VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY

HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY

FACULTY OF COMPUTER SCIENCE AND ENGINEERING

ANALYSIS ON PRICE AND COMPUTATION ABILITY OF CPUS

MT2013 - CC01 - SEMESTER 222

Supervisor: Mrs. Phan Thị Hường

Members:

Number	MSSV	First Name	Last Name	Sign
1	1952717	Lê Gia	Huy	
2	2153327	Nguyễn Hữu	Hào	
3	2153846	Phan Thành	Thông	
4	1952653	Phạm Thiên	Đăng	
5	2153110	Đỗ Thành	Vững	

HỒ CHÍ MINH CITY, May 18 2023

MEMBER LIST

Number	MSSV	First Name	Last Name	%Contribution	Note
1	1952717	Lê Gia	Huy	20	L
2	2153327	Nguyễn Hữu	Hào	20	
3	2153846	Phan Thành	Thông	20	
4	1952653	Phạm Thiên	Đăng	20	
5	2153110	Đỗ Thành	Vững	20	
Total				**100**	

Comment	Evaluation

Leader Information

Name: Lê Gia Huy

Email: huy.le0107@hcmut.edu.vn

Contents

I. Data Introduction

The dataset contains information about Intel processors. The data has a population of 2284 observations. Each sample has at most 38 parameters. The following Table summarily explains all the data types provides further information of them.

Column	Information given of the column
1. **Product_Collection**	the generation and model of the Intel processors
2. **Vertical_Segment**	whether the processor is for desktop, laptop, or server use
3. **Processor_Number**	the specific model number, such as i7-7Y75 or i5-8250U
4. **Status**	whether the processor is launched at end of life, or at the end of interactive support
5. **Launch_Date**	the quarter and year when the processor was initially released
6. **Lithography**	the manufacturing process technology, denoted in nanometers (nm), used to fabricate the processor, such as 14 nm, 22 nm, or 32 nm
7. **Recommended_Customer_Price**	the suggested price for the processor
8. **nb_of_Cores**	the number of independent processing units inside the processor
9. **nb_of_Threads**	how many tasks the processor can handle simultaneously
10. **Processor_Base_Frequency**	the starting frequency of the

	processor
11. Max_Turbo_Frequency	the highest frequency the processor can reach under turbo boost technology
12. Cache	a small amount of fast memory inside the processor that helps with performance, typically measured in megabytes (MB)
13. Bus_Speed	the speed at which the processor communicates with other components, denoted in gigatransfers per second (GT/s)
14. TDP	the Thermal Design Power, which tells us the maximum power the processor is designed to consume
15. Embedded_Options_Available	if there are special versions of the processor for specific uses
16. Conflict_Free	if the processor is made using materials free from conflict
17. Max_Memory_Size	the maximum amount of memory the processor can support
18. Memory_Types	the different types of memory the processor can work with, like LPDDR3 or DDR4
19. Max_nb_of_Memory_Channels	the maximum number of memory channels supported
20. Max_Memory_Bandwidth	the maximum data transfer rate supported by the memory

	subsystem
21. ECC_Memory_Supported	if the processor supports Error-Correcting Code (ECC) memory
22. Graphics_Base_Frequency	the starting frequency of the processor's integrated graphics
23. Graphics_Max_Dynamic_Frequency	the highest frequency it can reach
24. Graphics_Video_Max_Memory	the maximum amount of memory the graphics can use
25. Max_Resolution_HDMI	maximum resolutions supported by HDMI output
26. Max_Resolution_DP	maximum resolutions supported by DP output
27. Max_Resolution_eDP_Integrated_Flat_Panel	maximum resolutions supported by eDP output
28. DirectX_Support	the compatibility of the graphics unit with directX software
29. OpenGL_Support	the compatibility of the graphics unit with openGL software
30. PCI_Express_Revision	the version of the PCI Express technology used
31. PCI_Express_Configurations	the number of lanes available
32. T	the maximum temperature the processor can handle
33. Intel_Hyper_Threading_Technology	if the processor has a feature to improve multitasking
34. Intel_Virtualization_Technology_VTx	if the processor supports virtualization

35. Intel_64_	if the processor can handle 64-bit software
36. Instruction_Set	the types of software instructions the processor understands
37. Instruction_Set_Extensions	the types of software instructions the processor understands
38. Idle_States	if the processor can reduce power consumption

Table 1: Dataset's parameters and its information

Given the overall data of the dataset, our group subsequently classify 2 main factors of each processor: its computation ability and its price. We then consider the questions:

1. What is the conclusion we can reach from comparing all the processors' prices and computation abilities on the market?
2. How much each factor contributes to the price of each processor?
3. What is the future trends of the price of processors?

II. Background

1. ANOVA

1.1 Basic Concept of ANOVA

ANOVA, also known as Analysis of Variance, is a statistical method used to assess how the average value of a numerical variable varies based on the levels of two categorical variables. Specifically, a two-way ANOVA examines how two independent variables, when combined, impact a dependent variable.

In the context of ANOVA, a factor refers to the categorical variable being evaluated, and the various subdivisions within the factor are commonly referred to as levels or groups. The terms "1-way," "2-way," or "n-way" are used to indicate the number of factors being analyzed in the model.

1.2 How does the ANOVA test work?

ANOVA employs the F-test to evaluate statistical significance. The F-test compares the variance in group means to the overall variance in the dependent variable.

This comparison involves assessing two types of variances: the variance between groups and the variance within groups. The between-group variance is determined by comparing each group's mean to the overall mean, while the within-group variance measures the variation of each observation from its group mean.

To quantify these variances, sums of squares (SS) are calculated, summing the distances of each data point from the mean. The ratio of the between SS to the within SS yields the F-statistic, which serves as the test statistic in ANOVA.

Combining the F-statistic with the degrees of freedom (df) produces a p-value, which indicates statistical significance. This is the sought-after information in ANOVA analysis.

If the within-group variance is smaller than the between-group variance, the F-test will yield a higher F-value, indicating a greater likelihood that the observed difference is real and not due to chance.

In the case of n-way ANOVA with interaction, three null hypotheses are simultaneously tested:

There is no difference in group means across any level of the first independent variable.

There is no difference in group means across any level of the second independent variable.

The effect of one independent variable does not depend on the effect of the other independent variable (no interaction effect).

1.3 Levene Test for Homoscedasticity of Variance

Levene's test is a statistical method used to assess whether the variances of a variable are equal across two or more groups. It is employed in various statistical procedures that assume equal variances among the populations from which different samples are derived. Levene's test evaluates this assumption by examining the null hypothesis of homogeneity of variance, also known as homoscedasticity. This test compares the variances of multiple samples, with the number of samples (k) potentially exceeding two. Levene's test is an alternative to Bartlett's test and is less sensitive to deviations from normality. Given a variable (Y) with a sample size (N) divided into k subgroups, where Ni represents the sample size of the ith subgroup, the Levene test statistic is defined as follows:

$$W = \frac{(N-k)}{(k-1)} \frac{\sum_{i=1}^{k} N_i (\bar{Z}_{i.} - \bar{Z}_{..})^2}{\sum_{i=1}^{k} \sum_{j=1}^{N} (\bar{Z}_{ij} - \bar{Z}_{i.})^2}$$

Where N represents a total sample size and k represents the number of subgroups. The term "Zij" can take on one of three definitions, each affecting the robustness and power of the test:

Zij = Yij - Yi, where Yi is the mean of the ith subgroup.

Zij = Yij - Ŷi, where Ŷi is the median of the ith subgroup.

Zij = Yij - Yi', where Yi' is the 10% trimmed mean of the ith subgroup.

Here, Zi represents the group means of the Zij values, and Z represents the overall mean of the Zij values.

The choice of Zij definition in Levene's test influences how robust the test is in detecting unequal variances when the underlying data deviate from normal distribution and the variances are, in fact, equal. Power refers to the test's ability to detect unequal variances when they truly exist.

1.4 Tukey's Honestly Significant Difference (Tukey's HSD) post-hoc test

ANOVA helps determine if there are differences among group means, but it does not provide information about the specific nature of those differences. To identify which groups have statistically significant differences, a post-hoc test called Tukey's Honestly Significant Difference (Tukey's HSD) can be conducted.

Tukey's HSD requires an aov object as input and performs pairwise comparisons between all possible combinations of groups. It tests these pairs for significant differences in their means while adjusting the p-value to a higher threshold to account for multiple comparisons. This adjustment is necessary because conducting numerous statistical tests increases the chance of false positives.

Tukey's test compares the means of all treatments with the means of every other treatment and is considered the most suiTable method when confidence intervals are desired or when sample sizes are unequal.

The test statistic used in Tukey's test is denoted as q, which is essentially a modified t-statistic that corrects for multiple comparisons. The value of q can be calculated similarly to the t-statistic:

$$q_{\propto}, k, N - k$$

The studentized range distribution of q is defined as:

$$q_s = \frac{Y_{max} - Y_{min}}{SE}$$

Here, Ymax and Ymin represent the largest and smallest means of the two groups being compared, respectively. se denotes the standard error of the entire test.

2. Multiple Linear Regression Model

1.1 Definition

What is Multiple Regression?

Multiple regression, also known as multiple linear regression (MLR), is a statistical technique that uses two or more explanatory variables to predict the outcome of a response variable. It can explain the relationship between multiple independent variables against one dependent variable. These independent variables serve as predictor variables, while the single dependent variable serves as the criterion variable. You can use this technique in a variety of contexts, studies, and disciplines, including in econometrics and financial inference.

The multiple linear regression model supposes that the response y is related to the input values X_i, i = 1, ...n, through the relationship:

$$y = \alpha + \beta_1 X_1 + \cdots + \beta_n X_n + \varepsilon$$

, whereas

y: the predicted value of the dependent variable
α: the y-intercept
β, i = 1, …n: the regression coefficient of the i^{th} variable X_i
ε: model error

Test Hypothesis

As an example, to determine whether variable X_1 is a useful predictor variable in this model, we could test

$$H0: \beta 1 = 0$$
$$H1: \beta 1 \neq 0$$

If the null hypothesis above were the case, then a change in the value of X_1 would not change *y*, so *y* and X_1 are not linearly related. Also, we would still be left with variables X_2 and X_3 being present in the model. When we cannot reject the null hypothesis above, we should say that we do not need variable X_1 in the model given that variables X_2 and X_3 will remain in the model. In general, the interpretation of a slope in multiple regression can be tricky. Correlations among the predictors can change the slope values dramatically from what they would be in separate simple regressions.

1.2 MLR Parameter Test

For the simple linear regression model, there is only one slope parameter about which one can perform hypothesis tests. For the multiple linear regression model, there are three different hypothesis tests for slopes that one could conduct. They are:

- Hypothesis test for testing that all of the slope parameters are 0.
- Hypothesis test for testing that a subset more than one, but not all of the slope parameters are 0.
- Hypothesis test for testing that one slope parameter is 0.

1.3 Shapiro-Wilk test

The Shapiro-Wilk test is a statistical test used to determine whether a dataset is normally distributed or not. It tests the null hypothesis that a sample is drawn from a normal population and is widely used in many statistical analyses to assess the normality assumption.

The test calculates a test statistic and p-value based on the differences between the observed distribution and the expected distribution of a normal population. If the p-value is less than the significance level (typically 0.05), then the null hypothesis is rejected, indicating that the data is not normally distributed. On the other hand, if the p-value is greater than the significance level, then the null hypothesis is not rejected, indicating that the data may be normally distributed.

The Shapiro-Wilk test is widely used in many fields, including biology, engineering, social sciences, and finance, as the normal distribution is commonly used in many statistical models and analyses. However, it is important to note that the test may be sensitive to small deviations from normality, and it should be used in conjunction with other methods to assess the normality assumption.

The Shapiro–Wilk test tests the null hypothesis that a sample x1, ..., xn came from a normally distributed population. The test statistic is

$$(a_1, \ldots, a_n) = \frac{m^\mathsf{T} V^{-1}}{C},$$

where

• with parentheses enclosing the subscript index i is the ith order statistic, i.e., the th-smallest number in the sample (not to be confused with).

• is the sample mean.

The coefficients ai are given by:

$$(a_1, \ldots, a_n) = \frac{m^\mathsf{T} V^{-1}}{C},$$

where C is a vector norm

$$C = \|V^{-1} m\| = (m^\mathsf{T} V^{-1} V^{-1} m)^{1/2}$$

and the vector m:

$$m = (m_1, \ldots, m_n)^\mathsf{T}$$

is made of the expected values of the order statistics of independent and identically distributed random variables sampled from the standard normal distribution; finally, V is the covariance matrix of those normal order statistics.

There is no name for the distribution of W. The cutoff values for the statistics are calculated through Monte Carlo simulations.

1.4 Assumptions of multiple regression

In multiple regression analysis, there are several assumptions to meet for the results to be valid and reliable. These assumptions are as follows:

Linearity

A crucial assumption in multiple regression is that there exists a linear connection between the independent variables and the dependent variable. This implies that the relationship between the variables is anticipated to be a straight line rather than curved or non-linear. If the relationship deviates from linearity, the outcomes of the regression analysis may lack reliability.

Independence

Another assumption inherent in multiple regression is the independence of observations. This assumption states that the values of the independent variable have no influence on the values of the dependent variable. Each observation is considered to be independent of all other observations.

Homoscedasticity

Homoscedasticity refers to the assumption that the errors' variance remains constant across all levels of the independent variables. This assumption is crucial as it ensures that the residuals of the model, which represent the differences between predicted and actual values, exhibit equal variance. When this assumption is violated, the model is said to have heteroscedasticity, which can result in biased or inefficient estimates.

Normality

The errors of the regression model are assumed to be normally distributed. Normality implies that the errors adhere to a bell-shaped curve centered around zero, with a majority of the errors being close to zero. Normality is significant as it helps guarantee the reliability and unbiasedness of the results obtained from the regression analysis.

Multicollinearity

Multicollinearity arises when there is a high correlation between two or more independent variables. This situation can create challenges in multiple regression analysis, as it becomes problematic to discern the individual independent effects of each variable on the dependent variable. Furthermore, multicollinearity can introduce instability in the regression coefficients, rendering the interpretation of the analysis results difficult.

1.5 Interpreting Diagnostic Plots in R

2.2.1 Residuals vs. Leverage Plot

A Residuals vs. Leverage plot is a type of diagnostic plot that allows us to identify influential observations in a regression model.

Here is how this type of plot appears in the statistical programming language R:

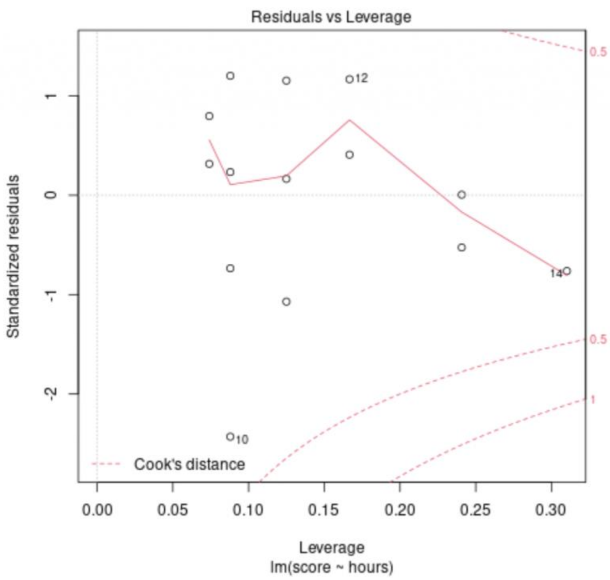

Figure 1: Example in R

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.

Standardized residuals refer to the standardized difference between a predicted value for an observation and the actual value of the observation.

It's worth noting that an observation can have a high absolute value for a standardized residual yet have a low value for leverage.

2.2.2 *Scale – Location Plot*

A scale-location plot is a type of plot that displays the fitted values of a regression model along the x-axis and the the square root of the standardized residuals along the y-axis.

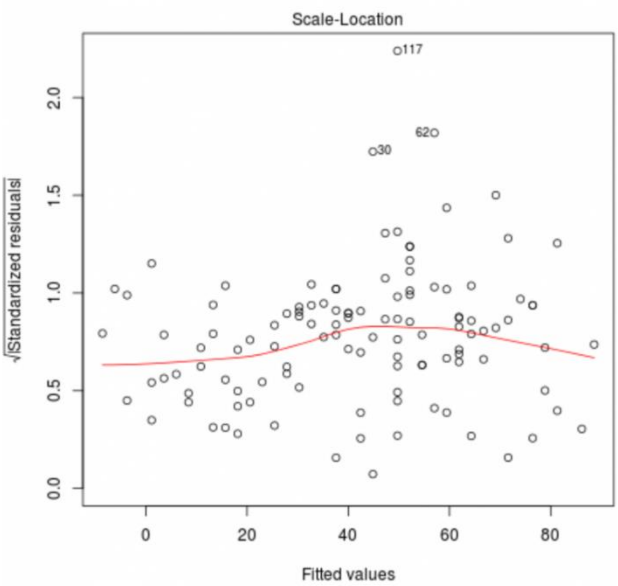

Figure 2: Example in R

When looking at this plot, we check for two things:

- Verify that the red line is roughly horizontal across the plot. If it is, then the assumption of homoscedasticity is likely satisfied for a given regression model. That is, the spread of the residuals is roughly equal at all fitted values.
- Verify that there is no clear pattern among the residuals. In other words, the residuals should be randomly scattered around the red line with roughly equal variability at all fitted values.

2.2.3 *Normal Q-Q Plot*

A Q-Q plot, short for "quantile-quantile" plot, is used to assess whether or not a set of data potentially came from some theoretical distribution.

In most cases, this type of plot is used to determine whether or not a set of data follows a normal distribution.

If the data is normally distributed, the points in a Q-Q plot will lie on a straight diagonal line.
Conversely, the more the points in the plot deviate significantly from a straight diagonal line, the less likely the set of data follows a normal distribution.

2.2.4 Residuals vs. Fitted Plot

The Residual vs Fitted plot allows us to detect several types of violations in the linear regression assumptions.
In the plot, the fitted values $\hat{y}$ is sketched on the x-axis and the residuals $y-\hat{y}$ are represented on the y-axis. The Residuals vs. Fitted plot is mainly useful for investigating:

- Whether Linearity holds. This is indicated by the mean residual value for every fitted value region being close to 0. In R this is indicated by the red line being close to the dashed line.
- Whether Homoscedasticity holds. The spread of residuals should be approximately the same across the x-axis.
- Whether there are outliers. This is indicated by some ’extreme’ residuals that are far from the rest.

III. Descriptive Statistics

1. Import Data

The built in read.csv("\\Intel_CPUs.csv") function call reads the data in ("...") as a data frame and assign the data frame to a variable (using <-) so that it is stored in R’s memory.

Then, we use function "head(Intel_CPUs)" to view the top few rows of a data frame Intel_CPUs. It allows you to quickly inspect the structure and contents of your data. By default, head() displays the first six rows of the object.

2. Data Cleaning

2.1 <u>Remove unused features</u>

In this part, the subset() function is a function used to create a subset of a data frame or a vector based on certain conditions. W use it to choose 8 columns that is necessary for this project in order to analyze and summary. It contains “Vertical_Segment”, “Launch_Date”, “Lithography”, “Recommended_Customer_Price”, “nb_of_Cores”, “nb_of_Threads”, “Processor_Base _Frequency”, “TDP” which affects the function of an

Intel chip when it is released to the market.

2.2 *Handling missing*

The na.omit() function removes all incomplete cases of a data object (IntelCPUs) so it will give us an object with data has been cleaned all NA values as well as the rows having not available value (clean_NA_c4).

However, the data is still remained the string "N/A" that the function na.omit() can Figure by it own. Thus, we will use subset() to clear all the "N/A" format in the data frame.

Because in frequency, there are 2 different units (MHz and GHz), we proceed to convert them into the same unit, namely GHz. We then subsequently convert "Lithography", "Recommended_Customer_Price", "nb_of_Cores", "nb_of_Threads", "Processor_Base _Frequency", "TDP" columns into numeric. There are also some entries that have wrong format in the dataset, we use gsub() command to correct the data.

2.3 *Data summary*

To descriptive statistics for each of the variables from data given, we use summary() function. The summary() function returns the value that depends on the class of its argument, in this situation (the summary() of a vector) it give us descriptive statistics such as the minimum (Min.), the 1st quantile (1st Qu.), the median, the mean, the 3rd quantile (3rd Qu.), and the maximum (Max.) value of our input data as Figure below in the console window.

```
> summary(sep_Intel)
 Vertical_Segment Quarter        Year       Lithography     Recommended_Customer_Price
 Desktop :269     Q1:354    13     :193   Min.   :14.00   Min.   :    9.62
 Embedded:155     Q2:329    14     :188   1st Qu.:14.00   1st Qu.:  182.00
 Mobile  :267     Q3:358    15     :170   Median :22.00   Median :  304.00
 Server  :445     Q4: 95    17     :160   Mean   :22.44   Mean   :  880.92
                            12     :130   3rd Qu.:22.00   3rd Qu.:  774.00
                            16     :102   Max.   :65.00   Max.   :13011.00
                            (Other):193
  nb_of_Cores      nb_of_Threads    Processor_Base_Frequency      TDP
 Min.   : 1.000   Min.   : 1.000   Min.   :0.400             Min.   :  2.20
 1st Qu.: 2.000   1st Qu.: 4.000   1st Qu.:2.000             1st Qu.: 35.00
 Median : 4.000   Median : 8.000   Median :2.400             Median : 54.00
 Mean   : 5.281   Mean   : 9.624   Mean   :2.468             Mean   : 65.42
 3rd Qu.: 6.000   3rd Qu.:12.000   3rd Qu.:3.000             3rd Qu.: 95.00
 Max.   :28.000   Max.   :56.000   Max.   :4.300             Max.   :205.00

> head(sep_Intel)
# A tibble: 6 × 9
  Vertical_Segment Quarter Year  Lithography Recommended_Customer_Pr…¹ nb_of_Cores nb_of_Threads
  <fct>            <fct>   <fct>       <dbl>                     <dbl>       <dbl>         <dbl>
1 Mobile           Q3      16             14                       393           2             4
2 Mobile           Q3      17             14                       297           4             8
3 Mobile           Q3      17             14                       409           4             8
4 Desktop          Q1      12             32                       305           4             8
5 Mobile           Q1      17             14                       281           2             4
6 Mobile           Q1      15             14                       107           2             2
# ℹ abbreviated name: ¹Recommended_Customer_Price
# ℹ 2 more variables: Processor_Base_Frequency <dbl>, TDP <dbl>
```

Figure 3: The result for summary

3. Plotting

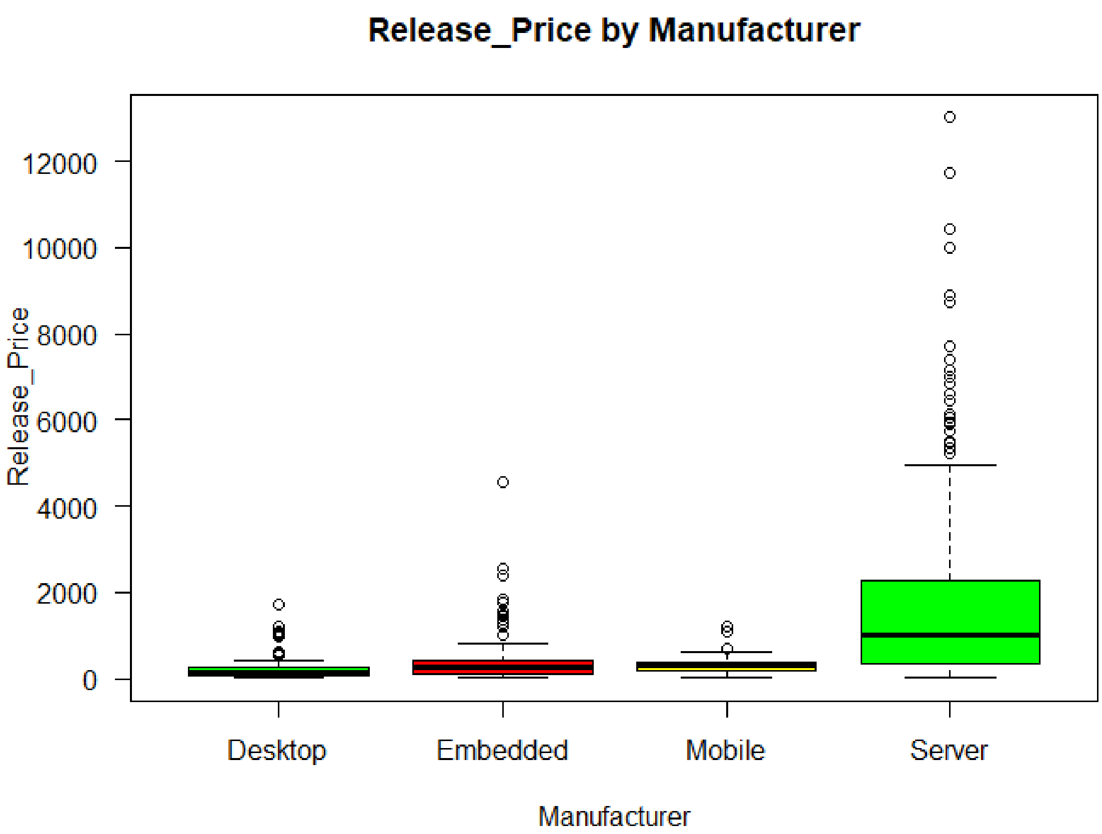

Figure 4: Initial prices of all processors (categorized)

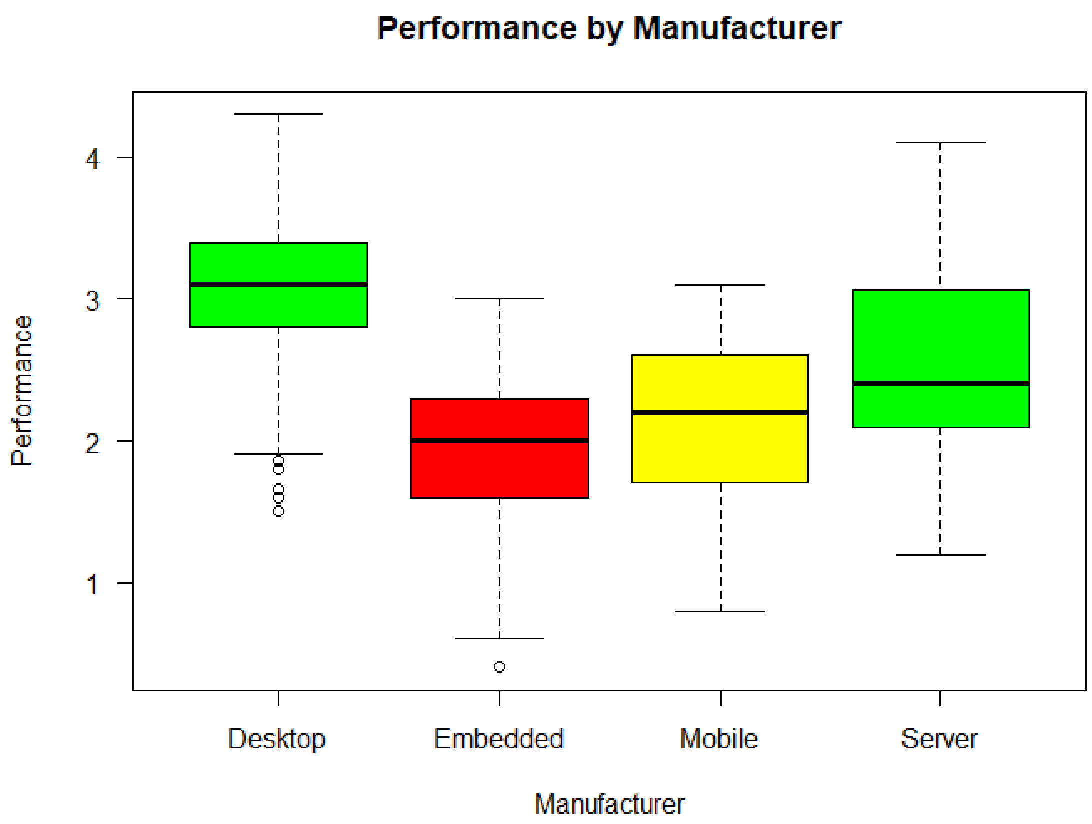

Figure 5: Base frequency of all processors (categorized)

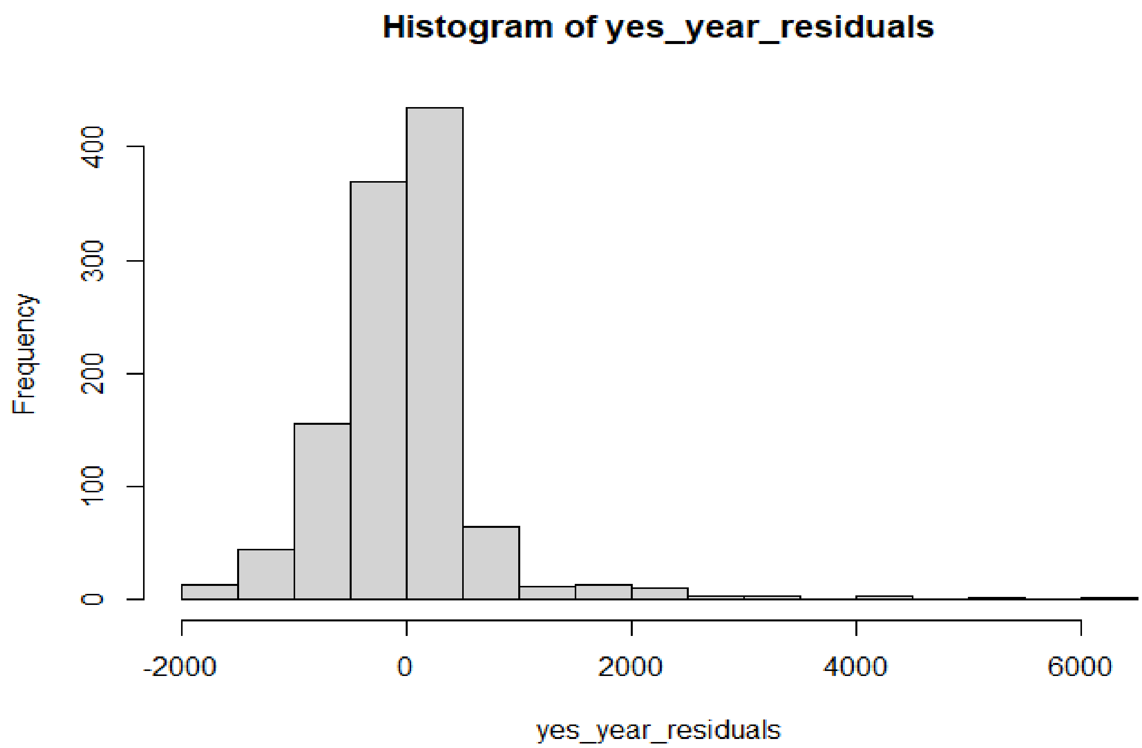

Figure 6: Histogram of residuals of the multiple linear regression model

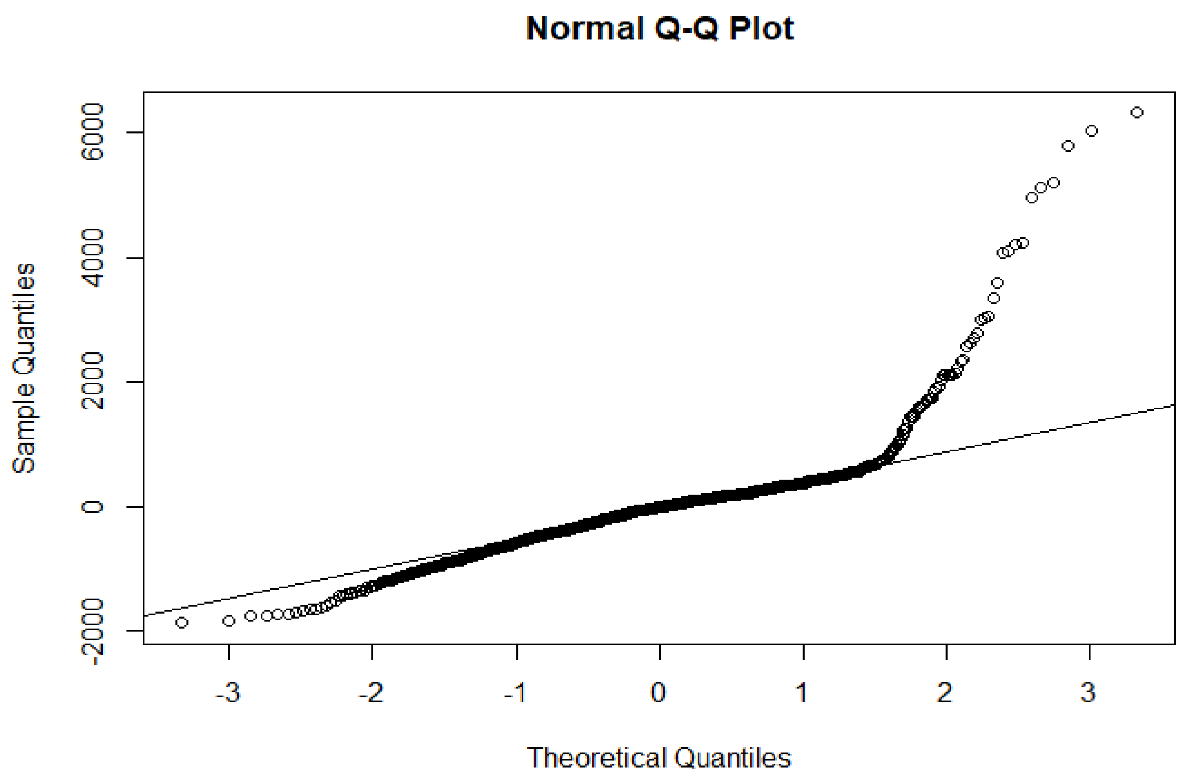

Figure 7: Q-Q plot and residuals

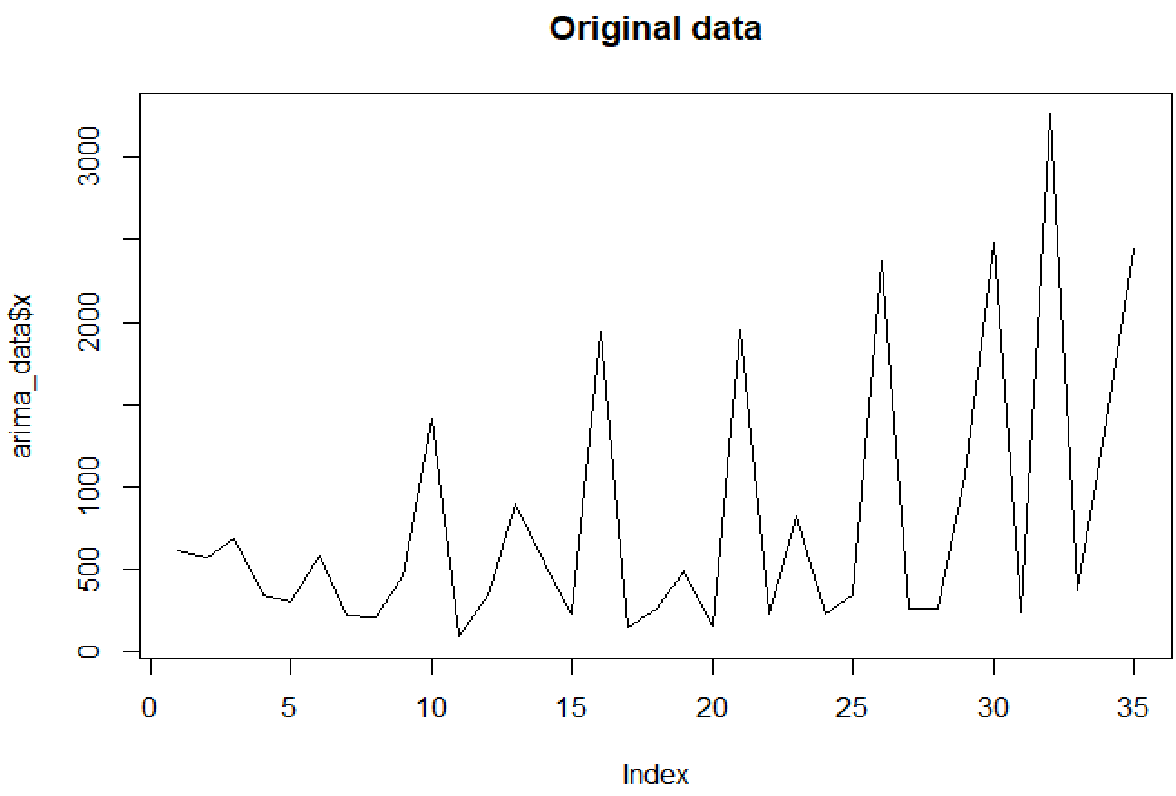

Figure 8: Mean price of all processors based on each quarter released

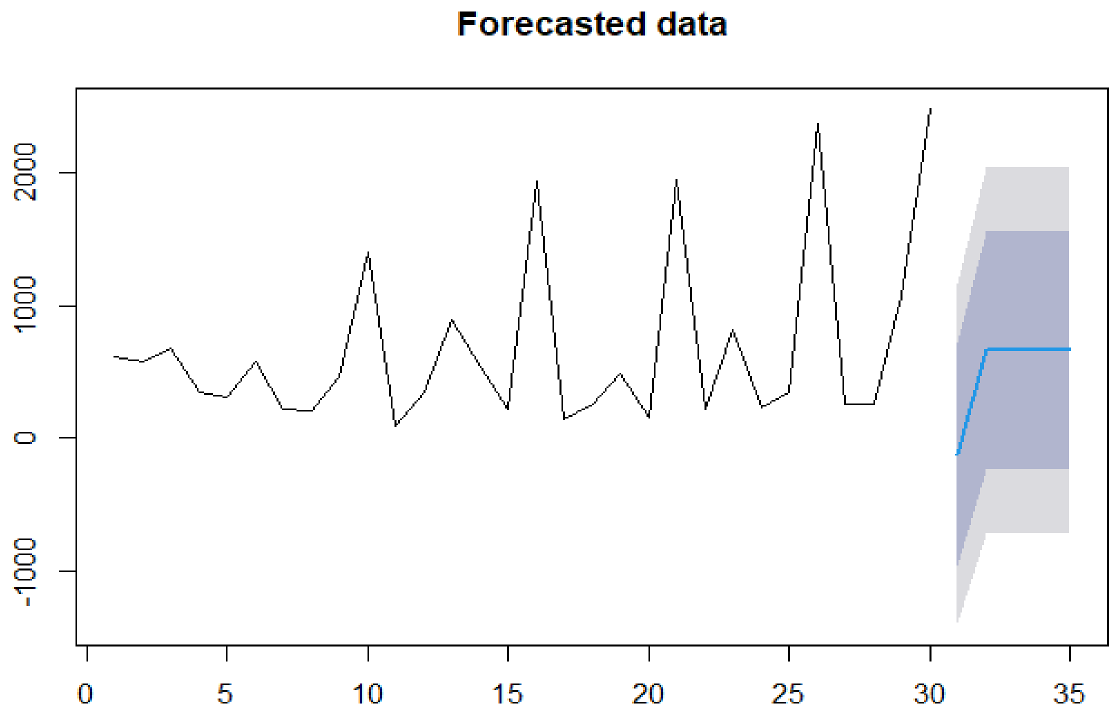

Figure 9: Forecasted processors' mean prices on 5 last time stamps

IV. Inferential Statistics

1 Chip Comparison

1.1 Price

1.1.1 Hypothesis

$H_0: \delta_{Price_{Desktop}}^{2} = \delta_{Price_{Server}}^{2} = \delta_{Price_{Embedded}}^{2} = \delta_{Price_{Mobile}}^{2}.$

$H_1: \delta_{Price_i}^{2} \neq \delta_{Price_j}^{2}$ for any i, j being in the group of 4 processor types.

We then consider the follow up question: Which processor type is the most expensive, which one is the cheapest?

1.1.2 ANOVA test

```
                               Df    Sum Sq    Mean Sq F value Pr(>F)    
sep_Intel$Vertical_Segment     3 6.07e+08 202341132   116.9 <2e-16 ***
Residuals                   1132 1.96e+09   1731379                   
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 10: ANOVA test on prices of 4 types

As we can see, the "***" characters denote that the test statistic conforms to the significant code of 0, meaning that we cannot reject H_0.

1.1.3 Tukey HSD test

Consider the follow up question, to determine the most expensive and cheapest processor type, we use Tukey's HSD (honest significance difference) test.

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Recommended_Customer_Price ~ sep_Intel$Vertical_Segment, data = sep_Intel)

$`sep_Intel$Vertical_Segment`
                       diff        lwr       upr     p adj
Embedded-Desktop  227.70438 -113.6863  569.0951 0.3157023
Mobile-Desktop     82.61628 -209.8403  375.0729 0.8863817
Server-Desktop   1574.21778 1312.7587 1835.6769 0.0000000
Mobile-Embedded  -145.08810 -486.9459  196.7697 0.6945496
Server-Embedded  1346.51340 1030.7655 1662.2613 0.0000000
Server-Mobile    1491.60150 1229.5328 1753.6702 0.0000000
```

Figure 11: Tukey HSD test on prices of 4 types

We can see that the Server processor type is the most expensive type because all 3 comparisons involving it far outweigh the others. Desktop processors tend to have the lowest prices, but Embedded processors have the smallest lower range.

1.2 Processor Frequency

1.2.1 Hypothesis

Do the frequencies of 4 processor types have the same variation?

$$H_0: \delta_{Freq}{}^2_{Desktop} = \delta_{Freq}{}^2_{Server} = \delta_{Freq}{}^2_{Embedded} = \delta_{Freq}{}^2_{Mobile}.$$

$H_1: \delta_{Freq}{}^2_{i} \neq \delta_{Freq}{}^2_{j}$ for any i, j being in the group of 4 processor types.

This section's follow up question is: Which processor type has the highest base frequency, and which has the least?

1.2.2 ANOVA test

```
                            Df Sum Sq Mean Sq F value Pr(>F)    
sep_Intel$Vertical_Segment    3  167.3   55.75   168.9 <2e-16 ***
Residuals                  1132  373.7    0.33                   
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 12: ANOVA test on frequencies of 4 types

Again, we see "***" characters denote that the test statistic conforms to the significant code of 0, meaning that we cannot reject H_0.

1.2.3 Tukey HSD test

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = sep_Intel$Processor_Base_Frequency ~ sep_Intel$Vertical_Segment, data = sep_Inte
l)

$`sep_Intel$Vertical_Segment`
                       diff         lwr        upr     p adj
Embedded-Desktop -1.1347464 -1.28381402 -0.9856787 0.0000000
Mobile-Desktop   -0.8992927 -1.02699334 -0.7715921 0.0000000
Server-Desktop   -0.5122440 -0.62640965 -0.3980784 0.0000000
Mobile-Embedded   0.2354537  0.08618207  0.3847253 0.0003078
Server-Embedded   0.6225024  0.48463158  0.7603731 0.0000000
Server-Mobile     0.3870487  0.27261687  0.5014805 0.0000000
```

Figure 13: Tukey HSD test on frequencies of 4 types

The result shows that Desktop processors tend to have higher frequency than the other 3, ranging from 0.5 to 1.3 GHz. On the contrary, Embedded processors tend to have the lowest frequencies.

2 Linear Regression: Upcoming Processor Trend

2.1 Hypothesis

Consider the question: “What factors determine a processor’s price?”

Having laid out the question, the next part is to determine what factors should be included in the analysis. Having as much data fitted into the model is good but the number of entries that have full data values on all columns are small. Therefore, we decided to apply Multiple Linear Regression on the following factors: Processor’s base frequency, lithography, number of cores, number of threads, TDP, their vertical segment (processor type), the quarter and year they were released.

2.2 Model Fitting

```
> summary(price_mlr_yes_year_model)

Call:
lm(formula = Recommended_Customer_Price ~ Processor_Base_Frequency +
    Lithography + nb_of_Threads + nb_of_Cores + TDP + Quarter +
    Year + Vertical_Segment, data = sep_Intel)

Residuals:
    Min      1Q  Median      3Q     Max
-1873.5  -387.5   -14.3   245.7  6317.4

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)              -2129.149    634.061  -3.358 0.000812 ***
Processor_Base_Frequency  -135.442     59.328  -2.283 0.022621 *
Lithography                 19.015      7.911   2.404 0.016392 *
nb_of_Threads               76.251     13.157   5.795 8.87e-09 ***
nb_of_Cores                 57.371     27.381   2.095 0.036368 *
TDP                         11.060      1.260   8.776  < 2e-16 ***
QuarterQ2                  111.786     65.658   1.703 0.088929 .
QuarterQ3                 -120.341     64.815  -1.857 0.063619 .
QuarterQ4                    9.693    101.364   0.096 0.923832
Year09                     259.050    502.474   0.516 0.606271
Year10                     783.492    487.000   1.609 0.107941
Year11                     938.501    491.356   1.910 0.056386 .
Year12                     879.847    498.380   1.765 0.077768 .
Year13                    1108.496    504.409   2.198 0.028182 *
Year14                    1124.888    507.766   2.215 0.026936 *
Year15                    1351.010    520.617   2.595 0.009583 **
Year16                     900.998    536.576   1.679 0.093401 .
Year17                    1216.463    529.342   2.298 0.021743 *
Vertical_SegmentEmbedded   105.840     96.027   1.102 0.270617
Vertical_SegmentMobile     459.357     85.889   5.348 1.08e-07 ***
Vertical_SegmentServer     -56.795     77.203  -0.736 0.462100
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 798.7 on 1115 degrees of freedom
Multiple R-squared:  0.7229,    Adjusted R-squared:  0.718
F-statistic: 145.5 on 20 and 1115 DF,  p-value: < 2.2e-16
```

Figure 14: Fitting multiple linear regression model

From Figure 14, we see that most predictors have significant value of more or equal to 0.05 and all of them have significant value of at least 0.1.

2.3 Confidence Intervals

```
> confint(price_mlr_yes_year_model)
                                  2.5 %      97.5 %
(Intercept)               -3373.236884 -885.061831
Processor_Base_Frequency   -251.849002  -19.035228
Lithography                   3.493527   34.536858
nb_of_Threads                50.435170  102.066908
nb_of_Cores                   3.647564  111.093960
TDP                           8.587079   13.532812
QuarterQ2                   -17.040441  240.613185
QuarterQ3                  -247.513481    6.831668
QuarterQ4                  -189.191750  208.578602
Year09                     -726.850391 1244.950584
Year10                     -172.048527 1739.031987
Year11                      -25.584853 1902.587570
Year12                      -98.021602 1857.716278
Year13                      118.797319 2098.194384
Year14                      128.603799 2121.171353
Year15                      329.511442 2372.509518
Year16                     -151.814307 1953.809522
Year17                      177.843562 2255.082473
Vertical_SegmentEmbedded    -82.574071  294.254116
Vertical_SegmentMobile      290.835129  627.878613
Vertical_SegmentServer     -208.275017   94.685855
```

Figure 15: Confidence intervals of variables of multiple linear regression model

2.4 Assumption Check

The key assumptions when build multiple linear regression model is that:

1. The residual values are normally distributed.

From Figure 6, we see that the histogram skewed to the right, therefore, we cannot conclude normality of the residuals.

From Figure 7, we can see that few portions of the residuals lie in a straight line. We can then assume that the residuals of the model do not follow a normal distribution.

2. Multicollinearity

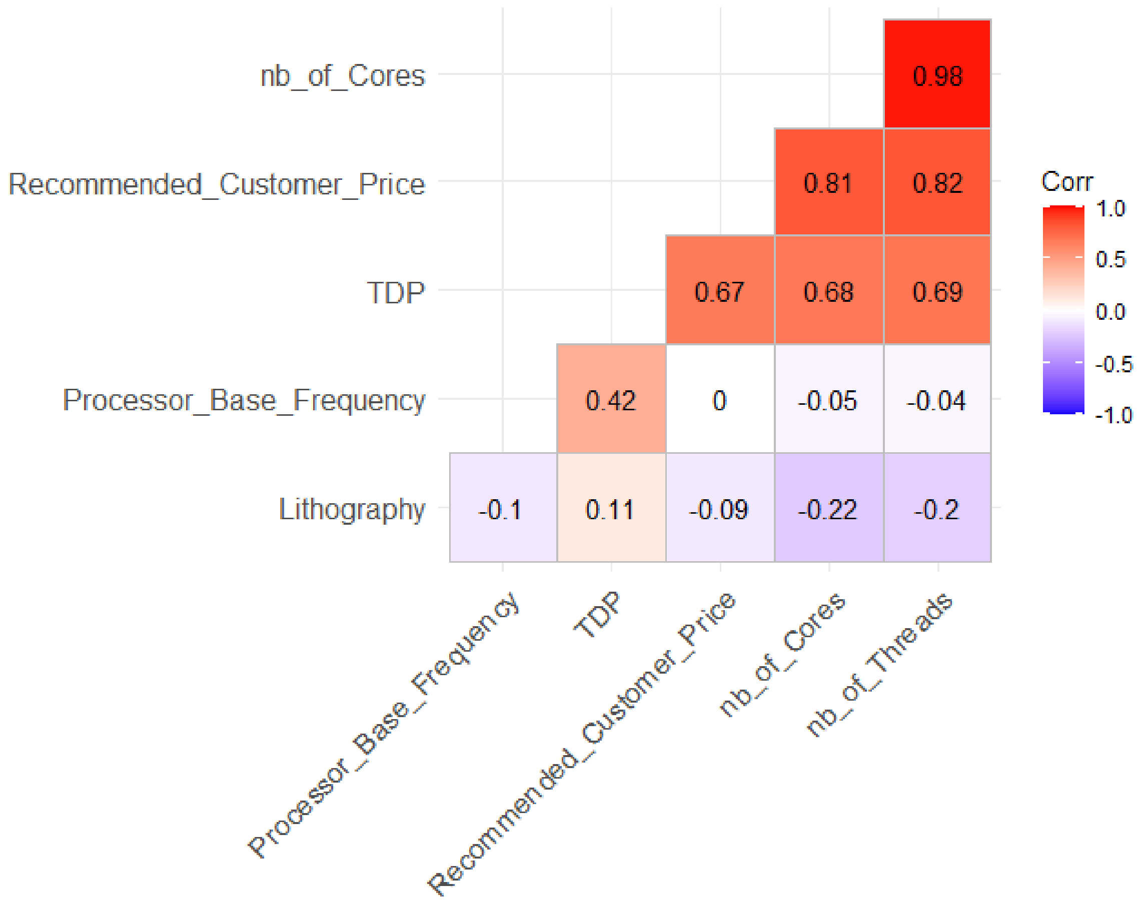

Figure 16: Multicollinearity between some variables of the dataset

From Figure 16, we see that there is evidence of multicollinearity, largely between the number of cores and number of threads; released price and number of cores and threads (>=0.8).

3. Homoscedasticity

```
> shapiro.test(yes_year_residuals)

        Shapiro-Wilk normality test

data:  yes_year_residuals
W = 0.77514, p-value < 2.2e-16
```

Figure 17: Shapiro-Wilk normality test on residuals

By using Shapiro-Wilk normality test on residuals, as in Figure 17, we can see there is homoscedasticity of the model.

2.5 Accuracy Check

```
> # Check accuracy
> sigma(price_mlr_yes_year_model)/mean(sep_Intel$Recommended_Customer_Price)
[1] 0.9066381
```

Figure 18: Accuracy checking for multiple linear regression model

2.6 Prediction

Suppose we want to estimate the price of a processor that has the specifications:

Requirement	Value
Base frequency	3.2 GHz
Lithography	14 nm
Thread number	4
Core number	8
TDP	130 W
Quarter released	4
Year released	2017
Vertical segment	Desktop

Table 2: Specifications for predicting processor price using MLR model

```
> # Make prediction
> pred_data<-data.frame(Processor_Base_Frequency=c(3.2),Lithography=c(14),nb_of_Threads=c(4),nb_of_
Cores=c(8),TDP=c(130),Quarter=c("Q4"),Year=c("17"),Vertical_Segment=c("Desktop"))
> predict(price_mlr_yes_year_model,pred_data)
       1
1131.568
```

Figure 19: Predicted price

From Figure 19, we see the model predicted that the processor following the specifications of Table 2 will have an initial price of $1131.568.

3 Times Series

3.1 Hypothesis

Consider the questions: “Is there a seasonal correspondence between the price of chips’ released?” To answer that, we use ARIMA model to predict the times series.

3.2 Stationary Testing

```
> print(adf.test(arima_data$x))

        Augmented Dickey-Fuller Test

data:  arima_data$x
Dickey-Fuller = -3.0448, Lag order = 3, p-value = 0.1674
alternative hypothesis: stationary
```

Figure 20: ADF Test to test stationary of the time series

3.3 Model Fitting

```
> # Fit model
> AutoArimaModel=auto.arima(arima_data[1:30,]$x)
> AutoArimaModel
Series: arima_data[1:30, ]$x
ARIMA(0,0,1) with non-zero mean

Coefficients:
          ma1      mean
      -0.4025  658.7329
s.e.   0.1887   71.4876

sigma^2 = 427353:  log likelihood = -236.1
AIC=478.2   AICc=479.13   BIC=482.41
```

Figure 21: Fitting ARIMA model on the time series using auto.arima command

Using auto.arima command, we can see the appropriate model is with p=0 and q=1.

4 Summary

We first move to the comparisons between processors. The ANOVA test concludes that there is no difference between mean prices and frequencies between processor types. However, there is substantial differences between the range of each processor type. Figure 3 shows price range for 4 processor types and it supports the conclusion we reached from analyzing Tukey's HSD test. It also highlights the fact that Server processor type has a much larger price variation than the other 3. Figure 4 also confirms the conclusion reach from Tukey HSD on frequencies of all processors.

Comparing it to real life, first regarding the price of the 4 processor types we compared earlier (which exclude Quantum Computer Processors), the most expensive Server CPU (which is a particular type of processor) is the Intel Xeon Platinum 8280L, which has a price of $36,718 and a base frequency of 2.7GHz[1], both of which confirms our answer in the first section and the assumption of Servers' clock rate range. On the other hand, the Padauk PFS173 is the cheapest processor at $0.09[2], which fits in the Embedded type, confirming the assumption that Embedded has the lowest lower range.

On the topic of Multiple Linear Regression, we predicted that the most significant factor contributing to price of a released chip is the time in which they are released, in particular, the year in which it is released, not the quarter. The processor's base frequency, on the other hand, lessens the price, which means the higher clock rate, the cheaper the processor.

On the topic of the times series question, we can see the predicted price actually corresponds with the actual price. With an initial drop, the price however increased steeply, and the actual price lies within the range of the predicted price. We can, moreover, confirm that there is a seasonal change in the price of the chip released.

V. Discussion and Extension

1. Chip Comparison

Our way of classifying processors into different groups (Embedded, Desktop, Mobile, Server) and then comparing the difference in price and computing capabilities are in fact the Analysis of Variance. A large problem is that the box plot does not keep the exact values and details of the distribution results, which is an issue with handling such large amounts of data in this graph type. A box plot shows only a simple summary of the distribution of results, so that it you can quickly view it and compare it with other data.

However, there are many alternative methods such as expanding our question to test more features in the dataset, given that there are many untouched features in our assignment. Correct analysis for each feature will help approximate cost for each feature. Other than that, we can use group together as many common features as possible and then proceed to compare it with other groups.

2. Multiple Linear Regression

[1] Goodley A., (09/01/2022), *8 Most Expensive CPUs in the Market Today (2022)*, From https://rarest.org/stuff/expensive-cpus

[2] Jenny L., (28/08/2019), *Everything You Want to Know About the Cheapest Processors Available*, From https://hackaday.com/2019/08/28/everything-you-want-to-know-about-the-cheapest-processors-available/

An alternative way of answering the factors that contributed to the price of processors is applying multiple linear regression on a much larger scale, namely all 45 columns of the initial dataset. However, this approach released extensive data cleaning and risk losing a lot of data because the number of entries that have at least 1 not available data is pretty large.

3. Times Series

Even though we have predicted an accuracy answer to the actual price, we can also consider alternative approaches. Namely, we can use ARMA, smooth-based or manually applying ARIMA on p and q, rather than letting R do it automatically.

VI. Code and Data Availability

- Code:
  1. https://github.com/huyle0107/Probability-Statistic_Assignment_HK222.git
  2. https://drive.google.com/file/d/1WSztSLlZrnGHKkvnQx40p2yli7rG-WWO/view?usp=sharing
- Dataset:
  https://www.kaggle.com/datasets/iliassekkaf/computerparts?resource=download&fbclid=IwAR3oVQyWuLZAOI_3w1ykkdT4sShaaoQ0I7ZPm-qhyGj0-sSSefkKBeWxpHE

VII. References

A. Sheldon M. Ross, (2010), *Introductory Statistics*, Publisher Academic Press, US

B. Douglas C. Montgomery, George C. Runger, (2013), *Applied Statistics and Probability for Engineers*, Publisher Wiley, US

C. Peter Dalgaard, (2008), *Introductory Statistics with R*, Publisher Springer Science + Business Media, NY, US

D. Alain F. Zuur, Elena N. Ieno, Erik H.W.G Meesters, (2009), *A Beginner's Guide to R*, Publisher Springer Science + Business Media, NY, US