# Statistical Analysis on Laptop Price Dataset

Professor: **Silvia Salini**

Presenter: **Maedeh Rabiee**

Matriculation number:**14175A**

Major**: Data science for economics**

September 2024

# Table of Contents

# 1. Introduction

In today's rapidly evolving laptop market, understanding the relationships between product specifications and pricing is essential for both consumers and manufacturers. With a vast range of models varying in RAM, storage, screen size, processor type, and other key features, it becomes critical to analyze patterns and trends that influence pricing. By exploring these relationships, we can gain insights into how various technical aspects of laptops affect their market value and consumer preferences.

The goal of this analysis is twofold: first, to identify trends and relationships between laptop specifications and their pricing; and second, to categorize laptops into distinct clusters based on their specifications. By employing clustering techniques such as K-Means and Hierarchical Clustering, we aim to group laptops with similar attributes and uncover meaningful categories, which can help segment the market and identify common characteristics within each group.

# 2. Abstract

In this report, we want to perform a series of data analysis steps including **data visualization** to explore trends, **data preprocessing** to handle missing values and standardize the data, and **operations on numerical and categorical data** to prepare the data for machine learning.

We will then apply **supervised learning algorithms** to predict laptop prices and **unsupervised learning algorithms**, for clustering similar laptops. The performance of these models will be evaluated to ensure the accuracy and relevance of the insights.

# 3. Dataset

This dataset provides a comprehensive collection of information on various laptops, enabling a detailed analysis of their specifications and pricing. It encompasses a wide range of laptops, encompassing diverse brands, models, and configuration. https://www.kaggle.com/datasets/juanmerinobermejo/laptops-price-dataset/data

Here is an overview of the columns(features):

- **Laptop**: A description of the laptop, including model and some specifications.
- **Status**: Indicates whether the laptop is new.
- **Brand**: The brand of the laptop (e.g., Asus, MSI, HP).
- **Model**: The specific model name.
- **CPU**: The type of processor (e.g., Intel Core i5, Intel Celeron).
- **RAM**: The amount of RAM (in GB).
- **Storage**: The storage capacity (in GB).
- **Storage type**: The type of storage (e.g., SSD).

- **GPU**: The graphics processing unit (e.g., RTX 3050).
- **Screen**: The screen size (in inches).
- **Touch**: Indicates whether the laptop has a touchscreen (Yes/No).
- **Final Price**: The final price of the laptop.

Here is a breakdown for each column in the dataset:

- **Laptop** (character): 2,160 unique laptop descriptions.
- **Status** (character): 2 values (New, Refurbished).
- **Brand** (character): 27 unique brands (e.g., Asus, HP, MSI).
- **Model** (character): 121 unique models.
- **CPU** (character): 28 different processor types (e.g., Intel Core i5, AMD Ryzen 5, Apple M1).
- **RAM** (integer): 9 distinct values (e.g., 4 GB, 8 GB, 16 GB, up to 128 GB).
- **Storage** (integer): 11 unique storage capacities (e.g., 256 GB, 512 GB, 1 TB).
- **Storage Type** (character): 2 values (SSD, eMMC).
- **GPU** (character): 44 unique graphics processors (e.g., RTX 3050, GTX 1650).
- **Screen** (numeric): 30 unique screen sizes (e.g., 15.6", 14", 17.3").
- **Touch** (character): 2 values (Yes, No).
- **Final Price** (numeric): Various unique prices reflect a wide range of laptop prices, from low to high.

```
> summary(data)
    Laptop             Status             Brand              Model
 Length:2160        Length:2160        Length:2160        Length:2160
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character


     CPU                RAM              Storage        Storage.type
 Length:2160        Min.   :  4.00   Min.   :   0.0   Length:2160
 Class :character   1st Qu.:  8.00   1st Qu.: 256.0   Class :character
 Mode  :character   Median : 16.00   Median : 512.0   Mode  :character
                    Mean   : 15.41   Mean   : 596.3
                    3rd Qu.: 16.00   3rd Qu.:1000.0
                    Max.   :128.00   Max.   :4000.0

     GPU                Screen          Touch            Final.Price
 Length:2160        Min.   :10.10   Length:2160        Min.   : 201.1
 Class :character   1st Qu.:14.00   Class :character   1st Qu.: 661.1
 Mode  :character   Median :15.60   Mode  :character   Median :1031.9
                    Mean   :15.17                      Mean   :1312.6
                    3rd Qu.:15.60                      3rd Qu.:1709.0
                    Max.   :18.00                      Max.   :7150.5
                    NA's   :4
```

# 4. Data Preprocessing

To handle missing or inconsistent values, this dataset needs to be preprocessed as it hasn't been cleansed.

## 4.1 Handling missing values

Missing values can lead to biased results or prevent certain algorithms from functioning correctly. Therefore, it's essential to impute or handle these missing values appropriately to maintain the integrity of the dataset.

The initial step was to replace any empty strings or cells containing NA values with “NA”, ensuring that missing values were consistently represented.

We checked for missing values in each column, which revealed the following:

GPU had the highest number of missing values (1371).

Screen had a few missing values (4).

Storage.type had a moderate number of missing values (42).

## 4.2 Handling Missing Values in GPU:

Given that the GPU column is categorical, missing values were replaced with "Unknown" to standardize the dataset and avoid introducing bias by using a more frequent category.

## 4.3 Imputing Missing Values in Screen:

For the Screen column, which contains numerical data, the mean of the existing values was used to impute missing entries. This is a common and straightforward imputation method.

## 4.4 Imputing Missing Values in Storage.type:

The mode of the Storage.type column was calculated and used to replace any missing values. This approach ensures that the most common storage type in the dataset is assumed for any missing entries, which is a reasonable assumption when no other information is available.

## 4.5 Replacing Zero Values in Storage:

Zero values in the Storage column were likely placeholders or incorrect entries. To address this, the median of the non-zero values was calculated and used to replace these zeros, which is a robust method.

**Theoretical Background of the Used Methods**

Mean Imputation: A simple and widely used method for numerical data, mean imputation replaces missing values with the average of the available data. It is most effective when the data is normally distributed.

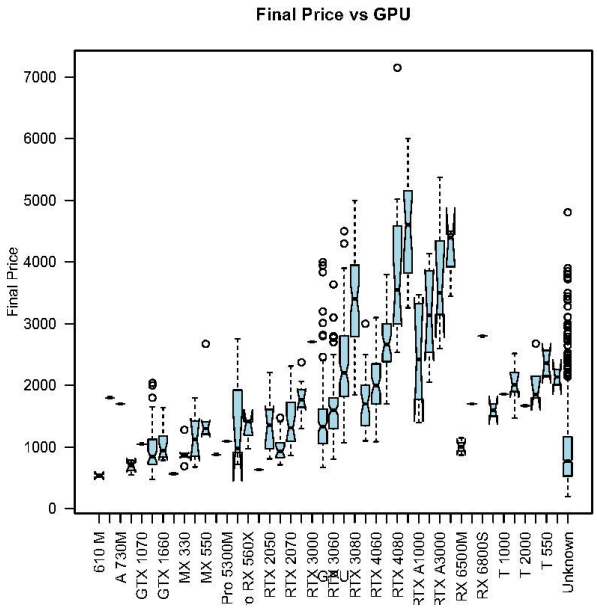
Mode Imputation: For categorical data, the mode (most frequent value) is a common choice for imputation. It is particularly useful when a single category is dominant in the dataset.

Median Imputation: The median is a robust measure of central tendency, particularly useful when the data contains outliers or is skewed. It is often preferred over the mean for imputing missing values in such cases.

Finally, after all imputations, a final check was conducted to ensure that no missing values remained, confirming that the dataset was ready for further analysis.

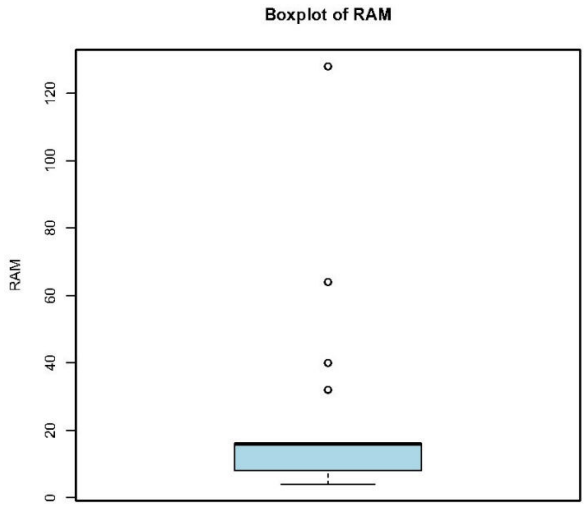
# 5. Visualizations

## 5.1 Final Price vs GPU (Boxplot)



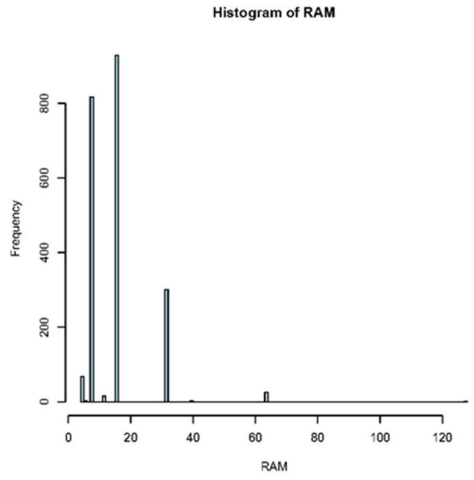

The boxplot compares the distribution of Final Price across different GPU models.
The wide range in the final price for each GPU type suggests a significant variation in the cost associated with different GPU models.
The GPU labeled as "Unknown" has a large number of data points and a wide range in prices, possibly indicating laptops with varying specifications that are not well defined in terms of GPU.
Some GPUs, particularly high-end models like the RTX series, have a higher median price, which aligns with their performance and market position as premium GPUs.
There are noticeable outliers in almost all GPU categories, indicating that some laptops are priced much higher or lower than the typical range for that GPU.

### 5.2 Boxplot of RAM



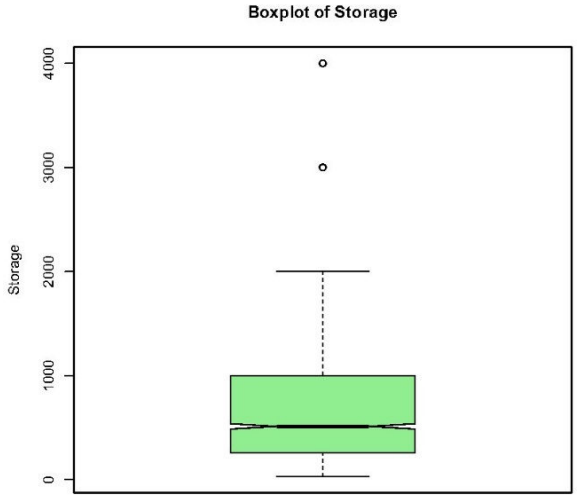

This boxplot shows the distribution of RAM sizes across the laptops in the dataset.
The majority of laptops have RAM sizes clustered around a lower range, with a few outliers extending up to higher RAM values, including a few above 64 GB.Most laptops have relatively moderate RAM sizes, which are typical for general consumer laptops.

### 5.3 Histogram of RAM



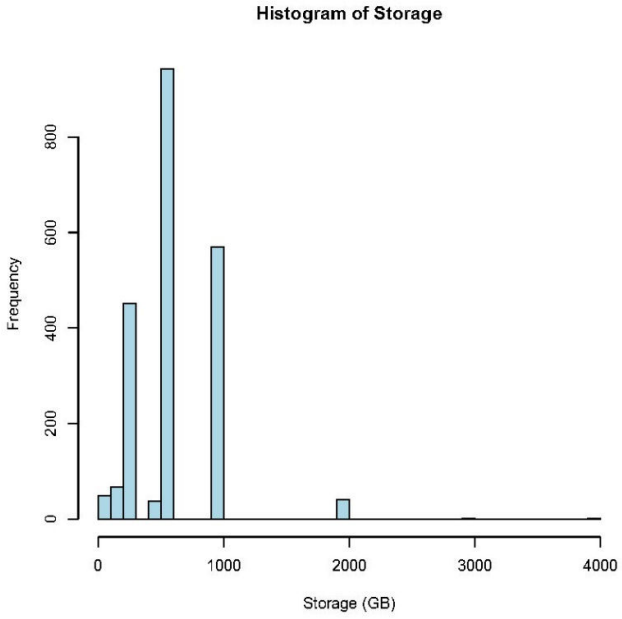

The histogram displays the frequency distribution of different RAM sizes.
The skewed nature of the distribution suggests that most laptops in the dataset are equipped with mid-range RAM, typical for standard use cases.

## 5.4 Boxplot of Storage



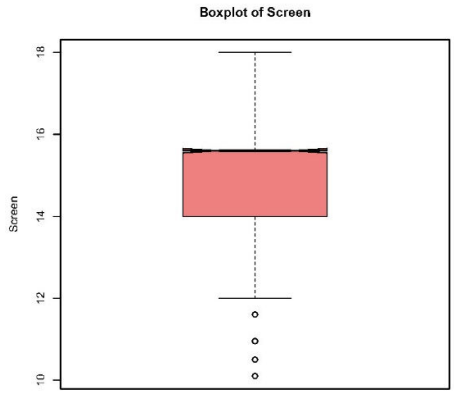

The presence of outliers suggests that some laptops in the dataset have exceptionally high storage capacities, which could be targeted towards professionals needing significant data storage.

## 5.5 Histogram of Storage



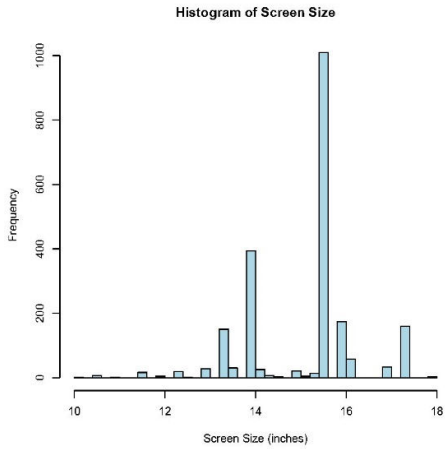
Similar to the boxplot, the distribution is skewed with most laptops having storage sizes around common capacities such as 256 GB, 512 GB, and 1 TB.
The histogram indicates that very few laptops exceed 2 TB of storage.
There is a sharp decline in frequency as storage capacity increases, indicating that higher storage laptops are less common and likely more expensive.

## 5.6 Boxplot of Screen Size



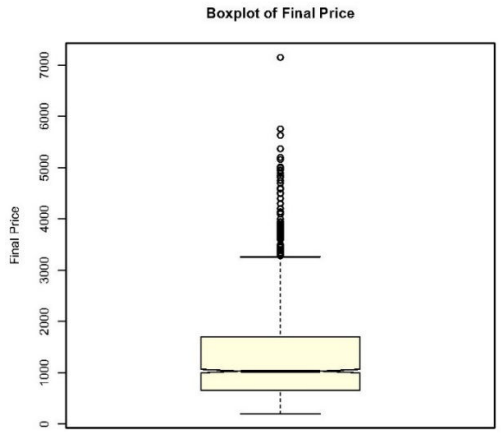

There are a few outliers with smaller or larger screens, likely representing ultra-portable laptops and large gaming or workstation laptops, respectively.

## 5.7 Histogram of Screen Size



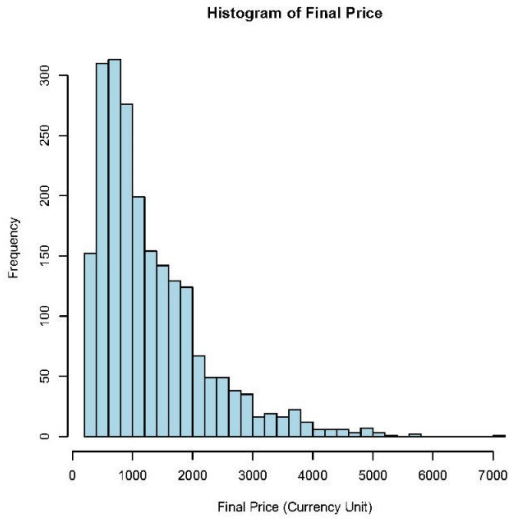

The histogram shows that most laptops have screen sizes clustered around 15 to 16 inches.
There is a clear peak at 15.6 inches, which is a common screen size for mainstream laptops.
Smaller screens around 13-14 inches and larger screens around 17-18 inches are less frequent, indicating that they cater to more niche markets (e.g., ultra-portables and gaming laptops).

### 5.8 Boxplot of Final Price




There are many outliers on the higher end, indicating a significant number of laptops that are priced well above the median, likely due to premium features or high-performance components.

### 5.9 Histogram of Final Price




The histogram shows that the distribution of laptop prices is heavily skewed towards the lower end, with a large number of laptops priced between 500 and 1500 currency units.
There is a long tail extending towards the higher prices, reflecting the high-end laptops that are priced significantly higher than the average.
The skewed distribution is typical in consumer electronics, where most products are within the reach of the average consumer, with a smaller segment catering to premium markets.Most laptops in the dataset fall within standard ranges for these features, but the presence of outliers indicates a diverse market

# 6. Operation on Numerical data

The numerical variables in the dataset are **RAM**, **Storage**, **Screen**, and **Final Price**

## 6.1 Handling Outliers:

We started preprocessing the laptop dataset by filtering and modifying specific attributes to remove extreme values and refine the data for better analysis. The process involved removing outliers based on our knowledge of typical laptop specifications available in today's market, rather than strictly adhering to statistical definitions of outliers, such as those indicated by **Box PlotS**.

### 6.1.1 Removing Laptops with More Than 64 GB of RAM

The original dataset contained laptops with RAM capacities ranging from 4 GB to 128 GB. Given that laptops with more than 64 GB of RAM are rare and typically designed for specialized, high-performance tasks (such as servers or advanced data processing), they were treated as outliers.

### 6.1.2 Removing Laptops with Storage Greater Than 2000 GB

The dataset originally included laptops with storage capacities as high as 4 TB (4000 GB). However, laptops with more than 2 TB of storage are rare in most consumer markets today. High-capacity storage laptops are typically found in specialized sectors such as gaming or enterprise computing, which are outside the scope of most common-use cases.

### 6.1.3 Capping the Final Prices at 5000 Units

The dataset included laptops with prices exceeding $7,000. These extremely high-priced laptops, while available, are outliers in most consumer markets. The decision to cap the **Final Price** at $5,000 ensures that excessively priced laptops do not disproportionately influence any analysis or model building.

## 6.2 Skewness Transformation

We analyzed the skewness of four key numerical variables: **RAM**, **Storage**, **Screen**, and **Final Price**. The goal was to reduce skewness and improve data distribution for better model performance. Skewness indicates the asymmetry of data, and highly skewed variables can negatively affect machine learning models.

### 6.2.1 Original Skewness Values

- **RAM**: 2.03
- **Storage**: 1.32
- **Screen**: -0.58
- **Final Price**: 1.52

### 6.2.2 Transformations Applied

To reduce the skewness of these variables, the following transformations were applied:

- **RAM**: A log transformation ($\log(\text{RAM} + 1)$) was applied to reduce the positive skewness.
- **Storage**: A square root transformation ($\text{sqrt}(\text{Storage})$) was applied, which is often useful for reducing positive skewness in variables with wide-ranging values.

- **Screen**: A reverse log transformation (log(max(Screen + 1) - Screen)) was applied to handle the slight negative skewness by reflecting the distribution and transforming the values.
- **Final Price**: A log transformation (log(Final.Price + 1)) was applied to reduce the positive skewness in the price data.

### 6.2.3 Skewness After Transformation

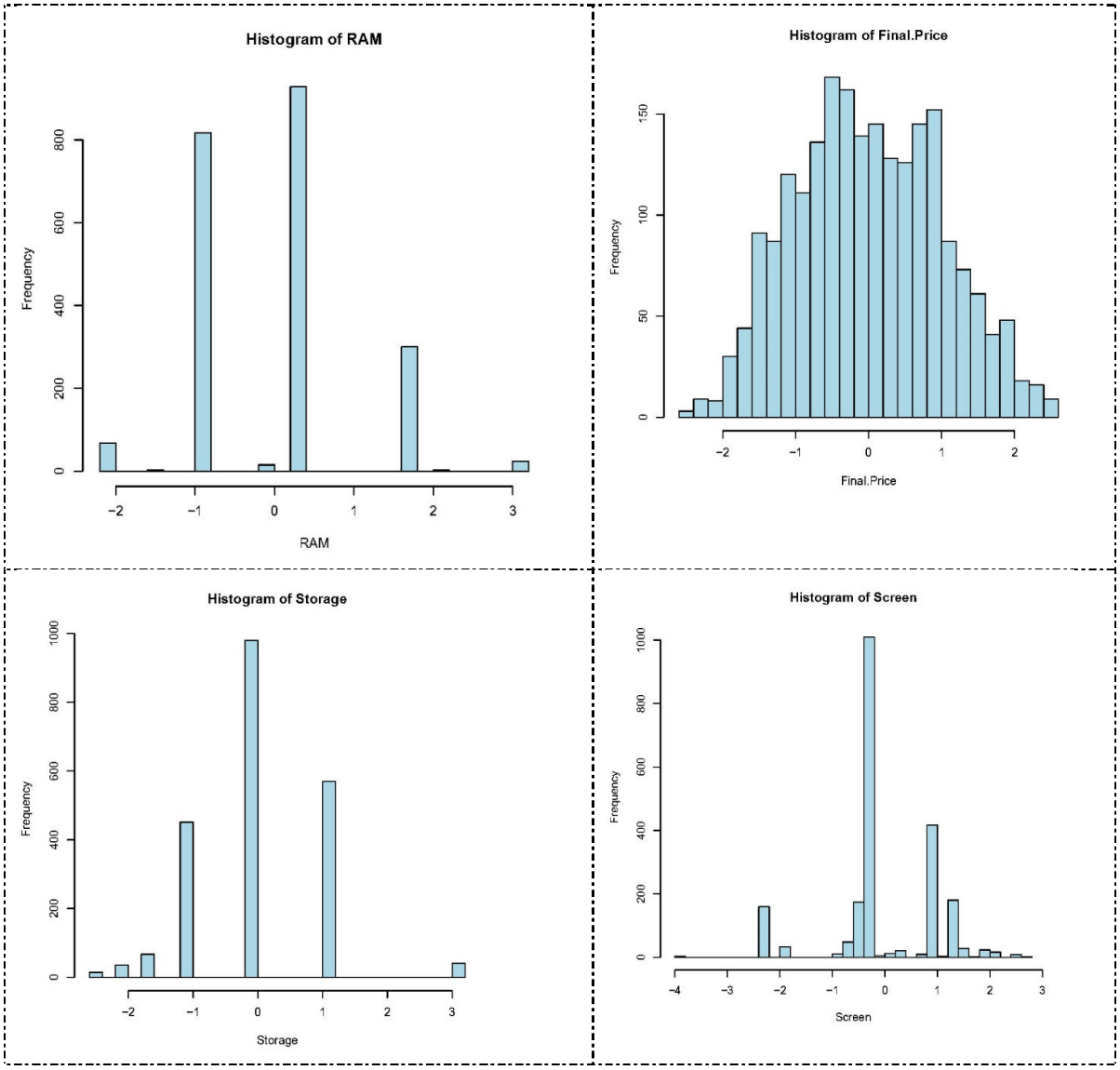
- **RAM**: 0.39
- **Storage**: 0.31
- **Screen**: -0.50
- **Final Price**: 0.10
- Transformations successfully reduced skewness, making the data more normally distributed.

## 6.3 scaling

The scale() function was applied to the numerical variables defined in numerical_vars (which includes "RAM", "Storage", "Screen", and "Final.Price").
Scaling involves standardizing each variable to have a mean of 0 and a standard deviation of 1. This process transforms the data into a common scale without distorting differences in the ranges of values.
Why Scaling is Important?
Numerical features RAM, Storage, Screen size, and Final Price originally have different ranges. For example, RAM might range from 4 to 64 GB, while Storage could range from 128 to 4000 GB. Without scaling, models that are sensitive to the magnitude of numerical values (like K-Means clustering, SVMs, or linear regression) might assign disproportionate importance to features with larger ranges, skewing results.

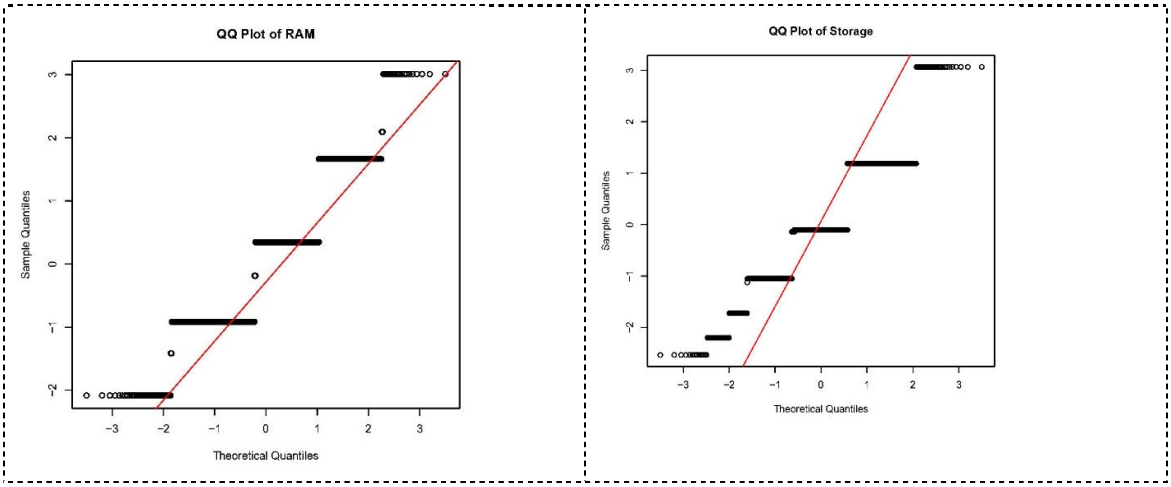
## 6.4 Checking Normality

### 6.4.1 histogram of numericals after scaling:



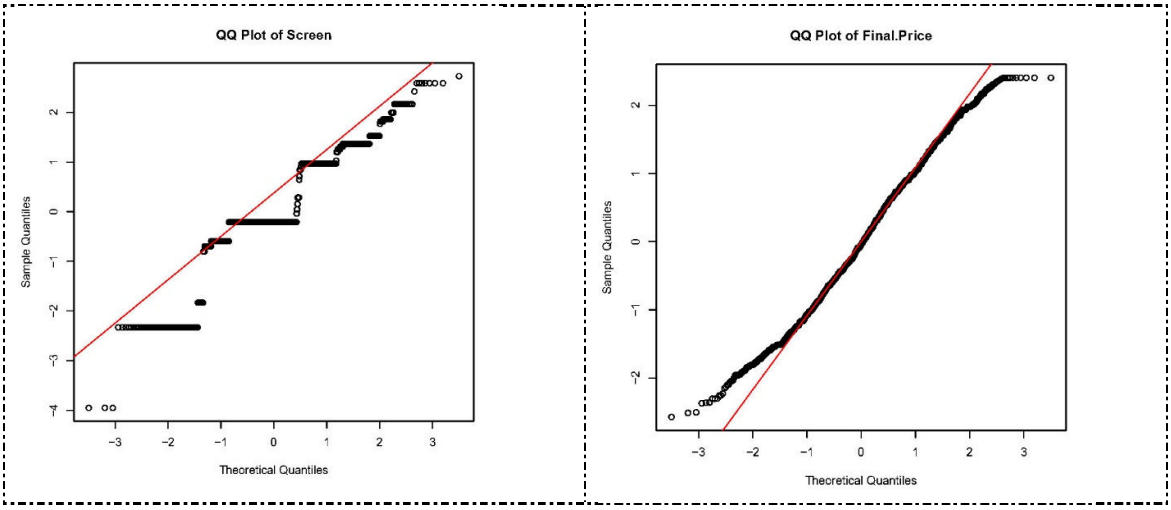

### 6.4.2 QQ Plots of numerical (RAM, Storage, Screen, Final Price)

QQ plots help assess the normality of the distributions by comparing the quantiles of the data with the theoretical quantiles from a normal distribution.




RAM: The QQ plot for RAM indicates that the data is not normally distributed. The points deviate significantly from the red line, especially at the extremes.

Storage: The QQ plot for storage also shows deviation from normality, with the points falling off the line, particularly in the tails, suggesting skewness

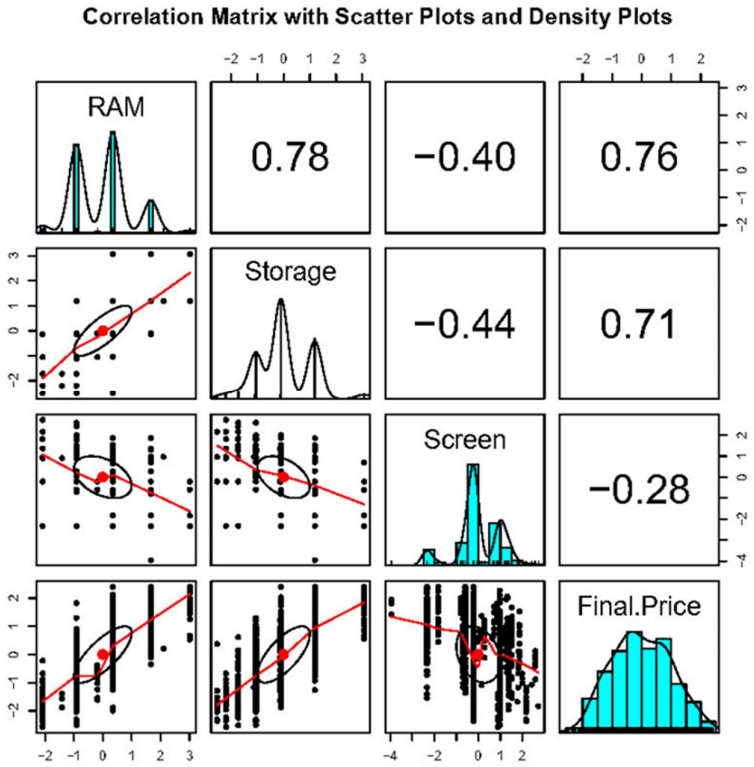



Screen Size: The QQ plot for screen size suggests non-normality, with significant deviations from the line, especially at the lower and upper ends.
Final Price: The QQ plot for final price shows some deviation from normality but less pronounced than the other variables. The data is moderately skewed.

### 6.4.3 Shapiro-Wilk Test

it is used to test the null hypothesis that the data is normally distributed.
RAM: W = 0.856, p-value = 1.505e-40. The p-value is extremely low, rejecting the null hypothesis and indicating that the RAM data is not normally distributed.
Storage: W = 0.882, p-value = 9.991e-38. Similar to RAM, the low p-value indicates non-normality.
Screen: W = 0.872, p-value = 7.766e-39. The screen size data is also not normally distributed.
Final Price: W = 0.991, p-value = 6.224e-10. Although the p-value is low, the deviation from normality in the final price is less pronounced than in the other variables.

## 6.5 Interpretation of the Correlation Matrix



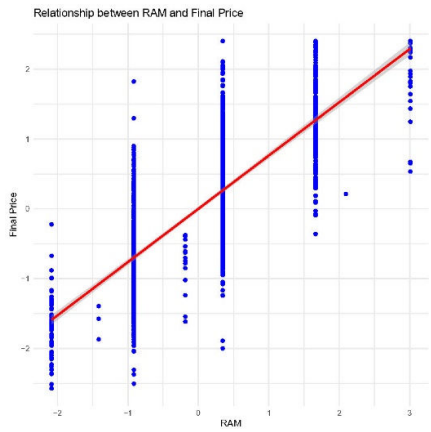

**Correlation Matrix:**

- **RAM and Final.Price (0.76):**
  - **Interpretation:** There is a strong positive correlation between RAM and Final.Price, indicating that as the amount of RAM in a laptop increases, the final price tends to increase as well.
- **Storage and Final.Price (0.71):**
  - **Interpretation:** There is also a strong positive correlation between Storage and Final.Price. This suggests that laptops with larger storage capacities tend to be more expensive.
- **Screen and Final.Price (-0.28):**
  - **Interpretation:** The correlation between Screen size and Final.Price is negative but weak. This means that larger screens are slightly associated with lower prices, but the relationship is not strong.
- **RAM and Storage (0.78):**
  - **Interpretation:** There is a strong positive correlation between RAM and Storage. Laptops with more RAM tend to also have more storage capacity, which makes sense as higher-end laptops often come with both larger RAM and storage.
- **Screen and RAM (-0.40):**
  - **Interpretation:** There is a moderate negative correlation between Screen size and RAM. Laptops with larger screens may have slightly less RAM, which could reflect design trade-offs or market segmentation.
- **Screen and Storage (-0.44):**

- **Interpretation:** Similarly, there is a moderate negative correlation between Screen size and Storage, suggesting that laptops with larger screens might have less storage, possibly reflecting differences in design priorities or target markets.

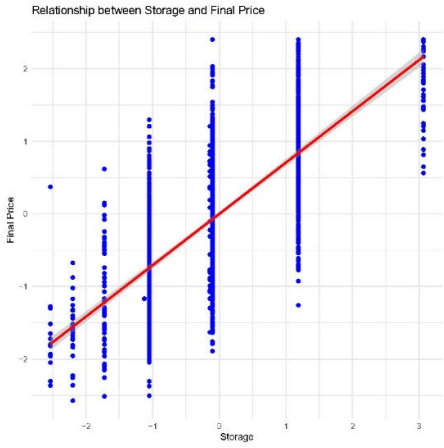
## 6.6 Scatter Plots of Numerical Variables Against Final Price)

**RAM vs Final.Price:**



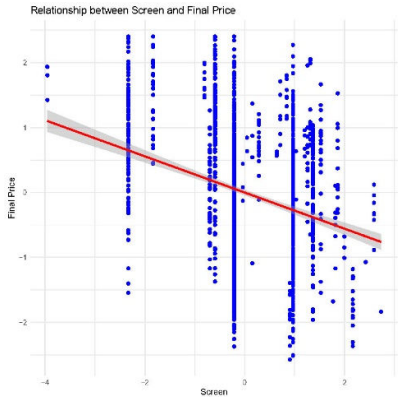

The scatter plot with a linear regression line shows a positive relationship between RAM and Final.Price. As RAM increases, the Final.Price tends to increase as well, which makes sense as more RAM typically corresponds to higher-performance laptops that are priced higher.

**Storage vs Final.Price:**




Similar to RAM, there is a positive relationship between Storage and Final.Price.
Laptops with larger storage capacities are generally more expensive, reflecting the additional cost of more or higher-quality storage hardware.

**Screen Size vsFinal Price:**




The scatter plot shows a negative relationship between Screen size and Final.Price.
As screen size increases, the final price tends to decrease slightly, as indicated by the downward slope of the red regression line.
This might suggest that larger screens are not necessarily associated with higher prices in this dataset. It could be that larger screens are more common in budget or mid-range laptops.

## 6.7 Interpretation of VIF (Variance Inflation Factor)

**What is VIF?**

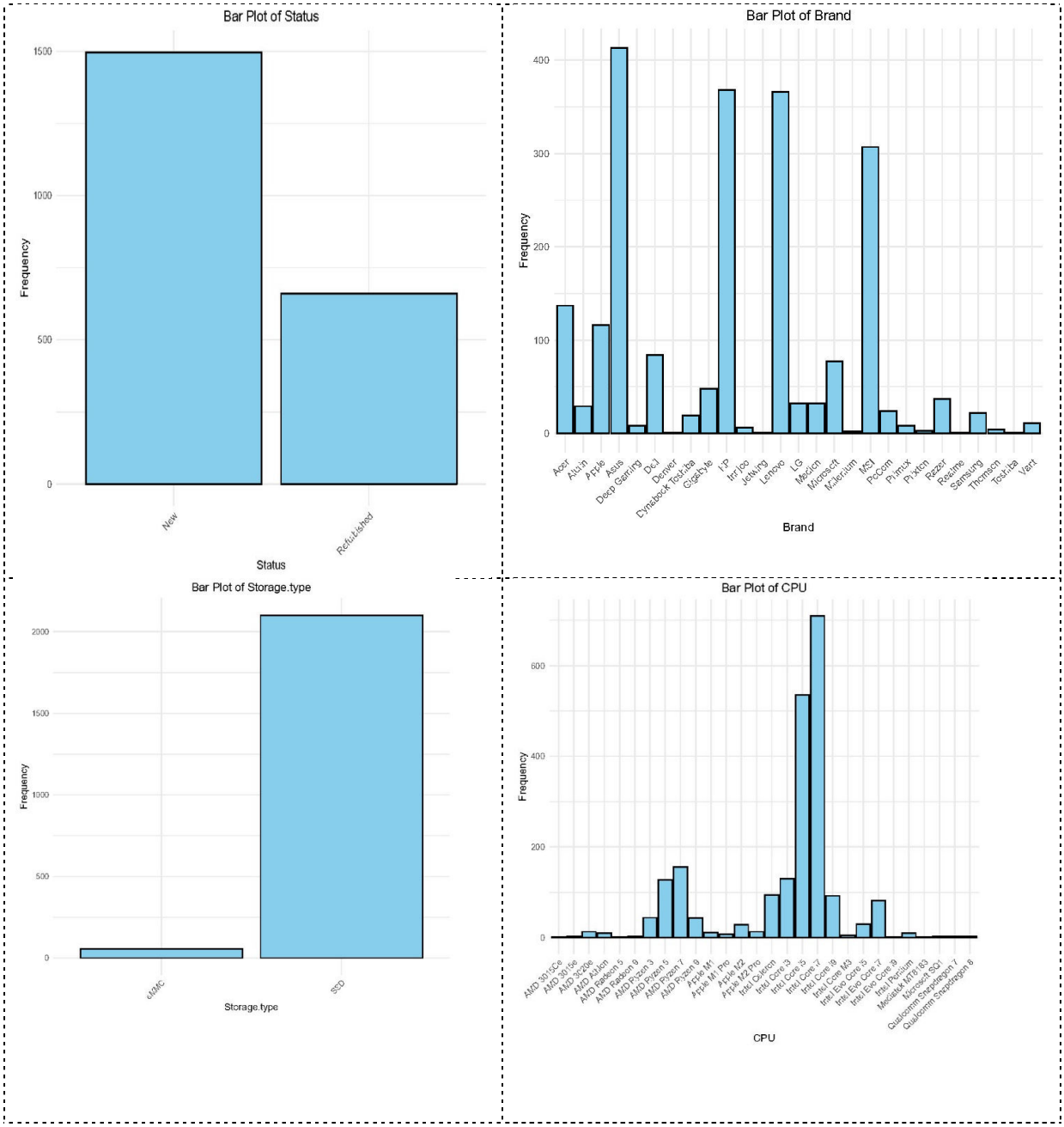
- **Variance Inflation Factor (VIF)** measures how much the variance of a regression coefficient is inflated due to multicollinearity among the predictors. Multicollinearity occurs when predictors are highly correlated with each other, which can make it difficult to estimate the relationship between each predictor and the outcome variable independently.

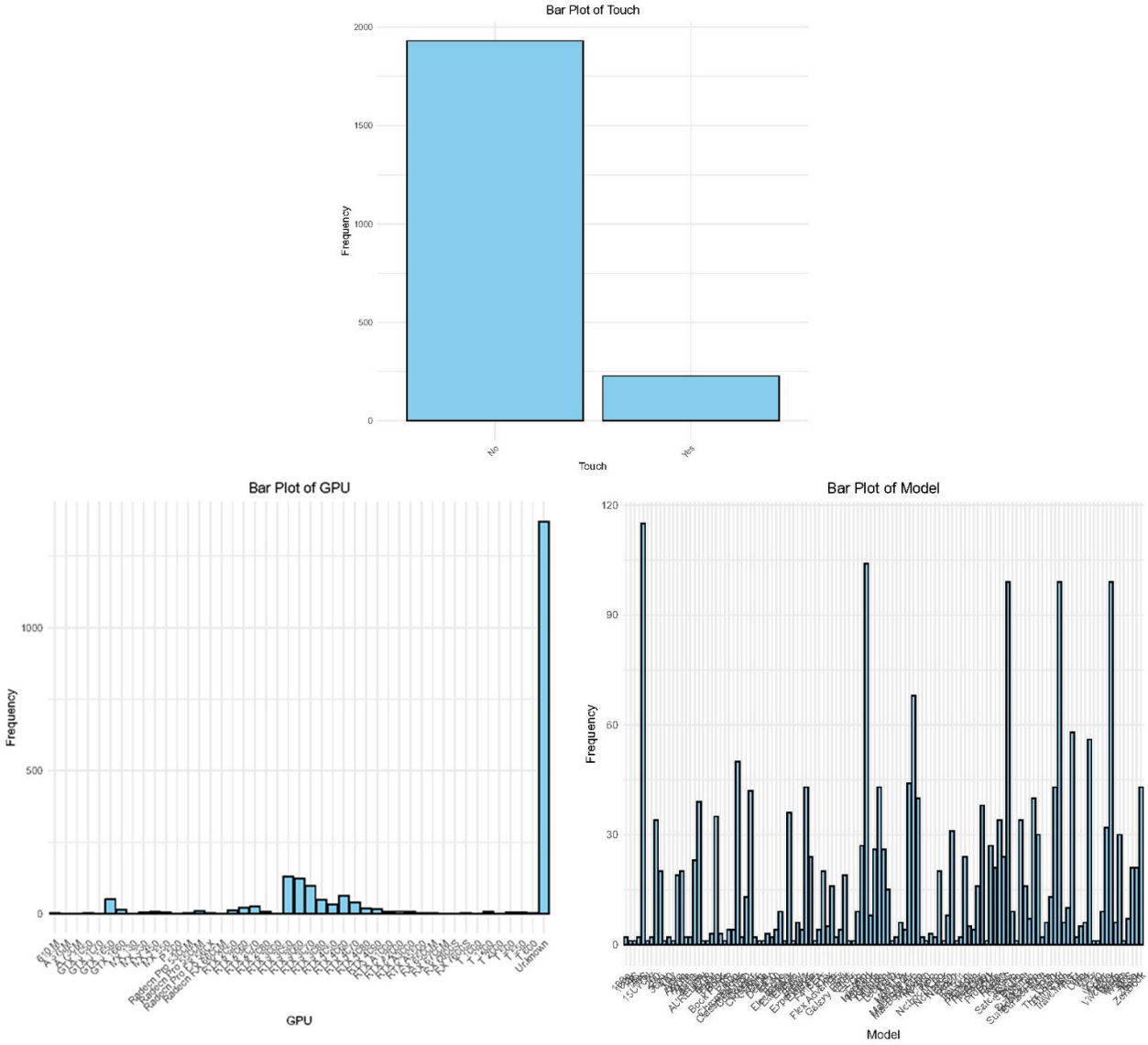
**Interpretation of VIF Values:**

- **RAM (2.59) and Storage (2.70):**
  - These VIF values are above 1 but below 5, indicating moderate multicollinearity between RAM and Storage. This is not usually a cause for concern, but it suggests that these two variables are somewhat correlated, which is consistent with the high correlation (0.78) between them observed in the correlation matrix.
- **Screen (1.25):**
  - The VIF for Screen is low, indicating that there is little multicollinearity between Screen and the other variables (RAM and Storage). This suggests that Screen size is relatively independent of RAM and Storage in predicting Final.Price.

# 7. Operations on Categorical data

## 7.1 Visualization






1. Bar Plots of Categorical Variables

**Status**:

The bar plot shows that most laptops in the dataset are New, with a smaller proportion being Refurbished. This indicates a dataset with a majority of new laptops, which is typical in many electronics datasets where new items dominate.

**Brand**:

The bar plot for Brand reveals that certain brands like Asus, Dell, and HP are more frequent in the dataset. The distribution suggests that the dataset is skewed towards these popular brands, which may dominate the market share.

**Model**:

The plot shows a wide variety of laptop models, with some models being much more common than others. The diversity in models indicates a dataset that covers a broad range of products, possibly spanning multiple years or product lines.

**CPU**:

The bar plot for CPU shows a concentration around a few popular processors, particularly those from Intel.This reflects the dominance of certain CPUs in the market, such as the Intel Core i5 and i7 series.

**Storage.type:**
The plot reveals that most laptops in the dataset use SSD (Solid State Drive) storage, with a few using eMMC.
This trend reflects the modern preference for SSDs, which are faster and more reliable than older storage technologies.
**GPU:**
The GPU plot shows a wide range of GPUs, with a significant number labeled as Unknown, indicating missing or unspecified data.
**Touch:**
The bar plot for Touch shows that the majority of laptops do not have a touchscreen feature.
This reflects the market trend where touchscreens are common in certain types of laptops (e.g., 2-in-1s) but not ubiquitous across all laptop categories.

## 7.2 Encoding of Categorical Variables

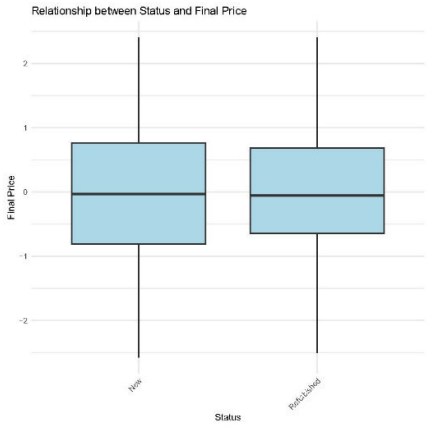
In this preprocessing step, we transformed categorical variables into numerical form to prepare the dataset for machine learning models. Below are the steps undertaken:

**1. Categorical Variable Encoding**

- **Touch**: Encoded as binary (1 for "Yes", 0 for "No").
- **Status**: Encoded as binary (0 for "New", 1 for "Refurbished").
- **Storage Type**: Encoded as binary (1 for "SSD", 0 for "eMMC").
- **Brand, Model, CPU, GPU**: These categorical variables were converted into numeric values based on their factor levels.

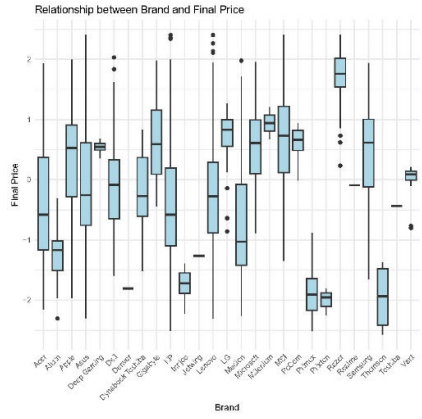
## 7.3 Relationship between categorical features and target variable:

### 7.3.1 Relationship between status and Final Price



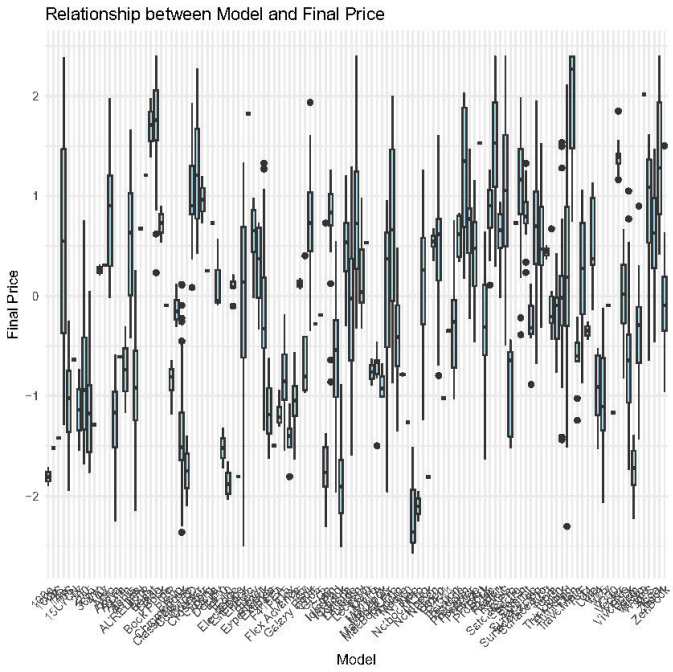

The box plot compares the Final.Price between New and Refurbished laptops.
New laptops have a slightly higher median price than Refurbished ones, though the difference is not substantial.
This indicates that while new laptops tend to be priced higher, refurbished laptops still cover a broad price range. The overlap in price ranges suggests that some refurbished laptops can still be quite expensive, possibly due to high-end specs or premium brands.

### 7.3.2 Relationship between Brand and Final Price:



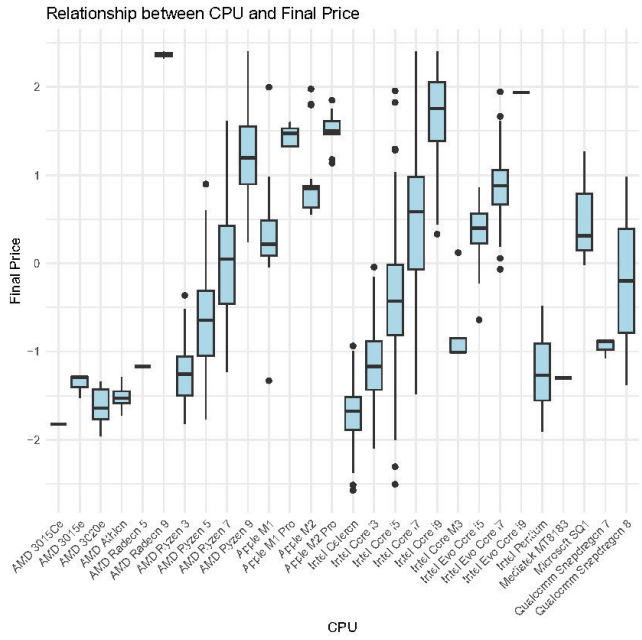

The box plot shows the distribution of Final.Price across different Brand categories.
Some brands, like Apple and Microsoft, have higher median prices, while others, like Acer and Dell, show a wider price range with a lower median.
The plot reflects brand positioning in the market. Premium brands like Apple and Microsoft are generally more expensive, while other brands offer a range of laptops, from budget to high-end, leading to a broader price distribution.

### 7.3.3 Relationship between Model and Final Price:




The box plot shows a significant variation in Final.Price across different Model categories. Some models are consistently priced higher, while others cover a broader range of prices.

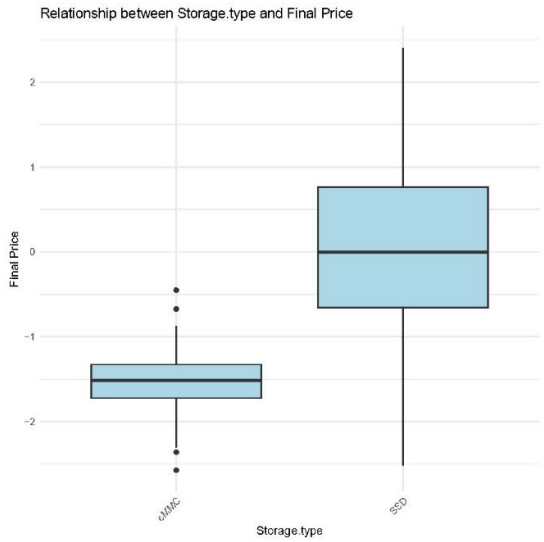
### 7.3.4 Relationship between CPU and Final Price:




The box plot shows the distribution of Final.Price for different CPU types.

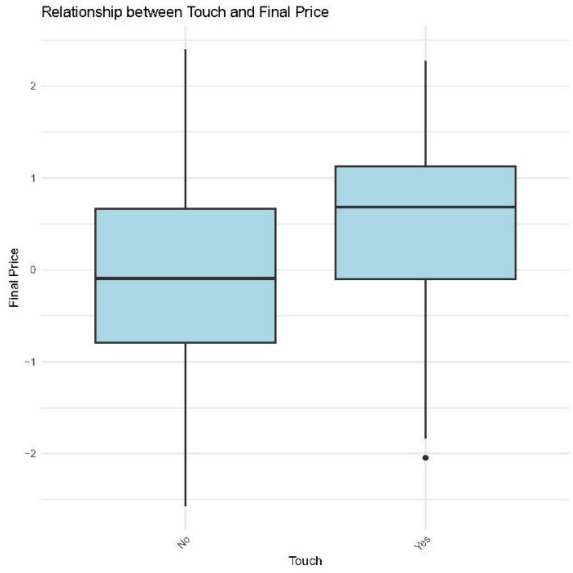
High-performance CPUs like Intel Core i7 and Intel Core i9 are associated with higher prices, while older or lower-end CPUs have a lower median price.

### 7.3.5 Relationship between Storage Type and Final Price:




The box plot compares Final.Price between laptops with SSD and eMMC storage types.
Laptops with SSD tend to have a higher final price compared to those with eMMC.
Interpretation:

### 7.3.6 Relationship between Touch and Final Price:




The box plot compares Final.Price between laptops with and without a touchscreen.
Laptops with touchscreens (Yes) tend to have a slightly higher median price compared to those without (No).

### 7.4 Feature Selection

The following features were selected for model training:

- **RAM**, **Storage**, **Screen**: Original numerical features.
- **Touch_encoded**, **Status_encoded**, **Storage.type_encoded**: Encoded categorical variables.
- **Brand_encoded**, **Model_encoded**, **CPU_encoded**, **GPU_encoded**: Encoded brand, model, CPU, and GPU.

The **target variable** is **Final Price**,

# 8. supervised machine learning methods

## 8.1 Linear Regression Model:

```
> summary(lm_model)

Call:
lm(formula = trainY ~ ., data = trainX)

Residuals:
     Min       1Q   Median       3Q      Max
-1.79810 -0.40124 -0.02696  0.35732  2.37502

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)          -0.5120402  0.1246415  -4.108 4.18e-05 ***
RAM                   0.4729539  0.0235268  20.103  < 2e-16 ***
Storage               0.2931493  0.0241743  12.127  < 2e-16 ***
Screen                0.0347129  0.0171873   2.020  0.04357 *
Touch_encoded         0.4555344  0.0487480   9.345  < 2e-16 ***
Status_encoded       -0.2194104  0.0315186  -6.961 4.78e-12 ***
Storage.type_encoded  0.1733891  0.0984540   1.761  0.07840 .
Brand_encoded         0.0037409  0.0024002   1.559  0.11927
Model_encoded         0.0017976  0.0004176   4.304 1.77e-05 ***
CPU_encoded           0.0236090  0.0036259   6.511 9.76e-11 ***
GPU_encoded          -0.0048795  0.0015213  -3.208  0.00136 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5727 on 1714 degrees of freedom
Multiple R-squared:  0.6667,    Adjusted R-squared:  0.6647
F-statistic: 342.8 on 10 and 1714 DF,  p-value: < 2.2e-16
```

### 8.1.1 Model Summary:

1. **Residuals**:
   - The residuals indicate the difference between observed and predicted values. The median is close to zero, which is a good sign, but the residuals range from -1.79810 to 2.37502, showing some variability in prediction accuracy.
2. **Coefficients**:
   - Each coefficient represents the effect of one unit change in the predictor (independent variable) on the target variable (final price), while holding other variables constant.
   - **Significant variables** (marked with ***, **, or *) have a low p-value (< 0.05), meaning they have a statistically significant impact on the target variable:
     - **RAM** (0.4729539): Strong positive effect on the target.
     - **Storage** (0.2931493): Positive effect.
     - **Touch_encoded** (0.4555344): Significant positive effect.
     - **Status_encoded** (-0.2194104): Significant negative effect.
     - **CPU_encoded** (0.0236090): Significant positive effect.
     - **GPU_encoded** (-0.0048795): Significant but small negative effect.
   - **Storage.type_encoded** and **Brand_encoded** are less significant based on their p-values (closer to 0.1).
3. **Model Fit**:
   - **Residual standard error**: 0.5727, which measures the average deviation of the predicted values from the actual values.

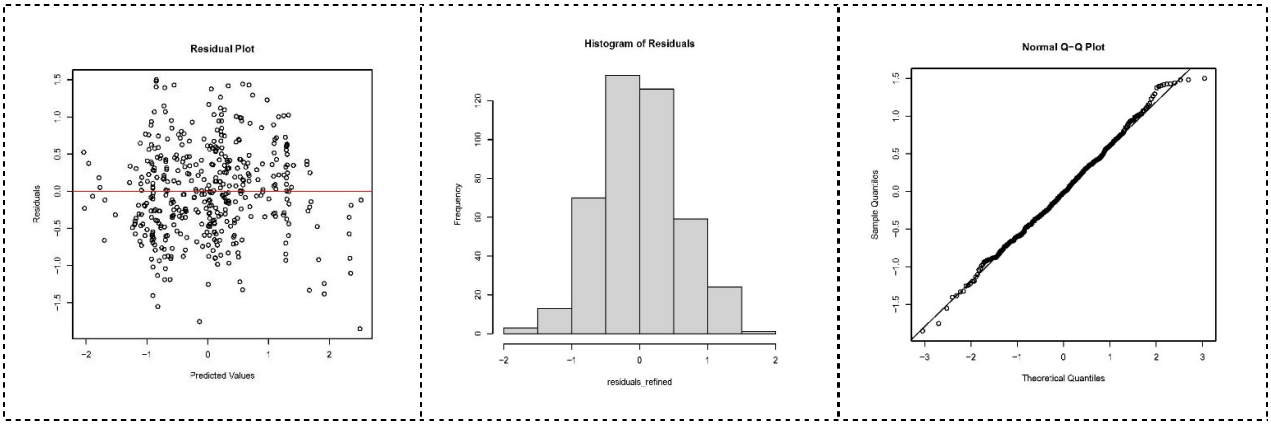
- **Multiple R-squared**: 0.6667, which indicates that about 66.67% of the variance in the target variable is explained by the model,however this amount is 69%for test data.
- **Adjusted R-squared**: 0.6647, adjusts the R-squared for the number of predictors in the model.
- **F-statistic**: 342.8, with a very small p-value (< 2.2e-16), suggesting that the model as a whole is statistically significant.

## Brief Conclusion:

- The model explains about 67% of the variance in the target variable (Final.Price).
- **RAM**, **Storage**, **Touchscreen feature**, and **CPU** have a strong positive impact on the target, while **Status** and **GPU** have significant negative effects.
- Overall, the model is statistically significant with a good fit.

### 8.1.2 Plots Interpretation:

Actual vs Predicted Plot: This plot shows how well the predicted prices match the actual prices. The points are scattered around the diagonal line, indicating a good fit. Some deviations are expected, but overall the model seems to perform well.




Histogram of Residuals: The residuals appear to be roughly normally distributed, which is a good sign that the model's assumptions are valid.

Normal Q-Q Plot: The Q-Q plot shows that the residuals follow a straight line, which indicates that they are normally distributed—a key assumption of linear regression.

### 8.1.3 Interpretation of the Results:

**A) Shapiro-Wilk normality test**

- **W = 0.98852**: This is the test statistic for the Shapiro-Wilk test. The value of W close to 1 generally indicates a distribution that is closer to normal. However, in this case, W is slightly less than 1, suggesting some deviation from normality.
- **p-value = 0.001791**: This is the key value for interpretation. A **p-value < 0.05** indicates that the residuals **do not follow a normal distribution**.

### B) Durbin-Watson Test:

**DW = 2.0327, p-value = 0.751**

**Interpretation:**

- The value of **2** indicates **no autocorrelation** in the residuals.
- Values closer to **0** indicate positive autocorrelation, while values closer to **4** indicate negative autocorrelation.
- Since **2.0327** is very close to 2, this suggests that there is **no significant autocorrelation** in the residuals.

**p-value = 0.751**:

- The **p-value** of **0.751** is much greater than 0.05, meaning we **fail to reject the null hypothesis**. The null hypothesis in the Durbin-Watson test is that **no autocorrelation** is present in the residuals.

**Summary:**

- **Shapiro-Wilk Normality Test**: the residuals do not follow a normal distribution.
- **Durbin-Watson Test:** there is no evidence of autocorrelation in the residuals of the regression model.

Therefore, **more robust methods** should be employed, as the assumptions of linear regression are not fully met .

## 8.2 Support Vector Machine

**Support Vector Machine (SVM)** is a supervised machine learning algorithm used for both **classification** and **regression** tasks, The core idea of SVM is to find a **hyperplane** (or decision boundary) that best separates data points from different classes in the feature space.

In this model, we used the **RBF kernel** to handle non-linear data and **epsilon regression** for continuous output.

- **R-squared: 0.80**
- **Explanation**: The SVM model with a radial kernel provides an R-squared value of 0.80. This means that the model explains about 80% of the variance in the Final. Price. SVM is particularly effective for non-linear relationships, and the relatively high R-squared value indicates that the model fits the data well, capturing a substantial portion of the variability in Final. Price.

### 8.3 Random Forest Model

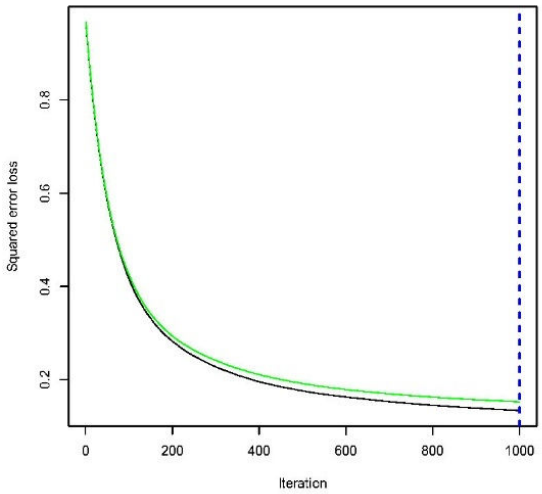
Random forest regression is a supervised learning algorithm and a type of bagging technique that employs ensemble learning for regression tasks in machine learning. In this approach, multiple decision trees are built independently and in parallel, with no interaction between the trees during their construction. During training, a large number of decision trees are created, and the final output is determined by either the mode of the predictions for classification tasks or the average of the predictions for regression tasks. This approach helps to overcome the limitations of individual decision trees, such as overfitting and high variance.

- **R-squared: 0.88**
- **Explanation**: This method, achieves the highest R-squared value of 0.88. This implies that 88% of the variance in Final.Price is explained by the model. The high R-squared value indicates that the Random Forest model is highly effective at capturing the underlying patterns in the data. It combines the predictions of multiple decision trees to improve accuracy and robustness, leading to a strong predictive performance.

### 8.4 Gradient Boosting Model

**Gradient Boosting** is a powerful machine-learning algorithm used for both **classification** and **regression** tasks. It builds models sequentially, where each new model corrects the errors of the previous one. Unlike Random Forest, where trees are built independently, Gradient Boosting builds one tree at a time, focusing on reducing the errors from the previous models.

- **R-squared: 0.87**
- **Explanation**: We used cross-validation to find the optimal number of iterations (best_iter)which is 1000 based on the picture. Then we used this optimal number for the number of trees when making predictions. This prevents using too many trees and overfitting the model.
- The Gradient Boosting model yields an R-squared value of 0.87, indicating that 87% of the variance in Final. Price is explained by the model.



### 8.5 Conclusion:

The models' R-squared values indicate the following ranking in terms of performance:

1. **Random Forest (0.88)**: Best performance, capturing the majority of the variability in Final.Price.
2. **Gradient Boosting (0.87)**: Close second, with strong predictive capabilities.
3. **Support Vector Machine (0.80)**: Good performance, especially in capturing non-linear relationships.
4. **Linear Regression (0.69)**: Moderate performance, capturing linear relationships.

# 9. unsupervised machine learning methods(clustering)

## 9.1 Steps before clustering:

### 9.1.1 Data Scaling

Before applying clustering algorithms, the data was scaled using the scale() function. This ensures that all features are on the same scale, preventing variables with larger ranges from dominating the clustering process.

### 9.1.2 finding optimal k

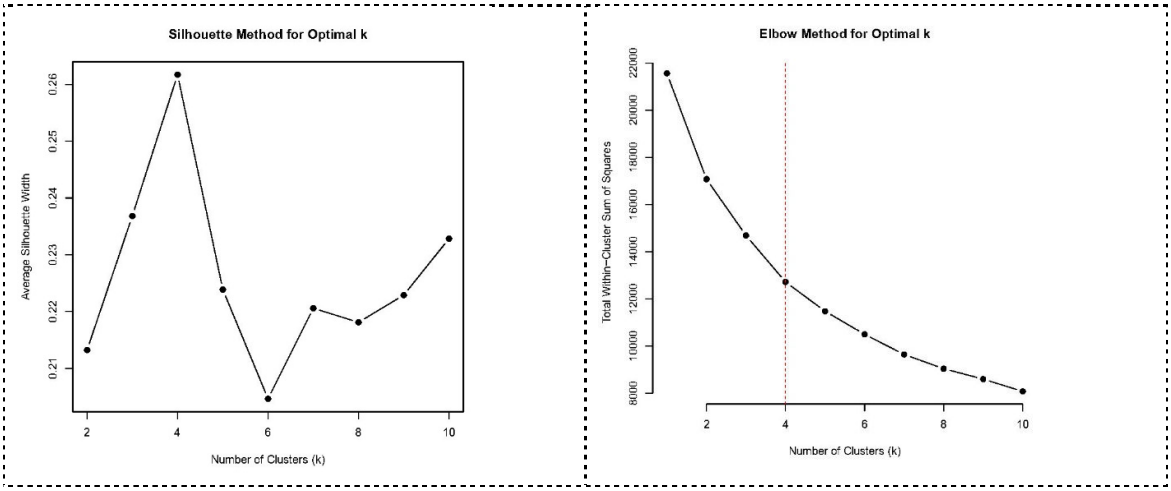
**A. Elbow Method Interpretation**

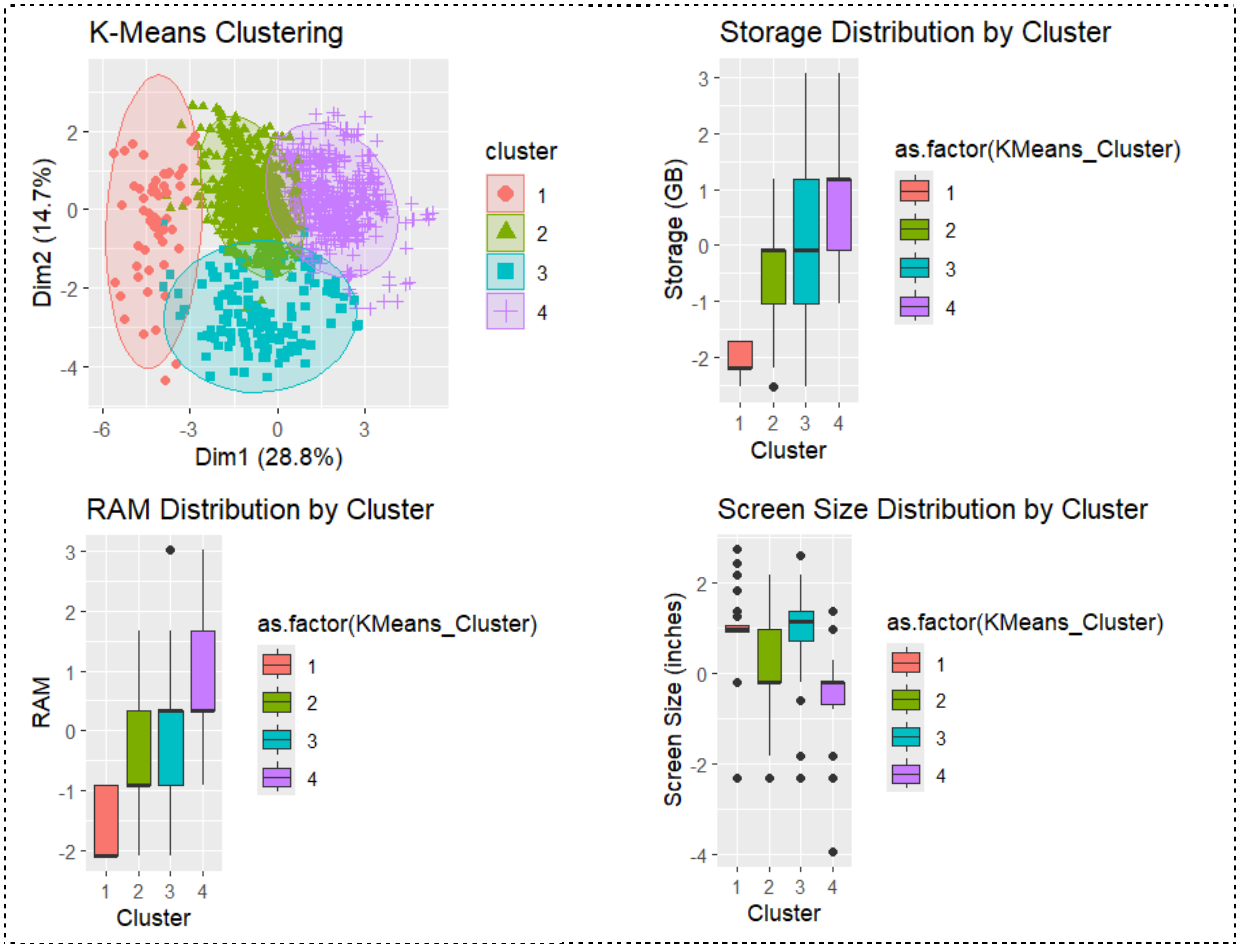
The Elbow Method helps determine the optimal number of clusters by plotting the total within-cluster sum of squares (WSS) for different values of k (number of clusters). The plot shows a clear "elbow" at **k = 4**, where the rate of decrease in WSS slows down significantly. This suggests that **4 clusters** is likely the optimal number of clusters for this dataset,

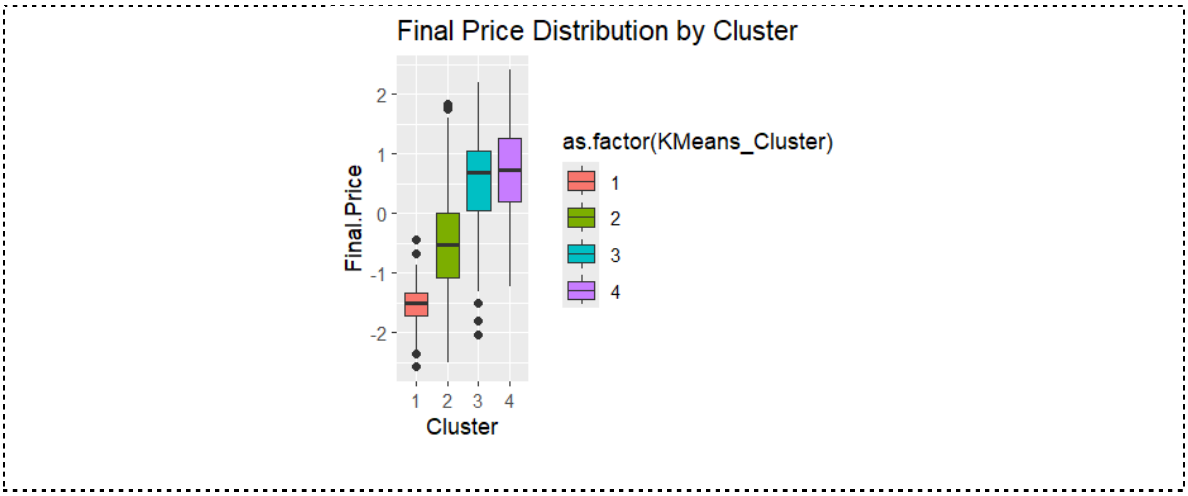
**B. Silhouette Method Interpretation**

The Silhouette Method measures how well-separated the clusters are by calculating the average silhouette width for each number of clusters. In this case, the plot shows that the highest silhouette width is at **k = 4**, which confirms that 4 clusters provide the best separation between data points. The silhouette width drops for k values greater than 4, indicating that the clustering quality decreases with more clusters.

Both the **Elbow Method** and **Silhouette Method** point to **k = 4** as the optimal number of clusters for this dataset. This means that the data can be meaningfully grouped into 4 distinct clusters, which provide both well-defined separation and optimal within-cluster variance.




K-Means Clustering
Dim2 (14.7%)
Dim1 (28.8%)
cluster
1
2
3
4
Storage Distribution by Cluster
Storage (GB)
Cluster
as.factor(KMeans_Cluster)
RAM Distribution by Cluster
RAM
Cluster
as.factor(KMeans_Cluster)
Screen Size Distribution by Cluster
Screen Size (inches)
Cluster
as.factor(KMeans_Cluster)


Final Price Distribution by Cluster
Final.Price
Cluster
as.factor(KMeans_Cluster)

# Interpretation of Clustering Results and Visualization

## 9.2 K-Means Clustering Analysis

### 9.2.1 K-Means Clustering Visualization:

- The first plot shows the K-Means clustering with 4 clusters, visualized using the first two principal components. Each color and shape represent a different cluster, and the ellipses indicate the spread of the data points within each cluster. The clear separation between clusters suggests that the K-Means algorithm was able to identify distinct groups within the data based on the selected features.

### 9.2.2 Cluster Descriptions:

- **Cluster 1**:
  - This cluster is characterized by lower RAM, Storage, and Final Price. It seems to consist of lower-end laptops with less powerful configurations.
- **Cluster 2**:
  - This cluster has moderate values for most features. The laptops in this cluster likely have mid-range configurations, making them suitable for average users.
- **Cluster 3**:
  - This cluster includes laptops with the highest average screen size and relatively high storage, indicating larger, possibly gaming or workstation laptops.
- **Cluster 4**:
  - This cluster has the highest average RAM and Final Price, suggesting that it contains high-end laptops, possibly used for gaming, content creation, or other resource-intensive tasks.

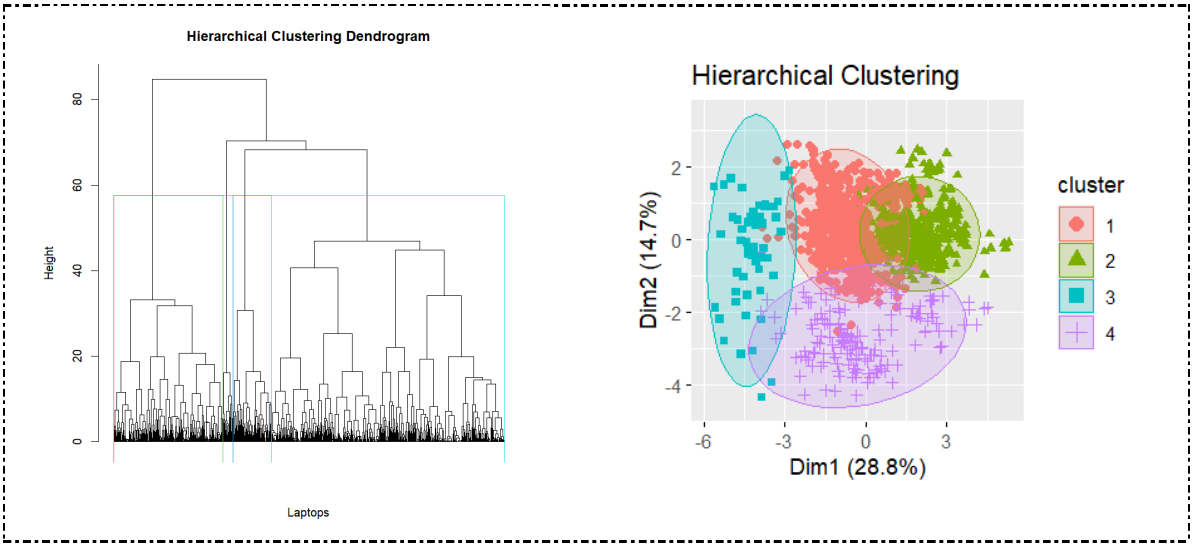
**Boxplots by Cluster**:

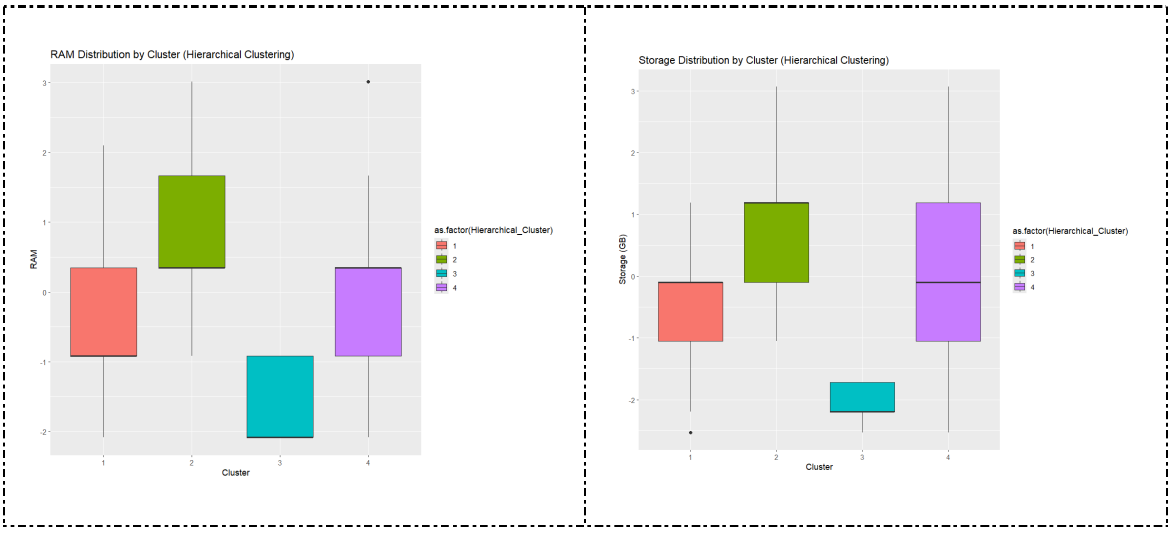
- **RAM Distribution**:
  - The RAM distribution across clusters shows that Cluster 4 has significantly higher RAM compared to the other clusters, consistent with high-end laptops. Clusters 1 and 2 have lower RAM, with Cluster 1 having the least.
- **Final Price Distribution**:
  - Final Price distribution reveals that Cluster 4 contains the most expensive laptops, followed by Cluster 3. Cluster 1 includes the least expensive laptops, which aligns with their lower specifications.
- **Screen Size Distribution**:
  - Cluster 3 stands out with the largest screen sizes, while Clusters 1 and 2 have smaller screens.
- **Storage Distribution**:
  - The storage distribution across clusters shows that Cluster 3 has significantly higher storage values compared to the other clusters, consistent with users who require more storage capacity. Cluster 4 also exhibits relatively high storage, though with slightly less variation than Cluster 3. In contrast, Clusters 1 and 2 have lower storage capacities, with Cluster 1 having the smallest storage values.

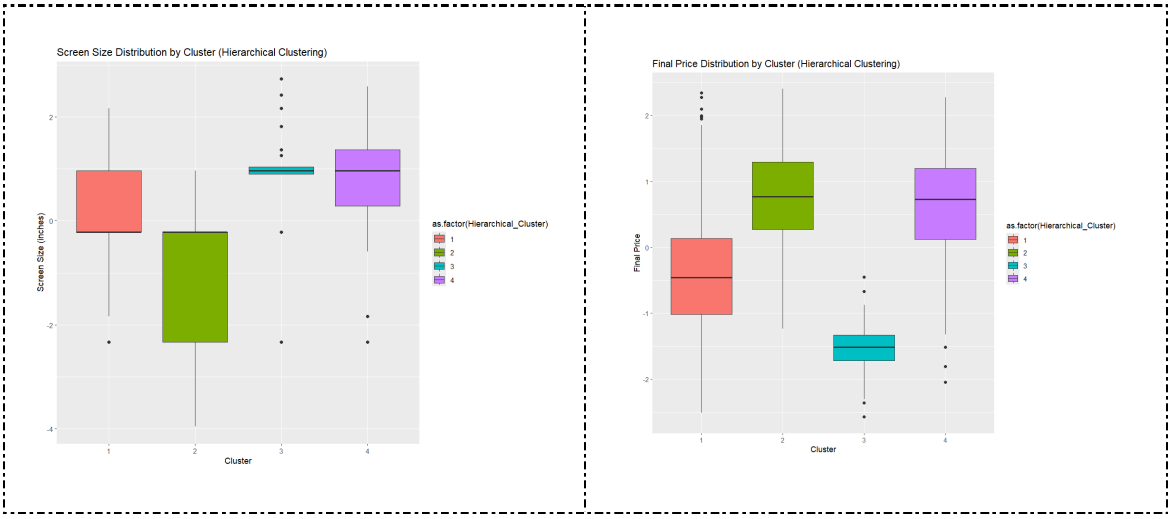
## 9.3 Hierarchical Clustering Analysis

### 9.3.1 Hierarchical Clustering Dendrogram:

- The dendrogram shows the hierarchical clustering process, where each merge is represented by a horizontal line. The height of the merge indicates the dissimilarity between the clusters being merged. The lower the height, the more similar the clusters. The dendrogram suggests a natural division into 4 clusters, consistent with the K-Means analysis.






### 9.3.2 Hierarchical Cluster Descriptions:

**Cluster 1:**

This cluster is characterized by lower RAM, moderate screen sizes, and lower final prices. It appears to represent lower-end laptops with basic configurations. Storage is relatively low, indicating less storage capacity.

**Cluster 2:**

Cluster 2 is marked by high RAM and storage, along with moderate screen sizes. The final prices for this cluster are moderate to high, suggesting that this group consists of mid-range to high-end laptops that balance both storage capacity and performance.

**Cluster 3:**

This cluster is defined by the lowest RAM, storage, and screen sizes among all clusters. Final prices are also relatively low. It likely represents budget or entry-level laptops, with minimal specifications aimed at cost-conscious users.

**Cluster 4:**

Cluster 4 stands out with high RAM, storage, and larger screen sizes. The final prices in this group are also the highest, making it indicative of premium laptops. This cluster includes devices with robust performance and significant storage, suitable for users requiring higher-end computing capabilities.

## Boxplots by Cluster:

**RAM Distribution:**

The RAM distribution across clusters shows that Cluster 2 has the highest RAM, indicating devices with more powerful memory capacities. Cluster 4 also has higher RAM, but not as much as Cluster 2. Clusters 1 and 3 have lower RAM, with Cluster 3 having the least, possibly representing lower-end devices.

**Storage Distribution:**

The storage distribution shows that Cluster 4 has the highest storage capacity, with a wide range of storage values, indicative of more robust storage devices. Cluster 2 follows with moderately high storage values. Cluster 3 shows the lowest storage distribution, while Cluster 1 also has lower storage compared to Clusters 2 and 4.

**Screen Size Distribution:**

The screen size distribution across clusters indicates that Cluster 4 has the largest screens, with a relatively narrow range, suggesting larger, more uniform screen sizes. Cluster 1 also has relatively large screen sizes, while Cluster 2 has a moderate spread. Cluster 3 has the smallest screen sizes.

**Final Price Distribution:**

The final price distribution indicates that Cluster 4 has the highest prices, aligning with its higher RAM, storage, and screen sizes. Cluster 2 also shows higher prices, though slightly lower than Cluster 4. Clusters 1 and 3 have lower price ranges, with Cluster 1 having the lowest prices, consistent with lower-end devices in terms of RAM and storage.

### 9.3.3 Number of Laptops in Each Hierarchical Cluster:

- **Hierarchical Cluster 1**:
  - 1145 laptops from K-Means Cluster 2
  - 142 laptops from K-Means Cluster 4
  - **Total**: 1287 laptops
- **Hierarchical Cluster 2**:
  - 9 laptops from K-Means Cluster 2
  - 592 laptops from K-Means Cluster 4
  - **Total**: 601 laptops
- **Hierarchical Cluster 3**:
  - 56 laptops from K-Means Cluster 1
  - 196 laptops from K-Means Cluster 3
  - **Total**: 252 laptops
- **Hierarchical Cluster 4**:
  - 17 laptops from K-Means Cluster 4
  - **Total**: 17 laptops.

| | K-Means Cluster 2 | K-Means Cluster 4 | K-Means Cluster 1 | K-Means Cluster 3 |
|---|---|---|---|---|
| Hierarchical Cluster 1 | 1145 | 142 | 0 | 0 |
| Hierarchical Cluster 2 | 9 | 592 | 0 | 0 |
| Hierarchical Cluster 3 | 0 | 0 | 56 | 196 |
| Hierarchical Cluster 4 | 0 | 17 | 0 | 0 |

## 9.4 Comparison with K-Means Clustering:

- The contingency table compares the clusters formed by K-Means and Hierarchical Clustering. The results show that there is a significant overlap between the clusters identified by both methods. However, there are some differences:
  - Cluster 2 in K-Means has some overlap with multiple clusters in hierarchical clustering.
  - Cluster 4 in K-Means corresponds mainly to a single cluster in hierarchical clustering, indicating a strong agreement between the methods for high-end laptops.

# 10. Conclusion

The clustering analysis revealed four distinct groups of laptops based on key features like RAM, storage, screen size, CPU type, and price. The K-Means and Hierarchical Clustering methods largely agreed on the cluster assignments, with some variations.

**Cluster 1** is similar in both algorithms, representing low-end devices, though hierarchical clustering shows a bit more variation in storage and RAM.

**Cluster 2** is where the biggest difference appears. In hierarchical clustering, it has the highest RAM, making it more like mid-to-high-end devices, while in K-Means, it's more mid-range.

**Cluster 3** remains the lowest in RAM, storage, and price in both methods, but hierarchical clustering shows an even bigger drop in specifications.

**Cluster 4** consistently represents the premium group with high RAM, storage, screen size, and price in both algorithms, though hierarchical clustering shows slightly more variation within the group.