Project 1

Joshua Vong

10/7/2021

R Markdown

The data set domain I choose to explore are Graphical Processing Units(GPU). The reason why this appeals to me is that I have a background of building desktop personal computers. Seeing all the graphic cards and their attributes really interests me on their performance. There are so many different types of GPUs with some of the same architecture, but their performance differs due to other attributes. As for where I got my data set, I got it from kaggle, from the user ilissek, who web scrapped from intel, game-debate, and the brand companies listed in the table such as Nvidia and AMD.

Data Science Questions:

1. Which Manufacturer will continue to support future resolutions with their graphic cards?
2. Will the Max power of GPUs need higher PSU(Power supply unit)?
3. Can we tell what the prices be for future GPUs depending on the core speed of the GPU?

Data science questions addressed:

For these questions I think what would be important for my graphs would be to group the Manufacturers and compare the performance of their GPUs with each other, like the first question about resolutions, I can group the companies and see which one has more for certain resolutions. The second question can have another graph to see the PSU and what the Max_Power the GPU can produce. As for the third question, maybe a point graph with a line going through to see how much the prices have increased over the years? I think the dataset I found should be sufficient to answer these questions. For the implications that these questions imply, I would say it will show which company has GPUs with different attributes and how they compare to each other.

```
suppressMessages(library(tidyverse))
suppressMessages(library(tinytex))
suppressMessages(library(ggplot2))
allgpu<-read.table("~/Downloads/archive/All_GPUs.csv",header = TRUE, sep = ",",stringsAsFactors = TRUE)
```

```
## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## EOF within quoted string
```

```
## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## number of items read is not a multiple of the number of columns
```

Variable Definitions

Architecture - Type of system that the GPU is based on, it sets a sort of base for the GPU, think of skeleton frame.

Best_Resolution - The preferred resolution that the GPU should run most effectively on

Boost_Clock - max frequency that the GPU will run with nominal power and heat

Core_Speed - Speed of the GPU, measured in MHz (mega hertz)

DVI_Connection - Amount of digital visual interface (DVI) ports available

Direct_x - collection of application programs that help run programs that are graphic heavy, column shows which version it is on

Dedicated - Represents dedicated memory, memory that is exclusively for the GPU, this column tells if it has one (yes/no)

DisplayPort_Connection - Interface technology that allows the GPU to be displayed, column shows how many ports are on the GPU

HDMI_Connection - High definition interface technology that displays GPU, shows how many ports are there

Integrated - GPU that is built into the processor

L2_cache - Memory bank that the GPU contains, measured in KB(kilobytes)

Manufacturer - Type of brand that the GPU came from

Max_Power - Max wattage that the GPU can produce, measured in watts

Memory - Amount of memory that a GPU has, used for running intensive graphic programs such as video games, the more intense the graphic the more larger the memory, measured in MB(megabytes)

Memory_Bandwidth - The rate at which data can be read from or stored into a semiconductor memory by a processor, measured in GB/sec

Memory_Bus - Made of two parts, the data bus and address bus, data bus is a transfer of information between the memory and the address bus, the address bus communicates with the system on where the specific information can be located or stored when data enters or leaves the memory, measured in bits

Memory_Speed - The amount of time that it takes RAM to receive a request from the processor and then read or write data, measured in MHz

Memory_Type - Type of memory that the GPU has

Name - Name of the graphic card

Notebook_GPU - GPU that are built into laptops

Open_GL - cross-language, cross-platform application programming interface for rendering 2D and 3D vector graphics, this column lists the version for the GPU

PSU - Power supply unit recommended to run the GPU, measured in wattage and amps

Pixel_Rate - Pixel rate is the maximum amount of pixels the GPU could possibly write to the local memory in one second, measured in Gpixel/s

Power_Connector - Power connectors are devices that allows an electrical current to pass through it for the exclusive purpose of providing power to GPU, shows what type of power connector is needed

Process - Size of the processor in the GPU, the processor helps perform calculations, for the interface in the GPU, measured in nm (nano meters)

ROPs - The render output unit, often abbreviated as “ROP”, and sometimes called raster operations pipeline, is a hardware component in modern graphics processing units and one of the final steps in the rendering process of modern graphics cards

Release__Date - The date in which the GPU was released

Release__Price - Price of the GPU at release

SLI__Crossfire - Able to use two of the same GPU together

Shader - Shader is a type of computer program originally used for shading in 3D scenes (the production of appropriate levels of light, darkness, and color in a rendered image). They now perform a variety of specialized functions in various fields within the category of computer graphics special effects, or else do video post-processing unrelated to shading, or even perform functions unrelated to graphics, column shows what type of shader the GPU has

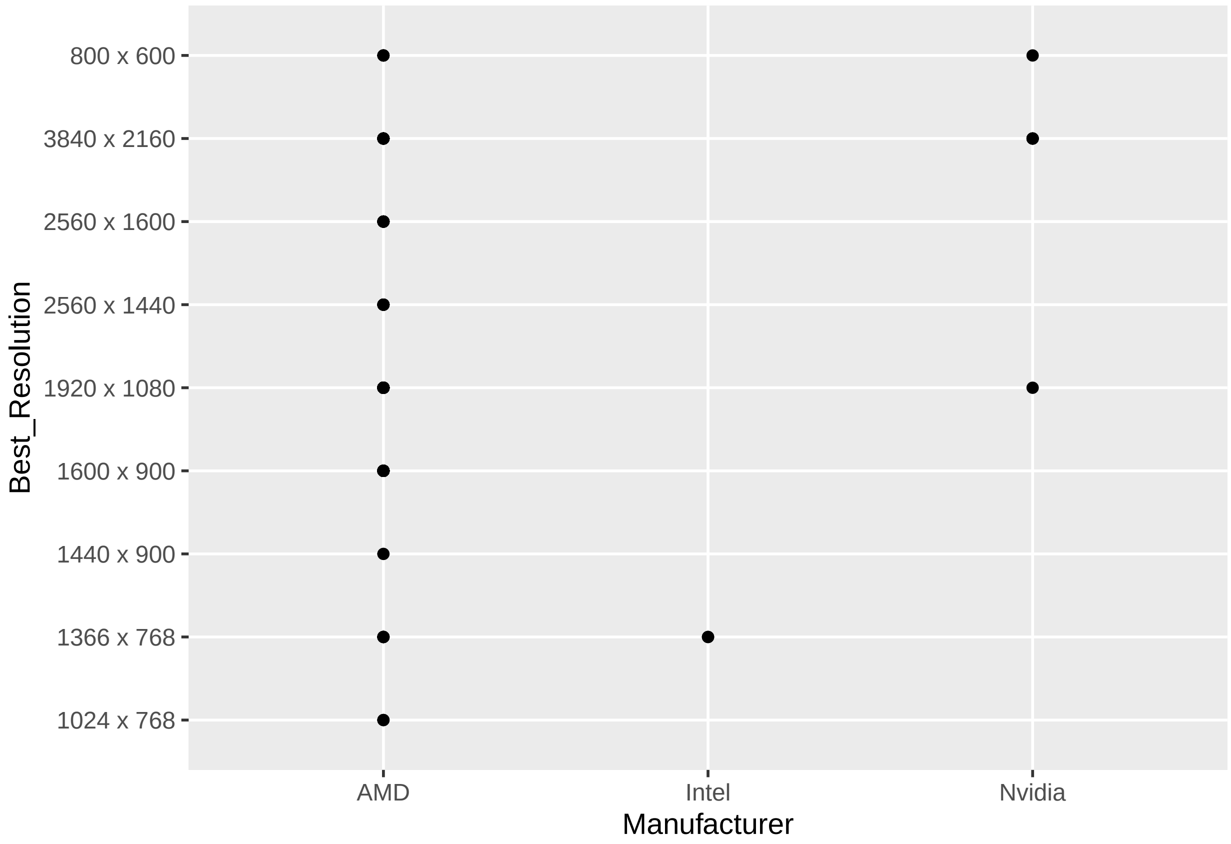
TMUs - A texture mapping unit (TMU) is a component in modern graphics processing units (GPUs). Historically it was a separate physical processor. A TMU is able to rotate, resize, and distort a bitmap image (performing texture sampling), to be placed onto an arbitrary plane of a given 3D model as a texture

Texture__Rate - The texture filter rate of the GPU is representative of how many pixels (specifically 'texels') the GPU can render per second, measured in Gtexel/s

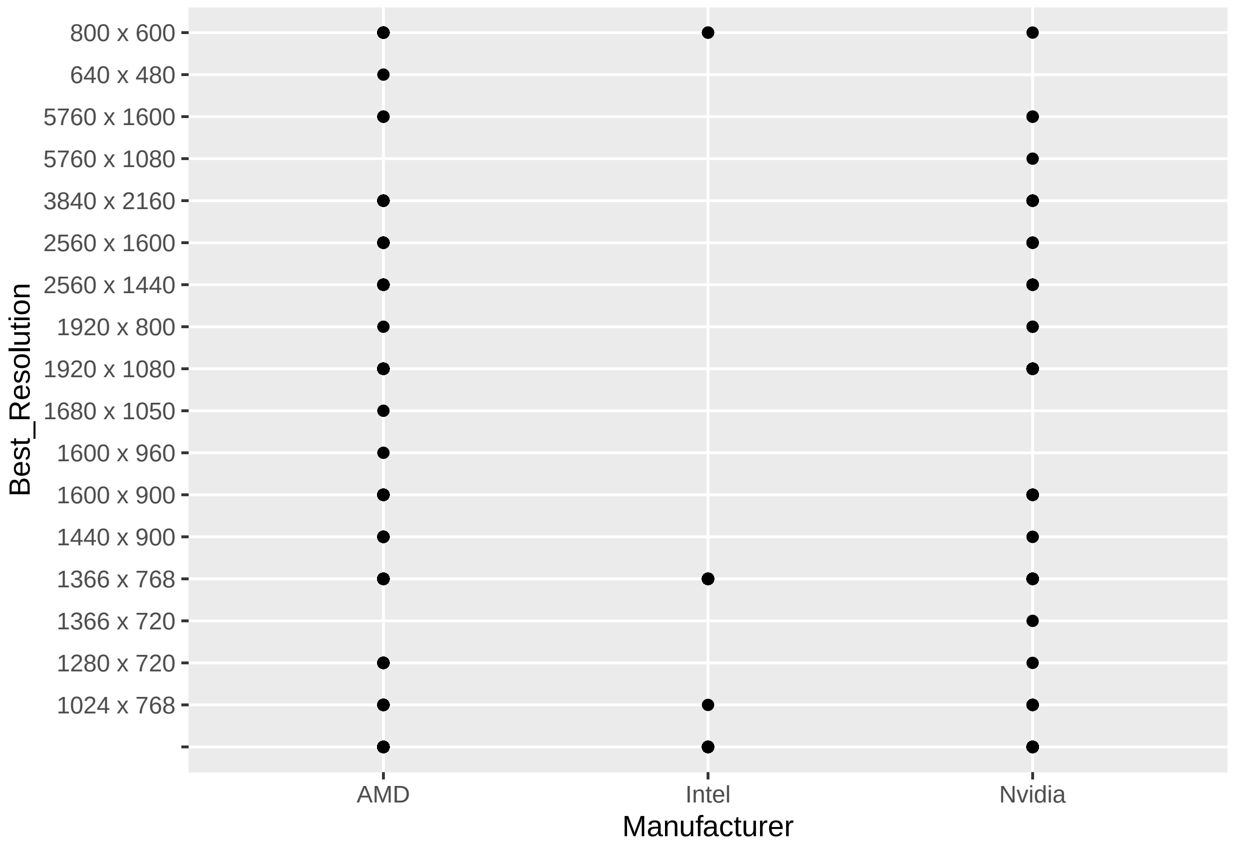
```
newDataGpu <- na.omit(allgpu)
```

What I did here was remove the values that had NA in its columns and row, in doing so the amount of objects has reduced to 88 obs but with the same columns. In hindsight I think this is better to get rid of unneeded data.

```
ggplot(data = newDataGpu)+geom_point(aes(x = Manufacturer, y = Best_Resolution))
```



```
ggplot(data = allgpu)+geom_point(aes(x = Manufacturer, y = Best_Resolution))
```

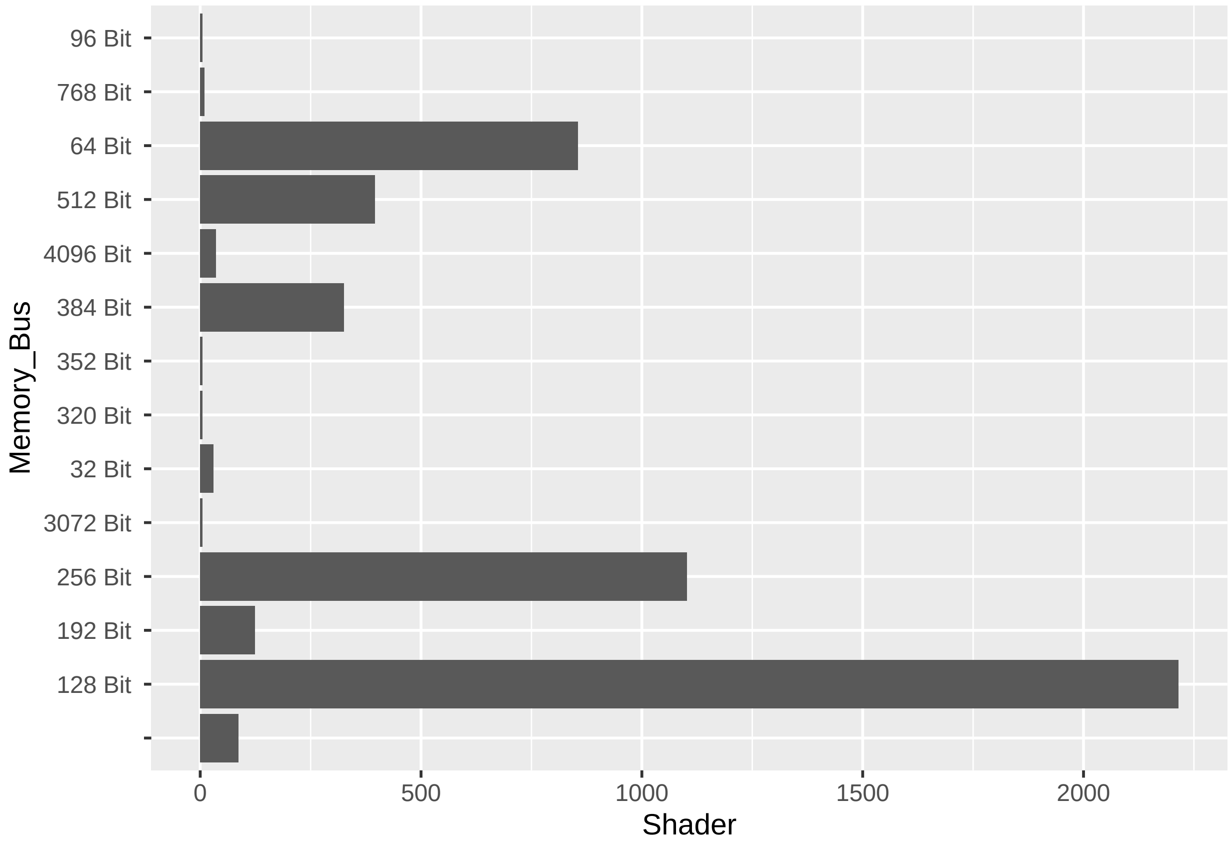


With this data visualization, I made a plot graph to see which company had graphics card for certain resolutions. According to this plot it looks like the manufacturer AMD has graphic cards for each resolution. Nvidia doesn't seem to have as much graphic cards dedicated to certain resolutions, they seem to be focused on 1920 x 1080 (1080p) and 3840 x 2160(4k), with an outlier of a 800 x 600. Intel only seems to have one graphic card which is 1366 x 768, this doesn't surprise me as intel are more known for their CPUs rather than their GPUs. I should also mention that if the GPU can run at a certain resoultion, it will be able to run resolutions lower than it. For example Nvidia has a graphic card that can run at a 4k resolution, as of right now that is the highest resolution, it will be able to run all the other resolutions easy.

```
ggplot(data = allgpu)+geom_histogram(aes(x = Shader, y = Memory_Bus), stat = "identity")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
## Warning: Removed 28 rows containing missing values (position_stack).
```

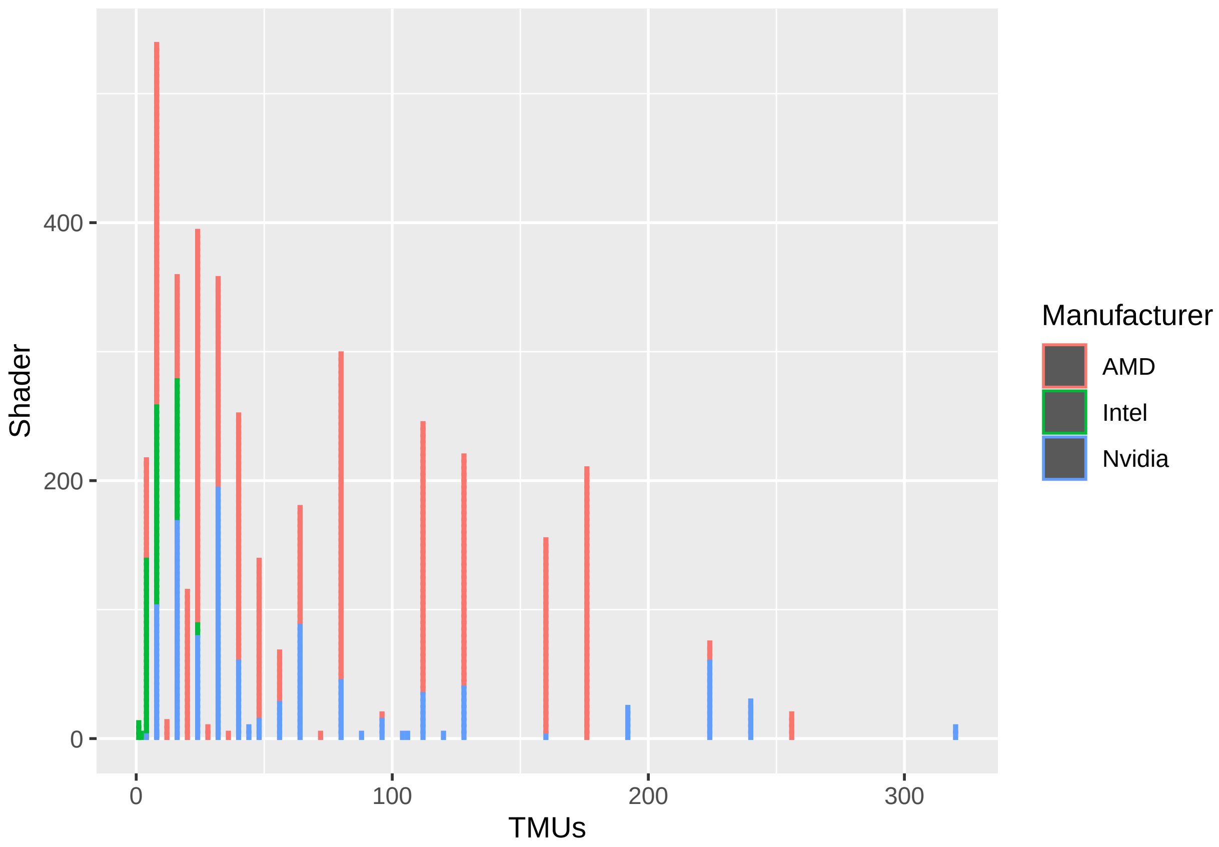


For this visualization, I wanted to compare the memory bits with the amount of shaders the GPUs have, my original thought was that the higher the memory bit the larger the amount of shaders it would have, this was not the case at all. It seems like the largest amount of shaders come from the 128 bit instead of the bits that are higher than 128 bits. It also seems like that higher memory bits are the lowest. I think the reason why higher memory bits don't have bigger shaders is because they might not be optimized especially with newer graphic cards. The graphic cards with 128 bits seem to be optimized since it has been out longer compared to a GPU with 4096 bits. The memory bits with 0 shaders could be because the dataset does not have graphic cards with those bits.

```
ggplot(data = allgpu)+geom_bar(aes(x = TMUs, y = Shader, color = Manufacturer), stat = "identity", binw
```

```
## Warning: Ignoring unknown parameters: binwidth
```

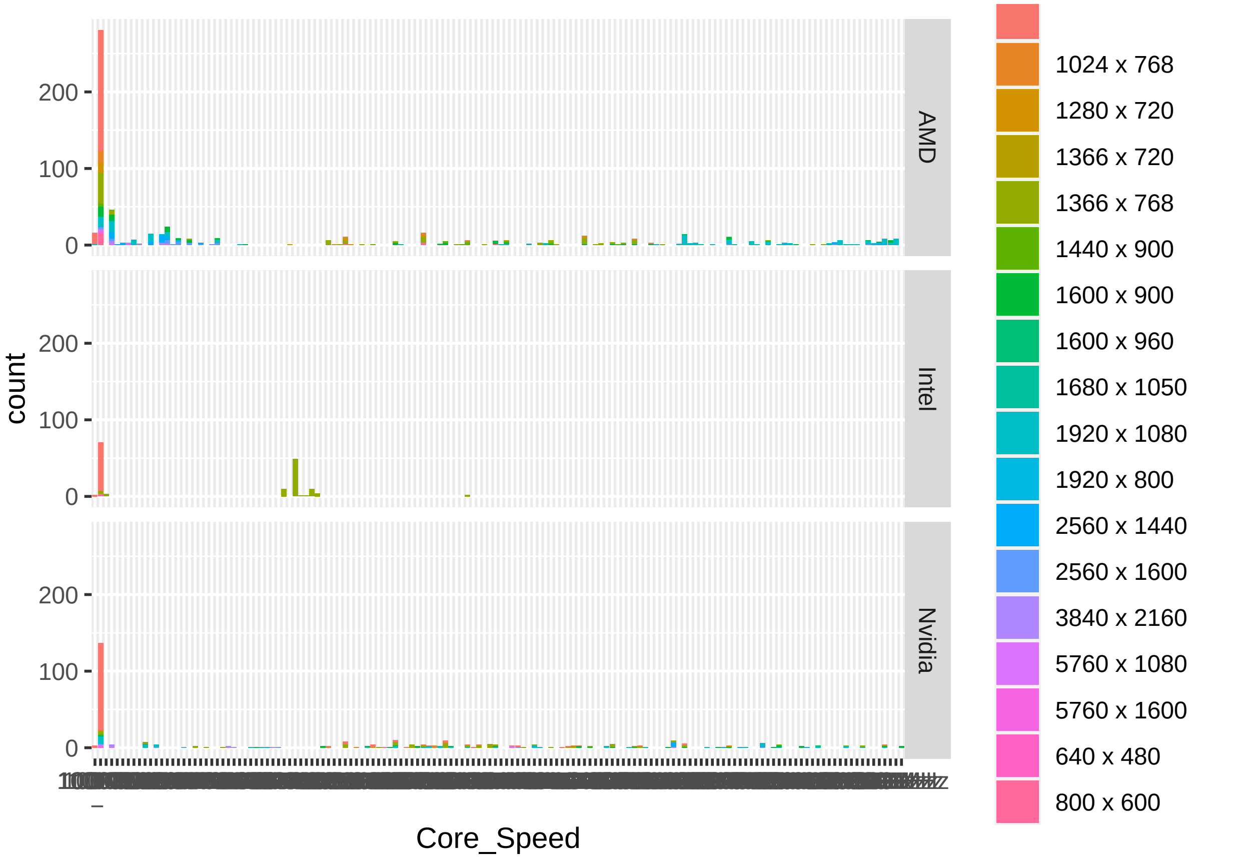
```
## Warning: Removed 331 rows containing missing values (position_stack).
```



Building off of my previous visualization involving shaders, I decided to the x axis be the TMUs since the texture mapping unit(TMU) deals with 3d graphical models and shaders add the rendering to the texture mapping. From this graph its estimated that TMUs that are around the 60-70 mark has the highest amount of shaders. To make the graph more interesting I grouped the manufacturers to see which company had the most shaders in their graphic cards. It seems that AMD has GPUs with more shader capacity and in a couple of columns Intel seems to have more shaders than Nvidia. Also like the previous visualization, I thought that the higher the TMU the bigger the shader, I was wrong in this case again, not sure why this could be? It could be possibly that there isn't really a correlation between TMUs and Shaders.

```
ggplot(data = allgpu)+geom_histogram(aes( x = Core_Speed, fill = Best_Resolution ),stat = "count")+face
```

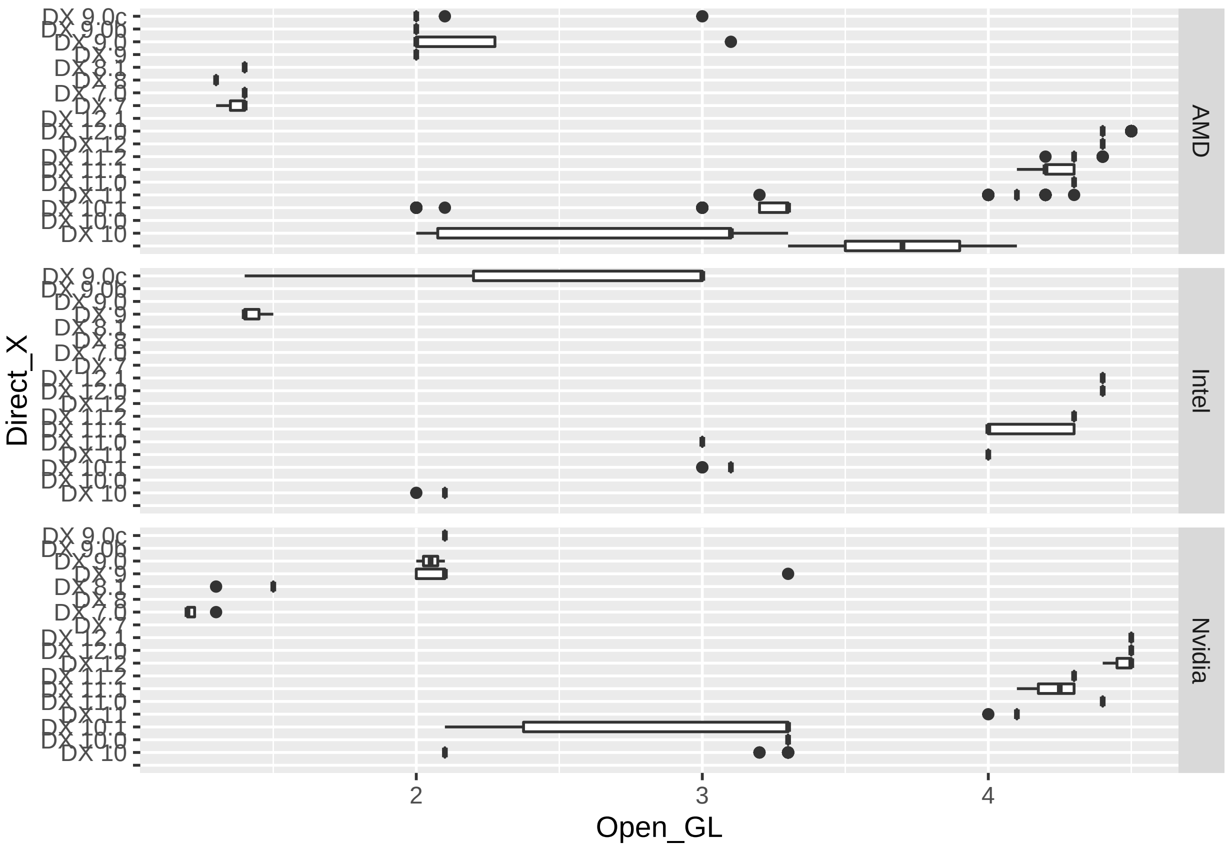
```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



With this visualization, I wanted to break down the manufactures into a facet grid and see which company has the most amount of core speed for their GPUs and what resolution works with those core speed. It seems like AMD has core speeds both high and low that can run at a resolution at around 1920 x 1080, but at a higher core speed it seems like running a resolution of 1920 x 1080 seem to be more common. Interestingly enough I would expect to see more 3840 x 2160 resolutions at the higher core speeds, but it doesn't seem like there's many. It could probably be the data set, the data was from four years age and 4k resolutions weren't really that common back then. Nvidia despite not having the largest amount of GPUs in the data set, has way more GPUs with higher core speed than intel, again intel is a more CPU oriented company so GPUs aren't really their main focus.

```
ggplot(data = allgpu)+geom_boxplot(aes(x = Open_GL, y = Direct_X))+facet_grid(allgpu$Manufacturer)
```

```
## Warning: Removed 28 rows containing non-finite values (stat_boxplot).
```



Here in this visualization I created a box plot to see which Manufacturer has GPUs with certain open_gl versions that has a version of direct x support. In the box plot graph it seems like Nvidia and AMD seem to be similar with both companies having GPUs have direct X 10 support with around the same open gl versions. However it seems that Nvidia has a bit more GPUs in the direct X 11 and direct x 12 version by just a bit. There are a couple of dots on all of the manufacturers grid, I am unsure what this means, it could mean that there is at least one GPU for the direct x version and open gl version.