# OSEMN Project 2016: Are there more JavaScript or Python repositories in GitHub?

*Latasha Papalal*

*28/11/2016*

## ABSTRACT

**GitHub is a code sharing and publishing service, or it can be explained as a social networking site for programmers. GitHub and Git are interrelated to each other. So, GitHub is a Git repository hosting service, although it adds many of its own features. While Git is a command line tool, GitHub provides a Web-based graphical interface along with access control and several collaboration features for every project. Some basic and very essential terms related to GitHub are command line, repositories, version control, commit , and branch. Here in this project the idea is to find out the number of JavaScript repositories, Python repositories, and compare them. A repository is a directory or storage space where your projects can reside, also known as "repo". It can be local to a folder on your computer, or it can be a storage space on GitHub or another online host. Inside a repository there can be code files, text files, image files, and many different file types.**

## DATA GATHERING

**To get started with the project, the very first step is to obtain the data related to JavaScript and Python repositories. The information on GitHub repositories can be accessed using the GitHub repos link. To extract the data associated with the respective repositories there is a need to pass parameters in the URL in which path, programming language name and results per page are included. Therefore, JavaScript_URL and Python_URL is for JavaScript and Python repositories, respectively.**

```
#The installation of packages is required.
#So that the libraries of the packages can be used throughout the project.
install.packages("jsonlite")
```

```
## Installing package into '/home/ad.ilstu.edu/lna/R/x86_64-pc-linux-gnu-library/3.3'
## (as 'lib' is unspecified)
```

```
install.packages("plyr")
```

```
## Installing package into '/home/ad.ilstu.edu/lna/R/x86_64-pc-linux-gnu-library/3.3'
## (as 'lib' is unspecified)
```

```r
install.packages("ggplot2")
```

```
## Installing package into '/home/ad.ilstu.edu/lna/R/x86_64-pc-linux-gnu-library/3.3'
## (as 'lib' is unspecified)
```

```r
library(jsonlite)
library(ggplot2)
library(plyr)
```

```r
Git_URL <- "https://api.github.com/search/repositories"
JavaScript_URL <- paste0(Git_URL,"?q=language:JavaScript&per_page=100")
Python_URL <- paste0(Git_URL,"?q=language:Python&per_page=100")
```

# DATA CLEANING

**JSON format which is an abbreviation for JavaScript Object Notation is used by GitHb API to provide the data about the repositories. Here, the fromJSON function from the jasonlite package is used to convert the JSON data obtained from the URL to a list. Thus, JS_List and P_List are the JavaScript and Python data lists, respectively. The lists are then converted to dataframes (JS_Df and P_Df) using data.frame. Similarly, Rps_Df is created to have two columns, namely, Programming_Langauge that will have the name of the programming language under it, and Repo_Count that will have the repositories count of both the languages.**

```r
JS_List <- jsonlite::fromJSON(JavaScript_URL)
JS_Df <- data.frame(JS_List)
```

```r
P_List <- jsonlite::fromJSON(Python_URL)
P_Df <- data.frame(P_List)
```

```r
JS_Rps_Count <- JS_Df[1,"total_count"]
P_Rps_Count <- P_Df[1,"total_count"]
Repo_Count <- c(JS_Rps_Count,P_Rps_Count)
Programming_Langauge <- c("JavaScript", "Python")
Rps_Df <- data.frame(Programming_Langauge, Repo_Count)
```

```r
#This will display the two columns with the result.
#Thus, there are more JavaScript repo than Python repo.
Rps_Df
```

```
##   Programming_Langauge Repo_Count
## 1           JavaScript    2904894
## 2               Python    1377454
```

# DATA EXPLORING

**In order to play with the data frame with the intent to obtain some charateristics of the data, class() is used to display the data type of the object, str() is used to display the internal structure of the object and summary() is used to provide statistical information of the object.**

```
class(Rps_Df)
```

```
## [1] "data.frame"
```

```
str(Rps_Df)
```

```
## 'data.frame':    2 obs. of  2 variables:
##  $ Programming_Langauge: Factor w/ 2 levels "JavaScript","Python": 1 2
##  $ Repo_Count          : int  2904894 1377454
```

```
summary(Rps_Df)
```

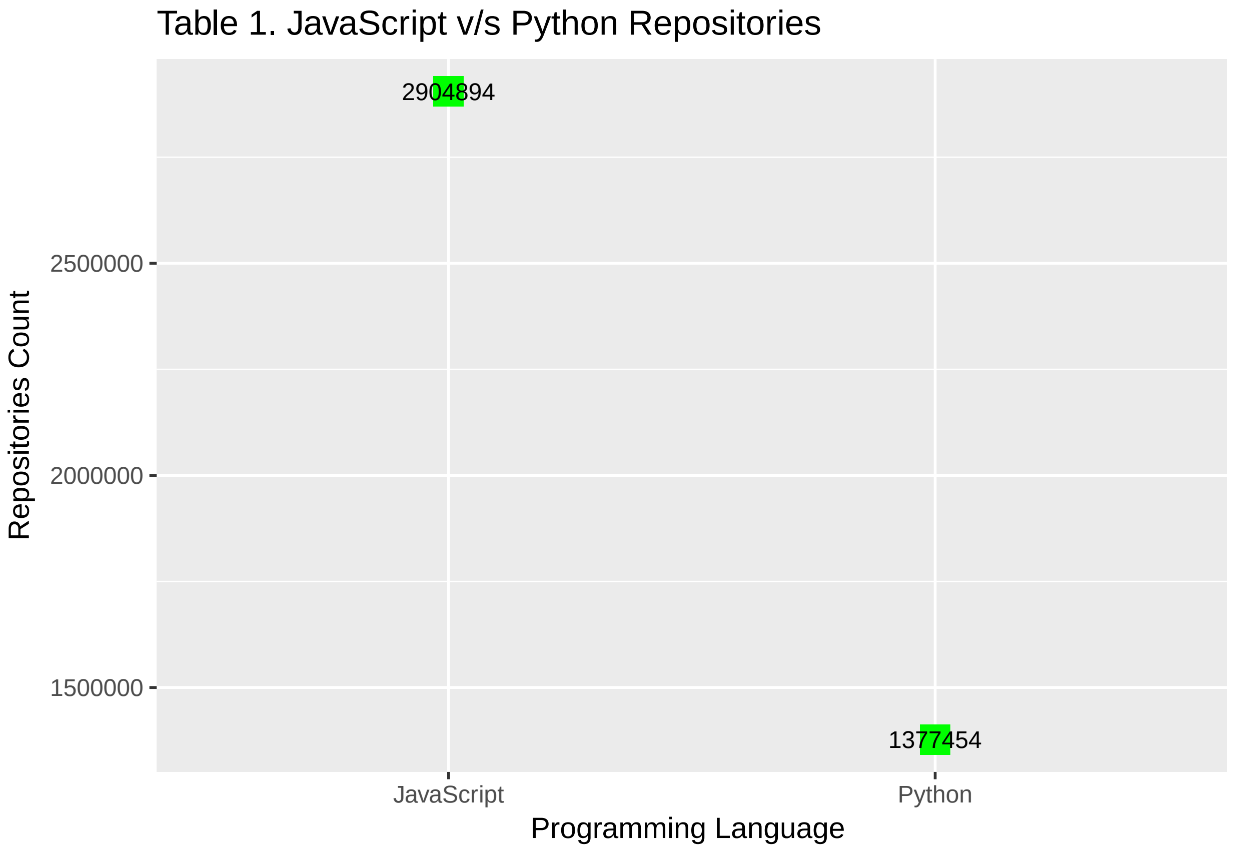
```
##  Programming_Langauge   Repo_Count
##  JavaScript:1         Min.   :1377454
##  Python    :1         1st Qu.:1759314
##                       Median :2141174
##                       Mean   :2141174
##                       3rd Qu.:2523034
##                       Max.   :2904894
```

# DATA GRAPHING

## Scatter plot

**A scatter plot is a valuable approach to envision the relationship between two factors. Like connections, scatter plots are regularly used to make introductory determinations before any factual investigations are directed. This instructional exercise will investigate the courses in which R can be utilized to make scatter plots.**

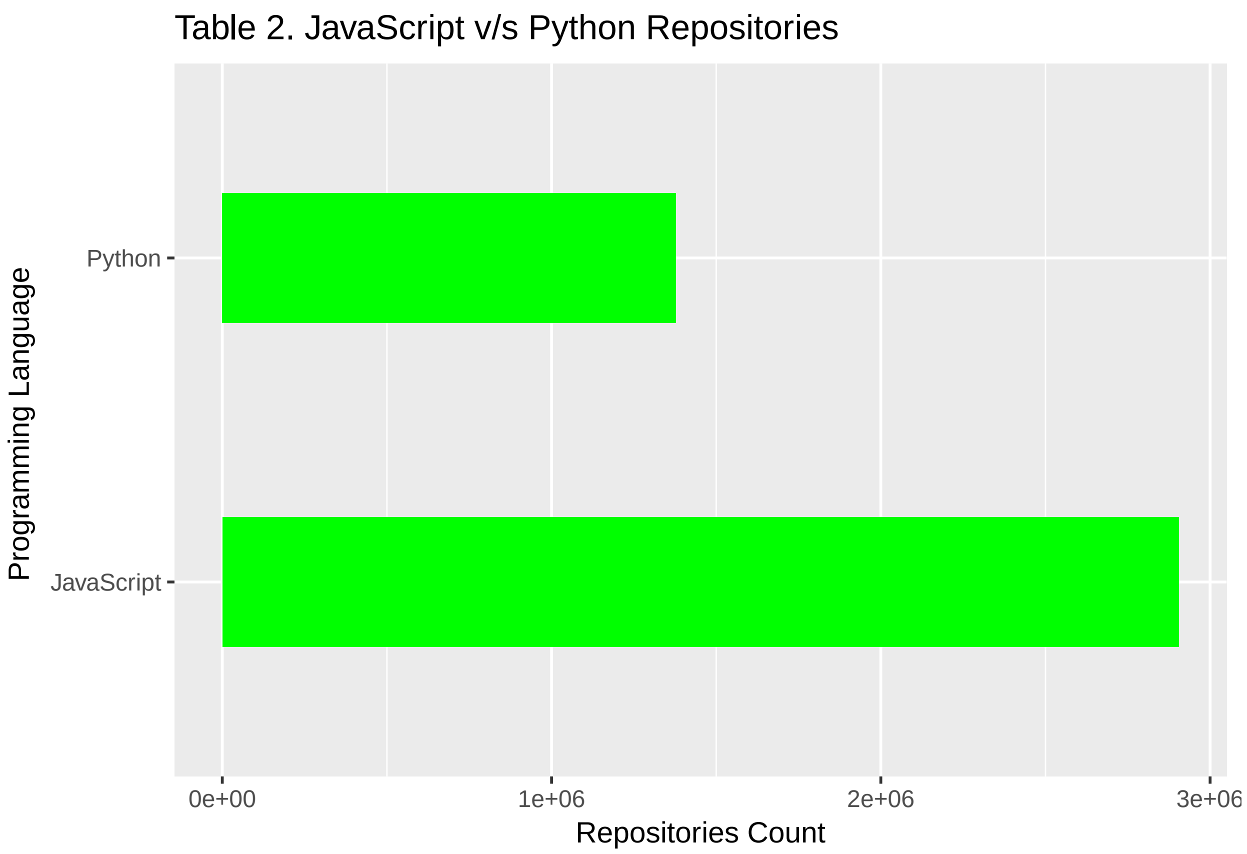
```
library(ggplot2)
#Using ggplot function of ggplot2 package to plot different graphs
ggplot(Rps_Df, aes(Programming_Langauge, Repo_Count, label= as.character(Repo_Count)))+
       geom_point(size=5,shape=15, color="green") +
#Adding text over the end point in the graph using geom_text
       geom_text(size = 3, color= "black") +
  xlab("Programming Language ") + ylab("Repositories Count") + labs(title="Table 1. JavaScript v/s Pyth
```



## Bar plot

**A bar graph of a subjective information test comprises of vertical parallel bars that demonstrates the recurrence conveyance graphically.**

```
library(ggplot2)
ggplot(data=Rps_Df, aes(x=Programming_Langauge, y=Repo_Count)) +
  geom_bar(fill="green",width=.4, stat="identity", position = "dodge") +
xlab("Programming Language ") + ylab("Repositories Count") + labs(title="Table 2. JavaScript v/s Python
  coord_flip()
```

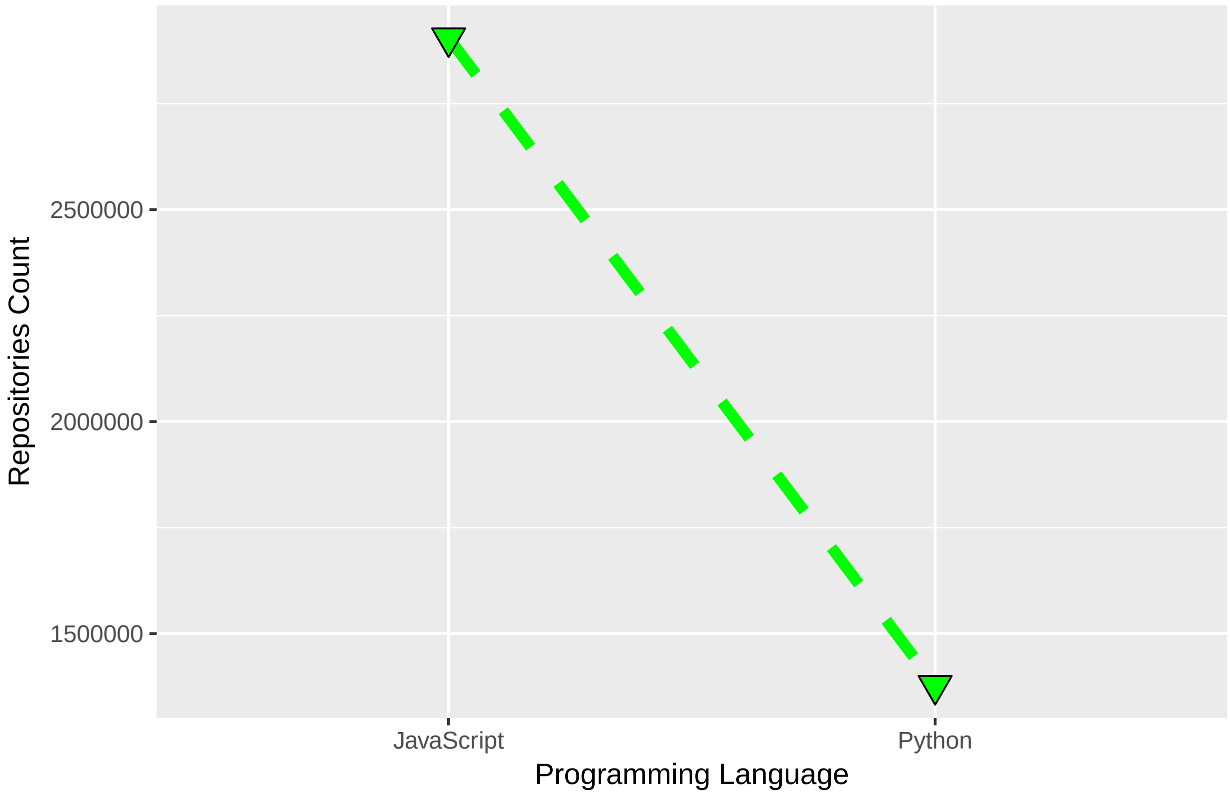


## Line plot

A line chart is a diagram that associates a progression of points by drawing line sections between them. These focuses are requested in one of their coordinate (more often than not the x-coordinate) esteem. Line charts are typically utilized as a part of distinguishing the patterns in information.

```
library(ggplot2)
ggplot(Rps_Df, aes(x=Programming_Langauge, y=Repo_Count, group=5)) +
  geom_line(colour="Green",linetype="dashed", size=2) +
#Adding point over the line in the graph using geom_text
  geom_point(colour="black",shape=25, size=4,fill="green")+
xlab("Programming Language") + ylab("Repositories Count") + labs(title="Table 3. JavaScript v/s Python
```

Table 3. JavaScript v/s Python Repositories



## RESULT

**The resulting repositories count and plotted graphs of both the languages clearly states that, there are more JavaScript repositories than Python repositories. In the same way, popularity of different languages among the developers can also be evaluted based on the number of repositories created for them.**