Final Project

Kevin Wang and Maximillian Fong

2022-12-23

Preliminaries

The goal of this project is to analyze and summarize data from Kaggle on CPU and GPU processors over time. First, we load the libraries needed later on and gather the required data, storing it in a variable named cpu_gpu_data.

```
library(tidyverse)
library(gridExtra)
library(ggmosaic)
library(knitr)
cpu_gpu_data <- read_csv("chip_dataset.csv")
cpu_gpu_data
```

```
# A tibble: 4,854 x 14
      ID Product    Type  Relea~1 Proce~2 TDP (~3 Die S~4 Trans~5 Freq ~6 Foundry
   <dbl> <chr>      <chr> <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>
 1     0 AMD Athl~ CPU   2007-0~      65      45      77     122    2200 Unknown
 2     1 AMD Athl~ CPU   2018-0~      14      35     192    4800    3200 Unknown
 3     2 Intel Co~ CPU   2020-0~      10      28      NA      NA    2600 Intel
 4     3 Intel Xe~ CPU   2013-0~      22      80     160    1400    1800 Intel
 5     4 AMD Phen~ CPU   2011-0~      45     125     258     758    3700 Unknown
 6     5 Intel Xe~ CPU   2013-0~      22      95     160    1400    2400 Intel
 7     6 AMD Phen~ CPU   2008-0~      65     125     285     450    2400 Unknown
 8     7 Intel Pe~ CPU   2006-0~      65     130     140     376    3000 Intel
 9     8 Intel Co~ CPU   2020-0~      10      28      NA      NA    2000 Intel
10     9 AMD Athl~ CPU   2006-0~      90      89     156     154    2200 Unknown
# ... with 4,844 more rows, 4 more variables: Vendor <chr>,
#   'FP16 GFLOPS' <dbl>, 'FP32 GFLOPS' <dbl>, 'FP64 GFLOPS' <dbl>, and
#   abbreviated variable names 1: 'Release Date', 2: 'Process Size (nm)',
#   3: 'TDP (W)', 4: 'Die Size (mm^2)', 5: 'Transistors (million)',
#   6: 'Freq (MHz)'
```

Task 1: Overall summaries of the data

Our first task is to summarize and analyze some of the data.

a) Physical and performance characteristics

Here, we look at the five physical and performance characteristics of the processors. Since we want to look at CPU and GPU's separately, we start by grouping the data:

```
cpu_gpu_data_grp <- group_by(cpu_gpu_data, Type)
```

Many of these summaries use the same plots or summary table; thus, we start by creating functions that will summarize numerically and graphically the data for a certain variable in order to reduce repetitive code. For numerical summaries, we create the function `get_numerical_summary`, and for graphical summaries, functions for each of the plots we may use. All of these functions take a data set (default `cpu_gpu_data_grp`) and a variable as inputs.

```
get_numerical_summary <- function(dataset = cpu_gpu_data_grp, variable) {
  summarise(dataset, Avg = mean(.data[[variable]],  na.rm = TRUE),
            Med = median(.data[[variable]], na.rm = TRUE),
            StD = sd(.data[[variable]], na.rm = TRUE),
            Min = min(.data[[variable]], na.rm = TRUE),
            Max = max(.data[[variable]], na.rm = TRUE),
            '25%ile' = quantile(.data[[variable]],0.25, na.rm = TRUE),
            '75%ile' = quantile(.data[[variable]],0.75, na.rm = TRUE),
            IQR = IQR(.data[[variable]], na.rm = TRUE))
}

get_boxplot_summary <- function(dataset = cpu_gpu_data_grp, variable) {
  boxplot = ggplot(dataset, aes(x = Type, y = .data[[variable]], fill = Type)) +
    stat_boxplot(geom = "errorbar", width = 0.25) + geom_boxplot() + ylab(variable)

  return(boxplot)
}

get_histogram_summary <- function(dataset = cpu_gpu_data_grp, variable, num_bins = 20) {
  histogram = ggplot(dataset, aes(x = .data[[variable]], group = Type, fill = Type)) +
    geom_histogram(bins = num_bins, col = "black") + xlab(variable) + facet_wrap(~Type)

  return(histogram)
}

get_density_summary <- function(dataset = cpu_gpu_data_grp, variable) {
  density_plot = ggplot(dataset, aes(x = .data[[variable]], col = Type)) +
    geom_density(size=1.5) + xlab(variable)

  return(density_plot)
}
```

Because the data contains some missing values, we set `na.rm = TRUE` for the summary functions in `get_numerical_summary` in order to get the relevant statistics summary table, ignoring the missing values.

Process Size

We'll start by analyzing process sizes. Here is the summary table:

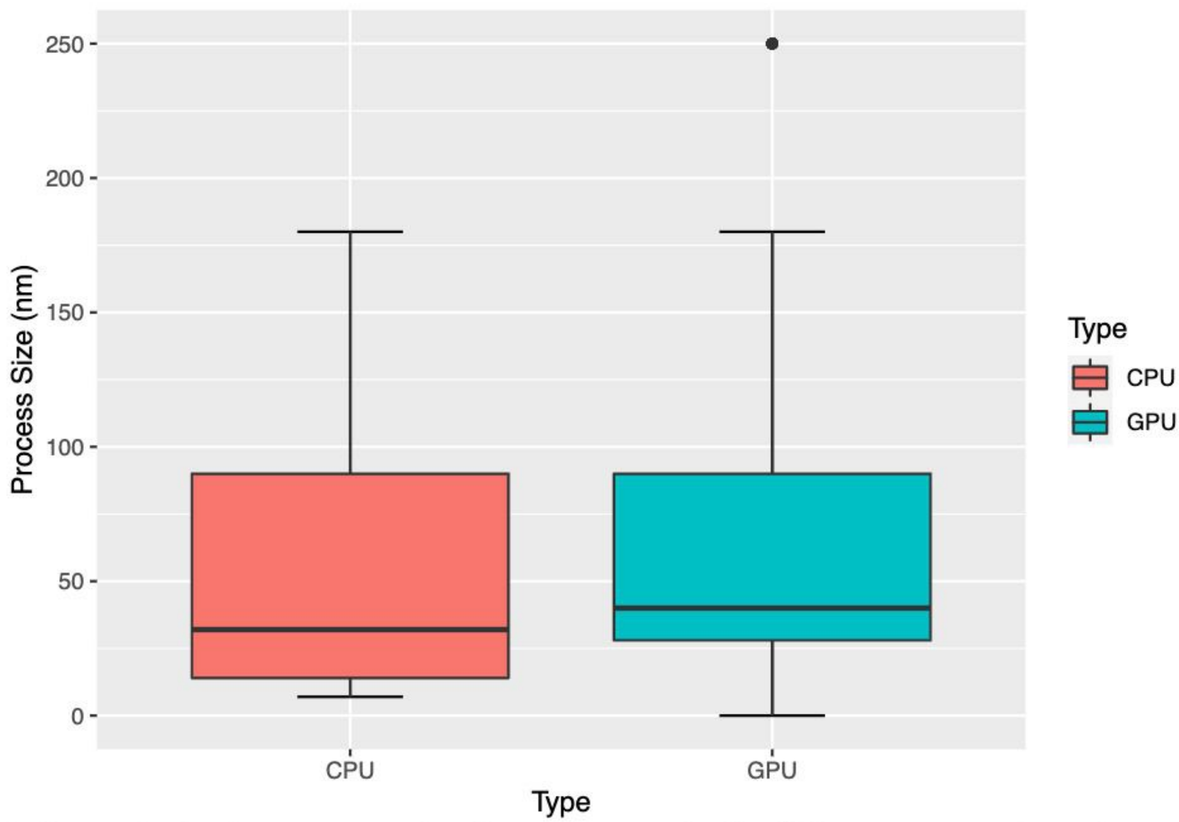
```
get_numerical_summary(variable = "Process Size (nm)")
```

```
# A tibble: 2 x 9
  Type     Avg    Med    StD    Min    Max ‘25%ile‘ ‘75%ile‘   IQR
```

```
  <chr> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>    <dbl> <dbl>
1 CPU    52.0    32  42.1     7   180       14       90    76
2 GPU    57.7    40  47.1     0   250       28       90    62
```

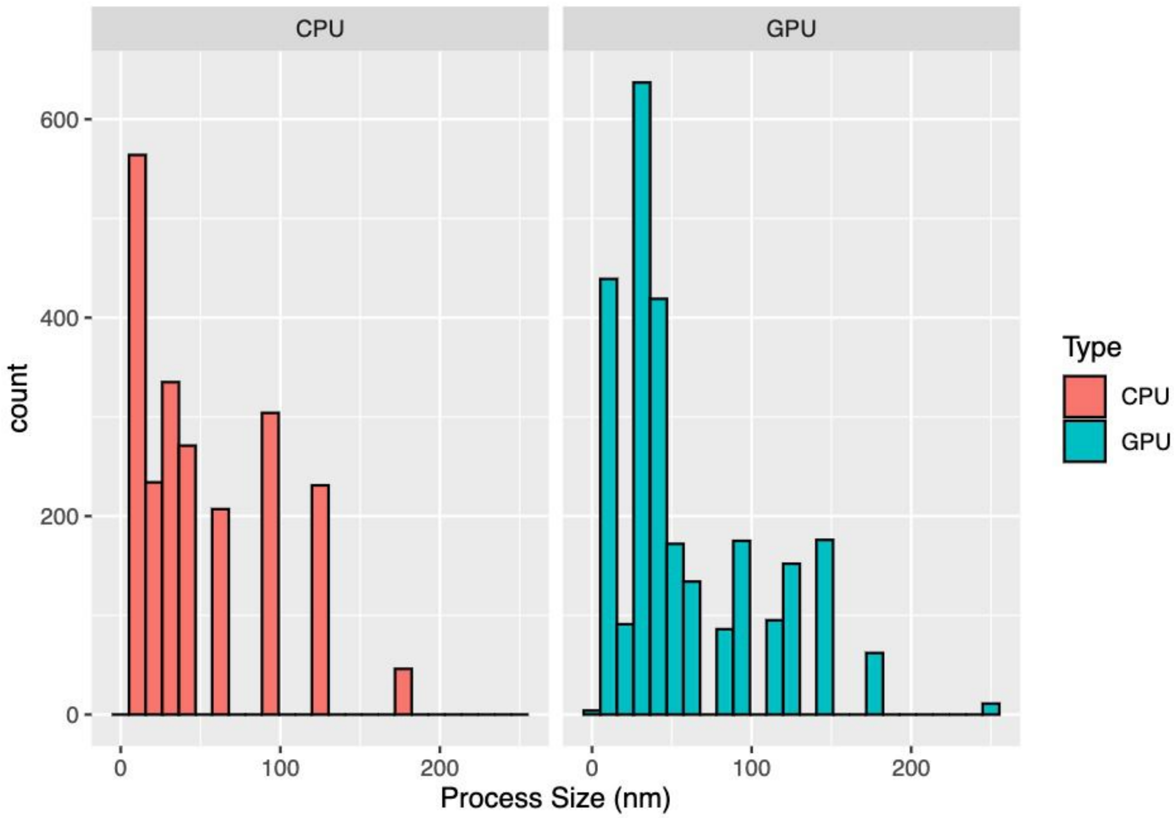
We can see that the central location (average and median) of GPU process sizes (57.70nm and 40nm respectively) is slightly larger than that of the CPUs (51.97nm and 32nm respectively). However, based on standard deviation, there appears to be slightly more variability in the data for GPUs.

```
get_boxplot_summary(variable = "Process Size (nm)" )
```



Looking at the boxplot, we can see that there exists an outlier for GPU process sizes – this outlier is the maximum value, the largest process size of all the GPUs (250 nm); on the other hand, there are no outliers for the CPUs. We can also see from this boxplot that the central portion of the CPU process size data is more spread than that of the GPUs (i.e., larger interquartile range).

```
get_histogram_summary(variable = "Process Size (nm)", num_bins = 25)
```



From these histograms, we can see that the data is somewhat positively skewed, peaking at small values of process size for both CPU and GPU. This is consistent with the fact that the means are greater than the medians.

Note that there were 9 missing values for this variable; these CPUs or GPUs that had process size missing were not part of the analyzed data.

```
sum(is.na(cpu_gpu_data_grp$`Process Size (nm)`)) #number of missing values
```

```
[1] 9
```

Thermal Design Power (TDP)

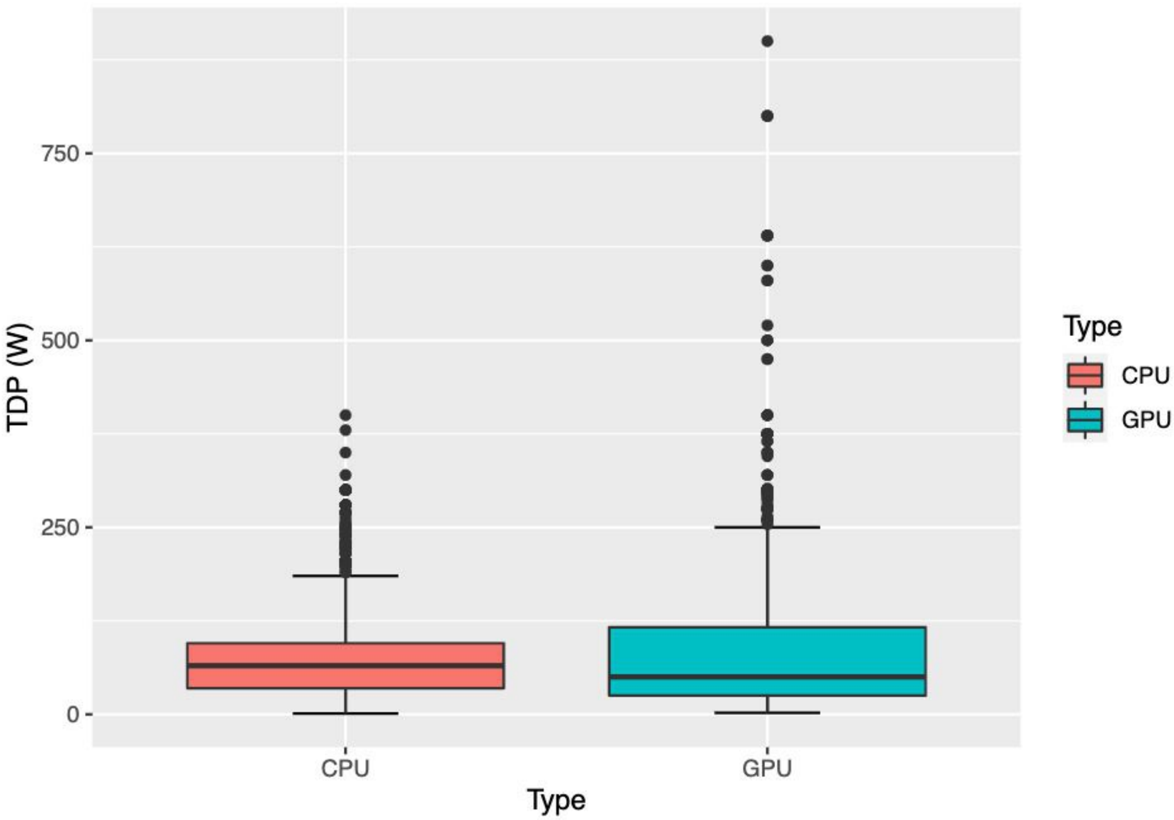
Next, we'll look at TDP. Here is the numerical summary table:

```
get_numerical_summary(variable = "TDP (W)")
```

```
# A tibble: 2 x 9
  Type    Avg   Med   StD   Min   Max `25%ile` `75%ile`   IQR
  <chr> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>    <dbl> <dbl>
1 CPU    75.4    65  54.4     1   400       35       95    60
2 GPU    87.8    50  94.8     2   900       25     116.  91.5
```

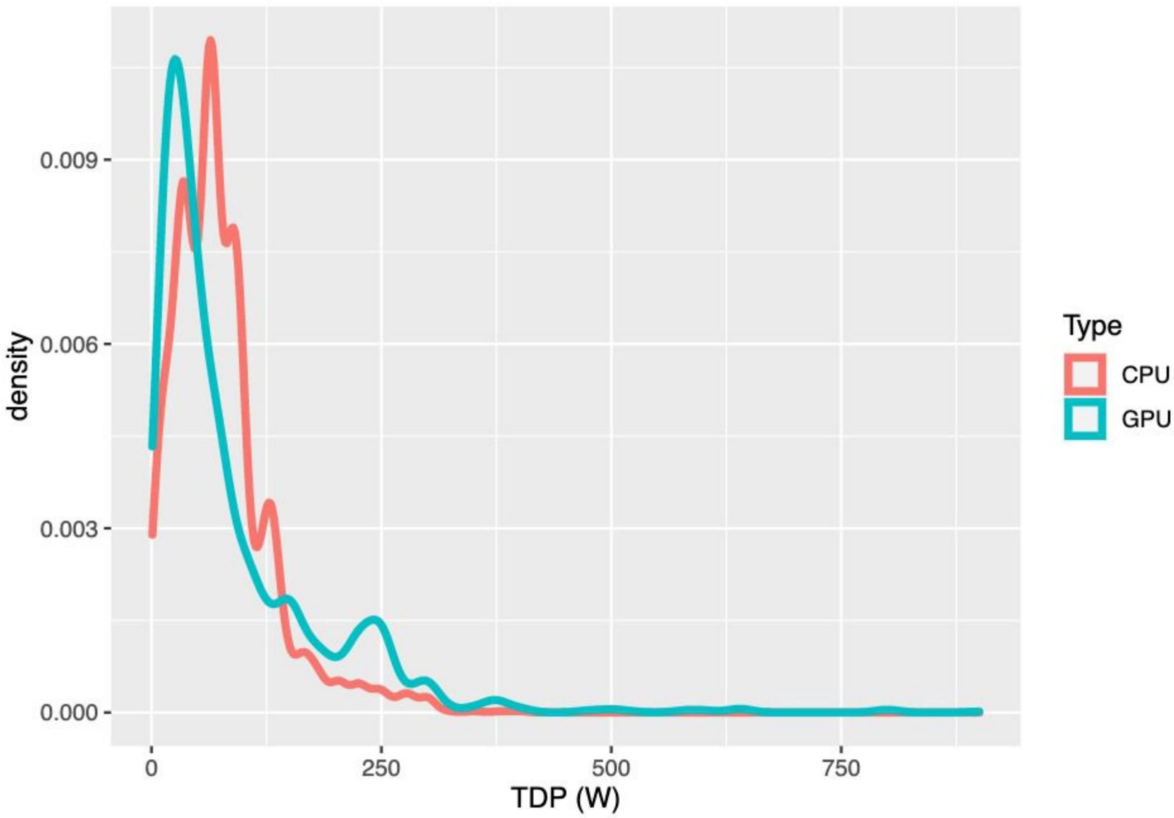
We can see that although the GPUs' TDP mean (87.77W) is higher than that of the CPUs (75.40W), its median is lower (50W for GPU, 65W for CPU). Furthermore, the GPU data has much more variability (larger standard deviation and interquartile range). The largest TDP value for GPU (900W) is also much bigger than that of CPU (400W). Based on these observations, it makes sense that the average TDP for GPUs is higher than the average TDP for CPUs, since the mean is being 'dragged' up by extreme TDP values for GPU.

```
get_boxplot_summary(variable = "TDP (W)")
```



From the histograms, we can see that there exists more extreme values (outliers) of TDP values for GPU. This is consistent with the fact that its interquartile range (and standard deviation) is larger than that of CPU TDP.

```
get_density_summary(variable = "TDP (W)")
```



By looking at the density plots, we can see that for both CPU and GPU, the distribution seems positively skewed. This is consistent with the fact that the means are greater than the medians.

Note that this time there were 626 missing values that were not considered.

```
sum(is.na(cpu_gpu_data_grp$`TDP (W)`)) #number of missing values
```

```
[1] 626
```

Die Size

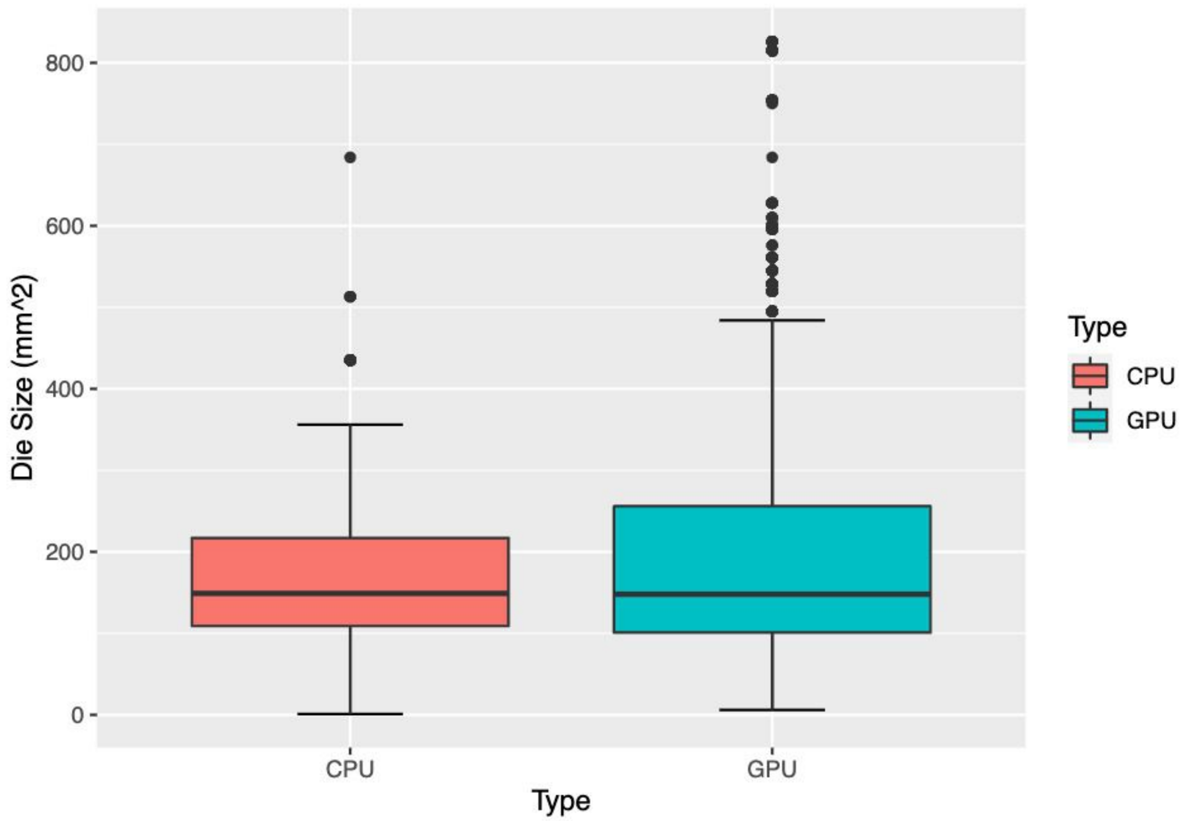
Here, we look at the die sizes.

```
get_numerical_summary(variable = "Die Size (mm^2)")
```

```
# A tibble: 2 x 9
  Type    Avg   Med   StD   Min   Max `25%ile` `75%ile`   IQR
  <chr> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>    <dbl> <dbl>
1 CPU    167.   149  79.7     1   684      109      217   108
2 GPU    203.   148 148.      6   826      101      256   155
```

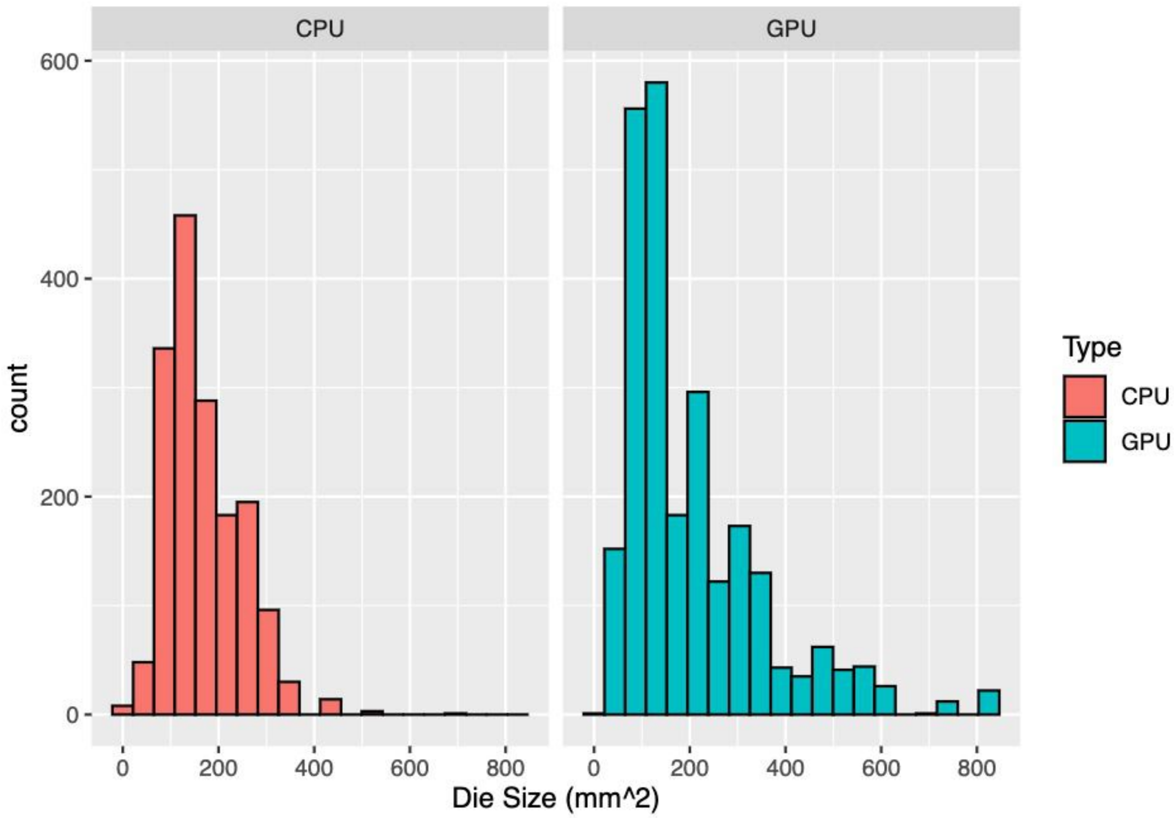
The average die size for GPU ($203.13mm^2$) is quite larger than that of CPU die size ($166.51mm^2$), although their medians are roughly the same ($149mm^2$ for GPU and $148mm^2$ for CPU). This can partly be explained by the fact that the GPU die sizes exhibit more variability, which seems to be coming from the positive end by the looks of their max value ($826mm^2$ vs. $684mm^2$) and 75% quartile ($256mm^2$vs. $217mm^2$), bringing the average up.

```
get_boxplot_summary(variable = "Die Size (mm^2)")
```



We can see that there exists more extreme values (outliers) of die size values for GPU. We can also see that GPU die size has a larger interquartile range, but the median is roughly the same as the median CPU die size (as stated above). Notice how there is also more variability in GPU die size based on the whiskers of the boxplots.

```
get_histogram_summary(variable = "Die Size (mm^2)")
```



From the histograms, we can see that the distribution for both CPU and GPU die size is positively skewed, again consistent with the fact that the means are greater than the medians.

Here, there were 715 missing values that were not used in the analysis.

```
sum(is.na(cpu_gpu_data_grp$`Die Size (mm^2)`)) #number of missing values
```

```
[1] 715
```

Transistors

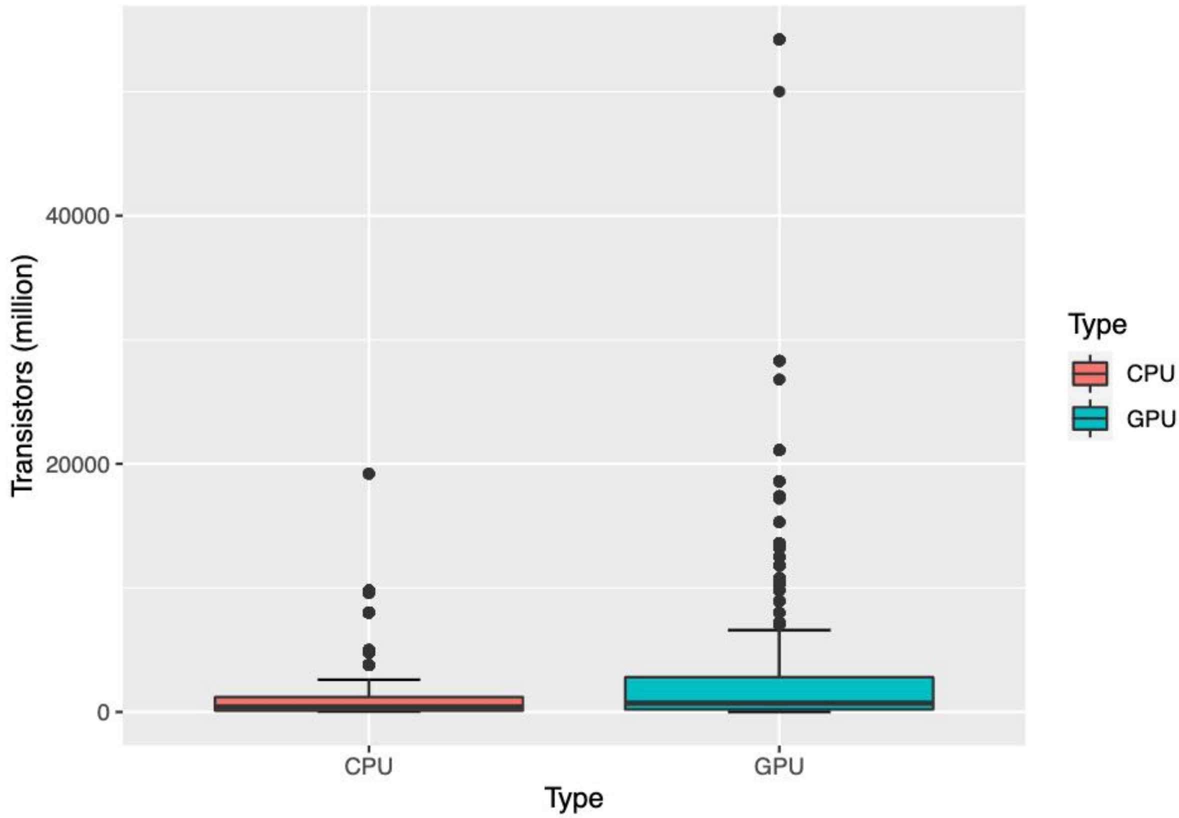
Now we look at the number of transistors in CPUs and GPUs.

```
get_numerical_summary(variable = "Transistors (million)")
```

```
# A tibble: 2 x 9
  Type    Avg   Med   StD   Min   Max `25%ile` `75%ile`   IQR
  <chr> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>    <dbl> <dbl>
1 CPU   1156.   410 2037.    37 19200      114     1200  1086
2 GPU   2455.   716 4896.     8 54200      210     2800  2590
```

We can see that both averages (1156.34 million for CPU and 2454.94 million for GPU) are much higher than their medians (410 million for CPU and 716 million for GPU). This seems to indicate positive skewness in both distributions. We can also see that the number of transistors in GPUs is much more variable, much more spread, than that of CPUs (larger standard deviation, interquartile range and larger max).

```r
get_boxplot_summary(variable = "Transistors (million)")
```



From these boxplots, we can see that there exists quite some positive extreme values (outliers) for both the number of transistors in CPUs and GPUs; however, GPUs have the most extreme values. This is consistent with the fact that both means are greater than their medians. Here, we can also see the larger interquartile range and spread of GPU transistor count.

```r
get_histogram_summary(variable = "Transistors (million)", num_bins = 25)
```