Lab Report 1

Andy Tan, Tong (Amy?) Wu and Morgan Yeung

Question 1: Linear Regression

1.1. (10 pts)

Give basic insights into your numeric variable you have picked as output variable using one categorical variable you selected.

- What are the min / max values and median of the output variable, Y?
- What is median of the output value among different classes of the categorical variable you picked? You must use `group_by` and `summarize` functions.

```
library('tidyverse')
```

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

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.5     v dplyr   1.0.3
## 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()
```

```
theme_set(theme_minimal())
# Step 1
data <- read.csv("data/anime.csv")
data.smaller <- select(data,popularity,genre)
group_by(data.smaller,genre)%>%
summarise(min_popularity = min(popularity))
```

```
## # A tibble: 41 x 2
##    genre     min_popularity
##  * <chr>              <int>
##  1 Action                 2
##  2 Adventure              3
##  3 Cars                 516
##  4 Comedy                 4
##  5 Dementia              50
##  6 Demons                16
##  7 Drama                  2
##  8 Ecchi                  9
##  9 Fantasy                2
## 10 Game                   3
## # ... with 31 more rows
```

```r
group_by(data.smaller,genre)%>%
summarise(max_popularity = max(popularity))
```

```
## # A tibble: 41 x 2
##    genre     max_popularity
##  * <chr>              <int>
##  1 Action             15461
##  2 Adventure          15432
##  3 Cars               15295
##  4 Comedy             15472
##  5 Dementia           15453
##  6 Demons             15231
##  7 Drama              15460
##  8 Ecchi              15356
##  9 Fantasy            15473
## 10 Game               15419
## # ... with 31 more rows
```

```r
group_by(data.smaller,genre)%>%
summarise(mid_popularity = median(popularity))
```

```
## # A tibble: 41 x 2
##    genre     mid_popularity
##  * <chr>              <dbl>
##  1 Action              2531
##  2 Adventure           4660.
##  3 Cars                8434
##  4 Comedy              3491
##  5 Dementia           10788
##  6 Demons              2288
##  7 Drama               2719
##  8 Ecchi               1618
##  9 Fantasy             3244
## 10 Game                3587
## # ... with 31 more rows
```

1.2. (10 pts)

Visualize the variables you selected.

- Draw histogram of the numeric variables you selected.
- Draw distribution of the output variable Y with respect to the different classes of your categorical variable. The plot must somehow show the distributional differences among different classes. You can use boxplot, histogram, or other visuals (e.g. density ringes).
- Draw scatter plot between one of your numeric inputs and the output variable. Discuss whether the plot indicate a relation, if it is linear, if there are outliers? Feel free to remove the outlier. Feel free to transform the data.

1.3. (15 pts)

Using the all dataset, fit a regression:

1. Using the one numeric input variable fit a simple regression model.

- Write down the model.
  - Fit the regression line.
  - Summarize the output.
  - Plot the input and output variables in a scatter plot and add the predicted values as a line.
  - Interpret the results. Is it a good fit? Is your input variable good in explaining the outputs?

2. Using all your input variables, fit a multiple linear regression model

   - Write down the model
   - Fit the regression line and summarize the output
   - Interpret the results. Is it a good fit? Are the input variables good in explaining the outputs?

3. Now, do the same things as you did, but this time add an interaction between one categorical and one numeric variable.

   - Write down the model, fit to the data, summarize and interpret the results.

4. Which model you fit is the best in predicting the output variable? Which one is the second and third best? Rank the models based on their performance.

1.4. (15 pts)

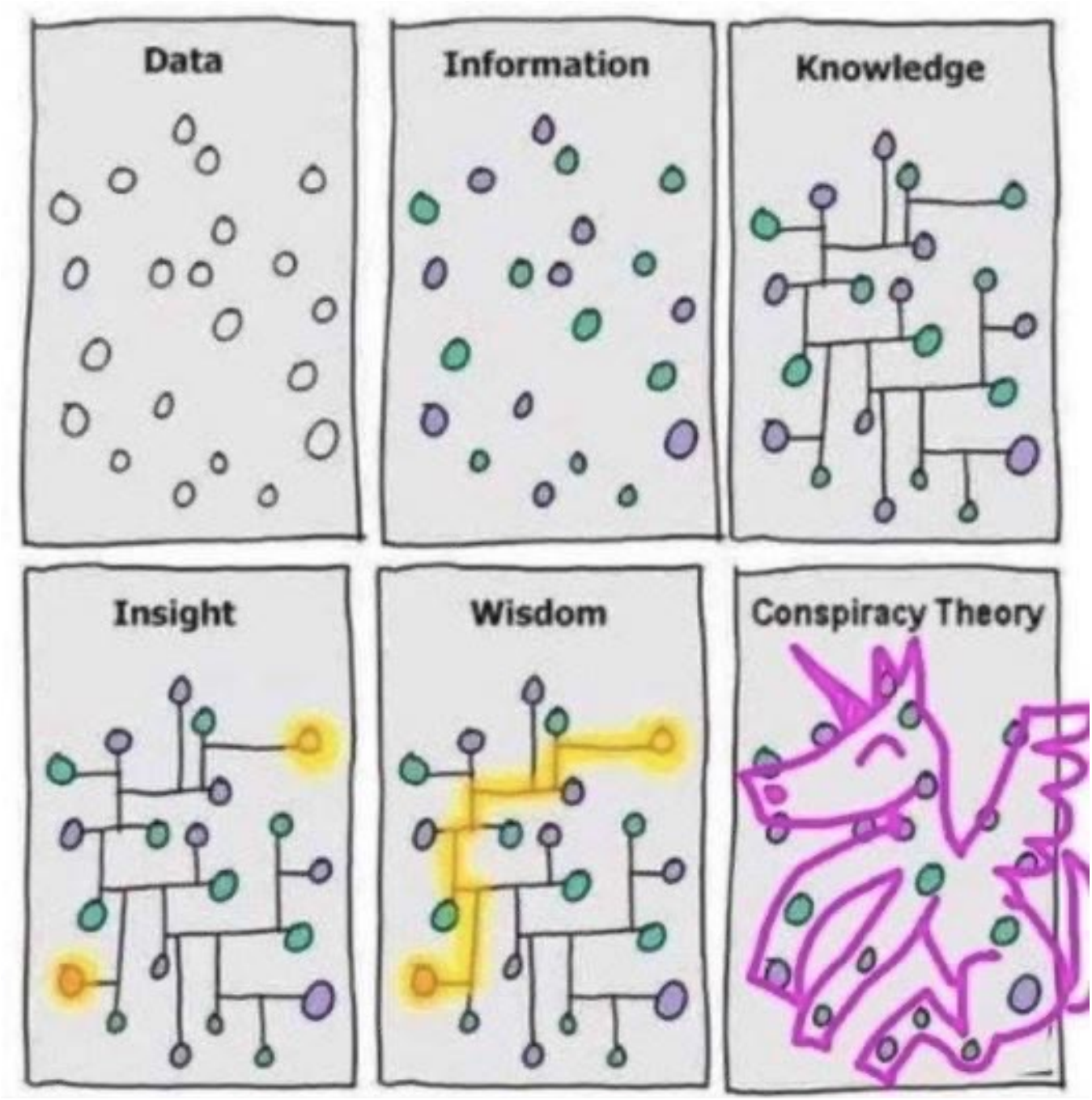
In this section, you will do the same you did in 1.3, but this time you will first split the data into train and test.

- Select seed to fix the random numbers you will generate using `set.seed(...)`.
- Split your data into test and train sets with 20/80 test-train ratio.
- Fit the model to the train set and evaluate the how well the model performed on test set.
- Which model performed the best on test set? Rank the models based ion their performance.
- Is the rank the same as the one you had in 1.3?

Question 2: Gradient Descent Algorithm (By hand)

In case you want to take a picture (screenshot) of your notebook (tablet), you can use the below lines to embed the image to the output PDF file:

```
knitr::include_graphics('conspiracy.jpg')
```



Question 3. Gradient Descent Algorithm

3.1. Get familiar

You will use horsepower as input variable and miles per gallon (mpg) as output:

1. Plot the scatterplot between `mpg` (Y) and `horsepower` (X).
   - Is the relationship positive or negative? Does mpg increase or reduce as horsepower increases?
   - Is the relationship linear?
2. Plot the scatterplot between `log(mpg)` and `log(horsepower)`.
   - Is the relationship positive or negative?
   - Is the relationship linear?
3. Which of the two versions is better for linear regression?

3.2. Fill in the code

The code below estimates the coefficients of linear regression using gradient descent algorithm. If you are given a single linear regression model;

$$Y = \beta_0 + \beta_1 X$$

where $Y = [Y_1, \ldots, Y_N]^T$ and $X = [X_1, \ldots, X_N]^T$ are output and input vectors containing the observations.

The algorithm estimates the parameter vector $\theta = [\beta_0, \beta_1]$ by starting with an arbitrary θ_0 and adjusting it with the gradient of the loss function as:

$$\theta := \theta + \frac{\alpha}{N} X^T (Y - \theta X)$$

where α is the step size (or learning rate) and $(Y - \theta X)^T X$ is the gradient. At each step it calculates the gradient of the loss and adjusts the parameter set accordingly.

3.3. Run GDA

1. Run the code with the above parameters. How many iterations did it take to estimate the parameters?
2. Reduce epsilon to `1e-6`, set `alpha=0.05` run the code.
   - How many iterations did it take to estimate the parameters?
   - Does the result improve? Why or why not?
3. Reduce alpha to `alpha=0.01`
   - How many iterations did it take?
   - Did the resulting line change? Why or why not?
4. Set alpha back to `alpha=0.05` and try `theta0=c(1,1)` vs. `theta0=c(1,-1)`:
   - How many iterations did it take? Which is less than the other?
   - Why starting with a negative slope have this effect?
5. Reduce epsilon to `epsilon = 1e-8` and try `alpha=0.01`, `alpha=0.05` and `alpha=0.1`.
   - What effect does alpha have on iterations and resulting fitted line?