Data Overview

In this data set there are details of all the chips manufactured from 1999. The columns included are “Product” which contains the name of the chip. “Type” shoes whether it's GPU or CPU. “Release.Date” shows when the chip was first released into the market. “Process.Size..nm.” shows the fabrication technology. “TDP..W.” Shows the thermal metrics. “Transistors..million.” shows the no of transistors used to fabricate the chip. “Freq..MHz.” shows the clocking speed of chip.

## R Markdown

chip\_dataset <- read.csv("")

head(chip\_dataset)

## X Product Type Release.Date Process.Size..nm. TDP..W.  
## 1 0 AMD Athlon 64 3500+ CPU 2007-02-20 65 45  
## 2 1 AMD Athlon 200GE CPU 2018-09-06 14 35  
## 3 2 Intel Core i5-1145G7 CPU 2020-09-02 10 28  
## 4 3 Intel Xeon E5-2603 v2 CPU 2013-09-01 22 80  
## 5 4 AMD Phenom II X4 980 BE CPU 2011-05-03 45 125  
## 6 5 Intel Xeon E5-2470 v2 CPU 2013-09-01 22 95  
## Die.Size..mm.2. Transistors..million. Freq..MHz. Foundry Vendor FP16.GFLOPS  
## 1 77 122 2200 Unknown AMD NA  
## 2 192 4800 3200 Unknown AMD NA  
## 3 NA NA 2600 Intel Intel NA  
## 4 160 1400 1800 Intel Intel NA  
## 5 258 758 3700 Unknown AMD NA  
## 6 160 1400 2400 Intel Intel NA  
## FP32.GFLOPS FP64.GFLOPS  
## 1 NA NA  
## 2 NA NA  
## 3 NA NA  
## 4 NA NA  
## 5 NA NA  
## 6 NA NA

summary(chip\_dataset)

## X Product Type Release.Date   
## Min. : 0 Length:4854 Length:4854 Length:4854   
## 1st Qu.:1213 Class :character Class :character Class :character   
## Median :2426 Mode :character Mode :character Mode :character   
## Mean :2426   
## 3rd Qu.:3640   
## Max. :4853   
##   
## Process.Size..nm. TDP..W. Die.Size..mm.2. Transistors..million.  
## Min. : 0.00 Min. : 1.00 Min. : 1.0 Min. : 8   
## 1st Qu.: 22.00 1st Qu.: 33.00 1st Qu.:104.0 1st Qu.: 154   
## Median : 40.00 Median : 65.00 Median :148.0 Median : 624   
## Mean : 55.11 Mean : 81.36 Mean :188.4 Mean : 1930   
## 3rd Qu.: 90.00 3rd Qu.:100.00 3rd Qu.:239.0 3rd Qu.: 1550   
## Max. :250.00 Max. :900.00 Max. :826.0 Max. :54200   
## NA's :9 NA's :626 NA's :715 NA's :711   
## Freq..MHz. Foundry Vendor FP16.GFLOPS   
## Min. : 100 Length:4854 Length:4854 Min. : 10.02   
## 1st Qu.: 590 Class :character Class :character 1st Qu.: 768.80   
## Median :1074 Mode :character Mode :character Median : 2965.50   
## Mean :1484 Mean : 8397.46   
## 3rd Qu.:2400 3rd Qu.: 10600.00   
## Max. :4700 Max. :184600.00   
## NA's :4318   
## FP32.GFLOPS FP64.GFLOPS   
## Min. : 12.8 Min. : 3.60   
## 1st Qu.: 257.3 1st Qu.: 38.30   
## Median : 696.0 Median : 89.28   
## Mean : 2134.8 Mean : 363.67   
## 3rd Qu.: 2116.8 3rd Qu.: 220.00   
## Max. :40000.0 Max. :11540.00   
## NA's :2906 NA's :3548

glimpse(chip\_dataset)

## Rows: 4,854  
## Columns: 14  
## $ X <int> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14…  
## $ Product <chr> "AMD Athlon 64 3500+", "AMD Athlon 200GE", "Inte…  
## $ Type <chr> "CPU", "CPU", "CPU", "CPU", "CPU", "CPU", "CPU",…  
## $ Release.Date <chr> "2007-02-20", "2018-09-06", "2020-09-02", "2013-…  
## $ Process.Size..nm. <dbl> 65, 14, 10, 22, 45, 22, 65, 65, 10, 90, 130, 22,…  
## $ TDP..W. <dbl> 45, 35, 28, 80, 125, 95, 125, 130, 28, 89, 62, 3…  
## $ Die.Size..mm.2. <dbl> 77, 192, NA, 160, 258, 160, 285, 140, NA, 156, 8…  
## $ Transistors..million. <dbl> 122, 4800, NA, 1400, 758, 1400, 450, 376, NA, 15…  
## $ Freq..MHz. <dbl> 2200, 3200, 2600, 1800, 3700, 2400, 2400, 3000, …  
## $ Foundry <chr> "Unknown", "Unknown", "Intel", "Intel", "Unknown…  
## $ Vendor <chr> "AMD", "AMD", "Intel", "Intel", "AMD", "Intel", …  
## $ FP16.GFLOPS <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …  
## $ FP32.GFLOPS <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …  
## $ FP64.GFLOPS <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …

chip\_dataset<- data.frame(chip\_dataset)

chip\_dataset$Release.Date<-as.Date(chip\_dataset$Release.Date)

This is to change the datatype of Release.Date from character to date-time.

chip\_dataset[!duplicated(chip\_dataset$product)]

This is to remove the duplicate listed products to avoid distributing the metrics.

chip\_dataset$Release\_Year <- year(chip\_dataset$Release.Date)

This is to remove get the year from the timestamp as it is useful to group time in the coming steps.

chip\_dataset <- chip\_dataset %>% mutate(Transistors..million. = replace(Transistors..million.,  
 is.na(Transistors..million.),  
 median(Transistors..million., na.rm = T)))

This is to clean the Transistors..million. column to replace the null values with the median of the whole data. This avoids manipulating the mean.

## data frame with 0 columns and 4854 rows

chip\_dataset=subset(chip\_dataset,select = -c(X))

Data cleaning is done here as we don't need the column named “X”.

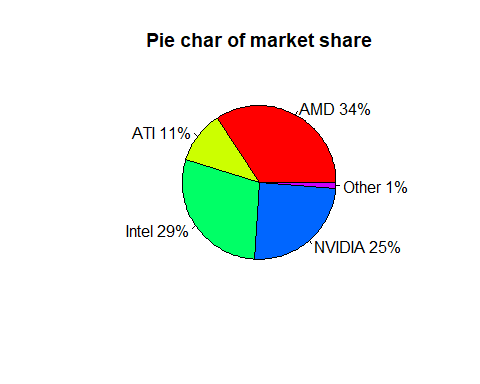
chip\_dataset<-chip\_dataset %>% arrange(chip\_dataset$Release.Date)

market\_share<-count(chip\_dataset,'Vendor')  
market\_share

## Vendor freq  
## 1 AMD 1662  
## 2 ATI 535  
## 3 Intel 1392  
## 4 NVIDIA 1201  
## 5 Other 64

After counting the number of course produced by individual manufacturer we can see that AMD stood at first with 1662 chips produced over the period of 1999 to 2021. Intel stood second by producing 1392 chips. At least ATI produced only 535 during the course of time.

vals<- market\_share$freq  
labs <- market\_share$Vendor  
percentage <- round(vals/sum(vals)\*100)  
labs <- paste(labs,percentage)  
labs <- paste(labs,"%",sep="")  
pie(vals,labs,col=rainbow(length(labs)),main = "Pie char of market share")



Looking at the Pie chart we can say that AMD capture 34% of the market and Intel stood as a competitor with 29%.

gpu\_count <- chip\_dataset %>% group\_by(chip\_dataset$Vendor,chip\_dataset$Type) %>% dplyr::summarise(rec\_count = n())

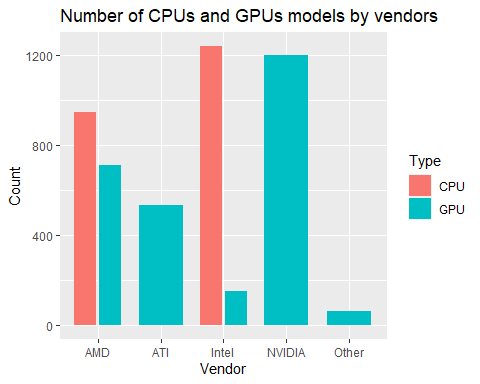
## `summarise()` has grouped output by 'chip\_dataset$Vendor'. You can override  
## using the `.groups` argument.

gpu\_count

## # A tibble: 7 × 3  
## # Groups: chip\_dataset$Vendor [5]  
## `chip\_dataset$Vendor` `chip\_dataset$Type` rec\_count  
## <chr> <chr> <int>  
## 1 AMD CPU 950  
## 2 AMD GPU 712  
## 3 ATI GPU 535  
## 4 Intel CPU 1242  
## 5 Intel GPU 150  
## 6 NVIDIA GPU 1201  
## 7 Other GPU 64

ggplot(gpu\_count, aes(x = gpu\_count$`chip\_dataset$Vendor`,fill= gpu\_count$`chip\_dataset$Type` )) +  
 geom\_bar(  
 aes(y = gpu\_count$rec\_count),  
 stat = "identity", position = position\_dodge(0.8),  
 width = 0.7  
 ) +  
 scale\_color\_manual(values = c("#0073C2FF", "#EFC000FF"))+  
 scale\_fill\_manual(values = c("#0073C2FF", "#EFC000FF"))+ labs(x = 'Vendor', y = 'Count',   
 title = "Number of CPUs and GPUs models by vendors")+scale\_fill\_discrete(name = "Type")

## Scale for 'fill' is already present. Adding another scale for 'fill', which  
## will replace the existing scale.



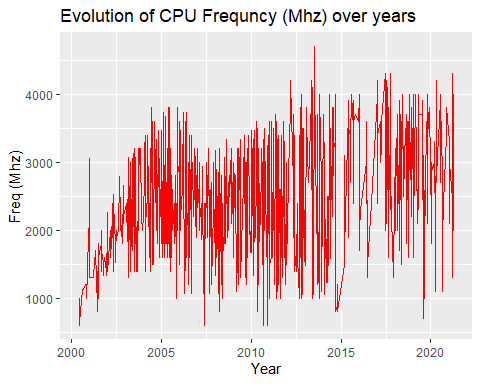
Looking at this we can conclude that producing almost equal number of CPU and GPU made the manufacturer the most selling brand. Intel was only concentrating on the CPU compare to the GPU, it produced 1242 CPU’s and just 150 GPU. Nvidia had concentrated only on the GPU’s and produced 1201 chips. The same idea was followed by ATI as it only Produced 530 GPU’s.

cpu\_freq <- filter(chip\_dataset, chip\_dataset$Type == "CPU")  
cpu\_freq$Release.Date<- as.Date(cpu\_freq$Release.Date)  
  
ggplot(cpu\_freq,aes(x = cpu\_freq$Release.Date, y = cpu\_freq$Freq..MHz.)) + geom\_line(color = "red") + labs(x = 'Year', y = 'Freq (Mhz)') +ggtitle("Evolution of CPU Frequncy (Mhz) over years")

## Warning: Use of `cpu\_freq$Release.Date` is discouraged. Use `Release.Date`  
## instead.

## Warning: Use of `cpu\_freq$Freq..MHz.` is discouraged. Use `Freq..MHz.` instead.

## Warning: Removed 3 row(s) containing missing values (geom\_path).



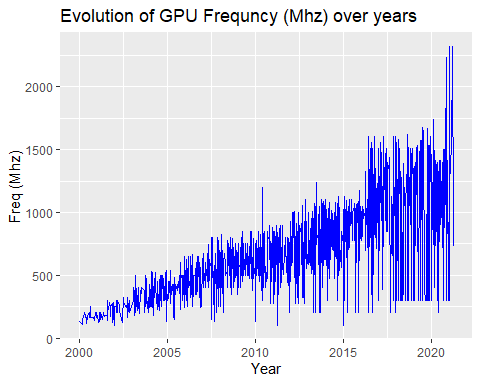
This graph shows on an average the frequency of CPU in the early 2000’s was around 430 Mhz and it increased steadily to around 4600Mhz in 2021. We can also see that over a course of time (2004-2008O there is no significant change in the frequency of the chips.

gpu\_freq <- filter(chip\_dataset,chip\_dataset$Type=="GPU")  
gpu\_freq$Release.Date<- as.Date(gpu\_freq$Release.Date)  
ggplot(gpu\_freq,aes(x = gpu\_freq$Release.Date, y = gpu\_freq$Freq..MHz.)) + geom\_line(color = "blue") + labs(x = 'Year', y = 'Freq (Mhz)') +ggtitle("Evolution of GPU Frequncy (Mhz) over years")

## Warning: Use of `gpu\_freq$Release.Date` is discouraged. Use `Release.Date`  
## instead.

## Warning: Use of `gpu\_freq$Freq..MHz.` is discouraged. Use `Freq..MHz.` instead.

## Warning: Removed 72 row(s) containing missing values (geom\_path).



This graph shows that GPU’S clocked at 200Mhz frequency in the year 2000 and increased steeply to above 2500 in the year 2021. During the time period of 2004 to 2008 the average frequency was around 800Mhz.

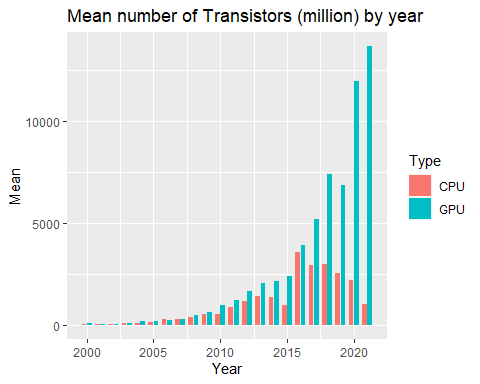
gpu\_count2 <- chip\_dataset %>% group\_by(Release\_Year,Type) %>% summarise(  
 a\_mean=(mean(Transistors..million.)))

## `summarise()` has grouped output by 'Release\_Year'. You can override using the  
## `.groups` argument.

ggplot(gpu\_count2, aes(x = gpu\_count2$Release\_Year,fill= gpu\_count2$Type )) +  
 geom\_bar(  
 aes(y = gpu\_count2$a\_mean),  
 stat = "identity", position = position\_dodge(0.8),  
 width = 0.7  
 ) +  
 scale\_color\_manual(values = c("#0073C2FF", "#EFC000FF"))+  
 scale\_fill\_manual(values = c("#0073C2FF", "#EFC000FF"))+ labs(x = 'Year', y = 'Mean',   
 title = "Mean number of Transistors (million) by year")+scale\_fill\_discrete(name = "Type")

## Scale for 'fill' is already present. Adding another scale for 'fill', which  
## will replace the existing scale.

## Warning: Removed 2 rows containing missing values (geom\_bar).



This graph shows the rapid increase of transistors in the chip. in 2000 there are around 200 million transistors and the no of transistors increased from 2010 and showed a positive trend until 2018 where the count was at 7500 million transistors. In the year 2021, the total number of transistors used in the fabrication was over 16000 million.