Statistical Practice

Taylor Shrode

5/3/2020

MSDS 660

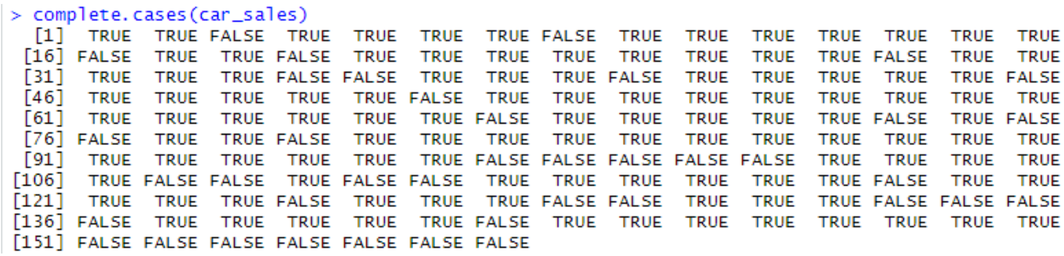
For this assignment, we will be using various statistical methods such as,

1. Simple Linear Regression
2. Multiple Linear Regression
3. One-way ANOVA
4. Two-way ANOVA
5. Logistic Regression

These techniques will allow us to identify relationships between variables and determine whether we can predict, or classify, a specific variable using other variables in the dataset. The dataset we will be using contains car sales data for various makes and models of cars and passenger vehicles (Car sales, n.d.). This dataset contains the following information:

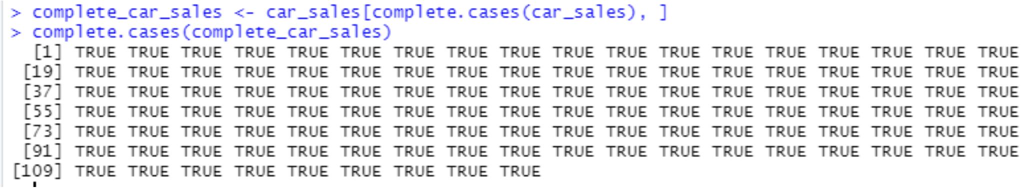
1. Manufacturer: Brand of vehicle.
2. Model: Model of vehicle.
3. Sales_in_thousands: Number of sales for vehicles.
4. _year_resale_value: Resale value of vehicle.
5. Price_in_thousands: Original price of vehicle.
6. Engine_size: Size of engine.
7. Horsepower: Horsepower of vehicle.
8. Wheelbase: Distance between the front and rear axles of a vehicle (inches).
9. Width: Width of vehicle (inches).
10. Length of vehicle (inches).
11. Curb_weight: Weight of vehicle without occupants or baggage.
12. Fuel_capacity: Fuel tank size of vehicle.
13. Fuel_efficiency: How far a vehicle can travel per unit of fuel (mpg).
14. Latest_launch: Date vehicle launched.
15. Power_perf_factor: Method of classifying vehicles based on performance points (Performance factor, n.d.).

Before we begin analyzing our data, we need to upload the dataset into RStudio. In this case, we will use the command**read.csv(file.choose(), header = TRUE)** and store the data into a variable called **car_sales**. Next, we need to check for missing data using the command **complete.cases(<dataset>)** (Vries & Meys, n.d.).



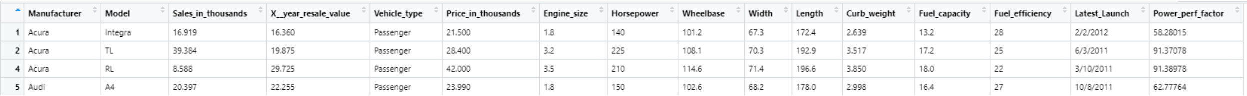
```
> complete.cases(car_sales)
  [1]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [16] FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE
 [31]  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
 [46]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [61]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE
 [76] FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [91]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE
[106]  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE
[121]  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE
[136] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[151] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

The value TRUE are for rows that are complete and FALSE are for the rows that have missing or NA values (Vries & Meys, n.d.). To remove the missing data, we use the command **<dataset>[complete.cases(<dataset>),]** and then we save the cleaned data in a new variable. To ensure we do not have any missing values, we can first use the **complete.cases(<dataset>)** command with our new dataset.



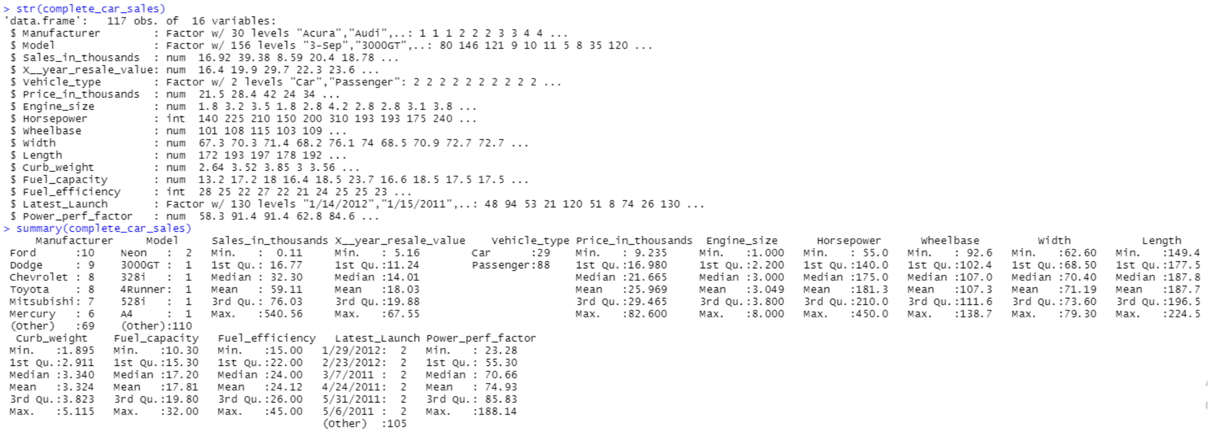
```
> complete_car_sales <- car_sales[complete.cases(car_sales), ]
> complete.cases(complete_car_sales)
  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
 [19] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
 [37] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
 [55] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
 [73] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
 [91] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[109] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

Notice how all the values are TRUE which means there is no missing data. To view our dataset, we can use the **View(<dataset>)** command.



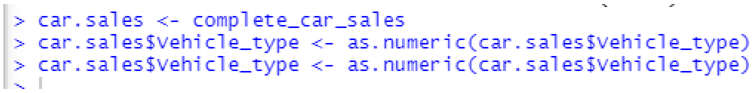
	Manufacturer	Model	Sales_in_thousands	X__year_resale_value	Vehicle_type	Price_in_thousands	Engine_size	Horsepower	Wheelbase	Width	Length	Curb_weight	Fuel_capacity	Fuel_efficiency	Latest_Launch	Power_perf_factor
1	Acura	Integra	16.919	16.360	Passenger	21.500	1.8	140	101.2	67.3	172.4	2.639	13.2	28	2/2/2012	58.28015
2	Acura	TL	39.384	19.875	Passenger	28.400	3.2	225	108.1	70.3	192.9	3.517	17.2	25	6/3/2011	91.37078
4	Acura	RL	8.588	29.725	Passenger	42.000	3.5	210	114.6	71.4	196.6	3.850	18.0	22	3/10/2011	91.38978
5	Audi	A4	20.397	22.255	Passenger	23.990	1.8	150	102.6	68.2	178.0	2.998	16.4	27	10/8/2011	62.77764

Now, we can use the **str(<dataset>)** and **summary(<dataset>)** commands to view the structure and the summary of our data.



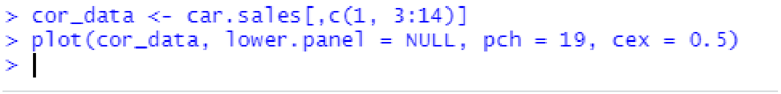
```
> str(complete_car_sales)
'data.frame':   117 obs. of  16 variables:
 $ Manufacturer        : Factor w/ 30 levels "Acura","Audi",..: 1 1 1 2 2 2 3 3 4 4 ...
 $ Model               : Factor w/ 156 levels "3-Sep","3000GT",..: 80 146 121 9 10 11 5 8 35 120 ...
 $ Sales_in_thousands  : num  16.92 39.38 8.59 20.4 18.78 ...
 $ X__year_resale_value: num  16.4 19.9 29.7 22.3 23.6 ...
 $ Vehicle_type        : Factor w/ 2 levels "Car","Passenger": 2 2 2 2 2 2 2 2 2 2 ...
 $ Price_in_thousands  : num  21.5 28.4 42 24 34 ...
 $ Engine_size         : num  1.8 3.2 3.5 1.8 2.8 4.2 2.8 2.8 3.1 3.8 ...
 $ Horsepower          : int  140 225 210 150 200 310 193 193 175 240 ...
 $ Wheelbase           : num  101 108 115 103 109 ...
 $ Width               : num  67.3 70.3 71.4 68.2 76.1 74 68.5 70.9 72.7 72.7 ...
 $ Length              : num  172 193 197 178 192 ...
 $ Curb_weight         : num  2.64 3.52 3.85 3 3.56 ...
 $ Fuel_capacity       : num  13.2 17.2 18 16.4 18.5 23.7 16.6 18.5 17.5 17.5 ...
 $ Fuel_efficiency     : int  28 25 22 27 22 21 24 25 25 23 ...
 $ Latest_Launch       : Factor w/ 130 levels "1/14/2012","1/15/2011",..: 48 94 53 21 120 51 8 74 26 130 ...
 $ Power_perf_factor   : num  58.3 91.4 91.4 62.8 84.6 ...
> summary(complete_car_sales)
     Manufacturer      Model      Sales_in_thousands X__year_resale_value    Vehicle_type Price_in_thousands  Engine_size      Horsepower      Wheelbase         Width           Length     
 Ford       :10    Neon    :  2   Min.   :  0.11     Min.   : 5.16        Car      :29    Min.   : 9.235    Min.   :1.000   Min.   : 55.0   Min.   : 92.6   Min.   :62.60   Min.   :149.4  
 Dodge      : 9    3000GT  :  1   1st Qu.: 16.77     1st Qu.:11.24        Passenger:88    1st Qu.:16.980    1st Qu.:2.200   1st Qu.:140.0   1st Qu.:102.4   1st Qu.:68.50   1st Qu.:177.5  
 Chevrolet : 8    328i    :  1   Median : 32.30     Median :14.01                        Median :21.665    Median :3.000   Median :175.0   Median :107.0   Median :70.40   Median :187.8  
 Toyota     : 8    4Runner:  1   Mean   : 59.11     Mean   :18.03                        Mean   :25.969    Mean   :3.049   Mean   :181.3   Mean   :107.3   Mean   :71.19   Mean   :187.7  
 Mitsubishi: 7    528i    :  1   3rd Qu.: 76.03     3rd Qu.:19.88                        3rd Qu.:29.465    3rd Qu.:3.800   3rd Qu.:210.0   3rd Qu.:111.6   3rd Qu.:73.60   3rd Qu.:196.5  
 Mercury    : 6    A4      :  1   Max.   :540.56     Max.   :67.55                        Max.   :82.600    Max.   :8.000   Max.   :450.0   Max.   :138.7   Max.   :79.30   Max.   :224.5  
 (Other)    :69    (Other):110                                                                                                                                                     
  Curb_weight     Fuel_capacity    Fuel_efficiency    Latest_Launch Power_perf_factor
 Min.   :1.895   Min.   :10.30   Min.   :15.00   1/29/2012:  2   Min.   : 23.28  
 1st Qu.:2.911   1st Qu.:15.30   1st Qu.:22.00   2/23/2012:  2   1st Qu.: 55.30  
 Median :3.340   Median :17.20   Median :24.00   3/7/2011 :  2   Median : 70.66  
 Mean   :3.324   Mean   :17.81   Mean   :24.12   4/24/2011:  2   Mean   : 74.93  
 3rd Qu.:3.823   3rd Qu.:19.80   3rd Qu.:26.00   5/31/2011:  2   3rd Qu.: 85.83  
 Max.   :5.115   Max.   :32.00   Max.   :45.00   5/6/2011 :  2   Max.   :188.14  
                                                 (Other)  :105                   
```

To better view our data, we can create a scatterplot of our data to identify linear relationships between variables. Creating a copy of our cleaned data, we then convert the columns **Manufacturer** and **Vehicle_type** using the **as.numeric()** command.

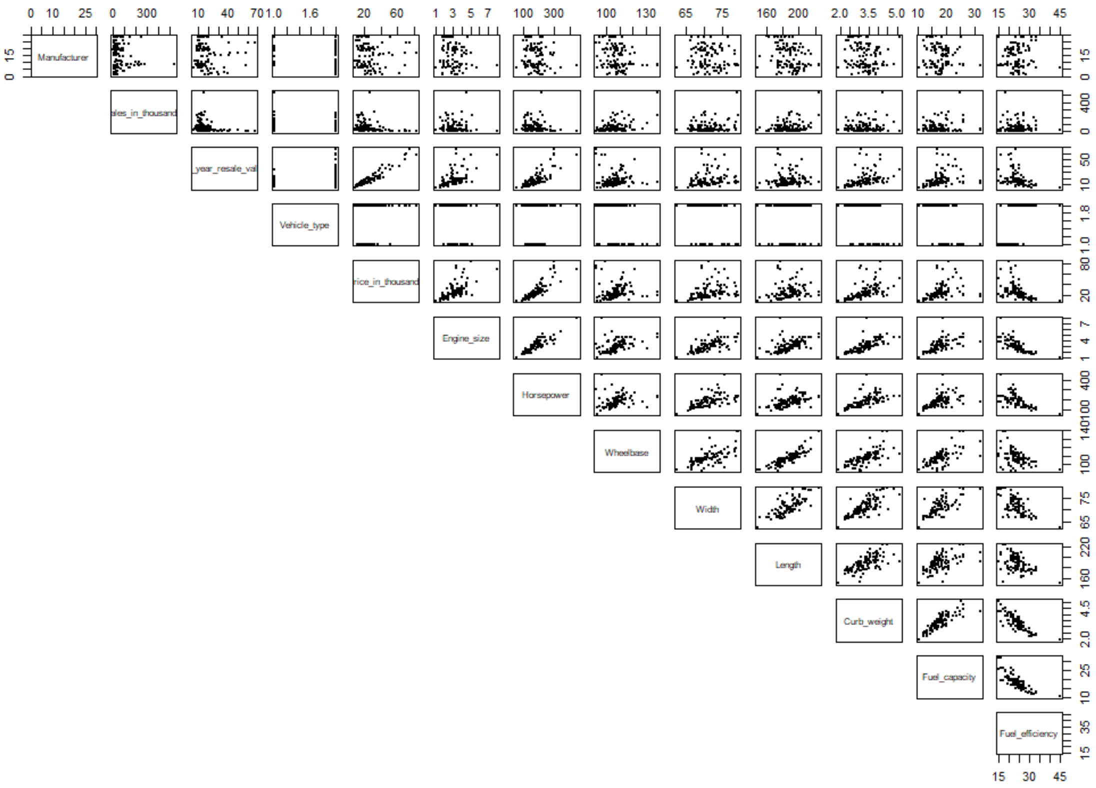


```
> car.sales <- complete_car_sales
> car.sales$Vehicle_type <- as.numeric(car.sales$Vehicle_type)
> car.sales$Vehicle_type <- as.numeric(car.sales$Vehicle_type)
```

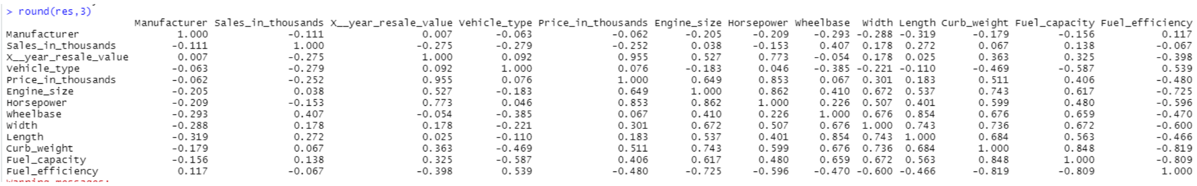
Now, we remove the columns **Model, Latest_Launch,** and **Power_perf_factor** and then plot the data we want included in the scatterplot.



```
> cor_data <- car.sales[,c(1, 3:14)]
> plot(cor_data, lower.panel = NULL, pch = 19, cex = 0.5)
> 
```



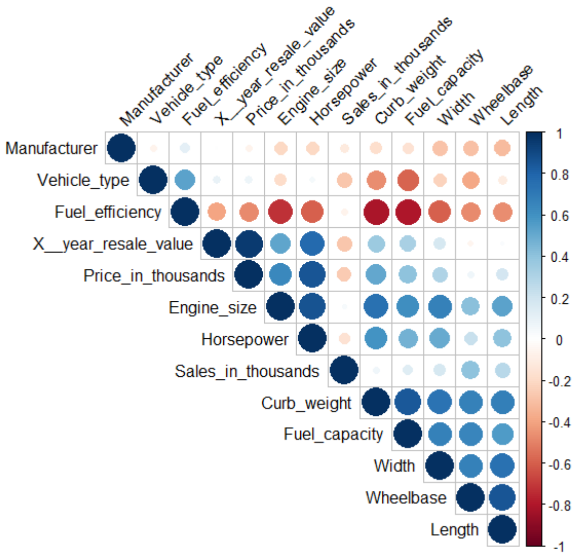
To better identify linear relationships, we can use the **cor()** function which calculates the correlation coefficients between each variable. Rounding the correlation matrix to 3 decimal places,



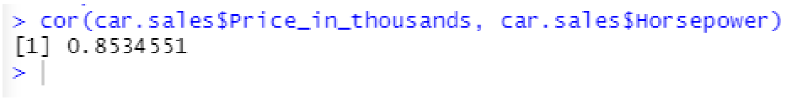
```
> round(res,3)
                     Manufacturer Sales_in_thousands X__year_resale_value Vehicle_type Price_in_thousands Engine_size Horsepower Wheelbase  Width Length Curb_weight Fuel_capacity Fuel_efficiency
Manufacturer                1.000             -0.111                0.007       -0.063             -0.062      -0.205     -0.209    -0.293 -0.288 -0.319      -0.179        -0.156           0.117
Sales_in_thousands         -0.111              1.000               -0.275       -0.279             -0.252       0.038     -0.153     0.407  0.178  0.272       0.067         0.138          -0.067
X__year_resale_value        0.007             -0.275                1.000        0.092              0.955       0.527      0.773    -0.054  0.178  0.025       0.363         0.325          -0.398
Vehicle_type               -0.063             -0.279                0.092        1.000              0.076      -0.183      0.046    -0.385 -0.221 -0.110      -0.469        -0.587           0.539
Price_in_thousands         -0.062             -0.252                0.955        0.076              1.000       0.649      0.853     0.067  0.301  0.183       0.511         0.406          -0.480
Engine_size                -0.205              0.038                0.527       -0.183              0.649       1.000      0.862     0.410  0.672  0.537       0.743         0.617          -0.725
Horsepower                 -0.209             -0.153                0.773        0.046              0.853       0.862      1.000     0.226  0.507  0.401       0.599         0.480          -0.596
Wheelbase                  -0.293              0.407               -0.054       -0.385              0.067       0.410      0.226     1.000  0.676  0.854       0.676         0.659          -0.470
Width                      -0.288              0.178                0.178       -0.221              0.301       0.672      0.507     0.676  1.000  0.743       0.736         0.672          -0.600
Length                     -0.319              0.272                0.025       -0.110              0.183       0.537      0.401     0.854  0.743  1.000       0.684         0.563          -0.466
Curb_weight                -0.179              0.067                0.363       -0.469              0.511       0.743      0.599     0.676  0.736  0.684       1.000         0.848          -0.819
Fuel_capacity              -0.156              0.138                0.325       -0.587              0.406       0.617      0.480     0.659  0.672  0.563       0.848         1.000          -0.809
Fuel_efficiency             0.117             -0.067               -0.398        0.539             -0.480      -0.725     -0.596    -0.470 -0.600 -0.466      -0.819        -0.809           1.000
Warning messages:
```

.

Now, loading the **corrplot** library, we can create a correlogram (Visualize correlation matrix using correlogram, n.d.).



The correlogram highlights the most correlated variables in the specified data. For example, we can easily see that the variables **Price_in_thousands** and **Horsepower** are highly, and positively, correlated. Using the **cor()** function between these variables, we can calculate the correlation coefficient.



```
> cor(car.sales$Price_in_thousands, car.sales$Horsepower)
[1] 0.8534551
> 
```

The ~0.85 indicates that these variables are positively correlated.

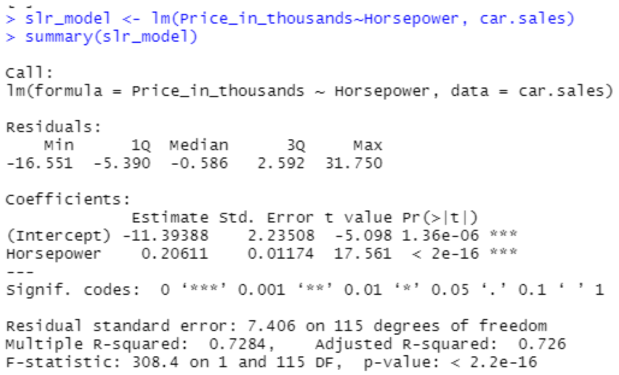
To study the relationship between these two variables, we can use simple linear regression. Recall that the purpose of simple linear regression is to create a model to predict the *y* variable when only the *x* variable is known. The mathematical equation for the model can be generalized as

$$Y = \beta_1 + \beta_2 X$$

where $\beta_1$is the intercept and β_2 is the slope (Prabhakaran, 2016). First, we need to construct our null and alternative hypotheses that will be tested. We are testing, using a 95% confidence level, whether there is a linear relationship between the **Price_in_thousands** and **Horsepower** variables. Thus,

$$H_0 : \beta_1 = 0; \; There\ is\ no\ linear\ relationship\ between\ Price\ and\ Horsepower.$$
$$H_A : \beta_1 \neq 0; \; There\ is\ a\ linear\ relationship\ between\ Price\ and\ Horsepower.$$

Now, we can fit our linear model.

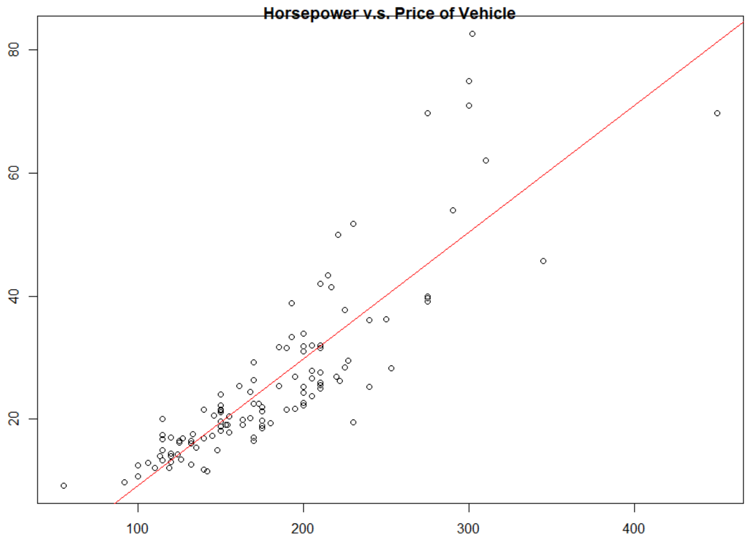


```
> slr_model <- lm(Price_in_thousands~Horsepower, car.sales)
> summary(slr_model)

Call:
lm(formula = Price_in_thousands ~ Horsepower, data = car.sales)

Residuals:
    Min      1Q  Median      3Q     Max 
-16.551  -5.390  -0.586   2.592  31.750 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -11.39388    2.23508  -5.098 1.36e-06 ***
Horsepower    0.20611    0.01174  17.561  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.406 on 115 degrees of freedom
Multiple R-squared:  0.7284,	Adjusted R-squared:  0.726 
F-statistic: 308.4 on 1 and 115 DF,  p-value: < 2.2e-16
```

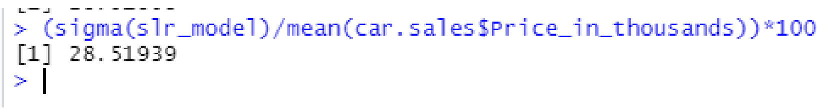
The first thing we can analyze from this model is the regression line equation,

$$Price \ = \ -11.39388 + 0.20611 * Horsepower$$

which tells us that the estimate of the intercept is $-11.39388$and the estimate of the Price coefficient (slope) is 0.20611. Thus, plotting our variables with the best fit regression line (Lillis, 2015),



From the plot, we see that as the horsepower of a vehicle increases, so does the price. From the summary output of our linear model, the p-value for **Horsepower** is less than our level of significance of 0.05. This indicates that the variable **Horsepower** is statistically significant, and we can reject our null hypothesis that there is no linear relationship between price and horsepower. Next, we check the accuracy of our model. The residual standard of error (RSE) measures the data that cannot be explained by the model (Simple linear regression in r, 2018). Above, our RSE is 7.406. To calculate the percentage error,

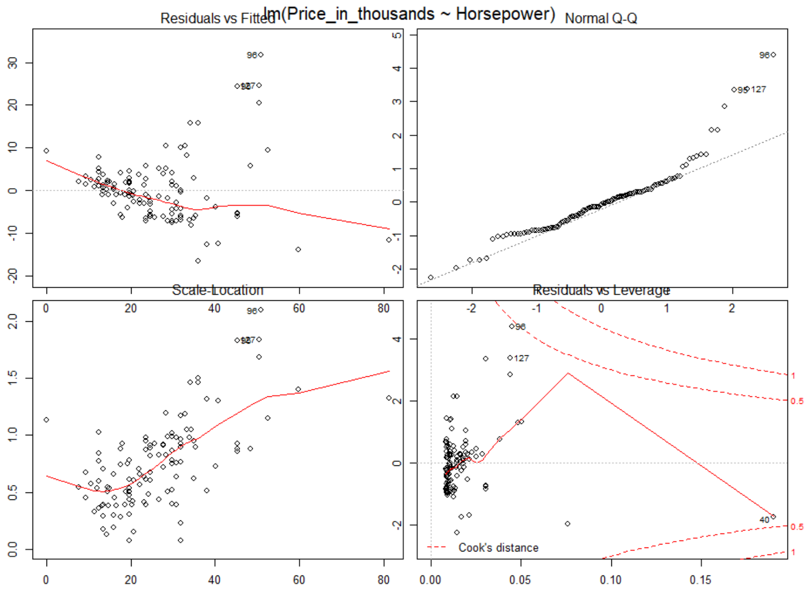


```
> (sigma(slr_model)/mean(car.sales$Price_in_thousands))*100
[1] 28.51939
> |
```

.

The R-squared value represents the proportion of information in the data that can be explained by the model (Simple linear regression in r, 2018). Above, the R-squared value is 0.7264 indicating that a large proportion of the variability in the outcome has been explained by the model.

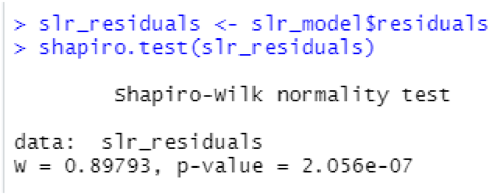
Now, we need to verify the model assumptions with residual analysis.



The assumptions for linear regression are (WorldClass Wk2 FTE):

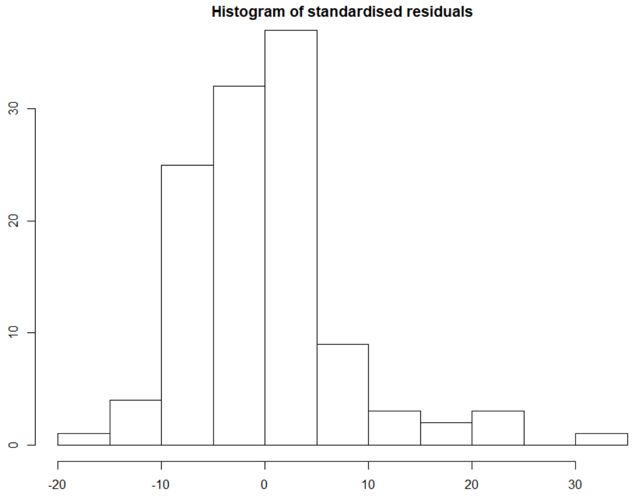
1. Linearity: Relationship between *x* and the mean of *y* is linear.
2. Homoscedasticity: The variance of residual is the same for any value of *x*.
3. Independence: Observations are independent of each other.
4. Normality: For any fixed value of *x*, *y* is normally distributed.

The scale-location plot can be used to identify homoscedasticity (Kassambara, 2018). We do not have a horizontal line which indicates we do not have homoscedasticity. Using the Normal Q-Q plot, we can verify normality (Kassambara, 2018). This plot compares the residuals to the ideal normal observations. From above, we can see that points are distributed along the diagonal line with a few outliers. To further check for normality, we can use the Shapiro-Wilk test or plot a histogram of the model's residuals (World Class Wk2 FTE). Using the Shapiro-Wilk test,



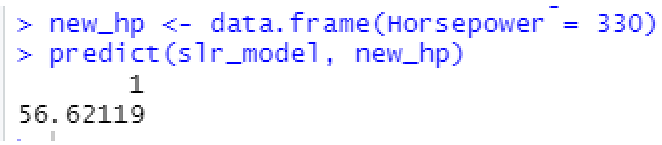
```
> slr_residuals <- slr_model$residuals
> shapiro.test(slr_residuals)

        Shapiro-Wilk normality test

data:  slr_residuals
W = 0.89793, p-value = 2.056e-07
```

The p-value from the Shapiro-Wilk test above is less than our level of significance (0.05), implying that the distribution of the data is significantly different from normal distribution. Plotting the residuals,



We can see that the residuals are skewed, indicating that there is no normality. To check for linearity, we can use the Residuals vs Fitted plot (Kassambara, 2018). The observations do not indicate a distinct pattern which indicates linearity.

Using our model, we can make predictions. The example below predicts the price (in thousands) for a vehicle with 330 horsepower.



```
> new_hp <- data.frame(Horsepower = 330)
> predict(slr_model, new_hp)
       1
56.62119
```

Our prediction using our model suggests that a vehicle with 330 horsepower will cost approximately $56,621.19.

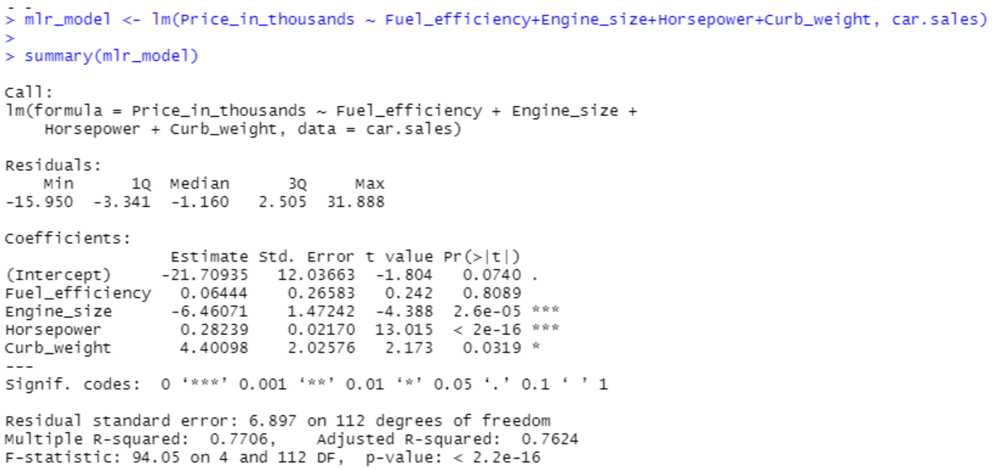
We can now expand on our simple linear regression model by adding parameters to the model. This technique is called multiple linear regression. The generic formula for multiple linear regression is

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

Where Y is the dependent variable, x_k is the explanatory variables, β_0 is the y-intercept, and β_k is the slope coefficients for each explanatory variable (Kenton, 2019). Our first multiple linear regression model will be testing, using a 95% confidence level, whether there is a linear relationship between the **Price_in_thousands** and the variables **Horsepower, Fuel Efficiency, Engine Size,** and **Curb Weight**. Thus, for our hypotheses, if there is no linear relationship, all coefficients will equal 0 (WorldClass Wk3 FTE). In other words,

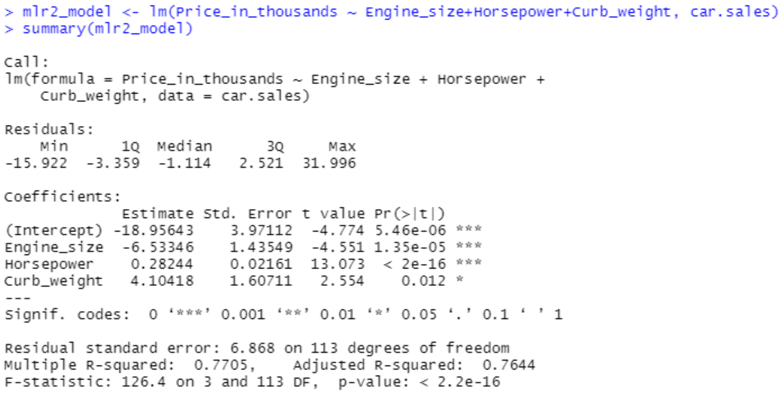
$$H_0: \beta_1 = \beta_2 = \cdots = \beta_n = 0.$$
$$H_A: At\ least\ one\ \beta\ \ is\ not\ zero.$$

Creating our model,



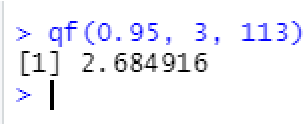
```
> mlr_model <- lm(Price_in_thousands ~ Fuel_efficiency+Engine_size+Horsepower+Curb_weight, car.sales)
>
> summary(mlr_model)

Call:
lm(formula = Price_in_thousands ~ Fuel_efficiency + Engine_size +
    Horsepower + Curb_weight, data = car.sales)

Residuals:
    Min      1Q  Median      3Q     Max
-15.950  -3.341  -1.160   2.505  31.888

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     -21.70935   12.03663  -1.804   0.0740 .
Fuel_efficiency   0.06444    0.26583   0.242   0.8089
Engine_size      -6.46071    1.47242  -4.388  2.6e-05 ***
Horsepower        0.28239    0.02170  13.015  < 2e-16 ***
Curb_weight       4.40098    2.02576   2.173   0.0319 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.897 on 112 degrees of freedom
Multiple R-squared:  0.7706,    Adjusted R-squared:  0.7624
F-statistic: 94.05 on 4 and 112 DF,  p-value: < 2.2e-16
```

We can see that the **Fuel_efficiency** term is not significant. Using backward elimination, we remove this term from the model.



```
> mlr2_model <- lm(Price_in_thousands ~ Engine_size+Horsepower+Curb_weight, car.sales)
> summary(mlr2_model)

Call:
lm(formula = Price_in_thousands ~ Engine_size + Horsepower + 
    Curb_weight, data = car.sales)

Residuals:
    Min      1Q  Median      3Q     Max 
-15.922  -3.359  -1.114   2.521  31.996 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -18.95643    3.97112  -4.774 5.46e-06 ***
Engine_size  -6.53346    1.43549  -4.551 1.35e-05 ***
Horsepower    0.28244    0.02161  13.073  < 2e-16 ***
Curb_weight   4.10418    1.60711   2.554    0.012 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.868 on 113 degrees of freedom
Multiple R-squared:  0.7705,	Adjusted R-squared:  0.7644 
F-statistic: 126.4 on 3 and 113 DF,  p-value: < 2.2e-16
```

Notice, all the terms in the model are significant. In other words, the p-values for all the terms are less than 0.05, which indicates we can reject our null hypothesis. We can also verify this assumption by comparing the F-statistic with the F-critical value (F distribution, n.d.).

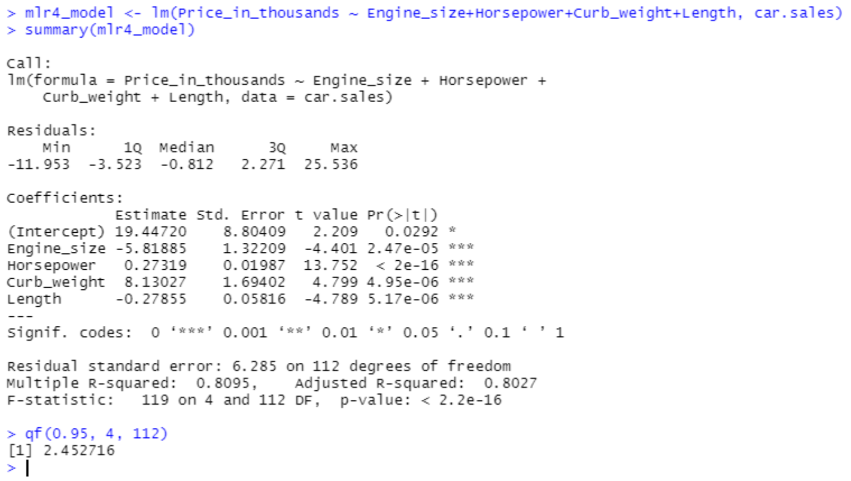


```
> qf(0.95, 3, 113)
[1] 2.684916
> 
```

The F-statistic in our model (126.4) is greater than the F-critical value, which also indicates that we can reject our null hypothesis. Our regression model equation is

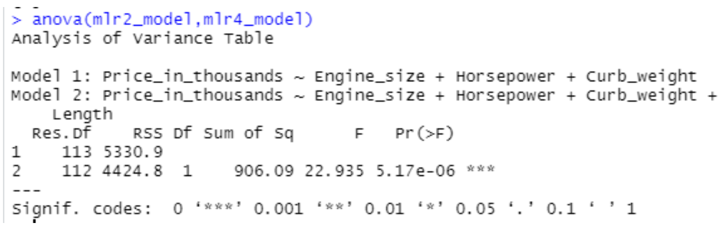
$$Price \ = \ -18.95643 - 6.53346 * Engine\ Size + 0.28244 * Horsepower + 4.10418 * Curb\ Weight.$$

This implies that the price of a vehicle depends on the decrease of engine size, the increase in horsepower, and the increase in curb weight. Now, expanding on this model, we add the variables **Length** and **Width**. Following the same steps as above, we find that **Width** is not significant, and we need to remove it from our model. Using backwards removal, we come to our final model.



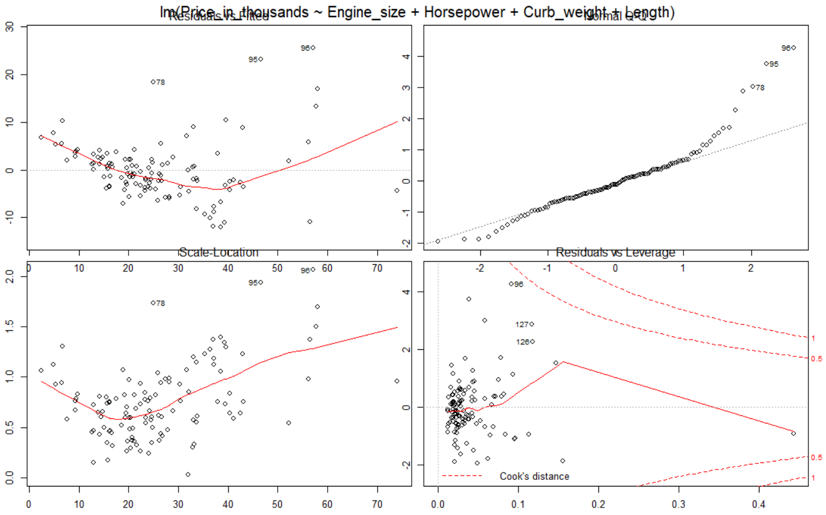
```
> mlr4_model <- lm(Price_in_thousands ~ Engine_size+Horsepower+Curb_weight+Length, car.sales)
> summary(mlr4_model)

Call:
lm(formula = Price_in_thousands ~ Engine_size + Horsepower + 
    Curb_weight + Length, data = car.sales)

Residuals:
    Min      1Q  Median      3Q     Max 
-11.953  -3.523  -0.812   2.271  25.536 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 19.44720    8.80409   2.209   0.0292 *  
Engine_size -5.81885    1.32209  -4.401 2.47e-05 ***
Horsepower   0.27319    0.01987  13.752  < 2e-16 ***
Curb_weight  8.13027    1.69402   4.799 4.95e-06 ***
Length      -0.27855    0.05816  -4.789 5.17e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.285 on 112 degrees of freedom
Multiple R-squared:  0.8095,	Adjusted R-squared:  0.8027 
F-statistic:   119 on 4 and 112 DF,  p-value: < 2.2e-16

> qf(0.95, 4, 112)
[1] 2.452716
> 
```

We see that all the terms in this model are statistically significant and we also see that our F-statistic is greater than our critical F-value, thus we can reject our null hypothesis. To compare our two nested (significant) models through the F-test, we use the **anova()** function (Multiple regression, n.d.).



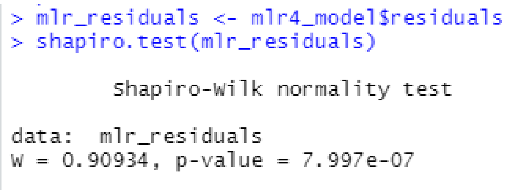
```
> anova(mlr2_model,mlr4_model)
Analysis of Variance Table

Model 1: Price_in_thousands ~ Engine_size + Horsepower + Curb_weight
Model 2: Price_in_thousands ~ Engine_size + Horsepower + Curb_weight + 
    Length
  Res.Df    RSS Df Sum of Sq      F   Pr(>F)    
1    113 5330.9                                 
2    112 4424.8  1    906.09 22.935 5.17e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Notice that the p-value ($5.17e-06$)is less than 0.05, which indicates that the larger model (**mlr4_model**) is significantly better than the smaller model. In other words, additional variables contribute significantly to the price of the vehicle. Now that we have determined the more significant model, we can verify the model assumption. The assumptions for multiple linear regression are the same as simple linear regression. Plotting our model,



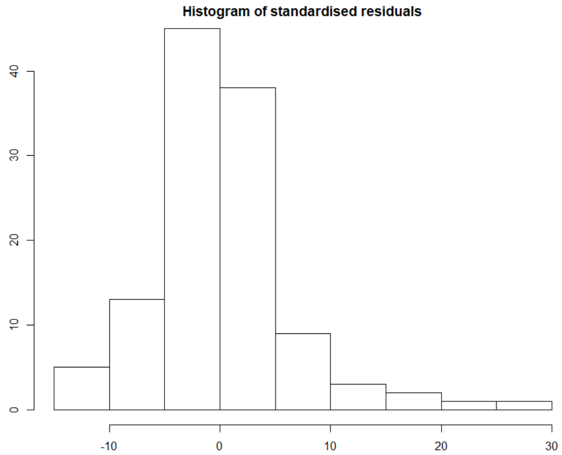
Following the same steps as simple linear regression,

1. Linearity
   a. The Residuals vs Fitted plot indicates that there is no linearity.
2. Normality
   a. Normal Q-Q plot indicates that we cannot assume normality
   b. Shapiro-Wilk test's p-value is less than 0.05 implying that the distribution of the data is significantly different from normal distribution.
      i.



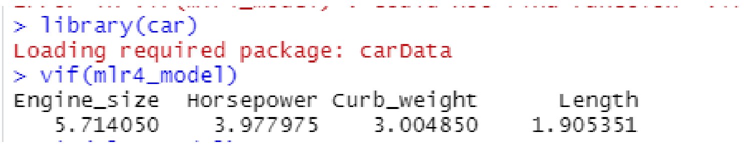
```
> mlr_residuals <- mlr4_model$residuals
> shapiro.test(mlr_residuals)

	Shapiro-Wilk normality test

data:  mlr_residuals
W = 0.90934, p-value = 7.997e-07
```

   c. The histogram of the residuals is skewed and indicates that we cannot assume normality.



i.

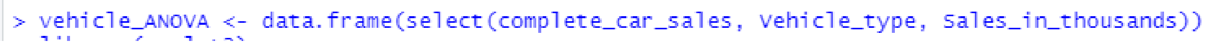
Now, we need to run tests to detect multicollinearity. Multicollinearity occurs when one predictor correlates with another predictor. We can detect multicollinearity by calculating the variance inflation factors.



```
> library(car)
Loading required package: carData
> vif(mlr4_model)
Engine_size  Horsepower Curb_weight      Length
   5.714050    3.977975    3.004850    1.905351
```

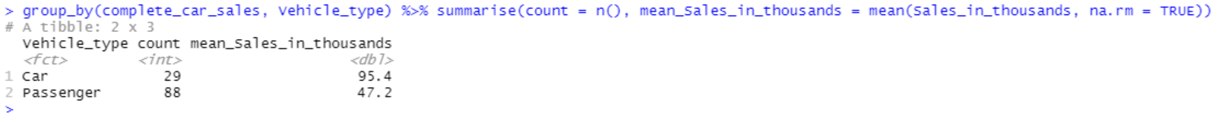
A VIF over 10 indicates a serious problem, a VIF that is greater than or equal to 5 indicates that the coefficient may be misleading due to multicollinearity, and a VIF of 1 indicates the absence of multicollinearity (Multicollinearity essentials, 2018). As we can see above, the variable **Engine_size** may be influenced by multicollinearity.

To continue with our analyses, we will be using One-way ANOVA to compare the mean differences of **Sales_in_thousands** across the levels of **Vehicle_type.** One-way ANOVA is essentially an extension of an independent two-sample t-test (One-way anova test in r, n.d.). We begin by creating a new data frame containing the data for **Sales_in_thousands** and **Vehicle_type**. We can do this by using the **select()** function in the **dplyr** library.



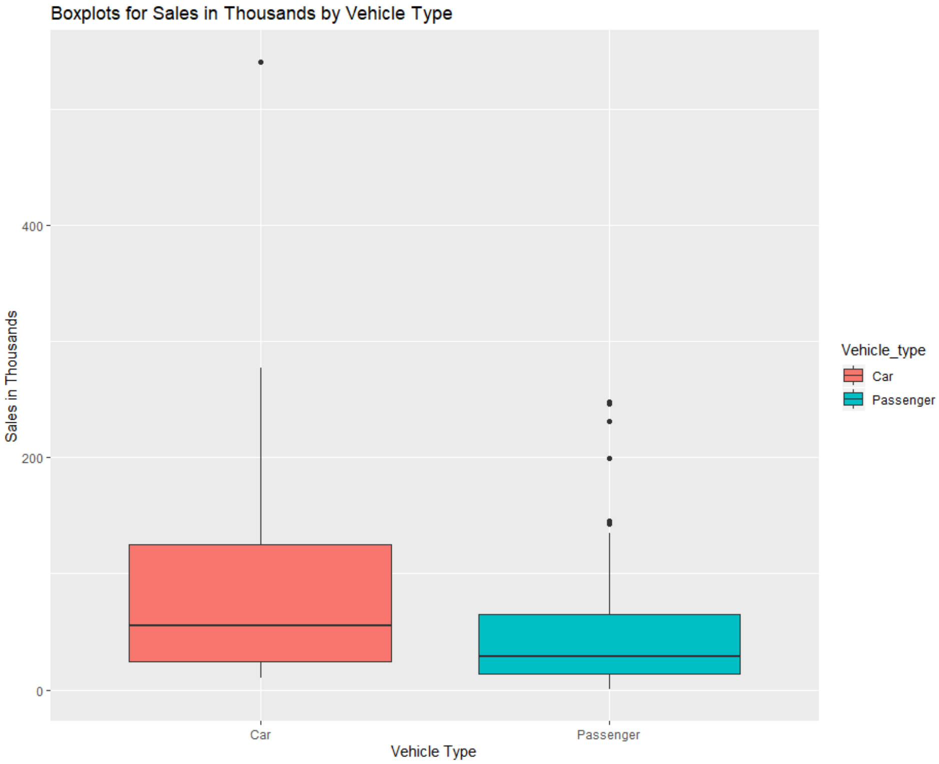
```
> vehicle_ANOVA <- data.frame(select(complete_car_sales, Vehicle_type, Sales_in_thousands))
```

Next, we compute summary statistics by group.

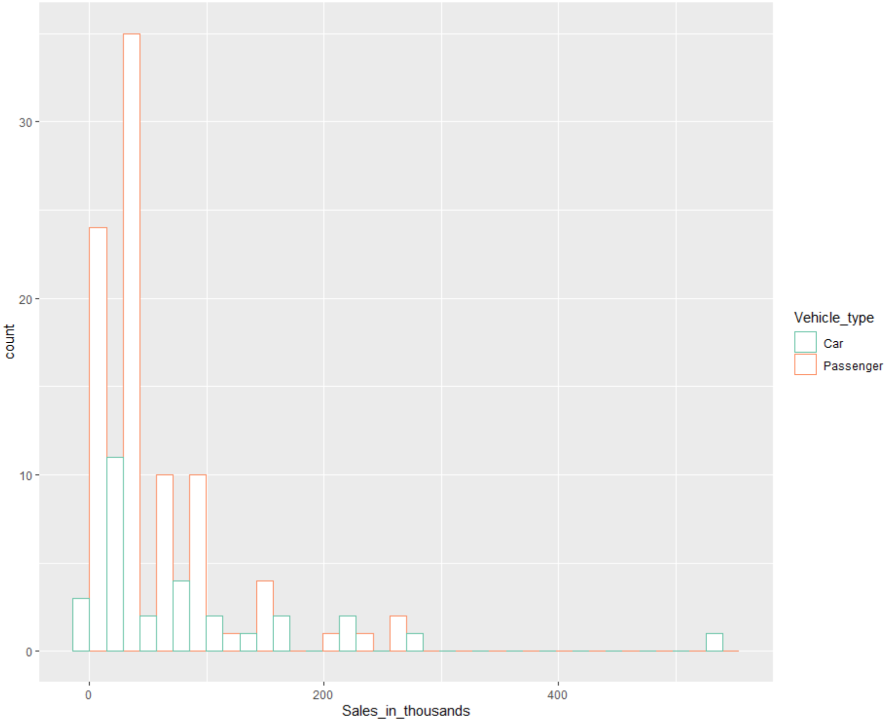


```
> group_by(complete_car_sales, Vehicle_type) %>% summarise(count = n(), mean_Sales_in_thousands = mean(Sales_in_thousands, na.rm = TRUE))
# A tibble: 2 x 3
  Vehicle_type count mean_Sales_in_thousands
  <fct>        <int>                   <dbl>
1 Car             29                    95.4
2 Passenger       88                    47.2
>
```

From this, we gather that there are 29 ‘Car’ vehicles and 88 ‘Passenger’ vehicles. The mean number of sales for ‘Car’ is 95,400 and the mean number of sales for ‘Passenger’ is 47,200. Then, we visualize our data with the use of boxplots and histograms. First, we look at the boxplots for sales by vehicle type.



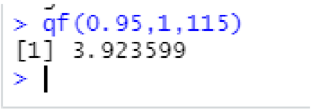
We can see that the range for ‘Car’ is wider than the range for ‘Passenger’. Now, we visualize our data using a histogram.



We can now begin the testing process for one-way ANOVA. We will be analyzing this data using a 5% significance level. First, we need to define our hypotheses. Our null hypothesis states that the means of each group are the same and the alternative hypothesis states that at least sample mean is not equal to the others. We have two groups, thus,

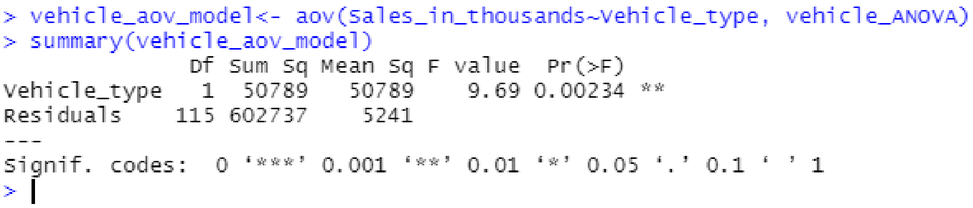
$$H_0: \mu_1 = \mu_2$$
$$H_A: The\ means\ of\ 'Car'\ and\ 'Passenger'\ are\ not\ equal.$$

Next, we need to select our test statistic. In this case, the F-statistic is used for ANOVA. For our decision rule, we will use the p-value for F and the F-value that is produced by our model. We will compare the F-value to the F-critical value.



```
> qf(0.95,1,115)
[1] 3.923599
> 
```

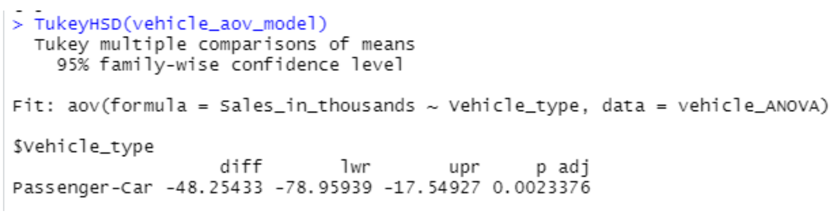
Our critical F-value is *3.923599*. Now, we can compute our test statistic.



```
> vehicle_aov_model<- aov(Sales_in_thousands~Vehicle_type, vehicle_ANOVA)
> summary(vehicle_aov_model)
              Df Sum Sq Mean Sq F value  Pr(>F)   
Vehicle_type   1  50789   50789    9.69 0.00234 **
Residuals    115 602737    5241                   
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> 
```

Our p-value of F is 0.00234, which is less than our level of significance (0.05). This indicates that we can reject the null hypothesis. The F-value (9.69) is also greater than our critical F-value calculated above, which also indicates that we can reject our null hypothesis. Therefore, a one-way analysis of variance shows a significant difference between sales of vehicle types (F(1,115) = 9.69, p = 0.00234).

Now, we perform post hoc tests to confirm where the differences are. One of the most used post hoc tests is Tukey's Honest Significant Difference (HSD), which assesses the significance of differences between pairs of group means (Post hoc tests, 2020).

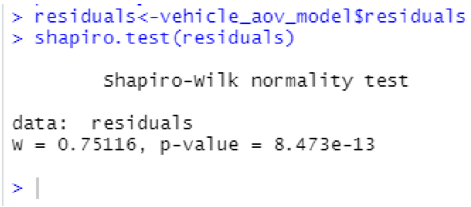


```
> TukeyHSD(vehicle_aov_model)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Sales_in_thousands ~ Vehicle_type, data = vehicle_ANOVA)

$Vehicle_type
                   diff       lwr       upr     p adj
Passenger-Car -48.25433 -78.95939 -17.54927 0.0023376
```

The adjusted p-value for the 'Passenger-Car' pair indicates that the difference between vehicle type is statistically significant. Now, we need to verify the assumptions for one-way ANOVA, which include:

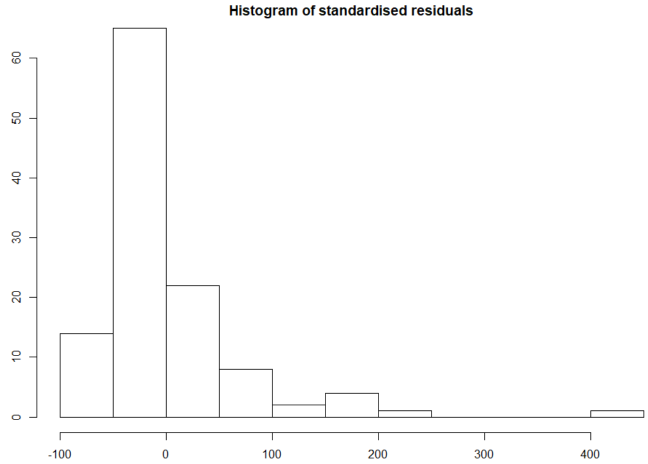
1. Normality.
2. Variance Equality.
3. Sample Independence.

To detect normality, we use the Shapiro-Wilk test and then plot a histogram of the residuals (One-way anova test in r, n.d.).

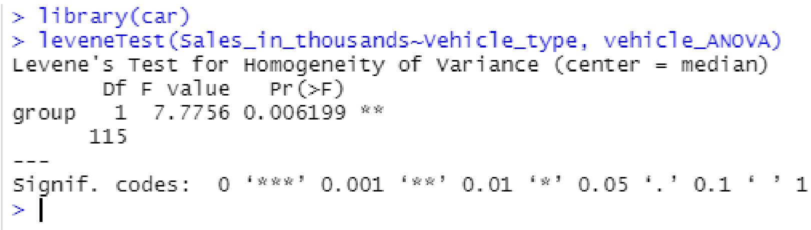


```
> residuals<-vehicle_aov_model$residuals
> shapiro.test(residuals)

        Shapiro-Wilk normality test

data:  residuals
W = 0.75116, p-value = 8.473e-13

> 
```

The output above indicates that normality is violated since the p-value is less than our level of significance.



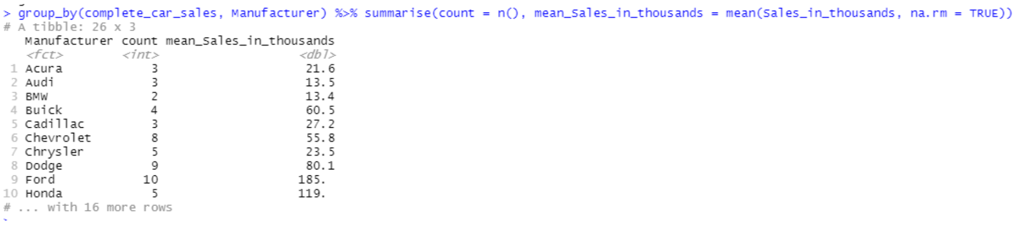
The histogram of the residuals is skewed which also indicates that normality is violated. To check the homogeneity of variance, we use Levene's test.



```
> library(car)
> leveneTest(Sales_in_thousands~Vehicle_type, vehicle_ANOVA)
Levene's Test for Homogeneity of Variance (center = median)
       Df F value   Pr(>F)
group   1  7.7756 0.006199 **
      115
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> 
```

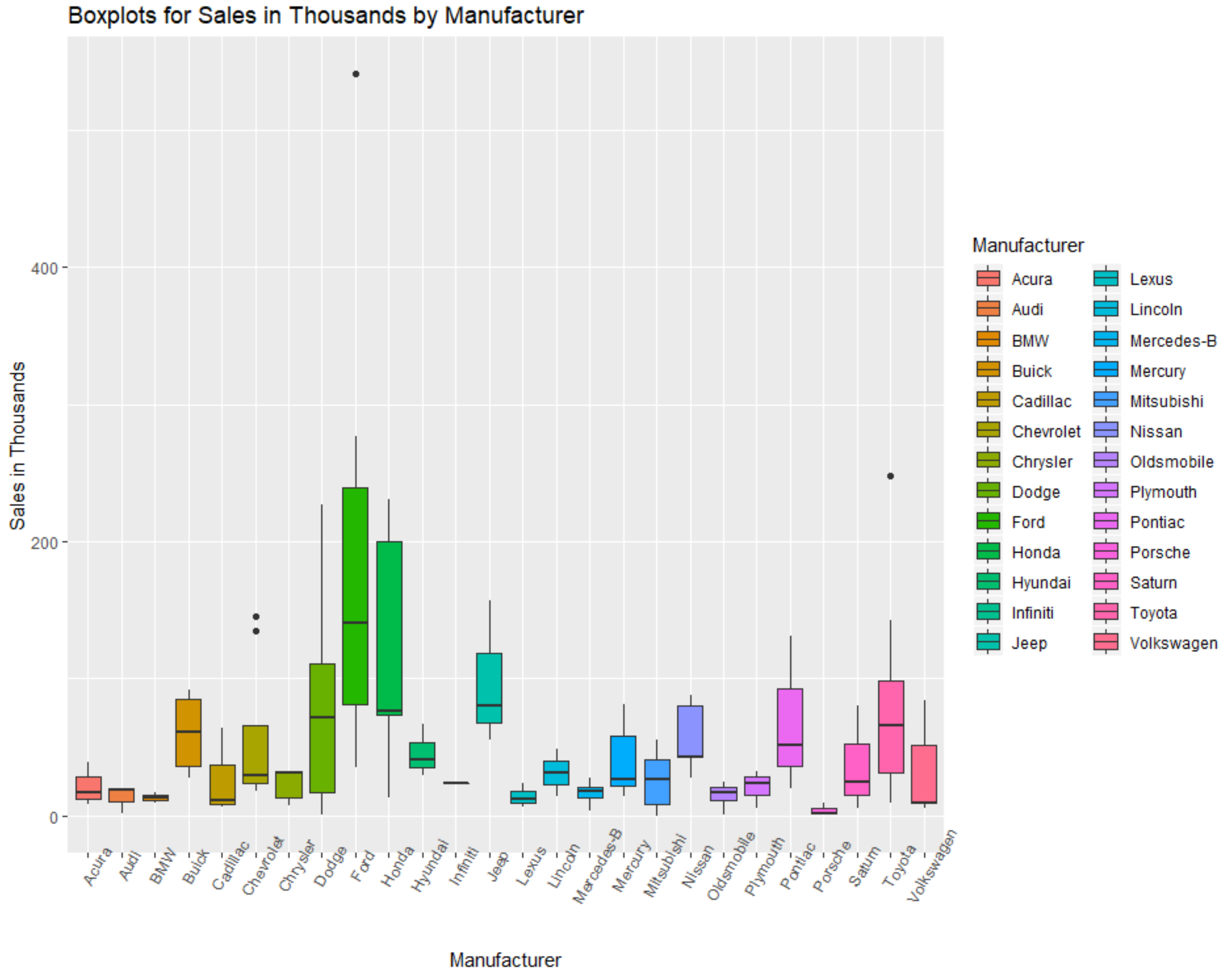
The p-value above is less than our level of significance, which suggests that the variance across groups is statistically significant.

Now that we have determined that the means of sales is statistically different across vehicle types, we will expand and examine whether there is any difference between the average number of sales for different manufacturers and vehicle types. Two-way ANOVA is an extension of one-way ANOVA since we now must account for interaction effects. An interaction is the effect one independent variable has on the other independent variables (Jones, 2020). First, we compute the mean number of sales by manufacturer.

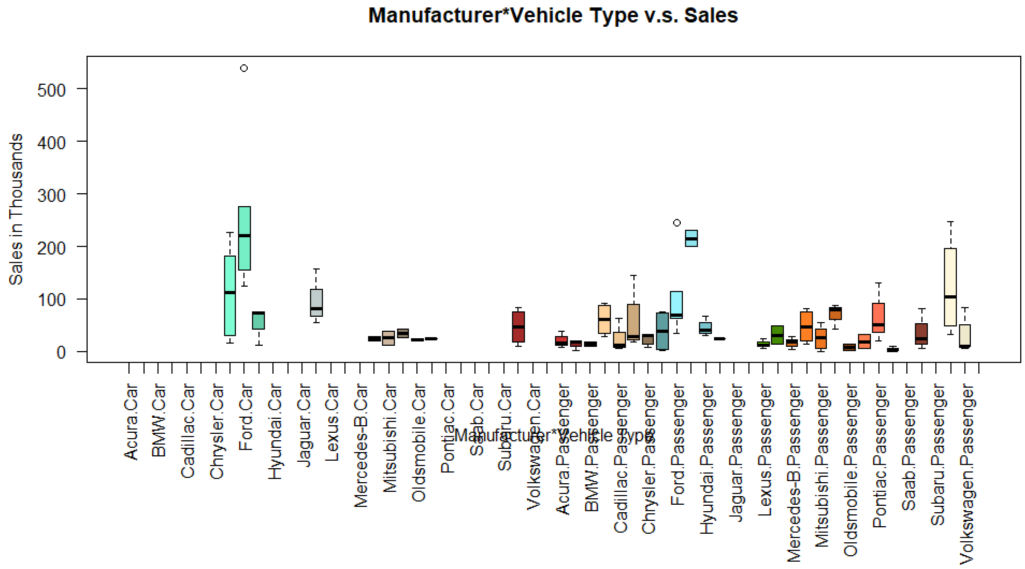


```
> group_by(complete_car_sales, Manufacturer) %>% summarise(count = n(), mean_Sales_in_thousands = mean(Sales_in_thousands, na.rm = TRUE))
# A tibble: 26 x 3
   Manufacturer count mean_Sales_in_thousands
   <fct>        <int>                   <dbl>
 1 Acura            3                    21.6
 2 Audi             3                    13.5
 3 BMW              2                    13.4
 4 Buick            4                    60.5
 5 Cadillac         3                    27.2
 6 Chevrolet        8                    55.8
 7 Chrysler         5                    23.5
 8 Dodge            9                    80.1
 9 Ford            10                   185. 
10 Honda            5                   119. 
# ... with 16 more rows
```

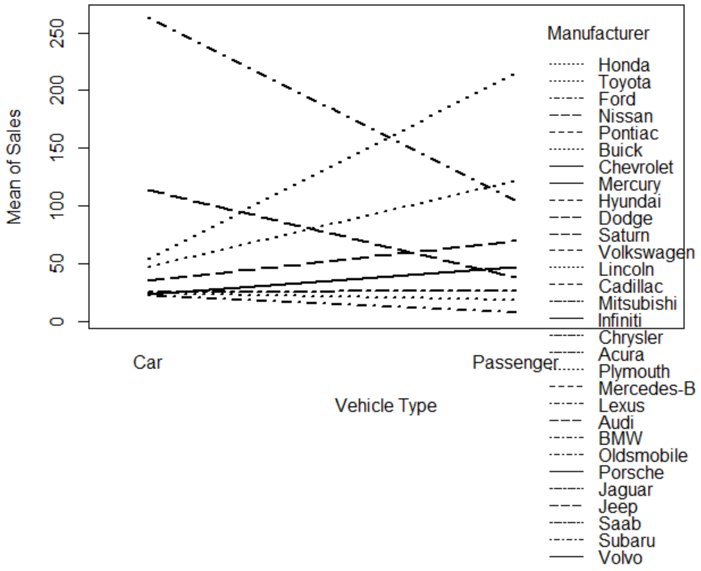
To better visualize our data, we can use boxplots. First, we view the boxplots for sales by manufacturer.



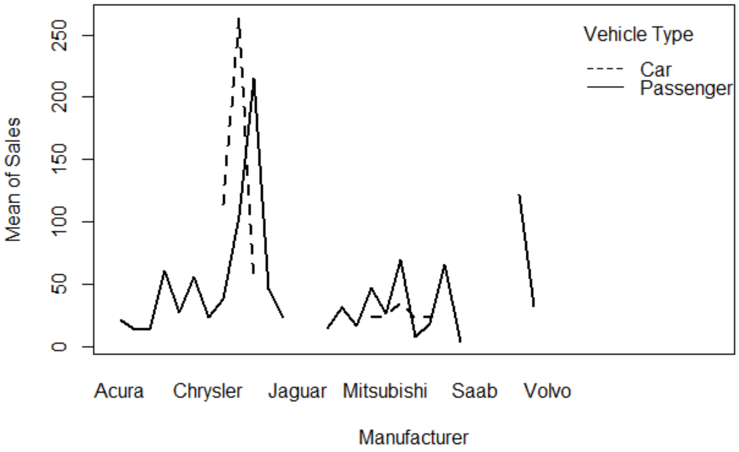
We can see that the widest range of sales are for Dodge, Ford, and Honda, while the smallest range of sales is for Infiniti, BMW, and Porsche. Recall that we viewed the boxplots for sales by vehicle type above. Now, we need to include the boxplot for the interaction between **Vehicle_type** and **Manufacturer**.



Before we determine the hypotheses, we need to identify whether there is an interaction effect or not (WorldClass Wk5 FTE). We can do this by creating interaction plots. First, we look at Sales versus Vehicle Type, with different lines representing different Manufacturers.



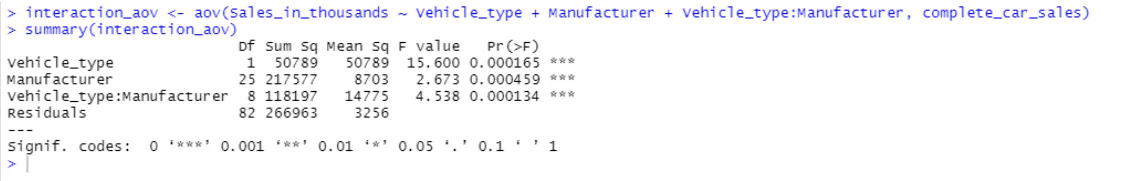
This plot indicates there is an interaction between the levels of Vehicle Type and Manufacturer. Next, we look at Sales versus Manufacturer, with different lines representing Vehicle Type.



This plot indicates that there is an interaction between the levels of Manufacturer and Vehicle Type. We can now continue and determine our hypotheses. There are three sets of hypotheses for two-way ANOVA (Jones, 2020). They include:

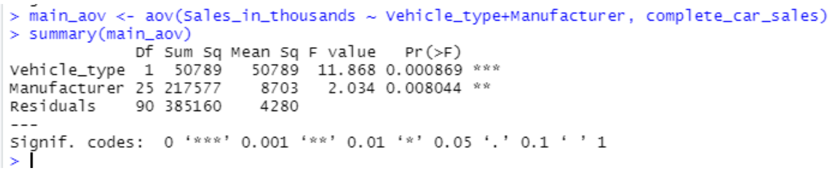
1. Tests for an interaction effect.
   a. H_{01}: *There is no interaction between the two independent variables*
   b. H_{A1}: *There is an interaction between the two independent variables*
2. Tests the main effect for the first independent variable (Manufacturer).
   a. H_{02}: *The means of all the first independent variable groups are equal*
   b. H_{A2}: *The means of at least one of the first independent variable groups are different*
3. Tests the main effect for the second independent variable (Vehicle Type).
   a. H_{03}: *The means of all the second independent variable groups are equal*
   b. H_{A3}: *The means of at least one of the second independent variable groups are differen*

We begin by examining the interaction effect with a level of significance of 0.05.



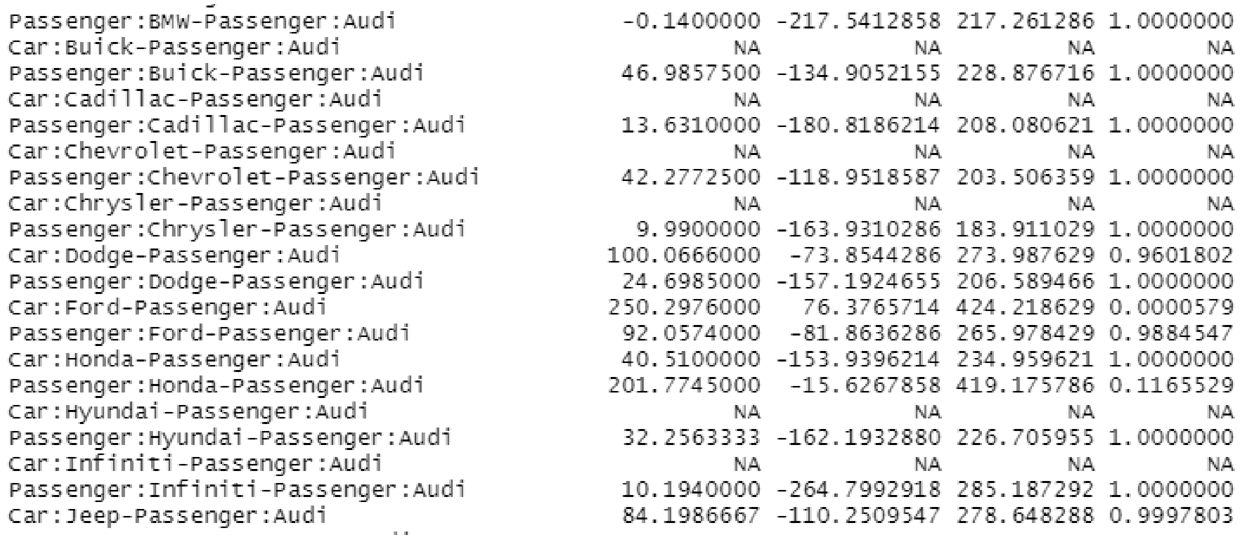
```
> interaction_aov <- aov(Sales_in_thousands ~ Vehicle_type + Manufacturer + Vehicle_type:Manufacturer, complete_car_sales)
> summary(interaction_aov)
                          Df Sum Sq Mean Sq F value   Pr(>F)    
Vehicle_type               1  50789   50789  15.600 0.000165 ***
Manufacturer              25 217577    8703   2.673 0.000459 ***
Vehicle_type:Manufacturer  8 118197   14775   4.538 0.000134 ***
Residuals                 82 266963    3256                     
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> 
```

We can see that our interaction term is significant because the p-value is less than our level of significance, thus rejecting our null (1) hypothesis. Although not necessary, we check the main effects.



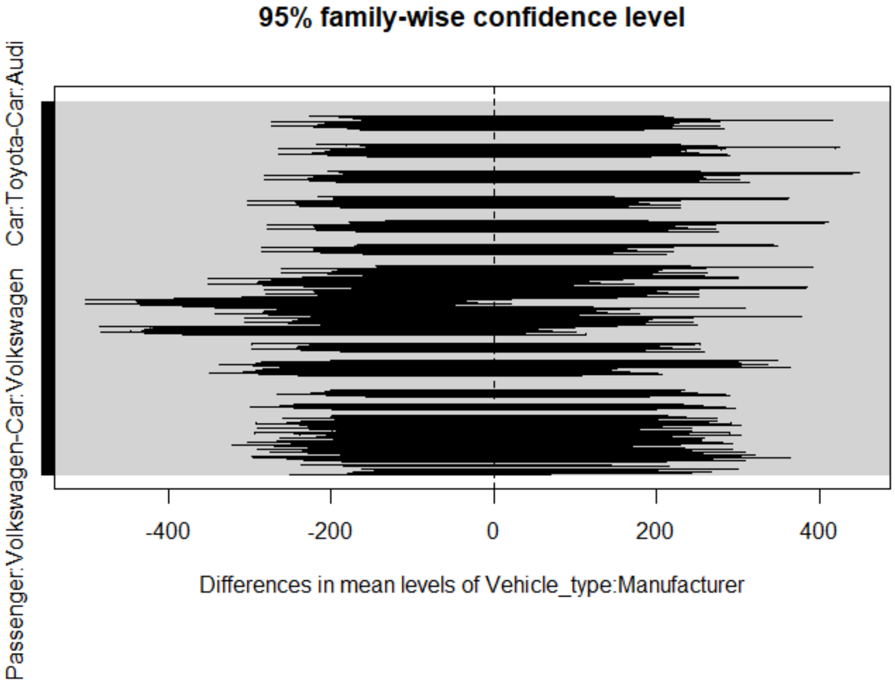
```
> main_aov <- aov(Sales_in_thousands ~ Vehicle_type+Manufacturer, complete_car_sales)
> summary(main_aov)
             Df Sum Sq Mean Sq F value   Pr(>F)
Vehicle_type  1  50789   50789  11.868 0.000869 ***
Manufacturer 25 217577    8703   2.034 0.008044 **
Residuals    90 385160    4280
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> 
```

This output suggests that all the main effects are significant, which also suggests that we can reject the null hypotheses, 2 and 3. Now, we apply a pairwise test on the levels of Manufacturer and Vehicle Type. Below is only a sample of the output from Tukey's Test.



```
Passenger:BMW-Passenger:Audi                    -0.1400000 -217.5412858 217.261286 1.0000000
Car:Buick-Passenger:Audi                                NA           NA         NA        NA
Passenger:Buick-Passenger:Audi                  46.9857500 -134.9052155 228.876716 1.0000000
Car:Cadillac-Passenger:Audi                             NA           NA         NA        NA
Passenger:Cadillac-Passenger:Audi               13.6310000 -180.8186214 208.080621 1.0000000
Car:Chevrolet-Passenger:Audi                            NA           NA         NA        NA
Passenger:Chevrolet-Passenger:Audi              42.2772500 -118.9518587 203.506359 1.0000000
Car:Chrysler-Passenger:Audi                             NA           NA         NA        NA
Passenger:Chrysler-Passenger:Audi                9.9900000 -163.9310286 183.911029 1.0000000
Car:Dodge-Passenger:Audi                       100.0666000  -73.8544286 273.987629 0.9601802
Passenger:Dodge-Passenger:Audi                  24.6985000 -157.1924655 206.589466 1.0000000
Car:Ford-Passenger:Audi                        250.2976000   76.3765714 424.218629 0.0000579
Passenger:Ford-Passenger:Audi                   92.0574000  -81.8636286 265.978429 0.9884547
Car:Honda-Passenger:Audi                        40.5100000 -153.9396214 234.959621 1.0000000
Passenger:Honda-Passenger:Audi                 201.7745000  -15.6267858 419.175786 0.1165529
Car:Hyundai-Passenger:Audi                              NA           NA         NA        NA
Passenger:Hyundai-Passenger:Audi                32.2563333 -162.1932880 226.705955 1.0000000
Car:Infiniti-Passenger:Audi                             NA           NA         NA        NA
Passenger:Infiniti-Passenger:Audi               10.1940000 -264.7992918 285.187292 1.0000000
Car:Jeep-Passenger:Audi                         84.1986667 -110.2509547 278.648288 0.9997803
```

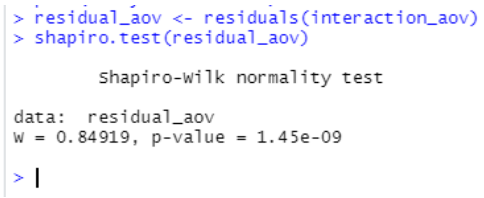
From this output, we can see that there is a difference in Sales mean between Ford Cars and Audi Passengers. Plotting out Tukey test output,



Finally, we need to verify the assumptions for two-way ANOVA (WorldClass Wk5 FTE):

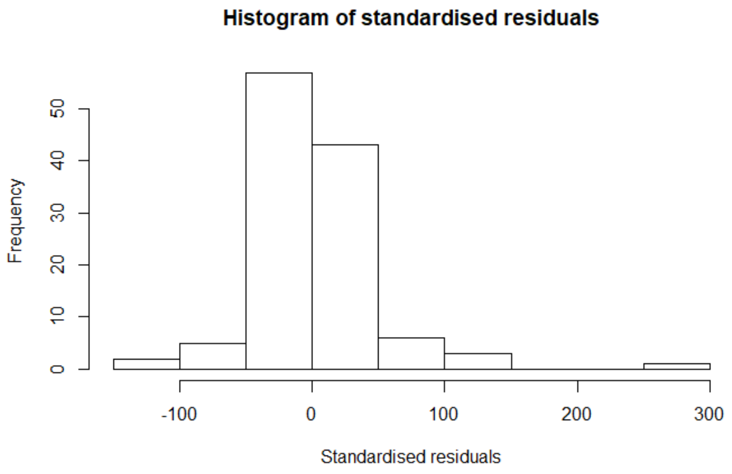
1. Normality
2. Homogeneity
3. Sample Independence

To check for normality, we can use the Shapiro-Wilk test and create a histogram of the residuals.

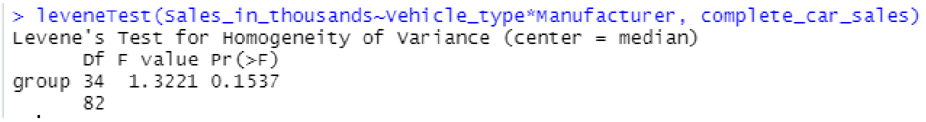


```
> residual_aov <- residuals(interaction_aov)
> shapiro.test(residual_aov)

        Shapiro-Wilk normality test

data:  residual_aov
W = 0.84919, p-value = 1.45e-09

> |
```

The output above indicates that normality is violated since the p-value is less than our level of significance.



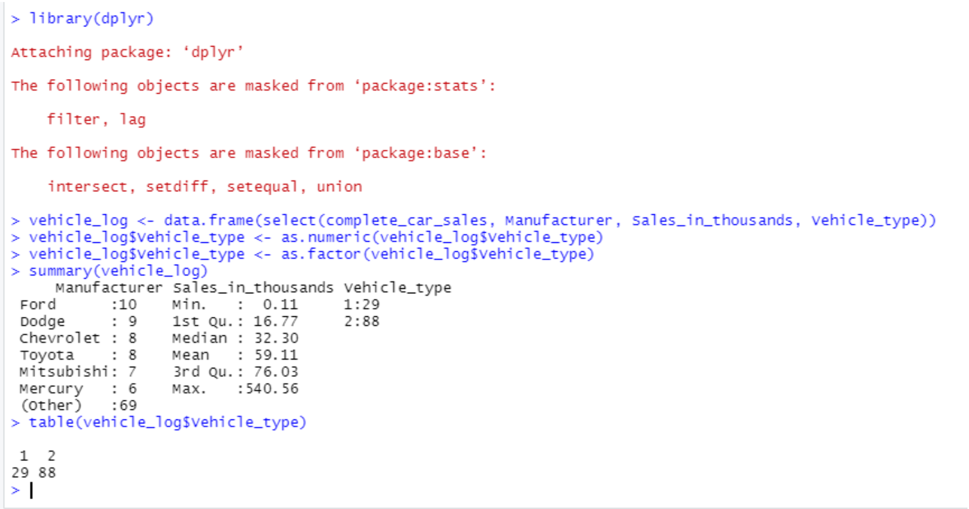
The histogram is skewed which also indicates that we cannot assume normality. To check for homogeneity of variances we use Levene's test.



```
> leveneTest(Sales_in_thousands~Vehicle_type*Manufacturer, complete_car_sales)
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group 34  1.3221 0.1537
      82
```

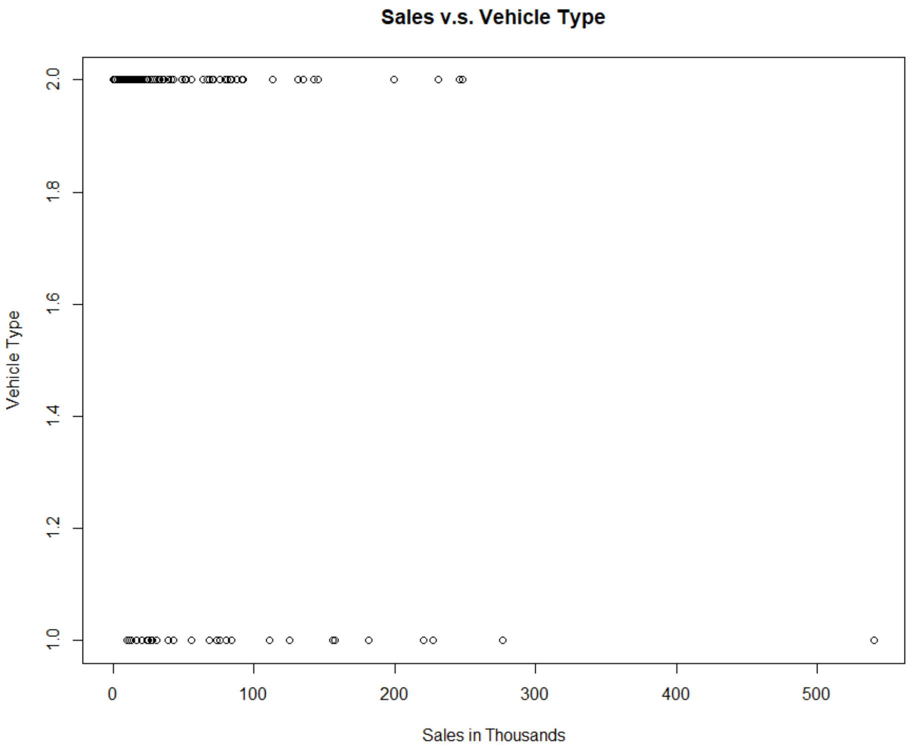
From the output above, we can see that the p-value is not less than 0.05 which suggests that the variance across groups is not statistically different.

Regression analysis solves problems regarding predictions by modeling a dataset with a function that is used to predict future values. Logistic regression is used to solve classification problems (WorldClass Wk6 FTE). For our final analysis method, we will be using logistic regression to examine whether sales has an influence on whether the vehicle will be a 'Passenger' vehicle or a 'Car'. This study will be using binomial logistic regression since our dependent variable, **Vehicle_type**, has two levels ('Car' and 'Passenger'). Our dataset will be a subset of our **complete_car_sales** dataset and contain **Sales_in_thousands** and **Vehicle_type**. We will need to convert the **Vehicle_type** column to a *factor* so that R understands that the data is a categorical variable (Alice, 2015).

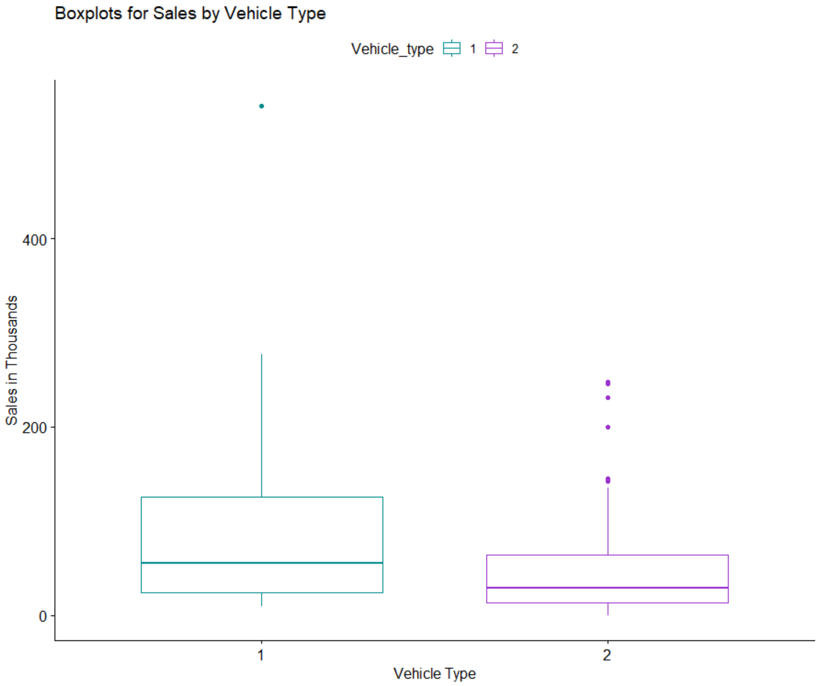


```
> library(dplyr)

Attaching package: ‘dplyr’

The following objects are masked from ‘package:stats’:

    filter, lag

The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union

> vehicle_log <- data.frame(select(complete_car_sales, Manufacturer, Sales_in_thousands, Vehicle_type))
> vehicle_log$Vehicle_type <- as.numeric(vehicle_log$Vehicle_type)
> vehicle_log$Vehicle_type <- as.factor(vehicle_log$Vehicle_type)
> summary(vehicle_log)
     Manufacturer Sales_in_thousands Vehicle_type
 Ford      :10    Min.   :  0.11     1:29
 Dodge     : 9    1st Qu.: 16.77     2:88
 Chevrolet : 8    Median : 32.30
 Toyota    : 8    Mean   : 59.11
 Mitsubishi: 7    3rd Qu.: 76.03
 Mercury   : 6    Max.   :540.56
 (Other)   :69
> table(vehicle_log$Vehicle_type)

 1  2
29 88
> |
```

It should be noted that ‘2’ represents ‘Passengers’ vehicles and ‘1’ represents ‘Car’. For logistic regression, 1 typically represents the presence of a characteristic and a 0 represents the absence of a characteristic (WorldClass Wk6 FTE). Here, a ‘2’ represents the presence of passenger vehicles and a ‘1’ represents the absence of a passenger vehicle. Plotting our sales data against vehicle type,



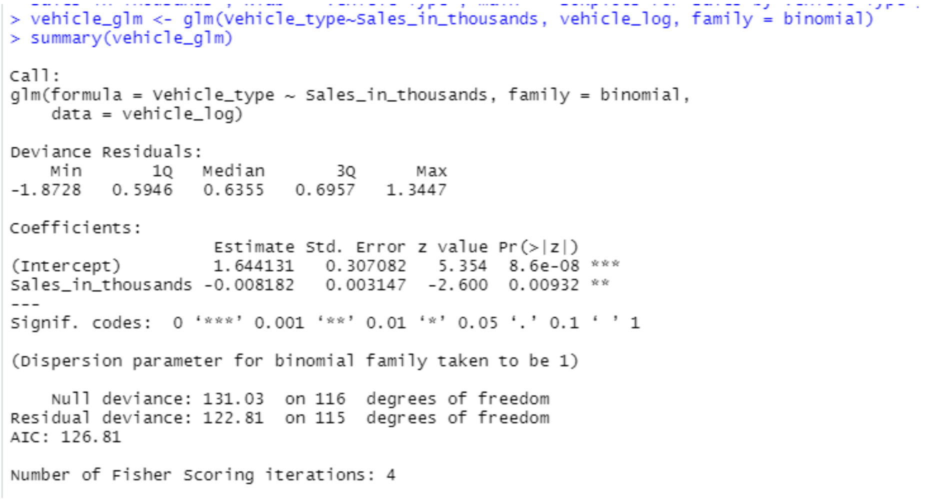
Now, we create a boxplot of sales by vehicle type.



Before we create our logistic regression model, we need to define our null and alternative hypotheses. We are determining whether sales has an effect on vehicle type, thus,

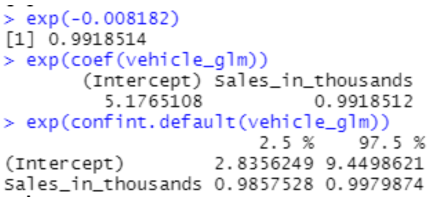
$$H_0: \beta_1 = 0;\ \textit{Sales coefficient is equal to zero}$$
$$H_A: \beta_1 \neq 0;\ \textit{Sales coefficient is not equal to zero.}$$

If the sales coefficient is equal to 0, then this indicates that there is no relationship between sales and vehicle type. We will be testing our hypotheses using a level of significance of 0.05. To create our model in R, the function **glm()** is called and then we can specify the type of logistic regression model that will be fitted (Alice, 2015). We are fitting a binary logistic regression model so we will specify the parameter **family = binomial** in the **glm()** function.



```
> vehicle_glm <- glm(vehicle_type~sales_in_thousands, vehicle_log, family = binomial)
> summary(vehicle_glm)

Call:
glm(formula = vehicle_type ~ sales_in_thousands, family = binomial,
    data = vehicle_log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8728   0.5946   0.6355   0.6957   1.3447

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)         1.644131   0.307082   5.354  8.6e-08 ***
sales_in_thousands -0.008182   0.003147  -2.600  0.00932 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 131.03  on 116  degrees of freedom
Residual deviance: 122.81  on 115  degrees of freedom
AIC: 126.81

Number of Fisher Scoring iterations: 4
```

First, we can see that sales is statistically significant since the p-value is less than our level of significance. Next, we can determine our logit (log-odds) model. A logistic regression model has a logit that is linear in x, which means that “the coefficients can be interpreted as the amount of change in the log-odds for one unit increase in the predictor (WorldClass Wk6 FTE).” Our sales coefficient is-0.008182.This indicates that for one unit change in sales, the log odds of the vehicle being a passenger vehicle decrease by 0.008182. This means that for one unit change in sales, the odds of the vehicle being a car go up by $exp(-0.008182) \ = \ 0.9918154$.

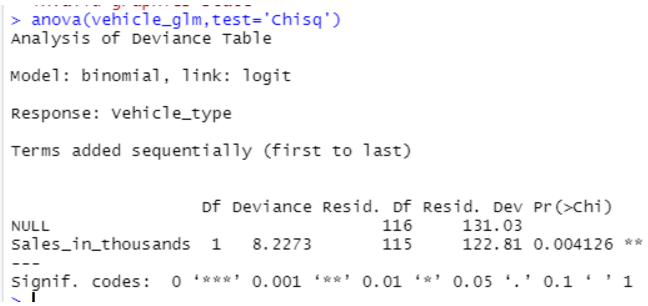


```
> exp(-0.008182)
[1] 0.9918514
> exp(coef(vehicle_glm))
       (Intercept) Sales_in_thousands
         5.1765108          0.9918512
> exp(confint.default(vehicle_glm))
                        2.5 %    97.5 %
(Intercept)        2.8356249 9.4498621
Sales_in_thousands 0.9857528 0.9979874
```

In other words, the odds of the vehicle being a passenger vehicle are about 0.9918512 times for each additional increase in sales. The logit of our model is

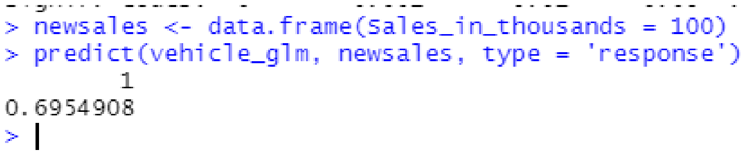
$$logit(p) \ = \ 1.644131 - 0.008181 * Sales.$$

 Now, we can test the overall model fit using the **anova()** function with the parameter **test = “Chisq”** (Alice, 2015). This test is also known as the likelihood ratio test, which compares the full model (with sales) with the null model (model only containing the intercept) (WorldClass Wk6 FTE).



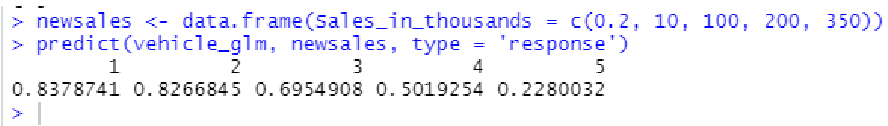
```
> anova(vehicle_glm,test='Chisq')
Analysis of Deviance Table

Model: binomial, link: logit

Response: Vehicle_type

Terms added sequentially (first to last)


                   Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL                                 116     131.03
Sales_in_thousands  1   8.2273       115     122.81 0.004126 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The p-value above is less than our level of significance which indicates that the fitted model is better than the null model. Now, we can use our model to predict probabilities using the **predict()** function. Suppose we want to predict the probability that a vehicle with 100 sales (in thousands) is a passenger vehicle.



```
> newsales <- data.frame(Sales_in_thousands = 100)
> predict(vehicle_glm, newsales, type = 'response')
        1
0.6954908
> |
```

This means that the probability of a vehicle with 10,000 sales being a ‘Passenger’ vehicle is ~69.5%. Expanding this to predict various sales, we can identify a trend.



```
> newsales <- data.frame(sales_in_thousands = c(0.2, 10, 100, 200, 350))
> predict(vehicle_glm, newsales, type = 'response')
        1         2         3         4         5 
0.8378741 0.8266845 0.6954908 0.5019254 0.2280032 
> 
```

The output above suggests that as the number of sales (in thousands) increase, the probability that the vehicle is a ‘Passenger’ decreases.

We began our data analysis by using simple linear regression to determine whether there is a relationship between the price of a vehicle and horsepower. We found that horsepower is statistically significant in predicting the price of a vehicle. Expanding on this conclusion, we performed multiple linear regression and found that engine size, horsepower, curb weight, and length of a vehicle are statistically significant to predicting price. After this, we used analysis of variance techniques to determine the difference in means for sales between the different levels of vehicle type and manufacturer. Our ANOVA tests determined that there are statistical differences between sales of ‘Passenger’ vehicles and ‘Cars’. We also found that there are statistical differences between sales for different manufacturers and vehicle types. After our analysis of variance tests, we perform post hoc tests to determine where the differences in sales occurred. For example, we found that there is a statistical difference in sales between Ford Cars and Audi Passengers. We ended our analyses with logistic regression. Here, we created a model to determine the probability that a vehicle would be classified as a ‘Passenger’ or a ‘Car’ using sales. Then, we used the **anova()** function to determine if the best model is the fitted model or the null model. We found that the fitted model is the better model.

Resources

Alice, M. (2015, September 13). How to perform a logistic regression in r. *R-Bloggers*. https://www.r-bloggers.com/how-to-perform-a-logistic-regression-in-r/

Car sales. (n.d.). Retrieved May 2, 2020, from https://kaggle.com/gagandeep16/car-sales

F distribution. (n.d.). Retrieved May 2, 2020, from http://www.r-tutor.com/elementary-statistics/probability-distributions/f-distribution

Jones, J. (2020, January 11). *Two-way anova*. https://people.richland.edu/james/lecture/m170/ch13-2wy.html

Kassambara. (2018, November 3). *Linear regression assumptions and diagnostics in r*. http://www.sthda.com/english/articles/39-regression-model-diagnostics/161-linear-regression-assumptions-and-diagnostics-in-r-essentials/

Kenton, W. (2019). *How multiple linear regression works*. Investopedia. https://www.investopedia.com/terms/m/mlr.asp

Lillis, D. (2015, April 10). *Linear models in r: Plotting regression lines*. The Analysis Factor. https://www.theanalysisfactor.com/linear-models-r-plotting-regression-lines/

Multicollinearity essentials. (2018, November 3). http://www.sthda.com/english/articles/39-regression-model-diagnostics/160-multicollinearity-essentials-and-vif-in-r/

Multiple regression. (n.d.). Retrieved May 2, 2020, from https://www.statmethods.net/stats/regression.html

One-way anova test in r. (n.d.). Retrieved May 3, 2020, from http://www.sthda.com/english/wiki/one-way-anova-test-in-r

Performance factor. (n.d.). https://www.fia.com/performance-factor

Post hoc tests. (2020). https://biostats.w.uib.no/post-hoc-tests-tukey-hsd/

Prabhakaran, S. (2016). *Linear regression with r*. http://r-statistics.co/Linear-Regression.html

Simple linear regression in r. (2018, October 3). http://www.sthda.com/english/articles/40-regression-analysis/167-simple-linear-regression-in-r/

Visualize correlation matrix using correlogram. (n.d.). Retrieved May 2, 2020, from http://www.sthda.com/english/wiki/visualize-correlation-matrix-using-correlogram

Vries, A. de, & Meys, J. (n.d.). How to remove rows with missing data in r. *Dummies*. Retrieved May 2, 2020, from https://www.dummies.com/programming/r/how-to-remove-rows-with-missing-data-in-r/