

OSEMN Assignment - Ruby vs Python Repositories on Github

Chris Higgins

April 19th, 2015

1 Introduction

In today's world, there are a ton of programming languages that can be utilized by a range of people, from novice computer users to the everyday hardcore programmer. You can write in NASM assembly to directly interact with your computer's processor and memory, you can write in HTML to create your website, and you can use Alice to introduce children to programming. The options you have are endless.

With programming projects it has become a recent trend to involve everything in a source code repository with a version control system. This offers many benefits, mainly source code history and easier collaboration with other programmers. Many version control systems include CVS, Sourcefire, and Git. One of the biggest hosted source code repository vendors out there is GitHub. They utilize Git to incorporate code history and collaboration, and offer wikis for documentation, issue tracking for various enhancements or bugs, and forking which allows you to work on your own copy of someone else's repository. It has become the de facto standard in source code project management.

2 Ruby vs Python

Two languages that are always brought up together are Ruby and Python. The two are both very powerful but also very simple, always highly recommended to new and experienced programmers alike.

2.1 Ruby

First we will take a look at Ruby, a language becoming popular for websites with frameworks such as Rails. The following description is taken from Ruby's page on Wikipedia:

Ruby is a dynamic, reflective, object-oriented, general-purpose programming language. It was designed and developed in the mid-1990s by Yukihiro "Matz" Matsumoto in Japan.

According to its authors, Ruby was influenced by Perl, Smalltalk, Eiffel, Ada, and Lisp. It supports multiple programming paradigms, including functional, object-oriented, and imperative. It also has a dynamic type system and automatic memory management.

As I said, Ruby has become very popular with the inclusion of the Rails framework, thusly called Ruby on Rails. There are a ton of major websites that use Ruby on Rails including Twitter, Bloomberg, and Groupon.

2.2 Python

Next we will look at Python. It is very popular and used a lot to teach younger kids the basics of computer programming. The following description is taken from Python's page on Wikipedia:

Python is a widely used general-purpose, high-level programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C++ or Java. The language provides constructs intended to enable clear programs on both a small and large scale.

Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.

In my personal experience I tend to use Python more and more. It is simple to use and very powerful. I do a lot of security research and I can create a proof of concept exploit program using Python more simply than I can using Ruby.

3 GitHub Repository Data Results

```
# There is a warning that comes up importing jsonlite. Cannot  
figure out how to get it to go away or jsut not show up in the  
compiled PDF.  
library(jsonlite)
```

```
Attaching package: 'jsonlite'
```

```
The following object is masked from 'package:utils':
```

```
View
```

```
library(ggplot2)
```

```
# API Calls from Github
```

```

ruby_api_url <- "https://api.github.com/search/repositories?q=
  language:ruby"
python_api_url <- "https://api.github.com/search/repositories?q=
  language:python"

# Grab the API results, JSON format
ruby_json <- jsonlite::fromJSON(ruby_api_url)
python_json <- jsonlite::fromJSON(python_api_url)

# From the JSON data, get the total repo count
ruby_total <- ruby_json$total_count
python_total <- python_json$total_count

```

From using the GitHub API, we are able to deduce that there are a total of **767511** Ruby repositories and **633761** Python repositories. I fathom that there are more Ruby repositories than Python because of how simply typed Ruby is. It mimics English so well that you can basically "speak" your code in plain English.

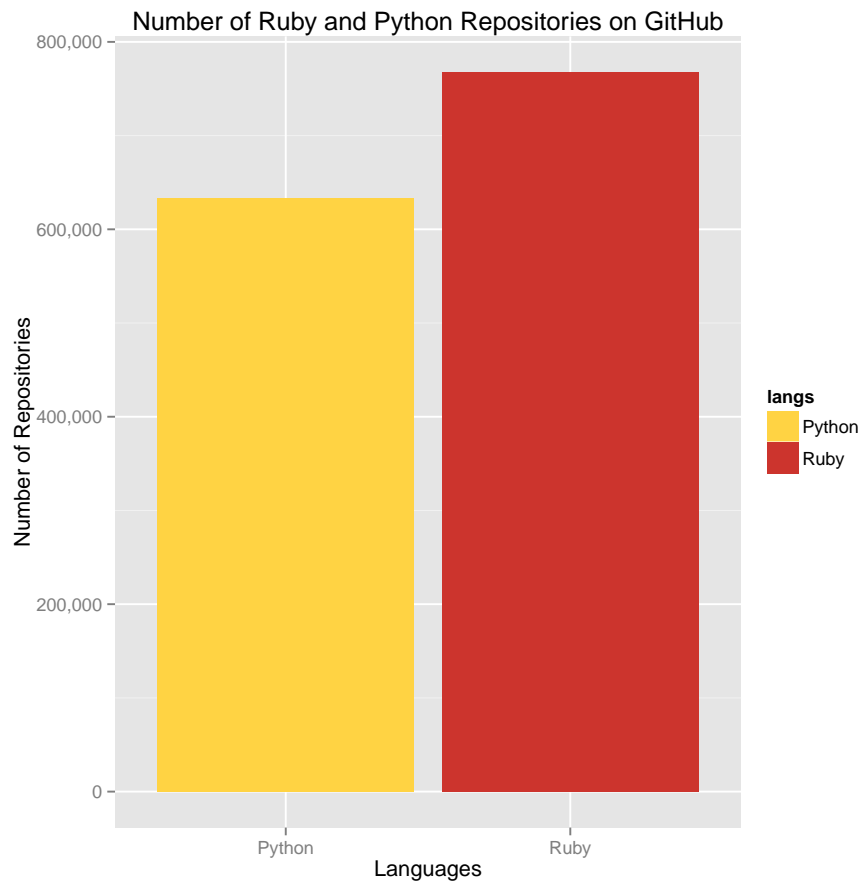
```

library(scales)
langs <- c("Ruby", "Python")
totals <- c(ruby_total, python_total)

totals_data <- data.frame(langs, totals)

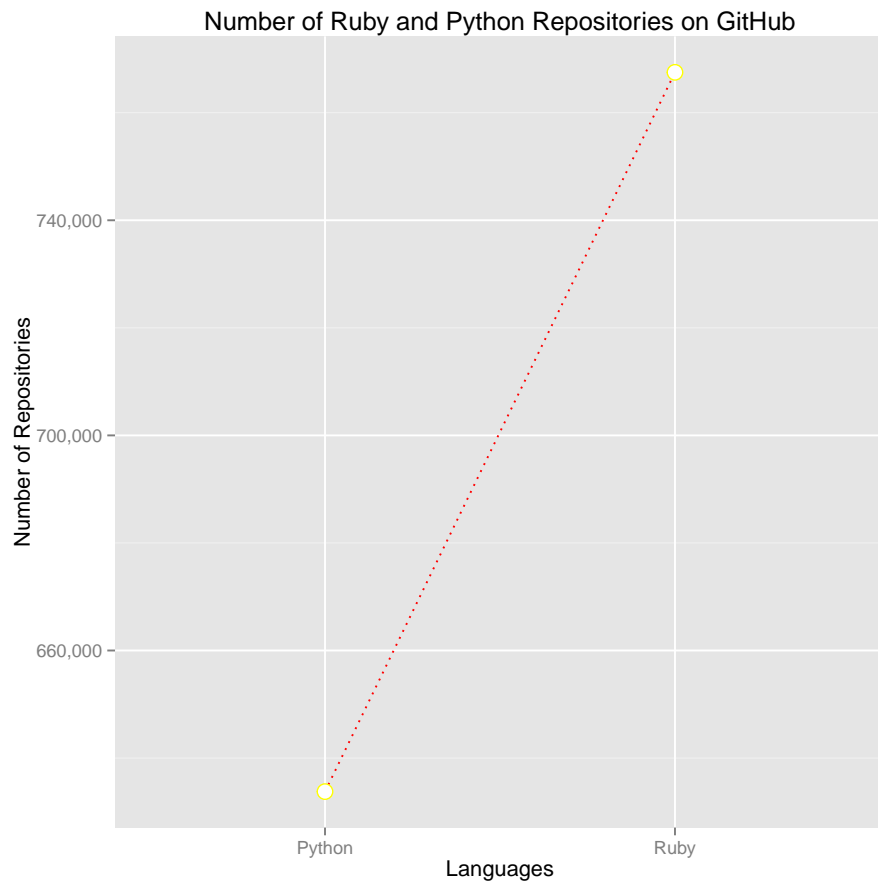
ggplot(data=totals_data, aes(x=langs, y=totals, fill=langs)) +
  geom_bar(stat="identity") + xlab("Languages") + ylab("Number of
    Repositories") + ggtitle("Number of Ruby and Python
    Repositories on GitHub") + scale_fill_manual(values=c("#FFD343",
    , "#CC342D")) + scale_y_continuous(labels = comma)

```



As we can see from the graph, there are more Ruby repositories, but they both have a very respectable number and are not too far off from each other. They are obviously competitors against each other.

```
ggplot(data=totals_data , aes(x=langs , y=totals , group=1)) +  
  geom_line(color="red", linetype="dotted") + geom_point(fill="white", color="yellow", shape=21, size=4) + xlab("Languages") +  
  ylab("Number of Repositories") + ggtitle("Number of Ruby and Python Repositories on GitHub") + scale_y_continuous(labels = comma)
```



3.1 Data Analysis

Here we will take a look at the various information R gives us regarding the data we have gathered.

3.1.1 Ruby

```
class() : integer
```

```
str() :
```

```
summary() : 7.675 × 105, 7.675 × 105, 7.675 × 105, 7.675 × 105, 7.675 × 105,  
7.675 × 105
```

3.1.2 Python

```
class() : integer
```

str() :

summary() : 6.338×10^5 , 6.338×10^5 , 6.338×10^5 , 6.338×10^5 , 6.338×10^5 ,
 6.338×10^5

In this version of R, the range of integer has a maximum which is 2147483647. If the amount of repositories shown above were to be greater than this maximum R would error out and we would need to define the number in a different way.

4 Summary

In doing this OSEMN project, I have learned a lot of things. For one, I finally was able to utilize the GitHub API in a practical way which I have not done before. Secondly, I learned that there are more Ruby repositories than Python which I found surprising. I assumed that there would be more Python repositories since Python is four years older than Ruby.

Along with learning about GitHub, Ruby, and Python, I also liked being able to utilize \LaTeX in a full format. I think I will definitely use \LaTeX in my directed graduate project for graduation.