Vietnam National University – Ho Chi Minh City

UNIVERSITY OF SCIENCE

FACULTY OF INFORMATION TECHNOLOGY

Applied Statistics for Engineers and Scientists II

STAT452

Topic: Final Project

Group 4 - 22APCS2

Group Members:
O Hon Sam(22125085)
Dao Xuan Thanh(22125095)
Phan Phuc Bao(22125010)
Le Phat Minh(22125056)

Supervisors:
Mrs. Nguyen Thi Mong Ngoc
Mr. Nguyen Huu Toan

2024-08-20

I. Task Schedules

Dataset	Task	Assigned Member	Completion
Activity 1	Import and preprocess data	Hon Sam	100%
	Descriptive statistics	Hon Sam, Xuan Thanh	100%
	Split data to train and test	Hon Sam	100%
	Build model	Phuc Bao, Phat Minh	100%
	Model Diagnostic	Phuc Bao, Phat Minh	100%
	Prediction	Xuan Thanh	100%
	Evaluation	Xuan Thanh	100%
Activity 2: Happiness	Introduce activity	Phuc Bao	100%
	Import and preprocess data	Phuc Bao, Phat Minh	100%
	Descriptive statistics	Phuc Bao, Phat Minh	100%
	Split data to train and test	Phuc Bao	100%
	Build model	Phat Minh	100%
	Model Diagnostic	Phat Minh	100%
	Prediction	Phuc Bao	100%
	Evaluation	Phuc Bao	100%
	Conclusion	Phuc Bao, Phat Minh	100%
	Proofread	Xuan Thanh, Hon Sam	100%
Activity 2: Suicide	Introduce activity	Hon Sam	100%
	Import and preprocess	Hon Sam	100%
	Descriptive statistics	Xuan Thanh, Hon Sam	100%
	Split data to train and test	Xuan Thanh	100%
	Build model	Xuan Thanh, Hon Sam	100%
	Model Diagnostic	Hon Sam	100%
	Prediction	Xuan Thanh	100%
	Evaluation	Xuan Thanh	100%
	Conclusion	Xuan Thanh, Hon Sam	100%
	Proofread	Phuc Bao, Phat Minh	100%
Making report (latex)		All members	100%

Contents

II. Activity 1

II.1. Import and preprocess dataset

II.1.1. Import dataset

```
[1] 398   9
'data.frame':   398 obs. of  9 variables:
 $ mgp         : num  18 15 18 16 17 15 14 14 14 15 ...
 $ cylinders   : int  8 8 8 8 8 8 8 8 8 8 ...
 $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
 $ horsepower  : chr  "130.0" "165.0" "150.0" "150.0" ...
 $ weight      : int  3504 3693 3436 3433 3449 4341 4354 4312 4425 3850 ...
 $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
 $ model_year  : int  70 70 70 70 70 70 70 70 70 70 ...
 $ origin      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ car_name    : chr  "chevrolet chevelle malibu" "buick skylark 320"
"plymouth satellite" "amc rebel sst" ...
```

Figure 1: Structure of auto-mpg dataset.

Figure 1 states that there are total 398 observations of 9 variables

II.1.2. Process missing value

When it comes to cleaning the missing value, we often think of removing NA value. However, in this auto-mpg dataset, the missing values are represented as "?" notation. Therefore, we need the first part of Code Snippet 2 to convert "?" into "NA" before further fine-tuning them.

```
          mgp    cylinders displacement   horsepower       weight
            0            0            0            6            0
acceleration   model_year       origin     car_name
            0            0            0            0
```

Figure 2: The number of missing values in auto-mpg dataset

Since only the 'horsepower' predictor has missing values in Figure 2, we concentrate on processing it. As we observed in Figure 1, the data type of 'horsepower' is chr instead of num. The second part of Code Snippet 2 will do the conversion job to make 'horsepower' ready for analysis.

The 'horsepower' column has only 6 missing values, which account for approximately 1.5%, so removing them will not affect the dataset significantly. It avoids the risk of introducing bias or inaccuracies that might occur if we replace missing values with the mean (or median) of the column.

II.1.3. Process duplicate rows

Based on the output of Code Snippet 3, there are no duplicate rows.

II.1.4. Process unnecessary variables

We decided that the variable 'car_name' isn't useful for extracting information. While it could potentially tell us the manufacturer of each engine, grouping the data by manufacturers is unnecessary since we already have the 'origin' variable. Creating an extra grouping based on manufacturers would be overly complicated as there will be so many dummy variables, which should be avoided in grouping technique. Therefore, I'll skip using 'car_name' for analysis by using Code Snippet 4.

II.1.5. Descriptive statistics

NOTE: "After preprocessing steps" refers to the point after completing all the steps in Section I.1: Import and preprocess dataset.

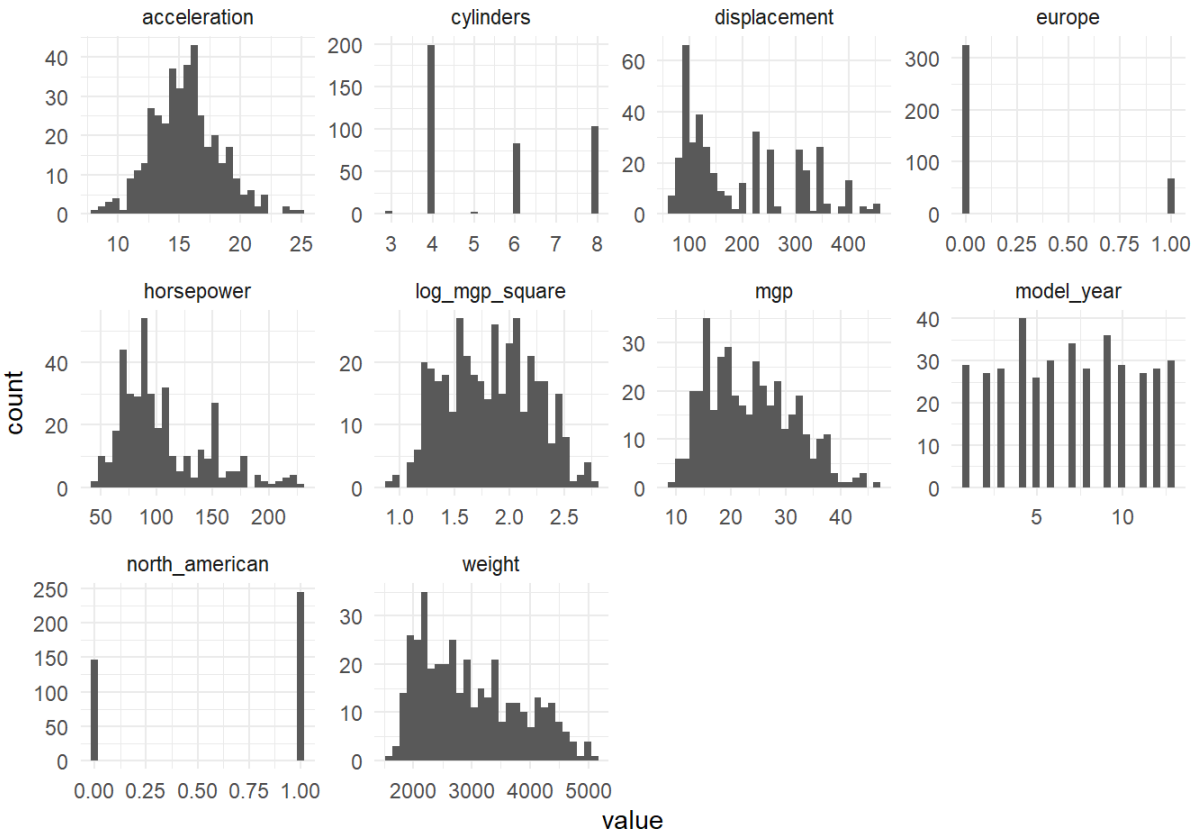


Figure 3: Histograms of all variables after preprocessing steps.

Figure 3 shows that our response variable, mpg, is right-skewed, so we need to normalize it.

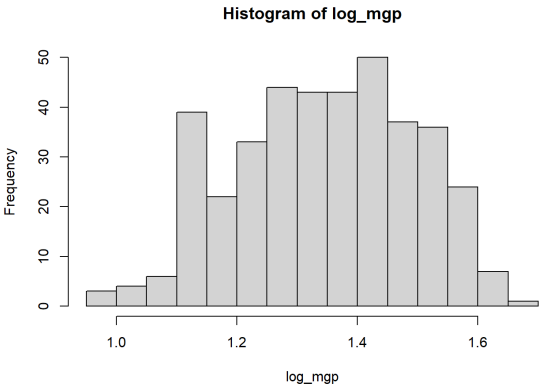


Figure 4: Histogram of $log_{10}(mgp)$.

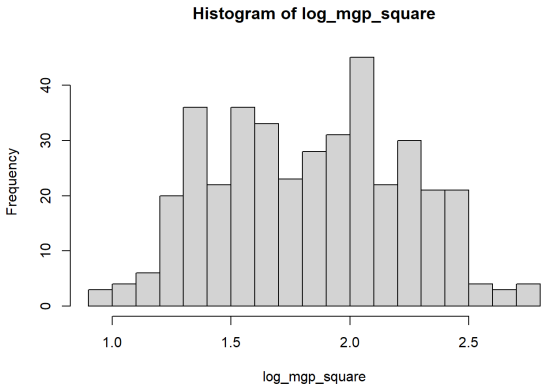


Figure 5: Histogram of $log_{10}(mgp)^2$.

A common way to fix right-skewness is by taking the logarithm of the variable. However, after taking the log, the graph shown in Figure 4 has a bit of left-skewness. To reduce this, we squared the $log_{10}(mpg)$. The histogram of $log_{10}(mpg)^2$ in Figure 5 shows a shape that is close to a normal bell curve, which is much better. The efficiency of $log_{10}(mpg)^2$ over $log_{10}(mpg)$ or mpg is proven in the hypothesis testing section.

```
     mgp          cylinders      displacement     horsepower
 Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0
 1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0
 Median :22.75   Median :4.000   Median :151.0   Median : 93.5
 Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5
 3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0
 Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0
     weight      acceleration     model_year        origin
 Min.   :1613   Min.   : 8.00   Min.   :70.00   Min.   :1.000
 1st Qu.:2225   1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000
 Median :2804   Median :15.50   Median :76.00   Median :1.000
 Mean   :2978   Mean   :15.54   Mean   :75.98   Mean   :1.577
 3rd Qu.:3615   3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000
 Max.   :5140   Max.   :24.80   Max.   :82.00   Max.   :3.000
```

Figure 6: Summary of auto-mpg dataset before processing categorical variables.

```
     mgp          cylinders      displacement     horsepower
Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0
1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0
Median :22.75   Median :4.000   Median :151.0   Median : 93.5
Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5
3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0
Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0
    weight      acceleration     model_year    log_mgp_square
Min.   :1613   Min.   : 8.00   Min.   : 1.00   Min.   :0.9106
1st Qu.:2225   1st Qu.:13.78   1st Qu.: 4.00   1st Qu.:1.5140
Median :2804   Median :15.50   Median : 7.00   Median :1.8414
Mean   :2978   Mean   :15.54   Mean   : 6.98   Mean   :1.8323
3rd Qu.:3615   3rd Qu.:17.02   3rd Qu.:10.00   3rd Qu.:2.1386
Max.   :5140   Max.   :24.80   Max.   :13.00   Max.   :2.7835
north_american      europe
Min.   :0.000   Min.   :0.0000
1st Qu.:0.000   1st Qu.:0.0000
Median :1.000   Median :0.0000
Mean   :0.625   Mean   :0.1735
3rd Qu.:1.000   3rd Qu.:0.0000
Max.   :1.000   Max.   :1.0000
```

Figure 7: Summary of cleaned auto-mpg dataset after preprocessing steps.

Comments on summary of datasets in Figure 7 and Figure 6:

- The mean of 'mpg' is 23.45, and the median is 22.75, showing that 'mpg' is slightly right-skewed. After transforming it to 'log_mgp_square,' the mean and median are almost the same, with only a 0.0091 unit difference. This shows that the transformation helped normalize skewness and stabilize the variance.
- The 'displacement' ranges from min value of 68.0 to max value of 455.0, indicating significant variation in engine size. Larger engines might be slightly more common in this dataset (left-skewed).
- 'horsepower' varies widely from 46.0 to 230.0. With the mean 'horsepower' (104.5) being higher than the median (93.5), there is a slight positive skew in this predictor, indicating that there are a few with significantly higher horsepower (such as sport cars, etc.) pulling the mean upwards.
- The mean 'acceleration' is 15.54 seconds, suggesting most vehicles have moderate acceleration.
- The mean 'weight' is 2978, with a slight skew towards heavier vehicles (as the mean is slightly higher than the median by 174 units).
- The mean 'model_year' in Figure 6 is 75.98, indicating a concentration around the mid-1970s.

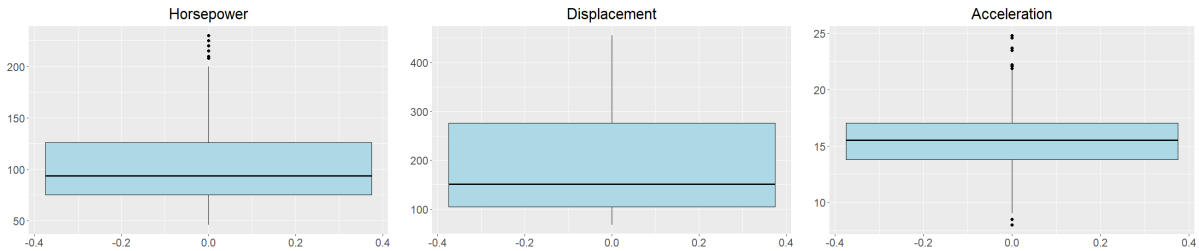


Figure 8: Boxplots of 'horsepower', 'displacement' and 'acceleration' in auto-mpg dataset after preprocessing steps.

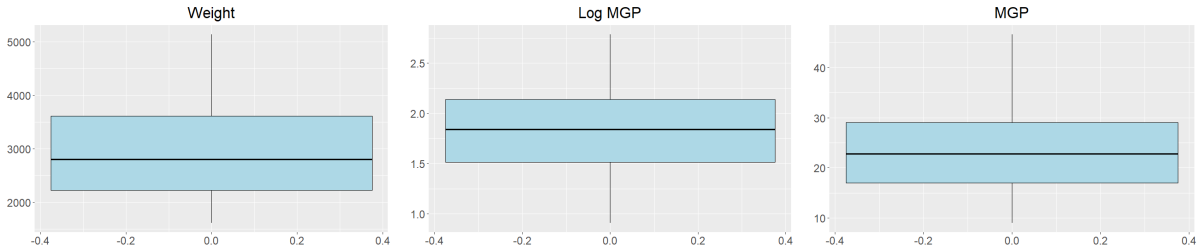


Figure 9: Boxplots of 'weight' and 'mgp' and transformed '$log_{10}(mgp)^2$' in auto-mpg dataset after preprocessing steps.

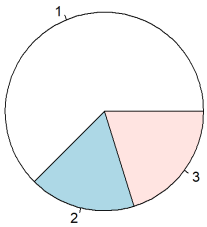


Figure 10: Pie chart of 'origin' variable.

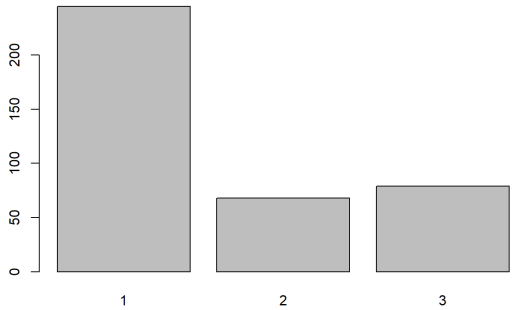


Figure 11: Barplot of 'origin' variable.

The pie chart 10 and frequency barplot 11 show that a significant portion of the cars are from North America (1) (62.56281%), while cars from Europe (2) and Asia (3) make up smaller portions, with 17.58794% and 19.84925%, respectively. The distribution of origins could impact the analysis, especially when comparing characteristics like fuel efficiency or engine power across different regions.

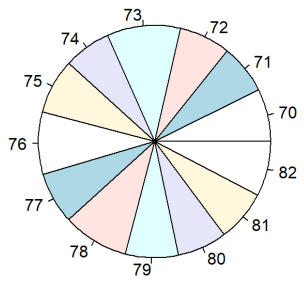


Figure 12: Pie chart of 'model_year' variable.

The 'model_year' distribution shown in Figure 12 is fairly even. This suggests that the dataset includes a well-balanced representation of cars from different years. 'model_year' allows for an analysis of trends over time, such as how car characteristics (e.g., 'mpg', 'horsepower') might have evolved over these years.

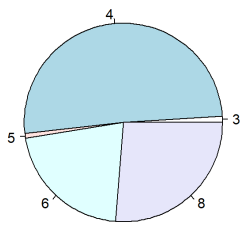


Figure 13: Pie chart of 'cylinders'.

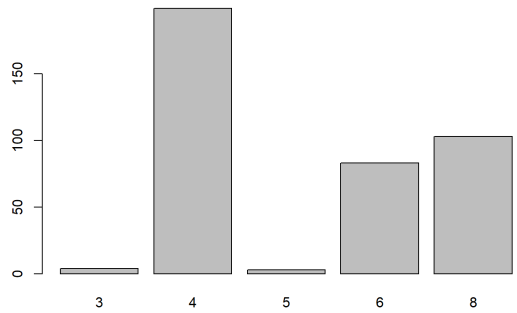


Figure 14: Barplot chart of 'cylinders'.

The distribution of 'cylinders' barplot 14 seems to be bimodal, with peaks at 4 and 8 cylinders.

The chart 14 and 13 shows that the most common number of cylinders among the vehicles is 4, with over 150 occurrences. The next most common is 8 cylinders, followed by 6 cylinders. Vehicles with 3 and 5 cylinders are rare, with very few occurrences.

II.1.6. Relationship between response variable and predictors

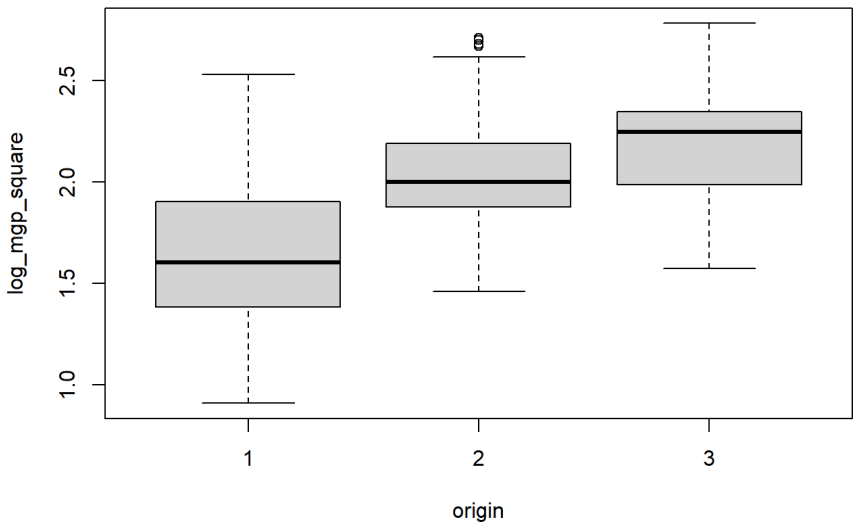


Figure 15: Boxplot showing distribution of 'log_mgp_square' across different 'origin'.

The highest 'log_mgp_square' consumption coming from Asia countries while the lowest one coming from North America. It indicates that the engines from North America might be better than the other areas.

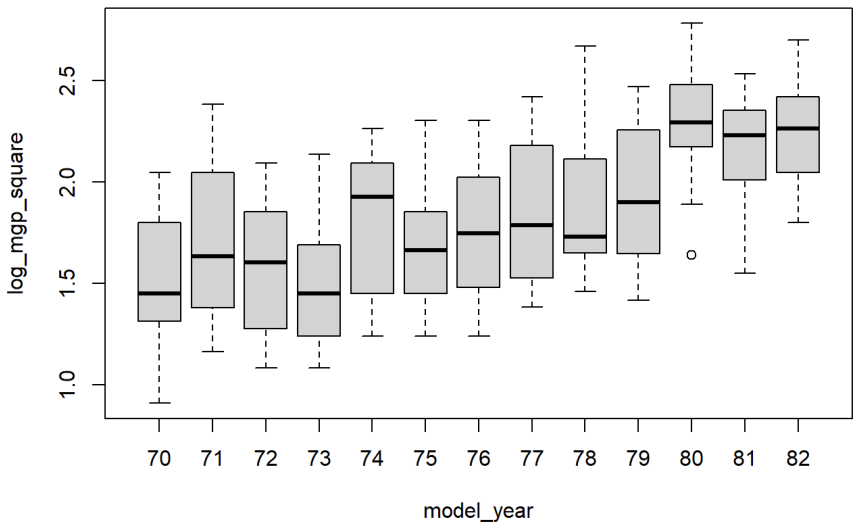


Figure 16: Boxplot showing fuel efficiency trends 'log_mgp_square' over 'model_year'.

From 1970 to 1982, the ‘log_mgp_square’ value shows a generally increasing trend.

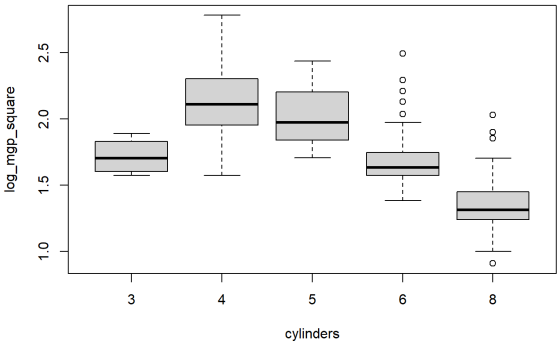


Figure 17: Boxplot showing the distribution of ‘log_mgp_square’ across different cylinder counts.

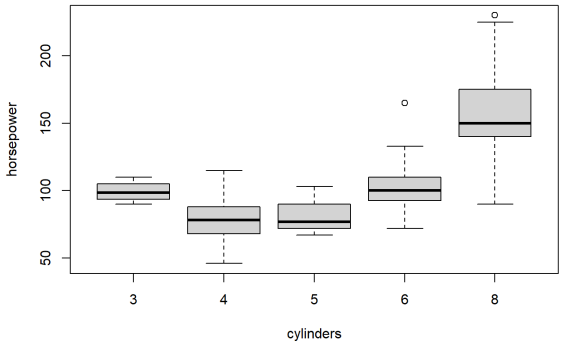


Figure 18: Boxplot showing the distribution of ‘log_mgp_square’ by horsepower categories.

Overall, engines having more cylinders seems to be stronger while consuming less fuel.

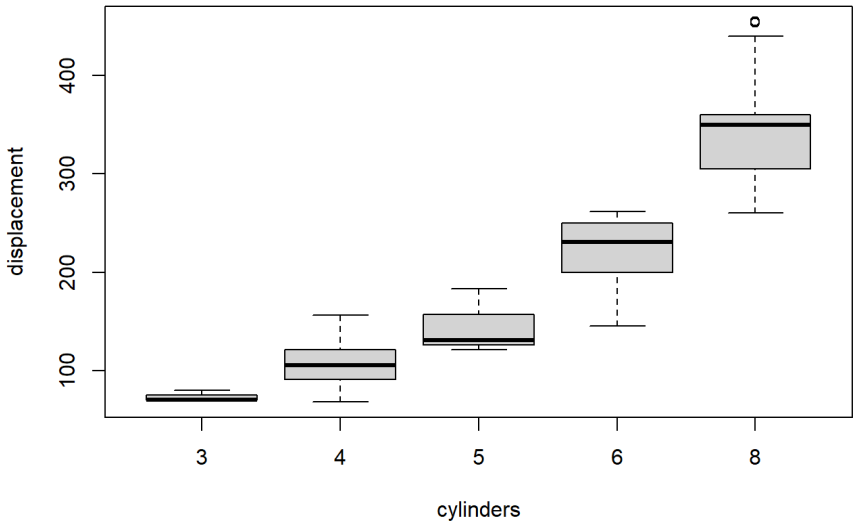


Figure 19: Boxplot showing the relationship between ‘displacement’ and ‘cylinders’.

The engines having more cylinders seem to have larger displacement (engine size).

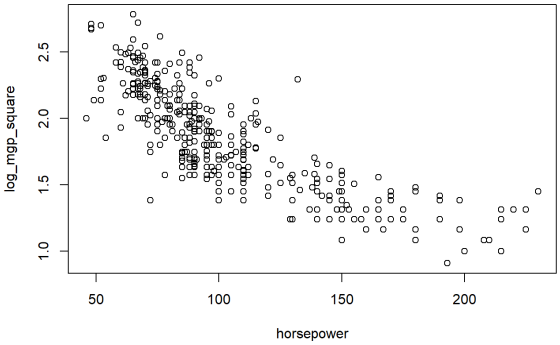


Figure 20: Scatter plot of 'log_mgp_square' vs. 'horsepower'.

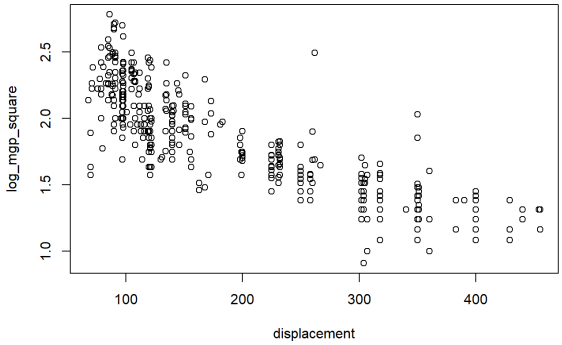


Figure 21: Scatter plot of 'log_mgp_square' vs. 'displacement'.

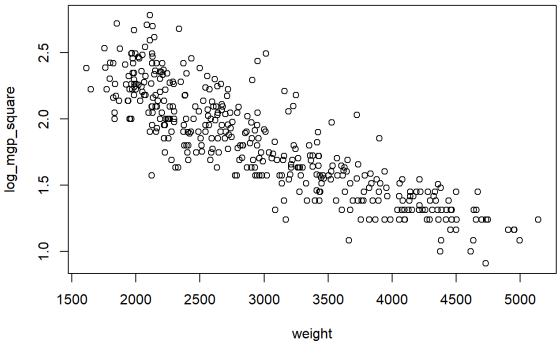


Figure 22: Scatter plot of 'log_mgp_square' vs. 'weight'.

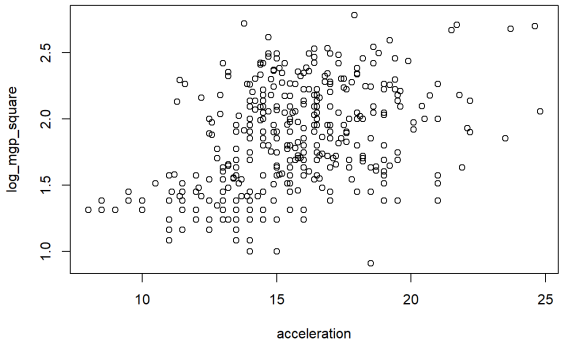


Figure 23: Scatter plot of 'log_mgp_square' vs. 'acceleration'.

Overall, there is a negative linear relation between 'log_mgp_square' and 'horsepower' as well as 'displacement', 'weight' while there is no linear relation between 'log_mgp_square' and 'acceleration'.

II.1.7. Process categorical variables

In the first part of Code Snippet 7, we see that the 'model_year' in this dataset ranges from 1970 to 1982. To make the data easier to work with, we convert these years into a new range from 1 to 13.

The second part of Code Snippet 7 is about creating dummy variables for 'origin' column. We add two new indicators: 'north_america' (1 if the origin is North America, 0 otherwise) and 'europe' (1 if the origin is Europe, 0 otherwise). If both of these indicators are 0, it means the origin is Asia.

II.1.8. Process outliers

As stated in Figure 8 and 9, the boxplot of all variables looks good, with only a few outliers. Moreover, we don't need to remove these outliers because they might be important observations.

II.2. Split data to train and test

Properly splitting the data into training and testing sets is a fundamental step in building a robust and reliable predictive model. It ensures that the model is tested on unseen data, giving a true indication of its performance and generalizability. The code is shown in Code Snippet 8

- Split Ratio: Split the dataset into 80% training and 20% testing sets.
- data_clean has 392 observations.
- data_train has 313 observations and data_test has 79 observations.

II.3. Model Building

II.3.1. Check multicolinearity

The code is in Code Snippet 9.
We consider correlation matrix and fit the model with the train data.

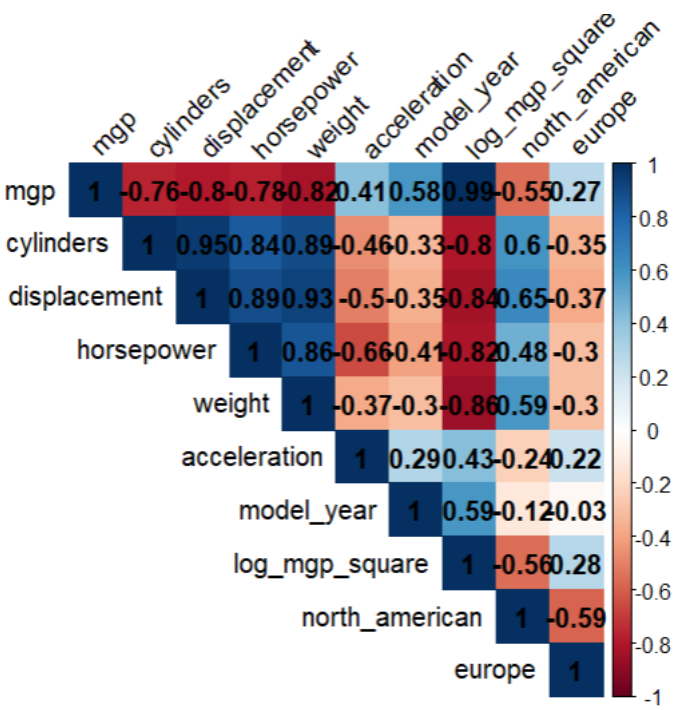


Figure 24: Correlation matrix

```
Call:
lm(formula = log_mgp_square ~ . - mgp, data = data_train)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.44020 -0.09079  0.00657  0.08377  0.45935 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)     2.623e+00  1.051e-01  24.968  < 2e-16 ***
cylinders      -1.702e-02  1.634e-02  -1.042 0.298342    
displacement    6.687e-04  3.882e-04   1.723 0.085975 .  
horsepower     -1.389e-03  6.738e-04  -2.061 0.040157 *  
weight         -3.107e-04  3.219e-05  -9.654  < 2e-16 ***
acceleration    2.182e-03  4.806e-03   0.454 0.650189    
model_year      3.804e-02  2.546e-03  14.943  < 2e-16 ***
north_american -9.847e-02  2.742e-02  -3.591 0.000384 ***
europe          2.340e-02  2.858e-02   0.819 0.413522    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1476 on 304 degrees of freedom
Multiple R-squared:  0.8651,	Adjusted R-squared:  0.8615 
F-statistic: 243.6 on 8 and 304 DF,  p-value: < 2.2e-16
```

Figure 25: First linear regression model

We will check for the presence of multicollinearity among the predictor variables using Variance Inflation Factor (VIF). If VIF > 10, it indicates high multicollinearity, and corrective measures should be taken.

```
   cylinders    displacement      horsepower          weight    acceleration 
    11.067172       22.109484        8.685624       10.245031        2.396089 
  model_year north_american          europe 
    1.269513        2.499680        1.624589 
```

Figure 26: First VIF

After reviewing the correlation matrix and calculating VIF values for each predictor (Figure 26), we observe that the 'displacement' variable has the highest VIF value (22.109484). By making regression between the 'displacement' and the other predictors, the R-squared $= 0.9548$. Moreover, there are strong correlation among displacement towards cylinders, horsepower, weight, acceleration, and north_american. To address multicollinearity and improve the model's stability, we will remove the 'displacement' variable from the model.

```
     cylinders     horsepower         weight   acceleration     model_year 
      5.941555       8.198353       8.462281       2.351212       1.258176 
north_american         europe 
      2.243098       1.624587 
```

Figure 27: Second VIF

Then we consider the vif again in Figure 27, we observe that 'weight' variable has the highest VIF value this time (8.462281). We try removing 'weight' variable. However, after removing 'weight' variable, the adjusted R-squared of the model decreases significantly from 0.8606 to 0.8173. Thus, we try removing predictor having the second highest VIF value (8.198353) which is 'horsepower' variable.

The result of removing 'horsepower' variable is actually way better than the one after removing 'weight' variable through the decrease of residual standard error from 0.1696 to 0.1486 and the climb of adjusted R-squared from 0.8173 to 0.8598. By making regression between the 'horsepower' variable and the other predictors, the R-squared = 0.878. Moreover, there are strong correlation among horsepower towards cylinders, weight, acceleration, model_year, and north_american. Hence, we will remove the 'horsepower' variable from the model instead of 'weight' variable.

```
cylinders         weight   acceleration     model_year north_american
 5.838135       5.206693       1.346716       1.191316       2.194114
   europe
 1.606393
```

Figure 28: Third VIF

Since VIF value of 'cylinders' is 5.692103 in Figure 28, we try removing cylinders. Before removing 'cylinders' variable, the adjusted R-squared = 0.8598. Then, after removing 'cylinders' variable, the model R-squared climbs significantly to 0.8602. Hence, we consider removing this variable. By making regression between the 'cylinders' variable and the other predictors, the R-squared = 0.8287. Moreover, there are strong correlation among cylinders towards weight, acceleration, and north_american. Therefore, we decide to remove 'cylinders' predictor variable.

Finally, we get the final VIF and final model after checking multicolinearity.

```
   weight   acceleration     model_year north_american         europe
 1.778246       1.241902       1.179036       2.147300       1.598528
```

Figure 29: Final VIF

```
Call:
lm(formula = log_mgp_square ~ . - mgp - displacement - horsepower -
    cylinders, data = data_train)

Residuals:
     Min       1Q   Median       3Q      Max
-0.44284 -0.09137  0.00173  0.08662  0.46863

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     2.456e+00  7.367e-02  33.336  < 2e-16 ***
weight         -3.193e-04  1.347e-05 -23.703  < 2e-16 ***
acceleration    6.390e-03  3.476e-03   1.838  0.06699 .
model_year      3.863e-02  2.465e-03  15.674  < 2e-16 ***
north_american -7.698e-02  2.553e-02  -3.015  0.00278 **
europe          2.858e-02  2.848e-02   1.003  0.31647
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1483 on 307 degrees of freedom
Multiple R-squared:  0.8625,    Adjusted R-squared:  0.8602
F-statistic:   385 on 5 and 307 DF,  p-value: < 2.2e-16
```

Figure 30: Model after checking multicolinearity

II.3.2. Variable selection

We decide to use both AIC and BIC Stepwise Regression with both forward and backward stepwise selection with the full model having predictors: weight, acceleration, model_year, north_american and europe. Then, we compare these final models using partial F-test with anova table as the BIC model has 1 variable fewer compared to AIC model. The code is in the Code Snippet 10

```
Analysis of Variance Table

Model 1: log_mgp_square ~ weight + model_year + north_american + acceleration
Model 2: log_mgp_square ~ weight + model_year + north_american
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    308 6.7772
2    309 6.8664 -1  -0.08915 4.0515  0.045 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 31: Anova table of AIC and BIC model

p_value $= 0.045 < 0.05 = \alpha$. Hence, we reject the reduced model which is BIC model and we choose AIC model.

```
Call:
lm(formula = log_mgp_square ~ weight + model_year + north_american +
    acceleration, data = data_train)

Residuals:
     Min       1Q   Median       3Q      Max
-0.45546 -0.08941  0.00204  0.08667  0.47021

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     2.461e+00  7.351e-02  33.480  < 2e-16 ***
weight         -3.184e-04  1.344e-05 -23.690  < 2e-16 ***
model_year      3.827e-02  2.438e-03  15.695  < 2e-16 ***
north_american -9.060e-02  2.163e-02  -4.189 3.66e-05 ***
acceleration    6.917e-03  3.436e-03   2.013    0.045 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1483 on 308 degrees of freedom
Multiple R-squared:  0.862,     Adjusted R-squared:  0.8602
F-statistic:   481 on 4 and 308 DF,  p-value: < 2.2e-16
```

Figure 32: AIC model

II.3.3. Diagnostic

We test Independence, Homoscedasticity and Normality of the model.

- Durbin-Watson test for autocorrelation

```
        Durbin-Watson test

data:  model2
DW = 1.9107, p-value = 0.2139
alternative hypothesis: true autocorrelation is greater than 0
```

Figure 33: Durbin-Watson test

H_0: There is no autocorrelation in the residuals
H_a: There is autocorrelation in the residuals
From the result (p_value = 0.2139), there is no autocorrelation in the residuals.

- Shapiro-Wilk test for normality

```
        Shapiro-Wilk normality test

data:  residuals(model2)
W = 0.99354, p-value = 0.201
```

Figure 34: Shapiro-Wilk Test

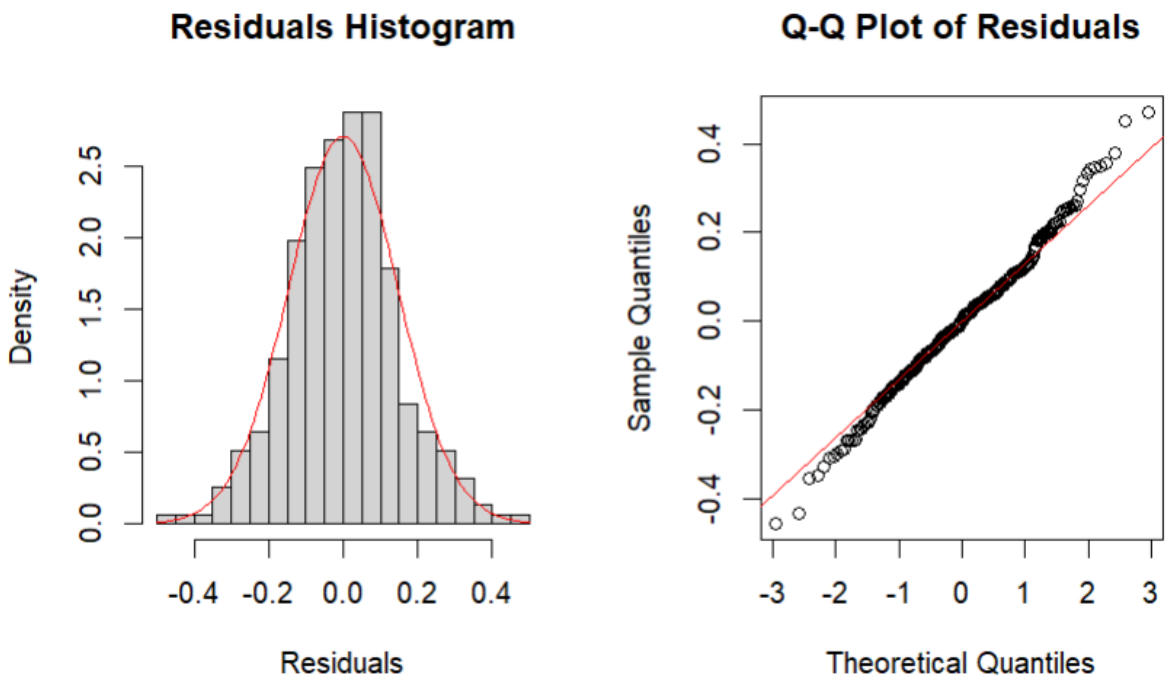


Figure 35: Residuals Histogram and Q-Q Plot of Residuals

H_0: the residuals are normally distributed
H_a: the residuals are not normally distributed

We can see that the scatter points of residuals is quite close to qqline. In addition, p-value $= 0.201 > \alpha = 0.05$, we doesn't have enough evidence to reject H0: $\mu_\epsilon = 0$, which states that residuals of the model are normally distributed.

- Studentized Breusch-Pagan test for heteroscedasticity

```
        studentized Breusch-Pagan test

data:  model3
BP = 12.529, df = 3, p-value = 0.005774
```

Figure 36: Studentized Breusch-Pagan test

H_0: The residuals have constant variance.
H_a: The residuals do not have constant variance.

p-value $= 0.005774 < \alpha = 0.05$, we have enough evidence to reject the null hypothesis, which suggests that the residuals don't have constant variance.

II.3.4. Box-Cox Transformation

As the model has failed the Studentized Breusch-Pagan test for homoscedasticity so we decide to apply box-cox transformation. The code is shown in the Code Snippet 13. Using the $\lambda = 0.5$, we have the model.

```
Call:
lm(formula = (((data_train$log_mgp_square^best_lambda) - 1)/best_lambda) ~
    data_train$weight + data_train$model_year + data_train$north_american +
        data_train$acceleration)

Residuals:
     Min       1Q   Median       3Q      Max
-0.34627 -0.05905  0.00135  0.06233  0.32484

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)              1.176e+00  5.213e-02  22.559  < 2e-16 ***
data_train$weight       -2.447e-04  9.531e-06 -25.675  < 2e-16 ***
data_train$model_year    2.796e-02  1.729e-03  16.170  < 2e-16 ***
data_train$north_american -5.934e-02  1.534e-02  -3.869 0.000133 ***
data_train$acceleration  5.432e-03  2.437e-03   2.229 0.026530 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1052 on 308 degrees of freedom
Multiple R-squared:  0.8753,    Adjusted R-squared:  0.8737
F-statistic: 540.6 on 4 and 308 DF,  p-value: < 2.2e-16
```

Figure 37: Box-cox transformation model

II.4. Model Diagnostic

The code is in the Code Snippet 12

II.4.1. Durbin-Watson test for autocorrelation

```
        Durbin-Watson test

data:  model_cox
DW = 1.9358, p-value = 0.2842
alternative hypothesis: true autocorrelation is greater than 0
```

Figure 38: Durbin-Watson test

H_0: There is no autocorrelation in the residuals
H_a: There is autocorrelation in the residuals
From the result (p_value = 0.2842), there is no autocorrelation in the residuals.

II.4.2. Shapiro-Wilk test for residual normality

```
        Shapiro-Wilk normality test

data:  residuals
W = 0.99342, p-value = 0.1883
```

Figure 39: Shapiro-Wilk test

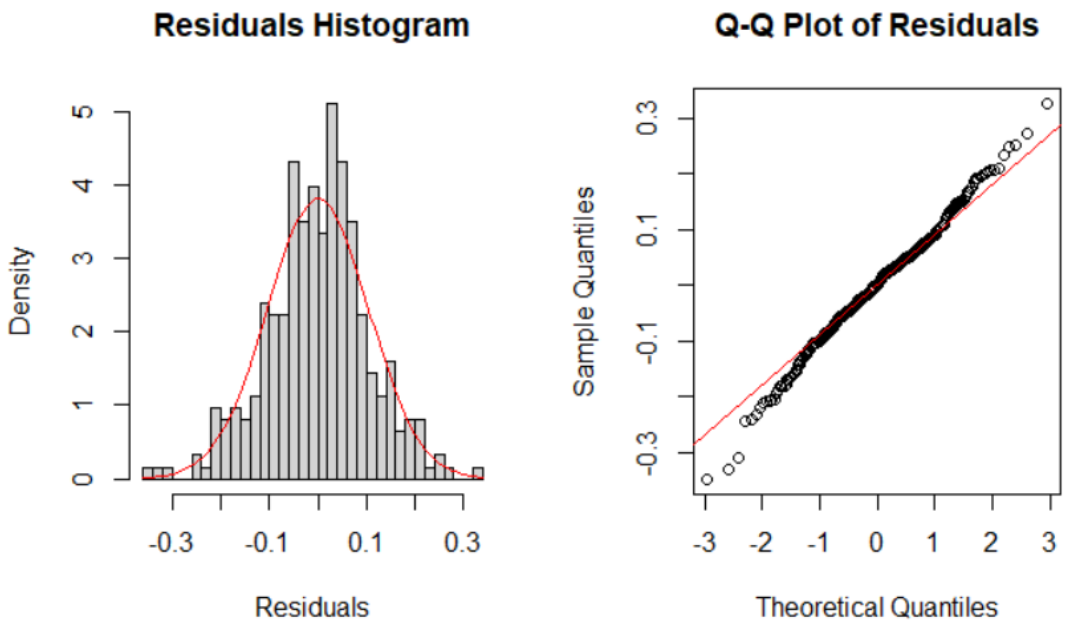


Figure 40: Residuals Histogram and Q-Q Plot of Residuals

H_0: the residuals are normally distributed
H_a: the residuals are not normally distributed

We can see that the scatter points of residuals is quite close to qqline. In addition, p-value $= 0.1883 > \alpha = 0.05$, we don't have enough evidence to reject H0: $\mu_\epsilon = 0$, which states that residuals of model are normally distributed.

II.4.3. Studentized Breusch-Pagan test for heteroscedasticity

```
        studentized Breusch-Pagan test

data:  model_cox
BP = 4.3089, df = 4, p-value = 0.3658
```

Figure 41: Studentized Breusch-Pagan test

H_0: The residuals have constant variance.
H_a: The residuals do not have constant variance.

p-value $= 0.3658 > \alpha = 0.05$, we do not have enough evidence to reject the null hypothesis, which suggests that the residuals have constant variance.

II.5. Model Interpretation

```
Call:
lm(formula = (((data_train$log_mgp_square^best_lambda) - 1)/best_lambda) ~
    data_train$weight + data_train$model_year + data_train$north_american +
        data_train$acceleration)

Residuals:
     Min       1Q   Median       3Q      Max
-0.34627 -0.05905  0.00135  0.06233  0.32484

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)               1.176e+00  5.213e-02  22.559  < 2e-16 ***
data_train$weight        -2.447e-04  9.531e-06 -25.675  < 2e-16 ***
data_train$model_year     2.796e-02  1.729e-03  16.170  < 2e-16 ***
data_train$north_american -5.934e-02  1.534e-02  -3.869 0.000133 ***
data_train$acceleration   5.432e-03  2.437e-03   2.229 0.026530 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1052 on 308 degrees of freedom
Multiple R-squared:  0.8753,    Adjusted R-squared:  0.8737
F-statistic: 540.6 on 4 and 308 DF,  p-value: < 2.2e-16
```

Figure 42: Best model

II.5.1. Quantile

- 25% of residuals are less than -0.05905.
- 50% of residuals are above 0.00135, 50% of residuals are below 0.00135.
- 75% of residuals are less than 0.06233.

II.5.2. Coefficent of predictors

We have $y = \frac{(log\_mgp\_square)^{\lambda_{best}} - 1}{\lambda_{best}}$

Regression model:
$y = \beta_0 + \beta_1 \cdot weight + \beta_2 \cdot model\_year + \beta_3 \cdot north\_american + \beta_4 \cdot acceleration + \epsilon$

- $\hat{\beta}_0$: y = 1.176 when weight = 0, model_year = 0, north_american = 0 and acceleration = 0.

- $\hat{\beta}_1$: For each unit increase in weight, on average, the expected value of y decreases by 2.447×10^{-4}, holding other predictors constant.
- $\hat{\beta}_2$: For each unit increase in model_year, on average, the expected value of y increases by 2.796×10^{-2}, holding other predictors constant.
- $\hat{\beta}_3$: if the engine is in north_american, on average, the expected value of y decreases by 5.934×10^{-2}, holding other predictors constant.
- $\hat{\beta}_4$: For each unit increase in acceleration, on average, the expected value of y increases by 5.432×10^{-3}, holding other predictors constant.

II.5.3. Multiple R-squared and Adjusted R-squared

Multiple R-squared: 0.8753 interprets that 87.53% of the variance in the response variable $\frac{(log\_mgp\_square)^{\lambda_{best}}-1}{\lambda_{best}}$ can be explained by the predictor variables (weight, model_year, north_american, acceleration) in the model. The adjusted R-squared is 0.8737, which is slightly lower than the multiple R-squared. This indicates that the additional predictors of the model are contributing meaningfully to the model's explanatory power.

II.5.4. Residual standard error

Residual standard error: rse = 0.1052 indicates that the model's predictions are, on average, approximately 0.1052 units away from the actual values of $\frac{(log\_mgp\_square)^{\lambda_{best}}-1}{\lambda_{best}}$. Some points are further from the line than this rse, other points are closer to the line than this rse.

II.6. Prediction

We use cross validation k-folds to do the task. The code is shown in the Code Snippet 14 This is the first part of the prediction

	actual_mpg <dbl>	predict_mpg <dbl>
4	16.0	15.52171
5	17.0	15.45044
6	15.0	11.95431
7	14.0	11.90970
8	14.0	12.05444
10	15.0	13.76745
18	21.0	19.79749
21	25.0	20.71650
32	25.0	24.33020
47	22.0	21.54464

1-10 of 79 rows Previous 1

Figure 43: Actual and prediction value of mgp

II.7. Evaluation

The Code Snippet 14 also calculates the rmse and R-squared of the model.
We have:

- RMSE: 2.474305
- R-squared: 0.8916217

The RMSE value is 2.475, which is relatively low compared to the range of the mpg (from 10 to around 40). This means that the model is able to predict the target variable with a high degree of accuracy. The R-squared value of 0.89 indicates that the model is able to explain 89% of the variance in the target variable. This is a good result, as it indicates that the model is able to capture a large proportion of the variation in the target variable.

III. Activity 2: Happiness

In this activity, our main objective is to study the factors that influence the happiness of individuals worldwide. By analyzing these factors, we aim to develop a predictive model that can estimate the happiness score of individuals. This model will provide valuable insights into understanding and predicting happiness levels, which can be beneficial for various fields such as psychology, sociology, and public policy.

III.1. Dataset description

Link to the dataset: World Happiness Report Dataset
The World Happiness Report dataset is sourced from Kaggle platform, a popular online community for data scientists and machine learning practitioners. The report measures the happiness of around 160 countries around the world in each year (from 2005 to 2020) by calculating the happiness score based on various factors: economic production, social support, life expectancy, freedom, etc. The dataset consists of 1949 rows on 11 variables. Below is the description of the variables in the dataset:

Variable	**Type**	**Description**
Country name	Multi-valued discrete	The name of the country where the survey was conducted. About 166 countries participate in the survey.
Year	Multi-valued discrete	The year of the survey, ranging from 2005 to 2020.
Life Ladder	Continuous	The happiness score of the country, ranging from 0 to 10. The question is: "Imagine a ladder, with steps numbered from 0 at the bottom to 10 at the top. The top of the ladder represents the best possible life for you and the bottom of the ladder represents the worst possible life for you. On which step of the ladder would you say you personally feel you stand at this time?"
Log GDP per capita	Continuous	The logarithm of the gross domestic product per capita in the year.
Social support	Continuous	The national average of the binary responses (either 0 or 1) to the question: "If you were in trouble, do you have relatives or friends you can count on to help you whenever you need them, or not?"
Healthy life expectancy at birth	Continuous	The number of years a newborn infant could expect to live in full health.

Variable	Type	Description
Freedom to make life choices	Continuous	The national average of the binary responses (either 0 or 1) to the question: "Are you satisfied or dissatisfied with your freedom to choose what you do with your life?"
Generosity	Continuous	The residual of regressing national average of response to the question: "Have you donated money to a charity in the past month?" on GDP per capita (capita is the Latin word of 'person').
Perceptions of corruption	Continuous	The national average of the binary responses (either 0 or 1) to the question: "Is corruption widespread throughout the government or not?"
Positive affect	Continuous	The average of three positive affect measures in the Gallup World Poll: happiness, laughter, and enjoyment.
Negative affect	Continuous	The average of three negative affect measures in the Gallup World Poll: worry, sadness, and anger.

The data has at least 1 qualitative (discrete) variable (country name) and at least 3 quantitative (continuous) variables as requirement.

III.2. Import and preprocess dataset

The dataset includes 2 files which are *"world-happiness-report-2021.csv"* and *"world-happiness-report.csv"*. In our activity, we just use the *"world-happiness-report.csv"* file for analysis. However, the *"world-happiness-report-2021.csv"* file is still used for categorizing data (which will be explained in the next part).

III.2.1. Process categorical data

In the data, country is a categorical data which has about 160 unique values. We use the file *"world-happiness-report-2021.csv"* to categorize the country into 11 regions based on the geolocation, since the file *"world-happiness-report.csv"* does not contain the region columns. The regions are:

- Western Europe
- North America and ANZ
- Middle East and North Africa
- Latin America and Caribbean
- Central and Eastern Europe

- East Asia
- Southeast Asia
- South Asia
- Sub-Saharan Africa
- Commonwealth of Independent States
- Southern Asia

Countries in each region often have the same characteristics in various aspects culture, economy, etc. So this categorization is reasonable for analyzing and we decided to add the region column for the data.
The code for categorizing country into region is in the Code Snippet 15. This code outputs the modified dataset into the file *"world-happiness-report-with-regions.csv"* Some of the country are not existed in the report 2021 so we fill the region column for those country using Excel·

III.2.2. Process missing value

When processing missing values, we have to assure that the data removed is not over 10%. We proposed the way to handle, that is, for each country in the dataset, if all of its data is missing, we will eliminate those incomplete observations. Otherwise, we will replace missing values with the average value of non-missing observations in the country's data. We used Excel to eliminate the incomplete observations while with replacing mean values, we used the Code Snippet 16

Before replacing using mean values, we import the dataset *"world-happiness-report-with-regions.csv"* with the Code Snippet 17

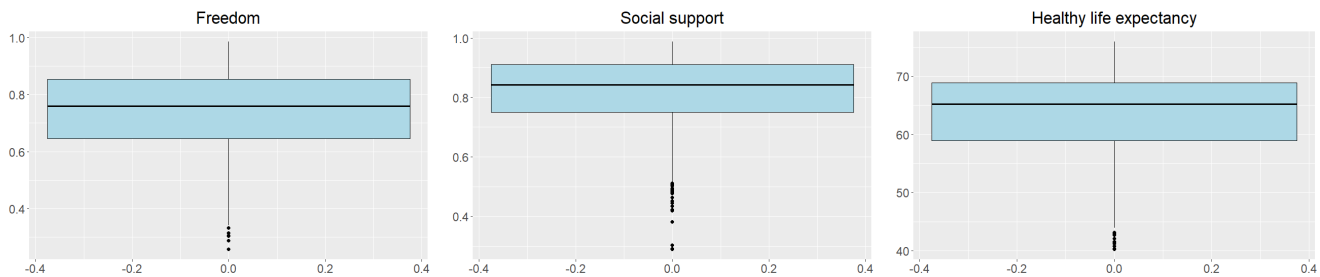
```
[1] 1791   12
gropd_df [1,791 × 12] (S3: grouped_df/tbl_df/tbl/data.frame)
 $ Country                        : chr [1:1791] "Afghanistan"
"Afghanistan" "Afghanistan" "Afghanistan" ...
 $ Region                         : chr [1:1791] "South Asia"
"South Asia" "South Asia" "South Asia" ...
 $ year                           : int [1:1791] 2008 2009 2010
2011 2012 2013 2014 2015 2016 2017 ...
 $ Life.Ladder                    : num [1:1791] 3.72 4.4 4.76
3.83 3.78 ...
 $ Log.GDP.per.capita             : num [1:1791] 7.37 7.54 7.65
7.62 7.71 ...
 $ Social.support                 : num [1:1791] 0.451 0.552
0.539 0.521 0.521 0.484 0.526 0.529 0.559 0.491 ...
 $ Healthy.life.expectancy.at.birth: num [1:1791] 50.8 51.2 51.6
51.9 52.2 ...
 $ Freedom.to.make.life.choices    : num [1:1791] 0.718 0.679
0.6 0.496 0.531 0.578 0.509 0.389 0.523 0.427 ...
 $ Generosity                     : num [1:1791] 0.168 0.19
0.121 0.162 0.236 0.061 0.104 0.08 0.042 -0.121 ...
 $ Perceptions.of.corruption      : num [1:1791] 0.882 0.85
0.707 0.731 0.776 0.823 0.871 0.881 0.793 0.954 ...
 $ Positive.affect                : num [1:1791] 0.518 0.584
0.618 0.611 0.71 0.621 0.532 0.554 0.565 0.496 ...
 $ Negative.affect                : num [1:1791] 0.258 0.237
0.275 0.267 0.268 0.273 0.375 0.339 0.348 0.371 ...
```

Figure 44: Structure of world happiness report dataset.

Figure 44 shows the structure of the original dataset, with 1791 observations and 12 variables.

III.3. Descriptive Statistics

III.3.1. Visualization



From the data visualization above, we can observe that many columns exist outliers

III.3.2. Process outliers

In the Code Snippet 19, we show the process of removing outliers. After processing, the dataset has 1674 observations. We need to accept removing about 7% from the original data.

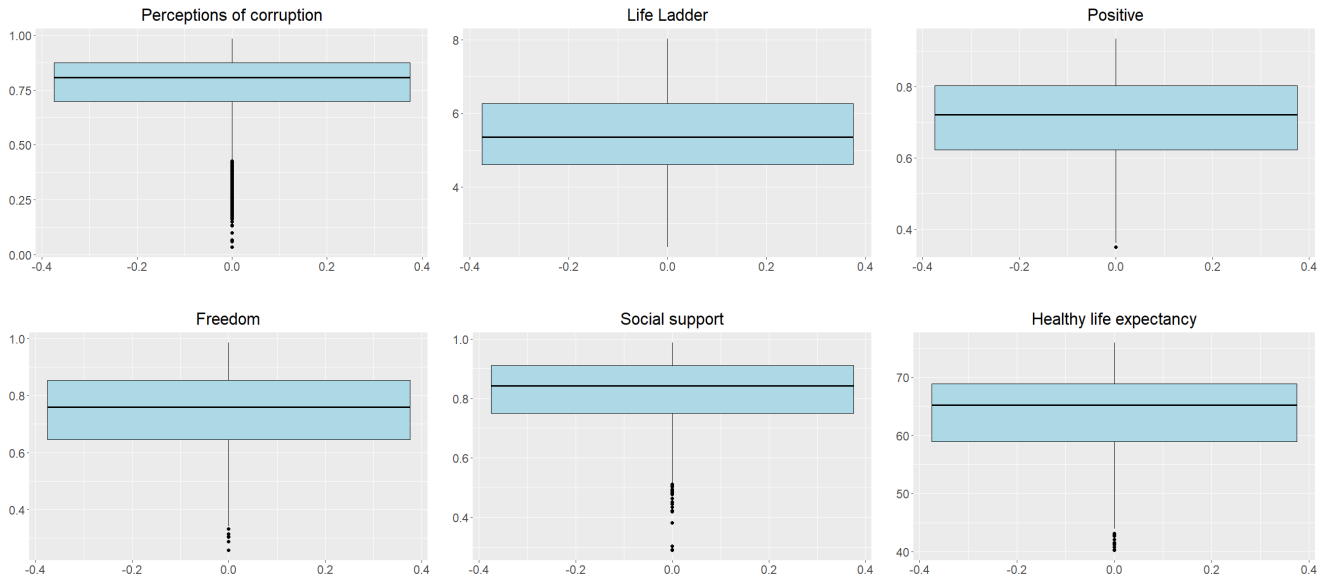


Figure 45: Boxplot of 9 variables in the Happiness dataset

III.3.3. Summarize data

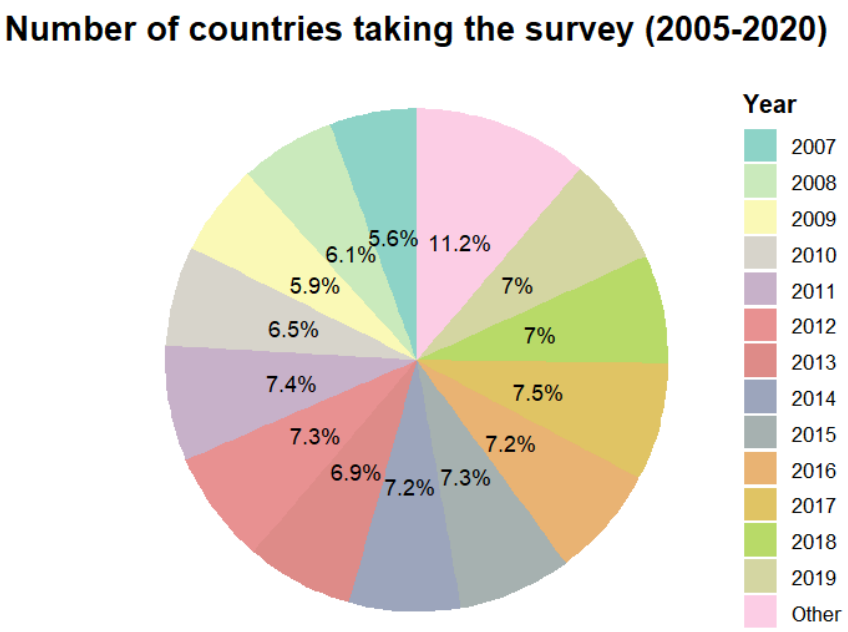


Figure 46: Number of countries taking the survey from 2005 to 2020

The pie chart above shows an even distribution of data from 2007 to 2019, showing the consistency in the number of countries participating in the survey.

- The ladder score distribution shows that the score is approximately normally distributed.

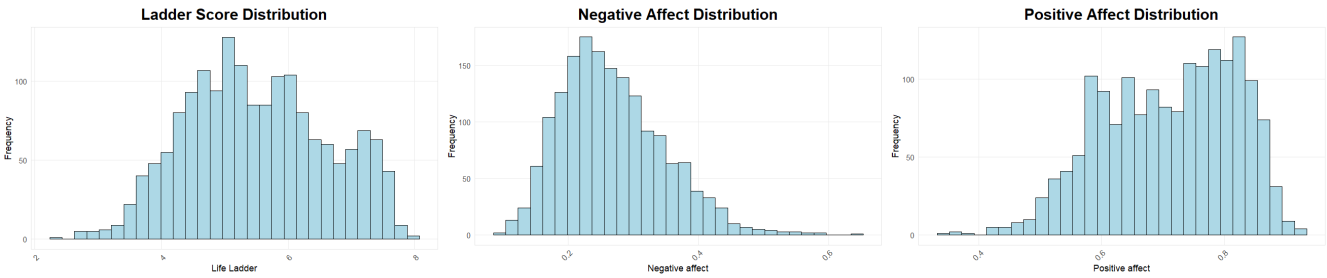


Figure 47: Ladder score, Negative affect, and Positive affect distribution

- The distribution of negative affect factor is right-skewed, with the majority of data lying between 0.2 and 0.4
- The distribution of positive affect factor is left-skewed, with the majority of data lying between 0.6 and 0.8

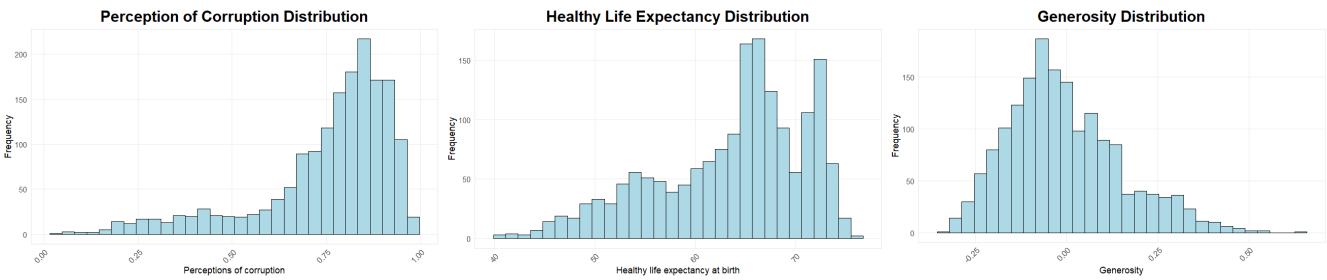


Figure 48: Corruption, Life Expectancy, and Generosity distribution

- The distribution of corruption is highly right-skewed with the majority of data lying between 0.8 and 0.9, showing that most of the answers agree to the response: "The corruption is widespread throughout the government"
- The healthy life expectancy distribution is also right-skewed with the majority of data lying between 65 and 72
- The generosity distribution is left-skewed, with the majority of data lying between -0.15 and 0.15. The residual is very close to 0, suggesting that the actual national average response to the question about charity donations deviates only slightly from the values predicted by GDP per capita.

- The distribution of freedom factor is right-skewed with the majority of data lying between 0.6 to 0.95, indicating that most of the countries' responses to the question are satisfaction.
- The distribution of social support factor is right-skewed with the majority of data lying between 0.75 to 0.95, indicating that most people get help whenever they are in trouble

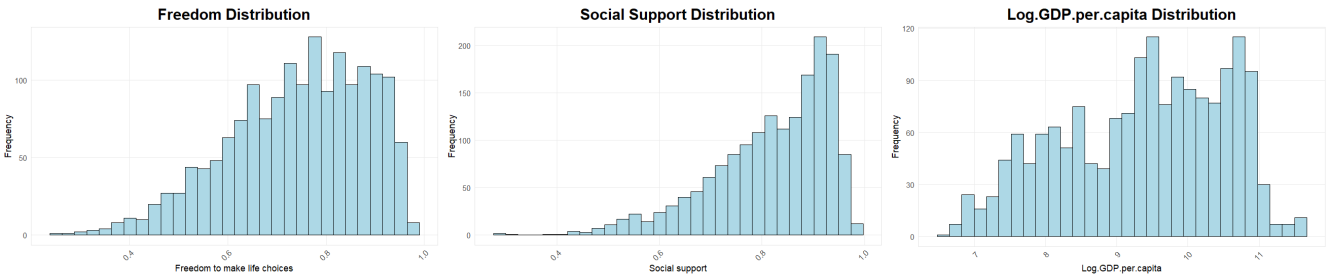


Figure 49: Freedom, Social support, and Log GDP distribution

- The distribution of the Log GDP per person is slightly right-skewed, most of them lying between 9 and 11.

III.3.4. Relationship between life ladder and predictors and between predictors

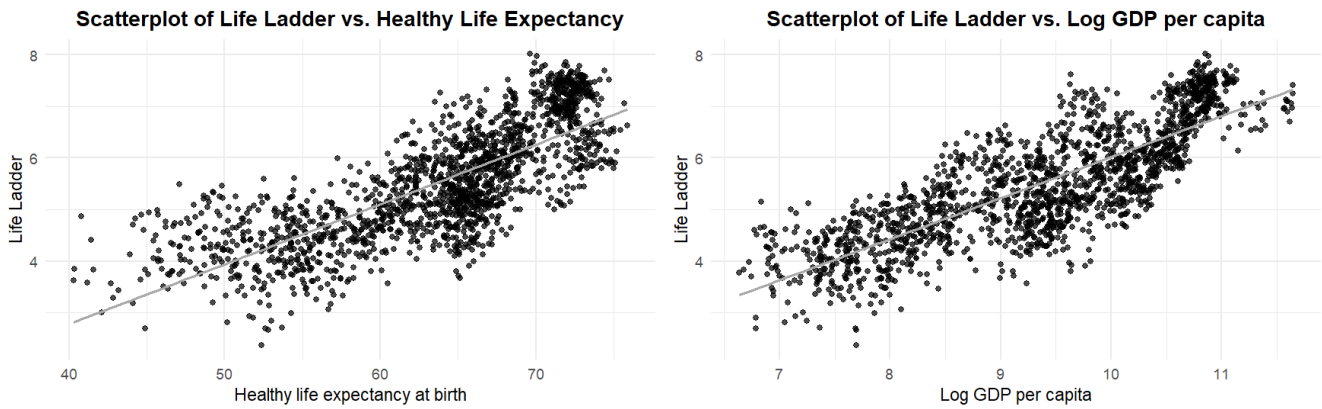


Figure 50: Relationship between life ladder and life expectancy and log GDP

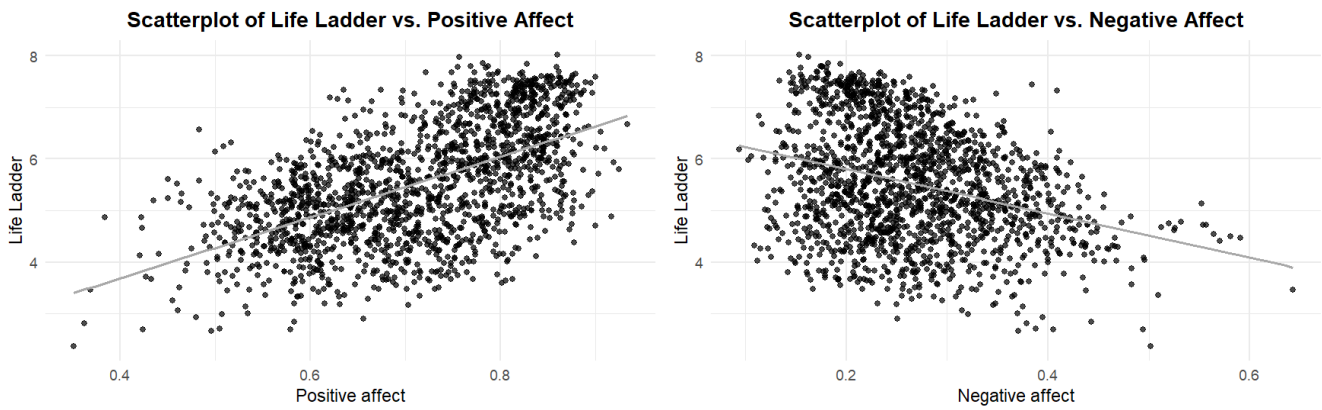


Figure 51: Relationship of life ladder versus positive affect and negative affect

From the scatterplot between life ladder and 8 other variables, we can imply that:

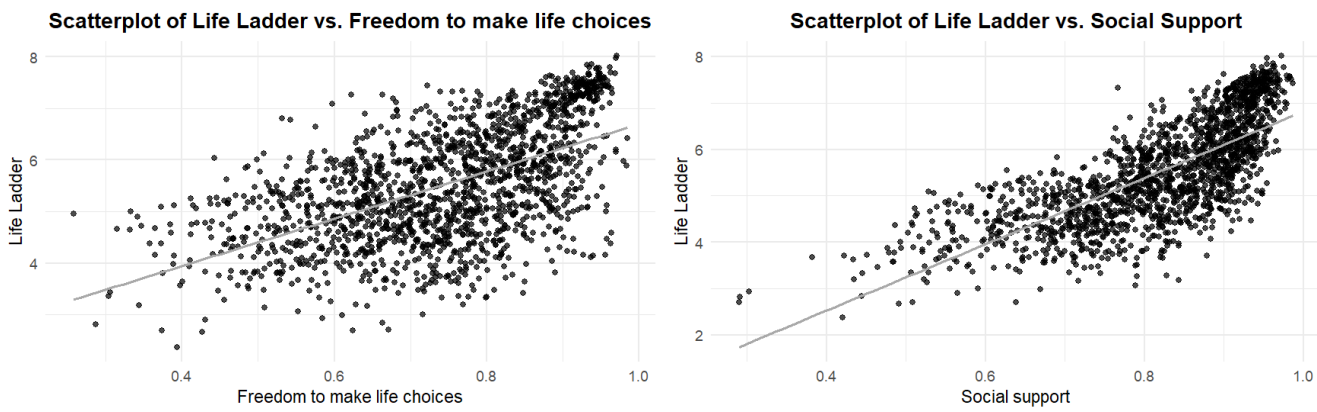


Figure 52: Relationship of life ladder versus freedom and social support

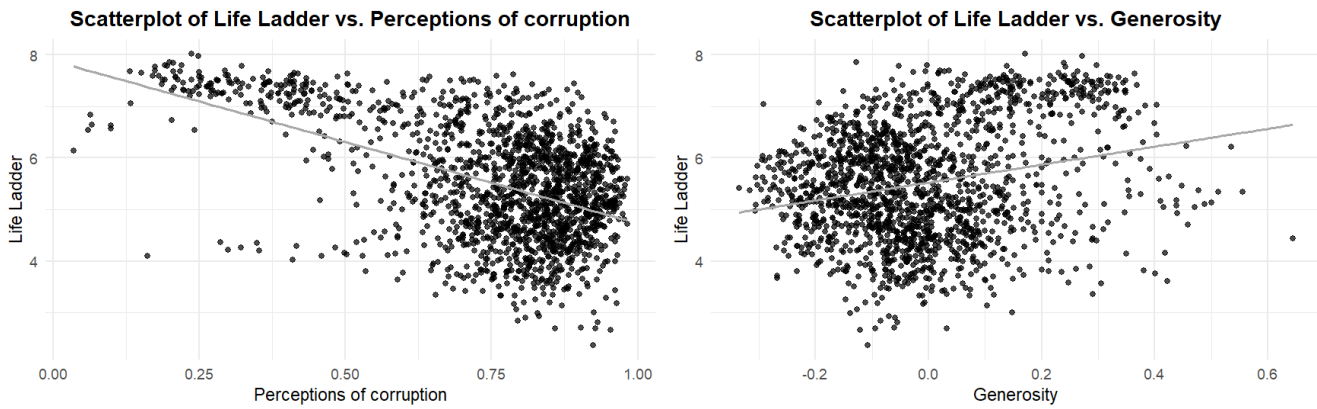


Figure 53: Relationship of life ladder versus corruption and generosity

- Generosity and negative affect seems to not have a linear relationship with the happiness score.
- Negative affect and perceptions of corruption has an inverse-proportional relationship with life ladder
- Healthy life expectancy, Log GDP, and social support has a strong, proportional relationship with life ladder

The boxplot of ladder score shows that this happiness score varies depend on the region. North America and Western Europe has the highest score (higher than 6.5) while South Asia and Sub-Saharan Africa has the lowest ladder score (lower than 5.0)

III.4. Model Building

Before stepping into building the model, we will split the data into training and testing sets. The training set is used to build the model and the testing set is used to evaluate the model. The code is shown in the Code Snippet 18

- Split ratio: Split the dataset into 80% training and 20% testing sets.

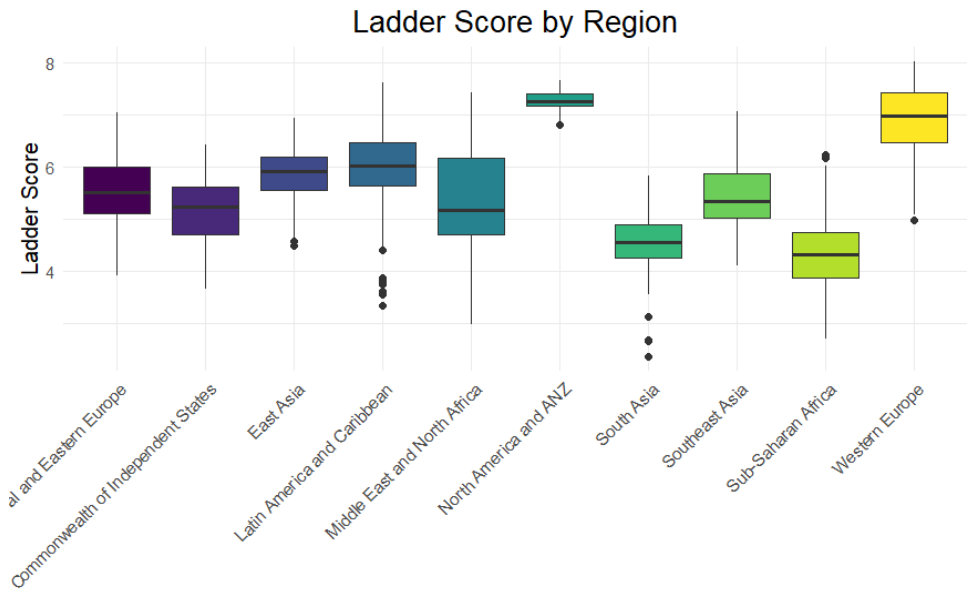


Figure 54: Ladder score in different regions

- data_clean has 1674 observations
- data_train has 1339 observations and data_test has 335 observations.

We also need to add Region as a dummy variable to our dataset. The code to do this is shown in the Code Snippet 20

III.4.1. Check multicollearity

The code is in Code Snippet 21.
We first take a look at the correlation matrix and fit the first model with all the variables.

```
                                   year                        Log.GDP.per.capita
                               1.366372                                  6.269113
                         Social.support          Healthy.life.expectancy.at.birth
                               2.712131                                  7.286729
             Freedom.to.make.life.choices                                Generosity
                               2.312388                                  1.709311
                Perceptions.of.corruption                           Positive.affect
                               1.961748                                  3.192274
                        Negative.affect `RegionCommonwealth of Independent States`
                               1.842334                                  1.860380
                     `RegionEast Asia`                        `RegionHorn Africa`
                               1.285442                                  1.105509
     `RegionLatin America and Caribbean`       `RegionMiddle East and North Africa`
                               3.240439                                  1.847460
           `RegionNorth America and ANZ`                        `RegionSouth Asia`
                               1.823036                                  1.869600
                  `RegionSoutheast Asia`                 `RegionSub-Saharan Africa`
                               2.035226                                  6.376271
                  `RegionWestern Europe`
                               3.043032
[1] 7.286729
```

Figure 55: VIF values

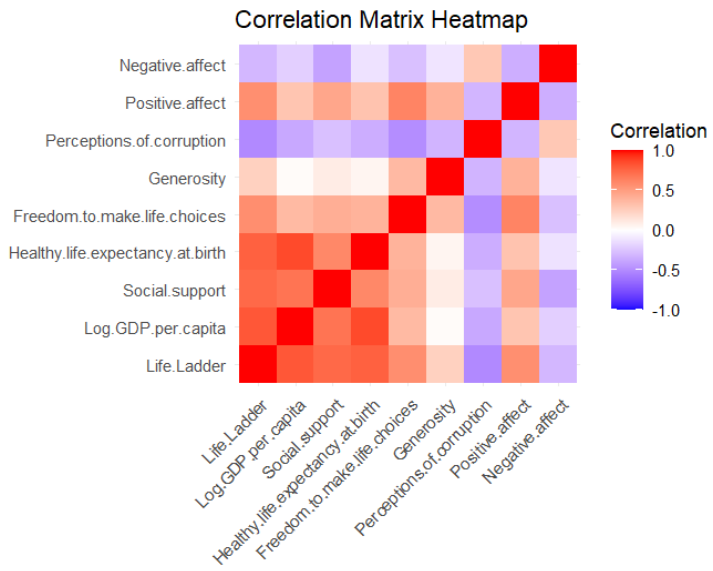


Figure 56: Correlation Matrix

There are lots of unnecessary variables here, so we check the multicollinearity.
The highest VIF value is 7.28 and it belongs to the healthy life expectancy. We next check if there's actually a relationship between this life expectancy and the other variables.

```
Call:
lm(formula = Life.Ladder ~ ., data = data_train)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.54609 -0.26527  0.01421  0.27556  1.27508 

Coefficients:
                                          Estimate Std. Error t value Pr(>|t|)    
(Intercept)                               9.467665   6.445535   1.469 0.142106    
year                                     -0.005850   0.003250  -1.800 0.072100 .  
Log.GDP.per.capita                        0.390893   0.026067  14.996  < 2e-16 ***
Social.support                            1.879788   0.171035  10.991  < 2e-16 ***
Healthy.life.expectancy.at.birth          0.025605   0.004267   6.001 2.53e-09 ***
Freedom.to.make.life.choices              0.871491   0.128552   6.779 1.82e-11 ***
Generosity                                0.638830   0.099417   6.426 1.83e-10 ***
Perceptions.of.corruption                -0.598102   0.091900  -6.508 1.08e-10 ***
Positive.affect                           1.154263   0.195444   5.906 4.46e-09 ***
Negative.affect                          -0.539353   0.199297  -2.706 0.006892 ** 
`RegionCommonwealth of Independent States` 0.002769   0.055688   0.050 0.960345    
`RegionEast Asia`                        -0.217792   0.075557  -2.882 0.004010 ** 
`RegionHorn Africa`                       0.266456   0.226358   1.177 0.239352    
`RegionLatin America and Caribbean`       0.429178   0.057717   7.436 1.86e-13 ***
`RegionMiddle East and North Africa`      0.168307   0.056772   2.965 0.003085 ** 
`RegionNorth America and ANZ`             0.286987   0.087097   3.295 0.001010 ** 
`RegionSouth Asia`                        0.207716   0.080946   2.566 0.010395 *  
`RegionSoutheast Asia`                   -0.209177   0.069907  -2.992 0.002821 ** 
`RegionSub-Saharan Africa`                0.076437   0.072389   1.056 0.291197    
`RegionWestern Europe`                    0.206224   0.057381   3.594 0.000338 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4299 on 1319 degrees of freedom
Multiple R-squared:  0.8519,	Adjusted R-squared:  0.8498 
F-statistic: 399.4 on 19 and 1319 DF,  p-value: < 2.2e-16
```

Figure 57: Fit the first linear regression model

```
Call:
lm(formula = Healthy.life.expectancy.at.birth ~ . - Life.Ladder,
    data = data_train)

Residuals:
     Min       1Q   Median       3Q      Max
-17.6717  -1.5824   0.1499   1.8277   7.1608

Coefficients:
                                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                             -458.03803   39.62337 -11.560  < 2e-16 ***
year                                       0.24681    0.01983  12.443  < 2e-16 ***
Log.GDP.per.capita                         2.86942    0.14846  19.328  < 2e-16 ***
Social.support                            -1.89476    1.10213  -1.719  0.08582 .
Freedom.to.make.life.choices               2.23978    0.82700   2.708  0.00685 **
Generosity                                -1.26452    0.64040  -1.975  0.04852 *
Perceptions.of.corruption                 -1.63149    0.59115  -2.760  0.00586 **
Positive.affect                            0.98163    1.26054   0.779  0.43627
Negative.affect                            0.58453    1.28558   0.455  0.64941
`RegionCommonwealth of Independent States`  -1.74202    0.35603  -4.893 1.12e-06 ***
`RegionEast Asia`                          2.02194    0.48424   4.176 3.17e-05 ***
`RegionHorn Africa`                       -6.72057    1.44849  -4.640 3.84e-06 ***
`RegionLatin America and Caribbean`        0.36330    0.37220   0.976  0.32920
`RegionMiddle East and North Africa`      -2.03437    0.36193  -5.621 2.32e-08 ***
`RegionNorth America and ANZ`              1.44666    0.56046   2.581  0.00995 **
`RegionSouth Asia`                        -2.85113    0.51626  -5.523 4.02e-08 ***
`RegionSoutheast Asia`                    -1.10760    0.44994  -2.462  0.01396 *
`RegionSub-Saharan Africa`                -8.66151    0.40155 -21.570  < 2e-16 ***
`RegionWestern Europe`                     2.21564    0.36511   6.068 1.69e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.773 on 1320 degrees of freedom
Multiple R-squared:  0.8628,    Adjusted R-squared:  0.8609
F-statistic:   461 on 18 and 1320 DF,  p-value: < 2.2e-16
```

Figure 58: Test the relationship between life expectancy and the other variables

Observe that the R^2 value is around 0.863, which shows a strong relationship between life expectancy and the other variables. Therefore, we decide to remove it from the model and check the multicollinearity again.

```
                                     year                        Log.GDP.per.capita
                                 1.222922                                  4.886235
                           Social.support                Freedom.to.make.life.choices
                                 2.706072                                  2.299610
                               Generosity                   Perceptions.of.corruption
                                 1.704277                                  1.950493
                          Positive.affect                              Negative.affect
                                 3.190808                                  1.842045
`RegionCommonwealth of Independent States`                         `RegionEast Asia`
                                 1.827241                                  1.268685
                      `RegionHorn Africa`        `RegionLatin America and Caribbean`
                                 1.087770                                  3.238102
     `RegionMiddle East and North Africa`              `RegionNorth America and ANZ`
                                 1.804275                                  1.813881
                       `RegionSouth Asia`                     `RegionSoutheast Asia`
                                 1.827377                                  2.025926
                `RegionSub-Saharan Africa`                      `RegionWestern Europe`
                                 4.714503                                  2.960442
```

Figure 59: VIF values

All the VIF values are less than 5, which indicates that there should be no multicollinearity problem in the model. All the codes test multicollinearity is demonstrated in the Code Snippet 21

III.4.2. Variable selection

We decide to use both AIC and BIC Stepwise Regression with both forward and backward stepwise selection with the full model having all the predictors except for the healthy life expectancy. We will do the F-partial test to decide which model to keep. The code to do this in the Code Snippet 22

```
lm(formula = Life.Ladder ~ Log.GDP.per.capita + Social.support +
    Freedom.to.make.life.choices + Generosity + Perceptions.of.corruption +
    Positive.affect + Negative.affect + `RegionLatin America and Caribbean` +
    `RegionSoutheast Asia` + `RegionSub-Saharan Africa` + `RegionWestern Europe` +
    `RegionNorth America and ANZ` + `RegionEast Asia` + `RegionMiddle East and North Africa` +
    `RegionSouth Asia`, data = data_train)

Residuals:
     Min       1Q   Median       3Q      Max
-1.46682 -0.26616  0.00883  0.26480  1.32131

Coefficients:
                                    Estimate Std. Error t value Pr(>|t|)
(Intercept)                         -1.38049    0.23501  -5.874 5.37e-09 ***
Log.GDP.per.capita                   0.46769    0.02204  21.223  < 2e-16 ***
Social.support                       1.83372    0.17277  10.614  < 2e-16 ***
Freedom.to.make.life.choices         0.92846    0.12447   7.459 1.57e-13 ***
Generosity                           0.61190    0.09987   6.127 1.18e-09 ***
Perceptions.of.corruption           -0.63428    0.09145  -6.935 6.31e-12 ***
Positive.affect                      1.18792    0.19745   6.016 2.31e-09 ***
Negative.affect                     -0.49809    0.19121  -2.605  0.00929 **
`RegionLatin America and Caribbean`  0.45262    0.05286   8.562  < 2e-16 ***
`RegionSoutheast Asia`              -0.22251    0.06534  -3.406  0.00068 ***
`RegionSub-Saharan Africa`          -0.12475    0.05210  -2.394  0.01679 *
`RegionWestern Europe`               0.27499    0.05389   5.103 3.83e-07 ***
`RegionNorth America and ANZ`        0.33507    0.08571   3.909 9.72e-05 ***
`RegionEast Asia`                   -0.15126    0.07280  -2.078  0.03791 *
`RegionMiddle East and North Africa` 0.13077    0.05162   2.533  0.01142 *
`RegionSouth Asia`                   0.15420    0.07384   2.088  0.03696 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4352 on 1323 degrees of freedom
Multiple R-squared:  0.8478,    Adjusted R-squared:  0.8461
F-statistic: 491.3 on 15 and 1323 DF,  p-value: < 2.2e-16
```

Figure 60: Model after doing AIC algorithm

```
Call:
lm(formula = Life.Ladder ~ Log.GDP.per.capita + Social.support +
    Freedom.to.make.life.choices + Generosity + Perceptions.of.corruption +
    Positive.affect + `RegionLatin America and Caribbean` + `RegionSoutheast Asia` +
    `RegionSub-Saharan Africa` + `RegionWestern Europe` + `RegionNorth America and ANZ`,
    data = data_train)

Residuals:
    Min      1Q  Median      3Q     Max
-1.3623 -0.2827  0.0097  0.2839  1.3254

Coefficients:
                                    Estimate Std. Error t value Pr(>|t|)
(Intercept)                         -1.31133    0.20198  -6.492 1.19e-10 ***
Log.GDP.per.capita                   0.44785    0.02121  21.111  < 2e-16 ***
Social.support                       1.85181    0.15865  11.672  < 2e-16 ***
Freedom.to.make.life.choices         0.91510    0.12309   7.434 1.88e-13 ***
Generosity                           0.61790    0.09889   6.248 5.58e-10 ***
Perceptions.of.corruption           -0.71469    0.08744  -8.174 6.90e-16 ***
Positive.affect                      1.30962    0.18800   6.966 5.12e-12 ***
`RegionLatin America and Caribbean`  0.38556    0.04633   8.322  < 2e-16 ***
`RegionSoutheast Asia`              -0.27127    0.06074  -4.466 8.65e-06 ***
`RegionSub-Saharan Africa`          -0.19232    0.04454  -4.317 1.70e-05 ***
`RegionWestern Europe`               0.24615    0.04977   4.946 8.54e-07 ***
`RegionNorth America and ANZ`        0.29151    0.08202   3.554 0.000392 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4376 on 1327 degrees of freedom
Multiple R-squared:  0.8457,    Adjusted R-squared:  0.8444
F-statistic:   661 on 11 and 1327 DF,  p-value: < 2.2e-16
```

Figure 61: Model after doing BIC algorithm

```
  Res.Df    RSS Df Sum of Sq      F   Pr(>F)
1   1323 250.63
2   1327 254.12 -4   -3.4964 4.6142 0.001057 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 62: F-partial test

From the F-partial test, Since P-value < 0.05, we reject H_0 and conclude that the model with more variables is better

III.4.3. Diagnostic

We test the homoscedasticity and normality of the model. The code to do this is in the Code Snippet 24

```
        Shapiro-Wilk normality test

data:  residuals(model2)
W = 0.99752, p-value = 0.037
```

Figure 63: Shapiro-Wilk normality test

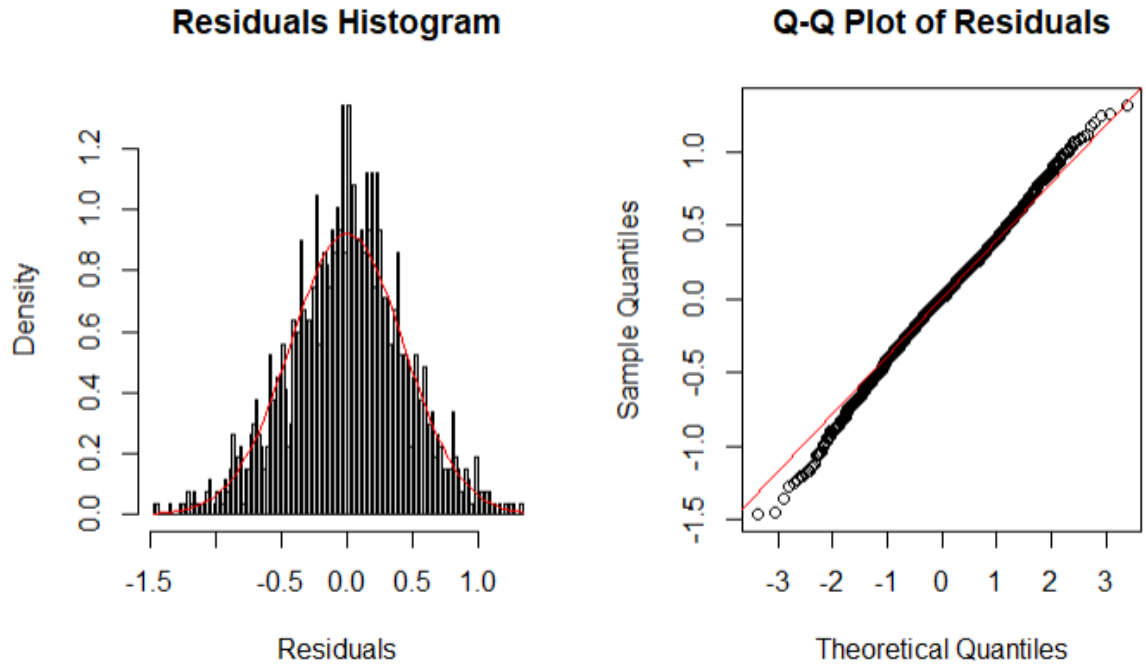


Figure 64: Residuals Histogram and Q-Q plot of Residuals

The p-value is 0.037, so we reject H_0 at 5% level, conclude that the distribution of residuals is not normal.

```
        studentized Breusch-Pagan test

data:  model2
BP = 106.59, df = 15, p-value = 7.268e-16
```

Figure 65: Studentized Breusch-Pagan test

The p-value is very small, so we reject H_0 at 5% level, conclude that the residuals don't have constant variance.
We fail on both the normality test and homoscedasticity test, which means our model fails to satisfy $\epsilon \in N(0, \sigma^2)$. We need to improve our model

III.4.4. Box-cox transformation

As the model has failed the Studentized Breusch-Pagan test and Shapiro-Wilk test, we decide to apply Box-cox transformation. We also try several combinations of dummy variables as well as applying Box-cox transformation and hope to trade off some very small percentages of R^2 against the normality and homoscedasticity of the model. One of our best result so far is applying Box-cox transformation and keep only the RegionLatin

America and Caribbean dummy variable. The code is shown in the Code Snippet 23. Using the $\lambda = 1.8$, we have the model.

```
Call:
lm(formula = ((((Life.Ladder)^best_lambda) - 1)/best_lambda) ~
    Log.GDP.per.capita + Social.support + Freedom.to.make.life.choices +
        Generosity + Perceptions.of.corruption + Positive.affect +
        `RegionLatin America and Caribbean`, data = data_train)

Residuals:
    Min      1Q  Median      3Q     Max
-5.3389 -1.2739  0.0052  1.1280  5.6704

Coefficients:
                                    Estimate Std. Error t value Pr(>|t|)
(Intercept)                        -15.66470    0.67964 -23.048  < 2e-16 ***
Log.GDP.per.capita                   2.08278    0.06374  32.675  < 2e-16 ***
Social.support                       7.14593    0.64350  11.105  < 2e-16 ***
Freedom.to.make.life.choices         2.91299    0.49501   5.885 5.04e-09 ***
Generosity                           3.23274    0.37373   8.650  < 2e-16 ***
Perceptions.of.corruption           -4.67932    0.33311 -14.047  < 2e-16 ***
Positive.affect                      4.56912    0.70781   6.455 1.51e-10 ***
`RegionLatin America and Caribbean`  1.63043    0.15673  10.403  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.782 on 1331 degrees of freedom
Multiple R-squared:  0.8362,    Adjusted R-squared:  0.8354
F-statistic: 970.9 on 7 and 1331 DF,  p-value: < 2.2e-16
```

Figure 66: Box-cox transformation model

III.5. Model Diagnostic

III.5.1. Durbin-Watson test for autocorrelation

```
lag Autocorrelation D-W Statistic p-value
  1      0.03448078      1.929832   0.162
Alternative hypothesis: rho != 0
```

Figure 67: Durbin-Watson test

H_0: There is no autocorrelation in the residuals
H_a: There is autocorrelation in the residuals
From the result (p_value = 0.162), there is no autocorrelation in the residuals, indicate that the residuals are independent.

III.5.2. Shapiro-Wilk test for residual normality

```
        Shapiro-Wilk normality test

data:  residuals(model_cox)
W = 0.99892, p-value = 0.6128
```

Figure 68: Shapiro-Wilk test

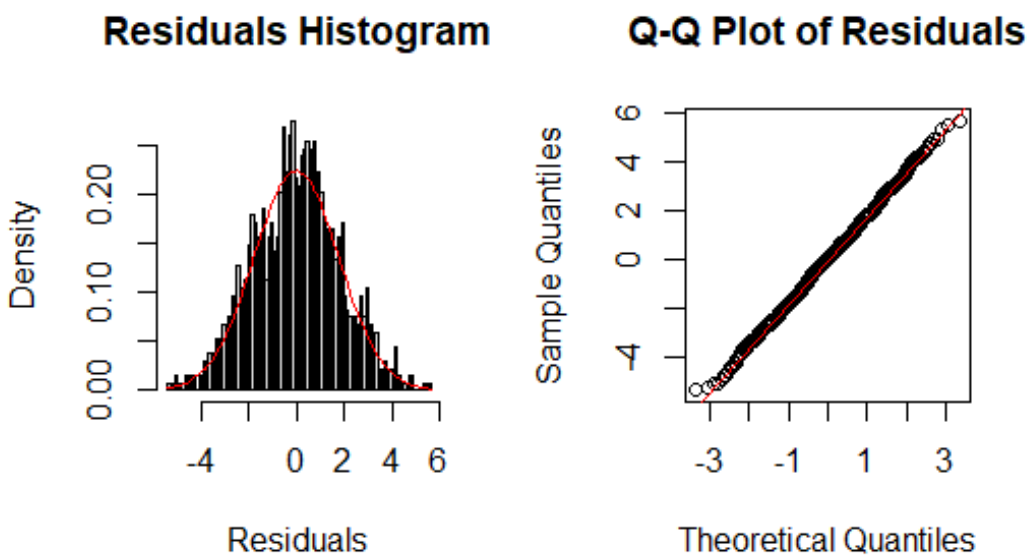


Figure 69: Residuals Histogram and Q-Q Plot of Residuals

H_0: the residuals are normally distributed
H_a: the residuals are not normally distributed

We can see that the scatter points of residuals is quite close to qqline, and the residuals is quite normal distributed when observing the histogram. In addition, p-value = 0.6128 > any level of significance, so we doesn't have enough evidence to reject H0: $\mu_\epsilon = 0$, which states that residuals of the model are normally distributed.

III.5.3. Studentized Breusch-Pagan test for heteroscedasticity

```
        studentized Breusch-Pagan test

data:  model_cox
BP = 20.489, df = 7, p-value = 0.004605
```

Figure 70: Studentized Breusch-Pagan test

H_0: The residuals have constant variance.
H_a: The residuals do not have constant variance.

Although p-value has increased siginificantly, but p-value $= 0.004605 < \alpha = 0.05$, we reject the null hypothesis, which suggests that the residuals do not have constant variance.

III.6. Model Interpretation

```
Call:
lm(formula = ((((Life.Ladder)^best_lambda) - 1)/best_lambda) ~
    Log.GDP.per.capita + Social.support + Freedom.to.make.life.choices +
        Generosity + Perceptions.of.corruption + Positive.affect +
        `RegionLatin America and Caribbean`, data = data_train)

Residuals:
    Min      1Q  Median      3Q     Max
-5.3389 -1.2739  0.0052  1.1280  5.6704

Coefficients:
                                    Estimate Std. Error t value Pr(>|t|)
(Intercept)                        -15.66470    0.67964 -23.048  < 2e-16 ***
Log.GDP.per.capita                   2.08278    0.06374  32.675  < 2e-16 ***
Social.support                       7.14593    0.64350  11.105  < 2e-16 ***
Freedom.to.make.life.choices         2.91299    0.49501   5.885 5.04e-09 ***
Generosity                           3.23274    0.37373   8.650  < 2e-16 ***
Perceptions.of.corruption           -4.67932    0.33311 -14.047  < 2e-16 ***
Positive.affect                      4.56912    0.70781   6.455 1.51e-10 ***
`RegionLatin America and Caribbean`  1.63043    0.15673  10.403  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.782 on 1331 degrees of freedom
Multiple R-squared:  0.8362,    Adjusted R-squared:  0.8354
F-statistic: 970.9 on 7 and 1331 DF,  p-value: < 2.2e-16
```

Figure 71: Best model

III.6.1. Quantile

- 25% of residuals are less than -1.2739.
- 50% of residuals are above 0.0052, 50% of residuals are below 0.0052.
- 75% of residuals are less than 1.128.

III.6.2. Coefficent of predictors

We have $y = \frac{(Life.Ladder)^{\lambda_{best}} - 1}{\lambda_{best}}$
Regression model:

$$\begin{aligned} y = \beta_0 &+ \beta_1 \cdot \text{Log.GDP.per.capita} + \beta_2 \cdot \text{Social.support} \\ &+ \beta_3 \cdot \text{Freedom.to.make.life.choices} + \beta_4 \cdot \text{Generosity} \\ &+ \beta_5 \cdot \text{Perceptions.of.corruption} + \beta_6 \cdot \text{Positive.affect} \\ &+ \beta_7 \cdot \text{Region Latin America and Caribbean} + \epsilon \end{aligned}$$

- $\hat{\beta}_0$: y = -15.663 when the remaining predictors are zero.
- $\hat{\beta}_1$: For each unit increase in log gdp per capita, on average, the expected value of y increases by 2.08, holding other predictors constant.
- $\hat{\beta}_2$: For each unit increase in the national average of binary response of Social Support, on average the expected value of y increases by 7.14, holding other predictors constant.
- $\hat{\beta}_3$: For each unit increase in the national average of binary response of Freedom to make life choices, on average, the expected value of y increases by 2.91, holding other predictors constant.
- $\hat{\beta}_4$: For each unit increase in the national average of binary response of Generosity, on average, the expected value of y increases by 3.23, holding other predictors constant.
- $\hat{\beta}_5$: For each unit increase in the national average of binary response of Perceptions of corruption, on average, the expected value of y decreases by 4.68, holding other predictors constant.
- $\hat{\beta}_6$: For each unit increase in the national average of binary response of Positive affect, on average, the expected value of y increases by 4.57, holding other predictors constant.
- $\hat{\beta}_6$: if the survey is in the region Latin America and Carribean, on average, the expected value of y increases 1.63, holding other predictors constant.

Overall, we can see that the model's coefficients are reasonable, positive predictors (such as social support, freedom to make life choices,...) have positive coefficients, and vice versa.

III.6.3. Multiple R-squared and Adjusted R-squared

Multiple R-squared: 0.8362 interprets that 83.62% of the variance in the response variable $\frac{(Life.Ladder)^{\lambda_{best}}-1}{\lambda_{best}}$ can be explained by the predictor variables in the model. The adjusted R-squared is 0.8354, which is slightly lower than the multiple R-squared. This indicates that the model is not overfitting the data.

III.6.4. Residual standard error

Residual standard error: rse = 1.782 indicates that the model's predictions are, on average, approximately 1.782 units away from the actual values of $\frac{(Life.Ladder)^{\lambda_{best}}-1}{\lambda_{best}}$. Some points are further from the line than this rse, other points are closer to the line than this rse. We can see that the model is not perfect, but it is reasonable enough.

III.7. Prediction

We use cross validation k-folds to make the prediction. The code is shown in the Code Snippet 25 .This is the first part of the prediction

	Actual <dbl>	Predicted <dbl>
2	4.402	3.688798
3	4.758	3.861982
4	3.832	3.689400
6	3.572	3.503286
11	2.694	2.515046
12	2.375	2.038411
24	4.995	5.104714
26	5.464	5.233613
30	5.249	5.126895
35	3.937	4.655325

Figure 72: Actual and prediction value of life ladder

III.8. Evaluation

The Code Snippet 25 also calculates the rmse and R-squared of the model.

We have:

- RMSE: 0.442
- R-squared: 0.852

The RMSE value is 0.442, which is quite low compared to the range of the target variable Life.Ladder (0-10). This means that the model is able to predict the target variable with a high degree of accuracy. The R-squared value (0.852) indicates that the model is able to explain 85.2% of the variance in the target variable. This is a good result, as it shows that the model is able to capture a large proportion of the variation in the target variable.

III.9. Conclusion

We built the best model in our ability, which is the model regressing on at least 4 variables and have acceptable results. However, there are still some limitations in our model. The first limitation is the data. The data we used is from the World Happiness Report, which is a survey-based report. The data can be subjective and may be inaccurate in some factors. The second limitation is the features. These features may still be not enough to predict the truth value of the happiness of a country. The third limitation is the model. In the future, we are expecting to improve the quality of analysis by using more features and trying different models. We can also use other data sources to improve the accuracy of our model. Overall, we believe that our model can be used to predict the happiness of a country with relatively high accuracy, but there is still room for improvement.

IV. Activity 2: Suicide

After analyzing about the positive side of the world: Happiness, we also want to provide an insight about one of the most serious problems, suicide. In this part, we will study factors that influence the suicide rate worldwide. By examining trends, patterns, and correlations within the dataset, we aim to build the model to predict the number of suicides with given factors. Through this process, we hope to contribute to a better understanding of suicide, which can be beneficial to build the strategies for the prevention.

IV.1. Dataset description

Link to the dataset: Suicide Rates Overview 1985 to 2016
The Suicide Rates Overview dataset is sourced from Kaggle platform, a popular online community for data scientists and machine learning practitioners. The dataset pulled from four other datasets (please see the link) linked by time and place, and was built to find signals correlated to increased suicide rates among different cohorts globally, across the socio-economic spectrum.

Variable	Type	Description
country	Multi-valued discrete	The name of the country where the survey was conducted. About 100 countries participate in the survey.
year	Multi-valued discrete	The year of the survey, ranging from 1987 to 2016.
sex	Categorical	The gender of the individuals (male or female).
age	Multi-value discrete	The age group of the individuals surveyed, categorized into ranges (15-24 years, 35-54 years, 25-34 years, 55-74 years).
suicides_no	Multi-value discrete	The number of suicides.
population	Multi-value discrete	The number of population.
suicides/ 100k pop	Continuous	The suicide rate per 100,000 population.
country-year	Multi-value discrete	A combined identifier for the country and year (e.g., "Albania1987").
gdp_for_ year ($)	Continuous	The GDP for the country in specific year, in US dollars.
gdp_per_ capita ($)	Continuous	The GDP per capita for the country in specific year, in US dollars.

Variable	Type	Description
generation	Categorical	The generation category based on the age group (Generation X, Silent, Boomers, Millenials).

The data has at least 1 qualitative (discrete) variable and at least 3 quantitative (continuous) variables as requirement.

IV.2. Import and preprocess dataset

IV.2.1. Import dataset

```
'data.frame':   27820 obs. of  12 variables:
 $ country           : chr  "Albania" "Albania" "Albania" "Albania" ...
 $ year              : int  1987 1987 1987 1987 1987 1987 1987 1987 1987 1987
...
 $ sex               : chr  "male" "male" "female" "male" ...
 $ age               : chr  "15-24 years" "35-54 years" "15-24 years" "75+
years" ...
 $ suicides_no       : int  21 16 14 1 9 1 6 4 1 0 ...
 $ population        : int  312900 308000 289700 21800 274300 35600 278800
257200 137500 311000 ...
 $ suicides.100k.pop : num  6.71 5.19 4.83 4.59 3.28 2.81 2.15 1.56 0.73 0 ...
 $ country.year      : chr  "Albania1987" "Albania1987" "Albania1987"
"Albania1987" ...
 $ HDI.for.year      : num  NA NA NA NA NA NA NA NA NA NA ...
 $ gdp_for_year....  : chr  "2,156,624,900" "2,156,624,900" "2,156,624,900"
"2,156,624,900" ...
 $ gdp_per_capita....: int  796 796 796 796 796 796 796 796 796 796 ...
 $ generation        : chr  "Generation X" "Silent" "Generation X" "G.I.
Generation" ...
[1] 27820    12
```

Figure 73: Structure of suicide dataset.

Figure 73 states that there are total 27820 observations of 9 variables

IV.2.2. Rename columns

To improve readability and ease of use, certain columns were renamed. Specifically: The 'gdp_for_year....' column was renamed to 'gdp_for_year'. The 'gdp_per_capita....' column was renamed to 'gdp_per_capita'.

IV.2.3. Process missing values

The Code Snippet 28 is used for this task.

```
       country              year               sex               age
             0                 0                 0                 0
   suicides_no        population suicides.100k.pop      country.year
             0                 0                 0                 0
  HDI.for.year        generation      gdp_for_year    gdp_per_capita
         19456                 0                 0                 0
```

Figure 74: Missing values.

The missing values of HDI.for.year account for about 70% of the dataset. Therefore, we decide to remove this variable. Moreover, country.year is combination string of country and year, which is redundant and suicides.100k.pop is suicides / (population / 100k), which includes our dependent variable suicides_no, so we also decide to remove those variables at first.

IV.2.4. Process unnecessary columns

Following the decision on removal of unnecessary variables (HDI.for.year, country.year, and suicides.100k.pop), the dataset was updated to create a cleaner version (data_clean). This step demonstrated in Code Snippet 29.

IV.2.5. Convert data types

The 'gdp_for_year' variable originally contained commas, which were removed using the gsub() function. The column was then converted from a character type to a numeric type using the as.numeric() function. This conversion in Code Snippet 30 is essential for performing any numerical operations on GDP data, such as statistical analysis or modeling.

IV.2.6. Process duplicate rows

The output of Code Snippet 31 shows no appearance of duplicate rows.

IV.2.7. Normalize Variables

Based on the Descriptive Statistics section, we normalize the 'suicides_no', 'gdp_per_capita', 'population', 'gdp_per_capita' in Code Snippet 32.

IV.2.8. Process important variables

In the first part of Code Snippet 33, we see that the 'model_year' in this dataset ranges from 1985 to 2016. To make the data easier to work with, we convert these years into a new range from 1 to 32.

The 'country' column, which contained categorical data, was converted into multiple binary (dummy) variables representing different regions. Each region corresponds to a group of countries, and a value of 1 indicates that a country belongs to that region, while 0 indicates it does not. The regions created are: 'regionEurope', 'regionAsia', 'regionAfrica', 'regionNorthAmerica', 'regionSouthAmerica', 'regionOceania'.

The third part of Code Snippet 7 is about creating dummy variables for 'sex', which is 'sexMale' (1: Male, 0: Female).

The forth part of Code Snippet 7 is about creating dummy variables for 'generation', which are 'geneX', 'geneMillenials', 'geneBoomers', 'geneSilent'

The fifth part of Code Snippet 7 is about creating dummy variables for 'age'. This age variables are seperated in 4 age group which are '15-24 years', '25-34 years', '35-54 years', '55-74 years'

IV.2.9. Process outliers

We use Cook's Distance instead of traditional IQR to process outlier as there is too many zero values in 'suicide_no' so we only concentrate on processing influential data. However, the threshold (4/length(cooksD)) is not processing enough of these special outliers. We decided to use (3/length(cooksD)), which does make the model better in terms of residuals normality and heteroscedasticity. The Code Snippet 34 stores this processing step. After applying this technique, the observations decline from 27820 to 25446 (lose 9.5% $<$ 10%), which is appropriate proportion when cutting down dataset.

IV.3. Descriptive Statistics

IV.3.1. Visualization and process data

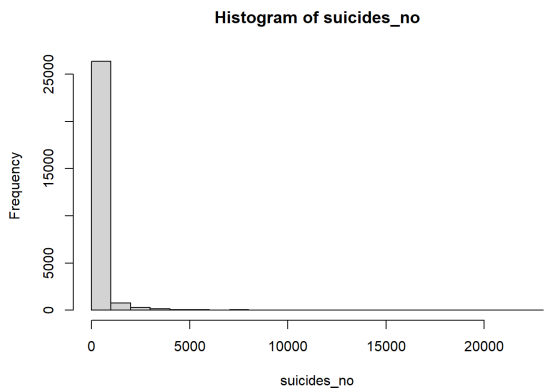


Figure 75: Histogram of suicide numbers.

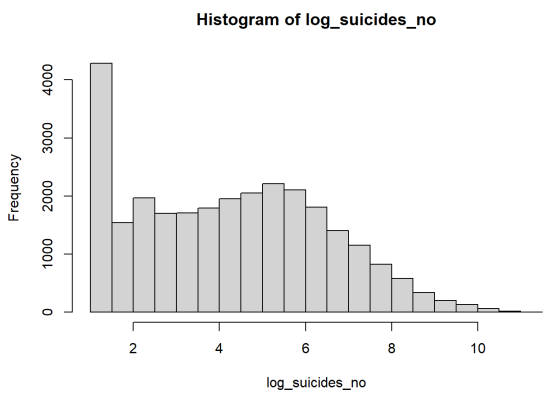


Figure 76: Histogram of log-transformed of suicide numbers.

The response variable we're focusing on is suicide_no. This variable has many zero values, making its distribution heavily right-skewed. To address this, we applied a logarithmic transformation. We added 1 inside the logarithm to avoid issues with taking the log of zero and added another 1 outside to prepare for a Box-Cox transformation later on (since Box-Cox cannot handle zero or negative values). Our new response variable is now defined as $\ln(suicide\_no + 1) + 1$.

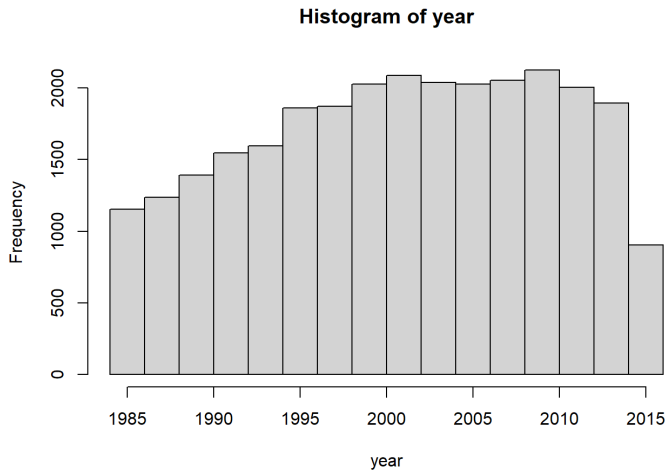


Figure 77: Histogram frequencies of 'year' having suicide records.

The most frequent years in the dataset are between 2000 and 2010, indicating that these years have the highest number of records related to suicides. The histogram shows a unimodal distribution, with a peak around the early 2000s, considering the relatively symmetric distribution around these years.

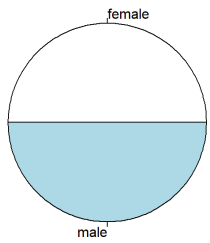


Figure 78: Distribution of suicide records by sex (Pie Chart).

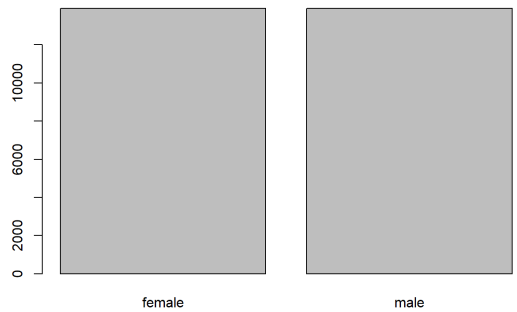


Figure 79: Distribution of suicide records by sex (Bar Plot).

The suicide records distributed evenly among male and female.

We noticed that it's necessary to normalize the predictors because the model without transformed predictors didn't pass the tests for normal residuals and constant variance in the Hypothesis. Testing section.

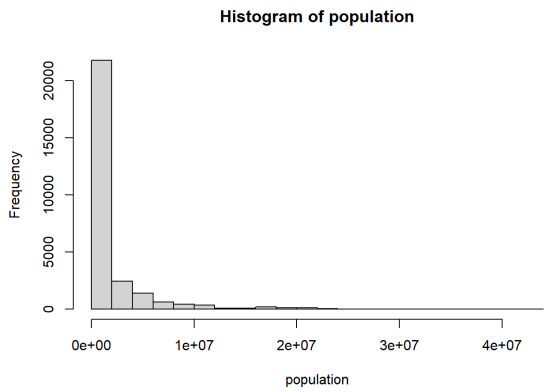


Figure 80: Histogram of the 'population'.

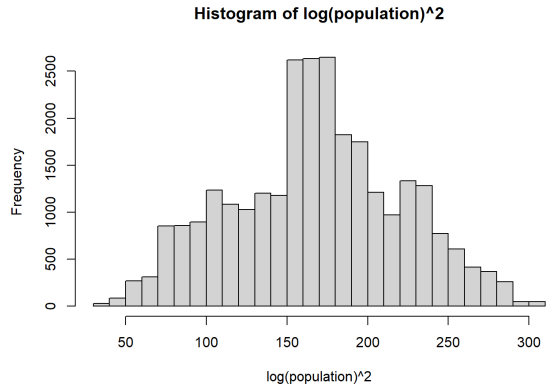


Figure 81: Histogram of the square of the log-transformed 'population'.

As shown in Figure 80, the population variable is highly right-skewed. To fix this, we used a logarithmic transformation, but the result shows that this made the data slightly left-skewed. To make it more symmetric and bell-shaped, we then squared the log-transformed values in Figure 81.

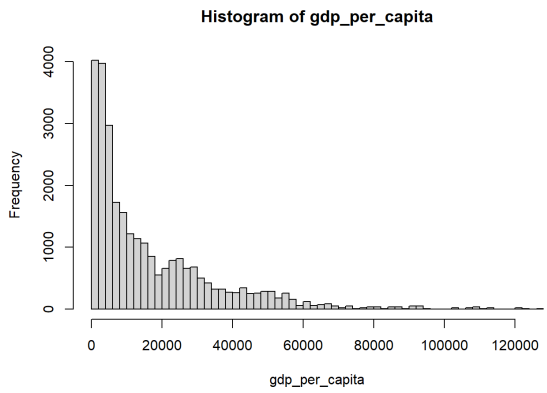


Figure 82: Histogram of 'gdp_per_capita'.

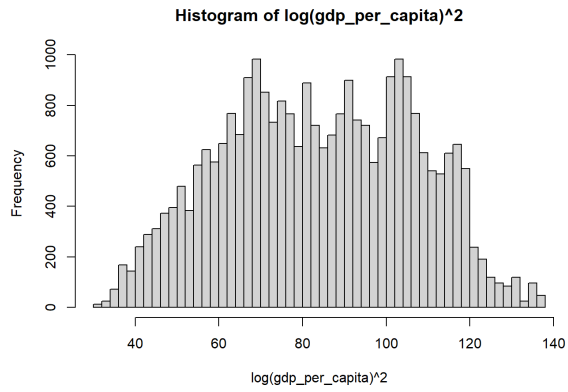


Figure 83: Histogram of the square of the log-transformed 'gdp_per_capita'.

The same goes for 'gdp_per_capita' variable so we apply the same processing steps. The results are shown in Figure 82, and 83.

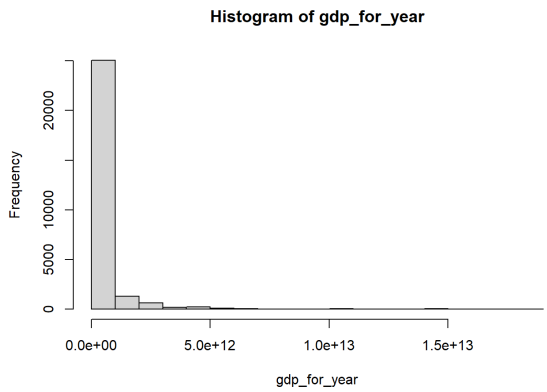


Figure 84: Histogram of 'gdp_for_year'.

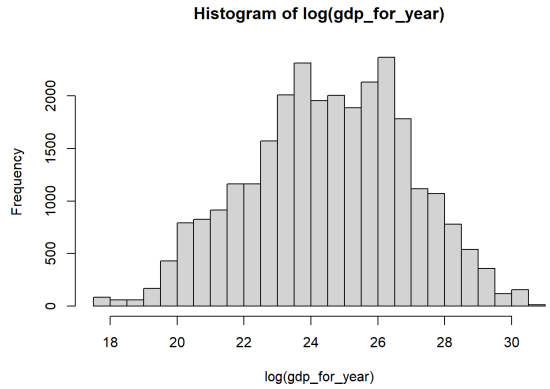


Figure 85: Histogram of the log-transformed 'gdp_for_year'.

In normalizing 'gdp_for_year', we observed the summary dataset table in Figure 88 that the 'transformed_gdp_for_year', which is $ln(gdp\_for\_year)$, has the mean and median very close to each other so we stop apply square to the formula.

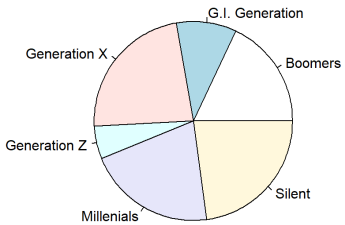


Figure 86: Distribution of suicide records by generation (pie chart).

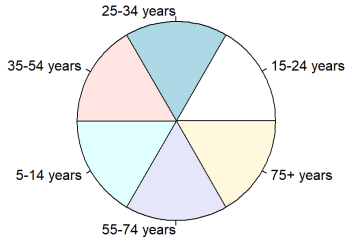


Figure 87: Distribution of suicide records by age group (pie chart).

In Figure 86, Generation Z has the lowest number of suicide records, which contrasts with older generations like the Boomers and the Silent Generation that show a higher proportion. This difference could be indicative of varying socio-economic pressures, mental health awareness, generational differences in coping mechanisms, or even differences in data availability across these generations (Generation Z has either lower suicide rates or that there is under-reporting or incomplete data collection for this group).

Based on the observation over Figure 87, the suicides cases distributed fairly evenly among different age group. The "55-74 years" age group spans three generations: Generation X, Boomers, and the Silent Generation. This is the age group with the most generational overlap.

```
##       year      log_suicides_no  transform_gdp_per_capita transform_population
##  Min.   : 1.0   Min.   : 1.000   Min.   : 30.53           Min.   : 31.67
##  1st Qu.:11.0   1st Qu.: 2.386   1st Qu.: 66.53           1st Qu.:132.28
##  Median :18.0   Median : 4.401   Median : 83.89           Median :168.83
##  Mean   :17.2   Mean   : 4.314   Mean   : 83.97           Mean   :168.29
##  3rd Qu.:24.0   3rd Qu.: 5.934   3rd Qu.:102.60           3rd Qu.:201.41
##  Max.   :32.0   Max.   :10.671   Max.   :137.99           Max.   :309.59
##  transform_gdp_for_year  regionEurope      regionAsia      regionAfrica
##  Min.   :17.66          Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:22.95          1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000
##  Median :24.59          Median :0.0000   Median :0.0000   Median :0.00000
##  Mean   :24.54          Mean   :0.4073   Mean   :0.1877   Mean   :0.02165
##  3rd Qu.:26.27          3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.00000
##  Max.   :30.53          Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
##  regionNorthAmerica regionSouthAmerica regionOceania        sexMale
##  Min.   :0.0000     Min.   :0.0000     Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.0000     1st Qu.:0.0000     1st Qu.:0.00000   1st Qu.:0.0000
##  Median :0.0000     Median :0.0000     Median :0.00000   Median :0.0000
##  Mean   :0.1913     Mean   :0.1185     Mean   :0.03741   Mean   :0.4971
##  3rd Qu.:0.0000     3rd Qu.:0.0000     3rd Qu.:0.00000   3rd Qu.:1.0000
##  Max.   :1.0000     Max.   :1.0000     Max.   :1.00000   Max.   :1.0000
##      geneX        geneMillenials    geneBoomers       geneSilent
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.000
##  Median :0.0000   Median :0.0000   Median :0.0000   Median :0.000
##  Mean   :0.2369   Mean   :0.2139   Mean   :0.1856   Mean   :0.231
##  3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.000
##    Age15to24        Age25to34        Age35to54        Age55to74
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.1756   Mean   :0.1752   Mean   :0.1694   Mean   :0.1693
##  3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
```

Figure 88: Summary of suicide dataset after preprocessing steps.

Comments on summary of preprocessed datasets in Figure 88:

- The variable 'transform_gdp_per_capita' has a range from 30.53 to 137.99, with a mean of 83.97 and a median of 83.89. The small difference between the mean and median suggests that the transformation has helped stabilize the distribution.
- 'log_suicides_no': The mean is 4.314 and the median is 4.401, indicating a slightly left-skewed distribution.
- The binary variables for regions (e.g., 'regionOceania', 'regionAfrica', etc.) show that the mean values are close to zero, indicating that these regions are less represented in the dataset.

IV.3.2. Relationship between response variable and predictors

The code is used in the Code Snippet 36

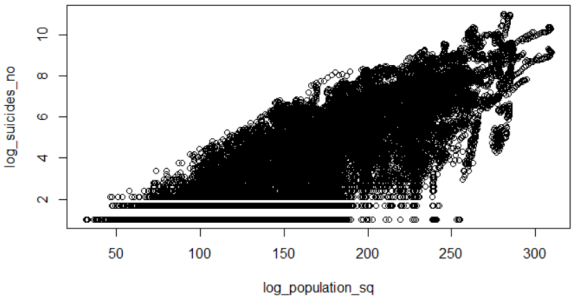


Figure 89: Suicide rate by population

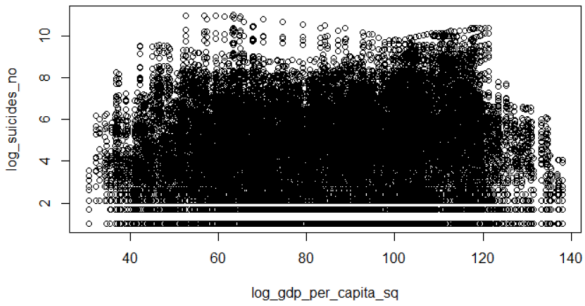


Figure 90: Suicide rate by gdp per capita

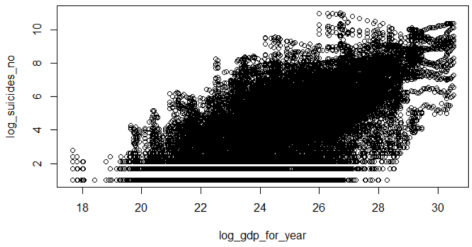


Figure 91: Suicide rate by gdp for year

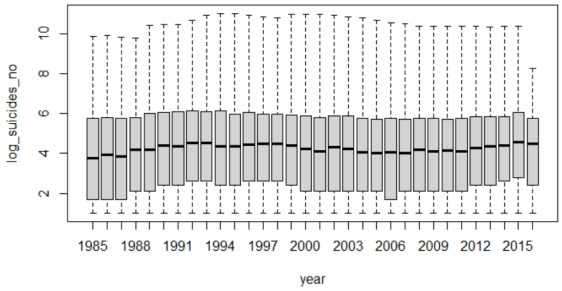


Figure 92: Suicide rate in the period from 1985 to 2016

Overall, through Figure 89, 90, 91 and 92, there is a positive linear relation between 'log_suicides_no' and 'log_population_square' as well as 'gdp_for_year' while there is no linear relation between 'log_suicides_no' and 'gdp_per_capita' as well as between 'log_suicides_no' and year.

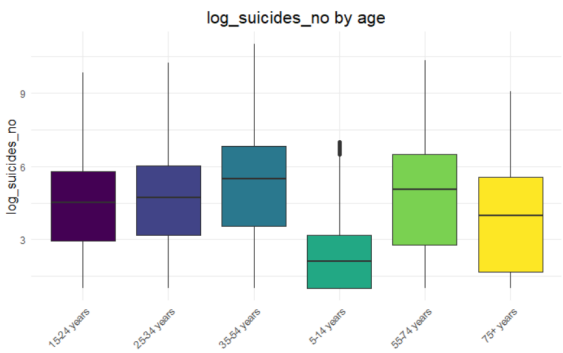


Figure 93: Suicide rate in different age

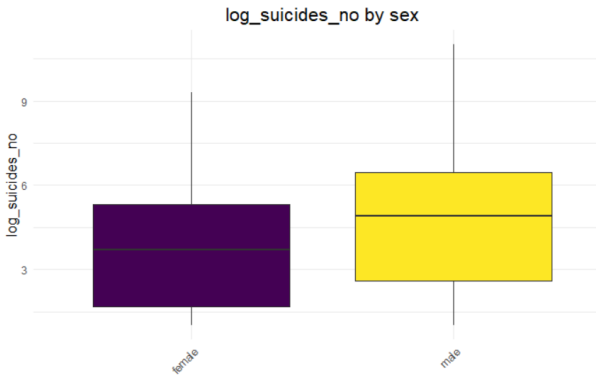


Figure 94: Suicide rate in male in female

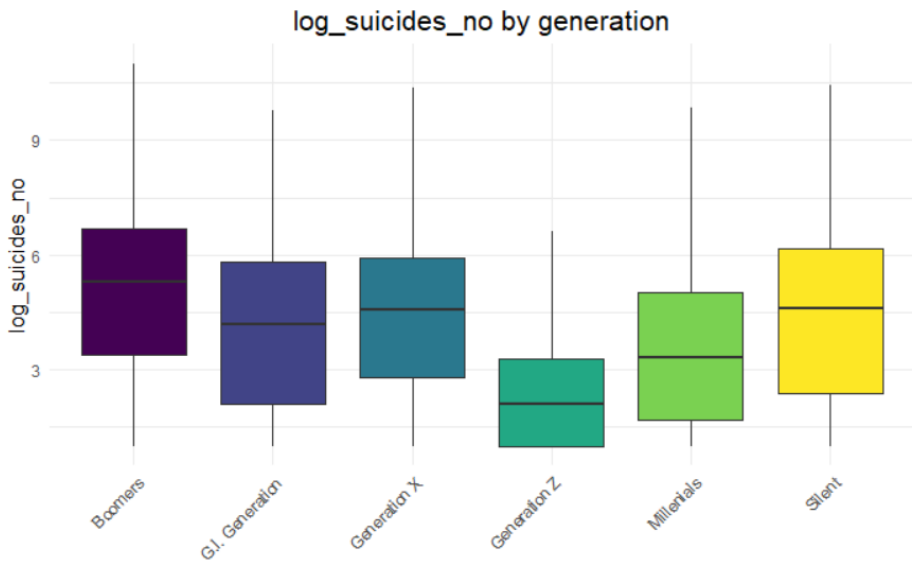


Figure 95: Suicide rate in different generation

Based on Figure 93, the age group from 5-14 year-old has the lowest suicide number while the other age group shows that the average log of suicide number lies about from 3 to 6. From the figure 94, male has more number of suicide than female. In addition, with respect to generation in Figure 95, generation Z has shown the lowest suicide number.

IV.4. Model Building

Before stepping into building the model, we will split the data into training and testing sets. The training set is used to build the model and the testing set is used to evaluate the model. The code is shown in the Code Snippet 37

- Split Ratio: Split the dataset into 80% training and 20% testing sets.
- data_clean has 25446 observations
- data_train has 20356 observations and data_test has 5090 observations.

IV.4.1. Check multicolinearity

The code to do this is in the Code Snippet 38 We fit the model with the train data.

```
Call:
lm(formula = log_suicides_no ~ ., data = data_train)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.64250 -0.55326  0.03595  0.59620  2.21050 

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)    
(Intercept)              -7.0165081  0.1861314 -37.697  < 2e-16 ***
year                      0.0060499  0.0007992   7.570 3.88e-14 ***
transform_gdp_per_capita -0.0154307  0.0007505 -20.561  < 2e-16 ***
transform_population      0.0222043  0.0004848  45.802  < 2e-16 ***
transform_gdp_for_year    0.3028174  0.0126812  23.879  < 2e-16 ***
regionEurope              0.7731988  0.0341066  22.670  < 2e-16 ***
regionAsia                0.3415597  0.0359580   9.499  < 2e-16 ***
regionAfrica              0.1990702  0.0507310   3.924 8.74e-05 ***
regionNorthAmerica        0.2105835  0.0347610   6.058 1.40e-09 ***
regionSouthAmerica        0.3464502  0.0377073   9.188  < 2e-16 ***
regionOceania             0.6888137  0.0442557  15.564  < 2e-16 ***
sexMale                   1.0909626  0.0118003  92.452  < 2e-16 ***
geneX                    -1.1212766  0.0283997 -39.482  < 2e-16 ***
geneMillenials           -1.5562038  0.0271608 -57.296  < 2e-16 ***
geneBoomers              -0.5826200  0.0290184 -20.078  < 2e-16 ***
geneSilent                0.2050384  0.0220915   9.281  < 2e-16 ***
Age15to24                 1.7095972  0.0217454  78.619  < 2e-16 ***
Age25to34                 1.6204107  0.0230932  70.168  < 2e-16 ***
Age35to54                 1.5167949  0.0273872  55.383  < 2e-16 ***
Age55to74                 0.8103206  0.0229191  35.356  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8383 on 20336 degrees of freedom
Multiple R-squared:  0.8523,	Adjusted R-squared:  0.8522 
F-statistic:  6178 on 19 and 20336 DF,  p-value: < 2.2e-16
```

Figure 96: First linear regression model

We will check for the presence of multicollinearity among the predictor variables using Variance Inflation Factor (VIF). If VIF > 10, it indicates high multicollinearity, and corrective measures should be taken.

```
                  year transform_gdp_per_capita     transform_population 
              1.298947                 8.329852                18.450468 
transform_gdp_for_year             regionEurope               regionAsia 
             25.791534                 8.139897                 5.688080 
          regionAfrica       regionNorthAmerica       regionSouthAmerica 
              1.601222                 5.423623                 4.317677 
         regionOceania                  sexMale                     geneX 
              2.031497                 1.008424                 4.233884 
        geneMillenials              geneBoomers               geneSilent 
              3.607707                 3.687870                 2.504603 
             Age15to24                Age25to34                Age35to54 
              2.022694                 2.213491                 3.064101 
             Age55to74 
              2.131516 
```

Figure 97: First VIF

After calculating VIF values for each predictor (Figure 97), we observe that the 'transform_gdp_for_year' variable has the highest VIF value (25.791534). To address multicollinearity and improve the model's stability, we will remove the 'displacement' variable from the model.

```
                year transform_gdp_per_capita     transform_population
            1.286351                 1.415712                 1.256022
      regionEurope                regionAsia              regionAfrica
            8.051713                 5.629264                 1.600051
regionNorthAmerica       regionSouthAmerica            regionOceania
            5.405438                 4.269006                 2.022247
             sexMale                    geneX           geneMillenials
            1.001929                 3.655131                 2.977923
        geneBoomers                geneSilent                Age15to24
            3.462813                 2.504552                 2.006674
           Age25to34                Age35to54                Age55to74
            2.181143                 2.500763                 1.677774
```

Figure 98: Second VIF

Then we consider the vif again in Figure 98, we observe that 'regionEurope' variable has the highest VIF value this time (8.051713). We try removing this variable. By making regression between the 'regionEurope' variable and the other predictors, the R-squared = 0.8758. Moreover, there are strong correlation among horsepower towards gdp_per_capita, population, regions, geneX, geneMillenials and age. Hence, we will remove the 'regionEurope' variable from the model.
Finally, we get the final VIF and final model after checking multicolinearity.

```
                year transform_gdp_per_capita     transform_population
            1.284615                 1.412827                 1.165669
          regionAsia              regionAfrica       regionNorthAmerica
            1.320168                 1.064831                 1.237978
regionSouthAmerica            regionOceania                   sexMale
            1.297614                 1.049945                 1.001905
               geneX           geneMillenials              geneBoomers
            3.654003                 2.977079                 3.462729
          geneSilent                Age15to24                Age25to34
            2.504092                 2.005668                 2.179530
           Age35to54                Age55to74
            2.495401                 1.675058
```

Figure 99: Final VIF

```
Call:
lm(formula = log_suicides_no ~ . - transform_gdp_for_year - regionEurope,
    data = data_train)

Residuals:
     Min       1Q   Median       3Q      Max
-2.77886 -0.55459  0.05403  0.60832  2.23842

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)              -2.0567292  0.0368312 -55.842  < 2e-16 ***
year                      0.0071900  0.0008180   8.790  < 2e-16 ***
transform_gdp_per_capita  0.0012501  0.0003181   3.930 8.54e-05 ***
transform_population      0.0342378  0.0001254 272.986  < 2e-16 ***
regionAsia               -0.3627247  0.0178302 -20.343  < 2e-16 ***
regionAfrica             -0.5100124  0.0425811 -11.977  < 2e-16 ***
regionNorthAmerica       -0.5120456  0.0170935 -29.955  < 2e-16 ***
regionSouthAmerica       -0.3491385  0.0212766 -16.410  < 2e-16 ***
regionOceania            -0.0142660  0.0327471  -0.436    0.663
sexMale                   1.1150233  0.0121064  92.102  < 2e-16 ***
geneX                    -1.3837385  0.0271555 -50.956  < 2e-16 ***
geneMillenials           -1.8376929  0.0253952 -72.364  < 2e-16 ***
geneBoomers              -0.7573088  0.0289416 -26.167  < 2e-16 ***
geneSilent                0.2102268  0.0227358   9.247  < 2e-16 ***
Age15to24                 1.6511175  0.0222874  74.083  < 2e-16 ***
Age25to34                 1.5379793  0.0235861  65.207  < 2e-16 ***
Age35to54                 1.2074024  0.0254387  47.463  < 2e-16 ***
Age55to74                 0.5371167  0.0209120  25.685  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8628 on 20338 degrees of freedom
Multiple R-squared:  0.8436,    Adjusted R-squared:  0.8434
F-statistic:  6451 on 17 and 20338 DF,  p-value: < 2.2e-16
```

Figure 100: Model after checking multicolinearity

IV.4.2. Variable selection

We decide to use AIC Stepwise Regression with backward stepwise selection with the full model as the model after checking multicolinearity because this model has 16 predictors which quite large and these predictors are considered statistically significant through the p_value in summary model. The code is in Code Snippet 39

```
Call:
lm(formula = log_suicides_no ~ year + transform_gdp_per_capita +
    transform_population + regionAsia + regionAfrica + regionNorthAmerica +
    regionSouthAmerica + sexMale + geneX + geneMillenials + geneBoomers +
    geneSilent + Age15to24 + Age25to34 + Age35to54 + Age55to74,
    data = data_train)

Residuals:
     Min       1Q   Median       3Q      Max
-2.77804 -0.55479  0.05352  0.60894  2.23968

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)              -2.0587721  0.0365307 -56.357  < 2e-16 ***
year                      0.0071976  0.0008178   8.801  < 2e-16 ***
transform_gdp_per_capita  0.0012538  0.0003180   3.943 8.09e-05 ***
transform_population      0.0342412  0.0001252 273.541  < 2e-16 ***
regionAsia               -0.3615952  0.0176403 -20.498  < 2e-16 ***
regionAfrica             -0.5086993  0.0424735 -11.977  < 2e-16 ***
regionNorthAmerica       -0.5108360  0.0168662 -30.288  < 2e-16 ***
regionSouthAmerica       -0.3479795  0.0211092 -16.485  < 2e-16 ***
sexMale                   1.1150594  0.0121059  92.109  < 2e-16 ***
geneX                    -1.3837826  0.0271548 -50.959  < 2e-16 ***
geneMillenials           -1.8377777  0.0253939 -72.371  < 2e-16 ***
geneBoomers              -0.7573298  0.0289410 -26.168  < 2e-16 ***
geneSilent                0.2101417  0.0227345   9.243  < 2e-16 ***
Age15to24                 1.6510224  0.0222859  74.084  < 2e-16 ***
Age25to34                 1.5379131  0.0235851  65.207  < 2e-16 ***
Age35to54                 1.2072322  0.0254352  47.463  < 2e-16 ***
Age55to74                 0.5370362  0.0209108  25.682  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8628 on 20339 degrees of freedom
Multiple R-squared:  0.8436,    Adjusted R-squared:  0.8434
F-statistic:  6854 on 16 and 20339 DF,  p-value: < 2.2e-16
```

Figure 101: AIC model

IV.4.3. Diagnostic

We test Independence, Homoscedasticity and Normality of the model. The code is shown in the Code Snippet 40

- Durbin-Watson test for autocorrelation

```
        Durbin-Watson test

data:  modelAIC
DW = 1.9835, p-value = 0.119
alternative hypothesis: true autocorrelation is greater than 0
```

Figure 102: Durbin-Watson test

H_0: There is no autocorrelation in the residuals
H_a: There is autocorrelation in the residuals
From the result (p_value = 0.119), there is no autocorrelation in the residuals.

- Shapiro-Wilk test for normality As the size of data_train is large, we take the maximum sample (5000) to use the Shapiro-Wilk test and get the result:

```
Shapiro-Wilk normality test

data:  subsample_residuals
W = 0.99452, p-value = 7.183e-13
```

Figure 103: Shapiro-Wilk Test

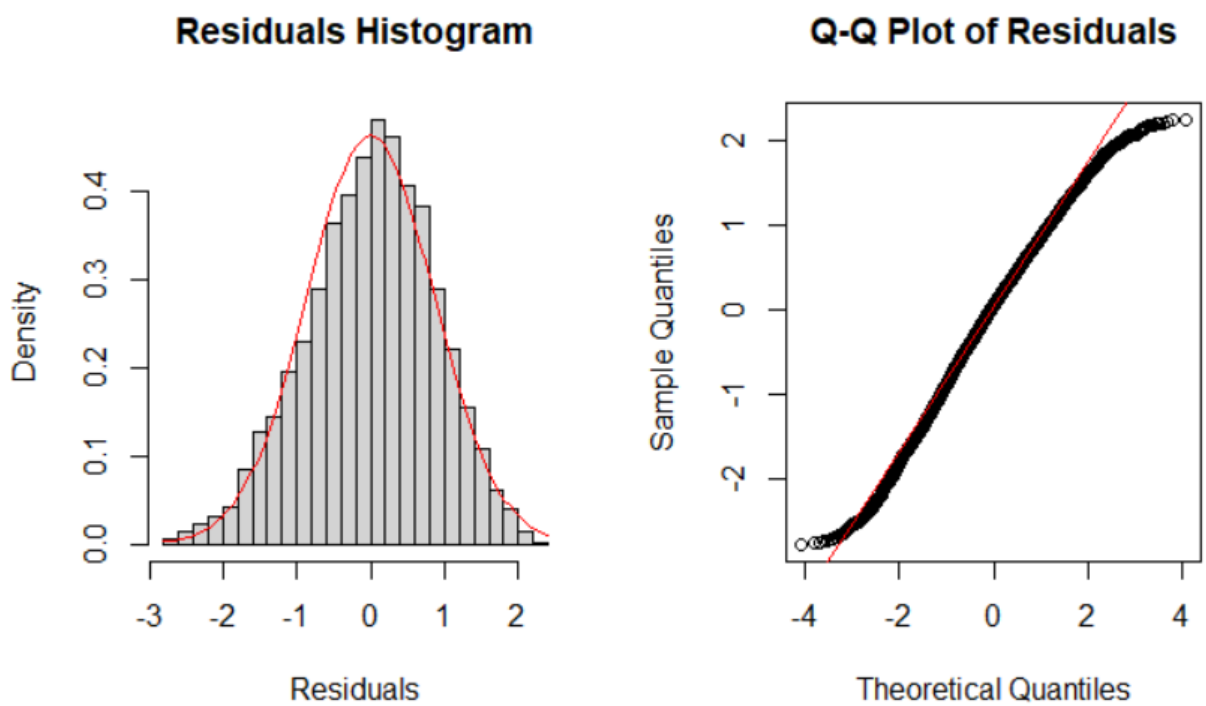


Figure 104: Residuals Histogram and Q-Q Plot of Residuals

H_0: the residuals are normally distributed
H_a: the residuals are not normally distributed

We observe that the residuals at both ends deviate from the qqline, indicating they might be not close to it. Additionally, p-value $= 1.922 \times 10^{-12} < \alpha = 0.05$, we have enough evidence to reject H0: $\mu_\epsilon = 0$, which states that residuals of the model are not normally distributed.

- Studentized Breusch-Pagan test for heteroscedasticity

```
        studentized Breusch-Pagan test

data:  modelAIC
BP = 3230.8, df = 16, p-value < 2.2e-16
```

Figure 105: Studentized Breusch-Pagan test

H_0: The residuals have constant variance.
H_a: The residuals do not have constant variance.
p-value $< 2.2 \times 10^{-16} < \alpha = 0.05$, we have enough evidence to reject the null hypothesis, which suggests that the residuals don't have constant variance.

IV.4.4. Box-Cox Transformation

As the model has failed the Shapiro-Wilk test for normality and the Studentized Breusch-Pagan test for heteroscedasticity so we decide to apply box-cox transformation.
The code is shown in the Code Snippet 42. Using the lambda = 1.5, we have the model.

```
Call:
lm(formula = (((data_train$log_suicides_no^best_lambda) - 1)/best_lambda) ~
    year + transform_gdp_per_capita + transform_population +
        regionAsia + regionAfrica + regionNorthAmerica + regionSouthAmerica +
        sexMale + geneX + geneMillenials + geneBoomers + geneSilent +
        Age15to24 + Age25to34 + Age35to54 + Age55to74, data = data_train)

Residuals:
    Min      1Q  Median      3Q     Max
-6.0861 -1.2347  0.0246  1.2400  6.7164

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)             -7.5002119  0.0796016 -94.222  < 2e-16 ***
year                     0.0094358  0.0017820   5.295  1.2e-07 ***
transform_gdp_per_capita 0.0060770  0.0006930   8.770  < 2e-16 ***
transform_population     0.0703295  0.0002728 257.838  < 2e-16 ***
regionAsia              -0.6156610  0.0384388 -16.017  < 2e-16 ***
regionAfrica            -0.8608630  0.0925510  -9.302  < 2e-16 ***
regionNorthAmerica      -0.7754588  0.0367520 -21.100  < 2e-16 ***
regionSouthAmerica      -0.7861718  0.0459976 -17.092  < 2e-16 ***
sexMale                  2.3842002  0.0263790  90.382  < 2e-16 ***
geneX                   -2.8162367  0.0591711 -47.595  < 2e-16 ***
geneMillenials          -3.7239452  0.0553341 -67.299  < 2e-16 ***
geneBoomers             -1.5224463  0.0630634 -24.142  < 2e-16 ***
geneSilent               0.4454278  0.0495391   8.991  < 2e-16 ***
Age15to24                3.2321189  0.0485617  66.557  < 2e-16 ***
Age25to34                3.0314198  0.0513927  58.985  < 2e-16 ***
Age35to54                2.5169117  0.0554241  45.412  < 2e-16 ***
Age55to74                1.1261165  0.0455653  24.714  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.88 on 20339 degrees of freedom
Multiple R-squared:  0.8272,    Adjusted R-squared:  0.8271
F-statistic:  6086 on 16 and 20339 DF,  p-value: < 2.2e-16
```

Figure 106: Box-cox transformation model

IV.5. Model Diagnostic

The code is shown in the Code Snippet 41.

IV.5.1. Durbin-Watson test for autocorrelation

```
        Durbin-Watson test

data:  model_cox
DW = 1.9942, p-value = 0.34
alternative hypothesis: true autocorrelation is greater than 0
```

Figure 107: Durbin-Watson test

H_0: There is no autocorrelation in the residuals
H_a: There is autocorrelation in the residuals
From the result (p_value = 0.34), there is no autocorrelation in the residuals.

IV.5.2. Shapiro-Wilk test for residual normality

```
        Shapiro-Wilk normality test

data:  subsample_residuals
W = 0.99943, p-value = 0.1286
```

Figure 108: Shapiro-Wilk test

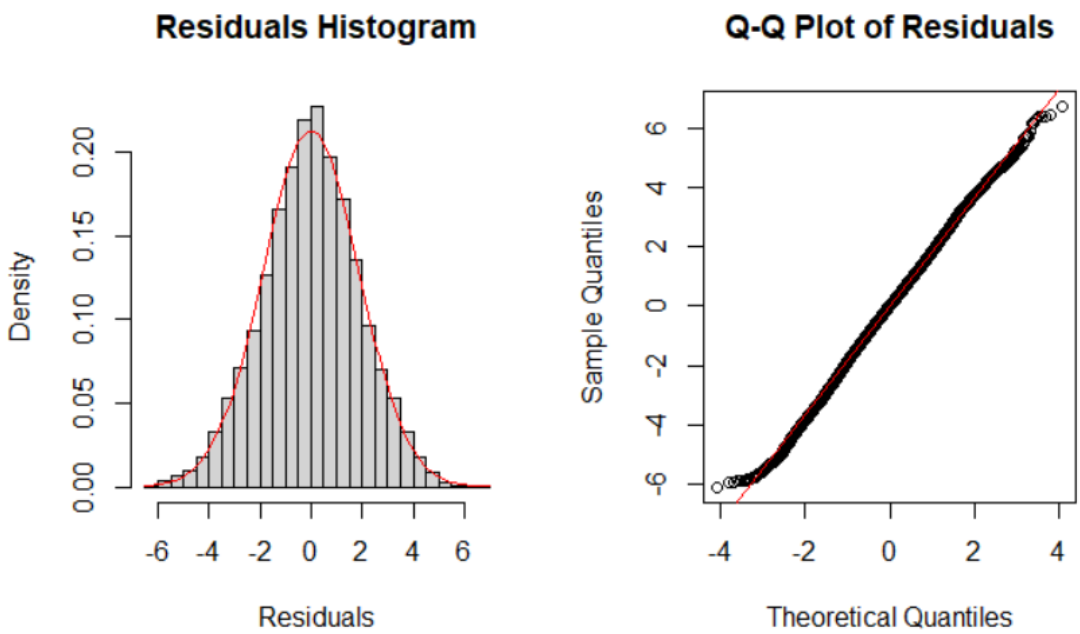


Figure 109: Residuals Histogram and Q-Q Plot of Residuals

H_0: the residuals are normally distributed
H_a: the residuals are not normally distributed

After transformation, we can see that the scatter points of residuals is more quite close to qqline and the residuals have normal distribution. In addition, p-value $= 0.1286 > \alpha = 0.05$ so we doesn't have enough evidence to reject H0: $\mu_\epsilon = 0$, which states that residuals of model are normally distributed.

IV.5.3. Studentized Breusch-Pagan test for heteroscedasticity

```
        studentized Breusch-Pagan test

data:  model_cox
BP = 2943.8, df = 16, p-value < 2.2e-16
```

Figure 110: Studentized Breusch-Pagan test

H_0: The residuals have constant variance.
H_a: The residuals do not have constant variance.

The model still fails the test as p-value remains extremely small, indicating that heteroscedasticity is still a significant issue. The transformation has not fully addressed the non-constant variance in the residuals. Hence, we decide to compare the BP statistic between 2 model. The BP statistic has decreased from 3230.76 to 2943.80 after the Box-Cox transformation. This suggests that the transformation has helped reduce heteroscedasticity, but it has not completely eliminated it.

IV.6. Model Interpretation

```
Call:
lm(formula = (((data_train$log_suicides_no^best_lambda) - 1)/best_lambda) ~
    year + transform_gdp_per_capita + transform_population +
        regionAsia + regionAfrica + regionNorthAmerica + regionSouthAmerica +
        sexMale + geneX + geneMillenials + geneBoomers + geneSilent +
        Age15to24 + Age25to34 + Age35to54 + Age55to74, data = data_train)

Residuals:
    Min      1Q  Median      3Q     Max
-6.0861 -1.2347  0.0246  1.2400  6.7164

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)             -7.5002119  0.0796016 -94.222  < 2e-16 ***
year                     0.0094358  0.0017820   5.295  1.2e-07 ***
transform_gdp_per_capita 0.0060770  0.0006930   8.770  < 2e-16 ***
transform_population     0.0703295  0.0002728 257.838  < 2e-16 ***
regionAsia              -0.6156610  0.0384388 -16.017  < 2e-16 ***
regionAfrica            -0.8608630  0.0925510  -9.302  < 2e-16 ***
regionNorthAmerica      -0.7754588  0.0367520 -21.100  < 2e-16 ***
regionSouthAmerica      -0.7861718  0.0459976 -17.092  < 2e-16 ***
sexMale                  2.3842002  0.0263790  90.382  < 2e-16 ***
geneX                   -2.8162367  0.0591711 -47.595  < 2e-16 ***
geneMillenials          -3.7239452  0.0553341 -67.299  < 2e-16 ***
geneBoomers             -1.5224463  0.0630634 -24.142  < 2e-16 ***
geneSilent               0.4454278  0.0495391   8.991  < 2e-16 ***
Age15to24                3.2321189  0.0485617  66.557  < 2e-16 ***
Age25to34                3.0314198  0.0513927  58.985  < 2e-16 ***
Age35to54                2.5169117  0.0554241  45.412  < 2e-16 ***
Age55to74                1.1261165  0.0455653  24.714  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.88 on 20339 degrees of freedom
Multiple R-squared:  0.8272,    Adjusted R-squared:  0.8271
F-statistic:  6086 on 16 and 20339 DF,  p-value: < 2.2e-16
```

Figure 111: Best model

IV.6.1. Quantile

- 25% of residuals are less than -1.2347.
- 50% of residuals are above 0.0246, 50% of residuals are below 0.0246.
- 75% of residuals are less than 1.24.

IV.6.2. Coefficent of predictors

We have $y = \frac{(log\_suicides\_no)^{\lambda_{best}} - 1}{\lambda_{best}}$
Regression model:

$$\begin{aligned} y = \beta_0 &+ \beta_1 \cdot \text{year} + \beta_2 \cdot \text{transform\_gdp\_per\_capita} + \beta_3 \cdot \text{transform\_population} \\ &+ \beta_4 \cdot \text{regionAsia} + \beta_5 \cdot \text{regionAfrica} + \beta_6 \cdot \text{regionNorthAmerica} + \beta_7 \cdot \text{regionSouthAmerica} \\ &+ \beta_8 \cdot \text{sexMale} + \beta_9 \cdot \text{geneX} + \beta_{10} \cdot \text{geneMillenials} + \beta_{11} \cdot \text{geneBoomers} + \beta_{12} \cdot \text{geneSilent} \\ &+ \beta_{13} \cdot \text{Age15to24} + \beta_{14} \cdot \text{Age25to34} + \beta_{15} \cdot \text{Age35to54} + \beta_{16} \cdot \text{Age55to74} + \epsilon \end{aligned}$$

- $\hat{\beta}_0$: y = -7.5 when the remaining predictors are zero.
- $\hat{\beta}_1$: For each unit increase in year, on average, the expected value of y increases by 0.01, holding other predictors constant.
- $\hat{\beta}_2$: For each unit increase in the gdp_per_capita, on average the expected value of y increases by 0.006, holding other predictors constant.
- $\hat{\beta}_3$: For each unit increase in the population, on average, the expected value of y increases by 0.07, holding other predictors constant.
- $\hat{\beta}_4$: if the region surveyed is Asia, the expected value of y decreases by 0.62, holding other predictors constant.
- $\hat{\beta}_5$: if the region surveyed is Africa, the expected value of y decreases by 0.86, holding other predictors constant.
- $\hat{\beta}_6$: if the region surveyed is NorthAmerica, the expected value of y decreases by 0.78, holding other predictors constant.
- $\hat{\beta}_7$: if the region surveyed is SouthAmerica, the expected value of y decreases by 0.79, holding other predictors constant.
- $\hat{\beta}_8$: if the gender surveyed is male, the expected value of y increases by 2.4, holding other predictors constant.
- $\hat{\beta}_9$: if the geneX is surveyed, the expected value of y decreases by 2.82, holding other predictors constant.
- $\hat{\beta}_{10}$: if the geneMillenials is surveyed, the expected value of y decreases by 3.72, holding other predictors constant.
- $\hat{\beta}_{11}$: if the geneBoomers is surveyed, the expected value of y decreases by 1.52, holding other predictors constant.
- $\hat{\beta}_{12}$: if the geneSilents is surveyed, the expected value of y increases by 0.45, holding other predictors constant.
- $\hat{\beta}_{13}$: if the age from 15 to 24 is surveyed, the expected value of y increases by 3.23, holding other predictors constant.
- $\hat{\beta}_{14}$: if the age from 25 to 34 is surveyed, the expected value of y increases by 3.03, holding other predictors constant.
- $\hat{\beta}_{15}$: if the age from 35 to 54 is surveyed, the expected value of y increases by 2.52, holding other predictors constant.
- $\hat{\beta}_{16}$: if the age from 55 to 74 is surveyed, the expected value of y increases by 1.13, holding other predictors constant.

IV.6.3. Multiple R-squared and Adjusted R-squared

Multiple R-squared: 0.8272 interprets that 82.72% of the variance in the response variable $\frac{(log\_suicides\_no)^{\lambda_{best}}-1}{\lambda_{best}}$ can be explained by the predictor variables in the model. The adjusted R-squared is 0.8271, which is slightly lower than the multiple R-squared. This indicates that the model is not overfitting the data.

IV.6.4. Residual standard error

Residual standard error: rse = 1.88 indicates that the model's predictions are, on average, approximately 1.88 units away from the actual values of $\frac{(log\_suicides\_no)^{\lambda_{best}}-1}{\lambda_{best}}$. Some points are further from the line than this rse, other points are closer to the line than this rse. We can see that the model is not perfect, but it is reasonable enough.

IV.7. Prediction

We use cross validation k-folds to do the task. The code is shown in the Code Snippet 43 This is the first part of the prediction

	Actual <dbl>	Predicted <dbl>
15	1	4.31
18	8	16.39
19	3	12.51
20	5	25.94
46	0	0.72
55	7	25.93
56	7	24.69
57	2	18.20
59	1	7.37
75	2	1.01

1-10 of 5,090 rows Previous 1

Figure 112: Actual and prediction value of suicides number

IV.8. Evaluation

The Code Snippet 43 also calculates the rmse and R-squared of the model.
We have:

- RMSE: 578.2537
- R-squared: 0.6606415

The RMSE value is 578.25, which is relatively small compared to the range of the target variable suicides_no (from 0 to roughly 22000). This means that the model is able to predict the number of suicides with a fair degree of accuracy. The R-squared value (0.66) reports that the model is able to explain around 66% of the variance in the target variable. This is an acceptable result, which shows that the model is able to capture a good proportion of the variation in the target variable.

IV.9. Conclusion

We have built the best model in our ability, which is the regression model on at least 4 variables and has acceptable results. However, there are still some limitations in our model. The first limitation is due to the data. The data may be subjective and lack attributes to ensure accuracy since this is just a survey. The second limitation comes from the target variable we want to predict. The number of suicides is a kind of sensitive information and may be underreported or inconsistently reported across different factors such as region or country, which can affect the accuracy of our predictions. In the future, we expect to improve the quality of analysis by using more features and trying different models. We can also use other data sources to produce better predictions. Overall, we believe that our model can be used to predict the number of suicides in a country with an acceptable accuracy, but there is still room for improvement.

A. Appendix: Code Listings

A.1. Activity 1

A.1.1. Import dataset

```
data <- read.csv("auto_mpg.csv", header = TRUE, sep = ";")
attach(data)
head(data)
dim(data)
str(data)
```

Listing 1: Import dataset

A.1.2. Process missing value

```
# Replace "?" with NA in the horsepower column
data$horsepower[data$horsepower == "?"] <- NA

# Convert horsepower data type to numeric
data$horsepower <- as.numeric(as.character(data$horsepower))

# Count the number of missing values
missing_values <- sapply(data, function(x) sum(is.na(x)))
print(missing_values)
str(data)
```

Listing 2: Process missing value

A.1.3. Process duplicate rows

```
# Identify duplicate rows based on all columns
duplicates <- duplicated(data_clean)
print(data_clean[duplicates, ])
```

Listing 3: Process duplicate rows

A.1.4. Process unnecessary variables

```
# Remove car_name column in data_clean
data_clean <- dplyr::select(data_clean, -car_name)
str(data_clean)
detach(data)
attach(data_clean)
```

Listing 4: Process unnecessary variables

A.1.5. Descriptive statistics.

```
# Visualize the dataset
data_clean %>%
  gather(key = "variable", value = "value") %>%
  ggplot(aes(x = value)) +
  facet_wrap(~ variable, scales = "free") +
  geom_histogram(bins = 30) +
```

```
  theme_minimal()
ggplot(data_clean, aes(y = acceleration)) +
  geom_boxplot(fill = "lightblue", color = "black") +
  labs(title = "Acceleration", y = " ") +
  theme(
    plot.title = element_text(size = 20, hjust = 0.5),  # Center the title horizontally
    axis.title.y = element_text(size = 16),
    axis.text.y = element_text(size = 14),
    axis.text.x = element_text(size = 14)
  )

ggplot(data_clean, aes(y = displacement)) +
  geom_boxplot(fill = "lightblue", color = "black") +
  labs(title = "Displacement", y = "") +
  theme(
    plot.title = element_text(size = 20, hjust = 0.5),  # Center the title horizontally
    axis.title.y = element_text(size = 16),
    axis.text.y = element_text(size = 14),
    axis.text.x = element_text(size = 14)
  )

ggplot(data_clean, aes(y = horsepower)) +
  geom_boxplot(fill = "lightblue", color = "black") +
  labs(title = "Horsepower", y = "") +
  theme(
    plot.title = element_text(size = 20, hjust = 0.5),  # Center the title horizontally
    axis.title.y = element_text(size = 16),
    axis.text.y = element_text(size = 14),
    axis.text.x = element_text(size = 14)
  )

ggplot(data_clean, aes(y = mgp)) +
  geom_boxplot(fill = "lightblue", color = "black") +
  labs(title = "MGP", y = "") +
  theme(
    plot.title = element_text(size = 20, hjust = 0.5),  # Center the title horizontally
    axis.title.y = element_text(size = 16),
    axis.text.y = element_text(size = 14),
    axis.text.x = element_text(size = 14)
  )
ggplot(data_clean, aes(y = (log10(mgp))^2)) +
  geom_boxplot(fill = "lightblue", color = "black") +
  labs(title = "Log MGP", y = "") +
  theme(
    plot.title = element_text(size = 20, hjust = 0.5),  # Center the title horizontally
    axis.title.y = element_text(size = 16),
    axis.text.y = element_text(size = 14),
    axis.text.x = element_text(size = 14)
```

```
  )
ggplot(data_clean, aes(y = weight)) +
  geom_boxplot(fill = "lightblue", color = "black") +
  labs(title = "Weight", y = "") +
  theme(
    plot.title = element_text(size = 20, hjust = 0.5),  # Center the
   title horizontally
    axis.title.y = element_text(size = 16),
    axis.text.y = element_text(size = 14),
    axis.text.x = element_text(size = 14)
  )
```

Listing 5: Visualize dataset

```
# Descriptive statistics for 'model_year'
pie(table(model_year))
plot(table(model_year))

## Calculate the frequency of each 'origin'
origin_counts <- table(data$model_year)

## Calculate the proportions
origin_proportions <- prop.table(origin_counts)

## Print the proportions
print(origin_proportions)


# Descriptive statistics for 'cylinders'
pie(table(cylinders))
barplot(table(cylinders))


# Descriptive statistics for 'mgp'
hist(mgp, breaks = 50)

log_mgp = log10(data_clean$mgp)
hist(log_mgp, breaks = 25)

log_mgp_square = (log10(data_clean$mgp))^2
hist(log_mgp_square, breaks = 25)

data_clean$log_mgp_square <- log_mgp_square


# Boxplot of 'log_mgp_square' by 'origin'
boxplot(log_mgp_square ~ origin)


# Boxplot of 'log_mgp_square' by 'model_year'
boxplot(log_mgp_square ~ model_year)


# Boxplot of 'log_mgp_square' by 'cylinders'
```

```r
boxplot(log_mgp_square ~ cylinders)
# Boxplot of 'horsepower' by 'cylinders'
boxplot(horsepower ~ cylinders, data = data_clean)


# Boxplot of 'displacement' by 'cylinders'
boxplot(displacement ~ cylinders)


# Scatter plot of 'log_mgp_square' vs. 'horsepower'
plot(log_mgp_square ~ horsepower)
# Scatter plot of 'log_mgp_square' vs. 'displacement'
plot(log_mgp_square ~ displacement)
# Scatter plot of 'log_mgp_square' vs. 'weight'
plot(log_mgp_square ~ weight)
# Scatter plot of 'log_mgp_square' vs. 'acceleration'
plot(log_mgp_square ~ acceleration)
```

Listing 6: Descriptive statistics among variables

A.1.6. Process categorical variables

```r
min(model_year)
max(model_year)
# Convert year from 1970-1982 to 1-13
data_clean$model_year <- data_clean$model_year - 1970 + 1

# Create dummy variables for 'origin' column
data_clean$north_american <- ifelse(data_clean$origin == 1, 1, 0)
data_clean$europe <- ifelse(data_clean$origin == 2, 1, 0)
data_clean$origin <- NULL
```

Listing 7: Process categorical variables

A.1.7. Split data to train and test

```r
set.seed(1)
sample_size <- floor(0.8 * nrow(data_clean))
train_indices <- sample(seq_len(nrow(data_clean)), size = sample_size)
data_train <- data_clean[train_indices, ]
data_test <- data_clean[-train_indices, ]
detach(data_clean)
attach(data_train)
```

Listing 8: Split data to train and test

A.1.8. Checking multicolinearity

```r
cor_matrix <- cor(data_train)
cor_matrix

corrplot(cor_matrix, method = "color", type = "upper",
         tl.col = "black", tl.srt = 45, addCoef.col = "black")

# Convert the correlation matrix to long format for ggplot2
```

```r
cor_matrix_melted <- melt(cor_matrix)

# Visualize with ggplot2
ggplot(data = cor_matrix_melted, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile(color = "white") +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1, 1), space = "Lab",
                       name = "Correlation") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 12, hjust = 1)) +
  coord_fixed()

model1 <- lm(log_mgp_square ~. -mgp, data = data_train)
summary(model1)
car::vif(model1)

model_displacement <- lm(displacement ~ . -mgp -log_mgp_square, data = data_train)
summary(model_displacement)

# Remove displacement
model1 <- lm(log_mgp_square ~ . -mgp -displacement, data = data_train)
summary(model1)
car::vif(model1)

# Remove weight
model1 <- lm(log_mgp_square ~ . -mgp -displacement -weight, data = data_train)
summary(model1)
# Remove horsepower not weight
model1 <- lm(log_mgp_square ~ . -mgp -displacement -horsepower, data = data_train)
summary(model1)

# Remove cylinders
model1 <- lm(log_mgp_square ~ . -mgp -displacement -horsepower -cylinders, data = data_train)
summary(model1)
model_cylinders <- lm(cylinders ~ . -mgp -log_mgp_square -displacement -horsepower, data = data_train)
summary(model_cylinders)
car::vif(model1)
```

Listing 9: Checking multicolinearity

A.1.9. Variable selection

```r
modFull = lm(log_mgp_square ~ . -mgp -displacement -horsepower -cylinders, data = data_train)
modZero = lm(log_mgp_square ~ 1, data = data_train)
modInter = lm(log_mgp_square ~ weight + model_year, data = data_train)

```

```r
model2 = MASS::stepAIC(modInter, direction = "both", scope = list(lower
    = modZero, upper = modFull), k = 2)

model3 = MASS::stepAIC(modInter, direction = "both", scope = list(lower
    = modZero, upper = modFull), k = log(nrow(data_train)))

anova(model2, model3)
```

Listing 10: Variable selection

A.1.10. Model diagnostic

```r
lmtest::dwtest(model2)

# Use Shapiro-Wilk test to test for normality of the residuals
shapiro.test(residuals(model2))

# Plot residuals to visually check for normality
par(mfrow=c(1,2))

hist_residuals <- residuals(model2)
hist(hist_residuals, main = "Residuals Histogram", xlab = "Residuals",
    breaks = 30, probability = TRUE)

mean_residuals <- mean(hist_residuals)
sd_residuals <- sd(hist_residuals)
curve(dnorm(x, mean = mean_residuals, sd = sd_residuals), col = "red",
    add = TRUE)

qqnorm(hist_residuals, main = "Q-Q Plot of Residuals")
qqline(hist_residuals, col = "red")

bp_test <- bptest(model3)
print(bp_test)
```

Listing 11: Model diagnostic for model AIC

```r
lmtest::dwtest(model_cox)

# Check normality of residuals
residuals <- residuals(model_cox)

## Shapiro-Wilk test for residuals normality
shapiro_test <- shapiro.test(residuals)
print(shapiro_test)

# Plot residuals to visually check for normality
par(mfrow=c(1,2))

hist_residuals <- residuals(model_cox)
hist(hist_residuals, main = "Residuals Histogram", xlab = "Residuals",
    breaks = 30, probability = TRUE)

mean_residuals <- mean(hist_residuals)
```

```r
sd_residuals <- sd(hist_residuals)
curve(dnorm(x, mean = mean_residuals, sd = sd_residuals), col = "red",
    add = TRUE)

qqnorm(hist_residuals, main = "Q-Q Plot of Residuals")
qqline(hist_residuals, col = "red")

# Breusch-Pagan test to check for heteroscedasticity (constant variance
    of residuals)
bp_test <- bptest(model_cox)
print(bp_test)
```

Listing 12: Model diagnostic for model after using Box-cox transformation

A.1.11. Box-cox transformation

```r
boxcox_result <- boxcox(model2, plotit = TRUE)
lambda <- boxcox_result$x
log_likelihood <- boxcox_result$y

# Find the lambda with the maximum log-likelihood
best_lambda <- lambda[which.max(log_likelihood)]

# Print the best lambda
print(best_lambda)

# Build the final model with Box-Cox transformation
best_lambda = 0.5
model_cox = lm((((data_train$log_mgp_square^best_lambda) - 1)/best_
    lambda) ~ data_train$weight + data_train$model_year + data_train$
    north_american + data_train$acceleration)
summary(model_cox)
```

Listing 13: Box-cox transformation

A.1.12. Cross validation k-folds

```r
set.seed(1)
# Define a train control with k-fold cross-validation
train_control <- trainControl(method = "cv", number = 10)  # 10-fold
    cross-validation

# Define the formula for the model
formula <- as.formula(paste0("((log_mgp_square^", best_lambda, " - 1)/",
     best_lambda, ") ~ weight + model_year + north_american"))

# Train the model using the training data
cv_model <- train(formula, data = data_train, method = "lm", trControl =
     train_control)

# Predict using the cross-validated model on the test data
predictions_log <- predict(cv_model, newdata = data_test)

# Calculate performance metrics on the test data
actual_values <- 10^(sqrt(data_test$log_mgp_square))
```

```r
predict_values <- 10^(sqrt((predictions_log * best_lambda + 1)^(1/best_lambda)))
results <- data.frame(
  Actual = actual_values,
  Predicted = predict_values
)

# Print the results
print(results)

# Calculate and print RMSE and R-squared
rmse <- sqrt(mean((results$Actual - results$Predicted)^2))
r_squared <- cor(results$Actual, results$Predicted)^2
cat("RMSE:", rmse, "\n")
cat("R-squared:", r_squared, "\n")
```

Listing 14: Cross validation k-folds

A.2. Activity 2: Happiness

A.2.1. Process categorical data

```r
library(dplyr)

# Read the CSV files into data frames
df_a <- read.csv("world-happiness-report-2021.csv")
df_b <- read.csv("world-happiness-report.csv")

# Create a mapping from df_a
country_regions <- df_a %>%
  dplyr::select(Country, Region) %>%
  distinct()

# Add the 'Region' column to df_b
df_b <- df_b %>%
  left_join(country_regions, by = "Country")

# Reorder the column (move to after the Country column)
df_b <- df_b %>% relocate(Region, .after = Country)
write.csv(df_b, "world-happiness-report-with-regions.csv", row.names = FALSE)
```

Listing 15: Process categorical data

A.2.2. Process missing values

```r
data_clean <- data %>%
  group_by(Country) %>%
  mutate(across(where(is.numeric), ~ ifelse(is.na(.), mean(., na.rm = TRUE), .)))
data_clean <- na.omit(data_clean)
data_clean <- unique(data_clean)
dim(data_clean)
str(data_clean)
```

Listing 16: Process missing values

A.2.3. Import dataset

```
data <- read.csv("world-happiness-report-with-regions.csv", header =
    TRUE, sep = ",")
```

Listing 17: Import dataset

A.2.4. Split data to train and test

```
set.seed(1)
sample_size <- floor(0.8 * nrow(data_clean))
train_indices <- sample(seq_len(nrow(data_clean)), size = sample_size)
data_train <- data_clean[train_indices, ]
data_test <- data_clean[-train_indices, ]
detach(data_clean)
attach(data_train)
```

Listing 18: Split data to train and test

A.2.5. Process outliers using Cook's distance

```
model <- lm(Life.Ladder ~ . -Country - Region, data = data_clean)

# Calculate Cook's distance
cooksd <- cooks.distance(model)

# Plot Cook's distance
plot(cooksd, pch="*", cex=2, main="Cook's distance", ylab="Cook's
    distance")
abline(h = 4/length(cooksd), col="red")  # Add a horizontal line at 4/n
text(x=1:length(cooksd)+1, y=cooksd, labels=ifelse(cooksd>4/length(
    cooksd),names(cooksd),""), col="red")

# Identify influential points
influential <- as.numeric(names(cooksd)[(cooksd > 4/length(cooksd))])

# Optionally, remove influential points from the dataset
data_clean <- data_clean[-influential, ]
```

Listing 19: Cook's distance

A.2.6. Add region as a factor

```
data_clean <- as.data.frame(data_clean)
data_clean$Region <- as.factor(data_clean$Region)

# Create dummy variables
dummy_vars <- model.matrix(~ Region, data = data_clean)

# Remove the intercept column
dummy_vars <- dummy_vars[, -1]

# Combine the dummy variables with the original data frame
data_clean <- cbind(data_clean, dummy_vars)

```

```r
data_clean <- dplyr::select(data_clean, -Country)
data_clean <- dplyr::select(data_clean, -Region)
# Display the first few rows of the data frame with dummy variables
head(data_clean)
```

Listing 20: Add Region as a dummy variable

A.2.7. Check multicollinearity

```r
model = lm(Life.Ladder ~ . , data = data_train)
summary(model)
# Print the VIF values
vif(model)
print(max(vif(model)))
# Test the collinearity of the healthy life variable
test <- lm(Healthy.life.expectancy.at.birth ~ . - Life.Ladder, data =
    data_train)
summary(test)
# R^2 = 0.85 -> high collinearity -> remove the variable
# Check VIF values without the healthy life variable
model <- lm(Life.Ladder ~ . - Healthy.life.expectancy.at.birth, data =
    data_train)
vif(model)
```

Listing 21: Check multicollinearity

A.2.8. Variable selection

```r
# Initialize 3 models for the algorithm
modFull = lm(Life.Ladder ~ . - Healthy.life.expectancy.at.birth, data =
    data_train)
modZero = lm(Life.Ladder ~ 1, data = data_train)
modInter = lm(Life.Ladder~ Log.GDP.per.capita + Social.support + Freedom
    .to.make.life.choices + Generosity
+ Perceptions.of.corruption + Positive.affect + Negative.affect, data =
    data_train)
# AIC algorithm
model2 = MASS::stepAIC(modInter, direction = "both", scope = list(lower
    = modZero, upper = modFull), k = 2)
summary(model2)
# BIC algorithm
model3 = MASS::stepAIC(modInter, direction = "both", scope = list(lower
    = modZero, upper = modFull), k = log(nrow(data_train)))
summary(model3)
# Compare the models using F-partial test
anova(model2, model3)
```

Listing 22: Variable selection

A.2.9. Box-cox transformation

```r
# Life.Ladder ~ Log.GDP.per.capita + Social.support + Freedom.to.make.
    life.choices +
#    Generosity + Perceptions.of.corruption + Positive.affect +
#    `RegionLatin America and Caribbean` + `RegionSoutheast Asia` +
```

```
#    `RegionSub-Saharan Africa` + `RegionWestern Europe` + `RegionNorth America and ANZ`+ `RegionCommonwealth of Independent States`
model_cox = lm(Life.Ladder~ Log.GDP.per.capita + Social.support + Freedom.to.make.life.choices +
    Generosity + Perceptions.of.corruption + Positive.affect + `RegionLatin America and Caribbean` , data = data_train)
summary(model_cox)
boxcox_result <- boxcox(model_cox, plotit = TRUE)
lambda <- boxcox_result$x
log_likelihood <- boxcox_result$y
best_lambda <- lambda[which.max(log_likelihood)]

# Print the best lambda
print(best_lambda)

best_lambda = 1.8
model_cox = lm(((((Life.Ladder)^best_lambda) - 1)/best_lambda) ~ Log.GDP.per.capita + Social.support + Freedom.to.make.life.choices +
    Generosity + Perceptions.of.corruption + Positive.affect + `RegionLatin America and Caribbean` , data = data_train)
summary(model_cox)
```

Listing 23: Box-cox transformation

A.2.10. Model diagnostic

```
# Normality test
shapiro.test(residuals(model2))
par(mfrow=c(1,2))

hist_residuals <- residuals(model2)
hist(hist_residuals, main = "Residuals Histogram", xlab = "Residuals", breaks = 100, probability = TRUE)

mean_residuals <- mean(hist_residuals)
sd_residuals <- sd(hist_residuals)
curve(dnorm(x, mean = mean_residuals, sd = sd_residuals), col = "red", add = TRUE)

qqnorm(hist_residuals, main = "Q-Q Plot of Residuals")
qqline(hist_residuals, col = "red")
# Homoscedasticity test
bp_test <- bptest(model2)
print(bp_test)
```

Listing 24: Model diagnostic

A.2.11. Cross validation k-folds

```
set.seed(1)
# Define a train control with k-fold cross-validation
train_control <- trainControl(method = "cv", number = 10)  # 10-fold cross-validation

# Define the formula for the model
```

```r
formula <- as.formula(paste0("((Life.Ladder^", best_lambda, " - 1)/",
    best_lambda, ") ~ Log.GDP.per.capita + Social.support + Freedom.to.
    make.life.choices + Generosity + Perceptions.of.corruption + Positive
    .affect + `RegionLatin America and Caribbean`"))

# Train the model using the training data
cv_model <- train(formula, data = data_train, method = "lm", trControl =
     train_control)

# Predict using the cross-validated model on the test data
predictions <- predict(cv_model, newdata = data_test)

# Calculate performance metrics on the test data
actual_values <- data_test$Life.Ladder
predict_values <- (predictions*best_lambda + 1) ^ (1/best_lambda)
results <- data.frame(
  Actual = actual_values,
  Predicted = predict_values
)

# Print the results
print(results)

# Calculate and print RMSE and R-squared
rmse <- sqrt(mean((results$Actual - results$Predicted)^2))
r_squared <- cor(results$Actual, results$Predicted)^2
cat("RMSE:", rmse, "\n")
cat("R-squared:", r_squared, "\n")
```

Listing 25: Cross validation k-folds

A.3. Activity 2: Suicide

A.3.1. Import dataset

```r
data <- read.csv("master.csv", header = TRUE)

attach(data)
head(data)
str(data)
dim(data)
```

Listing 26: Import dataset

A.3.2. Rename columns

```r
data$gdp_for_year <- data$gdp_for_year....
data$gdp_for_year.... <- NULL

data$gdp_per_capita <- data$gdp_per_capita....
data$gdp_per_capita.... <- NULL
```

Listing 27: Rename columns

A.3.3. Process missing values

```r
# Check for missing values
missing_values <- sapply(data, function(x) sum(is.na(x)))
print(missing_values)

# Remove unnecessary variables
data_clean <- data
data_clean <- dplyr::select(data_clean, -HDI.for.year, -country.year, -
    suicides.100k.pop)

#Change the format for gdp_for_year
data_clean <- data_clean %>%
mutate(gdp_for_year = as.numeric(gsub(",", "", gdp_for_year)))
```

Listing 28: Process missing values

A.3.4. Process unnecessary columns

```r
data_clean <- data
data_clean <- dplyr::select(data_clean, -HDI.for.year, -country.year, -
    suicides.100k.pop)
```

Listing 29: Process unnecessary columns

A.3.5. Convert data types

```r
# Convert gdp_for_year to numeric after removing commas
data_clean <- data_clean %>%
mutate(gdp_for_year = as.numeric(gsub(",", "", gdp_for_year)))

str(data_clean)
detach(data)
attach(data_clean)
```

Listing 30: Convert data types

A.3.6. Process duplicate rows

```r
duplicate_rows <- data_clean[duplicated(data_clean), ]
print(duplicate_rows)
```

Listing 31: Process duplicate rows

A.3.7. Normalize variables

```r
# Process 'suicides_no' column
data_clean$log_suicides_no <- log(suicides_no + 1) + 1
data_clean$suicides_no <- NULL

# Process 'gdp_per_capita' column
data_clean$transform_gdp_per_capita <- log(gdp_per_capita)^2
data_clean$gdp_per_capita <- NULL

# Process 'population' column
data_clean$transform_population <- log(population)^2
data_clean$population <- NULL

```

```r
# Process 'gdp_per_capita' column
data_clean$transform_gdp_for_year <- log(gdp_for_year)
data_clean$gdp_for_year <- NULL
```

Listing 32: Normalize variables

A.3.8. Process variables

```r
# Process 'year' column
data_clean$year <- data_clean$year - 1985 + 1

## Create dummy variables for the 'country' column
data_clean$regionEurope <- ifelse(data_clean$country %in% c("Austria", "Iceland", "Netherlands", "Belgium", "Bulgaria", "France", "Greece", "Ireland", "Italy", "Luxembourg", "Malta", "Norway", "Portugal", "Romania", "Spain", "Sweden", "United Kingdom", "Ukraine", "Finland", "Switzerland", "Serbia", "Slovenia", "Slovakia", "Albania", "Denmark", "Estonia", "Latvia", "Lithuania", "Belarus", "Croatia", "Czech Republic", "Germany", "Hungary", "Poland", "San Marino", "Bosnia and Herzegovina"), 1, 0)

data_clean$regionAsia <- ifelse(data_clean$country %in% c("Israel", "Japan", "Republic of Korea", "Singapore", "Turkmenistan", "Thailand", "Russian Federation", "Kazakhstan", "Kyrgyzstan", "Armenia", "Azerbaijan", "Philippines", "Cyprus", "Qatar", "Sri Lanka", "Maldives", "Turkey", "United Arab Emirates", "Oman", "Bahrain", "Uzbekistan", "Georgia", "Macau"), 1, 0)

data_clean$regionAfrica <- ifelse(data_clean$country %in% c("Mauritius", "South Africa", "Seychelles", "Cabo Verde"), 1, 0)

data_clean$regionNorthAmerica <- ifelse(data_clean$country %in% c("Canada", "Costa Rica", "Guatemala", "Mexico", "Puerto Rico", "United States", "Belize", "Saint Lucia", "Antigua and Barbuda", "Trinidad and Tobago", "Panama", "Saint Vincent and Grenadines", "Cuba", "El Salvador", "Bahamas", "Jamaica", "Saint Kitts and Nevis", "Dominica"), 1, 0)

data_clean$regionSouthAmerica <- ifelse(data_clean$country %in% c("Argentina", "Brazil", "Chile", "Colombia", "Ecuador", "Paraguay", "Suriname", "Uruguay", "Guyana"), 1, 0)

data_clean$regionOceania <- ifelse(data_clean$country %in% c("Australia", "New Zealand", "Fiji", "Kiribati", "Montenegro"), 1, 0)

## Remove the 'country' column after creating 'region'
data_clean$country <- NULL

## Create dummy variables for the 'sex' column
data_clean$sexMale <- ifelse(data_clean$sex == "male", 1, 0)
data_clean$sex <- NULL

## Create dummy variables for the 'generation' column
data_clean$geneX <- ifelse(data_clean$generation == "Generation X", 1,
```

```
    0)
data_clean$geneMillenials <- ifelse(data_clean$generation == "Millenials
    ", 1, 0)
data_clean$geneBoomers <- ifelse(data_clean$generation == "Boomers", 1,
    0)
data_clean$geneSilent <- ifelse(data_clean$generation == "Silent", 1, 0)
data_clean$generation <- NULL
# The other generation would be G.I. Generation

## Create dummy variables for the 'age' column
data_clean$Age15to24 <- ifelse(data_clean$age == "15-24 years", 1, 0)
data_clean$Age25to34 <- ifelse(data_clean$age == "25-34 years", 1, 0)
data_clean$Age35to54 <- ifelse(data_clean$age == "35-54 years", 1, 0)
data_clean$Age55to74 <- ifelse(data_clean$age == "55-74 years", 1, 0)
data_clean$age <- NULL
#The other generation would be 75+ years
```

Listing 33: Process variables

A.3.9. Cook's Distance

```
# Calculate Cook's distance
model <- lm(log_suicides_no ~ ., data = data_clean)
cooksD <- cooks.distance(model)

# Identify influential observations
influential <- cooksD[(cooksD > (3/length(cooksD)))]

# Extract the indices of influential observations
outliers <- as.numeric(names(influential))

# Remove influential observations to clean the data
data_clean <- data_clean[-outliers, ]
```

Listing 34: Cook's Distance

A.3.10. Descriptive statistics

```
# Descriptive statistics for the 'suicides_no' column
hist(suicides_no, breaks = 30)
log_suicides_no <- log(suicides_no + 1) + 1

hist(log_suicides_no, breaks = 30)

boxplot(log_suicides_no)

# Descriptive statistics for the 'year' column
boxplot(year)
hist(year, breaks = 20)

# Descriptive statistics for 'sex' column
pie(table(sex))
barplot(table(sex))

# Descriptive Statistics for 'generation' and 'age'
```

```r
barplot(table(generation))
pie(table(generation))
barplot(table(age))
pie(table(age))

# Descriptive statistics for 'population'
boxplot(log(population))
hist(population, breaks = 30)
hist(log(population), breaks = 30)
hist(log(population)^2, breaks = 30)

# Descriptive statistics for 'gdp_per_capita'
hist(gdp_per_capita, breaks = 50)
hist(log(gdp_per_capita), breaks = 50)
hist(log(gdp_per_capita)^2, breaks = 50)

# Descriptive statistics for 'gdp_for_year'
boxplot(log(gdp_for_year))
hist(gdp_for_year, breaks = 30)
hist(log(gdp_for_year), breaks = 30)
hist(log(gdp_for_year)^2, breaks = 30)
```

Listing 35: Descriptive statistics for variables

```r
log_suicides_no <- log(suicides_no + 1) + 1
log_population_sq <- log(population)^2
log_gdp_per_capita_sq <- log(gdp_per_capita)^2
log_gdp_for_year <- log(gdp_for_year)

plot(log_suicides_no ~ log_population_sq)
plot(log_suicides_no ~ log_gdp_per_capita_sq)
plot(log_suicides_no ~ log_gdp_for_year)


ggplot(data_clean, aes(x = age, y = log_suicides_no, fill = age)) +
  geom_boxplot() +
  scale_fill_viridis_d() +  # Use viridis color palette
  labs(title = "log_suicides_no by age",
       y = "log_suicides_no",
       x = NULL) +  # Remove x-axis label as it's redundant
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1),
    legend.position = "none",  # Remove legend as colors are already on x-axis
    plot.title = element_text(hjust = 0.5, size = 16),
    axis.title.y = element_text(size = 12)
  )

ggplot(data_clean, aes(x = generation, y = log_suicides_no, fill = generation)) +
  geom_boxplot() +
  scale_fill_viridis_d() +  # Use viridis color palette
  labs(title = "log_suicides_no by generation",
```

```
      y = "log_suicides_no",
      x = NULL) +  # Remove x-axis label as it's redundant
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1),
    legend.position = "none",  # Remove legend as colors are already on
   x-axis
    plot.title = element_text(hjust = 0.5, size = 16),
    axis.title.y = element_text(size = 12)
  )

ggplot(data_clean, aes(x = sex, y = log_suicides_no, fill = sex)) +
  geom_boxplot() +
  scale_fill_viridis_d() +  # Use viridis color palette
  labs(title = "log_suicides_no by sex",
      y = "log_suicides_no",
      x = NULL) +  # Remove x-axis label as it's redundant
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1),
    legend.position = "none",  # Remove legend as colors are already on
   x-axis
    plot.title = element_text(hjust = 0.5, size = 16),
    axis.title.y = element_text(size = 12)
  )

boxplot(log_suicides_no ~ year)
```

Listing 36: Descriptive statistics for relationship between response variable and predictors

A.3.11. Split data to train and test

```
set.seed(1)
sample_size <- floor(0.8 * nrow(data_clean))
train_indices <- sample(seq_len(nrow(data_clean)), size = sample_size)
data_train <- data_clean[train_indices, ]
data_test <- data_clean[-train_indices, ]
detach(data_clean)
attach(data_train)
```

Listing 37: Split data to train and test

A.3.12. Checking multicolinearity

```
model = lm(log_suicides_no ~ ., data = data_train)
summary(model)
car::vif(model)

model = lm(log_suicides_no ~ . -transform_gdp_for_year , data = data_
   train)
summary(model)
car::vif(model)

cor(data_train$log_suicides_no, data_train$regionEurope)
```

```
model_regionEurope = lm(regionEurope ~ . -transform_gdp_for_year -log_suicides_no, data = data_train)
summary(model_regionEurope)

model = lm(log_suicides_no ~ . -transform_gdp_for_year -regionEurope, data = data_train)
summary(model)
car::vif(model)
```

Listing 38: Checking multicolinearity

A.3.13. Variable selection

```
modFull = lm(log_suicides_no ~ . -transform_gdp_for_year -regionEurope, data = data_train)
modZero = lm(log_suicides_no ~ 1, data = data_train)
modInter = lm(log_suicides_no ~ year + transform_population + regionAsia + regionNorthAmerica + sexMale + geneBoomers + Age25to34, data = data_train)

modelAIC = MASS::stepAIC(modFull, direction = "backward", scope = list(lower = modZero, upper = modFull), k = 2)
summary(modelAIC)
```

Listing 39: AIC model

A.3.14. Model diagnostic

```
lmtest::dwtest(modelAIC)

# Use Shapiro-Wilk test to test for normality of the residuals
set.seed(22) # For reproducibility
subsample_residuals <- sample(residuals(modelAIC), size = 5000)
shapiro.test(subsample_residuals)
# Plot residuals to visually check for normality
par(mfrow=c(1,2))
hist_residuals <- residuals(modelAIC)
hist(hist_residuals, main = "Residuals Histogram", xlab = "Residuals", breaks = 30, probability = TRUE)
mean_residuals <- mean(hist_residuals)
sd_residuals <- sd(hist_residuals)
curve(dnorm(x, mean = mean_residuals, sd = sd_residuals), col = "red", add = TRUE)
qqnorm(hist_residuals, main = "Q-Q Plot of Residuals")
qqline(hist_residuals, col = "red")

bp_test <- lmtest::bptest(modelAIC)
print(bp_test)
```

Listing 40: Model diagnostic (model AIC)

```
lmtest::dwtest(model_cox)

# Use Shapiro-Wilk test to test for normality of the residuals
set.seed(22) # For reproducibility
```

```r
subsample_residuals <- sample(residuals(model_cox), size = 5000)
shapiro.test(subsample_residuals)

# Plot residuals to visually check for normality
par(mfrow=c(1,2))
hist_residuals <- residuals(model_cox)
hist(hist_residuals, main = "Residuals Histogram", xlab = "Residuals", breaks = 30, probability = TRUE)
mean_residuals <- mean(hist_residuals)
sd_residuals <- sd(hist_residuals)
curve(dnorm(x, mean = mean_residuals, sd = sd_residuals), col = "red", add = TRUE)
qqnorm(hist_residuals, main = "Q-Q Plot of Residuals")
qqline(hist_residuals, col = "red")

# Studentized Breusch-Pagan test for heteroscedasticity
bp_test <- lmtest::bptest(model_cox)
print(bp_test)
```

Listing 41: Model diagnostic (after using box-cox transformation)

A.3.15. Box-cox transformation

```r
boxcox_result <- boxcox(modelAIC, plotit = TRUE)
lambda <- boxcox_result$x
log_likelihood <- boxcox_result$y

# Find the lambda with the maximum log-likelihood
best_lambda <- lambda[which.max(log_likelihood)]

# Print the best lambda
print(best_lambda)

# Build the final model with Box-Cox transformation
best_lambda = 1.5
model_cox = lm((((data_train$log_suicides_no^best_lambda) - 1)/best_lambda) ~
    year + transform_gdp_per_capita +
    transform_population + regionAsia + regionAfrica + regionNorthAmerica +
    regionSouthAmerica + sexMale + geneX + geneMillenials + geneBoomers +
    geneSilent + Age15to24 + Age25to34 + Age35to54 + Age55to74, data = data_train)
summary(model_cox)
```

Listing 42: Box-cox transformation

A.3.16. Cross validation k-folds

```r
set.seed(1)
# Define a train control with k-fold cross-validation
train_control <- trainControl(method = "cv", number = 10)  # 10-fold cross-validation

```

```r
# Define the formula for the model
formula <- as.formula(paste0("((log_suicides_no^", best_lambda, " - 1)/", best_lambda, ") ~
    year + transform_gdp_per_capita +
    transform_population + regionAsia + regionAfrica + regionNorthAmerica +
    regionSouthAmerica + sexMale + geneX + geneMillenials + geneBoomers +
    geneSilent + Age15to24 + Age25to34 + Age35to54 + Age55to74"))

# Train the model using the training data
cv_model <- train(formula, data = data_train, method = "lm", trControl = train_control)

# Predict using the cross-validated model on the test data
predictions_log <- predict(cv_model, newdata = data_test)

# Calculate performance metrics on the test data
actual_values <- exp(data_test$log_suicides_no - 1) -1
predict_values <- round(exp((ifelse(predictions_log < 0, 0, predictions_log)  * best_lambda + 1)^(1/best_lambda) -1) -1, 2)
results <- data.frame(
  Actual = actual_values,
  Predicted = predict_values
)

# Print the results
print(results)

# Calculate and print RMSE and R-squared
rmse <- sqrt(mean((results$Actual - results$Predicted)^2))
r_squared <- cor(results$Actual, results$Predicted)^2
cat("RMSE:", rmse, "\n")
cat("R-squared:", r_squared, "\n")
```

Listing 43: Cross validation k-folds