1. Describe project goals and why it is interesting   FINISHED

We plan to use this data to predict which languages are being used, what they're being used for (commercial, fun, etc). The data behind Github is fascinating because it is one the most used version control software out there and allows the public to see the trend of languages throughout the years and see what type of projects people create, and how they collaborate with one another.

Took 5 sec. Last updated by anonymous at March 23 2018, 7:26:04 PM.

2. Describe data collection/source of data, data format, data preprocessing   FINISHED

This dataset gives us info on what languages are used, its files (including their contents) and their size along with info on commits. We will use a combination of the data from bigquery along with using Github's official api using javascript. Bigquery's data is listed as tables, while the api will give us info in the form of an API response as json file. With bigquery, we have to eliminate rows that have a date before the creation of github (which is about 0.04% (10 million) which is most likely an error from importing data to bigQuery) of the total rows. With the api, we eliminate some of the redundant information such as owner of repo (since unless you're searching, you get the repo based off the username).

One of our data collection source was Octokit's rest.js (https://github.com/octokit/rest.js), a JavaScript based API scraper tool adopted by the GitHub team. From there we could access a large amount github data arranging from, user info, repository details, gists, issues, and pull requests from 24 million users.This is an example of a "query" using rest.js:

```
//This specific function gets all the info of JavaFXMusicLibrary repository from user PrL327
octokit.repos.get({
  owner: 'prl327',
  repo: 'JavaFXMusicLibrary'
}).then(({data}) => {
  console.log(data);
});
```

Right now since we can only evaluate data that is public like public organizations and public repos, we had to put a safeguard in which we return 0, which in the end we filter out. We also have to worry about hitting the query limit for the api which is 1000 queries a hour. We actually combined the use of BigQuery and Rest.js in one example. We had a csv file which contained two columns for username and repository, with about 10,000 rows. to easily process that csv file through our query we needed to convert it to a JSON for easy parsing. For this we used csv a node library and it's resepective functions to do the conversion, it produce an object like such:

```
{ "User": "prl327", "Repository": "JavaFXMusicLibrary" }
```
And so on for 10,000 more user/repo combinations.

Took 0 sec. Last updated by anonymous at March 24 2018, 7:22:28 PM.

…Intermediate Report
3. Describe contents of data in detail (write code to analyse and visualize it)   FINISHED

Took 0 sec. Last updated by anonymous at March 23 2018, 9:11:30 AM.

```
%r
repos <- read.csv("c:/users/danny/documents/github/githubdatajs/repos.csv", header = TRUE)
repos2 = repos
repos = createDataFrame(repos)
registerTempTable(repos, "repos")
```
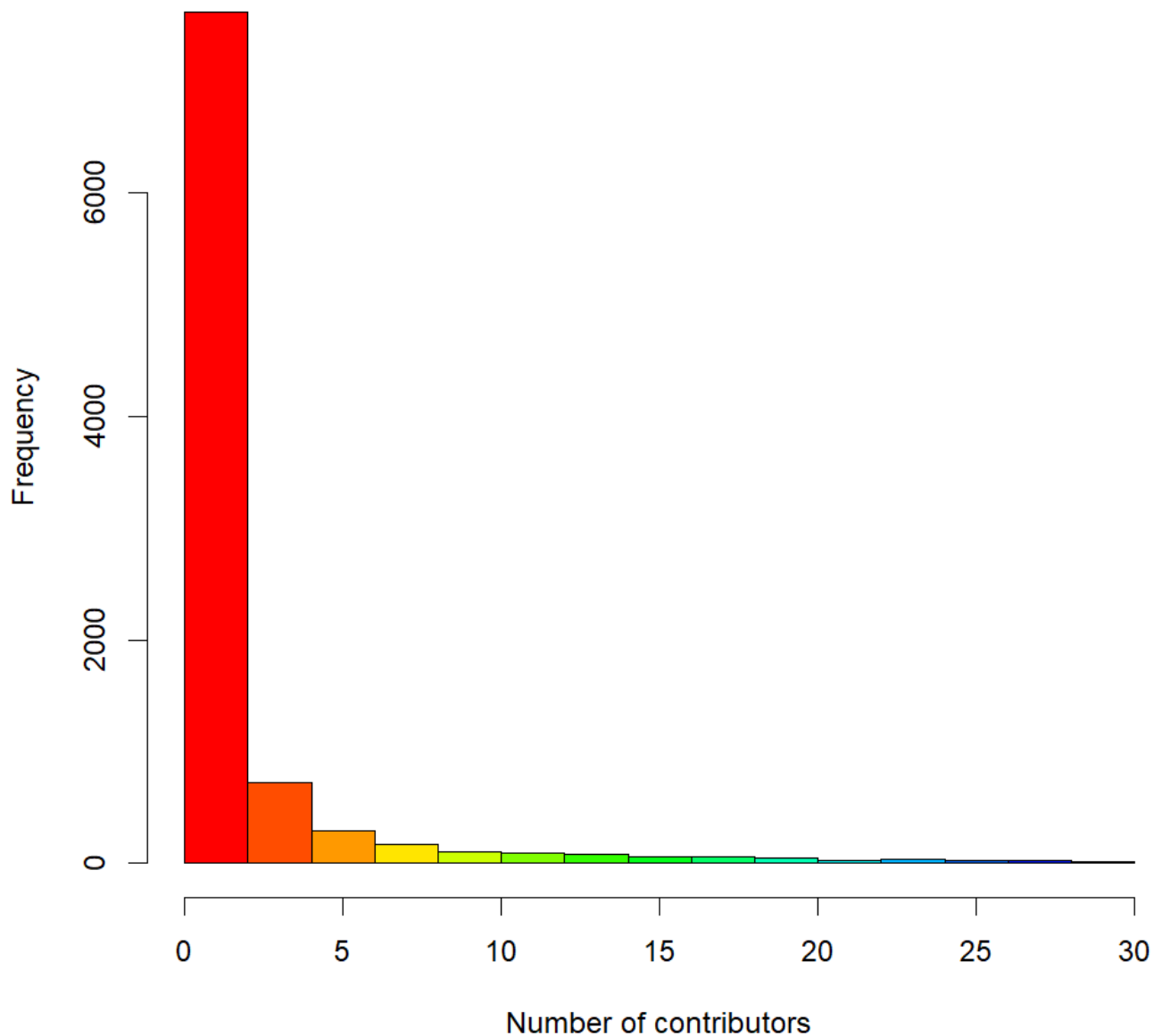FINISHED

Took 1 sec. Last updated by anonymous at March 24 2018, 6:59:41 PM.

```
%r
hist(repos2$contributors[repos2$contributors > 0 & repos2$contributors < 30], 10,
     col = rainbow(20), xlab = "Number of contributors", main = "Frequency of number of contributors\n(for repositories with less than 30 contributors)")
```
FINISHED

Frequency of number of contributors
(for repositories with less than 30 contributors)

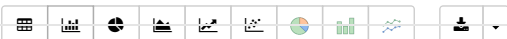Took 0 sec. Last updated by anonymous at March 24 2018, 7:02:43 PM.
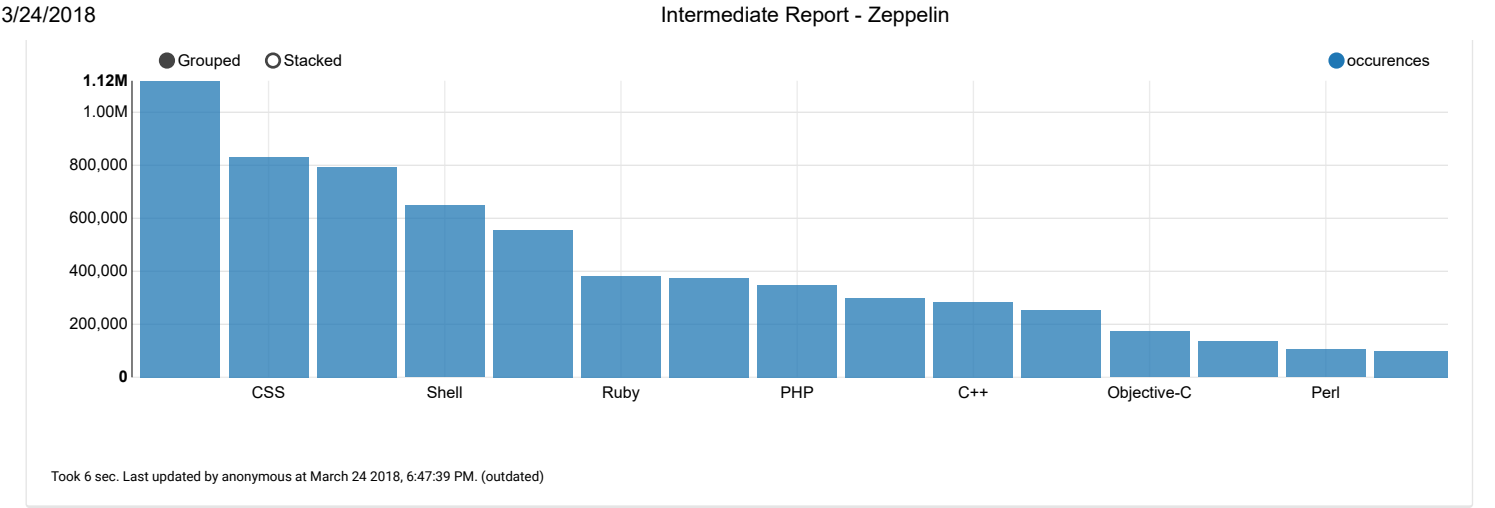
15 Languages used the most

FINISHED

Took 0 sec. Last updated by anonymous at March 24 2018, 6:54:43 PM.

FINISHED

```
%bigquery.sql
SELECT
   name,
   COUNT(*) AS occurences
FROM
   `bigquery-public-data.github_repos.languages` AS l,
   UNNEST(l.LANGUAGE)
GROUP BY
   name
ORDER BY
   occurences DESC
   limit 15
```

settings ▾

Took 6 sec. Last updated by anonymous at March 24 2018, 6:47:39 PM. (outdated)

15 Languages used the least   FINISHED

Took 0 sec. Last updated by anonymous at March 24 2018, 6:54:41 PM.

FINISHED

```
%bigquery.sql
SELECT
  name,
  COUNT(*) AS occurences
FROM
  `bigquery-public-data.github_repos.languages` AS l,
  UNNEST(l.LANGUAGE)
GROUP BY
  name
ORDER BY
  occurences
  limit 15
```
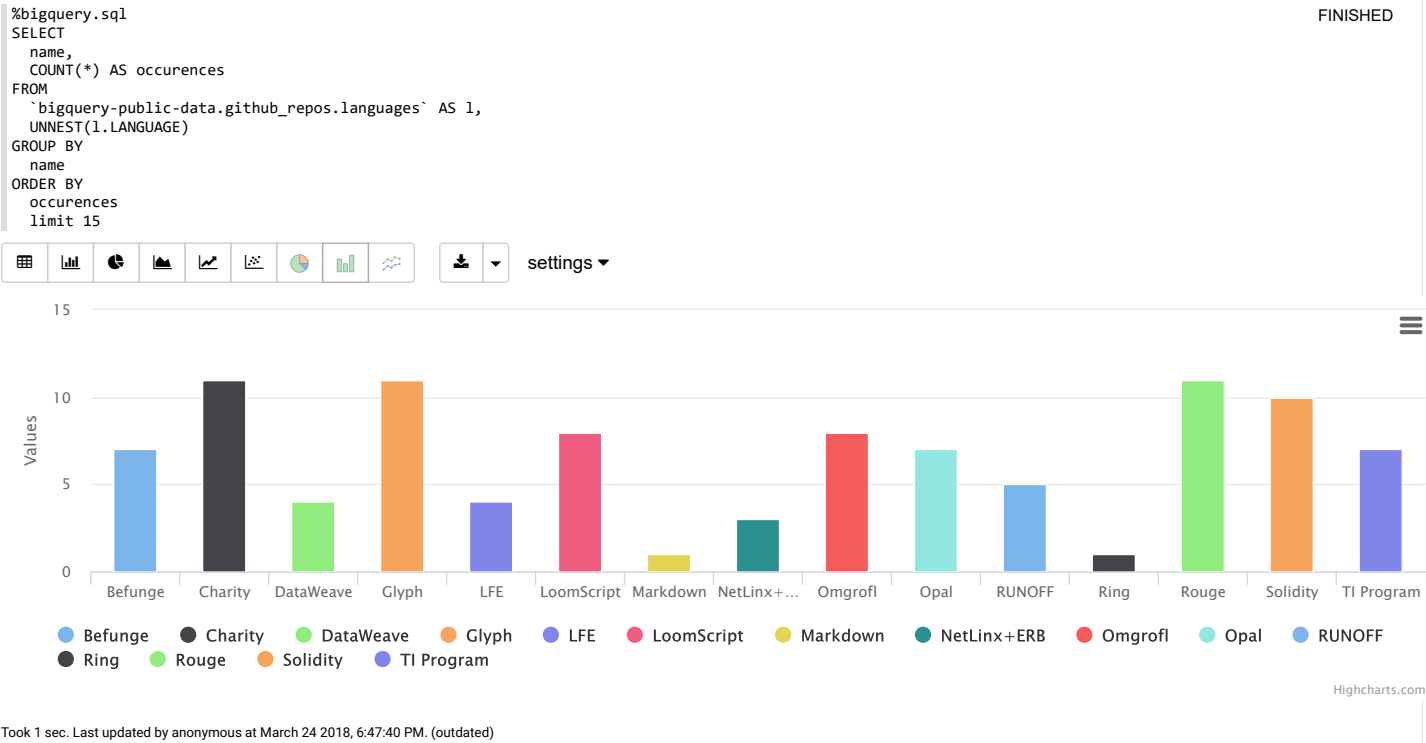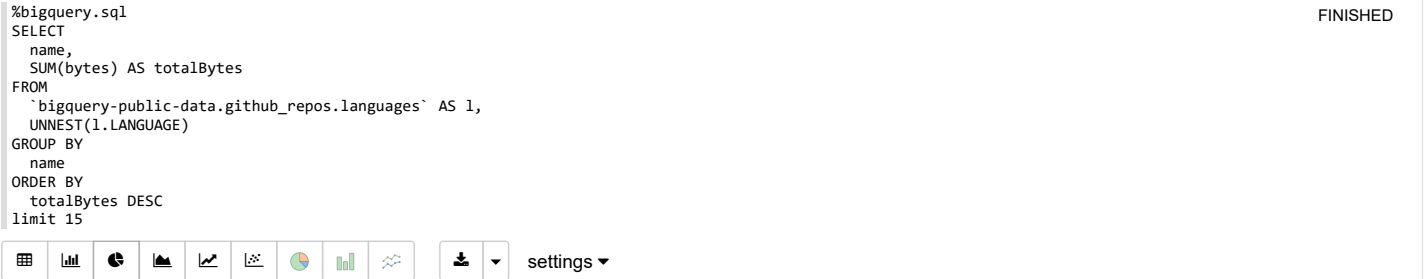

Took 1 sec. Last updated by anonymous at March 24 2018, 6:47:40 PM. (outdated)

Languages using the most bytes   FINISHED

Took 0 sec. Last updated by anonymous at March 24 2018, 6:54:09 PM.

FINISHED

```
%bigquery.sql
SELECT
  name,
  SUM(bytes) AS totalBytes
FROM
  `bigquery-public-data.github_repos.languages` AS l,
  UNNEST(l.LANGUAGE)
GROUP BY
  name
ORDER BY
  totalBytes DESC
limit 15
```

● C   ● JavaScript   ● C++   ● PHP   ● HTML   ● Java   ● Python   ● CSS   ● Assembly   ● C#   ● Go   ● Objective-C   ● Jupyter Notebook   ● Ruby
● TypeScript
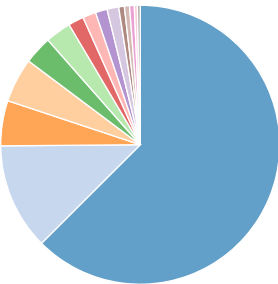
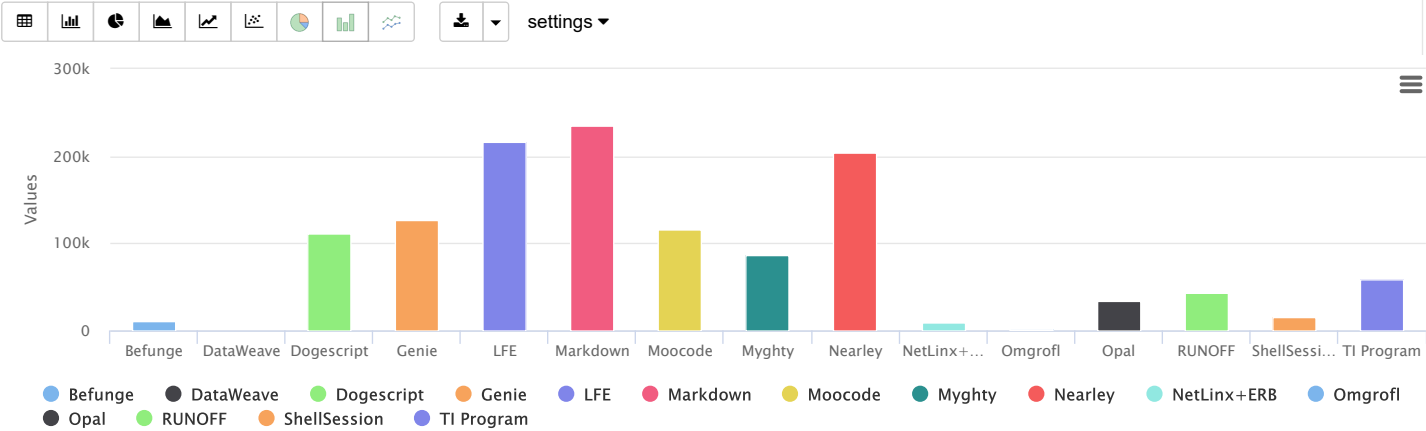Took 1 sec. Last updated by anonymous at March 24 2018, 6:47:41 PM.

Languages using the least amount of bytes

FINISHED

Took 1 sec. Last updated by anonymous at March 24 2018, 6:53:54 PM.

FINISHED

```
%bigquery.sql
SELECT
  name,
  SUM(bytes) AS totalBytes
FROM
  `bigquery-public-data.github_repos.languages` AS l,
  UNNEST(l.LANGUAGE)
GROUP BY
  name
ORDER BY
  totalBytes
limit 15
```

● Befunge   ● DataWeave   ● Dogescript   ● Genie   ● LFE   ● Markdown   ● Moocode   ● Myghty   ● Nearley   ● NetLinx+ERB   ● Omgrofl
● Opal   ● RUNOFF   ● ShellSession   ● TI Program

Highcharts.com

Took 1 sec. Last updated by anonymous at March 24 2018, 6:47:42 PM. (outdated)

4. Describe possible applications of this data, Including your ideas for the next phase

FINISHED

An application for this data is seeing what type of programs use more storage than others allowing a system administrator to give certain users more storage based on their favorite language. One interesting application will be seeing which libraries are commonly used and can go a step further and seeing what are the top 10 libraries for each language for each year. We will be using this data to predict which languages will be expected to be used the most in 2018 and 2019. Since we also get information like commits, we can use this data to see average commits, average time between commits, and see average lifetime of a repository. We are also able to see contents of every file in any public repository, we can use the contents to teach a machine learning algorithm to tell the difference between python and javascript or between java and c++, etc.

Took 4 sec. Last updated by anonymous at March 24 2018, 6:51:22 PM.

```
%md
```

READY