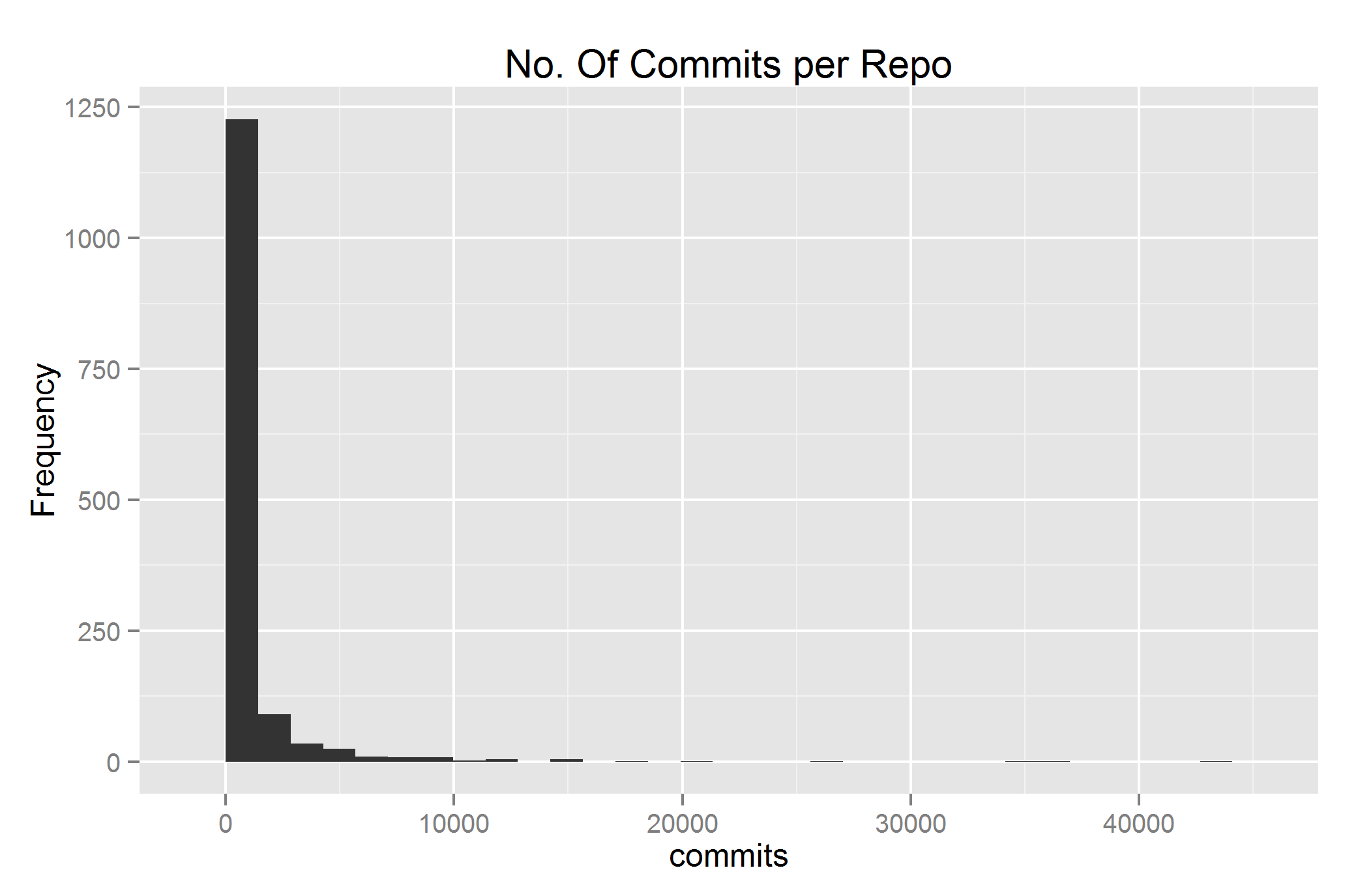
Name: Siti Nurafifah & Alice Yap

Report

To start off with the analyzing, we need to find out which set of repo to analyze from. To proceed with that, we divided the parameters and find out each pearson and spearman.

Firstly, we compare the relationship between the repository and the number of commits. Looking at the histogram, it is a right skewed distribution where the mean is greater than the median.



Then, we look at the median and the third quartile. To find the third quartile, we have to range the number of commits in order. To do this, in R:

attach(completeoverview) # overview for repo-pre-analysis

x <- completeoverview$commits[1:1418]

sort(x) # sort the data by the number of commits

quantile(x) # find the third quartile

The output given was:



This shows that the third quartile is 545.25. After finding out the third quartile, we have to subset the data according to the set of repo which consist of the number of commits more than the third quartile value.

To do this, using sql queries:

select r.id as 'repository'

from repositories r

left outer join scmlog s on s.repository\_id = r.id

group by r.id

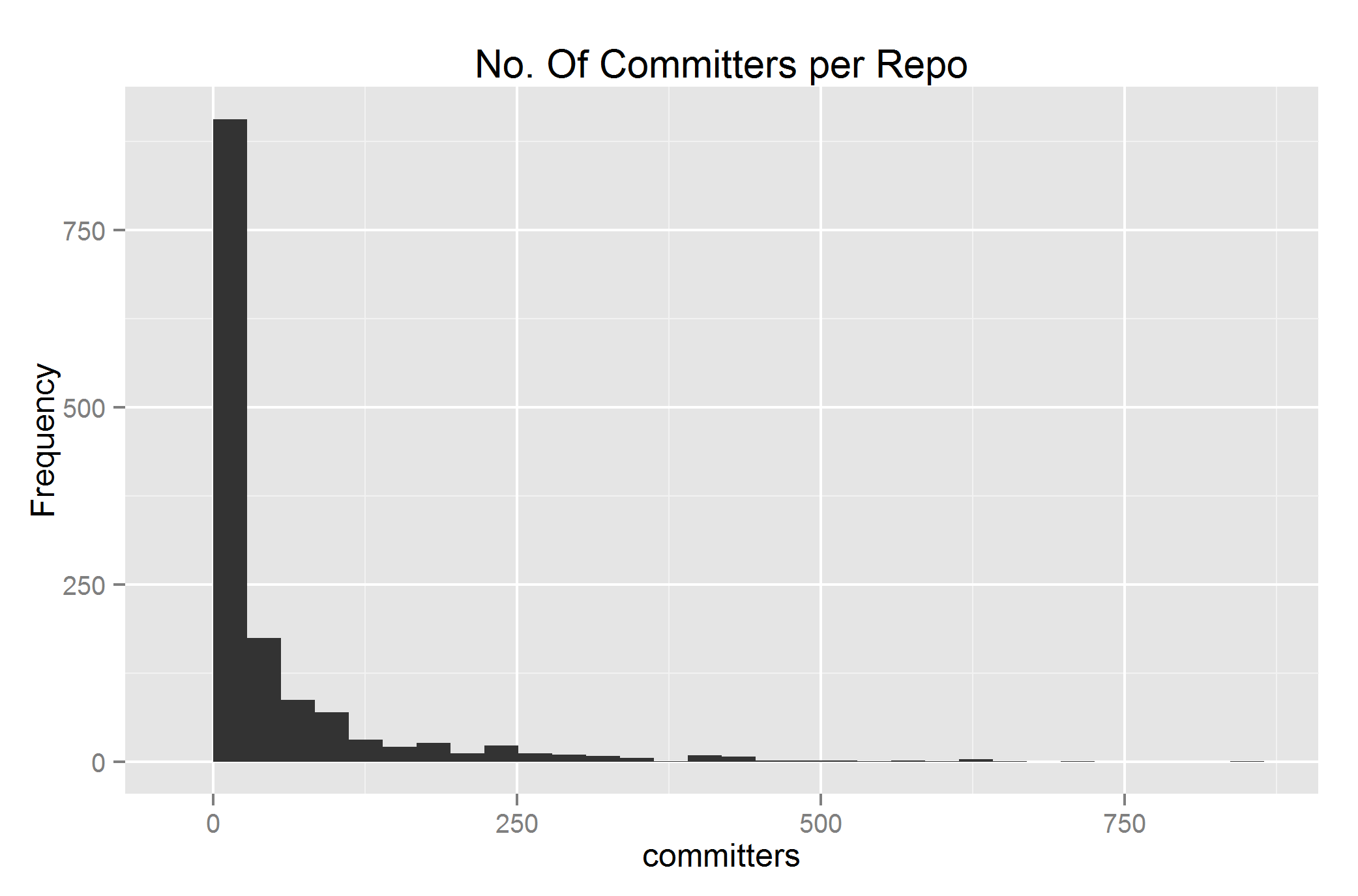
having count(distinct s.date) > 545;

(355 rows returned)

With this, it subsets the set of repository according to the number of commits more than the third quartile. The link below is a csv file of the set of repository of the results above:

<https://www.dropbox.com/s/kyl11dm9vvju9ng/analysis.csv?dl=0>

Secondly, we look at the relationship between the number of committers with each repository and repeat the steps above.



Then, we find out the third quartile by:

attach(completeoverview) # overview for repo-pre-analysis

x <- completeoverview$committers[1:1418]

sort(x) # sort the data by the number of commits

quantile(x) # find the third quartile

The output given was:



This shows that the third quartile of the number of committers is 52. With this, we can find out the set of repository with the number of committers more than the third quartile.

With the help of MySQL workbench:

select r.id as 'repository'

from repositories r

left outer join scmlog s on s.repository\_id = r.id

group by r.id

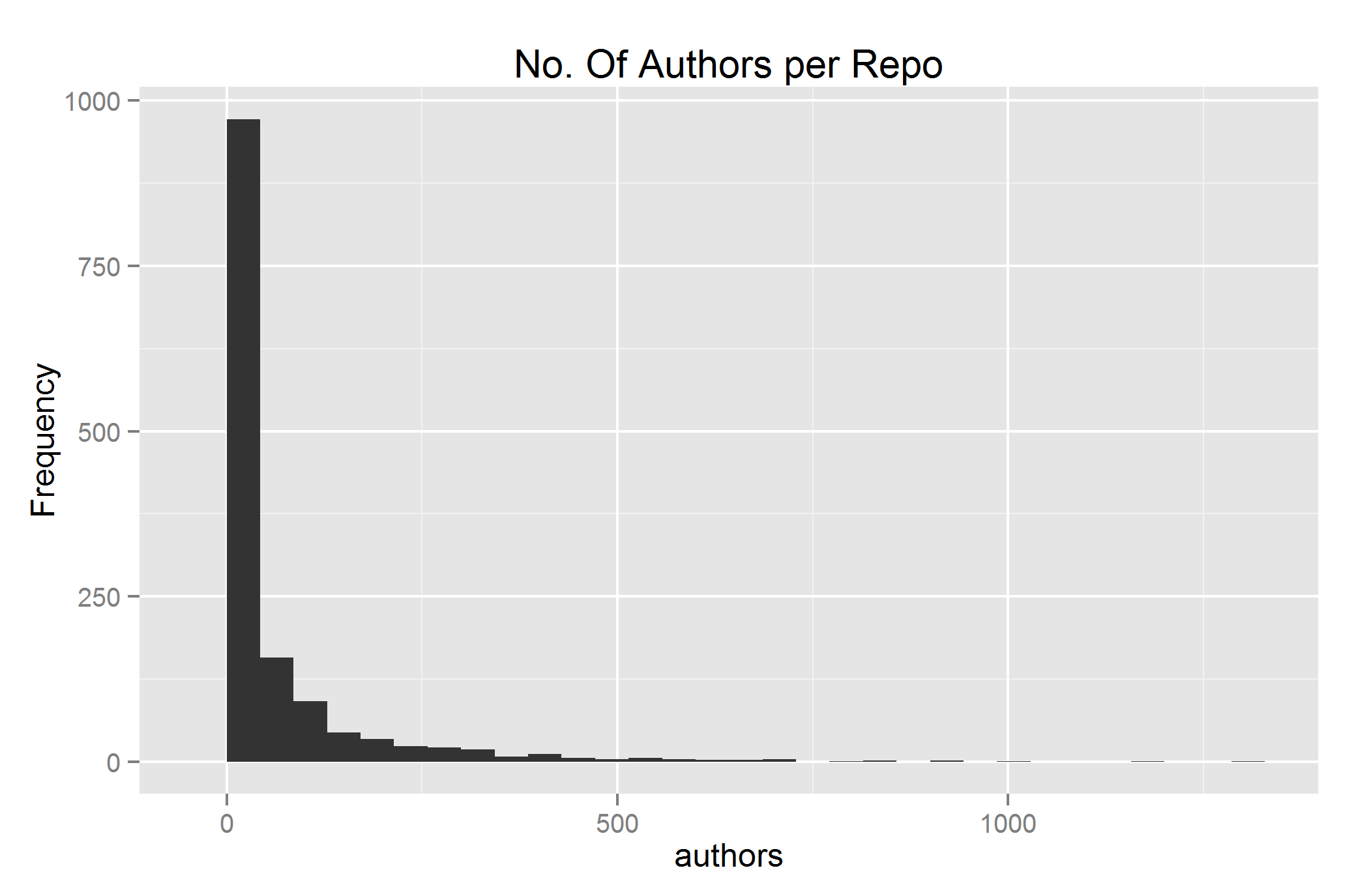
having count(distinct s.committer\_id) > 52;

(353 rows returned)

With this, it subsets the set of repository according to the number of committers more than the third quartile. The link below is a csv file of the set of repository of the results above:

<https://www.dropbox.com/s/kiau6pqt68o6rjb/secAnalysis.csv?dl=0>

Thirdly, we compare the relationship between the number of authors against the repository.



Looking at the histogram, we find out the third quartile since it is right skewed distribution.

To find the third quartile, we have to sort the authors. With this, we use R functions:

attach(completeoverview) # overview for repo-pre-analysis

x <- completeoverview$authors[1:1418]

sort(x) # sort the data by the number of authors

quantile(x) # find the third quartile

The output given was:



This shows that the third quartile is 62.75. With this result, we can subset the set of repository with the number of authors more than the third quartile value.

select r.id as 'repository'

from repositories r

left outer join scmlog s on s.repository\_id = r.id

group by r.id

having count(distinct s.author\_id) > 63;

(347 rows returned)

With this, it subsets the set of repository according to the number of authors more than the third quartile. The link below is a csv file of the set of repository of the results above:

<https://www.dropbox.com/s/ph13fi0jd7c9oa0/thirdAnalysis.csv?dl=0>

Lastly, we compare the number of files for each repository.

To do this, we find out the third quartile of the number of files by:

attach(completeoverview) # overview for repo-pre-analysis

x <- completeoverview$files[1:1418]

sort(x) # sort the data by the number of authors

quantile(x) # find the third quartile

The output given was:



This shows that the third quartile is 269.75. With this, we can subset the number of repository having the number of files more than the third quartile value.

select r.id as 'repository'

from repositories r

left outer join files f

on r.id = f.repository\_id

group by r.id

having count(distinct f.file\_name) >270;

(354 rows returned)

With this, it subsets the set of repository according to the number of files more than the third quartile. The link below is a csv file of the set of repository of the results above:

<https://www.dropbox.com/s/77l39rnklz4vddq/forthAnalysis.csv?dl=0>

The next step on choosing which parameter to focus from is to look at the calculation between Pearson and spearman.

This table shows the Pearson calculation between the relationships.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pearson calc. | Commits | Committers | Authors | Files |
| Commits | - | 0.76 | 0.82 | 0.68 |
| Committers | 0.75 | - | 0.99 | 0.46 |
| Authors | 0.82 | 0.99 | - | 0.50 |
| Files | 0.68 | 0.46 | 0.50 | - |

This table shows the Spearman calculation between the relationships.

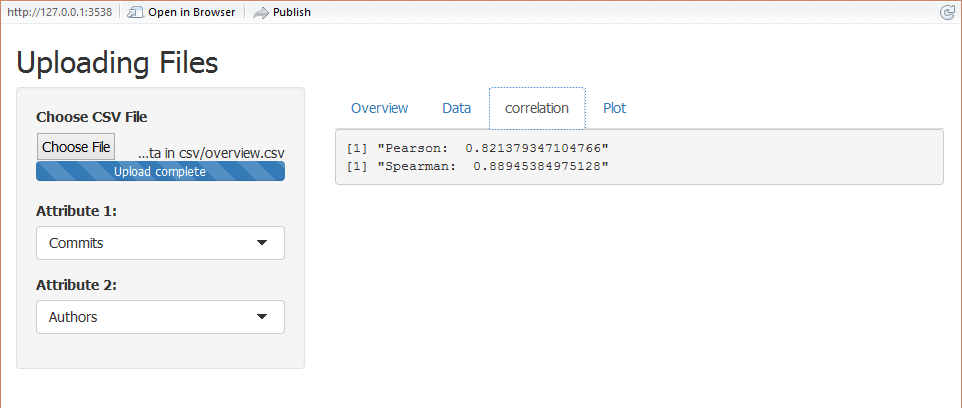
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Spearman calc. | Commits | Committers | Authors | Files |
| Commits | - | 0.87 | 0.89 | 0.82 |
| Committers | 0.87 | - | 0.99 | 0.73 |
| Authors | 0.89 | 0.99 | - | 0.74 |
| Files | 0.82 | 0.73 | 0.74 | - |

To look at the data individually:

Commits & Authors:

Pearson: 0.82

Spearman: 0.89

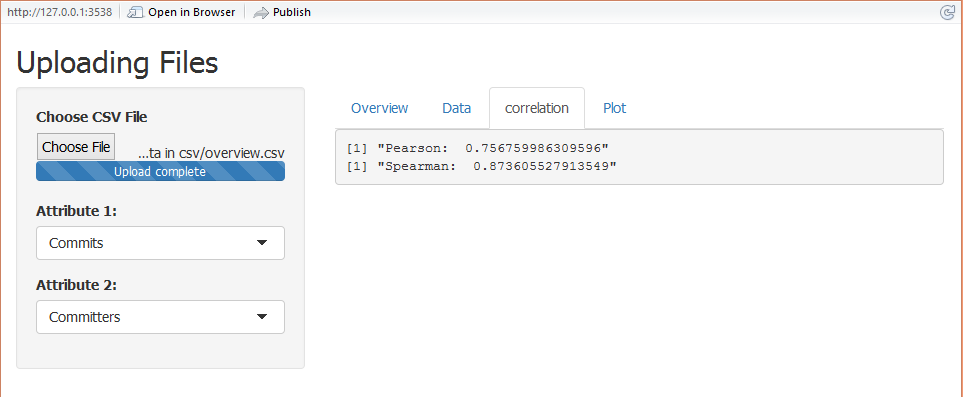


As seen above, the Pearson and Spearman between Commits and Authors are close to 1, which suggest that the correlation is strong.

Commits & Committers

Pearson: 0.76

Spearman: 0.87

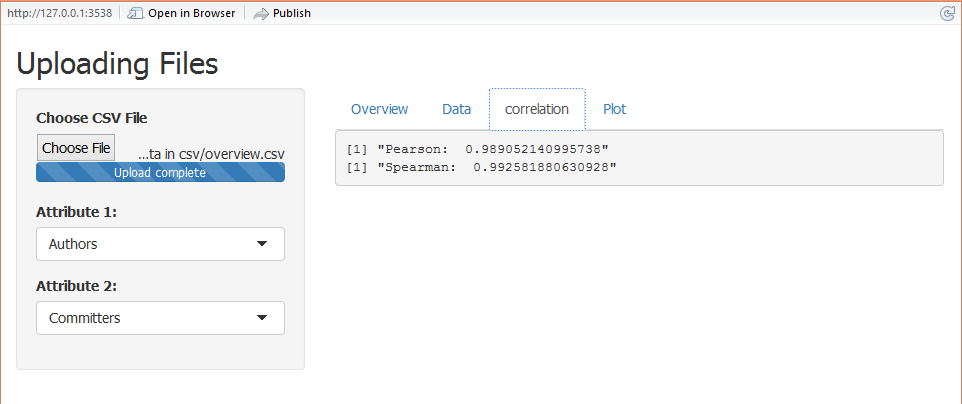


As seen above, the Pearson and Spearman between Commits and Committers are close to 1 however not as close as compared to Commits and Authors, which suggest that the correlation is weaker than between Commits and Authors.

Authors & Committers

Pearson: 0.99

Spearman: 0.99



As seen above, the Pearson and Spearman between Commits and Committers are the closest to 1, which suggest that the correlation is the strongest.

To conclude this, we compare the strongest relationship between each other. The relationship between committers and authors have the strongest amongst the others.

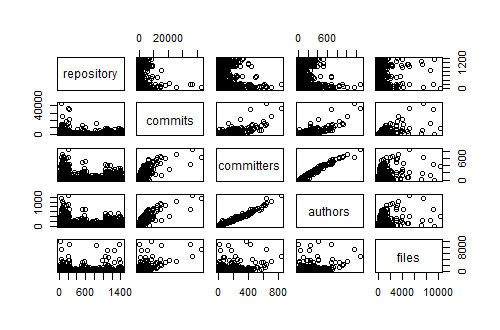
With this, we can say the parameter to look at is the number of authors as it has the strongest relationship with the others by looking at the correlation.

Regression

Before fitting our regression model we want to investigate how the relationship of the variables related to one another. To do this, we can graphically construct scatter plots of all pair-wise combinations of variables in the data frame. This can be done by typing:

Scatterplot:



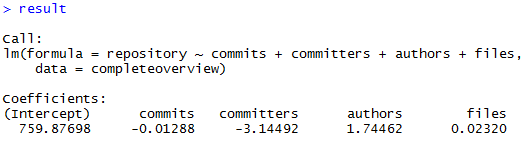


To fit a multiple linear regression model with repository as the response variable and

Number of commits, committers, authors as the explanatory variables, use the command:



When calling the function result:

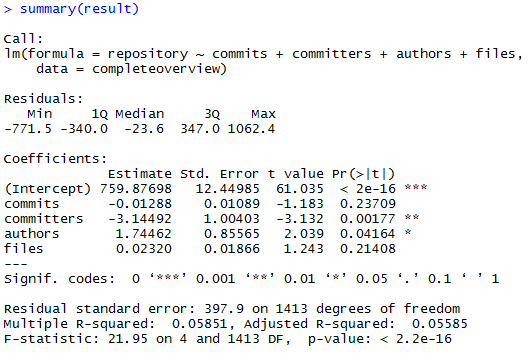


This output indicates that the fitted values is given by

= 759.87698 -0.01288 -3.14492 + 1.74462 + 0.02320

We begin by testing whether the explanatory variables collectively have an effect on the response variable. If we can reject this hypothesis, we continue by testing whether the individual regression coefficients are significant.

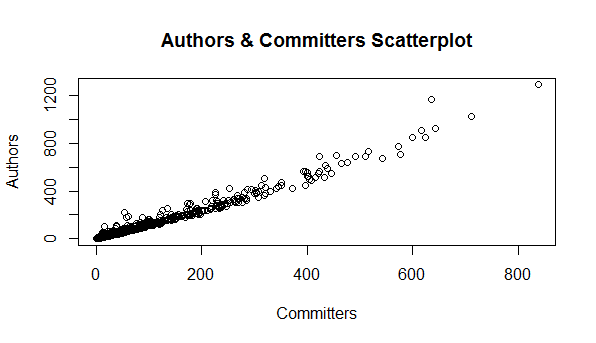
We can test out by typing:



The output shows that F=21.95 (p < 2.2e-16), indicating that we should clearly reject the null hypothesis that the variables commits and files have no effect on repository. The result also shows that p-value for the variable commits(0.23709) and files(0.21408) is greater than the common alpha level of 0.05, which indicates that it is not statistically significant. In addition, the output also shows that and adjusted = 0.5585. Therefore, we should consider removing commits and files variables.

We are interested in the variables authors and committers relationship. We can construct a scatterplot of committers against authors by typing:





We then measure the strength of the linear relationship by typing:

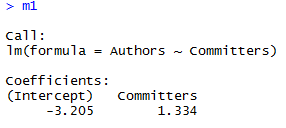


This high correlation between committers and authors suggests that these two effects are somewhat bounded together.

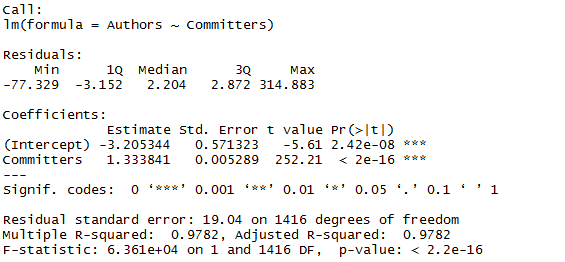
Running the regression:

The regression command is lm for linear model. We will store that model in a variable called m1. The order of the variables is dependent followed by a tilde "~" followed by a list of independent variables.





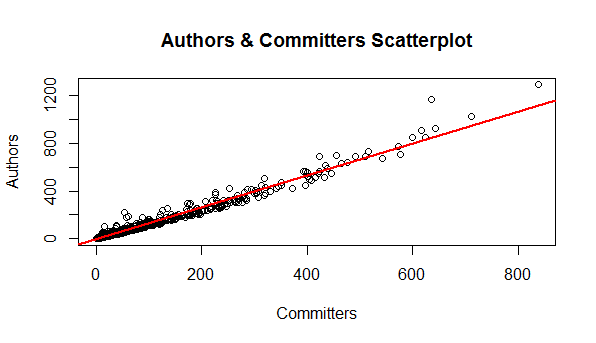




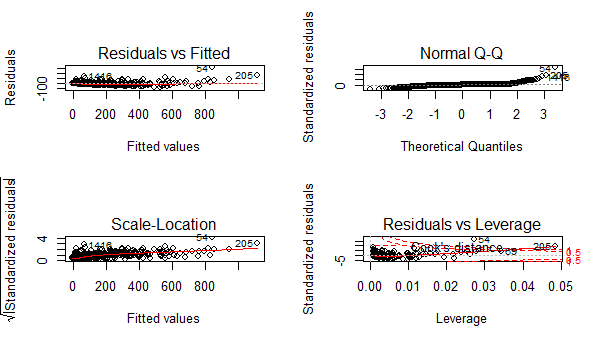
The regression equation is *authors* = -3.205344 + 1.333841\**committers.* The output shows that the p-value for that test is “2.42e-16”. In addition, the value is 0.9782 which shows that 97.8% of the variability in authors is explained by committers.

To plot the regression line on the scatterplot, type:

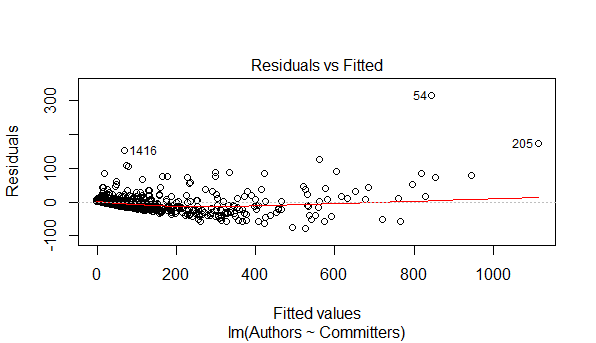


Then, we look at the diagnostic plot of the residuals:

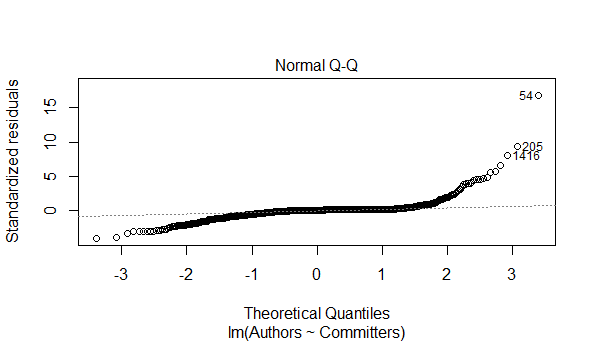




With the plots, we then focus on 2 plots, Residuals vs Fitted and Normal Q-Q plots.



As seen above, we can say that the regression line is flat which mean Authors can be expressed as a linear function of Committers and the variation of observations around the regression line is constant.



Seeing that the plot does not support normality, it seems that the distribution is right skewed with the reference of the histogram. In conclusion, there is a regression (linear) relationship between committers and authors.