Abstract

In this paper are analyzed commit history logs from a git repository. The dataset is created using a python script which saves logs in a csv file.

1. **The Dataset creation**

This paper uses custom dataset created from an open source repository AutoFixture [1]. This chapter contains general information about how input csv file was created.

## 1.1 Git log file

To get git logs history following command is used:

git log --numstat > AutoFixture.log

It saves log history in machine friendly format in AutoFixture.log file. An example of commit from AutoFixture.log file:

commit e64ce87243273e13faedb73ff2c1520ef0ed7b06

Author: Alex Povar <user.home.0000@gmail.com>

Date: Tue Aug 23 10:49:52 2016 +0300

Fix code style and add unit test

13 13 Src/AutoNSubstitute/NSubstituteVirtualMethodsCommand.cs

1 1 Src/AutoNSubstitute/SubstituteRequest.cs

100 65 Src/AutoNSubstituteUnitTest/AutoConfiguredFixtureIntegrationTest.cs

It contains commit id, Author full name and email, created date, commit comment and a list of modified files. First number before file name is number of added lines. Second number is number of removed lines. For example

100 65 Src/AutoNSubstituteUnitTest/AutoConfiguredFixtureIntegrationTest.cs

means that 100 new lines have been added and 65 have been removed to file.

**1.2 CSV file creation**

The log file obtained in previous step can’t be used directly in R. The file needs to be transformed to csv format. The data in Autofixture.log file is relational: one commit entry have many file changes. So, the relation between commit and changed file is **one to many**. The csv file will contain one entry for each changed file. In order to transform log file to csv a python script have been created. It can be found in src/python folder of repository [2]. The structure of csv file is following:

id, author, email, date, fileName, added, removed

Several lines of csv file:

id,author,email,date,fileName,added,removed

ab829640ed8e02776e4f4730d0e72ab3cc382339,Mark Seemann,mark@ploeh.dk, Mon Sep 5 15:21:45 2016 +0200,README.md,4,0

52da28891d60ede69d35df8d2040ef64890f17cd,Mark Seemann,mark@ploeh.dk, Wed Aug 31 18:21:21 2016 +0200,Src/AutoFakeItEasy/Properties/AssemblyInfo.cs,2,2

52da28891d60ede69d35df8d2040ef64890f17cd,Mark Seemann,mark@ploeh.dk, Wed Aug 31 18:21:21 2016 +0200,Src/AutoFakeItEasy2.UnitTest/Properties/AssemblyInfo.cs,2,2

Each line from csv file represents information about commit number, author, date and number of lines added and removed.

**2 Data analysis**

This chapter contains the data analysis itself.

## 2.1 Loading data and preparing environment

The csv file contains date-time variable which needs to be parsed. Before loading the csv file the locale is set to *us* and a custom data format is created to parse the date-time.

Sys.setlocale(locale='us')

setClass('git\_date')

setAs("character","git\_date", function(from) strptime(from, " %a %b %e %H:%M:%S %Y %z"))

Now we can load the dataset itself

dataset <- read.csv('commit\_log.csv', colClasses = c('factor', 'factor', 'factor', 'git\_date', 'factor', 'integer', 'integer'))

attach(dataset)

## 2.2 Descriptive statistics analysis

## 2.2.1 Data summary

To get a high overview of data summary command is run.

stats.summary <- summary(dataset)

The resulting summary is big so it was splatted in two parts shown which are shown in figure 2.1 and figure 2.2

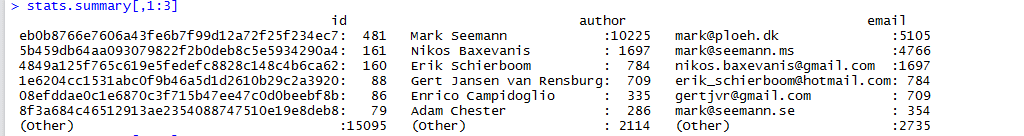


Figure 2.1 – First part of summary

Analyzing this figure 2.1 following conclusion affirmation can be made:

* The larges commit includes 481 files.
* User with name Mark Seemann modified 10225 files.
* User with email <mark@ploeh.gk> modified 5105 files which is much less than 10225. We can suppose that user Mark Seemann uses several emails to commit to github. It is possible because git users often works from different work stations which can be configured to use different user name and email.

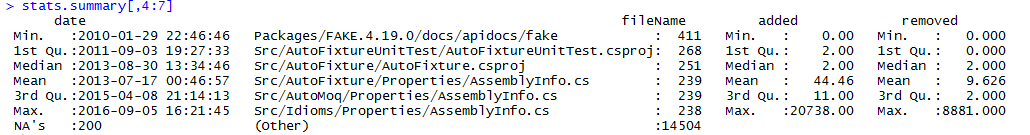


Figure 2.2 – Second part of summary

From second part of summary (Figure 2.2) following affirmation can be made:

* First commit was made on 2010-01-29
* The last commit was made on 2016-09-05
* 200 entries from csv file doesn’t contains commit date. It can be because of bug in R script or because the commit date is provided in wrong format. A quick manual analysis ( subset(dataset, is.na(date))) of csv data discovered that all unparsed dates has +1300 UTC offset, which is valid value.
* File Packages/FAKE.4.19.0/docs/apidocs/fake is included in 411 commits.
* The minimum number of added lines per file in commit is 0 which is obvious.
* The maximum number of added lines in one single commit for a file is 20738. Usually in git it means that a big file has been moved to another directory.
* 1st and 2nd quartile (median) are equals to 2. It means that in most cases only 2 new lines are added to a file. The mean is much bigger than 1st, median and 3rd quartiles – the distribution is ***positively skewed*.**
* The minimum number of removed lines per file in commit is 0 – obvious.
* The maximum number of removed lines in one single commit from a file is 8881.
* The distribution of removed lines if ***positively skewed*** as well, although it is not as much as number of added lines is (mean is closer to median).

Total number of authors is 61 and the total number of commits is 2985.

|  |
| --- |
| > length((unique(dataset$author)))  [1] 61  > length(unique(dataset$id))  [1] 2985 |

**2.2.2 Authors analysis**

To understand better the dataset several plots have been created using ggplot2 library.

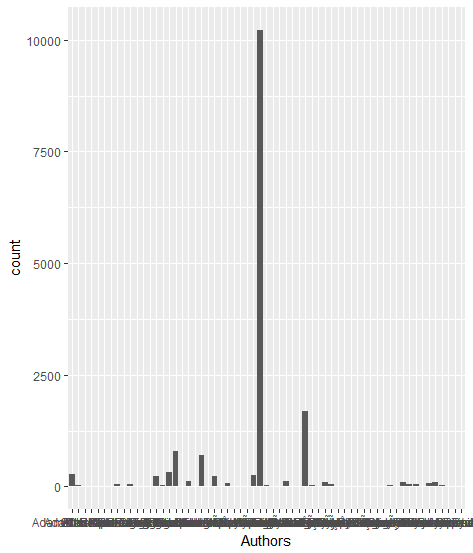


Figure 2.3 – All authors by number of changed files

Variable *author* is categorical variable. In Figure 2.3 is shown frequency distribution of authors. The X axis contains authors and Y axis contains the number of changed files. From figure we can see that there are several authors who made many changes. Source code for the plot is:

qplot(factor(author), data = dataset, geom = "bar", xlab = "Authors")

The X axis contains 61 authors so it’s difficult to see the authors name. Figure 2.4 contains same data on X and Y axis, but only top authors are shown.

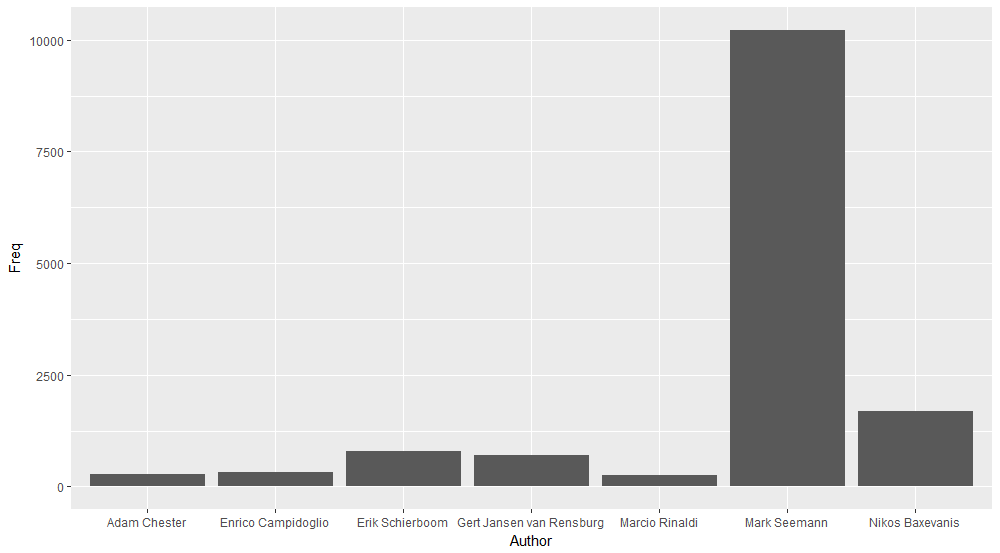


Figure 2.4 – Top contributors

To create plot following script is used:

stats.authors\_freq <- as.data.frame(table(dataset$author))

colnames(stats.authors\_freq) <- c("Author", "Freq")

stats.top\_authors <- subset(stats.authors\_freq, Freq >= mean(stats.authors\_freq$Freq))

ggplot(data = stats.top\_authors, aes(x=Author, y=Freq))+geom\_histogram(stat = "identity")

The script filters authors who changed more than mean (264) files. There are only 7 contributors who changed more than 264 files.

**2.2.3 Added and removed lines analysis**

Another two variables which presents interest are number of added and removed lines.

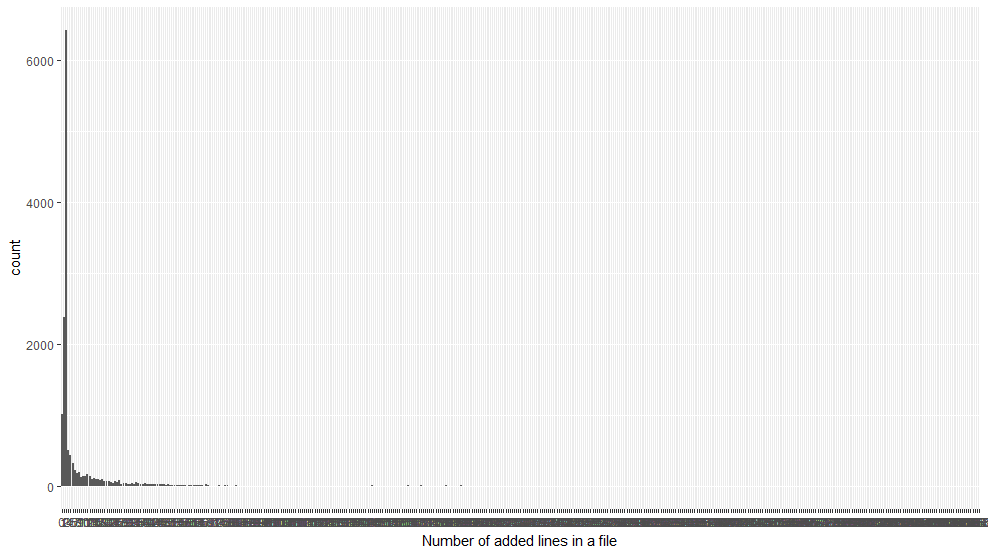


Figure 2.5 – Frequency distribution of Number of added lines

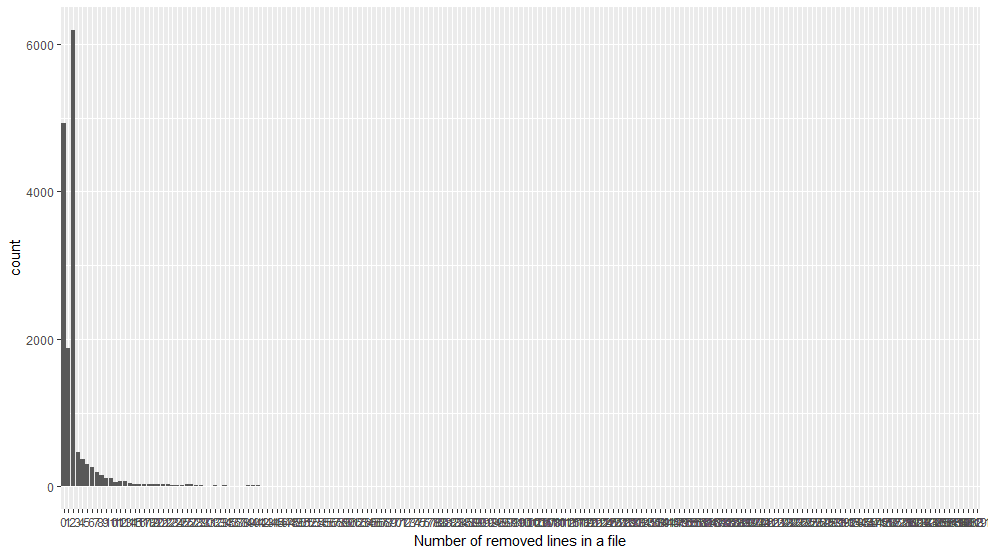


Figure 2.6 – Frequency distribution of Number of removed lines

Figures 2.5 and 2.6 confirms our assumption that the plots are positively skewed. Following scripts are used to create plots:

qplot(factor(added), data = dataset, geom = "bar", xlab = "Number of added lines in a file")

qplot(factor(removed), data = dataset, geom = "bar", xlab = "Number of removed lines in a file")

As in previous example with authors it can be useful to show plots with most frequent values. Figure 2.7 and 2.8 contains this plots. The source code of scripts is:

stats.authors\_freq <- as.data.frame(table(dataset$author))

colnames(stats.authors\_freq) <- c("Author", "Freq")

stats.top\_authors <- subset(stats.authors\_freq, Freq >= mean(stats.authors\_freq$Freq))

ggplot(data = stats.top\_authors, aes(x=Author, y=Freq)) + geom\_histogram(stat = "identity")

stats.added\_freq <- as.data.frame(table(dataset$added))

colnames(stats.added\_freq) <- c("Added", "Freq")

stats.top\_adds <- subset(stats.added\_freq, Freq > mean(dataset$added))

ggplot(data = stats.top\_adds, aes(x=Added, y=Freq)) + geom\_histogram(stat = "identity")

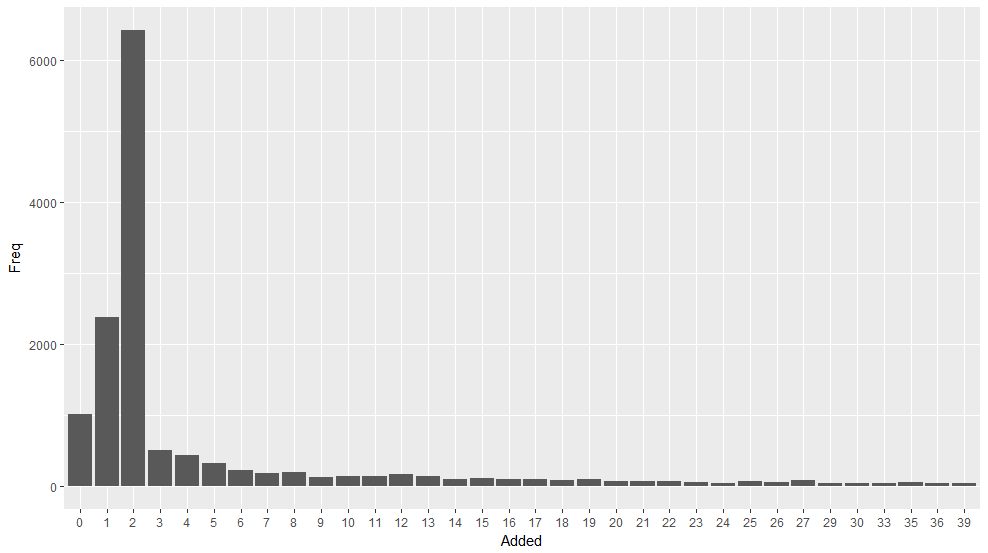


Figure 2.7 – Top frequency distribution of Number of added lines

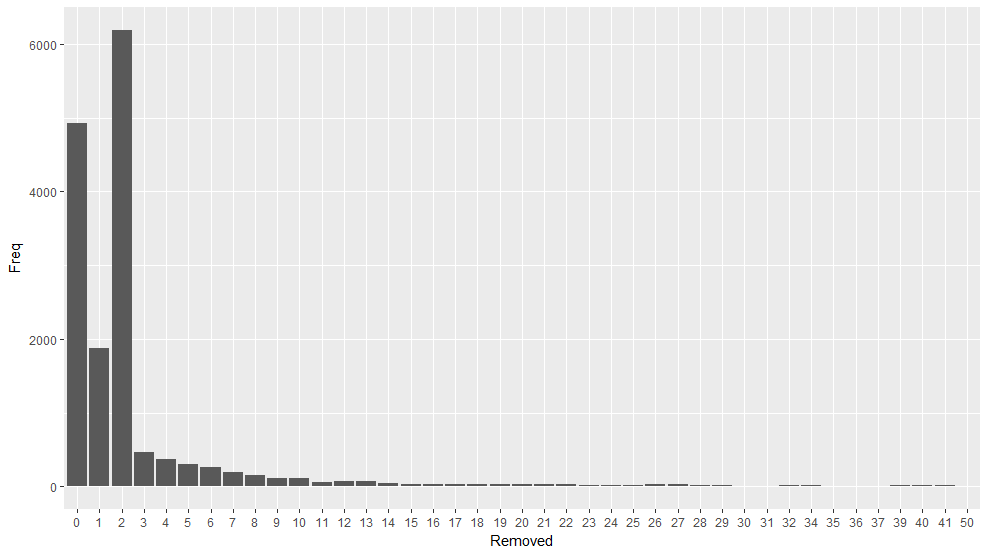


Figure 2.8 – Top frequency distribution of Number of removed lines

**2.2.4 Multivariate data analysis**

In this multivariate analysis is done to find relationships between variables.

Authors variable is a categorical variable, while commit date is a continues variable. The relation between two variables denotes contributor’s activity during repository life time. Figure 2.9 represents the contributor’s activity to the repository. The script is:

qplot(author, date, data=dataset, geom="boxplot", fill=author) + theme(axis.text.x = element\_text(angle = 90, hjust = 1))

Figure contains interesting information:

* There are 3 main active contributors – Enrico Campidoglio, Mark Seemann and Nikos Baxevanis.
* Mark Seemann – contributed since repository was created (he’s the repository owner). He still is contributing (because of max value). His contribution activity during 6 years was constant because median is placed in the middle between 1st and 3rd quartiles.
* Enrico Campidoglio – his activity is nearly the same as for Mark. The only difference is that he started contributing in middle of 2011 (the min value). One more small difference is that the median is little bit closer to 3rd quartile which means he’s more active now rather he was at the beginning.
* Nikos Baxevanis – started contributing in the same time as Enrico Campidoglio. He contributed actively very short time because median is very close to min value and 3rd quartile is close to min value as well. He contributed recently (max value) and it keeps him in the list of active contributors.
* The other contributors contributed only once or very short period

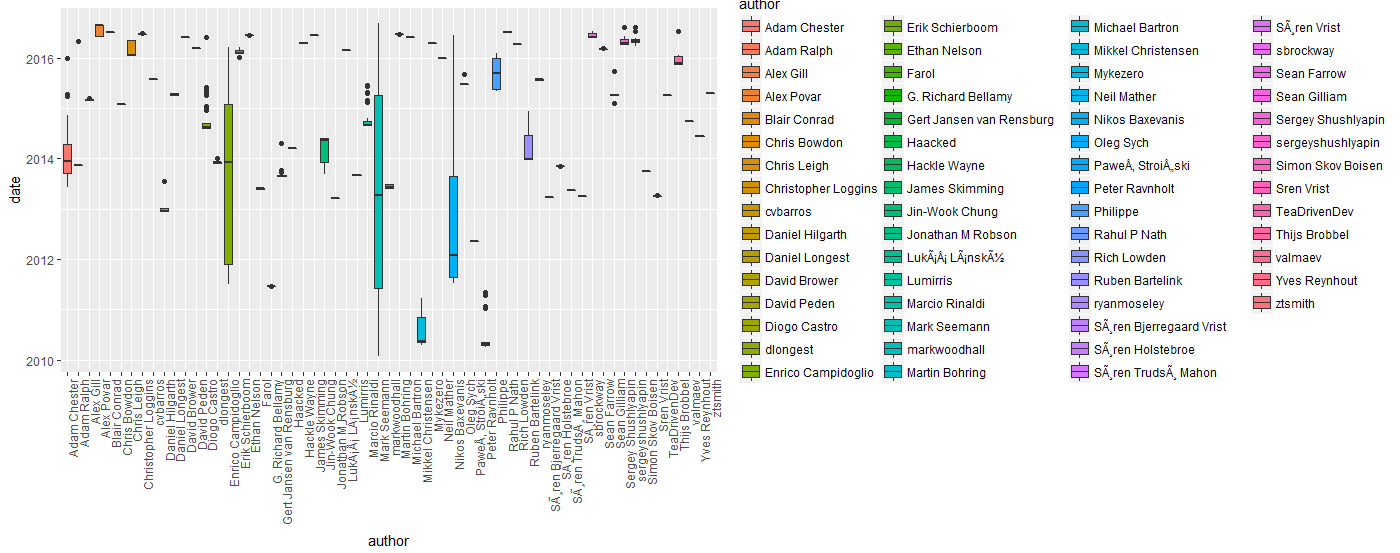


Figure 2.9 – Contributors activity during repository life