Graph Analysis on GitHub Repository Data

Ivan Pozdnyakov, Neha Gadigi, and Spandana Vallabhaneni

Table of Content

- Introduction
- Data Collection
- Design Architecture
- Analysis
- Conclusion
- Future Work

Introduction

- Construct graphs/networks representing social grouping
- Visualize for comprehension
- Analyze graph structure quantitatively
- Analyze individual nodes qualitatively

Data Collection (i)

- Github repository data
  - MSR14
  - 90 Projects, corresponding commits, issues, and pull requests.
  - Comments on commits, issues, and pull requests
  - Users, repos, watchers, followers

Data Collection (ii)

- Provided with two dump formats BSON and MySQL
    - BSON: binary form for documents readable by MongoDB (18.9GB)
    - MySQL: script that initializes and populates database (422MB)

Data Collection (ii)

- Provided with two dump formats BSON and MySQL
    - **BSON: binary form for documents readable by MongoDB (18.9GB)**
    - MySQL: script that initializes and populates database (422MB)

Design Architecture (i)

- Python
- igraph, pymongo, numpy, and scipy
- Develop a custom system to pipeline workflow
  - Graph construction
  - Data visualization and archival (allows for separation of labor)
  - Analysis

Design Architecture: Graph Construction (ii)

- Projects vs. projects, users vs. users, followers
- Read (repos)
  - add vertices to projects graph, add dictionary records (optimization)
- Read (commits or commit comments)
  - set values for projects, $<P_1: \{U_1,U_2,U_3\}>$ $<P_2: \{U_2,U_3,U_4\}>$
  - add vertices to users graph, add dictionary records (optimization)
- Create project edges dictionary
  - Match projects in $O(N^2)$ steps, $w = |intersect(\{U_1,U_2,U_3\}, \{U_2,U_3,U_4\})|$
  - $<(P_1,P_2): w>$
  - while iterating over projects, set values for users $<U_1: \{P_1,P_2,P_3\}>$ $<U_2: \{P_2,P_3,P_4\}>$
- Create user edges dictionary

Design Architecture: Graph Construction (iii)

- Add edges and corresponding weights
- Followers is a relationship table
  - Easily translated into a directional graph
  - Read through once to add followers and following as vertices
  - Read through a second time to add directional edge from follower to followed
- Check for empty (invalid) records
- Optimization concerns:
  - Large dictionaries require large heap size (64-bit python)
  - Add edges in one operation instead of one-by-one
  - Set limiters on the number of commits considered

Design Architecture: Visualize & Archive

- Output a visual and save graph to a pickle format

Design Architecture: Analysis (i)

- Read pickle file and save graph locally
- Map degree, strength, and pagerank to nodes
- Analytical Analysis
  - Extract specific nodes according condition
- Quantitative Analysis
  - Distribution degree, strength, or pagerank
- Numeric Correlation Analysis (Projects)
  - Degree vs strength
  - Strength vs size
  - Strength vs watchers
  - Strength vs forks

Design Architecture: Analysis (ii)

- Numeric Correlation Analysis (Users)
  - Degree vs strength
  - Strength vs followers
  - Strength vs following
- Categorical Analysis (Projects only)
  - Language categories
  - Create box plots
  - Measure significance
- Global metrics
  - Diameter or farthest points

Design Architecture: Analysis (iii)

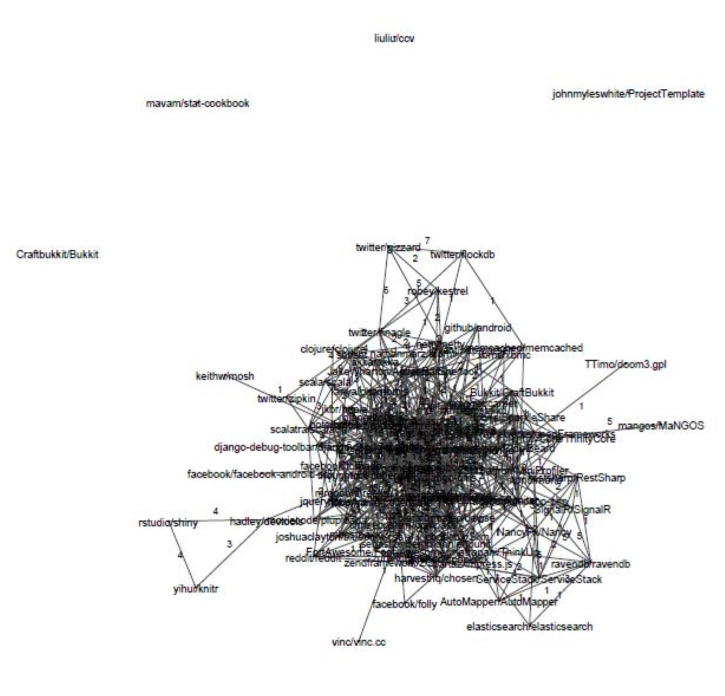
- Followers Analysis
  - Extract users with most followers
  - Cross reference with users graphs
  - Explore who these users are

Analysis: Visualization (i)

- Develop a comprehension for what data looks like
- Assert correctness of graphs

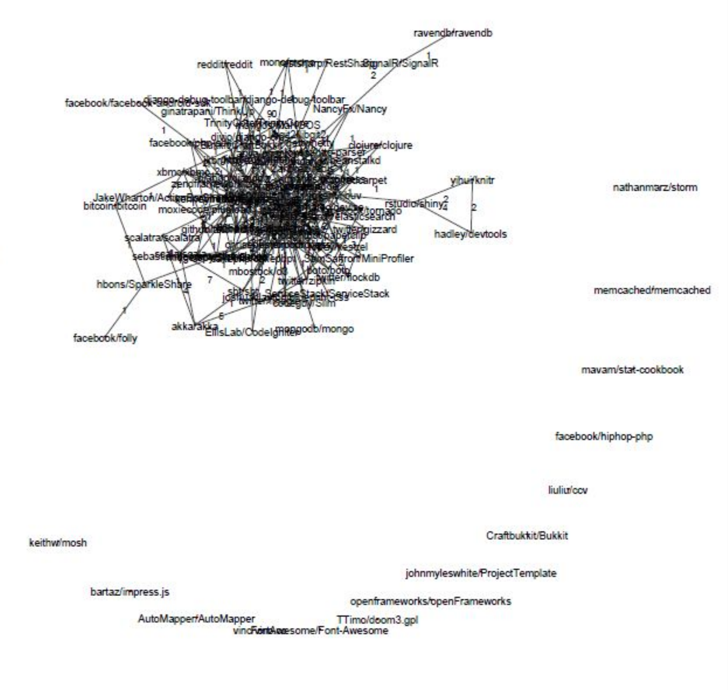
Analysis: Visualization (ii)

- Project vs. Project (Commits)
  - *P_1* and *P_2* share *w* users
  - Users that committed in both projects



Analysis: Visualization (iii)

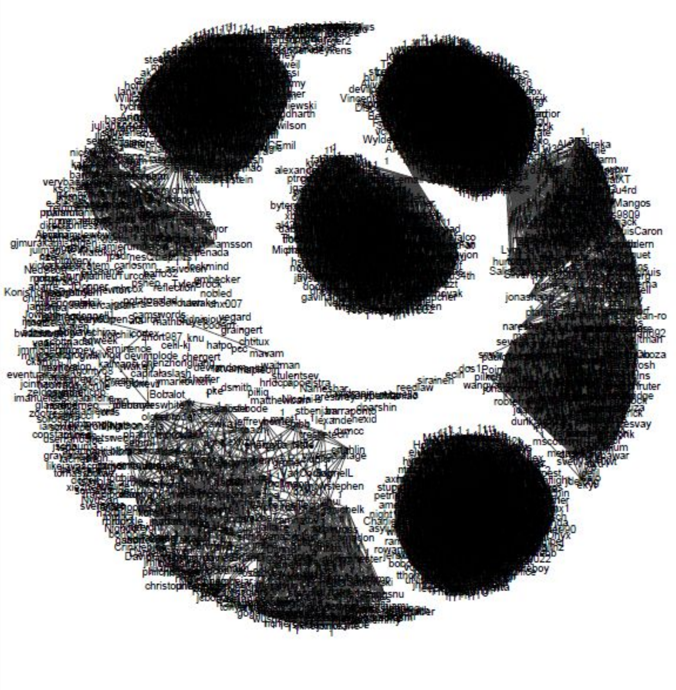
- Project vs. Project (Commit comments)
  - P_1 and P_2 share w users
  - Users that commented in both projects
  - Could be committers as well

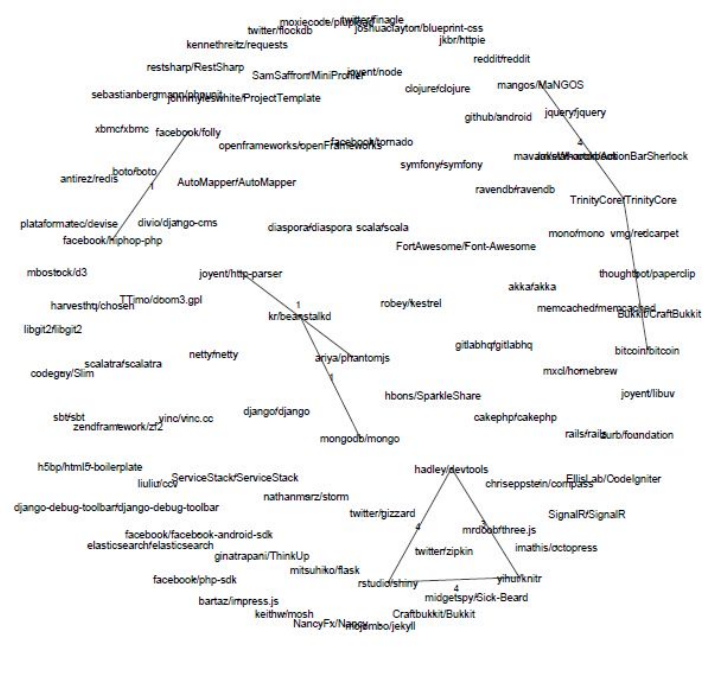


Analysis: Visualization (iv)

- User vs. User
  - U_1 and U_2 share w projects
  - Projects that had commits from both users
  - Projects that had commit comments from both users
- Too large to visually represent
  - Reduced to 1/10th size of original (next slide)
  - Analysis is done on full size graph

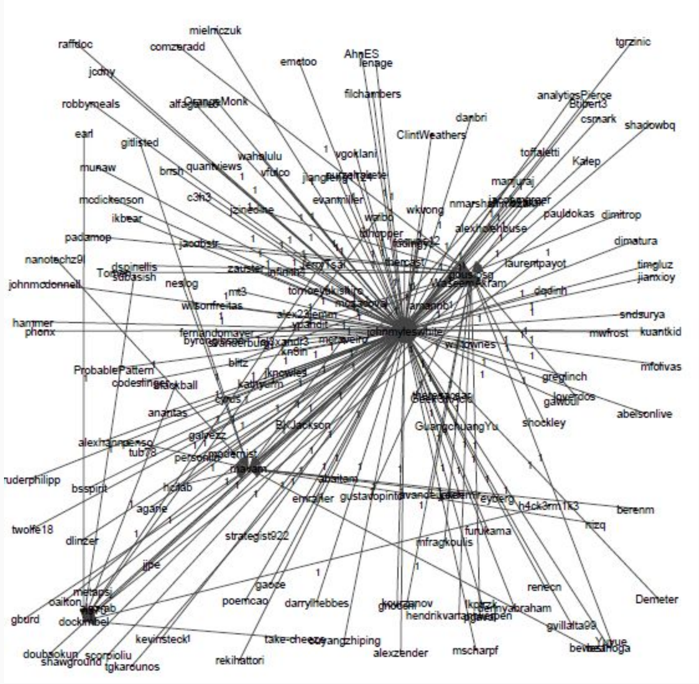
Analysis: Visualization (v)





Analysis: Visualization (vi)

- Followers
  - Reduced to 1/10000th the full size
  - Analysis is done on the full size



Analysis: Global characteristics (i)

- Projects (Commits)
  - 90 Nodes
  - 590 Edges
  - Diameter - 12
- Projects (Commit Comments)
  - 90 Nodes
  - 281 Edges
  - Diameter - 11

Analysis: Global characteristics (ii)

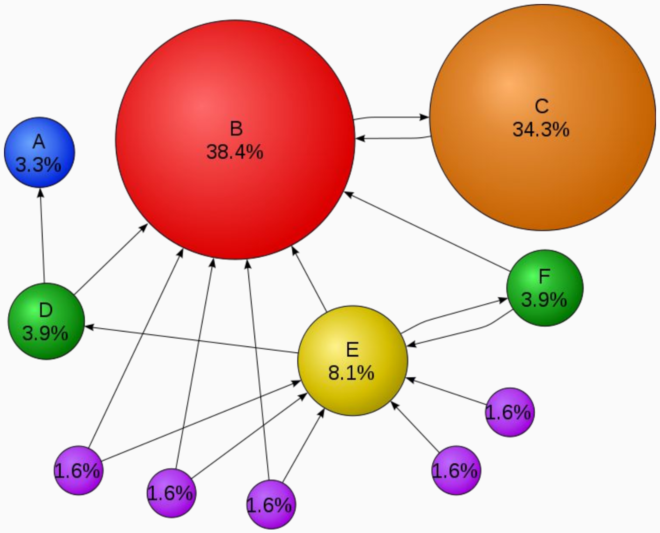
- Users (Commits)
  - 18871 Nodes
  - 2500803 Edges
  - Diameter - 6
- Users (Commit Comments)
  - 8369 Nodes
  - 2934006 Edges
  - Diameter - 8

Analysis: Global characteristics (iii)

- Followers
  - 363599 Nodes
  - 1596888 Edges
  - Diameter - 17

Analysis: PageRank

- Rate nodes based on how many incoming edges and from whom
- Works on undirected graphs as well
- Assumes edges are bidirectional



Projects Analysis: Analytical Analysis (i)

- Commits
- Top degree

Project	Degree	Language	Watchers	Forks	Size	Str	PR
homebrew	54	Ruby	13870	6911	1268	264	.0726
rails	42	Ruby	19587	6548	1272	314	.0771
requests	41	Python	7085	1105	106	128	.0362
node	35	JavaScript	24559	4736	214	94	.0289
gitlabhq	34	Ruby	10244	2521	244	88	.0248

Projects Analysis: Analytical Analysis (ii)

- Commits
- Bottom degree

Project	Degree	Language	Watchers	Forks	Size	Str	PR
ccv	0	C	2494	356	3	0	.0017
Bukkit	0	None	1	0	0	0	.0017
ProjectTemplate	0	R	194	4	4	0	.0017
Stat-cook	0	R	298	2	2	0	.0017
Vinc.cc	1	Ruby	1	0	1	1	.0019

Projects Analysis: Analytical Analysis (iii)

- Commits
- Top strength and pagerank

Project	Str	Language	Watchers	Forks	Size	Deg	PR
rails	314	Ruby	19587	6548	1272	42	.0771
homebrew	264	Ruby	13870	6911	1268	54	.0726
requests	128	Python	7085	1105	106	41	.0362
devise	116	Ruby	9167	1744	222	18	.0282
node	94	Javascript	24559	4736	214	35	.0280

Projects Analysis: Analytical Analysis (iv)

- Commits
- Bottom strength

Project	Str	Language	Watchers	Forks	Size	Degree	PR
ccv	0	C	2494	356	3	0	.0017
Bukkit	0	None	1	0	0	0	.0017
ProjectTemplate	0	R	194	4	4	0	.0017
Stat-cook	0	R	298	2	2	0	.0017
Vinc.cc	1	Ruby	1	0	1	1	.0019

Projects Analysis: Analytical Analysis (v)

- Commit Comments
- Top degree

Project	Degree	Language	Watchers	Forks	Size	Str	PR
rails	49	Ruby	19587	6548	2050	312	.1487
jquery	28	Javascript	23692	4920	245	96	.0499
homebrew	25	Ruby	13870	6911	281	72	.0370
node	25	JavaScript	24559	4736	236	93	.0469
symfony	23	PHP	7103	2542	291	47	.0280

Projects Analysis: Analytical Analysis (vi)

- Commit Comments
- Bottom degree

Project	Degree	Language	Watchers	Forks	Size	Str	PR
ccv	0	C	2494	356	3	0	.0019
Bukkit	0	None	1	0	0	0	.0019
Font-Awesome	0	CSS	16972	2073	1	0	.0019
mem-cache	0	C	2434	627	7	0	.0019
hiphop-php	0	C++	5773	856	0	0	.0019

Projects Analysis: Analytical Analysis (vii)

- Commit Comments
- Top strength

Project	Str	Language	Watchers	Forks	Size	Deg	PR
rails	312	Ruby	19587	6548	2050	49	.1487
TrinityCore	100	C++	2234	1999	847	6	.0308
jquery	96	Javascript	23692	4920	245	28	.0499
mangos	94	C++	1903	1148	13	5	.0279
node	93	Javascript	24559	4736	236	25	.0469

Projects Analysis: Analytical Analysis (viii)

- Commit Comments
- Bottom strength

Project	Str	Language	Watchers	Forks	Size	Degree	PR
ccv	0	C	2494	356	3	0	.0019
Bukkit	0	None	1	0	0	0	.0019
Font-Awesome	0	CSS	16972	2073	1	0	.0019
mem-cache	0	C	2434	627	7	0	.0019
hiphop-php	0	C++	5773	856	0	0	.0019

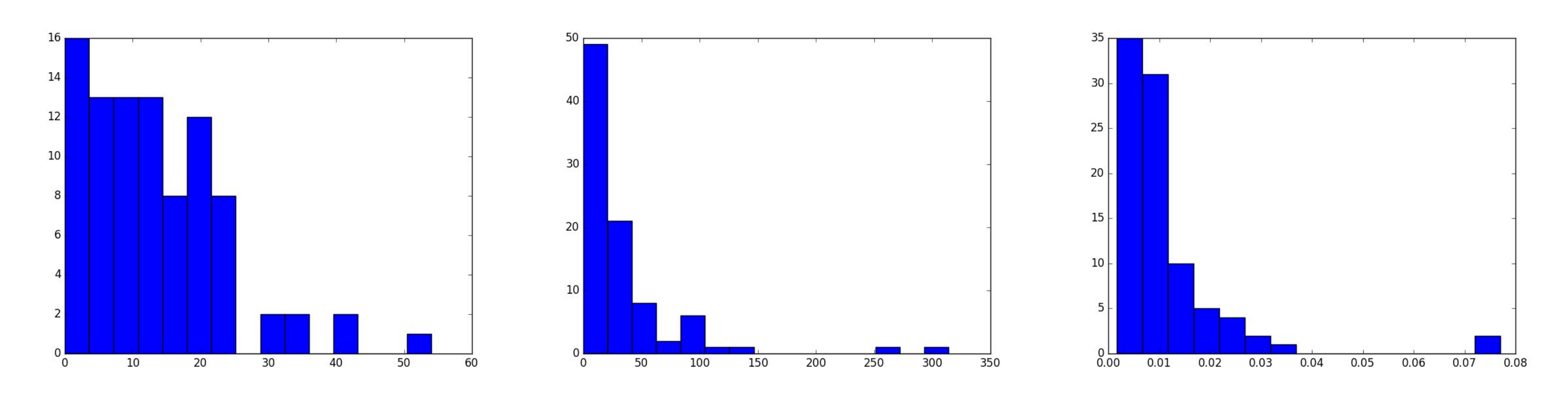
Projects Analysis: Analytical Analysis (ix)

- Commit Comments
- Top pagerank

Project	Str	Language	Watchers	Forks	Size	Deg	PR
rails	312	Ruby	19587	6548	2050	49	.1487
jquery	96	Javascript	23692	4920	245	28	.0499
node	93	Javascript	24559	4736	236	25	.0469
homebrew	72	Ruby	13870	6911	281	25	.0370
html5-boilerplate	63	CSS	22292	5434	266	25	.0326

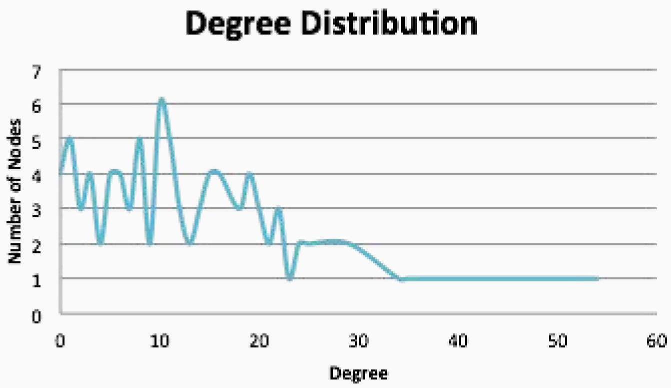
Projects Analysis: Distribution Analysis (i)

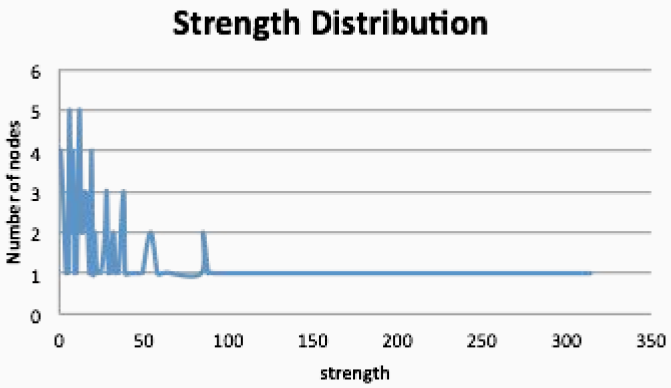
- Degree, strength, and pagerank distribution
  - commits



Projects Analysis: Distribution Analysis (ii)

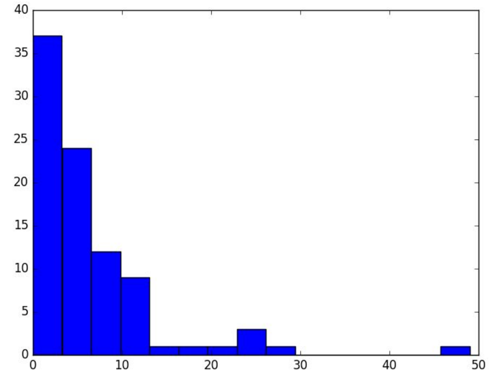
- Degree, strength, and pagerank distribution
  - commits

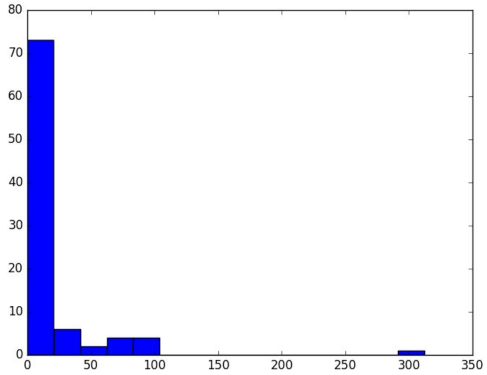


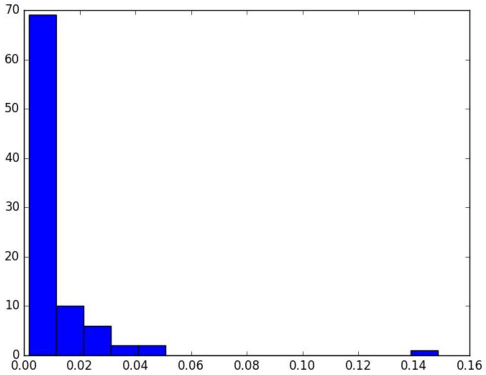


Projects Analysis: Distribution Analysis (iii)

- Degree, strength, and pagerank distribution
  - commit comments

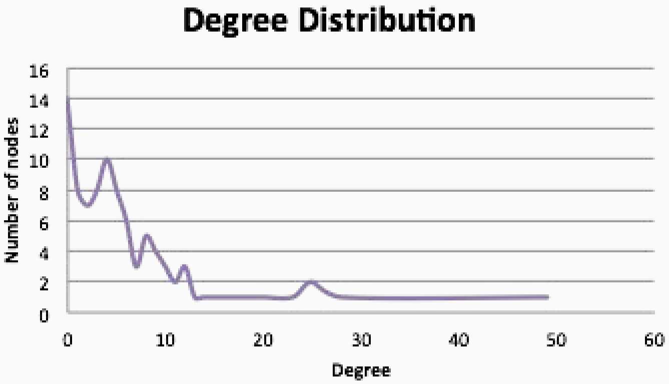


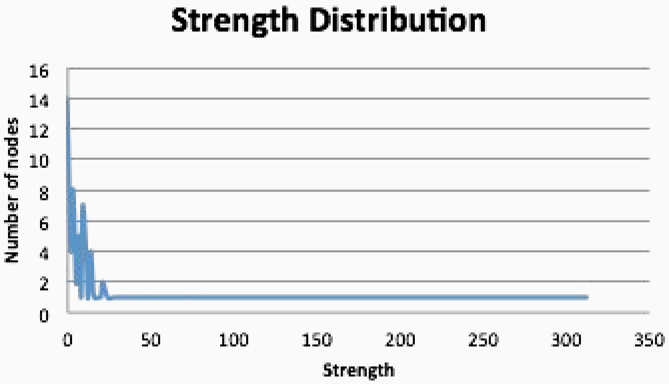




Projects Analysis: Distribution Analysis (iv)

- Degree, strength, and pagerank distribution
  - commit comments





Projects Analysis: Distribution Analysis (v)

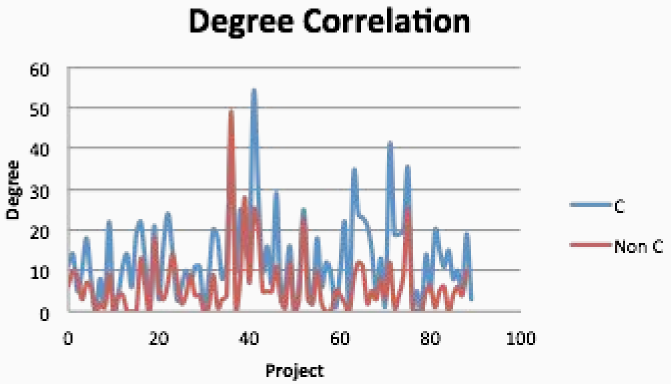
- Distribution Fitting
  - Use allfitdist: black box function
  - Fits a finite list of parametric distributions to the data
  - Sorts based on internal goodness metric outputs highest one
  - For final report

Projects Analysis: Numeric Correlation (i)

- Correlation
  - between degree for commits vs. commit comments
  - between strength for commits vs. commit comments
  - between degree vs. strength
  - between strength vs. other attributes
- Test
  - Pearson product-moment correlation coefficient

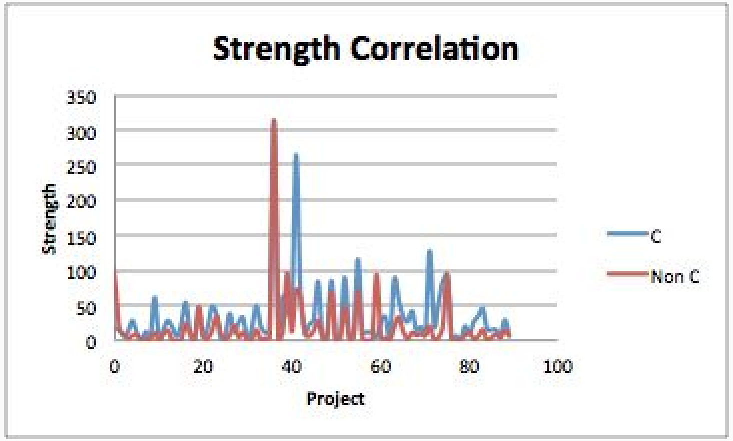
Projects Analysis: Numeric Correlation (ii)

- Degree
- Commits vs. commit comments
- Correlation coefficient: 0.6814



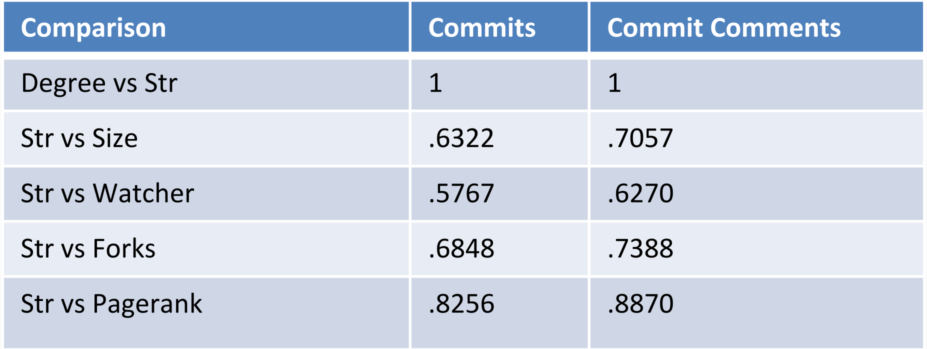
Projects Analysis: Numeric Correlation (iii)

- Strength
- Commits vs. commit comments
- Correlation coefficient: 0.681



Projects Analysis: Numeric Correlation (iv)

- Correlation between attributes



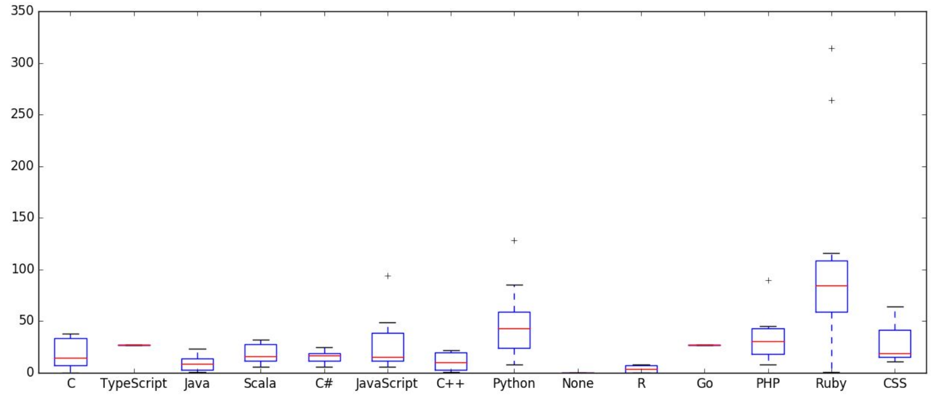
Comparison	Commits	Commit Comments
Degree vs Str	1	1
Str vs Size	.6322	.7057
Str vs Watcher	.5767	.6270
Str vs Forks	.6848	.7388
Str vs Pagerank	.8256	.8870

Projects Analysis: Categorical Correlation (i)

- Are distributions different for categories?
  - Categorize based on languages
  - Use box plots for visualization
- Significance testing
  - Exclude categories with less than 3 projects
  - Test categories’ distribution for normality (Shapiro-Wilk test)
  - Apply Anova test

Projects Analysis: Categorical Correlation (ii)

- Commits



Projects Analysis: Categorical Correlation (iii)

- Commits

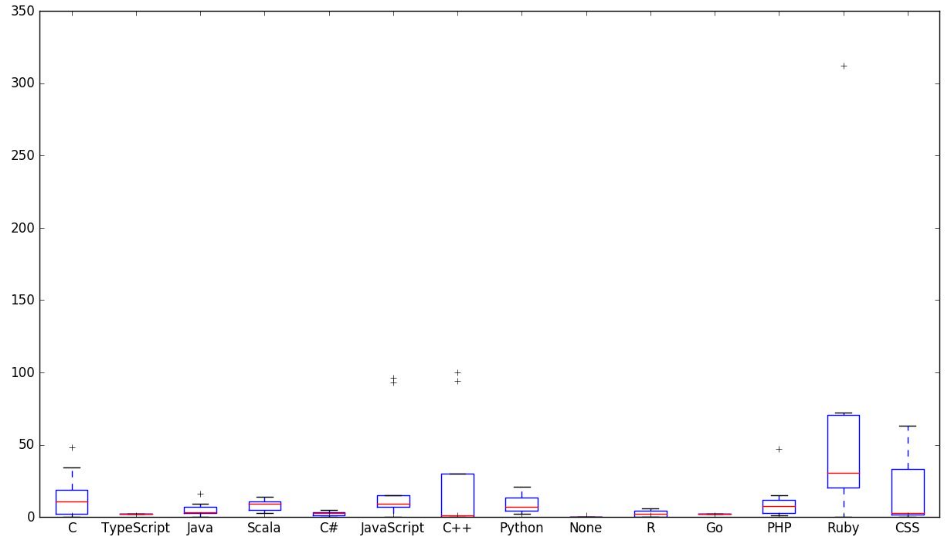
Language	Project #	Shapiro-Wilk Test	Mean
C	10	.12272	19.5
Java	8	.57739	9.625
Scala	9	.16944	19.44
C#	8	.93553	.9355
Javascript	9	.02206	29.33
C++	8	.08053	11.0
Python	10	.30977	49.1
R	4	.07391	3.75
PHP	8	.16859	34.625
Ruby	10	.01543	112.7

Projects Analysis: Categorical Correlation (iv)

- Commits
- Anova test
  - For all: 2.57787351355e-06
  - For Javascript and PHP: 0.696409878605

Projects Analysis: Categorical Correlation (v)

- Commit Comments



Projects Analysis: Categorical Correlation (vi)

- Commit Comments

Language	Project #	Shapiro-Wilk Test	Mean
C	10	.07505	14.7
Java	8	.16082	5.25
Scala	9	.41987	8.66
C#	8	.86192	2.375
Javascript	9	.00041	27.33
C++	8	.00025	25.625
Python	10	.13387	9.6
R	4	.22423	2.5
PHP	8	.00304	11.625
Ruby	10	.00011	64.0

Projects Analysis: Categorical Correlation (vii)

- Commit Comments
- Anova test
    - For all: 0.0280556360508
    - For Javascript and PHP: 0.933107213517

Users Analysis: Global characteristics (i)

- Users (Commits)
  - 18871 Nodes
  - 2500803 Edges
  - Diameter - 6
- Users (Commit Comments)
  - 8369 Nodes
  - 2934006 Edges
  - Diameter - 8

Users Analysis: Analytical Analysis (i)

- Commits
- Users with top degree and strength

Name	Degree	Followers	Following	Str	Commits
FooBarWidget	2879	315	0	2949	14
joneslee85	2742	4	41	2855	26
spagalloco	2648	49	33	2759	4
trevorturk	2648	256	126	2759	33
josevalim	2648	2374	21	2759	4821

Users Analysis: Analytical Analysis (ii)

- Commits
- Users with top Pagerank

Name	PR	Followers	Following	Degree	Str	Commits
invalid-email..	.000653	104	0	1557	1567	9606
dlo	.000410	57	53	2265	2303	20
michaelklishin	.000336	305	20	2399	2473	27
steveklabnik	.000319	1261	134	2219	2267	253
brynary	.000303	369	28	2166	2283	15

Users Analysis: Analytical Analysis (iii)

- Commit comments
- Users with top degree

Name	Degree	Followers	Following	Str	Commit Comments
darkstalker	3206	5	0	3206	61
Informpro	3183	2	0	3185	8
OhaiBBQ	2599	31	22	2663	63
misiav	2569	1034	18	2637	51
jdalton	2556	454	44	2605	168

Users Analysis: Analytical Analysis (iv)

- Commit comments
- Users with top strength

Name	Degree	Followers	Following	Str	commit comments
darkstalker	3206	5	0	3926	61
Informpro	3183	2	0	3185	8
OhaiBBQ	2599	31	22	2663	63
misiav	2569	1034	18	2637	51
visionmedia	2541	5885	123	2616	133

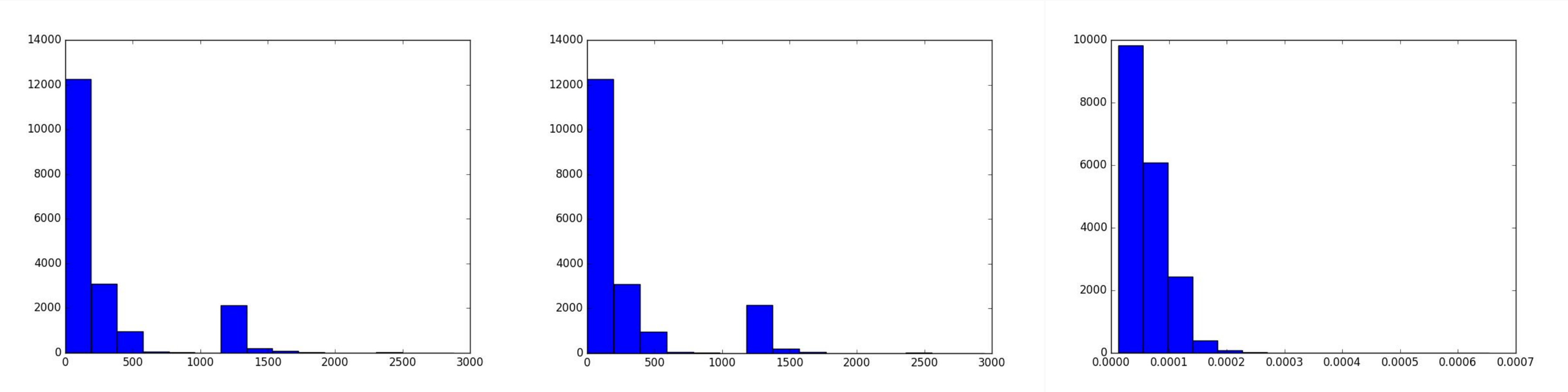
Users Analysis: Analytical Analysis (v)

- Commit comments
- Users with top Pagerank

Name	PR	Degree	Str	Followers	Following	Commit Comm...
kennethreitz	.000505	2493	2533	3833	180	64(2146)
Informpro	.000416	3183	3185	2	0	8(0)
OhaiBBQ	.000414	2599	2663	31	22	63(171)
visionmedia	.000405	2541	2616	5885	123	133(41)
jdalton	.000397	2556	2605	454	44	168(1)

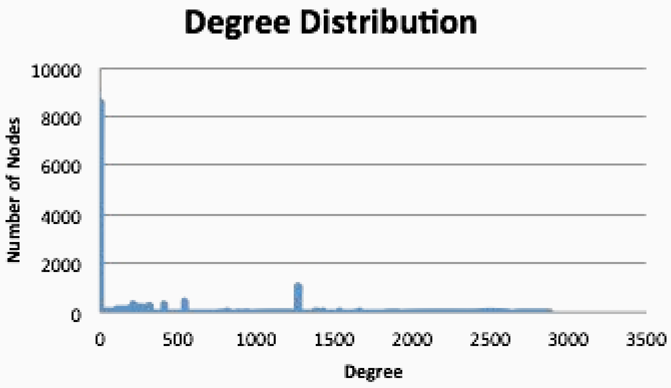
Users Analysis: Distribution Analysis (i)

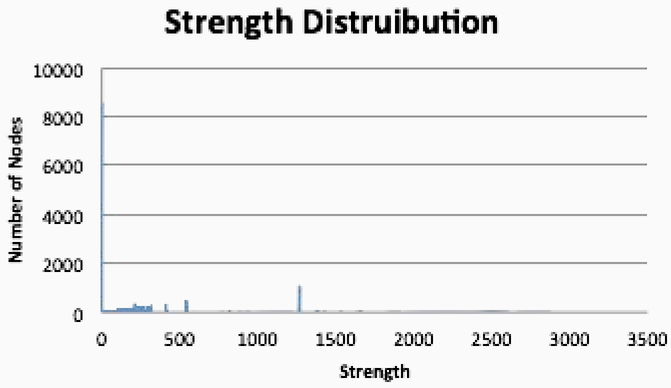
- Degree, strength, and pagerank distribution
  - commits



Users Analysis: Distribution Analysis (ii)

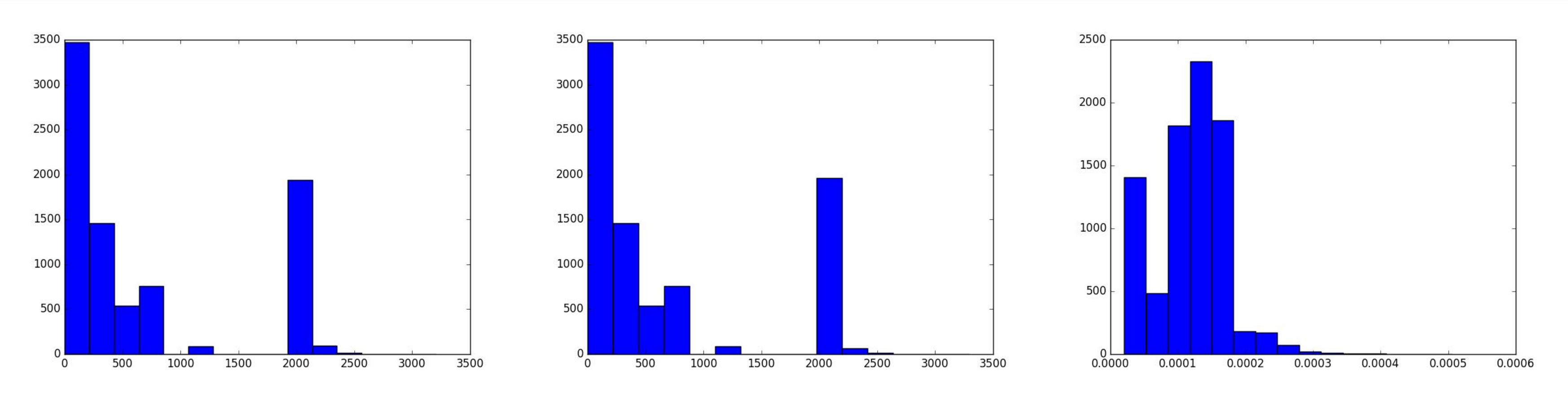
- Degree, strength distribution
  - commits





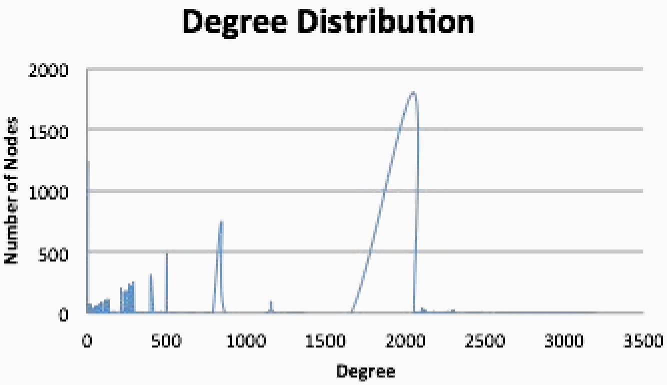
Users Analysis: Distribution Analysis (iii)

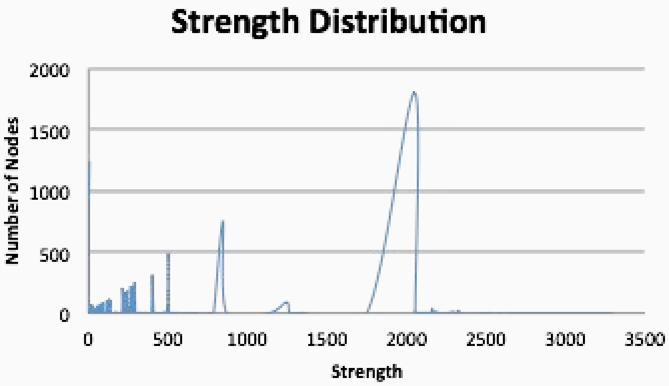
- Degree, strength, and pagerank distribution
  - commit comments



Users Analysis: Distribution Analysis (iv)

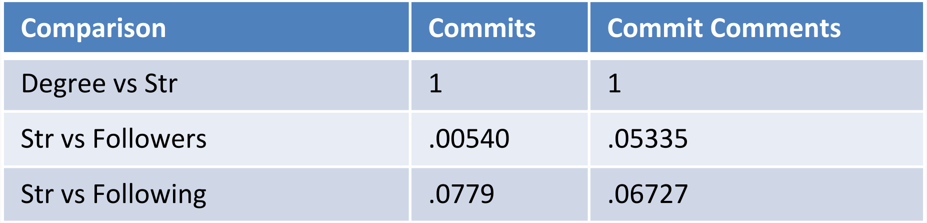
- Degree, strength, and pagerank distribution
  - commit comments





Users Analysis: Numeric Correlation

- Correlation



Comparison	Commits	Commit Comments
Degree vs Str	1	1
Str vs Followers	.00540	.05335
Str vs Following	.0779	.06727

Followers Analysis: Analytical Analysis (i)

Users with top InDegree:Most Followed persons

User	InDegree	OutDegree
defunkt	12340	171
schacon	9149	12
paulirish	8539	135
pjhyett	7547	33
visionmedia	5954	99

Followers Analysis: Analytical Analysis (ii)

Most followed persons: Are they connected with their followers in user graph-commit?

User	InDegree	OutDegree	Connected	Not connected	Followers not in users graph
defunkt	12340	171	0	836	11504
schacon	9149	12	7	392	8750
paulirish	8539	135	123	473	7943
pjhyett	7547	33	16	198	7333
visionmedia	5954	99	38	509	5407

Followers Analysis: Analytical Analysis (iii)

Most followed persons: Are they connected with their followers in user graph-commit comments?

User	InDegree	OutDegree	Connected	Not connected	Followers not in users graph
defunkt	12340	171	318	188	11834
schacon	9149	12	132	106	8911
paulirish	8539	135	183	131	8225
pjhyett	7547	33	-	-	-
visionmedia	5954	99	172	116	5666

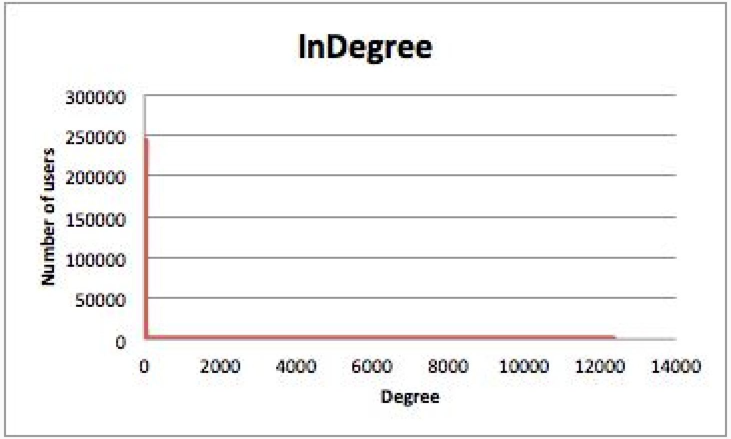
Followers Analysis: Analytical Analysis (iv)

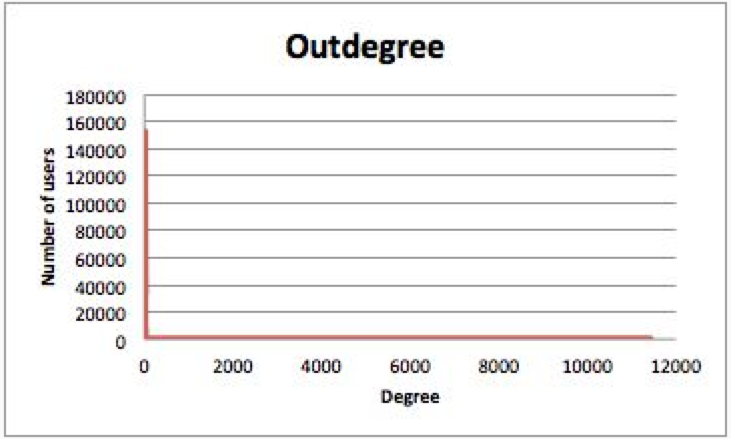
Users with top pagerank

User	PR	InDegree	OutDegree
defunkt	.00473	12340	171
schacon	.00304	9149	12
paulirish	.00303	8539	135
pjhyett	.00290	7547	33
visionmedia	.00246	5954	99

Followers Analysis: Distribution Analysis

- InDegree and outdegree distribution





Followers Analysis: Numeric Correlation

- Correlation between inDegree & outDegree
    - result : 0.08533999

Conclusion

- Developed a fast and expandable pipeline program for constructing and analyzing graphs using Github repository data
- Provided quantitative and qualitative analysis for constructed graphs
- Explored potential trends in software development
- Provided initiative to do more analysis on certain aspects

Future Work

- Refactor code into a library
- Do analysis on issue, issue comments, pull requests, etc…
- Expand the data set (more projects specifically)
- Consider more than just the top language for each project
- Vertex Clustering
    - Try developing a notion of “project types” based on which projects are heavily connected
    - Contract user graph node clusters to single vertices and compare to projects graph