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

PROBABILITY AND STATISTICS (MT2013)

Assignment

Computer Parts: GPUs Analysis Project

Instructor(s):

Nguyễn Tiến Dũng, Applied Science - HCMUT

Team name:

Faculty of Computer Science and Engineering
Class CC08 - Group 08 - Semester 232

Student(s):

Trịnh Anh Minh - 2252493
Cao Ngọc Lâm - 2252419
Hồ Khánh Nam - 2252500
Nguyễn Châu Hoàng Long - 2252444
Đoàn Viết Tiến Đạt - 2252141

HO CHI MINH CITY, APRIL 2024

Contents

1 Member List & Workload

No.	Full Name	Student ID	Work	Effort
1	Trinh Anh Minh	2252493	Exploratory Data Analysis: Loading and Cleaning Data; Latex Editor	20%
2	Ho Khanh Nam	2252500	Analysis of Variance; Latex Editor	20%
3	Nguyen Chau Hoang Long	2252444	Exploratory Data Analysis: Data Visualization; Latex Editor	20%
4	Cao Ngoc Lam	2252419	Multivariate Linear Regression; Residual Plots, Latex Editor	20%
5	Doan Viet Tien Dat	2252141	Methodology Researcher; Latex Editor	20%

2 Abstract

Thanks to advancements in technology and storage capabilities, the presence of data has become widespread, offering significant potential to improve various aspects of our daily lives. In the business world, data serves as a valuable resource, providing essential insights that guide decision-making processes. Similarly, in research endeavors, data plays a crucial role, serving as a fundamental tool for exploring the relationships between different variables and rigorously testing hypotheses.

This report serves as an in-depth exploration of the processes involved in handling raw data, encompassing a range of tasks from meticulous data cleaning to the application of descriptive and inferential statistical techniques. Through methodologies such as Exploratory Data Analysis, Multivariate Linear Regression, and Analysis of Variance, we aim to extract meaningful insights from the dataset, uncovering underlying patterns and correlations.

Moreover, this study ventures into the realm of predictive modeling by including code to develop categorization and prediction models based on comprehensive computer performance data. These models not only offer insights into the dynamics of computer performance but also facilitate a comprehensive examination of the various factors that influence it. By providing researchers with a detailed understanding of historical trends in computer hardware development and insights into future prospects, these models serve as valuable tools for driving innovation and advancement in the field.

3 Introduction

In today's digital age, computers rely on essential hardware components like Graphics Processing Units (GPUs) and Central Processing Units (CPUs) to power a wide range of tasks, from gaming to productivity. This report delves into a dataset containing detailed information about these vital components, split into two CSV files for GPUs and for CPUs. These files cover various attributes such as clock speeds, temperatures, display resolutions, power consumption, release dates, and prices, providing a comprehensive overview of each component's capabilities and characteristics. Understanding these details is crucial for consumers making purchasing decisions, manufacturers developing new products, and analysts tracking industry trends.

By applying statistical techniques and probability theory to this dataset, this report aims to uncover insights and trends within the computer hardware industry. Through analysis, we seek to identify patterns in the data, understand the factors influencing pricing decisions, and examine how technology evolves over time. By shedding light on these dynamics, we hope to provide valuable insights that can inform decision-making processes and contribute to a deeper understanding of the forces driving innovation and market behavior in the realm of computer hardware.

Ultimately, this exploration of the dataset aims to offer practical insights for a wide range of stakeholders, including consumers, manufacturers, and industry analysts. By unraveling the complexities of GPU and CPU specifications, release dates, and pricing trends, this report seeks to empower stakeholders to make informed decisions in a rapidly evolving technological landscape. Through a combination of statistical analysis and probabilistic reasoning, we aim to provide a clearer understanding of the dynamics shaping the computer hardware industry, helping stakeholders navigate and thrive in this dynamic environment.

4 Theory and Algorithm

4.1 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) serves as an essential approach in understanding datasets, focusing on uncovering key characteristics beyond formal statistical modeling. It emphasizes the utilization of statistical graphics and visualization methods to glean insights from data. By adopting EDA, analysts can delve into the data's nuances, revealing patterns and trends that may not be apparent through traditional hypothesis testing.

Through EDA, analysts embark on a visual exploration journey, utilizing techniques such as histograms, box plots, pair plots, and correlation matrices to understand variables' behaviors and relationships. This intuitive approach allows for a comprehensive understanding of the dataset's underlying structure, facilitating informed decision-making and hypothesis generation.

Visualizing raw data through various graphical methods aids in understanding the behavior and distribution of variables:

- Histograms illustrate the distribution of numerical variables.
- Box plots depict the distribution of numerical data values, especially useful for comparing across multiple groups.
- Pair plots reveal the relationships between variables or highlight distinct clusters.
- Correlation Matrix: This table displays correlation coefficients between variables, showing the association between two variables in each cell.
- Scatter plots: These graphs show the relationship between two continuous variables, allowing for visual examination of patterns or trends.
- Heatmaps: Heatmaps represent data values in a matrix with colors, making it easy to identify patterns, trends, or correlations across variables.
- Violin plots: These combine elements of box plots and kernel density plots to display the distribution of data across different levels of a categorical variable.
- QQ plots: Quantile-Quantile plots help assess whether a dataset follows a certain theoretical distribution by comparing its quantiles to those of the theoretical distribution.
- Time series plots: Particularly useful for analyzing data over time, these graphs display data points in chronological order, aiding in identifying trends, seasonality, and anomalies.

The report will primarily employ histograms, box plots, pair plots, and correlation matrices as visualization tools for the data analysis.

4.2 Factorial Analysis of Variance (ANOVA)

Factorial Analysis of Variance (ANOVA) is a statistical technique used to analyze the effect of two or more categorical independent variables (factors) on a continuous dependent variable. It extends the principles of one-way ANOVA to situations where there are multiple independent variables, allowing researchers to examine main effects, interaction effects, and overall differences among groups.

In a factorial ANOVA, each independent variable can have two or more levels, resulting in various combinations of conditions or treatments. The primary objectives of factorial ANOVA include:

1. Assessing main effects: Main effects refer to the individual effects of each independent variable on the dependent variable, averaging across the levels of other independent variables. For example, in a study examining the effects of both gender and age group on a test score, the main effect of gender would indicate whether there is a significant difference in scores between males and females, while the main effect of age group would indicate differences across different age categories.
2. Examining interaction effects: Interaction effects occur when the effect of one independent variable on the dependent variable varies depending on the level of another independent variable. In the aforementioned example, an interaction effect between gender and age group would suggest that the impact of gender on test scores differs depending on the age group, or vice versa.

3. Testing overall differences: Factorial ANOVA also allows researchers to test whether there are overall differences in the dependent variable among the various combinations of levels of the independent variables. This involves evaluating whether the interaction effects, main effects, or both contribute significantly to the variability in the dependent variable.

The hypotheses tested in factorial ANOVA include the null hypothesis of no significant effects (main or interaction) of the independent variables on the dependent variable, versus the alternative hypothesis of at least one significant effect. The significance of these effects is typically assessed using F-tests, similar to those used in one-way ANOVA.

4.3 Comparing Two Models with Analysis of Variance (ANOVA)

When conducting regression analysis or predictive modeling, researchers often face the task of comparing different models to assess their effectiveness in explaining variation in the dependent variable. ANOVA serves as a valuable tool for this purpose, enabling a formal evaluation of whether one model significantly outperforms another.

Consider two models, Model A and Model B, both aimed at explaining the variability in a dependent variable based on a set of independent variables. To compare these models using ANOVA, the following steps are followed:

1. Fit both models: Initially, both Model A and Model B are fitted to the dataset. These models may take the form of linear regression models, logistic regression models, or other appropriate models tailored to the data and research objectives.

2. Calculate the Residual Sum of Squares (RSS) for each model: RSS quantifies the unexplained variability in the dependent variable after considering the independent variables in the model. It is computed as the sum of the squared differences between the observed values and the predicted values from the model. Mathematically, RSS for Model A and Model B can be expressed as:

$$\text{RSS}_A = \sum_{i=1}^{n} (y_i - \hat{y}_i^A)^2$$

$$\text{RSS}_B = \sum_{i=1}^{n} (y_i - \hat{y}_i^B)^2$$

   where y_i is the observed value, $\hat{y}_i^A$ and $\hat{y}_i^B$ are the predicted values from Model A and Model B respectively, and n is the number of observations.

3. Conduct ANOVA: Subsequently, an ANOVA test is conducted to compare the two models. In this context, Model A serves as the null model (containing only the intercept), while Model B acts as the alternative model (incorporating additional predictors beyond those in Model A).

4. Compute the F-statistic: The F-statistic is derived as the ratio of the difference in RSS between the two models to the RSS of Model B, divided by the degrees of freedom associated with the difference in RSS. Mathematically, the F-statistic can be expressed as:

$$F = \frac{(\text{RSS}_A - \text{RSS}_B)/(p_B - p_A)}{\text{RSS}_B/(n - p_B)}$$

   where p_A and p_B are the number of parameters in Model A and Model B respectively, and n is the number of observations.

5. Evaluate significance: Finally, the significance of the F-statistic is assessed using an F-distribution with appropriate degrees of freedom. If the F-statistic surpasses a critical value (determined based on the chosen significance level and degrees of freedom), the null hypothesis is rejected, indicating a significant difference between the two models.

By comparing the models in this systematic manner, researchers can ascertain whether the inclusion of additional predictors or increased model complexity in Model B leads to a substantial improvement in explanatory power relative to Model A.

4.4 Multivariate Linear Regression (MLR)

In practical scenarios, regression analysis often requires considering multiple predictor variables to capture the nuanced relationships within the data. When a regression model incorporates several regressor variables, it's termed as a multiple regression model. This approach enables analysts to account for the combined influence of various factors on the response variable, offering a more comprehensive understanding of the data's relationships. Multiple regression analysis facilitates the exploration of complex dynamics by capturing the joint effects of several predictors simultaneously. This methodology provides nuanced insights that aid in informed decision-making and predictive modeling.

Multiple regression analysis goes beyond traditional single-variable analysis, allowing researchers to delve deeper into the data's intricacies. By considering multiple predictors simultaneously, analysts can unravel complex patterns and relationships, enhancing the depth of understanding. This approach empowers analysts to uncover subtle trends and interactions among variables, contributing to a more holistic understanding of the data. Such insights derived from multiple regression analysis are invaluable for making informed decisions and developing accurate predictive models.

In multiple linear regression (MLR), the null hypothesis and alternative hypothesis are framed in the context of assessing the significance of the relationship between the predictor variables and the response variable:

- H_0: There is no association between the variables and the response variables. $H_0 : \beta_0 = \beta_1 = ... = \beta_p = 0$
- H_1: The response variable and at least one explanatory variable are linearly related. $H_1 : \exists\beta_i \neq 0$

The general equation of MLR can be represented as follows:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \epsilon$$

In this equation, Y represents the dependent variable, β_0 is the intercept term, $\beta_1, \beta_2, \ldots, \beta_p$ are the the coefficients corresponding to the independent variables $X_1, X_2, \ldots, X_p$ respectively, and ϵ is the error term. This equation captures the linear relationship between the dependent variable and multiple independent variables in the MLR framework.

To determine the significance of variables in multiple linear regression, analysts often rely on statistical measures such as p-values and confidence intervals. The p-value associated with each coefficient (β_i) indicates the probability of observing a coefficient as extreme as the one obtained, assuming that the null hypothesis (H_0) is true. A small p-value (typically less than a predetermined significance level, such as 0.05) suggests that the corresponding predictor variable is statistically significant. Similarly, confidence intervals provide a range of plausible values for the coefficient, with narrower intervals indicating greater precision. By examining these statistical measures, analysts can assess the significance of each predictor variable and identify those that have a substantial impact on the response variable.

Confidence intervals provide a range of plausible values for each coefficient, reflecting estimation uncertainty. A narrower interval indicates greater precision, suggesting a more reliable estimate. Analysts use both p-values and confidence intervals to gauge the significance of predictor variables. This aids in identifying influential variables, guiding decision-making and model refinement.

5 Assignment's activities

5.1 Data Importation

5.1.1 Library

Our team load all the libraries the that we used in the assignment:

```
library('dplyr')
library('magrittr')
library('GGally')
library('corrplot')
library('MASS')
library('car')
```

5.1.2 Loading Data

In this part of the project, we will analyze a dataset from the file named `All_GPUs.csv` and import it into the dataframe `GPU_data`. We will also summary all the variables in the dataset.

```
GPU_data <- read.csv("All_GPUs.csv")
summary(GPU_data)
```

In the `All_GPUs.csv`, we will focus on these below variables and the reasons for choosing these variables will be discussed in **section 5.3**.

1. Manufacturer
2. Process
3. Max_Power
4. Memory_Speed
5. Memory_Bus
6. Memory
7. Core_Speed
8. Memory_Bandwidth

Our team will extract these variables from the `GPU_data` and store them in the new dataframe `new_data`.

```
new_data <- GPU_data %>% dplyr::select("Manufacturer","Process","Max_Power","Memory_Speed
    ","Memory_Bus","Memory","Core_Speed","Memory_Bandwidth")

str(new_data)
```

The ouput will be the data of 8 variables:

```
'data.frame': 3406 obs. of  8 variables:
 $ Manufacturer    : chr  "Nvidia" "AMD" "AMD" "AMD" ...
 $ Process         : chr  "55nm" "80nm" "80nm" "65nm" ...
 $ Max_Power       : chr  "141 Watts" "215 Watts" "200 Watts" "" ...
 $ Memory_Speed    : chr  "1000 MHz" "828 MHz" "800 MHz" "1150 MHz" ...
 $ Memory_Bus      : chr  "256 Bit " "512 Bit " "256 Bit " "128 Bit " ...
 $ Memory          : chr  "1024 MB " "512 MB " "512 MB " "256 MB " ...
 $ Core_Speed      : chr  "738 MHz" "\n- " "\n- " "\n- " ...
 $ Memory_Bandwidth: chr  "64GB/sec" "106GB/sec" "51.2GB/sec" "36.8GB/sec" ...
```

5.1.3 Cleaning Data

In this section, there are a lot of data in a variables that are missing the value and some have the units so we need to clean them to make it more easy to analysis. We need the variables just have the value is number so we will need to eliminate all the units in the data.

1. Eliminate the unit of `Process`

```
new_data$Process <- as.numeric(sub("nm", "", new_data$Process))
```

2. Eliminate the unit of `Max_Power`

```
new_data$Max_Power <- as.numeric(sub("Watts", "" , new_data$Max_Power))
```

3. Eliminate the unit of `Memory_Speed`

```
new_data$Memory_Speed <- as.numeric(sub("MHz", "", new_data$Memory_Speed))
```

4. Eliminate the unit of `Memory_Bus`

```
new_data$Memory_Bus <- as.numeric(sub("Bit", "", new_data$Memory_Bus))
```

5. Eliminate the unit of `Memory`

```
new_data$Memory <- as.numeric(sub("MB", "", new_data$Memory))
```

6. Eliminate the unit of `Core_Speed`

```
new_data$Core_Speed <- as.numeric(sub("MHz", "", new_data$Core_Speed))
```

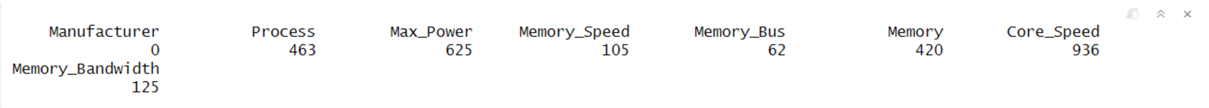
7. Eliminate the unit of `Memory_Bandwidth`

```
new_data$Memory_Bandwidth <- as.numeric(sub("GB/sec", "", new_data$Memory_Bandwidth)
    )
```

After that, we will check the NA and negatives for the value in `new_data`

```
colSums(is.na(new_data))
```

In here, the number of data that have NA values:

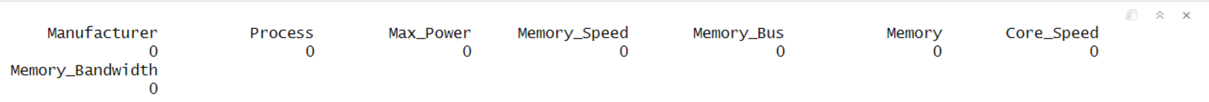


```
     Manufacturer          Process        Max_Power     Memory_Speed       Memory_Bus           Memory       Core_Speed 
                0              463              625              105               62              420              936 
 Memory_Bandwidth 
              125 
```

Then our team will start eliminate the data has NA value:

```
new_data <- na.omit(new_data)
```

After we eliminate all the NA, we will get:



```
     Manufacturer          Process        Max_Power     Memory_Speed       Memory_Bus           Memory       Core_Speed 
                0                0                0                0                0                0                0 
 Memory_Bandwidth 
                0 
```

5.2 Data Visualization

In this report, visualization allows us to explore data sets visually through histogram and boxplot, which are helpful in understanding the underlying structure of the data and formulating hypotheses for further investigation. Also, we show heatmap to represent the relationship between numeric variables pair by pair in the dataset in order to identify patterns or correlations between variables

1. We use summary() to sum up basic information about a portion of numeric variables in the dataset

```
summary(new_data[,c(2:8)])
```

2. Gain the overview of the relation between each pair of variables

```
ggpairs(new_data[,c(2:8)])
```

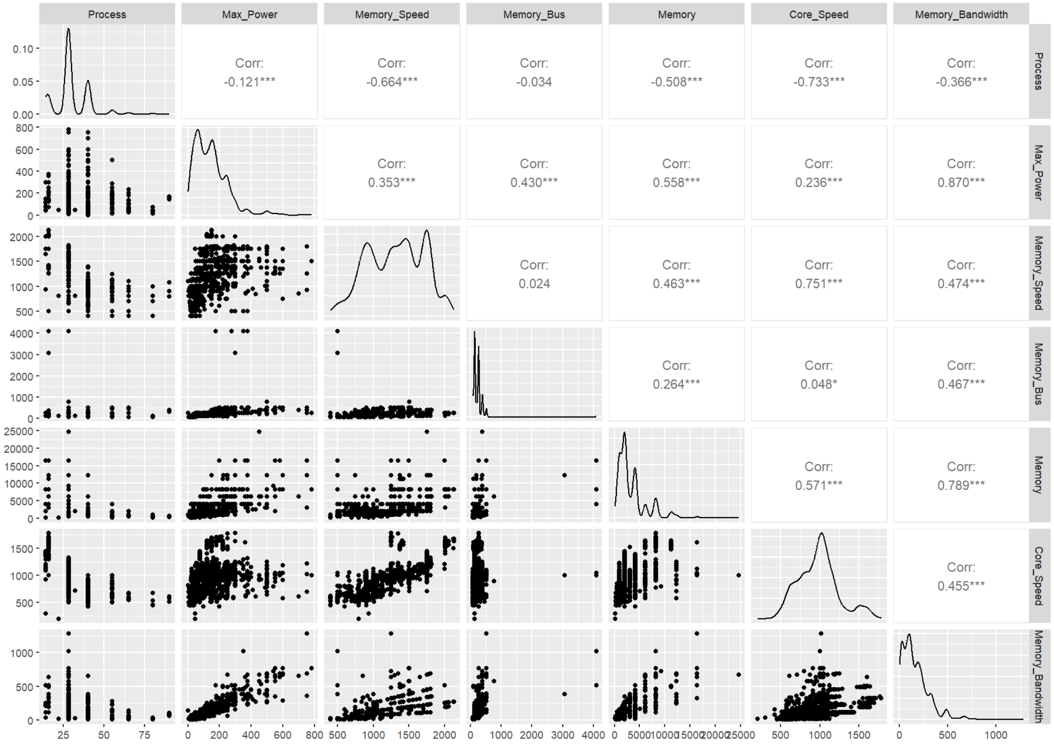


Figure 1: Pair plots of each pair of variables

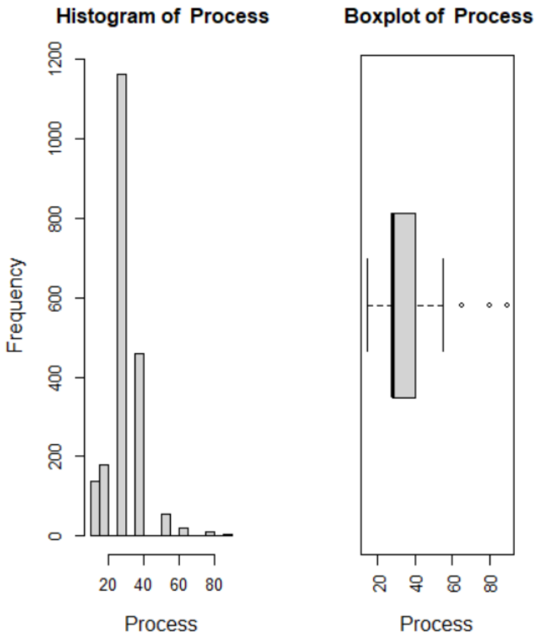
3. We define two functions in which setting up the Histogram and Boxplot to take a clearer look at the distribution of all numeric independent variables.

```
dtplot <- function (new_data , col , na.rm = TRUE) {
hist ( new_data [[ col ]] ,
main = paste ( " Histogram of " , col ) ,
xlab = col,cex.main=1.2,
cex.lab=1.2)
boxplot ( new_data [[ col ]] ,
main = paste ( " Boxplot of " , col ),
horizontal = T ,
xlab = col ,
ylab = "" ,
las = 3,
cex.main=1.2,
cex.lab=1.2)
}
```

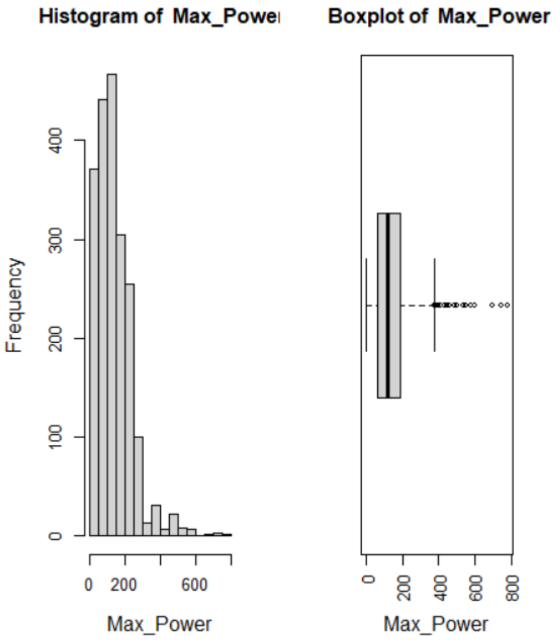
4. We plot each numeric variable in the dataset

```
plot_all <- function(new_data, na.rm = TRUE) {
    par(mfrow = c(1, 4))
    col_names <- names(new_data[,c(2:8)])
    for (col in col_names) {
        dtplot(new_data,col,na.rm)
    }
}
plot_all(new_data)
```

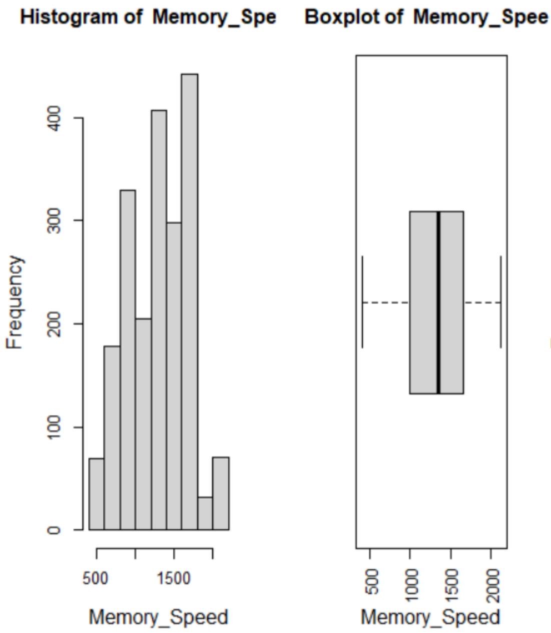
We get the univariate visualizations of each variable:



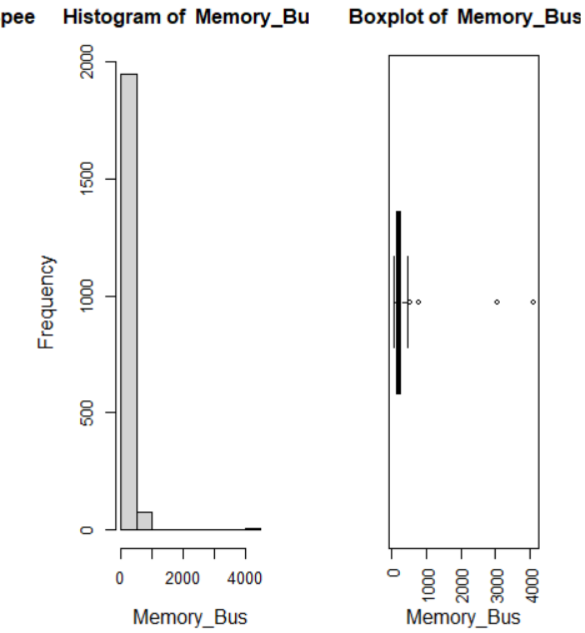
(*a*) **Process**



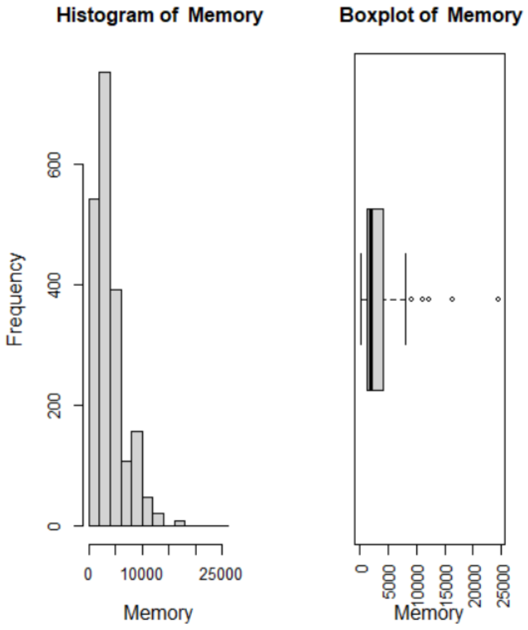
(*b*) **Max_Power**



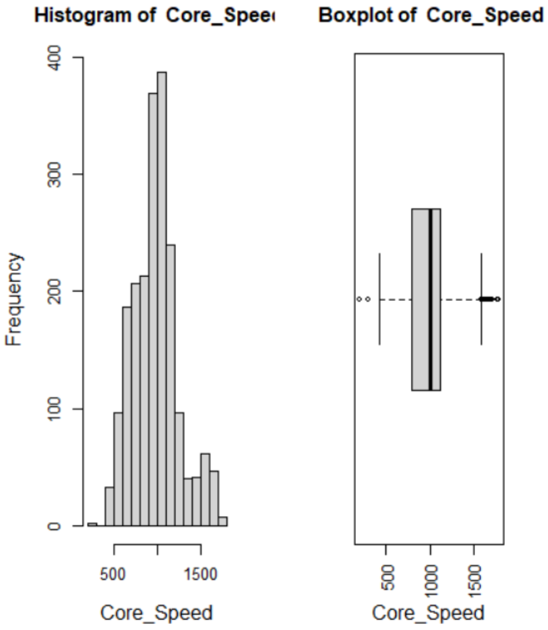
(c) **Memory_Speed**



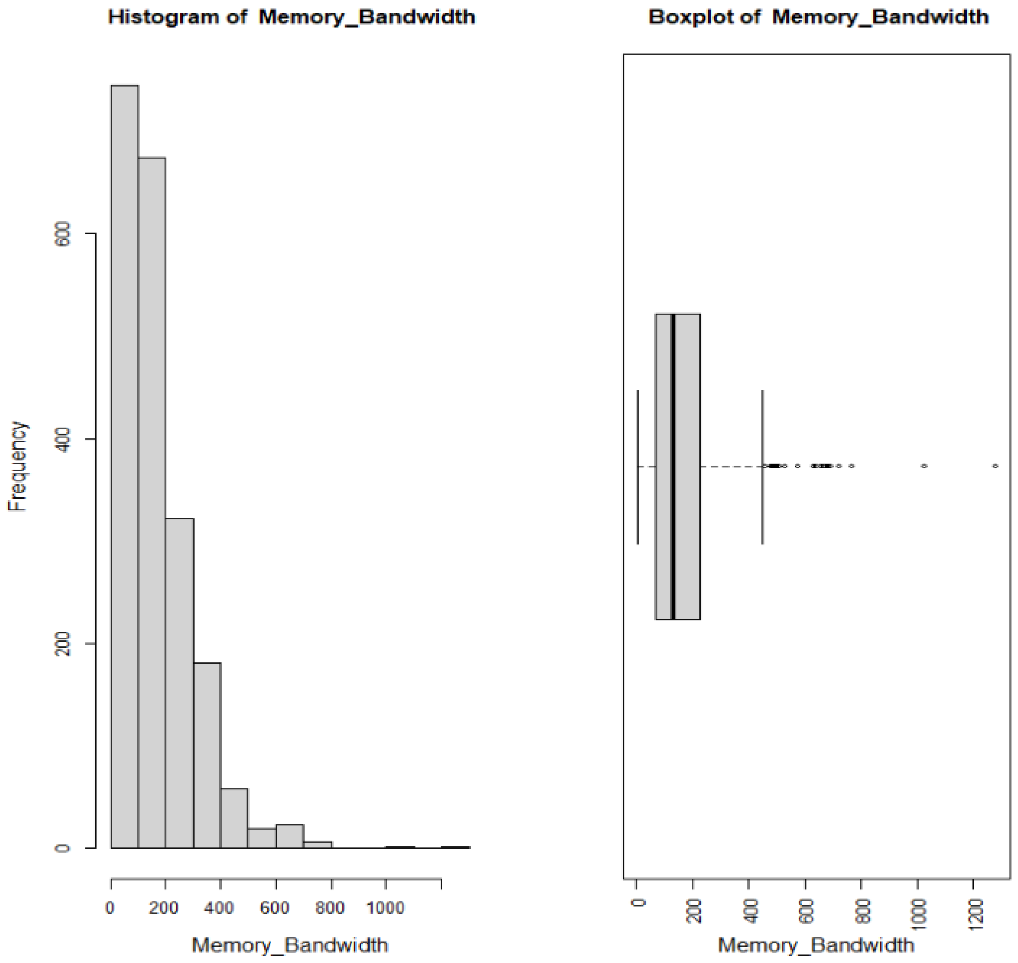
(d) **Memory_Bus**



(e) **Memory**



(f) **Core_Speed**



(g) **Memory_Bandwidth**

Figure 2: Histogram and Boxplot of variables

5. We are also interested in the correlations between all aforementioned variables to the rating of the corresponding machine. It is also shown in the ggpairs(). However, we can use heatmap to show up the variables correlations pair by pair using corrplot()

```
correlation_matrix <- cor(new_data[,c(2:8)])
corrplot(correlation_matrix,method = "square",addCoef.col ="yellow",tl.col="black",
    number.cex=1.2,tl.cex=1.1)
```

Most variables seem to have high correlations with Memory_Bandwidth. From here on we will be exploring methods of analyzing this relationship between the variables and Memory_Bandwidth.

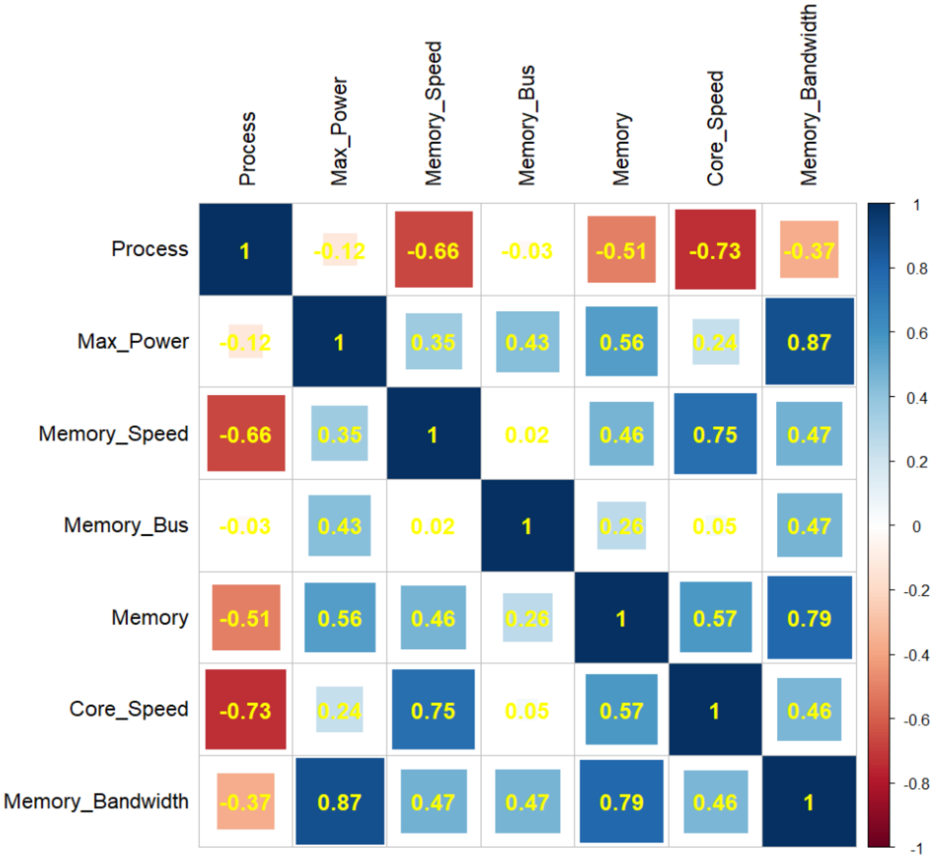


Figure 3: Correlation plot

5.3 Reason for choosing factors into MLR model

Upon delving deeper into the dataset, our group uncovered that several factors are unsuitable for inclusion in the MLR model, rendering them unusable for further analysis. We will elaborate on this issue in **Appendix** to substantiate our claim. However, the factors that remain, as outlined in section 5.1.2, are conducive to constructing a model that is well-suited for the subsequent analysis in the report.

5.4 Multivariate Linear Regression (MLR) & Analysis Of Variance (ANOVA) for 2 models

In this section, we will build a multiple linear regression model with `Memory_Bandwidth` as the dependent variable, and `Process`, `Max_Power`, `Memory_Speed`, `Memory_Bus`, `Memory`, `Pixel_Rate`, `Texture_Rate` as the independent variables. Our objective is to develop a predictive model that can accurately estimates `Memory_Bandwidth` based on other factors. This analysis will give more valuable insights for hardware designers and engineers looking to optimize memory subsystems for better performance.

5.4.1 Model fitting

First, we fit the model using `lm()` function in R. This function helps us to predict a dependant variable based on independant ones. After fitting the model, we use `summary()` function to give descriptions about the model.

```
#Fitting the model
model1 <- lm(formula = Memory_Bandwidth ~ Core_Speed + Memory + Memory_Bus + Memory_Speed
     + Max_Power + Process, data = new_data)
summary(model1)
```

The output is described in below:

```
lm(formula = Memory_Bandwidth ~ Core_Speed + Memory + Memory_Bus +
    Memory_Speed + Max_Power + Process, data = new_data)

Residuals:
    Min      1Q  Median      3Q     Max
-388.24  -22.38    3.11   20.78  380.17

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)  -2.900e+01  9.203e+00  -3.151 0.001650 **
Core_Speed    2.180e-02  6.571e-03   3.318 0.000924 ***
Memory        1.812e-02  5.071e-04  35.729  < 2e-16 ***
Memory_Bus    6.157e-02  4.250e-03  14.488  < 2e-16 ***
Memory_Speed  7.559e-03  4.074e-03   1.855 0.063697 .
Max_Power     7.887e-01  1.276e-02  61.821  < 2e-16 ***
Process      -8.114e-01  1.425e-01  -5.695 1.41e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 41.82 on 2023 degrees of freedom
Multiple R-squared:  0.9076,  Adjusted R-squared:  0.9074
F-statistic:  3313 on 6 and 2023 DF,  p-value: < 2.2e-16
```

Overall, the model seems to have a high explanatory power, indicated by the R-square value of 0.9076 and statistically significant coefficients. However, `Memory_Speed` has a p-value of 0.063697, which is slightly high compared to the conventional significance level 0.05, which indicates that the relationship between `Memory_Bandwidth` and `Memory_Speed` may not be statistically significant compared to other independent variables, which have much smaller p-value. As a result, the coefficient for `Memory_Speed` may not have a meaningful statistical interpretation and excluding it likely to improve the power of the model.

In order to investigate this hypothesis, we create a new model based on `model1`:

```
#Fitting the model
model2 <- lm(formula = Memory_Bandwidth ~ Core_Speed + Memory + Memory_Bus + Max_Power +
    Process, data = new_data)
summary(model2)
```

We got the result:

```
lm(formula = Memory_Bandwidth ~ Core_Speed + Memory + Memory_Bus +
    Max_Power + Process, data = new_data)

Residuals:
    Min      1Q  Median      3Q     Max
-391.55  -21.92    2.40   21.01  376.33

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.299e+01  8.620e+00  -2.668   0.0077 **
Core_Speed   2.795e-02  5.675e-03   4.926 9.08e-07 ***
Memory       1.795e-02  4.997e-04  35.932  < 2e-16 ***
Memory_Bus   6.026e-02  4.194e-03  14.370  < 2e-16 ***
Max_Power    7.978e-01  1.180e-02  67.631  < 2e-16 ***
Process     -8.983e-01  1.346e-01  -6.672 3.24e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 41.85 on 2024 degrees of freedom
Multiple R-squared:  0.9075,  Adjusted R-squared:  0.9073
F-statistic:  3970 on 5 and 2024 DF,  p-value: < 2.2e-16
```

In the second model, all coefficients are statistically significant at a high level, as denoted by the extremely low p-value (less than 0.05). This suggests that all independent variables have a significant impact on the dependent one. The R-Square value of 0.9075 also indicates that the model explains a large portion of the variance in the dependent variable, and the residuals are reasonably distributed.

5.4.2 Multicollinearity check

A variance inflation factor (VIF) is a measure of the amount of multicollinearity in regression analysis. Multicollinearity exists when there is a correlation between multiple independent variables in a multiple regression model. This can adversely affect the regression results. Therefore, we conducted a VIF analysis in `model2`:

```
# Multicollinearity check
print(vif(model2))
```

We got the output:

```
Core_Speed     Memory Memory_Bus  Max_Power    Process
  2.460403   2.201727   1.238946   1.732312   2.304289
```

Based on the analysis, it can be observed that all VIF values are relatively low than 5, indicating that there isn't severe multicollinearity among the predictor variables.

5.4.3 Factorial ANOVA

We use factorial ANOVA (an extension of one-way ANOVA) to assess the effect of the independent variables on the memory bandwidth by the MLR models that are mentioned in the section above. This statistical method is similar to one-way ANOVA (using only one factor) but it takes multiple factors' effect into account.

- H_0: For each independent variable, there is no difference among group means
- H_1: For each independent variable, there is at least 1 group mean different from the others.

Due to the difference in number of observations in each factor (unbalanced), we will utilize the type II test for ANOVA, available in `Anova()` function in the 'car' package

First, we consider the factorial ANOVA test for model1:

```
Anova(model1, type=2)
```

Then we get the result for model 1:

```
Anova Table (Type II tests)

Response: Memory_Bandwidth
              Sum Sq   Df   F value    Pr(>F)
Core_Speed     19252    1   11.0059 0.0009243 ***
Memory       2233098    1 1276.5828 < 2.2e-16 ***
Memory_Bus    367163    1  209.8942 < 2.2e-16 ***
Memory_Speed    6021    1    3.4422 0.0636969 .
Max_Power    6685480    1 3821.8519 < 2.2e-16 ***
Process        56739    1   32.4354 1.412e-08 ***
Residuals    3538789 2023
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p_value of `Core_Speed`, `Memory`, `Memory_bus`, `Max_Power` and `Process` are smaller than significance level 0.05. Therefore we can reject H_0 and conclude that they are significant. Since the `Memory_speed` has p_value greater the significance level, its significance is small and thus, it can be removed in model 2.

Next, we test the factorial ANOVA for model 2, which removed the `Memory_speed` factor:

```
Anova(model2, type=2)
```

Then we get the result of model 2:

```
Anova Table (Type II tests)

Response: Memory_Bandwidth
            Sum Sq   Df  F value    Pr(>F)
Core_Speed   42495    1   24.264 9.083e-07 ***
Memory     2261223    1 1291.103 < 2.2e-16 ***
Memory_Bus  361661    1  206.499 < 2.2e-16 ***
Max_Power  8010847    1 4573.998 < 2.2e-16 ***
Process      77967    1   44.517 3.244e-11 ***
Residuals  3544810 2024
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

After testing the factorial ANOVA for 2 models, we can conclude that all the variables are now significant and we cannot remove any further factors.

5.4.4 Using ANOVA to compare 2 models

In order to compare the model 1 and model 2, providing that they are from the same dataset, we can use the `anova()` function. By passing the model objects as arguments, the function will return an ANOVA testing, which determines whether the more complex model is significantly better than the simpler model in terms of capturing the data. If the resulting p_value less than the significant level (typically the significant level equals to 0.05), we conclude that the more complex model is significantly better than the simplier model. Conversely, if the p_value is greater than the significant level, we should favor the simpler one.

- H_0: There is not much difference between the simple and complex model, the simple model is better
- H_1: The complex model is better than the simple one

We will use `anova()` function to compare the mentioned MLR models: model 1 with Memory_speed and model 2 without Memory_speed:

```
anova(model1, model2)
```

Then, we get the result of the comparison test:

```
Analysis of Variance Table

Model 1: Memory_Bandwidth ~ Core_Speed + Memory + Memory_Bus + Memory_Speed +
    Max_Power + Process
Model 2: Memory_Bandwidth ~ Core_Speed + Memory + Memory_Bus + Max_Power +
    Process
  Res.Df     RSS Df Sum of Sq      F Pr(>F)
1   2023 3538789
2   2024 3544810 -1   -6021.3 3.4422 0.0637 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Specifically, we can elaborate on the test result:

- The test calculates based on the reduction of residual sum of square (SSE).
- The degrees of freedom (df) of row 2 is -1, which means that the model 2 is less than 1 parameter (`Memory_speed`) compared to the model 1 (the model 2 is simpler than the model 1).
- The sum of square is the difference between the SSE of 2 models.
- The p-value is greater than the significance level (0.05), which implies that we cannot reject H_0. Therefore, the factor `Memory_speed` has no statistical significance and we should favor the model 2 (the simpler model).

5.4.5 Final model

After VIF check and ANOVA, our final multiple linear regression model is Model 2. The regression coefficients for each independent variable in this model are as follows:

```
Memory_Bandwidth = -22.99344 + 0.02795 * Core_Speed + 0.01795 * Memory + 0.06026 * Memory
    _Bus + 0.79778 * Max_Power - 0.89826 * Process
```

5.5 Residual plots

From the final model chosen in above section (model2), we draw the residual plots for model2:

```
plot(model2)
```

We got the following plots:

5.5.1 Residuals vs Fitted

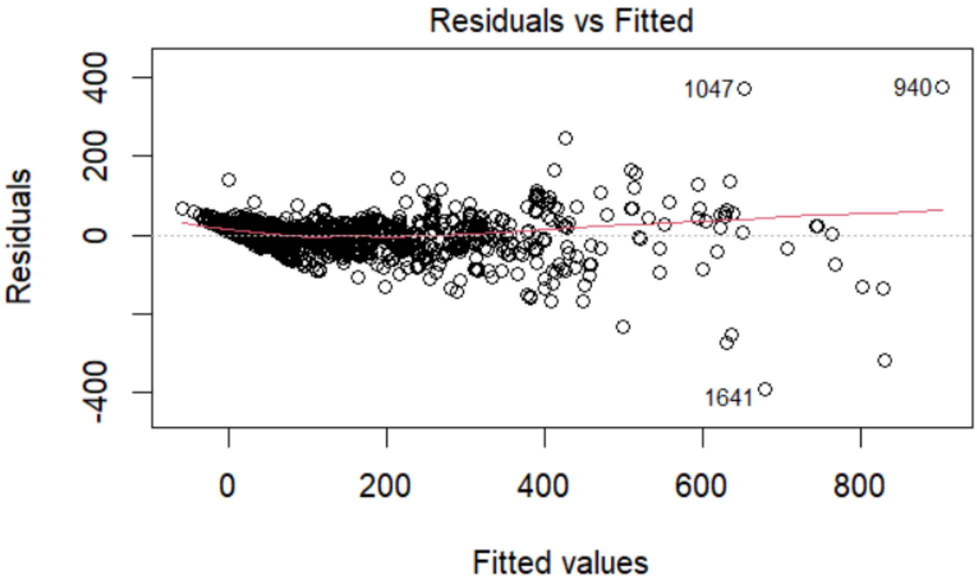


Figure 4: Residuals vs Fitted

The above scatterplot represents the residuals (the differences between the observed and predicted values) against the fitted values (the predicted values from the model). In our case, the residuals are randomly scattered around the zero line, indicating that the model adequately captures the variation in the data.

5.5.2 Normal Q-Q

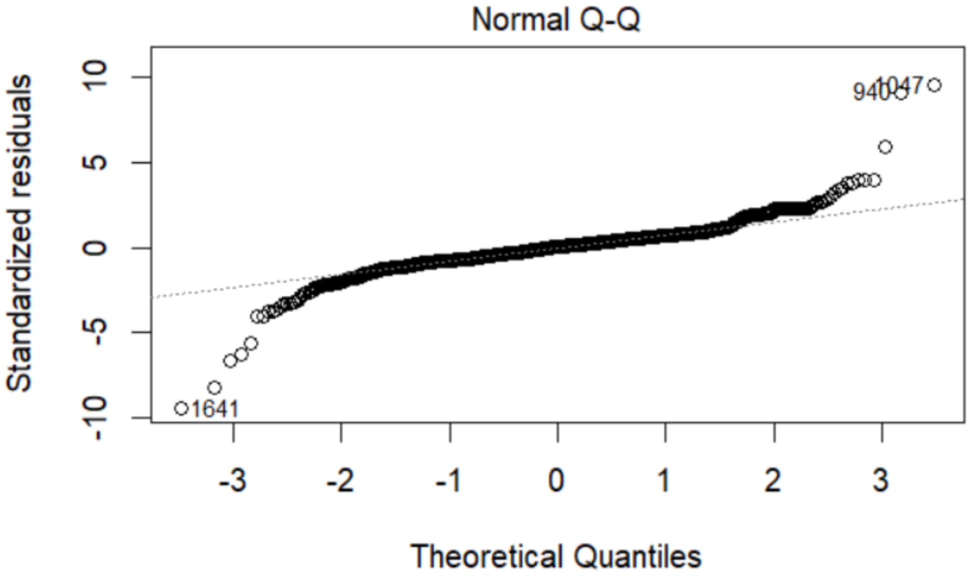


Figure 5: Normal Q-Q

Q-Q plot, or quantile-quantile plot, is a graphical representation created by plotting two sets of quantiles against one another. If both sets of quantiles came from the same distribution, we should see the points forming a line that's roughly straight. In our case, there would be two quantiles:

1. **Observed Quantiles**: The quantiles of the residuals from the model
2. **Expected Quantiles**: The quantiles that would be expected if the residuals followed a normal distribution

From the plotting, although there is some deviations in the head and tail, over by large of the dataset it fitted the straight line well, indicating that the residuals of our model is normally distributed.

5.5.3 Scale Location

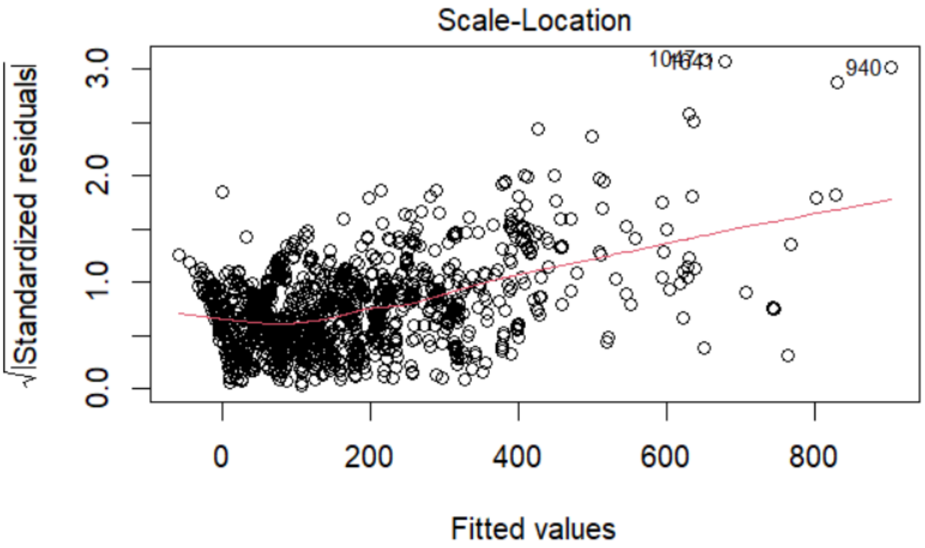


Figure 6: Scale-Location

The Scale-Location plot, or the variance plot, is a type of diagnostic plot used in regression analysis to verify the assumption of homoscedasticity (the constant variance) and linearity.

The Scale-Location plot contains the square root of the absolute standardized residuals in the y-axis and the fitted values, or predictors, in the x-axis. Moreover, the scale-location indicates the assumption of homoscedasticity through the spread of the points in the scatter plot. If the assumptions of homoscedasticity and linearity hold true, then the points on the plot should be distributed around a horizontal line with equal spread along the y-axis (the square root of standardize residuals) for all levels of the x-axis (the fitted values).

In the result of Scale-Location plot in figure 6, we can see that the red line is not really a horizontal line but rather a upward line, which indicates that the model violates the assumption of homoscedasticity.

5.5.4 Residuals vs Leverage

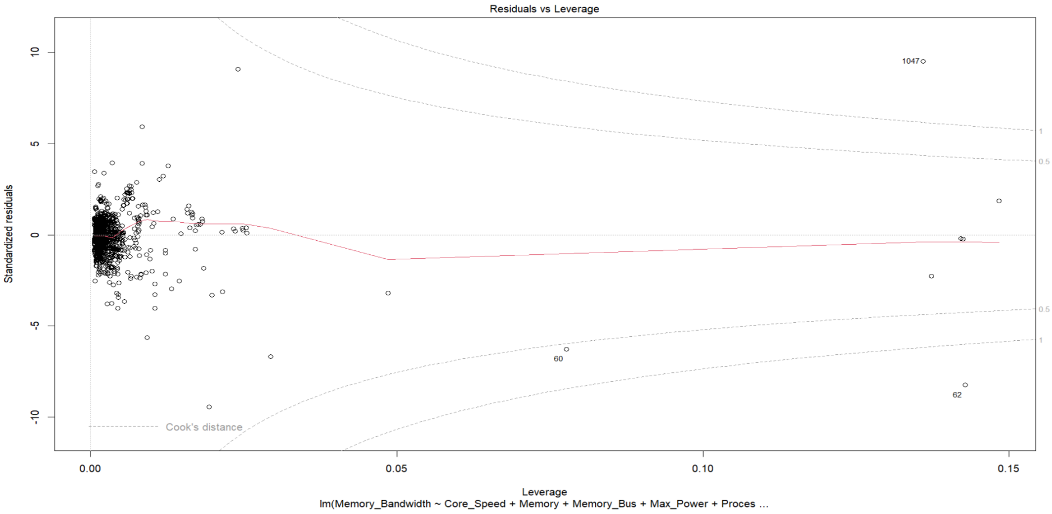


Figure 7: Residuals vs Leverage

The Residual vs Leverage plot is a diagnostic tool used in linear regression to identify influential observations in a regression model. As can be observed, there are a few points with high leverage, but there are no points with a Cook's distance greater than one, which indicated that these outliers are not likely to be having a large influence on the regression line. Therefore, the regression model fits the data well.

6 Conclusion

In this report, we conducted a comprehensive analysis of GPU specifications to understand their impact on performance, with a focus on memory bandwidth. Beginning with data importation and cleaning, we ensured the dataset's integrity and consistency. Visualizations such as histograms and correlation plots provided insights into the relationships between variables, highlighting significant correlations between memory bandwidth and attributes like memory size and core speed.

Moving forward, we built a multiple linear regression (MLR) model to predict memory bandwidth based on various GPU specifications. Through model fitting and ANOVA tests, we identified core speed, memory size, memory bus, max power, and process technology as significant predictors of memory bandwidth. Our analysis further validated the model's performance through checks for multicollinearity and residual analysis, demonstrating its robustness and suitability for predicting GPU performance accurately.

Overall, our analysis contributes valuable insights for hardware designers and engineers seeking to optimize GPU performance. By leveraging statistical techniques and probability theory, we have deepened our understanding of the factors driving GPU performance, laying the groundwork for future advancements in computer hardware design and optimization.

7 References

References

[1] Douglas C. Montgomery. *Applied Statistics and Probability for Engineers*

[2] ILLISEK, 2017. *Computer Parts (CPUs and GPUs) How did computer specifications and performance evolve over time?*. Links: https://www.kaggle.com/datasets/iliassekkaf/computerparts?resource=downlo

[3] Kassambara. March 10, 2018. *Simple Linear Regression in R*. Links: http://www.sthda.com/english/articles/40-regression-analysis/167-simple-linearregression-in-r/

[4] Learning Statistics with R. *Chapter 16 Factorial ANOVA*. Links: https://learningstatisticswithr.com/book/anova2.html#factorialanovasimple

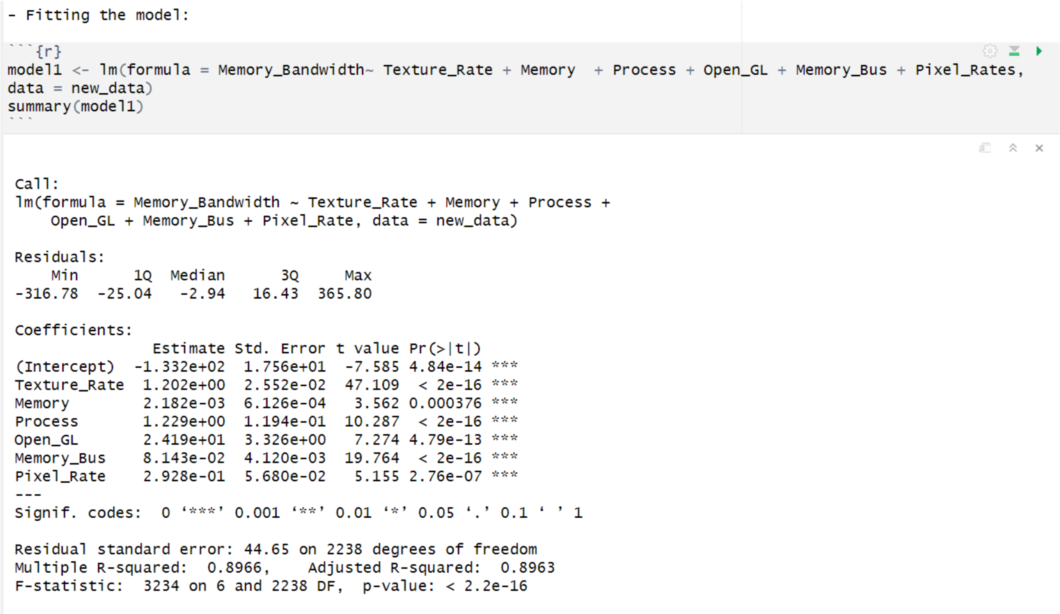
[5] YaRrr!. *The Pirate's Guide to R, 15.3 Comparing regression models with anova()*. Links https://bookdown.org/ndphillips/YaRrr/comparing-regression-models-with-anova.html

[6] Yang Lydia Yang. January 4, 2023 *Factorial ANOVA*. Links https://stats.libretexts.org/Courses/Kansas_State_University/EDCEP_917%3A_Experimental_Design_(Yang)/03%3A_Between-Subjects_Factorial_Design/3.02%3A_Factorial_ANOVA_-_Main_Effects

[7] National Library of Medicine *Introduction to Multivariate Regression Analysis* . Links https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3049417/

8 Appendix

Here are two example combination of factors make our team unconveniently to carry out the analysis for the next section of MLR model.



~~~
- Fitting the model:

```{r}
model1 <- lm(formula = Memory_Bandwidth~ Texture_Rate + Memory  + Process + Open_GL + Memory_Bus + Pixel_Rates,
data = new_data)
summary(model1)
```
~~~

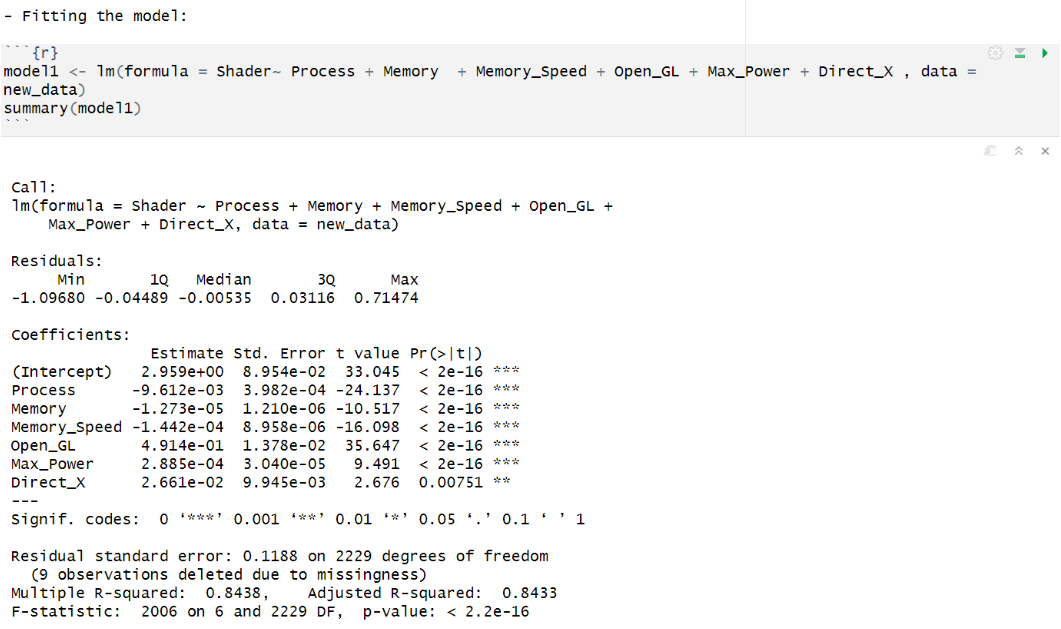
```
Call:
lm(formula = Memory_Bandwidth ~ Texture_Rate + Memory + Process + 
    Open_GL + Memory_Bus + Pixel_Rate, data = new_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-316.78  -25.04   -2.94   16.43  365.80 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -1.332e+02  1.756e+01  -7.585 4.84e-14 ***
Texture_Rate  1.202e+00  2.552e-02  47.109  < 2e-16 ***
Memory        2.182e-03  6.126e-04   3.562 0.000376 ***
Process       1.229e+00  1.194e-01  10.287  < 2e-16 ***
Open_GL       2.419e+01  3.326e+00   7.274 4.79e-13 ***
Memory_Bus    8.143e-02  4.120e-03  19.764  < 2e-16 ***
Pixel_Rate    2.928e-01  5.680e-02   5.155 2.76e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 44.65 on 2238 degrees of freedom
Multiple R-squared:  0.8966,	Adjusted R-squared:  0.8963 
F-statistic:  3234 on 6 and 2238 DF,  p-value: < 2.2e-16
```

In the first example, we use the factors `Memory_Bandwidth`, `Texture_Rate`, `Memory`, `Process`, `Open_GL`, `Memory_Bus`, `Pixel_Rates` in our MLR model, in which `Memory_BandWidth` is dependent variable. After fitting the model, it can be observed that all the p-value of all independent variables is less than 0.05, indicated that all the variables have meaningful statistical interpretation and we cannot exclude any variables to form a simpler model, which can be used for ANOVA compare two model section. Therefore, this combination of factors give our report inconvenient to analysis.



~~~
- Fitting the model:

```{r}
model1 <- lm(formula = Shader~ Process + Memory  + Memory_Speed + Open_GL + Max_Power + Direct_X , data =
new_data)
summary(model1)
```
~~~

```
Call:
lm(formula = Shader ~ Process + Memory + Memory_Speed + Open_GL + 
    Max_Power + Direct_X, data = new_data)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.09680 -0.04489 -0.00535  0.03116  0.71474 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)   2.959e+00  8.954e-02  33.045  < 2e-16 ***
Process      -9.612e-03  3.982e-04 -24.137  < 2e-16 ***
Memory       -1.273e-05  1.210e-06 -10.517  < 2e-16 ***
Memory_Speed -1.442e-04  8.958e-06 -16.098  < 2e-16 ***
Open_GL       4.914e-01  1.378e-02  35.647  < 2e-16 ***
Max_Power     2.885e-04  3.040e-05   9.491  < 2e-16 ***
Direct_X      2.661e-02  9.945e-03   2.676  0.00751 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1188 on 2229 degrees of freedom
  (9 observations deleted due to missingness)
Multiple R-squared:  0.8438,	Adjusted R-squared:  0.8433 
F-statistic:  2006 on 6 and 2229 DF,  p-value: < 2.2e-16
```

In the second example, we use the factors `Shader`, `Process`, `Memory`, `Memory_Speed`, `Open_GL`, `Max_Power`, `Direct_X` the scenario in example 1 is repeated, where all the p-value of all dependent variable is still less than 0.05, make the further analysis inconveniently.