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

PROBABILITY AND STATISTICS

Assignment

Computer Parts

Course code	MT2013
Lecturer	Dr. Nguyen Tien Dung
Semester	232
Class	CC02
Group	13

Ho Chi Minh City, April 2024

Member List & Workload

No.	Full name	Student ID	Tasks	Contribution
1	Nguyen Doan Hai Bang	2252078	Background and MLR model	25%
2	Nguyen Quang Duy	2252120	Descriptive statistics	25%
3	Tran Duy Duc Huy	2252263	Data preprocess, ANOVA	25%
4	Huynh Mai Quoc Khang	2252293	Background and MLR Model	25%
5	Tran Anh Khoa	2252362	LATEX, Data preprocess, ANOVA	25%

Lecturer's assessment

No.	Full name	Student ID	Assessment	Score
1	Nguyen Doan Hai Bang	2252078	Requirements analysis	
2	Nguyen Quang Duy	2252120	Requirements analysis	
3	Tran Duy Duc Huy	2252263	Requirements analysis	
4	Huynh Mai Quoc Khang	2252293	Requirements analysis	
5	Tran Anh Khoa	2252362	Requirements analysis	

Contents

List of Figures

1 Dataset introduction

1.1 Dataset description

The dataset on computer parts provides a comprehensive inventory of specifications, release dates, and release prices for Graphics Processing Units (GPUs) and Central Processing Units (CPUs). Within this dataset, various technical parameters and release dates play pivotal roles in shaping the landscape of computer hardware.
Exploring the significance of certain technical specifications, such as clock speeds, maximum temperatures, display resolutions, power draws, and number of threads, can offer valuable insights into the performance capabilities of different computer parts. By analyzing how these parameters vary across different GPUs and CPUs, we can discern trends in technological advancements and the evolution of computing power over time.
This project will use Probability & Statistics's knowledge and related tools to evaluate the factors influencing the release prices of CPUs. Here are some general details of the dataset:

- **Title:** Computer parts
- **Source Information:**
  - Companies: Intel, Game-Debate, companies involved in producing the part.
- **Number of Observations:** 2283
- **Number of Variables:** 45 (Described in section 1.2)

1.2 Variables description

Here's the table describing data type, unit, and description of all 45 variables available in the database `cpus.csv`.

Variable	**Variable Type** (Discrete/ Continuous/ Categorical)	**Unit**	**Description**
Product_Collection	Categorical	None	The product line or group to which the processor belongs
Vertical_Segment	Categorical	None	The target market or industry sector for which the processor is designed
Processor_Number	Categorical	None	A unique identifier assigned to the processor model
Status	Categorical	None	Indicates the current status of the processor (Launched/ End of Interactive Support/ Other)
Launch_Date	Categorical	None	The date when the processor was officially launched or released

Lithography	Continuous	nm	The semiconductor technology used to manufacture an integrated circuit
Recommended_Customer_Price	Continuous		The manufacturer's suggested retail price for the processor
nb_of_Cores	Discrete	None	Number of independent central processing units
nb_of_Threads	Discrete	None	Number of simultaneous threads that the processor can execute
Processor_Base_Frequency	Continuous	GHz	The rate at which the processor's transistors open and close
Max_Turbo_Frequency	Continuous	GHz	The maximum single core frequency at which the processor is capable of operating using Intel® Turbo Boost Technology
Cache	Continuous	MB Smart Cache	Area of fast memory located on the processor
Bus_Speed	Continuous	GT/s OPI	The speed at which data is transferred between the processor and other components
TDP	Continuous	W	The average power the processor dissipates when operating at Base Frequency with all cores
Embedded_Options_Available	Categorical	None	Indicates whether the processor has options for embedded applications
Conflict_Free	Categorical	None	Indicates whether the processor is manufactured using conflict-free materials
Max_Memory_Size	Continuous	GB	The maximum memory capacity supported by the processor
Memory_Types	Categorical	None	Types of memory supported by the processor
Max_nb_of_Memory_Channels	Discrete	None	Number of memory channels refers to the bandwidth operation for real-world application
Max_Memory_ Bandwidth	Continuous	GB/s	The maximum rate at which data can be read from or stored into a semiconductor memory by the processor

ECC_Memory_Supported	Categorical	None	Indicates whether the processor supports Error-Correcting Code (ECC) memory
Processor_Graphics_	Categorical	None	Indicates graphics processing circuitry integrated into the processor
Graphics_Base_Frequency	Continuous	MHz	The rated/guaranteed graphics render clock frequency
Graphics_Max_Dynamic_Frequency	Continuous	GHz	The maximum opportunistic graphics render clock frequency that can be supported using Intel® HD
Graphics_Video_Max_Memory	Continuous	GB	The maximum amount of memory accessible to processor graphics
Graphics_Output	Categorical	None	Types of video outputs supported by the integrated graphics
Support_4k	Categorical	None	Indicates whether the integrated graphics support 4K resolution
Max_Resolution_HDMI	Categorical	None	The maximum resolution supported by the processor via the HDMI interface
Max_Resolution_DP	Categorical	None	The maximum resolution supported by the processor via the DP interface
Max_Resolution_eDP_Integrated_Flat_Panel	Categorical	None	The maximum resolution supported by the processor via the eDP interface
DirectX_Support	Continuous	None	Indicates support for a specific version of DirectX, a Microsoft collection of APIs for handling multimedia
OpenGL_Support	Categorical	None	The version of OpenGL supported by the integrated graphics
PCI_Express_Revision	Discrete	None	The PCIe version supported by the processor
PCI_Express_Configurations_	Categorical	None	Configurations of PCI Express lanes supported by the processor
Max_nb_of_PCI_Express_Lanes	Discrete	None	The maximum number of PCI Express lanes supported by the processor
T	Continuous	°C	The maximum temperature allowed on the chip
Intel_Hyper_Threading_Technology_	Categorical	None	Indicates whether the processor supports Intel® Hyper-Threading Technology

Intel_Virtualization_technology_VTx_	Categorical	None	Indicates whether the processor supports Intel® Virtualization Technology (VT-x)
Intel_64_	Categorical	None	Indicates whether the processor supports 64-bit architecture
Instruction_Set	Categorical	None	The instruction set architecture supported by the processor
Instruction_Set_Extensions	Categorical	None	Extensions to the instruction set architecture supported by the processor
Idle_States	Categorical	None	Power-saving features that allow the processor to enter idle states when not in use
Thermal_Monitoring_Technologies	Categorical	None	Technologies used for monitoring and managing the temperature of the processor
Secure_Key	Categorical	None	Security feature that generates high-quality cryptographic keys
Execute_Disable_Bit	Categorical	None	Security feature that helps prevent certain types of malicious code from executing

2 Background

2.1 Multiple linear regression

2.1.1 Definition

Regression models are used to describe relationships between variables by fitting a line to the observed data. Regression allows you to estimate how a dependent variable changes as the independent variable(s) change.
Multiple linear regression, also known simply as multiple regression, is a statistical technique that uses several explanatory variables to predict the outcome of a response variable. It allows you to estimate how the dependent variable changes as the independent variables change. This type of regression is used when you have more than one explanatory variable and is an extension of simple linear regression, which involves only one independent variable. Multiple linear regression is a more specific calculation than simple linear regression. For straight-forward relationships, simple linear regression may easily capture the relationship between the two variables. However, for more complex relationships requiring more consideration, multiple linear regression is often better. Multiple linear regression is applicable in scenarios where we want to know:

- How strong the relationship is between two or more independent variables and one dependent variable (e.g. how rainfall, temperature, and amount of fertilizer added affect crop growth).

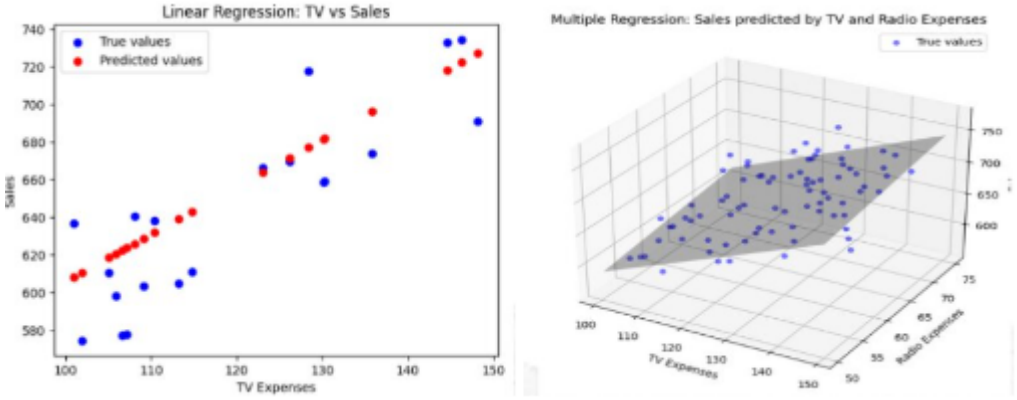

Figure 1: *Single linear regression vs Multiple linear regression*

- The value of the dependent variable at a certain value of the independent variables (e.g. the expected yield of a crop at certain levels of rainfall, temperature, and fertilizer addition).

2.1.2 Assumption

The multiple regression model is based on the following assumptions:

1. Homogeneity of variance: The variance of the errors is constant across all the values of the independent variable.
2. Independence of observations: the observations in the dataset were collected using statistically valid sampling methods, and there are no hidden relationships among variables.

In multiple linear regression, it is possible that some of the independent variables are actually correlated with one another, so it is important to check these before developing the regression model. If two independent variables are too highly correlated (r2 $> \sim 0.6$), then only one of them should be used in the regression model.

3. Normality: The data follows a normal distribution with a mean of 0 and variance σ.
4. Linearity: The relationship between the independent variables and the dependent variable is linear.

2.1.3 Formula

The formula for simple linear regression is: $Y = C_0 + C_1 X + \varepsilon$
Where:

- Y: the dependent variable
- X: the independent variable
- C_0: the y-intercept (constant term)
- $C_1 X$: the slope of line
- ϵ: the model's error term

The formula for multiple linear regression is: $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \varepsilon$
Where:

- y: the dependent variable
- β_0: the y-intercept (constant term)
- $\beta_1, \beta_2, \ldots, \beta_n$: the regression coefficients for each independent variable
- $x_1, x_2, \ldots, x_n$: the independent variables
- ϵ: the model's error term

2.2 Analysis of Variance (ANOVA)

2.2.1 Definition

ANOVA is a statistical method used to test differences between two or more means and determine if there is a significant difference between them. It is similar to the t-test, but the t-test is generally used for comparing two means, while ANOVA is used when you have more than two means to compare.
The ANOVA test allows a comparison of more than two groups at the same time to determine whether a relationship exists between them. The result of the ANOVA formula, the F statistic (also called the F-ratio), allows for the analysis of multiple groups of data to determine the variability between samples and within samples. If no real difference exists between the tested groups, which is called the null hypothesis, the result of the ANOVA's F-ratio statistic will be close to 1.

2.2.2 One-Way vs. Two-Way ANOVA

The two most common types of ANOVAs are the one-way ANOVA and the two-way ANOVA. One-way or two-way refers to the number of independent variables in your analysis of variance tests.
A one-way ANOVA evaluates the impact of a sole factor on a sole response variable. It determines whether all the samples are the same. The one-way ANOVA is used to determine whether there are any statistically significant differences between the means of three or more independent (unrelated) groups.

Figure 2: *One-way ANOVA*

A two-way ANOVA is an extension of the one-way ANOVA. With a one-way, you have one independent variable affecting a dependent variable. With a two-way ANOVA, there are two independents. For example, a two-way ANOVA allows a company to compare worker productivity based on two independent variables, such as salary and skill set. It is utilized to

observe the interaction between the two factors and tests the effect of two factors at the same time.

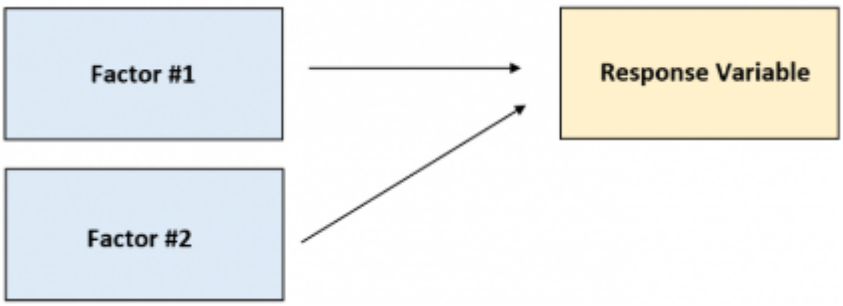

Figure 3: *Two-way ANOVA*

2.2.3 Assumption

There are several assumptions that must be met in order for ANOVA to be valid:

1. Independence of observations: This means that the observations are not related to each other in any way.
2. Normality: The data within each group should be normally distributed.
3. Equal variances: The variances of the groups being compared should be approximately equal.

2.2.4 Formula

Table 2: *ANOVA Table*

Source of Variation	Sum of Squares	Degree of Freedom	Mean Squares	F Value
Between Groups	SSB = $\sum n_j(X_j - \bar{X})^2$	$df_1 = k - 1$	MSB = $\frac{SSB}{k-1}$	$F = \frac{MSB}{MSE}$
Error	SSE = $\sum\sum(X - X_j)^2$	$df_2 = N - k$	MSE = $\frac{SSE}{N-k}$	
Total	SST = SSB + SSE	$df_3 = N - 1$		

2.3 Comparing Multiple linear regression model with ANOVA

Multiple linear regression and ANOVA are both statistical modeling techniques used to analyze the relationship between a dependent variable and one or more independent variables. However, they differ in their assumptions, applications, and the way they are presented.
Multiple linear regression is a generalization of simple linear regression, which models the relationship between a dependent variable and a single independent variable. Multiple linear regression extends this concept by allowing for multiple independent variables. It is used when there is a continuous dependent variable and one or more independent variables, which can

be either continuous or categorical. The model estimates the coefficients of the independent variables, which represent the change in the dependent variable for a one-unit change in the independent variable, while controlling for the effects of other independent variables.
On the other hand, ANOVA (Analysis of Variance) is a statistical technique used to compare the means of a dependent variable across two or more groups defined by one or more categorical independent variables. ANOVA tests the null hypothesis that all the group means are equal against the alternative hypothesis that at least one group mean is different. ANOVA partitions the total variance of the dependent variable into variance due to the independent variable and variance due to random error.
In summary, multiple linear regression and ANOVA are both useful statistical modeling techniques, but they are used in different contexts and for different purposes. Multiple linear regression is used to model the relationship between a continuous dependent variable and multiple independent variables, while ANOVA is used to compare the means of a continuous dependent variable across two or more groups defined by one or more categorical independent variables.

3 Hypothesis testing

3.1 Definition

Hypothesis testing can be defined as a statistical tool that is used to identify if the results of an experiment are meaningful or not. It involves setting up a null hypothesis and an alternative hypothesis. These two hypotheses will always be mutually exclusive. This means that if the null hypothesis is true then the alternative hypothesis is false and if the alternative hypothesis is true, the null hypothesis is necessarily false.
The null hypothesis (H_0) is a concise mathematical statement that is used to indicate that there is no difference between two possibilities. In other words, there is no difference between certain characteristics of data. This hypothesis assumes that the outcomes of an experiment are based on chance alone. Hypothesis testing is used to conclude if the null hypothesis can be rejected or not.
The alternative hypothesis (H_1)) is an alternative to the null hypothesis. It is used to show that the observations of an experiment are due to some real effect. and indicates that there is a statistical significance between two possible outcomes.
In hypothesis testing, the p-value is used to indicate whether the results obtained after conducting a test are statistically significant or not. It also indicates the probability of making an error in rejecting or not rejecting the null hypothesis. This value is always a number between 0 and 1. The p-value is compared to an alpha level or significance level. The alpha level can be defined as the acceptable risk of incorrectly rejecting the null hypothesis. The alpha level is usually chosen between 1% to 5%.

3.2 One Tailed and Two Tailed Hypothesis Testing

One tailed hypothesis testing is done when the rejection region is only in one direction. It can also be known as directional hypothesis testing because the effects can be tested in one direction only. Two tailed hypothesis testing is done when the critical region lies on both sides of the sampling distribution. It is also known as a non - directional hypothesis testing method.

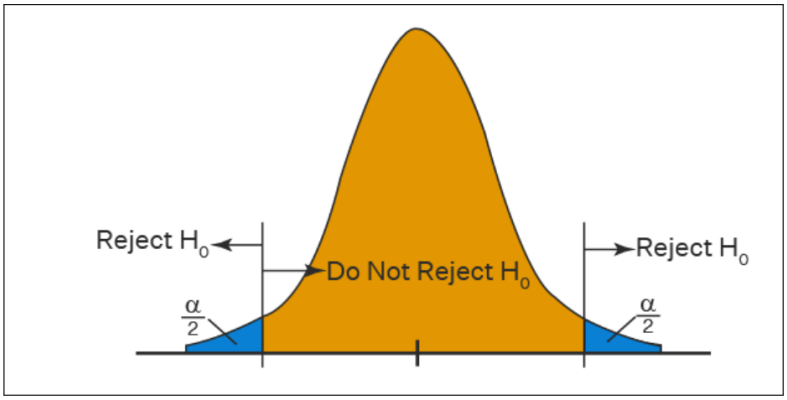

Figure 4: *Right tailed hypothesis testing*

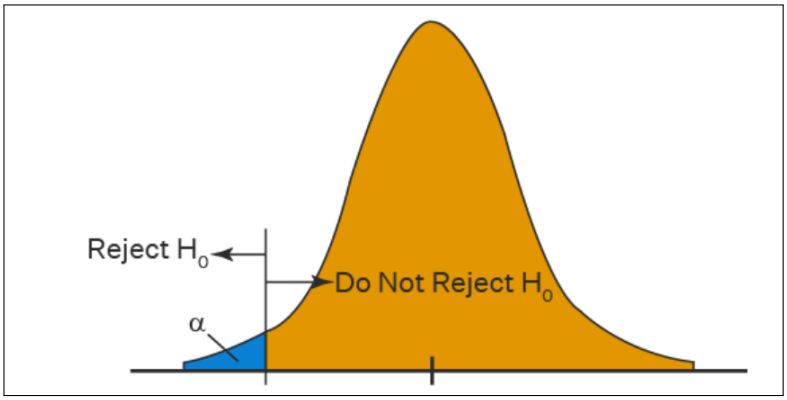

Figure 5: *Left tailed hypothesis testing*

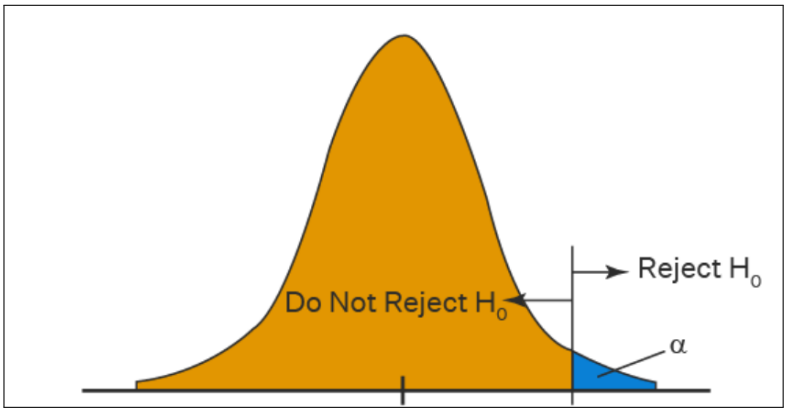

Figure 6: *Two tailed hypothesis testing*

4 Data pre-processing

4.1 Data reading

We need to install necessary libraries first:

```
# Library
# install.packages("ggplot2")
# install.packages("corrplot")
# install.packages("RColorBrewer")
# install.packages("DescTools")
library(ggplot2)
library(corrplot)
library(RColorBrewer)
library(DescTools)
```

Import Data: Use the read.csv function to import `Intel_CPUs.csv` data file into RStudio.
Display Data: Utilize R Markdown to create a structured document.
Knit to HTML: After setting up R Markdown document, Knit it to HTML. This process will execute the R code chunks, including importing and displaying the data, and generate an HTML file.

```
#Read CPUs data
Intel_CPUs = read.csv("D:/BTL_XSTK_HK232/Intel_CPUs.csv",na.strings = c("",
    "N/A"))
head(Intel_CPUs,6)
```

Here, the command `head(CPUs_DATA, 6)` is used to display the first 6 rows of data stored in the object `CPUs_DATA`.

```
##                         Product_Collection Vertical_Segment Processor_Number
## 1 7th Generation Intel® Core™ i7 Processors           Mobile          i7-7Y75
## 2 8th Generation Intel® Core™ i5 Processors           Mobile         i5-8250U
## 3 8th Generation Intel® Core™ i7 Processors           Mobile         i7-8550U
## 4          Intel® Core™ X-series Processors          Desktop          i7-3820
## 5 7th Generation Intel® Core™ i5 Processors           Mobile          i5-7Y57
## 6     Intel® Celeron® Processor 3000 Series           Mobile            3205U
##        Status Launch_Date Lithography Recommended_Customer_Price nb_of_Cores
## 1    Launched       Q3'16       14 nm                   $393.00            2
## 2    Launched       Q3'17       14 nm                   $297.00            4
## 3    Launched       Q3'17       14 nm                   $409.00            4
## 4 End of Life       Q1'12       32 nm                   $305.00            4
## 5    Launched       Q1'17       14 nm                   $281.00            2
## 6    Launched       Q1'15       14 nm                   $107.00            2
##   nb_of_Threads Processor_Base_Frequency Max_Turbo_Frequency            Cache
## 1             4                1.30 GHz            3.60 GHz  4 MB SmartCache
## 2             8                1.60 GHz            3.40 GHz  6 MB SmartCache
## 3             8                1.80 GHz            4.00 GHz  8 MB SmartCache
## 4             8                3.60 GHz            3.80 GHz 10 MB SmartCache
## 5             4                1.20 GHz            3.30 GHz  4 MB SmartCache
## 6             2                1.50 GHz                              2 MB
##     Bus_Speed   TDP Embedded_Options_Available Conflict_Free Max_Memory_Size
## 1  4 GT/s OPI 4.5 W                        No           Yes           16 GB
## 2  4 GT/s OPI  15 W                        No           Yes           32 GB
## 3  4 GT/s OPI  15 W                        No           Yes           32 GB
## 4 5 GT/s DMI2 130 W                        No                      64.23 GB
## 5  4 GT/s OPI 4.5 W                        No           Yes           16 GB
## 6 5 GT/s DMI2  15 W                        No           Yes           16 GB
##                      Memory_Types Max_nb_of_Memory_Channels
## 1          LPDDR3-1866, DDR3L-1600                        2
## 2           DDR4-2400, LPDDR3-2133                        2
## 3           DDR4-2400, LPDDR3-2133                        2
## 4              DDR3 1066/1333/1600                        4
## 5          LPDDR3-1866, DDR3L-1600                        2
## 6 DDR3L 1333/1600 LPDDR3 1333/1600                        2
```

Figure 7: *Data table [1]*

```
##   Max_Memory_Bandwidth ECC_Memory_Supported Processor_Graphics_
## 1            29.8 GB/s                   No                  NA
## 2            34.1 GB/s                   No                  NA
## 3            34.1 GB/s                   No                  NA
## 4            51.2 GB/s                   No                  NA
## 5            29.8 GB/s                   No                  NA
## 6            25.6 GB/s                                       NA
##   Graphics_Base_Frequency Graphics_Max_Dynamic_Frequency
## 1                 300 MHz                       1.05 GHz
## 2                 300 MHz                       1.10 GHz
## 3                 300 MHz                       1.15 GHz
## 4
## 5                 300 MHz                        950 MHz
## 6                 100 MHz                        800 MHz
##   Graphics_Video_Max_Memory Graphics_Output Support_4k Max_Resolution_HDMI
## 1                   16 GB eDP/DP/HDMI/DVI         NA      4096x2304@24Hz
## 2                   32 GB eDP/DP/HDMI/DVI         NA      4096x2304@24Hz
## 3                   32 GB eDP/DP/HDMI/DVI         NA      4096x2304@24Hz
## 4                                                 NA
## 5                   16 GB eDP/DP/HDMI/DVI         NA      4096x2304@24Hz
## 6                             eDP/DP/HDMI         NA
##   Max_Resolution_DP Max_Resolution_eDP_Integrated_Flat_Panel DirectX_Support
## 1    3840x2160@60Hz                          3840x2160@60Hz              12
## 2    4096x2304@60Hz                          4096x2304@60Hz              12
## 3    4096x2304@60Hz                          4096x2304@60Hz              12
## 4
## 5    3840x2160@60Hz                          3840x2160@60Hz              12
## 6                                                                    11.2/12
```

Figure 8: *Data table [2]*

```
##   OpenGL_Support PCI_Express_Revision PCI_Express_Configurations_
## 1             NA                    3   1x4, 2x2, 1x2+2x1 and 4x1
## 2             NA                    3   1x4, 2x2, 1x2+2x1 and 4x1
## 3             NA                    3   1x4, 2x2, 1x2+2x1 and 4x1
## 4             NA                    2
## 5             NA                    3   1x4, 2x2, 1x2+2x1 and 4x1
## 6             NA                    2                     4x1 2x4
##   Max_nb_of_PCI_Express_Lanes      T Intel_Hyper_Threading_Technology_
## 1                          10  100°C                               Yes
## 2                          12  100°C                               Yes
## 3                          12  100°C                               Yes
## 4                          40 66.8°C                               Yes
## 5                          10  100°C                               Yes
## 6                          12  105°C                                No
##   Intel_Virtualization_Technology_VTx_ Intel_64_ Instruction_Set
## 1                                  Yes       Yes          64-bit
## 2                                  Yes       Yes          64-bit
## 3                                  Yes       Yes          64-bit
## 4                                  Yes       Yes          64-bit
## 5                                  Yes       Yes          64-bit
## 6                                  Yes       Yes          64-bit
##   Instruction_Set_Extensions Idle_States Thermal_Monitoring_Technologies
## 1       SSE4.1/4.2, AVX 2.0         Yes                              Yes
## 2       SSE4.1/4.2, AVX 2.0         Yes                              Yes
## 3       SSE4.1/4.2, AVX 2.0         Yes                              Yes
## 4          SSE4.2, AVX, AES         Yes                              Yes
## 5       SSE4.1/4.2, AVX 2.0         Yes                              Yes
## 6                SSE4.1/4.2         Yes                              Yes
##   Secure_Key Execute_Disable_Bit
## 1        Yes                 Yes
## 2        Yes                 Yes
## 3        Yes                 Yes
## 4                            Yes
## 5        Yes                 Yes
## 6        Yes                 Yes
```

Figure 9: *Data table [3]*

We select specific columns from the `Intel_CPUs` dataset, presumably for further analysis.

```
# The selected columns include information about the vertical segment, status, lithography, recommended customer price, number of cores, number of threads, processor base frequency, TDP (Thermal Design Power), max memory size.
CPUs_data = Intel_CPUs[,c("Vertical_Segment","Status","Lithography"
,"Recommended_Customer_Price","nb_of_Cores","nb_of_Threads"
,"Processor_Base_Frequency","TDP","Max_Memory_Size")]
```

4.2 Checking missing values

There exists N/A values in this table. To determine the number of N/A values separately, we do as followings:

```
# Calculate the total number of missing values for each column in the CPUs_data dataset using the apply() function
print(apply(is.na(CPUs_data),2,sum))
```

```
##           Vertical_Segment                     Status
##                          0                          0
##                 Lithography Recommended_Customer_Price
##                         71                        982
##                nb_of_Cores               nb_of_Threads
##                          0                        856
##   Processor_Base_Frequency                        TDP
##                         18                         67
##            Max_Memory_Size
##                        880
```

Figure 10: *Number of missing values of chosen variables*

CONCLUSION:

- Vertical_Segment has 0 missing value.
- Status has 0 missing values.
- Lithography has 71 missing values.
- Recommended_Customer_Price has 982 missing values.
- nb_of_Cores has 0 missing values.
- nb_of_Threads has 856 missing values.
- Processor_Base_Frequency has 18 missing values.
- TDP has 67 missing values.
- Max_Memory_Size has 880 missing values.

To fill these missing values in each column, we replace with the corresponding mean or median value of each variable. Before that, as there are variables with units, we should remove these units for afterwards calculations.

```
# Process lithography
CPUs_data$Lithography <- as.double(gsub(" nm$", "", CPUs_data$Lithography))
median_Lithography <- median(CPUs_data$Lithography, na.rm = TRUE)
CPUs_data$Lithography[is.na(CPUs_data$Lithography)] <- median_Lithography

# Process Recommended Customer Price
recommend_price <- function(price_range) {
    if(grepl('-', price_range)) {
        range <- strsplit(price_range, "-")[[1]]
        return((as.double(range[1]) + as.double(range[2])) / 2)
    }
    return (price_range)
}
CPUs_data$Recommended_Customer_Price <- gsub("\\$", "",
    CPUs_data$Recommended_Customer_Price)
CPUs_data$Recommended_Customer_Price <- sapply(
    CPUs_data$Recommended_Customer_Price, recommend_price)
CPUs_data$Recommended_Customer_Price <- as.double(
    CPUs_data$Recommended_Customer_Price)
median_Recommended_Customer_Price <- median(
    CPUs_data$Recommended_Customer_Price, na.rm = TRUE)
CPUs_data$Recommended_Customer_Price[is.na(
    CPUs_data$Recommended_Customer_Price)] <-
    median_Recommended_Customer_Price

# Process number of threads
median_nb_of_Threads <- median(CPUs_data$nb_of_Threads, na.rm = TRUE)
CPUs_data$nb_of_Threads[is.na(CPUs_data$nb_of_Threads)] <-
    median_nb_of_Threads

# Process base frequency
base_frequency <- function(f) {
    if (grepl(' GHz', f)) {
        return (as.double(gsub(" GHz", "", f)) * 1000)
    }
    return (as.double(gsub(" MHz", "", f)))
}
CPUs_data$Processor_Base_Frequency <- as.integer(sapply(
    CPUs_data$Processor_Base_Frequency, base_frequency))
mean_Processor_Base_Frequency <- mean(CPUs_data$Processor_Base_Frequency, na
    .rm = TRUE)
CPUs_data$Processor_Base_Frequency[is.na(CPUs_data$Processor_Base_Frequency)
    ] <- mean_Processor_Base_Frequency

# Process TDP
CPUs_data$TDP <- as.double(gsub(" W", "", CPUs_data$TDP))
```

```
median_TDP <- median(CPUs_data$TDP, na.rm = TRUE)
CPUs_data$TDP[is.na(CPUs_data$TDP)] <- median_TDP

# Process max memory size
max_mem_size_clean <- function(size) {
    if(grepl('G', size)) {
        return (as.double(gsub(" GB", "", size)))
    }
    return (as.double(gsub(" TB", "", size)) * 1024)
}
CPUs_data$Max_Memory_Size <- sapply(CPUs_data$Max_Memory_Size,
    max_mem_size_clean)
median_Max_Memory_Size <- median(CPUs_data$Max_Memory_Size, na.rm = TRUE)
CPUs_data$Max_Memory_Size[is.na(CPUs_data$Max_Memory_Size)] <-
    median_Max_Memory_Size
```

5 Descriptive statistics

Descriptive statistics help to describe and summarize the dataset, providing insights into its central tendency, variability, distribution, and other relevant properties. This is an important step, since it gives a visual view about the sample data.

5.1 Data statistics

At first, the dataset is categorized into two main types of variables: numerical variables and categorical variables.
This classification helps to identify the nature and characteristics of different variables in the dataset. The numerical variables include "Lithography", "Recommended_Customer_Price", "nb_of_Cores", "nb_ of_Threads", "TDP", "Processor_Base_Frequency"and "Max_Memory _Size". The categorical variable includes "Vertical_Segment" and "Status".
For numerical variables, we calculate the descriptive statistical values including mean, standard deviation, median, first quantile, third quantile, minimum value, and maximum value using built-in RStudio functions.
Input 1:

```
# Numerical variables
numerical_cols = c("Lithography","Recommended_Customer_Price"
,"nb_of_Cores","nb_of_Threads","TDP",
"Processor_Base_Frequency", "Max_Memory_Size")
# Descriptive statistical table
summary_numeric_table <- data.frame(
Staticstic=c("Mean", "Sd", "Median",
"First Quantile", "Third Quantile","Min", "Max")
)
for (i in numerical_cols){
    mean<- mean(CPUs_data[[i]])
    sd <- sd(CPUs_data[[i]])
```

```
        median <- median(CPUs_data[[i]])
        first_quantile <- quantile(CPUs_data[[i]], probs = 0.25)
        third_quantile <- quantile(CPUs_data[[i]], probs = 0.75)
        min <- min(CPUs_data[[i]])
        max <- max(CPUs_data[[i]])
        summary_numeric_table[[i]] <- c(mean, sd, median,
        first_quantile, third_quantile, min, max)
    }
    colnames(summary_numeric_table)[-1] <- numerical_cols
```

Output1:

Staticstic	Lithography	Recommended_Customer_Price	nb_of_Cores	nb_of_Threads	TDP	Processor_Base_Frequency	Max_Memory_Size
Mean	48.45773	311.1804	4.066579	6.955322	59.85291	2222.6176	179.7234
Sd	44.67060	368.4393	6.329884	7.573578	44.22770	822.5345	457.4950
Median	32.00000	255.5000	2.000000	4.000000	47.00000	2260.0000	32.0000
First Quantile	22.00000	255.5000	1.000000	4.000000	26.80000	1660.0000	32.0000
Third Quantile	65.00000	255.5000	4.000000	8.000000	84.00000	2800.0000	32.0000
Min	14.00000	2.5400	1.000000	1.000000	0.02500	32.0000	1.0000
Max	250.00000	7408.0000	72.000000	56.000000	300.00000	4300.0000	4198.4000

Figure 11: *Statistical values of the numerical variables*

For categorical variable, we create a table to summarize the count, unique types, mode and frequency of the mode. By summarizing these statistics in a table, we can gain a better understanding about the distribution and characteristics of the categorical variable.

Input 2

```
    # Categorical variables
    categorical_cols = c("Vertical_Segment", "Status")
    summary_categorical_table <- data.frame(
    Staticstic = c("Count","Unique","Mode","Freq")
    )
    for (i in categorical_cols) {
        count <- length(CPUs_data[[i]])
        unique <- length(unique(CPUs_data[[i]]))
        mode <- Mode(CPUs_data[[i]])
        freq <- attr(mode,"freq")
        summary_categorical_table <-
        cbind(summary_categorical_table,new_col = c(count, unique, mode, freq))
    }
    colnames(summary_categorical_table) <- c("", categorical_cols)
```

	Vertical_Segment	Status
Count	2283	2283
Unique	4	4
Mode	Mobile	Launched
Freq	760	1043

Figure 12: *Categorical variables table*

5.2 Data visualization

In this section, we will explore the visual representations of the dataset to get a comprehensive and intuitive understanding of the data. Through various graphs and plots, this section aims to reveal patterns, trends, distributions and also relationships in the data.

5.2.1 Distribution of Numerical Variables

We employed the ggplot2 package to generate our visualization. Firstly, we show the distributions of all the numerical data using density histograms.

Input 1

```
# Density Histogram Lithography
ggplot(CPUs_data, aes(x = Lithography)) +
geom_histogram(aes(y = ..density..), binwidth = 13, color = "black", fill = "white") +
geom_density(colour = "black",fill = 4, alpha = 0.2, bw = 7) +
scale_x_continuous(breaks = seq(0, 250, by = 50)) +
scale_y_continuous(breaks = seq(0, 0.05, by = 0.01)) +
labs(x = "Lithography", y = "Density", title = "Density Histogram of Lithography") +
theme(plot.title = element_text(hjust = 0.5))
# Density Histogram Recommended_Customer_Price
ggplot(CPUs_data, aes(x = Recommended_Customer_Price)) +
geom_histogram(aes(y = ..density..), binwidth = 500, color = "black", fill = "white") +
geom_density(colour = "black",fill = 4, alpha = 0.2, bw = 250) +
scale_x_continuous(breaks = seq(0, 8000, by = 2000)) +
labs(x = "Recommended_Customer_Price", y = "Density",
title = "Density Histogram of Recommended Customer Price") +
theme(plot.title = element_text(hjust = 0.5))
# Density Histogram nb_of_Cores
ggplot(CPUs_data, aes(x = nb_of_Cores)) +
geom_histogram(aes(y = ..density..), binwidth = 5, color = "black", fill = "white") +
```

```
    geom_density(colour = "black",fill = 4, alpha = 0.2, bw = 3) +
    scale_x_continuous(breaks = seq(0, 80, by = 10)) +
    labs(x = "Number of Cores", y = "Frequency",
    title = "Density Histogram of Number of Cores") +
    theme(plot.title = element_text(hjust = 0.5))
    # Density Histogram nb_of_Threads
    ggplot(CPUs_data, aes(x = nb_of_Threads)) +
    geom_histogram(aes(y = ..density..), binwidth = 4, color = "black", fill = "
        white") +
    geom_density(colour = "black",fill = 4, alpha = 0.2, bw = 2) +
    scale_x_continuous(breaks = seq(0, 140, by = 20)) +
    labs(x = "Number of Threads", y = "Density",
    title = "Density Histogram of Number of Threads") +
    theme(plot.title = element_text(hjust = 0.5))
    # Density Histogram Processor_Base_Frequency
    ggplot(CPUs_data, aes(x = Processor_Base_Frequency)) +
    geom_histogram(aes(y = ..density..), binwidth = 400, color = "black", fill =
         "white") +
    geom_density(colour = "black",fill = 4, alpha = 0.2, bw = 200) +
    scale_x_continuous(breaks = seq(0, 4300, by = 1000)) +
    labs(x = "Processor Base Frequency", y = "Density",
    title = "Density Histogram of Processor Base Frequency") +
    theme(plot.title = element_text(hjust = 0.5))
    # Density Histogram TDP
    ggplot(CPUs_data, aes(x = TDP)) +
    geom_histogram(aes(y = ..density..), binwidth = 20, color = "black", fill =
        "white") +
    geom_density(colour = "black",fill = 4, alpha = 0.2, bw = 10) +
    scale_x_continuous(breaks = seq(0, 300, by = 50)) +
    labs(x = "TDP", y = "Density", title = "Density Histogram of TDP") +
    theme(plot.title = element_text(hjust = 0.5))
    # Density Histogram Max_Memory_Size
    ggplot(CPUs_data, aes(x = Max_Memory_Size)) +
    geom_histogram(aes(y = ..density..), binwidth = 400, color = "black", fill =
         "white") +
    geom_density(colour = "black",fill = 4, alpha = 0.2, bw = 200) +
    scale_x_continuous(breaks = seq(0, 4200, by = 1000)) +
    labs(x = "Max Memory Size", y = "Density",
    title = "Density Histogram of Max Memory Size") +
    theme(plot.title = element_text(hjust = 0.5))
```

Output 1

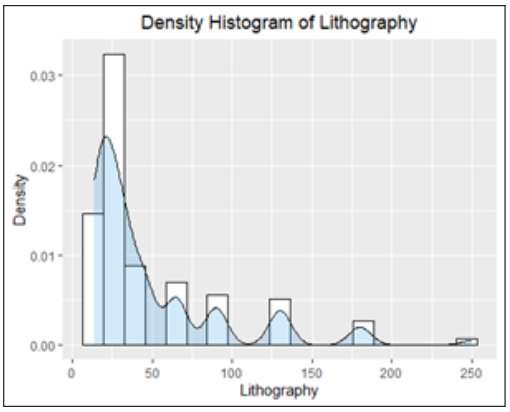

Figure 13: *Density Histogram of Lithography*

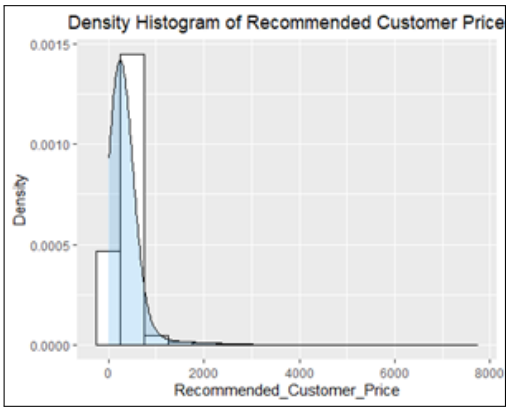

Figure 14: *Density Histogram of Recommended Customer Price*

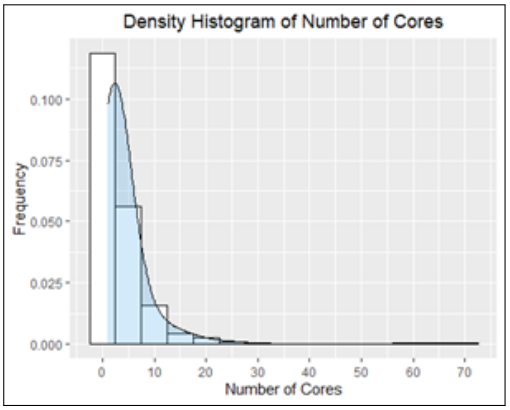

Figure 15: *Density Histogram of Number of Cores*

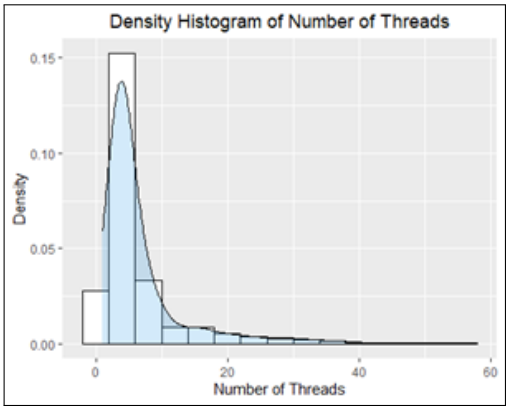

Figure 16: *Density Histogram of Number of Threads*

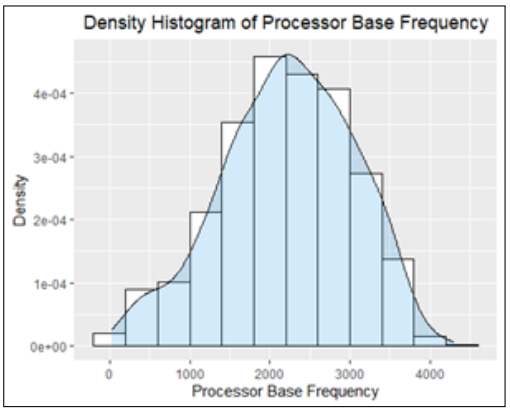

Figure 17: *Density Histogram of Processor Base Frequency*

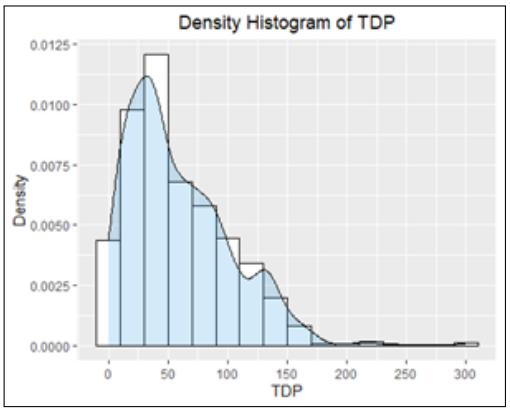

Figure 18: *Density Histogram of TDP*

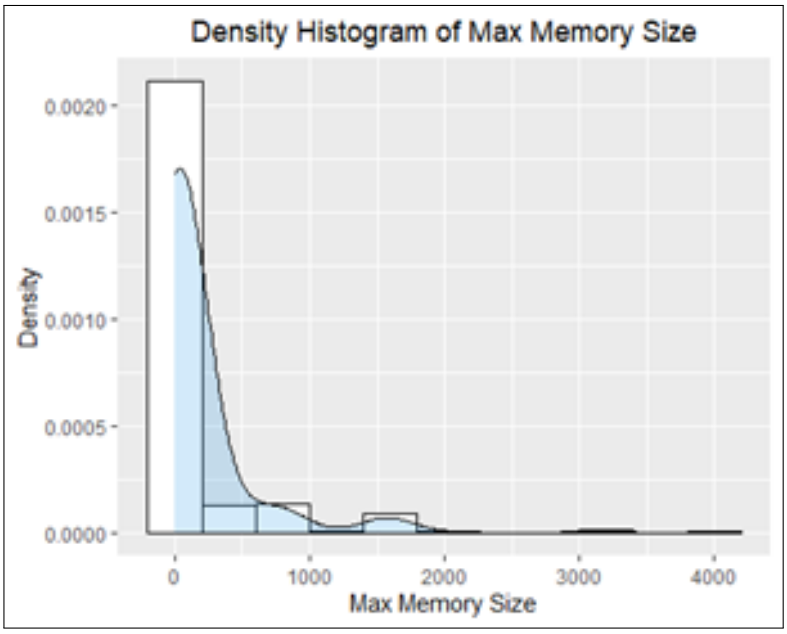

Figure 19: *Density Histogram of Max Memory Size*

Comment:

When examining the seven histograms, it is evident that the majority of them display right-skewed distributions, except for the Figure 17. The peaks of these histograms are located towards the left side, suggesting that the tail of the distribution extends towards the higher values, while the majority of the values are concentrated towards the lower end.
This makes the use of Median Imputation for these variables appropriate as it addresses the skewness while preserving the overall balance of the distribution.
As for the density histogram of Processor Base Frequency, the histogram appears to be almost symmetrical. This suggests that the distribution of the variable is roughly balanced.
Mean Imputation would be a suitable method to handle missing data for this variable.

5.2.2 Visualization of Categorical Variable

Input 1

```
# Bar plot Vertical_Segment
Vertical_Segment_table <- data.frame(table(CPUs_data$Vertical_Segment))
colnames(Vertical_Segment_table) <- c("Vertical_Segment", "Frequency")

ggplot(summary_categorical_table,
aes(x = Vertical_Segment, y = Frequency, fill = Vertical_Segment)) +
geom_bar(stat = "identity", color = "black") +
labs(x = "Vertical Segment", y = "Count",
title = "Bar Chart of Vertical Segment") +
theme(plot.title = element_text(hjust = 0.5)) +
theme(legend.position="none")
```

Output 1

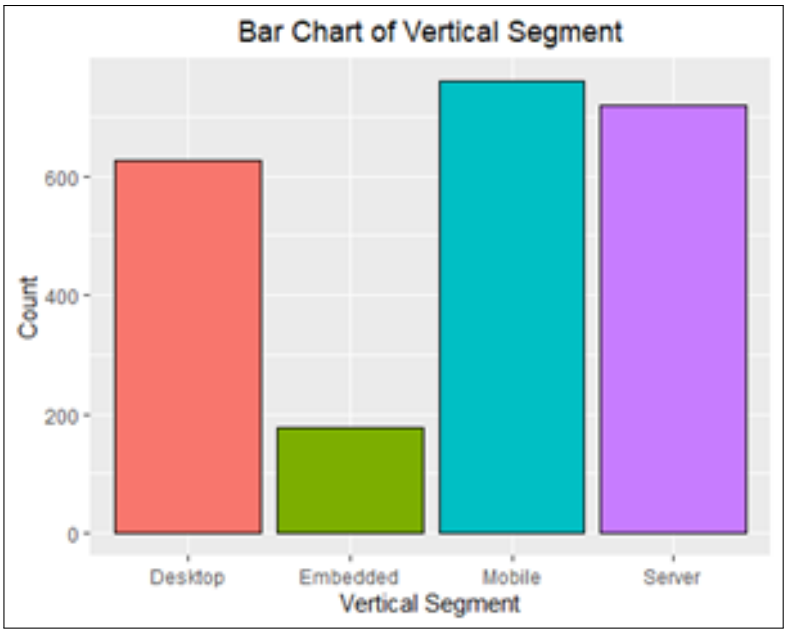

Figure 20: *Bar Chart of Vertical Segment*

It is evident from the bar chart that the Mobile bar is the tallest, suggesting that the majority of CPUs in the dataset is used for mobile. On the other hand, the Embedded bar is the shortest, indicating that this category has the lowest popularity.

Input 2

```
# Bar plot Status
Status_table <- data.frame(table(CPUs_data$Status))
colnames(Status_table) <- c("Status", "Frequency")

ggplot(Status_table,
aes(x = Status, y = Frequency, fill = Status)) +
geom_bar(stat = "identity", color = "black") +
labs(x = "Status", y = "Count",
title = "Bar Chart of Status") +
theme(plot.title = element_text(hjust = 0.5)) +
theme(legend.position="none")
```

Output 2

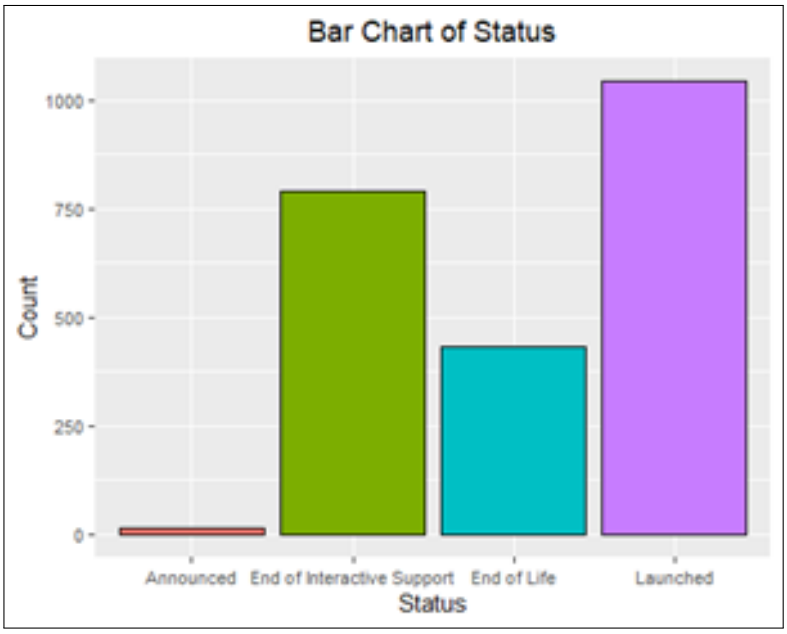

Figure 21: *Bar Chart of Status*

Based on the bar chart, the launched CPUs have the highest count among the four types mentioned. Conversely, the number of annouced CPUs is relatively low.

5.2.3 Relationship between Independent Variables and Dependent Variable

Input 1

```
# Scatter plot between Recommended_Customer_Price and Lithography
ggplot(CPUs_data, aes(x = Lithography, y = Recommended_Customer_Price)) +
geom_point() +
geom_smooth(method = "lm") +
labs(x = "Lithography", y = "Recommended Customer Price",
title = "Scatterplot of Recommended Customer Price vs Lithography") +
theme(plot.title = element_text(hjust = 0.5, size = 13))
# Scatter plot between Recommended_Customer_Price and nb_of_Cores
ggplot(CPUs_data, aes(x = nb_of_Cores, y = Recommended_Customer_Price)) +
geom_point() +
geom_smooth(method = "lm") +
labs(x = "Number of Cores", y = "Recommended Customer Price",
title = "Scatterplot of Recommended Customer Price
vs Number of Cores") +
theme(plot.title = element_text(hjust = 0.5, size = 13))
# Scatter plot between Recommended_Customer_Price and nb_of_Threads
ggplot(CPUs_data, aes(x = nb_of_Threads, y = Recommended_Customer_Price)) +
geom_point() +
geom_smooth(method = "lm") +
labs(x = "Number of Threads", y = "Recommended Customer Price",
title = "Scatterplot of Recommended Customer Price
```

```
    vs Number of Threads") +
    theme(plot.title = element_text(hjust = 0.5, size = 13))
    # Scatter plot between Recommended_Customer_Price and TDP
    ggplot(CPUs_data, aes(x = TDP, y = Recommended_Customer_Price)) +
    geom_point() +
    geom_smooth(method = "lm") +
    labs(x = "TDP", y = "Recommended Customer Price",
    title = "Scatterplot of Recommended Customer Price vs TDP") +
    theme(plot.title = element_text(hjust = 0.5, size = 13))
    # Scatter plot between Recommended_Customer_Price and Max_Memory_Size
    ggplot(CPUs_data, aes(x = Max_Memory_Size, y = Recommended_Customer_Price))
        +
    geom_point() +
    geom_smooth(method = "lm") +
    labs(x = "Max Memory Size", y = "Recommended Customer Price",
    title = "Scatterplot of Recommended Customer Price
    vs Max Memory Size") +
    theme(plot.title = element_text(hjust = 0.5, size = 13))
```

Output 1

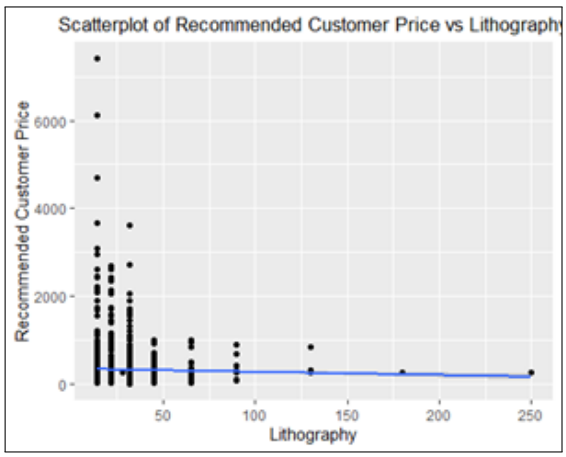

Figure 22: *Scatterplot of Recommended Customer Price vs Lithography*

The scatterplot above depicts a downward line of best fit, indicating a negative correlation between the variables. In real-life scenarios, the decrease in CPU lithography corresponds to an increase in required manufacturing technology. This advanced technology leads to higher production costs, resulting in an increased price for the CPU.

Figure 23: *Scatterplot of Recommended Customer Price vs Number of Cores*

The data points are scattered throughout the plot, with some points appearing to be denser than others. Based on the overall trend of the plot, it suggests that there is a positive correlation between the number of cores and the recommended customer price. This means that as the number of cores increases, the recommended customer price tends to increase as well.

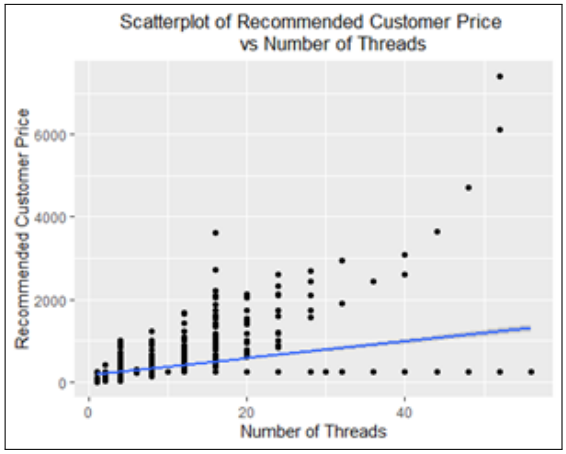

Figure 24: *Scatterplot of Recommended Customer Price vs Number of Threads*

The line of best fit of the above scatterplot suggests that there is a positive correlation between the number of threads and the recommended customer price.

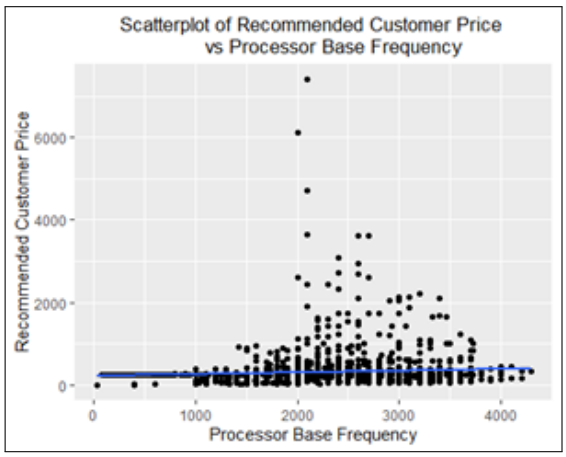

Figure 25: *Scatterplot of Recommended Customer Price vs Processor Base Frequency*

The scatterplot demonstrates a line of best fit that is nearly parallel to the x-axis. This suggests a weak or no correlation between the number of cores and the recommended customer price. The scattered distribution of data points further indicates a lack of a clear pattern between these variables.

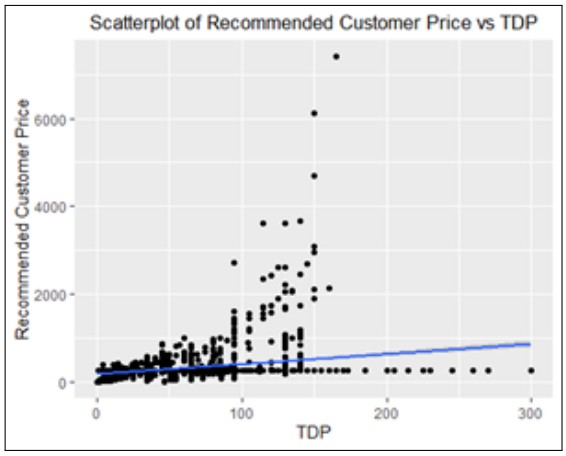

Figure 26: *Scatterplot of Recommended Customer Price vs TDP*

In the scatterplot above, it appears that the data points are gathered at one place rather than scattered throughout the plot, indicating a strong relationship between the recommended price and TDP. Based on the line of best fit, it seems that as TDP increases, the recommended price also tends to increase.

Figure 27: *Scatterplot of Recommended Customer Price vs Max Memory Size*

Based on the scatterplot, we can observe that there is a positive relationship between customer price and memory size. As customer price increases, memory size tends to increase as well.

Comment:

It is clear that there are some outliers present in the data and throughout these scatterplots. These outliers exist mainly due to the Median Imputation of the Recommended_Customer_Price (Md = 255.5) and the fact that Recommended_Customer_Price has many missing values. This forms a line of outliers that is parallel to the x-axis, which does not align with the general trend of the scatterplot. These outliers may have some impact on the accuracy of the line of best fit and overall understanding of the data.

Input 2

```
# Box plot between Recommended_Customer_Price and Vertical_Segment
ggplot(data = CPUs_data, aes(x = Vertical_Segment, y =
    Recommended_Customer_Price, fill = Vertical_Segment)) +
geom_boxplot() +
scale_y_continuous(trans = "log10") +
labs(x = "Vertical Segment", y = "Recommended Customer Price",
title = "Boxplot of Recommended Customer Price vs Vertical Segment") +
theme(plot.title = element_text(hjust = 0.5)) +
theme(legend.position="none")
# Box plot between Recommended_Customer_Price and Status
ggplot(data = CPUs_data, aes(x = Status, y = Recommended_Customer_Price,
    fill = Status)) +
geom_boxplot() +
scale_y_continuous(trans = "log10") +
labs(x = "Status", y = "Recommended Customer Price",
title = "Boxplot of Recommended Customer Price vs Status") +
theme(plot.title = element_text(hjust = 0.5)) +
theme(legend.position="none")
```

Output 2

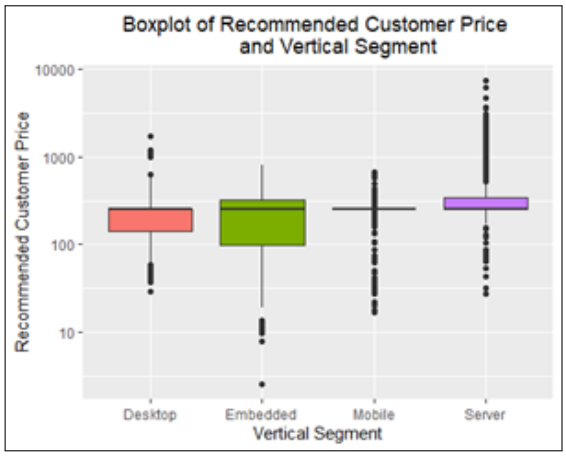

Figure 28: *Boxplot of Recommended Customer Price and Vertical Segment*

There are noticeable differences between four boxes suggests significant variations in the Recommended Customer Price and Vertical Segment across the different groups. The Embedded box appears larger than the others, it suggests that the data within this group has a wider spread or greater variability compared to the other groups. Conversely, the Mobile box appears almost as a line, meaning that the minimum, maximum, and quartile values are very close together, resulting in a narrow box.

Figure 29: *Boxplot of Recommended Customer Price and Status*

There are also noticeable differences between these four boxes. The line-like Annouced box suggests a narrow range of values or very little variability in the data. Meanwhile, the line-like End of Interactive box with significant number of outliers above and below it suggests that the majority of data points are concentrated around a single value, with a few data points that deviate significantly from the central value. The other two boxes have very different ranges of value, minimums, maximums, and quartile values.

Comment:

To address the issue caused by extreme outliers in the Recommended Customer Price, a transformation has been applied to the y-axis using the "log10" function. This helps to mitigate the impact of these extreme outliers, making the boxplots become clearer and easier to interpret.

5.3 Correlation Coefficients Calculation

Input 1

```
# Correlation matrix
correlation_matrix <- round(cor(CPUs_data[, c("Lithography",
"Recommended_Customer_Price"
,"nb_of_Cores","nb_of_Threads","TDP"
,"Max_Memory_Size")]), 2)
print(correlation_matrix)
```

Output 1

	Lithography	Recommended_Customer_Price	nb_of_Cores	nb_of_Threads	Processor_Base_Frequency	TDP	Max_Memory_Size
Lithography	1.00	-0.09	-0.28	-0.25	-0.31	-0.11	-0.21
Recommended_Customer_Price	-0.09	1.00	0.23	0.42	0.09	0.28	0.19
nb_of_Cores	-0.28	0.23	1.00	0.57	0.01	0.62	0.42
nb_of_Threads	-0.25	0.42	0.57	1.00	0.10	0.54	0.68
Processor_Base_Frequency	-0.31	0.09	0.01	0.10	1.00	0.47	0.07
TDP	-0.11	0.28	0.62	0.54	0.47	1.00	0.45
Max_Memory_Size	-0.21	0.19	0.42	0.68	0.07	0.45	1.00

Figure 30: *Table of Correlation Coefficients*

Input 2

```
corrplot(correlation_matrix, method = "number", type = "upper")
```

Output 2

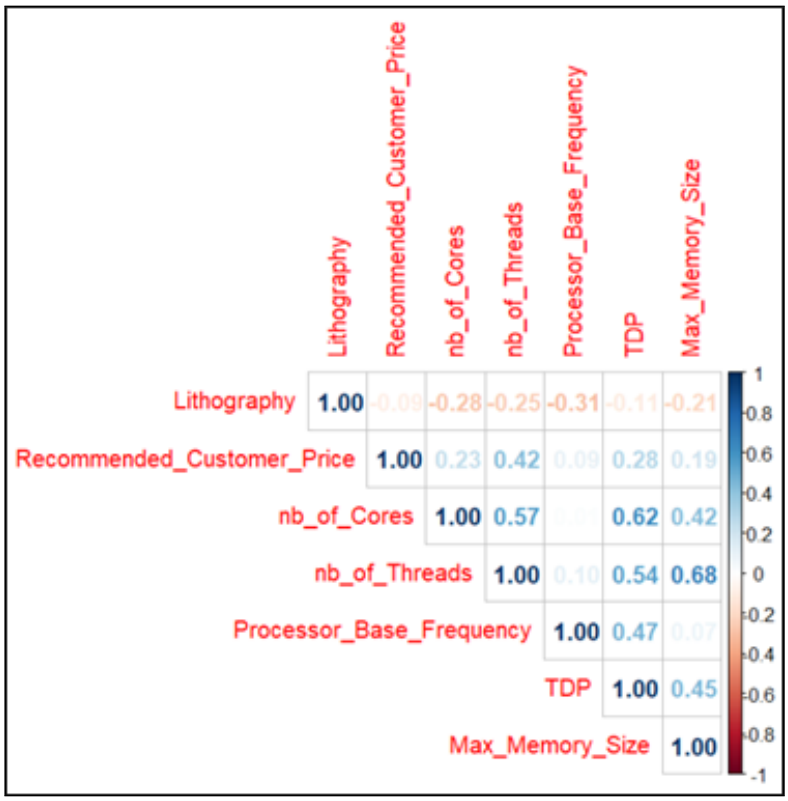

Figure 31: *Correlation coefficients between variables*

Comment:

From the corrplot graph, we can analyze the relationships between the variables in the dataset. Lithography shows a very weak correlation with Recommended Customer Price (-0.09) and TDP (-0.11). There is also a weak negative correlation between Lithography and Number of Cores (-0.28), Number of Threads (-0.25), Processor Base Frequency (-0.31), and Max Memory Size (-0.21).
On the other hand, Recommended Customer Price demonstrates weak positive correlations with Number of Cores and moderate correlations with Number of Threads, implying that higher prices are related to greater numbers of cores and threads in the CPUs.
Furthermore, Number of Cores and Number of Threads exhibit a moderate positive correlation (0.57), indicating a close association between these variables. This suggests that CPUs with a higher number of cores are likely to have a higher number of threads as well.
Additionally, TDP and Max Memory Size show moderate positive correlations with Number of Cores, Number of Threads, and each other. This suggests that CPUs with higher thermal design power (TDP) values and larger maximum memory sizes tend to have more cores and threads.
Overall, the variables are weakly correlated, it can be predicted that the variables are independent.

6 Inferential statistics

6.1 Multiple Linear Regression Model

First, we assume that the assumptions for MLR Model are met. Then we start building the model as followings :
Our model can be described as the function:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n + \epsilon$$

Where:

- Y: dependent variable
- β_0: intercept coefficient
- β_i: regression coefficients
- X_i: independent variables
- ϵ: error

We have two categorical variables:

- Vertical_Segment (Embedded, Mobile, Server, Desktop)
- Status (InteractiveSupport, Life, Launched, Announced)

From the data, we have the equation of the model:

$$\begin{aligned}\text{Recommended\_Customer\_Price} = \beta_0 &+ (\beta_1 \text{Lithography}) + (\beta_2 \text{nb\_of\_Cores}) + (\beta_3 \text{nb\_of\_Threads}) \\ &+ (\beta_4 \text{Processor\_Base\_Frequency}) + (\beta_5 \text{TDP}) + (\beta_6 \text{Max\_Memory\_Size}) \\ &+ (\beta_7 \text{Embedded}) + (\beta_8 \text{Mobile}) + (\beta_9 \text{Server}) \\ &+ (\beta_{10} \text{InteractiveSupport}) + (\beta_{11} \text{Life}) + (\beta_{12} \text{Launched}) + \epsilon\end{aligned}$$

Then we have the estimation equation:

$$\begin{aligned}\text{Recommended\_Customer\_Price} = \hat{\beta}_0 &+ (\hat{\beta}_1 \text{Lithography}) + (\hat{\beta}_2 \text{nb\_of\_Cores}) + (\hat{\beta}_3 \text{nb\_of\_Threads}) \\ &+ (\hat{\beta}_4 \text{Processor\_Base\_Frequency}) + (\hat{\beta}_5 \text{TDP}) + (\hat{\beta}_6 \text{Max\_Memory\_Size}) \\ &+ (\hat{\beta}_7 \text{Embedded}) + (\hat{\beta}_8 \text{Mobile}) + (\hat{\beta}_9 \text{Server}) \\ &+ (\hat{\beta}_{10} \text{InteractiveSupport}) + (\hat{\beta}_{11} \text{Life}) + (\hat{\beta}_{12} \text{Launched})\end{aligned}$$

Model 1

Input 1

```
Model_1<-lm(Recommended_Customer_Price ~ Lithography + nb_of_Cores +
    nb_of_Threads + Processor_Base_Frequency + TDP + Max_Memory_Size +
    Vertical_Segment + Status, data=CPUs_data)
summary(Model_1)
```

Output 1 From the analysis result, we have:

```
Call:
lm(formula = Recommended_Customer_Price ~ Lithography + nb_of_Cores +
    nb_of_Threads + Processor_Base_Frequency + TDP + Max_Memory_Size +
    Vertical_Segment + Status, data = CPUs_data)

Residuals:
    Min      1Q  Median      3Q     Max
-1277.0   -93.0   -12.4    42.1  6008.7

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                   -2.446e+02  9.232e+01  -2.649  0.00813 **
Lithography                    5.996e-03  2.259e-01   0.027  0.97883
nb_of_Cores                   -3.477e+00  1.766e+00  -1.969  0.04910 *
nb_of_Threads                  2.549e+01  1.448e+00  17.600  < 2e-16 ***
Processor_Base_Frequency       1.163e-02  1.246e-02   0.933  0.35106
TDP                            8.394e-01  3.267e-01   2.569  0.01026 *
Max_Memory_Size               -1.793e-01  2.112e-02  -8.489  < 2e-16 ***
Vertical_SegmentEmbedded       4.655e+01  3.204e+01   1.453  0.14649
Vertical_SegmentMobile         9.012e+01  2.097e+01   4.298  1.8e-05 ***
Vertical_SegmentServer         1.147e+02  2.259e+01   5.079  4.1e-07 ***
StatusEnd of Interactive Support  2.857e+02  8.737e+01   3.270  0.00109 **
StatusEnd of Life              2.997e+02  8.766e+01   3.419  0.00064 ***
StatusLaunched                 2.688e+02  8.567e+01   3.138  0.00173 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 326.4 on 2270 degrees of freedom
Multiple R-squared:  0.2191,    Adjusted R-squared:  0.215
F-statistic: 53.07 on 12 and 2270 DF,  p-value: < 2.2e-16
```

Figure 32: *Linear Regression Model [1]*

$$\hat{\beta}_0 = -244.6, \quad \hat{\beta}_1 = 0.006, \quad \hat{\beta}_2 = -3.477, \quad \hat{\beta}_3 = 25.49, \quad \hat{\beta}_4 = 0.0116, \quad \hat{\beta}_5 = 0.8394,$$

$$\hat{\beta}_6 = -0.1793, \quad \hat{\beta}_7 = 46.55, \quad \hat{\beta}_8 = 90.12, \quad \hat{\beta}_9 = 114.7, \quad \hat{\beta}_{10} = 285.7, \quad \hat{\beta}_{11} = 299.7, \quad \hat{\beta}_{12} = 268.8$$

$\Rightarrow$ Recommended_Customer_Price $= -244.6+0.006$Lithography-3.477nb_of_Cores$+25.49$nb_of_Threads$+0.0116$

-0.1793Max_Memory_Size$+46.55$Embedded$+90.12$Mobile$+114.7$Server$+285.7$InteractiveSupport$+299.7$Life$+268.8$

Set test hypothesis:

- H_0: The regression coefficients are not statistically significant ($\beta_i = 0$)
- H_1: The regression coefficients are statistically significant ($\beta_i \neq 0$)

Remark:
Some independent variables including Lithography, Processor_Base_Frequency, and Vertical_SegmentEmbedded in the above multiple linear regression model have p-values $>$ 5%. Therefore, the condition to reject H_0 is not satisfied, meaning we still have to accept H_0, meaning these variables do not bring statistical significance to the above multiple linear regression model. Therefore, we need to remove them from the model in descending order of p-values.

Model 2

Remove the variable Lithography from Model_1
Input 2

```
Model_2<-lm(Recommended_Customer_Price ~ nb_of_Cores + nb_of_Threads +
    Processor_Base_Frequency + TDP + Max_Memory_Size + Vertical_Segment +
    Status, data=CPUs_data)
summary(Model_2)
```

Output 2

Model 3

Remove the variable Processor_Base_Frequency from Model_2
Input 3

```
Model_3<-lm(Recommended_Customer_Price ~ nb_of_Cores + nb_of_Threads + TDP +
     Max_Memory_Size + Vertical_Segment + Status, data=CPUs_data)
summary(Model_3)
```

Output 3

```
Call:
lm(formula = Recommended_Customer_Price ~ nb_of_Cores + nb_of_Threads +
    Processor_Base_Frequency + TDP + Max_Memory_Size + Vertical_Segment +
    Status, data = CPUs_data)

Residuals:
    Min      1Q  Median      3Q     Max
-1276.9   -93.0   -12.3    42.1  6008.8

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                   -244.14245   90.97591  -2.684 0.007336 **
nb_of_Cores                     -3.48882    1.71016  -2.040 0.041460 *
nb_of_Threads                   25.48550    1.44403  17.649  < 2e-16 ***
Processor_Base_Frequency         0.01151    0.01170   0.984 0.325164
TDP                              0.84099    0.32128   2.618 0.008913 **
Max_Memory_Size                 -0.17933    0.02103  -8.529  < 2e-16 ***
Vertical_SegmentEmbedded        46.51097   32.01102   1.453 0.146371
Vertical_SegmentMobile          90.03734   20.74359   4.340 1.48e-05 ***
Vertical_SegmentServer         114.82339   22.39709   5.127 3.20e-07 ***
StatusEnd of Interactive Support 286.00461  86.53614   3.305 0.000964 ***
StatusEnd of Life              299.74417   87.62800   3.421 0.000636 ***
StatusLaunched                 268.77936   85.64615   3.138 0.001721 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 326.4 on 2271 degrees of freedom
Multiple R-squared:  0.2191,    Adjusted R-squared:  0.2153
F-statistic: 57.92 on 11 and 2271 DF,  p-value: < 2.2e-16
```

Figure 33: *Linear Regression Model [2]*

```
Call:
lm(formula = Recommended_Customer_Price ~ nb_of_Cores + nb_of_Threads +
    TDP + Max_Memory_Size + Vertical_Segment + Status, data = CPUs_data)

Residuals:
    Min      1Q  Median      3Q     Max
-1280.3   -92.0    -9.7    42.8  6006.4

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                   -223.22106   88.45654  -2.524 0.011687 *
nb_of_Cores                     -4.21904    1.54079  -2.738 0.006225 **
nb_of_Threads                   25.46040    1.44379  17.634  < 2e-16 ***
TDP                              1.02016    0.26470   3.854 0.000119 ***
Max_Memory_Size                 -0.18105    0.02095  -8.641  < 2e-16 ***
Vertical_SegmentEmbedded        39.69175   31.25183   1.270 0.204193
Vertical_SegmentMobile          87.17524   20.53856   4.244 2.28e-05 ***
Vertical_SegmentServer         108.16754   21.35137   5.066 4.39e-07 ***
StatusEnd of Interactive Support 281.90371  86.43515   3.261 0.001125 **
StatusEnd of Life              301.84320   87.60143   3.446 0.000580 ***
StatusLaunched                 273.13723   85.53100   3.193 0.001425 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 326.4 on 2272 degrees of freedom
Multiple R-squared:  0.2188,    Adjusted R-squared:  0.2153
F-statistic: 63.62 on 10 and 2272 DF,  p-value: < 2.2e-16
```

Figure 34: *Linear Regression Model [3]*

Model 4

Remove the variable Vertical_Segment from Model_3

Input 4

```
Model_4<-lm(Recommended_Customer_Price ~ nb_of_Cores + nb_of_Threads + TDP +
    Max_Memory_Size + Status, data=CPUs_data)
summary(Model_4)
```

Output 4

```
Call:
lm(formula = Recommended_Customer_Price ~ nb_of_Cores + nb_of_Threads +
    TDP + Max_Memory_Size + Status, data = CPUs_data)

Residuals:
    Min      1Q  Median      3Q     Max
-1294.9   -98.2    -7.4    35.9  5990.8

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                   -171.00456   86.94828  -1.967 0.049335 *
nb_of_Cores                     -3.21896    1.52180  -2.115 0.034519 *
nb_of_Threads                   26.33273    1.43900  18.299  < 2e-16 ***
TDP                              0.85941    0.22858   3.760 0.000174 ***
Max_Memory_Size                 -0.16823    0.02091  -8.046 1.36e-15 ***
StatusEnd of Interactive Support 281.30499   86.78504   3.241 0.001207 **
StatusEnd of Life               320.77412   87.61672   3.661 0.000257 ***
StatusLaunched                  289.97201   86.03395   3.370 0.000763 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 328.7 on 2275 degrees of freedom
Multiple R-squared:  0.2064,    Adjusted R-squared:  0.204
F-statistic: 84.53 on 7 and 2275 DF,  p-value: < 2.2e-16
```

Figure 35: *Linear Regression Model [4]*

Models Comparison

We compapre the Model_1 and Model_2:
Input 5

```
anova(Model_1,Model_2)
```

Output 5

```
> anova(Model_1,Model_2)
Analysis of Variance Table

Model 1: Recommended_Customer_Price ~ Lithography + nb_of_Cores + nb_of_Threads +
    Processor_Base_Frequency + TDP + Max_Memory_Size + Vertical_Segment +
    Status
Model 2: Recommended_Customer_Price ~ nb_of_Cores + nb_of_Threads + Processor_Base_Frequency +
    TDP + Max_Memory_Size + Vertical_Segment + Status
  Res.Df       RSS Df Sum of Sq     F Pr(>F)
1   2270 241905623
2   2271 241905698 -1   -75.071 7e-04 0.9788
```

Figure 36: *Linear Regression Model Comparison between Model 1 and 2*

Set test hypothesis:

$$\begin{aligned}\text{Recommended\_Customer\_Price} =& \beta_0 + (\beta_1 \text{Lithography}) + (\beta_2 \text{nb\_of\_Cores}) + (\beta_3 \text{nb\_of\_Threads}) \\ &+ (\beta_4 \text{Processor\_Base\_Frequency}) + (\beta_5 \text{TDP}) \\ &+ (\beta_6 \text{Max\_Memory\_Size}) + (\beta_7 \text{Embedded}) + (\beta_8 \text{Mobile}) \\ &+ (\beta_9 \text{Server}) + (\beta_{10} \text{InteractiveSupport}) + (\beta_{11} \text{Life}) + (\beta_{12} \text{Launched}) + \epsilon\end{aligned}$$

- H_0: Model_2 is better ($\beta_1 = 0$)
- H_1: Model_1 is better ($\beta_1 \neq 0$)

Remark: p-value $= 0.9788 > 5\%$. Therefore, the condition to reject the null hypothesis H_0 is not satisfied, implying that Model_2 is better.

We compare the Model_2 and Model_3:
Input 6

```
anova(Model_2,Model_3)
```

Output 6

```
> anova(Model_1,Model_2)
Analysis of Variance Table

Model 1: Recommended_Customer_Price ~ Lithography + nb_of_Cores + nb_of_Threads +
    Processor_Base_Frequency + TDP + Max_Memory_Size + Vertical_Segment +
    Status
Model 2: Recommended_Customer_Price ~ nb_of_Cores + nb_of_Threads + Processor_Base_Frequency +
    TDP + Max_Memory_Size + Vertical_Segment + Status
  Res.Df       RSS Df Sum of Sq     F Pr(>F)
1   2270 241905623
2   2271 241905698 -1   -75.071 7e-04 0.9788
```

Figure 37: *Linear Regression Model Comparison between Model 2 and 3*

Set test hypothesis:

$$\begin{aligned}\text{Recommended\_Customer\_Price} =& \beta_0 + \beta_2 \text{nb\_of\_Cores} + \beta_3 \text{nb\_of\_Threads} + \beta_4 \text{Processor\_Base\_Frequency} \\ &+ \beta_5 \text{TDP} + \beta_6 \text{Max\_Memory\_Size} \\ &+ \beta_7 \text{Embedded} + \beta_8 \text{Mobile} + \beta_9 \text{Server} \\ &+ \beta_{10} \text{InteractiveSupport} + \beta_{11} \text{Life} + \beta_{12} \text{Launched} + \epsilon\end{aligned}$$

- H_0: Model_3 is better ($\beta_4 = 0$)
- H_1: Model_2 is better ($\beta_4 \neq 0$)

Remark: p-value $= 0.3252 > 5\%$. Therefore, the condition to reject the null hypothesis H_0 is not satisfied, implying that Model_3 is better.

We compapre the Model_3 and Model_4:
Input 7

```
anova(Model_3,Model_4)
```

Output 7

```
> anova(Model_3,Model_4)
Analysis of Variance Table

Model 1: Recommended_Customer_Price ~ nb_of_Cores + nb_of_Threads + TDP +
    Max_Memory_Size + Vertical_Segment + Status
Model 2: Recommended_Customer_Price ~ nb_of_Cores + nb_of_Threads + TDP +
    Max_Memory_Size + Status
  Res.Df       RSS Df Sum of Sq      F    Pr(>F)
1   2272 242008860
2   2275 245834553 -3  -3825693 11.972 8.91e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 38: *Linear Regression Model Comparison between Model 3 and 4*

Set test hypothesis:

$$\begin{aligned}\text{Recommended\_Customer\_Price} =& \beta_0 + \beta_2 \text{nb\_of\_Cores} + \beta_3 \text{nb\_of\_Threads} + \beta_5 \text{TDP} + \beta_6 \text{Max\_Memory\_Size} \\ &+ \beta_7 \text{Embedded} + \beta_8 \text{Mobile} + \beta_9 \text{Server} + \beta_{10} \text{InteractiveSupport} \\ &+ \beta_{11} \text{Life} + \beta_{12} \text{Launched} + \epsilon\end{aligned}$$

- H_0: Model_4 is better ($\beta_7 = \beta_8 = \beta_9 = 0$)
- H_1: Model_3 is better ($\beta_7 \neq 0$ or $\beta_8 \neq 0$ or $\beta_9 \neq 0$)

Remark: p-value $= 8.91 \times 10^{-8} < 5\%$. Therefore, the condition to reject the null hypothesis H_0 is satisfied, implying that Model_3 is better.
Therefore, the regression model that best fits is Model 3:

$$\begin{aligned}\text{Recommended\_Customer\_Price} &= \hat{\beta}_0 + \hat{\beta}_2 \text{nb\_of\_Cores} + \hat{\beta}_3 \text{nb\_of\_Threads} + \hat{\beta}_5 \text{TDP} + \hat{\beta}_6 \text{Max\_Memory\_Size} \\ &= -223.2211 - 4.219 \text{nb\_of\_Cores} + 25.4604 \text{nb\_of\_Threads} + 1.0202 \text{TDP} - \\ &\quad 0.1811 \text{Max\_Memory\_Size} + 39.6918 \text{Embedded} + 87.1752 \text{Mobile} \\ &\quad + 108.1675 \text{Server} + 281.9037 \text{InteractiveSupport} + 301.8432 \text{Life} \\ &\quad + 273.1372 \text{Launched}\end{aligned}$$

Analyzing the impact of factors on the recommended customer price:

- First, we observe that the p-value associated with the model is 2.2×10^{-16}, which is highly significant. This indicates that there is at least one predictor variable that has a significant impact on explaining the Recommended_Customer_Price variable.

- To examine the specific influence of each independent variable, we look at the corresponding p-values. We observe that four variables, `nb_of_Threads`, `Max_Memory_Size`, `Server`, and `Mobile`, have very low p-values, less than 0.05, with values of less than 2×10^{-16}, 4.39×10^{-7}, and 2.28×10^{-5}, respectively. This indicates that these three variables have a significant impact on the recommended price, while the remaining variables `TDP`, `Life`, `Launched`, `Life`, `InteractiveSupport`, and `nb_of_Cores` have less influence, and `Embedded` has no influence on the price.
- Additionally, the regression coefficients ($\hat{B}_i$) of the variables are considered to have a moderate effect on the Recommended_Customer_Price. For example, if we have $\hat{\beta}_3 = 25.4604$, then when the number of Threads increases by 1, we can expect the Recommended_Customer_Price to increase by 25.4604 on average (assuming other predictor variables remain unchanged). The same applies to the other variables.
- The Adjusted R-squared coefficient is 0.2153, meaning that 21.53% of the variation in the Recommended_Customer_Price is explained by the independent variables included in the model.

Checking the assumptions of the model:

- Linear relationship between the outcome and the predictors
- Residuals are normally distributed: $\epsilon_i \sim N(0, \sigma^2)$
- Residual errors have constant variance
- Residual errors have a mean value of zero

We conduct residual analysis to check the assumptions of the models:

```
1    plot(Model_3) #Draw residual plots
```

Output of residual plots:
Linear Regression Plots: Fitted vs Residuals show the relationship between the fitted values (predicted values from the regression model) and the residuals (the differences between observed and predicted values).

In this plot, the horizontal axis represents the fitted values, which are the predicted values of the response variable based on the regression model. The vertical axis represents the residuals, which are the discrepancies between the observed values and the fitted values.

The main purpose of this plot is to assess the assumptions of linear regression:

- Linear relationship between the outcome and the predictors
- Residual Errors have a mean value of zero
- Residual Errors have constant variance

Remark:

- In the Residuals vs Fitted plot, we can see that the residual points are not all equally spread out, indicating that we failed to meet the assumption that Residual Errors have constant variance. The red line is almost horizontal and closely aligns with the line where residuals are equal to zero, indicating that it represents the assumption that there is a Linear relationship between the outcome and the predictors and Residual Errors have a mean value of zero.

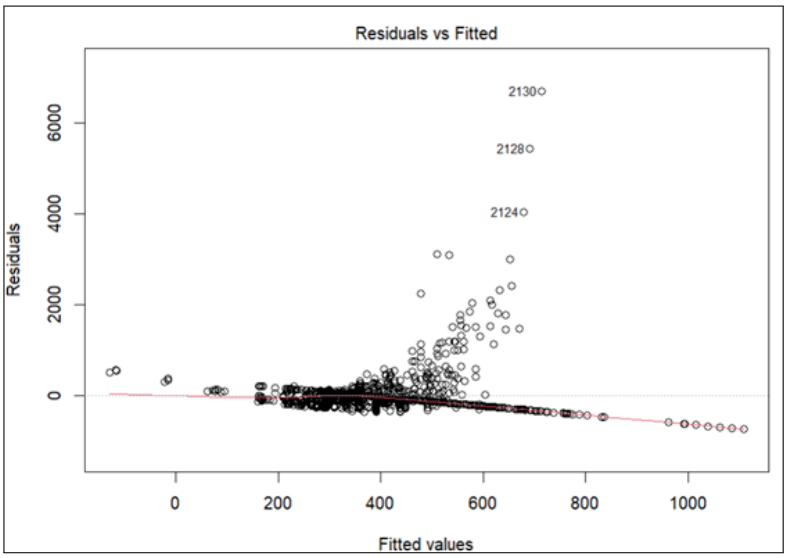

Figure 39: *Residuals and fitted plot*

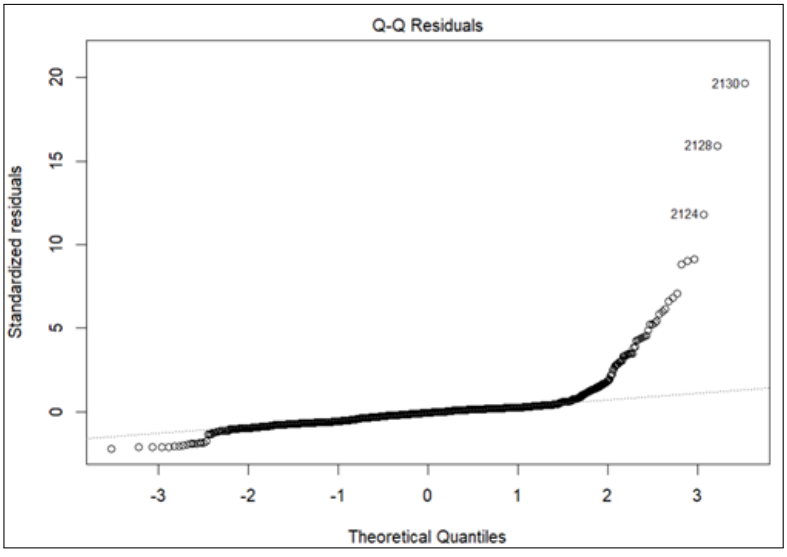

Figure 40: *Q-Q residuals plot*

The Q-Q Residuals plot is used to assess the assumptions of linear regression to determine whether the residuals (the differences between observed and predicted values) follow a normal distribution. In this plot, the x-axis represents the theoretical quantiles of a normal distribution, while the y-axis represents the observed residuals. If the residuals follow a normal distribution, the points on the Q-Q plot will fall approximately along a straight line. Any deviations from this straight line suggest departures from normality in the residuals.

Remark:

- In the Q-Q Residuals plot, we can see many points deviate from the expected line of Normal Distribution, indicating that there are some differences between the data and the reference distribution. This would indicate that we failed to meet the assumption that Residuals are normally distributed.

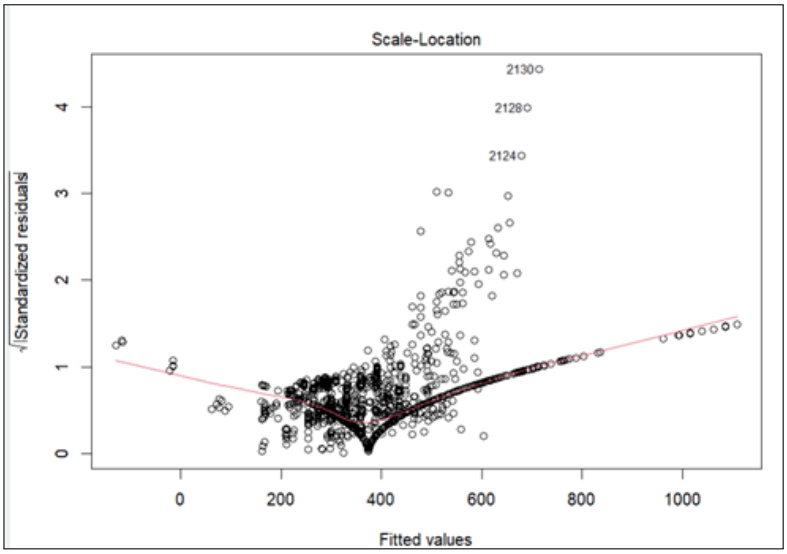

Figure 41: *Scale location plot*

The Scale-Location plot is a diagnostic plot used in regression analysis to assess the assumption that Residual Errors have constant variance. In this plot, the x-axis typically represents the fitted values, while the y-axis represents the square root of the absolute residuals or standardized residuals. The square root transformation is applied to make the spread more symmetric around the mean and stabilize the variance across different levels of the predictor variables.

Remark:

- In the Scale-Location plot, we can see that the red line is not horizontal and the residual points are not all equally spread along the ranges of predictors, indicating that we failed to meet the assumption that Residual Errors have constant variance.

A Residuals vs Leverage plot is a type of diagnostic plot that allows us to identify influential observations in a regression model. Each observation from the dataset is shown as a single point within the plot. The x-axis shows the leverage of each point, and the y-axis shows the standardized residual of each point. Leverage refers to the extent to which the coefficients in the regression model would change if a particular observation was removed from the dataset. Observations with high leverage have a strong influence on the coefficients in the regression model. If we remove these observations, the coefficients of the model would change noticeably.

Remark:

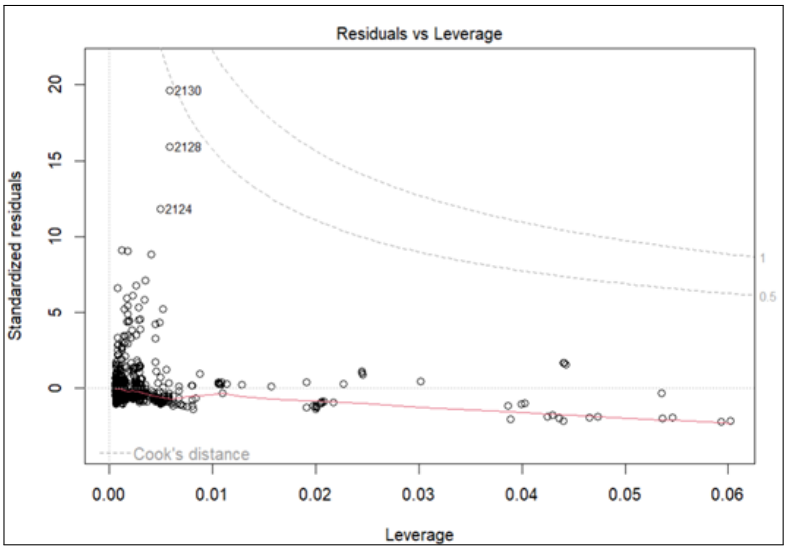

Figure 42: *Residuals and leverage plot*

- In the Residuals vs Leverage plot, we can see that observation #2130 lies closest to the border of Cook's distance, but it doesn't fall outside of the dashed line. This means there are not any influential points in our regression model.

Prediction:

Model 3 has been determined to be the most suitable model for fitting the data, and as such, it will be utilized to predict the value of Recommended_Customer_Price.

$$\begin{aligned}\text{Recommended\_Customer\_Price} = &-223.2211 - 4.219\text{nb\_of\_Cores} + 25.4604\text{nb\_of\_Threads} \\ &+ 1.0202\text{TDP} - 0.1811\text{Max\_Memory\_Size} + 39.6918\text{Embedded} \\ &+ 87.1752\text{Mobile} + 108.1675\text{Server} \\ &+ 281.9037\text{InteractiveSupport} + 301.8432\text{Life} \\ &+ 273.1372\text{Launched}\end{aligned}$$

```
new_x = data.frame(CPUs_data[,c(1,2,3,5,6,7,8,9)])
new_x$pred_Price = predict(Model_3, new_x)
head(new_x$pred_Price, 10)
```

The Recommended_Customer_Price for the initial 10 observations will be predicted as follows:

Result: Based on the provided actual and predicted prices, we can make the following

```
> new_x = data.frame(CPUs_data[,c(1,2,3,5,6,7,8,9)])
>
> new_x$pred_Price = predict(Model_3, new_x)
> head(new_x$pred_Price, 10)
 [1] 232.1888 333.4072 333.4072 386.4215 232.1888 191.9797 183.2366 101.1521 142.7142 327.4612
```

Figure 43: *Illustration of contribution of Recommended Customer Price for desktop CPUs*

conclusions about the MLR (Multiple Linear Regression) model:

Recommended_Customer_Price
393.00
297.00
409.00
305.00
281.00
107.00
255.50
255.50
42.00
255.50

Figure 44: *Illustration of contribution of Recommended Customer Price for desktop CPUs*

1. The model's predictions are not always accurate: The predicted prices do not match the actual prices in most cases, indicating that the model may not be capturing all the relevant information or patterns in the data.

2. The model may have a tendency to overestimate or underestimate certain prices: For example, the predicted price for observation 3 is much lower than the actual price, while the predicted price for observation 9 is much higher. This suggests that the model may be biased in certain ranges of the data.

3. The model may be more accurate for certain types of observations: For instance, the predicted prices for observations 7 and 8 are relatively close to their actual values, while the predicted prices for observations 1 and 5 are further off. This could indicate that the model is better at predicting prices for certain types of products or in certain market conditions.

Overall, based on this limited data, it is difficult to make definitive conclusions about the MLR model's performance. However, these observations suggest that there may be room for improvement in terms of model accuracy and further analysis would be needed to fully evaluate the model's strengths and weaknesses.

6.2 Two-way ANOVA Model

Here, we will define our model as **Comparing the average Recommended Customer Price across different vertical segments and status**.
ANOVA (Analysis of Variance) model is based on certain assumptions. Adherence to these assumptions is crucial to ensure the validity of the analysis results.

6.2.1 Normality

To check whether it follows a normal distribution, we can use the Shapiro-Wilk test, which is implemented in R programming language through the shapiro.test() function. This test evaluates the null hypothesis that a sample comes from a normally distributed population.

Vertical Segment

1. For desktop:

```
Desktop_data <- subset(CPUs_data,CPUs_data$Vertical_Segment=="Desktop")
shapiro.test(Desktop_data$Recommended_Customer_Price)
qqnorm(Desktop_data$Recommended_Customer_Price)
qqline(Desktop_data$Recommended_Customer_Price)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Desktop_data$Recommended_Customer_Price
## W = 0.6181, p-value < 2.2e-16
```

Figure 45: *Shapiro-Wilk normality test of Recommended Customer Price for desktop CPUs*

- Null hypothesis: H_0: The recommended customer price of desktops follows a normal distribution.
- Alternative hypothesis: H_1: The recommended customer price of desktops does not follow a normal distribution.
- Since the p-value $< 2.2 \times 10^{-16}$ (which is less than the significance level of 5%), we reject the null hypothesis. Therefore, we conclude that the recommended customer price of desktops does not follow a normal distribution.
- As the observations don't stick to the straight line, we can predict that the recommended customer price of desktops does not follow a normal distribution. This prediction matches with Shapiro- Wilk normality test.

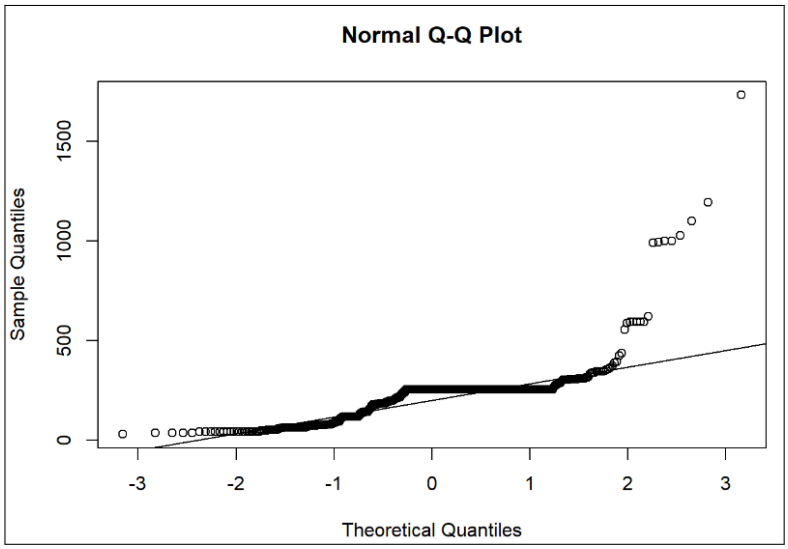

Figure 46: *Illustration of contribution of Recommended Customer Price for desktop CPUs*

2. For mobile:

```
Mobile_data <- subset(CPUs_data, CPUs_data$Vertical_Segment == "Mobile"
    )
shapiro.test(Mobile_data$Recommended_Customer_Price)
qqnorm(Mobile_data$Recommended_Customer_Price)
qqline(Mobile_data$Recommended_Customer_Price)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Mobile_data$Recommended_Customer_Price
## W = 0.76581, p-value < 2.2e-16
```

Figure 47: *Shapiro-Wilk normality test of Recommended Customer Price for mobile CPUs*

- Null hypothesis: H_0: The recommended customer price of mobile devices follows a normal distribution.
- Alternative hypothesis: H_1: The recommended customer price of mobile devices does not follow a normal distribution.
- Since the p-value $< 2.2 \times 10^{-16}$ (which is less than the significance level of 5%), we reject the null hypothesis. Therefore, we conclude that the recommended customer price of mobile devices does not follow a normal distribution.

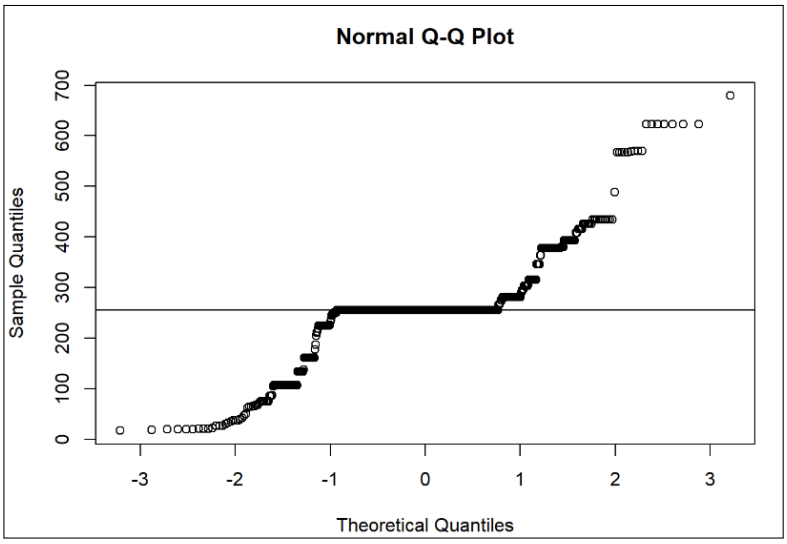

Figure 48: *Illustration of contribution of Recommended Customer Price for mobile CPUs*

3. For server:

```
Server_data <- subset(CPUs_data,CPUs_data$Vertical_Segment=="Server")
shapiro.test(Server_data$Recommended_Customer_Price)
qqnorm(Server_data$Recommended_Customer_Price)
qqline(Server_data$Recommended_Customer_Price)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Server_data$Recommended_Customer_Price
## W = 0.42001, p-value < 2.2e-16
```

Figure 49: *Shapiro-Wilk normality test of Recommended Customer Price for server CPUs*

- Null hypothesis: H_0: The recommended customer price of servers follows a normal distribution.
- Alternative hypothesis: H_1: The recommended customer price of servers does not follow a normal distribution.
- Since the p-value $< 2.2 \times 10^{-16}$ (which is less than the significance level of 5%), we reject the null hypothesis. Therefore, we conclude that the recommended customer price of servers does not follow a normal distribution.

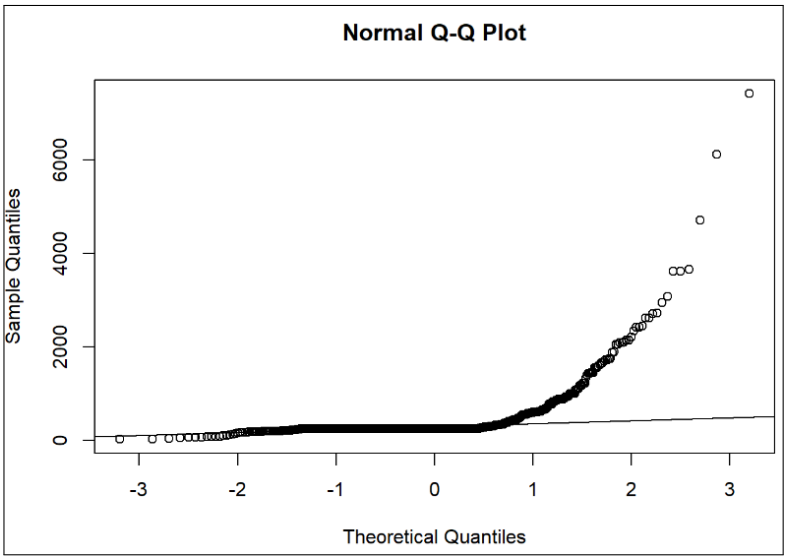

Figure 50: *Illustration of contribution of Recommended Customer Price for server CPUs*

4. For embedded:

```
Embedded_data <- subset(CPUs_data, CPUs_data$Vertical_Segment == "
    Embedded")
shapiro.test(Embedded_data$Recommended_Customer_Price)
qqnorm(Embedded_data$Recommended_Customer_Price)
qqline(Embedded_data$Recommended_Customer_Price)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Embedded_data$Recommended_Customer_Price
## W = 0.93651, p-value = 4.825e-07
```

Figure 51: *Shapiro-Wilk normality test of Recommended Customer Price for embedded CPUs*

- Null hypothesis: H_0: The recommended customer price of embedded devices follows a normal distribution.
- Alternative hypothesis: H_1: The recommended customer price of embedded devices does not follow a normal distribution.
- Since the p-value $= 4.825 \times 10^{-7}$ (which is less than the significance level of 5%), we reject the null hypothesis. Therefore, we conclude that the recommended customer price of embedded devices does not follow a normal distribution.

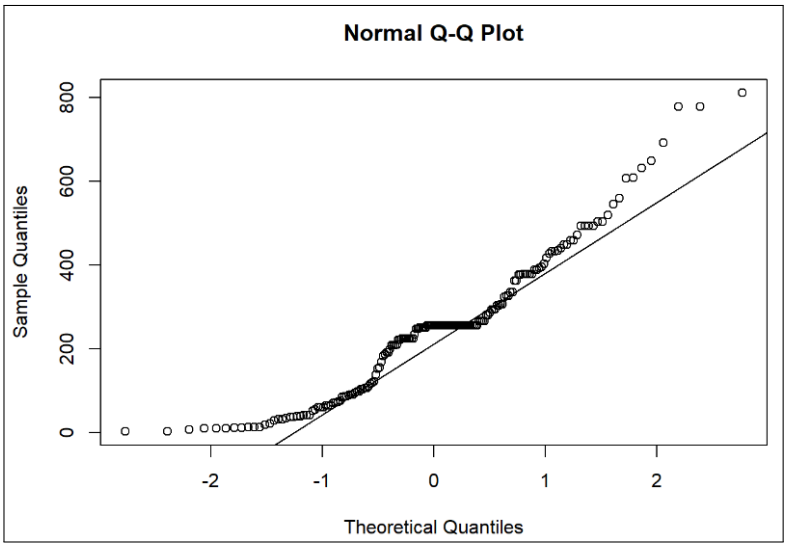

Figure 52: *Illustration of contribution of Recommended Customer Price for embedded CPUs*

Status

1. For Announced:
   Statistical analysis aims to uncover patterns, trends, or relationships within data, but if all the values are the same, there is no variation to analyze. Essentially, statistical analysis relies on differences and variability in data to draw meaningful conclusions. When there is no variability, there is no uncertainty or randomness to investigate, making further analysis unnecessary.
   In statistics, if a dataset consists of constant values or lacks variability, there is no need for analysis because there is no information to be gained from such data. Here, Announced accompanies with a constant recommended price, that's why we won't take it into account.

```
Announced_data <- subset(CPUs_data,CPUs_data$Status=="Announced")
```

	Vertical_Segment	Status	Lithography	Recommended_Customer_Price	nb_of_Cores	nb_of_Threads	Processor_Base_Frequency	TDP	Max_Memory_Size
464	Desktop	Announced	32	255.5	14	28	3100	165.0	128
465	Desktop	Announced	32	255.5	16	32	2800	165.0	128
466	Desktop	Announced	32	255.5	18	36	2600	165.0	128
567	Mobile	Announced	28	255.5	2	2	1000	47.0	1
574	Mobile	Announced	28	255.5	4	4	1200	47.0	2
575	Mobile	Announced	28	255.5	4	4	1200	47.0	2
582	Embedded	Announced	14	255.5	2	2	1300	6.5	8
583	Embedded	Announced	14	255.5	4	4	1600	9.5	8
585	Embedded	Announced	14	255.5	4	4	1600	12.0	8
761	Server	Announced	22	255.5	16	32	2300	135.0	768
770	Server	Announced	22	255.5	18	36	2300	145.0	768
2088	Server	Announced	14	255.5	12	24	2100	85.0	768
2091	Server	Announced	14	255.5	14	28	1900	85.0	768
2227	Mobile	Announced	14	255.5	2	4	2200	15.0	32
2228	Mobile	Announced	14	255.5	2	4	3300	28.0	32

Figure 53: *Announced_data set*

2. For End of Interactive Support:

```
End_of_Interactive_Support_data <- subset(CPUs_data,
    CPUs_data$Status=="End of Interactive Support")
shapiro.test(
    End_of_Interactive_Support_data$Recommended_Customer_Price)
qqnorm(End_of_Interactive_Support_data$Recommended_Customer_Price)
qqline(End_of_Interactive_Support_data$Recommended_Customer_Price)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  End_of_Interactive_Support_data$Recommended_Customer_Price
## W = 0.38612, p-value < 2.2e-16
```

Figure 54: *Shapiro-Wilk normality test of Recommended Customer Price for End of Interactive Support Status*

- Null hypothesis: H_0: The recommended customer price of End of Interactive Support status follows a normal distribution.
- Alternative hypothesis: H_1: The recommended customer price of End of Interactive Support status does not follow a normal distribution.
- Since the p-value $< 2.2 \times 10^{-16}$ (which is less than the significance level of 5%), we reject the null hypothesis. Therefore, we conclude that the recommended customer price of mobile devices does not follow a normal distribution.

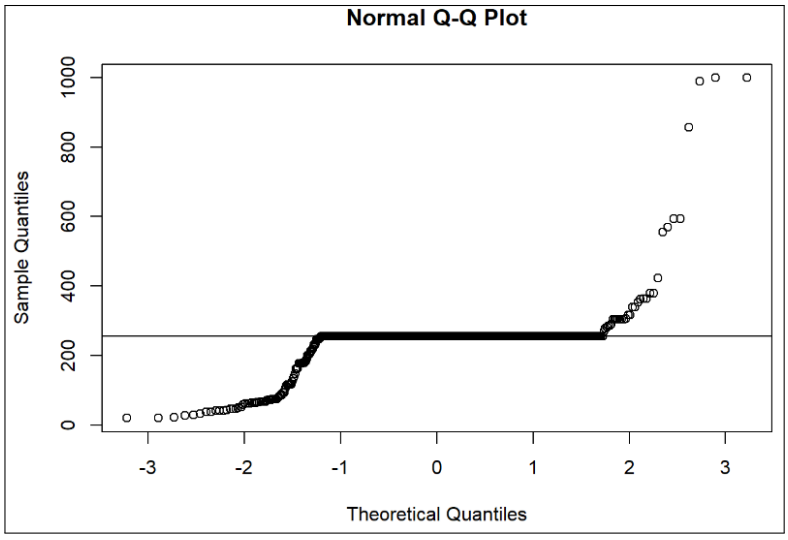

Figure 55: *Illustration of contribution of Recommended Customer Price for End of Interactive Support status*

3. End of life:

```
End_of_Life_data <- subset(CPUs_data,CPUs_data$Status=="End of Life
    ")
shapiro.test(End_of_Life_data$Recommended_Customer_Price)
qqnorm(End_of_Life_data$Recommended_Customer_Price)
qqline(End_of_Life_data$Recommended_Customer_Price)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  End_of_Life_data$Recommended_Customer_Price
## W = 0.50026, p-value < 2.2e-16
```

Figure 56: *Shapiro-Wilk normality test of Recommended Customer Price for End of life status*

- Null hypothesis: H_0: The recommended customer price of End of life status follows a normal distribution.
- Alternative hypothesis: H_1: The recommended customer price of End of life status does not follow a normal distribution.
- Since the p-value $< 2.2 \times 10^{-16}$ (which is less than the significance level of 5%), we reject the null hypothesis. Therefore, we conclude that the recommended customer price of servers does not follow a normal distribution.

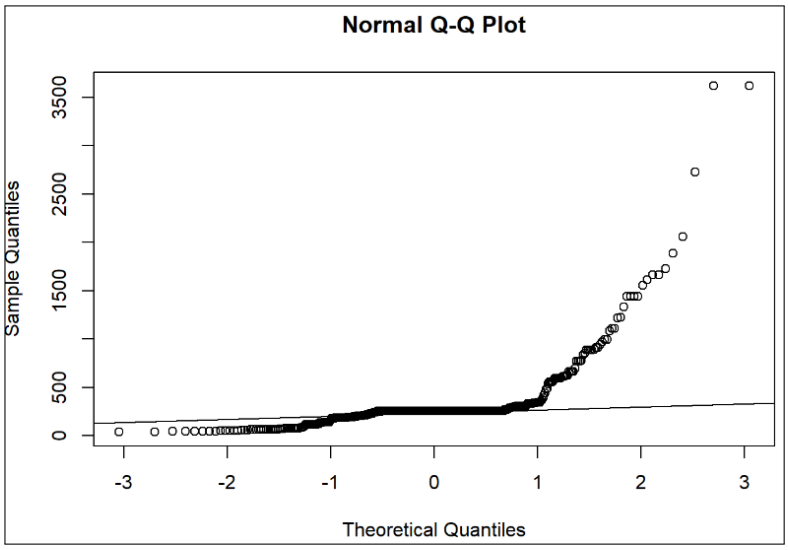

Figure 57: *Illustration of contribution of Recommended Customer Price for End of life status*

4. Launched:

```
Launched_data <- subset(CPUs_data,CPUs_data$Status=="Launched")
shapiro.test(Launched_data$Recommended_Customer_Price)
qqnorm(Launched_data$Recommended_Customer_Price)
qqline(Launched_data$Recommended_Customer_Price)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Launched_data$Recommended_Customer_Price
## W = 0.40321, p-value < 2.2e-16
```

Figure 58: *Shapiro-Wilk normality test of Recommended Customer Price for Launched status*

- Null hypothesis: H_0: The recommended customer price of Launcheds status follows a normal distribution.
- Alternative hypothesis: H_1: The recommended customer price of Launched status does not follow a normal distribution.
- Since the p-value $= 2,.2 \times 10^{-16}$ (which is less than the significance level of 5%), we reject the null hypothesis. Therefore, we conclude that the recommended customer price of embedded devices does not follow a normal distribution.

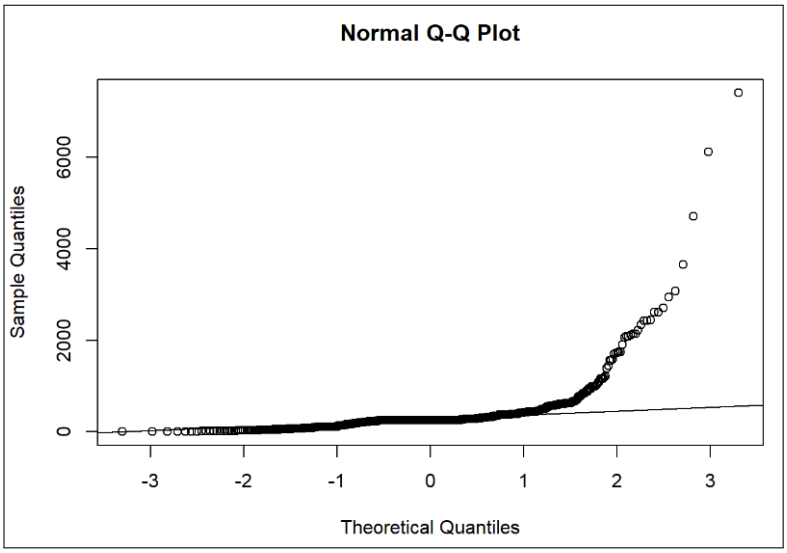

Figure 59: *Illustration of contribution of Recommended Customer Price for Launched status*

6.2.2 Equal variances

Vertical segment

To test whether the variances of the Recommended Customer Price differ among different vertical segments, we can use the Levene's test. Here's how we can formulate the hypotheses and interpret the results:
Hypotheses:

- Null hypothesis (H_0): The variances of Recommended Customer Price are equal across all vertical segments.
- Alternative hypothesis (H_1): At least two variances of Recommended Customer Price across different vertical segments are not equal.

```
LeveneTest(Recommended_Customer_Price~as.factor(Vertical_Segment),CPUs_data
    )
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value    Pr(>F)
## group    3  33.789 < 2.2e-16 ***
##       2279
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 60: *Leven's test on variances of vertical segment*

Interpretation:

- Since the p-value (2.2×10^{-16}) is less than the significance level of 5%, we reject the null hypothesis.
- Therefore, we conclude that there is evidence to suggest that at least two variances of Recommended Customer Price across different vertical segments are not equal.

This implies that the variance of Recommended Customer Price differs significantly among the different vertical segments.

Status

To test whether the variances of the Recommended Customer Price differ among different status, we can use the Levene's test. Here's how we can formulate the hypotheses and interpret the results:
Hypotheses:

- Null hypothesis (H_0): The variances of Recommended Customer Price are equal across all status.
- Alternative hypothesis (H_1): At least two variances of Recommended Customer Price across different status are not equal.

```r
filtered_data <- subset(CPUs_data,CPUs_data$Status %in% c("End of
    Interactive Support","End of Life","Launched"))
LeveneTest(Recommended_Customer_Price~as.factor(Status),filtered_data)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value    Pr(>F)
## group    2  42.315 < 2.2e-16 ***
##       2265
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 61: *Leven's test on variances of status*

Interpretation:

- Since the p-value (2.2×10^{-16}) is less than the significance level of 5%, we reject the null hypothesis.
- Therefore, we conclude that there is evidence to suggest that at least two variances of Recommended Customer Price across different status are not equal.

This implies that the variance of Recommended Customer Price differs significantly among the different status.

6.2.3 Independence of observations

All observations of each column are taken separately, which means they are independent.

6.2.4 Performing two-way ANOVA model

We observe that assumption of normality and equal variance are not satisfied, making it inappropriate to apply a two-way ANOVA model.
Assuming that all assumptions are met, the results of this two-way ANOVA model are only for reference purposes.

```r
anova_2_way_model <- aov(Recommended_Customer_Price~Vertical_Segment+Status,
    data=CPUs_data)
summary(anova_2_way_model)
```

```
##                   Df    Sum Sq Mean Sq F value Pr(>F)
## Vertical_Segment    3  23323800 7774600  61.689 <2e-16 ***
## Status              2   1328864  664432   5.272 0.0052 **
## Residuals        2262 285076313  126028
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 62: *Performing ANOVA model*

Vertical segment

Hypotheses:

- Null hypothesis (H_0): The mean of Recommended Customer Price is the same across all vertical segments.
- Alternative hypothesis (H_1): At least two vertical segments have different mean of Recommended Customer Price.

Interpretation:

- Since the p-value ($< 2 \times 10^{-16}$) is less than the significance level of 5%, we reject the null hypothesis.
- Therefore, we conclude that there is evidence to suggest that at least two vertical segments have different mean Recommended Customer Price.

Status

Hypotheses:

- Null hypothesis (H_0): The mean of Recommended Customer Price is the same across all status.
- Alternative hypothesis (H_1): At least two status have different mean of Recommended Customer Price.

Interpretation:

- Since the p-value 0.0052 is less than the significance level of 5%, we reject the null hypothesis.
- Therefore, we conclude that there is evidence to suggest that at least two status have different mean Recommended Customer Price.

6.2.5 Further analysis

To further clarify the differences indicated by H_1, we conduct a deeper analysis using the Tukey's HSD method.

```
1   TukeyHSD(anova_2_way_model)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Recommended_Customer_Price ~ Vertical_Segment + Status, data = filtered_data)
##
## $Vertical_Segment
##                       diff       lwr       upr     p adj
## Embedded-Desktop  24.14491 -54.08725 102.37708 0.8574414
## Mobile-Desktop    35.47089 -13.88655  84.82833 0.2513467
## Server-Desktop   236.24390 186.24877 286.23904 0.0000000
## Mobile-Embedded   11.32598 -65.42543  88.07738 0.9814066
## Server-Embedded  212.09899 134.93595 289.26203 0.0000000
## Server-Mobile    200.77301 153.12828 248.41774 0.0000000
##
## $Status
##                                           diff        lwr       upr     p adj
## End of Life-End of Interactive Support -21.82468 -71.559191  27.90984 0.5585007
## Launched-End of Interactive Support     34.13448  -5.120270  73.38923 0.1031490
## Launched-End of Life                    55.95916   8.400874 103.51744 0.0160826
```

Figure 63: *Tukey multiple comparisons of means*

DEEP ANALYSIS:

Vertical segment

- **Embedded-desktop**
  - Null hypothesis:
    H_0: The mean of Recommended Customer Price of embedded and desktop are equal.
  - Alternative hypothesis:
    H_1: The mean of Recommended Customer Price of embedded and desktop are different.
  - P-value = 0.8574414 > significance level (5%) => Accept H_0 => The mean of Recommended Customer Price of embedded and desktop are equal.
- **Mobile-desktop**
  - Null hypothesis:
    H_0: The mean of Recommended Customer Price of mobile and desktop are equal.
  - Alternative hypothesis:
    H_1: The mean of Recommended Customer Price of mobile and desktop are different.
  - P-value = 0.2513467 > significance level (5%) => Accept H_0 => The mean of Recommended Customer Price of mobile and desktop are equal.
- **Server-desktop**
  - Null hypothesis:
    H_0: The mean of Recommended Customer Price of server and desktop are equal.
  - Alternative hypothesis:
    H_1: The mean of Recommended Customer Price of server and desktop are different.
  - P-value = 0 < significance level (5%) => Reject H_0 => The mean of Recommended Customer Price of server and desktop are different. Moreover, the difference equals mean of Recommended Customer Price of server - mean of Recommended Customer Price of desktop = 236.24390 > 0 => The mean of Recommended Customer Price of server is greater than the mean of Recommended Customer Price of desktop.

- **Mobile-embedded**
  - Null hypothesis:
    H_0: The mean of Recommended Customer Price of mobile and embedded are equal.
  - Alternative hypothesis:
    H_1: The mean of Recommended Customer Price of mobile and embedded are different.
  - P-value = 0.9814066 > significance level (5%) => Accept H_0 => The mean of Recommended Customer Price of mobile and embedded are equal.
- **Server-embedded**
  - Null hypothesis:
    H_0: The mean of Recommended Customer Price of server and embedded are equal.
  - Alternative hypothesis:
    H_1: The mean of Recommended Customer Price of server and embedded are different.
  - P-value = 0 < significance level (5%) => Reject H_0 => The mean of Recommended Customer Price of server and embedded are different. Moreover, the difference equals mean of Recommended Customer Price of server - mean of Recommended Customer Price of embedded = 212.09899 > 0 => The mean of Recommended Customer Price of server is greater than the mean of Recommended Customer Price of embedded.
- **Server-mobile**
  - Null hypothesis:
    H_0: The mean of Recommended Customer Price of server and mobile are equal.
  - Alternative hypothesis:
    H_1: The mean of Recommended Customer Price of server and mobile are different.
  - P-value = 0 < significance level (5%) => Reject H_0 => The mean of Recommended Customer Price of server and mobile are different. Moreover, the difference = mean Recommended Customer Price of server - mean of Recommended Customer Price of mobile = 200.77301 > 0 => The mean of Recommended Customer Price of server is greater than the mean of Recommended Customer Price of mobile.

Comment:
Therefore, the recommended customer price of servers (Server) is greater than the recommended customer price of mobile devices (Mobile), which is equal to the recommended customer price of desktops (Desktop), and also equal to the recommended customer price of embedded systems (Embedded):

$$\text{Server} > \text{Mobile} = \text{Desktop} = \text{Embedded}$$

Status

- **End of Life - End of Interactive Support**
  - Null hypothesis:
    H_0: The mean of Recommended Customer Price of End of Life and End of Interactive are equal.

- Alternative hypothesis:
  H_1: The mean of Recommended Customer Price of End of Life and End of Interactive are different.
- P-value = 0.5585007 > significance level (5%) => Accept H_0 => The mean of Recommended Customer Price of server and embedded are equal

- **Launched - End of Interactive Support**
  - Null hypothesis:
    H_0: The mean of Recommended Customer Price of Launched and End of Interactive Support are equal.
  - Alternative hypothesis:
    H_1: The mean of Recommended Customer Price of Launched and End of Interactive Support are different.
  - P-value = 0.1031490 > significance level (5%) => Accept H_0 => The mean of Recommended Customer Price of Launched and End of Interactive Support are equal

- **Launched - End of Life**
  - Null hypothesis:
    H_0: The mean of Recommended Customer Price of Launched and End of Life are equal.
  - Alternative hypothesis:
    H_1: The mean of Recommended Customer Price of Launched and End of Life are different.
  - P-value =0.0160826 < significance level (5%) => Reject H_0 => The mean of Recommended Customer Price of Launched and End of Life are different. Moreover, the difference = mean Recommended Customer Price of Launched - mean of Recommended Customer Price of End of Life = 55.95916 > 0 => The mean of Recommended Customer Price of Launched is greater than the mean of Recommended Customer Price of End of Life.

  Comment:
  Therefore, the recommended customer price of products at the Launched stage (Launched) is greater than the recommended customer price of products at the End of Life stage (End of Life), which is equal to the recommended customer price of products at the End of Interactive Support stage (End of Interactive Support):

$$\text{Launched} > \text{End of Life} = \text{End of Interactive Support}$$

For illustration of Tukey comparisons, we can do as followings

```
plot(TukeyHSD(anova_2_way_model))
```

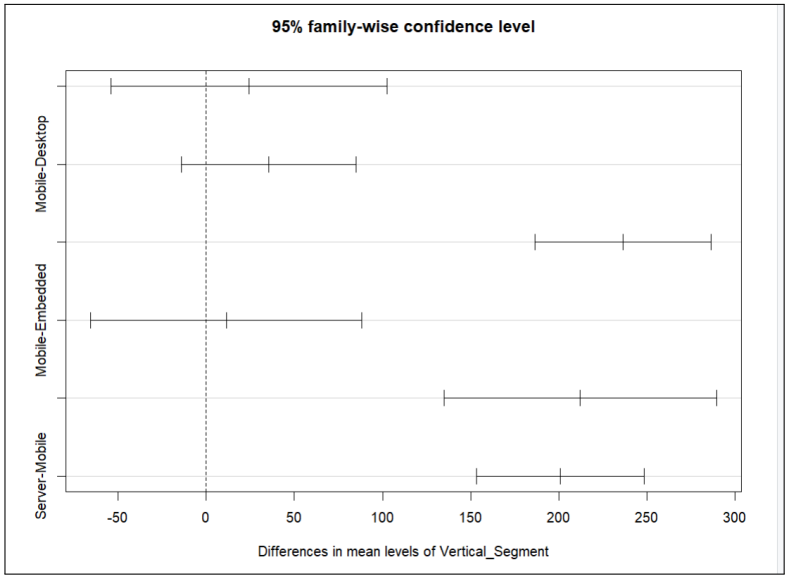

Figure 64: *Tukey multiple comparisons plot of means of vertical segment*

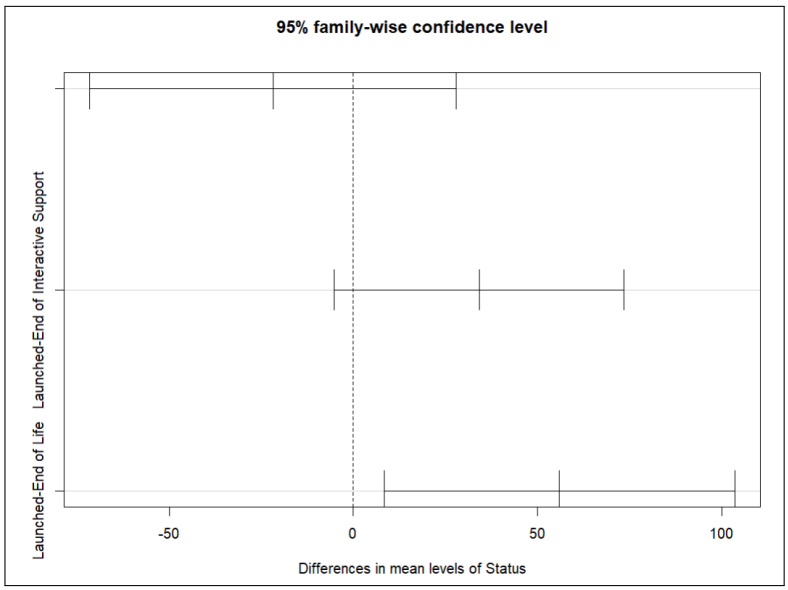

Figure 65: *Tukey multiple comparisons plt of means of status*

7 Discussion and Extension

7.1 Discussion

7.1.1 Advantages of Multiple Linear Regression (MLR)

1. Multiple Predictors: MLR allows us to model the relationship between multiple predictor variables and a response variable, which can provide a more complete picture of the relationship between variables than simple linear regression.
2. Predictive Power: MLR can be very effective for making predictions about a response variable based on multiple predictor variables.
3. Interpretability: MLR provides coefficients for each predictor variable, which can be interpreted to understand the relationship between each predictor and the response variable.
4. Handling of Quantitative and Qualitative Variables: MLR can handle both quantitative and qualitative predictor variables, which can be useful in a wide range of applications.

7.1.2 Disadvantages of Multiple Linear Regression (MLR)

1. Assumptions: MLR relies on several assumptions, such as linearity, independence, homoscedasticity, and normality. Violations of these assumptions can lead to incorrect inferences and predictions.
2. Overfitting: If we include too many predictor variables in our model, we can run the risk of overfitting the data, which can lead to poor performance on new data.
3. Multicollinearity: If our predictor variables are highly correlated, it can lead to unstable estimates of the regression coefficients.
4. Outliers: MLR can be sensitive to outliers, which can have a big impact on the estimates of the regression coefficients.
5. Non-linear Relationships: MLR assumes a linear relationship between the predictor variables and the response variable. If this assumption is violated, the model may not provide an accurate representation of the relationship between the variables.

7.1.3 Advantages of Analysis of Variance (ANOVA)

1. Comparing Multiple Groups: ANOVA allows researchers to compare means across three or more groups simultaneously, which can save time and reduce the chances of making type I errors.
2. Identifying Significant Differences: ANOVA helps identify whether there is a statistically significant difference between the means of the groups being compared.
3. Understanding Factor Effects: In experimental designs or observational studies with multiple independent variables, ANOVA enables researchers to understand the impact of each factor on the dependent variable.

4. Flexibility: ANOVA comes in various forms, such as one-way ANOVA, two-way ANOVA, and factorial ANOVA, which allows researchers to choose the appropriate model based on the complexity of their data and research question.

5. Provides Detailed Analysis: ANOVA provides a detailed breakdown of variances and interactions between variables which can be useful in understanding the underlying factors affecting the outcome.

7.1.4 Disadvantages of Analysis of Variance (ANOVA)

1. Assumptions: ANOVA relies on several assumptions, such as normality of the residuals, homoscedasticity of the residuals, and independence of observations. Violations of these assumptions can lead to incorrect inferences.

2. Post-hoc Tests: If the null hypothesis is rejected, researchers may need to conduct post-hoc tests to identify which specific group means differ significantly from one another, which can increase the risk of type I error (false positive conclusion).

3. Non-linear Relationships: ANOVA assumes a linear relationship between the predictor variables and the response variable. If this assumption is violated, the model may not provide an accurate representation of the relationship between the variables.

4. Outliers: ANOVA can be sensitive to outliers. A single extreme value in one group can affect the sum of squares and consequently influence the F-statistic and the overall result of the test.

5. Requires Larger Sample Sizes: To detect an effect of a certain size, ANOVA generally requires larger sample sizes than a t-test.

7.2 Extension

Missing data handling is a pivotal aspect of statistical analysis, exerting a profound influence on the outcomes and reliability of models. In the context of our selected dataset, which exhibits a substantial proportion of missing values (exceeding 10% of the total values), we address this challenge through mean and median imputation methods. Specifically, we apply these imputation techniques to ANOVA 2-Way and Multiple Linear Regression models. Furthermore, we introduce an alternative strategy wherein missing values are entirely excluded, creating a scenario akin to a dataset devoid of any missing values.
Our objective is to delve into how the choice of missing value handling methods impacts the outcomes of statistical models. By juxtaposing the results derived from imputed datasets against those obtained through the complete dataset approach, we aim to discern the nuanced effects on model outcomes stemming from various missing value handling techniques.
In essence, this study endeavors to offer comprehensive insights into the ramifications of missing data imputation strategies on the reliability and interpretability of statistical models. Through meticulous comparison and analysis, we strive to elucidate the most effective approaches for addressing missing data issues, thereby enhancing the robustness of statistical analyses in empirical research.

8 Code and data availability

The source code can be accessed here : Code_R
The source data can be accessed here: CPUs.csv

9 Conclusion

Under the expert guidance of Mr. Dung during our classes, we've adeptly navigated through the intricate realm of ANOVA model and Regression model, with a specific focus on the challenging domain of Probability and Statistics. This note serves as an expression of our deep gratitude for the insightful assignment that delved into the nuances of this complex field.

The assignment, centered around exploring the relationships between variables in real-world datasets, has been instrumental in broadening our understanding of statistical analysis and its practical applications. The thought-provoking nature of the tasks assigned has not only strengthened our grasp on stochastic programming but has also played a pivotal role in the development of critical skills crucial for our academic and professional journey.

The exposure to the complexities of analyzing data and drawing meaningful conclusions has proven invaluable, providing us with a profound comprehension of advanced statistical techniques. Your commitment to presenting stimulating assignments has been a catalyst for our academic and professional growth. We genuinely appreciate the effort you invest in guiding us through challenging yet rewarding tasks. Once again, thank you for this enriching and enlightening experience!

10 References

References

[1] Adam Hayes, December 20th, 2023. "Multiple Linear Regression (MLR) Definition, Formula, and Example". *Investopedia*. Available online: https://www.investopedia.com/terms/m/mlr.asp

[2] Rebecca Bevans, June 22nd, 2023. "Multiple Linear Regression | A Quick Guide (Examples)". *Scribbr*. Available online: https://www.scribbr.com/statistics/multiple-linear-regression/

[3] Andriy Blokhin, August 13th, 2023. "Linear vs. Multiple Regression: What's the Difference?". *Investopedia*. Available online: https://www.investopedia.com/ask/answers/060315/what-difference-between-linear-regression-and-multiple-regression.asp

[4] Muhammad Hassan, March 26th, 2024. "ANOVA (Analysis of variance) – Formulas, Types, and Examples". Available online: https://researchmethod.net/anova/

[5] Will Kenton, February 26th, 2024. "Analysis of Variance (ANOVA) Explanation, Formula, and Applications". *Investopedia*. Available online: https://www.investopedia.com/terms/a/anova.asp

[6] Zach Bobbitt, March 31st, 2021. "One-Way vs. Two-Way ANOVA: When to Use Each". Available online: https://www.statology.org/one-way-vs-two-way-anova/

[7] Wallstreetmojo Team, March 14th, 2024. "Regression vs ANOVA". *Wallstreetmojo*. Available online: https://www.wallstreetmojo.com/regression-vs-anova/

[8] Gianfranco, March 22nd, 2019. "What is the difference between ANOVA and regression (and which one to choose)". Available online: https://www.statsimprove.com/en/what-is-the-difference-between-anova-and-regression-and-which-one-to-choose/

[9] Cuemath. "Hypothesis Testing - Definition , Examples, Formula, Types". Available online: https://www.cuemath.com/data/hypothesis-testing/

[10] Donald E. Knuth. *The TEX Book*. Addison-Wesley Professional, 1986.

[11] Leslie Lamport. *LATEX: a Document Preparation System*. Addison Wesley, Massachusetts, 2nd edition, 1994.

[12] Frank Mittelbach, Michel Gossens, Johannes Braams, David Carlisle, and Chris Rowley. *The LATEX Companion*. Addison-Wesley Professional, 2nd edition, 2004.

[13] Helmut Kopka, Patrick W. Daly. *A Guide to LATEX and Electronic Publishing*. Addison-Wesley Long Man Limited, 4th edition, 2004.