```
options(repos = c(CRAN = "https://cloud.r-project.org"))
```

```
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)
library(magrittr)

install.packages("ggplot2")
```

```
## Installing package into 'C:/Users/laure/AppData/Local/R/win-library/4.4'
## (as 'lib' is unspecified)
```

```
## package 'ggplot2' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\laure\AppData\Local\Temp\RtmpIjFQyj\downloaded_packages
```

```
library(ggplot2)
```

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

```
polite::use_manners(save_as = "polite_scrape_tvshows.R")
```

```
## ✓ Setting active project to "C:/Users/laure/Documents/Karl's Stuff/ISATU/2nd
##   Year/Data Science/R Analytics".
```

```
url <- "https://www.imdb.com/chart/toptv/?ref_=nv_tvv_250"
session <- bow(url, user_agent = "Educational")
session
```

```
## <polite session> https://www.imdb.com/chart/toptv/?ref_=nv_tvv_250
##     User-agent: Educational
##     robots.txt: 35 rules are defined for 3 bots
##    Crawl delay: 5 sec
##   The path is scrapable for this user-agent
```

#Show tv titles

```
#Title
title_list <- scrape(session) %>% html_nodes("h3.ipc-title__text") %>% html_text(trim = TRUE)
#filter
```

```
title_list <- title_list[!grepl("Recently viewed", title_list)]
title_list
```

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

#List of the Top 50 TV Shows

```
class(title_list)
```

```
## [1] "character"
```

```
listtitle <- as.data.frame(title_list[2:51])
listtitle
```

```
##                       title_list[2:51]
## 1                      1. Breaking Bad
## 2                   2. Planet Earth II
## 3                      3. Planet Earth
## 4                  4. Band of Brothers
## 5                         5. Chernobyl
## 6                          6. The Wire
## 7        7. Avatar: The Last Airbender
## 8                    8. Blue Planet II
## 9                      9. The Sopranos
## 10     10. Cosmos: A Spacetime Odyssey
## 11                          11. Cosmos
## 12                      12. Our Planet
## 13                 13. Game of Thrones
## 14                           14. Bluey
## 15                15. The World at War
## 16 16. Fullmetal Alchemist Brotherhood
## 17                  17. Rick and Morty
## 18                            18. Life
## 19                  19. The Last Dance
## 20               20. The Twilight Zone
## 21                 21. The Vietnam War
## 22                        22. Sherlock
## 23                 23. Attack on Titan
## 24     24. Batman: The Animated Series
## 25                          25. Arcane
## 26                                <NA>
## 27                                <NA>
## 28                                <NA>
## 29                                <NA>
## 30                                <NA>
## 31                                <NA>
## 32                                <NA>
## 33                                <NA>
## 34                                <NA>
## 35                                <NA>
## 36                                <NA>
## 37                                <NA>
## 38                                <NA>
## 39                                <NA>
## 40                                <NA>
## 41                                <NA>
## 42                                <NA>
## 43                                <NA>
## 44                                <NA>
## 45                                <NA>
## 46                                <NA>
## 47                                <NA>
## 48                                <NA>
## 49                                <NA>
## 50                                <NA>
```

#Rank number and the TV Show title.

```
colnames(listtitle) <- "ranks"
split_df <- strsplit(as.character(listtitle$ranks),".",fixed = TRUE)
split_df <- data.frame(do.call(rbind,split_df))
split_df <- split_df[-c(3:4)]
colnames(split_df) <- c("Ranks","Title")
str(split_df)
```

```
## 'data.frame':    50 obs. of  2 variables:
##  $ Ranks: chr  "1" "2" "3" "4" ...
##  $ Title: chr  " Breaking Bad" " Planet Earth II" " Planet Earth" " Band of Brothers" ...
```

#The Rank and the Title of the TV Shows

```
class(split_df)
```

```
## [1] "data.frame"
```

```
split_df
```

```
##    Ranks                           Title
## 1      1                    Breaking Bad
## 2      2                 Planet Earth II
## 3      3                    Planet Earth
## 4      4                Band of Brothers
## 5      5                       Chernobyl
## 6      6                        The Wire
## 7      7      Avatar: The Last Airbender
## 8      8                  Blue Planet II
## 9      9                    The Sopranos
## 10    10     Cosmos: A Spacetime Odyssey
## 11    11                          Cosmos
## 12    12                      Our Planet
## 13    13                 Game of Thrones
## 14    14                           Bluey
## 15    15                The World at War
## 16    16 Fullmetal Alchemist Brotherhood
## 17    17                  Rick and Morty
## 18    18                            Life
## 19    19                  The Last Dance
## 20    20               The Twilight Zone
## 21    21                 The Vietnam War
## 22    22                        Sherlock
## 23    23                 Attack on Titan
## 24    24     Batman: The Animated Series
## 25    25                          Arcane
## 26  <NA>                            <NA>
## 27  <NA>                            <NA>
## 28  <NA>                            <NA>
## 29  <NA>                            <NA>
## 30  <NA>                            <NA>
## 31  <NA>                            <NA>
## 32  <NA>                            <NA>
## 33  <NA>                            <NA>
## 34  <NA>                            <NA>
## 35  <NA>                            <NA>
## 36  <NA>                            <NA>
## 37  <NA>                            <NA>
## 38  <NA>                            <NA>
## 39  <NA>                            <NA>
## 40  <NA>                            <NA>
## 41  <NA>                            <NA>
## 42  <NA>                            <NA>
## 43  <NA>                            <NA>
## 44  <NA>                            <NA>
## 45  <NA>                            <NA>
## 46  <NA>                            <NA>
## 47  <NA>                            <NA>
## 48  <NA>                            <NA>
## 49  <NA>                            <NA>
## 50  <NA>                            <NA>
```

#Top 50 TV Show Rating

```
rating <- scrape(session) %>% html_nodes("span.ipc-rating-star--rating") %>% html_text
tv_rating <- as.data.frame(rating [1:50])
tv_rating
```

```
##    rating[1:50]
## 1           9.5
## 2           9.5
## 3           9.4
## 4           9.4
## 5           9.3
## 6           9.3
## 7           9.3
## 8           9.3
## 9           9.2
## 10          9.2
## 11          9.3
## 12          9.2
## 13          9.2
## 14          9.3
## 15          9.2
## 16          9.1
## 17          9.1
## 18          9.1
## 19          9.0
## 20          9.0
## 21          9.1
## 22          9.1
## 23          9.1
## 24          9.0
## 25          9.0
## 26         <NA>
## 27         <NA>
## 28         <NA>
## 29         <NA>
## 30         <NA>
## 31         <NA>
## 32         <NA>
## 33         <NA>
## 34         <NA>
## 35         <NA>
## 36         <NA>
## 37         <NA>
## 38         <NA>
## 39         <NA>
## 40         <NA>
## 41         <NA>
## 42         <NA>
## 43         <NA>
## 44         <NA>
## 45         <NA>
## 46         <NA>
## 47         <NA>
## 48         <NA>
## 49         <NA>
## 50         <NA>
```

#Number of People who Voted

```
tv_votes <- scrape(session) %>% html_nodes("span.ipc-rating-star--voteCount") %>% html_text
total_tv_votes <- as.data.frame(tv_votes[1:50])
total_tv_votes
```

```
##    tv_votes[1:50]
## 1          (2.2M)
## 2          (162K)
## 3          (224K)
## 4          (546K)
## 5          (908K)
## 6          (391K)
## 7          (390K)
## 8           (49K)
## 9          (499K)
## 10         (131K)
## 11          (46K)
## 12          (54K)
## 13         (2.4M)
## 14          (33K)
## 15          (31K)
## 16         (209K)
## 17         (627K)
## 18          (44K)
## 19         (160K)
## 20          (97K)
## 21          (29K)
## 22           (1M)
## 23         (562K)
## 24         (122K)
## 25         (308K)
## 26           <NA>
## 27           <NA>
## 28           <NA>
## 29           <NA>
## 30           <NA>
## 31           <NA>
## 32           <NA>
## 33           <NA>
## 34           <NA>
## 35           <NA>
## 36           <NA>
## 37           <NA>
## 38           <NA>
## 39           <NA>
## 40           <NA>
## 41           <NA>
## 42           <NA>
## 43           <NA>
## 44           <NA>
## 45           <NA>
## 46           <NA>
## 47           <NA>
## 48           <NA>
## 49           <NA>
## 50           <NA>
```

#Number of Episodes of each TV Shows

```
episodes <- scrape(session) %>% html_nodes("span.sc-5bc66c50-6.OOdsw") %>% html_text
cl_episodes <- gsub("\\D", "", episodes)
cleaned_episodes <- str_extract(episodes, "\\d+(?=\\s*eps)")
cleaned_episodes <- as.numeric(cleaned_episodes)
cleaned_episodes <- cleaned_episodes[!is.na(cleaned_episodes)]
cleaned_episodes <- as.data.frame(cleaned_episodes[1:25])
cleaned_episodes
```

```
##    cleaned_episodes[1:25]
## 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
## 21                     NA
## 22                     NA
## 23                     NA
## 24                     NA
## 25                     NA
```

#Year of TV Shows released

```
tv_years <- scrape(session) %>% html_nodes("span.sc-5bc66c50-6.OOdsw") %>% html_text
clyear <- gsub(".*?(\\d{4}(-\\d{4})?).*", "\\1", tv_years)
yeartv <- str_extract(tv_years, "\\b\\d{4}(-\\d{4})?\\b")
yeartv <- as.numeric(yeartv)
yeartv <- yeartv[!is.na(yeartv)]
tv_year_of_air <- as.data.frame(yeartv[1:25])
tv_year_of_air
```

```
##    yeartv[1:25]
## 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
## 21           NA
## 22           NA
## 23           NA
## 24           NA
## 25           NA
```

#Data frame of TV Shows

```
final_data <- cbind(split_df,tv_rating,cleaned_episodes,tv_year_of_air)
colnames(final_data) <- c("Ranks", "TV Rating", "Number of Votes", "Number of Episodes", "Year R
eleased")
final_data
```

```
##    Ranks                        TV Rating Number of Votes Number of Episodes
## 1      1                     Breaking Bad             9.5                 NA
## 2      2                  Planet Earth II             9.5                 NA
## 3      3                     Planet Earth             9.4                 NA
## 4      4                 Band of Brothers             9.4                 NA
## 5      5                        Chernobyl             9.3                 NA
## 6      6                         The Wire             9.3                 NA
## 7      7       Avatar: The Last Airbender             9.3                 NA
## 8      8                   Blue Planet II             9.3                 NA
## 9      9                     The Sopranos             9.2                 NA
## 10    10      Cosmos: A Spacetime Odyssey             9.2                 NA
## 11    11                           Cosmos             9.3                 NA
## 12    12                       Our Planet             9.2                 NA
## 13    13                  Game of Thrones             9.2                 NA
## 14    14                            Bluey             9.3                 NA
## 15    15                 The World at War             9.2                 NA
## 16    16  Fullmetal Alchemist Brotherhood             9.1                 NA
## 17    17                   Rick and Morty             9.1                 NA
## 18    18                             Life             9.1                 NA
## 19    19                   The Last Dance             9.0                 NA
## 20    20                The Twilight Zone             9.0                 NA
## 21    21                  The Vietnam War             9.1                 NA
## 22    22                         Sherlock             9.1                 NA
## 23    23                  Attack on Titan             9.1                 NA
## 24    24       Batman: The Animated Series             9.0                 NA
## 25    25                           Arcane             9.0                 NA
## 26  <NA>                             <NA>            <NA>                 NA
## 27  <NA>                             <NA>            <NA>                 NA
## 28  <NA>                             <NA>            <NA>                 NA
## 29  <NA>                             <NA>            <NA>                 NA
## 30  <NA>                             <NA>            <NA>                 NA
## 31  <NA>                             <NA>            <NA>                 NA
## 32  <NA>                             <NA>            <NA>                 NA
## 33  <NA>                             <NA>            <NA>                 NA
## 34  <NA>                             <NA>            <NA>                 NA
## 35  <NA>                             <NA>            <NA>                 NA
## 36  <NA>                             <NA>            <NA>                 NA
## 37  <NA>                             <NA>            <NA>                 NA
## 38  <NA>                             <NA>            <NA>                 NA
## 39  <NA>                             <NA>            <NA>                 NA
## 40  <NA>                             <NA>            <NA>                 NA
## 41  <NA>                             <NA>            <NA>                 NA
## 42  <NA>                             <NA>            <NA>                 NA
## 43  <NA>                             <NA>            <NA>                 NA
## 44  <NA>                             <NA>            <NA>                 NA
## 45  <NA>                             <NA>            <NA>                 NA
## 46  <NA>                             <NA>            <NA>                 NA
## 47  <NA>                             <NA>            <NA>                 NA
## 48  <NA>                             <NA>            <NA>                 NA
## 49  <NA>                             <NA>            <NA>                 NA
## 50  <NA>                             <NA>            <NA>                 NA
##    Year Released
## 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
## 21            NA
## 22            NA
## 23            NA
## 24            NA
## 25            NA
## 26            NA
## 27            NA
## 28            NA
## 29            NA
## 30            NA
## 31            NA
## 32            NA
## 33            NA
## 34            NA
## 35            NA
## 36            NA
## 37            NA
## 38            NA
## 39            NA
## 40            NA
## 41            NA
## 42            NA
## 43            NA
## 44            NA
## 45            NA
## 46            NA
## 47            NA
## 48            NA
## 49            NA
## 50            NA
```

```
#4.)
urls <- c('https://www.amazon.com/s?i=specialty-aps&bbn=16225009011&rh=n%3A%2116225009011%2Cn%3A
281407&ref=nav_em__nav_desktop_sa_intl_accessories_and_supplies_0_2_5_2',
          'https://www.amazon.com/s?i=specialty-aps&bbn=16225009011&rh=n%3A%2116225009011%2Cn%3A
502394&ref=nav_em__nav_desktop_sa_intl_camera_and_photo_0_2_5_3',
          'https://www.amazon.com/s?i=specialty-aps&bbn=16225009011&rh=n%3A%2116225009011%2Cn%3A
3248684011&ref=nav_em__nav_desktop_sa_intl_car_and_vehicle_electronics_0_2_5_4',
          'https://www.amazon.com/s?i=specialty-aps&bbn=16225009011&rh=n%3A%2116225009011%2Cn%3A
2811119011&ref=nav_em__nav_desktop_sa_intl_cell_phones_and_accessories_0_2_5_5',
          'https://www.amazon.com/s?i=specialty-aps&bbn=16225009011&rh=n%3A%2116225009011%2Cn%3A
541966&ref=nav_em__nav_desktop_sa_intl_computers_and_accessories_0_2_5_6')
```

```
#5
df <- list()

for (i in seq_along(urls)) {

down <- bow(urls[i], user_agent = "Educational")

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

product_price <- scrape(down) %>%
    html_nodes('span.a-price') %>%
    html_text() %>%
    head(30)

price <- as.numeric(str_extract(product_price, "\\d+\\.\\d+"))

product_description <- scrape(down) %>%
    html_nodes('.a-spacing-mini:nth-child(1) .a-list-item') %>%
    html_text() %>%
    head(30)

product_rating <- scrape(down) %>%
    html_nodes('span.a-icon-alt') %>%
    html_text() %>%
    head(30)

ratings <- as.numeric(str_extract(product_rating, "\\d+\\.\\d"))

product_review <- scrape(down) %>%
    html_nodes('div.review-text-content') %>%
    html_text() %>%
    head(30)

Temporary_df <- data.frame(Product_Name = product_name[1:30],
                       Description = product_description[1:30],
                       Rating = ratings[1:30],
                       Price = price[1:30],
                       stringsAsFactors = FALSE)

#colnames(Temporary_df) <- c("Product Name")
  df[[i]] <- Temporary_df
}

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

```
##
Product_Name
## 1  Datacolor Spyder Print - Advanced Data Analysis and Calibration Tool for Optimal Print Res
ults, Perfect for Photographers, Graphic Designers, and Printing Professionals
## 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>
## 21
<NA>
## 22
<NA>
## 23
<NA>
## 24
<NA>
## 25
<NA>
```

```
## 26
<NA>
## 27
<NA>
## 28
<NA>
## 29
<NA>
## 30
<NA>
##    Description Rating  Price
## 1         <NA>    2.9 332.99
## 2         <NA>     NA 349.00
## 3         <NA>     NA     NA
## 4         <NA>     NA     NA
## 5         <NA>     NA     NA
## 6         <NA>     NA     NA
## 7         <NA>     NA     NA
## 8         <NA>     NA     NA
## 9         <NA>     NA     NA
## 10        <NA>     NA     NA
## 11        <NA>     NA     NA
## 12        <NA>     NA     NA
## 13        <NA>     NA     NA
## 14        <NA>     NA     NA
## 15        <NA>     NA     NA
## 16        <NA>     NA     NA
## 17        <NA>     NA     NA
## 18        <NA>     NA     NA
## 19        <NA>     NA     NA
## 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]])
```

```
##
Product_Name
## 1  Datacolor Spyder Print - Advanced Data Analysis and Calibration Tool for Optimal Print Res
ults, Perfect for Photographers, Graphic Designers, and Printing Professionals
## 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>
## 21
<NA>
## 22
<NA>
## 23
<NA>
## 24
<NA>
## 25
<NA>
```

```
## 26
<NA>
## 27
<NA>
## 28
<NA>
## 29
<NA>
## 30
<NA>
##    Description Rating  Price
## 1         <NA>    2.9 332.99
## 2         <NA>     NA 349.00
## 3         <NA>     NA     NA
## 4         <NA>     NA     NA
## 5         <NA>     NA     NA
## 6         <NA>     NA     NA
## 7         <NA>     NA     NA
## 8         <NA>     NA     NA
## 9         <NA>     NA     NA
## 10        <NA>     NA     NA
## 11        <NA>     NA     NA
## 12        <NA>     NA     NA
## 13        <NA>     NA     NA
## 14        <NA>     NA     NA
## 15        <NA>     NA     NA
## 16        <NA>     NA     NA
## 17        <NA>     NA     NA
## 18        <NA>     NA     NA
## 19        <NA>     NA     NA
## 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[[3]])
```

```
##    Product_Name Description Rating Price
## 1          <NA>        <NA>     NA    NA
## 2          <NA>        <NA>     NA    NA
## 3          <NA>        <NA>     NA    NA
## 4          <NA>        <NA>     NA    NA
## 5          <NA>        <NA>     NA    NA
## 6          <NA>        <NA>     NA    NA
## 7          <NA>        <NA>     NA    NA
## 8          <NA>        <NA>     NA    NA
## 9          <NA>        <NA>     NA    NA
## 10         <NA>        <NA>     NA    NA
## 11         <NA>        <NA>     NA    NA
## 12         <NA>        <NA>     NA    NA
## 13         <NA>        <NA>     NA    NA
## 14         <NA>        <NA>     NA    NA
## 15         <NA>        <NA>     NA    NA
## 16         <NA>        <NA>     NA    NA
## 17         <NA>        <NA>     NA    NA
## 18         <NA>        <NA>     NA    NA
## 19         <NA>        <NA>     NA    NA
## 20         <NA>        <NA>     NA    NA
## 21         <NA>        <NA>     NA    NA
## 22         <NA>        <NA>     NA    NA
## 23         <NA>        <NA>     NA    NA
## 24         <NA>        <NA>     NA    NA
## 25         <NA>        <NA>     NA    NA
## 26         <NA>        <NA>     NA    NA
## 27         <NA>        <NA>     NA    NA
## 28         <NA>        <NA>     NA    NA
## 29         <NA>        <NA>     NA    NA
## 30         <NA>        <NA>     NA    NA
```

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

```
##    Product_Name Description Rating Price
## 1          <NA>        <NA>     NA    NA
## 2          <NA>        <NA>     NA    NA
## 3          <NA>        <NA>     NA    NA
## 4          <NA>        <NA>     NA    NA
## 5          <NA>        <NA>     NA    NA
## 6          <NA>        <NA>     NA    NA
## 7          <NA>        <NA>     NA    NA
## 8          <NA>        <NA>     NA    NA
## 9          <NA>        <NA>     NA    NA
## 10         <NA>        <NA>     NA    NA
## 11         <NA>        <NA>     NA    NA
## 12         <NA>        <NA>     NA    NA
## 13         <NA>        <NA>     NA    NA
## 14         <NA>        <NA>     NA    NA
## 15         <NA>        <NA>     NA    NA
## 16         <NA>        <NA>     NA    NA
## 17         <NA>        <NA>     NA    NA
## 18         <NA>        <NA>     NA    NA
## 19         <NA>        <NA>     NA    NA
## 20         <NA>        <NA>     NA    NA
## 21         <NA>        <NA>     NA    NA
## 22         <NA>        <NA>     NA    NA
## 23         <NA>        <NA>     NA    NA
## 24         <NA>        <NA>     NA    NA
## 25         <NA>        <NA>     NA    NA
## 26         <NA>        <NA>     NA    NA
## 27         <NA>        <NA>     NA    NA
## 28         <NA>        <NA>     NA    NA
## 29         <NA>        <NA>     NA    NA
## 30         <NA>        <NA>     NA    NA
```

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

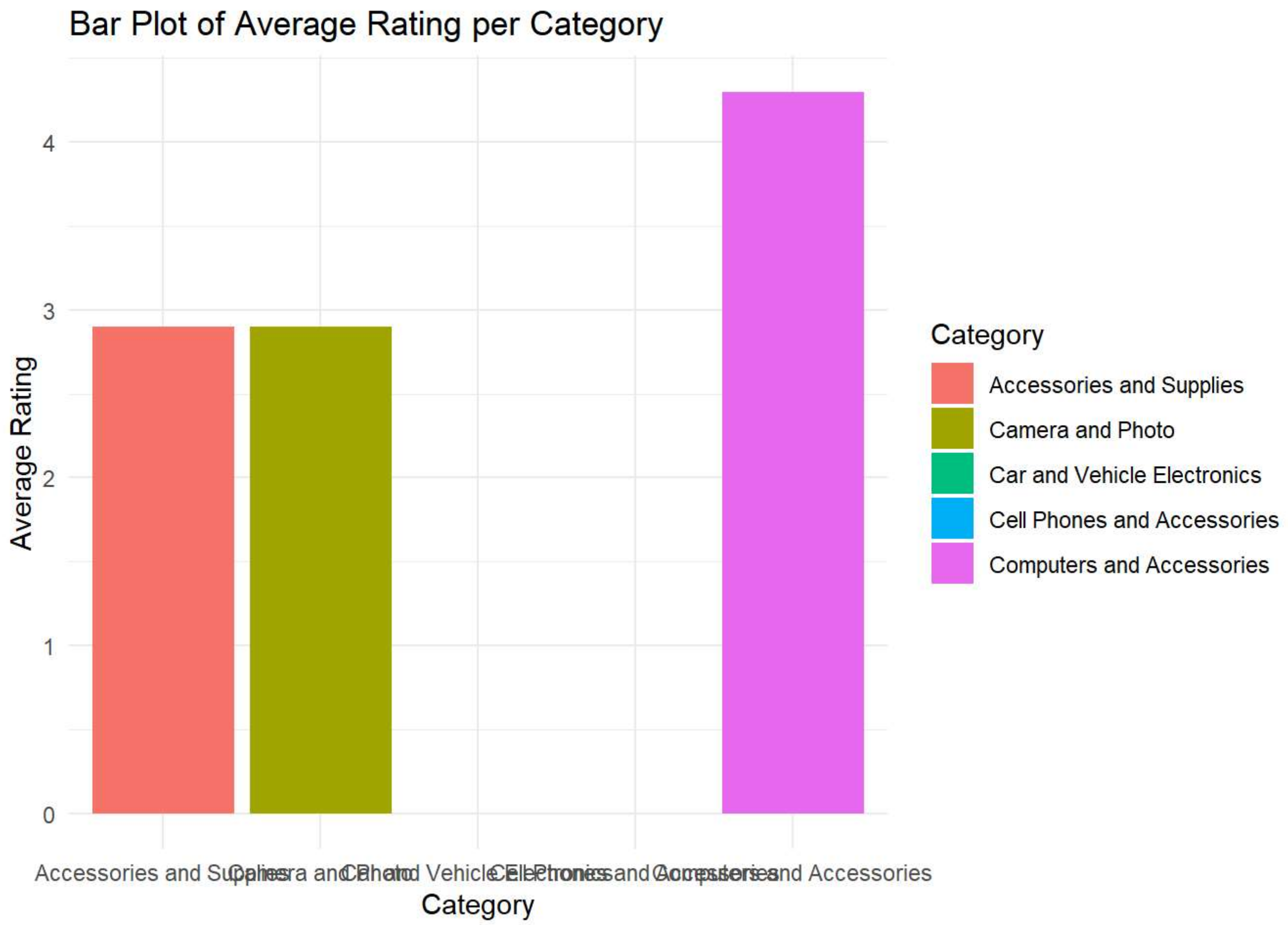
```
##                        Product_Name Description Rating Price
## 1  Logitech 720p Webcam Pro 9000          <NA>    4.3    NA
## 2                             <NA>        <NA>     NA    NA
## 3                             <NA>        <NA>     NA    NA
## 4                             <NA>        <NA>     NA    NA
## 5                             <NA>        <NA>     NA    NA
## 6                             <NA>        <NA>     NA    NA
## 7                             <NA>        <NA>     NA    NA
## 8                             <NA>        <NA>     NA    NA
## 9                             <NA>        <NA>     NA    NA
## 10                            <NA>        <NA>     NA    NA
## 11                            <NA>        <NA>     NA    NA
## 12                            <NA>        <NA>     NA    NA
## 13                            <NA>        <NA>     NA    NA
## 14                            <NA>        <NA>     NA    NA
## 15                            <NA>        <NA>     NA    NA
## 16                            <NA>        <NA>     NA    NA
## 17                            <NA>        <NA>     NA    NA
## 18                            <NA>        <NA>     NA    NA
## 19                            <NA>        <NA>     NA    NA
## 20                            <NA>        <NA>     NA    NA
## 21                            <NA>        <NA>     NA    NA
## 22                            <NA>        <NA>     NA    NA
## 23                            <NA>        <NA>     NA    NA
## 24                            <NA>        <NA>     NA    NA
## 25                            <NA>        <NA>     NA    NA
## 26                            <NA>        <NA>     NA    NA
## 27                            <NA>        <NA>     NA    NA
## 28                            <NA>        <NA>     NA    NA
## 29                            <NA>        <NA>     NA    NA
## 30                            <NA>        <NA>     NA    NA
```

```
#6.
#Our code scraped the first 30 elements of the product's name, price, description, ratings and reviews. There are a total of 5 categories and each containing 30 products so the product equal all in all 150 products.
```

```
#7

#The data we have collected can be used for a variety of purposes such as determining the top 30 products that appears when selecting a certain category. We can also determine the product's name, price, ratings, description, and reviews which can totally save a shopper's time by scrolling through each one.
```

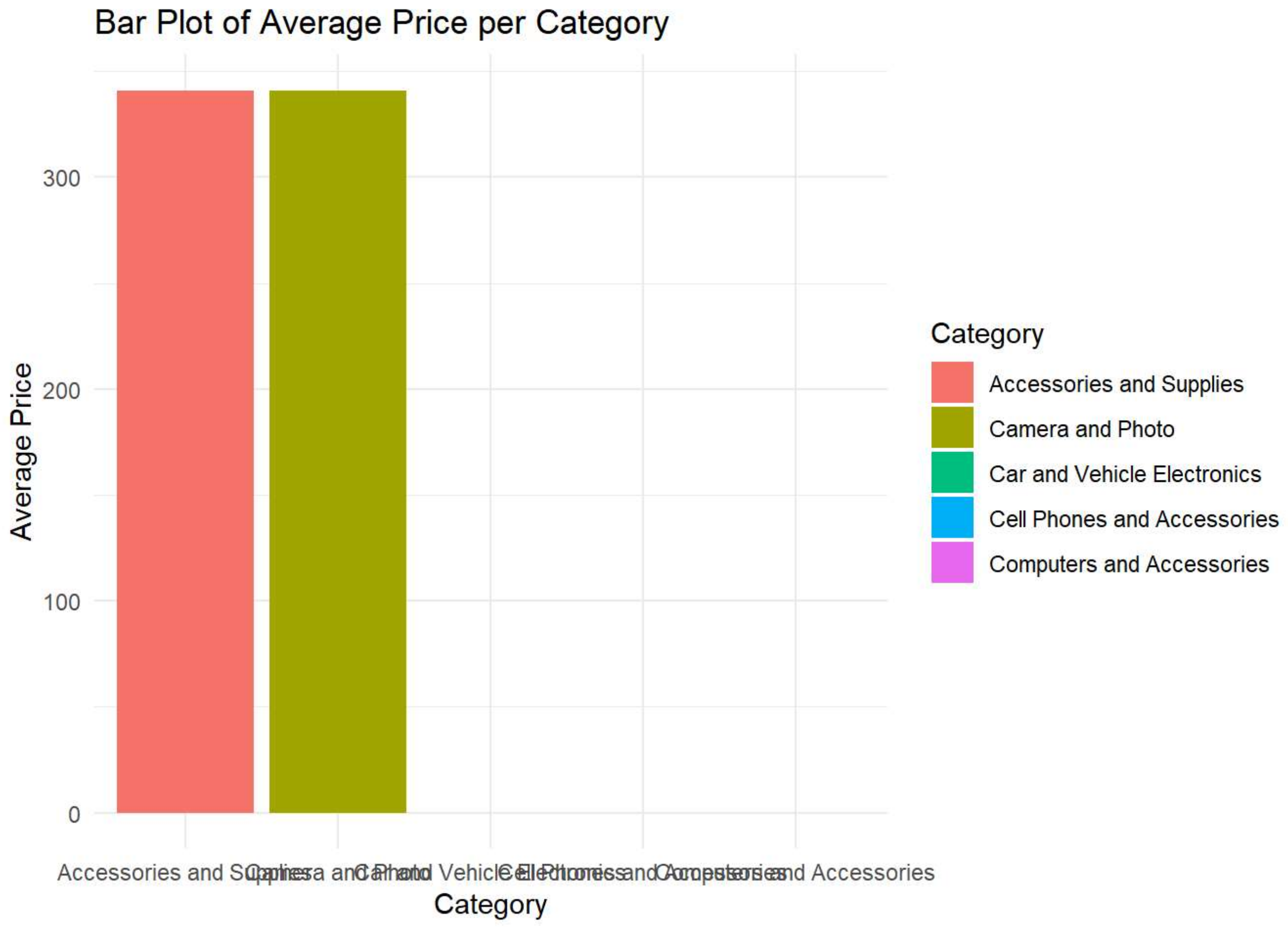
```
#8
merged_df <- do.call(rbind, df)
merged_df$Category <- rep(c("Accessories and Supplies", "Camera and Photo", "Car and Vehicle Ele
ctronics", "Cell Phones and Accessories", "Computers and Accessories"), each = 30)

rating_average <- merged_df %>%
  group_by(Category) %>%
  summarize(Average_Ratings = mean(Rating, na.rm = TRUE))

ggplot(rating_average, aes(x = Category, y = Average_Ratings, fill = Category)) + geom_bar(stat
= "identity") + labs(title = "Bar Plot of Average Rating per Category", x = "Category", y = "Ave
rage Rating") + theme_minimal()
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_bar()`).
```



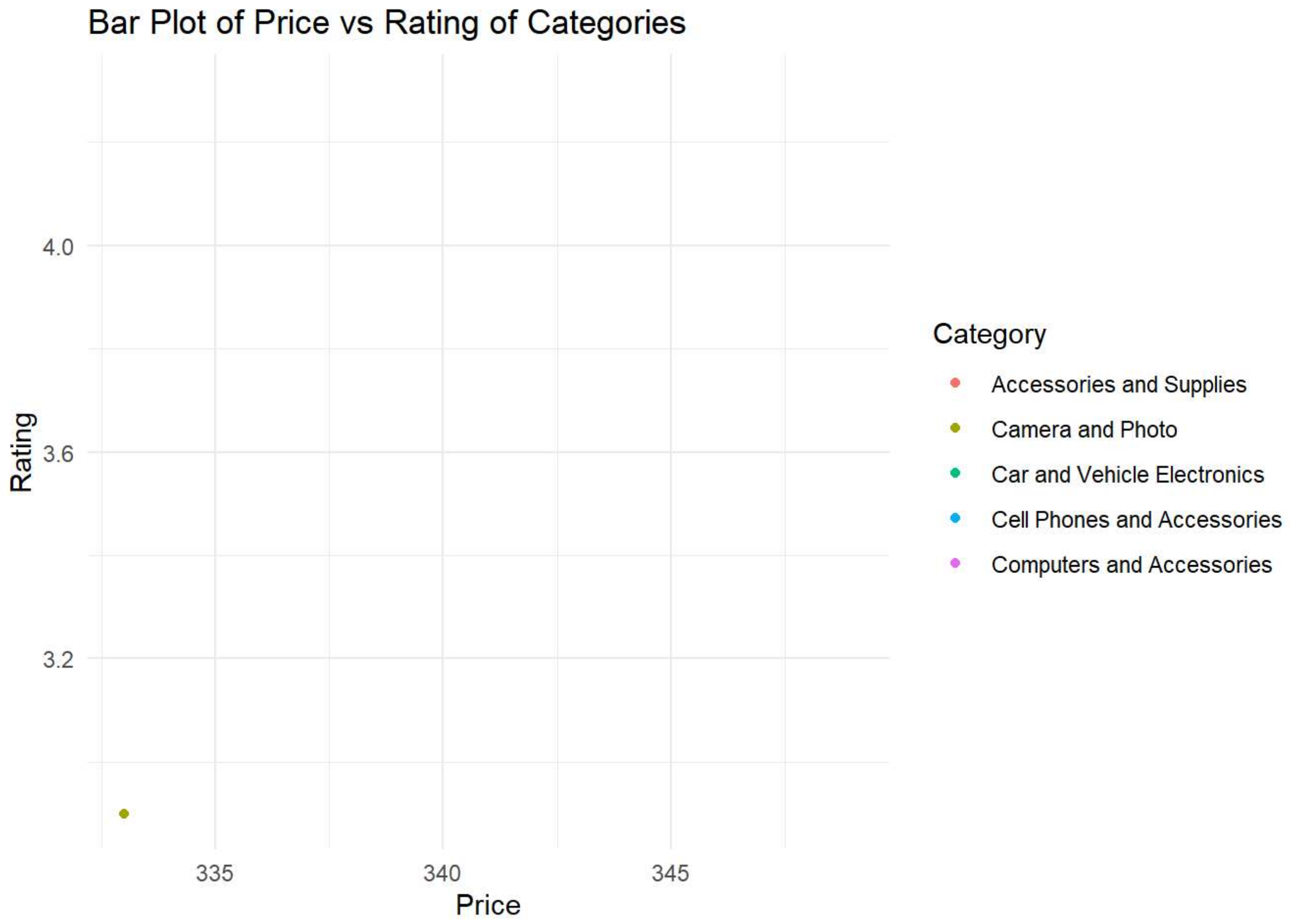
```
avg_price <- merged_df %>%
  group_by(Category) %>%
  summarize(Average_Price = mean(Price, na.rm = TRUE))

ggplot(avg_price, aes(x = Category, y = Average_Price, fill = Category)) +
  geom_bar(stat = "identity") +
  labs(title = "Bar Plot of Average Price per Category", x = "Category", y = "Average Price") +
  theme_minimal()
```

```
## Warning: Removed 3 rows containing missing values or values outside the scale range
## (`geom_bar()`).
```



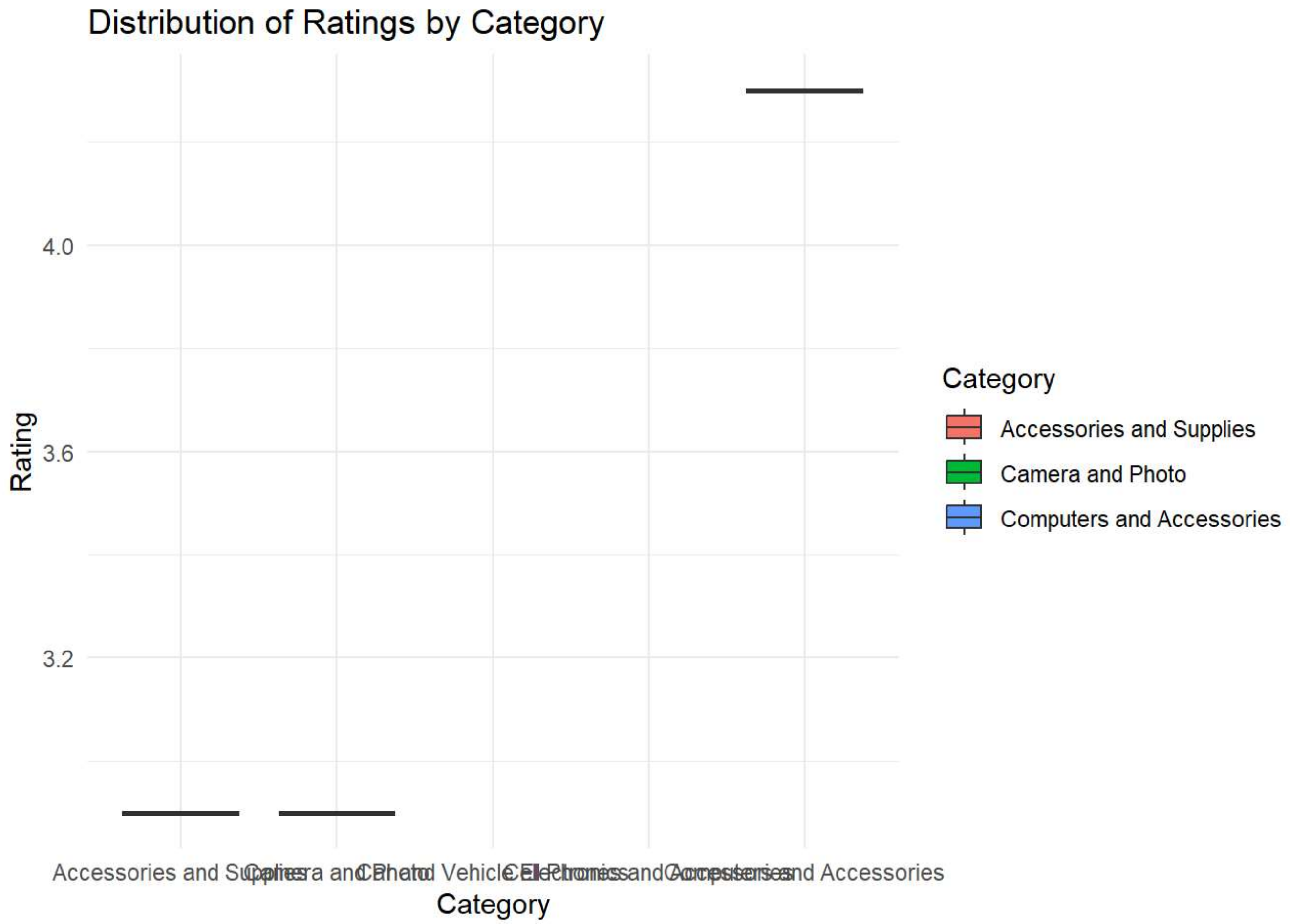
```
ggplot(merged_df, aes(x = Price, y = Rating, color = Category)) +
  geom_point() +
  labs(title = "Bar Plot of Price vs Rating of Categories", x = "Price", y = "Rating") +
  theme_minimal()
```

```
## Warning: Removed 148 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



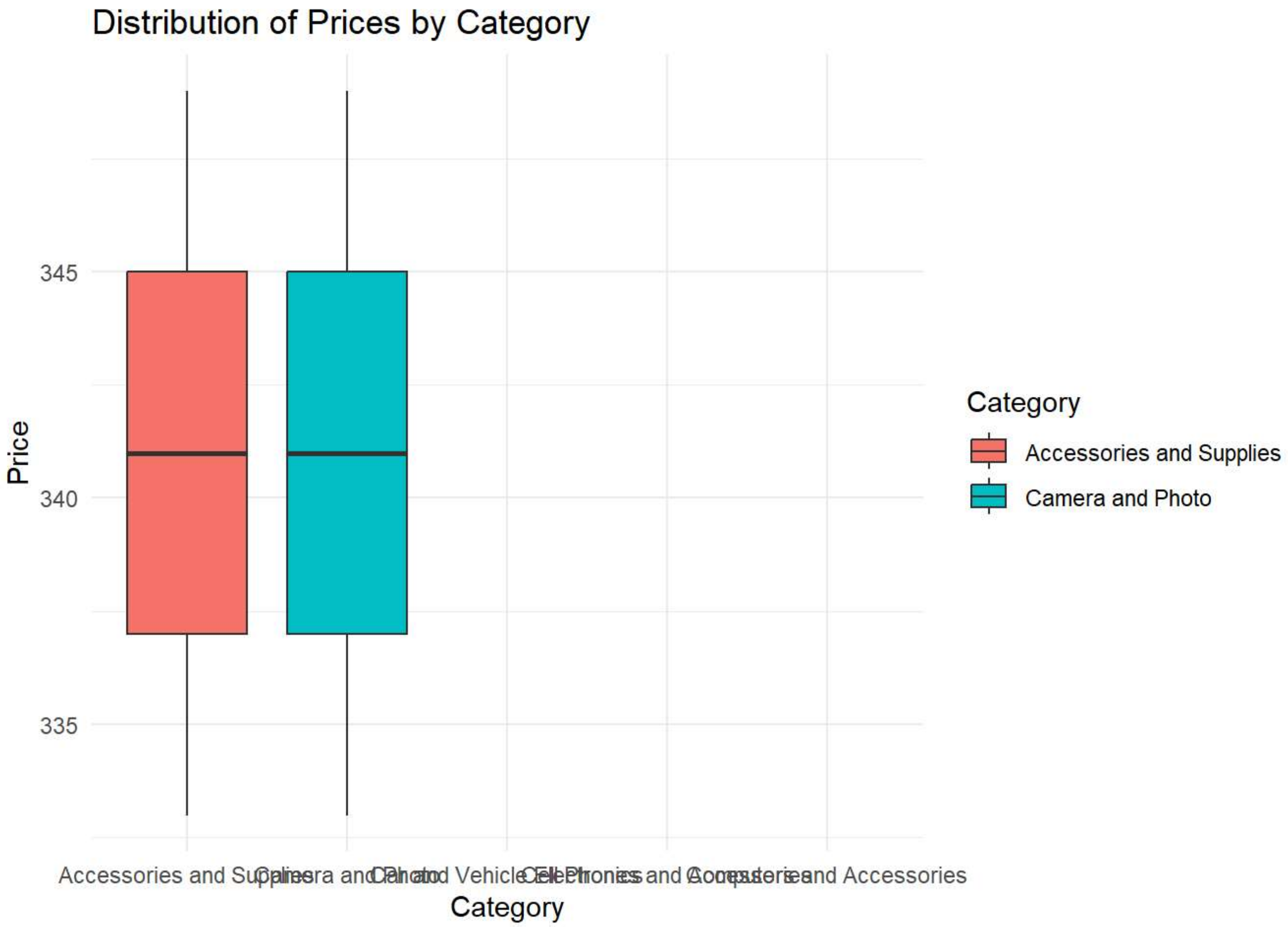
```
#9
ggplot(merged_df, aes(x = Category, y = Rating, fill = Category)) +
  geom_boxplot() +
  labs(title = "Distribution of Ratings by Category", x = "Category", y = "Rating") +
  theme_minimal()
```

```
## Warning: Removed 147 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



```
ggplot(merged_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 146 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```


Distribution of Prices by Category
345
340
335
Price
Accessories and Supplies
Camera and Photo
Car and Vehicle Electronics
Cell Phones and Accessories
Computers and Accessories
Category
Category
Accessories and Supplies
Camera and Photo

```
#10
ranked_elements <- lapply(df, function(df_category) {
  df_category %>%
    arrange(desc(Rating), Price) %>%
    mutate(Rank = row_number()) %>%
    select(Rank, everything())
})

categories <- c("Accessories and Supplies", "Camera and Photo", "Car and Vehicle Electronics",
"Cell Phones and Accessories", "Computers and Accessories")

for (i in seq_along(ranked_elements)) {
  ranked_elements[[i]]$Category <- categories[i]
}

arranged_merged_df <- do.call(rbind, ranked_elements)
arranged_merged_df <- arranged_merged_df %>%
  arrange(Category, Rank) %>%
  group_by(Category) %>%
  select(Rank, Category, everything()) %>%
  slice(1:5)


colnames(arranged_merged_df) <- c("Rank", "Category", "Product Name", "Product Description", "Ra
ting", "Price")
print(arranged_merged_df)
```

```
## # A tibble: 25 × 6
## # Groups:   Category [5]
##     Rank Category              `Product Name` `Product Description` Rating Price
##    <int> <chr>                 <chr>          <chr>                  <dbl> <dbl>
##  1     1 Accessories and Supp… "Datacolor Sp… <NA>                     2.9  333.
##  2     2 Accessories and Supp…  <NA>          <NA>                    NA    349
##  3     3 Accessories and Supp…  <NA>          <NA>                    NA     NA
##  4     4 Accessories and Supp…  <NA>          <NA>                    NA     NA
##  5     5 Accessories and Supp…  <NA>          <NA>                    NA     NA
##  6     1 Camera and Photo      "Datacolor Sp… <NA>                     2.9  333.
##  7     2 Camera and Photo       <NA>          <NA>                    NA    349
##  8     3 Camera and Photo       <NA>          <NA>                    NA     NA
##  9     4 Camera and Photo       <NA>          <NA>                    NA     NA
## 10     5 Camera and Photo       <NA>          <NA>                    NA     NA
## # ℹ 15 more rows
```

```
write.csv(arranged_merged_df, file = "ScrapedAmazonData.csv", row.names = FALSE)
```