CS INFO 5100 Project 1 Report

Group Members: Baoyue Wang (bw476), Tanvi Mehta (tmm259), Yiyi Zhai (yz895)

A. A description of the data. Report where you got the data. Describe the variables. If you had to reformat the data or filter it in any way, provide enough details that someone could repeat your results. If you combined multiple datasets, specify how you integrated them. Mention any additional data that you used, such as shape files for maps. Editing is important! You are not required to use every part of the dataset. Selectively choosing a subset can improve usability. Describe any criteria you used for data selection. (10 pts)

Data Retreival:

We used the GitHub archive as a source for all of our datasets. https://www.githubarchive.org/ The necessary data was retrieved using BigQuery which queried the database and generated csv files with the necessary data.

The original dataset consisted of a large amount of information that we did not need for our project. We used the following queries to get all the required data:

1. Get Location of 5000 repositories that use JavaScript as their programming language

```
SELECT repository_language, actor_attributes_location FROM [githubarchive:year.2014] where
repository_language = "JavaScript"
OMIT RECORD IF actor_attributes_location IS NULL OR actor_attributes_location = "" OR
repository_language IS NULL
LIMIT 5000;
```

Similar queries were used to get information about Java and C. This query gave us a csv files with the repository language and the location of the repository. It also filtered out entries with no location or language specified(NULL).

2. Get the number of open source repositories for the ten most popular languages for six-month periods.

Take the dataset from January to June, 2012 as an example. The total number of repositories for each language was found using:

```
SELECT COUNT(*) AS count
FROM (
 TABLE_QUERY([githubarchive:month],
   'REGEXP_MATCH(table_id, r"^20120[1-6]")'
 ));
```

Followed by the number of repositories for each language (37909146 is the total number of repository we got from the above query):

```
SELECT repository_language, COUNT(repository_language) AS count, (COUNT(repository_language) /
37909146) AS percentage
```

```
FROM (
 TABLE_QUERY([githubarchive:month],
     'REGEXP_MATCH(table_id, r"^20120[1-6]")'
 ))
GROUP BY repository_language ORDER BY count DESC;
```

As for data from July to December, 2012, we used the following query:

```
SELECT COUNT(*) AS count
FROM (
 TABLE_QUERY([githubarchive:month],
   'REGEXP_MATCH(table_id, r"^2012((0[7-9])|(1[0-2]))")'
 ));

SELECT repository_language, COUNT(repository_language) AS count, (COUNT(repository_language) /
37804806) AS percentage
FROM (
 TABLE_QUERY([githubarchive:month],
   'REGEXP_MATCH(table_id, r"^2012((0[7-9])|(1[0-2]))")'
 ))
GROUP BY repository_language ORDER BY count DESC;
```

The above queries gave us a dataset with the percentage of each programming language to the total number of repositories.

Filtering:

As for the popularity trend data, it is easy to select the top ten languages and reformat the csv files. We merged the files of different time periods into one csv file and set the time periods 0, 1, 2, 3, 4, 5, which represents 2012.1-6, 2012.7-12, 2013.1-6, 2013.7-12, 2014.1-6, 2014.7-12. But filtering the location data is a little complicate. The location data from the GitHub archive was in the form of names (ex: New York, United States etc.). Furthermore, the location names were not uniform and could span from being countries, cities, regions etc.

We re-formatted the data using a python script that used the geopy library (https://github.com/geopy/geopy) to convert all the locations to latitude and longitude coordinates. This helped make the data uniform and easy to plot on the map.

Additional Data:

We used the world-50m.json file to plot the world map for our visualizations.

B. A description of the mapping from data to visual elements. Describe the scales you used, such as position, color, or shape. Mention any transformations you performed, such as log scales. (10 pts)

For the “Popularity of GitHub Repository Programming Languages from 2012 to 2014” chart, at first we generated the csv file of the percentage of each language in six-month time period. We chose to display the top ten most popular languages. The x axis of the chart represents time periods through 2012 to 2014. The y axis represents represents the percentage of open source collaborators. First, we mapped the data to the position of the chart using linear scale on both axes with circles. Then we used lines to connect contiguous circles of the same languages to

create paths. The reason for why we did not simply use path to map data is that with circles on different time period points user could understand better what the percentage is. We used different colors for different languages. And for paths which are really close to each other, such as C#, Shell and Objective-C, we used colors that are quite different with each other to make it easy to distinguish.

For the “Popularity of Different Programming Language by Location” charts, we generated data in csv format with git commit location (longitude, latitude) and how many times the commit in the place occurred. Then we read the data into an array, extracted the location and mapped it to the world map using d3’s geographic projection. The commits’ locations are displayed by using circles, the radius of which is related to the occurrences. For each language, the linear scale is used to map the domain of minimum to maximum occurrences to the radius range of 7 to 27 (we did some experiments and found out this range looks better). Besides, the color of each language is the same as the color used in the line chart. The opacity of 0.6 is also applied for each color to better show the overlapping.

C. The story. What does your visualization tell us? What was surprising about it? (5 pts)

The visualization represents the popularity of programming languages through time (from 2012 to 2014) and by location (in 2014).

Popularity through time

The visualization shows the trend of programming language usage. There are several interesting observations in the data. JavaScript has been consistently the most popular language since 2012. Ruby on the other hand was the second most popular language in 2012 and then consistently dropped in numbers since then. Java and Python have been the frontrunners and slightly increased in popularity through the years.

Popularity by Location

This visualization tries to find trends in the popularity of languages by location. One of the interesting findings from these visualizations is the popularity of JavaScript in Europe and United States. JavaScript is the most popular language in Europe whereas in United States, the map shows that the East Coast has more users of the language than the West Coast. Its popularity in India is also interesting as it is unparalleled by the other two languages.

Furthermore, while JavaScript users are concentrated on the coats, Java users are spread uniformly in the country. Java also seems to be more popular than JavaScript in South America.

Although C is only the seventh most popular language in the GitHub community, its usage pattern is more diverse and dispersed than the concentrated nature of the other two languages. C is seen to be more popular in South America than JavaScript and more popular in Australia than both JavaScript and Java.