VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE & ENGINEERING

PROBABILITY AND STATISTICS

THE EVOLUTION OF COMPUTER PROCESSORS: A STATISTIC OF COMMON PROPERTIES

Supervisor: Nguyen Thi Mong Ngoc, PhD
Students: Chau Dang Minh - 2013748

Ho Chi Minh City, April 2024

EVALUATION

N.O.	Student	ID	Works	Completed
1	Chau Dang Minh	2212287	Dataset overview Preprocessing	100%
2	Ha Khoi Nguyen	2212287	Descriptive statistics	100%
3	Nguyen Thi Mai Anh	2210103	Theories Slides and Presentation	
4			Inferential statistics	
5			Inferential statistics	

Table of Contents

Chapter I

Introduction

Phenomena that are meaningful to humans appear not to be stochastic. In the same sense, datasets produced by humans, or nature in time circulations have insights to be analyzed, which is accounted by Statistics. Thanks to Dr. Nguyen Thi Mong Ngoc's supervision in Probability and Statistics course, we have a chance to study basic statistics within an assignment with a tiny dataset. We organized our report in the following structure

1. Overview of the dataset. In this chapter, we carefully describe in details as much as possible the dataset, specifically the properties of each instance. We also notice which features to be used for later statistical tasks.
2. Preprocessing. We process data cleaning and some computations.
3. Descriptive statistics. We calculate some qualitative features of the dataset.
4. Inferential statistics. Our problems are explicitly stated and solved.

Chapter II

Overview of the Dataset

As Computer Science students, we are assigned to analyze a dataset about computer processors, namely CPUs and GPUs. Our dataset is credited to Intel, Game-Debate, and the companies involved in producing the part. Information of CPUs and GPUs are collected separately into two files, namely `Intel_CPUs.csv` and `All_GPUs.csv`. Let us get familiar to some technical features, given in the metadata or acquired over the internet.

Table 2.1: Some technical features of CPUs and GPUs

N.O.	Feature name	Relevant to	Details
1	Lithography	CPU, GPU	The semiconductor technology used to manufacture an integrated circuit, and is reported in nanometer
2	Number of Cores	CPU	A hardware term that describes the number of independent central processing units
3	Number of Threads	CPU	A Thread, or thread of execution, is a software term for the basic ordered sequence of instructions that can be passed
4	Base Frequency	CPU	Describes the rate at which the processor's transistors open and close.
5	Cache	CPU	An area of fast memory located on the processor.
6	Thermal Design Power	CPU	Represents the average power, in watts, the processor dissipates when operating at Base Frequency with all cores active under an Intel-defined, high-complexity workload.
7	Embedded Availability	CPU	In essence, an embedded processor is a CPU chip used in a system which is not a general-purpose workstation, laptop or desktop computer.
8	Embedded Availability	CPU	In essence, an embedded processor is a CPU chip used in a system which is not a general-purpose workstation, laptop or desktop computer.
9	Memory Types	CPU	Single Channel, Dual Channel, Triple Channel, and Flex Mode.
10	Instruction Set	CPU	

Continued on next page

Table 2.1: Some technical features of CPUs and GPUs (Continued)

11	Maximal Temperature	CPU	
12	Architecture	GPU	
13	Dedicated and Integrated	GPU	Whether the GPU is solely used or shares memory with a CPU
14	(Front-side) Bus Speed	CPU, GPU	The speed at which data is transferred between the processors and other components such as the memory, chipset, and peripherals.
15	No-Execute Bit	CPU	Hardware-based security feature that can reduce exposure to viruses and malicious-code attacks.
16	Thermal Monitoring Technologies	CPU	Protects the processor package and the system from thermal failure through several thermal management features.

Chapter III

Preprocessing

1 Data Cleaning

With `RStudio`, the working directory is automatically determined. Otherwise, it can be indicated by `here` library.

Listing 3.1: Required libraries and working directory setup

```
# Libraries and options
library(dplyr)
library(here)
library(knitr)
library(kableExtra)

# Self-defined functions
source("utils.R")

# Working directory
setwd(here())
```

Now our working directory have been explicated, we can use relative paths to read the data. With `RMarkdown`, we can prettify the rendering.

Listing 3.2: RStudio data object initialization

```
  # Read the CSV file into a data frame
  cpu_data <- read.csv("dataset/Intel_CPUs.csv")
  gpu_data <- read.csv("dataset/All_GPUs.csv")

  # Inspect the CPU data
  kable(head(cpu_data), format = "html") %>%
    kable_styling()
```

Invalid cells may contain `NA`, an empty string, or other values showing us that this cell's data was not collecting correctly. At the very first step, we want to selected only columns whose the percentage of valid cells exceeds our predefined value. Then we filter out all instances with invalid features. Note that careful column selection possibly remains more instances for later tasks. For CPU data, we concerning about the price, so let us choose a quota for which the column of prices is maintained.

Listing 3.3: Cleaning functions

```
# Check if a cell has a valid value
is_valid <- function(value) {
  return(!is.na(value)
         & !is.null(value)
         & !value == ""
         & !value == "N/A"
```

Product_Collection	Vertical_Segment	Processor_Number	Status	Launch_Date	Lithography	Recommended_Customer_Price	nb_of_Cores	nb_of_Threads	Pr
7th Generation Intel® Core™ i7 Processors	Mobile	i7-7Y75	Launched	Q3'16	14 nm	$393.00	2	4	1.3
8th Generation Intel® Core™ i5 Processors	Mobile	i5-8250U	Launched	Q3'17	14 nm	$297.00	4	8	1.6
8th Generation Intel® Core™ i7 Processors	Mobile	i7-8550U	Launched	Q3'17	14 nm	$409.00	4	8	1.8
Intel® Core™ X-series Processors	Desktop	i7-3820	End of Life	Q1'12	32 nm	$305.00	4	8	3.6
7th Generation Intel® Core™ i5 Processors	Mobile	i5-7Y57	Launched	Q1'17	14 nm	$281.00	2	4	1.2
Intel® Celeron® Processor 3000 Series	Mobile	3205U	Launched	Q1'15	14 nm	$107.00	2	2	1.5

Figure III.1: *First instances of CPUs data*

```
          & !trimws(value) == "-"
          & !value == "missing"
          & !value == "unknown")
    # Add your criteria
  }


  # Select columns with enough valid cells
  filtered_data <- function(data, valid_percentage=0.5) {
  selected_columns <- character(0)

  for (col in colnames(data)) {
    valid_count <- sum(is_valid(data[[col]]))
    total_instances <- length(data[[col]])

    if ((valid_count / total_instances) >= fill) {
      selected_columns <- c(selected_columns, col)
    }
  }

  return(data[selected_columns])
}
```

Listing 3.4: Cleaned CPU data

```
filtered_cpu_data <- filtered_data(cpu_data, valid_percentage=0.5)

processed_cpu_data <-
  filtered_cpu_data[
    apply(filtered_cpu_data, 1, function(row) all(sapply(row, is_valid))), ]

selected_cpu_data <- processed_cpu_data[] # Adjust selected columns for your later needs
selected_cpu_data <- unique(selected_cpu_data)
kable(head(selected_cpu_data), format = "html") %>%
  kable_styling()
```

Listing 3.5: Cleaned GPU data

```
filtered_gpu_data <- filtered_data(gpu_data, valid_percentage=0.5)

```

```r
processed_gpu_data <-
  filtered_gpu_data[
    apply(filtered_gpu_data, 1, function(row) all(sapply(row, is_valid))), ]

selected_gpu_data <- processed_gpu_data[] # Adjust selected columns for your later needs
selected_gpu_data <- unique(selected_gpu_data)
kable(head(selected_gpu_data), format = "html") %>%
  kable_styling()
```

	Product_Collection	Vertical_Segment	Processor_Number	Status	Launch_Date	Lithography	Recommended_Customer_Price	nb_of_Cores	nb_of_Threads
1	7th Generation Intel® Core™ i7 Processors	Mobile	i7-7Y75	Launched	Q3'16	14 nm	$393.00	2	4
2	8th Generation Intel® Core™ i5 Processors	Mobile	i5-8250U	Launched	Q3'17	14 nm	$297.00	4	8
3	8th Generation Intel® Core™ i7 Processors	Mobile	i7-8550U	Launched	Q3'17	14 nm	$409.00	4	8
5	7th Generation Intel® Core™ i5 Processors	Mobile	i5-7Y57	Launched	Q1'17	14 nm	$281.00	2	4
11	Intel® Pentium® Processor 2000 Series	Mobile	2020M	Launched	Q3'12	22 nm	$134.00	2	2
14	Intel® Pentium® Processor 4000 Series	Mobile	4405U	Launched	Q3'15	14 nm	$161.00	2	4

Figure III.2: *First instances of selected CPUs data*

	Architecture	Best_Resolution	Core_Speed	DVI_Connection	Dedicated	Direct_X	HDMI_Connection	Integrated	L2_Cache	Manufacturer	Max_Power
2	R600 XT	1366 x 768	-	2	Yes	DX 10	0	No	0KB	AMD	215 Watts
3	R600 PRO	1366 x 768	-	2	Yes	DX 10	0	No	0KB	AMD	200 Watts
5	RV630	1024 x 768	-	2	Yes	DX 10	0	No	0KB	AMD	45 Watts
6	RV630	1024 x 768	-	2	Yes	DX 10	0	No	0KB	AMD	50 Watts
7	R700 RV790 XT	1920 x 1080	870 MHz	1	Yes	DX 10.1	1	No	0KB	AMD	190 Watts
8	R600 GT	1024 x 768	-	2	Yes	DX 10	0	No	0KB	AMD	150 Watts

Figure III.3: *First instances of selected GPUs data*

It's worth finding the key of the data to know which feature uniquely determine an instance. We

prioritize categorical features, specifically pure names.

Listing 3.6: Keys of the data

```
nrow(selected_cpu_data) - nrow(selected_cpu_data[, c("Product_Collection",
                                                     "Processor_Number")])
nrow(selected_gpu_data) - nrow(selected_gpu_data[, c("Name")])
# Outputs: 0
```

2 Data Pre-computation

Some features in our data have values that need to be reformatted for easily later sorting and analyses. Therefore, we need to gain a good understand on the features.

Listing 3.7: A processing for selected features

```
cpu_columns <- colnames(cpu_data)
gpu_columns <- colnames(gpu_data)
intersect(cpu_columns, gpu_columns)
# Output: character(0)
```

Since the data files have no common features, we took a look at them independently and decided to pre-compute some features.

1. Use the common column name `Release_Date` for both data. Extract `Release_Year` and `Release_Quarter` for each instance.
2. If `Recommended_Customer_Price` is a range, compute the average.

Listing 3.8: Extract Year and Quarter from Dates

```
month_to_quarter <- function(month) {
quarter <- switch(month,
                  "Jan" = "1",
                  "Feb" = "1",
                  "Mar" = "1",
                  "Apr" = "2",
                  "May" = "2",
                  "Jun" = "2",
                  "Jul" = "3",
                  "Aug" = "3",
                  "Sep" = "3",
                  "Oct" = "4",
                  "Nov" = "4",
                  "Dec" = "4",
                  "Unknown")
 return(quarter)
}

names(selected_cpu_data)[names(selected_cpu_data) == "Launch_Date"] <- "Release_Date"
selected_cpu_data$Release_Year <- as.integer(sub("Q[1-4]'(\\d+)", "\\1",
                                        gsub("\\s+", "", selected_cpu_data$Release_Date))) +
                                           2000
selected_cpu_data$Release_Quarter <- as.integer(sub("Q([1-4])'.*", "\\1",
                                          gsub("\\s+", "", selected_cpu_data$Release_Date)))

selected_gpu_data$Release_Year <-
 as.integer(sub(".*-(\\d{4}) .*", "\\1", selected_gpu_data$Release_Date))
selected_gpu_data$Release_Quarter <-
 as.character(sub(".*-(\\w+)-\\d{4}.*", "\\1", selected_gpu_data$Release_Date))
selected_gpu_data$Release_Quarter <- sapply(selected_gpu_data$Release_Quarter, month_to_quarter)
```