What Makes a Good Graphics Card?

My client in this case study represents a desktop graphics card producing company that wants to know what specs are most important in a graphics card, and which ones can make the most practical impact. We will be looking at factors such as memory size, memory bus width, and other specs to compare them with price and benchmark scores to see which factors are the most practically effective to make a graphics card better. I will be using data on Kaggle since this data doesn't encroach on the privacy of any of the companies and uses information that is widely available to the public. My audience includes my stakeholders, who we assume to be a new graphics card producing company trying to figure out what specs to put into their new gaming-focused graphics cards to make the most cost effective card to compete with other popular companies such as Nvidia, AMD, and Intel.

I will compile each dataset into one dataset to effectively measure variables and create compelling visuals that show which variables are most effective to performance with cost of production in mind. I found these sources from Kaggle, one of them being "NVIDIA & AMD GPUs Full Specs" and the other being "GPU Benchmarks Compilation", with both of these datasets being published by the user, 'TESLA, INC'. After looking through the data I decided that I would use SQL to properly sort the data since there are a significant number of variables to go through, and use Python to analyze and create visualizations of the data.

I started on SQL where I found that many of the important data points in the Benchmarks Compilation table were either NULL or irrelevant to my research, such as price being null and device being anything besides desktop since my stakeholders are a desktop graphics card company. I also used this opportunity to remove other NULL variables including from the pricePerformance column.

```sql
CREATE TABLE GPU_Data.GPU_Benchmarks_Desktop AS

SELECT * FROM GPU_Data.GPU_Benchmarks

WHERE category = "Desktop" AND price IS NOT NULL AND powerPerformance IS NOT NULL;
```

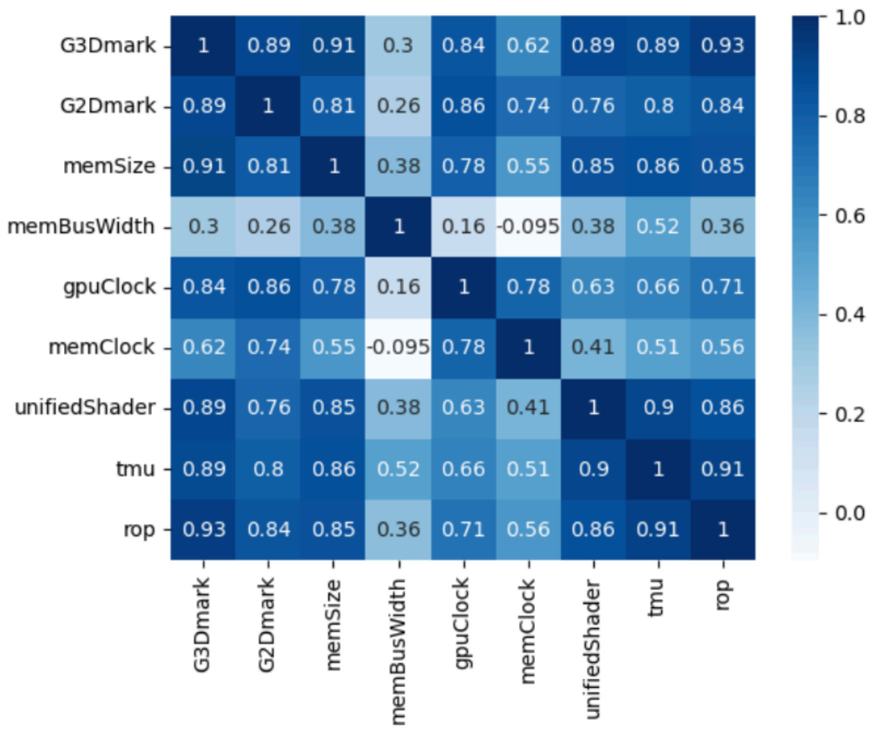
Once I corrected and sorted the data to fit my needs, I needed to join this table with the Full Specs table.

```sql
SELECT
 *
  FROM `nvidia-specs-data.GPU_Data.GPU_Benchmarks_Desktop_Complete`
   INNER JOIN `nvidia-specs-data.GPU_Data.GPU_Specs`
    ON gpuName = productName;
```

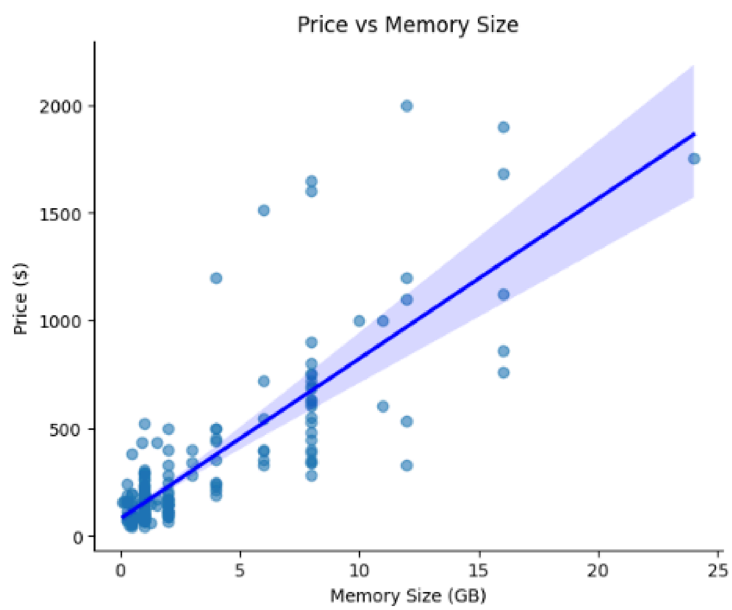
Once I finished sorting and joining the two tables together using SQL, I could begin analyzing the data using Python. I chose Python for this project since I will need quality plots and there are several open source packages available online that can be used to make quality plots using Python.

The first package I used was pandas, an open source tool that is mainly used to read in the CSV file I created earlier using SQL, into a local dataset in the python notebook that I could work with. I read it in to make sure that it read in properly and everything is correct, as well as using the .info command to check that all the variables are there and determine which ones are measurable and which ones are categorized.

I then wanted to make a correlation matrix to find which measurable variables correlate most to a graphics cards performance. I've used correlation matrixes before, so I know how they work and how to read them properly. I started by using a template provided by geeksforgeeks.org, which I modified to use my dataset rather than the provided one, in which I ended with a correlation matrix showing that the variables with the most correlation to performance were Memory Size (Gigabytes), Clocking Speed, Unified Shading Units, Texture Mapping Units (tmu's), and Render Output Units (rop).



Now that I understood what variables I wanted to look at to compare to G3Dmark score, I wanted to find which Memory Size would be the most price efficient since it has one of the highest correlations with G3Dmark score and is generally much less varied than the other variables. To do this I decided to use Seaborn's lmplot since it can properly show a line graph but still represent each specific data point, which I believe will allow me to properly interpret the data even if both variables are completely linear with eachother.



As we can see from the data, the most common data points which contribute to high G3Dmark scores tend to be in the 8GB, 12GB, and 16GB. After looking closely, we can see that despite offering more, 12GB cards seem to have the same performance as a 16GB card despite being much cheaper. The 8GB cards also seem to perform well given their price point, so we'll be using 8GB and 12GB in our next plot which will help us determine what GPU Clocking Speed best fits in terms of price to performance for our aforementioned Memory Sizes.

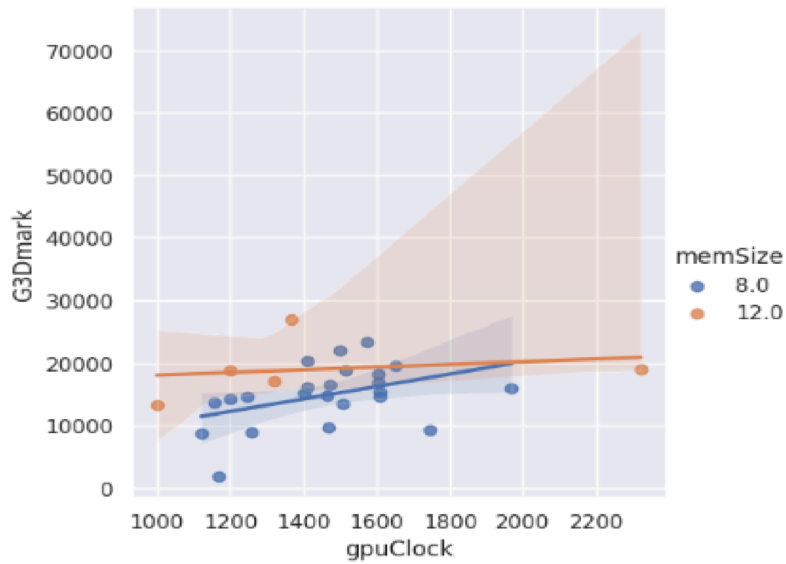
To make our dataframes that only contain 8gb and 12gb cards individually, we need to use what’s called a mask which “masks” our data only showing us the relevant datapoints.

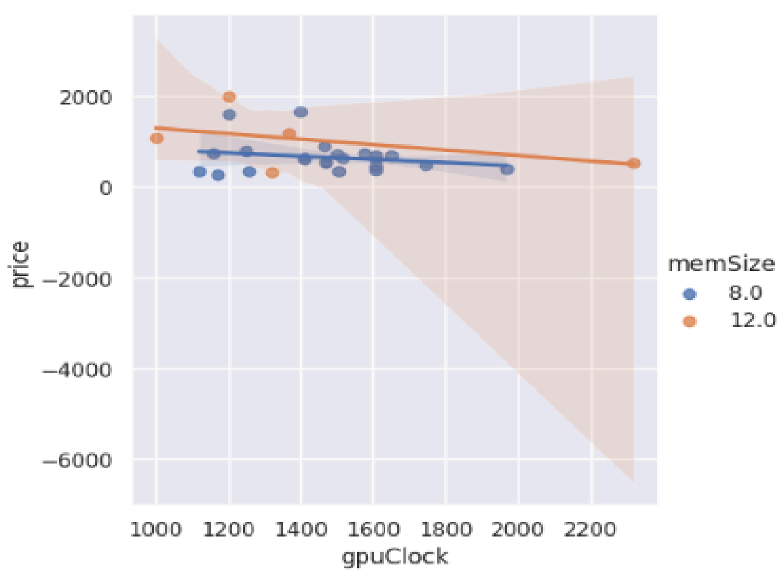
```
eight_gb = (GPU_Data["memSize"] == 8)
GPU_eight_gb = GPU_Data[eight_gb]
print(sum(GPU_eight_gb['memSize'] == 8) == len(GPU_eight_gb['memSize']))

twelve_gb = (GPU_Data["memSize"] == 12)
GPU_twelve_gb = GPU_Data[twelve_gb]
print(sum(GPU_twelve_gb['memSize'] == 12) == len(GPU_twelve_gb['memSize']))
```

```
True
True
```

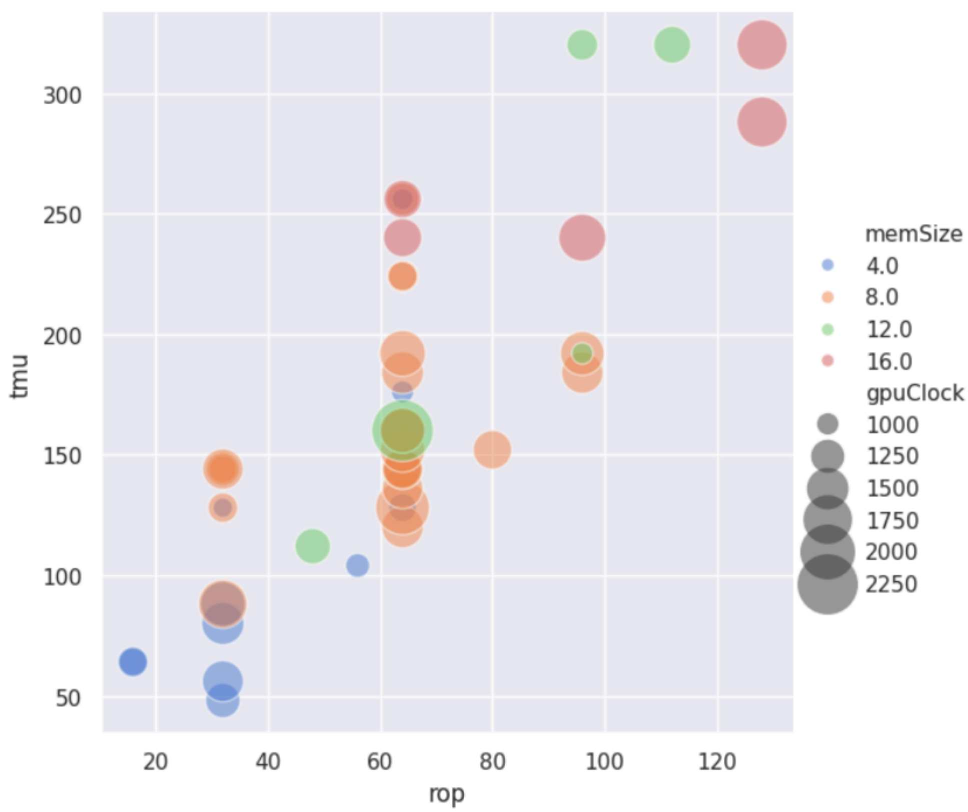
As we can see from our test, our mask worked properly as the number of datapoints that are equal to each respective memory size, are equal to the total number of datapoints in each masked dataframe. Now when I graphed the results for these two variables compared to GPU Clocking Speeds, the results were rather confusing.





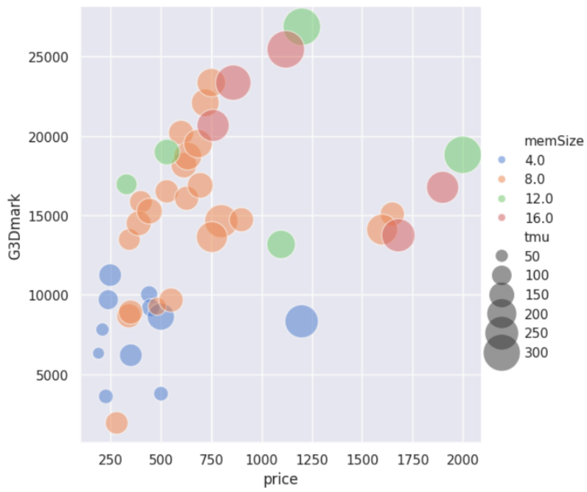
These results weren't what I would've expected, as the price of the card seems to decrease as the GPU Clocking Speed increases, which would imply that clocking speed is negatively correlated to it. But the cards performance increases with an increase in clocking speed, which would mean that cheaper cards give better performance, which we obviously know is false. I believe this may be because there is not sufficient enough data to tell what the trend direction is, but judging from the linear trend we will use what we can judge from the line of best fit for clocking speed being 1800 for a lower end card, and 2200 for a higher end card.

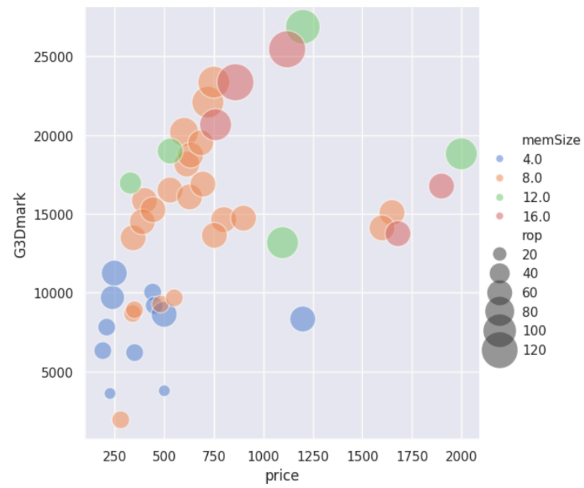
The only variables we have left to determine are the number of Unified Shaders, number of Texture Mapping Units, and the number of Render Output Units. Between those three I wanted to start with number of Texture Mapping Units and the number of Render Output Units since both of these variables have a higher correlation with gpuClock. To do this, I made a new masked dataset that has two new Memory sizes including 4gb cards and 16gb cards, so it will be easier for me to spot a trend between memory size and number of Texture Mapping Units/number of Render Output Units. A scatterplot also allowed me to put a significant number of variables compared to other plots, since we can not only color the points to match memory size, but also have the points be varied in size based on their respective clocking speed.



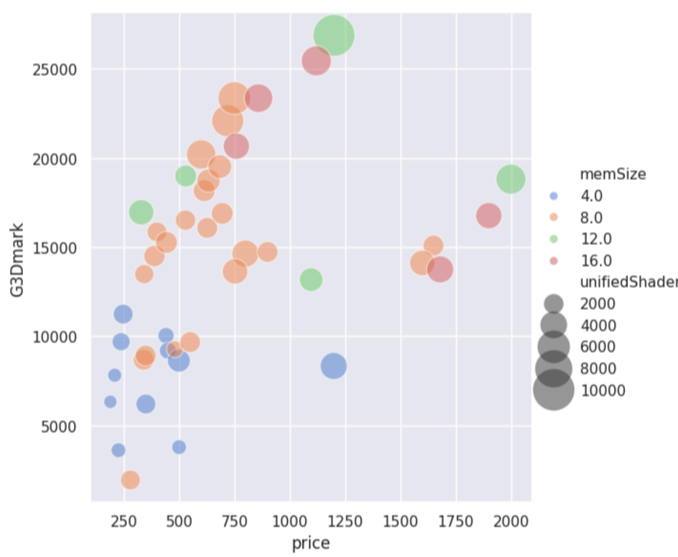
As we can see from the plot, the number of Render Output Units has a positive linear correlation with the number of Texture Mapping Units, around 12 Render Output Units per 30 Texture Mapping Units. We also see that gpu clocking speed doesn't have a strong enough correlation to either units to warrant further consideration, however, memory size does seem to have a positive linear relationship with the two variables. Something I noticed specifically was that despite being a lower memory size, 12gb was nearly effectively able to cover any range of units from 8gb cards all the way up to 16gb cards, meaning it may be the sweet spot of memory size for modern graphics cards in the gaming industry.

Something I wanted to see was how expensive Texture Mapping Units and Render Output Units were, and also how they scored in G3Dmark to determine just how many of each unit we would want in the ideal card. To do this I used the same plot but replaced memory size and clocking speed with price and G3Dmark score respectively.





As we can see from the plots, both tmu's and rop’s have a positive linear relationship with price and G3Dmark score, which means we’ll have to figure out what would be a good quantity of each variable. Since we decided to work with 8gb and 12gb cards, we will infer that the best values for each card would be around 150 tmu's and 60 rop’s for our 8gb card, and 175 tmu's and 70 rop’s for our 12gb card. Which leaves us with one more variable to find, which is unified shader units. To find this variable, I will make a scatterplot similar to what we did with the texture mapping units and render output units.



We can see from the plot that anywhere from 2500-3250 Unified Shader Units seems to be the sweet spot in terms of cost to performance, so from that we will conclude that our 8gb card should have 2500 unified shader units and our 12gb card should have 3250 Unified Shader Units.

All this data gives us the specs for our 8gb (low end) card and our 12gb (mid-high end) card. Our low end card has 8gb of GDDR6 memory, a base clocking speed of 1800, 150 Texture Mapping Units, 60 Render Output Units, and 2500 Unified Shader Units. Our mid-high end card on the other hand will have 12gb of GDDR6 memory, a base clocking speed of 2200, 175 Texture Mapping Units, 70 Render Output Units, and 3250 Unified Shader Units.

There were several variables that we went over in this case study, and many variables that weren't even in our dataset such as software, drivers, and many other properties that could change how a graphics card performs. However, as far as hardware variables go there are many different pieces that all fit together that determine how the card performs. But, no card is the same as another, and there are nearly endless ways that a card could be built to make it perform better or worse in specific scenarios.

To my stakeholder, I would suggest to use the specs stated above as a guideline for what a card should have on average for each given spec, with the 8gb card costing around $250 and the 12gb card costing around $325. However, many graphics card producers have turned toward making their cards have higher spec values over others to change its performance. For example, AMD makes cards with higher clock speeds which is generally good for all instances except for tasks involving compiling multiple textures and shaders. Nvidia on the other hand, prioritizes unified shading units and texture mapping units for their "RTX" effect, making their cards preferable for high level gaming and graphical design purposes.

So, there are general variables that all graphics card producing companies use. But, if my stakeholders want to produce a card that is well regarded by the customer population, it needs to have something special to it that other cards don't have. For example, their cards could have higher memory size for and render output units, making them a powerhouse for model creation and prediction.

References

NVIDIA & AMD GPUs Full Specs💠. (2024, August 27). Kaggle.

https://www.kaggle.com/datasets/alanjo/graphics-card-full-specs

💥*GPU Benchmarks Compilation*. (2022, December 13). Kaggle.

https://www.kaggle.com/datasets/alanjo/gpu-benchmarks

GeeksforGeeks. (2025, January 2). *Create a correlation Matrix using Python*. GeeksforGeeks.

https://www.geeksforgeeks.org/create-a-correlation-matrix-using-python/

Waskom, M. L., (2021). seaborn: statistical data visualization. Journal of Open Source Software, 6(60), 3021,

https://doi.org/10.21105/joss.03021.