MATH208 Final Project

Barry Chen

2022-12-03

```
cpu_gpu_data<-read_csv("/Users/cccxxx/Downloads/chip_dataset.csv",show_col_types = FA
LSE)
names(cpu_gpu_data)
```

```
##  [1] "ID"                      "Product"                 "Type"
##  [4] "Release Date"            "Process Size (nm)"       "TDP (W)"
##  [7] "Die Size (mm^2)"         "Transistors (million)" "Freq (MHz)"
## [10] "Foundry"                 "Vendor"                  "FP16 GFLOPS"
## [13] "FP32 GFLOPS"             "FP64 GFLOPS"
```

TASK 1

a.

Analyze NA value

```
apply(is.na(cpu_gpu_data),2,sum)
```

```
##                    ID               Product                  Type
##                     0                     0                     0
##          Release Date     Process Size (nm)               TDP (W)
##                     0                     9                   626
##       Die Size (mm^2) Transistors (million)            Freq (MHz)
##                   715                   711                     0
##               Foundry                Vendor           FP16 GFLOPS
##                     0                     0                  4318
##           FP32 GFLOPS           FP64 GFLOPS
##                  2906                  3548
```

```
cpu_gpu_data %>% summarise_all(list(~sum(is.na(.)))) %>%
pivot_longer(cols=everything(),names_to = "Variable")
```

```
## # A tibble: 14 × 2
##    Variable               value
##    <chr>                  <int>
##  1 ID                         0
##  2 Product                    0
##  3 Type                       0
##  4 Release Date               0
##  5 Process Size (nm)          9
##  6 TDP (W)                  626
##  7 Die Size (mm^2)          715
##  8 Transistors (million)    711
##  9 Freq (MHz)                 0
## 10 Foundry                    0
## 11 Vendor                     0
## 12 FP16 GFLOPS             4318
## 13 FP32 GFLOPS             2906
## 14 FP64 GFLOPS             3548
```

There are 625 missing values for TPD measures, 715 missing values for Die size measures, 711 missing values for Transistors measures, and 0 missing value for Freq measures.

Overall numerical summary of mean and median for each measures

```
# Find the group size for TYPE
cpu_gpu_data <- cpu_gpu_data %>% mutate(TYPE = factor(Type))
cpu_gpu_data %>% group_by(TYPE) %>% summarise(count = n())
```

```
## # A tibble: 2 × 2
##   TYPE  count
##   <fct> <int>
## 1 CPU    2192
## 2 GPU    2662
```

```
# Include the NA value
cpu_gpu_data %>% group_by(TYPE) %>% select(`Process Size (nm)`,`TDP (W)`,`Die Size (m
m^2)`,`Transistors (million)`,`Freq (MHz)`) %>% summarise_all(list(Avg=mean,Med=media
n))
```

```
## # A tibble: 2 × 11
##   TYPE  Proces…¹ TDP (…² Die S…³ Trans…⁴ Freq …⁵ Proce…⁶ TDP (…⁷ Die S…⁸ Trans…⁹
##   <fct>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 CPU       52.0    75.4      NA      NA   2482.      32      65      NA      NA
## 2 GPU       NA      NA        NA      NA    663.      NA      NA      NA      NA
## # … with 1 more variable: `Freq (MHz)_Med` <dbl>, and abbreviated variable
## #   names ¹`Process Size (nm)_Avg`, ²`TDP (W)_Avg`, ³`Die Size (mm^2)_Avg`,
## #   ⁴`Transistors (million)_Avg`, ⁵`Freq (MHz)_Avg`, ⁶`Process Size (nm)_Med`,
## #   ⁷`TDP (W)_Med`, ⁸`Die Size (mm^2)_Med`, ⁹`Transistors (million)_Med`
```

```
# Exclude the NA value and pivot longer
cpu_gpu_data %>% group_by(TYPE) %>% select(`Process Size (nm)`,`TDP (W)`,`Die Size (m
m^2)`,`Transistors (million)`,`Freq (MHz)`) %>% summarise_all(list(Avg=mean,Med=media
n),na.rm = TRUE) %>% pivot_longer(cols = c("Process Size (nm)_Avg":"Freq (MHz)_Med"),
names_to = "Measure")
```

```
## # A tibble: 20 × 3
##    TYPE  Measure                    value
##    <fct> <chr>                      <dbl>
##  1 CPU   Process Size (nm)_Avg       52.0
##  2 CPU   TDP (W)_Avg                 75.4
##  3 CPU   Die Size (mm^2)_Avg        167. 
##  4 CPU   Transistors (million)_Avg 1156. 
##  5 CPU   Freq (MHz)_Avg            2482. 
##  6 CPU   Process Size (nm)_Med       32  
##  7 CPU   TDP (W)_Med                 65  
##  8 CPU   Die Size (mm^2)_Med        149  
##  9 CPU   Transistors (million)_Med  410  
## 10 CPU   Freq (MHz)_Med            2400  
## 11 GPU   Process Size (nm)_Avg       57.7
## 12 GPU   TDP (W)_Avg                 87.8
## 13 GPU   Die Size (mm^2)_Avg        203. 
## 14 GPU   Transistors (million)_Avg 2455. 
## 15 GPU   Freq (MHz)_Avg             663. 
## 16 GPU   Process Size (nm)_Med       40  
## 17 GPU   TDP (W)_Med                 50  
## 18 GPU   Die Size (mm^2)_Med        148  
## 19 GPU   Transistors (million)_Med  716  
## 20 GPU   Freq (MHz)_Med             600  
```

```
# Exclude the NA value and pivot wider
cpu_gpu_data %>% group_by(TYPE) %>% select(`Process Size (nm)`,`TDP (W)`,`Die Size (m
m^2)`,`Transistors (million)`,`Freq (MHz)`) %>% summarise_all(list(Avg=mean,Med=media
n),na.rm = TRUE) %>% pivot_longer(cols = c("Process Size (nm)_Avg":"Freq (MHz)_Med"),
names_to = "Measure") %>% pivot_wider(id_cols = "Measure",names_from = "TYPE") %>% ar
range(desc(Measure))
```

```
## # A tibble: 10 × 3
##    Measure                      CPU    GPU
##    <chr>                      <dbl>  <dbl>
##  1 Transistors (million)_Med  410    716  
##  2 Transistors (million)_Avg 1156.  2455. 
##  3 TDP (W)_Med                 65     50  
##  4 TDP (W)_Avg                 75.4   87.8
##  5 Process Size (nm)_Med       32     40  
##  6 Process Size (nm)_Avg       52.0   57.7
##  7 Freq (MHz)_Med            2400    600  
##  8 Freq (MHz)_Avg            2482.   663. 
##  9 Die Size (mm^2)_Med        149    148  
## 10 Die Size (mm^2)_Avg        167.   203. 
```

From the last table which generated by pivot wider, it’s easy to summarize:

- There are 2192 numbers of observations for CPU and 2662 numbers of observations for GPU.

- GPU has a larger value on Transistors.
- GPU has a larger average value on TDP.
- GPU has a larger value on Process Size.
- CPU has a significant larger value on Freq. GPU has a larger value on Die Size.

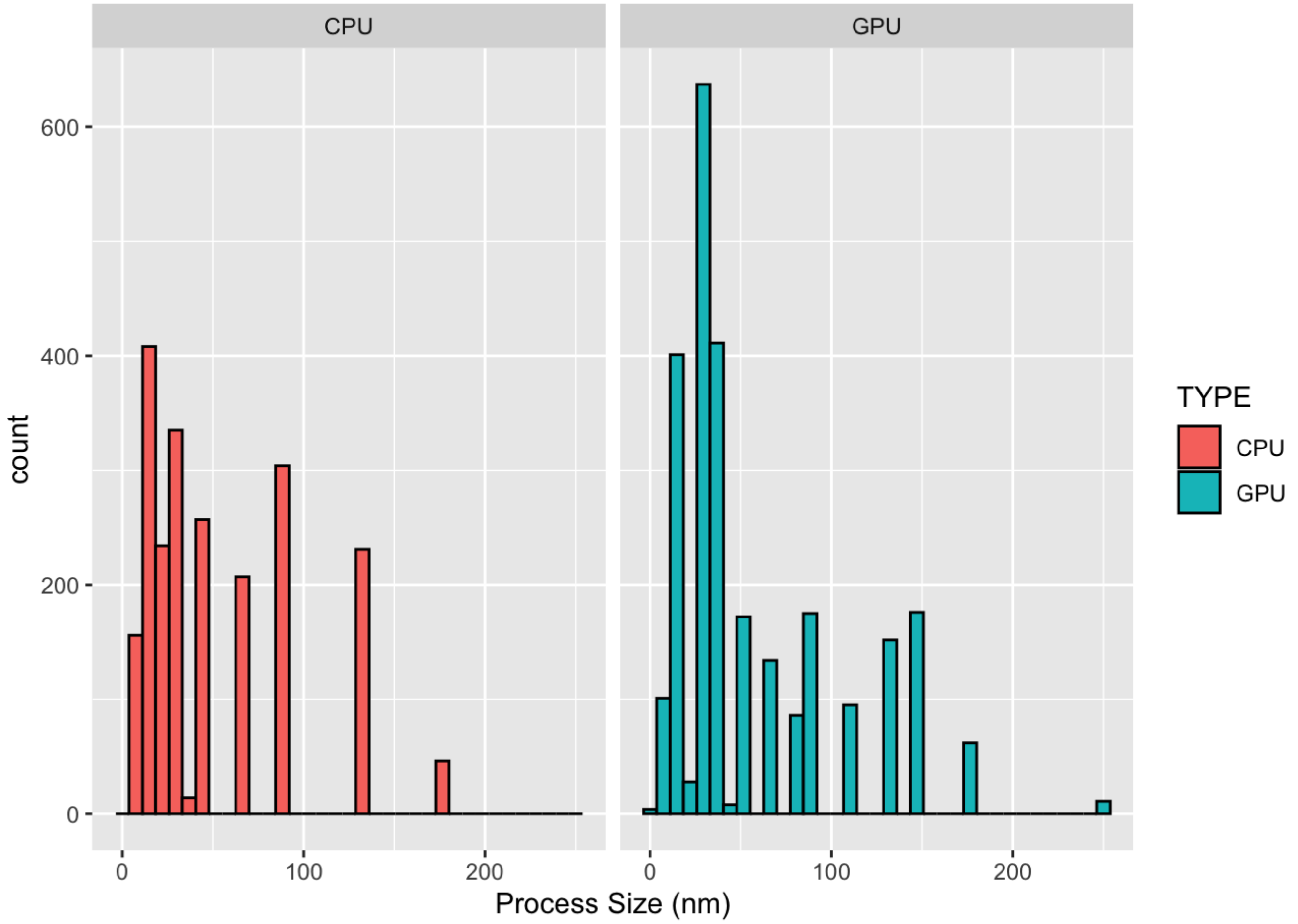
Graphically distribution of Process Size (nm)

```
# summary table
process_time <- cpu_gpu_data %>% group_by(TYPE,`Process Size (nm)`) %>% summarise(n =
n())
process_time
```

```
## # A tibble: 36 × 3
## # Groups:   TYPE [2]
##    TYPE  `Process Size (nm)`     n
##    <fct>               <dbl> <int>
##  1 CPU                     7    97
##  2 CPU                    10    59
##  3 CPU                    12    35
##  4 CPU                    14   373
##  5 CPU                    22   234
##  6 CPU                    28    28
##  7 CPU                    32   307
##  8 CPU                    40    14
##  9 CPU                    45   257
## 10 CPU                    65   207
## # … with 26 more rows
```

```
# histogram
ggplot(cpu_gpu_data, aes(x=`Process Size (nm)`,group=TYPE,fill=TYPE)) +
geom_histogram(bins=35,col="black",na.rm = FALSE) +
facet_wrap(~TYPE)
```

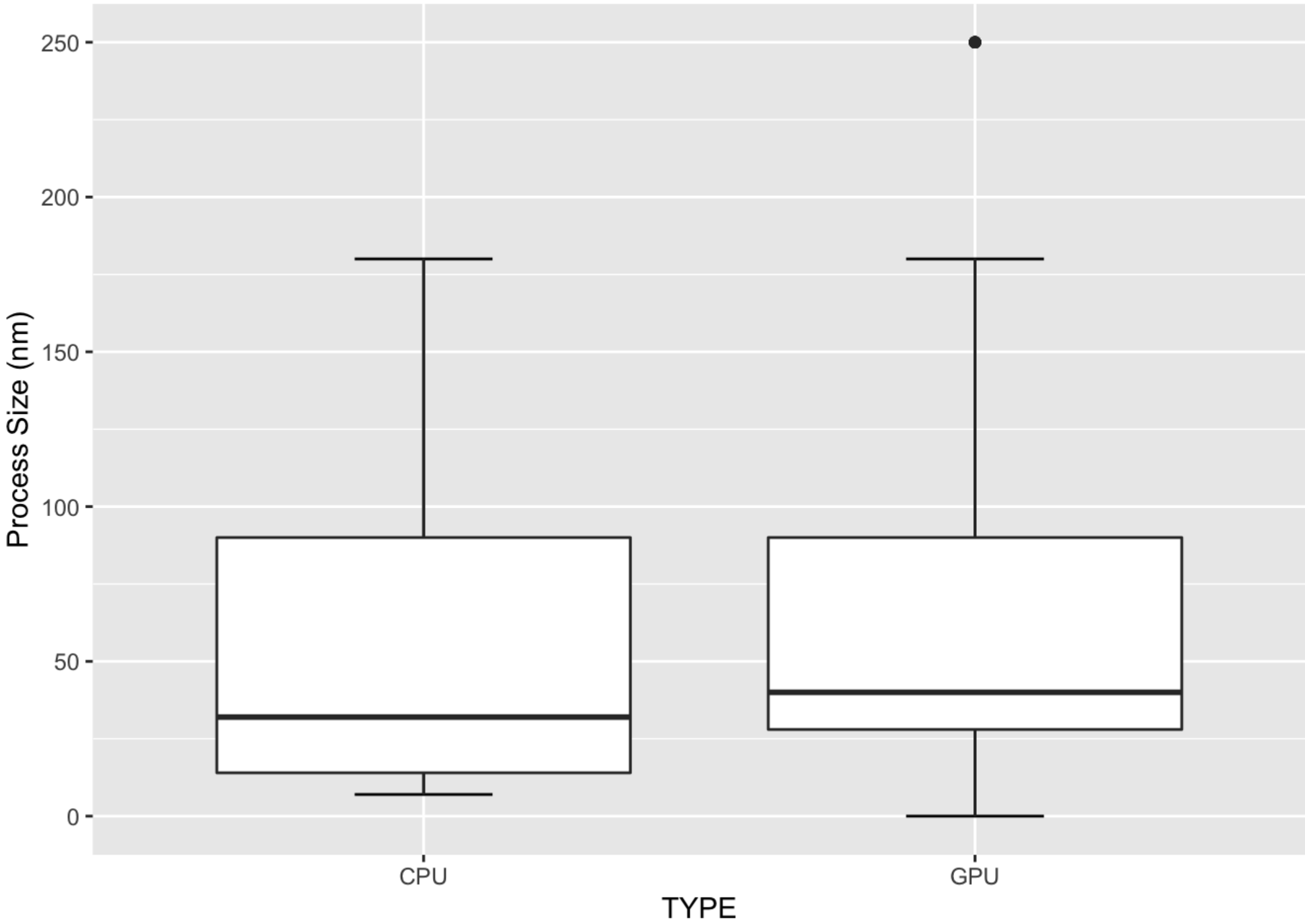
```
## Warning: Removed 9 rows containing non-finite values (stat_bin).
```




```
# box plot
ggplot(cpu_gpu_data,aes(x=TYPE,y=`Process Size (nm)`)) +
stat_boxplot(geom="errorbar",width=0.25) + geom_boxplot() +
ylab("Process Size (nm)")
```

```
## Warning: Removed 9 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 9 rows containing non-finite values (stat_boxplot).
```



From both histogram and box plot, the GPU seems have a larger mean value of process size. They both have a positive skew. But CPU seems have a wider spread on process size. Furthermore, GPU has an outlier showing in the the box plot, but CPU does not.

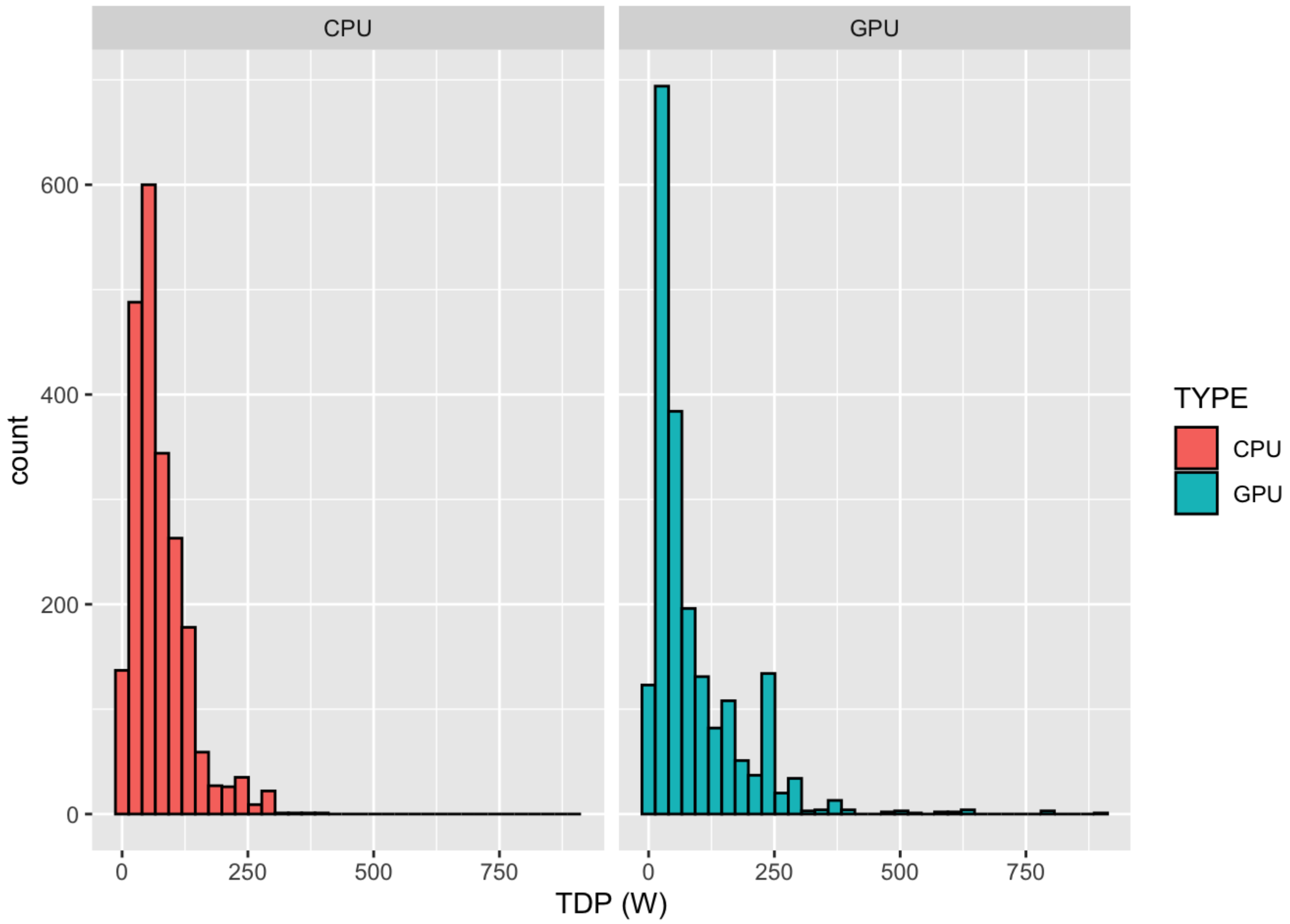
Graphically distribution of TDP (W)

```
TDP <- cpu_gpu_data %>% group_by(TYPE,`TDP (W)`) %>% summarise(n = n())
TDP
```

```
## # A tibble: 304 × 3
## # Groups:   TYPE [2]
##    TYPE  `TDP (W)`     n
##    <fct>     <dbl> <int>
##  1 CPU           1     6
##  2 CPU           2    14
##  3 CPU           3    15
##  4 CPU           4    12
##  5 CPU           5    16
##  6 CPU           6     6
##  7 CPU           7    17
##  8 CPU           8     8
##  9 CPU           9    16
## 10 CPU          10    17
## # … with 294 more rows
```

```
ggplot(cpu_gpu_data, aes(x=`TDP (W)`,group=TYPE,fill=TYPE)) +
geom_histogram(bins=35,col="black",na.rm = FALSE) +
facet_wrap(~TYPE)
```

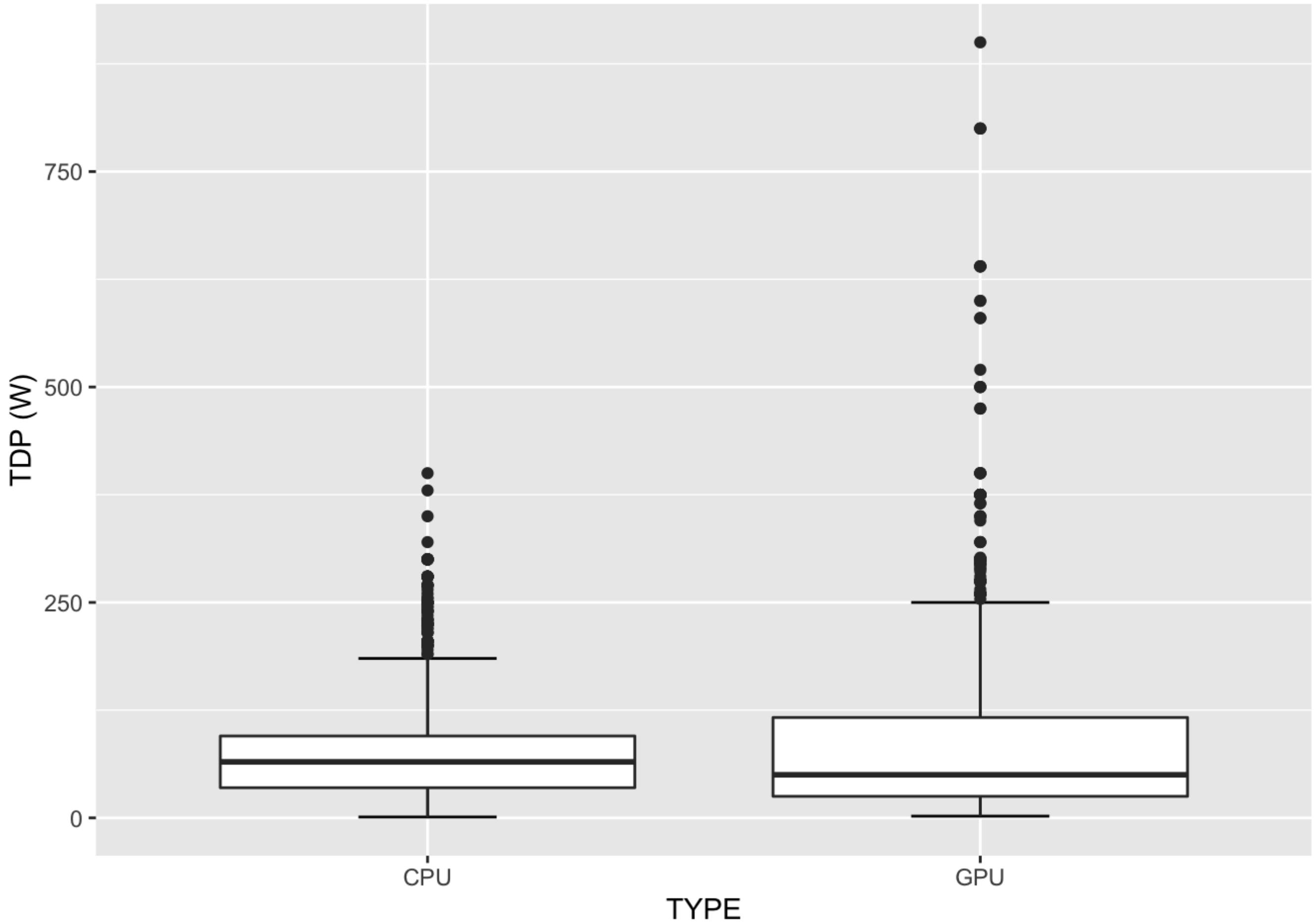
```
## Warning: Removed 626 rows containing non-finite values (stat_bin).
```




```
ggplot(cpu_gpu_data,aes(x=TYPE,y=`TDP (W)`)) +
stat_boxplot(geom="errorbar",width=0.25) + geom_boxplot() +
ylab("TDP (W)")
```

```
## Warning: Removed 626 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 626 rows containing non-finite values (stat_boxplot).
```



In term of TDP, CPU and GPU have a really similar central location, i.e, their mean and median are very close. They are both positive skew and have a narrow spread on TDP. GPU seem has a wider spread for central data than CPU. GPU has a lot outliers and spread widely toward large TDP compared to CPU.

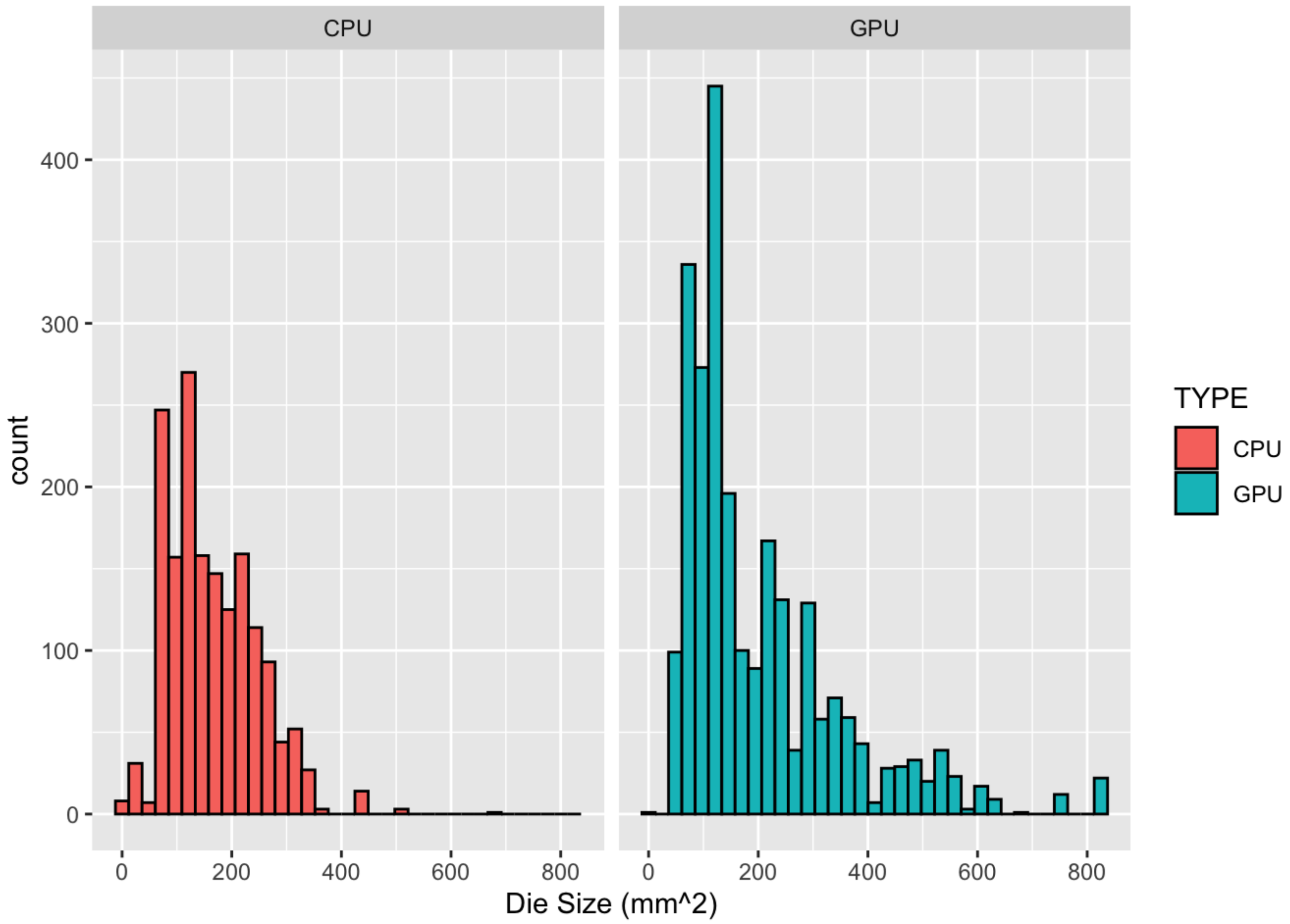
Graphically distribution of Die Size (mm^2)

```
die_size <- cpu_gpu_data %>% group_by(TYPE,`TDP (W)`) %>% summarise(n = n())
die_size
```

```
## # A tibble: 304 × 3
## # Groups:   TYPE [2]
##    TYPE  `TDP (W)`     n
##    <fct>     <dbl> <int>
##  1 CPU           1     6
##  2 CPU           2    14
##  3 CPU           3    15
##  4 CPU           4    12
##  5 CPU           5    16
##  6 CPU           6     6
##  7 CPU           7    17
##  8 CPU           8     8
##  9 CPU           9    16
## 10 CPU          10    17
## # … with 294 more rows
```

```
ggplot(cpu_gpu_data, aes(x=`Die Size (mm^2)`,group=TYPE,fill=TYPE)) +
geom_histogram(bins=35,col="black",na.rm = FALSE) +
facet_wrap(~TYPE)
```

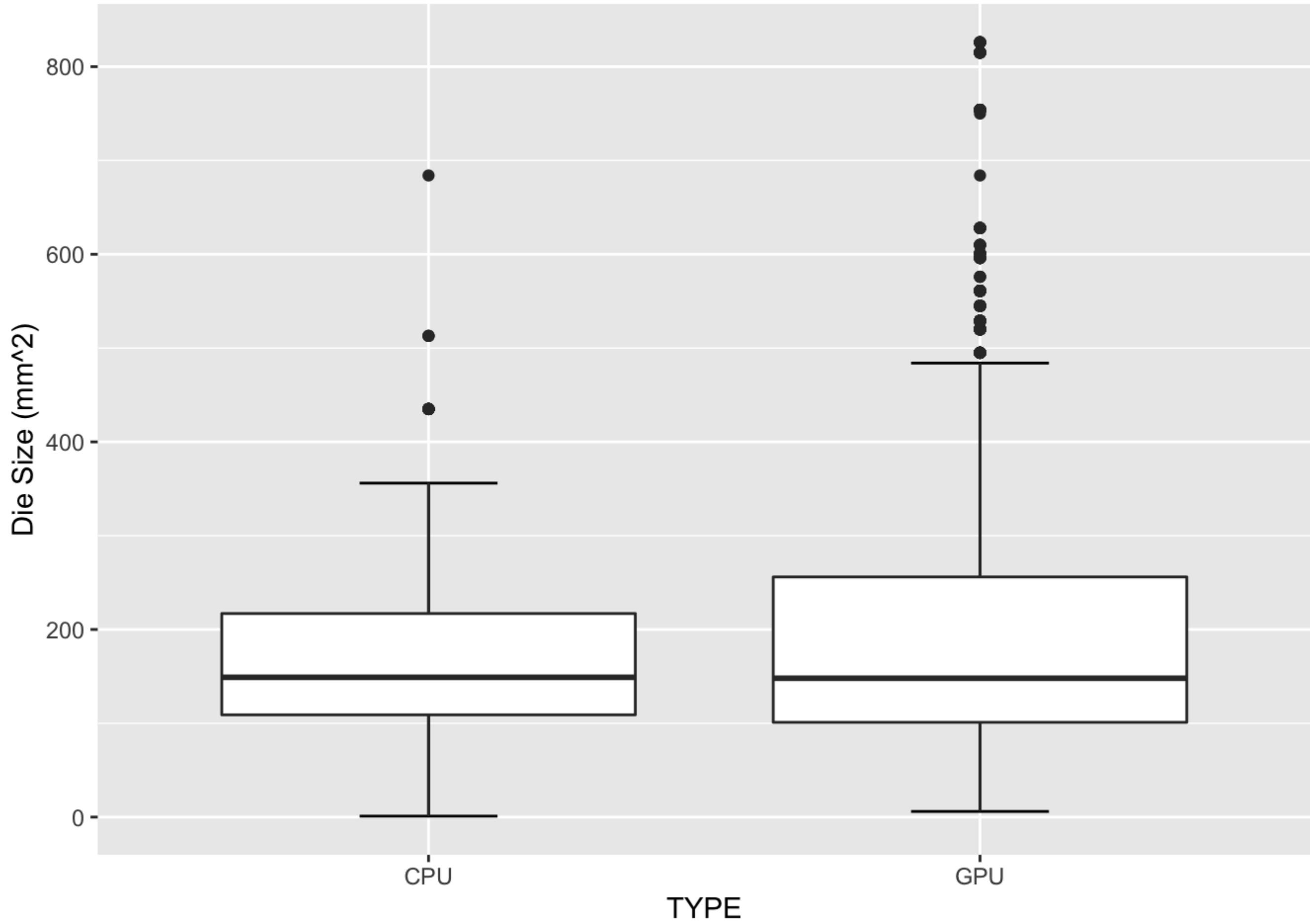
```
## Warning: Removed 715 rows containing non-finite values (stat_bin).
```




```
ggplot(cpu_gpu_data,aes(x=TYPE,y=`Die Size (mm^2)`)) +
stat_boxplot(geom="errorbar",width=0.25) + geom_boxplot() +
ylab("Die Size (mm^2)")
```

```
## Warning: Removed 715 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 715 rows containing non-finite values (stat_boxplot).
```



In terms of Die Size, there is no obvious differences between the median and mean. They are both positive skew. The die size of GPU spread relative wider than CPU, and it has more outliers than CPU.

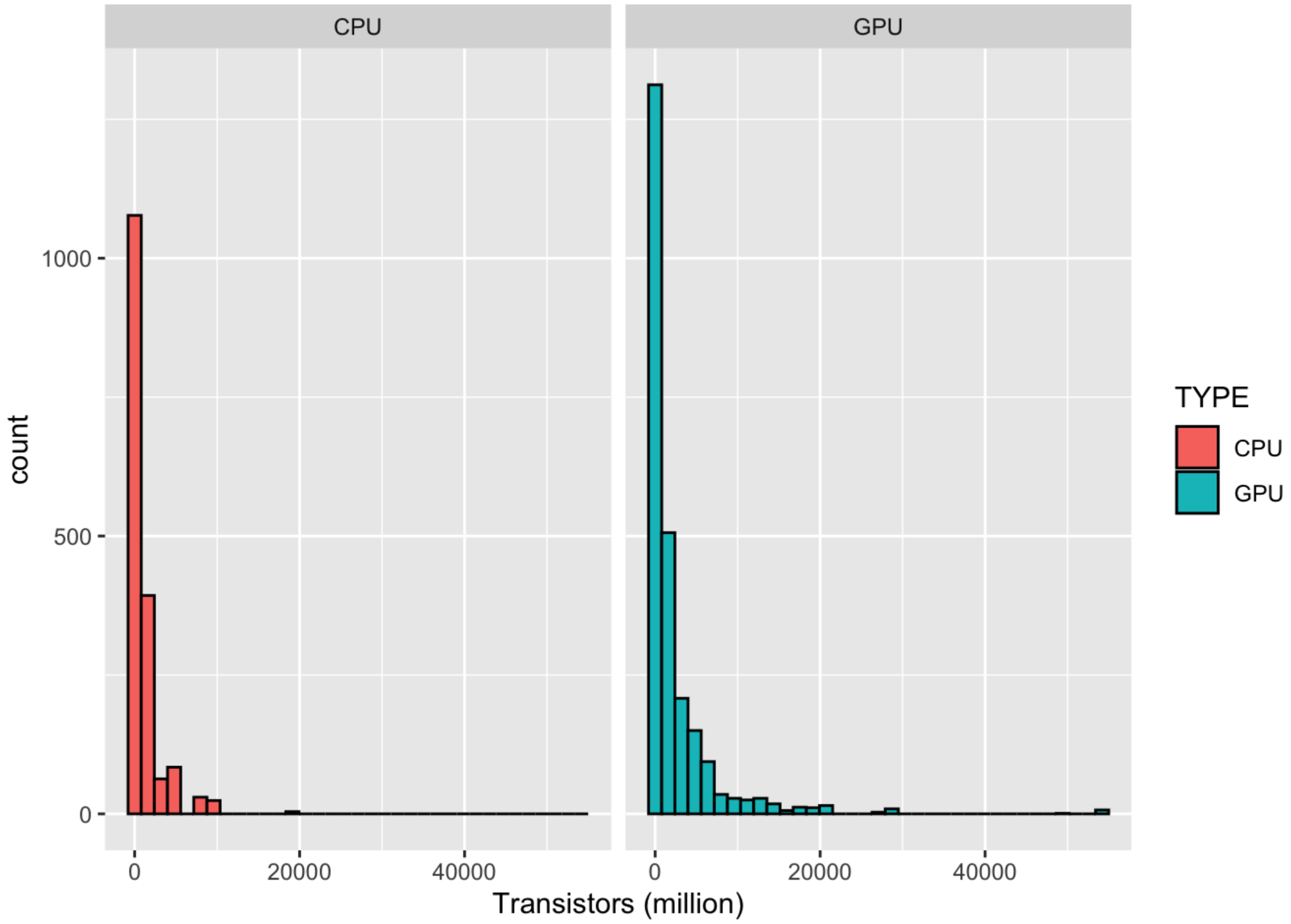
Graphically distribution of Transistors (million)

```
transistors <- cpu_gpu_data %>% group_by(TYPE,`TDP (W)`) %>% summarise(n = n())
transistors
```

```
## # A tibble: 304 × 3
## # Groups:   TYPE [2]
##    TYPE  `TDP (W)`     n
##    <fct>     <dbl> <int>
##  1 CPU           1     6
##  2 CPU           2    14
##  3 CPU           3    15
##  4 CPU           4    12
##  5 CPU           5    16
##  6 CPU           6     6
##  7 CPU           7    17
##  8 CPU           8     8
##  9 CPU           9    16
## 10 CPU          10    17
## # … with 294 more rows
```

```
ggplot(cpu_gpu_data, aes(x=`Transistors (million)`,group=TYPE,fill=TYPE)) +
geom_histogram(bins=35,col="black",na.rm = FALSE) +
facet_wrap(~TYPE)
```

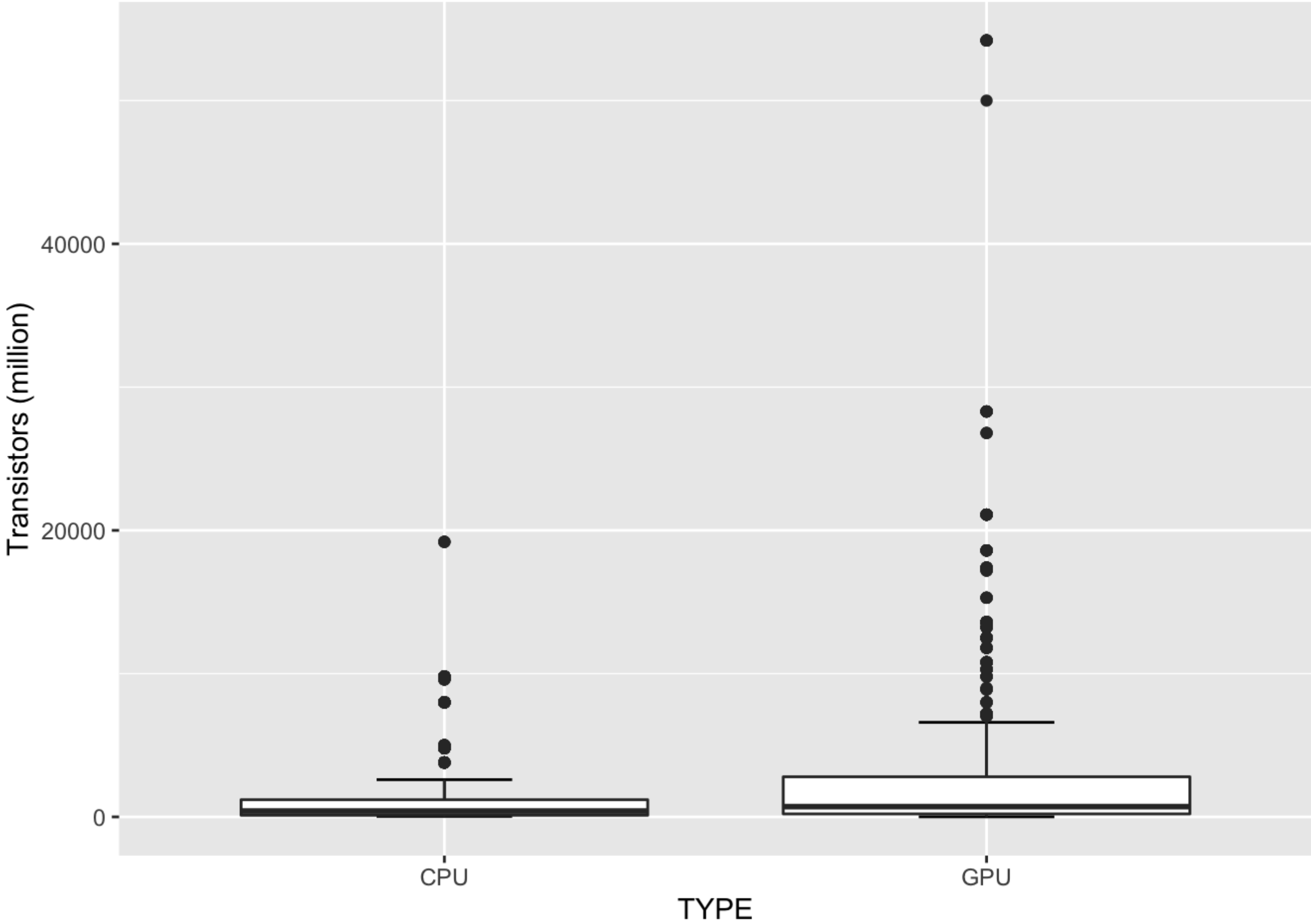
```
## Warning: Removed 711 rows containing non-finite values (stat_bin).
```




```
ggplot(cpu_gpu_data,aes(x=TYPE,y=`Transistors (million)`)) +
stat_boxplot(geom="errorbar",width=0.25) + geom_boxplot() +
ylab("Transistors (million)")
```

```
## Warning: Removed 711 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 711 rows containing non-finite values (stat_boxplot).
```



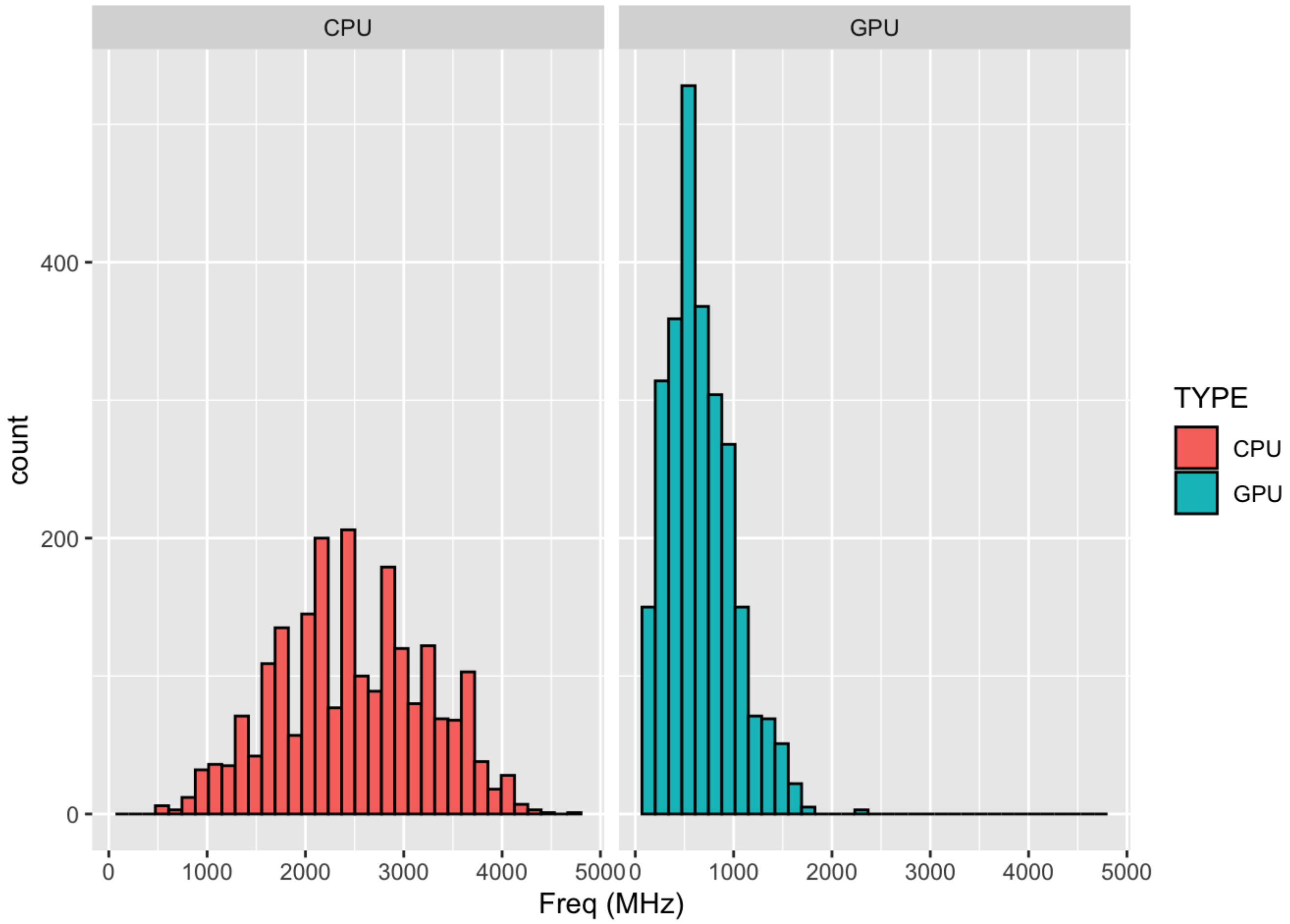
For Transistors, the spread of CPU and GPU are both very narrow. They have similar central location, and both positive skewed. But GPU seems has more and extreme outliers than CPU.

Graphically distribution of Freq (MHz)

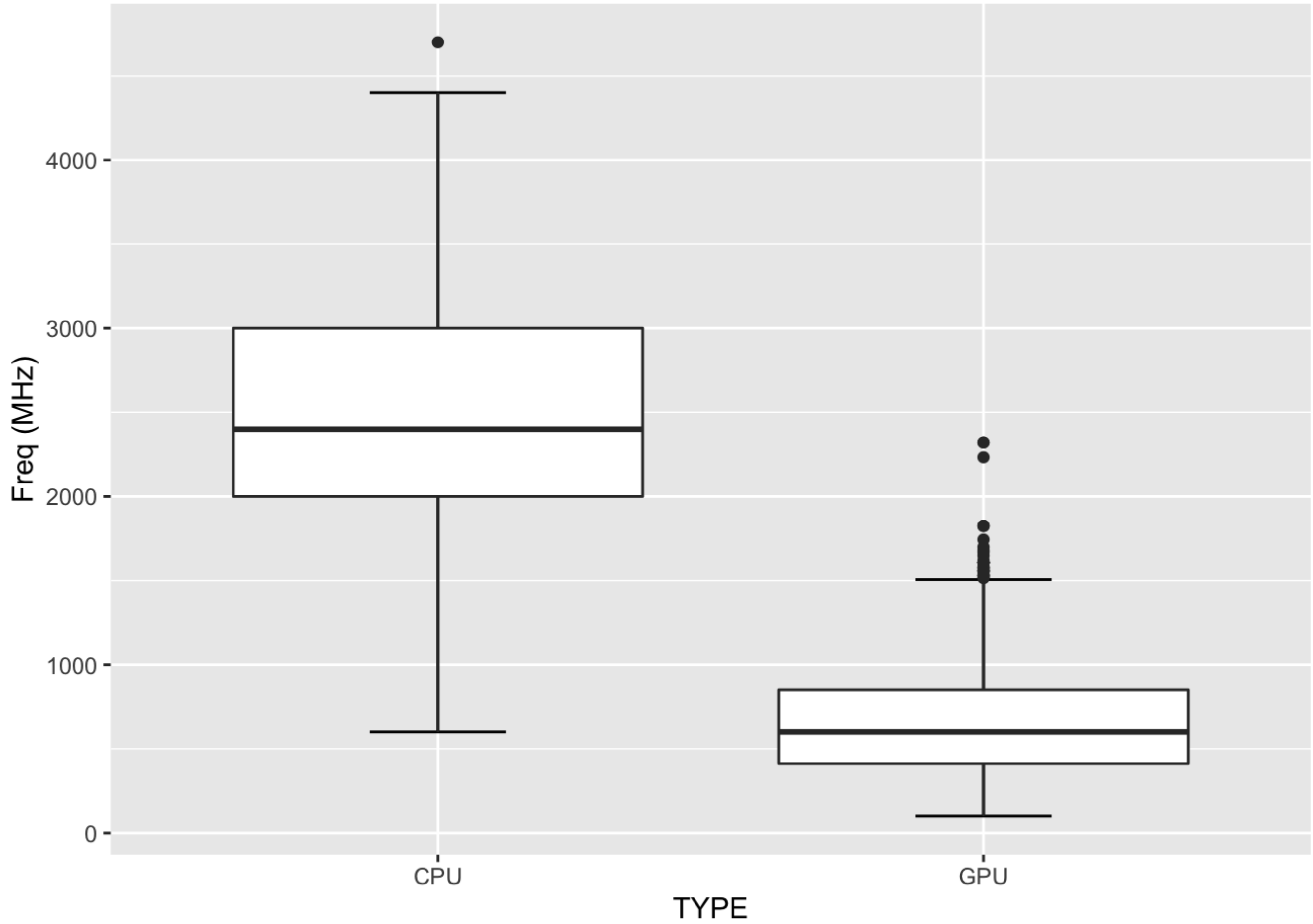
```
freq <- cpu_gpu_data %>% group_by(TYPE,`Freq (MHz)`) %>% summarise(n = n())
freq
```

```
## # A tibble: 548 × 3
## # Groups:   TYPE [2]
##    TYPE  `Freq (MHz)`     n
##    <fct>        <dbl> <int>
##  1 CPU            600     6
##  2 CPU            650     1
##  3 CPU            700     2
##  4 CPU            750     1
##  5 CPU            800     9
##  6 CPU            850     1
##  7 CPU            866     1
##  8 CPU            900     4
##  9 CPU            933     1
## 10 CPU            950     1
## # … with 538 more rows
```

```
ggplot(cpu_gpu_data, aes(x=`Freq (MHz)`,group=TYPE,fill=TYPE)) +
geom_histogram(bins=35,col="black",na.rm = FALSE) +
facet_wrap(~TYPE)
```




```
ggplot(cpu_gpu_data,aes(x=TYPE,y=`Freq (MHz)`)) +
stat_boxplot(geom="errorbar",width=0.25) + geom_boxplot() +
ylab("Freq (MHz)")
```



For Freq, there is a clear evidence that CPU has a larger mean and median than GPU. CPU looks very like a normally distribution and spread widely, but GPU is positive skewed and spread narrowly. Also, GPU has more outlier than CPU.

b.

Association between the number of processors released by the vendors and the foundries

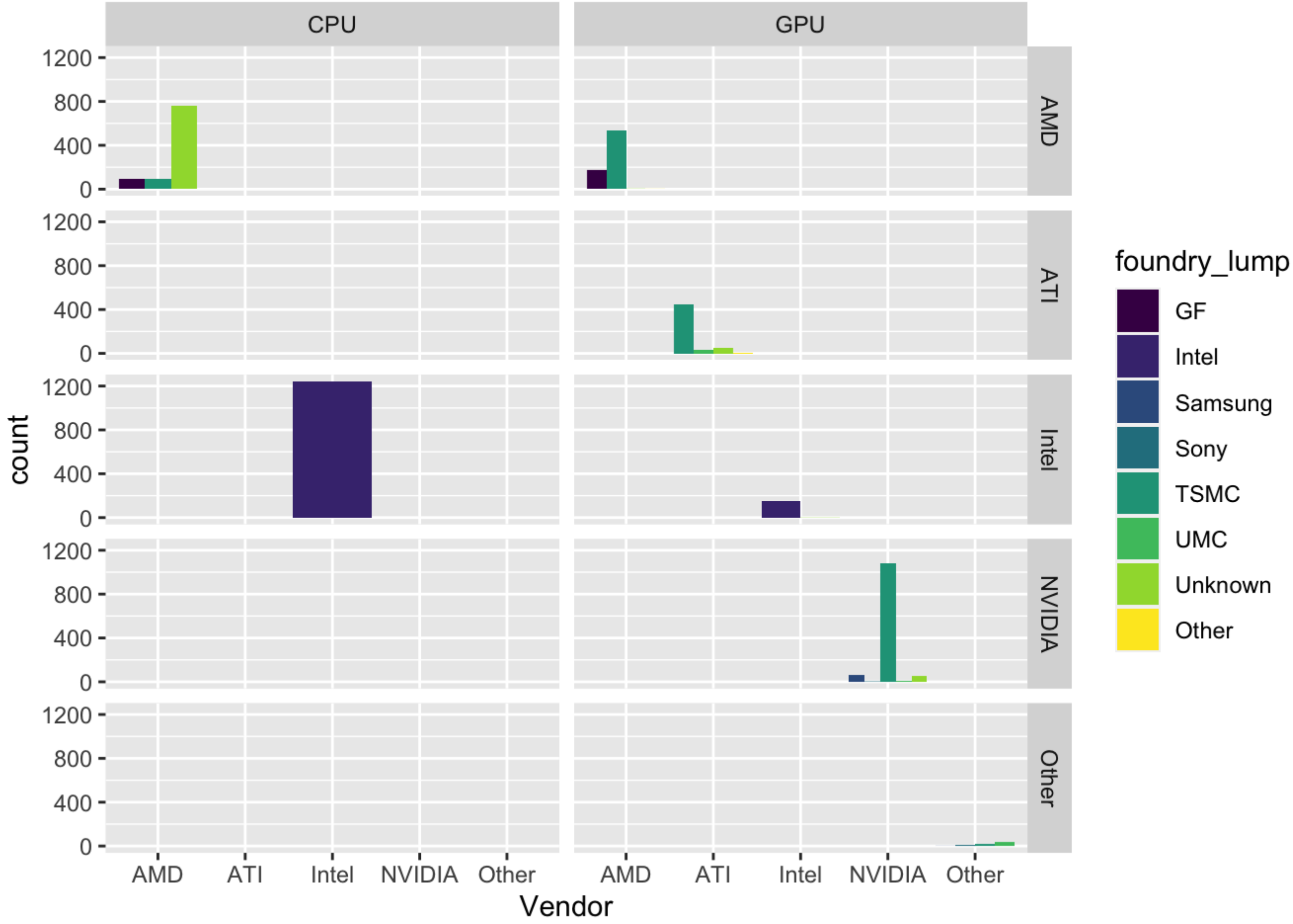
```
cpu_gpu_data %>% group_by(Vendor,Foundry,TYPE) %>% summarise(n=n())
```

```
## # A tibble: 24 × 4
## # Groups:   Vendor, Foundry [20]
##    Vendor Foundry TYPE      n
##    <chr>  <chr>   <fct> <int>
##  1 AMD    GF      CPU      93
##  2 AMD    GF      GPU     172
##  3 AMD    Renesas GPU       1
##  4 AMD    TSMC    CPU      97
##  5 AMD    TSMC    GPU     537
##  6 AMD    Unknown CPU     760
##  7 AMD    Unknown GPU       2
##  8 ATI    IBM     GPU       3
##  9 ATI    NEC     GPU       2
## 10 ATI    TSMC    GPU     449
## # … with 14 more rows
```

```
cpu_gpu_data %>% group_by(Vendor,Foundry,TYPE) %>% summarise(n=n()) %>% pivot_wider(i
d_cols = c(Vendor,Foundry), names_from = TYPE, values_from = n)
```

```
## # A tibble: 20 × 4
## # Groups:   Vendor, Foundry [20]
##    Vendor Foundry   CPU   GPU
##    <chr>  <chr>   <int> <int>
##  1 AMD    GF         93   172
##  2 AMD    Renesas    NA     1
##  3 AMD    TSMC       97   537
##  4 AMD    Unknown   760     2
##  5 ATI    IBM        NA     3
##  6 ATI    NEC        NA     2
##  7 ATI    TSMC       NA   449
##  8 ATI    UMC        NA    31
##  9 ATI    Unknown    NA    50
## 10 Intel  Intel    1242   148
## 11 Intel  Unknown    NA     2
## 12 NVIDIA Samsung    NA    59
## 13 NVIDIA Sony       NA     4
## 14 NVIDIA TSMC       NA  1078
## 15 NVIDIA UMC        NA     8
## 16 NVIDIA Unknown    NA    52
## 17 Other  Samsung    NA     1
## 18 Other  Sony       NA     6
## 19 Other  TSMC       NA    17
## 20 Other  UMC        NA    40
```

```
cpu_gpu_data <- cpu_gpu_data %>% mutate(foundry_lump = fct_lump(Foundry,7))
ggplot(cpu_gpu_data,aes(x=Vendor,fill= foundry_lump)) +
geom_bar(position="dodge")  + scale_fill_viridis_d() + facet_grid(Vendor~TYPE)
```



From the wider summary table in which one can see the numerical summary of the number of chip processor released across vendor and foundry grouped by CPU and GPU and the bar plot that has foundry lump collapsed into seven levels, one can find that for Vendor of AMD, it release semiconductors nearly exclusively from unknown foundries for CPU, but release most semiconductors from TSMC. Thus, there is an association between the number of processors released but it depends on the type. For ATL, its GPU semiconductors are almost all released from TSMC, and doesn't depend on type. For Intel, no matter which type, it released exclusively from Intel itself. For NVIDA, It released almost exclusively from TSMC, and doesn't depend on type. Overall, I would like to conclude that there is a strong association between the number of processors released by the vendors and the foundries, and doesn't depend on much whether they are CPU or GPU.

c.

Association between die size and TDP

```
corrolation = vector("numeric",2)
names(corrolation) = c("CPU","GPU")

data_split <- with(cpu_gpu_data,split(cpu_gpu_data,TYPE))
cor1 = with(data_split[[1]],cor(`Die Size (mm^2)`,`TDP (W)`,use="complete.obs"))
corrolation[1] = cor1

data_split <- with(cpu_gpu_data,split(cpu_gpu_data,TYPE))
cor2 = with(data_split[[2]],cor(`Die Size (mm^2)`,`TDP (W)`,use="complete.obs"))
corrolation[2] = cor2
print(corrolation)
```

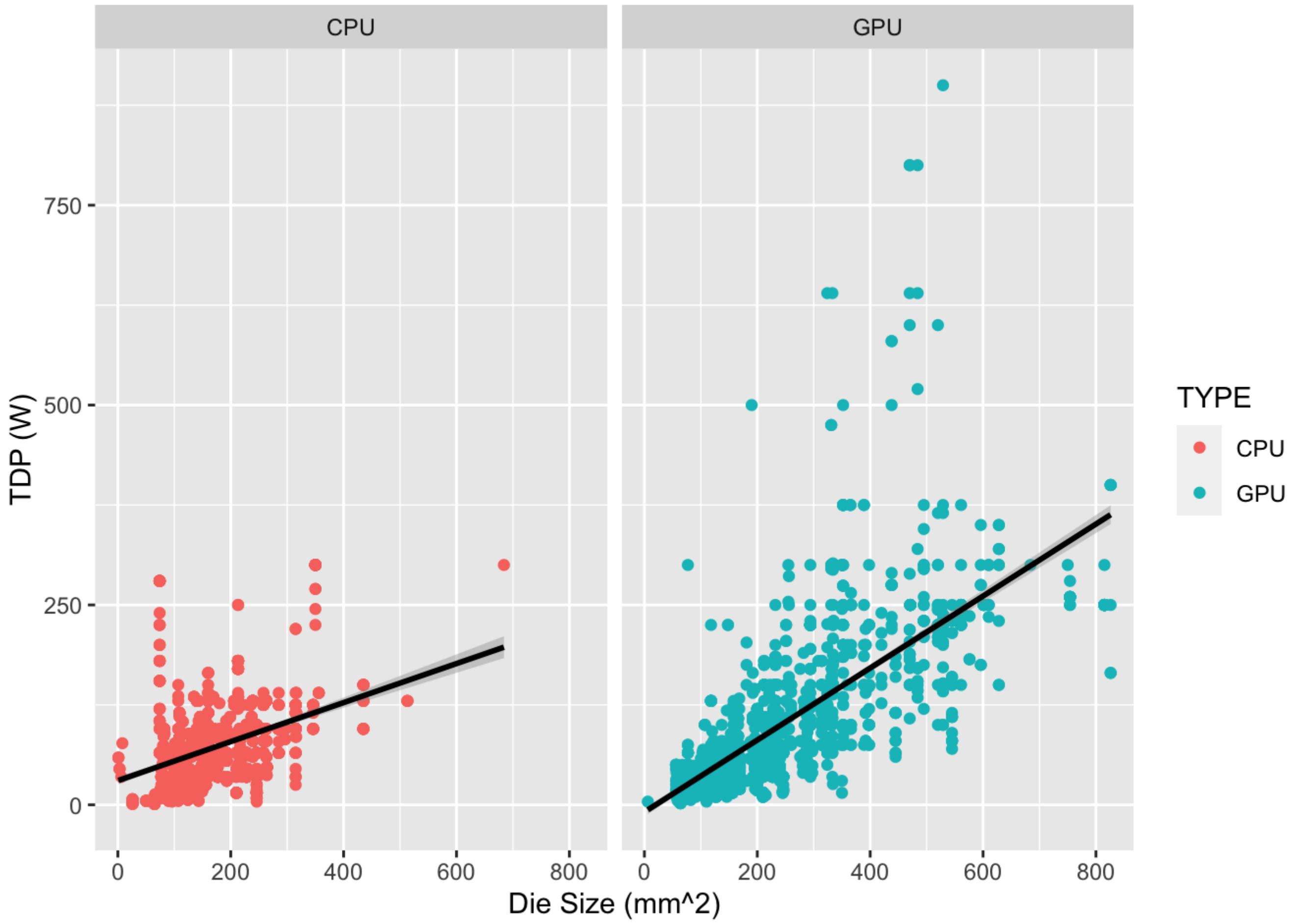
```
##       CPU       GPU
## 0.4108921 0.7309420
```

```
ggplot(cpu_gpu_data,aes(x=`Die Size (mm^2)`,y=`TDP (W)`,col=TYPE)) +
geom_point() + facet_wrap(~TYPE) + geom_smooth(method="lm",col="black")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 1286 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 1286 rows containing missing values (geom_point).
```



I computed the correlations between die size and TPD using `cor()` function and split the result according to the type. You can see the result that there is a stronger correlation for GPU between die size and TDP, but there is also a positive correlation for CPU between die size and TDP. But from the graph, there is more larger value of TDP as die size going larger for GPU. Therefore, It seems that this correlation is dependent on the type.

TASK 2