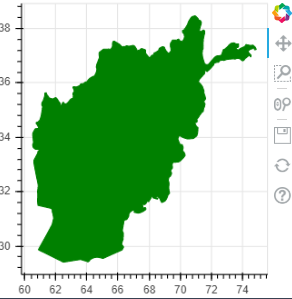
|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Activity** | **Problem** | **Solution** |
| 01-01-2020 | Download the data files in csv format  The Csv files are ordered that every row has country, year, data | - | - |
| 05-05-2020 | Set up virtual environment  Download packages:   * Bokeh * Pandas * geopandas | Geopandas cannot run on windows virtual env | Set up an Anaconda environment |
| 06-05-2020 | -Load the data into a jupyter notebook and add it to a geopandas data frame. | - | - |
| 07-05-2020 | -Download a shapefile of the world and open it in the notebook | -Had to learn more about the inner workings of shapefiles. A shapefile can be loaded where the shapes of the countries is a geometrical calculation saved in the column of a data frame. You can then plot these numbers which will create a graph that looks exactly like a country. | -Shapefiles can’t exists on their own and need some complimentary files to work |
| 08-08-2020 | - merge the date frame with the country data with the data of the data frame in.  - Merge the country based on country names. | - Structure of code, in order to implement multiple data sources the data needs to be cleaned in an external file.  - Country names are not similar so a merge would not be perfect. Countries are difficult object to work with because the names differ a lot between organization. A standard way of noting countries is the two letter ISO method. | - Create an external .py file to clean data and import the data.  - I downloaded a new shapefile which included these county codes and found a library called pycountry. This library can determine the code for a country |
| 13-5 | Download different shapefile with alpha 3 country codes which can be merged with the data. Started working in Jupyter notebook instead of multiple files. | - | - |
|  | Tried to create a data frame with the data where on every row the country and columns with data per year are represented | * might be slow, have a lot of nan values. | * Tried creating a dictionaries which would improve speed. |
|  | Decided to put the data in a dictionary where the key is the country name and the value is the shapefile object. Followed by the data for each year. I did this by writing a function that takes a 4 column data frame, a geodata frame with the shapefile and outputs a dict. |  |  |
| 15-5 | The next step is to input the data into a json file which can act as input to bokeh | -recursion error | None |
| 17-5 | Decided to try to go back to a geopandas data frame. Used the dictionary to structure the data and kept in the country code |  |  |
|  | Got the countries and code back into the data frame but the year data was still in dict format | -Make columns out of the values | - Got columns where each year is a column |
| 18-5 | Merge in the shapefile data with the data frame.  Succeeded in getting the data into a json data source |  |  |
| 19-05 | Write skeleton code for visualization and try to get output |  |  |
| 20-5 | Decided to dive into my data again. The goal is two things: Get prettier code (less) and a single data frame from where to subtract data from.  At the moment I have 4 data frames for all my different data sets. I noticed how when transforming the data I had to do the same operations over and over. So I wrote a function called transform() in which I would do all the transformation of the data. Here I flattened the columns so that ever year would be a column instead of a row. I also merged in the shapefile data frame and created a new column keeping track of the type of polygon ( polygon or multiplygons) don’t know if this is really necessary so it might not make it into the final version. Also added a column tracking to which data source it belongs  Also noticed some odd things with the data. South Sudan didn’t had a shapefile, maybe because it is a relatively new country and Taiwan had no data at all, this is probably china’s influence on the world bank, might search for some extra data so I can include them as well. | Cleaning the code | Single data frame and single function for transformation. |
| 23-5 | Tried to visualize the different countries. The first step is to try to visualize a single country. For bokeh you have to ‘unpack’ the shapefile. I had to lean move about the bokeh .patch() functionality to learn how to do this. With a single country you have to take all the x values and put them in a list and do the same with y values. When you then plot the figure it looks like a country (see appendix 1).  Problem emerge when she shape is a multiplygons instead of a polygon. This is the case when a country consist of multiple separate parts( Like Russia and its exclave Kaliningrad)  I tried to unpack the multiplygons and create a new row for each polygon with all the data for that year |  |  |
| 24-25-26 | Working with the multiplygons proved to be very difficult. The problem was that I had all the data and the shapes in a big single data frame. I did not manage to plot the multiplygons in bokeh ( see appendix 2)but I could unpack the individual polygons and plot them. So for about two days I tried different ways to plot the polygons, the biggest problem I ran into was speed.  With 245 countries merged and appending to a data frame is not that heavy, but with the multiplygons unpacked my data frame over 3500 rows ( Indonesia being the winner with over 250 polygons)  I decided to start over for a large part and take a different approach. Instead of merging the shapefile at the beginning I would unpack all the geographical data first. Do All the operations that would enable bokeh to plot the shapefile and then merge the data per year, This meant that the geo data frame would only consist of three rows ( Country code, xs and ys) and would later add a fourth row the data for that year.  This means that my initial data frame with all the data should be trimmed down a lot as it doesn’t have to include the shapefile data. |  |  |
| 27-5 | Split the code into the part for the data transformations and a part for the visualization this resulted in 2 python where the visualization file calls the transformation file |  |  |
|  | Got my visualization (partly) working it showed the data for a given year and the color value per country  Problems would emerge when applying the interactivity as it would become very slow. With every change in interactivity it would have to perform a merger of multiple thousand rows which would lead to slow interactivity  I decided to take a different approach, I was working with GeoJSONDataSource instead of a ColumnDataSource as this was advised in the Geopandas Documentation. The problem with the GeoJSONDataSource is that when the input becomes too large a recursion error ( 15 may I also had this problem) emerges. To my understanding this is an error in geopandas itself as the people working on this project no longer provide support for it and the internet doesn’t provide a good fix for the error.  Went back to create a single frame with all my data ( 4 different datasets each with 3500 rows of polygons + country data) and merge them together and put them into a ColumnDataSource. Got an error that was caused by the names of my columns which were years (integers) something that is not allowed. Changed the columns and managed to plot the map again. | Too much data operations in the visualization |  |
| 28-5 | Plotted my map and added a hover function.  Learned about adding a color palette to use for the countries and the data. Goal was that every color showed a different value where a darker color was a higher value |  |  |
| 29-5 | Started adding the functionality to the visualization. I added a dropdown menu and a slider and created a function they would call when a change was made.  A problem I encountered was the way that bokeh handles the colors in the visualization. I had to find a way to adjust the maximum so I hardcoded it into the function that fetched the data.  Removed total internet users as it correlated too strongly with population size and didn’t really say anything. |  |  |

1. Afghanistan as a plot of xs and ys



2. Only single polygons

