Lab Section & Group Number: C3G2
Team Members: Mi−Ru Youn, Zhikun Wen, Zijie Zhou, Hitaishi Hitaishi, Yunbae Lim

Optimizing Used BMW Pricing: A Predictive Analysis of Features, Mileage, and Seasonality

Abstract

This study aims to predict the auction prices of used BMW cars based on key features, mileage, and seasonal trends by using a linear regression model. We have used the BMW Pricing dataset to divide the data into training and validation subsets for building and validating the model. Preprocessing of data included handling missing or implausible values, and transformation of skewed variables to be able to get a better fit of the model. Exploratory data analysis identified key predictors, such as mileage and engine power, and specific features besides the seasonal pricing trend. The final model was robust in predictive ability, explaining almost 74.4% of the variance in price. Our findings confirm that both the attributes of vehicles and seasonality of demand do influence price and hence will be helpful in improving pricing strategies. Limitations include decreased accuracy for luxury vehicles and potential effects of outliers.

Introduction

In our analysis of the BMW Pricing dataset, we explored the effects of different car features on auction prices for used BMW cars. Car pricing depends on several factors and analyzing what features are important can provide valuable insights for selling cars and/ or making car purchases. Sellers can optimize their pricing strategies by identifying key features that add value, while buyers can make more informed decisions when purchasing a vehicle.

Our research focuses on three main questions:

- How much impact does each provided equipment have on the price of the car?
  According to the description of the dataset, the requester expects each feature (equipment of the car) to correlate with the price of the used car. In other words, the requester believes that the presence of each equipment will positively impact the price. We will determine whether these features are relevant to the price and whether they show a positive, negative, or no correlation. If it turns out that certain features are relevant and have a positive correlation, the requester may consider setting higher prices for used cars that include those features.
- How does the number of days since the car was sold affect its price?
  The requester expects that the time elapsed since the car was sold may influence its value. We will analyze whether a visible trend exists between the number of days since the sale and the registration of the used car. If we find a significant relationship, the requester can use this information to develop strategies for pricing and selling cars based on how long they have been on the market.
- How can we use mileage and/or other variables to predict the price of a used car?
  Mileage, engine power, and is a critical factor in determining the depreciation of a car's value over time. By constructing a linear regression model, we can predict the price of a used car based on its mileage. This predictive model will provide an estimate of how much value a car loses as its mileage or engine power increase, or based on the absence or presence of certain equipment, helping the requester make pricing decisions for used cars with varying features.

Data

BMW Pricing Challenge: The dataset includes 4,843 BMW used cars sold via B2B auction in 2018, and the price shows the highest bid during the auction. Cars with damaged engines have already been sorted out from the dataset, so we assume that used cars in the dataset might have minor damages but no critical issues. Eight features are car equipment that the requester considers crucial factors in a used car's price. The dataset includes information on various attributes of BMW vehicles. Independent variables cover features such as mileage (distance driven), engine power (performance), registration date (vehicle age), fuel type (diesel, petrol, hybrid petrol, or electric), paint color, car type (e.g., convertible, SUV, etc), and eight unspecified car features(equipment of cars). The dependent variable is price, representing the auction value of each car in the used market. One preprocessing step we performed was converting the registration date and the sold date into a

date format, then calculating the difference between them to determine how many days had passed between the car's registration date and its sold date. We created a new variable called 'diffdate' to represent this difference and included it in our model. We conducted a thorough check for missing values across all variables, finding them to be minimal and preserving the original dataset as much as possible. During model development, however, we identified some unrealistic values (e.g., negative mileages, extreme mileages, and zero engine power). To ensure data quality, we preprocessed the data to exclude these unrealistic values. We analyzed price, mileage, engine power, and registration date and all four variables exhibit large outliers and are heavily skewed to the right.

Price Statistics: `summary(bmw$price)`

```
Min. 1st Qu.  Median  Mean   3rd Qu.  Max.
100   10900   14200   15837   18625  178500
```

Mileage Statistics: `summary(bmw$mileage)`

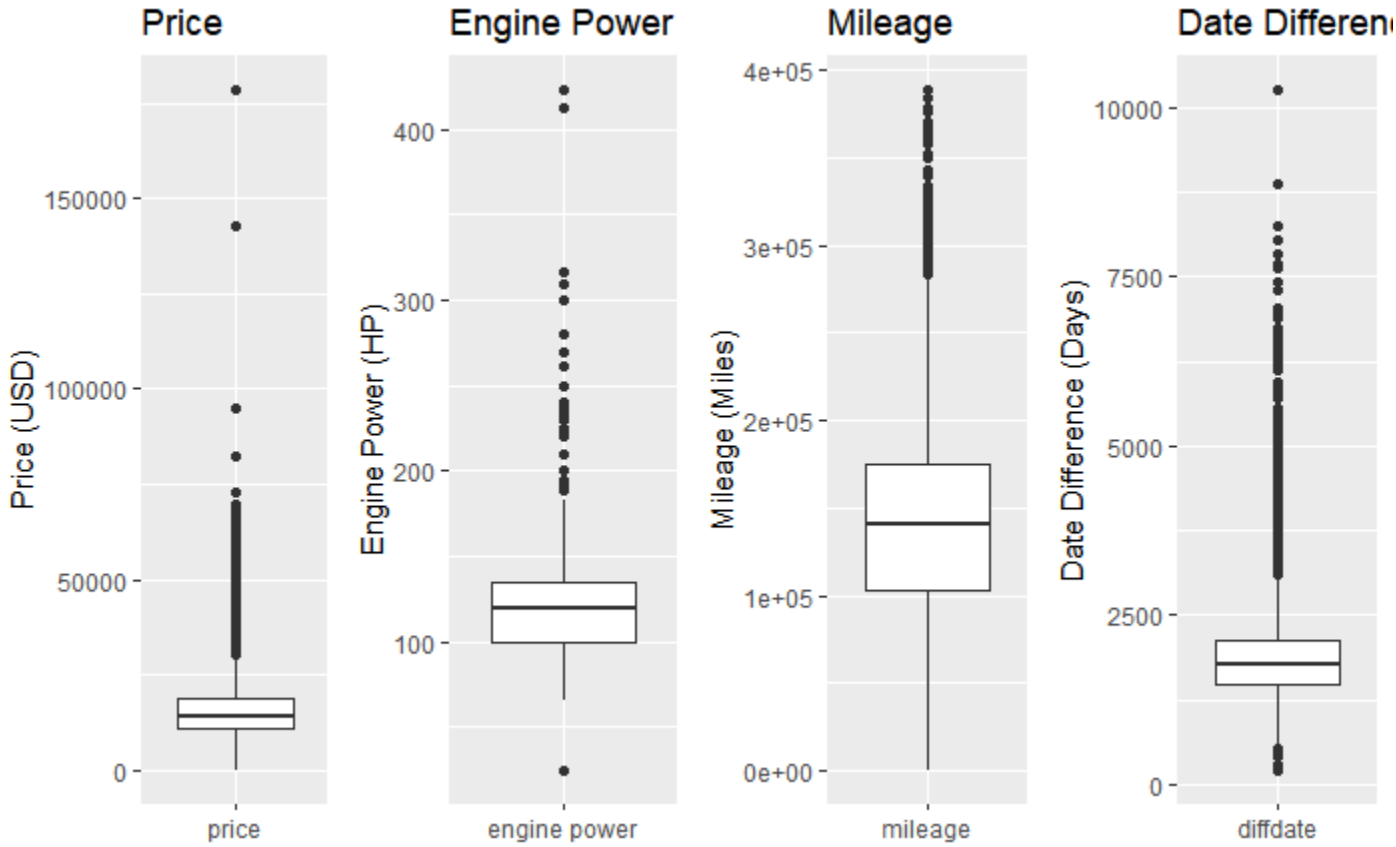
```
Min. 1st Qu.  Median  Mean   3rd Qu.  Max.
476  102877  140948  140358  175056  388616
```

Engine Power Statistics: `summary(bmw$engine_power)`

```
Min. 1st Qu.  Median   Mean   3rd Qu.  Max.
25     100     120     129     135     423
```

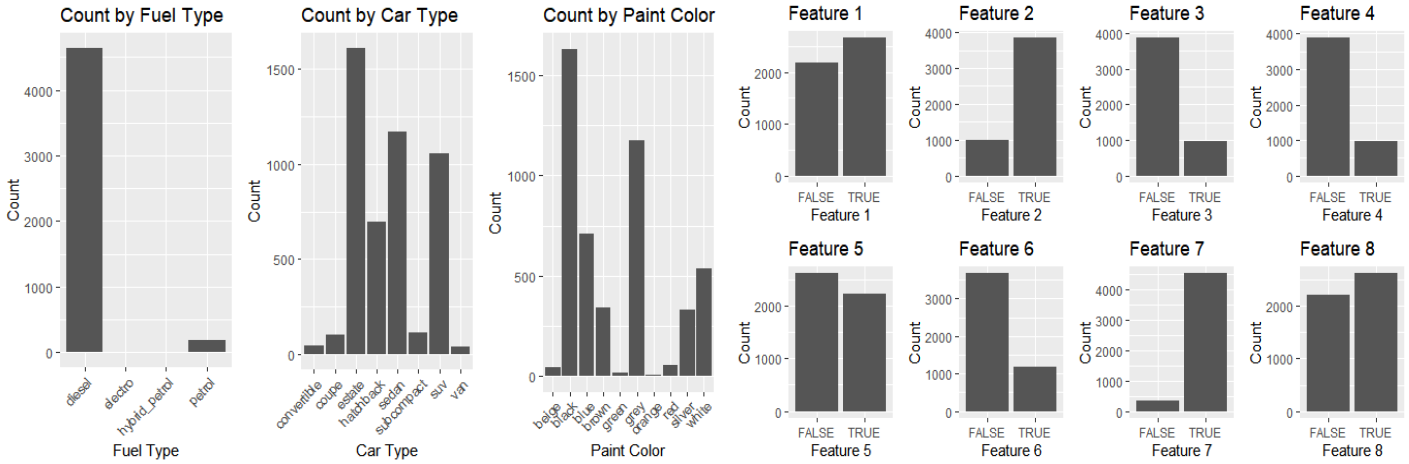
Date Difference Statistics: `summary(bmw$diffdate)`

```
Min.  1st Qu.  Median  Mean   3rd Qu.  Max.
215    1489    1765    1980    2130   10258
```



Additionally, we analyzed all the categorical variables and observed that the majority of them have an uneven distribution of counts. For fuel types, it is clear that diesel is the preferred fuel, with petrol, hybrid, and electric significantly lower. For car types, estate has the highest count, followed by sedan, SUV, and hatchback, while the rest of the car types have significantly lower counts. For paint color, black, gray, and blue are popular, followed by white, brown, and silver as moderately popular, with the rest significantly less common. For the other 8 unspecified features, Feature 2, Feature 3, Feature 4, and Feature 7 exhibit highly skewed distributions,

favoring one value (True or False), while features like Feature 1, Feature 5, and Feature 8 show a more balanced distribution.



fuel <chr>	Count <int>	Percentage <dbl>
diesel	4638	95.83
electro	3	0.06
hybrid_petrol	8	0.17
petrol	191	3.95

car_type <chr>	Count <int>	Percentage <dbl>
convertible	47	0.97
coupe	104	2.15
estate	1606	33.18
hatchback	699	14.44
sedan	1167	24.11
subcompact	116	2.40
suv	1057	21.84
van	44	0.91

paint_color <chr>	Count <int>	Percentage <dbl>
beige	41	0.85
black	1631	33.70
blue	710	14.67
brown	341	7.05
green	18	0.37
grey	1175	24.28
orange	6	0.12
red	52	1.07
silver	329	6.80
white	537	11.10

Modeling and Analysis for the Data Set

Model 1: Initial Model

Formula: price = $\beta 0 + (\beta 1 \cdot \text{mileage}) + (\beta 2 \cdot \text{engine\_power}) + (\beta 3 \cdot \text{diffdate}) + (\beta 4 \cdot \text{feature\_1}) + (\beta 5 \cdot \text{feature\_2}) + (\beta 6 \cdot \text{feature\_3}) + (\beta 7 \cdot \text{feature\_4}) + (\beta 8 \cdot \text{feature\_5}) + (\beta 9 \cdot \text{feature\_6}) + (\beta 10 \cdot \text{feature\_7}) + (\beta 11 \cdot \text{feature\_8}) + (\beta 12 \cdot \text{fuel\_diesel}) + (\beta 13 \cdot \text{fuel\_electro}) + (\beta 14 \cdot \text{fuel\_hybrid\_petrol}) + (\beta 15 \cdot \text{fuel\_petrol}) + (\beta 16 \cdot \text{car\_convertible}) + (\beta 17 \cdot \text{car\_coupe}) + (\beta 18 \cdot \text{car\_estate}) + (\beta 19 \cdot \text{car\_hatchback}) + (\beta 20 \cdot \text{car\_subcompact}) + (\beta 21 \cdot \text{car\_SUV}) + (\beta 22 \cdot \text{car\_van}) + (\beta 23 \cdot \text{paint\_beige}) + (\beta 24 \cdot \text{paint\_blue}) + (\beta 25 \cdot \text{paint\_brown}) + (\beta 26 \cdot \text{paint\_green}) + (\beta 27 \cdot \text{paint\_grey}) + (\beta 28 \cdot \text{paint\_orange}) + (\beta 29 \cdot \text{paint\_red}) + (\beta 30 \cdot \text{paint\_silver}) + (\beta 31 \cdot \text{paint\_white}) + \epsilon$

We started with a full model and examined whether each categorical and continuous variable was significant in predicting price. Mileage, engine power, days (date difference), fuel, paint color, and features were used as independent variables, with price as the dependent variable. Aside from the preprocessing step, we minimized changes or transformations to the data to ensure consistency with the assumptions of the linear model. Our initial multiple linear regression analysis revealed that some features have a more significant impact on price. From the summary statistics, mileage, engine power, certain fuel types, car types, date difference, and features 1, 3, 6, and 8 were identified as significant predictors of price. Through Type II ANOVA analysis, we observed that the same significant predictors from the summary statistics had p-values well below the threshold

of 0.05. This suggests that a reduced model can be created using the strongest predictors: mileage, engine power, fuel, car type, date difference, and features 1, 3, 6, and 8. Variables like paint color, feature 2, feature 4, feature 5, and feature 7 do not significantly affect the response (price) in this model and could potentially be removed to simplify the model. The residual sum of squares indicates that while some predictors are significant, there remains a considerable amount of unexplained variance in price.

The residuals vs fitted plot indicates that the linearity assumption is not fully met, as there appears to be a slight U-shaped curve in the residuals. Additionally, the residuals show a fanning-out pattern as the fitted values increase, which suggests heteroscedasticity. The Q-Q plot reveals deviations in the tails, indicating that the residuals are not normally distributed, and therefore, the normality assumption is not met. Finally, the variance inflation factor (VIF) analysis shows that almost all values are below 2, confirming that multicollinearity is not a concern, and there is no need to remove or combine variables.

Model 2: Reduced Model

Formula: price = $\beta 0 + (\beta 1 \cdot \text{mileage}) + (\beta 2 \cdot \text{engine\_power}) + (\beta 3 \cdot \text{diffdate}) + (\beta 4 \cdot \text{feature\_1}) + (\beta 5 \cdot \text{feature\_3}) + (\beta 6 \cdot \text{feature\_4}) + (\beta 7 \cdot \text{feature\_6}) + (\beta 8 \cdot \text{feature\_8}) + (\beta 9 \cdot \text{fuel\_diesel}) + (\beta 10 \cdot \text{fuel\_electro}) + (\beta 11 \cdot \text{fuel\_hybrid\_petrol}) + (\beta 12 \cdot \text{fuel\_petrol}) + (\beta 13 \cdot \text{car\_convertible}) + (\beta 14 \cdot \text{car\_coupe}) + (\beta 15 \cdot \text{car\_estate}) + (\beta 16 \cdot \text{car\_hatchback}) + (\beta 17 \cdot \text{car\_subcompact}) + (\beta 18 \cdot \text{car\_SUV}) + (\beta 19 \cdot \text{car\_van}) + (\beta 20 \cdot \text{paint\_beige}) + (\beta 21 \cdot \text{paint\_blue}) + (\beta 22 \cdot \text{paint\_brown}) + (\beta 23 \cdot \text{paint\_green}) + (\beta 24 \cdot \text{paint\_grey}) + (\beta 25 \cdot \text{paint\_orange}) + (\beta 26 \cdot \text{paint\_red}) + (\beta 27 \cdot \text{paint\_silver}) + (\beta 28 \cdot \text{paint\_white}) + \epsilon$

We started with the reduced model by focusing on the strongest predictors identified in the initial model. Mileage, engine power, fuel, car type, date difference, paint color, and features 1, 3, 6, and 8 were used as independent variables, with price as the dependent variable. We excluded less impactful variables (e.g., feature 2, feature 4, feature 5, feature 7) to simplify the model while preserving interpretability and predictive accuracy. No additional transformations were made to maintain consistency with the assumptions of the linear model. Our reduced multiple linear regression analysis confirmed that mileage, engine power, certain fuel types, car types, date difference, and features 1, 3, 6, and 8 remain significant predictors of price. Paint color, although retained in the model, was not statistically significant in influencing price. The model's performance metrics (e.g., R-squared of 0.669) are consistent with the initial model, confirming minimal loss of explanatory power despite reducing predictors. Through Type II ANOVA analysis, the significant predictors showed p-values below the 0.05 threshold, validating their inclusion. Variables like paint color and certain car types, however, did not have significant impacts on price, indicating areas for further refinement. The residual sum of squares highlighted that a substantial portion of the variance in price remains unexplained. Lastly, the Variance Inflation Factor (VIF) analysis indicated no multicollinearity concerns, with all VIF values well below 2. The residuals vs fitted plot for Model 2 reveals notable issues with linearity and homoscedasticity. A slight curve in the red trend line suggests deviations from linearity, particularly at higher and lower fitted values, where the residuals consistently fall above or below the horizontal axis. Additionally, the residuals exhibit a fanning pattern as fitted values increase, indicating heteroscedasticity. However, some deviations are observed at higher fitted values, suggesting potential instability in predicting high-priced vehicles. This aligns with the model's limitations in capturing all variability in price. This reduced model simplifies interpretation while retaining key predictors, but its inability to significantly improve predictive power or reduce unexplained variance led us to explore further simplifications, such as the stepwise regression model.

Model 3: Stepwise Regression

Formula: price = $\beta 0 + (\beta 1 \cdot \text{mileage}) + (\beta 2 \cdot \text{engine\_power}) + (\beta 3 \cdot \text{diffdate}) + (\beta 4 \cdot \text{feature\_1}) + (\beta 5 \cdot \text{feature\_3}) + (\beta 6 \cdot \text{feature\_4}) + (\beta 7 \cdot \text{feature\_6}) + (\beta 8 \cdot \text{feature\_8}) + (\beta 9 \cdot \text{fuel\_diesel}) + (\beta 10 \cdot \text{fuel\_electro}) + (\beta 11 \cdot \text{fuel\_hybrid\_petrol}) + (\beta 12 \cdot \text{fuel\_petrol}) + (\beta 13 \cdot \text{car\_convertible}) + (\beta 14 \cdot \text{car\_coupe}) + (\beta 15 \cdot \text{car\_estate}) + (\beta 16 \cdot \text{car\_hatchback}) + (\beta 17 \cdot \text{car\_subcompact}) + (\beta 18 \cdot \text{car\_SUV}) + (\beta 19 \cdot \text{car\_van}) + (\beta 20 \cdot \text{paint\_beige}) + (\beta 21 \cdot \text{paint\_blue}) + (\beta 22 \cdot \text{paint\_brown}) + (\beta 23 \cdot \text{paint\_green}) + (\beta 24 \cdot \text{paint\_grey}) + (\beta 25 \cdot \text{paint\_orange}) + (\beta 26 \cdot \text{paint\_red}) + (\beta 27 \cdot \text{paint\_silver}) + (\beta 28 \cdot \text{paint\_white}) + \epsilon$

The stepwise regression produced the same model as the reduced model we created. The coefficients of the variables in this model show only slight changes compared to those from the full model. The direction of the

correlation between explanatory variables and the response variable also remains unchanged. The adjusted R-squared for this model is 0.6312, which is higher than that of other models, indicating that this model explains more of the variability in the response variable. Additionally, this model has the smallest AIC value among all the models, suggesting it is the most parsimonious and fits the data better. Since the stepwise regression produced the same model as the reduced model, the analysis and conclusions are identical to those of the reduced model.

Model 4: Add Transformation

Formula: $\text{sqrt(price)} = \beta0 + (\beta1 \cdot \text{mileage}) + (\beta2 \cdot \text{engine\_power}) + (\beta3 \cdot \text{diffdate}) + (\beta4 \cdot \text{feature\_1}) + (\beta5 \cdot \text{feature\_3}) + (\beta6 \cdot \text{feature\_4}) + (\beta7 \cdot \text{feature\_6}) + (\beta8 \cdot \text{feature\_8}) + (\beta9 \cdot \text{fuel\_diesel}) + (\beta10 \cdot \text{fuel\_electro}) + (\beta11 \cdot \text{fuel\_hybrid\_petrol}) + (\beta12 \cdot \text{fuel\_petrol}) + (\beta13 \cdot \text{car\_convertible}) + (\beta14 \cdot \text{car\_coupe}) + (\beta15 \cdot \text{car\_estate}) + (\beta16 \cdot \text{car\_hatchback}) + (\beta17 \cdot \text{car\_subcompact}) + (\beta18 \cdot \text{car\_SUV}) + (\beta19 \cdot \text{car\_van}) + (\beta20 \cdot \text{paint\_beige}) + (\beta21 \cdot \text{paint\_blue}) + (\beta22 \cdot \text{paint\_brown}) + (\beta23 \cdot \text{paint\_green}) + (\beta24 \cdot \text{paint\_grey}) + (\beta25 \cdot \text{paint\_orange}) + (\beta26 \cdot \text{paint\_red}) + (\beta27 \cdot \text{paint\_silver}) + (\beta28 \cdot \text{paint\_white}) + \epsilon$

The final model included one transformation, where the response variable (price) was square-rooted. The residuals vs fitted plot indicates that the linearity assumption is met, as there is no longer a slight U-shaped curve in the residuals. Homoscedasticity is also largely satisfied, with residuals relatively evenly spread and no discernible patterns, such as fanning out or a funnel shape. Additionally, the independence assumption is met, as the residuals appear randomly scattered around zero, with no visible patterns indicating dependence. However, the Q-Q plot reveals that the normality assumption is not fully satisfied. While most residuals lie approximately on the line, deviations in the tails suggest the presence of extreme outliers or a heavy-tailed distribution. Finally, the Variance Inflation Factor (VIF) values are all below 2, confirming that multicollinearity is not a concern in the model.

Final Model: Ridge Regression

Formula: $\text{sqrt(price)} = \beta0 + (\beta1 \cdot \text{mileage}) + (\beta2 \cdot \text{engine\_power}) + (\beta3 \cdot \text{diffdate}) + (\beta4 \cdot \text{feature\_1}) + (\beta5 \cdot \text{feature\_3}) + (\beta6 \cdot \text{feature\_4}) + (\beta7 \cdot \text{feature\_6}) + (\beta8 \cdot \text{feature\_8}) + (\beta9 \cdot \text{fuel\_diesel}) + (\beta10 \cdot \text{fuel\_electro}) + (\beta11 \cdot \text{fuel\_hybrid\_petrol}) + (\beta12 \cdot \text{fuel\_petrol}) + (\beta13 \cdot \text{car\_convertible}) + (\beta14 \cdot \text{car\_coupe}) + (\beta15 \cdot \text{car\_estate}) + (\beta16 \cdot \text{car\_hatchback}) + (\beta17 \cdot \text{car\_subcompact}) + (\beta18 \cdot \text{car\_SUV}) + (\beta19 \cdot \text{car\_van}) + (\beta20 \cdot \text{paint\_beige}) + (\beta21 \cdot \text{paint\_blue}) + (\beta22 \cdot \text{paint\_brown}) + (\beta23 \cdot \text{paint\_green}) + (\beta24 \cdot \text{paint\_grey}) + (\beta25 \cdot \text{paint\_orange}) + (\beta26 \cdot \text{paint\_red}) + (\beta27 \cdot \text{paint\_silver}) + (\beta28 \cdot \text{paint\_white}) + \epsilon$

In this analysis, we applied Ridge regression to predict the target variable (sqrt_price) using a set of features, including mileage, engine_power, fuel, car_type, diff_date, and other features. Ridge regression was chosen to address potential multicollinearity among predictors and to regularize the model, shrinking coefficients to reduce overfitting while retaining all features. We used cross-validation to determine the optimal regularization parameter (λ) that minimizes the mean squared error. The resulting model retained all features, but the coefficients reflected that they were all important. For instance, feature_8, car_type, and engine_power had the largest coefficients, indicating a strong positive influence on the target variable, while mileage and diffdate had coefficients close to zero, suggesting minimal impact. The model achieved an R^2 of 0.751 and an adjusted R^2 of 0.750, meaning it explains about 75% of the variance in the target variable, demonstrating a good fit. However, one thing to note is there was no change in features between the previous three models.

Prediction

The validation dataset consists of used BMW cars not included in the training set. To assess its similarity to the training set, we compared key summary statistics (e.g., price, mileage, engine power) and distribution plots. The comparison shows the validation set shares similar ranges and distributions for key variables, ensuring compatibility with the model trained on the training set. However, some minor differences were noted, such as slightly fewer cars with extreme mileage and slightly higher representation of certain fuel types (e.g., hybrid vehicles).

- Scatter Plot of True vs. Predicted values:

    - This scatter plot demonstrates the alignment between actuarial and predicted prices. Most predictions are close to the true values, particularly in the middle price range, though slight deviations are observed for luxury vehicles.
- Residual plot:
    - The residual plot demonstrates randomly scattered residuals around zero, validating the assumptions of the model. However, for higher prices, residuals tend to diverge, suggesting minor prediction instability for high-value cars.
- Residual Histogram:
    - A histogram of residuals demonstrates a normal distribution centered around zero, confirming the model's suitability for predicting mid-range prices
- Root Mean Squared Error (RMSE) : 16.8422
- Mean Absolute Error (MAE) : 10.6664
- R-squared (R^2) Score: 0.7404
- Mean Absolute Percentage Error (MPE): 0.1381

Discussion

Our main findings from this study are based on 3 questions we are analyzing. First we found some significant predictors, which includes the mileage with a strong negative correlation with price, the engine power with positive influence, and those feature variables, each of them displaying either positive or negative effects. These findings align with our expectations regarding feature value additions and specific functional impacts on the vehicle's desirability. In addition, sale months were included, but not all months were statistically significant predictors, certain months associated with spring and summer showed higher prices. Our model performance on the ANOVA results shows us the variables are highly significant ($p < 0.01$), which validates the inclusion in our model. The residual vs. fitted plots show us that the model assumptions of homoscedasticity are largely met. But the impact of outliers and high-leverage points slightly skew the model's accuracy, especially in high price ranges.

The primary goal is to develop a predictive model to identify key variables influencing car prices and quantify their impacts. Analysis has successfully identified critical factors affecting prices, these variables were consistently significant across different model iterations. Our goals yet to be fully realized includes explaining more variance in prices. Also address data complexity, since certain categorical variables were included but showed limited impact on pricing. Simplifying these variables or excluding less relevant variables might enhance the model's interpretability.

Model- related limitations: We still have percentages of variability in car prices remains unexplained, indicating potential omitted variables(e.g. The regional market trends, specific model preferences or consumer sentiment). The linear model may simplify certain dynamics of the relationship. For instance the impact of mileage on price may decrease or plateau at higher value, the interaction effects between variables were not explored, potentially missing combined impacts. Data Limitation: The outliers and high-leverage observations disproportionately influence the regression coefficients. Variables such as Feature1 through Fearture8 are not explicitly described, which makes it challenging to interpret to real-world implications.

Overall Author Contribution Statement

- **Hitaishi Hitaishi: Interpretation, Writing (review and editing)**
- **Yunbae Lim :Writing (review and editing), Interpretation**
- **Zhikun Wen: Interpretation, Writing (review and editing)**
- **Zijie Zhou: Formal analysis (coding), Interpretation, Writing (review and editing)**
- **Mi−Ru Youn: Formal analysis (coding), Interpretation, Writing (review and editing)**

Lab Report 4

2024-10-31

Clean the data

```
summary(bmw$mileage)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     -64  102914  141080  140963  175196 1000376
```

```
bmw <- bmw[bmw$mileage > 0, ]
bmw <- bmw[bmw$mileage < max(bmw$mileage), ]

summary(bmw$engine_power)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0     100     120     129     135     423
```

```
bmw <- bmw[bmw$engine_power > 0, ]

bmw$sqrt_price <- sqrt(bmw$price)
bmw$log_price <- log(bmw$price)
bmw$date <- as.Date(bmw$registration_date, format = "%m/%d/%Y")
bmw$sold_at <- as.Date(bmw$sold_at, format = "%m/%d/%Y")
bmw$diffdate <- as.numeric(difftime(bmw$sold_at, bmw$date, units = "days"))
```

Descriptive statistics of continious variables

```
summary(bmw$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     100   10800   14200   15820   18600  178500
```

```
summary(bmw$mileage)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     476  103011  141085  140827  175185  484615
```

```
summary(bmw$engine_power)
```

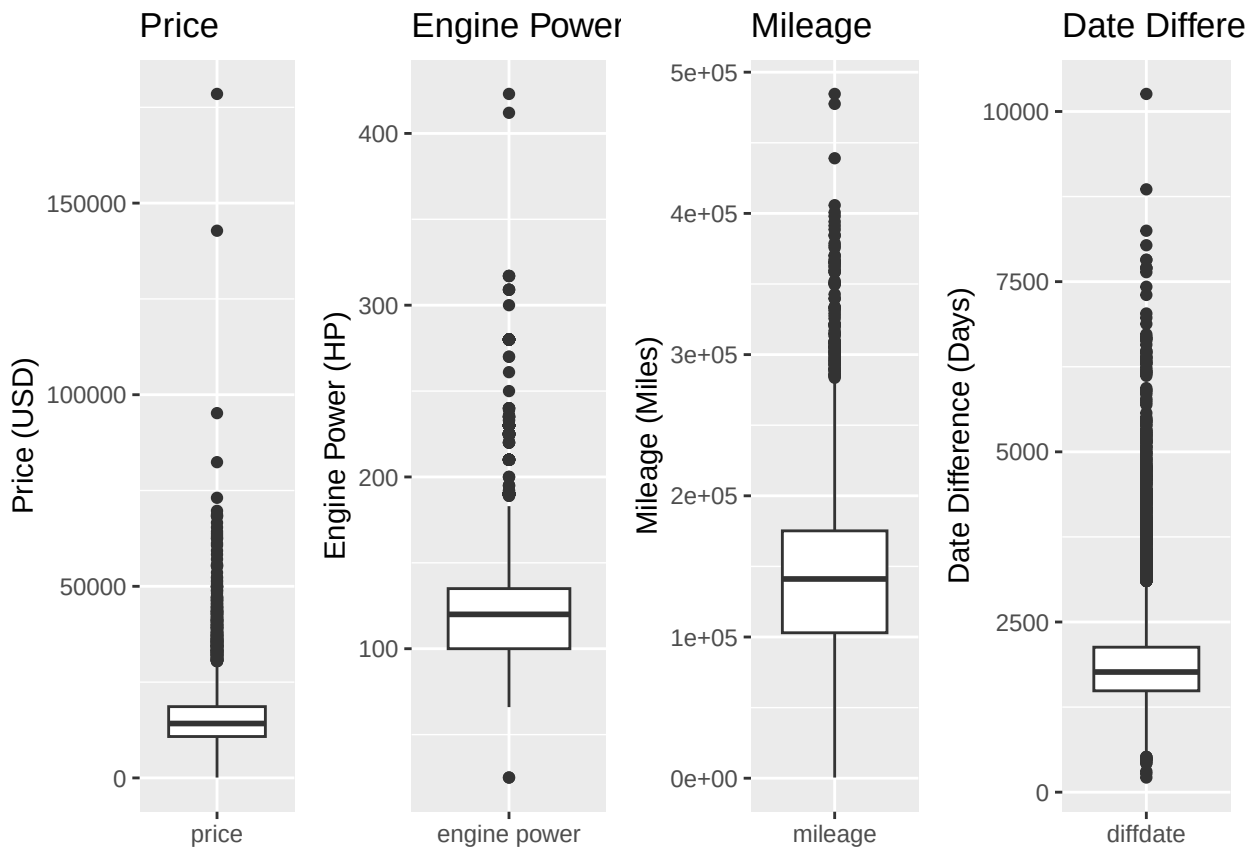
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      25     100     120     129     135     423
```

```r
summary(bmw$diffdate)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     215    1489    1765    1984    2130   10258
```

Plots of continuous Variables

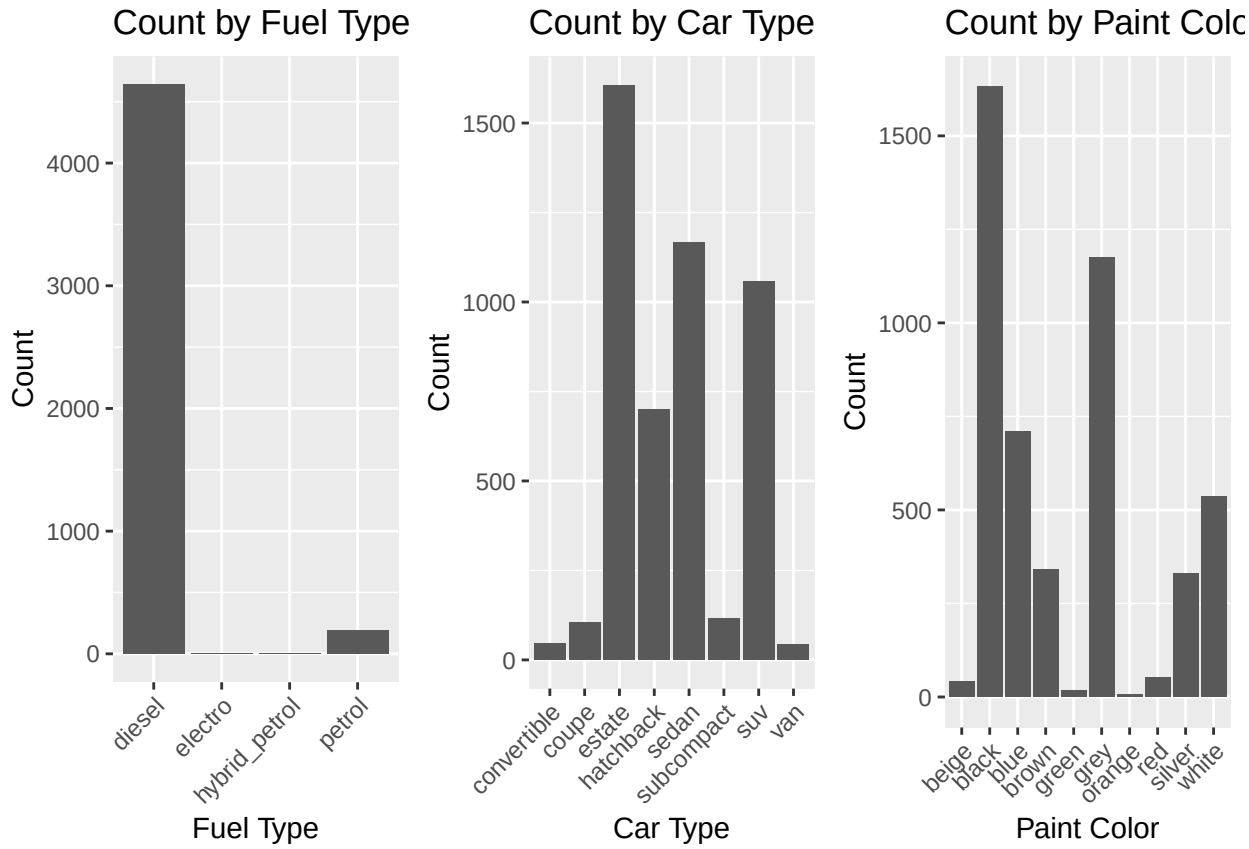
```r
# Create individual boxplots with y-axis labels and proper formatting
p1 <- ggplot() +
  geom_boxplot(aes(y = bmw$price, x = "price")) +
  labs(y = "Price (USD)", x = NULL) +
  ggtitle("Price")

p2 <- ggplot() +
  geom_boxplot(aes(y = bmw$engine_power, x = "engine power")) +
  labs(y = "Engine Power (HP)", x = NULL) +
  ggtitle("Engine Power")

p3 <- ggplot() +
  geom_boxplot(aes(y = bmw$mileage, x = "mileage")) +
  labs(y = "Mileage (Miles)", x = NULL) +
  ggtitle("Mileage")

# Box plot for Days since Registration Date
p4 <- ggplot() +
  geom_boxplot(aes(y = bmw$diffdate, x = "diffdate")) +
  labs(y = "Date Difference (Days)", x = NULL) +
  ggtitle("Date Difference")

# Arrange the three plots side by side
grid.arrange(p1, p2, p3, p4, ncol = 4)
```



Plots of categorical variables

```r
# Bar plot for counts of Fuel Type
p5 <- ggplot(bmw, aes(x = fuel)) +
  geom_bar() +
  labs(y = "Count", x = "Fuel Type") +
  ggtitle("Count by Fuel Type") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Bar plot for counts of Car Type
p6 <- ggplot(bmw, aes(x = car_type)) +
  geom_bar() +
  labs(y = "Count", x = "Car Type") +
  ggtitle("Count by Car Type") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Bar plot for counts of Paint Color
p7 <- ggplot(bmw, aes(x = paint_color)) +
  geom_bar() +
  labs(y = "Count", x = "Paint Color") +
  ggtitle("Count by Paint Color") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
# Arrange the three plots side by side
grid.arrange(p5, p6, p7, ncol = 3)
```




```
fuel_summary <- bmw %>%
  group_by(fuel) %>%
  summarise(Count = n()) %>%
  mutate(Percentage = round((Count / sum(Count)) * 100, 2))

print(fuel_summary)
```

```
## # A tibble: 4 x 3
##   fuel          Count Percentage
##   <chr>         <int>      <dbl>
## 1 diesel         4638      95.8
## 2 electro           3       0.06
## 3 hybrid_petrol     8       0.17
## 4 petrol          191       3.95
```

```
# Car Type Counts and Percentages
car_type_summary <- bmw %>%
  group_by(car_type) %>%
  summarise(Count = n()) %>%
  mutate(Percentage = round((Count / sum(Count)) * 100, 2))

print(car_type_summary)
```

```
## # A tibble: 8 x 3
##   car_type    Count Percentage
##   <chr>       <int>      <dbl>
## 1 convertible    47       0.97
## 2 coupe         104       2.15
## 3 estate       1606      33.2
## 4 hatchback     699      14.4
## 5 sedan        1167      24.1
## 6 subcompact    116       2.4
## 7 suv          1057      21.8
## 8 van            44       0.91
```

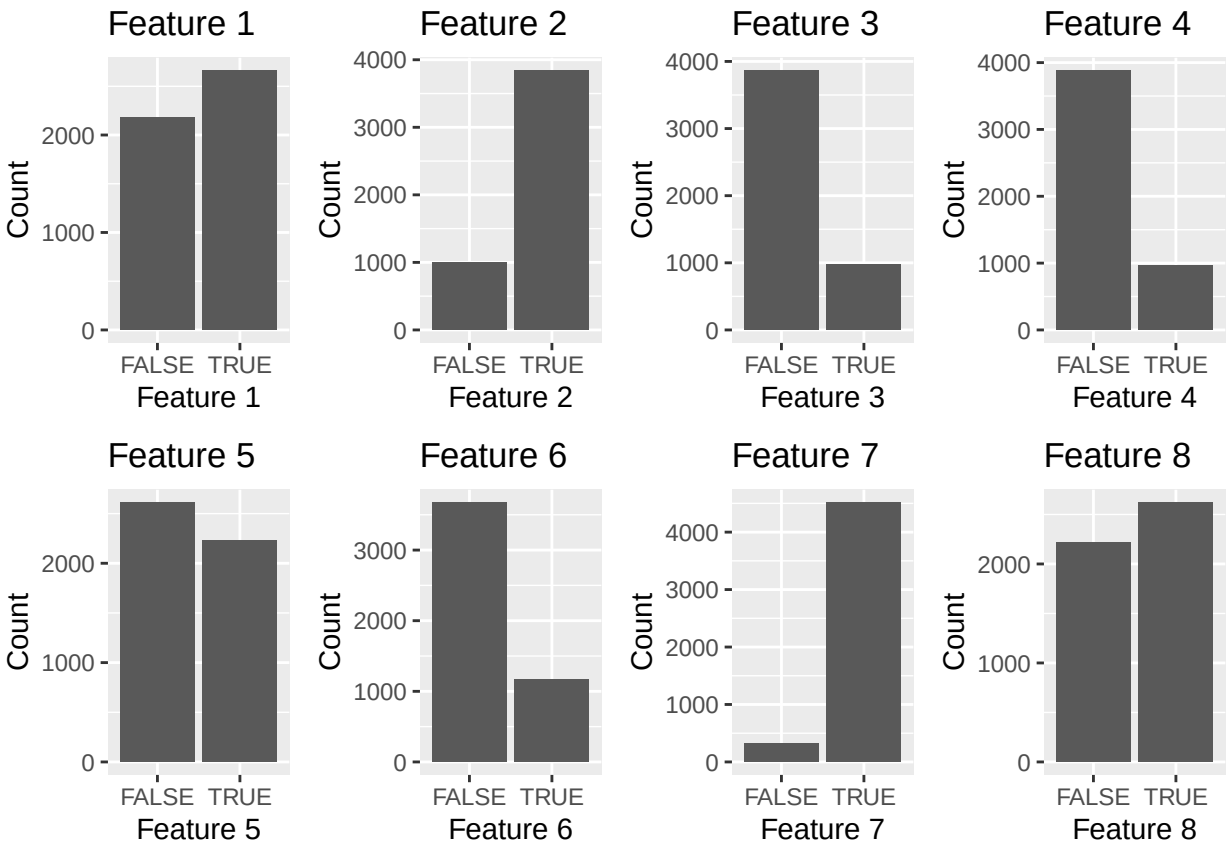
```
# Paint Color Counts and Percentages
paint_color_summary <- bmw %>%
  group_by(paint_color) %>%
  summarise(Count = n()) %>%
  mutate(Percentage = round((Count / sum(Count)) * 100, 2))

print(paint_color_summary)
```

```
## # A tibble: 10 x 3
##    paint_color Count Percentage
##    <chr>       <int>      <dbl>
##  1 beige          41       0.85
##  2 black        1631      33.7
##  3 blue          710      14.7
##  4 brown         341       7.05
##  5 green          18       0.37
##  6 grey         1175      24.3
##  7 orange          6       0.12
##  8 red            52       1.07
##  9 silver        329       6.8
## 10 white         537      11.1
```

Plots of features

```
# Bar plot for counts of Feature 1
f1 <- ggplot(bmw, aes(x = feature_1)) +
  geom_bar() +
  labs(y = "Count", x = "Feature 1") +
  ggtitle("Feature 1")

# Bar plot for counts of Feature 2
f2 <- ggplot(bmw, aes(x = feature_2)) +
  geom_bar() +
  labs(y = "Count", x = "Feature 2") +
  ggtitle("Feature 2")

# Bar plot for counts of Feature 3
f3 <- ggplot(bmw, aes(x = feature_3)) +
  geom_bar() +
```

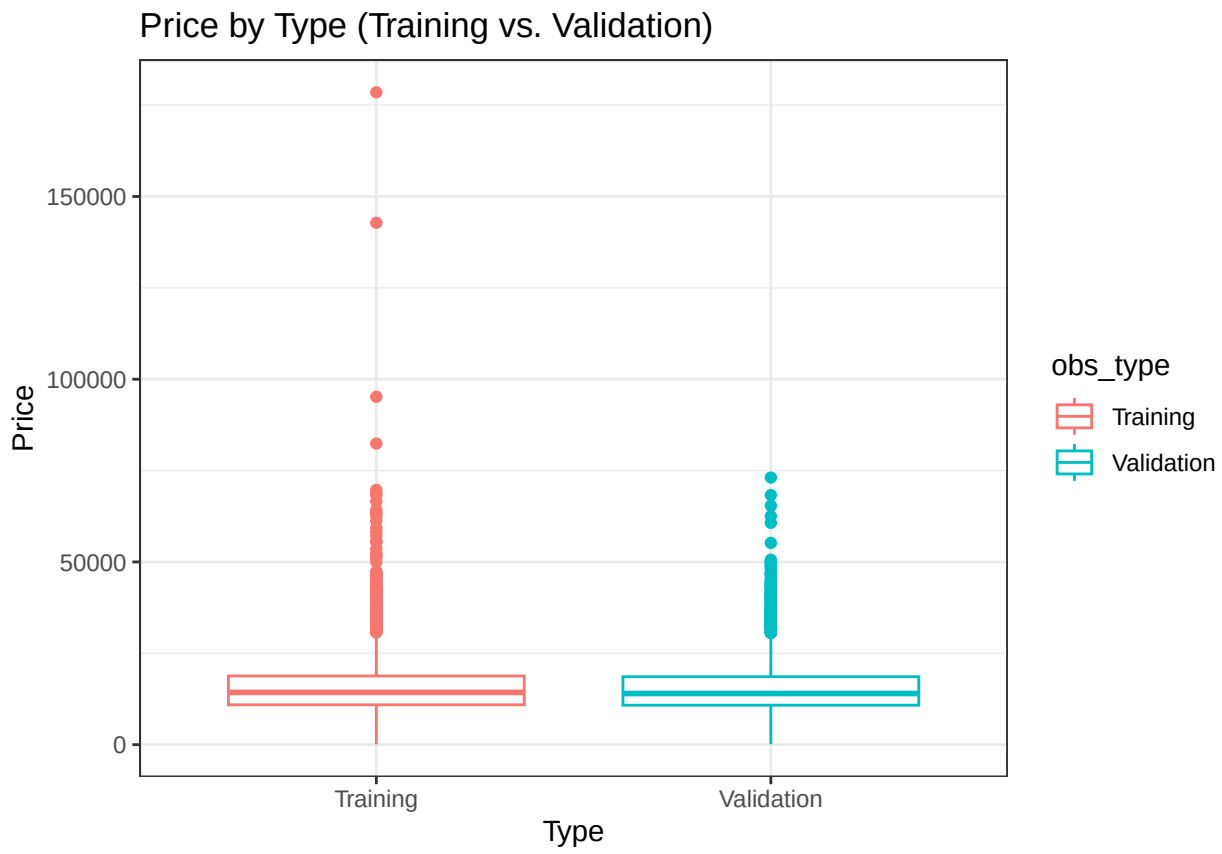
```
  labs(y = "Count", x = "Feature 3") +
  ggtitle("Feature 3")

# Bar plot for counts of Feature 4
f4 <- ggplot(bmw, aes(x = feature_4)) +
  geom_bar() +
  labs(y = "Count", x = "Feature 4") +
  ggtitle("Feature 4")

# Bar plot for counts of Feature 5
f5 <- ggplot(bmw, aes(x = feature_5)) +
  geom_bar() +
  labs(y = "Count", x = "Feature 5") +
  ggtitle("Feature 5")

# Bar plot for counts of Feature 6
f6 <- ggplot(bmw, aes(x = feature_6)) +
  geom_bar() +
  labs(y = "Count", x = "Feature 6") +
  ggtitle("Feature 6")

# Bar plot for counts of Feature 7
f7 <- ggplot(bmw, aes(x = feature_7)) +
  geom_bar() +
  labs(y = "Count", x = "Feature 7") +
  ggtitle("Feature 7")

# Bar plot for counts of Feature 8
f8 <- ggplot(bmw, aes(x = feature_8)) +
  geom_bar() +
  labs(y = "Count", x = "Feature 8") +
  ggtitle("Feature 8")

# Arrange the three plots side by side
grid.arrange(f1, f2, f3, f4, f5, f6, f7, f8, ncol = 4)
```



Grouped boxplot for Price by Type

```
ggplot(bmw, aes(x = obs_type, y = price, color = obs_type)) +
  geom_boxplot(position = position_dodge(width = 0.8)) +
  labs(x = "Type", y = "Price") +
  scale_fill_manual(name = "Dataset Type", values = c("Training" = "blue", "Validation" = "red")) +
  ggtitle("Price by Type (Training vs. Validation)") +
  theme_bw()
```

```
## Warning: No shared levels found between `names(values)` of the manual scale and the
## data's fill values.
```



Grouped boxplot for Mileage by Type

```
ggplot(bmw, aes(x = obs_type, y = mileage, color = obs_type)) +
  geom_boxplot(position = position_dodge(width = 0.8)) +
  labs(x = "Type", y = "Mileage") +
  scale_fill_manual(name = "Dataset Type", values = c("Training" = "blue", "Validation" = "red")) +
  ggtitle("Mileage by Type (Training vs. Validation)") +
  theme_bw()
```

```
## Warning: No shared levels found between `names(values)` of the manual scale and the
## data's fill values.
```


```
Training_data <- bmw[bmw$obs_type == "Training", ]
Validation_data <- bmw[bmw$obs_type == "Validation", ]
```

Select significant predictors

```
m.ols <- lm(price ~ mileage + engine_power + fuel + car_type + diffdate + paint_color + feature_1 + fea
summary(m.ols)
```

```
##
## Call:
## lm(formula = price ~ mileage + engine_power + fuel + car_type +
##     diffdate + paint_color + feature_1 + feature_2 + feature_3 +
##     feature_4 + feature_5 + feature_6 + feature_7 + feature_8,
##     data = Training_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -27934  -2351   -241   1801 156885
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.370e+04  1.929e+03   7.100 1.64e-12 ***
## mileage             -3.813e-02  2.620e-03 -14.553  < 2e-16 ***
```

```
## engine_power        1.072e+02  4.189e+00  25.578  < 2e-16 ***
## fuelelectro         4.102e+03  4.263e+03   0.962 0.336091
## fuelhybrid_petrol   1.279e+04  2.287e+03   5.594 2.47e-08 ***
## fuelpetrol         -1.961e+03  6.722e+02  -2.917 0.003569 **
## car_typecoupe      -1.801e+03  1.491e+03  -1.208 0.227358
## car_typeestate     -5.371e+03  1.290e+03  -4.164 3.24e-05 ***
## car_typehatchback  -4.066e+03  1.313e+03  -3.097 0.001975 **
## car_typesedan      -3.141e+03  1.290e+03  -2.436 0.014926 *
## car_typesubcompact -3.597e+03  1.489e+03  -2.416 0.015761 *
## car_typesuv        -9.410e+02  1.314e+03  -0.716 0.474063
## car_typevan        -6.126e+03  1.779e+03  -3.443 0.000585 ***
## diffdate           -2.723e+00  1.765e-01 -15.432  < 2e-16 ***
## paint_colorblack   -1.862e+02  1.251e+03  -0.149 0.881661
## paint_colorblue    -4.047e+02  1.273e+03  -0.318 0.750526
## paint_colorbrown   -5.570e+02  1.315e+03  -0.423 0.671973
## paint_colorgreen   -1.913e+02  2.376e+03  -0.081 0.935831
## paint_colorgrey    -1.094e+01  1.258e+03  -0.009 0.993059
## paint_colororange  -4.421e+03  3.305e+03  -1.338 0.181082
## paint_colorred      5.014e+02  1.677e+03   0.299 0.764952
## paint_colorsilver  -8.221e+02  1.319e+03  -0.623 0.533126
## paint_colorwhite   -9.166e+00  1.288e+03  -0.007 0.994322
## feature_1TRUE       1.610e+03  2.845e+02   5.658 1.71e-08 ***
## feature_2TRUE       4.608e+02  3.664e+02   1.258 0.208567
## feature_3TRUE       9.915e+02  3.288e+02   3.016 0.002591 **
## feature_4TRUE       7.431e+02  4.020e+02   1.848 0.064665 .
## feature_5TRUE      -3.911e+02  2.868e+02  -1.364 0.172818
## feature_6TRUE       7.011e+02  3.064e+02   2.288 0.022197 *
## feature_7TRUE       2.873e+02  5.718e+02   0.502 0.615365
## feature_8TRUE       1.961e+03  2.998e+02   6.542 7.41e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5991 on 2402 degrees of freedom
## Multiple R-squared:  0.6342, Adjusted R-squared:  0.6296
## F-statistic: 138.8 on 30 and 2402 DF,  p-value: < 2.2e-16
```

```
Anova(m.ols, type = "II")
```

```
## Anova Table (Type II tests)
##
## Response: price
##                   Sum Sq   Df  F value    Pr(>F)
## mileage      7.6024e+09    1 211.7960 < 2.2e-16 ***
## engine_power 2.3484e+10    1 654.2386 < 2.2e-16 ***
## fuel         1.4937e+09    3  13.8711 5.744e-09 ***
## car_type     5.5389e+09    7  22.0441 < 2.2e-16 ***
## diffdate     8.5486e+09    1 238.1554 < 2.2e-16 ***
## paint_color  2.1336e+08    9   0.6605  0.745373
## feature_1    1.1491e+09    1  32.0126 1.714e-08 ***
## feature_2    5.6793e+07    1   1.5822  0.208567
## feature_3    3.2643e+08    1   9.0939  0.002591 **
## feature_4    1.2264e+08    1   3.4166  0.064665 .
## feature_5    6.6744e+07    1   1.8594  0.172818
## feature_6    1.8798e+08    1   5.2370  0.022197 *
```

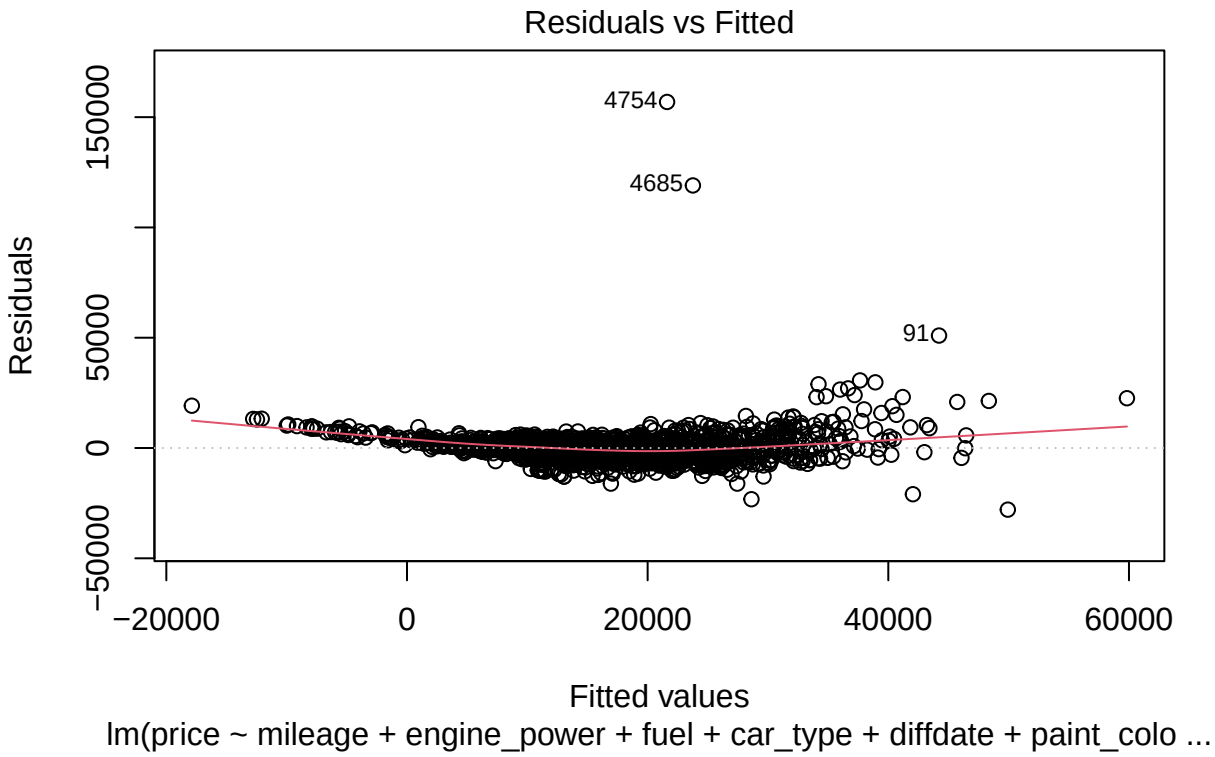
```
## feature_7    9.0636e+06    1   0.2525  0.615365
## feature_8    1.5360e+09    1  42.7922 7.414e-11 ***
## Residuals    8.6220e+10 2402
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

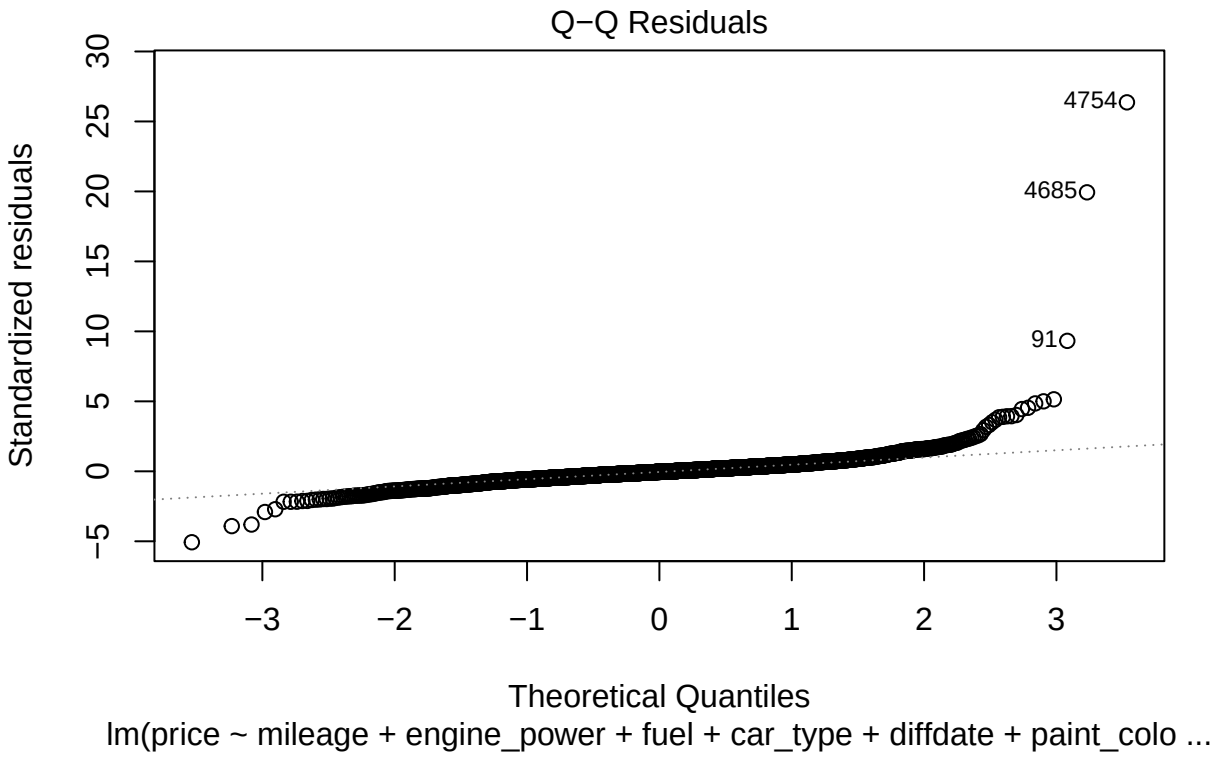
```
vif(m.ols)
```

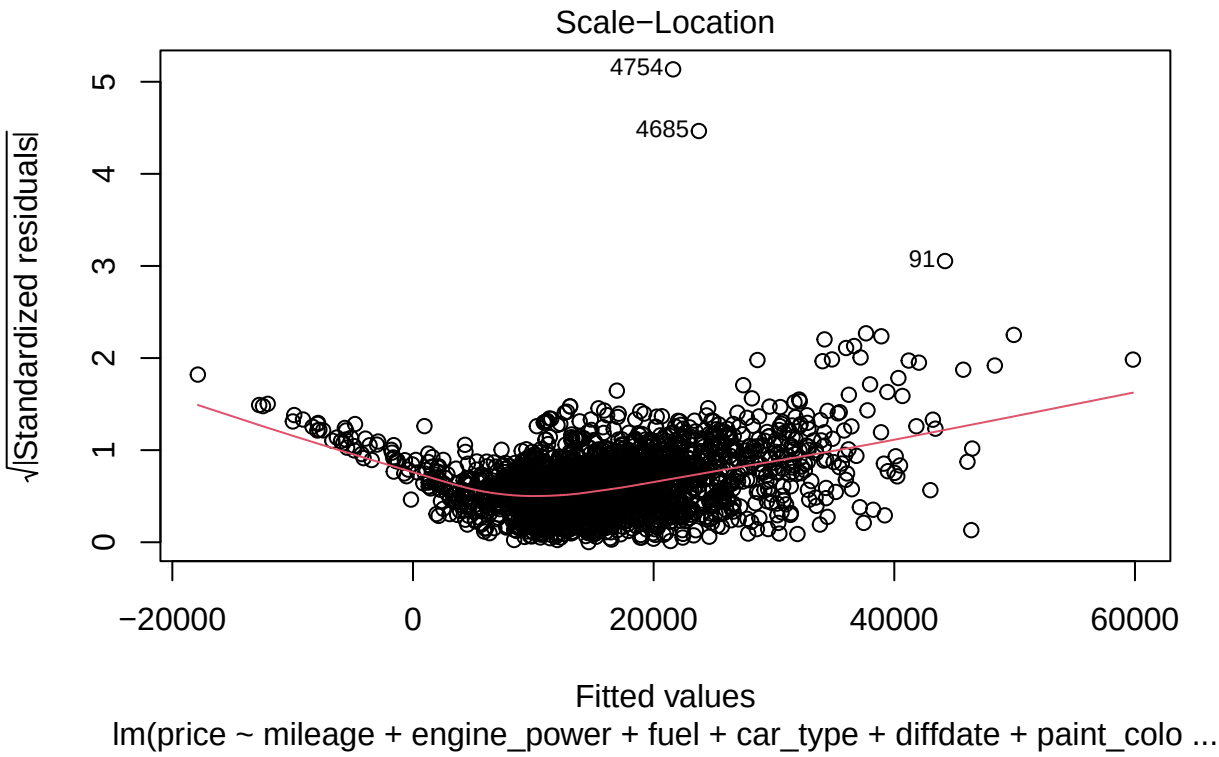
```
##                   GVIF Df GVIF^(1/(2*Df))
## mileage      1.586728  1        1.259654
## engine_power 1.927475  1        1.388335
## fuel         1.240573  3        1.036582
## car_type     2.188759  7        1.057547
## diffdate     1.817560  1        1.348169
## paint_color  1.255443  9        1.012718
## feature_1    1.352290  1        1.162880
## feature_2    1.429376  1        1.195565
## feature_3    1.212306  1        1.101048
## feature_4    1.734883  1        1.317150
## feature_5    1.381497  1        1.175371
## feature_6    1.149496  1        1.072146
## feature_7    1.306015  1        1.142810
## feature_8    1.513161  1        1.230106
```

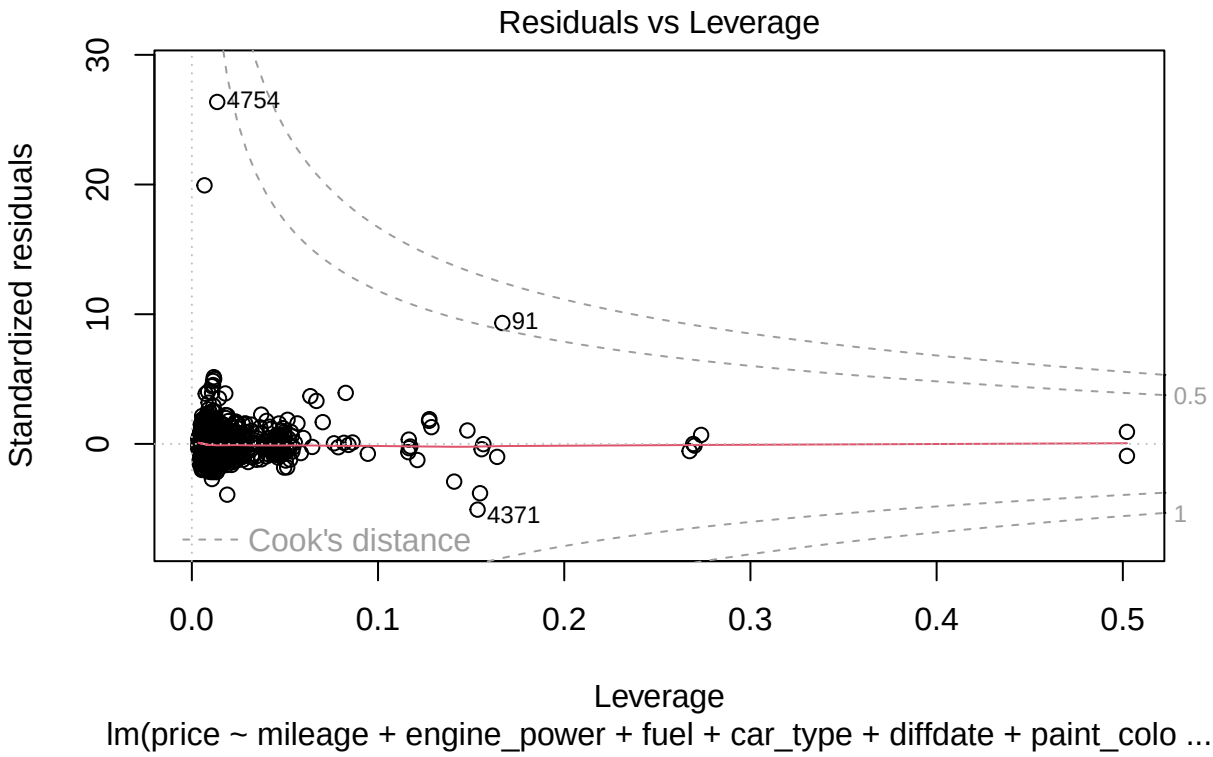
Model diagnostic plots for full model

```
plot(m.ols)
```

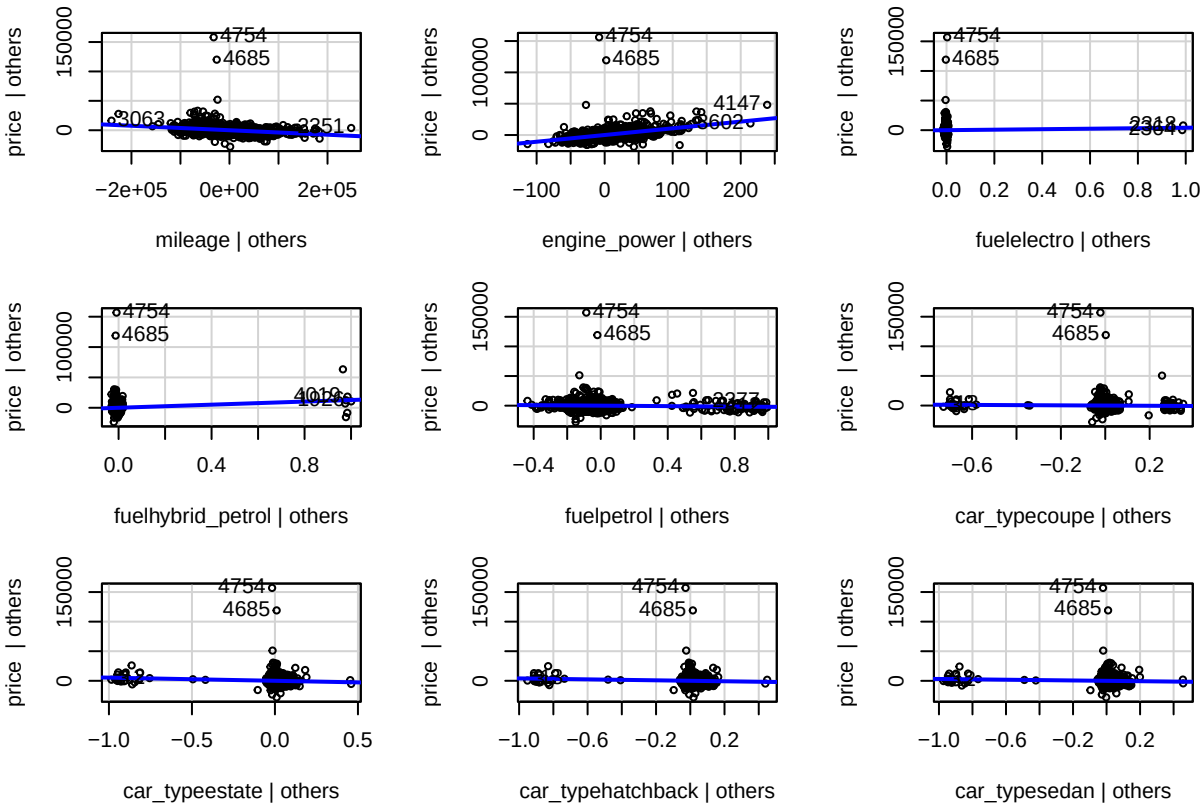

Residuals vs Fitted
Residuals
Fitted values
lm(price ~ mileage + engine_power + fuel + car_type + diffdate + paint_colo ...
4754
4685
91
150000
50000
0
−50000
−20000
0
20000
40000
60000

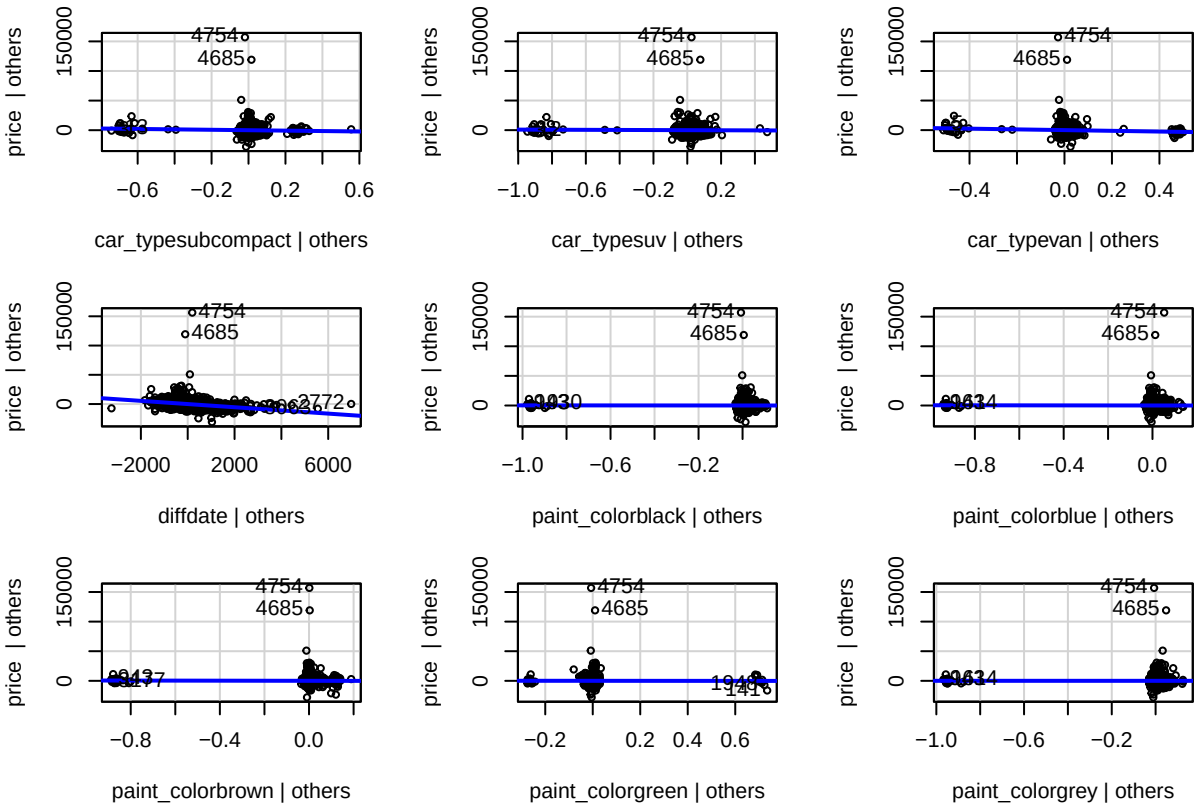

Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(price ~ mileage + engine_power + fuel + car_type + diffdate + paint_colo ...
4754
4685
91
30
25
20
15
10
5
0
−5
−3
−2
−1
0
1
2
3

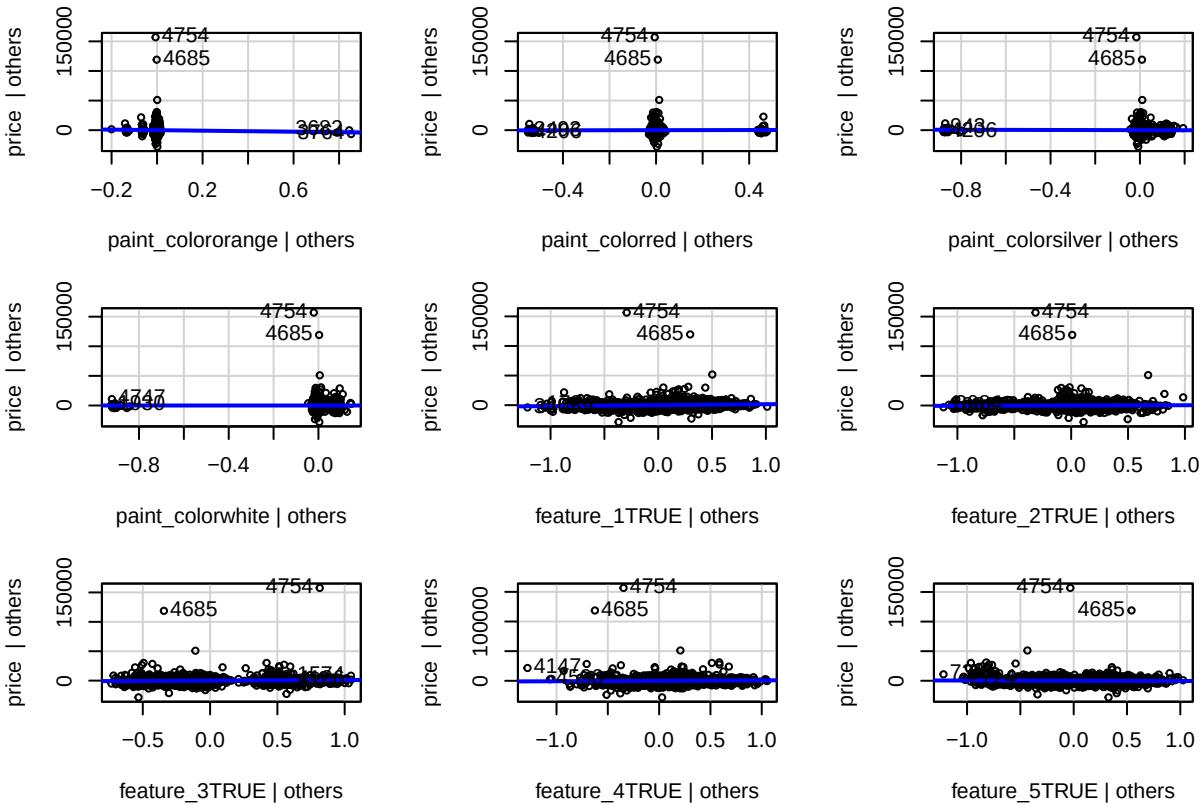

Scale−Location
4754
4685
91
√|Standardized residuals|
0
1
2
3
4
5
−20000
0
20000
40000
60000
Fitted values
lm(price ~ mileage + engine_power + fuel + car_type + diffdate + paint_colo ...

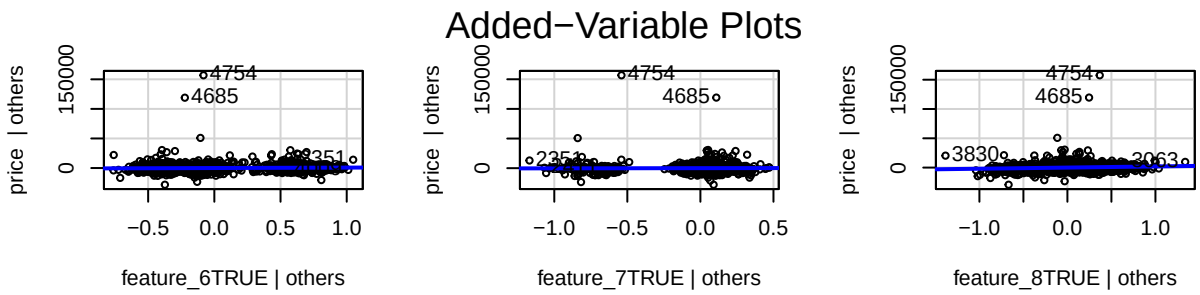



```
avPlots(m.ols)
```


price | others
mileage | others
engine_power | others
fuelelectro | others
fuelhybrid_petrol | others
fuelpetrol | others
car_typecoupe | others
car_typeestate | others
car_typehatchback | others
car_typesedan | others
4754
4685
3063
2351
4147
3602
2318
2364
4010
1024
3277


price | others
150000
0
4754
4685
−0.6
−0.2
0.2
0.6
car_typesubcompact | others
−1.0
−0.6
−0.2
0.2
car_typesuv | others
−0.4
0.0
0.2
0.4
car_typevan | others
2772
−2000
2000
6000
diffdate | others
−1.0
−0.6
−0.2
paint_colorblack | others
−0.8
−0.4
0.0
paint_colorblue | others
paint_colorbrown | others
−0.2
0.2
0.4
0.6
paint_colorgreen | others
−1.0
−0.6
−0.2
paint_colorgrey | others


price | others
paint_colororange | others
paint_colorred | others
paint_colorsilver | others
paint_colorwhite | others
feature_1TRUE | others
feature_2TRUE | others
feature_3TRUE | others
feature_4TRUE | others
feature_5TRUE | others
4754
4685



The reduced model

```
reduced_m <- lm(price ~ mileage + engine_power + fuel + car_type + diffdate + feature_1 + feature_3 + f
summary(reduced_m)
```

```
##
## Call:
## lm(formula = price ~ mileage + engine_power + fuel + car_type +
##     diffdate + feature_1 + feature_3 + feature_6 + feature_8,
##     data = Training_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -27853  -2365   -174   1824 155941
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.346e+04  1.382e+03   9.735  < 2e-16 ***
## mileage             -3.770e-02  2.568e-03 -14.684  < 2e-16 ***
## engine_power         1.089e+02  3.973e+00  27.406  < 2e-16 ***
## fuelelectro          4.511e+03  4.257e+03   1.060  0.28938
## fuelhybrid_petrol    1.274e+04  2.284e+03   5.579 2.70e-08 ***
## fuelpetrol          -2.071e+03  6.661e+02  -3.108  0.00190 **
```

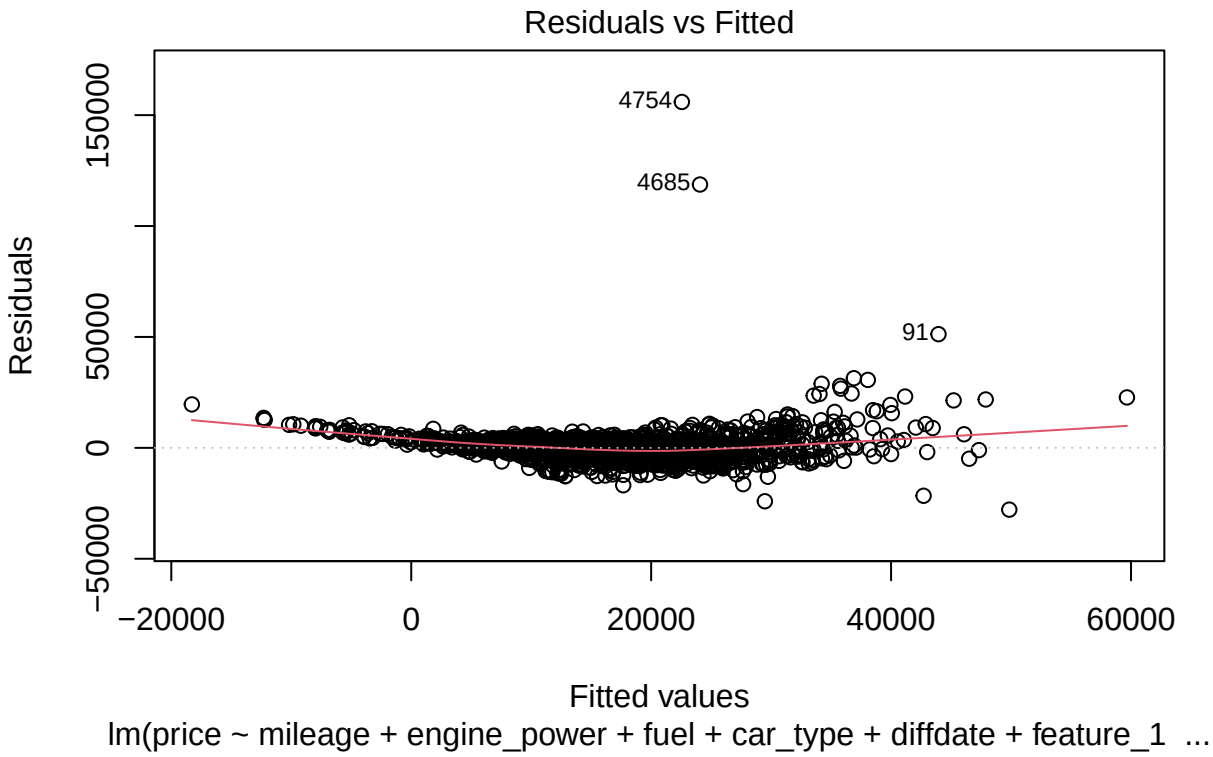
```
## car_typecoupe      -1.652e+03  1.463e+03  -1.129  0.25919
## car_typeestate     -5.024e+03  1.260e+03  -3.986 6.92e-05 ***
## car_typehatchback  -3.695e+03  1.283e+03  -2.880  0.00401 **
## car_typesedan      -2.797e+03  1.260e+03  -2.219  0.02658 *
## car_typesubcompact -3.303e+03  1.465e+03  -2.255  0.02424 *
## car_typesuv        -3.193e+02  1.269e+03  -0.252  0.80134
## car_typevan        -5.647e+03  1.753e+03  -3.220  0.00130 **
## diffdate           -2.781e+00  1.657e-01 -16.787  < 2e-16 ***
## feature_1TRUE       1.715e+03  2.693e+02   6.366 2.31e-10 ***
## feature_3TRUE       1.029e+03  3.266e+02   3.151  0.00165 **
## feature_6TRUE       7.300e+02  3.000e+02   2.434  0.01501 *
## feature_8TRUE       1.928e+03  2.887e+02   6.679 2.98e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5991 on 2415 degrees of freedom
## Multiple R-squared:  0.6322, Adjusted R-squared:  0.6296
## F-statistic: 244.2 on 17 and 2415 DF,  p-value: < 2.2e-16
```

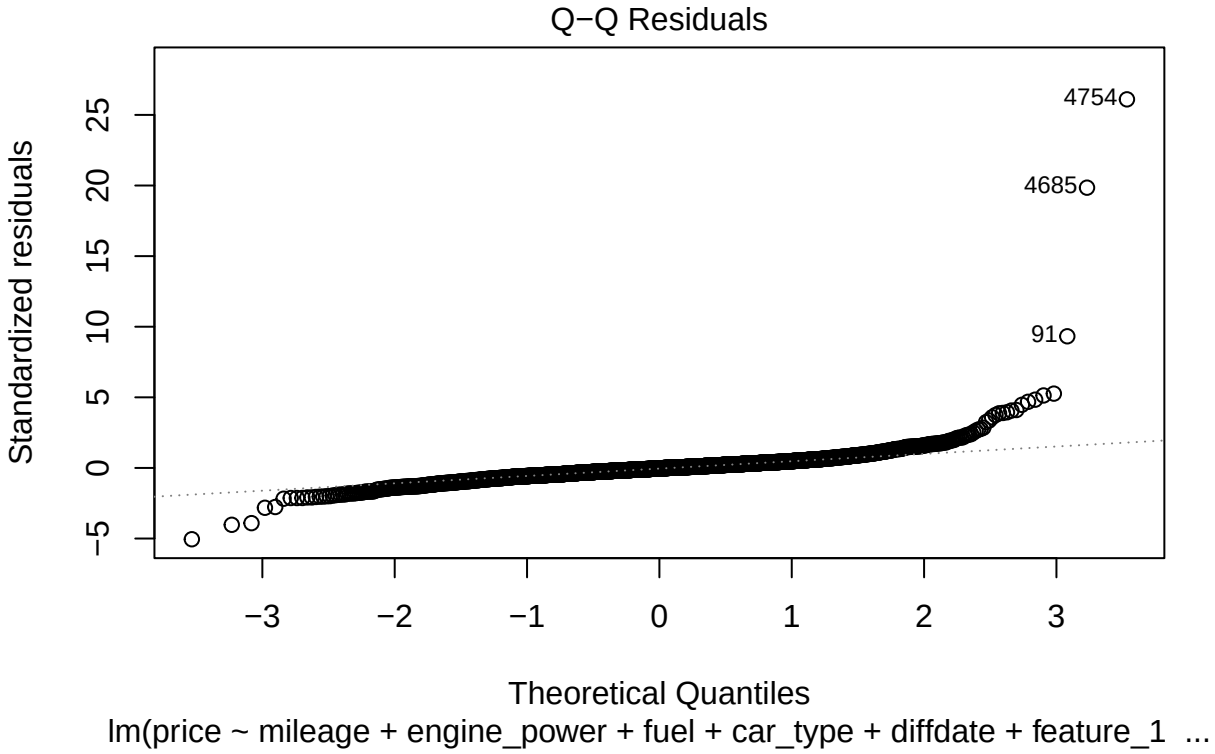
```
Anova(reduced_m, type = "II")
```

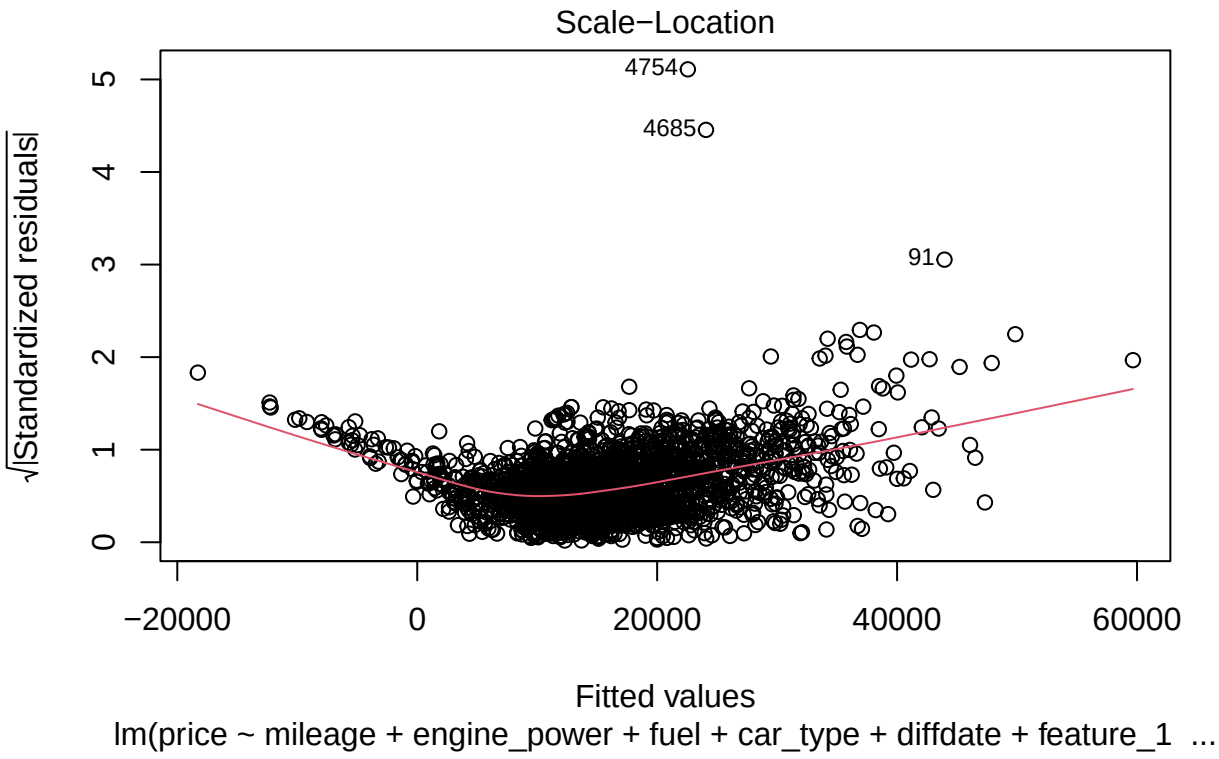
```
## Anova Table (Type II tests)
##
## Response: price
##                  Sum Sq   Df  F value    Pr(>F)
## mileage      7.7394e+09    1 215.6056 < 2.2e-16 ***
## engine_power 2.6962e+10    1 751.1143 < 2.2e-16 ***
## fuel         1.5336e+09    3  14.2414 3.366e-09 ***
## car_type     6.9520e+09    7  27.6672 < 2.2e-16 ***
## diffdate     1.0115e+10    1 281.7905 < 2.2e-16 ***
## feature_1    1.4547e+09    1  40.5258 2.315e-10 ***
## feature_3    3.5648e+08    1   9.9307  0.001645 **
## feature_6    2.1262e+08    1   5.9232  0.015015 *
## feature_8    1.6011e+09    1  44.6029 2.985e-11 ***
## Residuals    8.6689e+10 2415
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

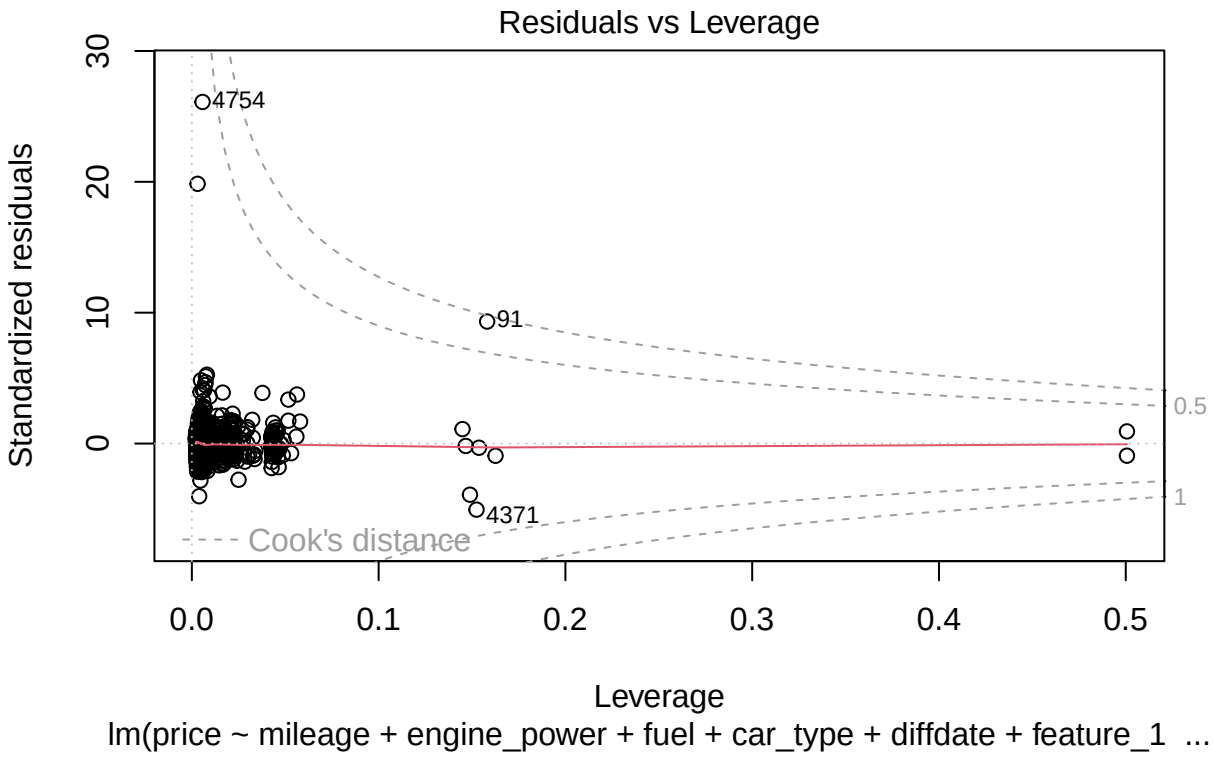
Model diagnostic plots for reduced model

```
plot(reduced_m)
```

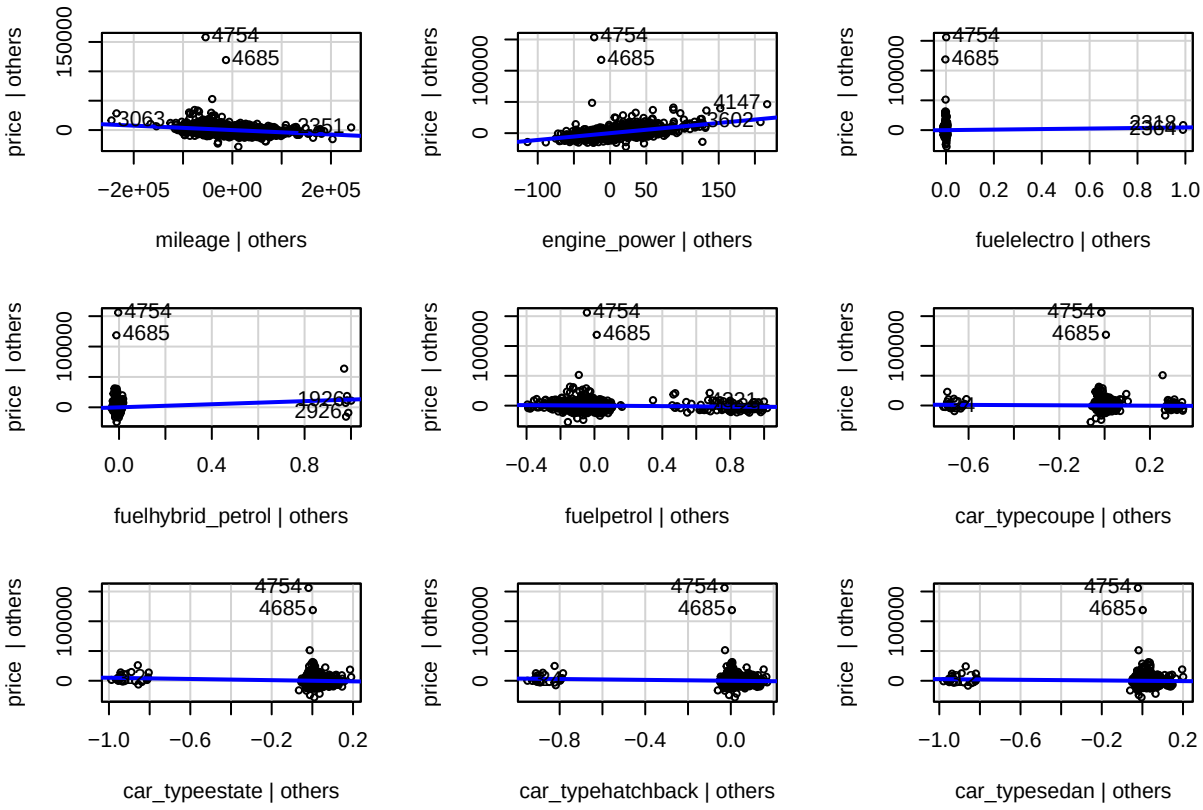

Residuals vs Fitted
Residuals
150000
50000
0
−50000
4754
4685
91
−20000
0
20000
40000
60000
Fitted values
lm(price ~ mileage + engine_power + fuel + car_type + diffdate + feature_1 ...

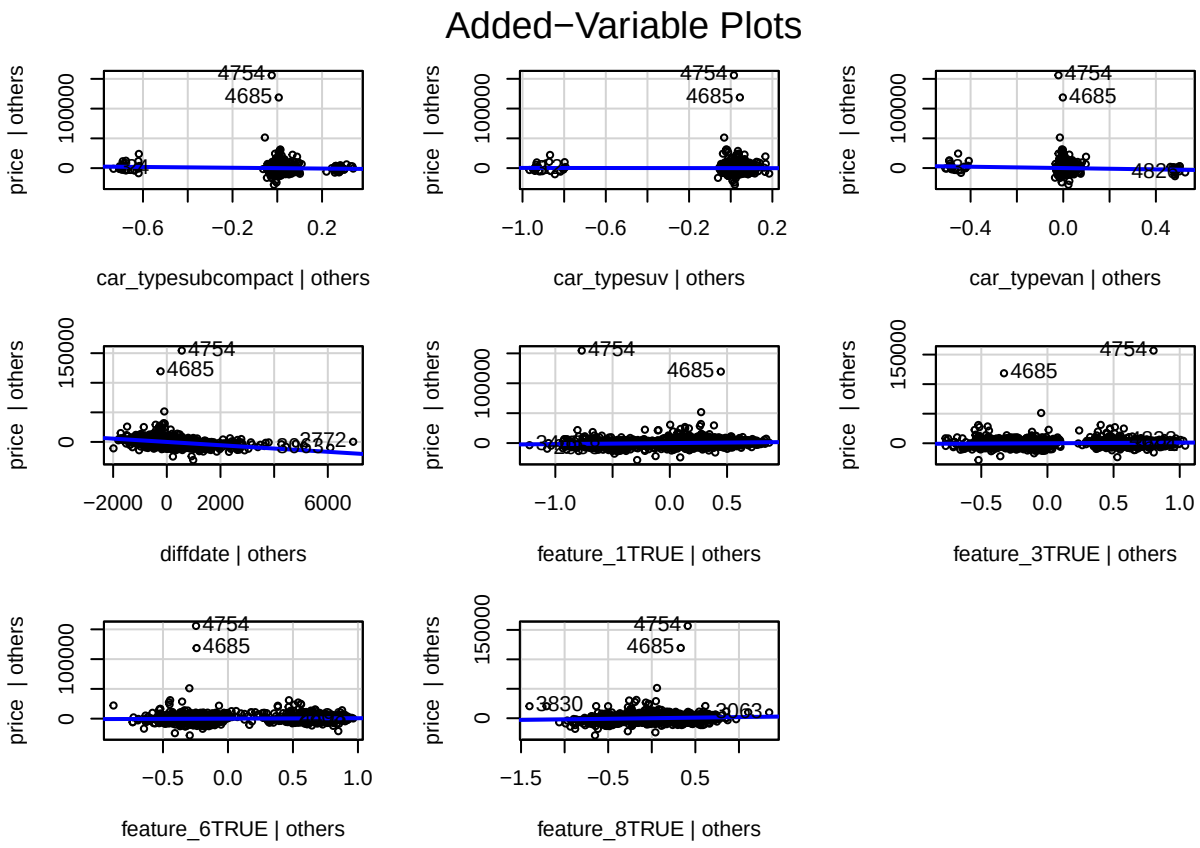

Q−Q Residuals
Standardized residuals
−5
0
5
10
15
20
25
4754
4685
91
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(price ~ mileage + engine_power + fuel + car_type + diffdate + feature_1 ...


Scale−Location
4754
4685
91
√|Standardized residuals|
0
1
2
3
4
5
−20000
0
20000
40000
60000
Fitted values
lm(price ~ mileage + engine_power + fuel + car_type + diffdate + feature_1 ...




```
avPlots(reduced_m)
```


price | others
mileage | others
engine_power | others
fuelelectro | others
fuelhybrid_petrol | others
fuelpetrol | others
car_typecoupe | others
car_typeestate | others
car_typehatchback | others
car_typesedan | others
4754
4685
3063
2351
4147
3602
2926



Perform the partial F-test for the excluded predictors

```
anova(m.ols, reduced_m)
```

```
## Analysis of Variance Table
##
## Model 1: price ~ mileage + engine_power + fuel + car_type + diffdate +
##     paint_color + feature_1 + feature_2 + feature_3 + feature_4 +
##     feature_5 + feature_6 + feature_7 + feature_8
## Model 2: price ~ mileage + engine_power + fuel + car_type + diffdate +
##     feature_1 + feature_3 + feature_6 + feature_8
##   Res.Df        RSS  Df  Sum of Sq      F Pr(>F)
## 1   2402 8.6220e+10
## 2   2415 8.6689e+10 -13 -469858063 1.0069 0.4413
```

Adjusted R-squared for the full model

```
summary(m.ols)$adj.r.squared
```

```
## [1] 0.6296165
```

Adjusted R-squared for the reduced model

```
summary(reduced_m)$adj.r.squared
```

```
## [1] 0.6296027
```

AIC and BIC for two models

```
AIC(m.ols, reduced_m)
```

```
##           df      AIC
## m.ols     32 49262.08
## reduced_m 19 49249.31
```

```
BIC(m.ols, reduced_m)
```

```
##           df      BIC
## m.ols     32 49447.58
## reduced_m 19 49359.45
```

Stepwise regression

```
stepwise_model <- step(reduced_m, direction = "both")
```

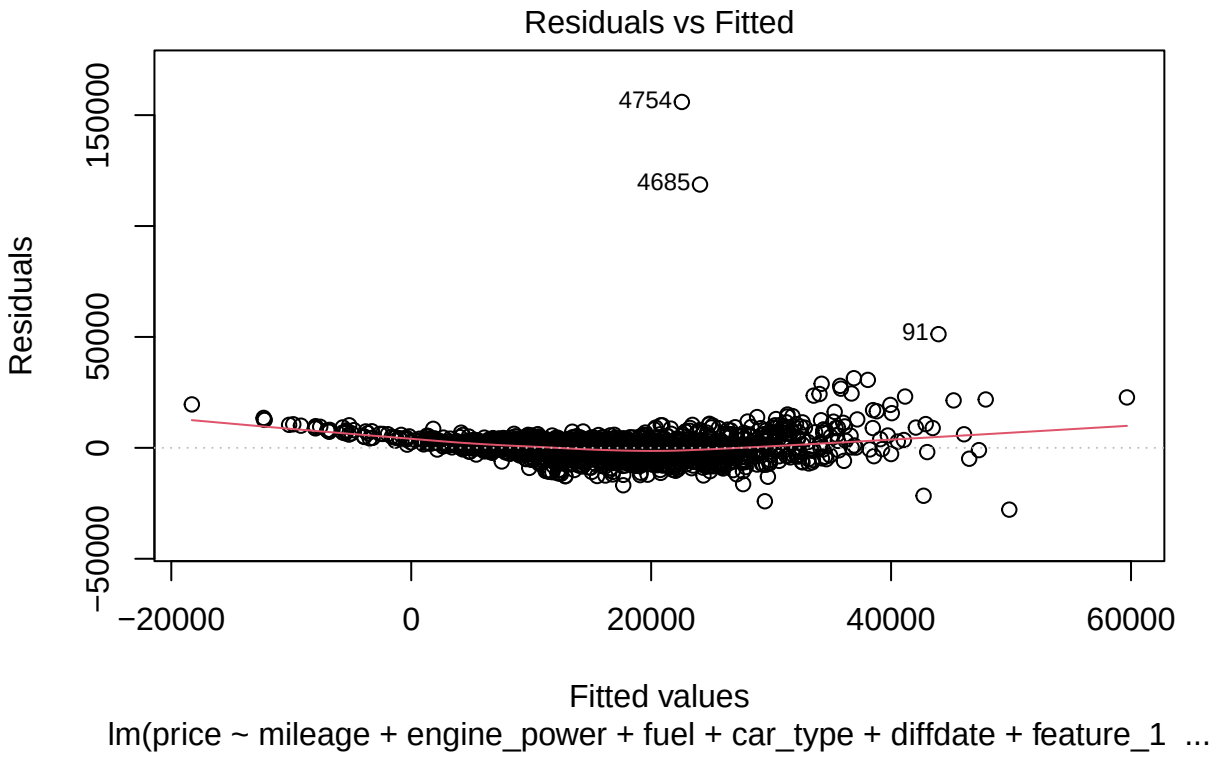
```
## Start:  AIC=42342.75
## price ~ mileage + engine_power + fuel + car_type + diffdate +
##     feature_1 + feature_3 + feature_6 + feature_8
##
##                Df  Sum of Sq        RSS   AIC
## <none>                       8.6689e+10 42343
## - feature_6     1 2.1262e+08 8.6902e+10 42347
## - feature_3     1 3.5648e+08 8.7046e+10 42351
## - fuel          3 1.5336e+09 8.8223e+10 42379
## - feature_1     1 1.4547e+09 8.8144e+10 42381
## - feature_8     1 1.6011e+09 8.8291e+10 42385
## - car_type      7 6.9520e+09 9.3642e+10 42516
## - mileage       1 7.7394e+09 9.4429e+10 42549
## - diffdate      1 1.0115e+10 9.6805e+10 42609
## - engine_power  1 2.6962e+10 1.1365e+11 43000
```

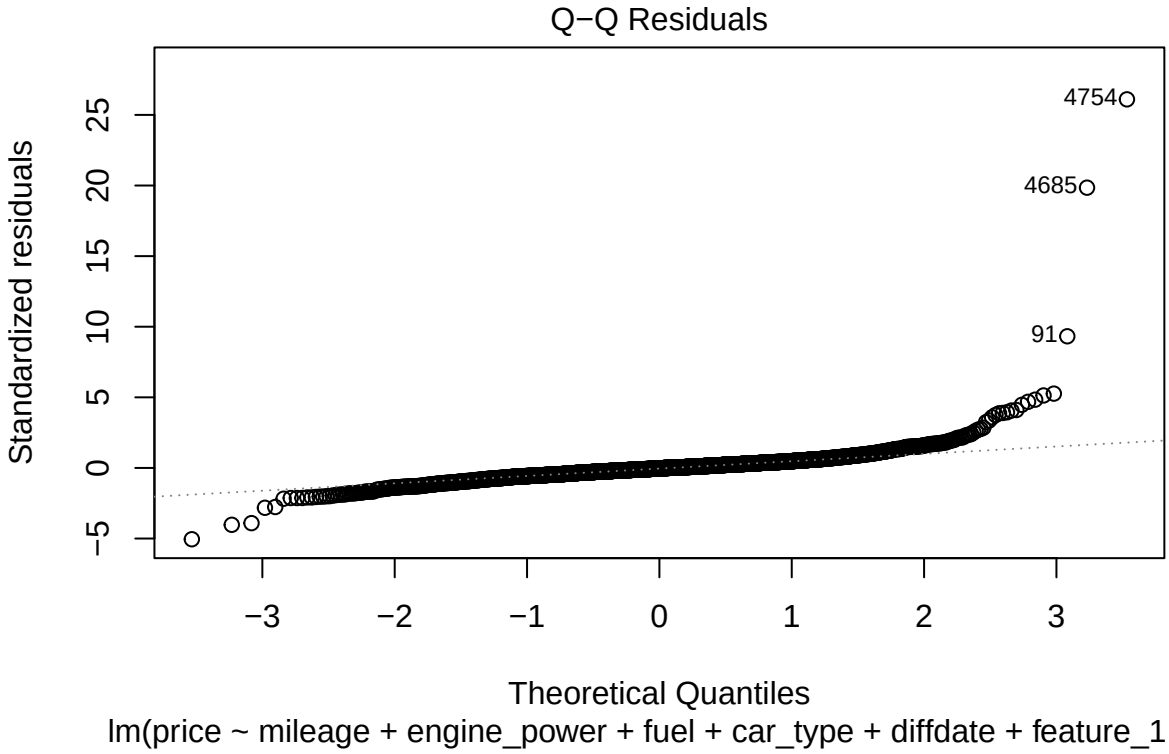
```
summary(stepwise_model)
```

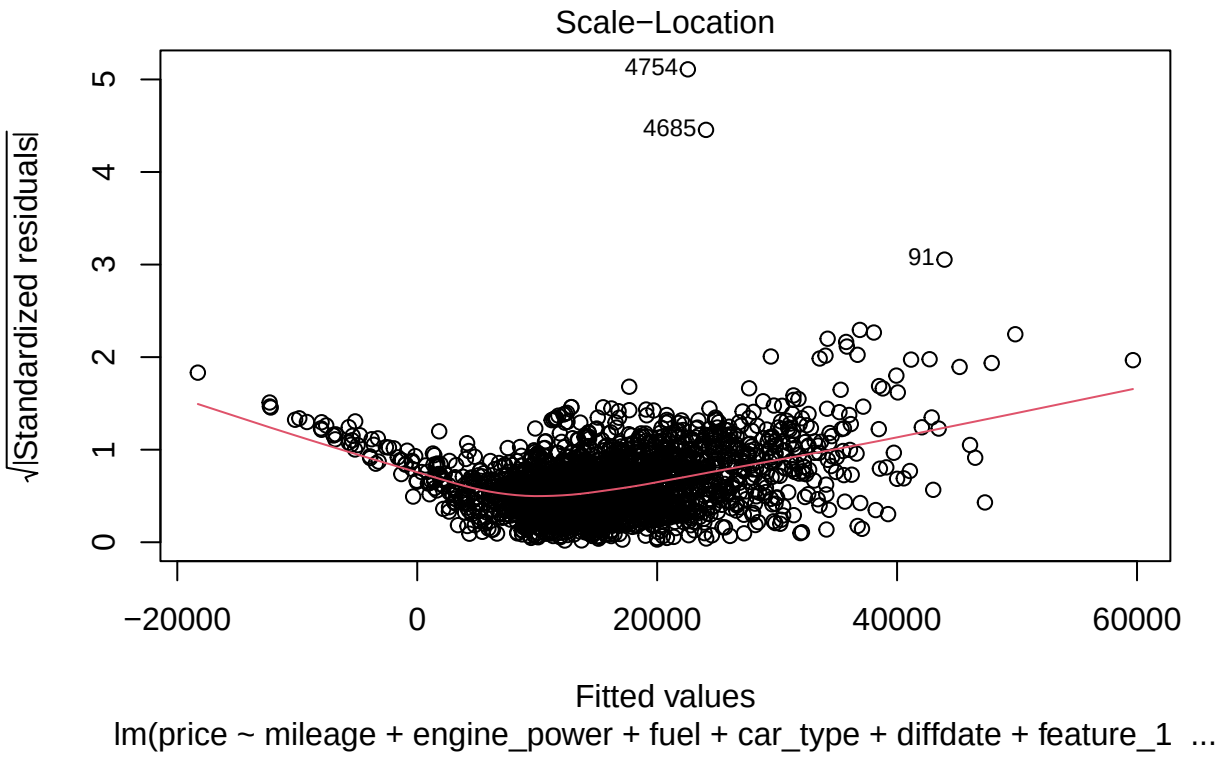
```
##
## Call:
## lm(formula = price ~ mileage + engine_power + fuel + car_type +
##     diffdate + feature_1 + feature_3 + feature_6 + feature_8,
```

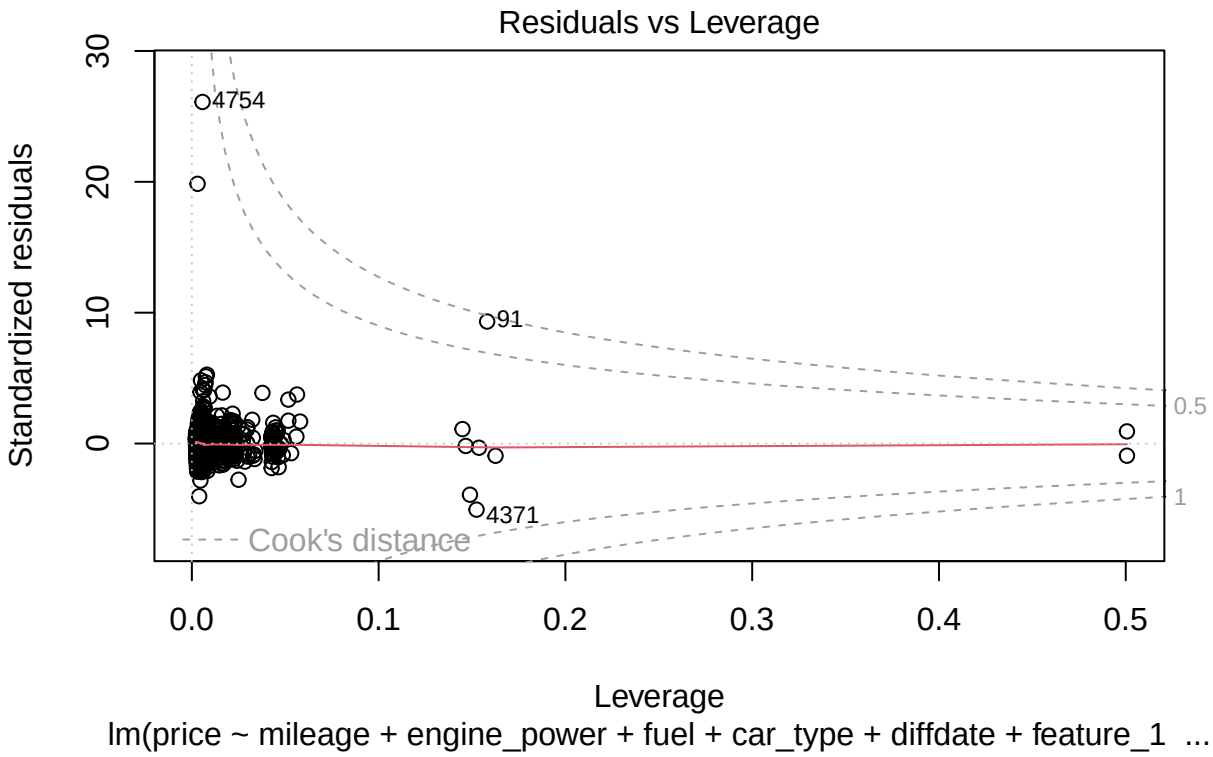
```
##     data = Training_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -27853  -2365   -174   1824 155941
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.346e+04  1.382e+03   9.735  < 2e-16 ***
## mileage             -3.770e-02  2.568e-03 -14.684  < 2e-16 ***
## engine_power         1.089e+02  3.973e+00  27.406  < 2e-16 ***
## fuelelectro          4.511e+03  4.257e+03   1.060  0.28938
## fuelhybrid_petrol    1.274e+04  2.284e+03   5.579 2.70e-08 ***
## fuelpetrol          -2.071e+03  6.661e+02  -3.108  0.00190 **
## car_typecoupe       -1.652e+03  1.463e+03  -1.129  0.25919
## car_typeestate      -5.024e+03  1.260e+03  -3.986 6.92e-05 ***
## car_typehatchback   -3.695e+03  1.283e+03  -2.880  0.00401 **
## car_typesedan       -2.797e+03  1.260e+03  -2.219  0.02658 *
## car_typesubcompact -3.303e+03  1.465e+03  -2.255  0.02424 *
## car_typesuv         -3.193e+02  1.269e+03  -0.252  0.80134
## car_typevan         -5.647e+03  1.753e+03  -3.220  0.00130 **
## diffdate            -2.781e+00  1.657e-01 -16.787  < 2e-16 ***
## feature_1TRUE        1.715e+03  2.693e+02   6.366 2.31e-10 ***
## feature_3TRUE        1.029e+03  3.266e+02   3.151  0.00165 **
## feature_6TRUE        7.300e+02  3.000e+02   2.434  0.01501 *
## feature_8TRUE        1.928e+03  2.887e+02   6.679 2.98e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5991 on 2415 degrees of freedom
## Multiple R-squared:  0.6322, Adjusted R-squared:  0.6296
## F-statistic: 244.2 on 17 and 2415 DF,  p-value: < 2.2e-16
```

```
plot(stepwise_model)
```

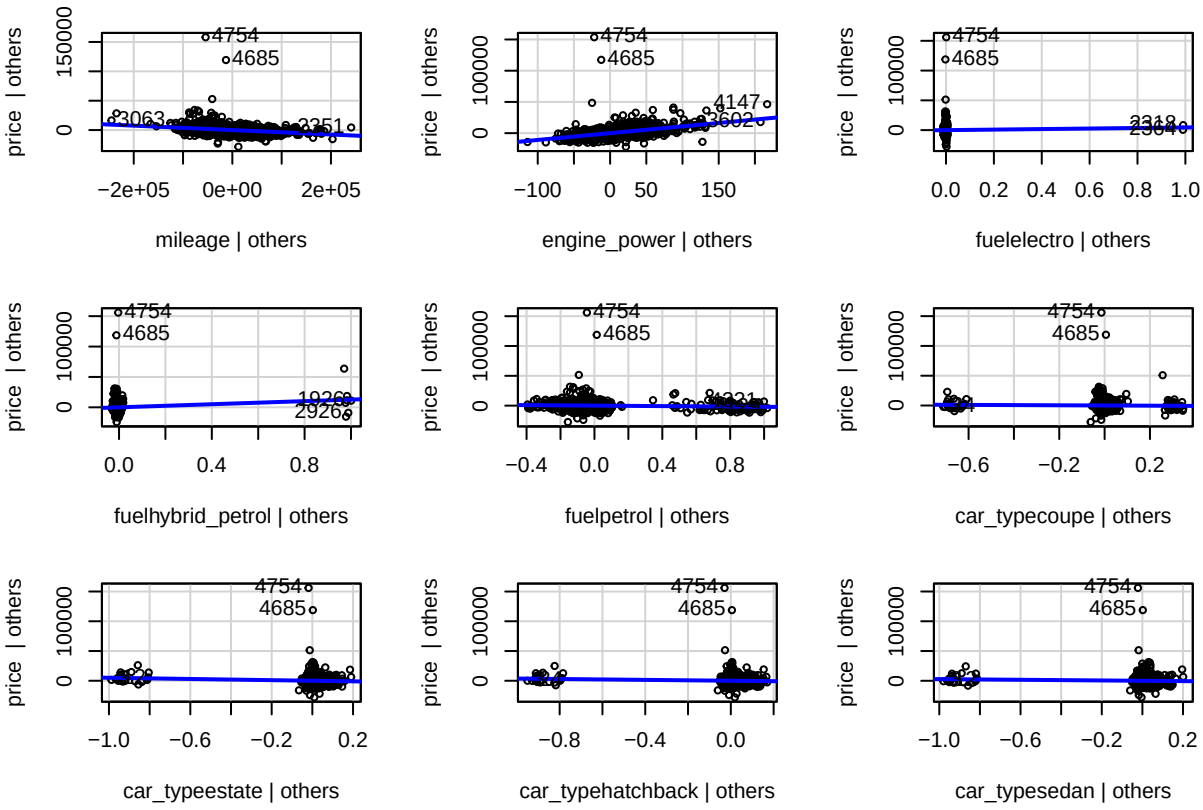

Residuals vs Fitted
Residuals
150000
50000
0
−50000
4754
4685
91
−20000
0
20000
40000
60000
Fitted values
lm(price ~ mileage + engine_power + fuel + car_type + diffdate + feature_1 ...

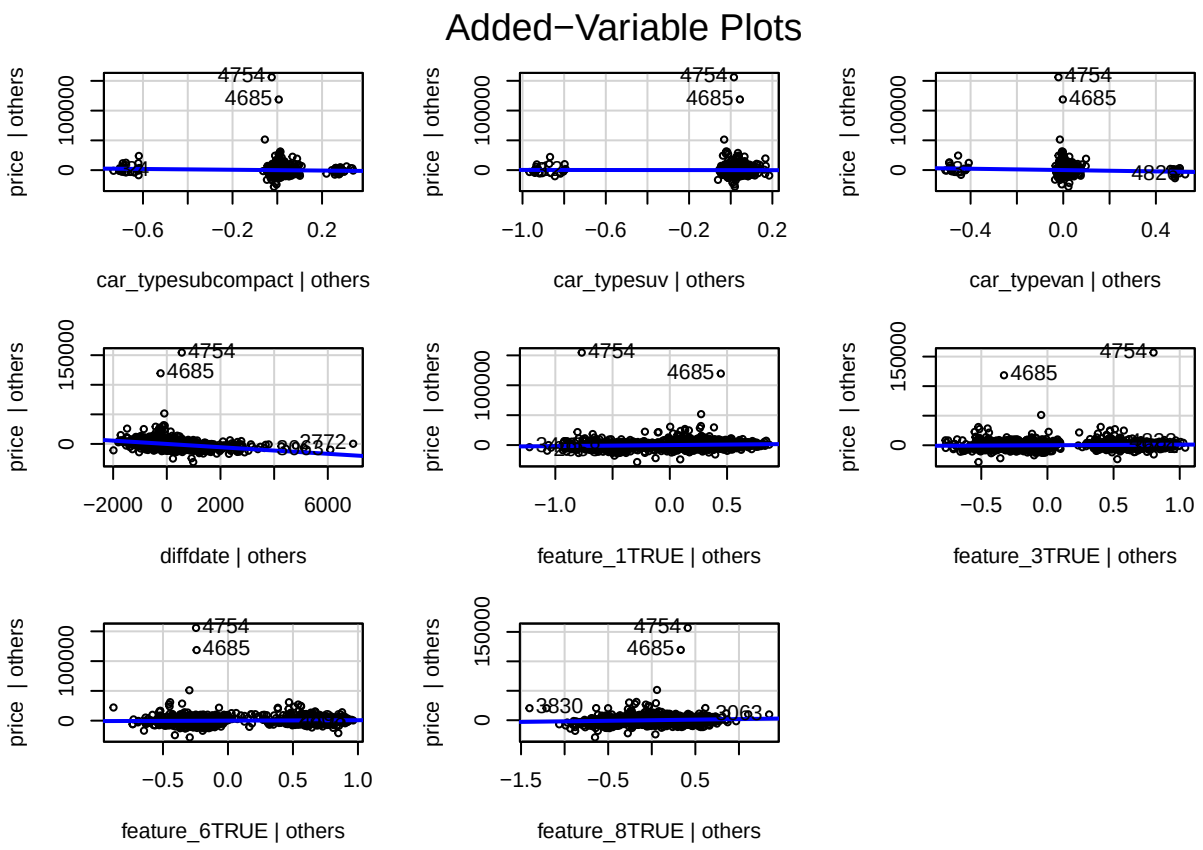

Q−Q Residuals
Standardized residuals
−5
0
5
10
15
20
25
4754
4685
91
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(price ~ mileage + engine_power + fuel + car_type + diffdate + feature_1 ...


Scale−Location
4754
4685
91
√|Standardized residuals|
0
1
2
3
4
5
−20000
0
20000
40000
60000
Fitted values
lm(price ~ mileage + engine_power + fuel + car_type + diffdate + feature_1 ...




```
avPlots(stepwise_model)
```


price | others
mileage | others
engine_power | others
fuelelectro | others
fuelhybrid_petrol | others
fuelpetrol | others
car_typecoupe | others
car_typeestate | others
car_typehatchback | others
car_typesedan | others
4754
4685




```
vif(stepwise_model)
```

```
##                  GVIF Df GVIF^(1/(2*Df))
## mileage      1.524239  1        1.234601
## engine_power 1.733778  1        1.316730
## fuel         1.212003  3        1.032565
## car_type     1.482990  7        1.028547
## diffdate     1.601648  1        1.265562
## feature_1    1.211684  1        1.100765
## feature_3    1.196035  1        1.093634
## feature_6    1.101908  1        1.049718
## feature_8    1.403305  1        1.184612
```

```
summary(stepwise_model)$adj.r.squared
```

```
## [1] 0.6296027
```

```
AIC(stepwise_model)
```

```
## [1] 49249.31
```

```
BIC(stepwise_model)
```

```
## [1] 49359.45
```

Ridge Regression

```
y <- Training_data$sqrt_price
x <- data.matrix(Training_data[, c('mileage', 'engine_power', 'fuel', 'car_type', 'diffdate', 'feature_
```

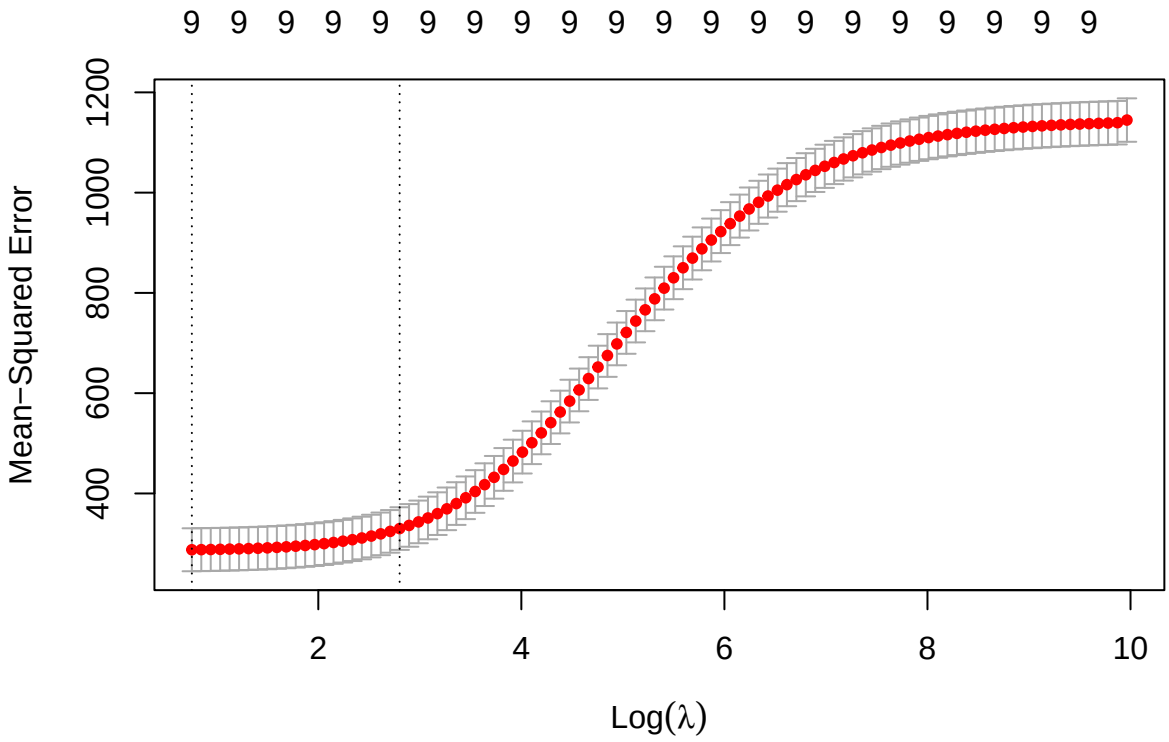
```
model <- glmnet(x, y, alpha = 0)

summary(model)
```

```
##           Length Class     Mode
## a0        100    -none-    numeric
## beta      900    dgCMatrix S4
## df        100    -none-    numeric
## dim         2    -none-    numeric
## lambda    100    -none-    numeric
## dev.ratio 100    -none-    numeric
## nulldev     1    -none-    numeric
## npasses     1    -none-    numeric
## jerr        1    -none-    numeric
## offset      1    -none-    logical
## call        4    -none-    call
## nobs        1    -none-    numeric
```

```
#perform k-fold cross-validation to find optimal lambda value
cv_model <- cv.glmnet(x, y, alpha = 0)

#find optimal lambda value that minimizes test MSE
best_lambda <- cv_model$lambda.min
best_lambda
```

```
## [1] 2.126533
```

```
#produce plot of test MSE by lambda value
plot(cv_model)
```

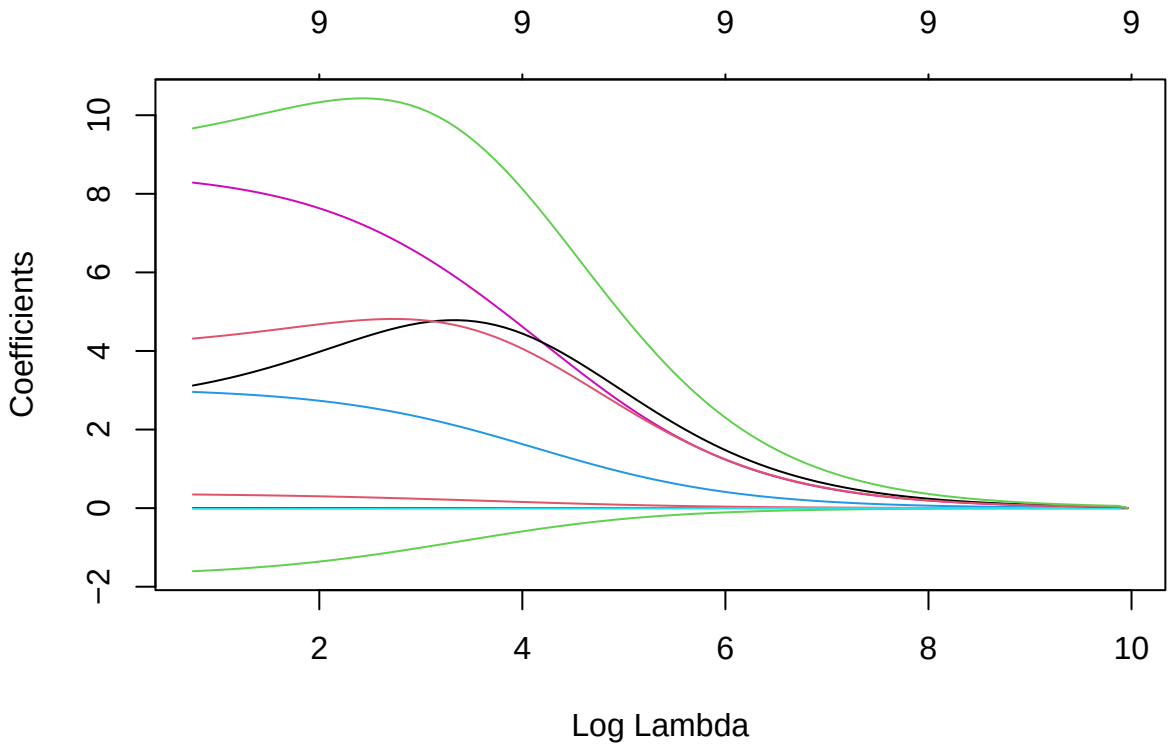



```
best_model <- glmnet(x, y, alpha = 0, lambda = best_lambda)
coef(best_model)
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                            s0
## (Intercept)  98.0097022568
## mileage      -0.0001321719
## engine_power  0.3468269276
## fuel         -1.6041527254
## car_type      2.9580224794
## diffdate     -0.0129823951
## feature_1     8.2852282515
## feature_3     3.1208314633
## feature_6     4.3162317470
## feature_8     9.6674736203
```

Produce Ridge trace plot

```
plot(model, xvar = "lambda")
```




```
y_predicted <- predict(model, s = best_lambda, newx = x)

sst <- sum((y - mean(y))^2)
sse <- sum((y_predicted - y)^2)

rsq <- 1 - sse/sst
rsq
```

```
## [1] 0.7512921
```

```
adj_rsq <- 1-(1-rsq)*2432/2422
adj_rsq
```

```
## [1] 0.7502653
```

```
final_model <- lm(sqrt(price) ~ mileage + engine_power + fuel + car_type +
    diffdate + feature_1 + feature_3 + feature_6 + feature_8,
    data = Training_data)
```

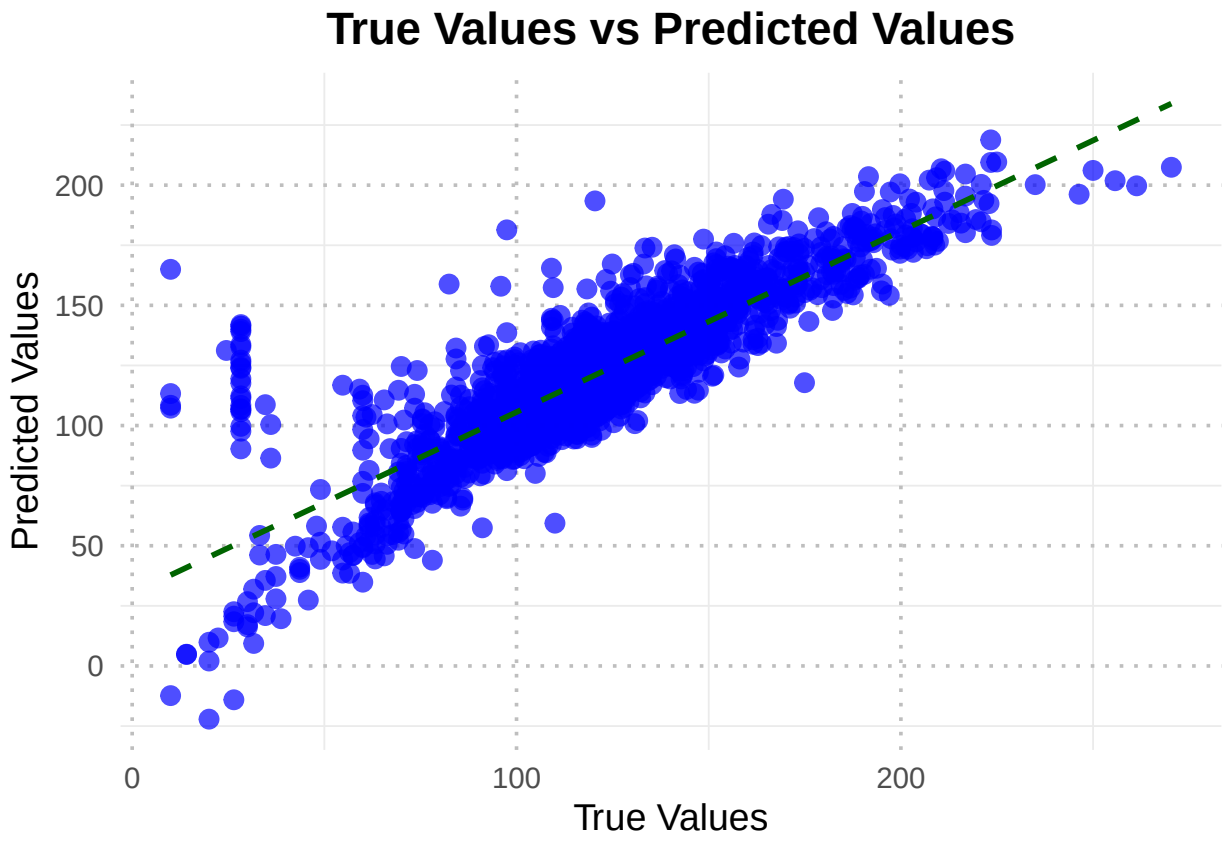
Prediction

```r
Validation_data$predicted_price <- predict(final_model, newdata = Validation_data)
```

Scatter Plot

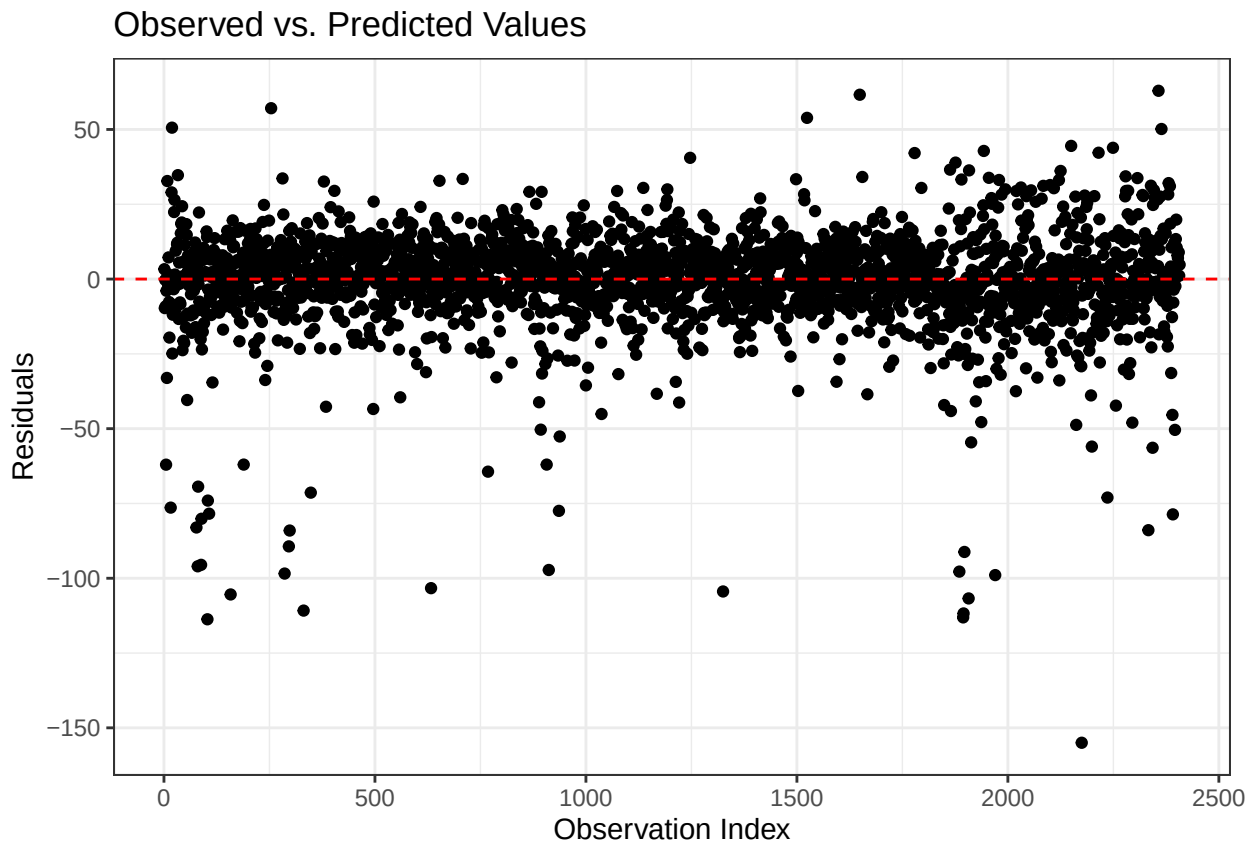
```r
ggplot(Validation_data, aes(x = sqrt_price, y = predicted_price)) +
  geom_point(color = "blue", size = 3, alpha = 0.7) +  # Points with transparency
  geom_smooth(method = "lm", se = FALSE, color = "darkgreen", linetype = "dashed") +  # Linear trend li
  labs(
    title = "True Values vs Predicted Values",
    x = "True Values",
    y = "Predicted Values"
  ) +
  theme_minimal(base_size = 14) +  # Minimal theme with larger font
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    panel.grid.major = element_line(color = "gray", linetype = "dotted")
  )
```

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



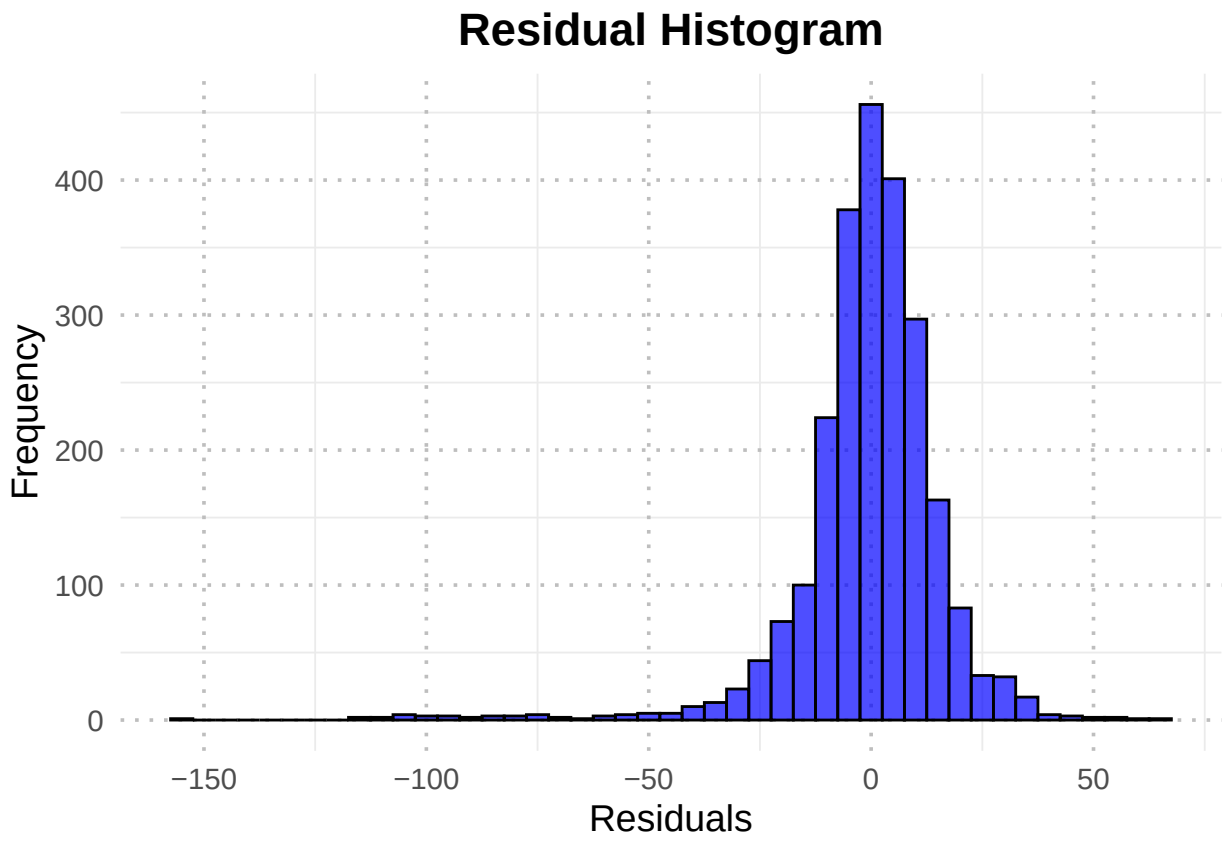
Residual plot

```
ggplot(Validation_data, aes(x = 1:nrow(Validation_data), y = sqrt_price-predicted_price)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 0, color = "red", linetype = "dashed") +
  labs(x = "Observation Index", y = "Residuals",
       title = "Observed vs. Predicted Values") +
  theme_bw()
```



Residual plot

```
ggplot(Validation_data, aes(x = sqrt_price-predicted_price)) +
  geom_histogram(binwidth = 5, fill = "blue", color = "black", alpha = 0.7) +  # Histogram bars
  labs(
    title = "Residual Histogram",
    x = "Residuals",
    y = "Frequency"
  ) +
  theme_minimal(base_size = 14) +  # Minimal theme with larger font
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    panel.grid.major = element_line(color = "gray", linetype = "dotted")
  )
```



Measure the performance of the final model

```
observed_values <- Validation_data$sqrt_price
predicted_values <- Validation_data$predicted_price
```

```
rmse <- RMSE(predicted_values, observed_values)

mae <- MAE(predicted_values, observed_values)

mape <- MAPE(predicted_values, observed_values)

r_squared <- R2_Score(predicted_values, observed_values)

cat("Root Mean Squared Error (RMSE):", round(rmse, digits = 4))
```

```
## Root Mean Squared Error (RMSE): 16.8422
```

```
cat("\nMean Absolute Error (MAE):", round(mae, digits = 4))
```

```
##
## Mean Absolute Error (MAE): 10.6664
```

```r
cat("\nR-squared (R^2) Score:", round(r_squared, digits = 4))
```

```
##
## R-squared (R^2) Score: 0.7404
```

```r
cat("\nMean Absolute Percentage Error (MPE):", round(mape, digits = 4))
```

```
##
## Mean Absolute Percentage Error (MPE): 0.1381
```