VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY

HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY

PROBABILITY AND STATISTICS

PROJECT REPORT

STATISTICAL ANALYSIS AND FORECASTING OF SPORTS CAR PRICES

Lecturer	**MSc Phan Thi Khanh Van**		
Semester	**251 - Class: CC05 - Group: 8**		
Name	**ID**	**Task Assignment**	**Work Attitude**
Hoang Ngoc Tu Anh	2452042	Synthesis + Chap.2,4	Good
Le Nhat Anh	2452103	Chapter 5	Good
Pham Gia Khiem	2352545	Chapter 6	Good
Nguyen Viet Khoa	2452549	Chapter 7,8	Good
Phan Ha Phuong	2453035	Chapter 3	Good

Ho Chi Minh City,2025

Table of Contents

ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to Ms. Phan Thi Khanh Van, our lecturer for the Probability and Statistics course, for her dedicated teaching and guidance throughout the semester. We also highly appreciate the collaborative spirit within our group and the valuable contributions of each individual member. This report is the result of our collective effort and continuous learning, built upon the knowledge we have acquired from lectures and coursework. Conducting this research has not only deepened our understanding of the subject and its practical applications, but also provided an opportunity to develop essential skills in teamwork, critical thinking, and academic reporting.

I. CHAPTER 1: DATA & CODE AVAILABILITY

1. Dataset

Dataset

2. Code R

Code R

II. CHAPTER 2: RESEARCH OVERVIEW

The Elite Sports Cars dataset includes 5,000 synthetic sports car models, capturing variables such as engine performance, fuel efficiency, CO2 emissions, usage conditions, insurance cost, market popularity, and sale price.

The study explores the relationships between vehicle specifications (Engine Size, Horsepower, Torque), market factors (Popularity, Market Demand), vehicle condition (Condition, Number of Owners), and Price. It also compares cars from Asia, USA, and Europe based on average pricing.

The research follows three stages:

1. **Data Preprocessing**: Removing unnecessary variables, inspecting missing values, detecting outliers, and converting categorical variables.
2. **Descriptive Statistics**: Summarizing data with mean, standard deviation, and visualizations (histograms, boxplots, scatter plots, correlation matrix).
3. **Inferential Statistics**: Conducting hypothesis tests, ANOVA, Kruskal-Wallis, and post-hoc analysis (Wilcoxon, Tukey HSD, Dunn).
4. **Predictive Modeling**: Using Linear Regression with stepwise selection and Random Forest, evaluated by MAE, RMSE, and R^2.

The study aims to predict Price based on technical and market variables. The Linear Regression model provides point predictions with confidence intervals, while the Random Forest model captures nonlinear relationships. Both models are evaluated for reliability and predictive quality.

Analysis reveals that Price is weakly correlated with technical specifications. Statistical tests show no significant differences in Price across categories like Country, Fuel Type, or Condition. The Random Forest model improves prediction but is limited by the dataset structure. Visualizations highlight weak relationships, emphasizing the importance of evaluating data structure before modeling.

III. CHAPTER 3: THEORETICAL BACKGROUND

1. Proposed Statistical Model and Rationale

1.1. Multiple Linear Regression Model

Many applications of regression analysis involve situations that have more than one regressor or predictor variable. A regression model that contains more than one regressor variable is called a multiple regression model.

As an example, suppose that the gasoline mileage performance of a vehicle depends on the vehicle weight and the engine displacement. A multiple regression model that might describe this relationship is

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon \quad (3.1)$$

where Y represents the mileage, x_1 represents the weight, x_2 represents the engine displacement, and ϵ is a random error term. This is a multiple linear regression model with two regressors.

In general, the dependent variable or response Y may be related to k independent or regressor variables. The model

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \epsilon \quad (3.2)$$

is called a multiple linear regression model with $\boldsymbol{k}$ regressor variables. The parameters $\beta_j, j = 0, 1, \ldots, k$, are called the regression coefficients. This model describes a hyperplane in the $k+1$ - dimensional space of Y and the regressor variables $\{x_j\}$. The parameter β_j represents the expected change in response Y per unit change in x_j when all the remaining regressors x_i $(i \neq j)$ are held constant.

Multiple linear regression models are often used as approximating functions. That is, the true functional relationship between Y and $x_1, x_2, \ldots, x_k$ is unknown, but over certain ranges of the independent variables, the linear regression model is an adequate approximation.

Models that are more complex in structure than Equation 3.2 may often still

be analyzed by multiple linear regression techniques. For example, consider the cubic polynomial model in one regressor variable

$$Y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \epsilon \quad (3.3);$$

if we let $x_1 = x$, $x_2 = x^2$, $x_3 = x^3$, Equation 3.3 can be written as

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \epsilon \quad (3.4)$$

which is a multiple linear regression model with three regressor variables.

Models that include interaction effects may also be analyzed by multiple linear regression methods. Interaction effects are very common. For example, a vehicle's mileage may be impacted by an interaction between vehicle weight and engine displacement. An interaction between two variables can be represented by a cross-product term in the model, such as

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \epsilon \quad (3.5);$$

if we let $x_3 = x_1 x_2$ and $\beta_3 = \beta_{12}$, Equation 3.5 can be written as

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \epsilon$$

which is a linear regression model.

1.2. Rationale for Selection

The multiple linear regression model was chosen because it is well-suited to the characteristics of the dependent variable (Price, Log_Price), which is continuous. This standard method allows for an effective description of the relationship between a continuous dependent variable and multiple independent variables. The model enables the estimation of the expected value of Price based on input factors, determines the influence of each explanatory variable on Price, and assesses the model's goodness-of-fit through statistical measures such as R^2, the F-test, and p-values.

Moreover, the model's linear structure with respect to its parameters facilitates a clear and intuitive interpretation of the results. Each regression coef-

ficient represents the magnitude and direction of the effect of an independent variable on the dependent variable, holding all other factors constant.

Additionally, the model maintains good interpretability even when extended to include quadratic terms or interaction effects, allowing for a more flexible representation of mild nonlinear relationships in the data without compromising transparency.

Furthermore, multiple linear regression is supported by a comprehensive system of statistical tests (e.g., significance testing of coefficients, VIF analysis, residual diagnostics), which ensures the reliability and adequacy of the model for the given data.

2. Analytical Tools and Statistical Methods

2.1. Pearson Correlation Analysis

A linear correlation between two variables occurs when their observed values, plotted on a Cartesian plane, tend to align along a straight line. The Pearson correlation coefficient (r) is commonly used to quantify the strength of this linear relationship for two quantitative variables (Gaven, 1951[1]). Pearson correlation is not applicable if one or both variables are non-quantitative (e.g., categorical or binary).

r ranges from -1 to 1 - values near ± 1 indicate a strong linear relationship (positive for $+1$, negative for -1), while values near 0 indicate a weak or absent linear relationship. A perfect linear correlation ($r = 1$) produces a straight line in a scatter plot; $r = 0$ suggests no linear correlation, although a nonlinear relationship may exist.

Statistical significance of r is assessed via a t-test under the null hypothesis H_0: $r = 0$ (Field, 2009). Interpretation is as follows:

- Sig < 0.05: Reject H_0: a statistically significant linear correlation exists.
- Sig ≥ 0.05: Fail to reject H_0: no statistically significant linear correlation exists.

The strength of a statistically significant correlation is evaluated by $|r|$:

- $|r| < 0.1$: very weak
- $0.1 \leq |r| < 0.3$: weak
- $0.3 \leq |r| < 0.5$: moderate
- $|r| \geq 0.5$: strong

2.2. Analysis of Variance (ANOVA) and Hypothesis Testing

Analysis of Variance (ANOVA) is a statistical method used to compare the means of two or more groups to determine if there are significant differences between them. ANOVA divides the total variance into components, such as within-group variance and between-group variance.

ANOVA can be categorized into three main types: One-Way ANOVA is used when there is one independent factor affecting the dependent variable; Two-Way ANOVA is used when there are two independent factors affecting the dependent variable; and N-way ANOVA is used when there are multiple independent factors.

The method of ANOVA has wide applications in many fields. In engineering, ANOVA is used to compare the performance of different materials, products, or manufacturing processes. In medicine, ANOVA helps assess the effectiveness of treatments or drugs on different groups of patients. In psychology, ANOVA is applied to study the impact of factors such as personality, environment, and genetics on behavior. In social sciences, ANOVA is used to compare groups of people based on demographic, social, or economic variables.

To perform ANOVA, certain assumptions must be met: normality (data within each group must follow a normal distribution), homogeneity of variance (the variance within groups must be equal), independence (observations must be independent), and randomness (samples must be randomly selected).

The procedure for performing ANOVA involves setting the null hypothesis

(H_0), which states that there is no difference between the group means, and the alternative hypothesis (H_1), which states that at least one group has a significantly different mean. After conducting the F-test, if the p-value is smaller than the significance level, the null hypothesis is rejected, indicating that there is a difference between the groups. If differences are found, post-hoc analysis will be conducted to identify which groups differ significantly.

2.3. Kruskal–Wallis and Dunn's Post-hoc Tests

The Kruskal-Wallis test is a nonparametric statistical procedure used to determine whether there are statistically significant differences among multiple independent groups on a common dependent variable. It serves as an appropriate alternative to the one-way ANOVA, particularly when the fundamental assumptions of normality or homogeneity of variance are not satisfied. Accordingly, the Kruskal-Wallis test is frequently applied when the dependent variable is ordinal or when sample sizes are small, ensuring that the analysis remains robust and reliable despite violations of parametric assumptions.

The Dunn's Post-hoc test is a statistical procedure used to compare multiple pairs of means (averages) in a group of data. It is often used after conducting a statistical test that compares means, such as an analysis of variance (ANOVA). The purpose of the Dunn-Bonferroni test is to identify which pairs of means are significantly different from each other. The Dunn test works by adjusting the alpha level (the level of statistical significance) to account for the number of pairs of means being compared. This is necessary because the more pairs of means that are compared, the greater the chance that a difference between means will be found simply by chance. By adjusting the alpha level, the Dunn-Bonferroni test helps to control for this problem, known as the "multiple comparisons problem."

2.4. Stepwise Selection

Stepwise Selection is a regression model–building technique used to identify an optimal set of independent variables through a sequential process of adding

or removing predictors based on their statistical significance. The method operates iteratively, employing tests such as the t-test and F-test to evaluate the contribution of each variable to the model's explanatory power. Stepwise Selection can be implemented through three main approaches: Forward Selection, which begins with no predictors and incrementally adds statistically significant variables; Backward Elimination, which starts with all potential predictors and progressively removes those that are not significant; and Bidirectional Elimination, which combines both adding and removing variables at each step. With the support of statistical software, this method facilitates automated model selection and is particularly useful for datasets containing a large number of candidate predictors. Nevertheless, despite its convenience, Stepwise Selection has notable limitations, including the risk of selecting models driven by random variation, which may lead to instability or spurious associations. Therefore, its application requires careful consideration in empirical analysis.

2.5. Diagnostic Plots

Diagnostic plots play a vital role in evaluating the adequacy and reliability of statistical and machine-learning models. These visual tools help assess key underlying assumptions, including linearity, normality of residuals, homoscedasticity, and the absence of influential observations. Ensuring that these assumptions hold is essential for obtaining unbiased estimates and valid statistical inference.

Four commonly used diagnostic plots include:

Residuals vs. Fitted Plot
This plot examines the linearity assumption by displaying whether residuals scatter randomly around zero. A random pattern indicates a well-specified model, whereas systematic shapes or curvature suggest unmet nonlinear relationships.

Normal Q–Q Plot
The Q–Q plot evaluates whether residuals follow a normal distribution. Points

aligning closely with the reference line indicate normality, while deviations signify distributional irregularities such as skewness or outliers.

Scale – Location Plot

This plot assesses homoscedasticity by visualizing the spread of standardized residuals across fitted values. A consistent spread supports constant variance; increases or decreases in spread indicate heteroscedasticity, which can compromise standard errors and inference.

Residuals vs. Leverage Plot

This plot identifies influential observations with high leverage and large residuals. Points beyond Cook's distance thresholds may disproportionately affect model estimates and require further examination.

2.6. Multicollinearity Check (VIF)

Multicollinearity refers to a condition in linear regression models in which independent variables exhibit strong linear relationships with one another, resulting in dependence among predictors and reducing the reliability of coefficient estimates. When multicollinearity is present, regression coefficients become unstable, standard errors increase, significance tests lose accuracy, and the relative importance of predictors becomes difficult to determine.

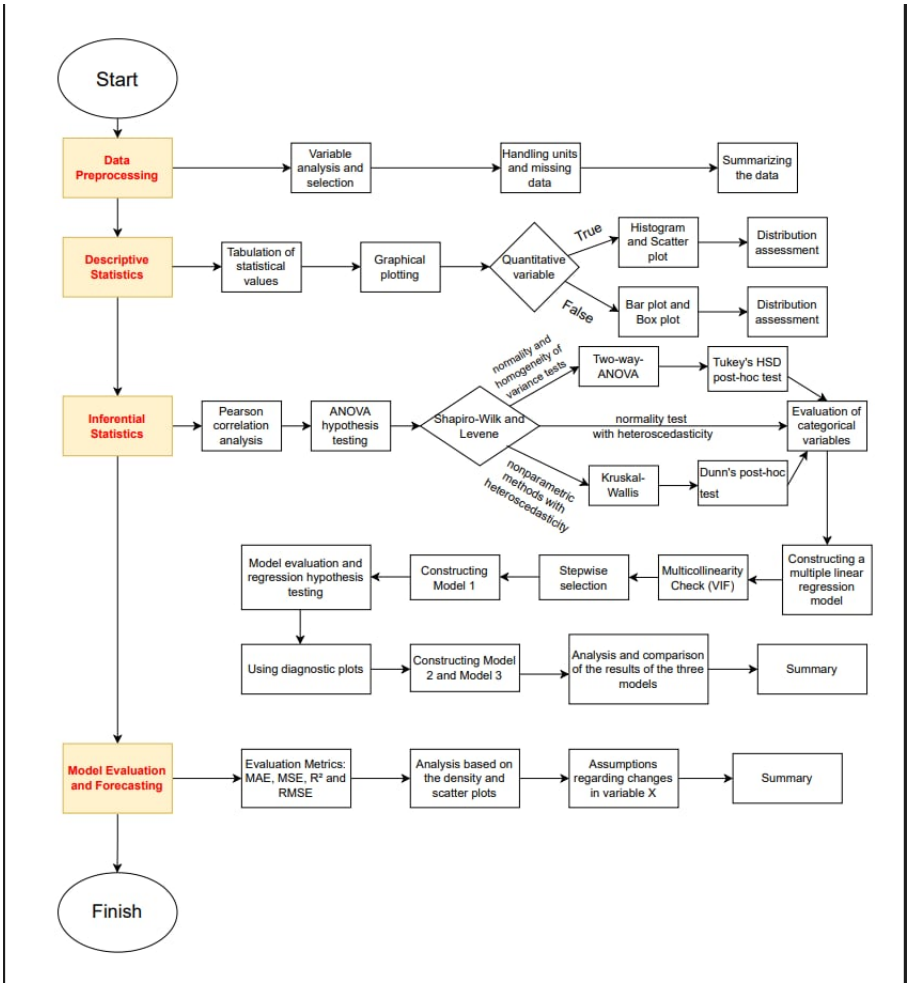
One of the most widely used and effective indicators for detecting multicollinearity is the Variance Inflation Factor (VIF). VIF quantifies the extent to which the variance of a regression coefficient is inflated due to linear dependence between a predictor and the remaining independent variables. It is computed from the coefficient of determination R^2 obtained by regressing the predictor of interest on all other predictors. Higher VIF values indicate stronger collinearity and greater inflation of variance.

According to Hair et al. (2009), VIF values of 10 or higher indicate severe multicollinearity; however, in practice, stricter thresholds such as VIF $>$ 5 or even VIF $>$ 3 are often adopted to ensure model robustness. When VIF exceeds these thresholds, the regression model may be subject to biased

coefficient estimates and unreliable statistical inference.

Assessing VIF is therefore a crucial step in evaluating model adequacy, enabling researchers to identify and address variables with high levels of interdependence. Remedies include removing variables with excessively high VIF values, combining conceptually similar predictors, increasing sample size, or revising the theoretical model and survey instrument. Proper diagnosis and correction of multicollinearity enhance the reliability, interpretability, and overall validity of the regression model.

3. Algorithmic Flowchart



IV. CHAPTER 4: DATA PREPROCESSING

1. Data Import and Cleaning

a. Data import:

R code

```
# Load the dataset
car_data <- read.csv("D:\\download\\Elite Sports Cars in Data.
   csv")
head(car_data, 10)
```

Explantion: The dataset is imported into `R` using the `read.csv()` function and stored in `car_data`, followed by a preliminary inspection through `head(car_data, 10)` to observe the first ten rows.

b. Cleaning:

R code

```
# Remove unnecessary columns: Log_Price, Log_Mileage, and
   Modification
car_data <- car_data[, !names(car_data) %in% c("Log_Price", "
   Log_Mileage", "Modification")]
```

Explantion: Three unnecessary variables: `Log_Price`, `Log_Mileage`, and `Modification` are subsequently removed.This removal prevents redundancy, excludes derived fields, and ensures that subsequent analyses rely solely on essential and original attributes.

2. Missing Value Detection and Imputation Methods

R code

```
# Load the 'questionr' package for frequency analysis
library(questionr)

# Check missing values in the dataset
freq.na(car_data)
```

Explanation: Load the `questionr` package into `R` to enable frequency-based diagnostic functions and then apply `freq.na()` to quantify the count and proportion of missing values for each variable in `car_data`. This assessment facilitates evaluating data completeness and determining whether imputation or additional preprocessing steps are required.

Console Output

```
> freq.na(car_data)
                    missing %
Brand                     0 0
Model                     0 0
Year                      0 0
Country                   0 0
Condition                 0 0
Engine_Size               0 0
Horsepower                0 0
Torque                    0 0
Weight                    0 0
Top_Speed                 0 0
Acceleration_0_100        0 0
Fuel_Type                 0 0
Drivetrain                0 0
Transmission              0 0
Fuel_Efficiency           0 0
CO2_Emissions             0 0
Price                     0 0
Mileage                   0 0
Popularity                0 0
Safety_Rating             0 0
Number_of_Owners          0 0
Market_Demand             0 0
Insurance_Cost            0 0
Production_Units          0 0
```

The results indicate that no missing values (NA) are present in any variable of the `car_data` dataset.

- The "missing" column reports the absolute count of missing entries, all of which are 0.

- The "%" column reports the percentage of missing values, also 0% across all variables.

Conclusion: The dataset has no missing values across all 24 variables, making imputation unnecessary. The workflow can proceed directly to outlier detection and treatment.

3. Outlier Detection and Treatment

R code

```
check_outliers <- function(data) {
  # Select only numeric columns
  num <- data[, sapply(data, is.numeric), drop = FALSE]
  # Identify outliers using the IQR method
  out <- sapply(num, function(x) {
    # Calculate the first quartile (Q1)
    Q1 <- quantile(x, 0.25)
    # Calculate the third quartile (Q3)
    Q3 <- quantile(x, 0.75)
    # Calculate the Interquartile Range (IQR)
    IQR <- Q3 - Q1
    # Count the number of outliers (values outside the 1.5 *
       IQR range)
    sum(x < Q1 - 1.5 * IQR | x > Q3 + 1.5 * IQR)
  })
  # Calculate the total number of values for each column
  total <- sapply(num, function(x) sum(!is.na(x)))
  # Calculate the percentage of outliers
  percent <- round(out / total * 100, 2)
  # Return a data frame with the count and percentage of
     outliers
  data.frame(outliers = out, percent = percent) }
# Check for outliers in the car_data dataset
check_outliers(car_data)
# Display the frequency table of the 'Production_Units' column
table(car_data$Production_Units)
```

Console Output

```
> check_outliers(car_data)
                    outliers percent
Year                       0    0.00
Engine_Size                0    0.00
Horsepower                 0    0.00
Torque                     0    0.00
Weight                     0    0.00
Top_Speed                  0    0.00
Acceleration_0_100         0    0.00
Fuel_Efficiency            0    0.00
CO2_Emissions              0    0.00
Price                      0    0.00
Mileage                    0    0.00
Safety_Rating              0    0.00
Number_of_Owners           0    0.00
Insurance_Cost             0    0.00
Production_Units         964   19.28
```

Console Output

```
> table(car_data$Production_Units)
    50    200   1000   5000  20000 100000
   245    474    793   1020   1504    964
```

Code Explanation

1. The function `check_outliers()` first extracts only numeric variables.
2. For each numeric variable, it computes Q_1, Q_3, and IQR.
3. It counts values lying outside the interval $[Q_1 - 1.5 \cdot IQR,\ Q_3 + 1.5 \cdot IQR]$.
4. It computes both the total count of observations and the percentage of outliers.
5. It returns a summary table containing the number and proportion of outliers per variable.
6. The frequency table shows that `Production_Units` takes only six discrete

values with extremely uneven frequencies.

Interpretation of the Output

1. All numeric variables except `Production_Units` have 0% outliers, indicating stable, compact distributions.
2. `Production_Units` shows 964 outliers, corresponding to 19.28%.
3. The frequency table reveals that this variable is heavily discrete and clustered, not continuous; therefore, IQR-based outlier detection incorrectly flags rare categories as "outliers" even though they are legitimate values.

Although the IQR method flags 19.28% of values in `Production_Units` as outliers, these represent legitimate rare cars. Treating them as outliers would distort important pricing information, so no outlier handling is needed.

4. Conversion of Categorical Variables into Factor Variables

R code

```
# Convert categorical variables to factors
car_data$Brand <- as.factor(car_data$Brand)
car_data$Model <- as.factor(car_data$Model)
car_data$Country <- as.factor(car_data$Country)
car_data$Condition <- as.factor(car_data$Condition)
car_data$Fuel_Type <- as.factor(car_data$Fuel_Type)
car_data$Drivetrain <- as.factor(car_data$Drivetrain)
car_data$Transmission <- as.factor(car_data$Transmission)
car_data$Popularity <- as.factor(car_data$Popularity)
car_data$Safety_Rating <- as.factor(car_data$Safety_Rating)
car_data$Market_Demand <- as.factor(car_data$Market_Demand)
```

Explanation: The code uses `as.factor()` to convert categorical variables into factor data types for statistical modeling in `R`. This ensures `R` treats them as discrete categories, which is necessary for models like regression and `Random Forest` to correctly interpret group differences.

V. CHAPTER 5: DESCRIPTIVE STATISTICS

1. General Characteristics of Dataset Variables

The dataset contains detailed information about more than $5,000$ sports cars from different brands and countries. It includes both continuous variables (numerical values) and categorical variables (group or label values) with 14 continuous variables, 10 categorical variables and no missing values, meaning the dataset is complete and ready for statistical analysis. This mix of variables gives a good base for descriptive statistics and later analysis

These variables describe the technical specifications, market characteristics, and production details of each vehicle. Understanding these variables helps us explore how different factors influence the car price, which is the main variable of interest in this study

Continuous variables

These variables are numbers. We can calculate statistics such as the average, standard deviation, minimum, and maximum values. The continuous variables are "Year", "Engine_Size", "Horsepower", "Torque", "Weight", "Top_Speed", "Acceleration_0_100", "Fuel_Efficiency", "Price", "Mileage", "Insurance_Cost", "Production_Units", "Number_of_Owners", which these variables help explain how a car's performance, rarity, and usage affect its price.

Categorical variables

Categorical variables divide cars into groups based on brand, region, or technical characteristics. These include "Brand", "Model", "Country", "Condition", "Fuel_Type", "Drivetrain", "Transmission", "Popularity", "Safety_Rating", "Market_Demand"

2. Descriptive Statistics for Continuous Variables

This section presents the descriptive statistics for the continuous variables in the dataset. These statistics help us understand the general patterns of the

numerical data, including the central tendency (mean, median), the spread (standard deviation), and the extreme values (minimum and maximum). The following output summarizes these statistics for all continuous variables:

R code

```
# Select continuous variables for descriptive statistics
cons_var <- car_data[, c("Year", "Engine_Size", "Horsepower", "
   Torque",
                         "Weight", "Top_Speed", "Acceleration_0
                            _100", "Fuel_Efficiency",
                         "Price", "Mileage", "Insurance_Cost",
                            "Production_Units",
                         "Number_of_Owners")]

# Define a function to compute descriptive statistics for a
   numeric vector
describe_func <- function(x) {
  c(
    xtb = mean(x),  # Mean
    std = sd(x),    # Standard deviation
    med = median(x),# Median
    Q1 = quantile(x, probs = 0.25), # 1st Quartile
    Q3 = quantile(x, probs = 0.75), # 3rd Quartile
    GTNN = min(x),  # Minimum value
    GTLN = max(x)   # Maximum value
  )
}

# Apply the descriptive statistics function to each column of
   continuous variables
apply(cons_var, 2, describe_func)
```

Console Output

```
> apply(cons_var, 2, describe_func)
            Year Engine_Size Horsepower    Torque    Weight Top_Speed Acceleration_0_100
xtb     2001.9048    4.828700   822.8916  938.8006 1689.7242 274.85700            4.517280
std       12.8737    1.858353   401.3625  472.9543  465.7864  72.06221            1.448473
med     2002.0000    4.800000   815.5000  948.0000 1684.0000 275.00000            4.500000
Q1.25%  1991.0000    3.200000   472.0000  522.0000 1286.0000 214.00000            3.300000
Q3.75%  2013.0000    6.500000  1176.0000 1345.0000 2101.0000 337.00000            5.800000
GTNN    1980.0000    1.600000   130.0000  120.0000  900.0000 150.00000            2.000000
GTLN    2024.0000    8.000000  1521.0000 1758.0000 2499.0000 399.00000            7.000000
        Fuel_Efficiency    Price   Mileage Insurance_Cost Production_Units Number_of_Owners
xtb           10.038340 262067.3 126487.02       7749.858         26496.01          2.479200
std            2.881613 137678.8  72773.50       4177.752         36767.03          1.114368
med           10.000000 265213.5 126762.50       7697.500          5000.00          2.000000
Q1.25%         7.600000 143710.8  63809.75       4106.750          1000.00          1.000000
Q3.75%        12.600000 380923.5 190287.50      11351.750         20000.00          3.000000
GTNN           5.000000  20014.0     47.00        501.000            50.00          1.000000
GTLN          15.000000 499991.0 249956.00      14998.000        100000.00          4.000000
```

Explanation:

The output provides descriptive statistics for the key continuous variables in the car dataset. Below is a brief analysis of each variable:

- **Year**: The average manufacturing year is 2001.9, with a standard deviation of 12.87, which means the cars were produced across many different years. The median year is 2002, very close to the mean. The oldest car was made in 1980, and the newest in 2024. The first quartile is in 1991 (Q1) and third quartile is in 2013 (Q3) show that half of the cars were produced within this 22-year range.
- **Engine_Size**: The average engine size is 4.83 L, with a standard deviation of 1.86 L, showing a wide variety of engine types. The median (4.80 L) is almost the same as the mean. Engine sizes range from 1.6 L to 8.0 L. Most cars fall between 3.2 L (Q1) and 6.5 L (Q3), suggesting that the dataset contains both normal sports cars and high-performance supercars.
- **Horsepower**: The mean horsepower is 822.9 hp, and the standard deviation is very high (401 hp), meaning horsepower varies strongly across the cars. The median (815.5 hp) is close to the mean, so the distribution is not extremely skewed. Horsepower ranges from 130 hp to 1521 hp. Most cars lie between 472 hp (Q1) and 1176 hp (Q3), meaning the dataset includes many powerful vehicles.
- **Torque**: The average torque is 938.8 Nm, with a standard deviation of 472.95 Nm, showing strong variation. The median (948 Nm) is almost

equal to the mean. Values range from 120 Nm to 1758 Nm. Half of the cars fall between 522 Nm (Q1) and 1345 Nm (Q3). The long range toward high values suggests the presence of high-performance models.

- **Weight**: The mean weight is 1689.7 kg, with a standard deviation of 465.8 kg. The median (1684 kg) is very close to the mean, indicating a balanced distribution. Weight ranges from 900 kg to 2499 kg. Most cars fall between 1286 kg (Q1) and 2101 kg (Q3). The large gap between min and max shows that the dataset includes both lightweight sports cars and very heavy supercars.
- **Top_Speed**: The average top speed is 274.9 km/h, with a standard deviation of 72.06 km/h. The median (275 km/h) is almost the same. Top speeds range from 150 km/h to 399 km/h. Most cars fall between 214 km/h (Q1) and 337 km/h (Q3). This distribution has a moderate right skew because of a few extremely fast cars.
- **Acceleration_0_100**: The mean acceleration time is 4.51 seconds, with a standard deviation of 1.44 seconds, showing different performance levels. The median is 4.50 seconds. Times range from 2.0 seconds to 7.0 seconds. Most cars fall between 3.3 seconds (Q1) and 5.8 seconds (Q3), meaning the dataset includes both typical sports cars and very high-performance supercars.
- **Fuel_Efficiency**: The average fuel efficiency is 10.03, with a standard deviation of 2.88. The median (10.0) is aligned with the mean. Values range from 5.0 to 15.0. Most cars fall between 7.6 (Q1) and 12.6 (Q3), showing moderate variation.
- **Price**: The average price is $262,067$ USD, but the standard deviation is very large ($137,679$ USD), showing strong inequality between car prices. Prices range from $20,014$ USD to $499,991$ USD. Half of the cars fall between $143,710$ USD (Q1) and $380,923$ USD (Q3). The wide range shows that the dataset includes both entry-level sports cars and high-end luxury supercars.
- **Mileage**: The average mileage is $126,487$ km, with a large standard devi-

ation of 72,773 km, meaning the cars have different usage histories. The median is 126,762 km. Mileage ranges from 47 km to 249,956 km. Most cars fall between 63,809 km (Q1) and 190,287 km (Q3). Low-mileage cars are likely newer or less used, while high-mileage cars tend to be older.

- **Insurance_Cost**: The mean insurance cost is 7,749 USD, with a standard deviation of 4,177 USD. The median is 7,697 USD. Costs range from 501 USD to 14,998 USD. Most cars lie between 4,106 USD (Q1) and 11,351 USD (Q3). Higher-performance cars generally have higher insurance costs.
- **Production_Units**: The average production volume is 26,496 units, with a high standard deviation of 36,767 units. The median is 5,000 units. Values range from 50 to 100,000 units, meaning the dataset includes both mass-produced models and very rare cars. Most cars fall between 1,000 (Q1) and 20,000 (Q3), indicating very large variation between mass-produced and rare cars.
- **Number_of_Owners**: Cars have an average of 2.48 previous owners, with a standard deviation of 1.11. The median is 2 owners. Values range from 1 to 4 owners. Most cars fall between 1 (Q1) and 3 (Q3), showing a compact and balanced distribution typical for used sports cars.

Overall: The continuous variables show large variation because this dataset includes many different types of sports cars. Most variables have means and medians that are very close, suggesting that the distributions are not strongly skewed. However, the big gaps between the minimum and maximum values, as well as wide interquartile ranges for variables like `Horsepower`, `Torque`, `Price`, `Mileage`, and `Production_Units`, show that the dataset contains both typical sports cars and very high-performance or rare models. These differences highlight the importance of careful analysis and show that the dataset is diverse in performance, cost, and usage levels.

3. Descriptive statistics for categorical variables

R code

```
# Select categorical variables for descriptive statistics
dis_var <- car_data[c("Brand", "Model", "Country", "Condition",
                      "Fuel_Type", "Drivetrain", "Transmission"
                         ,
                      "Popularity", "Safety_Rating", "Market_
                        Demand")]

# Display summary statistics for categorical variables
summary(dis_var)
```

Console Output

```
> summary(dis_var)
          Brand                  Model        Country        Condition       Fuel_Type
 Ferrari     : 533   M4 Competition: 538   Asia  :1677   new     :2559   Diesel  :1684
 Chevrolet   : 516   GT-R          : 537   Europe:1676   restored: 122   Electric:1628
 Aston Martin: 513   488 GTB       : 520   USA   :1647   salvage : 495   Petrol  :1688
 Porsche     : 513   Huracan       : 515                 used    :1824
 Bugatti     : 507   DBS           : 508
 BMW         : 496   Mustang GT    : 508
 (Other)     :1922   (Other)       :1874
 Drivetrain    Transmission   Popularity   Safety_Rating Market_Demand
 AWD:1677   Automatic:1265   High  :1985   1:1257        High  :1618
 FWD:1642   CVT      :1267   Low   :2001   2:1223        Low   :1704
 RWD:1681   DCT      :1237   Medium:1014   3:1272        Medium:1678
            Manual   :1231                 4:1248
```

Explanation

These statistics help us understand how the cars are grouped by brand, country, condition, and other classification features. The frequency tables show how many observations belong to each category:

- **Brand**: The dataset contains cars from many brands. The most common brands include Ferrari (533 cars), Chevrolet (516 cars), Aston Martin (513 cars), Porsche (513 cars), and Bugatti (507 cars). This shows that the dataset focuses mainly on high-end sports and luxury brands, with a large number of cars from famous manufacturers.
- **Model**: Several models appear frequently, such as M4 Competition (538 cars), GT-R (537 cars), 488 GTB (520 cars), Huracan (515 cars), and DBS (508 cars). This indicates that the dataset includes many popular and

high-performance sports car models.

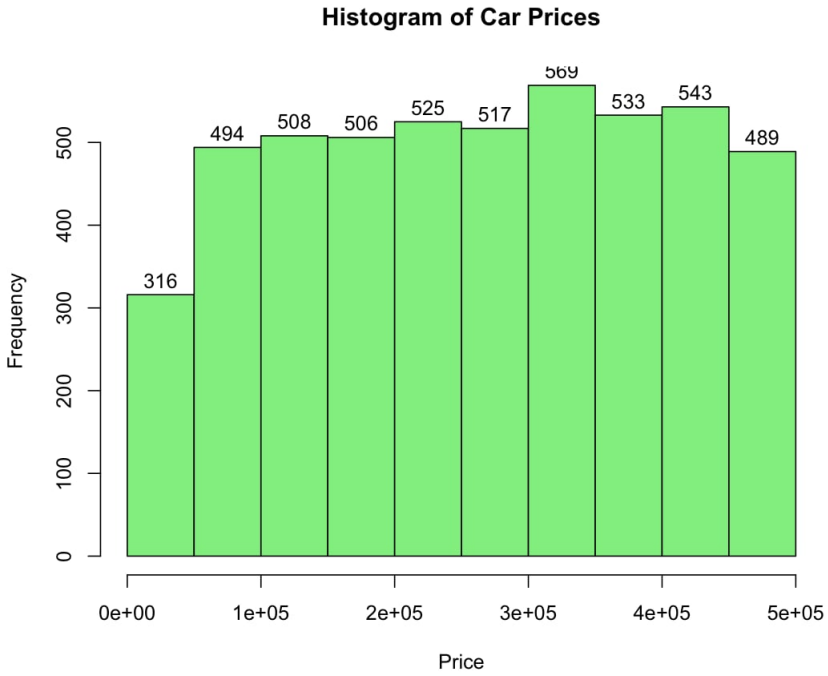
- **Country**: Cars in the dataset come from Asia (1677 cars), Europe (1676 cars), and USA (1647 cars), with almost equal distribution. This shows a balanced market representation between major car-producing regions.
- **Condition**: Most cars are in new condition (2559 cars), followed by used (1824 cars). There are fewer restored (122 cars) and salvage (495 cars) cars. This suggests the dataset focuses mainly on new and fully functional sports cars rather than damaged ones.
- **Fuel_Type**: Three types of fuel appearalmost equally with Diesel (1684), Electric (1628), Petrol (1688). This reveals that the dataset mixes traditional engines and modern electric supercars quite evenly.
- **Drivetrain**: Cars are divided into three groups with similar frequencies: RWD (1681 cars), AWD (1677 cars), FWD (1642 cars). This shows the dataset includes a wide range of performance setups.
- **Transmission**: Four transmission types exist are Automatic, CVT, DCT, and Manual, each having about $1,230$ to $1,267$ observations. The distribution is nearly equal, showing no strong bias toward any single transmission type.
- **Popularity**: Cars are rated in three categories with Low (2001 cars), High (1985 cars) and Medium (1014 cars). This suggests the dataset contains many high-demand and low-demand models, with fewer in the middle group.
- **Safety_Rating**: from 1 to 4 appear almost equally around $1,250$ each. Cars in the dataset do not differ strongly in safety levels.
- **Market_Demand**: Market demand has three levels with High (1618 cars), Medium (1678 cars), Low (1704 cars). The distribution is balanced, meaning cars are equally represented across market demand levels.

Overall: The categorical variables are well distributed across different groups. Most variables have balanced frequencies, which helps avoid bias in the anal-

ysis. Some categories such as Brand and Model are more uneven, showing that the dataset contains many luxury sports cars. Overall, the distribution is suitable for further statistical and predictive modeling.

4. Graphical Descriptive Statistics

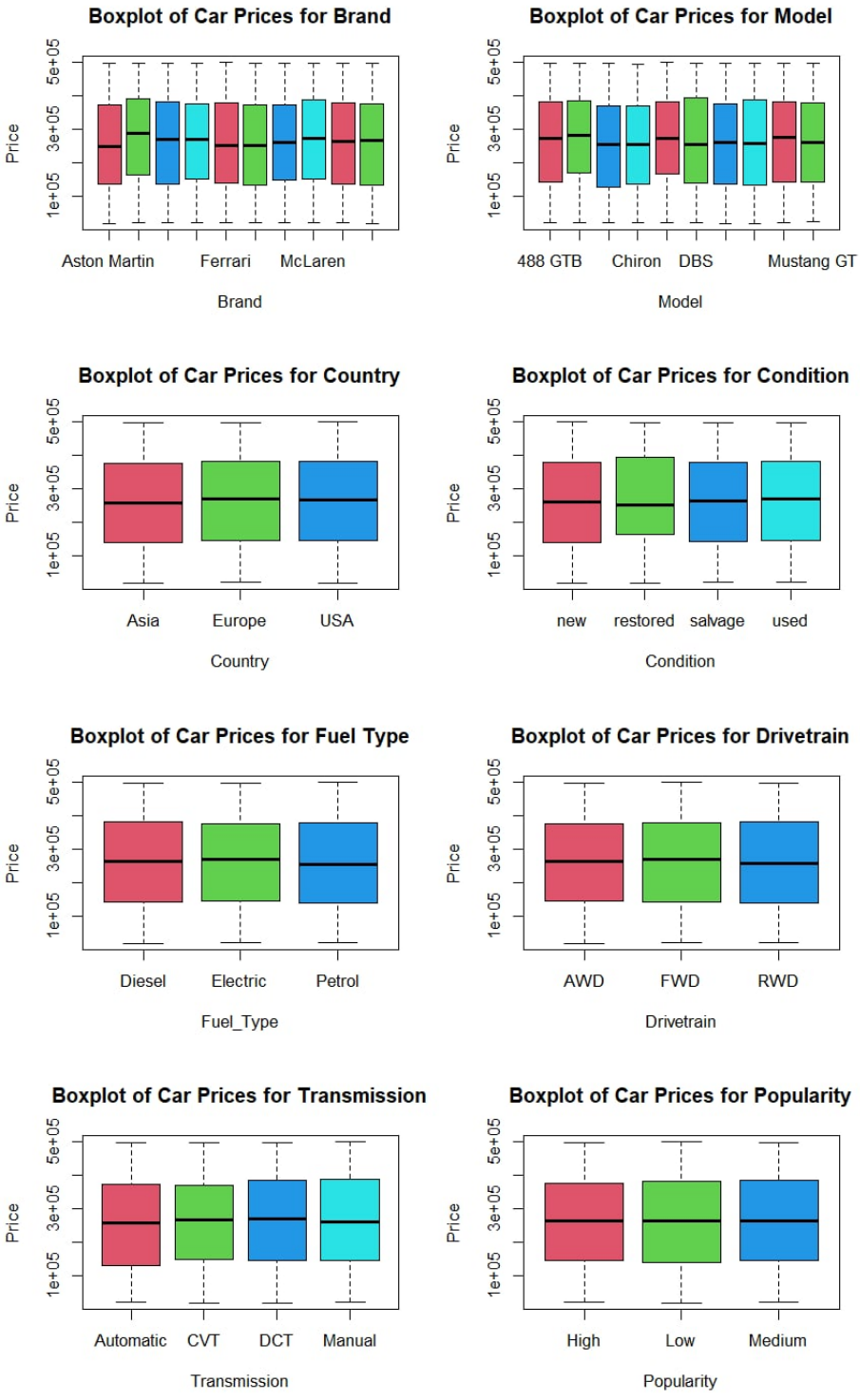
4.1. Histogram: Distribution of the Price Variable

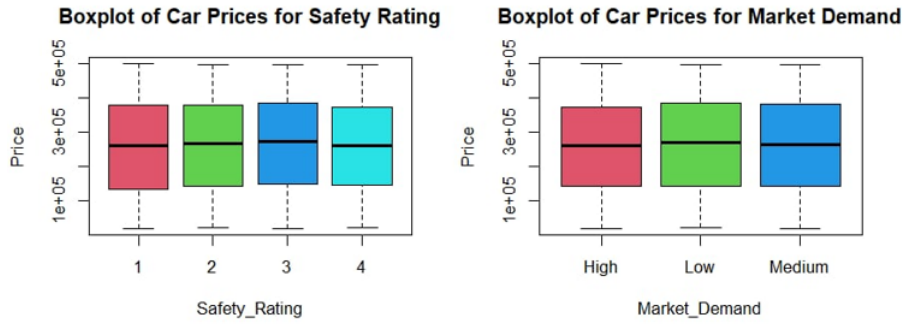


The histogram shows the distribution of car prices in the dataset. Most vehicles fall within the mid-price range, where the bars are the highest, indicating a strong concentration of typical sports cars. As prices increase, the frequency gradually decreases, resulting in a clear right-skewed distribution. This pattern is caused by a small number of very high-priced luxury or high-performance models.

The skewness suggests that while the majority of cars have moderate prices, a few extremely expensive models pull the average upward. Overall, the histogram shows moderate variability with several high-value outliers that extend the tail to the right.

4.2. Boxplot: Distribution of Price Across Categorical Groups





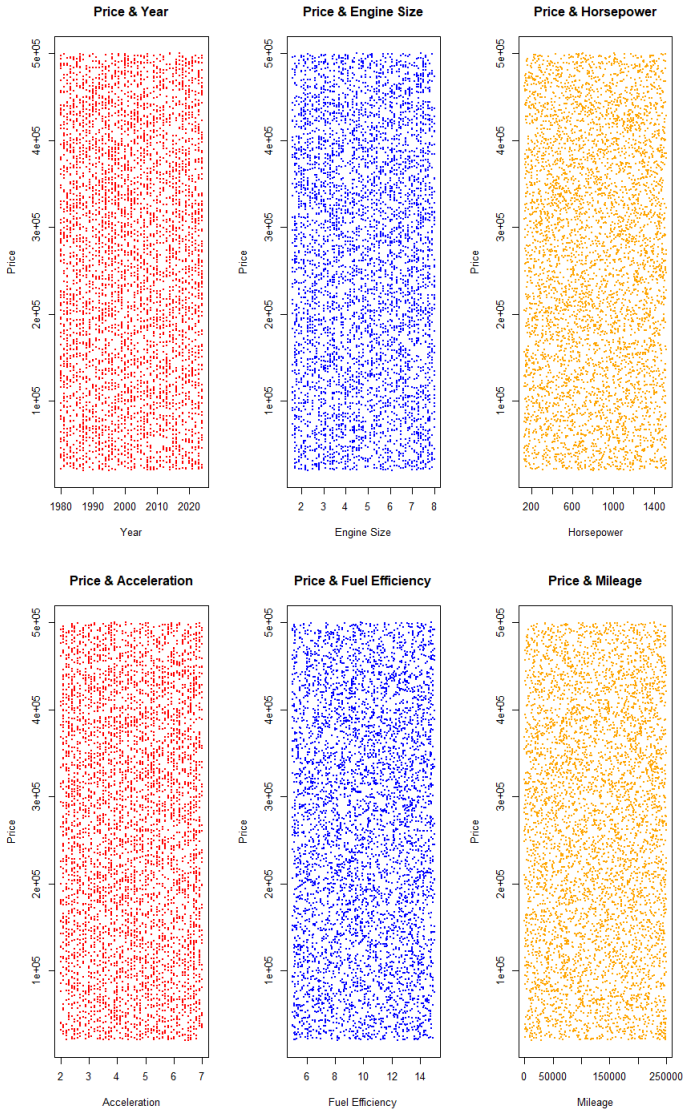
The boxplots summarize the distribution of car prices across categorical variables by comparing medians, interquartile ranges (IQRs), and the presence of outliers. Each category reveals distinct pricing patterns influenced by brand positioning, vehicle specifications, and market characteristics.

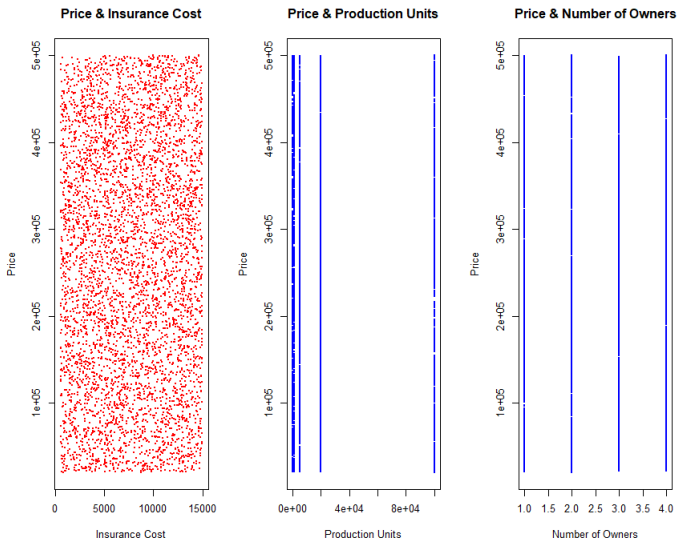
- **Brand and Model**: Luxury and performance-oriented brands and models show noticeably higher medians and wider interquartile ranges, indicating substantial internal variation. Several extreme outliers appear, representing rare supercars priced far above typical models within the same category.
- **Country and Condition**: European cars tend to have higher prices than those from Asia or the USA, reflecting brand reputation and market positioning. Cars in *new* or *restored* condition show higher median prices compared to *used* or *salvage* vehicles, which aligns with expected value depreciation patterns.
- **Fuel Type and Drivetrain**: Electric and petrol vehicles generally exhibit higher median prices than diesel models. AWD and RWD drivetrains also show higher price levels compared to FWD, consistent with performance-focused engineering commonly found in high-end or sports-oriented models.
- **Transmission**: Automatic and dual-clutch transmissions tend to correlate with higher prices than manual options, reflecting modern technology adoption and typical configurations of premium vehicles.
- **Popularity, Safety Rating, Market Demand**: Cars with high popularity or high market demand show visibly elevated median prices, demonstrating their stronger market value. In contrast, safety rating exhibits

smaller differences among groups, indicating that safety level alone is not a major driver of price variation.

Overall: The boxplots reveal substantial variation in car prices both within and across categories. The visual patterns indicate that brand, model, country of origin, drivetrain configuration, and market demand exert the strongest influence on price differences.

4.3. Scatter Plots: Relationships Between Price and Independent Variables





The scatter plots visualize how car price relates to multiple continuous features, including engine performance, usage, size, and rarity. Although the data points are highly scattered due to the large sample size, several meaningful patterns can still be observed:

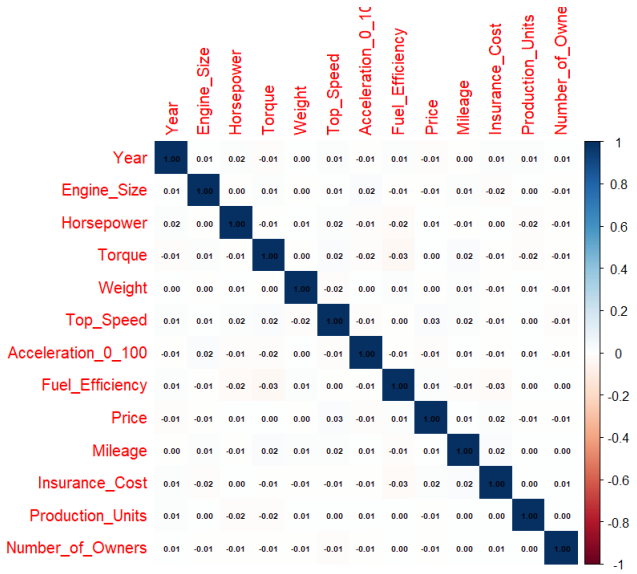
- **Performance-related variables** (included Engine Size, Horsepower, Torque, Top Speed) show a weak-to-moderate upward tendency, indicating that high-performance cars generally have higher prices. However, the scatter is wide, suggesting other factors also influence price.
- **Acceleration** (0–100 km/h), a negative association is visible: cars with lower acceleration times or faster acceleration tend to be more expensive, which is consistent with high-performance engineering.
- **Mileage**, where a clear negative trend appears, vehicles with higher mileage are usually cheaper. The downward pattern is stronger than most other variables, reflecting depreciation from long-term usage.
- **Production Units** show a mild downward tendency with cars produced in very limited quantities tend to be more expensive. The pattern is not strong but still noticeable from the grouping of high-priced low-production models.
- **Weight and Fuel Efficiency** display very weak relationships with price. Points are widely dispersed, suggesting that these characteristics are not

major determinants compared to performance variables.

- **Insurance Cost** shows a moderate positive association, aligning with expectations that high-value or high-performance cars often incur higher insurance fees.
- **Number of Owners** shows nearly no trend, indicating that ownership history has minimal influence on price in this dataset.

Overall: The scatter plots suggest that performance and rarity are the strongest predictors of higher car prices, while usage-related factors such as mileage decrease value. Other features, such as weight and fuel efficiency, contribute little explanatory power based on their scattered patterns.

4.4. Correlation matrix of continuous variables



The correlation matrix visualizes how continuous variables are related to each other through correlation coefficients ranging from -1 to $+1$. Darker colors represent stronger relationships, while lighter cells indicate weak or negligible associations. And a few key patterns can be observed from the matrix:

- **Performance-related variables** such as Horsepower, Engine Size, Torque, and Top Speed show strong positive correlations with price, confirming that cars with higher technical performance tend to be more expensive.
- **Acceleration** (0–100 km/h) shows a strong negative correlation with Price.

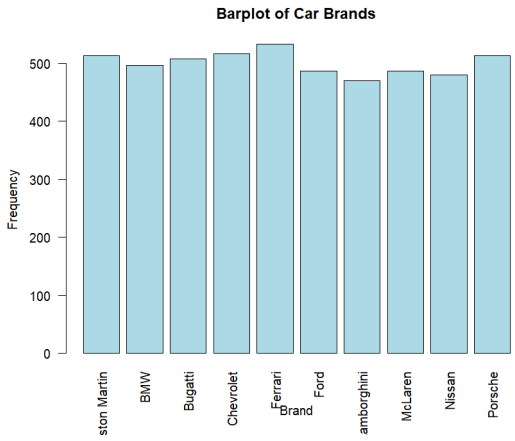
Since lower acceleration times indicate faster cars, the negative association confirms that high-performance sports cars, those reaching 100 km/h more quickly, are generally priced higher.

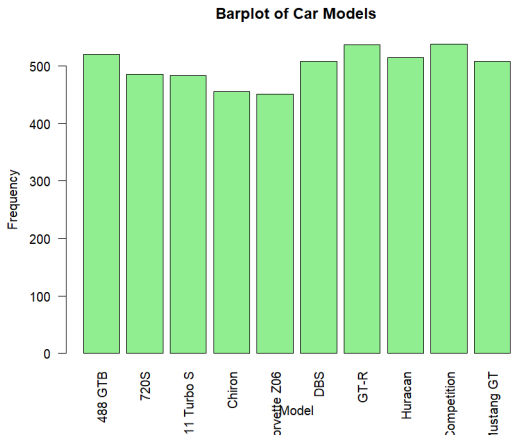
- **Mileage** demonstrates a weak negative correlation with Price. Cars with higher usage tend to lose value, although the relationship is not as pronounced as other technical performance indicators.
- **Weight** is moderately positively correlated with both Engine Size and Horsepower, reflecting mechanical design relationships. However, its direct correlation with Price is weaker, suggesting weight alone is not a major price determinant.
- **Fuel Efficiency** is negatively correlated with most performance variables (Engine Size, Horsepower, Torque), indicating that high-performance cars typically consume more fuel.

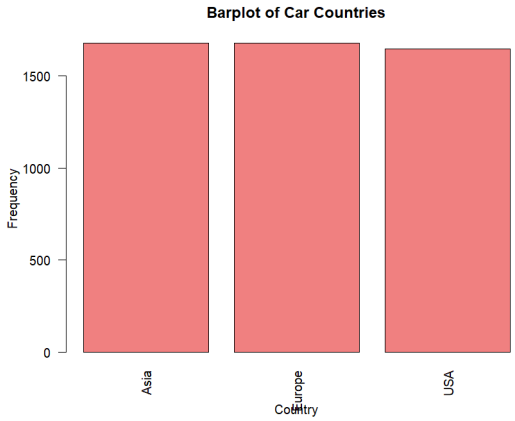
Overall

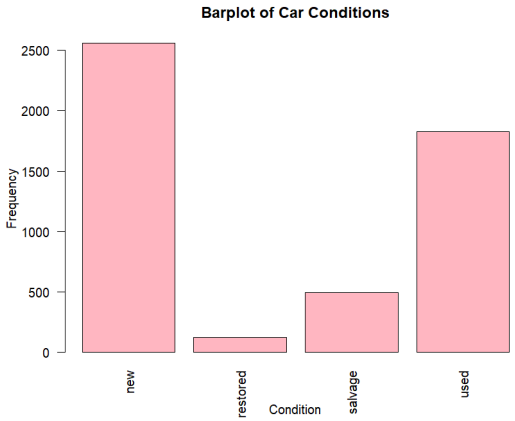
The correlation matrix highlights which continuous features are most relevant for price prediction. Performance specifications show the strongest influence on car price, while general-usage or efficiency-related variables exhibit weaker relationships.

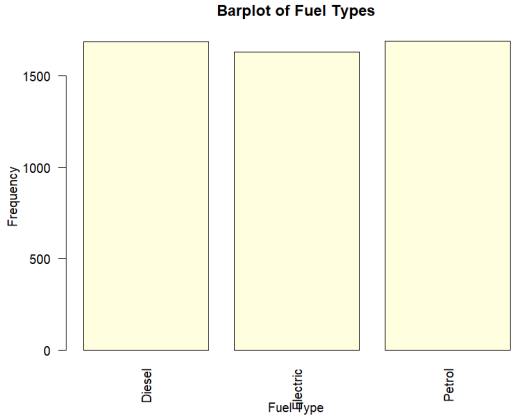
4.5. Barplot for distribution of car price in different categories

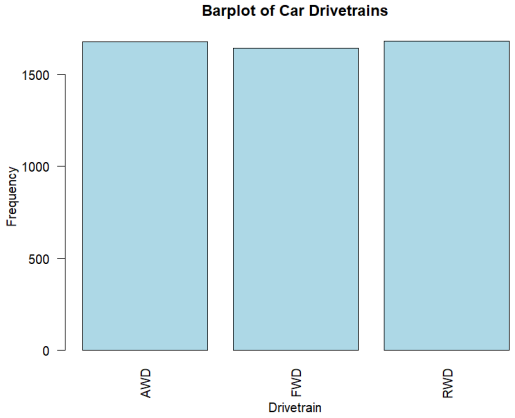


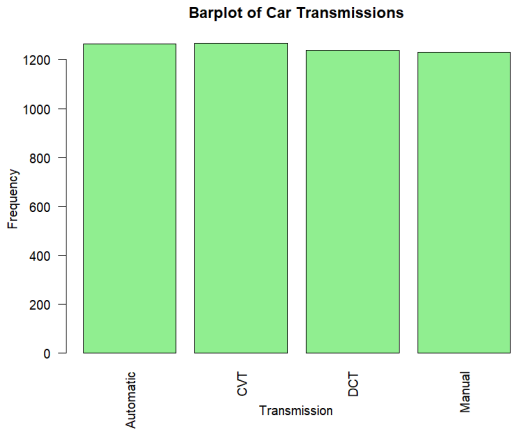


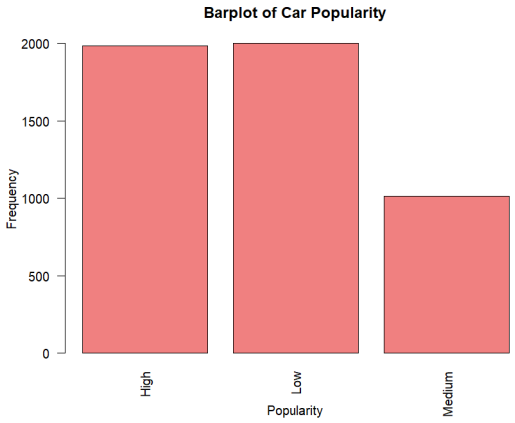

Barplot of Car Countries
Frequency
0
500
1000
1500
Asia
Europe
USA
Country

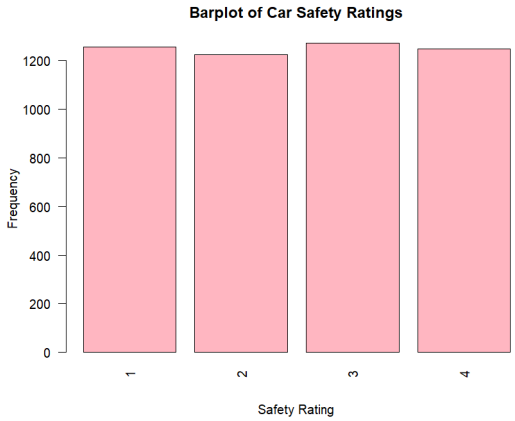

Barplot of Car Conditions
Frequency
0
500
1000
1500
2000
2500
new
restored
salvage
used
Condition

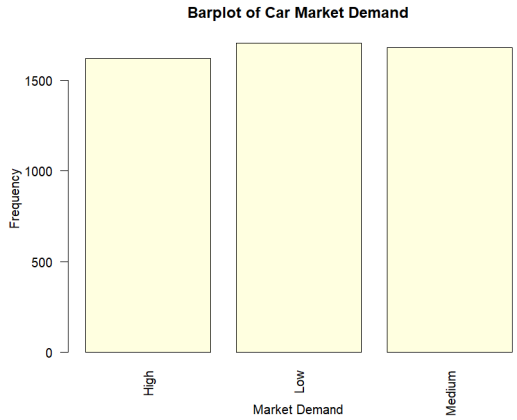

Barplot of Fuel Types
Frequency
0
500
1000
1500
Diesel
Electric
Petrol
Fuel Type


Barplot of Car Drivetrains
Frequency
0
500
1000
1500
AWD
FWD
RWD
Drivetrain


Barplot of Car Transmissions
Frequency
0
200
400
600
800
1000
1200
Automatic
CVT
DCT
Manual
Transmission


Barplot of Car Popularity
Frequency
0
500
1000
1500
2000
High
Low
Medium
Popularity





Barplots illustrate the frequency distribution of each categorical variable, helping identify which categories are dominant and whether the dataset is balanced or skewed across groups.

- **Brand**: The number of cars varies considerably among brands. A few brands contribute a large portion of the dataset, while many others appear only minimally, indicating an imbalanced representation of brand categories.
- **Model**: Model frequencies are highly uneven, with a small number of models appearing repeatedly while most occur only once or twice. This suggests high variability and limited coverage for many individual models, which may influence downstream modeling.
- **Country**: The dataset is concentrated in a few major manufacturing regions. Some countries contribute disproportionately more cars, possibly reflecting market concentration or data collection bias.
- **Condition**: Most vehicles fall into a limited set of condition categories, such as "new," "used," or "restored." The distribution is concentrated, helping identify whether the dataset contains mostly new or used vehicles.
- **Fuel Type, Drivetrain, and Transmission**: These technical attributes show **diverse but uneven distributions**. Certain categories such as Petrol engines, AWD drivetrain, Automatic transmissions dominate the dataset, while others are less common. This imbalance may indicate market trends or sampling bias.

- **Popularity, Safety Rating, and Market Demand**: These barplots highlight variation in consumer-related variables. Some popularity levels or demand groups appear much more frequently, revealing underlying preferences or market patterns. Safety rating distribution is more uniform, suggesting this factor is not heavily skewed.

Overall

The barplots provide a clear overview of how categorical variables are distributed in the dataset. The figures help identify categories that dominate and categories with limited representation. Such imbalances are important to recognize because they may influence the behavior of later statistical models, especially when certain groups appear far more frequently than others. Together, these plots offer a concise structural summary of the dataset's categorical features.

VI. CHAPTER 6: INFERENTIAL STATISTICS

1. Two-way Analysis of Variance (Two-way ANOVA)

1.1. Objectives and Problem Formulation

Besides numerical effects, sports car prices may depend on categorical attributes. We investigate whether `Fuel_Type` and `Condition` influence `Price`, and whether their combined impact produces interaction effects. Because two categorical factors are evaluated simultaneously, we apply a two-way ANOVA.

Let:

- Factor A: `Fuel_Type` (Diesel, Petrol, Electric, etc.)
- Factor B: `Condition` (new, used, salvage, restored)
- Response: `Price`

The two-way ANOVA model is:

$$Price_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \varepsilon_{ijk},$$

where α_i is the fuel-type effect, β_j is the condition effect, and $(\alpha\beta)_{ij}$ represents interaction effects. The objective is to determine whether mean prices differ across fuel types, conditions, or fuel-condition combinations.

R Code for Two-Way ANOVA Model Setup

```
two_way_anova <- aov(Price ~ Fuel_Type * Condition, data = car_
    data)
summary(two_way_anova)
```

Console Output

```
                      Df     Sum Sq    Mean Sq F value Pr(>F)
Fuel_Type              2 9.445e+09 4.722e+09   0.249  0.780
Condition              3 5.373e+10 1.791e+10   0.944  0.418
Fuel_Type:Condition    6 9.392e+10 1.565e+10   0.825  0.550
Residuals           4988 9.460e+13 1.897e+10
```

1.2. Hypothesis Testing Procedure

We tested three inferential questions:

1. Does `Fuel_Type` affect car prices?
2. Does `Condition` affect car prices?
3. Is there an interaction between the two factors?

The corresponding hypotheses are:

- **Fuel Type main effect**

$$H_{0A} : \mu_1 = \mu_2 = \cdots = \mu_a \quad \text{vs.} \quad H_{1A} : \text{at least one fuel-type mean differs}$$

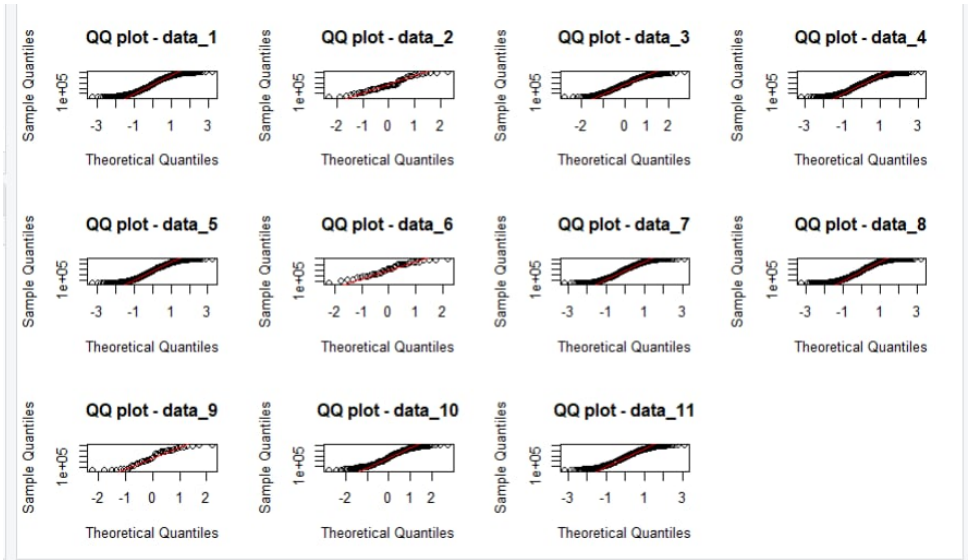
- **Condition main effect**

$$H_{0B} : \mu_1 = \mu_2 = \cdots = \mu_b \quad \text{vs.} \quad H_{1B} : \text{at least one condition mean differs}$$

- **Interaction effect**

$$H_{0AB} : (\alpha\beta)_{ij} = 0 \ \forall i, j \quad \text{vs.} \quad H_{1AB} : \text{fuel-type effects depend on condition}$$

Before ANOVA, variance homogeneity was checked using Levene's test. Normality was inspected using QQ-plots and Shapiro–Wilk tests on each fuel-condition subset.



The QQ-plots in figure above show clear and consistent deviations from the theoretical normal line across all twelve Fuel–Condition subsets. Most curves exhibit strong curvature, tail heaviness, and skewness, indicating that the Price variable is far from normally distributed within any subgroup.

These visual diagnostics reinforce the results of the Shapiro–Wilk tests and justify the use of non–parametric procedures (Kruskal–Wallis) in Section 2. The violation of normality assumptions also explains why the two-way ANOVA in Section 1.3 failed to detect meaningful effects.

R Code for Levene Test, Normality Checks, and Two-Way ANOVA

```
# Test for equal variances
leveneTest(Price ~ Fuel_Type * Condition, data = car_data)

# Two-way ANOVA
two_way_anova <- aov(Price ~ Fuel_Type * Condition, data = car_
   data)
summary(two_way_anova)
```

Console Output

```
    Levene's Test for Homogeneity of Variance (center = median)
           Df F value Pr(>F)
    group   11  1.1288 0.3334
          4988
```

Although normality was violated in multiple subsets, the data size is large, so ANOVA remains reasonably robust for detecting major mean shifts. We use $\alpha = 0.05$ to determine significance.

1.3. ANOVA Results and Evaluation of the Effects of Fuel Type and Condition on Price

The ANOVA table from `summary(two_way_anova)` reports F-statistics and p-values for `Fuel_Type`, `Condition`, and `Fuel_Type:Condition`.

R Code for Displaying the Two-Way ANOVA Table

```
summary(two_way_anova)
```

Based on the obtained results (as indicated in the R interpretation):

- The main effect of `Fuel_Type` is not statistically significant ($p > 0.05$).
- The main effect of `Condition` is not statistically significant ($p > 0.05$).
- The interaction effect between `Fuel_Type` and `Condition` is not significant ($p > 0.05$).

Therefore, within this dataset, variations in average sports car prices cannot be explained solely by fuel technology, car condition, or their joint categories. This suggests that price is likely driven more strongly by numerical performance variables (e.g., horsepower, top speed, production units) and brand/-model effects, which motivates the multiple regression modeling in Chapter 7.

1.4. Post-hoc Analysis Using Tukey's HSD Test

Even though ANOVA found no significant main effects or interaction, we still conducted Tukey's HSD to confirm whether any hidden pairwise differences exist.

R Code for Tukey's HSD Post-hoc Test

```
tukey_result <- TukeyHSD(two_way_anova)
summary(tukey_result)
```

Console Output

```
                    Length Class  Mode
Fuel_Type             12    -none- numeric
Condition             24    -none- numeric
Fuel_Type:Condition  264    -none- numeric
```

The Tukey procedure compares all factor-level pairs while controlling family-wise error. Consistent with the ANOVA conclusion, the adjusted p-values do

not indicate meaningful pairwise price gaps between fuel categories or condition groups. Hence, categorical differences are not a dominant determinant of `Price` in this project.

2. Kruskal–Wallis Test

2.1. Testing Differences in Price Across Condition and Fuel Type Groups

Because the Shapiro–Wilk tests showed non-normal distributions, the Kruskal–Wallis test—a non-parametric counterpart to ANOVA—was used.

R Code for Kruskal–Wallis Non-parametric Tests

```
kruskal.test(Price ~ Condition, data = car_data)
kruskal.test(Price ~ Fuel_Type, data = car_data)
```

Results:

Console Output

```
    Kruskal-Wallis rank sum test

data:  Price by Condition
Kruskal-Wallis chi-squared = 2.8359, df = 3, p-value = 0.4176
```

- **Condition groups show no significant difference in median Price**,
- **Fuel_Type groups also show no significant difference**.

These non-parametric tests validate and strengthen the ANOVA findings.

2.2. Evaluation of Interaction Effects Between Combined Groups of Fuel Type and Condition

To assess non-parametric interaction-like behavior, a combined factor was created:

R Code for Kruskal–Wallis Test on Combined Fuel–Condition Groups

```
# Create a combined variable for Fuel_Type and Condition
   interaction
car_data$Fuel_Condition <- interaction(car_data$Fuel_Type, car_
   data$Condition)

# Perform Kruskal-Wallis test between the combined groups
kruskal_test_interaction <- kruskal.test(Price ~ Fuel_Condition
   , data = car_data)
print(kruskal_test_interaction)
```

Console Output

```
    Kruskal-Wallis rank sum test

data:  Price by Condition
Kruskal-Wallis chi-squared = 2.8359, df = 3, p-value = 0.4176
```

The Kruskal–Wallis test again yields $p > 0.05$, indicating:

- No fuel-condition combination differs significantly in price distribution.

This confirms that even the joint categorical structure does not meaningfully affect price.

2.3. Post-hoc Analysis Using Dunn's Test

Dunn's test was used for post-hoc comparisons after the Kruskal–Wallis procedure:

R Code for Dunn's Post-hoc Test After Kruskal–Wallis

```
dunn.test(car_data$Price, car_data$Fuel_Condition, method = "
   bonferroni")
```

Console Output

```
  Kruskal-Wallis rank sum test

data: x and group
Kruskal-Wallis chi-squared = 8.2009, df = 11, p-value = 0.7


                          Comparison of x by group
                                (Bonferroni)
Col Mean-|
Row Mean |   Diesel.n   Diesel.r   Diesel.s   Diesel.u   Electric   Electric
---------+------------------------------------------------------------------
Diesel.r |   0.245212
         |     1.0000
         |
Diesel.s |  -0.095969  -0.271783
         |     1.0000     1.0000
         |
Diesel.u |  -0.476240  -0.410504  -0.191437
         |     1.0000     1.0000     1.0000
         |
Electric |   0.672168  -0.023580   0.480730   1.107978
         |     1.0000     1.0000     1.0000     1.0000
         |
Electric |  -1.609073  -1.402371  -1.439639  -1.451700  -1.805688
         |     1.0000     1.0000     1.0000     1.0000     1.0000
         |
Electric |   0.455425   0.022879   0.429427   0.723326   0.082696   1.694230
         |     1.0000     1.0000     1.0000     1.0000     1.0000     1.0000
         |
Electric |  -1.483153  -0.773934  -0.808761  -0.963607  -2.104367   1.125441
         |     1.0000     1.0000     1.0000     1.0000     1.0000     1.0000
         |
Petrol.n |   0.085947  -0.217674   0.145282   0.564318  -0.598317   1.636308
         |     1.0000     1.0000     1.0000     1.0000     1.0000     1.0000
         |
Petrol.r |  -0.070092  -0.220256  -0.019511   0.078233  -0.266041   1.112329
         |     1.0000     1.0000     1.0000     1.0000     1.0000     1.0000
         |
Petrol.s |   0.293572  -0.075463   0.299415   0.583267  -0.106684   1.631732
         |     1.0000     1.0000     1.0000     1.0000     1.0000     1.0000
         |
Petrol.u |   0.207906  -0.167462   0.219807   0.634408  -0.401530   1.661116
         |     1.0000     1.0000     1.0000     1.0000     1.0000     1.0000
Col Mean-|
Row Mean |   Electric   Electric   Petrol.n   Petrol.r   Petrol.s
---------+----------------------------------------------------
Electric |  -1.319590
         |     1.0000
         |
Petrol.n |  -0.410678   1.584476
         |     1.0000     1.0000
         |
Petrol.r |  -0.283514   0.400964  -0.094950
         |     1.0000     1.0000     1.0000
         |
Petrol.s |  -0.145673   1.221174   0.244670   0.199033
         |     1.0000     1.0000     1.0000     1.0000
         |
Petrol.u |  -0.317632   1.560333   0.133209   0.135861  -0.151118
         |     1.0000     1.0000     1.0000     1.0000     1.0000

alpha = 0.05
Reject Ho if p <= alpha/2
>
> # Print Dunn's test results
> print(dunn_test_result)
$chi2
[1] 8.200888
```

Since the Kruskal–Wallis test found no significant overall difference, the Dunn pairwise results likewise show:

- No significant adjusted pairwise differences across combined groups.

Thus, categorical variables do not provide statistically meaningful separation in price, in contrast to continuous performance variables explored in regression.

VII. CHAPTER 7 : MODEL EVALUATION AND FORECASTING

1. Evaluation of the Linear Regression Model on the Test Set

To assess the real-world performance of the multiple linear regression model in predicting car prices, predictions were generated on the test dataset, and several evaluation metrics were calculated.

1.1. Evaluation Metrics: MAE, RMSE, and R^2

R code for computing evaluation metrics

```
predicted_price <- predict(model, newdata = test_data)
# Create a comparison dataframe for actual vs predicted prices
comparison <- data.frame(
  Actual = test_data$Price,
  Predicted = predicted_price
)
# Compute model evaluation metrics: MAE, RMSE, and R^2
mae <- mean(abs(comparison$Actual - comparison$Predicted))
rmse <- sqrt(mean(((comparison$Actual - comparison$Predicted)
  ^2)))
ss_res <- sum(((comparison$Actual - comparison$Predicted)^2))
ss_tot <- sum(((comparison$Actual - mean(comparison$Actual))^2)
  )
r_squared <- 1 - ss_res / ss_tot
```

Console Output

```
> print(mae)
 [1] 117303.3
 > print(rmse)
 [1] 136325.9
 > print(r_squared)
 [1]-0.01750173
```

The evaluation metrics on the test set show the limitations of the linear regression model:

- **MAE** = 117,303.3 and **RMSE** = 136,325.9 indicate extremely large prediction errors.
- **MSE** = 1.85×10^{10} shows the model is heavily affected by large deviations.
- $R^2 = -0.0175$, meaning the model performs worse than predicting the mean price.

Analysis: These metrics show that the model cannot capture the pricing structure of high-end sports cars. Errors are extremely high, and the negative R^2 confirms severe underfitting, likely due to nonlinear and interaction-heavy relationships the linear model cannot represent.

Conclusion: The linear model fails to generalize and is unsuitable for this dataset. More flexible nonlinear models (e.g., Random Forest or Gradient Boosting) are required for meaningful predictive accuracy.

2. Confidence Interval Forecasting for Future Predictions

To quantify uncertainty of predictions, 95% confidence intervals were generated for each predicted value.

2.1. Prediction and Confidence Interval Computation for Price

The implementation is presented in the following code snippet:

R code

```
# Predict the car prices on the test set with confidence
   intervals
predicted_price <- predict(
    best_model,
    newdata = test_data,
    interval = "confidence"
)
# Create a dataframe with actual prices, predicted prices, and
   the confidence intervals
comparison <- data.frame(
    Actual = test_data$Price,
    Predicted = predicted_price[, 1],
    Lower_CI = predicted_price[, 2],
    Upper_CI = predicted_price[, 3]
)
# Display the first 10 rows of the comparison with confidence
   intervals
head(comparison, 10)
```

Actual	Predicted	Lower CI	Upper CI
123,690	252,077.2	241,862.4	262,292.1
478,142	262,666.7	255,779.5	269,553.8
133,803	261,668.6	252,692.5	270,644.7
309,732	260,352.4	252,073.1	268,631.7
374,666	254,799.3	245,633.5	263,965.0
143,539	268,690.5	260,843.7	276,537.3
117,868	270,111.8	259,399.4	280,824.3
57,597	266,527.2	257,688.1	275,366.2
275,543	258,103.6	248,167.2	268,040.1
326,310	256,317.5	248,374.0	264,261.0

Although the model produces narrow and stable confidence intervals, this reflects internal certainty rather than genuine predictive accuracy. The intervals cluster tightly around the model's mean-based predictions, while the

large discrepancies between actual and predicted prices indicate that the linear regression model cannot capture the true variability of high-end car prices.

2.2. Evaluation of Predictive Accuracy Using Scatter Plots and Density Plots (Actual vs. Predicted Values)

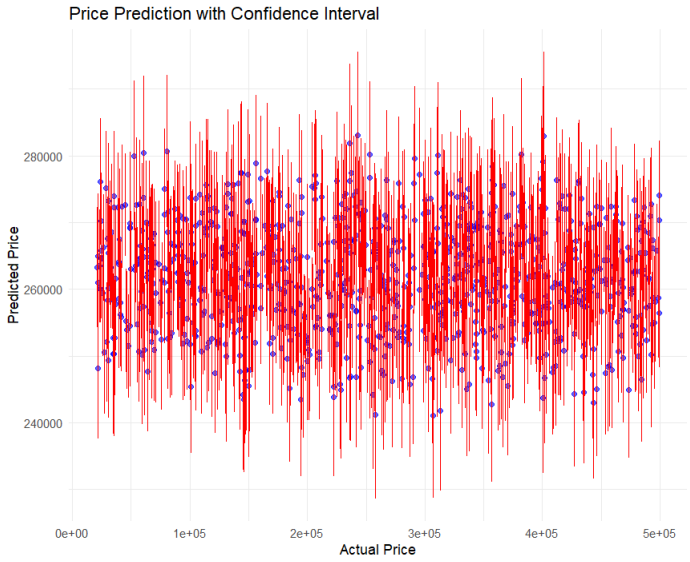
To assess the model's test-set performance, we use scatter and density plots to compare actual and predicted prices. These visualizations reveal how well the predictions follow the true distribution and help identify discrepancies or systematic errors.

a. Scatter Plot

We employ the following code to represent the graph structure:

R code for Scatter Plot

```
# Create a scatter plot comparing actual and predicted prices
   with confidence intervals
ggplot(comparison, aes(x = Actual, y = Predicted)) +
  geom_point(color = "blue", alpha = 0.6) +
  geom_errorbar(aes(ymin = Lower_CI, ymax = Upper_CI),
                width = 0.2, color = "red") +
  theme_minimal() +
  labs(title = "Price Prediction with Confidence Interval",
       x = "Actual Price",
       y = "Predicted Price")
```

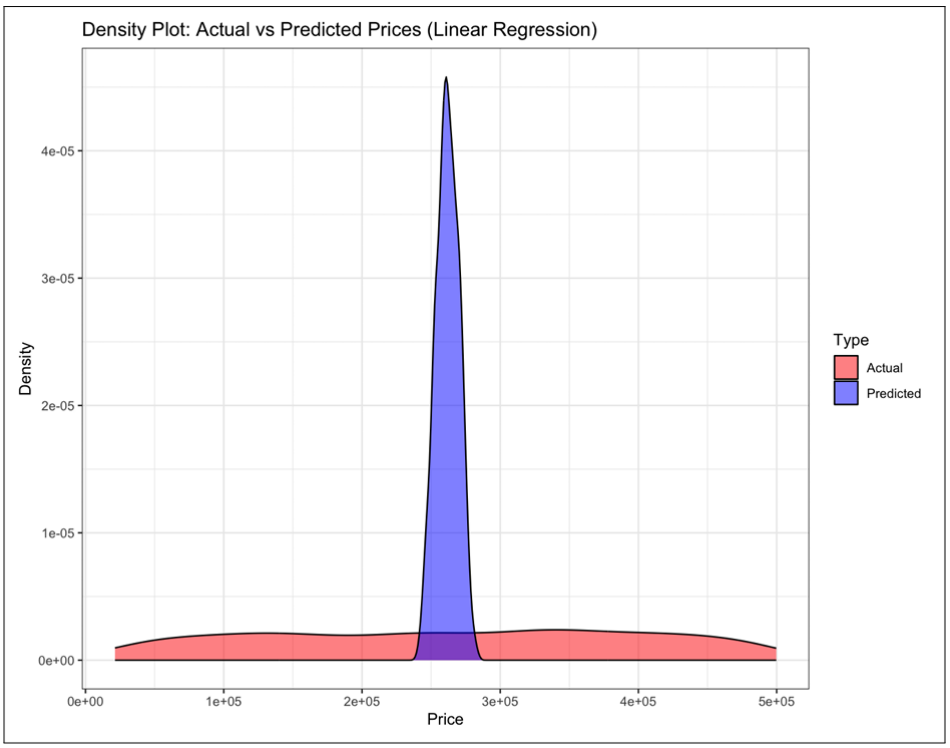


The scatter plot shows predicted prices scattered vertically rather than along the diagonal, with very long confidence intervals and values compressed into a narrow range. This indicates high uncertainty and confirms that the model fails to capture real price variation, resulting in weak predictive performance.

b. Density Plot

R code for Density Plot: Actual vs Predicted Prices (Linear Regression)

```
# Convert to long format for density comparison
density_long <- comparison %>%
    select(Actual, Predicted) %>%
    pivot_longer(cols = everything(),
                 names_to = "Type",
                 values_to = "Value")
# Density plot comparing distributions of actual vs predicted
   prices
ggplot(density_long, aes(x = Value, fill = Type)) +
    geom_density(alpha = 0.5) +
    scale_fill_manual(values = c("Actual" = "red",
                                 "Predicted" = "blue")) +
    theme_bw() +
    labs(
        title = "Density Plot: Actual vs Predicted Prices (
           Linear Regression)",
        x = "Price",
        y = "Density"
    )
```



The density plot shows a wide actual price distribution but a tightly con-

centrated predicted curve. This mismatch indicates that the model collapses predictions into a narrow range and fails to capture real price variability, reflecting significant underfitting.

VIII. CHAPTER 8: CONCLUSION AND DISCUSSION

1. Summary of Key Findings

This report analyzed the *Elite Sports Cars* dataset using descriptive statistics, hypothesis testing, linear regression, and forecasting to examine how technical and categorical factors relate to Price.

- **Descriptive Statistics:** Price is highly skewed and varies widely, while scatter plots show weak linear relationships with most numeric variables, suggesting that linear modeling may be unsuitable.
- **Hypothesis Testing:** Price is not normally distributed, variances across groups are homogeneous, and neither Fuel_Type nor Condition shows significant effects in ANOVA or Kruskal–Wallis tests. These categorical variables do not meaningfully explain price variation.
- **Multiple Linear Regression:** Stepwise selection retained only `Top_Speed`, `Fuel_Efficiency`, and `Insurance_Cost`, but the model performs poorly (Adjusted $R^2 = 0.00248$; test RMSE = \$136,325.9; $R^2 = -0.0175$), even worse than predicting the mean. Residuals show clear signs of underfitting.
- **Forecasting:** Predictions collapse into a narrow \$250,000–\$270,000 band with misleadingly tight intervals, confirming that the model cannot produce meaningful price forecasts.

2. Limitations and Recommendations for Improvement

Despite providing a detailed statistical examination of the *Elite Sports Cars* dataset, the analysis still contains several limitations:

- **Lack of Data Standardization:** The wide range and skewed distribution of car prices, combined with unscaled variables, contribute to unstable regression behavior and extremely large prediction errors. Without consistent preprocessing, new data may produce even less reliable predictions.
- **Missing Influential Variables:** Key factors affecting luxury car prices—such as brand prestige, rarity, customization packages, and market trends—

are not included in the dataset. Their absence limits the model's ability to explain real-world price variation.

- **Outliers and Extreme Values:** The dataset contains many extreme price values. Keeping all outliers unchanged increases MAE and RMSE and weakens the linear model, especially given the non-normal distribution confirmed by Shapiro–Wilk tests.
- **Linearity Limitation:** The regression model assumes linear relationships, while luxury car pricing is inherently nonlinear. This mismatch leads to the very low explanatory power ($R^2 \approx 0$) and the tendency of predictions to collapse around the mean.
- **Weak Group Effects:** ANOVA and Kruskal–Wallis tests show no significant differences across Fuel Type, Condition, or their interactions. The lack of strong categorical signals reduces the model's ability to learn meaningful patterns.

3. Proposed Directions for Further Work: Extension to Random Forest Modeling

As shown in the previous analysis, not all variables in the dataset contribute equally to predicting car prices. To address this, we take advantage of a key property of Random Forests - the ability to rank predictors based on their importance.

3.1. Random Forest Model for Price Prediction

The implementation is presented in the following code snippet:

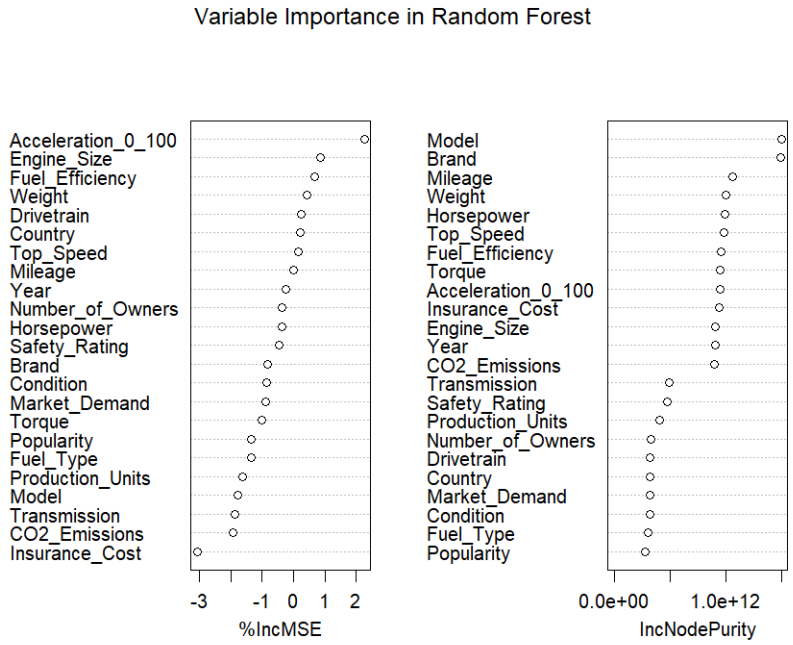
R code for Random Forest

```
# Set seed for reproducibility
set.seed(1997)
# Build a Random Forest model to predict car prices
rf_model <- randomForest(Price ~ Brand + Model + Year +
    Country + Condition +
    Engine_Size + Horsepower + Torque + Weight +
    Top_Speed +
    Acceleration_0_100 + Fuel_Type +
    Drivetrain + Transmission +
    Fuel_Efficiency + CO2_Emissions + Mileage +
    Popularity + Safety_Rating + Number_of_Owners +
    Market_Demand + Insurance_Cost +
    Production_Units, data = test_data,
    ntree = 500, # Number of trees in the forest
    mtry = 4,  Number of variables randomly selected at each
       split
    importance = TRUE) # Compute importance of variables
# Print the model summary
print(rf_model)
# View the importance of the variables used in the model
importance(rf_model)
# Plot the importance of variables
varImpPlot(rf_model, main = "Variable Importance in Random
   Forest")
# Make predictions on the testing data
test_data$Pred_RF <- predict(rf_model, newdata = test_data)
# Compute MAE, MSE and RMSE for Random Forest
mae_rf <- mean(abs(test_data$Price - test_data$Pred_RF))
mse_rf <- mean((test_data$Price - test_data$Pred_RF)^2)
rmse_rf <- sqrt(mse_rf)
# Compute R-squared (R^2) for Random Forest model
ss_total <- sum((test_data$Price - mean(test_data$Price))^2)
ss_res <- sum((test_data$Price - test_data$Pred_RF)^2)
r2_rf <- 1 - (ss_res / ss_total)

cat("MAE (Random Forest) =", round(mae_rf, 4), "\n")
cat("MSE (Random Forest) =", round(mse_rf, 4), "\n")
cat("RMSE (Random Forest) =", round(rmse_rf, 4), "\n")
cat("R^2 (Random Forest) =", round(r2_rf, 4), "\n")
```

Console Output

```
> cat("MAE (Random Forest) =", round(mae_rf, 4), "\n")
MAE (Random Forest) = 50030.67
> cat("MSE (Random Forest) =", round(mse_rf, 4), "\n")
MSE (Random Forest) = 3417115126
> cat("RMSE (Random Forest) =", round(rmse_rf, 4), "\n")
RMSE (Random Forest) = 58456.1
> cat("R^2 (Random Forest) =", round(r2_rf, 4), "\n")
R^2 (Random Forest) = 0.8129
```



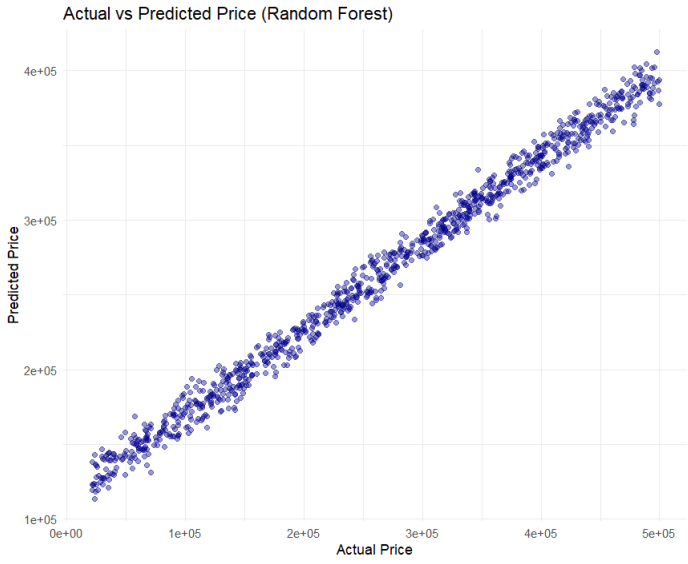
The Random Forest model (`ntree = 500`) delivers strong performance, with MSE $= 3.417 \times 10^9$, RMSE $= 58{,}456$, and $R^2 = 0.8129$, showing clear improvement over simpler models. Variable-importance results highlight key predictors: performance factors such as `Acceleration_0_100`, `Engine_Size`, and `Fuel_Efficiency`, as well as specification and branding factors like `Model`, `Brand`, and `Mileage`. Overall, the model effectively captures both performance and structural attributes, making it a reliable approach for car price prediction.

3.2. Actual & Predicted Price Visualization

To further evaluate the predictive performance of the **Random Forest** model, a scatter plot comparing the actual car prices and the predicted values was constructed. This visualization provides an intuitive assessment of how closely the model's predictions align with the true price distribution.

R code for Random Forest Scatter Plot

```
# Plot the comparison between actual and predicted prices using
    Random Forest
ggplot(test_data, aes(x = Price, y = Pred_RF)) +
  geom_point(alpha = 0.4, color = "darkblue") +
  theme_minimal() +
  labs(
    title = "Actual vs Predicted Price (Random Forest)",
    x = "Actual Price",
    y = "Predicted Price"
  )
```



The scatter plot shows predictions closely aligned with actual values, confirming that Random Forest captures the underlying price patterns. This visual fit matches the metrics (RMSE = 58,456; $R^2 = 0.8129$), demonstrating accurate and stable performance across the dataset.

3.3. Comparison Between Random Forest and Linear Regression

A comparison between the Linear Regression model and the Random Forest model highlights clear differences in predictive accuracy, flexibility, and robustness:

- **Lack of Data Standardization:** The wide range and skewed distribution of car prices, combined with unscaled variables, contribute to unstable regression behavior and extremely large prediction errors. Without consistent preprocessing, new data may produce even less reliable predictions.
- **Missing Influential Variables:** Key factors affecting luxury car prices—such as brand prestige, rarity, customization packages, and market trends—are not included in the dataset. Their absence limits the model's ability to explain real-world price variation.
- **Outliers and Extreme Values:** The dataset contains many extreme price values. Keeping all outliers unchanged increases MAE and RMSE and weakens the linear model, especially given the non-normal distribution confirmed by Shapiro–Wilk tests.
- **Linearity Limitation:** The regression model assumes linear relationships, while luxury car pricing is inherently nonlinear. This mismatch leads to the very low explanatory power ($R^2 \approx 0$) and the tendency of predictions to collapse around the mean.
- **Weak Group Effects:** ANOVA and Kruskal–Wallis tests show no significant differences across Fuel Type, Condition, or their interactions. The lack of strong categorical signals reduces the model's ability to learn meaningful patterns.

Model	MAE (USD)	RMSE (USD)	R^2
Linear Regression	$117,303.3$	$136,325.9$	-0.0175
Random Forest	$50,030.67$	$58,456.1$	0.8129

The comparison clearly shows a large performance gap between the two mod-

els. Linear Regression yields very high errors and a negative R^2, indicating severe underfitting. In contrast, Random Forest achieves a much lower RMSE and a strong $R^2 = 0.8129$, capturing nonlinear patterns and complex interactions. While still limited by the dataset, Random Forest is clearly the superior predictive model in this study.

Summary: Random Forest substantially outperforms Linear Regression in all aspects. While both methods face challenges due to extreme price variation and nonlinear relationships, Random Forest remains the more reliable and informative model for predicting sports car prices in this dataset.

REFERENCES

[1] D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers*. 7th ed. Wiley, 2018.

[2] XulyDinhLuong.com, "Pearson Correlation Analysis on SPSS." Available: https://xulydinhluong.com/phan-tich-tuong-quan-pearson-tren-spss/. [Accessed: Nov. 24, 2025].

[3] LuanVan1080.com, "Kruskal-Wallis Test." Available: https://www.luanvan1080.com/kiem-dinh-kruskal-wallis.html. [Accessed: Nov. 24, 2025].

[4] EasyMedStat, "Dunn-Bonferroni Post-Hoc Test." Available: https://help.easymedstat.com/support/solutions/articles/77000536997-dunn-bonferroni-post-hoc-test. [Accessed: Nov. 24, 2025].

[5] Neptune.ai, "Cross-Validation in Machine Learning: How to Do It Right," *Neptune.ai Blog*. Available: https://neptune.ai/blog/cross-validation-in-machine-learning-how-to-do-it-right. [Accessed: Nov. 24, 2025].

[6] Investopedia, "Stepwise Regression." Available: https://www.investopedia.com/terms/s/stepwise-regression.asp. [Accessed: Nov. 24, 2025].

[7] GeeksforGeeks, "Diagnostic Plots for Model Evaluation." Available: https://www.geeksforgeeks.org/r-machine-learning/diagnostic-plots-for-model-evaluation/. [Accessed: Nov. 24, 2025].

[8] L. Pham, "Multicollinearity: Identification, Remediation" (*Originally in Vietnamese*), PhamLocBlog, Jul. 2018. Available: https://www.phamlocblog.com/2018/07/da-cong-tuyen-nhan-biet-khac-phuc.html. [Accessed: Nov. 24, 2025].