Table of Contents

[I Introduction 3](#_Toc24103856)

[II Research Questions 3](#_Toc24103857)

[III Definition of Main Variables & Visual Exploration 4](#_Toc24103858)

[3.1 Definition of Main Variables 4](#_Toc24103859)

[3.2 Visual Exploration 5](#_Toc24103860)

[IV Data Cleaning & Preparation 7](#_Toc24103861)

[4.1 Importance Analysis 7](#_Toc24103862)

[4.2 Data Cleaning 8](#_Toc24103863)

[V Multi-level Longitudinal Model 9](#_Toc24103864)

[5.1 Model A 9](#_Toc24103865)

[5.2 Model B 10](#_Toc24103866)

[5.3 Model C 12](#_Toc24103867)

[5.3.1 Model C1 12](#_Toc24103868)

[5.3.2 Model C2 13](#_Toc24103869)

[5.3.3 Model C3 14](#_Toc24103870)

[5.3.4 Model C4 15](#_Toc24103871)

[5.4 Model D 16](#_Toc24103872)

[5.4.1 Model D1 16](#_Toc24103873)

[5.4.2 Model D2 17](#_Toc24103874)

[5.4.3 Model D3 18](#_Toc24103875)

[5.4.4 Model D4 19](#_Toc24103876)

[5.4.5 Model D5 20](#_Toc24103877)

[5.4.6 Model D6 21](#_Toc24103878)

[5.5 Model E 22](#_Toc24103879)

[5.5.1 Model E1 22](#_Toc24103880)

[5.5.2 Model E2 23](#_Toc24103881)

[VI Results & Discussions 24](#_Toc24103882)

[6.1 Evaluations 24](#_Toc24103883)

[6.2 Limitations & Future Work 24](#_Toc24103884)

[VII Conclusion 25](#_Toc24103885)

[References 25](#_Toc24103886)

# I Introduction

GitHub is a web-based Git that provides version-control storage and hosting (Dabbish et al., 2012). According to the annual report released by GitHub in 2015, the GitHub community has reached 24 million developers, 1.5 million organizations including global technology giants such as Microsoft, Facebook, Google and Apple, and more than 67 million resource database information (Blischak et al., 2016). Since September 2016, the number of submissions has reached 1 billion, and the number is proliferation (Blischak et al., 2016). Projects hosted on GitHub can be accessed and operated with standard Git commands, and GitHub offers a range of social networking features, such as following and commenting. Up to now, GitHub has more than 12 million open source projects, and the number is still growing. The analysis of influencing factors and correlation in the development process can reveal the development level of GitHub open source software and the progress of the project to some extent (Kalliamvakou et al., 2015). By analyzing the data generated by open-source software during development, we proposed eight significant factors affecting development quality: Time, Forks, Members, Commits, Issues, Watchers, PullReq, CommitCmnt, and analyzed the correlation among these factors. Section 1 gives a brief introduction to GitHub, focusing on the factors that influence the development process of GitHub open-source software. Section 2 proposes the Research Question. Section 3 defines the main variables and explores the data visually. Section 4 completes the cleaning and preparation of the data. Section 5 constructed the Longitudinal Multi-level Model and analyzed the results in detail. Section 6 explains the superior results and offers a few suggestions for developers. Finally, the paper summarizes and looks forward to the follow-up work.

# II Research Questions

Many researchers have paid attention to the influential factors in the development process of open source software services (Perez-Riverol et al., 2016), but the correlation analysis of these influential factors has not received enough attention. Correlation analysis of influencing factors can help contributors of open-source software better participate in the development and maintenance of software, and also improve the efficiency and quality of open-source software development (Tsay et al., 2014). For example, whether it is possible to accelerate the problem-solving speed in the development process of open source software by adjusting one or several influential factors in the development process of open-source software. Such questions have become the primary motivation of this paper. What influences can we adjust to make the development of open-source software faster and better? How to adjust? Given this, this paper proposes the following research questions: (1) Whether there is a correlation between the significant factors that affect the development process of GitHub open-source software, and if so, what is the correlation? (2) How to make use of the correlation of influential factors in the development process of GitHub open-source software to guide the development of open-source software better?

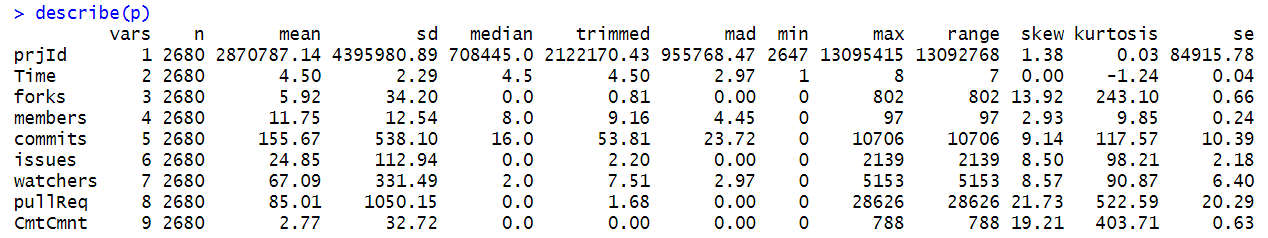
# III Definition of Main Variables & Visual Exploration

## 3.1 Definition of Main Variables

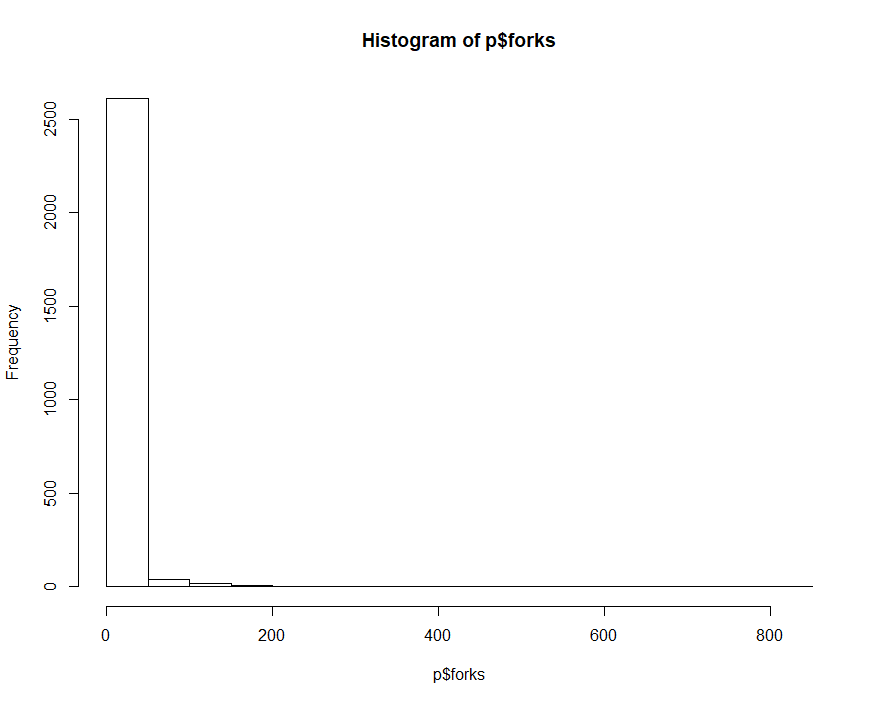
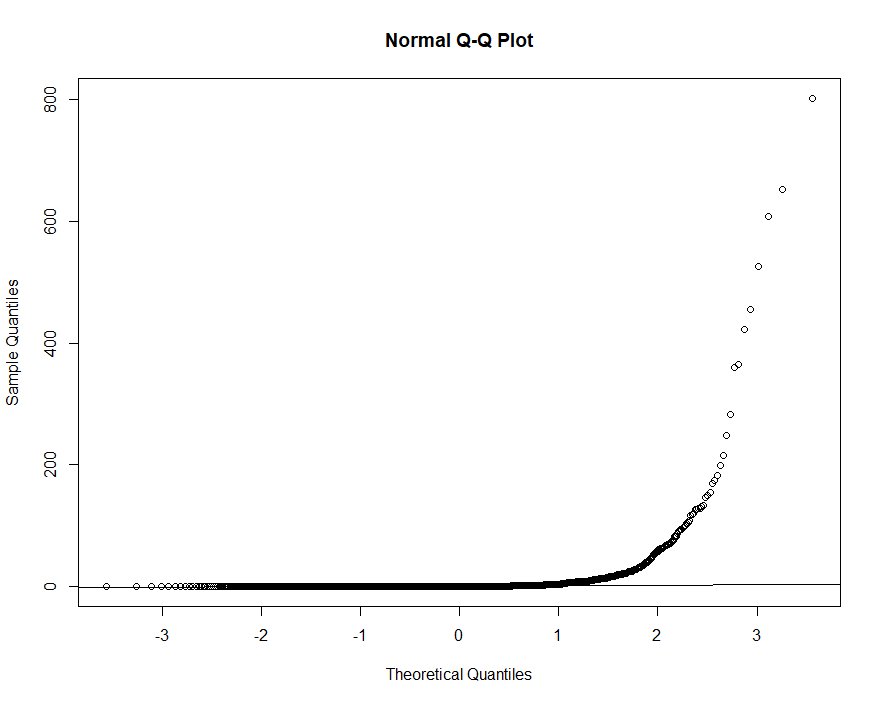
Table 1 Definition of Main Variables

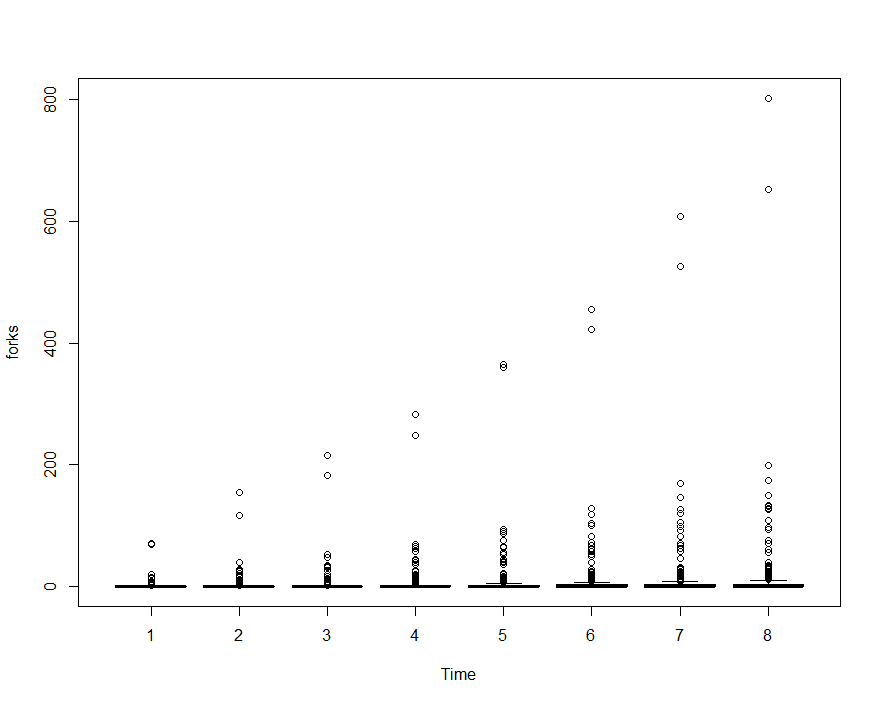
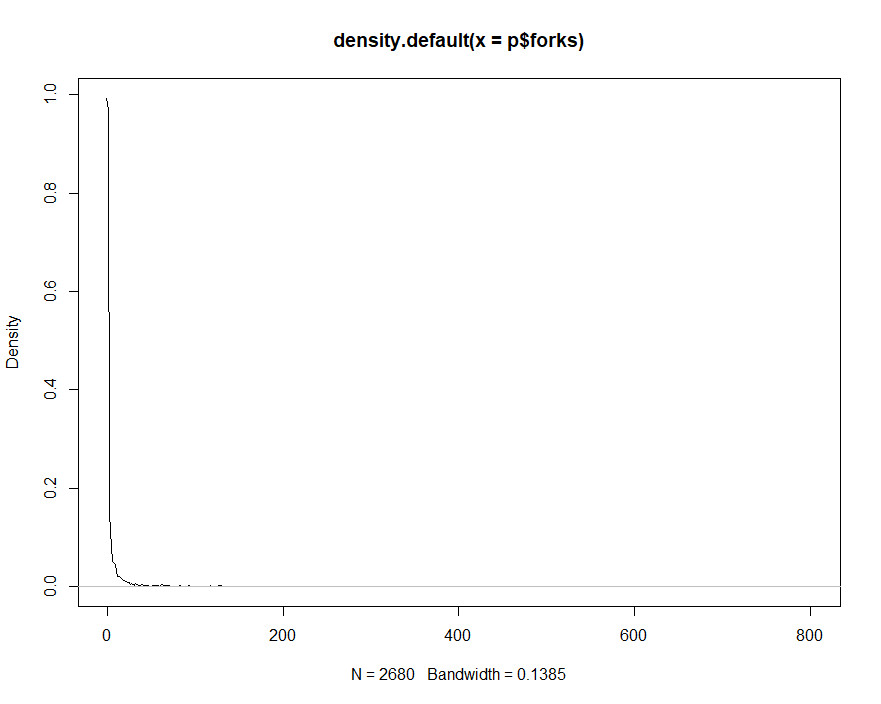
|  |  |
| --- | --- |
| **Variable** | **Definition** |
| PrjID | A unique id number for each project |
| Period | Represents the current record contains data for which period of year |
| Time | A sequence for time of observations |
| SatrtDate | Beginning of observation |
| EndDate | End of observation |
| Forks | Number of times a project is Forksed |
| Members | Number of members |
| Commits | Number of coding activities |
| Issues | Number of problem/bugs raised or requests for new features |
| Watchers | Number of people interested in project |
| PullReq | Number of code changes request for review. |
| CommitCmnt | Number of discussion on commits |

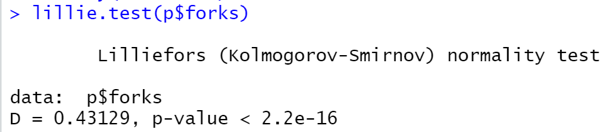
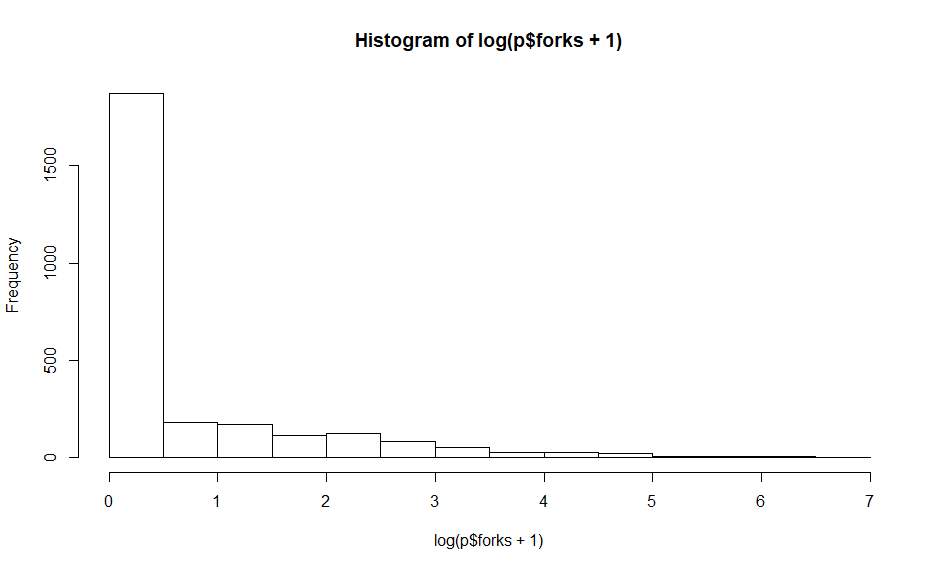
Since there are inclusion and inclusion relationships in all 31 variables, such as PullReqCmnt is a subset of CommitCmnt, MemIssue is a subset of Issues, so we only focus on the top 8 variables in this report: Time, Forks, Members, Commits, Issues, Watchers, PullReq and CommitCmnt.



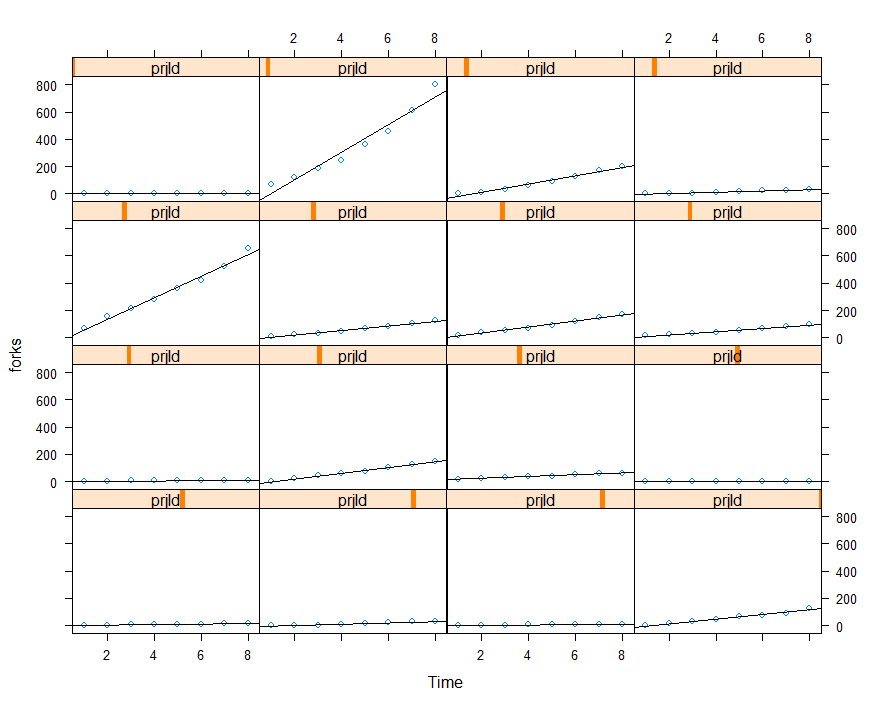
## 3.2 Visual Exploration

****

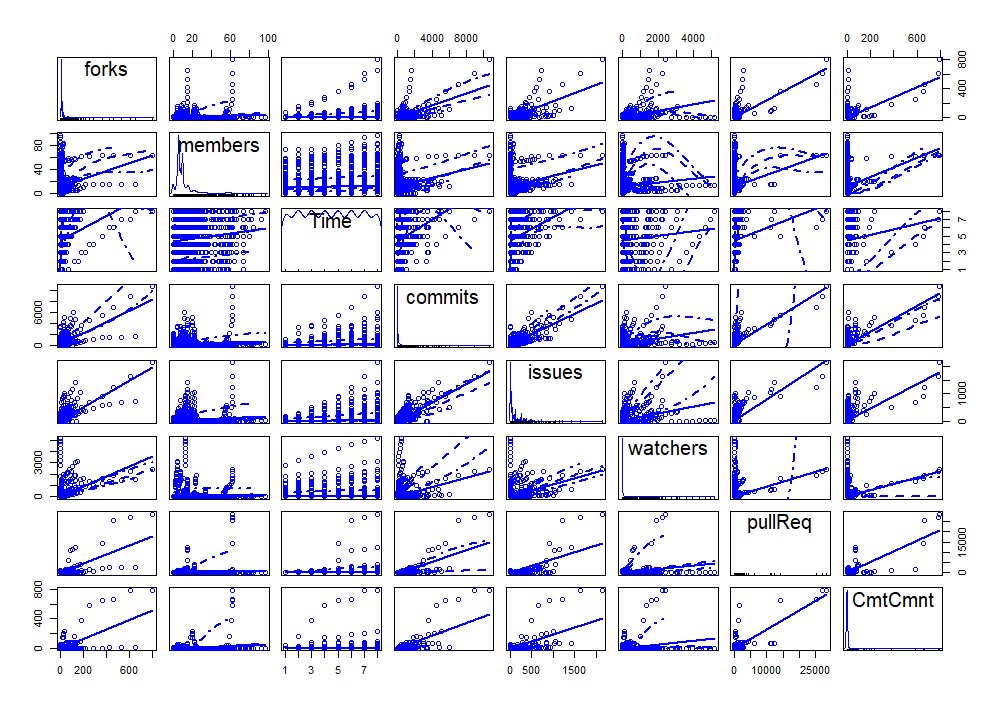
****

****

First, we use the graph method to test the normal distribution of the dependent variable Forks. By observing the graph, we guess that Forks are not normally distributed. Subsequently, KS Test was carried out on Forks using statistics. The calculated results showed that, p < 0.05, reject H0, so Forks did not meet the normal distribution.

****

The purpose of this paper is to explore the influence of different factors on Forks in the time dimension, so we selected 16 projects (5% of the total number of projects) to observe the relationship between Forks and Time, and found that Forks and Time are linearly related.

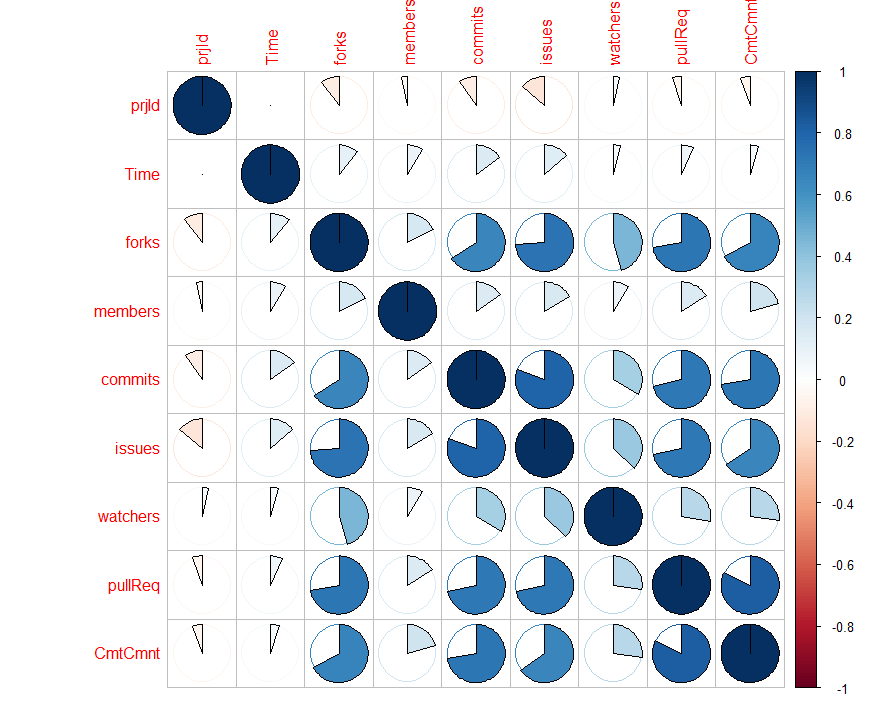


We have conducted Linearity Check for eight main variables. The result shows that there is a linear relationship between 8 main variables. The Linearity assumption is satisfied.

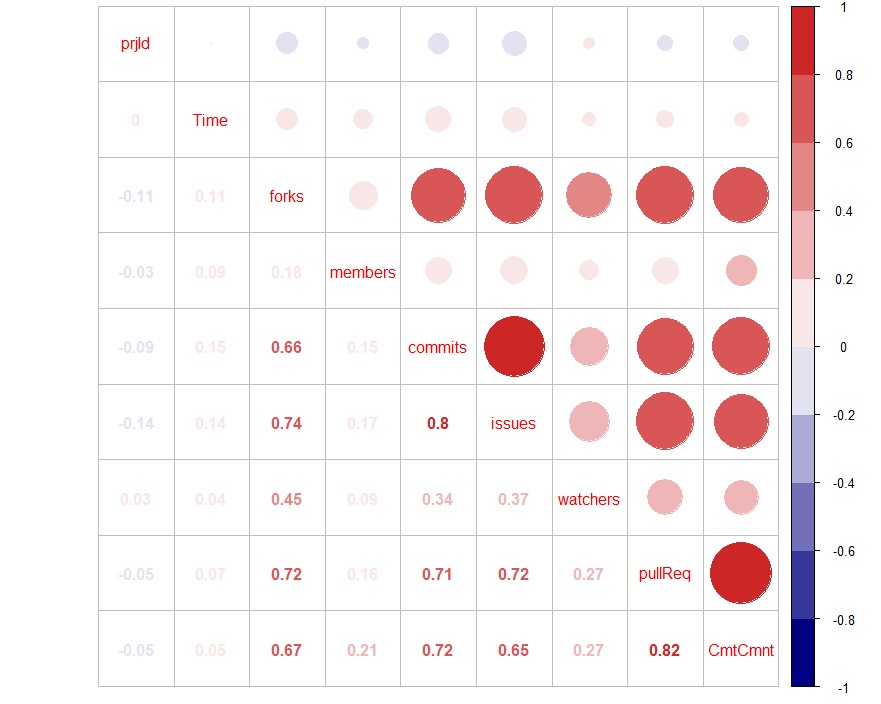
# IV Data Cleaning & Preparation

## 4.1 Importance Analysis

First, we need to find the covariates suitable for study. The method is to find out the connection between the primary variable except for the dependent variable (Forks) and the dependent variable (Forks) and sort them according to the importance degree of correlation to complete the preliminary screening of covariables.

****

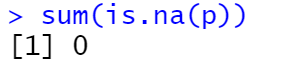
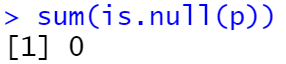
We performed a visual analysis of the correlation between the eight significant variables. It was observed that variables significantly correlated with Forks were Commits, Issues, PullReq, and CommitCmnt. However, graphical methods cannot accurately describe the degree of connection, so we use more accurate methods to sort.

****

As shown in the figure above, the correlation between significant variables and Forks can be sorted in order of importance: Issues (0.74), PullReq (0.72), CommitCmnt (0.67), Commits (0.66), Watchers (0.45), Members (0.18).

Watchers and member were discarded because the correlation coefficient was less than 0.5. Finally, Issues, PullReq, CommitCmnt, and Commits were selected as covariables for the study.

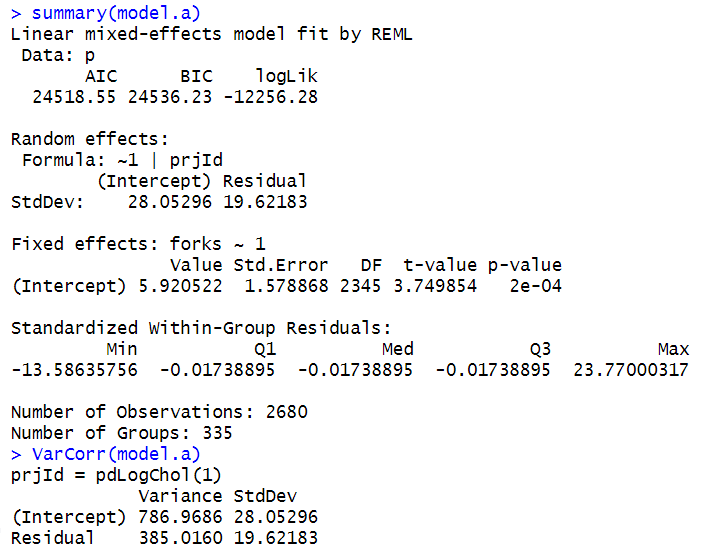
## 4.2 Data Cleaning



Using is. null () function and is.na() function, we find that there is no null value and missing value in the main variables.

# V Multi-level Longitudinal Model

## 5.1 Model A



**Composite Model A** : Forks = 5.92 + e

Estimate of fixed effects: the initial status of Forks at the Time 0 is 5.92(1.58) at 0.05 level of significance. (p-value = 0 < 0.05)

**Variance components:**

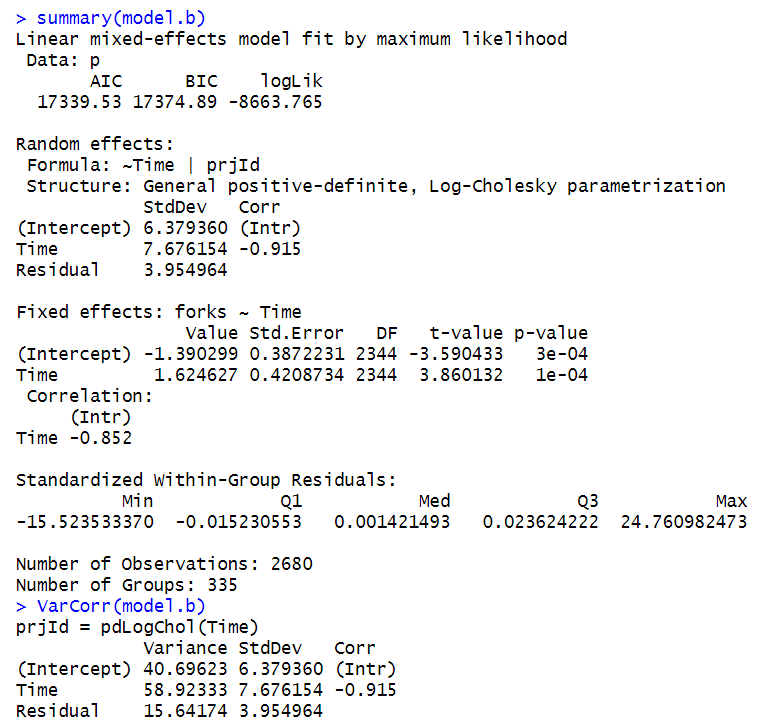
**Level 1** (within project variance) gets the estimate of 385.02(19.62)

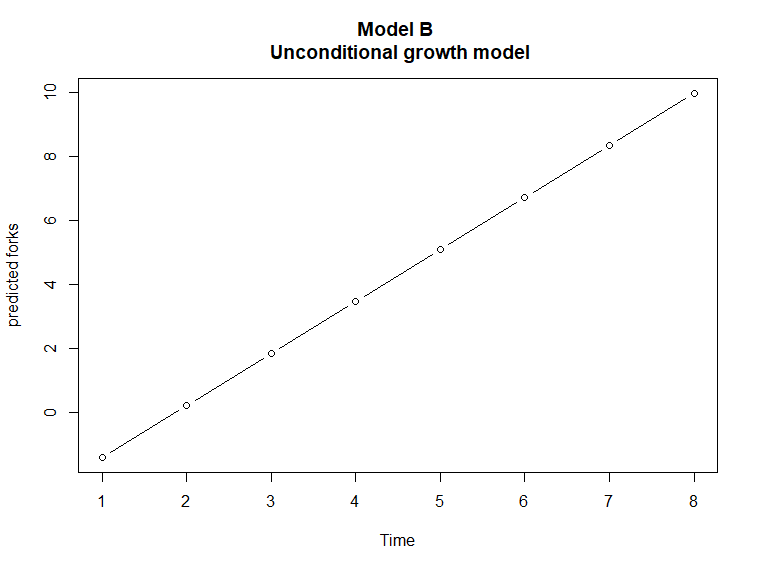
**Level 2** (between project variance) receives the estimate of 786.87(28.05)

**ICC** = 786.87 / (385.02 + 786.87) = 0.6715

Therefore, 67.15% variation in the Forks is attribute to differences among projects.

## 5.2 Model B





Composite Model B:

Level1: Forks = a + b\* Time + j

Level 2: a = -1.39 + y\_0i

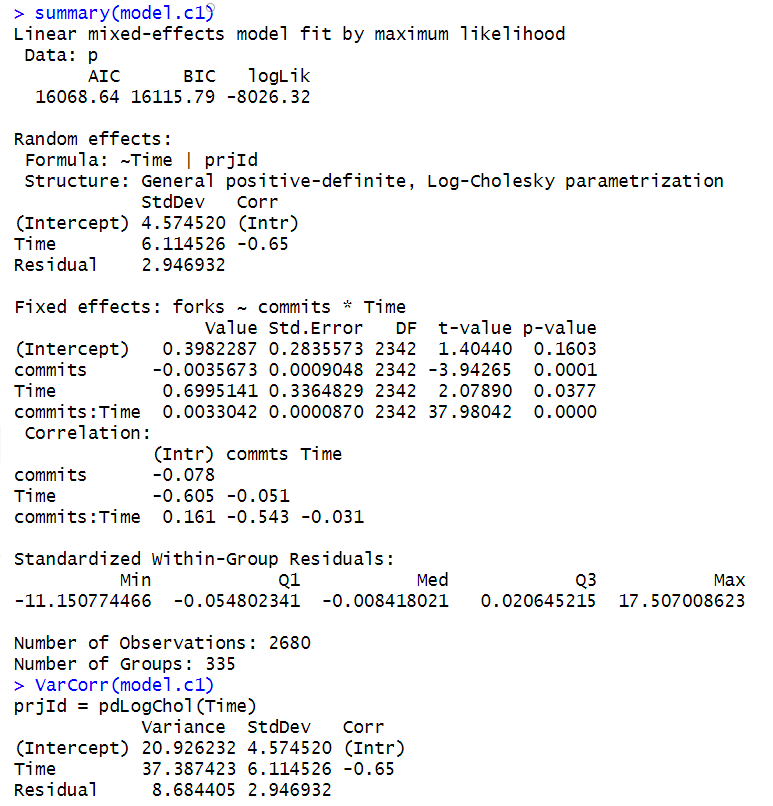
b = 1.62 + y\_1i

Forks = -1.39 + 1.62 \* Time + e , where e = y\_0i + y\_1i \* Time + j

Estimates of fixed effects show significant values. From the result, we can see that the rate of change is the estimates of Time. The estimated rate of change in Forks for projects is -1.39 (p-value < 0.05) where the estimated initial test score is 1.62 (p-value = 0<0.05). Therefore, we can interpret this estimate: there is an increase in Forks over time from one period to the next, there will be an increase of 1.62 in Forks each period.

## 5.3 Model C

### 5.3.1 Model C1

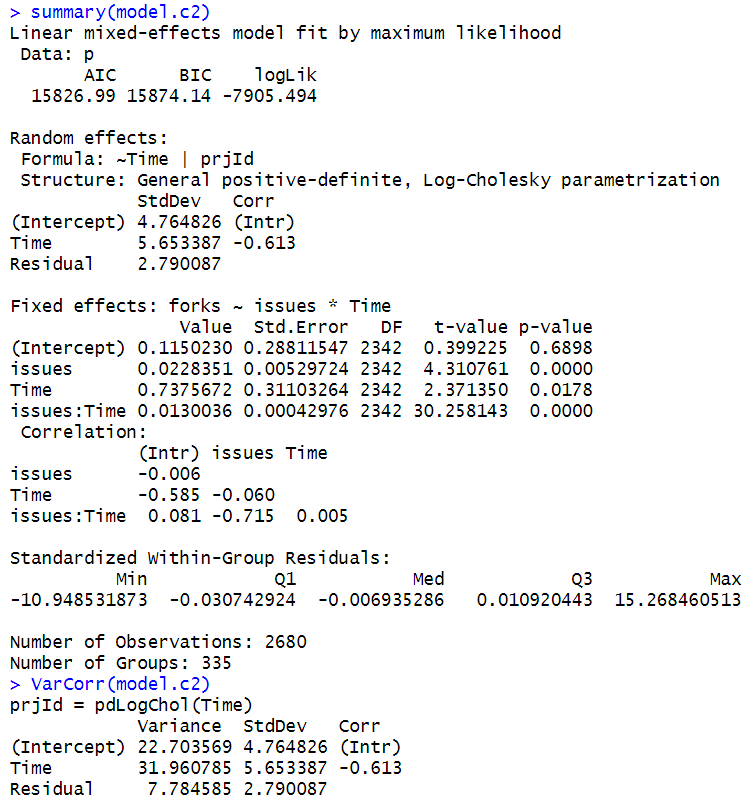


The fixed effects values are presenting the significant impact of Commits on Forks both at initial status(estimate = -0.003 at 0.05 l.o.s) and over time (estimate = 0.003 at 0.05 l.o.s).The estimate initial Forks for projects with Commits is 0.398 at 0.05 level of significance.

The estimate at the rate of change in Forks for projects with Commits is 0.699. Then, there is no significant gap between Forks at the initial status and in the rate of change of Commits.

Rseudo R2 is (40.696-20.926)/40.696 = 0.4858, which means approx. 48.58% of between projects variance in Forks is associated with Commits\*Time.

### 5.3.2 Model C2

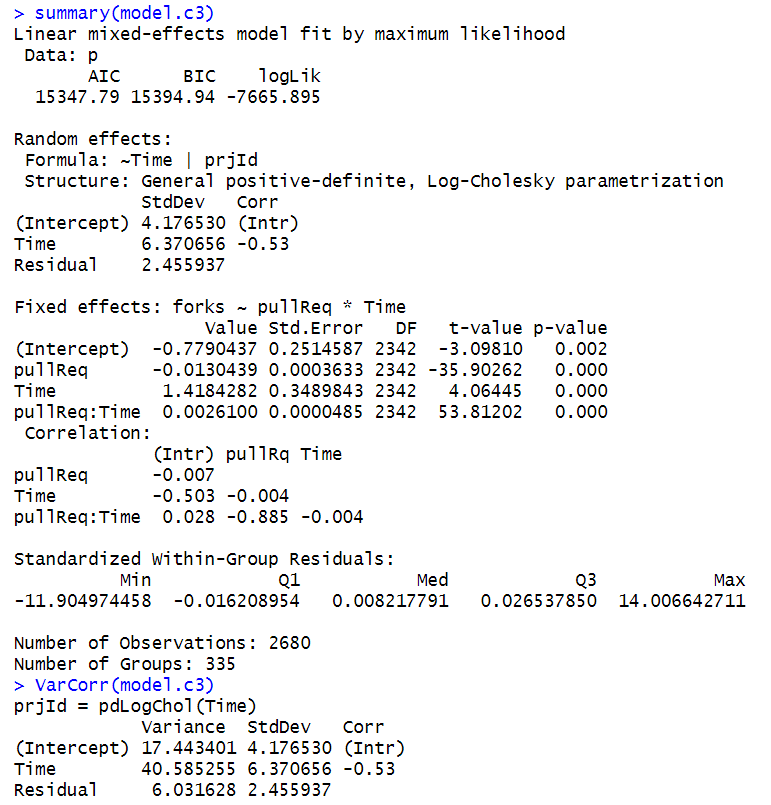


The fixed effects values are presenting the significant impact of Issues on Forks both at initial status(estimate = 0.022 at 0.05 l.o.s) and over time (estimate = 0.013 at 0.05 l.o.s). The initial estimate Forks for projects with Issues is 0.115 at 0.05 level of significance.

The estimate at the rate of change in Forks for projects with Issues is 0.738. Then, there is no significant gap between Forks at the initial status and in the rate of change of Issues.

Rseudo R2 is (40.696-22.703)/40.696 = 0.4421, which means approx. 44.21% of between projects variance in Forks is associated with Issues \*Time.

### 5.3.3 Model C3

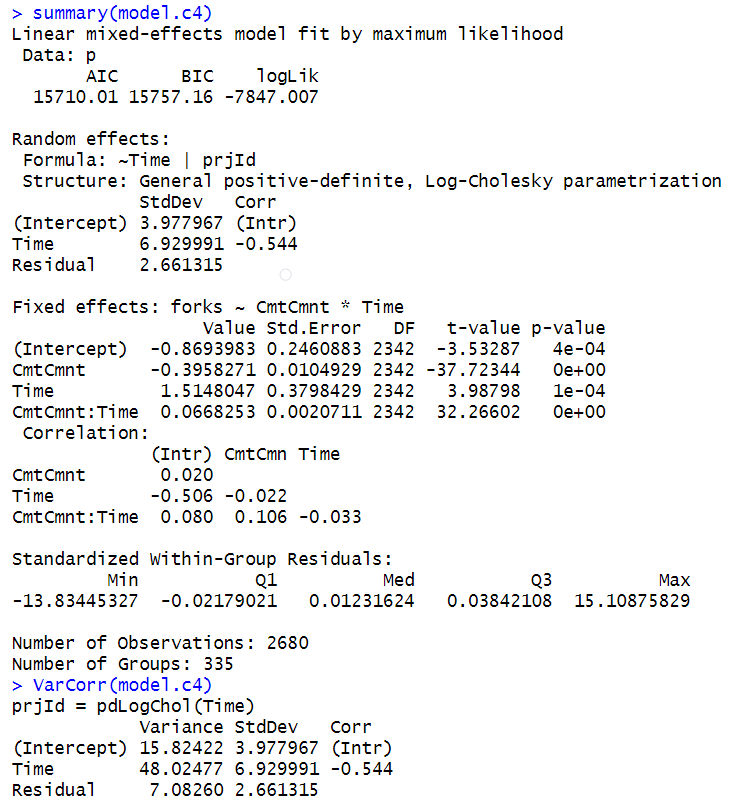


The fixed effects values are presenting the significant impact of PullReq on Forks both at initial status(estimate = -0.013 at 0.05 l.o.s) and over time (estimate = 0.002 at 0.05 l.o.s). The estimate initial Forks for projects with Issues is -0.779 at 0.05 level of significance.

The estimate at the rate of change in Forks for projects with PullReq is 1.418. Then, there is no significant gap between Forks at the initial status and in the rate of change of PullReq.

Rseudo R2 is (40.696-17.443)/40.696 = 0.5713, which means approx. 57.13% of between projects variance in Forks is associated with PullReq \*Time.

### 5.3.4 Model C4



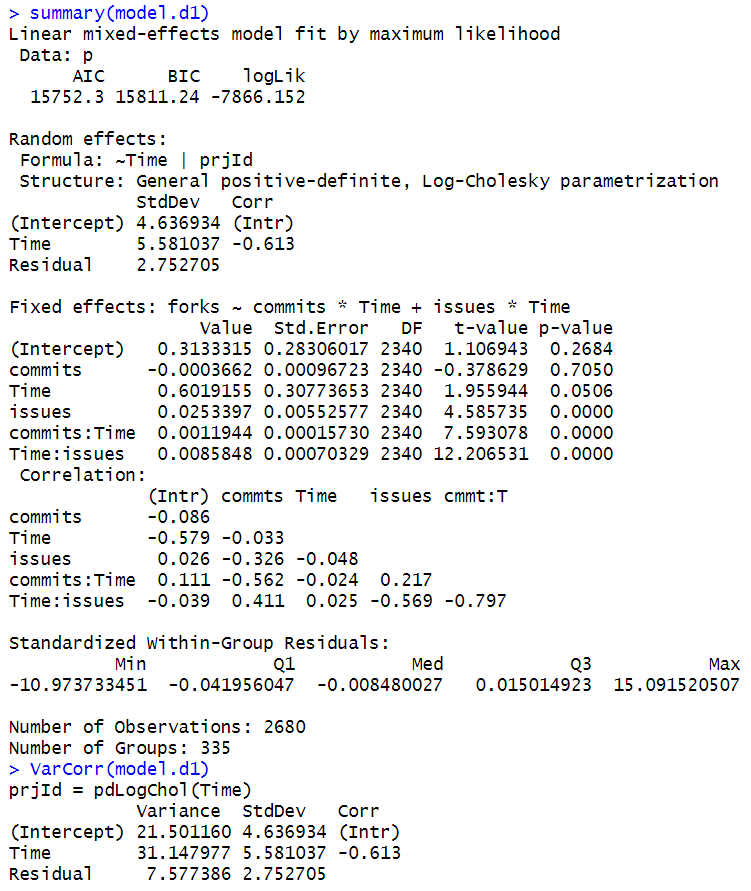
The fixed effects values are presenting the significant impact of CommitCmnt on Forks both at initial status(estimate = -0.395 at 0.05 l.o.s) and over time (estimate = 0.066 at 0.05 l.o.s). The estimate initial Forks for projects with Issues is -0.869 at 0.05 level of significance.

The estimate at the rate of change in Forks for projects with CommitCmnt is 1.514. Then, there is no significant gap between Forks at the initial status and in the rate of change of CommitCmnt.

Rseudo R2 is (40.696-15.824)/40.696 = 0.6111, which means approx. 61.11% of between projects variance in Forks is associated with CommitCmnt \*Time.

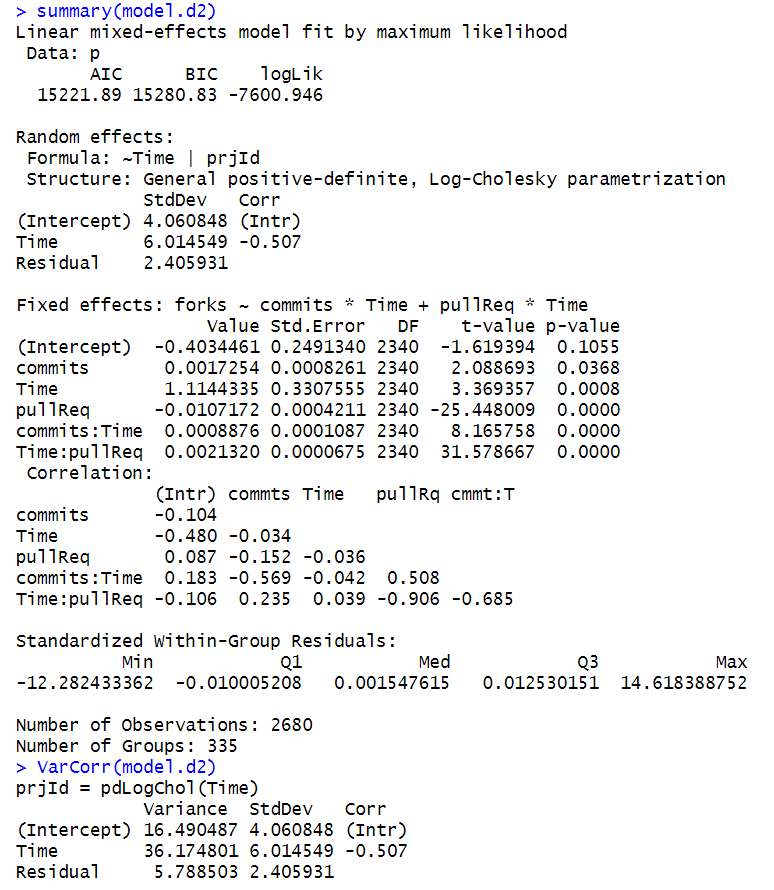
## 5.4 Model D

### 5.4.1 Model D1



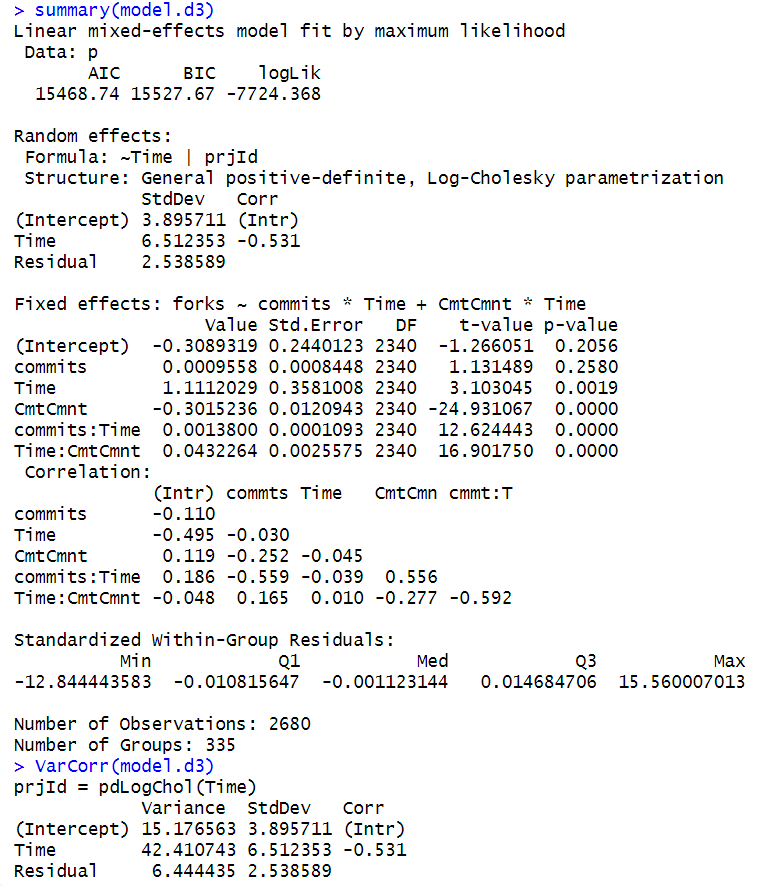
The p-value of Commits is 0.705 (0.705 > 0.05 at 0.05 l.o.s), thus Commits have no significant effect on Forks；The p-value of Commits is 0.0 (0.705 > 0.05 at 0.05 l.o.s), thus Commits have no significant effect on Forks；Issues have a positive impact on Forks. Over time, both Issues and Commits have a positive impact on the Forks. Therefore, the impact of Commits on Forks depends on Time.

### 5.4.2 Model D2



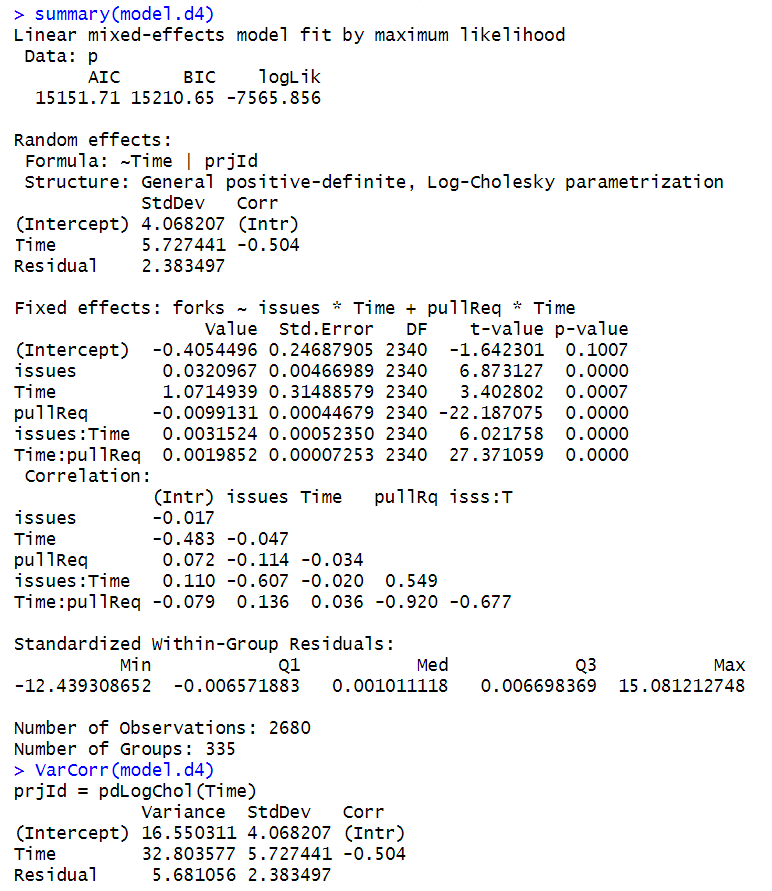
It is worth noting that PullReq has a negative impact on the Forks, but over time, PullReq will have a positive impact on the Forks.

### 5.4.3 Model D3



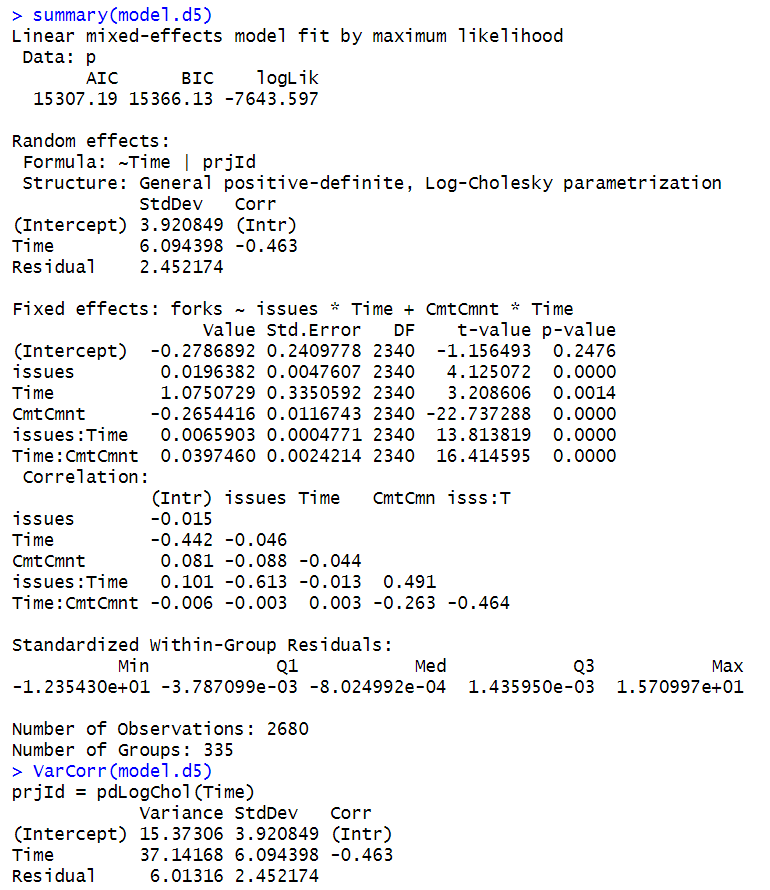
Commits had no significant effect on the Forks (p > 0.05 l.o.s). However, over time, Commits have a weak positive impact on the Forks. CommitCmnt has a negative impact on Forks, but over time, CommitCmnt has a positive impact on Forks.

### 5.4.4 Model D4



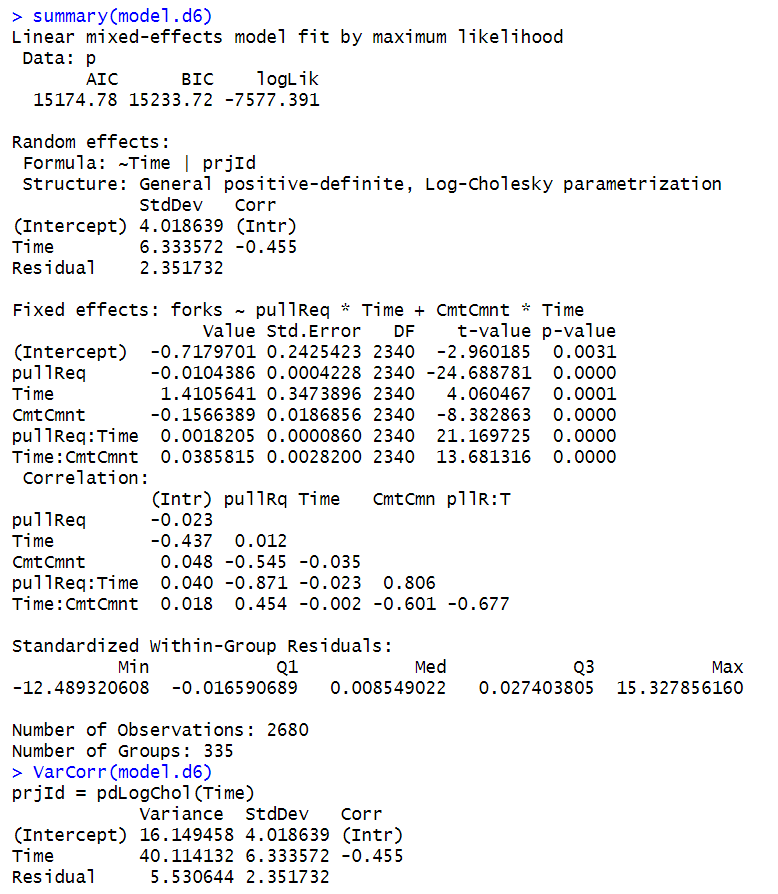
Issues have a positive impact on Forks, and PullReq has a negative impact on Forks. Over time, the positive impact of Issues on Forks will diminish, and PullReq will have a positive impact on Forks.

### 5.4.5 Model D5



Issues have a positive impact on Forks, and CommitCmnt has a negative impact on Forks. Over time, the positive impact of Issues on Forks will diminish, and CommitCmnt will have a positive impact on Forks.

### 5.4.6 Model D6

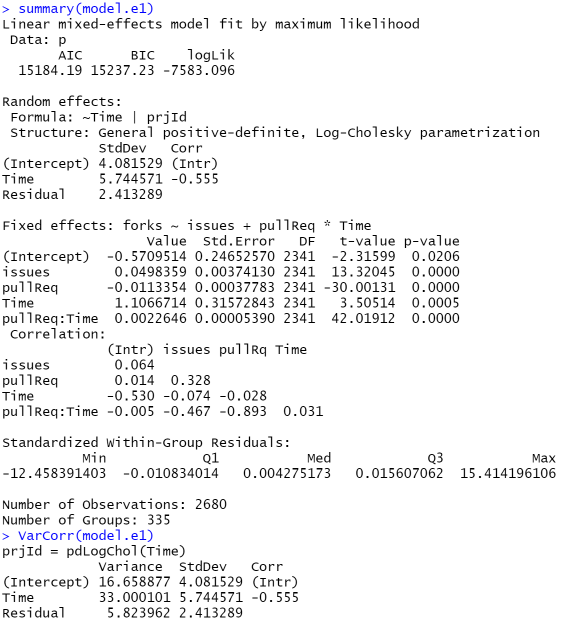


PullReq and CommitCmnt have a negative impact on Forks. Over time, both will have a positive impact on the Forks.

## 5.5 Model E

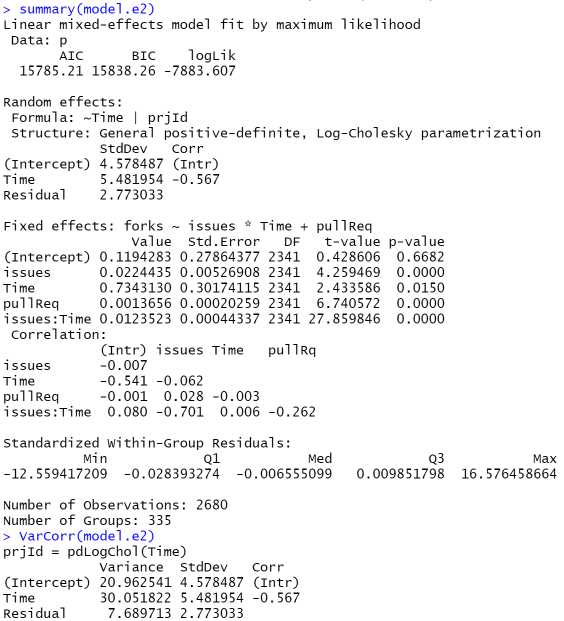
In Model E, we selected two variables, Issues(0.74) and PullReq(0.72), which are most relevant to Forks for analysis.

### 5.5.1 Model E1



Issues have a negative impact on the Forks. In the beginning, PullReq has a negative impact on the Forks, but over time, PullReq will have a positive impact on the Forks.

### 5.5.2 Model E2



Both Issues and PullReq have a positive impact on the Forks. But as time goes by, the impact of Issues on Forks will diminish.

# VI Results & Discussions

## 6.1 Evaluations

|  |  |  |
| --- | --- | --- |
| **Models** | **AIC** | **BIC** |
| Model A | 24518.55 | 24536.23 |
| Model B | 17339.53 | 17374.89 |
| Model C1 | 16068.64 | 16115.79 |
| Model C2 | 15826.99 | 15874.14 |
| Model C3 | 15347.79 | 15394.94 |
| Model C4 | 15710.01 | 15757.16 |
| Model D1 | 15752.30 | 15811.24 |
| Model D2 | 15221.89 | 15280.83 |
| Model D3 | 15468.74 | 15527.67 |
| Model D4 | 15151.71 | 15210.65 |
| Model D5 | 15307.19 | 15366.13 |
| Model D6 | 15174.78 | 15233.72 |
| **Model E1** | **15132.34** | **15203.06** |
| Model E2 | 15290.82 | 15361.54 |

In summary, the results of Model E1 has the lowest value of AIC and BIC. Thus, Model E1 is the best model of all. From Model E1, we can conclude that Issues have a negative impact on the Forks. In the beginning, PullReq has a negative impact on the Forks, but over time, PullReq will have a positive impact on the Forks.

## 6.2 Limitations & Future Work

Although the correlation analysis of influential factors in the development process of GitHub open-source software has drawn many exciting conclusions, we only considered eight significant factors for the study and did not consider subsets of 8 significant factors, such as PullReqCmnt, IssuesCmnt and MemCommitters. Therefore, more influencing factors should be considered for further verification in future studies. Besides, the sample project selected for this article is only 335 open source projects, which is still a minimal number compared to GitHub's more than 12 million projects. Therefore, the results may have some contingency.

# VII Conclusion

This paper analyzed the factors influencing the development of GitHub open-source software, proposed the effects of Issues, PullReq, CommitCmnt, and Commits on Forks under the time dimension, and analyzed the correlation among these factors. In future studies, we will consider more influencing factors and the correlation between multiple influencing factors with larger sample size.

# References

Perez-Riverol, Y., Gatto, L., Wang, R., Sachsenberg, T., Uszkoreit, J., Leprevost, F. da V., … Vizcaíno, J. A. (2016). Ten Simple Rules for Taking Advantage of Git and GitHub. *PLoS Computational Biology*. https://doi.org/10.1371/journal.pcbi.1004947

Blischak, J. D., Davenport, E. R., & Wilson, G. (2016). A Quick Introduction to Version Control with Git and GitHub. *PLoS Computational Biology*. https://doi.org/10.1371/journal.pcbi.1004668

Kalliamvakou, E., Damian, D., Blincoe, K., Singer, L., & German, D. M. (2015). Open source-style collaborative development practices in commercial projects using GitHub. *Proceedings - International Conference on Software Engineering*. https://doi.org/10.1109/ICSE.2015.74

Dabbish, L., Stuart, C., Tsay, J., & Herbsleb, J. (2012). Social coding in GitHub: Transparency and collaboration in an open software repository. *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW*. https://doi.org/10.1145/2145204.2145396

Tsay, J., Dabbish, L., & Herbsleb, J. (2014). Influence of social and technical factors for evaluating contribution in GitHub. *Proceedings - International Conference on Software Engineering*. https://doi.org/10.1145/2568225.2568315