Midterm Presentation

UNIVERSITY OF CALGARY

Suchina Parihar

Sourish Roy

The research questions are the following:

RQ1: What is the frequency distribution of the number of releases across various apps? What is the frequency distribution of the number of days, weeks or months of app releases on Github?

RQ2: How do you do a cluster analysis and model your topics based on the data provided to you?

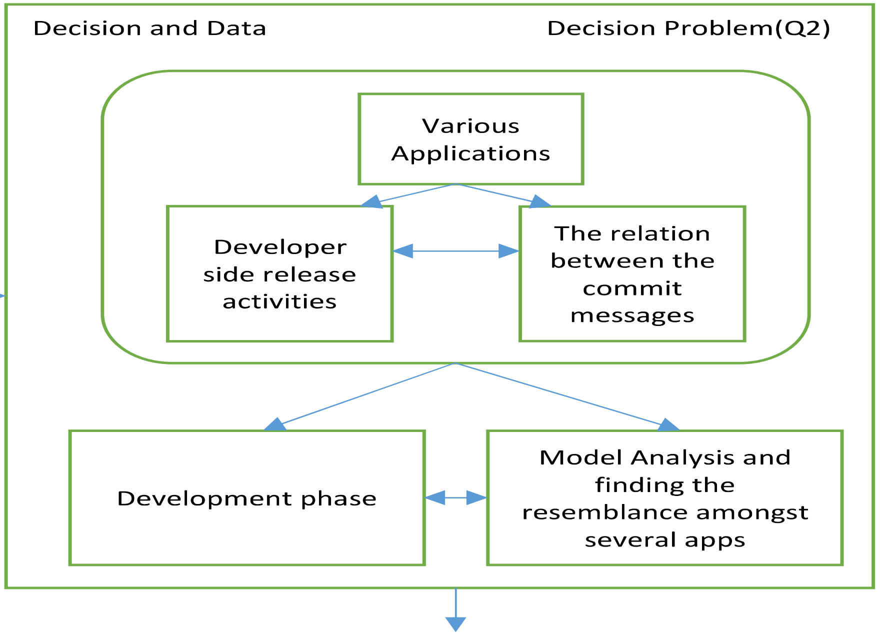
Analysis Design Sheet

Problem & Solution

Modeling

Context (Q1)

- Analyzing the software model being followed by individual apps.
- Defining the various phases of each application’s lifecycle.
- Applying data analysis techniques in each phase of the app lifecycle
- The mining techniques should be based on processing of commit messages.



Decision and Data

Decision Problem(Q2)

Various Applications

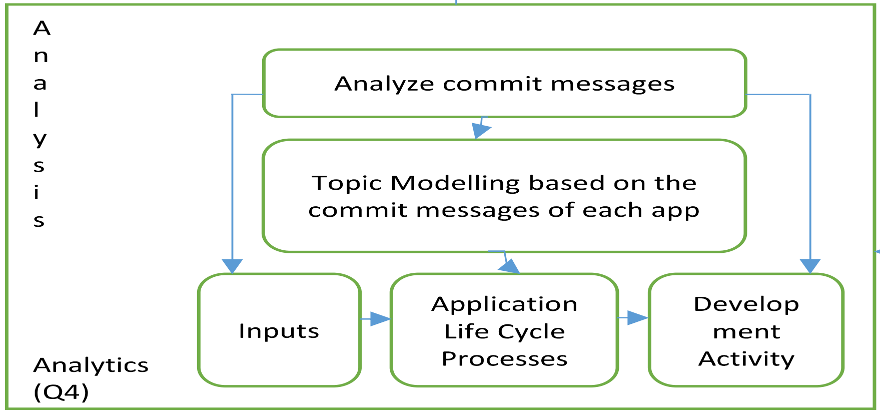
Developer side release activities

The relation between the commit messages

Development phase

Model Analysis and finding the resemblance amongst several apps

Entity	Type	Availability	Importance
Release Time of app	Date	High	High
Commit msg.	Text	High	High
Nature of Data	Text	Medium	Medium Data(Q3)



Analysis

Analyze commit messages

Topic Modelling based on the commit messages of each app

Inputs

Application Life Cycle Processes

Development Activity

Analytics (Q4)

An example of The Data set comprising of commit id, commit date and commit message:

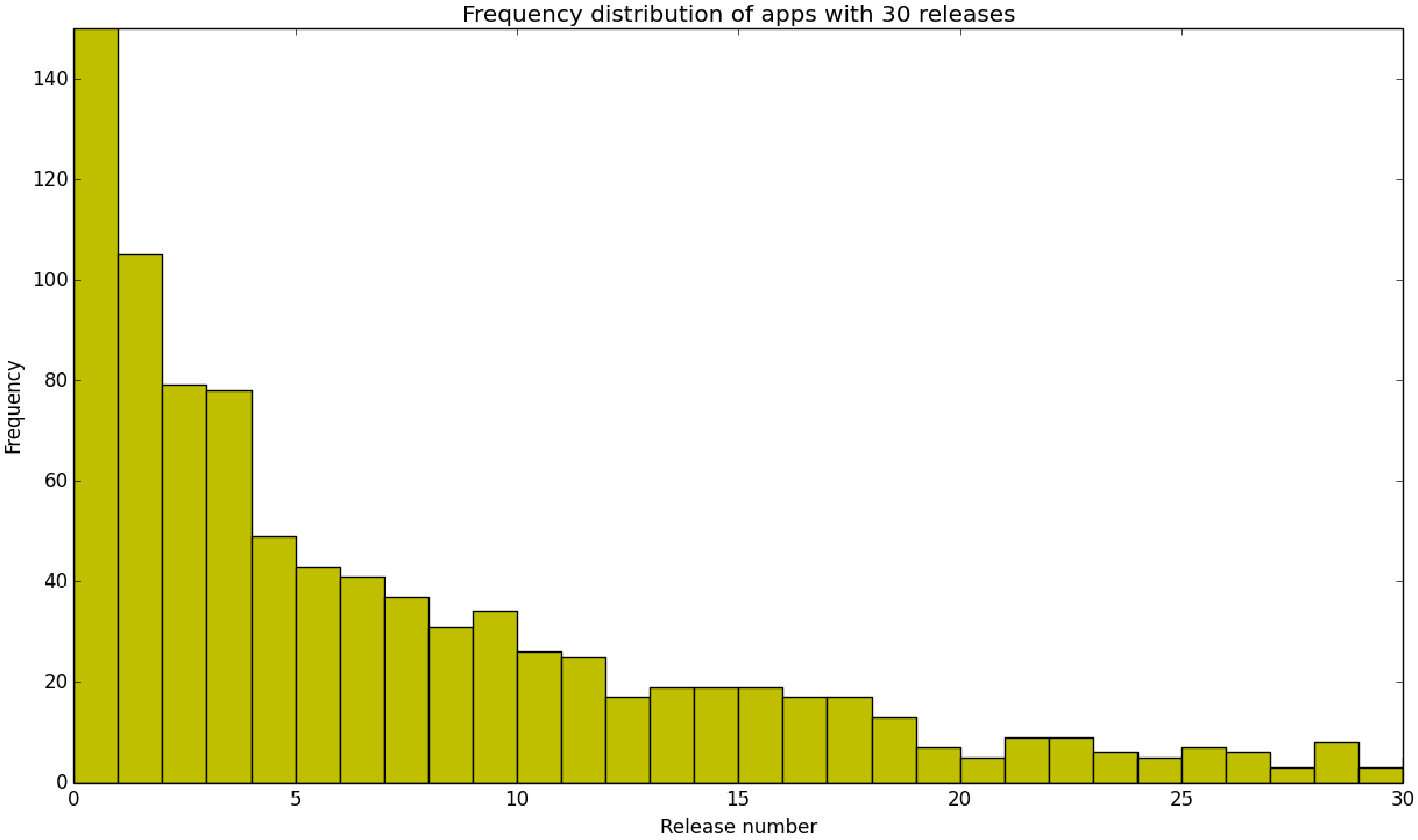
commit id	commit date	commit message
95901b8b60cb29c6c9d3cffdb7150cb65e34389e	2012-03-14 18:09:51-07:00	initial commit
fbae248ff3262162204cfdba784f242d9e2c68bc	2012-03-15 20:47:21-07:00	clipping bugfix
f8fce7d26fbdab6d52db4cf6082c4ed9ab424a50	2012-03-15 21:43:31-07:00	sexy focus polish
3714cafd3fcf9c3582a7fa0686e390a5c684cabe	2012-03-15 22:38:35-07:00	condense train display by destination
789a7a9cdd9ff317c354bf81b52b789ac92984e2	2012-03-16 17:35:55-07:00	route response parsing
0c9d7cdbd684fec8c18df277b96c3fbde6f6e31a	2012-03-16 19:09:57-07:00	initial route display
228b41568dc7bf2027fe9e0b782a5dfd5996d967	2012-03-19 01:27:20-07:00	route display, special schedule and warning messages
cb355cd071f187d188314f36c7ccd4208d467440	2012-03-19 19:41:10-07:00	tweak
8a30db365c34c525daca6df7d83aaca0802946dd	2012-03-25 00:05:04-07:00	special message display and animations
f4da1b1075c6aa0d4b6472ccc2d01dd91017da0b	2012-03-25 21:40:28-07:00	visual refinements and sjxp inclusion
ceadb1ad22e9baf637dd98e4e494cd939d3d2a6e	2012-03-25 21:42:42-07:00	add sjxp javadoc
7fc935e849b548dbad39fd2dfe9a60e062febf5f	2012-03-25 21:46:06-07:00	properly link to sjxp in eclipse project setting
bdbcfde194c5a325dd7af0aa6be115f30011630f	2012-03-25 21:54:14-07:00	icon
d9188b723c557e61f6a4410bef27a5c5eaa85bd5	2012-03-25 21:57:53-07:00	Merge pull request #1 from Miserlou/master Icon v1

Observations

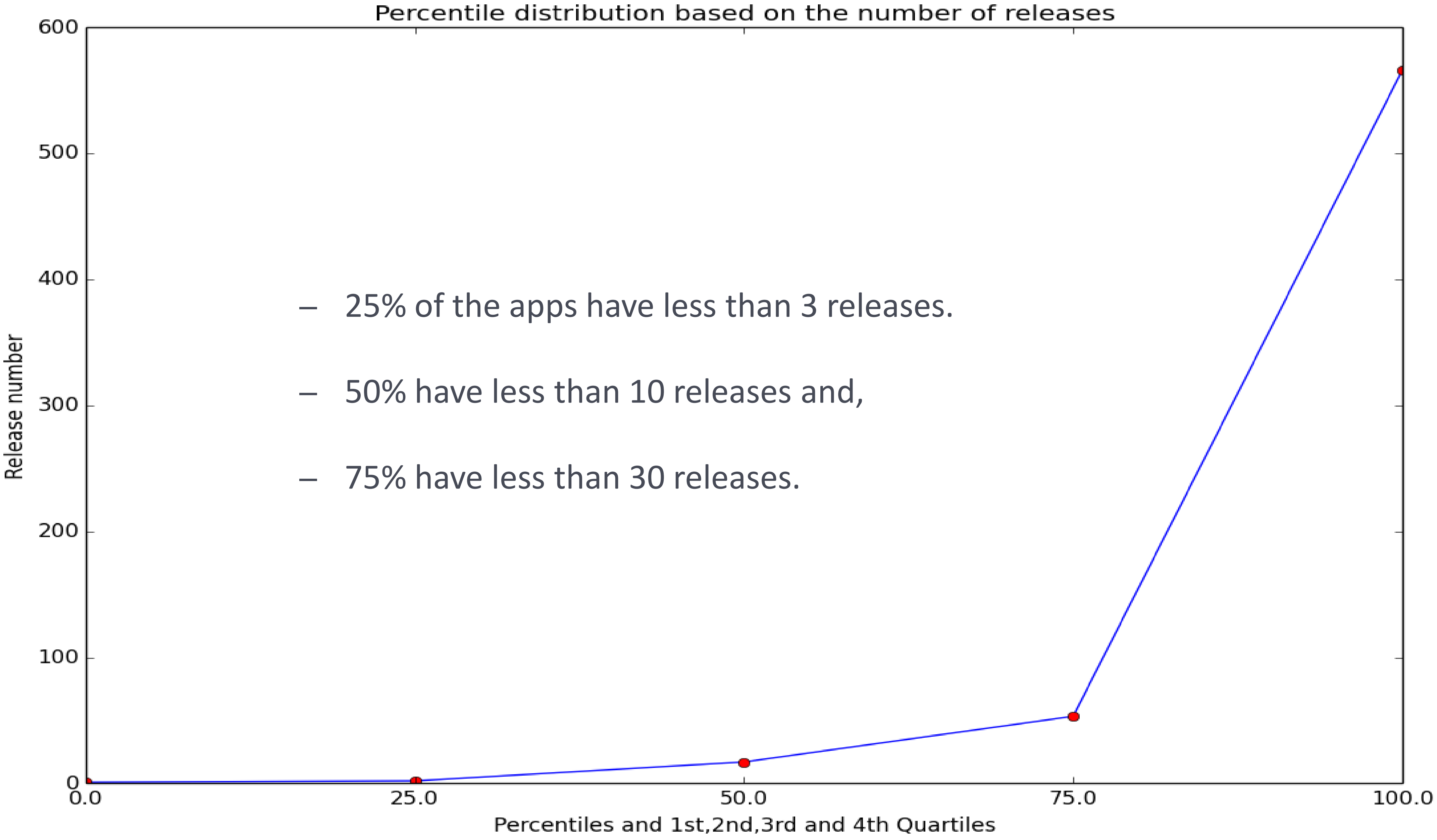
- We observe that the frequency of release is considerably small for apps with large number of releases.

- If we dig deeper it is clear from the data provided to us, that apps with less than 40 total releases tend to have higher frequencies.

- Lets see the following graph for 30 total releases. For clarity we’ve scaled the y axis to show a frequency of up to 150 apps and the x axis up to 30 releases.

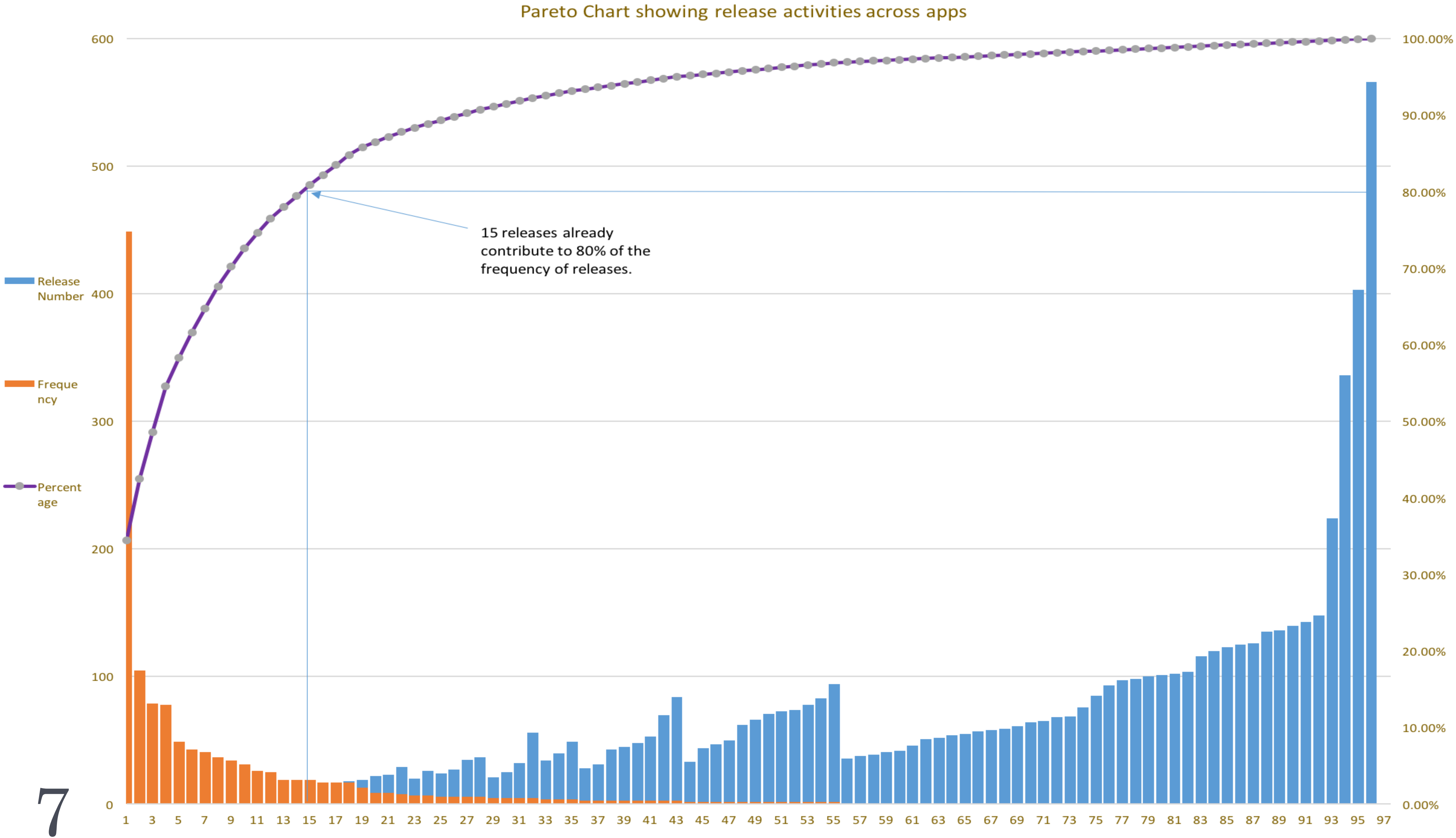


Histogram generated using Python Script



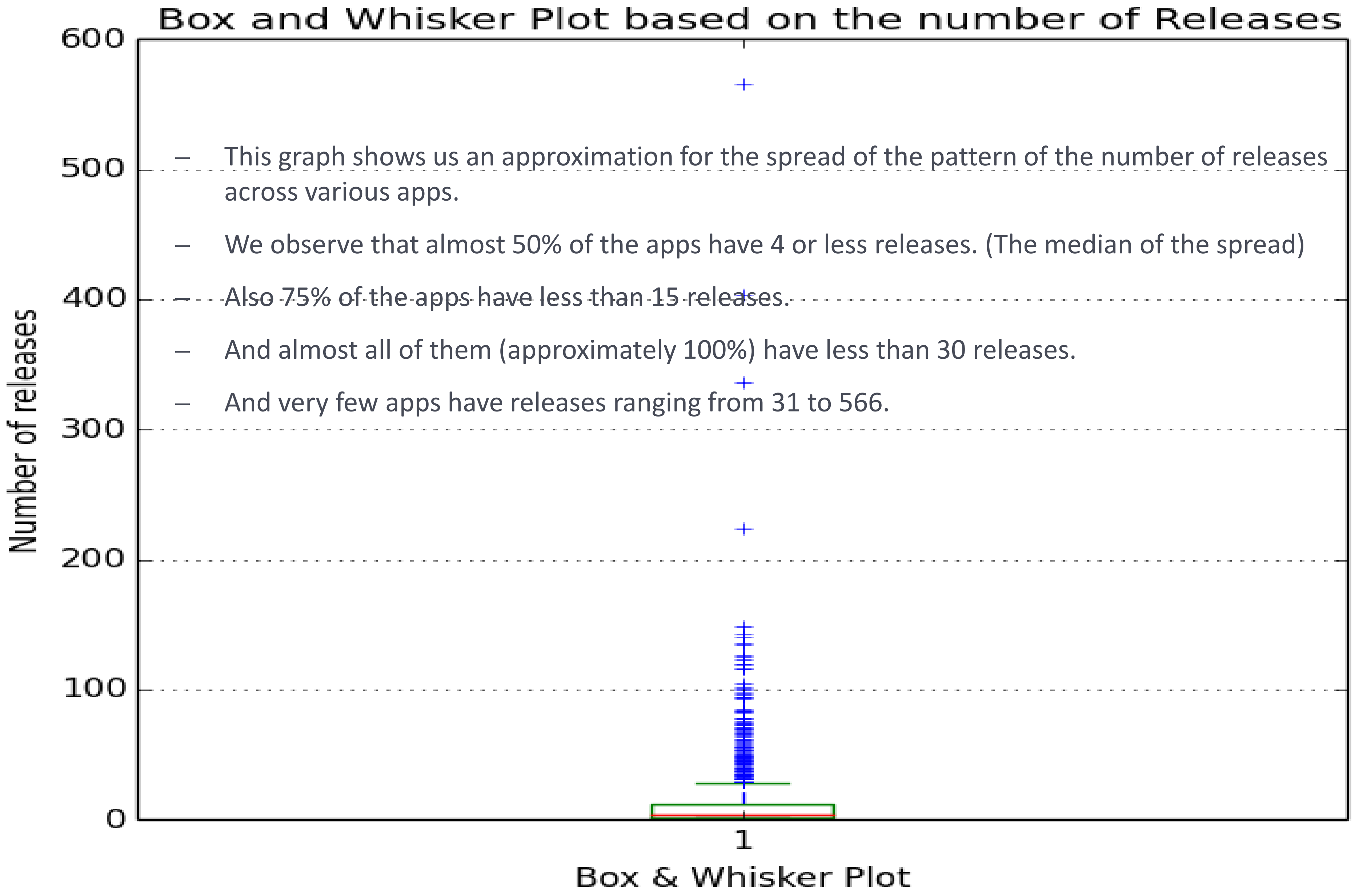
- 25% of the apps have less than 3 releases.
- 50% have less than 10 releases and,
- 75% have less than 30 releases.

Graph generated using Python Script

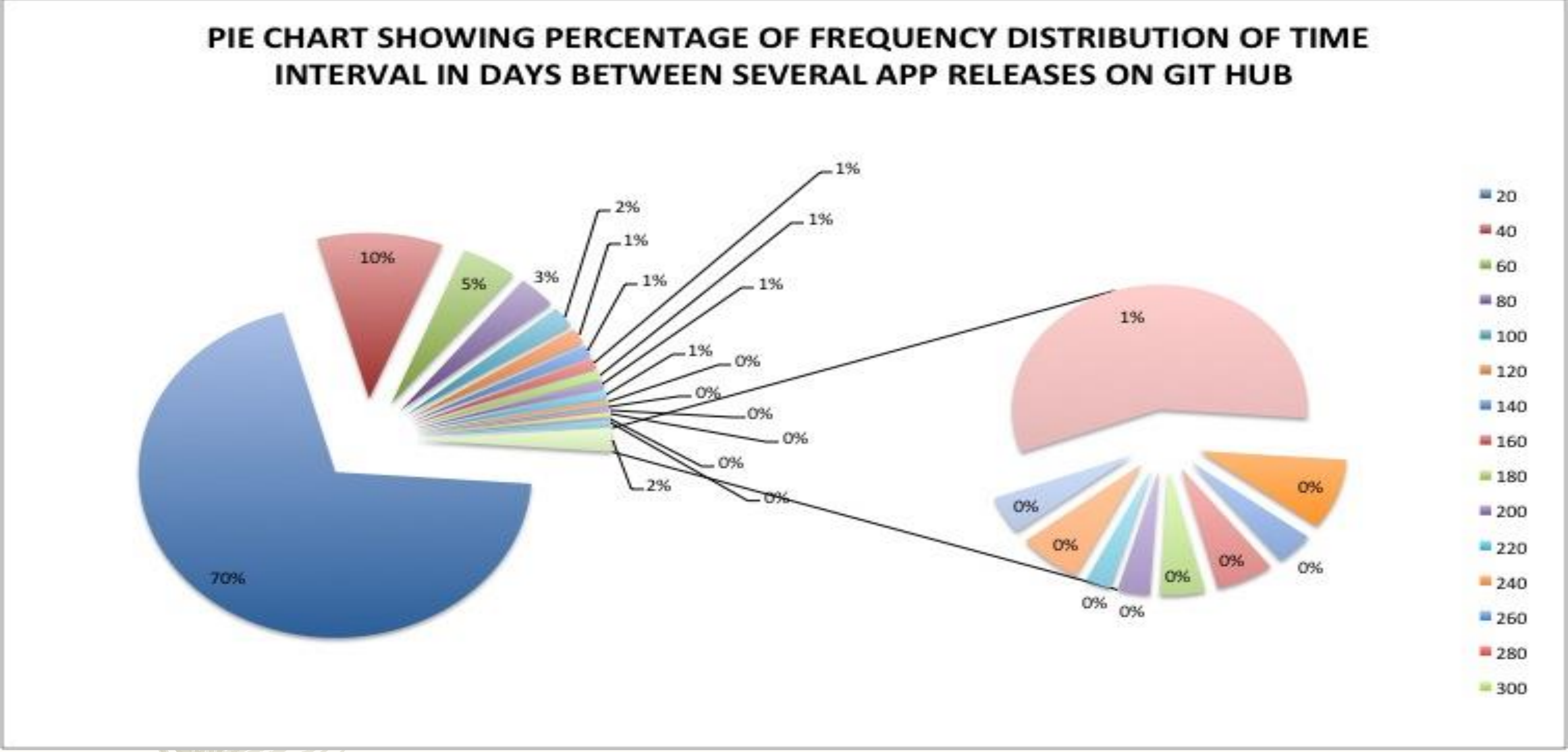


Pareto Chart showing release activities across apps

15 releases already contribute to 80% of the frequency of releases.



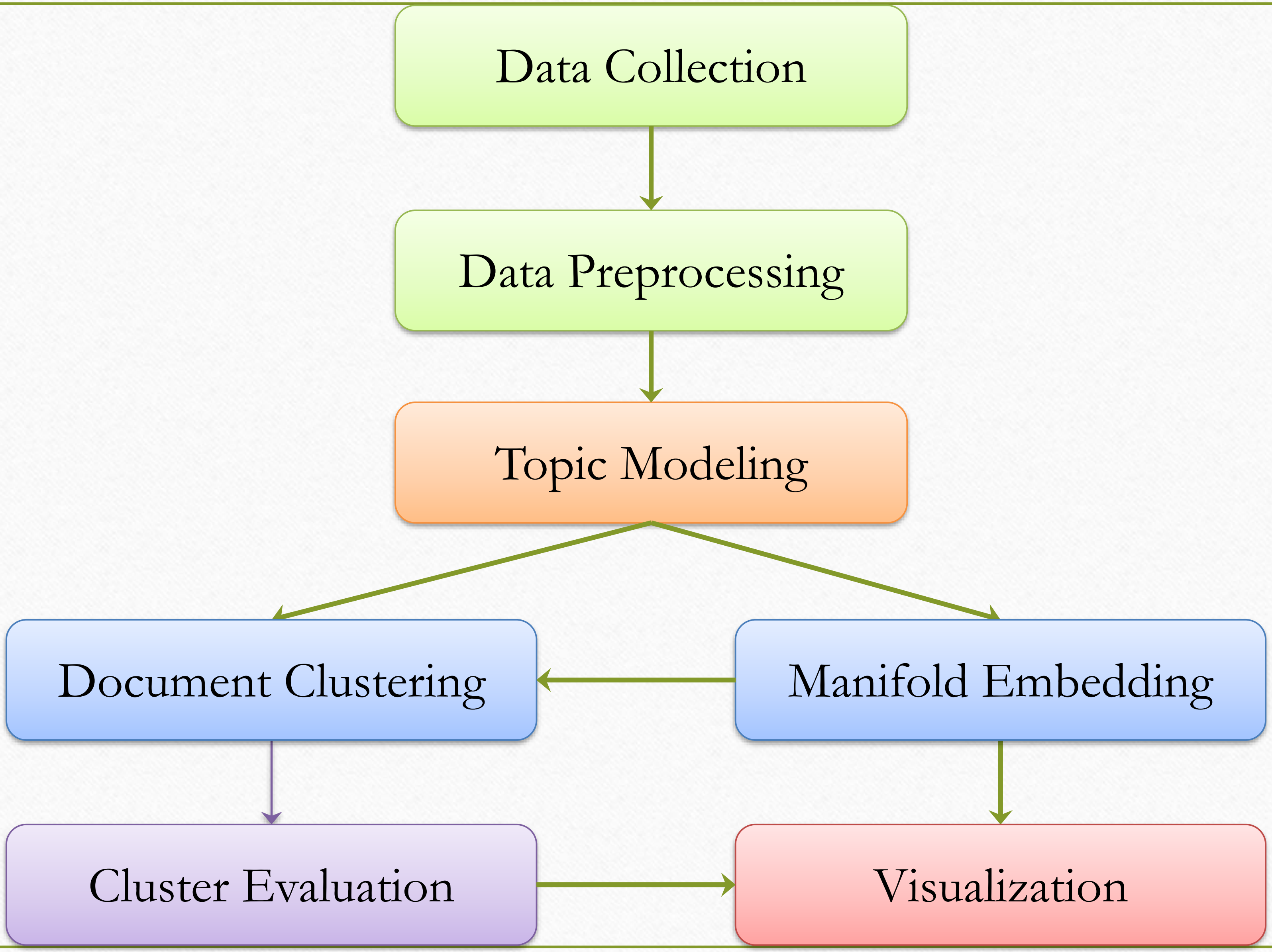
- This graph shows us an approximation for the spread of the pattern of the number of releases across various apps.
- We observe that almost 50% of the apps have 4 or less releases. (The median of the spread)
- Also 75% of the apps have less than 15 releases.
- And almost all of them (approximately 100%) have less than 30 releases.
- And very few apps have releases ranging from 31 to 566.



PIE CHART SHOWING PERCENTAGE OF FREQUENCY DISTRIBUTION OF TIME INTERVAL IN DAYS BETWEEN SEVERAL APP RELEASES ON GIT HUB

Pie Chart generated using MS Excel

RQ2


Data Collection

Data Preprocessing

Topic Modeling

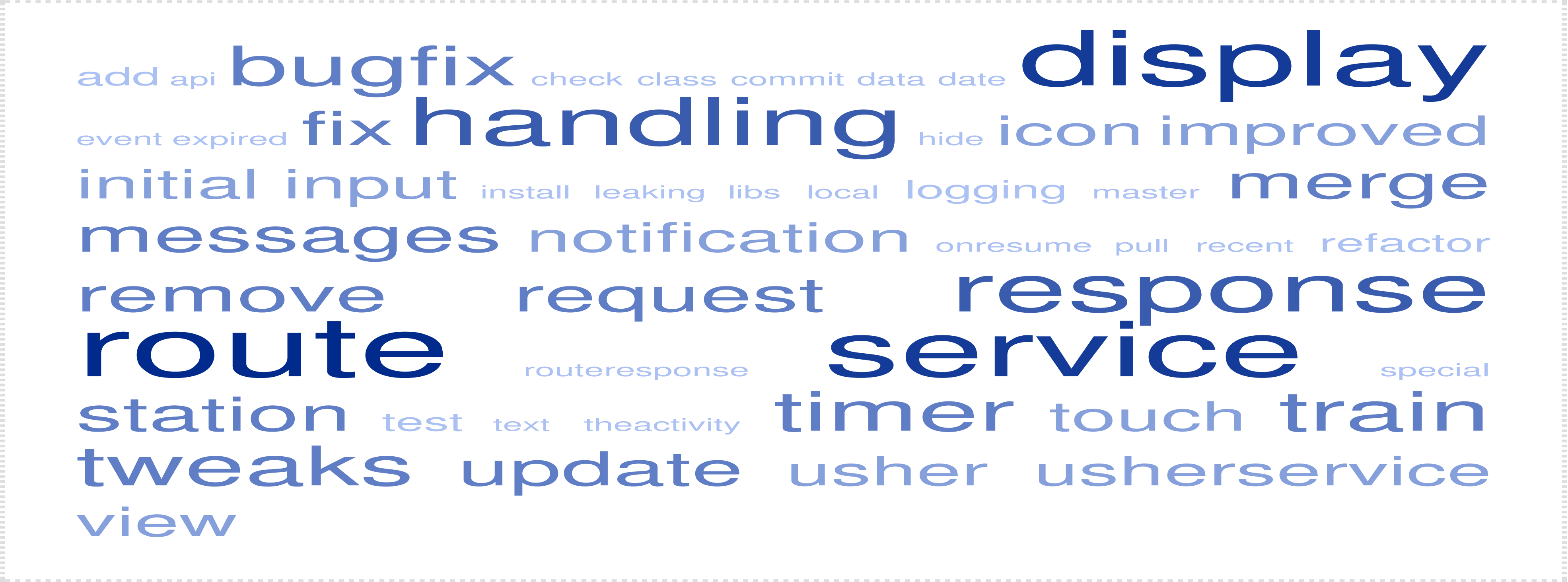
Document Clustering

Manifold Embedding

Cluster Evaluation

Visualization

A word Cloud for our messages



Data Preprocessing

1. PDF to TXT conversion
2. Text tokenization
3. Text normalization
   - Upper to lower case
4. Lemmatization
5. Feature selection
6. Serialization
   - Dictionary
   - Bag-of-word model

python

NLTK

gensim

The results are as follows. As you can see, each topic is made up of a mixture of terms. Thus Topic Modelling is done:

```
i.    2016-11-06 19:19:18,162 : INFO : topic #0 (0.200): 0.045*"usher" + 0.040*"service" + 0.036*"usherservice" + 0.035*"initial" +
      0.031*"updaterouteresponsewithetd" + 0.030*"route" + 0.027*"view" + 0.026*"bugfix" + 0.024*"timers" + 0.024*"leaking"

ii.   2016-11-06 19:19:18,163 : INFO : topic #1 (0.200): 0.038*"->" + 0.037*"train" + 0.034*"bart" + 0.033*"new" + 0.033*"special" +
      0.032*"refactor" + 0.032*"display" + 0.031*"fix" + 0.028*"destination" + 0.028*"usherservice"

iii.  2016-11-06 19:19:18,163 : INFO : topic #2 (0.200): 0.040*"notification" + 0.036*"stations" + 0.035*"main" + 0.033*"begin" +
      0.032*"route" + 0.028*"response" + 0.027*"recent" + 0.026*"about" + 0.026*"messages" + 0.025*"tweakin"

iv.   2016-11-06 19:19:18,164 : INFO : topic #3 (0.200): 0.041*"from" + 0.041*"add" + 0.040*"icon" + 0.034*"service" + 0.033*"improved"
      + 0.031*"merge" + 0.029*"handling" + 0.029*"crittercism" + 0.026*"update" + 0.024*"response"

v.    2016-11-06 19:19:18,164 : INFO : topic #4 (0.200): 0.050*"service" + 0.038*"visual" + 0.037*"check" + 0.030*"remove" + 0.030*"sjxp"
      + 0.027*"properly" + 0.026*"timer" + 0.024*"test" + 0.023*"install" + 0.023*"allow"
```

Top 5 terms for topic #0: usher, service, usherservice, initial, updaterouteresponsewithetd

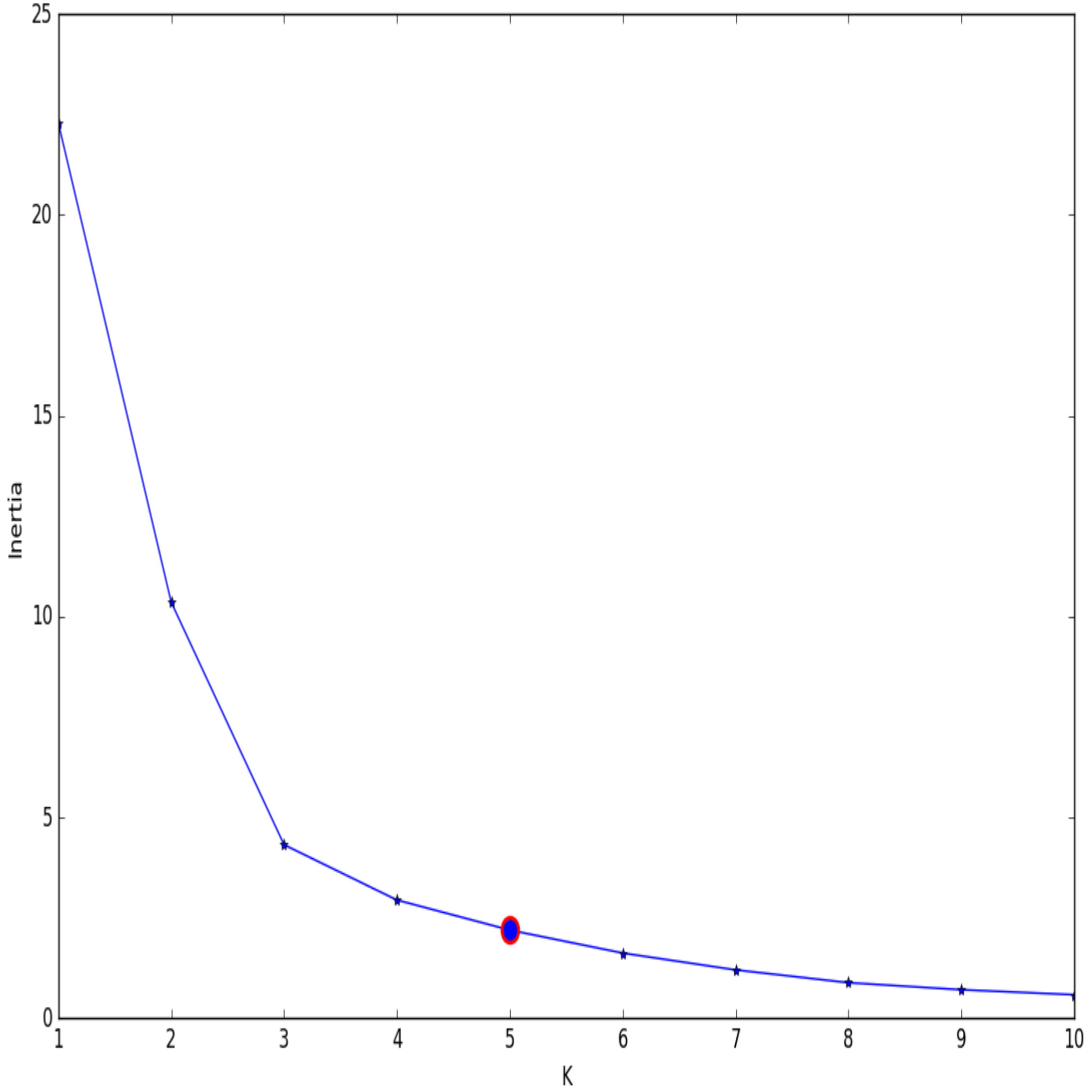
Top 5 terms for topic #1: ->, train, bart, new, special

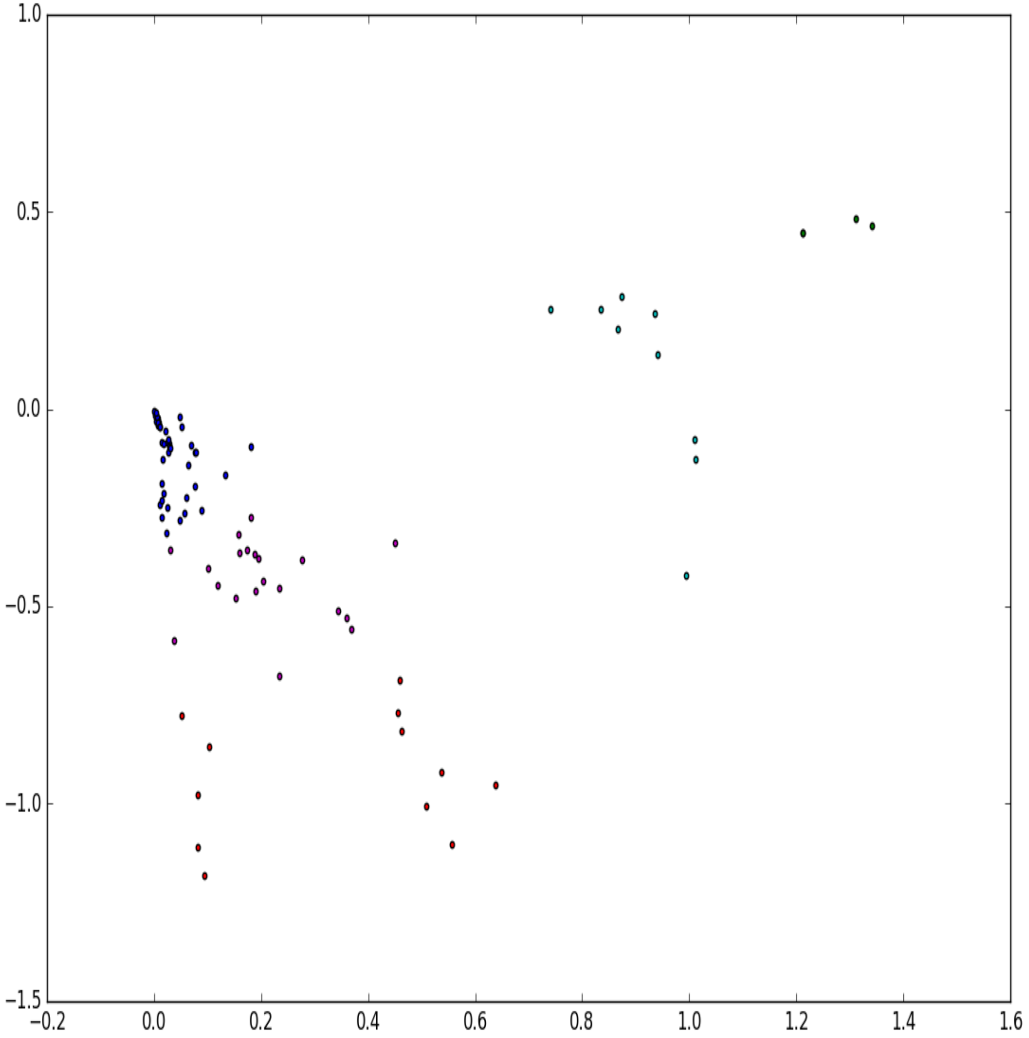
Top 5 terms for topic #2: notification, stations, main, begin, route

Top 5 terms for topic #3: from, add, icon, service, improved

Top 5 terms for topic #4: service, visual, check, remove, sjxp

The K Means Graph(gives us K=5) and generated clusters





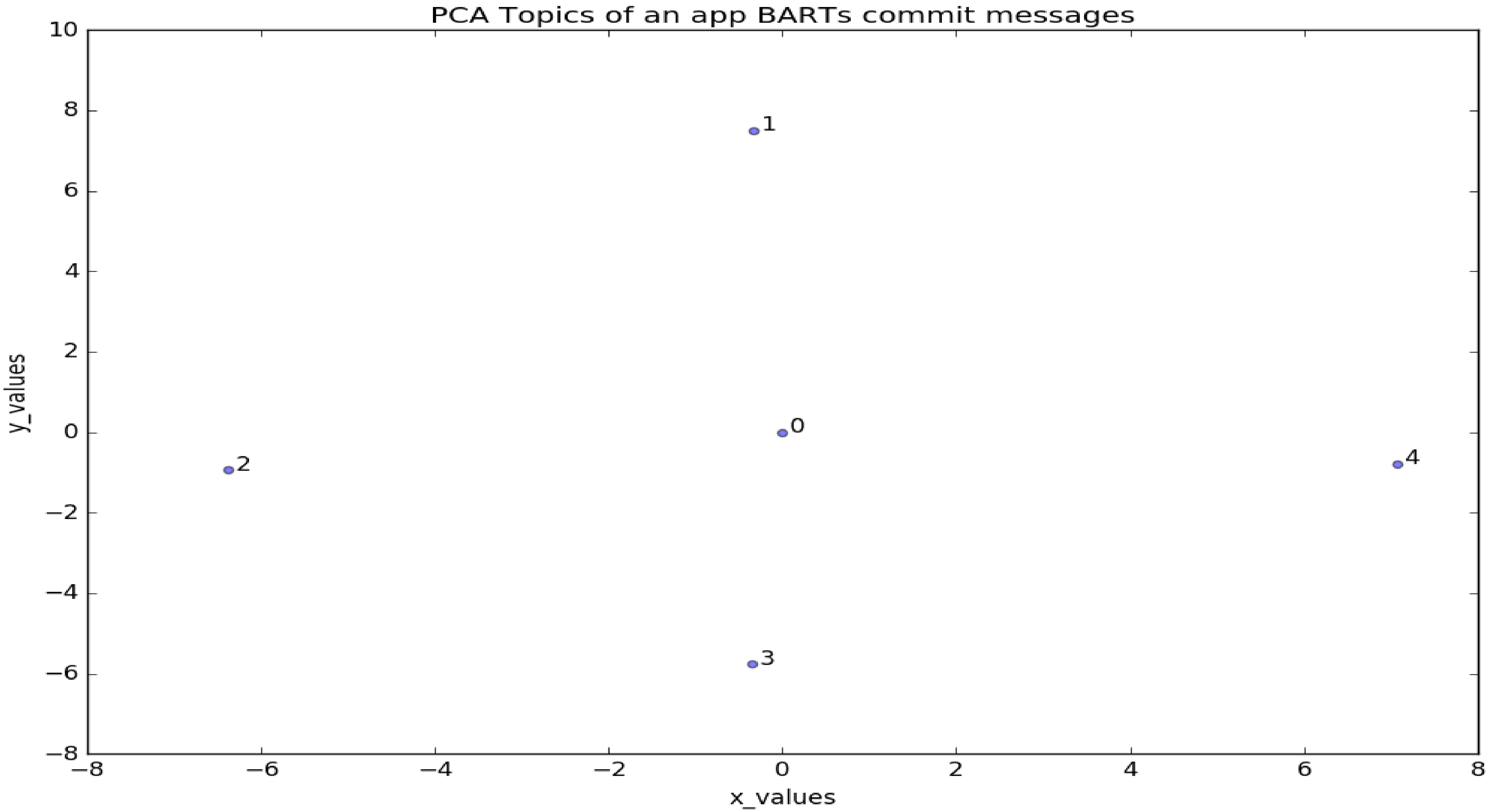
Manifold Learning

- LSI uses a Singular Value Decomposition (SVD) of the term-document matrix X to identify a linear subspace (so-called latent semantic space) that captures most of the variance in the data set. [1]

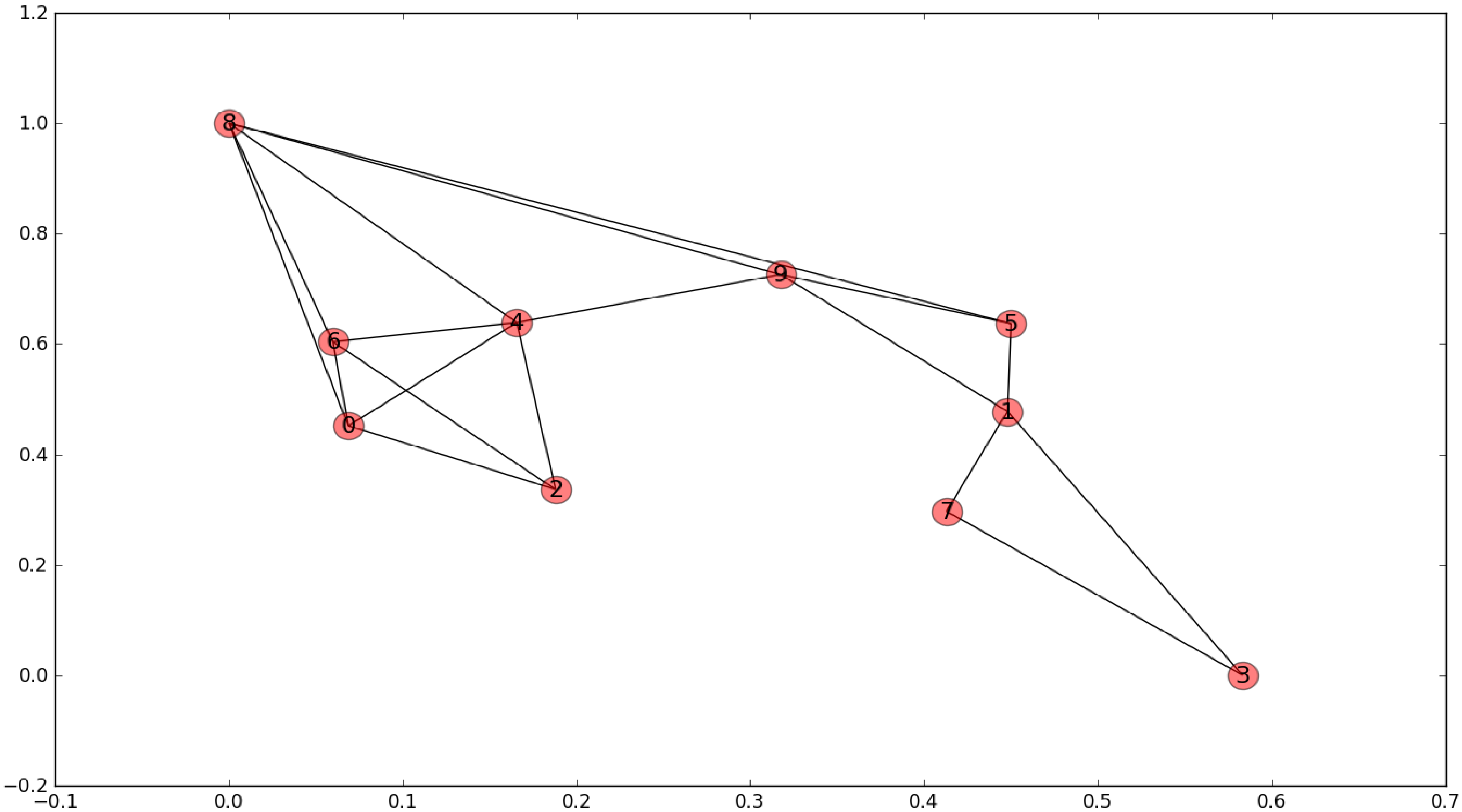
- PLSI models each word in a document as a sample from a mixture model, where the mixture components are multinomial random variables that can be viewed as representations of “topics.” [1]

- We need to use LapPLSI, in order to visualize the hidden topics in the document. [1]

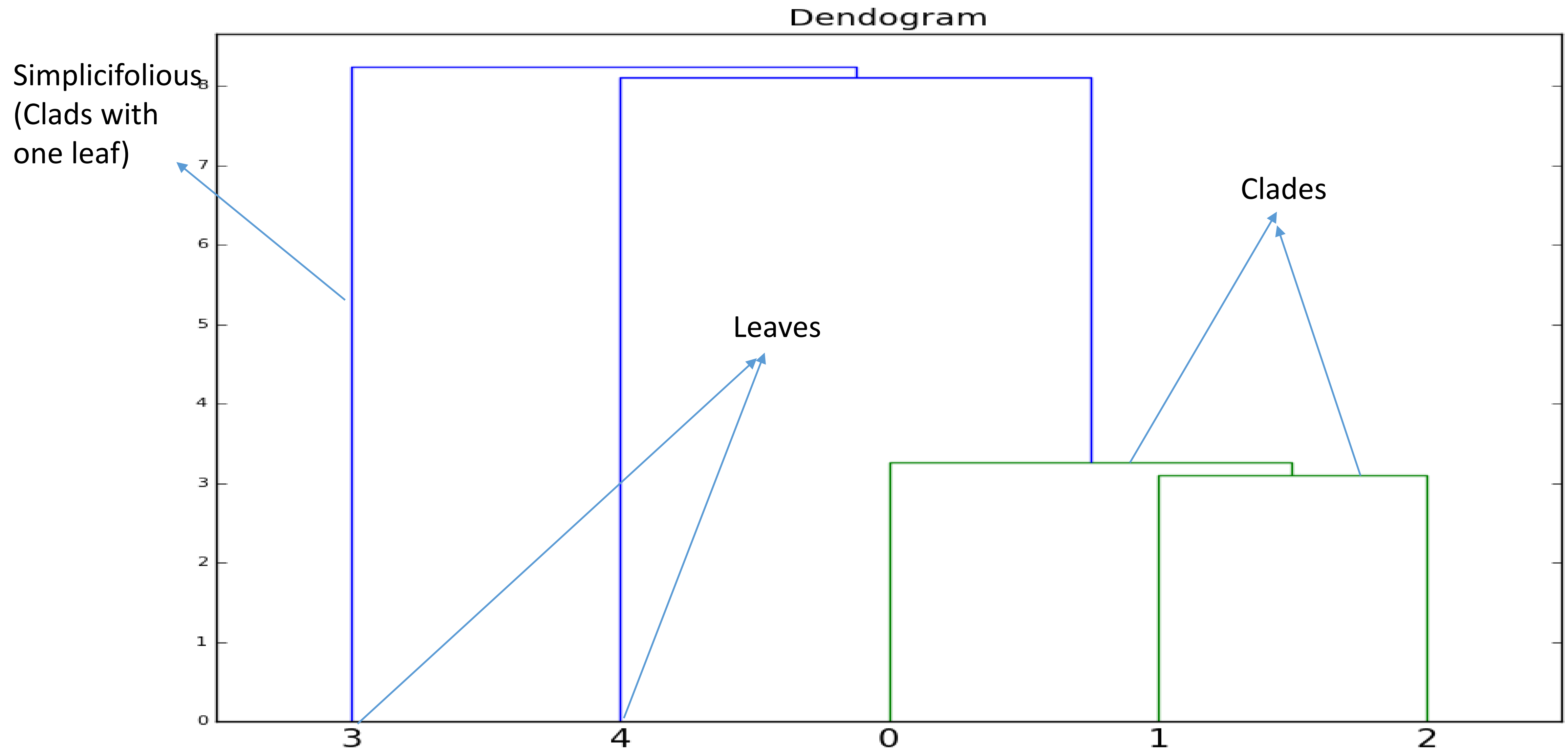
PCA graph (Topics)



Network Analysis by increasing the no. of topics to 10



Step 11: Hierarchical cluster Analysis (BART app)



Simplicifolious (Clads with one leaf)

Leaves

Clades

RQ3

- RQ3 →
- What? Can we define lifecycle stages of an app based on the topics after the topic modelling process?
- Why? Would help us get a sense of the Software Development Lifecycle Process followed by the apps.
- Challenge? The topics modelled follow an unsupervised LDA machine learning technique, so how should we label the topics. [2]
- RQ4 (Future possibility) →
- Does popularity correlate with characteristics of a repository like **number of commits**? [5]

References

[1] Modeling Hidden Topics on Document Manifold (Deng Cai, Qiaozhu Mei, Jiawei Han, Chengxiang Zhai)

[2] Automated topic naming to support analysis of software maintenance activities (Abram Hindle ,Neil A. Ernst, Michael W. Godfrey ,John Mylopoulos)

[3] What's hot and what's not: Windowed developer topic analysis. (A. Hindle, M. W. Godfrey, and R. C. Holt)

[4] https://radimrehurek.com/gensim/

[5] Understanding the Factors that Impact the Popularity of GitHub Repositories (Hudson Borges, Andre Hora, Marco Tulio Valente)

[6] The key technology of topic detection based on K-means (Shengdong Li ; Xueqiang Lv ; Tao Wang ; Shuicai Shi)