H00389369, yb210

F21DV: Lab 3

1 Introduction

This report will serve as an accompaniment to html files containing the JavaScript code to solve lab 4 of the F21DV course. It will go through the explanation of how the code in my javascript files are structured, how my implementation satisfies the requirements and answers to the queries. This report assumes some prior knowledge of JavaScript and will focus on describing the use of the d3.js v7 library.

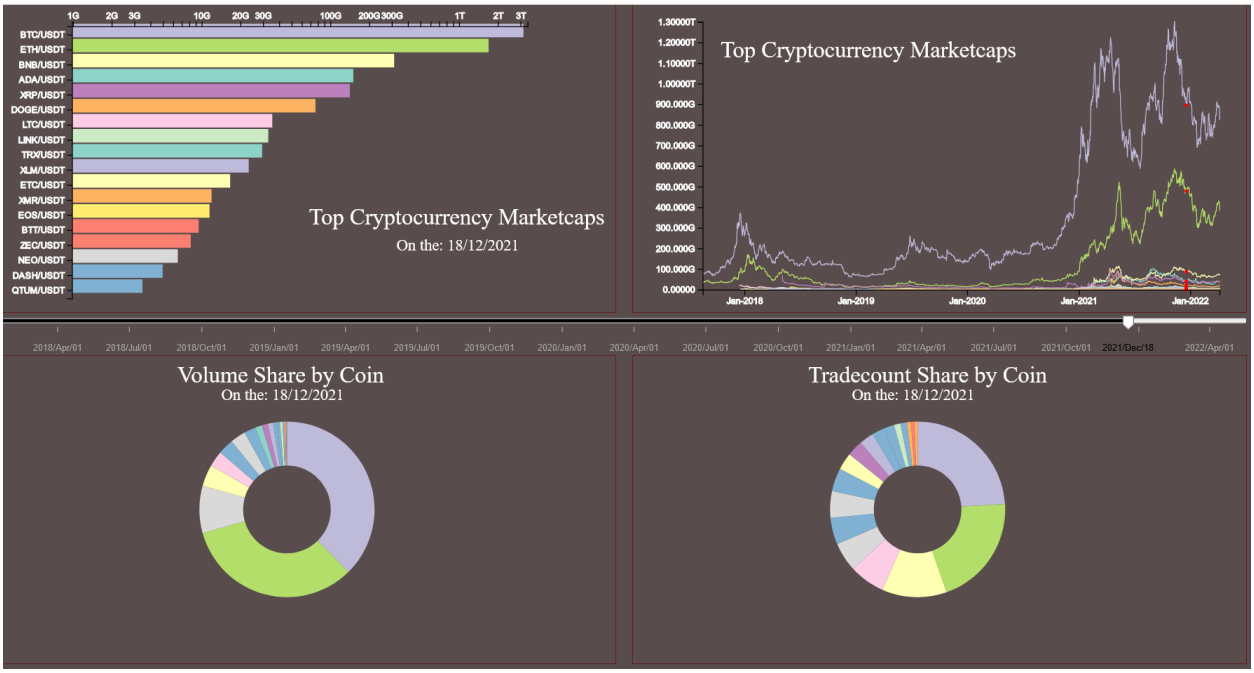
I demonstrated to Benjamin Kenwright on the 8th of April 2022.

A GitHub repository was created and showed to be helpful with code management. With the help of the desktop app for GitHub, I managed to smoothly operate on both my IDE of choice (WebStorm) and see the progress on my GitHub page. The repository is the following but may be set to private depending on when this link is clicked: https://github.com/YoussefBonnaire/DataVisualisation.

2 Visualization Story

I decided to use a public data set on the evolution of a select few crypto currencies from the Binance database of https://www.cryptodatadownload.com/. This decision was based on the ever-growing importance these digital currencies within everyday life.

The biggest and most known cryptocurrency is Bitcoin and this can be seen in the bar and line graph with it maintaining the largest market capitalization (market cap) throughout the data we have at our disposal from mid-2017 to April 2022. The interesting part we wish to highlight is that although it is clear Bitcoin is still the leading cryptocurrency, we can see in the donut charts showing the volume share by currency (indicating amount of the currency used in a day) and trade counts by currency (indicating the amount of trades for this currency) that bitcoin is slowing down compared to some of these up-and-coming crypto currencies in some metrics. Specifically, the now household name, Ethereum, which is keeping up with bitcoin in market cap has days where it's volume and trade count are higher than that of bitcoin (i.e. 18/12/2021).



3 Code Structure

The code is separated in 7 files, main.html, main.js, LoadDataLab4, Race.js, Line.js, Force_trades.js and Force_volume.js. Main.html serves to host the local server importing all other files and using the main.js in its body.

In the main.js, I have established the svg's each of the graphs, charts and the slider belong in. Furthermore, I also initialize useful constants such as the width and height of the svgs, a list of the coin pairings and the radius and arc of the donut charts. This file then calls on functions in the other files to accomplish the visualization.

The first 2 called being dataByDate() and dataByCoin(). They are hosted in the LoadData_Lab4.js file and load the data in grouping it in different ways for different usages in order not to have to call these each time we wish to change the date we are using. The data sets we have downloaded were all separated by coin, therefore, for ease, we combined all of these into one variable. Moreover, the data did not include market cap and we thus created our own csv file with the circulating supply of our selected coins in order to calculate it.

Following the data being loaded, we initialize the bar chart with the DrawChart() function from Race.js. It uses the data_byDate generated from the previously described function and adds the axes and the bars in a simple bar chart manner using a preset color scale dependent on the currency.

Then, our line graph is placed using the DrawLine() function in the Line.js file. It populates the line for each of the coins graphing the market cap against time also adding a circle to show the date currently shown by the other graphs.

The two pie charts work in the exact same way and initialized by the functions ShowTrades() and ShowVolume() in the files Force_trades.js and Force_volume.js respectively. The name of the files is due

to my initial intention to have the visualization be a force simulation, however, in the end I decided the pie charts worked better, the code for the force simulation is still present but commented out in the Force_volume.js file.

Finally, we also have a slider which updates information on all of the charts smoothly. It is initialized after the line chart in its own svg and changes to the slider call functions in each of the files loading charts. The UpdateBars(date) chooses the data for the date specified by the slider and smoothly transitions the scales axes and bars. The updateDot(date) function moves the circles indicating the date on the line chart and moves them to the new date. Due to weird visual glitches I decided to temporarily remove the old dots populating the chart with new dots. The updateTrades(date) and updateVolume(date) functions update the pie charts by using the new date and transitioning the each of the arcs to their new location. For ease of understanding, I separated the updating in three functions, first changing the path of the existing slices, then adding the additional paths transitioning them in and finally removing extra paths.

4 Conclusion

With this report I have discussed the structure of my code and the capabilities of my application which complete the requirements of this course work with the files accompanying this report analyzing the data to help visualize the evolution of crypto currency market caps.