What is the most popular programming language on GitHub (Java or Python)

Gaurav Gaikwad
IT497 Research Methodology
School of Information Technology
Illinois State University
ggaikwa@ilstu.edu

October 22, 2014

1 Introduction

GitHub is one of the most widely used web-based repository hosting service usually to open source projects which provides functionality for distributed revision control and source code management. This Assignment provides a deep insights to differentiate the popularity and usage of Java and Python programming over to GitHub.

Java is object oriented class based programming language designed for low implementation dependencies which makes it possible to run compiled java code on java virtual machine that can reside in any operating system.

On other hand Python is an object oriented procedural high level language designed for concise and readable code constituting large set of libraries, dynamic system and memory management.

2 GitHub Data Gathering

So as to start with analysis we should have GitHub data about the repositories hosted which can be queried against the GitHub timeline archive which provides you with production live data of GitHub.

Google Developer Console provides with Big Query tool which can be used to query the GitHub Archive

https://bigquery.cloud.google.com

```
> #SELECT repository_language AS Repository_Language,
> #   count(repository_language) AS No_of_repo,
> #   SUM(repository_size) AS Total_repo_file_size
> #FROM [githubarchive:github.timeline]
```

```
> #WHERE repository_fork == "false"
> #   AND type == "CreateEvent"
> #   AND PARSE_UTC_USEC(repository_created_at)
> #                >= PARSE_UTC_USEC('2000-01-01 00:00:00')
> #   AND PARSE_UTC_USEC(repository_created_at)
> #                < CURRENT_TIMESTAMP()
> #GROUP BY Repository_Language
> #ORDER BY No_of_repo DESC
> #LIMIT 5;
```

2.1 Understanding the Query

The above represented SQL is a simple query just as in any other query language.

Thus Select has similar impact as what are the column to be derived which here are Repository Language, number of repositories with the same language, and Total size of all the repositories with same language.

FROM clause further identifies data source which here is githubarchive github timeline, the direct timeline github archive.

Third part in query constitutes filtering of data with WHERE clause with multiple filters. The filters we used here are first checking forking capability of repository to be off as this will add another copy of same repository which will not count towards repository count. The second filter is of type where we are checking for initial creation of repository, following with checking all the repositories created from year 2000 till current time.

Fourth and final part to query is the sort function with Group by and Order by clause.

Finally we were just interested to study Java and Python which are in top five or language used on GitHub, thus limiting the result to just top 5 results.

2.2 Data Cleaning

The BigQuery provides you capability of querying against data source with searching and sorting capabilities which provides you with functionality to get cleaned data that you want to analyze finally.

3 Importing

This step will prepare the cleaned data for creating a final analysis demographics where RStudio with LaTeX is used to create a reproducible report. Thus the final cleaned data from BigQuery can be downloaded in CSV file format and then provided to a data frame into R.

This import is done through read csv function and thus imported to a data frame

```
> library(ggplot2)
> git_df <- read.csv("Git_data.csv")
```

4 Data

This is the data from GitHub directly. As the Data cleaning is already done through query itself the data will be our final feed for data analysis.

String function str shows the structure of data frame as a R object.

```
> str(git_df)
```

```
'data.frame':        5 obs. of  3 variables:
 $ Repository_Language : Factor w/ 5 levels "Java","JavaScript",..: 2 5 1 3 4
 $ No_of_repo          : int  1747149 1421545 1355477 924699 843829
 $ Total_repo_file_size: num  1.55e+10 5.96e+09 1.16e+11 1.18e+10 1.90e+10
```

git_df shows the data that is imported to the data frame.

```
> git_df
```

```
  Repository_Language No_of_repo Total_repo_file_size
1          JavaScript    1747149          15451253125
2                Ruby    1421545           5964487818
3                Java    1355477         116000000000
4                 PHP     924699          11822979122
5              Python     843829          19006112673
```

There are three columns Repository Language which is the language of repository, No of repo is the total number of repositories for a given language and Total repo file size which is sum of all the repositories for a given language respectively.

5 Result

Graph is developed by ggplot2 which is a package library for R which is explained as:

ggplot2 is a plotting system for R, based on the grammar of graphics, which tries to take the good parts of base and lattice graphics and none of the bad parts. It takes care of many of the fiddly details that make plotting a hassle (like drawing legends) as well as providing a powerful model of graphics that makes it easy to produce complex multi-layered graphics.

ggplot2 package is installed before calling the function by following command in the previous code chunk: install.packages("ggplot2")

Following installation library is called for installed package such library(ggplot2)

5.1 Explaining ggplot function

The ggplot function firstly gets the data from the data frame, followed by the aesthetic mapping, usually constructed with aes or aes_string. This aesthetic

```
> #Plot Bar Graph for Number of Repositories
> ggplot(git_df, aes(x=Repository_Language, y=No_of_repo,
+                    fill=Repository_Language)) +
+   geom_bar(stat="identity", position= position_dodge()) +
+   scale_fill_manual(values=c("red", "black", "black", "blue", "black")) +
+   ylab("Number of Repositories") + ggtitle("Number of Repositories")
```

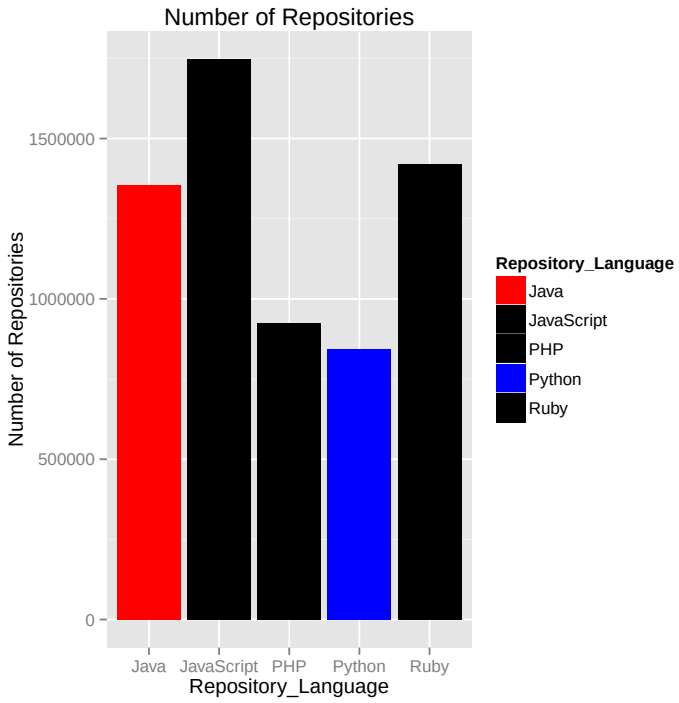

Figure 1: Top 5 Languages for highest Number of Repositories

mapping controls data for x axis, y axis and what data parameter should be filled to produce graph. This is the first parameter for ggplot.

Second parameter describes the graph type which here as geom_bar for bar graph. scale fill manual is for manually filling the bars with your choice of color, ylab defines y axis label and ggtitle sets the title of the plot.

The first graph reflects the total number of repositories for a programming language.

The second graph determines the total size of repositories for a particular language.

```
> #Plot Bar Graph for Total_file_size
> ggplot(git_df, aes(x=Repository_Language,
+                    y=Total_repo_file_size, fill=Repository_Language)) +
+   geom_bar(stat="identity", position= position_dodge()) +
+   scale_fill_manual(values=c("red", "black", "black", "blue", "black")) +
+   ylab("File Size") + ggtitle("Repositories total File Size")
```

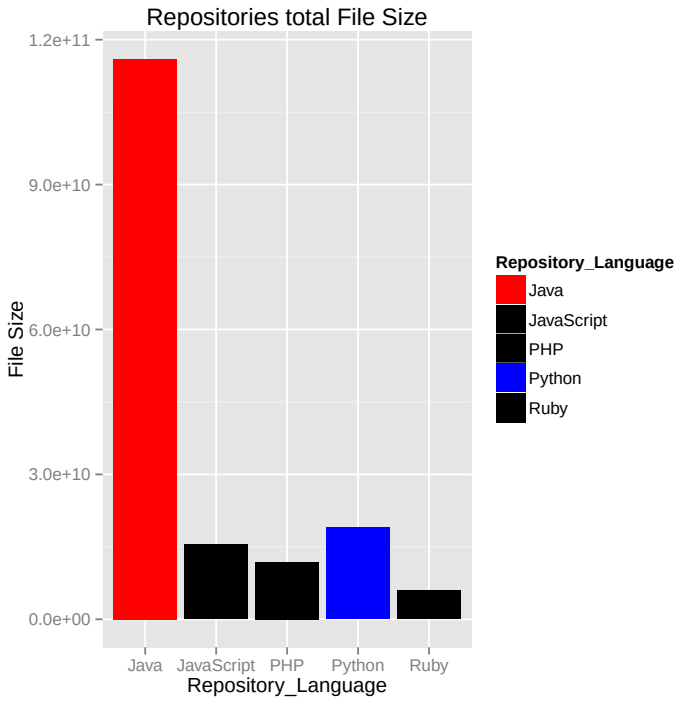

Figure 2: Top 5 Repositories total size by language

6 Data Analysis

First graph shows that GitHub host largest of JavaScript repositories in number followed by Ruby, Java, PHP and then Python. Thus Java wins over Python in the scenario when compared in terms of number of repositories.

On the contrary the second graph for total file size for all projects of a language displays clearly that Java repositories are way larger than any other language repository or even not even close when other summed up together.

Thus finally analyzing both the graphs provides with a broad look of GitHub repositories, and shows that Java is more popular that Python in both the criteria of Number of repositories as well the total file size of all language repositories.

7 Conclusion

As our data analysis report suggest that Java is clearly more popular than Python on GitHub. Lets conclude some features of Java and Python which might be reasons for the popularity of Java.

- Java is platform independent which means Java doesn't need compiler to run on a system rather just a Java virtual machine,
- Java is an Object Oriented Programming Language with static typing over to Python with dynamic typing which. Thus Java with Static typing reduce the risk for undetected errors.
- Java is much faster compared to Python which is the enterprise industry requirement for huge systems and large population of devices.
- Availability of great collection of Open Source libraries for Java with larger user base than Python.
- Android being the current most famous and most widely used smartphone platform globally is also coded in Java which increases the demand and popularity clearly.

8 Reference

1. ggplot2. (n.d.). Retrieved from http://ggplot2.org/
2. Python vs Java: Key Differences. (n.d.). Retrieved from https://www.udemy.com/blog/python-vs-java/
3. Java vs Python - Which Programming Language Should Learn First. (n.d.). Retrieved from http://javarevisited.blogspot.com/2013/11/java-vs-python-which-programming-laungage-to-learn-first.html