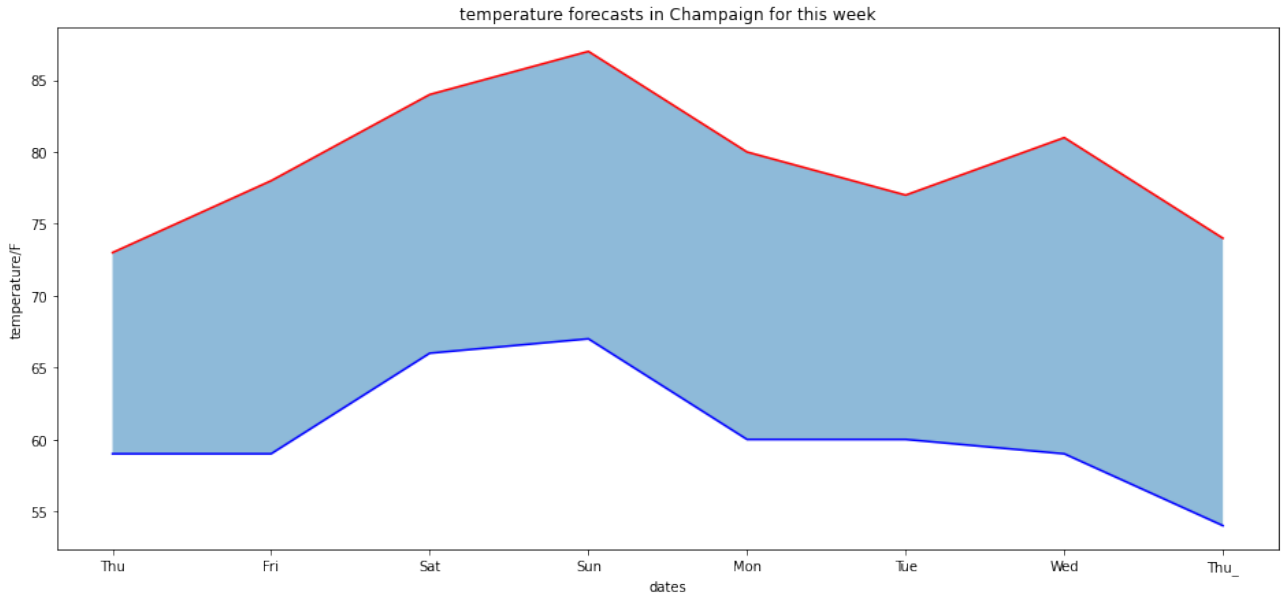
Compare Libraries

David Zhu

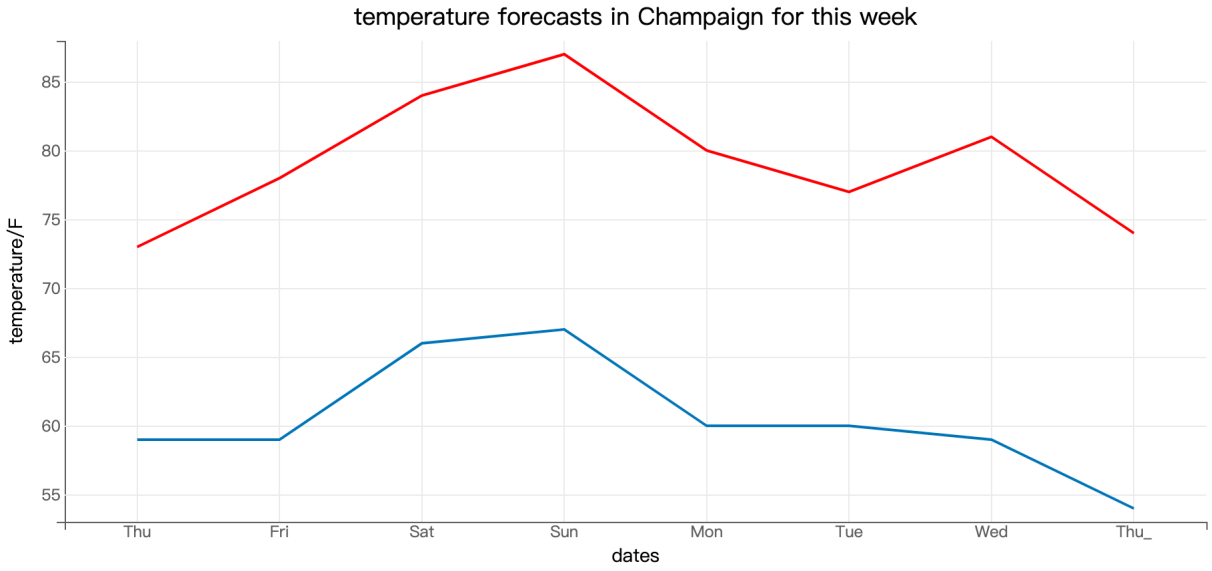
For this assignment, I explore a range of different plot libraries including matplotlib, bqplot, vega-lite, and d3.js. These libraries varies a lot in syntaxes, descriptions, and focuses. Both matplotlib and bqplot are under environment of Python while the other two are based on javascript and json. In this assignment, I used the dataset of the temperatures in a week of Champaign, Illinois.

The first one is the plot using matplotlib. The outcome is this picture:

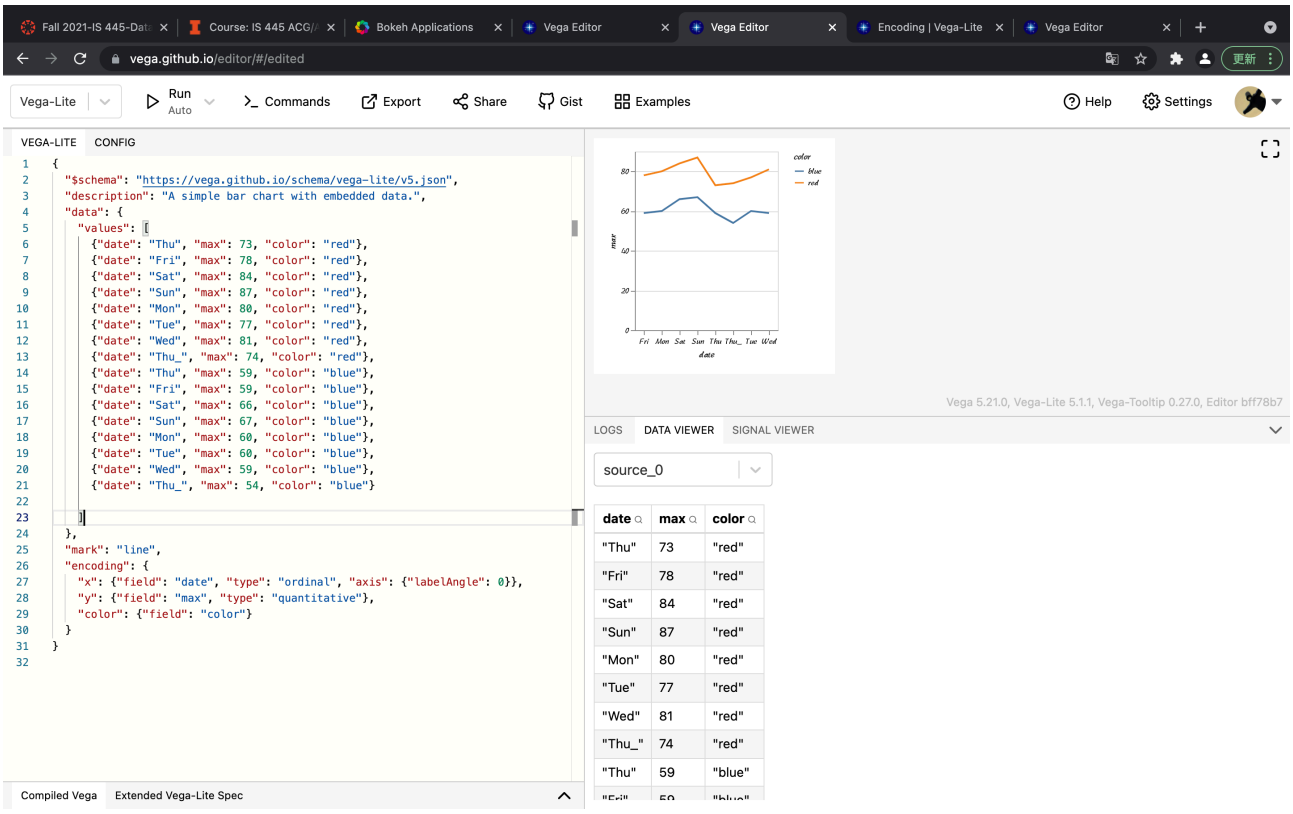


Matplotlib give the copyright to the contributors when they create new things out of the repository as long as mentioning matplotlib as a tool, “matplotlib uses a shared copyright model. Each contributor maintains copyright over their contributions to matplotlib. But, it is important to note that these contributions are typically only changes to the repositories” (matplotlib). The focus of matplotlib is basically for plotting data into graphs in pure python with lines of code. Matplotlib has little to no interactions because it outputs a static image. The good parts of matplotlib are it is powerful and basic. It has a relatively more complicated syntax but we can visualize a lot of things based on a wide range of functions and tools matplotlib provide to us. However, matplotlib only gives a limited theme to choose. Also, this library can only be operated in Python so it can not transfer through platforms easily. Also, the static image output can not be interacted by viewer.

This is the plot output generated by bqplot:

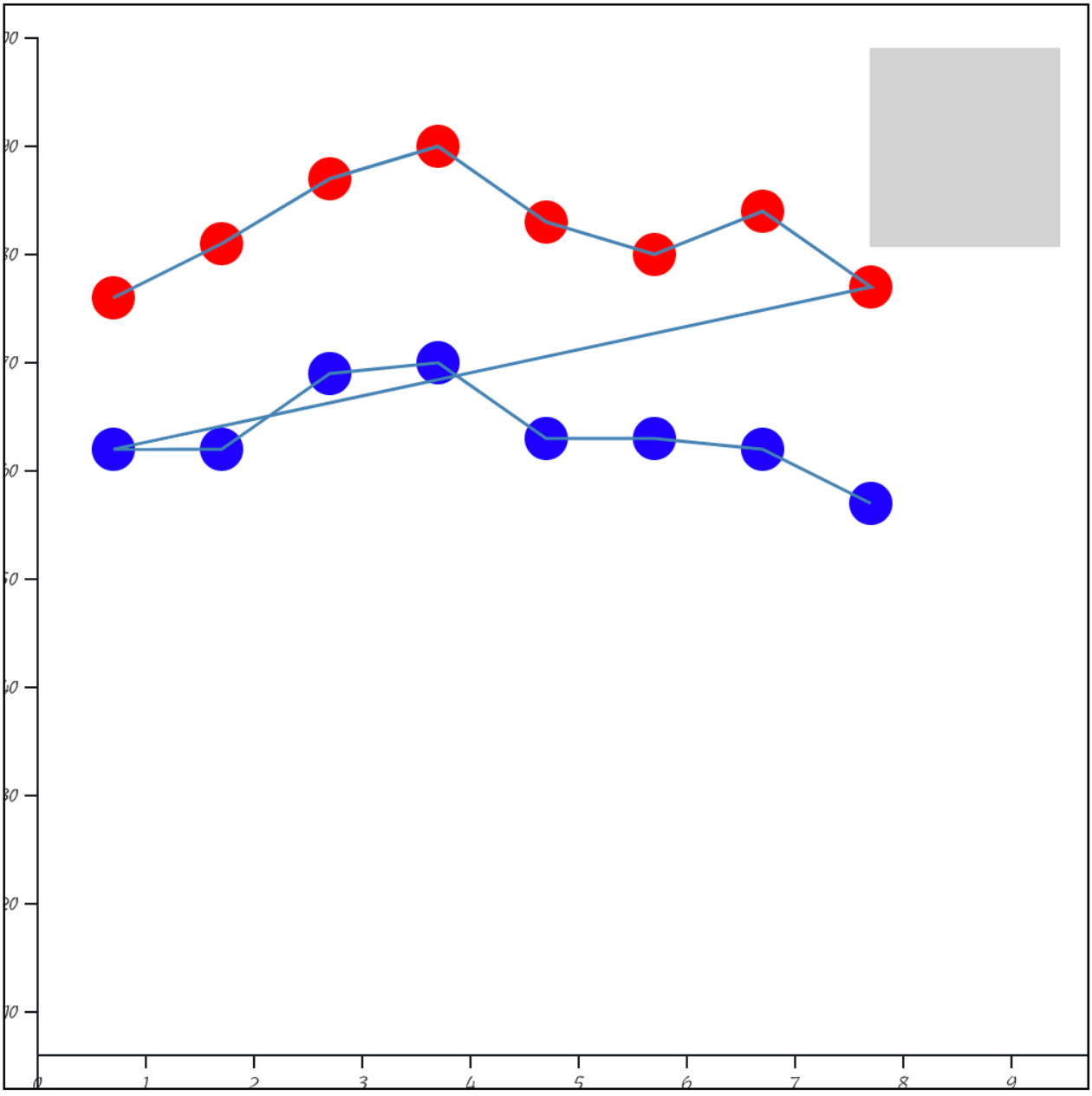


The license of bqplot is based on Apache 2.0, which give the permission to commercialize, modify, distribute, patent use, and personal use to the contributors. Bqplot focus more on the interaction part to matplotlib, while it has a better visual hierarchy than matplotlib. But it is still a library in Python so it is hard to migrate to another environment. Doing interactions in bqplot is a bit complicated because it uses a lot of new APIs that are not familiar with the original Python context. But it is succinct enough to do the basic parts of adding interactions by just declaring a interaction class and add it to the plot function.



From vega-lite, we start to migrate from Python context to the browser context. Vega-lite is a tool that we can convert json files to a visualization graph. This is the graph I create through vega-lite:
Because Vega has a great syntax jump from Python so I have a lot of bugs that can not figure out how to solve. However, Bega provide the license of “BSD 3-Clause "New" or "Revised" License”, which give the contributors the right of commercial use, modification, distributions and private use. The focus of Vega is to convert a single json file to a visualized graph so it is more declarative. Also, because Vega is more browser based, the interaction can be present because the dots are just Dom elements on a web page; however, because of the declarative syntax, adding interactions in Vega can be more complicated than adding it in Python. The good parts of Vega is it is light weighted so we can transfer a visualization just though a json file quite easy and convenient. While it is based on browser, we can make a lot interactions than in Python context. However, the bad parts of Vega is it is declaritive so all sentences are clustered together (arrays, dictionaries, another array…) that make it hard for human to read.

The final one is a visualization I made from d3.js:



I spend a lot of time on arranging the x axis because if I let it to be nominal, it just will order it in the alphabetical series. So I have to add a numerical indicator to tell d3 Thursday goes first and then end up in Wednesday. So d3.js is a open source library for javascript to do visualization on web browsers. However, what we did in class is slightly different from doing d3.js in pure javascript context, which will make us create instances of d3 first and then add characteristics to it. The license of d3.js is ISC License, which is basically a combination of BSD-2 clause and MIT license. We can commercial use, distribute, modify, and private use this library. D3 can do a lot of interactions while is it basically a library for javascript where creates the interface for viewers and the computer screen. Also, d3 can do interactions quite easily:

```
svg
    .call( d3.brush()
      .extent( [ [0,0], [width, height] ] )
    )
```

Just the code above can create a brush through the image. So d3 is really good at creating visualization widgets and interactions on web browsers. Also, because these elements in the graph are basically Dom elements so we can get a lot of manipulations than Python. However, d3 can be limited only to javascript and browser so it cannot be transferred to Python and other programming languages.