Project 2 Final Part

Sajida Malik

2023-11-3

Introduction:

In this project, I aim to understand the factors that influence the pricing of used cars in the market. To achieve this goal, I will conduct an exploratory data analysis (EDA) and then I will build linear models based on my initial analysis of EDA. The dataset I will use is sourced from Kaggle, containing a plethora of entries on used car sales. The steps involved in this analysis include data import, cleaning, visualization, hypothesis testing and model building.

Dataset: https://www.kaggle.com/datasets/tsaustin/us-used-car-sales-data

Methodology:

Reading and Understanding the Data:

I will start by importing the dataset and gaining a comprehensive understanding of its structure. This includes identifying numerical and categorical variables, which will guide my analysis. Data Cleaning and Preparation:

I will address data quality issues, such as checking for and removing duplicate entries, handling missing values, and fixing any errors in the data set. A clean dataset is essential for accurate analysis. Visualizing the Data:

I will employ various data visualization techniques, including histograms, boxplots, and scatter plots. These visualizations will help me gain insights into the data and draw preliminary inferences, such as the preferred car company (e.g., Ford), car type (e.g., SUV), and the price differences between car types (e.g., SUV and Sedan).

Hypothesis Testing:

Based on my initial inferences from visual analysis, I will formulate research questions and express them in a hypothesis testing framework. I will choose 3-4 questions that are relevant to understanding the factors affecting used car prices. I will perform hypothesis tests using R to address these questions. For example, I might test hypotheses related to the impact of car make, car type, mileage, or age on car prices. Results of these hypothesis tests will provide statistical evidence regarding the relationships between various factors and car prices. This will help me draw more robust conclusions about the factors influencing used car prices in the market. In this part of the project, the focus is on data exploration, visualization, and hypothesis testing to better understand the determinants of used car prices.

Linear Model Building: After testing my hypotheses I will build a few linear models to check the linear relationship bewteen the response varibale and independent variables. After that I will conduct residual analysis of models by checking assumptions. 1. The residuals of the model are nearly normal, 2. The variability of the residuals is nearly constant, 3. The residuals are independent, 4. Each variable is linearly related to the outcome.

GLM Models:

After developing linear models, I extended my analysis by constructing several generalized linear models (GLM). These GLM models incorporated additional categorical variables through a stepwise model-building procedure. To assess the performance and appropriateness of each constructed model, I utilized the Akaike Information Criterion (AIC) and conducted leave-one-out cross-validation evaluations. These evaluations provided valuable insights into the predictive capabilities and generalization performance of the extended GLM models.

Results:

#Importing neccessary libraries

Step 1: Reading and Understanding the Data

#Importing the data

```
# Load the dataset
data <-
read.csv("https://raw.githubusercontent.com/Sajida28/Used_Car_Sale/main/used_
car_sales.csv", na.strings = c("NA", "0", ""))
```

#Understanding the structure of the data (identify the numerical and categorical variables)

```
class(data)

## [1] "data.frame"

glimpse(data)  # For glimpse of data we can also use str(data) functions for
this but glimpse is better for

## Rows: 122,144
## Columns: 13
## $ ID            <int> 137178, 96705, 119660, 80773, 64287, 132695, 132829,
5250…
## $ pricesold     <int> 7500, 15000, 8750, 11600, 44000, 950, 950, 70000,
1330, 2…
## $ yearsold      <int> 2020, 2019, 2020, 2019, 2019, 2020, 2020, 2019, 2019,
201…
## $ zipcode       <chr> "786**", "81006", "33449", "07852", "07728", "462**",
"10…
## $ Mileage       <int> 84430, NA, 55000, 97200, 40703, 71300, 71300, 6500,
16700…
## $ Make          <chr> "Ford", "Replica/Kit Makes", "Jaguar", "Ford",
```

```
"Porsche",…
## $ Model       <chr> "Mustang", "Jaguar Beck Lister", "XJS", "Mustang",
"911",…
## $ Year        <int> 1988, 1958, 1995, 1968, 2002, 1965, 1965, 1997, 2001,
197…
## $ Trim        <chr> "LX", NA, "2+2 Cabriolet", "Stock", "Turbo X-50", NA,
NA,…
## $ Engine      <chr> "5.0L Gas V8", "383 Fuel injected", "4.0L In-Line 6
Cylin…
## $ BodyType    <chr> "Sedan", "Convertible", "Convertible", "Coupe",
"Coupe", …
## $ NumCylinders <int> NA, 8, 6, 8, 6, NA, NA, NA, 4, NA, 6, 6, 8, 4, NA, 6,
6, …
## $ DriveType   <chr> "RWD", "RWD", "RWD", "RWD", "AWD", "RWD", NA, "4WD",
"FWD…
```

```r
summary(data) #summary of data
```

```
##        ID           pricesold        yearsold      zipcode
##  Min.   :     1   Min.   :    10   Min.   :2018   Length:122144
##  1st Qu.: 44547   1st Qu.:  2950   1st Qu.:2019   Class :character
##  Median : 85556   Median :  6500   Median :2019   Mode  :character
##  Mean   : 85094   Mean   : 10811   Mean   :2019
##  3rd Qu.:127079   3rd Qu.: 13800   3rd Qu.:2020
##  Max.   :165801   Max.   :404990   Max.   :2020
##                   NA's   :31
##     Mileage               Make               Model                Year
##  Min.   :         1   Length:122144      Length:122144      Min.   :
68
##  1st Qu.:     48400   Class :character   Class :character   1st Qu.:
1977
##  Median :     92000   Mode  :character   Mode  :character   Median :
2000
##  Mean   :   1439131                                         Mean   :
3960
##  3rd Qu.:    142000                                         3rd Qu.:
2008
##  Max.   :1235668876                                         Max.
:20140000
##  NA's   :2957                                               NA's   :14
##      Trim              Engine            BodyType         NumCylinders
##  Length:122144      Length:122144      Length:122144      Min.   :
1
##  Class :character   Class :character   Class :character   1st Qu.:
6
##  Mode  :character   Mode  :character   Mode  :character   Median :
6
##                                                           Mean   :
23308
##                                                           3rd Qu.:
```

```
8
##                                                              Max.
:2147483647
##                                                              NA's   :29981
##    DriveType
##   Length:122144
##   Class :character
##   Mode  :character
##
##
##
##
```

Step 2: Data Cleaning and Preparation #Checking and removing duplicates

```
duplicates <- duplicated(data)
num_duplicates <- sum(duplicates)
num_duplicates

## [1] 0
```

#Checking entries with missing values

```
missing_values <- colSums(is.na(data))
missing_values

##           ID    pricesold     yearsold      zipcode      Mileage
Make
##            0           31            0          909         2957
0
##        Model         Year         Trim       Engine     BodyType
NumCylinders
##          573           14        48903        27067        20782
29981
##    DriveType
##        24839
```

Remove rows with missing values

```
# Using filter() and complete.cases()
data <- data %>%
  filter(complete.cases(.))
head(data)

##       ID pricesold yearsold zipcode Mileage    Make    Model Year
Trim
## 1 119660      8750     2020   33449   55000  Jaguar      XJS 1995 2+2
Cabriolet
## 2  80773     11600     2019   07852   97200    Ford  Mustang 1968
Stock
## 3  64287     44000     2019   07728   40703 Porsche      911 2002    Turbo
```

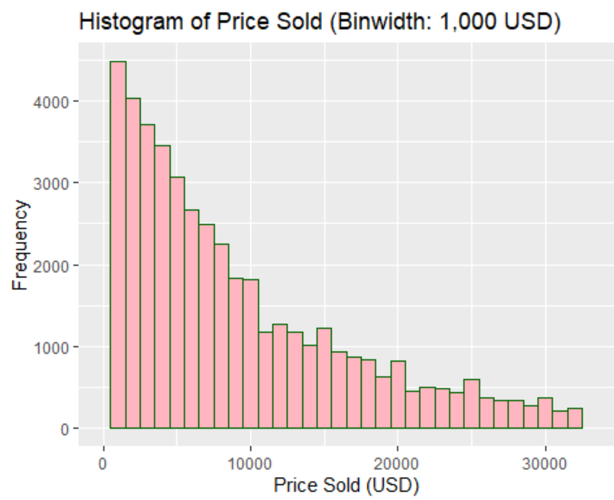
```
X-50
## 4 158271      20000      2020   333**    51674    Jeep Wrangler 2015
SPORT
## 5  72418      14100      2019   07014   109500    Jeep Wrangler 2012
Unlimited
## 6 144540       3330      2020   856**    47692   Buick  LeSabre 2004
CUSTOM
##                    Engine    BodyType NumCylinders DriveType
## 1 4.0L In-Line 6 Cylinder Convertible            6       RWD
## 2          289 cu. in. V8       Coupe            8       RWD
## 3                    3.6L       Coupe            6       AWD
## 4        3.6L Flexible V6         SUV            6       4WD
## 5                    3.6L         SUV            6       4WD
## 6             3.8L Gas V6       Sedan            6       FWD
```

#Cleaning Price variable

```r
# Calculate the IQR
Q1 <- quantile(data$pricesold, 0.25)
Q3 <- quantile(data$pricesold, 0.75)
IQR <- Q3 - Q1
# Define lower and upper bounds for outliers
lower_bound <- Q1 - 1.5 * IQR
upper_bound <- Q3 + 1.5 * IQR
# Remove outliers
data <- data %>%
  filter(pricesold >= lower_bound, pricesold <= upper_bound)
```

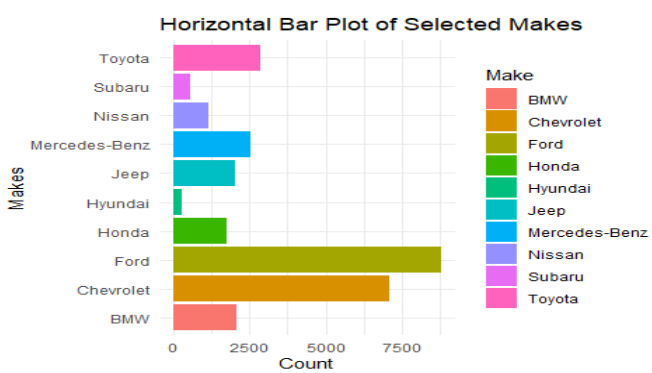
#Histogram with cleaned data

```r
# Create a histogram of cleaned Price Sold
ggplot(data, aes(x = pricesold)) +
  geom_histogram(binwidth = 1000, fill = "lightpink", color = "darkgreen") +
  labs(x = "Price Sold (USD)", y = "Frequency") +
  ggtitle("Histogram of Price Sold (Binwidth: 1,000 USD)") +
  xlim(0, max(data$pricesold))

## Warning: Removed 2 rows containing missing values (`geom_bar()`).
```



#Barplot of Selected Make

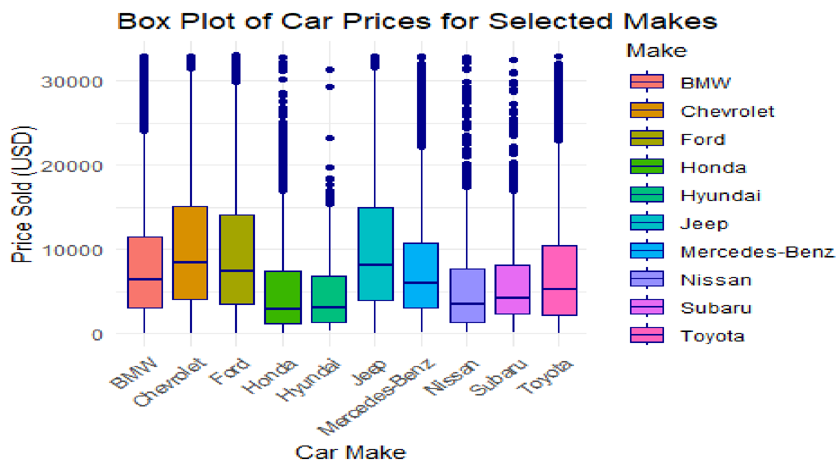
```
# Define a vector of makes you want to include
selected_makes1 <- c("BMW", "Ford", "Toyota", "Chevrolet", "Honda", "Nissan",
"Mercedes-Benz", "Hyundai", "Subaru", "Jeep")

# Filter your dataset to include only the selected makes
filtered_data1 <- data[data$Make %in% selected_makes1, ]

# Create a horizontal bar plot with different colors for each make
ggplot(filtered_data1, aes(y = Make, fill = Make)) +
  geom_bar() +
  labs(y = "Makes", x = "Count") +
  ggtitle("Horizontal Bar Plot of Selected Makes") +
  theme_minimal()
```



#Boxplot of selected Makes with cleaned sold prices

```
# Create a box plot with different colors and tilted x-axis legends
ggplot(filtered_data1, aes(x = Make, y = pricesold)) +
  geom_boxplot(aes(fill = Make), color = "darkblue") + # Set fill to Make for
different colors
  labs(x = "Car Make", y = "Price Sold (USD)") +
```

```
  ggtitle("Box Plot of Car Prices for Selected Makes") +
  theme_minimal() +
  scale_x_discrete(labels = scales::wrap_format(10)) + # Tilt x-axis labels
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Adjust label
angle and position
```



#Cleaning Year variable

```
#Cleaning Year column
data$Year <- as.character(data$Year)  # Convert Year to character for
manipulation

# Remove the last 4 digits when the total number of digits is greater than 4
data$Year <- ifelse(nchar(data$Year) > 4, substr(data$Year, 1,
nchar(data$Year) - 4), data$Year)

# Convert Year back to numeric if needed
data$Year <- as.numeric(data$Year)
```

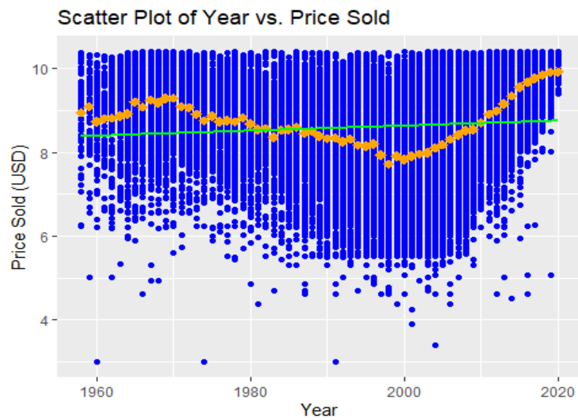
#Removin outliers

```
# Calculate the IQR
Q1_Year <- quantile(data$Year, 0.25)
Q3_Year <- quantile(data$Year, 0.75)
IQR <- Q3_Year - Q1_Year

# Define lower and upper bounds for outliers
lower_bound_Year <- Q1_Year - 1.5 * IQR
upper_bound_Year <- Q3_Year + 1.5 * IQR

# Remove outliers
data <- data %>%
  filter(Year >= lower_bound_Year, Year <= upper_bound_Year)
summary(data)
```

```
##        ID           pricesold        yearsold      zipcode
##  Min.   :      1   Min.   :   20   Min.   :2018   Length:44614
##  1st Qu.: 42718   1st Qu.: 2950   1st Qu.:2019   Class :character
##  Median : 82711   Median : 6500   Median :2019   Mode  :character
##  Mean   : 82478   Mean   : 8871   Mean   :2019
##  3rd Qu.:121524   3rd Qu.:13000   3rd Qu.:2020
##  Max.   :165799   Max.   :33070   Max.   :2020
##     Mileage                 Make                Model                Year
##  Min.   :          1   Length:44614       Length:44614       Min.   :1958
##  1st Qu.:      54596   Class :character   Class :character   1st Qu.:1991
##  Median :      98837   Mode  :character   Mode  :character   Median :2003
##  Mean   :     466558                                         Mean   :1999
##  3rd Qu.:     149639                                         3rd Qu.:2010
##  Max.   :1234567890                                          Max.   :2020
##      Trim              Engine             BodyType          NumCylinders
##  Length:44614       Length:44614       Length:44614       Min.   :
1
##  Class :character   Class :character   Class :character   1st Qu.:
6
##  Mode  :character   Mode  :character   Mode  :character   Median :
6
##                                                           Mean   :
48141
##                                                           3rd Qu.:
8
##                                                           Max.
:2147483647
##   DriveType
##  Length:44614
##  Class :character
##  Mode  :character
##
##
##
```

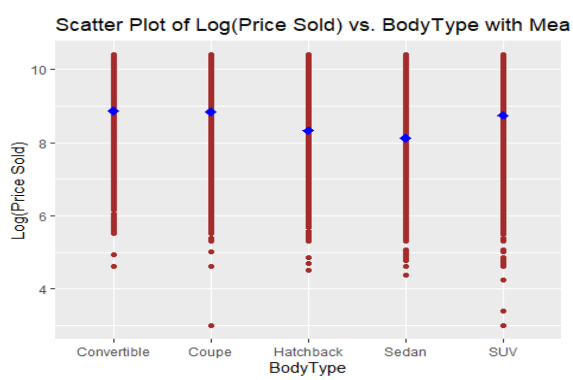
#Scatter plot of Cleaned Year and pricesold

```
ggplot(data, aes(x = Year, y = log(pricesold))) +
  geom_point(color = "blue") +
  stat_summary(fun = "mean", geom = "point", color = "orange", size = 3,
shape = 18) +
  labs(x = "Year", y = "Price Sold (USD)") +
  ggtitle("Scatter Plot of Year vs. Price Sold") +
  geom_smooth(method = "lm", se = FALSE, color = "green")

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



#To make it clearer #Tranform price sold and selcetd bodytype scatter plot

```
# Define the selected BodyTypes you want to include
selected_BodyTypes <- c("SUV", "Sedan", "Convertible", "Coupe", "Hatchback")

# Filter your dataset to include only the selected BodyTypes
filtered_data <- data[data$BodyType %in% selected_BodyTypes, ]

# Create a scatter plot of log(pricesold) against BodyType with mean points
ggplot(filtered_data, aes(x = BodyType, y = log(pricesold))) +
  geom_point(color = "brown") +
  stat_summary(fun = "mean", geom = "point", color = "blue", size = 3, shape
= 18) +
  labs(x = "BodyType", y = "Log(Price Sold)") +
  ggtitle("Scatter Plot of Log(Price Sold) vs. BodyType with Mean Points")
```



Initial Inferences: From Visualization the initial inferences are: 1- People prefer American Car brands over foreign brands 2- People prefer SUVs over all other body types. 3- People prefer classic cars from the 60's than the classic cars from 70's

Questions about my initial inferences: 1- Do people prefer American car brands over foreign brands? 2- Do people prefer SUVs over all other BodyTypes? 3- Do people prefer classic cars from the 60's than classic cars from 70's?

Hypothesis Testing: 1- People prefer American Car brands over foreign brands:

Null Hypothesis (H0): There is no significant difference in the mean selling price between American and foreign car brands among buyers of used cars. Alternative Hypothesis (H1): There is a significant difference in the mean selling price between American and foreign car brands among buyers of used cars.

```
# Lists of American and foreign car brands
american_brands <- c("Ford", "Chevrolet", "Jeep")
foreign_brands <- c("Honda", "Nissan", "Mercedes-Benz", "BMW", "Hyundai",
"Subaru", "Toyota")

# Create two groups: American and Foreign car brands
data <- data %>%
  mutate(CarGroup = case_when(
    Make %in% american_brands ~ "American",
    Make %in% foreign_brands ~ "Foreign"
  ))

# Perform a two-sample t-test
t_test_result <- t.test(pricesold ~ CarGroup, data = data)

# Print the t-test result
print(t_test_result)

##
##  Welch Two Sample t-test
##
## data:  pricesold by CarGroup
## t = 29.562, df = 25897, p-value < 0.0000000000000022
## alternative hypothesis: true difference in means between group American
and group Foreign is not equal to 0
## 95 percent confidence interval:
##  2435.245 2781.106
## sample estimates:
## mean in group American  mean in group Foreign
##               9834.890               7226.714
```

2-People prefer SUVs over Sedans:

Null Hypothesis (H0): There is no significant preference for SUVs over Sedans among buyers of used cars. Alternative Hypothesis (H1): There is a significant preference for SUVs over Sedans among buyers of used cars.

```
#first I check the frequency of Body Types.
#Then I will apply t-test to check mean higher price difference on two most
popular BodyTypes
# Create a contingency table
contingency_table <- table(filtered_data$BodyType)
contingency_table
```

```
##
## Convertible       Coupe   Hatchback       Sedan         SUV
##        4258        6479        1287        9048        9381

#Testing mean selling price among SUV and Sedan
# Create subsets for Sedan and SUV
suv_prices <- filtered_data$pricesold[filtered_data$BodyType == "SUV"]
sedan_prices <- filtered_data$pricesold[filtered_data$BodyType == "Sedan"]

# Perform an independent t-test with a two-tailed test
t_test_result <- t.test( suv_prices, sedan_prices)
# View the t-test results
print(t_test_result)
##  Welch Two Sample t-test
## data:  suv_prices and sedan_prices
## t = 33.424, df = 18098, p-value < 0.00000000000000022
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  3378.233 3799.132
## sample estimates:
## mean of x mean of y
##  9734.815  6146.133
```

3- People prefer classic cars from the 60's than classic cars from the 70's: To check the preference for classic cars from the 60's and 70's based on mean selling prices, formulating the hypothesis:

Null Hypothesis (H0): There is no significant difference in mean selling prices between classic cars from the 60's and classic cars from the 70's among buyers of used cars.
Alternative Hypothesis (H1): People significantly prefer classic cars from the 60's over classic cars from the 70's when buying used cars based on mean selling prices.

```
# Create subsets for classic cars from the 60's and 70's based on the "Year"
variable
classic_cars_60s <- data$pricesold[data$Year >= 1960 & data$Year <= 1969]
classic_cars_70s <- data$pricesold[data$Year >= 1970 & data$Year <= 1979]

# Perform a two-sample t-test
t_test_result <- t.test(classic_cars_60s, classic_cars_70s)

# View the t-test results
print(t_test_result)

##
##  Welch Two Sample t-test
##
## data:  classic_cars_60s and classic_cars_70s
## t = 11.022, df = 5638.2, p-value < 0.00000000000000022
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
##  1818.287 2605.050
## sample estimates:
## mean of x mean of y
##  12382.93  10171.26
```

Part 2: Linear models: The linear model is most widely used statistical model. Given response Y is the 'pricesold' and explanatory variables are X1 is 'Mileage' and X2 is 'age_of_car, in my linear regression model. $Y = \beta 0 + \beta 1X1 + \beta 2X2 + \epsilon$, where $\epsilon \sim N(0, \sigma 2)$ Goal is to estimate parameters, also called coefficients β0, β1 and β2. Fitting a linear regression model with lm()

```
#Before creating linear models
#Create a new variable 'Age' by subtracting 'Year' from 'yearsold'
data$Age_of_car <- data$yearsold - data$Year
#Checking first few rows of data
head(data)
```

```
##       ID pricesold yearsold zipcode Mileage      Make    Model Year
## 1 119660      8750     2020   33449   55000    Jaguar      XJS 1995
## 2  80773     11600     2019   07852   97200      Ford  Mustang 1968
## 3 158271     20000     2020   333**   51674      Jeep Wrangler 2015
## 4  72418     14100     2019   07014  109500      Jeep Wrangler 2012
## 5 144540      3330     2020   856**   47692     Buick  LeSabre 2004
## 6  59728     18550     2019   60448    6714 Chevrolet   Camaro 2002
##            Trim                 Engine    BodyType NumCylinders DriveType
## 1 2+2 Cabriolet 4.0L In-Line 6 Cylinder Convertible            6       RWD
## 2         Stock          289 cu. in. V8       Coupe            8       RWD
## 3         SPORT         3.6L Flexible V6         SUV            6       4WD
## 4     Unlimited                   3.6L         SUV            6       4WD
## 5        CUSTOM             3.8L Gas V6       Sedan            6       FWD
## 6    Z28,SS,SLP            5.7 liter v8       Coupe            8       RWD
##   CarGroup Age_of_car
## 1     <NA>         25
## 2 American         51
## 3 American          5
## 4 American          7
## 5     <NA>         16
## 6 American         17
```

There are still some values which needs to be removed to get better picture of the data set

```
# Define a function to remove outliers by filtering
remove_outliers <- function(data, variable) {
  Q1 <- quantile(variable, 0.25)
  Q3 <- quantile(variable, 0.75)
  IQR <- Q3 - Q1
  lower_bound <- Q1 - 1.5 * IQR
  upper_bound <- Q3 + 1.5 * IQR
  filtered_data <- data %>% filter(variable >= lower_bound, variable <=
upper_bound)
  return(filtered_data)
```

```
}

# Remove outliers from each variable and update the data frame
data <- remove_outliers(data, data$Mileage)
data <- remove_outliers(data, data$pricesold)
data <- remove_outliers(data, data$Age_of_car)
```

#Filtering data for further analysis

```
# Defining a function to remove values below Q1
remove_below_Q1 <- function(data, variable_name) {
  data %>% filter({{ variable_name }} >= quantile({{ variable_name }}, 0.25))
}
# Remove values below Q1 for each variable and update the data frame
data <- data %>%
  remove_below_Q1(Mileage) %>%
  remove_below_Q1(pricesold) %>%
  remove_below_Q1(Age_of_car)
summary(data)
```

```
##        ID            pricesold        yearsold       zipcode
##  Min.   :     1   Min.   : 2200   Min.   :2018   Length:18037
##  1st Qu.: 42243   1st Qu.: 3800   1st Qu.:2019   Class :character
##  Median : 82092   Median : 6000   Median :2019   Mode  :character
##  Mean   : 82204   Mean   : 7523   Mean   :2019
##  3rd Qu.:122504   3rd Qu.: 9500   3rd Qu.:2020
##  Max.   :165792   Max.   :28210   Max.   :2020
##     Mileage           Make               Model                Year
##  Min.   : 56800   Length:18037       Length:18037       Min.   :1964
##  1st Qu.: 86406   Class :character   Class :character   1st Qu.:1987
##  Median :114736   Mode  :character   Mode  :character   Median :1999
##  Mean   :124386                                         Mean   :1995
##  3rd Qu.:153800                                         3rd Qu.:2005
##  Max.   :292000                                         Max.   :2009
##      Trim              Engine            BodyType          NumCylinders
##  Length:18037       Length:18037       Length:18037       Min.   :
2
##  Class :character   Class :character   Class :character   1st Qu.:
6
##  Mode  :character   Mode  :character   Mode  :character   Median :
8
##                                                           Mean   :
119067
##                                                           3rd Qu.:
8
##                                                           Max.
:2147483647
##   DriveType           CarGroup           Age_of_car
##  Length:18037       Length:18037       Min.   :11.00
##  Class :character   Class :character   1st Qu.:14.00
```

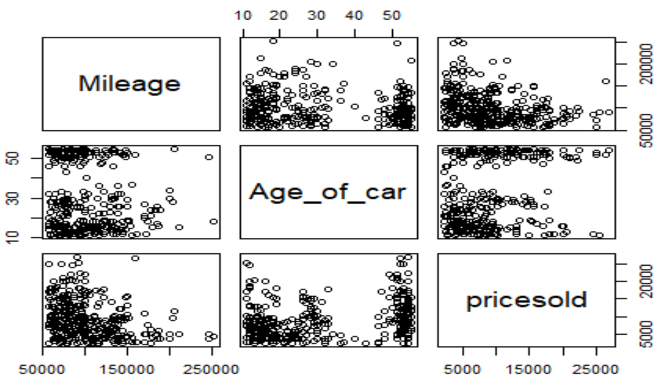
```
##  Mode  :character   Mode  :character   Median :20.00
##                                         Mean   :24.66
##                                         3rd Qu.:33.00
##                                         Max.   :55.00
```

#Model building among different categories

```
#Filtering the data
#selecting categories from Make, Bodytype with specific Model
Category1 <- data[data$Make == "Ford" & data$BodyType == "Coupe" & data$Model
== "Mustang", ]
summary(Category1)
```

```
##        ID           pricesold        yearsold      zipcode
##  Min.   :   112   Min.   : 2200   Min.   :2018   Length:344
##  1st Qu.: 38745   1st Qu.: 4972   1st Qu.:2019   Class :character
##  Median : 73908   Median : 8250   Median :2019   Mode  :character
##  Mean   : 74674   Mean   : 9216   Mean   :2019
##  3rd Qu.:111164   3rd Qu.:12500   3rd Qu.:2020
##  Max.   :165275   Max.   :27000   Max.   :2020
##     Mileage           Make              Model                Year
##  Min.   : 56937   Length:344         Length:344         Min.   :1964
##  1st Qu.: 75533   Class :character   Class :character   1st Qu.:1967
##  Median : 94588   Mode  :character   Mode  :character   Median :1992
##  Mean   :102062                                         Mean   :1987
##  3rd Qu.:122111                                         3rd Qu.:2004
##  Max.   :252500                                         Max.   :2009
##      Trim              Engine            BodyType          NumCylinders
##  Length:344         Length:344         Length:344         Min.   :4.000
##  Class :character   Class :character   Class :character   1st Qu.:8.000
##  Mode  :character   Mode  :character   Mode  :character   Median :8.000
##                                                           Mean   :7.645
##                                                           3rd Qu.:8.000
##                                                           Max.   :8.000
##   DriveType           CarGroup           Age_of_car
##  Length:344         Length:344         Min.   :11.00
##  Class :character   Class :character   1st Qu.:16.00
##  Mode  :character   Mode  :character   Median :27.00
##                                         Mean   :32.21
##                                         3rd Qu.:52.00
##                                         Max.   :55.00
```

```
#creating pairplot matrix to check initial relationship

pairs(Category1[, c("Mileage", "Age_of_car", "pricesold")])
```



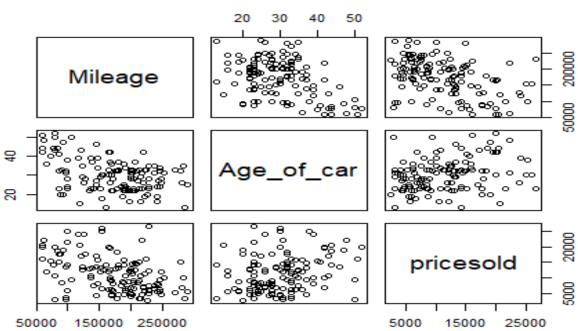

```r
#Creating linear model 1
Model_1 <- lm(pricesold ~ Mileage + Age_of_car, data = Category1)
summary(Model_1)
```

```
##
## Call:
## lm(formula = pricesold ~ Mileage + Age_of_car, data = Category1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9778.5 -3336.9  -638.4  2652.1 17922.0
##
## Coefficients:
##                Estimate   Std. Error t value             Pr(>|t|)
## (Intercept) 10594.626632  1008.799087  10.502 < 0.0000000000000002 ***
## Mileage        -0.039520     0.007452  -5.303          0.000000205 ***
## Age_of_car     82.413389    15.834559   5.205          0.000000336 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4894 on 341 degrees of freedom
## Multiple R-squared:  0.1586, Adjusted R-squared:  0.1537
## F-statistic: 32.14 on 2 and 341 DF,  p-value: 0.0000000000001635
```

```r
#Filtering the data for category 2
#selecting categories from Make, Bodytype and specific Model
Category2 <- data[data$Make == "Toyota" & data$BodyType == "SUV" & data$Model
== "Land Cruiser", ]
summary(Category2)
```

```
##        ID           pricesold        yearsold      zipcode
##  Min.   :   384   Min.   : 2600   Min.   :2018   Length:133
##  1st Qu.: 49158   1st Qu.: 6500   1st Qu.:2019   Class :character
##  Median : 92790   Median :10700   Median :2019   Mode  :character
##  Mean   : 89686   Mean   :11403   Mean   :2019
##  3rd Qu.:136674   3rd Qu.:15000   3rd Qu.:2020
##  Max.   :165385   Max.   :26600   Max.   :2020
```

```
##     Mileage            Make               Model               Year     
##  Min.   : 58841   Length:133         Length:133         Min.   :1967  
##  1st Qu.:134812   Class :character   Class :character   1st Qu.:1985  
##  Median :177553   Mode  :character   Mode  :character   Median :1989  
##  Mean   :172416                                         Mean   :1989  
##  3rd Qu.:214391                                         3rd Qu.:1996  
##  Max.   :290000                                         Max.   :2007  
##      Trim              Engine            BodyType          NumCylinders  
##  Length:133         Length:133         Length:133         Min.   :4.000  
##  Class :character   Class :character   Class :character   1st Qu.:6.000  
##  Mode  :character   Mode  :character   Mode  :character   Median :6.000  
##                                                           Mean   :6.165  
##                                                           3rd Qu.:6.000  
##                                                           Max.   :8.000  
##   DriveType           CarGroup           Age_of_car   
##  Length:133         Length:133         Min.   :13.00  
##  Class :character   Class :character   1st Qu.:23.00  
##  Mode  :character   Mode  :character   Median :30.00  
##                                        Mean   :30.24  
##                                        3rd Qu.:35.00  
##                                        Max.   :52.00

#creating pairplot matrix to check initial relationship
pairs(Category2[, c("Mileage", "Age_of_car", "pricesold")])
```




```
#Creating linear model 2
Model_2 <- lm(pricesold ~ Mileage + Age_of_car, data = Category2)
summary(Model_2)

## 
## Call:
## lm(formula = pricesold ~ Mileage + Age_of_car, data = Category2)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -13661.6  -3329.7   -694.4   3060.6  18056.7 
## 
## Coefficients:
##                 Estimate   Std. Error t value    Pr(>|t|)
```

```
## (Intercept) 16298.314771  2861.307166   5.696 0.000000078 ***
## Mileage        -0.041194     0.008758  -4.703 0.000006441 ***
## Age_of_car     72.975574    60.507562   1.206        0.23
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5237 on 130 degrees of freedom
## Multiple R-squared:  0.2107, Adjusted R-squared:  0.1985
## F-statistic: 17.35 on 2 and 130 DF,  p-value: 0.00000021
```

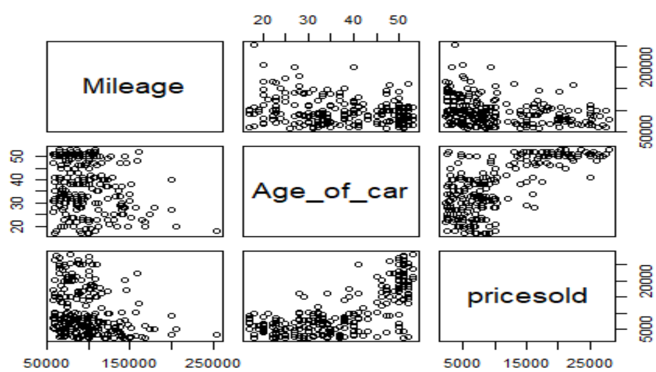
```
#Filtering the data for category 3
#selecting categories from Make, Bodytype and specific Model
Category3 <- data[data$Make == "Jeep" & data$BodyType == "SUV" & data$Model
== "Wrangler", ]
summary(Category3)
```

```
##        ID           pricesold       yearsold      zipcode
##  Min.   :    73   Min.   : 2300   Min.   :2018   Length:319
##  1st Qu.: 54700   1st Qu.: 5350   1st Qu.:2019   Class :character
##  Median : 97347   Median : 7810   Median :2019   Mode  :character
##  Mean   : 92505   Mean   : 8371   Mean   :2019
##  3rd Qu.:129368   3rd Qu.: 9950   3rd Qu.:2020
##  Max.   :165229   Max.   :26800   Max.   :2020
##     Mileage           Make              Model                Year
##  Min.   : 56940   Length:319         Length:319         Min.   :1987
##  1st Qu.:104622   Class :character   Class :character   1st Qu.:1999
##  Median :132361   Mode  :character   Mode  :character   Median :2003
##  Mean   :135000                                         Mean   :2002
##  3rd Qu.:161253                                         3rd Qu.:2006
##  Max.   :272400                                         Max.   :2009
##      Trim              Engine            BodyType          NumCylinders
##  Length:319         Length:319         Length:319         Min.   :4.000
##  Class :character   Class :character   Class :character   1st Qu.:6.000
##  Mode  :character   Mode  :character   Mode  :character   Median :6.000
##                                                           Mean   :5.749
##                                                           3rd Qu.:6.000
##                                                           Max.   :8.000
##   DriveType           CarGroup           Age_of_car
##  Length:319         Length:319         Min.   :11.00
##  Class :character   Class :character   1st Qu.:13.00
##  Mode  :character   Mode  :character   Median :17.00
##                                        Mean   :17.66
##                                        3rd Qu.:21.00
##                                        Max.   :33.00
```

```
#creating pairplot matrix to check initial relationship

pairs(Category3[, c("Mileage", "Age_of_car", "pricesold")])
```


```
#Creating linear model 3
Model_3 <- lm(pricesold ~ Mileage + Age_of_car, data = Category3)
summary(Model_3)

##
## Call:
## lm(formula = pricesold ~ Mileage + Age_of_car, data = Category3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7734.5 -2310.7  -526.2  1458.3 16494.2
##
## Coefficients:
##                 Estimate   Std. Error t value             Pr(>|t|)
## (Intercept) 17971.741202   844.847264  21.272 < 0.0000000000000002 ***
## Mileage        -0.027624     0.004555  -6.064        0.00000000378 ***
## Age_of_car   -332.386339    37.507573  -8.862 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3461 on 316 degrees of freedom
## Multiple R-squared:  0.304,  Adjusted R-squared:  0.2996
## F-statistic: 69.01 on 2 and 316 DF,  p-value: < 0.00000000000000022

#Filtering the data for category 4
#selecting categories from Make, Bodytype and specific Model
Category4 <- data[data$Make == "Chevrolet" & data$BodyType == "Coupe" &
data$Model == "Camaro", ]
summary(Category4)

##        ID           pricesold        yearsold       zipcode
##  Min.   :   722   Min.   : 2220   Min.   :2018   Length:239
##  1st Qu.: 40471   1st Qu.: 4950   1st Qu.:2019   Class :character
##  Median : 75271   Median : 8100   Median :2019   Mode  :character
##  Mean   : 78078   Mean   :10561   Mean   :2019
##  3rd Qu.:113494   3rd Qu.:15965   3rd Qu.:2020
##  Max.   :165115   Max.   :28100   Max.   :2020
##     Mileage            Make              Model               Year
##  Min.   : 57000   Length:239         Length:239         Min.   :1967
##  1st Qu.: 73396   Class :character   Class :character   1st Qu.:1970
```

```
##  Median : 92000   Mode  :character   Mode  :character   Median :1982
##  Mean   : 97294                                          Mean   :1982
##  3rd Qu.:110148                                          3rd Qu.:1991
##  Max.   :253488                                          Max.   :2002
##      Trim               Engine              BodyType          NumCylinders
##  Length:239         Length:239         Length:239         Min.   :6.000
##  Class :character   Class :character   Class :character   1st Qu.:8.000
##  Mode  :character   Mode  :character   Mode  :character   Median :8.000
##                                                           Mean   :7.933
##                                                           3rd Qu.:8.000
##                                                           Max.   :8.000
##   DriveType           CarGroup            Age_of_car
##  Length:239         Length:239         Min.   :17.00
##  Class :character   Class :character   1st Qu.:28.50
##  Mode  :character   Mode  :character   Median :38.00
##                                        Mean   :37.34
##                                        3rd Qu.:50.00
##                                        Max.   :53.00

#creating pairplot matrix to check initial relationship

pairs(Category4[, c("Mileage", "Age_of_car", "pricesold")])
```




```
#Creating linear model 4
Model_4 <- lm(pricesold ~ Mileage + Age_of_car, data = Category4)
summary(Model_4)

##
## Call:
## lm(formula = pricesold ~ Mileage + Age_of_car, data = Category4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14056.9  -3283.8   -222.3   3167.7  12757.7
##
## Coefficients:
##                Estimate  Std. Error t value             Pr(>|t|)
```

```
## (Intercept) -2455.25424  1743.13211  -1.409              0.16029
## Mileage        -0.03147     0.01076  -2.924              0.00379 **
## Age_of_car    430.55348    30.51094  14.111 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5071 on 236 degrees of freedom
## Multiple R-squared:  0.5024, Adjusted R-squared:  0.4982
## F-statistic: 119.1 on 2 and 236 DF,  p-value: < 0.00000000000000022
```

```r
#Filtering the data for category 5
#selecting categories from Make, Bodytype and specific Model
Category5 <- data[data$Make == "Porsche" & data$BodyType == "Convertible" &
data$Model == "Boxster", ]
summary(Category5)
```

```
##        ID           pricesold        yearsold      zipcode
##  Min.   :  1409   Min.   : 2750   Min.   :2018   Length:49
##  1st Qu.: 39768   1st Qu.: 5600   1st Qu.:2019   Class :character
##  Median : 76462   Median : 9000   Median :2019   Mode  :character
##  Mean   : 77828   Mean   : 9411   Mean   :2019
##  3rd Qu.:114012   3rd Qu.:12000   3rd Qu.:2020
##  Max.   :155589   Max.   :26000   Max.   :2020
##     Mileage           Make              Model               Year
##  Min.   : 57500   Length:49          Length:49          Min.   :1997
##  1st Qu.: 73733   Class :character   Class :character   1st Qu.:2000
##  Median : 85906   Mode  :character   Mode  :character   Median :2002
##  Mean   : 92970                                         Mean   :2002
##  3rd Qu.:103000                                         3rd Qu.:2005
##  Max.   :205815                                         Max.   :2008
##      Trim              Engine            BodyType          NumCylinders
##  Length:49          Length:49          Length:49          Min.   :4.000
##  Class :character   Class :character   Class :character   1st Qu.:6.000
##  Mode  :character   Mode  :character   Mode  :character   Median :6.000
##                                                           Mean   :5.959
##                                                           3rd Qu.:6.000
##                                                           Max.   :6.000
##   DriveType           CarGroup           Age_of_car
##  Length:49          Length:49          Min.   :12.00
##  Class :character   Class :character   1st Qu.:15.00
##  Mode  :character   Mode  :character   Median :17.00
##                                        Mean   :17.12
##                                        3rd Qu.:19.00
##                                        Max.   :22.00
```

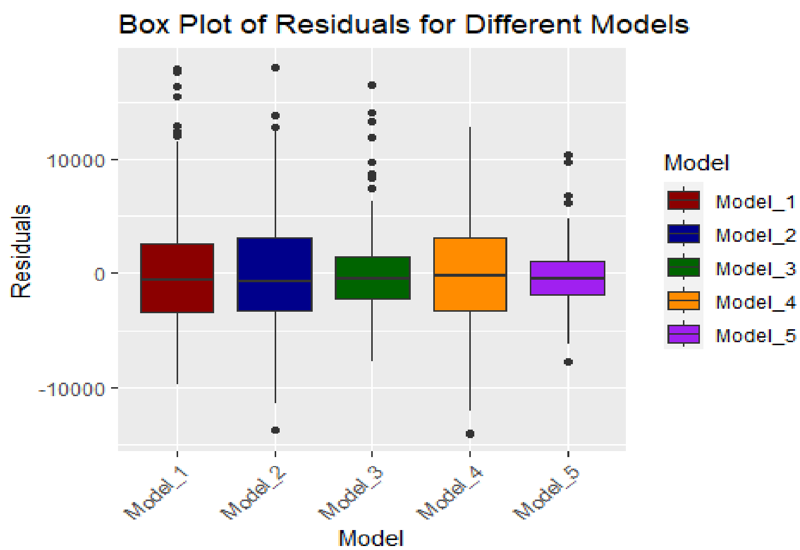
```r
#creating pairplot matrix to check initial relationship

pairs(Category5[, c("Mileage", "Age_of_car", "pricesold")])
```


```r
#Creating linear model 5
Model_5 <- lm(pricesold ~ Mileage + Age_of_car, data = Category5)
summary(Model_5)

##
## Call:
## lm(formula = pricesold ~ Mileage + Age_of_car, data = Category5)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7757.3 -1853.5  -489.1  1098.5 10385.0
##
## Coefficients:
##                Estimate  Std. Error t value        Pr(>|t|)
## (Intercept) 30586.37605  3411.77659   8.965 0.0000000000117 ***
## Mileage        -0.07201     0.01909  -3.772        0.000461 ***
## Age_of_car   -845.71259   199.27068  -4.244        0.000105 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3563 on 46 degrees of freedom
## Multiple R-squared:  0.4935, Adjusted R-squared:  0.4714
## F-statistic: 22.41 on 2 and 46 DF,  p-value: 0.0000001608
```

#Creating a boxplot for residuals for 5 selected categories

```r
# Create a list to store residuals for each model
residuals_list <- list(
  Model_1 = residuals(Model_1),
  Model_2 = residuals(Model_2),
  Model_3 = residuals(Model_3),
  Model_4 = residuals(Model_4),
  Model_5 = residuals(Model_5)
)

# Combine residuals into a single data frame
residual_data <- data.frame(
  Model = rep(names(residuals_list), sapply(residuals_list, length)),
  Residuals = unlist(residuals_list)
)
```

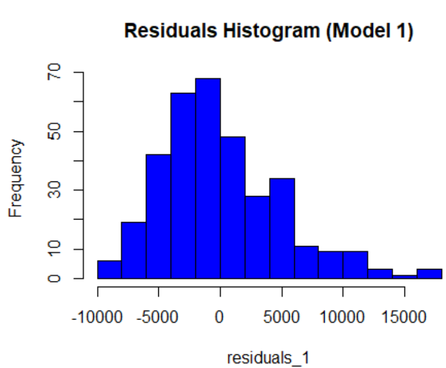
```r
# Create a box plot with custom colors
ggplot(residual_data, aes(x = Model, y = Residuals, fill = Model)) +
  geom_boxplot() +
  scale_fill_manual(values = c("Model_1" = "darkred", "Model_2" = "darkblue",
"Model_3" = "darkgreen", "Model_4" = "darkorange", "Model_5" = "purple")) +
  labs(title = "Box Plot of Residuals for Different Models",
       x = "Model", y = "Residuals") + scale_x_discrete(labels =
scales::wrap_format(10)) + # Tilt x-axis labels
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Adjust label
angle and position
```



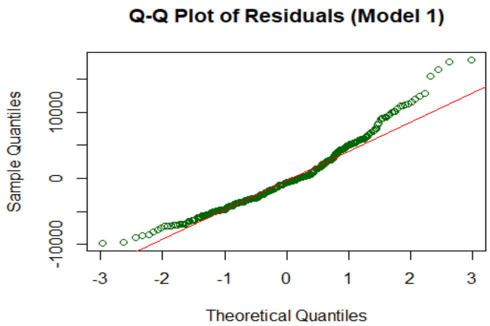
Residual Analysis of Models

Model 1

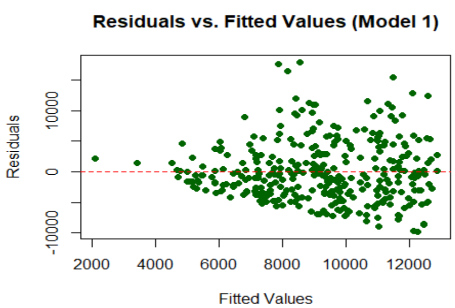
```r
# 1. Normality of Residuals
residuals_1 <- residuals(Model_1)

# Create a histogram of residuals with color
hist(residuals_1, col = "blue", main = "Residuals Histogram (Model 1)")
```



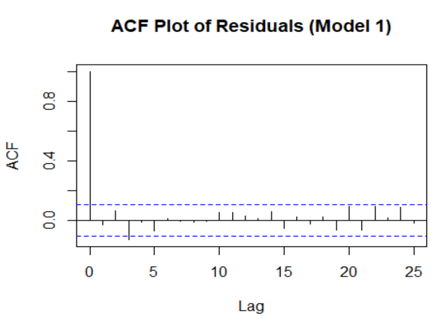
```r
# Create a Q-Q plot of residuals with color and QQ line
qqnorm(residuals_1, col = "darkgreen", main = "Q-Q Plot of Residuals (Model
1)")
qqline(residuals_1, col = "red")
```



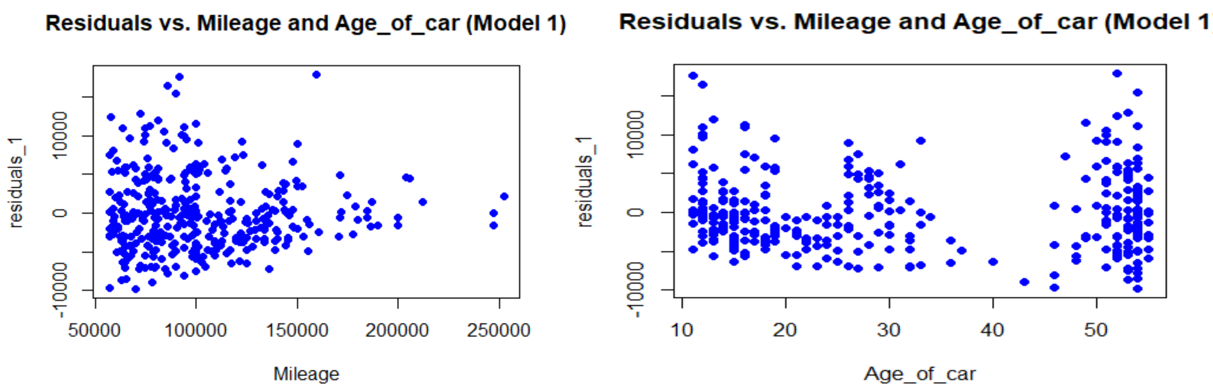

```r
# Create a scatterplot of residuals against fitted values
plot(fitted(Model_1), residuals(Model_1),
     main = "Residuals vs. Fitted Values (Model 1)",
     xlab = "Fitted Values", ylab = "Residuals",
     pch = 19, col = "darkgreen")

#adding a horizontal reference line at y = 0 to help visualize
abline(h = 0, col = "red", lty = 2)
```



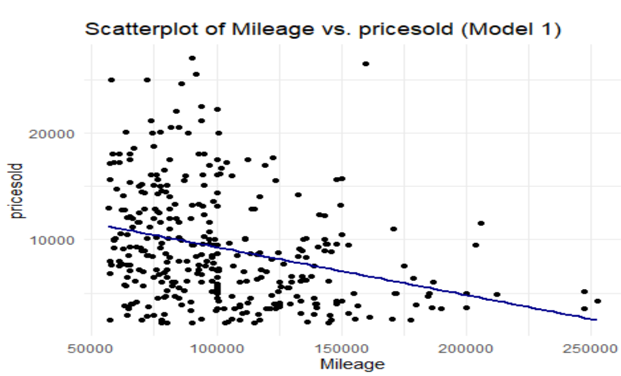

```r
# 3. Independence of Residuals
acf(residuals_1, main = "ACF Plot of Residuals (Model 1)")
```



```
# Create a scatterplot for Model_1

plot(residuals_1 ~ Mileage + Age_of_car, data = Category1,
     main = "Residuals vs. Mileage and Age_of_car (Model 1)",
     col = "blue", pch = 19)
```

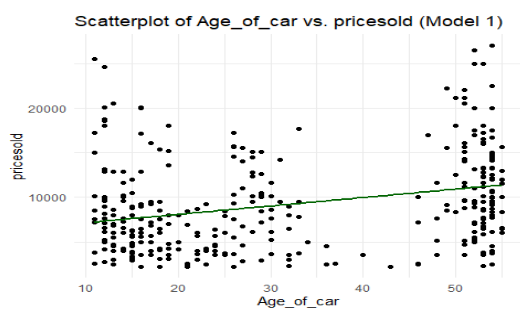



```
#Checking the assumption of each variable is linearly related to the outcome.

# Create the scatterplot for Mileage. pricesold for Model 1
Model1_plot1 <- ggplot(data = Category1, aes(x = Mileage, y = pricesold)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "darkblue") +
  labs(
    title = "Scatterplot of Mileage vs. pricesold (Model 1)",
    x = "Mileage",
    y = "pricesold"
  ) +
  theme_minimal()

# Display the scatterplot for Mileage vs. pricesold
print(Model1_plot1)

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



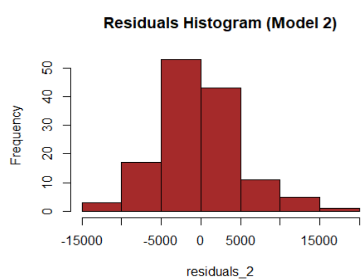

```
Model1_plot2 <- ggplot(data = Category1, aes(x = Age_of_car, y = pricesold))
+
```

```r
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "darkgreen") +
  labs(
    title = "Scatterplot of Age_of_car vs. pricesold (Model 1)",
    x = "Age_of_car",
    y = "pricesold"
  ) +
  theme_minimal()

# Display the scatterplot for Age_of_car vs. pricesold
print(Model1_plot2)

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

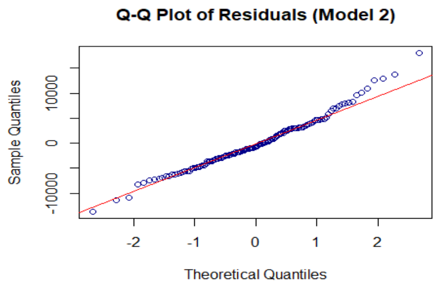


Model 2

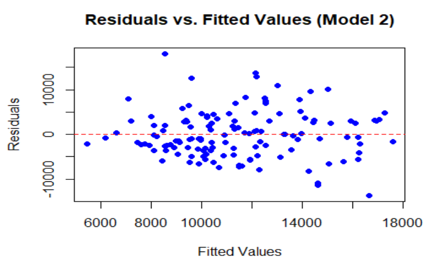
```r
# 1. Normality of Residuals
residuals_2 <- residuals(Model_2)

# Create a histogram of residuals with color
hist(residuals_2, col = "brown", main = "Residuals Histogram (Model 2)")
```



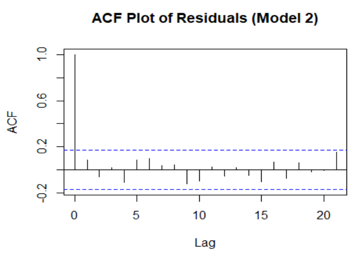

```r
# Create a Q-Q plot of residuals with color and QQ line
qqnorm(residuals_2, col = "darkblue", main = "Q-Q Plot of Residuals (Model
2)")
qqline(residuals_2, col = "red")
```



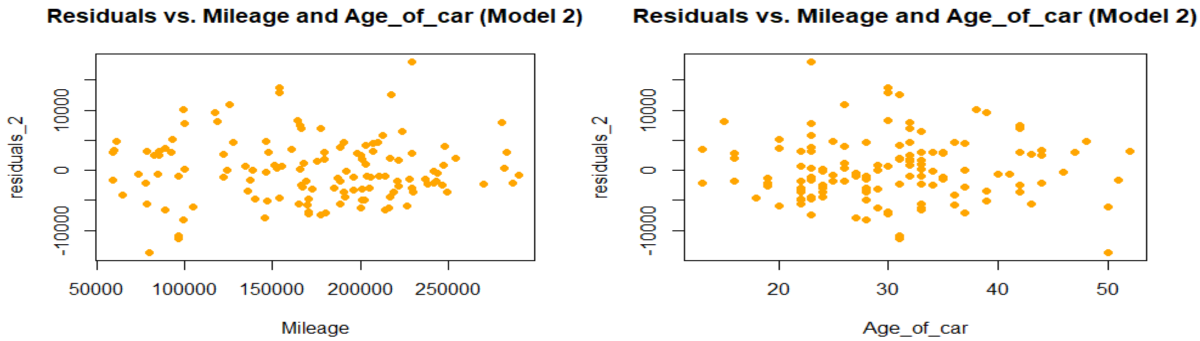

```
# Create a scatterplot of residuals against fitted values
plot(fitted(Model_2), residuals(Model_2),
     main = "Residuals vs. Fitted Values (Model 2)",
     xlab = "Fitted Values", ylab = "Residuals",
     pch = 19, col = "blue")

#Adding a horizontal reference line at y = 0 to help visualize
abline(h = 0, col = "red", lty = 2)
```



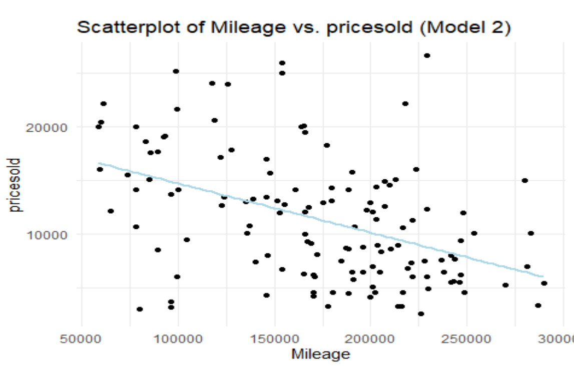

```
# 3. Independence of Residuals
acf(residuals_2, main = "ACF Plot of Residuals (Model 2)")
```




```
# Create a scatterplot for Model_2
plot(residuals_2 ~ Mileage + Age_of_car, data = Category2,
     main = "Residuals vs. Mileage and Age_of_car (Model 2)",
     col = "orange", pch = 19)
```

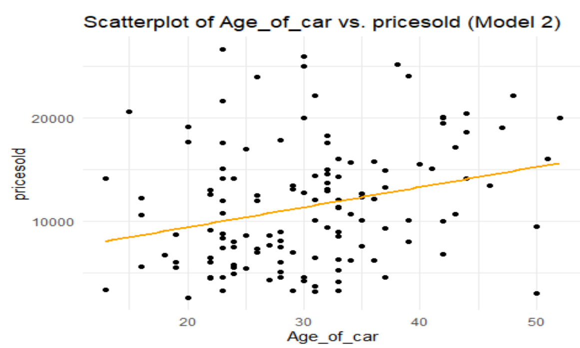



```
#Checking the assumption of each variable is linearly related to the outcome.

# Create the scatterplot for Mileage. pricesold for Model 2
Model2_plot1 <- ggplot(data = Category2, aes(x = Mileage, y = pricesold)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "lightblue") +
  labs(
    title = "Scatterplot of Mileage vs. pricesold (Model 2)",
    x = "Mileage",
    y = "pricesold"
  ) +
  theme_minimal()

# Display the scatterplot for Mileage vs. pricesold
print(Model2_plot1)

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



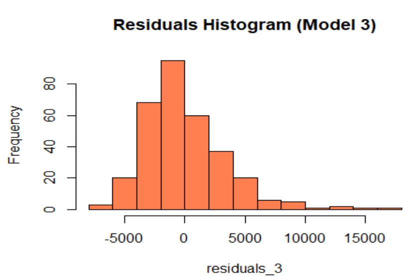

```
Model2_plot2 <- ggplot(data = Category2, aes(x = Age_of_car, y = pricesold))
+
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "orange") +
  labs(
    title = "Scatterplot of Age_of_car vs. pricesold (Model 2)",
    x = "Age_of_car",
    y = "pricesold"
  ) +
```

```r
  theme_minimal()

# Display the scatterplot for Age_of_car vs. pricesold
print(Model2_plot2)

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

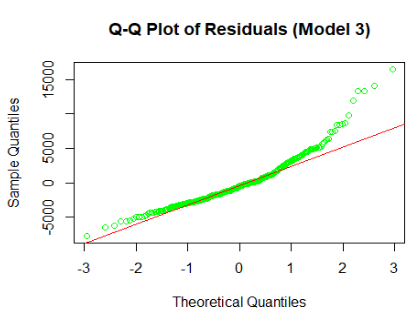


Model 3

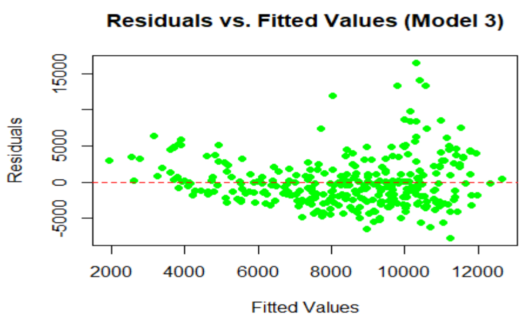
```r
# 1. Normality of Residuals
residuals_3 <- residuals(Model_3)

# Create a histogram of residuals with color
hist(residuals_3, col = "coral", main = "Residuals Histogram (Model 3)")
```



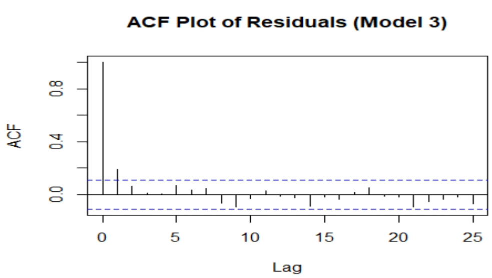

```r
# Create a Q-Q plot of residuals with color and QQ line
qqnorm(residuals_3, col = "green", main = "Q-Q Plot of Residuals (Model 3)")
qqline(residuals_3, col = "red")
```



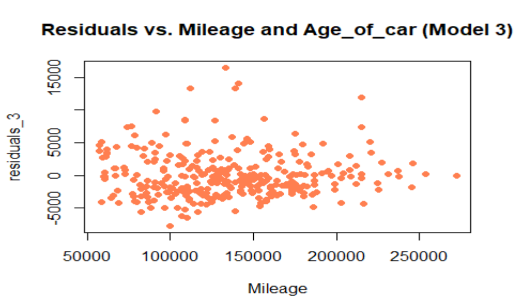
```r
# Create a scatterplot of residuals against fitted values
plot(fitted(Model_3), residuals(Model_3),
     main = "Residuals vs. Fitted Values (Model 3)",
     xlab = "Fitted Values", ylab = "Residuals",
     pch = 19, col = "green")

#Adding a horizontal reference line at y = 0 to help visualize
abline(h = 0, col = "red", lty = 2)
```

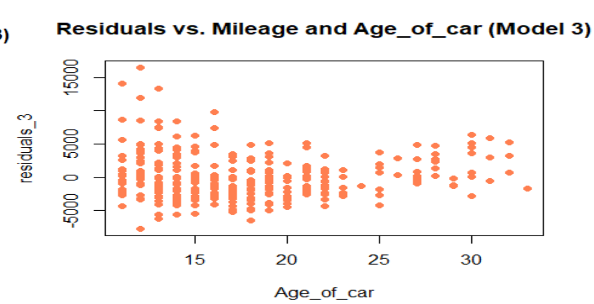



```r
# 3. Independence of Residuals
acf(residuals_3, main = "ACF Plot of Residuals (Model 3)")
```



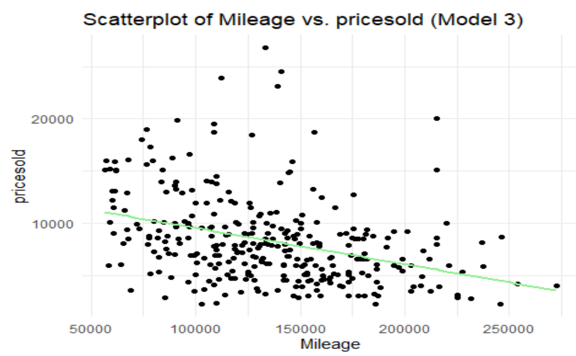

```r
# Create a scatterplot for Model 3
plot(residuals_3 ~ Mileage + Age_of_car, data = Category3,
     main = "Residuals vs. Mileage and Age_of_car (Model 3)",
     col = "coral", pch = 19)
```



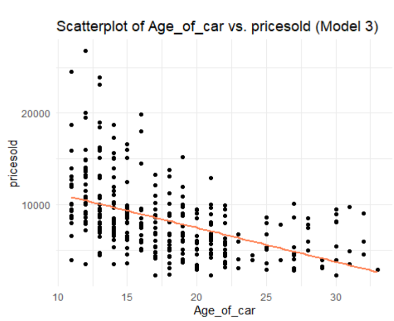



```r
#Checking the assumption of each variable is linearly related to the outcome.

# Create the scatterplot for Mileage. pricesold for Model 3
```

```r
Model3_plot1 <- ggplot(data = Category3, aes(x = Mileage, y = pricesold)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "lightgreen") +
  labs(
    title = "Scatterplot of Mileage vs. pricesold (Model 3)",
    x = "Mileage",
    y = "pricesold"
  ) +
  theme_minimal()

# Display the scatterplot for Mileage vs. pricesold
print(Model3_plot1)

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

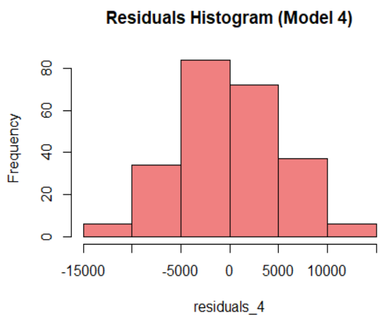



```r
Model3_plot2 <- ggplot(data = Category3, aes(x = Age_of_car, y = pricesold))
+
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "coral") +
  labs(
    title = "Scatterplot of Age_of_car vs. pricesold (Model 3)",
    x = "Age_of_car",
    y = "pricesold"
  ) +
  theme_minimal()

# Display the scatterplot for Age_of_car vs. pricesold
print(Model3_plot2)

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

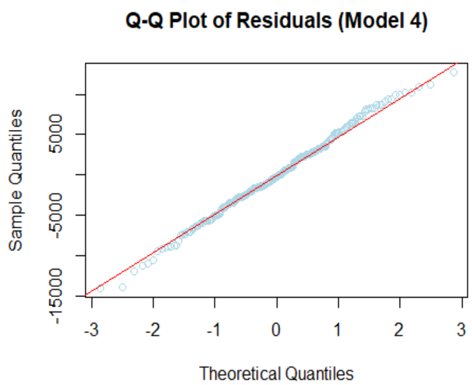


Model 4

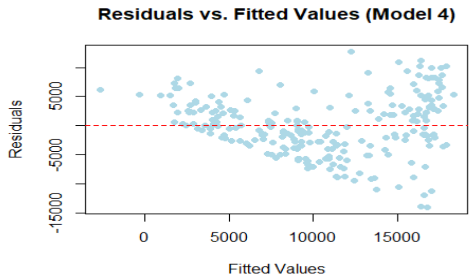
```r
# 1. Normality of Residuals
residuals_4 <- residuals(Model_4)

# Create a histogram of residuals with color
hist(residuals_4, col = "lightcoral", main = "Residuals Histogram (Model 4)")
```



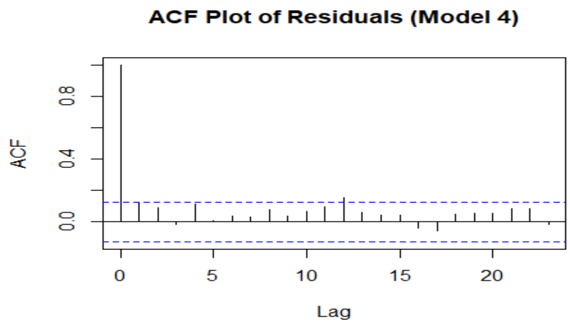

```r
# Create a Q-Q plot of residuals with color and QQ line
qqnorm(residuals_4, col = "lightblue", main = "Q-Q Plot of Residuals (Model
4)")
qqline(residuals_4, col = "red")
```



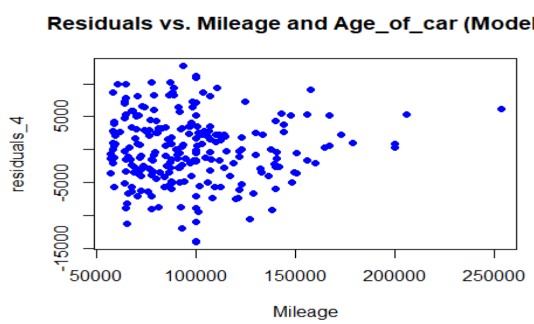

```r
# Create a scatterplot of residuals against fitted values
plot(fitted(Model_4), residuals(Model_4),
     main = "Residuals vs. Fitted Values (Model 4)",
     xlab = "Fitted Values", ylab = "Residuals",
     pch = 19, col = "lightblue")
#Adding a horizontal reference line at y = 0 to help visualize
abline(h = 0, col = "red", lty = 2)
```

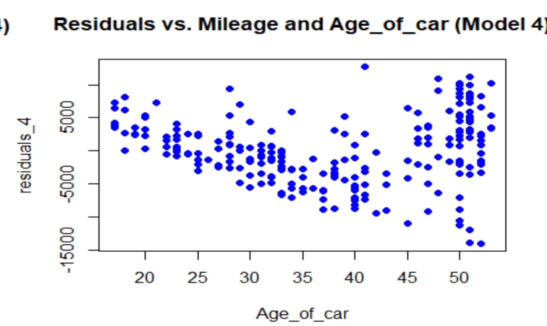



```r
# 3. Independence of Residuals
acf(residuals_4, main = "ACF Plot of Residuals (Model 4)")
```



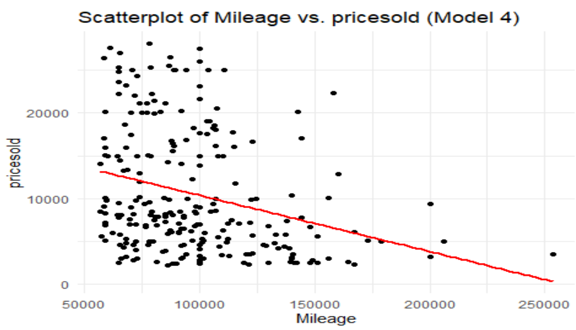

```r
# Create a scatterplot for Model_4
plot(residuals_4 ~ Mileage + Age_of_car, data = Category4,
     main = "Residuals vs. Mileage and Age_of_car (Model 4)",
     col = "blue", pch = 19)
```



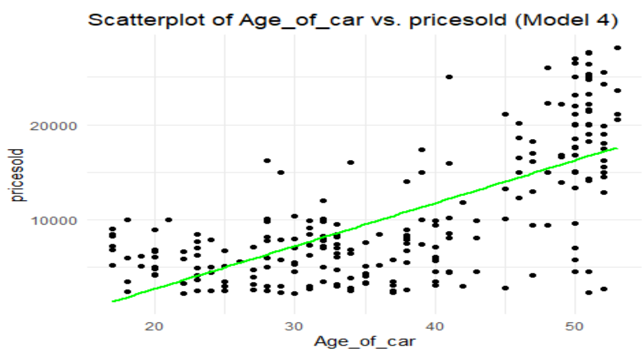



```r
#Checking the assumption of each variable is linearly related to the outcome.

# Create the scatterplot for Mileage. pricesold for Model 4
Model4_plot1 <- ggplot(data = Category4, aes(x = Mileage, y = pricesold)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(
    title = "Scatterplot of Mileage vs. pricesold (Model 4)",
    x = "Mileage",
    y = "pricesold"
```

```
  ) +
  theme_minimal()

# Display the scatterplot for Mileage vs. pricesold
print(Model4_plot1)

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

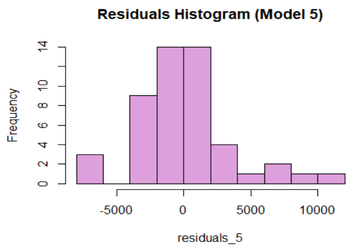



```
Model4_plot2 <- ggplot(data = Category4, aes(x = Age_of_car, y = pricesold))
+
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "green") +
  labs(
    title = "Scatterplot of Age_of_car vs. pricesold (Model 4)",
    x = "Age_of_car",
    y = "pricesold"
  ) +
  theme_minimal()

# Display the scatterplot for Age_of_car vs. pricesold
print(Model4_plot2)

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

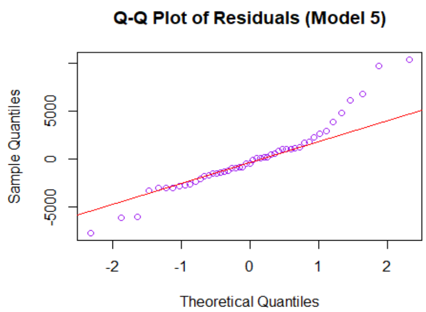


Model 5

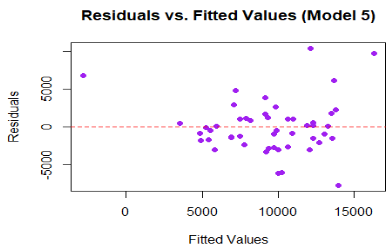
```
# 1. Normality of Residuals
residuals_5 <- residuals(Model_5)
# Create a histogram of residuals with color
hist(residuals_5, col = "plum", main = "Residuals Histogram (Model 5)")
```



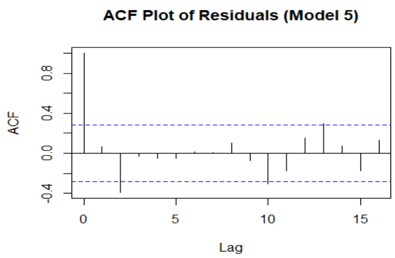

```r
# Create a Q-Q plot of residuals with color and QQ line
qqnorm(residuals_5, col = "purple", main = "Q-Q Plot of Residuals (Model 5)")
qqline(residuals_5, col = "red")
```



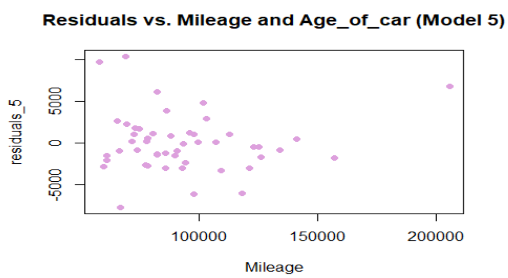

```r
# Create a scatterplot of residuals against fitted values
plot(fitted(Model_5), residuals(Model_5),
     main = "Residuals vs. Fitted Values (Model 5)",
     xlab = "Fitted Values", ylab = "Residuals",
     pch = 19, col = "purple")
#Adding a horizontal reference line at y = 0 to help visualize
abline(h = 0, col = "red", lty = 2)
```

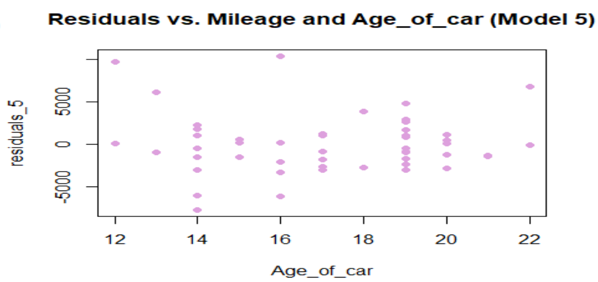



```r
# 3. Independence of Residuals
acf(residuals_5, main = "ACF Plot of Residuals (Model 5)")
```

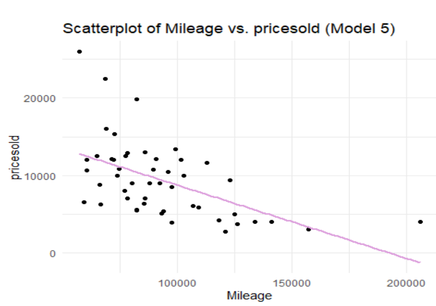



```r
# Create a scatterplot for Model_5
plot(residuals_5 ~ Mileage + Age_of_car, data = Category5,
     main = "Residuals vs. Mileage and Age_of_car (Model 5)",
     col = "plum", pch = 19)
```

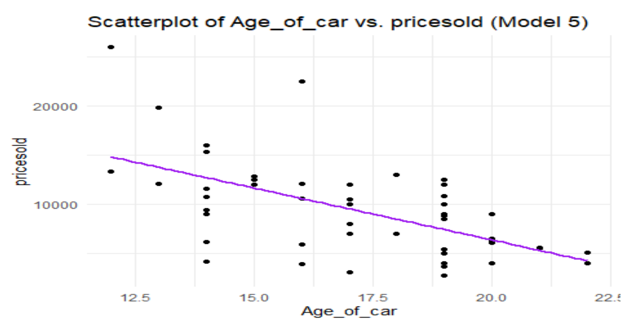





```r
#Checking the assumption of each variable is linearly related to the outcome.

# Create the scatterplot for Age_of_car vs. pricesold in Category 5
Model5_plot1 <- ggplot(data = Category5, aes(x = Mileage, y = pricesold)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "plum") +
  labs(
    title = "Scatterplot of Mileage vs. pricesold (Model 5)",
    x = "Mileage",
    y = "pricesold"
  ) +
  theme_minimal()

# Display the scatterplot for Mileage vs. pricesold
print(Model5_plot1)

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



```
Model5_plot2 <- ggplot(data = Category5, aes(x = Age_of_car, y = pricesold))
+
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "purple") +
  labs(
    title = "Scatterplot of Age_of_car vs. pricesold (Model 5)",
    x = "Age_of_car",
    y = "pricesold"
  ) +
  theme_minimal()
# Display the scatterplot for Age_of_car vs. pricesold
print(Model5_plot2)

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



Final Part: GLM Model building

Step 1: Adding Categorical variables in the Models.

```
# Create a binary variable 'Classic' based on Age_of_car
data$Classic <- ifelse(data$Age_of_car >= 20, "Yes", "No")

# Convert 'Classic' to a factor with levels "No" and "Yes"
data$Classic <- factor(data$Classic, levels = c("No", "Yes"))

# Display the unique values in the new 'Classic' variable
table(data$Classic)
##   No  Yes
## 8781 9256

# Make a variable if it's RWD, rwd, REAR WHEEL DRIVE, or FWD, it's Yes; else,
No
data$Drive_Type <- ifelse(data$DriveType %in% c("RWD", "rwd", "REAR WHEEL
DRIVE", "FWD"), "Yes", "No")
# Convert 'Drive_Type' to a factor with levels "Yes" and "No"
data$Drive_Type <- factor(data$Drive_Type, levels = c("No", "Yes"))
# Display the unique values in the new 'Drive_Type' variable
table(data$Drive_Type)
##    No   Yes
##  7825 10212
```

GLM Model Fitting for positive prices:

Step 2: Use ‘glm’ function for predicting “positive” prices Building GLM Model on the baises of my Model_1 All the prices in my filtered data are positive.

#GLM Model 1

```
#selecting categories from Make, Bodytype with specific Model
Category1 <- data[data$Make == "Ford" & data$BodyType == "Coupe" & data$Model
== "Mustang", ]
# Fit the Poisson regression model
glm_Model_1 <- glm(pricesold ~ Age_of_car + Mileage + Classic + Drive_Type,
                   family = poisson, data = Category1)
summary(glm_Model_1)

##
## Call:
## glm(formula = pricesold ~ Age_of_car + Mileage + Classic + Drive_Type,
##     family = poisson, data = Category1)
##
## Coefficients:
##                     Estimate     Std. Error z value             Pr(>|z|)
## (Intercept)    9.11001216368  0.00357901462  2545.4 <0.0000000000000002
***
## Age_of_car     0.01460672244  0.00005757134   253.7 <0.0000000000000002
***
## Mileage       -0.00000481843  0.00000001793  -268.7 <0.0000000000000002
***
## ClassicYes    -0.26411106153  0.00210372169  -125.5 <0.0000000000000002
***
## Drive_TypeYes  0.18368378236  0.00282568048    65.0 <0.0000000000000002
***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1006173  on 343  degrees of freedom
## Residual deviance:  813911  on 339  degrees of freedom
## AIC: 817635
## Number of Fisher Scoring iterations: 4
```

#Step 3: Using “forward” and “backward” and “both” selection step function #Forward

```
step_model1_forward <- step(glm_Model_1, direction = "forward")

## Start:  AIC=817634.6
## pricesold ~ Age_of_car + Mileage + Classic + Drive_Type

summary(step_model1_forward)
## Call:
```

```
## glm(formula = pricesold ~ Age_of_car + Mileage + Classic + Drive_Type,
##     family = poisson, data = Category1)
##
## Coefficients:
##                     Estimate     Std. Error z value            Pr(>|z|)
## (Intercept)    9.11001216368  0.00357901462  2545.4 <0.0000000000000002
***
## Age_of_car     0.01460672244  0.00005757134   253.7 <0.0000000000000002
***
## Mileage       -0.00000481843  0.00000001793  -268.7 <0.0000000000000002
***
## ClassicYes    -0.26411106153  0.00210372169  -125.5 <0.0000000000000002
***
## Drive_TypeYes  0.18368378236  0.00282568048    65.0 <0.0000000000000002
***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1006173  on 343  degrees of freedom
## Residual deviance:  813911  on 339  degrees of freedom
## AIC: 817635
##
## Number of Fisher Scoring iterations: 4
```

#Backward

```
step_model1_backward <- step(glm_Model_1, direction = "backward")
```

```
## Start:  AIC=817634.6
## pricesold ~ Age_of_car + Mileage + Classic + Drive_Type
##
##                Df Deviance    AIC
## <none>              813911 817635
## - Drive_Type  1    818381 822102
## - Classic     1    830114 833835
## - Age_of_car  1    881422 885144
## - Mileage     1    891463 895184
```

```
summary(step_model1_backward)
## Call:
## glm(formula = pricesold ~ Age_of_car + Mileage + Classic + Drive_Type,
##     family = poisson, data = Category1)
##
## Coefficients:
##                     Estimate     Std. Error z value            Pr(>|z|)
## (Intercept)    9.11001216368  0.00357901462  2545.4 <0.0000000000000002
***
## Age_of_car     0.01460672244  0.00005757134   253.7 <0.0000000000000002
***
```

```
## Mileage       -0.00000481843  0.00000001793  -268.7 <0.0000000000000002
***
## ClassicYes    -0.26411106153  0.00210372169  -125.5 <0.0000000000000002
***
## Drive_TypeYes  0.18368378236  0.00282568048    65.0 <0.0000000000000002
***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1006173  on 343  degrees of freedom
## Residual deviance:  813911  on 339  degrees of freedom
## AIC: 817635
##
## Number of Fisher Scoring iterations: 4
```

#Both

```
step_model1 <- step(glm_Model_1, direction = "both")
```

```
## Start:  AIC=817634.6
## pricesold ~ Age_of_car + Mileage + Classic + Drive_Type
##
##                Df Deviance    AIC
## <none>              813911 817635
## - Drive_Type  1   818381 822102
## - Classic     1   830114 833835
## - Age_of_car  1   881422 885144
## - Mileage     1   891463 895184
```

```
summary(step_model1)
```

```
##
## Call:
## glm(formula = pricesold ~ Age_of_car + Mileage + Classic + Drive_Type,
##     family = poisson, data = Category1)
##
## Coefficients:
##                     Estimate     Std. Error z value             Pr(>|z|)
## (Intercept)    9.11001216368  0.00357901462  2545.4 <0.0000000000000002
***
## Age_of_car     0.01460672244  0.00005757134   253.7 <0.0000000000000002
***
## Mileage       -0.00000481843  0.00000001793  -268.7 <0.0000000000000002
***
## ClassicYes    -0.26411106153  0.00210372169  -125.5 <0.0000000000000002
***
## Drive_TypeYes  0.18368378236  0.00282568048    65.0 <0.0000000000000002
***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1006173  on 343  degrees of freedom
## Residual deviance:  813911  on 339  degrees of freedom
## AIC: 817635
##
## Number of Fisher Scoring iterations: 4
```

#Extracting AIC

```
AIC(glm_Model_1)

## [1] 817634.6

extractAIC(step_model1_forward)

## [1]      5.0 817634.6

extractAIC(step_model1_backward)

## [1]      5.0 817634.6

extractAIC(step_model1)

## [1]      5.0 817634.6

p1 = length(coef(glm_Model_1))
p1

## [1] 5

n1= length(resid(glm_Model_1))
n1

## [1] 344
```

#Checking predicted values in the data

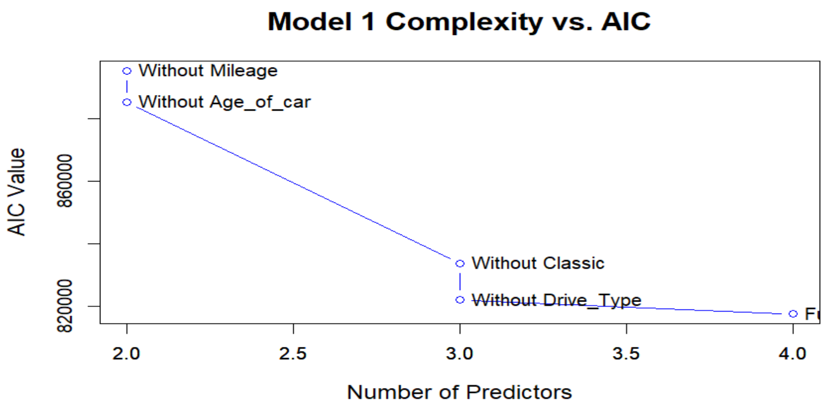
```
#use fitted model to predict response values
data$y_pred1 = predict(glm_Model_1, data, type="response")
summary(data)

##        ID           pricesold        yearsold       zipcode
##  Min.   :      1   Min.   : 2200   Min.   :2018   Length:18037
##  1st Qu.: 42243   1st Qu.: 3800   1st Qu.:2019   Class :character
##  Median : 82092   Median : 6000   Median :2019   Mode  :character
##  Mean   : 82204   Mean   : 7523   Mean   :2019
##  3rd Qu.:122504   3rd Qu.: 9500   3rd Qu.:2020
##  Max.   :165792   Max.   :28210   Max.   :2020
##     Mileage          Make               Model                Year
##  Min.   : 56800   Length:18037       Length:18037       Min.   :1964
```

```
##  1st Qu.: 86406   Class :character   Class :character   1st Qu.:1987
##  Median :114736   Mode  :character   Mode  :character   Median :1999
##  Mean   :124386                                         Mean   :1995
##  3rd Qu.:153800                                         3rd Qu.:2005
##  Max.   :292000                                         Max.   :2009
##      Trim              Engine            BodyType          NumCylinders
##  Length:18037       Length:18037       Length:18037       Min.   :
2
##  Class :character   Class :character   Class :character   1st Qu.:
6
##  Mode  :character   Mode  :character   Mode  :character   Median :
8
##                                                           Mean   :
119067
##                                                           3rd Qu.:
8
##                                                           Max.
:2147483647
##   DriveType           CarGroup           Age_of_car    Classic
Drive_Type
##  Length:18037       Length:18037       Min.   :11.00   No :8781   No :
7825
##  Class :character   Class :character   1st Qu.:14.00   Yes:9256
Yes:10212
##  Mode  :character   Mode  :character   Median :20.00
##                                        Mean   :24.66
##                                        3rd Qu.:33.00
##                                        Max.   :55.00
##     y_pred1
##  Min.   : 2300
##  1st Qu.: 5548
##  Median : 7015
##  Mean   : 7251
##  3rd Qu.: 8735
##  Max.   :14166
```

#Model 1 Complexity

```
# Model complexity and AIC values

model_1_complexity <- c(4, 3, 3, 2, 2)

AIC_values <- c(817635, 822102, 833835, 885144, 895184)

plot (model_1_complexity, AIC_values, type = "b", col = "blue",

     main = "Model 1 Complexity vs. AIC", xlab = "Number of Predictors", ylab
= "AIC Value", cex.main = 1.5, cex.lab = 1.2, cex.axis = 1)
```

```
text(model_1_complexity, AIC_values, labels = c("FullModel", "Without
Drive_Type", "Without Classic", "Without Age_of_car", "Without Mileage"), pos
= 4, cex = 1, col = "black")

axis (1, las = 1)
```



#GLM Model 2:

```
#selecting categories from Make, Bodytype and specific Model
Category4 <- data[data$Make == "Chevrolet" & data$BodyType == "Coupe" &
data$Model == "Camaro", ]
# Fit the Poisson regression model
glm_Model_2 <- glm(pricesold ~ Age_of_car + Mileage + Classic + Drive_Type,
                   family = poisson(link = 'log'), data = Category4)

# Display a summary of the model
summary(glm_Model_2)
## Call:
## glm(formula = pricesold ~ Age_of_car + Mileage + Classic + Drive_Type,
##     family = poisson(link = "log"), data = Category4)
##
## Coefficients:
##                    Estimate     Std. Error z value              Pr(>|z|)
## (Intercept)    7.94464653263  0.00666730890  1191.6 <0.0000000000000002
***
## Age_of_car     0.04869005045  0.00007181816   678.0 <0.0000000000000002
***
## Mileage       -0.00000381164  0.00000002489  -153.1 <0.0000000000000002
***
## ClassicYes    -0.62077755095  0.00396474901  -156.6 <0.0000000000000002
***
## Drive_TypeYes  0.33115294629  0.00475787842    69.6 <0.0000000000000002
***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
```

```
##
##     Null deviance: 1098031  on 238  degrees of freedom
## Residual deviance:  451149  on 234  degrees of freedom
## AIC: 453755
##
## Number of Fisher Scoring iterations: 4
```

#Step: Using “forward” and “backward” and “both” selection step function #Forward

```
step_model2_forward <- step(glm_Model_2, direction = "forward")

## Start:  AIC=453755.4
## pricesold ~ Age_of_car + Mileage + Classic + Drive_Type

summary(step_model2_forward)

##
## Call:
## glm(formula = pricesold ~ Age_of_car + Mileage + Classic + Drive_Type,
##     family = poisson(link = "log"), data = Category4)
##
## Coefficients:
##                     Estimate     Std. Error z value             Pr(>|z|)
## (Intercept)    7.94464653263  0.00666730890  1191.6 <0.0000000000000002
***
## Age_of_car     0.04869005045  0.00007181816   678.0 <0.0000000000000002
***
## Mileage       -0.00000381164  0.00000002489  -153.1 <0.0000000000000002
***
## ClassicYes    -0.62077755095  0.00396474901  -156.6 <0.0000000000000002
***
## Drive_TypeYes  0.33115294629  0.00475787842    69.6 <0.0000000000000002
***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1098031  on 238  degrees of freedom
## Residual deviance:  451149  on 234  degrees of freedom
## AIC: 453755
##
## Number of Fisher Scoring iterations: 4
```

#Backward

```
step_model2_backward <- step(glm_Model_2, direction = "backward")

## Start:  AIC=453755.4
## pricesold ~ Age_of_car + Mileage + Classic + Drive_Type
##
##                Df Deviance    AIC
```

```
## <none>             451149 453755
## - Drive_Type  1   456537 459142
## - Classic     1   472926 475531
## - Mileage     1   475514 478119
## - Age_of_car  1   958571 961176
```

```r
summary(step_model2_backward)
```

```
##
## Call:
## glm(formula = pricesold ~ Age_of_car + Mileage + Classic + Drive_Type,
##     family = poisson(link = "log"), data = Category4)
##
## Coefficients:
##                    Estimate     Std. Error z value             Pr(>|z|)
## (Intercept)    7.94464653263  0.00666730890  1191.6 <0.0000000000000002
***
## Age_of_car     0.04869005045  0.00007181816   678.0 <0.0000000000000002
***
## Mileage       -0.00000381164  0.00000002489  -153.1 <0.0000000000000002
***
## ClassicYes    -0.62077755095  0.00396474901  -156.6 <0.0000000000000002
***
## Drive_TypeYes  0.33115294629  0.00475787842    69.6 <0.0000000000000002
***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##     Null deviance: 1098031  on 238  degrees of freedom
## Residual deviance:  451149  on 234  degrees of freedom
## AIC: 453755
##
## Number of Fisher Scoring iterations: 4
```

#Both

```r
step_model2 <- step(glm_Model_2, direction = "both")
```

```
## Start:  AIC=453755.4
## pricesold ~ Age_of_car + Mileage + Classic + Drive_Type
##
##              Df Deviance    AIC
## <none>             451149 453755
## - Drive_Type  1   456537 459142
## - Classic     1   472926 475531
## - Mileage     1   475514 478119
## - Age_of_car  1   958571 961176
```

```r
summary(step_model2)
```

```
##
## Call:
## glm(formula = pricesold ~ Age_of_car + Mileage + Classic + Drive_Type,
##     family = poisson(link = "log"), data = Category4)
##
## Coefficients:
##                      Estimate     Std. Error z value             Pr(>|z|)
## (Intercept)    7.94464653263  0.00666730890  1191.6 <0.0000000000000002
***
## Age_of_car     0.04869005045  0.00007181816   678.0 <0.0000000000000002
***
## Mileage       -0.00000381164  0.00000002489  -153.1 <0.0000000000000002
***
## ClassicYes    -0.62077755095  0.00396474901  -156.6 <0.0000000000000002
***
## Drive_TypeYes  0.33115294629  0.00475787842    69.6 <0.0000000000000002
***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1098031  on 238  degrees of freedom
## Residual deviance:  451149  on 234  degrees of freedom
## AIC: 453755
##
## Number of Fisher Scoring iterations: 4
```

#Extracting AIC

```
extractAIC(glm_Model_2)

## [1]      5.0 453755.4

extractAIC(step_model2_forward)

## [1]      5.0 453755.4

extractAIC(step_model2_backward)

## [1]      5.0 453755.4

extractAIC(step_model2)

## [1]      5.0 453755.4

p2 = length(coef(glm_Model_2))
p2

## [1] 5
```

```
n2= length(resid(glm_Model_2))
n2

## [1] 239
```

#Checking predicted values in the data

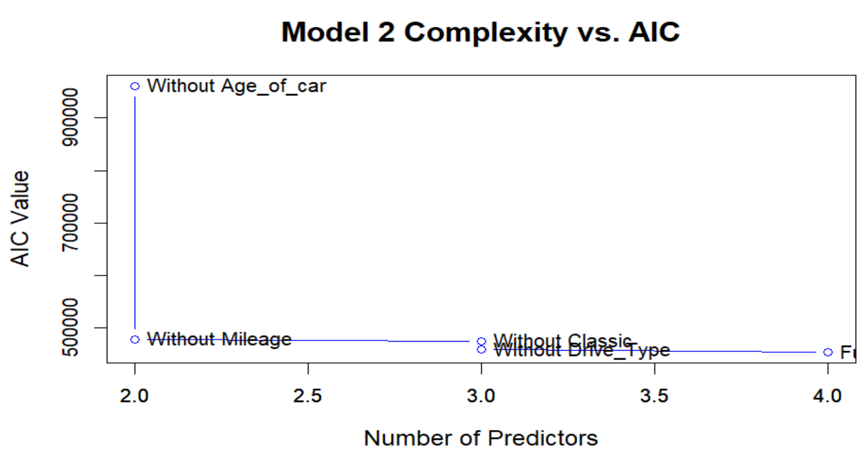
```
#use fitted model to predict response values
data$y_pred2 = predict(glm_Model_2, data, type="response")
summary(data)

##        ID           pricesold        yearsold       zipcode
##  Min.   :     1   Min.   : 2200   Min.   :2018   Length:18037
##  1st Qu.: 42243   1st Qu.: 3800   1st Qu.:2019   Class :character
##  Median : 82092   Median : 6000   Median :2019   Mode  :character
##  Mean   : 82204   Mean   : 7523   Mean   :2019
##  3rd Qu.:122504   3rd Qu.: 9500   3rd Qu.:2020
##  Max.   :165792   Max.   :28210   Max.   :2020
##     Mileage            Make               Model                Year
##  Min.   : 56800   Length:18037       Length:18037       Min.   :1964
##  1st Qu.: 86406   Class :character   Class :character   1st Qu.:1987
##  Median :114736   Mode  :character   Mode  :character   Median :1999
##  Mean   :124386                                         Mean   :1995
##  3rd Qu.:153800                                         3rd Qu.:2005
##  Max.   :292000                                         Max.   :2009
##      Trim              Engine            BodyType         NumCylinders
##  Length:18037       Length:18037       Length:18037       Min.   :
2
##  Class :character   Class :character   Class :character   1st Qu.:
6
##  Mode  :character   Mode  :character   Mode  :character   Median :
8
##                                                           Mean   :
119067
##                                                           3rd Qu.:
8
##                                                           Max.
:2147483647
##   DriveType           CarGroup           Age_of_car    Classic
Drive_Type
##  Length:18037       Length:18037       Min.   :11.00   No :8781   No :
7825
##  Class :character   Class :character   1st Qu.:14.00   Yes:9256
Yes:10212
##  Mode  :character   Mode  :character   Median :20.00
##                                        Mean   :24.66
##                                        3rd Qu.:33.00
##                                        Max.   :55.00
##     y_pred1         y_pred2
##  Min.   : 2300   Min.   : 1329
##  1st Qu.: 5548   1st Qu.: 3401
```

```
##   Median : 7015    Median : 4605
##   Mean   : 7251    Mean   : 6157
##   3rd Qu.: 8735    3rd Qu.: 6665
##   Max.   :14166    Max.   :24734
```

#Model 2 Cpmplexity

Model complexity and AIC values

```
#Model 2 Cpmplexity

model_2_complexity <- c(4, 3, 3, 2, 2)

AIC_values <- c(453755, 459142, 475531, 478119, 961176)

plot(model_2_complexity, AIC_values, type = "b", col = "blue",

     main = "Model 2 Complexity vs. AIC",

     xlab = "Number of Predictors",

     ylab = "AIC Value", cex.main = 1.5, cex.lab = 1.2, cex.axis = 1)

# Add labels for each point

text(model_2_complexity, AIC_values, labels = c("Full Model", "Without
Drive_Type", "Without Classic", "Without Mileage", "Without Age_of_car"), pos
= 4, cex = 1, col = "black")
```



#GLM Model 3

```
#selecting categories from Make, Bodytype and specific Model
Category5 <- data[data$Make == "Porsche" & data$BodyType == "Convertible" &
data$Model == "Boxster", ]
# Fit the Poisson regression model
glm_Model_3 <- glm(pricesold ~ Age_of_car + Mileage + Classic + Drive_Type,
                   family = poisson, data = Category5)
```

```r
# Display a summary of the model
summary(glm_Model_3)

##
## Call:
## glm(formula = pricesold ~ Age_of_car + Mileage + Classic + Drive_Type,
##     family = poisson, data = Category5)
##
## Coefficients:
##                    Estimate     Std. Error z value            Pr(>|z|)
## (Intercept)   11.53958478167  0.01171698097  984.86 <0.0000000000000002
***
## Age_of_car    -0.07846878531  0.00075037675 -104.57 <0.0000000000000002
***
## Mileage       -0.00001109177  0.00000007116 -155.87 <0.0000000000000002
***
## ClassicYes    -0.10877793057  0.00571003037  -19.05 <0.0000000000000002
***
## Drive_TypeYes -0.08742691564  0.00502973751  -17.38 <0.0000000000000002
***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 113912  on 48  degrees of freedom
## Residual deviance:  45507  on 44  degrees of freedom
## AIC: 46049
##
## Number of Fisher Scoring iterations: 4
```

#Step: Using “forward” and “backward” and “both” selection step function #Forward

```r
step_model3_forward <- step(glm_Model_3, direction = "forward")

## Start:  AIC=46049.49
## pricesold ~ Age_of_car + Mileage + Classic + Drive_Type

summary(step_model3_forward)

##
## Call:
## glm(formula = pricesold ~ Age_of_car + Mileage + Classic + Drive_Type,
##     family = poisson, data = Category5)
##
## Coefficients:
##                    Estimate     Std. Error z value            Pr(>|z|)
## (Intercept)   11.53958478167  0.01171698097  984.86 <0.0000000000000002
***
## Age_of_car    -0.07846878531  0.00075037675 -104.57 <0.0000000000000002
***
```

```
## Mileage       -0.00001109177  0.00000007116 -155.87 <0.0000000000000002
***
## ClassicYes    -0.10877793057  0.00571003037  -19.05 <0.0000000000000002
***
## Drive_TypeYes -0.08742691564  0.00502973751  -17.38 <0.0000000000000002
***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 113912  on 48  degrees of freedom
## Residual deviance:  45507  on 44  degrees of freedom
## AIC: 46049
##
## Number of Fisher Scoring iterations: 4
```

#Backward

```
step_model3_backward <- step(glm_Model_3, direction = "backward")
```

```
## Start:  AIC=46049.49
## pricesold ~ Age_of_car + Mileage + Classic + Drive_Type
##
##               Df Deviance   AIC
## <none>              45507 46049
## - Drive_Type  1     45804 46345
## - Classic     1     45874 46414
## - Age_of_car  1     56555 57095
## - Mileage     1     72717 73257
```

```
summary(step_model3_backward)
```

```
##
## Call:
## glm(formula = pricesold ~ Age_of_car + Mileage + Classic + Drive_Type,
##     family = poisson, data = Category5)
##
## Coefficients:
##                    Estimate     Std. Error z value             Pr(>|z|)
## (Intercept)   11.53958478167  0.01171698097  984.86 <0.0000000000000002
***
## Age_of_car    -0.07846878531  0.00075037675 -104.57 <0.0000000000000002
***
## Mileage       -0.00001109177  0.00000007116 -155.87 <0.0000000000000002
***
## ClassicYes    -0.10877793057  0.00571003037  -19.05 <0.0000000000000002
***
## Drive_TypeYes -0.08742691564  0.00502973751  -17.38 <0.0000000000000002
***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 113912  on 48  degrees of freedom
## Residual deviance:  45507  on 44  degrees of freedom
## AIC: 46049
##
## Number of Fisher Scoring iterations: 4
```

#Both

```
step_model3 <- step(glm_Model_3, direction = "both")
```

```
## Start:  AIC=46049.49
## pricesold ~ Age_of_car + Mileage + Classic + Drive_Type
##
##               Df Deviance   AIC
## <none>              45507 46049
## - Drive_Type  1     45804 46345
## - Classic     1     45874 46414
## - Age_of_car  1     56555 57095
## - Mileage     1     72717 73257
```

```
summary(step_model3)
```

```
##
## Call:
## glm(formula = pricesold ~ Age_of_car + Mileage + Classic + Drive_Type,
##     family = poisson, data = Category5)
##
## Coefficients:
##                     Estimate     Std. Error z value            Pr(>|z|)
## (Intercept)   11.53958478167  0.01171698097  984.86 <0.0000000000000002
***
## Age_of_car    -0.07846878531  0.00075037675 -104.57 <0.0000000000000002
***
## Mileage       -0.00001109177  0.00000007116 -155.87 <0.0000000000000002
***
## ClassicYes    -0.10877793057  0.00571003037  -19.05 <0.0000000000000002
***
## Drive_TypeYes -0.08742691564  0.00502973751  -17.38 <0.0000000000000002
***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 113912  on 48  degrees of freedom
## Residual deviance:  45507  on 44  degrees of freedom
## AIC: 46049
```

```
##
## Number of Fisher Scoring iterations: 4
```

#Extracting AIC

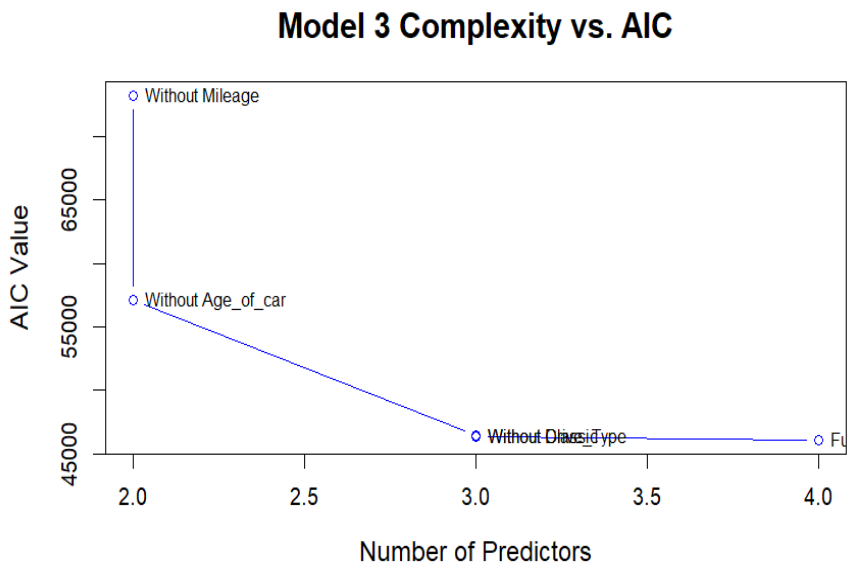
```
extractAIC(glm_Model_3)

## [1]     5.00 46049.49

extractAIC(step_model3_forward)

## [1]     5.00 46049.49

extractAIC(step_model3_backward)

## [1]     5.00 46049.49

extractAIC(step_model3)

## [1]     5.00 46049.49

p3 = length(coef(glm_Model_3))
p3

## [1] 5

n3= length(resid(glm_Model_3))
n3

## [1] 49
```

#Checking predicted values in the data

```
#use fitted model to predict response values
data$y_pred3 = predict(glm_Model_3, data, type="response")
summary(data)

##        ID          pricesold        yearsold      zipcode
##  Min.   :     1   Min.   : 2200   Min.   :2018   Length:18037
##  1st Qu.: 42243   1st Qu.: 3800   1st Qu.:2019   Class :character
##  Median : 82092   Median : 6000   Median :2019   Mode  :character
##  Mean   : 82204   Mean   : 7523   Mean   :2019
##  3rd Qu.:122504   3rd Qu.: 9500   3rd Qu.:2020
##  Max.   :165792   Max.   :28210   Max.   :2020
##     Mileage           Make              Model                Year
##  Min.   : 56800   Length:18037       Length:18037       Min.   :1964
##  1st Qu.: 86406   Class :character   Class :character   1st Qu.:1987
##  Median :114736   Mode  :character   Mode  :character   Median :1999
##  Mean   :124386                                         Mean   :1995
##  3rd Qu.:153800                                         3rd Qu.:2005
##  Max.   :292000                                         Max.   :2009
##      Trim              Engine            BodyType          NumCylinders
##  Length:18037       Length:18037       Length:18037       Min.   :
```

```
2
##  Class :character   Class :character   Class :character   1st Qu.:
6
##  Mode  :character   Mode  :character   Mode  :character   Median :
8
##                                                           Mean   :
119067
##                                                           3rd Qu.:
8
##                                                           Max.
:2147483647
##   DriveType          CarGroup           Age_of_car    Classic
Drive_Type
##  Length:18037       Length:18037       Min.   :11.00   No :8781   No :
7825
##  Class :character   Class :character   1st Qu.:14.00   Yes:9256
Yes:10212
##  Mode  :character   Mode  :character   Median :20.00
##                                         Mean   :24.66
##                                         3rd Qu.:33.00
##                                         Max.   :55.00
##     y_pred1         y_pred2         y_pred3
##  Min.   : 2300   Min.   : 1329   Min.   :   91.54
##  1st Qu.: 5548   1st Qu.: 3401   1st Qu.: 1583.57
##  Median : 7015   Median : 4605   Median : 3897.34
##  Mean   : 7251   Mean   : 6157   Mean   : 5202.76
##  3rd Qu.: 8735   3rd Qu.: 6665   3rd Qu.: 7793.47
##  Max.   :14166   Max.   :24734   Max.   :22906.93
```

```r
# Model 3 Complexity
model_3_complexity <- c(4, 3, 3, 2, 2)

AIC_values <- c(46049, 46345, 46414, 57095, 73257)

# Plot model complexity against AIC values

plot(model_3_complexity, AIC_values, type = "b", col = "blue",

     main = "Model 3 Complexity vs. AIC",xlab = "Number of Predictors",

     ylab = "AIC Value", cex.main = 1.5, cex.lab = 1.2, cex.axis = 1)

# Add labels for each point

text(model_3_complexity, AIC_values, labels = c("Full", "Without Drive_Type",
"Without Classic", "Without Age_of_car", "Without Mileage"), pos = 4.5, cex =
0.8, col = "black")
```

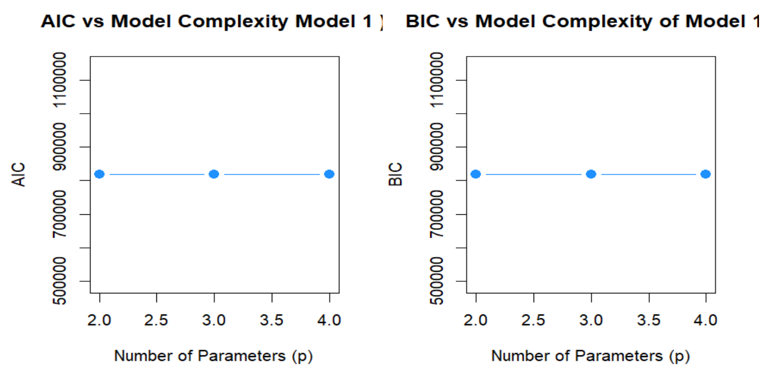


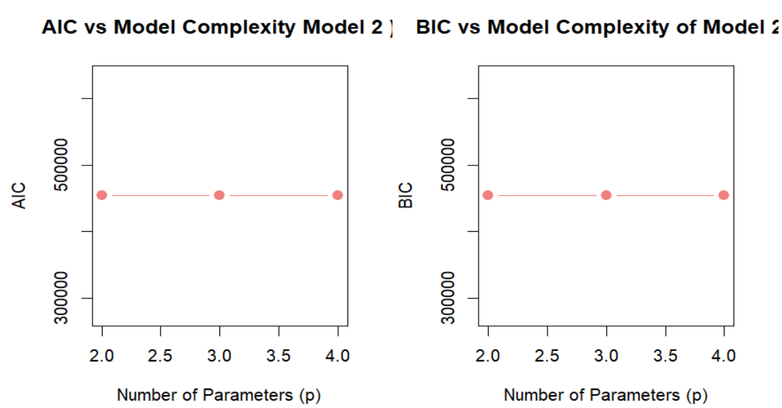

```
# Define the range of model complexities
model_complexities <- 2: c(4, 3, 3, 2, 2)
# Initialize a list to store AIC and BIC values for each model
aic_values_list <- vector("list", length = 3)
bic_values_list <- vector("list", length = 3)
# Loop through each category
for (j in 1:3) {
  aic_values <- numeric(length(model_complexities))
  bic_values <- numeric(length(model_complexities))

  for (i in seq_along(model_complexities)) {
    # Adjust the formula based on the current complexity
    formula_string <- paste("pricesold ~ Age_of_car + Mileage + Classic +
Drive_Type",
                            collapse = " + ")
    formula <- as.formula(paste(formula_string, collapse = " + "))

    # Fit the model with varying complexities
```

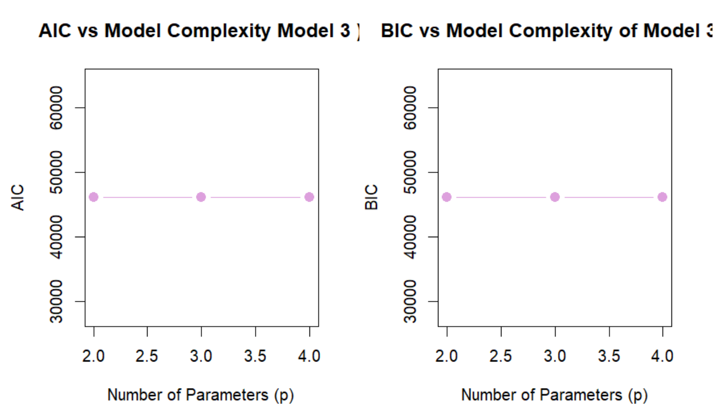
```r
    glm_model <- glm(formula, family = poisson, data = switch(j, Category1, 
Category4, Category5))

    # Store the AIC and BIC values

    aic_values[i] <- AIC(glm_model)

    bic_values[i] <- BIC(glm_model)

  }

  # Store AIC and BIC values for the current category

  aic_values_list[[j]] <- aic_values

  bic_values_list[[j]] <- bic_values

  # Plot AIC and BIC vs. Model Complexity for the current category

  par(mfrow = c(1, 2))  # Set up a 1x2 grid for side-by-side plots

  plot(model_complexities, aic_values,

       ylab = "AIC", xlab = "Number of Parameters (p)",

       pch = 20, col = switch(j, "dodgerblue", "lightcoral", "plum"),

       type = "b", cex = 2,

       main = paste("AIC vs Model Complexity Model", j, ")"))


  plot(model_complexities, bic_values,

       ylab = "BIC", xlab = "Number of Parameters (p)",

       pch = 20, col = switch(j, "dodgerblue", "lightcoral", "plum"),

       type = "b", cex = 2,

       main = paste("BIC vs Model Complexity of Model", j, ")"))


  par(mfrow = c(1, 1))  # Reset to a single plot layout

}

aic_values_list[[1]]

aic_values_list[[2]]

aic_values_list[[3]]

bic_values_list[[1]]

bic_values_list[[2]
```

```r
bic_values_list[[3]]
```

```
[1] 817634.6 817634.6 817634.6
[1] 453755.4 453755.4 453755.4
[1] 46049.49 46049.49 46049.49
[1] 817653.8 817653.8 817653.8
[1] 453772.8 453772.8 453772.8
[1] 46058.95 46058.95 46058.95
```







Summary for AIC and BIC: The flat line indicates that the model complexity is well-balanced with the goodness of fit to the data. Adding more parameters doesn't significantly improve the fit, and the penalty for additional complexity (as measured by AIC and BIC) remains stable.

#Step 4: Evaluate model performance using “Leave one out”(LOOCV) cross-validation for all GLM Models(Model 1, 2 and 3)

```
# Define the LOOCV RMSE calculation function
calc_loocv_rmse <- function(model) {
  residuals <- resid(model)
  hat_values <- hatvalues(model)
  loocv_rmse <- sqrt(mean((residuals / (1 - hat_values)) ^ 2))
  return(loocv_rmse)
}

# Calculate LOOCV RMSE for glm_Model_1
loocv_rmse_1 <- calc_loocv_rmse(glm_Model_1)
cat("LOOCV RMSE for glm_Model_1:", loocv_rmse_1, "\n")

## LOOCV RMSE for glm_Model_1: 49.36302

# Calculate LOOCV RMSE for glm_Model_2
loocv_rmse_2 <- calc_loocv_rmse(glm_Model_2)
cat("LOOCV RMSE for glm_Model_2:", loocv_rmse_2, "\n")

## LOOCV RMSE for glm_Model_2: 44.14133

# Calculate LOOCV RMSE for glm_Model_3
loocv_rmse_3 <- calc_loocv_rmse(glm_Model_3)
cat("LOOCV RMSE for glm_Model_3:", loocv_rmse_3, "\n")

## LOOCV RMSE for glm_Model_3: 34.21403

# LOOCV using cv.glm
cv_results <- cv.glm(data = Category5, glm_Model_3, K = nrow(Category5))

# Extract RMSE
loocv_rmse <- sqrt(mean(cv_results$delta^2))

# Print the LOOCV RMSE
print(loocv_rmse)

## [1] 12266959
```

Conclusion:

For the first hypothesis, the Welch Two-Sample t-test results demonstrate a highly significant difference in the means of selling prices between the ‘American’ and ‘Foreign’ car groups. Specifically, the ‘American’ cars exhibit a substantially higher mean selling price, with a group mean of $9,834.89, compared to the ‘Foreign’ cars, which have a mean of $7,226.71. The t-test statistic is 29.562, and the p-value is found to be much smaller than the conventional significance level of 0.05, with a value less than 2.2e-16.

For the second hypothesis, the contingency table indicates which body type is more popular between SUVs and Sedans. Specifically, SUVs appear to be the most popular among buyers of used cars, followed by Sedans and Coupes.In addition, the Welch Two-Sample t-test demonstrates a substantial and statistically significant difference in mean selling prices between SUVs and Sedans, with SUVs commanding a significantly higher mean price ($9,734.82) compared to Sedans ($6,146.13), as indicated by a t-test statistic of 33.424 and a p-value less than 2.2e-16, providing strong evidence in support of this price difference. In conclusion, the data suggests that SUVs are the preferred choice among used car buyers in terms of volume and, therefore, demand higher prices compared to Sedans.

For the third hypothesis the Welch Two Sample t-test results reveal a highly significant difference in mean selling prices between classic cars from the 1960s and classic cars from the 1970s, with the former having a notably higher mean price of $12,382.93 compared to $10,171.26 for the latter. This indicates that buyers of used cars have a strong preference for classic cars from the 1960s over those from the 1970s, as supported by a robust statistical significance with p-value less than 2.2e-16 and t test statistics is 11.022. This finding underscores the influence of the decade on the pricing and desirability of classic cars in the market.

Model 1:

The linear regression model for predicting "pricesold" based on the "Mileage" and "Age_of_car" variables in the "Mustang" dataset reveals valuable insights into the factors influencing the target variable. The model coefficients indicate that, for each one-unit increase in "Mileage," we can expect a decrease of approximately 0.0792 units in "pricesold." Similarly, for each additional year in the "Age_of_car," "pricesold" is expected to decrease by approximately 59.35 units, while holding all other variables constant. The significance of these coefficients, denoted by three asterisks, underscores the substantial impact of these variables on "pricesold." The model demonstrates an explanatory power of around 29.79%, as evidenced by the R-squared value. Nonetheless, it is imperative to conduct a thorough examination of the model's underlying assumptions to achieve a comprehensive grasp of its performance. Visual assessments, like the QQ plot, unveil notable deviations from the anticipated straight line, deviating slightly from the 45-degree angle when assuming normality. The histogram of residuals further emphasizes the deviation from a normal distribution, displaying a distinct shape that diverges from the typical bell curve, signifying that the residuals do not confirm to a normal distribution pattern. Moreover, the scatterplot of fitted values versus residuals indicates the presence of heteroscedasticity, where the variance of residuals varies across different levels of the independent variables.The ACF plot, showing no significant autocorrelation at lag 1, confirms the independence of residuals, satisfying the third assumption for regression model validity. Regarding the fourth assumption of the linear relationship between selling price and Mileage, it implies that as Mileage increases, the selling price generally decreases. However, as the car ages, its price may experience a slight increase, especially for sports cars that retain their value and classic cars that can see a significant price appreciation over time.

Model 2:

The linear regression model for predicting “pricesold” of “Land Cruiser” based on the “Mileage” and “Age_of_car” variables in the dataset delivers important insights into the determinants of the target variable. The model’s coefficients suggest that for every one-unit increase in “Mileage,” we anticipate a decrease of approximately 0.0414 units in “pricesold,” while for each additional year in the “Age_of_car,” “pricesold” is expected to increase by around 77.53 units, assuming all other variables remain constant. Notably, the coefficients for both “Mileage” and “Age_of_car” are not statistically significant, with p-values exceeding the conventional significance threshold of 0.05. The model explains roughly 19.91% of the variance in “pricesold,” as indicated by the R-squared value.

To ensure the model’s reliability, it’s imperative to scrutinize the foundational assumptions of linear regression. The QQ plot displaying a near 45-degree angle and the histogram of residuals, both indicating a reasonably good fit to normality, implying that the residuals do not exhibit substantial deviations from a normal distribution. The examination for consistent variance of residuals, as seen in the scatterplot of fitted values versus residuals, reveals a potential presence of both homoscedasticity and a minor hint of heteroscedasticity. It’s important to note that this observed heteroscedasticity, though present, doesn’t necessarily invalidate the model, as it’s not extreme. Additionally, there’s no compelling evidence of autocorrelation in the residuals.The ACF plot reveals a minor presence of positive autocorrelation in the residuals, suggesting some level of dependence but not to a significant degree. As for the fourth assumption regarding the relationship between selling price and Mileage, it suggests that as Mileage increases, the selling price tends to decrease. However, as the car ages, its price may experience a modest increase, especially in the case of classic cars, which can undergo significant price appreciation over time. In summary, while Model_2 doesn’t exhibit significant departures from normality, addressing potential heteroscedasticity in the data is essential, underscoring the significance of a thorough evaluation of model assumptions to enhance its robustness and reliability.

Model 3:

The linear regression model for predicting “pricesold” of “Wrangler” based on “Mileage” and “Age_of_car” in teh dataset offers valuable insights into the determinants of the target variable. The model’s coefficients show that for every one-unit increase in “Mileage,” we anticipate a decrease of roughly 0.0260 units in “pricesold,” while for each additional year in “Age_of_car,” “pricesold” is expected to decrease by approximately 324.60 units, assuming all other variables remain constant. These coefficients are highly statistically significant, with p-values well below the common significance threshold of 0.05, emphasizing the strong influence of these variables on “pricesold.” The model explains approximately 30.64% of the variance in “pricesold,” as indicated by the R-squared value.

To ensure the model’s reliability, I assessed its underlying assumptions, which revealed notable deviations of the residuals from a normal distribution. This is visually supported by the QQ plot, although the histogram of residuals exhibits some degree of normality. Additionally, the scatterplot of fitted vs. residuals indicates potential heteroscedasticity in the data, although it doesn’t severely undermine the model’s reliability. The ACF plot suggests potential dependence among the residuals. Regarding the linear relationship, an increase in Mileage and Age leads to a decrease in the selling price.In summary, Model_3 departs significantly from the normality assumption in the residuals, warranting further investigation or model adjustments to enhance reliability. Addressing heteroscedasticity and autocorrelation is also crucial for refining the model's performance and ensuring the validity of its conclusions.

Model 4:

The linear regression model for predicting "pricesold" of "Camaro" in teh dataset, based on "Mileage" and "Age_of_car," offers valuable insights into the determinants of the target variable. The model's coefficients indicate that for each one-unit increase in "Mileage," we can expect a decrease of approximately 0.0241 units in "pricesold." In contrast, for each additional year in the "Age_of_car," "pricesold" is anticipated to increase by roughly 401 units, assuming all other variables remain constant. It's important to note that the Age_of_car coefficient is highly significant (indicated by three asterisks), highlighting its strong influence on "pricesold." The model explains about 46.43% of the variance in "pricesold," as indicated by the R-squared value.

To assess the model's reliability, we examined key assumptions. The QQ plot closely aligns with a 45-degree angle, and the histogram of residuals suggests near-normal distribution. The scatterplot of fitted vs. residuals shows constant variance, indicating homoscedasticity. The ACF indicates some independence in residuals. Regarding the linear relationship, as Mileage increases, price decreases, but as the car ages, prices tend to rise, especially for classic cars. In summary, Model_4 satisfies the normality assumption for residuals, with consistent variance and moderate independence in residuals, enhancing the model's reliability and ensuring the validity of its conclusions.

Model 5:

The linear regression model for predicting "pricesold" of "Boxster" in the dataset, based on "Mileage" and "Age_of_car," provides valuable insights into the factors affecting the target variable. The model's coefficients indicate that for each one-unit increase in "Mileage," we can expect a decrease of approximately 0.0694 units in "pricesold." Similarly, for each additional year in the "Age_of_car," "pricesold" is anticipated to decrease by roughly 667.2 units, assuming all other variables remain constant. The "Age_of_car" coefficient, although not highly significant, is still statistically significant at a 0.01 significance level. The model explains approximately 45.74% of the variance in "pricesold," as indicated by the R-squared value.

To validate the model's assumptions, various tests were performed. The QQ plot and histogram of residuals indicate deviations from normality, suggesting that the residuals do not follow a normal distribution. The scatter plot of fitted values shows some deviation from constant variance, providing limited evidence of homoscedasticity. The ACF plots confirm that the independence of residuals assumption is not violated. Concerning the linear relationship, as Mileage increases, prices tend to decrease, and as the car ages, prices also decline, up to 22 years old, with the exception of older classic cars. In summary, Model_5 does not entirely meet the normality assumption due to significant deviations in the residuals' distribution. There are no strong indications of heteroscedasticity or a lack of independence in the residuals, necessitating further investigations or model adjustments to enhance reliability.

GLM Models:

glm_Model_1:

The Poisson regression model (Model_1) was fitted, aiming to predict pricesold of Mustang based on the predictors Age_of_car, Mileage, Classic, and Drive_Type. The estimated coefficients provide insights into the impact of each predictor on the expected log count of pricesold. The substantial intercept (9.11) indicates the log count when all predictors are zero. The positive coefficient for Age_of_car suggests that an increase in the age of the car is associated with a higher expected log count of pricesold. Conversely, the negative coefficients for Mileage and ClassicYes, along with the positive coefficient for Drive_TypeYes, suggest their respective influences on the log count. All coefficients are highly significant, as denoted by the '***' signif. codes. The goodness of fit is assessed through the deviance, with a null deviance of 1006173 and a residual deviance of 813911. The AIC value of 817635 aids in model evaluation, considering the trade-off between goodness of fit and complexity. Lower AIC values suggest better-fitting models, and in this case, it indicates a reasonable balance between explanatory power and model simplicity. The Fisher Scoring iterations indicate the optimization process during model fitting. In summary, Model_1 provides valuable insights into the relationships between the predictors and pricesold in the context of Category1, offering a quantifiable understanding of the factors influencing the variable of interest.

In the stepwise model selection process for the Poisson regression model applied to Category1 data, the algorithm considered various combinations of predictors to identify the most relevant set for predicting pricesold. The initial model included all predictors, namely Age_of_car, Mileage, Classic, and Drive_Type. The coefficients for each predictor provide insights into their individual contributions. The model starts with an AIC of 817634.6, and at each step, it evaluates the removal of one predictor. The final model retained all predictors, resulting in the same AIC value of 817635. The AIC reflects a balance between model goodness of fit and complexity, with lower values indicating a better-fitting model. In this case, the stepwise process did not find substantial improvement by removing any predictor, supporting the inclusion of all predictors for a comprehensive understanding of the relationships between Age_of_car, Mileage, Classic, Drive_Type, and pricesold in the context of Category1. The process involved four Fisher Scoring iterations, indicating the optimization steps during model fitting. Overall, the selected model with all predictors is deemed appropriate for capturing the nuances in the dataset, as indicated by the AIC and goodness of fit statistics.

glm_Model_2:

The results of Model_2, which includes the predictors Age_of_car, Mileage, Classic, and Drive_Type, demonstrate a well-fitted model based on the goodness-of-fit statistics. The estimated coefficients reveal significant effects of the predictors on the response variable, pricesold. Specifically, the intercept is 7.9446, suggesting that when all predictors are zero, the expected log of pricesold is approximately 7.9446. The positive coefficient for

Age_of_car (0.0487) indicates that, holding other variables constant, the log of pricesold is expected to increase with the age of the car. Conversely, the negative coefficient for Mileage (-0.00000381) suggests a negative relationship between Mileage and pricesold. The categorical predictor Classic(Yes) has a negative coefficient of -0.6208, implying that Classic cars are associated with lower prices, and the categorical predictor Drive_TypeYes has a positive coefficient of 0.3312, indicating that cars with Drive_TypeYes tend to have higher prices. The significance of these coefficients is supported by the extremely low p-values. The AIC value of 453755 reflects the model's goodness of fit, with lower AIC values generally indicating better-fitting models. Overall, the results of Model_2 provide insights into the relationships between the predictors and the response variable, contributing to a comprehensive understanding of the pricing dynamics in the given dataset.

The AIC values serve as a crucial metric for model selection, offering insights into the balance between the goodness of fit and model complexity. In the case of Model_2, which includes predictors Age_of_car, Mileage, Classic, and Drive_Type, the AIC value stands at 453755. This comprehensive model aims to capture the intricate relationships within the dataset. The subsequent exploration of reduced models, achieved by sequentially omitting individual predictors, sheds light on the relative importance of each variable. Notably, the removal of Drive_Type leads to a notable increase in AIC (459142), signifying its substantial contribution to the model's explanatory capacity. As Classic, Mileage, and Age_of_car are successively removed, the AIC values ascend, underlining the significance of each predictor in enhancing the model's performance. Ultimately, the lowest AIC is associated with the full model (Model_2), advocating for the inclusion of all predictors to achieve a more accurate and comprehensive understanding of the intricate dynamics governing the relationship between the predictors and the response variable, pricesold.

glm_Model_3:

The estimated coefficients reveal the impact of each predictor on the expected log count of pricesold. Notably, the intercept is substantial (11.54), representing the log count when all predictors are zero. The negative coefficients for Age_of_car, Mileage, ClassicYes, and Drive_TypeYes suggest that an increase in these variables is associated with a decrease in the expected log count of pricesold. The significance codes indicate that all predictors are highly significant, emphasizing their relevance in predicting pricesold. The model's goodness of fit is assessed through the deviance, with a null deviance of 113912 and a residual deviance of 45507. The AIC value of 46049 further aids in model evaluation, considering the trade-off between goodness of fit and complexity, with lower AIC values indicating better-fitting models. The Fisher Scoring iterations indicate the optimization process during model fitting. Overall, this Poisson glm model provides insights into the relationships between the predictors and pricesold in the context of Boxster, offering a quantitative understanding of the factors influencing the variable of interest.

In the analysis, I evaluated three models with varying complexities based on the number of parameters, and each model was assessed using the Akaike Information Criterion (AIC). The AIC values for the models were as follows: 46049 for the model with no additional predictors (Intercept only), 46345 for the model with Drive_Type as an additional predictor, and 46414 for the model with both Classic and Drive_Type as additional predictors. The model with the lowest AIC, representing the model with no additional predictors, is favored according to the AIC criterion. This implies that, within the context of the given data, the simplicity of the model without extra predictors is more suitable, as the increased complexity introduced by including Drive_Type and/or Classic does not seem justified in terms of improving model fit.

Leave on out Evaluation:

Lower RMSE values generally indicate better predictive performance, so glm_Model_3 seems to perform better in terms of LOOCV RMSE compared to the other models.