2IMP25 - Software Evolution Assignment 3

D. J. van den Brand 0772180
J.P.A. Biesbroeck 0835900

March 30, 2017

1 Introduction

Software projects are always prone to change. These changes can often be implemented wrongly. As a consequence, even more changes are required to improve the product. So, a good insight in how these changes propagate through a project is vital for effective software development. Especially for larger projects the management of these changes can become quite the hassle to do manually. This paper will focus on the time people spend on a change and if this can be related to the data that open source projects gather during their development.

2 Methodology

We make use of the standardized model of [4] to model these changes. There are a few large open source projects that cohere to this model, so this makes it easier to analyze and compare with different studies. They all make use of the code review process of Gerrit [1]. This system denotes CHANGES as pieces of code that have to be reviewed. Next to that REVISIONS are used to make additional changes to a CHANGE before it is reviewed. And all messages and discussions are stored as HISTORY for each CHANGE as well.

We started by importing all the data into an SQL database. Then using multiple queries we managed to export the data to csv files. To ensure there is not much redundant information in the csv files we only include a single grouped join per file.

- change_with_last_revision: CHANGES matched to its last REVISION
- change_with_files: CHANGES matched to FILES
- change_without_files: List of CHANGES that are not linked to FILES
- people_with_changes: CHANGES matched to PEOPLE
- people_with_history_review: CHANGES matched to review comments in HISTORY
- people_with_history_review+-2: CHANGES matched to review comments in HISTORY with a grade of -2 or +2

After the relevant data has been exported to csv files, we make use of R to analyze the data. Using the merge function we can join several tables and with the lm function we performed multiple linear regression. Afterwards outliers are removed using a script [2] that makes use of the Tukey's method to identify the outliers in the range above and below 1.5 times IQR. Next, we look at the VIF value of the independent variables and exclude the variables with a VIF higher than 3 to avoid multicollinearity between the independent variables.

[1] https://www.gerritcodereview.com/
[2] https://www.r-bloggers.com/identify-describe-plot-and-remove-the-outliers-from-the-dataset/

3 Case study

We have chosen to analyze the projects OpenStack and LibreOffice. Both of these projects make use of a new database. AOSP, for example, does not make use of this new database and Qt is an incomplete dataset. OpenStack is an open source cloud operating system that controls resources throughout a datacenter. LibreOffice is an open source office suite. It comprises programs for word processing, editing spreadsheets, working with databases and composing mathematical formulas.

4 Results

In order to detect and remove outliers we have made use of the Tukey's method to identify the outliers ranged above and below 1.5 times IQR. We made use of a script that uses this method to remove outliers [3]. The script shows a boxplot and histogram to represent the changes before and after removing outliers. Outliers have been removed for all the listed independent variables listed in the assignment description except for the variables change activity in LibreOffice and whether at least one file modified represents source code in LibreOffice and OpenStack. Figure 1 shows these visualizations for the variable ecosystem tenure from LibreOffice. After all the outliers have been removed we looked at the VIF of the variables and two variables had a value higher than 3. For LibreOffice the variable with the largest VIF, change activity, has been removed and afterwards the VIF of all the variables was below 3 (see Figure 9). For OpenStack the variable with the largest VIF, review_activity_2, has been removed and afterwards the VIF of all the variables was below 3 (see Figure 17). Figure 2 and 10 show the summary output R provides for both projects. Figure 3 and 11 show the residual vs fitted plots and Figure 4 and 12 show the QQ plots. We did not apply the logarithmic transformation. Figure 5 and 13 show the anova tables.

Figure 6 and 14 show the amount of revisions created per month. Figure 7 and 15 show the amount of active revisions per month. Figure 8 and 16 show boxplots of the number of reviews created per day of the week over the entire dataset.

5 Discussion

These results do not fit very well. We did expect this beforehand. A good statistical fit to a certain project is hard to find. It would be nice to find a clear linear connection between the review time and one of these simple metrics, but this is unlikely. The most promising candidate is the revision count. This denotes the number of revisions that were performed on a given change. It is likely that if there are more revisions there is more discussion and communication needed before a change can be closed. Because this is an open source project, not everyone is working full time. So it is likely that developers are waiting on each other. However this does not become apparent from the data.

The revisions created per weekday however does show that most of the changes are created during weekdays for LibreOffice. In OpenStack a lot less reviews are created so this becomes harder to see.

Like it can be seen in Figure 7 there are a few moments where the number of revisions that are opened spiked. Especially these are clear on Jan 2014, Jul 2014, Aug 2015 and Jan 2016. In these months many pending changes were open. This could be due to a new release. All the issues have to be fixed before a release can be published, so it is likely that more changes will be made towards an upcoming release.

6 Threats to validity

To specify if a file contains documentation or not is very crude. Documentation could also be provided in files where source code is written. This is not taken into account. Also, the classification of files into source code and documentation was simplified. We mostly used a filename extension approach to classify. However, the purpose of a certain extension for a file is not always the same. For example, many files that contain .html code could either be generated by the program, be used as a component of the program or could be used as documentation of the project. Specifying it as documentation is therefore quite naive. A better approach would have been to also look at the folder structure of the files. This would have required an in depth understanding of the project itself.

[3] https://www.r-bloggers.com/identify-describe-plot-and-remove-the-outliers-from-the-dataset/

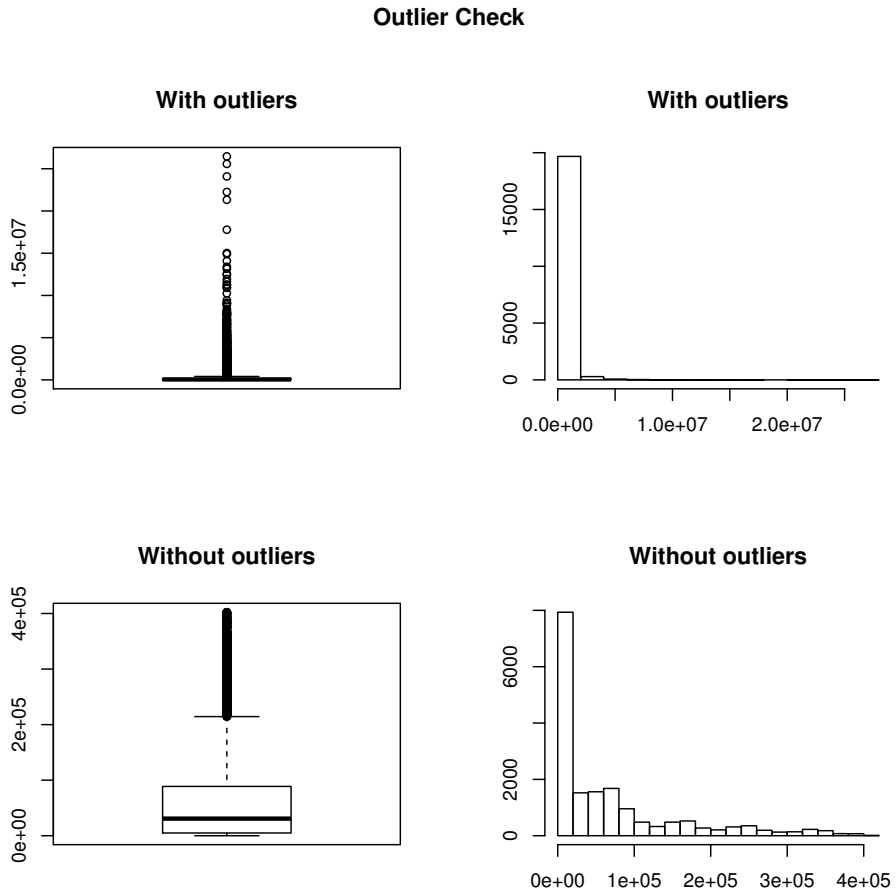

Figure 1: Ecosystem tenure outliers check (LibreOffice)

The data might not be completely valid. We found several cases that were reviewed before they were opened, without any clear cause. This difference was sometimes just a few milliseconds. We rounded these negative times to zero. However, also a fixed difference of an hour was seen multiple times. This could have been caused by a wrong implementation of the time, because this project has been developed in different timezones. This raised some concerns about the reliability of the time measurements.

7 Related work

The statistical results that were found are hard to relate to other work because the support is so low. However we added some additional analyses that were also present in other work. Like in [3] it can be seen that the number of reviews for LibreOffice, as seen in Figure 7, also fluctuates a lot per month. For OpenStack, as seen in Figure 15, the total period is not long enough to draw this conclusion.

A comparison with [1] could be made. However a more sophisticated classification is needed. A classification algorithm should be used to classify a description field of a change (in combination with the corresponding comments) to similar categories. However, this article did leave out the evolvability aspect of a software project. It only drew conclusions about the state of the project as a whole. However, the project was developed over time, so the experience and activity of the developers (personally and in the project itself) will change.

Like in [2] also a graph of the reviews per day has been created. Also here can be seen that the contributions to LibreOffice are mainly in the weekdays as opposite to the weekends.

```
Residuals:
    Min      1Q  Median      3Q     Max
-287994  -58626  -35363   21260  342739

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                 4.202e+04  1.034e+04   4.065 4.84e-05 ***
m4$p_ecosystem_tenure       3.516e-05  6.484e-05   0.542 0.587624
m4$p_review_tenure         -1.159e-04  5.606e-05  -2.067 0.038752 *
m4$ch_revision_count        1.743e+04  8.366e+02  20.834  < 2e-16 ***
m4$p_review_tenure_2        3.128e-05  1.070e-04   0.292 0.770167
m4$p_review_activity_2     -5.379e+00  1.471e+00  -3.657 0.000257 ***
m4$ch_total_lines_inserted  3.852e+01  3.663e+01   1.052 0.293042
m4$ch_total_lines_deleted  -3.360e+01  6.661e+01  -0.504 0.613944
m4$ch_files_changed        -3.902e+02  4.937e+02  -0.790 0.429400
m4$ch_contains_source      -1.048e+03  3.503e+03  -0.299 0.764781
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 87460 on 9619 degrees of freedom
  (10462 observations deleted due to missingness)
Multiple R-squared:  0.04958,   Adjusted R-squared:  0.04869
F-statistic: 55.76 on 9 and 9619 DF,  p-value: < 2.2e-16
```

Figure 2: Summary (LibreOffice)

8 Conclusions

The repository of two open source projects, LibreOffice and OpenStack, have been analysed. SQL and R have been used to extract statistical data from a standardized model of these repositories. A linear regression is applied to the review time versus multiple variables. However, few statistical relevance is found for the linear regression on both projects. Next to that additional metrics on the behavior of the reviews is analysed and compared with related work.

References

[1] Moritz Beller, Alberto Bacchelli, Andy Zaidman, and Elmar Juergens. Modern code reviews in open-source projects: Which problems do they fix? In *Proceedings of the 11th Working Conference on Mining Software Repositories*, MSR 2014, pages 202–211, New York, NY, USA, 2014. ACM.

[2] Murtuza Mukadam, Christian Bird, and Peter C. Rigby. Gerrit software code review data from android. pages 45–48. IEEE, 2013.

[3] Peter C. Rigby, Daniel M. German, and Margaret-Anne Storey. Open source software peer review practices: A case study of the apache server. In *Proceedings of the 30th International Conference on Software Engineering*, ICSE '08, pages 541–550, New York, NY, USA, 2008. ACM.

[4] Xin Yang, Raula Gaikovina Kula, Norihiro Yoshida, and Hajimu Iida. Mining the modern code review repositories: A dataset of people, process and product. In *Proceedings of the 13th International Conference on Mining Software Repositories*, pages 460–463, 2016.

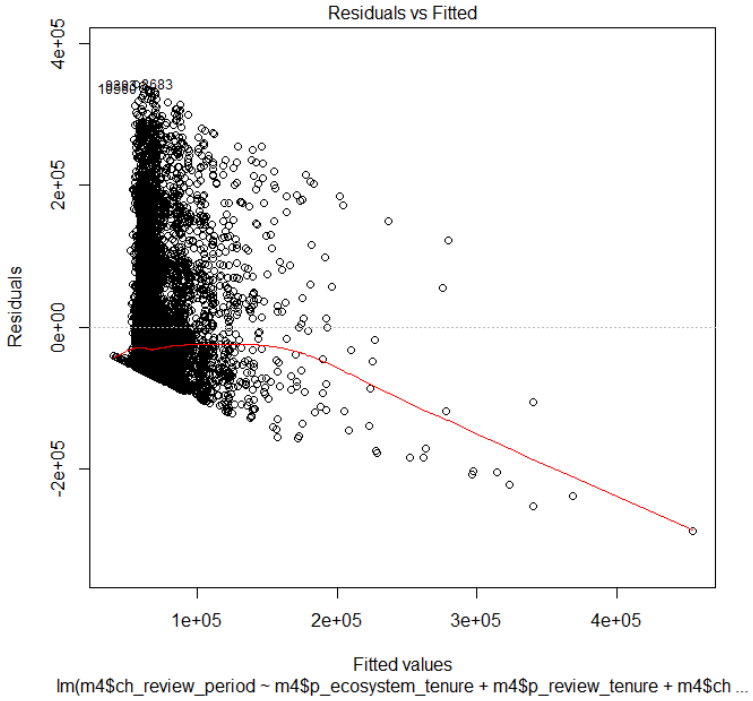

Figure 3: Residuals vs fitted (LibreOffice)

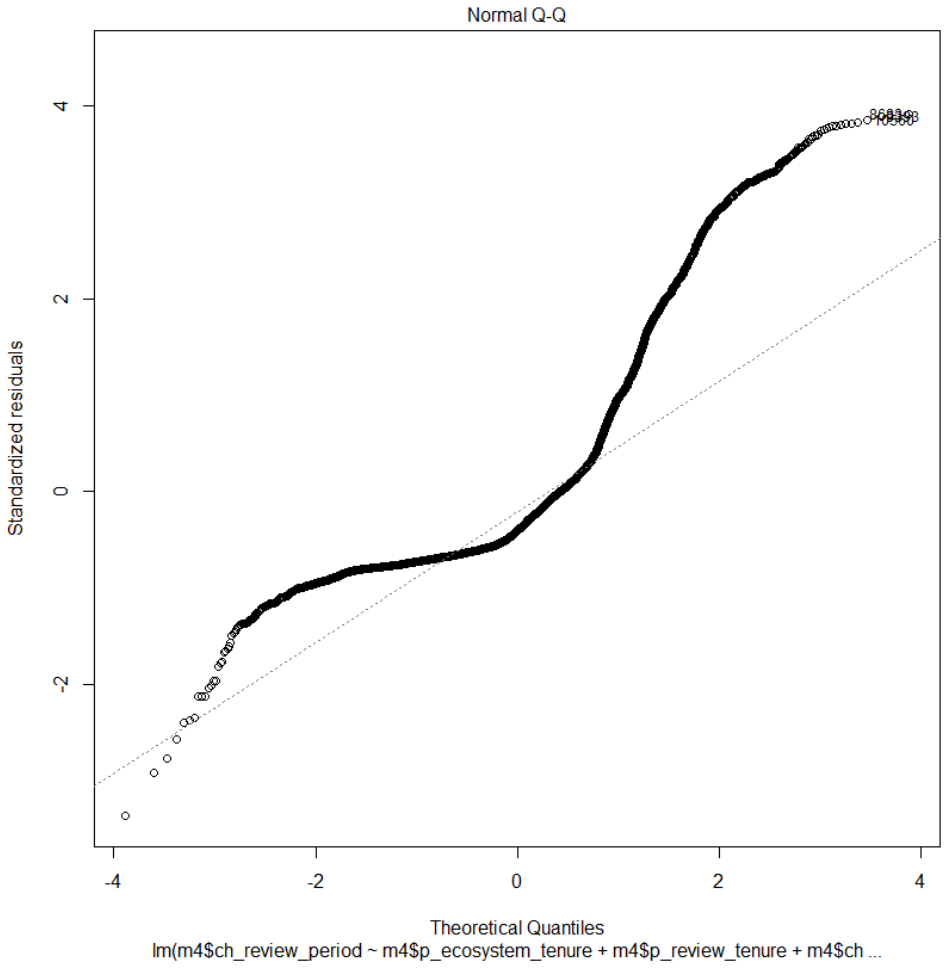

Figure 4: QQ plot (LibreOffice)

```
Response: m4$ch_review_period
                          Df     Sum Sq    Mean Sq  F value    Pr(>F)
m4$p_ecosystem_tenure      1 8.0486e+10 8.0486e+10  10.5226 0.0011833 **
m4$p_review_tenure         1 2.9787e+11 2.9787e+11  38.9429 4.548e-10 ***
m4$ch_revision_count       1 3.3461e+12 3.3461e+12 437.4593 < 2.2e-16 ***
m4$p_review_tenure_2       1 3.7951e+08 3.7951e+08   0.0496 0.8237369
m4$p_review_activity_2     1 1.0086e+11 1.0086e+11  13.1864 0.0002835 ***
m4$ch_total_lines_inserted 1 2.7252e+09 2.7252e+09   0.3563 0.5505860
m4$ch_total_lines_deleted  1 4.3127e+09 4.3127e+09   0.5638 0.4527358
m4$ch_files_changed        1 4.9641e+09 4.9641e+09   0.6490 0.4204922
m4$ch_contains_source      1 6.8481e+08 6.8481e+08   0.0895 0.7647811
Residuals               9619 7.3574e+13 7.6489e+09
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 5: Anova table (LibreOffice)

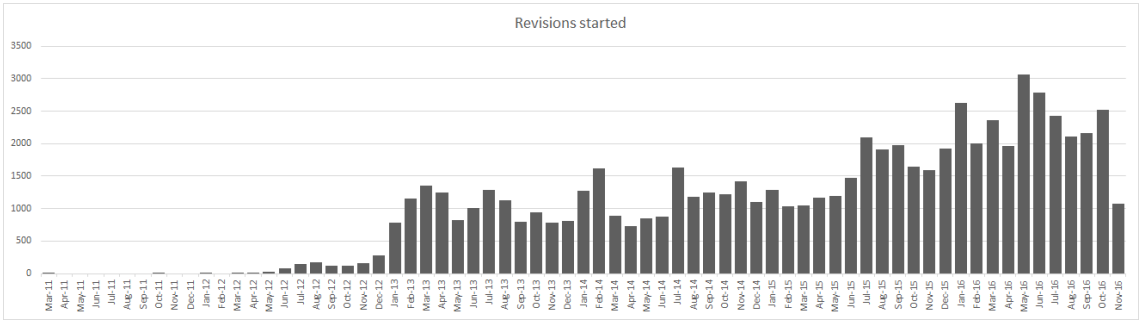

Figure 6: Revision created dates (LibreOffice)

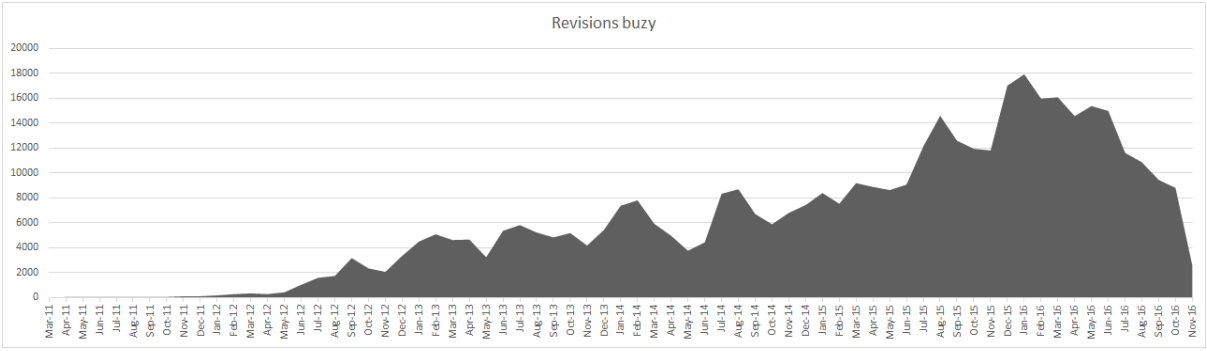

Figure 7: Revision created heatmap (LibreOffice)

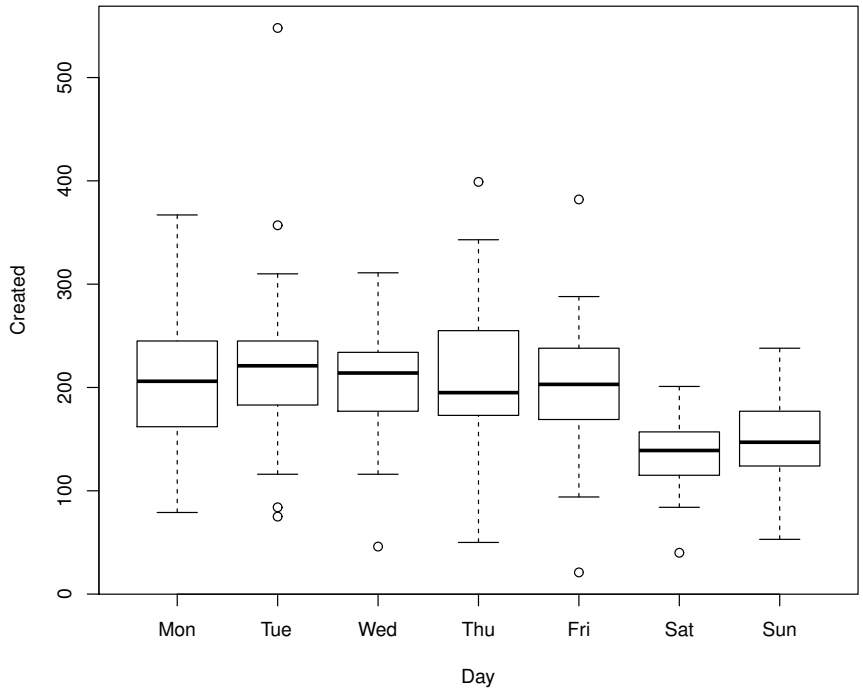

Figure 8: Boxplot revision created per weekday (LibreOffice)

```
 m4$p_ecosystem_tenure         m4$p_review_tenure       m4$ch_revision_count       m4$p_review_tenure_2
              1.481985                   2.902483                   1.017303                   2.348778
m4$p_review_activity_2 m4$ch_total_lines_inserted  m4$ch_total_lines_deleted       m4$ch_files_changed
              1.209499                   1.280914                   1.229085                   1.285461
 m4$ch_contains_source
              1.019462
```

Figure 9: VIF values (LibreOffice)

```
 Residuals:
     Min       1Q   Median       3Q      Max
-3256833  -777883  -393179   -53307 61708274

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                -9.234e+05  7.789e+04 -11.855  < 2e-16 ***
l4$p_ecosystem_tenure       1.818e-03  5.281e-04   3.442 0.000578 ***
l4$p_change_activity       -5.163e+02  4.700e+01 -10.983  < 2e-16 ***
l4$p_review_tenure          6.507e-03  1.244e-03   5.232 1.68e-07 ***
l4$ch_revision_count        3.693e+05  5.775e+03  63.958  < 2e-16 ***
l4$p_review_tenure_2       -2.841e-04  7.137e-04  -0.398 0.690547
l4$ch_total_lines_inserted  4.645e+02  1.557e+02   2.984 0.002847 **
l4$ch_total_lines_deleted  -2.526e+03  1.048e+03  -2.409 0.015983 *
l4$ch_files_changed         2.145e+03  5.760e+03   0.372 0.709599
l4$ch_contains_source      -2.092e+04  2.722e+04  -0.769 0.442110
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2515000 on 52038 degrees of freedom
  (41698 observations deleted due to missingness)
Multiple R-squared:  0.07724,   Adjusted R-squared:  0.07708
F-statistic:   484 on 9 and 52038 DF,  p-value: < 2.2e-16
```

Figure 10: Summary (OpenStack)

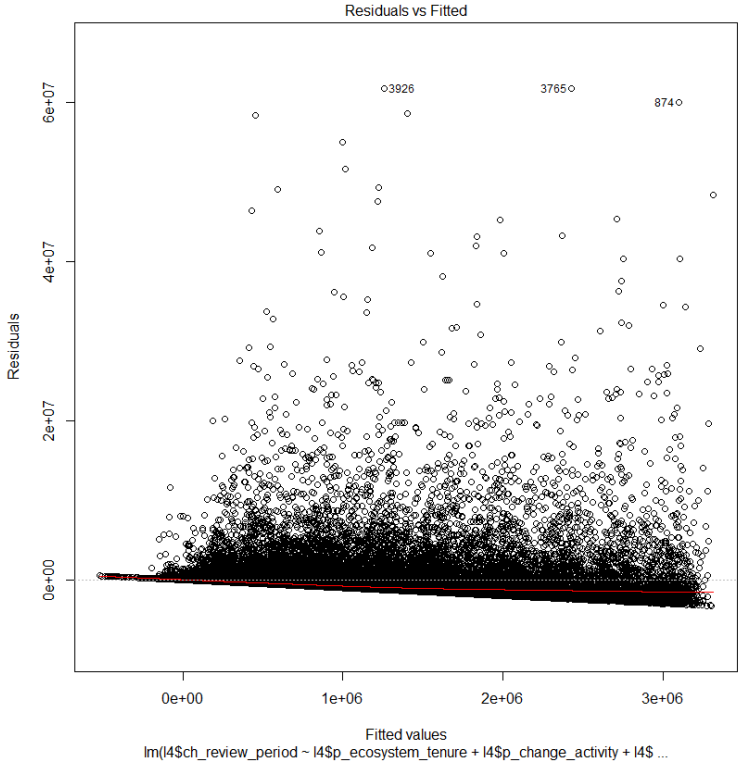

Figure 11: Residuals vs fitted (OpenStack)

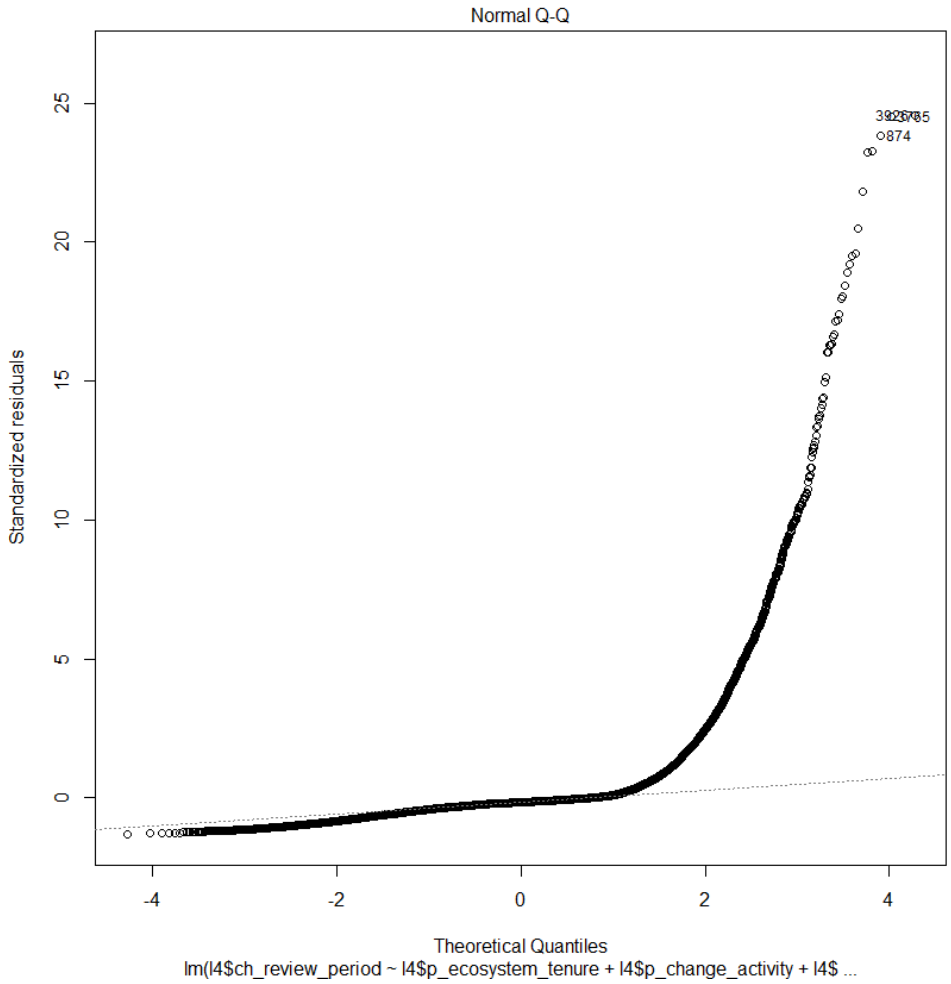

Figure 12: QQ plot (OpenStack)

```
Response: l4$ch_review_period
                            Df     Sum Sq    Mean Sq   F value    Pr(>F)
l4$p_ecosystem_tenure        1 5.8532e+13 5.8532e+13    9.2557  0.002349 **
l4$p_change_activity         1 1.2762e+15 1.2762e+15  201.8079 < 2.2e-16 ***
l4$p_review_tenure           1 2.1262e+14 2.1262e+14   33.6213 6.734e-09 ***
l4$ch_revision_count         1 2.5886e+16 2.5886e+16 4093.3781 < 2.2e-16 ***
l4$p_review_tenure_2         1 1.0679e+12 1.0679e+12    0.1689  0.681115
l4$ch_total_lines_inserted   1 6.5701e+13 6.5701e+13   10.3894  0.001268 **
l4$ch_total_lines_deleted    1 4.1277e+13 4.1277e+13    6.5272  0.010626 *
l4$ch_files_changed          1 2.3937e+11 2.3937e+11    0.0379  0.845743
l4$ch_contains_source        1 3.7362e+12 3.7362e+12    0.5908  0.442110
Residuals                52038 3.2908e+17 6.3239e+12
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 13: Anova table (OpenStack)

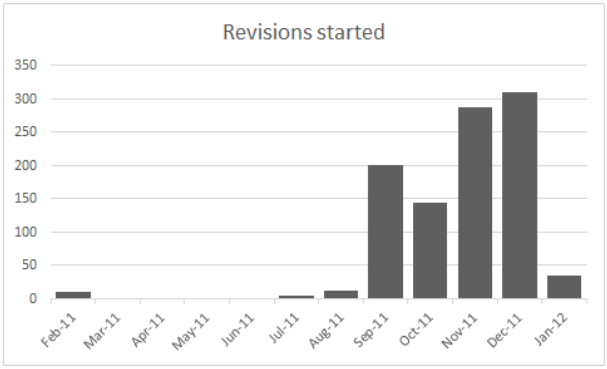

Figure 14: Revision created dates (OpenStack)

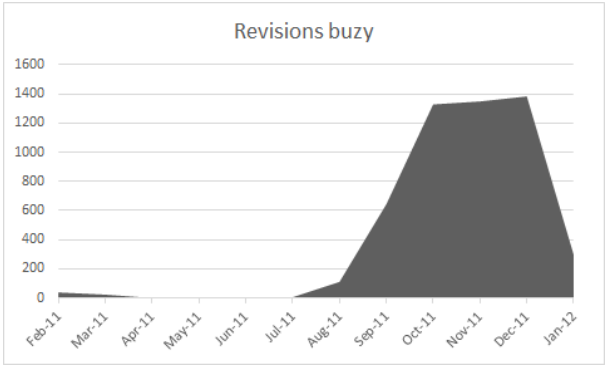

Figure 15: Revision created heatmap (OpenStack)

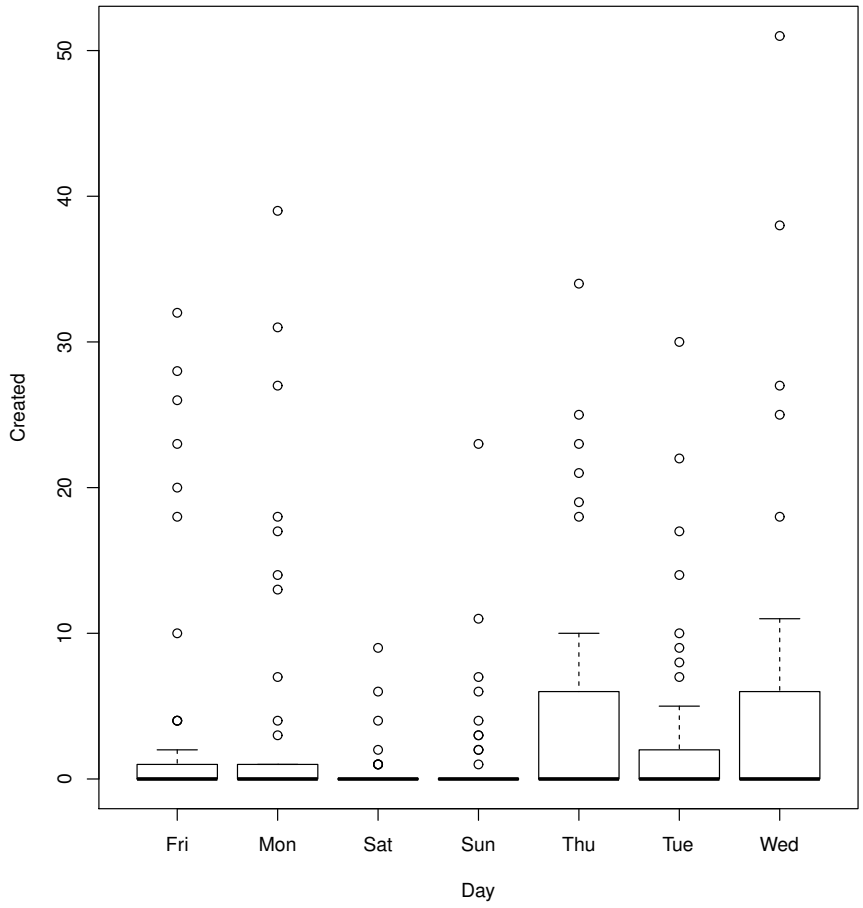

Figure 16: Boxplot revision created per weekday (OpenStack)

```
l4$p_ecosystem_tenure       l4$p_change_activity        l4$p_review_tenure       l4$ch_revision_count
             2.234074                   1.192327                  2.211607                   1.004091
 l4$p_review_tenure_2 l4$ch_total_lines_inserted  l4$ch_total_lines_deleted        l4$ch_files_changed
             2.060290                   1.332598                  1.144289                   1.572276
l4$ch_contains_source
             1.108157
```

Figure 17: VIF values (OpenStack)