VIETNAM NATIONAL UNIVERSITY - HO CHI MINH CITY

UNIVERSITY OF SCIENCE

Faculty of Information Technology

Applied Statistics for Engineers and Scientists II

Final Project Report

Group 6

Subject: Applied Statistics for Engineers and Scientists II

Student name:	*Student ID:*
Luu Quoc Bao	22125008
Le Minh Hoang	22125029
Le Duc Nhuan	22125070
Dang Minh Nhut	22125071

August 20, 2024

Contents

List of Tables

List of Figures

1 Task 1

1.1 Summary of data

There are a total of 398 rows and 9 columns in the data. The columns are ”mpg”, ”cylinders”, ”displacement”, ”horsepower”, ”weight”, ”acceleration”, ”model year”, ”origin”, and ”car name”. The description of each column is as follows:

- ”mpg”: (continuous) fuel consumption in miles per gallon,
- ”cylinders”: (multi-valued discrete) number of cylinders,
- ”displacement”: (continuous) engine size,
- ”horsepower”: (continuous) engine power,
- ”weight”: (continuous) mass,
- ”acceleration”: (continuous) vehicle acceleration,
- ”model year”: (multi-valued discrete) model year (last 2 digits)
- ”origin”: (multi-valued discrete) place of manufacture: 1 - North American, 2 - Europe, 3 - Asia
- ”car name”: (multi-valued discrete) car name

From the meaning of the variables, we have some predictions on the relationship between the fuel consumption and other variables.

- ”cylinders” (number of cylinders): the more cylinders the car has, the higher the fuel consumption.
- ”displacement” (engine size): the larger the engine, the higher the fuel consumption.
- ”horsepower” (engine power): the more powerful the car, the higher the fuel consumption.
- ”weight”: the heavier the car is, the more fuel it consumes.
- ”acceleration”: the car need more fuel if it has bigger acceleration.
- ”model_year”: as the time goes, the less fuel consumption, as the engines get more efficiency.

1.2 Data preprocessing

1.2.1 Removing ”car name”

The car name is not useful for our analysis, so we removed it.

1.2.2 Removing missing data

Observing the data, we see some rows with ’?’ values in the horsepower column. We remove these rows.

1.2.3 Convert horsepower to numeric

The horsepower column is not numeric yet, so we convert it to numeric.

1.2.4 Remove duplicate rows

There are no duplicate rows in the data. Therefore, we do not need to remove any rows.

1.2.5 Change variables to factor

We first plot bar plots of each column in Figure 1 to see if any variables should be changed to factors. As ”cylinders” and ”origin” are the two variables with small numbers of unique values, we start considering whether these become factors.

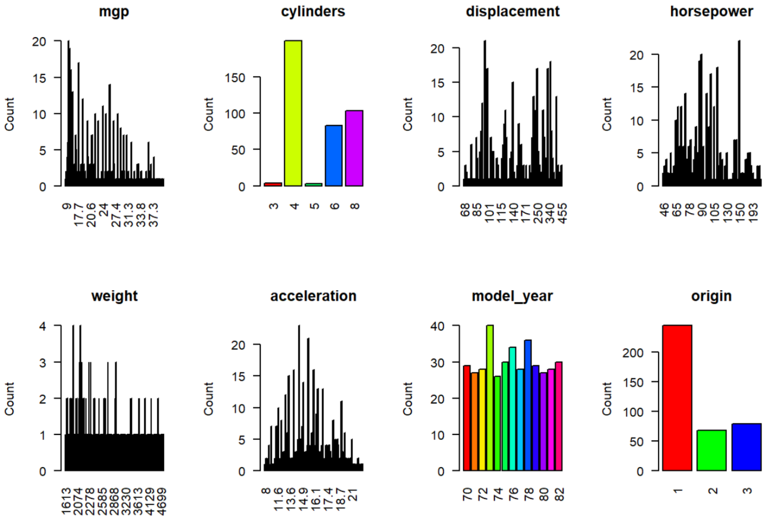


Figure 1: The bar plots of all variables in the data

We plot the bar plot of the ”cylinder” in Figure 2. Even though the ”cylinders” column has only 5 values, we **will NOT change it to a factor** because it is quantitatively meaningful.

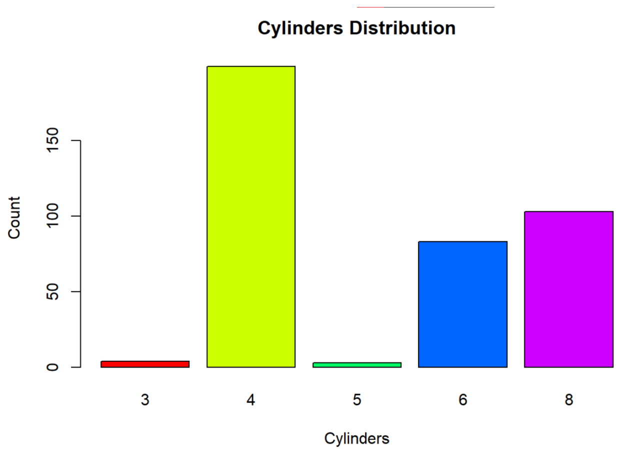


Figure 2: The bar plot of ”cylinders” variable

We plot the bar plot of ”origin” in Figure 3. As we can see, the ”origin” column has only 3 values. Moreover, these 3 values are not quantitatively meaningful. Therefore, we **will change the ”origin” column to a factor**.

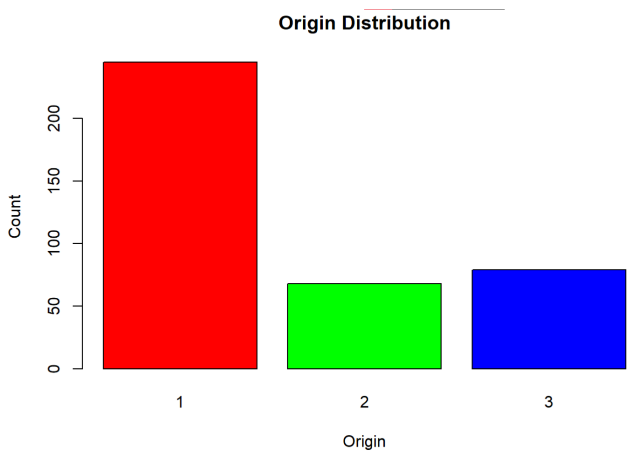


Figure 3: The bar plot of ”origin” variable

After data preprocessing, the data now has **392 rows and 8 columns**. Codes for data preprocessing can be found in Section A.1.

1.3 Data splitting

We split the data into training and testing sets. The ratio of the training set to the testing set is 80:20.

The training set now has 313 rows while the testing set has 79, which is approximately 80% and 20% of the original data, respectively.

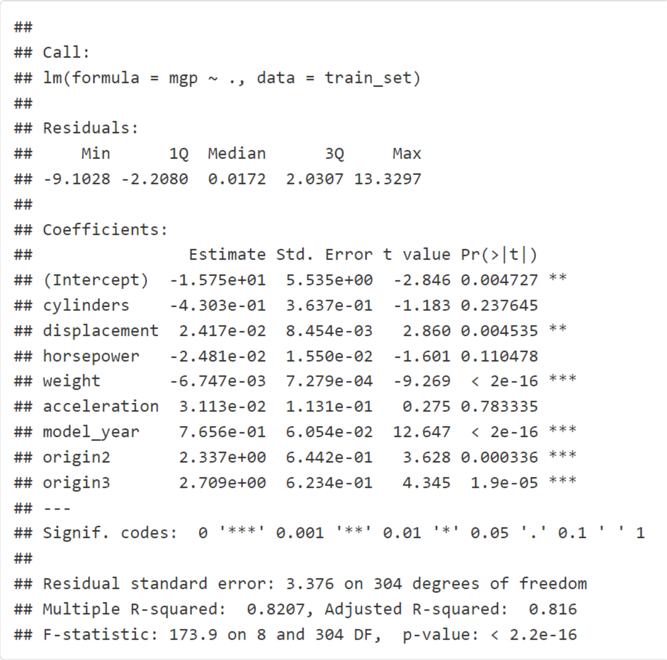
Code for data splitting can be found in Section A.2.

1.4 Baseline model

In this section, we try to create a baseline linear model. Codes for this section can be found in Section A.3.

We fit a linear regression model with all original variables to predict ”mpg” as in Equation (1). Figure 4 shows the summary of this model.

$$\begin{aligned}\text{mpg} =&\beta_0 + \beta_1 \times \text{cylinders} + \beta_2 \times \text{displacement} + \beta_3 \times \text{horsepower} + \beta_4 \times \text{weight} \\ &+ \beta_5 \times \text{acceleration} + \beta_6 \times \text{model\_year} + \beta_7 \times \text{origin2} + \beta_8 \times \text{origin3}\end{aligned} \tag{1}$$

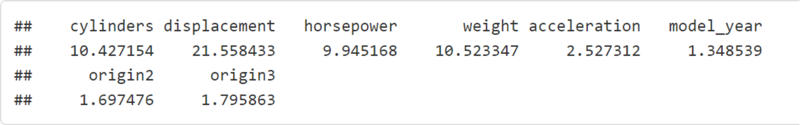


```
##
## Call:
## lm(formula = mgp ~ ., data = train_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.1028 -2.2080  0.0172  2.0307 13.3297
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.575e+01  5.535e+00  -2.846 0.004727 **
## cylinders    -4.303e-01  3.637e-01  -1.183 0.237645
## displacement  2.417e-02  8.454e-03   2.860 0.004535 **
## horsepower   -2.481e-02  1.550e-02  -1.601 0.110478
## weight       -6.747e-03  7.279e-04  -9.269  < 2e-16 ***
## acceleration  3.113e-02  1.131e-01   0.275 0.783335
## model_year    7.656e-01  6.054e-02  12.647  < 2e-16 ***
## origin2       2.337e+00  6.442e-01   3.628 0.000336 ***
## origin3       2.709e+00  6.234e-01   4.345  1.9e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.376 on 304 degrees of freedom
## Multiple R-squared:  0.8207, Adjusted R-squared:  0.816
## F-statistic: 173.9 on 8 and 304 DF,  p-value: < 2.2e-16
```

Figure 4: Summary of baseline model in Task 1

1.4.1 Multicollinearity

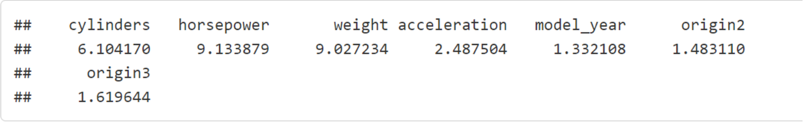
First, we need to check for multicollinearity of the model in Figure 4. We calculate the Variance Inflation Factor (VIF) of the variables as in Figure 5.



```
##    cylinders displacement   horsepower       weight acceleration   model_year
##    10.427154    21.558433     9.945168    10.523347     2.527312     1.348539
##      origin2      origin3
##     1.697476     1.795863
```

Figure 5: VIF of baseline model in Task 1

We see that the "displacement" variable has a high VIF value. Therefore, we will remove it from the model and recalculate the VIF of the reduced model as in Figure 6.



```
##    cylinders   horsepower       weight acceleration   model_year      origin2
##     6.104170     9.133879     9.027234     2.487504     1.332108     1.483110
##      origin3
##     1.619644
```

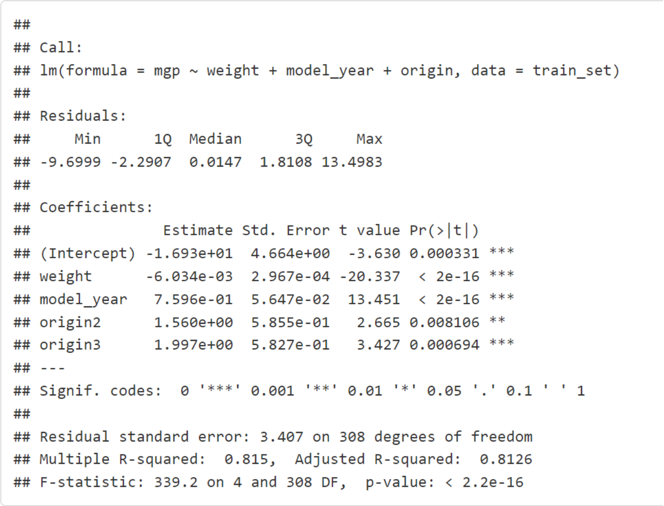
Figure 6: VIF of baseline model without "displacement" in Task 1

Now all variables have VIF values less than 10, indicating no strong multicollinearity in the model.

1.4.2 Stepwise Algorithm

We apply the Stepwise Algorithm to select the best model. The result can be found in Figure 7. The model can be expressed as in Equation (2).

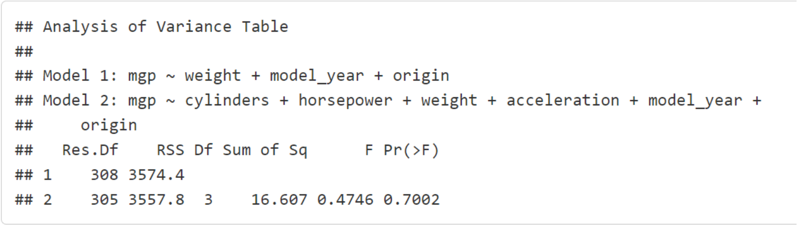
$$\text{mpg} = \beta_0 + \beta_1 \times \text{weight} + \beta_2 \times \text{model\_year} + \beta_3 \times \text{origin2} + \beta_4 \times \text{origin3} \tag{2}$$



```
##
## Call:
## lm(formula = mgp ~ weight + model_year + origin, data = train_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.6999 -2.2907  0.0147  1.8108 13.4983
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.693e+01  4.664e+00  -3.630 0.000331 ***
## weight      -6.034e-03  2.967e-04 -20.337  < 2e-16 ***
## model_year   7.596e-01  5.647e-02  13.451  < 2e-16 ***
## origin2      1.560e+00  5.855e-01   2.665 0.008106 **
## origin3      1.997e+00  5.827e-01   3.427 0.000694 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.407 on 308 degrees of freedom
## Multiple R-squared:  0.815,  Adjusted R-squared:  0.8126
## F-statistic: 339.2 on 4 and 308 DF,  p-value: < 2.2e-16
```

Figure 7: Summary of the baseline model after Stepwise in Task 1

We perform an ANOVA test to validate that the removed variables are not significant as in Figure 8. The p-value of the ANOVA test is much higher than the significance level of 0.05, indicating that the removed variables are not significant.



```
## Analysis of Variance Table
##
## Model 1: mgp ~ weight + model_year + origin
## Model 2: mgp ~ cylinders + horsepower + weight + acceleration + model_year +
##     origin
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    308 3574.4
## 2    305 3557.8  3    16.607 0.4746 0.7002
```

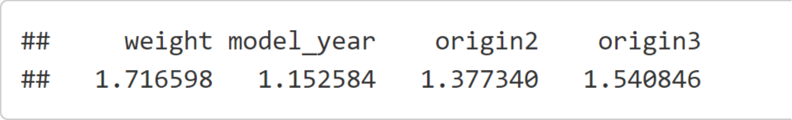
Figure 8: ANOVA for the result baseline model after Stepwise Algorithm in Task 1

1.4.3 Model diagnostics

We now conduct some diagnostics on the final baseline model to verify its ideality for the task.

- Multicollinearity

  We check for the multicollinearity of the variables in the model by applying the VIF test in Figure 9.



```
##     weight model_year    origin2    origin3
##   1.716598   1.152584   1.377340   1.540846
```

Figure 9: VIF result of variable in the final baseline model

Compared to the previous model, the VIF values of the variables are much lower, indicating no multicollinearity in the model.

- Normality of residuals

  We now plot the Q-Q plot (Figure 10) of the residuals to check for normality. The residuals are close to the line in the middle, suggesting that the residuals in this range are approximately normally distributed. However, the points at the ends (tails) of the plot deviate from the red line, especially on the right side where the points are higher than the line. This is potentially due to the presence of outliers in the data, which can affect the normality of the residuals.

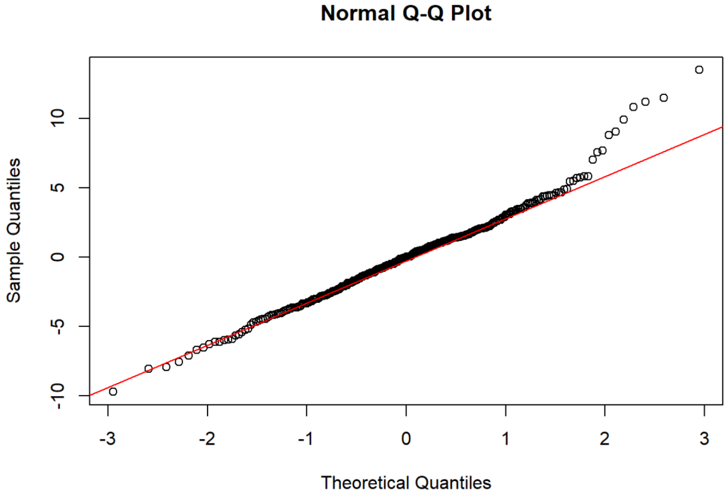
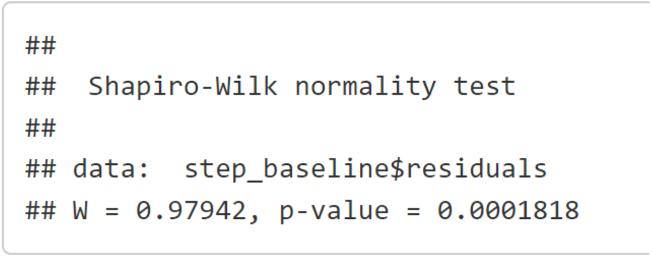


Figure 10: Q-Q plot of the residuals in the final baseline model in Task 1

To test the normality of the residuals, we perform the Shapiro-Wilk test in Figure 11. The p-value of the Shapiro-Wilk test is less than 0.05, indicating that the residuals are not normally distributed. This is consistent with the Q-Q plot, where the residuals have heavier tails than a normal distribution.

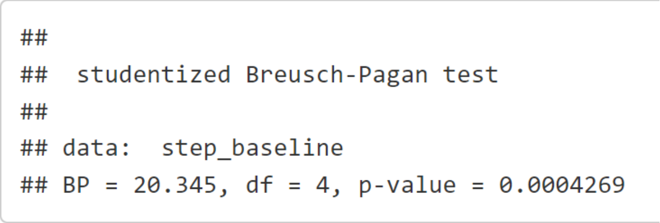


```
##
##  Shapiro-Wilk normality test
##
## data:  step_baseline$residuals
## W = 0.97942, p-value = 0.0001818
```

Figure 11: Shaprio-Wilk test of final baseline model in Task 1

- Homoscedasticity

  We also perform the Breusch-Pagan test to check for homoscedasticity as in Figure 12. The p-value of the Breusch-Pagan test is less than 0.05, indicating that the residuals are heteroscedastic.

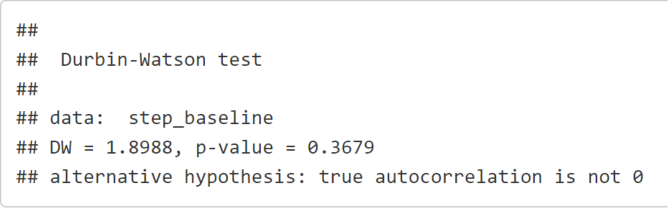


```
##
##  studentized Breusch-Pagan test
##
## data:  step_baseline
## BP = 20.345, df = 4, p-value = 0.0004269
```

Figure 12: Breusch-Pagan test of final baseline model in Task 1

- Autocorrelation

  We perform the Durbin-Watson test to check for autocorrelation in Figure 13. The Durbin-Watson test gives a DW value close to 2, indicating that there is no autocorrelation.



```
##
##  Durbin-Watson test
##
## data:  step_baseline
## DW = 1.8988, p-value = 0.3679
## alternative hypothesis: true autocorrelation is not 0
```

Figure 13: Durbin-Watson test of final baseline model in Task 1

- Conclusion on model diagnostics

  In conclusion, the model is not ideal as the residuals are not normally distributed and heteroscedastic. However, the model is still acceptable as the residuals are approximately normally distributed in the middle range and there is no autocorrelation in the residuals.

  However, we believe that the absence of ”horsepower” in the model is contradictory to the real-world relationship between ”mpg” and ”horsepower”, as **the more powerful the engine, the higher the fuel consumption**. This suggests that the model can be improved by including ”horsepower” in the model.

1.5 Improving the model

In this section, we try to improve our baseline model by adding new variables. Codes for this section can be found in Section A.4.

1.5.1 Adding log transformation

We investigate the relationship between ”mpg” and other variables by plotting box plots of all variables in Figure 14. We can observe that ”horsepower” has many outliers. Therefore, we will **remove these outliers using the Interquartile Range (IQR) method**. After this step, the training set has **296 rows**.

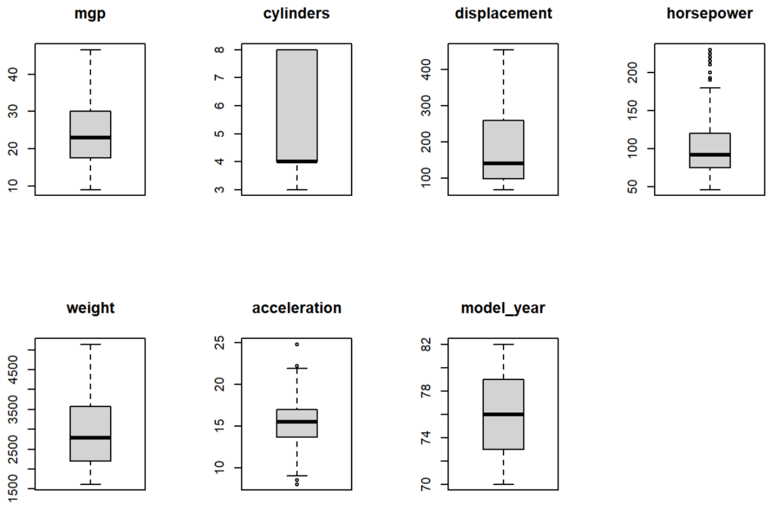


Figure 14: Box plots of all original variables in Task 1

We plot scatter plots of ”mpg” and other variables in Figure 15 to investigate further their relationships. We can see a curvilinear relationship between ”mpg” and ”horsepower”. Therefore, we will **include a log transformation of ”horsepower”** in the data. Not only aligning with the relationship between ”mpg” and ”horsepower”, the log transformation also helps to reduce the effect of outliers in the ”horsepower” column.

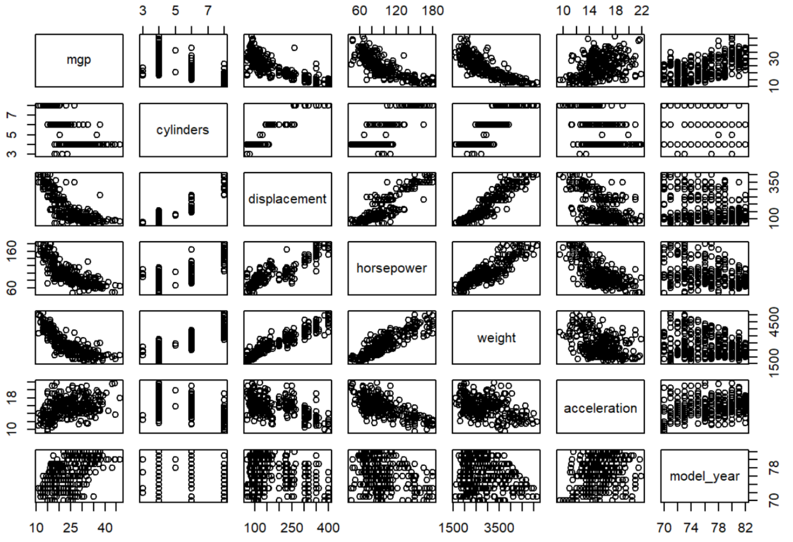
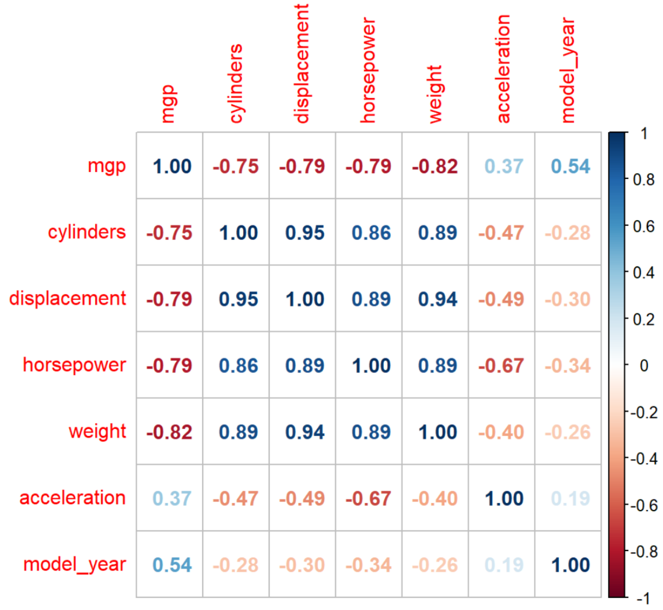


Figure 15: Box plots of all pairs of original variables in Task 1

Moreover, the scatter plots of "displacement" and "weight" with respect to "mpg" have similar patterns to that of "horsepower". To validate our assumption, we use a correlation matrix in Figure 16 to see the relationship between the variables. As "horsepower", "displacement", and "weight" are highly correlated, we will **include the log transformation of "displacement" and "weight"** in the data.



	mgp	cylinders	displacement	horsepower	weight	acceleration	model_year
mgp	1.00	-0.75	-0.79	-0.79	-0.82	0.37	0.54
cylinders	-0.75	1.00	0.95	0.86	0.89	-0.47	-0.28
displacement	-0.79	0.95	1.00	0.89	0.94	-0.49	-0.30
horsepower	-0.79	0.86	0.89	1.00	0.89	-0.67	-0.34
weight	-0.82	0.89	0.94	0.89	1.00	-0.40	-0.26
acceleration	0.37	-0.47	-0.49	-0.67	-0.40	1.00	0.19
model_year	0.54	-0.28	-0.30	-0.34	-0.26	0.19	1.00

Figure 16: Correlation matrix in Task 1

1.5.2 Adding interactions term

In reality, the work done by the engine is the product of the force applied to the car and the distance the car moves. The force applied to the car is the product of the mass of the car and the acceleration ($F = m \times a$, where F is the force, m is the mass, and a is the acceleration). Therefore, we will **include the interaction between "weight" and "acceleration"** in the data.

1.5.3 Adding polynomial terms

Moreover, observing Figure 17, the fuel consumption from the 1970s to the 1980s has a quadratic increase. Therefore, we will **include the square of time from "model year" to 1970 in the data**.

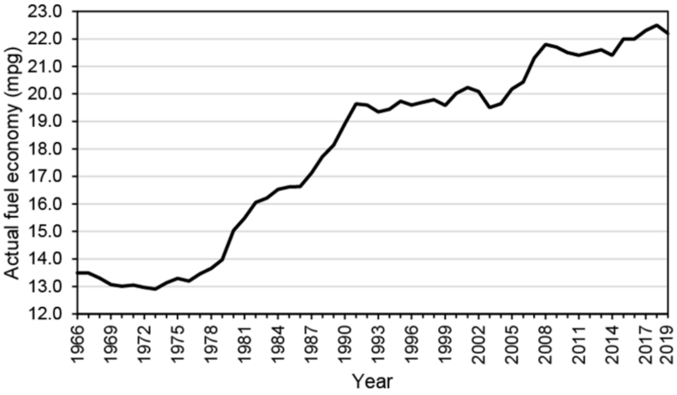
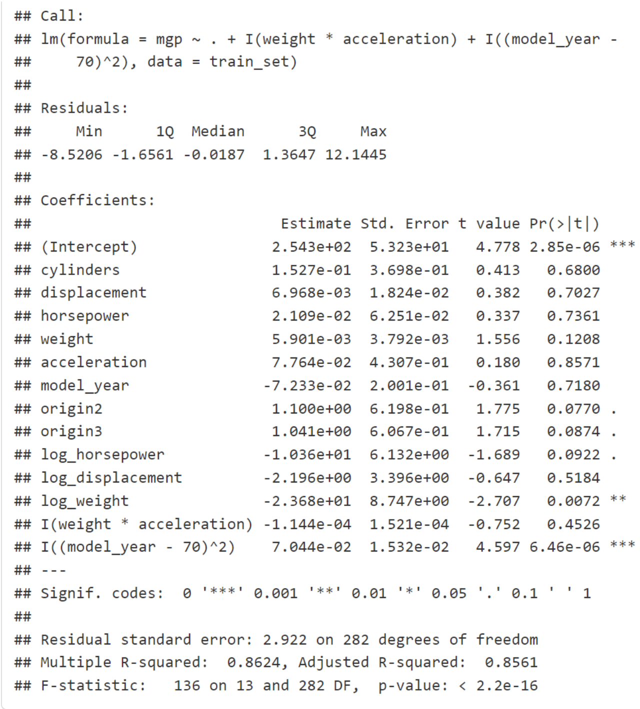


Figure 17: Changes in actual vehicle fuel economy from 1966 through 2019. Source: Actual fuel economy of cars and light trucks: 1966-2019, Michael Sivak, Sivak Applied Research

1.5.4 New model summary

The summary of the new model with new variables can be found in Figure 18. We added "log_horsepower", "log_displacement", "log_weight", "weight" × "acceleration" and ("model_year" $- 70)^2$ to the initial full linear model.

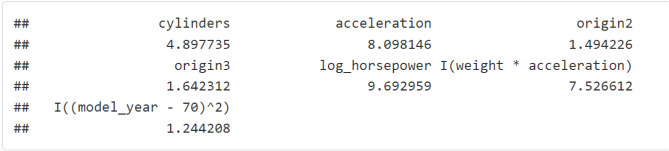


```
## Call:
## lm(formula = mgp ~ . + I(weight * acceleration) + I((model_year -
##     70)^2), data = train_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.5206 -1.6561 -0.0187  1.3647 12.1445
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              2.543e+02  5.323e+01   4.778 2.85e-06 ***
## cylinders                1.527e-01  3.698e-01   0.413   0.6800
## displacement             6.968e-03  1.824e-02   0.382   0.7027
## horsepower               2.109e-02  6.251e-02   0.337   0.7361
## weight                   5.901e-03  3.792e-03   1.556   0.1208
## acceleration             7.764e-02  4.307e-01   0.180   0.8571
## model_year              -7.233e-02  2.001e-01  -0.361   0.7180
## origin2                  1.100e+00  6.198e-01   1.775   0.0770 .
## origin3                  1.041e+00  6.067e-01   1.715   0.0874 .
## log_horsepower          -1.036e+01  6.132e+00  -1.689   0.0922 .
## log_displacement        -2.196e+00  3.396e+00  -0.647   0.5184
## log_weight              -2.368e+01  8.747e+00  -2.707   0.0072 **
## I(weight * acceleration) -1.144e-04  1.521e-04  -0.752   0.4526
## I((model_year - 70)^2)    7.044e-02  1.532e-02   4.597 6.46e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.922 on 282 degrees of freedom
## Multiple R-squared:  0.8624, Adjusted R-squared:  0.8561
## F-statistic:   136 on 13 and 282 DF,  p-value: < 2.2e-16
```

Figure 18: Summary of the new model in Task 1

1.5.5 Multicollinearity

To check for multicollinearity, we calculate the Variance Inflation Factor (VIF) of the variables. After introducing many new variables, the VIF values of many variables are very high. Therefore, we progressively remove the variable with the highest VIF value and repeat calculating the VIF values until all VIF values are less than 10. The remaining variables and their corresponding VIF values after this process can be viewed in Figure 19.

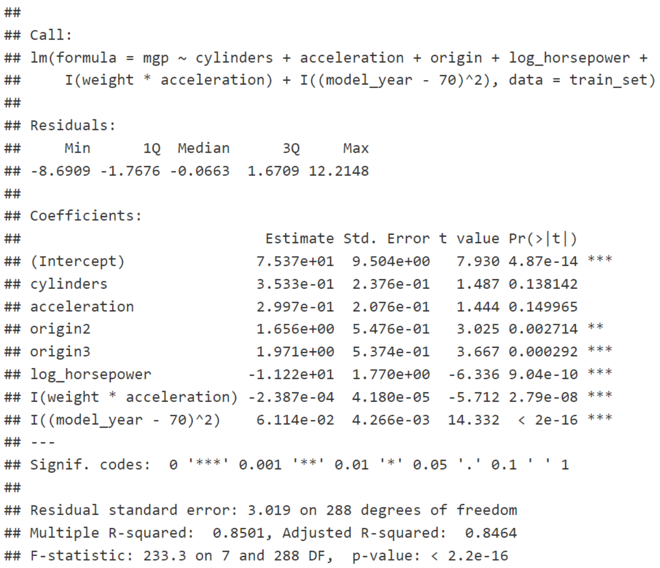


```
##                cylinders             acceleration                  origin2
##                 4.897735                 8.098146                 1.494226
##                  origin3           log_horsepower I(weight * acceleration)
##                 1.642312                 9.692959                 7.526612
##   I((model_year - 70)^2)
##                 1.244208
```

Figure 19: VIF values of the new model

1.5.6 Stepwise Algorithm

We now apply the Stepwise Algorithm to select the best model. The Stepwise Algorithm selects the same model as the one we manually selected, suggesting that no other variables should be excluded from the previous model. The summary of the model can be found in Figure 20.

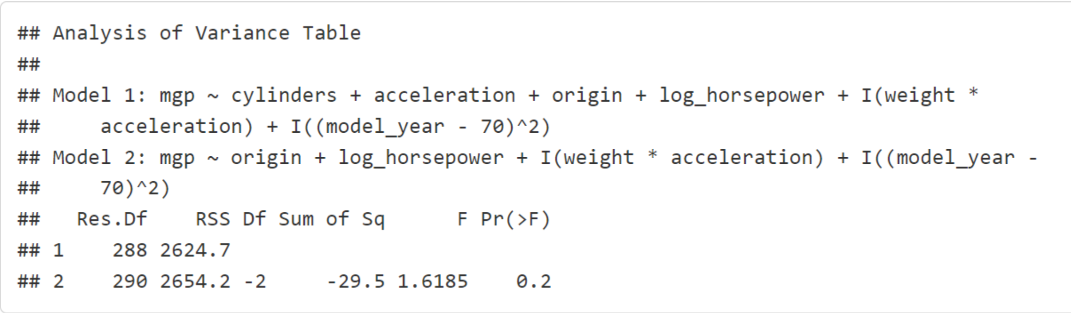


```
##
## Call:
## lm(formula = mgp ~ cylinders + acceleration + origin + log_horsepower +
##     I(weight * acceleration) + I((model_year - 70)^2), data = train_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.6909 -1.7676 -0.0663  1.6709 12.2148
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              7.537e+01  9.504e+00   7.930 4.87e-14 ***
## cylinders                3.533e-01  2.376e-01   1.487 0.138142
## acceleration             2.997e-01  2.076e-01   1.444 0.149965
## origin2                  1.656e+00  5.476e-01   3.025 0.002714 **
## origin3                  1.971e+00  5.374e-01   3.667 0.000292 ***
## log_horsepower          -1.122e+01  1.770e+00  -6.336 9.04e-10 ***
## I(weight * acceleration) -2.387e-04  4.180e-05  -5.712 2.79e-08 ***
## I((model_year - 70)^2)    6.114e-02  4.266e-03  14.332  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.019 on 288 degrees of freedom
## Multiple R-squared:  0.8501, Adjusted R-squared:  0.8464
## F-statistic: 233.3 on 7 and 288 DF,  p-value: < 2.2e-16
```

Figure 20: Summary of the new model after Stepwise Algorithm in Task 1

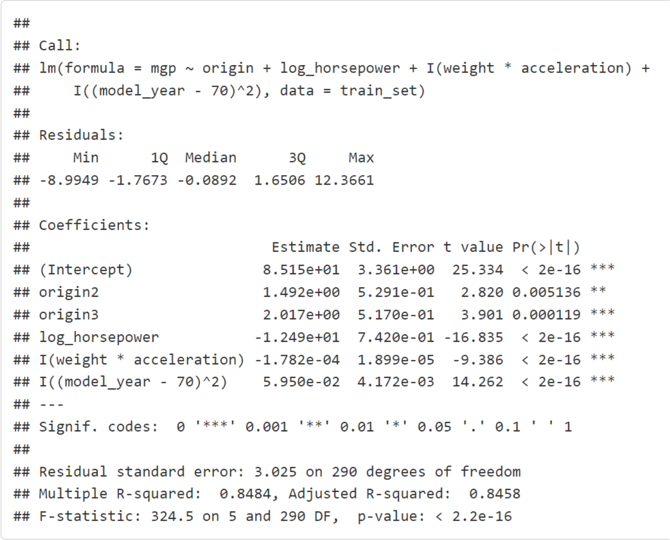
As the p-values of "cylinders" and "acceleration" are higher than the significance level of 0.05, we suspect that these variables are not significant. To validate this, we perform an ANOVA test as in Figure 21. The p-value of the ANOVA test is much higher than 0.05, indicating that the removed variables are not significant. Therefore, we remove "cylinders" and "cylinders" from the model. Our final model can be expressed by Equation (3) and summarized in Figure 22.

$$
\begin{aligned}
\text{mpg} = \beta_0 + \beta_1 \times \text{origin2} + \beta_2 \times \text{origin3} + \beta_3 \times \log(\text{horsepower}) \\
+ \beta_4 \times (\text{weight} \times \text{acceleration}) + \beta_5 \times (\text{model\_year} - 70)^2
\end{aligned} \tag{3}
$$



```
## Analysis of Variance Table
##
## Model 1: mgp ~ cylinders + acceleration + origin + log_horsepower + I(weight *
##     acceleration) + I((model_year - 70)^2)
## Model 2: mgp ~ origin + log_horsepower + I(weight * acceleration) + I((model_year -
##     70)^2)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    288 2624.7
## 2    290 2654.2 -2     -29.5 1.6185    0.2
```

Figure 21: ANOVA of "cylinders" and "acceleration" in the new model in Task 1



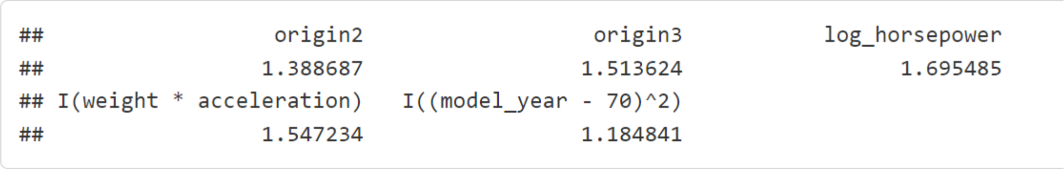
```
##
## Call:
## lm(formula = mgp ~ origin + log_horsepower + I(weight * acceleration) +
##     I((model_year - 70)^2), data = train_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.9949 -1.7673 -0.0892  1.6506 12.3661
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             8.515e+01  3.361e+00  25.334  < 2e-16 ***
## origin2                 1.492e+00  5.291e-01   2.820 0.005136 **
## origin3                 2.017e+00  5.170e-01   3.901 0.000119 ***
## log_horsepower         -1.249e+01  7.420e-01 -16.835  < 2e-16 ***
## I(weight * acceleration) -1.782e-04  1.899e-05  -9.386  < 2e-16 ***
## I((model_year - 70)^2)  5.950e-02  4.172e-03  14.262  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.025 on 290 degrees of freedom
## Multiple R-squared:  0.8484, Adjusted R-squared:  0.8458
## F-statistic: 324.5 on 5 and 290 DF,  p-value: < 2.2e-16
```

Figure 22: Summary of our final improved model

1.5.7 Final model diagnostic

- Multicollinearity

  To check for multicollinearity of our final model, we calculate the Variance Inflation Factor (VIF) of the variables (Figure 23). All VIF values are now much lower than 10, indicating that there is no multicollinearity in the model.



```
##                  origin2                  origin3           log_horsepower
##                 1.388687                 1.513624                 1.695485
## I(weight * acceleration)   I((model_year - 70)^2)
##                 1.547234                 1.184841
```

Figure 23: VIF values of our final model in Task 1

- Normality of residuals

  We now plot the Q-Q plot (Figure 24) of the residuals to check for normality. The Q-Q plot shows that most points lie close to the line in the center, but there are deviations at both ends (tails). This suggests that while the residuals are roughly normally distributed in the middle range, there are issues in the tails. This suggests that the normality assumption may not be fully satisfied.

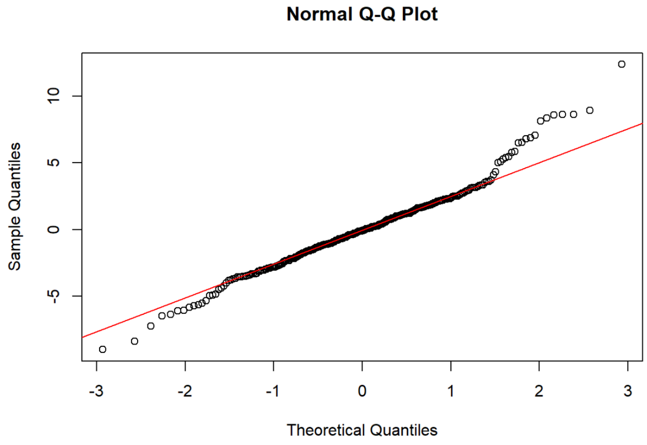
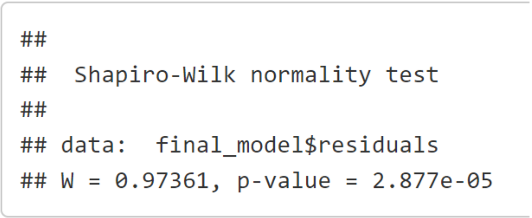


Figure 24: Q-Q plot of our final model in Task 1

We perform the Shapiro-Wilk test to check for normality of the residuals as in Figure 25. The p-value of the Shapiro-Wilk test is less than 0.05, indicating that the residuals are not normally distributed. This once again aligns with the Q-Q plot, where the residuals have heavier tails than a normal distribution.

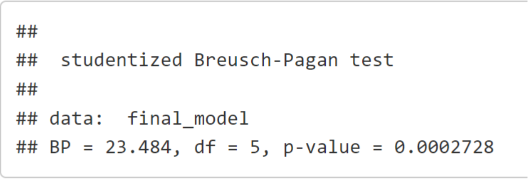


```
##
##  Shapiro-Wilk normality test
##
## data:  final_model$residuals
## W = 0.97361, p-value = 2.877e-05
```

Figure 25: Shapiro-Wilk test of our final model in Task 1

- Homoscedasticity

We also perform the Breusch-Pagan test to check for homoscedasticity as in Figure 26. The p-value of the Breusch-Pagan test is less than 0.05, indicating that the residuals are heteroscedastic.

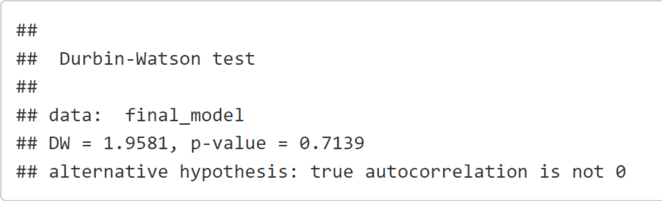


```
##
##  studentized Breusch-Pagan test
##
## data:  final_model
## BP = 23.484, df = 5, p-value = 0.0002728
```

Figure 26: Breusch-Pagan test of our final model in Task 1

- Autocorrelation

  We perform the Durbin-Watson test to check for autocorrelation as in Figure 27. The Durbin-Watson test gives a DW value close to 2, indicating that there is no autocorrelation in the residuals.



```
##
##  Durbin-Watson test
##
## data:  final_model
## DW = 1.9581, p-value = 0.7139
## alternative hypothesis: true autocorrelation is not 0
```

Figure 27: Durbin-Watson test for our final model in Task 1

- Model's semantic meaning

  As ”mpg” or miles per gallon is used for evaluating the fuel consumption, the smaller ”mpg” is, the higher the fuel consumption. Therefore, our predictions on the sign of the coefficient is

  1. $\beta_3 < 0$, as the more powerful the engine, the higher the fuel consumption.
  2. $\beta_4 < 0$, as the more force created by the vehicle, the higher the fuel consumption
  3. $\beta_5 > 0$, as according to the graph in Figure 17.

  The calculated coefficient is $\hat{\beta}_3 \approx -12.49, \hat{\beta}_4 \approx -1.782 \times 10^{-4}, \hat{\beta}_5 \approx 5.950 \times 10^{-2}$, which all align with our predictions.

- Conclusion on model diagnostics

  In conclusion, the model is not ideal as the residuals are not normally distributed and heteroscedastic. However, the model is still acceptable as the residuals are approximately normally distributed in the middle range and there is no autocorrelation in the residuals. Moreover, the R-squared value of our new model on the training set is higher than the baseline, suggesting that the new model is more effective in predicting ”mpg” compared to the baseline.

1.6 Model testing

The R-squared of both baseline and our final model on two training/testing set can be viewed in Table 1. Our proposed model outperforms the baseline model in both datasets. This suggests that our new variable added to the model is significant in improving the model fitting. The code of testing two models can be found in Section A.5.

	Training Set	Testing Set
Baseline model	0.8150	0.8268
Our model	**0.8484**	**0.8634**

Table 1: R-squared result of two models on training/testing set in Task 1

1.7 Conclusion

In this task, we first preprocessed the data by removing the ”car name” column, removing rows with missing values, converting the ”horsepower” column to numeric, and changing the ”origin” column to a factor. We then split the data into training and testing sets. We fitted a baseline linear regression model with all original variables to predict ”mpg”.

To improve our baseline model, we **apply the knowledge of the in-reality relationship** between ”mpg” and other variables to create a new model. We include the log transformation of ”horsepower”, ”displacement”, and ”weight”, the interaction between ”weight” and ”acceleration”, and the square of time from ”model year” to 1970 in the data. We then fit a new model with these variables. Our final model can be expressed in Equation 3. The new model has a higher R-squared value than the baseline model, suggesting that it is more effective in predicting ”mpg” compared to the baseline approach.

However, the new model is not ideal as the residuals are not normally distributed and heteroscedastic. Nevertheless, the new model is still acceptable as the residuals are approximately normally distributed in the middle range and there is no autocorrelation in the residuals. Finally, we test the new model on the test set and find that it has a higher R-squared value than the baseline model, indicating that the new model is more effective in predicting ”mpg”.

2 Task 2 - Dataset 1

Source: Fish Market

2.1 Summary of data

There are a total of 159 rows and 7 columns in the data. The columns are ”Weight”, ”Length1”, ”Length2”, ”Length3”, ”Width”, ”Height”, ”Species”. The description of each column is as follows:

- ”Species”: (multi-valued discrete) This column represents the species of the fish. It is a categorical variable that categorizes each fish into one of seven species. The species may include names like ”Perch,” ”Bream,” ”Roach,” ”Pike,” ”Smelt,” ”Parkki,” and ”Whitefish.” This column is the target variable for the polynomial regression analysis, where we aim to predict the fish’s weight based on its other attributes.

- ”Weight”: (continuous) This column represents the weight of the fish. It is a numerical variable that is typically measured in grams. The weight is the dependent variable we want to predict using polynomial regression.

- ”Length1”: (continuous) This column represents the first measurement of the fish’s length. It is a numerical variable, typically measured in centimetres.

- ”Length2”: (continuous) This column represents the second measurement of the fish’s length. It is another numerical variable, typically measured in centimetres.

- ”Length3”: (continuous) This column represents the third measurement of the fish’s length. Similar to the previous two columns, it is a numerical variable, usually measured in centimetres.

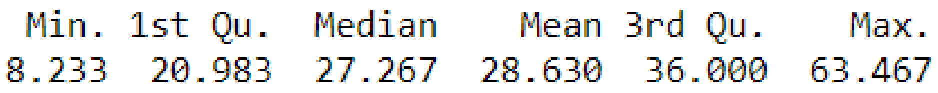
- ”Height”: (continuous) This column represents the height of the fish. It is a numerical variable, typically measured in centimetres.

- ”Width”: (continuous) This column represents the width of the fish. Like the other numerical variables, it is also typically measured in centimetres.

2.2 Data preprocessing

2.2.1 Combine length variables

Since Length1, Length2, and Length3 are the result of three different measure times, we will calculate the average of them to have the most accurate measurement.



```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 8.233  20.983  27.267  28.630  36.000  63.467 
```

Figure 28: Summary of new column Length

Now we remove old columns of Length

2.2.2 Remove duplicate rows

There are no duplicate rows in the data. Therefore, we do not need to remove any rows.

After data preprocessing, the data now has **159 rows and 5 columns**. Codes for data preprocessing can be found in Section B.1.

2.3 Histograms and plots

2.3.1 Distribution of Species

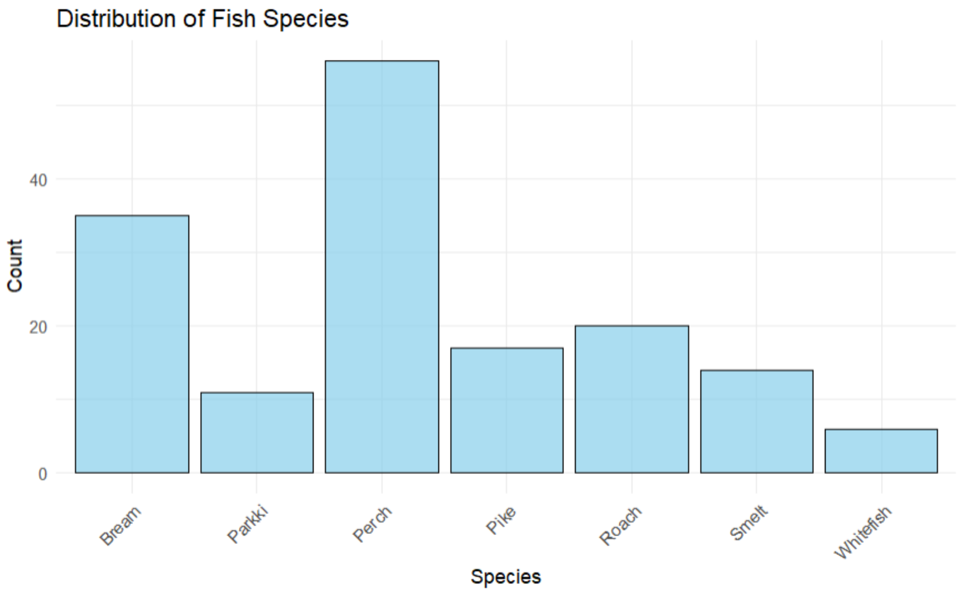


Figure 29: Distribution of Fish Species

We can see that Perch and Bream have more appearance in the dataset than other species.

2.3.2 Histogram of numeric column

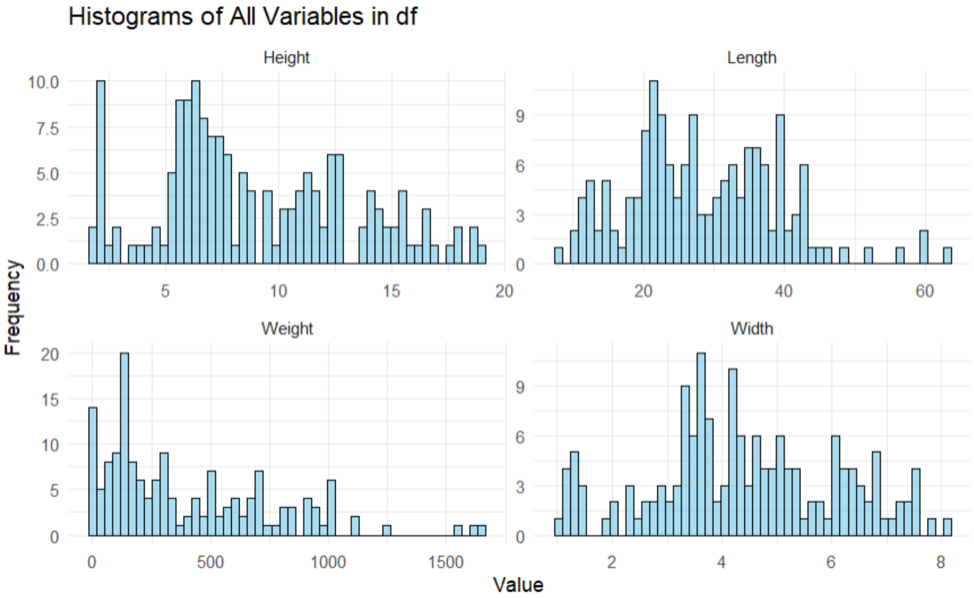


Figure 30: Histogram of All Variables in df

The histograms are not ”smooth”, suggesting the difference in measurements between different species. We now investigate each measurement on different species.

2.3.3 Height by Species

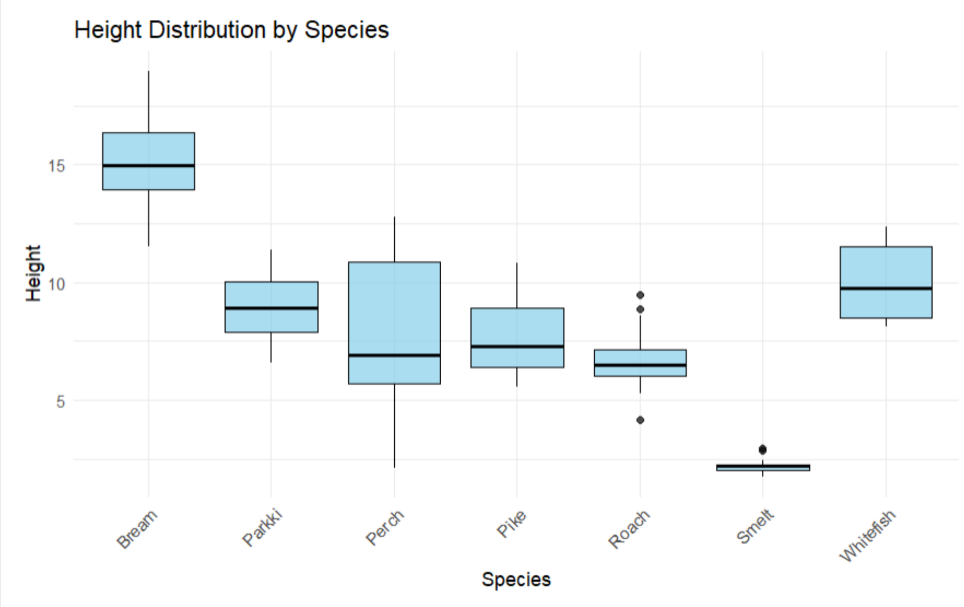


Figure 31: Height Distribution by Species

The boxplot suggests that Parkki, Perch, Pike, and Roach species has relatively small different in height. On the other hand, Smelt seems to have the smallest height, and Bream has largest height.

2.3.4 Width by Species

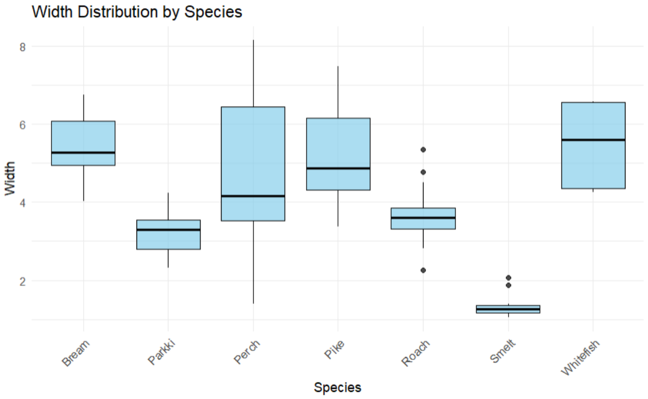


Figure 32: Width Distribution by Species

The boxplot suggests that Bream, Pike, and Whitefish species have the largest width. On the other hand, Smelt has the smallest width.

2.3.5 Length by Species

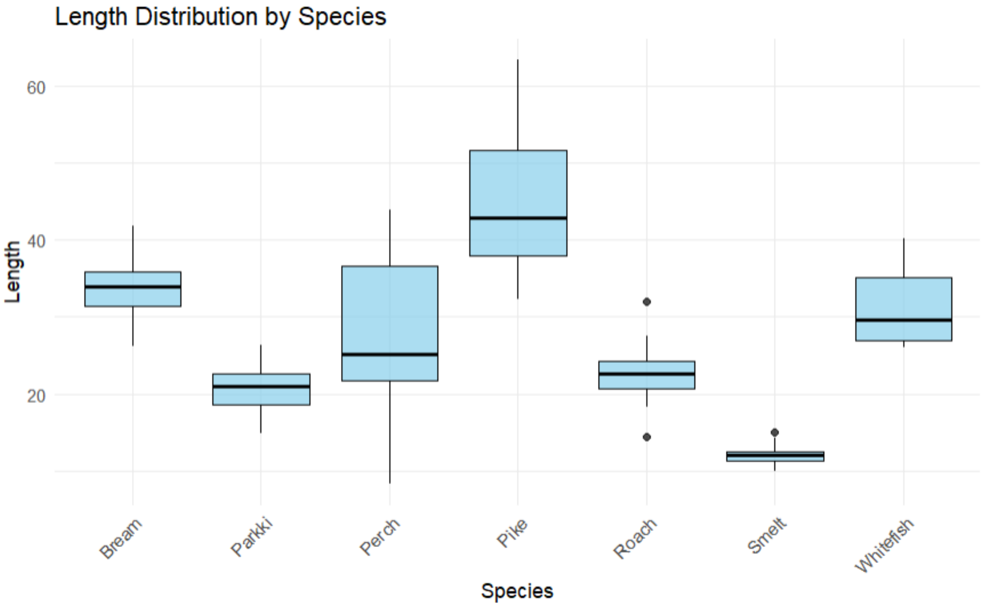


Figure 33: Length Distribution by Species

The boxplot suggests that Pike species has the largest length. On the other hand, Smelt has the smallest width.

2.3.6 Weight by Species

From 3 diagrams above, we can expect that Smelt is the smallest fish and hence has the smallest weight. And Pike, Bream might have the largest weight, since their size measurements are larger than the others.
The following boxplot confirms that.

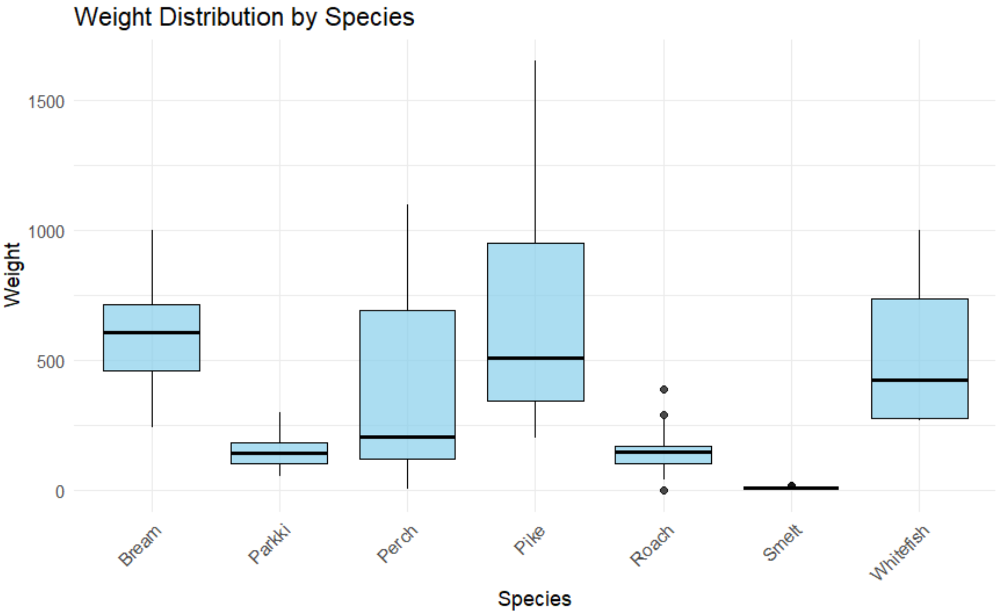


Figure 34: Weight Distribution by Species

2.3.7 Boxplots and outliers detection

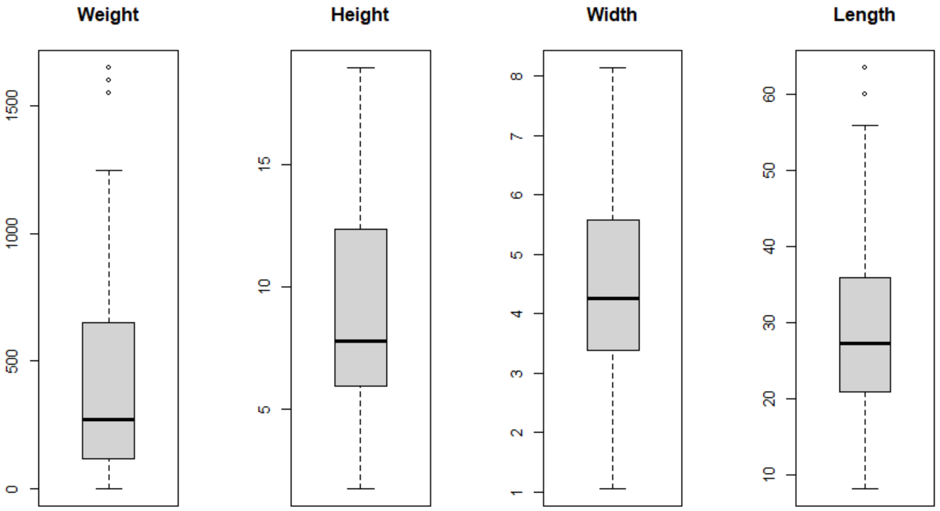


Figure 35: Outliers Detection of Fish Market

The box plots suggest that there are not many outliers in the variables' distributions. So we will not process any outliers removal. Code for histogram and plots can be found in Section B.2

2.4 Data splitting

2.4.1 Convert Species to factor

We convert Species to factor allowing the linear regression model to use each separate value of the column as an independent variable. Since each Species may have unique signatures that affects the Weight.

2.4.2 Split Data

We split the data into training and testing sets. The ratio of the training set to the testing set is 80:20.

The training set now has 127 rows while the testing set has 32, which is approximately 80% and 20% of the original data, respectively.

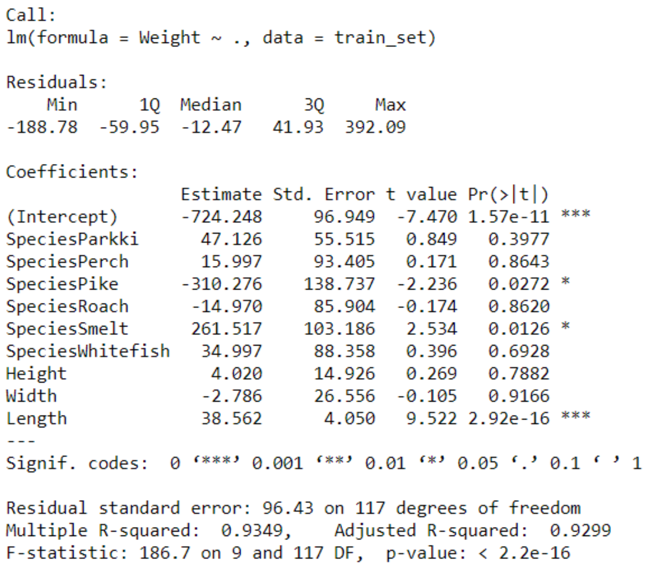
Code for data splitting can be found in Section B.3.

2.5 Baseline model

In this section, we try to create a baseline linear model. Codes for this section can be found in Section B.4.

We fit a linear regression model with all original variables to predict "Weight" as in Equation (1). Figure 36 shows the summary of this model.

$$\begin{aligned}\text{Weight} =&\beta_0 + \beta_1 \times \text{SpeciesParkki} + \beta_2 \times \text{SpeciesPerch} + \beta_3 \times \text{SpeciesPike} + \beta_4 \times \text{SpeciesRoach}\\ &+ \beta_5 \times \text{SpeciesSmelt} + \beta_6 \times \text{SpeciesWhitefish} + \beta_7 \times \text{Height} + \beta_8 \times \text{Width} + \beta_9 \times \text{Length}\end{aligned} \tag{4}$$



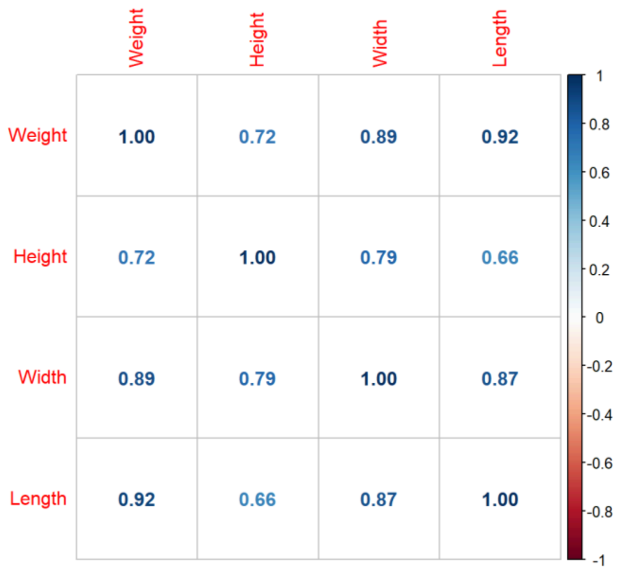
```
Call:
lm(formula = Weight ~ ., data = train_set)

Residuals:
    Min      1Q  Median      3Q     Max 
-188.78  -59.95  -12.47   41.93  392.09 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)      -724.248     96.949  -7.470 1.57e-11 ***
SpeciesParkki      47.126     55.515   0.849   0.3977    
SpeciesPerch       15.997     93.405   0.171   0.8643    
SpeciesPike      -310.276    138.737  -2.236   0.0272 *  
SpeciesRoach      -14.970     85.904  -0.174   0.8620    
SpeciesSmelt      261.517    103.186   2.534   0.0126 *  
SpeciesWhitefish   34.997     88.358   0.396   0.6928    
Height              4.020     14.926   0.269   0.7882    
Width              -2.786     26.556  -0.105   0.9166    
Length             38.562      4.050   9.522 2.92e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 96.43 on 117 degrees of freedom
Multiple R-squared:  0.9349,	Adjusted R-squared:  0.9299 
F-statistic: 186.7 on 9 and 117 DF,  p-value: < 2.2e-16
```

Figure 36: Summary of baseline model in Task 2 - Dataset 1

The T tests for significant of each variables show that there is some insignificant variables that we can remove.

2.5.1 Multicollinearity

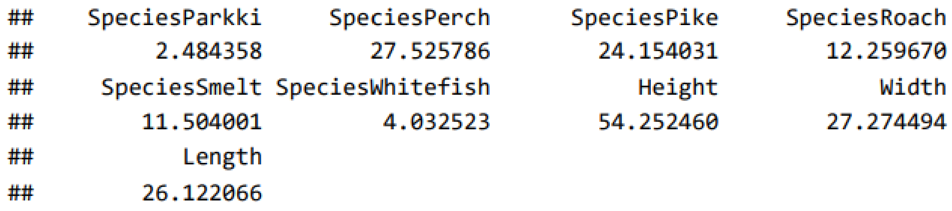
We now check for multicollinearity to consider removing highly correlated variables.



	Weight	Height	Width	Length
Weight	1.00	0.72	0.89	0.92
Height	0.72	1.00	0.79	0.66
Width	0.89	0.79	1.00	0.87
Length	0.92	0.66	0.87	1.00

Figure 37: Correlation Matrix

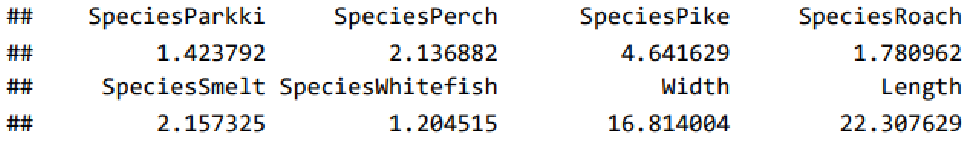
We calculate the Variance Inflation Factor (VIF) of the variables in Figure 38



```
##    SpeciesParkki     SpeciesPerch      SpeciesPike     SpeciesRoach 
##         2.484358        27.525786        24.154031        12.259670 
##     SpeciesSmelt SpeciesWhitefish           Height            Width 
##        11.504001         4.032523        54.252460        27.274494 
##           Length 
##        26.122066
```

Figure 38: VIF of baseline model

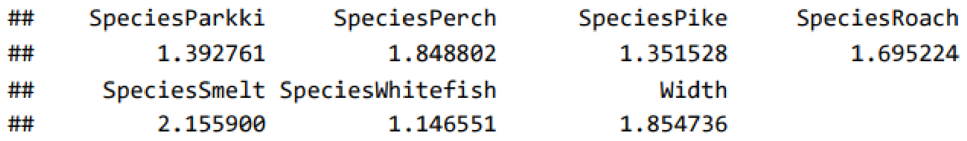
“Height” variable has the highest VIF value and greater than 10. Therefore, we will remove it from the model (see in Figure 39).



```
##    SpeciesParkki     SpeciesPerch      SpeciesPike     SpeciesRoach 
##         1.423792         2.136882         4.641629         1.780962 
##     SpeciesSmelt SpeciesWhitefish            Width           Length 
##         2.157325         1.204515        16.814004        22.307629
```

Figure 39: VIF of baseline model without ”Height”

“Length” variable has the highest VIF value and greater than 10. Therefore, we will remove it from the model (see in Figure 40).



```
##    SpeciesParkki     SpeciesPerch      SpeciesPike     SpeciesRoach
##         1.392761         1.848802         1.351528         1.695224
##     SpeciesSmelt SpeciesWhitefish            Width
##         2.155900         1.146551         1.854736
```

Figure 40: VIF of baseline model without "Height" and "Length"

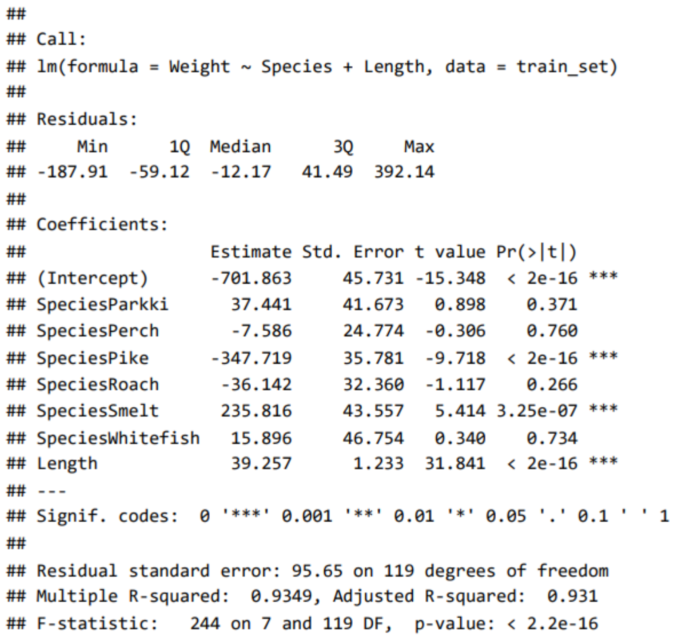
Now all variables have VIF values less than 10, indicating no strong multicollinearity in the model.

2.5.2 First Degree model

We apply the Stepwise Algorithm to select the best model. The result can be found in Figure 41. The model can be expressed as in Equation (6).

$$\text{Weight} = \beta_0 + \beta_1 \times \text{SpeciesParkki} + \beta_2 \times \text{SpeciesPerch} + \beta_3 \times \text{SpeciesPike} + \beta_4 \times \text{SpeciesRoach} \tag{5}$$

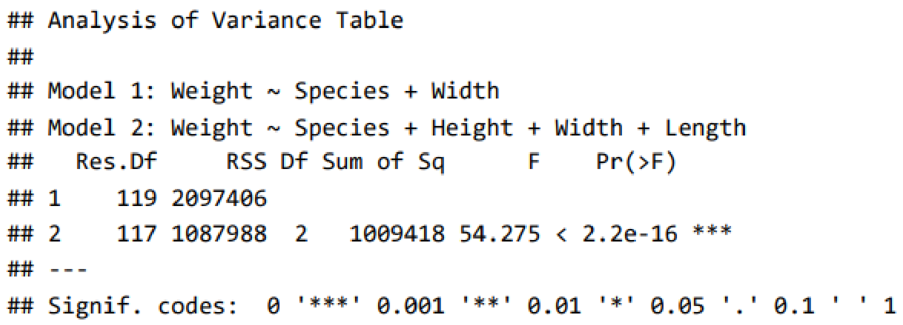
$$+ \beta_5 \times \text{SpeciesSmelt} + \beta_6 \times \text{SpeciesWhitefish} + \beta_7 \times \text{Length} \tag{6}$$



```
##
## Call:
## lm(formula = Weight ~ Species + Length, data = train_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -187.91  -59.12  -12.17   41.49  392.14
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -701.863     45.731 -15.348  < 2e-16 ***
## SpeciesParkki      37.441     41.673   0.898    0.371
## SpeciesPerch       -7.586     24.774  -0.306    0.760
## SpeciesPike      -347.719     35.781  -9.718  < 2e-16 ***
## SpeciesRoach      -36.142     32.360  -1.117    0.266
## SpeciesSmelt      235.816     43.557   5.414 3.25e-07 ***
## SpeciesWhitefish   15.896     46.754   0.340    0.734
## Length             39.257      1.233  31.841  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 95.65 on 119 degrees of freedom
## Multiple R-squared:  0.9349, Adjusted R-squared:  0.931
## F-statistic:   244 on 7 and 119 DF,  p-value: < 2.2e-16
```

Figure 41: Summary of the baseline model after Stepwise in Task 2 - Dataset 1

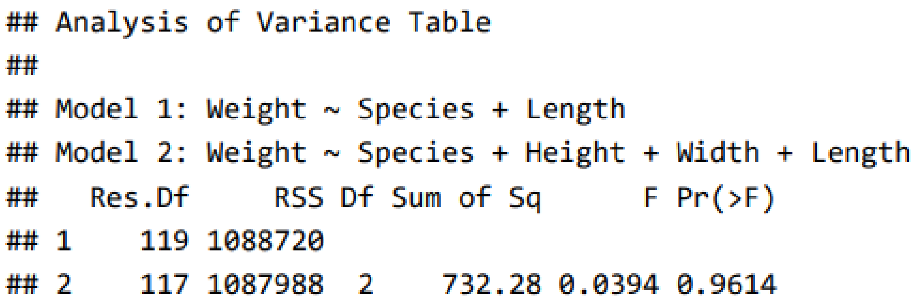
The T test of SpeciesPike, SpeciesSmelt align with our expectation since these two species have the most significant different in “Weight” among all species.



```
## Analysis of Variance Table
##
## Model 1: Weight ~ Species + Width
## Model 2: Weight ~ Species + Height + Width + Length
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1    119 2097406
## 2    117 1087988  2   1009418 54.275 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 42: ANOVA between the reduced model by checking VIF and the baseline

The p-value of the ANOVA test is much less than the significance level of 0.05, indicating that the removed variables are significant, so we should not use model1_vif .



```
## Analysis of Variance Table
##
## Model 1: Weight ~ Species + Length
## Model 2: Weight ~ Species + Height + Width + Length
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1    119 1088720
## 2    117 1087988  2    732.28 0.0394 0.9614
```

Figure 43: ANOVA between the reduced model by using stepwise search and the baseline

The p-value of the ANOVA test is much higher than the significance level of 0.05, indicating that the removedvariables are not significant, and we can use model1_step

2.5.3 Model diagnostics

- Normality of residuals

  We now plot the Q-Q plot (Figure 44) of the residuals to check for normality. The Q-Q plot shows that most points lie close to the line in the center, but there are deviations at both ends (tails). This suggests that while the residuals are roughly normally distributed in the middle range, there are issues in the tails. This suggests that the normality assumption may not be fully satisfied.

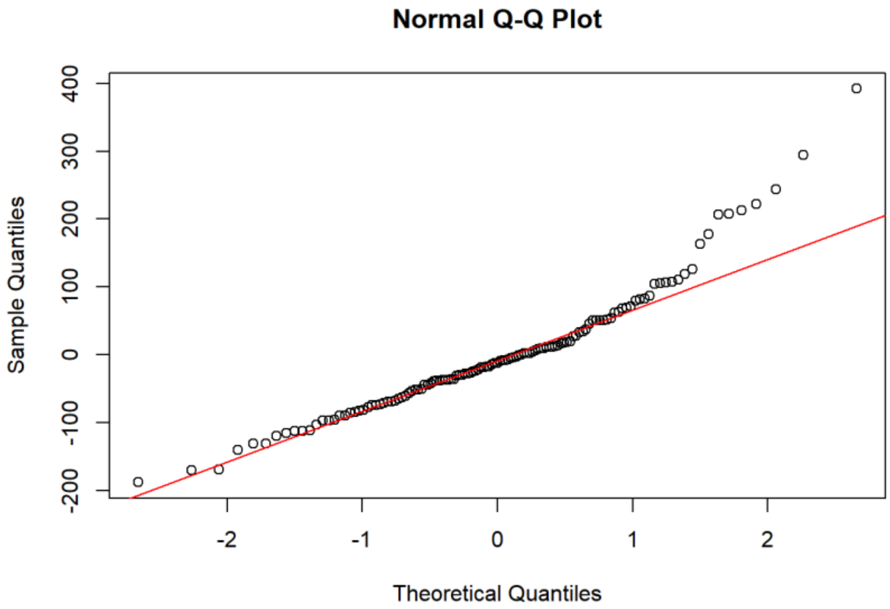
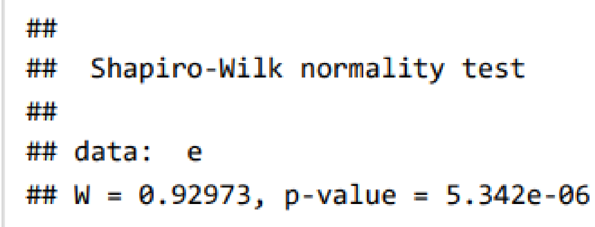


Figure 44: Q-Q plot of the residuals in the final baseline model in Task 2 - Dataset 1

To test the normality of the residuals, we perform the Shapiro-Wilk test in Figure 45. The p-value of the Shapiro-Wilk test is less than 0.05, indicating that the residuals are not normally distributed. This once again aligns with the Q-Q plot, where the residuals have heavier tails than a normal distribution.

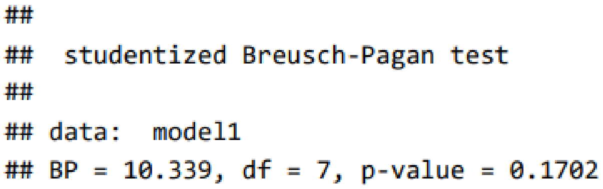


```
##
##  Shapiro-Wilk normality test
##
## data:  e
## W = 0.92973, p-value = 5.342e-06
```

Figure 45: Shaprio-Wilk test of final baseline model in Task 2 - Dataset 1

- Homoscedasticity

We also perform the Breusch-Pagan test to check for homoscedasticity as in Figure 46. The p-value of the Breusch-Pagan test is greater than 0.05, so we fail to reject the null hypothesis of homoscedasticity. We can conclude that homoscedasticity is present.

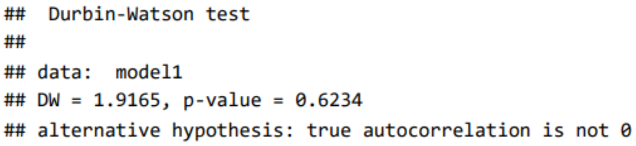


```
##
##  studentized Breusch-Pagan test
##
## data:  model1
## BP = 10.339, df = 7, p-value = 0.1702
```

Figure 46: Breusch-Pagan test of final baseline model in Task 2 - Dataset 1

- Autocorrelation

  We perform the Durbin-Watson test to check for autocorrelation as in Figure 47. The Durbin-Watson test gives a DW value close to 2, indicating that there is no autocorrelation.



```
##  Durbin-Watson test
##
## data:  model1
## DW = 1.9165, p-value = 0.6234
## alternative hypothesis: true autocorrelation is not 0
```

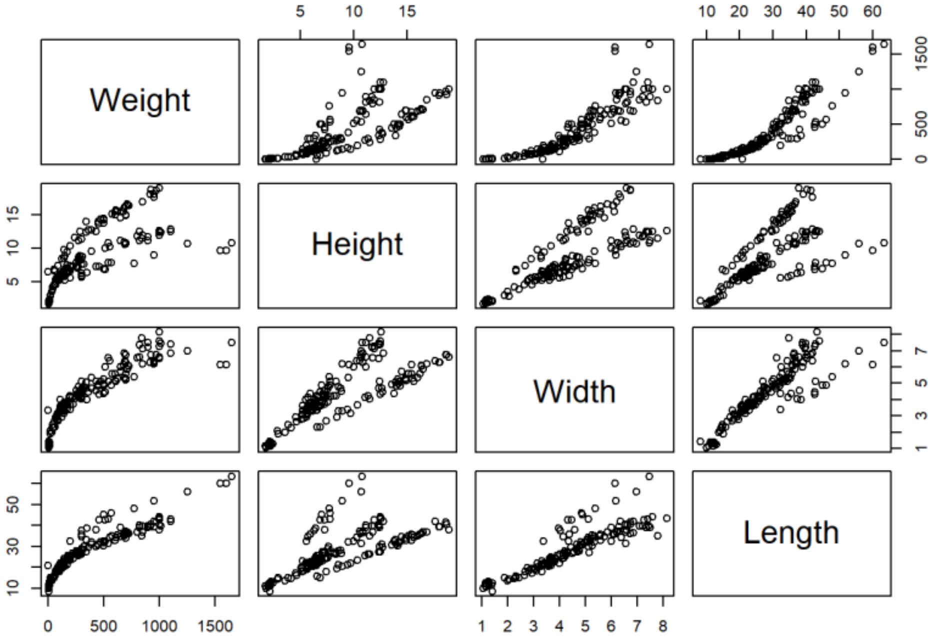
Figure 47: Durbin-Watson test of final baseline model in Task 2 - Dataset 1

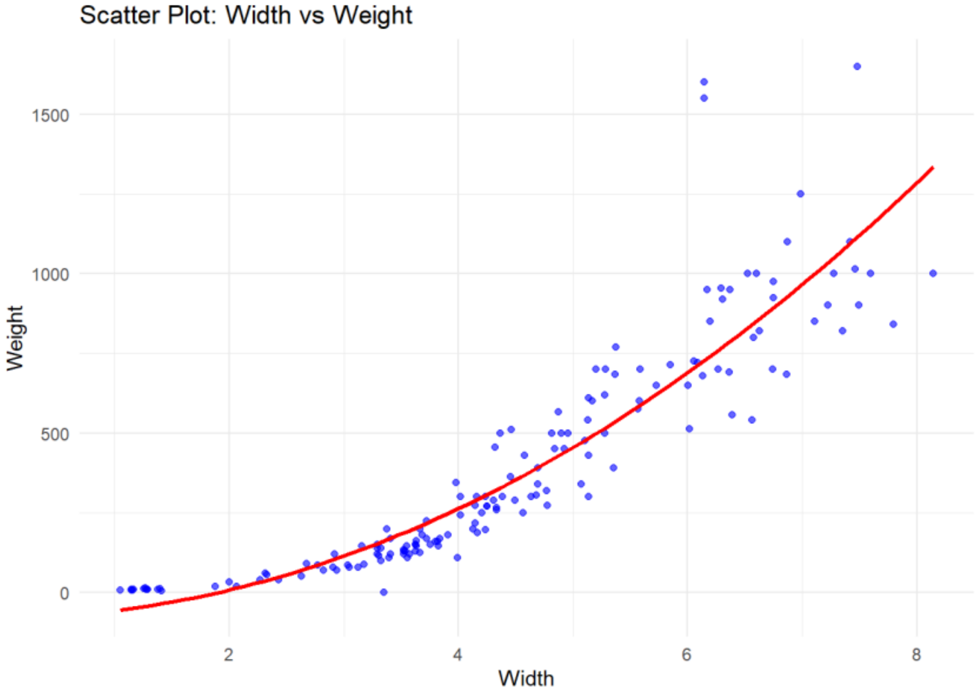
- Conclusion on model diagnostics

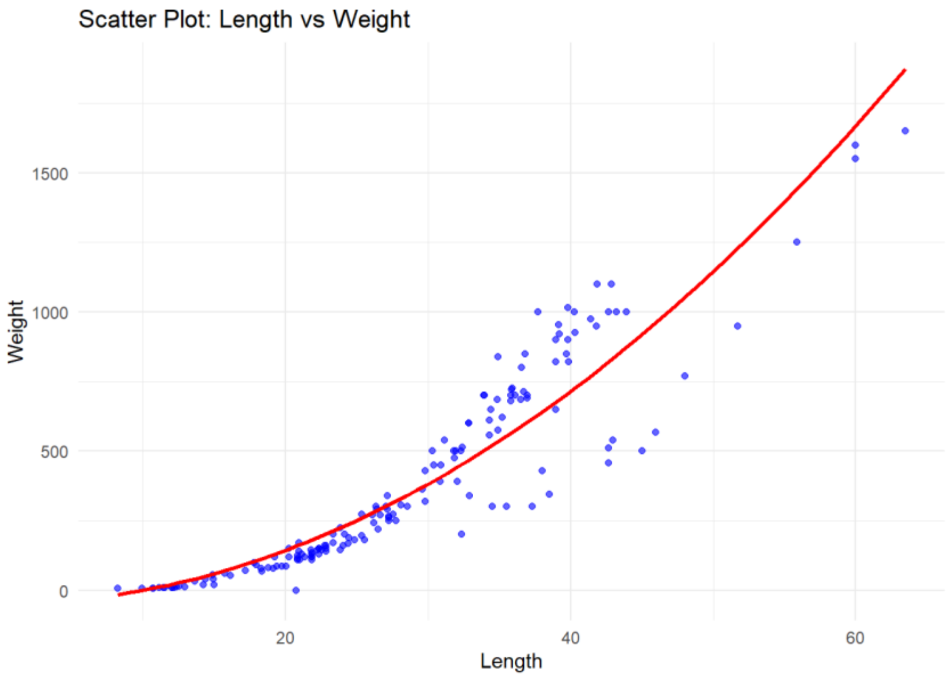
  In conclusion, the model is not ideal as the residuals are not normally distributed. However, the model is still acceptable as the residuals are homoscedastic, approximately normally distributed in the middle range and there is no autocorrelation in the residuals. Moreover, the R-squared value of our new model on the training set is higher than the baseline, suggesting that the new model is more effective in predicting ”Weight” compared to the baseline.

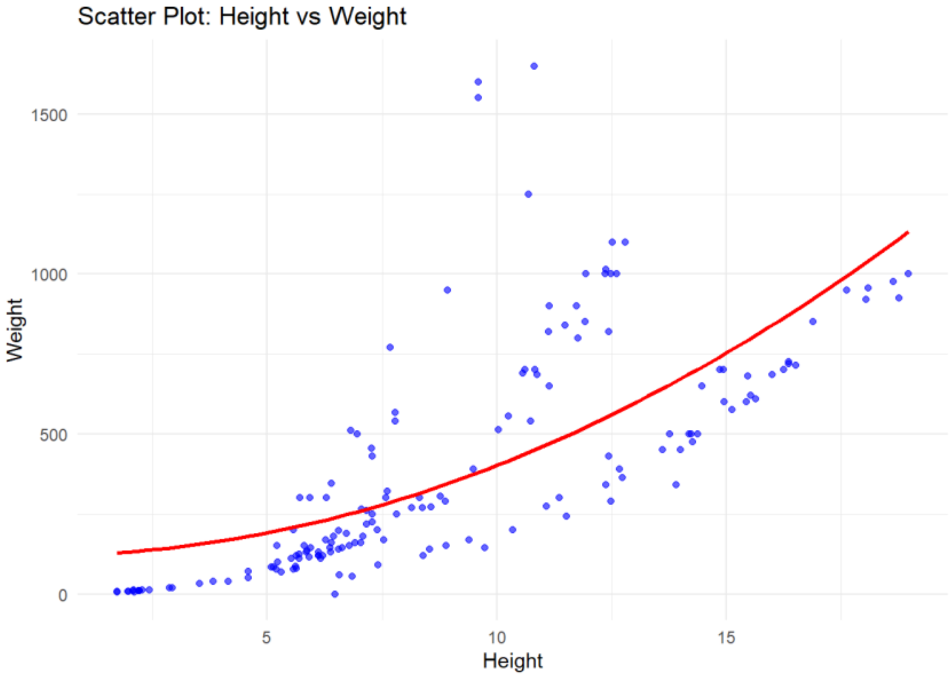
2.6 Quadratic model

2.6.1 Plot



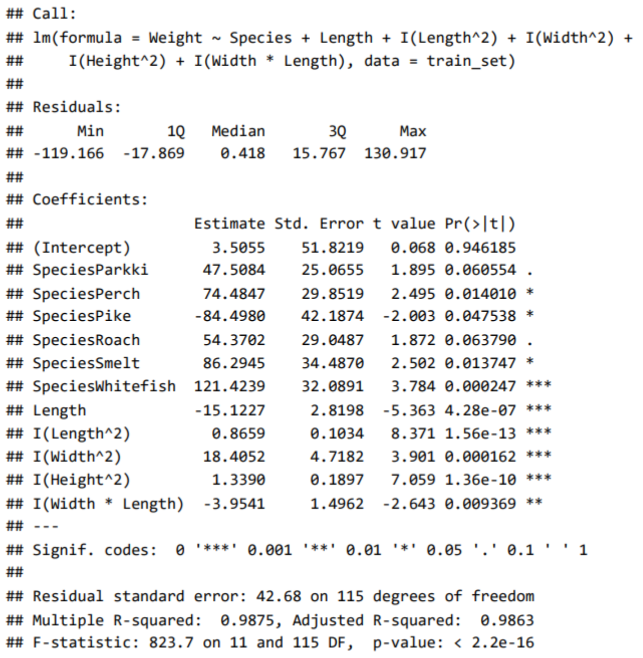




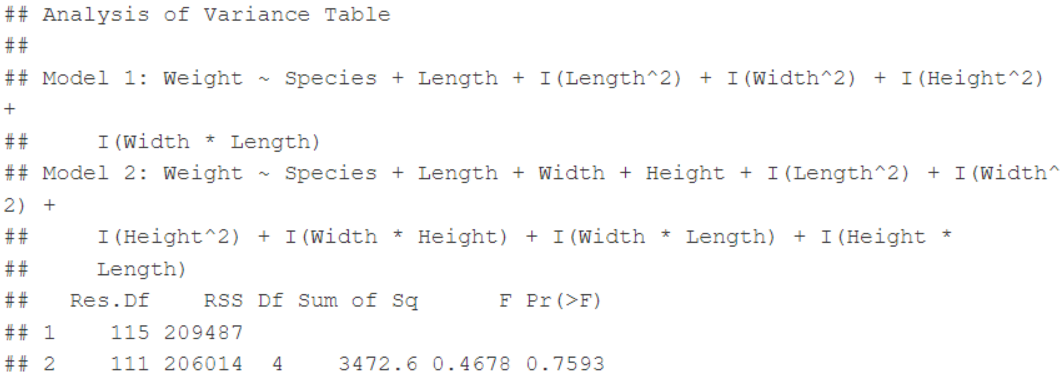


2.6.2 Selecting variable

We consider the quadratic regression model with all variables then run stepwise search and the result



```
## Call:
## lm(formula = Weight ~ Species + Length + I(Length^2) + I(Width^2) +
##     I(Height^2) + I(Width * Length), data = train_set)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -119.166  -17.869    0.418   15.767  130.917
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         3.5055    51.8219   0.068 0.946185
## SpeciesParkki      47.5084    25.0655   1.895 0.060554 .
## SpeciesPerch       74.4847    29.8519   2.495 0.014010 *
## SpeciesPike       -84.4980    42.1874  -2.003 0.047538 *
## SpeciesRoach       54.3702    29.0487   1.872 0.063790 .
## SpeciesSmelt       86.2945    34.4870   2.502 0.013747 *
## SpeciesWhitefish  121.4239    32.0891   3.784 0.000247 ***
## Length            -15.1227     2.8198  -5.363 4.28e-07 ***
## I(Length^2)         0.8659     0.1034   8.371 1.56e-13 ***
## I(Width^2)         18.4052     4.7182   3.901 0.000162 ***
## I(Height^2)         1.3390     0.1897   7.059 1.36e-10 ***
## I(Width * Length)  -3.9541     1.4962  -2.643 0.009369 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 42.68 on 115 degrees of freedom
## Multiple R-squared:  0.9875, Adjusted R-squared:  0.9863
## F-statistic: 823.7 on 11 and 115 DF,  p-value: < 2.2e-16
```



```
## Analysis of Variance Table
##
## Model 1: Weight ~ Species + Length + I(Length^2) + I(Width^2) + I(Height^2)
+
##     I(Width * Length)
## Model 2: Weight ~ Species + Length + Width + Height + I(Length^2) + I(Width^
2) +
##     I(Height^2) + I(Width * Height) + I(Width * Length) + I(Height *
##     Length)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    115 209487
## 2    111 206014  4    3472.6 0.4678 0.7593
```

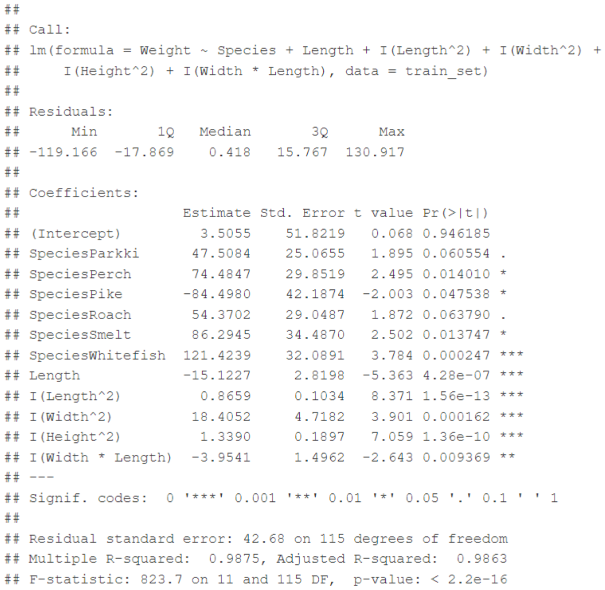
Figure 48: ANOVA test between the quadratic model before and after stepwise search in Task 2 - Dataset 1

The p-value of the ANOVA test is much higher than the significance level of 0.05, indicating that the removed variables are not significant, and we can use the reduced model.

$$\text{Weight} = \beta_0 + \beta_1 \times \text{SpeciesParkki} + \beta_2 \times \text{SpeciesPerch} + \beta_3 \times \text{SpeciesPike} + \beta_4 \times \text{SpeciesRoach} \tag{7}$$

$$+ \beta_5 \times \text{SpeciesSmelt} + \beta_6 \times \text{SpeciesWhitefish} + \beta_7 \times \text{Length} + \beta_8 \times \text{Length}^2 + \beta_9 \times \text{Width}^2 \tag{8}$$

$$+ \beta_{10} \times \text{Height}^2 + \beta_{11} \times (\text{Width} \times \text{Length}) \tag{9}$$



```
##
## Call:
## lm(formula = Weight ~ Species + Length + I(Length^2) + I(Width^2) +
##     I(Height^2) + I(Width * Length), data = train_set)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -119.166  -17.869    0.418   15.767  130.917
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         3.5055    51.8219   0.068 0.946185
## SpeciesParkki      47.5084    25.0655   1.895 0.060554 .
## SpeciesPerch       74.4847    29.8519   2.495 0.014010 *
## SpeciesPike       -84.4980    42.1874  -2.003 0.047538 *
## SpeciesRoach       54.3702    29.0487   1.872 0.063790 .
## SpeciesSmelt       86.2945    34.4870   2.502 0.013747 *
## SpeciesWhitefish  121.4239    32.0891   3.784 0.000247 ***
## Length            -15.1227     2.8198  -5.363 4.28e-07 ***
## I(Length^2)         0.8659     0.1034   8.371 1.56e-13 ***
## I(Width^2)         18.4052     4.7182   3.901 0.000162 ***
## I(Height^2)         1.3390     0.1897   7.059 1.36e-10 ***
## I(Width * Length)  -3.9541     1.4962  -2.643 0.009369 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 42.68 on 115 degrees of freedom
## Multiple R-squared:  0.9875, Adjusted R-squared:  0.9863
## F-statistic: 823.7 on 11 and 115 DF,  p-value: < 2.2e-16
```

Figure 49: Summary of our final improved model

2.6.3 Final model diagnostic

- Normality of residuals

  We now plot the Q-Q plot (Figure 50) of the residuals to check for normality. The Q-Q plot shows that most points lie close to the line in the center, but there are deviations at both ends (tails). This suggests that while the residuals are roughly normally distributed in the middle range, there are issues in the tails. This suggests that the normality assumption may not be fully satisfied.

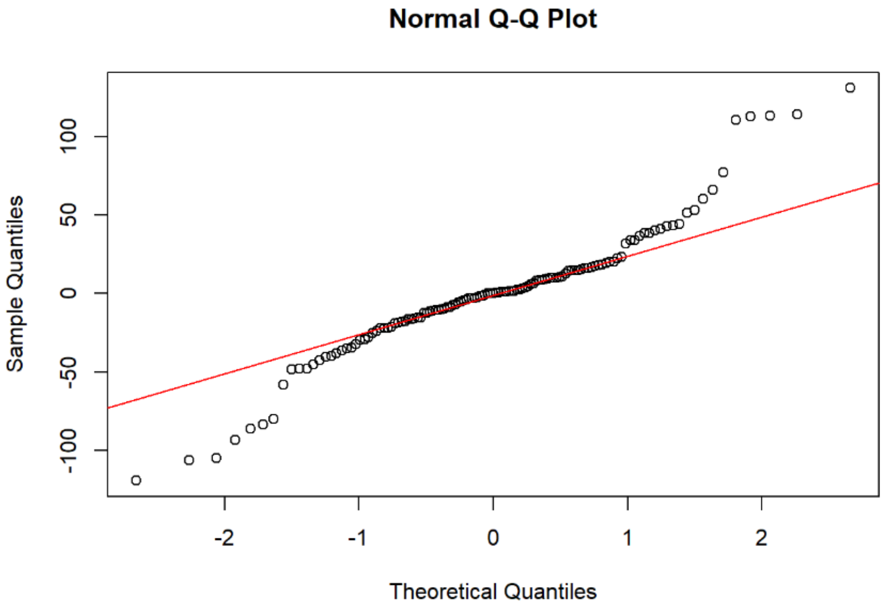
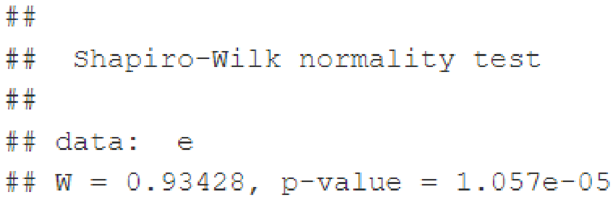


Figure 50: Q-Q plot of our final model in Task 2 - Dataset 1

We perform the Shapiro-Wilk test to check for normality of the residuals as in Figure 51. The p-value of the Shapiro-Wilk test is less than 0.05, indicating that the residuals are not normally distributed. This once again aligns with the Q-Q plot, where the residuals have heavier tails than a normal distribution.

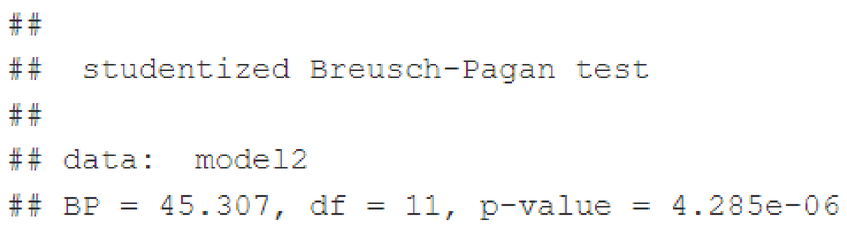


```
##
##  Shapiro-Wilk normality test
##
## data:  e
## W = 0.93428, p-value = 1.057e-05
```

Figure 51: Shapiro-Wilk test of our final model Task 2 - Dataset 1

- Homoscedasticity

  We also perform the Breusch-Pagan test to check for homoscedasticity as in Figure 52. The p-value of the Breusch-Pagan test is less than 0.05, indicating that the residuals are heteroscedastic.

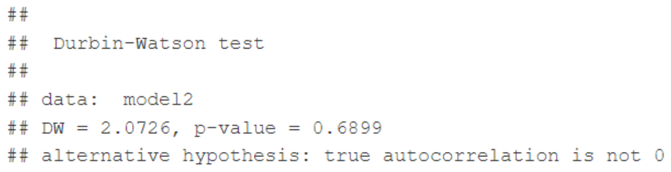


```
##
##  studentized Breusch-Pagan test
##
## data:  model2
## BP = 45.307, df = 11, p-value = 4.285e-06
```

Figure 52: Breusch-Pagan test of our final model in Task 2 - Dataset 1

- Autocorrelation

  We perform the Durbin-Watson test to check for autocorrelation as in Figure 53. The Durbin-Watson test gives a DW value close to 2, indicating that there is no autocorrelation in the residuals.



```
##
##  Durbin-Watson test
##
## data:  model2
## DW = 2.0726, p-value = 0.6899
## alternative hypothesis: true autocorrelation is not 0
```

Figure 53: Durbin-Watson test for our final model in Task 2 - Dataset 1

- Conclusion on model diagnostics

  In conclusion, the model is not ideal as the residuals are not normally distributed and heteroscedastic. However, the model is still acceptable as the residuals are approximately normally distributed in the middle range and there is no autocorrelation in the residuals. Moreover, the R-squared value of our new model on the training set is higher than the baseline, suggesting that the new model is more effective in predicting ”Weight” compared to the baseline.

2.7 Model testing

The R-squared of both baseline and our final model on two training/testing set can be viewed in Table 2. Our proposed model outperforms the baseline model in both datasets. This suggests that our new variable added to the model is significant in improving the model fitting. The code of testing two models can be found in Section B.7.

	Training Set	Testing Set
Baseline model	0.9349	0.9160
Our model	**0.9875**	**0.9639**

Table 2: R-squared result of two models on training/testing set in Task 2 - Dataset 1

2.8 Conclusion

In this task, we first preprocessed the data by combining the “Length” column, and changing the “Species” column to a factor. We then split the data into training and testing sets. We fitted a baseline linear regression model with all original variables to predict “Weight”.

To improve our baseline model, we **apply the observation achieved from scatter plots** between the predictors and the target to create a new model. We include the full quadratic model of (Length, Height, Width) and add the dummy variables of “Species”. Then we run the model through a backward stepwise search to keep significant variables. Our final model can be expressed in Equation 9.

However, both of the models is not ideal as the residuals are not normally distributed and heteroscedastic. Nevertheless, the second model is still acceptable as the residuals are approximately normally distributed in the middle range and there is no autocorrelation in the residuals. Finally, we test the new quadratic model on the test set and find that it has a higher R-squared value than the linear model, indicating that the new model is more effective in predicting “Weight”, the ANOVA test confirms that.

3 Task 2 - Dataset 2

Source: Body Fat Prediction Dataset

3.1 Summary of data

There are a total of 252 rows and 15 columns (15 quantitative variables) in the data.

The columns are ”Density”, ”BodyFat”, ”Age”, ”Weight”, ”Height”, ”Neck”, ”Chest”, ”Abdomen”, ”Hip”, ”Thigh”, ”Knee”, ”Ankle”, ”Biceps”, ”Forearm”, and ”Wrist”.

Since there is no qualitative variable in the dataset, we create a new column named ”BMI” following these steps:

- Calculate BMI score by using the formula: BMI score $= \dfrac{\text{Weight}}{\text{Height}^2} \times 703$
- Map BMI score into 6 categories: "Underweight" (from -infinity to 18.4), "Normal" (from higher than 18.4 to 24.9), "Overweight" (from higher 24.9 to 29.9), "Moderately Obese" (from higher 29.9 to 34.9), "Severely Obese" (from higher 34.9 to 39.9), and "Morbidly Obese" (from higher 39.9 to infinity)

Codes for creating the "BMI" column can be found in Section C.1.

The description of each column is as follows:

- "Density": (continuous) Density determined from underwater weighing
- "BodyFat" (Target): (continuous) Percent of body fat from Siri's (1956) equation: $Fat\% = \dfrac{495}{\text{Density}} - 450$.
- "Age": (continuous) Age (years)
- "Weight": (continuous) Weight (lbs)
- "Height": (continuous) Height (inches)
- "Neck": (continuous) Neck circumference (cm)
- "Chest": (continuous) Chest circumference (cm)
- "Abdomen": (continuous) Abdomen circumference (cm)
- "Hip": (continuous) Hip circumference (cm)
- "Thigh": (continuous) Thigh circumference (cm)
- "Knee": (continuous) Knee circumference (cm)
- "Ankle": (continuous) Ankle circumference (cm)
- "Biceps": (continuous) Biceps (extended) circumference (cm)
- "Forearm": (continuous) Forearm circumference (cm)
- "Wrist": (continuous) Wrist circumference (cm)

- ”BMI”: (category) 6 BMI categories (Underweight, Normal, Overweight, Moderately Obese, Severely Obese, and Morbidly Obese)

From the meaning of the variables, we have some predictions on the relationship between the Percent of body fat and other variables.

- ”Density”: can be calculated by reversing the Siri’s (1956) equation.
- ”Age”: the older you are, the higher the body fat percentage.
- ”Weight”: the heavier you are, the higher the body fat percentage.
- ”Height”: the taller you are, the higher the body fat percentage.
- ”Neck”: the larger the neck circumference, the higher the body fat percentage.
- ”Chest”: the larger the chest circumference, the higher the body fat percentage.
- ”Abdomen”: the larger the abdomen circumference, the higher the body fat percentage.
- ”Hip”: the larger the hip circumference, the higher the body fat percentage.
- ”Thigh”: the larger the thigh circumference, the higher the body fat percentage.
- ”Knee”: the larger the knee circumference, the higher the body fat percentage.
- ”Ankle”: the larger the ankle circumference, the higher the body fat percentage.
- ”Biceps”: the larger the biceps circumference, the higher the body fat percentage.
- ”Forearm”: the larger the forearm circumference, the higher the body fat percentage.
- ”Wrist”: the larger the wrist circumference, the higher the body fat percentage.
- ”BMI”: The higher category level, the higher the body fat percentage.

3.2 Data preprocessing

3.2.1 Removing ”Density”

Since ”Bodyfat” can be calculated from ”Density” by using Siri’s equation and Siri’s equation is a linear function then the correlation of ”Bodyfat” and ”Density” will be approximately 1. Therefore, we need to remove ”Density”.

3.2.2 Removing missing data

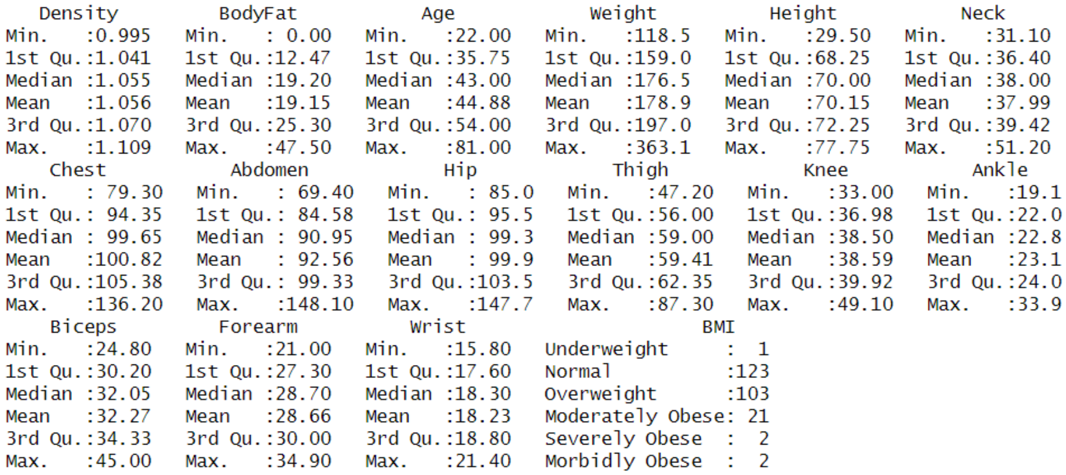
There are no missing values in the data. Therefore, we do not need to handle anything.

3.2.3 Remove duplicate rows

There are no duplicate rows in the data. Therefore, we do not need to remove any rows.

3.2.4 Create dummy variables

We first summarize the data in Figure 54 to see the statistics of the ”BMI” column. Since the ”Underweight”, ”Severely Obese”, and ”Morbidly Obese” categories are very few, we can merge ”Underweight” and ”Normal” into ”Underweight_and_Normal”, and merge ”Moderately Obese”, ”Severely Obese”, and ”Morbidly Obese” into ”Obese”. Then, we create 2 dummy variables ”BMI_Overweight” and ”BMI_Obese”.



```
    Density          BodyFat           Age            Weight          Height           Neck      
 Min.   :0.995   Min.   : 0.00   Min.   :22.00   Min.   :118.5   Min.   :29.50   Min.   :31.10  
 1st Qu.:1.041   1st Qu.:12.47   1st Qu.:35.75   1st Qu.:159.0   1st Qu.:68.25   1st Qu.:36.40  
 Median :1.055   Median :19.20   Median :43.00   Median :176.5   Median :70.00   Median :38.00  
 Mean   :1.056   Mean   :19.15   Mean   :44.88   Mean   :178.9   Mean   :70.15   Mean   :37.99  
 3rd Qu.:1.070   3rd Qu.:25.30   3rd Qu.:54.00   3rd Qu.:197.0   3rd Qu.:72.25   3rd Qu.:39.42  
 Max.   :1.109   Max.   :47.50   Max.   :81.00   Max.   :363.1   Max.   :77.75   Max.   :51.20  
     Chest           Abdomen            Hip            Thigh            Knee           Ankle     
 Min.   : 79.30   Min.   : 69.40   Min.   : 85.0   Min.   :47.20   Min.   :33.00   Min.   :19.1  
 1st Qu.: 94.35   1st Qu.: 84.58   1st Qu.: 95.5   1st Qu.:56.00   1st Qu.:36.98   1st Qu.:22.0  
 Median : 99.65   Median : 90.95   Median : 99.3   Median :59.00   Median :38.50   Median :22.8  
 Mean   :100.82   Mean   : 92.56   Mean   : 99.9   Mean   :59.41   Mean   :38.59   Mean   :23.1  
 3rd Qu.:105.38   3rd Qu.: 99.33   3rd Qu.:103.5   3rd Qu.:62.35   3rd Qu.:39.92   3rd Qu.:24.0  
 Max.   :136.20   Max.   :148.10   Max.   :147.7   Max.   :87.30   Max.   :49.10   Max.   :33.9  
     Biceps          Forearm          Wrist                      BMI     
 Min.   :24.80   Min.   :21.00   Min.   :15.80   Underweight     :  1  
 1st Qu.:30.20   1st Qu.:27.30   1st Qu.:17.60   Normal          :123  
 Median :32.05   Median :28.70   Median :18.30   Overweight      :103  
 Mean   :32.27   Mean   :28.66   Mean   :18.23   Moderately Obese: 21  
 3rd Qu.:34.33   3rd Qu.:30.00   3rd Qu.:18.80   Severely Obese  :  2  
 Max.   :45.00   Max.   :34.90   Max.   :21.40   Morbidly Obese  :  2  
```

Figure 54: Summary of the data

3.3 Data splitting

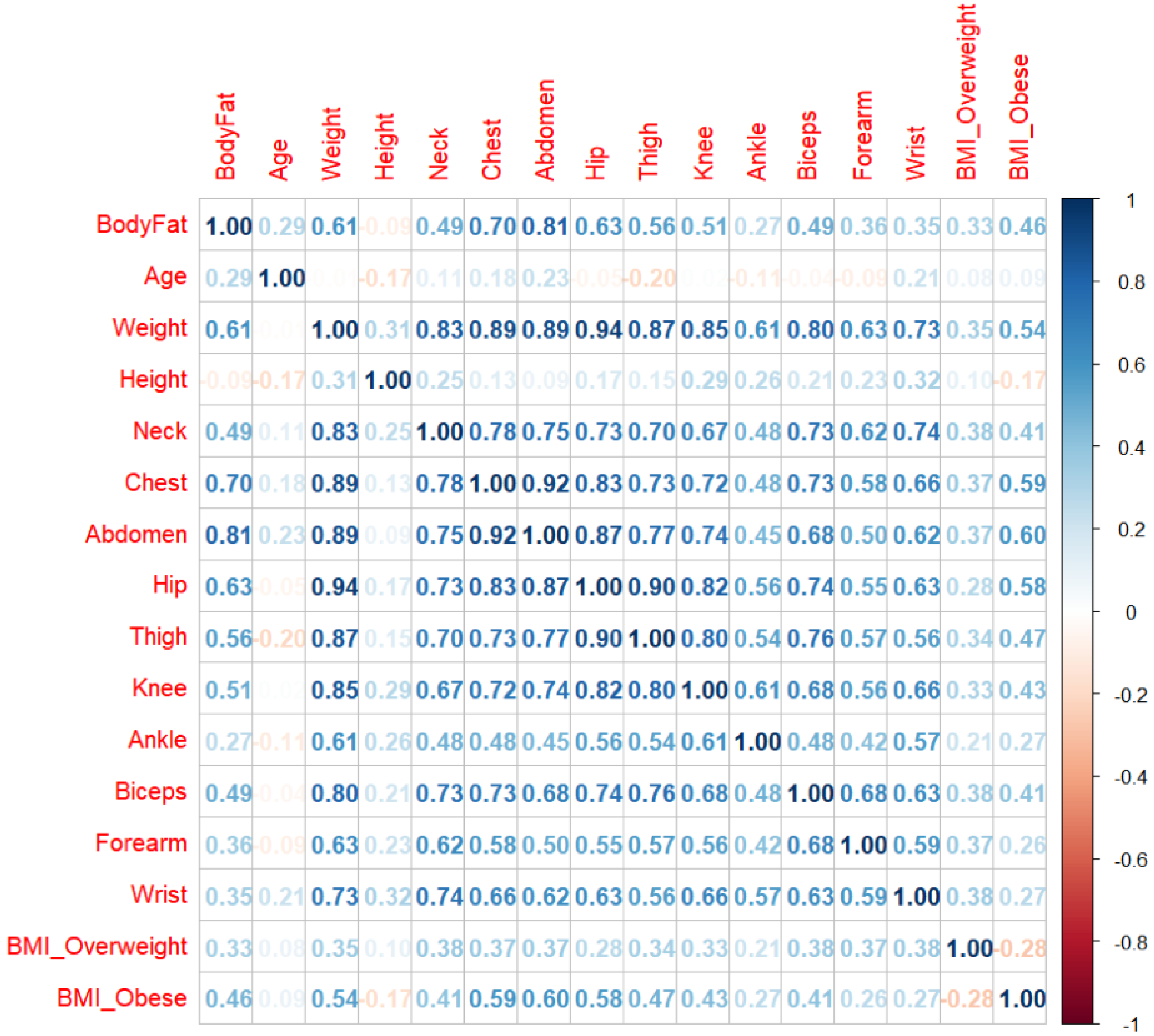
We split the data into training and testing sets. The ratio of the training set to the testing set is 80:20.

The training set now has 201 rows while the testing set has 51, which is approximately 80% and 20% of the original data, respectively.

Code for data splitting can be found in Section C.2.

3.4 Descriptive Statistics

From the Figure 55, it seems that ”Neck”, ”Chest”, ”Abdomen”, ”Hip”, ”Thigh”, ”Knee”, ”Biceps”, ”Forearm”, and ”Wrist” have relatively large correlations, we will try to eliminate variables in this group after obtaining the baseline model. We also notice that the correlation between ”BodyFat” and ”Height”, ”BodyFat” and ”Ankle” is not high (-0.09 and 0.27), it seems like these variables will not be important in our regression model.



	BodyFat	Age	Weight	Height	Neck	Chest	Abdomen	Hip	Thigh	Knee	Ankle	Biceps	Forearm	Wrist	BMI_Overweight	BMI_Obese
BodyFat	1.00	0.29	0.61	-0.09	0.49	0.70	0.81	0.63	0.56	0.51	0.27	0.49	0.36	0.35	0.33	0.46
Age	0.29	1.00	-0.01	-0.17	0.11	0.18	0.23	-0.05	-0.20	0.02	-0.11	-0.04	-0.09	0.21	0.08	0.09
Weight	0.61	-0.01	1.00	0.31	0.83	0.89	0.89	0.94	0.87	0.85	0.61	0.80	0.63	0.73	0.35	0.54
Height	-0.09	-0.17	0.31	1.00	0.25	0.13	0.09	0.17	0.15	0.29	0.26	0.21	0.23	0.32	0.10	-0.17
Neck	0.49	0.11	0.83	0.25	1.00	0.78	0.75	0.73	0.70	0.67	0.48	0.73	0.62	0.74	0.38	0.41
Chest	0.70	0.18	0.89	0.13	0.78	1.00	0.92	0.83	0.73	0.72	0.48	0.73	0.58	0.66	0.37	0.59
Abdomen	0.81	0.23	0.89	0.09	0.75	0.92	1.00	0.87	0.77	0.74	0.45	0.68	0.50	0.62	0.37	0.60
Hip	0.63	-0.05	0.94	0.17	0.73	0.83	0.87	1.00	0.90	0.82	0.56	0.74	0.55	0.63	0.28	0.58
Thigh	0.56	-0.20	0.87	0.15	0.70	0.73	0.77	0.90	1.00	0.80	0.54	0.76	0.57	0.56	0.34	0.47
Knee	0.51	0.02	0.85	0.29	0.67	0.72	0.74	0.82	0.80	1.00	0.61	0.68	0.56	0.66	0.33	0.43
Ankle	0.27	-0.11	0.61	0.26	0.48	0.48	0.45	0.56	0.54	0.61	1.00	0.48	0.42	0.57	0.21	0.27
Biceps	0.49	-0.04	0.80	0.21	0.73	0.73	0.68	0.74	0.76	0.68	0.48	1.00	0.68	0.63	0.38	0.41
Forearm	0.36	-0.09	0.63	0.23	0.62	0.58	0.50	0.55	0.57	0.56	0.42	0.68	1.00	0.59	0.37	0.26
Wrist	0.35	0.21	0.73	0.32	0.74	0.66	0.62	0.63	0.56	0.66	0.57	0.63	0.59	1.00	0.38	0.27
BMI_Overweight	0.33	0.08	0.35	0.10	0.38	0.37	0.37	0.28	0.34	0.33	0.21	0.38	0.37	0.38	1.00	-0.28
BMI_Obese	0.46	0.09	0.54	-0.17	0.41	0.59	0.60	0.58	0.47	0.43	0.27	0.41	0.26	0.27	-0.28	1.00

Figure 55: Correlation matrix for all variables in the data in Task 2 - Dataset 2

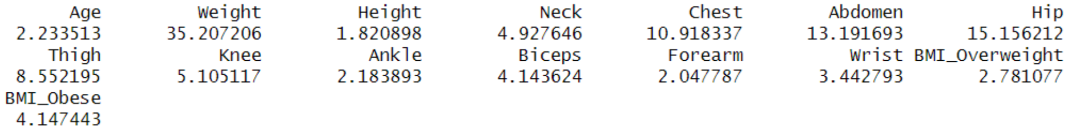
3.5 Baseline model

In this section, we try to create a baseline linear model. Codes for this section can be found in Section C.3.

We fit a linear regression model with all original variables to predict ”BodyFat”. Then we use the car::vif function to check the multicollinearity.

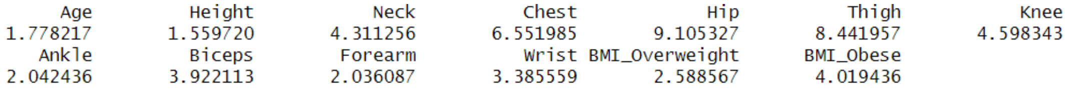
3.5.1 Multicollinearity

To check for multicollinearity of the baseline model, we calculate the Variance Inflation Factor (VIF) of the variables as in Figure 56. After introducing many new variables, the VIF values of many variables are very high. Therefore, we progressively remove the variable with the highest VIF value and repeat calculating the VIF values until all VIF values are less than 10. The remaining variables and their corresponding VIF values after this process can be viewed in Figure 57.



```
      Age       Weight       Height         Neck        Chest      Abdomen          Hip 
 2.233513    35.207206     1.820898     4.927646    10.918337    13.191693    15.156212 
    Thigh         Knee        Ankle       Biceps      Forearm        Wrist BMI_Overweight 
 8.552195     5.105117     2.183893     4.143624     2.047787     3.442793     2.781077 
BMI_Obese 
 4.147443 
```

Figure 56: VIF of baseline model in Task 2 - Dataset 2



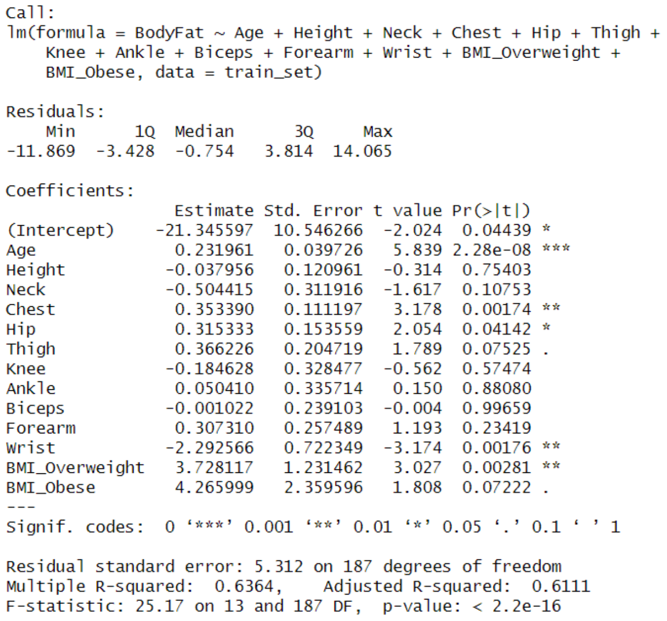
```
      Age       Height         Neck        Chest          Hip        Thigh         Knee 
 1.778217     1.559720     4.311256     6.551985     9.105327     8.441957     4.598343 
    Ankle       Biceps      Forearm        Wrist BMI_Overweight    BMI_Obese 
 2.042436     3.922113     2.036087     3.385559     2.588567     4.019436 
```

Figure 57: VIF of baseline model without ”Weight” and ”Abdomen” in Task 2 - Dataset 2

Now all variables have VIF values less than 10, indicating no strong multicollinearity in the model.

3.6 Reduced model

Summary of the baseline model after eliminating the multicollinearity can be viewed in Figure 58.

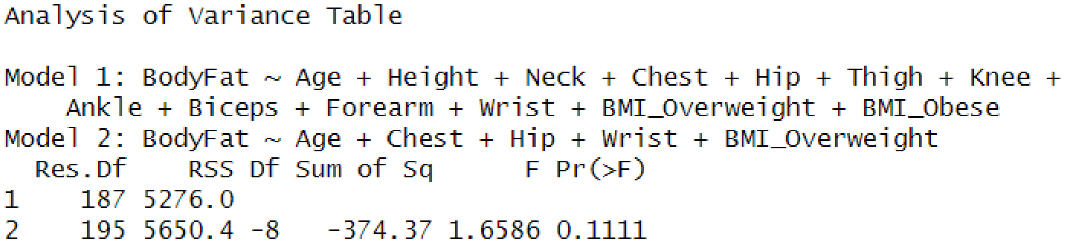


```
Call:
lm(formula = BodyFat ~ Age + Height + Neck + Chest + Hip + Thigh +
    Knee + Ankle + Biceps + Forearm + Wrist + BMI_Overweight +
    BMI_Obese, data = train_set)

Residuals:
    Min      1Q  Median      3Q     Max
-11.869  -3.428  -0.754   3.814  14.065

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)    -21.345597  10.546266  -2.024  0.04439 *
Age              0.231961   0.039726   5.839 2.28e-08 ***
Height          -0.037956   0.120961  -0.314  0.75403
Neck            -0.504415   0.311916  -1.617  0.10753
Chest            0.353390   0.111197   3.178  0.00174 **
Hip              0.315333   0.153559   2.054  0.04142 *
Thigh            0.366226   0.204719   1.789  0.07525 .
Knee            -0.184628   0.328477  -0.562  0.57474
Ankle            0.050410   0.335714   0.150  0.88080
Biceps          -0.001022   0.239103  -0.004  0.99659
Forearm          0.307310   0.257489   1.193  0.23419
Wrist           -2.292566   0.722349  -3.174  0.00176 **
BMI_Overweight   3.728117   1.231462   3.027  0.00281 **
BMI_Obese        4.265999   2.359596   1.808  0.07222 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.312 on 187 degrees of freedom
Multiple R-squared:  0.6364,	Adjusted R-squared:  0.6111
F-statistic: 25.17 on 13 and 187 DF,  p-value: < 2.2e-16
```

Figure 58: Summary of baseline model in Task 2 - Dataset 2

We will create a reduced model that does not include variables whose p-value is greater 0.05 (BodyFat ∼ Age + Chest + Hip + Wrist + BMI_Overweight) and use partial F-test (ANOVA function) to check the significance of variables whose p-value is greater than 0.05.

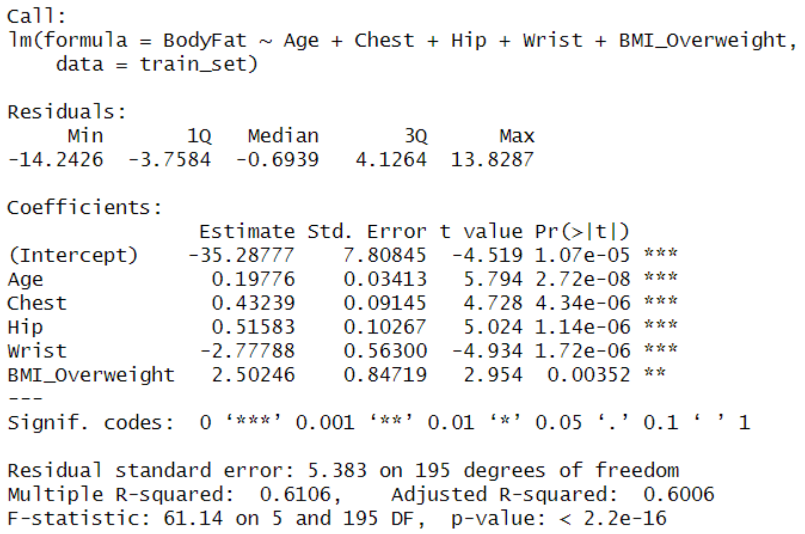
From the result of ANOVA function in Figure 59, we can see that the p-value is greater than the significance level of 0.05, indicating that the variables, whose p-value in the summary of the baseline model is greater than 0.05, are not significant. Then, our current model is changed to (BodyFat ∼ Age + Chest + Hip + Wrist + BMI_Overweight).



```
Analysis of Variance Table

Model 1: BodyFat ~ Age + Height + Neck + Chest + Hip + Thigh + Knee +
    Ankle + Biceps + Forearm + Wrist + BMI_Overweight + BMI_Obese
Model 2: BodyFat ~ Age + Chest + Hip + Wrist + BMI_Overweight
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    187 5276.0
2    195 5650.4 -8   -374.37 1.6586 0.1111
```

Figure 59: The result of the ANOVA function of the partial F-test between the baseline model and the reduced model in Task 2 - Dataset 2

The summary for our current model can be viewed as Figure 60.



```
Call:
lm(formula = BodyFat ~ Age + Chest + Hip + Wrist + BMI_Overweight,
    data = train_set)

Residuals:
     Min       1Q   Median       3Q      Max
-14.2426  -3.7584  -0.6939   4.1264  13.8287

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    -35.28777    7.80845  -4.519 1.07e-05 ***
Age              0.19776    0.03413   5.794 2.72e-08 ***
Chest            0.43239    0.09145   4.728 4.34e-06 ***
Hip              0.51583    0.10267   5.024 1.14e-06 ***
Wrist           -2.77788    0.56300  -4.934 1.72e-06 ***
BMI_Overweight   2.50246    0.84719   2.954  0.00352 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.383 on 195 degrees of freedom
Multiple R-squared:  0.6106,    Adjusted R-squared:  0.6006
F-statistic: 61.14 on 5 and 195 DF,  p-value: < 2.2e-16
```

Figure 60: The summary of the reduced model in Task 2 - Dataset 2

3.7 Stepwise Algorithm

We apply the Stepwise Algorithm to select the best model. The result of the Stepwise Algorithm says that the current model found in the ”Reduced model” section is the best model. Then, we will choose (BodyFat ∼ Age + Chest + Hip + Wrist + BMI_Overweight) as the final model.

3.8 Model diagnostics for baseline model

We now conduct some diagnostics on the baseline model to verify its ideality for the task.

- Normality of residuals

  We now plot the Q-Q plot (Figure 61) of the residuals to check for normality. The residuals are close to the line in the middle, suggesting that the residuals in this range are approximately normally distributed.

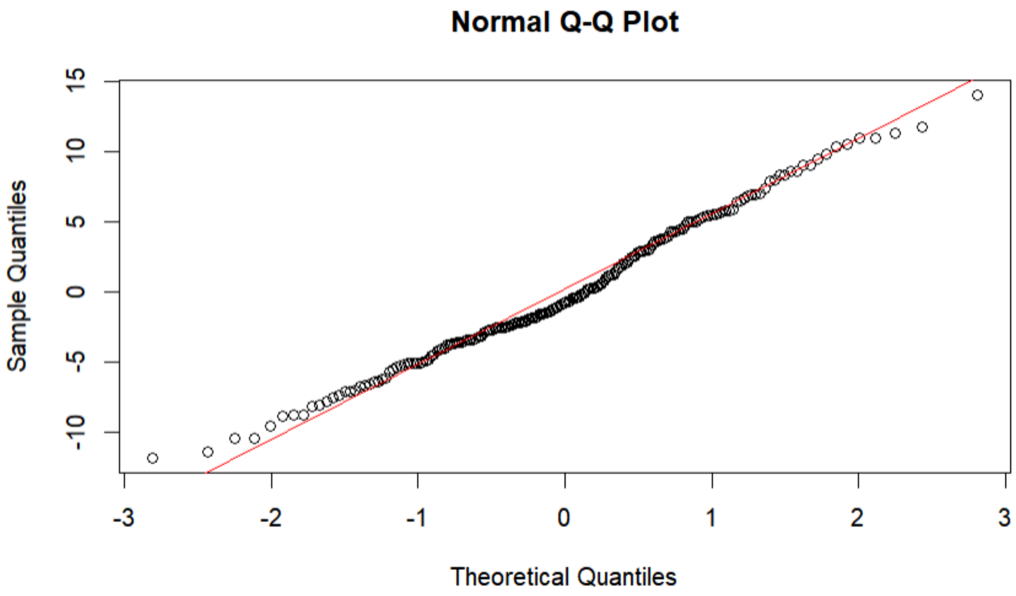
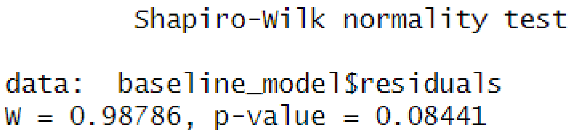


Figure 61: Q-Q plot of the residuals in the final model in Task 2 - Dataset 2

To test the normality of the residuals, we perform the Shapiro-Wilk test in Figure 62. The p-value of the Shapiro-Wilk test is greater than 0.05, indicating that the residuals are normally distributed. This is consistent with the Q-Q plot.

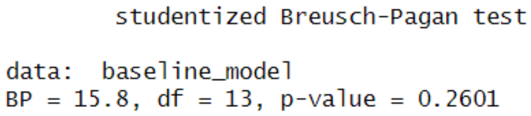


```
                Shapiro-Wilk normality test

data:  baseline_model$residuals
W = 0.98786, p-value = 0.08441
```

Figure 62: Shaprio-Wilk test of the final model in Task 2 - Dataset 2

- Homoscedasticity

  We also perform the Breusch-Pagan test to check for homoscedasticity as in Figure 63. The p-value of the Breusch-Pagan test is greater than 0.05, indicating that the residuals are homoscedastic.

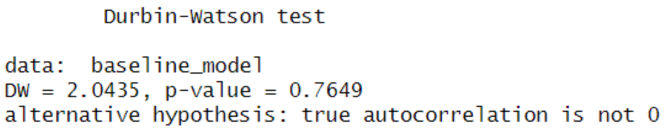


```
                studentized Breusch-Pagan test

data:  baseline_model
BP = 15.8, df = 13, p-value = 0.2601
```

Figure 63: Breusch-Pagan test of the final model in Task 2 - Dataset 2

- Autocorrelation

We perform the Durbin-Watson test to check for autocorrelation as in Figure 64. The Durbin-Watson test gives a DW value close to 2, indicating that there is no autocorrelation in the residuals.



```
                Durbin-Watson test

data:  baseline_model
DW = 2.0435, p-value = 0.7649
alternative hypothesis: true autocorrelation is not 0
```

Figure 64: Durbin-Watson test of the final model in Task 2 - Dataset 2

- Conclusion on model diagnostics

  In conclusion, the model is ideal in theory as the residuals are normally distributed and homoscedastic. Moreover, there is no autocorrelation in the residuals. The R-squared on the training set (Figure 58) is also acceptable.

3.9 Model diagnostics for final model

We now conduct some diagnostics on the final model to verify its ideality for the task.

- Normality of residuals

  We now plot the Q-Q plot (Figure 65) of the residuals to check for normality. The residuals are close to the line in the middle, suggesting that the residuals in this range are approximately normally distributed.

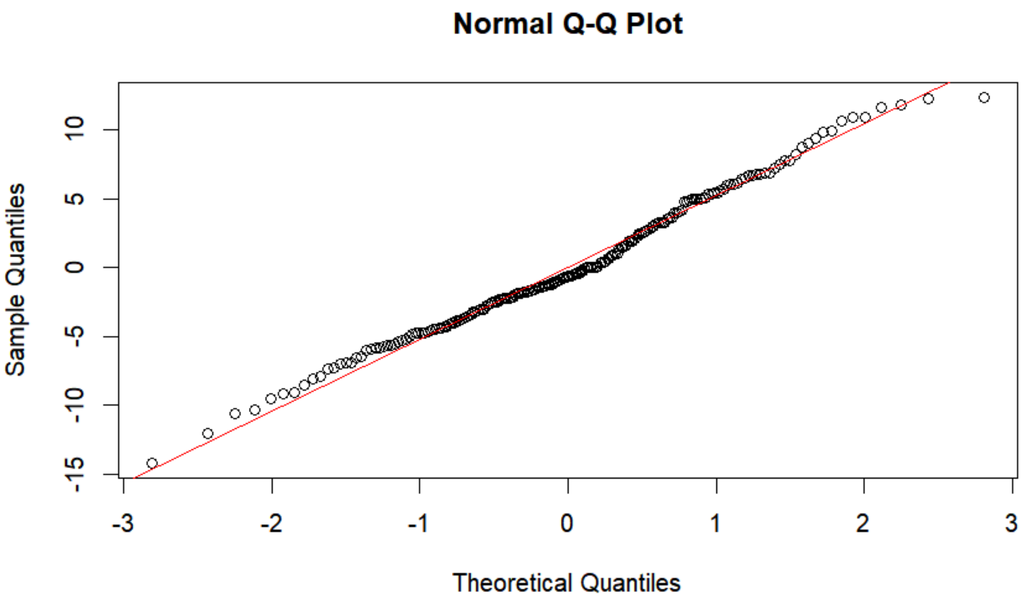
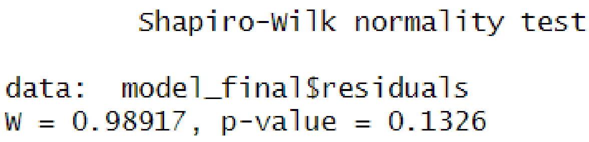


Figure 65: Q-Q plot of the residuals in the final model in Task 2 - Dataset 2

To test the normality of the residuals, we perform the Shapiro-Wilk test in Figure 66. The p-value of the Shapiro-Wilk test is greater than 0.05, indicating that the residuals are normally distributed. This is consistent with the Q-Q plot.

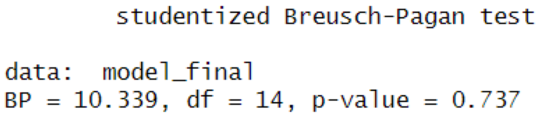


```
        Shapiro-Wilk normality test

data:  model_final$residuals
W = 0.98917, p-value = 0.1326
```

Figure 66: Shaprio-Wilk test of the final model in Task 2 - Dataset 2

- Homoscedasticity

  We also perform the Breusch-Pagan test to check for homoscedasticity as in Figure 67. The p-value of the Breusch-Pagan test is greater than 0.05, indicating that the residuals are homoscedastic.

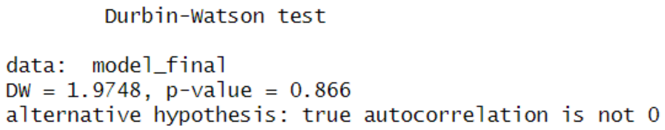


```
        studentized Breusch-Pagan test

data:  model_final
BP = 10.339, df = 14, p-value = 0.737
```

Figure 67: Breusch-Pagan test of the final model in Task 2 - Dataset 2

- Autocorrelation

We perform the Durbin-Watson test to check for autocorrelation as in Figure 68. The Durbin-Watson test gives a DW value close to 2, indicating that there is no autocorrelation in the residuals.



```
                Durbin-Watson test

data:  model_final
DW = 1.9748, p-value = 0.866
alternative hypothesis: true autocorrelation is not 0
```

Figure 68: Durbin-Watson test of the final model in Task 2 - Dataset 2

- Conclusion on model diagnostics

  In conclusion, the model is ideal in theory as the residuals are normally distributed and homoscedastic. Moreover, there is no autocorrelation in the residuals. The R-squared on the training set (Figure 60) is also acceptable.

3.10 Model testing

The R-squared of the baseline model and final model on two training/testing sets can be viewed in Table 3. Even though the proposed linear model is theoretically ideal for the task, our R-squared result is not impressive. However, this is still an acceptable result. The code for testing the baseline model and the final model can be found in Section C.4.

Although the final model has a lower R-squared than the baseline model, it is much less variable than the baseline model and the R-squared difference is not too much (only 0.0258 for training set and 0.0159 for testing set).

	Training Set	Testing Set	Number of variables
Baseline model	**0.6364**	**0.5934**	13
Final model	**0.6106**	**0.5775**	5

Table 3: R-squared result of the baseline model and final model on training/testing set in Task 2 - Dataset 2

3.11 Conclusion

In this task, we first preprocessed the data by removing the ”Density” column and adding 2 dummy variables for the ”BMI” column. We then split the data into training and testing sets. We fitted a baseline linear regression model with all original variables to predict ”BodyFat” and then reduced it based on its summary model.

According to the ”Model diagnostics for baseline model” section and the ”Model diagnostics for final model” section, the baseline model and the final model is ideal as the residuals are normally distributed and homoscedastic, there is no autocorrelation in the residuals, and R squared is not bad.

Finally, we test the baseline model and the final model on the test set and find that the R-squared values on the test set is not as high as the R-squared value on the training set but it does not deviate too much.

References

A Task 1

A.1 Data preprocessing code

```
# Remove "car name"
df$car_name <- NULL

# Remove rows with missing values
nrow(df[df$horsepower == ’?’,]) #6 rows with missing value in horsepower
df <- df[df$horsepower != ’?’,]

# Convert horsepower to numeric
is.numeric(df$horsepower) #FALSE
# As we can see, the horsepower column is not numeric, so we convert it to numeric.
df$horsepower <- as.numeric(df$horsepower)
is.numeric(df$horsepower) #TRUE

# Remove duplicate rows
#We count the number of duplicate rows in the data.
nrow(df[duplicated(df),]) #0
# There are no duplicate rows in the data. Therefore, we do not need to remove any rows.

# Plot bar plots of all variables

# Determine the number of columns in df
num_cols <- ncol(df)
# Calculate the number of rows and columns for the plot layout
plot_rows <- 2
plot_cols <- 4
# Set up the plot layout
par(mfrow = c(plot_rows, plot_cols))
# Loop through each column and create a barplot
```

```r
for (col_name in names(df)) {
  if (col_name == ’mpg’) {
    next
  }
  # Get counts for the current column
  col_counts <- table(df[[col_name]])
  # Create a barplot for the current column
  barplot(col_counts,
          main = col_name,
          xlab = "",
          ylab = "Count",
          col = rainbow(length(col_counts)),
          las = 2)  # Rotate x-axis labels if needed
}
# Reset the plot layout
par(mfrow = c(1, 1))

# The "cylinders" column
is.factor(df$cylinders)
cylinders_counts <- table(df$cylinders)
barplot(cylinders_counts, main = "Cylinders Distribution", xlab = "Cylinders", ylab = "Count", col = rainbow(length(cylinders_counts)))

# Even though the "cylinders" column has only 5 values, we will not change it to a factor because it is quantitatively meaningful.

# The "origin" column
is.factor(df$origin)
origin_counts <- table(df$origin)
barplot(origin_counts, main = "Origin Distribution", xlab = "Origin", ylab = "Count", col = rainbow(length(origin_counts)))
# As we can see, the "origin" column has only 3 values. Moreover, these 3 values are not quantitatively meaningful. Therefore, we will change the "origin" column to a factor.

df$origin <- as.factor(df$origin)
```

A.2 Data splitting code

```
# Define the split ratio
train_ratio <- 0.8

# Determine the number of rows in the training set
train_size <- floor(train_ratio * nrow(df))

# Randomly sample row indices for the training set
train_indices <- sample(seq_len(nrow(df)), size = train_size)

# Split the data into training and testing sets
train_set <- df[train_indices, ]
test_set <- df[-train_indices, ]

# Display the number of rows in each set
dim(train_set) # 313, which is 80%*392
dim(test_set)  # 79, which is 20%*392
```

A.3 Baseline model code

```
# Baseline model
baseline <- lm(mpg ˜ ., data = train_set)
summary(baseline)

# Check for multicollinearity
vif(baseline)

# Remove displacement variable due to high VIF
baseline <- update(baseline, . ˜ . - displacement)
vif(baseline)

# Stepwise Algorithm to select the best model
step_baseline <- step(baseline, direction = "both")
summary(step_baseline)

# ANOVA test to validate removed variables
```

```r
anova(step_baseline, baseline)

# Check for multicollinearity in the final model
vif(step_baseline)

# Model diagnostics
# Normality of residuals
qqnorm(step_baseline$residuals)
qqline(step_baseline$residuals, col = "red")

# Shapiro-Wilk test for normality
shapiro.test(step_baseline$residuals)

# Breusch-Pagan test for homoscedasticity
lmtest::bptest(step_baseline)

# Durbin-Watson test for autocorrelation
lmtest::dwtest(step_baseline, alternative = "two.sided")
```

A.4 Improved model code

```r
# Adding new variables

# Plotting boxplots to investigate the relationship between "mpg" and other variables
par(mfrow = c(2, 4))

for (col in names(train_set)) {
  if (is.numeric(train_set[[col]]))
    boxplot(train_set[col], main = col)
}

par(mfrow = c(1, 1))

# Remove outliers in "horsepower" using IQR method
identify_outliers_IQR <- function(x) {
  Q1 <- quantile(x, 0.25)
  Q3 <- quantile(x, 0.75)
  IQR <- Q3 - Q1
  lower_bound <- Q1 - 1.5 * IQR
  upper_bound <- Q3 + 1.5 * IQR
  !(x >= lower_bound & x <= upper_bound)
}

outlier_matrix <- train_set %>%
  select(where(is.numeric)) %>%
  mutate(across(everything(), identify_outliers_IQR))

train_set <- train_set[!apply(outlier_matrix, 1, any), ]

dim(train_set)

# Plotting scatter plots to investigate curvilinear relationships
pairs(subset(train_set, select = - c(origin)))

# Adding log transformations
train_set$log_horsepower <- log(train_set$horsepower)
train_set$log_displacement <- log(train_set$displacement)
train_set$log_weight <- log(train_set$weight)

# Adding interaction and polynomial terms
new_model <- lm(mpg ~ . + I(weight * acceleration) + I((model_year-70)^2), data = train_set)
summary(new_model)

# Checking for multicollinearity and progressively removing variables with high VIF
vif(new_model)
new_model <- update(new_model, . ~ . -weight)
vif(new_model)
new_model <- update(new_model, . ~ . -horsepower)
vif(new_model)
new_model <- update(new_model, . ~ . -log_weight)
vif(new_model)
```

```r
new_model <- update(new_model, . ~ . -log_displacement)
vif(new_model)
new_model <- update(new_model, . ~ . -displacement)
vif(new_model)
new_model <- update(new_model, . ~ . -model_year)
vif(new_model)

# Applying Stepwise Algorithm to select the best model
stepwise_new_model <- step(new_model, direction = "both")
summary(stepwise_new_model)

# ANOVA test to validate insignificant variables
anova(stepwise_new_model, update(stepwise_new_model , . ~ . -cylinders -acceleration))

# Final model after removing insignificant variables
final_model <- update(stepwise_new_model, . ~ . -cylinders -acceleration)
summary(final_model)

# Checking for multicollinearity in the final model
vif(final_model)

# Model diagnostics

# Normality of residuals
qqnorm(final_model$residuals)
qqline(final_model$residuals, col = "red")

# Shapiro-Wilk test for normality
shapiro.test(final_model$residuals)

# Breusch-Pagan test for homoscedasticity
lmtest::bptest(final_model)

# Durbin-Watson test for autocorrelation
lmtest::dwtest(final_model, alternative = "two.sided")
```

A.5 Model testing code

```
# Testing the baseline model on the test set

# Generate predictions
predictions <- predict(step_baseline, newdata = test_set)

# Actual values from the test set
actual_values <- test_set$mpg

# Calculate Mean Squared Error (MSE)
mse <- mean((predictions - actual_values)^2)

# Calculate R-squared
rss <- sum((predictions - actual_values)^2)
tss <- sum((actual_values - mean(actual_values))^2)
r_squared <- 1 - (rss / tss)

# Print metrics
cat("Mean Squared Error (MSE):", mse, "\n")
cat("R-squared:", r_squared, "\n")
```

```
# Model testing for the improved model

# Add log transformations to the test set
test_set$log_horsepower <- log(test_set$horsepower)
test_set$log_displacement <- log(test_set$displacement)
test_set$log_weight <- log(test_set$weight)

# Generate predictions using the final model
predictions <- predict(final_model, newdata = test_set)

# Actual values from the test set
actual_values <- test_set$mpg

# Calculate Mean Squared Error (MSE)
mse <- mean((predictions - actual_values)^2)

```

```r
# Calculate R-squared
rss <- sum((predictions - actual_values)^2)
tss <- sum((actual_values - mean(actual_values))^2)
r_squared <- 1 - (rss / tss)

# Print metrics
cat("Mean Squared Error (MSE):", mse, "\n")
cat("R-squared:", r_squared, "\n")
```

B Task 2 - Dataset 1

B.1 Data preprocessing code

```r
df <- read.csv("Fish.csv", header = TRUE)
head(df)

sum(is.na(df))
sum(duplicated(df))
dim(df)
summary(df)

df$Length <- (df$Length1 + df$Length2 + df$Length3)/3
summary(df$Length)

df$Length1 <- NULL
df$Length2 <- NULL
df$Length3 <- NULL
```

B.2 Histogram and Plots code

```r

ggplot(df, aes(x = Species)) +
    geom_bar(fill = "skyblue", color = "black", alpha = 0.7) +
    theme_minimal() +
    labs(title = "Distribution of Fish Species", x = "Species", y = "Count") +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```r

df_long <- subset(df, select = -c(Species)) %>%
  pivot_longer(cols = everything(), names_to = "variable", values_to = "value")

# Plotting histograms for all variables
ggplot(df_long, aes(x = value)) +
  geom_histogram(fill = "skyblue", color = "black", alpha = 0.7, bins=50) +
  facet_wrap(~variable, scales = "free") +
  theme_minimal() +
  labs(title = "Histograms of All Variables in df", x = "Value", y = "Frequency")

ggplot(df, aes(x = Species, y = Height)) +
  geom_boxplot(fill = "skyblue", color = "black", alpha = 0.7) +
  theme_minimal() +
  labs(title = "Height Distribution by Species", x = "Species", y = "Height") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))


ggplot(df, aes(x = Species, y = Width)) +
  geom_boxplot(fill = "skyblue", color = "black", alpha = 0.7) +
  theme_minimal() +
  labs(title = "Width Distribution by Species", x = "Species", y = "Width") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

ggplot(df, aes(x = Species, y = Length)) +
  geom_boxplot(fill = "skyblue", color = "black", alpha = 0.7) +
  theme_minimal() +
  labs(title = "Length Distribution by Species", x = "Species", y = "Length") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

ggplot(df, aes(x = Species, y = Weight)) +
  geom_boxplot(fill = "skyblue", color = "black", alpha = 0.7) +
  theme_minimal() +
  labs(title = "Weight Distribution by Species", x = "Species", y = "Weight") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

```

```
par(mfrow = c(1, 4))

for (col in names(df)) {
  if (is.numeric(df[[col]]))
    boxplot(df[col], main = col)
}

par(mfrow = c(1, 1))
```

B.3 Data splitting code

```
df$Species <- as.factor(df$Species)

### Split
# Define the split ratio
train_ratio <- 0.8

# Determine the number of rows in the training set
train_size <- floor(train_ratio * nrow(df))

# Randomly sample row indices for the training set
train_indices <- sample(seq_len(nrow(df)), size = train_size)

# Split the data into training and testing sets
train_set <- df[train_indices, ]
test_set <- df[-train_indices, ]

# Display the number of rows in each set
dim(train_set) # Should be approximately 80
dim(test_set)  # Should be approximately 20
```

B.4 Baseline code

```
model1_baseline <- lm(Weight ~ ., data = train_set)
summary(model1_baseline)
```

B.5 Multicollinearity code

```r
corrplot(cor(subset(df, select = -Species)), method = "number")

vif(model1_baseline)

model1_vif<-update(model1_baseline, . ~ . -Height)
vif(model1_vif)

model1_vif<-update(model1_vif, . ~ . -Length)
vif(model1_vif)
```

B.6 First Degree model code

```r
summary(model1_vif)

model1_step <- step(model1_baseline, direction="backward")

summary(model1_step)

anova(model1_vif, model1_baseline)

anova(model1_step, model1_baseline)
model1 <- model1_step
```

B.7 Model testing of First degree model code

```r
predictions <- predict(model1, newdata = test_set)
# Actual values from the test set
actual_values <- test_set$Weight
# Calculate Mean Squared Error (MSE)
mse <- mean((predictions - actual_values)^2)
# Calculate R-squared
rss <- sum((predictions - actual_values)^2)
tss <- sum((actual_values - mean(actual_values))^2)
r_squared <- 1 - (rss / tss)
```

```
# Print metrics
cat("Mean Squared Error (MSE):", mse, "\n")

cat("R-squared:", r_squared, "\n")
```

B.8 Model diagnostics of First degree model code

```
e <- model1$residuals
qqnorm(e)
qqline(e, col = "red")

shapiro.test(e)

bptest(model1)

dwtest(model1, alternative = "two.sided")
```

B.9 Quadaric Degree model code

```
model2_baseline <- lm(Weight ~ Species + Length + Width + Height + I(Length^2) + I(Width^2) +
I(Height^2) + I(Width * Height) + I(Width*Length) + I(Height*Length), data = train_set)
summary(model2_baseline)

model2 <- step(model2_baseline, direction = "backward")

summary(model2)

anova(model2, model2_baseline)
```

B.10 Model testing of Quadaric degree model code

```
predictions <- predict(model2, newdata = test_set)
# Actual values from the test set
actual_values <- test_set$Weight
```

```
# Calculate Mean Squared Error (MSE)
mse <- mean((predictions - actual_values)^2)
# Calculate R-squared
rss <- sum((predictions - actual_values)^2)
tss <- sum((actual_values - mean(actual_values))^2)
r_squared <- 1 - (rss / tss)
# Print metrics
cat("Mean Squared Error (MSE):", mse, "\n")

cat("R-squared:", r_squared, "\n")
```

B.11 Model diagnostics of Quadaric degree model code

```
e <- model2$residuals
qqnorm(e)
qqline(e, col = "red")

shapiro.test(e)

bptest(model2)

dwtest(model2, alternative = "two.sided")
```

B.12 Compare with first model code

```
anova(model1, model2)
```

C Task 2 - Dataset 2

C.1 Create the ”BMI” column

```
df[’BMI’] = 703 * df$Weight / (df$Height * df$Height)
df$BMI <- cut(df$BMI,
                       breaks = c(-Inf, 18.4, 24.9, 29.9, 34.9, 39.9, Inf),
                       labels = c("Underweight",
                                  "Normal",
                                  "Overweight",
                                  "Moderately Obese",
                                  "Severely Obese",
                                  "Morbidly Obese"),
                       right = TRUE)
```

C.2 Data splitting code

```
# Define the split ratio
train_ratio <- 0.8

# Determine the number of rows in the training set
train_size <- floor(train_ratio * nrow(df))

# Randomly sample row indices for the training set
train_indices <- sample(seq_len(nrow(df)), size = train_size)

# Split the data into training and testing sets
train_set <- df[train_indices, ]
test_set <- df[-train_indices, ]

# Display the number of rows in each set
dim(train_set) # 201 - Should be approximately 80%
dim(test_set)  # 51  - Should be approximately 20%
```

C.3 Baseline model code

```
# Baseline model
model_full = lm(BodyFat ~ ., data = train_set)

# Check for multicollinearity
vif(model_full)

# Remove Weight variable due to high VIF
model_full = update(model_full, . ~ . - Weight)
```

```r
vif(model_full)

# Remove Abdomen variable due to high VIF
model_full = update(model_full, . ~ . - Abdomen, data = train_set)
vif(model_full)

# Summary model
baseline_model = model_full
summary(model_full)

# I will try to remove variables that has the p-value is greater than 0.05
model_reduced = update(model_full, . ~ . - Height - Neck - Thigh - Knee - Ankle - Biceps - Forearm - BMI_Obese)
anova(model_full, model_reduced)

# since the p-value of the partial F-test is 0.1111 > 0.05
# we choose the reduced model
model_full = model_reduced
summary(model_full)

# Stepwise Algorithm to select the best model
model_bw = step(model_full, data = train_set, direction = "backward")

# since Stepwise Algorithm do nothing, our final model is the full model
model_final = model_full
summary(model_final)

# Model diagnostics
# Normality of residuals
qqnorm(model_final$residuals)
qqline(model_final$residuals, col = "red")

# Shapiro-Wilk test for normality
shapiro.test(model_final$residuals)

# Breusch-Pagan test for homoscedasticity
lmtest::bptest(model_final)
```

```r

# Durbin-Watson test for autocorrelation
lmtest::dwtest(model_final, alternative="two.sided")
```

C.4 Model testing code

```r
# Testing the final model on the test set

# Generate predictions
predictions <- predict(model_final, newdata = test_set)

# Actual values from the test set
actual_values <- test_set$BodyFat

# Calculate Mean Squared Error (MSE)
mse <- mean((predictions - actual_values)^2)

# Calculate R-squared
rss <- sum((predictions - actual_values)^2)
tss <- sum((actual_values - mean(actual_values))^2)
r_squared <- 1 - (rss / tss)

# Print metrics
cat("R-squared:", r_squared, "\n")
```

```r
# Testing the baseline model on the test set

# Generate predictions
predictions <- predict(baseline_model, newdata = test_set)

# Actual values from the test set
actual_values <- test_set$BodyFat

# Calculate Mean Squared Error (MSE)
mse <- mean((predictions - actual_values)^2)

# Calculate R-squared
```

```r
rss <- sum((predictions - actual_values)^2)
tss <- sum((actual_values - mean(actual_values))^2)
r_squared <- 1 - (rss / tss)

# Print metrics
cat("R-squared:", r_squared, "\n")
```