Analysis the commit activity for OSS projects.

Catalogue

1. Introduction

Open source software projects are software developed and maintained by developer and volunteers. Because of their open source nature, open source software licenses facilitate collaboration and sharing, allowing others to modify the source code and incorporate those changes into their own projects. Open source platforms encourage users to access, view, and modify open source software at any time, as long as they let others do the same when they share work.

In the open source platform, not only developers but also non developers can participate in the project. The diversity of roles has a positive contribution to the progress of the project, because different types of OSS project participants have different types of knowledge, and this diversity in the community will have a positive impact on the success of the project (Daniel et al., 2013).

Daniel et al. (2013) also found that diversity has a positive impact on market success. In recent years, the economic benefits of open source projects attract more and more software companies to cooperate with or provide sponsorship. There is evidence that 78% of enterprises are running OSS (Vaughan Nichols, 2015). In order to enable software companies to select OSS projects worthy of cooperation among numerous open source projects, a good understanding of evolution of OSS projects is an essential step.

As the largest open source community in the world, GitHub has millions of open source projects. This project analyzes a panel dataset of projects hosted on GitHub to help Mr. Richard Thompson, a software architect at Mega software better understand OSS projects' progress and how they change over time.

2. Research questions and Hypotheses

By analyzing the panel data provided by GitHub, we explore the internal relationship and progress process of OSS. The dataset contains two years (8 quarters) of data for

each project。 This dataset contains 28 attributes including "prjid", "period", "time", "StartDate", "enddate", "Forks", "members", "commitments", "issues", "watchers", etc. In order to help Richard Thompson better understand the situation of this project, we need to select the attributes that can best represent the progress of the project.

2.1 Research question

Delone and McLean's (1992) model provides two methods to measure the success of OSS. The first is whether the project has been completed, and the second is whether the project has achieved the goal. However, because OSS has been in continuous development, it is difficult to say whether it is completed or not. The second is different from traditional projects, there is no formal development process, so there is a lack of formal requirements specification (Crowston et al., 2013).

Considering that OSS is a continuous process, we can use the attributes that represent the development process of the project in the dataset as a measure of whether OSS is progressing. There are many ways for users to contribute to OSS, such as raising issue, editing document, coding, etc. However, only when the project owner accepts the user's contribution and merges the changes of other users into his own project, will commits appear in the project owner's project. Commits represents the total number of coding activities, which can be used as the main contribution of users in the OSS development process and the performance of user participation.

Our main research questions are: do trajectories of total number of coding activities different by: (1) total number of discussion / comments on Commons; and (2) Type of the owner of the project (organization, user, etc.）?

2.2 Hypotheses

Emotion has great influence on work efficiency, task quality, creativity, team relationship and job satisfaction. Comments on commits convey users' feelings for the OSS projects they participate in. For example, users tend to have more negative

submission comments for java development projects, and more positive feelings for projects with more distributed teams (Guzman et al., 2014). The emotions conveyed in these comments are likely to affect people's enthusiasm to participate in the project, and the number of comments may also affect the number of user commits.

H1. OSS projects with more comments on commits may have more commits.

Some users contribute OSS projects to improve their existing skills or build public artifacts that help them grow a reputation (and a career). These users will be more willing to participate in the projects owned by the organization. These OSS projects are more formal and standardized, and some of them exist for profit or to explore excellent talents. On the other hand, some organizations' financial support for external developers also encourages other users to participate more actively in projects owned by the organization. This promotes the number of user commits.

H2. Projects owned by organization have more commits than other users。

3. Definition of main variables, Visual Exploration

3.1 Definition of main variables

The data set used in this study is from GitHub, the largest open source community in the world. This data set allows us to observe the evolution of the number of user commits in two years, because the data set includes two years (8 quarters) of data for each project. We define four main variables to analyze this evolution.

```
> pd <- read.csv(file.choose(),header=TRUE)
> names(pd)
 [1] "X."              "prjId"           "Period"          "Time"            "StartDate"       "EndDate"         "forks"
 [8] "members"         "commits"         "issues"          "watchers"        "pullReq"         "CmtCmnt"         "pullReqCmnt"
[15] "PR.Issue.Cmnt"   "issueCmnt"       "committers"      "MemCommitters"   "PRClosedCnt"     "IssueClosedCnt"  "PRClosedTime"
[22] "IssueClosedTime" "Health"          "Licence"         "ContribFile"     "OwnerFollower"   "AvgFollower"     "OwnerType"
> dim(pd)
[1] 2680   28
```

```
> str(pd)
'data.frame':   2680 obs. of  28 variables:
 $ X.              : int  1 2 3 4 5 6 7 8 9 10 ...
 $ prjId           : int  2647 2647 2647 2647 2647 2647 2647 2647 3085 3085 ...
 $ Period          : chr  "2011-1" "2011-4" "2011-7" "2011-10" ...
 $ Time            : int  1 2 3 4 5 6 7 8 1 2 ...
 $ StartDate       : chr  "1/01/2011 12:00:00 a.m." "1/04/2011 12:00:00 a.m." "1/07/2011 12:00:00 a.m." "1/10/2011 12:00:00 a.m." ...
 $ EndDate         : chr  "1/04/2011 12:00:00 a.m." "1/07/2011 12:00:00 a.m." "1/10/2011 12:00:00 a.m." "1/01/2012 12:00:00 a.m." ...
 $ forks           : int  1 1 1 2 2 3 4 6 69 117 ...
 $ members         : int  4 4 5 6 6 6 6 6 58 59 ...
 $ commits         : int  81 81 81 81 120 205 423 429 1065 2221 ...
 $ issues          : int  5 5 5 5 5 6 6 6 92 257 ...
 $ watchers        : int  2 2 2 2 8 24 30 37 855 1027 ...
 $ pullReq         : int  11 13 13 13 13 16 16 16 184 518 ...
 $ CmtCmnt         : int  0 0 0 0 0 0 3 3 37 169 ...
 $ pullReqCmnt     : int  0 0 0 0 0 0 0 0 47 47 ...
 $ PR.Issue.Cmnt   : int  3 3 3 3 3 3 3 3 317 1034 ...
 $ issueCmnt       : int  3 3 3 3 3 3 3 3 346 1176 ...
 $ committers      : int  6 6 6 6 7 7 7 7 63 95 ...
 $ MemCommitters   : int  3 3 3 3 4 4 4 4 19 29 ...
 $ PRClosedCnt     : int  6 8 8 8 8 10 10 10 100 299 ...
 $ IssueClosedCnt  : int  0 0 0 0 0 0 0 0 26 251 ...
 $ PRClosedTime    : num  858 25442 25442 25442 25442 ...
 $ IssueClosedTime: num  NA NA NA NA NA ...
 $ Health          : int  66 66 66 66 66 66 66 66 83 83 ...
 $ Licence         : chr  "Other" "Other" "Other" "Other" ...
 $ ContribFile     : chr  "https://api.github.com/repos/arx/ArxLibertatis/contents/CONTRIBUTING.md" "https://api.github.com/repos/arx/Ar
is/contents/CONTRIBUTING.md" "https://api.github.com/repos/arx/ArxLibertatis/contents/CONTRIBUTING.md" "https://api.github.com/repos/ar
rtatis/contents/CONTRIBUTING.md" ...
 $ OwnerFollower   : int  6 6 6 6 7 7 8 8 1 1 ...
 $ AvgFollower     : int  6 6 5 4 5 6 6 6 1 1 ...
 $ OwnerType       : chr  "ORG" "ORG" "ORG" "ORG" ...
```

Prjid: A unique id number for each project. Represents the ID of each project, and each ID is a unique identifier for the project. Prjid is used to distinguish one project from others.

Time: A sequence for time of observations includes records for eight quarters in two years. Int type, ranging from 1 to 8, the number represents in which quarter.

Commits: Total number of coding activities (commits). Refers to the user's activity level in the OSS project, int type, ranging from 0 to 10706. The higher the commit value is, the more active the project is.

```
> # minimum and maximum
> min(pd$commits)
[1] 0
> max(pd$commits)
[1] 10706
```

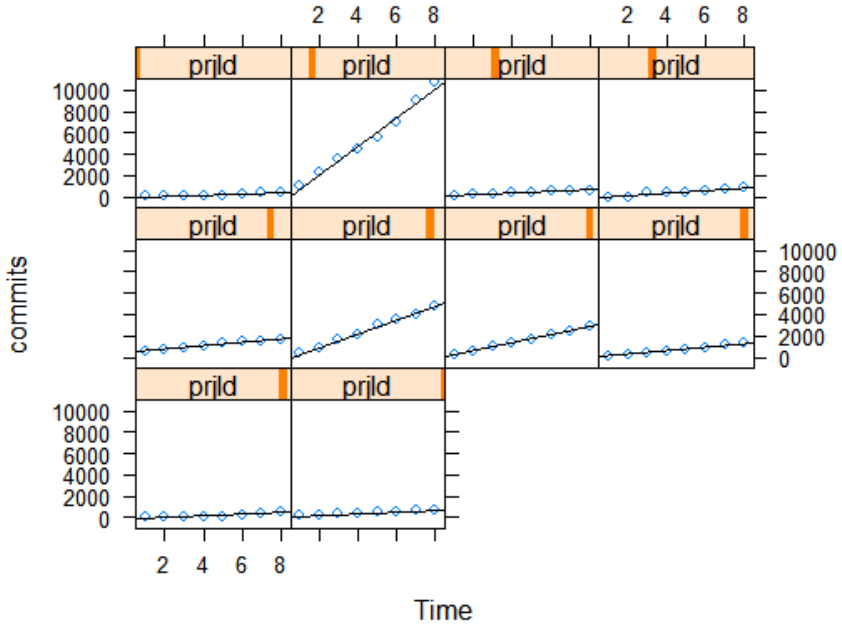
CmtCmnt: total number of discussion / comments on commits, refers to the enthusiasm of users to comment on commits, int type. The higher the CmtCmnt value is, the more positive the user's comments on commits are.

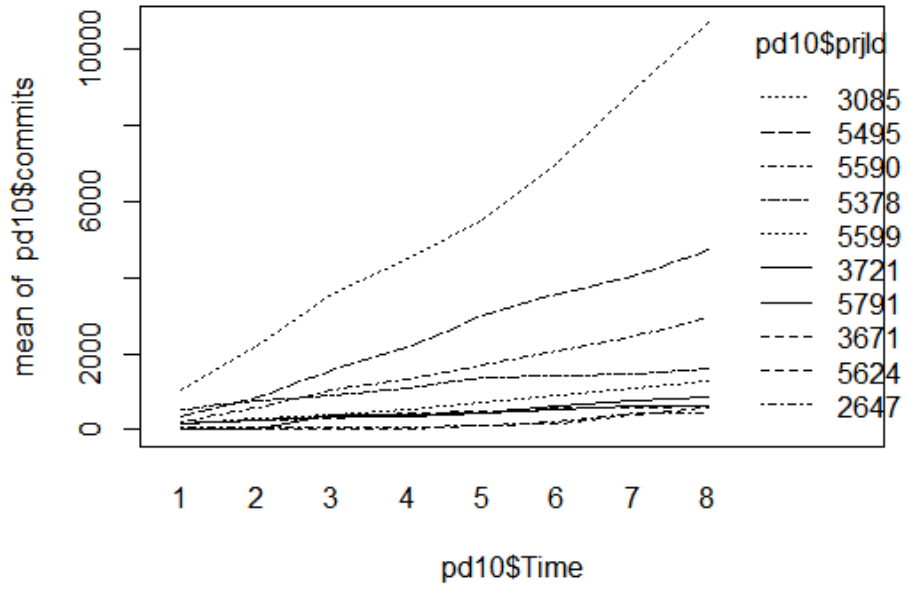
OwnerType: Type of the owner of the project (organization, user, etc.). The owner of OSS project, chr type, is divided into two categories, organization and user.

```
> table(pd$OwnerType)

 ORG  USR
1512 1168
```

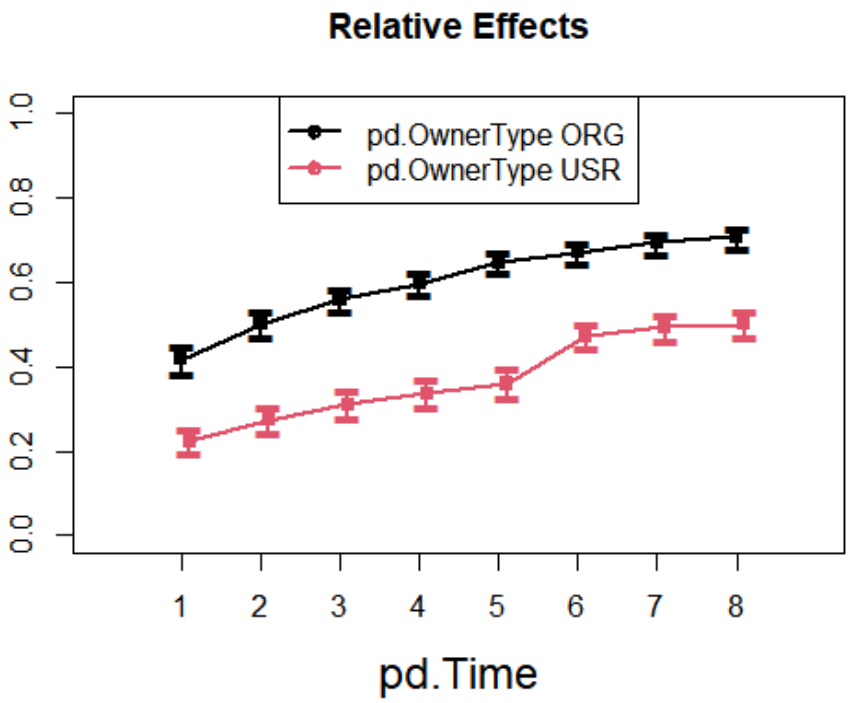
3.2 Visual Exploration

This is the empirical growth plots with superimposed OLS trajectories for 10 projects in the study, which Identified a suitable functional form for the level-1 sub-model





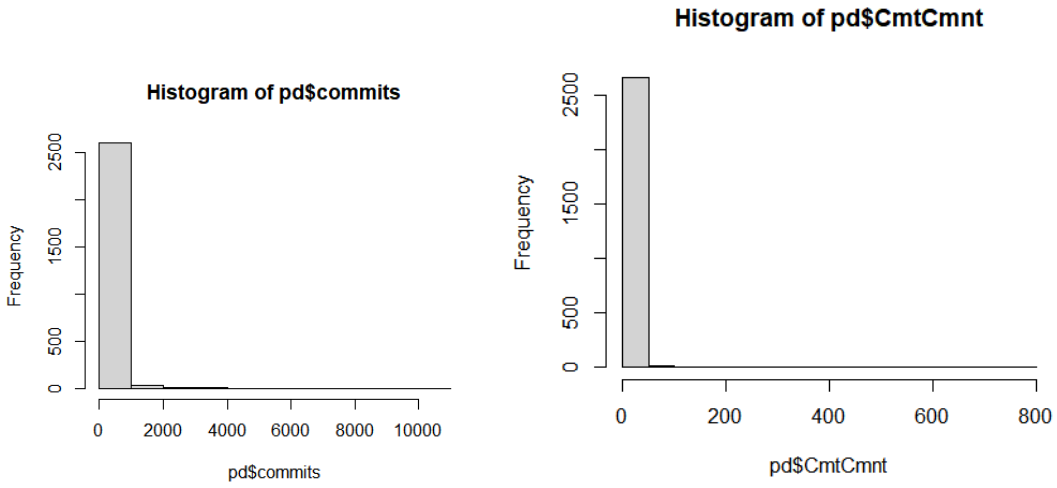
Commits and Time appears linear between time 1 and 8. Prjid 3085: Commits tends to increase rapidly from quarter of 1 to 8. Prjid 5495 and 55990： Commits increase steadily over 8 quarters. The rest projects: Commits increased slowly from quarter of 1 to 8. Generally, there is considerable changes in commits over 8 quarters. And it is a linear trajectory for the identified change.

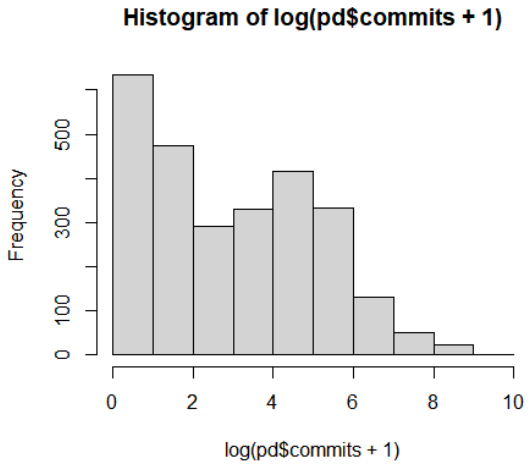


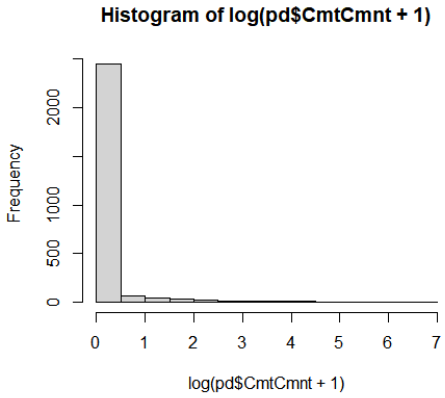
The owner will influence the trajectory of the project over time. Regardless of the owner, the commitment trajectory of the project is gradually rising from quarter1 to quarter8. Projects owned by organizations always have more commits than users in eight quarters.

3.3 Assumption checks

(1) Check Normality







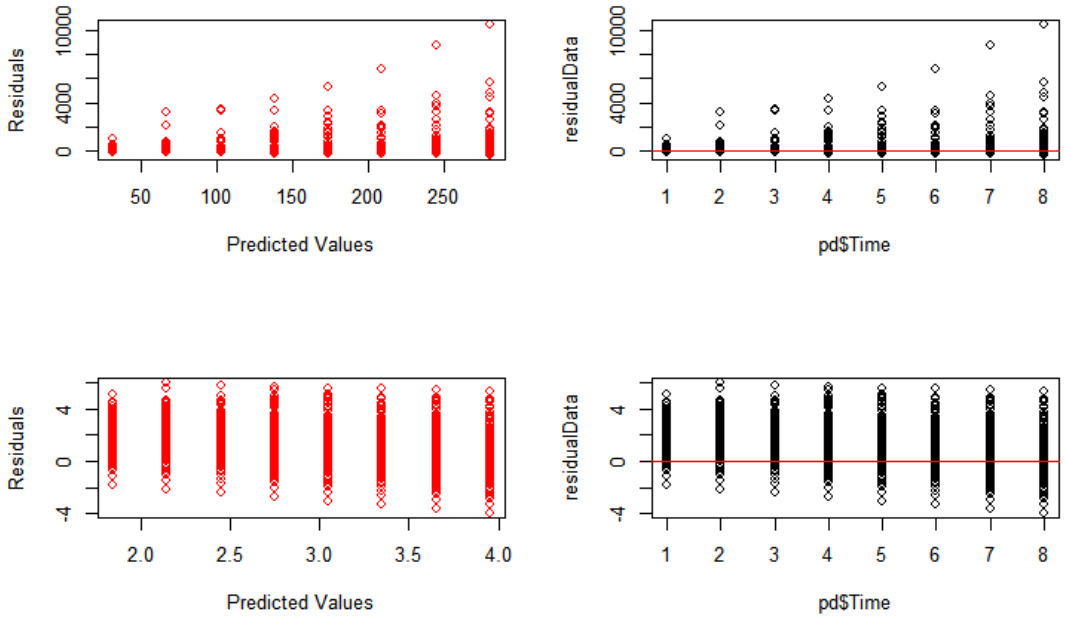
Log transformation reduces the right skewness.

```
> shapiro.test(pd$commits)

        Shapiro-Wilk normality test

data:  pd$commits
W = 0.27672, p-value < 2.2e-16

> shapiro.test(log(pd$commits+1))

        Shapiro-Wilk normality test

data:  log(pd$commits + 1)
W = 0.9428, p-value < 2.2e-16
```

Shapiro-Wilk Test

We want to see non-significant results. However, both p_value=2.2e-16 < 0.01 (level of significance). SW tests gave significant results, thus normality assumption is not satisfied even applying the log transformation.

(2) Check Homoscedasticity



Scatter plots provide evidence of heteroscedasticity. For all levels of prediction, there are errors with different variances. The error values are mainly distributed in the second half of the scatter plot. After log transformation, errors are almost evenly distributed to both sides of the predicted value, resulting in the Homoscedasticity.

4. Data cleaning and preparation

4.1 Transformation

In order to facilitate the construction of the model and simplify the analysis process, we transform the data set. Remove useless variables and select only the main variables we need to form a data set. Considering that there may be computation in the modeling process, we convert the OwerType of chr type to num type.

```
> OwnerType1<-NULL
> OwnerType1<-(pd1$OwnerType=="ORG")*1
> pd2<-cbind(OwnerType1, pd1)
> pd<-pd2[,c("prjId","commits","Time","CmtCmnt","OwnerType","OwnerType1")]
> str(pd)
'data.frame':   2680 obs. of  6 variables:
 $ prjId     : int  2647 2647 2647 2647 2647 2647 2647 2647 3085 3085 ...
 $ commits   : int  81 81 81 81 120 205 423 429 1065 2221 ...
 $ Time      : int  1 2 3 4 5 6 7 8 1 2 ...
 $ CmtCmnt   : int  0 0 0 0 0 0 3 3 37 169 ...
 $ OwnerType : chr  "ORG" "ORG" "ORG" "ORG" ...
 $ OwnerType1: num  1 1 1 1 1 1 1 1 1 1 ...
```

4.2 Check missing value

Although there are missing values in the data set, the number of missing values of the main variables we selected is 0.

```
> colSums(is.na(pd2))
     OwnerType1              X.            prjId           Period             Time
              0               0                0                0                0
      StartDate         EndDate           forks          members          commits
              0               0                0                0                0
         issues        watchers         pullReq         CmtCmnt     pullReqCmnt
              0               0                0                0                0
  PR.Issue.Cmnt       issueCmnt      committers   MemCommitters      PRClosedCnt
              0               0                0                0                0
 IssueClosedCnt    PRClosedTime IssueClosedTime           Health          Licence
              0            2069             2210              368                0
    ContribFile   OwnerFollower     AvgFollower        OwnerType
              0               0                0                0
> colSums(is.na(pd))
     prjId    commits       Time    CmtCmnt OwnerType1
         0          0          0          0          0
```

5. Methodology

5.1 Model A - Unconditional means model

```
> #Model A
> library(nlme)
> model.a <- lme(log(commits+1)~ 1, pd, random= ~1 |prjId)
> summary(model.a)
Linear mixed-effects model fit by REML
  Data: pd
       AIC      BIC    logLik
  9308.617 9326.296 -4651.308

Random effects:
 Formula: ~1 | prjId
        (Intercept) Residual
StdDev:    1.871528  1.12779

Fixed effects:  log(commits + 1) ~ 1
                Value Std.Error   DF  t-value p-value
(Intercept) 2.897081 0.1045474 2345 27.71069       0

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-4.94874876 -0.43089615  0.04717525  0.49685034  4.12371376

Number of Observations: 2680
Number of Groups: 335
```

Estimate of fixed effects: The initial status of commits at quarter of 1 is 2.90 at 0.01 level of significance

Composite model is log(commits+1) = 2.9 + e

```
> VarCorr(model.a)
prjId = pdLogChol(1)
            Variance StdDev
(Intercept) 3.502616 1.871528
Residual    1.271910 1.127790
> mm=VarCorr(model.a)
> ICC<-as.numeric(mm[1,1])/(as.numeric(mm[2,1])+as.numeric(mm[1,1]))
> ICC
[1] 0.733605
```

Variance components:

Level 1 (within person variance) gets the estimate of 1.27

Level 2 (between person variance) receives the estimate of 3.50

Intra class correlation:

Quantifying proportion of outcome variation “explained”

ICC = 3.5 /(3.5+1.27) =0.73 →

73% variation in commits is attributable to differences between projects

5.2 Model B– The unconditional growth model

```
> #Model B
> model.b <- lme(log(commits+1) ~Time  , data=pd, random= ~ Time | prjId, method="ML")
> summary(model.b)
Linear mixed-effects model fit by maximum likelihood
  Data: pd
       AIC      BIC    logLik
  6916.325 6951.686 -3452.162

Random effects:
 Formula: ~Time | prjId
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev    Corr
(Intercept) 2.0737648 (Intr)
Time        0.2554526 -0.422
Residual    0.5782825

Fixed effects:  log(commits + 1) ~ Time
                Value  Std.Error   DF  t-value p-value
(Intercept) 1.5396481 0.11598892 2344 13.27410       0
Time        0.3016518 0.01478935 2344 20.39655       0
 Correlation:
     (Intr)
Time -0.452

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-5.79637113 -0.25711825  0.03675543  0.33738294  4.83370525

Number of Observations: 2680
Number of Groups: 335
```

Interpret the fixed effects based on the composite model:

Level1: log(commits+1)= a + b Time+ j

Level 2: a = 1.54 + y_0i

b = 0.30 + y_1i

log(commits+1) = 1.54 + 0.3Time + e (With composite residual: e = y_0i+y_1i*Time+ j)

(1) The estimated initial log(commits+1) is 1.54 (p=0 < 0.01)

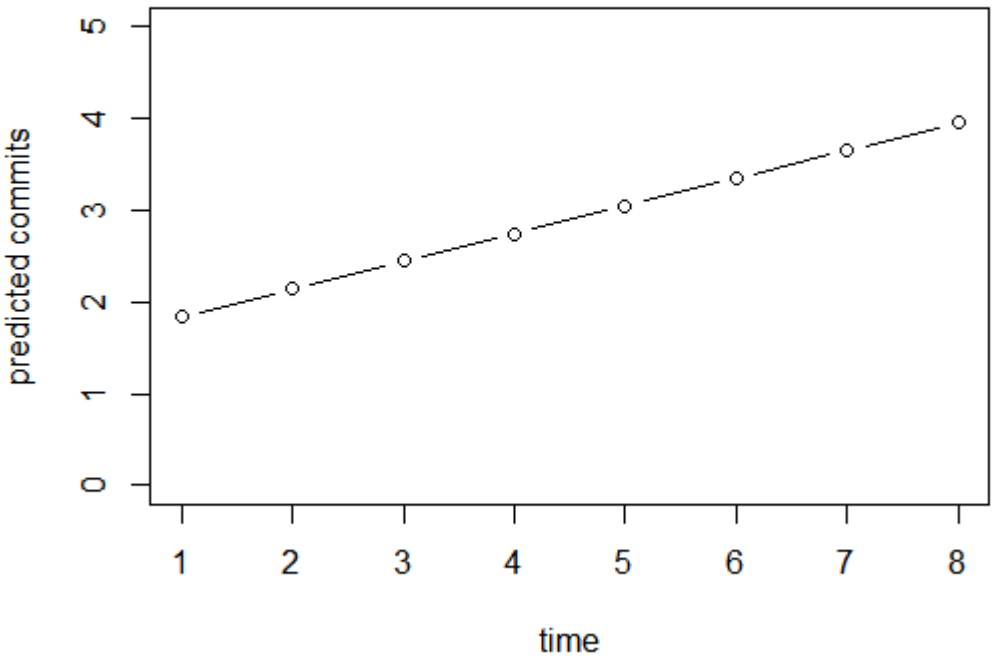
(2) the estimated rate of change at the quarter1 is 0.3 (p =0<0 .01)

```
> mm1=VarCorr(model.b)
> print(mm1)
prjId = pdLogChol(Time)
            Variance   StdDev    Corr
(Intercept) 4.30050033 2.0737648 (Intr)
Time        0.06525603 0.2554526 -0.422
Residual    0.33441067 0.5782825
```

Variance components:

Level 1 (within person variance) gets the estimate of 0.334

Level 2 (between person variance) receives the estimate of 4.3 for the initial status and 0.0 for the rate 0.07 of change.



In the unconditional growth model, the predicted log (commit + 1) increases with quarter.

5.3 Model C

```
> #Model C
> model.c <- lme(log(commits+1) ~OwnerType*Time , data=pd, random= ~ Time | prjId, method="ML")
> summary(model.c)
Linear mixed-effects model fit by maximum likelihood
  Data: pd
       AIC      BIC    logLik
  6848.537 6895.685 -3416.268

Random effects:
 Formula: ~Time | prjId
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev    Corr
(Intercept) 1.9177485 (Intr)
Time        0.2552184 -0.475
Residual    0.5782825

Fixed effects:  log(commits + 1) ~ OwnerType * Time
                      Value  Std.Error   DF   t-value p-value
(Intercept)        2.2332324 0.14340152 2343 15.573283  0.0000
OwnerTypeUSR      -1.5914435 0.21721996  333 -7.326415  0.0000
Time               0.3112637 0.01968103 2343 15.815412  0.0000
OwnerTypeUSR:Time -0.0220547 0.02981219 2343 -0.739788  0.4595
 Correlation:
                  (Intr) OwTUSR Time
OwnerTypeUSR      -0.660
Time              -0.503  0.332
OwnerTypeUSR:Time  0.332 -0.503 -0.660

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-5.80366335 -0.25037276  0.03449057  0.34008185  4.81170509

Number of Observations: 2680
Number of Groups: 335
```

Level1: log(commits+1)= a + b*Time + j

Level2: a = 2.23 -1.59OwerType + y_0i

b = 0.31 - 0.02OwerType + y_1i

The composite model:

Log(commits+1) = 2.23 -1.59OwerType +0.31Time - 0.02OwerType*Time+ y_0i + y_1iTime + j

=2.23 -1.59OwerType +0.31Time - 0.02OwerType*Time+e

(1) The estimated initial log(commits+1) for the average projects owned by organization is 2.23 (p=0.00< 0.001)

(2) the estimated differential in initial log(commits+1) between projects owned by organization and user is 1.59 (p=0.00 < .001)

(3) the estimated rate of change in Log(commits+1) for an average projects owned by organization is 0.31 (p=0.00 < .001)

(4) the estimated differential in the rate of change in Log(commits+1) between projects owned by organization and user is indistinguishable from 0 (p_value=0.46>0.1)

```
> mm2=VarCorr(model.c)
> print(mm2)
prjId = pdLogChol(Time)
            Variance   StdDev    Corr
(Intercept) 3.67775925 1.9177485 (Intr)
Time        0.06513643 0.2552184 -0.475
Residual    0.33441067 0.5782825
> #Pseudo R2
> r2<- (as.numeric(mm1[1,1])-as.numeric(mm2[1,1]))/as.numeric(mm1[1,1])
> print(r2)
[1] 0.1448067
```

Variance components:

Level 1 (within person variance) gets the estimate of 0.33

Level 2 (between person variance) receives the estimate of 3.68 for the initial status and 0.07 for the rate of change

Pseudo R2 (estimate the percentage of between projects variation in commits associated with predictors) = 4.3-3.68/4.3= 0.145

14.5% of the between-person variability in commits is associated with linear time.

5.4 Model D

```
> model.d <- lme(log(commits+1) ~ log(CmtCmnt+1)*Time+OwnerType*Time , data=pd, random= ~ Time
 prjId, method="ML",control = lmeControl(opt = "optim"))
> summary(model.d)
Linear mixed-effects model fit by maximum likelihood
  Data: pd
       AIC      BIC  logLik
  6795.801 6854.736 -3387.9

Random effects:
 Formula: ~Time | prjId
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev    Corr
(Intercept) 1.8513576 (Intr)
Time        0.2544868 -0.477
Residual    0.5739476

Fixed effects:  log(commits + 1) ~ log(CmtCmnt + 1) * Time + OwnerType * Time
                          Value  Std.Error   DF   t-value p-value
(Intercept)           2.1549223 0.13934944 2341 15.464163  0.0000
log(CmtCmnt + 1)      0.6920427 0.09132652 2341  7.577675  0.0000
Time                  0.3157220 0.02016451 2341 15.657313  0.0000
OwnerTypeUSR         -1.5301476 0.21034854  333 -7.274344  0.0000
log(CmtCmnt + 1):Time -0.0833973 0.01439789 2341 -5.792327  0.0000
Time:OwnerTypeUSR    -0.0254973 0.02993661 2341 -0.851708  0.3945
 Correlation:
                      (Intr) lg(CC+1) Time   OwTUSR l(CC+1):
log(CmtCmnt + 1)      -0.079
Time                  -0.507  0.048
OwnerTypeUSR          -0.661  0.041    0.334
log(CmtCmnt + 1):Time  0.097 -0.812   -0.170 -0.049
Time:OwnerTypeUSR      0.339 -0.025   -0.665 -0.507  0.088

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-5.83506138 -0.25501014  0.02810498  0.33982217  4.84644758

Number of Observations: 2680
```

Interpret the fixed effects based on the composite model:

Level1: log(commits+1)= a + b Time+ j

Level2: a = 2.15 -1.53OwerType+0.69*log(CmtCmnt+1) + y_0i

b = 0.32 -0.03OwerType-0.08log(CmtCmnt+1) + y_1i

The composite model:

Log(commits+1)=2.15-1.53OwerType+0.69*log(CmtCmnt+1)+0.32Time- - -0.03OwerType*Time-0.08log(CmtCmnt+1) *Time+ y_0i + Time*y_1i+j

=2.15-1.53OwerType+0.69*log(CmtCmnt+1)+0.32Time-0.03OwerType*Time-0.08log(CmtCmnt+1) *Time +e

(1) The estimated initial Log(commits+1) for the average projects owned by organization is 2.15 (p p=0.00<0.001)

(2) the estimated differential in initial Log(commits+1) between projects owned by organization and user, controlling for CmtCmnt is 1.53 (p=0.00 < 0.001)

(3) the estimated differential in initial Log(commits+1) for 1 unit difference in log(CmtCmnt+1) controlling for OwnerType at the initial stage is 0.69 (p=0.00<0.001)

(4) the estimated rate of change in Log(commits+1) for the average projects owned by organization is 0.03 (p=0.00 < 0.001)

(5) the estimated differential in the rate of change in Log(commits+1) between projects owned by organization and user is indistinguishable from 0 (p_value=0.00>0.1)

(6) the estimated differential in the rate of change in Log(commits+1) of log(CmtCmnt+1) is distinguishable from 0 (p=0.00<0.001).

5.4 Model E

```
> model.e <- lme(log(commits+1) ~ log(CmtCmnt+1)*Time+OwnerType , data=pd, random= ~ Time | prjI
d, method="ML",control = lmeControl(opt = "optim"))
> summary(model.e)
Linear mixed-effects model fit by maximum likelihood
  Data: pd
       AIC      BIC    logLik
  6794.527 6847.569 -3388.263

Random effects:
 Formula: ~Time | prjId
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev    Corr
(Intercept) 1.8521560 (Intr)
Time        0.2545353 -0.479
Residual    0.5740656

Fixed effects:  log(commits + 1) ~ log(CmtCmnt + 1) * Time + OwnerType
                          Value  Std.Error   DF   t-value p-value
(Intercept)            2.1952939 0.13107878 2342 16.747896       0
log(CmtCmnt + 1)       0.6901799 0.09130558 2342  7.559011       0
Time                   0.3042733 0.01505425 2342 20.211783       0
OwnerTypeUSR          -1.6211491 0.18121028  333 -8.946231       0
log(CmtCmnt + 1):Time -0.0822599 0.01434268 2342 -5.735325       0
 Correlation:
                      (Intr) lg(CC+1) Time   OwTUSR
log(CmtCmnt + 1)      -0.075
Time                  -0.402  0.042
OwnerTypeUSR          -0.603  0.033   -0.006
log(CmtCmnt + 1):Time  0.071 -0.813   -0.150 -0.005

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-5.84142845 -0.25096081  0.02812843  0.33650833  4.85255949

Number of Observations: 2680
Number of Groups: 335
```

Model D include CmtCmnt as a predictor of both initial status and change. But

OwnerType as a predictor of only initial status. So, in Model E we remove OwnerType as the predictor of change.

Interpret the fixed effects based on the composite model:

Level1: log(commits+1)= a + b Time+ j

Level2: a = 2.2 -1.62OwerType+0.69*log(CmtCmnt+1) + y_0i

b = 0.3 -0.08log(CmtCmnt+1) + y_1i

The composite model:

Log(commits+1)=2.2-1.62OwerType+0.69*log(CmtCmnt+1)+0.3Time-0.08log(CmtCmnt+1) *Time+ y_0i + Time*y_1i+j

=2.2-1.62OwerType+0.69*log(CmtCmnt+1)+0.3Time-0.08log(CmtCmnt+1) *Time +e

(1) Controlling for the effects of CmtCmnt, the estimated differential in initial Log(commits+1) between projects owned by organization and user is 1.62($p=0.00<0.001$).

(2) Controlling for the effect of OwnerType, for each 1-point difference in log(CmtCmnt+1): the average initial commits is 0.69 higher and the average rate of change in Log(commits+1) is 0.08 lower.

We conclude that projects owned by organizations have more commits initially than projects owned by users but their rate of change in commits between quarter 1 and quarter 8 is no different. We also conclude that CmtCmnt is positively associated with early number of commits but negatively associated with the rate of change in the number of commits. Projects of quarter 1 whose commits have more comments tend to have more commits at that quarter, but they have a slower rate of increase in the number of commits over time.

6. Results and Discussions

6.1 Conclusion

In this project, we built five models to analyze the longitudinal data set provided by GitHub. In order to get the most suitable model, we compare their AIC and BIC and find that model E has the smallest AIC and BIC, and model e is the best model for us to analyze this dataset. Log(commits+1)=2.2-1.62OwerType+0.69*log(CmtCmnt+1)+0.3Time-0.08log(CmtCmnt+1) *Time +e

	Model A	Model B	Model C	Model D	Model E
AIC	9308.617	6916.325	6848.537	6795.801	6794.527
BIC	9326.296	6951.686	6895.685	6854.736	6847.569

In model e, we found that all P values were equal to 0.000, far less than the significance level of 0.05. Therefore, we can reject the null hypothesis and conclude that all independent variables have a statistically significant effect on the dependent variable as a whole.

The results show that CmtCmnt has an impact on the initial state and rate of change of OSS project commit. The more comments on commit, the more users commit to the project. Therefore, we accept H1: OSS projects with more comments on comments may have more comments

Furthermore, the type of project owner also has a significant impact on the number of user commits. The OSS projects owned by organizations encourage users to commit, but the owner type only affects the initial state. With the change of time, the influence of owners on commits is always the same. This is still consistent with our hypothesis H2: projects owned by organization have more commitments than other users, so H2 can be accepted.

Based on the above results, we can draw the following conclusions: trajectories of total number of coding activities different by: (1) total number of discussion / comments on

common; and (2) Type of the owner of the project (organization, user, etc.).

6.2 Limitation and Future Research

This project still has many limitations, such as transition fitting, small data set capacity, less variables selected and so on. In the future research, we need to make the model more in line with the actual situation. For software companies, what are the characteristics they value, and whether some characteristics or requirements of the enterprise will also affect their understanding of OSS projects. In addition, the update of technology is very fast. The data of two years in the software industry seems to be a relatively long period. Whether such a period is reasonable or not, we need to further study.

7. References

Crowston, K., Annabi, H., & Howison, J. (2003). Defining open source software project success. *ICIS 2003 Proceedings*, 28

Daniel, S., Agarwal, R., & Stewart, K. J. (2013). The effects of diversity in global, distributed collectives: a study of open source project success. *Information Systems Research*, *24*(2), 312+. https://link.gale.com/apps/doc/A334487402/ITOF?u=learn&sid=bookmark-ITOF&xid=0666b892

DeLone, W. H., & McLean, E. R. (1992). Information systems success: The quest for the dependent variable. *Information systems research, 3*(1), 60-95.

Guzman, E., Azócar, D., & Li, Y. (2014). Sentiment analysis of commit comments in GitHub: an empirical study. *In Proceedings of the 11th working conference on mining software repositories* (pp. 352-355).

Vaughan-Nichols, S. J. (2015). *It's an open-source world: 78 percent of companies run open-source software.* ZDNet. https://www.zdnet.com/article/its-an-open-source-world-78-percent-of-companies-run-open-source-software/