STAT452 - Final Project Report

Final Project Report

STAT452

Group's member:

Le Van Cuong - 22125013

Truong Minh Dat - 22125015

Bui Danh Nghe - 22125063

Phung Khanh Vinh - 22125122

Contents

1 Task Distribution 1

2 Question 1 1

2.1 Data Preprocessing . . . 1

2.1.1 Missing data . . . 1

2.1.2 Outlier . . . 1

2.2 Data overview . . . 2

2.3 Data splitting . . . 4

2.4 Model building . . . 4

2.4.1 Response . . . 4

2.4.2 Predictor . . . 5

2.4.3 Model selection method . . . 10

2.4.4 Evaluation parameter . . . 10

2.4.5 Baseline model . . . 11

2.5 Improving to a better model . . . 13

2.5.1 Fixing mpg skewness . . . 13

2.5.2 Multicollinearity . . . 14

2.5.3 Displacement statistical significance issues . . . 15

2.5.4 Transformation of weight . . . 16

2.5.5 Transformation of acceleration . . . 18

2.5.6 Model year as multi-value discrete variable vs. continuous variable . . . 21

2.5.7 The improved model . . . 22

2.6 Hypothesis testing . . . 23

2.6.1 Model utility . . . 23

2.6.2 Predictor statistical significance . . . 23

2.6.3 Error normality assumption . . . 24

2.6.4 Homoskedastic residual variance . . . 25

2.6.5 Autocorrelation . . . 25

2.7 Validation and testing . . . 26

2.8 Model meaning . . . 28

3 Question 2 - Dataset 1 30

3.1 Introduction . . . 30

3.2 Dataset Overview . . . 30

3.2.1 Data Cleaning . . . 31

3.2.2 Descriptive Statistics . . . 31

3.2.3 Data Visualization . . . 32

3.2.4 Correlation Analysis . . . 33

3.2.5 Distribution of Scaled Revenue . . . 33

3.3 Data splitting . . . 34

3.4 Model building . . . 34

3.5 Improvement to a new model . . . 35

3.5.1 Addition of indicators . . . 36

3.5.2 Interaction terms . . . 38

3.5.3 Transformation of Average Meal Price . . . 39

3.5.4 Result model . . . 40

3.6 Hypothesis testing . . . 40

3.6.1 Model utility . . . 41

3.6.2 Error normality assumption . . . 41

3.6.3 Autocorrelation . . . 41

3.7 Validation and testing . . . 42

3.8 Model Meaning . . . 43

4 Question 2: Dataset 2 45

4.1 Data Preproccessing . . . 45

4.1.1 Missing data . . . 46

4.1.2 Dependent variables choosing . . . 46

4.1.3 Outlier . . . 46

4.2 Data overview . . . 46

4.2.1 Key Observations: . . . 46

4.3 Data deepdive . . . 47

4.4 Data splitting . . . 49

4.5 Building baseline model 49
4.5.1 Splitting data 52
4.6 Improving to a better model 52
4.6.1 Inspection of interaction terms 53
4.6.2 Adding the interaction terms into the model 53
4.6.3 The improved model 53
4.7 Hypothesis testing 55
4.7.1 Model utility 55
4.7.2 Error normality assumption 56
4.7.3 Homoskedastic residual variance 56
4.7.4 Autocorrelation 57
4.8 Validation and testing 57
4.9 Model Meaning 59

A Appendix **62**
A.1 Code 62
A.1.1 Question 1 62
A.1.2 Question 2: Dataset 1 69
A.1.3 Question 2: Dataset 2 73

List of Tables

1 Task Distribution . . . 1
2 Basic univariate summary statistics for the continuous quantitative variables . . . 2
3 Basic univariate summary statistics for model_year . . . 3
4 Basic univariate summary statistics for origin . . . 3
5 Basic univariate summary statistics for cylinders . . . 3
6 Summary of model parameter in R . . . 11
7 Summary of the second model parameter in R . . . 12
8 Summary of the baseline model parameter in R . . . 13
9 VIF Analysis . . . 14
10 VIF Analysis of the reduced model . . . 15
11 ANOVA result of the full and reduced model . . . 16
12 Evaluation parameter of M_1' and M_2' . . . 23
13 Evaluation parameter of full and reduced . . . 24
14 Shapiro-Wilk test result . . . 24
15 Levene test result . . . 25
16 Durbin-Watson test result . . . 25
17 Descriptive Statistics of the Dataset . . . 31
18 Restaurant model summary . . . 40
19 Shapiro-Wilk test result . . . 40
20 Shapiro-Wilk test result . . . 41
21 Durbin-Watson test result . . . 42
22 Summary of the new model . . . 54
23 Evaluation parameter of the baseline and improved model . . . 55
24 Shapiro-Wilk test result . . . 55
25 Shapiro-Wilk test result . . . 56
26 Levene test result . . . 56
27 Durbin-Watson test result . . . 57

List of Figures

1 Box plot of mpg . . . 2
2 Histogram of mpg . . . 4
3 Histogram of displacement . . . 6
4 Histogram of horsepower . . . 7
5 Histogram of weight . . . 8
6 Histogram of acceleration . . . 9
7 Barplot of model_year . . . 9
8 Origin barplot . . . 10
9 Histogram of the natural logarithm of mpg . . . 13
10 Box Cox plot of weight . . . 16
11 Histogram and boxplot of weight and log(weight) . . . 17
12 Histogram and boxplot of sqrt(weight) and $\frac{1}{sqrt(weight)}$ . . . 18
13 Box Cox plot of acceleration . . . 19
14 Histogram and boxplot of acceleration and log(acceleration) . . . 20
15 Histogram and boxplot of sqrt(acceleration) and $\frac{1}{sqrt(acceleration)}$ . . . 21
16 Prediction of log(mpg) . . . 26
17 Prediction of mpg . . . 27
18 Histogram of actual mpg . . . 27
19 Scatter Plot: Revenue vs. Average Meal Price . . . 32
20 Boxplot: Revenue by Cuisine Type . . . 32
21 Correlation Matrix . . . 33
22 Histogram: Distribution of Scaled Revenue (Revenue per Seat) . . . 34
23 Scatterplot: Revenue by SeatingCapacity*AverageMealPrice . . . 35
24 Scatterplot: Revenue by Reservation . . . 37
25 Histogram of the reservations . . . 38
26 Scatterplot: Revenue by SeatingCapacity grouped by Location and AverageMealPrice grouped by Cuisine . . . 39
27 Box Cox plot of Average Meal Price . . . 39
28 Box plot of registered . . . 46

29 Correlation Matrix 48
30 Relation between temp and atemp 49
31 Impact of Weekday 50
32 Impact of Workingday 50
33 Impact of holiday 51
34 Impact of temp 51
35 Plot of how registered depend on humidity 52
36 Scatter plot showing the interaction of indicators influencing the response 53

Table 1: Task Distribution

	(22125013)	(22125015)	(22125063)	(22125122)
Dataset Finding (10% Total)			5%	5%
Model Building (40% Total)	10%	20%	5%	5%
Report Writing & Research (40% Total)	15%		15%	10%
Validation + Proof Reading (10% Total)		5%		5%
Total	25%	25%	25%	25%

1 Task Distribution

2 Question 1

2.1 Data Preprocessing

We first import the data from the .csv file. Our data contains 398 observations, 9 columns including mpg, cylinders, displacement, horsepower, weight, acceleration, model_year, origin, car_name. Before we could use this data, we need to check for a few problem:

2.1.1 Missing data

There is some observation with horsepower column missing, this is denoted by the character ”?” in our data set. We need to remove these before we can continue working on this data set. After removing them, we are left with 392 observations.

2.1.2 Outlier

We then have to check for outlier data of the data set. Here is the box plot for mpg.

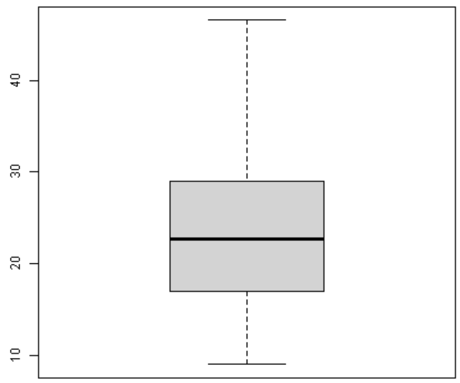


Figure 1: Box plot of mpg

So, there is no outlier in our data set. This can be validated by using the data in the summary of the data set in the next section, where all data point value deviate no more than 3 time the standard error from the mean.

2.2 Data overview

Here is the basic univariate summary statistics for our variables.

Table 2: Basic univariate summary statistics for the continuous quantitative variables

	N	Mean	SD	Min	Q1	Median	Q3	Max
mpg	392	23.45	7.81	9.00	17.00	22.75	29.00	46.60
displacement	392	194.41	104.64	68.00	105.00	151.00	284.50	455.00
horsepower	392	104.47	38.49	46.00	75.00	93.50	127.00	230.00
weight	392	2977.58	849.40	1613.00	2224.50	2803.50	3616.50	5140.00
acceleration	392	15.54	2.76	8.00	13.75	15.50	17.05	24.80

Table 3: Basic univariate summary statistics for model_year

	Level	N	%
model_year	70	29	7.4
	71	27	6.9
	72	28	7.1
	73	40	10.2
	74	26	6.6
	75	30	7.7
	76	34	8.7
	77	28	7.1
	78	36	9.2
	79	29	7.4
	80	27	6.9
	81	28	7.1
	82	30	7.7

Table 4: Basic univariate summary statistics for origin

	Level	N	%
origin	1	245	62.5
	2	68	17.3
	3	79	20.2

Table 5: Basic univariate summary statistics for cylinders

	Level	N	%
cylinders	3	4	1.0
	4	199	50.8
	5	3	0.8
	6	83	21.2
	8	103	26.3

2.3 Data splitting

The data is split into two part, training dataset and validation dataset. Training dataset contains 80% of the observations and the validation dataset contains 80% of the observations. The split is performed randomly with RStudio.
For replication, we use the seed 0.

2.4 Model building

2.4.1 Response

Fuel efficiency, calculated as miles per gallon (mpg), is the target output of our model. The mpg values for each car are positive real numbers. The following histogram illustrates the spread of fuel efficiency across our dataset.

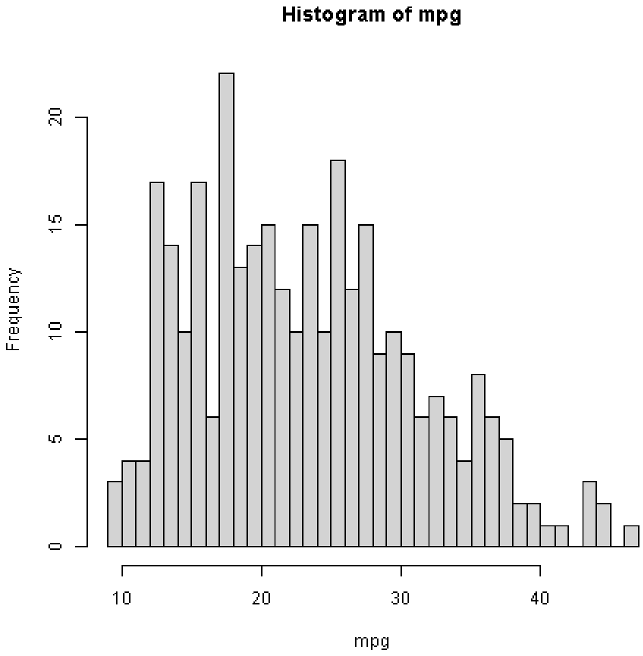
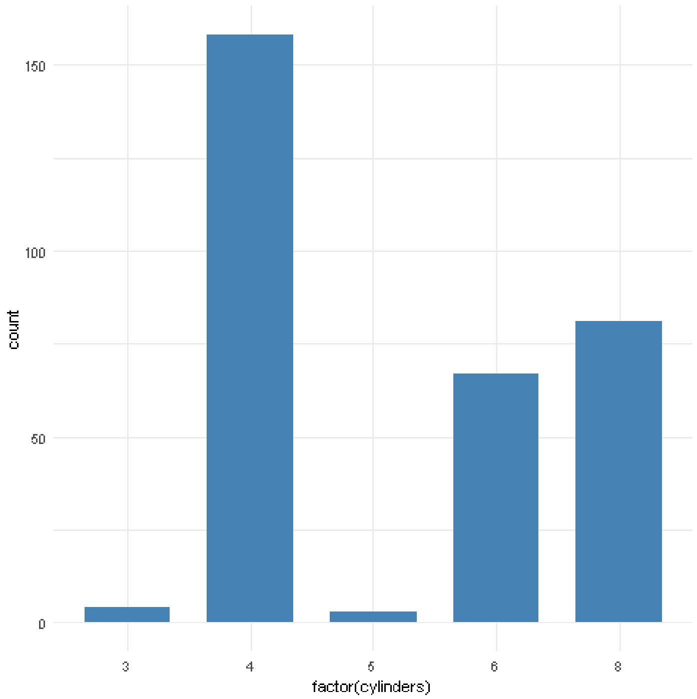


Figure 2: Histogram of mpg

As the histogram clearly shown, the data is skewed to the right. We will fix this in a later section.

2.4.2 Predictor

- Cylinders value for each car is a positive number which denote the number of cylinders in our car. Here is the bar plot for cylinders:



- Displacement is a continous variable in our dataset. It is the size of the engine of the car. Here is a histogram of it:

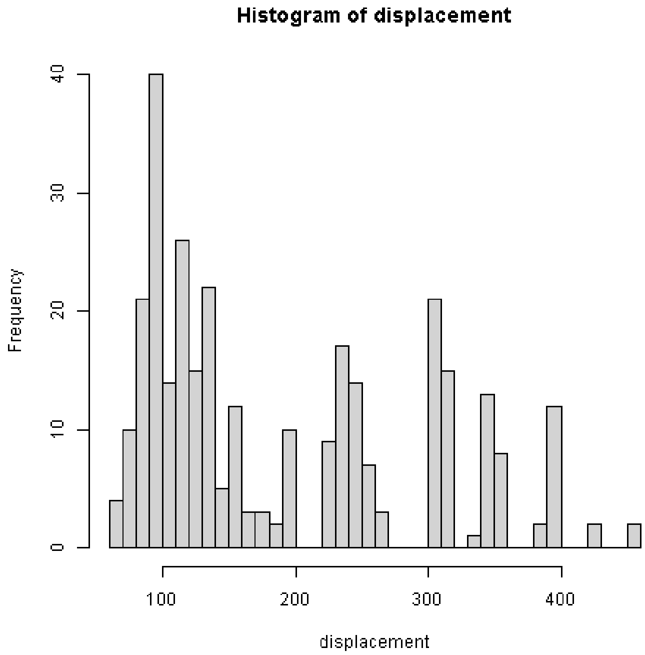


Figure 3: Histogram of displacement

- The engine's horsepower reflects how much power the car can produce. It is an important indicator of performance, particularly for sports and high-performance vehicles. Here is the histogram showing the distribution of horsepower across the dataset:

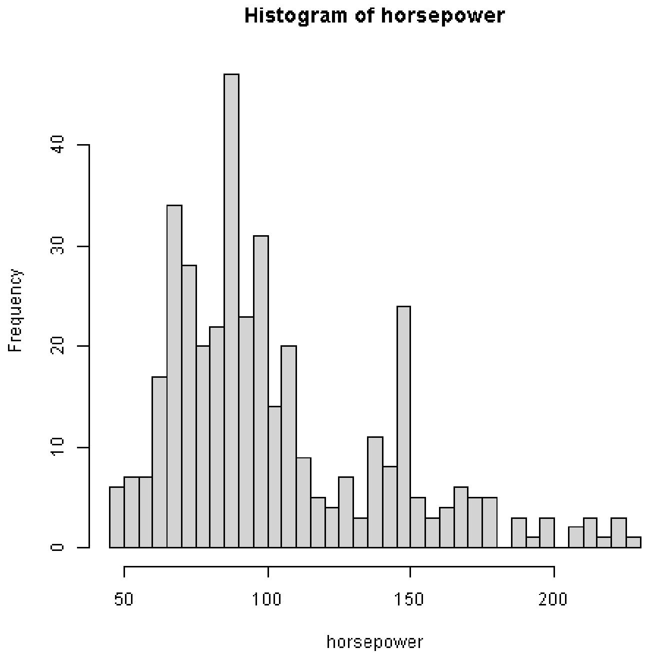


Figure 4: Histogram of horsepower

- Weight measures how heavy each car is. Heavier cars tend to have higher fuel consumption but can also offer more stability. Below is the histogram illustrating the distribution of car weights in our dataset:

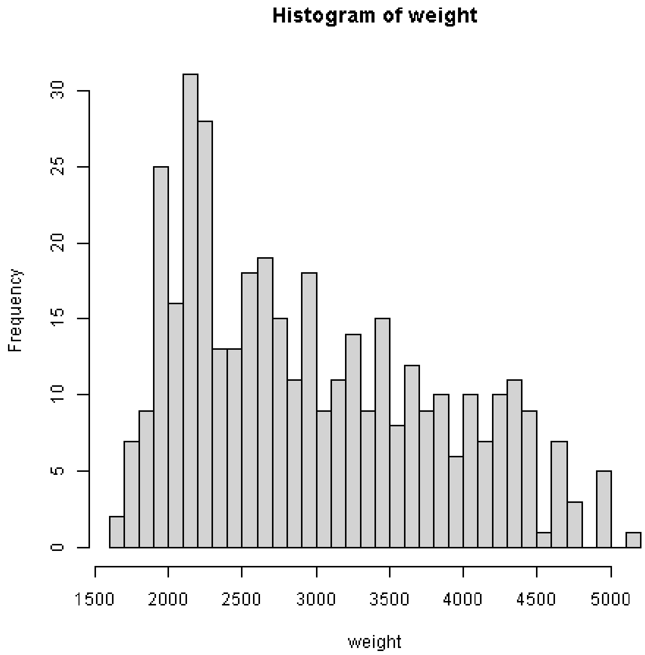


Figure 5: Histogram of weight

- Acceleration represents the time taken by the car to reach a certain speed from a standstill. Cars with better acceleration offer quicker response and can handle rapid speed changes more effectively. Here is a histogram of the acceleration values:

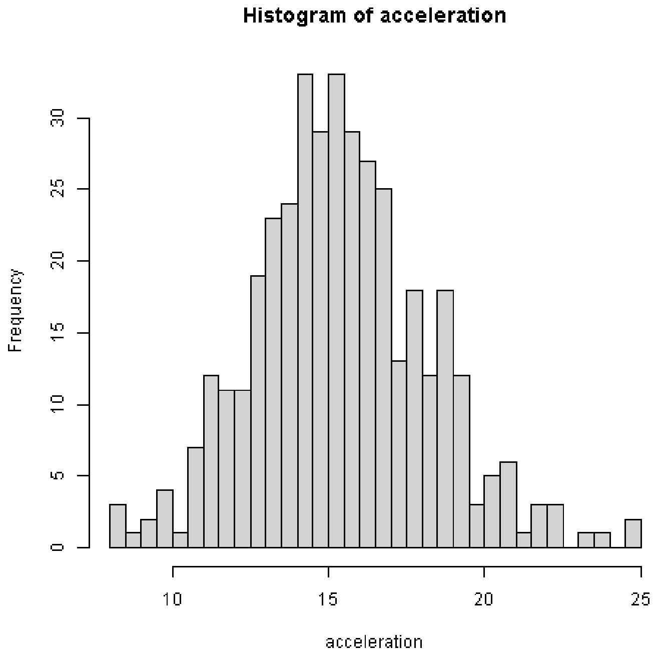


Figure 6: Histogram of acceleration

- The model year provides insights into when the car was manufactured. Over time, car designs and technologies have evolved, and the model year can reflect those changes. The bar plot below shows the distribution of model years in the dataset:

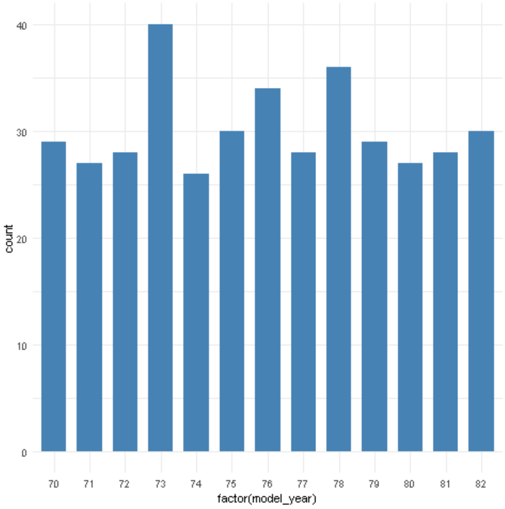


Figure 7: Barplot of model_year

- Origin indicates the geographical region where the car was manufactured. This dataset categorizes cars by their origins as North America (1), Europe (2), and Asia (3). The bar plot below shows how the cars are distributed across these regions:

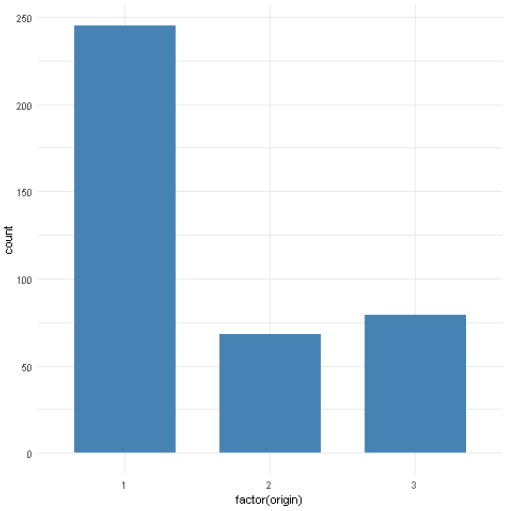


Figure 8: Origin barplot

2.4.3 Model selection method

We use the stepwise method as it is a simple, easy to implement and effective method for model selection. It combines both forward and backward selection, adding and removing predictors iteratively until no further improvement is observed.

2.4.4 Evaluation parameter

The model built should be evaluate using the following parameter:

- R^2: It helps us assess the overall fit of your model. A higher R-squared generally implies a better fit
- Adjusted R^2: As we add more independent variables to a model, R-squared will always increase or stay the same, even if the added variables are not significant. Adjusted R-squared accounts for this by penalizing the model for adding unnecessary variables
- AIC and BIC, AIC is mainly use to perform stepwise procedure. These parameter is similar to adjusted R^2 as they imposed a penalty on complex model.
- Sharpio-Wilk test p-value, this is used to determine whether a sample of data comes from a normally distributed population. This is crucial if we want to do ANOVA or t-test with our result model.

- Durbin-Waston test value, this determine whether the residuals of a regression model are correlated with each other. This violates one of the key assumptions of linear regression, which is that the errors are independent as this may lead to inaccuracy in our model.
- Levene test p-value, this check the assumption of homogeneity of variance. This assumption is crucial for many statistical tests, particularly ANOVA and t-tests.

2.4.5 Baseline model

Our first model is just taking the linear model of mpg as response and all other 7 variables as linear predictor.

So our first model will take the forms of:

$$mpg \;=\; \beta_1 \;+\; \beta_2\, cylinders \;+\; \beta_3\, displacement \;+\; \beta_4\, horsepower \;+\; \beta_5\, weight \;+ \\ \beta_6\, acceleration \;+\; \beta_7\, model\_year \;+\; \beta_8\, origin \;+\; \epsilon$$

We run this model in R and obtain the result:

Table 6: Summary of model parameter in R

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-19.5296	5.2529	-3.72	0.0002
cylinders	-0.1863	0.3794	-19	0.6237
displacement	0.0125	0.0089	1.40	0.1612
horsepower	-0.0167	0.0157	-1.06	0.2889
weight	-0.0063	0.0007	-8.48	0.0000
acceleration	0.1274	0.1123	1.13	0.2573
model_year	0.7652	0.0585	13.08	0.0000
origin	1.3138	0.3217	4.08	0.0001

Residual standard error: 3.454 on 305 degrees of freedom
Multiple R-squared: 0.8131, Adjusted R-squared: 0.8088
F-statistic: 189.5 on 7 and 305 DF, p-value: ¡ 2.2e-16

There is a few change we can make to our model:

- cylinders: It's p-value is very high, and it's likely that it does not have a statistical significance to our model.
- Displacement: It's p-value is very high, and it's likely that it does not have a statistical significance to our model.

- Horsepower: It's p-value is very high, and it's likely that it does not have a statistical significance to our model.

- Acceleration: It's p-value is very high, and it's likely that it does not have a statistical significance to our model.

- Origin: It represent the production region of the car (Asia, North America, Europe) as such, it reasonable for it to not have a linear relation to our model.

- Model_year: It represent the year that the car is manufactured. While it's reasonable for car manufactured technique to become better, the change may not be linear. [1]

Firstly, we can make indicator variable for origin, the resulting model will be:

$$mpg \ = \ \beta_1 \ + \ \beta_2 \ cylinders \ + \ \beta_3 \ displacement \ + \ \beta_4 \ horsepower \ + \ \beta_5 \ weight \ + \ \beta_6 \ acceleration \ + \ \beta_7 \ model\_year \ + \ \beta_8 \ origin_2 \ + \ \beta_9 \ origin_3 \ + \ \epsilon$$

Other variable with little statistical significance will be removed later after we perform the stepwise procedure, so no need to removing them now.

Putting this model through R provide us with the following result:

Table 7: Summary of the second model parameter in R

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-21.0709	5.2560	-4.01	0.0001
cylinders	-0.2000	0.3751	-0.53	0.5943
displacement	0.0170	0.0089	1.91	0.0573
horsepower	-0.0166	0.0155	-1.07	0.2850
weight	-0.0066	0.0007	-8.84	0.0000
acceleration	0.1216	0.1110	1.10	0.2740
model_year	0.7990	0.0590	13.54	0.0000
origin2	2.9302	0.6493	4.51	0.0000
origin3	2.5944	0.6360	4.08	0.0001

Residual standard error: 3.414 on 304 degrees of freedom
Multiple R-squared: 0.818, Adjusted R-squared: 0.8132
F-statistic: 170.7 on 8 and 304 DF, p-value: ¡ 2.2e-16

Then, we perform the stepwise regression technique on this model to obtain our baseline:

[1] We do not try to evaluate this variable as a qualitative variable, even though it may improved the accuracy of our model. We will explain this at a later chapter

Table 8: Summary of the baseline model parameter in R

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-23.7380	4.7118	-5.04	0.0000
displacement	0.0120	0.0063	1.89	0.0594
weight	-0.0070	0.0007	-117	0.0000
acceleration	0.1902	0.0889	2.14	0.0333
model_year	0.8109	0.0578	14.04	0.0000
origin2	2.8420	0.6442	4.41	0.0000
origin3	2.4230	0.6180	3.92	0.0001

Residual standard error: 3.411 on 306 degrees of freedom
Multiple R-squared: 0.8172, Adjusted R-squared: 0.8136
F-statistic: 227.9 on 6 and 306 DF, p-value: ¡ 2.2e-16

So our baseline model will have the form:

$$mpg =$$

$$\beta_1 + \beta_2\, displacement + \beta_3\, weight + \beta_4\, acceleration + \beta_5\, model\_year + \beta_6\, origin_2 + \beta_7\, origin_3 + \epsilon$$

2.5 Improving to a better model

2.5.1 Fixing mpg skewness

As mention before, mpg is currently skewing to the left.

We can fix this by taking the log of mpg:

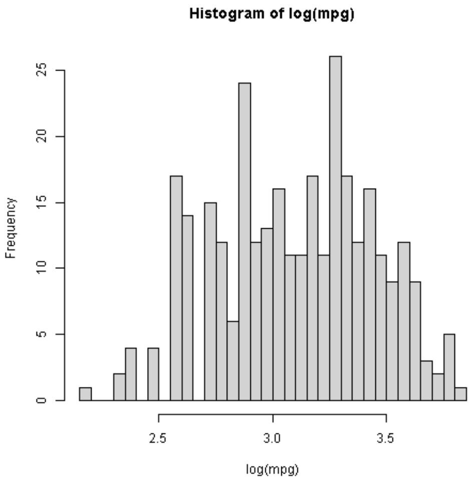


Figure 9: Histogram of the natural logarithm of mpg

2.5.2 Multicollinearity

We still haven't discussed the possibility of multicollinearity in our model, for good reasons. Some of our variable can be relate to each other, in a very natural way as they are describing the characteristics of a car. For examples, we can make some guess about their relation:

- Displacement and weight: The engine make up a significant portion of a car's weight. As such it's reasonable to assume a car with higher weight tend to have a larger engine.
- Weight and acceleration: A car that's heavier is harder to accelerate.

As such, we need to consider the possibility of multicollinearity in our model.

To ensure the model is properly specified and functioning correctly, there are tests that can be run for multicollinearity. The variance inflation factor is one such measuring tool. Using variance inflation factors helps to identify the severity of any multicollinearity issues so that the model can be adjusted.

Variable	GVIF	Df	GVIF^(1/(2*Df))
displacement	10.70	1.00	3.27
weight	8.04	1.00	2.83
acceleration	1.51	1.00	1.23
model_year	1.21	1.00	1.10
origin	1.93	2.00	1.18

Table 9: VIF Analysis

Interpretation of GVIF Output:

- **GVIF**: Generalized VIF adjusts the VIF for degrees of freedom. The larger the GVIF, the more multicollinearity there is in the variable.
- **GVIF$\hat{(}$1/(2*Df))**: This column normalizes GVIF, making it comparable to standard VIF values when Df (degrees of freedom) is greater than 1. This normalized value should be used for interpretation.

Common Thresholds:

- **GVIF$\hat{(}$1/(2*Df)) ¿** $\sqrt{(}5)$: Suggests moderate multicollinearity.

- **GVIF$\hat{(}$1/(2*Df)) ¿** $\sqrt{(}$10): Indicates high multicollinearity and might require corrective actions like removing or combining variables.

We can see at a glance that displacement have a really high value GVIF$\hat{(}$1/(2*Df)) ¿ $\sqrt{(}$10), this indicate a severe multicollinearity issue in our model. While we still haven't removed this variable yet. This is an important information for later.
Note: Displacement is going to be removed eventually with the stepwise method. This will solve our multicollinearity problem. We can see this through the VIF analysis of the reduced model without displacement.

Table 10: VIF Analysis of the reduced model

	GVIF	Df	GVIF^(1/(2*Df))
weight	1.78	1.00	1.34
acceleration	1.25	1.00	1.12
model_year	1.18	1.00	1.09
origin	1.60	2.00	1.13

So even though we didn't removed displacement now, after we arrived with our improved model, as displacement is no longer included in our final model. Multicollinearity is no longer an issue for our model.

2.5.3 Displacement statistical significance issues

As we can see from the summary of our baseline model, displacement may not have a large statistical significance to our model. In addition, Displacement also has a relative high GVIF value. As such, it can be beneficial to try to remove displacement from our model.
This mean we are testing the hypothesis

$$H_0 : \beta_2 = 0$$
$$H_1 : \beta_2 \neq 0$$

In our baseline model:

$$mpg =$$
$$\beta_1 + \beta_2\, displacement + \beta_3\, weight + \beta_4\, acceleration + \beta_5\, model\_year + \beta_6\, origin_2 + \beta_7\, origin_3 + \epsilon$$

To do this we can do a partial F-test on the baseline model and the reduced model:

$$mpg = \beta_1 + \beta_3\ weight + \beta_4\ acceleration + \beta_5\ model\_year + \beta_6\ origin_2 + \beta_7\ origin_3 + \epsilon$$

The test statistics value we are trying to calculate is

$$f = \frac{(SSE_l - SSE_k)}{k-l} \frac{SSE_k}{n-(k+1)}$$

With $k = 6$ (number of predictors), $l = 1$ (number of removed predictor), n = number of observations in our dataset. SSE_l, SSE_k is the unexplained variation for the reduced and full model, respectively. This calculation can be done easily through ANOVA function in R.

Table 11: ANOVA result of the full and reduced model

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	306	3559.85				
2	307	3601.52	-1	-41.67	3.58	0.0594

The table give us an $f = 3.58$ and corresponding to a p-value of 0.0594.
This mean that we rejected H_0 and conclude that displacement has a statistical significance in our dataset and we cannot removed displacement yet. However, this statistical significance is still very small and only barely make the threshold.

2.5.4 Transformation of weight

We examine the Box Cox plot of weight:

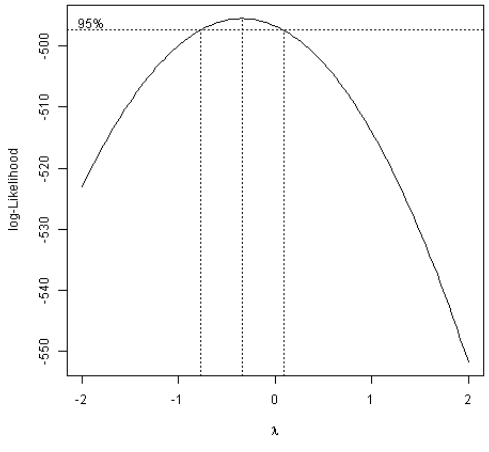


Figure 10: Box Cox plot of weight

The Box Cox plot show us a few way to transform weight.

A simple solution is to take the log of weight. As $\lambda = 0$ is in the range of the confidence interval.

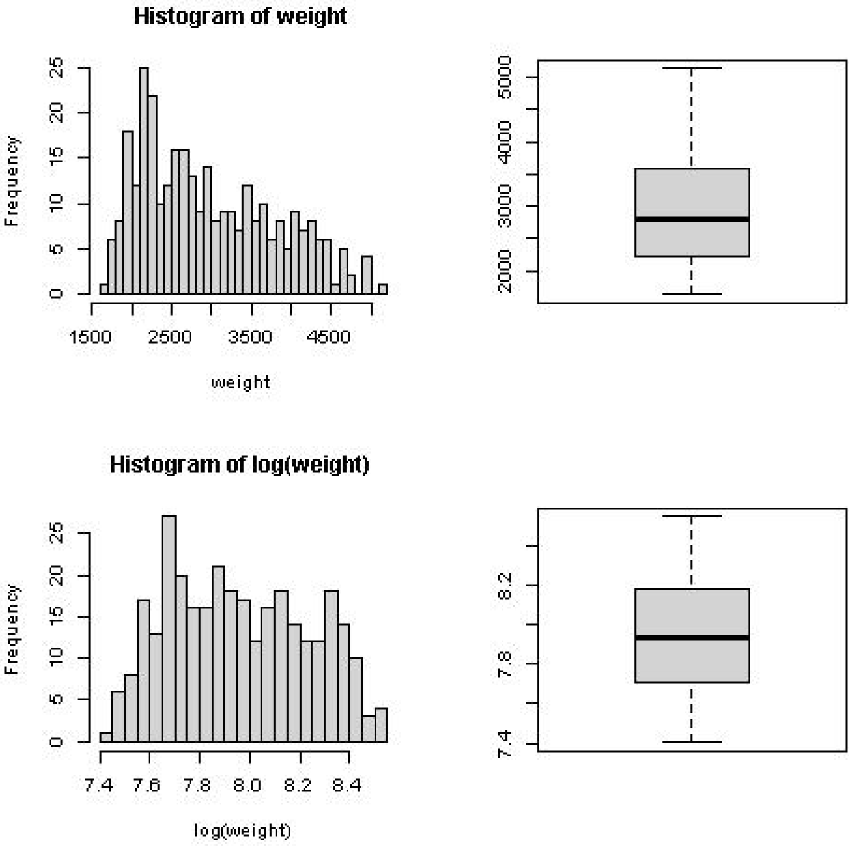


Figure 11: Histogram and boxplot of weight and log(weight)

Another possible path is to take the inverse square root of weight, or $\frac{1}{sqrt(weight)}$ as $\lambda = -0.5$ is also in the interval, or $sqrt(weight)$ as $\lambda = 0.5$ is also pretty close to the confidence inteval.

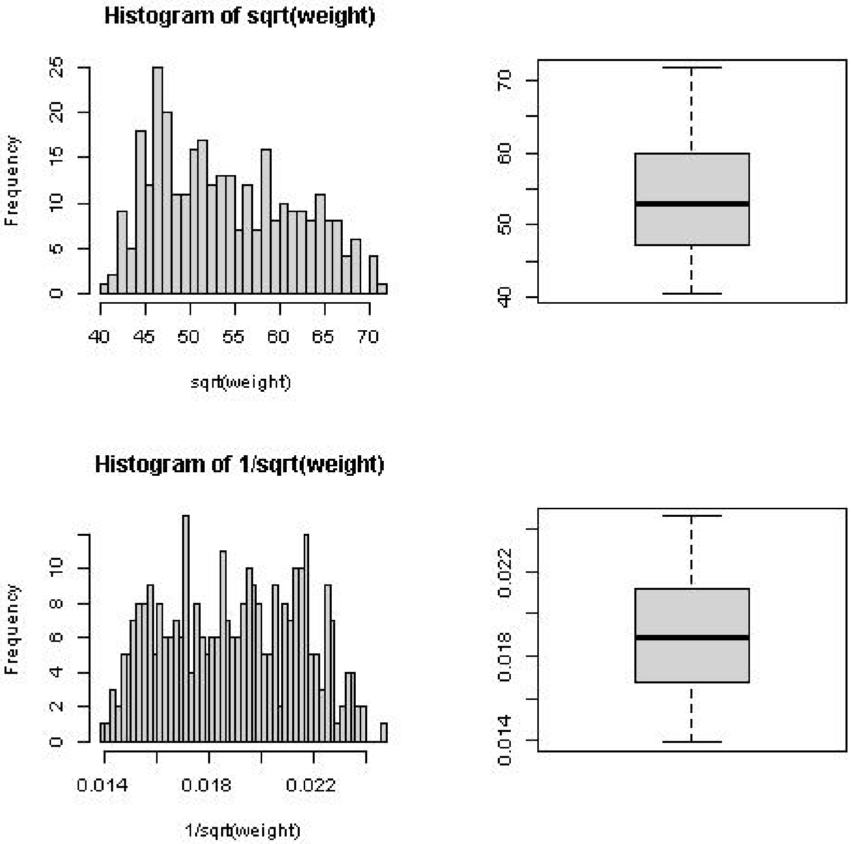


Figure 12: Histogram and boxplot of sqrt(weight) and $\frac{1}{sqrt(weight)}$

We decide to pick log(weight) as it is both simple to implement and having a histogram that pretty close to normal distribution.

2.5.5 Transformation of acceleration

We do the same thing as the previous section. We examine the Box Cox plot of acceleration:

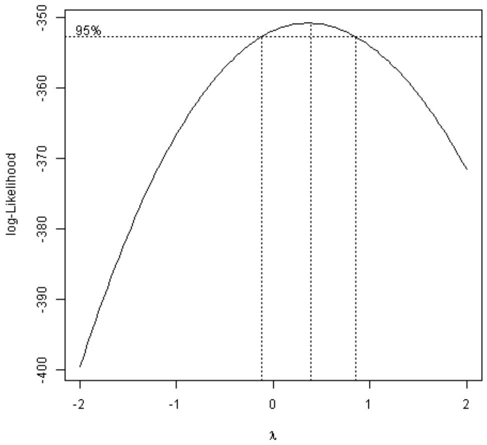


Figure 13: Box Cox plot of acceleration

The Box Cox plot show us a few way to transform acceleration.

A simple solution is to take the log of acceleration. As $\lambda = 0$ is in the range of the confidence interval.

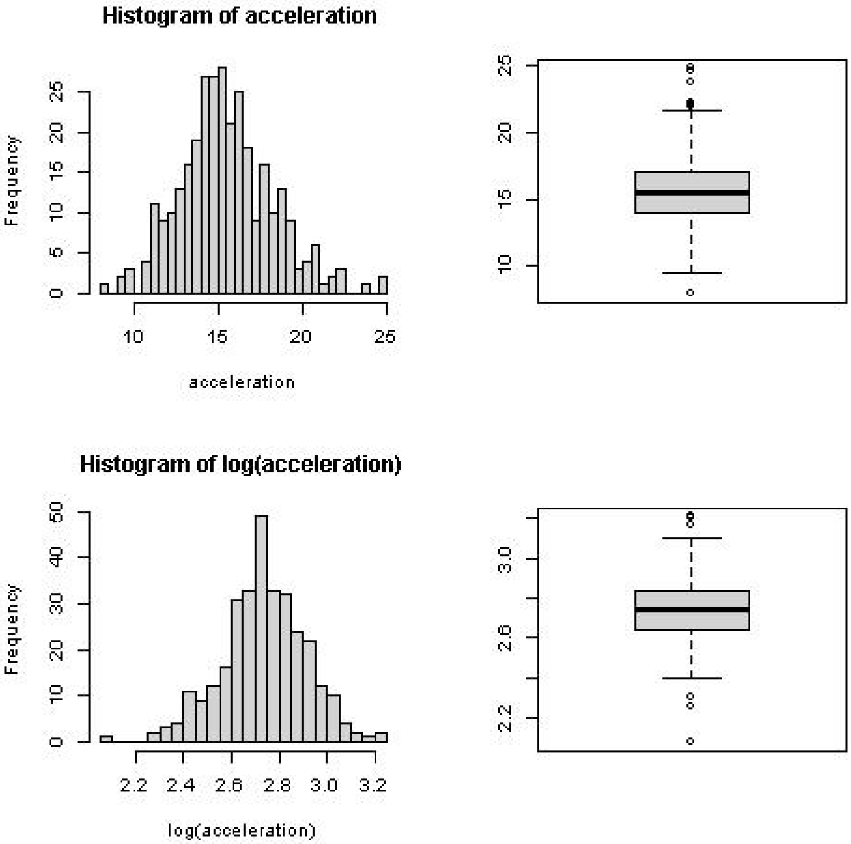


Figure 14: Histogram and boxplot of acceleration and log(acceleration)

Another possible path is to take the inverse sqrt of acceleration, or $\frac{1}{sqrt(acceleration)}$ as $\lambda = -0.5$ is also pretty close to the interval, or sqrt(acceleration) as $\lambda = 0.5$ is in the confidence inteval.

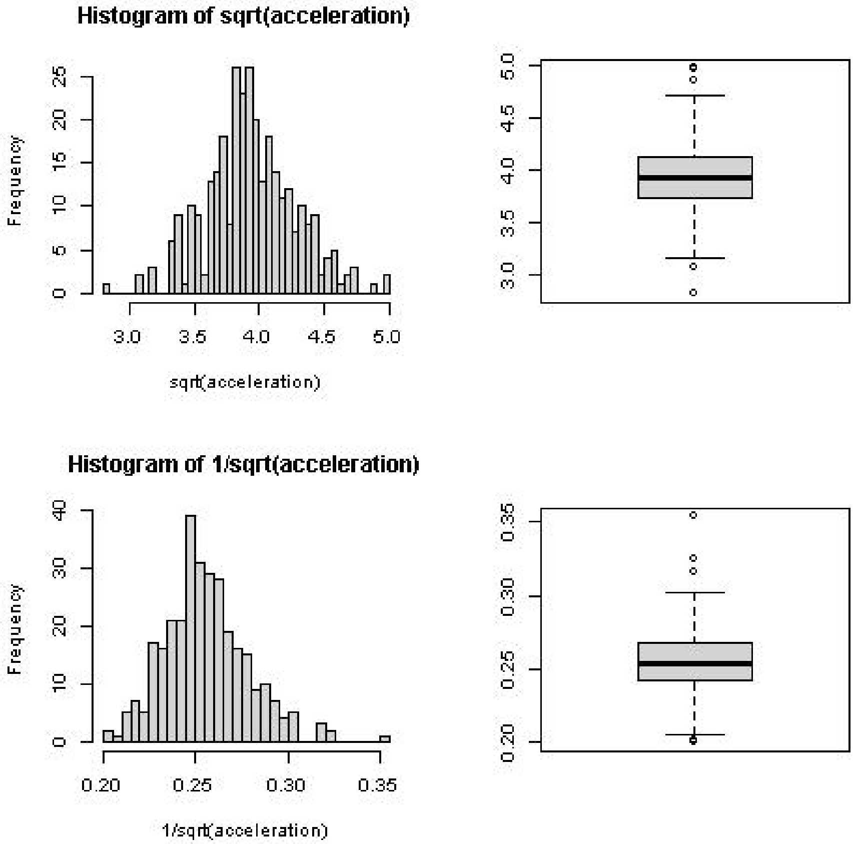


Figure 15: Histogram and boxplot of sqrt(acceleration) and $\frac{1}{sqrt(acceleration)}$

We decide to test both acceleration and log(acceleration).

2.5.6 Model year as multi-value discrete variable vs. continuous variable

Here, we are having to make an important decision. Making model year as an multi-value discrete variable or a continuous variable. While model year only take integer value from 70 to 78. We decide to make model year a continuous variable after weighting the pros and cons:
Advantages of making model_year a multi-value discrete variable:

- We can make indicator variable for each year, or each group of year. Improving the accuracy of our model.

Disadvantages of making model_year a multi-value discrete variable:

- Harder to implement, more variable to consider.

- Can only be used to estimate car made from 1970 to 1978.

Advantages of making model_year a continuous variable:

- Easier to implement.
- More general, can be use to predict future value, beyond the value of data set provided.

Disadvantages of making model_year a continuous variable:

- Less accurate, as the rate of change between year may not be linear.

In the end, we pick making model_year a continuous variable for generalibilty.

2.5.7 The improved model

From various adjustment from the previous sections, we are now left with two model M_1 and M_2:

$$M_1: \; log(mpg) \; = \; \beta_1 \; + \; \beta_2 \, displacement \; + \; \beta_3 \, log(weight) \; + \; \beta_4 \, acceleration \; + \\ \beta_5 \, model\_year \; + \; \beta_6 \, origin_2 \; + \; \beta_7 \, origin_3 \; + \; \epsilon$$

$$M_2: \; log(mpg) \; = \; \beta_1 \; + \; \beta_2 \, displacement \; + \; \beta_3 \, log(weight) \; + \; \beta_4 \, log(acceleration) \; + \\ \beta_5 \, model\_year \; + \; \beta_6 \, origin_2 \; + \; \beta_7 \, origin_3 \; + \; \epsilon$$

We run the stepwise procedure again on these model and obtain our result, let's called these result M_1' and M_2'.

$$M_1': \; log(mpg) \; = \\ \beta_1 \; + \; \beta_3 \, log(weight) \; + \; \beta_4 \, acceleration \; + \; \beta_5 \, model\_year \; + \; \beta_6 \, origin_2 \; + \; \beta_7 \, origin_3 \; + \; \epsilon$$

$$M_2': \; log(mpg) \; = \\ \beta_1 \; + \; \beta_3 \, log(weight) \; + \; \beta_4 \, log(acceleration) \; + \; \beta_5 \, model\_year \; + \; \beta_6 \, origin_2 \; + \; \beta_7 \, origin_3 \; + \; \epsilon$$

So both come to the conclusion to exclude displacement from the model, this is expected as from the previous section, displacement is a candidate for multicollinearity and does not have a large statistical significance to our model.

We also get the evaluation parameter of these two model.

Table 12: Evaluation parameter of M_1' and M_2'

	M_1'	M_2'
R^2	**0.8842**	0.8837
Adjusted R^2	**0.8823**	0.8818
AIC	**-443.8934**	-442.4572
BIC	**-417.67**	-416.2338

This show that M_1' is the superior model, as such we take M_1' as our final model.

2.6 Hypothesis testing

Before proceeding, we rewrite our model a little bit:

$$M : \ log(mpg) \ =$$
$$\beta_1 \ + \ \beta_2 \ log(weight) \ + \ \beta_3 \ acceleration \ + \ \beta_4 \ model\_year \ + \ \beta_5 \ origin_2 \ + \ \beta_6 \ origin_3 \ + \ \epsilon$$

2.6.1 Model utility

We are testing the hypothesis:

$$H_0 : \beta_i = 0, i \in \{2,3,4,5,6\}$$
$$\text{H}_1 : \text{At least one of } \beta_i \neq 0, i \in \{2,3,4,5,6\}$$

This mean we are calculating the test statistic value $f = \frac{R^2/k}{1-R^2} \frac{1}{(n-(k+1)}$.
Plugging in the number we get $f = 468.8$, this corresponding to a p-value of $2, 2.10^{-16}$. This mean we can be certain to reject H_0. And conclude that our model contains useful linear relationship between our the response mpg and at least one of the variable.

2.6.2 Predictor statistical significance

This mean, for each $i \in \{2,3,4,5,6\}$. We are testing the hypothesis.

$$H_0 : \beta_i = 0$$
$$H_1 : \beta_i \neq 0$$

To do this we can do a partial F-test on the reduced model and our full model. This is similar to the method describe in a previous section.

	f	P-value
log(weight)	-27.00	0.0000
acceleration	2.37	0.0182
model_year	17.33	0.0000
origin2	3.33	0.0010
origin3	1.54	0.1253

Again, this can be easily done with the ANOVA function in R. Here are the result for each partial F-test for each variables:

The only variable that failed this test is the origin3, or that's mean we failed to reject the hypothesis that $\beta_6 = 0$. However, let's consider the evaluation parameter of the reduced model (the model without origin 3:

Table 13: Evaluation parameter of full and reduced

	Full model	Reduced model
R^2	**0.8842**	0.8837
Adjusted R^2	**0.8823**	0.8818
AIC	**-443.8934**	-442.4572
BIC	**-417.67**	-416.2338

We can see that removing origin3 does not neccessary improve our model. In addition, removing origin3 mean that the user need to do an additional layer of preprocessing when using our model. Hence, we decided to keep origin3.

2.6.3 Error normality assumption

We are testing the hypothesis of:

$$H_0 : \epsilon \text{ is normal distributed}$$
$$H_1 : \epsilon \text{ is not normal distributed}$$

To test this, we can use the test data. Let's predict_mpg be our result after passing the test observation through the model. Let's mpg be the corresponding value of that observation. Then, $\epsilon = predict\_mpg - mpg$. We will run the Shapiro-Wilk test through this resulting error.

Here the result of our Shapiro-Wilk test:

Table 14: Shapiro-Wilk test result

W	p-value
0.97088	0.06765

This mean we have failed to reject the hypothesis that ϵ is normal distributed.

2.6.4 Homoskedastic residual variance

Here we are testing the hypothesis of:

$$H_0 : \text{The residuals variances of the groups are equal.}$$
$$H_1 : \text{At least one group has a different residuals variance.}$$

To test this, we can run the Levene's test, which is available in R. After running the test, we get the result:

Table 15: Levene test result

F	p-value
2.7299	0.0668

This mean we have failed to reject the null hypothesis. This indicate that the residuals variances of the groups are equal.

2.6.5 Autocorrelation

We are testing the hypothesis:

$$H_0 : \text{There is no autocorrelation in the error terms of the linear regression model.}$$
$$H_1 : \text{There is positive autocorrelation in the error terms of the linear regression model.}$$

To test this, we can run the Durbin-Watson test, which is available in R. After running the test, we get the result:

Table 16: Durbin-Watson test result

D-W	p-value
1.950639	0.668

This mean we have failed to reject the null hypothesis. This indicate that there is no autocorrelation in the error terms of the linear regression model.

2.7 Validation and testing

We used the same definition of predict_mpg as the previous section. It's the result of passing the test data through our model.

Here is the histogram of predict_log(mpg):

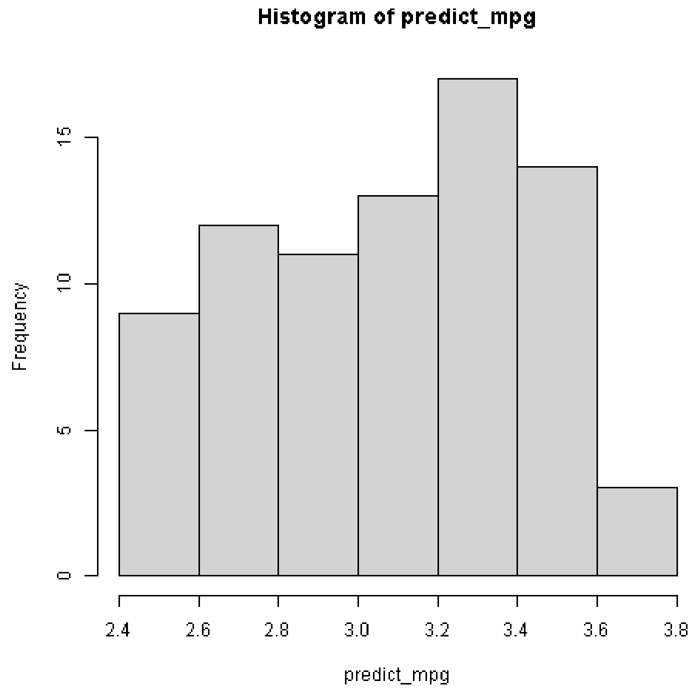


Figure 16: Prediction of log(mpg)

As our model estimate log(mpg), we have to take this to the power of e in order to be comparable to mpg. After taking predict_mpg to the power of e. Here is the histogram of it:

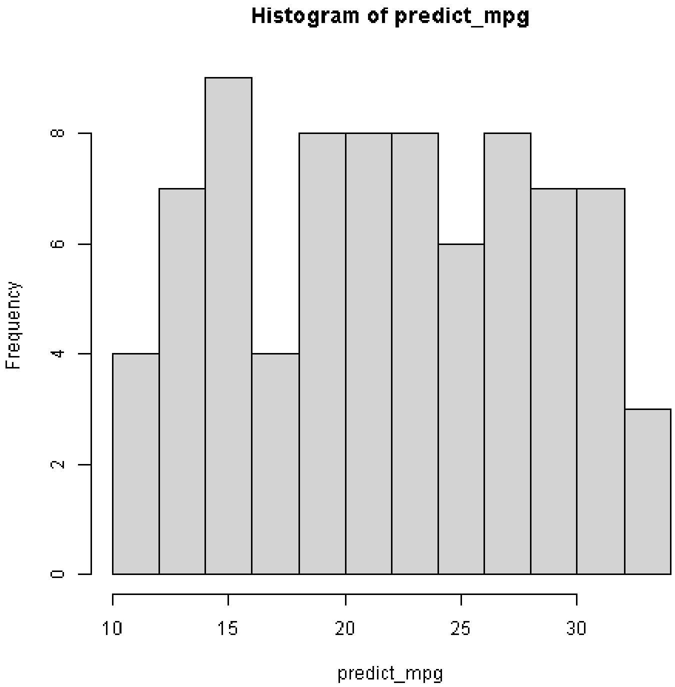


Figure 17: Prediction of mpg

We compare this to the actual value of mpg, here is the scatter plot of them:

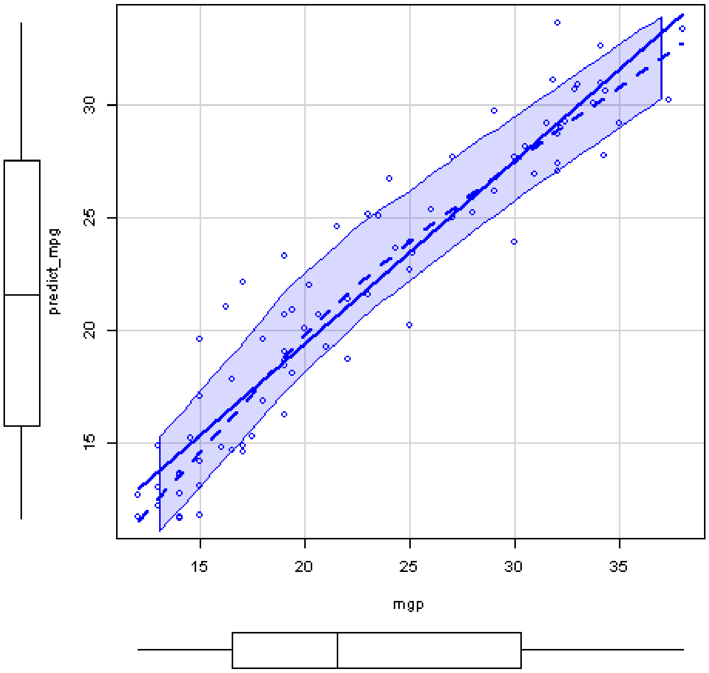


Figure 18: Histogram of actual mpg

As they form a almost 45 degree line in the diagonal of the graph, we can say that our model have predict the value of mpg pretty closely.

2.8 Model meaning

Our model is capable of estimating the fuel efficiency of car model with 4 parameter: weight, acceleration, model_year and origin. As model_ year is used as a continuous variable, this model can also be used for future car model, though in that case, we may need to reexamine the accuracy of the model.
Overall, Here is our final model:

$$M : log(mpg) =$$
$$\beta_1 + \beta_2\, log(weight) + \beta_3\, acceleration + \beta_4\, model\_year + \beta_5\, origin_2 + \beta_6\, origin_3 + \epsilon$$

With:

$$\beta_1 = 7.375331\ \beta_2 = -0.874849\ \beta_3 = 0.006518\ \beta_4 = 0.033764\ \beta_5 = 0.069592\ \beta_6 = 0.031755$$

Variable meaning:

- mpg: Fuel efficiency, calculated as miles per gallon (mpg)
- weight: Car weight (lbs)
- acceleration: Car's acceleration (ft/s^2)
- model_year: Number of year after 1900 that the car is manufactured.
- origin2: 1 if manufactured in Europe, 0 otherwise.
- origin3: 1 if manufactured in Asia, 0 otherwise.

This mean that:

- The initial log(mpg) of a car is 7.375331. Which equate to 228.5 miles per gallon.
- When log(weight) increase by 1, the log(mpg) decrease by -0.874849.
- When acceleration increase by 1 unit, log(mpg) increase by 0.006518

- For a car that is built one year later than another, it's log(mpg) is increase by 0.033764.
- A car built at Europe has its log(mpg) 0.069592 unit higher than a car built at North America.
- A car built at Asia has its log(mpg) 0.031755 unit higher than a car built at North America.
- This assume all other variable is the same.

In layman's terms, this mean:

- The heavier the car, the less fuel efficient it is.
- The faster the car can accelerate, the more fuel efficient it is.
- Car are more fuel efficient in the future than in the past.
- Car built at Europe is more fuel efficient than the same car built at Asia, which is also more fuel efficient than the same car built at North America.

Please note these are infer from the data set and may not truly represent the real world. However, in the scope of our data set, our model have accurately estimate the fuel efficiency of the car, given the data. It also show potential to be used to estimate cars which are made beyond 1978, which is the limit of the data set.

3 Question 2 - Dataset 1

3.1 Introduction

This report analyzes a dataset containing various features related to restaurants and their revenue. The goal is to explore relationships between different variables and the revenue, and to generate insights that may help in predicting future revenue for restaurants.

3.2 Dataset Overview

The dataset contains 8,368 records of restaurants with the following key features:

- **Name**: Name of the restaurant.
- **Location**: The location of the restaurant (Downtown, Rural, Suburban).
- **Cuisine**: The type of cuisine offered by the restaurant (e.g., American, Italian, Japanese).
- **Rating**: Average rating of the restaurant (scale of 1 to 5).
- **Seating Capacity**: The number of seats available at the restaurant.
- **Average Meal Price**: The average price of a meal.
- **Marketing Budget**: The marketing budget of the restaurant.
- **Social Media Followers**: Number of social media followers.
- **Chef Experience**: The number of years of experience the chef has.
- **Number of Reviews**: Total number of reviews received.
- **Average Review Length**: The average length of the reviews.
- **Ambience Score**: Score reflecting the ambience of the restaurant (scale of 1 to 10).
- **Service Quality Score**: Score reflecting the quality of service (scale of 1 to 10).
- **Parking Availability**: Whether parking is available (Yes/No).
- **Weekend Reservations**: Number of reservations during weekends.

- **Weekday Reservations**: Number of reservations during weekdays.
- **Revenue**: The total revenue generated by the restaurant.

3.2.1 Data Cleaning

The dataset had no missing values, so the main focus was on transforming categorical variables like *Location*, *Cuisine*, and *Parking Availability* into factors for analysis. Imputations for missing values would involve replacing them with median values if necessary.

3.2.2 Descriptive Statistics

Table 17 provides a summary of key statistics for the numeric variables in the dataset.

Table 17: Descriptive Statistics of the Dataset

Variable	Mean	Median	Min	Max	SD
Revenue (in $)	656,070	604,242	184,709	1,531,868	267,414
Rating	4.01	4.00	3.00	5.00	0.50
SeatingCapacity	60.21	60.00	30.00	90.00	16.57
AverageMealPrice (in $)	47.90	45.53	25.00	76.00	13.86
MarketingBudget (in $)	3,218	2,846	604	9,978	1,673
SocialMediaFollowers	36,191	32,519	5,277	103,777	19,830
ChefExperience (Years)	10.05	10.00	1	19	5.47
NumberofReviews	523	528	50	999	275

3.2.3 Data Visualization

Revenue vs. Average Meal Price The relationship between revenue and average meal price is visualized in Figure 19. A positive linear trend can be observed, suggesting that restaurants with higher average meal prices tend to generate more revenue.

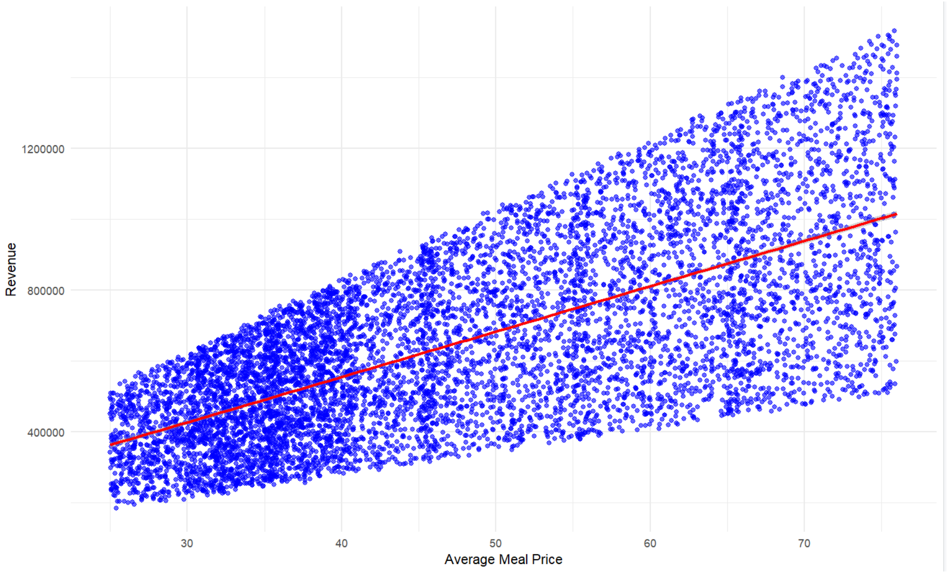


Figure 19: Scatter Plot: Revenue vs. Average Meal Price

Revenue by Cuisine Type Figure 20 shows the distribution of revenue across different cuisine types. Certain cuisines, such as Japanese and French, tend to have higher median revenues.

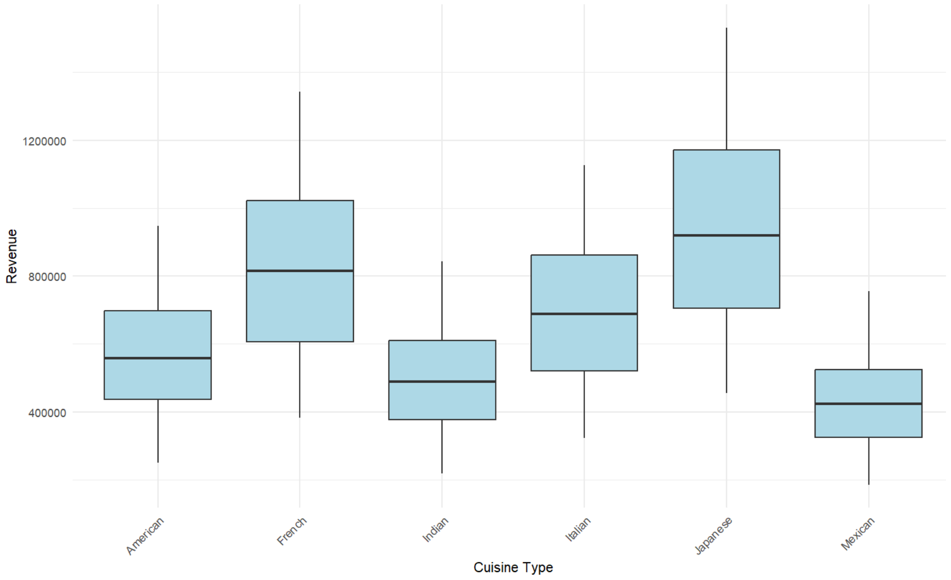


Figure 20: Boxplot: Revenue by Cuisine Type

3.2.4 Correlation Analysis

A correlation analysis was conducted to identify relationships between numeric variables. Figure 21 presents a correlation heatmap that highlights strong correlations, such as between seating capacity and revenue, and between average meal price and revenue.

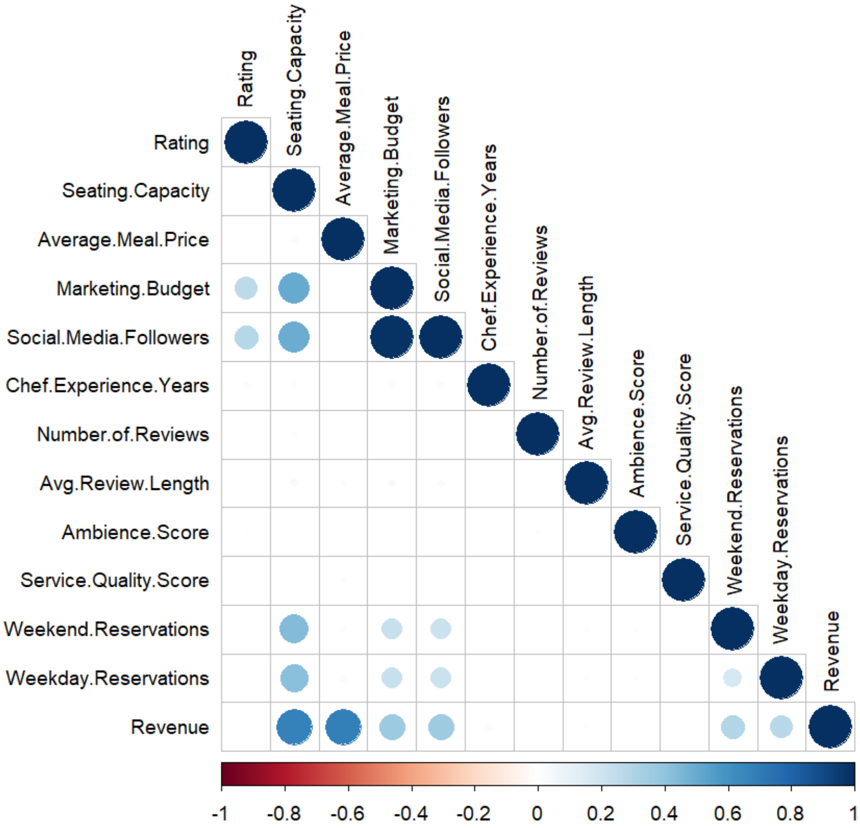


Figure 21: Correlation Matrix

3.2.5 Distribution of Scaled Revenue

The distribution of scaled revenue (revenue divided by seating capacity) is visualized in Figure 4.8. This metric helps normalize revenue across restaurants of different sizes, providing a more equitable comparison.

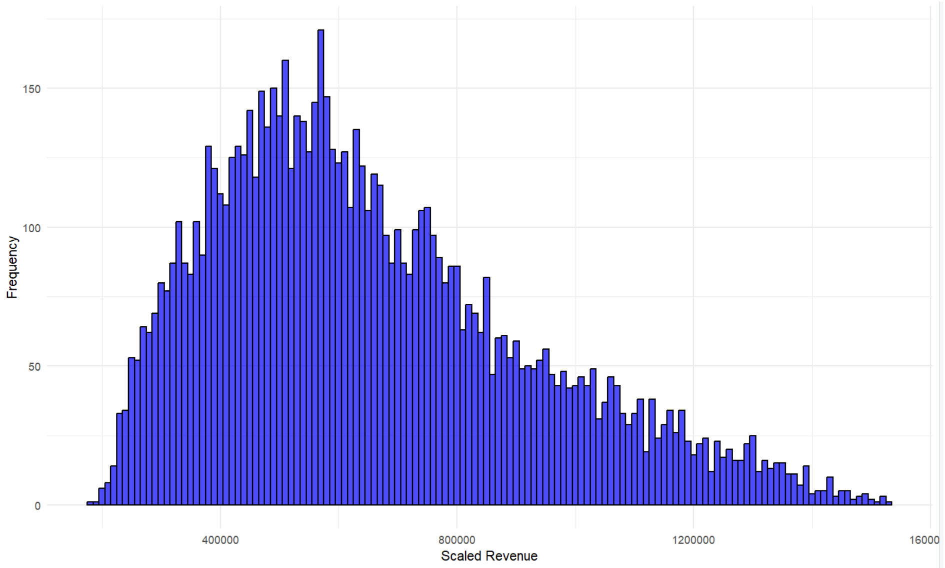


Figure 22: Histogram: Distribution of Scaled Revenue (Revenue per Seat)

The histogram shows that while most restaurants fall within a certain range of scaled revenue, there are a few outliers with significantly higher or lower revenue per seat. This could be indicative of unique business models or market conditions.

3.3 Data splitting

The data is split into two part, training dataset and validation dataset. Training dataset contains 80% of the observations and the validation dataset contains 80% of the observations. The split is performed randomly with RStudio.
For replication, we use the seed 777489.

3.4 Model building

Baseline model When inspecting the interaction between indicators, we notice that the interaction term between 'SeatingCapacity' and 'AverageMealPrice' can practically determines 'Revenue' alone, as shown by the plot below:

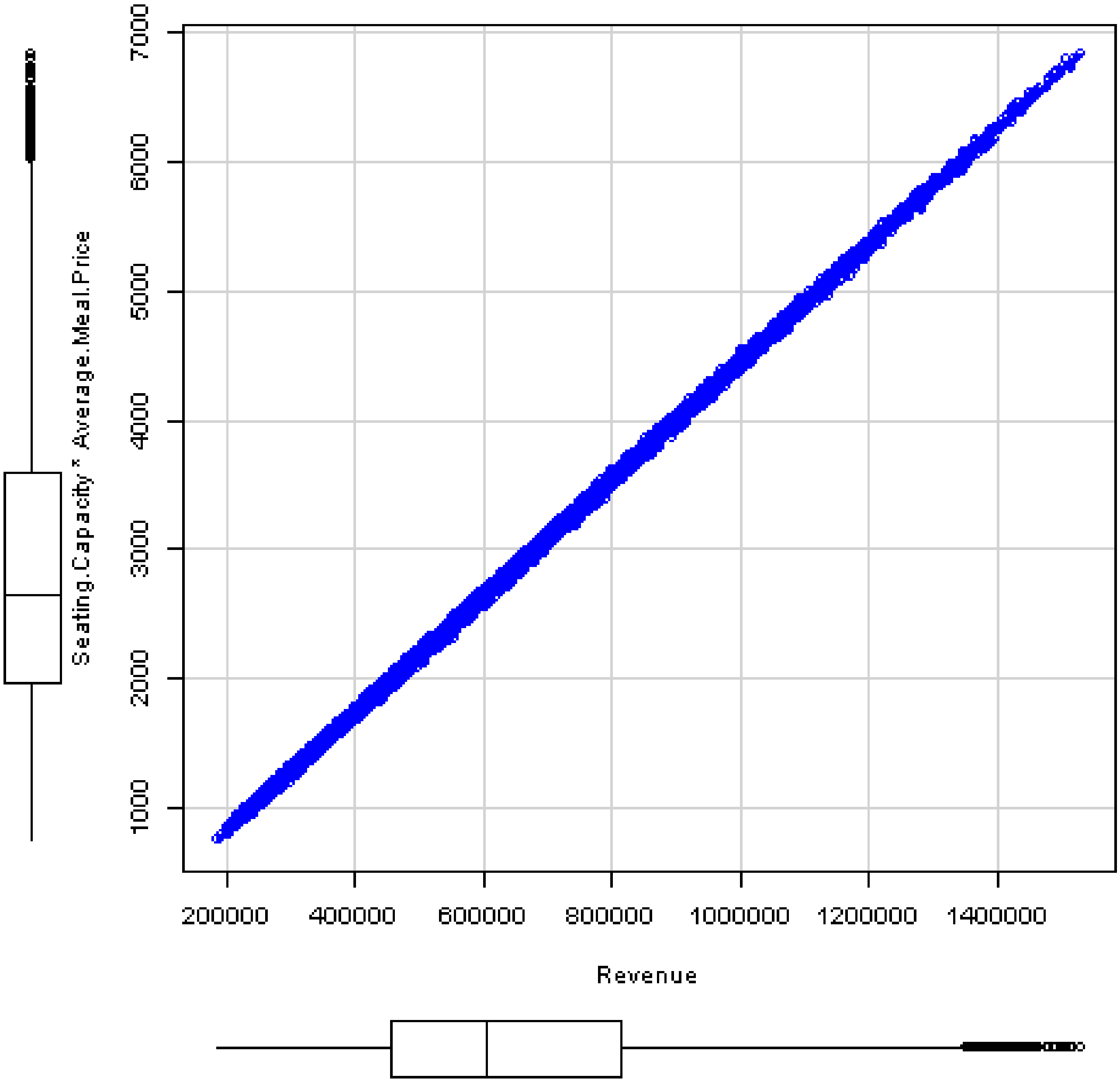


Figure 23: Scatterplot: Revenue by SeatingCapacity*AverageMealPrice

Thus, we choose the baseline model to be:

$$M : \ Revenue \ = \beta_0 + \beta_1 \left(\text{SeatingCapacity} \times \left(\text{AverageMealPrice}\right)\right)$$

3.5 Improvement to a new model

So our baseline model have the form of:

$$M: \ \mathit{Revenue} \ = \beta_0 + \beta_1 \left(\text{SeatingCapacity} \times (\text{AverageMealPrice})\right)$$

So with such a high R^2 and such simple formula for the baseline, it is unneccessary to construct a improved model. In addition, to build a model with R^2 and adjust R^2 surpassing $99,9\%$ is no simple task.

Hence, the main aim for our new model is to fix a problem of the baseline model: The error of the baseline model is not normal distributed. As many statistical tests, like t-tests and F-tests, rely on the assumption of normally distributed errors. These tests allow us to draw inferences about population parameters based on sample data.

Luckily, as our baseline model has high R^2 and adjust R^2 already, we can sacrifice some accuracy for normalizing the error.

3.5.1 Addition of indicators

We start by determine which other variables can indicate the 'Revenue'. From all the indicators available, only 'WeekdayReservation' and 'WeekendReservation' shows signs of indication, but not so much.

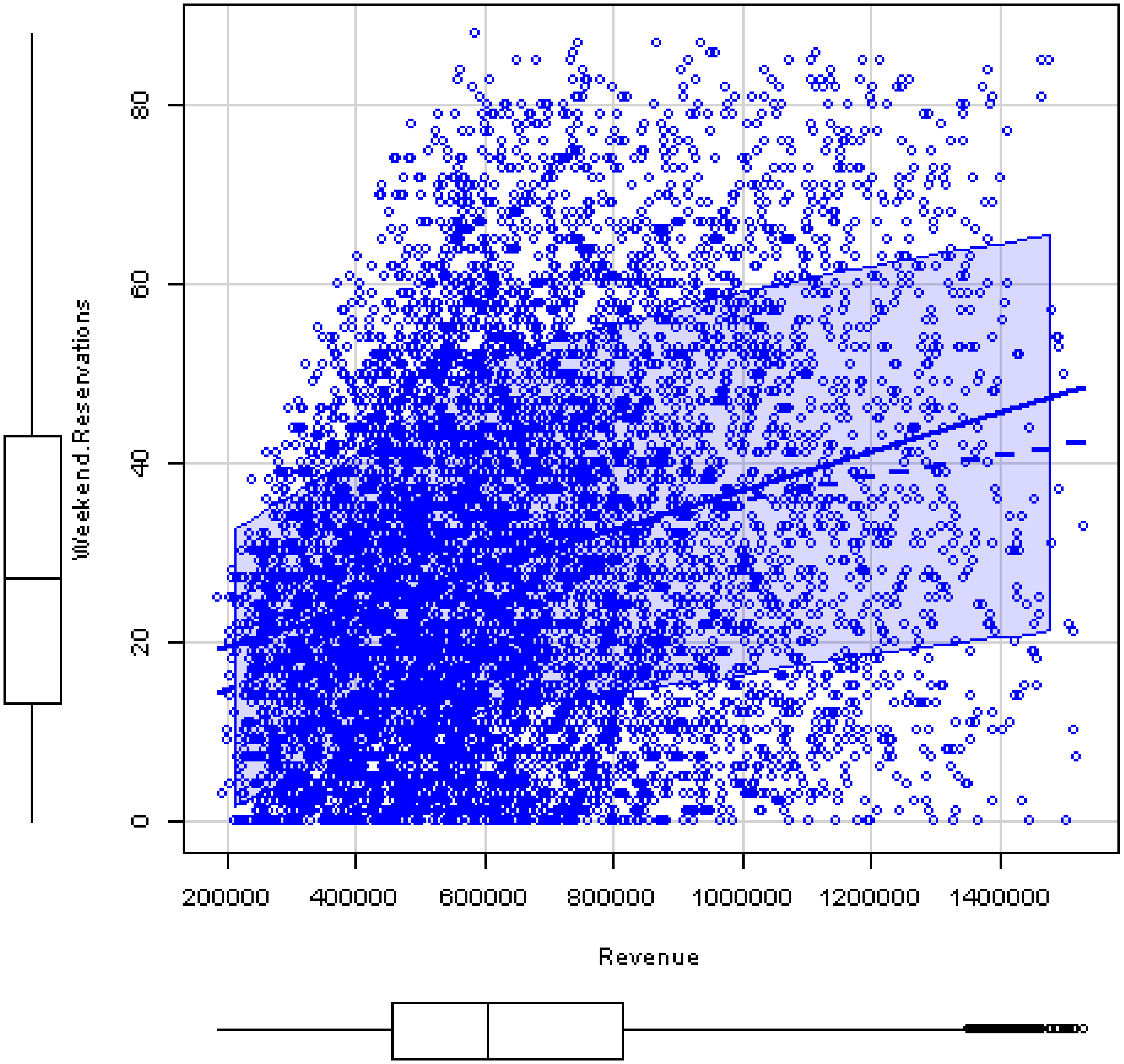


Figure 24: Scatterplot: Revenue by Reservation

Drawing the histogram of the two, we also see that while the distribution is fairly normalized, it is skewed.

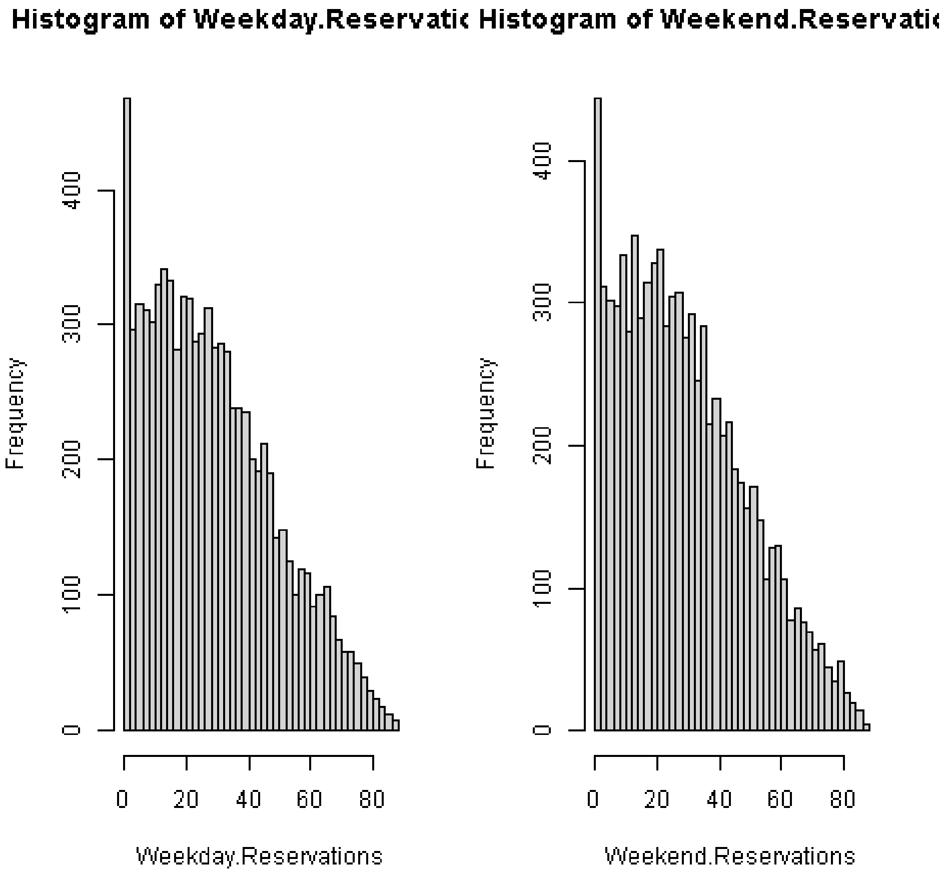
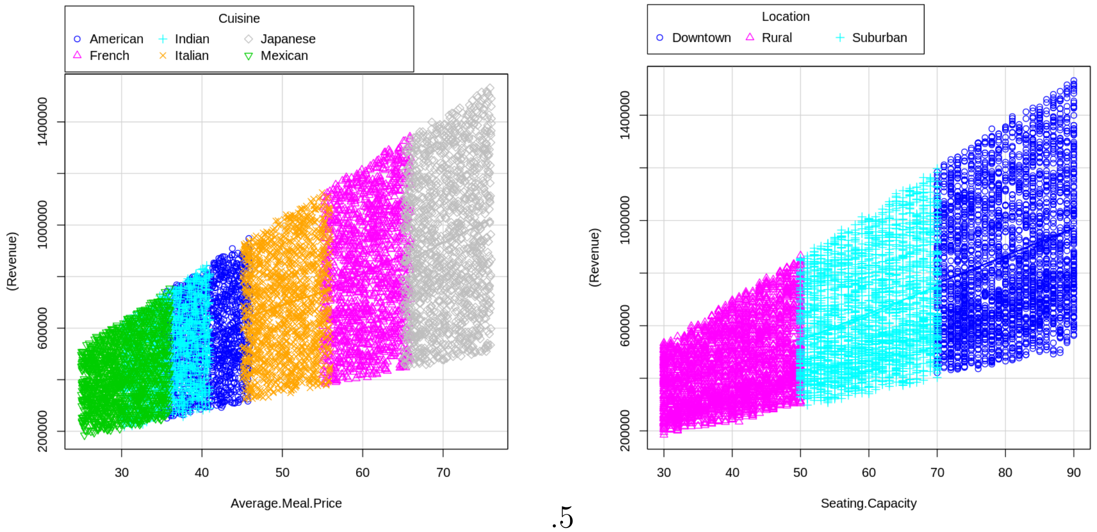


Figure 25: Histogram of the reservations

Therefore, we perform boxcox transformation on the two of them and indicate the λ value to be close to 0.5. As a result, we add the two terms $\sqrt{WeekdayReservation}$ and $\sqrt{WeekdayReservation}$ into the model.

3.5.2 Interaction terms

In order to improve the model to ensure the normality of the residuals, we inspect the interaction terms between a quantitative indicator and a qualitative one. After checking all the interaction terms we can, we conclude that the interaction terms that have an impact on the response is SeatingCapacity*Location and AverageMealPrice*Cuisine, as shown by the graphs below:



.5 .5

Figure 26: Scatterplot: Revenue by SeatingCapacity grouped by Location and AverageMealPrice grouped by Cuisine

Therefore, we will add this in the model for further checking.

3.5.3 Transformation of Average Meal Price

We examine the Box Cox plot of weight:

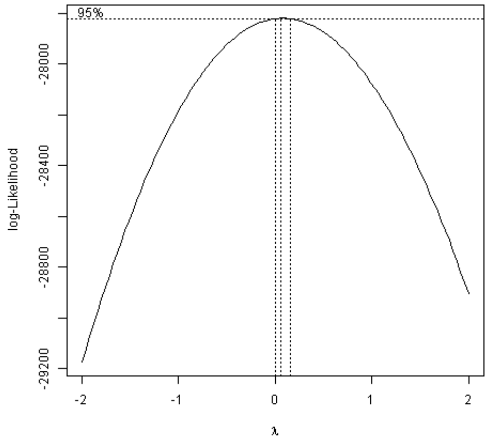


Figure 27: Box Cox plot of Average Meal Price

As Average Meal Price box cox interval is almost 0. We decide to apply the log transformation to it.

3.5.4 Result model

From the previous section, we have the new model:

$$\begin{gathered} \textit{Revenue} \; = \\ \beta_0 + \beta_1\,(\text{SeatingCapacity} \times \log(\text{AverageMealPrice})) + \beta_2\,(\text{SeatingCapacity} \times \text{Location}) + \\ \beta_3\,(\log(\text{AverageMealPrice}) \times \text{Cuisine}) + \beta_4\sqrt{\text{WeekdayReservations}} + \beta_5\sqrt{\text{WeekendReservations}} + \epsilon \end{gathered}$$

Putting this model through stepwise procedure we got the same model.

Table 18: Restaurant model summary

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-148660.8591	5420.5851	-27.43	0.0000
sqrt(Weekday.Reservations)	-647.7837	316.8199	-2.04	0.0409
sqrt(Weekend.Reservations)	-842.6644	323.4639	-2.61	0.0092
Seating.Capacity:log(Average.Meal.Price)	3110.1851	18.1174	171.67	0.0000
Seating.Capacity:LocationRural	984.2925	69.6286	14.14	0.0000
Seating.Capacity:LocationSuburban	285.8088	30.4116	9.40	0.0000
log(Average.Meal.Price):CuisineFrench	47357.0901	515.9533	91.79	0.0000
log(Average.Meal.Price):CuisineIndian	-10379.0297	587.1932	-17.68	0.0000
log(Average.Meal.Price):CuisineItalian	24025.5973	535.5268	44.86	0.0000
log(Average.Meal.Price):CuisineJapanese	69096.7374	514.4629	134.31	0.0000
log(Average.Meal.Price):CuisineMexican	-21938.8074	626.6495	-35.01	0.0000

Multiple R-squared: 0.9659
Adjusted R-squared: 0.9659

Hence, we check the error normality assumption of this resulting model and our baseline.

Table 19: Shapiro-Wilk test result

	W	p-value
Baseline	0.98703	4.078e-11 Improved model
0.99837	0.1033	

So our new model preserve the error normality assumption.

3.6 Hypothesis testing

Before proceeding, we rewrite our model a little bit:

$$\begin{gathered} M : \; \textit{Revenue} \; = \\ \beta_0 + \beta_1\,(\text{SeatingCapacity} \times \log(\text{AverageMealPrice})) + \beta_2\,(\text{SeatingCapacity} \times \text{Location}) + \\ \beta_3\,(\log(\text{AverageMealPrice}) \times \text{Cuisine}) + \beta_4\sqrt{\text{WeekdayReservations}} + \beta_5\sqrt{\text{WeekendReservations}} + \epsilon \end{gathered}$$

3.6.1 Model utility

We are testing the hypothesis:

$$H_0 : \beta_i = 0, i \in \{1, 2, 3, 4, 5\}$$
$$\text{H}_1 : \text{At least one of } \beta_i \neq 0, i \in \{1, 2, 3, 4, 5\}$$

This mean we are calculating the test statistic value $f = \frac{R^2/k}{1-R^2} \frac{1}{(n-(k+1))}$.
Based on the results of your linear regression model, the F-statistic is $F = 18{,}930$, which corresponds to a p-value of less than 2.2×10^{-16}. This extremely low p-value indicates that we can confidently reject the null hypothesis (H_0) that all the coefficients are zero. Therefore, we can conclude that the model contains a significant linear relationship between the response variable `Revenue` and at least one of the predictor variables included in the model.

3.6.2 Error normality assumption

We are testing the hypothesis of:

$$H_0 : \epsilon \text{ is normally distributed}$$
$$H_1 : \epsilon \text{ is not normally distributed}$$

To test this, we can use the test data. Let `predict_revenue` be our result after passing the test observation through the model. Let `revenue` be the corresponding value of that observation. Then, $\epsilon =$ `predict_revenue` $-$ `revenue`. We will run the Shapiro-Wilk test on this resulting error.
Here is the result of our Shapiro-Wilk test:

Table 20: Shapiro-Wilk test result

W	p-value
0.99837	0.1033

This means we have failed to reject the hypothesis that ϵ is normally distributed.

3.6.3 Autocorrelation

We are testing the hypothesis:

$$H_0 : \text{There is no autocorrelation in the error terms of the linear regression model.}$$

$$H_1 : \text{There is autocorrelation in the error terms of the linear regression model.}$$

To test this, we can run the Durbin-Watson test, which is available in R. After running the test, we get the result:

Table 21: Durbin-Watson test result

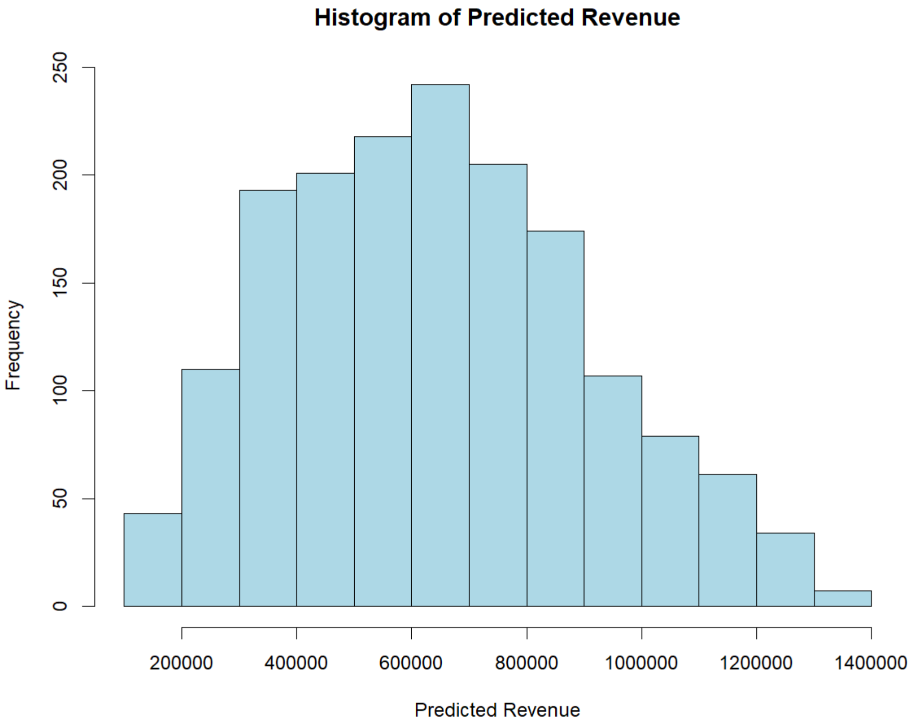
D-W Statistic	p-value
2.012079	0.598

This means we have failed to reject the null hypothesis. This indicates that there is no significant autocorrelation in the error terms of the linear regression model.

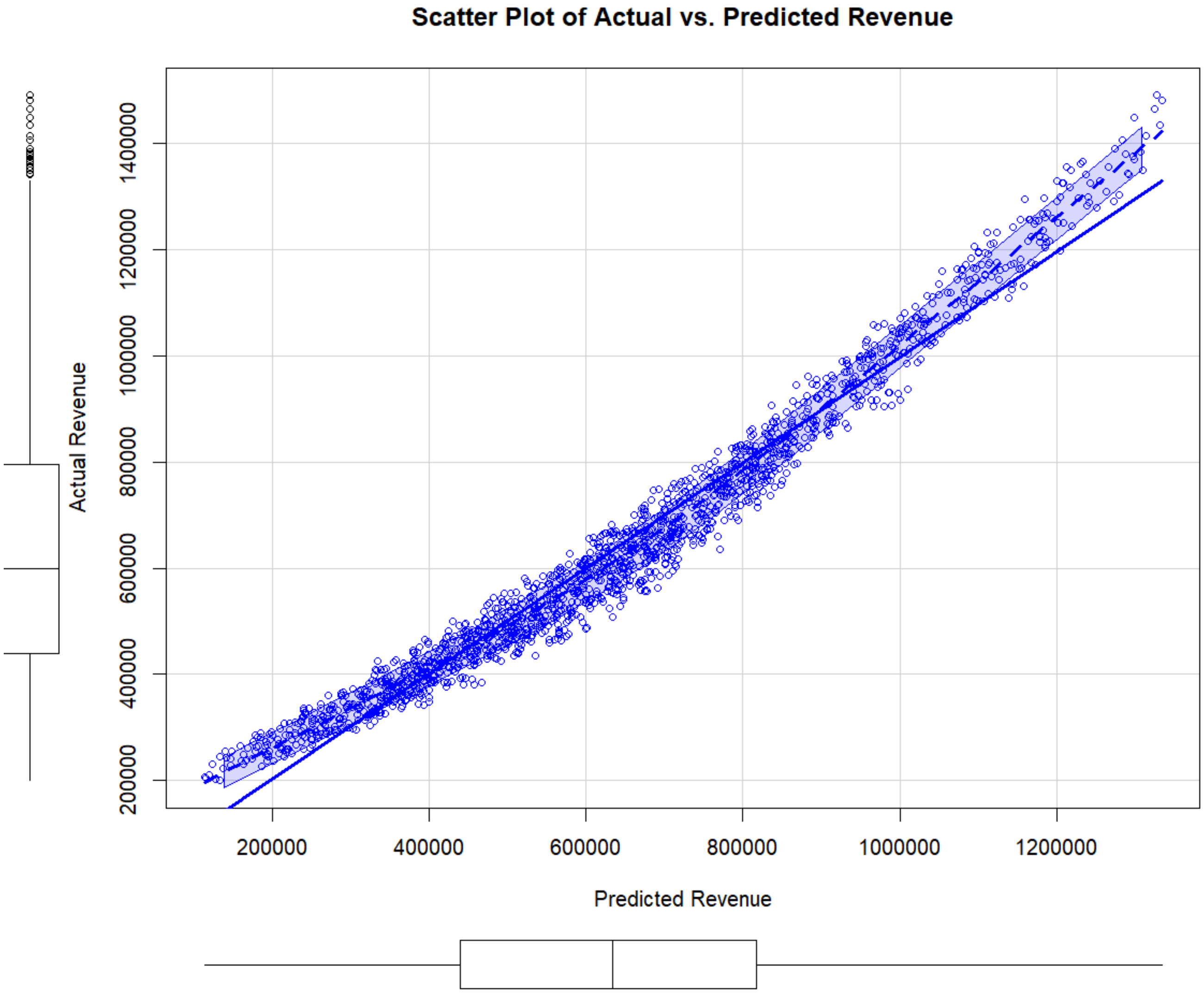
3.7 Validation and testing

We used the same definition of predict_revenue as the previous section. It's the result of passing the test data through our model.

Here is the histogram of predict_revenue:



We compare this to revenue, here is the scatter plot of them:



As they form a almost 45 degree line in the diagonal of the graph, we can say that our model have predict the value of revenue pretty closely.

3.8 Model Meaning

Our model is designed to estimate restaurant revenue based on various factors, including seating capacity, meal price, location, cuisine type, and reservation data. The following is our final model:

$$\begin{gathered} M: \text{ Revenue } = \beta_1 + \beta_2\sqrt{\text{Weekday Reservations}} + \beta_3\sqrt{\text{Weekend Reservations}} + \\ \beta_4 \text{ Seating Capacity} \times \log(\text{Average Meal Price}) + \beta_5 \text{ Seating Capacity} \times \text{Location } + \\ \beta_6 \ \log(\text{Average Meal Price}) \times \text{Cuisine } + \epsilon \end{gathered}$$

With:

$$\beta_1 = -148660.86$$
$$\beta_2 = -647.78$$
$$\beta_3 = -842.66$$
$$\beta_4 = 3110.19$$
$$\beta_5 = \text{Varies by Location}$$
$$\beta_6 = \text{Varies by Cuisine}$$

Variable Definitions:

- Revenue: The restaurant's total revenue.
- Weekday Reservations: Number of reservations made during weekdays.
- Weekend Reservations: Number of reservations made during weekends.
- Seating Capacity: The number of seats available in the restaurant.
- Average Meal Price: The average price of a meal at the restaurant.
- Location: The location of the restaurant (Urban, Suburban, or Rural).
- Cuisine: The type of cuisine served by the restaurant (e.g., French, Indian, Italian, etc.).

Key Takeaways:

- The initial revenue intercept is -148660.86.
- An increase in weekday or weekend reservations tends to decrease revenue slightly.
- Larger seating capacity combined with higher meal prices significantly increases revenue.
- Location impacts revenue, with rural locations having a higher effect than suburban ones.
- The type of cuisine influences the impact of meal price on revenue, with Japanese cuisine showing the highest positive effect.

In layman's terms:

- Restaurants with more seats and higher-priced meals tend to earn more revenue.
- Weekday and weekend reservations slightly reduce revenue, possibly due to operational costs.
- Rural restaurants generally have higher revenue compared to suburban ones, when all other factors are the same.
- Japanese cuisine is the most lucrative when combined with higher meal prices.

Please note that these conclusions are drawn from the dataset used and may not fully represent real-world dynamics. However, within the scope of our data, the model provides accurate estimates of restaurant revenue.

4 Question 2: Dataset 2

4.1 Data Preproccessing

First of all, we import data from the day.csv file. The dataset contains 731 observations with 16 columns, including factors such as weather conditions, seasonal information, temperature, and rental counts.

Dataset Overview Date Range: The data spans from January 1, 2011, to December 31, 2012. Key Variables: season : season (1:springer, 2:summer, 3:fall, 4:winter) workingday : if day is neither weekend nor holiday is 1, otherwise is 0. temp: Normalized temperature in Celsius. atemp: Normalized feeling temperature in Celsius. hum: Normalized humidity. windspeed: Normalized wind speed. casual: Count of casual users. registered: Count of registered users. cnt: Total count of bike rentals (sum of casual and registered).

In this report, we choose to analyze the relationship between total number of rentals due to environment settings. Therefore we will skip the following collumn: instant, date

season	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	registered	cnt
1	0	4	1	2	0.40	0.40	0.67	0.19	3571	3761
4	0	5	1	1	0.36	0.36	0.54	0.21	5283	5992
2	0	1	1	1	0.53	0.53	0.59	0.18	3698	4362
2	0	3	1	2	0.62	0.58	0.77	0.10	4494	5260
2	0	0	0	1	0.61	0.57	0.51	0.23	4286	7132
4	0	3	1	2	0.48	0.47	0.72	0.15	3490	3894

4.1.1 Missing data

According to the author description, there is no missing value in this dataset, so it is not a problem that we should care for.

4.1.2 Dependent variables choosing

Because of a few factors, we chose to use registered as our dependent variable: registered is one of the 3 dependent variables that we can choose and all other variables yr, season, holiday, workingday, weathersit, mnth have some relation with registered in an acceptable level.

4.1.3 Outlier

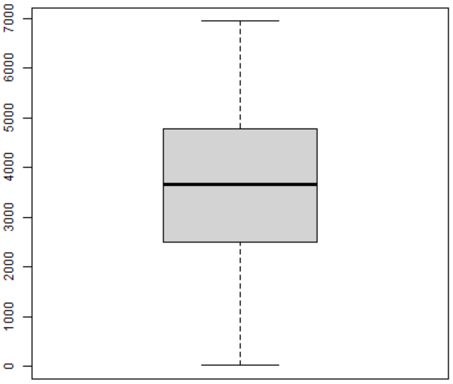


Figure 28: Box plot of registered

Looking at the box plot, we can see that there is no outliers for our dependent variable.

4.2 Data overview

First of all, we might want to look at the summary table contain the statistic descriptive to have the overview about our dataset.

4.2.1 Key Observations:

- **Variation**: The continuous variables show significant variation, particularly **humidity** and **windspeed**.

	temp	atemp	hum	windspeed	registered	cnt
Min	0.05913	0.07907	0.0000	0.02239	20	22
1st Qu.	0.33708	0.33784	0.5200	0.13495	2497	3152
Median	0.49833	0.48673	0.6267	0.18097	3662	4548
Mean	0.49538	0.47435	0.6279	0.19049	3656	4504
3rd Qu.	0.65542	0.60860	0.7302	0.23321	4776	5956
Max	0.86167	0.84090	0.9725	0.50746	6946	8714

- **Central Tendency**: The mean and median values for most variables are close, indicating a relatively symmetrical distribution.
- **Range**: The wide range in values for **registered** indicates days with both low and high rental counts.

This summary provides a good foundation for further analysis, such as examining relationships between variables or identifying potential outliers.

4.3 Data deepdive

To have a deeper look about our dataset and how the variable actually related with each other, we will have to look at the correlation heatmap plot, which was conducted to identify relationships between numeric variables

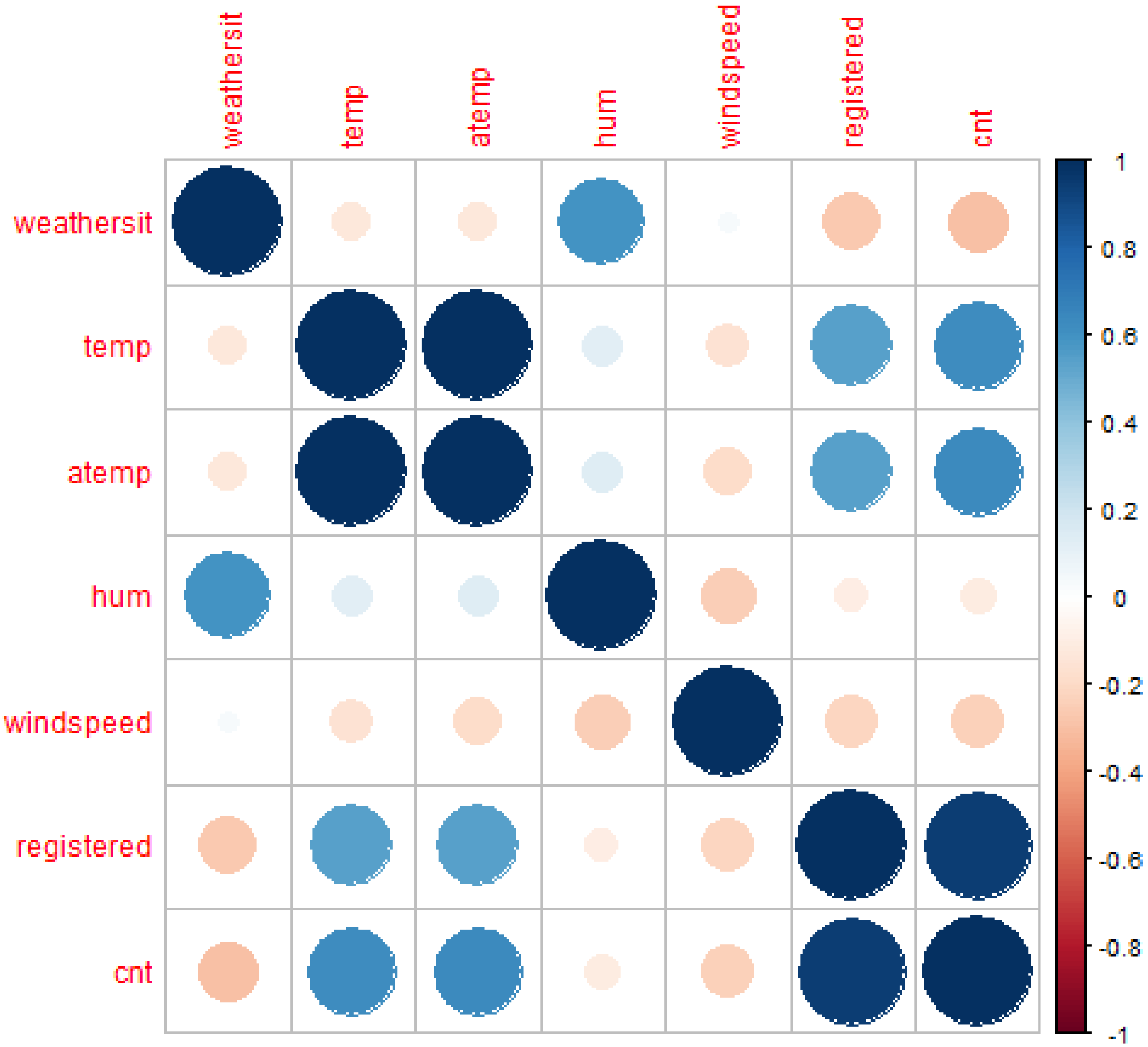


Figure 29: Correlation Matrix

We can observe that there is a strong correlation in some variables, especially between temp and atemp.

4.4 Data splitting

The data is split into two part, training dataset and validation dataset. Training dataset contains 80% of the observations and the validation dataset contains 80% of the observations. The split is performed randomly with RStudio.

For replication, we use the seed 777489.

4.5 Building baseline model

From the above section we can see that temp and atemp have a strong relationship. Below is a plot that describe how much the relation is:

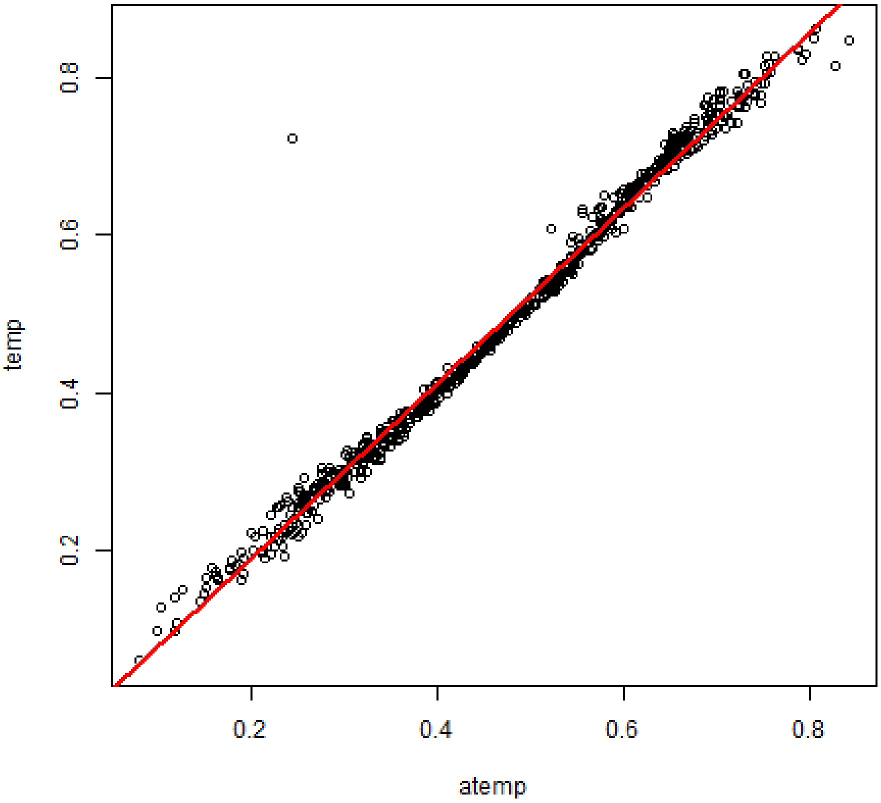


Figure 30: Relation between temp and atemp

As we can see from the plot, temp and atemp have a strong linear relationship. Therefore, we will not use atemp in our model.

In our decription, holiday and workingday and weekday also have strong relation. If it is a holiday then it is not a working day and weekday is 6 it also not a workingday.

To choose which one we will include in our baseline model, we look on how much they impact to the registered through box-plot.

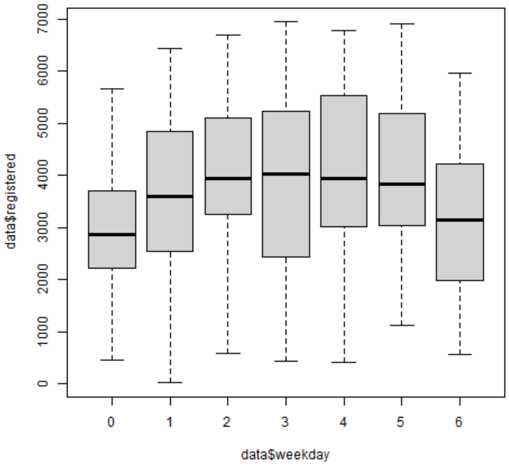


Figure 31: Impact of Weekday

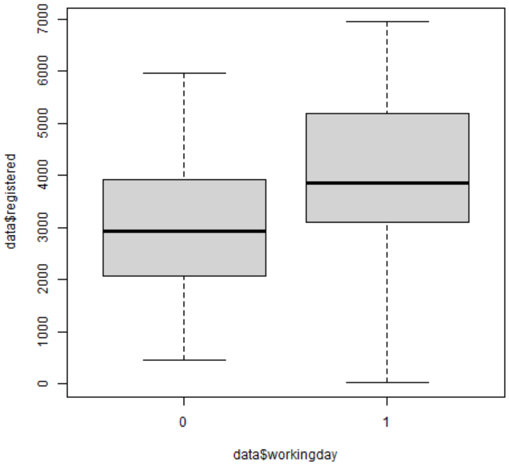


Figure 32: Impact of Workingday

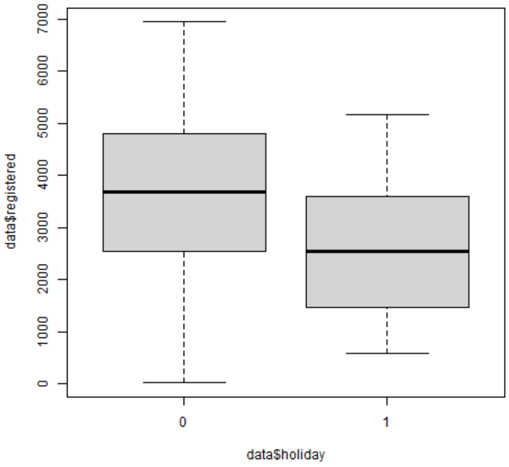


Figure 33: Impact of holiday

From the definition, we can clearly figure out that when weekday is 0 or 6, working day is 0, this lead to multicollinearity. Consider that the impact of the weekday is not more significant than the other two, we decide not to use this variable in out model.

temp:

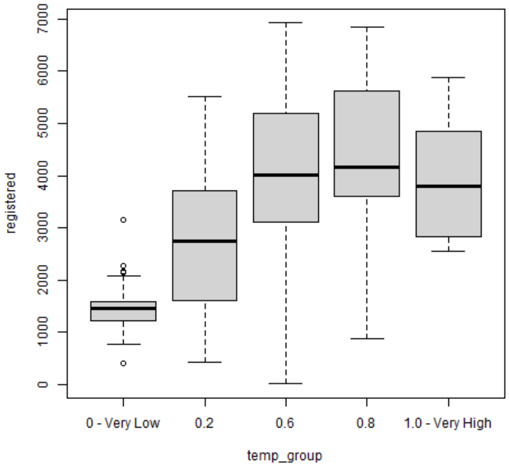


Figure 34: Impact of temp

People tend to use rental bicycles more when temperature is warmer but not too hot.

Humidity:

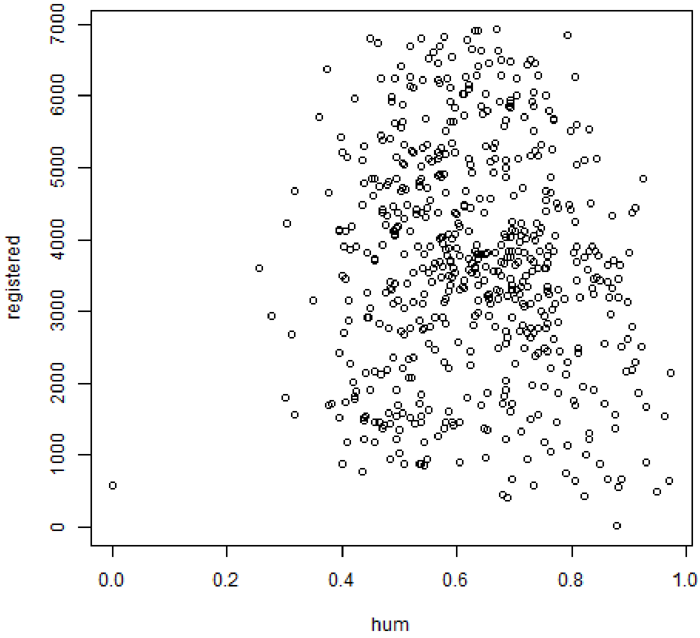


Figure 35: Plot of how registered depend on humidity

As we can see, most of the data collected is in range [0.4,0.8] and humidity is clearly have relation on registered, so we are going to keep it.

Similarly, we will keep windspeed, season and yr, mnth (we may want to eliminate them later) We have our first baseline model

$$registered = \beta_1 \ + \ \beta_2 \ holiday \ + \ \beta_3 \ yr \ + \ \beta_{4,i} \ mnth_i \ + \ \beta_{5,j} \ weathersit_j \ + \ \beta_6 \ temp \ + \\ \beta_7 \ hum \ + \ \beta_8 \ windspeed \ + \ \beta_{9,k} \ season_k$$

4.5.1 Splitting data

set.seed(777489) Randomly, we choose 777489 as seed

We will split 80-20, 80 for the training dataset and 20 for test dataset

4.6 Improving to a better model

Currently, our baseline model is in the form:

$$registered = \beta_1 \ + \ \beta_2 \ holiday \ + \ \beta_3 \ yr \ + \ \beta_{4,i} \ mnth_i \ + \ \beta_{5,j} \ weathersit_j \ + \ \beta_6 \ temp \ + \\ \beta_7 \ hum \ + \ \beta_8 \ windspeed \ + \ \beta_{9,k} \ season_k$$

With:

$$i \in \{2, 3, ..., 12\}$$
$$j \in \{2, 3\}$$
$$k \in \{1, 2, 3\}$$

4.6.1 Inspection of interaction terms

In order to improve the model, we inspect the interaction between qualitative and quantitative variables to determine which interactions impact the response. Using scatter plots, we draw and inspect all combinations between a qualitative variable with a quantitative variable. After all combinations have been tested with, we end up with 4 pairs of indicators that has a relationship with the response.

Figure 36: Scatter plot showing the interaction of indicators influencing the response

4.6.2 Adding the interaction terms into the model

We then adding the interaction terms into the model firstly by removing all the qualitative terms that appears in the interaction terms and then adding all the interaction terms into the model. The new model, while having decent adjusted R^2 value, fails to pass the Shapiro-Wilk test with significance level 0.05 when tested with test data.

Then, from the removed quantitative terms, we check to see if adding any subset of them into the model improve the outcome. The results show that only the subset of the 'weathersit' term improve the model enough to pass the Shapiro-Wilk test while still having decent R^2 value.

4.6.3 The improved model

From various adjustment from the previous sections, we are now left with our new model:

$registered = \beta_1 + \beta_2\, holiday + \beta_3\, yr + \beta_{4,i}\, (temp \times mnth_i) + \beta_{5,i}\, (hum \times mnth_i) + \beta_{6,j}\, (hum \times weathersit_j) + \beta_{7,k}\, (temp \times season_k) + \beta_{8,j}\, weathersit_j + \beta_9\, temp + \beta_{10}\, hum + \beta_{11}\, windspeed$

Table 22: Summary of the new model

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1145.0924	256.8485	4.46	0.0000
weekday1	-276.9376	170.9994	-1.62	0.1059
weekday2	-40.7826	185.5718	-0.22	0.8261
weekday3	-48.8690	185.8878	-0.26	0.7927
weekday4	36.9159	184.6410	0.20	0.8416
weekday5	-77.5401	184.0552	-0.42	0.6737
weekday6	268.8323	94.4005	2.85	0.0046
workingday1	1142.0029	161.2402	7.08	0.0000
yr	1713.6644	54.1621	31.64	0.0000
weathersit2	687.6415	368.8965	1.86	0.0629
weathersit3	-562.2423	4012.8893	-0.14	0.8886
temp	2949.3213	1055.1483	2.80	0.0054
hum	-1475.3014	554.5679	-2.66	0.0080
windspeed	-1794.6719	370.5045	-4.84	0.0000
temp:season2	2283.9905	407.7851	5.60	0.0000
temp:season3	2504.4678	440.2888	5.69	0.0000
temp:season4	3621.3753	453.9504	7.98	0.0000
temp:mnth2	-394.2637	1310.5266	-0.30	0.7636
temp:mnth3	289.4883	1305.9652	0.22	0.8247
temp:mnth4	-813.6620	1263.2824	-0.64	0.5198
temp:mnth5	-2917.4178	1313.2722	-2.22	0.0267
temp:mnth6	-2330.3152	1294.6958	-1.80	0.0724
temp:mnth7	-4711.1287	1255.5836	-3.75	0.0002
temp:mnth8	-2009.8108	1379.9392	-1.46	0.1458
temp:mnth9	16.0030	1427.8247	0.01	0.9911
temp:mnth10	-305.6074	1423.1634	-0.21	0.8301
temp:mnth11	-1701.7563	1733.2287	-0.98	0.3266
temp:mnth12	1919.7097	1745.4852	1.10	0.2719
hum:mnth2	360.2283	605.7862	0.59	0.5523
hum:mnth3	347.2989	695.7390	0.50	0.6179
hum:mnth4	360.2651	634.4346	0.57	0.5704
hum:mnth5	2139.9813	744.6248	2.87	0.0042
hum:mnth6	1713.1093	875.0156	1.96	0.0508
hum:mnth7	3677.8064	816.0531	4.51	0.0000
hum:mnth8	963.3453	953.7447	1.01	0.3129
hum:mnth9	-24.5940	877.4098	-0.03	0.9776
hum:mnth10	206.7723	747.9856	0.28	0.7823
hum:mnth11	830.5844	883.6153	0.94	0.3476
hum:mnth12	-736.0092	783.1167	-0.94	0.3477
weathersit2:hum	-1557.9168	544.4051	-2.86	0.0044
weathersit3:hum	-1323.4362	4450.3771	-0.30	0.7663

Multiple R-squared: 0.8577

Adjusted R-squared: 0.8473

F-statistic: 81.85 on 40 and 543 DF, p-value: ¡ 2.2e-16

We put this through the stepwise method, to our surprise, no variable is eliminated.

Therefore, we just compare the evaluation variable to the baseline.

Table 23: Evaluation parameter of the baseline and improved model

	Improved model	Baseline
R^2	**0.7859**	0.7688
Adjusted R^2	**0.7726**	0.7602
AIC	**9406.626**	9425.379
BIC	9563.943	**9525.887**

There are improvement in 3 out of 4 parameter. This does not assure us the improved model is better. Therefore, we use a wildcard parameter. We run the Shapiro-Wilk test on both model to check whether the error is normal distributed.

Table 24: Shapiro-Wilk test result

	W	p-value
Baseline	0.9594	0.0002534
Improved model	0.9837	0.07929

So the result of the test indicate that the error for the improved model is more likely to be normal distributed. Hence, we decide to pick this as our final model.

4.7 Hypothesis testing

Before proceeding, we rewrite our model a little bit:

$$registered = \beta_1 + \beta_2\, holiday + \beta_3\, yr + \beta_{4,i}\,(temp \times mnth_i) + \beta_{5,i}\,(hum \times mnth_i) + \beta_{6,j}\,(hum \times weathersit_j) + \beta_{7,k}\,(temp \times season_k) + \beta_{8,j}\, weathersit_j + \beta_9\, temp + \beta_{10}\, hum + \beta_{11}\, windspeed$$

4.7.1 Model utility

We are testing the hypothesis:

$$H_0 : \beta_i = 0, i \in \{2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$$
$$\text{H}_1 : \text{At least one of } \beta_i \neq 0, i \in \{2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$$

This mean we are calculating the test statistic value $f = \frac{R^2/k}{1-R^2}\ \frac{1}{(n-(k+1))}$.

Plugging in the number we get $f = 81.85$, this corresponding to a p-value of $2, 2.10^{-16}$. This mean we can be certain to reject H_0. And conclude that our model contains useful linear relationship between our the response `Registered` and at least one of the variable.

4.7.2 Error normality assumption

We are testing the hypothesis of:

$$H_0 : \epsilon \text{ is normal distributed}$$
$$H_1 : \epsilon \text{ is not normal distributed}$$

To test this, we can use the test data. Let's *predict_registred* be our result after passing the test observation through the model. Let's mpg be the corresponding value of that observation. Then, $\epsilon = predict\_registered - registered$. We will run the Shapiro-Wilk test through this resulting error. Here the result of our Shapiro-Wilk test:

Table 25: Shapiro-Wilk test result

W	p-value
0.9837	0.07929

This mean we have failed to reject the hypothesis that ϵ is normal distributed.

4.7.3 Homoskedastic residual variance

Here we are testing the hypothesis of:

$$H_0 : \text{The residuals variances of the groups are equal.}$$
$$H_1 : \text{At least one group has a different residuals variance.}$$

To test this, we can run the Levene's test, which is available in R. After running the test, we get the result:

Table 26: Levene test result

F	p-value
3.392	0.06602

This mean we have failed to reject the null hypothesis. This indicate that the residuals variances of the groups are equal.

4.7.4 Autocorrelation

We are testing the hypothesis:

$$H_0 : \text{There is no autocorrelation in the error terms of the linear regression model.}$$
$$H_1 : \text{There is positive autocorrelation in the error terms of the linear regression model.}$$

To test this, we can run the Durbin-Watson test, which is available in R. After running the test, we get the result:

Table 27: Durbin-Watson test result

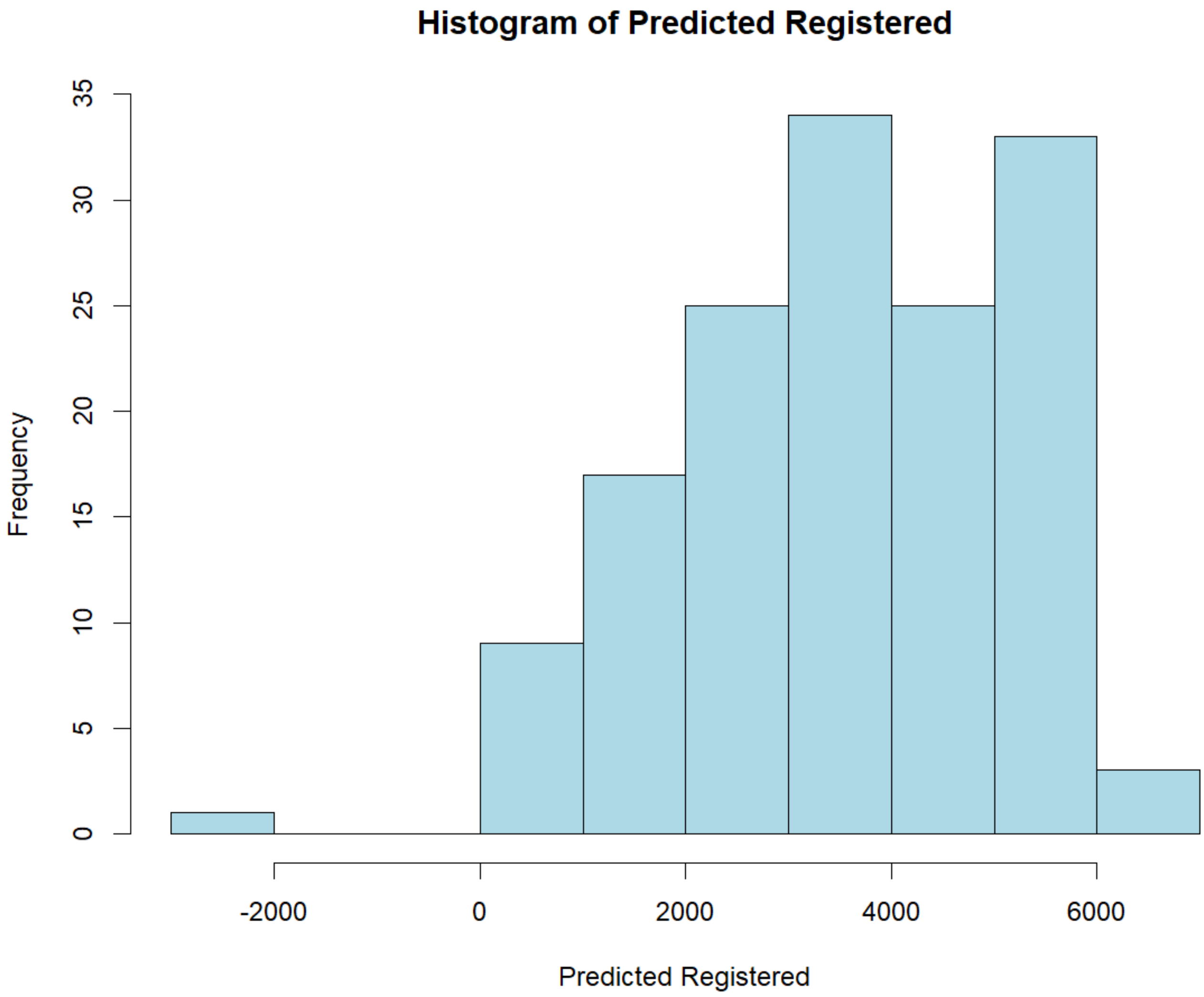
D-W	p-value
2.061362	0.47

This mean we have failed to reject the null hypothesis. This indicate that there is no autocorrelation in the error terms of the linear regression model.

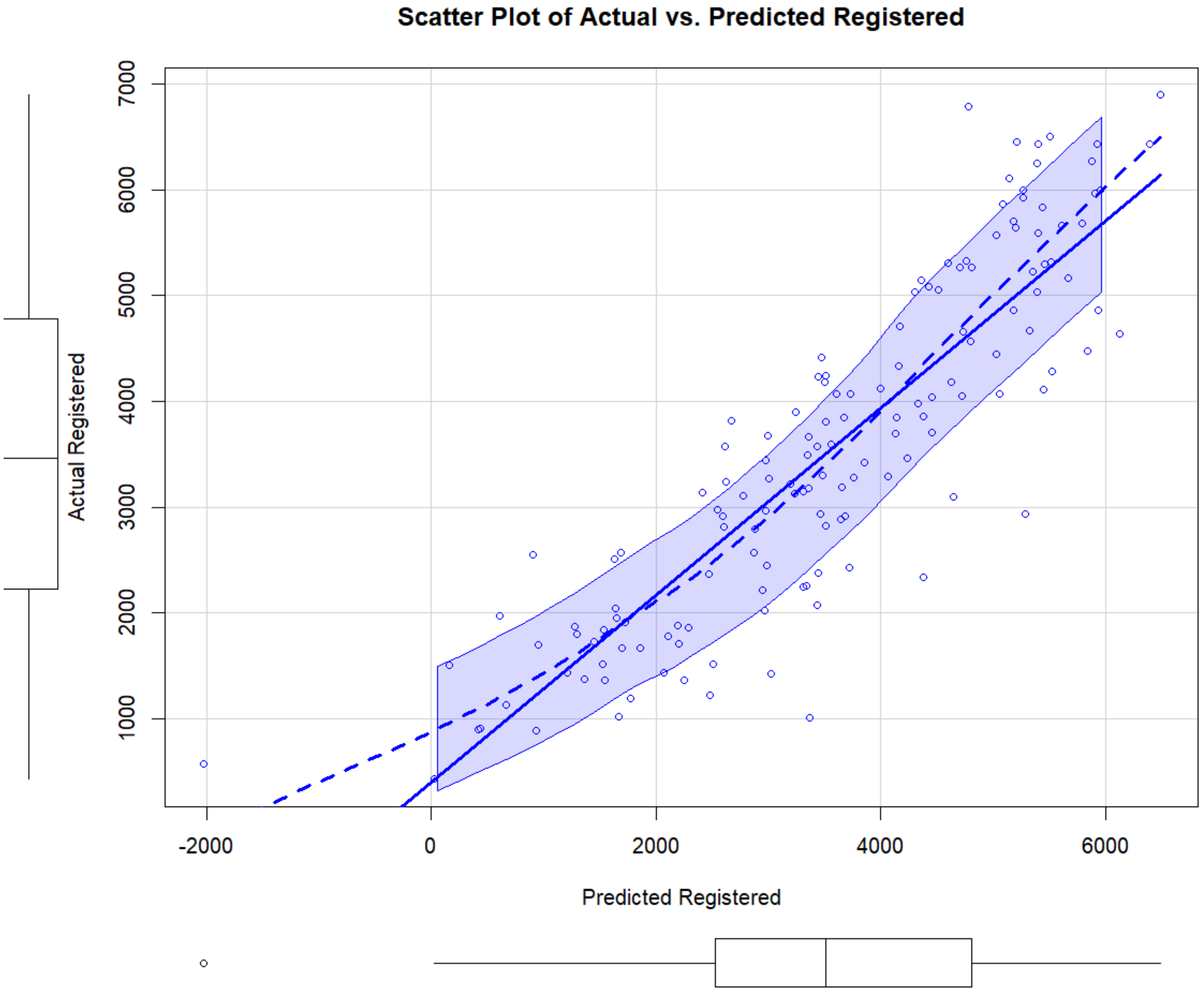
4.8 Validation and testing

We used the same definition of predict_registered as the previous section. It's the result of passing the test data through our model.

Here is the histogram of predict_registered:



We compare this to registered, here is the scatter plot of them:



As they form a almost 45 degree line in the diagonal of the graph, we can say that our model have predict the value of mpg pretty closely.

4.9 Model Meaning

Our model is capable of estimating the number of registered bike rentals based on several parameters: holiday, year, weather situation, temperature, humidity, windspeed, and their interactions with month and season. Since the year is used as a continuous variable, this model can also be used for future predictions, though the accuracy may need to be reexamined for future years.

Overall, here is our final model:

$$M:\ registered\ =$$
$$\beta_1 + \beta_2\, holiday + \beta_3\, yr + \beta_{4,j}\, weathersit_j + \beta_5\, temp + \beta_6\, hum + \beta_7\, windspeed + \beta_{8,k}\, (temp \times season_k) + \beta_{9,i}\, (temp \times mnth_i) + \beta_{10,i}\, (hum \times mnth_i) + \beta_{11,j}\, (hum \times weathersit_j) + \epsilon$$

With:

$$\begin{aligned}
\beta_1 &= 1727.34 \\
\beta_2 &= -954.56 \\
\beta_3 &= 1717.52 \\
\beta_4 &= \begin{cases} 1253.98 & \text{(weathersit2)} \\ -4887.25 & \text{(weathersit3)} \end{cases} \\
\beta_5 &= 2806.06 \\
\beta_6 &= -1103.72 \\
\beta_7 &= -1923.17 \\
\beta_8 &= \begin{cases} 2377.63 & \text{(temp} \times \text{season2)} \\ 2734.72 & \text{(temp} \times \text{season3)} \\ 3958.73 & \text{(temp} \times \text{season4)} \end{cases} \\
\beta_9 &= \text{varies by month (temp} \times \text{mnth)} \\
\beta_{10} &= \text{varies by month (hum} \times \text{mnth)} \\
\beta_{11} &= \text{varies by weather situation (hum} \times \text{weathersit)}
\end{aligned}$$

Variable meanings:

- registered: Number of registered bike rentals.
- holiday: 1 if the day is a holiday, 0 otherwise.
- yr: Year (0 for 2011, 1 for 2012).

- weathersit: Categorical variable representing the weather situation.
- temp: Normalized temperature in Celsius.
- hum: Normalized humidity.
- windspeed: Normalized windspeed.
- season: Categorical variable representing the season.
- mnth: Categorical variable representing the month.

This means that:

- The base number of registered bike rentals, when all other variables are at their base levels, is 1727.34.
- If the day is a holiday, the number of registered rentals decreases by 954.56.
- An increase in the year variable (from 2011 to 2012) increases the number of rentals by 1717.52.
- Different weather situations have varying impacts, with ‘weathersit2‘ increasing rentals by 1253.98 and ‘weathersit3‘ decreasing rentals by 4887.25.
- As temperature increases, rentals increase by 2806.06 units.
- Higher humidity tends to decrease rentals, but this effect varies by month and weather situation.
- Higher windspeed significantly reduces the number of rentals by 1923.17 units.
- The interaction effects show that temperature and humidity have different impacts depending on the season, month, and weather situation.

In layman’s terms, this means:

- Registered bike rentals are higher in the year 2012 compared to 2011.
- Holidays see fewer bike rentals.

- Good weather (as represented by 'weathersit2') increases rentals, while bad weather ('weathersit3') significantly reduces them.
- Warmer days generally lead to more bike rentals.
- High humidity and windspeed can decrease the number of rentals.

Please note that these conclusions are based on the dataset used and may not fully represent real-world conditions. However, within the scope of our data, the model accurately estimates the number of registered bike rentals.

A Appendix

A.1 Code

A.1.1 Question 1

```
{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
Sys.setenv(LANG = "en")
setwd("C:/Users/Lecuo/Desktop/StatLabWithR")
library(dplyr)
#install.packages("xtable")
library(xtable)
#install.packages("papeR")
library(papeR)
library(car)
library(MASS)
library(ggplot2)
#install.packages("plotrix")
library(plotrix)
rm(list = ls())

data<-read.csv("auto_mpg.csv",header=TRUE,sep=';')
dim(data)
sink(file = "dataheading.txt")
data
```

```
sink(file = NULL)

data<-subset(data, data$horsepower != "?")
data$horsepower<-as.double(data$horsepower)
gooddata<-subset(data, select = -c(model_year,origin,car_name))
gooddata
xtable(summarize(gooddata), caption = "Basic univariate summary statistics for
    mgp, cylinders, displacement, horsepower, weight, acceleration")
sink(file = "datasummary.txt")
summary(data)
sink()

jpeg(file="mpgtrans.jpeg")
par(mfrow=c(2,2))
hist(data$weight, breaks = 40)
boxplot(data$weight)
#dev.off()
#jpeg(file="mpgtrans2.jpeg")
#par(mfrow=c(1,2))
hist(log(data$weight), breaks = 40)
boxplot(log(data$weight))
dev.off()
jpeg(file="mpgtrans3.jpeg")
par(mfrow=c(2,2))
hist(sqrt(data$weight), breaks = 40)
boxplot(sqrt(data$weight))
#dev.off()
#jpeg(file="mpgtrans4.jpeg")
#par(mfrow=c(1,2))
hist(1/sqrt(data$weight), breaks = 40)
boxplot(1/sqrt(data$weight))
dev.off()

png(file="mpgloghist.png")
par(mfrow=c(1,1))
mpg<-data$mgp
hist(log(mpg), breaks = 40)
```

```r
dev.off()

png(file="disphist.png")
par(mfrow=c(1,1))
displacement<-data$displacement
hist(displacement, breaks = 40)
dev.off()

png(file="accelhist.png")
par(mfrow=c(1,1))

acceleration <- data$acceleration
hist(acceleration, breaks = 40)
dev.off()

png(file="hphist.png")
par(mfrow=c(1,1))

horsepower <- data$horsepower
hist(horsepower, breaks = 40)
dev.off()

png(file="weighthist.png")
par(mfrow=c(1,1))

weight <- data$weight
hist(weight, breaks = 40)
dev.off()

png(file="cylinderbarplot.png")
par(mfrow=c(1,1))
ggplot(data, aes(x=factor(cylinders))+
  geom_bar(stat="count", width=0.7, fill="steelblue")+
  theme_minimal()
dev.off()

labels(data)
```

```r
gooddata<-data
gooddata$model_year = as.factor((gooddata$model_year))
gooddata$origin = as.factor(gooddata$origin)
gooddata$cylinders = as.factor(gooddata$cylinders)
xtable(summarize(gooddata, type = "factor", variables = "model_year"), caption = "Basic univariate summary statistics for model_year")
xtable(summarize(gooddata, type = "factor", variables = "origin"), caption = "Basic univariate summary statistics for origin")
xtable(summarize(gooddata, type = "factor", variables = "cylinders"), caption = "Basic univariate summary statistics for cylinders")

data = subset(data,data$horsepower!="?")
data
data$horsepower<-as.double(data$horsepower)
data
set.seed(0)
n<-nrow(data)
train_indice<-sample(seq_len(n), size = 0.8 * n)
train_data<-data[train_indice, ]
test_data<-data[-train_indice, ]
train_data
test_data
data<-train_data

data$origin<-as.factor(data$origin)
data
attach(data)
model<-lm(data$mgp~data$cylinders+data$displacement+data$horsepower+data$weight+data$acceleration+data$model_year+data$origin)
model<-step(model)
xtable(summary(model))
#sink(file = "test.txt")
summary(model)
#sink()
AIC(model)
BIC(model)

```

```
summary(model)

model_full<-lm(data$mgp ~ data$displacement + data$weight + data$acceleration + data$model_year + data$origin)
model_reduced<-lm(data$mgp ~ data$weight + data$acceleration + data$model_year + data$origin)
anova(model_full,model_reduced)

boxcox(lm(data$mgp~1))
png("weightboxcox.png")
boxcox(lm(data$weight~1))
dev.off()
png("accelboxcox.png")
boxcox(lm(data$acceleration~1))
dev.off()

acceleration<-data$acceleration
jpeg(file="acceltrans1.jpeg")
par(mfrow=c(2,2))
hist(acceleration, breaks = 40)
boxplot(acceleration)
#dev.off()
#jpeg(file="mpgtrans2.jpeg")
#par(mfrow=c(1,2))
hist(log(acceleration), breaks = 40)
boxplot(log(acceleration))
dev.off()
jpeg(file="acceltrans2.jpeg")
par(mfrow=c(2,2))
hist(sqrt(acceleration), breaks = 40)
boxplot(sqrt(acceleration))
#dev.off()
#jpeg(file="mpgtrans4.jpeg")
#par(mfrow=c(1,2))
hist(1/sqrt(acceleration), breaks = 40)
boxplot(1/sqrt(acceleration))
dev.off()
```

```

shapiro.test(data$acceleration)
shapiro.test(log(data$acceleration))
shapiro.test(sqrt(data$acceleration))
shapiro.test(1/sqrt(data$acceleration))

shapiro.test(data$weight)
shapiro.test(log(data$weight))
shapiro.test(sqrt(data$weight))
shapiro.test(1/sqrt(data$weight))

shapiro.test(data$weight)
shapiro.test(data$acceleration)
shapiro.test(log(data$acceleration))
shapiro.test(log(data$acceleration))

logmgp<-sqrt(sqrt(data$mgp))
shapiro.test(logmgp)
hist(logmgp,breaks=40)

model<-lm(mgp~displacement+horsepower+weight+acceleration+model_year+origin)
summary(model)
AIC(model)
BIC(model)

model<-lm(mgp~displacement+horsepower+weight+model_year+origin)
summary(model)
AIC(model)
BIC(model)

model<-lm(mgp~displacement+weight+model_year+origin)
summary(model)
AIC(model)
BIC(model)

model<-lm(mgp~weight+model_year+origin)
summary(model)
```

```
AIC(model)
BIC(model)

model<-lm(mgp~weight+model_year)
summary(model)
goodmodel<-model
AIC(model)
BIC(model)

model<-lm(mgp~weight)
summary(model)
AIC(model)
BIC(model)

boxcox(lm(data$mgp~1))
boxcox(lm(data$a~1))

#model_year<-as.factor(model_year)
model<-lm(mgp~weight+model_year)
summary(model)
AIC(model)
BIC(model)

eps<-test_data$mgp-predict(goodmodel, newdata=test_data)
eps

shapiro.test(eps)

hist(sqrt(mpg),breaks = 40)

model2<-lm(log(mgp)~data$displacement+log(weight) + acceleration + model_year +
   origin)
#model2<-lm(sqrt(sqrt(mgp))~log(weight) + log(acceleration) + model_year +
   origin)
summary(model2)

model2_2<-step(model2)
```

```
summary(model2_2)

test_data$origin <- as.factor(test_data$origin)
eps<-log(test_data$mgp)-predict(model2_2, newdata=test_data)
eps

shapiro.test(eps)

durbinWatsonTest(model2_2)

leveneTest(residuals(model2_2)˜data$origin)
```

A.1.2 Question 2: Dataset 1

```
{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
Sys.setenv(LANG = "en")
setwd("C:/Users/Lecuo/Desktop/StatLabWithR")
#install.packages("readxl")
library(readxl)
#install.packages("car")
library(car)
#install.packages("corrplot")
library(corrplot)
#install.packages("MASS")
#install.packages("xtable")
#install.packages("papeR")
library(MASS)
library(papeR)
library(xtable)
rm(list = ls())

restaurant<-read.csv(’restaurant_data.csv’, header = T)
restaurant$Parking.Availability<-as.integer(ifelse(restaurant$Parking.Availability == ’Yes’, 1, 0))
restaurant$Cuisine<-as.factor(restaurant$Cuisine)
restaurant$Location<-as.factor(restaurant$Location)
```

```
head(restaurant)

# Get the unique values in the ’group’ column
unique_location <- unique(restaurant$Location)

# Initialize a data.frame to hold the dummy variables
dummy_location <- as.data.frame(matrix(0, nrow = nrow(restaurant), ncol = length(unique_location)))
colnames(dummy_location) <- paste0(unique_location)

# Fill the dummy variables
for (i in seq_along(unique_location)) {
  dummy_location[[i]] <- as.integer(restaurant$Location == unique_location[i])
}

# Combine the original data.frame with the dummy variables
restaurant <- cbind(restaurant, dummy_location)
all_location<-names(dummy_location)
all_location_formula <- paste(all_location, collapse = " + ")
all_location_formula
head(restaurant)

# Get the unique values in the ’group’ column
unique_cuisine <- unique(restaurant$Cuisine)

# Initialize a data.frame to hold the dummy variables
dummy_cuisine <- as.data.frame(matrix(0, nrow = nrow(restaurant), ncol = length(unique_cuisine)))
colnames(dummy_cuisine) <- paste0(unique_cuisine)

# Fill the dummy variables
for (i in seq_along(unique_cuisine)) {
  dummy_cuisine[[i]] <- as.integer(restaurant$Cuisine == unique_cuisine[i])
}

# Combine the original data.frame with the dummy variables
restaurant <- cbind(restaurant, dummy_cuisine)
```

```
all_cuisine<-names(dummy_cuisine)
all_cuisine_formula <- paste(all_cuisine, collapse = " + ")
all_cuisine_formula
head(restaurant)

attach(restaurant)
str(restaurant)

full_model<-lm(log(Revenue) ~ Seating.Capacity + Average.Meal.Price + log(Marketing.Budget) + Weekday.Reservations + Weekend.Reservations,data = restaurant)
vif(full_model)

library(ggplot2)

# Sample data
set.seed(123)
dataset <- data.frame(
  X1 = rnorm(100),
  X2 = rnorm(100),
  Y = 5 + 3 * rnorm(100) + 2 * rnorm(100) * rnorm(100)
)

# Create interaction plot
ggplot(dataset, aes(x = X1, y = Y, color = X2)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Interaction between X1 and X2",
       x = "X1",
       y = "Response Y",
       color = "X2") +
  theme_minimal()

  scatterplot((Revenue)~ c(Weekend.Reservations + Weekday.Reservations) | Cuisine)

  par(mfrow = c(1,2))
```

```
response<-Revenue
predictor<-Cuisine
boxplot((response)~predictor)
boxplot(log(response)~predictor)

png("Averboxcox.png")
boxcox(lm(Average.Meal.Price~1))
dev.off()

tmp <- Average.Meal.Price
out <- boxcox(lm(tmp~1))
range(out$x[out$y > max(out$y)-qchisq(0.95,1)/2])

set.seed(777489)
n<-nrow(restaurant)
train_indice<-sample(seq_len(n), size = 0.8 * n)
train_data<-restaurant[train_indice, ]
test_data<-restaurant[-train_indice, ]

str(restaurant)

model_1<-lm(log(Revenue) ~ log(Seating.Capacity) + log(Average.Meal.Price) + (Marketing.Budget) + Weekday.Reservations * Weekend.Reservations + Parking.Availability,data = train_data)
model_1<-update(model_1, as.formula(paste(". ~ . + ", all_location_formula)), data =train_data)
model_1<-update(model_1, as.formula(paste(". ~ . + ", all_cuisine_formula)), data =train_data)
model_1<-step(model_1)
summary(model_1)

durbinWatsonTest(model_1)

prediction<-predict(model_1, newdata = test_data)
residuals<-log(test_data$Revenue) - prediction
shapiro.test(residuals)

```

```
model_2<-lm(Revenue ~ Seating.Capacity:log(Average.Meal.Price) + Seating.
    Capacity:Location + log(Average.Meal.Price):Cuisine + sqrt(Weekday.
    Reservations) + sqrt(Weekend.Reservations), data = train_data)
#model_2<-update(model_2, as.formula(paste(". ~ . + ", all_location_formula)),
    data =train_data)
#model_2<-update(model_2, as.formula(paste(". ~ . + ", all_cuisine_formula)),
    data =train_data)
summary(model_2)
model_2<-step(model_2)
summary(model_2)

durbinWatsonTest(model_2)

prediction<-predict(model_2, newdata = test_data)
residuals<-(test_data$Revenue) - prediction
shapiro.test(residuals)

xtable(summary(model_2))
```

A.1.3 Question 2: Dataset 2

```
    {r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
Sys.setenv(LANG = "en")
setwd("C:/Users/Lecuo/Desktop/StatLabWithR")
#install.packages("readxl")
library(readxl)
#install.packages("car")
library(car)
#install.packages("corrplot")
library(corrplot)
#install.packages("MASS")
library(MASS)
rm(list = ls())

hour<-read.csv("day.csv", header = T)
hour$season<-as.factor(hour$season)
```

```
#hour$year<-as.factor(hour$year)
hour$holiday<-as.factor(hour$holiday)
hour$weekday<-as.factor(hour$weekday)
hour$workingday<-as.factor(hour$workingday)
hour$weathersit<-as.factor(hour$weathersit)
hour$mnth<-as.factor(hour$mnth)

head(hour)

# Get the unique values in the ’group’ column
unique_hr <-c(’early’)

# Initialize a data.frame to hold the dummy variables
dummy_hr <- as.data.frame(matrix(0, nrow = nrow(hour), ncol = length(unique_hr)))
colnames(dummy_hr) <- paste0("hr_", unique_hr)

first_half <- hour$hr <= 12
#dummy_hr$hr_early <- ifelse(first_half, 1, 0)
#dummy_hr$hr_late <- ifelse(first_half, 0, 1)

# Combine the original data.frame with the dummy variables
hour <- cbind(hour, dummy_hr)
all_hr<-names(dummy_hr)
all_hr_formula <- paste(all_hr, collapse = " + ")
all_hr_formula
head(hour)

# Get the unique values in the ’group’ column
unique_mnth <- c(’mid’)

# Initialize a data.frame to hold the dummy variables
dummy_mnth <- as.data.frame(matrix(0, nrow = nrow(hour), ncol = length(unique_mnth)))
colnames(dummy_mnth) <- paste0("mnth_", unique_mnth)

first_half <- ((as.integer(hour$mnth) <= 10) & (as.integer(hour$mnth) >= 5))
```

```
dummy_mnth$mnth_mid <- ifelse(first_half , 1,0)

# Combine the original data.frame with the dummy variables
hour <- cbind(hour, dummy_mnth)
all_mnth<-names(dummy_mnth)
all_mnth_formula <- paste(all_mnth, collapse = " + ")
all_mnth_formula
head(hour)

# Get the unique values in the 'group' column
unique_season <- unique(hour$season)

# Initialize a data.frame to hold the dummy variables
dummy_season <- as.data.frame(matrix(0, nrow = nrow(hour), ncol = length(
    unique_season)))
colnames(dummy_season) <- paste0("season_", unique_season)

# Fill the dummy variables
for (i in seq_along(unique_season)) {
  dummy_season[[i]] <- as.integer(hour$season == unique_season[i])
}

# Combine the original data.frame with the dummy variables
hour <- cbind(hour, dummy_season)
all_season<-names(dummy_season)
all_season_formula <- paste(all_season, collapse = " + ")
all_season_formula
head(hour)

attach(hour)

str(hour)

data<-hour
numeric_data <- data[sapply(data, is.numeric)]
cor_matrix<-cor(numeric_data)
corrplot(cor_matrix, method = "circle", type = "upper",
          tl.col = "black", tl.srt = 45, # Text label color and rotation
          col = colorRampPalette(c("red", "white", "blue"))(200), # Color palette
          addCoef.col = "black", # Add correlation coefficient values
          diag = FALSE) # Remove the diagonal

full_model<-lm(casual~ temp + hum + windspeed,data = hour)
vif(full_model)

boxcoxnc(hour$w, method=’sw’, lambda2=2, lambda =seq(-5,5,0.01),plot=TRUE, alpha=0.05, verbose=TRUE)

par(mfrow = c(2,2))
target<-registered
hist(target, breaks = 40)
boxplot(target)
hist(log(target), breaks = 40)
boxplot(log(target))

guess<-casual
shapiro.test(guess)
shapiro.test(log(guess))

par(mfrow = c(1,2))
response<-registered
predictor<-weathersit
boxplot((response)~predictor)
boxplot(log(response)~predictor)

par(mfrow = c(2,2))
response<-registered
predictor<- temp * windspeed
plot(predictor, response)
plot(predictor, log(response))
plot(log(predictor), response)
plot(log(predictor), log(response))

holiday <- as.double(holiday)
```

```
workingday <- as.double(workingday)
weekday <- as.double(workingday)
vif(lm(registered~holiday+workingday))
cor(data.frame(registered,holiday,workingday,weekday))
holiday <- as.factor(holiday)
workingday <- as.factor(workingday)
weekday <- as.factor(workingday)

scatterplot((registered)~ hum | weathersit)

set.seed(777489)
n<-nrow(hour)
train_indice<-sample(seq_len(n), size = 0.8 * n)
train_data<-hour[train_indice, ]
test_data<-hour[-train_indice, ]

model1<-lm((registered) ~ weekday + workingday + holiday + yr + mnth + weathersit + (temp) + (hum) + (windspeed), data = train_data)
#model1<-update(model1, as.formula(paste(". ~ . + ", all_mnth_formula)), data = train_data)
model1<-update(model1, as.formula(paste(". ~ . + ", all_season_formula)), data = train_data)
#model1<-update(model1, as.formula(paste(". ~ . + ", all_hr_formula)), data = train_data)
model1<-step(model1)
summary(model1)
AIC(model1)
BIC(model1)

durbinWatsonTest(model1)

prediction<-predict(model1, newdata = test_data)
residuals<-test_data$registered - prediction
shapiro.test(residuals)

leveneTest(residuals(model1)~ train_data$weathersit)

```

```
model2<-lm((registered) ~ holiday + yr + weathersit + temp + (temp):season + temp:mnth + (hum) + hum:mnth + hum:weathersit + (windspeed) , data = train_data)
summary(model2)
#model2<-update(model2, as.formula(paste(". ~ . + ", all_mnth_formula)), data = train_data)
#model2<-update(model2, as.formula(paste(". ~ . + ", all_season_formula)), data =train_data)
#model2<-update(model2, as.formula(paste(". ~ . + ", all_hr_formula)), data = train_data)
model2<-step(model2)
summary(model2)
AIC(model2)
BIC(model2)

prediction<-predict(model2, newdata = test_data)
residuals<-test_data$registered - prediction
shapiro.test(residuals)

xtable(summary(model2))
```