Milestone 3

Group 6: S&P 500 Visualization

<table>
<thead>
<tr>
<th>Name</th>
<th>CSID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emily Lian</td>
<td>l6q1b</td>
</tr>
<tr>
<td>Tianhang Cui</td>
<td>y7r1b</td>
</tr>
<tr>
<td>Yuchen Liu</td>
<td>d6h0b</td>
</tr>
</tbody>
</table>

Overview

S&P 500 (Standard and Poor’s 500) is a stock market index that tracks 500 publicly traded domestic companies in the United State. It is considered by many investors to be the best overall measurement of American stock market performance. Many investors need to keep track of the changes of the S&P 500 in detail and make investment decisions based on it. However, they need a tool to help them know better about these 500 most representative companies and see how the past and current events, such as Covid-19, affected the stock market. If the investor could well understand this information, he can make a better investment decision in the future. To address this challenge, we propose building a data visualization that allows investors to visually explore a dataset about the companies in the S&P 500 index. Our app will show the distribution of industries and companies contributing to the S&P 500 index (and this index is representative of the whole US stock market) and allow users to explore different aspects of this data by reordering on different variables (such as capitalization, stock price, etc) to see how the market has changed in recent years. The intended audience is investors.
Data

Data preprocessing

We realized that the data obtained from datasets have different companies. Thus, we looked up the news and timeline, then decided to use the companies listed as of the last day of our visualization range (2021-01-29). Therefore, we need to use yahoo finance API from Pandas to fetch and trim new data. Moreover, we realized that the amount of data is too huge for visualization, it is posing performance issues towards our charts. Therefore, we decide to use javascript to transfer CSV data into JSON data, which has a faster lookup time. For detailed implementation, please refer to /data/436_preprocess.ipynb and /js/preprocess.js. Below are the general steps we took to preprocess the data.

1. We use google sheet to open `data/marketcap.csv`, entering the function
   
   `=GOOGLEFINANCE(symbol, "marketcap")` to obtain market cap for each company.
2. We use the d3 rollup function to get the market cap for each sector, then export to `data/industryMC.csv`.

3. We pass `data/data/*.csv`, `data/marketcap.csv` into the jupyter notebook `data/436_preprocess.ipynb`, with the help of pandas and yahoo finance, we can get `data/marketcap_preprocessed.csv` and `preprocessed_data.csv`. Detailed explanations are documented in the Jupyter notebook for reference.

4. We use preprocess.js to read `data/marketcap_preprocessed.csv` and `preprocessed_data.csv`, convert the data into JSON format in the browser console, then copy the JSON data into `data/companyData.json` and `sectorIndex.json`.

5. Lastly, we use `data/companyData.json`, `data/sectorIndex.json`, `data/industryMC.csv`, `data/marketcap_preprocessed.csv` and `SP500HistoricalData.csv` in main.js to perform data visualization.

**Wikipedia API**

1. "https://en.wikipedia.org/w/api.php?action=opensearch&format=json&origin=*&search = + company name is called to get the title of the Wikipedia page of the company
2. "https://en.wikipedia.org/w/api.php?format=json&origin=*&action=query&prop=extracts&exintro&explaintext&redirects=1&titles=" + title is used to get the first paragraph of wikipedia content for the given title.

**Goals and tasks**

Albert is an investor, he has some money and he is considering whether he should invest this money in a company on the stock market and which industry and which company is the best place to invest?

- **Discover the Trend of the stock prices in the American stock market in general**
  First, he may wonder how the overall American stock market performed by comparing the current with the past to know when is a good time to invest in the stock market. When Albert logs on to the app "S&P 500 Companies Visualization", he can simply get the answer by checking the line chart in our visualization and get the S&P 500 index changed over time to get a whole picture about the past and current stock market. The single S&P 500 is a good index to show the overall situation of the US stock market. (Abstract goal:

- **Identify the distribution of capitalization among all the industries**
To get more information, he can check the treemap to see the distribution of capitalization among all the industries (the S&P 500 index is representative of the total capitalization of the whole industry).

- **Compare the feature of capitalization among all the companies (in the selected industry if any)**
  Suppose that he wants to explore more and hopes to see which industry seems to have a better future, he can **select an industry on the treemap to filter out all other industries**. Within the selected industry, he can **see the distribution of the capitalization by the companies in the bubble chart**. At the same time, he will **see the line chart** showing an index which is the sum of stock prices from all the companies in the selected industry. In this way, he can tell the performance by checking whether the selected industry is doing better and better, or it is suffering a decline. He can order the companies according to their capitalization, stock price, or the first letter of the company name.

- **Identify and compare the trend of the stock price for the selected company**
  Assuming he has decided which industry he wants to enter, he may wonder which company is a good place to put his investment in. By **clicking a circle for a company on the bubble chart**, he can see the stock price for the selected company in recent years in the line chart. By **selecting multiple companies**, he can compare the history of stock prices among many companies. This can help Albert to see the trend of different stock prices, evaluating the risk in each stock, and estimating how each stock will be affected by the possible future events from the experience of the past. From above, we can see Albert may need the information in a wide range of time to see the trend of the development of an industry/company, but he may also need to check the information in a small range of the date to evaluate the effect of a specific event at that time.

- **Identify and compare the trend of the total stock price in customized investment baskets**
  Finally, Albert may have a few plans for investment, and he can make each of his plans to be a basket as an investor. He wants to see which combination of investments is more desirable by comparing the stock price trend for each basket. He can achieve this goal by dragging the bubble from the bubble chart to customize the basket and he can compare 2 baskets at one time on the line chart.

**Visualization design explanation and justification**

**BubblePlot View**

- **Overview**
  Each circle represents an individual company. Circle size encodes the market cap of the company. The color hue of the circles corresponds to the sector the companies
are in. Horizontal and vertical position on a common scale encodes market capitalization and percentage change in stock price. We are doing a redundant encoding of the market capitalization, as we want to improve the salience of this attribute, in the context of the stock market. When a circle is selected, the line chart will display the selected company's stock price. (We did not explicitly show the corresponding relationship between circle size and the value of market cap in the legend, this is because the user can directly tell the market cap from the x-axis due to redundant encoding, adding this information into legend seems to be unnecessary)

- **Zooming + Panning**

Because we are visualizing a maximum of 505 companies in the bubble chart. The screen real estate is quite limited. There is a lot of overlapping for the small companies. Thus we introduced the functionality of zooming and panning. With the help of the initial loading animation in the bubble chart, we bought extra time for the rendering of the bubbles and provided a hint to the user that zooming and panning are supported on the chart. After the initial transition is complete, the user can pan, scroll and double click to zoom into the chart to see the detailed positions of the small bubbles.

- **Animation**

Moreover, we want to provide the user with a sense of the scale of each sector, with regard to the whole SP500 market. The animation of the circles and axis provides the user with a sense of the price change by the vertical transition and a sense of the change in market cap by the horizontal transition. The animation when the user clicks on a treemap block to filter data provides a sense of the relative position of companies in one sector with regard to all the companies.

- **Customizable basket**

We added the functionality that allows dragging the bubble to a rectangular area. By using the basket. By using this new interaction, we have a new and innovative way to interact from the bubble plot to the line chart: the user can create their own basket of stocks (more on the 'Changes since proposal' part), and the line chart will display the combined stock price for the basket as the user-specified. For the intermediate box “basket”, which is for putting the dragged bubble (they will turn into a checkmark after drag), we would say the mark is the point type, and we use the 2D size to encode the
market capitalization for each company as the bubble chart, and we used the color hue channel to encode the sector for each company.

Treemap view

Each point mark represents a sector. Color hues of high saturation and variance are chosen to encode categorical attribute, sector. The area of rectangles reflects the market capitalization of S&P 500 companies in the given sector. Treemap captures relative sizes of data categories, allowing for the quick perception of the items that are large contributors. A treemap helps users easily compare the size of each rectangle as they are side by side.

Users can filter companies by sector by selecting the corresponding rectangle in the treemap. This change is shown in both the line chart and the bubble chart. Filtering by sector reduces the number of elements shown on the bubble chart and allows users to focus on the sector of interest. The line chart shows the combined stock price of the selected sector by aggregating and deriving new values from existing data (given that no company is selected).

LineChart View

- **line chart in general**

This view displays the S&P 500 index stock price over time as default and displays individual stock price if a company is selected in either the bar chart or the bubble chart. When a sector is selected in the treemap view, the line chart could display the sector index over time. Users can select the desired time interval and the view will be updated. The mark is the point that represents the stock price on a particular date, and the line connection between them. The channels are the vertical position on a common scale encodes the amount of the stock price, the horizontal position on a common scale encodes the dates, the color hue of the line represents the sectors for the target company/sector (‘white’ for no sector, which is only for sp500 index itself), and we used the shape (dashed/solid line) to distinguish whether the line shows a price for the basket or not.
We also used the multi-view visualizations: Overview/Detail with the same format, by adding an overview line chart below the detailed chart and allowing the interaction to move and drag a brush and show the selected time period. The user can drag the brush to change the time period they selected, and they can also see the whole chart by clicking the overview chart outside the brush. We also added the tooltips with a fuzzy position to show the exact price/index, this allows the user to compare the price of different stocks for a specific time with accurate numbers.

- **brush**

We added the brush function to allow users to adjust the time period. In terms of the data abstraction, we have the 2 quantitative attributes in the line chart(stock price and the time), so a line chart with a brush would be appropriate and help us to filter the data by time. In terms of task abstraction, users sometimes may need to summarize or compare the trend for the stock price in a long duration, but sometimes they want to focus on a short period to identify the effect of some event on a particular stock. Compared to other options to change time, the brush is more flexible and intuitive to satisfy a user's multiple tasks for the line chart. When the user is moving the brush, the bubbles will change their position dynamically through animation.

- **Search**

Sometimes the user’s task is to summarize the trend or loop up the price for a particular stock, it may take them some time to locate the stock by using the filter, so the search option can help them to quickly find the data they need. The user only needs to input the partial name or some letters in the stock name, and the search bat can provide the autocomplete suggestion to make the process more efficient.

**How to interact with the SP500 visualization and how they are linked:**

First, the investor may wonder how the overall American stock market performed by comparing the current with the past to know when is a good time to invest in the stock market. When they log on to the app "S&P 500 Companies Visualization", they can simply get the answer by **checking the line chart** in our visualization and get the S&P 500 index changed over time to get a whole picture about the past and current stock market. The single S&P 500 is a good index to show the overall situation of the US stock market. To get more information, he can check the treemap
to see the distribution of capitalization among all the industries (the S&P 500 index is representative of the total capitalization of the whole industry).

The investor may decide to focus on a single sector, he can select the sector on the treemap to filter out all other industries in the bubble chart. When a sector is selected, the line chart shows the sector index which is the sum of stock prices from all the companies in the selected sector. (The screenshot shows
By clicking a circle for a company on the bubble chart, the investor can see the stock price for the selected company in recent years in the line chart. By selecting multiple companies, they can compare the history of stock prices among many companies.
To look closer at companies in the bubble chart, the investor can zoom in to the bubble chart view.
The investor can click on the reset view button to reset the scale of the bubble chart and deselect companies. The investor can drag company circles into the customize basket and analyze the combined stock price trend of companies.

The investor may need the information in a wide range of time to see the trend of the development of an industry/company, but he may also need to check the information in a small range of the date to evaluate the effect of a specific event at that time. This can be done by moving the time interval brush around in the line chart.
To investigate a stock that the investor had in mind, he can search the company name in the search bar. The stock price of the given company will be shown in the line chart and the company will be located in the bubble chart (given that the company is not filtered out).
Credits

Overall:

Data:
1. data/marketcap.csv
   https://datahub.io/core/s-and-p-500-companies
2. data/data/*.csv
3. data/SP500HistoricalData.csv
4. GoogleFinance API
5. YahooFinance API
6. Wikipedia API

BubbleChart:
1. Mainframe
   https://codesandbox.io/s/github/UBC-InfoVis/2021-436V-examples/tree/master/d3-interactive-scatter-plot
   Used to construct the whole frame of the bubble chart.
2. Zoom
   Allow the user to zoom and pan using the mouse.
   https://bl.ocks.org/puzzler10/63c0eff1756ca7cb62213932f9ef6825
   Put the zoom behavior on top of a rectangle listener, in order to separate from the drag basket, which shares the same SVG with the bubble chart.
   https://observablehq.com/@d3/programmatic-zoom
   Makes the initial zoom possible. Gives the user a hint that the bubble chart is zoomable.
3. Animation
   Makes the animation possible for bubbles, to show trends in a price change, market cap change, and sector's relative position in an overall view.
4. Dragging:
Line chart

1. Animation:
   
   https://observablehq.com/@jurestabuc/animated-line-chart

   It is used to show the line on the line chart to give the user an intuition about how the price changed with time. We changed it to make it fit our line and the animation will only play when the selections are changed.

2. Brush for the line chart
   

   It is used for selection on time and the user can move, expand, shrink the brush by dragging. We changed it to let it only show the line (no area) and apply the brush as default, so the user can easily notice the brush.

Treemap

https://www.d3-graph-gallery.com/treemap

The code block is used to create the static view of the treemap. Tooltips, Interactions, and filtering functions were built upon it.

Reflection

Describe how your project has developed from your initial proposal, through your first submission, to your final product.

- We started the proposal with three traditional views: treemap, bubble chart, line chart. After we initially rendered the data into views, we realized that the screen real estate
is quite limited. There is a lot of overlapping for the small companies in the bubble chart. Thus we introduced the functionality of zooming and panning.

- To further improve the innovativeness of the visualization, after many rounds of meetings and office hours, we decide to add the functionality that allows the dragging of the bubbles to a rectangular area (customizable basket) and visualizes the derived data in the line chart.
- Because we are visualizing a maximum of 505 companies in the bubble chart, a huge number of data, thus we added the search bar, as it facilitated the tasks that are related to locating a particular stock.
- After Milestone 2, we decide to add the live API call, to pull the introduction of each company on the fly, as we want the user to learn about the stories behind each company. For most of the time, we are focusing on fixing the bugs in interactions between views and user interaction with the whole system.

**How have your visualization goals changed?**

- Support the visualization for ‘Basket’
  We noticed that many investors may have the need to use baskets to evaluate their personal investment, so we included the goal of using this visualization to see the change prices of an investor’s customized basket by allowing dragging the bubble to a rectangular area.
- For dragging the bubble to the basket
  We used to want to drag the bubble from the bubble chart to the basket directly, but we realized that the user may still want to see the bubble’s position on the bubble chart, so we decided to copy the target bubble for dragging. However, this may confuse the user as they may think there are multiple bubbles in the same place, so we decided to change the dragged item to be a checkmark rather than a bubble to distinguish it from the original item.
- Support the search for specific stocks
  We realized that sometimes the user may need to do the filtering step by step to find their stock, instead, they are clear which stock they are looking for. Thus we added the search bar to allow the user to directly find the single stock they are looking for without filtering.
- Remove the year/month/day selection, and personalize the line chart interval
  We used to provide some time interval options(1w/1m/6m/1yr/All) for users to select and the user can move the line chart to the next unit of time, but it seems this will limit the range of stocks that the user can see in the detail view. (For example, the user may want to carefully see the trend in 2 months and they have to see 6 months without enough detail or go back and forth from 1 month to the next month.) Therefore, we added the brush (it is used to be an optional task in our original schedule) and this allows the user to fully control the dates, and it is intuitive enough to allow users to quickly learn how to use it.
- Unchangeable x, y-axis for the bubble chart
  This is optional in the original task, but after careful consideration, we decided to remove the functionality that allows the user to change the x, y axes of the bubble chart. For the y-axis, as we mentioned in the ‘preprocess data’ part, we cannot use the publicly available financial data to have an accurate percentage change for the market capitalization with date, so we decided to make the y-axis can only be the
price change unless we do have a way to get the accurate data later. For the x-axis, we searched for a lot of information about finance, and it turns out that the comparison of the magnitude of the stock price is not usually meaningful, and seldomly used by the investors.

- Add the tooltip about the overview information for each company
  We further realized that the user may want to know more than the stock price when they decide whether to invest in a company or not, and these companies are typically those companies they selected to show in the line chart (i.e. those stocks they are interested in). Therefore, we added the tooltip for each company on the line chart to show the overview information to let the user have a basic image of the target company.

- We added the tab that allows the user to switch line charts among different kinds of data (for SP500 index in total, for each sector, and each company) and more easily see the switch among different kinds of line charts.
  As we finished the design of the line chart, we realized that the user may not notice that the line chart automatically changed from showing the SP500 index to the company stock price, based on their selection, as the change of the title only is not obvious enough. Also, they may want to see the SP500 index or the sector total price even when they have selected companies. We gained the idea from the tab in the web browser to make it more clear when the tab changed and allow the user to switch back and forth.

How have your technical goals changed?

- We recognized that most of the financial APIs are not free of charge, especially for the retrieving of temporal data on the market cap. Thus, we decide to only use the latest value of the market capitalization for visualization as the market cap for each company does not vary much in the range of 3 years.

- As for the bubble chart, we initially plan to use plain transitions to support the animation of the bubbles. Because of the complexity of data and a large quantity of data, we decide to use zooming and panning on top of animated transitions.

- Another reason that we let the bubble be copied as a checkmark rather than directly moved the bubble away from the bubble plot is because we find it is difficult to implement a draggable element on a zoomable plot, so we decided to make a new element on another layer of the visualization to avoid the problem.

- Initially, we loaded data from a CSV file and the speed was too slow due to the large file size. We decided to parse data into JSON format and increased the data access speed.
How realistic was your original proposal in terms of what is technically possible in D3?

- We had a relatively realistic proposal in Milestone 1 and we have built more features on top of the initial plan.

Was there anything you wanted to implement that you ultimately couldn’t figure out how to do if so, then what workarounds did you employ, or did you abandon your original idea?

- We were planning to use an API to get the market cap of each company on the fly, but we couldn’t find a free API that supports this functionality. We also thought about the use of an API to get the daily stock price, however, free APIs allow only a small number of calls in a short period of time, as we have 505 companies, it is not possible to retrieve the new daily price for free. Therefore, we decided to substitute the financial API for an API to pull the info of each company on the fly.

If you were to make the project again from scratch (or any other interactive visualization), what would you do differently?

- We did not realize the amount of data for the 505 companies would cause performance issues. If we were to do the project again we would write a simple visualization first to test the effect of data size on rendering performance.

Project Management & Team assessment

Task: newly added tasks, tasks from M2

All the tasks are done.

<table>
<thead>
<tr>
<th>Task and Description</th>
<th>Estimate completion Date</th>
<th>Actual completion Date</th>
<th>Time Estimate</th>
<th>Actual time spent</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Come up with the idea for the visualization, break down the work, and make a schedule</td>
<td>March 7</td>
<td>March 7</td>
<td>5hr</td>
<td>5hr</td>
<td>All</td>
</tr>
<tr>
<td>Write the overview and goals and tasks for the M1 documentation</td>
<td>March 10</td>
<td>March 10</td>
<td>3hr</td>
<td>3hr</td>
<td>Tianhang</td>
</tr>
<tr>
<td>Write the Description of visualization &amp; sketch for the M1 documentation</td>
<td>March 10</td>
<td>March 10</td>
<td>3hr</td>
<td>3hr</td>
<td>Emily</td>
</tr>
<tr>
<td>Write the data, and Project Management &amp; Team Assessment part for M1 documentation</td>
<td>March 10</td>
<td>March 10</td>
<td>3hr</td>
<td>3hr</td>
<td>Yuchen</td>
</tr>
<tr>
<td>Preprocess the data using python, javascript, and excel.</td>
<td>March 15</td>
<td>March 14</td>
<td>5hr</td>
<td>8hr</td>
<td>Yuchen</td>
</tr>
<tr>
<td>Task</td>
<td>Start Date</td>
<td>End Date</td>
<td>Duration</td>
<td>Person(s)</td>
<td></td>
</tr>
<tr>
<td>---------------------------------------------------------------------</td>
<td>------------</td>
<td>----------</td>
<td>----------</td>
<td>-----------------</td>
<td></td>
</tr>
<tr>
<td>Get market cap data using google finance</td>
<td>NA</td>
<td>March 13</td>
<td>NA</td>
<td>Emily</td>
<td></td>
</tr>
<tr>
<td>Parse data from CSV into a JSON object to improve performance for stock data and derive historical data for sector index.</td>
<td>NA</td>
<td>March 15</td>
<td>NA</td>
<td>Emily</td>
<td></td>
</tr>
<tr>
<td>Discuss and construct the struct of index.html and project structure: main.js,index.html, and js files for the views</td>
<td>March 12</td>
<td>March 12</td>
<td>1hr</td>
<td>Emily</td>
<td></td>
</tr>
<tr>
<td>Create a static treemap view. Colour code each sector into individual blocks where the area reflects the sum of the market cap of that sector Add a hovering tooltip that shows detailed info</td>
<td>March 18</td>
<td>March 18</td>
<td>6hr</td>
<td>Emily</td>
<td></td>
</tr>
<tr>
<td>Create a static line chart that shows the price of one given company / S&amp;P 500 index Add a hovering tooltip showing the price on the selected day.</td>
<td>March 18</td>
<td>March 18</td>
<td>6hr</td>
<td>Tianhang</td>
<td></td>
</tr>
<tr>
<td>Create a bubble chart with the x-axis to display the market cap and the y-axis to display the rate of change in the market cap. Size code and color code circle Show tooltip on detailed company info when hovering over each circle.</td>
<td>March 18</td>
<td>March 19</td>
<td>8hr</td>
<td>Emily</td>
<td></td>
</tr>
<tr>
<td>When a block is clicked on the treemap, the bubble chart will only display the companies in the selected sector.</td>
<td>March 21</td>
<td>March 19</td>
<td>4hr</td>
<td>Emily</td>
<td></td>
</tr>
<tr>
<td>When I click on a particular circle, the circle will be highlighted. The detailed price level change will show up in the line chart.</td>
<td>March 21</td>
<td>March 21</td>
<td>6hr</td>
<td>Tianhang</td>
<td></td>
</tr>
<tr>
<td>Derive data for sector index</td>
<td>March 25</td>
<td>March 22</td>
<td>2hr</td>
<td>Emily</td>
<td></td>
</tr>
<tr>
<td>Allow users to zoom and pan on the bubble chart view.</td>
<td>NA</td>
<td>March 28</td>
<td>NA</td>
<td>Yuchen</td>
<td></td>
</tr>
<tr>
<td>A bubble chart will show the change rate between the start day of the range and the end day of the range. When I click on the sector, if no circle is selected in the bubble chart, the line chart will display the calculated index value for the sector.</td>
<td>March 29</td>
<td>March 23</td>
<td>7hr</td>
<td>Tianhang</td>
<td></td>
</tr>
<tr>
<td>Meeting time once per week+extra one before submission: what did we finished last week, what problems we encountered and what we need to do for the next week</td>
<td>March 14, March 21, March 28, March 30</td>
<td>March 14, March 21, March 28, March 30</td>
<td>4hr</td>
<td>All</td>
<td></td>
</tr>
<tr>
<td>Add search bar and update line chart.</td>
<td>NA</td>
<td>March 24</td>
<td>NA</td>
<td>Tianhang</td>
<td></td>
</tr>
</tbody>
</table>
Add the functionality to allow users to drag the bubble from the bubble chart and create their own baskets. Also, the line chart will show the total price over time.

<table>
<thead>
<tr>
<th>Task</th>
<th>Due</th>
<th>Completed</th>
<th>Time</th>
<th>Responsible</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add transitions of bubbles.</td>
<td>April 3</td>
<td>March 28</td>
<td>4hr</td>
<td>Yuchen</td>
</tr>
<tr>
<td>Add transitions of axes for the bubble chart.</td>
<td>NA</td>
<td>March 28</td>
<td>NA</td>
<td>Yuchen</td>
</tr>
<tr>
<td>Add initial zooming transition for bubble chart. Disable user zooming while the initial transition is in progress.</td>
<td>NA</td>
<td>March 29</td>
<td>NA</td>
<td>Yuchen</td>
</tr>
<tr>
<td>Add the brush feature to drag the interval of the line chart to the left and to the right.</td>
<td>April 8</td>
<td>March 29</td>
<td>8hr</td>
<td>Tianhang</td>
</tr>
<tr>
<td>Revise CSS styling and add CSS animation if needed</td>
<td>April 8</td>
<td>April 7</td>
<td>3hr</td>
<td>Emily</td>
</tr>
<tr>
<td>Adding or adjusting views depends on progress. and fix bugs</td>
<td>April 8</td>
<td>April 9</td>
<td>3hr</td>
<td>All</td>
</tr>
<tr>
<td>New: Use API to get company overview data</td>
<td>NA</td>
<td>April 8</td>
<td>3hr</td>
<td>Emily</td>
</tr>
<tr>
<td>New: Render company overview information</td>
<td>NA</td>
<td>April 9</td>
<td>NA</td>
<td>Tianhang</td>
</tr>
<tr>
<td>New: Add tabs for the line chart to allow users to switch between the views</td>
<td>NA</td>
<td>April 10</td>
<td>NA</td>
<td>Tianhang</td>
</tr>
<tr>
<td>New: Improve interaction between line chart brush and bubble chart transitions.</td>
<td>NA</td>
<td>April 10</td>
<td>NA</td>
<td>Yuchen</td>
</tr>
<tr>
<td>New: add loading spinner when the user initially loads the page</td>
<td>NA</td>
<td>April 10</td>
<td>NA</td>
<td>Emily</td>
</tr>
<tr>
<td>Add buttons to control zooming and selections on the bubble chart.</td>
<td>NA</td>
<td>April 10</td>
<td>1 hr</td>
<td>Yuchen</td>
</tr>
</tbody>
</table>

Due: March 31, git repo + canvas
<table>
<thead>
<tr>
<th>Task</th>
<th>Start Date</th>
<th>End Date</th>
<th>Hours</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add functionality to move the camera, when the user submits a search bar search. (optional)</td>
<td>NA</td>
<td>April 10</td>
<td>3hr</td>
<td>Yuchen</td>
</tr>
<tr>
<td>Meeting time before submission: work progress and what needs to be mentioned in the final documentation.</td>
<td>April 8</td>
<td>April 10</td>
<td>2hr</td>
<td>All</td>
</tr>
<tr>
<td>Write the overview and goals and tasks for the final documentation updated from M1, M2 writeup</td>
<td>April 13</td>
<td>April 13</td>
<td>2hr</td>
<td>Tianhang</td>
</tr>
<tr>
<td>Write the Visualization and reflection part for the final documentation updated from M1, M2 writeup</td>
<td>April 13</td>
<td>April 13</td>
<td>2hr</td>
<td>Emily</td>
</tr>
<tr>
<td>Write the data credits, and Project Management &amp; Team Assessment part for the final documentation updated from M1, M2 writeup</td>
<td>April 13</td>
<td>April 13</td>
<td>2hr</td>
<td>Yuchen</td>
</tr>
</tbody>
</table>

Due: Apr 14, git repo

- Tianhang mainly did the work related to the line chart and the basket part in the bubble chart and the interaction between the bubble chart and the line chart.
  
  # of work hours: 5+3+4+6+4+4+5+6+1+1+1+1+3+3+2+2 = 55hrs

- Yuchen mainly did the work related to the data preprocessing, bubble chart animation, and the interaction within the bubble chart.
  
  # of work hours: 5+3+8+12+4+3+3+4+6+3+1+1+1+1+2+2 = 59 hrs

- Emily mainly did the work related to the treemap, part of data preprocessing, and created a static view of the bubble chart.
  
  # of work hours 5+3+3+6+2+4+6+3+2+4+3+3+1+1+4+1+2+2 = 55hrs