STAT 435 Homework 1

Jaiden Atterbury

2024-04-02

Note:
In this homework, I briefly collaborated with Aarav Vishesh Dewangan and Annabel Wade.

Exercise 1:
In this exercise, we will consider the statistical model $y = f(x) + \epsilon$, and assume that $E[\epsilon] = 0$ and $\epsilon \perp x$. Using training data, we estimate the true function f by an approximation $\hat{f}$. Given that we have a new test data point $\{x_0, y_0\}$, the goal of this exercise is to show that the expected test mean squared error for x_0 can always be written as the sum of the variance of $\hat{f}(x_0)$, the squared bias of $\hat{f}(x_0)$, and the variance of the stochastic error ϵ. In other words, our goal is to show that

$$E\left[(y_0 - \hat{f}(x_0))^2\right] = \text{Var}(\hat{f}(x_0)) + \left(\text{Bias}(\hat{f}(x_0))\right)^2 + \text{Var}(\epsilon)$$

For brevity, we will denote $\hat{f}(x_0) = \hat{f}$ and $f(x_0) = f$. Hence, noting that $y_0 = f(x_0) + \epsilon$ and switching the notation, we can see that

$$E\left[(y_0 - \hat{f}(x_0))^2\right] = E\left[(f + \epsilon - \hat{f})^2\right]$$

Using the hint given in the problem description, we can expand $E\left[(f + \epsilon - \hat{f})^2\right]$ and write it as

$$\begin{aligned} E\left[(f + \epsilon - \hat{f})^2\right] = E[&(f - E(\hat{f}))^2 + \epsilon^2 + (\hat{f} - E(\hat{f}))^2 + 2\epsilon(f - E(\hat{f})) + 2\epsilon(\hat{f} - E(\hat{f})) \\ &+ 2(f - E(\hat{f}))(\hat{f} - E(\hat{f}))] \end{aligned}$$

Using the linearity of expectation, we can change the problem from finding an expectation of the sum of many terms to finding the sum of the expectations of the various terms. Using this property we can write the problem as

$$\begin{aligned} E\left[(f + \epsilon - \hat{f})^2\right] = &E\left[(f - E(\hat{f}))^2\right] + E\left[\epsilon^2\right] + E\left[(\hat{f} - E(\hat{f}))^2\right] + 2E\left[\epsilon(f - E(\hat{f}))\right] + 2E\left[\epsilon(\hat{f} - E(\hat{f}))\right] \\ &+ 2E\left[(f - E(\hat{f}))(\hat{f} - E(\hat{f}))\right] \end{aligned}$$

Noting that at a specific x_0 the function f is a constant, and that $E[\hat{f}]$ is always a constant, it follows that these terms are not random. Therefore, we can remove certain functions of f and $E[\hat{f}]$ from their expectations. Thus we can rewrite the problem as

$$\begin{aligned} E\left[(f + \epsilon - \hat{f})^2\right] = &(f - E(\hat{f}))^2 + E\left[\epsilon^2\right] + E\left[(\hat{f} - E(\hat{f}))^2\right] + 2(f - E(\hat{f}))E[\epsilon] + 2E[(\hat{f} - E(\hat{f}))\epsilon] \\ &+ 2(f - E(\hat{f}))E[(\hat{f} - E(\hat{f}))] \end{aligned}$$

Our next step in the process of this proof is to simplify the terms $2(f - E(\hat{f}))E[(\hat{f} - E(\hat{f}))]$ and $2E[(\hat{f} - E(\hat{f}))\epsilon]$. Starting off with, $2(f - E(\hat{f}))E[(\hat{f} - E(\hat{f}))]$, we can use the linearity of expectation to see that $E[(\hat{f} - E(\hat{f}))] = E[\hat{f}] - E[E(\hat{f})]$. Furthermore, since $E[\hat{f}]$ is a constant it follows that $E[E(\hat{f})] = E[\hat{f}]$, and hence $E[(\hat{f} - E(\hat{f}))] = E[\hat{f}] - E[\hat{f}] = 0$, which means that $2(f - E(\hat{f}))E[(\hat{f} - E(\hat{f}))] = 0$. Moving

onto $2E[(\hat{f} - E(\hat{f}))\epsilon]$, we will start by distributing the ϵ and using the linearity of expectation to obtain $2E[\hat{f}\epsilon] - 2E[E(\hat{f})\epsilon]$. Using the fact that $E[\hat{f}]$ is a constant, along with the fact that x_0 being a new data point shows us that $x_0 \perp \epsilon \implies \hat{f}(x_0) \perp \epsilon$, we can rewrite the expression as $2E[\hat{f}]E[\epsilon] - 2E[\hat{f}]E[\epsilon] = 0$. Removing these terms that are equal to zero we can rewrite the problem as

$$E\left[(f + \epsilon - \hat{f})^2\right] = (f - E(\hat{f}))^2 + E\left[\epsilon^2\right] + E\left[(\hat{f} - E(\hat{f}))^2\right] + 2(f - E(\hat{f}))E[\epsilon]$$

Notice that we assumed that $E[\epsilon] = 0$, hence any term including this expectation will equal zero. Removing these terms that are equal to zero, reduces the problem to the following

$$E\left[(f + \epsilon - \hat{f})^2\right] = (f - E(\hat{f}))^2 + E\left[\epsilon^2\right] + E\left[(\hat{f} - E(\hat{f}))^2\right]$$

In order to complete this exercise, we will go one term at a time from left to right. First off, if we use the fact that $(a - b)^2 = (b - a)^2$, we can rewrite $(f - E(\hat{f}))^2$ as $(E(\hat{f}) - f))^2$. Furthermore, since $\text{Bias}(\hat{f}) = E[\hat{f}] - f$, it follows that $(E(\hat{f}) - f))^2 = (\text{Bias}(\hat{f}))^2$. Moving onto the next term, if we use the fact that $\text{Var}(X) = E[X^2] - E[X]^2$ and $E[\epsilon]^2 = 0$, then the term $E[\epsilon^2]$ can be rewritten as $E[\epsilon^2] - E[\epsilon]^2$, which is in turn $\text{Var}(\epsilon)$. Finally, moving onto the last term, if we use the fact that $\text{Var}(X) = E[(X - E[X])^2]$, then $E\left[(\hat{f} - E(\hat{f}))^2\right]$ can be rewritten as $\text{Var}(\hat{f})$. Altogether this leaves us with the following

$$E\left[(f + \epsilon - \hat{f})^2\right] = \Big(\text{Bias}(\hat{f})\Big)^2 + \text{Var}(\epsilon) + \text{Var}(\hat{f})$$

Simply using the commutative property and changing the notation back to its original form, we can match the exact form of the expected test mean squared error that was provided in the problem description. Therefore we have shown that the expected test mean squared error for x_0 can be written as

$$E\left[(y_0 - \hat{f}(x_0))^2\right] = \text{Var}(\hat{f}(x_0)) + \Big(\text{Bias}(\hat{f}(x_0))\Big)^2 + \text{Var}(\epsilon)$$

Which is the bias-variance tradeoff equation.

Exercise 2
In this exercise, we will consider a dataset of housing prices from a fictional city. We are tasked with developing a model to predict the price of a house based on certain features of the house itself. In particular, we have access to a training set containing 1000 home records and a testing set containing 300 home records. The goal is to develop a model that generalizes well to the testing dataset. To achieve this, we must explore the bias-variance tradeoff when selecting the optimal model complexity.

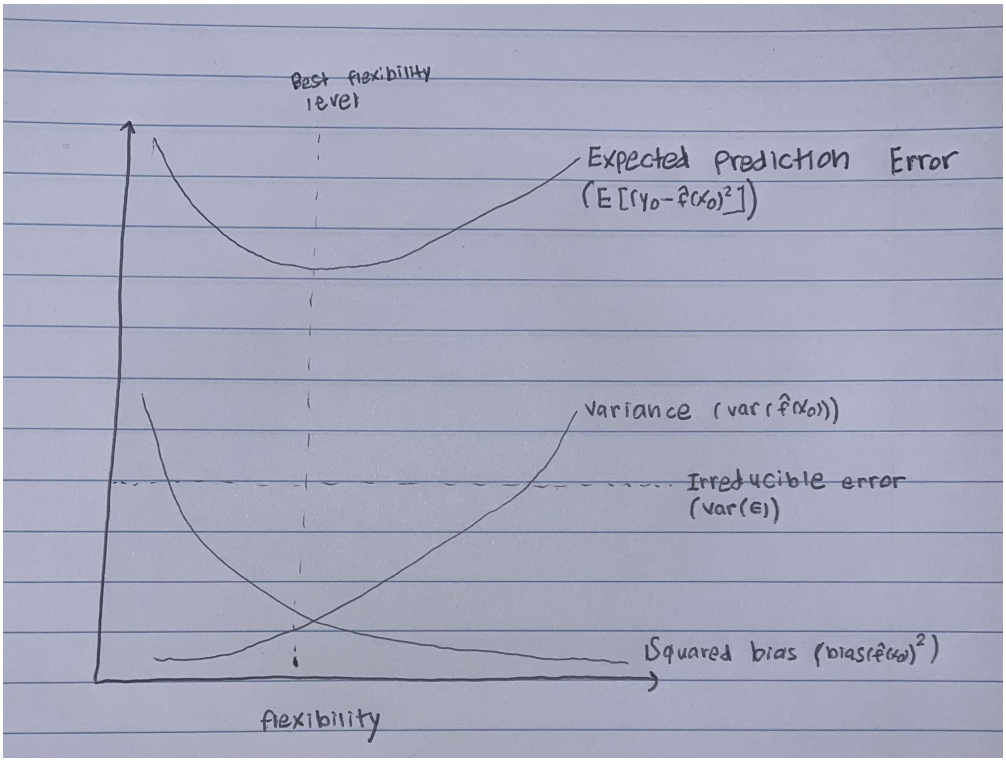
Part a
In this subpart, we will explain the concepts of bias and variance in the context of model predictions and discuss their relationship with the model complexity.

In the context of model predictions, bias refers to systematic errors that your model is making when predicting some target value. More rigorously, bias is defined as the difference between the expected prediction of our model and the true value of the response variable which we are trying to predict. Through the lens of our current problem, the bias would encapsulate how well our model predicts the true house price, up to a certain amount of irreducible error. High bias usually occurs when incorrect assumptions are made about the underlying problem that is trying to be predicted, such as fitting a linear model when the true relationship is quadratic. In this case, our model will never be able to account for the fact that the true relationship is quadratic, and thus will make systematic errors when making future predictions on unseen data. This situation is called underfitting. As a model gets more complex/flexible, it assumes less and is able to capture true relationships in the data, and thus has low bias. On the other hand, as a model gets less complex/flexible, it assumes more and is unable to capture certain relationships in the data, and thus has high bias (and subsequently underfits the data).

On the other hand, in the context of model predictions, the variance in our predictions measures how thoroughly our model captures different patterns and "noise" in the dataset. In other words, it is how sensitive a model is to fluctuations in the dataset. Through the lens of our current problem, the variance would capture how much our predictions on the true house price would typically fluctuate if the model were to be fit on completely new training sets. A model with high variance is great at picking up on all of patterns in the dataset, even those which don't truly exist in reality. The problem with this is that, depending on the training set that a model is fit on, vastly different models will be created. In this case, "different" corresponds to the differences in the results of predicting the same individual in different models. A model with high variance has great training accuracy, but often time performs poorly on new unseen data. This situation is called overfitting. As a model gets more complex/flexible, it has more leeway to understand all of the intricate details in a data set, and thus has high variance (and subsequently overfits the data). On the other hand, as a model gets less complex/flexible, it has less ability to find patterns in the data, and thus each model predicts more or less the same than any other model trained on a different training set, and thus has low variance.

In any model, low bias and low variance is wanted. However, as one decreases, the other increases. Thus, in order to create the most optimal model, one must find the model complexity/flexibility that allows for the combination of bias and variance that minimizes the testing error. This situation is what is known as the bias-variance tradeoff.

Part b
In this part of the exercise, we will sketch and label the following curves with "flexibility" on the x-axis: squared bias, variance, irreducible error, and the expected prediction error. On the curve the "best" level of flexibility will be indicated. This sketch is presented below.



Part c
In this part of the exercise, we will consider three different models used to predict the housing prices:

i. A linear model with only the number of bedrooms as a feature.
ii. A linear model with the number of bedrooms, square footage, and location score as features.
iii. A polynomial model of degree 10 using the number of bedrooms, square footage, and location score as features.

The goal of this part of the exercise is to rank these models, from least to greatest, with respect to the bias, variance, and training MSE that we would expect them to achieve. Furthermore, we will explain if it is possible to rank these models with respect to test MSE.

We will start with ranking the models with respect to bias. In terms of the bias that we would expect each model to achieve, from least to greatest, these models would be ranked in the following order: model iii, model ii, and model i. With only one feature and the assumption of a linear relationship, model i is the least flexible and contains the strongest assumptions, thus we would expect the average prediction to differ quite a bit from its true value, and thus this model would have high bias, and furthermore the highest expected bias out of the three models. Model ii, on the other hand, has three features, but still includes the strong linear relationship assumption, so we would expect this model to have a fair amount of bias, but still less than that of model i. Finally, model iii includes more predictors than model i, while at the same time being more flexible than model ii. Thus, model iii has the best chance of capturing intricate relationships in the data, and thus we'd expect it to have the lowest bias out of these three models.

We will now rank the models with respect to variance. In terms of the variance that we would expect each model to achieve, from least to greatest, these models would be ranked in the following order: model i, model ii, and model iii. As seen in the bias-variance tradeoff equation, we'd expect this order to be the exact opposite of the one for the bias metric. More concretely, we'd expect the most complex and flexible model to have the highest variance, and in this case that's model iii, the polynomial model of degree 10. Furthermore, the linear model that includes more predictors would be expected to have the higher variance, which in this case is model ii with three predictors. That leaves us with the most simple and rigid model, model i, the

linear model with one variable. We would expect this model to have the lowest variance out of these three models.

Lastly, we will now rank the models with respect to the training MSE. In terms of the training MSE that we would expect each model to achieve, from least to greatest, these models would be ranked in the following order: model iii, model ii, and model i. The reasoning for this is very similar to the reasoning for the bias ranking. As flexibility increases, the training MSE is always expected to decrease. Since model i only contains one feature and is the least flexible out of the three models, we expect it to underfit the data and lead to a relatively high training MSE, especially compared to the other two models. Model ii, on the other hand, has three features, which offers a bit more flexibility to understand the true relationship between the features and response, leading to a lower training MSE than model i. Lastly, since model iii is the most complex and flexible of all of the models, we expect this model to be able to understand the intricate details in the model, and thus perform better on data its already seen (the training data). Therefore model iii is expected to have the lowest training MSE out of the three models.

Unlike bias, variance, and the training MSE, which have direct relationships with the flexibility of a model, the testing MSE does not have a direct relationship with the flexibility of a model. Instead, how the testing MSE fluctuates with flexibility requires knowledge on the true relationship of the predictors with the response. Therefore, it is not possible, with any degree of certainty, to rank models in terms of testing MSE. However, given the nature of three models, is is possible to make an educated guess. For example, in our case, we have three models that are pretty distinct. Namely, we have a model that is almost as rigid as can be; a linear model with one predictor. We would expect this model to underfit almost any training set it is given. On the other hand, we have a model that is very flexible; a degree ten quadratic model with three predictors. We would expect this model to overfit almost any training set it is given. Lastly, we have a model somewhere in between each of these models; a linear model with three predictors. Therefore, with the knowledge of these three models, if I were forced to rank these models in terms of the testing MSE I would rank them, from least to greatest, as follows: model ii, model i, and model iii.

Exercise 3

In this exercise, we will consider the prediction of the k-nearest neighbor classifier $\hat{y}(x)$, which is defined as $\hat{y}(x) = \text{argmax}_j \hat{P}(y = j|X = x)$. The estimated probability, $\hat{P}(y = j|X = x)$ is defined as $\hat{P}(y = j|X = x) = \frac{1}{k}\sum_{i \in \mathcal{N}_k(x)} I(y_i = j)$, where $\mathcal{N}_k(x)$ are the indices of the k closest points to x in terms of the Manhattan distance. The Manhattan distance between two points $a = (a_1, a_2, \ldots, a_m)$ and $b = (b_1, b_2, \ldots, b_m)$ is defined as $||a - b||_1 = \sum_\ell^m |a_\ell - b_\ell|$. To make a prediction one would simply find the k closest points. Out of these k nearest points, the most common label is declared our prediction.

In particular, in this exercise, we are looking at an animal shelter that holds an annual cat adoption event, in which they bring the cats that they predict will be adopted. The animal shelter claims that each cat's age, number of previous homes, and months in the shelter influence whether they will be adopted. The animal shelter is trying to decide whether to bring Catlord to the adoption event. Catlord is 6 months old (0.5 years), has had 0 previous homes, and has been in the animal shelter for 6 months. Using a k-nearest neighbors approach, we will help predict whether Catlord will be adopted or not. In this exercise, we will consider the following list of cats the animal shelter brought to the adoption event in previous years.

Cat Number	Age in years (x_1)	Number of previous homes (x_2)	Months in shelter (x_3)	Adopted (y)
1	2	2	4	No
2	4	2	1	Yes
3	5	3	1	Yes
4	3	1	2	No
5	1	1	4	Yes
6	7	3	6	No

Part a

In this part of the exercise, we will compute the Manhattan distance between Catlord and all of the other cats. Catlord's point is defined as $(x_1 = 0.5, x_2 = 0, x_3 = 6)$. Below we will calculate the Manhattan distance between Catlord and all of the other cats.

The Manhattan distance between Catlord and cat number one is

$$|0.5 - 2| + |0 - 2| + |6 - 4| = |-1.5| + |-2| + |2| = 1.5 + 2 + 2 = \boxed{5.5}$$

The Manhattan distance between Catlord and cat number two is

$$|0.5 - 4| + |0 - 2| + |6 - 1| = |-3.5| + |-2| + |5| = 3.5 + 2 + 5 = \boxed{10.5}$$

The Manhattan distance between Catlord and cat number three is

$$|0.5 - 5| + |0 - 3| + |6 - 1| = |-4.5| + |-3| + |5| = 4.5 + 3 + 5 = \boxed{12.5}$$

The Manhattan distance between Catlord and cat number four is

$$|0.5 - 3| + |0 - 1| + |6 - 2| = |-2.5| + |-1| + |4| = 2.5 + 1 + 4 = \boxed{7.5}$$

The Manhattan distance between Catlord and cat number five is

$$|0.5 - 1| + |0 - 1| + |6 - 4| = |-0.5| + |-1| + |2| = 0.5 + 1 + 2 = \boxed{3.5}$$

The Manhattan distance between Catlord and cat number six is

$$|0.5 - 7| + |0 - 3| + |6 - 6| = |-6.5| + |-3| + |0| = 6.5 + 3 = \boxed{9.5}$$

Part b

In this part of the exercise, with $k = 1$, we will determine what our prediction is. Since $k = 1$ implies that we find the nearest neighbor, it follows that cat number five will determine our prediction. Based on cat

number five, our nearest neighbor, the most common label is $y =$ Yes. Therefore, our prediction is "Yes," Catlord will be adopted.

Part c
In this part of the exercise, with $k = 3$, we will determine what our prediction is. Since $k = 3$ implies that we find the three nearest neighbor, it follows that cat number five, one and four will determine our prediction. Based on cat number five, four, and one, our three nearest neighbors, the three labels are "Yes," "No," and "No." This means that the most common label within our three nearest neighbors is $y =$ No. Therefore, our prediction is "No," Catlord will not be adopted.

Part d
In this part of the exercise, we will suppose we were told that the Bayes decision boundary is approximately linear. Our goal is to decide if we should choose $k = 1$ or $k = 3$ for our k-nearest neighbors classifier. This decision is made and elaborated upon below.

If the Bayes decision boundary was truly linear, then we would want to choose $k = 3$ for k-nearest neighbors classifier. The rationale behind this is that small values of k lead to overfitting. This overfitting happens because, for small values of k, the classifier would learn very specific patterns and discrepancies in the data in order to create the most accurate classifier on the training data. As shown on slide 29 of the Chapter 2 notes, when $k = 1$, a non-linear k-nearest neighbors boundary is common, especially one with irregular parts of the curve, such as distinct circles around specific points. Therefore, if the decision boundary is truly linear, we would want to choose the higher value of k, which in this case is $k = 3$.

On the other hand, if the Bayes decision boundary was highly non-linear, then we would want to choose $k = 1$ (although this could be a bit extreme in some cases). The rationale behind this is that small values of k, as mentioned above, lead to extremely non-linear k-nearest neighbors boundaries. Therefore, if the decision boundary is truly highly non-linear, we would want to choose the smaller value of k, which in this case is $k = 1$. However, most of the times there are individual circles in the boundary for $k = 1$ which are almost certainly not apart of the true boundary. Thus, in some situations a case could be made that $k = 3$ is better because it is still relatively small. However, in this case $k = 3$ is not small relative to the number of total neighbors that we have, so we will stick with the smallest number $k = 1$ for use in our k-nearest neighbors classifier.

Part e
In this part of the exercise, we will explain what happens to the k-nearest neighbors classifier predictions in the case where not all of the numerical features are approximately on the same scale.

In the case where not all of the numerical features are approximately on the same scale, the k-nearest neighbors classifier would be heavily influenced by the variable on the larger scale, even if the variable on the larger scale isn't truly more important. The reason why the variable on the larger scale is more influential is because that variable has a higher probability of being larger than the rest of the variables, which means its distance from other points has a higher chance of being larger than variables on smaller scales. Since the variable on the larger scale is more volatile, these larger differences can make usually large differences on smaller scales seem minute. This usually amounts to the omission of points from being apart of the k-nearest neighbors rather than the other way around.

Part f
In this part of the exercise, we will suppose that we found out that the animal shelter also uses breed and gender information to make their decision. We will further consider the fact that breed and gender are categorical variables. Furthermore, the animal shelter encodes a number for each gender and each breed. The goal of this part of the exercise is to highlight the mathematical and ethical concerns that this problematic model raises. This is done below.

Starting with the mathematical problems, the most apparent problem is that numerical labeling of categorical variables does not mean anything mathematically. In particular, these numbers are simply just a different way to label things, they don't imply any notion of a distance between them, especially when the data isn't ordinal. This implies that computing the Manhattan distance with the inclusion of these categorical variables is the incorrect thing to do (since they are merely categories). However, by displaying the categories of these

categorical variables as numbers along with our already numerical data, one could confuse these categorical variables for quantitative variables and accidentally compute the Manhattan distance with these variables included. In the case of the gender variable (which is binary in our problem), the Manhattan distance would make more sense in the fact that if it is 0 the label is the same, and if it is 1 the label is different. However, this distance of 1 does not have the same meaning as the distance of 1 in the continuous sense. Furthermore, nothing is stopping us from using 1 and 7 to denote these two groups (since they are in fact categories and not supposed to be interpreted as numbers). Lastly, in the case where more than one category exists (like breed type), taking the Manhattan distance with the inclusion of these values is even more problematic. For example, say we have group 1 and 8, the distance between these two points is 7, but if we instead had group 1 and 2, the distance would only be 1. In both cases the groups are different, but one contributes 7 in distance and the other one contributes only 1 in distance. In conclusion, there is no way to quantify the distance between different groups using the Manhattan distance, and doing so erroneously could lead to incorrect decisions.

However, using gender and breed also imposes ethical concerns. For example, if the Manhattan distances were erroneously calculated to include the categorical variables, then depending on the specific numbering given to certain breeds, when finding the k-nearest neighbors the breeds with larger numbers will often not be paired with the breeds with lower numbers (due to the large "distance" between the two groups). Therefore, our model could impose systematic biases that could lead to the discrimination of certain groups due to this mistake. In such a high stakes operation like pet adoption, the inclusion of these variables could be the difference between a cat being adopted or not. This is why, when it comes to hiring algorithms and other predictive models, it is often not a good idea to use metrics such as these that could explicitly lead to bias and discrimination in our results.

Exercise 4

In this exercise, we will assume that we are given n pairs of data points $\{(x_1, y_1), \ldots, (x_n, y_n)\}$. We want to fit a linear model to predict y_i using x_i, specifically we will use the model $\hat{y}_i = \beta_0 + \beta_1 x_i$. Furthermore, the residual sum of squares is given by $RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$. By minimizing RSS, our goal is to show that the least squares approach to choosing β_0 and β_1 gives us

$$\beta_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$
$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

In order to find the β_0 and β_1 that minimize RSS, we must use the fact that $\hat{y}_i = \beta_0 + \beta_1 x_i$ to rewrite RSS as

$$\begin{aligned} RSS &= \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \\ &= \sum_{i=1}^{n}(y_i - (\beta_0 + \beta_1 x_i))^2 \\ &= \sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2 \end{aligned}$$

Now, in order to minimize RSS in terms of β_0 and β_1, we must first find the partial derivatives of RSS with respect to β_0 and β_1. We will start with finding $\frac{\partial RSS}{\partial \beta_0}$ below.

$$\begin{aligned} \frac{\partial RSS}{\partial \beta_0} &= \frac{\partial}{\partial \beta_0}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2 \\ &= 2\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i) \cdot (-1) \\ &= -2\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i) \end{aligned}$$

As shown above, $\frac{\partial RSS}{\partial \beta_0} = -2\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)$. We will now find $\frac{\partial RSS}{\partial \beta_1}$ below.

$$\begin{aligned} \frac{\partial RSS}{\partial \beta_1} &= \frac{\partial}{\partial \beta_1}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2 \\ &= 2\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i) \cdot (-x_i) \\ &= -2\sum_{i=1}^{n}(y_i x_i - \beta_0 x_i - \beta_1 x_i^2) \end{aligned}$$

As shown above, $\frac{\partial RSS}{\partial \beta_1} = -2\sum_{i=1}^{n}(y_i x_i - \beta_0 x_i - \beta_1 x_i^2)$. We will now set the partial derivatives of RSS with respect to β_0 and β_1 equal to zero and solve for β_0 and β_1, respectively. We will start with solving for β_0

below.

$$
\begin{aligned}
-2\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i) &= 0 \\
\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i) &= 0 \\
\sum_{i=1}^{n} y_i - \beta_0 \sum_{i=1}^{n} - \beta_1 \sum_{i=1}^{n} x_i &= 0 \\
n\bar{y} - \beta_0 n - \beta_1 n\bar{x} &= 0 \quad (\text{since } \sum_{i=1}^{n} 1 = n \text{ and } \sum_{i=1}^{n} x_i = n\bar{x}) \\
n\bar{y} - \beta_1 n\bar{x} &= \beta_0 n \\
\beta_0 &= \bar{y} - \beta_1 \bar{x}
\end{aligned}
$$

As shown above, $\beta_0 = \bar{y} - \beta_1 \bar{x}$. We will now solve for β_1 below. It is important to note that in the final step we will use the following fact

$$
\frac{\frac{1}{n}\sum_{i=1}^{n} x_i y_i - \bar{y}\bar{x}}{\frac{1}{n}\sum_{i=1}^{n} x_i^2 - \bar{x}^2} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}
$$

Now that we have that out of the way, we can begin with solving for β_1, this is done below.

$$
\begin{aligned}
-2\sum_{i=1}^{n}(y_i x_i - \beta_0 x_i - \beta_1 x_i^2) &= 0 \\
\sum_{i=1}^{n}(y_i x_i - \beta_0 x_i - \beta_1 x_i^2) &= 0 \\
\sum_{i=1}^{n} y_i x_i - \beta_0 \sum_{i=1}^{n} x_i - \beta_1 \sum_{i=1}^{n} x_i^2 &= 0 \\
\sum_{i=1}^{n} y_i x_i - \beta_0 n\bar{x} - \beta_1 \sum_{i=1}^{n} x_i^2 &= 0 \quad (\text{since } \sum_{i=1}^{n} x_i = n\bar{x}) \\
\sum_{i=1}^{n} y_i x_i - (\bar{y} - \beta_1 \bar{x}) n\bar{x} - \beta_1 \sum_{i=1}^{n} x_i^2 &= 0 \quad (\text{since } \beta_0 = \bar{y} - \beta_1 \bar{x}) \\
\sum_{i=1}^{n} y_i x_i - n\bar{y}\bar{x} + n\beta_1 \bar{x}^2 - \beta_1 \sum_{i=1}^{n} x_i^2 &= 0 \\
\frac{1}{n}\sum_{i=1}^{n} y_i x_i - \bar{y}\bar{x} + \beta_1 \bar{x}^2 - \beta_1 \frac{1}{n}\sum_{i=1}^{n} x_i^2 &= 0 \\
\beta_1 \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \beta_1 \bar{x}^2 &= \frac{1}{n}\sum_{i=1}^{n} y_i x_i - \bar{y}\bar{x} \\
\beta_1 \left(\frac{1}{n}\sum_{i=1}^{n} x_i^2 - \bar{x}^2 \right) &= \frac{1}{n}\sum_{i=1}^{n} y_i x_i - \bar{y}\bar{x} \\
\beta_1 &= \frac{\frac{1}{n}\sum_{i=1}^{n} y_i x_i - \bar{y}\bar{x}}{\frac{1}{n}\sum_{i=1}^{n} x_i^2 - \bar{x}^2} \\
\beta_1 &= \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \quad (\text{using the hint given in the problem description})
\end{aligned}
$$

As shown above, $\beta_1 = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=1}^{n}(x_i-\bar{x})^2}$. Since RSS is convex in β_0 and β_1, it follows that the values for β_0 and β_1 that we have obtained, are in fact minimizers (without the need for the second derivative test). Therefore we have shown that, by minimizing RSS, the least squares approach to choosing β_0 and β_1 gives us

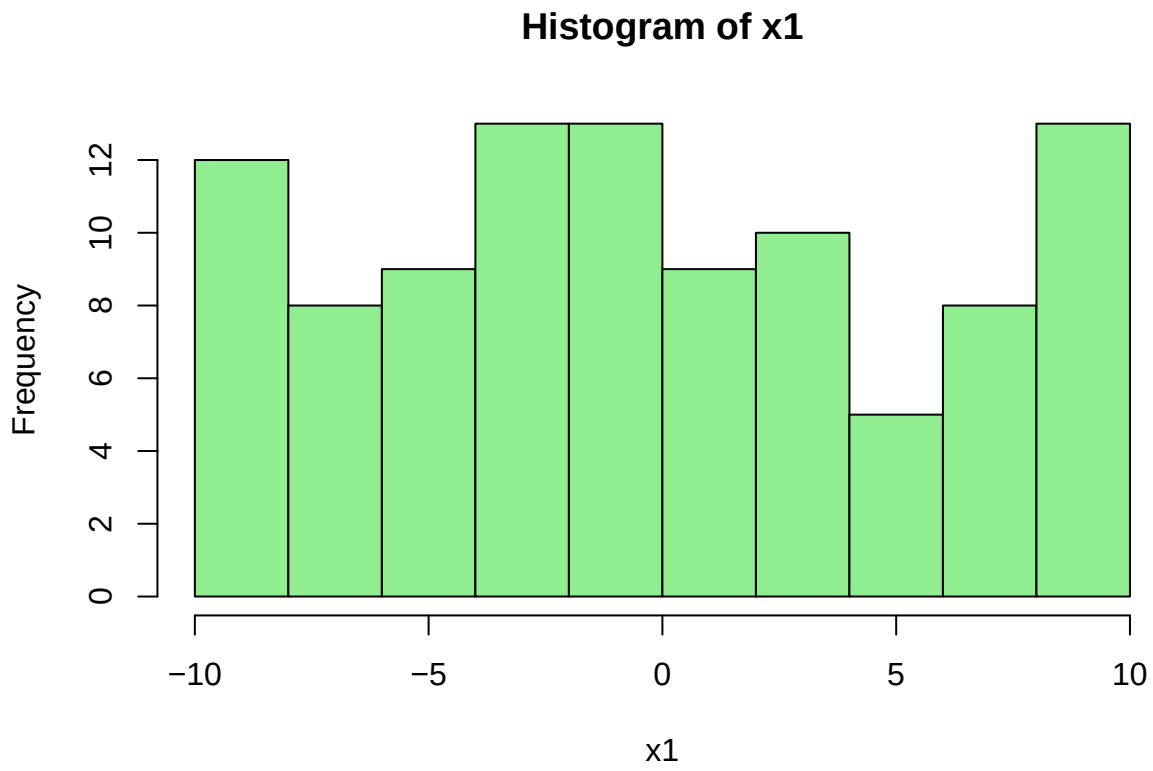
$$\beta_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$
$$\beta_0 = \bar{y} - \beta_1\bar{x}$$

These values are called the OLS estimators of β_0 and β_1.

Exercise 5
In this exercise, we will work with simulated data and explore some linear regression models.

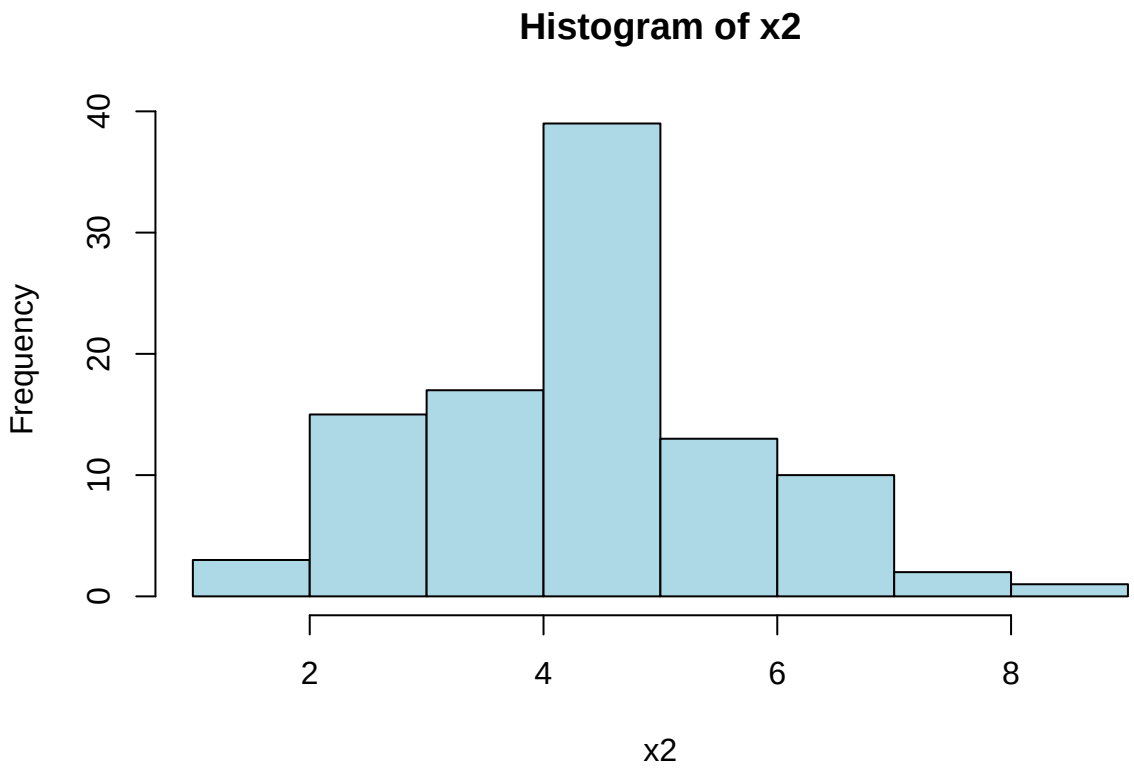
Part a
In this part of the exercise, we will generate our independent variables. In particular we will create a vector x_1 which contains 100 realizations from a $Unif(-10, 10)$ distribution, and a vector x_2 which contains 100 realizations from a $Gamma(\alpha = 9, \beta = 2)$ distribution (where α is the shape parameter and β is the rate parameter). Furthermore, we will showcase these vectors through the creation of histograms. This is done below.

```
# Set the seed for reproducibility
set.seed(333)

# Create 100 realizations of a Unif(-10, 10) distribution
x1 <- runif(n=100, min=-10, max=10)

# Create 100 realizations of a Gamma(9, 2) distribution
x2 <- rgamma(n=100, shape=9, rate=2)

# Create and plot the histograms of x1
hist(x1, xlab = "x1", main="Histogram of x1", col = "lightgreen")
```



As can be seen from the above histogram of x_1, the 100 realizations from a $Unif(-10, 10)$ distribution, as expected, the histogram roughly looks like a $Unif(-10, 10)$ distribution.

```r
# Create and plot the histograms of x2
hist(x2, xlab = "x2", main="Histogram of x2", col = "lightblue")
```

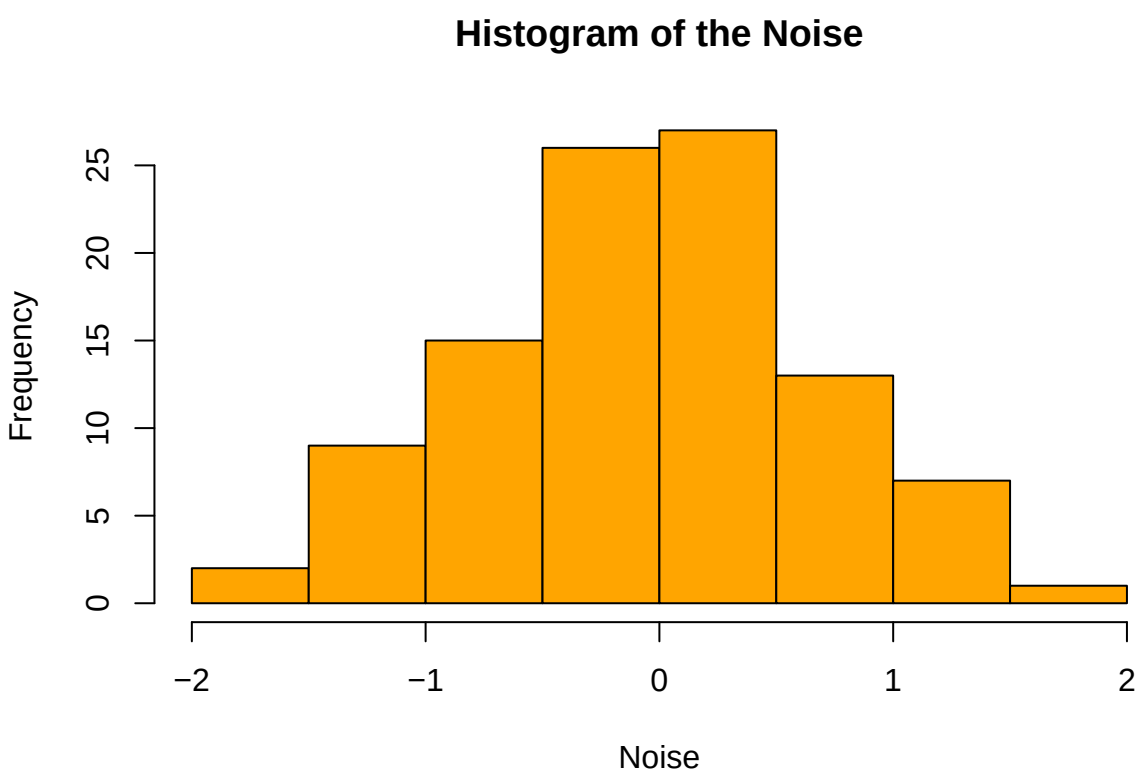


As can be seen from the above histogram of x_2, the 100 realizations from a $Gamma(\alpha = 9, \beta = 2)$ distribution, we can see that the histogram is unimodal (which is expected since $\alpha > 1$). Therefore, as expected, the histogram roughly looks like a $Gamma(\alpha = 9, \beta = 2)$ distribution.

Part b

In this part of the exercise, we will now generate the error/noise terms. In particular, we will create a vector ϵ that contains 100 realizations from a $N(0, \sigma^2)$ distribution. Specifically we will take $\sigma^2 = \frac{1}{2}$. Furthermore we will showcase this vector through the creation of a histogram. This is done below.

```r
# Create 100 realizations of a Norm(0, sqrt(1/2)) distribution
epsilon <- rnorm(n=100, mean=0, sd=sqrt(1/2))

# Create and plot the histograms of epsilon
hist(epsilon, xlab = "Noise", main="Histogram of the Noise", col = "orange")
```

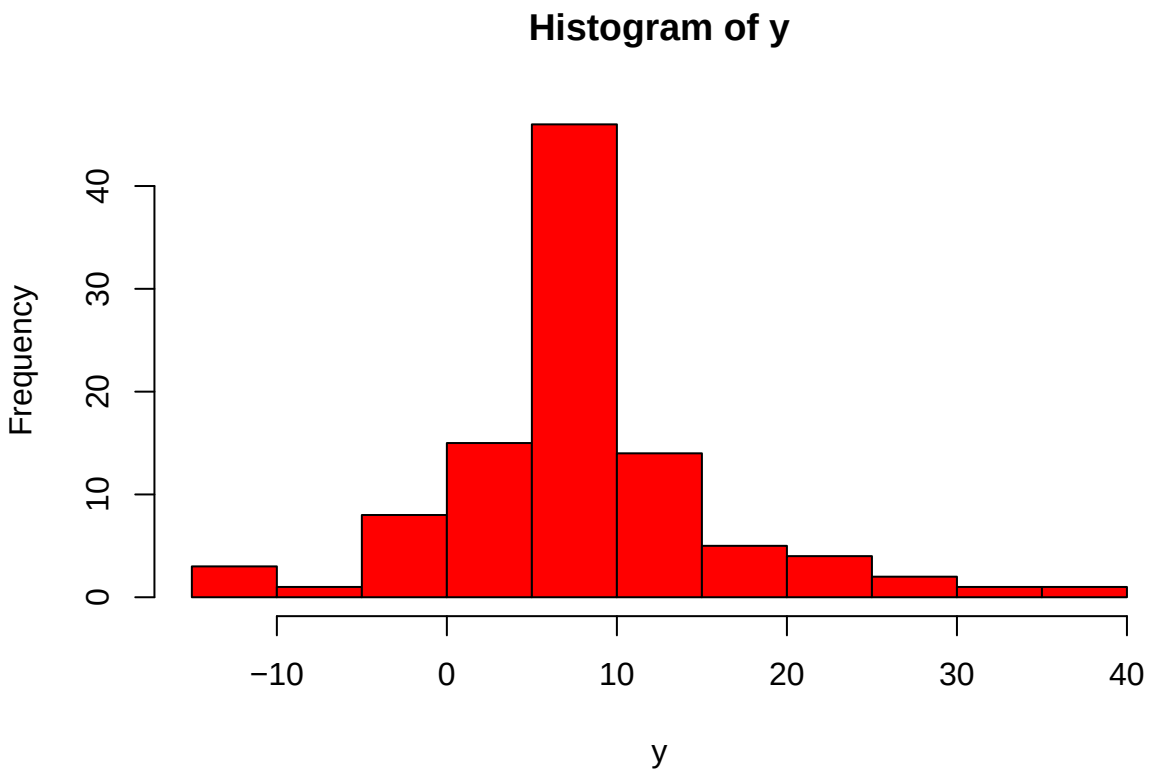


As can be seen from the above histogram of ϵ, the 100 realizations from a Norm$(\mu = 0, \sigma^2 = 1/2)$ distribution, we can see that the histogram is centered around the mean value of 0, with a relatively small spread/deviation. Therefore, as expected, the histogram roughly looks like a Norm$(\mu = 0, \sigma^2 = 1/2)$ distribution.

Part c

In this part of the exercise, we will compute the response variable y using the following function: $y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i1} x_{i2} + \epsilon$. In particular, we will use $\beta_0 = 1$, $\beta_1 = -3$, $\beta_2 = 2$, and $\beta_3 = 0.85$. Furthermore we will showcase this response vector through the creation of a histogram. This is done below.

```
# Create the response vector y based on the given formula and parameters
y <- 1+(-3*x1)+(2*x2)+(0.85*x1*x2)+epsilon

# Create and plot the histograms of epsilon
hist(y, xlab = "y", main="Histogram of y", col = "red")
```



As can be seen from the above histogram of the response variable y, it appears as if the histogram is symmetric about a value of around 8 or 9. This makes sense because if our error/noise terms are normally distributed, then this would imply that our response variable y should be normally distributed as well (since our predictors are assumed to be fixed).

Part d

In this part of the exercise, we will fit a least squares model to predict y using only x_1 and x_2. Using R to fit this model we will compare our estimated coefficients with the true coefficients by printing out the model summary. This is done in R below.

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.3215  -2.3938   0.9027   2.9634  17.7566
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.3847     2.2094   0.174 0.862147
## x1            0.8695     0.1139   7.634 1.59e-11 ***
## x2            1.8194     0.4833   3.764 0.000286 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.463 on 97 degrees of freedom
```

```
## Multiple R-squared:  0.3951, Adjusted R-squared:  0.3827
## F-statistic: 31.68 on 2 and 97 DF,  p-value: 2.573e-11
```

We can compare the coefficients in this estimated model with the coefficients in the true model by observing the difference between the estimated coefficients and the true coefficients. Since this model does not include an interaction term, and thus only contains 3 coefficients, we cannot compare β_3 with anything, so we will only compare the intercept along with the other two parameters. In the fitted model above, the intercept was estimated to be $\hat{\beta}_0 = 0.3847$, which is 0.6153 units smaller than the true value of the intercept which was $\beta_0 = 1$. Furthermore, in the fitted model above, the estimated coefficient of the x_1 vector was $\hat{\beta}_1 = 0.8695$, which is 3.8695 units larger than the true value of the coefficient which was $\beta_1 = -3$. Lastly, in the fitted model above, the estimated coefficient of the x_2 vector was $\hat{\beta}_2 = 1.8194$, which is 0.1806 units smaller than the true value of the coefficient which was $\beta_2 = 2$. As expected, due to the omission of the interaction term, which was apart of the true model, there are some noticeable differences between the true parameter values and the estimated parameter values.

Part e

In this part of the exercise, we will fit a least squares model to predict y using x_1, x_2, x_1^2, x_2^2, and x_1x_2. Using R to fit and display the results of this model we will: comment on the fit by comparing it to the model in (d), compare our estimated coefficients with the true coefficients, as well as assess the feasibility of dropping x_1^2 and x_2^2 based on their p-values. This is done in R below.

```
##
## Call:
## lm(formula = y ~ x1 + x2 + I(x1^2) + I(x2^2) + x1:x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.58324 -0.40632  0.00587  0.43066  1.81174
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.3492757  0.6058086   0.577    0.566
## x1          -2.9247809  0.0490629 -59.613  < 2e-16 ***
## x2           2.2508180  0.2725084   8.260 9.08e-13 ***
## I(x1^2)     -0.0003045  0.0023397  -0.130    0.897
## I(x2^2)     -0.0251918  0.0292544  -0.861    0.391
## x1:x2        0.8348163  0.0103474  80.679  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7318 on 94 degrees of freedom
## Multiple R-squared:  0.9925, Adjusted R-squared:  0.9921
## F-statistic:  2482 on 5 and 94 DF,  p-value: < 2.2e-16
```

We will start by commenting on the fit of this model by comparing it to the model in (d). In the model from part (d), the adjusted R^2 value was 0.3827. This means that, adjusting for the number of parameters, 38.27% of the variability in y is being explained by the predictors. In the new model, the adjusted R^2 value was 0.9921. This means that, adjusting for the number of parameters, 99.21% of the variability in y is being explained by the predictors. It is obvious that this new model is explaining more of the variability of y, even accounting for the fact that it has more predictors. Furthermore, the model from part (d) has a residual standard error of 6.463, while the new model has a residual standard error of 0.7318. This implies that the predictions for the new model are more precise than the predictions from the model in part (d). However, it is important to notice that, as expected, the x_1^2 and x_2^2 terms in the new models are not significant predictors at any reasonable significance level (which is expected since they aren't included in the true model).

We will now compare the coefficients in this new estimated model with the coefficients from the true model by observing the difference between the estimated coefficients and the true coefficients. Since this new model includes extra terms such as x_1^2 and x_2^2, we cannot compare the coefficients of these terms with anything, so we will only compare the intercept along with the other three parameters. In the fitted model above, the intercept was estimated to be $\hat{\beta}_0 = 0.3492757$, which is 0.6507243 units smaller than the true value of the intercept which was $\beta_0 = 1$. Furthermore, in the fitted model above, the estimated coefficient of the x_1 vector was $\hat{\beta}_1 = -2.9247809$, which is 0.0752191 units larger than the true value of the coefficient which was $\beta_1 = -3$. Also, in the fitted model above, the estimated coefficient of the x_2 vector was $\hat{\beta}_2 = 2.2508180$, which is 0.250818 units larger than the true value of the coefficient which was $\beta_2 = 2$. Lastly, in the fitted model above, the estimated coefficient of the x_1x_2 vector was $\hat{\beta}_3 = 0.8348163$, which is 0.0151837 units smaller than the true value of the coefficient which was $\beta_3 = 0.85$. As expected, due to the inclusion of two terms that were not apart of the true model, there are some differences between the true parameter values and the estimated parameter values. However, it is important to note that some of these coefficients are very close to there counterparts in the true model. Furthermore, in comparison to the model in part (d), the coefficients are much closer to their true values (with the exception of the intercept, which is not significant in this model, as opposed to model (d)).

As can be seen from the above model fit, the p-values associated with the t-tests for the coefficients of x_1^2 and x_2^2 were 0.897 and 0.391, respectively. At any reasonable significance level, we would come to the conclusion that these variables are insignificant predictors of y and add little to the model with their inclusion. Due to these high p-values, dropping either variable could be justified. However, due to the fact that the coefficients in multiple linear regression are calculated simultaneously, it would be hard to justify omitting both at the same time because removing one variable could change the p-values of the other variables drastically when compared to the previous fit. Instead, the better practice would be to drop one predictor at a time, and then analyze the new model to see if the other predictor is still insignificant. To emphasize this rationale, we will drop x_1^2 from the model (since it has the higher p-value of the two variables we are considering), and rerun the model to check if x_2^2 is still insignificant. This is done below.

```
##
## Call:
## lm(formula = y ~ x1 + x2 + I(x2^2) + x1:x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.58736 -0.41050  0.00654  0.42639  1.82058
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.35128    0.60247   0.583    0.561
## x1          -2.92588    0.04809 -60.846  < 2e-16 ***
## x2           2.24628    0.26887   8.355 5.36e-13 ***
## I(x2^2)     -0.02482    0.02896  -0.857    0.394
## x1:x2        0.83502    0.01018  82.064  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.728 on 95 degrees of freedom
## Multiple R-squared:  0.9925, Adjusted R-squared:  0.9922
## F-statistic:  3136 on 4 and 95 DF,  p-value: < 2.2e-16
```

In this new model, after dropping x_1^2, we see that x_2^2 is still insignificant at any reasonable significance level due to its p-value of 0.394. This shows that it can be dropped as well. This offers some validation to the idea of dropping both of the predictors at the same time, however, this result won't always be the case when dropping one variable at a time. Alternatively, it is the best practice to use an ANOVA F-test to compare the models with and without x_1^2 and x_2^2 before dropping them. We will do this below

```
## Analysis of Variance Table
##
## Model 1: y ~ x1 + x2 + x1:x2
## Model 2: y ~ x1 + x2 + I(x1^2) + I(x2^2) + x1:x2
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     96 50.744
## 2     94 50.345  2   0.39827 0.3718 0.6905
```

Since the p-value of the ANOVA test was 0.6905, we failed to reject the null hypothesis and conclude that x_1^2 and x_2^2 were simultaneously insignificant at any reasonable significance level. It is this test alone that validates dropping both predictors at the same time (it would be wrong to do so without doing this test first).

Part f

In this part of the exercise, we will perform an ANOVA F-test in R using the `anova` function, in order to identify which of the two models (d) or (e) fit the data best. To do this we will report the F-statistic, relevant degrees of freedom, and the p-value. After we have selected the best model, we will interpret the coefficients of the best model.

Based on the STAT 423 notes on the ANOVA F-test, since we are comparing the model

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}$$

to a larger model containing 3 more parameters, defined as

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i1}^2 + \beta_4 x_{i2}^2 + \beta_5 x_{i1} x_{i2}$$

Therefore our competing hypotheses are

$$H_0 : \beta_3 = \beta_4 = \beta_5 = 0$$
$$H_1 : \beta_j \neq 0 \text{ for at least one } j \in \{3, 4, 5\}$$

In essence, the null hypothesis is saying that the more complex model does not add any variables which lead to an improved model fit, and the alternative hypothesis is saying that the complex model is adding at least one variable that improves the model fit. In the latter case the more complex model is selected, and in the latter case the simpler model is selected. We will now run this test using the `anova` function in R, the output is shown below

```
## Analysis of Variance Table
##
## Model 1: y ~ x1 + x2
## Model 2: y ~ x1 + x2 + I(x1^2) + I(x2^2) + x1:x2
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1     97 4051.5
## 2     94   50.3  3    4001.2 2490.2 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The F-statistic from the above F-test is 2490.2, which is very large. Since the p-value is calculated from a $F_{p-q,n-p-1}$ distribution, it follows that $n - p - 1 = 97$ represents the denominator degrees of freedom (which is also the degrees of freedom of the smaller model). Furthermore, $p - q = 97 - 94 = 3$ represents the numerator degrees of freedom (which is also the difference of the degrees of freedom between the smaller and the larger model; i.e. 94 is the degrees of freedom of the larger model). Lastly, the p-value of the test was calculated as less than 2.2×10^{-16}, which is extremely small. Thus, at any reasonable significance level we

reject the null hypothesis and conclude that the more complex model did in fact add variables that improved the overall fit. Thus, using the output from part (e), our selected model is

$$y_i = 0.3492757 - 2.9247809x_{i1} + 2.2508180x_{i2} - 0.0003045x_{i1}^2 - 0.0251918x_{i2}^2 + 0.8348163x_{i1}x_{i2}$$

We will now interpret these estimated coefficients. Since there exists polynomial and interaction terms in the model, individually interpreting the coefficients associated with these terms is very tricky, so we will start by explaining what each coefficient means, and then interpret what happens with a one unit increase in x_1 and x_2, respectively.

Starting off with individually explaining what each coefficient means, it turns out that we can interpret $\hat{\beta}_0$. In particular, when x_1 and x_2 are equal to 0, the estimated average value for y is $\hat{\beta}_0 = 0.3492757$. Moving onto the more complicated coefficients, $\hat{\beta}_1 = -2.9247809$ and $\hat{\beta}_2 = 2.2508180$ are the linear/main effects, $\hat{\beta}_3 = -0.0003045$ and $\hat{\beta}_4 = -0.0251918$ are the quadratic effects, and $\hat{\beta}_5 = 0.8348163$ is the interaction effect.

Before we interpret what happens with a one unit increase in x_1 and x_2, we must first develop a general form for what happens when we increase x_1 or x_2 by one unit. With x_2 fixed and a given x_1, our model is $\hat{y}_1 = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_1^2 + \hat{\beta}_4 x_2^2 + \hat{\beta}_5 x_1 x_2$. Now increasing x_1 by one unit we obtain $x_1 + 1$, plugging this new value into the given model we obtain the following result

$$\begin{aligned}
\hat{y}_2 &= \hat{\beta}_0 + \hat{\beta}_1(x_1 + 1) + \hat{\beta}_2 x_2 + \hat{\beta}_3(x_1 + 1)^2 + \hat{\beta}_4 x_2^2 + \hat{\beta}_5(x_1 + 1)x_2 \\
&= \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3(x_1^2 + 2x_1 + 1) + \hat{\beta}_4 x_2^2 + \hat{\beta}_5(x_1 x_2 + x_2) \\
&= \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_1^2 + 2\hat{\beta}_3 x_1 + \hat{\beta}_3) + \hat{\beta}_4 x_2^2 + \hat{\beta}_5 x_1 x_2 + \hat{\beta}_5 x_2 \\
&= \left(\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_1^2 + \hat{\beta}_4 x_2^2 + \hat{\beta}_5 x_1 x_2\right) + \left(\hat{\beta}_1 + 2\hat{\beta}_3 x_1 + \hat{\beta}_3 + \hat{\beta}_5 x_2\right) \\
&= \hat{y}_1 + \left(\hat{\beta}_1 + 2\hat{\beta}_3 x_1 + \hat{\beta}_3 + \hat{\beta}_5 x_2\right)
\end{aligned}$$

Therefore, when holding x_2 constant in this model, while increasing x_1 by one unit, we estimate that y increases by $\hat{\beta}_1 + 2\hat{\beta}_3 x_1 + \hat{\beta}_3 + \hat{\beta}_5 x_2$ on average. Where x_1 is the original value of x_1 before the one unit increase, and x_2 is the constant value of x_2. This same result holds for x_2. **Note:** This calculation was based on a similar calculation done in quiz section, I am not 100% sure it is correct, but I thought I'd give it an attempt.

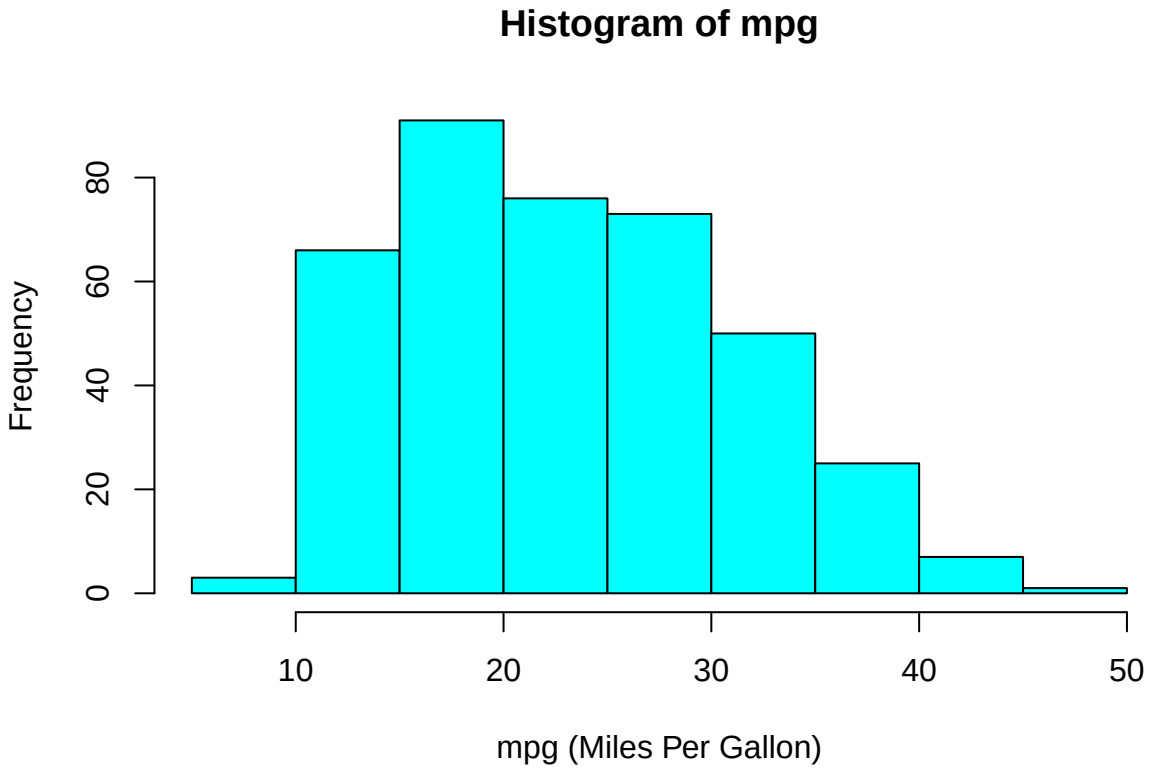
We will now move on to interpreting what happens with a one unit increase in x_1 and x_2, respectively. While holding x_2 constant, increasing x_1 by 1 unit, we estimate that y increases by $\hat{\beta}_1 + 2\hat{\beta}_3 x_1 + \hat{\beta}_3 + \hat{\beta}_5 x_2 = -2.9247809 + 2(-0.0003045)x_1 - 0.0003045 + 0.8348163x_2$ on average. Similarly, while holding x_1 constant, increasing x_2 by 1 unit, we estimate that y increases by $\hat{\beta}_2 + 2\hat{\beta}_4 x_2 + \hat{\beta}_4 + \hat{\beta}_5 x_1 = 2.2508180 + 2(-0.0251918)x_2 - 0.0251918 + 0.8348163x_1$ on average.

Exercise 6

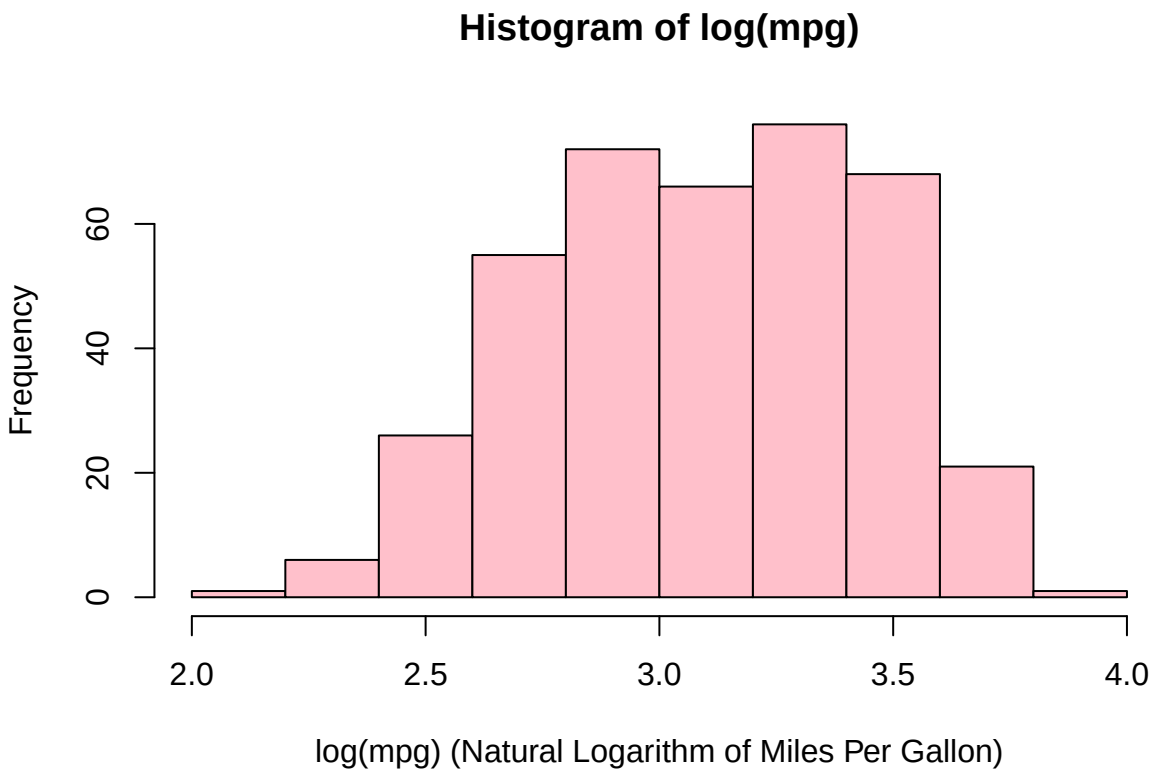
In this exercise, we will use the `Auto` dataset. This dataset contains several variables including `mpg` (miles per gallon), `cylinders` (number of cylinders), `displacement` (engine displacement), `horsepower`, `weight`, `acceleration`, and `year`. The goal of this exercise is to predict the mileage (miles per gallon) using other variables.

Part a

In this part of the exercise, we will plot the histogram of `mpg` and assess its normality. If we find that `mpg` does not appear to be normally distributed, we will log-transform `mpg` in order to get the linear regression working properly. Below we will display the histogram of `mpg` using R.



As can be seen from the above histogram of `mpg`, due to the apparent right skew in the histogram, it appears as if `mpg` is not normally distributed. Therefore, in order to attain normality and have the linear regression work properly, we will log-transform `mpg` and display its histogram. This is done in R below.

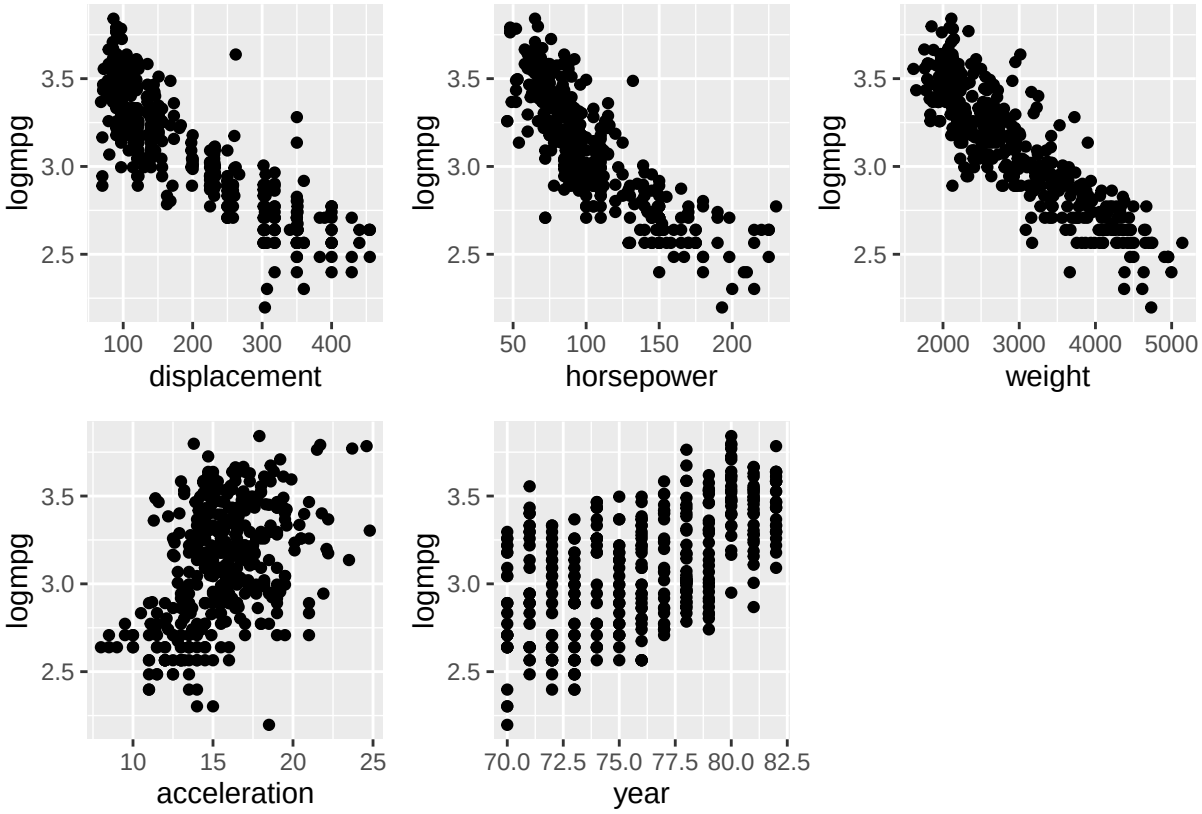


Based on the above histogram of `log(mpg)`, despite the presence of a small left skew in the data, we have more justification to claim that `log(mpg)` is normally distributed, especially in comparison to the histogram of `mpg` alone. Thus, we can use `log(mpg)` for the rest of the exercise in order to get the regression to work correctly.

Part b

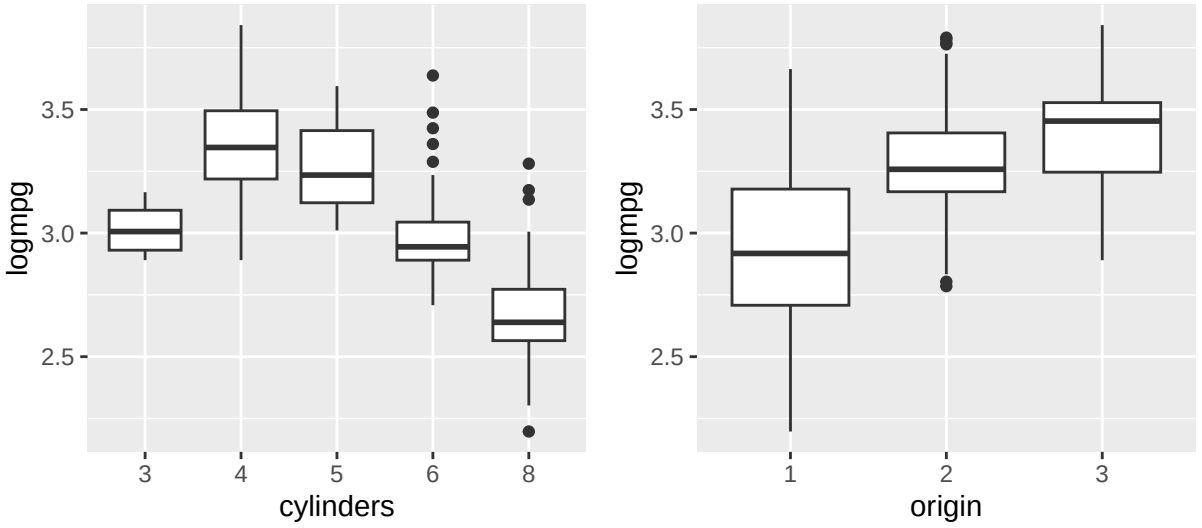
In this part of the exercise, we will generate pairwise scatterplots between `log(mpg)` and the other variables. In addition to this, we will also generate the correlation matrix between `log(mpg)` and the other variables. By analyzing these two metrics, we will decide which predictors seem relevant.

We will start by plotting the scatterplots between `log(mpg)` and the other variables individually so that we can better see the relationships between the response and the possible predictors. We won't be using a scatterplot matrix due to the large amount of variables that we have. It will be important to note that `origin` and `cylinders` are categorical variables, thus we will use boxplots to visualize these variables, instead of scatterplots. We omitted the `name` variable due to the large amount of categories (301 categories). These scatterplots were created using `ggplot2` and `patchwork` and are shown below.



Based on these pairwise scatterplots, we can see that variables such as `displacement`, `horsepower`, and `weight` have strong negative linear relationships with `logmpg`. Also, it appears as if `acceleration` has a moderate positive linear relationship with `logmpg`. Lastly, it appears as if `year` also has a moderate positive linear relationship with `logmpg`, this relationship would be stronger if the variance at each unique year was not as large as it is.

We will now use boxplots to analyze the relationship between `logmpg` and the two categorical variables `cylinders` and `origin`. Again, this is done below using `ggplot2` and `patchwork`.



Moving onto our categorical variables, it appears there might be a moderate relationship between `cylinders` and `logmpg`. In particular, as the number of cylinders increases it appears as if `logmpg` decreases. Lastly, due to the fact that there doesn't appear to be any order in the `origin` variable, we can't say as `origin` increases, `logmpg` increases. Instead we can only see that there are differences in the ranges of `logmpg` between the different levels of the `origin` variable (except between origin level 2 and 3, which would imply making origin level 1 the reference level if included in a linear model). Based on this visual analysis, it appears that the most relevant predictors would be `displacement`, `horsepower`, `weight`, and possibly `year`. Categorical predictors like `cylinders` and `origin` could also be relevant, but in comparison to the four continuous variables previously mentioned, they do not seem to be as relevant. Lastly, `acceleration` could also be relevant due to its possible linear relationship with `logmpg`, just not as relevant as these other variables.

We will now move onto looking at the correlations between the continuous variables and `logmpg`. The categorical variables will be omitted because the correlation coefficient between a categorical variable and a continuous variable does not exist. This correlation matrix will be displayed using R below.

```
##              displacement horsepower     weight acceleration       year
## displacement    1.0000000  0.8972570  0.9329944   -0.5438005 -0.3698552
## horsepower      0.8972570  1.0000000  0.8645377   -0.6891955 -0.4163615
## weight          0.9329944  0.8645377  1.0000000   -0.4168392 -0.3091199
## acceleration   -0.5438005 -0.6891955 -0.4168392    1.0000000  0.2903161
## year           -0.3698552 -0.4163615 -0.3091199    0.2903161  1.0000000
## logmpg         -0.8536910 -0.8301551 -0.8756582    0.4475743  0.5772748
##                  logmpg
## displacement -0.8536910
## horsepower   -0.8301551
## weight       -0.8756582
## acceleration  0.4475743
## year          0.5772748
## logmpg        1.0000000
```

Based on the above correlation matrix, `displacement`, `horsepower` and `weight` all have strong negative linear relationships with `logmpg`, as expected from the scatterplot analysis. Furthermore, based on the above correlation matrix, the `acceleration` variable has a moderate positive linear relationship with `logmpg`, as expected from the scatterplot analysis. Lastly, based on the above correlation matrix, `year` also has a moderate positive linear relationship with `logmpg` (although it has a greater linear relationship compared to `acceleration`). After this correlation matrix analysis, just as we concluded in the scatterplot analysis, it appears that the most relevant predictors would be `displacement`, `horsepower`, and `weight`, with `year` following shortly behind. However, in the next part, we will show why these variables may not be the best for a linear model predicting `logmpg`.

Part c

In this part of the exercise, based on the observations from part (b), we will choose three predictors to be included in a linear model. After this, we will build a linear model to predict `log(mpg)` and report the estimated coefficients.

Based on the analysis from part (b), due to there large negative correlations with `logmpg`, the three variables that I decided to choose were `displacement`, `horsepower` and `weight`. The correlation coefficients of these variables with `logmpg` were -0.8536910, -0.8301551, and -0.8756582, respectively. However, as can be seen from the above correlation matrix, the correlation coefficients between `displacement`, `horsepower` and `weight` are all large. This implies that there may be severe multicollinearity between the predictors in the model. Therefore, in order to check if this possible multicollinearity will be a problem in our fitted model, we will compute the variance inflation factor (VIF) of each of the predictors in our proposed model, this is done in R below using the `vif()` function.

```
## Auto$displacement   Auto$horsepower       Auto$weight
##         10.310539          5.287295          7.957383
```

As can be seen from the above output, all three predictors have a VIF greater than 5. Furthermore, the `displacement` predictor has a VIF greater than 10. This gives us evidence that there persists high multicollinearity in this model, and thus we will need to create a new model. This is done below.

First off, since we know there persists high multicollinearity between `displacement`, `horsepower` and `weight`, it would make sense to drop two of the predictors to ensure we are not introducing any multicollinearity into our model. In this case, we will drop `displacement` and `horsepower` from the model, and keep `weight` in our new model. Furthermore, as noted in the analysis from part (b), it would make sense to include `year` as a predictor due to its moderate positive linear relationship to `logmpg`. Lastly, it would also make sense to include `origin` as a predictor, since we won't have concerns of multicollinearity with its inclusion, and we are pretty sure that at all of the levels will be significant when including it in the model. We will compute the VIF of each of these predictors to ensure that our model does not include any multicollinearity. This is done below.

```
##                              GVIF Df GVIF^(1/(2*Df))
## Auto$weight              1.711383  1        1.308199
## Auto$year                1.128435  1        1.062278
## as.factor(Auto$origin) 1.612360  2        1.126848
```

As can be seen above, the generalized variance inflation factor (an extension of VIF to allow for categorical predictors) of each predictor is well below 5. Therefore we have evidence that no serious multicollinearity persists in our model. We will now fit this initial model and show the model output through the use of the `summary()` function in R. This is done below.

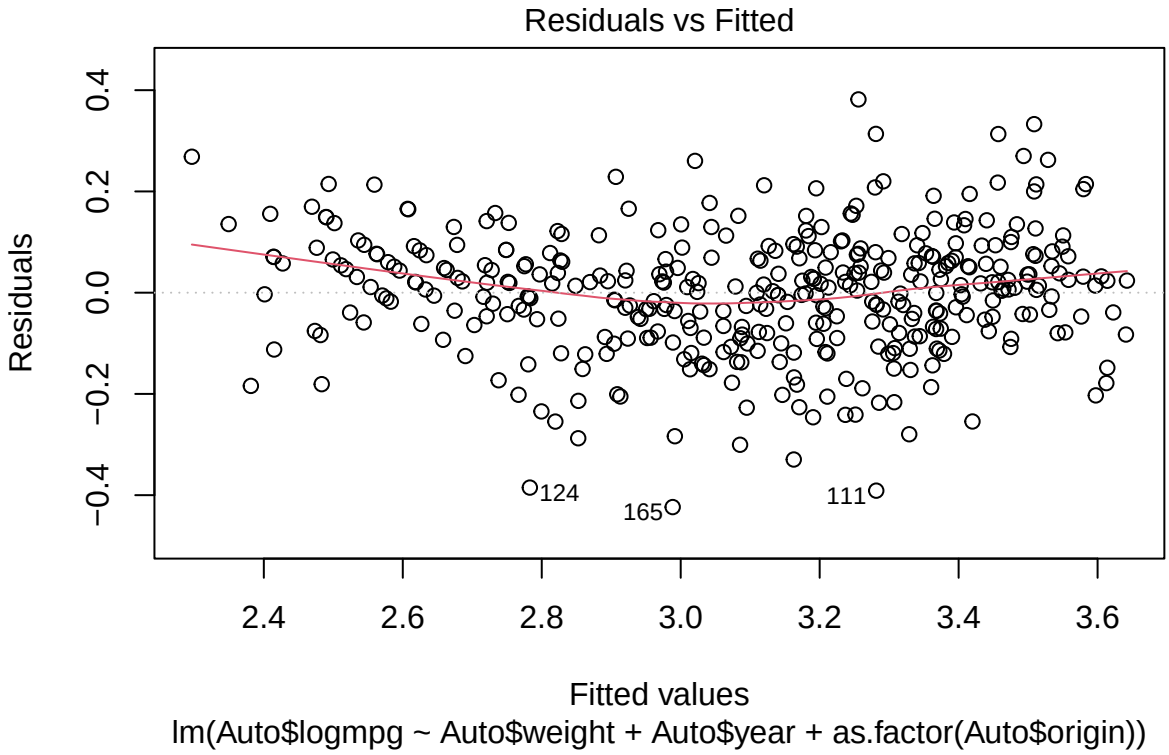
```
##
## Call:
## lm(formula = Auto$logmpg ~ Auto$weight + Auto$year + as.factor(Auto$origin))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.42352 -0.07273  0.01168  0.07197  0.38176
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)              1.504e+00  1.448e-01  10.386  < 2e-16 ***
## Auto$weight             -2.864e-04  9.370e-06 -30.565  < 2e-16 ***
## Auto$year                3.189e-02  1.754e-03  18.180  < 2e-16 ***
## as.factor(Auto$origin)2  7.031e-02  1.867e-02   3.766 0.000192 ***
## as.factor(Auto$origin)3  5.750e-02  1.870e-02   3.074 0.002259 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1203 on 387 degrees of freedom
## Multiple R-squared:  0.8761, Adjusted R-squared:  0.8748
## F-statistic: 684.2 on 4 and 387 DF,  p-value: < 2.2e-16
```

As can be seen above, all of the variables in our initial model are significant at any reasonable significance level (including all levels of our categorical predictor). Furthermore, our model contains a relatively low residual standard error of $\hat{\sigma} = 0.1203$, as well as a high adjusted R-squared value of $R^2_{adj} = 0.8748$. This means that, adjusting for the number of parameters, 87.48% of the variability in `logmpg` is being explained by the predictors.

Part d

In this part of the exercise, we will generate the residual plots for model diagnostics and check if any of our

assumptions are violated. If there are problems present in our model we will propose methods to fix them, and correct these issues via the use of our proposed methods (as well as displaying the output of our new model). If there are no issues, we will leave our model as is (and explicitly say so). We will start by checking the TA plot/residuals versus fitted plot below.

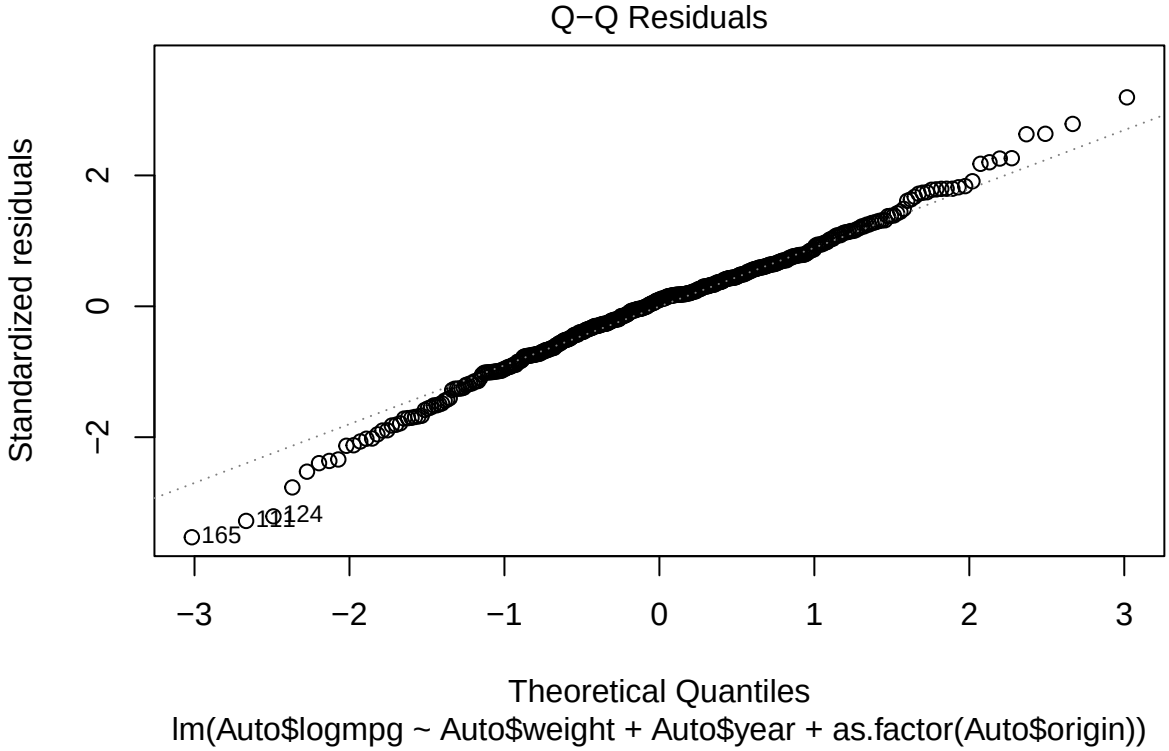


In a Tukey-Anscombe/"residuals vs. fitted" plot, one can check two assumptions. These two assumptions are: if $E[\epsilon_i] = 0$ is satisfied, and if $Var[\epsilon_i] = \sigma^2$ is satisfied.

Based on the above plot, since the plot shows an approximately flat scatter around 0, which is apparent due to the general lack of curvature in the loess curve, we have evidence that $E[\epsilon_i] = 0$. It is important to note that there seems to be a little but of curvature in the loess curve, which could imply that there persists a pattern in the residuals, and thus gives us evidence that $E[\epsilon_i] \neq 0$. However, this curvature is minuscule, and is mainly caused due to the low number of observations at the lower end of the range (perhaps due to an outlier). In this case, we will say that the $E[\epsilon_i] = 0$ is not violated.

Furthermore, based on the above plot, the width of the points is greater in the middle range of the fitted values, than it is for the upper and lower ranges of the fitted values. However, the difference in width is not great, thus we have evidence that there is constant variance, that is, $Var[\epsilon_i] = \sigma^2$. It is important to note that, although the width across the ranges isn't drastically different, there still exists a difference, so this assumption would need to be tested further in a more rigorous modeling scenario.

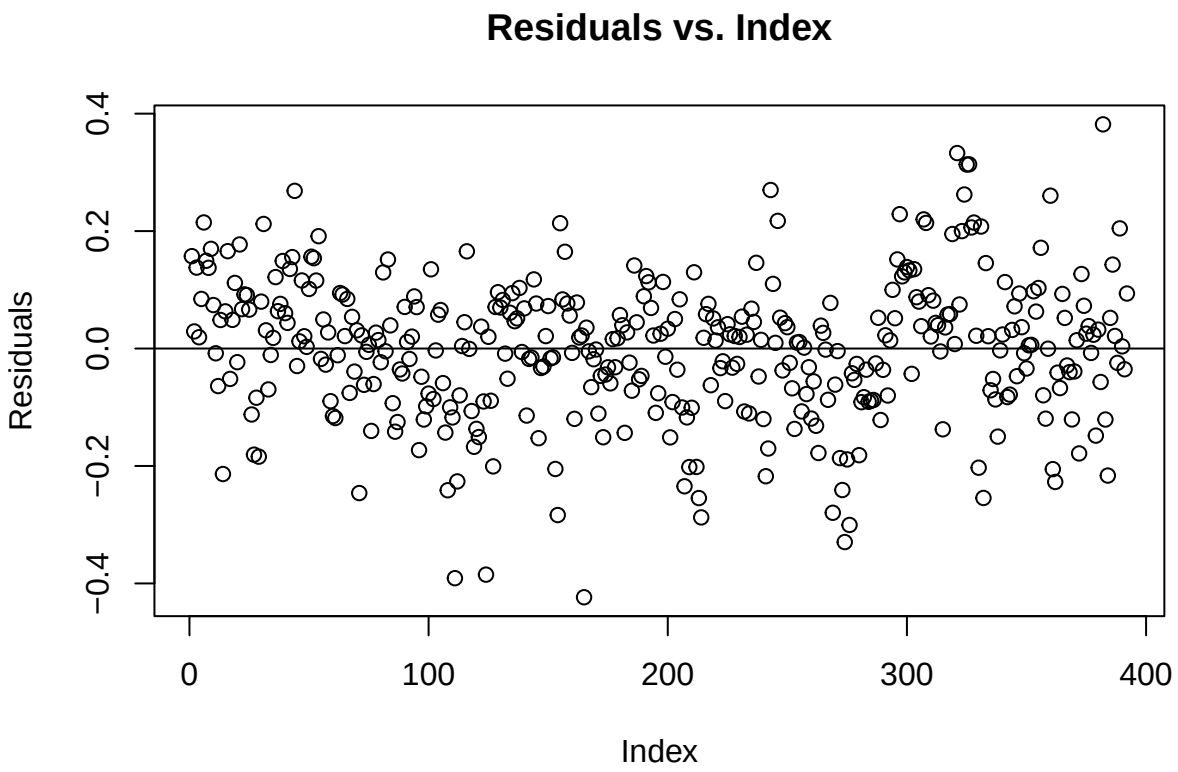
We will now plot the QQ plot of the residuals below.



Although it is possible to check that $E[\epsilon_i] = 0$ and $Var[\epsilon_i] = \sigma^2$ assumption with a QQ plot, the main assumption that is checked with a QQ plot is the $\epsilon_i \sim N(0, \sigma^2)$ assumption. If $\epsilon_i \sim N(0, \sigma^2)$, then the ordered standardized residuals $(\hat{\epsilon}_{(1)}, \ldots, \hat{\epsilon}_{(n)})$, should correspond linearly with the quantiles of a standard normal distribution.

Based on the above plot, it seems as if, for the most part, the ordered standardized residuals, $(\hat{\epsilon}_{(1)}, \ldots, \hat{\epsilon}_{(n)})$, correspond linearly with the quantiles of a standard normal distribution. Hence we can see that the normality of the residuals assumption is not violated in this model. It is important to note that, since the lower tail minimally deviates from the normal line, it is possible that the normality assumption is violated, therefore more testing would need to be done in order to confirm this conclusion.

The last assumption we have to check is the independence of the errors assumption. This will be done below using the residuals versus index plot.



Due to the lack of pattern in the residuals versus index plot, we safely concluded that the independence of the errors assumption was satisfied in this model.

Although there were slight indications of certain model assumptions being violated, we will conclude that all of our assumptions are satisfied, and thus we will keep the model as it is.

Part e

In this part of the exercise, we will expand on our model from part (d) to include two more predictors. We will justify our selection of the specific predictors chosen, as well as fit the new model and report the estimated coefficients. This is done below.

Based on our analysis in parts (b) and (c) we know that we will not be including `displacement` or `horsepower`. Furthermore, since `acceleration` was moderately positively linearly associated with `logmpg`, as well as having a low correlation with `weight`, we will include `acceleration` as one of our two new predictors. Furthermore, based on prior knowledge it appears as if certain combinations of `weight` and `acceleration` can have different affects on the `mpg` of a car (and thus different affects on the `logmpg`). Therefore, we will also include an interaction between `weight` and `acceleration` in our new model. We will now fit this new model and show the model output through the use of the `summary()` function in R. This is done below.

```
##
## Call:
## lm(formula = Auto$logmpg ~ Auto$weight + Auto$year + as.factor(Auto$origin) +
##     Auto$acceleration + Auto$weight:Auto$acceleration)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.40424 -0.07051  0.01254  0.07186  0.37849
```

```
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     7.638e-01  2.095e-01   3.645 0.000304 ***
## Auto$weight                    -9.489e-05  4.167e-05  -2.277 0.023311 *
## Auto$year                       3.308e-02  1.785e-03  18.529  < 2e-16 ***
## as.factor(Auto$origin)2         6.575e-02  1.825e-02   3.603 0.000356 ***
## as.factor(Auto$origin)3         4.734e-02  1.840e-02   2.573 0.010458 *
## Auto$acceleration               4.238e-02  8.686e-03   4.879 1.57e-06 ***
## Auto$weight:Auto$acceleration -1.271e-05  2.758e-06  -4.610 5.49e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.117 on 385 degrees of freedom
## Multiple R-squared:  0.8834, Adjusted R-squared:  0.8815
## F-statistic: 485.9 on 6 and 385 DF,  p-value: < 2.2e-16
```

As can be seen above, most of the variables in our final model are significant at any reasonable significance level. However, the `weight` and third `origin` level aren't as significant as the other predictors and predictor levels, although they still have low p-values of 0.023311 and 0.010458, respectively. Furthermore, our model contains a relatively low residual standard error of $\hat{\sigma} = 0.117$, as well as a high adjusted R-squared value of $R^2_{adj} = 0.8815$. This means that, adjusting for the number of parameters, 88.15% of the variability in y is being explained by the predictors. The RSE and adjusted R-squares are both improved in this new model.

Part f

In this part of the exercise, we will perform an ANOVA F-test in R using the `anova` function, in order to identify which of the two models (d) or (e) fit the data best. To do this we will report the F-statistic, relevant degrees of freedom, and the p-value. After we have selected the best model, we will interpret the coefficients of the best model.

We will start by running the ANOVA F-test between the model in part (c) and the model in part (e) using the `anova` function in R, the output is shown below.

```
## Analysis of Variance Table
##
## Model 1: Auto$logmpg ~ Auto$weight + Auto$year + as.factor(Auto$origin)
## Model 2: Auto$logmpg ~ Auto$weight + Auto$year + as.factor(Auto$origin) +
##     Auto$acceleration + Auto$weight:Auto$acceleration
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    387 5.6007
## 2    385 5.2737  2   0.32699 11.936 9.348e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The F-statistic from the above F-test is 11.936, which is somewhat large. Since the p-value is calculated from a $F_{p-q,n-p-1}$ distribution, it follows that $n-p-1 = 387$ represents the denominator degrees of freedom (which is also the degrees of freedom of the smaller model). Furthermore, $p - q = 387 - 385 = 2$ represents the numerator degrees of freedom (which is also the difference of the degrees of freedom between the smaller and the larger model; i.e. 385 is the degrees of freedom of the larger model). Lastly, the p-value of the test was calculated as 9.348×10^{-6}, which is extremely small. Thus, at any reasonable significance level we reject the null hypothesis and conclude that the more complex model did in fact add variables that improved the overall fit. Thus, using the output from part (e), our selected model is

$$\widehat{\log(mpg)} = 0.7638 - 0.000095x_{weight} + 0.03308x_{year} + 0.06575x_{origin2} + 0.04734x_{origin3} + 0.04238x_{acceleration} - 0.000013x_{weight}x_{acceleration}$$

Where $x_{weight} =$ `weight`, $x_{origin2}$ and $x_{origin2}$ are the dummy variables associated with the `origin` variable, $x_{year} =$ `year`, and $x_{acceleration} =$ `acceleration`. We will now interpret these estimated coefficients. This is done below.

Since we have categorical variables, we will start with interpreting the three possible different intercepts we could have. When x_{weight}, $x_{acceleration}$, $x_{origin2}$, and $x_{origin3}$ are all equal to zero (the car is from the `origin` class 1), the estimated average value for `logmpg` is $\hat{\beta}_0 = 0.7638$. Furthermore, when x_{weight}, $x_{acceleration}$, and $x_{origin3}$ are all equal to zero, and $x_{origin2} = 1$ (the car is from the `origin` class 2), the estimated average value for `logmpg` is $\hat{\beta}_0 + \hat{\beta}_{origin2} = 0.7638 + 0.06575$. Lastly, when x_{weight}, $x_{acceleration}$, and $x_{origin2}$ are all equal to zero, and $x_{origin3} = 1$ (the car is from the `origin` class 3), the estimated average value for `logmpg` is $\hat{\beta}_0 + \hat{\beta}_{origin3} = 0.7638 + 0.04734$.

We will now move onto interpreting the rest of the coefficients. Since there exists an interaction term in the model, individually interpreting the coefficients associated with this term is very tricky, so we will deal with a unit increase in x_{weight} and $x_{acceleration}$, separately.

Since we already interpreted $\hat{beta}_0$ in the last paragraph, we will move onto the other coefficients. In particular, while holding all other variables constant, increasing `year` by one unit, we estimate that `logmpg` will increase by $\hat{\beta}_{year} = 0.03308$ on average. Furthermore, $\beta_{origin2} = 0.06575$ can be interpreted as the difference in the average `logmpg` between cars from the `origin` class 2 and the `origin` class 1. Similarly, $\beta_{origin3} = 0.04734$ can be interpreted as the difference in the average `logmpg` between cars from the `origin` class level 3 and the `origin` class level 1.

Lastly, we will now move on to interpreting what happens with a one unit increase in `weight` and `acceleration`, respectively. While holding `acceleration` constant, increasing `weight` by one unit, we estimate that `logmpg` increases by $\hat{\beta}_{weight} + \hat{\beta}_{weight:acceleration} x_{acceleration} = -0.000095 - 0.000013 x_{acceleration}$ on average. Similarly, while holding `weight` constant, increasing `acceleration` by one unit, we estimate that `logmpg` increases by $\hat{\beta}_{acceleration} + \hat{\beta}_{weight:acceleration} x_{weight} = 0.04238 - 0.000013 x_{weight}$ on average.