MTH208: Worksheet 7

Web scraping...

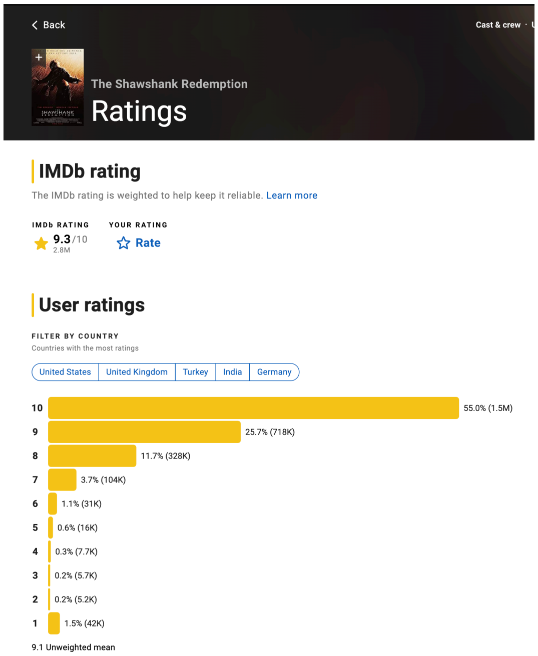
In this worksheet, we will continue scraping the web. First we load some packages

```
library(tidyverse)
library(rvest)
```

We will continue with our IMDb database example, and extract more detailed information. Any given movie on IMDb has a unique code attached to it. For example, the move “The Shawshank Redemption” has the code “tt0111161”. This code is part of the URL of the movie. For example, the URL of “The Shawshank Redemption” is `https://www.imdb.com/title/tt0111161`

Similarly, any movie with code “xyz” has URL `https://www.imdb.com/title/xyz`. Further, IMDb also contains further details of the rating of every movie. This is available at the URL `https://www.imdb.com/title/xyz/ratings`

Go to the URL: `https://www.imdb.com/title/tt0111161/ratings`to see what the page looks like.



The Shawshank Redemption

Ratings

IMDb rating

The IMDb rating is weighted to help keep it reliable. Learn more

IMDb RATING: 9.3/10 (2.8M)

YOUR RATING: Rate

User ratings

FILTER BY COUNTRY

Countries with the most ratings

9.1 Unweighted mean

1. Our goal is to extract the “Unweighted mean” of all the IMDB top 250 movies, and create a dataset of the name of the movies, their unweighted mean, and actual ratings.
2. Create a vector of length 250, having the URL for the poster for each movie in the top 250.

3. For each movie poster, calculate the proportion of pixels with a Euclidean distance of 0.2 units from black.

4. Using `html_table()` extract the whole table in this link:

   `https://www.boxofficemojo.com/chart/top_lifetime_gross`

   of 200 top grossing movies. Clean the final dataset and store in a `csv` file using `write.csv()`.