

VIET NAM NATIONAL UNIVERSITY HO CHI MINH CITY  
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY  
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



Course name h

---

Report type h

Report title h

---

Advisor: Advisor h

HO CHI MINH CITY, APRIL 2023



## Contents



# 1 Abstract



## 2 Background



## 3 Data description

### 3.1 Importing data

For the first part of our project, we need to select a suitable dataset for us to analyze, as we are computer science students, we have decided to select a CPU data set, the data of which can be found [here](#):

The data give us information about 2283 CPU and 45 of their feature which include:

- Product\_Collection: tell us which type of series the core belongs to.
- Vertical\_Segment: show what kind of system the CPU was designed for (embedded, mobile, desktop, or sever).
- Processor\_Number : process ID.
- Status: show the status of the CPU (announce, launched, end of life, end of support).
- Launch\_Date: The date the product was first introduced.
- Lithography: refers to the semiconductor technology used to manufacture an integrated circuit, and is reported in nanometers (nm), indicative of the size of features built on the semiconductor.
- Recommended\_Customer\_Price: recommended customer price.
- nb\_of\_Cores: total number of cores in a processor.
- nb\_of\_Threads: total number of thread in a processor.
- Processor\_Base\_Frequency: Describes the rate at which the processor's transistors open and close.
- Max\_Turbo\_Frequency: The maximum single core frequency at which the processor is capable of operating using Intel® Turbo Boost Technology.
- Cache: CPU Cache is an area of fast memory located on the processor.
- Bus\_Speed: refers to how much data can move across the bus simultaneously.
- TDP(thermal design power): Represents the average power, in watts, the processor dissipates when operating at Base Frequency with all cores.
- Embedded\_Options\_Available: is it allow to be embedded system
- Conflict\_Free: Defined by the U.S. Securities and Exchange Commission rules to mean products that do not contain conflict minerals (tin, tantalum, tungsten).
- Max\_Memory\_Size: The maximum memory capacity supported by the processor.



- Memory\_Types: Single Channel, Dual Channel, Triple Channel, and Flex Mode. The maximum memory capacity supported by the processor.
- Max\_nb\_of\_Memory\_Channels: The number of memory channels refers to the bandwidth operation for real world application.
- Max\_Memory\_Bandwidth: The maximum rate at which data can be read from or stored into a semiconductor memory by the processor (in GB/s).
- ECC\_Memory\_Supported: ECC memory is a type of system memory that can detect and correct common kinds of internal data corruption.
- Processor\_Graphics: integrated graphics processing unit (GPU) that is built into some of Intel's processors.
- Graphics\_Base\_Frequency: The rated/guaranteed graphics render clock frequency in MHz.
- Graphics\_Max\_Dynamic\_Frequency: The maximum opportunistic graphics render clock frequency (in MHz) that can be supported using Intel HD Graphics with Frequency feature.
- Graphics\_Video\_Max\_Memory: The maximum amount of memory accessible to processor graphics. Processor graphics operates on the same physical memory as the CPU (subject to OS, driver, and other system limitations).
- Graphics\_Output: Graphics Output defines the interfaces available to communicate with display devices.
- Support\_4k: indicates the product's support of 4K
- Max\_Resolution\_HDMI: the maximum resolution supported by the processor via the HDMI interface (24bits per pixel & 60Hz). System or device display resolution is dependent on multiple system design factors; actual resolution may be lower on your system.
- Max\_Resolution\_DP: The maximum resolution supported by the processor via the DP interface (24bits per pixel & 60Hz). System or device display resolution is dependent on multiple system design factors.
- Max\_Resolution\_eDP\_Integrated\_Flat\_Panel
- DirectX\_Support: Indicates support for a specific version of DirectX, a Microsoft collection of APIs for handling multimedia compute tasks.
- OpenGL\_Support: Indicates support for OpenGL, a cross-language, multi-platform API for rendering 2D and 3D vector graphics.
- PCI\_Express\_Revision: The PCIe version supported by the processor.
- PCI\_Express\_Configurations\_: The available PCIe lane configurations that can be used to link the PCH PCIe lanes to PCIe devices.



- `T` : The maximum temperature allowed on the chip.
- `Max_nb_of_PCI_Express_Lanes`: maximum number of PCI Express Lanes that are supported.
- `Intel_Hyper_Threading_Technology` : Delivers two processing threads per physical core. Highly threaded applications can get more work done in parallel, completing tasks sooner.
- `Intel_Virtualization_Technology_VTx` : Allows one hardware platform to function as multiple “virtual” platforms. It offers improved manageability by limiting downtime and maintaining productivity by isolating computing activities into separate partitions.
- `Intel_64` : Delivers 64-bit computing on server, workstation, desktop and mobile platforms when combined with supporting software. Intel 64 architecture improves performance by allowing systems to address more than 4 GB of both virtual and physical memory.
- `Instruction_Set`: Which instruction set the CPU use.
- `Instruction_Set_Extensions` : Instruction set extension
- `Idle_States`: Used to save power when the processor is idle.
- `Thermal_Monitoring_Technologies`: Protects the processor package and the system from thermal failure through several thermal management features.
- `Secure_Key`: The CPU is supported with secure key or not.
- `Execute_Disable_Bit`: Hardware-based security feature that can reduce exposure to viruses and malicious code attacks.

For coding the data, our team use a wide range of package which include

- `rio`.
- `ggplot2`.
- `zoo`

Two primary packages used in this process are:

- `rio` : for intuitive I/O code. With this package, import and export dataset is easier and safer. It could also handle multiple file formats, so that we do not have to change the command each time we change the file format.
- `zoo` : for year-quarter format. In our data, the `Launch date` is in non-standard format, and difficult to be operated on. This package helps to transform into standard year-quarter format, and provides useful operations, such as plotting and taking difference on these formats.



```
1 pacman::p_load(  
2   rio,      # for imports & exports  
3   ggplot2, # for plots  
4   zoo      # for year-quarter formats  
5 )
```

for importing in the code, we use the function `import()` from `rio` to import our data set.

```
1 # Import data  
2  
3 data <- import("./cpu-raw.csv") # rio::import  
4
```

MAY NEN NOI GI DO THEM, VI CAI NAY TAO CHUA VIET XONG!!!!

Viet tung do this.

You should describe how the features are chosen, why it is chosen, and stuff...

NK merge this preprocessing section with this section.

### 3.2 data preprocessing

After examining the code, as we only want to focus on the performance and trend of the cpu designing, as there are many unnesscessary such as `Product_Collection` and `Processor_Number` which only tell us about series of cpu they are from, some give only information about cpu hardware support, and some feature which also has a lot of `NA` values which make it hard to anylyze the data. So our team has decided to cut back some feature and use only a couple of them, that inlcude:

**Need:** `Vertical_Segment`, `Status`, `Launch_Date`, `Lithography`, `Recommended_Customer_Price`, `nb_of_Cores`, `nb_of_Threads`, `Processor_Base_Frequency`, `TDP`, `Max_Memory_Bandwidth`, `T`.

The reason for the choice of these feature is that they are all give us a lot of information about the performance and effiecency. They are also often the most important factors in measuring the performance of each cpu.

```
1 #Data feature selection  
2 data <- data[, c("Vertical_Segment", "Status", "Launch_Date",  
3                 "Lithography", "Recommended_Customer_Price",  
4                 "nb_of_Cores", "nb_of_Threads", "Processor_Base_Frequency",  
5                 "TDP", "Max_Memory_Bandwidth", "T")]  
6
```

After we have select the feature, as the feature name are long and it has a lot of `_` in it which will make it difficult for coding, we decided to rename to shorter name.





```
1 # Rename labels
2 names(data) <- c("market", "status", "ldate", "litho",
3                 "rprice", "ncore", "nthread", "bfreq", "tdp",
4                 "memband", "temp") # Rename columns, easier to deal with
5
```

### 3.3 Data cleaning

After choosing the appropriate features, we now have the subset of the original raw dataset. However, since the values vary in types (such as string, non-standard year-quarter format and numeric-string), we might want transform them into reproducible types, so that the analysis later on is easier, homogeneous and accurate.

Note that this cleaning process **does not** remove the **NA** values, unless necessary. The reason is that, in one instance, there might be important labels that should not be eliminated, under a specific scope of study, so we do not treat instances with **NA** as an invalid datum. In later sections, when we focus on a specific pattern of the data, only by then that the data will have a specialized **NA** cleaning, and we do not, by chance, loose any important instance.

This process took `/rcode/cpu-short.csv` from the importing procedure above as an input, and produce `/rcode/cpu-clean.csv` as an output.

**Changing features' name**

**Start cleaning**



## 4 Data clarification



## 5 Data analysis

### 5.1 Analysis of Variance (ANOVA)

### 5.2 Regression



## 6 Conclusion



## 7 Appendix