Problem Set 3

MGSC 310

Prof. Shahryar Doosti

Follow the submission instructions stated in the course website

IMDB's Top 5000 Movies

In this problem set, we will work on a data from kaggle.com (link: https://www.kaggle.com/carolzhangdc/imdb-5000-movie-dataset) providing information on top 5000 IMDB movies. You can download the data from Canvas as well.

1. Data Exploration

a. Load the dataset in a dataframe named `movies`. What are the dimensions of the dataset?
b. What are the names of variables in the dataset?
c. How many missing values are there in `budget`? Remove them using the following code. Check the dimensions again.

```
movies <- movies[!is.na(movies$budget),]
```

d. How many unique directors are there? Hint: use `unique()` function to get the unique values for `director_names`, then use `length()` to count them. Use these two functions in a single command.
e. Use `ggplot2` package to create a scatterplot of IMDB score on the x-axis and movie budget 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.

```
movies <- movies[movies$budget<400000000,]
```

Now, how many movies do we have in our dataset?

g. Use geom_smooth() to create a *linear* 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?

2. Data Manipulation

Use the code below to create new variables that are simplified versions of the genres and budget variables.

```
# to create budget and gross columns in millions
movies$grossM <- movies$gross/1e+6
movies$budgetM <- movies$budget/1e+6
# note how we created new columns

# to create a column for main genre
movies$genre_main <- do.call('rbind',strsplit(as.character(movies$genres), '|', fixed=TRUE))[,1]
```

a. Generate a new column `profitM` which is the difference between a movie's gross and its budget, and the variable `ROI` which is the return on investment (profit as a ratio of budget).
b. What is the average ROI for films in the dataset? Hint: use `na.rm=TRUE` option in `mean()` function to ignore missing values.
c. Create a histogram of ROI for movies in the dataset.
d. From the plot above, it should be clear several outliers throw off the plot. Filter out films that have an ROI greater than 10. First, count the number of films which match this criteria, then remove them.
e. Create the histogram again. Optional: you can use `geom_histogram()` from `ggplot2` library with option `aes(fill = genre_main)` to create different histograms for main genres.
f. Use the `summaryBy()` function from `doBy` package to create mean ROI by genre_main. Which film genres have the highest return on investment (ROI)?
g. Use ggplot to create plots of the average ROI by genre using geom_point().

3. Simple Linear Regression

In this exercise, you split the dataset into train and test sets. Then, you perform a simple linear regression.

a. Now, split the movies dataset into testing and training sets, with the training set 80% of the size of the original dataset. Be sure to use set.seed(310) to ensure your results are comparable to mine and your classmates. Hint: use sample function as `sample(1:nrow(movies), .8*nrow(movies))` to generate indexes for train set `train_idx`. For test, use exclude operator to choose whatever is not in the training. You should have two dataframes (training and test).
b. What are the dimenstions for training and test sets?
c. 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. Hint: use `lm(profitM ~ imdb_score,train_movies)`
d. Get the coefficients (parameters) of the model using `coef(mod1)`. Explain what these parameters mean?