PES University, Bangalore

Established under Karnataka Act No. 16 of 2013

UE21CS342AA2 - Data Analytics - Worksheet 2a - Linear and Logistic Regression

Designed by Aaditya S Goel, Dept. of CSE - aadityasgoel@gmail.com

Welcome to DATA Motors

India is poised to become the third largest economy in the world. To fuel and sustain this growth, Indian businesses are looking to increase their footprint and expand into different markets across the world.

DATA Motors is the leading automotive manufacturer in the country and they're now looking to enter the second largest auto market in the world, the United States of America.

But there's a catch! Pricing of cars in USA seems to be very different to that in India. Now DATA motors wants to enter this market with a bang and they need to get their pricing spot-on. So they have hired you, a consultant at the prestigious Bangalore Consulting Group. Now the onus is on you to understand what factors drive the pricing models of the most successful car companies currently in the market. Let's get to work!

Regression

Regression is a statistical method used to model the connection between variables, understanding how changes in one influence another. It's vital for predicting outcomes, finding patterns, and making informed decisions.

Regression is essential across diverse fields like economics and medicine due to its ability to quantify relationships and make predictions for new data. Its popularity arises from its simplicity, adaptability, and its central role in data-driven decision-making.

In this worksheet we will be exploring 3 concepts. Namely:

- Simple Linear Regression
- Multiple Linear Regression
- Logistic Regression

Before we go any further, let's have a look at the dataset and it's different columns

Data Dictionary

```
price: price of the car in dollars
fuel_type: gas or diesel
CompanyName: name of the manufacturer
aspiration: std (standard or naturally aspirated engine) or turbo (turbocharged engine)
doornumber: number of doors in the car
carbody: type of car (sedan, wagon, hatchback, convertible, hardtop)
drivewheel: rwd (Rear-wheel drive) or fwd(front-wheel drive)
enginelocation: front or rear
wheelbase: distance between front and rear axles in inches
carlength: length of car in inches
carwidth: width of car in inches
```

```
carheight: height of car in inches
curbweight: weight of car with a full tank and standard equipment
cylindernumber: number of cylinders in the engine
horsepower: power generated by the engine in horsepower (hp)
mpg: fuel economy of car in miles per gallon
```

Data Visualising

Let's visualize this all in the form of a Data Frame

```r
cars <- read.csv('Dataset_2a.csv')
head(cars)
```

```
##   price car_ID fueltype CompanyName aspiration doornumber     carbody
## 1 13495      1      gas alfa-romero        std        two convertible
## 2 16500      2      gas alfa-romero        std        two convertible
## 3 16500      3      gas alfa-romero        std        two   hatchback
## 4 13950      4      gas        audi        std       four       sedan
## 5 17450      5      gas        audi        std       four       sedan
## 6 15250      6      gas        audi        std        two       sedan
##   drivewheel enginelocation wheelbase carlength carwidth carheight curbweight
## 1        fwd          front      88.6     168.8     64.1      48.8       2548
## 2        rwd          front      88.6     168.8     64.1      48.8       2548
## 3        rwd          front      94.5     171.2     65.5      52.4       2823
## 4        fwd          front      99.8     176.6     66.2      54.3       2337
## 5        fwd          front      99.4     176.6     66.4      54.3       2824
## 6        fwd          front      99.8     177.3     66.3      53.1       2507
##   cylindernumber horsepower mpg
## 1           four        111  27
## 2           four        111  27
## 3            six        154  26
## 4           four        102  30
## 5           five        115  22
## 6           five        110  25
```

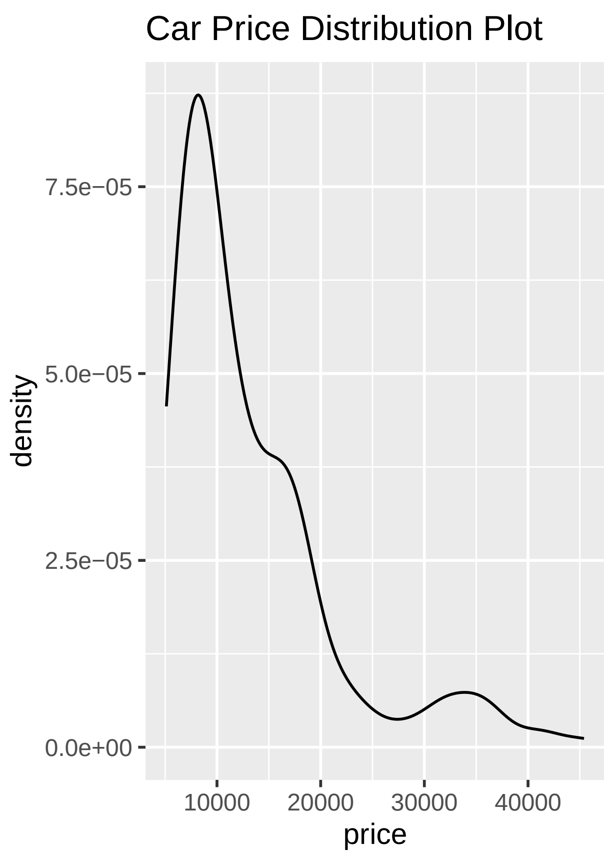
```r
summary(cars)
```

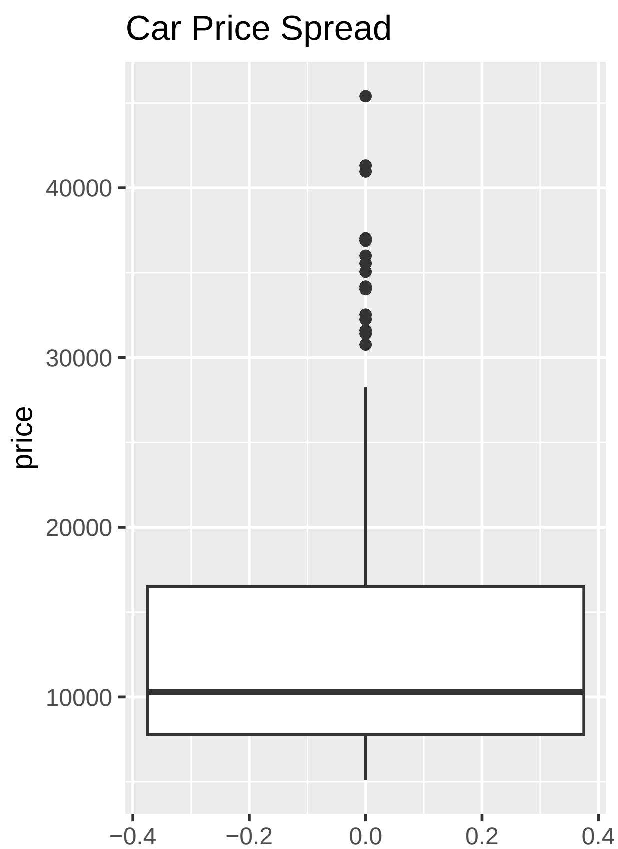
```
##      price           car_ID      fueltype         CompanyName       
##  Min.   : 5118   Min.   :  1   Length:205         Length:205        
##  1st Qu.: 7788   1st Qu.: 52   Class :character   Class :character  
##  Median :10295   Median :103   Mode  :character   Mode  :character  
##  Mean   :13277   Mean   :103                                        
##  3rd Qu.:16503   3rd Qu.:154                                        
##  Max.   :45400   Max.   :205                                        
##   aspiration         doornumber          carbody           drivewheel       
##  Length:205         Length:205         Length:205         Length:205        
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##  enginelocation       wheelbase        carlength        carwidth    
##  Length:205         Min.   : 86.60   Min.   :141.1   Min.   :60.30  
##  Class :character   1st Qu.: 94.50   1st Qu.:166.3   1st Qu.:64.10  
##  Mode  :character   Median : 97.00   Median :173.2   Median :65.50  
##                     Mean   : 98.76   Mean   :174.0   Mean   :65.91  
```

```
##                      3rd Qu.:102.40   3rd Qu.:183.1   3rd Qu.:66.90
##                      Max.   :120.90   Max.   :208.1   Max.   :72.30
##    carheight       curbweight   cylindernumber       horsepower
##  Min.   :47.80   Min.   :1488   Length:205         Min.   : 48.0
##  1st Qu.:52.00   1st Qu.:2145   Class :character   1st Qu.: 70.0
##  Median :54.10   Median :2414   Mode  :character   Median : 95.0
##  Mean   :53.72   Mean   :2556                      Mean   :104.1
##  3rd Qu.:55.50   3rd Qu.:2935                      3rd Qu.:116.0
##  Max.   :59.80   Max.   :4066                      Max.   :288.0
##       mpg
##  Min.   :16.00
##  1st Qu.:25.00
##  Median :30.00
##  Mean   :30.75
##  3rd Qu.:34.00
##  Max.   :54.00
```

Let us plot the distribution of car prices and see what the spread looks like.

```r
library(ggplot2)
library(gridExtra)

plot1 <- ggplot(cars, aes(x = price)) +
  geom_density() +
  labs(title = "Car Price Distribution Plot")

# Create the second plot (Car Price Spread)
plot2 <- ggplot(cars, aes(y = price)) +
  geom_boxplot() +
  labs(title = "Car Price Spread")

# Combine the plots and display
grid.arrange(plot1, plot2, ncol = 2)
```





We can clearly see that the prices are heavily right-skewed with some outliers. This seems to explain the exclusive, luxurious vehicles only affordable for a few.

Regression Analysis

Before proceeding to a full analysis, your client DATA Motors have some questions they want you to answer.

1. Simple Linear Regression

From experience, they have understood that the more powerful their car is, the higher they are able to price it at to the public. They want to know if this trend holds perfectly in this new market too. Have a look at the data, pick the right variables and find the if this relationship is true. Create a scatter plot between the dependent and independent variable with the best-fit line passing through. (Hint: use the ggplot library)

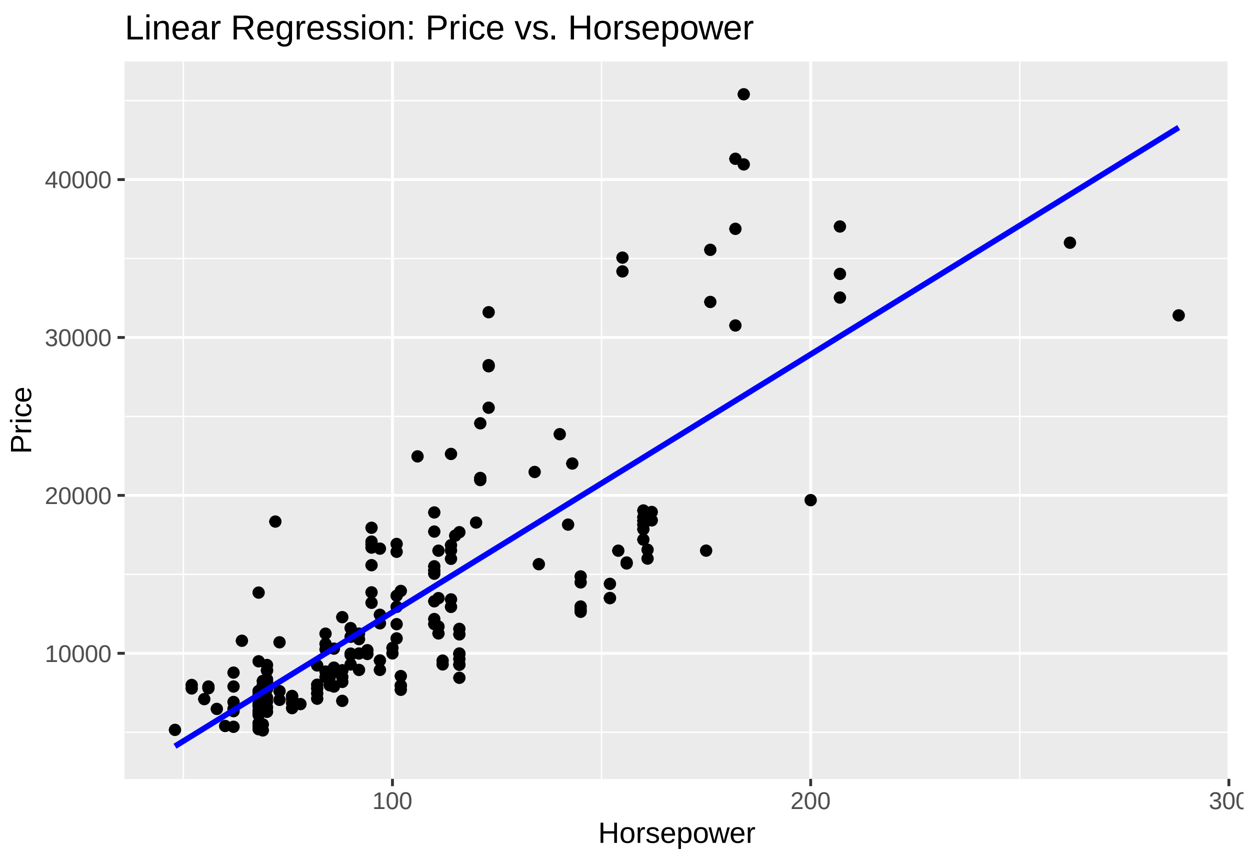
```r
model <- lm(price ~ horsepower, data = cars)
# Print the regression summary
summary(model)
```

```
##
## Call:
## lm(formula = price ~ horsepower, data = cars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11897.5  -2350.4   -711.1   1644.6  19081.4
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3721.761    929.849  -4.003 8.78e-05 ***
## horsepower    163.263      8.351  19.549  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4717 on 203 degrees of freedom
## Multiple R-squared:  0.6531, Adjusted R-squared:  0.6514
## F-statistic: 382.2 on 1 and 203 DF,  p-value: < 2.2e-16
```

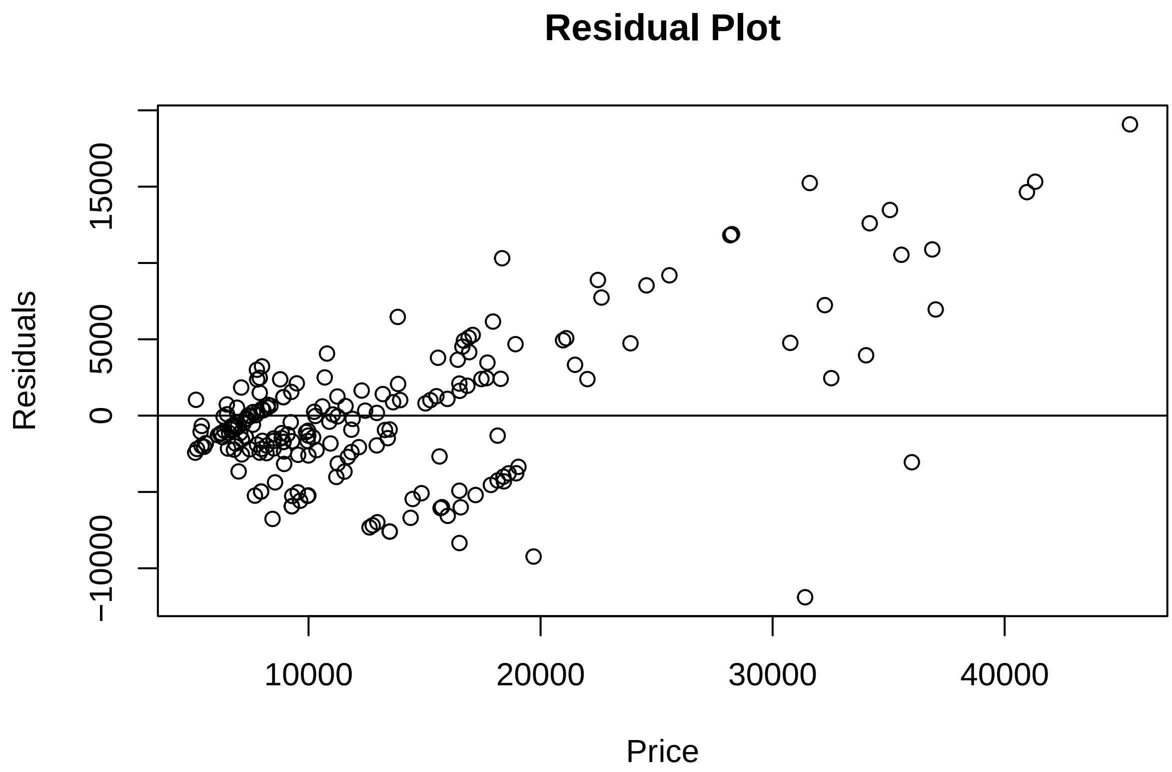
```r
ggplot(cars, aes(x = horsepower, y = price)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(title = "Linear Regression: Price vs. Horsepower",
       x = "Horsepower",
       y = "Price")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



What do you infer from your graph? The results don't seem to be very surprising. But there's something that's off about the scatter plot itself. Try plotting the residuals and analyzing if it's only white noise.

```r
res <- resid(model)
plot(cars$price, res,ylab = 'Residuals', xlab = 'Price', main = 'Residual Plot')
abline(0,0)
```



How will you tackle this problem? (Hint: Think about the different kind of transformations you've learnt in class)

```r
library(dplyr)
```

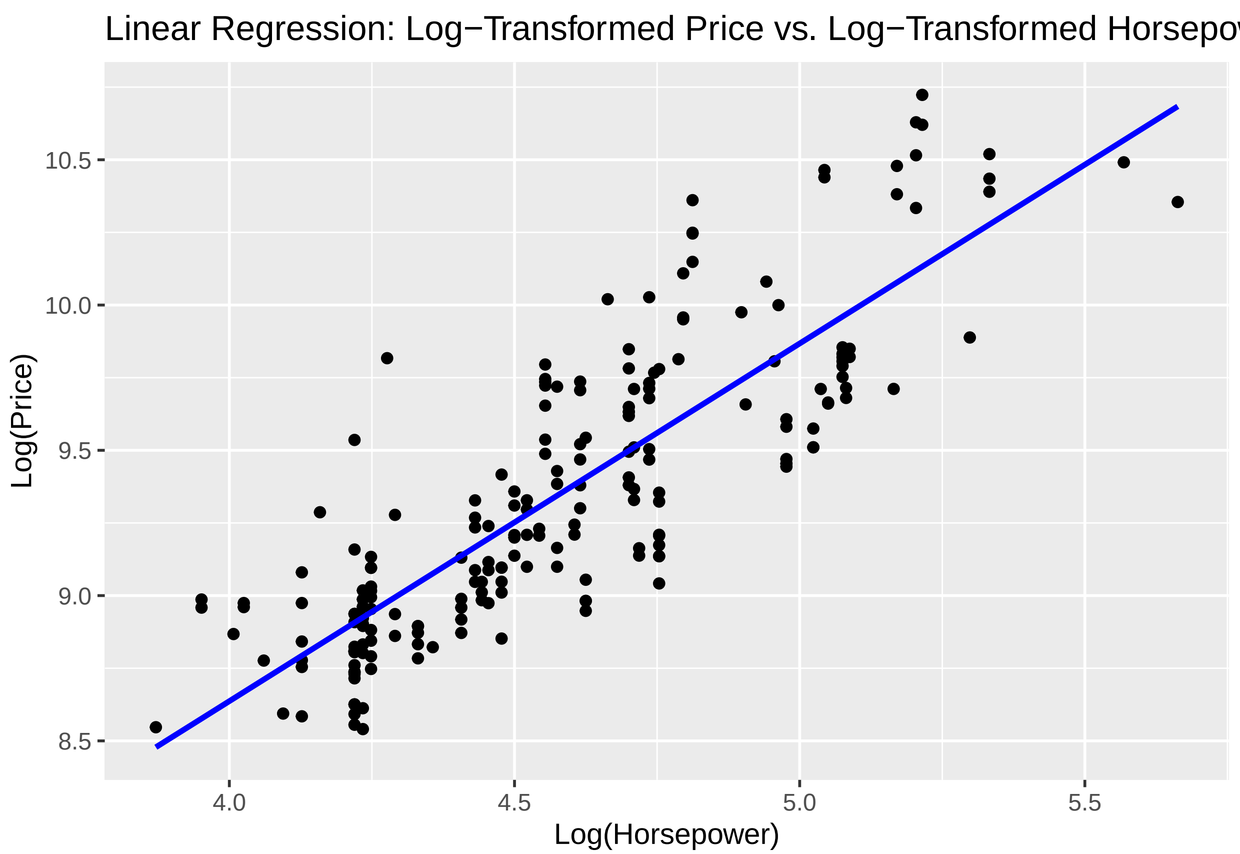
```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
log_price <- log(cars$price)
log_horsepower <- log(cars$horsepower)

# Perform linear regression on the transformed variables
model <- lm(log_price ~ log_horsepower)

# Print the regression summary
summary(model)
```

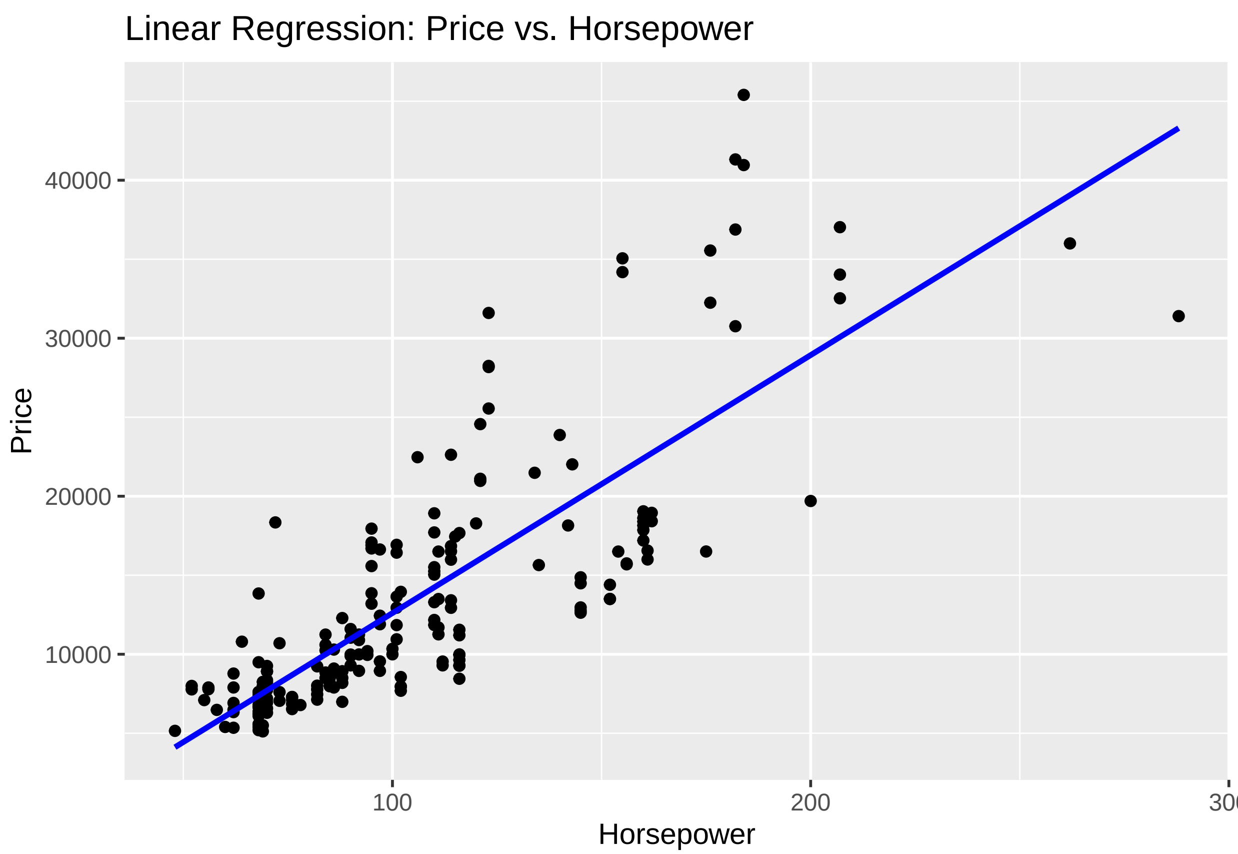
```
##
## Call:
## lm(formula = log_price ~ log_horsepower)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52252 -0.17992 -0.06097  0.17770  0.83985
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.71248    0.25367   14.63   <2e-16 ***
## log_horsepower  1.23103    0.05519   22.30   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2719 on 203 degrees of freedom
## Multiple R-squared:  0.7102, Adjusted R-squared:  0.7088
## F-statistic: 497.5 on 1 and 203 DF,  p-value: < 2.2e-16
```

```r
# Create a scatter plot with regression line for transformed variables
ggplot(cars, aes(x = log_horsepower, y = log_price)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(title = "Linear Regression: Log-Transformed Price vs. Log-Transformed Horsepower",
       x = "Log(Horsepower)",
       y = "Log(Price)")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
weights <- 1 / (cars$horsepower^2)

# Perform weighted linear regression
model <- lm(price ~ horsepower, data = cars, weights = weights)


# Create a scatter plot with regression line using weights
ggplot(cars, aes(x = horsepower, y = price)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE, color = "blue") +
  labs(title = "Linear Regression: Price vs. Horsepower",
       x = "Horsepower",
       y = "Price")
```



---

2. Logistic Regression

Logistic regression is an algorithm that estimates the parameters, or coefficients, of the linear combination of the logit model. The logistic or logit model is used to predict the probability 'p' of a binary dependent variable taking on one of two possible outcomes. This feature makes Logistic Regression useful even in problems of binary classification

DATA motors currently only build vehicles with rear-wheel drive. In America however, front-wheel drive is known to be quite popular too. Development of this technology will require significant investments into Research & Development. The client wants to know if they can recover costs quickly by charging a premium on front-wheel drive vehicles.

Analyze the price at which these two types of cars are sold and try to find out if Front-wheel Drive cars are indeed the premium variety in the market, or if rear-wheel drive vehicles can fetch high rates.

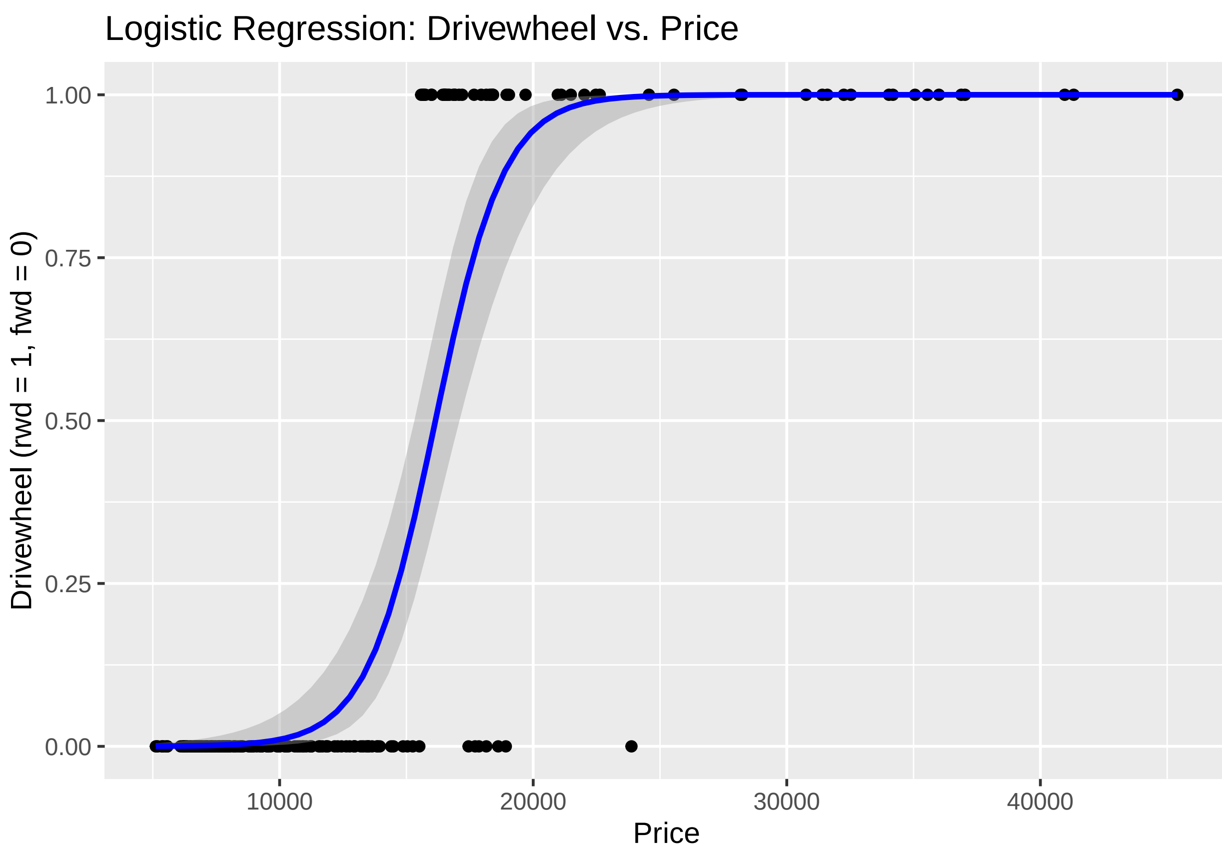
```r
# Convert 'drivewheel' to a binary factor variable: rwd = 1, fwd = 0
cars$drivewheel <- ifelse(cars$drivewheel == "rwd", 1, 0)

# Perform logistic regression
model <- glm(drivewheel ~ price, data = cars, family = binomial)

# Print the regression summary
summary(model)
```

```
##
## Call:
```

```
## glm(formula = drivewheel ~ price, family = binomial, data = cars)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3817  -0.1452  -0.0632   0.0001   1.3593
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.194e+01  2.115e+00  -5.645 1.66e-08 ***
## price        7.393e-04  1.329e-04   5.561 2.68e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 236.405  on 204  degrees of freedom
## Residual deviance:  68.364  on 203  degrees of freedom
## AIC: 72.364
##
## Number of Fisher Scoring iterations: 8
```

```r
# Create a scatter plot with logistic regression curve
ggplot(cars, aes(x = price, y = drivewheel)) +
  geom_point() +
  geom_smooth(method = "glm", method.args = list(family = "binomial"), color = "blue") +
  labs(title = "Logistic Regression: Drivewheel vs. Price",
       x = "Price",
       y = "Drivewheel (rwd = 1, fwd = 0)")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Is this good news or bad news for the client? As with most things, it's a bit of both. Go ahead and think about why that might be the case here.

Meanwhile let us try and see how good our logistic regression models are performing on the data. (Hint: Use the inbuilt functions in the pROC library)

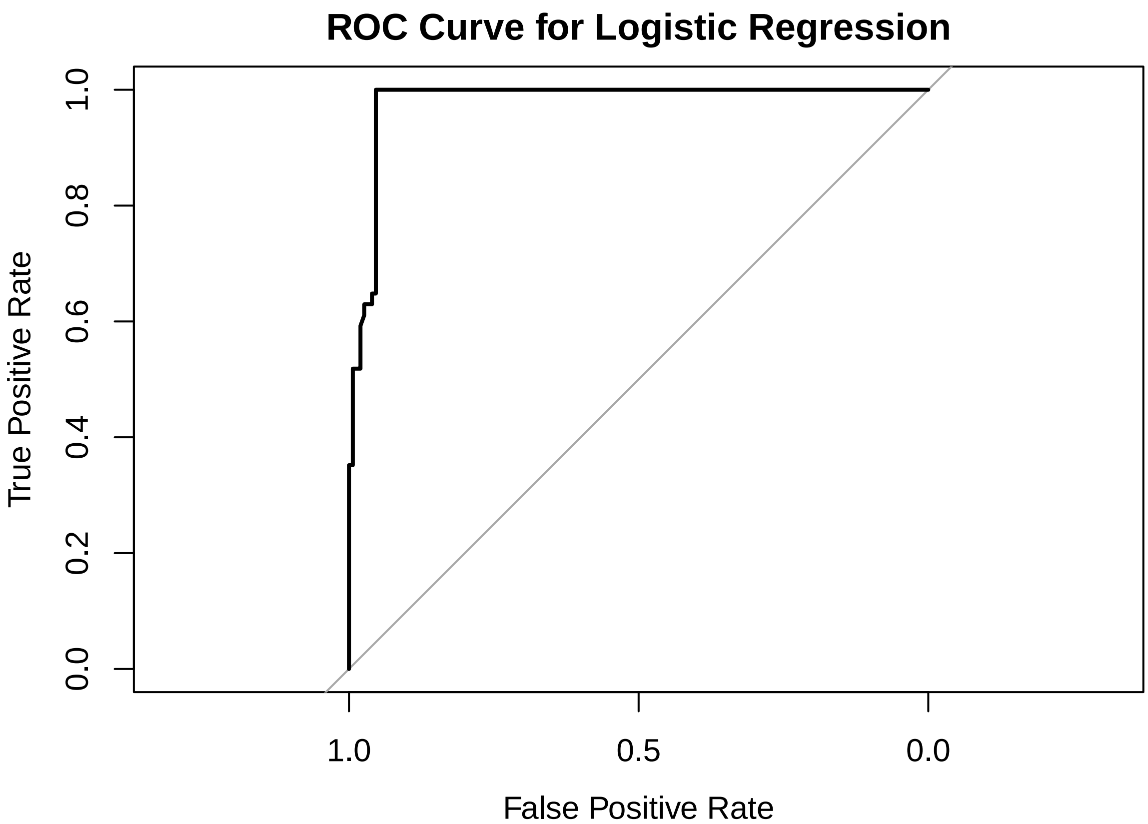
```r
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```r
# Calculate ROC curve
roc_curve <- roc(cars$drivewheel, predict(model, type = "response"))
```

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```

```r
plot(roc_curve, main = "ROC Curve for Logistic Regression",
     xlab = "False Positive Rate", ylab = "True Positive Rate")
```




```r
# Find optimal threshold
parameters <- coords(roc_curve, "best")
print(parameters)
```

```
##   threshold specificity sensitivity
## 1 0.3908323   0.9536424           1
```

Those are striking numbers. What does it say about our the drivewheel variable that our Logistic Regression models are able to achieve such high scores across metrics?

---

3. Multiple Linear Regression

For our Multiple Linear Regression models, we could use all the attributes and try to predict the price. But the aim is to always predict the maximum variation in the target, with the minimum variables.

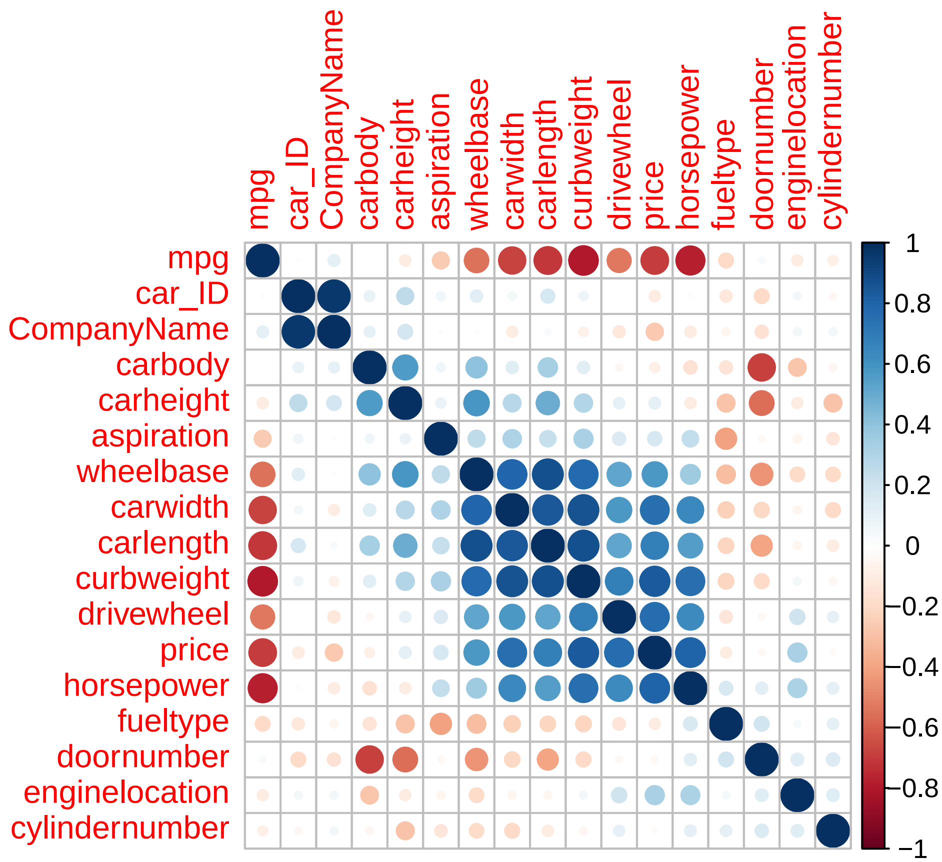
Thus, it's important to identify which features are most important to predict our target variable. Use the help of a correlogram to visually analyze the correlation between different independent variables and the one dependent variable. (Don't forget to keep an eye on the correlation between independent variables. Try and identify why it is important to do this.)

```r
library(dplyr)

cars <- cars %>%
  mutate(across(where(is.character), as.factor)) %>%
  mutate(across(where(is.factor), as.numeric))
```

```r
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```r
# Calculate the correlation matrix
correlation_matrix <- cor(cars)
testRes = cor.mtest(cars, conf.level = 0.95)

# Create the correlation heatmap
corrplot(correlation_matrix, p.mat = testRes$price, sig.level = 0.05, order = 'hclust')
```



We can now see that there are features positively correlated to price, and features negatively correlated to price. Let us use all the significant variables we have noticed in the correlogram in our Multiple Linear regression model.

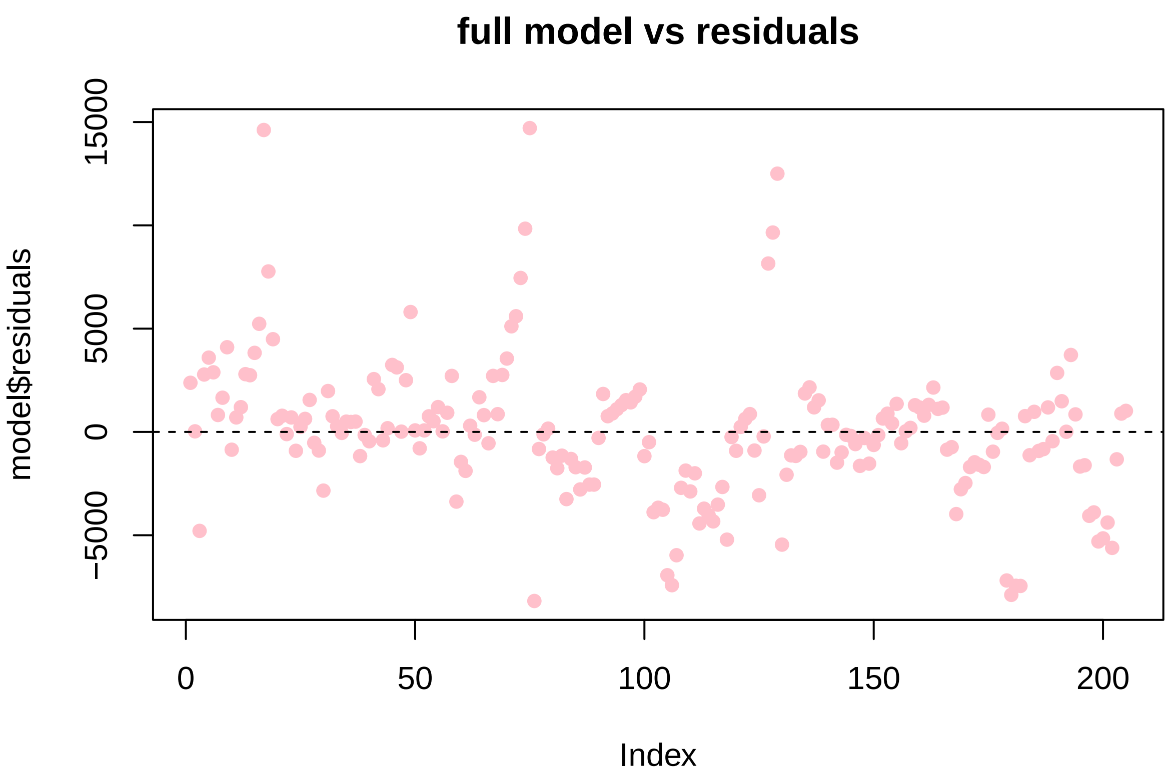
Use different variables to create the Multiple Linear Regression model and analyze the difference in residual values and F-statistic scores between each of them.

```r
library(dplyr)

model <- lm(price ~ curbweight + drivewheel + horsepower + carwidth, data = cars)

summary(model)
```

```
##
## Call:
## lm(formula = price ~ curbweight + drivewheel + horsepower + carwidth,
##     data = cars)
```

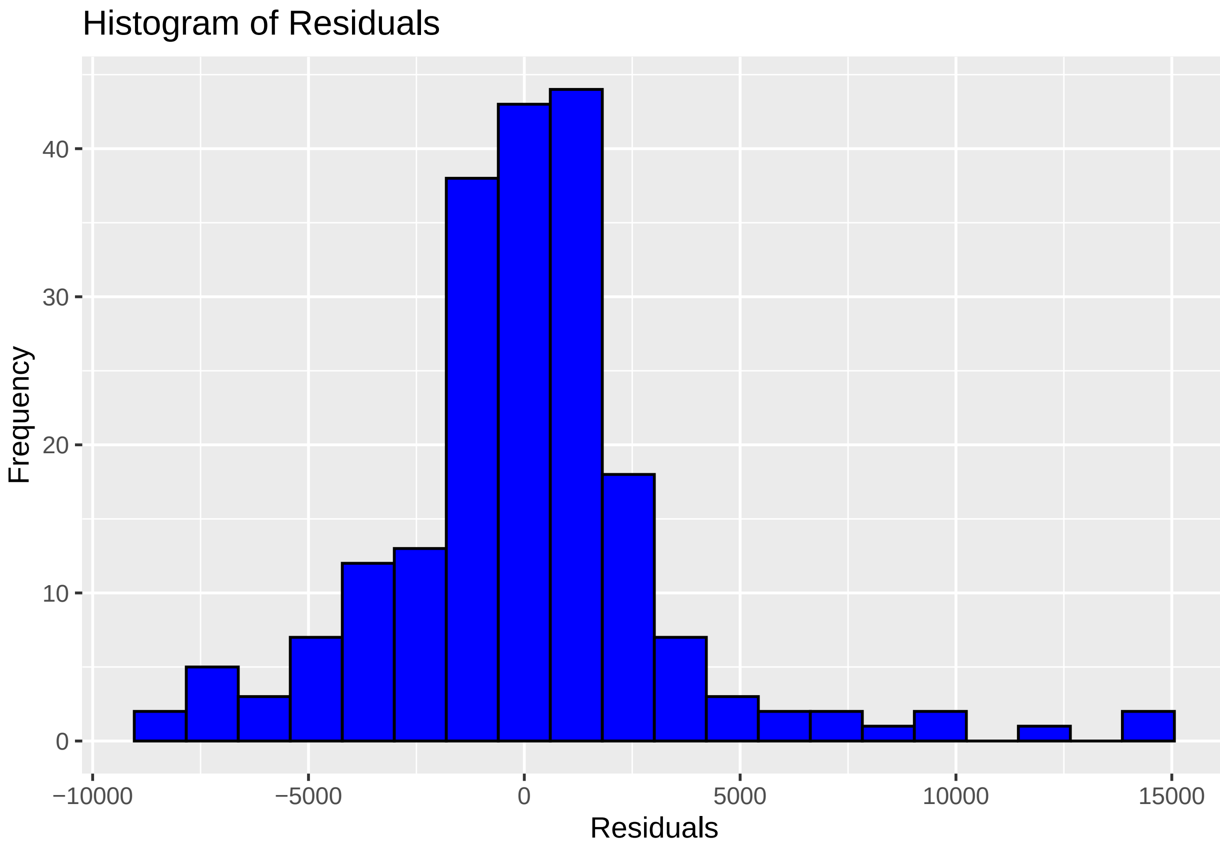
```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8183.4 -1534.9    14.9  1197.7 14707.8
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -47645.280  12708.453  -3.749 0.000232 ***
## curbweight       3.400      1.125   3.022 0.002844 **
## drivewheel    5355.430    767.763   6.975 4.36e-11 ***
## horsepower      68.759      9.397   7.317 6.02e-12 ***
## carwidth       662.494    223.135   2.969 0.003353 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3401 on 200 degrees of freedom
## Multiple R-squared:  0.8223, Adjusted R-squared:  0.8187
## F-statistic: 231.4 on 4 and 200 DF,  p-value: < 2.2e-16
```

```
plot(model$residuals, pch = 16, col = "pink",main ="full model vs residuals")
abline(h =0, lty =2)
```




```
library(ggplot2)

residuals <- resid(model)

ggplot() +
```

```r
geom_histogram(aes(x = residuals), bins = 20, fill = "blue", color = "black") +
labs(title = "Histogram of Residuals",
     x = "Residuals",
     y = "Frequency")
```



What can you infer about the fit of Multiple Linear Regression on to the given dataset?

Which are the most important variables to predict the price of the car?

How many variables did you use in your best fitting model? Which ones were they?

Good job with the analysis! DATA motors and Bangalore Consulting Group have both picked up valuable information from the work you just did.

The methods used in this worksheet form the fundamental basis for many more complex techniques and algorithms. As internship season is upon is, those of you who get to work in Data Science, Analytics etc will find yourselves using these very same techniques to answer the business questions posed by your organizations.

In a world where ChatGPT and DALL-E get all the Spotlight, classic ML techniques like Linear Regression still form the backbone of real world Analytics. The simplicity and interpretability of these models have made these models invaluable in providing insights to business owners across industries make informed, data-driven decisions.

Happy Learning!