**Continuing Your Adventure after Iowa State:**

**An Interactive Map to View Employment Data**

By Connor Jennings

**Abstract:** This paper purposes a new way of displaying the employment data at Iowa State University. The Engineering Career Services currently creates static one page summary reports that do not allow the user to see even a tenth of the data they collect or to interact with the document. To better present all the data the is collect, an interactive map for the web was created with plotted point of where Iowa State University Engineering students get their first job after college. The paper will also show the website’s functionality and how users will interact with the website.

**Introduction:** Currently at Iowa State University each College has their own career services. Each downloads data from Cyhire, the Iowa State Career Services Management System, then each College produces their own reporting format and creates annual reports that are available for download off each of their websites.

The first problem with this process is potential students cannot easily find all the data in one place and if the potential student is crafty, all the formats for each College are different and not easily compared. In a time of your life when you are selecting a major, your career for the rest of your life, students deserve full access to all the data.

Now all the reports have generally majors separated by level (i.e. Bachelor’s, Master’s, PhD) then an average salary and range for each year. The Engineering College gives a little more data by adding the number of each major living in Iowa, out of Iowa and the placement rates.

But ranges and averages of salaries can hide skewness, what if only one person made above six figures, this will make the range much higher looking and drag the average up higher due to one outlier. Also, if a student has a city or region they want to work in, they should not look at the range for the entire department, but rather a distribution of the salaries of past students placed from that department.

**Data Collection and Cleaning:** The data is downloaded directly from Cyhire in a csv format. The data is from a survey sent out to graduating students a few weeks before graduation. The data is in a standard form and has variables: Employer, City, State, Semester/Quarter, Work Type, Job Title, Compensation, Major 1 at Graduation, Major 2 at Graduation, Degree Level, Region, and Industry.

The survey allowed users to put anything into the cell for Employer, City, Job Title, Compensation, Major 1, and Major 2, while State, Semester/Quarter, Work Type were all drop down menus. Region and Industry were both auto populated in Cyhire.

**Cleaning Compensation:** The first step in cleaning up the data was cleaning all the salary data. Since the survey allowed users to input anything to that cell, users added commas and dollar signs to their answers. After using *gsub* to remove the “$” and “,” and converting all the zeros to NAs, not all the values were numeric. Some of internships in the data set had put their hourly rate with “/hr” or “per hour”. Using *gsub* we removed all none numeric characters and then assuming all the “salaries” that were under 100 were actually hourly rates, we multiplied then by 40 hour/week and then 54 weeks/year to get their projected annual salary.

**Cleaning Major:** Majors were a lot more uniform for allowing user to input anything. Degree with two parts like “Industrial and Manufacturing Systems Engineering” and “Agricultural and Biosystems Engineering” both saw people using “&” instead of “and” and dropping the second part of the major. I standardize this to the full name of the major with an “and” to make it easily searchable later.

**Add Latitude and Longitude:** To plot the data on the maps, the data set needs latitudes and longitudes for each Job listed in the data set. I downloaded a csv file containing all the city, state, latitude and longitude for all the towns in the United States from Maxmind.com. The state data in this data set was all abbreviations, so using the *state.abb* function we converted the full state names in the Cyhire data. To get the same format as the GeoLocation data, we used *paste* to group the “City” then a comma then “State”. Then the two data sets were put together using the *merge* function from *plyr.*

**Layout and Functionality:** The goal of this project was to create a *shiny* application that potential students could fully understand and use with-in seconds of loading the site.

**Map and Reactive Data Table:** The first step in creating an interactive map was finding the prefect package to create the map. The package *leafletR* was the prefect package for this project. *leafletR* was created by Joe Cheng, a software engineer at Rstudio and can be downloaded from his github using *devtools.*  It allows users to move and zoom in on a map of the world. The feature that makes this map package stand above the rest is its ability to be aware of the latitude and longitude boundaries of the map as it is displayed on the screen. Meaning, it gives programmers the ability to filter the data frames by using the limits of the map. This functionality gives users the ability to zoom into an area, say California, and all the search results, charts and graphs are only for California.

The points are plotted using size to indicate the number of Cyclones employed in that city. This gives users a chance to see where people are working at a quick glance.

**Filters:** Beyond allowing the user to filter data using the map, drop down boxes were added that allow users to filter by College, Major, Degree Level, and Work Type. College, Degree Level, and Work Type are independently populated by the factors found in the data set. Majors, however, is dependent of the College. But every one of the drop down boxes has all the levels of the data set plus “All” which makes the filter not remove any data.

**Slider:** Since the points’ size is weight by the total number of alumni in the town, as users filter, the circles get smaller and smaller. To combat this, a slider was added to manually adjust circle size on the map.

**Tables and Charts:** After the data set is filtered by the map and numerous dropdown boxes, it populates a data table under the map. Displayed in the data table are the variables, employer, job title, semester/quarter, and degree level. The data table updates automatically whenever one of the filters is changed. The data table is also searchable to give the user the ability to search through the remaining data for specific job titles or companies.

In the side bar there is some additional data displayed. First is a table listing the top 5 employers in the area. Second is a density graph showing the distribution of salaries. To protect confidentiality, if the user zooms into a region with five or less employees, the salary graph disappears to protect individual’s salaries. Also both are depend on the data table and are auto recalculated when the filter or map is changed.

**Next Steps:** After showing the finish produce to the Engineering Career Services, the director spent several minutes moving the map around and trying to analyze the data. The non-technical people that work in that office never had the tools available to them to analyze the data at this scale with this ease.

He said that employers ask him all the time what a competitive salary is and now he will be able direct them to their website and look for themselves.

In the future, I would like to add all the Colleges to this shiny. The College of Engineering has expressed interest in creating a similar website or an add-on to this existing one for job posting on Cyhire.

**Conclusion:** The goal of this project was to create a shiny application that would better display the data gather by the Career Services at the University. The Shiny application not only will allow potential and current students to decide majors better, but employers to give more fair wages and gives the Career Services a better tool to analyze their data.

**Appendix:**

**Geolocation data set:**

<http://dev.maxmind.com/geoip/legacy/geolite/>

**Shiny Map Example by Joe Cheng:**

<http://glimmer.rstudio.com/jcheng/leaflet-demo/>

**Shiny Package Github Site by Joe Cheng:**

<https://github.com/jcheng5/leaflet-shiny>