# Anahuac ML: Problem Set 1

*Prof. Jonathan Hersh*

## Question 1, Plotting IMDB's Top 5000 Movies.

a) A note on directories. To see the curent working directory – execute the code `getwd()`. Let's set our current working directory to the current location of our project file. This should happen by default if we are using R project files.

**Note if you are using windows you must change all your backword slashes ("\") to forward slashes ("/") for any directory structure.**

```
getwd()
## [1] "/Users/hersh/Dropbox/Anahuac_ML/psets"
```

b) Download the Top 5000 movies on IMDB from the following link: https://www.kaggle.com/carolzhangdc/imdb-5000-movie-dataset/downloads/imdb-5000-movie-dataset.zip/1. Be sure to save it in a subdirectory of your MGSC_310 folder called "datasets".

```
# note we must use '/' -- not '\'!  change to the directory
# where you stored your movie metadata.
getwd()
## [1] "/Users/hersh/Dropbox/Anahuac_ML/psets"
movies <- read.csv(here::here("datasets", "movie_metadata.csv"))
```

c) What are the dimensions of the dataset? Programatically determine this using a function.

d) What are the names of the variables in the data set? Hint: use the function `names` here.

e) Use the package `ggplot2` to create a scatterplot of IMDB on the x-axis and movie budgets on the y-axis.

f) It looks like there are some outliers in terms of budget. The highest budget movie of all time was *Pirates of the Caribbean: On Stranger Tides* which cost $387.8m. Any movies with a budget higher than this must be a data anomaly. Run the code below to remove rows of movies which cost more than $400m to produce.

Now how many movies do we have in our data set?

```
library("tidyverse")
movies <- movies %>% filter(budget < 4e+08)
```

g) Use `stat_smooth()` to create a trendline to the above figure. Is there a relationship between IMDB score and budget?

h) Use `facet_wrap()` to create sub-plots of relationship between IMDB Score and Budget. (Note, within the function `facet_wrap()` use the option `, scales = "free"` to allow the x-axes and y-axes to vary per sub-plot.For which content ratings do we see the strongest relationship between budget and IMDB score?

i) Install the R package `ggrides` to produce ridgeline density plots of simplified genre plots using the code below.

```
install.packages("ggridges")
library("ggridges")
movies <- movies %>% mutate(genre_main = unlist(map(strsplit(as.character(movies$genres),
    "\\|"), 1)), grossM = gross/1e+06, budgetM = budget/1e+06)

ggplot(movies, aes(x = grossM, y = genre_main, fill = genre_main)) +
    geom_density_ridges() + scale_x_continuous(limits = c(0,
    500)) + labs(x = "Box Office Gross (USD Millions)", y = "Main Genre")
```

j) In a series of graphs (at least two) explore the relationship between budget and gross.

## Question 2, Plotting IMDB's Top 5000 Movies.

a) We're going to work with the movies dataset again. Download the Top 5000 movies on IMDB from the following link: https://www.kaggle.com/carolzhangdc/imdb-5000-movie-dataset/downloads/imdb-5000-movie-dataset.zip/1. Be sure to save it in a subdirectory of your MGSC_310 folder.

b) Run the code below to filter out films with unreasonably large budgets. Also use the code to create new variables (mutate) that are simplified versions of the genres and budget variables.

```
library("tidyverse")
movies <- read.csv(here::here("datasets", "movie_metadata.csv"))
movies <- movies %>% filter(budget < 4e+08)
movies <- movies %>% mutate(genre_main = unlist(map(strsplit(as.character(movies$genres),
    "\\|"), 1)), grossM = gross/1e+06, budgetM = budget/1e+06)
movies <- movies %>% mutate(genre_main = factor(genre_main) %>%
    fct_drop())
```

c) Use the `mutate` function to generate `profitM` which is the difference between a movie's gross and its budget, and the variable `ROI` which is the return on investment, specifically profit as a ratio of budget.

d) What is the average ROI for firms in the dataset? Use the function `geom_histogram()` to create a histogram of ROI for movies in the database.

e) From the plot above, it should be clear several outliers throw off the plot. Use the filter command to filter out films that have an ROI greater than 10. Just to be careful, count the number of films which match this criteria using the `count()` function.

f) Use the `group_by()` function to aggregate films by `genre_main` and create mean ROI by genre using the `summarize()` command. Which film genres have the highest return on investment (ROI)? (Note we can also create standard errors by including in the `summarize()` command `se_ROI_genre = sd(ROI, na.rm = TRUE)/sqrt(n())`)

g) Use `ggplot` to create plots of the average ROI by genre using `geom_point()`. Note you can also use `geom_errorbar()` to create standard error bars using the code below.

```
# geom_errorbar(mapping = aes(ymin = avg_ROI_genre -
# se_ROI_genre, ymax = avg_ROI_genre + se_ROI_genre),...
```

h) Use the summarize command to compute averages by `actor_1_name` for ROI and profit. Also within your summarize command calculate `num_films` as the number of films by actor. Use the `arrange()` command to sort the dataframe in descending order by average actor ROI. Finally use `slice()` to print the first twenty rows. Which three actors have the highest ROIs?

i) Use `geom_point()` to plot the 30 actors with the highest ROI. Note we can use the `top_n()` command to plot only the top 30 actors. We can also use `fct_reorder()` to order the actors by highest ROI.

```
library(forcats)
# ggplot(data = actor_summary %>% top_n(30, wt =
# avg_ROI_actor), mapping = aes(x = fct_reorder(actor_1_name,
# avg_ROI_actor) %>% fct_drop(),...
```

j) Plot the 30 actors with the lowest return on investment

## Questions 3 ISLR Chapter 2, problem 3

## Question 4, What Predicts Movie Profitability?

a) We're going to work with the movies dataset again. Download the Top 5000 movies on IMDB from the following link: https://www.kaggle.com/carolzhangdc/imdb-5000-movie-dataset/downloads/imdb-5000-movie-dataset.zip/1. Be sure to save it in a subdirectory of your MGSC_310 folder.

b) Run the code below to filter out films with unreasonably large budgets and movies with

missing content ratings. Also use the code to create new variables (mutate) that are simplified versions of the genres, budget variables, and cast facebook likes.

```
library("tidyverse")
options(scipen = 10)
movies <- read.csv(here::here("datasets", "movie_metadata.csv"))
movies <- movies %>% filter(budget < 400000000) %>% filter(content_rating !=
    "", content_rating != "Not Rated")
movies <- movies %>% mutate(genre_main = unlist(map(strsplit(as.character(movies$genres),
    "\\|"), 1)), grossM = gross/1000000, budgetM = budget/1000000,
    profitM = grossM - budgetM, cast_total_facebook_likes000s = cast_total_facebook_likes/1000)
movies <- movies %>% mutate(genre_main = factor(genre_main) %>%
    fct_drop())
```

c) Split the movies dataset into a testing and training set, with the training set 80% of the size of the original dataset. Be sure to use `set.seed(1861)` to ensure your results are comparable to mine and your classmates.

d) How many rows are in the test and trainig datasets?

e) In building a regression model, a good place to start is producing a correlation matrix that shows which variables are positively or negatively correlated with the variable we want to predict. We can only correlate numeric variables so run the code below to produce the correlation matrix. This code does two things: 1) the `select_if(is.numeric)` selects only the numeric variables, and 2) the `drop_na()` removes missing values from the correlation matrix. It then prints the correlation coefficient between `profitM` and all the numeric varibles in the data frame. Which variables are most strongly (positivelty or negatively) correlated with profits?

```
cormat <- cor(movies_train %>% select_if(is.numeric) %>% drop_na())
print(cormat[, "profitM"])
```

f) *Extra Credit.* Use the `corrplot` package to produce a plot of the correlation matrix.

g) Let's regress `profitM` against `imdb_score` and store this as `mod1`. Use the `summary()` function over `mod1` to print the regerssion summary. Be sure to estimate our model against the training dataset.

h) Interpret the coefficient for `imdb_score`, being specific about the impact regarding magnitude and sign.

i) What is the p-value associated with the estimate of `imdb_score`? In your own words, what does a p-value mean? What does this estimate p-value imply about the relationship between `imdb_score` and profit?

j) Include `cast_total_facebook_likes000s` as a predictor in addition to `imdb_score`. Store this model as `mod2` and use `summary()` to output the results.

k) What is the estimated impact of cast facebook likes on movie profits?

l) What is the impact of content rating on a movie's expected profit? To answer this question we will have to clean `content_rating` a little bit. Use the `fct_lump()` function to create factor levels for the four most common factor levels, leaving the rest as a category "other". Call this variable `rating_simple` and store it in the `movies_train` data frame.

```
table(movies_train$rating_simple)
##
##     G    PG PG-13     R Other
##    92   509  1107  1559   129
```

m) Estimate a model with `profitM` on the left-hand-side and `imdb_score`, `cast_total_facebook_likes000s` and `rating_simple` on the right-hand side. Interpret the coefficient for `rating_simpleR`.

n) Why does the coefficient for G not appear in the regression table above?