Mustapha Brima - DACSS 601 Assignment Final Project

Mustapha Brima

5/6/2021

---

title: Getting More for Less (DACSS 601 Final Paper) (Mustapha Brima) (05/06/2021)

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```
#install.packages("tidyverse")
#install.packages("ggplot2")
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(ggplot2)
library(knitr)
library(png)
library(grid)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(rmarkdown)
library(tidyverse)
library(readxl)
```

Introduction:

In todays day and age, computers happen to dominate the world. From their first conception, they were complex machines that would take up the space of a whole room. Today, computers are an even greater marvel since, though they are even more complex, they only need a small amount of space to function. as they can fit in your hand. The CPU (or 'Central Processing Unit'), is a piece of hardware involved in the interpreting and calculating instructions within a computer "... while you're surfing the web, creating documents, playing games, or running software programs"[1]. Like many products in todays day ad age, technology has become a commmodity, and computer hardware is no exception to this trend. Companies often compete with each other to produce computer hardware like 'NVIDIA' 'AMD' and 'ASUS' and the graphics cards that each company produces [2]. However, each piece of computer hardware comes with its own specifications relative to the models of other companies, (or the models featured outside of its own company). In most of these cases, the better the hardware, the more expensive it is. this topic is further explained in the internet article, 'Investing In Technology: Should I Always Go For The Most Expensive One?'[3]. However, this is not always the case. In this paper I will be analyzing a dataset to identify if this mindset is true, or if there exist products with the best specifications at the lowest prices. I will be conducting this research on a dataset containing different Intel CPUs using the dataset found at https://www.kaggle.com/bwolfram/intel-cpus [4].

Data:

The data used originates from the company 'Intel' itself as according to the creator of the dataset (Brandon W.), the data is "... provided from Intel through their investor relations page" [0].

Loading the Data Set into RStudio:

```
library(readr)
CPU_prices <- read_csv("CPU prices - Desktop-Mobile.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   DorM = col_double(),
##   `Cache(M)` = col_double(),
##   Cores = col_double(),
##   Threads = col_double(),
##   `Speed(GHz)` = col_double(),
##   Price = col_double(),
##   Name = col_character()
## )
```

```
View(CPU_prices)
```

To find the dimensions of the data set.

```
dim(CPU_prices)
```

```
## [1] 225   7
```

The dataset consists of 225 entries. Each with specifications depiced through seven variables.

To identify the names of the variables used in the dataset.

```r
colnames(CPU_prices)
```

```
## [1] "DorM"       "Cache(M)"   "Cores"      "Threads"    "Speed(GHz)"
## [6] "Price"      "Name"
```

These names are also the titles for the columns used throughout the dataset.

This dataset contains 225 rows meaning that there were 225 entries for different kinds of computer CPU units (as well as additional information about them), documented in the dataset. There are seven variables in total used to describe the specifications of each CPU entry.

Information Regarding the Data

For the purpose of answering my research question, I will be focusing on the seven variables of interest in order to better get an understanding of how each CPU unit entry compares to one another. . . .

DorM: According to the creator of the dataset, DorM simply indicates whether each CPU entry was used on a Desktop or Mobile Computer (or laptop). A CPU being used on a Desktop coputer would be indicated by a 1, while a CPU being usd on a laptop computer would be indicated by a 0.

Cache(M): The CPU cache stores data that the processor might need at a later time. Measured in Megabytes (or MB), the more cache that a CPU contains, the less time a computer will have to acess he main memory and as such, "...programs may run faster" [5].

Cores: CPU cores are composed of billions of transistors used to help carry out tasks of the CPU. They are responsible for the amount of virtual tasks that the cpu can handle at once as it is generally the case that the more cores within the CPU, "...the easier it is to work on a number of tasks at once" [6].

Threads: Threads further contribute t the multitasking abilities of a CPU as, through their use, allow the processor to execute "...highest level of code" [7]. Each CPU core uses two threads, so a dual core processor would use four, a quad core processor would use 8, and so on [7].

Speed(GHz): The procssing speed of a CPU is measured in Ghz (or gigahertz), as the higher the rate of speed the more clock cyles it can perform. For example, "... CPU with a clock rate of 1.8 GHz can perform 1,800,000,000 clock cycles per second" [8].

Price:The price of each CPU entry "...reflected for October 2018" in US dollars [4].

Name: The model name of each CPU entry. Consisting of Letters, numbers, and dashes.

Statistics of the Data:

Mean(average):

```r
summarize_all(CPU_prices, mean, na.rm = TRUE)
```

```
## Warning in mean.default(Name, na.rm = TRUE): argument is not numeric or logical:
## returning NA
```

```
## # A tibble: 1 x 7
##    DorM `Cache(M)` Cores Threads `Speed(GHz)` Price  Name
##   <dbl>      <dbl> <dbl>   <dbl>        <dbl> <dbl> <dbl>
## 1 0.498       6.37  3.82    6.62         2.76  353.    NA
```

Median (or middle values):

```
summarize_all(CPU_prices, median, na.rm = TRUE)
```

```
## # A tibble: 1 x 7
##    DorM `Cache(M)` Cores Threads `Speed(GHz)` Price Name
##   <dbl>      <dbl> <dbl>   <dbl>        <dbl> <dbl> <chr>
## 1     0          4     4       4          2.8   281 i5-7600K
```

Standard Deviations:

```
summarize_all(CPU_prices, sd, na.rm = TRUE)
```

```
## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm =
## na.rm): NAs introduced by coercion
```

```
## # A tibble: 1 x 7
##    DorM `Cache(M)` Cores Threads `Speed(GHz)` Price  Name
##   <dbl>      <dbl> <dbl>   <dbl>        <dbl> <dbl> <dbl>
## 1 0.501       4.65  2.80    5.74        0.712  315.    NA
```

Variances:

```
summarize_all(CPU_prices, var, na.rm = TRUE)
```

```
## Warning in (function (x, y = NULL, na.rm = FALSE, use) : NAs introduced by
## coercion
```

```
## # A tibble: 1 x 7
##    DorM `Cache(M)` Cores Threads `Speed(GHz)`  Price  Name
##   <dbl>      <dbl> <dbl>   <dbl>        <dbl>  <dbl> <dbl>
## 1 0.251       21.6  7.86    33.0        0.507 99340.    NA
```

#Looking at distributions of Discrete variables:

Distribution of 'DorM':

```
table(CPU_prices$DorM)
```

```
##
##   0   1
## 113 112
```

Distribution of 'Cores'

```
table(CPU_prices$Cores)
```

```
##
##   2   4   6   8  10  12  14  16  18
## 109  77  21   6   4   2   2   2   2
```

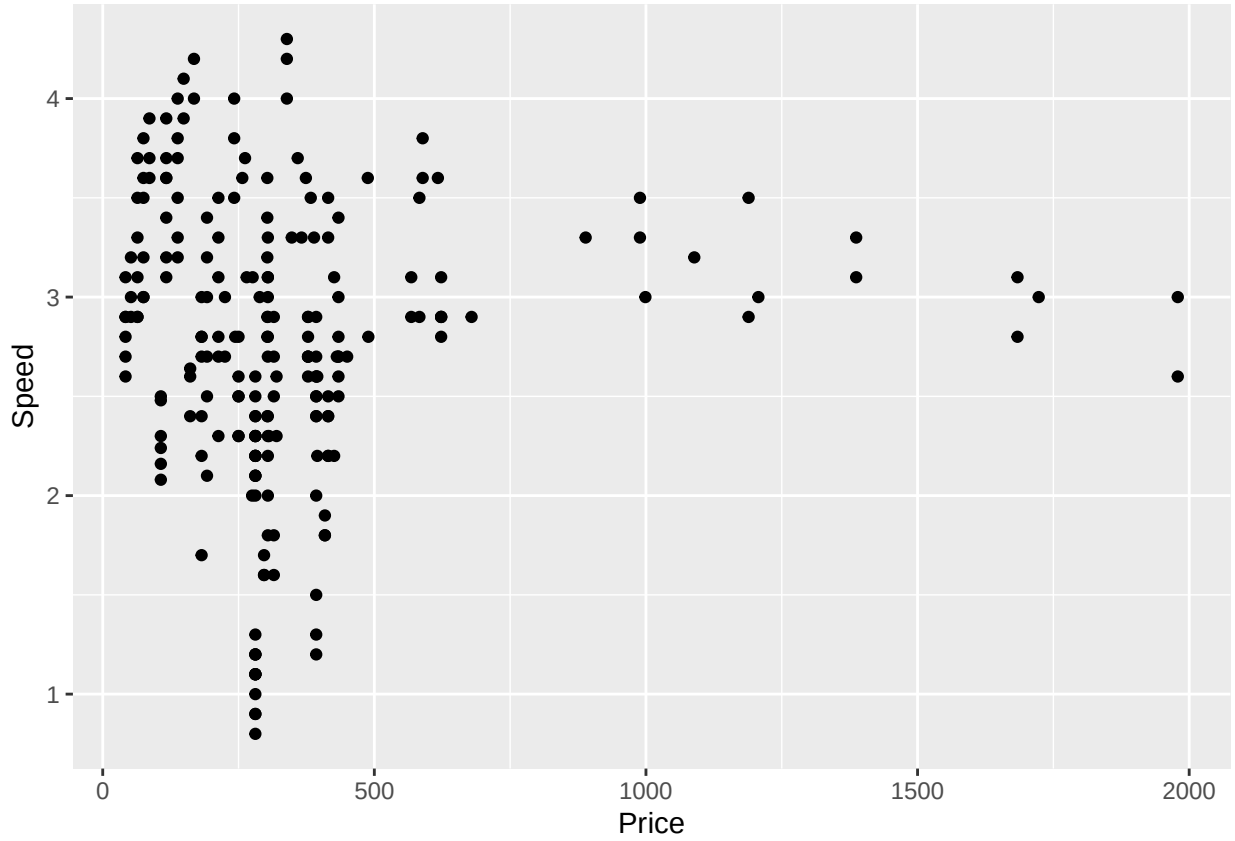
Visualization:

Prices Compared to Speed:

```
library(tidyverse)
library(readxl)
library(dplyr)
CPU_prices <- read_csv("CPU prices - Desktop-Mobile.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   DorM = col_double(),
##   `Cache(M)` = col_double(),
##   Cores = col_double(),
##   Threads = col_double(),
##   `Speed(GHz)` = col_double(),
##   Price = col_double(),
##   Name = col_character()
## )
```

```
CPU_prices %>%
  rename(Speed = "Speed(GHz)")%>%
  select(Speed, Price)%>%
  arrange(Price)%>%
  ggplot(aes(x = Price, y = Speed)) + geom_point()
```

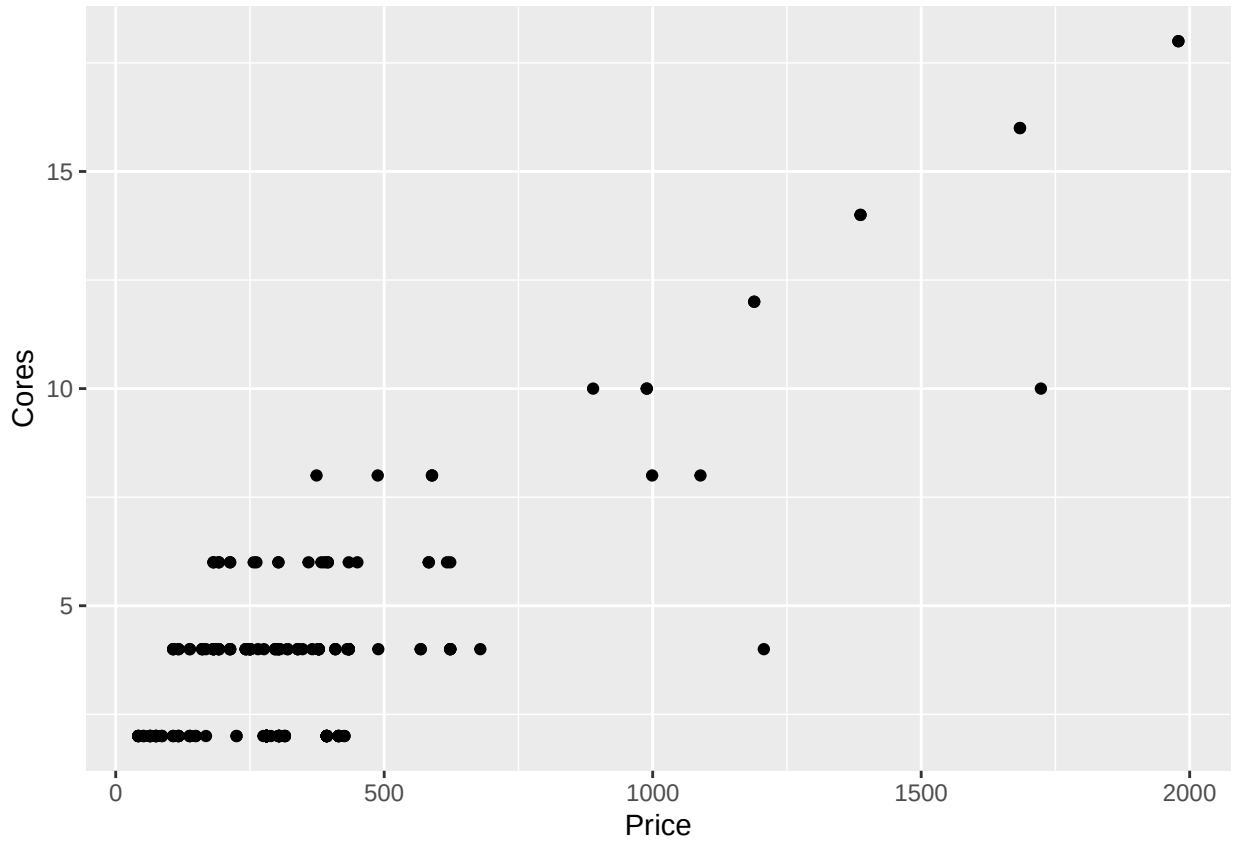


Prices Compared to Number of Cores:

```
library(tidyverse)
library(readxl)
library(dplyr)
CPU_prices <- read_csv("CPU prices - Desktop-Mobile.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   DorM = col_double(),
##   `Cache(M)` = col_double(),
##   Cores = col_double(),
##   Threads = col_double(),
##   `Speed(GHz)` = col_double(),
##   Price = col_double(),
##   Name = col_character()
## )
```

```
CPU_prices %>%
  select(Cores, Price)%>%
  arrange(Price)%>%
  ggplot(aes(x = Price, y = Cores)) + geom_point()
```



Finding the best option:

This code checks to see which entry will be the best option in that it will feature the highest speeds of a CPU (using the column Speeds(GHz)), for the lowest price (using the column 'Price').

```
library(dplyr)
CPU_prices %>%
    select(DorM, "Cache(M)", Cores, Threads, "Speed(GHz)", Price, Name) %>%
    slice(which.max("Speed(GHz)"), which.min(Price))
```

```
## Warning in which.max("Speed(GHz)"): NAs introduced by coercion
```

```
## # A tibble: 1 x 7
##    DorM `Cache(M)` Cores Threads `Speed(GHz)` Price Name
##   <dbl>      <dbl> <dbl>   <dbl>        <dbl> <dbl> <chr>
## 1     1          2     2       2          3.1    42 G4900
```

This code checks to see which entry will be the best option in that it will feature the maximum number of cores of a CPU (using the column Cores), for the lowest price (using the column 'Price').

```
library(dplyr)
CPU_prices %>%
    select(DorM, "Cache(M)", Cores, Threads, "Speed(GHz)", Price, Name) %>%
    slice(which.max(Cores), which.min(Price))
```

```
## # A tibble: 2 x 7
##    DorM `Cache(M)` Cores Threads `Speed(GHz)` Price Name
##   <dbl>      <dbl> <dbl>   <dbl>        <dbl> <dbl> <chr>
## 1     1       24.8    18      36          3    1979 i9-9980XE
## 2     1        2       2       2          3.1    42 G4900
```

This code checks to see which entry will be the best option in that it will feature the maximum number of cores of a CPU (using the column Cores), for the lowest price (using the column 'Price').

```
library(dplyr)
CPU_prices %>%
    select(DorM, "Cache(M)", Cores, Threads, "Speed(GHz)", Price, Name) %>%
    slice(which.max(Cores), which.max("Speed(GHz)"), which.min(Price))
```

```
## Warning in which.max("Speed(GHz)"): NAs introduced by coercion
```

```
## # A tibble: 2 x 7
##    DorM `Cache(M)` Cores Threads `Speed(GHz)` Price Name
##   <dbl>      <dbl> <dbl>   <dbl>        <dbl> <dbl> <chr>
## 1     1       24.8    18      36          3    1979 i9-9980XE
## 2     1        2       2       2          3.1    42 G4900
```

```
library(dplyr)
CPU_prices %>%
    select(DorM, "Cache(M)", Cores, Threads, "Speed(GHz)", Price, Name) %>%
    slice(which.max(Cores), which.max("Speed(GHz)"), which.max("Cache(M)"), which.max(Threads), which.m
```

```
## Warning in which.max("Speed(GHz)"): NAs introduced by coercion
```

```
## Warning in which.max("Cache(M)"): NAs introduced by coercion
```

```
## # A tibble: 3 x 7
##    DorM `Cache(M)` Cores Threads `Speed(GHz)` Price Name
##   <dbl>      <dbl> <dbl>   <dbl>        <dbl> <dbl> <chr>
## 1     1       24.8    18      36          3    1979 i9-9980XE
## 2     1       24.8    18      36          3    1979 i9-9980XE
## 3     1        2       2       2          3.1    42 G4900
```

Reflection:

Reflecting on this project, I would have liked to use more columns to determine the best CPU economically wise, but I was worried that using too many variables might skew the result, as having a decent processor would come down to having more than 1 core ('Cores' as well as having a considerably high clock speed (Speed(Ghz)).

Conclusion:

Considering all of these findings, the Intel G4900 CPU model seems to be the best option economically as it holds some of the best specifications in number of cores and speed. It is a dual-core processor (meaning that it has two cores in its composition), a speed of 3.1 GHz (or gigahertz), for a price of $42.00 in US currency.

I was going to show a CHI-squared test as well, but I decided not to do so as the CPU models were most likely developed in succession of one another. Though the dataset does not indicate which models were developed first or later on by Intel, this group of CPU models might be less of a random set of samples (as most Chi-Squared tests tend to be used for), but more so an archive of each of Intel's releases in CPUs throughout the years. I also didn't use a chi-squared test because I used plots to show the correlations between different specifications such as Price related to CPU speed (which didn't really have a correlation) or between Price and the number of cores on a CPU (in which a greater price correlated with a greater number of cores).

Bibliography:

[0] B. W, "Intel CPUs," Kaggle, 28-Oct-2018. [Online]. Available: https://www.kaggle.com/bwolfram/intel-cpus. [Accessed: 06-May-2021].

[1] M. Wilson, "What Is a CPU And How To Monitor Its Usage: HP® Tech Takes," What Is a CPU And How To Monitor Its Usage | HP® Tech Takes, 25-Feb-2020. [Online]. Available: https://www.hp.com/us-en/shop/tech-takes/what-is-cpu#:~:text=All%20kinds%20of%20computing%20devices,games%2C%20or%20running%20softwa
[Accessed: 07-May-2021].

[2] D. Edwards, "Top 10 graphics processing unit manufacturers: Nvidia clearly in the lead," Robotics & Automation News, 11-Aug-2017. [Online]. Available: https://roboticsandautomationnews.com/2017/08/11/top-10-graphics-processing-unit-manufacturers-nvidia-clearly-in-the-lead/13709/. [Accessed: 07-May-2021].

[3] C. R. Budiman, "Investing In Technology: Should I Always Go For The Most Expensive One?," Medium, 21-Jun-2020. [Online]. Available: https://medium.com/illumination/investing-in-technology-should-i-always-go-for-the-most-expensive-one-b1f863ba8e9c. [Accessed: 07-May-2021].

[4] B. W, "Intel CPUs," Kaggle, 28-Oct-2018. [Online]. Available: https://www.kaggle.com/bwolfram/intel-cpus. [Accessed: 06-May-2021].

[5] "Explain how cache memory can improve system performance.," MyTutor. [Online]. Available: https://www.mytutor.co.uk/answers/12808/GCSE/Computing/Explain-how-cache-memory-can-improve-system-performance/. [Accessed: 07-May-2021].

[6] D. Horowitz, “CPU Cores How Many Do I Need: HP® Tech Takes,” CPU Cores How Many Do I Need | HP® Tech Takes, 24-Aug-2020. [Online]. Available: https://www.hp.com/us-en/shop/tech-takes/cpu-cores-how-many-do-i-need. [Accessed: 08-May-2021].

[7] Randy, “What Are Threads in a Processor?,” WhatsaByte, 29-Mar-2021. [Online]. Available: https://whatsabyte.com/blog/processor-threads/. [Accessed: 07-May-2021].

[8] C. Hoffman, “Why You Can’t Use CPU Clock Speed to Compare Computer Performance,” How-To Geek, 06-Jul-2017. [Online]. Available: https://www.howtogeek.com/177790/why-you-cant-use-cpu-clock-speed-to-compare-computer-performance/#:~:text=CPU%20clock%20speed%2C%20or%20clock,1%2C800%2C000%2C000%20cloc [Accessed: 07-May-2021].