GitHub User Analysis

Xin Guan | Ziqian Ge

About GitHub

- a web-based hosting service of version control using Git. (Wikipedia)
- The world's **largest** community of developers to discover.
- 31 million users (October 2018)
- 100 million repositories (November 2018)

Goal of Project

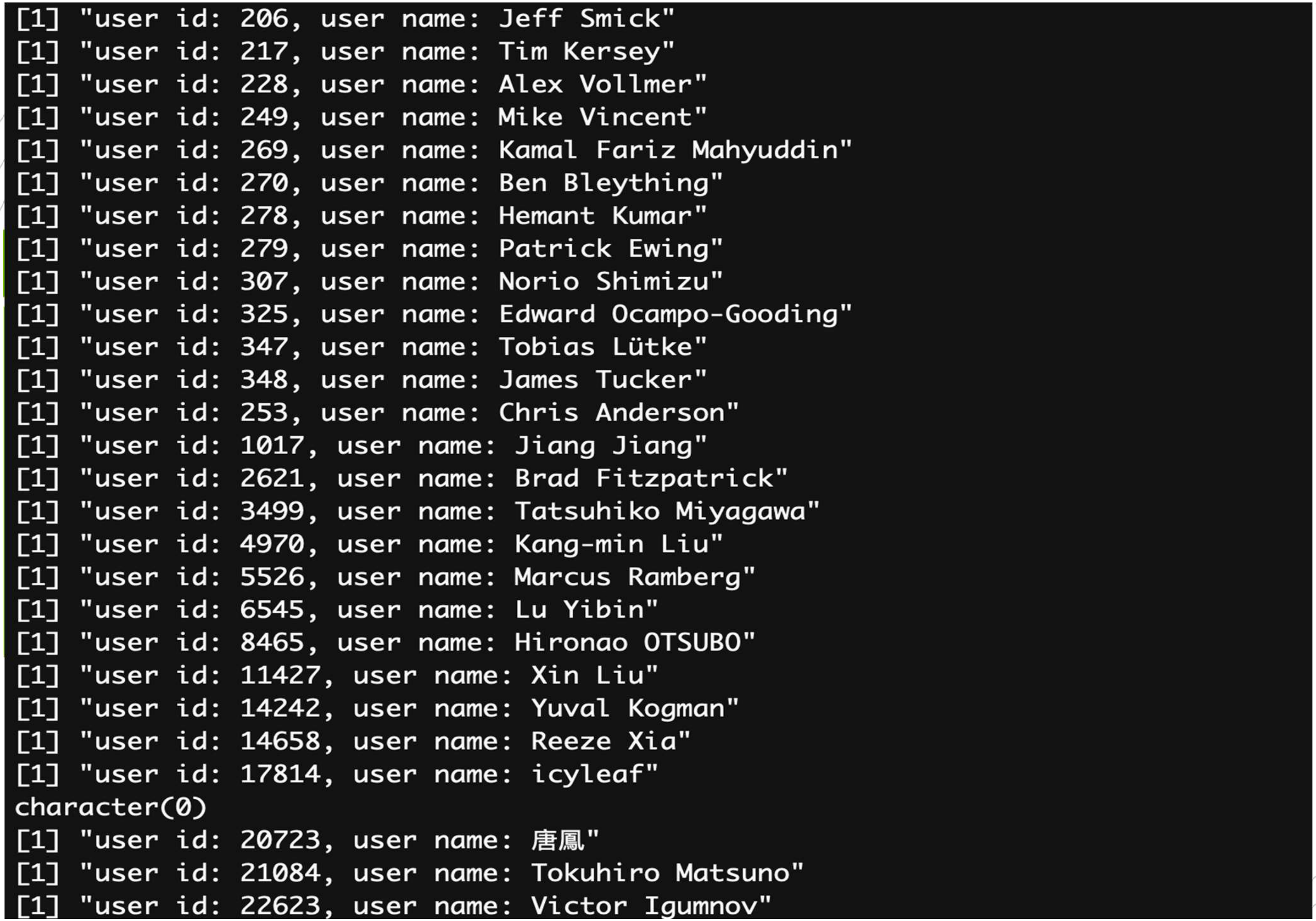
- Who are those developers
- Trends among developers

Content of the Project

- Data Collection
- Data Cleaning
- User Analysis
- Repository Analysis
- Prediction

Data Collection

- GitHub API
- Third party data from GHTorrent



```
[1] "user id: 206, user name: Jeff Smick"
[1] "user id: 217, user name: Tim Kersey"
[1] "user id: 228, user name: Alex Vollmer"
[1] "user id: 249, user name: Mike Vincent"
[1] "user id: 269, user name: Kamal Fariz Mahyuddin"
[1] "user id: 270, user name: Ben Bleything"
[1] "user id: 278, user name: Hemant Kumar"
[1] "user id: 279, user name: Patrick Ewing"
[1] "user id: 307, user name: Norio Shimizu"
[1] "user id: 325, user name: Edward Ocampo-Gooding"
[1] "user id: 347, user name: Tobias Lütke"
[1] "user id: 348, user name: James Tucker"
[1] "user id: 253, user name: Chris Anderson"
[1] "user id: 1017, user name: Jiang Jiang"
[1] "user id: 2621, user name: Brad Fitzpatrick"
[1] "user id: 3499, user name: Tatsuhiko Miyagawa"
[1] "user id: 4970, user name: Kang-min Liu"
[1] "user id: 5526, user name: Marcus Ramberg"
[1] "user id: 6545, user name: Lu Yibin"
[1] "user id: 8465, user name: Hironao OTSUBO"
[1] "user id: 11427, user name: Xin Liu"
[1] "user id: 14242, user name: Yuval Kogman"
[1] "user id: 14658, user name: Reeze Xia"
[1] "user id: 17814, user name: icyleaf"
character(0)
[1] "user id: 20723, user name: 唐鳳"
[1] "user id: 21084, user name: Tokuhiro Matsuno"
[1] "user id: 22623, user name: Victor Igumnov"
```

Difficulty

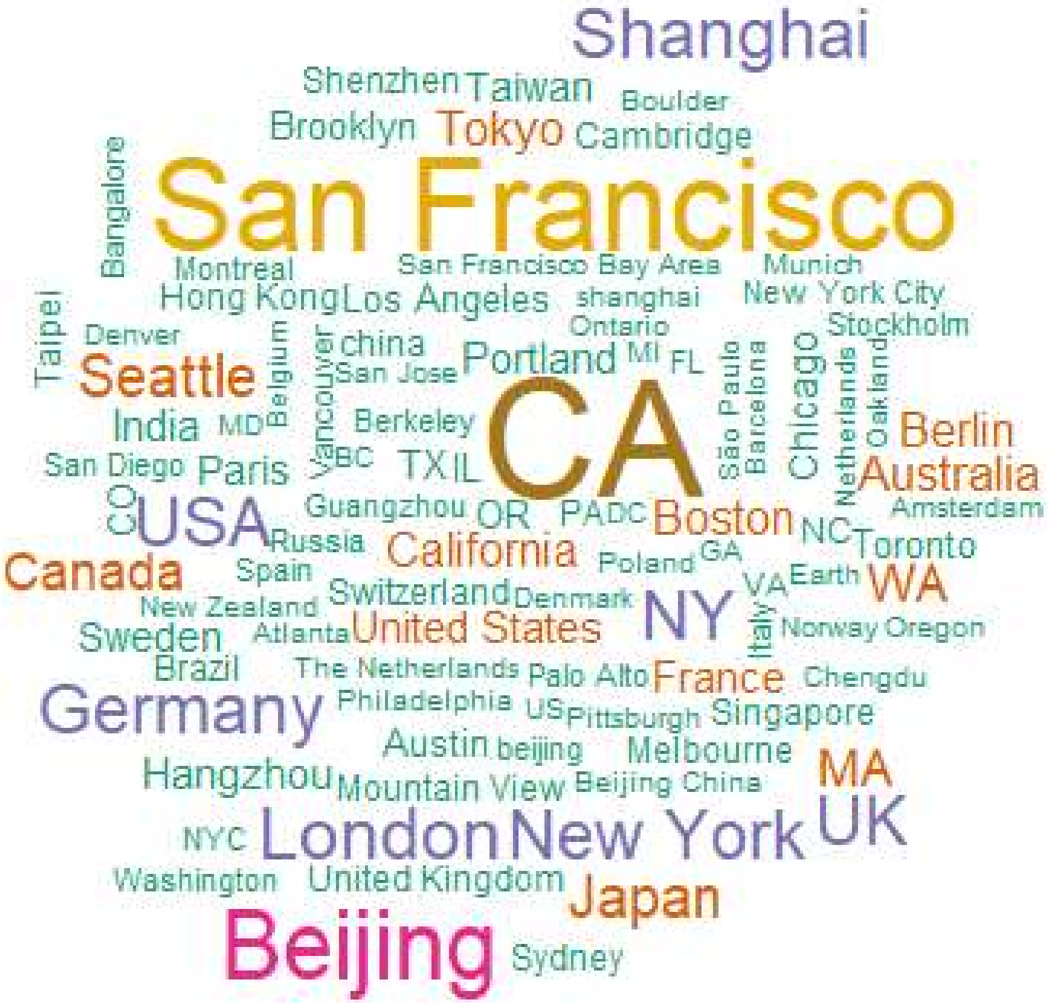
- Large amount of data
- restricted access to API (5000 per hour)
- Not in the form of relational data base
- Uncleaned Data

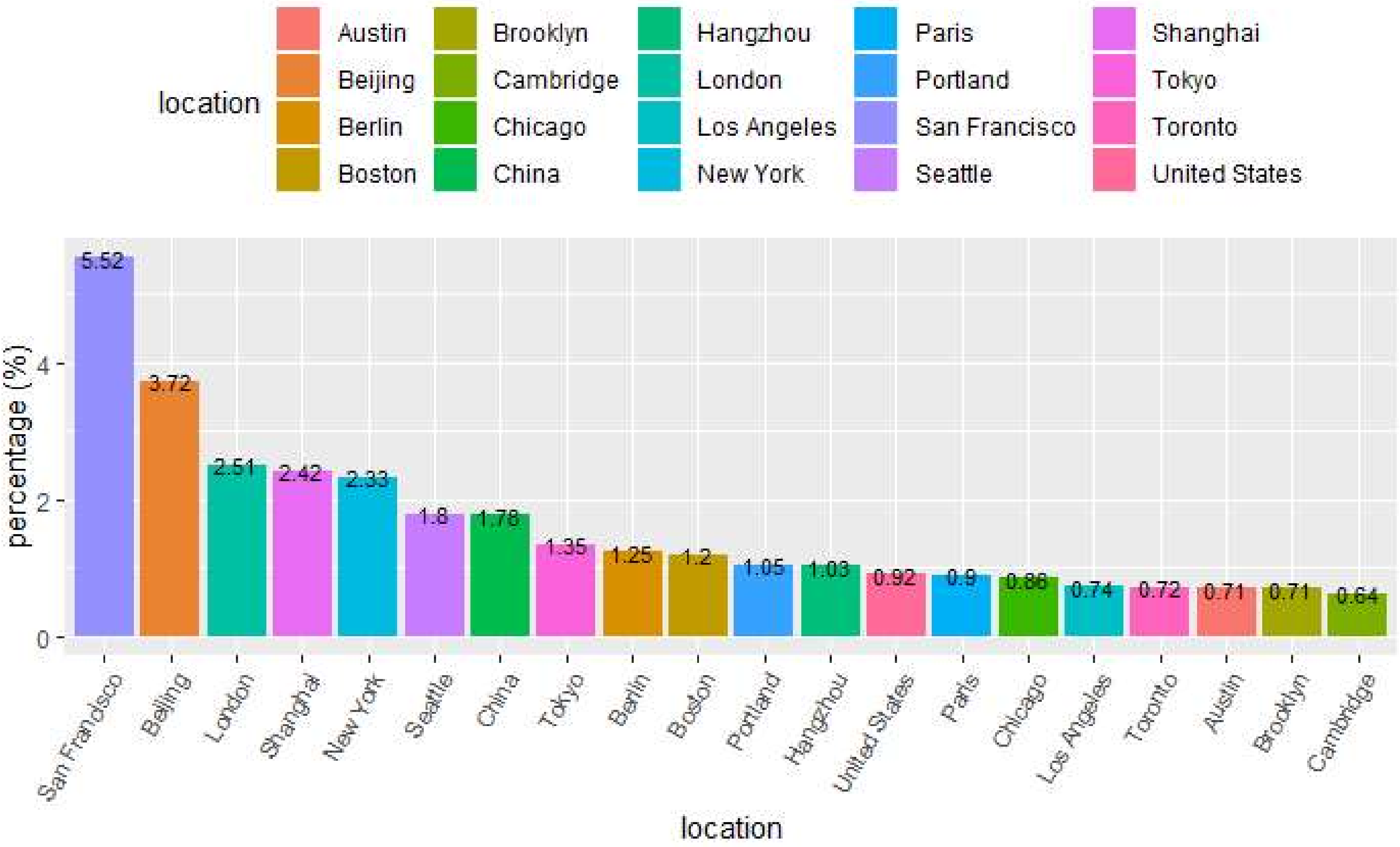
Data Cleaning

- Ignore users with 0 repositories, 0 followers and 0 following
- Ignore users without self-description (no location, company or bio)
- Ignore repos with 0 stars, 0 forks, 0 open issues, or 0 subscibers
- Remove NA
- Outliers

User Analysis

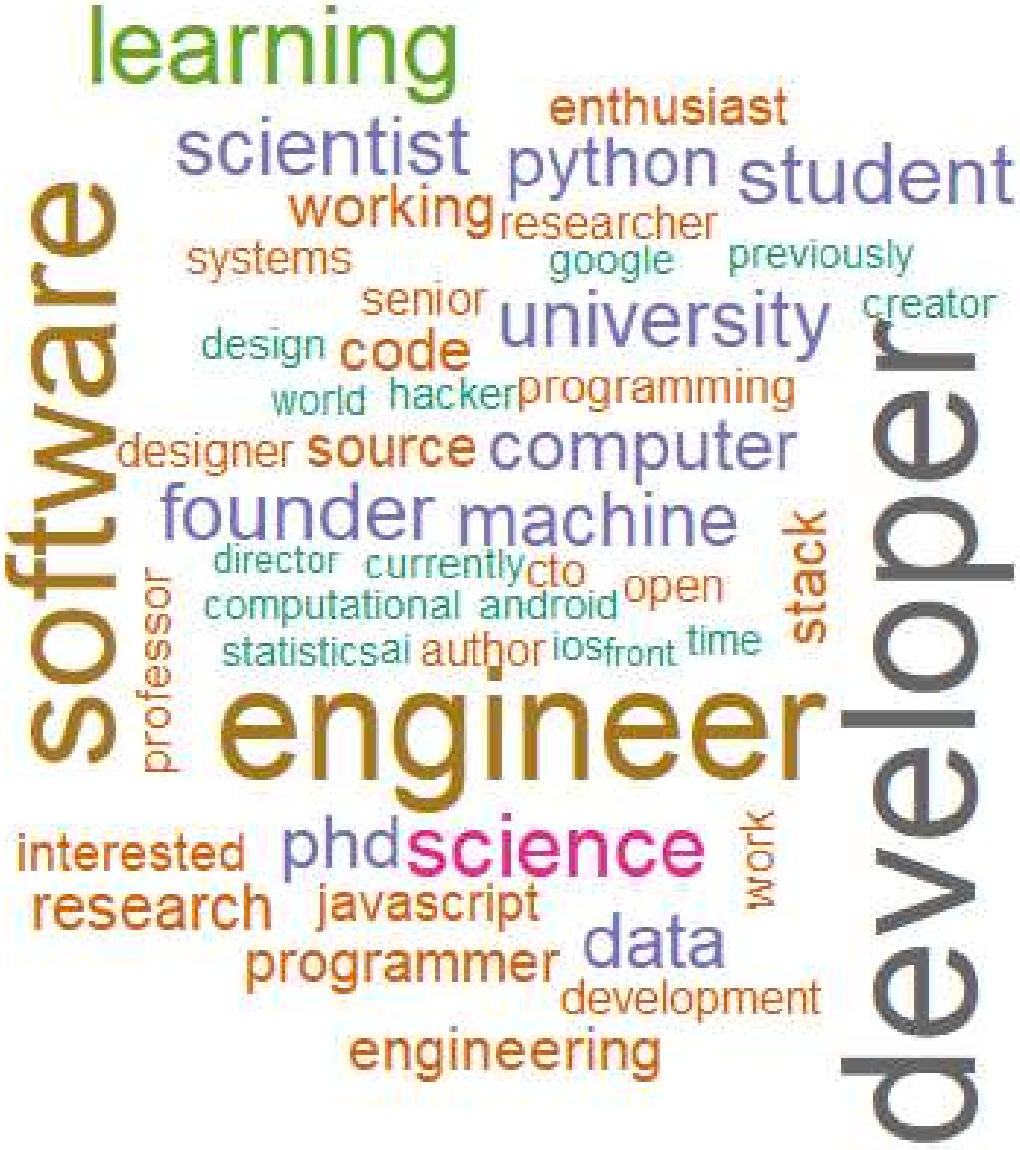
- Where?
  1. Selecting Users with location data
  2. Clean users' information with Regex
  3. Count, Sort, Plot

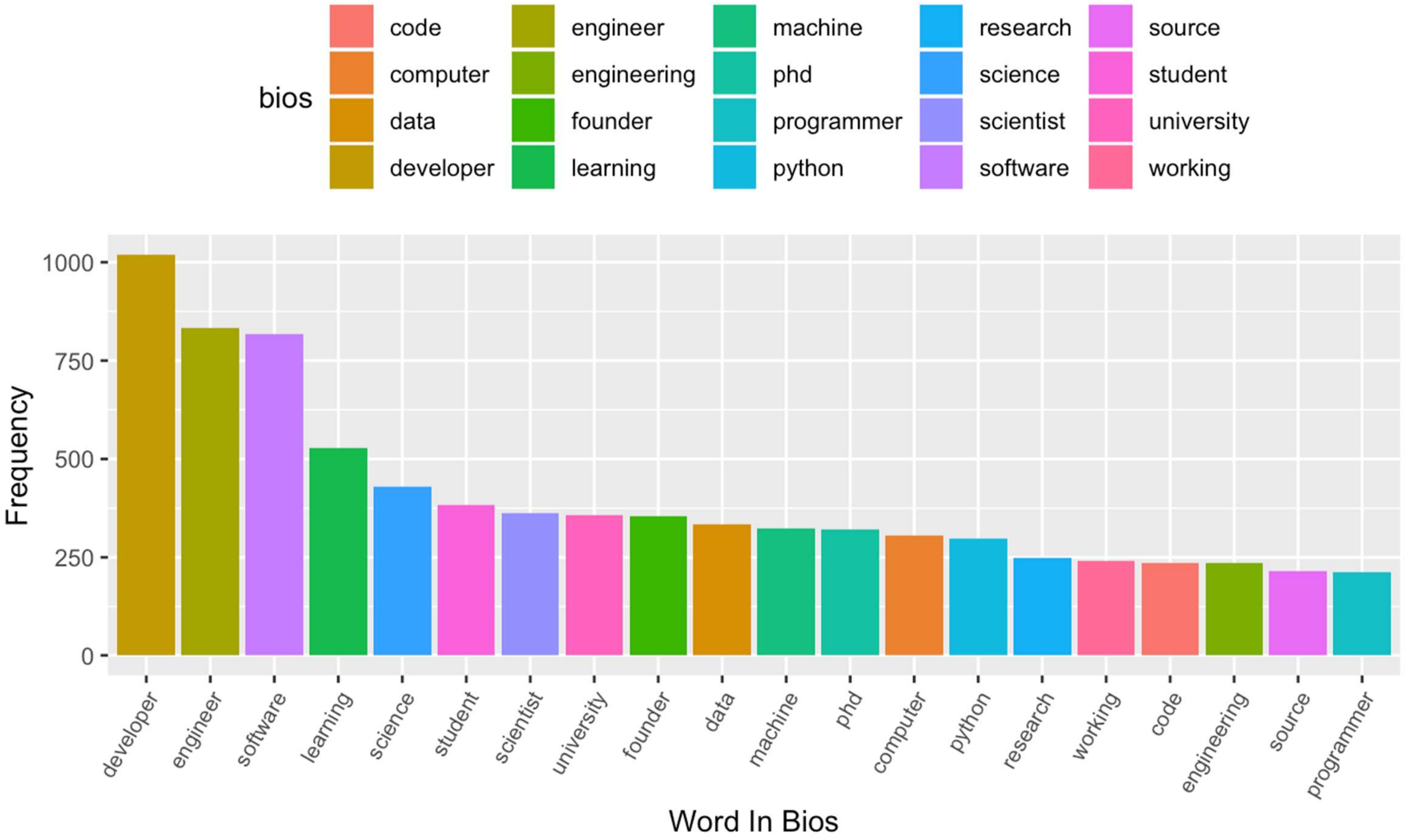

Shanghai
Shenzhen
Taiwan
Boulder
Brooklyn
Tokyo
Cambridge
Bangalore
San Francisco
Montreal
San Francisco Bay Area
Munich
Hong Kong
Los Angeles
shanghai
New York City
Taipei
Denver
Ontario
Stockholm
Belgium
Vancouver
china
San Jose
Portland
MI
FL
Seattle
India
MD
Berkeley
CA
São Paulo
Barcelona
Chicago
Netherlands
Oakland
Berlin
San Diego
Paris
BC
TX
IL
Australia
CO
USA
Guangzhou
OR
PA
DC
Boston
Amsterdam
Russia
California
NC
Toronto
Canada
Spain
Poland
GA
VA
Earth
WA
New Zealand
Switzerland
Denmark
NY
Sweden
Atlanta
United States
Italy
Norway
Oregon
Brazil
The Netherlands
Palo Alto
France
Chengdu
Germany
Philadelphia
US
Pittsburgh
Singapore
Austin
beijing
Melbourne
MA
Hangzhou
Mountain View
Beijing China
NYC
London
New York
UK
Washington
United Kingdom
Japan
Beijing
Sydney


location
Austin
Beijing
Berlin
Boston
Brooklyn
Cambridge
Chicago
China
Hangzhou
London
Los Angeles
New York
Paris
Portland
San Francisco
Seattle
Shanghai
Tokyo
Toronto
United States
percentage (%)
0
2
4
5.52
3.72
2.51
2.42
2.33
1.8
1.78
1.35
1.25
1.2
1.05
1.03
0.92
0.9
0.86
0.74
0.72
0.71
0.71
0.64
San Francisco
Beijing
London
Shanghai
New York
Seattle
China
Tokyo
Berlin
Boston
Portland
Hangzhou
United States
Paris
Chicago
Los Angeles
Toronto
Austin
Brooklyn
Cambridge
location

User Analysis

- Who?
  1. Extract users with introduction
  2. Delete meaningless words
  3. Count the frequency of words
  4. Sort and Plot

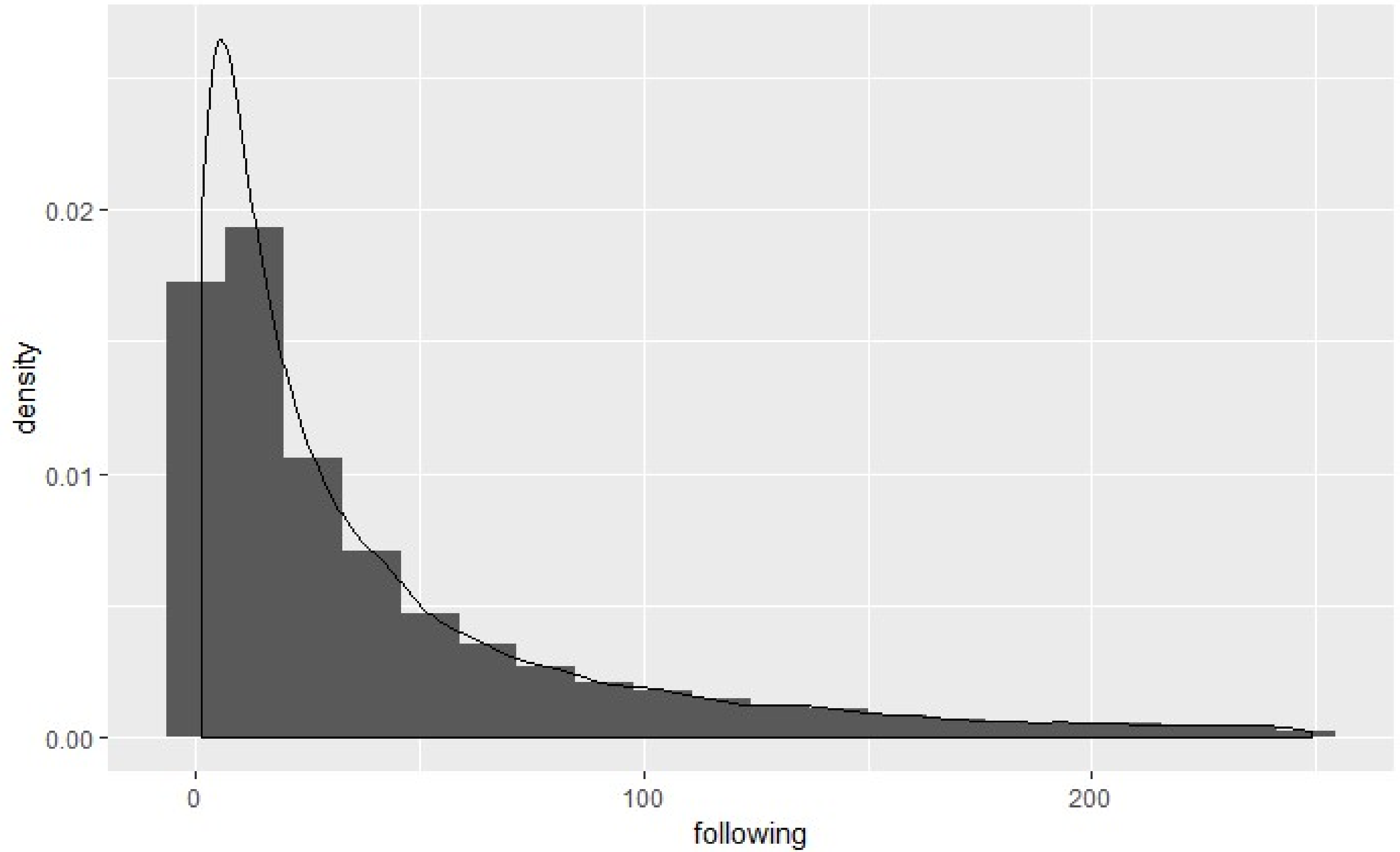

learning
enthusiast
scientist
python
student
software
working
researcher
systems
google
previously
senior
university
creator
design
code
developer
world
hacker
programming
designer
source
computer
founder
machine
stack
director
currently
cto
open
professor
computational
android
statistics
ai
author
ios
front
time
engineer
interested
phd
science
work
research
javascript
data
programmer
development
engineering

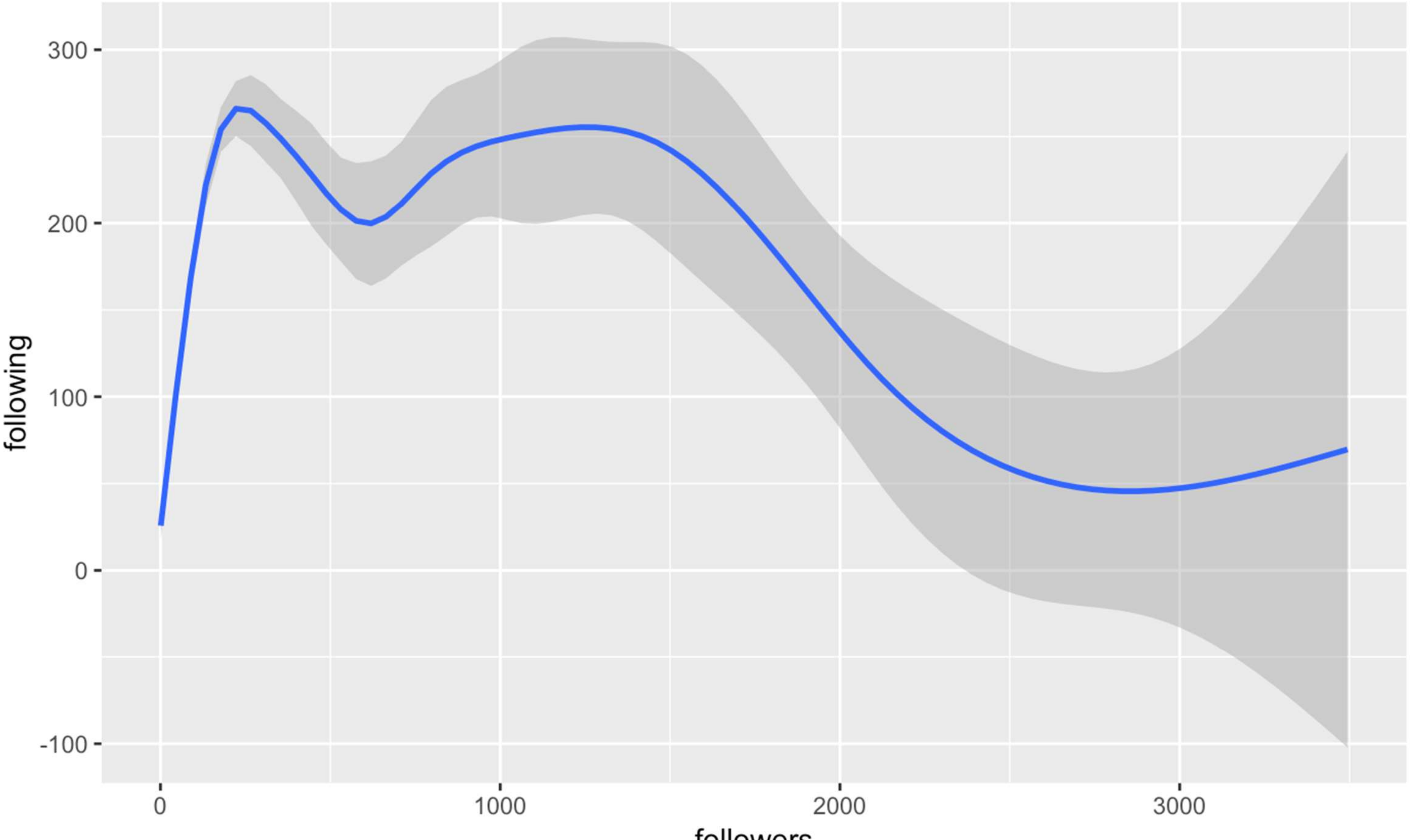


bios
code
computer
data
developer
engineer
engineering
founder
learning
machine
phd
programmer
python
research
science
scientist
software
source
student
university
working
Frequency
1000
750
500
250
0
developer
engineer
software
learning
science
student
scientist
university
founder
data
machine
phd
computer
python
research
working
code
engineering
source
programmer
Word In Bios

User Analysis

- Followers and Following
  - Clean Up data – NA / 0 cases
  - Outliers (93 % < 250)
  - Plot


density
0.02
0.01
0.00
0
100
200
following


300
200
100
0
-100
following
0
1000
2000
3000
followers

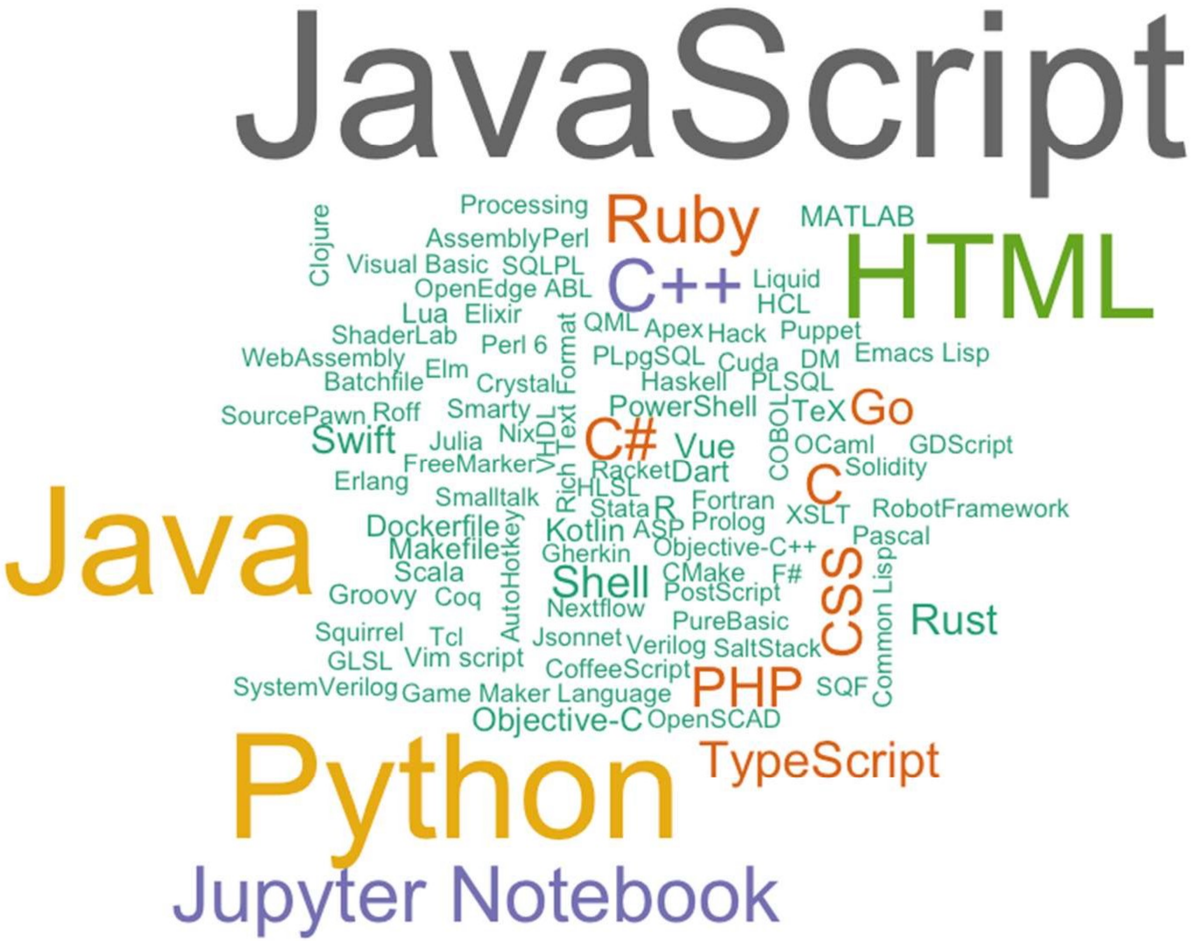
User Analysis

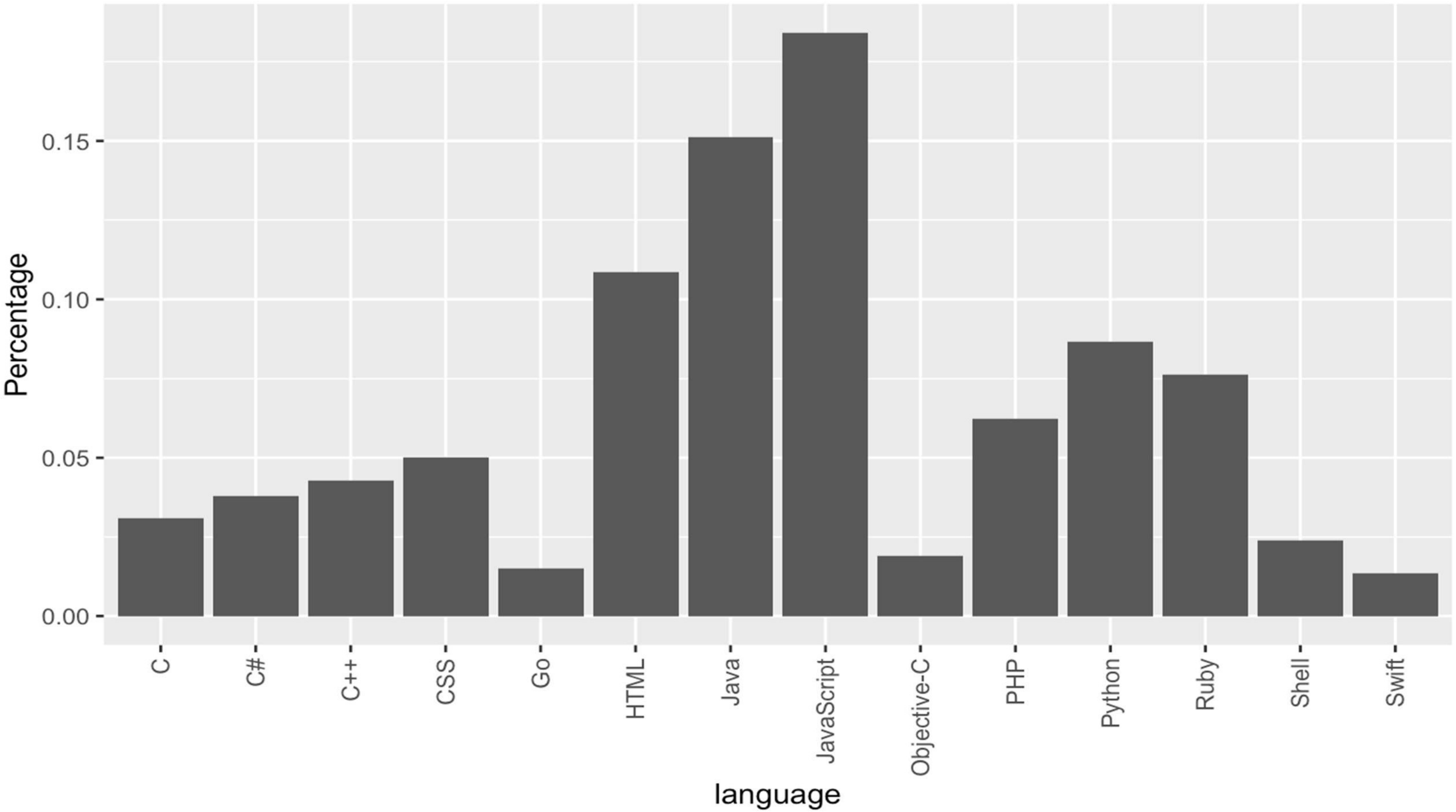
- Not Finished Task:
  - Identify their company / institution / job
  - Relate users with their repositories

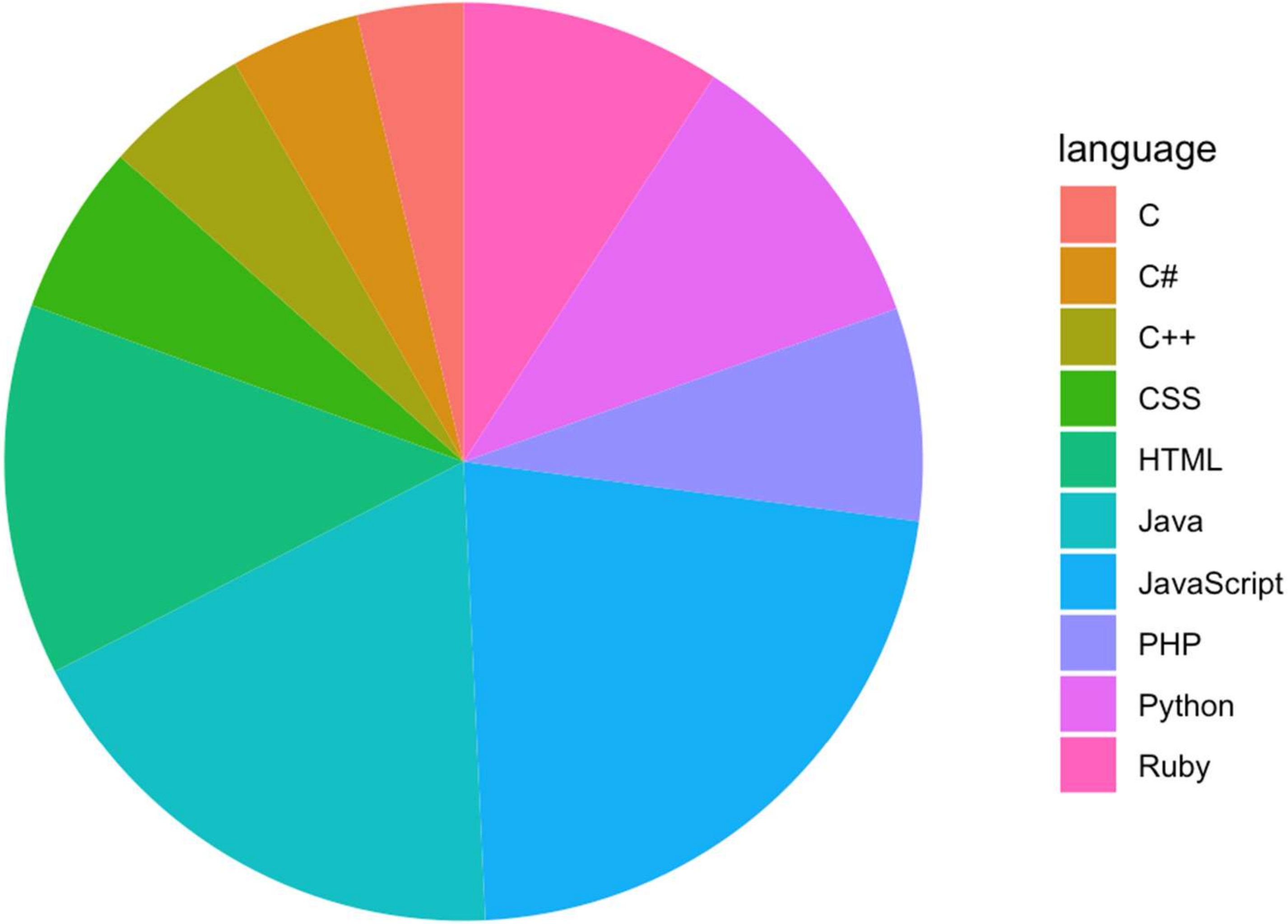
Repo Analysis

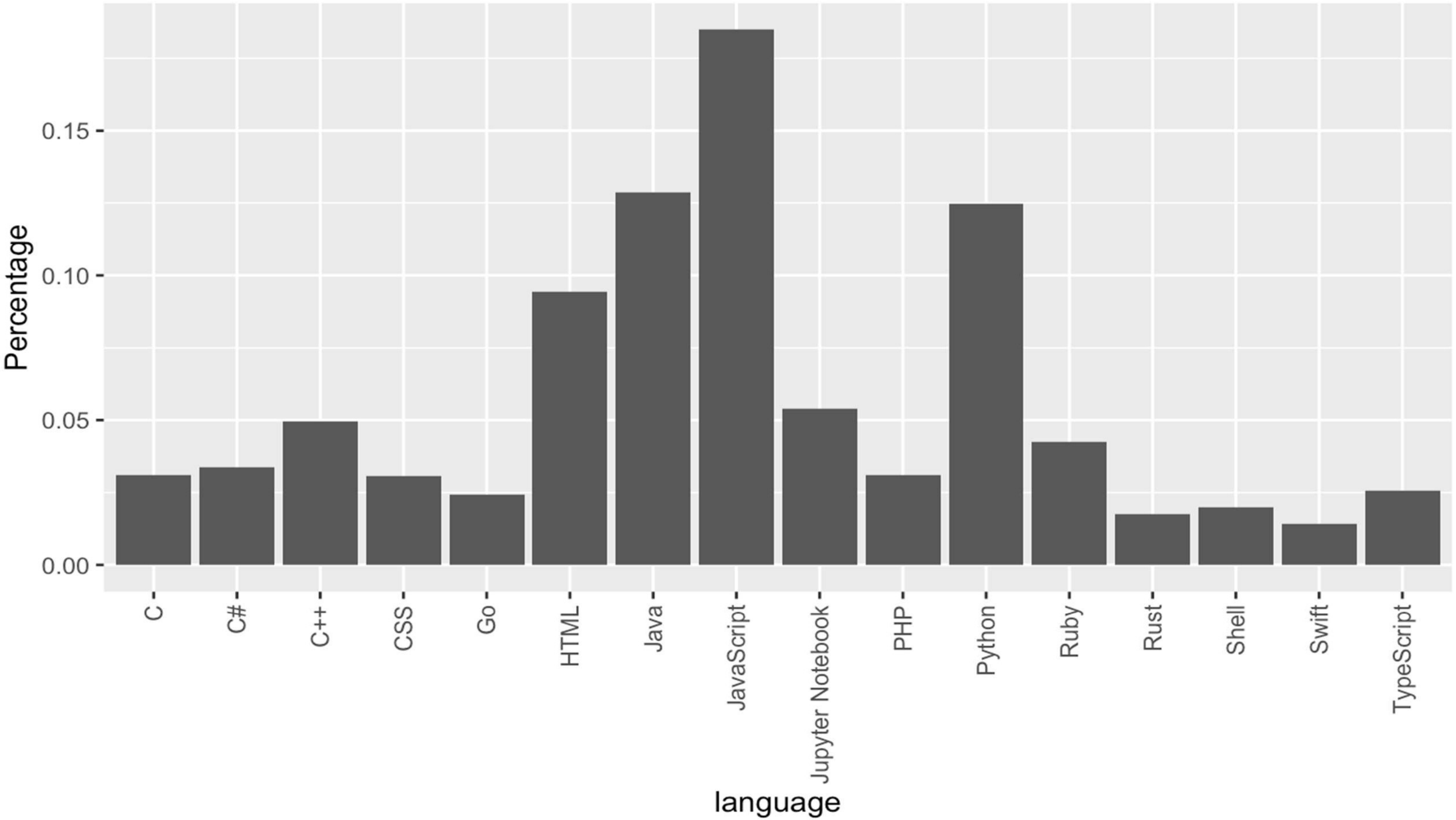
- Languages
  - 1. Count repos that has changed in a day by language
  - 2. Compare popular language with history
- Popularity Analysis
  - Open issues, forks, stars, subscribers

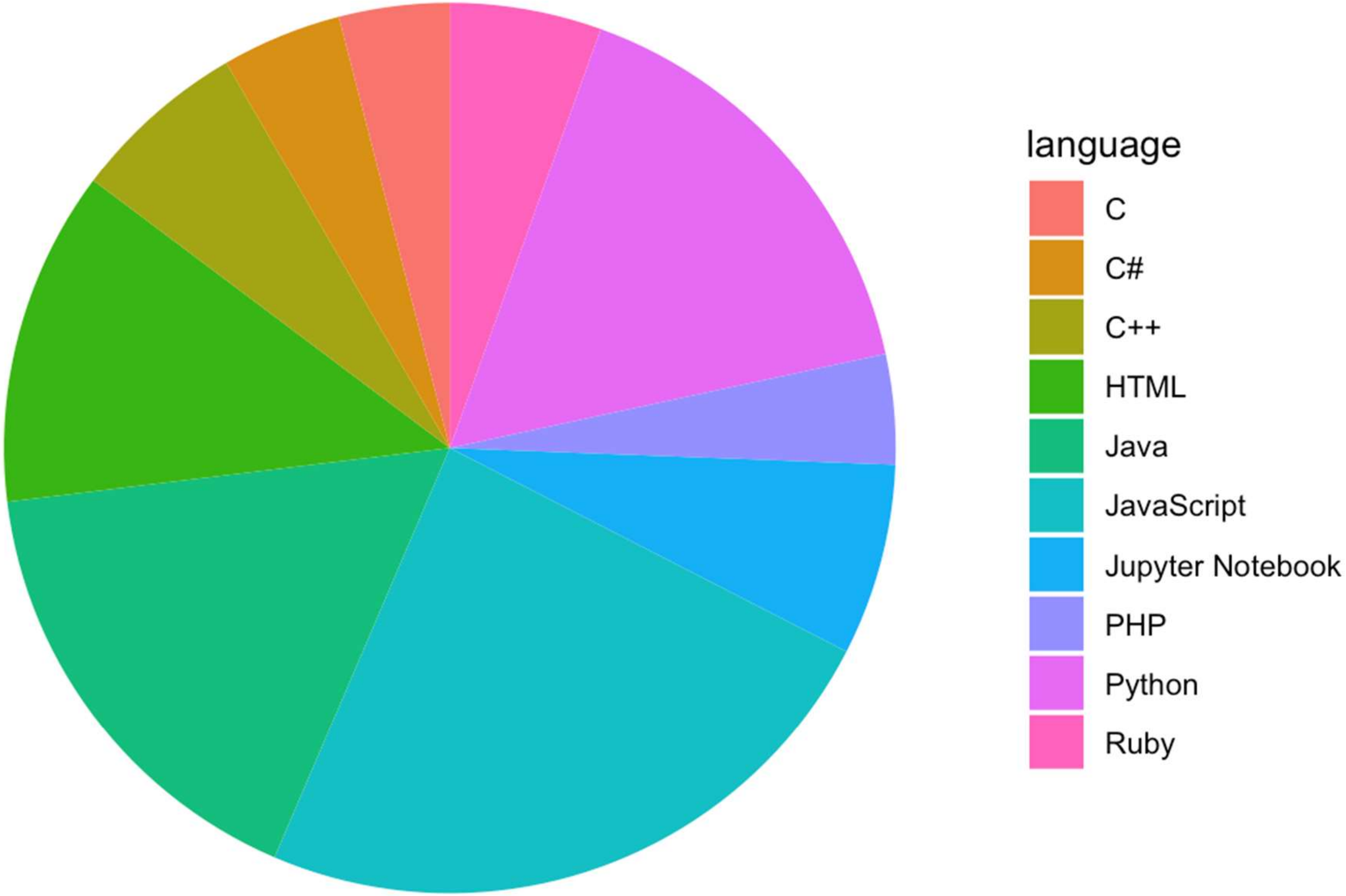
JavaScript
Python
Cuda
SystemVerilog
ApacheConf
Pascal
Makefile
Crystal
GLSL
Assembly
Jupyter Notebook
AutoHotkey
Emacs Lisp
PowerShell
DTrace
CoffeeScript
HTML
LabVIEW
VimL
Kotlin
Matlab
ActionScript
ColdFusion
R
Lua
Groff
SaltStack
Julia
HCL
Nginx
C#
C
C++
Dart
Common Lisp
Groovy
SQLPL
TeX
Coq
Erlang
Scheme
DM
Vue
SAS
Papyrus
F#
Haxe
CMake
Scala
Go
Web Ontology Language
Scilab
PLpgSQL
Slash
KiCad
FORTRAN
Apex
ASP
Rust
Swift
Nix
PostScript
SourcePawn
Racket
Arduino
D
Gosu
VHDL
Smarty
Perl6
Batchfile
Mathematica
XSLT
Haskell
Processing
Elixir
TypeScript
Java
Prolog
QML
Shell
PLSQL
Elm
SQF
Tcl
Clojure
Verilog
Puppet
Eagle
NSIS
OCaml
Objective-C++
Perl
GDScript
Visual Basic
PHP
CSS
Objective-C
Ruby
DIGITAL Command Language

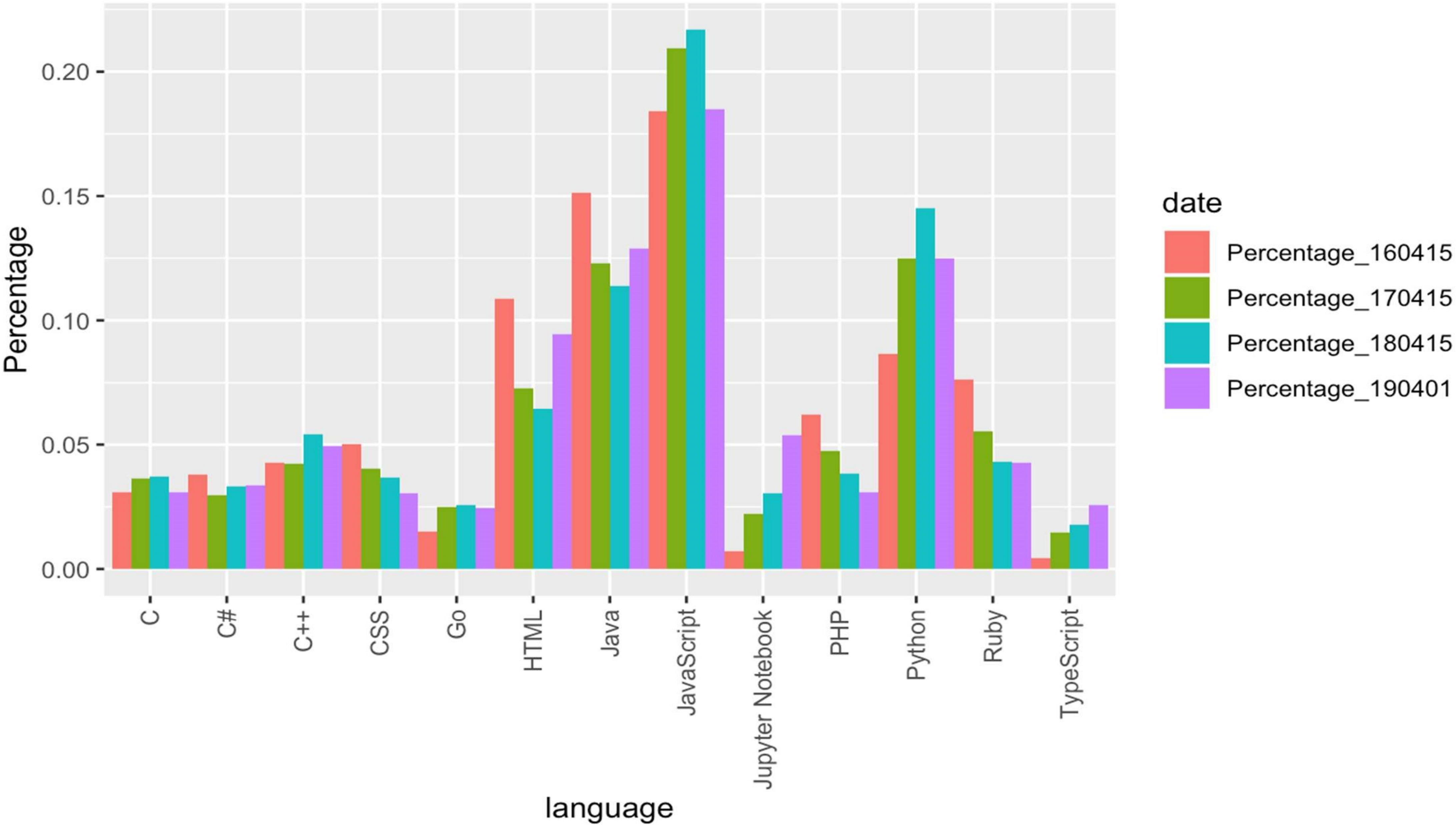

JavaScript
Processing
Ruby
MATLAB
Clojure
AssemblyPerl
HTML
Visual Basic
SQLPL
C++
Liquid
OpenEdge ABL
HCL
Lua
Elixir
Rich Text Format
QML
ShaderLab
Perl 6
Apex
Hack
Puppet
WebAssembly
PLpgSQL
Cuda
DM
Emacs Lisp
Batchfile
Elm
Crystal
Haskell
PLSQL
SourcePawn
Roff
Smarty
PowerShell
COBOL
TeX
Go
Swift
Julia
Nix
VHDL
C#
Vue
OCaml
GDScript
FreeMarker
Racket
Dart
Solidity
Erlang
HLSL
C
Smalltalk
Stata
R
Fortran
XSLT
RobotFramework
Java
Dockerfile
AutoHotkey
Kotlin
ASP
Prolog
Pascal
Makefile
Gherkin
Objective-C++
Scala
CMake
F#
CSS
Common Lisp
Groovy
Coq
Shell
PostScript
Nextflow
Rust
Squirrel
PureBasic
Tcl
Jsonnet
Verilog
SaltStack
GLSL
Vim script
CoffeeScript
SystemVerilog
Game Maker Language
PHP
SQF
Objective-C
OpenSCAD
TypeScript
Python
Jupyter Notebook

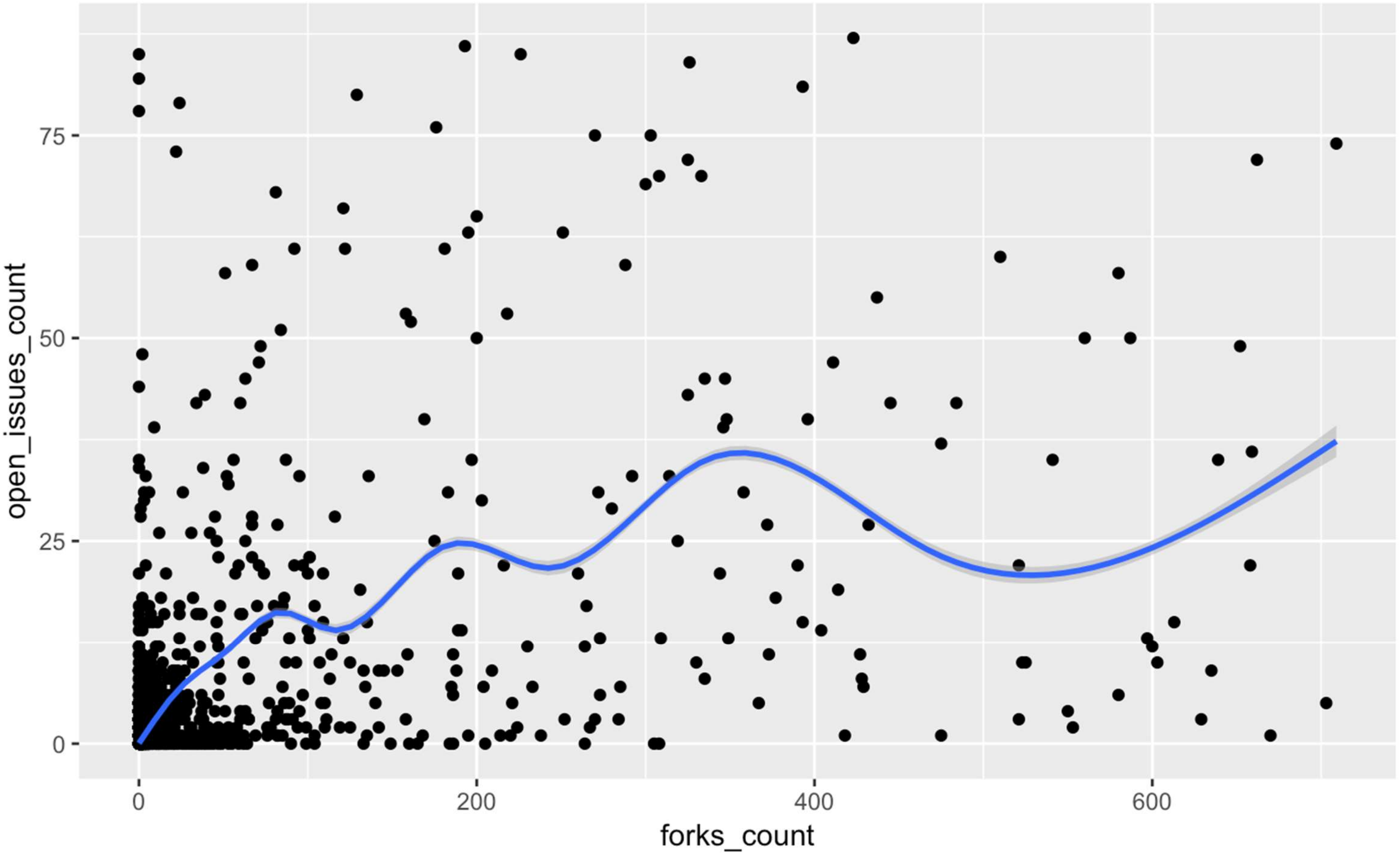

Percentage
0.00
0.05
0.10
0.15
C
C#
C++
CSS
Go
HTML
Java
JavaScript
Objective-C
PHP
Python
Ruby
Shell
Swift
language


language
C
C#
C++
CSS
HTML
Java
JavaScript
PHP
Python
Ruby


Percentage
0.15
0.10
0.05
0.00
C
C#
C++
CSS
Go
HTML
Java
JavaScript
Jupyter Notebook
PHP
Python
Ruby
Rust
Shell
Swift
TypeScript
language


language
C
C#
C++
HTML
Java
JavaScript
Jupyter Notebook
PHP
Python
Ruby


Percentage
0.20
0.15
0.10
0.05
0.00
C
C#
C++
CSS
Go
HTML
Java
JavaScript
Jupyter Notebook
PHP
Python
Ruby
TypeScript
language
date
Percentage_160415
Percentage_170415
Percentage_180415
Percentage_190401


open_issues_count
75
50
25
0
0
200
400
600
forks_count

Repo Analysis

- To be finished
  - Relate with users, i.e. repo owner and contributors
  - Relate with commits

Prediction

- Based on properties of the owner of a repository, and his/her other repositories, predict the popularity of the repo.
  - Evaluate the popularity by a popularity index
  - Build a model on properties of a user
  - Predict using regression