Assignment 4

Will Rauen

2024-03-28

Setup

For this investigation, we will be using a multitude of packages to answer a research question related to multinomial regression.

Data

```
car_data <- read.csv("/Users/williamrauen/Desktop/DS 3100/Assignments/Assignment 4/Data/car_details_v4.

dim(car_data)
```

```
## [1] 2059   20
```

The data used in this dataset comes Kaggle and can be accessed at:

https://www.kaggle.com/datasets/nehalbirla/vehicle-dataset-from-cardekho

The data contains 2059 rows and 20 columns with each row representing a different vehicle and and each column representing a feature of the car. These columns include car model, brand, price, model-year, number of miles, cost of the car, fuel type, drivetrain, etc... which illustrates a combination of categorical and continuous data. There is also some missing data points contained in the data set which will be dealt with in the data wrangling portion of the assignment. The variable of interest in this dataset is the drivetrain variable which determines whether the vehicle has all-wheel drive, front-wheel drive, or rear-wheel drive. Since these classes are not continuous and not ranked, we can consider this variable as a nominal variable data making it viable to use in a multinomial regression.

Varaibles of Interest

The dependent variable of interest as aforementioned is the drivetrain variable which is a nominal class of three variables: all-wheel drive (AWD), front-wheel drive (FWD), or rear-wheel drive (RWD). The classes we want to use to predict the drivetrain of a car should be continuous in order to do a multinomial regression. Thus, the prediction variables we will be looking at this for regression will be price of the car ($), model year, number of kilometers on the car (km), engine size (cc), and fuel tank capacity (liters). This sets up a regression model that has a three class categorical variable predicted by 5 continuous variables which fits the requirements of a multinomial regression.

Research Question

In this assignment, I aim to determine if there are significant features that predict the drivetrain of a vehicle. If there does exist significant features, I aim to determine what specific features these are for each drivetrain.

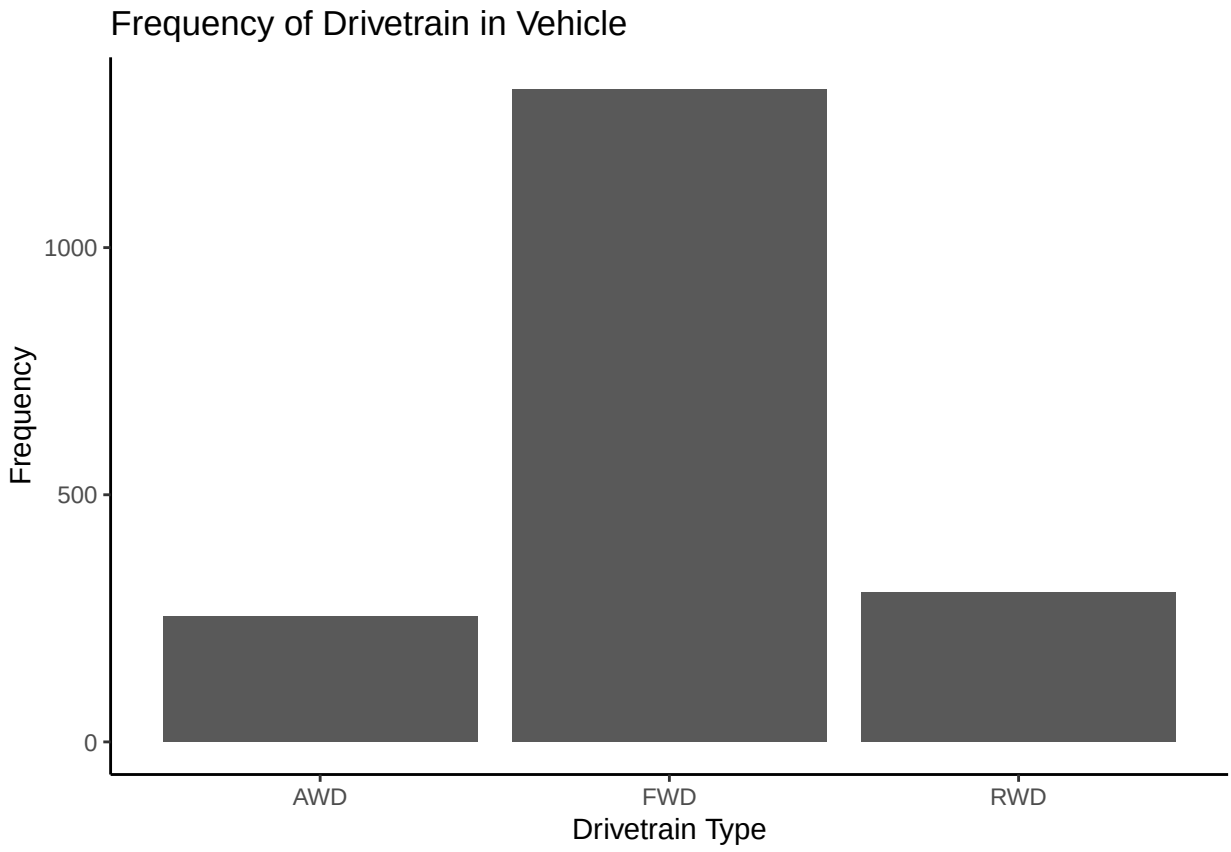
Data Wrangling

Firstly, we will remove rows that contain missing data and then count the frequency of each instance of the drivetrain classification

```
# remove missingness from data set

car_data <- car_data[complete.cases(car_data), ]

#accounts for absence of classification in Drivetrain variable

car_data <- car_data %>%
  filter(Drivetrain == "AWD" |
           Drivetrain == "FWD" |
           Drivetrain == "RWD")

dim(car_data)
```

```
## [1] 1874   20
```

```
car_data %>%
  ggplot(aes(x = Drivetrain)) +
  geom_bar() +
  labs(title = "Frequency of Drivetrain in Vehicle",
       x = "Drivetrain Type",
       y = "Frequency"
       ) +
  theme_classic()
```



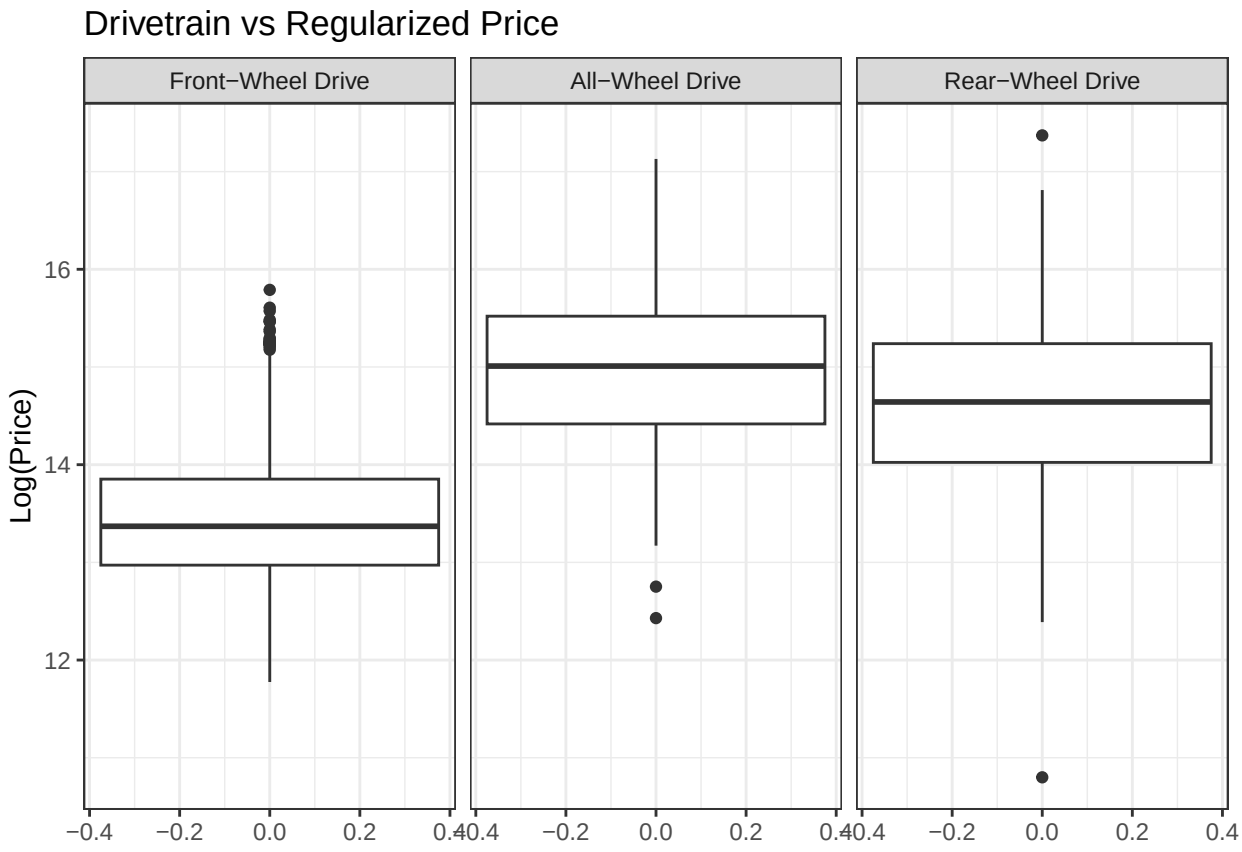
This reduces the number of rows we have for the data set down to 1874 rows from 2059. Nonetheless, this is still a large enough sample size to proceed with our investigation as it remains statistically significant. Furthermore, while most vehicle drivetrains have a front-wheel drive, there are enough counts of the other classifications to move forward with the investigation.

Next, I will subset the data so that only the relevant variables remain in the data and make sure each variable is the right type for this regression. Alongside this I will set the levels of the classification variable of Drivetrain by making front-wheel drive the reference category for our regression.

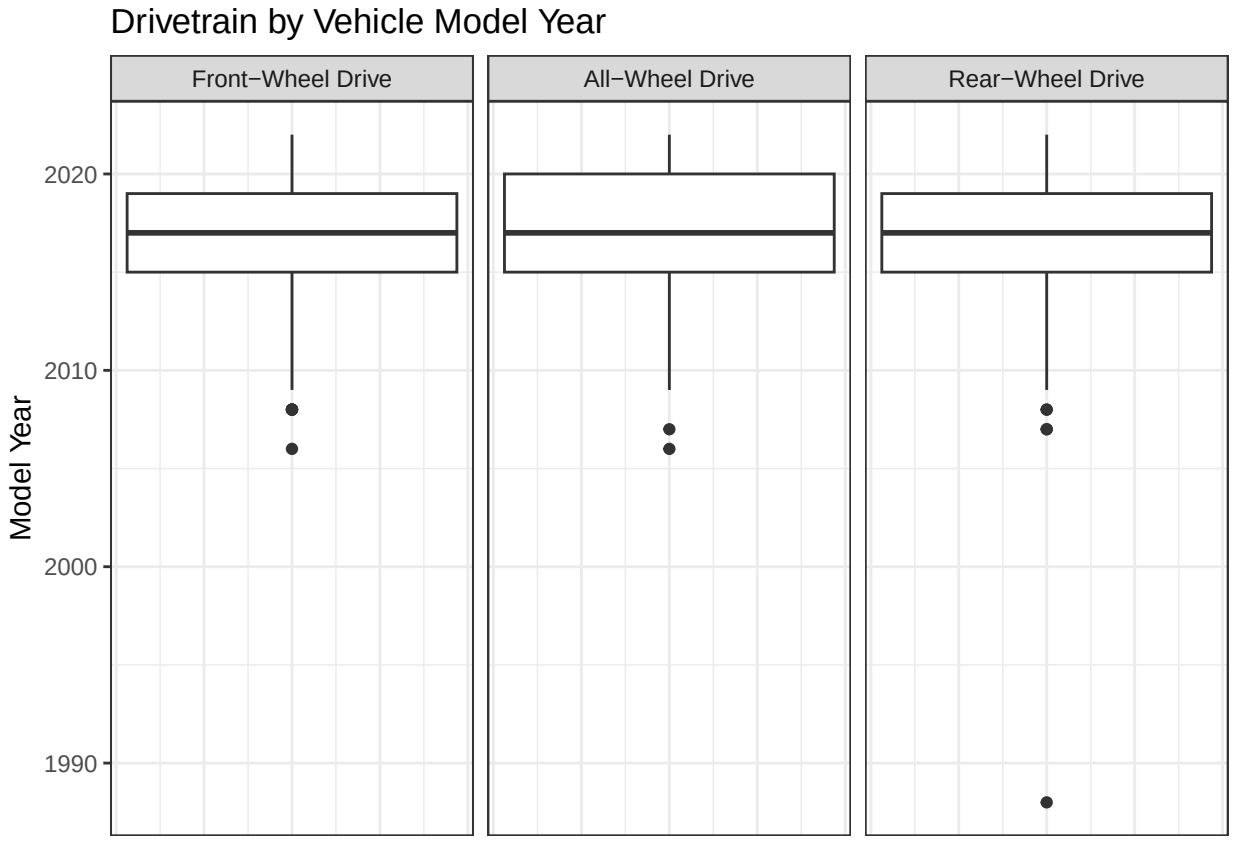
```
#Subsets data by each column
car_data <- car_data[c(3,4,5,12,15,20)]

car_data <- car_data %>% #converting to correct class types
  mutate(Price = as.numeric(Price)) %>%
  mutate(Year = as.numeric(Year)) %>%
  mutate(Kilometer = as.numeric(Kilometer)) %>%
  mutate(Fuel_Tank = as.numeric(Fuel.Tank.Capacity)) %>%
  mutate(Engine = gsub("cc", "", as.character(Engine))) %>% #removes letters
  mutate(Engine = as.numeric(Engine)) #changes type from chr. to num. class


# Set front-wheel drive as reference category for investigation
car_data$Drivetrain <- factor(car_data$Drivetrain,
                         levels = c('FWD', 'AWD', 'RWD'))
```

Lastly, lets visualize some of these variables by their relation to the class of drivetrain and see if there are any readily apparent relationships.

```
#Labels for data
custom_labels <- labeller(Drivetrain = c("AWD" = "All-Wheel Drive",
                                         "FWD" = "Front-Wheel Drive",
                                         "RWD" = "Rear-Wheel Drive"))

#Visualizing each of the relationships

car_data %>%
  ggplot(aes(x = log(Price))) +
  geom_boxplot() +
  coord_flip() +
  facet_wrap(~Drivetrain, labeller = custom_labels) +
  theme_bw() +
  labs(title = "Drivetrain vs Regularized Price",
       x = "Log(Price)")
```

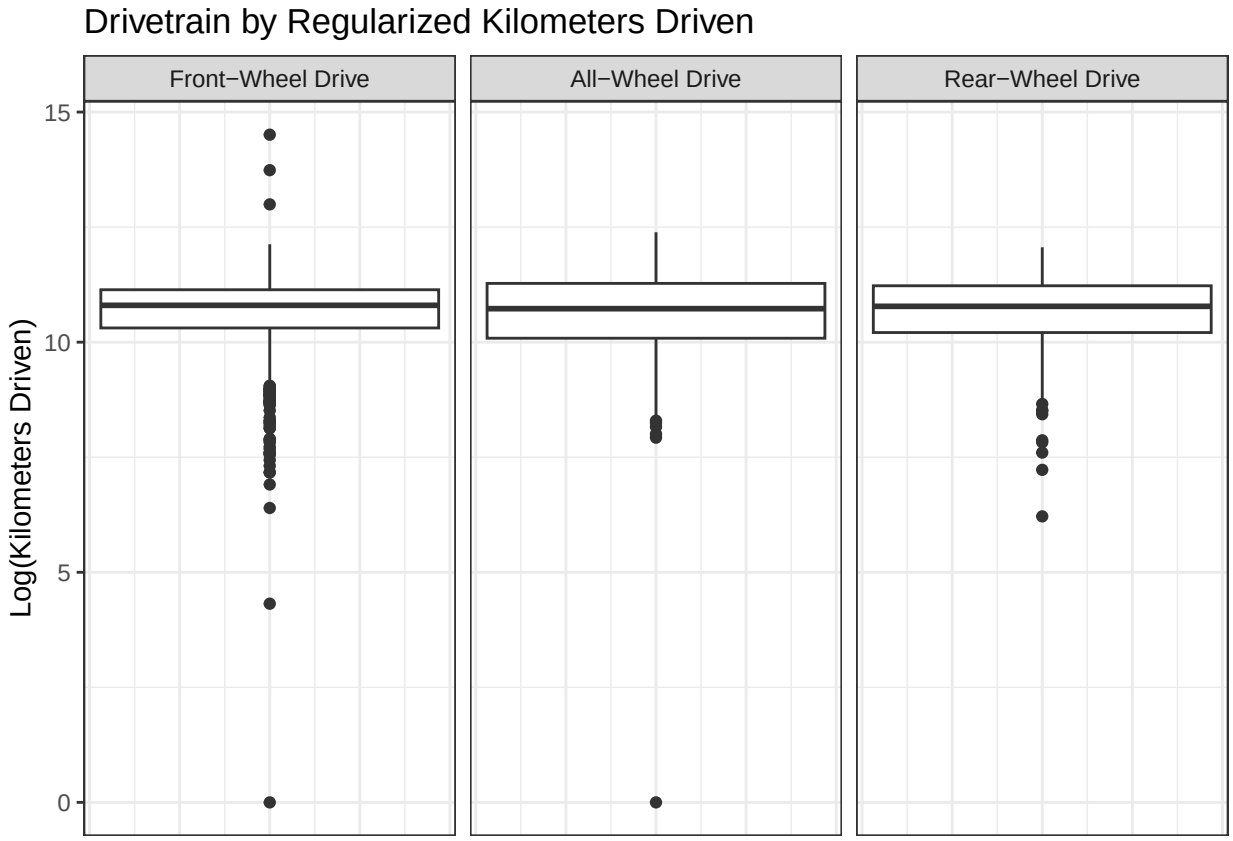



```
car_data %>%
  ggplot(aes(x = Year)) +
  geom_boxplot() +
  coord_flip() +
  facet_wrap(~Drivetrain, labeller = custom_labels) +
  theme_bw() +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank()) +
  labs(title = "Drivetrain by Vehicle Model Year",
       x = "Model Year")
```

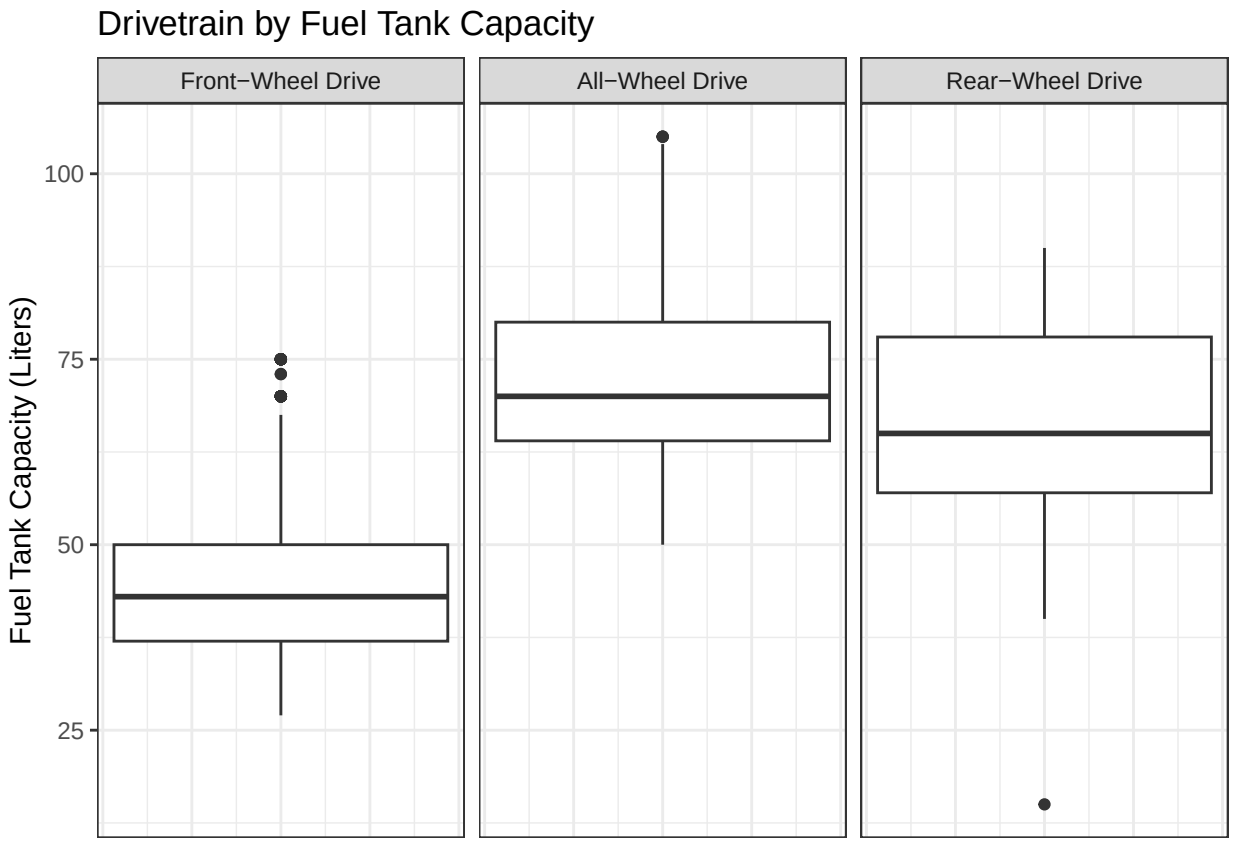



```
car_data %>%
  ggplot(aes(x = log(Kilometer))) +
  geom_boxplot() +
  coord_flip() +
  facet_wrap(~Drivetrain, labeller = custom_labels) +
  theme_bw() +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank()) +
  labs(title = "Drivetrain by Regularized Kilometers Driven",
       x = "Log(Kilometers Driven)")
```

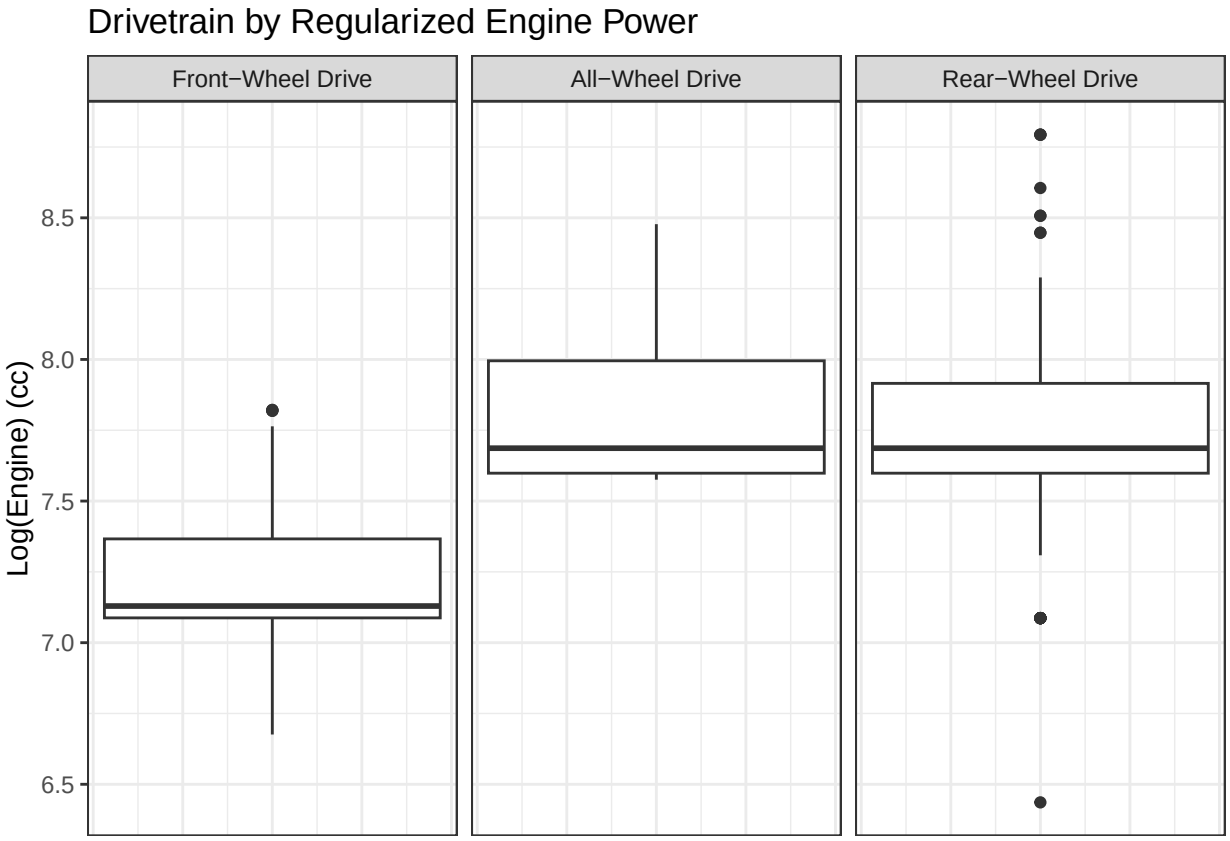
```
## Warning: Removed 1 rows containing non-finite values (`stat_boxplot()`).
```




```
car_data %>%
  ggplot(aes(x = Fuel_Tank)) +
  geom_boxplot() +
  coord_flip() +
  facet_wrap(~Drivetrain, labeller = custom_labels) +
  theme_bw() +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank()) +
  labs(title = "Drivetrain by Fuel Tank Capacity",
       x = "Fuel Tank Capacity (Liters)")
```




```
car_data %>%
  ggplot(aes(x = log(Engine))) +
  geom_boxplot() +
  coord_flip() +
  facet_wrap(~Drivetrain, labeller = custom_labels) +
  theme_bw() +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank()) +
  labs(title = "Drivetrain by Regularized Engine Power",
       x = "Log(Engine) (cc)")
```



The variables for Price, Engine, and Kilometers Driven were all skewed in the dataset so in order to "regularize" them I applied a logarithmic function to their values for the purpose of the data visualization. Despite this, the boxplots still revealed a plethora of outliers within the data set for each of these variables. Thus, it is difficult to determine if there are any blatantly apparent relationships between our prediction variable and outcome variable. As a result, the first step of our data analysis should be to first do some outlier detection to weed out heavily influencing data points.

Data Analysis

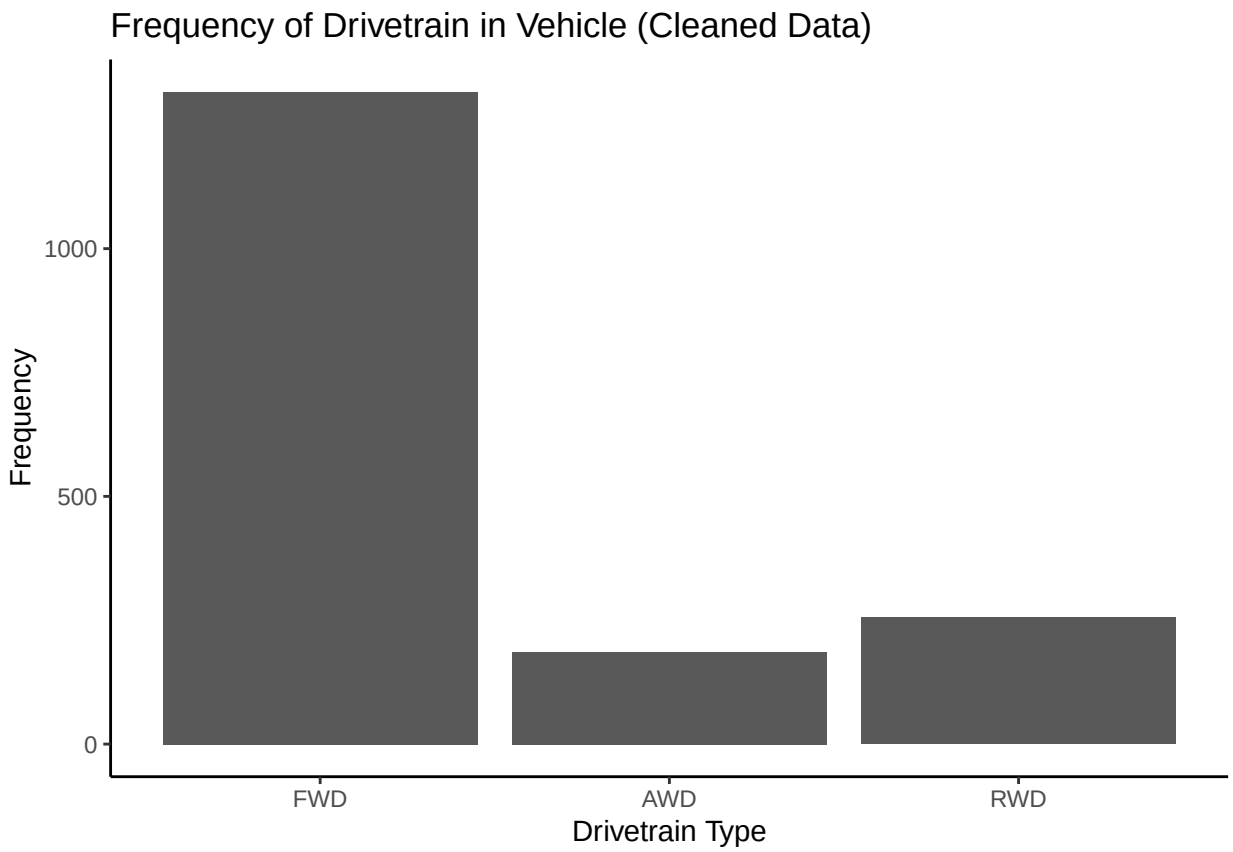
Outlier Detection

To determine the outliers of the data set, I scaled all the numerical data into their z-scores and used a condition that all varaibles with a z-score $> 1{,}5$ would be removed from the data set. Although the standard practice is to remove vehicles with z-scores greater than 2, there were still too many high influencing data points after re-examining the boxplots so I decided on this threshold for elimination.

```
#Finds z scores of numerical data
zscale <- scale(car_data[c(1,2,3,4,7)])

#Finds rows that violate threshold
outliers_z <- which(abs(zscale) > 1.5)

#Removes rows that violated threshold from original data set
car_data_clean <- car_data[-outliers_z, ]

# New
dim(car_data_clean)
```

```
## [1] 1758    7
```

```
car_data_clean %>%
  ggplot(aes(x = Drivetrain)) +
  geom_bar() +
  labs(title = "Frequency of Drivetrain in Vehicle (Cleaned Data)",
       x = "Drivetrain Type",
       y = "Frequency"
       ) +
  theme_classic()
```



Now we have 1758 rows of data and the frequency graph indicates that there are still more than 100 observations for each classification. Although this is not as many samples as we would like per classification, we can still move forward after completing this part of the data cleaning.

Model Creation and Interpretation

To create a more level playing for each of the classification of the Drivetrain variable, we can randomly sample some of the data from vehicles that have front-wheel drive and use that for our model instead of all the front-wheel drivetrain data.

```
#Dividing the data by classification
car_data_FWD <- car_data_clean %>%
  filter(Drivetrain == "FWD")

car_data_AWD <- car_data_clean %>%
  filter(Drivetrain == "AWD")
```

```
car_data_RWD <- car_data_clean %>%
  filter(Drivetrain == "RWD")

# Randomly sampling the Front-Wheel Drive Data
set.seed(123)

car_data_FWD_subset <- sample(1:nrow(car_data_FWD),
                              round(nrow(car_data_FWD) * 0.25),
                      replace = FALSE)

car_data_newFWD <- car_data_FWD[car_data_FWD_subset,]

#Creating new data set
car_data_new <- bind_rows(car_data_AWD, car_data_RWD, car_data_newFWD)

#Recount of each varaible instance
car_data_new %>%
  count(Drivetrain)
```

```
##   Drivetrain   n
## 1        FWD 329
## 2        AWD 186
## 3        RWD 256
```

This new data subset still has front-wheel drive vehicles as the majority but it is closer to the levels of the other two classification of drivetrains. From here, we can move forward with creating our model for the multinomial regression.

```
#Multinomial Function to create Model
car_mn <- multinom(Drivetrain~ Price + Engine + Kilometer + Fuel_Tank + Year,
                   data=car_data_new, model = TRUE)
```

```
## # weights:  21 (12 variable)
## initial  value 847.030075
## iter  10 value 516.873715
## iter  20 value 455.900675
## iter  30 value 455.889390
## iter  40 value 455.880110
## iter  50 value 455.641958
## final  value 455.641921
## converged
```

```
#summary of findings
tidy(car_mn)
```

```
## # A tibble: 12 x 6
##    y.level term             estimate    std.error statistic   p.value
##    <chr>   <chr>               <dbl>        <dbl>     <dbl>     <dbl>
##  1 AWD     (Intercept) -35.7         0.0000000312  -1.14e+9 0
##  2 AWD     Price         0.000000384 0.000000103    3.73e+0 1.88e-  4
##  3 AWD     Engine        0.00365     0.000128       2.85e+1 4.03e-178
##  4 AWD     Kilometer     0.00000454  0.00000377     1.20e+0 2.28e-  1
##  5 AWD     Fuel_Tank     0.153       0.00000112     1.37e+5 0
##  6 AWD     Year          0.00921     0.0000630      1.46e+2 0
##  7 RWD     (Intercept)  22.5         0.0000000803   2.80e+8 0
##  8 RWD     Price         0.000000219 0.000000108    2.02e+0 4.30e-  2
```

```
##  9 RWD     Engine        0.00382     0.0000673      5.68e+1 0
## 10 RWD     Kilometer     0.000000545 0.00000366     1.49e-1 8.81e-  1
## 11 RWD     Fuel_Tank     0.105       0.00000210     5.01e+4 0
## 12 RWD     Year         -0.0177      0.000162      -1.09e+2 0
```

The initial multinomial regression revealed that nearly all variables were statistically significant with p-values very close to zero. It tells us that all variables except the Kilometers variable has a significant impact on the log-odds of a vehicle being classified as rear-wheel drive compared to front-wheel drive. However, it is interesting to note that Price was the second least statistically significant variable. Similarly, the regression reveals all variables have a significant impact on the log-odds of a vehicle being classified as all-wheel drive with the least significant being the Price and Kilometers variables.

More specifically, the log odds show positive correlations with the variables of Price, Engine, Fuel_Tank, Year, and a negative correlation with the Kilometer variable for classifying a vehicle as all-wheel drive compared to front-wheel drive. For classifying rear-wheel drive, the log odds show positive correlations with the variables of Price, Engine, and Fuel Tank, and negative correlations with the variables of Year and Kilometers.

```
exp(coef(car_mn))
```

```
##      (Intercept) Price   Engine Kilometer Fuel_Tank     Year
## AWD 3.283649e-16     1 1.003657  1.000005  1.165902 1.009258
## RWD 6.073196e+09     1 1.003831  1.000001  1.110693 0.982413
```

The odds ratio tells us for every one unit increase in the Price variable and Kilometer variable, the odds of the car being all-wheel drive or rear-wheel drive changes by a factor of ~1, holding all other predictors constant, which essentially tells us that Price and Kilometer do not have a massive impact on determining whether a car can be classified as all-wheel drive or rear-wheel drive in reference to front-wheel drive. The Year variable indicates that for every 1 unit increase, the odds of the car being all-wheel drive increases by a factor of 1.01, and the odds of the car being rear-wheel drive decreases by a factor of 0.98, holding other variables constant. The Engine variable reveals that for every one unit increase, the odds of the car being classified as rear-wheel drive or all-wheel drive increases by a factor of 1.004 for both drivetrains compared to forward-wheel drive. Lastly, the Fuel_tank variable reveals that for every unit increase, the odds of the vehicle being classified as all-wheel drive increases by a factor 1.17 and being classified as rear-wheel drive increases by a factor of 1.11, holding other variables constant.

Using the marginaleffects and Desctools packages, we can also find the McFadden R squared value along with the average marginal effects.

```
#Average Marginal Effects
car_marg_mn1 <- slopes(car_mn)
summary(car_marg_mn1)
```

```
##
##  Group      Term    Contrast  Estimate Std. Error       z Pr(>|z|)     2.5 %
##    AWD Engine    mean(dY/dX)  5.06e-05   2.38e-05   2.130  0.03314  4.05e-06
##    AWD Fuel_Tank mean(dY/dX)  8.52e-03   3.28e-04  26.018  < 0.001  7.88e-03
##    AWD Kilometer mean(dY/dX)  5.45e-07   3.96e-07   1.376  0.16894 -2.31e-07
##    AWD Price     mean(dY/dX)  2.63e-08   1.01e-08   2.594  0.00947  6.43e-09
##    AWD Year      mean(dY/dX)  3.27e-03   1.09e-04  30.046  < 0.001  3.05e-03
##    FWD Engine    mean(dY/dX) -2.47e-04   1.39e-05 -17.796  < 0.001 -2.75e-04
##    FWD Fuel_Tank mean(dY/dX) -7.82e-03   4.29e-04 -18.224  < 0.001 -8.66e-03
##    FWD Kilometer mean(dY/dX) -1.13e-07   2.25e-07  -0.501  0.61603 -5.54e-07
##    FWD Price     mean(dY/dX) -1.76e-08   5.77e-09  -3.040  0.00237 -2.89e-08
##    FWD Year      mean(dY/dX)  6.43e-04   5.78e-05  11.120  < 0.001  5.30e-04
##    RWD Engine    mean(dY/dX)  1.97e-04   2.21e-05   8.925  < 0.001  1.54e-04
##    RWD Fuel_Tank mean(dY/dX) -6.98e-04   4.90e-04  -1.425  0.15424 -1.66e-03
##    RWD Kilometer mean(dY/dX) -4.32e-07   4.49e-07  -0.962  0.33609 -1.31e-06
```

```
##    RWD Price     mean(dY/dX) -8.73e-09   1.20e-08  -0.725  0.46845 -3.23e-08
##    RWD Year      mean(dY/dX) -3.91e-03   6.23e-05 -62.728  < 0.001 -4.03e-03
##     97.5 %
##   9.72e-05
##   9.16e-03
##   1.32e-06
##   4.61e-08
##   3.48e-03
##  -2.20e-04
##  -6.98e-03
##   3.28e-07
##  -6.24e-09
##   7.56e-04
##   2.40e-04
##   2.62e-04
##   4.48e-07
##   1.49e-08
##  -3.79e-03
##
## Columns: term, group, contrast, estimate, std.error, statistic, p.value, conf.low, conf.high
## Type:  probs
```

```
#Pseudo R Squared Test
PseudoR2(car_mn)
```

```
##  McFadden
## 0.4489827
```

The average marginal effects reveal similar results to our odds and log-odds explanations. It found the same associations of significance for each variable as stated before; however measures of propensity were different in some variables being positive or negative in determining classification of the drivetrain. An example interpretation of the average marginal effects can be: for every one unit increase in the Year variable, the propensity of a vehicle being classified as all-wheel drive instead of forward-wheel drive or rear-wheel drive increases by 0.003, net the other explanatory variables.

The Pseudo R-Squared test (or McFadden Test) revealed value of 0.449. Considering that strong models are between the values of 0.2 and 0.4, this may indicate that our model may have slightly above mediocre strength.

Model Prediction

```
#Create dataframe with multinomial regression fitted values
car_pred <- as.data.frame(car_mn$fitted.values)

# Finds the max of each row to determine classification
car_pred$predict <- apply(car_mn$fitted.values, 1, which.max)

# Labels number with corresponding drivetrain classification
car_pred$predict <- factor(car_pred$predict,levels = 1:3,
                           labels  = c('FWD', 'AWD', 'RWD'))


# Inputs the actual data to the dataframe so they can be compared
car_pred <- car_pred %>%
  mutate(actual = car_data_new$Drivetrain)
```

```
# Creates confusion matrix
confusionMatrix(car_pred$predict, car_pred$actual)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FWD AWD RWD
##        FWD 302   0  21
##        AWD   0  67  58
##        RWD  27 119 177
##
## Overall Statistics
##
##                Accuracy : 0.7082
##                  95% CI : (0.6747, 0.7401)
##     No Information Rate : 0.4267
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5462
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: FWD Class: AWD Class: RWD
## Sensitivity              0.9179     0.3602     0.6914
## Specificity              0.9525     0.9009     0.7165
## Pos Pred Value           0.9350     0.5360     0.5480
## Neg Pred Value           0.9397     0.8158     0.8237
## Prevalence               0.4267     0.2412     0.3320
## Detection Rate           0.3917     0.0869     0.2296
## Detection Prevalence     0.4189     0.1621     0.4189
## Balanced Accuracy        0.9352     0.6305     0.7040
```

The confusion matrix reveals an overall accuracy of roughly 71% making this not the strongest model. In general, measures of specificity were strong with a value of 0.95 for front-wheel drive, 0.90 for all-wheel drive, and 0.71 for rear-wheel drive. However, measures of sensitivity revealed lower values with 0.92 in front-wheel drive, 0.69 for rear-wheel drive, and a very low value of 0.36 for all-wheel drive. It would appear that predicting the classification between all-wheel drive and rear-wheel drive was much more difficult that classifying front-wheel drive.

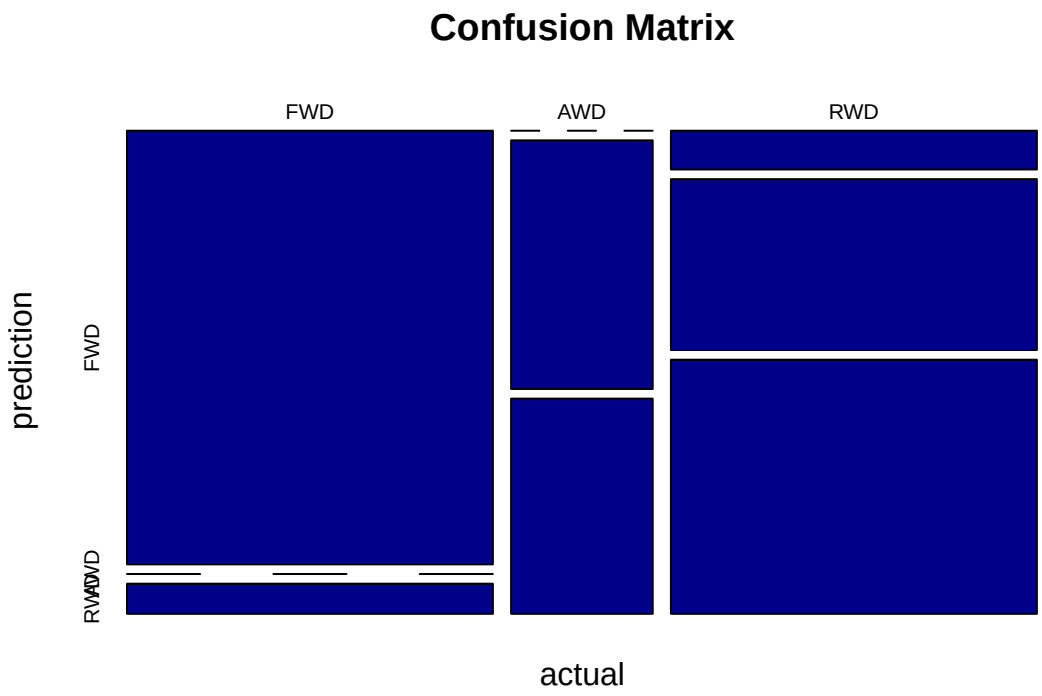
Nonetheless, Mcnemar's Test P-value (less than 0.05) reveals that this is still a statistically significant model despite its mediocre accuracy. Furthermore, the Kappa value of 0.53 indicates that the model has a mediocre level of reliability.

The last thing we will show is the correlated mosaic plot to more easily visualize our results.

```
#Creates mosaic plot
mostab <- table(car_pred$predict, car_pred$actual)
mostab
```

```
##
##       FWD AWD RWD
##   FWD 302   0  21
##   AWD   0  67  58
##   RWD  27 119 177
```

```
mosaicplot(mostab, main = 'Confusion Matrix', xlab = 'actual', ylab = 'prediction',
           color = 'dark blue')
```



Clearly, classifying front-wheel drive was the easiest category to predict while classifying all-wheel and rear-wheel drive were much more difficult. Interestingly, it would appear this model did a very good job in classifying between front-wheel drive and all-wheel drive as the accuracy is extremely high if we only consider those two varaibles in the confusion matrix.

Assumptions of The Model

IIA

The Independence of Irrelevant Alternatives (IIA) assumption tells us that adding or deleting outcome classifications does not affect how the model chooses the original classification. For example, if a vehicle was given the option of having a front-wheel or rear-wheel drivetrain and predicted to have a front-wheel drivetrain, the introduction of the possibility of it having all-wheel drive would not change its predicted classification to rear-wheel drive. Under this assumption, we say that the Independence of Irrelevant Alternatives has been satisfied. While there are no good tests to see if IIA is completely true, checking measures of multicollinearity is a good way to support our belief in this assumption.

Collinearity

To measure collinearity, we need to recreate a normal logistic model such that the drivetrain classifications can be coded in binary. This leads us to having three new subsets of our varaibles. We will, then use the Variance Inflation Factor scores to see if there are any instances of multicollinearity.

```
#Filter data to create subset for logistic regressions
car_aw <- car_data_new %>%
  filter(Drivetrain  == 'AWD' | Drivetrain == 'FWD')

car_rw <- car_data_new %>%
  filter(Drivetrain  == 'RWD' | Drivetrain == 'FWD')
```

```
car_3 <- car_data_new %>%
  filter(Drivetrain  == 'RWD' | Drivetrain == 'AWD')



# Code each Drivetrain numerically
car_aw <- car_aw %>%
  mutate(Drivetrain_num = ifelse(car_aw$Drivetrain == 'AWD', 1, 0))

car_rw <- car_rw %>%
  mutate(Drivetrain_num = ifelse(car_rw$Drivetrain == 'RWD', 1, 0))

car_3 <- car_3 %>%
  mutate(Drivetrain_num = ifelse(car_3$Drivetrain == 'RWD', 1, 0))


# Creating the logistic models
car_awglm1 <- glm(Drivetrain_num ~ Price + Engine + Kilometer + Fuel_Tank + Year,
                  data = car_aw,
            family = binomial(link='logit'))
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
car_rwglm1 <- glm(Drivetrain_num ~ Price + Engine + Kilometer + Fuel_Tank + Year,
                  data = car_rw,
            family = binomial(link='logit'))

car_3glm1 <- glm(Drivetrain_num ~ Price + Engine + Kilometer + Fuel_Tank + Year,
                  data = car_3,
            family = binomial(link='logit'))

# Running the Variance Inflation Factor Function
vif(car_awglm1)
```

```
##     Price    Engine Kilometer Fuel_Tank      Year
##  1.264213  1.039854  2.233386  1.250427  2.402464
```

```
vif(car_rwglm1)
```

```
##     Price    Engine Kilometer Fuel_Tank      Year
##  2.251121  1.496035  1.754406  1.501454  1.820068
```

```
vif(car_3glm1)
```

```
##     Price    Engine Kilometer Fuel_Tank      Year
##  1.728553  1.524758  1.509222  1.871640  1.841352
```

All values in the VIF test are below 10 which indicates that is very unlikely the predictors are collinear. This likely satisfies the assumption of multicollinearity for our original multinomial logistic regression.

Outliers

Since multinomial regression can't use the same outlier detection method as logistic regression like Cook's Distance or DFBETAS, the best way to detect for outliers is to analyze the continuous varaibles. Since I tested for outliers at the beginning of the data analysis using the continuous prediction variables, I was able to subset the data so it didn't include those rows of data. While this seem relatively simple, it might be considered enough to satisfy the outlier assumption of multinomial logistic regression.

Discussion

The goal of this investigation was to determine if there are statistically significant variables that can predict the drivetrain of different vehicles. To begin this exploration, we first set front-wheel drive as our reference variable in the drivetrain class, and compared it to other continuous data. After getting rid of statistical outliers using their z-score, we created a new data frame for our model.

In the logistic regression there were 3 classes: front-wheel drive, rear-wheel drive, and all-wheel drive. I randomly sampled front-wheel drivetrains from the dataset to level the frequencies in the new data set as to not get ansers that are too biased towards front-wheel drivetrains. Upon creating the multinomial logistic regression model, we found that nearly all the continuous variables were statistically significant. However, this did not mean that the variables were impactful. According to the odds ratios and marginal effects (slopes), the Price and Kilometer had minimal impact on the classification of the drivetrain of the vehicle. The most significant difference between the all-wheel drive and rear-wheel drive classification is that older cars wre slightly more likely to have rear wheel-drive compared to a front-wheel drivetrain and new cars were more likely to have all-wheel drivetrain.

I then examined the accuracy of the model as it had an overall 70% level accuracy, with difficulty in measuring Sensitivity (high level of false positives). Nonetheless, Macnemar's test found the model to be statistically significant. Lastly, I examined the assumptions of multinomial regressions and found that the variables were not collinear using the variance inflation factor of the corresponding logistic regressions. Since there are no good measures of the IIA argument and I accounted for outliers at the beginning of the model, those assumptions were also discussed to be upheld.

In future investigations, a data set with more points with similar frequency of drivetrains would be beneficial in understanding if there are truly significant features in predicting drivetrain. While this exploration seems to have statistically significant features, I'm not unsure of whether or not these conclusions are truly generalizable.