

Project 2: Final Report

My process for creating the visualisation

After deciding on starting again and working on a fresh topic I had to decide what it would be, I spent the first week of my mid-semester break analysing potential topics and datasets but no matter what the potential idea may have been, I noticed a common theme about the most interesting ideas. They were all based around financial topics, I had looked at crypto currencies & analysed stock market trends. I decided I wanted further analyse the stock market in some capacity. This was due to the more traditional understanding of the concepts that most people have meaning would be able to skip the introduction & explanation of cryptocurrencies to those who weren't aware of their increasing presence in online finance.

Since I need a new dataset I had to start the search, companies tended to not provide their stock history so an investigation on a single company or a comparison between 2 related companies would prove to be extremely difficult. However, I could check if there were any organisations that keep documentation on the stock market as a whole and utilise this in some analysis tool for the entire market. This lead me to the NASDAQ website which held listings for the market and near live updates. I thought that there may be hope for finding particular company data for a visualisation tool yet however I stumbled upon something even better. After reaching the company list page I was presented with an option to download several csv files, one of them being the entire NASDAQ listings.

The NASDAQ csv files would be unparalleled in accuracy due to the nature of their information collecting, the stock market is a precise and exact environment where specificity is required. This and the ruthless tracking of values within companies would ensure that the data presented would be suitable for even the highest level of analysis as long as it is not presented as the most current and up to date dataset. This meant I could trust that the csv would be trustworthy for my project.

Looking through the csv ended up being what inspired my project idea because there were several fields that were contained in the csv, each individual element had a label to identify it in the stock market alongside its name, stock value & market capitalisation. While the individual stock price does not indicate the value of a company due to the presence of varying share amounts. However the market capitalisation drew my interest. All these different companies can be valued at absurdly large amounts beyond even those of the richest people. This lead me to question how much these values are in relation to more tangible concepts.

People love talking about these extremely large figures however numbers are an abstract concept, unless we contextualised them with a physical amount it's hard to convey any tangible meaning behind the quantitative figures. By taking their values in the form of market capitalisation and measuring their proportionality. This will show their size through association; however, this can only be done through the construction of a visualisation that shows size. Many visualisations that show proportionality are not very good for showing information, such examples of this include the infamous pie chart. A pie chart will show proportionality but an approximate measure for the elements present is very difficult and the chart is unable to hold many elements as to make most of the elements unreadable.

One method of visualisation that is not commonly employed is the Treemap, by drawing boxes of size within a global bound these elements are automatically drawn in proportion to each other highlighting the main concept of my visualisation project. Although this will be the visualisation method I end up choosing, a simple Treemap does not suit the project requirements of a dynamic visualisation. This idea was already considered when I first encountered the NASDAQ csv. Due to every company on the NASDAQ being listed I had 3000 elements which due to the sheer number of companies listed and the magnitudes of difference between their market capitalisations it would be impossible to view all information on a single screen. In order to solve both of these issues an option to filter the data would be required. An immediate system would be to filter based on value as to see grouped data of smaller sizes without it being overshadowed by the larger ones. If we simply remove all items that don't fit the criteria the same scaling issues will exist, however if we redraw the visualisation every time it is filtered we can scale everything to fit.

Now that I knew what I wanted to do it was a matter of planning how I was going to go about this, the two top ideas were to make this as a Java app in Processing or as a web app in HTML & Javascript. I decided to go with a web app since I will be able to host this project online if I ever decide to show it off as a personal project, while this project serves a purpose with this task I will be able to also use it to display personal experience. This also ties in with preparing me for Web Development in 2nd year.

Since a web app will be created the usage of visualisation libraries will make this project much easier than manually creating all the required functions which would be far out of the bounds of my personal abilities. After a quick search online, I found that there are two major visualisation libraries and they are D3 & Google Charts. Because of the in-depth guides provided by Google themselves on the functions and creation of visualisations would prove to get me started up as soon as possible. Another useful function is to turn an array of data straight into the data table format that the Google API uses. The reason an array is used is during testing the time and resources required to host a csv or JSON for testing would add a further constrain on time that could be spent on creating a usable product, while this will increase load times they are not too long and it can always be changed later when I use it as a personal project.

By creating a separate .js file with a function to create the data table, the clutter that comes with having a 3000 element array can be kept separate from the rest of the code. One of the fields that the data table takes for the Treemap is a colour scale, by feeding in one column of data to the Treemap you can set the colours of the cells. I decided to use a logarithmic scale of the market capitalisations as to have an even distribution of colour, otherwise it will be based on proportion and due to the exponential increase in value only the very smallest values will be at the bottom of the scale.

With the html file the data table is initialised and using the Google API the Treemap is drawn, this on its own creates my visualisation and presents every element. While I now have something to show it is only static and adding a dimension of interactivity is the next step forward. The two methods that Treemaps can employ to add a dynamic dimension to them are tooltips & data filtering. While tooltips are specifically implemented for Treemaps the filtering option is not directly supported, it still works with any Google chart but it required a little toying with to get to work.

a1724710

In order to ensure the data could be filtered instead of calling a function to create the Treemap a function will create a Dashboard that will draw a filter and Treemap as part of a single setup. By filtering the scale that was set in the data table the user will be able to select what data they want to view and will improve the readability. The filter being used is a sliding range filter, by dragging two pointer heads across the range a custom range set with user defined start and end points allows as many or as few elements as they desire. It also allows users to view the lower end of the dataset without obstruction by the elements that are orders of magnitudes larger.

With the filter now giving the correct ranges for the data collected its functionality needs to be connected to the visualisation itself, this is done by binding them. Once they are binded it can be set to redraw the visualisation each time the filter criteria are changed. This means the visualisation dynamically updates thus expanding itself upon a traditional static visualisation which is one of the project specifications set.

Since filtering is not the only dynamic action that is set in the requirements elicitation, the final needs to be implemented. This is the mouse over tooltip which cannot be implemented before the dashboard has been due to compatibility issues. However, using the standard tooltip in the Google API and modifying the function to display the correct data would allow the user to access all necessary information presented in the Datatable.

This was the process involved in the creation of my visualisation, this process was aligned with the goals I had determined during the brainstorming part of my project. These goals were part of my original intentions with the project and ideas that were brought up throughout my feedback from my peers.

GOALS:

Find a dataset for market capitalisation that is close to the time of this project's creation

By downloading the NASDAQ database I had relatively up to date information that I required for visualisation

Find data for other points that can be cross referenced with previous data for proportionality

A similar process was undertaken on the US & AU treasuries for other usable information

Create a Treemap in js that can visually represent my data

Using the tutorial provided by Google, implementing a Treemap that can visualise my data was a trivial task.

Develop a system of scale to accurately divide the dataset into informative sections for user analysis

Using the logarithmic scale for even distribution of colour on the Treemap this could also be used to bind the range filter into smaller groups of data, however due to reported bugs in the Google API the increment on the filter was not able to go below 1.0 and so a rudimentary scale had to be implemented. This scale happened to be $10^{\log(\text{value})}$, this is equivalent to having an increment of 0.1 had the filter worked as intended.

Make my Treemap dynamic by having a filterable dataset

By using the scale to linearly distribute the data points this allows a filtering system to group the data in a reasonable scale. This prevents certain large or small outliers in the dataset to exist so far along the scale to create stretches of redundant range.

Have my visualisation draw from a data set that is not otherwise visualised

By cross referencing Treasury & Stock market data which is not commonly referenced alongside a Treemap which is also a not commonly implemented visualisation method creates a unique experience that encourages an alternative path of learning.

Have my js visualisation able to be interacted with in real time by a user

By using a Google Dashboard to house the web app and binding the range filter and the Treemap, interacting with the filter redraws the Treemap to suit elements that fulfill the criteria set in real time.

Make my visualisation scalable to enhance the readability

The binding of the Treemap inside of the Dashboard redraws the visualisation everytime the user updates their filter selection, this recalculates the proportion of the elements and resizes them to fit the users screen.

How I worked with my Stretch Goals

The goals I had set for this project were required to reach a usable product that fulfilled the requirements set by the task. By completing these I had a product I could show off on its own but the progression of this app doesn't stop there. These stretch goals were designed to be quality of life improvements that didn't contribute to the completion of the project but instead created a higher quality project to display.

Minimise code clutter that comes with the large datatable using JSON

This stretch goal was set with the best intentions in mind however it was not completed. The reasoning behind this one was to make the values inside the data table more readable to people analysing the code since every attribute is labelled inside of the element, this would also speed up the drawing process of the Treemap. While this is a significant improvement the requirement of hosting the JSON to be able to test it since browser won't load them in locally provided too much of a hassle to set up every time the project was worked on caused me to give this goal a skip. If this project were to be improved, due to the nature of having to host it to become viewable by external parties would make this a suitable improvement.

Create a mouse over tooltip to present the provided information for each data point

The inbuilt functionality in the API made creating a tooltip a task that was achievable in an instant, however it would only display the labels for each data point which would not suffice in being informative. By altering the tooltip function to draw information from other attributes in the data table and present them in the tooltip to summarise the highlighted box. This is especially useful for smaller elements that have been drawn as presenting all the information in them can be impossible, this also ensures that despite the crowding of elements if lenient filter criteria are used, a learning experience can always be provided. Due to it being an easier goal to implement and the fact that displaying all the attributes inside each cell would prove to detract from the experience this became the first stretch goal to be completed. It was even completed before some of the final base goals and was present in the presentation of the first cut.

Group data by additional categories to extend filtration

By including more categories to the data while it would increase the size of the data table, more options for filtering would be possible. This would be essential for further improving the app beyond the original requirements elicited in the planning stage. Fortunately, the NASDAQ csv file also included additional information about each company that I could include in the data table. While it was present here it had to be limited due to the lack of presence in the treasury data. Only one additional information field was included due to the fact that it had to be manually entered in the treasury data. This data was classed as the "Sector" of the company or what field (I.e. Finance or Technology etc). This had lead to this stretch goal being completed during the data collection phase.

Expand filtering options to multiple criteria

This goal is heavily reliant on the prior goal since additional element attributes must exist to filter the data by more attributes. Luckily the last stretch goal was completed and due to the presence of the range slider in the visualisation the addition of multiple filter objects was not out of the scope of my abilities. The only additional filter option I had considered was a drop-down menu that drew from the Sector attributes present in the data table. This would present a list of all sectors viewable in the data and each could be checked and unchecked to create a custom combination of selected values. Only these sectors would be displayed in the Treemap as to add further specificity to the visualisation. While this was able to be accomplished with the skills and knowledge I had accrued through the project, the constraints on time through spending a considerable time researching how to finish even the base goals lead me to cover more important issues and eventually skip this stretch goal. However, upon extension this would be the first point of improvement due to the nature of its implementation and the presence of a foundation to complete it.

Why I considered my Wishlist

While the base goals set a finishing point for the project and the stretch goals were chosen to extend the possible scope of the project, the wishlist acted as a purely hypothetical list of goals that would be completed to provide a top-notch product despite the fact they fall entirely out of the scope of the project in terms of time provided and personal ability. These goals are important to consider as to provide context to the project quality I am trying to achieve later in my programming career. They also set points of improvement I could implement on this visualisation later on when they become within the bounds of my skill and show an updated representation of my ability.

Add a method to summarise datapoints in an informative tooltip

By drawing a synopsis of the selected company or treasury figure that provided more information about the element would vastly enhance the learning experience that my project would provide. While it is not directly related to the information I was trying to convey if a user found a company they were not familiar with or were not certain about, they could be informed with a short paragraph. This amount of information could not be stored in the data table due to its size, the detriments in performance would not be worth it. However, if the data table could fetch information online and present it in the visualisation it would save space and time. Such an algorithm to summarise the companies and an implementation of the fetch of external data far extend beyond the scope of my abilities. While this is the case I am aware of its possibility and it has not been ruled out for future improvements.

Add a comparison tool between elements to further highlight proportionality

Since some elements are multiple magnitudes of order larger than others, the existence of both on a single filter page would practically erase the smaller one due to its minute proportionality. If an option to select multiple elements and have the visualisation draw comparison like quantifying the proportionality between them or comparing what sectors they are a part of. This option would also implement a search where you could select elements based on their label without having to manually select them. This would allow a level of analysis achievable for the user that extends far deeper than any other interactivity I have added or considered yet. This is also far outside my knowledge and would need a considerable level of research to be conducted before its implementation could even be considered.

What I learned

This project was by far the most invaluable piece of self-directed learning I had ever undertaken in university so far. While I had learned many skills throughout my other projects & subjects I had to learn a great deal in a short amount of time. This mainly involved learning web development from scratch as I had not used html in over 5 years and I had to work back from square one. This meant working with the basics and slowly implementing the Javascript over time. With my first mouse over tooltip I had to learn some CSS to implement a tooltip that followed the cursor. While this was present in my first cut I had to remove it due to the incompatibility with the Dashboard for filtering. So even though CSS did not play a major role in my final product I did do some substantial learning in this area. I also developed my skills in data visualisation far beyond what I learned in the first project. The procedures and skills I utilised throughout this task will aid me next semester as I undertake subjects based around web dev.

References & Resources

<http://www.nasdaq.com/screening/company-list.aspx> - NASDAQ csv source

<http://www.abs.gov.au/browse?opendocument&ref=topBar> - ABS data, works with the AU treasury

<https://www.treasury.gov/resource-center/faqs/Markets/Pages/national-debt.aspx> - US treasury data

<https://developers.google.com/chart/interactive/docs/gallery/treemap> - Google Treemap Info

<https://developers.google.com/chart/interactive/docs/gallery/controls> - Google Dashboard Info