DataW Report

"Pathway from F-1 Student to U.S. Citizen" Visualization

Yanzhong Li Ran Xu Yusheng Fang

yli185@ucsc.edu rxu3@ucsc.edu yfang27@ucsc.edu

(1) Data Wrangling:

1. H1-B Applicants data (H-1B_Disclosure_Data_FY16.xlsx):

We obtain an excel format datafile, called "H-1B_Disclosure_Data_FY16.xlsx", which contains very clean information about H-1B visa. We are using this file to plot the graphs for "H1B Companies". For the purpose of easy process, I convert the data to csv file, and name it as "H-1B_FY16.csv". The data is relatively clean and informatic, except that there are some empty cells. I decided to delete the columns that lack chunk of information, and for the columns that include important information but with couple empty cells, I decide to delete the rows since there are over 64,000 records, couple rows doesn't influence much for the result.

2. PERM data (PERM_Disclosure_Data_FY16.xlsx):

We obtain an excel format datafile, "PERM_Disclosure_Data_FY16.xlsx", on the DOL website. We are using this file to plot the graphs for "Green Card Companies". However, there are still blank cells and dirty data in the file we need to deal with.

For the blank cells, we tried to infer its meaning from other cells. For example, if there's a blank cell under column "wage_range_max", we will try to find if the "wage_range_min" is also empty. If both are empty, we have to abandon this row. If we have the "wage_range_min", we assume this row wasn't given a range, but a definite figure. If both are given, we use the median value for future calculation.

For the visualization-ready files rendering, we create .csv files for each graph, so that while plotting the graph, there is no need to wrangle the data. But we have encountered some difficulties while doing so. For example, the data files use some kind of abbreviation, whereas the visualization library use a different version of abbreviation. We found a docs describing the relation between the abbreviations in viz library to the full country names. And we try to correspond the abbreviation on data file to full name, then to library abbreviation. For some reason, there are some countries that can't be mapped, we have to google it to make sure it maps to the correct abbreviation in viz library, except the viz library didn't include such countries.

3. EB Final Action Dates data (Cut-off_Dates_China.pdf, Cut-off_Dates_India.pdf)

The original data file is pdf including the EB queuing information of China and India in more 20 years. I tried to scrape the data from the original file, however, after trying a couple times, python library cannot recognize the table in each page of the file, so we tried another way, converting the pdf file into csv by using Adobe Acrobat. First we convert it to word, and then

save the table in csv file, which are called “EmploymentChina.csv” and “EmploymentIndia.csv”. But still, the file is not ready to use since each year’s EB queuing information of the each country is separate table, so in each file (say, EmploymentChina.csv), there are more than 20 separate tables for year from 1991 to 2016. So I write a script “EB_WaitingTime.py” to wraggle the data into one table for each country with showing the year in one column and the month in another column, and the EB processing time in other column, then I save it as two csv files named “Employment_process_China.csv” and “Employment_process_India.csv”. Next, I need to calculate the time length of the waiting. Except that the tables are separate, there are some letters in the data file, and for example, “C” which means no waiting time for EB visa, and “U” means unavailable. When I process the time length between the priority date and the time this particular priority date petition finally got processed, I set the time length for “C” and “U” to 0. Thus the script gets the waiting time lengths for EB1-3 visa and writes back into csv file.

(2) Data Analysis:

During this process, we not only analyze the data and obtain plenty of intermediate results (see ./output/queries/), but also select the best fraction of data to produce those visualization-ready data files in .csv format.

1. **H1B Applicants data**
   Input: H1B_FY16.csv

   Scripts: H1B_Company.py

   Output: state_count_workers.csv, AK.csv, AL.csv, ..(and all other states of U.S.).csv

We use the file “H1B_FY16.csv” produced by previous step as input, and write a python script “H1B_Company.py” to analyze the data. Because we are using python to manipulate a postgreSQL database for data analysis, we first need to run a PostgreSQL server on the machine. The steps for setting up postgreSQL is as following: i) download binary package for your system in the link https://www.postgresql.org/download/, and install postgreSQL following the instruction, and ii) you will need to configure postgreSQL and you can setup your host, dbname, user, and password. In python, first we use psycopg2 library to connect with the postgreSQL, and then use SQL to select the data under some criteria. In the python code, you will need to change the “conn_string” to the correct configuration in order to connect to postgreSQL when running python. This script ends up with 50 csv files total representing 50 states in the United States. In each file, it includes the information of companies in that state and the number of H1B applicants in the company. Those 50 files are ready to be visualized.

2. **PERM Applicants data**
   Input: PERM_Disclosure_Data_FY16.csv

Scripts: EB_Company.py

Output: _PERM_StateByCase.csv, AK.csv, AL.csv, ..(and all other states of U.S.).csv

q1_1.csv, q1_2.csv, …(all other query results).csv

For analyzing the data file “PERM_Disclosure_Data_FY16”, the python script file is “perm.py”. We first try to raise some questions, in order to create some useful features to be used in the visualization. We ask ourselves question like “What are the top 30 foreign-worker-friendly U.S. companies?”, “How good do they paid?”, “What kind of jobs they are offering the most for foreign workers?”. We have asked and answered 16 of these questions. We answered each of them by writing a fraction of python codes. And we saved each of these answers to a .csv files under directory ./output/queries.

At the end, we have figured out the answer for “how many foreign workers are being supported in 2016 in each state”, “in each state, what companies provide the most sponsorships for foreign workers”, “How much do they pay these employees”. After looking at all these answers, we choose those most useful to visualize them.

3. **EB Final Action Dates data**
   Input: Cut-off_Dates_China_online.pdf, Cut-off_Dates_India_online.pdf

   Scripts: EB_WaitingTime.py

   Output: Employment_process_China.csv, Employment_process_India.csv

Script “EB_WaitingTime.py” will be able to calculate the time length between the priority date and the time this particular priority date petition finally got processed. In order to do calculation, I need to change the month representation from words to number, for instance: change Jan to 1, etc. So I created a array to replace all the words to numbers. Since the wrangling result of the original file include two columns representing the time, which is the year and the month, so in order to calculate the time length, I multiply 12 to the year and plus the month number, and calculate the time length (which is the waiting time) for during each month. Thus the script gets the waiting time lengths for EB1-3 visa and writes back into csv file. The output file is called **Employment_process_China.csv** and **Employment_process_India.csv**. Those two files are ready to be visualized.

(3) Data Visualization:

While we haven’t directly used d3.js to program our visualizations, we have paid our attentions on python library like pygal, plotly, bokeh. Our visualization has graphs coming from each of these 3 libraries, and all of the output graphs are saved under directory “./output/viz/”. We use JavaScript selected which graphs to be rendered, interactively. We also use JavaScript to

actively render navigation bars, graph-switching buttons, etc. Our visualization are rendered on 3 different page (you get one page by clicking each button on the navigation bar).

1. The “H1B Company” page, and “Green Card Company” page:

Input: state_count_workers.csv, _PERM_StateByCase.csv, AK.csv, AL.csv, ..(and all other states of U.S.).csv

Script: Company_plot.py

Output: _PERM_Choropleth.html, AK.svg, AL.svg, ..(and all other states of U.S.).svg

We draw a horizontal bar graph on the left side, displaying the top 30 foreign-worker-friendly companies in one specific state. You can view the graph for other state, simply by hover your mouse above the state-switching table on the right. And if you hover over each bar on the graph, it will display a tooltip showing the company name and applicants number of that company. This graph is plotted using python library “Pygal”. We configure the suitable color scale, font size, bar style, legend style, etc.

On the right side of the page, a choropleth map is colored according to the total amount of applicants in each state. The darker the color of a state, the more applicants in that state. While hovering your house on each state, a tooltip would be shown to display the exact number and state name. And the map supports pan and zoom. This graph is plotted using python library “Plotly”. We configure the suitable color scale, font size, bar style, legend style, etc.

2. The “Green Card Waiting Time” page:

Input: Employment_process_China.csv, Employment_process_India.csv

Script: EB_WT_plot.py

Output: ALL_CHINA.html, ALL_INDIA.html, EB1_CHINA.html, …(totally 12 html files.)

The target files of this script are the output of previous step: **Employment_process_China.csv** and **Employment_process_India.csv**. The two files include the EB1-3 waiting time for China and India in recent over 20 years. The purpose of the script is to utilize python library “Bokeh” to visualize the EB1-3 visa waiting time for nationals from India and China. It can show the waiting time trend of every single type of visa regarding different nationality as well as every combination of the visa type and the country to do comparisons. So totally we generate 12 charts showing the visa waiting time. Each of this graphs is a scatterplot. And particularly in following 3 graphs (EB1_BOTH, EB2_BOTH, EB3_BOTH), we draw vertical dash lines between the two points, which indicating the waiting time difference between Chinese and Indian applicants in that certain month. It is able to pan and zoom the graph to view the data during certain time period.

(4)Large Data Files explanation:

Two of our data files are larger than 100MB, therefore it is unable to upload to Github. We are providing the link for downloading our three data files as follows:

1. PERM_Disclosure_Data_FY2016.xlsx
2. H-1B_Disclosure_Data_FY2016.xlsx