

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



## XÁC SUẤT VÀ THỐNG KÊ (MT2013)

Assignment

## "Tổng quan về hiệu năng CPU"

GVHD: Phan Thị Hường

Nhóm sinh viên thực hiện: Nguyễn Huy Hoàng - 2211091 - L13

Nguyễn Hàm Hoàng - 2211089 - L13

Hồ Nguyễn Phi Hùng - 2211327 - L13

Nguyễn Trịnh Ngọc Huân - 2211144 - L13

THÀNH PHỐ HỒ CHÍ MINH, 2023



## Mục lục

<b>1</b>	<b>Thành viên và Khối lượng công việc</b>	<b>2</b>
<b>2</b>	<b>Tổng quan dữ liệu</b>	<b>3</b>
2.1	Ngữ cảnh dữ liệu . . . . .	3
2.2	Tổng quan các loại biến . . . . .	3
<b>3</b>	<b>Kiến thức nền</b>	<b>6</b>
3.1	Analysis of Variance - Phân tích phương sai (ANOVA) . . . . .	6
3.2	Multivariate Linear Regression - Hồi quy Tuyến tính (MLR) . . . . .	6
<b>4</b>	<b>Tiền xử lý số liệu</b>	<b>8</b>
4.1	Đọc dữ liệu . . . . .	8
4.2	Xử lý định dạng dữ liệu . . . . .	9
4.3	Xử lý dữ liệu khuyết . . . . .	10
<b>5</b>	<b>Thống kê mô tả</b>	<b>12</b>
5.1	Tìm giá trị ngoại lai . . . . .	12
5.2	Vẽ đồ thị . . . . .	13
5.3	Các giá trị thống kê mô tả . . . . .	19
<b>6</b>	<b>Thống kê suy diễn</b>	<b>20</b>
6.1	Phương pháp ANOVA . . . . .	20
6.1.1	Anova 1 yếu tố . . . . .	20
6.1.2	Anova 2 yếu tố . . . . .	21
6.2	Hồi quy đa tuyến tính . . . . .	23
6.2.1	Xây dựng mô hình hồi quy tuyến tính . . . . .	23
6.2.2	Kiểm định hệ số hồi quy: . . . . .	23
6.2.3	Kiểm tra giả định của mô hình hồi quy . . . . .	25
6.2.4	Dự báo . . . . .	29
<b>7</b>	<b>Thảo luận và mở rộng</b>	<b>30</b>
7.1	Thảo luận . . . . .	30
7.2	Mở rộng . . . . .	31
7.2.1	Mô hình hồi quy Ridge và Lasso . . . . .	31
<b>8</b>	<b>Tài liệu tham khảo</b>	<b>33</b>
<b>9</b>	<b>Nguồn dữ liệu và nguồn code</b>	<b>33</b>



## 1 Thành viên và Khối lượng công việc

Fullname	Student ID	Problems	Điểm cộng
Nguyễn Huy Hoàng	2211091	Thống kê suy diễn, Thảo luận và mở rộng, Code R	0.
Nguyễn Trịnh Ngọc Huân	2211144	Viết báo cáo, Tổng quan dữ liệu, Kiến thức nền, Thống kê mô tả.	0
Nguyễn Hàm Hoàng	2211089	Viết báo cáo, Kiến thức nền, Thống kê mô tả, Thảo luận và mở rộng.	0
Hồ Nguyễn Phi Hùng	2211327	Làm sạch , Làm rõ dữ liệu, thống kê suy diễn, Code R.	0

## 2 Tổng quan dữ liệu

### 2.1 Ngữ cảnh dữ liệu

Tập dữ liệu được cho chứa thông tin về các chi tiết kỹ thuật, ngày sản xuất và giá bán của các linh kiện máy tính bao gồm GPU và CPU. Trong bài tập lớn này, nhóm chọn tập tin *Intel\_CPUs.csv* để đánh giá *tổng quan về hiệu năng CPU*. Sau đây là một vài thông tin chung về tập dữ liệu:

- **Tiêu đề:** Computer Parts (CPUs and GPUs)
- **Thông tin tham khảo của nguồn dữ liệu:**
  - (a) Tác giả: ILISSEK
  - (b) Ngày đưa ra dữ liệu: 6 năm trước
- **Số biến:**
  - (a) *ALL\_GPUs.csv*: 34
  - (b) *Intel\_CPUs.csv*: 45

### 2.2 Tổng quan các loại biến

Trong bài tập lớn này, để thuận tiện cho việc phân tích và đánh giá, nhóm đã chọn ra 9 biến để phân tích, bao gồm:



Tên biến	Kiểu biến	Đơn vị	Mô tả
Số lượng nhân (nb_of_Cores)	$\{x \in \mathbb{N} \mid 1 \leq x \leq 72\}$ , liên tục	Không có	Thuật ngữ phần cứng mô tả số lượng CPU độc lập.
Tốc độ cơ bản của bộ xử lý (Processor_Base_Frequency)	$\{x \in \mathbb{Z} \mid 32 \leq x \leq 4300\}$ , liên tục	MHz	Mô tả tốc độ mà các transistor của bộ xử lý mở và đóng.
Công suất tiêu thụ tối đa (TDP)	$\{x \in \mathbb{Z} \mid 0.025 \leq x \leq 300\}$ , liên tục	W	Đại diện cho công suất trung bình, tính bằng watt, mà bộ xử lý tiêu thụ khi hoạt động ở Tần số Cơ bản với tất cả các lõi hoạt động dưới một công việc có độ phức tạp cao, được định nghĩa bởi Intel.
Kích thước bộ nhớ tối đa (Max_Memory_Size)	$\{x \in \mathbb{Z} \mid 1.0 \leq x \leq 4198.4\}$ , liên tục	GB	Khả năng hỗ trợ dung lượng bộ nhớ tối đa của bộ xử lý.
Max_nb_of_PCI_Express_Lanes	$\{x \in \mathbb{N} \mid 0 \leq x \leq 48\}$ , liên tục	Không có	Số lượng tối đa các làn PCI Express (PCIe) được hỗ trợ.
Băng thông bộ nhớ tối đa (Max_Memory_Bandwidth)	$\{x \in \mathbb{Z} \mid 1.6 \leq x \leq 352\}$ , liên tục	GB/s	Tốc độ tối đa mà dữ liệu có thể được đọc từ hoặc lưu vào bộ nhớ bán dẫn bởi bộ xử lý (tính bằng GB/s).
Lithography	$\{x \in \mathbb{N} \mid 14 \leq x \leq 250\}$ , liên tục	nm	Công nghệ bán dẫn được sử dụng để sản xuất một mạch tích hợp, và được tính bằng đơn vị nanômét (nm).
nb_of_Threads	$\{x \in \mathbb{N} \mid 1 \leq x \leq 56\}$ , liên tục	Không có	Một Thread, hay luồng thực thi, là một thuật ngữ phần mềm chỉ một dãy lệnh cơ bản và có thứ tự mà có thể được truyền qua hoặc xử lý bởi một lõi CPU duy nhất.
Vertical_Segment	("Desktop", "Embedded", "Sever", "Mobile"), rời rạc	Không có	loại nền tảng mà CPU chạy trên đó.

9 biến trên được chọn sau khi nhóm đã tham khảo và tìm hiểu ý nghĩa của các biến qua nhiều nguồn khác nhau.

Tên biến	Ý nghĩa và nguồn tham khảo
Số lượng nhân (nb_of_Cores)	Mỗi nhân trong CPU có thể thực hiện một luồng dữ liệu. Do đó, số lượng nhân càng nhiều, CPU càng có khả năng xử lý nhiều tác vụ cùng một lúc.
Tốc độ cơ bản của bộ xử lý (Processor_Base_Frequency)	Tốc độ cơ bản của bộ xử lý cho biết số lượng chu kỳ mà CPU có thể thực hiện trong một giây. Tốc độ càng cao, CPU càng có khả năng xử lý nhiều tác vụ trong cùng một khoảng thời gian.
Công suất tiêu thụ tối đa (TDP)	TDP cho biết lượng nhiệt tối đa mà hệ thống làm mát cần loại bỏ khi CPU hoạt động ở tốc độ cơ bản. TDP càng thấp, hiệu quả năng lượng của CPU càng cao.
Kích thước bộ nhớ tối đa (Max_Memory_Size)	Kích thước bộ nhớ tối đa cho biết lượng bộ nhớ tối đa mà CPU có thể hỗ trợ. Điều này ảnh hưởng đến khả năng xử lý dữ liệu của CPU1.
Max_nb_of_PCI_Express_Lanes	PCIe lanes cung cấp các đường truyền dữ liệu tốc độ cao cho việc giao tiếp giữa CPU (Central Processing Unit) và các thiết bị ngoại vi như card đồ họa, thiết bị lưu trữ, bộ điều hợp mạng và card âm thanh.
Băng thông bộ nhớ tối đa (Max_Memory_Bandwidth)	Băng thông bộ nhớ tối đa cho biết lượng dữ liệu tối đa mà CPU có thể truy cập từ bộ nhớ trong một giây. Băng thông càng cao, hiệu suất CPU càng tốt.
Lithography	Kích thước của các transistor này, thường được đo bằng nanomet (nm), cho biết kích thước nhỏ nhất mà một transistor có thể có trên chip. Khi kích thước transistor nhỏ hơn, chúng ta có thể đặt nhiều transistor hơn trên cùng một diện tích chip, từ đó tăng cường hiệu suất và giảm mức tiêu thụ năng lượng.
nb_of_Threads	Số lượng luồng thực thi trên một bộ xử lý (CPU) đề cập đến tổng số luồng thực thi mà CPU có thể xử lý đồng thời. Số luồng càng nhiều, CPU có khả năng xử lý đa nhiệm và các tác vụ đồng thời một cách hiệu quả hơn.
Vertical_Segment	Phân đoạn theo chiều dọc (Vertical_Segment) cho phép các kỹ sư tối ưu hóa các đoạn khác nhau của bộ xử lý để tối ưu trong các bộ phận tương ứng của chúng.

Một số nguồn tham khảo khác có thể tìm được ở [BBC Bitesize](#), [MakeUseOf](#), [Make Tech Easier](#).

## 3 Kiến thức nền

### 3.1 Analysis of Variance - Phân tích phương sai (ANOVA))

Phân tích phương sai (Analysis of Variance) hay còn gọi là kiểm định ANOVA là một kỹ thuật thống kê tham số được sử dụng để so sánh các bộ dữ liệu. Nói một cách dễ hiểu, phân tích ANOVA có chức năng đánh giá sự khác biệt tiềm năng trong một biến phụ thuộc mức quy mô bằng một biến mức danh nghĩa có từ 2 loại trở lên. Các nhà phân tích sử dụng thử nghiệm ANOVA để xác định ảnh hưởng của các biến độc lập đối với biến phụ thuộc trong nghiên cứu hồi quy. Kỹ thuật kiểm định ANOVA này được phát triển bởi Ronald Fisher năm 1918. Hai loại phân tích ANOVA:

- ANOVA một yếu tố là một loại thử nghiệm thống kê so sánh phương sai trong nhóm có nghĩa là trong một mẫu trong khi chỉ xem xét một yếu tố hoặc một biến độc lập. Phương sai một yếu tố so sánh ba hoặc nhiều hơn ba nhóm phân loại để xác định xem có sự khác biệt giữa chúng hay không. Trong mỗi nhóm nên có ba hoặc nhiều quan sát và phương tiện của các mẫu được so sánh.
- ANOVA hai yếu tố là một phần mở rộng của phân tích phương sai một yếu tố. Với One Way, bạn có một biến độc lập ảnh hưởng đến biến phụ thuộc. Còn với two-way ANOVA, sẽ có 2 biến độc lập.

### 3.2 Multivariate Linear Regression - Hồi quy Tuyến tính (MLR)

Hồi quy tuyến tính là một kỹ thuật phân tích dữ liệu dự đoán giá trị của dữ liệu không xác định bằng cách sử dụng một giá trị dữ liệu liên quan và đã biết khác. Nó mô hình toán học biến không xác định hoặc phụ thuộc và biến đã biết hoặc độc lập như một phương trình tuyến tính. Ví dụ, giả sử rằng bạn có dữ liệu về chi phí và thu nhập của bạn trong năm ngoái. Kỹ thuật hồi quy tuyến tính phân tích dữ liệu này và xác định rằng chi phí của bạn là một nửa thu nhập của bạn. Sau đó, họ tính toán một chi phí trong tương lai không rõ bằng cách giảm một nửa thu nhập được biết đến trong tương lai.

Phân tích hồi quy tuyến tính phải sửa đổi hoặc biến đổi các giá trị dữ liệu về mặt toán học để đáp ứng bốn giả định sau đây:

1. Mỗi quan hệ tuyến tính
2. Phần dư độc lập
3. Tính chuẩn
4. Phương sai không đổi

Công thức tổng quát của MLR như sau:

$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \dots + \beta_k x_{i,k} + \epsilon_i. \quad (1)$$

Trong đó:

- $y$ : biến phụ thuộc, là biến chịu tác động của biến khác.
- $x, x_1, x_2, x_n$ : biến độc lập, là biến tác động lên biến khác.

- $\beta_0$ : hằng số hồi quy, hay còn được gọi là hệ số chặn. Đây là chỉ số nói lên giá trị của  $y$  sẽ là bao nhiêu nếu tất cả  $x$  cùng bằng 0. Nói cách khác, chỉ số này cho chúng ta biết giá trị của  $y$  là bao nhiêu nếu không có các  $x$ . Khi biểu diễn trên đồ thị Oxy,  $\beta_0$  là điểm trên trục Oy mà đường hồi quy cắt qua.
- $\beta_1, \beta_2, \beta_n$ : hệ số hồi quy, hay còn được gọi là hệ số góc. Chỉ số này cho chúng ta biết về mức thay đổi của  $y$  gây ra bởi  $x$  tương ứng. Nói cách khác, chỉ số này nói lên có bao nhiêu đơn vị  $y$  sẽ thay đổi nếu  $x$  tăng hoặc giảm một đơn vị.
- $\epsilon$ : sai số. Chỉ số này càng lớn càng khiến cho khả năng dự đoán của hồi quy trở nên kém chính xác hơn hoặc sai lệch nhiều hơn so với thực tế. Sai số trong hồi quy tổng thể hay phần dư trong hồi quy mẫu đại diện cho hai giá trị, một là các biến độc lập ngoài mô hình, hai là các sai số ngẫu nhiên.



## 4 Tiền xử lý số liệu

### 4.1 Đọc dữ liệu

Đọc dữ liệu bằng read.csv và hiển thị dữ liệu đến thiết bị đầu cuối để kiểm tra xem dữ liệu có được nhập thành công hay không.

```
# Set the path of the CSV file into a variable called 'path'
path <- "PathToFile/Intel_CPUs.csv"
# Use the 'read.csv' function to read the file
data <- read.csv(file = path, header = TRUE, sep = ",")
# Print the first 6 lines of the data
head(data)
```

	Product_Collection	Vertical_Segment	Processor_Number	Status	Launch_Date	Lithography
1	7th Generation Intel® Core™ i7 Processors	Mobile	i7-7Y75	Launched	Q3'16	14 nm
2	8th Generation Intel® Core™ i5 Processors	Mobile	i5-8250U	Launched	Q3'17	14 nm
3	8th Generation Intel® Core™ i7 Processors	Mobile	i7-8550U	Launched	Q3'17	14 nm
4	Intel® Core™ X-series Processors	Desktop	i7-3820	End of Life	Q1'12	32 nm
5	7th Generation Intel® Core™ i5 Processors	Mobile	i5-7Y57	Launched	Q1'17	14 nm
6	Intel® Celeron® Processor 3000 Series	Mobile	3205U	Launched	Q1'15	14 nm
	Recommended_Customer_Price	nb_of_Cores	nb_of_Threads	Processor_Base_Frequency	Max_Turbo_Frequency	Cache
1	\$393.00	2	4	1.30 GHz	3.60 GHz	4 MB SmartCache
2	\$297.00	4	8	1.60 GHz	3.40 GHz	6 MB SmartCache
3	\$409.00	4	8	1.80 GHz	4.00 GHz	8 MB SmartCache
4	\$305.00	4	8	3.60 GHz	3.80 GHz	10 MB SmartCache
5	\$281.00	2	4	1.20 GHz	3.30 GHz	4 MB SmartCache
6	\$107.00	2	2	1.50 GHz		2 MB
	Bus_Speed	TDP	Embedded_Options_Available	Conflict_Free	Max_Memory_Size	Memory_Types
1	4 GT/s OPI	4.5 W	No	Yes	16 GB	LPDDR3-1866, DDR3L-1600
2	4 GT/s OPI	15 W	No	Yes	32 GB	DDR4-2400, LPDDR3-2133
3	4 GT/s OPI	15 W	No	Yes	32 GB	DDR4-2400, LPDDR3-2133
4	5 GT/s DMI2	130 W	No		64.23 GB	DDR3 1066/1333/1600
5	4 GT/s OPI	4.5 W	No	Yes	16 GB	LPDDR3-1866, DDR3L-1600
6	5 GT/s DMI2	15 W	No	Yes	16 GB	DDR3L 1333/1600 LPDDR3 1333/1600
	Max_nb_of_Memory_Channels	Max_Memory_Bandwidth	ECC_Memory_Supported	Processor_Graphics	Graphics_Base_Frequency	
1	2	29.8 GB/s	No	NA	300 MHz	
2	2	34.1 GB/s	No	NA	300 MHz	
3	2	34.1 GB/s	No	NA	300 MHz	
4	4	51.2 GB/s	No	NA		
5	2	29.8 GB/s	No	NA	300 MHz	
6	2	25.6 GB/s		NA	100 MHz	

Hình 1: 6 dòng đầu tiên của dữ liệu (1)



	Graphics_Max_Dynamic_Frequency	Graphics_Video_Max_Memory	Graphics_Output	Support_4k	Max_Resolution_HDMI	Max_Resolution_DP
1	1.05 GHz	16 GB	eDP/DP/HDMI/DVI	NA	4096x2304@24Hz	3840x2160@60Hz
2	1.10 GHz	32 GB	eDP/DP/HDMI/DVI	NA	4096x2304@24Hz	4096x2304@60Hz
3	1.15 GHz	32 GB	eDP/DP/HDMI/DVI	NA	4096x2304@24Hz	4096x2304@60Hz
4				NA		
5	950 MHz	16 GB	eDP/DP/HDMI/DVI	NA	4096x2304@24Hz	3840x2160@60Hz
6	800 MHz		eDP/DP/HDMI	NA		
	Max_Resolution_eDP_Integrated_Flat_Panel	DirectX_Support	OpenGL_Support	PCI_Express_Revision	PCI_Express_Configurations_	
1	3840x2160@60Hz	12	NA	3	1x4, 2x2, 1x2+2x1 and 4x1	
2	4096x2304@60Hz	12	NA	3	1x4, 2x2, 1x2+2x1 and 4x1	
3	4096x2304@60Hz	12	NA	3	1x4, 2x2, 1x2+2x1 and 4x1	
4			NA	2		
5	3840x2160@60Hz	12	NA	3	1x4, 2x2, 1x2+2x1 and 4x1	
6		11.2/12	NA	2	4x1 2x4	
	Max_nb_of_PCI_Express_Lanes	T	Intel_Hyper_Threading_Technology_	Intel_Virtualization_Technology_VTx_	Intel_64_	
1	10	100°C	Yes		Yes	
2	12	100°C	Yes		Yes	
3	12	100°C	Yes		Yes	
4	40	66.8°C	Yes		Yes	
5	10	100°C	Yes		Yes	
6	12	105°C	No		Yes	
	Instruction_Set	Instruction_Set_Extensions	Idle_States	Thermal_Monitoring_Technologies	Secure_Key	Execute_Disable_Bit
1	64-bit	SSE4.1/4.2, AVX 2.0	Yes		Yes	Yes
2	64-bit	SSE4.1/4.2, AVX 2.0	Yes		Yes	Yes
3	64-bit	SSE4.1/4.2, AVX 2.0	Yes		Yes	Yes
4	64-bit	SSE4.2, AVX, AES	Yes		Yes	Yes
5	64-bit	SSE4.1/4.2, AVX 2.0	Yes		Yes	Yes
6	64-bit	SSE4.1/4.2	Yes		Yes	Yes

Hình 2: 6 dòng đầu tiên của dữ liệu (2)

Tạo một dữ liệu mới gồm các biến chính mà ta quan tâm, lưu với tên là df. Sau đó hiển thị dữ liệu df ra màn hình để kiểm tra.

```
# Create a new dataframe containing only the columns of interest
df <- data[, c("Vertical_Segment", "Lithography", "nb_of_Cores", "nb_of_Threads",
               "Processor_Base_Frequency", "TDP", "Max_Memory_Size", "Max_Memory_Bandwidth",
               "Max_nb_of_PCI_Express_Lanes")]
# Print the first 6 rows of the new data
head(df)
```

	Vertical_Segment	Lithography	nb_of_Cores	nb_of_Threads	Processor_Base_Frequency	TDP	Max_Memory_Size	Max_Memory_Bandwidth
1	Mobile	14 nm	2	4	1.30 GHz	4.5 W	16 GB	29.8 GB/s
2	Mobile	14 nm	4	8	1.60 GHz	15 W	32 GB	34.1 GB/s
3	Mobile	14 nm	4	8	1.80 GHz	15 W	32 GB	34.1 GB/s
4	Desktop	32 nm	4	8	3.60 GHz	130 W	64.23 GB	51.2 GB/s
5	Mobile	14 nm	2	4	1.20 GHz	4.5 W	16 GB	29.8 GB/s
6	Mobile	14 nm	2	2	1.50 GHz	15 W	16 GB	25.6 GB/s
	Max_nb_of_PCI_Express_Lanes							
1	10							
2	12							
3	12							
4	40							
5	10							
6	12							

Hình 3: Dữ liệu df

## 4.2 Xử lý định dạng dữ liệu

Ta cần chuyển đổi biến Vertical\_Segment thành biến phân loại:

```
# "Convert the 'Vertical_Segment' column to a categorical variable."
df$Vertical_Segment <- as.factor(df$Vertical_Segment)
```

Tiếp theo cần định dạng lại số liệu:

```
# Function to Convert GHz and MHz to MHz:
convert_frequency <- function(frequency) {
  if (grepl("GHz", frequency)) {
    return(as.numeric(gsub(" GHz", "", frequency)) * 1000)
  } else if (grepl("MHz", frequency)) {
    return(as.numeric(gsub(" MHz", "", frequency)))
  } else {
    return(NA)
  }
}

# Apply the function to the Processor_Base_Frequency column.
df$Processor_Base_Frequency <- sapply(df$Processor_Base_Frequency, convert_frequency)
```

Tương tự với các biến còn lại. Sau khi định dạng xong ta có dữ liệu sau:

	Vertical_Segment	Lithography	nb_of_Cores	nb_of_Threads	Processor_Base_Frequency	TDP	Max_Memory_Size	Max_Memory_Bandwidth
1	Mobile	14	2	4	1300	4.5	16.00	29.8
2	Mobile	14	4	8	1600	15.0	32.00	34.1
3	Mobile	14	4	8	1800	15.0	32.00	34.1
4	Desktop	32	4	8	3600	130.0	64.23	51.2
5	Mobile	14	2	4	1200	4.5	16.00	29.8
6	Mobile	14	2	2	1500	15.0	16.00	25.6

	Max_nb_of_PCI_Express_Lanes
1	10
2	12
3	12
4	40
5	10
6	12

Hình 4: Dữ liệu sau khi xử lý định dạng

### 4.3 Xử lý dữ liệu khuyết

Thống kê số lượng và tỉ lệ dữ liệu khuyết trong từng biến:

```
# Statistics on the quantity of missing data in variables.
apply(is.na(df), 2, sum)
```

	Vertical_Segment	Lithography	nb_of_Cores	nb_of_Threads
	0	71	0	856
	Processor_Base_Frequency	TDP	Max_Memory_Size	Max_Memory_Bandwidth
	18	67	880	1136
	Max_nb_of_PCI_Express_Lanes			
	1104			

Hình 5: Số lượng dữ liệu khuyết

```
# Statistics on the percentage of missing data in variables.
apply(is.na(df), 2, mean)
```



---

Vertical_Segment	Lithography	nb_of_Cores	nb_of_Threads
0.000000000	0.031099431	0.000000000	0.374945247
Processor_Base_Frequency	TDP	Max_Memory_Size	Max_Memory_Bandwidth
0.007884363	0.029347350	0.385457731	0.497590889
Max_nb_of_PCI_Express_Lanes			
0.483574244			

Hình 6: Tỷ lệ dữ liệu khuyết

Thay thế các giá trị NA bằng trung vị của các giá trị còn lại trong cột:

---

```
# Apply the function only to numerical columns.
numeric_columns <- sapply(df, is.numeric)
df[numeric_columns] <- sapply(df[numeric_columns], replace_na_with_median)
# Convert the result back into a DataFrame
df <- as.data.frame(df)
```

---

Kiểm tra lại số lượng dữ liệu khuyết trong từng biến:

---

```
# Verify the missing data in the variables.
apply(is.na(df), 2, sum)
```

---

---

Vertical_Segment	Lithography	nb_of_Cores	nb_of_Threads
0	0	0	0
Processor_Base_Frequency	TDP	Max_Memory_Size	Max_Memory_Bandwidth
0	0	0	0
Max_nb_of_PCI_Express_Lanes			
0			

Hình 7: Dữ liệu khuyết trong từng biến

## 5 Thống kê mô tả

### 5.1 Tìm giá trị ngoại lai

Giá trị ngoại lai có thể là một giá trị phi thực tế như số tuổi âm, hoặc một giá trị khác xa với phần còn lại, một hạng mục nằm ngoài những khả năng có thể xảy ra, một địa danh không có trên bản đồ,... Các giá trị có tần xuất xảy ra vô cùng thấp trong một cột dữ liệu cũng có khả năng là một giá trị ngoại lai.

Với biến có kiểu dữ liệu là ký tự như `Vertical_Segment`, thực hiện thống kê và nhận thấy không có giá trị khác thường.

```
# Print the count of occurrences for each category.  
table(data$Vertical_Segment)
```

Desktop	Embedded	Mobile	Server
628	177	760	718

Hình 8: Số lượng xuất hiện của mỗi hạng mục

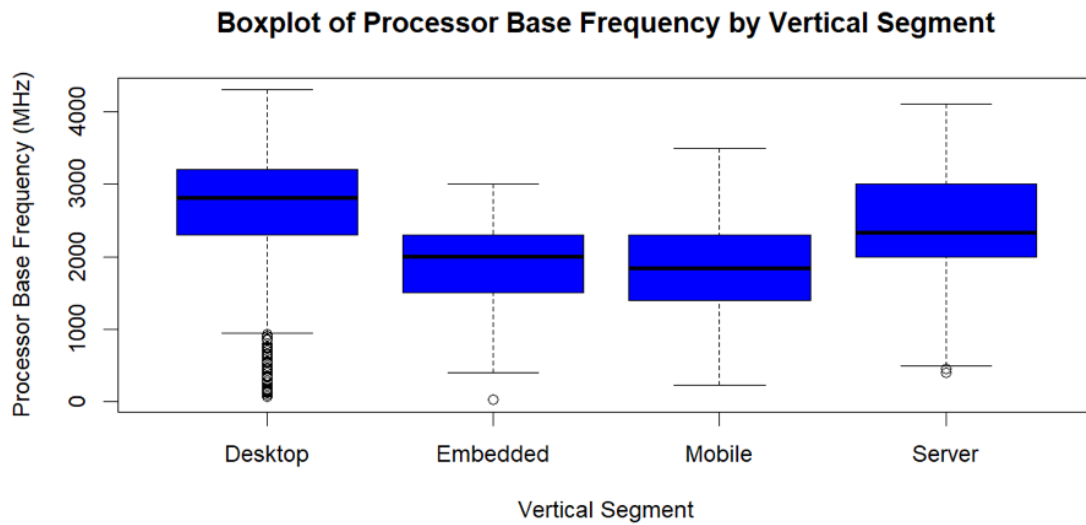
Với biến có kiểu dữ liệu là số: Có nhiều cách để nhận biết giá trị ngoại lai, trong BTL này nhóm em sẽ dùng biểu đồ hộp (boxplot). Một ý tưởng phổ biến để tìm giá trị ngoại lai là sử dụng phương pháp IQR (interquartile range). Giá trị ngoại lai thường được định nghĩa là những giá trị nằm dưới giá trị  $Q1 - 1.5IQR$  hoặc nằm trên giá trị  $Q3 + 1.5IQR$ .

```
# Calculate the Interquartile Range (IQR) for each numerical column.  
IQR_values <- sapply(df[sapply(df, is.numeric)], IQR)  
  
# Calculate the lower and upper bounds.  
lower_bounds <- sapply(df[sapply(df, is.numeric)], quantile, probs = 0.25) - 1.5 *  
  IQR_values  
upper_bounds <- sapply(df[sapply(df, is.numeric)], quantile, probs = 0.75) + 1.5 *  
  IQR_values
```

Tìm giá trị ngoại lai

```
# Identify outlier values.  
outliers <- lapply(names(df), function(i) {  
  if (is.numeric(df[[i]])) {  
    df[[i]] < lower_bounds[i] | df[[i]] > upper_bounds[i]  
  } else {  
    rep(FALSE, length(df[[i]]))  
  }  
})
```

Vẽ biểu đồ boxplot để thấy rõ các dữ liệu ngoại lai, các dữ liệu này được xác định là các dấu chấm ở 2 đầu biểu đồ



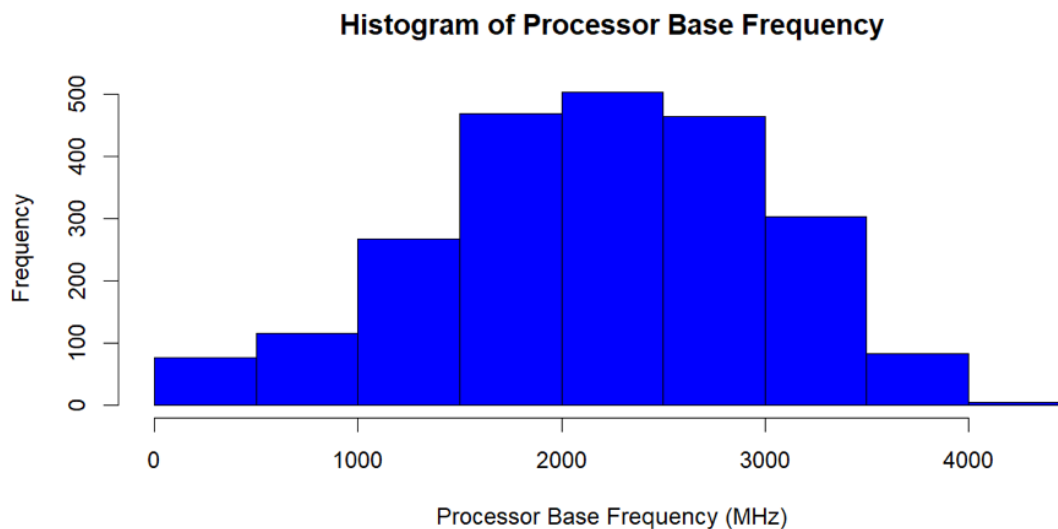
Hình 9: Biểu đồ Boxplot của biến P\_B\_Frequency chia theo biến Vertical\_Segment

**Nhận xét:** Nhóm máy tính để bàn có 1 số giá trị ngoại lai ( $< 950$  MHz), nhóm thiết bị nhúng và máy chủ có số lượng ngoại lai ít và không thể nhìn thấy giá trị ngoại lai nào ở nhóm điện thoại.

## 5.2 Vẽ đồ thị

Đồ thị Histogram thể thể hiện phân phối của các biến

```
# Draw a histogram
hist(df$Processor_Base_Frequency, main = "Histogram of Processor Base Frequency",
     xlab = "Processor Base Frequency (MHz)", col = "blue", border = "black")
```



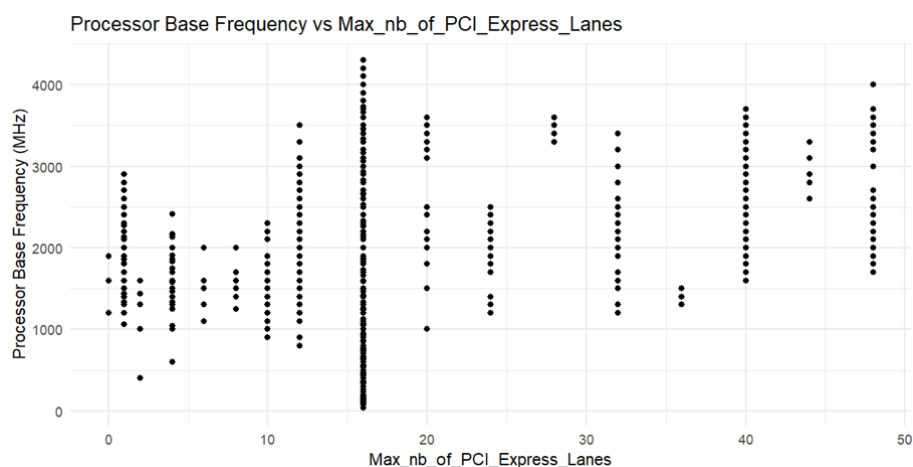
Hình 10: Đồ thị Histogram của Tần suất Cơ bản của Bộ xử lý

**Nhận xét:** Biến Processor Base Frequency tập trung ở giá trị 2 đến 3 GHz. Giá trị tần số này có thể đáp ứng được nhu cầu của đại bộ phận con người với CPU. Đi sâu hơn vào phân tích các thiết bị thì ở biểu đồ boxplot (hình 9) cho thấy nhóm máy tính để bàn có trung vị và khoảng dữ liệu lớn hơn 3 nhóm còn lại, giá trị không lệch về một phía nào quá nhiều. Nhóm điện thoại cũng có một phân phối khá đều. Nhóm thiết bị nhúng có khoảng dữ liệu nhỏ nhất, giá trị lệch về phía dưới trung vị. Ngược lại nhóm máy chủ lại có khoảng dữ liệu rộng và lệch về phía trên trung vị.

## Vẽ biểu đồ phân tán thể hiện phân phối của Processor\_Base\_Frequency theo các biến

Biểu đồ phân tán:

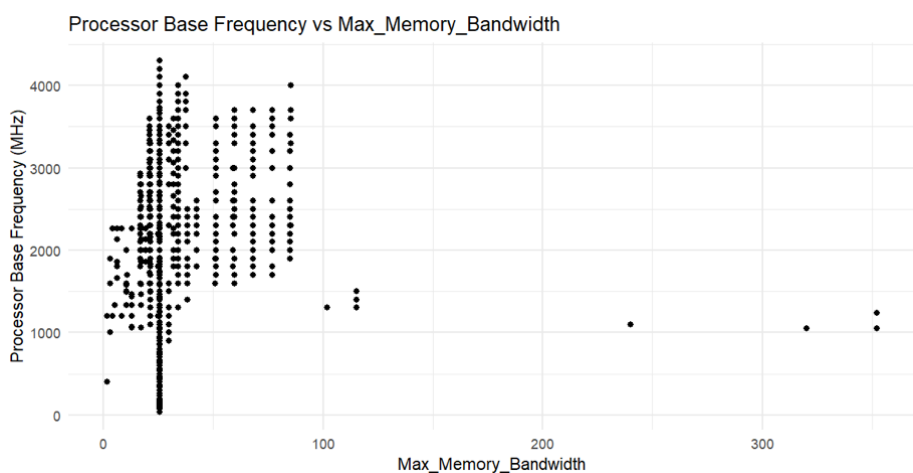
```
# Draw a scatter plot for each column.
for (col_name in names(numeric_columns)) {
  p <- ggplot(df, aes(x = !!sym(col_name), y = Processor_Base_Frequency)) +
    geom_point() +
    labs(x = col_name, y = "Processor Base Frequency (MHz)", title = paste("Processor
      Base Frequency vs", col_name)) +
    theme_minimal()
  print(p)
}
```



Hình 11: Biểu đồ phân tán (1)

**Nhận xét:**

- Biểu đồ phân tán cho thấy mối quan hệ giữa tần số cơ bản của bộ xử lý và số làn PCI Express tối đa. Nhìn chung, có một mối quan hệ thuận chiều (không mạnh) giữa hai thông số này, nghĩa là các bộ xử lý có tần số cơ bản cao hơn thường có số làn PCI Express tối đa cao hơn.
- Tuy nhiên, biểu đồ cũng cho thấy có một số điểm nằm ngoài xu hướng chung. Một số bộ xử lý có tần số cơ bản cao nhưng số làn PCI Express tối đa thấp, và ngược lại.



Hình 12: Biểu đồ phân tán (2)

**Nhận xét:**

- Biểu đồ phân tán cho thấy mối quan hệ giữa tần số cơ bản của bộ xử lý và băng thông bộ nhớ tối đa. Nhìn chung, hai yếu tố này có xu hướng tăng cùng nhau.
- Tuy nhiên, cũng có một số điểm dữ liệu nằm dưới đường xu hướng chung.

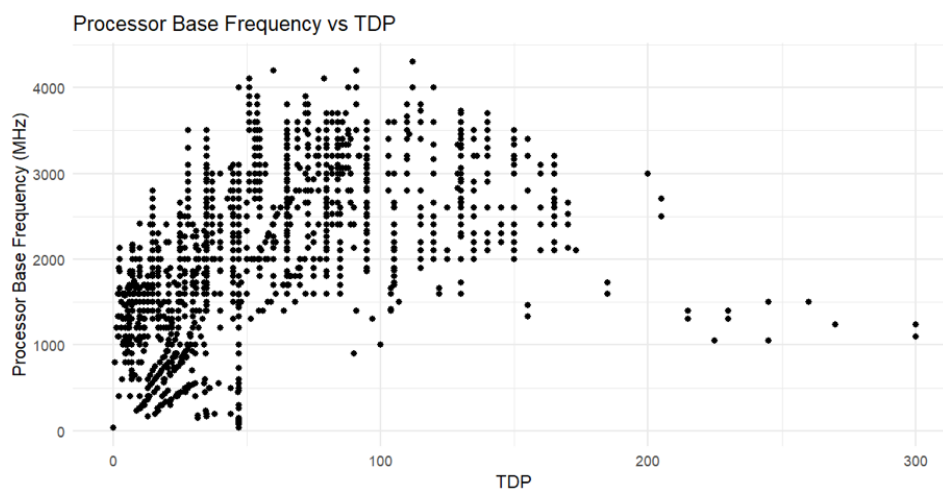




Hình 13: Biểu đồ phân tán (3)

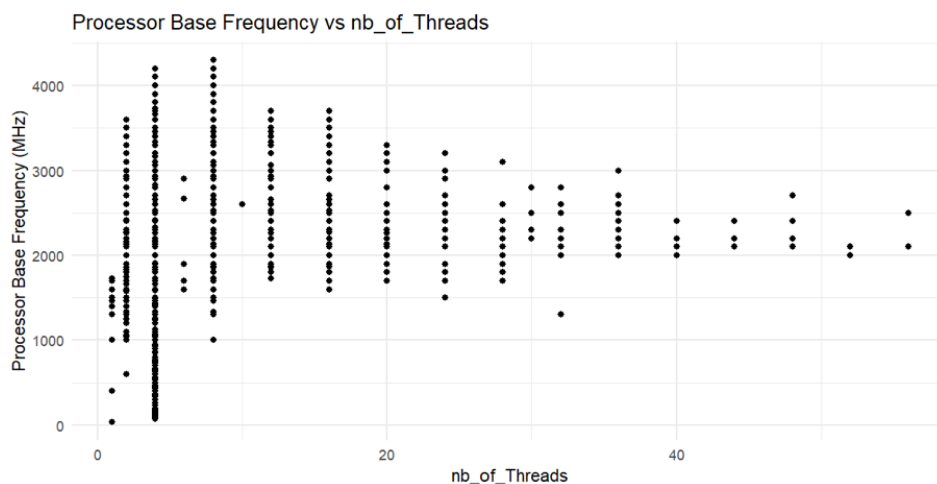
**Nhận xét:**

- Nhìn chung 2 biến này không có mối quan hệ mật thiết, tức là không có mối tương quan giữa kích thước bộ nhớ tối đa và tần số cơ bản CPU.



Hình 14: Biểu đồ phân tán (4)

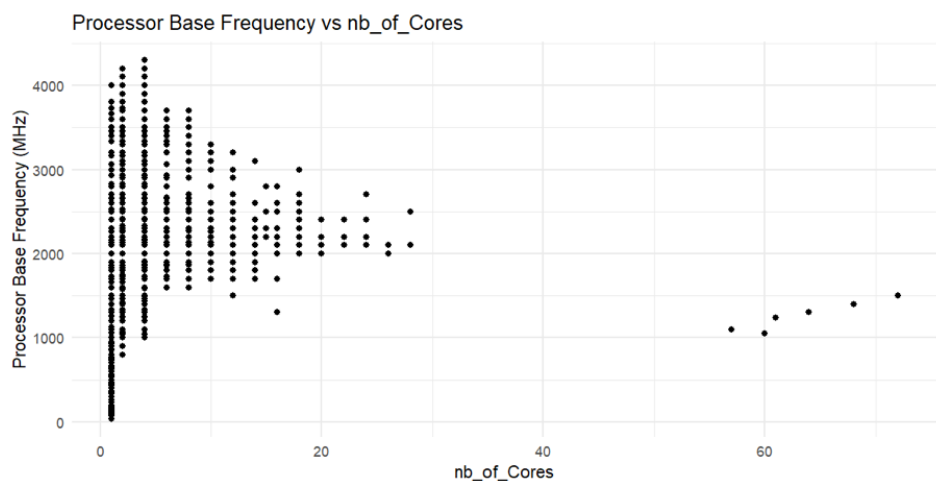
**Nhận xét:** Nhìn chung, có một mối quan hệ thuận yếu hai biến này, nghĩa là TDP càng cao thì tần số cơ bản cũng càng cao, nhưng sự tăng của tần số cơ bản không phụ thuộc nhiều vào TDP.



Hình 15: Biểu đồ phân tán (5)

**Nhận xét:**

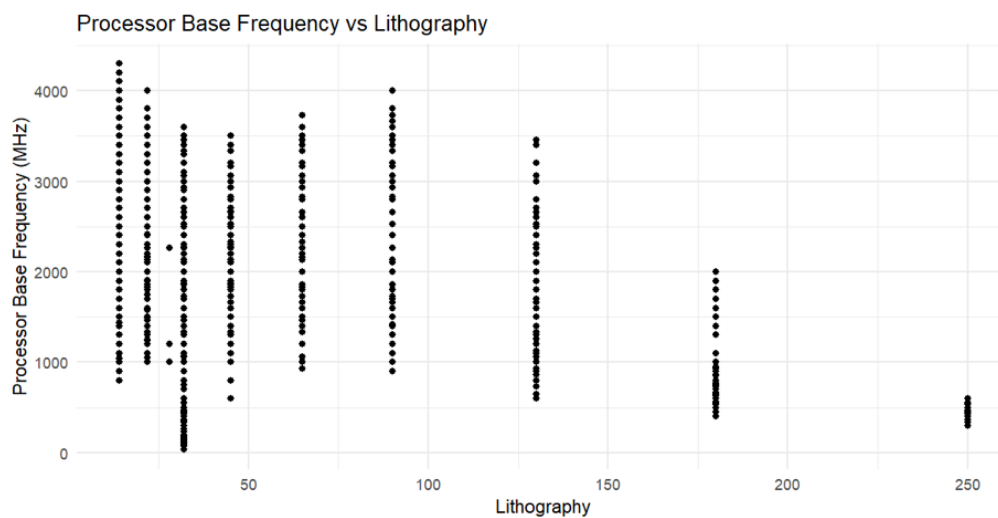
Không có sự tương quan nhiều giữa số lượng luồng và tần số cơ bản bộ xử lý



Hình 16: Biểu đồ phân tán (6)

**Nhận xét:**

Nhìn chung không có mối tương quan giữa 2 biến, nhưng có thể dự đoán khi số lượng nhân CPU tăng thì tần số cơ bản có xu hướng tập trung quanh giá trị trung bình (ngoại trừ 1 số điểm nằm ngoài đường xu hướng).



### 5.3 Các giá trị thống kê mô tả

Tính các giá trị thống kê theo từng nhóm dữ liệu được phân theo biến `Vertical_Segment` và thống kê trên toàn bộ dữ liệu

```
# Descriptive statistics for numerical data.  
summary(df)
```

```
Vertical_Segment  Lithography    nb_of_Cores    nb_of_Threads  Processor_Base_Frequency  
Desktop :628      Min.   : 14.00    Min.   : 1.000    Min.   : 1.000    Min.   : 32  
Embedded:177      1st Qu.: 22.00    1st Qu.: 1.000    1st Qu.: 4.000    1st Qu.:1660  
Mobile :760       Median : 32.00    Median : 2.000    Median : 4.000    Median :2260  
Server :718       Mean   : 48.46    Mean   : 4.067    Mean   : 6.955    Mean  :2223  
          3rd Qu.: 65.00    3rd Qu.: 4.000    3rd Qu.: 8.000    3rd Qu.:2800  
          Max.   :250.00    Max.   :72.000    Max.   :56.000    Max.   :4300  
  
          TDP      Max_Memory_Size  Max_Memory_Bandwidth  Max_nb_of_PCI_Express_Lanes  
Min.   : 0.025    Min.   : 1.0    Min.   : 1.60    Min.   : 0.00  
1st Qu.: 26.800    1st Qu.: 32.0    1st Qu.: 25.60    1st Qu.:16.00  
Median : 47.000    Median : 32.0    Median : 25.60    Median :16.00  
Mean   : 59.853    Mean   : 179.7    Mean   : 30.36    Mean   :18.27  
3rd Qu.: 84.000    3rd Qu.: 32.0    3rd Qu.: 25.60    3rd Qu.:16.00  
Max.   :300.000    Max.   :4198.4    Max.   :352.00    Max.   :48.00
```

Hình 18: Thống kê các dữ liệu số

```
by(df, df$Vertical_Segment, summary)
```

```
df$Vertical_Segment: Desktop  
Vertical_Segment  Lithography    nb_of_Cores    nb_of_Threads  Processor_Base_Frequency  TDP      Max_Memory_Size  
Desktop :628      Min.   : 14.00    Min.   : 1.000    Min.   : 1.000    Min.   : 75    Min.   : 4.00    Min.   : 4.00  
Embedded: 0      1st Qu.: 22.00    1st Qu.: 1.000    1st Qu.: 4.000    1st Qu.:2300    1st Qu.: 35.00    1st Qu.: 32.00  
Mobile : 0      Median : 32.00    Median : 2.000    Median : 4.000    Median :2800    Median : 65.00    Median : 32.00  
Server : 0      Mean   : 61.48    Mean   : 2.339    Mean   : 4.454    Mean :2573    Mean : 64.75    Mean : 35.69  
          3rd Qu.: 90.00    3rd Qu.: 4.000    3rd Qu.: 4.000    3rd Qu.:3200    3rd Qu.: 84.00    3rd Qu.: 32.00  
          Max.   :250.00    Max.   :18.000    Max.   :36.000    Max.   :4300    Max.   :165.00    Max.   :128.00  
  
Max_Memory_Bandwidth  Max_nb_of_PCI_Express_Lanes  
Min.   : 6.40    Min.   : 4.00  
1st Qu.:25.60    1st Qu.:16.00  
Median :25.60    Median :16.00  
Mean   :25.46    Mean   :16.48  
3rd Qu.:25.60    3rd Qu.:16.00  
Max.   :68.00    Max.   :44.00
```

Hình 19: Thống kê các dữ liệu số (2)

```
df$Vertical_Segment: Embedded  
Vertical_Segment  Lithography    nb_of_Cores    nb_of_Threads  Processor_Base_Frequency  TDP      Max_Memory_Size  
Desktop : 0      Min.   :14.00    Min.   : 1.000    Min.   : 1.000    Min.   : 32    Min.   : 0.025    Min.   : 2.0  
Embedded:177      1st Qu.:22.00    1st Qu.: 2.000    1st Qu.: 2.000    1st Qu.:1500    1st Qu.: 17.000    1st Qu.: 16.0  
Mobile : 0      Median :22.00    Median : 2.000    Median : 4.000    Median :2000    Median : 35.000    Median : 32.0  
Server : 0      Mean   :25.68    Mean   : 3.859    Mean   : 6.814    Mean :1857    Mean : 35.289    Mean : 171.4  
          3rd Qu.:32.00    3rd Qu.: 4.000    3rd Qu.: 8.000    3rd Qu.:2300    3rd Qu.: 47.000    3rd Qu.: 64.0  
          Max.   :45.00    Max.   :22.000    Max.   :44.000    Max.   :3000    Max.   :145.000    Max.   :1577.0  
  
Max_Memory_Bandwidth  Max_nb_of_PCI_Express_Lanes  
Min.   : 1.60    Min.   : 1.00  
1st Qu.:25.60    1st Qu.: 8.00  
Median :25.60    Median :16.00  
Mean   :30.65    Mean   :16.38  
3rd Qu.:34.10    3rd Qu.:16.00  
Max.   :76.80    Max.   :40.00
```

Hình 20: Thống kê các dữ liệu số (3)

```
-----
df$Vertical_Segment: Mobile
Vertical_Segment  Lithography  nb_of_Cores  nb_of_Threads  Processor_Base_Frequency  TDP  Max_Memory_Size  Max_Memory_Bandwidth
Desktop : 0      Min. : 14.00  Min. : 1.000  Min. : 1.000  Min. : 233      Min. : 0.65  Min. : 1.00  Min. : 1.60
Embedded: 0      1st Qu.: 22.00  1st Qu.: 1.000  1st Qu.: 4.000  1st Qu.: 1400    1st Qu.: 15.00  1st Qu.: 16.00  1st Qu.: 25.60
Mobile : 760     Median: 32.00  Median: 2.000  Median: 4.000  Median: 1830    Median: 25.00  Median: 32.00  Median: 25.60
Server : 0      Mean : 46.98  Mean : 2.096  Mean : 4.158  Mean : 1849    Mean : 26.27  Mean : 25.04  Mean : 24.44
              3rd Qu.: 65.00  3rd Qu.: 2.000  3rd Qu.: 4.000  3rd Qu.: 2300    3rd Qu.: 35.00  3rd Qu.: 32.00  3rd Qu.: 25.60
              Max. : 180.00  Max. : 4.000  Max. : 8.000  Max. : 3500    Max. : 88.00  Max. : 64.00  Max. : 34.10
Max_nb_of_PCI_Express_Lanes
Min. : 0.00
1st Qu.: 12.00
Median : 16.00
Mean : 13.73
3rd Qu.: 16.00
Max. : 20.00
```

Hình 21: Thống kê các dữ liệu số (4)

```
-----
df$Vertical_Segment: Server
Vertical_Segment  Lithography  nb_of_Cores  nb_of_Threads  Processor_Base_Frequency  TDP  Max_Memory_Size
Desktop : 0      Min. : 14.00  Min. : 1.000  Min. : 2.00  Min. : 400      Min. : 6.00  Min. : 6.0
Embedded: 0      1st Qu.: 14.00  1st Qu.: 2.000  1st Qu.: 4.00  1st Qu.: 2000    1st Qu.: 66.50  1st Qu.: 32.0
Mobile : 0      Median: 32.00  Median: 4.000  Median: 8.00  Median: 2330    Median: 95.00  Median: 128.0
Server : 718     Mean : 44.25  Mean : 7.714  Mean : 12.14  Mean : 2401     Mean : 97.17  Mean : 471.5
              3rd Qu.: 45.00  3rd Qu.: 8.000  3rd Qu.: 16.00  3rd Qu.: 3000    3rd Qu.: 130.00  3rd Qu.: 768.0
              Max. : 250.00  Max. : 72.000  Max. : 56.00  Max. : 4100     Max. : 300.00  Max. : 4198.4
Max_Memory_Bandwidth  Max_nb_of_PCI_Express_Lanes
Min. : 10.60  Min. : 4.00
1st Qu.: 25.60  1st Qu.: 16.00
Median : 25.60  Median : 16.00
Mean : 40.84  Mean : 25.11
3rd Qu.: 51.20  3rd Qu.: 40.00
Max. : 352.00  Max. : 48.00
```

Hình 22: Thống kê các dữ liệu số (5)

## 6 Thống kê suy diễn

### 6.1 Phương pháp ANOVA

**Định nghĩa:** ANOVA là viết tắt của "Analysis of Variance" (Phân tích phương sai) và là một phương pháp thống kê được sử dụng để kiểm tra sự khác biệt giữa các giá trị trung bình của ba hoặc nhiều nhóm khác nhau. ANOVA phân tích biến động dữ liệu để xác định xem sự biến động đó có xuất phát từ sự khác biệt giữa các nhóm hay không.

**Phân loại:** Có hai dạng chính của ANOVA: ANOVA một yếu tố và ANOVA hai yếu tố.

**Điều kiện:**

- Các nhóm đều có phân phối chuẩn.
- Phương sai đồng nhất giữa các nhóm.
- Các mẫu quan sát lấy độc lập.

#### 6.1.1 Anova 1 yếu tố

**Định nghĩa:** Phân tích phương sai một yếu tố - One way Anova là một mô hình phân tích phương sai cho một yếu tố, so sánh trung bình biến ngẫu nhiên ở từng nhóm khác nhau. Dựa vào các mẫu quan sát thu được, các nhóm được phân biệt qua các yếu tố đang xem xét.

**Hypotheses:**

$H_0$ : Các giá trị trung bình bằng nhau (Không có sự khác biệt).

$H_1$ : Có ít nhất một sự khác biệt về giá trị trung bình.

### 6.1.2 Anova 2 yếu tố

**Định nghĩa:** Phân tích Anova 2 yếu tố hay phân tích Anova 2 chiều – Two way anova là việc ta xem xét cùng lúc hai yếu tố nguyên nhân (dưới dạng dữ liệu định tính) ảnh hưởng đến yếu tố kết quả (dưới dạng dữ liệu định lượng) đang nghiên cứu. So với phân tích Anova một yếu tố thì phân tích Anova hai yếu tố mang lại nhiều giá trị hơn cho nghiên cứu.

**Phân loại:** Anova 2 yếu tố có lặp và Anova 2 yếu tố không lặp.

**Trong bài báo cáo này, nhóm tác giả sử dụng mô hình Anova 2 yếu tố có lặp để thống kê dữ liệu**

**Mục tiêu chung:** Kiểm định được sự khác biệt giữa giá trị trung bình giữa ba hoặc nhiều nhóm được lấy độc lập của biến phụ thuộc (Processor\_Base\_Frequency) giữa trên sự thay đổi của 2 yếu tố làm ảnh hưởng đến hiệu năng của CPU là Lithography và TDP .

Biến phụ thuộc: Processor\_Base\_Frequency

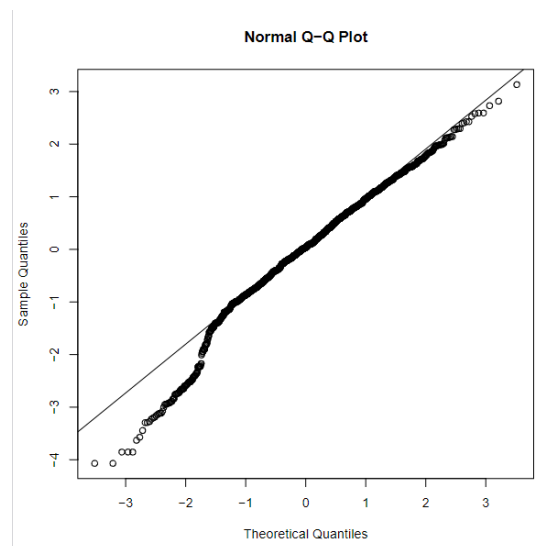
Biến độc lập: Lithography, TDP

## Kiểm định các giả thuyết cho phân tích ANOVA

- **Giả định 1:**

**Mục tiêu:** Kiểm định dữ liệu phân phối chuẩn.

```
av_residual <- rstandard(aov(Processor_Base_Frequency ~ Lithography*TDP,data = df))  
qqnorm(av_residual)  
qqline(av_residual)
```



Hình 23: Kiểm tra phân phối chuẩn giữa biến Processor\_Base\_Frequency với từng nhóm của biến Lithography và TDP

**Nhận xét:** Hầu như các quan sát đều nằm trên đường thẳng .

**Kết luận giả định 1:** Các biến Lithography và TDP tuân theo phân phối chuẩn.

- **Giả định 2:**

**Mục tiêu:** Kiểm tra đồng nhất phương sai.

Đặt:  $H_0$  là biến Processor\_Base\_Frequency theo từng nhóm của biến Lithography và TDP đồng nhất phương sai.

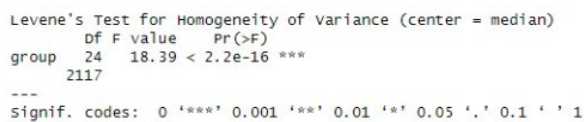
$H_1$  là biến Processor\_Base\_Frequency theo từng nhóm của biến Lithography và TDP không đồng nhất phương sai.

---

```
leveneTest(Processor_Base_Frequency ~ Lithography*TDP, data = df)
```

---

**Kết quả:**



```
Levene's Test for Homogeneity of Variance (center = median)
      Df F value    Pr(>F)
group  24  18.39 < 2.2e-16 ***
  2117
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Hình 24: Kết quả kiểm định Levene Test cho Lithography và TDP

**Nhận xét:** Vì giá trị  $\Pr(>F) = 2.2 \times 10^{-16} < 0.05$  nên bác bỏ giả thuyết  $H_0$ , chấp nhận  $H_1$ .

**Kết luận giả định 2:** Các biến Lithography và TDP không đồng nhất phương sai, tuy nhiên phân tích ANOVA vẫn có thể thực hiện được khi cỡ mẫu của các nhóm nghiên cứu bằng nhau.

- **Tính Anova**

**Mục tiêu:** Tính toán được sự phụ thuộc của tần số hoạt động của vi xử lý (Processor\_Base\_Frequency) vào kích thước của transistor (Lithography) và lượng nhiệt tối đa mà hệ thống làm mát cần loại bỏ khi CPU hoạt động (TDP).

Đặt:  $H_0$  là biến Processor\_Base\_Frequency tuân theo 2 biến độc lập Lithography và TDP theo từng nhóm bằng nhau.

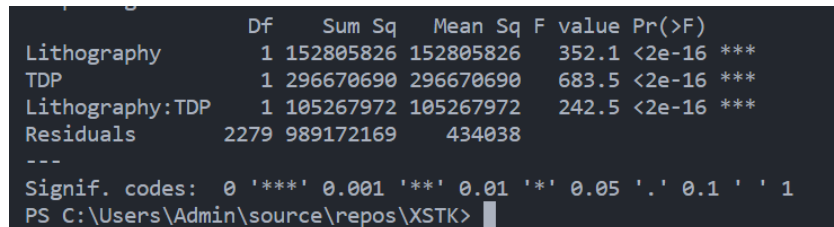
$H_1$  là biến Processor\_Base\_Frequency tuân theo 2 biến độc lập Lithography và TDP theo từng nhóm khác nhau.

**Kết quả:**

---

```
model <- aov(Processor_Base_Frequency ~ Lithography*TDP, data = df)
summary(model)
```

---



```
              Df    Sum Sq   Mean Sq F value Pr(>F)
Lithography    1 152805826 152805826   352.1 <2e-16 ***
TDP             1 296670690 296670690   683.5 <2e-16 ***
Lithography:TDP 1 105267972 105267972   242.5 <2e-16 ***
Residuals     2279 989172169    434038
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
PS C:\Users\Admin\source\repos\XSTK>
```

Hình 25: Kết quả chạy ANOVA

**Nhận xét:** Vì  $\Pr(>F) = 2 \times 10^{-16} < 0,05$  nên bác bỏ  $H_0$ , chấp nhận  $H_1$

**Kết luận chung:** Sự ảnh hưởng của Lithography và TDP đến tần số hoạt động của CPU là khác nhau ở từng nhóm.

## 6.2 Hồi quy đa tuyến tính

### *Định nghĩa hồi quy tuyến tính đa biến*

- Mô hình hồi quy tuyến tính đa biến được sử dụng để dự đoán giá trị của một biến phụ thuộc (hoặc biến mục tiêu) dựa trên nhiều biến độc lập (hoặc biến dự đoán). Mục tiêu là xác định mối quan hệ tuyến tính giữa các biến độc lập và biến phụ thuộc.

### 6.2.1 Xây dựng mô hình hồi quy tuyến tính

- Mục tiêu:** Ở đây, chúng ta muốn phân tích những yếu tố như: nb\_of\_Cores, TDP, Max\_Memory\_Size, Max\_np\_of\_PCI\_Express\_Lanes, Max\_Memory\_Bandwidth, Lithography, np\_of\_threads sẽ tác động như thế nào đến tần số của bộ xử lý CPU (Processor\_Base\_Frequency)
- Biến phụ thuộc: Processor\_Base\_Frequency
- Biến độc lập: Các biến còn lại
- Mô hình của chúng ta có thể được biểu diễn dưới dạng hàm:

$$x = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n$$

Trong đó:

- $\beta_0, \beta_1, \dots, \beta_n$  là các hệ số mô hình.
- $x$  là biến phụ thuộc.
- $X_1, X_2, \dots, X_n$  là các biến đầu vào của mô hình.

Ta thực hiện ước lượng các hệ số  $\beta_i, i = 0 \dots 9$  dựa trên tập dữ liệu x.

```
# Build a linear regression model.
model <- lm(Processor_Base_Frequency ~ ., data = df)
# Print the results.
summary(model)
```

### 6.2.2 Kiểm định hệ số hồi quy:

- Giả thuyết H0:  $\beta_i = 0, i = 1, 2 \dots$  : Hệ số hồi quy không có ý nghĩa thống kê.
- Giả thuyết H1:  $\beta_i \neq 0, i = 1, 2 \dots$  : Hệ số hồi quy có ý nghĩa thống kê.
- Vì pvalue ứng với nb\_of\_Threads, Max\_Memory\_Bandwidth, Max\_nb\_of\_PCI\_Express\_Lanes lớn hơn mức ý nghĩa 5% nên ta chưa bác bỏ H0. Tức hệ số hồi quy ứng với biến này không có ý nghĩa thống kê  $\rightarrow$  Hệ số ứng với các biến này bằng 0.
- Vì pvalue ứng với các biến còn lại bé hơn mức ý nghĩa 5% nên ta bác bỏ được H0. Tức hệ số hồi quy ứng với các biến còn lại có ý nghĩa thống kê.



```
Call:
lm(formula = Processor_Base_Frequency ~ ., data = df)

Residuals:
    Min       1Q   Median       3Q      Max
-2561.15  -247.91    32.55   335.20  1830.53

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   2304.93367    45.55605   50.596 < 2e-16 ***
Vertical_SegmentEmbedded -431.64695    51.34026  -8.408 < 2e-16 ***
Vertical_SegmentMobile -288.11917    34.89218  -8.257 2.49e-16 ***
Vertical_SegmentServer -324.64934    34.58835  -9.386 < 2e-16 ***
Lithography     -8.04624     0.28226 -28.507 < 2e-16 ***
nb_of_Cores    -66.11395     3.44773 -19.176 < 2e-16 ***
nb_of_Threads   -2.92949     3.01148  -0.973    0.331
TDP             14.96380     0.45519   32.874 < 2e-16 ***
Max_Memory_Size  -0.17512     0.03723  -4.704 2.71e-06 ***
Max_Memory_Bandwidth -0.23178     0.83924  -0.276    0.782
Max_nb_of_PCI_Express_Lanes -1.57706     1.97554  -0.798    0.425
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 558.4 on 2272 degrees of freedom
Multiple R-squared:  0.5411,    Adjusted R-squared:  0.5391
F-statistic: 267.9 on 10 and 2272 DF,  p-value: < 2.2e-16
```

Hình 26: Mô hình hồi quy tuyến tính

- Tiến hành xây dựng mô hình hồi quy loại bỏ biến không có ý nghĩa thống kê.

```
# Rebuild the linear regression model.
model <- lm(Processor_Base_Frequency ~ Vertical_Segment + Lithography + nb_of_Cores +
  TDP + Max_Memory_Size, data = df)
summary(model)
```

Dựa vào các hệ số mô hình, phương trình hồi quy tuyến tính có thể được viết như sau:

$$\begin{aligned} \text{Processor\_Base\_Frequency} = & \\ & \mathbf{2304.94} - \mathbf{431.65} \times \text{Vertical\_SegmentEmbedded} \\ & - \mathbf{288.12} \times \text{Vertical\_SegmentMobile} \\ & - \mathbf{324.65} \times \text{Vertical\_SegmentServer} \\ & - \mathbf{8.05} \times \text{Lithography} \\ & - \mathbf{66.11} \times \text{nb\_of\_Cores} \\ & + \mathbf{14.96} \times \text{TDP} \\ & - \mathbf{0.18} \times \text{Max\_Memory\_Size} \end{aligned} \quad (2)$$

#### Coefficients:

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	2273.5820	38.5814	58.929	< 2e-16	***
Vertical_SegmentEmbedded	-436.3572	50.9828	-8.559	< 2e-16	***
Vertical_SegmentMobile	-289.5184	34.6253	-8.361	< 2e-16	***
Vertical_SegmentServer	-334.8504	34.1332	-9.810	< 2e-16	***
Lithography	-8.0102	0.2803	-28.575	< 2e-16	***
nb_of_Cores	-67.5603	2.5526	-26.467	< 2e-16	***
TDP	14.7916	0.4410	33.541	< 2e-16	***
Max_Memory_Size	-0.2107	0.0304	-6.930	5.46e-12	***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 558.5 on 2275 degrees of freedom  
Multiple R-squared: 0.5405, Adjusted R-squared: 0.539  
F-statistic: 382.2 on 7 and 2275 DF, p-value: < 2.2e-16

Hình 27: Xây dựng lại mô hình hồi quy tuyến tính

### 6.2.3 Kiểm tra giả định của mô hình hồi quy

Các giả định cần kiểm tra:

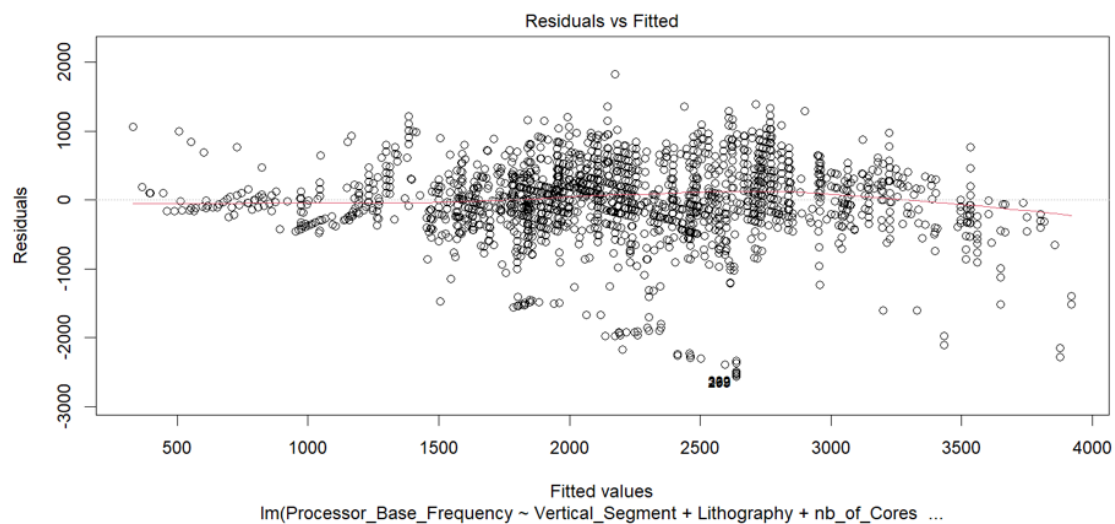
- Tính tuyến tính của dữ liệu: mối quan hệ giữa biến dự báo X và biến phụ thuộc Y được giả sử là tuyến tính.
- Sai số có phân phối chuẩn.
- Sai số có kỳ vọng bằng 0.
- Phương sai của các sai số là hằng số.  $\epsilon_n \sim N(0, \sigma_2)$ .
- Các sai số  $\epsilon_1, \dots, \epsilon_n$  độc lập với nhau.

Ta có thể vẽ các biểu đồ để kiểm tra giả định:

```
# Draw a graph to check assumptions.  
plot(model)
```

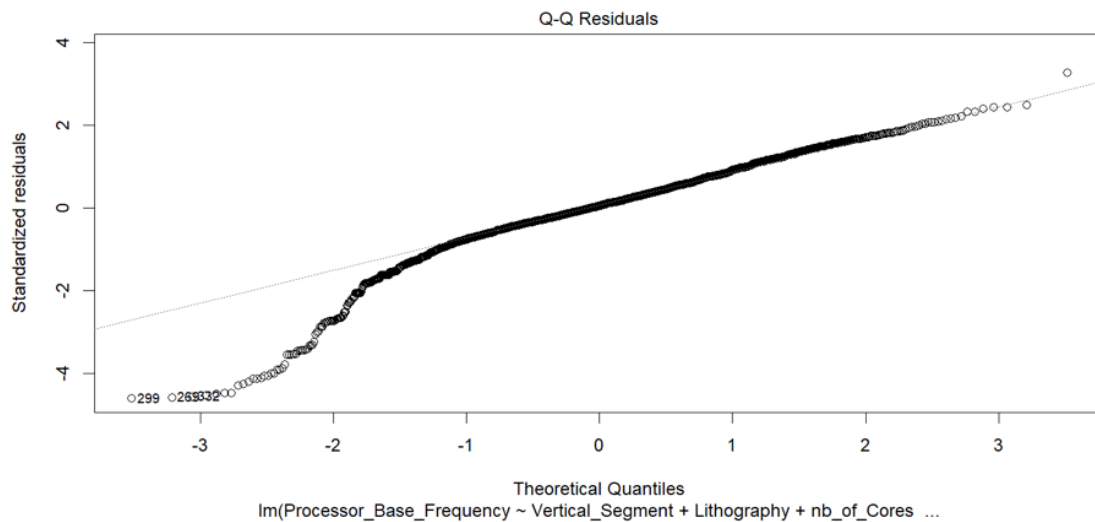
#### Nhận xét:

- Đồ thị này vẽ các giá trị dự báo với các giá trị sai số tương ứng, dùng để kiểm tra giả định các sai số có kỳ vọng bằng 0 và tính tuyến tính của dữ liệu.
- Trục tung biểu thị giá trị của sai số, trục hoành biểu thị giá trị tiên lượng của biến phụ thuộc. Nếu như giả thiết về tính tuyến tính của dữ liệu KHÔNG thỏa, ta sẽ quan sát thấy rằng đường màu đỏ trên đồ thị sẽ phân bố theo một hình mẫu (pattern) đặc trưng nào đó (ví dụ parabol). Nếu đường màu đỏ trên đồ thị phân tán là đường thẳng nằm ngang mà không phải là đường cong, thì giả thiết tính tuyến tính của dữ liệu được thỏa mãn. Giả thiết sai số có kỳ vọng bằng 0 thỏa mãn nếu sai số phân tán đều so với đường nằm ngang (ứng với sai số = 0).



Hình 28: Đồ thị kiểm tra giả định

- Đường màu đỏ chưa phải là đường thẳng nên tính tuyến tính chưa thỏa mãn giả định mối quan hệ tuyến tính giữa X và Y. Các sai số phân tán ngẫu nhiên đều quanh đường  $\text{Residuals} = 0$ , nên giả định các sai số có kỳ vọng bằng 0 thỏa mãn.

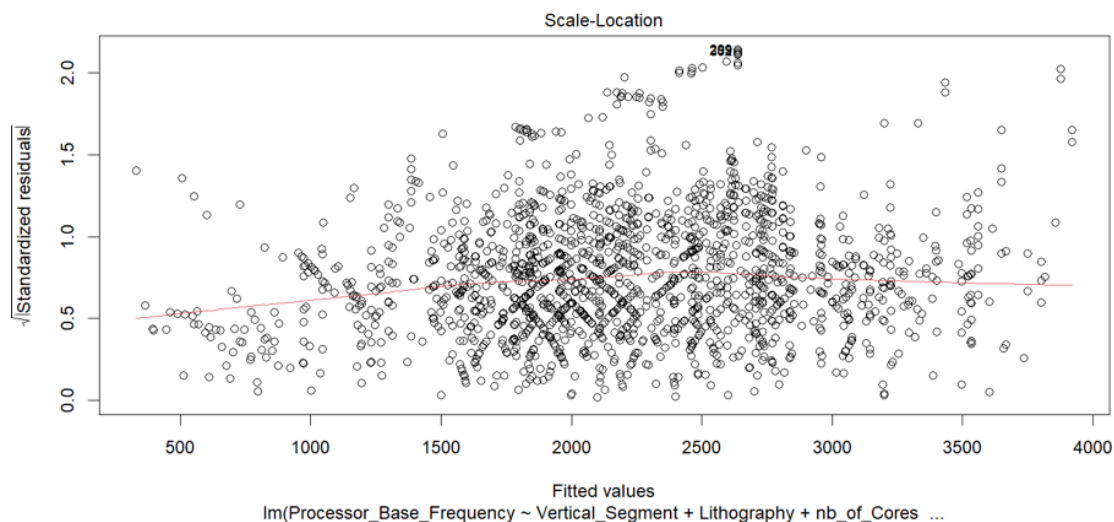


Hình 29: Đồ thị kiểm tra giả định (2)

#### Nhận xét:

Đồ thị này vẽ các giá trị sai số được chuẩn hoá, cho phép kiểm tra giả định về phân phối chuẩn của các sai số.

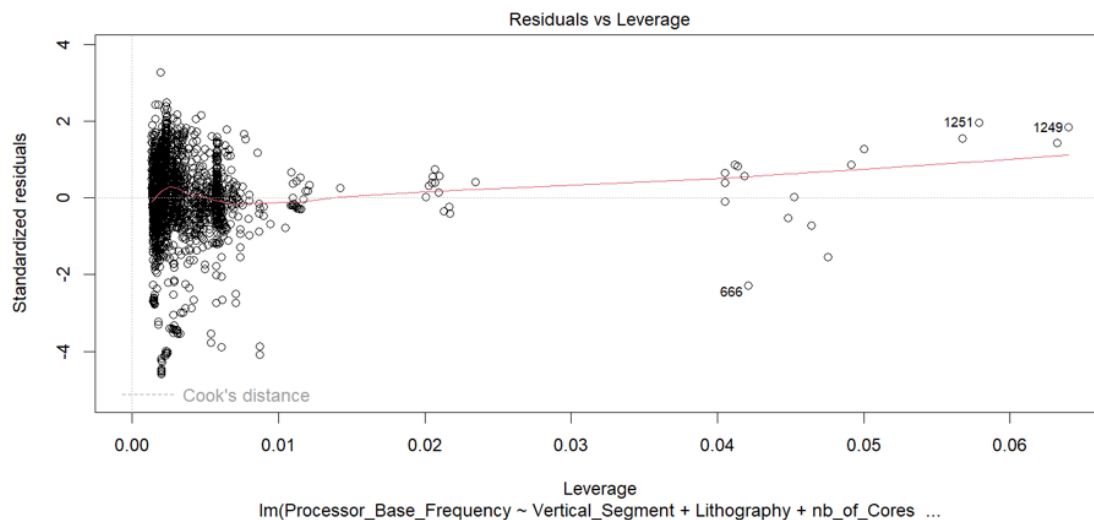
- Dựa trên đồ thị ta thấy các sai số đa phần tập trung nằm trên đường thẳng kỳ vọng phân phối chuẩn nhưng không hoàn toàn nằm trên một đường thẳng. Điều này cho thấy rằng dữ liệu có thể tuân theo phân phối chuẩn không hoàn toàn.
- Cụ thể, các điểm dữ liệu có xu hướng tập trung ở phần đuôi của phân phối, đặc biệt là ở đuôi bên trái. Điều này cho thấy rằng dữ liệu có thể bị lệch trái (skewed to the left).



Hình 30: Đồ thị kiểm tra giả định (3)

**Nhận xét:**

Đồ thị này vẽ căn bậc hai của các giá trị sai số được chuẩn hóa với các giá trị dự báo, được dùng để kiểm tra giả định phương sai của các sai số là hằng số. Trục tung là căn bậc hai của giá trị sai số (đã được chuẩn hóa), trục hoành là giá trị tiên lượng của biến phụ thuộc từ mô hình. Nếu như đường màu đỏ trên đồ thị là đường thẳng nằm ngang và các điểm thặng dư.



Hình 31: Đồ thị kiểm tra giả định (4)

**Nhận xét:**

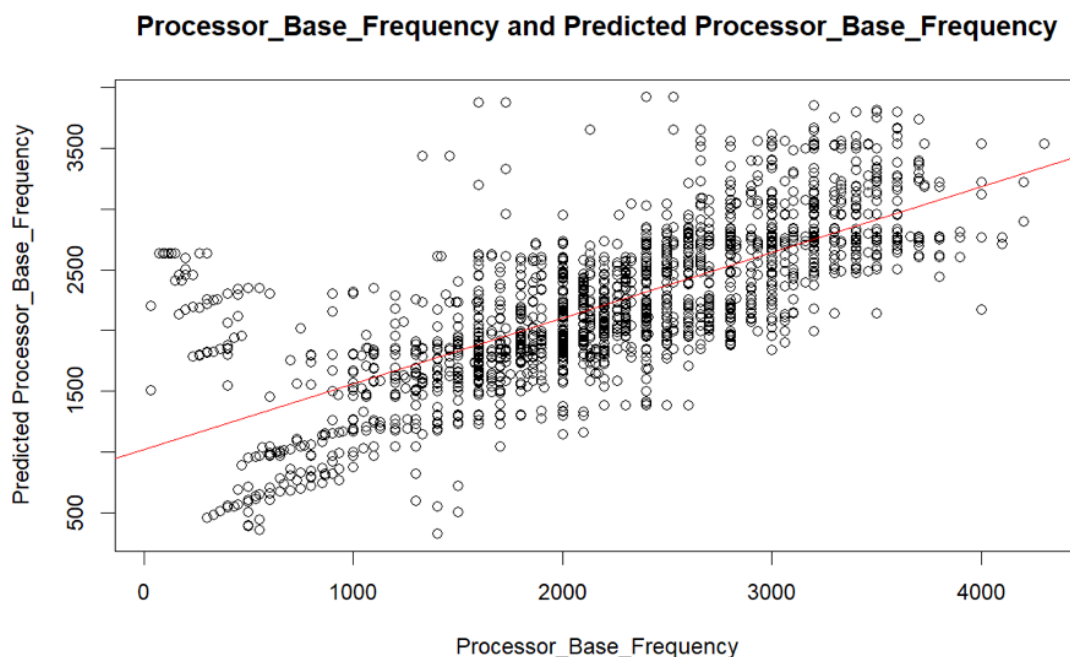
- Đồ thị này cho phép xác định những điểm có ảnh hưởng cao (influential observations), nếu

chúng có hiện diện trong bộ dữ liệu. Những điểm có ảnh hưởng cao này có thể là các điểm outliers, là những điểm có thể gây nhiều ảnh hưởng nhất khi phân tích dữ liệu. Nếu như ta quan sát thấy một đường thẳng màu đỏ đứt nét (Cook's Distance), và có một số điểm vượt qua đường thẳng khoảng cách này, nghĩa là các điểm đó là các điểm có ảnh hưởng cao. Nếu như ta chỉ quan sát thấy đường thẳng khoảng cách Cook ở góc của đồ thị và không có điểm nào vượt qua nó, nghĩa là không có điểm nào thực sự có ảnh hưởng cao.

- Dựa trên đồ thị ta thấy có các quan trắc thứ 666, 1249, 1251 có thể là các điểm có ảnh hưởng cao trong bộ dữ liệu.

#### 6.2.4 Dự báo

```
# Prediction
plot (df$Processor_Base_Frequency , predict ( model , df),xlab ="
Processor_Base_Frequency ",ylab =" Predicted Processor_Base_Frequency ",main ="
Processor_Base_Frequency and Predicted Processor_Base_Frequency ")
compair <-lm( predict ( model , df)~Processor_Base_Frequency , data = df)
abline ( compair ,col =" red ")
```



Hình 32: Dự báo tần số cơ bản của bộ xử lý

#### Nhận xét:

Dựa trên biểu đồ ta thấy các quan trắc phân tán xung quanh trên đường thẳng màu đỏ, chứng tỏ giá trị dự báo giá trị quan trắc ban đầu có quan hệ tuyến tính mạnh. Ta có thể kết luận mô hình hồi quy ta dự báo khá đủ tốt.

## 7 Thảo luận và mở rộng

CPU (Central Processing Unit) hay còn gọi là bộ xử lý trung tâm được xem là não bộ chính của một thiết bị có vai trò, nhiệm vụ chính là xử lý các chương trình, dữ kiện đầu vào từ phần mềm và phần cứng chạy trên máy tính. Tốc độ CPU hay còn gọi là tốc độ xung nhịp CPU được đo bằng đơn vị gigahertz hay GHz biểu thị số chu kỳ xử lý mỗi giây mà CPU có thể thực hiện được. Tốc độ xung nhịp CPU là một thước đo để đánh giá hiệu suất hoạt động của CPU đó xử lý dữ liệu nhanh tới đâu.

Trên thực tế, tốc độ xung nhịp của CPU không hoàn toàn phụ thuộc vào số Thread, số Core, công suất nhiệt, kích thước con chip, kích thước bộ nhớ và thời gian ra mắt như trong bài tập lớn đề cập. Tốc độ xung nhịp của CPU còn có thể phụ thuộc vào:

- Điện Áp và Nguồn Cung Cấp: Mức điện áp và nguồn cung cấp là yếu tố quan trọng. Một số CPU có thể tăng tần số cơ bản khi được cung cấp điện áp và nguồn tốt hơn.
- Công nghệ làm tăng tốc độ xử lý của CPU (pipeline, turbo boost, siêu phân luồng,...).
- Bộ nhớ đệm dùng để lưu các lệnh/dữ liệu thường dùng hay có khả năng sẽ được dùng trong tương lai gần, giúp giảm bớt thời gian chờ đợi của CPU.
- Yếu Tố Phần Mềm: Phần mềm quản lý CPU, chẳng hạn như BIOS hoặc firmware, có thể có các thiết lập và quyết định về cách CPU hoạt động và tần số cơ bản.

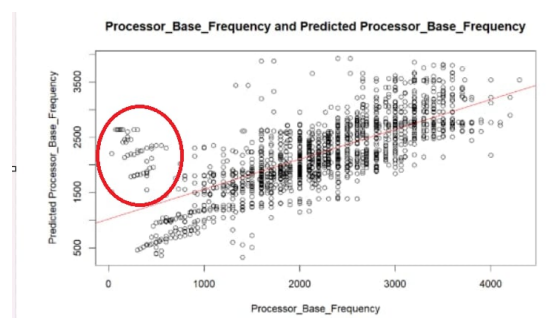
### 7.1 Thảo luận

#### Ưu điểm:

- Dễ hiểu và triển khai: Mô hình hồi quy tuyến tính là một mô hình đơn giản và dễ hiểu. Nó thường được sử dụng như là một bước đầu tiên trong quá trình mô hình hóa do tính đơn giản và khả năng giải thích cao.
- Dễ dàng điều chỉnh và mở rộng : Có thể dễ dàng điều chỉnh mô hình bằng cách thêm hoặc loại bỏ các biến độc lập. Ngoài ra, mô hình hồi quy tuyến tính có thể được mở rộng để bao gồm các biến tương tác và các biến động.

#### Hạn chế:

Linear Regression là nó rất nhạy cảm với nhiễu (sensitive to noise). Trong ví dụ về mối quan hệ giữa hiệu năng của CPU với các biến độc lập còn lại nếu có những điểm bị nhiễu thì kết quả sai khác đi rất nhiều:



## 7.2 Mở rộng

### 7.2.1 Mô hình hồi quy Ridge và Lasso

#### a) Hiện tượng quá khớp (Overfitting)

Quá khớp (Overfitting) xảy ra khi mô hình thống kê khớp chuẩn xác với bộ dữ liệu huấn luyện. Điều này khiến cho giải thuật không thể biểu diễn chính xác trên dữ liệu mới. Sự tổng quát hóa của mô hình đối với dữ liệu mới giúp chúng ta sử dụng được giải thuật học máy (machine learning algorithms) để dự đoán và phân loại dữ liệu.

Khi một giải thuật học máy được tạo nên, một bộ dữ liệu mẫu (training data) sẽ được sử dụng để huấn luyện mô hình. Tuy nhiên khi một mô hình được huấn luyện quá lâu với bộ dữ liệu mẫu hoặc khi mô hình quá phức tạp, mô hình bắt đầu thích nghi với dữ liệu nhiễu, những biến không ảnh hưởng đến kết quả của mô hình hay phân tích dự đoán hoặc phân loại biến. Điều này dẫn tới việc mô hình không đủ tổng quát đối với những dữ liệu mới thì mô hình sẽ không thể thực hiện được các tác vụ phân loại hay dự đoán chính xác.

#### b) Cách để giải quyết hiện tượng quá khớp (Overfitting)

Trong thống kê và máy học, "regularization" là một kỹ thuật được sử dụng để kiểm soát và giảm thiểu quá mức phức tạp của mô hình, ngăn chặn hiện tượng quá khớp overfitting. Mục tiêu của regularization là tối ưu hóa hiệu suất dự đoán của mô hình trên dữ liệu mới bằng cách kiểm soát các tham số của mô hình.

Có hai kỹ thuật regularization chính: L1 regularization và L2 regularization.

#### c) Hồi quy Ridge - L2 regularization

Hồi quy Ridge là một sửa đổi của hồi quy bình phương tối thiểu để làm cho nó phù hợp hơn cho việc lựa chọn biến. Trong hồi quy Ridge, chúng ta không chỉ cố gắng giảm thiểu tổng bình phương của phần dư mà còn một thành phần khác bằng tổng bình phương của các tham số hồi quy nhân với một tham số điều chỉnh. Nói cách khác, trong hồi quy Ridge, chúng ta cố gắng giảm thiểu lượng dưới đây:

$$\sum_{i=1}^n (y_i - \beta_0 - \beta_i X_i)^2 + \lambda \sum_{i=1}^n (\beta_i)^2$$

Trong đó:  $\lambda \sum_{i=1}^n (\beta_i)^2$  là thành phần điều chuẩn (regularization term)

Trong phương trình trên, giá trị  $\lambda \geq 0$ .  $\sum_{i=1}^n (y_i - \beta_0 - \beta_i X_i)^2$  chính là tổng bình phương phần dư và  $\lambda \sum_{i=1}^n (\beta_i)^2$  là thành phần điều chuẩn

- Trường hợp  $\lambda = 0$ , thành phần điều chuẩn bị tiêu giảm và chúng ta quay trở về bài toán hồi quy tuyến tính.
- Trường hợp  $\lambda$  nhỏ thì vai trò của thành phần điều chuẩn trở nên ít quan trọng. Mức độ kiểm soát quá khớp của mô hình sẽ trở nên kém hơn.
- Trường hợp  $\lambda$  lớn chúng ta muốn gia tăng mức độ kiểm soát lên độ lớn của các hệ số ước lượng và qua đó giảm bớt hiện tượng quá khớp

Khi tăng dần hệ số  $\lambda$  thì hồi quy Ridge sẽ có xu hướng thu hẹp hệ số ước lượng từ mô hình.



#### d) Hồi quy Lasso - L1 regularization

Trong hồi quy Lasso, thay vì sử dụng thành phần điều chuẩn là chuẩn bậc hai thì chúng ta sử dụng chuẩn bậc 1.

$$\sum_{i=1}^n (y_i - \beta_0 - \beta_i X_i)^2 + \lambda \sum_{i=1}^n |\beta_i|$$

Khi tiến hành hồi quy mô hình Lasso trên một bộ dữ liệu mà có các biến đầu vào đa cộng tuyến (multicollinear) thì mô hình hồi quy Lasso sẽ có xu hướng lựa chọn ra một biến trong nhóm các biến đa cộng tuyến và bỏ qua những biến còn lại. Trong khi ở mô hình hồi quy tuyến tính thông thường và hồi quy Ridge thì có xu hướng sử dụng tất cả các biến đầu vào.

## 8 Tài liệu tham khảo

- BigDataUni. Phân tích phương sai Anova (Analysis of Variance) (P.2). Truy cập từ: <https://bigdatauni.com/tin-tuc/phan-tich-phuong-sai-anova-analysis-of-variance-p-2.html>
- Thanh Nguyen. Hướng dẫn BTL tt. Truy cập từ: <https://www.youtube.com/watch?v=EBSbrV4dRq4&list=PLYUTMcHNDpCcJht6QNXvrw750BjOjFyut&index=7>
- IBM. Learn Linear Regression và Overfitting form Machine Learning. Truy cập từ: <https://machinelearningcoban.com/2016/12/28/linearregression/>
- IBM. Learn Hồi quy Ridge. Truy cập từ: <https://phamdinhkhanh.github.io/deepai-book/chml/RidgedRegression.html>
- R Documentation, Multi-factor ANOVA . Truy cập từ: <https://search.r-project.org/CRAN/refmans/bruceR/html/MANOVA.html>

## 9 Nguồn dữ liệu và nguồn code

Nguồn code có thể truy cập ở đây [lopl13\\_nhom7.R](#)

Nguồn dữ liệu có thể truy cập ở đây [lopl13\\_nhom7.xlsx](#)