IMDB - Exploratory Data Analysis

For this assignment, we are using the IMDB dataset. IMDB is one of the biggest movie rating websites in the world. The website stored almost every movie that has ever been released or that is on the planning to be released. The website stored more than 6 million titles with different types of information: movie director, movie cast, ratings, ratings, etc. IMDB is owned by Amazon since 1998.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

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

```
library(dlookr)
```

```
## Warning: package 'dlookr' was built under R version 4.0.5

##
## Attaching package: 'dlookr'

## The following object is masked from 'package:tidyr':
##
##     extract

## The following object is masked from 'package:base':
##
##     transform
```

```
IMDB <- read_csv("data/IMDB.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   color = col_character(),
##   director_name = col_character(),
##   actor_2_name = col_character(),
```

```
##   genres = col_character(),
##   actor_1_name = col_character(),
##   movie_title = col_character(),
##   actor_3_name = col_character(),
##   plot_keywords = col_character(),
##   movie_imdb_link = col_character(),
##   language = col_character(),
##   country = col_character(),
##   content_rating = col_character()
## )
## i Use `spec()` for the full column specifications.
```

```
str(IMDB)
```

```
## tibble [4,000 x 28] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ color                     : chr [1:4000] "Color" "Color" "Color" "Color" ...
##  $ director_name             : chr [1:4000] "Richard LaGravenese" "Malcolm D. Lee" "Danny Leiner" NA
##  $ num_critic_for_reviews    : num [1:4000] 131 81 106 9 144 139 51 97 157 99 ...
##  $ duration                  : num [1:4000] 126 86 83 60 89 98 88 94 115 120 ...
##  $ director_facebook_likes   : num [1:4000] 98 92 8 NA 406 235 6 32 0 11 ...
##  $ actor_3_facebook_likes    : num [1:4000] 37 466 612 460 759 229 854 453 35 NA ...
##  $ actor_2_name              : chr [1:4000] "Harry Connick Jr." "Eddie Griffin" "Mary Lynn Rajskub" "
##  $ actor_1_facebook_likes    : num [1:4000] 18000 744 3000 666 989 4000 2000 1000 85 16 ...
##  $ gross                     : num [1:4000] 53680848 38230435 46729374 NA 10824921 ...
##  $ genres                    : chr [1:4000] "Drama|Romance" "Action|Comedy" "Comedy|Mystery" "Action|
##  $ actor_1_name              : chr [1:4000] "Gerard Butler" "Dave Chappelle" "Jennifer Garner" "Nick
##  $ movie_title               : chr [1:4000] "P.S. I Love You<U+00A0>" "Undercover Brother<U+00A0>" "Du
##  $ num_voted_users           : num [1:4000] 167967 29661 116625 5817 200293 ...
##  $ cast_total_facebook_likes: num [1:4000] 18726 3001 6454 4043 3462 ...
##  $ actor_3_name              : chr [1:4000] "Nellie McKay" "Chi McBride" "Marla Sokoloff" "Jeff Marlo
##  $ facenumber_in_poster      : num [1:4000] 1 3 2 2 0 1 1 4 0 0 ...
##  $ plot_keywords             : chr [1:4000] "birthday|friendship|hairy chest|letter|widow" "african a
##  $ movie_imdb_link           : chr [1:4000] "http://www.imdb.com/title/tt0431308/?ref_=fn_tt_tt_1" "h
##  $ num_user_for_reviews      : num [1:4000] 243 179 469 25 621 385 54 71 361 230 ...
##  $ language                  : chr [1:4000] "English" "English" "English" "English" ...
##  $ country                   : chr [1:4000] "USA" "USA" "USA" "USA" ...
##  $ content_rating            : chr [1:4000] "PG-13" "PG-13" "PG-13" "TV-14" ...
##  $ budget                    : num [1:4000] 30000000 25000000 13000000 NA 10000000 10000000 70000000 1
##  $ title_year                : num [1:4000] 2007 2002 2000 NA 1999 ...
##  $ actor_2_facebook_likes    : num [1:4000] 631 489 934 579 939 464 1000 626 79 0 ...
##  $ imdb_score                : num [1:4000] 7.1 5.8 5.5 7.1 7.8 5.9 5.6 5.4 8.1 7.8 ...
##  $ aspect_ratio              : num [1:4000] 1.85 1.85 1.37 16 1.85 1.85 NA 1.85 1.85 1.37 ...
##  $ movie_facebook_likes      : num [1:4000] 20000 0 0 0 16000 0 5000 0 11000 3000 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   color = col_character(),
##   ..   director_name = col_character(),
##   ..   num_critic_for_reviews = col_double(),
##   ..   duration = col_double(),
##   ..   director_facebook_likes = col_double(),
##   ..   actor_3_facebook_likes = col_double(),
##   ..   actor_2_name = col_character(),
##   ..   actor_1_facebook_likes = col_double(),
##   ..   gross = col_double(),
```

```
##   ..   genres = col_character(),
##   ..   actor_1_name = col_character(),
##   ..   movie_title = col_character(),
##   ..   num_voted_users = col_double(),
##   ..   cast_total_facebook_likes = col_double(),
##   ..   actor_3_name = col_character(),
##   ..   facenumber_in_poster = col_double(),
##   ..   plot_keywords = col_character(),
##   ..   movie_imdb_link = col_character(),
##   ..   num_user_for_reviews = col_double(),
##   ..   language = col_character(),
##   ..   country = col_character(),
##   ..   content_rating = col_character(),
##   ..   budget = col_double(),
##   ..   title_year = col_double(),
##   ..   actor_2_facebook_likes = col_double(),
##   ..   imdb_score = col_double(),
##   ..   aspect_ratio = col_double(),
##   ..   movie_facebook_likes = col_double()
##   .. )
```

Missing values are frequently encountered within a big dataset. Before we analyse and visualise the data, the following script is applied for filtering out all the rows that contain at least one NA value. This to have a complete dataset that can be used for data analysis. Besides removing all the rows with NA, we will also remove the columns which we will not use for this analysis anyways.

```
IMDB <- IMDB %>% drop_na()
IMDB <- select(IMDB, -actor_2_facebook_likes, -plot_keywords, -movie_imdb_link, -actor_1_facebook_likes
IMDB[1:50,]
```

```
## # A tibble: 50 x 15
##    color director_name duration actor_2_name  gross genres actor_1_name
##    <chr> <chr>            <dbl> <chr>         <dbl> <chr>  <chr>
##  1 Color "Richard LaG~      126 Harry Conni~ 5.37e7 Drama~ Gerard Butl~
##  2 Color "Malcolm D. ~       86 Eddie Griff~ 3.82e7 Actio~ Dave Chappe~
##  3 Color "Danny Leine~       83 Mary Lynn R~ 4.67e7 Comed~ Jennifer Ga~
##  4 Color "Mike Judge"        89 Stephen Root 1.08e7 Comedy Gary Cole
##  5 Color "Peyton Reed"       98 Lindsay Slo~ 6.84e7 Comed~ Kirsten Dun~
##  6 Color "Brian Levan~       94 Amber Valle~ 2.43e7 Actio~ Madeline Ca~
##  7 Color "Alejandro G~      115 Jorge Salin~ 5.38e6 Drama~ Adriana Bar~
##  8 Color "David Ayer"       109 Chris Evans  2.64e7 Actio~ Keanu Reeves
##  9 Color "Ang Lee"          120 Pei-Pei Che~ 1.28e8 Actio~ Chen Chang
## 10 Color "Mark Herman"       94 Sheila Hanc~ 9.03e6 Drama~ Richard Joh~
## # ... with 40 more rows, and 8 more variables: movie_title <chr>,
## #   actor_3_name <chr>, language <chr>, country <chr>, content_rating <chr>,
## #   budget <dbl>, title_year <dbl>, imdb_score <dbl>
```

The top 10 movies based on the clean dataset is the following:

```
IMDB %>%
    select(movie_title, director_name, imdb_score, duration, genres) %>%
    top_n(10, imdb_score) %>%
    arrange(desc(imdb_score)) %>%
    head(10)
```

```
## # A tibble: 10 x 5
##    movie_title                director_name   imdb_score duration genres
##    <chr>                      <chr>                <dbl>    <dbl> <chr>
##  1 "The Godfather\xa0"        Francis Ford C~        9.2      175 Crime|Drama
##  2 "The Dark Knight\xa0"      Christopher No~        9        152 Action|Crime|~
##  3 "Pulp Fiction\xa0"         Quentin Tarant~        8.9      178 Crime|Drama
##  4 "Schindler's List\xa0"     Steven Spielbe~        8.9      185 Biography|Dra~
##  5 "The Lord of the Rings: T~ Peter Jackson          8.9      192 Action|Advent~
##  6 "The Lord of the Rings: T~ Peter Jackson          8.8      171 Action|Advent~
##  7 "Fight Club\xa0"           David Fincher          8.8      151 Drama
##  8 "Star Wars: Episode V - T~ Irvin Kershner         8.8      127 Action|Advent~
##  9 "Inception\xa0"            Christopher No~        8.8      148 Action|Advent~
## 10 "One Flew Over the Cuckoo~ Milos Forman           8.7      133 Drama
```

To compute the statistics of the numerical variables we are using summary():

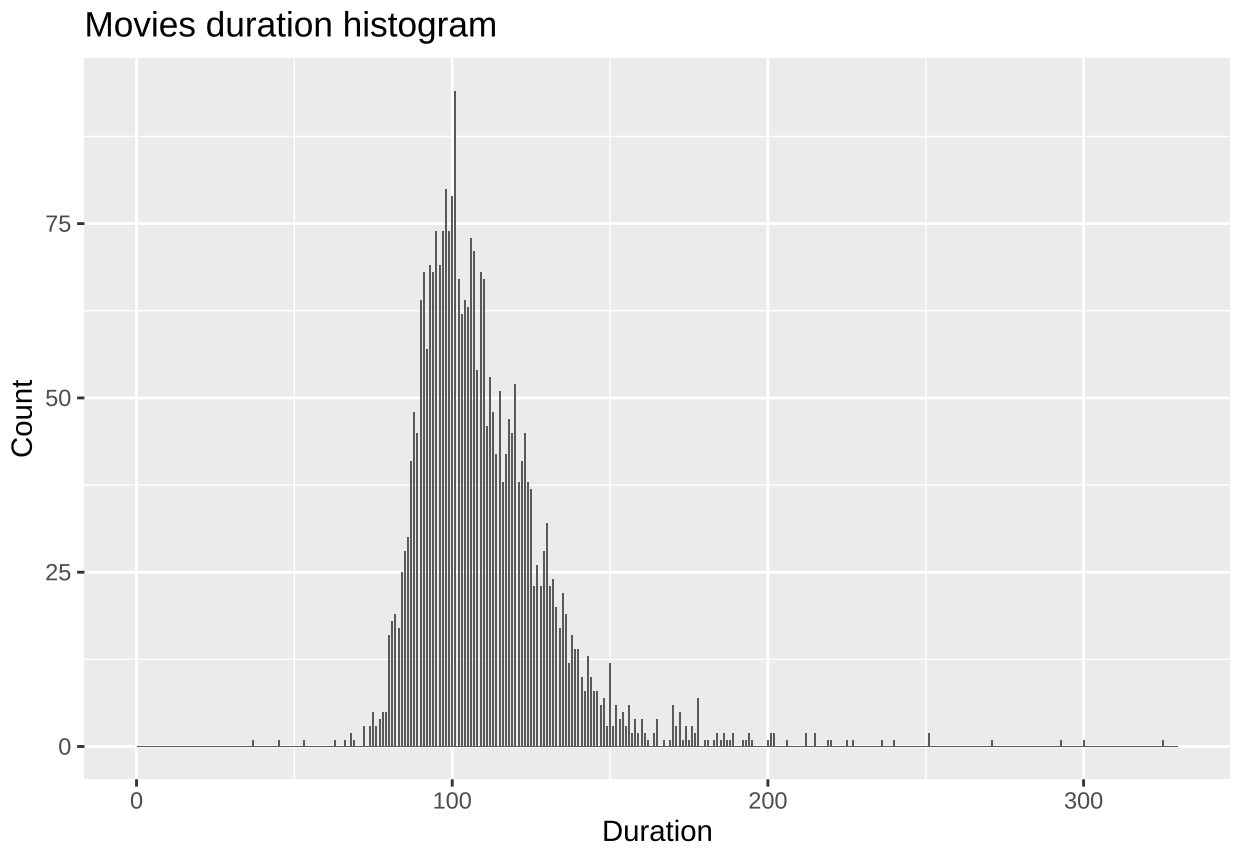
```
summary(IMDB)
```

```
##     color           director_name         duration     actor_2_name
##  Length:2964        Length:2964        Min.   : 37.0   Length:2964
##  Class :character   Class :character   1st Qu.: 96.0   Class :character
##  Mode  :character   Mode  :character   Median :106.0   Mode  :character
##                                        Mean   :110.2
##                                        3rd Qu.:120.0
##                                        Max.   :330.0
##      gross              genres          actor_1_name       movie_title
##  Min.   :      162   Length:2964        Length:2964        Length:2964
##  1st Qu.:  8740578   Class :character   Class :character   Class :character
##  Median : 30242898   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 52776132
##  3rd Qu.: 67271408
##  Max.   :760505847
##  actor_3_name         language           country          content_rating
##  Length:2964        Length:2964        Length:2964        Length:2964
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##      budget            title_year     imdb_score
##  Min.   :1.100e+03   Min.   :1927   Min.   :1.600
##  1st Qu.:1.000e+07   1st Qu.:1999   1st Qu.:5.900
##  Median :2.500e+07   Median :2004   Median :6.500
##  Mean   :4.379e+07   Mean   :2003   Mean   :6.452
##  3rd Qu.:5.000e+07   3rd Qu.:2010   3rd Qu.:7.200
##  Max.   :4.200e+09   Max.   :2016   Max.   :9.200
```

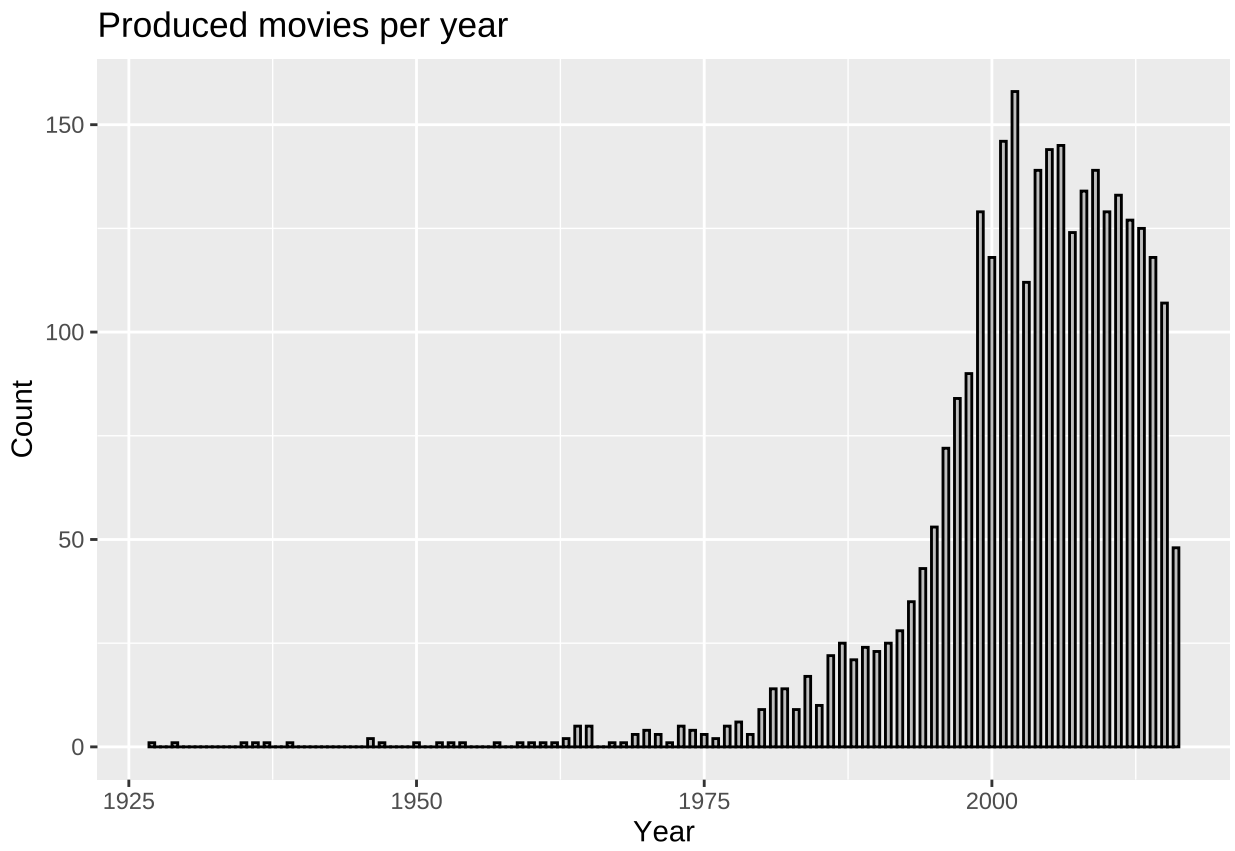
Individual histograms We are going to plot some histograms of the numerical variables from the IMDB dataset, which in this case are the Duration, Year, and the IMDB ratings.

```
ggplot(data = IMDB, aes(x = duration)) +
  geom_histogram(binwidth = 0.5) +
  xlim(c(0, quantile(IMDB$duration, 1))) +
  labs(title = "Movies duration histogram", x = "Duration", y = "Count")
```

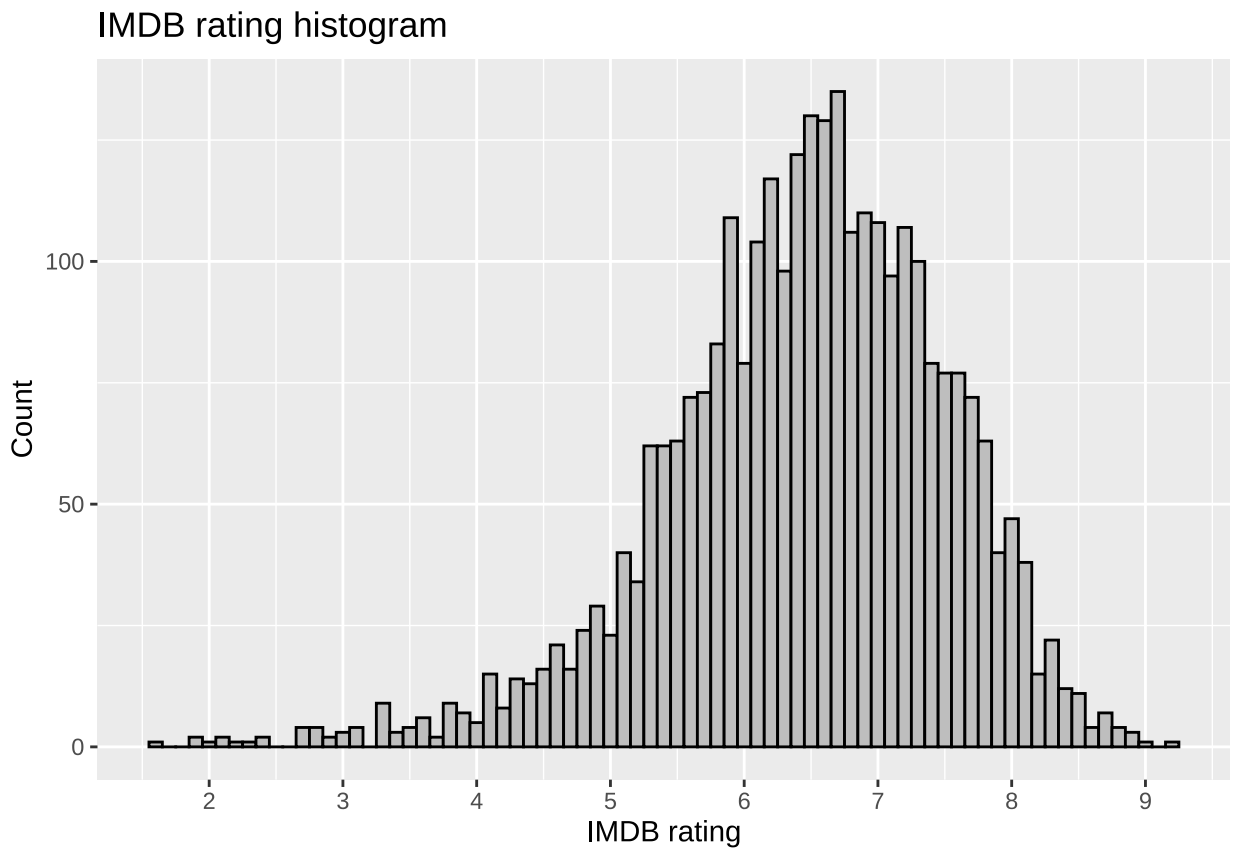
```
## Warning: Removed 2 rows containing missing values (geom_bar).
```




```
ggplot(data = IMDB, aes(x = title_year)) +
  geom_histogram(binwidth = 0.5, fill = "grey", color = "black") +
  labs(title = "Produced movies per year", x = "Year", y = "Count")
```




```
ggplot(data = IMDB, aes(x = imdb_score)) +
  geom_histogram(binwidth = 0.1, fill = "grey", color = "black") +
  labs(title = "IMDB rating histogram", x = "IMDB rating", y = "Count") +
  scale_x_continuous(breaks = seq(0,10,1))
```



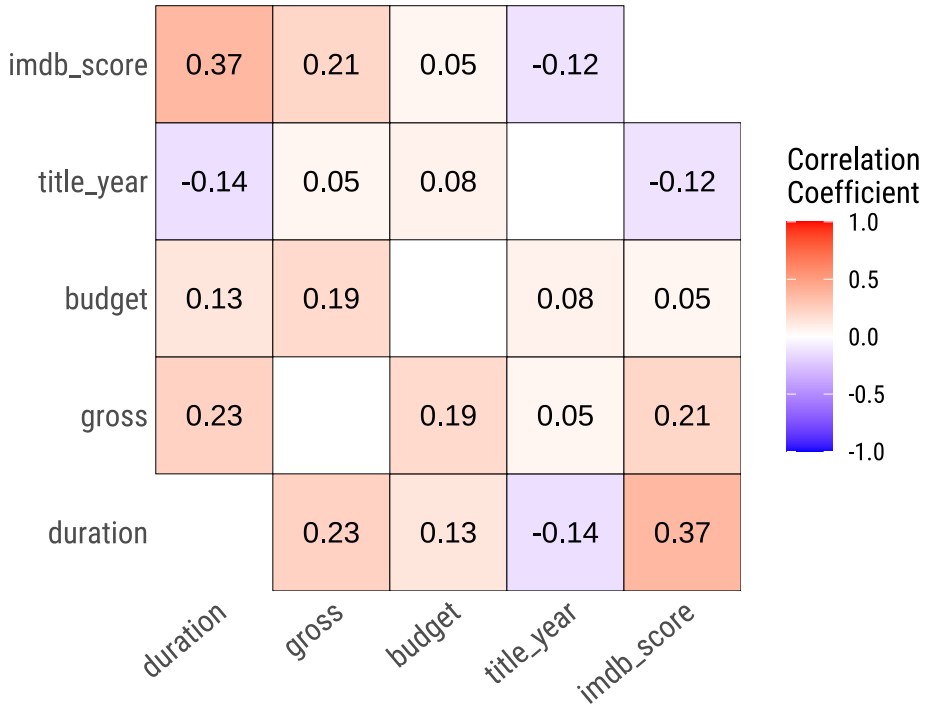
We can draw some conclusions from the individual histograms:

1. The duration and IMDB score both look normal distributed.
2. The distribution of the years has been growing exponentially throughout the years.
3. Most movies have an average duration of 110 minutes.
4. Most movies receive a rating of 6.5.

Univariate Exploratory Data Analysis

For the numerical values, we can plot a correlation matrix to see which variables are related to each other:

```
plot_correlate(IMDB)
```

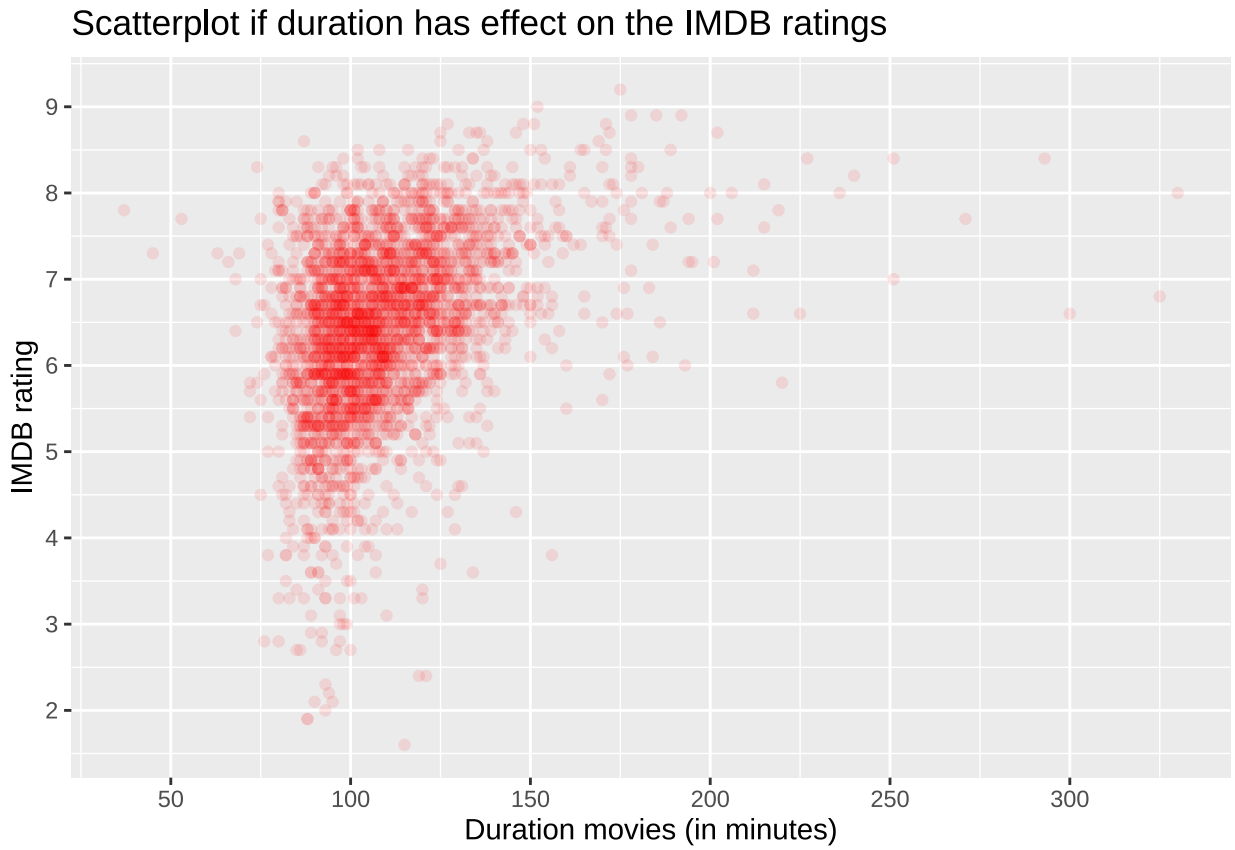


Based on the correlation matrix, we can notice the following items:

1. There is a relationship between the duration and the IMDB score.
2. There is no correlation between the duration and year of the movie that has been produced.
3. There is also no correlation the year and IMDB score of the movie.

To find out more about the positive correlation, we will see if movies with a longer or shorter duration have better ratings on IMDB.

```
IMDB %>%
    group_by(movie_title) %>%
    unique() %>%
    ggplot(aes(duration, imdb_score)) +
    geom_point(alpha = 0.1, color="Red") +
    scale_y_continuous(breaks=seq(0,10,1)) +
    scale_x_continuous(breaks=seq(0,300,50)) +
        labs(title = "Scatterplot if duration has effect on the IMDB ratings", x = "Duration movies (in
             y = "IMDB rating")
```

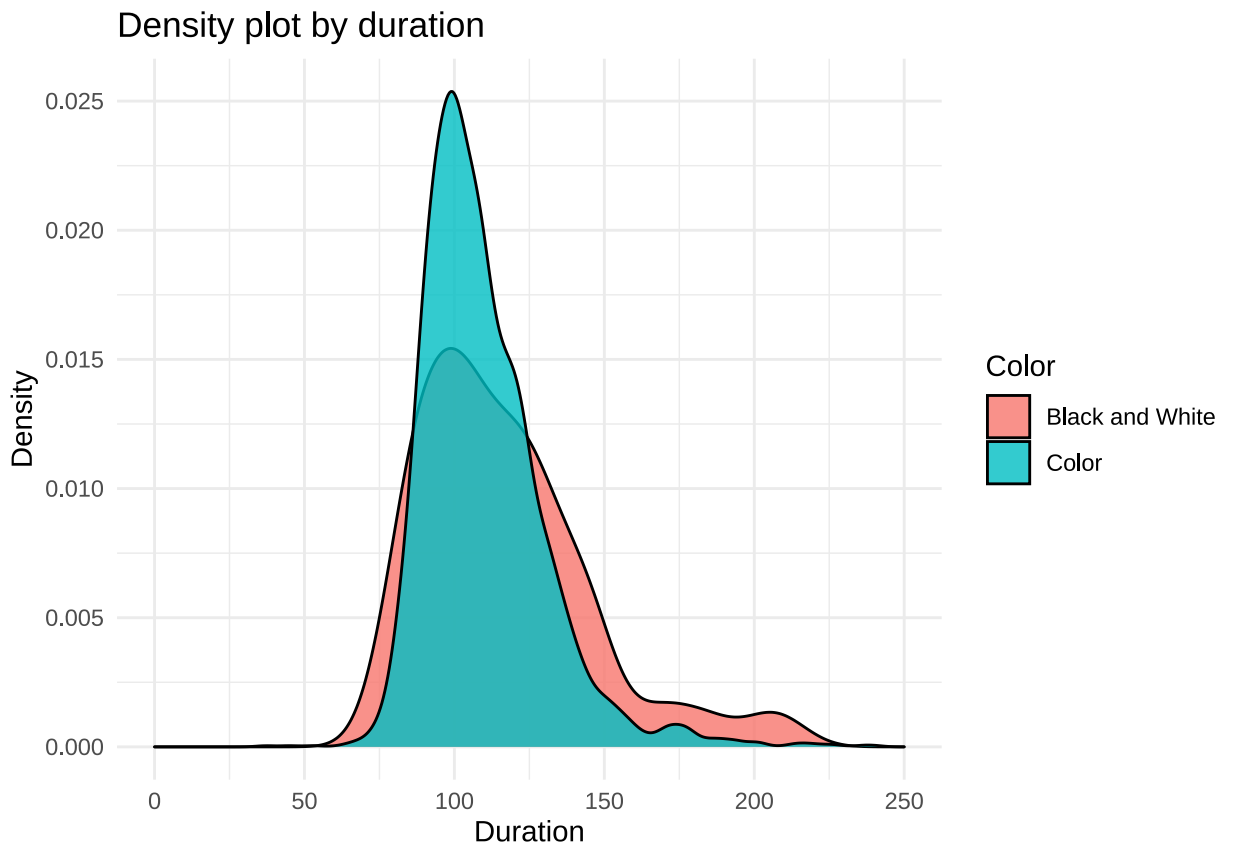


The movies with a duration longer or equal to 150 minutes receive a higher rating than the movies which are shorter than 150 minutes. But this is mainly caused because of the fact that most of the produced movies have an average duration of 110-120 minutes. So it is not entirely representative. On the other hand, movies with a duration of 100 minutes or less are receiving bad ratings based on this scatter plot.

Are colored movies have a longer movie duration than movies which are black and white? Based on the density plot below, we notice that it is almost identical. Both are normally distributed, but the black and white movies have a slightly thicker tail on the right side related to the duration. This implies that the right tail is positively skewed. It can be derived as well that most movies around the 100-minute mark are colored.

```
ggplot(IMDB, aes(duration)) +
  geom_density(aes(fill = factor(color)), alpha = 0.8) +
  labs(title = "Density plot by duration", x = "Duration",
             y = "Density", fill = "Color") +
  theme_minimal() +
  xlim(0, 250)
```

```
## Warning: Removed 7 rows containing non-finite values (stat_density).
```



If we want to know which countries produce the most and best movies, then we first need to filter it to the top 5 countries that produce the most movies:

```
top_5_country <- IMDB %>%
  group_by(country) %>%
  summarise(count = n()) %>%
  top_n(5) %>%
  arrange(desc(count))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)

## Selecting by count
```

```
top_5_country
```

```
## # A tibble: 5 x 2
##   country count
##   <chr>   <int>
## 1 USA      2350
## 2 UK        253
## 3 France     79
## 4 Germany    69
## 5 Canada     43
```

When the countries have been filtered to the top 5 countries with the most produced movies, then we will add the variables that we need to have for the scatter plot based on the selected variables:

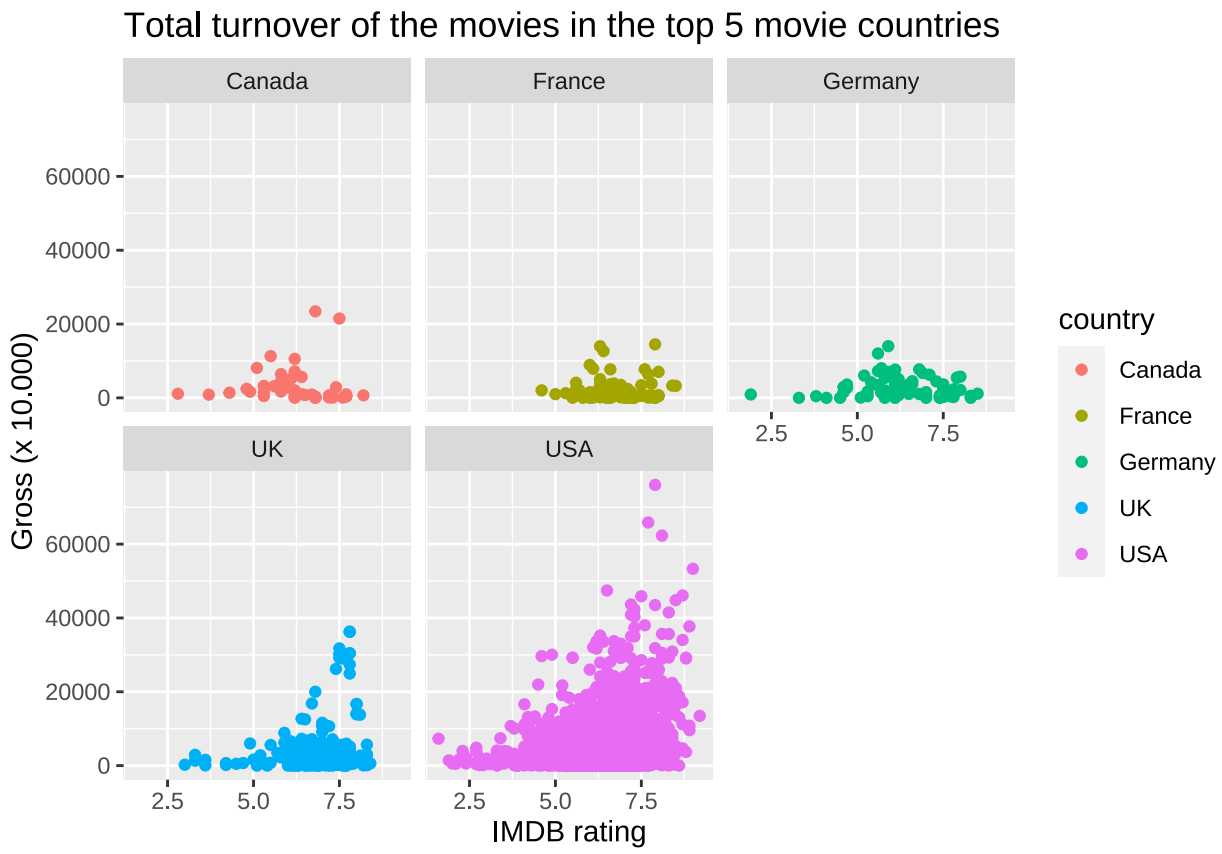
```r
top_5 <- IMDB %>%
  select(country, imdb_score, gross) %>%
  filter(country %in% top_5_country$country)

top_5 <- mutate(top_5, gross_divide = top_5$gross / 10000)

top_5
```

```
## # A tibble: 2,794 x 4
##    country imdb_score    gross gross_divide
##    <chr>        <dbl>    <dbl>        <dbl>
##  1 USA            7.1 53680848        5368.
##  2 USA            5.8 38230435        3823.
##  3 USA            5.5 46729374        4673.
##  4 USA            7.8 10824921        1082.
##  5 USA            5.9 68353550        6835.
##  6 USA            5.4 24268828        2427.
##  7 USA            6.8 26415649        2642.
##  8 UK             7.8  9030581         903.
##  9 USA            6.5 20275446        2028.
## 10 USA            6.3 13391174        1339.
## # ... with 2,784 more rows
```

```r
ggplot(data = top_5, mapping = aes(x = imdb_score, y = gross_divide)) +
  geom_point(aes(colour = country)) +
    facet_wrap(~ country) +
    labs(title = "Total turnover of the movies in the top 5 movie countries", x = "IMDB rating", y = "G
```



Based on this scatter plot, the USA produces the most movies with the highest turnover.