

CSCI 572 Homework3

Team Member:

Chenyin Sang, 2812623778

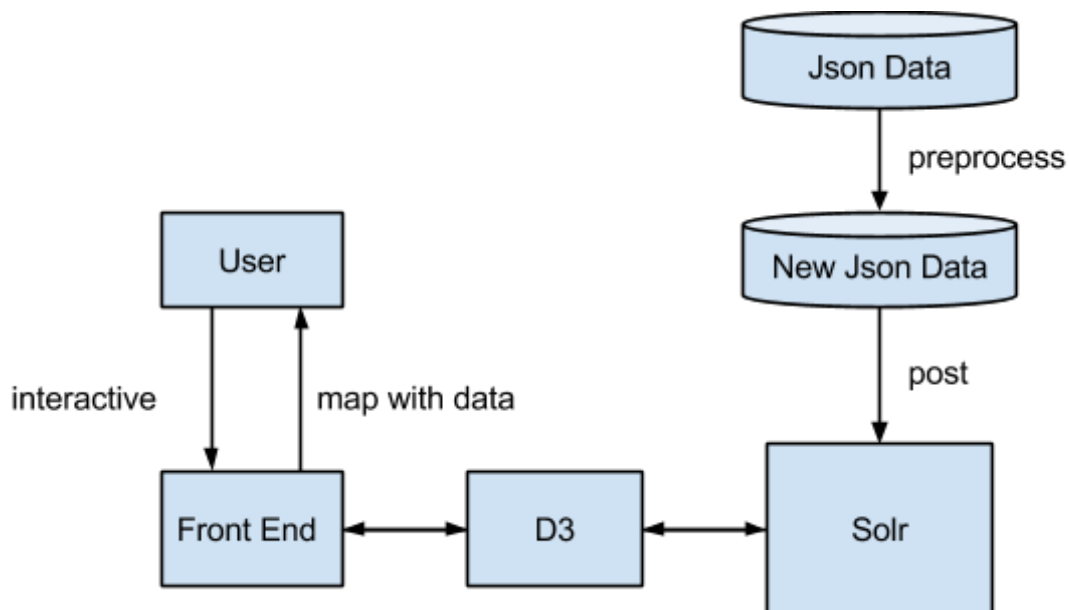
Dongni Zhao, 3282822727

Hongzhi Li, 1322106452

Wenhai Zhu, 9262786977

1. Introduction

This homework is asking us to visualize the job information we worked on in the former two homework. We should use D3 to plot the job postings in a map according to different requirements. The whole process is as followings. First, we need to build a web front end. In the front page, we need to show a map of the world and this map is used to draw dots later. Second, for different requirements, like “visualize job postings geospatially and temporally by various attributes such as company”, we need to get data from Solr. In order to doing this, we need to write queries and get the return in Json format. Now, we have a front end and we also have a way to get data. The last step is to combine them together. Use D3 to load data into front end and then show the map with job postings information.



2. Data preprocess

Due to what we want to search, we decided to add three columns on our data set. The first one is the attribute of year, the second one is the attribute of month, the last one is season. As this dataset has the data through the whole year, there must be two winter inside the data. We classified season as “year-

season” format in order to make a difference. Such as, “2012-winter”, “2013-winter”, “2012-spring” and so on. The month and year attributes are added by writing pure Java code. Moreover, there is no need to keep pagerank, so that we delete pagerank from our original dataset.

3. create a map-based interface

3.1 D3

D3.js is a currently popular data-driven visualization tool. It uses a combination of HTML, CSS and SVG. The most intelligent and advanced feature in D3 is interaction. When you want to show a part of data view, just shrink the range by mouse, or you can control the view directly by clicking the map. In our homework, we added two check boxes on the map view and one text field to make the interaction with users. Moreover, D3 dynamically binds data and shows data on the view from a data array. We use this feature in showing our data by seasonal and year. D3 has four advanced and easy-to-use technique among other data visualization method. The first one is called “Selector”, the same as the jquery selector, to retrieve position based on HTML tags. The second one is called “Transition”, values for attributes and styles can be smoothly interpolated over a certain time, the same as user interaction. The third one is called “Data Binding”, which means dynamically binding data onto selected tag position. The last one is called “Append”, which means the data array can use enter or exit to call for each data that even not exist on valid position. The exit function will be called when you want to remove all the data away.

3.2 How easy to use was D3? What was the hardest part, loading data, or visualizing it? What are the advantages and disadvantages of D3.

D3 has a lot of built in APIs for creating map with GeoJson or TopoJson and rendering data on map. All the APIs are really powerful and facilitate our job. However, it is not easy to figure out how these API works and it does take time to learn.

Through the whole process, from loading data to visualizing, the most difficult part lies on visualizing the needed data to map. Firstly, we have to retrieve and filter the needed data from the whole dataset returned from Solr. Then, when rendering on map, different data entry has different format, like shape, color. That’s also what we needed to handle before rendering. Last, animation for data transition is not easy as well.

The most significant strong point of D3 is the vast data visualization template that we can access to. We can customize them or build anything we want. And D3 is also good at handling multiple data format, such as XML, JSON and CSV. Last but not least, it can traverse the dataset for us and we do what we want in the callback function.

Unluckily, D3 is not that easy to master. It costs a lot of time to learn before you can really utilize its powerful function. And D3 is also not compatible in some old version browser.

how do we build a map?

To build a map, firstly, we need the world map json - world.json and then we can project the world data to SVG with mercator projection which adjust the geographic coordinates for display on our flat screen. With the map, then we can load the data from Solr RESTfully for our data rendering and interaction. For rendering, each cycle on the map represents one single job posting. Or we group several adjacent job postings and scale the cycle radius based on the job postings number count. In order to show the seasonal transition on data and the trends of job postings, we utilize the some d3 animation effect to redraw the data on map.

4. Solr web-based REST

Select the category, longitude and latitude grouping by area when company equals to the value user provides and for each season.

```
http://localhost:8983/solr/select?wt=json&indent=true&q:*:*&fq=comapny:"+input+"&fq=season:spring
&group=true&group.field=Area&rows=5000&fl=category,longitude,latitude&group.ngroups=true
```

Search the locations of all the job postings grouping by area for each season.

```
http://localhost:8983/solr/select?wt=json&indent=true&q:*:*&fq=season:spring&group=true&group.fiel
d=Area&rows=5000&fl=category,longitude,latitude&group.ngroups=true
```

Search the longitude, latitude grouping by area when company equals to the value user provides.

```
http://localhost:8983/solr/select?wt=json&indent=true&q:*:*&fq=company:"+input+"&group=true&grou
p.field=Area&rows=5000&fl=longitude,latitude&group.ngroups=true
```

However, the json which solr calls back is not compatible to D3, since the cross domain problem, and we will discuss this problem in section 5.

5. Combine D3 and Solr

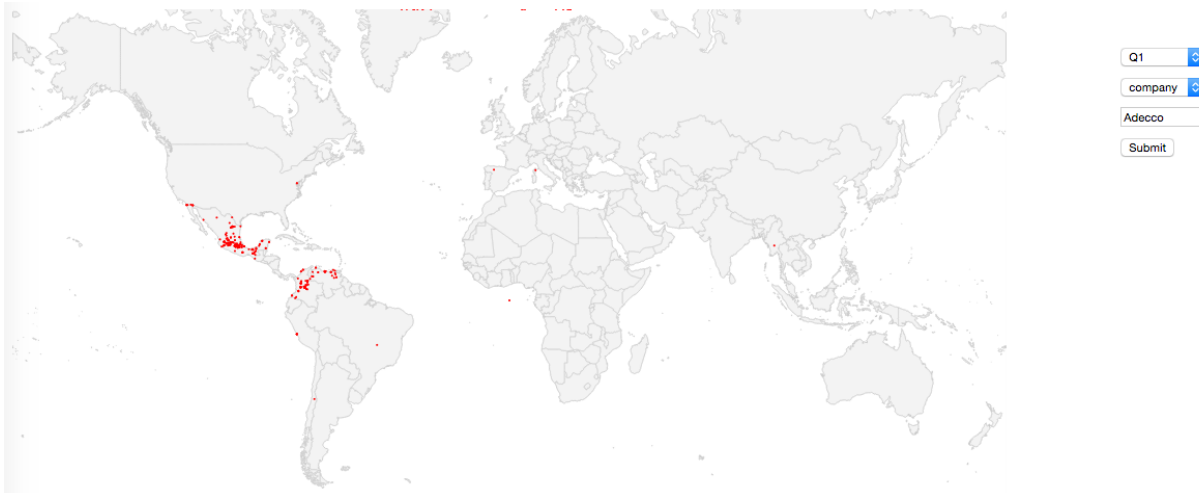
There is a big problem when we use D3 to getl the data from Solr. When we use the query as a parameter of D3, it won't work due to the cross domain issue. We can't get the response data. So we spent some time to fix this problem. Finally we use ajax and jsonp fixed it. The code is like:

```
$.ajax({
  'url':
"http://localhost:8983/solr/select?wt=json&indent=true&q:*:*&fq=company:"+input+"&group=true&grou
p.field=Area&rows=5000&fl=category,longitude,latitude&group.ngroups=true",
  'data': {'wt':'json', 'q':'your search goes here'},
  'success': function(d){ projectDot(d,"red")},
  'dataType': 'jsonp',
  'jsonp': 'json.wrf'
});
```

6. Describe in detail how your map answers the challenge questions presented in Task #3. Specify how to use the map to visualize the results.

a. Visualize job postings geospatially and temporally by various attributes such as company, job type, salary, etc.

This question is asking us to visualize the distribution of job postings according to different attributes, like company, job types and salary. So we need to show two information of job postings. First, we settle down one attribute, and then show how job postings distributed geospatially, and then show how they varies with time changing. In order to do this, we add a drop down list menu which a user can choose different attributes and add an input box which can input a string which user can input the value of the attribute, like the name of a company, a certain salary, or a jobtype. For example, you can choose “company” and input “Adecco”, the interactive data-driven model D3 will give back the result and quickly show the result on the map. In order to show the distribution of job postings varies with time, we use seasons to represent the time, we define first season as January to March, second season as April to June, the third season as July to September, and the forth season as October to December. So after the first distribution shown up, it will disappear after a while and then the distribution of second season is shown up, and then the third season and the forth season. One example map is liking follows:

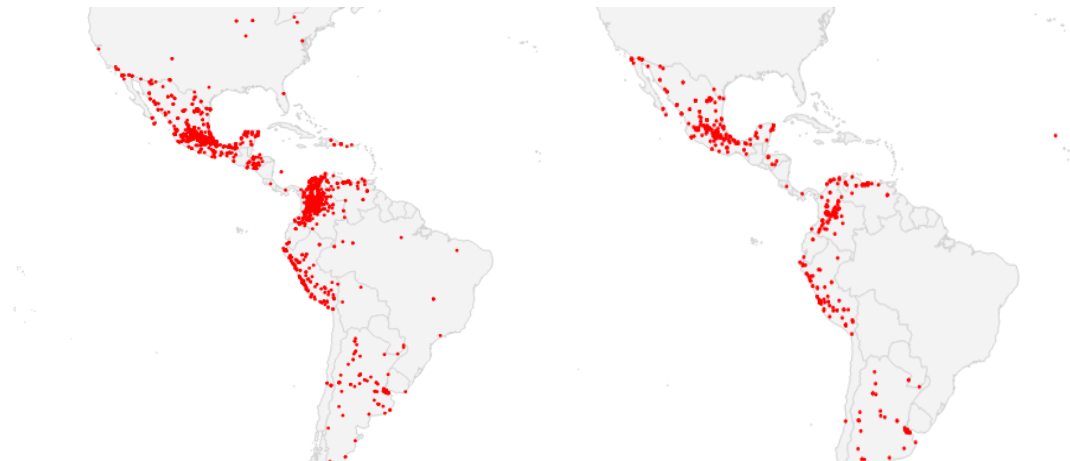


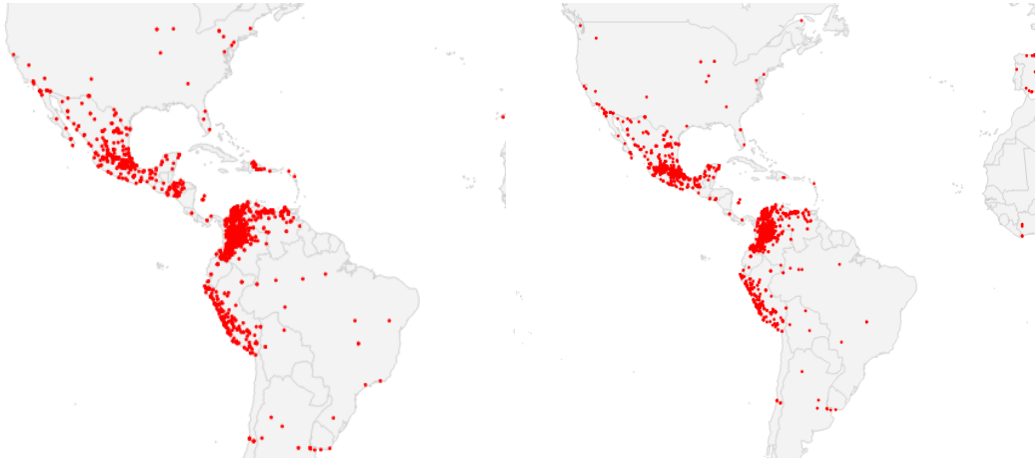
From the maps, we can see, for the company "Adecco", its job postings are across about 3 countries. When you choose “company with temporally”, you can see it will change by colors. These 4 colors represent for 4 seasons. As you can see, you can also choose other attributes like job type and salary and see the changes temporally.



b. Visualize how companies are changing over time based on job postings. Are they growing geographically? Are the entering new domains to make money

This question is to visualize how the distribution of a company changes with the time change. We also show this trend by dividing time into four seasons. This time we extract all the data of all companies and show their trends. After the user clicks the submit button, it will showing the map about all companies changing from first season to fourth season. For each season, we use same color to represent them. The maps from first season to fourth season are looking like followings:

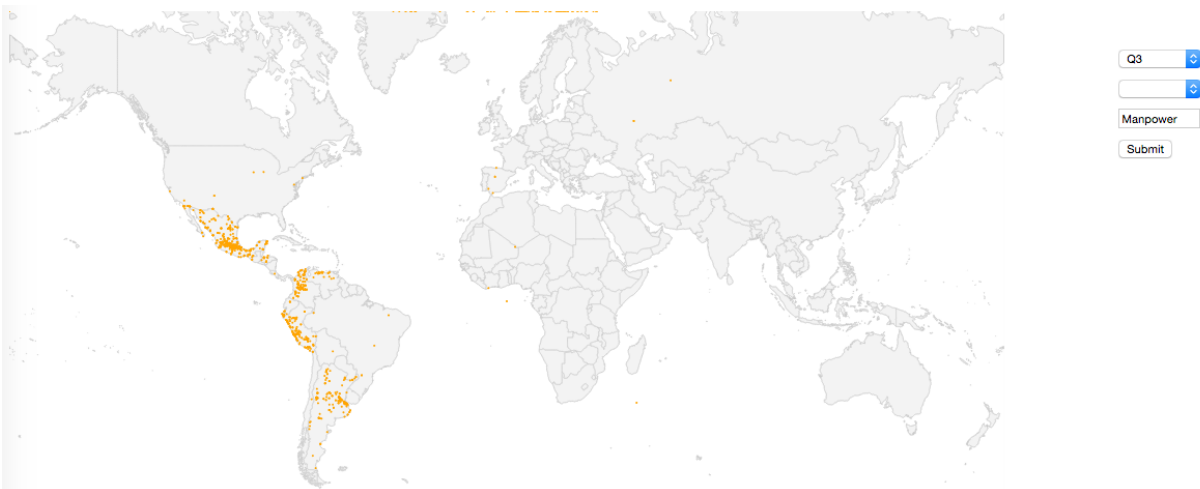




From the maps we can see, from first season to second season, the companies are decreasing geographically, which means they are not entering new domains to make money. But from second season to third and fourth season, they are growing geographically, which means they are entering new domains to make money.

c. Provide a map of corporate presence across South America. Are there territories?

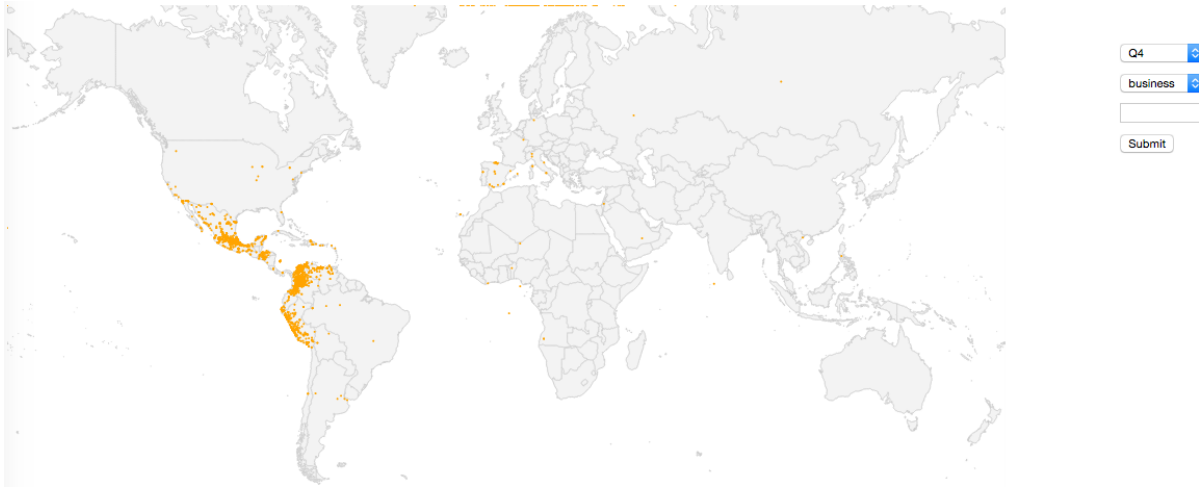
In South America, there are so many companies that have already posted their job data on the website. We primarily decided to fetch the posting data by company. Such that when you type a company, and click the submit button on the below, all the posting data grouped by a certain named company will be shown in color. For example, we want to search posting data for company “Manpower”, the result will be shown up. As we can see, from the map, this company crosses a large area, about 5 to 6 countries.



d. Plot and analyze the seasonal trends of different job categories.

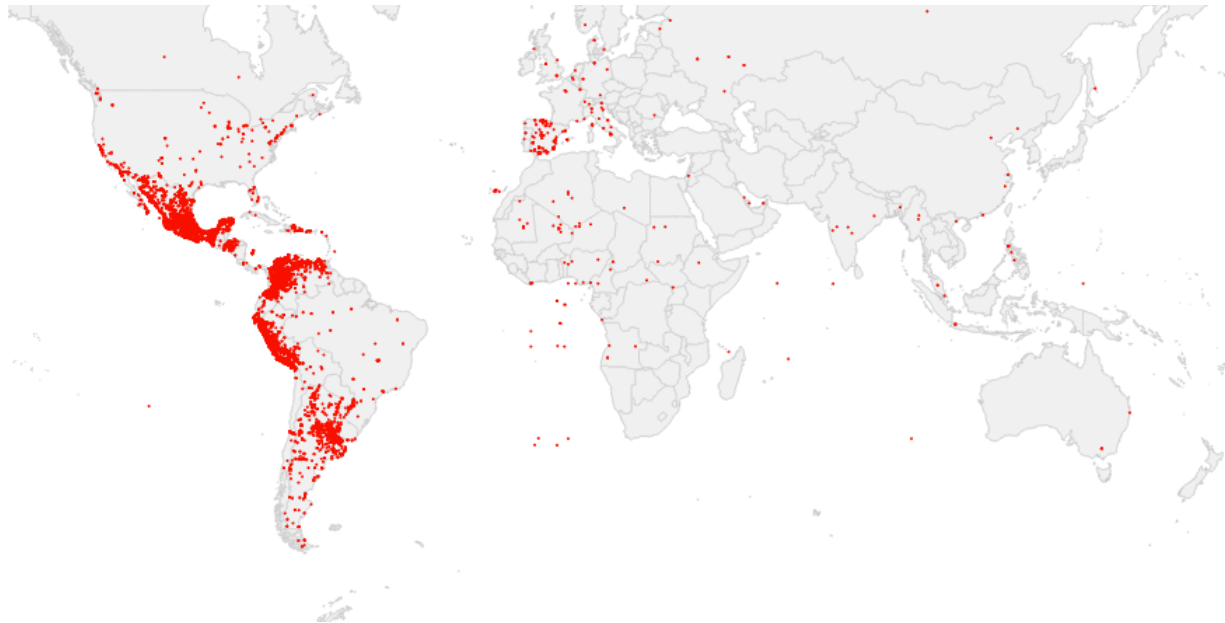
For this question, we decided to split data based on season, like stated in the first question, we define the season into four seasons. We provide a drop down list menu for users to choose the job category like “business” to analyze the result from visual map. Each season will show its result and then delay for seconds. For each season, we will plot the job postings for a certain category. From the quick flash show, we realized that summer has the least posting data, spring has a great volume of data, while fall stays

static as spring, and winter decreases in a small amount. This can be regarded as a trend like: from spring to summer, decrease; from summer to fall, increase; from fall to winter, decrease not that many. It seems that this company likes to hire more people in spring and fall.



7. Recreate the map in PDF:

We also trying to recreate the map in pdf. We got a similar one:



8. Video link:

<https://www.youtube.com/watch?v=rtfVm8EFt9Q&feature=youtu.be>