OSEMN Project Using GitHub API

Karishma Thadishetty

Research Methodologies - Fall 2016

Illinois State University

kthadis@ilstu.edu

Research Question

Are there more GitHub R repositories created in Febrary than January?

Introduction

GitHub is a web-based Git repository hosting service. It offers all of the distributed version control and source code management functionality of Git as well as adding its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project. GitHub offers both plans for private repositories, and free accounts which are commonly used to host open-source software projects. It is the largest host of source code in the world.

Development of the GitHub platform began on 1 October 2007. The site was launched in April 2008 by Tom Preston-Werner, Chris Wanstrath, and PJ Hyett after it had been made available for a few months prior as a beta release. 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 that integrate with the platform.

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. The software that runs GitHub was written using Ruby on Rails and Erlang by GitHub, Inc. developers Chris Wanstrath PJ Hyett, and Tom Preston-Werner. GitHub Enterprise is similar to GitHub's public service but is designed for use by large-scale enterprise software development teams where the enterprise wishes to host their repositories behind a corporate firewall.

Connecting to GitHub API

Below are the packages required.

```
library(bitops)
library(ggplot2)
library(RCurl)
library(rjson)
library(stringr)
```

The following client id and client secret keys are used to connect to GitHub API.

```
client.id <- "XXXXXXXXXXXXXXXX"
client.secret <- "XXXXXXXXXXXXXXX"
```

Obtaining Data

The below function is for defining a generic function for the GitHub API. the arguements for the function would be search type, query, and page number (optional).

```
search <- function(search_type, q, page=1){
    Sys.sleep(10)

    fromJSON(getURL(paste0(
        "https://api.github.com/search/",
        search_type,
        "?client_id=", client.id,
        "&client_secret=", client.secret,
        "&q=", curlEscape(q),
        "&page=", page
    ), httpheader=c("User-Agent"= "BCable")))
}
```

The below function is for fetching total number of results by language, year, and month.

```
search_month <- function(language, year, month){
    data <- NULL
    for(start_month in month){
        last_month <- as.integer(start_month)+1
        last_date <- 1

        if(last_month > 12){
            last_month <- 12
            last_date <- 31
        }

        start_month <- str_pad(start_month, 2, "left", "0")
        last_month <- str_pad(last_month, 2, "left", "0")
        last_date <- str_pad(last_date, 2, "left", "0")

        output <- search("repositories", paste0(
            "language:", language,
            ' created:"',
            year, "-", start_month, "-01 .. ",
            year, "-", last_month, "-", last_date,
            '"'
        ))$total_count

        data <- c(data, output)
    }
    data
}
```

Scrubbing Data

The below command is used to search for R Repositories in 2016 for the month of January and February.

```
count_2016 <- search_month("R", 2016, 1:11)
```

The below command creates a data frame to combine the counts of months.

```
aggregate_data <- data.frame(
    Value=count_2016,
    date=as.POSIXlt(paste0(
        c(
            rep(2016, length(count_2016))
        ), "-", str_pad(c(
            seq(1, length(count_2016))
        ), 2, "left", "0"), "-01"
    ), format="%Y-%m-%d")
)
```

The following command sorts the data by date.

```
agg_data <- aggregate_data[order(as.numeric(aggregate_data$date)),]
```

The below command is to get the month name,year from the POSIXlt date and convert them to a factor and to finally strip the the POSIXlt date.

```
agg_data$Month <- strftime(agg_data$date, format="%B")
agg_data$Month <- factor(agg_data$Month, levels=unique(agg_data$Month))
agg_data$Year <- strftime(agg_data$date, format="%Y")
agg_data$Year <- factor(agg_data$Year, levels=unique(agg_data$Year))
agg_data$date <- NULL
```

Exploring Data

```
class(agg_data)
```

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

```
str(agg_data)
```

```
## 'data.frame':    11 obs. of  3 variables:
##  $ Value: num  5399 5692 5635 5260 5383 ...
##  $ Month: Factor w/ 11 levels "January","February",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Year : Factor w/ 1 level "2016": 1 1 1 1 1 1 1 1 1 1 ...
```

```
summary(agg_data)
```

```
##      Value           Month     Year
##  Min.   :4797   January :1   2016:11
##  1st Qu.:5224   February:1
##  Median :5399   March   :1
##  Mean   :5472   April   :1
##  3rd Qu.:5706   May     :1
##  Max.   :6179   June    :1
##                 (Other) :5
```

Summarizing Data

The data has three columns namely month, year and the total number of R repositories for the corresponding months.

```
agg_data
```

```
##    Value     Month Year
## 1   5399   January 2016
## 2   5692  February 2016
## 3   5635     March 2016
## 4   5260     April 2016
## 5   5383       May 2016
## 6   5027      June 2016
## 7   4797      July 2016
## 8   5189    August 2016
## 9   5720 September 2016
## 10  6179   October 2016
## 11  5910  November 2016
```

Analyzing Data

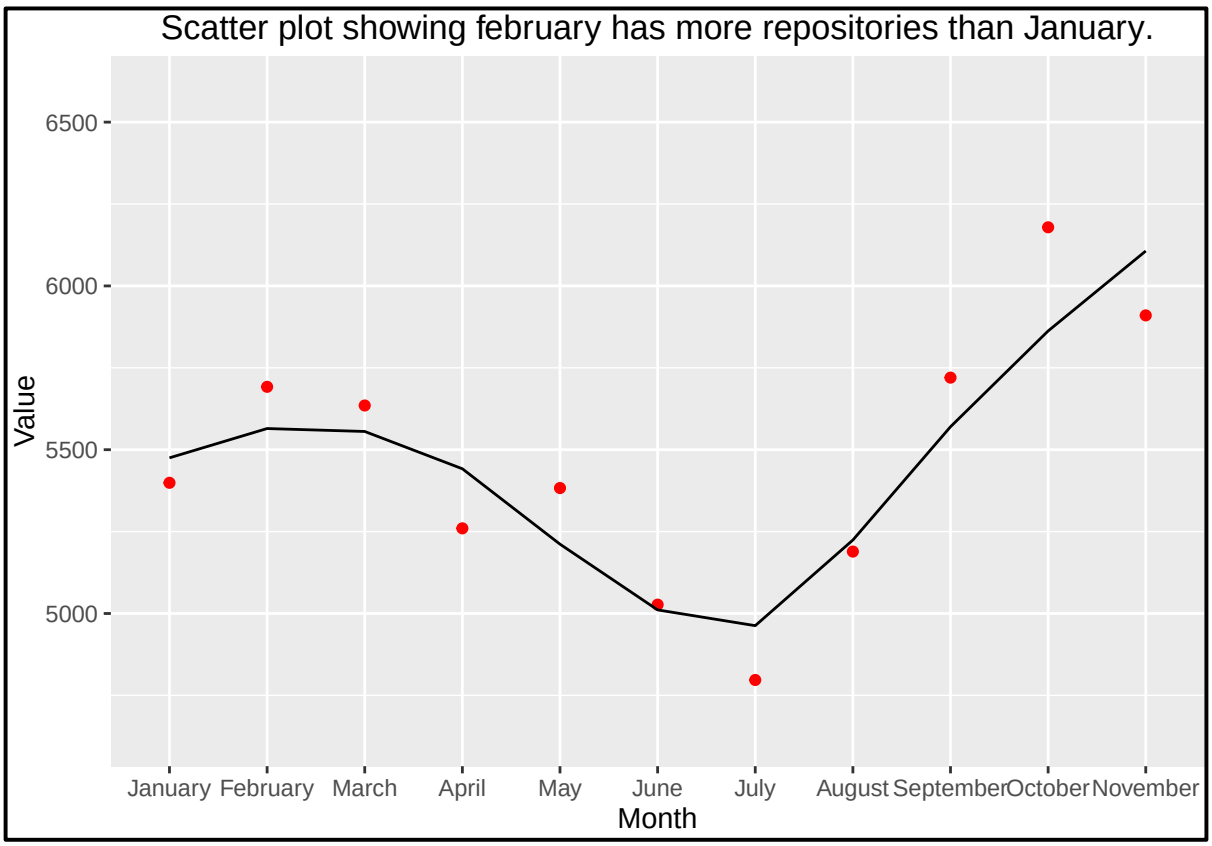
The number of repositories created in January is 5399 whereas it is 5690 in February. Hence, February has more repositories than January in 2016.

Modelling Data

Scatter plot

The scatter plot is drawn having months on X-axis and the count of repositories on Y-axis.

```
library(ggplot2)
library(grid)
library(gridExtra)
g<-ggplot(agg_data, aes(x = Month, y = Value, group = Year)) + geom_point(color = "red") +
  geom_line(stat = "smooth", method = "loess") +
  ggtitle("Scatter plot showing february has more repositories than January.")
grid.arrange(g,ncol=1)
grid.rect(width = .98, height = .98, gp = gpar(lwd = 2, col = "black", fill = NA))
```



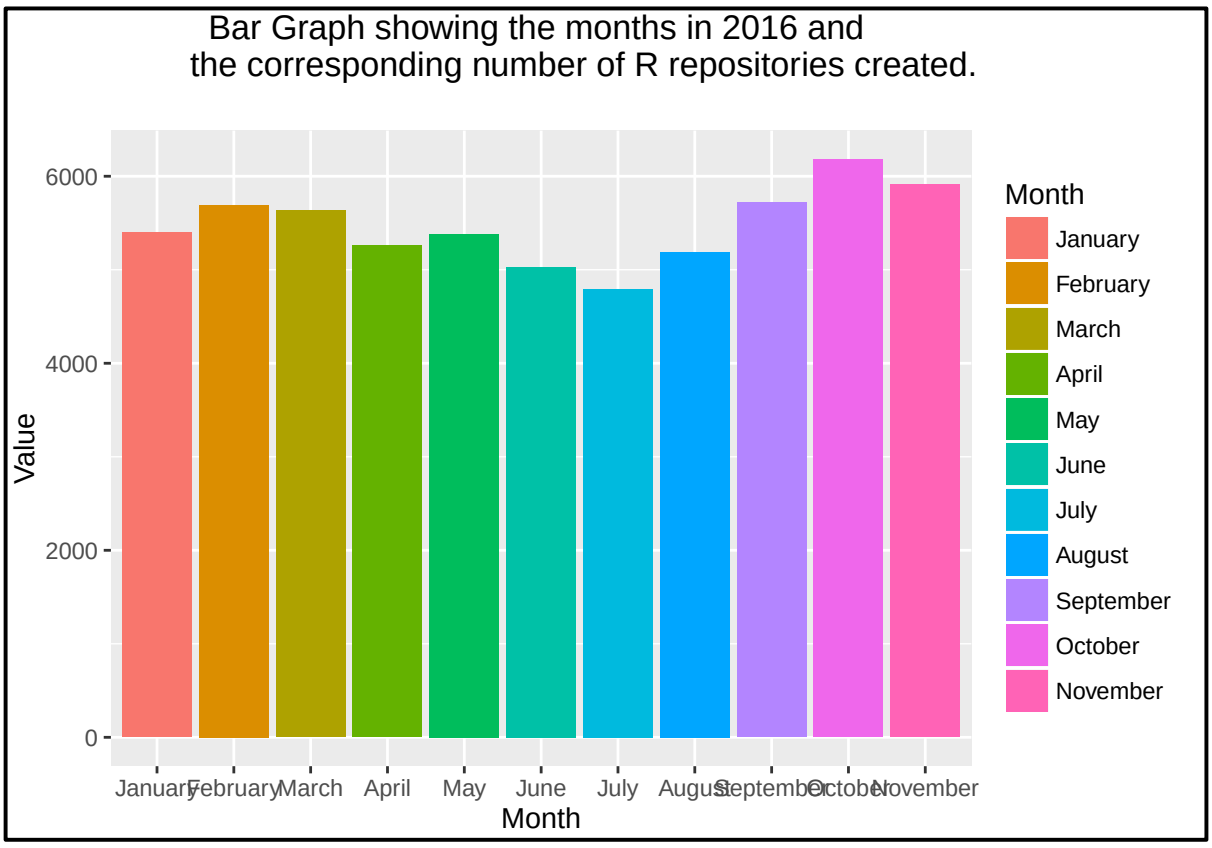
It shows that there is a moderate positive correlation in data.

Data Results

Bar Graph

The below bar graph is drawn considering Months on the horizontal axis and their corresponding number of repositories on the Y-axis.

```
g1 <- ggplot(agg_data, aes(x = Month, y = Value, fill = Month)) +
  geom_bar(stat = "identity") +
  ggtitle("Bar Graph showing the months in 2016 and
          the corresponding number of R repositories created. \n")
grid.arrange(g1,ncol=1)
grid.rect(width = .98, height = .98, gp = gpar(lwd = 2, col = "black", fill = NA))
```

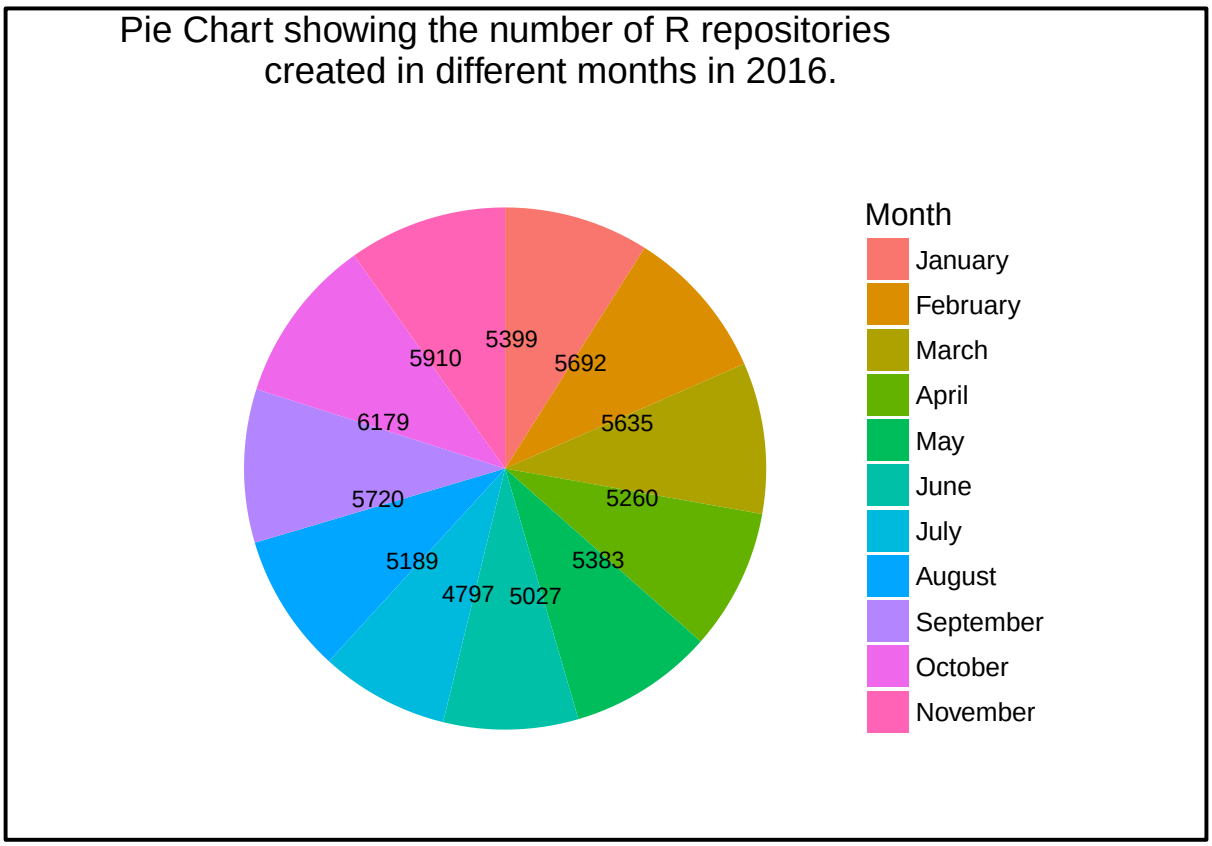


It is evident from the chart, that the number of repositories created in February is more than January.

Pie Chart

The below pie chart is drawn considering the data frame, having months and count of repositories(Y-axis).

```
library(scales)
blank_theme <- theme_minimal()+
  theme(
  axis.title.x = element_blank(),
  axis.title.y = element_blank(),
  panel.border = element_blank(),
  panel.grid=element_blank(),
  axis.ticks = element_blank()
  )
g2 <- ggplot(agg_data, aes(x = "", y= Value, fill = Month)) +
  geom_bar(stat="identity",width = 1) +
  coord_polar("y") +  blank_theme +
  theme(axis.text.x=element_blank()) +
  geom_text(aes(y = Value/11 + c(0, cumsum(Value)[-length(Value)]),
                label = Value), size=3) +
  ggtitle("Pie Chart showing the number of R repositories
          created in different months in 2016. \n")
grid.arrange(g2,ncol=1)
grid.rect(width = .98, height = .98, gp = gpar(lwd = 2, col = "black", fill = NA))
```



The graph displays the count and the division based on Months. We can infer from the chart that February has more repositories than January.

Parametric Test

The t-test is probably the most commonly used Statistical Data Analysis procedure for hypothesis testing. There are several kinds of t-tests, but the most common is the "two-sample t-test" also known as the "Student's t-test" or the "independent samples t-test".

The count of repositories in January and February is considered and T-test is performed on the data.

```
Jan_feb_data
```

```
##   count    month year
## 1  5399  January 2016
## 2  5692 February 2016
```

```
t.test(Jan_feb_data$count)
```

```
##
##  One Sample t-test
##
## data:  Jan_feb_data$count
## t = 37.853, df = 1, p-value = 0.01681
## alternative hypothesis: true mean is not equal to 0
```

```
## 95 percent confidence interval:
##  3684.041 7406.959
## sample estimates:
## mean of x
##    5545.5
```

The null hypothesis is the number of repositories created in February has no correlation with January. The obtained p-value from the t-test is less than 0.05 which implies that the null hypothesis is false and February has more repositories than January.