Modelling Used Car Prices in R

Alejandro Vazquez

Load Libraries and Data

```
# load libraries
library(ggplot2)
library(GGally)
library(tidyr)
library(dplyr)
library(fixest)
library(car)
library(caret)

# clear environment
rm(list = ls())

# set working directory
setwd("/Users/alejandrovazquez/Desktop/econ121/car-sales-data")

# load data
df <- read.csv("Ad_table.csv")
```

Cleaning and Transformation

```
# verify data types are what they should be
sapply(df, class)
```

```
##        Maker     Genmodel  Genmodel_ID       Adv_ID     Adv_year    Adv_month
##  "character"  "character"  "character"  "character"    "integer"    "integer"
##        Color     Reg_year     Bodytype Runned_Miles   Engin_size      Gearbox
##  "character"    "integer"  "character"  "character"  "character"  "character"
##    Fuel_type        Price     Seat_num     Door_num
##  "character"  "character"    "integer"    "integer"
```

```
# change 'Runned_Miles' to integer type
df <- df %>% mutate(Runned_Miles = na_if(Runned_Miles, ""))
df$Runned_Miles <- as.integer(df$Runned_Miles)
```

```
## Warning: NAs introduced by coercion
```

```
# change 'Price' to integer type
df <- df %>% mutate(Price = na_if(Price, "Uknown"))
df$Price <- as.integer(df$Price)

# change blank values to NA
df <- df %>% mutate(Bodytype = na_if(Bodytype, "")) # Bodytype
df <- df %>% mutate(Color = na_if(Color, "")) # Color
df <- df %>% mutate(Engin_size = na_if(Engin_size, "")) # Engin_size
df <- df %>% mutate(Gearbox = na_if(Gearbox, "")) # Gearbox
df <- df %>% mutate(Fuel_type = na_if(Fuel_type, "")) # Fuel_type

# Remove 'L' from the end of the Engin_size values and convert to numeric
df$Engin_size <- gsub("L", "", df$Engin_size)
df$Engin_size <- as.numeric(df$Engin_size)

# view a summary of the data to see if any other adjustments are needed
summary(df)
```

```
##     Maker              Genmodel         Genmodel_ID           Adv_ID
##  Length:268255      Length:268255      Length:268255      Length:268255
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##     Adv_year      Adv_month         Color              Reg_year
##  Min.   :2012   Min.   : 1.000   Length:268255      Min.   :1900
##  1st Qu.:2018   1st Qu.: 4.000   Class :character   1st Qu.:2010
##  Median :2018   Median : 5.000   Mode  :character   Median :2014
##  Mean   :2018   Mean   : 5.626                      Mean   :2013
##  3rd Qu.:2018   3rd Qu.: 7.000                      3rd Qu.:2016
##  Max.   :2021   Max.   :33.000                      Max.   :2019
##                                                     NA's   :7
##    Bodytype          Runned_Miles       Engin_size         Gearbox
##  Length:268255      Min.   :      0   Min.   :   0.100   Length:268255
##  Class :character   1st Qu.:  14160   1st Qu.:   1.400   Class :character
```

```
##  Mode  :character   Median :  39296   Median :   1.800   Mode  :character
##                     Mean   :  48170   Mean   :   1.964
##                     3rd Qu.:  75000   3rd Qu.:   2.000
##                     Max.   :6363342   Max.   :3500.000
##                     NA's   :1313      NA's   :2064
##   Fuel_type            Price            Seat_num         Door_num
##  Length:268255      Min.   :    100   Min.   : 1.000   Min.   :0.000
##  Class :character   1st Qu.:   4990   1st Qu.: 5.000   1st Qu.:4.000
##  Mode  :character   Median :   9299   Median : 5.000   Median :5.000
##                     Mean   :  14756   Mean   : 4.904   Mean   :4.372
##                     3rd Qu.:  17150   3rd Qu.: 5.000   3rd Qu.:5.000
##                     Max.   :9999999   Max.   :17.000   Max.   :7.000
##                     NA's   :1145      NA's   :6474     NA's   :4553
```

There may be some mis-entered data in the month column as evidenced by the max value being 33. The number of null values shouldn't be an issue considering the size of the dataset.

```
# Lets take a look at the month outlier
month <- df %>% arrange(desc(Adv_month))
head(month['Adv_month'], 10)
```

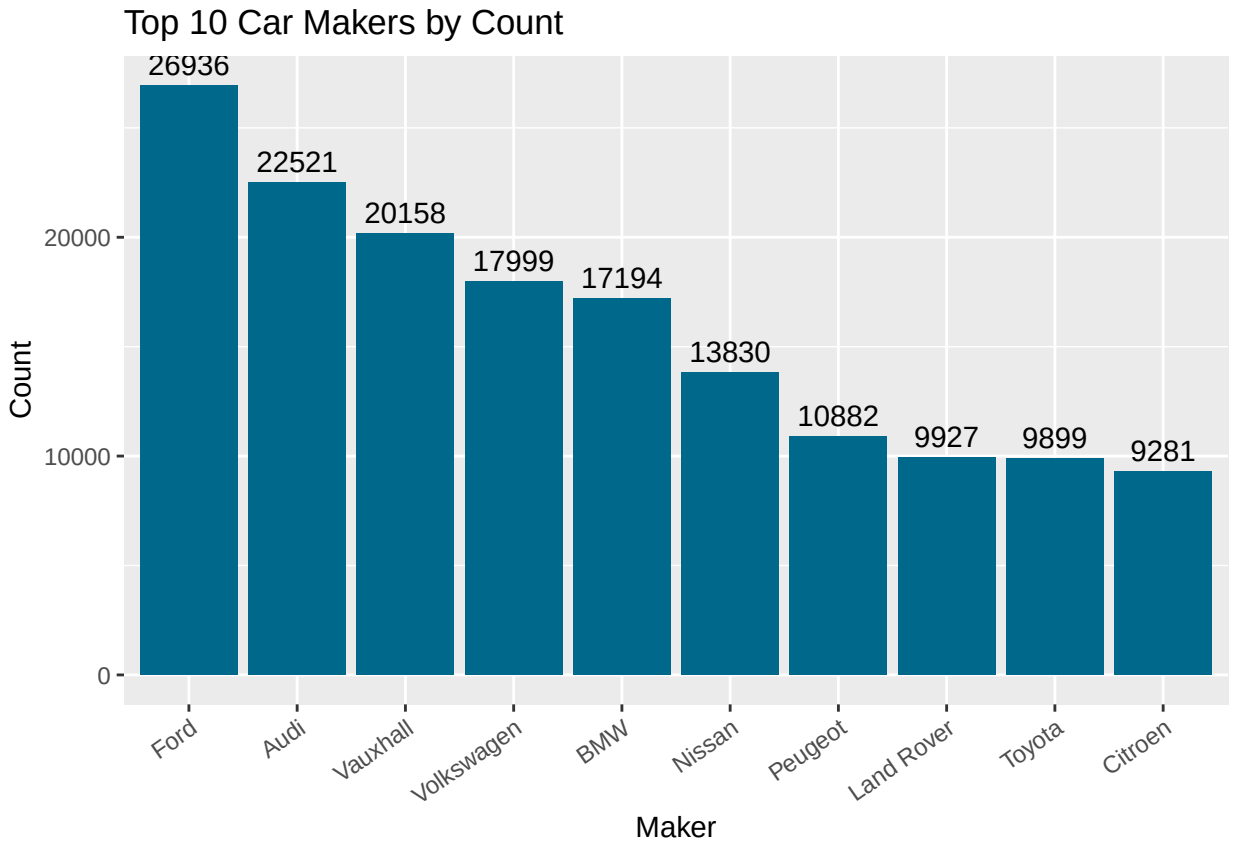
```
##    Adv_month
## 1         33
## 2         17
## 3         13
## 4         12
## 5         12
## 6         12
## 7         12
## 8         12
## 9         12
## 10        12
```

It appears that there are 3 observations where the month is above 12. I will remove them from the dataset since it is just 3 observations and won't have a big impact on the analysis.

```
# Remove month outliers
df <- df %>% filter(Adv_month <= 12)
```

Exploratory Analysis: Maker

```
# view the top 10 manufacturers represented in this dataset
make_counts <- df %>%
  group_by(Maker) %>%
  summarize(count = n()) %>%
  arrange(desc(count))

top_10 <- head(make_counts, 10) # top 10
ggplot(top_10, aes(x = reorder(Maker, -count), y = count)) +
  geom_bar(stat = "identity", fill = "deepskyblue4") +
  geom_text(aes(label = count), vjust = -0.5, position = position_dodge(width = 0.9)) +
  theme(axis.text.x = element_text(angle = 35, hjust = 1)) +
  xlab("Maker") +
  ylab("Count") +
  ggtitle("Top 10 Car Makers by Count")
```




```
summary(make_counts$count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1       3     296    3048    3422   26936
```

It seems that 25% of the makes in our dataset have less than 3 vehicles. Because of this we may need to remove these makes before creating dummies for 'Maker' to prevent overfitting and reduce model complexity.

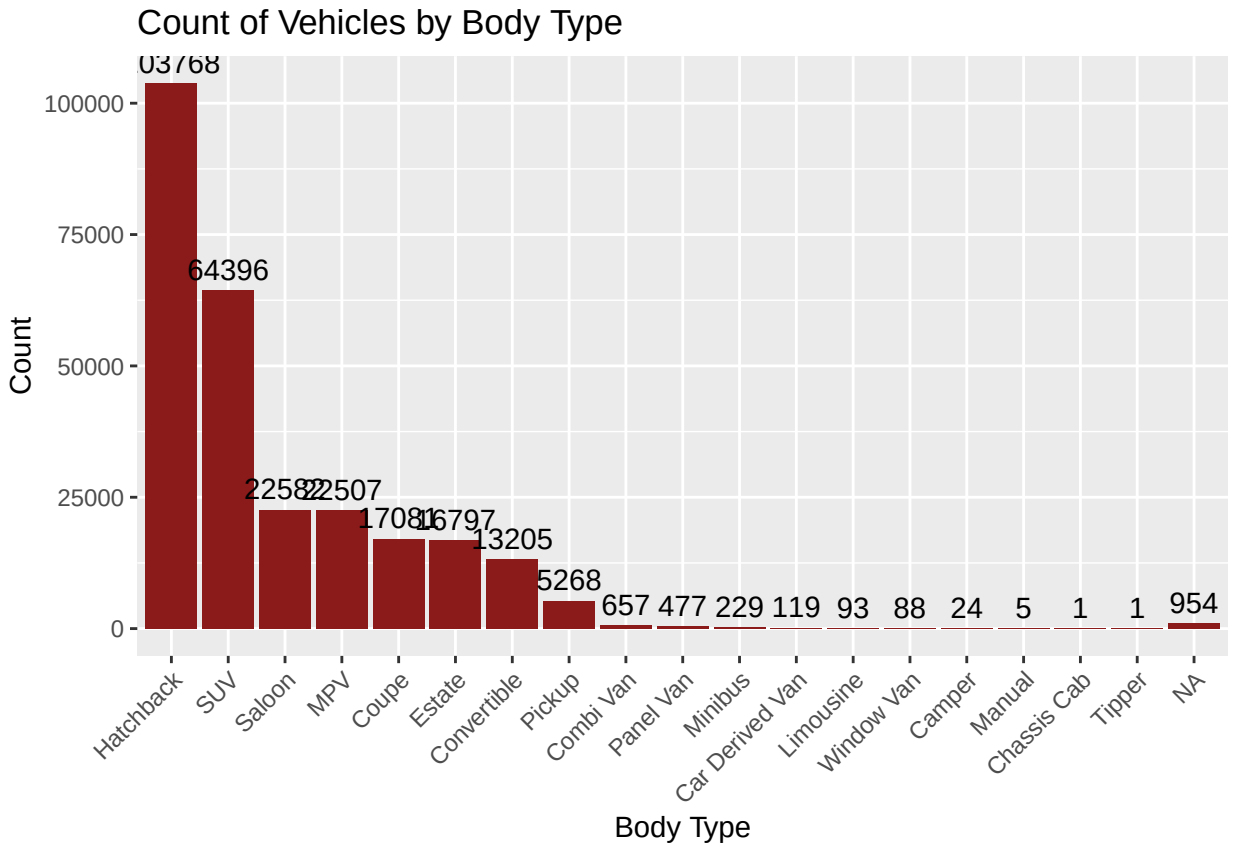
```
# Lets set a threshold at 50 vehicles.
df_fil <- df %>% filter(Maker %in% make_counts$Maker[make_counts$count >= 50])
```

I believe this is reasonable because removes makes infrequently represented in the data while maintaining a

majority of the data. Plus, this threshold ensures most high-end low-volume manufacturers like McLaren and Aston Martin remain in the dataset.

Exploratory Analysis: Body Type

```
# create a bar chart to show the body types in our dataset
bodytype_counts <- df %>%
  group_by(Bodytype) %>%
  summarise(count = n()) %>%
  arrange(desc(count))

ggplot(bodytype_counts, aes(x = reorder(Bodytype, -count), y = count)) +
  geom_bar(stat = "identity", fill = "firebrick4") +
  geom_text(aes(label = count), vjust = -0.5, position = position_dodge(width = 0.9)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  xlab("Body Type") +
  ylab("Count") +
  ggtitle("Count of Vehicles by Body Type")
```



We will need to remove some of the body types with low counts if we wish to create a dummy variable for 'Bodytype'.

```
summary(bodytype_counts$count)
```

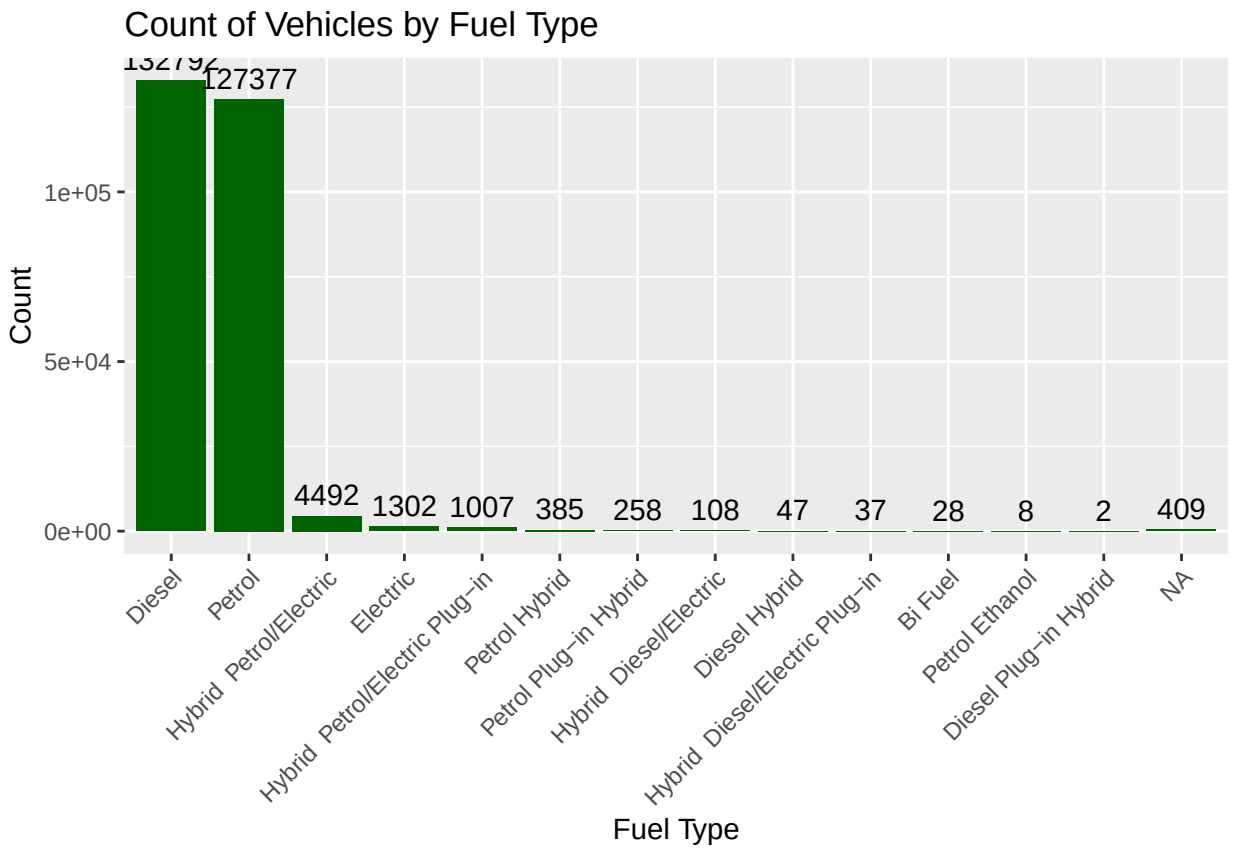
```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##      1.0     90.5    657.0  14118.5  16939.0 103768.0
```

```
# We will set a threshold at 700.
df_fil <- df_fil %>% filter(Bodytype %in% bodytype_counts$Bodytype[bodytype_counts$count >= 700])
```

This removes all the specialty body types while maintaining all the standard body types, which includes the majority of the data.

Exploratory Analysis: Fuel Type

```
# create a bar chart to show the fuel types in our dataset
fueltype_counts <- df %>%
  group_by(Fuel_type) %>%
  summarise(count = n()) %>%
  arrange(desc(count))

ggplot(fueltype_counts, aes(x = reorder(Fuel_type, -count), y = count)) +
  geom_bar(stat = "identity", fill = "darkgreen") +
  geom_text(aes(label = count), vjust = -0.5, position = position_dodge(width = 0.9)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  xlab("Fuel Type") +
  ylab("Count") +
  ggtitle("Count of Vehicles by Fuel Type")
```



It seems the vast majority of vehicles are Diesel or Petrol (~ 97%), with a small proportion being hybrid (~ 1.7%) or electric (~ 0.5%). This is also a concern if we wish to create dummy variables for fuel type.

```
summary(fueltype_counts$count)
```

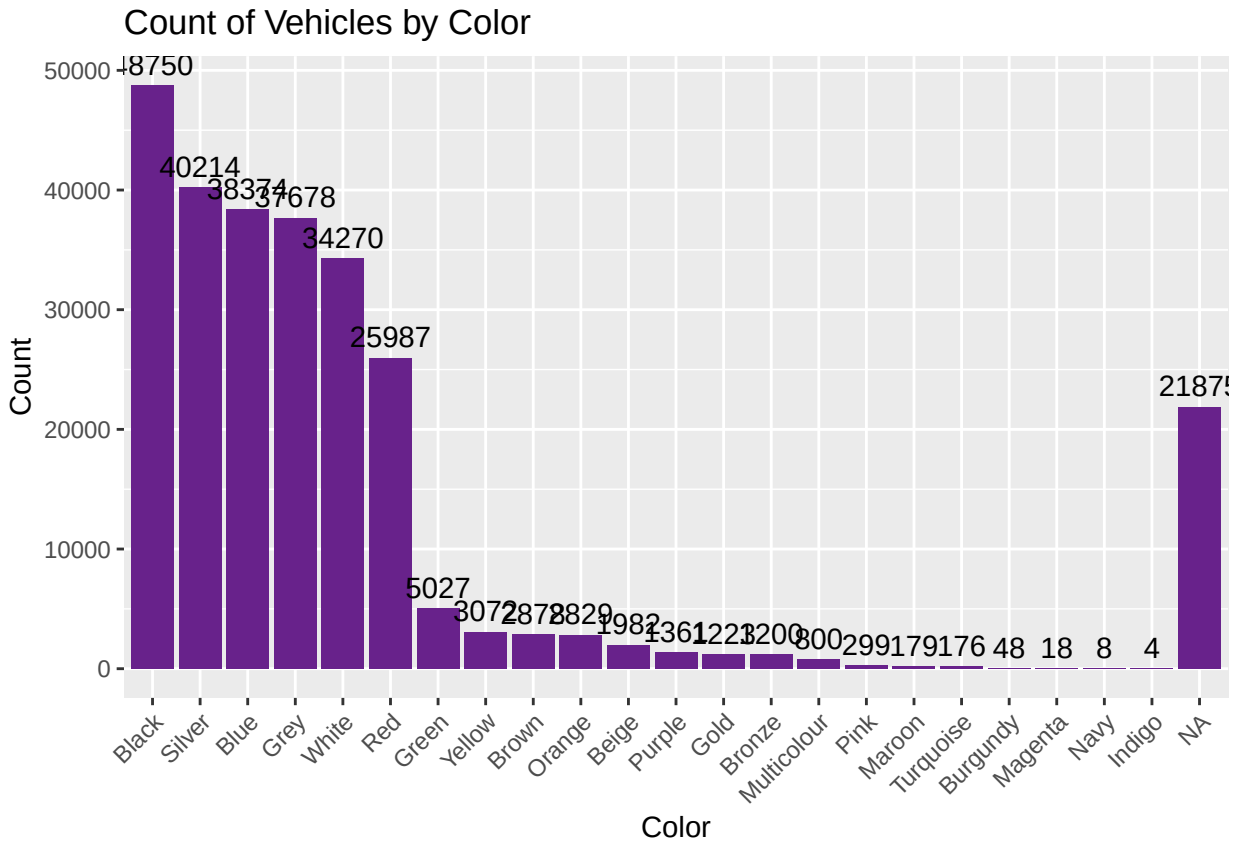
```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##      2.0     39.5    321.5  19160.9   1228.2 132792.0
```

```
# We will set a threshold at 300.
df_fil <- df_fil %>% filter(Fuel_type %in% fueltype_counts$Fuel_type[fueltype_counts$count >= 300])
```

This removes the fuel types that are not common while maintaining the most common fuel types which represent the majority of the data.

Exploratory Analysis: Color

```
# create a bar chart to show the colors in our dataset
color_counts <- df %>%
  group_by(Color) %>%
  summarise(count = n()) %>%
  arrange(desc(count))

ggplot(color_counts, aes(x = reorder(Color, -count), y = count)) +
  geom_bar(stat = "identity", fill = "darkorchid4") +
  geom_text(aes(label = count), vjust = -0.5, position = position_dodge(width = 0.9)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  xlab("Color") +
  ylab("Count") +
  ggtitle("Count of Vehicles by Color")
```



Most of the vehicles in our dataset are black, silver, blue, grey, white, and red. If we wish to create dummy variables for color we may have to remove the other colors, but there is a large amount of NAs which may prevent us from doing so.

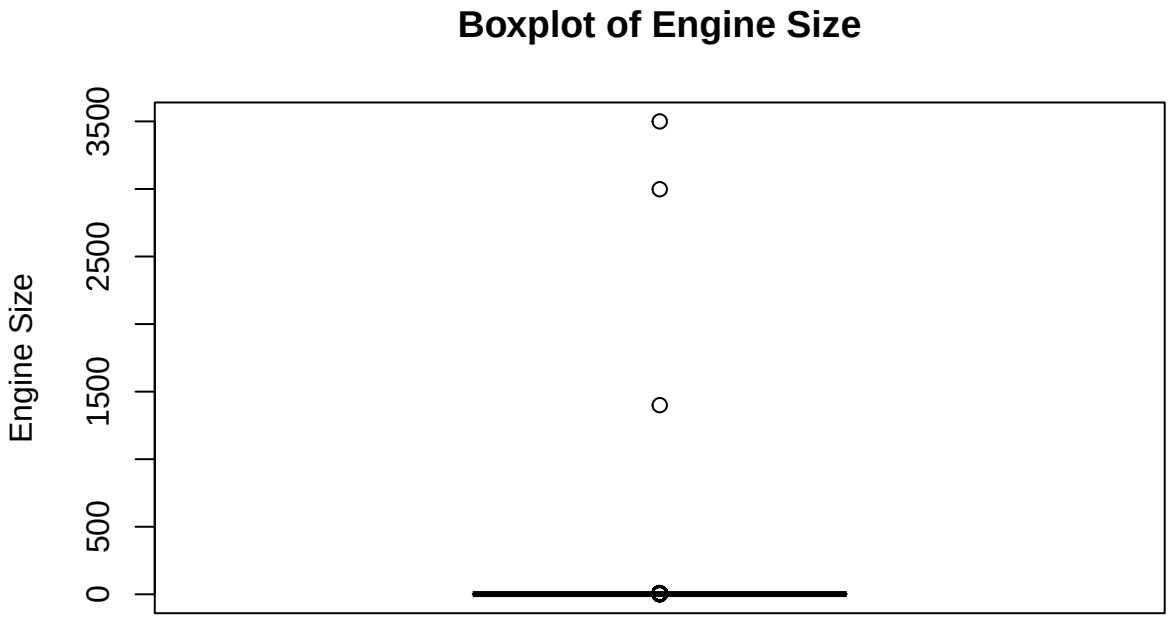
```
summary(color_counts$count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       4     239    1982   11663   23931   48750
```

For now I won't filter out any colors because I do not think color has that large of an impact on sale price. Although I may revisit this later.

Exploratory Analysis: Engine Size

```
# plot a box plot for engine size to see outliers
boxplot(df_fil$Engin_size, main = "Boxplot of Engine Size", ylab = "Engine Size")
```



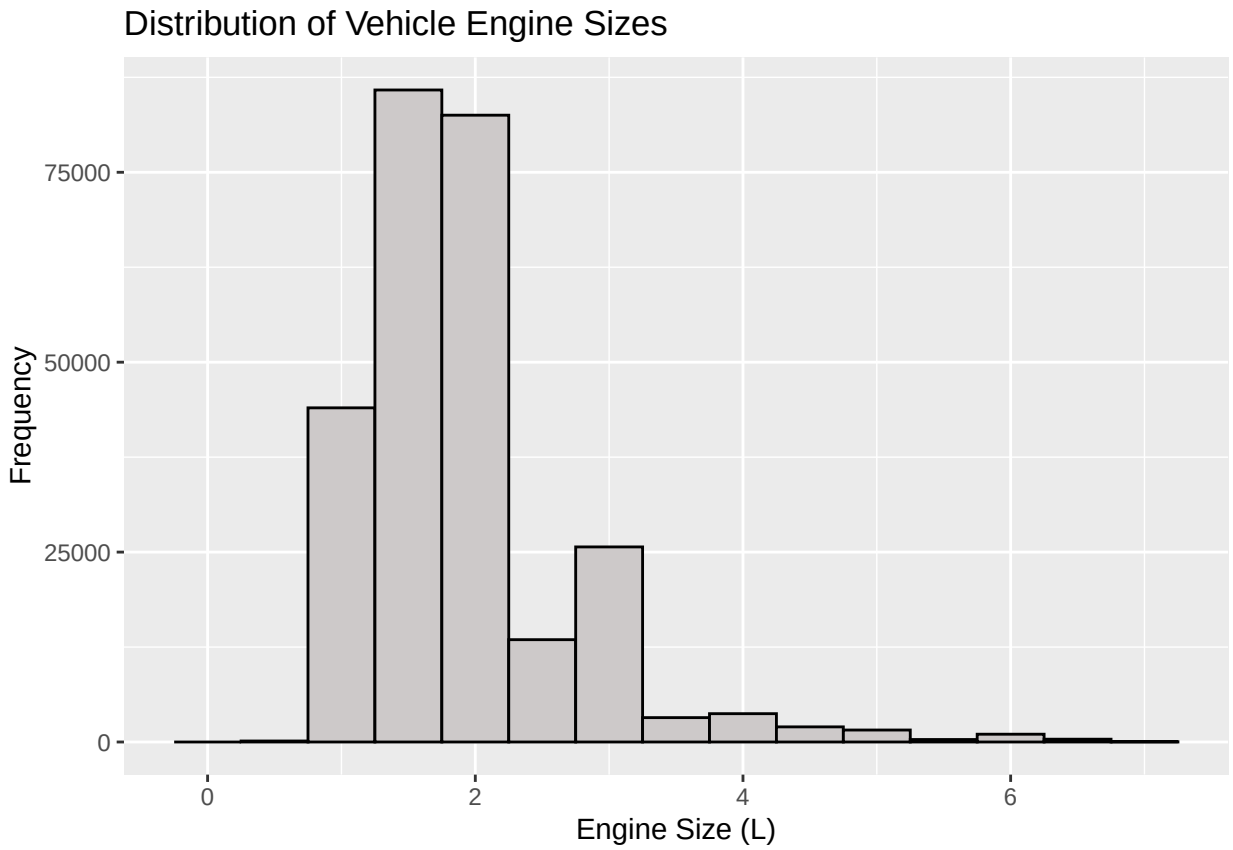
Looks like there are some significant outliers, one vehicle even has 3000 Liters! Lets remove them since it is only a few.

```
# identify outliers by engine size
IQR_values <- IQR(df$Engin_size, na.rm = TRUE)
Q1 <- quantile(df$Engin_size, 0.25, na.rm = TRUE)
Q3 <- quantile(df$Engin_size, 0.75, na.rm = TRUE)

lower_bound <- Q1 - 1.5 * IQR_values # 0.5
upper_bound <- Q3 + 1.5 * IQR_values # 2.9

outliers <- subset(df, Engin_size < lower_bound | Engin_size > upper_bound)
nrow(outliers)
```

```
## [1] 37314
```

Because there are 37,314 outliers, I think we should only remove the extreme outliers so as not to lost too much data.

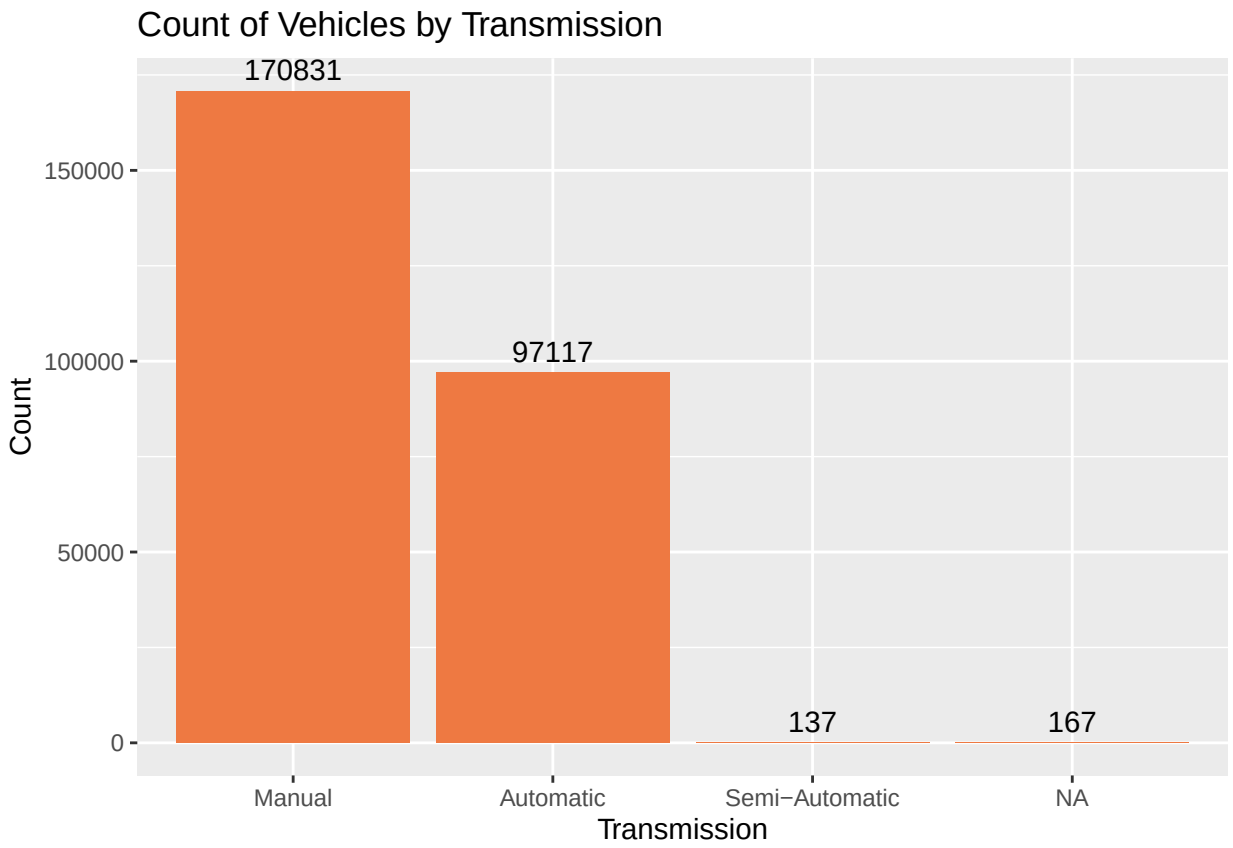
```
# Remove outliers with extremely large engine size (5 observations)
df_fil <- subset(df_fil, Engin_size <= 8)

# create a histogram to show the engine sizes in our dataset
ggplot(df_fil, aes(x = Engin_size)) +
  geom_histogram(binwidth = 0.5, fill = "snow3", color = "black") +
  xlab("Engine Size (L)") +
  ylab("Frequency") +
  ggtitle("Distribution of Vehicle Engine Sizes")
```




```
# Log transformation of Engine Size to mitigate effect of outliers
df_fil$log_Engin_size <- log(df_fil$Engin_size + 1)
```

Exploratory Analysis: Gearbox

```r
# create a bar chart to show the transmissions in our dataset
gearbox_counts <- df %>%
  group_by(Gearbox) %>%
  summarise(count = n()) %>%
  arrange(desc(count))

ggplot(gearbox_counts, aes(x = reorder(Gearbox, -count), y = count)) +
  geom_bar(stat = "identity", fill = "sienna2") +
  geom_text(aes(label = count), vjust = -0.5, position = position_dodge(width = 0.9)) +
  xlab("Transmission") +
  ylab("Count") +
  ggtitle("Count of Vehicles by Transmission")
```



If we create a dummy variable for Gearbox we can just remove semi-automatic since the vast majority of the data is either manual or automatic.

```r
# We will set a threshold at 200.
df_fil <- df_fil %>% filter(Gearbox %in% gearbox_counts$Gearbox[gearbox_counts$count >= 200])
```

This removes semi-automatic vehicles which represent a tiny fraction of the data.

Exploratory Analysis: Model Years

```
# view the distribution of model years for vehicles sold
year_counts <- df %>%
  group_by(Reg_year) %>%
  summarise(count = n()) %>%
  arrange(desc(count))
summary(year_counts$Reg_year)
```

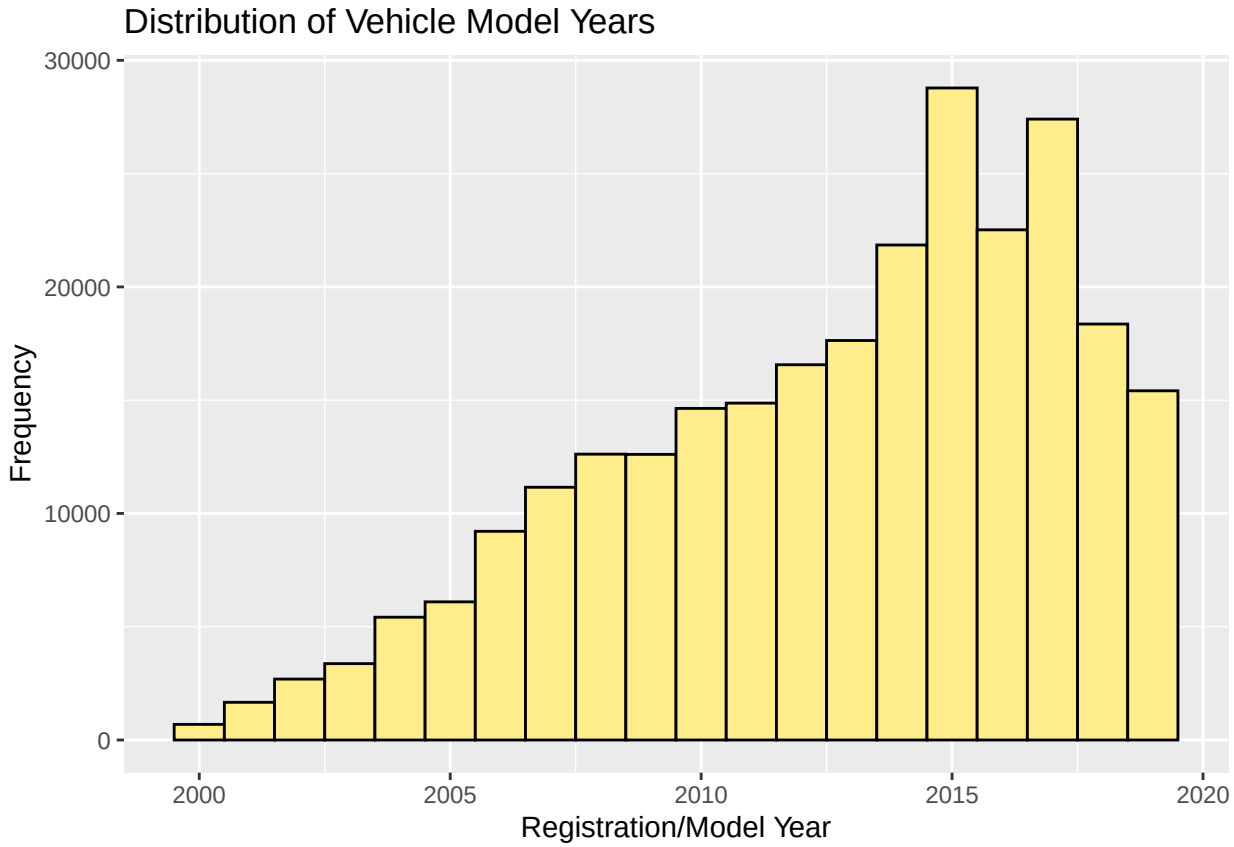
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    1900    2000    2006    1999    2013    2019       1
```

```
tail(year_counts, 10)
```

```
## # A tibble: 10 x 2
##    Reg_year count
##       <int> <int>
##  1     2002  2713
##  2     2001  1685
##  3     2000   699
##  4     1995    88
##  5     1990    37
##  6       NA     7
##  7     1970     3
##  8     1980     3
##  9     1960     2
## 10     1900     1
```

Since there are only 134 vehicles made before 2000, we can filter our data to only include post-2000 vehicles to reduce model complexity.

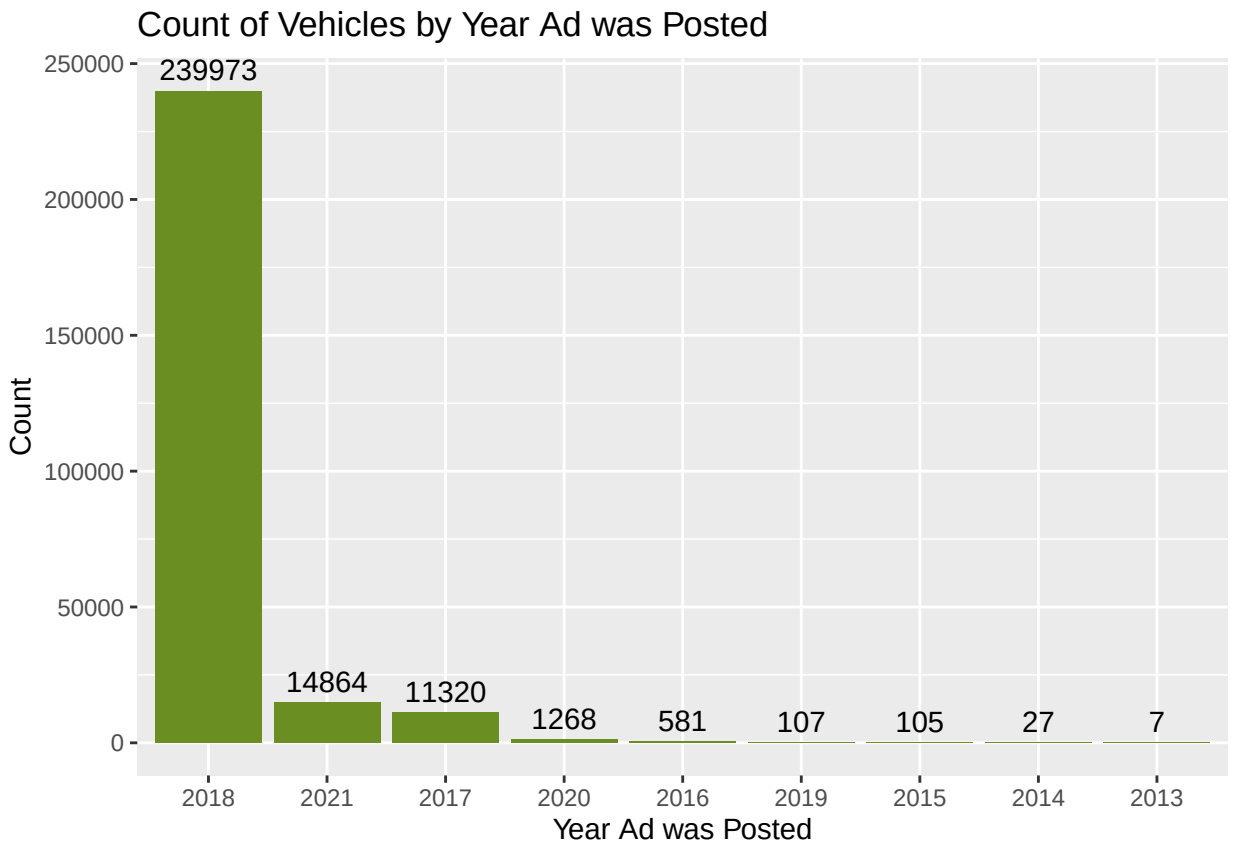
```
# remove vehicles older than 2000
df_fil <- df_fil %>% filter(Reg_year >= 2000)

# now we can plot a histogram of the years
ggplot(df_fil, aes(x = Reg_year)) +
  geom_histogram(binwidth = 1, fill = "lightgoldenrod1", color = "black") +
  xlab("Registration/Model Year") +
  ylab("Frequency") +
  ggtitle("Distribution of Vehicle Model Years")
```



Left skewed. It seems the majority of our data is relatively newer vehicles manufactured within the past decade.

Exploratory Analysis: Ad Years

```
# plot a bar chart of the years the ads were posted
adyear_counts <- df %>%
  group_by(Adv_year) %>%
  summarise(count = n()) %>%
  arrange(desc(count))

ggplot(adyear_counts, aes(x = reorder(Adv_year, -count), y = count)) +
  geom_bar(stat = "identity", fill = "olivedrab") +
  geom_text(aes(label = count), vjust = -0.5, position = position_dodge(width = 0.9)) +
  xlab("Year Ad was Posted") +
  ylab("Count") +
  ggtitle("Count of Vehicles by Year Ad was Posted")
```

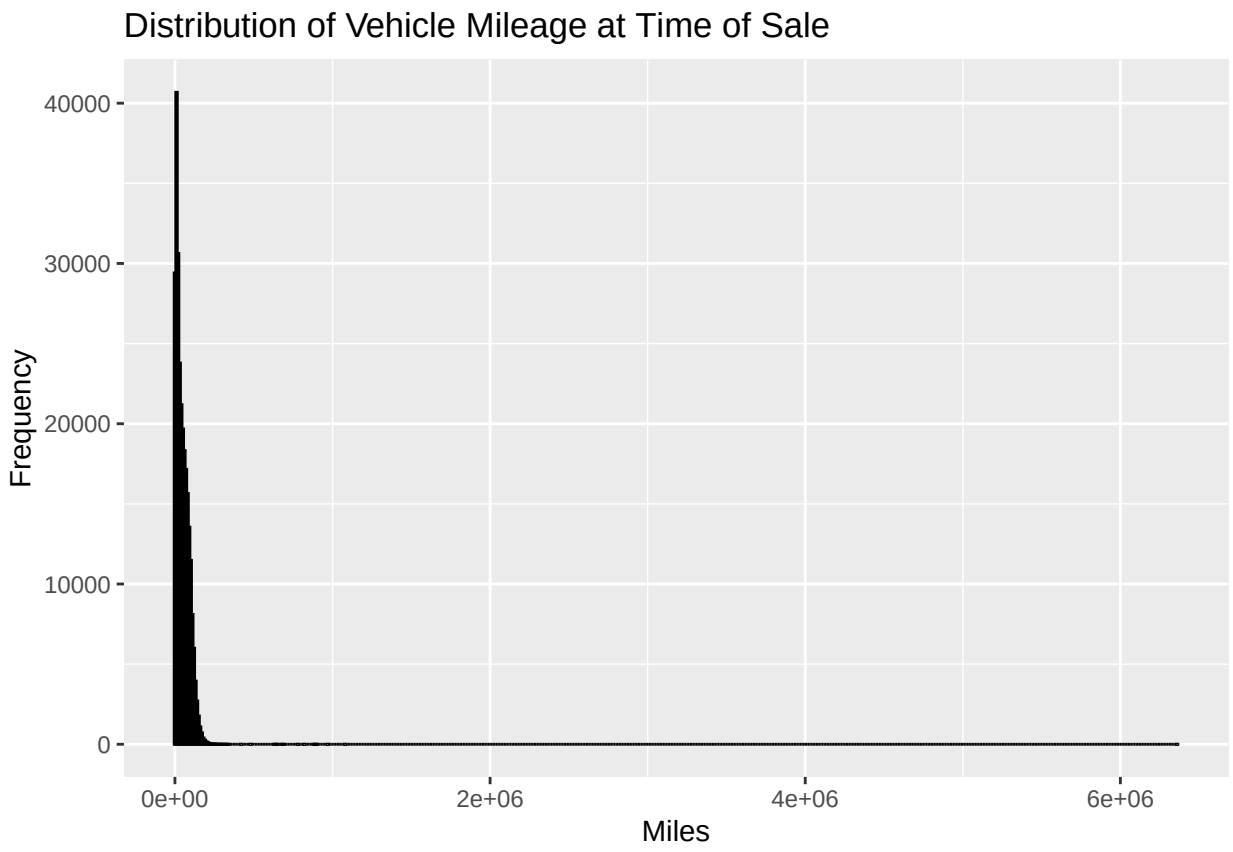


Most of the vehicles in our data were listed for sale in 2018, which means our model will be most accurate for predicting sale price in 2018 value.

Exploratory Analysis: Miles

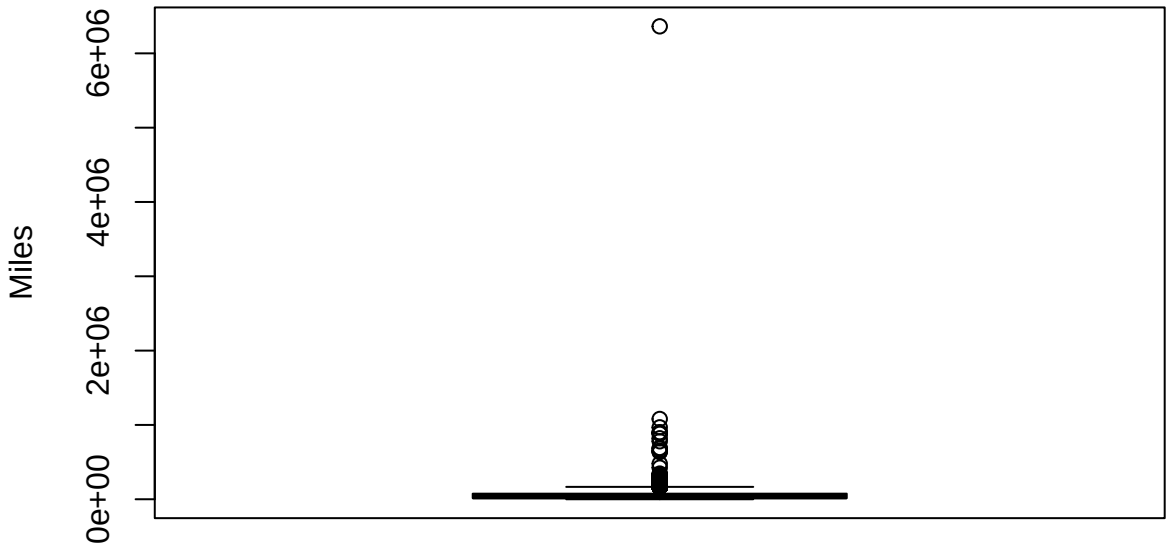
```
# plot a histogram for miles
ggplot(df, aes(x = Runned_Miles)) +
  geom_histogram(binwidth = 10000, fill = "dodgerblue2", color = "black") +
  xlab("Miles") +
  ylab("Frequency") +
  ggtitle("Distribution of Vehicle Mileage at Time of Sale")
```

```
## Warning: Removed 1313 rows containing non-finite values (`stat_bin()`).
```



We have some outliers because the histogram is extremely right skewed. Let's get rid of those observations.

```
# plot a box plot for miles to see outliers more clearly
boxplot(df$Runned_Miles, main = "Boxplot of Runned Miles", ylab = "Miles")
```



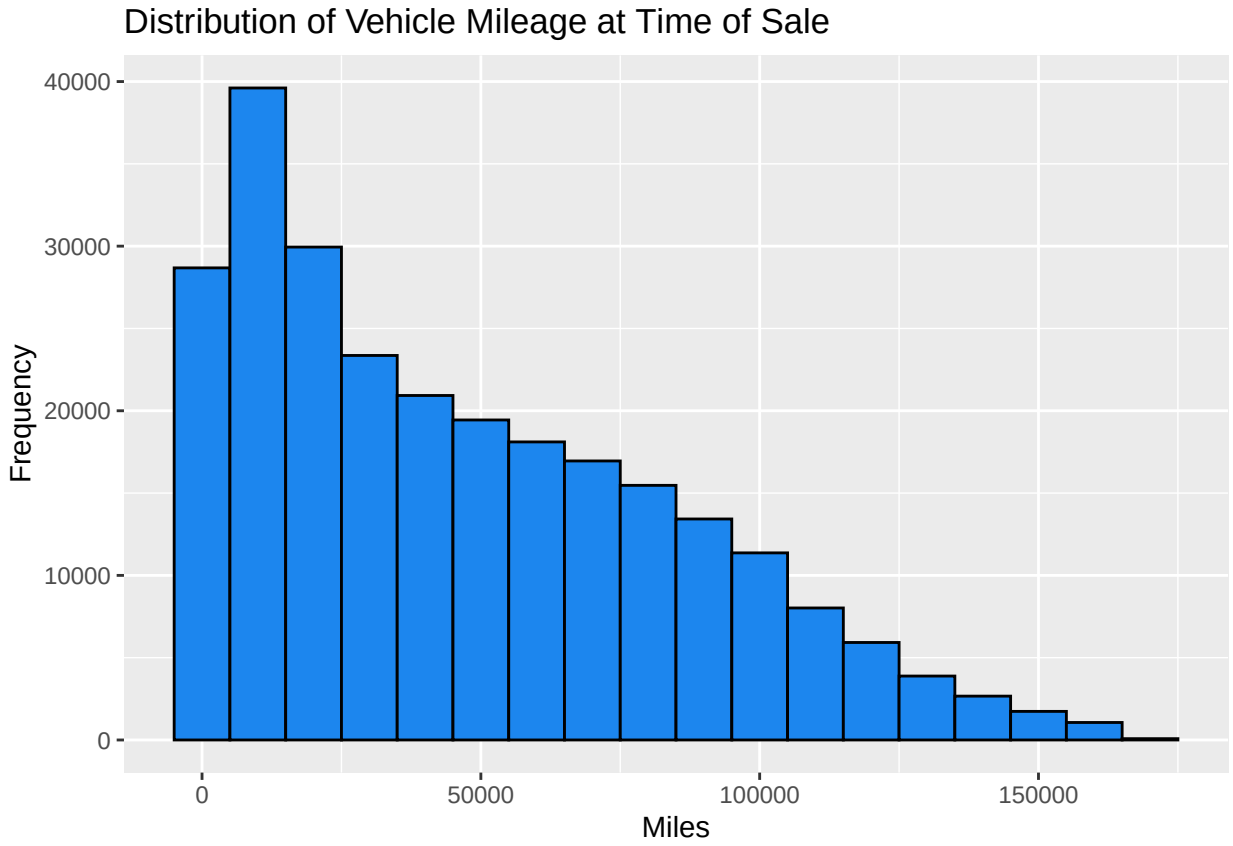
It seems one of the vehicles somehow sold with over 6,000,000 miles. We should remove this vehicle along with the other outliers.

```
# identify outliers by miles
IQR_values <- IQR(df$Runned_Miles, na.rm = TRUE)
Q1 <- quantile(df$Runned_Miles, 0.25, na.rm = TRUE)
Q3 <- quantile(df$Runned_Miles, 0.75, na.rm = TRUE)

lower_bound <- Q1 - 1.5 * IQR_values # -77,120
upper_bound <- Q3 + 1.5 * IQR_values # 166,272

# Because a vehicle cannot have negative miles, we can disregard the lower bound
outliers <- subset(df, Runned_Miles > upper_bound)
nrow(outliers)
```

```
## [1] 1749
```

We have 1,749 outliers. Lets remove them since it is only a small portion of the data and it will give us a clearer effect of miles on sale price in our model.

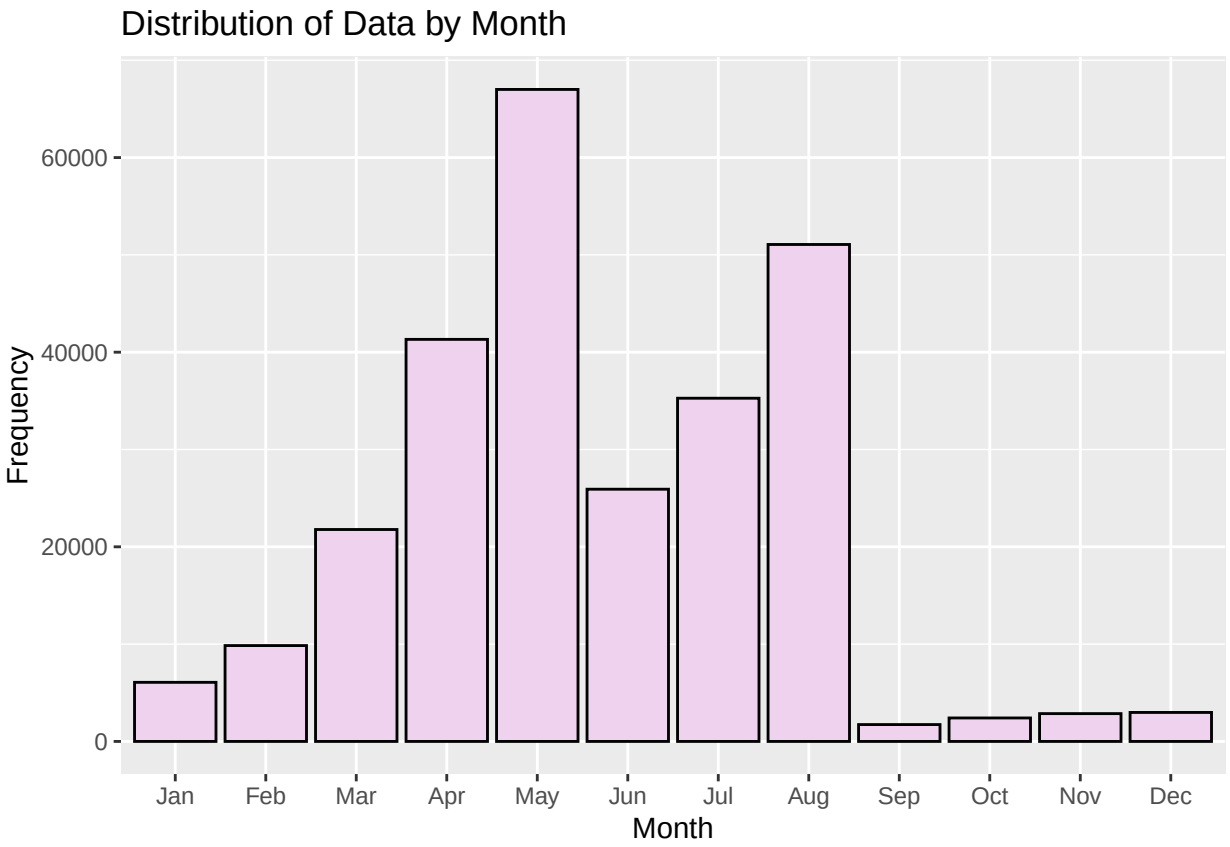
```
# Remove outliers with high mileage
df_fil <- subset(df_fil, Runned_Miles <= upper_bound)

# re-plot a histogram for miles
ggplot(df_fil, aes(x = Runned_Miles)) +
  geom_histogram(binwidth = 10000, fill = "dodgerblue2", color = "black") +
  xlab("Miles") +
  ylab("Frequency") +
  ggtitle("Distribution of Vehicle Mileage at Time of Sale")
```



Now that we can properly interpret this histogram, we can see it is right skewed, which means most of the vehicles in our data have lower mileage (less than 50,000).

Exploratory Analysis: Month of Sale

```
# create bar plot for months of sale
ggplot(df, aes(x = as.factor(Adv_month))) +
  geom_bar(fill = "thistle2", color = "black") +
  scale_x_discrete(labels = c('Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov
  xlab("Month") +
  ylab("Frequency") +
  ggtitle("Distribution of Data by Month")
```



For some reason there is a lack of sales during the last four months of the year.

Exploratory Analysis: Price

```
# check how many rows have no selling price
sum(is.na(df_fil$Price))
```

```
## [1] 1114
```

```
# remove rows without a selling price since it is a tiny portion of the data
df_fil <- df_fil[!is.na(df_fil$Price), ]

# plot a box plot for sale price to see outliers
boxplot(df_fil$Price, main = "Boxplot of Sale Price", ylab = "Sale Price")
```

There are some extreme outliers that we will need to remove.

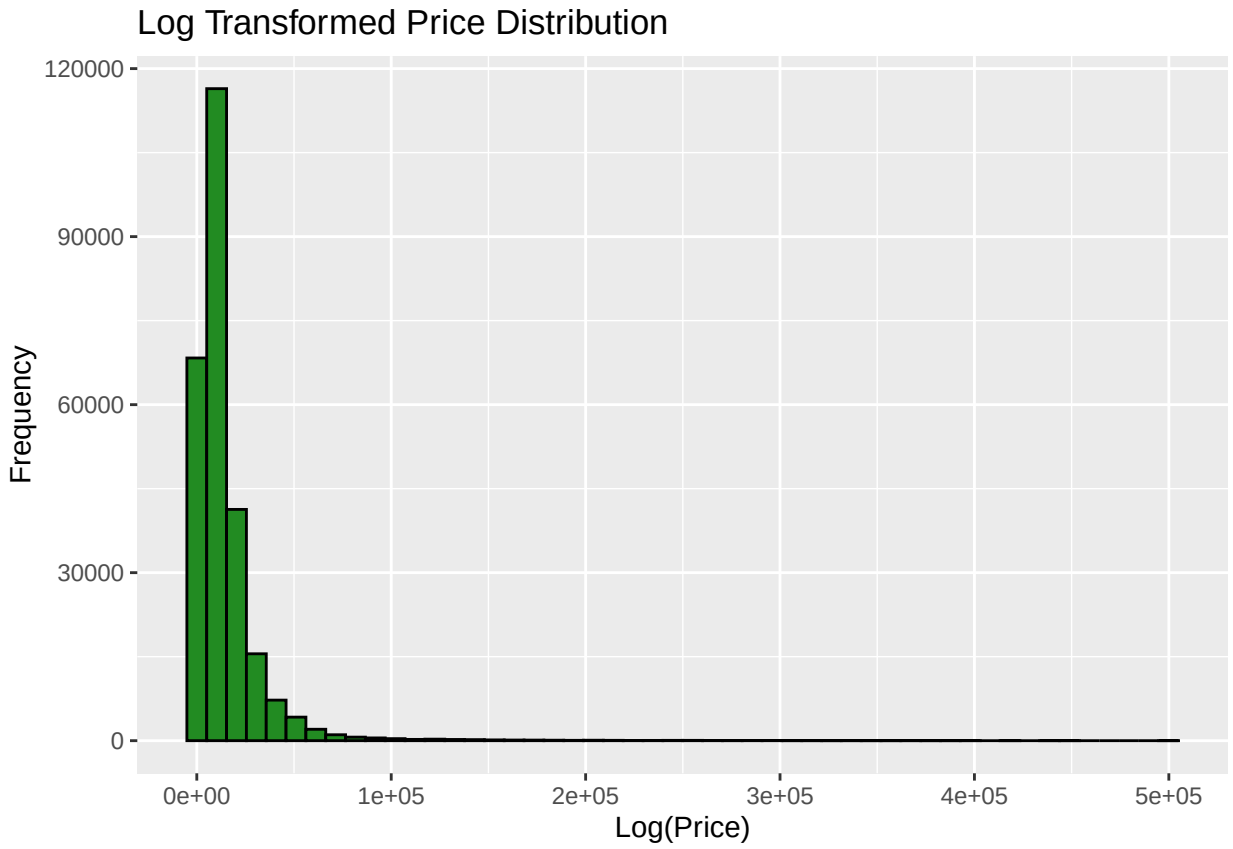
```
# identify outliers by price
IQR_values <- IQR(df$Price, na.rm = TRUE)
Q1 <- quantile(df$Price, 0.25, na.rm = TRUE)
Q3 <- quantile(df$Price, 0.75, na.rm = TRUE)

lower_bound <- Q1 - 1.5 * IQR_values # -13,250
upper_bound <- Q3 + 1.5 * IQR_values # 35,390

# Because a vehicle cannot sell for a negative price, we can disregard the lower bound
outliers <- subset(df, Price > upper_bound)
nrow(outliers)
```

```
## [1] 19033
```

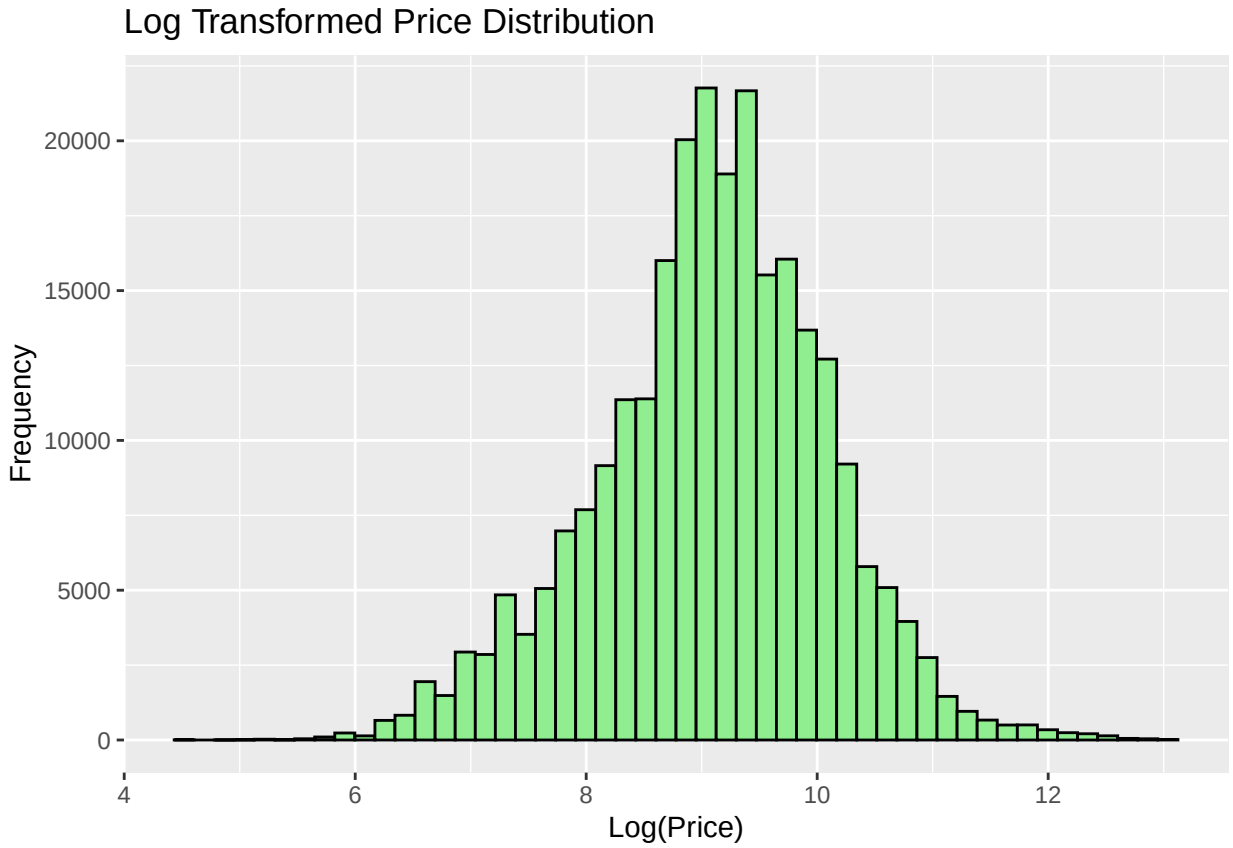
Since there are 19,033 outliers, we will need to be careful not to remove all of them. I will set a threshold at 500,000 to remove the extreme outliers while retaining data on some high-value vehicles.

```
# Remove outliers with extremely high price
df_fil <- subset(df_fil, Price <= 500000)

# Log transformation of Price to mitigate effect of outliers
```

```r
df_fil$log_Price <- log(df_fil$Price)

# Histogram for Price data
ggplot(df_fil, aes(x = Price)) +
  geom_histogram(bins = 50, fill = "forestgreen", color = "black") +
  ggtitle("Log Transformed Price Distribution") +
  xlab("Log(Price)") +
  ylab("Frequency")
```




```r
# Histogram for the log-transformed Price data
ggplot(df_fil, aes(x = log_Price)) +
  geom_histogram(bins = 50, fill = "lightgreen", color = "black") +
  ggtitle("Log Transformed Price Distribution") +
  xlab("Log(Price)") +
  ylab("Frequency")
```



The distribution is much closer to normal for log transformed price.

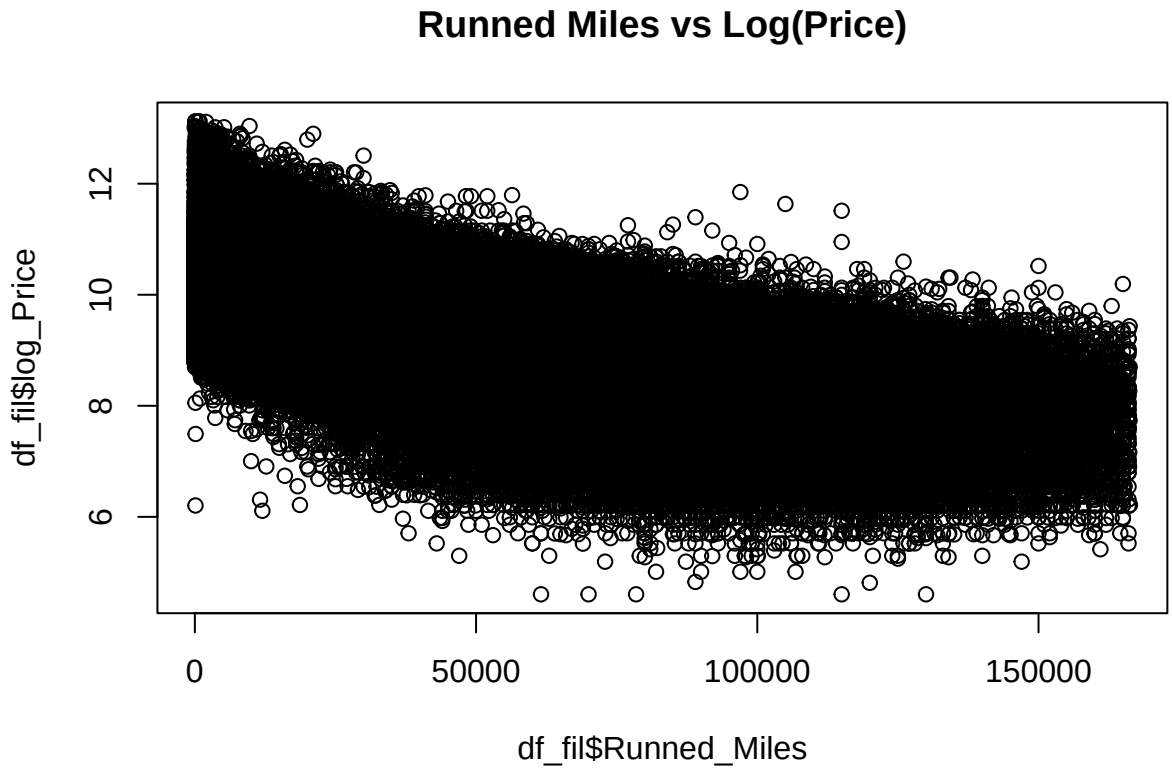
Exploratory Analysis Conclusion

```r
# Lets calculate what % of observations were dropped from my original dataset
original_count <- nrow(df)
filtered_count <- nrow(df_fil)
percentage_filtered_out <- ((original_count - filtered_count) / original_count) * 100
paste("Percentage of observations filtered out: ", round(percentage_filtered_out, 2), "%", sep = "")
```

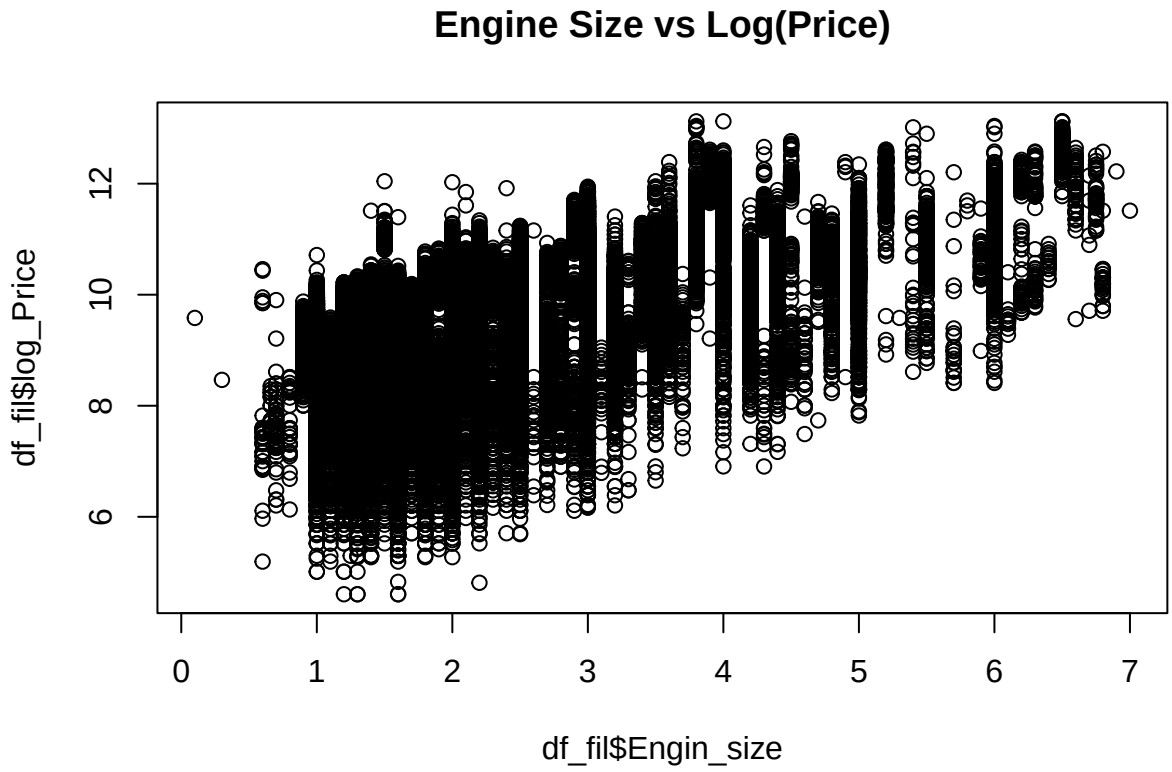
```
## [1] "Percentage of observations filtered out: 3.26%"
```

Scatter plots

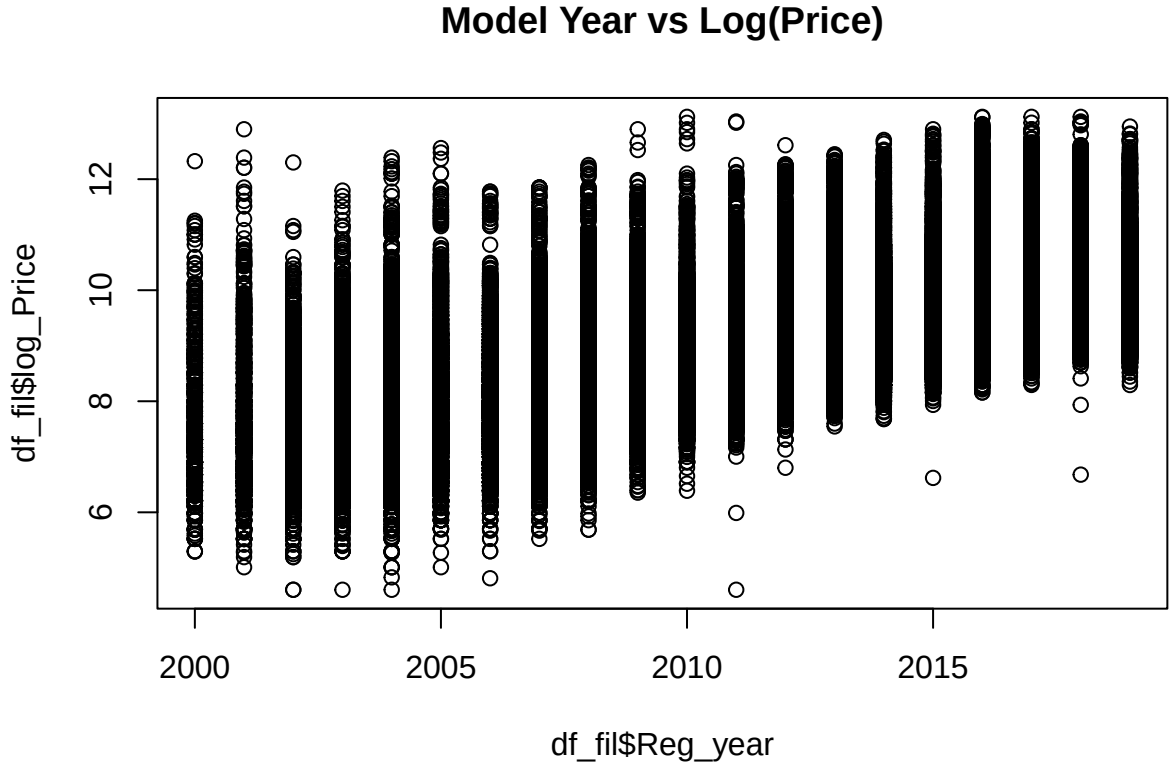
```
# plot miles vs log price
plot(df_fil$Runned_Miles, df_fil$log_Price, main = "Runned Miles vs Log(Price)")
```




```
# plot engine size vs log price
plot(df_fil$Engin_size, df_fil$log_Price, main = "Engine Size vs Log(Price)")
```




```
# plot model year vs log price
plot(df_fil$Reg_year, df_fil$log_Price, main = "Model Year vs Log(Price)")
```



Split Data into Training, Validation, Test

```
set.seed(792002) # Set a seed for reproducibility

# convert categorical variables to factors
df_fil$Maker <- as.factor(df_fil$Maker)
df_fil$Fuel_type <- as.factor(df_fil$Fuel_type)
df_fil$Bodytype <- as.factor(df_fil$Bodytype)
df_fil$Gearbox <- as.factor(df_fil$Gearbox)
df_fil$Color <- as.factor(df_fil$Color)

# view baselines for categorical variables
levels(df_fil$Maker)[1] # Make
```

```
## [1] "Abarth"
```

```
levels(df_fil$Bodytype)[1] # Body Type
```

```
## [1] "Convertible"
```

```
levels(df_fil$Gearbox)[1] # Gearbox
```

```
## [1] "Automatic"
```

```
levels(df_fil$Fuel_type)[1] # Fuel Type
```

```
## [1] "Diesel"
```

```
levels(df_fil$Color)[1] # Color
```

```
## [1] "Beige"
```

```
# change the baselines for easier interpretation of coefficients
df_fil$Maker <- relevel(df_fil$Maker, ref = "Toyota")
df_fil$Bodytype <- relevel(df_fil$Bodytype, ref = "Saloon")
df_fil$Fuel_type <- relevel(df_fil$Fuel_type, ref = "Petrol")
df_fil$Color <- relevel(df_fil$Color, ref = "White")

# Randomize the dataset by sampling rows
df_fil <- df_fil[sample(nrow(df_fil)), ]

# Split dataset into training (60%) and the test + validation (40%)
trainingIndex <- createDataPartition(df_fil$Price, p = 0.60, list = FALSE)
df_training <- df_fil[trainingIndex, ]
tempSet <- df_fil[-trainingIndex, ]

# Split the remaining 40% into validation (50% of 40%) and test sets (50% of 40%)
validationIndex <- createDataPartition(tempSet$Price, p = 0.5, list = FALSE)
df_validation <- tempSet[validationIndex, ]
df_test <- tempSet[-validationIndex, ]
```

Linear Model 1

```r
# Model 1 - contains the most possible relevant dependent variables
model_1 <- lm(log_Price ~ Runned_Miles + Reg_year + Engin_size + Bodytype + Maker
            + Fuel_type + Gearbox + Color,
            data = df_training)

# View the model summary
summary(model_1)
```

```
##
## Call:
## lm(formula = log_Price ~ Runned_Miles + Reg_year + Engin_size +
##     Bodytype + Maker + Fuel_type + Gearbox + Color, data = df_training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4684 -0.1514  0.0070  0.1612  3.4122
##
## Coefficients:
##                                        Estimate Std. Error  t value
## (Intercept)                          -2.411e+02  5.961e-01 -404.537
## Runned_Miles                         -6.382e-06  3.379e-08 -188.883
## Reg_year                              1.241e-01  2.954e-04  420.027
## Engin_size                            3.920e-01  1.626e-03  241.086
## BodytypeConvertible                   2.009e-01  4.622e-03   43.460
## BodytypeCoupe                         2.190e-01  4.300e-03   50.933
## BodytypeEstate                        7.121e-02  4.190e-03   16.993
## BodytypeHatchback                    -4.939e-02  3.480e-03  -14.190
## BodytypeMPV                           9.338e-02  4.196e-03   22.253
## BodytypePickup                        3.121e-01  6.848e-03   45.582
## BodytypeSUV                           2.315e-01  3.451e-03   67.075
## MakerAbarth                           5.257e-02  1.885e-02    2.789
## MakerAlfa Romeo                       5.794e-02  1.145e-02    5.060
## MakerAston Martin                     6.115e-01  3.440e-02   17.777
## MakerAudi                             3.163e-01  5.059e-03   62.531
## MakerBentley                          5.067e-01  1.366e-02   37.110
## MakerBMW                              1.919e-01  5.413e-03   35.450
## MakerChevrolet                       -4.892e-01  1.486e-02  -32.916
## MakerChrysler                        -3.531e-01  1.938e-02  -18.219
## MakerCitroen                         -2.328e-01  5.692e-03  -40.904
## MakerDacia                           -3.800e-01  1.042e-02  -36.458
## MakerDaihatsu                        -1.696e-02  3.817e-02   -0.444
## MakerDodge                           -2.896e-01  3.625e-02   -7.988
## MakerDS                              -6.894e-02  1.485e-02   -4.643
## MakerFerrari                          1.542e+00  1.722e-02   89.576
## MakerFiat                            -2.446e-01  6.299e-03  -38.840
## MakerFord                            -1.348e-01  4.782e-03  -28.181
## MakerHonda                            5.166e-02  6.684e-03    7.729
## MakerHyundai                         -1.543e-01  6.231e-03  -24.765
## MakerInfiniti                        -7.548e-02  1.853e-02   -4.074
## MakerIsuzu                           -7.713e-02  1.692e-02   -4.558
## MakerJaguar                           1.776e-01  6.438e-03   27.593
## MakerJeep                            -2.100e-01  1.088e-02  -19.298
```

```
## MakerKia                                -1.469e-01  5.833e-03  -25.180
## MakerLamborghini                         1.056e+00  2.409e-02   43.822
## MakerLand Rover                          4.067e-01  6.092e-03   66.759
## MakerLexus                              -5.543e-02  8.922e-03   -6.213
## MakerLotus                               6.472e-01  5.341e-02   12.119
## MakerMaserati                            3.851e-01  1.704e-02   22.605
## MakerMazda                              -1.216e-01  6.685e-03  -18.191
## MakerMcLaren                             1.299e+00  2.645e-02   49.091
## MakerMercedes-Benz                       1.986e-01  6.072e-03   32.709
## MakerMG                                 -4.126e-01  1.484e-02  -27.807
## MakerMINI                                1.100e-01  7.938e-03   13.852
## MakerMitsubishi                         -1.956e-01  7.894e-03  -24.783
## MakerNissan                             -9.082e-02  5.378e-03  -16.886
## MakerPeugeot                            -2.451e-01  5.544e-03  -44.200
## MakerPorsche                             7.678e-01  7.490e-03  102.508
## MakerRenault                            -2.470e-01  6.025e-03  -40.988
## MakerRolls-Royce                         6.477e-01  5.123e-02   12.643
## MakerRover                              -5.755e-01  3.088e-02  -18.633
## MakerSaab                               -2.841e-01  1.364e-02  -20.834
## MakerSEAT                               -7.416e-02  7.557e-03   -9.813
## MakerSKODA                              -7.201e-02  6.321e-03  -11.394
## MakerSmart                              -2.991e-01  1.411e-02  -21.191
## MakerSsangyong                          -3.096e-01  1.295e-02  -23.908
## MakerSubaru                              2.727e-01  1.079e-02   25.274
## MakerSuzuki                             -2.473e-01  8.261e-03  -29.937
## MakerTesla                               1.050e+00  2.917e-01    3.601
## MakerVauxhall                           -2.656e-01  4.959e-03  -53.568
## MakerVolkswagen                          4.123e-02  5.117e-03    8.059
## MakerVolvo                               1.187e-01  6.136e-03   19.341
## Fuel_typeDiesel                          9.270e-02  1.886e-03   49.163
## Fuel_typeElectric                        3.746e-01  7.798e-02    4.803
## Fuel_typeHybrid  Petrol/Electric         2.676e-01  7.285e-03   36.728
## Fuel_typeHybrid  Petrol/Electric Plug-in 5.797e-01  1.350e-02   42.953
## Fuel_typePetrol Hybrid                  -9.404e-02  1.968e-02   -4.777
## GearboxManual                           -1.029e-01  2.053e-03  -50.127
## ColorBeige                              -1.189e-02  8.541e-03   -1.393
## ColorBlack                              -2.091e-02  2.655e-03   -7.873
## ColorBlue                               -3.862e-02  2.804e-03  -13.775
## ColorBronze                              3.627e-02  1.087e-02    3.337
## ColorBrown                               2.881e-02  7.216e-03    3.993
## ColorBurgundy                            1.417e-02  5.742e-02    0.247
## ColorGold                               -6.899e-02  1.078e-02   -6.401
## ColorGreen                              -2.086e-02  5.673e-03   -3.677
## ColorGrey                               -9.592e-03  2.794e-03   -3.433
## ColorIndigo                              7.673e-03  2.811e-01    0.027
## ColorMagenta                             8.572e-02  8.478e-02    1.011
## ColorMaroon                             -7.665e-03  2.816e-02   -0.272
## ColorMulticolour                         1.426e-02  1.354e-02    1.053
## ColorNavy                               -2.919e-02  1.623e-01   -0.180
## ColorOrange                              3.667e-02  7.273e-03    5.042
## ColorPink                                1.552e-02  2.140e-02    0.725
## ColorPurple                             -2.078e-02  1.016e-02   -2.047
## ColorRed                                -3.231e-02  3.078e-03  -10.497
## ColorSilver                             -4.278e-02  2.813e-03  -15.206
```

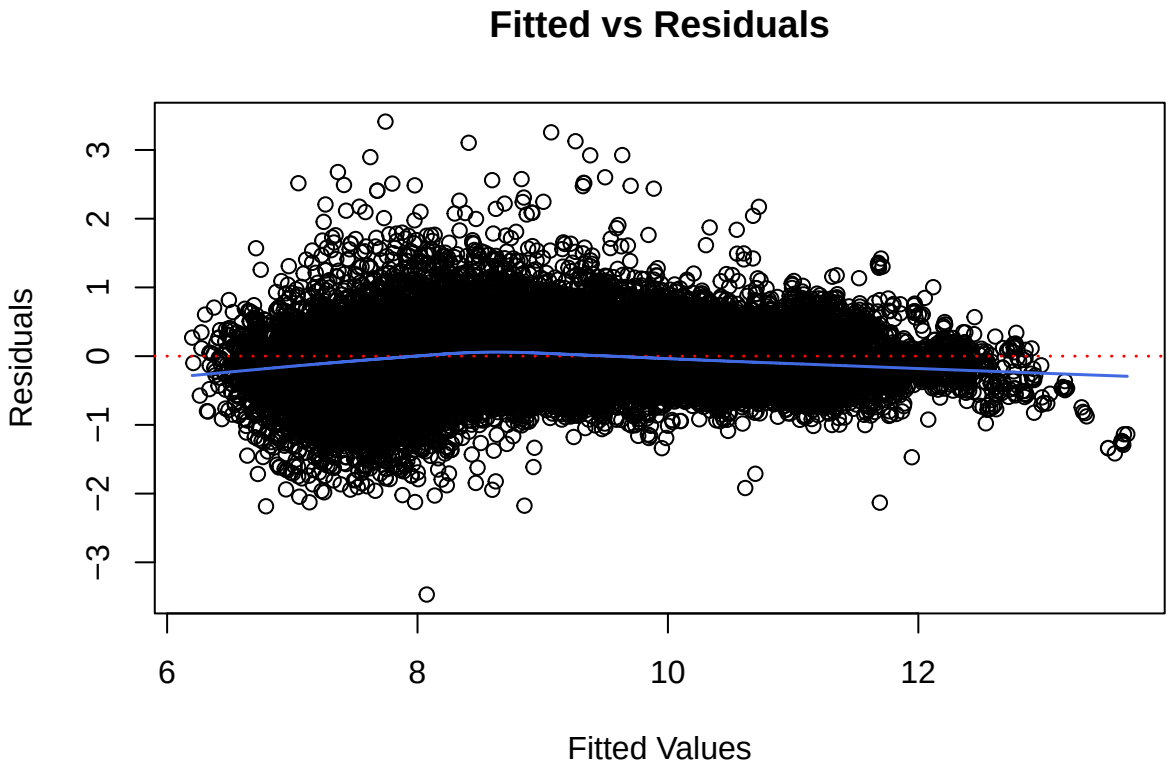
```
## ColorTurquoise                    -2.793e-02  2.791e-02   -1.001
## ColorYellow                        4.552e-02  6.952e-03    6.548
##                                   Pr(>|t|)
## (Intercept)                        < 2e-16 ***
## Runned_Miles                       < 2e-16 ***
## Reg_year                           < 2e-16 ***
## Engin_size                         < 2e-16 ***
## BodytypeConvertible                < 2e-16 ***
## BodytypeCoupe                      < 2e-16 ***
## BodytypeEstate                     < 2e-16 ***
## BodytypeHatchback                  < 2e-16 ***
## BodytypeMPV                        < 2e-16 ***
## BodytypePickup                     < 2e-16 ***
## BodytypeSUV                        < 2e-16 ***
## MakerAbarth                       0.005285 **
## MakerAlfa Romeo                   4.19e-07 ***
## MakerAston Martin                  < 2e-16 ***
## MakerAudi                          < 2e-16 ***
## MakerBentley                       < 2e-16 ***
## MakerBMW                           < 2e-16 ***
## MakerChevrolet                     < 2e-16 ***
## MakerChrysler                      < 2e-16 ***
## MakerCitroen                       < 2e-16 ***
## MakerDacia                         < 2e-16 ***
## MakerDaihatsu                     0.656808
## MakerDodge                        1.38e-15 ***
## MakerDS                           3.43e-06 ***
## MakerFerrari                       < 2e-16 ***
## MakerFiat                          < 2e-16 ***
## MakerFord                          < 2e-16 ***
## MakerHonda                        1.09e-14 ***
## MakerHyundai                       < 2e-16 ***
## MakerInfiniti                     4.62e-05 ***
## MakerIsuzu                        5.18e-06 ***
## MakerJaguar                        < 2e-16 ***
## MakerJeep                          < 2e-16 ***
## MakerKia                           < 2e-16 ***
## MakerLamborghini                   < 2e-16 ***
## MakerLand Rover                    < 2e-16 ***
## MakerLexus                        5.22e-10 ***
## MakerLotus                         < 2e-16 ***
## MakerMaserati                      < 2e-16 ***
## MakerMazda                         < 2e-16 ***
## MakerMcLaren                       < 2e-16 ***
## MakerMercedes-Benz                 < 2e-16 ***
## MakerMG                            < 2e-16 ***
## MakerMINI                          < 2e-16 ***
## MakerMitsubishi                    < 2e-16 ***
## MakerNissan                        < 2e-16 ***
## MakerPeugeot                       < 2e-16 ***
## MakerPorsche                       < 2e-16 ***
## MakerRenault                       < 2e-16 ***
## MakerRolls-Royce                   < 2e-16 ***
## MakerRover                         < 2e-16 ***
```

```
## MakerSaab                                  < 2e-16 ***
## MakerSEAT                                  < 2e-16 ***
## MakerSKODA                                 < 2e-16 ***
## MakerSmart                                 < 2e-16 ***
## MakerSsangyong                             < 2e-16 ***
## MakerSubaru                                < 2e-16 ***
## MakerSuzuki                                < 2e-16 ***
## MakerTesla                                0.000317 ***
## MakerVauxhall                              < 2e-16 ***
## MakerVolkswagen                           7.76e-16 ***
## MakerVolvo                                 < 2e-16 ***
## Fuel_typeDiesel                            < 2e-16 ***
## Fuel_typeElectric                         1.56e-06 ***
## Fuel_typeHybrid  Petrol/Electric           < 2e-16 ***
## Fuel_typeHybrid  Petrol/Electric Plug-in   < 2e-16 ***
## Fuel_typePetrol Hybrid                    1.78e-06 ***
## GearboxManual                              < 2e-16 ***
## ColorBeige                                0.163754
## ColorBlack                                3.49e-15 ***
## ColorBlue                                  < 2e-16 ***
## ColorBronze                               0.000848 ***
## ColorBrown                                6.54e-05 ***
## ColorBurgundy                             0.805024
## ColorGold                                 1.55e-10 ***
## ColorGreen                                0.000236 ***
## ColorGrey                                 0.000597 ***
## ColorIndigo                               0.978221
## ColorMagenta                              0.311976
## ColorMaroon                               0.785453
## ColorMulticolour                          0.292196
## ColorNavy                                 0.857271
## ColorOrange                               4.61e-07 ***
## ColorPink                                 0.468247
## ColorPurple                               0.040697 *
## ColorRed                                   < 2e-16 ***
## ColorSilver                                < 2e-16 ***
## ColorTurquoise                            0.316937
## ColorYellow                               5.84e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.281 on 142217 degrees of freedom
##   (13401 observations deleted due to missingness)
## Multiple R-squared:  0.9212, Adjusted R-squared:  0.9212
## F-statistic: 1.89e+04 on 88 and 142217 DF,  p-value: < 2.2e-16
```

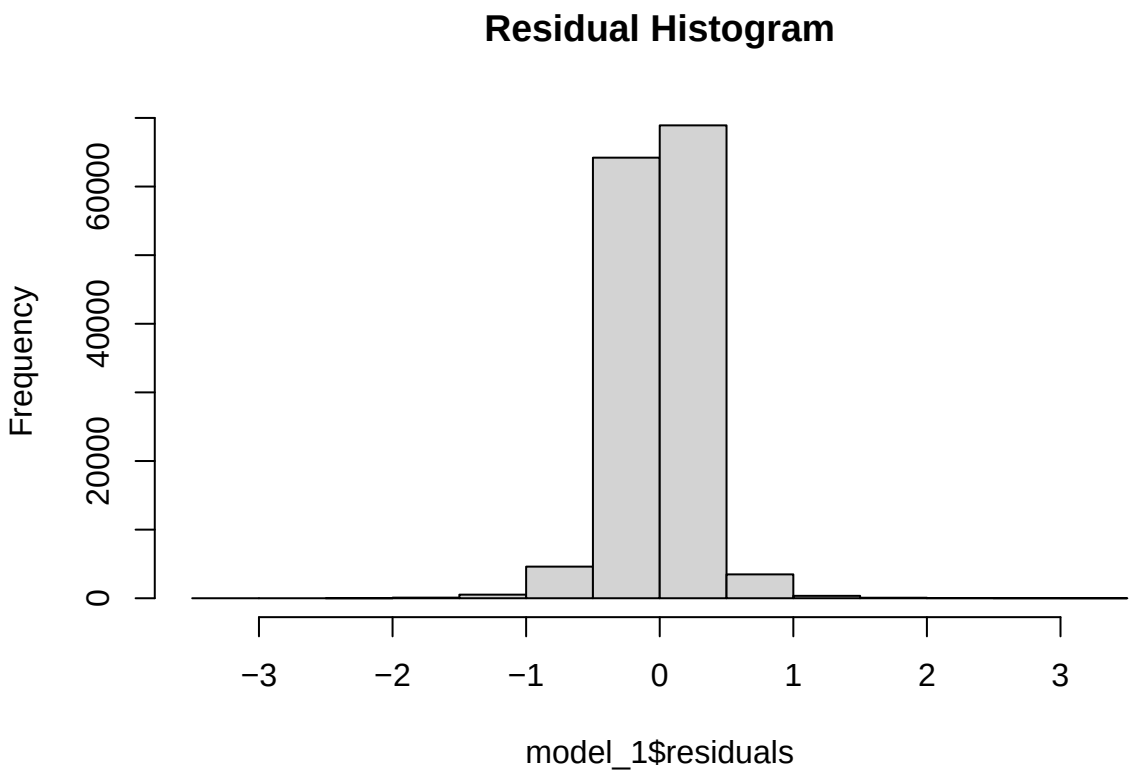
Note: since the dependent variable is log transformed, we must exponentiate the coefficient (e^coeff) to get the multiplicative factor for every one-unit increase in the independent variable. For example, the coefficient on MakerBMW is 1.393e-01 –> exp(1.393e-01) is 1.149469. This means for every 1 unit increase in the MakerBMW variable, the sale price increases by a factor of 1.149, or approximately 15%.

```r
# create fitted vs residuals plot
residuals <- residuals(model_1)
fitted_values <- fitted(model_1)
plot(fitted_values, residuals, xlab = "Fitted Values", ylab = "Residuals", main = "Fitted vs Residuals")
```

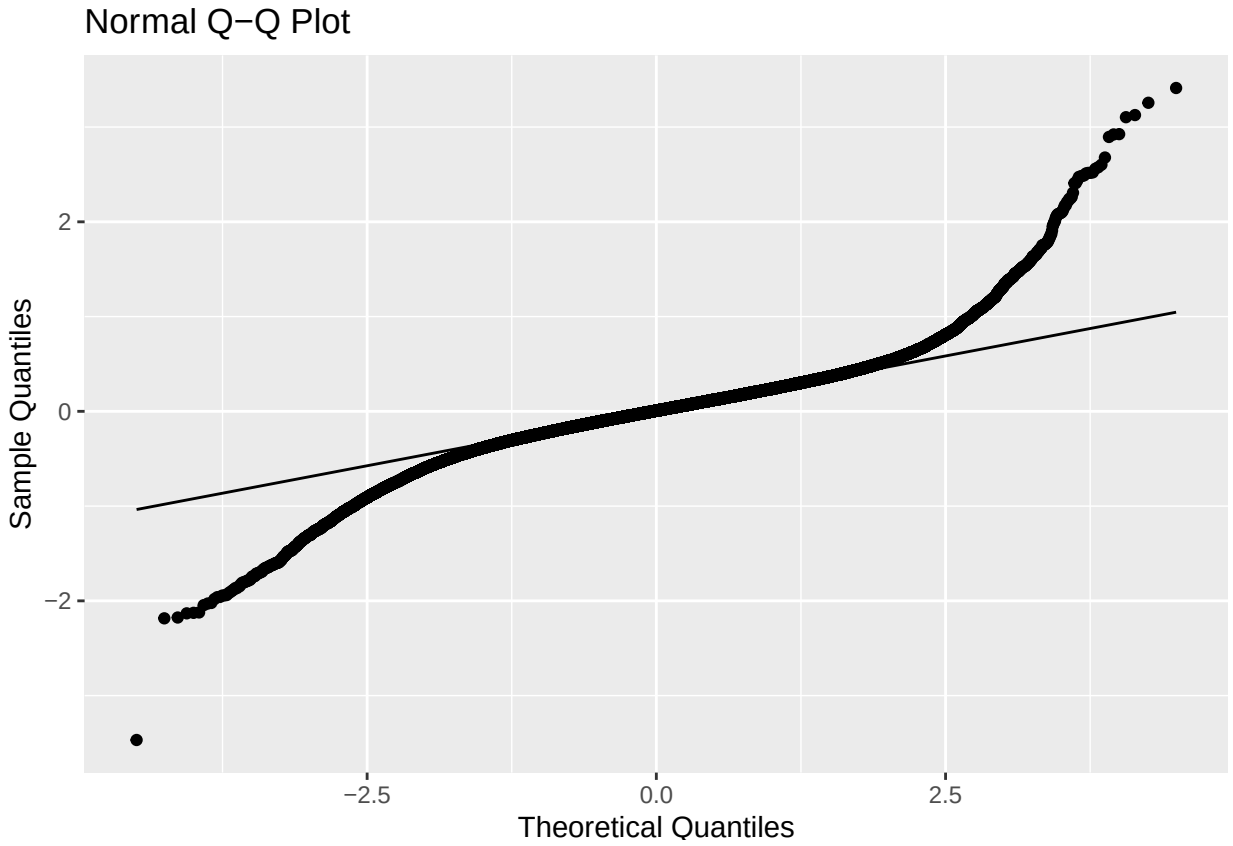
```r
abline(h = 0, col = "red", lty = "dotted", lwd = 1.5) # add a horizontal line at 0
lowess_fit <- lowess(fitted_values, residuals)
lines(lowess_fit, col = "royalblue", lwd = 1.5) # add lowess line
```




```r
# create histogram of residuals
hist(model_1$residuals, main = "Residual Histogram")
```




```
# create normal Q-Q plot
plot_data <- data.frame(Residuals = residuals, Fitted = fitted_values)
ggplot(plot_data, aes(sample = Residuals)) +
  geom_qq() +
  geom_qq_line() +
  ggtitle("Normal Q-Q Plot") +
  xlab("Theoretical Quantiles") +
  ylab("Sample Quantiles")
```


Normal Q−Q Plot
Sample Quantiles
2
0
−2
−2.5
0.0
2.5
Theoretical Quantiles

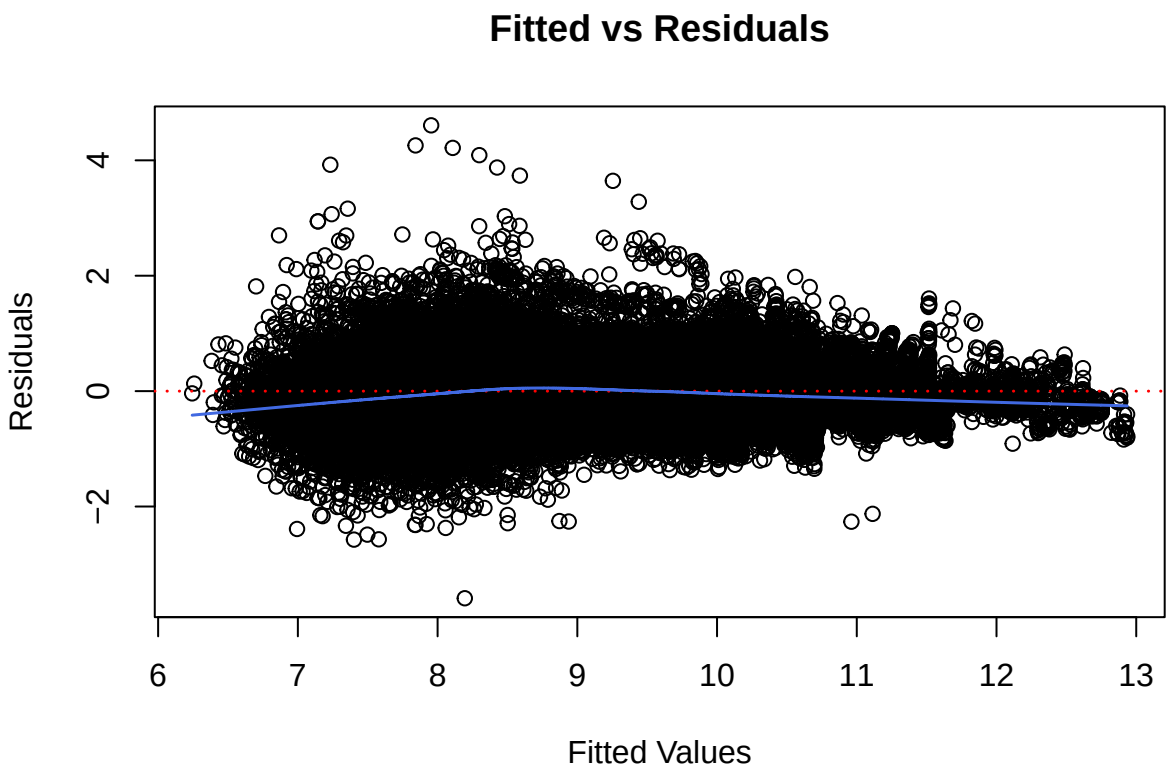
Linear Model 2

```r
# Model 2 - simpler model with only make, mileage, and year
model_2 <- lm(log_Price ~ Runned_Miles + Reg_year + Maker,
              data = df_training)

# View the model summary
summary(model_2)
```

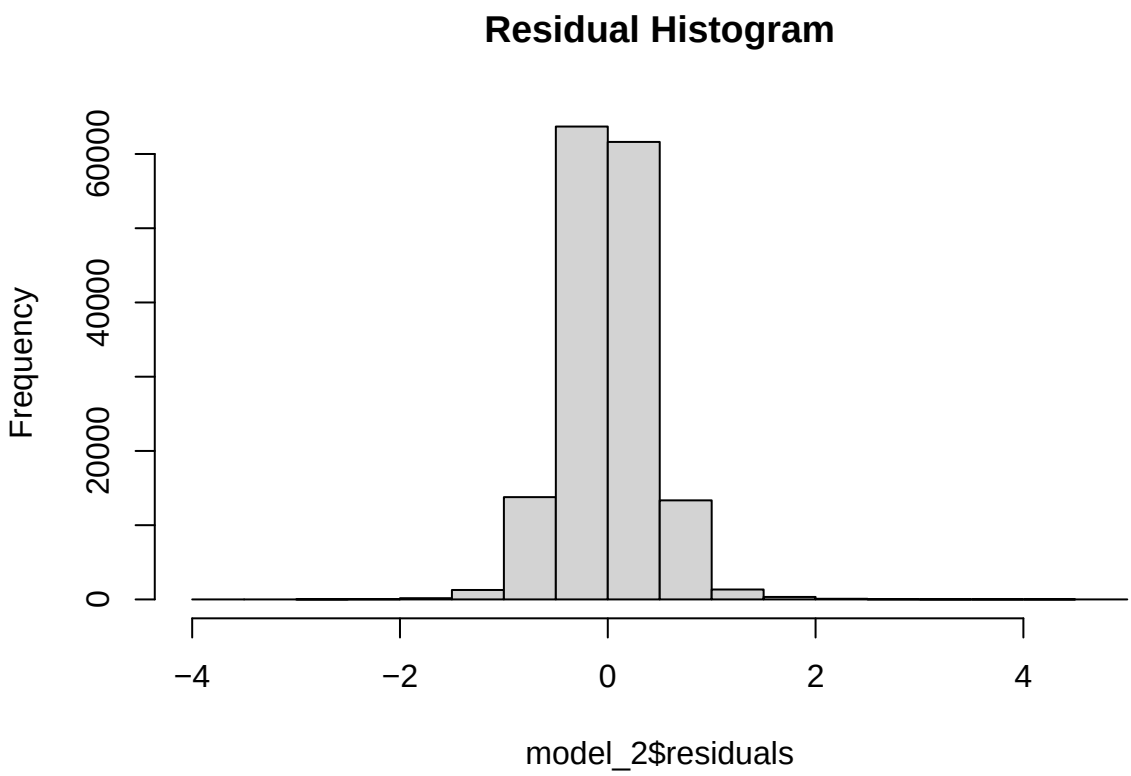
```
##
## Call:
## lm(formula = log_Price ~ Runned_Miles + Reg_year + Maker, data = df_training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5895 -0.2548 -0.0063  0.2524  4.6062
##
## Coefficients:
##                       Estimate Std. Error  t value Pr(>|t|)
## (Intercept)         -2.580e+02  7.715e-01 -334.393  < 2e-16 ***
## Runned_Miles        -5.217e-06  4.452e-08 -117.184  < 2e-16 ***
## Reg_year             1.328e-01  3.826e-04  347.107  < 2e-16 ***
## MakerAbarth         -2.518e-01  2.734e-02   -9.210  < 2e-16 ***
## MakerAlfa Romeo     -6.605e-02  1.638e-02   -4.032 5.53e-05 ***
## MakerAston Martin    2.204e+00  4.913e-02   44.856  < 2e-16 ***
## MakerAudi            5.293e-01  6.466e-03   81.860  < 2e-16 ***
## MakerBentley         2.131e+00  1.718e-02  124.009  < 2e-16 ***
## MakerBMW             6.086e-01  6.873e-03   88.542  < 2e-16 ***
## MakerChevrolet      -6.697e-01  2.112e-02  -31.709  < 2e-16 ***
## MakerChrysler       -5.913e-02  2.803e-02   -2.110   0.0349 *
## MakerCitroen        -4.047e-01  7.703e-03  -52.538  < 2e-16 ***
## MakerDacia          -5.976e-01  1.406e-02  -42.496  < 2e-16 ***
## MakerDaihatsu       -2.633e-01  5.577e-02   -4.722 2.34e-06 ***
## MakerDodge          -2.479e-02  4.913e-02   -0.505   0.6139
## MakerDS             -3.092e-01  2.119e-02  -14.595  < 2e-16 ***
## MakerFerrari         2.820e+00  2.278e-02  123.778  < 2e-16 ***
## MakerFiat           -5.635e-01  8.627e-03  -65.315  < 2e-16 ***
## MakerFord           -2.585e-01  6.291e-03  -41.087  < 2e-16 ***
## MakerHonda           3.932e-02  9.274e-03    4.240 2.24e-05 ***
## MakerHyundai        -3.144e-01  8.578e-03  -36.650  < 2e-16 ***
## MakerInfiniti        1.650e-01  2.643e-02    6.241 4.35e-10 ***
## MakerIsuzu           4.093e-01  1.925e-02   21.258  < 2e-16 ***
## MakerJaguar          5.475e-01  8.036e-03   68.134  < 2e-16 ***
## MakerJeep            1.247e-01  1.422e-02    8.774  < 2e-16 ***
## MakerKia            -2.746e-01  7.992e-03  -34.363  < 2e-16 ***
## MakerLamborghini     2.770e+00  3.124e-02   88.669  < 2e-16 ***
## MakerLand Rover      1.052e+00  7.651e-03  137.526  < 2e-16 ***
## MakerLexus           5.712e-01  1.092e-02   52.323  < 2e-16 ***
## MakerLotus           1.313e+00  7.660e-02   17.137  < 2e-16 ***
## MakerMaserati        1.108e+00  2.303e-02   48.095  < 2e-16 ***
## MakerMazda          -1.435e-01  8.457e-03  -16.963  < 2e-16 ***
## MakerMcLaren         2.274e+00  3.735e-02   60.889  < 2e-16 ***
## MakerMercedes-Benz   5.762e-01  7.938e-03   72.597  < 2e-16 ***
## MakerMG             -5.208e-01  2.047e-02  -25.446  < 2e-16 ***
```

```
## MakerMINI          -1.147e-02  1.080e-02   -1.062   0.2882
## MakerMitsubishi     6.118e-02  1.079e-02    5.668 1.45e-08 ***
## MakerNissan        -1.209e-01  7.105e-03  -17.022  < 2e-16 ***
## MakerPeugeot       -4.121e-01  7.412e-03  -55.606  < 2e-16 ***
## MakerPorsche        1.534e+00  9.315e-03  164.709  < 2e-16 ***
## MakerRenault       -4.555e-01  8.188e-03  -55.632  < 2e-16 ***
## MakerRolls-Royce    2.629e+00  7.183e-02   36.598  < 2e-16 ***
## MakerRover         -6.316e-01  4.002e-02  -15.781  < 2e-16 ***
## MakerSaab          -2.304e-01  1.864e-02  -12.363  < 2e-16 ***
## MakerSEAT          -2.606e-01  1.055e-02  -24.694  < 2e-16 ***
## MakerSKODA         -2.631e-01  8.371e-03  -31.426  < 2e-16 ***
## MakerSmart         -6.319e-01  1.993e-02  -31.699  < 2e-16 ***
## MakerSsangyong     -9.954e-02  1.850e-02   -5.382 7.39e-08 ***
## MakerSubaru         3.914e-01  1.535e-02   25.500  < 2e-16 ***
## MakerSuzuki        -4.694e-01  1.125e-02  -41.728  < 2e-16 ***
## MakerTesla          1.762e+00  4.115e-01    4.281 1.86e-05 ***
## MakerVauxhall      -4.940e-01  6.564e-03  -75.264  < 2e-16 ***
## MakerVolkswagen     2.761e-02  6.753e-03    4.088 4.35e-05 ***
## MakerVolvo          2.559e-01  8.119e-03   31.523  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4114 on 155653 degrees of freedom
## Multiple R-squared:  0.8317, Adjusted R-squared:  0.8316
## F-statistic: 1.451e+04 on 53 and 155653 DF,  p-value: < 2.2e-16
```

```r
# create fitted vs residuals plot
residuals <- residuals(model_2)
fitted_values <- fitted(model_2)
plot(fitted_values, residuals, xlab = "Fitted Values", ylab = "Residuals", main = "Fitted vs Residuals")
abline(h = 0, col = "red", lty = "dotted", lwd = 1.5) # add a horizontal line at 0
lowess_fit <- lowess(fitted_values, residuals)
lines(lowess_fit, col = "royalblue", lwd = 1.5) # add lowess line
```

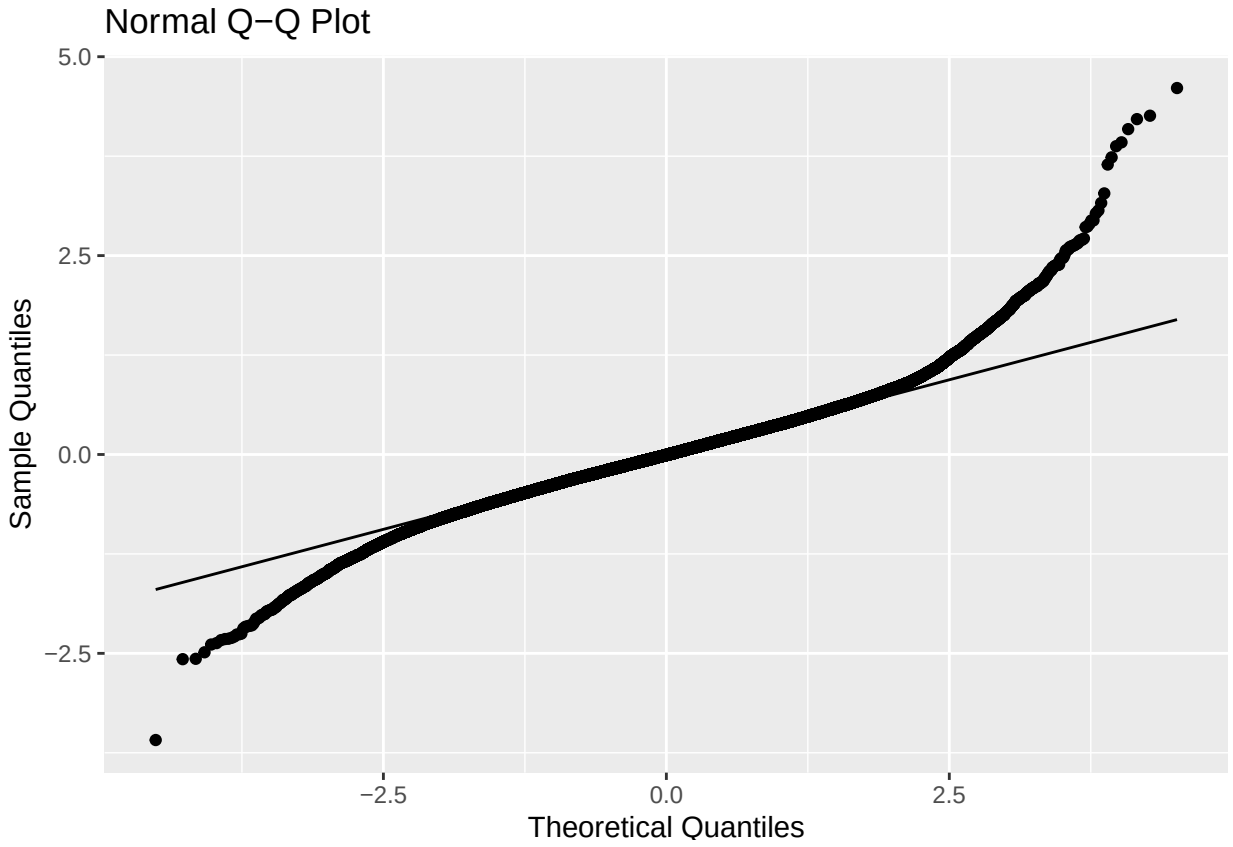



```r
# create histogram of residuals
hist(model_2$residuals, main = "Residual Histogram")
```




```r
# create normal Q-Q plot
plot_data <- data.frame(Residuals = residuals, Fitted = fitted_values)
ggplot(plot_data, aes(sample = Residuals)) +
```

```
  geom_qq() +
  geom_qq_line() +
  ggtitle("Normal Q-Q Plot") +
  xlab("Theoretical Quantiles") +
  ylab("Sample Quantiles")
```



Linear Model 3

```r
# Model 3 - Model 1 but without engine size or color because I believe they have a negligible effect on
model_3 <- lm(log_Price ~ Runned_Miles + Reg_year + Bodytype + Maker + Fuel_type
              + Gearbox,
              data = df_training)

# View the model summary
summary(model_3)
```

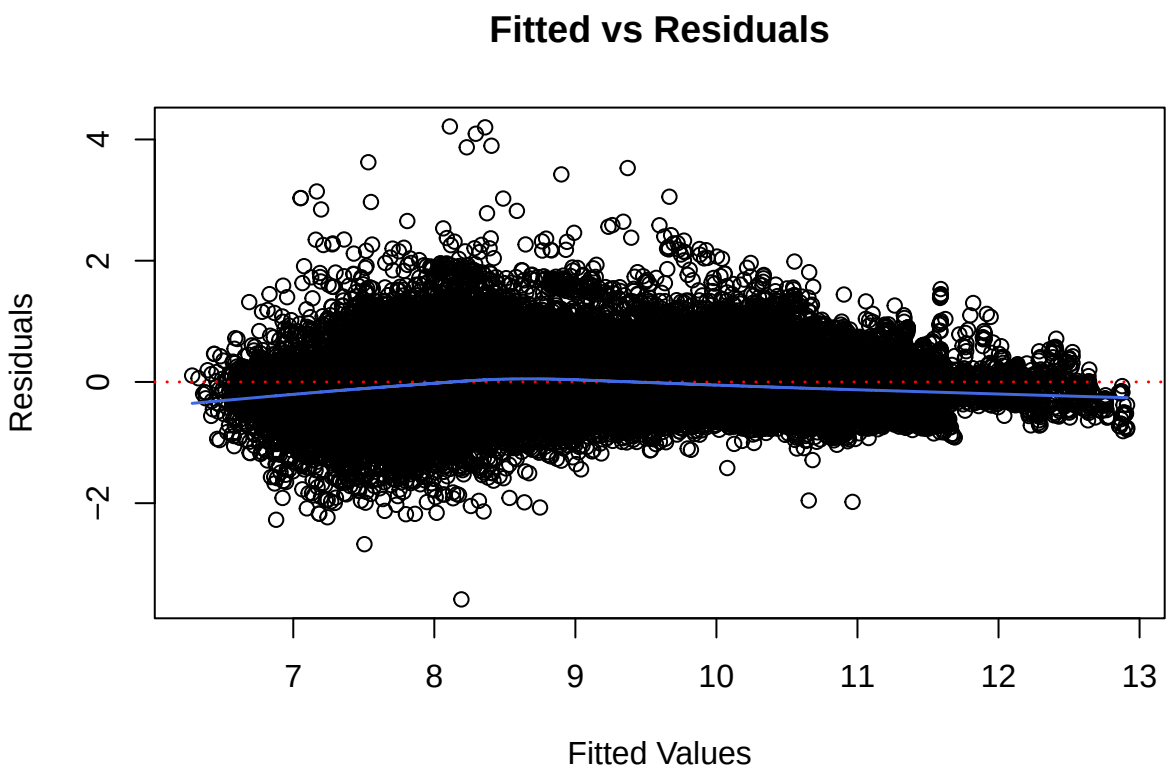
```
##
## Call:
## lm(formula = log_Price ~ Runned_Miles + Reg_year + Bodytype +
##     Maker + Fuel_type + Gearbox, data = df_training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5867 -0.1926  0.0000  0.1825  4.2137
##
## Coefficients:
##                                       Estimate Std. Error  t value
## (Intercept)                         -2.265e+02  6.735e-01 -336.311
## Runned_Miles                        -6.070e-06  3.868e-08 -156.935
## Reg_year                             1.172e-01  3.341e-04  350.777
## BodytypeConvertible                  2.388e-01  5.295e-03   45.095
## BodytypeCoupe                        3.565e-01  4.950e-03   72.018
## BodytypeEstate                       6.743e-02  4.846e-03   13.916
## BodytypeHatchback                   -1.501e-01  3.981e-03  -37.701
## BodytypeMPV                          5.552e-02  4.785e-03   11.602
## BodytypePickup                       5.465e-01  7.935e-03   68.870
## BodytypeSUV                          2.600e-01  3.963e-03   65.610
## MakerAbarth                          6.132e-02  2.245e-02    2.731
## MakerAlfa Romeo                      8.029e-02  1.355e-02    5.927
## MakerAston Martin                    1.870e+00  4.034e-02   46.348
## MakerAudi                            4.265e-01  5.768e-03   73.938
## MakerBentley                         1.800e+00  1.446e-02  124.474
## MakerBMW                             3.731e-01  6.146e-03   60.705
## MakerChevrolet                      -5.207e-01  1.735e-02  -30.005
## MakerChrysler                       -1.430e-01  2.301e-02   -6.214
## MakerCitroen                        -2.749e-01  6.616e-03  -41.545
## MakerDacia                          -4.484e-01  1.169e-02  -38.343
## MakerDaihatsu                       -2.604e-01  4.561e-02   -5.708
## MakerDodge                          -1.171e-01  4.020e-02   -2.913
## MakerDS                             -5.237e-02  1.746e-02   -3.000
## MakerFerrari                         2.467e+00  1.905e-02  129.505
## MakerFiat                           -3.267e-01  7.319e-03  -44.635
## MakerFord                           -1.371e-01  5.453e-03  -25.142
## MakerHonda                           3.889e-02  7.801e-03    4.985
## MakerHyundai                        -2.002e-01  7.243e-03  -27.641
## MakerInfiniti                        5.563e-02  2.180e-02    2.551
## MakerIsuzu                          -1.229e-01  1.709e-02   -7.189
## MakerJaguar                          3.596e-01  7.400e-03   48.591
## MakerJeep                           -1.129e-01  1.193e-02   -9.457
## MakerKia                            -1.799e-01  6.757e-03  -26.624
```

```
## MakerLamborghini                              2.432e+00  2.583e-02   94.173
## MakerLand Rover                               6.712e-01  6.915e-03   97.069
## MakerLexus                                    2.518e-01  1.028e-02   24.504
## MakerLotus                                    1.197e+00  6.269e-02   19.101
## MakerMaserati                                 8.780e-01  1.921e-02   45.703
## MakerMazda                                   -2.159e-02  7.184e-03   -3.006
## MakerMcLaren                                  1.926e+00  3.080e-02   62.553
## MakerMercedes-Benz                            3.419e-01  7.024e-03   48.678
## MakerMG                                      -3.771e-01  1.687e-02  -22.355
## MakerMINI                                     9.811e-02  9.120e-03   10.758
## MakerMitsubishi                              -1.768e-01  9.246e-03  -19.128
## MakerNissan                                  -1.519e-01  6.150e-03  -24.697
## MakerPeugeot                                 -3.264e-01  6.366e-03  -51.275
## MakerPorsche                                  1.262e+00  8.177e-03  154.319
## MakerRenault                                 -3.364e-01  6.957e-03  -48.357
## MakerRolls-Royce                              2.350e+00  5.880e-02   39.957
## MakerRover                                   -6.220e-01  3.284e-02  -18.942
## MakerSaab                                    -3.179e-01  1.550e-02  -20.511
## MakerSEAT                                    -1.046e-01  8.830e-03  -11.845
## MakerSKODA                                   -1.218e-01  7.121e-03  -17.101
## MakerSmart                                   -6.371e-01  1.655e-02  -38.504
## MakerSsangyong                               -3.348e-01  1.531e-02  -21.869
## MakerSubaru                                   3.797e-01  1.274e-02   29.807
## MakerSuzuki                                  -3.222e-01  9.423e-03  -34.190
## MakerTesla                                    1.531e+00  3.489e-01    4.388
## MakerVauxhall                                -2.829e-01  5.700e-03  -49.641
## MakerVolkswagen                               3.027e-02  5.849e-03    5.175
## MakerVolvo                                    1.280e-01  7.052e-03   18.154
## Fuel_typeDiesel                               1.539e-01  2.135e-03   72.080
## Fuel_typeElectric                             3.931e-01  9.327e-02    4.215
## Fuel_typeHybrid  Petrol/Electric              2.621e-01  8.530e-03   30.723
## Fuel_typeHybrid  Petrol/Electric Plug-in      4.473e-01  1.558e-02   28.705
## Fuel_typePetrol Hybrid                       -1.192e-01  2.349e-02   -5.074
## GearboxManual                                -2.563e-01  2.235e-03 -114.698
##                                              Pr(>|t|)
## (Intercept)                                   < 2e-16 ***
## Runned_Miles                                  < 2e-16 ***
## Reg_year                                      < 2e-16 ***
## BodytypeConvertible                           < 2e-16 ***
## BodytypeCoupe                                 < 2e-16 ***
## BodytypeEstate                                < 2e-16 ***
## BodytypeHatchback                             < 2e-16 ***
## BodytypeMPV                                   < 2e-16 ***
## BodytypePickup                                < 2e-16 ***
## BodytypeSUV                                   < 2e-16 ***
## MakerAbarth                                   0.00631 **
## MakerAlfa Romeo                              3.09e-09 ***
## MakerAston Martin                             < 2e-16 ***
## MakerAudi                                     < 2e-16 ***
## MakerBentley                                  < 2e-16 ***
## MakerBMW                                      < 2e-16 ***
## MakerChevrolet                                < 2e-16 ***
## MakerChrysler                                5.18e-10 ***
## MakerCitroen                                  < 2e-16 ***
```

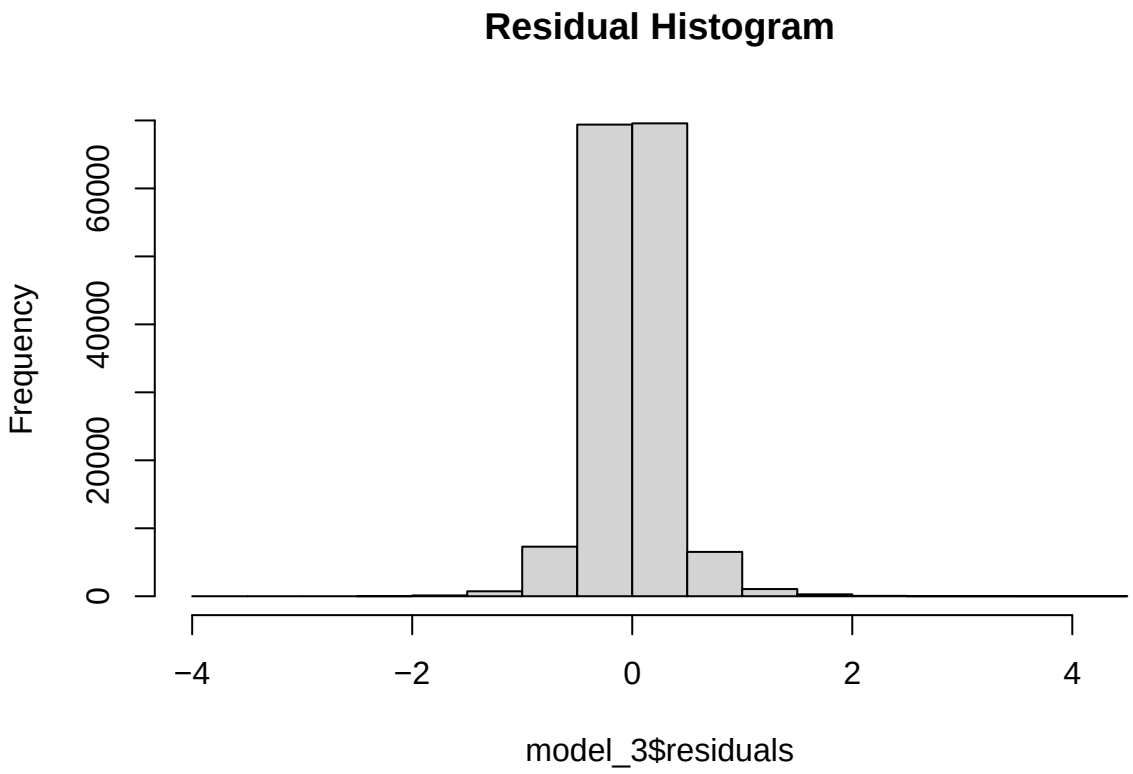
```
## MakerDacia                                  < 2e-16 ***
## MakerDaihatsu                              1.15e-08 ***
## MakerDodge                                  0.00358 **
## MakerDS                                     0.00270 **
## MakerFerrari                                < 2e-16 ***
## MakerFiat                                   < 2e-16 ***
## MakerFord                                   < 2e-16 ***
## MakerHonda                                 6.20e-07 ***
## MakerHyundai                                < 2e-16 ***
## MakerInfiniti                               0.01073 *
## MakerIsuzu                                 6.53e-13 ***
## MakerJaguar                                 < 2e-16 ***
## MakerJeep                                   < 2e-16 ***
## MakerKia                                    < 2e-16 ***
## MakerLamborghini                            < 2e-16 ***
## MakerLand Rover                             < 2e-16 ***
## MakerLexus                                  < 2e-16 ***
## MakerLotus                                  < 2e-16 ***
## MakerMaserati                               < 2e-16 ***
## MakerMazda                                  0.00265 **
## MakerMcLaren                                < 2e-16 ***
## MakerMercedes-Benz                          < 2e-16 ***
## MakerMG                                     < 2e-16 ***
## MakerMINI                                   < 2e-16 ***
## MakerMitsubishi                             < 2e-16 ***
## MakerNissan                                 < 2e-16 ***
## MakerPeugeot                                < 2e-16 ***
## MakerPorsche                                < 2e-16 ***
## MakerRenault                                < 2e-16 ***
## MakerRolls-Royce                            < 2e-16 ***
## MakerRover                                  < 2e-16 ***
## MakerSaab                                   < 2e-16 ***
## MakerSEAT                                   < 2e-16 ***
## MakerSKODA                                  < 2e-16 ***
## MakerSmart                                  < 2e-16 ***
## MakerSsangyong                              < 2e-16 ***
## MakerSubaru                                 < 2e-16 ***
## MakerSuzuki                                 < 2e-16 ***
## MakerTesla                                 1.15e-05 ***
## MakerVauxhall                               < 2e-16 ***
## MakerVolkswagen                            2.28e-07 ***
## MakerVolvo                                  < 2e-16 ***
## Fuel_typeDiesel                             < 2e-16 ***
## Fuel_typeElectric                          2.50e-05 ***
## Fuel_typeHybrid  Petrol/Electric            < 2e-16 ***
## Fuel_typeHybrid  Petrol/Electric Plug-in    < 2e-16 ***
## Fuel_typePetrol Hybrid                     3.89e-07 ***
## GearboxManual                               < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3362 on 155001 degrees of freedom
##   (639 observations deleted due to missingness)
## Multiple R-squared:  0.8876, Adjusted R-squared:  0.8876
```

```
## F-statistic: 1.855e+04 on 66 and 155001 DF,  p-value: < 2.2e-16
```

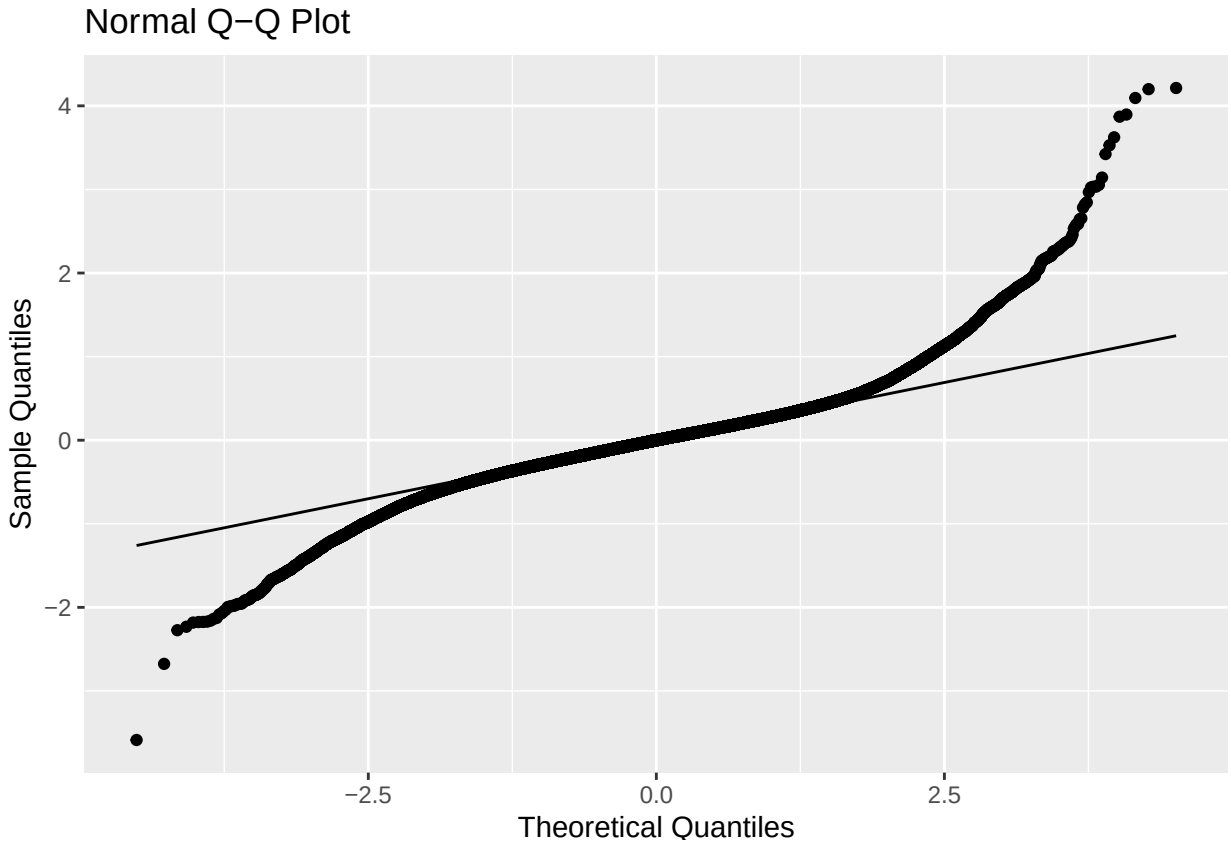
```r
# create fitted vs residuals plot
residuals <- residuals(model_3)
fitted_values <- fitted(model_3)
plot(fitted_values, residuals, xlab = "Fitted Values", ylab = "Residuals", main = "Fitted vs Residuals")
abline(h = 0, col = "red", lty = "dotted", lwd = 1.5) # add a horizontal line at 0
lowess_fit <- lowess(fitted_values, residuals)
lines(lowess_fit, col = "royalblue", lwd = 1.5) # add lowess line
```




```r
# create histogram of residuals
hist(model_3$residuals, main = "Residual Histogram")
```

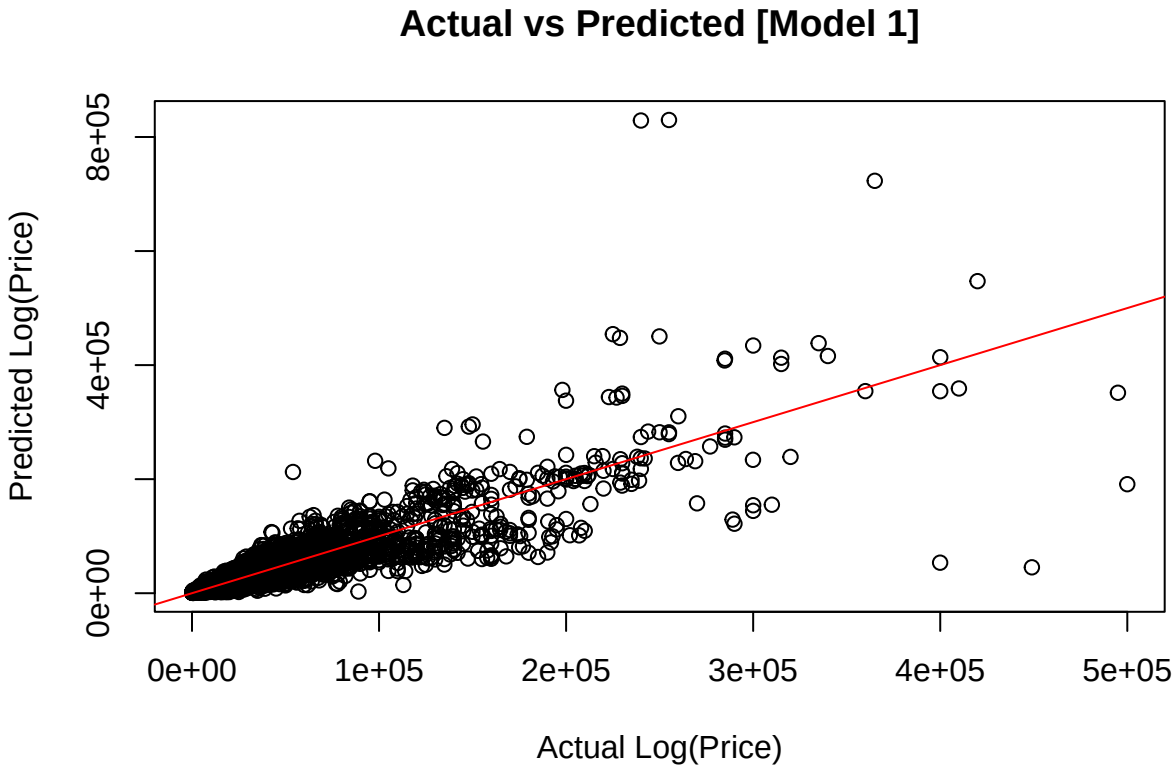



```r
# create normal Q-Q plot
plot_data <- data.frame(Residuals = residuals, Fitted = fitted_values)
ggplot(plot_data, aes(sample = Residuals)) +
  geom_qq() +
  geom_qq_line() +
  ggtitle("Normal Q-Q Plot") +
  xlab("Theoretical Quantiles") +
  ylab("Sample Quantiles")
```


Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles
4
2
0
−2
−2.5
0.0
2.5

Make Predictions and Evaluate Model

```
# predict models on validation set
predictions_1 <- predict(model_1, newdata = df_validation)
predictions_2 <- predict(model_2, newdata = df_validation)
predictions_3 <- predict(model_3, newdata = df_validation)

# get the actual values that the predictions correspond to
actual_values <- df_validation$log_Price

# plot actual vs predicted log_Price for Model 1
plot(exp(actual_values), exp(predictions_1), main = "Actual vs Predicted [Model 1]",
     xlab = "Actual Log(Price)", ylab = "Predicted Log(Price)")
abline(a = 0, b = 1, col='red')
```



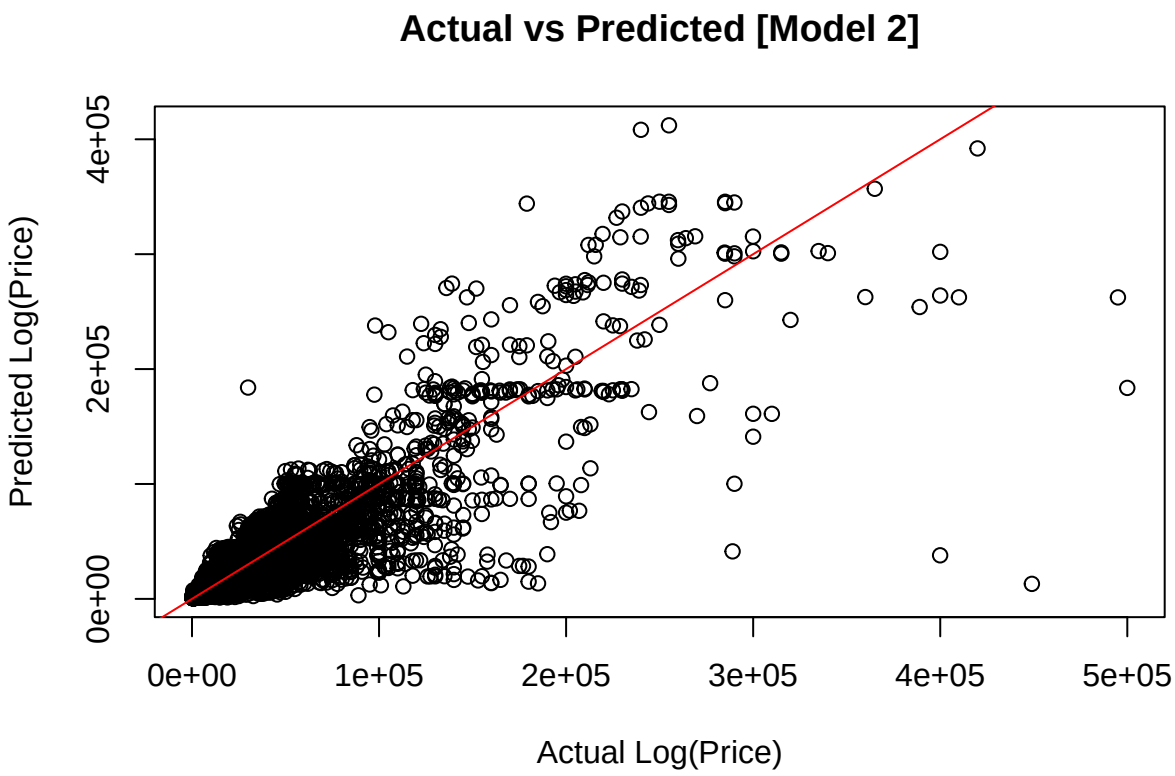

```
# Calculate performance metrics for Model 1
mae1 <- mean(abs(exp(predictions_1) - exp(actual_values)), na.rm = TRUE)
rmse1 <- sqrt(mean((exp(predictions_1) - exp(actual_values))^2, na.rm = TRUE))
cat(paste("Model 1 - Mean Absolute Error (MAE):", format(mae1, nsmall = 4)), "\n")
```

```
## Model 1 - Mean Absolute Error (MAE): 2924.6159
```

```
cat(paste("Model 1 - Root Mean Squared Error (RMSE):", format(rmse1, nsmall = 4)), "\n")
```

```
## Model 1 - Root Mean Squared Error (RMSE): 8816.9185
```

```
# plot actual vs predicted log_Price for Model 2
plot(exp(actual_values), exp(predictions_2), main = "Actual vs Predicted [Model 2]",
     xlab = "Actual Log(Price)", ylab = "Predicted Log(Price)")
abline(a = 0, b = 1, col='red')
```



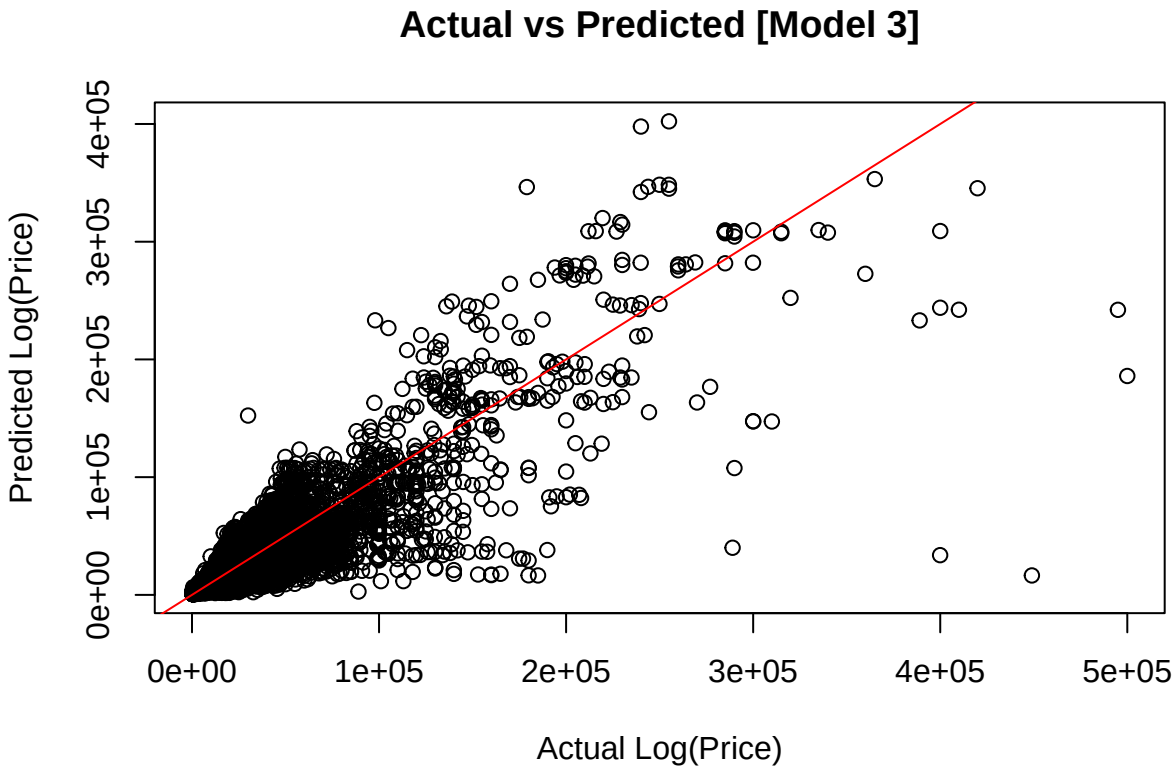

```
# Calculate performance metrics for Model 2
mae2 <- mean(abs(exp(predictions_2) - exp(actual_values)), na.rm = TRUE)
rmse2 <- sqrt(mean((exp(predictions_2) - exp(actual_values))^2, na.rm = TRUE))
cat(paste("Model 2 - Mean Absolute Error (MAE):", format(mae2, nsmall = 4)), "\n")
```

```
## Model 2 - Mean Absolute Error (MAE): 4385.9555
```

```
cat(paste("Model 2 - Root Mean Squared Error (RMSE):", format(rmse2, nsmall = 4)), "\n")
```

```
## Model 2 - Root Mean Squared Error (RMSE): 9751.5000
```

```
# plot actual vs predicted log_Price for Model 3
plot(exp(actual_values), exp(predictions_3), main = "Actual vs Predicted [Model 3]",
     xlab = "Actual Log(Price)", ylab = "Predicted Log(Price)")
abline(a = 0, b = 1, col='red')
```




```
# Calculate performance metrics for Model 3
mae3 <- mean(abs(exp(predictions_3) - exp(actual_values)), na.rm = TRUE)
rmse3 <- sqrt(mean((exp(predictions_3) - exp(actual_values))^2, na.rm = TRUE))
cat(paste("Model 3 - Mean Absolute Error (MAE):", format(mae3, nsmall = 4)), "\n")
```

```
## Model 3 - Mean Absolute Error (MAE): 3603.2432
```

```
cat(paste("Model 3 - Root Mean Squared Error (RMSE):", format(rmse3, nsmall = 4)), "\n")
```

```
## Model 3 - Root Mean Squared Error (RMSE): 9108.6796
```

```
# It seems that model 1 is the best model.
```

Test the Best Model

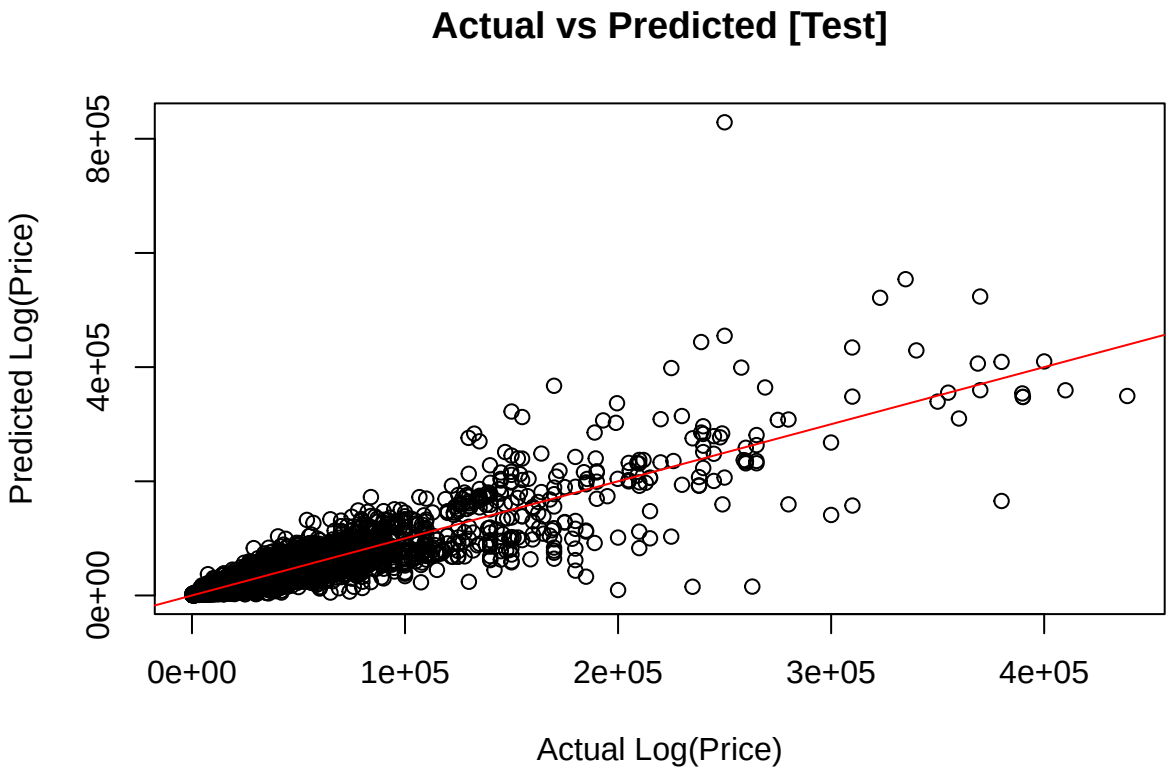
```
# make predictions on test set
predictions_test <- predict(model_1, newdata = df_test)

# get actual log_Price values from test set
actual_values_test <- df_test$log_Price

# Calculate performance metrics
mae_test <- mean(abs(exp(predictions_test) - exp(actual_values_test)), na.rm = TRUE)
rmse_test <- sqrt(mean((exp(predictions_test) - exp(actual_values_test))^2, na.rm = TRUE))
cat(paste("Test - Mean Absolute Error (MAE):", format(mae_test, nsmall = 4)), "\n")
```

```
## Test - Mean Absolute Error (MAE): 2892.9390
```

```
cat(paste("Test - Root Mean Squared Error (RMSE):", format(rmse_test, nsmall = 4)), "\n")
```

```
## Test - Root Mean Squared Error (RMSE): 7930.8646
```

```
# plot actual vs predicted log_Price for test data
plot(exp(actual_values_test), exp(predictions_test), main = "Actual vs Predicted [Test]",
     xlab = "Actual Log(Price)", ylab = "Predicted Log(Price)")
abline(a = 0, b = 1, col='red')
```



Template (ignore)

```
# The PDF will show the code you write here but not the output.
```

```
# The PDF will show the code AND output here.
```