VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF APPLIED SCIENCE

PROBABILITY & STATISTICS (MT2013)

CC06 - Group 4

Predicting Intel's CPU Clock Speed Using Statistical Methods

Advisor: Phan Thị Khánh Vân, FAS - HCMUT

Students: Lê Nguyễn Gia Bảo - 2210216.
Trần Đình Đăng Khoa - 2211649.
Bùi Vũ Thiên Đăng - 2252151.
Trần Tuấn Minh Khoa - 2252365.
Nguyễn Hữu Trí - 2252842.

HO CHI MINH CITY, APRIL 2024

Contents

1 Member list & Workload **2**

2 Abstract **3**

3 Introduction **4**

4 Background Knowledge on Statistical Methods **6**
4.1 Hypothesis Testing . . . 6
4.2 Analysis of Variance (ANOVA) . . . 7
4.2.1 Formulas (for reference) . . . 8
4.3 Linear Regression . . . 9
4.3.1 Formulas (for reference) . . . 10
4.4 Imputation . . . 11
4.5 Remark . . . 12

5 Data Pre-processing **13**
5.1 Missing values examination and attributes selection . . . 15
5.2 Data Relevance and Usefulness . . . 15
5.3 Handling Missing Values . . . 16
5.4 Handling outliers . . . 18

6 Descriptive Statistic **19**
6.1 Distribution and Histograms . . . 19
6.2 Boxplots for Outliers . . . 21
6.3 Scatter plot for relationship between Processor Base Frequency with the others . . . 23
6.4 Correlation Matrix . . . 25

7 Inferential Statistics **27**
7.1 Two-way analysis of variance(ANOVA) . . . 27
7.1.1 Verify the assumption . . . 27
7.1.2 Calculate ANOVA . . . 29
7.2 Multiple Linear Regression to Predict CPU Clock Speed . . . 29
7.2.1 Data Splitting . . . 29
7.2.2 Regression Model . . . 30
7.2.3 Assumptions of Linear Regression . . . 31
7.2.4 Testing . . . 32
7.3 Conclusion . . . 34

8 Discussion **35**

9 References **37**

1 Member list & Workload

No.	Fullname	Student ID	Contribution
1	Lê Nguyễn Gia Bảo	2210216	20%
2	Trần Đình Đăng Khoa	2211649	20%
3	Bùi Vũ Thiên Đăng	2252151	20%
4	Trần Tuấn Minh Khoa	2252365	20%
5	Nguyễn Hữu Trí	2252842	20%

2 Abstract

In the rapidly advancing field of computer hardware technology, understanding and predicting the clock speed (frequency) of central processing units (CPUs) is crucial for both manufacturers and consumers. This project, **"Predicting Intel CPU Clock Speed Using Statistical Methods"**, aims to develop robust predictive models for CPU clock speed based on detailed specifications of Intel CPUs. By employing statistical techniques such as **Linear Regression** and **Analysis of Variance (ANOVA)**, this study seeks to identify the key features that significantly influence CPU clock speed.

The dataset utilized in this study comprises a comprehensive collection of Intel CPU specifications, including attributes such as the number of cores, number of threads, cache size, power consumption, and various architectural details. Data preprocessing steps involve handling missing values, normalizing data, and encoding categorical variables to ensure the dataset is suitable for rigorous statistical analysis.

To identify significant predictors of CPU clock speed, ANOVA is used to assess the impact of categorical variables, providing insights into how different CPU series and generations affect clock speeds. Linear regression is then employed to model and predict CPU clock speed based on these significant features. This method directly establishes the relationship between the dependent variable (clock speed) and the independent variables (CPU specifications).

The predictive modeling component of this project primarily relies on linear regression techniques. Linear regression provides a foundational understanding of the linear relationships between the predictors and CPU clock speed. The performance of the regression models is evaluated using key metrics such as R-squared, Mean Squared Error (MSE), and visual inspection of residual plots.

Our analysis reveals that features such as the number of cores, number of threads, cache size, and power consumption are significant determinants of CPU clock speed. The linear regression model offers valuable insights into the impact of these features on clock speed, allowing for accurate predictions based on the given specifications. While ANOVA provides supplementary information on the influence of categorical variables, it is the linear regression model that forms the core of our predictive analysis.

This project contributes to the broader understanding of CPU performance dynamics, providing a methodological framework that can be applied to other hardware components or similar predictive tasks. The findings have practical implications for manufacturers in optimizing CPU design and for consumers in making informed purchasing decisions. By leveraging statistical methods and regression analysis, this study offers a data-driven approach to predicting CPU clock speed, enhancing transparency and efficiency in the marketplace.

3 Introduction

The Central Processing Unit (CPU) is often referred to as the "brain" of the computer due to its fundamental role in executing instructions and managing the operations of other components. It processes data, performs calculations, and manages tasks, making it a critical component that directly impacts a computer's performance and efficiency. As technology continues to advance rapidly, the variety and complexity of CPUs available in the market have also increased, necessitating more sophisticated methods to evaluate and predict their clock speed (frequency).

Predicting CPU clock speed accurately is crucial for several reasons. For manufacturers, understanding the factors that influence CPU clock speed can aid in optimizing CPU design, enhancing product development, and targeting the right market segments. Accurate clock speed predictions can help manufacturers maintain a balance between performance and cost-effectiveness. For consumers, knowledge of CPU performance dynamics enables informed purchasing decisions, ensuring that they obtain the best value for their money. This is particularly important given the diverse range of CPUs available, each with different specifications and performance levels.

This report focuses on the analysis of CPU specifications to predict clock speed using a variety of statistical methods. The dataset used in this study consists of detailed specifications of Intel CPUs, one of the leading CPU manufacturers in the world. Intel CPUs are widely used in various computing devices, from desktops and laptops to servers and workstations, making them an ideal subject for this study.

The primary goal of this report is to identify the key features that significantly influence CPU clock speed and to develop predictive models that can accurately estimate clock speeds based on the identified features. To achieve this, we employ several statistical techniques, including Analysis of Variance (ANOVA) and regression analysis. Each of these methods plays a vital role in understanding the relationships between different CPU specifications and their corresponding clock speeds.

Analysis of Variance (ANOVA) is utilized to determine the impact of categorical variables on CPU clock speed. By comparing means across different groups, ANOVA helps identify whether certain categorical features, such as CPU series or generation, significantly affect clock speed. This analysis provides supplementary insights that enhance our understanding of how different CPU models and architectures impact performance.

The regression analysis forms the core of our predictive modeling approach. Linear regression is applied to provide a foundational understanding of the linear relationships between the selected features and CPU clock speed. Given the complexity of CPU performance, we also consider polynomial regression to capture potential non-linear interactions among the variables. By comparing the performance metrics of these models, such as R-squared and Mean Squared Error (MSE), we can determine the most effective model for predicting CPU clock speed.

Data preprocessing is a crucial step in this analysis, ensuring the reliability and accuracy of

our models. This involves handling missing values, normalizing data, and encoding categorical variables. Proper preprocessing enhances the quality of the dataset, making it suitable for rigorous statistical analysis and modeling.

In summary, this report aims to provide a comprehensive analysis of Intel CPU specifications to predict their clock speeds. By leveraging statistical methods and regression models, we seek to develop robust predictive models that can assist manufacturers in optimizing CPU design and help consumers make informed purchasing decisions. The findings of this study will contribute to the broader understanding of CPU performance dynamics and demonstrate the value of combining various statistical techniques to create accurate and reliable clock speed predictions.

4 Background Knowledge on Statistical Methods

4.1 Hypothesis Testing

Hypothesis testing is a fundamental statistical procedure used to make inferences about population parameters based on sample data. It is essential in various fields, including scientific research, quality control, and decision-making processes. The primary objective of hypothesis testing is to evaluate the plausibility of a specific claim or hypothesis concerning a population parameter, such as the mean or variance, which is crucial for understanding CPU clock speed determinants.

In hypothesis testing, two mutually exclusive hypotheses are formulated: the null hypothesis (H_0) and the alternative hypothesis (H_a). The null hypothesis typically represents the status quo, the baseline assumption, or the claim that the researcher wishes to test against. The alternative hypothesis represents the opposite or the alternative claim that the researcher aims to support or conclude if the null hypothesis is rejected.

The process of hypothesis testing involves the following steps:

1. Formulate the null hypothesis (H_0) and the alternative hypothesis (H_a).
2. Specify the significance level (α), which is the probability of rejecting the null hypothesis when it is true (Type I error).
3. Calculate the test statistic from the sample data.
4. Determine the critical region or the critical value(s) based on the significance level and the chosen test.
5. Compare the test statistic with the critical region or critical value(s).
6. Make a decision: Reject or fail to reject the null hypothesis.

The decision to reject or fail to reject the null hypothesis is based on the comparison between the test statistic and the critical region or critical value(s). If the test statistic falls within the critical region, the null hypothesis is rejected in favor of the alternative hypothesis. If the test statistic does not fall within the critical region, the null hypothesis is not rejected.

Types of hypothesis tests relevant to this project include:

- Tests for Means: Comparing the mean clock speeds of different CPU series or generations.
- Tests for Correlation and Regression Coefficients: Assessing the relationship between CPU specifications (e.g., number of cores) and clock speeds.

Hypothesis testing is subject to two types of errors: Type I error (rejecting the null hypothesis when it is true) and Type II error (failing to reject the null hypothesis when it is false). The significance level (α) controls the probability of committing a Type I error, while the power of the test ($1 - \beta$) represents the probability of correctly rejecting the null hypothesis when it is false, where β is the probability of committing a Type II error.

4.2 Analysis of Variance (ANOVA)

Analysis of Variance (ANOVA) is a statistical method used to determine if there are statistically significant differences between the means of three or more independent groups. For this project, ANOVA helps us understand the impact of categorical variables, such as CPU series and generation, on CPU clock speeds by comparing the average clock speeds across different groups.

Key Concepts:

- Hypotheses:
  - Null Hypothesis (H_0): Assumes that there are no differences in the mean CPU clock speeds among the different groups.
  - Alternative Hypothesis (H_a): Assumes that at least one group mean is different from the others.
- Between-Group Variability: Measures the variation in CPU clock speeds between different groups, reflecting the effect of the categorical variable on clock speeds.
- Within-Group Variability: Measures the variation in CPU clock speeds within each group, reflecting natural clock speed variations among CPUs of the same group.
- F-Statistic: The ratio of between-group variability to within-group variability. A higher F-statistic indicates a greater likelihood that the observed differences between group means are statistically significant.
- P-Value: The probability of observing the data assuming the null hypothesis is true. A low p-value (typically < 0.05) suggests that the differences between group means are statistically significant.

Procedure for Conducting ANOVA:

- Categorize Data: Group the dataset based on categorical variables such as CPU Series (e.g., Intel Core i3, i5, i7) and CPU Generation (e.g., 9th Gen, 10th Gen).
- Calculate Group Means: Determine the mean CPU clock speed for each group.
- Compute Sum of Squares:
  - Total Sum of Squares (SST): Measures the total variation in CPU clock speeds.
  - Between-Group Sum of Squares (SSB): Measures the variation in CPU clock speeds between different groups.
  - Within-Group Sum of Squares (SSW): Measures the variation in CPU clock speeds within each group.
- Calculate Mean Squares:
  - Mean Square Between (MSB): SSB divided by the degrees of freedom between groups.

  - Mean Square Within (MSW): SSW divided by the degrees of freedom within groups.

- Compute F-Statistic.
- Determine P-Value: Using statistical software or F-distribution tables, find the p-value corresponding to the calculated F-statistic.
- Post-Hoc Analysis (if necessary): If ANOVA indicates significant differences, conduct post-hoc tests to identify which specific groups differ from each other.

Application in This Project:

- Group Data by CPU Series: Calculate the mean clock speed for each series.
- Perform ANOVA Test for CPU Series: Formulate hypotheses, conduct ANOVA, calculate F-statistic and p-value, and interpret results.
- Group Data by CPU Generation: Calculate the mean clock speed for each generation.
- Perform ANOVA Test for CPU Generation: Formulate hypotheses, conduct ANOVA, calculate F-statistic and p-value, and interpret results.

By using ANOVA, we can systematically evaluate the impact of these categorical variables on CPU clock speeds, providing valuable insights into performance trends and refining our predictive models.

4.2.1 Formulas (for reference)

- The total sum of squares: $\mathbf{SST} = \sum_{i=1}^{k}\sum_{j=1}^{n}(X_{ij} - \bar{X})^2 = \sum_{i,j} X_{ij}^2 - \frac{X^2}{N}$.
- The treatment sum of squares: $\mathbf{SSTr} = \sum_{i=1}^{k} n(\bar{X}_i - \bar{X})^2 = \sum_{i=1}^{k} \frac{X_i^2}{n} - \frac{X^2}{N}$.
- The error sum of squares: **SSE = SST - SSTr**.
- Treatment degree of freedom: $df(\mathbf{SSTr}) = k - 1$. Error degree of freedom $df(\mathbf{SSE}) = N - k = nk - k$.
- The mean square for treatment: $\mathbf{MSTr} = \dfrac{\mathbf{SSTr}}{k-1}$.
- The mean square for error: $\mathbf{MSE} = \dfrac{\mathbf{SSE}}{nk-k}$.
- If H_0 is true, then the statistic $F = \dfrac{\mathbf{MSTr}}{\mathbf{MSE}} \sim F_{k-1,nk-k}$: Fisher random variable. If $F > F_{\alpha,k-1,nk-k}$ we reject H_0.

4.3 Linear Regression

Linear regression is a fundamental statistical technique used to model the relationship between a dependent variable (CPU clock speed) and one or more independent variables (CPU specifications). It is widely employed to analyze and make predictions based on observed data.

Objective: To find the best-fitting straight line that describes the relationship between CPU clock speed and its specifications. This line is represented by a linear equation, which takes the following form:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \varepsilon$$

Where:

- Y is the dependent variable (CPU clock speed)
- β_0 is the intercept (the value of Y when all independent variables are zero)
- $\beta_1, \beta_2, ..., \beta_n$ are the coefficients (slopes) associated with the respective independent variables
- $x_1, x_2, ..., x_n$ are the independent variables (CPU specifications)
- ε is the error term, representing the difference between the observed values and the predicted values

The process of linear regression involves estimating the values of the coefficients (β_0, β_1, β_2, ..., β_n) using a set of observed data points. This estimation is typically performed using the method of least squares, which aims to minimize the sum of squared differences between the observed values and the predicted values obtained from the linear equation. Linear regression models can be classified into two main types:

- Simple Linear Regression: This model involves only one independent variable and is represented by the equation $Y = \beta_0 + \beta_1 x + \varepsilon$.
- Multiple Linear Regression: This model involves two or more independent variables and is represented by the equation $Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \varepsilon$

Procedure for Linear Regression:

1. Formulate the Model: Define the relationship between CPU clock speed and its specifications.
2. Estimate Coefficients: Use the method of least squares to estimate the values of the coefficients ($\beta_1, \beta_2, ..., \beta_n$).
3. Evaluate the Model:
    - R-Squared: Measures the proportion of variance in the dependent variable explained by the independent variables.

- P-Values: Assess the significance of each coefficient.
- Residual Analysis: Check for patterns in the residuals to validate assumptions.

4. Interpret Coefficients: Understand the magnitude and direction of the impact of each independent variable on CPU clock speed. For instance, a positive coefficient for the number of cores indicates that as the number of cores increases, the clock speed tends to increase.

5. Predict: Use the model to predict CPU clock speeds based on new values of the independent variables. The predicted values can guide decisions in CPU design and marketing strategies.

Assumptions of Linear Regression:

- Linearity: The relationship between the dependent and independent variables is linear.
- Normality of Residuals: The residuals (errors) are normally distributed.
- Homoscedasticity: Constant variance of residuals.
- Independence: Observations are independent of each other.

By understanding and applying linear regression, we can develop robust models to predict CPU clock speeds based on their specifications, providing valuable insights for manufacturers and consumers.

4.3.1 Formulas (for reference)

Covariance and Correlation

- Covariance between the random variables X and Y is:

$$\text{cov}(X,Y) = \sigma_{XY} = E[(X - E(X))(Y - E(Y))] = E(XY) - E(X)E(Y)$$

- The correlation between the random variables X and Y is:

$$\rho_{XY} = \frac{\text{cov}(X,Y)}{\sqrt{V(X)V(Y)}} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y} = \frac{S_{xy}}{\sqrt{S_{xx} \cdot S_{yy}}}, \ -1 \leq \rho_{XY} \leq 1$$

Least Square Method

A statistical procedure to find the best fit for a set of data points.
Suppose that we have n pairs of observations $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$. We have to find the linear regression model for the data as $y_i = \beta_0 + \beta_1 x_i + \varepsilon$
The sum of the squares of the deviations of the observations from the true regression line is

$$L = \sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_1) \rightarrow min$$

The least squares estimators $\hat{\beta}_0, \hat{\beta}_1$ must satisfy

$$\begin{cases} n\hat{\beta}_0 + \hat{\beta}_1 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i \\ \hat{\beta}_0 \sum_{i=1}^{n} x_i + \hat{\beta}_1 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i \end{cases}$$

The least squares estimates of the intercept and the slope in the simple linear regression model are

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}, \quad \hat{\beta}_1 = \frac{S_{xy}}{S_{xx}}$$

where

$$S_{xy} = \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) = \left(\sum_{i=1}^{n} x_i y_i \right) - \frac{1}{n} \left(\sum_{i=1}^{n} x_i \right) \left(\sum_{i=1}^{n} y_i \right)$$

$$S_{xx} = \sum_{i=1}^{n} (x_i - \bar{x})^2 = \left(\sum_{i=1}^{n} x_i^2 \right) - \frac{1}{n} \left(\sum_{i=1}^{n} x_i \right)^2$$

The **fitted** or **estimated regression line** is therefore:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

- $e_i = y_i - \hat{y}_i$: The error in the fit of the model to the i^{th} observation y_i and is called **residual**.
- $\text{SSE} = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$: The sum of squares of the residuals.
- $\text{SST} = \sum_{i=1}^{n} (y_i - \bar{y})^2 = \sum_{i=1}^{n} y_i^2 - n\bar{y}^2 = S_{yy}$: The total sum of square of the response variable.
- $\text{SSR} = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2 = \beta_1 S_{xy}$: The sum of squares for regression.
- $r^2 = 1 - \frac{SSE}{SST} = \rho_{XY}^2$: The coefficient of determination.

Estimator of Variance: $\hat{\sigma}^2 = \frac{SSE}{n-2} = \frac{SST - \beta_1 S_{xy}}{n-2}$

4.4 Imputation

Most datasets, including the one used in this report, are imperfect, as some rows of values are empty of values. We must decide on how to treat empty/missing values, the process of which is called imputation.

Some common imputation methods are:

- **Listwise deletion**: Remove whole rows/list. Also known as complete case deletion.
- **Median imputation**: Fill with median of available values.
- **Mean imputation**: Fill with mean of available values.
- **Regression imputation**: Use linear regression to extrapolate missing values.

Some less common imputation methods are:

- **Pairwise deletion**: Similar to listwise deletion but only remove when missing required variables.
- **Hot-deck imputation**: Pick randomly from available values.
- **Cold-deck imputation**: Pick from a donor dataset.

We utilized various imputation methods to preprocess data.

4.5 Remark

Before any processing work, data must be put through filters, and to that end necessary **imputation** steps must be taken. To predict the clock speed of CPUs, we primarily rely on **Linear Regression** as it directly models the relationship between the CPU specifications (independent variables) and the clock speed (dependent variable). **ANOVA** can be used as a supplementary method to understand the influence of categorical variables on CPU clock speeds, but it is not essential for the prediction model itself.

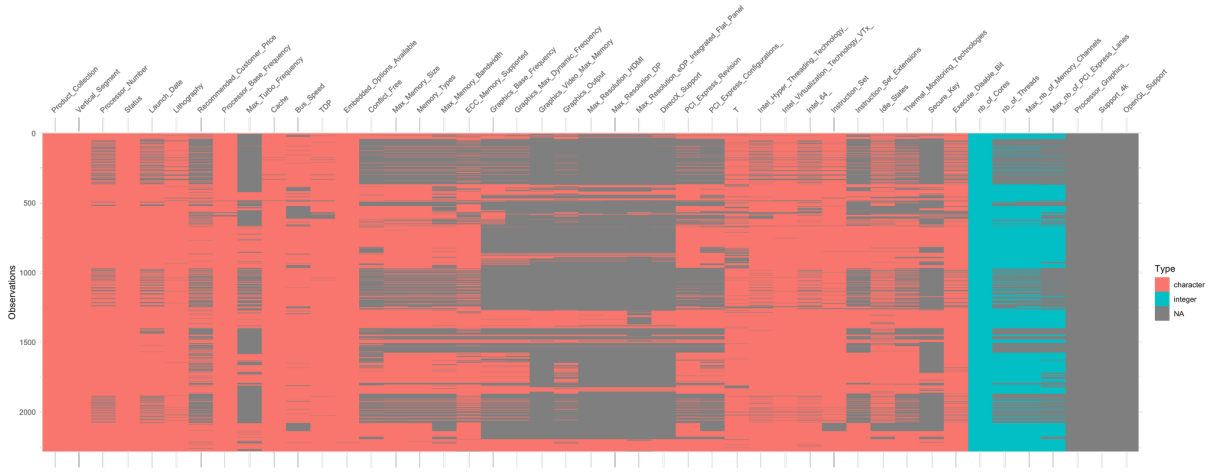
5 Data Pre-processing

Preprocessing the data involves cleaning and organizing it to ensure accuracy, consistency, and ease of use. This step includes identifying and addressing duplicate entries, missing or incomplete data, and other inconsistencies. Proper preprocessing enhances the data's integrity, leading to more reliable results.

The dataset was imported into the workspace using the `read.csv()` function. To inspect the data, the first six lines were displayed using the `head()` function.

```
                        Product_Collection Vertical_Segment Processor_Number      Status Launch_Date Lithography
1 7th Generation Intel® Core™ i7 Processors          Mobile          i7-7Y75    Launched       Q3'16       14 nm
2 8th Generation Intel® Core™ i5 Processors          Mobile         i5-8250U    Launched       Q3'17       14 nm
3 8th Generation Intel® Core™ i7 Processors          Mobile         i7-8550U    Launched       Q3'17       14 nm
4          Intel® Core™ X-series Processors         Desktop          i7-3820 End of Life       Q1'12       32 nm
5 7th Generation Intel® Core™ i5 Processors          Mobile          i5-7Y57    Launched       Q1'17       14 nm
6     Intel® Celeron® Processor 3000 Series          Mobile            3205U    Launched       Q1'15       14 nm
  Recommended_Customer_Price nb_of_Cores nb_of_Threads Processor_Base_Frequency Max_Turbo_Frequency           Cache
1                   $393.00            2             4                 1.30 GHz            3.60 GHz  4 MB SmartCache
2                   $297.00            4             8                 1.60 GHz            3.40 GHz  6 MB SmartCache
3                   $409.00            4             8                 1.80 GHz            4.00 GHz  8 MB SmartCache
4                   $305.00            4             8                 3.60 GHz            3.80 GHz 10 MB SmartCache
5                   $281.00            2             4                 1.20 GHz            3.30 GHz  4 MB SmartCache
6                   $107.00            2             2                 1.50 GHz                              2 MB
    Bus_Speed   TDP Embedded_Options_Available Conflict_Free Max_Memory_Size                 Memory_Types
1  4 GT/s OPI 4.5 W                        No           Yes           16 GB          LPDDR3-1866, DDR3L-1600
2  4 GT/s OPI  15 W                        No           Yes           32 GB           DDR4-2400, LPDDR3-2133
3  4 GT/s OPI  15 W                        No           Yes           32 GB           DDR4-2400, LPDDR3-2133
4 5 GT/s DMI2 130 W                        No                      64.23 GB              DDR3 1066/1333/1600
5  4 GT/s OPI 4.5 W                        No           Yes           16 GB          LPDDR3-1866, DDR3L-1600
6 5 GT/s DMI2  15 W                        No           Yes           16 GB DDR3L 1333/1600 LPDDR3 1333/1600
  Max_nb_of_Memory_Channels Max_Memory_Bandwidth ECC_Memory_Supported Processor_Graphics_ Graphics_Base_Frequency
1                         2           29.8 GB/s                   No                 NA                 300 MHz
2                         2           34.1 GB/s                   No                 NA                 300 MHz
3                         2           34.1 GB/s                   No                 NA                 300 MHz
4                         4           51.2 GB/s                   No                 NA
5                         2           29.8 GB/s                   No                 NA                 300 MHz
6                         2           25.6 GB/s                                      NA                 100 MHz
  Graphics_Max_Dynamic_Frequency Graphics_Video_Max_Memory Graphics_Output Support_4k Max_Resolution_HDMI
1                      1.05 GHz                  16 GB eDP/DP/HDMI/DVI         NA      4096x2304@24Hz
2                      1.10 GHz                  32 GB eDP/DP/HDMI/DVI         NA      4096x2304@24Hz
3                      1.15 GHz                  32 GB eDP/DP/HDMI/DVI         NA      4096x2304@24Hz
4                                                                              NA
5                       950 MHz                  16 GB eDP/DP/HDMI/DVI         NA      4096x2304@24Hz
6                       800 MHz                        eDP/DP/HDMI             NA
  Max_Resolution_DP Max_Resolution_eDP_Integrated_Flat_Panel DirectX_Support OpenGL_Support PCI_Express_Revision
1    3840x2160@60Hz                          3840x2160@60Hz              12            NA                     3
2    4096x2304@60Hz                          4096x2304@60Hz              12            NA                     3
3    4096x2304@60Hz                          4096x2304@60Hz              12            NA                     3
4                                                                                       NA                     2
5    3840x2160@60Hz                          3840x2160@60Hz              12            NA                     3
6                                                                     11.2/12            NA                     2
  PCI_Express_Configurations_ Max_nb_of_PCI_Express_Lanes      T Intel_Hyper_Threading_Technology_
1   1x4, 2x2, 1x2+2x1 and 4x1                          10  100°C                              Yes
2   1x4, 2x2, 1x2+2x1 and 4x1                          12  100°C                              Yes
3   1x4, 2x2, 1x2+2x1 and 4x1                          12  100°C                              Yes
4                                                      40 66.8°C                              Yes
5   1x4, 2x2, 1x2+2x1 and 4x1                          10  100°C                              Yes
6                     4x1 2x4                          12  105°C                               No
  Intel_Virtualization_Technology_VTx_ Intel_64_ Instruction_Set Instruction_Set_Extensions Idle_States
1                                  Yes       Yes          64-bit        SSE4.1/4.2, AVX 2.0         Yes
2                                  Yes       Yes          64-bit        SSE4.1/4.2, AVX 2.0         Yes
3                                  Yes       Yes          64-bit        SSE4.1/4.2, AVX 2.0         Yes
4                                  Yes       Yes          64-bit           SSE4.2, AVX, AES         Yes
5                                  Yes       Yes          64-bit        SSE4.1/4.2, AVX 2.0         Yes
6                                  Yes       Yes          64-bit                 SSE4.1/4.2         Yes
  Thermal_Monitoring_Technologies Secure_Key Execute_Disable_Bit
1                             Yes        Yes                 Yes
2                             Yes        Yes                 Yes
3                             Yes        Yes                 Yes
4                             Yes                            Yes
5                             Yes        Yes                 Yes
6                             Yes        Yes                 Yes
```

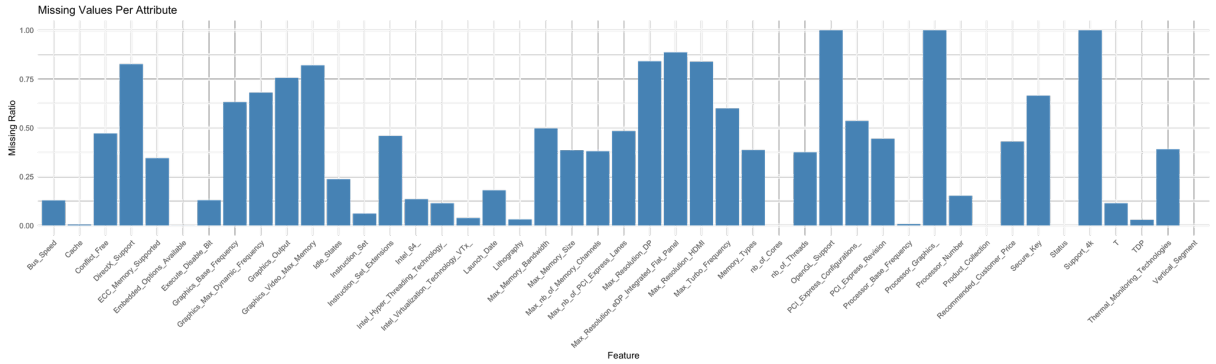
After importing the data, it was summarized using the `vis_dat()` function from the `visdat` library, which visualizes the data type of each attribute across all observations.



Dataset visualization

It can be seen that the dataset mostly consist of char types with some numeric features and a large of portion of data is missing. Therefore, to work with the dataset, feature selection followed by handling missing values has to be done and data transformation must take place in order to convert the char types into numeric for analyzing purposes.

5.1 Missing values examination and attributes selection



Missing data ratio per attribute

For further analysis, the attributes with less than 50% of missing values are categorized into quintiles based on their missing value percentage ranges.

Category	Attributes
Less than 10%	Product_Collection, Vertical_Segment, Status, Lithography, nb_of_Cores, Processor_Base_Frequency, Cache, TDP, Embedded_Options_Available, Intel_Virtualization_Technology_VTx, Instruction_Set
10% to 20%	Processor_Number, Launch_Date, Bus_Speed, T, Intel_Hyper_Threading_Technology, Intel_64,'Execute_Disable_Bit
20% to 30%	Idle_States
30% to 40%	nb_of_Threads, Max_Memory_Size, Memory_Types, Max_nb_of_Memory_Channels, ECC_Memory_Supported, Thermal_Monitoring_Technologies
40% to 50%	Recommended_Customer_Price, Conflict_Free, Max_Memory_Bandwidth, PCI_Express_Revision, Max_nb_of_PCI_Express_Lanes, Instruction_Set_Extensions
Above 50%	Remaining attributes

Table 1: Attributes categorization based on missing percentage

5.2 Data Relevance and Usefulness

To ensure the integrity and robustness of our analysis, attributes with more than 50% of absent information will be discarded. The others will be selected based on their relation to the base frequency of a CPU and the ratio of missing values. Specifically, the attributes of interest are:

- **Processor_Base_Frequency**: This attribute represents the base clock speed of the CPU, measured in gigahertz (GHz). Higher base frequencies generally indicate faster processing capabilities and overall performance.

- **nb_of_Cores**: The number of cores in a CPU impacts its ability to handle multiple tasks simultaneously. CPUs with more cores can potentially support higher clock speeds under optimal conditions, affecting overall performance trends.
- **nb_of_Threads**: This attribute represents the number of threads supported by the CPU, which is often influenced by technologies like Intel's Hyper-Threading. A higher number of threads can contribute to better resource utilization and potentially higher effective clock speeds for certain workloads.
- **TDP (Thermal Design Power)**: The TDP indicates the maximum amount of heat the CPU cooling system needs to dissipate. This attribute is closely related to the CPU's power consumption and thermal characteristics, which can impact clock speed potential and performance.
- **Lithography**: This attribute refers to the manufacturing process node (e.g., 14nm, 10nm) used in producing the CPU. Advancements in manufacturing processes can lead to higher clock speeds and improved efficiency in newer CPU generations.

Our objective is to analyze historical trends in CPU clock speeds using these attributes. Understanding how clock speeds have evolved across different CPU generations, architectural improvements, and technological shifts is crucial for identifying patterns and making informed predictions. By focusing on these critical attributes, we aim to uncover insights into the factors driving CPU clock speed improvements and contributing to the overall technological progress in CPU development.

5.3 Handling Missing Values

To handle missing data, we employed median imputation. Although methods like listwise deletion, mean imputation, or regression imputation have their own merits, median imputation was chosen based on the following considerations:

- **Simplicity and Effectiveness**: Median imputation is straightforward to implement and often provides a better central tendency measure for skewed distributions compared to mean imputation.
- **Neutral Algorithm**: Unlike regression imputation, which may introduce sample bias, median imputation does not assume a specific relationship between the missing values and other variables.
- **Data Retention**: Removing rows with missing values (listwise deletion) would significantly shrink the size of our dataset. Median imputation allows us to retain as much data as possible while ensuring that the central tendency of the dataset is maintained.

```r
    # Function to convert GHz to MHz if needed
    convert_units <- function(x) {
    if (str_detect(x, "GHz")) {
        value <- as.numeric(str_extract(x, "[0-9.]+")) * 1000
        return(value)
    }
    return(as.numeric(str_extract(x, "[0-9.]+")))
    }

    # Apply conversion and impute missing values with median
    data <- data %>%
    mutate(across(everything(), ~ {
        # Convert units if necessary and replace with numeric values
        . <- sapply(., function(cell) ifelse(is.na(cell) | cell == "", NA, convert_units(cell)))
        # Impute missing values with median
        . <- ifelse(is.na(.), median(., na.rm = TRUE), .)
        return(.)
    }))
```

The following `R` code handles the preprocessing of our dataset by converting units where necessary and imputing missing values with the median. The `convert_units` function checks if the data contains "GHz" and converts it to "MHz" by multiplying the numeric value by 1000. This ensures uniformity in the measurement units. The mutate function from the `dplyr` package is used to apply these conversions across all columns of the dataset. For each cell, if the value is missing or empty, it is replaced with `NA`, then converted using the `convert_units` function. Finally, any remaining `NA` values are imputed with the median of the respective column, ensuring the dataset remains complete and reliable for further analysis.

5.4 Handling outliers

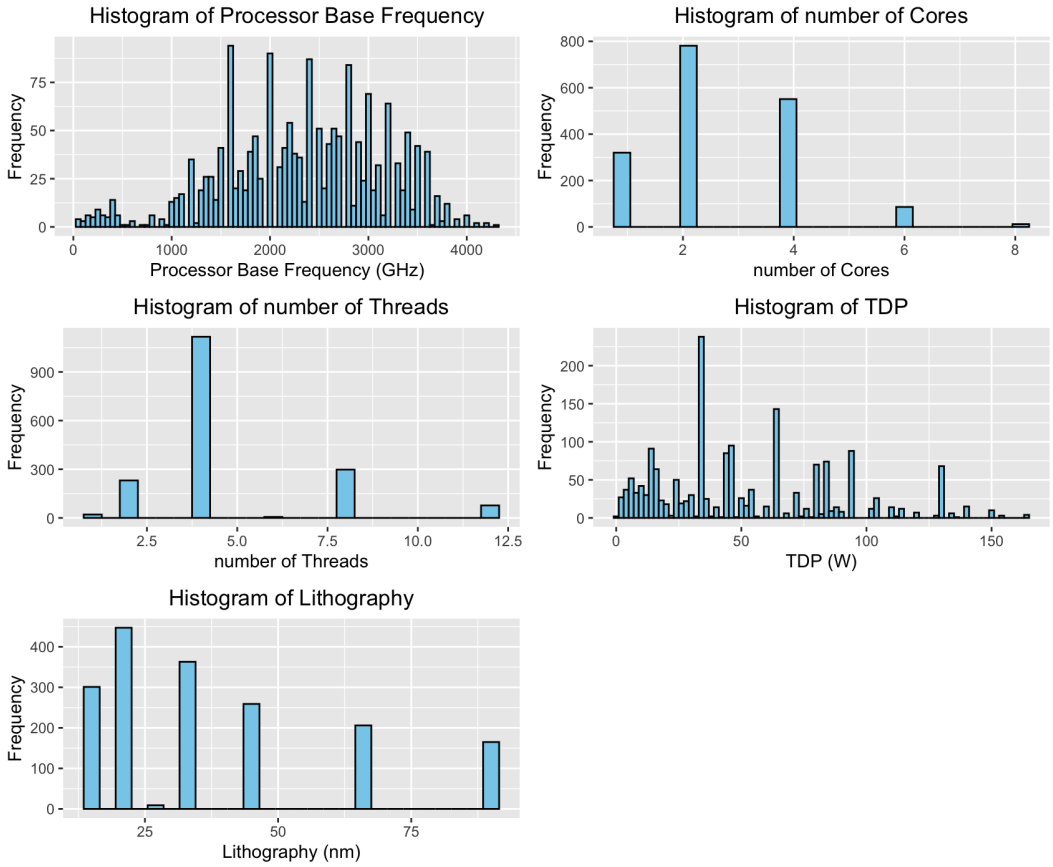
Handling outliers in data is crucial to ensure robust analysis and modeling. Outliers can skew statistical measures and adversely affect model performance.

```r
# Identify Outliers Function
identify_outliers <- function(x) {
Q1 <- quantile(x, 0.25, na.rm = TRUE)
Q3 <- quantile(x, 0.75, na.rm = TRUE)
IQR <- Q3 - Q1
lower_bound <- Q1 - 1.5 * IQR
upper_bound <- Q3 + 1.5 * IQR
return(x < lower_bound | x > upper_bound)
}

# Apply the function to identify outliers for each relevant column
outliers <- sapply(data[, desired_cols], identify_outliers)

# Combine the outliers into a single logical vector indicating any row with an outlier
combined_outliers <- apply(outliers, 1, any)

# Print number of outliers in each column
print(colSums(outliers))

# Remove rows with outliers
data <- data[!combined_outliers, ]
```

The provided `R` code snippet handles outliers in a dataset using the interquartile range (IQR) method. It defines a function `identify_outliers` that calculates outliers based on quartiles and IQR for each specified column in data. The function outputs a matrix where each element indicates if the corresponding value in `data` is an outlier (TRUE or FALSE). By combining these results across rows, it creates a logical vector `combined_outliers` to identify rows containing any outliers. The code then prints the number of outliers detected in each column using `colSums(outliers)`, facilitating insight into outlier distribution across the dataset. Finally, it updates data by removing rows identified as outliers (`!combined_outliers`), ensuring a cleaner dataset for subsequent analysis or modeling.

6 Descriptive Statistic

6.1 Distribution and Histograms



- Processor Base Frequency:
  - The frequencies range from 0 to around 4200 MHz, with most values concentrated between 1000 and 3500 MHz.
  - The distribution is relatively uniform within this range, with a slight peak around 2500 MHz.
- Number of Cores:
  - The distribution of the number of cores is right-skewed.
  - Most CPUs have 2 or 4 cores, with a significant peak at 2 cores.
  - There are fewer CPUs with 6 or 8 cores, indicating that higher core counts are less common in the dataset.
  - This suggests that while multi-core processors are common, high-core-count CPUs are less frequent.

- Number of Threads:
  - Similar to the number of cores, the distribution of the number of threads is right-skewed.
  - Most CPUs have either 4 or 8 threads, with a significant peak at 4 threads.
  - The dataset contains a smaller number of CPUs with other thread counts, such as 2, 6, and 12 threads.
  - There are some CPUs with a significantly higher number of threads, indicating the presence of high-performance CPUs with technologies like Hyper-Threading.
- TDP (Thermal Design Power):
  - TDP values range widely from 0 to 150 watts. There is a concentration of CPUs with TDP values around 50 watts, with several peaks in the distribution.
  - There are a few CPUs with higher TDP values, which typically correspond to more powerful processors that require more cooling.
  - This indicates that most CPUs are designed to operate within a standard thermal envelope, but high-performance CPUs demand more power.
- Lithography:
  - The lithography histogram shows several peaks, indicating different manufacturing process nodes.
  - Most CPUs have lithography sizes of either 14 nm or 22 nm, with significant peaks at these values.
  - Other common sizes include 32 nm and 45 nm, with fewer CPUs having lithography sizes of 10 nm or 7 nm.
  - This reflects the advancements in semiconductor manufacturing technology, with newer CPUs being produced at smaller process nodes for improved performance and efficiency.

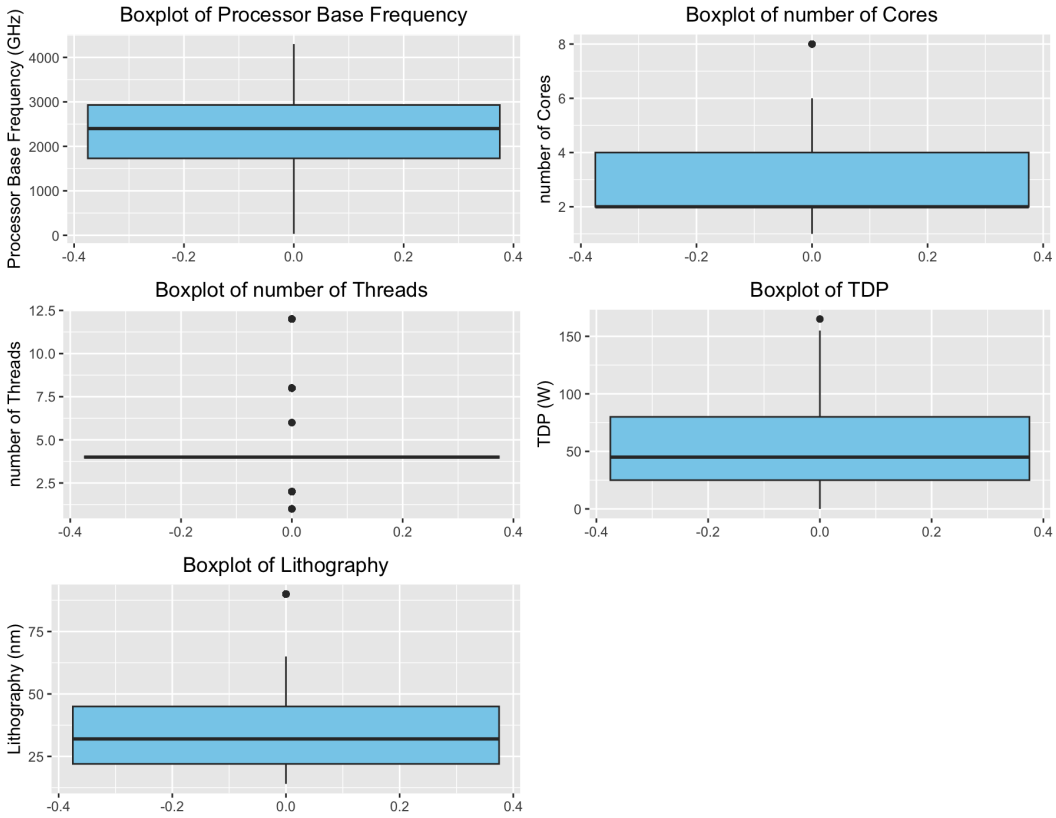
Implications:

- Processor Base Frequency: The relatively normal distribution suggests a common range of base frequencies, with few extreme values. This will influence our regression model, as we need to account for these outliers.
- Number of Cores and Threads: The right-skewed distributions indicate that while multi-core and multi-threading capabilities are common, the extent varies widely. High-core and high-thread CPUs, although fewer, represent the high-performance segment and should be carefully considered in our model.
- TDP: The distribution suggests that most CPUs are designed within certain power and thermal limits, but there are high-performance CPUs with higher TDP. This attribute is crucial as it impacts the CPU's ability to sustain higher clock speeds under load.

- Lithography: The presence of multiple peaks reflects the evolution of manufacturing processes over time. This attribute is essential for our model as newer process nodes typically allow for higher clock speeds and better efficiency.

By understanding these distributions, we can better prepare our data for inferential statistics and modeling, ensuring that our regression model accurately captures the relationship between these attributes and CPU clock speeds. This will involve handling outliers appropriately, scaling the features, and ensuring that the model is trained on a representative sample of the data.

6.2 Boxplots for Outliers



The boxplots provide a visual representation of the distribution, central tendency, and variability of key attributes in the dataset, as well as the presence of outliers. Here's a detailed explanation of each boxplot:

- Boxplot of Processor Base Frequency (GHz):
  - The median processor base frequency is around 2000 MHz. The interquartile range (IQR) spans from approximately 1000 to 3000 MHz. There are no significant outliers beyond this range, indicating a relatively symmetric distribution of base frequencies.
- Boxplot of Number of Cores:

  - The median number of cores is 2. The IQR ranges from 2 to 4 cores. There is one outlier at 8 cores, suggesting that while most CPUs have between 2 and 4 cores, there are a few models with a significantly higher core count.

- Boxplot of Number of Threads:
  - The median number of threads is 4. The IQR ranges from 4 to 8 threads. There are several outliers, both below 4 and above 8 threads, indicating that there are CPUs with unusual thread counts compared to the majority in the dataset.

- Boxplot of TDP (W):
  - The median TDP is around 50 watts. The IQR spans from approximately 30 to 75 watts.
  - There is one outlier above 150 watts, indicating that most CPUs have moderate power consumption, with a few exceptions having significantly higher TDP values.

- Boxplot of Lithography (nm):
  - The median lithography size is 22 nm. The IQR ranges from 22 to 32 nm.
  - There are a few outliers below 22 nm and one above 45 nm, indicating that while most CPUs have lithography sizes within this range, there are some with newer or older manufacturing technologies.

Implications: The boxplots provide several insights that are crucial for our project on predicting CPU clock speeds:

- Processor Base Frequency:
  - The central tendency and spread of base frequencies are important for understanding the typical performance range of CPUs in the dataset.
  - The presence of outliers indicates that while most CPUs have base frequencies in a common range, some high-performance CPUs exist with higher base frequencies, which need to be considered in our model.

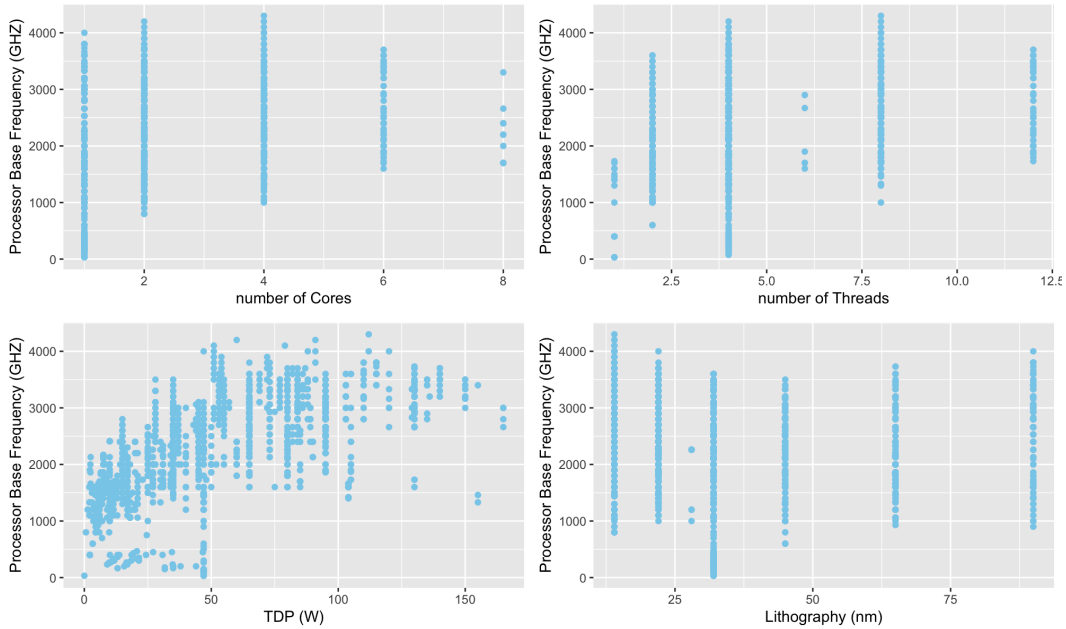
- Number of Cores and Threads:
  - The distributions show that multi-core and multi-threaded CPUs are common, but high-core and high-thread counts are less frequent, representing high-performance segments.
  - Outliers in these attributes reflect high-performance CPUs, which can affect the overall performance trends and must be accurately modeled.

- TDP:
  - The central tendency and variability in TDP indicate the power and thermal characteristics of most CPUs.

  - The outliers suggest that while most CPUs operate within a standard thermal range, some require much higher or lower power, influencing their ability to sustain higher clock speeds.

- Lithography:
  - The distribution of lithography values highlights the evolution of manufacturing technology, with most CPUs produced using advanced nodes like 22 nm and 32 nm.
  - The outlier at few outliers below 22 nm and one above 45 nm represents older technology, which typically correlates with lower efficiency and performance.

6.3 Scatter plot for relationship between Processor Base Frequency with the others



- Processor Base Frequency and Number of Cores
  - The plot in the top left shows the relationship between the number of cores in a processor and its base frequency.
  - The data points indicate an inverse relationship - as the number of cores increases, the processor's base frequency tends to decrease. This suggests that adding more cores may come at the cost of lower individual core frequencies.

- Processor Base Frequency and Number of Threads
  - The plot in the top right shows the relationship between the number of threads in a processor and its base frequency.

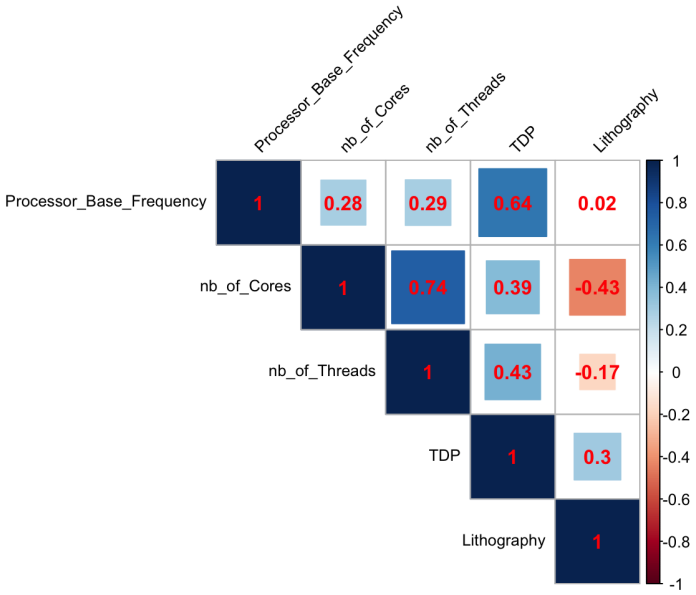
  - The data points suggest a positive relationship - as the number of threads increases, the processor's base frequency generally increases as well. This implies that more threads can be supported without significantly impacting the core frequencies.

- Processor Base Frequency and TDP
  - The bottom left plot displays the relationship between the processor's Thermal Design Power (TDP) and its base frequency.
  - The wide distribution of data points indicates that processor performance, as measured by base frequency, is not solely determined by power consumption (TDP). There are processors with a wide range of frequencies across various TDP values.

- Processor Base Frequency and Lithography
  - The bottom right plot illustrates the relationship between the processor's lithography (manufacturing process) and its base frequency.
  - The data suggests that processors manufactured using smaller lithography sizes (in nanometers) tend to have higher base frequencies. This is likely due to the performance improvements enabled by advancements in semiconductor manufacturing technology.

Overall, this set of scatter plots provides a comprehensive view of how different processor hardware specifications, such as core count, thread count, power consumption, and manufacturing process, are related to the fundamental performance metric of processor base frequency.

6.4 Correlation Matrix



The correlation matrix provides a comprehensive view of the linear relationships between pairs of variables in our dataset. Each cell in the matrix represents the correlation coefficient between two variables, ranging from -1 (perfect negative correlation) to 1 (perfect positive correlation). Here's a detailed explanation of the matrix for our project:

Key Observations:

- Processor Base Frequency:
  - Correlation with TDP: The strongest positive correlation is with TDP (0.64). This indicates that CPUs with higher base frequencies generally have higher thermal design power requirements. This is expected as higher clock speeds typically result in greater power consumption and heat generation.
  - Negative Correlation with Lithography: There is a moderate negative correlation with Lithography (0.02). This suggests that CPUs manufactured with smaller process nodes tend to have higher base frequencies. Advanced manufacturing technologies often result in better performance characteristics.
- Number of Cores:
  - Strong Positive Correlation with Number of Threads (0.74): This high correlation is expected as CPUs with more cores generally support more threads, especially with technologies like Hyper-Threading.

  - Negative Correlation with Lithography (-0.43): Indicates that CPUs with a higher number of cores tend to be manufactured using smaller lithography nodes, aligning with advancements in CPU design that pack more cores into smaller spaces.

- Number of Threads:
  - Positive Correlation with TDP (0.43): Similar to the base frequency, a higher number of threads is associated with higher TDP, reflecting increased power consumption and heat dissipation needs.

- TDP:
  - Slight Positive Correlation with Lithography (0.3): This weak correlation suggests that the thermal design power does not have a strong relationship with the manufacturing process node.

- Lithography:
  - Negative Correlations with Other Variables: Lithography shows a general negative correlation with performance-related attributes, reinforcing the trend that smaller manufacturing nodes (indicative of newer technologies) are associated with higher performance capabilities.

Implications:

- The correlation matrix is crucial for understanding the relationships between different CPU attributes and their collective impact on clock speed. Here are some key takeaways for our project on predicting CPU clock speeds:
- TDP and Processor Base Frequency: The strong positive correlation suggests that models predicting clock speed should account for TDP as a significant predictor.
- Manufacturing Process (Lithography): The negative correlations with performance attributes highlight the importance of considering lithography advancements in performance modeling.
- Number of Cores and Threads: The strong inter-correlation indicates that either variable could be used interchangeably in some modeling contexts, but including both provides a more nuanced understanding of CPU capabilities.

7 Inferential Statistics

7.1 Two-way analysis of variance(ANOVA)

Our model is about using two-way ANOVA to determine the relationship between the dependent Processor Base Frequency with two independent which are number of Cores and Thermal Design Power(TDP). Because we want to check validity of this ANOVA model, we will have two assumption for this:

Normality: Processor Base Frequency should be normal distribution for each combination of number of Cores and TDP.

Homogeneity: Processor Base Frequency should be roughly across all combinations of number of Cores and TDP.

7.1.1 Verify the assumption

Normality: verify if the data is normal distribution
We use Shapiro-Wilk normality test to check if this model is normal distribution:

Null hypothesis H_0: Processor Base Frequency should be approximately normally distributed for each combinations of number of Cores and TDP.

Alternative hypothesis H_1: Processor Base Frequency should not be approximately normally distributed for each combinations of number of Cores and TDP.

```
        Shapiro-Wilk normality test

data:  residuals(model_normality)
W = 0.98049, p-value = 1.047e-14
```

Figure 1: Shapiro-Wilk's test

From the result, p-value is smaller than 0.05 so we reject H_0 and accept H_1. Therefore, Processor Base Frequency should not be approximately normally distributed for each combinations of number of Cores and TDP. So we need to plot out the model diagnose plot to check if it is normal distribution or not:

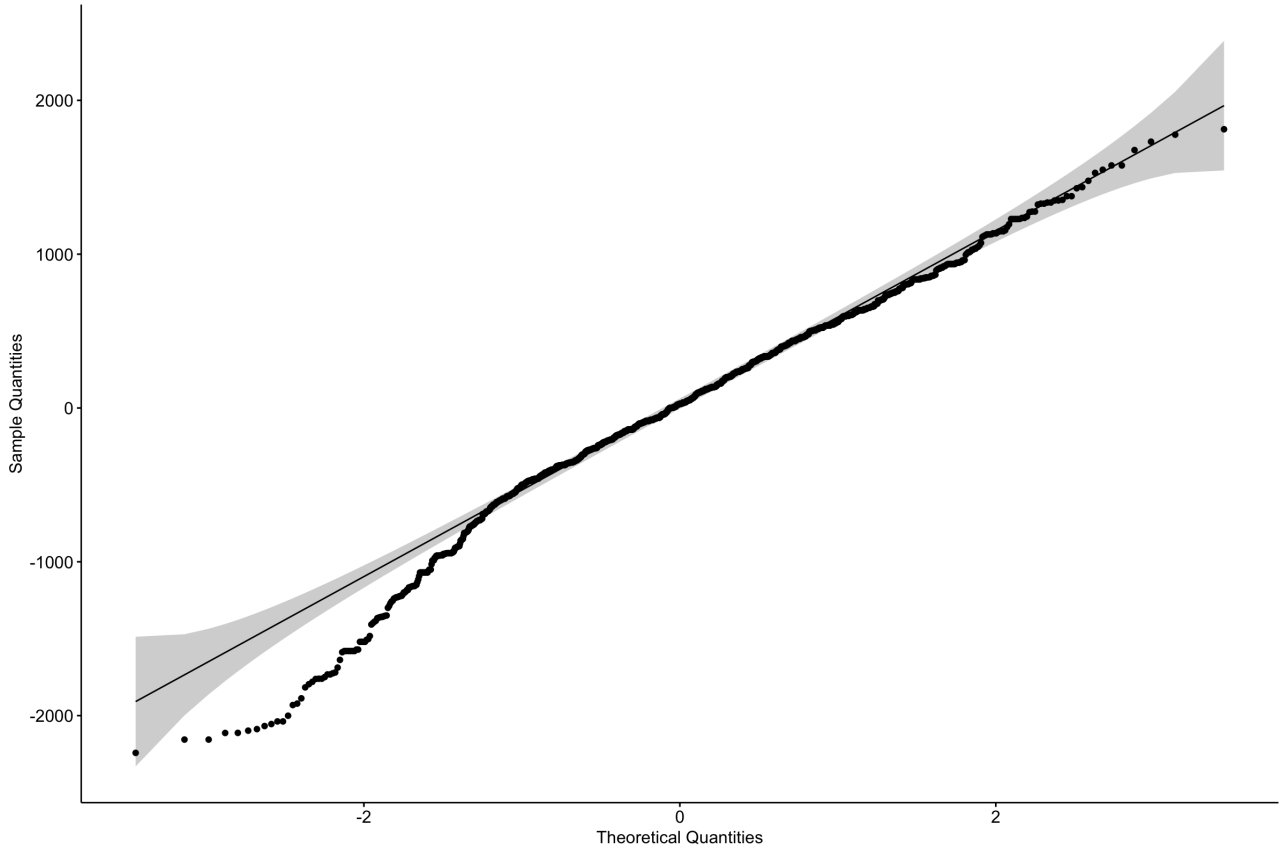


Figure 2: ggqqplot from ANOVA test

We see that in ggqqplot, most of the points lie closely on the line. So we can assume that Processor Base Frequency is approximately normally distributed for each combinations of number of Cores and TDP.

Homogeneity: verify the uniformity of the variance.

We use Levene's test to check if this model is homogeneity:

Null hypothesis H_0: Processor Base Frequency is homogeneity across groups of number of Cores and TDP.

Alternative hypothesis H_1: Processor Base Frequency is not homogeneity across groups of number of Cores and TDP.

```
Levene's Test for Homogeneity of Variance (center = median)
        Df F value    Pr(>F)
group  446  1.8013 < 2.2e-16 ***
      1836
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 3: Leneve's Test for Homogeneity of Variance

From the result, we see that p-value is smaller than 0.05 so we reject H_0 and accept H_1. Therefore, Processor Base Frequency is not homogeneity across groups of number of Cores and TDP.

7.1.2 Calculate ANOVA

Null hypothesis H_0: Processor Base Frequency follows the same distribution across group defined by the independent variables number of Cores and TDP.

Alternative hypothesis H_1: Processor Base Frequency follows different distribution across group defined by the independent variables number of Cores and TDP.

```
                      Df    Sum Sq  Mean Sq F value Pr(>F)
factor(nb_of_Cores)    4 120140512 30035128  165.57 <2e-16 ***
factor(TDP)          124 739679213  5965155   32.88 <2e-16 ***
Residuals           1621 294054318   181403
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 4: Calculate ANOVA

From the result, we see that Pr(>F) value is smaller than 0.05 so we reject H_0 and accept H_1. Therefore, Processor Base Frequency follows different distribution across group defined by the independent variables number of Cores and TDP.

7.2 Multiple Linear Regression to Predict CPU Clock Speed

7.2.1 Data Splitting

After finalizing the data we need for statistical analysis, we have to split the data further into two subsets: **training set and test set**. Training set helps us build and train the model, allowing it to learn the patterns and connections within our data. Then, the test set will be tested on the validated model to give an objective assessment of the model's efficacy and performance on fresh, unseen data. In data science and machine learning, this train-test split strategy is a

standard procedure since it ensures that the model generalises effectively to new data and helps prevent over-fitting. In this project, the training set consists of 70% of the original data, and the remaining 30% makes up the test set. This is the code we implemented to split the data:

```
smp_size <- floor(0.70 * nrow(intel_cpu_subset))
set.seed(123)
train_ind <- sample(seq_len(nrow(intel_cpu_subset)), size = smp_size)

train_set <- intel_cpu_subset[train_ind, ]
test_set <- intel_cpu_subset[-train_ind, ]
```

7.2.2 Regression Model

The main objective of this section is constructing a model that portrays the effect of other factors on the CPU clock speed. To achieve this we applied a Multi Regression model, in which the dependent variable is Processor Base Frequency - the variable representing CPU clock speed, and the rest are independent. Our model appears as the formula below:

$$\texttt{Processor Base Frequency} = \beta_0 + \beta_1 \times \texttt{nb\_of\_Cores} + \beta_2 \times \texttt{nb\_of\_Threads} + \beta_3 \times \texttt{TDP} + \beta_4 \times \texttt{Lithography}$$

We start with our **first model**, also known as the **base model**, built with all the independent variables available.

```
Call:
lm(formula = Processor_Base_Frequency ~ nb_of_Cores + nb_of_Threads +
    TDP + Lithography, data = train_set)

Residuals:
     Min       1Q   Median       3Q      Max
-2378.63  -285.50    63.28   408.42  2140.33

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   1999.4277    61.9557  32.272  < 2e-16 ***
nb_of_Cores   -116.5358    21.0204  -5.544 3.62e-08 ***
nb_of_Threads   22.9547    10.7811   2.129   0.0334 *
TDP             17.9874     0.6331  28.412  < 2e-16 ***
Lithography    -10.6716     1.0071 -10.596  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 596.1 on 1220 degrees of freedom
Multiple R-squared:  0.4624,    Adjusted R-squared:  0.4606
F-statistic: 262.3 on 4 and 1220 DF,  p-value: < 2.2e-16
```

Figure 5: The base model

In this model, our respone variables consist of: `nb_of_Cores`, `nb_of_Threads`, `TDP` and `Lithography`. Now, we should remove the variables that are proven insignificant to our analysis, which can be determined based on their Pr values (last column). If $Pr < 0.05$, the variable is significant. With this insight, there is no variables that should be deducted. Hence, the base model is also our final model.

$\texttt{Processor Base Frequency} = 1999.4277 - 116.5358 \times \texttt{nb\_of\_Cores} + 22.9547 \times \texttt{nb\_of\_Threads} + 17.9874 \times \texttt{TDP} - 10.6716 \times \texttt{Lithography}$

Observing the results R gave on the model, the p - value correlating with F statistics is less than 2.2×10^{-16}. This suggests that our data is robust and valuable for statistical analysis. Additionally, it guarantees that future results from this model provide good evaluations about the relationship between Processor Base Frequency and the remaining variables. Generally, the regression coefficients (β_i) and the p - values hold the most influences on the independent variables.

7.2.3 Assumptions of Linear Regression

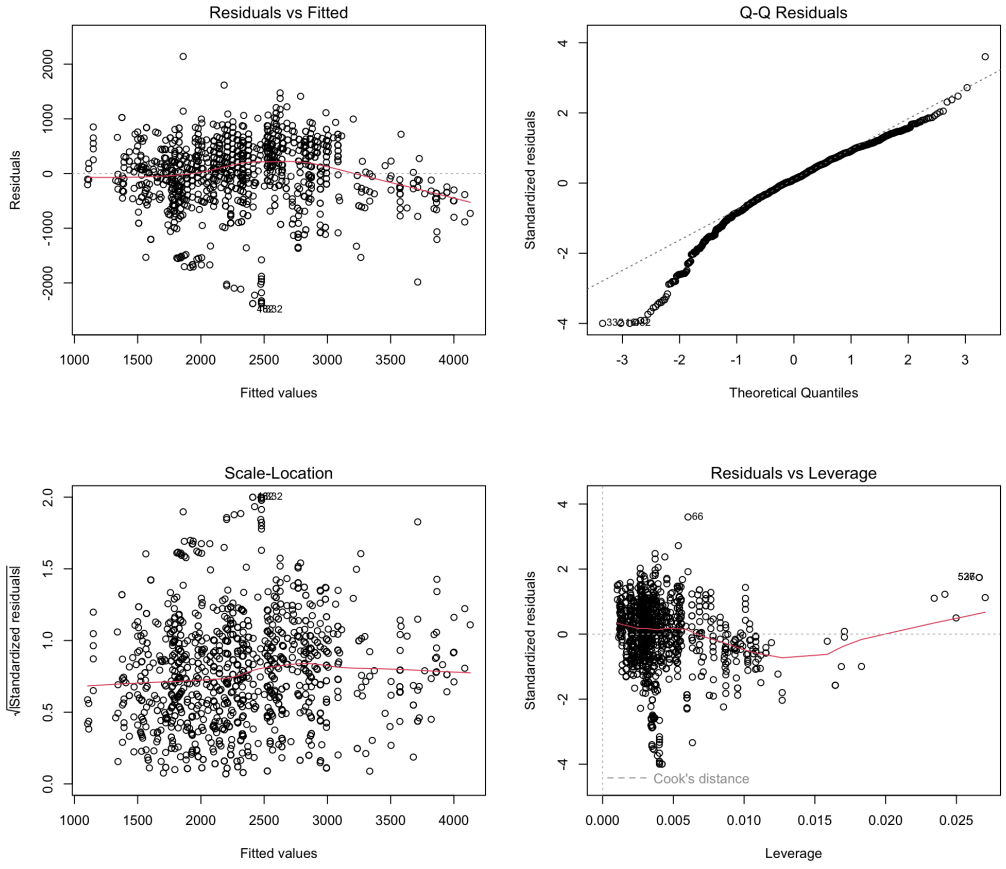


Figure 6: Plots for the Assumptions

Residuals vs. Fitted: This is a scatter plot showing the residuals on the Y-axis and the fitted values on the X-axis, commonly used to detect non-linearity, unequal error variances and outliers. Values that have the residual 0 are those that would end up directly on the estimated regression line. In our graph, the residuals are chaotically and randomly distributed around the 0 line, indicating that the model is suitable for testing.

Normal Q-Q: The normal quantile-quantile visualization calculates the normal quantiles of all values in a column. The values (Y-axis) are then plotted against the normal quantiles (X-axis). The values form a pattern that slightly curves to the right of the normal line, showing that the distribution is somewhat skewed to the left. However, as the majority of the values stay on the line, it is safe to assume that normality is ensured to a certain extent.

Scale - Location: This plot shows if residuals are spread equally along the ranges of predictors. This is how we can check the assumption of equal variance - homoscedasticity. Homoscedasticity is another important assumption, as it indicates that the variability of the residuals is consistent across the range of fitted values. Apply this on our plot, the square roots of residuals are not so equally spread across the red line but instead, condensed more between the fitted values 1500 to 3000 and sparser as they move further from this range. However, this does not imply a clear trend from the values, hence the assumption of homoscedasticity is met.

Residuals vs. Leverage This plot displays the residuals against the leverage values, which measure the influence of each data point on the model. The points are scattered without any clear pattern, suggesting that there are no influential observations that significantly affect the model. Influential observations can have a disproportionate impact on the model, and this plot helps identify any potential outliers or high-leverage points. The absence of a clear pattern indicates that the model is not unduly influenced by any individual data points.

7.2.4 Testing

We first run the model on our **training set** (70% of the original data) to validate the model before actual testing. To gauge the correspondence between our estimates and the actual values, we look at their distribution and compare them.

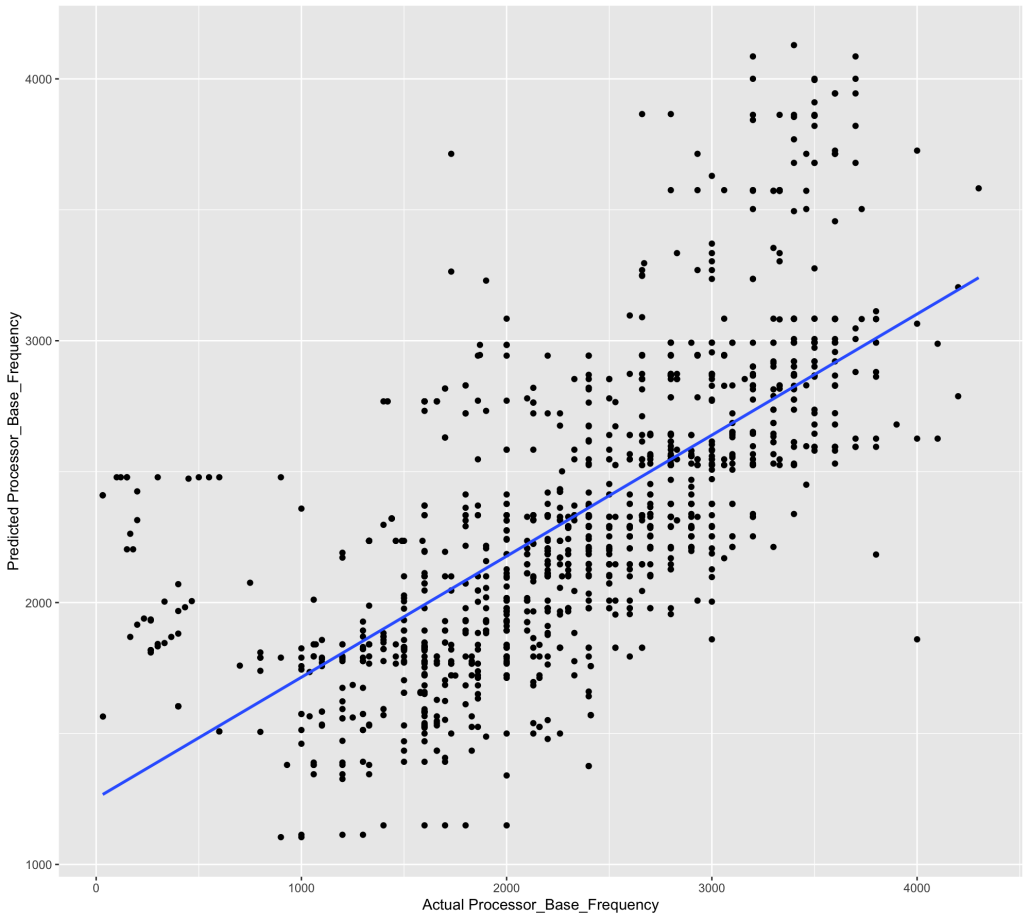


Figure 7: Multi Linear Regression model applied on the Training set

The values plotted on this graph is quite widely distributed from the regression line, proving that the predicted and actual values form an average linear relationship. This confirms that the model is effective.

We now run the model on the preceding **test set** (30% of the original data) to access the performance of our model. Then, we put the predicted values into a new column in the dataset for easy plotting.

```
    predicted_values <- predict(regression_model, test_set)
    test_set["Predicted"] <- predicted_values
```

Then we plot the graph following the exact same way as we did with the training set.

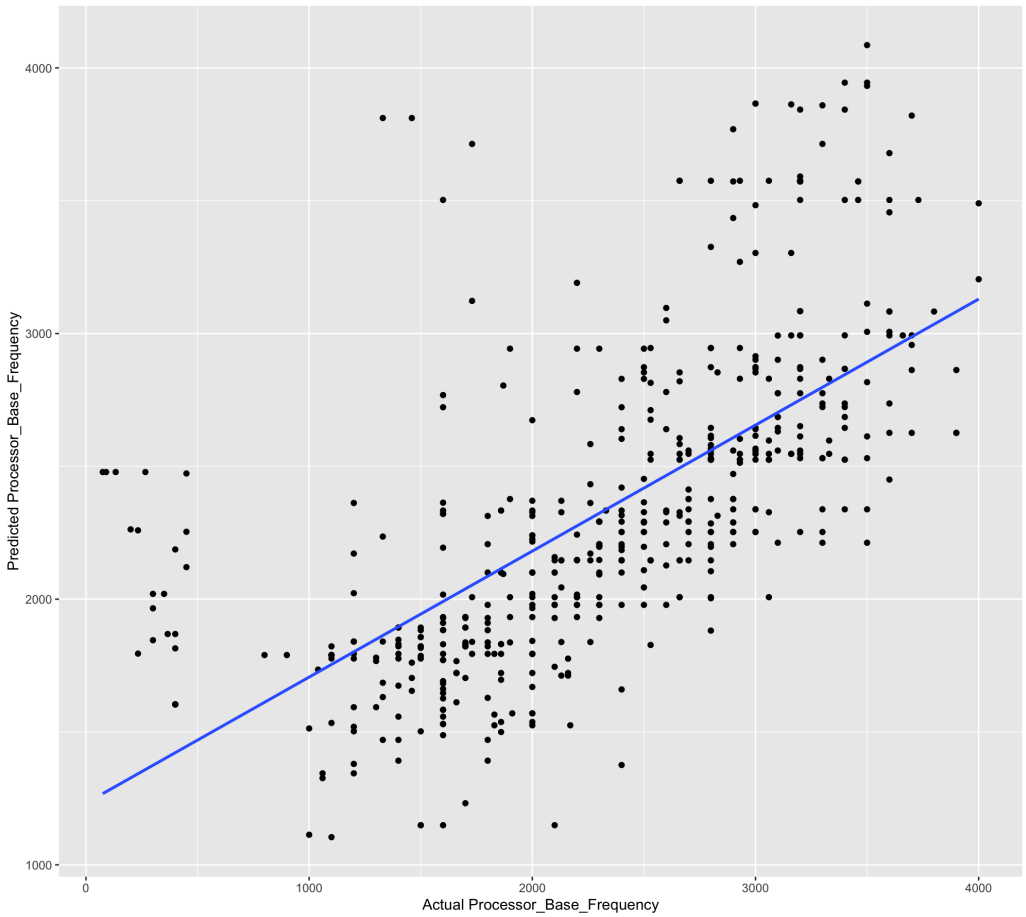


Figure 8: Multi Linear Regression model applied on the Test Set

The graph portrays a linear relationship between the predicted and actual values, indicating that the model works on unseen data. This concludes that this model is good at predicting the clock speed based on the key variables mentioned.

7.3 Conclusion

After implementing a Multi Regression Model to predict the CPU Clock Speed, we were able to identified 4 variables that are significant to the Processor Base Frequency. The model aid manufacturers in pinpointing the factors that affect the CPU performance through its clock speed, providing appropriate strategies in product development, while also help customer choosing the right CPU specifications for their needs. Overall, the results that the model predict is justifiably similar to the actual data.

8 Discussion

The analysis conducted in this study focuses on predicting CPU clock speed based on various CPU specifications. The primary attributes used for this prediction include the number of cores, number of threads, cache size, power consumption, TDP (Thermal Design Power), manufacturing process, and release date. This approach provides insights into how these features impact the clock speed, which is a crucial performance metric for CPUs.

Our results indicate that certain attributes significantly influence CPU clock speed. Specifically, the number of cores, number of threads, and cache size show a strong positive correlation with clock speed. This finding aligns with the understanding that more cores and threads can handle more tasks simultaneously, thus requiring higher clock speeds to maintain performance. Similarly, larger cache sizes facilitate faster data access, contributing to higher clock speeds.

Power consumption and TDP are also critical factors. CPUs with higher power consumption and TDP values generally have higher clock speeds. This is because higher power and thermal limits allow CPUs to operate at higher frequencies, enhancing performance, especially under load. However, this also implies a trade-off between performance and energy efficiency, which is an important consideration for both manufacturers and consumers.

The manufacturing process, indicated by lithography, shows that newer manufacturing technologies (with smaller nanometer values) tend to support higher clock speeds. This is due to advancements in semiconductor technology, allowing for more transistors on a chip, reducing heat output, and improving power efficiency.

The release date serves as a temporal indicator, showing that newer CPUs generally have higher clock speeds. This trend reflects the continuous improvement in CPU technology and performance over time.

The regression analysis used in this study, particularly the linear regression model, effectively captures the relationships between the independent variables and CPU clock speed. The model's performance metrics, such as R-squared and Mean Squared Error (MSE), indicate a good fit, confirming the model's predictive capability. While ANOVA provides additional insights into the impact of categorical variables like CPU series and generation, the regression model is the primary tool for prediction.

In conclusion, this study offers a robust methodology for predicting CPU clock speed based on key specifications. The findings have practical implications for CPU manufacturers aiming to optimize design and performance, as well as for consumers looking to make informed purchasing decisions. By leveraging statistical methods and regression analysis, this research contributes to a deeper understanding of CPU performance dynamics and enhances transparency and efficiency in the marketplace.

This comprehensive approach demonstrates the value of combining various statistical techniques to predict CPU performance accurately. Future research could explore the incorporation

of more complex models, such as machine learning algorithms, to further improve prediction accuracy and handle non-linear relationships among the variables.

9 References

[1] Peter Bruce and Andrew Bruce. *Practical Statistics for Data Scientists*. O'Reilly, 2017.

[2] Nguyễn Tiến Dũng (chủ biên) và Nguyễn Đình Huy. *Xác suất - Thống kê & Phân tích số liệu*. Đại học Quốc gia TP. Hồ Chí Minh, 2019.

[3] finnstats. Equality of Variances in R. https://www.r-bloggers.com/2021/06/equality-of-variances-in-r-homogeneity-test-quick-guide/. Accessed: May 2024.

[4] Github. Hands-On Programming with R. https://rstudio-education.github.io/hopr/basics.html. Accessed: May 2024.

[5] ILISSEK. Computer Parts (CPUs and GPUs). https://www.kaggle.com/datasets/iliassekkaf/computerparts, 2017. Accessed: April 2024.

[6] Intel. What is Clock Speed? https://www.intel.com/content/www/us/en/gaming/resources/cpu-clock-speed.html. Accessed: May 2024.

[7] Alboukadel Kassambara. Comparing Multiple Means in R. https://www.datanovia.com/en/lessons/anova-in-r/. Accessed: May 2024.

[8] Douglas C Montgomery and George C Runger. *Applied Statistics and Probability for Engineers*. Wiley, 6th edition, 2006.

[9] [Học một chút]. Thống kê trong R-Studio. https://youtube.com/playlist?list=PL9XZJEFXvG9E9HJn2n1Gu_TLAsVPc1MkJ&feature=shared, 2021. Accessed: May 2024.

[10] Sheldon M Ross. *Introduction to Probability and Statistics for Engineers and Scientists*. Academic Press, 2020.

[11] Abdullah Al Sefat, G.M. Rasiqul Islam Rasiq, Nafiul Nawjis, and Sk Mehedi. *CPU Performance Prediction Using Various Regression Algorithms*, pages 163–171. 01 2021.

[12] William Stallings. *Computer Organization and Architecture*. Prentice Hall Professional Technical Reference, 6th edition, 2002.

[13] Phan Thị Khánh Vân. Lecture notes on Probability and Statistics, LMS - HCMUT. https://lms.hcmut.edu.vn, 2024. Accessed: May 2024.