RWorksheet#5a

Condag, Gagante, and Gonzaga

2024-11-06

Extracting IMDB.

1. Each group needs to extract the top 50 tv shows in Imdb.com. It will include the rank, the title of the tv show, tv rating, the number of people who voted, the number of episodes, the year it was released.

```
library(polite)

polite::use_manners(save_as = 'polite_scrape.R')
```

```
## v Setting active project to "/cloud/project".
```

```
url <- 'https://www.imdb.com/chart/top/?ref_=nv_mv_250&sort=rank%2Casc'

session <- bow(url,
               user_agent = "Educational")
session
```

```
## <polite session> https://www.imdb.com/chart/top/?ref_=nv_mv_250&sort=rank%2Casc
##     User-agent: Educational
##     robots.txt: 35 rules are defined for 3 bots
##    Crawl delay: 5 sec
##   The path is scrapable for this user-agent
```

```
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(httr)

title_list2 <- scrape(session) %>%
  html_nodes('h3.ipc-title__text') %>%
  html_text()

title_list_sub <- as.data.frame(title_list2[2:26])
colnames(title_list_sub) <- "Ranks"
```

```
split_df <- strsplit(as.character(title_list_sub$Ranks), ".", fixed = TRUE)
split_df <- data.frame(do.call(rbind, split_df))
colnames(split_df) <- c("Ranks", "Title")
titleAndRank <- data.frame(split_df)

ratings <- scrape(session) %>%
  html_nodes('span.ipc-rating-star--rating') %>%
  html_text()

if (length(ratings) < nrow(titleAndRank)) {
  ratings <- c(ratings, rep(NA, nrow(titleAndRank) - length(ratings)))
}
ratingsDf <- data.frame(ratings)

numberOfPeopleVoted <- scrape(session) %>%
  html_nodes('span.ipc-rating-star--voteCount') %>%
  html_text()

cleanedVotes <- gsub('[()]', '', numberOfPeopleVoted)

if (length(cleanedVotes) < nrow(titleAndRank)) {
  cleanedVotes <- c(cleanedVotes, rep(NA, nrow(titleAndRank) - length(cleanedVotes)))
}
cleanedVotesDf <- data.frame(cleanedVotes)

numEpisodes <- scrape(session) %>%
  html_nodes('span.sc-5bc66c50-6.OOdsw.cli-title-metadata-item:nth-of-type(2)') %>%
  html_text()

if (length(numEpisodes) < nrow(titleAndRank)) {
  numEpisodes <- c(numEpisodes, rep(NA, nrow(titleAndRank) - length(numEpisodes)))
}
numEpisodesDf <- data.frame(numEpisodes)

Year <- scrape(session) %>%
  html_nodes('span.sc-5bc66c50-6.OOdsw.cli-title-metadata-item:nth-of-type(1)') %>%
  html_text()

if (length(Year) < nrow(titleAndRank)) {
  Year <- c(Year, rep(NA, nrow(titleAndRank) - length(Year)))
}
YearDf <- data.frame(Year)

topShows <- cbind(titleAndRank, ratingsDf, cleanedVotesDf, numEpisodesDf, YearDf)

topShows
```

```
##    Ranks                                          Title ratings
## 1      1                       The Shawshank Redemption     9.3
## 2      2                                  The Godfather     9.2
## 3      3                                The Dark Knight     9.0
## 4      4                         The Godfather Part II     9.0
## 5      5                                  12 Angry Men     9.0
## 6      6      The Lord of the Rings: The Return of the King     9.0
## 7      7                                       Schindler's List     9.0
## 8      8                                           Pulp Fiction     8.9
## 9      9  The Lord of the Rings: The Fellowship of the Ring     8.9
## 10    10                         The Good, the Bad and the Ugly     8.8
## 11    11                                           Forrest Gump     8.8
## 12    12                  The Lord of the Rings: The Two Towers     8.8
## 13    13                                             Fight Club     8.8
## 14    14                                              Inception     8.8
## 15    15     Star Wars: Episode V - The Empire Strikes Back     8.7
## 16    16                                             The Matrix     8.7
## 17    17                                             Goodfellas     8.7
## 18    18                        One Flew Over the Cuckoo's Nest     8.7
## 19    19                                           Interstellar     8.7
## 20    20                                                 Se7en     8.6
## 21    21                                  It's a Wonderful Life     8.6
## 22    22                                          Seven Samurai     8.6
## 23    23                               The Silence of the Lambs     8.6
## 24    24                                    Saving Private Ryan     8.6
## 25    25                                            City of God     8.6
##    cleanedVotes numEpisodes Year
## 1            3M        <NA> <NA>
## 2          2.1M        <NA> <NA>
## 3            3M        <NA> <NA>
## 4          1.4M        <NA> <NA>
## 5          898K        <NA> <NA>
## 6            2M        <NA> <NA>
## 7          1.5M        <NA> <NA>
## 8          2.3M        <NA> <NA>
## 9          2.1M        <NA> <NA>
## 10         834K        <NA> <NA>
## 11         2.3M        <NA> <NA>
## 12         1.8M        <NA> <NA>
## 13         2.4M        <NA> <NA>
## 14         2.6M        <NA> <NA>
## 15         1.4M        <NA> <NA>
## 16         2.1M        <NA> <NA>
## 17         1.3M        <NA> <NA>
## 18         1.1M        <NA> <NA>
## 19         2.2M        <NA> <NA>
## 20         1.9M        <NA> <NA>
## 21         513K        <NA> <NA>
## 22         377K        <NA> <NA>
## 23         1.6M        <NA> <NA>
## 24         1.5M        <NA> <NA>
## 25         822K        <NA> <NA>
```

B. It will also include the number of user reviews and the number of critic reviews, as well as the popularity rating for each tv shows.

```
homePage <- 'https://www.imdb.com/chart/toptv/'
mainPage <- read_html(homePage)

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

```
showInfo <- lapply(links, function(link) {
  fullLink <- paste0("https://imdb.com", link)

  userRevLink <- read_html(fullLink)
  userRevPageLink <-  userRevLink  %>%
    html_nodes('a.isReview') %>%
    html_attr("href")

  criticRev <- userRevLink %>%
            html_nodes("span.score") %>%
            html_text()
  criticDf <- data.frame(Critic_Reviews = criticRev[2], stringsAsFactors = FALSE)

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

  userRev <- read_html(paste0("https://imdb.com",  userRevPageLink[1]))
  userRevCount <- userRev %>%
    html_nodes('[data-testid="tturv-total-reviews"]') %>%
    html_text()

  return(data.frame(User_Reviews = userRevCount, Critic = criticDf, Popularity_Rating = popularityRatin
})


showUrlDf <- do.call(rbind, showInfo)
showUrlDf
```

```
##     User_Reviews Critic_Reviews Popularity_Rating
## 1  5,120 reviews            176                24
## 2  5,120 reviews            176                24
## 3    158 reviews              6             1,082
## 4    158 reviews              6             1,082
## 5    111 reviews             10             2,066
## 6    111 reviews             10             2,066
## 7  1,059 reviews             34               173
## 8  1,059 reviews             34               173
## 9  3,541 reviews             88               146
## 10 3,541 reviews             88               146
## 11   787 reviews             77               112
## 12   787 reviews             77               112
## 13 1,002 reviews             57               327
## 14 1,002 reviews             57               327
## 15    53 reviews              9             4,399
## 16    53 reviews              9             4,399
## 17   966 reviews             93                26
## 18   966 reviews             93                26
## 19   205 reviews             12             1,457
## 20   205 reviews             12             1,457
## 21    80 reviews              8             3,902
## 22    80 reviews              8             3,902
## 23   245 reviews             15             2,798
## 24   245 reviews             15            2,798
## 25 5,915 reviews            368               16
## 26 5,915 reviews            368               16
## 27   369 reviews              4              368
## 28   369 reviews              4              368
## 29   126 reviews              5            2,290
## 30   126 reviews              5            2,290
## 31   468 reviews             16              505
## 32   468 reviews             16              505
## 33   909 reviews             94              126
## 34   909 reviews             94              126
## 35    12 reviews              9            3,107
## 36    12 reviews              9            3,107
## 37   541 reviews             28            1,593
## 38   541 reviews             28            1,593
## 39   214 reviews             85              366
## 40   214 reviews             85              366
## 41   175 reviews             13            1,841
## 42   175 reviews             13            1,841
## 43 1,098 reviews            121              172
## 44 1,098 reviews            121              172
## 45 2,376 reviews             65               52
## 46 2,376 reviews             65               52
## 47   219 reviews             25              518
## 48   219 reviews             25              518
## 49 2,301 reviews             59                2
## 50 2,301 reviews             59                2
```

```
allShows <- cbind(topShows, showUrlDf)
allShows
```

```
##    Ranks                                          Title ratings
## 1      1                       The Shawshank Redemption     9.3
## 2      2                                  The Godfather     9.2
## 3      3                                The Dark Knight     9.0
## 4      4                          The Godfather Part II     9.0
## 5      5                                   12 Angry Men     9.0
## 6      6      The Lord of the Rings: The Return of the King     9.0
## 7      7                                 Schindler's List     9.0
## 8      8                                   Pulp Fiction     8.9
## 9      9  The Lord of the Rings: The Fellowship of the Ring     8.9
## 10    10                    The Good, the Bad and the Ugly     8.8
## 11    11                                   Forrest Gump     8.8
## 12    12              The Lord of the Rings: The Two Towers     8.8
## 13    13                                     Fight Club     8.8
## 14    14                                      Inception     8.8
## 15    15     Star Wars: Episode V - The Empire Strikes Back     8.7
## 16    16                                     The Matrix     8.7
## 17    17                                     Goodfellas     8.7
## 18    18                  One Flew Over the Cuckoo's Nest     8.7
## 19    19                                   Interstellar     8.7
## 20    20                                          Se7en     8.6
## 21    21                           It's a Wonderful Life     8.6
## 22    22                                  Seven Samurai     8.6
## 23    23                       The Silence of the Lambs     8.6
## 24    24                                 Saving Private Ryan     8.6
## 25    25                                         City of God     8.6
## 26     1                            The Shawshank Redemption     9.3
## 27     2                                       The Godfather     9.2
## 28     3                                     The Dark Knight     9.0
## 29     4                               The Godfather Part II     9.0
## 30     5                                        12 Angry Men     9.0
## 31     6      The Lord of the Rings: The Return of the King     9.0
## 32     7                                    Schindler's List     9.0
## 33     8                                        Pulp Fiction     8.9
## 34     9  The Lord of the Rings: The Fellowship of the Ring     8.9
## 35    10                     The Good, the Bad and the Ugly     8.8
## 36    11                                        Forrest Gump     8.8
## 37    12              The Lord of the Rings: The Two Towers     8.8
## 38    13                                          Fight Club     8.8
## 39    14                                           Inception     8.8
## 40    15     Star Wars: Episode V - The Empire Strikes Back     8.7
## 41    16                                          The Matrix     8.7
## 42    17                                          Goodfellas     8.7
## 43    18                    One Flew Over the Cuckoo's Nest     8.7
## 44    19                                        Interstellar     8.7
## 45    20                                               Se7en     8.6
## 46    21                               It's a Wonderful Life     8.6
## 47    22                                       Seven Samurai     8.6
## 48    23                           The Silence of the Lambs     8.6
## 49    24                                 Saving Private Ryan     8.6
## 50    25                                         City of God     8.6
##    cleanedVotes numEpisodes Year  User_Reviews Critic_Reviews Popularity_Rating
## 1            3M        <NA> <NA> 5,120 reviews            176                24
## 2          2.1M        <NA> <NA> 5,120 reviews            176                24
## 3            3M        <NA> <NA>   158 reviews              6             1,082
## 4          1.4M        <NA> <NA>   158 reviews              6             1,082
## 5          898K        <NA> <NA>   111 reviews             10             2,066
## 6            2M        <NA> <NA>   111 reviews             10             2,066
## 7          1.5M        <NA> <NA> 1,059 reviews             34               173
## 8          2.3M        <NA> <NA> 1,059 reviews             34               173
## 9          2.1M        <NA> <NA> 3,541 reviews             88               146
## 10         834K        <NA> <NA> 3,541 reviews             88               146
## 11         2.3M        <NA> <NA>   787 reviews             77               112
## 12         1.8M        <NA> <NA>   787 reviews             77               112
## 13         2.4M        <NA> <NA> 1,002 reviews             57               327
## 14         2.6M        <NA> <NA> 1,002 reviews             57               327
## 15         1.4M        <NA> <NA>    53 reviews              9             4,399
## 16         2.1M        <NA> <NA>    53 reviews              9             4,399
## 17         1.3M        <NA> <NA>   966 reviews             93                26
## 18         1.1M        <NA> <NA>   966 reviews             93                26
## 19         2.2M        <NA> <NA>   205 reviews             12             1,457
## 20         1.9M        <NA> <NA>   205 reviews             12             1,457
## 21         513K        <NA> <NA>    80 reviews              8             3,902
## 22         377K        <NA> <NA>    80 reviews              8             3,902
## 23         1.6M        <NA> <NA>   245 reviews             15             2,798
## 24         1.5M        <NA> <NA>   245 reviews             15             2,798
## 25         822K        <NA> <NA> 5,915 reviews            368                16
## 26           3M        <NA> <NA> 5,915 reviews            368                16
## 27         2.1M        <NA> <NA>   369 reviews             4             368
## 28           3M        <NA> <NA>   369 reviews             4             368
## 29         1.4M        <NA> <NA>   126 reviews             5           2,290
## 30         898K        <NA> <NA>   126 reviews             5           2,290
## 31           2M        <NA> <NA>   468 reviews            16             505
## 32         1.5M        <NA> <NA>   468 reviews            16             505
## 33         2.3M        <NA> <NA>   909 reviews            94             126
## 34         2.1M        <NA> <NA>   909 reviews            94             126
## 35         834K        <NA> <NA>    12 reviews             9           3,107
## 36         2.3M        <NA> <NA>    12 reviews             9           3,107
## 37         1.8M        <NA> <NA>   541 reviews            28           1,593
## 38         2.4M        <NA> <NA>   541 reviews            28           1,593
## 39         2.6M        <NA> <NA>   214 reviews            85             366
## 40         1.4M        <NA> <NA>   214 reviews            85             366
## 41         2.1M        <NA> <NA>   175 reviews            13           1,841
## 42         1.3M        <NA> <NA>   175 reviews            13           1,841
## 43         1.1M        <NA> <NA> 1,098 reviews           121             172
## 44         2.2M        <NA> <NA> 1,098 reviews           121             172
## 45         1.9M        <NA> <NA> 2,376 reviews            65              52
## 46         513K        <NA> <NA> 2,376 reviews            65              52
## 47         377K        <NA> <NA>   219 reviews            25             518
## 48         1.6M        <NA> <NA>   219 reviews            25             518
## 49         1.5M        <NA> <NA> 2,301 reviews            59               2
## 50         822K        <NA> <NA> 2,301 reviews            59               2
```

2. From the 50 tv shows, select at least 5 tv shows to scrape 20 user reviews that will include the reviewer's name, date of reviewed, user rating, title of the review, the numbers for "is helpful" and "is not helpful", and text reviews.

```r
library(rvest)
library(dplyr)

urlsOfFiveShows <- c(
  "https://www.imdb.com/title/tt0903747/reviews/?ref_=ttexr_ql_2",
  "https://www.imdb.com/title/tt5491994/reviews/?ref_=tt_ov_ql_2",
  "https://www.imdb.com/title/tt0185906/reviews/?ref_=tt_ov_ql_2",
  "https://www.imdb.com/title/tt7366338/reviews/?ref_=tt_ov_ql_2",
  "https://www.imdb.com/title/tt0944947/reviews/?ref_=tt_ov_ql_2"
)

fiveShowsUrlDf <- data.frame(
  Title = c(
    "Breaking Bad",
    "Planet Earth II",
    "Band of Brothers",
    "Chernobyl",
    "Game of Thrones"
  ),
  URLs = urlsOfFiveShows
)

scrapeReviews <- function(show_url) {
  page <- read_html(show_url)

  userNames <- page %>%
    html_nodes('[data-testid="author-link"]') %>%
    html_text()

   reviewDates <- page %>%
    html_nodes('li.review-date') %>%
    html_text()

     userRating <- page %>%
    html_nodes('span.ipc-rating-star--rating') %>%
    html_text()

     revTitle <- page %>%
    html_nodes('h3.ipc-title__text') %>%
    html_text()

     helpfulRev <- page %>%
    html_nodes('span.count--up') %>%
    html_text()

      notHelpful <- page %>%
    html_nodes('span.count--down') %>%
    html_text()

       data.frame(Usernames = head(userNames, 20), Dates = head(reviewDates, 20), userRating = head(use
}

reviews_data <- lapply(fiveShowsUrlDf$URLs, scrapeReviews)
names(reviews_data) <- fiveShowsUrlDf$Title
reviews_data[["Breaking Bad"]]
```

```
##              Usernames        Dates userRating
## 1              FiRE010  Jul 3, 2021         10
## 2           bruhperson  Mar 6, 2019         10
## 3          KinoKoopaKid Jul 29, 2021        10
## 4          jehuschultz Feb 18, 2020         10
## 5        Supermanfan-13  Nov 8, 2021        10
## 6   manishsingh-03299 May 30, 2019         10
## 7              Rob1331  Dec 8, 2022         10
## 8             xpinerhd Nov 15, 2019         10
## 9  dhanushreddy-14919 Jul 17, 2021         10
## 10         tushv-31482  Dec 8, 2022         10
## 11             dyarutd Sep 28, 2024          7
## 12        Shopaholic35 Feb 11, 2014          5
## 13        vlucky-40551  Mar 7, 2021         10
## 14  TheLittleSongbird Nov 12, 2017         10
## 15         FishDrowned  Nov 8, 2021         10
## 16         gogoschka-1 Jan 11, 2014         10
## 17          joegalgano Aug 11, 2021         10
## 18        agatt-87232 May 19, 2019         10
## 19      Leofwine_draca  May 4, 2021         10
## 20       dristysultana Jun 23, 2021         10
##
## 1
## 2
## 3
## 4
## 5
## 6                                                                                                       Those days a
## 7
## 8                                                                                                                   I
## 9                                                                                                                   A
## 10                                                                                                               Once
## 11                                                                                                       My Review For
## 12                                                                                                     Fantastic show
## 13                                                                                                               Pret
## 14                                                                                         Among the best and most ac
## 15                                                                                               By far the greatest
## 16 If you mix Scarface, Robin Hood and maybe Tyler Durden with enough meth - you'll get a mean cockta
## 17                                                                                                          in a ca
## 18                                                                         Since GOT is over, this is Officially the C
## 19                                                                                                     Every bit a
## 20
```

```r
reviews_data[["Planet Earth II"]]
```

```
##            Usernames        Dates userRating
## 1        arjanhylkema  Nov 7, 2016         10
## 2            Wentloog  Nov 5, 2016         10
## 3       john-m-madsen  Nov 5, 2016         10
## 4        thespookybuz  Nov 9, 2016         10
## 5         pjdickinson  Nov 5, 2016         10
## 6            dbijis33  Nov 8, 2016         10
## 7      dhanrajjughead Nov 17, 2016         10
## 8         NeilBarnett Nov 13, 2016         10
## 9       salmanu-27386  Nov 6, 2016         10
## 10 panagiotiskatsanos Dec 31, 2016         10
## 11            ianrobo Nov 19, 2016         10
## 12         adamonIMDb Dec 28, 2016          7
## 13          tinyfordst May 19, 2019         10
## 14       larask-21775 Oct 20, 2018         10
## 15         BobFillmore Sep 29, 2017         10
## 16      farshidkarimi Nov 22, 2016         10
## 17  TheLittleSongbird Oct 12, 2017         10
## 18   myersei-165-4350  Dec 4, 2016         10
## 19  fierceeagle-40009 Apr 23, 2020         10
## 20      adam-whitmore  Jan 5, 2017         10
##
## 1
## 2                                                                                           At once awe-inspiring a
## 3                                                                       Yet another masterpiece from BBC Nature & Davi
## 4
## 5
## 6                                                                                                             Dangero
## 7                                                                                                 Greatest documentary
## 8                                                                                         Best thing on TV since las
## 9
## 10                                                                                     One of the best documentaries I
## 11                                                                                                 In times of climate
## 12
## 13                                                      More irritated with IMDb for the bias than
## 14
## 15                                                                         Should be required view
## 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                                                                                       Absol
## 20                                                                 Peerless evocation of nature ac
```

```
reviews_data[["Band of Brothers"]]
```

```
##                Usernames        Dates userRating
## 1                Rob1331 Sep 27, 2022         10
## 2            sanderson777 Oct 14, 2001         10
## 3             wildcatt268 Jan 18, 2002         10
## 4                arjay24 Apr 18, 2004         10
## 5              rbverhoef Feb 13, 2003         10
## 6             yodaschoda Jan 23, 2005         10
## 7     philip_vanderveken Sep 16, 2004         10
## 8          Supermanfan-13  May 6, 2022         10
## 9              thiagoutp  Nov 4, 2019         10
## 10            bsmith5552  Nov 5, 2001         10
## 11 faded_english_monkey Aug 25, 2004         10
## 12          planktonrules May 30, 2015          7
## 13         stilonkostrzyn Apr 10, 2021          5
## 14            faded_Glory  May 2, 2006         10
## 15               jazmodo  Jun 3, 2019         10
## 16              kait2007 Jan 26, 2005         10
## 17            mickman91-1  May 3, 2022         10
## 18            kipmcmillan Oct 24, 2018          9
## 19         erwan_ticheler  Dec 7, 2002         10
## 20              grahamsj3 Nov 25, 2002         10
##                                                        Review_Title
## 1                                                       Incredible!!
## 2                              Possibly the finest 10 hours ever created
## 3                                 One of the best war movies/series ever
## 4                                                          Realistic
## 5                                                          Excellent
## 6                        One of, if not the best, mini series' ever made
## 7                  This series is so unbelievably realistic, so authentic.
## 8                              One of the best mini-series ever created!
## 9                                                Probably the best ever
## 10                            Realistic WWII Drama With Warts Included
## 11                                                     war, no frills
## 12                                                You can't beat this....
## 13                                                        Overrated??
## 14                                                 Not very realistic at all
## 15                       Without Doubt, the Best Mini-Series Ever Recorded
## 16                                                    Great Miniseries
## 17 A series like this won't be made again (see below), so treasure it
## 18                                                Share With Your Children
## 19                                                   Best Mini series ever
## 20                            A-1, TOPS, the BOMB what else can I say?
```

```r
reviews_data[["Chernobyl"]]
```

```
##             Usernames        Dates userRating
## 1    curiosityonmars May 23, 2019         10
## 2           stelmakh May 10, 2019         10
## 3        natashapekar  May 9, 2019         10
## 4          m-porpaczi May 14, 2019         10
## 5            Lladerat  May 7, 2019         10
## 6            jfirebug May 20, 2019         10
## 7             thegldt  May 6, 2019         10
## 8  alexander-phoenix May 13, 2019         10
## 9          wmeduardowm  May 6, 2019         10
## 10    Leofwine_draca Nov 27, 2019         10
## 11       Jamie_Seaton May 23, 2019          7
## 12  garybarker-51255 Jul 31, 2019          5
## 13        tutajdaniel Jun 15, 2019          8
## 14            frimark May 20, 2019         10
## 15   krzysztof-18241 May 30, 2019         10
## 16         ahmetkozan  Jun 7, 2019          9
## 17            Rob1331 Sep 27, 2022         10
## 18     stephenpdodds  May 6, 2019          9
## 19    Supermanfan-13 Jul 10, 2022          9
## 20          emholberg May 26, 2019         10
##                                                       Review_Title
## 1                                                  They got it right
## 2                                               Goosebumps and tears
## 3                                         I highly recommend this film!
## 4                                         No hero wakes up wanting to die
## 5                                                 So far looks excellent
## 6                                                          Incredible
## 7                                Bleak, Unsettling, Haunting All Throughout
## 8                                                        Unbelievable
## 9                                                    HBO did it again!
## 10                                                          Exemplary
## 11                                                           Amazing!
## 12                               Great viewing bit the science is flawed
## 13 The movie is far from thuth. A lot of fake info to create a drama...
## 14                                                Emotionally drained...
## 15                                                     Just watch it (!)
## 16                               Now you look like the minister of coal!
## 17                                                        Must Watch!
## 18                                                          Cracking.
## 19                                                         Brilliant!
## 20             It is hard to overestimate the importance of this show.
```

```r
reviews_data[["Game of Thrones"]]
```

```
##             Usernames        Dates userRating
## 1          danielkpkp May 11, 2020          9
## 2   samxxx-671-826221 May 24, 2019          8
## 3           slowcando May 20, 2019          8
## 4            SaifOVGU  Apr 8, 2020         10
## 5    jacobnoble-02524  May 9, 2019          9
## 6       Dan_W_Reviews  Feb 5, 2023          9
## 7      heavenacceptme Aug 22, 2022         10
## 8      Supermanfan-13  Dec 9, 2023         10
## 9          tweaknhoe May 25, 2019          8
## 10  TheLittleSongbird  Nov 8, 2017         10
## 11            kunkell  Nov 1, 2019          6
## 12   akhil-marsonya27 May 20, 2019          1
## 13            Rob1331 Nov 10, 2023         10
## 14    alshamari-marwa May 18, 2020          9
## 15       tobiascramon  Nov 8, 2022          9
## 16 adamheritage-15333  May 5, 2019          9
## 17         LASTRONOME  Jun 2, 2020         10
## 18    SpitOnAStranger  May 9, 2019          8
## 19        psypeterson Apr 16, 2011         10
## 20            el-absy May 19, 2019          7
##                                                        Review_Title
## 1                      It could have been the best TV series ever made...
## 2                 A perfect example of: Falling in Love with the Wrong Guy
## 3             Seasons 1-6: outstanding. 7: daft but good. 8: disappointing
## 4                          Can you just make the remake the season finale?
## 5                                                        Game of Thrones
## 6                Captivating and Gripping but a Disappointing Final Season
## 7  Despite the final season, Game of Thrones remains an all time classic
## 8                                              One of the best shows ever
## 9                   Why just why? This show could have been the best ever.
## 10                                             This is a television show?
## 11                                              Imagine an Ice Cream Shop
## 12                                              A Message to Dan and Dave
## 13                                                                Amazing
## 14                                            Extraordinary untill season 8
## 15                                               one the best shows ever made
## 16  Can we please just restart season 8, perhaps 7 as well, but mainly 8
## 17                                                         almost perfect
## 18                                           This was an 10/10 until S08E03
## 19                                                  Excellent adaptation.
## 20                                           Great Beginning, WORST Ending!
```

3.

```r
library(ggplot2)
years <- substr(Year, 1,4)
years <- as.numeric(Year)

ggplot(data.frame(Year = years), aes(x = Year)) +
  geom_line(stat = "count", fill = "pink", color = "purple") +
  labs(title = "Number of TV Shows Released by Year",
       x = "Year",
       y = "Number of TV Shows") +
  theme_minimal()
```

```
## Warning in geom_line(stat = "count", fill = "pink", color = "purple"): Ignoring
## unknown parameters: `fill`

## Warning in min(x): no non-missing arguments to min; returning Inf

## Warning in max(x): no non-missing arguments to max; returning -Inf

## Warning in min(d[d > tolerance]): no non-missing arguments to min; returning
## Inf
```

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

```r
mostShowsYear <- as.data.frame(table(Year))
mostShowsYear <- mostShowsYear[which.max(mostShowsYear$Freq), ]
print(mostShowsYear)
```

```
## integer(0)
```

4 and 5. Select 5 categories from Amazon and select 30 products from each category. Extract the price, description, ratings and reviews of each product.

```r
library(httr)
library(polite)
library(rvest)

polite::use_manners(save_as = 'polite_scrape.R')

url <- 'https://www.amazon.com/'


session <- bow(url,
               user_agent = "Educational")
session
```

```
## <polite session> https://www.amazon.com/
##     User-agent: Educational
##     robots.txt: 138 rules are defined for 5 bots
##    Crawl delay: 5 sec
##   The path is scrapable for this user-agent
```

Shoes Category

```
shoes_url <- 'https://www.amazon.com/s?i=specialty-aps&bbn=16225020011&rh=n%3A7141123011%2Cn%3A1622502

scrape_shoes <- function(url, category) {
  page <- read_html(url)

  product_titles <- page %>%
    html_nodes("h2.a-size-mini") %>%
    html_text(trim = TRUE)

  price <- page %>%
    html_nodes(".a-price .a-offscreen") %>%
    html_text(trim = TRUE)

  ratings <- page %>%
    html_nodes("span.a-icon-alt") %>%
    html_text(trim = TRUE)

  review <- page %>%
    html_nodes("div.a-sectionr-celwidget") %>%
    html_text(trim = TRUE)

  data.frame(
    Product_titles = product_titles[1:30],
    Price = price[1:30],
    Ratings = ratings[1:30],
    Review = review[1:30]
  )
}

shoes_products <- scrape_shoes(shoes_url, "Shoes")
shoes_products
```

```
##       Product_titles   Price            Ratings Review
## 1                UGG $124.95 4.6 out of 5 stars   <NA>
## 2                UGG $119.99 4.4 out of 5 stars   <NA>
## 3              Crocs  $39.95 4.8 out of 5 stars   <NA>
## 4  Koolaburra by UGG  $69.99 4.6 out of 5 stars   <NA>
## 5  Koolaburra by UGG  $64.99 4.7 out of 5 stars   <NA>
## 6                UGG  $69.99 4.7 out of 5 stars   <NA>
## 7             adidas $139.95 4.5 out of 5 stars   <NA>
## 8                UGG  $64.95 4.7 out of 5 stars   <NA>
## 9            Nfinity  $70.00 4.6 out of 5 stars   <NA>
## 10            dubuto $149.95 4.6 out of 5 stars   <NA>
## 11               UGG $129.99 4.9 out of 5 stars   <NA>
## 12            Stelle $144.87 4.7 out of 5 stars   <NA>
## 13              BOGS  $15.99 4.7 out of 5 stars   <NA>
## 14               UGG  $16.99 3.6 out of 5 stars   <NA>
## 15               UGG  $69.95 4.7 out of 5 stars   <NA>
## 16 Koolaburra by UGG  $16.99 4.7 out of 5 stars   <NA>
## 17             Crocs  $23.99 4.7 out of 5 stars   <NA>
## 18 Koolaburra by UGG  $65.00 4.7 out of 5 stars   <NA>
## 19           Fadezar  $69.95 4.8 out of 5 stars   <NA>
## 20     Sesame Street  $74.95 4.7 out of 5 stars   <NA>
## 21          Hey Dude  $84.99 4.8 out of 5 stars   <NA>
## 22             Crocs  $69.99 4.9 out of 5 stars   <NA>
## 23 Koolaburra by UGG  $44.99 4.7 out of 5 stars   <NA>
## 24               UGG  $54.99 4.7 out of 5 stars   <NA>
## 25            adidas  $69.99 4.6 out of 5 stars   <NA>
## 26      Under Armour  $74.99 4.5 out of 5 stars   <NA>
## 27           KVbabby  $19.99 4.5 out of 5 stars   <NA>
## 28             Sorel  $16.00 4.8 out of 5 stars   <NA>
## 29 Koolaburra by UGG  $39.99 4.6 out of 5 stars   <NA>
## 30            KRABOR  $49.99 4.4 out of 5 stars   <NA>
```

Makeup Category

```
makeup_url <- 'https://www.amazon.com/s?i=specialty-aps&bbn=16225006011&rh=n%3A%2116225006011%2Cn%3A1105

scrape_makeup <- function(url, category) {
  page <- read_html(url)

  product_titles <- page %>%
    html_nodes("h2.a-size-mini") %>%
    html_text(trim = TRUE)

  price <- page %>%
    html_nodes(".a-price .a-offscreen") %>%
    html_text(trim = TRUE)

  ratings <- page %>%
    html_nodes("span.a-icon-alt") %>%
    html_text(trim = TRUE)

  review <- page %>%
    html_nodes("span.a-size-base.review-text") %>%
    html_text(trim = TRUE)

  data.frame(
    Product_titles = product_titles[1:30],
    Price = price[1:30],
    Ratings = ratings[1:30],
    Review = review[1:30]
  )
}

makeup_products <- scrape_makeup(makeup_url, "Makeup")
makeup_products
```

```
##    Product_titles Price Ratings Review
## 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>
```

Jewelry Category

```
jewelry_url <- 'https://www.amazon.com/s?i=specialty-aps&bbn=16225018011&rh=n%3A7141123011%2Cn%3A162250

scrape_jewelry <- function(url, category) {
  page <- read_html(url)

  product_titles <- page %>%
    html_nodes("h2.a-size-mini") %>%
    html_text(trim = TRUE)

  price <- page %>%
    html_nodes(".a-price .a-offscreen") %>%
    html_text(trim = TRUE)

  ratings <- page %>%
    html_nodes("span.a-icon-alt") %>%
    html_text(trim = TRUE)

  review <- page %>%
    html_nodes("span.a-size-base.review-text") %>%
    html_text(trim = TRUE)

  data.frame(
    Product_titles = product_titles[1:30],
    Price = price[1:30],
    Ratings = ratings[1:30],
    Review = review[1:30]
  )
}

jewelry_products <- scrape_jewelry(jewelry_url, "Jewelry")
head(jewelry_products, 30)
```

```
##     Product_titles   Price            Ratings Review
## 1     Kendra Scott  $43.00 4.6 out of 5 stars   <NA>
## 2          Yesteel  $55.00 4.4 out of 5 stars   <NA>
## 3          DEARMAY  $15.99 4.4 out of 5 stars   <NA>
## 4          ALLHOLA  $16.99 4.5 out of 5 stars   <NA>
## 5     Kendra Scott  $16.99 4.6 out of 5 stars   <NA>
## 6           NONNYL   $2.83 4.6 out of 5 stars   <NA>
## 7        Swarovski  $19.99 4.5 out of 5 stars   <NA>
## 8  HOPE LOVE SHINE  $14.99 4.6 out of 5 stars   <NA>
## 9          FAXHION  $40.00 4.4 out of 5 stars   <NA>
## 10       Swarovski  $50.00 4.6 out of 5 stars   <NA>
## 11       Swarovski $198.97 4.6 out of 5 stars   <NA>
## 12      P3 POMPEII3 $199.99 4.3 out of 5 stars   <NA>
## 13       Swarovski $300.00 4.6 out of 5 stars   <NA>
## 14            IFKM  $36.99 3.8 out of 5 stars   <NA>
## 15          Rotnso  $25.99 4.5 out of 5 stars   <NA>
## 16       Swarovski  $72.97 4.5 out of 5 stars   <NA>
## 17           wgoud $129.00 4.4 out of 5 stars   <NA>
## 18           Eiito  $77.54 4.3 out of 5 stars   <NA>
## 19          S.Leaf $119.00 4.6 out of 5 stars   <NA>
## 20     LADY COLOUR $299.99 4.4 out of 5 stars   <NA>
## 21    Kendra Scott  $87.11 4.6 out of 5 stars   <NA>
## 22       Swarovski $149.00 4.5 out of 5 stars   <NA>
## 23           Wssxc  $22.97 4.5 out of 5 stars   <NA>
## 24         PANDORA  $12.99 4.6 out of 5 stars   <NA>
## 25       Miabella  $19.99 4.5 out of 5 stars   <NA>
## 26          Barzel $112.00 4.6 out of 5 stars   <NA>
## 27          Tewiky $189.00 4.4 out of 5 stars   <NA>
## 28          Jstyle  $13.99 4.4 out of 5 stars   <NA>
## 29          Poxtex  $14.79 4.3 out of 5 stars   <NA>
## 30          YADOCA   $5.95 4.4 out of 5 stars   <NA>
```

Girls_Clothing Category

```
girls_clothing_url <- 'https://www.amazon.com/s?i=specialty-aps&bbn=16225020011&rh=n%3A7141123011%2Cn%3A

scrape_girls_clothing <- function(url, category) {
  page <- read_html(url)

  product_titles <- page %>%
    html_nodes("h2.a-size-mini") %>%
    html_text(trim = TRUE)
```

```r
  price <- page %>%
    html_nodes(".a-price .a-offscreen") %>%
    html_text(trim = TRUE)

  ratings <- page %>%
    html_nodes("span.a-icon-alt") %>%
    html_text(trim = TRUE)

  review <- page %>%
    html_nodes("span.a-size-base.review-text") %>%
    html_text(trim = TRUE)

  data.frame(
    Product_titles = product_titles[1:30],
    Price = price[1:30],
    Ratings = ratings[1:30],
    Review = review[1:30]
  )
}

girls_clothing_products <- scrape_girls_clothing(girls_clothing_url, "Girls_Clothing")
head(girls_clothing_products, 30)
```

```
##    Product_titles  Price            Ratings Review
## 1            <NA> $16.99 4.7 out of 5 stars   <NA>
## 2            <NA>  $2.83 4.8 out of 5 stars   <NA>
## 3            <NA> $21.95 4.7 out of 5 stars   <NA>
## 4            <NA> $25.99 4.8 out of 5 stars   <NA>
## 5            <NA>  $9.99 4.8 out of 5 stars   <NA>
## 6            <NA> $13.00 4.8 out of 5 stars   <NA>
## 7            <NA> $11.98 4.3 out of 5 stars   <NA>
## 8            <NA> $12.99 4.6 out of 5 stars   <NA>
## 9            <NA> $24.99 4.7 out of 5 stars   <NA>
## 10           <NA> $15.99 4.6 out of 5 stars   <NA>
## 11           <NA>  $2.67 4.8 out of 5 stars   <NA>
## 12           <NA> $19.99 4.8 out of 5 stars   <NA>
## 13           <NA> $26.37 4.7 out of 5 stars   <NA>
## 14           <NA> $49.95 4.5 out of 5 stars   <NA>
## 15           <NA> $13.97 4.8 out of 5 stars   <NA>
## 16           <NA> $19.95 4.7 out of 5 stars   <NA>
## 17           <NA> $14.99 4.6 out of 5 stars   <NA>
## 18           <NA> $14.99 4.5 out of 5 stars   <NA>
## 19           <NA> $14.99 4.6 out of 5 stars   <NA>
## 20           <NA>  $2.50 4.7 out of 5 stars   <NA>
## 21           <NA> $16.99 4.7 out of 5 stars   <NA>
## 22           <NA> $15.99 4.7 out of 5 stars   <NA>
## 23           <NA>  $2.67 4.7 out of 5 stars   <NA>
## 24           <NA> $14.99 4.8 out of 5 stars   <NA>
## 25           <NA> $52.99 4.8 out of 5 stars   <NA>
## 26           <NA> $79.99 4.7 out of 5 stars   <NA>
## 27           <NA> $19.50 4.2 out of 5 stars   <NA>
## 28           <NA> $19.71 4.5 out of 5 stars   <NA>
## 29           <NA> $27.99 4.6 out of 5 stars   <NA>
## 30           <NA> $11.70 4.6 out of 5 stars   <NA>
```

BabyToys Category

```r
babytoys_url <- 'https://www.amazon.com/s?i=specialty-aps&bbn=16225005011&rh=n%3A%2116225005011%2Cn%3A1

scrape_babytoys <- function(url, category) {
  page <- read_html(url)

  product_titles <- page %>%
    html_nodes("h2.a-size-mini") %>%
    html_text(trim = TRUE)

  price <- page %>%
    html_nodes(".a-price .a-offscreen") %>%
    html_text(trim = TRUE)

  ratings <- page %>%
    html_nodes("span.a-icon-alt") %>%
    html_text(trim = TRUE)

  review <- page %>%
    html_nodes("span.a-size-base.review-text") %>%
    html_text(trim = TRUE)

  data.frame(
    Product_titles = product_titles[1:30],
    Price = price[1:30],
    Ratings = ratings[1:30],
    Review = review[1:30]
  )
}

babytoys_products <- scrape_babytoys(babytoys_url, "Babytoys")
head(babytoys_products, 30)
```

```
##    Product_titles  Price            Ratings Review
## 1            <NA>  $6.15 4.8 out of 5 stars   <NA>
## 2            <NA>  $9.39 4.6 out of 5 stars   <NA>
## 3            <NA>  $9.99 4.8 out of 5 stars   <NA>
## 4            <NA> $19.99 4.7 out of 5 stars   <NA>
## 5            <NA>  $7.49 4.6 out of 5 stars   <NA>
## 6            <NA> $12.99 4.2 out of 5 stars   <NA>
## 7            <NA> $15.99 4.7 out of 5 stars   <NA>
## 8            <NA> $17.97 4.8 out of 5 stars   <NA>
## 9            <NA> $24.99 4.7 out of 5 stars   <NA>
## 10           <NA> $19.99 4.8 out of 5 stars   <NA>
## 11           <NA> $35.69 4.4 out of 5 stars   <NA>
## 12           <NA>  $7.99 4.8 out of 5 stars   <NA>
## 13           <NA>  $9.99 4.8 out of 5 stars   <NA>
## 14           <NA> $19.54 4.6 out of 5 stars   <NA>
## 15           <NA> $24.99 4.8 out of 5 stars   <NA>
## 16           <NA>  $5.99 4.6 out of 5 stars   <NA>
## 17           <NA> $19.99 4.8 out of 5 stars   <NA>
## 18           <NA> $19.23 4.8 out of 5 stars   <NA>
## 19           <NA> $26.99 4.6 out of 5 stars   <NA>
## 20           <NA> $12.89 4.8 out of 5 stars   <NA>
## 21           <NA> $17.99 4.8 out of 5 stars   <NA>
## 22           <NA> $12.99 4.8 out of 5 stars   <NA>
## 23           <NA> $19.99 4.8 out of 5 stars   <NA>
## 24           <NA>  $7.99 4.8 out of 5 stars   <NA>
## 25           <NA>  $9.99       4 Stars & Up   <NA>
## 26           <NA>  $8.88               <NA>   <NA>
## 27           <NA> $12.22               <NA>   <NA>
## 28           <NA> $14.99               <NA>   <NA>
## 29           <NA> $14.99               <NA>   <NA>
## 30           <NA> $19.99               <NA>   <NA>
```

6. Describe the data you have extracted.

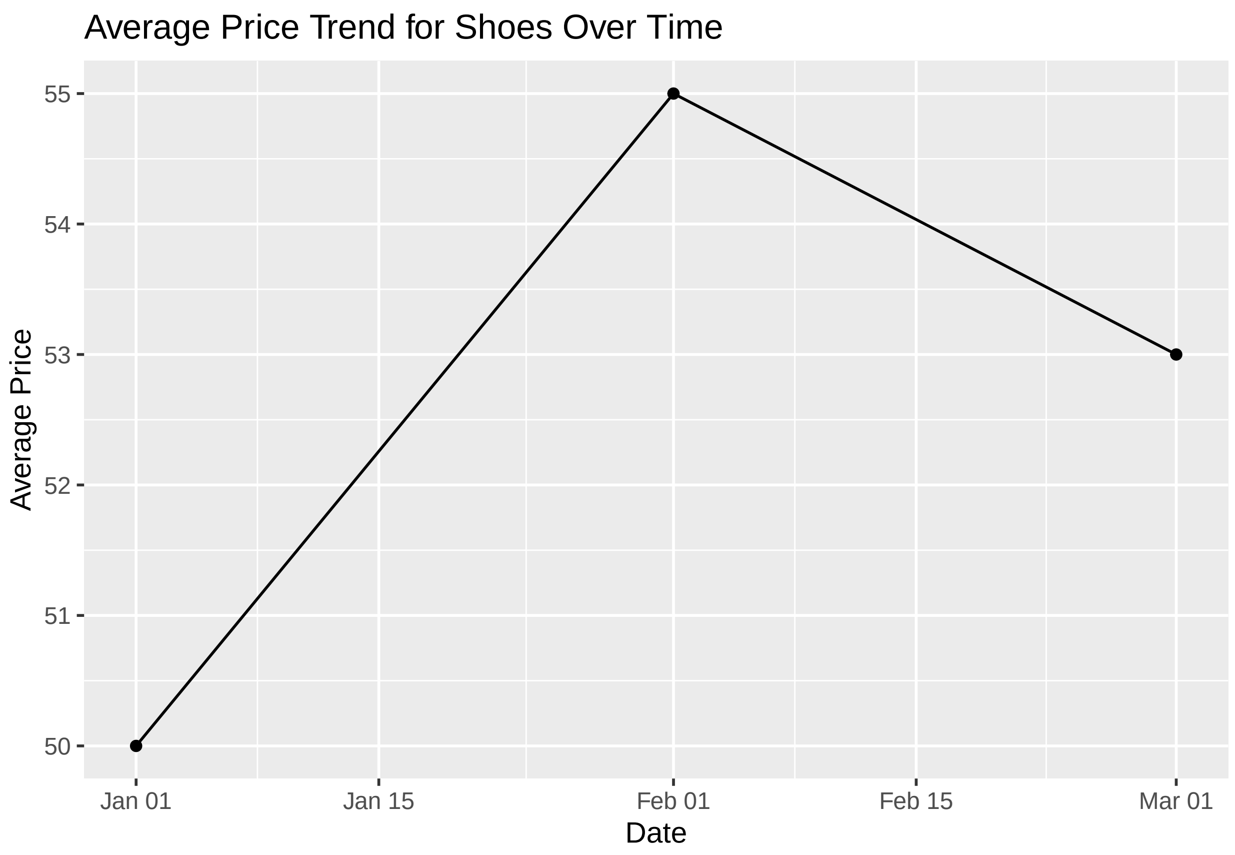
- We extracted 30 products for each category from amazon. The categories that we chose are shoes, makeups, jewelry, girls' clothing, and babytoys. For each categories, we extracted 30 product titles, price, rating and reviews that is shown inside a data frame.

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

- The use case for the data extacted is trend analysis.

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

```r
library(ggplot2)

shoes_data <- data.frame(Date = c('2024-01-01', '2024-02-01', '2024-03-01'),
                        Avg_Price = c(50, 55, 53))
shoes_data$Date <- as.Date(shoes_data$Date)

ggplot(shoes_data, aes(x = Date, y = Avg_Price)) +
  geom_line() +
  geom_point() +
  labs(title = "Average Price Trend for Shoes Over Time", x = "Date", y = "Average Price")
```



The Trend Analysis graph tracks changes in a key metric (e.g., average price) over time. The x-axis shows the time period (like months), while the y-axis shows the value of the metric. For example, a graph could show the average price of shoes increasing from $50 in January to $55 in February, then dropping to $53 in March. This helps identify patterns, like price fluctuations, and informs decisions on when prices might rise or fall, aiding in pricing strategies and market forecasting.

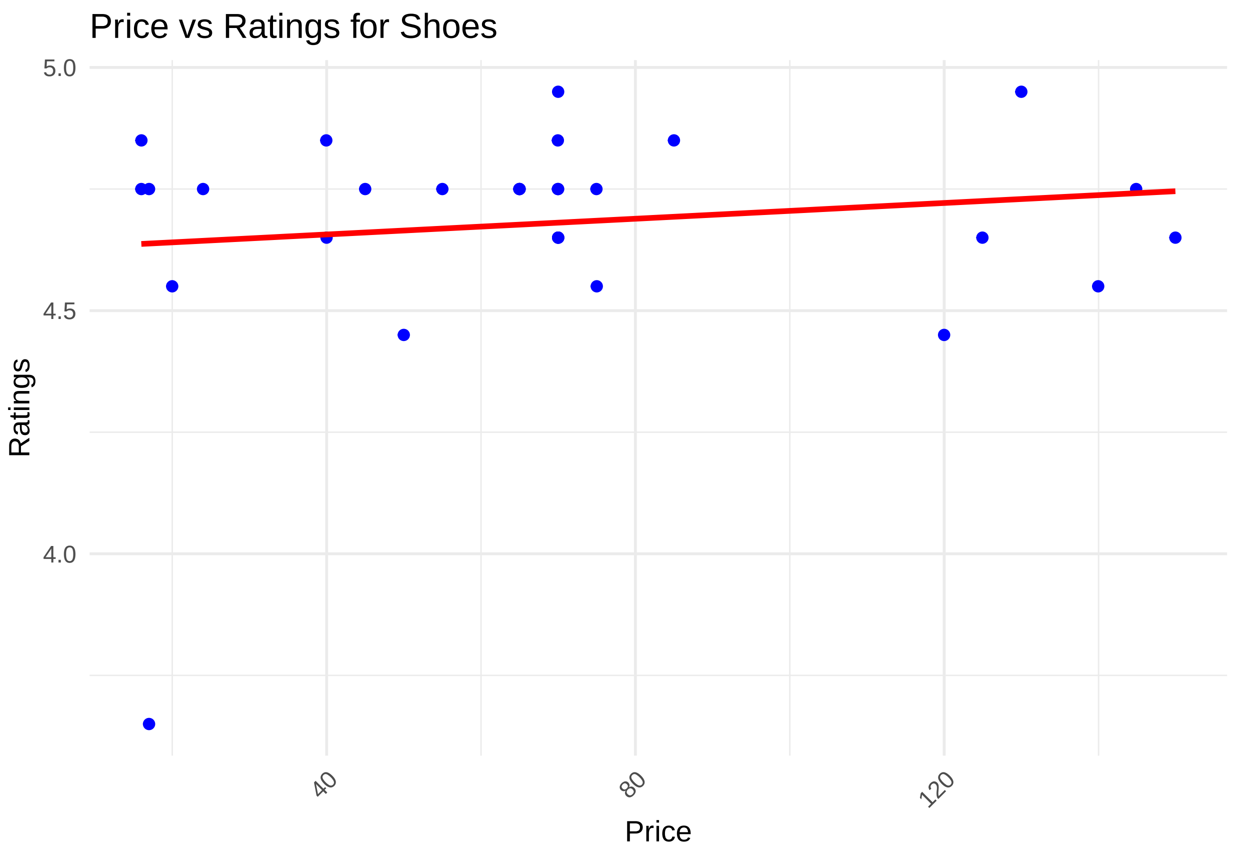
9. Graph the price and the ratings for each category. Use basic plotting functions and ggplot2 package.

Shoes Category

```
library(dplyr)
library(ggplot2)

shoes_products$Price <- as.numeric(gsub("[^0-9.]", "", shoes_products$Price))
shoes_products$Ratings <- as.numeric(gsub("[^0-9.]", "", shoes_products$Ratings))

shoes_price_ratings <- function(data, category_name) {
  ggplot(data, aes(x = Price, y = Ratings)) +
    geom_point(color = "blue") +
    geom_smooth(method = "lm", se = FALSE, color = "red") +
    theme_minimal() +
    labs(title = paste("Price vs Ratings for", category_name),
         x = "Price",
         y = "Ratings") +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
}
```

```r
shoes_price_ratings(shoes_products, "Shoes")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Makeup Category

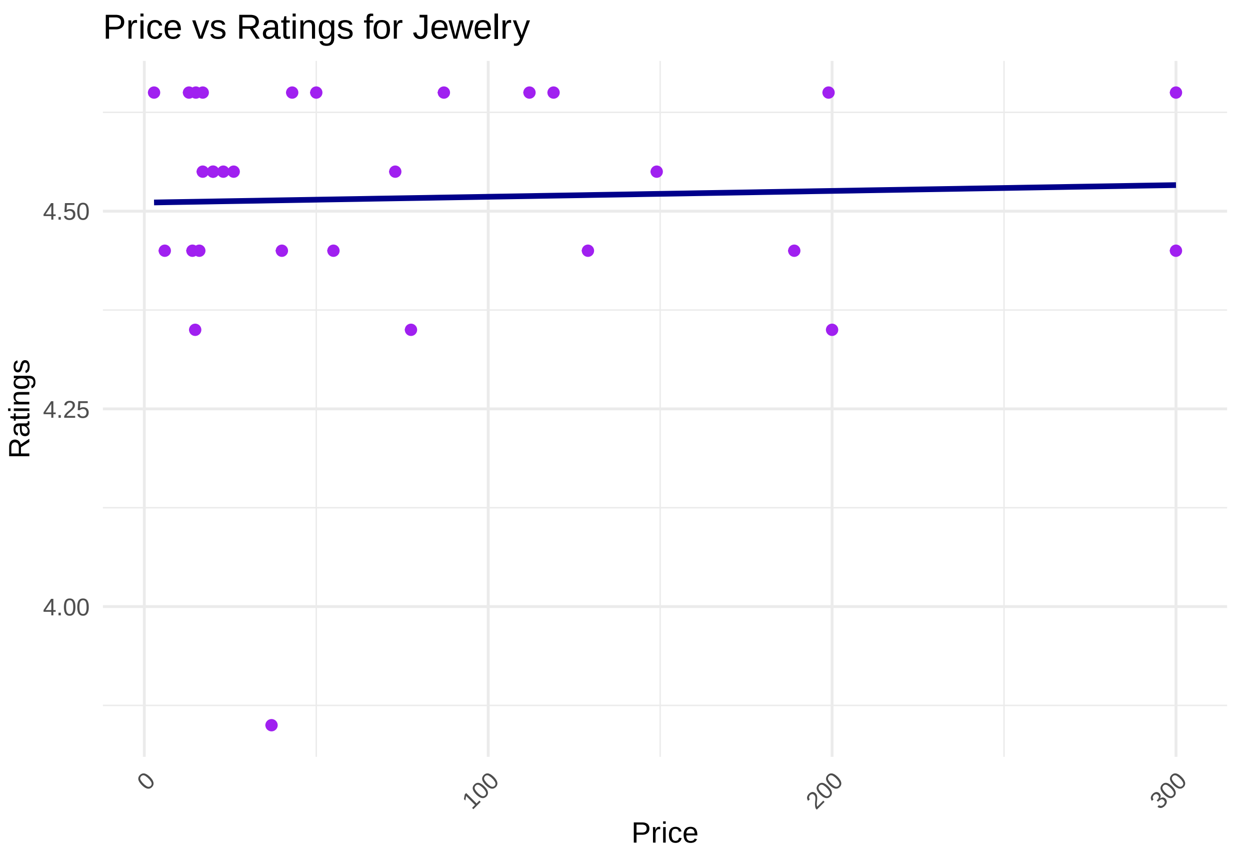
```r
library(dplyr)
library(ggplot2)

makeup_products$Price <- as.numeric(gsub("[^0-9.]", "", makeup_products$Price))
makeup_products$Ratings <- as.numeric(gsub("[^0-9.]", "", makeup_products$Ratings))

makeup_price_ratings <- function(data, category_name) {
  ggplot(data, aes(x = Price, y = Ratings)) +
    geom_point(color = "coral1") +
    geom_smooth(method = "lm", se = FALSE, color = "chocolate4") +
    theme_minimal() +
    labs(title = paste("Price vs Ratings for", category_name),
         x = "Price",
         y = "Ratings") +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
}

makeup_price_ratings(makeup_products, "Shoes")
```

```
## `geom_smooth()` using formula = 'y ~ x'
## Warning: Removed 30 rows containing non-finite outside the scale range
## (`stat_smooth()`).

## Warning: Removed 30 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

# Jewelry Category

```r
library(dplyr)
library(ggplot2)

jewelry_products$Price <- as.numeric(gsub("[^0-9.]", "", jewelry_products$Price))
jewelry_products$Ratings <- as.numeric(gsub("[^0-9.]", "", jewelry_products$Ratings))

jewelry_price_ratings <- function(data, category_name) {
  ggplot(data, aes(x = Price, y = Ratings)) +
    geom_point(color = "purple") +
    geom_smooth(method = "lm", se = FALSE, color = "darkblue") +
    theme_minimal() +
    labs(title = paste("Price vs Ratings for", category_name),
         x = "Price",
         y = "Ratings") +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
}

jewelry_price_ratings(jewelry_products, "Jewelry")
```

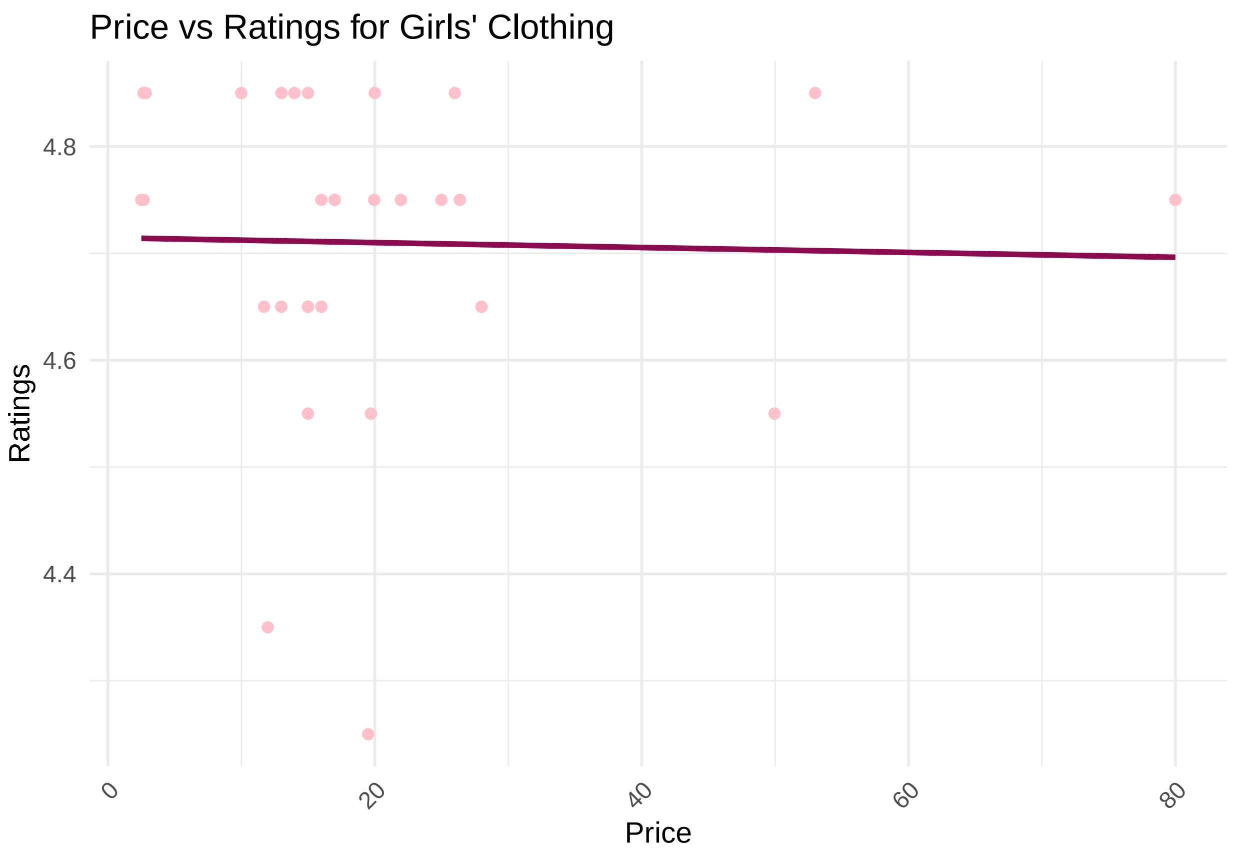
```
## `geom_smooth()` using formula = 'y ~ x'
```



Girl’s Clothing Category

```
library(dplyr)
library(ggplot2)

girls_clothing_products$Price <- as.numeric(gsub("[^0-9.]", "", girls_clothing_products$Price))
girls_clothing_products$Ratings <- as.numeric(gsub("[^0-9.]", "", girls_clothing_products$Ratings))

girls_clothing_price_ratings <- function(data, category_name) {
  ggplot(data, aes(x = Price, y = Ratings)) +
    geom_point(color = "pink") +
    geom_smooth(method = "lm", se = FALSE, color = "deeppink4") +
    theme_minimal() +
    labs(title = paste("Price vs Ratings for", category_name),
         x = "Price",
         y = "Ratings") +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
}

girls_clothing_price_ratings(girls_clothing_products, "Girls' Clothing")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



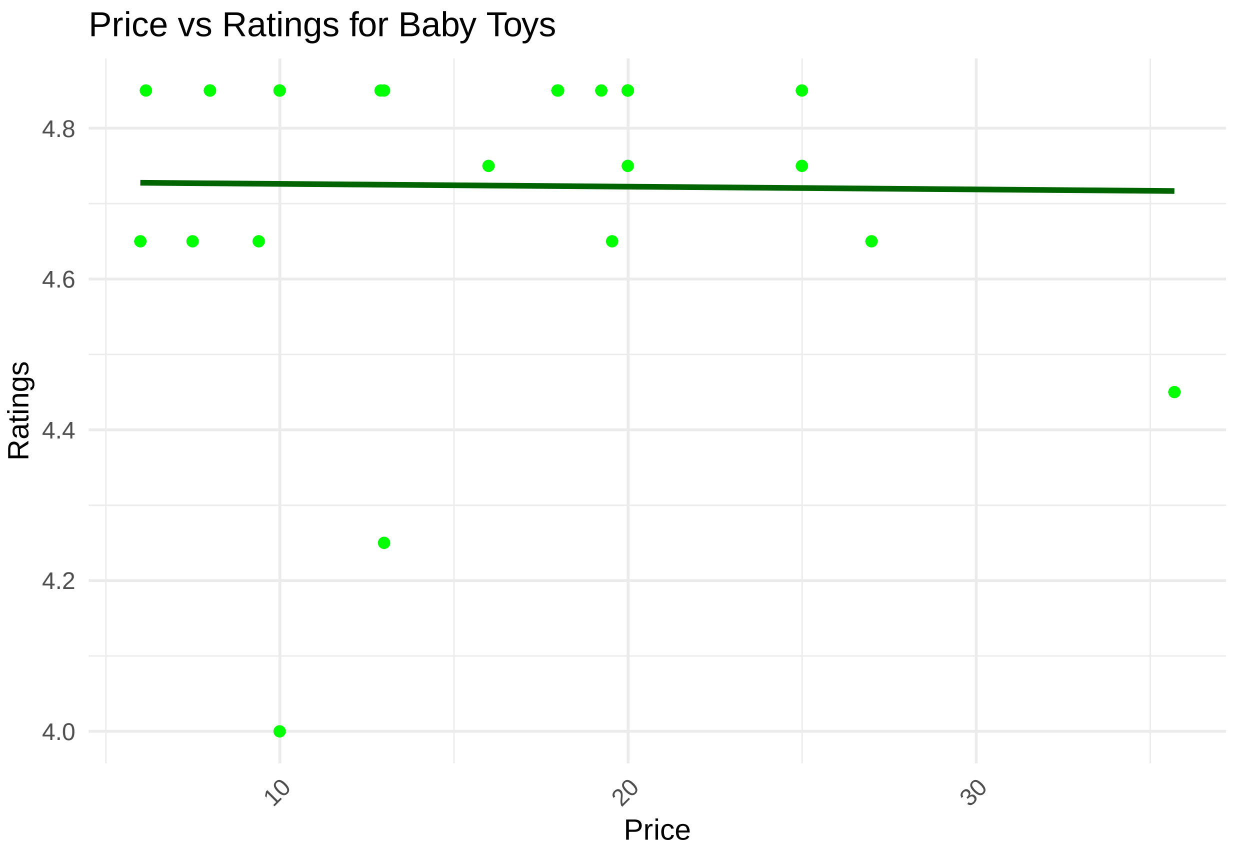
Baby Toys Category

```r
library(dplyr)
library(ggplot2)

babytoys_products$Price <- as.numeric(gsub("[^0-9.]", "", babytoys_products$Price))
babytoys_products$Ratings <- as.numeric(gsub("[^0-9.]", "", babytoys_products$Ratings))

babytoys_price_ratings <- function(data, category_name) {
  ggplot(data, aes(x = Price, y = Ratings)) +
    geom_point(color = "green") +
    geom_smooth(method = "lm", se = FALSE, color = "darkgreen") +
    theme_minimal() +
    labs(title = paste("Price vs Ratings for", category_name),
         x = "Price",
         y = "Ratings") +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
}

babytoys_price_ratings(babytoys_products, "Baby Toys")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 5 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 5 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



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

```
rank_products <- function(data, shoes_products) {
  data <- data %>%
    mutate(
      Price = as.numeric(gsub("[^0-9.]", "", Price)),
      Ratings = as.numeric(gsub("[^0-9.]", "", Ratings))
    ) %>%
    mutate(
      Rank_by_Price = rank(-Price, ties.method = "min"),
      Rank_by_Ratings = rank(-Ratings, ties.method = "min")
    ) %>%
    arrange(Rank_by_Price) %>%
    select(Product_titles, Price, Ratings, Rank_by_Price, Rank_by_Ratings) %>%
    head(10)
}

shoes_ranked <- rank_products(shoes_products, "Shoes")
print("Top 10 Shoes by Price and Ratings")
```

```
## [1] "Top 10 Shoes by Price and Ratings"
```

```
print(shoes_ranked)
```

```
##    Product_titles  Price Ratings Rank_by_Price Rank_by_Ratings
## 1          dubuto 149.95    4.65             1              19
## 2          Stelle 144.87    4.75             2               7
## 3          adidas 139.95    4.55            3             25
## 4             UGG 129.99    4.95            4              1
## 5             UGG 124.95    4.65            5             19
## 6             UGG 119.99    4.45            6             28
## 7        Hey Dude  84.99    4.85            7              3
## 8    Under Armour  74.99    4.55            8             25
## 9   Sesame Street  74.95    4.75            9              7
## 10        Nfinity  70.00    4.65           10             19
```

```
makeup_ranked <- rank_products(makeup_products, "Makeup")
print("Top 10 Makeup Products by Price and Ratings")
```

```
## [1] "Top 10 Makeup Products by Price and Ratings"
```

```
print(makeup_ranked)
```

```
##    Product_titles Price Ratings Rank_by_Price Rank_by_Ratings
## 1            <NA>    NA      NA             1               1
## 2            <NA>    NA      NA             2               2
## 3            <NA>    NA      NA             3               3
## 4            <NA>    NA      NA             4               4
## 5            <NA>    NA      NA             5               5
## 6            <NA>    NA      NA             6               6
## 7            <NA>    NA      NA             7               7
## 8            <NA>    NA      NA             8               8
## 9            <NA>    NA      NA             9               9
## 10           <NA>    NA      NA            10              10
```

```
jewelry_ranked <- rank_products(jewelry_products, "Jewelry")
print("Top 10 Jewelry Products by Price and Ratings")
```

```
## [1] "Top 10 Jewelry Products by Price and Ratings"
```

```
print(jewelry_ranked)
```

```
##    Product_titles  Price Ratings Rank_by_Price Rank_by_Ratings
## 1       Swarovski 300.00    4.65             1               1
## 2     LADY COLOUR 299.99    4.45             2              19
## 3     P3 POMPEII3 199.99    4.35             3              27
## 4       Swarovski 198.97    4.65             4               1
## 5          Tewiky 189.00    4.45             5              19
## 6       Swarovski 149.00    4.55             6              12
## 7           wgoud 129.00    4.45             7              19
## 8          S.Leaf 119.00    4.65             8               1
## 9          Barzel 112.00    4.65             9               1
## 10   Kendra Scott  87.11    4.65            10               1
```

```
girls_clothing_ranked <- rank_products(girls_clothing_products, "Girl's Clothing")
print("Top 10 Girl's Clothing Products by Price and Ratings")
```

```
## [1] "Top 10 Girl's Clothing Products by Price and Ratings"
```

```
print(girls_clothing_ranked)
```

```
##    Product_titles Price Ratings Rank_by_Price Rank_by_Ratings
## 1            <NA> 79.99    4.75             1              10
## 2            <NA> 52.99    4.85             2               1
## 3            <NA> 49.95    4.55             3              26
## 4             <NA> 27.99    4.65             4             20
## 5             <NA> 26.37    4.75             5             10
## 6             <NA> 25.99    4.85             6              1
## 7             <NA> 24.99    4.75             7             10
## 8             <NA> 21.95    4.75             8             10
## 9             <NA> 19.99    4.85             9              1
## 10            <NA> 19.95    4.75            10             10
```

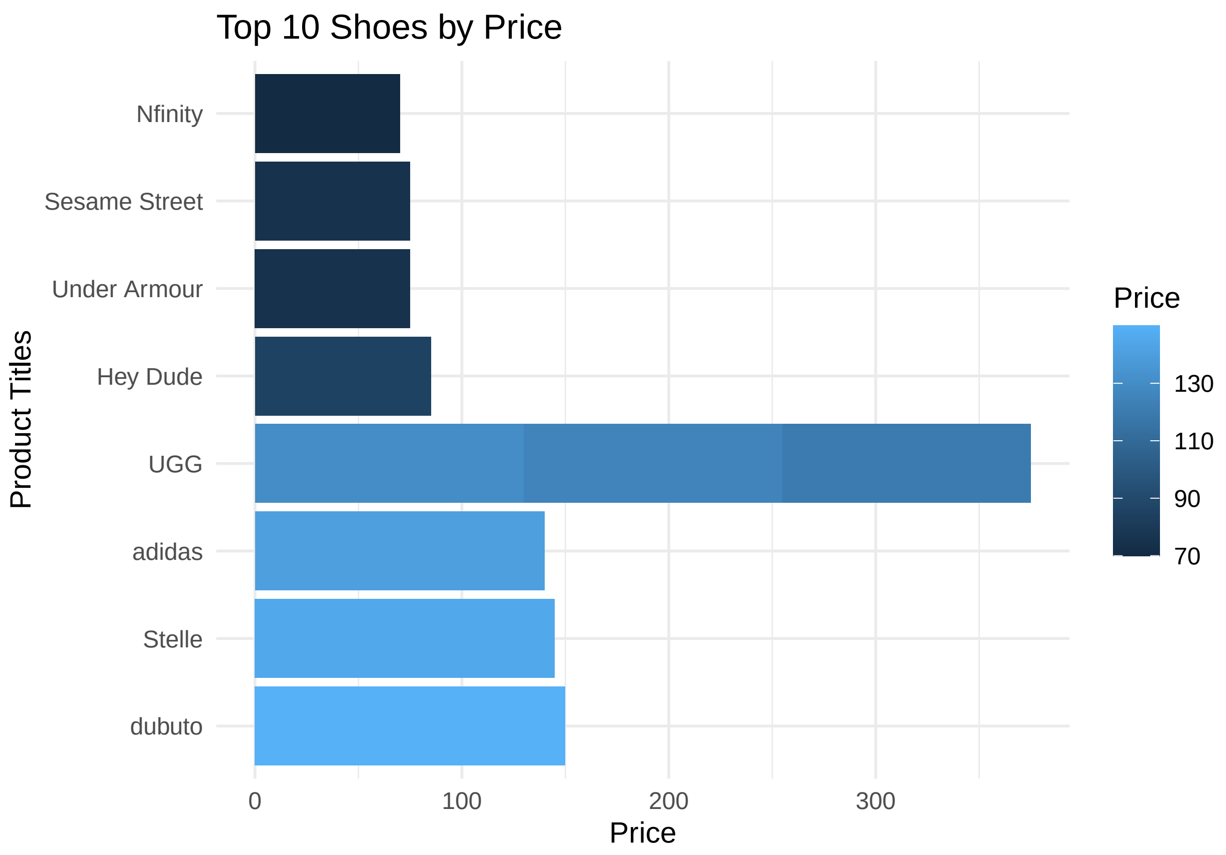
```
babytoys_ranked <- rank_products(babytoys_products, "Baby Toys")
print("Top 10 Baby Toys Products by Price and Ratings")
```

```
## [1] "Top 10 Baby Toys Products by Price and Ratings"
```

```
print(babytoys_ranked)
```

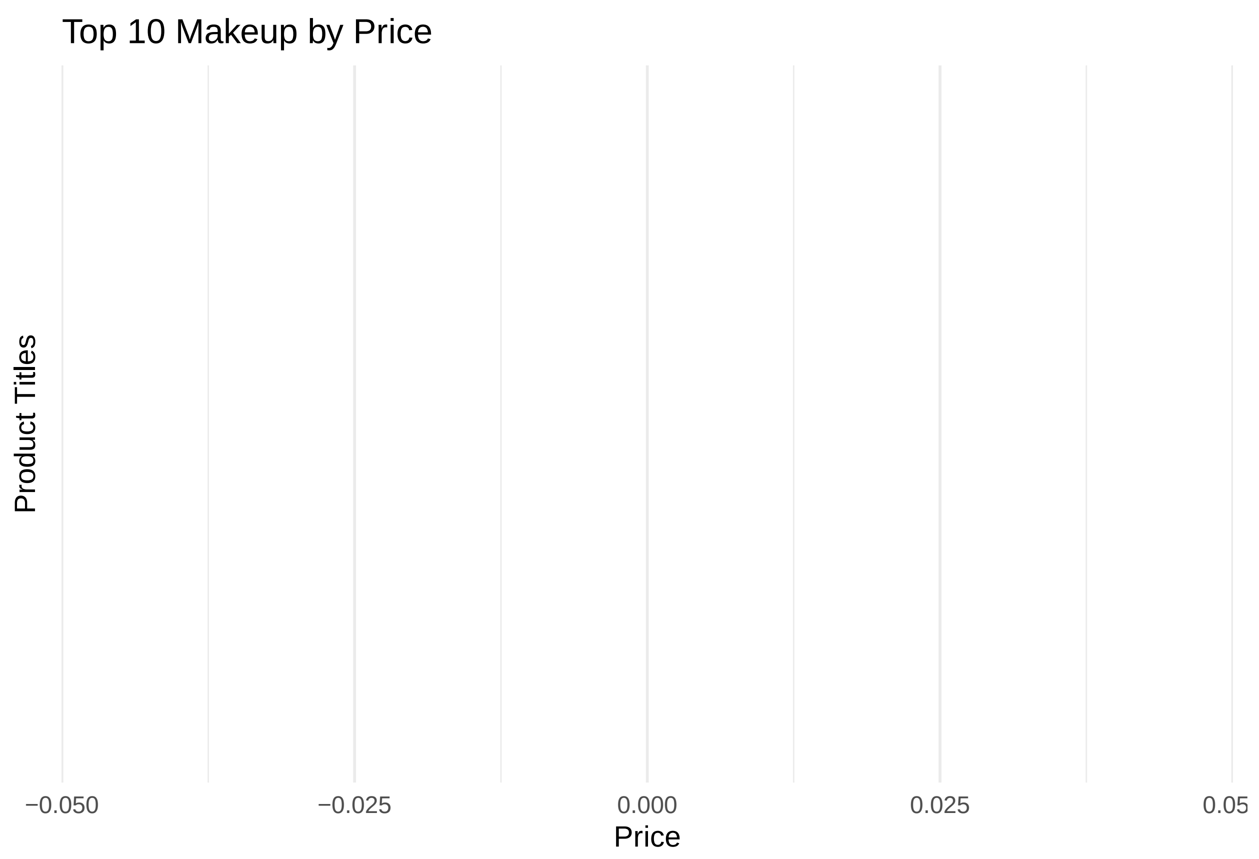
```
##    Product_titles Price Ratings Rank_by_Price Rank_by_Ratings
## 1            <NA> 35.69    4.45             1             23
## 2            <NA> 26.99    4.65             2             18
## 3            <NA> 24.99    4.75             3             15
## 4            <NA> 24.99    4.85             3              1
## 5            <NA> 19.99    4.75             5             15
## 6            <NA> 19.99    4.85             5              1
## 7            <NA> 19.99    4.85             5              1
## 8            <NA> 19.99    4.85             5              1
## 9            <NA> 19.99      NA             5             30
## 10           <NA> 19.54    4.65            10             18
```

```
# Shoes - Price Ranking
plot_rankings <- function(data, shoes_ranked, rank_col, value_col, value_name) {
  ggplot(data, aes(x = reorder(Product_titles, -!!sym(value_col)), y = !!sym(value_col), fill = !!sym(va
    geom_bar(stat = "identity") +
    coord_flip() +
    labs(
      title = paste("Top 10", shoes_ranked, "by", value_name),
      x = "Product Titles",
      y = value_name
    ) +
    theme_minimal()
}

plot_rankings(shoes_ranked, "Shoes", "Rank_by_Price", "Price", "Price")
```

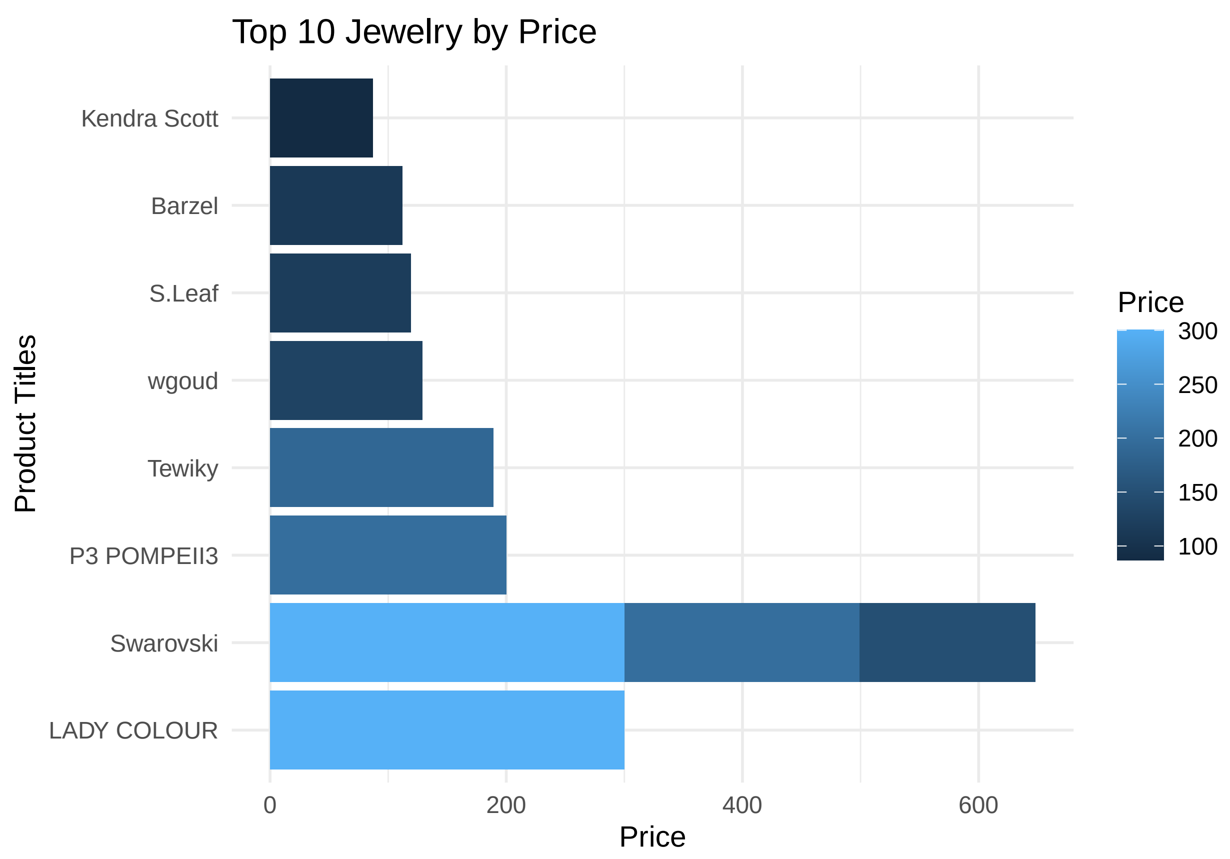


```r
# Makeup - Price Ranking
plot_rankings <- function(data, makeup_ranked, rank_col, value_col, value_name) {
  ggplot(data, aes(x = reorder(Product_titles, -!!sym(value_col)), y = !!sym(value_col), fill = !!sym(va
    geom_bar(stat = "identity") +
    coord_flip() +
    labs(
      title = paste("Top 10", makeup_ranked, "by", value_name),
      x = "Product Titles",
      y = value_name
    ) +
    theme_minimal()
}

plot_rankings(makeup_ranked, "Makeup", "Rank_by_Price", "Price", "Price")
```

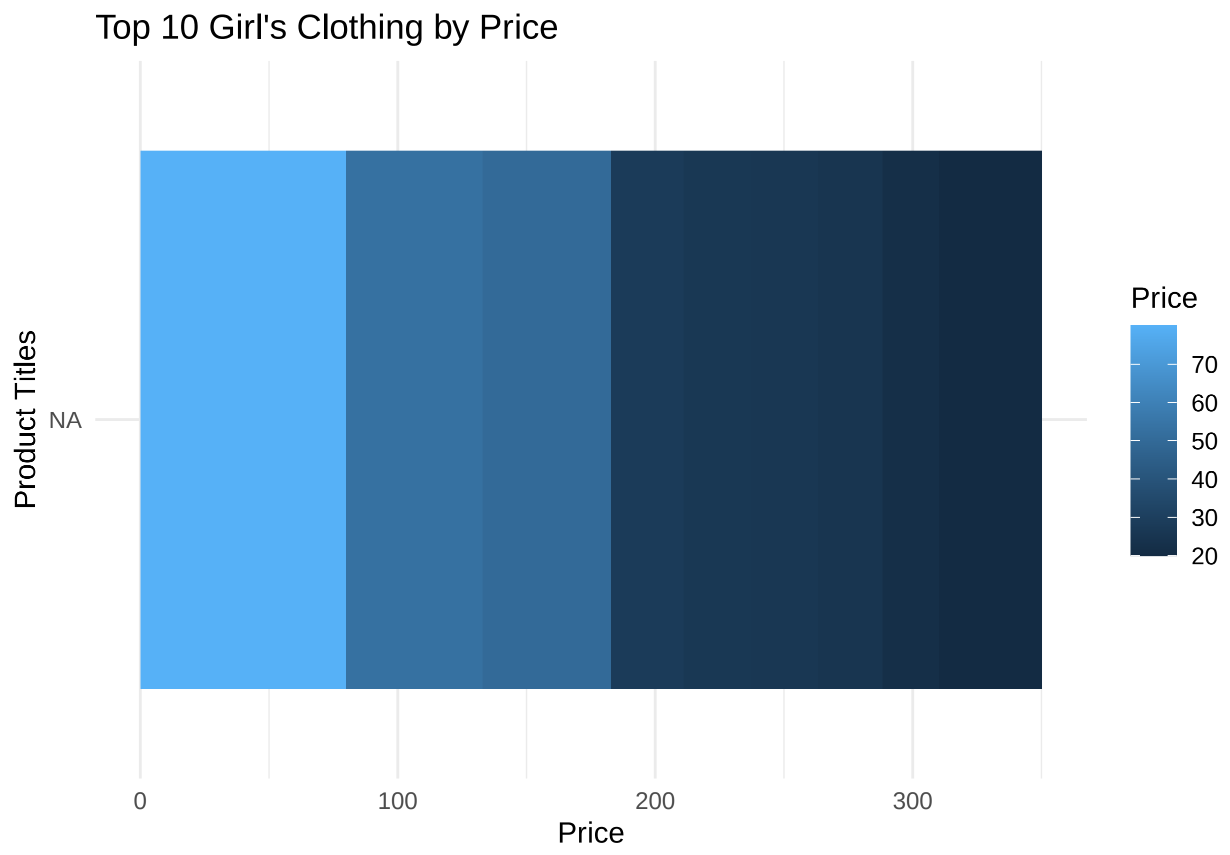
```
## Warning: Removed 10 rows containing missing values or values outside the scale range
## (`geom_bar()`).
```



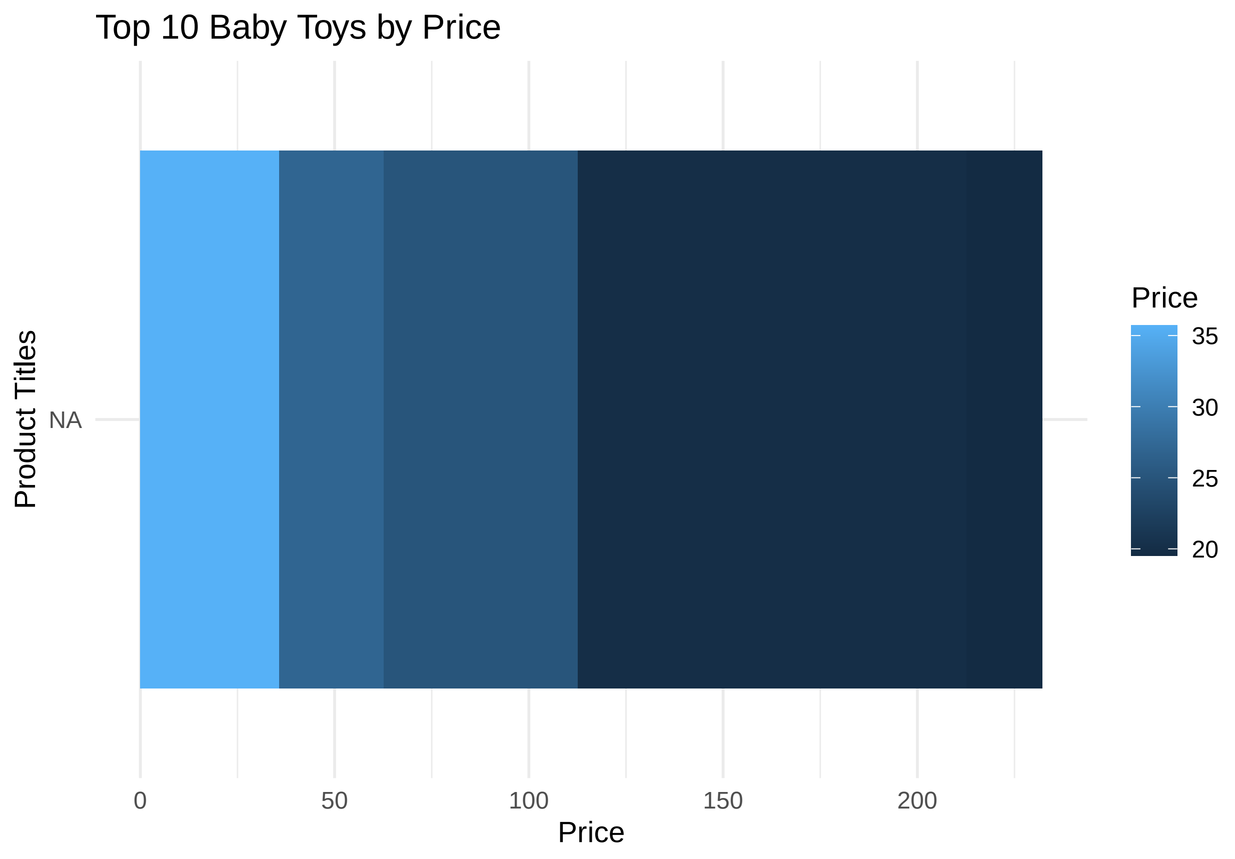
```r
# Jewelry - Price Ranking
plot_rankings <- function(data, jewelry_ranked, rank_col, value_col, value_name) {
  ggplot(data, aes(x = reorder(Product_titles, -!!sym(value_col)), y = !!sym(value_col), fill = !!sym(va
    geom_bar(stat = "identity") +
    coord_flip() +
    labs(
      title = paste("Top 10", jewelry_ranked, "by", value_name),
      x = "Product Titles",
      y = value_name
    ) +
    theme_minimal()
}

plot_rankings(jewelry_ranked, "Jewelry", "Rank_by_Price", "Price", "Price")
```



```r
# Girl's Clothing - Price Ranking
plot_rankings <- function(data, girls_clothing_ranked, rank_col, value_col, value_name) {
  ggplot(data, aes(x = reorder(Product_titles, -!!sym(value_col)), y = !!sym(value_col), fill = !!sym(va
    geom_bar(stat = "identity") +
    coord_flip() +
    labs(
      title = paste("Top 10", girls_clothing_ranked, "by", value_name),
      x = "Product Titles",
      y = value_name
    ) +
    theme_minimal()
}

plot_rankings(girls_clothing_ranked, "Girl's Clothing", "Rank_by_Price", "Price", "Price")
```



```r
# Baby Toys - Price Ranking
plot_rankings <- function(data, babytoys_ranked, rank_col, value_col, value_name) {
  ggplot(data, aes(x = reorder(Product_titles, -!!sym(value_col)), y = !!sym(value_col), fill = !!sym(va
    geom_bar(stat = "identity") +
    coord_flip() +
    labs(
      title = paste("Top 10", babytoys_ranked, "by", value_name),
      x = "Product Titles",
      y = value_name
    ) +
    theme_minimal()
}

plot_rankings(babytoys_ranked, "Baby Toys", "Rank_by_Price", "Price", "Price")
```


Top 10 Baby Toys by Price
Product Titles
NA
0
50
100
150
200
Price
Price
35
30
25
20