```
## Loading required package:  ggplot2
## Warning:  package 'ggplot2' was built under R version 3.1.2
## Loading required package:  reshape2
## Warning:  package 'reshape2' was built under R version 3.1.2
## Loading required package:  ROCR
## Warning:  package 'ROCR' was built under R version 3.1.2
## Loading required package:  gplots
## Warning:  package 'gplots' was built under R version 3.1.2
## KernSmooth 2.23 loaded
## Copyright M. P. Wand 1997-2009
##
## Attaching package:  'gplots'
##
## Nastpujcy obiekt zosta zakryty from 'package:stats':
##
##    lowess
##
## Loading required package:  xtable
## Warning:  package 'xtable' was built under R version 3.1.2
```

Github games - data analysis

WikiTeams.pl

11 January 2015

```
options("warn" = -1)
```

1 Read in the data

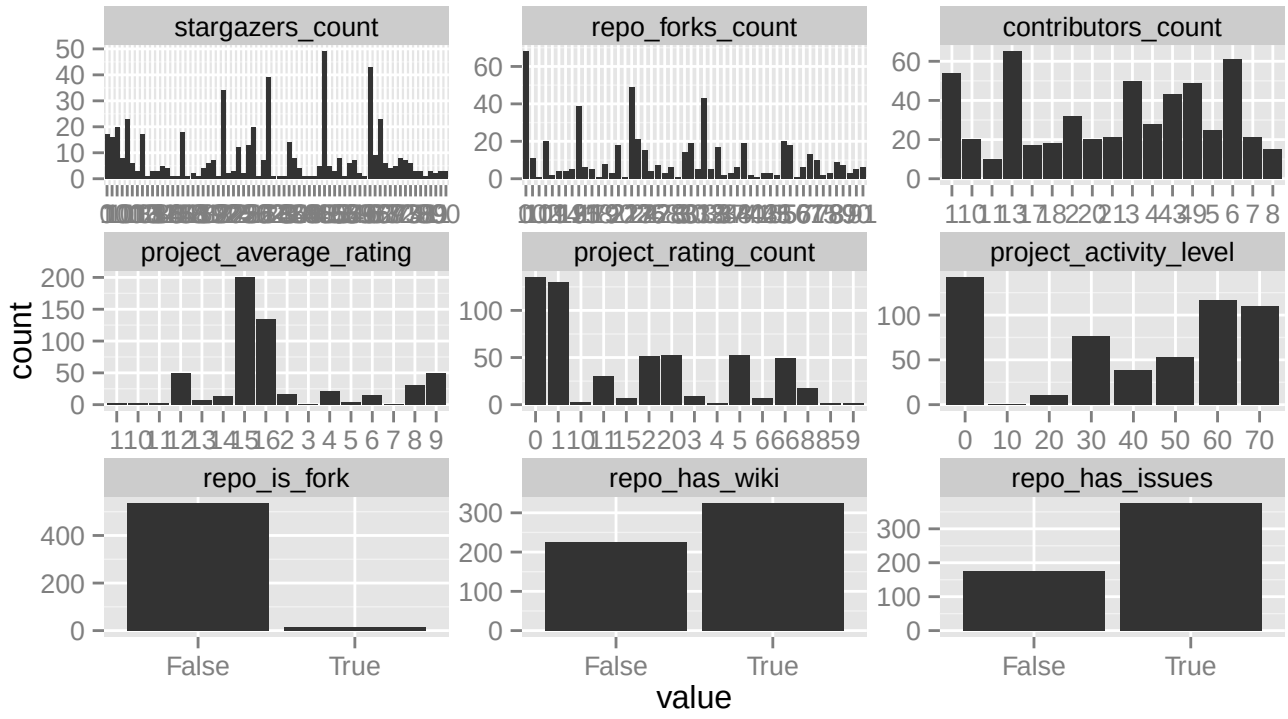
```
D <- read.table("../results_game.csv", sep=";", quote = "\"", header=T)
names(D)
```

```
##  [1] "ordinal_id"                    "github_repo_id"
##  [3] "repo_full_name"                "repo_html_url"
##  [5] "repo_forks_count"              "stargazers_count"
##  [7] "contributors_count"            "repo_created_at"
##  [9] "repo_is_fork"                  "repo_has_issues"
## [11] "repo_open_issues_count"        "repo_has_wiki"
## [13] "repo_network_count"            "repo_pushed_at"
## [15] "repo_size"                     "repo_updated_at"
## [17] "repo_watchers_count"           "project_id"
## [19] "project_name"                  "project_url"
## [21] "project_htmlurl"               "project_created_at"
## [23] "project_updated_at"            "project_homepage_url"
## [25] "project_average_rating"        "project_rating_count"
## [27] "project_review_count"          "project_activity_level"
## [29] "project_user_count"            "twelve_month_contributor_count"
## [31] "total_contributor_count"       "twelve_month_commit_count"
## [33] "total_commit_count"            "total_code_lines"
## [35] "main_language_name"            "developer_works_during_bd"
## [37] "developer_works_period"        "developer_all_pushes"
## [39] "developer_all_stars_given"     "developer_all_creations"
## [41] "developer_all_issues_created"  "developer_all_pull_requests"
```

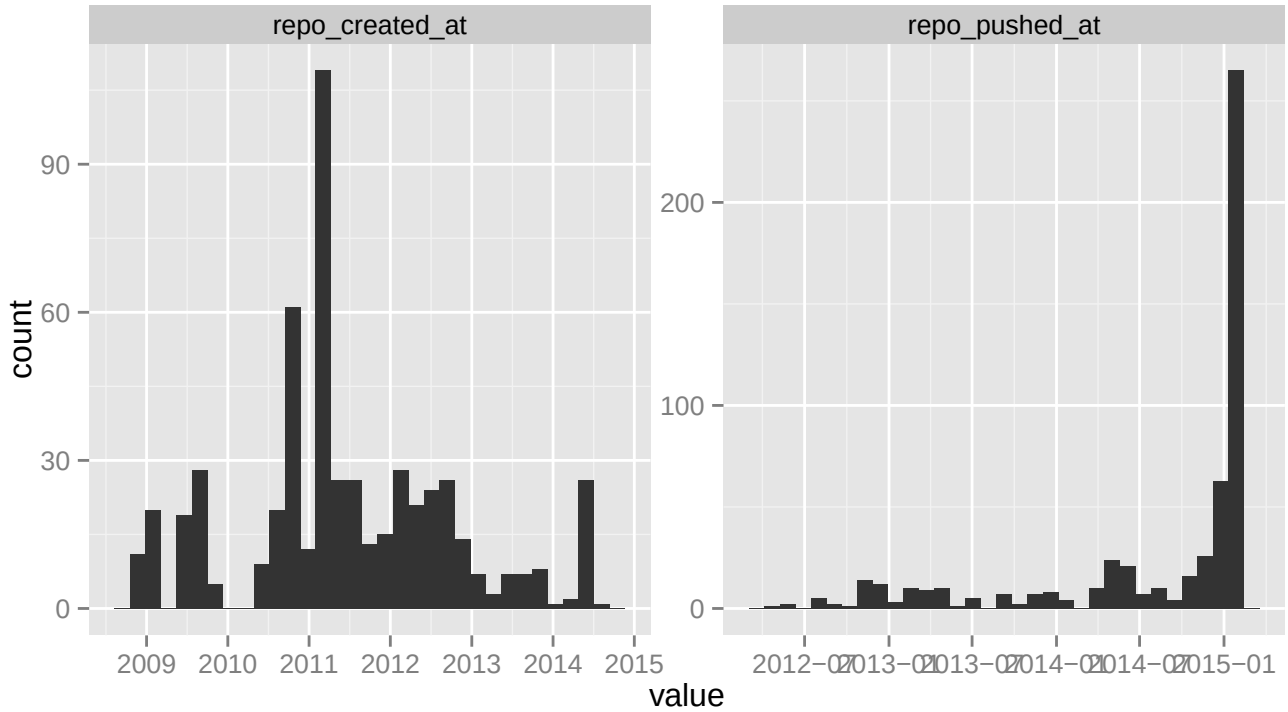
```
D$repo_created_at <- as.Date(D$repo_created_at)
D$repo_pushed_at <- as.Date(D$repo_pushed_at)
# convert some factors to numeric for easier computations
D$project_average_rating <- as.numeric(D$project_average_rating)
D$project_rating_count <- as.numeric(D$project_rating_count)
D$project_activity_level <- as.numeric(D$project_activity_level)
#D£repository_has_downloads <- as.numeric(D£repository_has_downloads)
```

Read 549 recods.

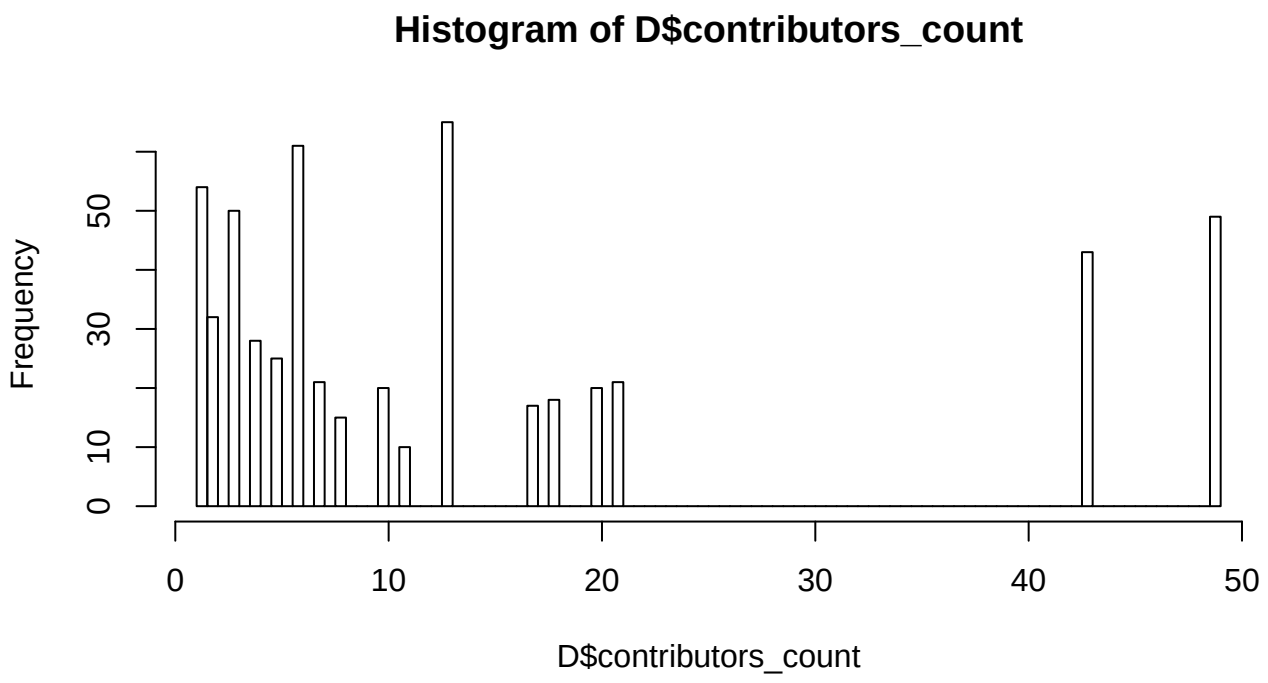
```
# discrete
plot_mhist(D, attrs=c("stargazers_count", "repo_forks_count", "contributors_count", "project_average_rating",
```



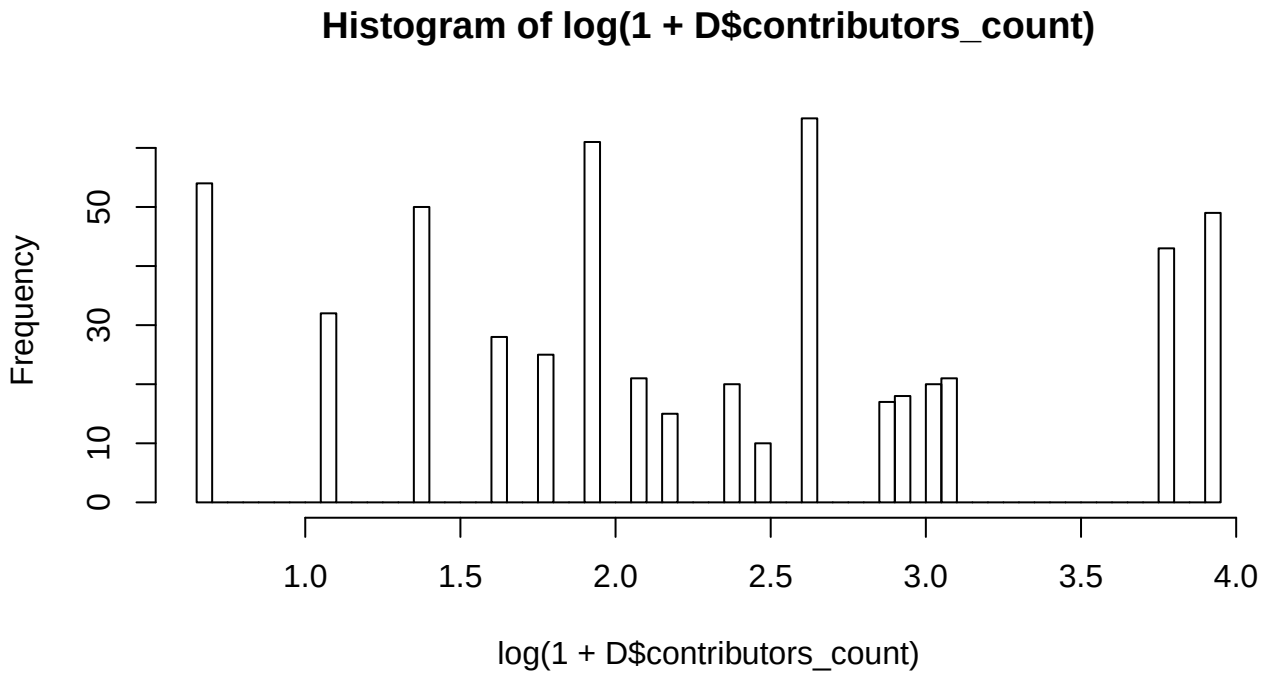

```
# continuous
plot_mhist(D, attrs=c("repo_created_at", "repo_pushed_at"), date.values = T)

## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
```



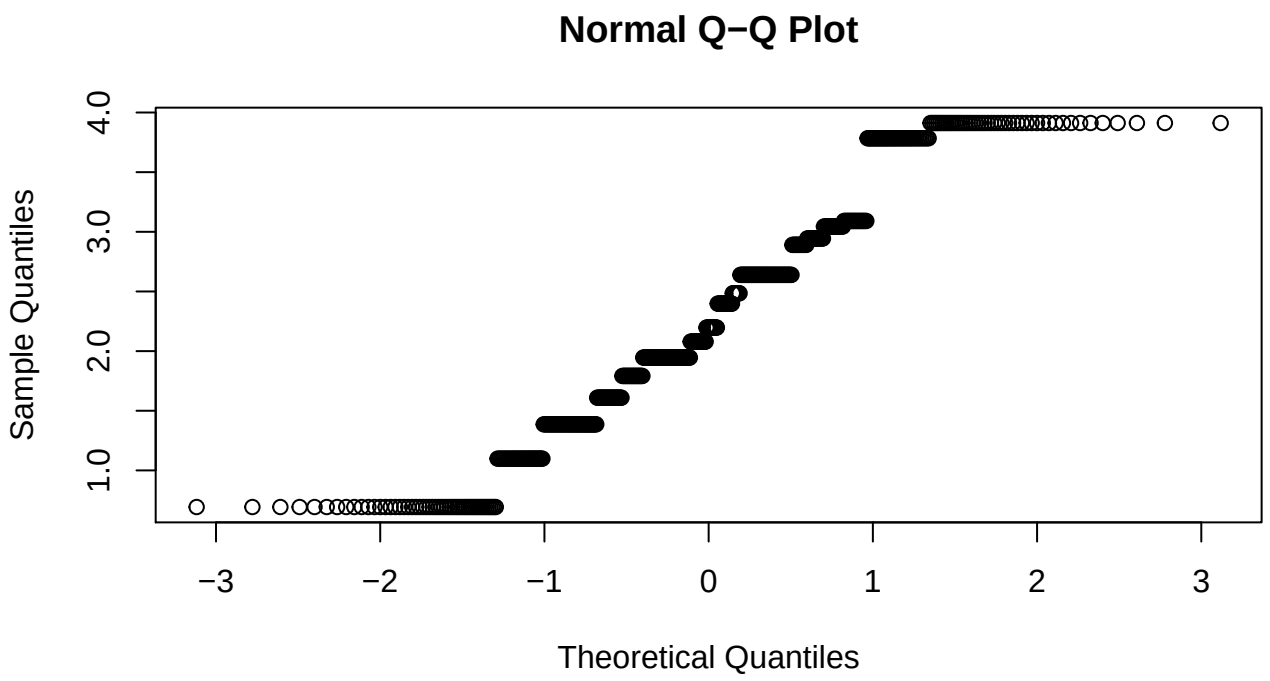

```
# contrib count
hist(D$contributors_count, breaks=100)
```



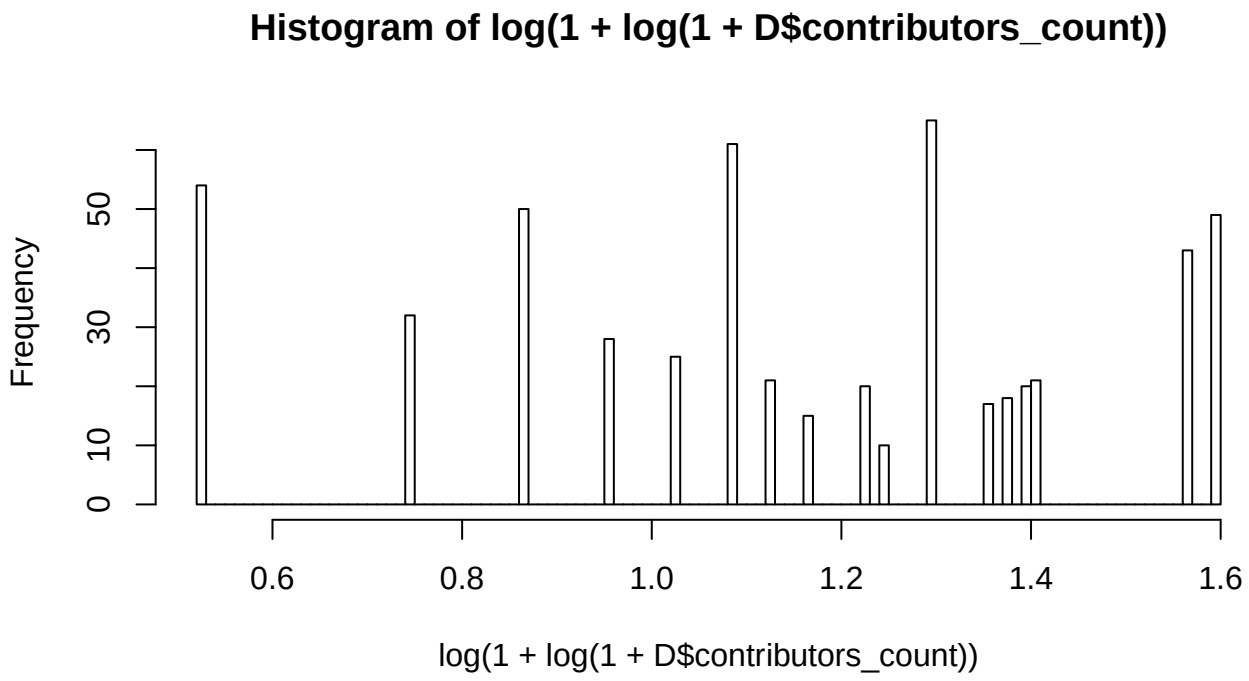

```
summary(D$contributors_count, breaks=100)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    4.00    8.00   14.57   18.00   49.00

hist(log(1+D$contributors_count), breaks=100)
```



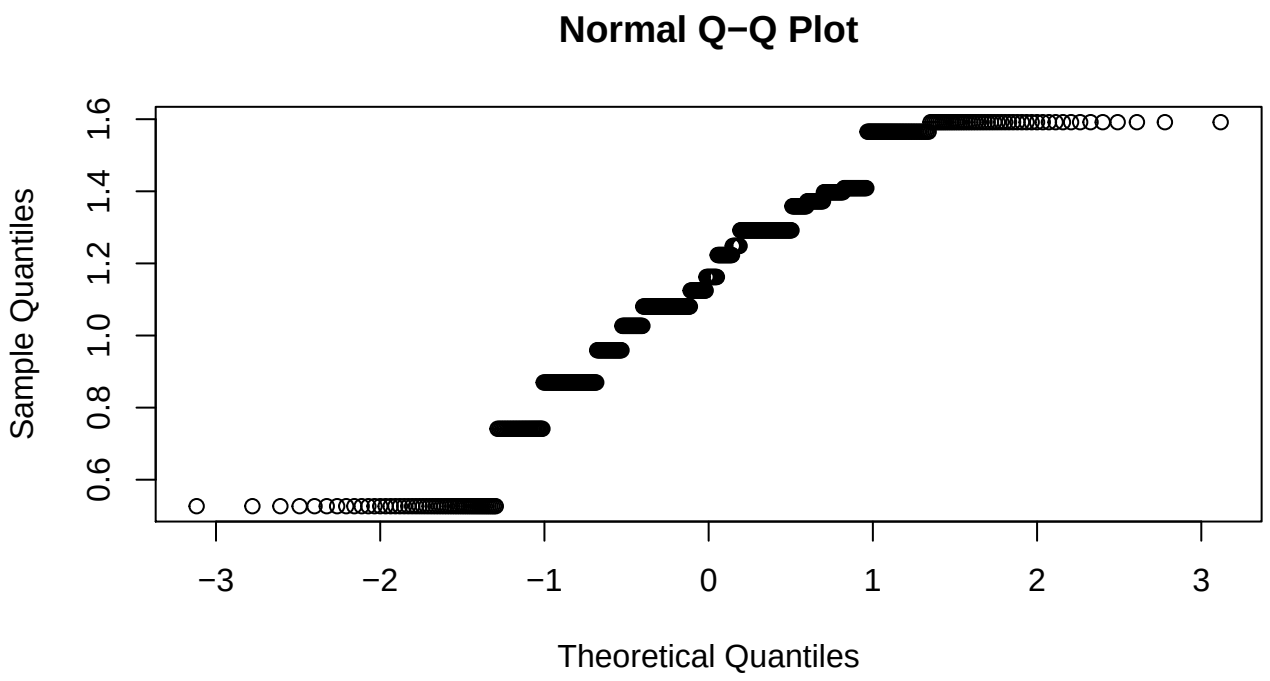
```
qqnorm(log(1+D$contributors_count))
```



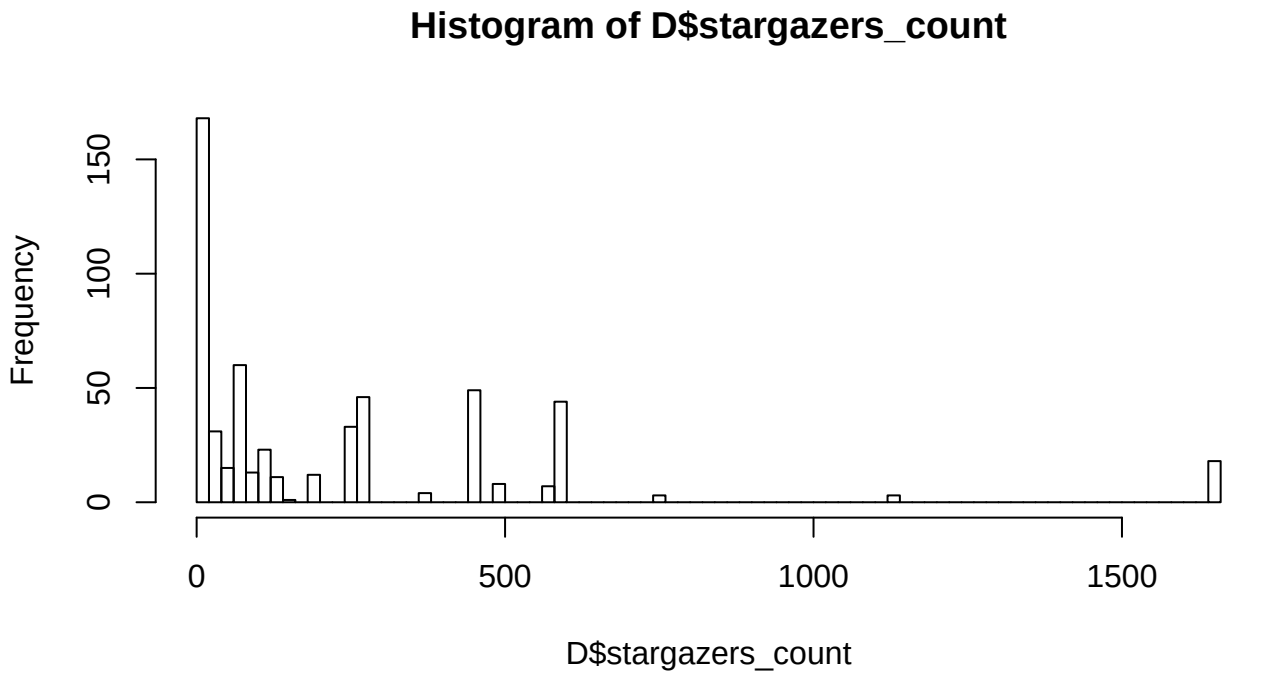

```
hist(log(1+log(1+D$contributors_count)), breaks=100)
```



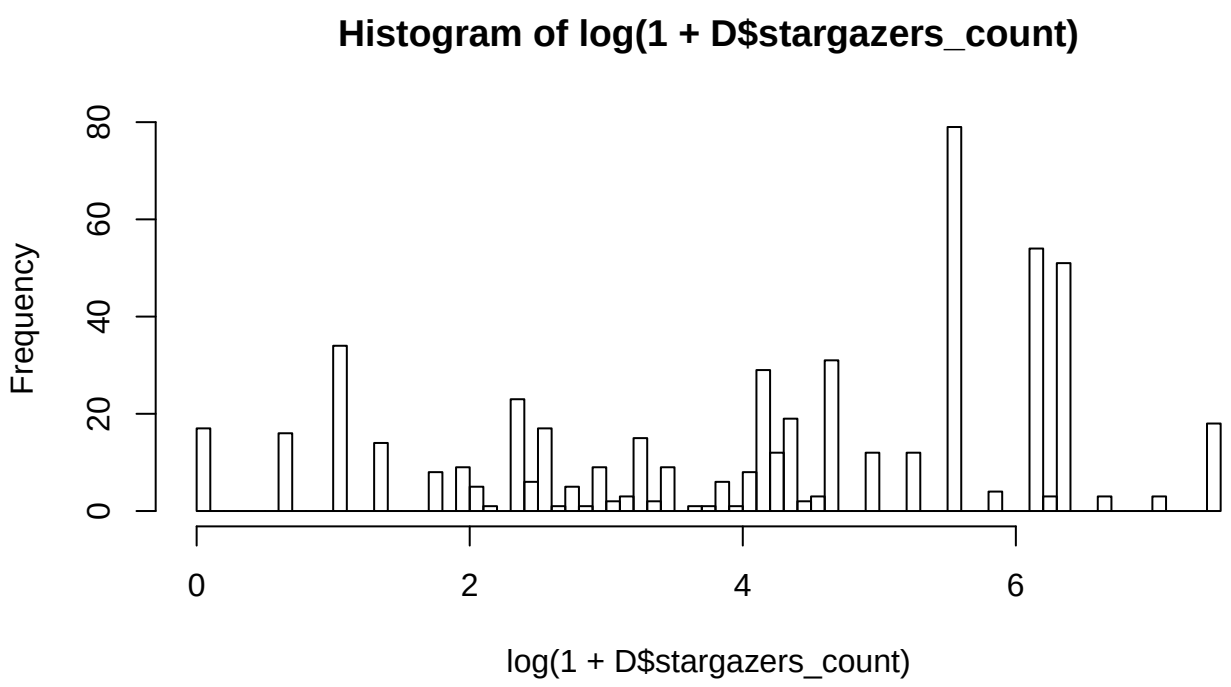

```
qqnorm(log(1+log(1+D$contributors_count)))
```



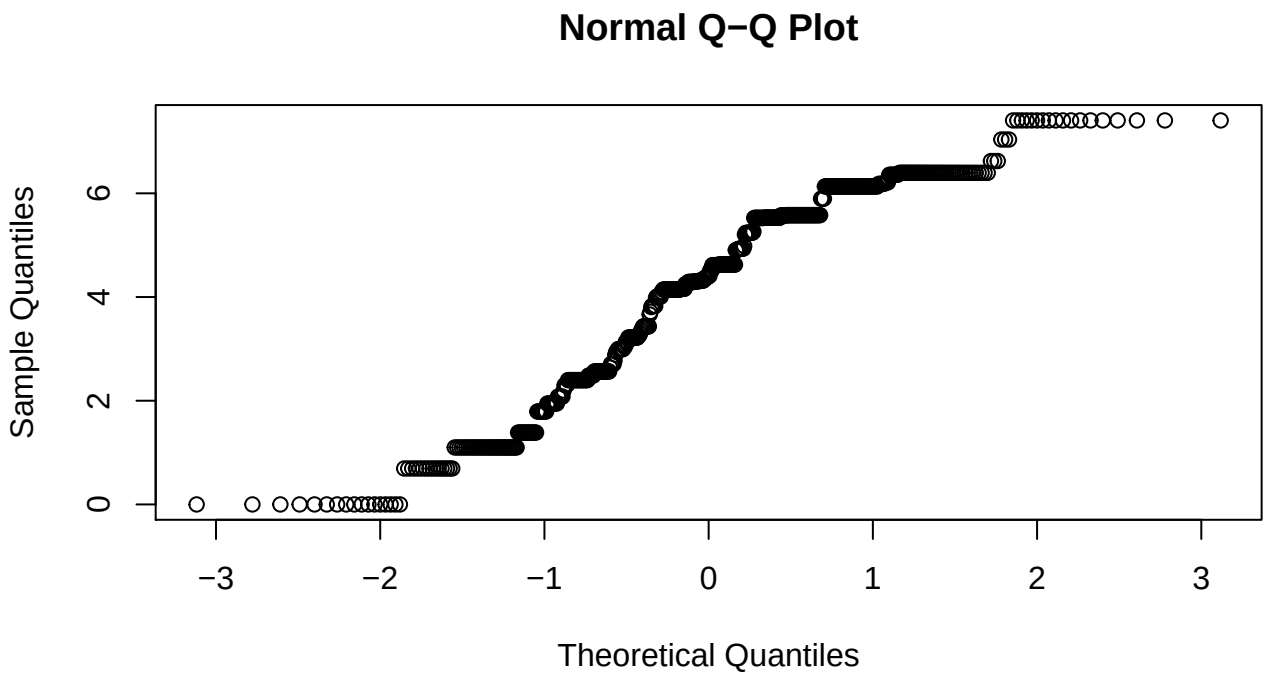

```
summary(log(1+D$contributors_count), breaks=100)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.6931  1.6090  2.1970  2.2840  2.9440  3.9120

# stargazers count
hist(D$stargazers_count, breaks=100)
```



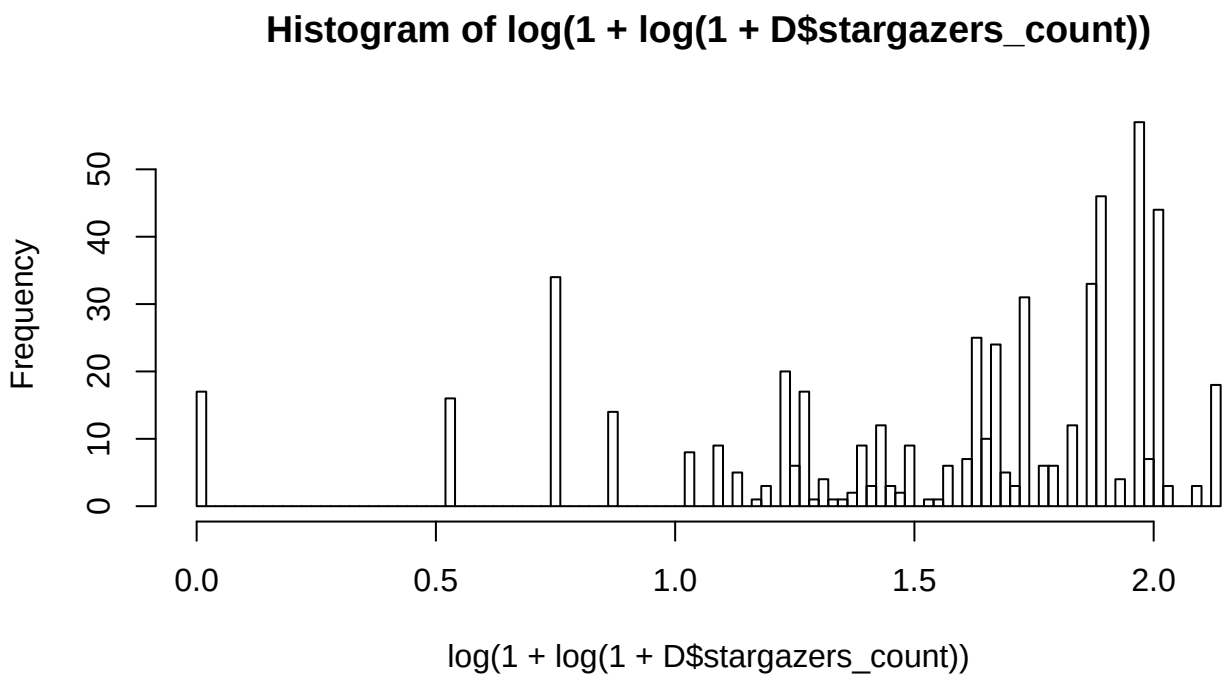
```
hist(log(1+D$stargazers_count), breaks=100)
```



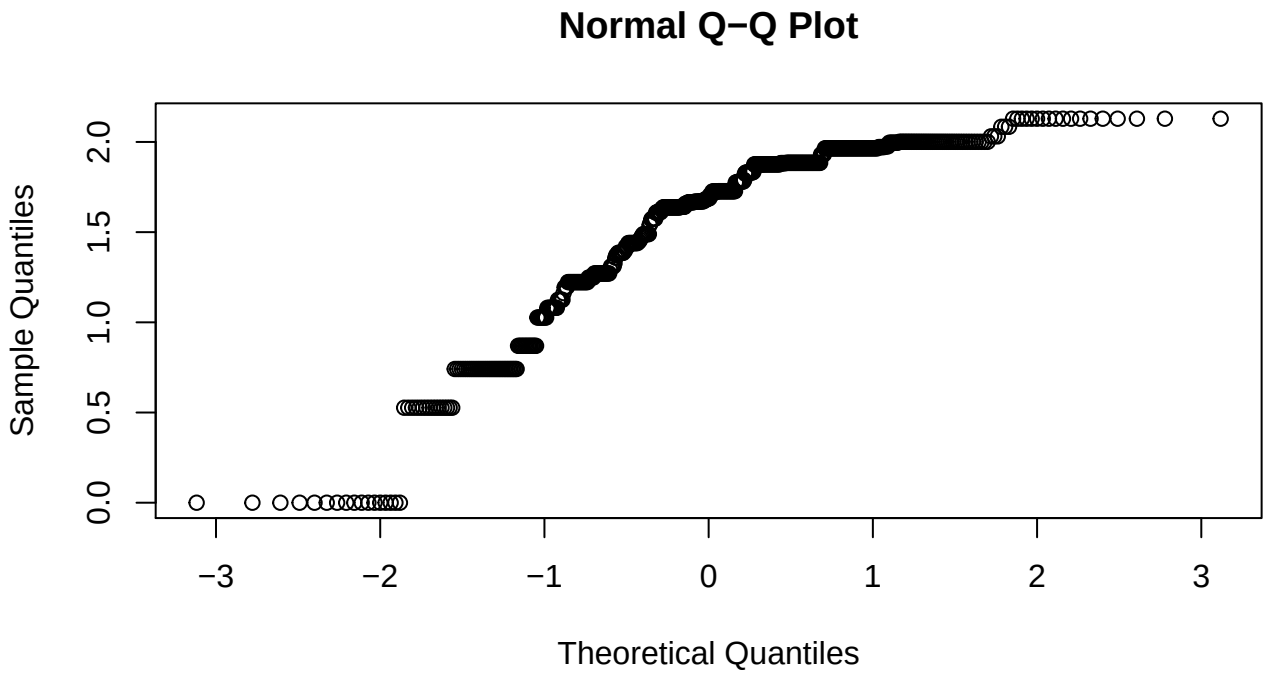

```
qqnorm(log(1+D$stargazers_count))
```



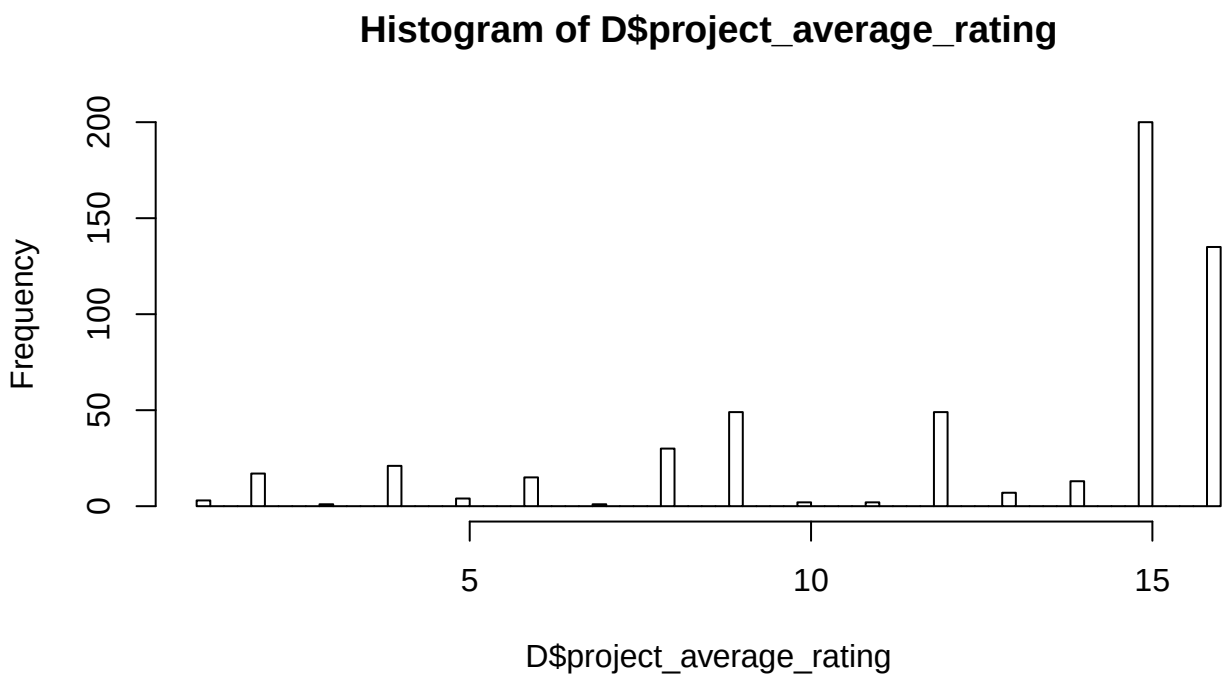

```
hist(log(1+log(1+D$stargazers_count)), breaks=100)
```



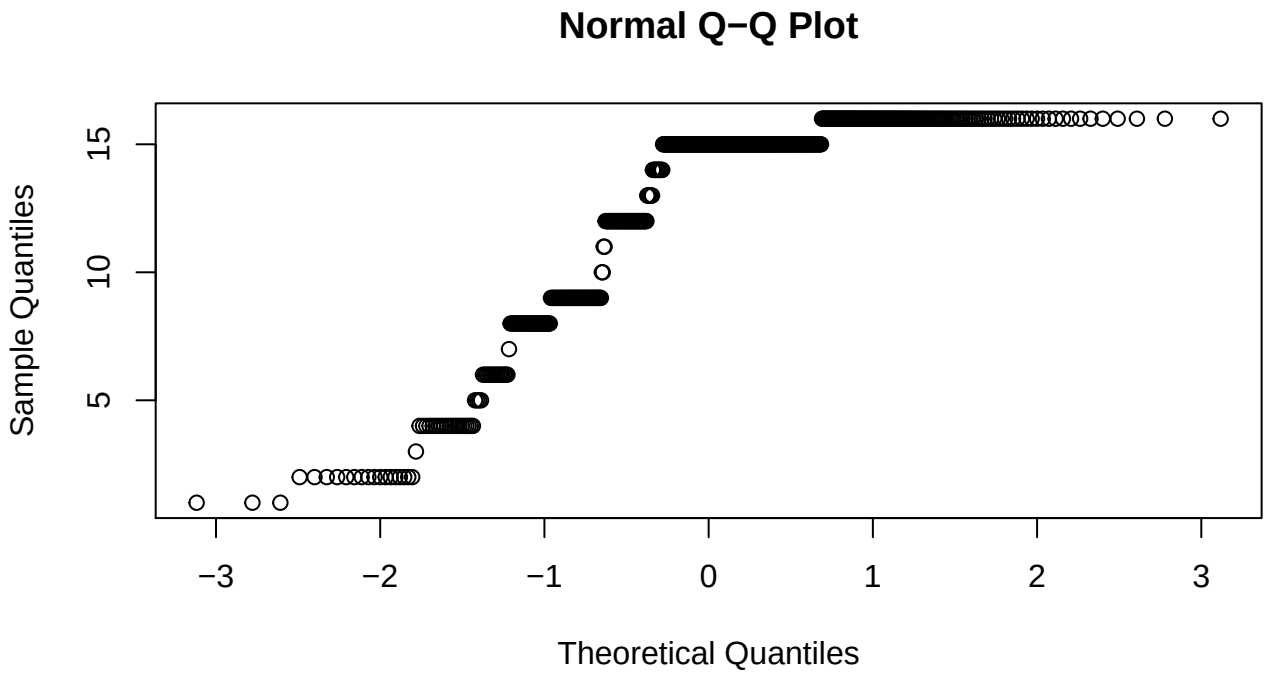

```
qqnorm(log(1+log(1+D$stargazers_count)))
```



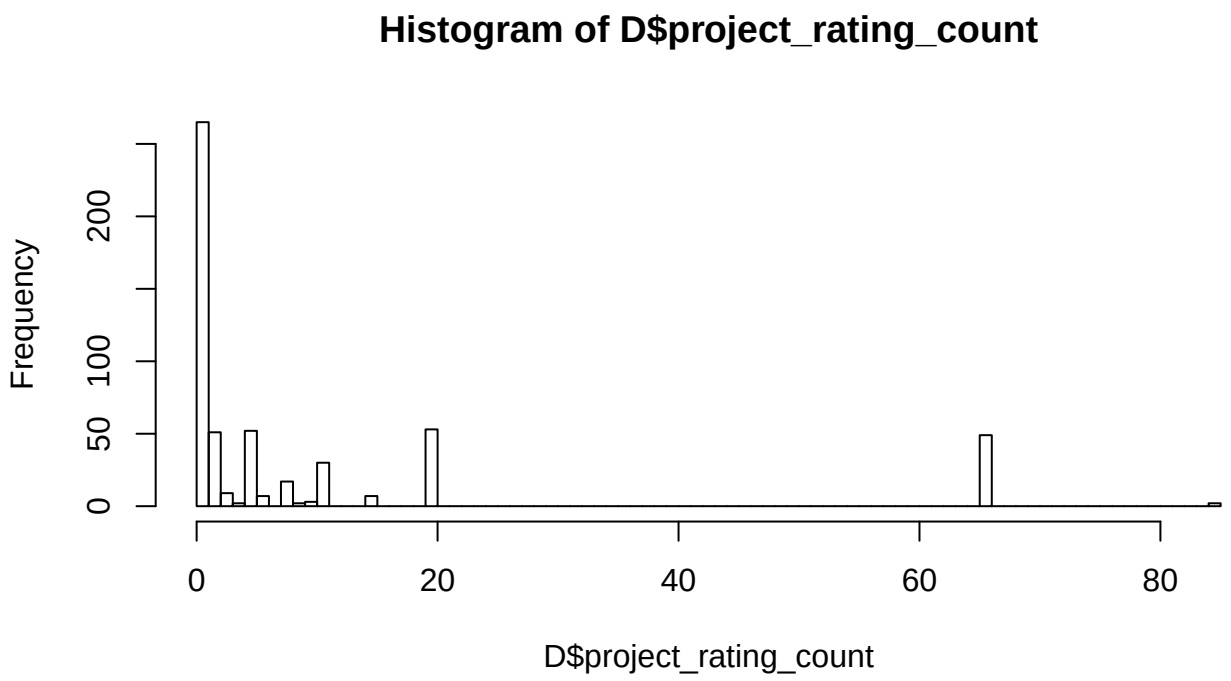

```
summary(D$stargazers_count)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0    12.0    81.0   233.4   264.0  1645.0

# openhub rating
hist(D$project_average_rating, breaks=100)
```



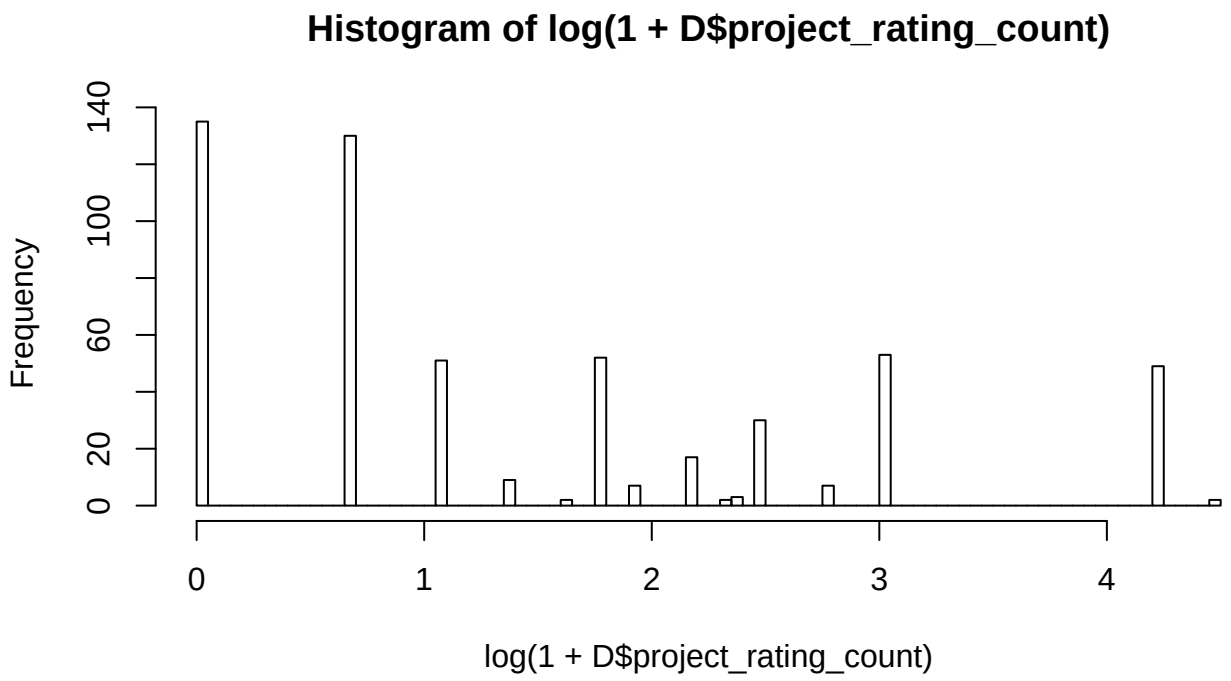

```
qqnorm(D$project_average_rating)
```



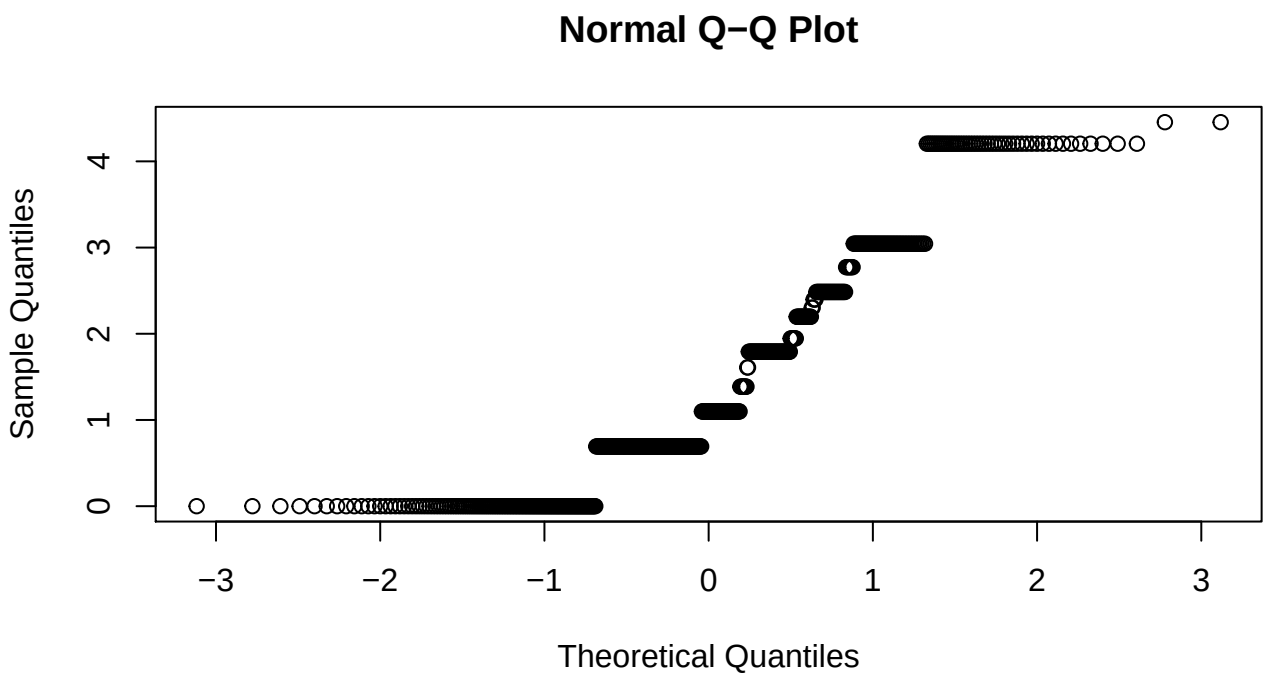

```
summary(D$project_average_rating)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    9.00   15.00   12.72   15.00   16.00

# openhub rating count
hist(D$project_rating_count, breaks=100)
```



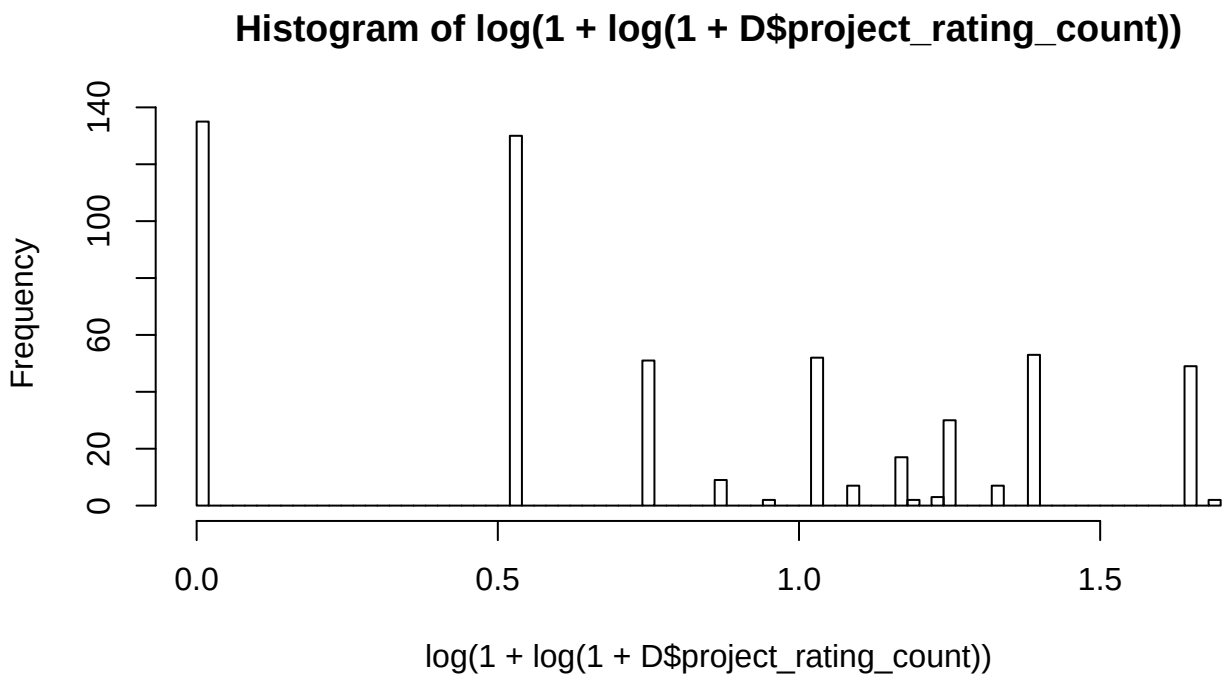

```
hist(log(1+D$project_rating_count), breaks=100)
```



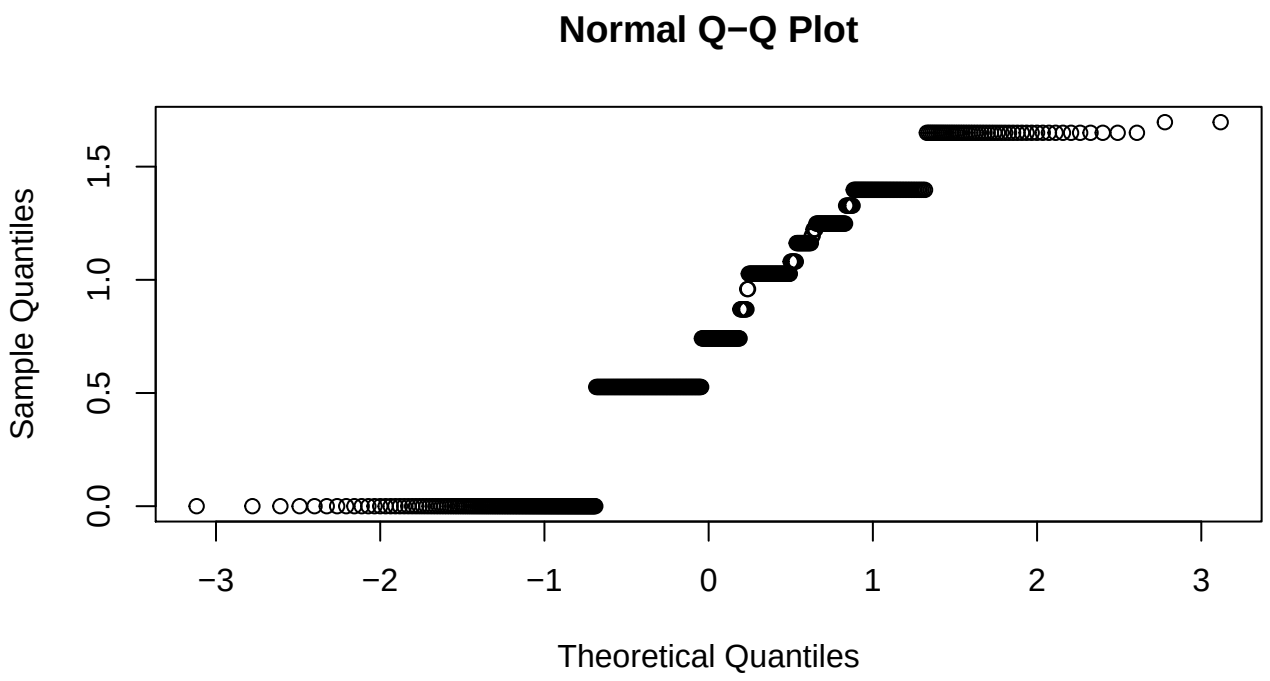

```
qqnorm(log(1+D$project_rating_count))
```



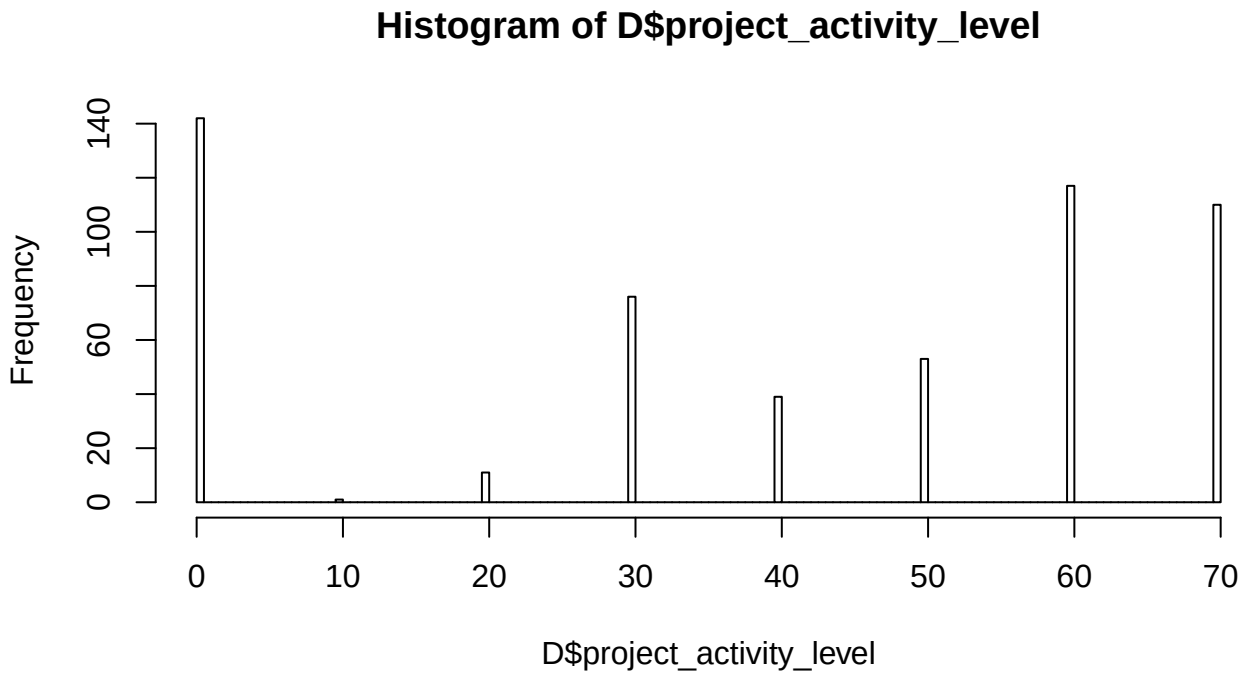

```
hist(log(1+log(1+D$project_rating_count)), breaks=100)
```



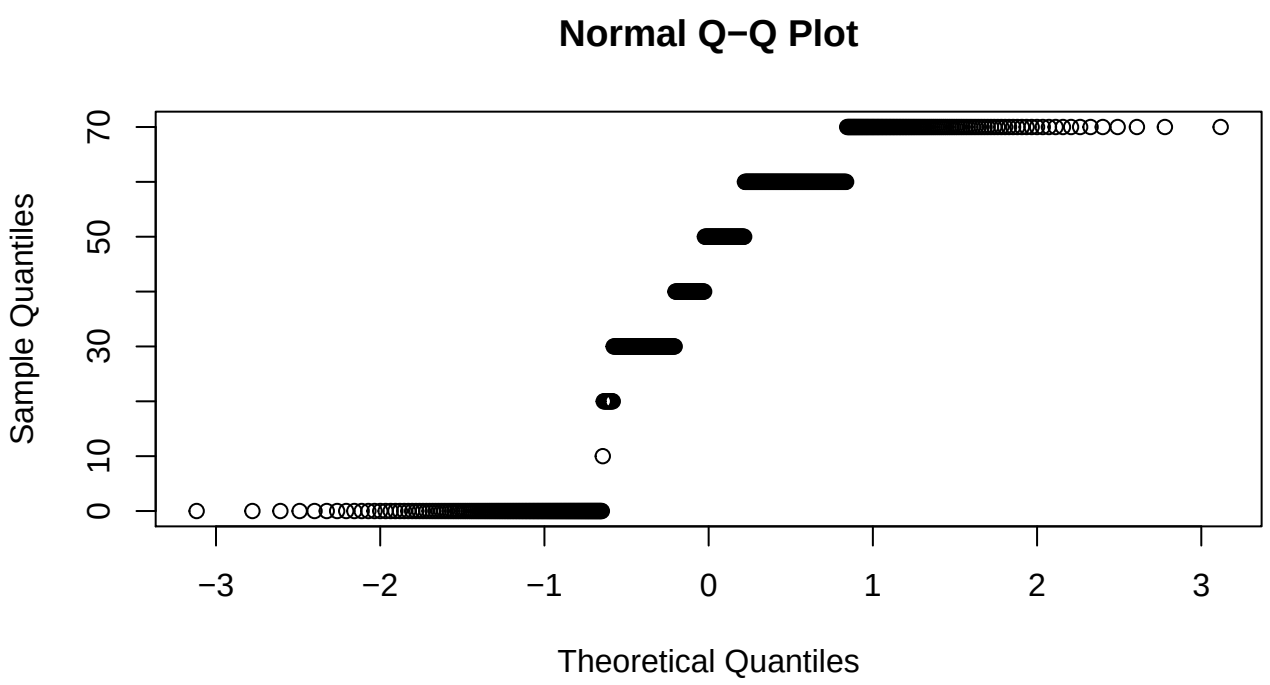

```
qqnorm(log(1+log(1+D$project_rating_count)))
```




```
summary(D$project_rating_count)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     1.0     2.0    10.3    11.0    85.0

# openhub activity level
hist(D$project_activity_level, breaks=100)
```



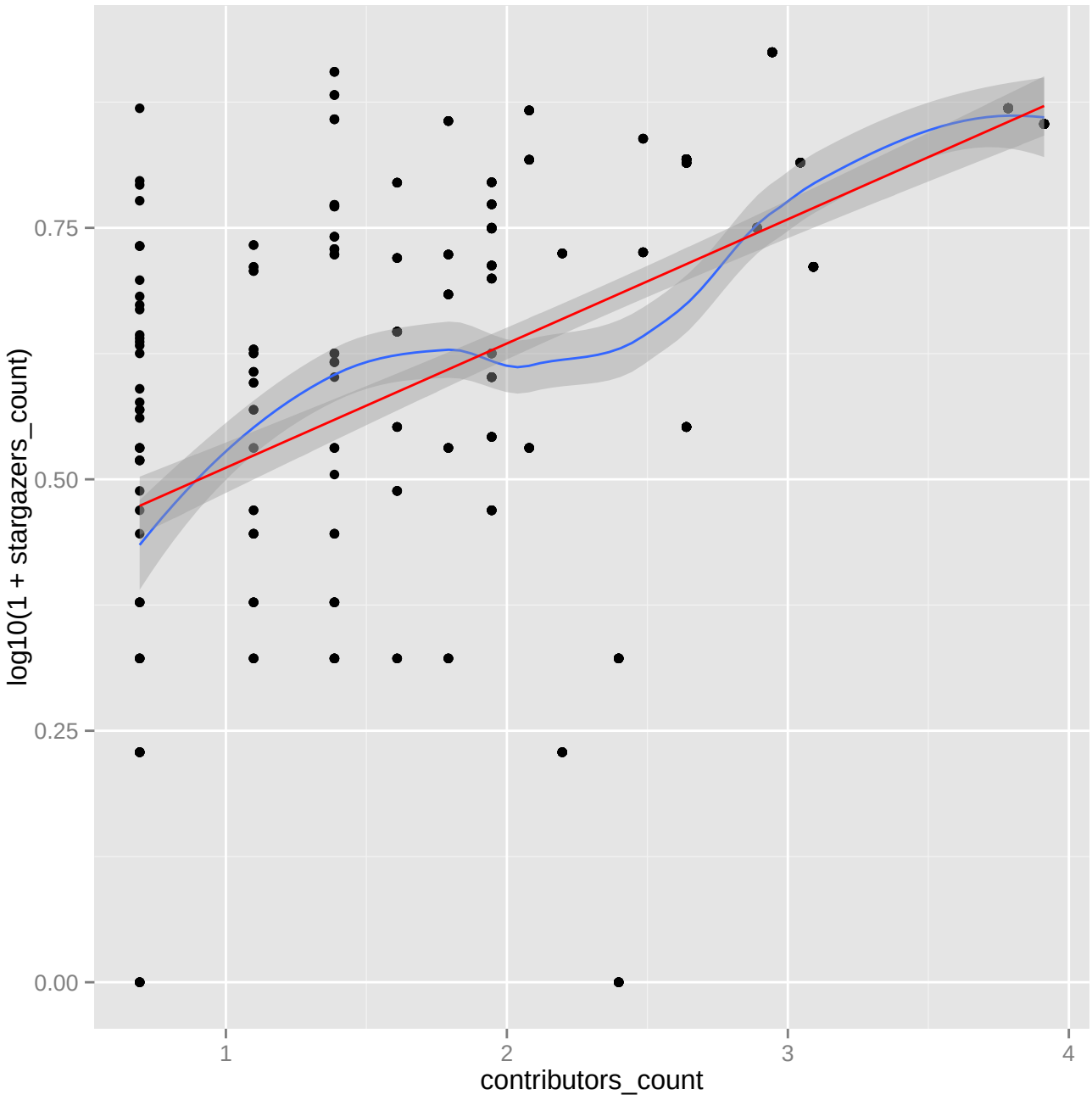
```
qqnorm(D$project_activity_level)
```




```
summary(D$project_activity_level)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00   50.00   39.05   60.00   70.00
```

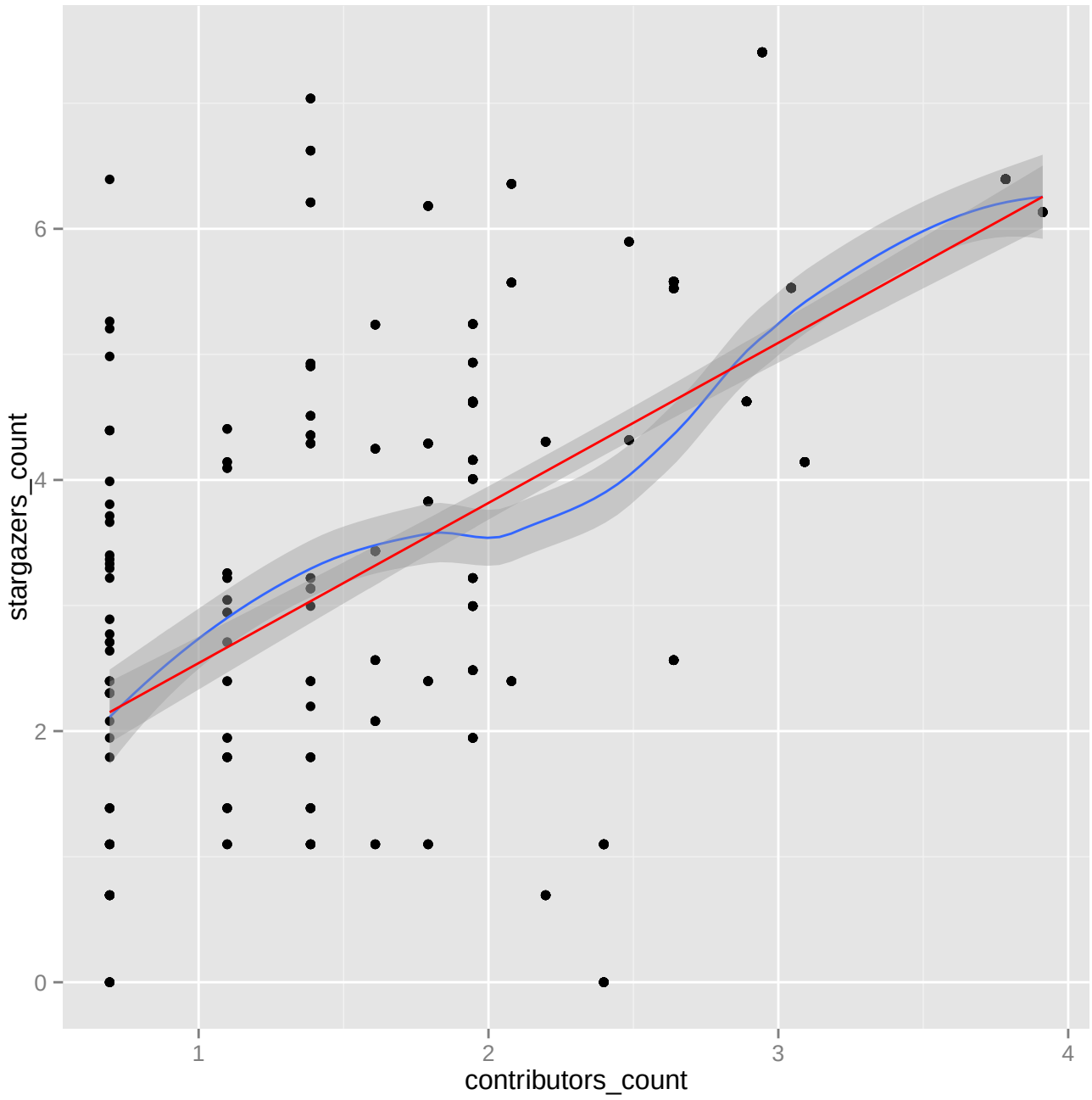
```
D$contributors_count <- log(1+D$contributors_count)
D$stargazers_count <- log(1+D$stargazers_count)
D$project_rating_count <- log(1+D$project_rating_count)
```

```
ggplot(D, aes(x=contributors_count, y=log10(1+stargazers_count))) + geom_point() + geom_smooth() + geom_smoot

## geom_smooth:  method="auto" and size of largest group is <1000, so using loess.  Use 'method = x' to
change the smoothing method.
```

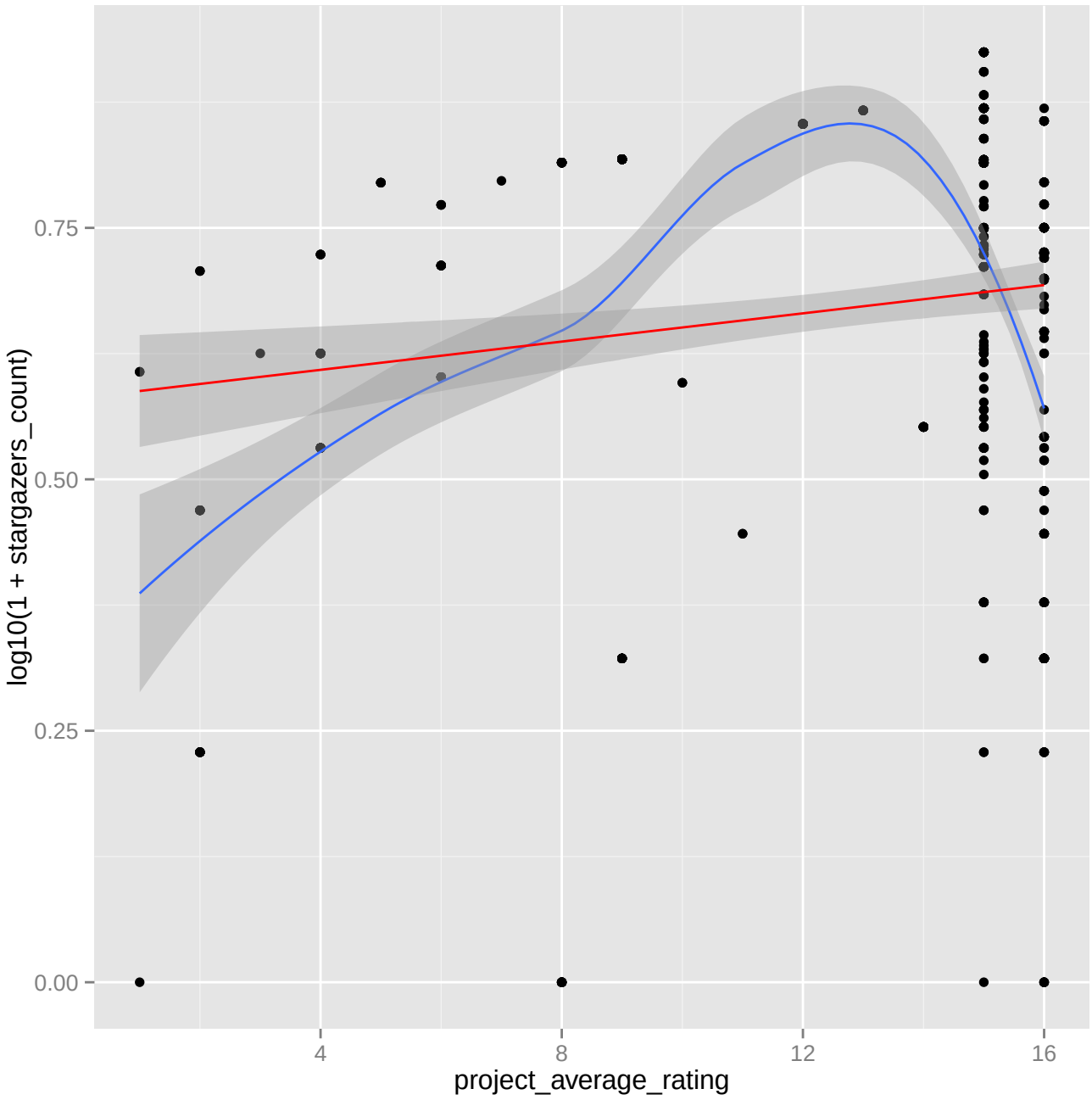



```
ggplot(D, aes(x=contributors_count, y=stargazers_count)) + geom_point() + geom_smooth() + geom_smooth(method=
```

```
## geom_smooth:  method="auto" and size of largest group is <1000, so using loess.  Use 'method = x' to
change the smoothing method.
```

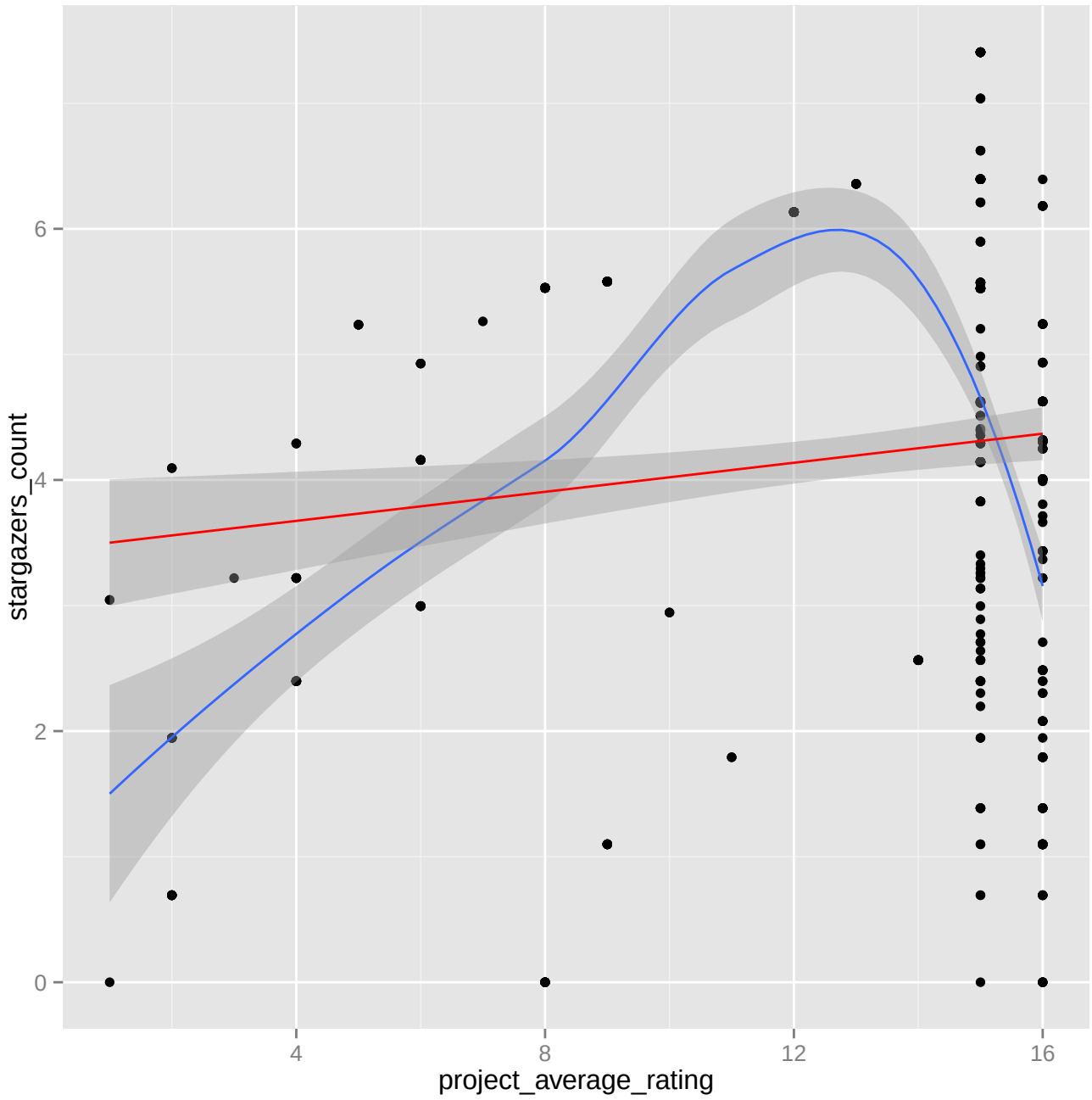



```
ggplot(D, aes(x=project_average_rating, y=log10(1+stargazers_count))) + geom_point() + geom_smooth() + geom_s

## geom_smooth:  method="auto" and size of largest group is <1000, so using loess.  Use 'method = x' to
change the smoothing method.
```

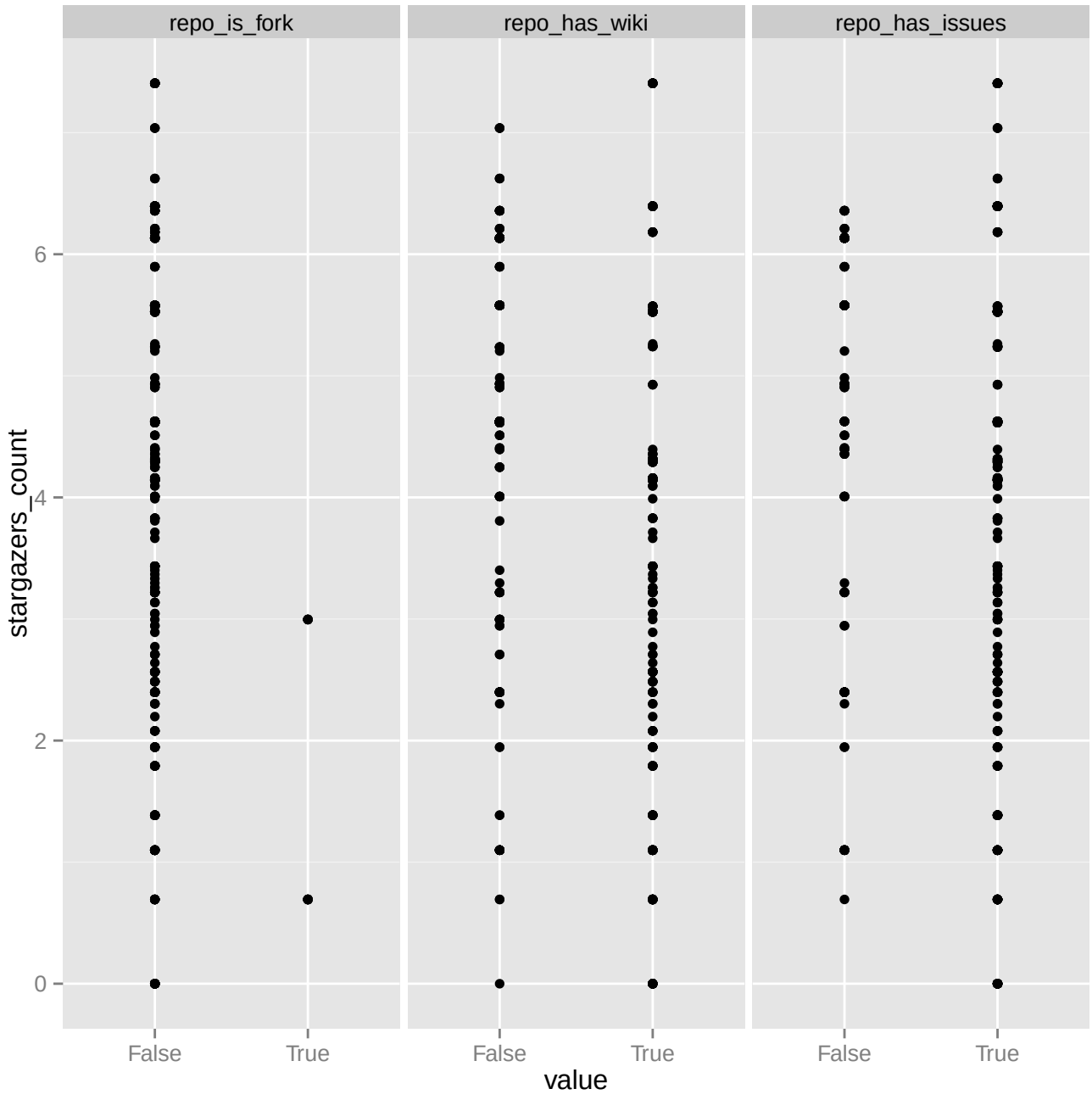



```
ggplot(D, aes(x=project_average_rating, y=stargazers_count)) + geom_point() + geom_smooth() + geom_smooth(met
```

```
## geom_smooth:  method="auto" and size of largest group is <1000, so using loess.  Use 'method = x' to
change the smoothing method.
```

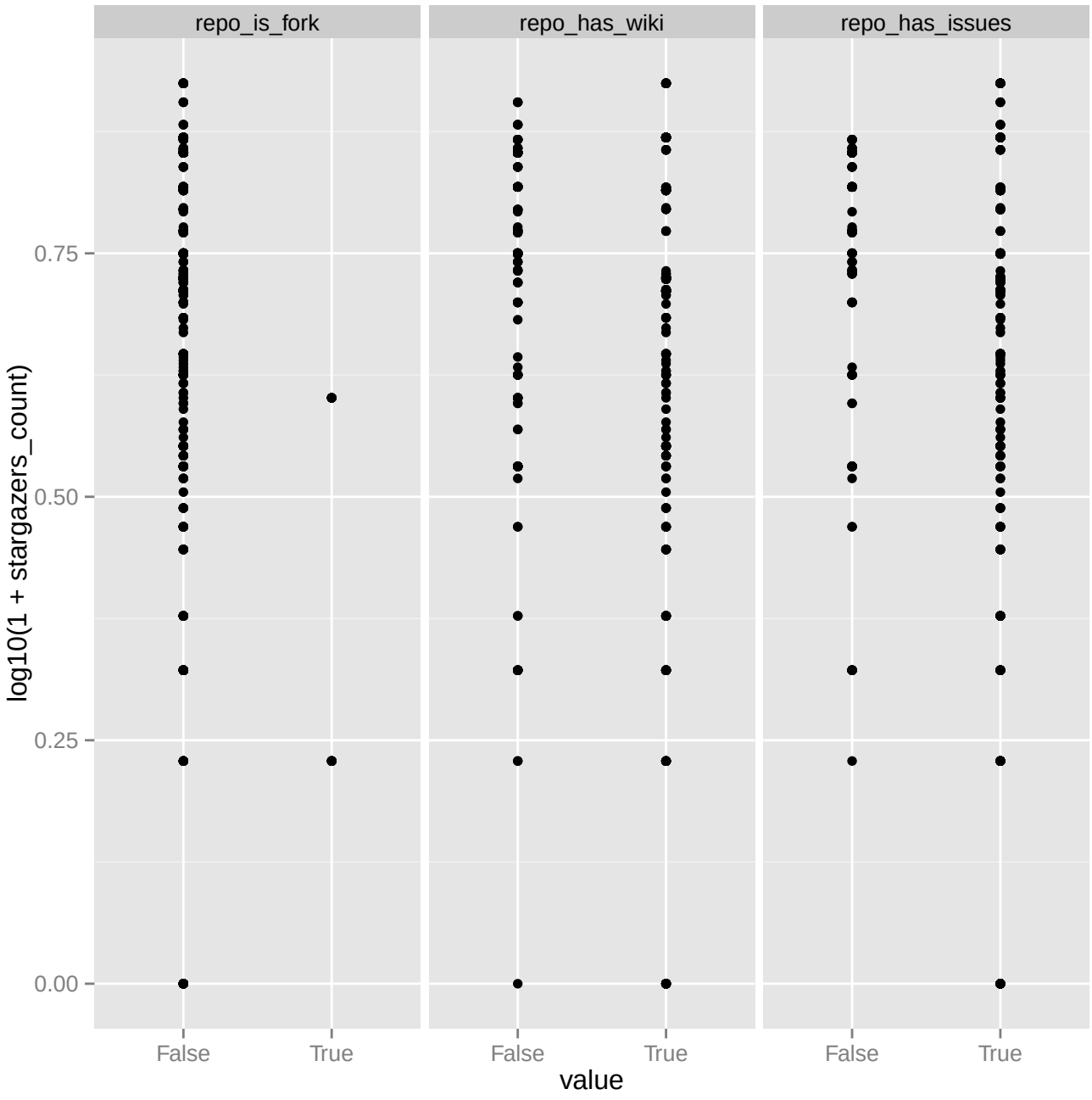



```
attrs <- c("repo_is_fork",
                  "repo_has_wiki", "repo_has_issues")
d <- cbind(melt(D[,attrs], id.vars=c()), stargazers_count=D$stargazers_count)
ggplot(d,aes(x = value, y=stargazers_count)) +
       facet_wrap(~variable, scales = "free_x") +
       geom_point() + geom_smooth() + geom_smooth(method=lm, color="red")

## geom_smooth:  Only one unique x value each group.Maybe you want aes(group = 1)?
## geom_smooth:  Only one unique x value each group.Maybe you want aes(group = 1)?
## geom_smooth:  Only one unique x value each group.Maybe you want aes(group = 1)?
## geom_smooth:  Only one unique x value each group.Maybe you want aes(group = 1)?
## geom_smooth:  Only one unique x value each group.Maybe you want aes(group = 1)?
## geom_smooth:  Only one unique x value each group.Maybe you want aes(group = 1)?
```



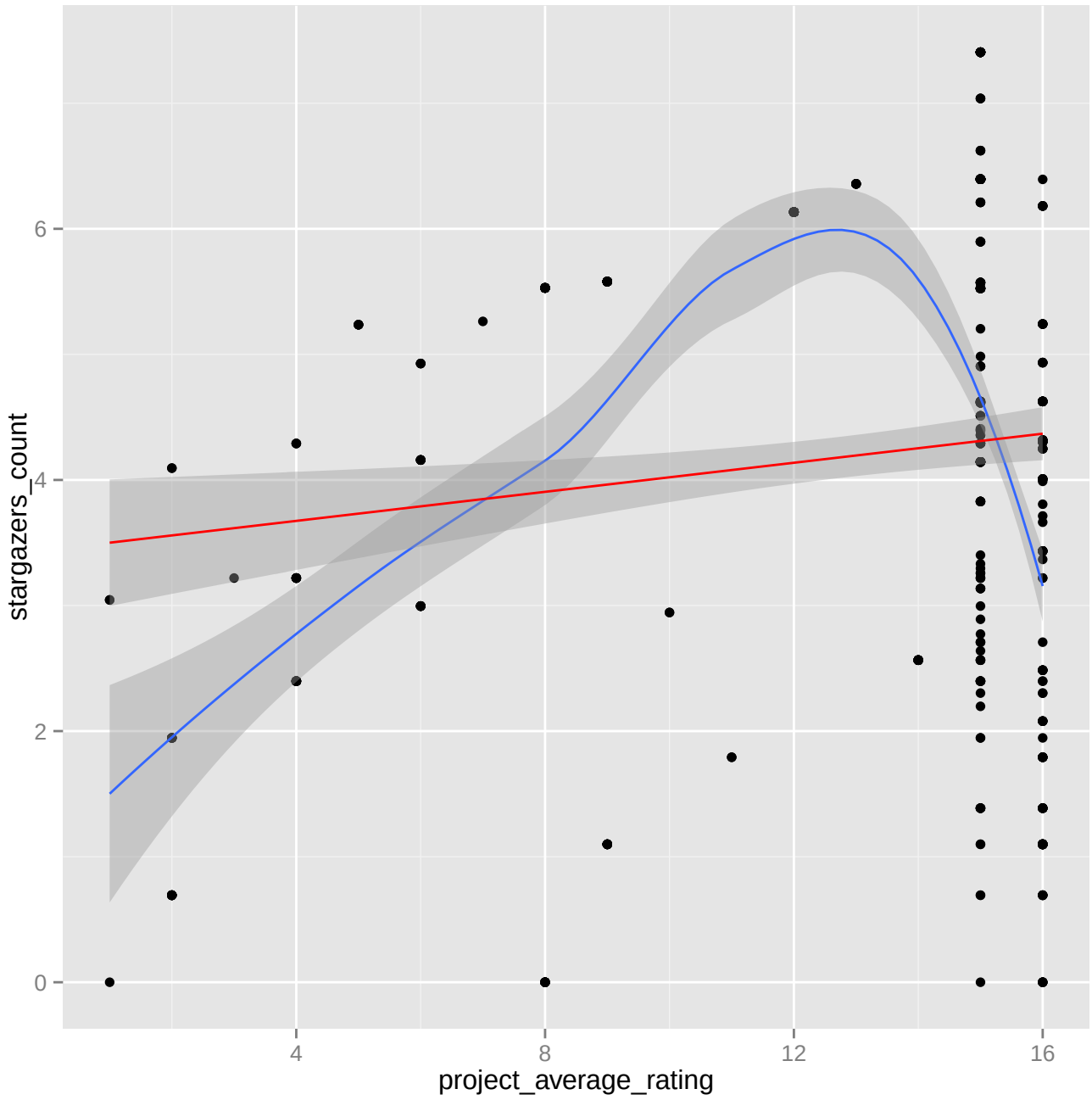

```
ggplot(d,aes(x = value, y=log10(1+stargazers_count))) +
       facet_wrap(~variable, scales = "free_x") +
       geom_point() + geom_smooth() + geom_smooth(method=lm, color="red")
```

```
## geom_smooth:  Only one unique x value each group.Maybe you want aes(group = 1)?
## geom_smooth:  Only one unique x value each group.Maybe you want aes(group = 1)?
## geom_smooth:  Only one unique x value each group.Maybe you want aes(group = 1)?
## geom_smooth:  Only one unique x value each group.Maybe you want aes(group = 1)?
## geom_smooth:  Only one unique x value each group.Maybe you want aes(group = 1)?
## geom_smooth:  Only one unique x value each group.Maybe you want aes(group = 1)?
```

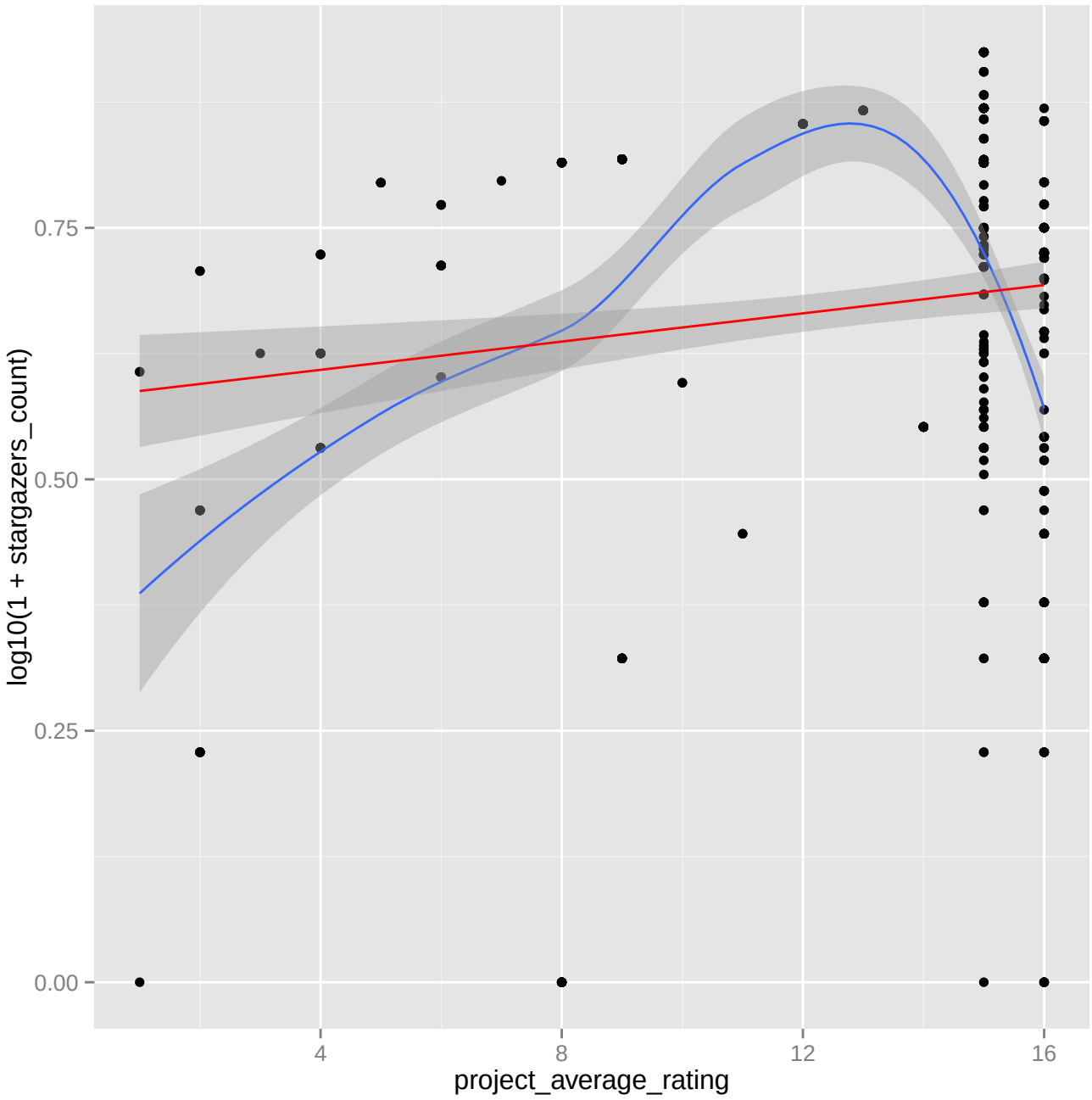



```
ggplot(D,aes(x = project_average_rating, y=stargazers_count)) +
       geom_point() + geom_smooth() + geom_smooth(method=lm, color="red")
```

```
## geom_smooth:  method="auto" and size of largest group is <1000, so using loess.  Use 'method = x' to
change the smoothing method.
```

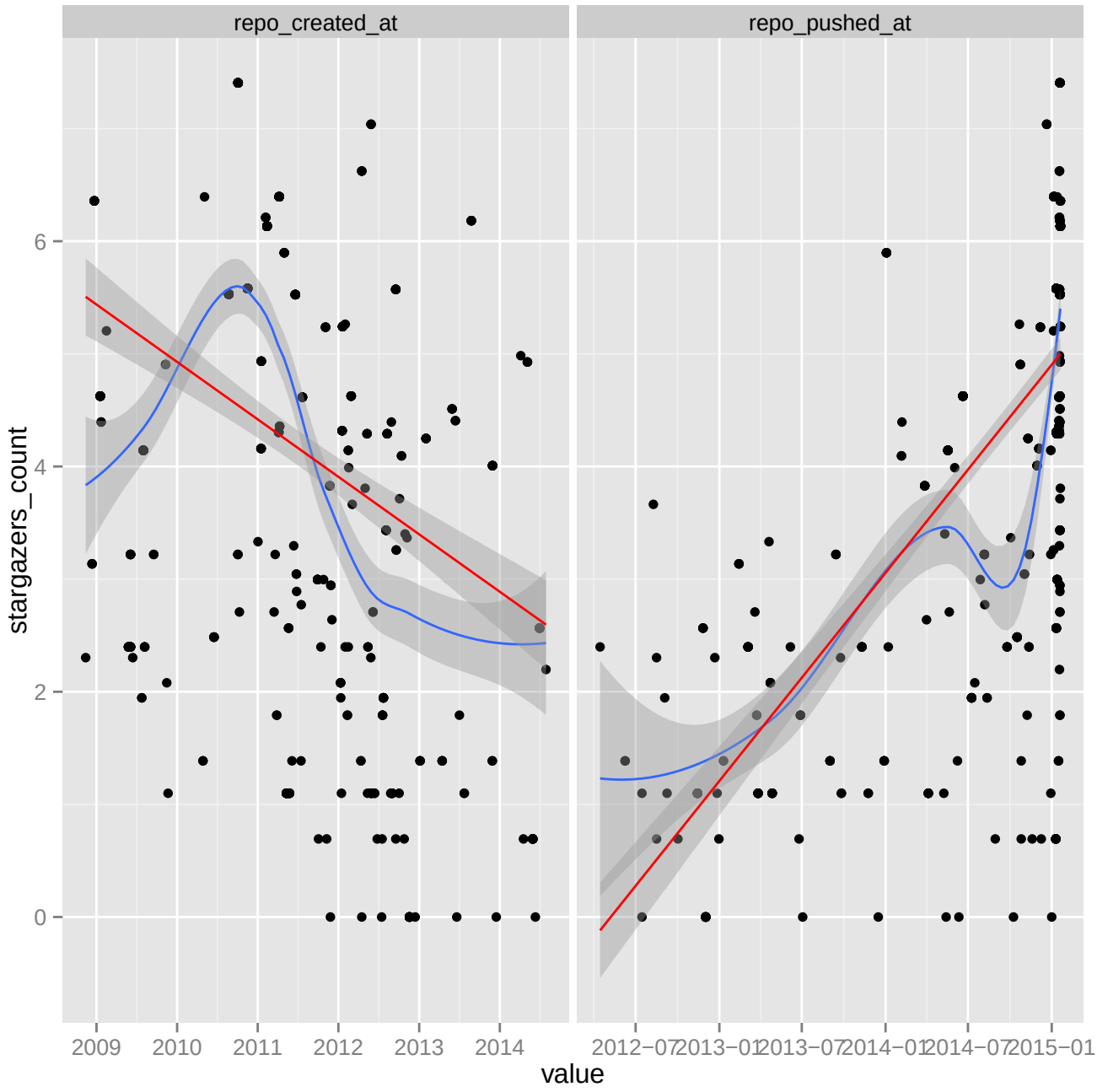



```
ggplot(D,aes(x = project_average_rating, y=log10(1+stargazers_count))) +
       geom_point() + geom_smooth() + geom_smooth(method=lm, color="red")

## geom_smooth:  method="auto" and size of largest group is <1000, so using loess.  Use 'method = x' to
change the smoothing method.
```



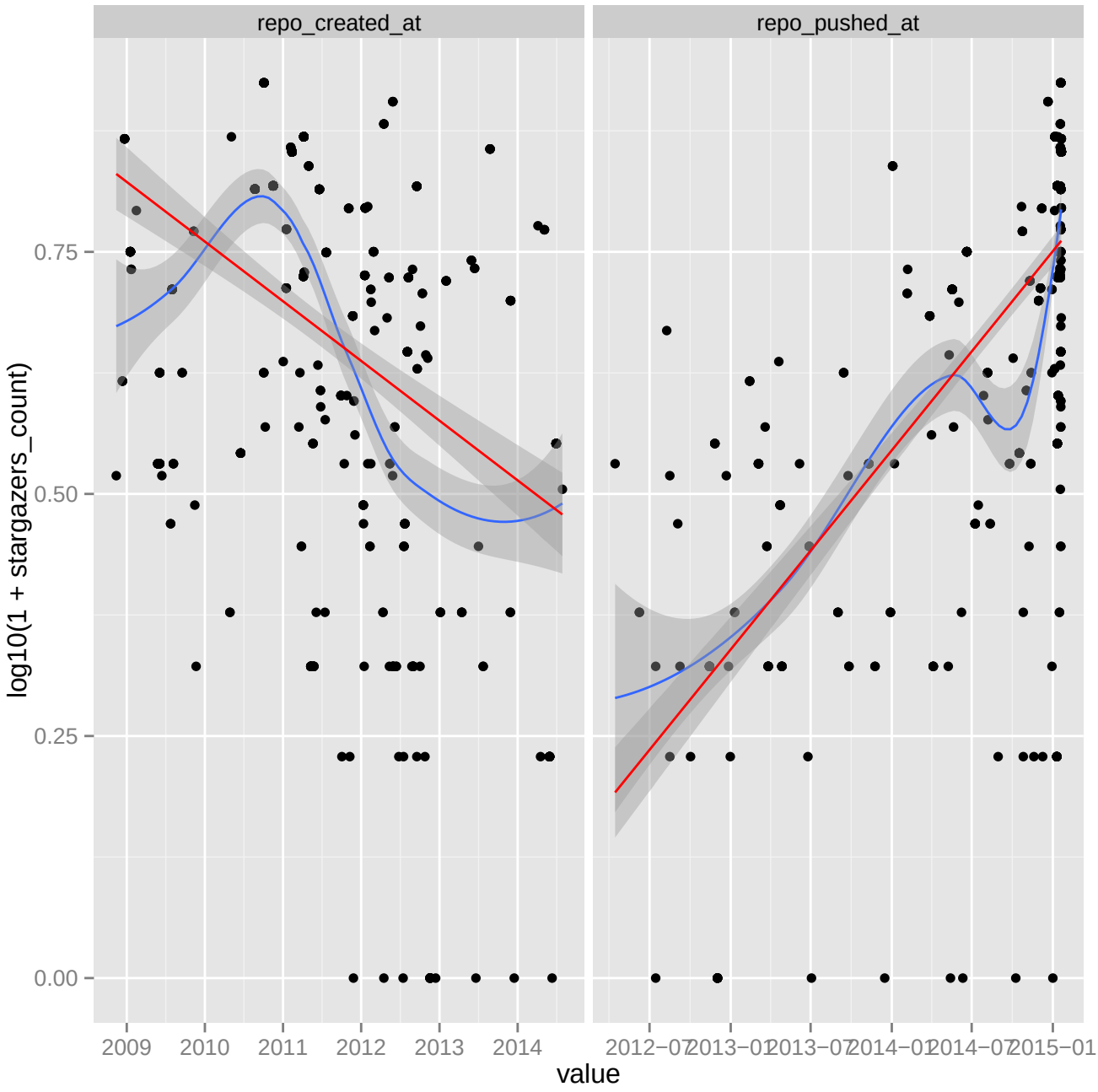

```
d <- cbind(melt(D[,c("repo_created_at", "repo_pushed_at")], id.vars=c()), stargazers_count=D$stargazers_count
d$value <- as.Date(d$value, origin="1970-10-01")
ggplot(d,aes(x = value, y=stargazers_count)) +
       facet_wrap(~variable, scales = "free_x") +
       geom_point() + geom_smooth() + geom_smooth(method=lm, color="red")
```

```
## geom_smooth:  method="auto" and size of largest group is <1000, so using loess.  Use 'method = x' to
change the smoothing method.
## geom_smooth:  method="auto" and size of largest group is <1000, so using loess.  Use 'method = x' to
change the smoothing method.
```




```
ggplot(d,aes(x = value, y=log10(1+stargazers_count))) +
       facet_wrap(~variable, scales = "free_x") +
       geom_point() + geom_smooth() + geom_smooth(method=lm, color="red")
```

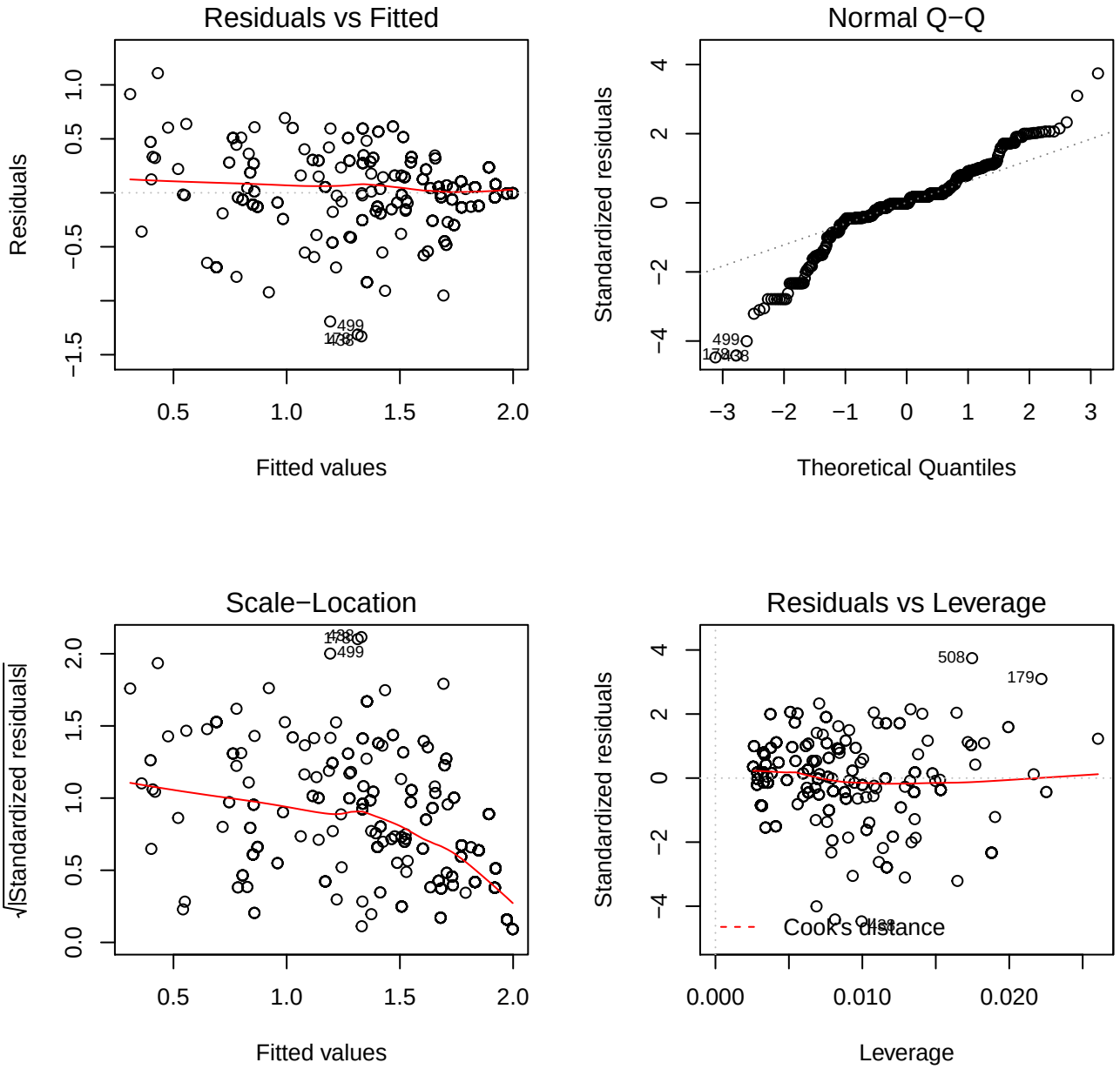
```
## geom_smooth:  method="auto" and size of largest group is <1000, so using loess.  Use 'method = x' to
change the smoothing method.
## geom_smooth:  method="auto" and size of largest group is <1000, so using loess.  Use 'method = x' to
change the smoothing method.
```




```
D$stargazers_count <- log(1+D$stargazers_count)
```

```
par(mfrow=c(2,2))
m <- lm(stargazers_count ~ contributors_count + repo_pushed_at + repo_created_at, D, na.action=na.exclude)
summary(m)

##
## Call:
## lm(formula = stargazers_count ~ contributors_count + repo_pushed_at +
##     repo_created_at, data = D, na.action = na.exclude)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.32988 -0.12103 -0.00253  0.12444  1.10872
##
```

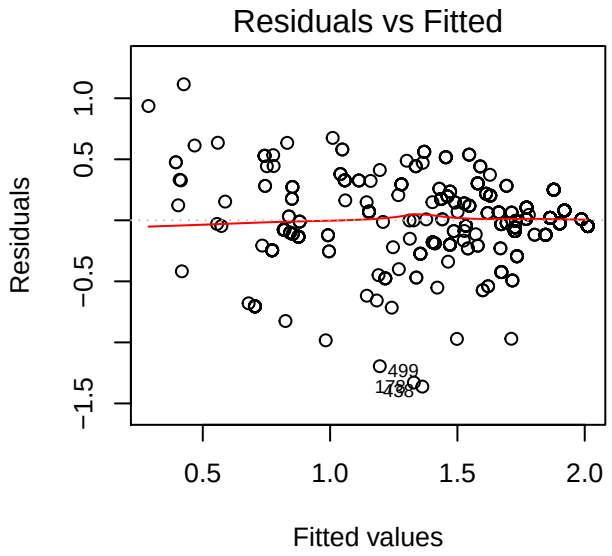
```
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -1.227e+01  9.090e-01 -13.500  < 2e-16 ***
## contributors_count  1.249e-01  1.489e-02   8.389 4.22e-16 ***
## repo_pushed_at      1.133e-03  5.384e-05  21.051  < 2e-16 ***
## repo_created_at    -3.260e-04  2.792e-05 -11.676  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2987 on 545 degrees of freedom
## Multiple R-squared:  0.6456,Adjusted R-squared:  0.6437
## F-statistic:   331 on 3 and 545 DF,  p-value: < 2.2e-16

plot(m)
```

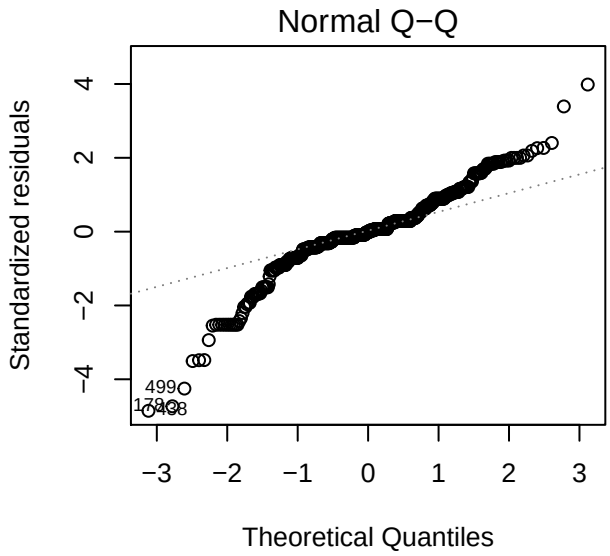


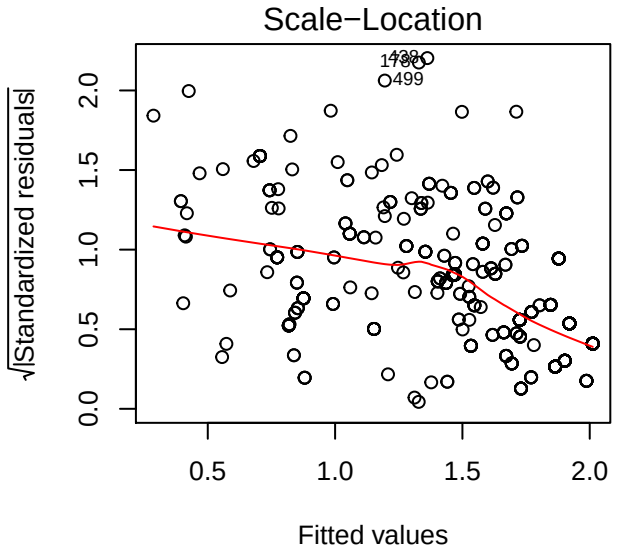
```r
par(mfrow=c(2,2))
m2 <- lm(stargazers_count ~ contributors_count + repo_pushed_at + repo_created_at + repo_is_fork + repo_has_w
summary(m2)
```

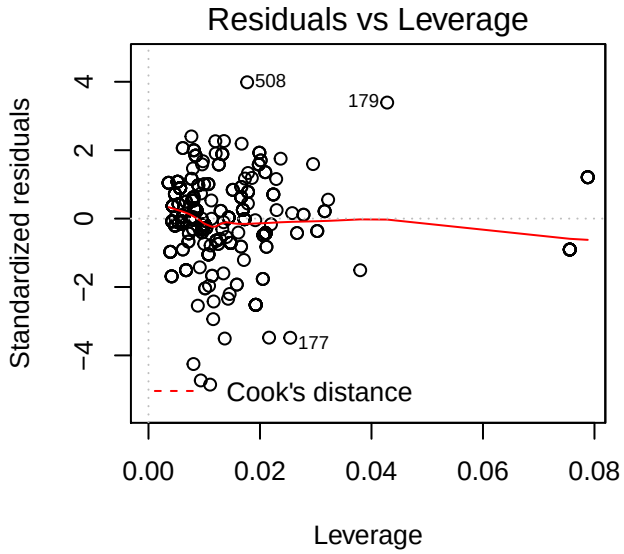
```
##
## Call:
## lm(formula = stargazers_count ~ contributors_count + repo_pushed_at +
##     repo_created_at + repo_is_fork + repo_has_wiki + repo_has_issues,
##     data = D, na.action = na.exclude)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.36227 -0.08847 -0.00453  0.10422  1.11454
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -1.356e+01  8.729e-01 -15.530  < 2e-16 ***
## contributors_count   1.260e-01  1.431e-02   8.806  < 2e-16 ***
## repo_pushed_at       1.157e-03  5.160e-05  22.420  < 2e-16 ***
## repo_created_at     -2.637e-04  2.853e-05  -9.243  < 2e-16 ***
## repo_is_forkTrue    -6.496e-01  8.016e-02  -8.103 3.57e-15 ***
## repo_has_wikiTrue   -5.660e-02  3.987e-02  -1.420    0.156
## repo_has_issuesTrue  1.158e-02  4.130e-02   0.280    0.779
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2822 on 542 degrees of freedom
## Multiple R-squared:  0.6855,Adjusted R-squared:  0.682
## F-statistic: 196.9 on 6 and 542 DF,  p-value: < 2.2e-16
```

```r
plot(m2)
```



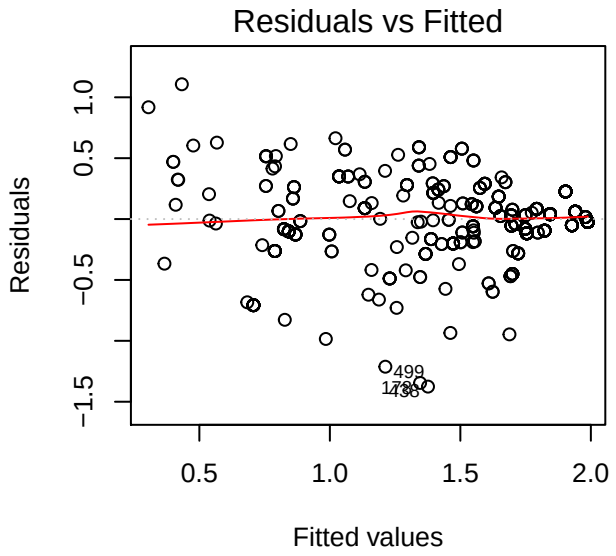


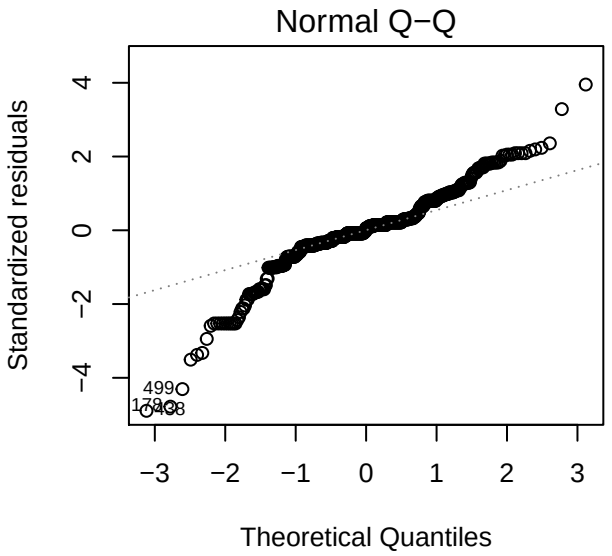


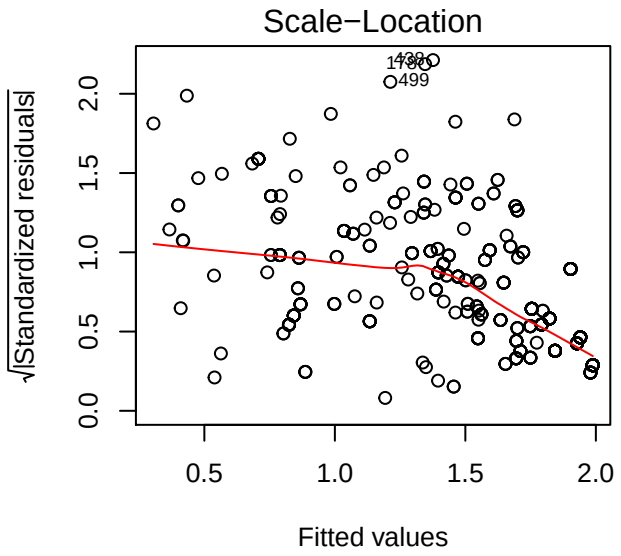


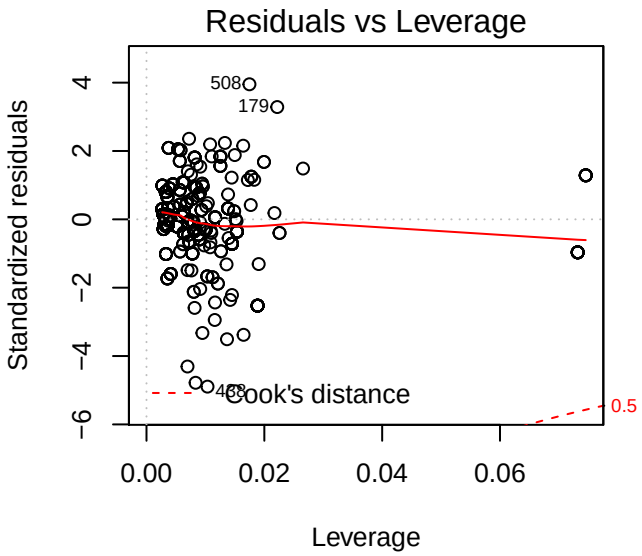

```
anova(m, m2)

## Analysis of Variance Table
##
## Model 1: stargazers_count ~ contributors_count + repo_pushed_at + repo_created_at
## Model 2: stargazers_count ~ contributors_count + repo_pushed_at + repo_created_at +
##     repo_is_fork + repo_has_wiki + repo_has_issues
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    545 48.631
## 2    542 43.159  3    5.4721 22.906 5.651e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

par(mfrow=c(2,2))
m3 <- lm(stargazers_count ~ contributors_count + repo_pushed_at + repo_created_at + repo_is_fork, D, na.actio
summary(m3)
```

```
##
## Call:
## lm(formula = stargazers_count ~ contributors_count + repo_pushed_at +
##     repo_created_at + repo_is_fork, data = D, na.action = na.exclude)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.37599 -0.10128  0.01637  0.10440  1.10693
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -1.352e+01  8.737e-01 -15.476  < 2e-16 ***
## contributors_count  1.247e-01  1.409e-02   8.850  < 2e-16 ***
## repo_pushed_at      1.169e-03  5.113e-05  22.873  < 2e-16 ***
## repo_created_at    -2.810e-04  2.699e-05 -10.411  < 2e-16 ***
## repo_is_forkTrue   -6.353e-01  7.877e-02  -8.066 4.66e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2826 on 544 degrees of freedom
## Multiple R-squared:  0.6835,Adjusted R-squared:  0.6811
## F-statistic: 293.7 on 4 and 544 DF,  p-value: < 2.2e-16

plot(m3)
```










```
anova(m, m3)

## Analysis of Variance Table
##
## Model 1: stargazers_count ~ contributors_count + repo_pushed_at + repo_created_at
## Model 2: stargazers_count ~ contributors_count + repo_pushed_at + repo_created_at +
##     repo_is_fork
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    545 48.631
## 2    544 43.437  1    5.1947 65.059 4.664e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
D$star_resid <- resid(m3)
```

```r
save(D, file = "../project_stars.RData")
```