

Interactive visualization in Python

Yichen Dong



December 8, 2019

# Abstract

The purpose of this paper is to explore the interactive visualizations possible in Python. This will be mainly explored using Jupyter notebook, ipywidgets, and Bokeh. These will allow the user to select variables to plot and to filter the data. Bokeh will also allow a user to drag and zoom into a visualization. This will result in greater control of the visualizations that an end user might want to see.

# Introduction

As more and more data becomes available, tools need to be developed in order to help the analyst or other end user parse it. These tools often convert the raw data into visualizations that can more easily convey the information. We have dealt with different ways of making visualizations more effective in class, such as using color or other properties of visuals like enclosure and area. This is good in that it allows someone who is knowledgeable about the data and the tools to produce charts that describes the relationships in the data. However, often the data becomes very large over time, or very wide as business needs evolve. In cases like this, it is useful to give tools that simplify the process of generating new visuals. This is where the process of interactive visualization comes in.

By allowing a user to change a visualization based on an user interface, we give more control to the user to display the data that they truly care about. Such a visualization is useful in the case of a presentation, when the data being presented might be needed to be changed on the fly. Other uses is allowing a less technical user to also play around with the data without having to write code themselves. This is the benefit of a system like Excel PivotTables or Tableau, in that their graphical user interfaces allows the end user to intuitively specify what they want and to see it appear on the screen.

Python and Pandas might seem intimidating to a non-technical user, but with the right packages and interactive visualizations, it too can be as versatile as Excel and Tableau. This paper will explore ways that interactivity can be brought to Python through Jupyter Notebook.

# Background

Many packages have been developed for Python for the purposes of interactive visualizations. A lot of these are based on Jupyter notebook, which itself is an environment based on iPython (interactive Python). So you can see that Jupyter already provides a great baseline for interactivity. However, in order to fully take advantage of that interactivity, the user must be able to at least write simple code. There is no GUI interactivity built into Jupyter. This is why we need to use additional packages to fully perform the interactivity that we need.

The most well known of these packages is the Bokeh package. Bokeh is a tool that allows us to build interactive plots. Built into each plot is the ability to drag and zoom on individual parts of the display. However, Bokeh also allows easy building of widgets that allows a user to use a mouse pointer to interact with the visualization display. Bokeh has a large number of visualization options available as well, from the basic barplots to more advanced Geographical data.

Another useful tool for interactivity in Jupyter notebook is ipywidgets. This package allows us to build similar widgets to the ones used in Bokeh, but for a wider variety of applications. This can be useful when allowing the user to choose different ways to display a Pandas pivot table, for example.

# Approach

The approach that this author will take will be to use a smattering of datasets and produce interactivity that allows another user to get at the data. The datasets that I will be using are

1. <https://www.kaggle.com/arjunbhasin2013/ccdata>
   1. This dataset has 8950 observations
   2. It has 18 columns:

* **CUST\_ID** : Identification of Credit Card holder (Categorical)
* **BALANCE** : Balance amount left in their account to make purchases
* **BALANCE\_FREQUENCY** : How frequently the Balance is updated, score between 0 and 1 (1 = frequently updated, 0 = not frequently updated)
* **PURCHASES** : Amount of purchases made from account
* **ONEOFF\_PURCHASES** : Maximum purchase amount done in one-go
* **INSTALLMENTS\_PURCHASES** : Amount of purchase done in installment
* **CASH\_ADVANCE** : Cash in advance given by the user
* **PURCHASES\_FREQUENCY** : How frequently the Purchases are being made, score between 0 and 1 (1 = frequently purchased, 0 = not frequently purchased)
* **ONEOFFPURCHASESFREQUENCY** : How frequently Purchases are happening in one-go (1 = frequently purchased, 0 = not frequently purchased)
* **PURCHASESINSTALLMENTSFREQUENCY** : How frequently purchases in installments are being done (1 = frequently done, 0 = not frequently done) **C**
* **ASHADVANCEFREQUENCY** : How frequently the cash in advance being paid
* **CASHADVANCETRX** : Number of Transactions made with "Cash in Advanced"
* **PURCHASES\_TRX** : Numbe of purchase transactions made
* **CREDIT\_LIMIT** : Limit of Credit Card for user
* **PAYMENTS** : Amount of Payment done by user
* **MINIMUM\_PAYMENTS** : Minimum amount of payments made by user
* **PRCFULLPAYMENT** : Percent of full payment paid by user
* **TENURE** : Tenure of credit card service for user

1. https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset
   1. This dataset has 30k observations
   2. It has 25 columns:

* **ID**: ID of each client
* **LIMIT\_BAL**: Amount of given credit in NT dollars (includes individual and family/supplementary credit
* **SEX**: Gender (1=male, 2=female)
* **EDUCATION**: (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)
* **MARRIAGE**: Marital status (1=married, 2=single, 3=others)
* **AGE**: Age in years
* **PAY\_0**: Repayment status in September, 2005 (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months, ... 8=payment delay for eight months, 9=payment delay for nine months and above)
* **PAY\_2**: Repayment status in August, 2005 (scale same as above)
* **PAY\_3**: Repayment status in July, 2005 (scale same as above)
* **PAY\_4**: Repayment status in June, 2005 (scale same as above)
* **PAY\_5**: Repayment status in May, 2005 (scale same as above)
* **PAY\_6**: Repayment status in April, 2005 (scale same as above)
* **BILL\_AMT1**: Amount of bill statement in September, 2005 (NT dollar)
* **BILL\_AMT2**: Amount of bill statement in August, 2005 (NT dollar)
* **BILL\_AMT3**: Amount of bill statement in July, 2005 (NT dollar)
* **BILL\_AMT4**: Amount of bill statement in June, 2005 (NT dollar)
* **BILL\_AMT5**: Amount of bill statement in May, 2005 (NT dollar)
* **BILL\_AMT6**: Amount of bill statement in April, 2005 (NT dollar)
* **PAY\_AMT1**: Amount of previous payment in September, 2005 (NT dollar)
* **PAY\_AMT2**: Amount of previous payment in August, 2005 (NT dollar)
* **PAY\_AMT3**: Amount of previous payment in July, 2005 (NT dollar)
* **PAY\_AMT4**: Amount of previous payment in June, 2005 (NT dollar)
* **PAY\_AMT5**: Amount of previous payment in May, 2005 (NT dollar)
* **PAY\_AMT6**: Amount of previous payment in April, 2005 (NT dollar)
* **default.payment.next.month**: Default payment (1=yes, 0=no)

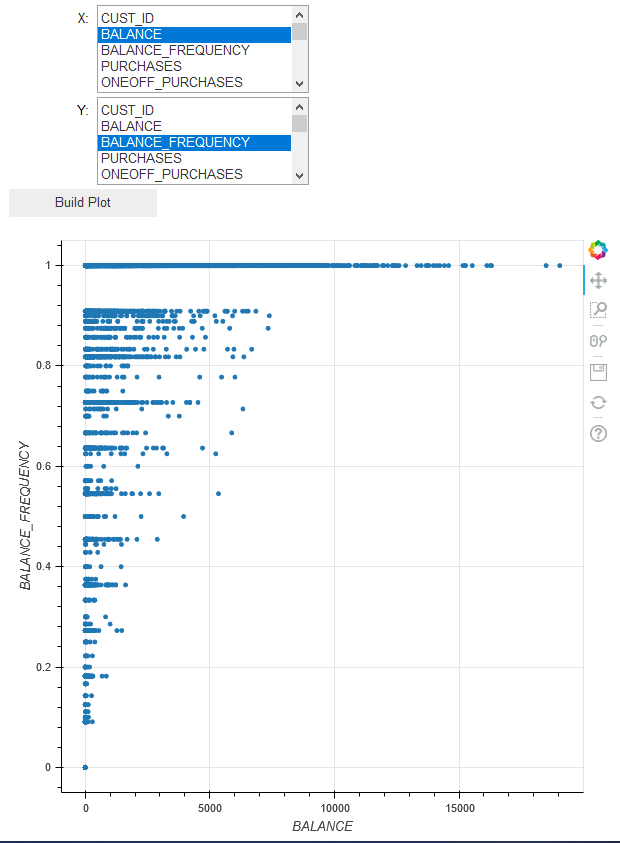
The reason that I chose these datasets is because of the fact that these datasets are related to my current field of work, which is the credit card industry. The idea is to explore all the interactive elements of Bokeh and ipywidgets piece by piece, so that I and the reader can get a good understanding of the interactivity possible with software like Bokeh.

Each of the visualizations will be displayed inline with the corresponding widgets. We will be going over some of the basic visualization techniques, such as scatterplots, line plots, barplots, and stacked area plots. We will play around with some of the tools offered by Bokeh, like the BoxSelectTool or the LassoSelectTool. WheelZoomTool, and HoverTool (to display values on hover). The widgets that we will be using are Button, CheckboxGroup, Select, MultiSelects, RadioGroup, Sliders, and RangeSliders.

# Results

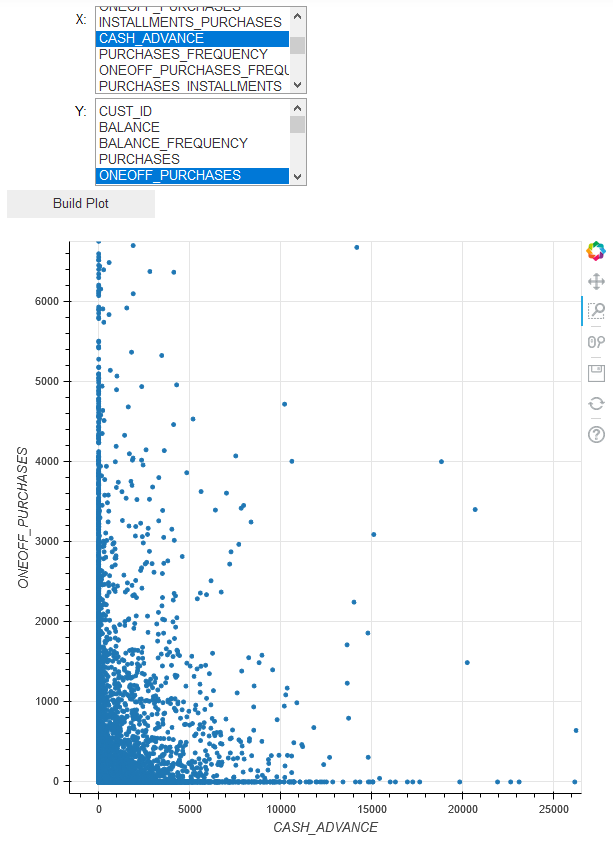
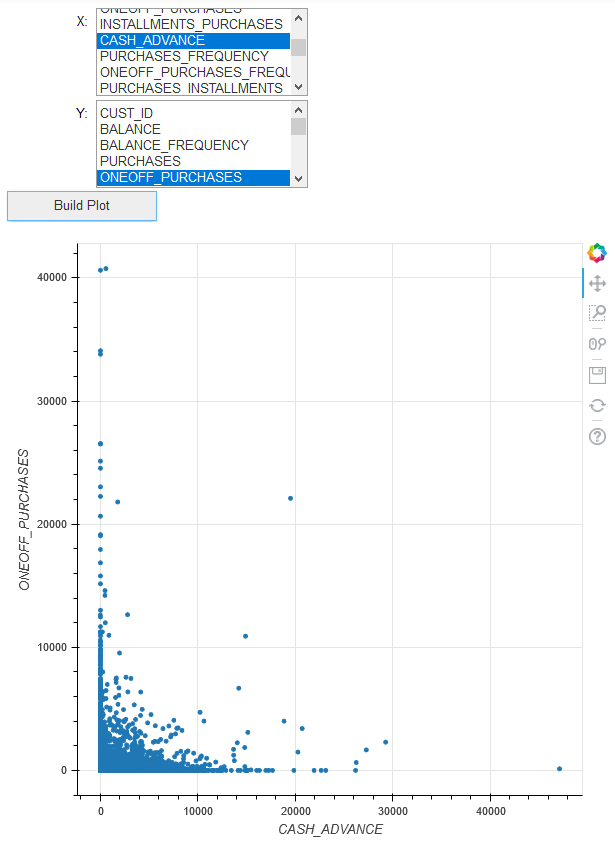
We were able to construct 7 different visualizations with interactions in them. They will be described as follows, and their function will be captured in multiple screenshots.

## Scatterplot of data elements in cc data using two menus



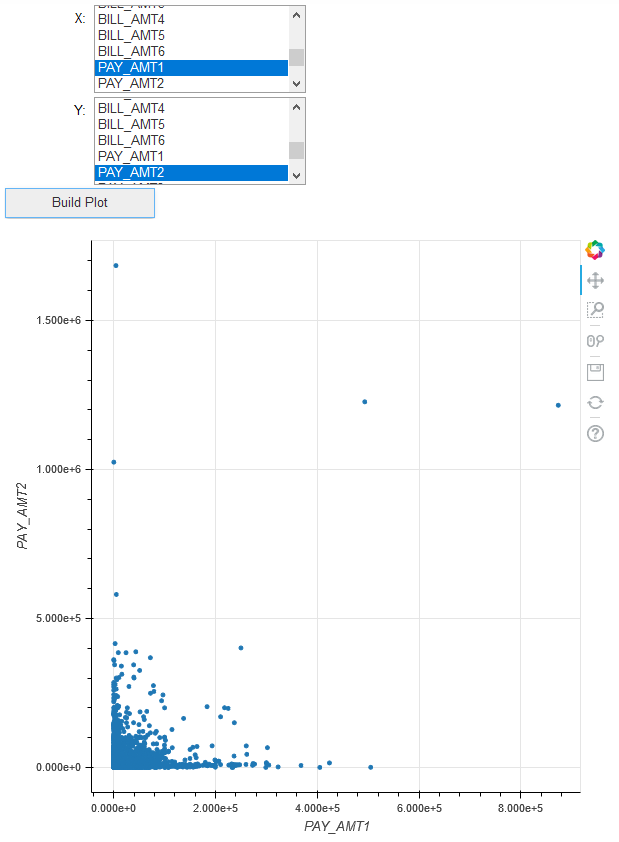
The user is able to plot two variables that appear in cc\_data against each other by selecting the relevant variables in the menus. In this example Balance and Balance Frequency are plotted against each other, but we can easily change the variables.

Below we have Cash advance vs. oneoff purchase amount. It should be noted that we can easily change the zoom in tbe bokeh plot using the box select as well.



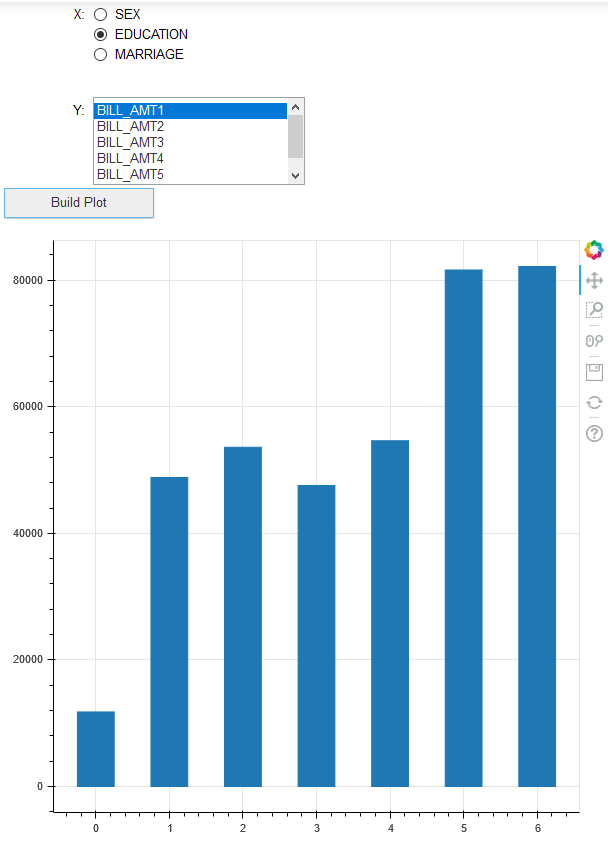
## Scatterplot of data elements in cc defaults using two menus

We can produce the same scatterplot utility in the cc defaults dataset.



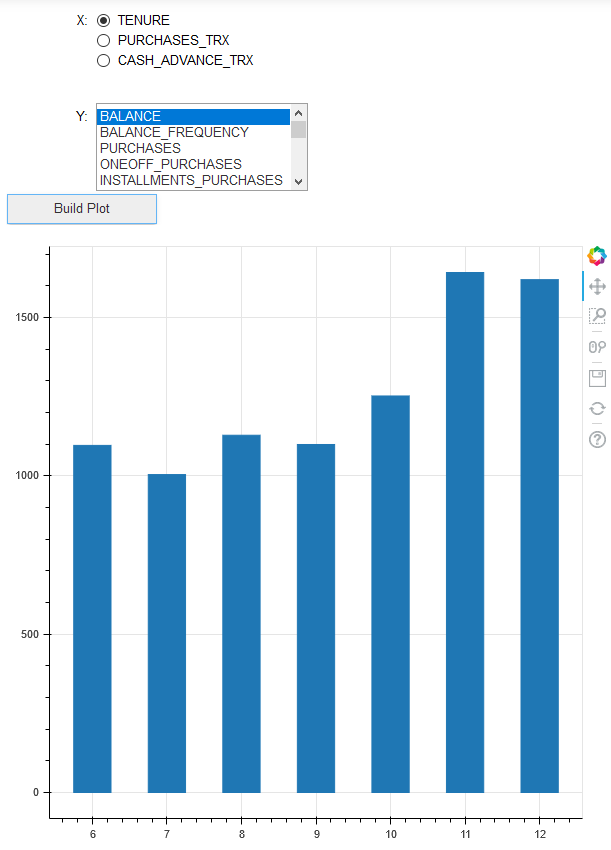
## Barplot of categorical variable vs. average bill\_amts in cc defaults data

You can see here that python also has the option of using radio buttons. Bokeh also has the option to produce barcharts based on categorical data, and that it has the option to create averages based on the categorical variable.



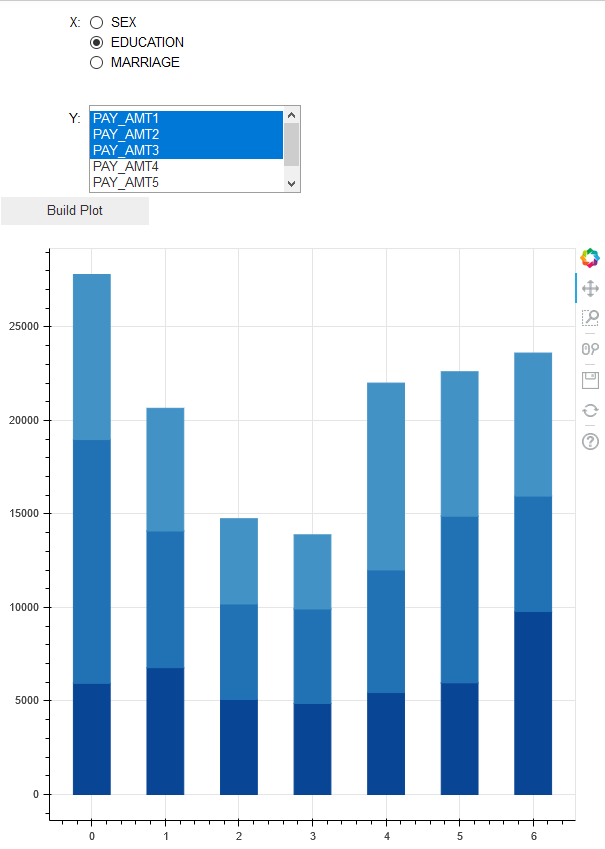
## Barplot of categorical variable vs. average bill\_amts in ccdata

We can apply the same logic as before to produce interactive bar plots in ccdata.



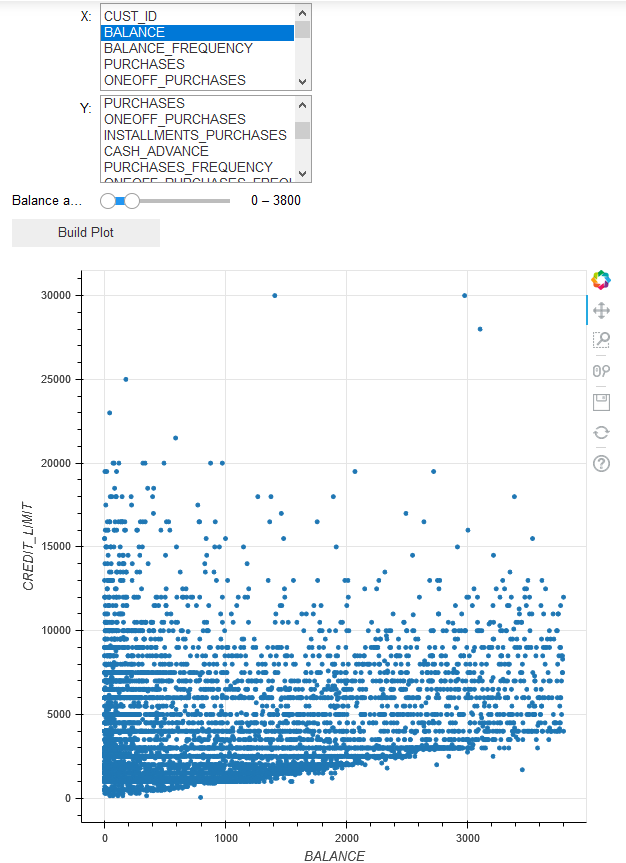
## Stacked barplot of categorical variable vs. average pay\_amts in ccdata

This interactivity allows the user to choose the the pay amounts they want and stack them on top of each other. This allows the user to see the total pay amounts over a time period compared to the other categorical variables. In this example we used 3 pay periods to show how they stacked up against each other and how they can be compared to each other.



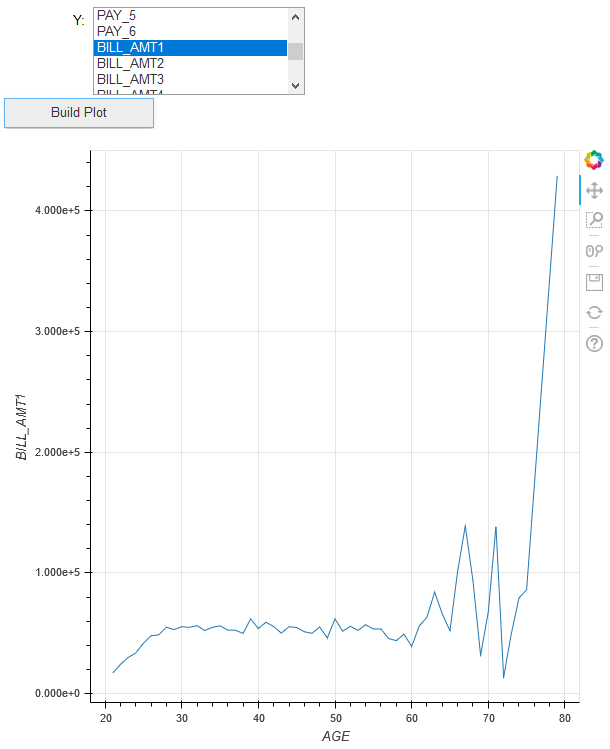
## Scatterplot of ccdata with range filter for balance

This plot modifies the previous scatterplot to provide a filter for balance. We can see that when we filter balance to be less than 3800, the scatterplot for balance vs. credit limit indeed ends at 3800.



## Line plot of cc default data with age vs. another variable

Here we can see an average of a metric, such as bill amount, across the age. We can see that the bill amounts increase from 20 to 25, at which point they plateau until 60. This allows the user to show the trend of a bill amount or pay amount across different ages.



# Conclusions

Jupyter notebook is a great tool for interactivity. Through the use of widgets, we can build in customization to a lot of different plots. This allows the end user to select different parameters without having to write the code themselves. We looked at allowing the user to select two variables to plot against each other. We also looked at allowing the user to select multiple variables to plot against one variable. In addition, we allowed the user to filter the dataset based on a range. Many other possibilities exist, such as geographical filtering, categorical filtering, stacked line plots, etc. The purpose of this paper was to establish that such an interactivity was possible in Python, not to explore the whole range of interactive possibilities. I hope that the reader goes away with a sense of what Bokeh and ipywidgets can do.

# References

Bokeh. (2019, 11 03). *Adding Widgets*. Retrieved from Bokeh: <https://docs.bokeh.org/en/latest/docs/user_guide/interaction/widgets.html>

This reference focuses on the use of widgets in Jupyter notebooks for the purposes of Bokeh. Its main contribution is to give me a sense of what is possible with widgets. It lists out the different types of widgets that Bokeh can use, and gives examples of how to use each widget. This will allow me to add interactivity to my graphs.

Bokeh. (2019, 11 03). *User Guide*. Retrieved from Bokeh: <https://docs.bokeh.org/en/latest/docs/user_guide.html>

This is the main user guide for Bokeh. Its main contribution is to give an overview of all the plots and interactivity possible with Bokeh. This has all the functions that I can play around with for my project, as well as examples to go with each function.

D'Angio, L. (2018, 11 19). *Interactive Data Visualization in Python With Bokeh*. Retrieved from Real Python: <https://realpython.com/python-data-visualization-bokeh/>

This article is a “quickstart” guide to Bokeh. Its main contribution will be allowing me an easy entryway into using Bokeh, and will allow me to play around with Bokeh before diving in more deeply. This is a pretty good overview that an in-depth resource like the documentations might not provide me.

that they want to generate word clouds for.

Koen, S. (2019, 05 03). *Bring your Jupyter Notebook to life with interactive widgets*. Retrieved from Towards Data Science: <https://towardsdatascience.com/bring-your-jupyter-notebook-to-life-with-interactive-widgets-bc12e03f0916>

This article talks about using ipywidgets to add interactivity to Jupyter notebooks. It does a great job showcasing all of the different widgets and how they can be used. It also provides examples of each widget.

Project Jupyter. (2019, 11 03). *ipywidgets*. Retrieved from Jupyter Widgets: <https://ipywidgets.readthedocs.io/en/latest/>

The main contribution of this is to be an in-depth guide to using Jupyter Widgets. This will allow me to fully use widgets to their full potential in my Jupyter notebook. These widgets will be used when not working with Bokeh.