~~~
---
title: "HighestGrossingFilms"
author: "Oscar Sucre"
output: html_document
---


```{r setup, include=FALSE}
library(tidyverse)
library(rvest)
library(ggthemes)
library(here)
```


## R Markdown


I used Wikipedia here to scrape data from the List of highest-grossing films. Using this link
https://en.wikipedia.org/wiki/List_of_highest-grossing_films


```{r scraping, echo=FALSE}
url <- "https://en.wikipedia.org/wiki/List_of_highest-grossing_films"


Films <- read_html(url) %>%
  html_elements("table") %>%
  .[[3]] %>%
  html_table()


```
~~~

This was the most challenging part of this data set because there were a ton of different inputs in both columns that had to be removed or changed in order to understand the data numerically. Many had ranging data of numbers so I had to chose the first number they put in order to understand the data properly. The only row I filtered out completely was "The Red Shoe" because it was the only row in Euros and wasn't even the highest grossing movie of that year so it was not necessary to be included. Most of the work here was solved with either gsub function or map_chr and str_split to extract the data I wanted to work with.

~~~
```{r data cleaning, echo = FALSE}
Films_Clean <- Films%>%
  select(-`Reference(s)`)%>%
  #Cleaning Worldwide gross column
  mutate(`Worldwide gross`= map_chr(str_split(`Worldwide gross`,"R"),1) ,
         `Worldwide gross`= gsub(",","",`Worldwide gross`),
         `Worldwide gross`= gsub("\\$","",`Worldwide gross`),
         `Worldwide gross`= map_chr(str_split(`Worldwide gross`,"–"),1),
         `Worldwide gross`= map_chr(str_split(`Worldwide gross`,"\\*"),1),
         `Worldwide gross`= map_chr(str_split(`Worldwide gross`,"\\("),1),
         `Worldwide gross`= map_chr(str_split(`Worldwide gross`,"\\+"),1),
         `Worldwide gross`= map_chr(str_split(`Worldwide gross`,"M"),1),
         `Worldwide gross`= map_chr(str_split(`Worldwide gross`,"C"),1),
         `Worldwide gross`= map_chr(str_split(`Worldwide gross`,"\\/"),1),
         `Worldwide gross` = as.numeric(`Worldwide gross`))%>%
  #Cleaning Budget
  filter(Title != "The Red Shoes")%>%
  mutate(Budget = gsub(",","",Budget),
         Budget = gsub("\\$","",Budget),
         Budget = map_chr(str_split(Budget,"–"),1),
         Budget = map_chr(str_split(Budget,"H"),1),
         Budget = as.numeric( Budget ))
~~~

~~~
Films_Clean

```
~~~

This is the data set that took me a while to clean and now can be manipulated to someone's purpose so it is the set I chose to download as the csv file.

~~~
```{r csv file, echo = FALSE}
Films_Clean%>%
 write_csv(here(paste0("HighestGrossingFilms", lubridate::today(), ".csv")))
```
~~~

Here wasn't too complicated since the data was clean. I thought of two interesting ways to use the data. One being the most profitable films and the other being the films with the highest budgets. So I wanted, both are methods to see if the budget was worth the outcome.The names of the variables explain

~~~
```{r manipulating data,echo = FALSE}
Top_Films <- Films_Clean %>%
  mutate(Profit = `Worldwide gross` - Budget)%>%
  arrange(desc(Profit))%>%
  head(10)

Top_Films

Budgeting <- Films_Clean %>%
  arrange(desc(Budget))%>%
  mutate(Budget_Percent =( Budget/`Worldwide gross`)*100 )%>%
  head(10)

Budgeting
```
~~~

Here I plotted the data using ggplot and focused on making it easy to understand and appealing with the columns I made "Profit" and "Budget_Percent". For "Profit" once arranging the data as a bar plot I added color by `Year` and it shows how most of the movies are relatively new with the exception of Titanic. In order to fit the movie titles had to put them on an angle for it to be readable.Dollar format for Y is important for reading the information, as it is showing that it's in USD,also it looks much better then it's original display of scientific notation.

Using the Budgeting data I made a second graph to show off the "Budget Percent" variable I made. Firstly had to change it back to decimal form because formatting with scalles::percent multiplies the variable by 100 to turn it into a percentage. Another change with the second graph is that the order goes from left to right because the formatting makes more sense fitting more information because the y axis takes up less space in this graph compared to the Profit graph, so the angle of the x axis text also has to change.

```{r plotting, echo = FALSE}
 Top_Films%>%
 ggplot(aes(x = reorder(Title, -Profit),Profit, fill = Year ))+
 geom_col(alpha =.8)+
 labs(title = "Top 10 Most Profitable films",x = "Movie Titles",y = "Profit in USD")+
 theme(axis.text.x = element_text(size = 10, angle = 25, hjust = 1),
       axis.text.y = element_text(size = 10, hjust = 1))+
 scale_y_continuous(labels=scales::dollar_format())


 Budgeting%>%
 mutate(Budget_Percent = Budget_Percent/100)%>%
 ggplot(aes(x = reorder(Title, +Budget_Percent),Budget_Percent, fill = Year ))+
 geom_col(alpha =.8)+
 labs(title = "Top 10 Highest Percentage of Budget over Gross",x = "Movie Titles",y = "Budget Percent")+
 theme(axis.text.x = element_text(size = 10, angle = 45, hjust = 1),
       axis.text.y = element_text(size = 10, hjust = 1),)+
 scale_y_continuous(labels = scales::percent)
```

On my honor, I have neither received nor given any unauthorized assistance on this assignment.

Oscar Sucre