Week 2

From last week I had a little change of heart from my initial feline-themed dataset and have to a more feasible idea. This was mainly from the workshop this week we were shown a few more examples of ways to visualise the data. As fun as cats are, I couldn’t seem to be able draft an interesting display from the data given. There were definitely some limitations in the number of fields that were available.

Back to the drawing board… or in this case **Kaggle**! I found a large dataset with over three thousand different PC graphics cards with many different fields to draw out information. So much so that I was really drawn towards the idea of using a **parallel coordinate** **graph.** Where each vertical measure being in a different unit or scale it would be ideal for this particular csv file.

Though, this is not without issues. The data is going to need a LOT of cleaning when being applied to my code. There are many fields that are missing or include the units along with the values (watts, Ghz, Gb, etc). I think this will have to be my main challenge for next week. As of APIs I haven’t really had too much of a look currently but is something to consider for next week.

Week 3

This week within the workshop I began to load in the CSV file containing all the information about the graphics cards. Since there were a lot of columns (20+) I decided to strip out all that I deemed unnecessary for the visualisation while keeping enough in for my code to clean. Fields that has barely information or very similar information was simply deleted with the help of Office Excel.

When all of this information was loaded into my JavaScript, I sorted each column into their own separate array using the ‘get.Column’ command. I then had to then tackle the cleaning. I created a small function that loops through each of these separate arrays, takes off a number of characters from the end of the string (removing the units) and then is stored into another array just for ‘clean’ data. Since these arrays are only numerical data, I can then find the minimum and maximum values for each column – this will be used for each vertical scale within the graph.

Though, I’m not without issues at this point. For some reason one of the columns that is loaded from the CSV file doesn’t seen to want to separate each part of data – even though there are commas in place to do so. Looking through this there are some other special characters that have been used such as: ‘+’. This may be affecting the CSV file in a weird way that I’m unaware of, so further help is needed on this.   
  
Another issue that I faced was that I haven’t managed to clean the spaces in the array that are empty. I’m torn between going through and removing all rows that have any spaces or allowing the program to ‘skip’ over these values and display the next available field. This is influencing the minimum values also, as “null” displays as zero, it won’t show the true minimum number for each column. At this stage I’m not too sure how to go about this. I would need to make sure that each of the column sync up correctly if I do revmove the blank spaces, so I’m thinking I would need to put all the clean data back into one array and go from there? I will get help next week if I can’t figure this out.