Where Rose Goes

Final Report

MA 490 - Data Mining
Elias White

Project Objectives:

1. To investigate the veracity of the "Indiana Brain Drain" phenomenon, described here, as it applies to Rose-Hulman

2. To investigate the current state-of-the-art data visualization tools within the python community

3. To expand awareness concerning the IPython Notebook and its capabilities, specifically its visualization capabilities

Introduction:

Indiana, despite its low cost of living and tax breaks, perennially finds itself on a list envied by none: The Wall Street Journal's "The Worst States for Keeping College Grads." This inability is referred to as "brain drain", in which we see disproportionate emigration of highly trained or educated people. Below are some statistics for Indiana as a whole:

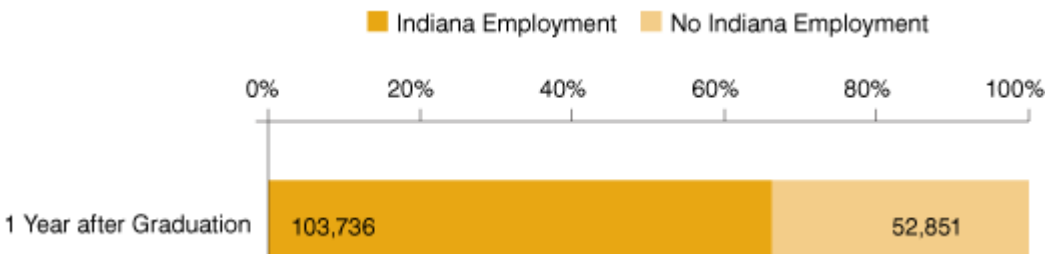

Figure 1) Probability of Working in Indiana after Graduation. Retrieved from: http://www.incontext.indiana.edu/2014/jan-feb/article1.asp

Here's the same graphic broken down by majors relevant to Rose-Hulman:

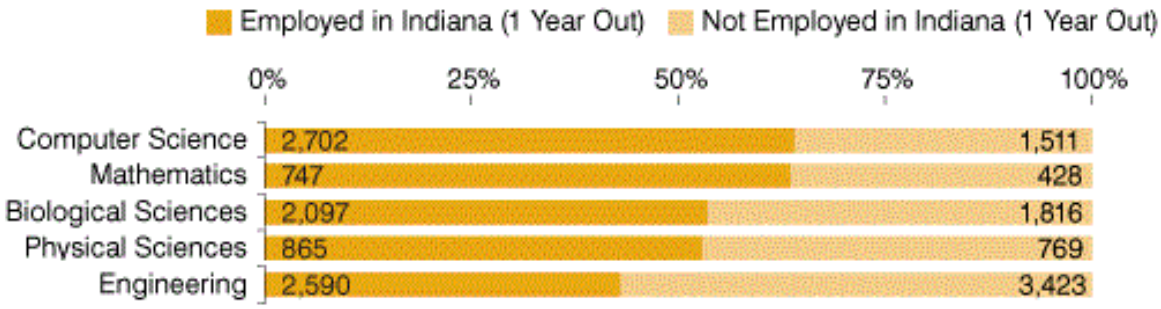

Figure 2) Probability of Working in Indiana One Year after Graduation by Major. Retrieved from: http://www.incontext.indiana.edu/2014/jan-feb/article1.asp

Datasets:

After speaking with the Career services office I was able to get two data sets, each containing information from the last 5 years.

The first of these detailed the all the offers submitted to the career services office, along with the company that gave the student the offer, the major of the student, the location (city and state) of the offer, and, when available, whether the offer was ultimately accepted or rejected. Here is what the first three rows of this dataset looked like:

	Company	Major	City	State	Accept
0	Accelerated Machine & Design	ME	Rockford	IL	A
1	ADM	CHE	Decatur	IL	A
2	ADM	CHE	Clinton	IA	A

The second data set also contained location and student major data, additionally it contained the salary offered to the student. Here's what its first three rows looked like:

	City	State	Major	Salary
0	Indianapolis	IN	CHE	58000
1	Indianapolis	IN	EE	58000
2	Clinton	IA	CHE	63000

The data as I received it was pretty messy, with lots of spelling mistakes, inconsistent syntax, as well as missing values, so a heavy amount of standardization was necessary. This involved sanitizing the inputs with some python string manipulation, as well as determining where the misspellings or abbreviates were and retrieving the correct value from a dictionary I created.

Because I wanted distance values I built up a little application that aggregated the distinct city-state pairs, querying Google Maps API with that information, retrieving and then parsing the returned XML string for the latitude, longitude, and county information. All of this was used to construct a dictionary that was then utilized to concatenate with the existing pandas dataframe columns containing the appropriate latitude, longitude, and county values. I then applied the Haversine big circle distance formula to the new latitude, longitude pairs to determine the distance and bearing from Rose-Hulman to each city-state pair. These were also included in the dataframe. Lastly there were several outliers (students accepting jobs in Japan or the Philippines), so for visualizations I discard those values.

Visualization Libraries Tested:

Because data visualization with the python ecosystem was a primary objective of mine, I spent a fair amount of time investigating different libraries. A lot of these libraries are relatively young, but with them I was able to produce some very descriptive and aesthetically pleasing graphics. Below are some of the libraries I looked at.

Pure Python

Basemap

Basemap is a matplotlib library for plotting 2D data on maps. Basemap does not do any plotting on its own, but does the coordinate transforms to one of 25 different map projections. Matplotlib is then used to plot in the transformed coordinates. Shoreline, river and political boundary datasets are provided, along with methods for plotting them.

Fiona

Fiona provides a minimal, uncomplicated Python interface to the open source GIS community's most trusted geodata access library and integrates with other Python GIS packages such as Shapely and GDAL's OGR, a libray that provids read (and sometimes write) access to a variety of vector file formats, including ESRI Shapefiles, PostGIS, and Mapinfo mid/mif formats.

Kartograph.py

This is a simple and lightweight framework for creating interactive vector maps.

Matplotlib

Matplotlib is the grand-daddy python plotting library. It allows for the production of an incredible variety of graphics using a MATLAB-like syntax. It provides the basis of a number of these other library's, who take it and expand either functionality, aesthetics, or both.

Seaborn

Seaborn is also built on top of matplotlib. Seaborn's core is a library of high-level functions for drawing statistical graphics. These functions are very smart about what they are doing, with a lot of advanced statistical analysis being done behind the scenes, including the calculation and representation of measurement uncertainty. Seaborn is tightly integrated with Pandas, so most of its functions take advantage of pandas' attributes to assign names to plot elements.

Shapely

Shapely is a Python package for manipulation and analysis of planar geometric objects. For a great example of how to use this and some of the other libraries mentioned here to create beautiful maps in python visit this link.

Python with some Javascript:

Bokeh

Bokeh is a python library produced by Continuum Analytics, the guys who put together Anaconda, for interactive visualizations using JavaScript and other web technologies. Bokeh goal is to provide python tools the style of JavaScript's Protovis and D3.

Google Maps JavaScript API v3

According to Google: The Google Maps API provides web services as an interface for requesting Maps API data from external services and using them within your Maps applications. These services are designed to be used in conjunction with a map.

Plotly

Plotly's python library allows for interactive, publication-quality plots in the browser. It saves the graph in your Plotly account, meaning the plot is online, and because it is online you can share it with that unique url or embed it in a web page. Regarding data rights: When you make a graph on Plotly you retain the rights to your content. You also control whether your graphs are public or private.

Vincent

Vincent is a bridge allowing pandas and the Vega visualizations it collaborate. You should be familiar with pandas, and Vega is a visualization grammar for creating and saving visualization designs. You can describe data visualizations in a JSON format, and then it generates interactive views using either HTML5 Canvas or SVG. Vincent allows for the data capabilities of Python and the visualization capabilities of JavaScript.

The libraries that produced the results I was most excited about were Seaborn, Bokeh, and Plotly. This is a result of their aesthetic beauty, their ability to tell the data's story better by allowing me to see otherwise hidden connections and trends, and, where applicable, the added benefit of interactivity. Interacting with the Google API in the IPython environment was also very fun. A lot of the visualizations I was able to accomplish weren't possible even a year ago and haven't been widely explored. It was very exciting.

Data Visualizations:

I experimented with a number of data visualizations, ranging from geospatial maps to violinplots, and I will summarize a few of them here.

Multivariate kernel density estimation:

This is a nonparametric technique for density estimation i.e., estimation of probability density functions. It can be viewed as a generalization of histogram density estimation, and it provides improved statistical properties.

Below is a visual desciption of what this means. The left is the 2-D histogram, the right is the corresponding multivariate kernel density estimation plot:

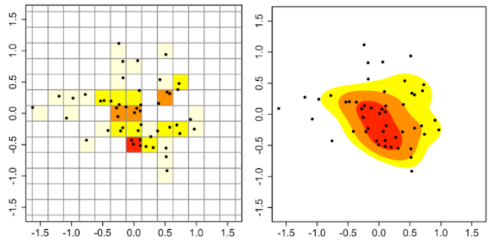

Figure 3) Comparison of a 2d histogram and the corresponding mkde plot

Here is a mkde plot I generated using Seaborn that shows the number of people accepting a job at a given location (capped at 5) on the y axis and the distance from Rose-Hulman on the x-axis:

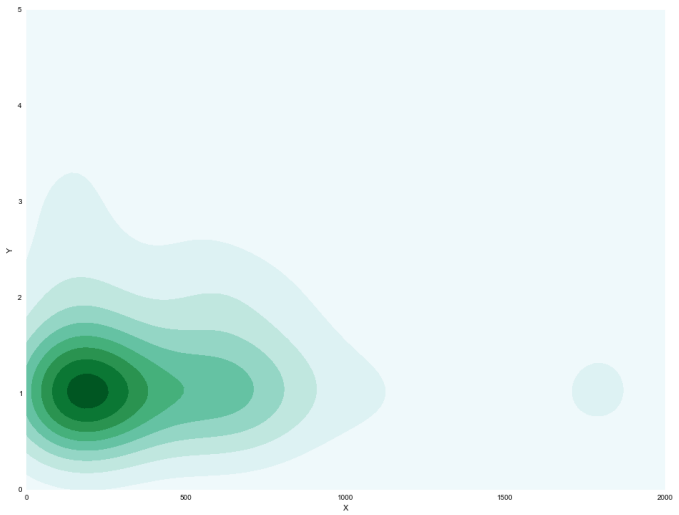

Figure 4) My multivariate kde plot showing the number of acceptances, capped at 5, versus distance

What this graphic shows is that if a person isn't going to one of the primary destination locations, like Houston, Indianapolis, or Seattle, they will most likely stay within 300 miles of Rose-Hulman, with a few drifting farther away. With regard to the Brain Drain, this shows Rose students tend to stick around Indiana if they aren't drawn to a major hub.

Violin plots:

A violin plot starts as box plot, then it adds a rotated kernel density plot to each side of the box plot. The result of this is a visual representation of the probability density of the data at different values. I think they are very cool, and here are two of mine. This first shows starting distance by major:

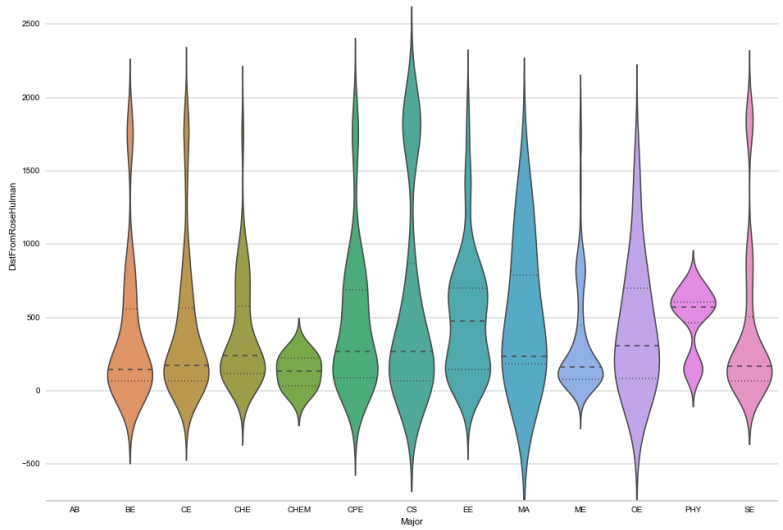

Figure 5) Violin plot showing the starting job distance from Rose-Hulman by Major

This plot is saying that, for the most part, students either stay close, go to around 1000 miles, Texas, Florida, and the D.C. area) or jump to between 1500 and 2000, which is the west coast. Very few people go to the Mountain region. Here is the violinplot for salary and by major:

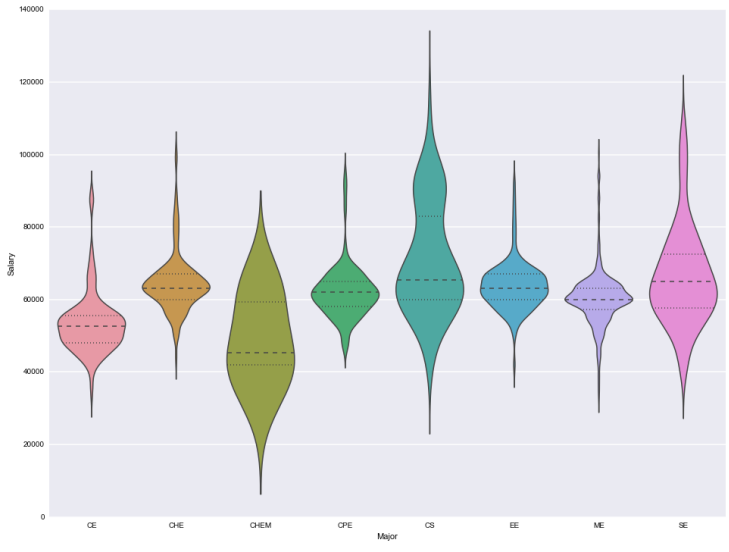

Figure 6) Violin plot showing the starting salary by Major

The salary data isn't surprising, with computer related fields making the most.

Scatterplots with Regression Analysis:

This is just a simple scatter plot along with some regression analysis included, giving some indication of how a dependent variable changes with the independent variables is varied.

Seaborn provides some amazing regression plotting facilities. It does the regression analysis and includes histograms for each axis. This is my regression plot showing salary vs. distance for computer science graduates:

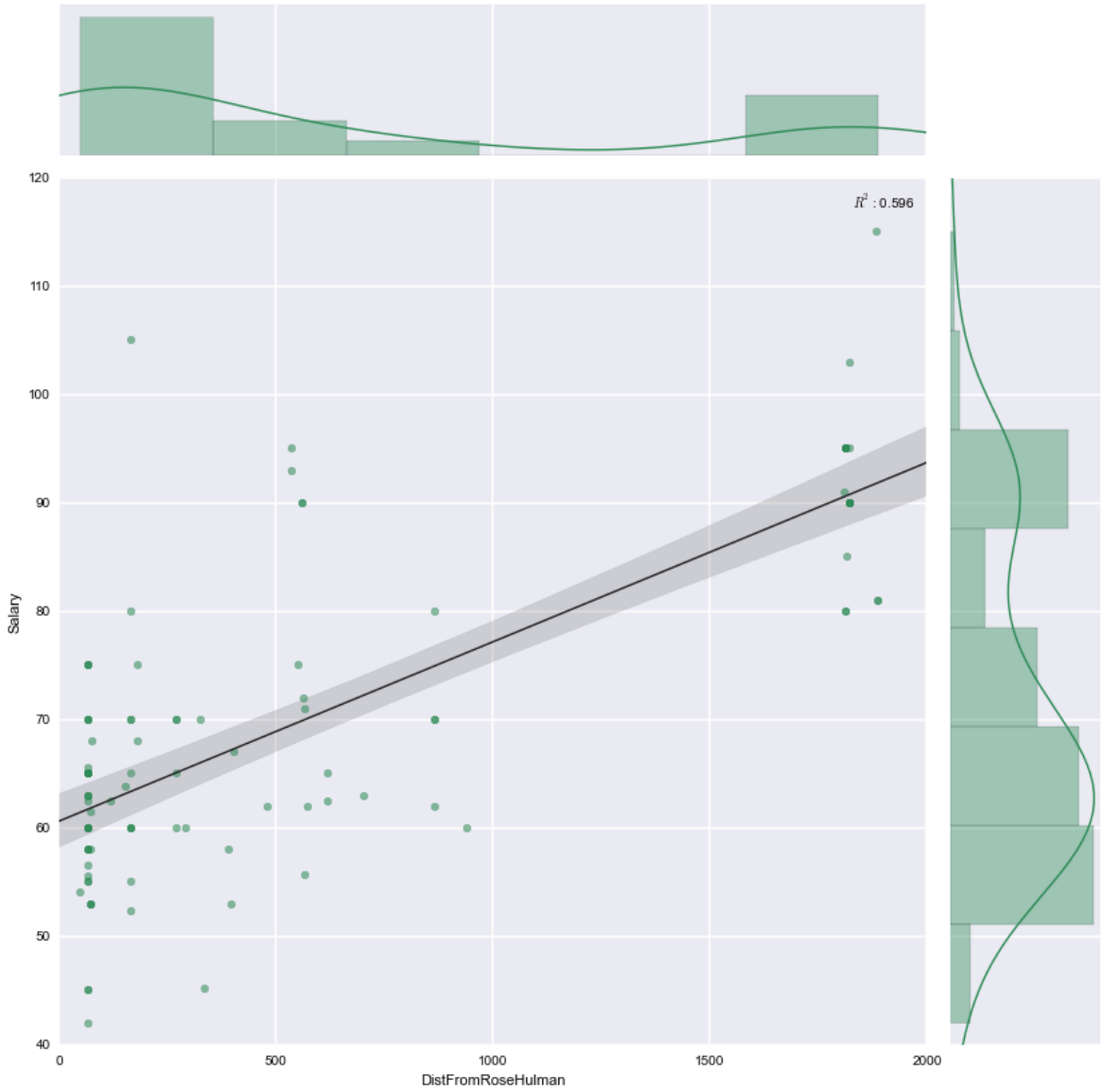

Figure 7) Regression plot of salary vs. distance for computer science graduates, with associated histograms

There is a strong linear relationships between salary and distance from Rose-Hulman for CS graduates. This was one of the stronger correlations I found, motived by information technology hubs typically being located along the coasts, with higher costs of living around those hubs.

Heatmaps:

Heatmaps offer intuitive visualizations of the values in a matrix. I used, because of its intractability and aesthetics, the Plotly library to generate the heatmaps below. One of the ways I used heatmaps was to view how the various majors interact with different states. The first shows the values counts by major for each state:

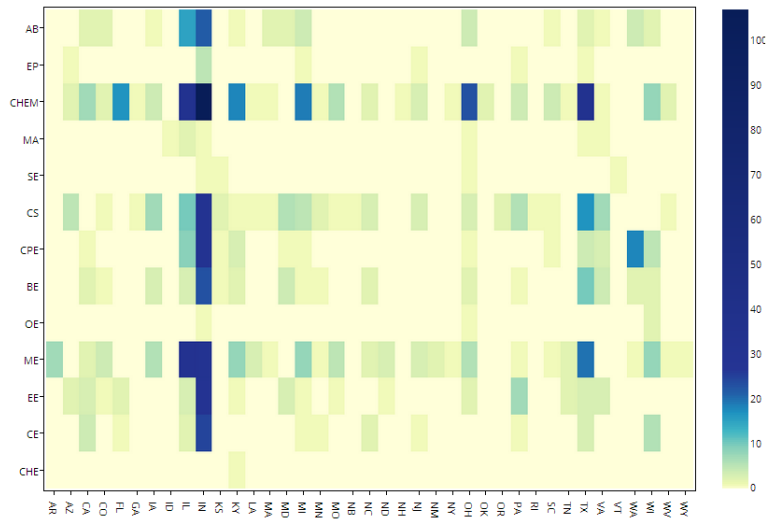

Figure 8) Values counts by major for each state. Note the logarithmic colormap scale

Clearly the most popular state is Indiana, but it is interesting to see how preconceived ideas about where each majors end up compare to reality, specifically CS (also CHEM and CHE seem to actually be off. I'll investigate this further, checking my aggregation code). Below is a heatmap showing offer acceptance percentages by state and major:

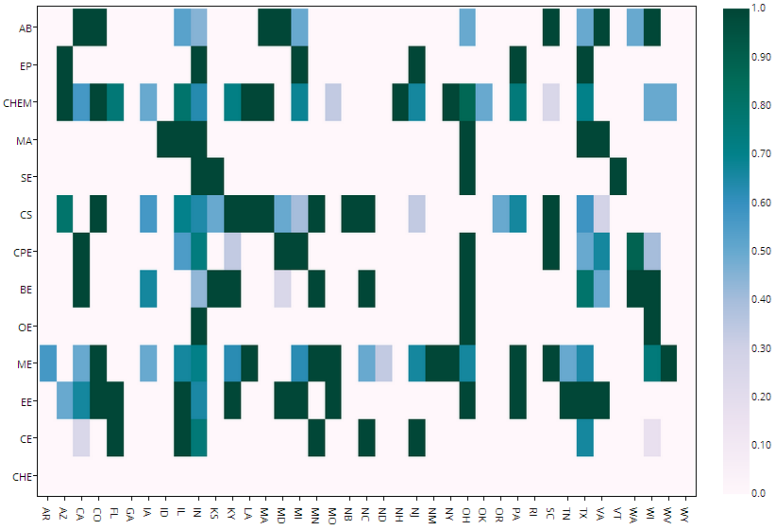

Figure 9) Acceptance percentages by state and major. In the IPython Notebook hovering over a cell gives the sample size

Interestingly Ohio has some of the best overall acceptance percentages, and EE's appear to not be very discriminating. Indiana tends to hover around 50% for each major.

To get a better understanding of just Indiana's retention percentage by major I created a bar graph using Plotly with just that information. The values for the bargraph below were calculated by dividing the number of each major staying in Indiana by the total number of graduates in each major:

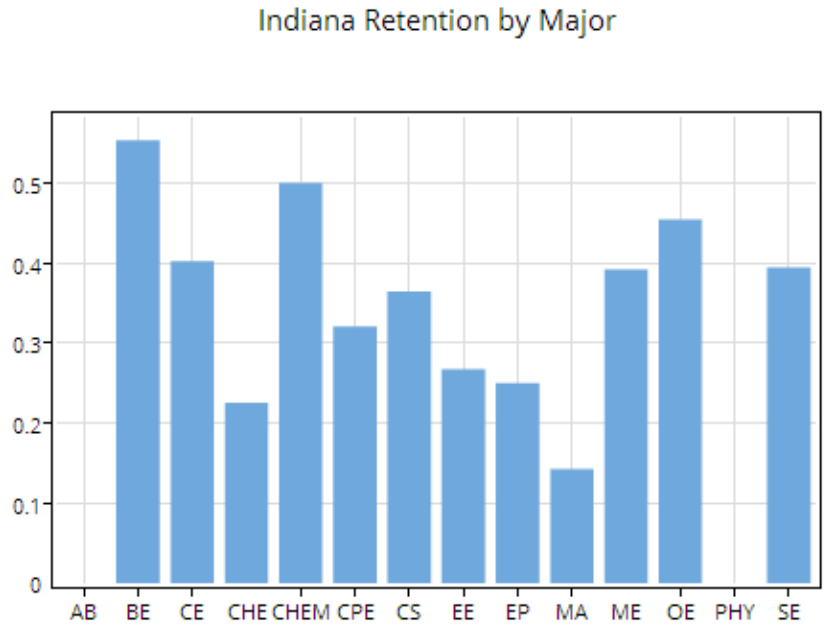

Figure 10) Retention percentages for Indiana by major

Here, again, are the overall statistics for Indiana:

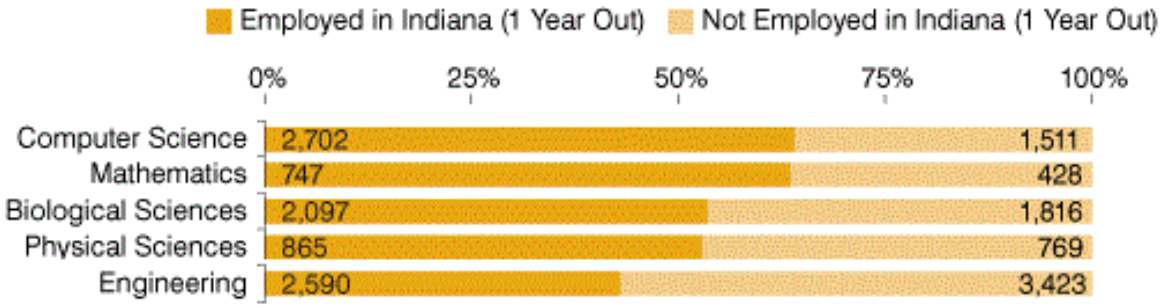

Figure 2) Probability of Working in Indiana One Year after Graduation by Major. Retrieved and then modified from: http://www.incontext.indiana.edu/2014/jan-feb/article1.asp

Rose-Hulman is pretty close to Indiana's overall retention statistics for engineering: hovering at the low 40 percent mark. Our statistics for mathematics, computer science, and the sciences are significantly below the average though. I attribute this to the fact that other institutions offer degrees in mathematics and the sciences as a part of their education program, meaning that the graduates will be elementary, middle, and high school teachers. These tend to stay closer to home. Rose-Hulman's mathematics and science graduate tend to attend graduate school or go to research labs, etc.

Geospatial Plotting:

Basic Basemap Visualization

My first map was a basic Basemap visualization with a great circle plotted to each destination. I set the alpha value to 0.5 so that the most popular destinations would have the darkest lines entering them. It distinctly shows the channels for our graduates to certain cities, specifically those in Texas, the Pacific Northwest, and Chicago.

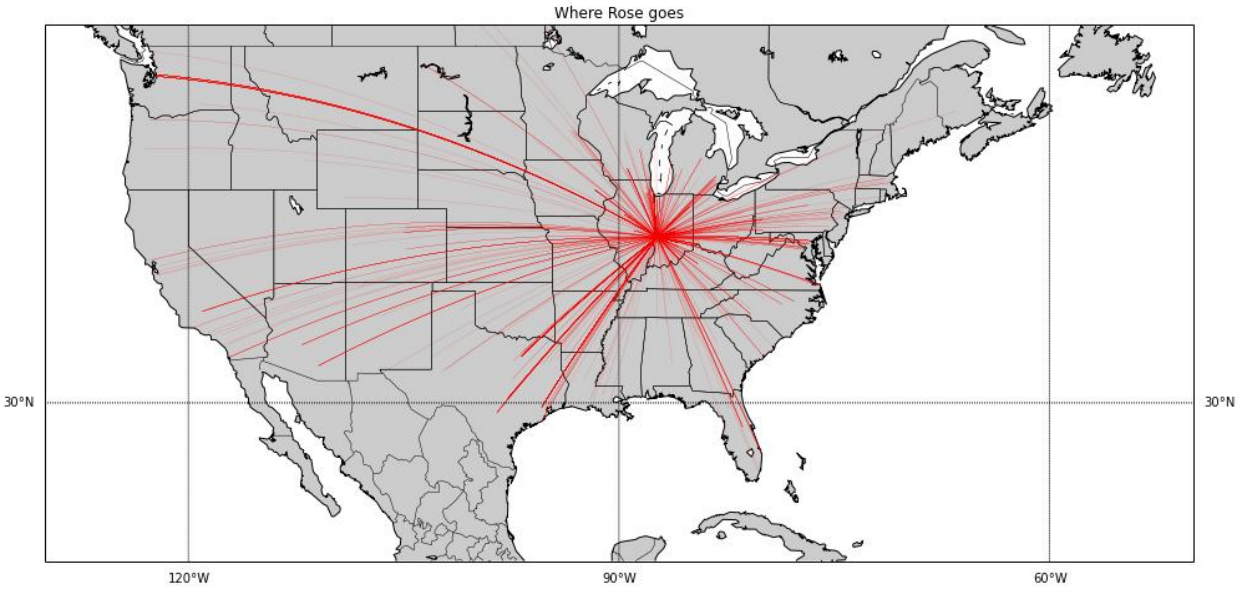

Figure 11) Basemap produced plot with great circle plots to each destination location

I thought this was a simple but effective visualization of the national footprint Rose-Hulman has. We are sending students to every corner of the continental United States, from California to Washington to Maine to Florida.

Choropleth Maps

A choropleth map is a map in which regions are shaded in proportion to the measurement of the statistical variable being displayed on the map. It provides an easy way to visualize how a measurement varies across a geographic area.

I've included two choropleth maps, one detailing the total number of students heading to each state and one detailing the total number of students heading to every county in each state. I generated these maps using the Bokeh library. This requires reading state and county shapefiles and generating a color value for each state or county. Below is my choropleth map on the state level:

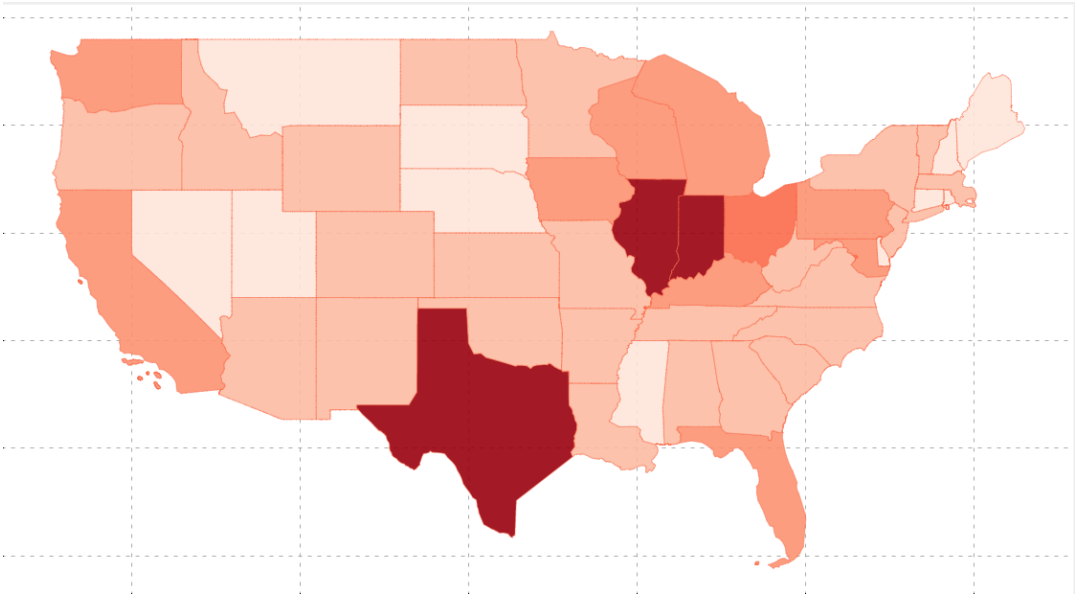

Figure 12) Choropleth map by state. Darker colors indicate a higher number of students heading there.

The most popular states for Rose-Hulman states are either those in the immediate vicinity, Indiana, Illinois, and Ohio, as well as Texas and states at the extremes. The states at the corners of the US, with the exception of Maine, are both well populated and have expanding technical markets.

Below is the county version of this choropleth map:

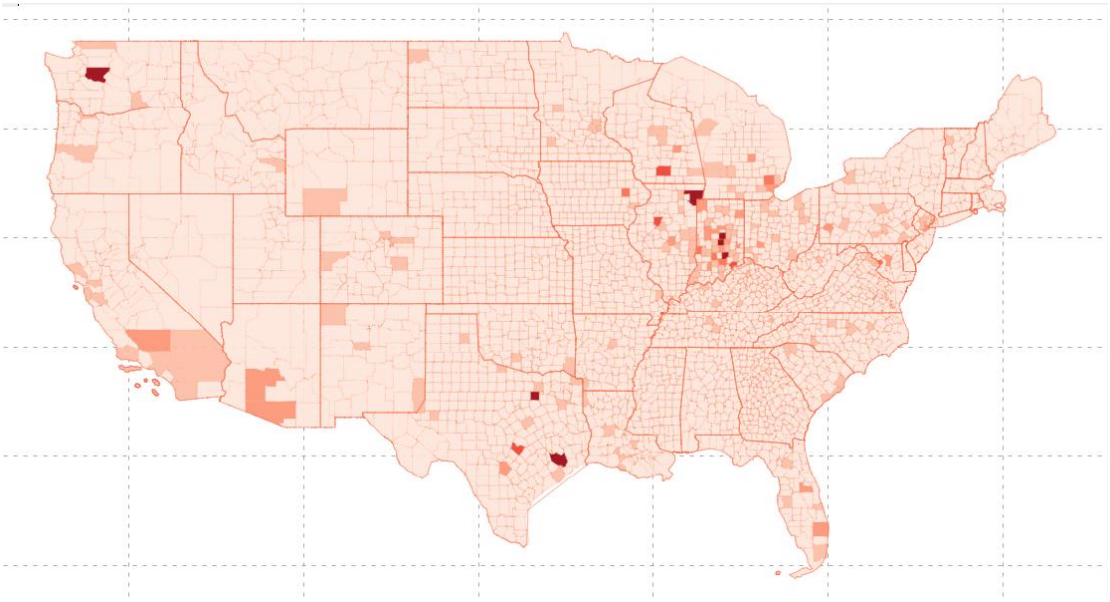

Figure 13) Choropleth map by county. Darker colors indicate a higher number of students heading there.

This map gives a more detailed representation of where the students go, but the overall impression for Rose's destinations isn't as obvious. Beyond Indiana the darker regions are typically major metropolitan areas that harbor a major technical population.

Google Maps API

The last map I'll mention here is a heatmap I generated using the Google Maps API. It shows the areas where the Rose students end up while providing all the extra functionality of Google which includes interaction, including zooming and panning, automatic location updates, and all the environment knowledge Google has amassed. I generated this in an IPython Notebook and embedded it in an output cell. The full extent of the API is available for use here, but I mostly used the heatmap. Below is a screen capture showing the heatmap over the full United States:

Out[31]:

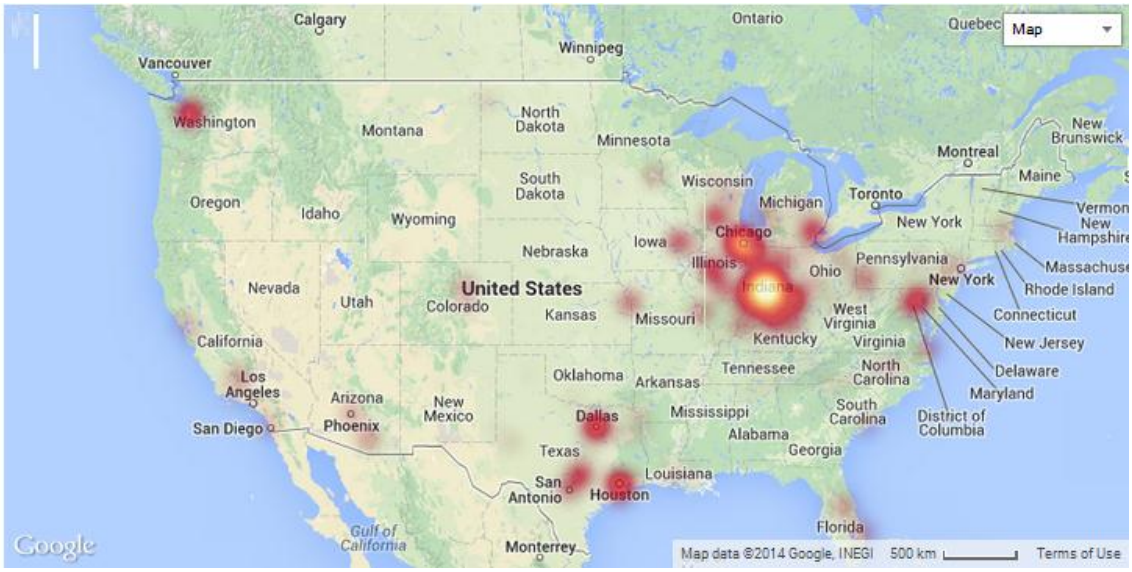

Figure 14) Heatmap of Rose student destinations on top of Google maps

This shows some of the same information that has been shown before, albeit it in a slightly different package. The primary point is to show that this sort of thing is possible in the IPython Notebook environment. The Notebook is capable of synthesizing python and JavaScript to produce impressive results. You can have the awesomeness of Python with the graphics, interaction, and power that JavaScript and the web provide.

Conclusions:

Indiana Brain Drain:

Rose-Hulman does play a significant role in the Brain Drain phenomenon. Rose-Hulman represents some of the most marketable talent that Indiana produces at the undergraduate level, and, as is shown by Figures 9 and 10, the majority of almost every major accepts their first job outside of Indiana. However, according to the heatmap in Figure 8, the most popular destination for Rose-Hulman graduates is still Indiana. This is primarily because of Indianapolis and Columbia, as is shown by the Choropleth maps, as well as the various little towns are Indiana the graduate return to, as is shown by the multidimensional kernel density plot. Some of the motivating factors that lead a student to leave Indiana are, besides Indianapolis and Columbia, the lack of major technical destinations as indicated by the county choropleth map, as well as the salary increase that accompanies some of the other locations, as shown for CS students by the regression plot.

Python's data visualization:

Python's data visualization capabilities are continually increasing, having beginnings with matplotlib and being carried forward by efforts like Seaborn, Bokeh, Plotly, Vincent, and increasing interoperability with the already mature JavaScript data visualization environment. I have shown a little of what is possible in this report, but this represents only a fraction of what can be done with the highlighted libraries. Many of these projects are in their, relative to other communities, infancy, and will only continue to improve.

IPython Notebook:

The IPython Notebook, already an impressive and exceedingly useful achievement, continues to be developed and has begun to create a niche of its own, with library developers working to make their visualization products interoperable with this tool. It is, as shown above, already possible to integrate Google Maps with the Notebook, and efforts are ongoing towards making the IPython Notebook capable of being a truly interactive in-browser data visualization tool. Prominent among these are the Python and JavaScript efforts of Bokeh and Plotly, as well as projects I wasn't able to delve into very much, including Jake Vanderplas' efforts to use the Notebook as a viewer for D3 applications. Lastly, I should mention the preexisting ability to intermix python with R, Octave, Perl, and Ruby, capacities I wasn't able to explore in this project.

Future Work:

There is still much that could be done in each of this project's objectives. The brain drain phenomenon could continue to be analyzed by individual state and major, as well in conjunction with salary and distance. The shear breadth of this, seeing as there are about 13 distinct majors and 39 states, is daunting. Beyond just looking at how this informs and reject the brain drain analysis of salary figures and accept/reject values is independently intriguing and could spawn completely new projects. I have begun analysis in many of these areas, but they weren't included in the project report due to relevance.

My search through python plotting libraries, while definitely pretty extensive, was by no means exhaustive. CairoPlot and Pysal specifically come to mind as examples of a Python libraries with which I did very little exploration. Part of this was due to the type of visualizations I was interested into, mostly geospatial with some regression/kernel estimation, not to mention the desire for interaction. I did go the entire Python Package Index, but, given more time, I would go through it with an even more fine-tooth comb, looking for up-and-coming plotting libraries with compelling capabilities.

As mentioned there are several efforts at integration with the IPython Notebook that I would like, but haven't been able to, investigate further, including the Jake Vanderplas D3 integration. I would also like to investigate how to manually pass data from the JavaScript runtime to the IPython Kernel and back, which would make interactive applications easier, allowing for execute python statement from and access Python output in, JavaScript. The results of this would be more, and prettier, interactive and animated visualizations in the Notebook.