```
## 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 frameworks - data analysis

WikiTeams.pl

28 December 2014 - 7 January 2015

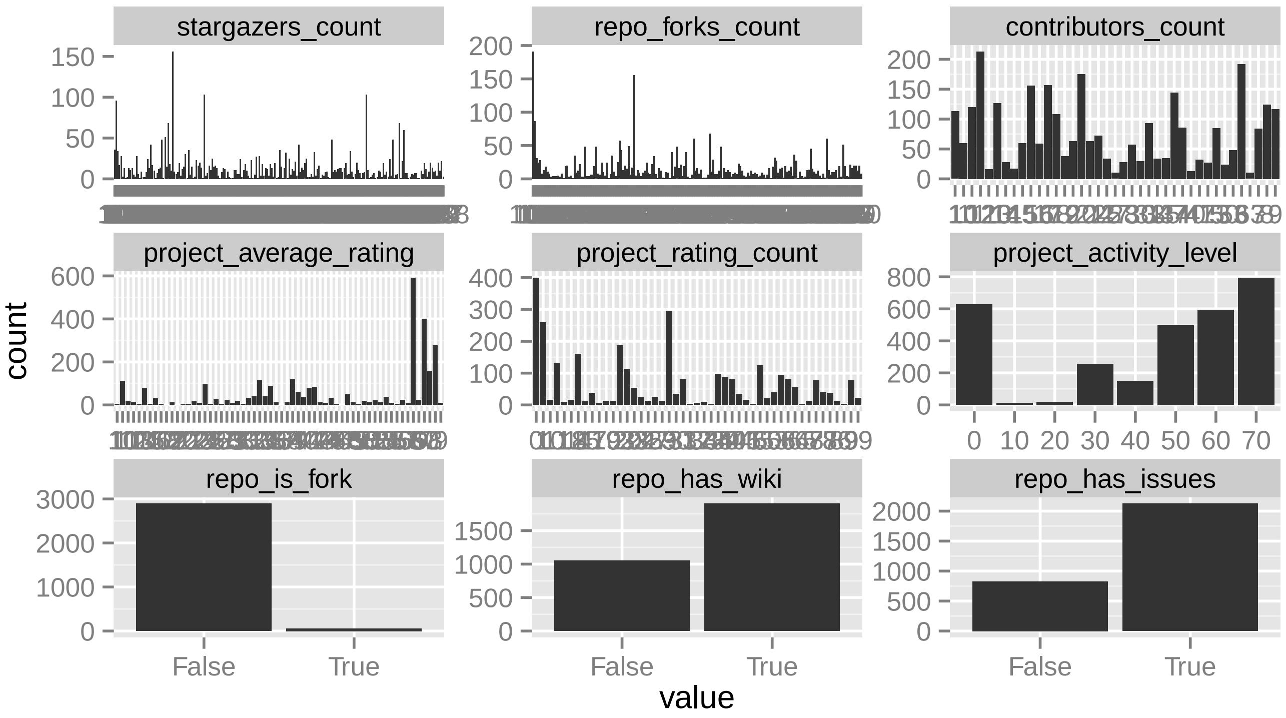
```
options("warn" = -1)
```

1 Read in the data

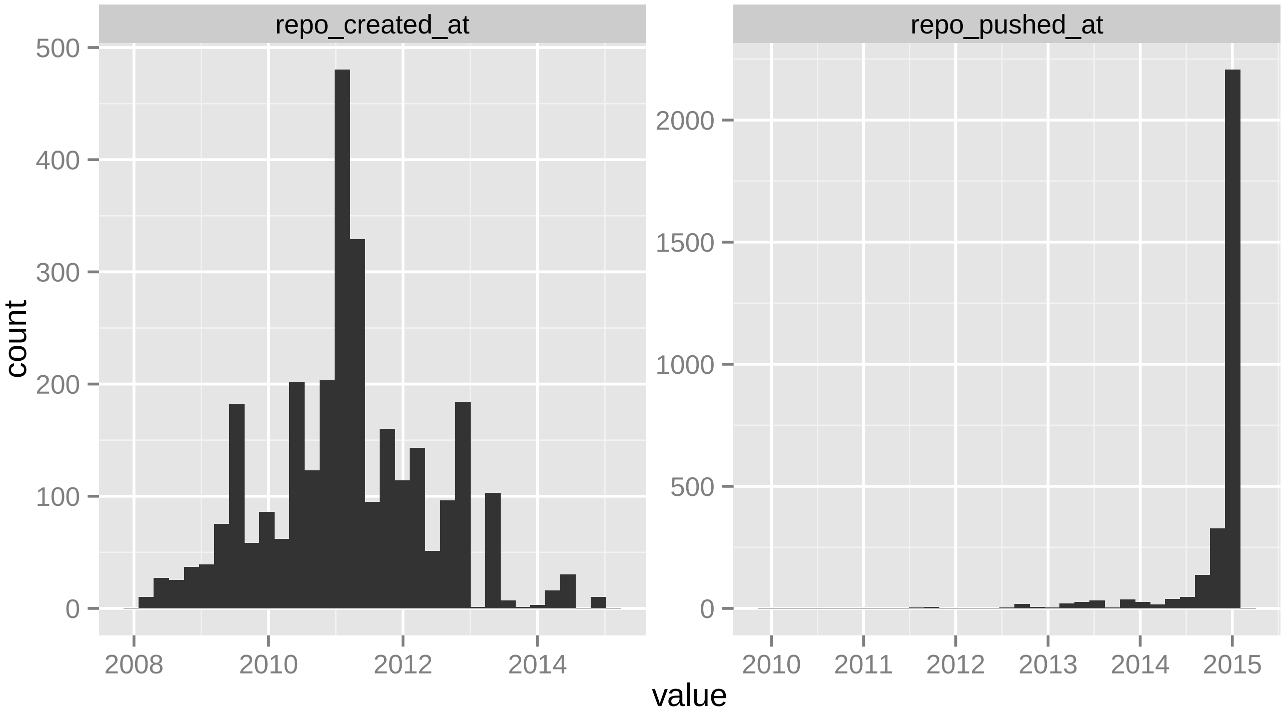
```
D <- read.table("../results.csv", sep=";", quote = "\"", header=T)
names(D)

##  [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 2952 recods.

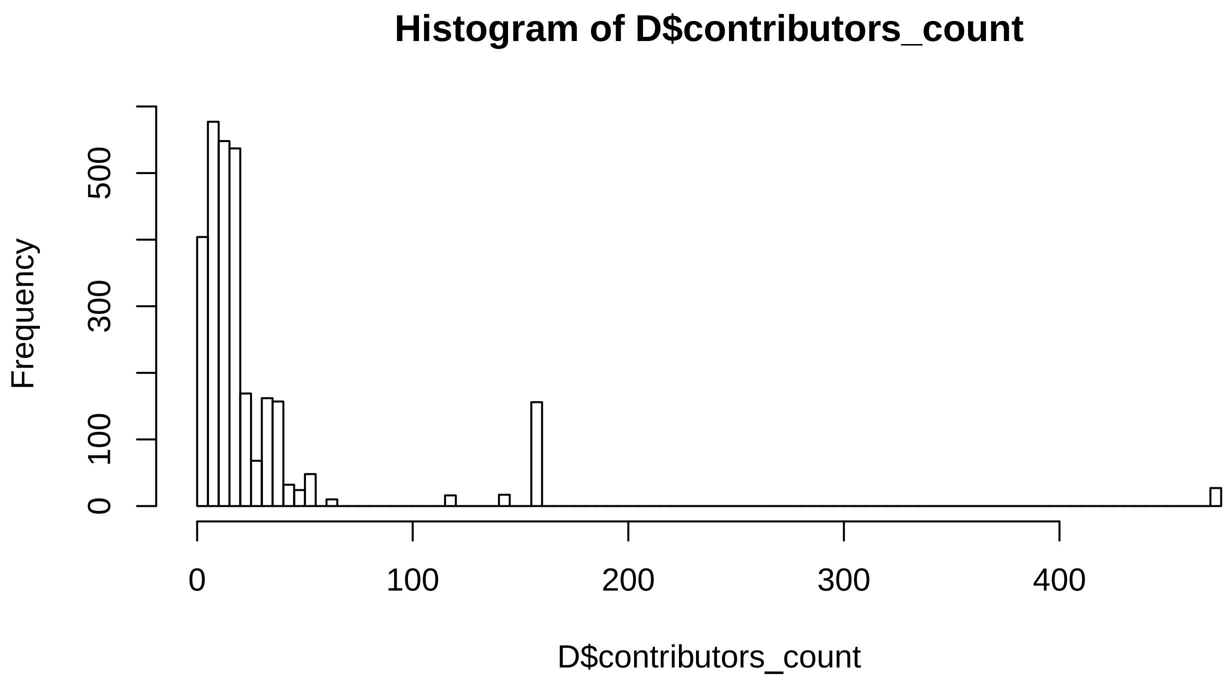
```
# discrete
plot_mhist(D, attrs=c("stargazers_count", "repo_forks_count", "contributors_count", "project_average_rating",
```



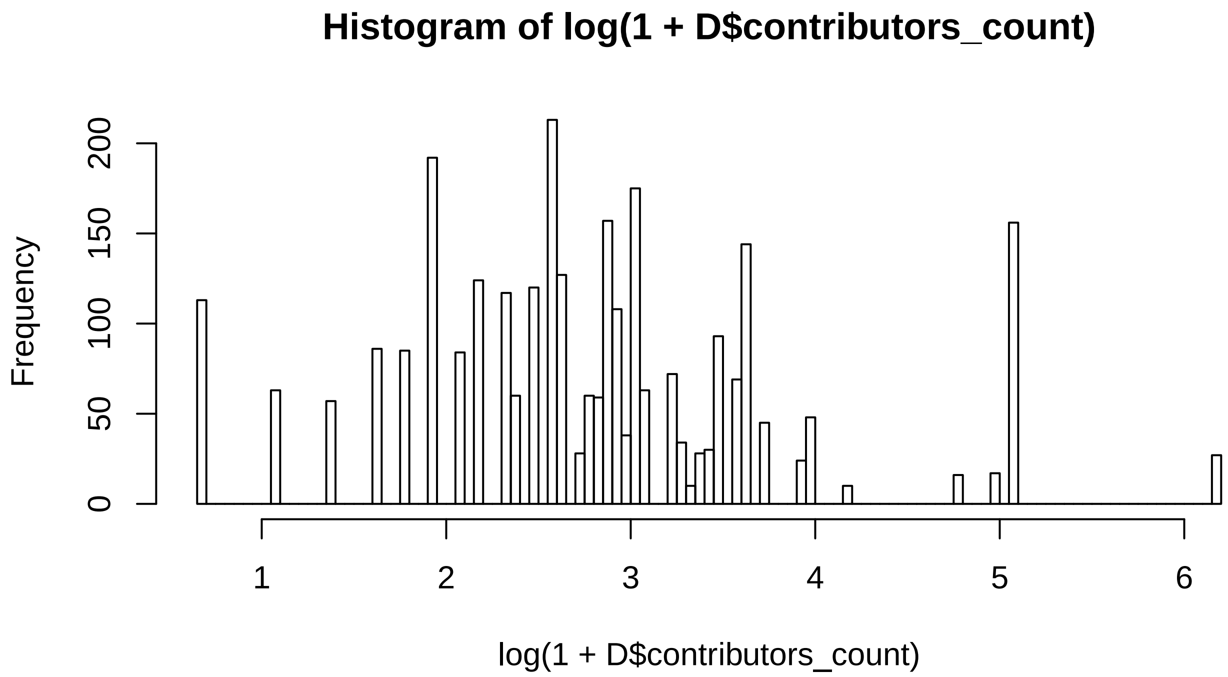
```
# continuous
plot_mhist(D, attrs=c("repo_created_at", "repo_pushed_at"), date.values = T)

## 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.
```



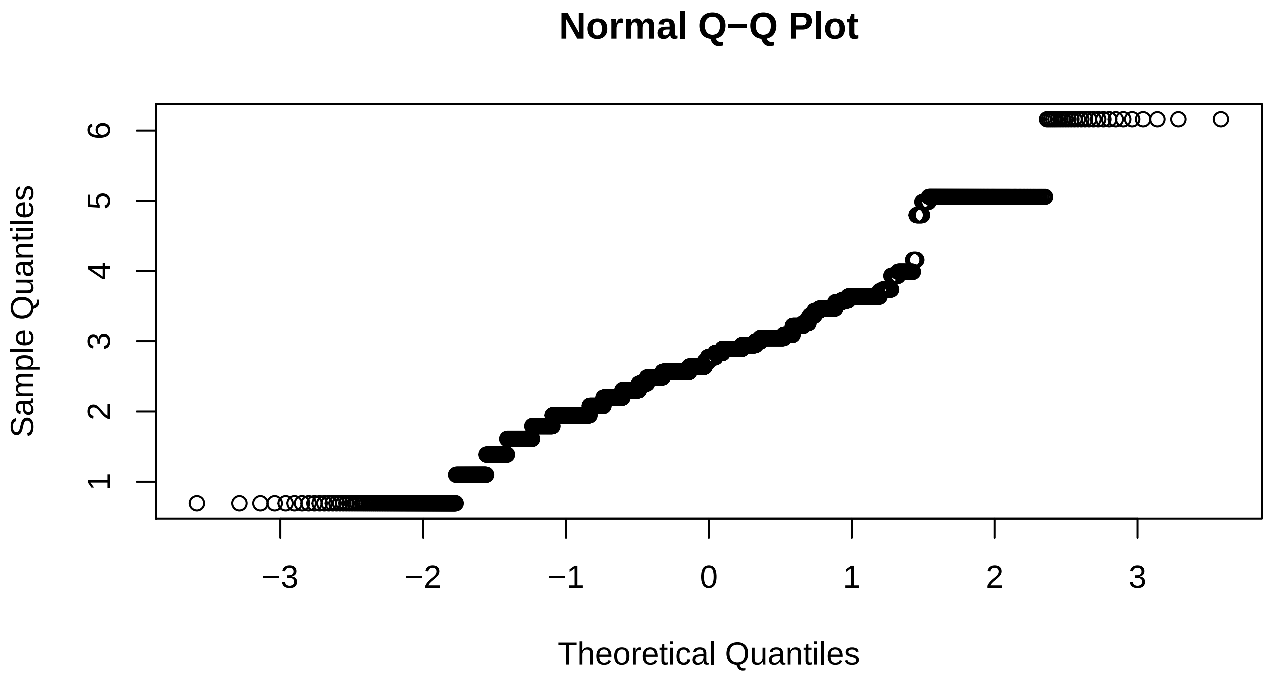
```
# contrib count
hist(D$contributors_count, breaks=100)
```



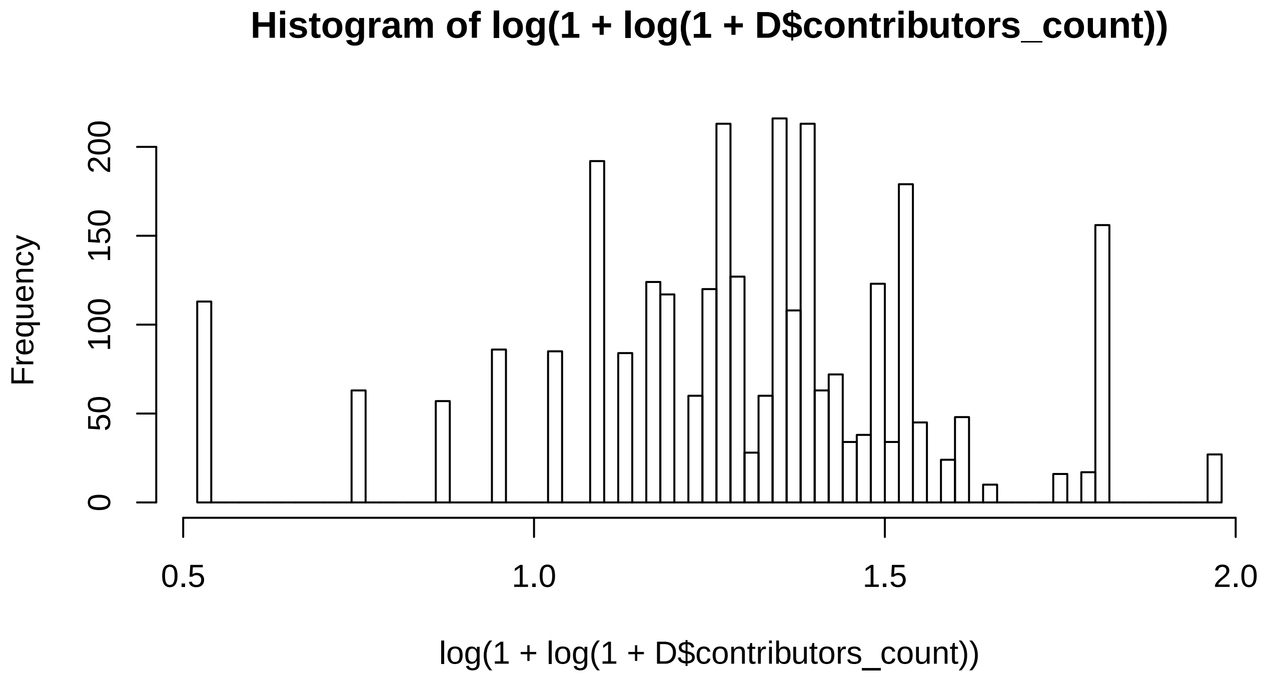
```
summary(D$contributors_count, breaks=100)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    8.00   15.00   29.24   25.00  473.00

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



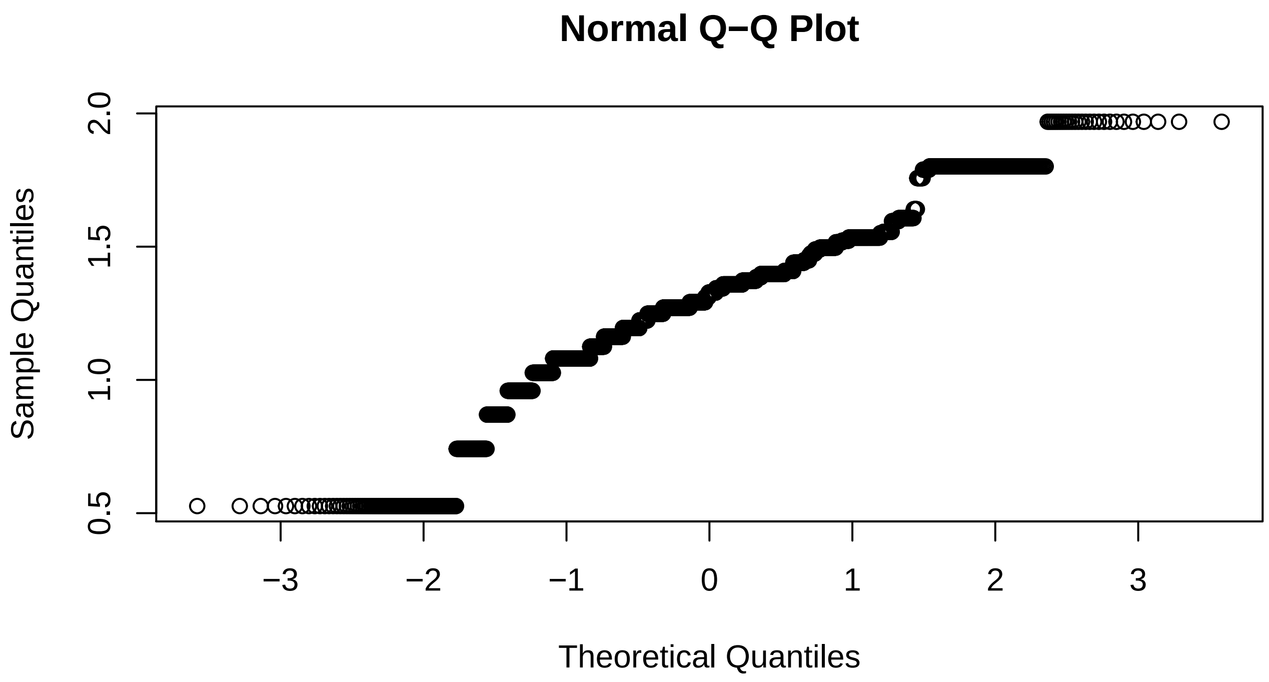
```
qqnorm(log(1+D$contributors_count))
```



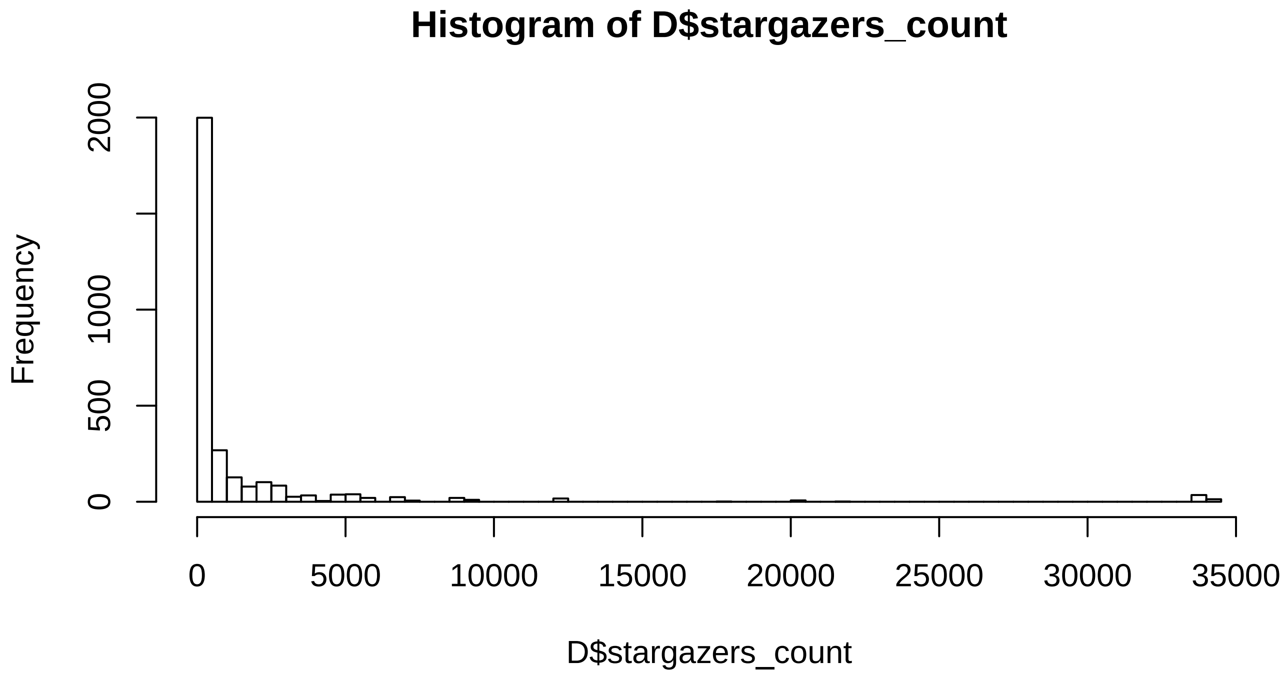
```
hist(log(1+log(1+D$contributors_count)), breaks=100)
```



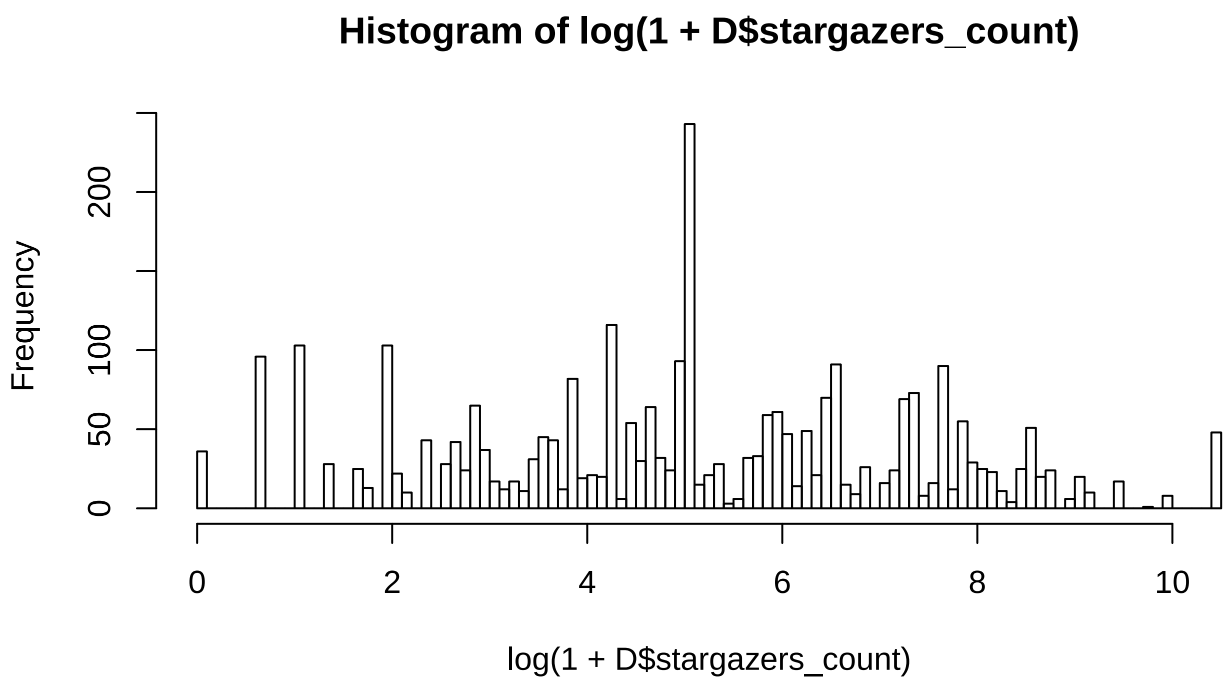
```
qqnorm(log(1+log(1+D$contributors_count)))
```



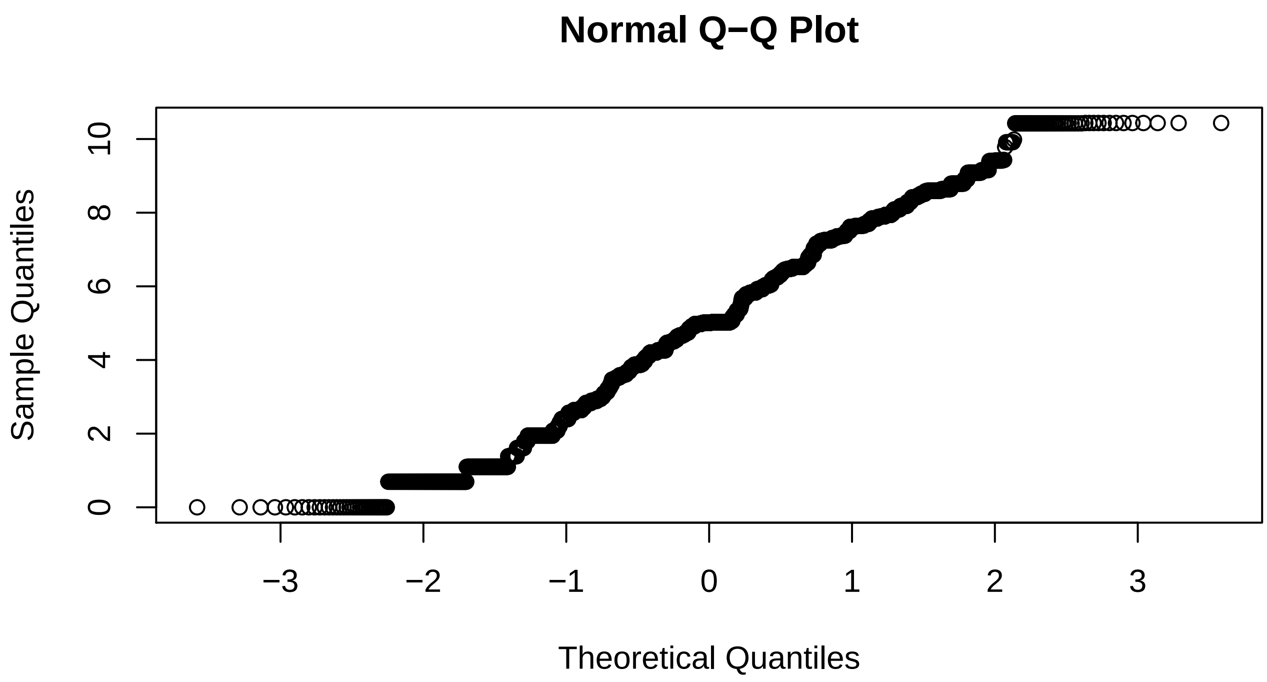
```
summary(log(1+D$contributors_count), breaks=100)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.6931  2.1970  2.7730  2.7880  3.2580  6.1610

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



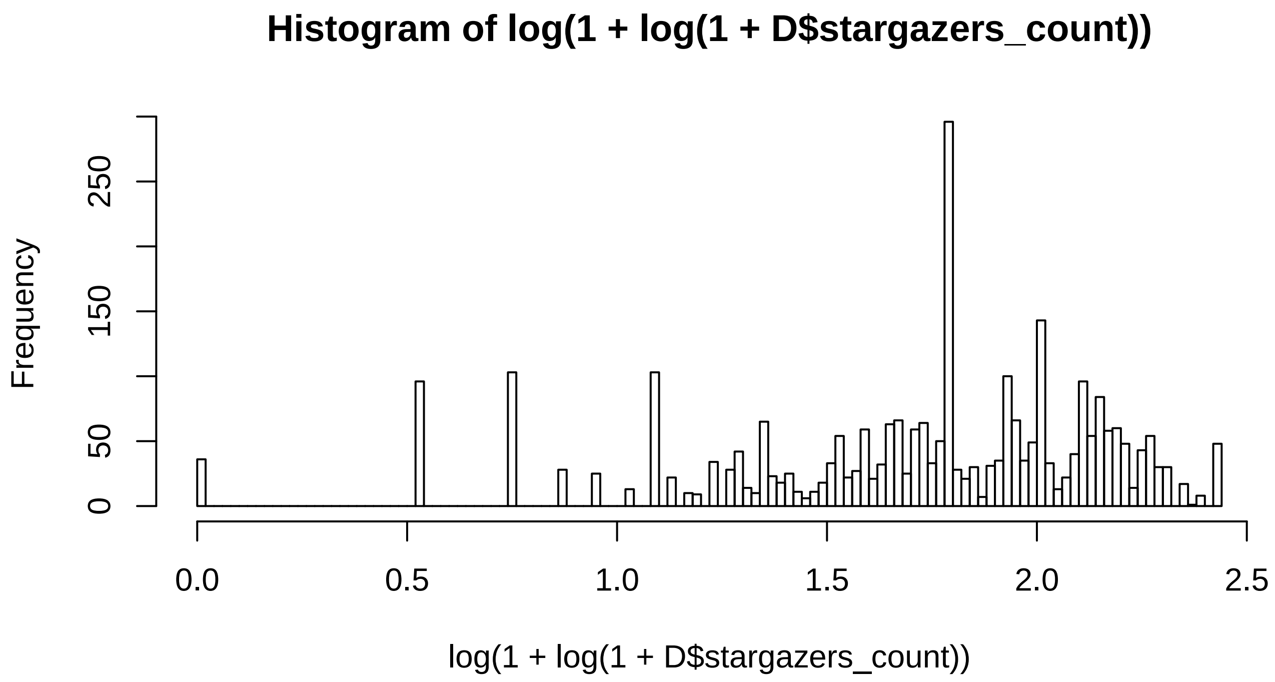
```
hist(log(1+D$stargazers_count), breaks=100)
```



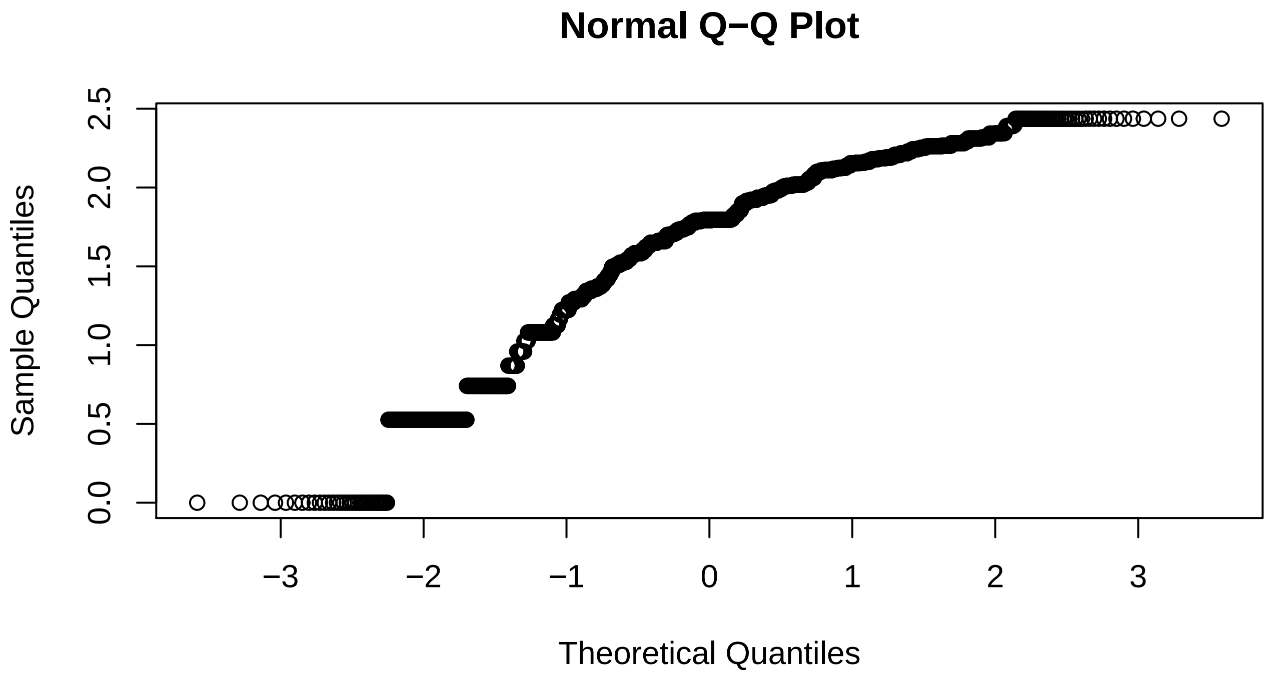
```
qqnorm(log(1+D$stargazers_count))
```



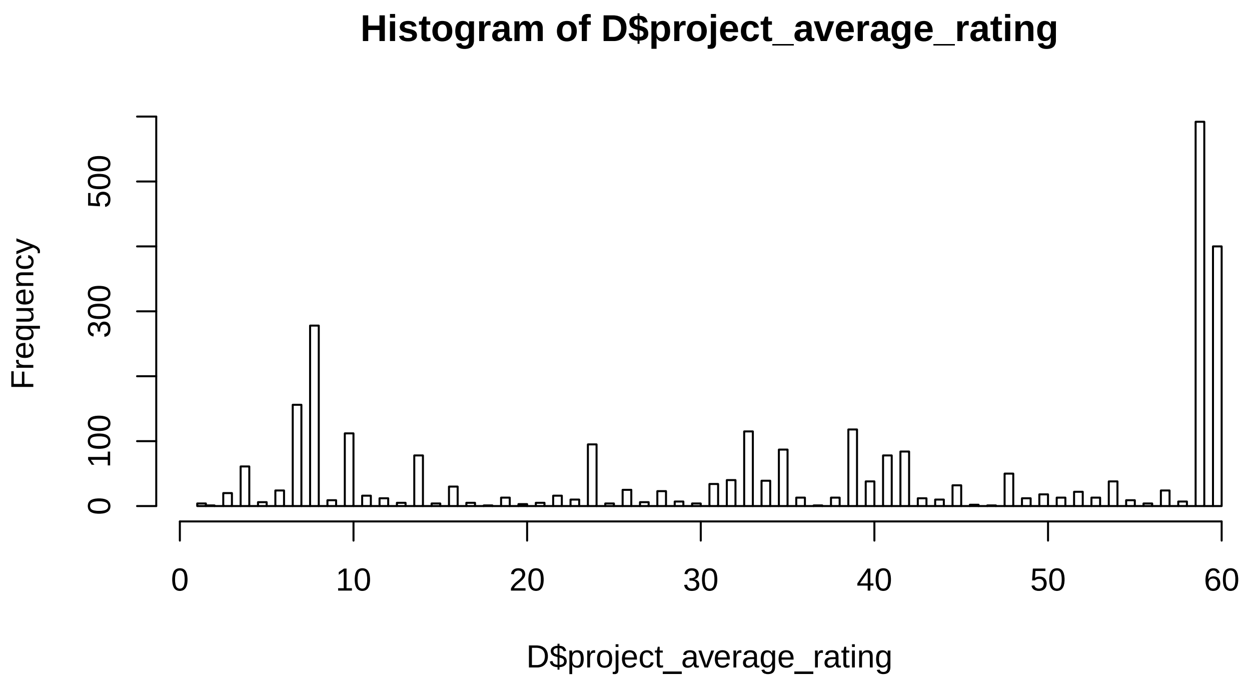
```
hist(log(1+log(1+D$stargazers_count)), breaks=100)
```



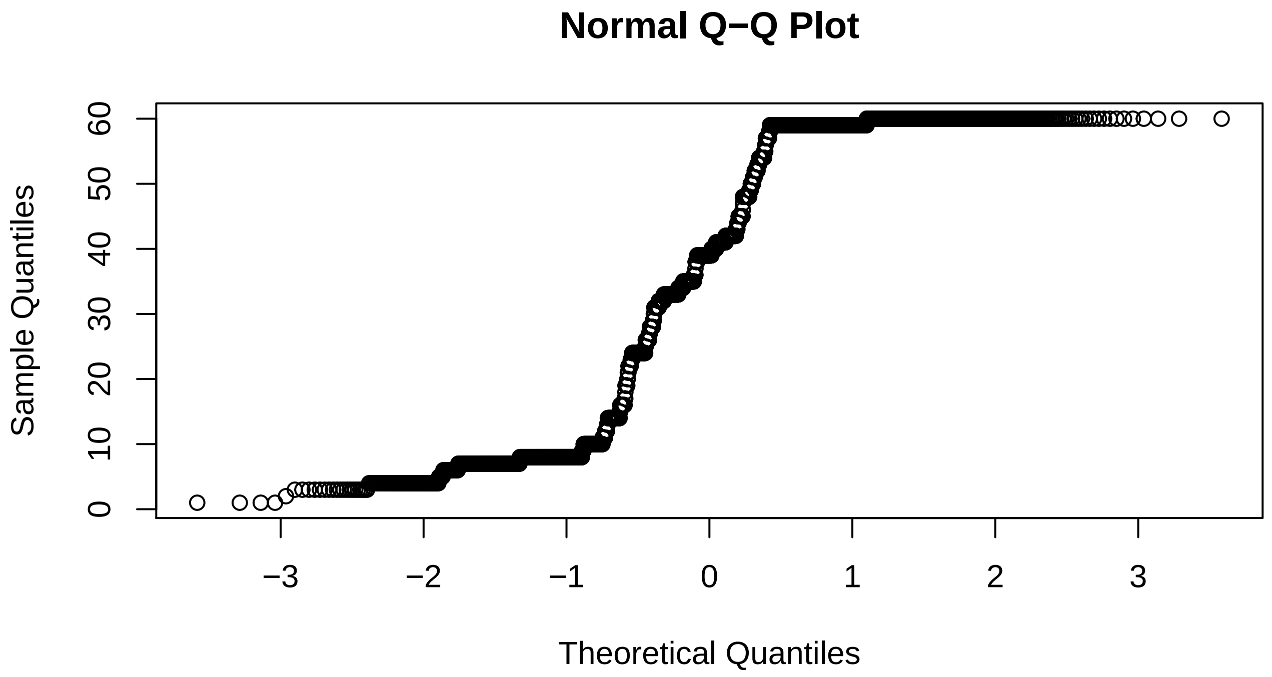
```
qqnorm(log(1+log(1+D$stargazers_count)))
```



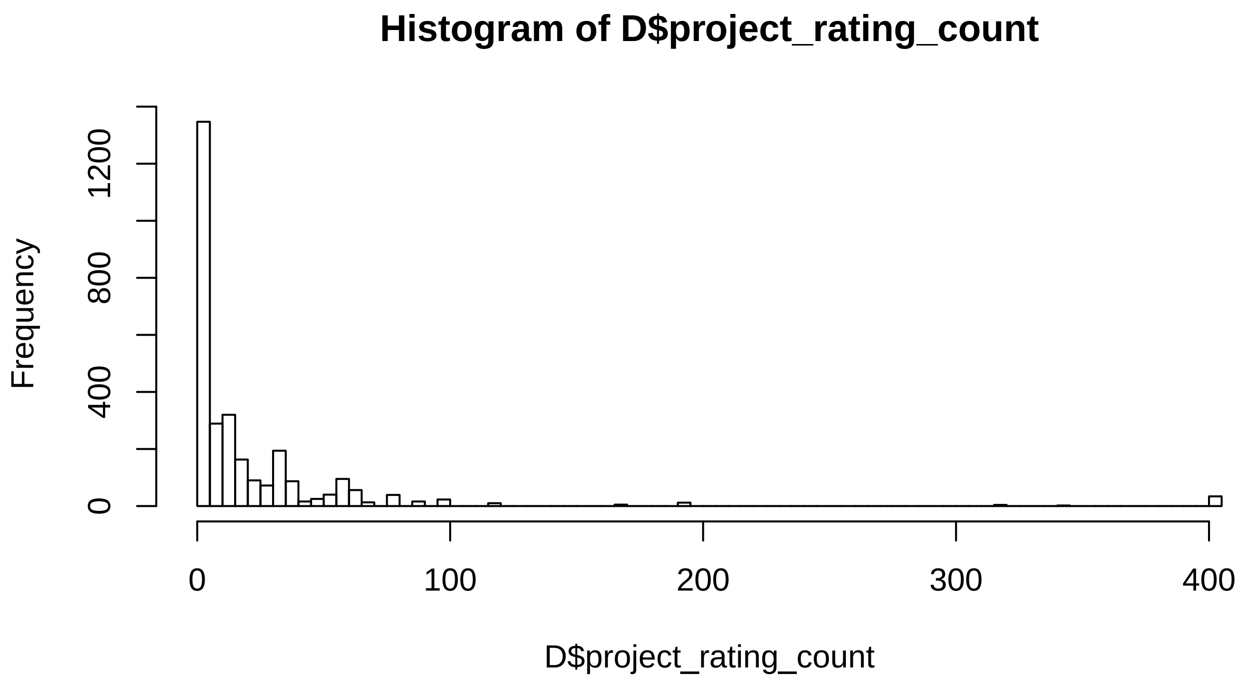
```
summary(D$stargazers_count)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0    31.0   149.0  1471.0   727.8 34030.0

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



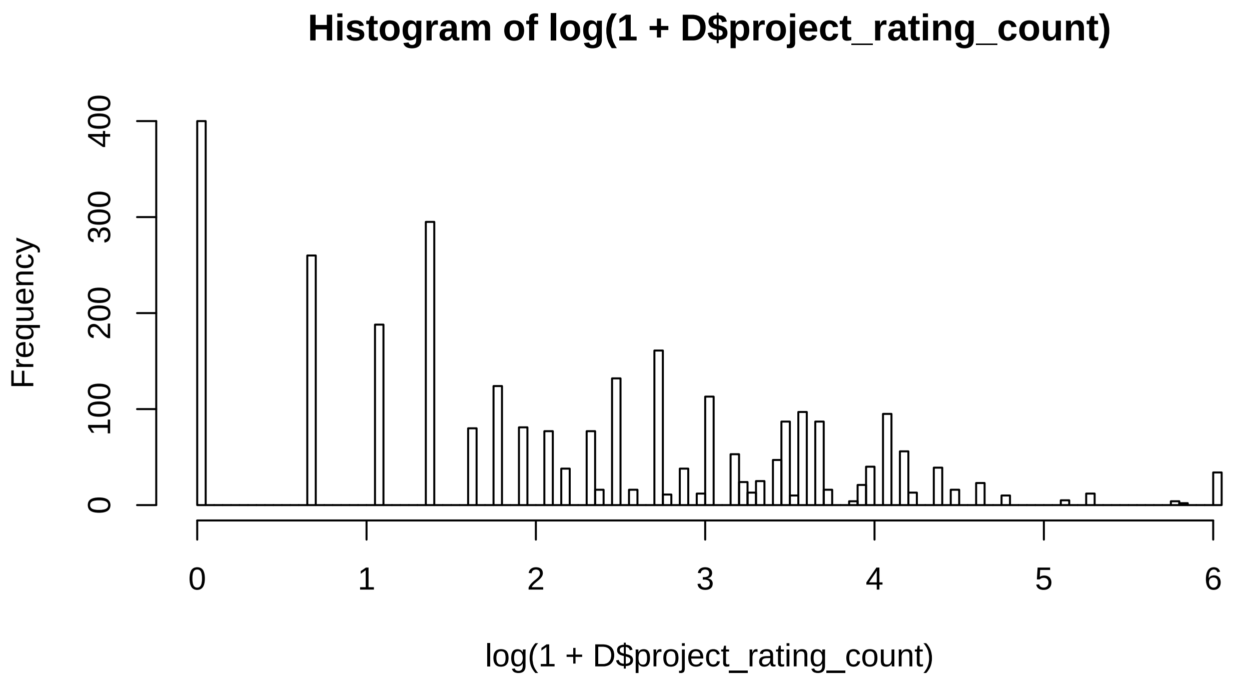
```
qqnorm(D$project_average_rating)
```



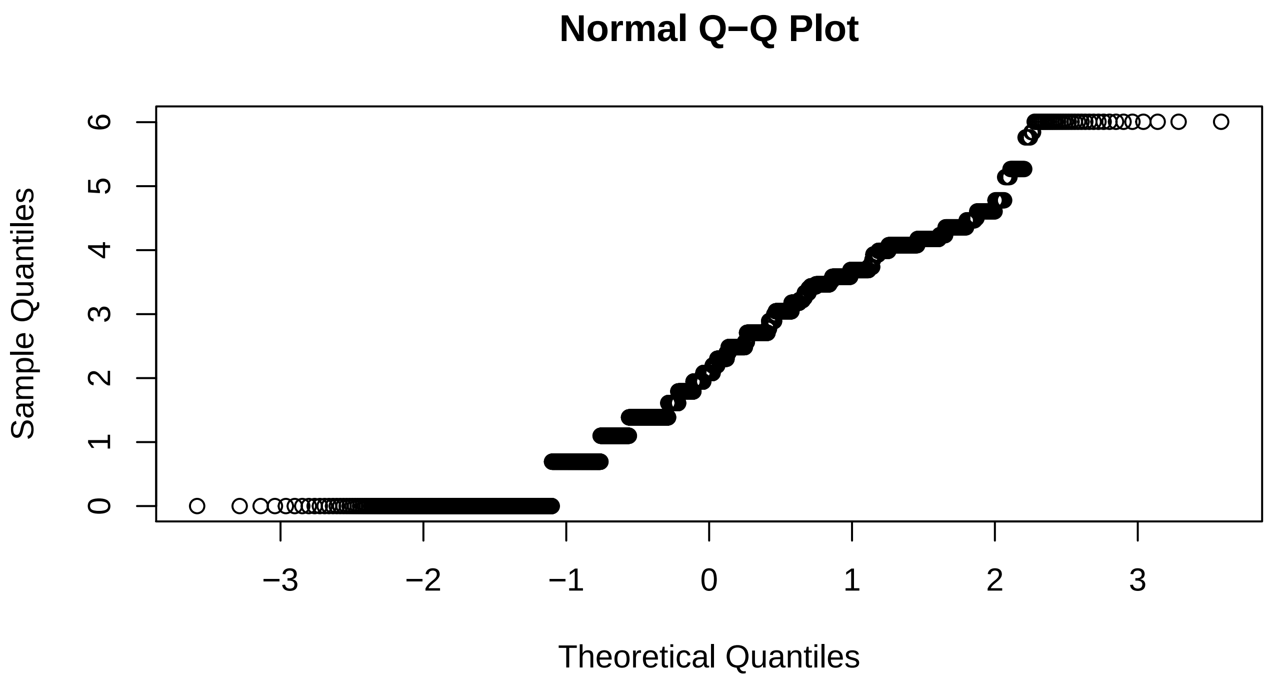
```
summary(D$project_average_rating)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00   14.00   39.00   36.93   59.00   60.00

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



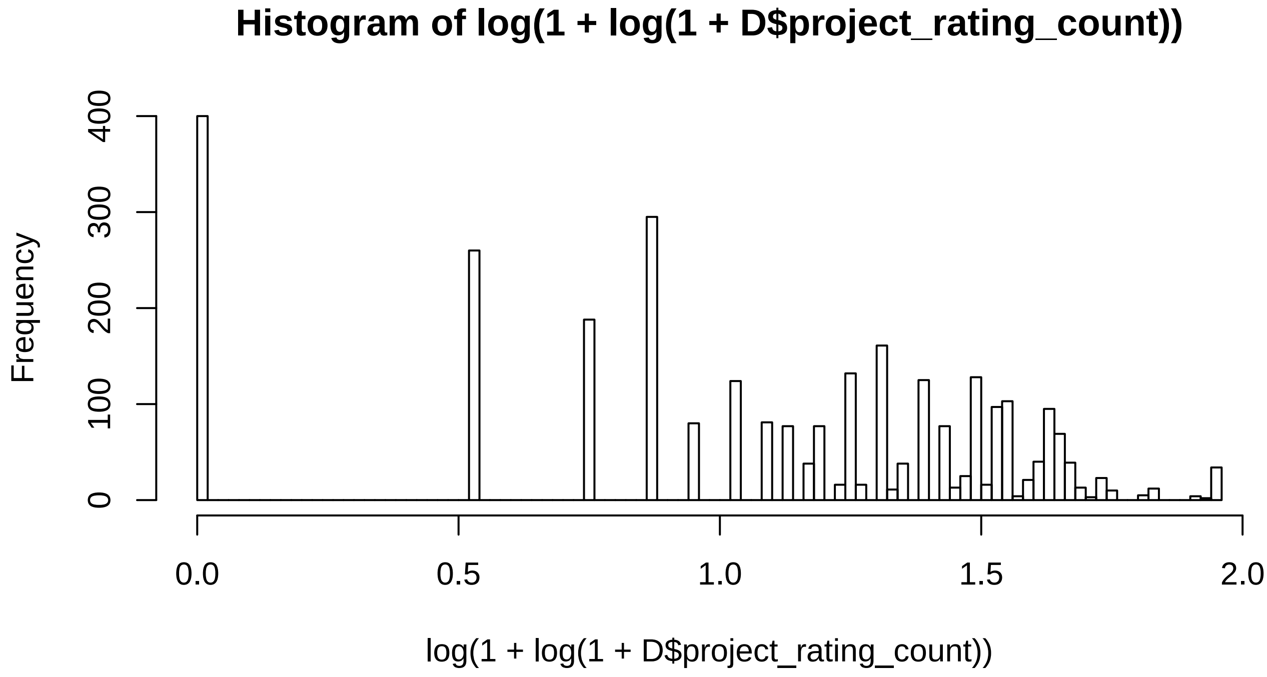
```
hist(log(1+D$project_rating_count), breaks=100)
```



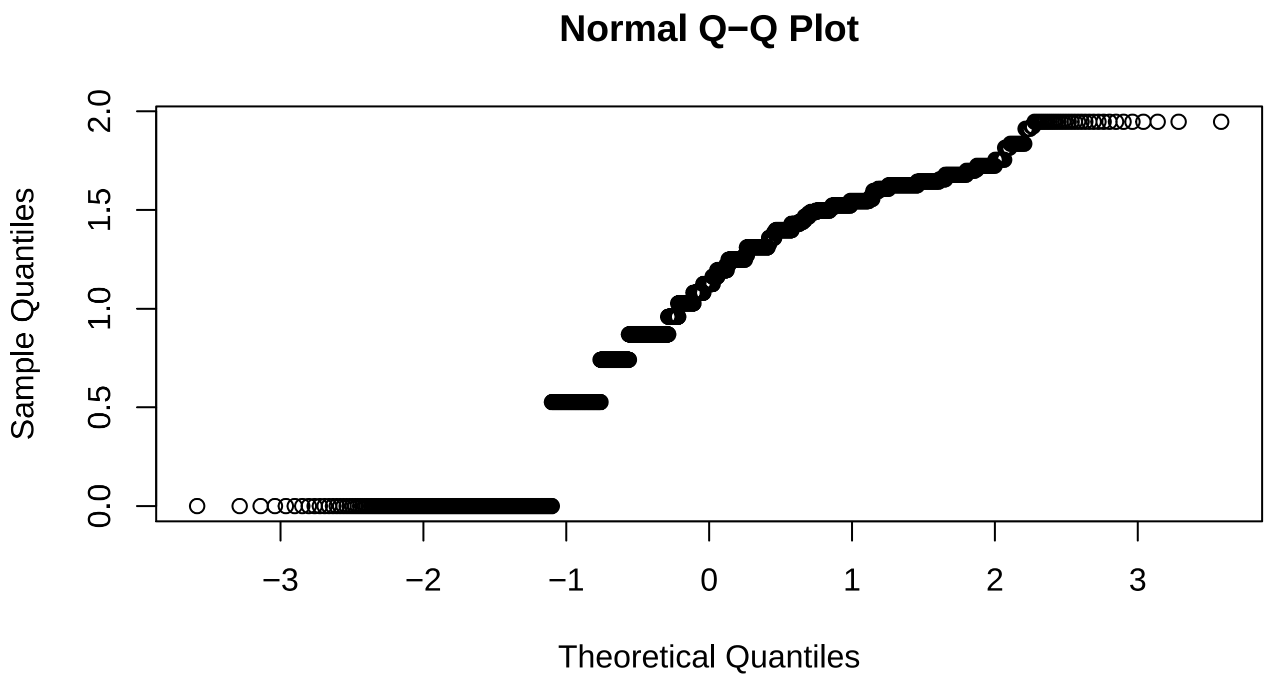
```
qqnorm(log(1+D$project_rating_count))
```



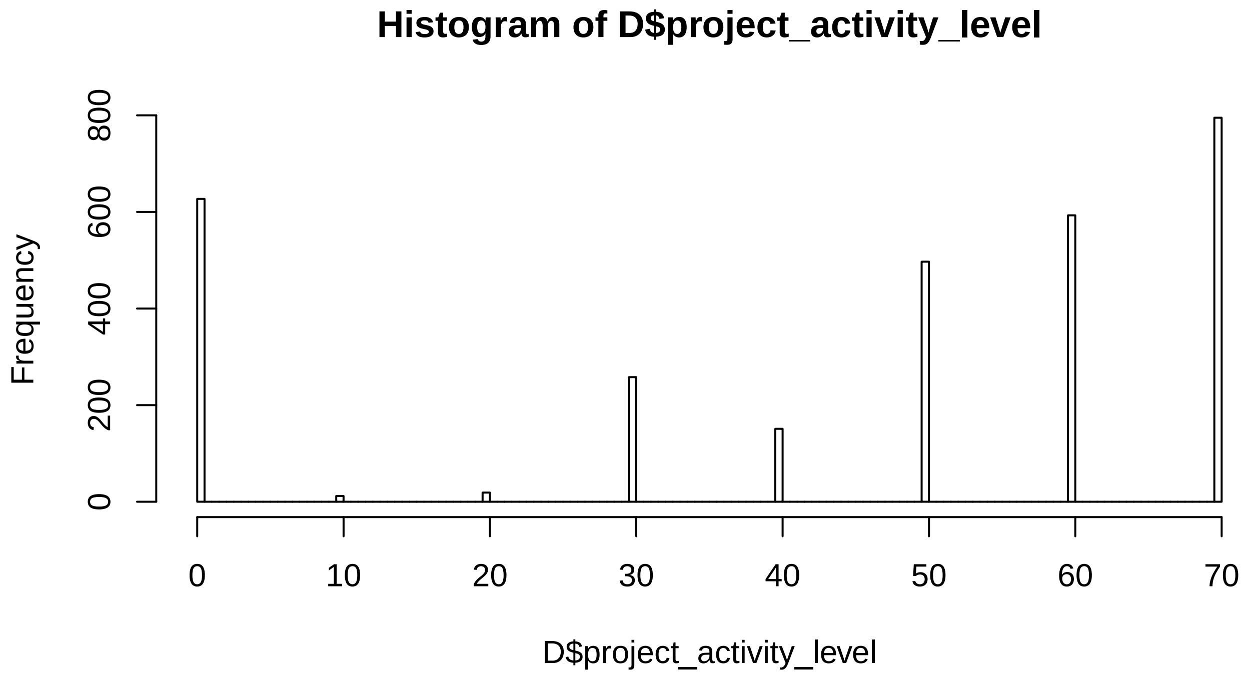
```
hist(log(1+log(1+D$project_rating_count)), breaks=100)
```



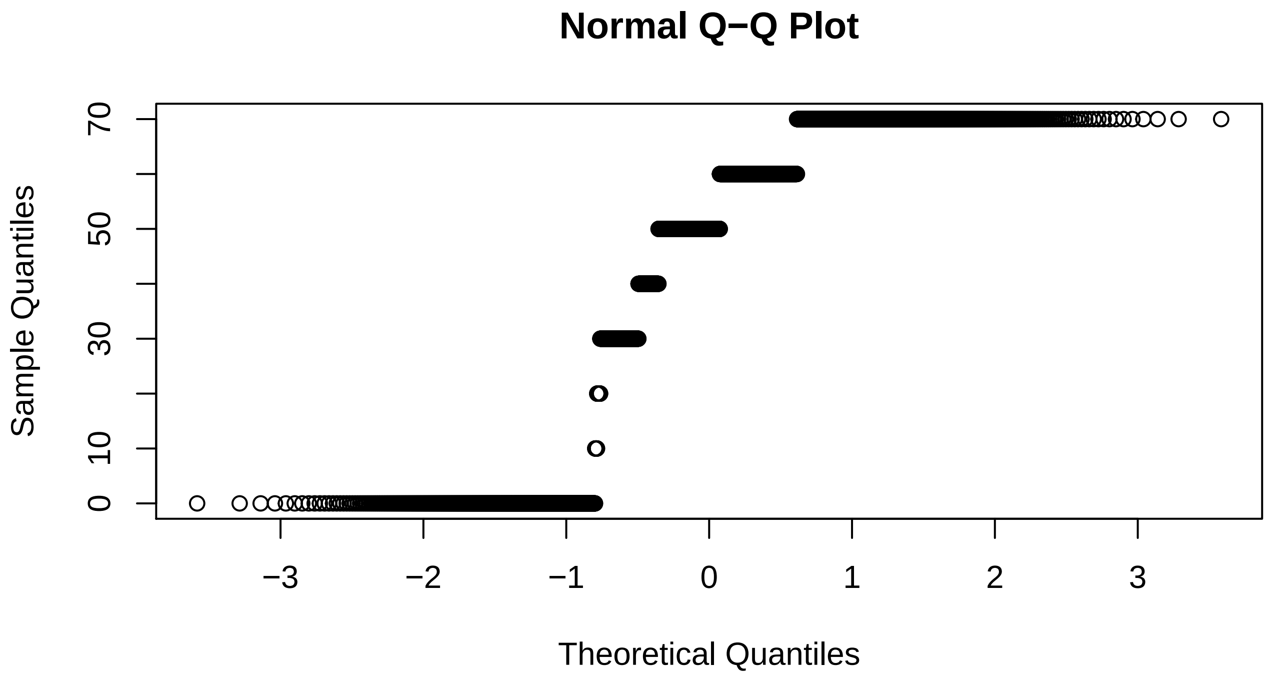
```
qqnorm(log(1+log(1+D$project_rating_count)))
```



```
summary(D$project_rating_count)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    2.00    7.00   22.55   27.00  405.00

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



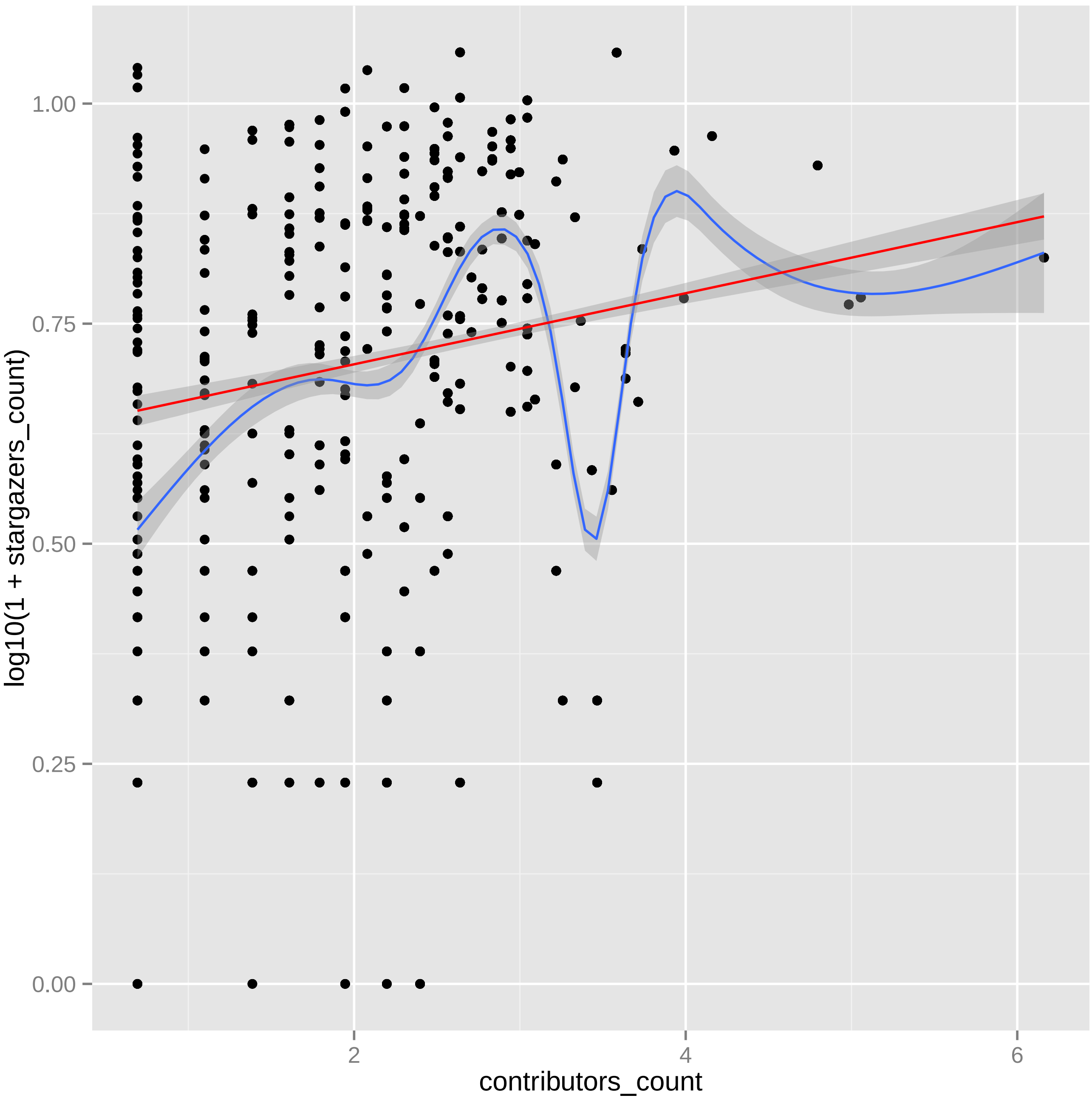
```
qqnorm(D$project_activity_level)
```



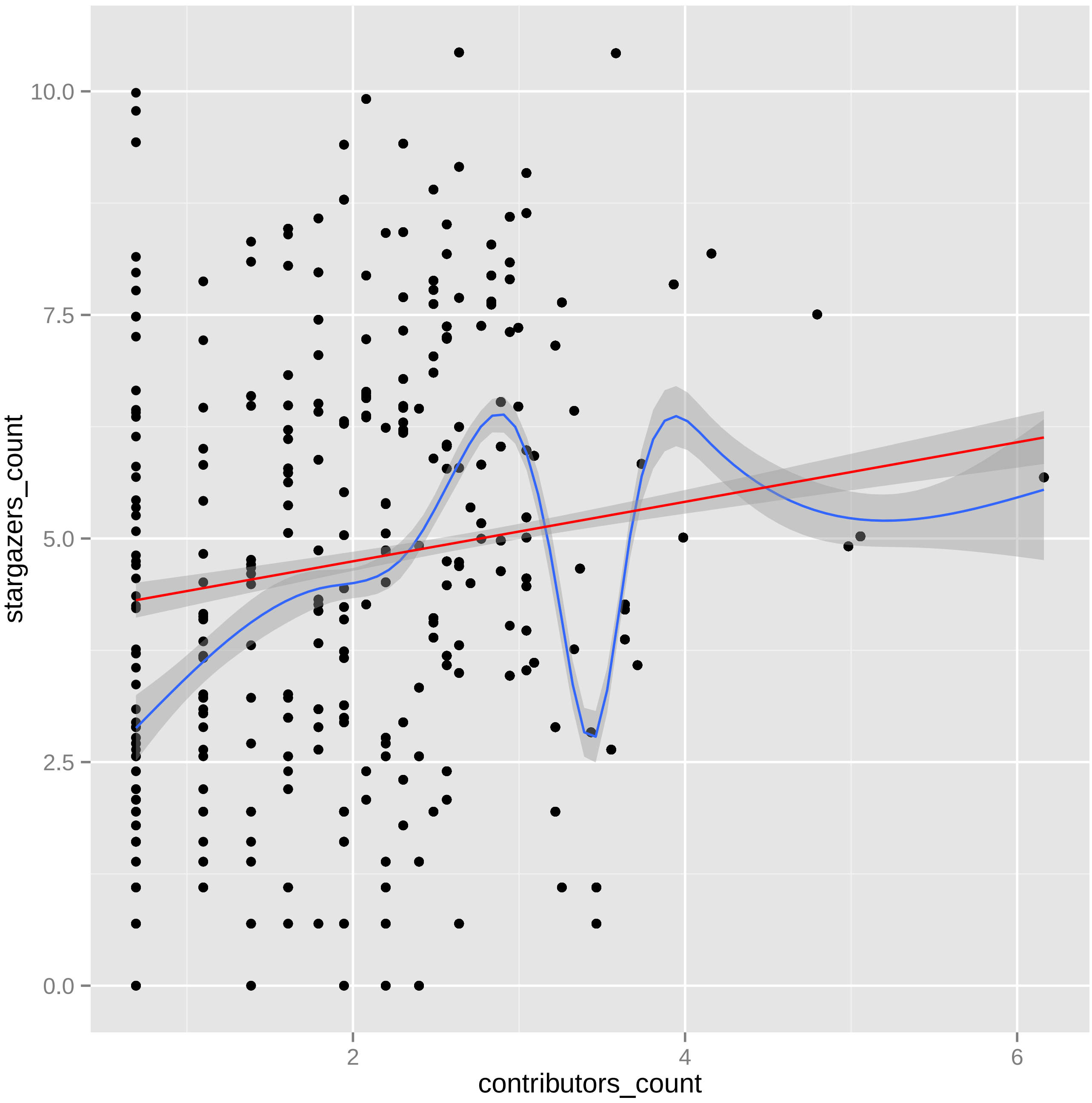
```
summary(D$project_activity_level)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   30.00   50.00   44.16   70.00   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)
```

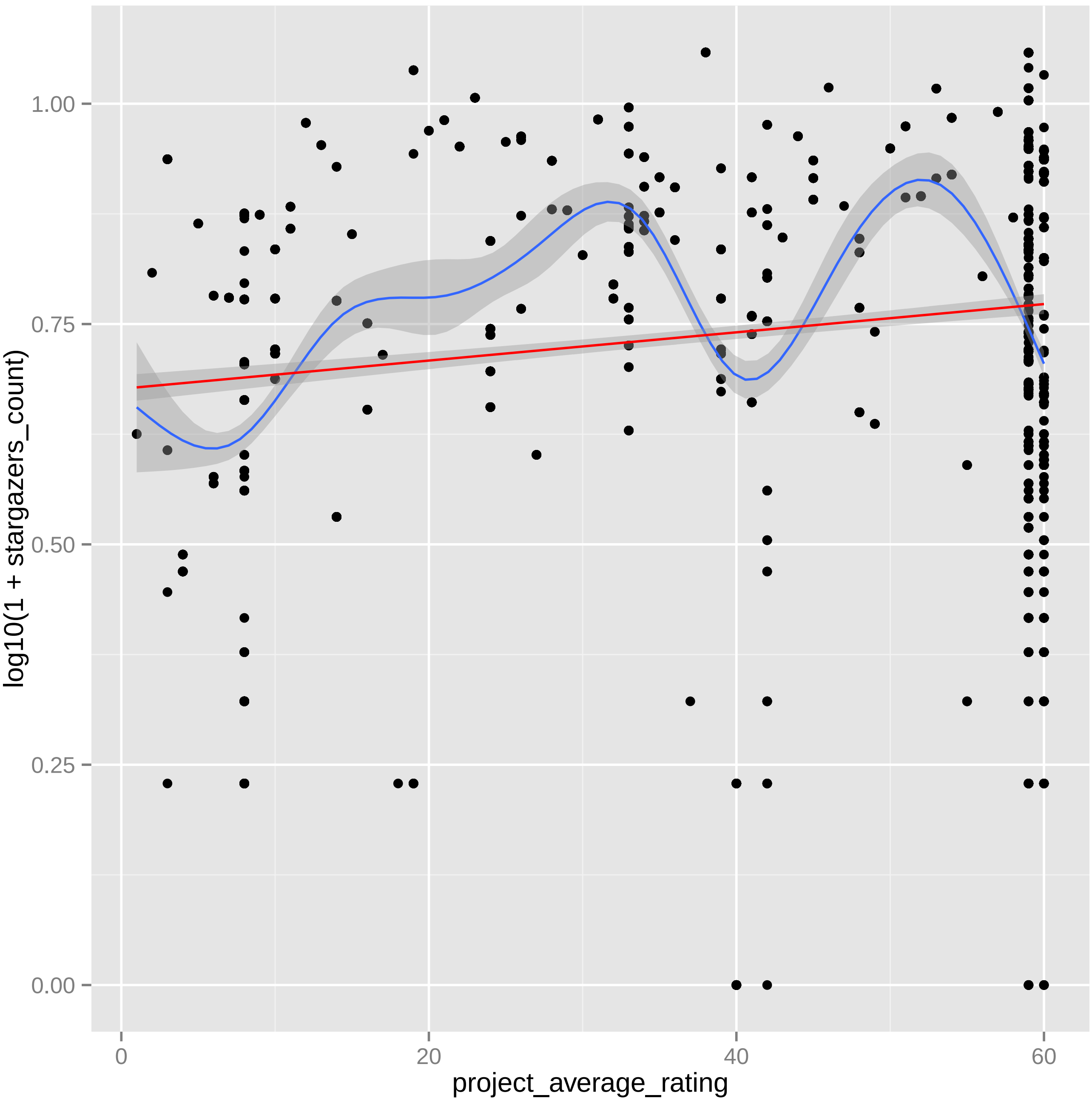
```
ggplot(D, aes(x=contributors_count, y=log10(1+stargazers_count))) + geom_point() + geom_smooth() + geom_smoot

## geom_smooth:  method="auto" and size of largest group is >=1000, so using gam with formula:  y ~ s(x,
bs = "cs").  Use 'method = x' to change the smoothing method.
```



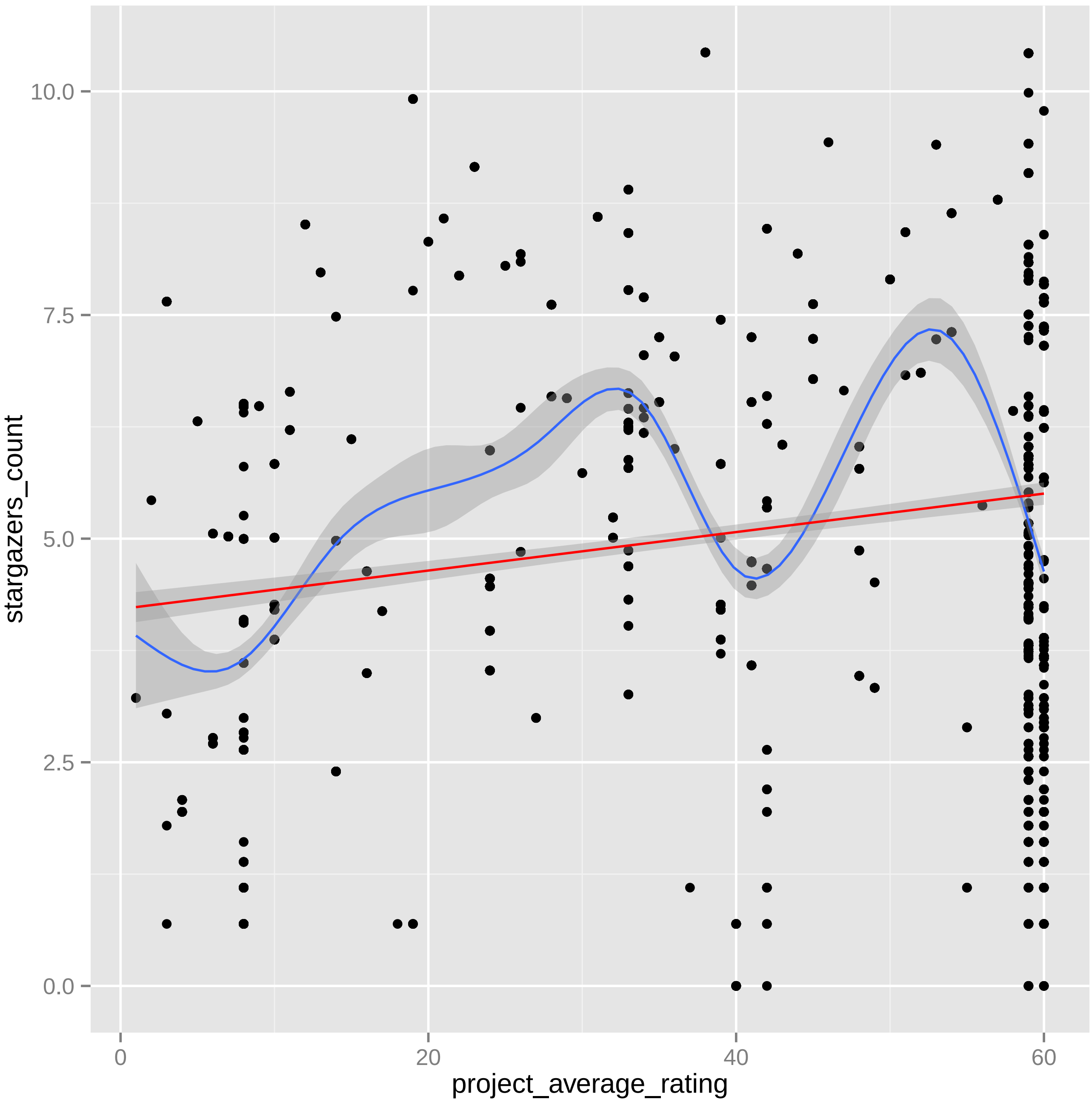
```
ggplot(D, aes(x=contributors_count, y=stargazers_count)) + geom_point() + geom_smooth() + geom_smooth(method=

## geom_smooth:  method="auto" and size of largest group is >=1000, so using gam with formula:  y ~ s(x,
bs = "cs").  Use 'method = x' to change the smoothing method.
```



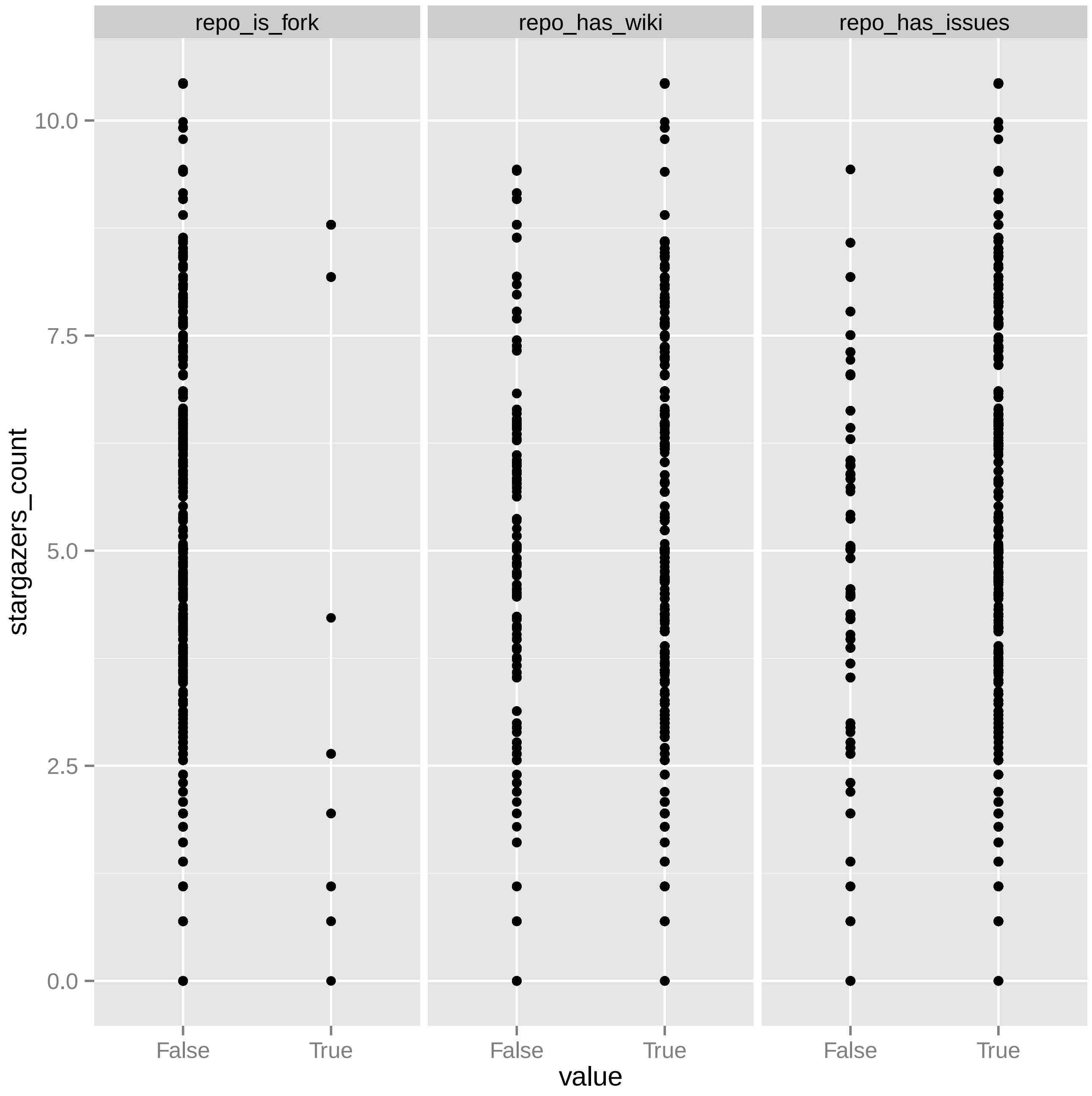
```
ggplot(D, aes(x=project_average_rating, y=log10(1+stargazers_count))) + geom_point() + geom_smooth() + geom_s

## geom_smooth:  method="auto" and size of largest group is >=1000, so using gam with formula:  y ~ s(x,
bs = "cs").  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 gam with formula:  y ~ s(x,
bs = "cs").  Use 'method = x' to change the smoothing method.
```

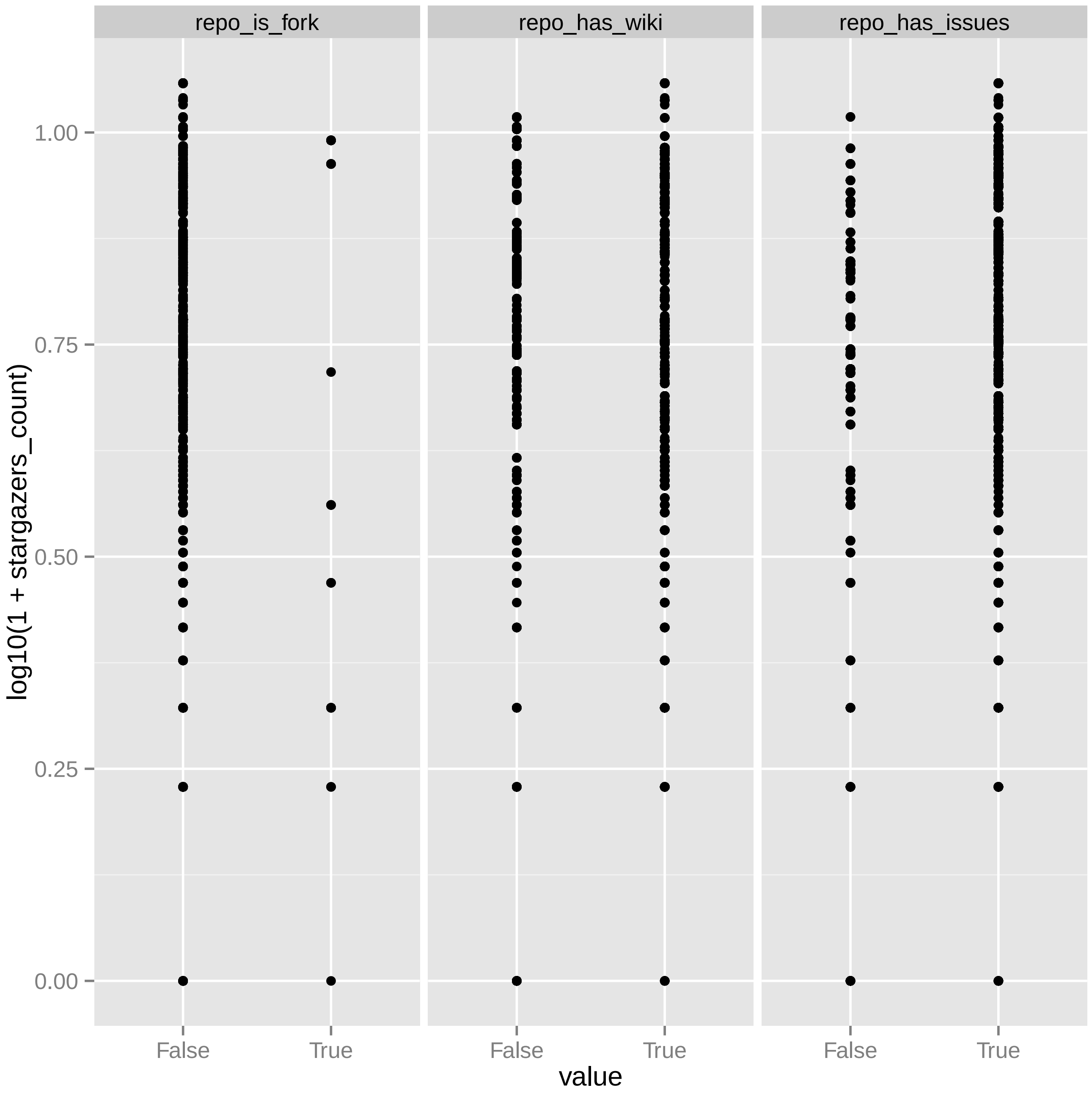


```
attrs <- c("repo_is_fork",
           "repo_has_wiki", "repo_has_issues")
d <- cbind(melt(D[,attrs], id.vars=c()), stargazers_count=D$stargazers_count)
ggplot(d,aes(x = value, y=stargazers_count)) +
       facet_wrap(~variable, scales = "free_x") +
       geom_point() + geom_smooth() + geom_smooth(method=lm,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

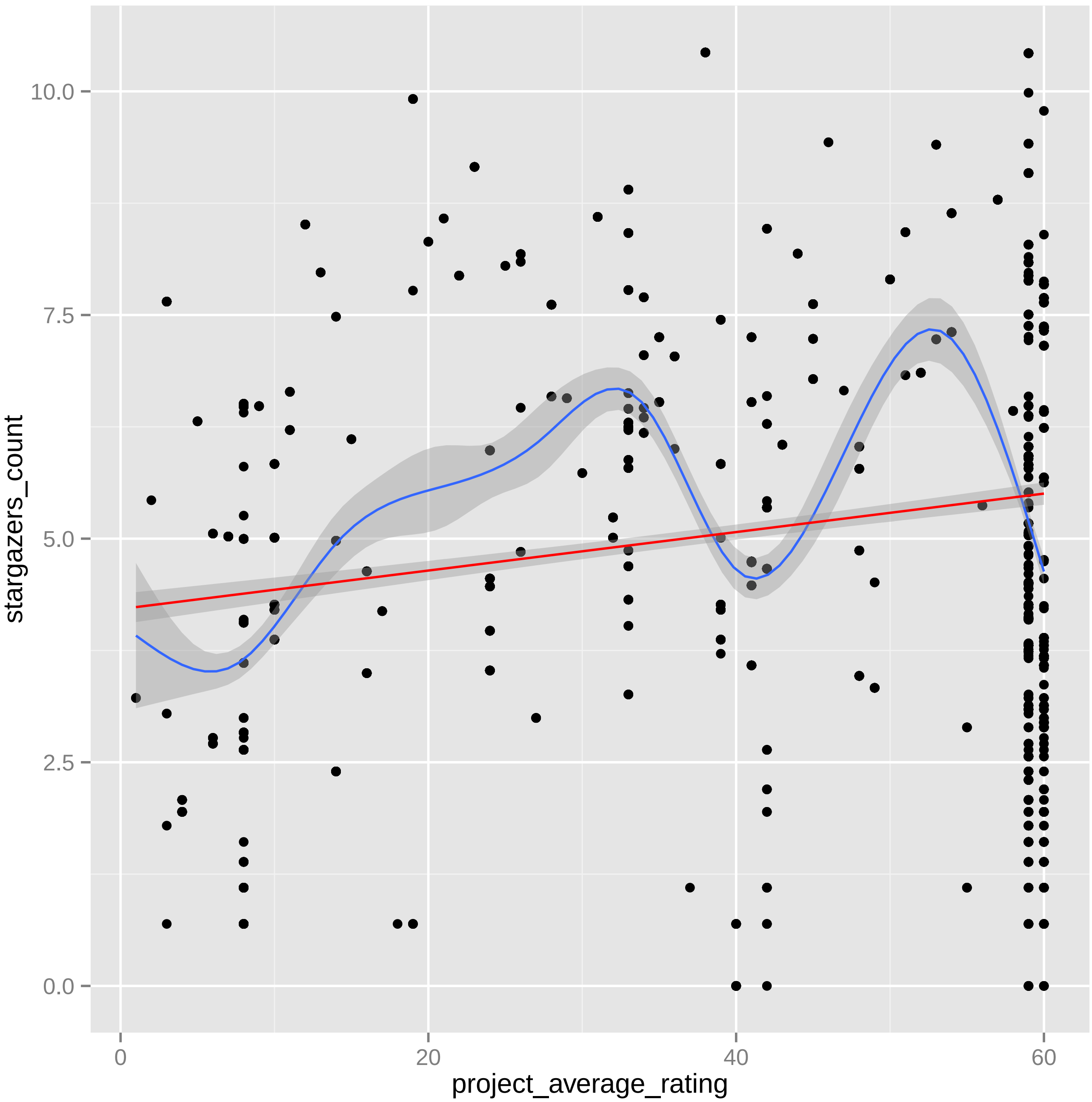


```
ggplot(d,aes(x = value, y=log10(1+stargazers_count))) +
       facet_wrap(~variable, scales = "free_x") +
       geom_point() + geom_smooth() + geom_smooth(method=lm, color="red")

## 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)?
```

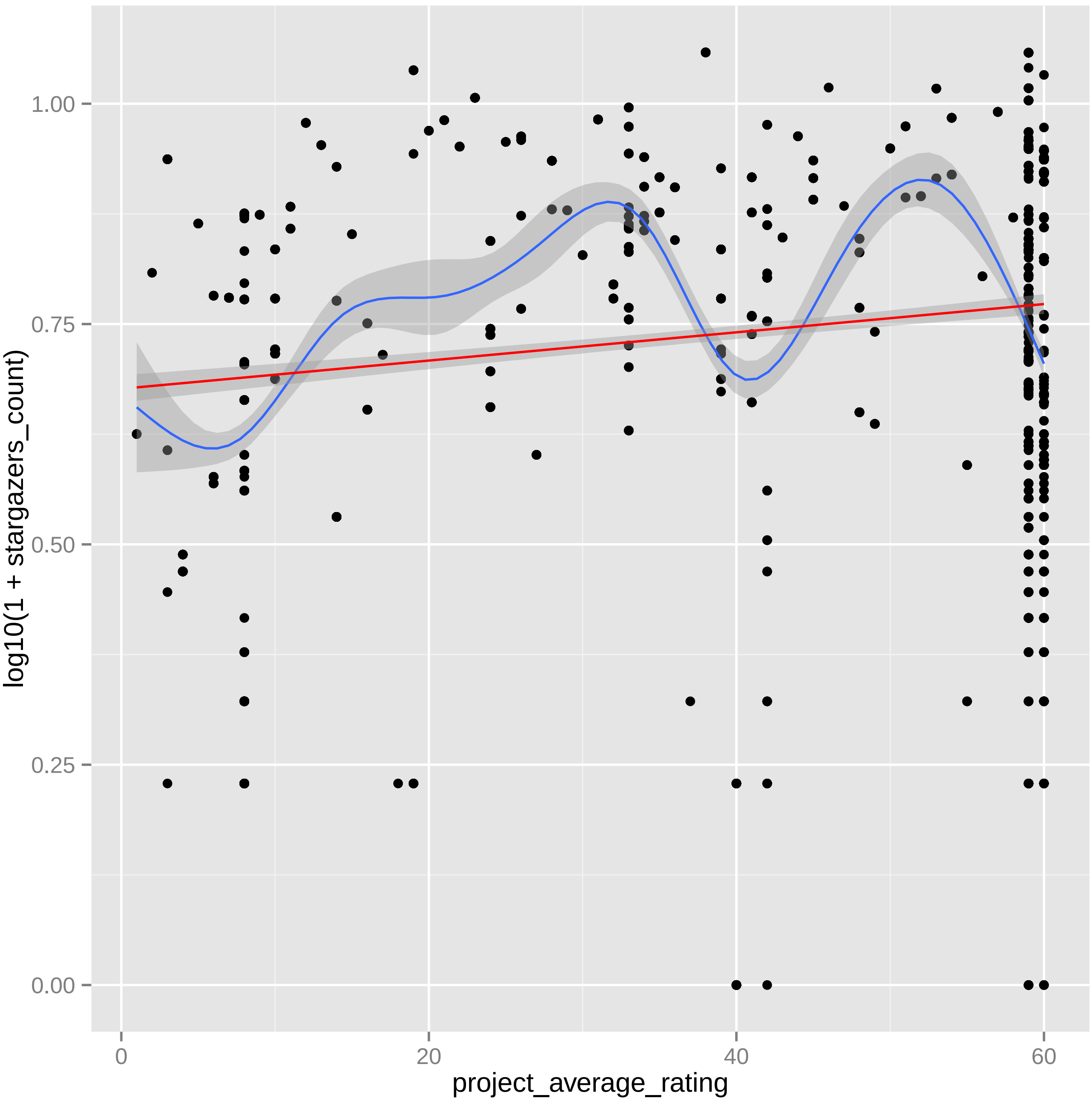
]



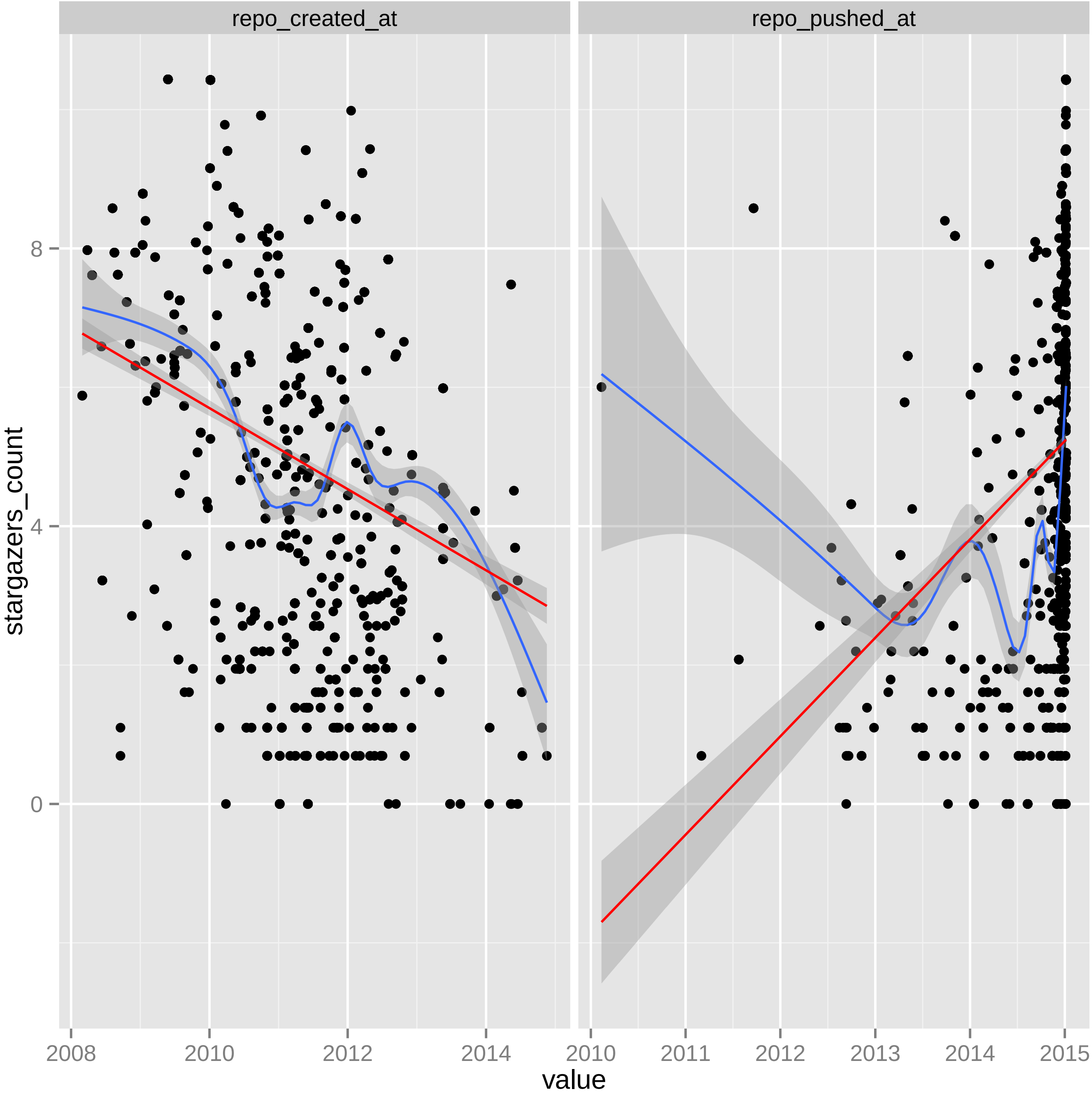
```
ggplot(D,aes(x = project_average_rating, y=stargazers_count)) +
       geom_point() + geom_smooth() + geom_smooth(method=lm, color="red")

## geom_smooth:  method="auto" and size of largest group is >=1000, so using gam with formula:  y ~ s(x,
bs = "cs").  Use 'method = x' to change the smoothing method.
```



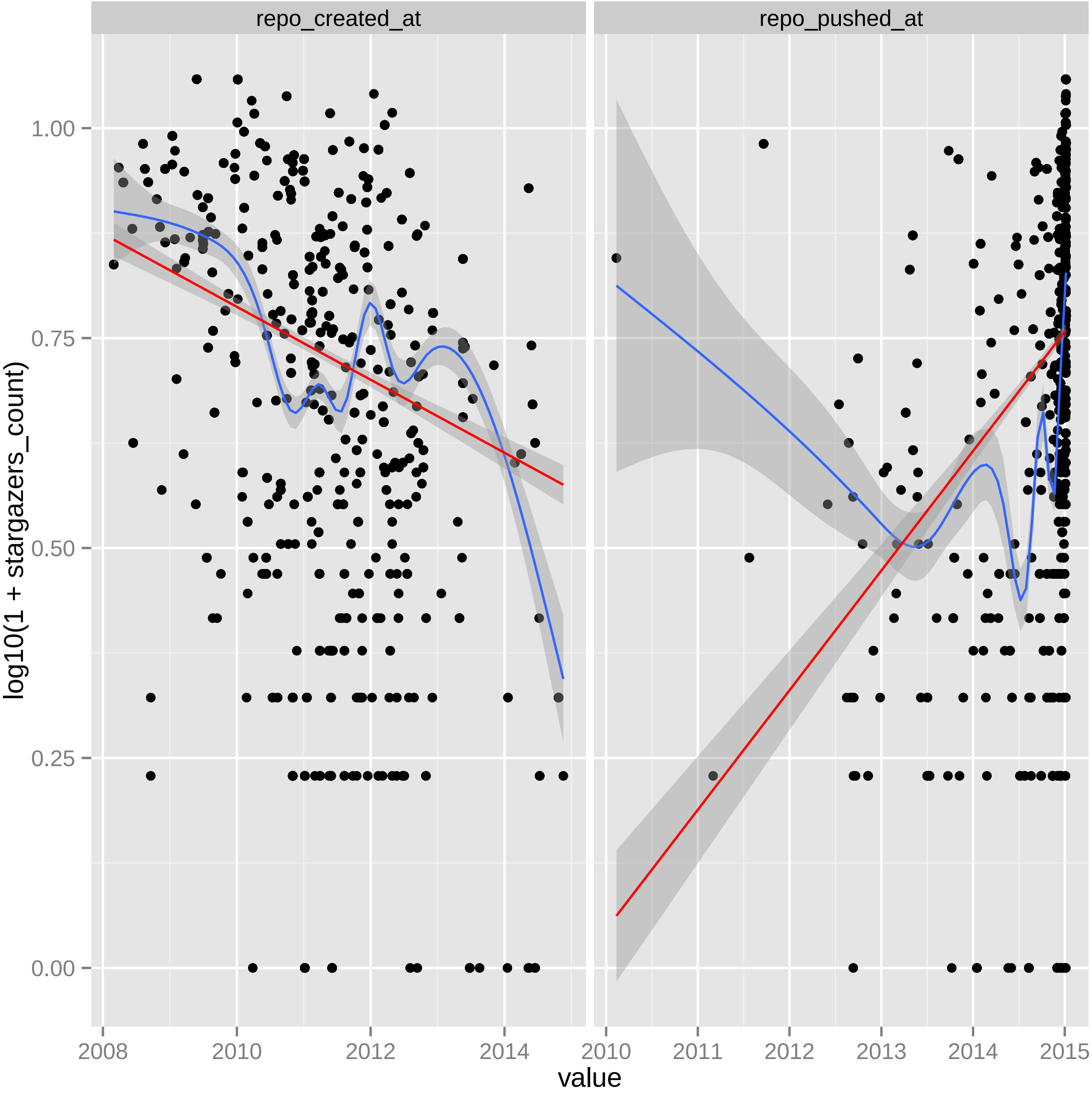
```
ggplot(D,aes(x = project_average_rating, y=log10(1+stargazers_count))) +
       geom_point() + geom_smooth() + geom_smooth(method=lm, color="red")

## geom_smooth:  method="auto" and size of largest group is >=1000, so using gam with formula:  y ~ s(x,
bs = "cs").  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 gam with formula:  y ~ s(x,
bs = "cs").  Use 'method = x' to change the smoothing method.
## geom_smooth:  method="auto" and size of largest group is >=1000, so using gam with formula:  y ~ s(x,
bs = "cs").  Use 'method = x' to change the smoothing method.
```



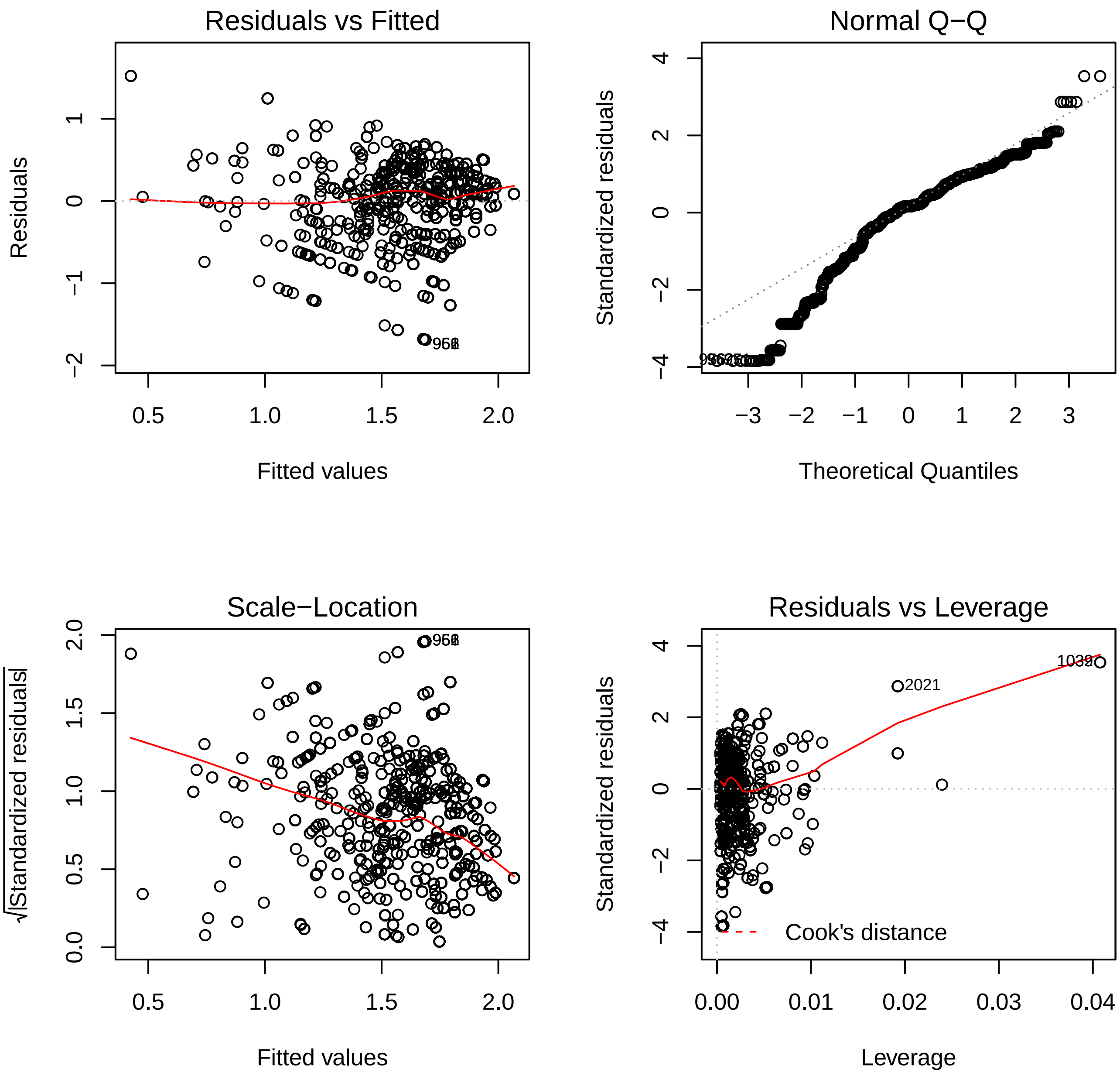
```
ggplot(d,aes(x = value, y=log10(1+stargazers_count))) +
       facet_wrap(~variable, scales = "free_x") +
       geom_point() + geom_smooth() + geom_smooth(method=lm, color="red")

## geom_smooth:  method="auto" and size of largest group is >=1000, so using gam with formula:  y ~ s(x,
bs = "cs").  Use 'method = x' to change the smoothing method.
## geom_smooth:  method="auto" and size of largest group is >=1000, so using gam with formula:  y ~ s(x,
bs = "cs").  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
## -1.68759 -0.16563  0.07233  0.31130  1.52133
##
```

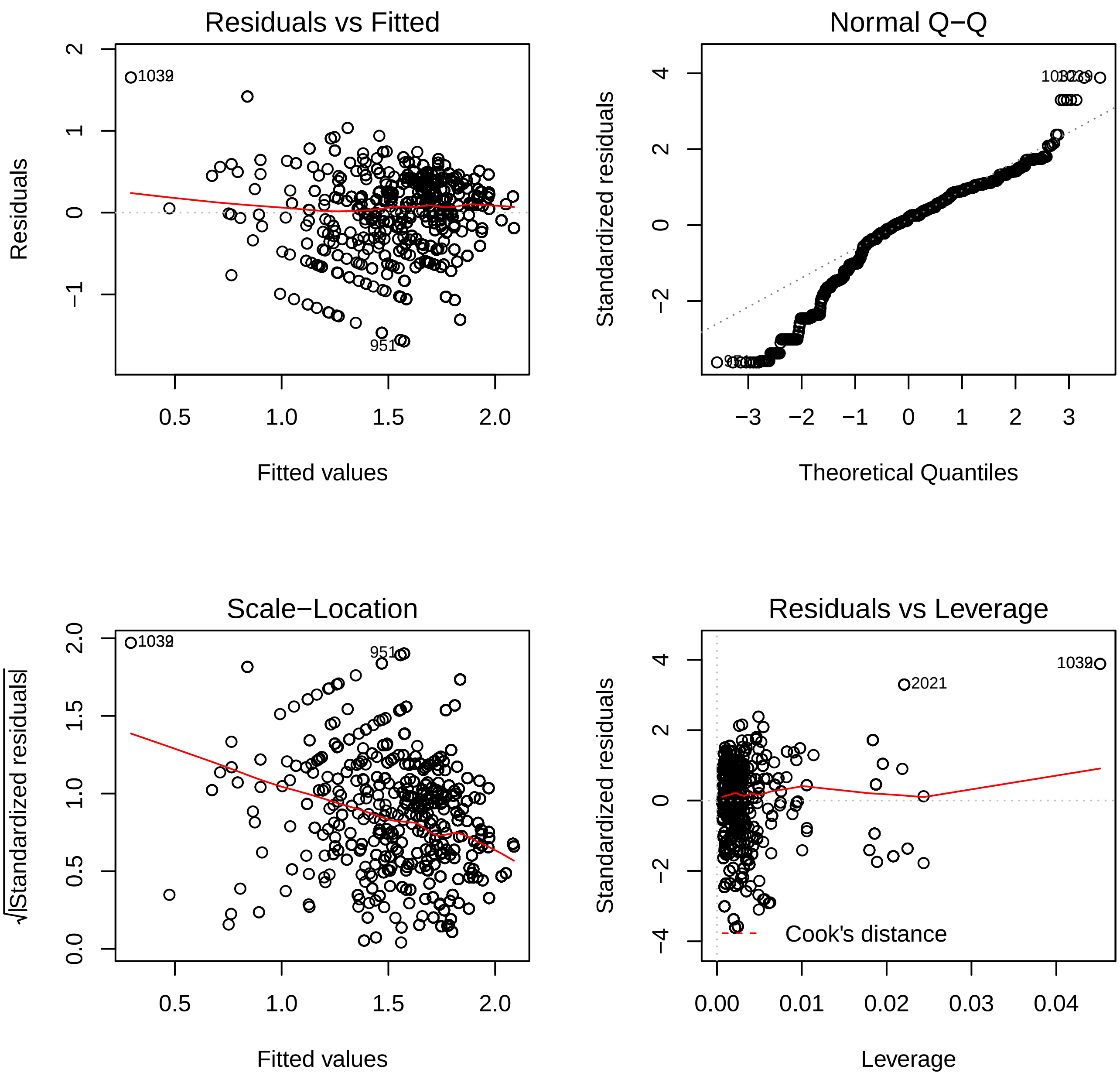
```
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -6.724e+00  8.975e-01  -7.492 8.93e-14 ***
## contributors_count  7.909e-02  8.534e-03   9.268  < 2e-16 ***
## repo_pushed_at      7.852e-04  5.233e-05  15.005  < 2e-16 ***
## repo_created_at    -3.100e-04  1.838e-05 -16.866  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4394 on 2948 degrees of freedom
## Multiple R-squared:  0.1805,Adjusted R-squared:  0.1797
## F-statistic: 216.4 on 3 and 2948 DF,  p-value: < 2.2e-16

plot(m)
```



```
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
## -1.57349 -0.16296  0.08774  0.28645  1.65279
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -7.138e+00  9.039e-01  -7.896 4.02e-15 ***
## contributors_count  1.083e-01  9.435e-03  11.481  < 2e-16 ***
## repo_pushed_at      7.611e-04  5.303e-05  14.351  < 2e-16 ***
## repo_created_at    -2.679e-04  1.917e-05 -13.976  < 2e-16 ***
## repo_is_forkTrue    1.734e-01  5.806e-02   2.987  0.00284 **
## repo_has_wikiTrue  -3.661e-02  1.855e-02  -1.974  0.04844 *
## repo_has_issuesTrue 1.580e-01  2.218e-02   7.122 1.33e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4356 on 2945 degrees of freedom
## Multiple R-squared:  0.1955,Adjusted R-squared:  0.1939
## F-statistic: 119.3 on 6 and 2945 DF,  p-value: < 2.2e-16
```

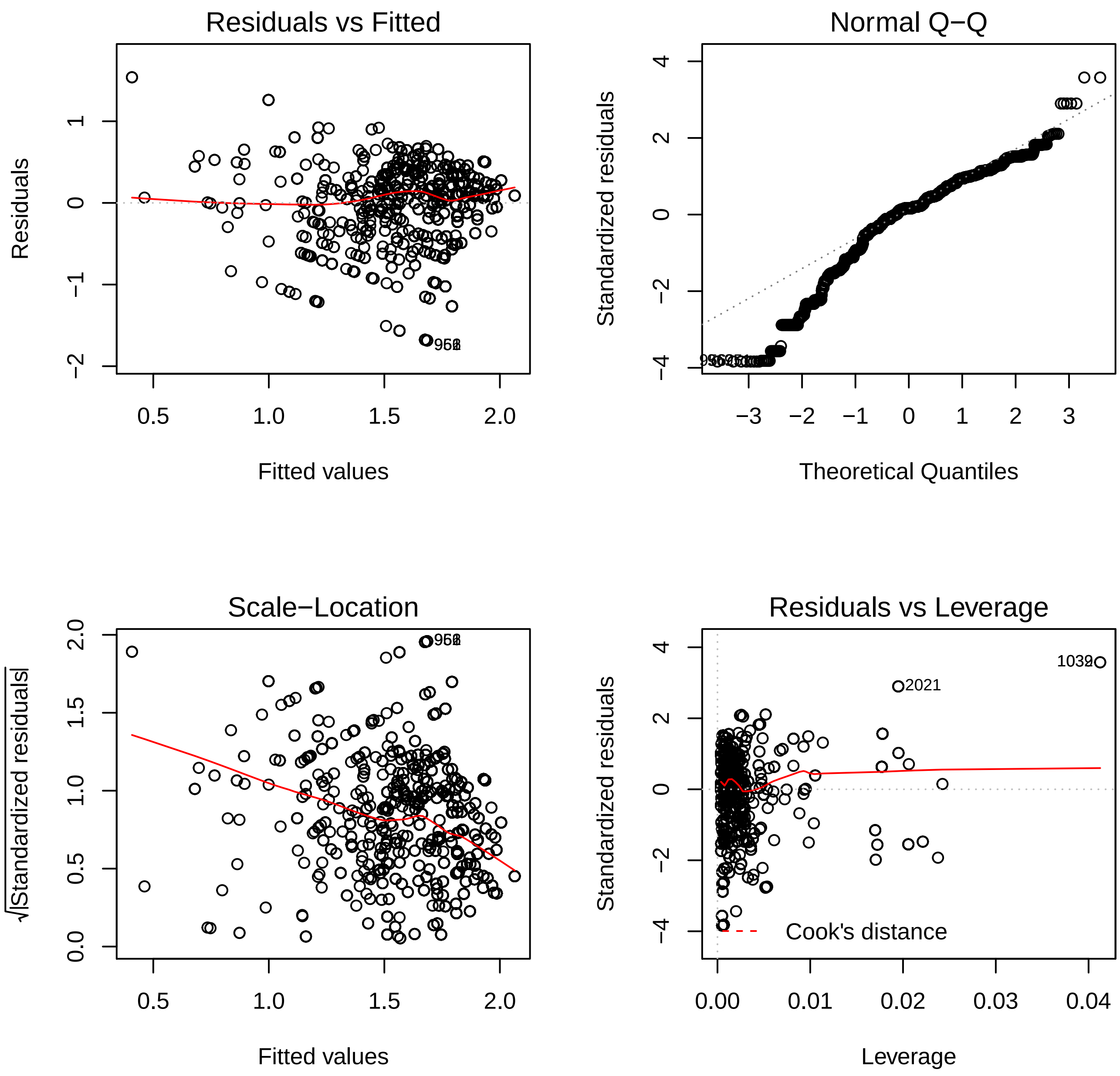
```
plot(m2)
```



```
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   2948 569.23
## 2   2945 558.81  3    10.421 18.306 9.127e-12 ***
## ---
## 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
## -1.68505 -0.16469  0.07116  0.29698  1.53868
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -6.869e+00  9.007e-01  -7.626 3.25e-14 ***
## contributors_count  7.999e-02  8.545e-03   9.361  < 2e-16 ***
## repo_pushed_at      7.926e-04  5.247e-05  15.106  < 2e-16 ***
## repo_created_at    -3.087e-04  1.839e-05 -16.791  < 2e-16 ***
## repo_is_forkTrue    1.046e-01  5.774e-02   1.812   0.0701 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4393 on 2947 degrees of freedom
## Multiple R-squared:  0.1814,Adjusted R-squared:  0.1803
## F-statistic: 163.3 on 4 and 2947 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   2948 569.23
## 2   2947 568.60  1   0.63359 3.2838 0.07007 .
## ---
## 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")
```