Github Repositories for Java and JavaScript

GitHub is a web-based Git repository hosting service, which offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features. Unlike Git, which is strictly a command-line tool, GitHub provides a web-based graphical interface and desktop as well as mobile integration. It also provides access control and several collaboration features such as wikis, task management, and bug tracking and feature requests for every project.

GitHub offers both paid plans for private repositories and free accounts, which are usually used to host open-source software projects. As of 2015, GitHub reports having over 9 million users and over 21.1 million repositories, making it the largest code hoster in the world.

Projects on GitHub can be accessed and manipulated using the standard git command-line interface and all of the standard git commands work with it. GitHub also allows registered and non-registered users to browse public repositories on the site. Multiple desktop clients and git plugins have also been created by GitHub and other third parties which integrate with the platform.

The site provides social networking-like functions such as feeds, followers, wikis (using wiki software called gollum) and a social network graph to display how developers work on their versions ("forks") of a repository and which fork (and branch within that fork) is newest.

A user must create an account in order to contribute content to the site, but public repositories can be browsed and downloaded by anyone. With a registered user account, users are able to discuss, manage, create repositories, submit contributions to others' repositories, and review changes to code.

For this assignemt, I have to calculate the number of repository of Java and JavaScript present on the github and plot graphs based on the outcome of results.

```
mydata <- read.csv(file="result_javaScript.csv", sep=",")
abc <- na.omit(mydata$total_count)
#abc1<-gsub("[:print:]","",mydata)
#abc1<-c(abc1)
#abc1<-  abc1[-(2:30),]
str(abc)

##  atomic [1:1] 59594
##  - attr(*, "na.action")=Class 'omit'  int [1:29] 2 3 4 5 6 7 8 9 10 11 ...

class(abc)

## [1] "integer"

summary(abc)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   59590   59590   59590   59590   59590   59590
```

```r
mydata1 <- read.csv(file="result_Java.csv", sep=",")
abc2 <- na.omit(mydata1$total_count)
#abc3<-gsub("[:print:]","",mydata1)
#abc3<-c(abc3)
#abc3<- abc3[-(2:30),]
str(abc2)
```

```
##  atomic [1:1] 49897
##  - attr(*, "na.action")=Class 'omit'  int [1:29] 2 3 4 5 6 7 8 9 10 11 ...
```

```r
class(abc2)
```

```
## [1] "integer"
```

```r
summary(abc2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   49900   49900   49900   49900   49900   49900
```

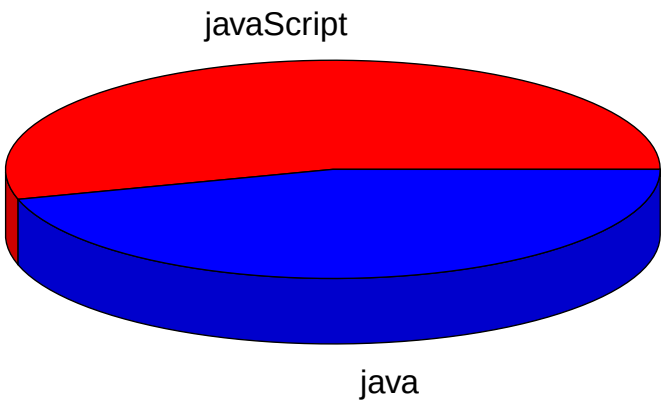
```r
df1<-(c("javaScript","java"))
df2<-(c(abc, abc2))

install.packages('plotrix')
```

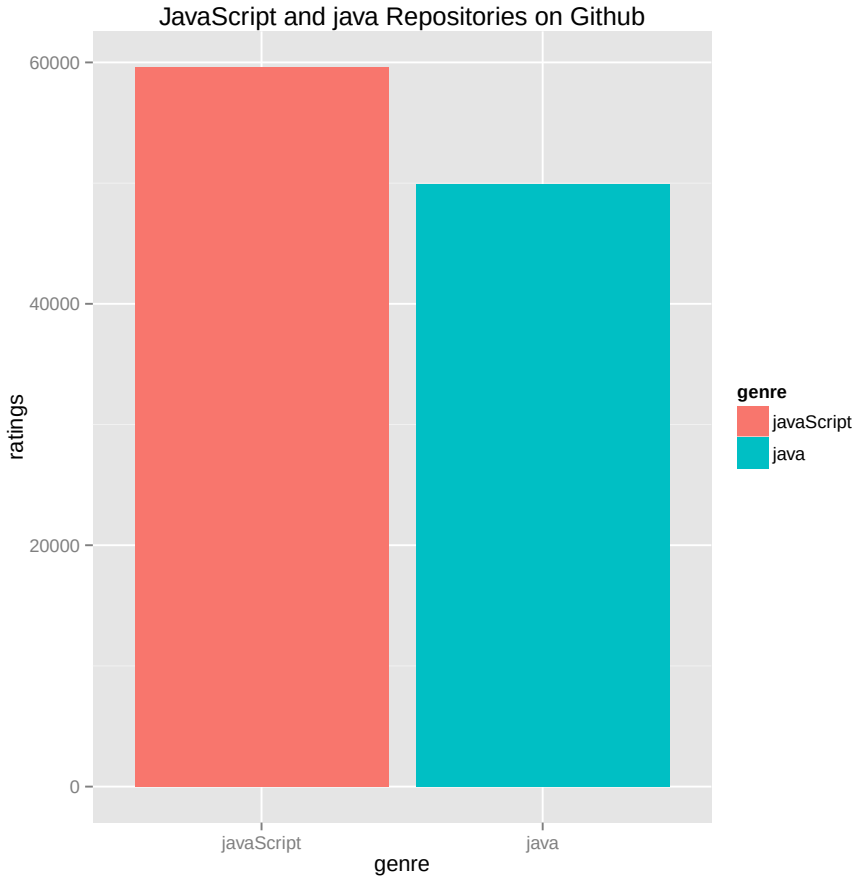
```
## Installing package into '/home/hkhuran/R/x86_64-pc-linux-gnu-library/3.1'
## (as 'lib' is unspecified)
## Error in contrib.url(repos, type):  trying to use CRAN without setting
a mirror
```

```r
library(plotrix)
slices<-c(abc,abc2)
aaa<-c("javaScript", "java")
colors<-c("red","blue")
pie3D(slices, labels=aaa, col=colors, main= "This is Awesome")

library(ggplot2)
```




```
df3<-data.frame(genre=factor(df1, levels=df1), ratings=df2)
ggplot(data=df3, aes(x=genre, y=ratings, fill=genre)) + geom_bar(stat="identity")+
ggtitle("JavaScript and java Repositories on Github")
```



The two graphs are used to draw the output for comparasion of repositories of JavaScript and Java on Github.

In one representation i have used Piechart to display the result and the other ggplot to draw the graph.

This was an AWESOME Assignment!!!!!! :-) :-/ :-O :-D :-S