IMDB Movie Data

Retrieving data through web scraping in R

Emmanuel Messori

13/09/2021

Objectives

We want to extract data from the top 50 movies on IMDB between September 2020 and 2021. There is a wealth of information we can pick from, but we will focus just on these fields:

- title
- description,
- genre,
- runtime
- ratings

Then, we want to check which relationship ratings have with the number of user votes. For instance, do the highest-rated movies also have the highest number of votes?

Libraries

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(rvest)
```

```
##
## Attaching package: 'rvest'

## The following object is masked from 'package:readr':
##
##     guess_encoding
```

To obtain the movies' list we made an advanced search on the IMDB website :

```
url <- "https://www.imdb.com/search/title/?title_type=feature&release_date=2020-09-01,2021-09-01"
```

We will now read the page content with the `read_html()` function of the `rvest` package:

```
content <- read_html(url)
```

Extracting the data

Using this Chrome extension we can easily identify a functional CSS selector for the elements of interest.

Titles

```
titles <- content %>%
    html_nodes(".lister-item-header a") %>%
    html_text()

head(titles)
```

```
## [1] "Shang-Chi et la Légende des Dix Anneaux"
## [2] "Il est trop bien"
## [3] "Cruella"
## [4] "Candyman"
## [5] "Free Guy"
## [6] "The Suicide Squad"
```

Years

```
# we're using parse number to just read the year and not the quarter info
years <- content %>%
    html_nodes(".text-muted.unbold") %>%
    html_text() %>%
    readr::parse_number()

unique(years)
```

```
## [1] 2021 2020
```

Movie's runtimes and genres

```
# again parsing as a numeric vector
runtimes <- content %>%
    html_nodes(".runtime") %>%
    html_text() %>%
    readr::parse_number()

head(runtimes)
```

```
## [1] 132  88 134  91 115 132
```

```
genres <- content %>%
    html_nodes(".genre") %>%
    html_text() %>%
    stringr::str_squish()

head(genres)
```

```
## [1] "Action, Adventure, Fantasy" "Comedy, Romance"
## [3] "Comedy, Crime, Drama"       "Horror, Thriller"
## [5] "Action, Adventure, Comedy"  "Action, Adventure, Comedy"
```

User ratings and metascores

The ratings bar is a bit more complex than the other components. It contains: * average user rating, repeated two times * A 'rate this' element * The Metascore (missing for some movies)

We will just focus on the avg. user rating and the Metascore. We could just scrape the rating and the metascore individually but in that way we won't preserve the relationship (e.g. we wouldn't were the NA metascores are located)

```
ratings <- content %>%
    html_nodes(".ratings-bar") %>%
    html_text() %>%
    stringr::str_squish() %>%
    str_remove("Rate this 1 2 3 4 5 6 7 8 9 10") %>%
    str_split(" X ")


v_ratings <- 1:50
v_meta <- 1:50

for (i in seq_along(ratings)) {
    v_ratings[i] <- str_remove_all(ratings[[i]][1], "\\d\\.?\\d?/\\d\\d") %>%
        readr::parse_number()
    v_meta[i] <- readr::parse_number(ratings[[i]][2])
}


# 7 movies have missing metascores
```

Votes

```
n_votes <- content %>%
    html_nodes(".sort-num_votes-visible span:nth-child(2)") %>%
    html_text() %>%
    readr::parse_number()

head(n_votes)
```

```
## [1]  64822  15877 150147  14134  52624 194335
```

Description

```r
desc <- content %>%
    html_nodes(".ratings-bar+ .text-muted") %>%
    html_text() %>%
    stringr::str_squish()
```

Final dataframe

```r
IMDBtop50 <- tibble(title = titles, year = years, genre = genres, runtime = runtimes,
    rating = v_ratings, metascore = v_meta, votes = n_votes, description = desc)

slice_max(IMDBtop50[-8], n = 10, order_by = votes) %>%
    knitr::kable(caption = "Top 10 movies by number of votes")
```

Table 1: Top 10 movies by number of votes

title	year	genre	runtime	rating	metascore	votes
Zack Snyder's Justice League	2021	Action, Adventure, Fantasy	242	8.1	54	334570
Wonder Woman 1984	2020	Action, Adventure, Fantasy	151	5.4	60	229248
Black Widow	2021	Action, Adventure, Sci-Fi	134	6.8	67	211270
The Suicide Squad	2021	Action, Adventure, Comedy	132	7.4	72	194335
Godzilla vs Kong	2021	Action, Sci-Fi, Thriller	113	6.4	59	171469
The Tomorrow War	2021	Action, Adventure, Drama	138	6.6	45	160081
Nobody	2021	Action, Crime, Drama	92	7.4	64	155772
Cruella	2021	Comedy, Crime, Drama	134	7.4	59	150147
Un homme en colère	2021	Action, Crime, Thriller	119	7.2	57	109055
Luca	2021	Animation, Adventure, Comedy	95	7.5	71	105175

```r
slice_max(IMDBtop50[-8], n = 10, order_by = rating) %>%
    knitr::kable(caption = "Top 10 movies by rating")
```

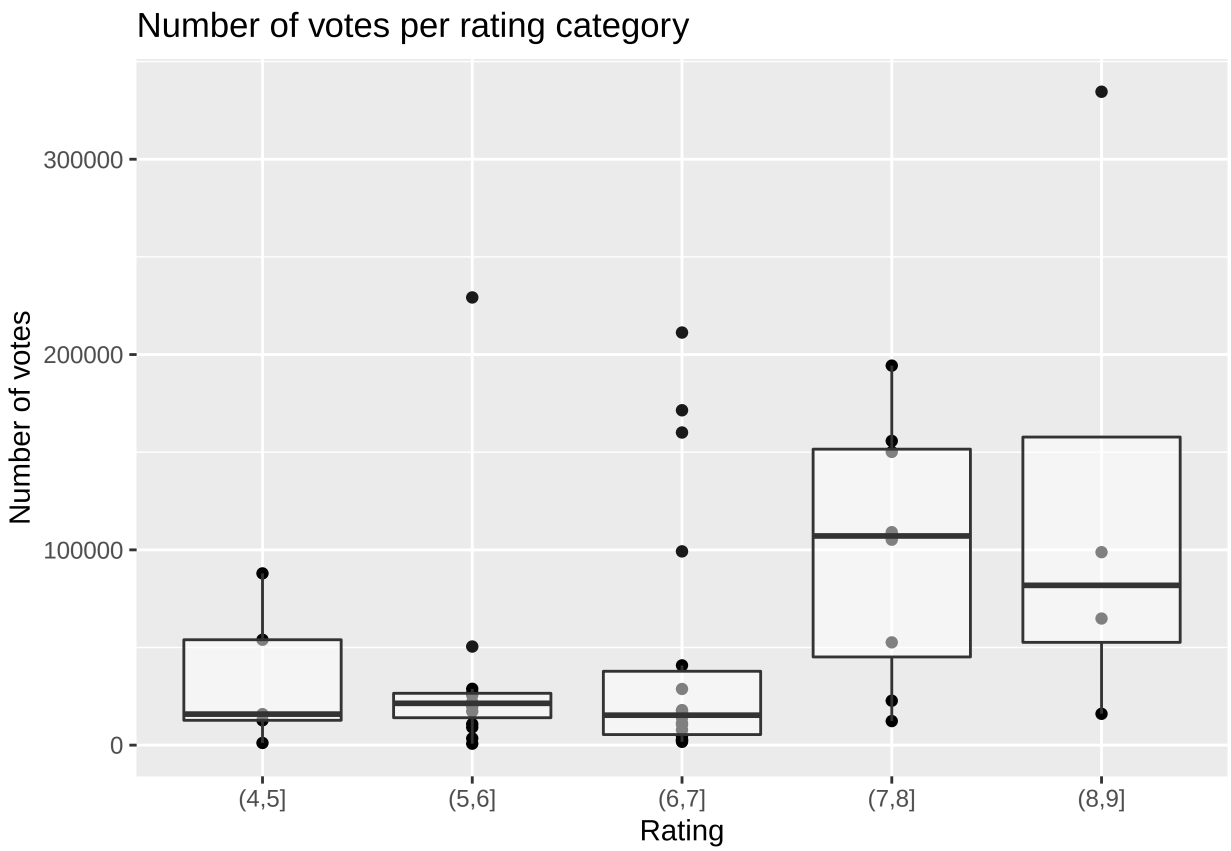
Table 2: Top 10 movies by rating

title	year	genre	runtime	rating	metascore	votes
Shershaah	2021	Action, Biography, Drama	135	8.8	NA	98818
CODA	2021	Drama, Music	111	8.1	75	16035
Zack Snyder's Justice League	2021	Action, Adventure, Fantasy	242	8.1	54	334570
Shang-Chi et la Légende des Dix Anneaux	2021	Action, Adventure, Fantasy	132	8.0	71	64822
Free Guy	2021	Action, Adventure, Comedy	115	7.6	62	52624
Luca	2021	Animation, Adventure, Comedy	95	7.5	71	105175
Cruella	2021	Comedy, Crime, Drama	134	7.4	59	150147

title	year	genre	runtime	rating	metascore	votes
The Suicide Squad	2021	Action, Adventure, Comedy	132	7.4	72	194335
The Witcher: Le cauchemar du Loup	2021	Animation, Action, Adventure	83	7.4	67	22740
Nobody	2021	Action, Crime, Drama	92	7.4	64	155772

Relation between number of votes and user rating

```
IMDBtop50 %>%
    ggplot(aes(cut(rating, 5, dig.lab = 1), votes)) + geom_point() + geom_boxplot(alpha = 0.5) +
    scale_y_continuous(labels = function(x) format(x, scientific = FALSE)) + labs(title = "Number of vo
    x = "Rating", y = "Number of votes")
```



It seems that the highest rated movies have also a notably higher number of votes. The two variables are slighty correlated:

```
cor.test(~rating + votes, data = IMDBtop50)
```

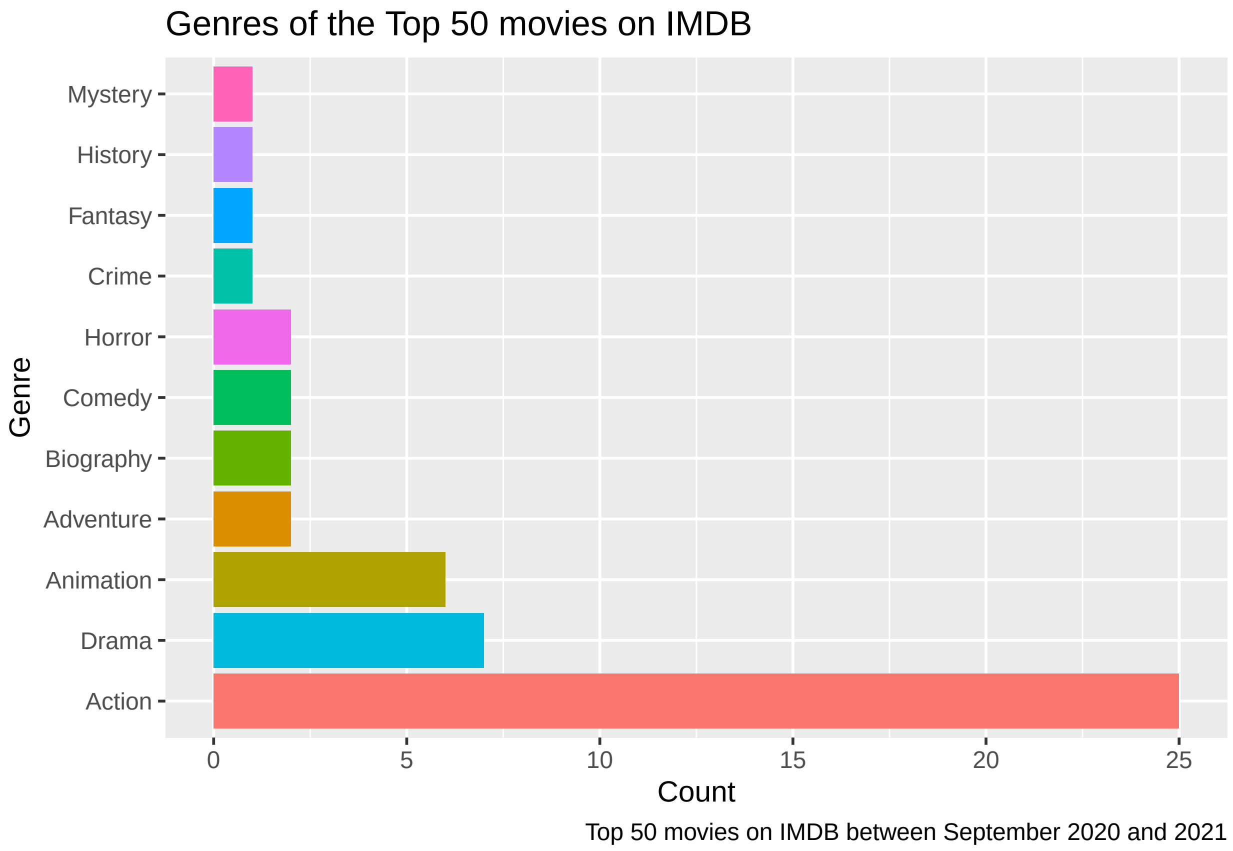
```
##
##  Pearson's product-moment correlation
##
## data:  rating and votes
## t = 2.6965, df = 48, p-value = 0.009635
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.09383099 0.58226857
## sample estimates:
##       cor
## 0.3627057
```

Relationship between genre and rating

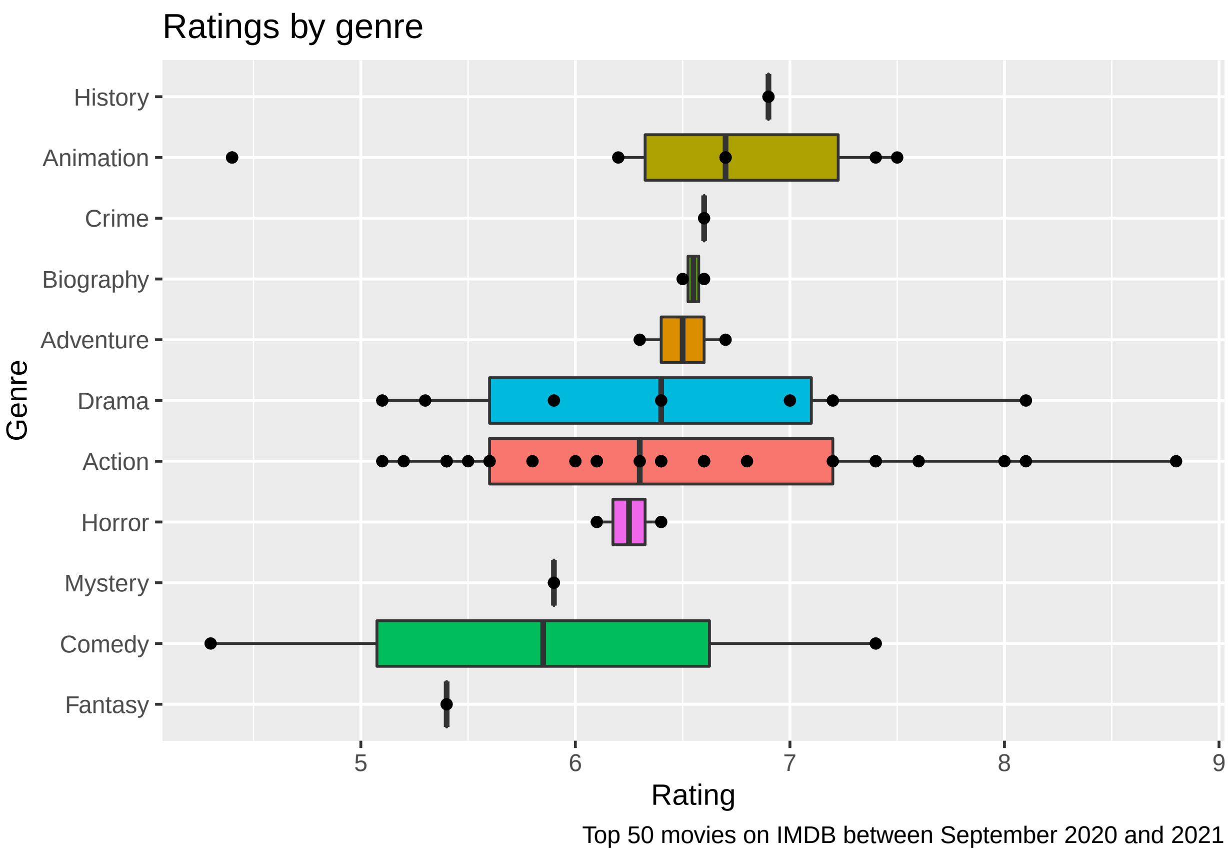
Before studying this relationship, we have to divide the genre column which is multivalued into four columns. Then we will choose for our purposes the main genre.

```
IMDBtop50_split <- separate(IMDBtop50, genre, into = c("g1", "g2", "g3", "g4")) %>%
    mutate(across(g1:g4, as.factor))

ggplot(IMDBtop50_split, aes(fct_infreq(g1), fill = g1)) + geom_bar(show.legend = FALSE) +
    coord_flip() + labs(title = "Genres of the Top 50 movies on IMDB", caption = "Top 50 movies on IMDB
    x = "Genre", y = "Count")
```



```
ggplot(IMDBtop50_split, aes(reorder(g1, rating, median), rating)) + geom_boxplot(aes(fill = g1),
    show.legend = FALSE) + geom_point() + coord_flip() + labs(title = "Ratings by genre",
    y = "Rating", x = "Genre", caption = "Top 50 movies on IMDB between September 2020 and 2021")
```



The genres groups are quite imbalanced with a strong predominance of the action genre. To study furthermore this relationship we could use an ANOVA after collecting more data (at present there are not enough individuals in certain groups and the homogeneity of variance is not respected).