Longitudinal Data Analysis: The Factors of how Commits Numbers Changing overtime in GitHub

Name: Ziteng Li

Student ID: 733922565

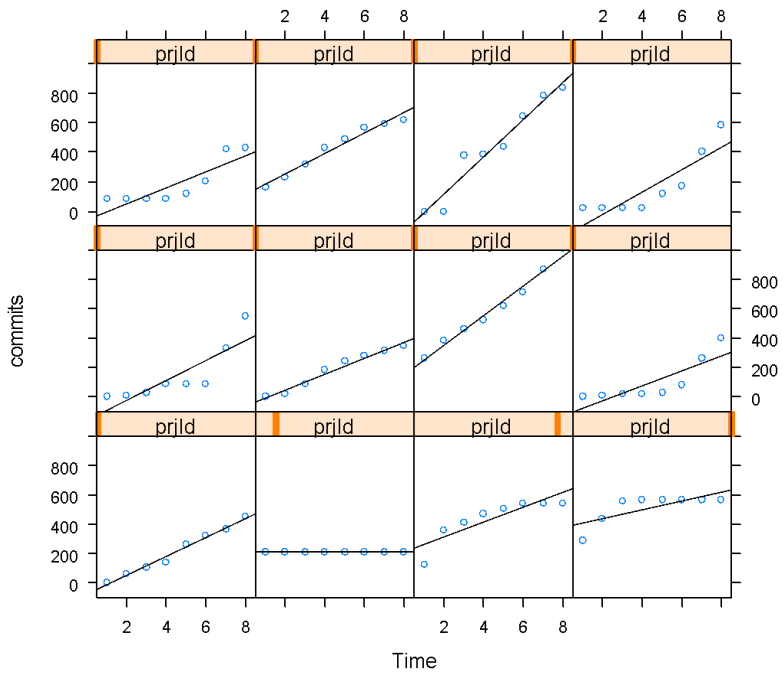
Email: zli786@aucklanduni.ac.nz

Introduction

Nowadays Open-Source Software is developed mostly by decentralised teams of developers cooperating on-line. GitHub portal is an online social network that supports the development of software by virtual teams of programmers (Jarczyk et al., 2014). In this project, it is interesting to investigate is that there are any significant correlations between different project types, and the project commits. We would use the steps and the proposals in the longitudinal data analysis (Punske, J. et al., 2020), to evaluate the research and assess the result. The initial dataset is a panel dataset of projects hosted on GitHub. It contains two years (8 quarters) of data for each project. We used R Studio as our major software; the R language is required in this project.

In the next section, the research questions and hypotheses, we are going to discuss the research question that related to the dataset and give two reasonable hypotheses for our research. In the methodology section, we will discuss five models in longitudinal data analysis strategies. Then, we provide some definitions of the main variables in this research and the method of data cleaning and transformation. In the discussion section, we will interpret our result based on five models, respectively, and we want to find out the essential features that could predict the changes in the number of project's commits.

Research questions and Hypotheses



(Figure 1. The commits changes overtime period randomly selected 12 projects)

The growth plot with superimposed OLS trajectories for 12 projects from the dataset, which indicates that most of the project's commits increase over the period. In this project, we want to indicate what factors make the number of commits change over the period? After understanding the dataset, we found that different owner type of projects makes commits have a different rate of increase, as a result, we have the first hypotheses in this paper.

Hypotheses 1: The GitHub project from organisations may have a higher rate of increases than the project from the individual user.

In terms of finding more factors which may change the number of projects' commits, we analysis the comments and discussion attribute in the dataset, including the comments or discussion of the issues, the commits, the pull requests, and the issues related to pulling requests. We found that the issue comments have more correlation with the commits, so we have the second hypotheses.

Hypotheses 2: The issue comments of the project may increase the commits of this project.

Definition of main variables, Visual Exploration

In the dataset, there are 28 columns, contains 2680 observations. In the dataset, some variables defining the time, such as 'Period', 'Time', 'StartDate', 'EndDate'. Some variables are used to provide the general information about the project on GitHub, including 'Members', 'watchers', 'Committers', 'MemCommitters', 'Health', 'License', 'ContribFile', 'OwnerFollower', 'AvgFollwer', 'OwnerType'. Some variables are the showing the changes of a project during the period, for example, 'Fork', 'Commits', 'Issues', 'PullReq', 'CommitCmnt', 'PullReqCmnt', 'PR Issue Cmnt', 'IssueCmnt', 'PRClosedCnt', 'IssueClosedCnt', 'PRClosedTime', 'IssueClosedTime'.

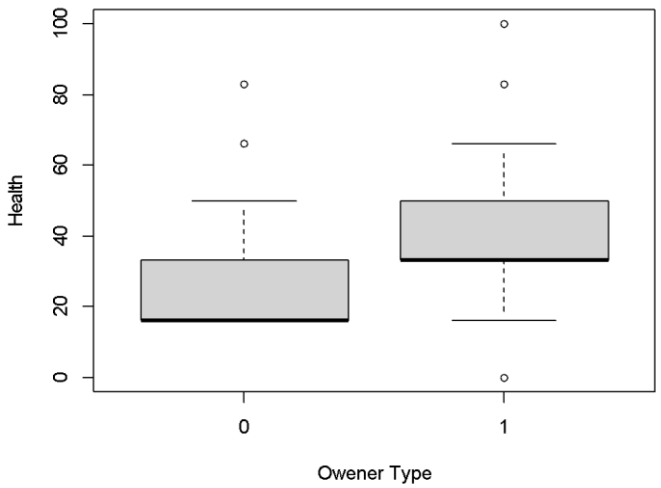
As we mentioned in the research problem, we used the 'Commit', 'time', 'owner type', 'health', 'issue comment', in the data analysis models, and data selection. We selected some variables associated with the research problems. In table 1, some of the main variables' definition and data type has been described.

Variable	Data Type	Definition
PrjID	Numeric	A unique id number for each project
Time	Numeric	A sequence for the time of observations
StartDate	Numeric	Beginning of data collection for this period.
EndDate	Numeric	End of data collection for this period
Commits	Numeric	Total number of coding activities (commits)
Issues	Numeric	Total number of problem/bugs raised or requests for new features
Watchers	Numeric	Total number of people interested in the project (number of users who have this project on their watchlist)
IssueCmnt	Numeric	Total number of discussion/comments on issues.
Health	Numeric	The health indicator of a project scaled (0-100). 100 means best.
OwnerType	String	Type of the owner of the project (organisation, user, etc.)

(Table 1. Details of main variables)

Data cleaning and preparation

- ## Data Cleaning

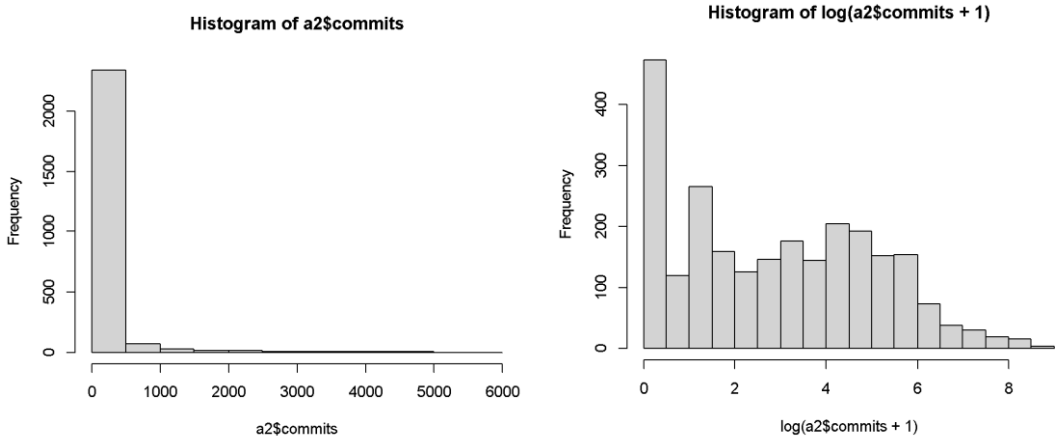


(Figure 2. The box plot of Owner Type against Health)

In order to preserve the diversity and completeness of the data, and by distinguishing the projects is from organisations or individual users, we used the 'owner type' and 'health' variables to separate the dataset. In figure 2, we found that the health of most of the project is in the range from 16 to 66, so we defined that if the health of this project is equal to 100, 83 or 0 as the outliers, then removed 200 observations from the whole dataset.

- ## Data transformation

Log transformation is a substantial transformation with a major effect on distribution shape. In this section, we have used log transformation method for the data transformation steps, as we could see the left histogram below, details that in our dataset, most of the commits number are from 0 to 500, is not met the normality. On the other hand, the histogram on the right shows that the normality of commits is satisfied.



(Figure 3, Left. The histogram of the commit before log transformation)

(Figure 3, Right. The histogram of the commit after log transformation)

Methodology

Based on the Longitudinal Data Analysis steps, we generate five models in this project. At the previous step, we used the log transformation to process our data; therefore, we should use log(commit+1) to fit the models.

Model A, unconditional means model.

AIC	BIC	logLik			
8673.928	8691.375	-4333.964			
Random effects:					
Formula: ~1 | prjId					
	(Intercept)	Residual			
StdDev:	1.842824	1.145646			
Fixed effects: log(commits + 1) ~ 1					
	Value	Std.Error	DF	t-value	p-value
(Intercept)	2.882673	0.1071638	2170	26.8997	0
Covariance					
	Variance	StdDev			
(Intercept)	3.395999	1.842824			
Residual	1.312506	1.145646			

(Table 2. Results of Model A)

Based on the results from table 2, we could conclude that the composite model could predict as Commit = 2.8826 + e. Additionally, fixed effects estimate that the initial status of Commit of average projects is 2. 8826 at 0.01 level of significance.

As regards to the variance components, the level 1 model (within-person variance) gets the estimate of 1.313, the level 2 model (between person) gets the value of 3.396 approximately. Furthermore, for that Intra-class Correlation Coefficient (ICC) of this model, we calculate the formula ICC = 3.396/(1.313 + 3.396) is estimated equal to 0.7212, which means 72% variation in commits is attributable to differences among projects variances.

Model B, The unconditional growth model

AIC	**BIC**	logLik			
6479.019	**6513.915**	-3233.509			
Random effects: Formula: ~Time | prjId					
	StdDev	Corr			
(Intercept)	2.0584636	(Intr)			
Time	0.2567378	-0.435			
Residual	0.5911779				
Fixed effects: log(commits + 1) ~ Time					
	Value	Std.Error	DF	t-value	**p-value**
(Intercept)	**1.4986895**	0.11985280	2169	12.50442	**0**
Time	**0.3075518**	0.01548104	2169	19.86635	**0**
	Variance	StdDev	Corr		
(Intercept)	**4.2372725**	2.0584636	(Intr)		
Time	**0.0659143**	0.2567378	-0.435		
Residual	**0.3494913**	0.5911779			

(Table 3. Results of Model B)

Based on the results of Model B in table 3, we could interpret the fixed effects based on the composite model:

Level 1 : Commit = a + b * Time + i

Level 2: a = 1.499 + y_0i

b = 0.308 + y_1i

Composite model : Commit = 1.499 + 0.307 * Time + e

(while e = I + Time * y_1i + y_0i)

Model C includes the owner type as a predictor of both initial status and rate of change.

The previous model confirmed our research problem; then in model c, we would like to test our Hypothesis one.

AIC	**BIC**	logLik			
6396.322	**6442.85**	-3190.161			
Random effects: Formula: ~Time | prjId					
	StdDev	Corr			
(Intercept)	1.8742899	(Intr)			
Time	0.2561311	-0.51			
Residual	0.5911779				
Fixed effects: log(commits + 1) ~ dummyOwnerType * Time					
	Value	Std.Error	DF	t-value	**p-value**
(Intercept)	**0.5789763**	0.16153053	2168	3.584315	**0.0003**
dummyOwnerType	**1.7072520**	0.22007830	308	7.757476	**0.0000**
Time	**0.2884888**	0.02275497	2168	12.678056	**0.0000**
dummyOwnerType:Time	**0.0353865**	0.03100265	2168	1.141403	**0.2538**
	Variance	StdDev	Corr		
(Intercept)	**4.2372725**	2.0584636	(Intr)		
Time	**0.0659143**	0.2567378	-0.435		
Residual	**0.3494913**	0.5911779			

(Table 4. Results of Model C)

Based on the results of Model C in table 4, we could interpret the fixed effects based on the composite model:

Level 1 : Commit = a + b * Time + j

Level 2: a = 0.579 + 1.707 * OwnerType + y_0i

b = 0.288 + 0.035 * OwnerType + y_1i

Composite model :

COMMIT = 0.579 + 0.74*OWNERTYPE + 0.29*TIME – 0.05*OWNERTYPE*TIME + y_0i + j + Time*y_1i

= 0.31 + 0.74*OWNERTYPE + 0.29*TIME – 0.05*OWNERTYPE*TIME + e

(while e = j + y_0i + Time*y_1i)

Model D

Model D is based on Model C, which includes the owner type as a predictor of both initial status and rate of change, add another factor (issue comment), which may influence the rate of change of commits number, we also want to use this model to test our Hypothesis two.

AIC	**BIC**	logLik			
6153.461	**6211.621**	-3066.731			
Random effects: Formula: ~Time | prjId					
	StdDev	Corr			
(Intercept)	1.7787885	(Intr)			
Time	0.2339922	-0.62			
Residual	0.5744291				
Fixed effects: log(commits + 1) ~ dummyOwnerType * Time + log(issueCmnt + 1) * Time					
	Value	Std.Error	DF	t-value	**p-value**
(Intercept)	**0.5188505**	0.15377654	2166	3.374055	**0.0008**
dummyOwnerType	**1.4914919**	0.21016303	308	7.096833	**0.0000**
Time	**0.2709900**	0.02115545	2166	12.809469	**0.0000**
log(issueCmnt + 1)	**0.6151127**	0.04220196	2166	14.575455	**0.0000**
dummyOwnerType:Time	**0.0140534**	0.02911441	2166	0.482697	**0.6294**
Time:log(issueCmnt + 1)	**-0.0378925**	0.00686449	2166	-5.520070	**0.0000**

(Table 5. Results of Model D)

Based on the results of Model D in table 5, we could interpret the fixed effects based on the composite model:

Level 1 : Commit = a + b * Time + j

Level 2: a = 0.5188 + 1.491 * OwnerType + 0.615 * IssueCmnt + y_0i

b = 0.271 + 0.014 * OwnerType – 0.038 * IssueCmnt + y_1i

Composite model :

COMMIT = 0.5188 + 1.491*OWNERTYPE + 0.615*ISSUECMNT + y_0i + (0.271 + 0.014*OWNERTYPE – 0.038*ISSUECMNT + y_1i)*Time + j

= 0.316 + 1.491*OWNERTYPE + 0.615*ISSUECMNT + 0.271*TIME+ 0.014*OWNERTYPE*TIME- 0.038*ISSUECMNT*TIME+ e

(while e = j + y_0i + Time*y_1i)

Model E.

Based on the result from Model D, we found that there is a non-significant impact of the owner type over the time, so we removed the time variable and fit the model again to check the changes.

AIC	**BIC**	logLik			
6151.693	**6204.038**	-3066.847			
Random effects: Formula: ~Time | prjId					
	StdDev	Corr			
(Intercept)	1.7790304	(Intr)			
Time	0.2337471	-0.62			
Residual	0.5745966				
Fixed effects: log(commits + 1) ~ dummyOwnerType * Time + log(issueCmnt + 1) * Time					
	Value	Std.Error	DF	t-value	**p-value**
(Intercept)	**0.4855950**	0.13717920	2167	3.539859	**4e-04**
dummyOwnerType	**1.5556401**	0.16233500	308	9.582900	**0e+00**
log(issueCmnt + 1)	**0.6140397**	0.04214182	2167	14.570792	**0e+00**
Time	**0.2778869**	0.01538822	2167	18.058419	**0e+00**
log(issueCmnt + 1):Time	**-0.0372632**	0.00676521	2167	-5.508063	**0e+00**

(Table 6. Results of Model E)

Based on the results of Model E in table 6, we could interpret the fixed effects based on the composite model:

Level 1 : Commit = a + b * Time + j

Level 2: a = 0.486 + 1.556 * OwnerType + 0.614 * IssueCmnt + y_0i

b = 0.277 – 0.037 * IssueCmnt + y_1i

Composite model :

COMMIT = 0.486 + 1.556*OwnerType + 0.614*IssueComment + y_0i + (0.277 – 0.037*IssueComment + y_1i)*Time + j

= 0.486+ 1.556*OwnerType + 0.614*IssueComment + 0.277*TIME
- 0.038*IssueComment*TIME + e

(while e = j + y_0i + Time*y_1i)

Results and discussions

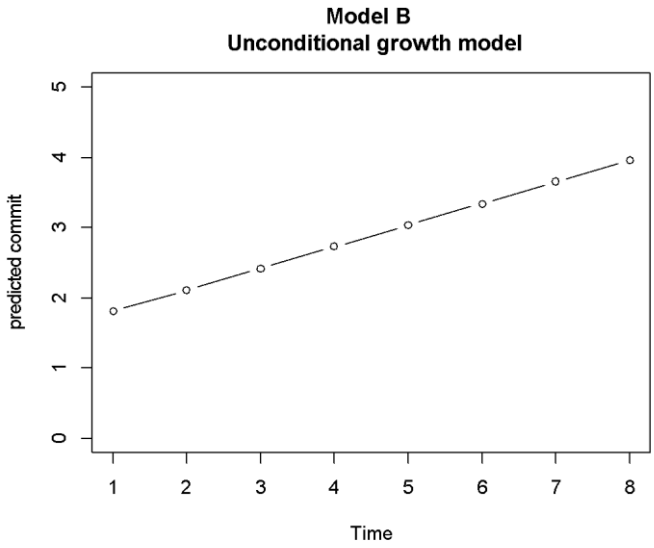
In this section, we will analyse the results of our models, and test the hypothesis; meanwhile, compare and choose the best model.

● Interpretation of the results

Model B.

The estimate of the time is significant at 0.01 level of significance with the p-value is 0, while at initial status the estimated value of commit (after log calculation) is 1.499 (0.01 level of significance). This means that for each period of measured time, the commit number of the project will increase a value of 0.307 approximately.

As regards to the variance components, the level 1 model (within-person variance) gets the estimate of 0.349, the level 2 model (between person) receives the estimate of 4.237 for the initial status and 0.065 for the rate of change.

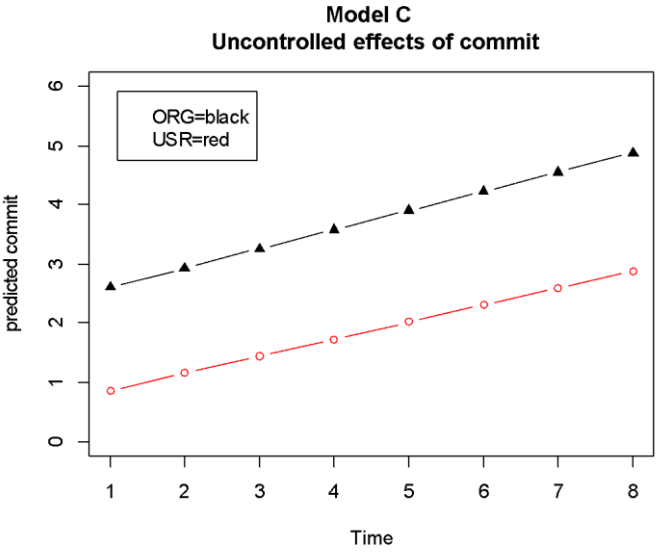


(Figure 4. The fitted line of composite model – model B)

The graph shows the fitted line for the composite model. We could see the commit number of the project at the initial around 1.806, and it increases gradually with 20.4% of the rate of change for every period.

Model C.

The fixed effects values are presenting the significant positive impact of owner type on commit number both at initial status (estimate = 1.707 at 0.01 level of significance) and over time (estimate = 0.288 at 0.05 level of significance). Additionally, the estimate initial commit number for the average 'USR' (an individual user builds the project) is 0.578 at 0.01 level of significance. The estimate at the rate of change in commit numbers for 'USR' type project is 0.288. As regards to the variance components, between-person variances receive the estimate of 4.237 at the initial status and of 0.066 at the rate of change. In comparison, Within-person variance receives the value of 0.349.



(Figure 5. *The fitted line of the composite model for different owner type*)

The plot displays the two-separate trend for 'ORG' (organisation) and 'USR' (individual user) projects' commits number. We can identify the higher value of commit for 'ORG' across

eight time periods compared to 'USR' project. While the rate of change in 'ORG' is also a little bit higher than 'USR', and this gap seems to increase over time. This plot also verifies the hypothesis one is correct.

Model D.

Controlling for the effect of project commit number, the estimated differential in initial test score between different owner type with 'ORG' and 'USR' project is equal to 0.519 (at 0.01 level of significance). While there is a significant impact of the number of issue comments on commit number of projects at the beginning of the study, and there is a significant value at the rate of change (0.00<0.01 level of significance). However, over time, owner type has a non-significant impact on commit number at a value of 0.000 for each measured period (0.629 > 0.01 level of significance)

Model E.

Controlling for the effect of project commit number, the estimated differential in initial test score between different owner type with 'ORG' and 'USR' project is equal to 0.486 (at 0.01 level of significance). While there is a significant impact of the number of issue comment on commit number of projects at the beginning of the study, and there is a significant value at the rate of change (0.00<0.01 level of significance). As a result, the issue comment will decrease the commit number of a project with a minimal value over time.

In conclusion, we have tested our two hypotheses. For **hypotheses 1**, we verified this hypothesis in model C, and the results show that the GitHub project from organisations has a higher rate of increases than the project from the individual user. Meanwhile, we could conclude that the value of commit for 'ORG' across eight time periods is higher than 'USR' project. For **hypotheses 2**, we verified this hypothesis in Model D, and Model E, both of the results show that the issue comments may decrease the number of commits over the time with the rate of change is 0.038 approximately. So, the second hypotheses are incorrect.

● Assess and evaluate models and results

	Model A	Model B	Model C	Model D	Model E
AIC	8673.928	6479.019	6396.322	6153.461	**6151.693**
BIC	8691.375	6513.915	6442.85	6211.621	**6204.038**

In summary, model E results in the lowest value for BIC and AIC.

We can use either AIC or BIC in order to choose the best model. In this case, we prefer using BIC because a more significant difference between two value of AIC than BIC in model D and E is identified (~3 < ~7). This may be because we removed the owner type in the rate of change of Model D. As a result, Model E is the best model of all (based on AIC criteria).

From model E, we can conclude that commit number change over time and the commit result will be influenced by the issue comments at the early stage and by owner type at the early stage and over the period.

References

Jarczyk, O., Gruszka, B., Jaroszewicz, S., Bukowski, L., Wierzbicki, A. (2014). GitHub Projects. Quality Analysis of Open-Source Software. In: Aiello L.M., McFarland D. (eds) Social Informatics. SocInfo 2014. Lecture Notes in Computer Science, 8851. Springer, Cham. https://doi.org/10.1007/978-3-319-13734-6_6

Punske, J., Sanders, Nathan C., & Fountain, Amy. (2020). Language invention in linguistics pedagogy (First ed., Oxford scholarship online).