vietnam War    9.1    30K       10      2017
## 22   22                        Sherlock    9.1     1M       15 2010-2017
## 23   23                  Attack on Titan    9.1   561K       98 2013-2023
## 24   24     Batman: The Animated Series    9.0   122K       85 1992-1995
## 25   25                          Arcane    9.0   304K       18 2021-2024
## 26    1                     Breaking Bad    9.5   2.2M       62 2008-2013
## 27    2                  Planet Earth II    9.5   162K        6      2016
## 28    3                     Planet Earth    9.4   224K       11      2006
## 29    4                 Band of Brothers    9.4   546K       10      2001
## 30    5                        Chernobyl    9.3   907K        5      2019
## 31    6                         The Wire    9.3   391K       60 2002-2008
## 32    7       Avatar: The Last Airbender    9.3   390K       62 2005-2008
## 33    8                   Blue Planet II    9.3    49K        7      2017
## 34    9                     The Sopranos    9.2   498K       86 1999-2007
## 35   10      Cosmos: A Spacetime Odyssey    9.2   132K       13      2014
## 36   11                           Cosmos    9.3    46K       13      1980
```

```
## 37   12                       Our Planet    9.2    54K       12 2019-2023
## 38   13                  Game of Thrones    9.2   2.4M       74 2011-2019
## 39   14                            Bluey    9.3    34K      194    2018-
## 40   15                The World at War    9.2    31K       26 1973-1974
## 41   16 Fullmetal Alchemist: Brotherhood    9.1   209K       68 2009-2010
## 42   17                  Rick and Morty    9.1   627K       78    2013-
## 43   18                            Life    9.1    44K       11      2009
## 44   19                  The Last Dance    9.0   160K       10      2020
## 45   20               The Twilight Zone    9.0    97K      156 1959-1964
## 46   21                 The Vietnam War    9.1    30K       10      2017
## 47   22                        Sherlock    9.1     1M       15 2010-2017
## 48   23                 Attack on Titan    9.1   561K       98 2013-2023
## 49   24     Batman: The Animated Series    9.0   122K       85 1992-1995
## 50   25                          Arcane    9.0   304K       18 2021-2024
##     User_Reviews Critic_Reviews Popularity_Rating
## 1  5,092 reviews            175                22
## 2  5,092 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   998 reviews             57               354
## 14   998 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 5,899 reviews            368                14
## 26 5,899 reviews            368                14
## 27   367 reviews              4               380
## 28   367 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   911 reviews             94               127
## 34   911 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 1,096 reviews            121              160
## 44 1,096 reviews            121              160
## 45 2,361 reviews             64               45
## 46 2,361 reviews             64               45
## 47   219 reviews             25              453
## 48   219 reviews             25              453
## 49 1,959 reviews             50                2
## 50 1,959 reviews             50                2
```

```r
#2.
# Define URL for Breaking Bad
BreakingBad_urls <- "https://www.imdb.com/title/tt0903747/reviews/?ref_=tt_ov_urv"

# Initialize list to store data frames
df <- list()
df_names <- "Breaking_Bad"

# Read HTML session for the current URL
session <- read_html(BreakingBad_urls)

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

# Scrape review dates
review_date <- session %>%
  html_nodes(".ipc-inline-list__item.review-date") %>%
  html_text() %>%
  head(20)

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

# Scrape reviews' titles
review_title <- session %>%
  html_nodes(".ipc-title__text") %>%
  html_text() %>%
  head(20)

# Scrape helpful reviews
helpful_reviews <- session %>%
  html_nodes(".ipc-voting__label__count.ipc-voting__label__count--up") %>%
  html_text() %>%
  head(20)

# Scrape not helpful reviews
```

```r
not_helpful_reviews <- session %>%
  html_nodes(".ipc-voting__label__count.ipc-voting__label__count--down") %>%
  html_text() %>%
  head(20)

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

# Ensure each column has exactly 20 entries, filling with NA if fewer than 20 were scraped
reviewer_name <- c(reviewer_name, rep(NA, 20 - length(reviewer_name)))[1:20]
review_date <- c(review_date, rep(NA, 20 - length(review_date)))[1:20]
user_rating <- c(user_rating, rep(NA, 20 - length(user_rating)))[1:20]
review_title <- c(review_title, rep(NA, 20 - length(review_title)))[1:20]
helpful_reviews <- c(helpful_reviews, rep(NA, 20 - length(helpful_reviews)))[1:20]
not_helpful_reviews <- c(not_helpful_reviews, rep(NA, 20 - length(not_helpful_reviews)))[1:20]
text_reviews <- c(text_reviews, rep(NA, 20 - length(text_reviews)))[1:20]

# Create a temporary data frame for the current URL
dfTemp <- data.frame(
  reviewer_name = reviewer_name,
  review_date = review_date,
  user_rating = user_rating,
  review_title = review_title,
  helpful_reviews = helpful_reviews,
  not_helpful_reviews = not_helpful_reviews,
  text_reviews = text_reviews,
  stringsAsFactors = FALSE

)

# Append the temporary data frame to the list with a custom name
df[[df_names]] <- dfTemp

# View the data frame 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 Nov 15, 2019          10
## 8           Permalink  Dec 8, 2022          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           5
## 13           xpinerhd Feb 20, 2021          10
## 14          Permalink  Dec 8, 2022          10
```

```
## 15            Rob1331 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                                                                                                           I
## 9
## 10                                                                                                          A
## 11                                                                                  Among the best and most ac
## 12                                                                                                          A
## 13
## 14                                                                                                      Slow
## 15                                                                                                      Once
## 16 If you mix Scarface, Robin Hood and maybe Tyler Durden with enough meth - you'll get a mean cockta
## 17                                                                                        By far the greatest
## 18                                                                                                in a ca
## 19                                                                  Since GOT is over, this is Officially the
## 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
# Define URL for Planet Earth II
PlanetEarthII_urls <- "https://www.imdb.com/title/tt5491994/reviews/?ref_=tt_ov_urv"

# Initialize list to store data frames
df <- list()
df_names <- "Planet_Earth_II"

# Read HTML session for the current URL
session <- read_html(PlanetEarthII_urls)

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

# Scrape review dates
review_date <- session %>%
  html_nodes(".ipc-inline-list__item.review-date") %>%
  html_text() %>%
  head(20)

# Scrape user ratings (update CSS selector)
# First, inspect the correct selector for user rating from the page structure.
user_rating <- session %>%
  html_nodes(".ipc-rating-star--rating") %>%  # Adjust this selector if needed (check the page source)
  html_text() %>%
  head(20)

# Scrape reviews' titles
review_title <- session %>%
  html_nodes(".ipc-title__text") %>%
  html_text() %>%
  head(20)

# Scrape helpful reviews
helpful_reviews <- session %>%
  html_nodes(".ipc-voting__label__count.ipc-voting__label__count--up") %>%
```

```
  html_text() %>%
  head(20)

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

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

# Handle case where some elements might be missing, ensuring we have exactly 20 entries
reviewer_name <- c(reviewer_name, rep(NA, 20 - length(reviewer_name)))[1:20]
review_date <- c(review_date, rep(NA, 20 - length(review_date)))[1:20]
user_rating <- c(user_rating, rep(NA, 20 - length(user_rating)))[1:20]
review_title <- c(review_title, rep(NA, 20 - length(review_title)))[1:20]
helpful_reviews <- c(helpful_reviews, rep(NA, 20 - length(helpful_reviews)))[1:20]
not_helpful_reviews <- c(not_helpful_reviews, rep(NA, 20 - length(not_helpful_reviews)))[1:20]
text_reviews <- c(text_reviews, rep(NA, 20 - length(text_reviews)))[1:20]

# Create a temporary data frame for the current URL
dfTemp <- data.frame(
  reviewer_name = reviewer_name,
  review_date = review_date,
  user_rating = user_rating,
  review_title = review_title,
  helpful_reviews = helpful_reviews,
  not_helpful_reviews = not_helpful_reviews,
  text_reviews = text_reviews,
  stringsAsFactors = FALSE
)

# Append the temporary data frame to the list with a custom name
df[[df_names]] <- dfTemp

# View the data frame for "Planet Earth II"
print(df$Planet_Earth_II)
```

```
##         reviewer_name  review_date user_rating
## 1        arjanhylkema  Nov 7, 2016          10
## 2           Permalink  Nov 5, 2016          10
## 3            Wentloog  Nov 5, 2016          10
## 4           Permalink  Nov 9, 2016          10
## 5       john-m-madsen  Nov 5, 2016          10
## 6           Permalink  Nov 8, 2016          10
## 7        thespookybuz Nov 17, 2016          10
## 8           Permalink Nov 13, 2016          10
## 9         pjdickinson  Nov 6, 2016          10
## 10          Permalink Dec 31, 2016          10
## 11           dbijis33 Nov 19, 2016          10
```

```
## 12          Permalink Dec 28, 2016          7
## 13     dhanrajjughead May 19, 2019         10
## 14          Permalink Sep 29, 2017         10
## 15        NeilBarnett Nov 22, 2016         10
## 16          Permalink Oct 12, 2017         10
## 17      salmanu-27386  Dec 4, 2016         10
## 18          Permalink Oct 20, 2018         10
## 19 panagiotiskatsanos Apr 23, 2020         10
## 20          Permalink  Jan 5, 2017         10
##
## 1
## 2
## 3                                                                                                  At once awe-inspiring a
## 4                                                                                 Yet another masterpiece from BBC Nature & Davi
## 5
## 6
## 7                                                                                                                  Dangero
## 8                                                                                                Greatest documentary
## 9                                                                                      Best thing on TV since las
## 10
## 11                                                                                         One of the best documentaries
## 12                                                                                                 In times of climat
## 13
## 14                                                                                 More irritated with IMDb for the bias thar
## 15                                                                                                 Should be required viev
## 16                                                                                                 What a Beautiful Planet
## 17 Like the first 'Planet Earth', does for nature and our planet as 'Walking with Dinosaurs' did with
## 18                                                                                               This masterpiece deserves
## 19
## 20                                                                                                                   Absol
##    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
## 11
## 12
## 13
## 14
## 15
## 16 Absolutely adore the first 'Planet Earth' from 2007, one of the best documentaries ever made and a
## 17
## 18
## 19
## 20
```

```r
# Define URL for Planet Earth
PlanetEarth_urls <- "https://www.imdb.com/title/tt0795176/reviews/?ref_=tt_ov_urv"

# Initialize list to store data frames
df <- list()
df_names <- "Planet_Earth"

# Read HTML session for the current URL
session <- read_html(PlanetEarth_urls)

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

# Scrape review dates
review_date <- session %>%
  html_nodes(".ipc-inline-list__item.review-date") %>%
  html_text() %>%
  head(20)

# Scrape user ratings (corrected CSS selector)
user_rating <- session %>%
  html_nodes(".ipc-rating-star--rating") %>%  # Adjust this selector if needed (inspect page for correc
  html_text() %>%
  head(20)

# Scrape reviews' titles
review_title <- session %>%
  html_nodes(".ipc-title__text") %>%
  html_text() %>%
  head(20)

# Scrape helpful reviews
```

```r
helpful_reviews <- session %>%
  html_nodes(".ipc-voting__label__count.ipc-voting__label__count--up") %>%
  html_text() %>%
  head(20)

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

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


# Handle case where some elements might be missing, ensuring we have exactly 20 entries
reviewer_name <- c(reviewer_name, rep(NA, 20 - length(reviewer_name)))[1:20]
review_date <- c(review_date, rep(NA, 20 - length(review_date)))[1:20]
user_rating <- c(user_rating, rep(NA, 20 - length(user_rating)))[1:20]
review_title <- c(review_title, rep(NA, 20 - length(review_title)))[1:20]
helpful_reviews <- c(helpful_reviews, rep(NA, 20 - length(helpful_reviews)))[1:20]
not_helpful_reviews <- c(not_helpful_reviews, rep(NA, 20 - length(not_helpful_reviews)))[1:20]
text_reviews <- c(text_reviews, rep(NA, 20 - length(text_reviews)))[1:20]

# Create a temporary data frame for the current URL
dfTemp <- data.frame(
  reviewer_name = reviewer_name,
  review_date = review_date,
  user_rating = user_rating,
  review_title = review_title,
  helpful_reviews = helpful_reviews,
  not_helpful_reviews = not_helpful_reviews,
  text_reviews = text_reviews,
  stringsAsFactors = FALSE
)

# Append the temporary data frame to the list with a custom name
df[[df_names]] <- dfTemp

# View the data frame for "Planet Earth"
print(df$Planet_Earth)
```

```
##      reviewer_name  review_date user_rating
## 1     robert-kamer  Feb 8, 2007          10
## 2        Permalink Nov 19, 2008          10
## 3         jim-1409  Jan 4, 2009          10
## 4        Permalink Dec 15, 2006          10
## 5  ccthemovieman-1  Sep 1, 2007          10
## 6        Permalink Aug 27, 2006          10
## 7         cmcoveos Apr 30, 2006          10
## 8        Permalink Jun 29, 2015           9
```

```
## 9         Loordssm Jul 20, 2006          10
## 10       Permalink Jan 28, 2009          10
## 11        ultimorn  Jun 1, 2015           7
## 12       Permalink  Oct 8, 2020           3
## 13     bob the moo  Dec 4, 2007          10
## 14       Permalink Jan 15, 2007          10
## 15           alfeu Jul 30, 2008          10
## 16       Permalink Dec 25, 2017          10
## 17         Cabrone Sep 14, 2009          10
## 18       Permalink Sep 20, 2020           9
## 19        berndt65 Jul 27, 2014           9
## 20       Permalink May 31, 2020           9
##                                                                review_title
## 1                                                              User reviews
## 2                                                              11 out of 10
## 3                                          A masterpiece of a documentary
## 4                                                      In A Word: Amazing
## 5      The most amazing achievement in natural history TV has ever given
## 6                                                     Simply put, stunning
## 7                             An amazing trip around our beautiful planet.
## 8  A visually impressive and memorable look at the world that we live in
## 9                                    Is it real? I mean, actual footagge?
## 10                                                            Beautiful
## 11                                        Are you kidding me people?
## 12                                   It doesn't get any better than this.
## 13                                          Only 4 Eps can touch my soul!
## 14                           Should be called "BBC - Yeah, animals suck"
## 15                                        Brilliant Documentary Series
## 16                            Explanation to those low-rating reviews...
## 17                                                    Truly Astonishing
## 18                                                The Greatest Series Ever
## 19                               Words fail me to describe such greatness
## 20                                                Absolutely Mindblowing!
##    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  As the influence of man expands across the globe, fewer and fewer truly untouched wilderness exis
## 8
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16
## 17
## 18
## 19
## 20
```

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

# Initialize list to store data frames
df <- list()
df_names <- "Band_Of_Brothers"

# Read HTML session for the current URL
session <- read_html(BandOfBrothers_urls)

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

# Scrape review dates
review_date <- session %>%
  html_nodes(".ipc-inline-list__item.review-date") %>%
  html_text() %>%
  head(20)

# Scrape user ratings (corrected CSS selector)
user_rating <- session %>%
  html_nodes(".ipc-rating-star--rating") %>%
  html_text() %>%
  head(20)

# Scrape reviews' titles
review_title <- session %>%
  html_nodes(".ipc-title__text") %>%
  html_text() %>%
```

```
  head(20)

# Scrape helpful reviews
helpful_reviews <- session %>%
  html_nodes(".ipc-voting__label__count.ipc-voting__label__count--up") %>%
  html_text() %>%
  head(20)

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

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


# Handle case where some elements might be missing, ensuring we have exactly 20 entries
reviewer_name <- c(reviewer_name, rep(NA, 20 - length(reviewer_name)))[1:20]
review_date <- c(review_date, rep(NA, 20 - length(review_date)))[1:20]
user_rating <- c(user_rating, rep(NA, 20 - length(user_rating)))[1:20]
review_title <- c(review_title, rep(NA, 20 - length(review_title)))[1:20]
helpful_reviews <- c(helpful_reviews, rep(NA, 20 - length(helpful_reviews)))[1:20]
not_helpful_reviews <- c(not_helpful_reviews, rep(NA, 20 - length(not_helpful_reviews)))[1:20]
text_reviews <- c(text_reviews, rep(NA, 20 - length(text_reviews)))[1:20]

# Create a temporary data frame for the current URL
dfTemp <- data.frame(
  reviewer_name = reviewer_name,
  review_date = review_date,
  user_rating = user_rating,
  review_title = review_title,
  helpful_reviews = helpful_reviews,
  not_helpful_reviews = not_helpful_reviews,
  text_reviews = text_reviews,
  stringsAsFactors = FALSE
)

# Append the temporary data frame to the list with a custom name
df[[df_names]] <- dfTemp

# View the data frame for "band of brothers"
print(df$Band_Of_Brothers)
```

```
##         reviewer_name  review_date user_rating
## 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_title
## 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_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
## 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
# Define URL for Chernobyl
Chernobyl_urls <- "https://www.imdb.com/title/tt7366338/reviews/?ref_=tt_ov_urv"

# Initialize list to store data frames
df <- list()
df_names <- "Chernobyl"

# Read HTML session for the current URL
session <- read_html(Chernobyl_urls)

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

# Scrape review dates
review_date <- session %>%
  html_nodes(".ipc-inline-list__item.review-date") %>%
  html_text() %>%
  head(20)

# Scrape user ratings (corrected CSS selector)
user_rating <- session %>%
  html_nodes(".ipc-rating-star--rating") %>%
  html_text() %>%
  head(20)

# Scrape reviews' titles
```

```r
review_title <- session %>%
  html_nodes(".ipc-title__text") %>%
  html_text() %>%
  head(20)

# Scrape helpful reviews
helpful_reviews <- session %>%
  html_nodes(".ipc-voting__label__count.ipc-voting__label__count--up") %>%
  html_text() %>%
  head(20)

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

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

# Handle case where some elements might be missing, ensuring we have exactly 20 entries
reviewer_name <- c(reviewer_name, rep(NA, 20 - length(reviewer_name)))[1:20]
review_date <- c(review_date, rep(NA, 20 - length(review_date)))[1:20]
user_rating <- c(user_rating, rep(NA, 20 - length(user_rating)))[1:20]
review_title <- c(review_title, rep(NA, 20 - length(review_title)))[1:20]
helpful_reviews <- c(helpful_reviews, rep(NA, 20 - length(helpful_reviews)))[1:20]
not_helpful_reviews <- c(not_helpful_reviews, rep(NA, 20 - length(not_helpful_reviews)))[1:20]
text_reviews <- c(text_reviews, rep(NA, 20 - length(text_reviews)))[1:20]

# Create a temporary data frame for the current URL
dfTemp <- data.frame(
  reviewer_name = reviewer_name,
  review_date = review_date,
  user_rating = user_rating,
  review_title = review_title,
  helpful_reviews = helpful_reviews,
  not_helpful_reviews = not_helpful_reviews,
  text_reviews = text_reviews,
  stringsAsFactors = FALSE
)

# Append the temporary data frame to the list with a custom name
df[[df_names]] <- dfTemp

# View the data frame for "Chernobyl"
print(df$Chernobyl)
```

```
##        reviewer_name  review_date user_rating
## 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           7
## 12         Permalink Jan 27, 2024           1
## 13           thegldt Jun 29, 2019           8
## 14         Permalink May 20, 2019          10
## 15 alexander-phoenix May 30, 2019          10
## 16         Permalink  Jun 7, 2019          10
## 17       wmeduardowm  May 6, 2019           9
## 18         Permalink Sep 27, 2022           9
## 19    Leofwine_draca May 26, 2019           9
## 20         Permalink Jul 10, 2022          10
##                                            review_title helpful_reviews
## 1                                          User reviews            <NA>
## 2                                    They got it right             <NA>
## 3                                 Goosebumps and tears             <NA>
## 4                          I highly recommend this film!           <NA>
## 5                        No hero wakes up wanting to die           <NA>
## 6                                So far looks excellent            <NA>
## 7                                           Incredible             <NA>
## 8              Bleak, Unsettling, Haunting All Throughout           <NA>
## 9                                           Unbelievable           <NA>
## 10                                    HBO did it again!            <NA>
## 11                                           Exemplary             <NA>
## 12                                            Amazing!             <NA>
## 13           Unveiling Human Errors and Political Shadows          <NA>
## 14                                    How cost the lie?            <NA>
## 15                                Emotionally drained...           <NA>
## 16                                     Just watch it (!)           <NA>
## 17                Now you look like the minister of coal!          <NA>
## 18                                            Cracking.            <NA>
## 19                                           Must Watch!           <NA>
## 20 It is hard to overestimate the importance of this show.          <NA>
##    not_helpful_reviews
## 1                 <NA>
## 2                 <NA>
## 3                 <NA>
## 4                 <NA>
## 5                 <NA>
## 6                 <NA>
## 7                 <NA>
## 8                 <NA>
## 9                 <NA>
## 10                <NA>
## 11                <NA>
## 12                <NA>
## 13                <NA>
## 14                <NA>
## 15                <NA>
```

```
## 16                <NA>
## 17                <NA>
## 18                <NA>
## 19                <NA>
## 20                <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
#3.

# Convert the 'Year' column to numeric if it isn't already
top_tv_shows$Year <- as.numeric(top_tv_shows$Year)
```

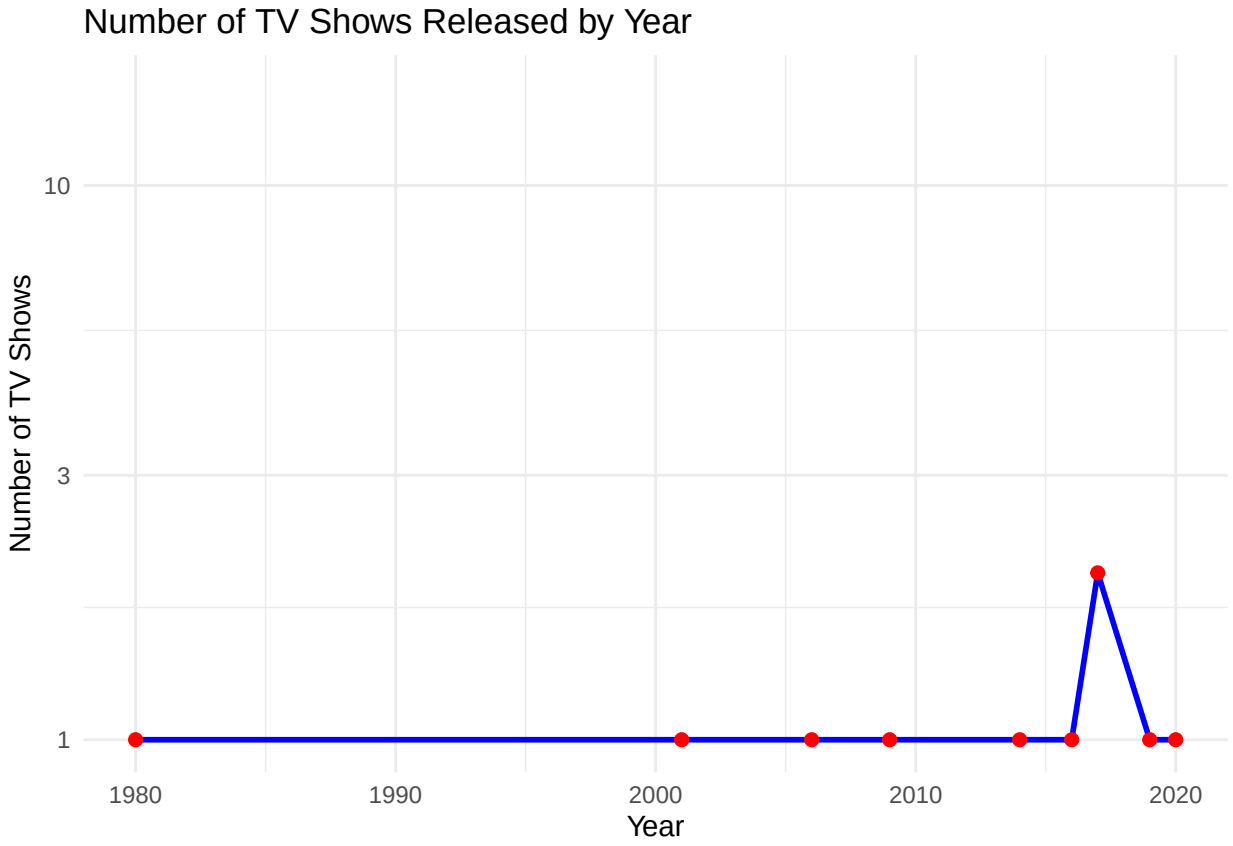
```
## Warning: NAs introduced by coercion
```

```r
# Group the data by Year and count the number of shows per year
shows_by_year <- top_tv_shows %>%
  group_by(Year) %>%
  summarise(Count = n())

# Plot the number of shows released by year
ggplot(shows_by_year, aes(x = Year, y = Count)) +
  geom_line(color = "blue", size = 1) +
  geom_point(color = "red", size = 2) +
  labs(title = "Number of TV Shows Released by Year",
       x = "Year",
       y = "Number of TV Shows") +
  scale_y_log10() +  # Use log scale for y-axis
  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()`).
```




```
# Find the year with the most TV shows released
most_shows_year <- shows_by_year %>%
  filter(Count == max(Count))

# Print the year with the most releases
print(most_shows_year)
```

```
## # A tibble: 1 x 2
##    Year Count
##   <dbl> <int>
## 1    NA    15
```

2. Extracting Amazon Product Reviews

```
#4. URLs
urls <- c('https://www.amazon.com/s?k=PC&crid=3OOHDEISP3NAZ&sprefix=pc%2Caps%2C610&ref=nb_sb_noss_1',
          'https://www.amazon.com/s?k=graphics+card&crid=4BJILUAEFT5I&sprefix=graphics%2Caps%2C372&ref=
          'https://www.amazon.com/s?k=keyboard&crid=3647UWDU9HOTF&sprefix=keyboa%2Caps%2C331&ref=nb_sb_
          'https://www.amazon.com/s?k=mouse&crid=QMZNC639VEYB&sprefix=mouse%2Caps%2C336&ref=nb_sb_ss_ts
          'https://www.amazon.com/s?k=motherboard&crid=1ZHTORVTCHK1A&sprefix=motherb%2Caps%2C346&ref=nb
```

```
#5
df <- list()

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

```
  session <- bow(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


  dfTemp <- data.frame(Product_Name = product_name[1:30], Description = product_description[1:30], Ratin

  df[[i]] <- dfTemp
}

print(df[[1]])
```

```
##
## 1                                                CyberPowerPC Gamer Master Gaming PC, AMD Ryzen 5 55
## 2        iBUYPOWER Y60 Black Gaming PC Computer Desktop Y60BA9N47TS03 (AMD Ryzen 9 7900X CPU, NVIDIA
## 3                                          Dell Optiplex 7050 SFF Desktop PC Intel i7-7700 4-Cores 3
## 4              Dell Optiplex Small Desktop Computer (SFF) PC | Quad Core Intel i5 (3.2GHz) | 16GB DDR
## 5                                       CyberPowerPC Gamer Xtreme VR Gaming PC, Intel Core i5-13400F 2
## 6   STGAubron Gaming Desktop PC Computer, Intel Core I7 3.4 GHz up to 3.9 GHz, Radeon RX 580 8G GDDR5
## 7                                CyberPowerPC Gamer Xtreme VR Gaming PC, Intel Core i9-14900KF 3.2GHz, G
## 8                          STGAubron Prebuilt Gaming PC Desktop, AMD Radeon RX 550 4G GDDR5, Intel Core
## 9                                       Dell OptiPlex Computer Desktop PC, Intel Core i5 3rd Gen 3.2 GH
## 10                           Skytech Archangel Gaming PC Desktop, Ryzen 5 5500 3.6 GHz (4.2GHz Turbo Bo
## 11                                ASUS ROG Strix G16CHR Gaming Desktop Intel 20-core i7-14700F (Beats i
## 12                                   Dell Optiplex 9020 Desktop Computer PC, Intel Quad-Core i5, 500GB
## 13    HP 21.5 inch All-in-One Desktop PC, 16GB RAM, 512GB PCIe SSD & 512GB External Storage, Intel Du
## 14                                        Alienware 2024 Newest Aurora R16 Gaming Desktop, 24-core i9-
## 15      Beelink S12 Pro Mini PC, Intel 12th Gen Alder Lake- N100(up to 3.4GHz), 16GB DDR4 RAM 500GB M
## 16 KAMRUI Mini PC Computers, AK1 PRO 12GB RAM 256GB SSD Mini Desktop Computer Intel Celeron N5105 CPU
## 17
## 18
## 19
## 20
## 21
## 22
## 23
## 24
## 25
## 26
```

```
## 27
## 28
## 29
## 30
##    Description Rating  Price
## 1         <NA>    4.3 649.99
## 2         <NA>    4.5 899.99
## 3         <NA>    4.7 223.91
## 4         <NA>    4.6 204.92
## 5         <NA>    4.4 899.99
## 6         <NA>    3.9 439.99
## 7         <NA>    3.6 839.99
## 8         <NA>    4.1 409.99
## 9         <NA>    3.7 168.00
## 10        <NA>    4.3 177.97
## 11        <NA>    4.0 779.99
## 12        <NA>    3.6 899.00
## 13        <NA>    4.7 133.00
## 14        <NA>    4.6 499.00
## 15        <NA>    3.8 549.00
## 16        <NA>    4.7 499.00
## 17        <NA>    5.0 209.00
## 18        <NA>    4.4 179.99
## 19        <NA>    4.4 259.99
## 20        <NA>     NA     NA
## 21        <NA>     NA     NA
## 22        <NA>     NA     NA
## 23        <NA>     NA     NA
## 24        <NA>     NA     NA
## 25        <NA>     NA     NA
## 26        <NA>     NA     NA
## 27        <NA>     NA     NA
## 28        <NA>     NA     NA
## 29        <NA>     NA     NA
## 30        <NA>     NA     NA
```

```
print(df[[2]])
```

```
##
## 1                                                                                MSI Gami
## 2                                                                GIGABYTE GeForce RTX 3060 Gaming OC 1
## 3  ASUS Dual GeForce RTX 4060 EVO OC Edition 8GB GDDR6 (PCIe 4.0, 8GB GDDR6, DLSS 3, HDMI 2.1a, Displ
## 4              MSI GeForce RTX 4070 Ti Super 16G Ventus 3X Black OC Graphics Card (NVIDIA RTX 4070
## 5                                                                ASUS ProArt GeForce RTX 4060
## 6
## 7             ASUS Dual NVIDIA GeForce RTX 3060 V2 OC Edition 12GB GDDR6 Gaming Graphics Card (PCIe
## 8                                                                     ASUS ProArt GeForce RT
## 9                                                                GIGABYTE Radeon RX 7800 XT Ga
## 10                                                               GIGABYTE GeForce RTX 3050 WIN
## 11                                                                 XFX Radeon RX 580 GTS XXX Ed
## 12                                                               GIGABYTE Radeon RX 7600 XT Ga
## 13                                          ASUS TUF Gaming NVIDIA GeForce RTX 4070 Ti Super OC Edi
## 14                                                                 GIGABYTE GeForce RTX 4060
## 15                                                       MSI Gaming GeForce GT 710 2GB GD
## 16                                                                 GIGABYTE GeForce RTX
```

```
## 17
## 18
## 19
## 20
## 21
## 22
## 23
## 24
## 25
## 26
## 27
## 28
## 29
## 30
##    Description Rating  Price
## 1         <NA>    4.6 284.99
## 2         <NA>    4.7 287.51
## 3         <NA>    4.7 319.99
## 4         <NA>    4.7 304.99
## 5         <NA>    4.7 839.99
## 6         <NA>    4.6 499.99
## 7         <NA>    4.4 529.99
## 8         <NA>    4.7 299.99
## 9         <NA>    4.5 349.99
## 10        <NA>    4.7 299.97
## 11        <NA>    4.6 149.97
## 12        <NA>    4.6 489.00
## 13        <NA>    4.4 169.99
## 14        <NA>    4.5 179.99
## 15        <NA>    4.5 130.52
## 16        <NA>    4.7 314.97
## 17        <NA>    4.7 329.99
## 18        <NA>    4.4 849.00
## 19        <NA>    4.8 299.99
## 20        <NA>     NA 319.99
## 21        <NA>     NA  46.99
## 22        <NA>     NA  69.99
## 23        <NA>     NA 319.99
## 24        <NA>     NA     NA
## 25        <NA>     NA     NA
## 26        <NA>     NA     NA
## 27        <NA>     NA     NA
## 28        <NA>     NA     NA
## 29        <NA>     NA     NA
## 30        <NA>     NA     NA
```

```
print(df[[3]])
```

```
##
## 1                                                                  Logitech K120 Wired Keyboard for Wi
## 2            Logitech MX Keys S Wireless Keyboard, Low Profile, Fluid Precise Quiet Typing, Programm
## 3                                                  Logitech MK270 Wireless Keyboard And Mouse Combo
## 4                                                                  Logitech G213 Prodigy Gaming Keyboa
## 5                                       SteelSeries Apex Pro TKL HyperMagnetic Gaming Keyboard - Ad
## 6                                         Logitech K270 Wireless Keyboard for Windows, 2.4 GHz Wirel
```

```
## 7          AULA F75 Pro Wireless Mechanical Keyboard,75% Gasket Hot Swappable Custom Keyboard,RGB Ba
## 8                                                                                                Rii
## 9
## 10                                                                    65% Gaming Keyboard, Wired Backlit Mini Keyboard, U
## 11                                                                   Logitech MK120 Wired Keyboard and Mouse Combo for Wi
## 12                                  SABLUTE Large Print Backlit Computer Keyboards, Wired Lighted USB Keyboards w
## 13                                               MageGee Portable 60% Mechanical Gaming Keyboard, MK-Box LED Bacl
## 14                                                Logitech MK345 Wireless Combo Full-Sized Keyboard with Palm Res
## 15                             Gaming Keyboard, 7-Color Rainbow LED Backlit, 104 Keys Quiet Light Up Keyboard, Wri
## 16 TECKNET RGB Gaming Keyboard, 105 Keys, All-Metal Panel, 15-Zone RGB Illumination, Silent Keyboard
## 17
## 18
## 19
## 20
## 21
## 22
## 23
## 24
## 25
## 26
## 27
## 28
## 29
## 30
##    Description Rating  Price
## 1         <NA>    4.6  12.99
## 2         <NA>    4.6 109.99
## 3         <NA>    4.5  27.99
## 4         <NA>    4.5  39.99
## 5         <NA>    4.3  69.99
## 6         <NA>    4.4 166.99
## 7         <NA>    4.6 189.99
## 8         <NA>    4.4  24.95
## 9         <NA>    4.5  29.99
## 10        <NA>    4.2  82.89
## 11        <NA>    4.5   9.99
## 12        <NA>    4.3  14.99
## 13        <NA>    4.4  20.89
## 14        <NA>    4.5  13.97
## 15        <NA>    4.5  15.97
## 16        <NA>    4.4  14.49
## 17        <NA>     NA  19.99
## 18        <NA>     NA  31.99
## 19        <NA>     NA  29.99
## 20        <NA>     NA  33.89
## 21        <NA>     NA  39.99
## 22        <NA>     NA  22.99
## 23        <NA>     NA  29.99
## 24        <NA>     NA  29.99
## 25        <NA>     NA  36.99
## 26        <NA>     NA     NA
## 27        <NA>     NA     NA
## 28        <NA>     NA     NA
## 29        <NA>     NA     NA
```

```
## 30        <NA>     NA     NA
```

```
print(df[[4]])
```

```
##
## 1                                                    Logitech M185 Wireless Mouse, 2.4GHz with USB M
## 2
## 3                     TECKNET Wireless Mouse, 2.4G Ergonomic Optical Mouse, Computer Mouse for Laptop
## 4                           Logitech G PRO X SUPERLIGHT Wireless Gaming Mouse, Ultra-Lightweight, HI
## 5                                    Logitech G305 LIGHTSPEED Wireless Gaming Mouse, Hero 12K Sens
## 6         Logitech MX Master 3S - Wireless Performance Mouse, Ergo, 8K DPI, Track on Glass, Quiet Cli
## 7                                                                                                 App
## 8                                       Logitech G502 HERO High Performance Wired Gaming Mouse, HI
## 9                                                                                  Amazon Basi
## 10
## 11                                                                                    Amazon Ba
## 12                                                  Logitech MX Vertical Wireless Mouse - Ergonomic De
## 13 Logitech G PRO X SUPERLIGHT 2 LIGHTSPEED Wireless Gaming Mouse, 8K Polling, Lightweight, LIGHTFORC
## 14                                                                        VssoPlor Wireless Mouse, 1
## 15                                                  Razer DeathAdder Essential Gaming Mouse: 6400
## 16
## 17
## 18
## 19
## 20
## 21
## 22
## 23
## 24
## 25
## 26
## 27
## 28
## 29
## 30
##    Description Rating  Price
## 1         <NA>    4.6  13.98
## 2         <NA>    4.4  14.99
## 3         <NA>    4.6  27.99
## 4         <NA>    4.5   9.99
## 5         <NA>    4.6  19.99
## 6         <NA>    4.5  81.97
## 7         <NA>    4.6 159.99
## 8         <NA>    4.6  31.99
## 9         <NA>    4.6  49.99
## 10        <NA>    4.6  99.99
## 11        <NA>    4.6  67.98
## 12        <NA>    4.6  79.00
## 13        <NA>    4.5  39.99
## 14        <NA>    4.3  79.99
## 15        <NA>    4.4   7.19
## 16        <NA>    4.5   9.72
## 17        <NA>    4.6   8.09
## 18        <NA>    4.4  12.99
## 19        <NA>    4.6  76.10
```

```
## 20        <NA>     NA  99.99
## 21        <NA>     NA 128.99
## 22        <NA>     NA 159.00
## 23        <NA>     NA   9.98
## 24        <NA>     NA  12.99
## 25        <NA>     NA  20.98
## 26        <NA>     NA  29.99
## 27        <NA>     NA   7.99
## 28        <NA>     NA   9.99
## 29        <NA>     NA     NA
## 30        <NA>     NA     NA
```

```
print(df[[5]])
```

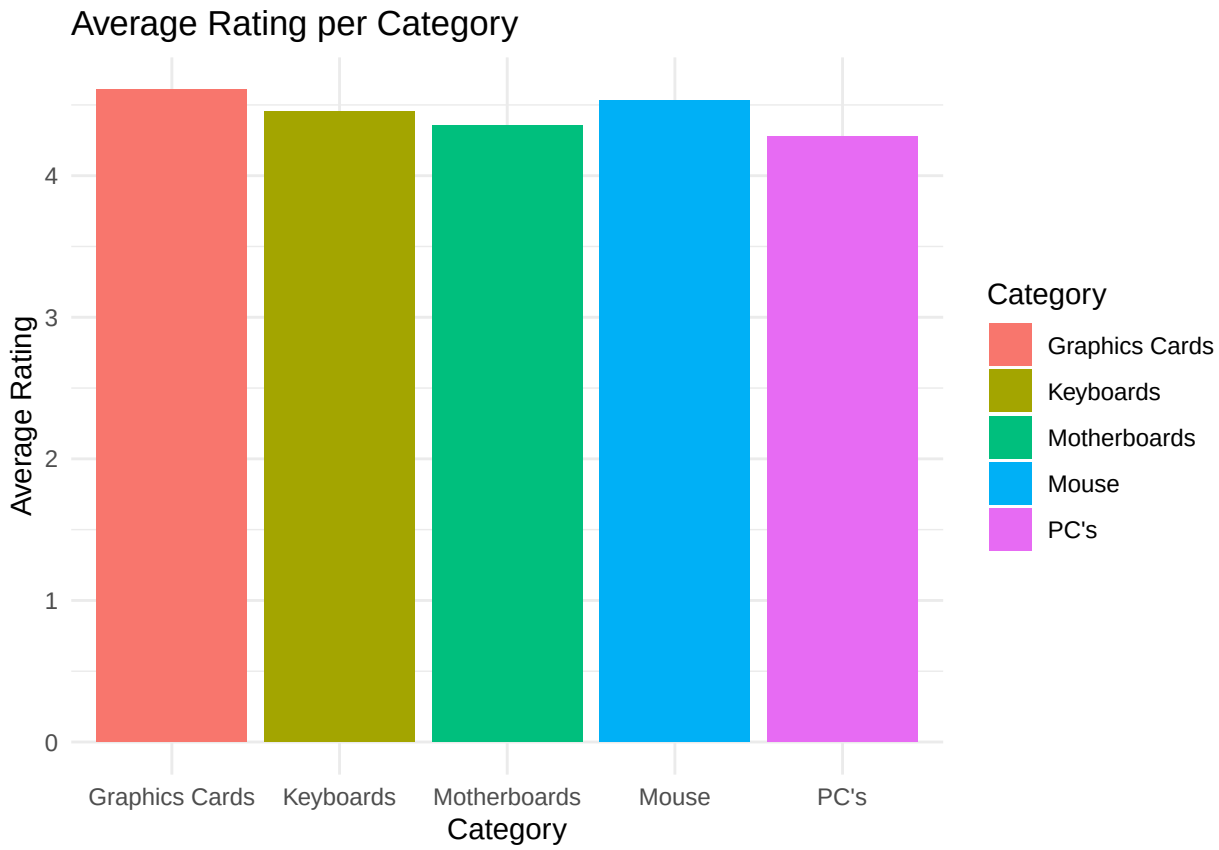
```
##
## 1                          Asus ROG Strix B550-F Gaming WiFi II AMD AM4 (3rd Gen Ryzen) ATX Motherboa
## 2                                                                                MSI B550-A PRO Pro
## 3  ASUS TUF Gaming Z790-Plus WiFi LGA 1700(Intel 14th,12th &13th Gen) ATX Gaming Motherboard(PCIe 5.0
## 4           ASUS Prime B550-PLUS AMD AM4 Zen 3 Ryzen 5000 & 3rd Gen Ryzen ATX Motherboard (PCIe 4.0,
## 5      MSI MAG X670E Tomahawk WiFi Gaming Motherboard (AMD Ryzen 9000/8000/7000 Series Processors, AM
## 6                                    ASUS ROG Strix B650-A Gaming WiFi 6E AM5 (LGA1718) Ryzen 7000 Mo
## 7                                                          GIGABYTE B650 Eagle AX AM5
## 8     ASUS ROG Strix B760-F Gaming WiFi Intel® B760(13th and 12th Gen) LGA 1700 ATX Motherboard,16 +
## 9              MSI B760 Gaming Plus WiFi Gaming Motherboard (Supports 12th/13th/14th Gen Intel Proces
## 10                                                        MSI B550 Gaming GEN3 Gaming Mother
## 11                          NZXT N7 B650E - AMD B650 Chipset - Supports AMD Ryzen 9000 8000 & 7000 Se
## 12  ASUS ROG Strix Z790-E Gaming WiFi II LGA 1700(Intel 14th & 13th & 12th Gen)ATX gaming motherboard
## 13           MSI PRO B760-P WiFi DDR4 ProSeries Motherboard (Supports 12th/13th/14th Gen Intel Proces
## 14
## 15                                                                                ASRock B650I Li
## 16
## 17
## 18
## 19
## 20
## 21
## 22
## 23
## 24
## 25
## 26
## 27
## 28
## 29
## 30
##    Description Rating  Price
## 1         <NA>    4.5 159.99
## 2         <NA>    4.2 189.99
## 3         <NA>    4.4 109.95
## 4         <NA>    4.6 119.99
## 5         <NA>    4.6 223.66
## 6         <NA>    4.4 249.99
## 7         <NA>    4.5 123.51
## 8         <NA>    4.2 239.99
## 9         <NA>    4.2 299.99
```

```
## 10        <NA>    4.3 239.99
## 11        <NA>    4.3 159.99
## 12        <NA>    4.4 179.99
## 13        <NA>    4.5 199.99
## 14        <NA>    4.4 269.99
## 15        <NA>    4.3 159.99
## 16        <NA>    4.6 169.99
## 17        <NA>    4.1 105.00
## 18        <NA>    4.1 119.99
## 19        <NA>    4.2 249.99
## 20        <NA>     NA 319.99
## 21        <NA>     NA 379.99
## 22        <NA>     NA 459.99
## 23        <NA>     NA 149.99
## 24        <NA>     NA 159.99
## 25        <NA>     NA 199.99
## 26        <NA>     NA 219.99
## 27        <NA>     NA 179.99
## 28        <NA>     NA 199.99
## 29        <NA>     NA 219.99
## 30        <NA>     NA 299.99
```

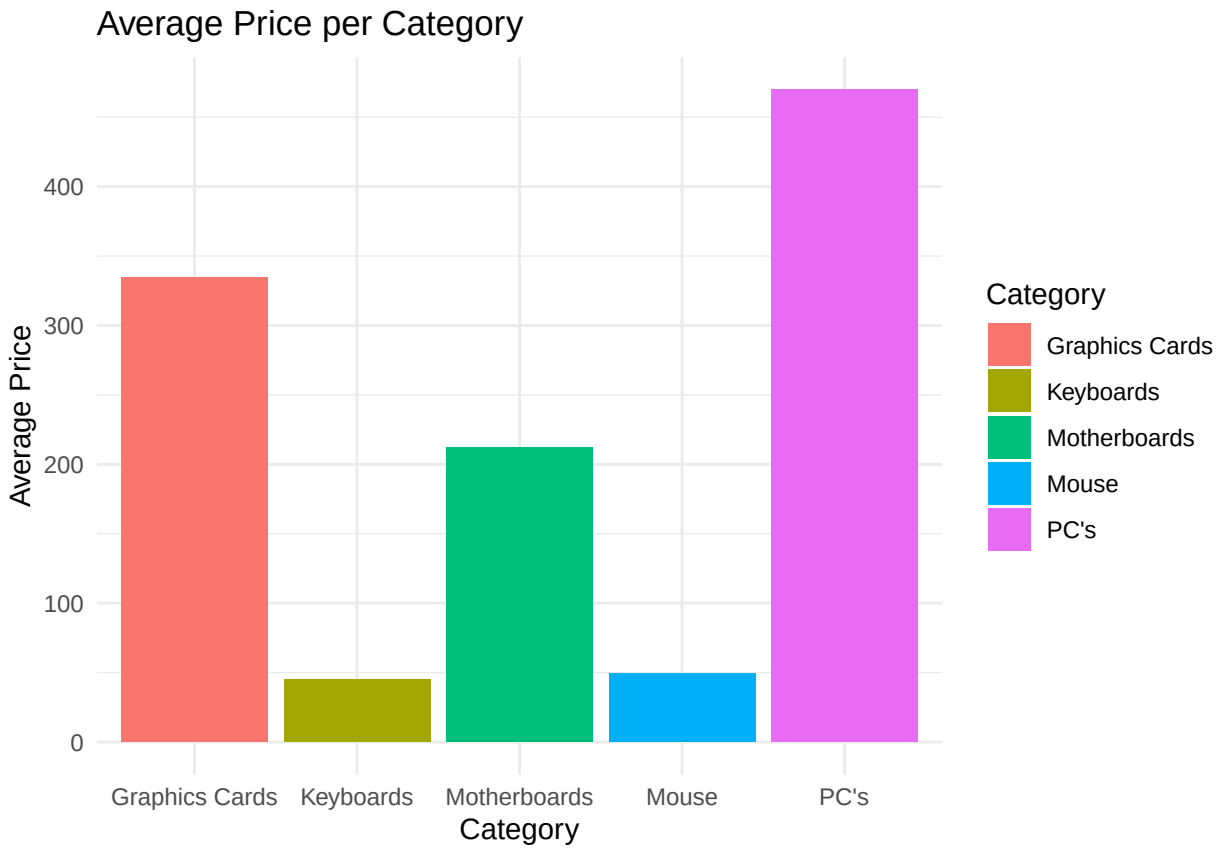
```
#6.

#The code extracts data from Amazon product listing pages based on different search queries, such as "P
```

```
#7

#This data can be used to compare product popularity, analyze price trends, examine the relationship be
```

```
#8
combined_df <- do.call(rbind, df)
combined_df$Category <- rep(c("PC's", "Graphics Cards", "Keyboards", "Mouse", "Motherboards"), each = 3

avg_rating <- combined_df %>%
  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()
```

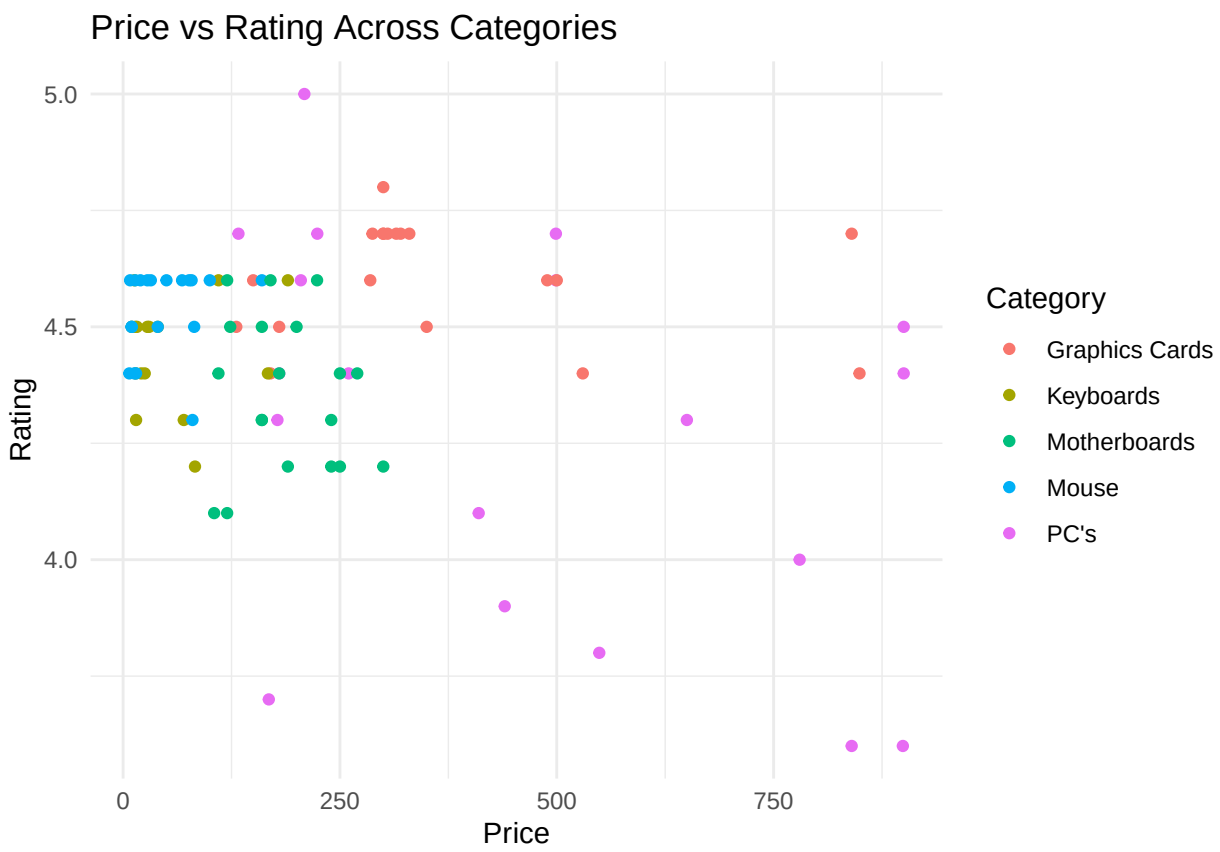



```
avg_price <- combined_df %>%
  group_by(Category) %>%
  summarize(Average_Price = mean(Price, na.rm = TRUE))

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()
```



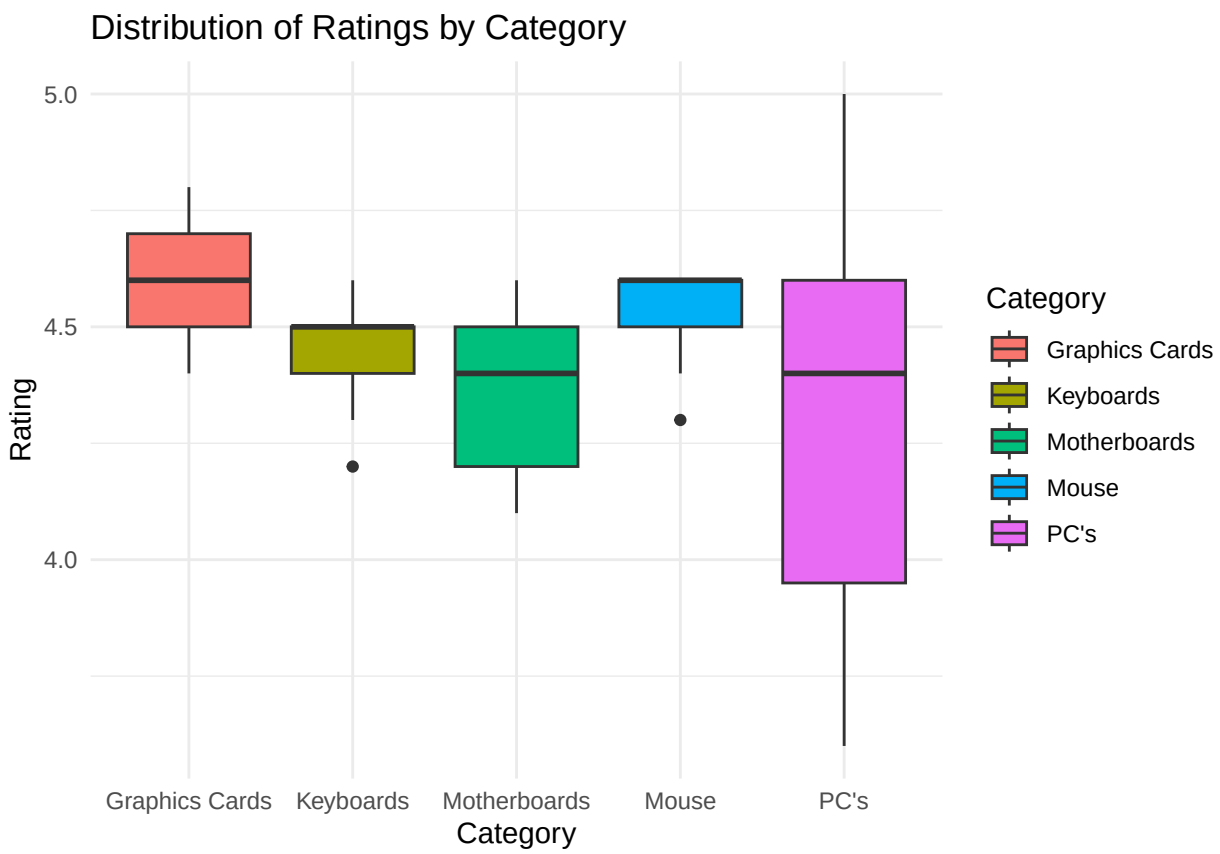

```
ggplot(combined_df, aes(x = Price, y = Rating, color = Category)) +
  geom_point() +
  labs(title = "Price vs Rating Across Categories", x = "Price", y = "Rating") +
  theme_minimal()
```

```
## Warning: Removed 58 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



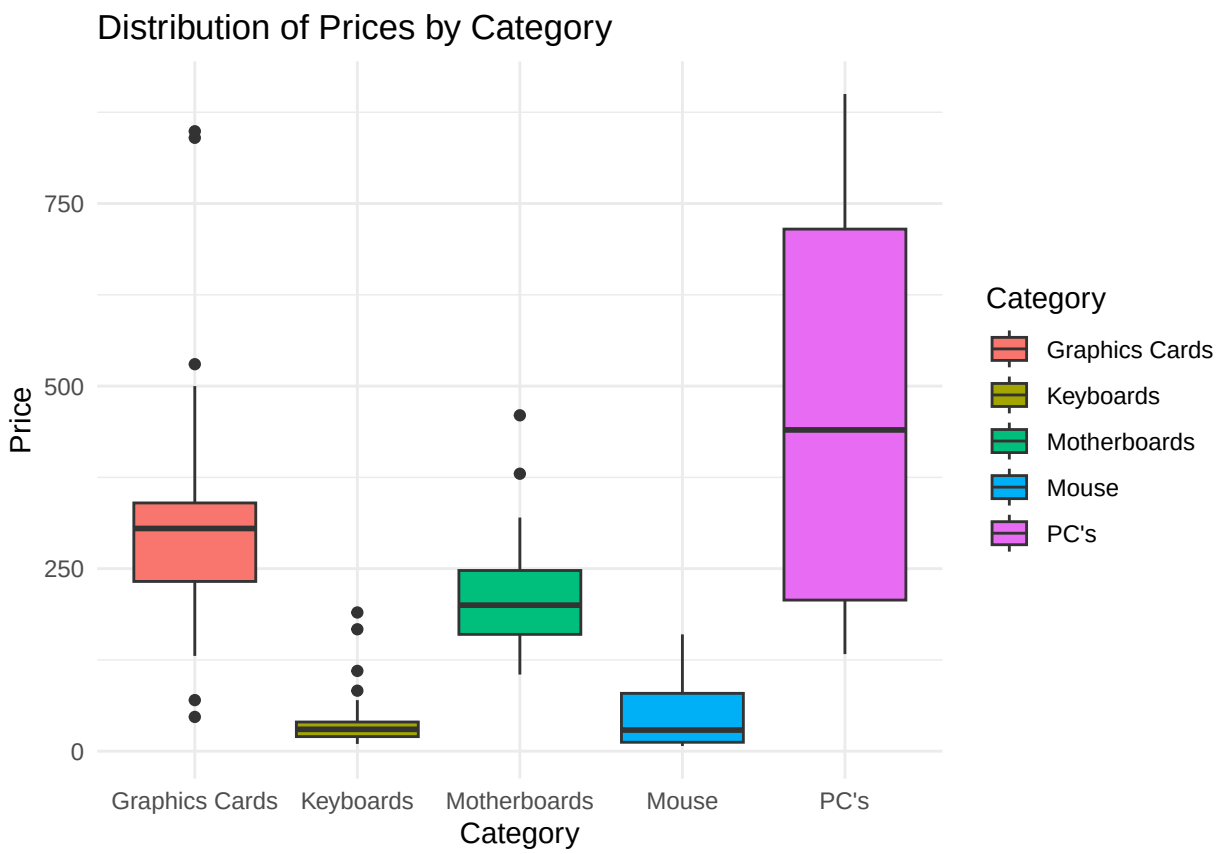

```
#9
ggplot(combined_df, aes(x = Category, y = Rating, fill = Category)) +
  geom_boxplot() +
  labs(title = "Distribution of Ratings by Category", x = "Category", y = "Rating") +
  theme_minimal()
```

```
## Warning: Removed 58 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```




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

```
## Warning: Removed 25 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```




```
#10
ranked_data <- lapply(df, 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_data)) {
  ranked_data[[i]]$Category <- categories[i]
}

ranked_combined_df <- do.call(rbind, ranked_data)
ranked_combined_df <- ranked_combined_df %>%
  arrange(Category, Rank) %>%
  group_by(Category) %>%
  slice(1:5)

print(ranked_combined_df)
```

```
## # A tibble: 25 x 6
## # Groups:   Category [5]
##     Rank Product_Name                          Description Rating  Price Category
##    <int> <chr>                                 <chr>        <dbl>  <dbl> <chr>
##  1     1  <NA>                                 <NA>           4.8 300.   Graphic~
##  2     2 "GIGABYTE GeForce RTX 3060 Gaming O~ <NA>           4.7 288.   Graphic~
```

```
##  3     3 "GIGABYTE GeForce RTX 3050 WINDFORC~ <NA>          4.7 300.   Graphic~
##  4     4 "ASUS ProArt GeForce RTX 4080 Supe~ <NA>          4.7 300.   Graphic~
##  5     5 "MSI GeForce RTX 4070 Ti Super 16G ~ <NA>          4.7 305.   Graphic~
##  6     1 "Logitech K120 Wired Keyboard for W~ <NA>          4.6  13.0  Keyboar~
##  7     2 "Logitech MX Keys S Wireless Keyboa~ <NA>          4.6 110.   Keyboar~
##  8     3 "AULA F75 Pro Wireless Mechanical K~ <NA>          4.6 190.   Keyboar~
##  9     4 "Logitech MK120 Wired Keyboard and ~ <NA>          4.5   9.99 Keyboar~
## 10     5 "Logitech MK345 Wireless Combo Full~ <NA>          4.5  14.0  Keyboar~
## # i 15 more rows
```