```
## 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 web technologies - data analysis

WikiTeams.pl

11 January 2015

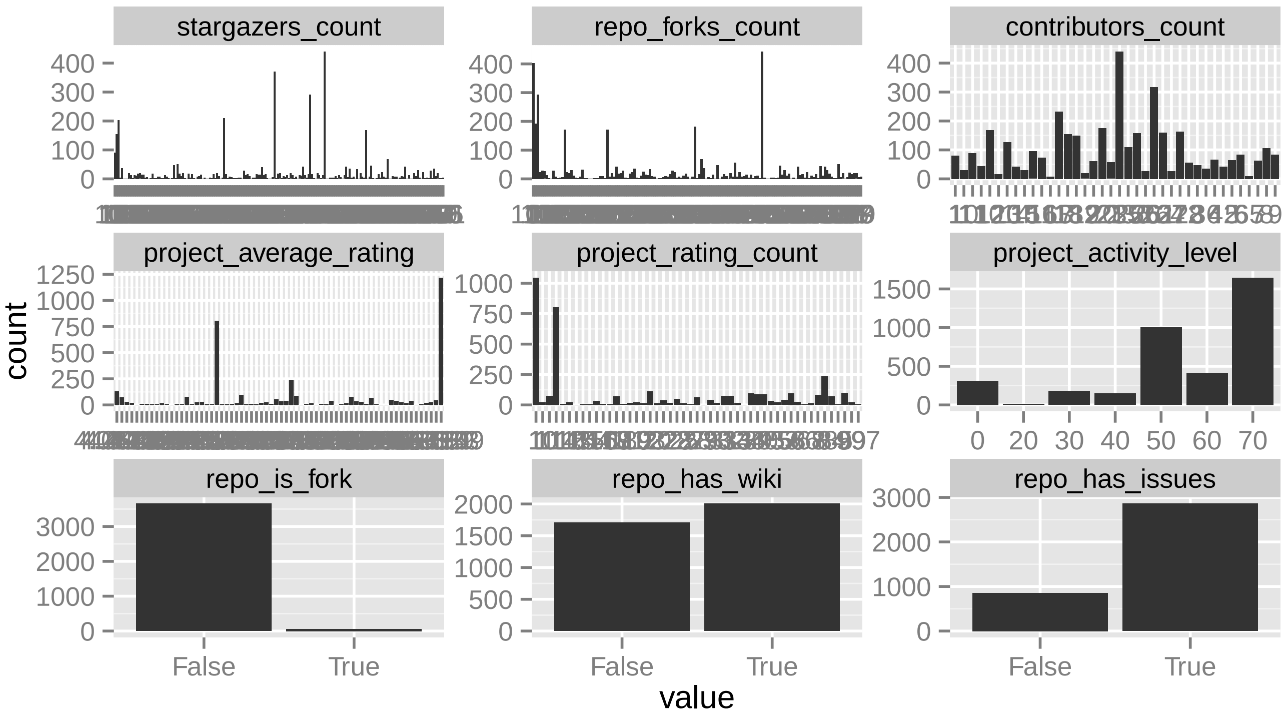
```
options("warn" = -1)
```

1 Read in the data

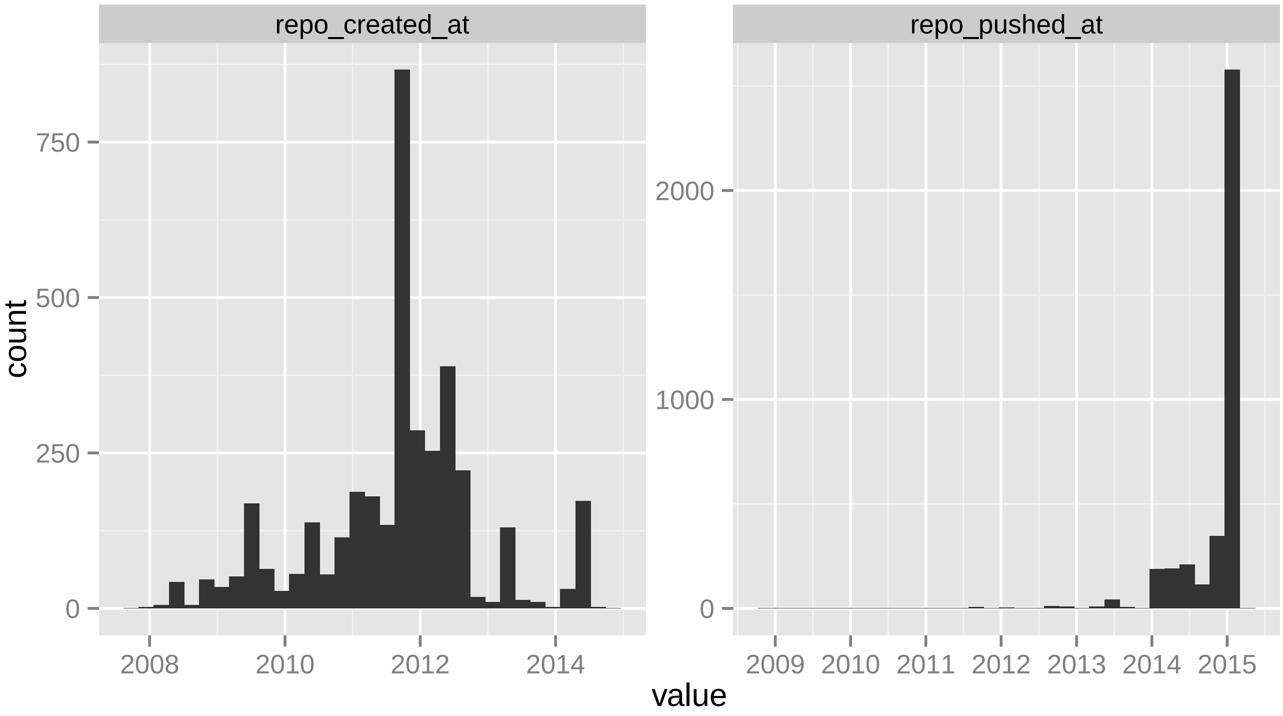
```
D <- read.table("../results_web.csv", sep=";", quote = "\"", header=T)
names(D)

##  [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 3712 recods.

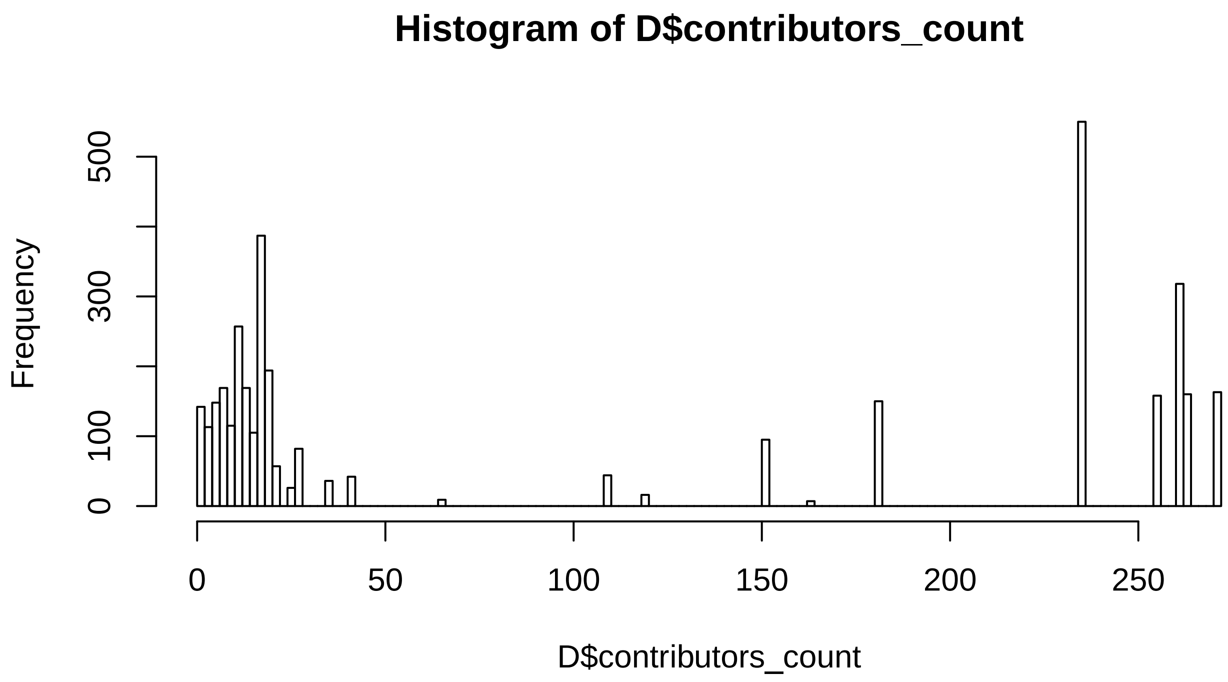
```
# discrete
plot_mhist(D, attrs=c("stargazers_count", "repo_forks_count", "contributors_count", "project_average_rating",
```



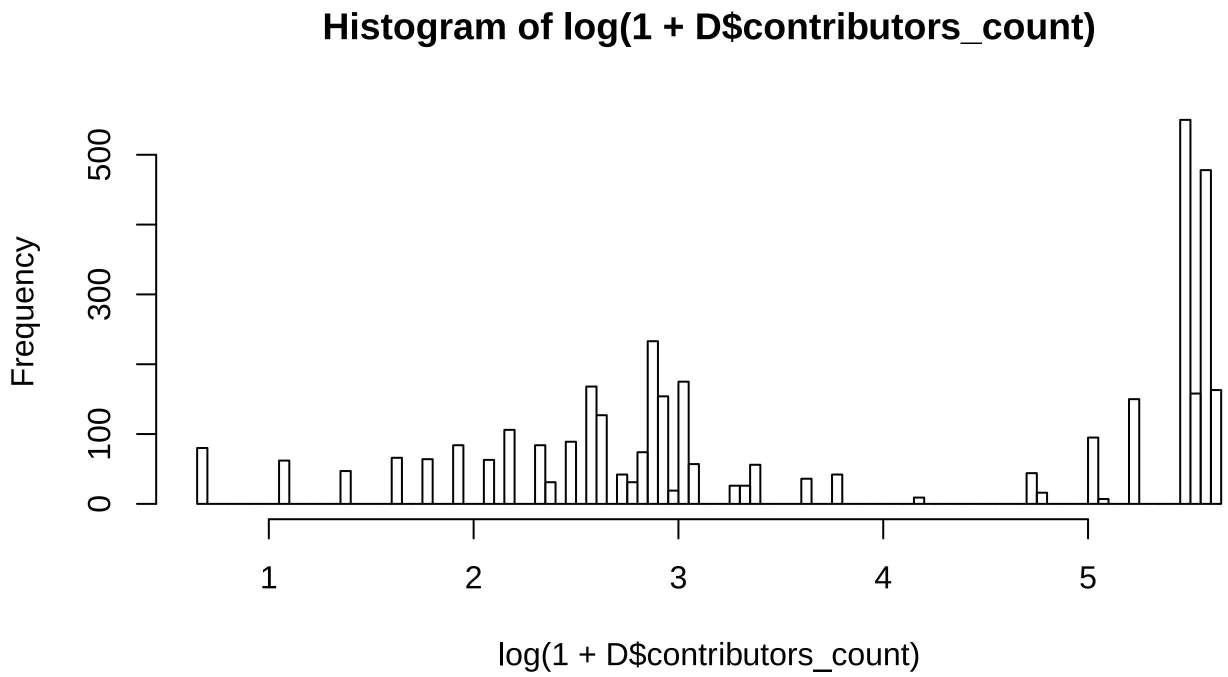
```r
# continuous
plot_mhist(D, attrs=c("repo_created_at", "repo_pushed_at"), date.values = T)

## 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.
```



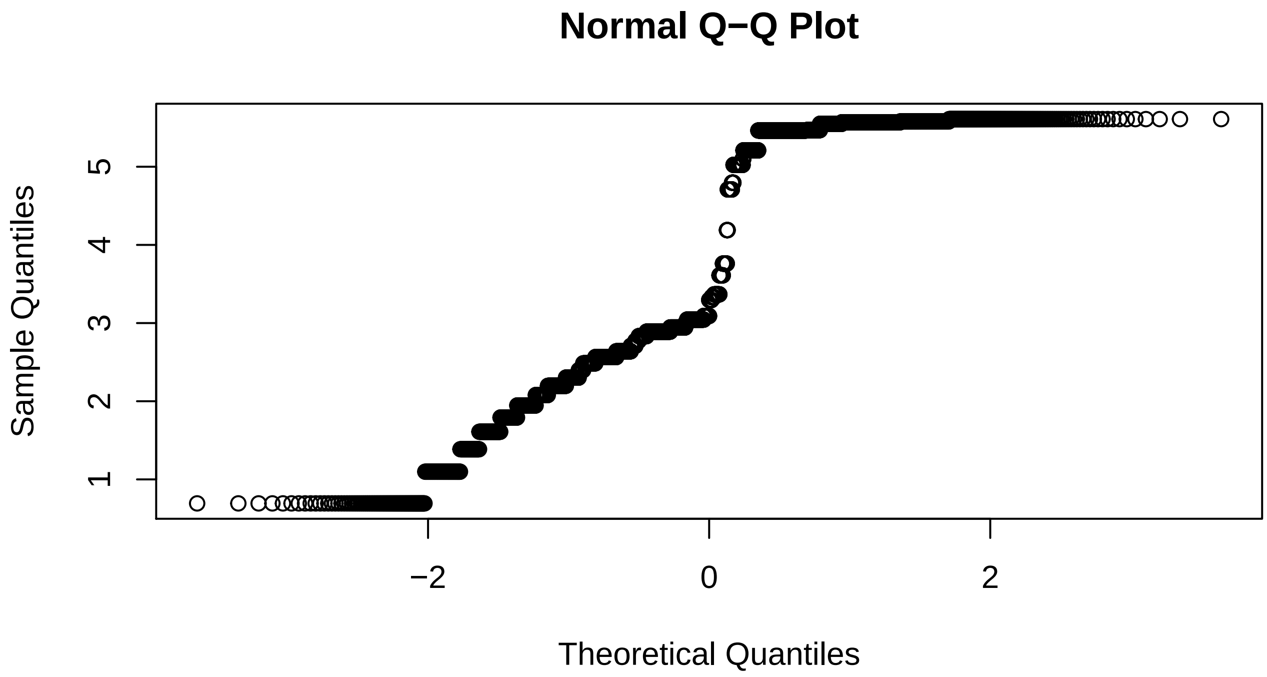
```r
# contrib count
hist(D$contributors_count, breaks=100)
```



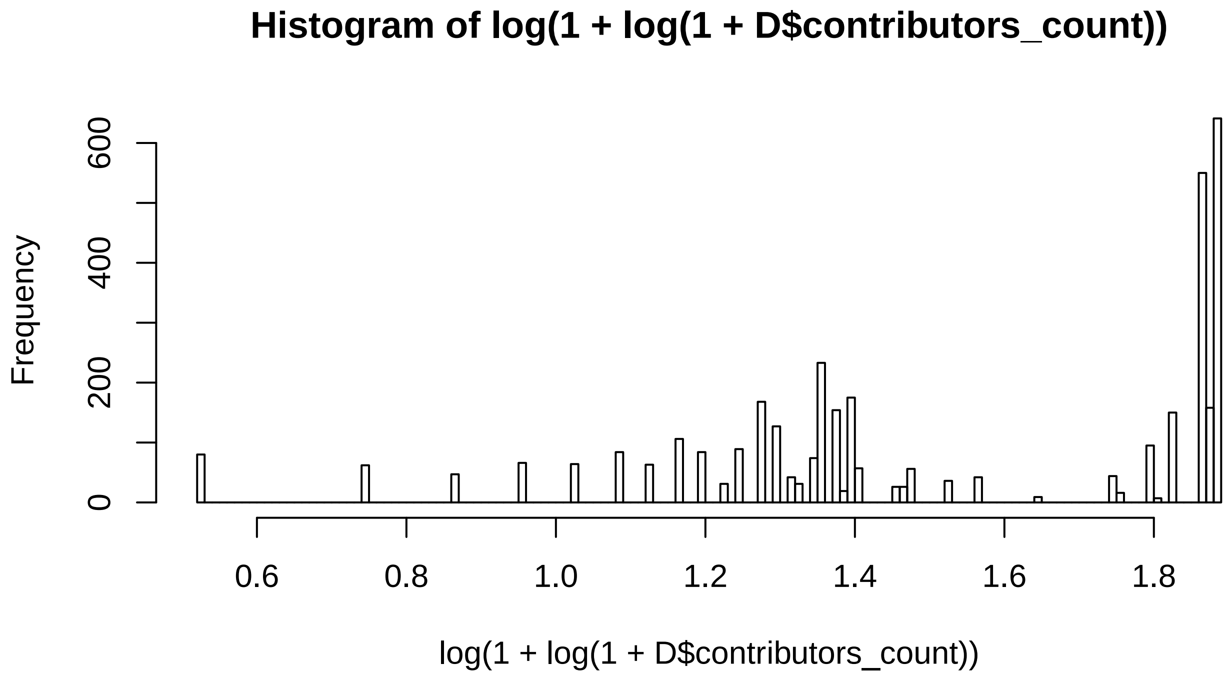
```
summary(D$contributors_count, breaks=100)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0    12.0    23.5   112.6   235.0   272.0

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



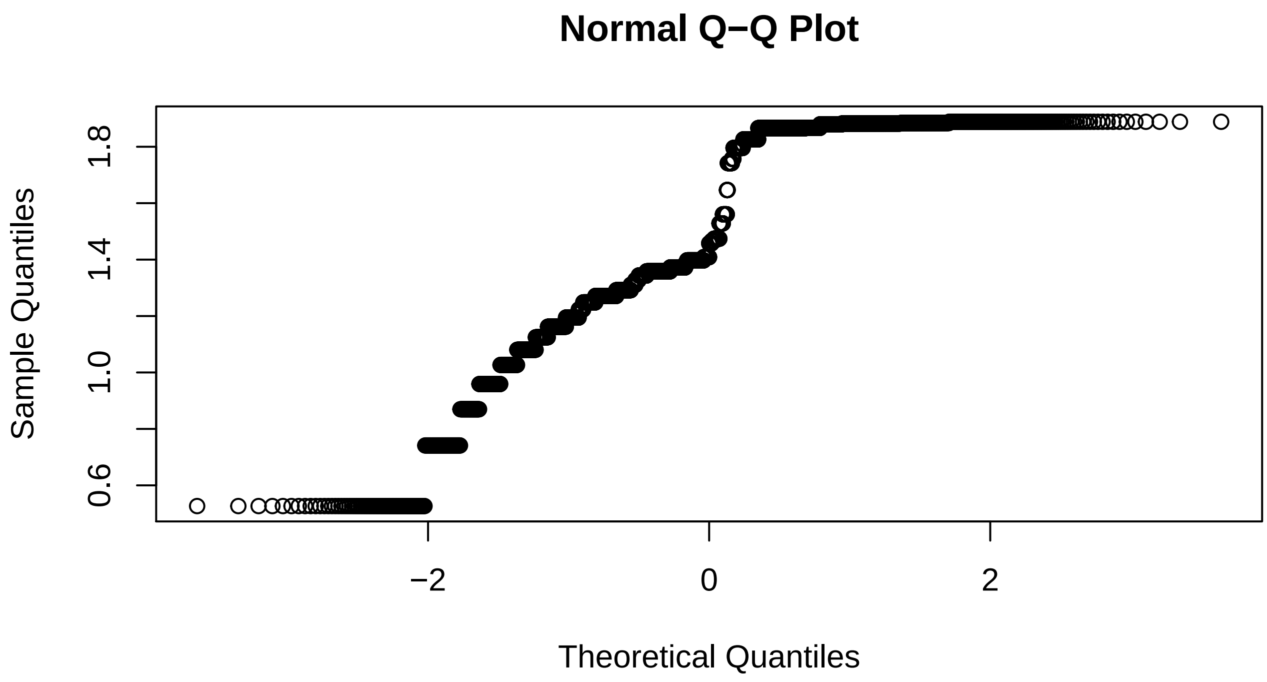
```
qqnorm(log(1+D$contributors_count))
```



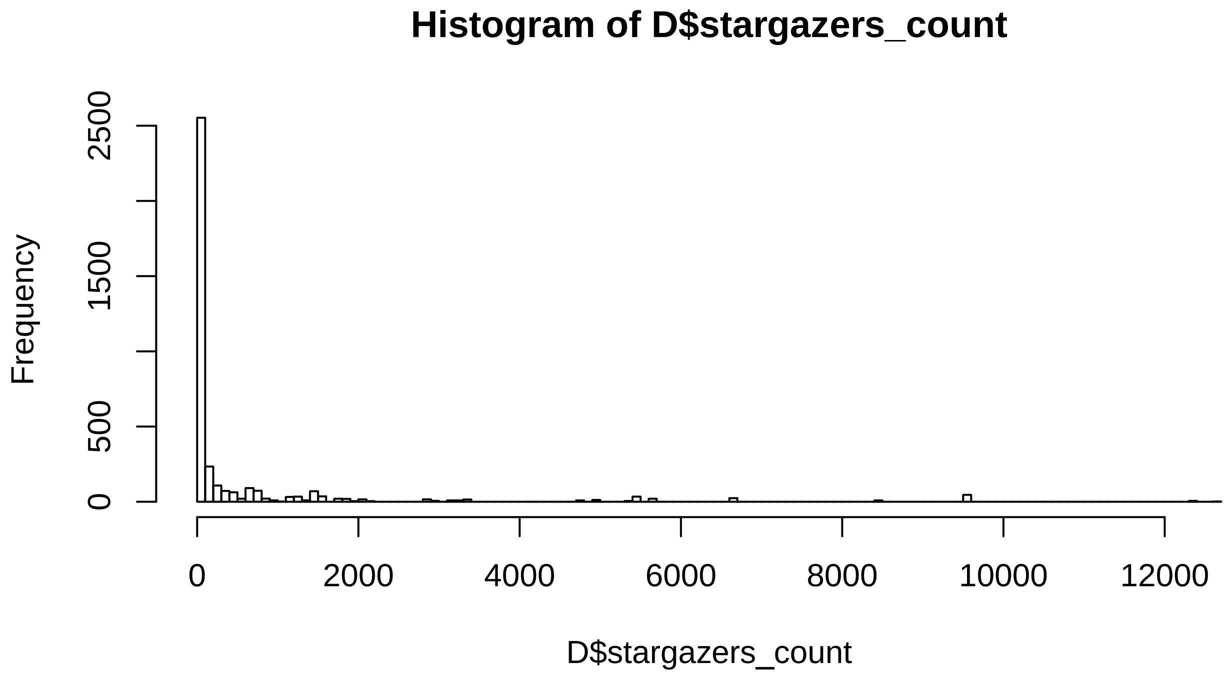
```
hist(log(1+log(1+D$contributors_count)), breaks=100)
```



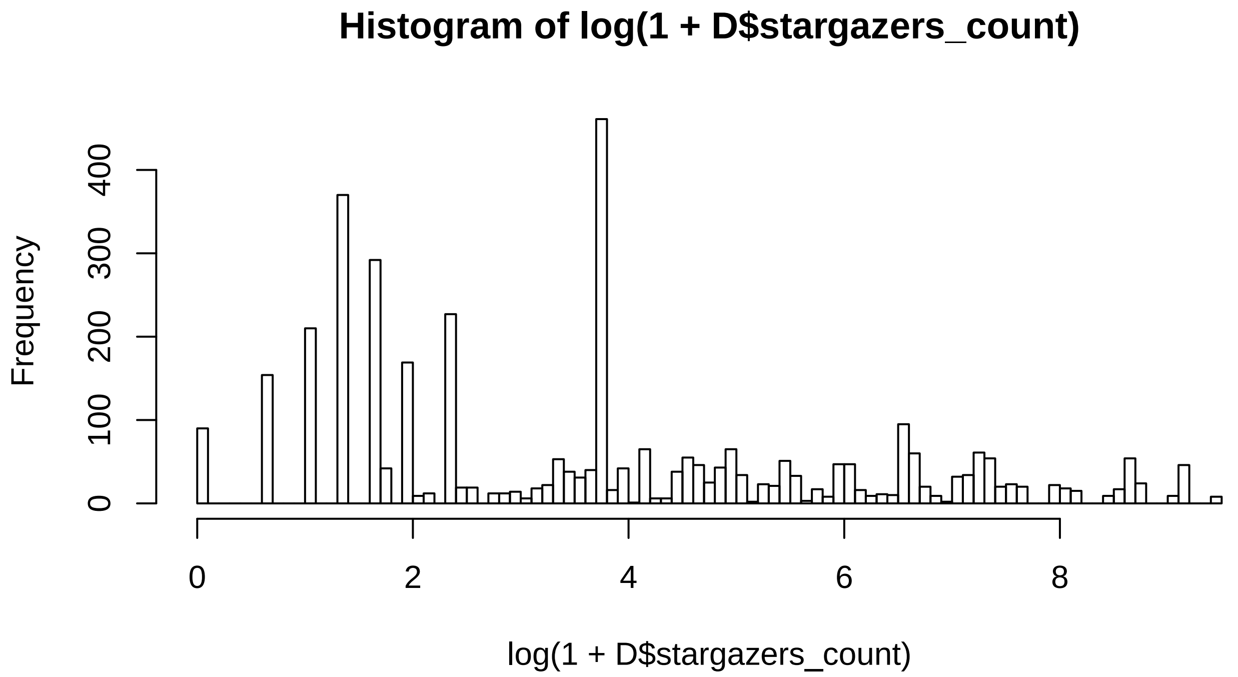
```
qqnorm(log(1+log(1+D$contributors_count)))
```



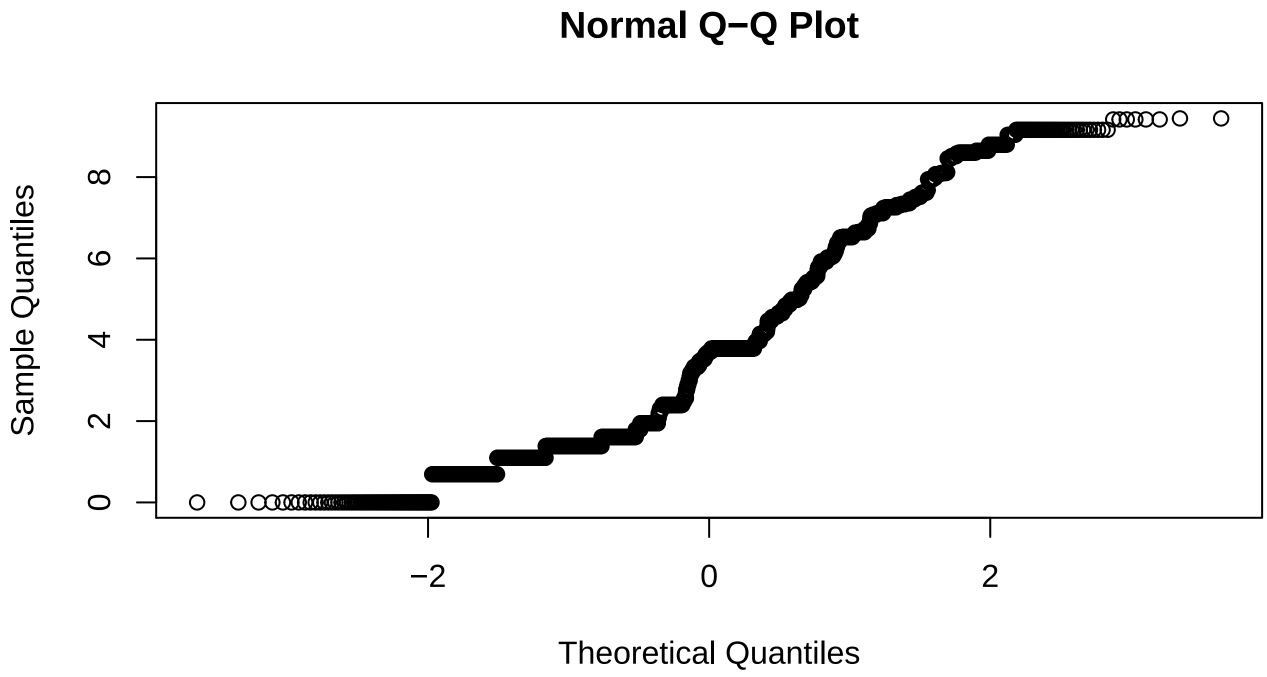
```
summary(log(1+D$contributors_count), breaks=100)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.6931  2.5650  3.1930  3.8310  5.4640  5.6090

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



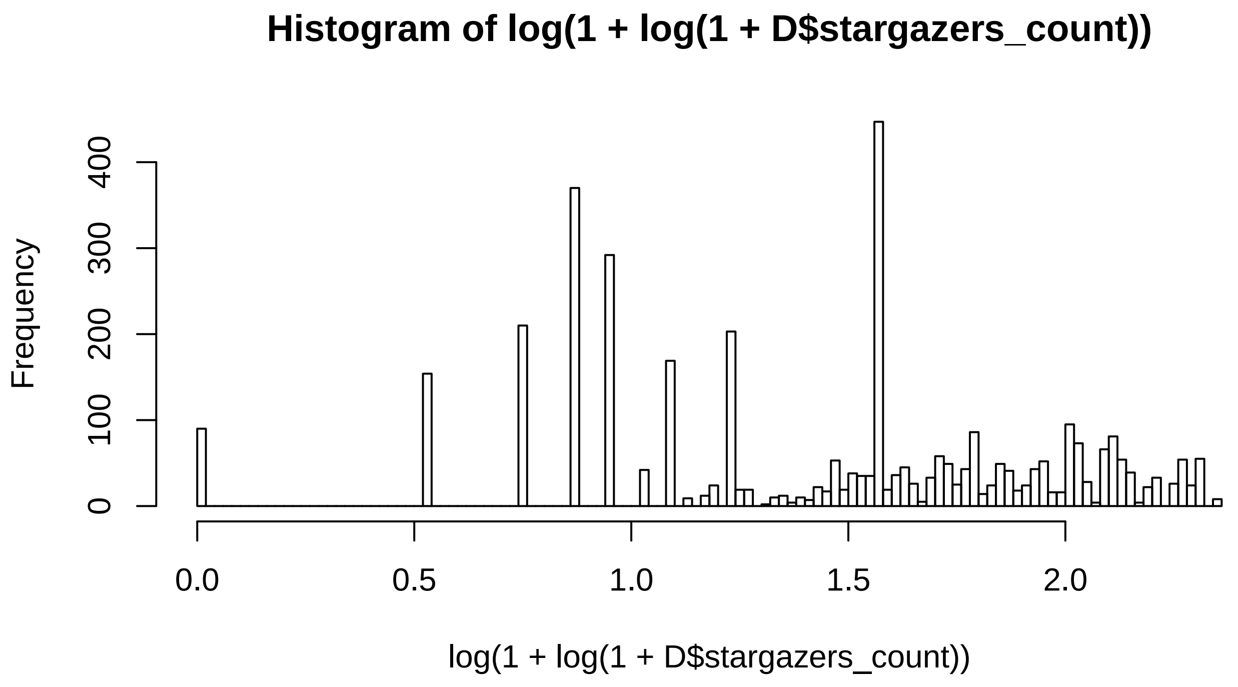
```
hist(log(1+D$stargazers_count), breaks=100)
```



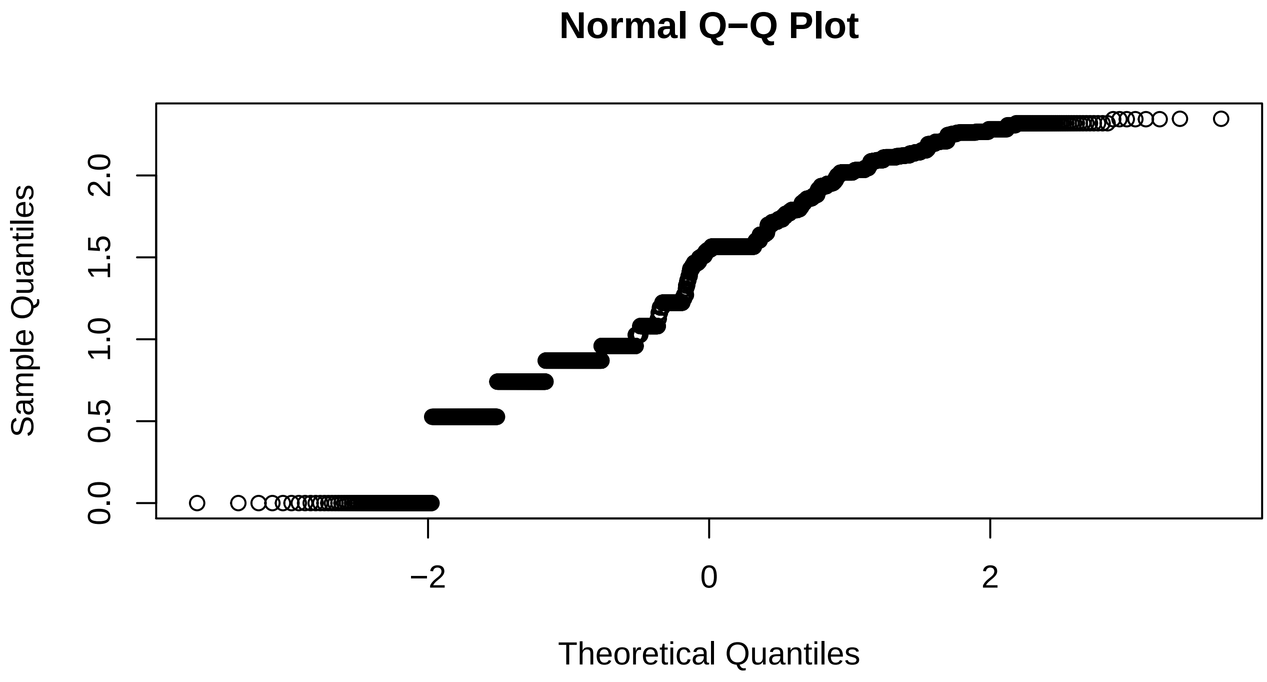
```
qqnorm(log(1+D$stargazers_count))
```



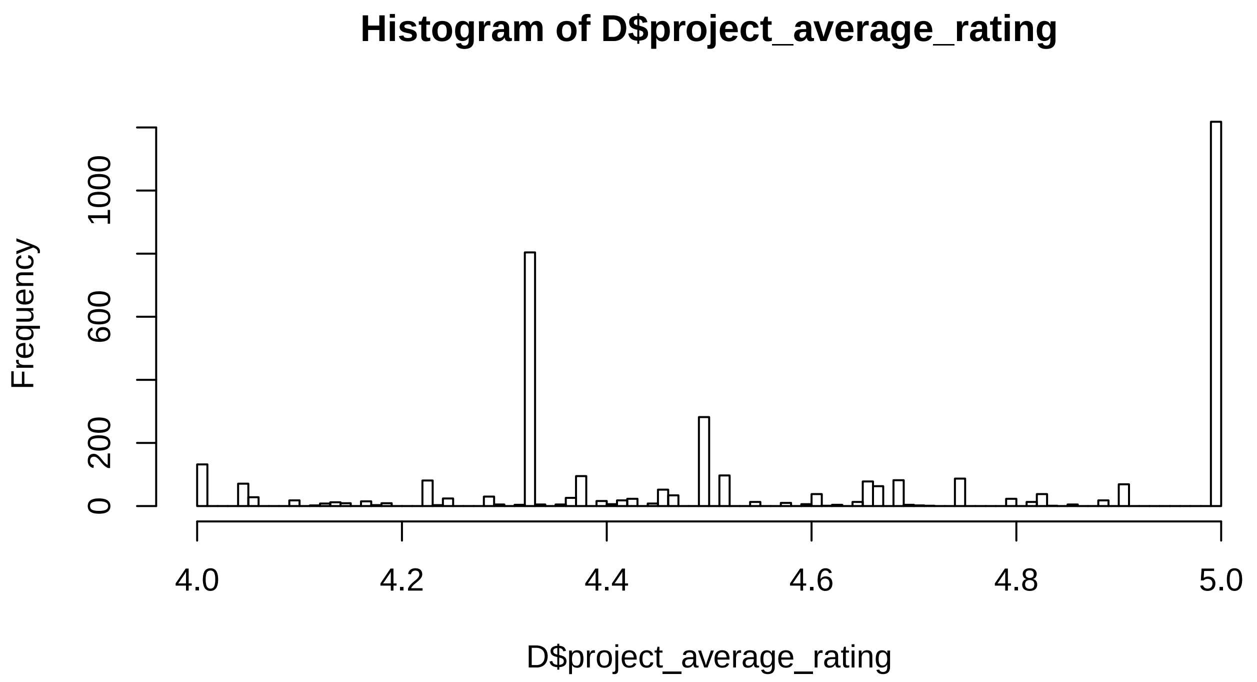
```
hist(log(1+log(1+D$stargazers_count)), breaks=100)
```



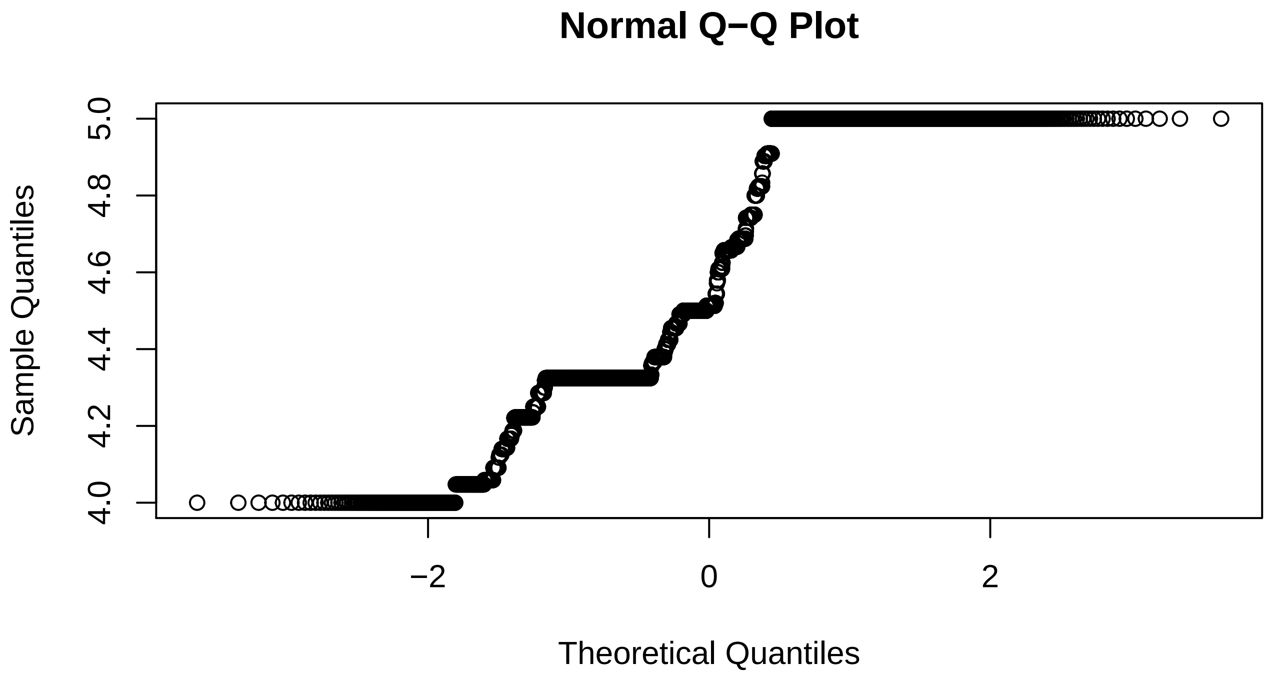
```
qqnorm(log(1+log(1+D$stargazers_count)))
```



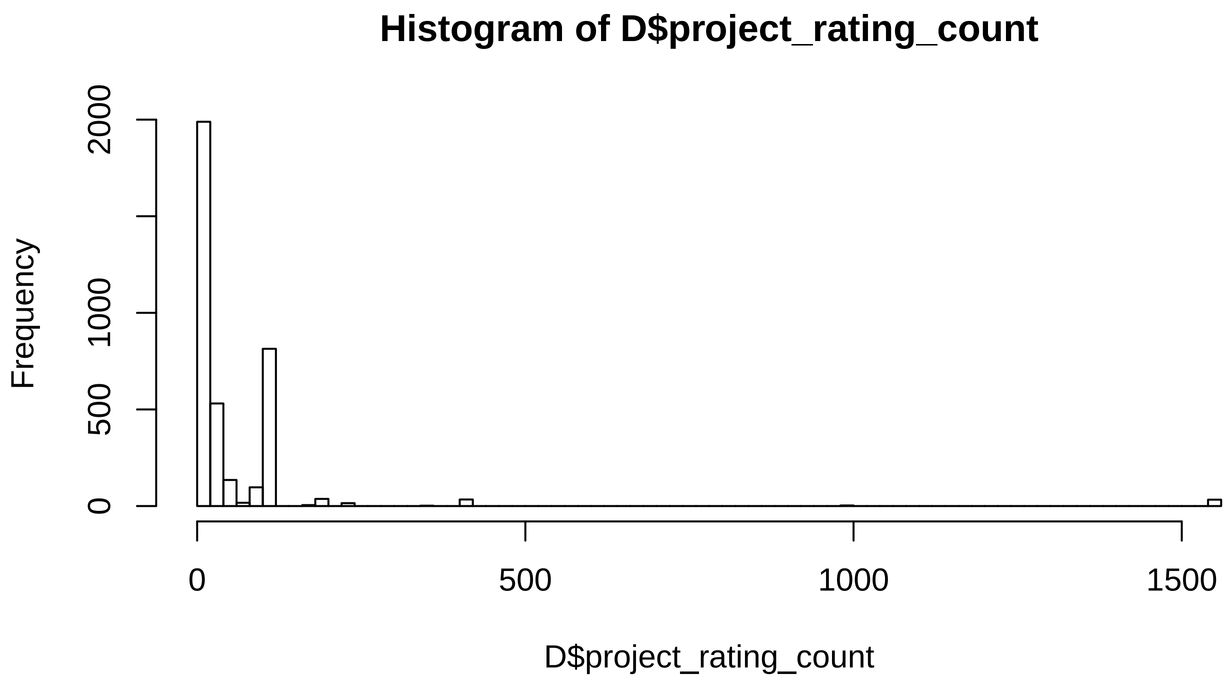
```
summary(D$stargazers_count)

##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##      0.0      4.0     39.0    552.8    189.0  12610.0

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



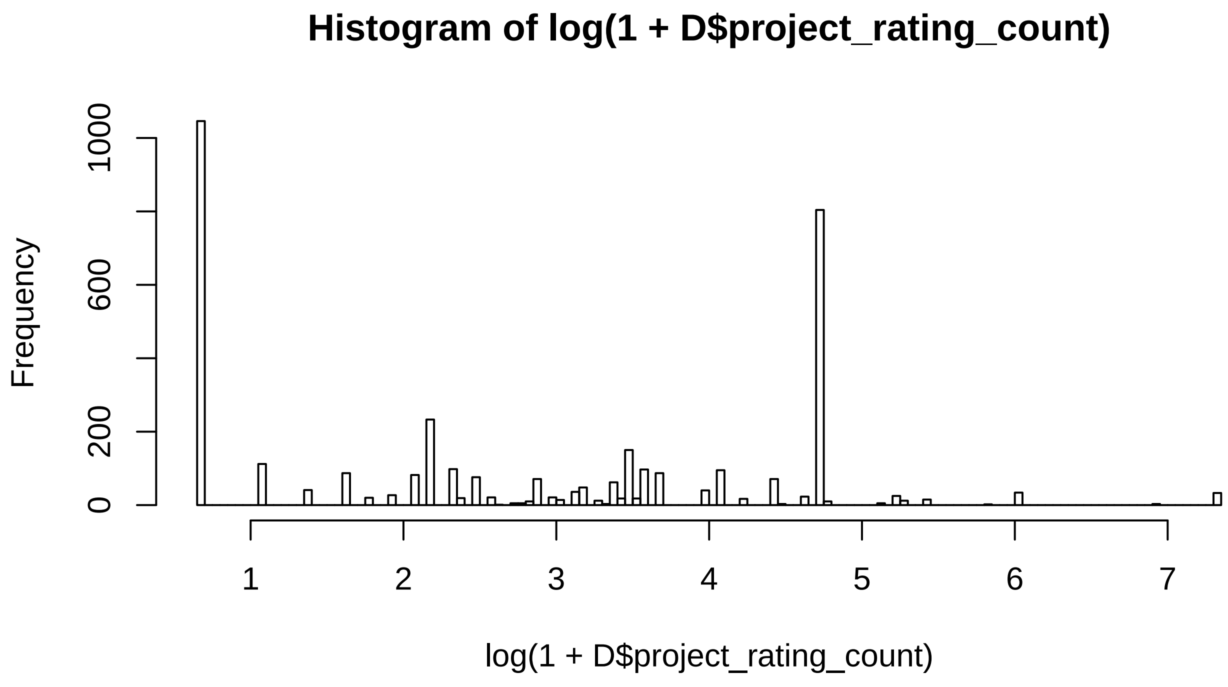
```
qqnorm(D$project_average_rating)
```



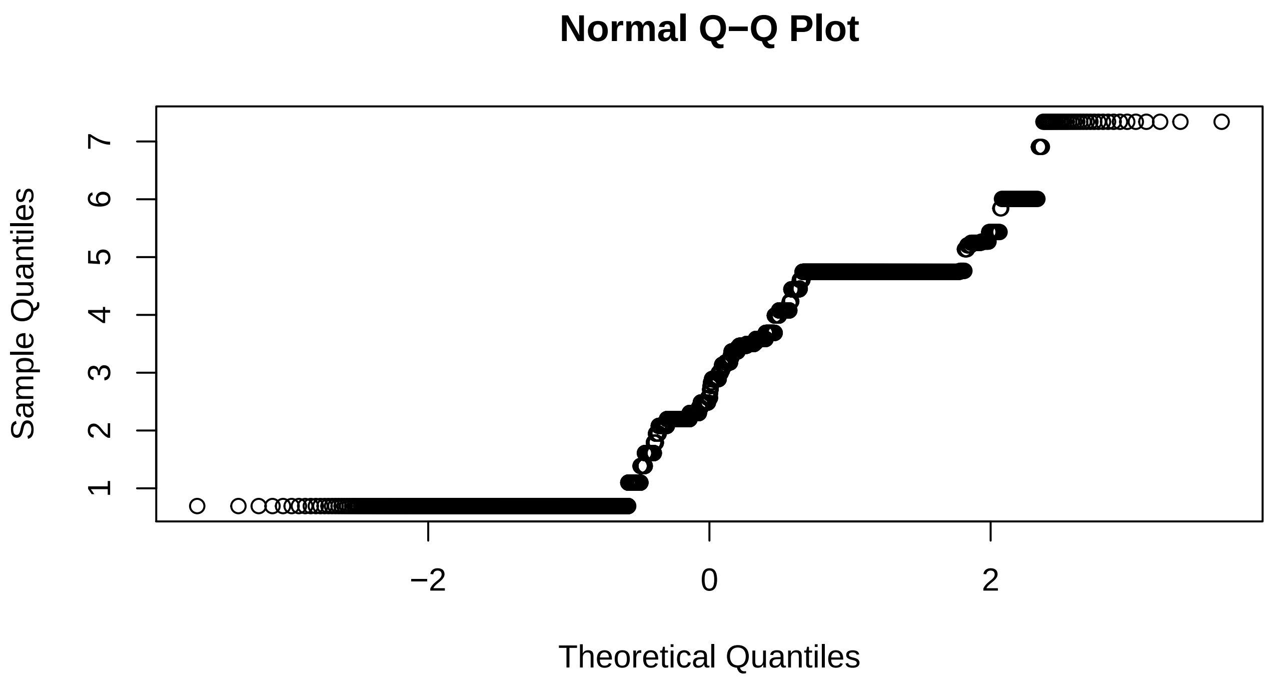
```
summary(D$project_average_rating)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   4.000   4.325   4.513   4.607   5.000   5.000

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



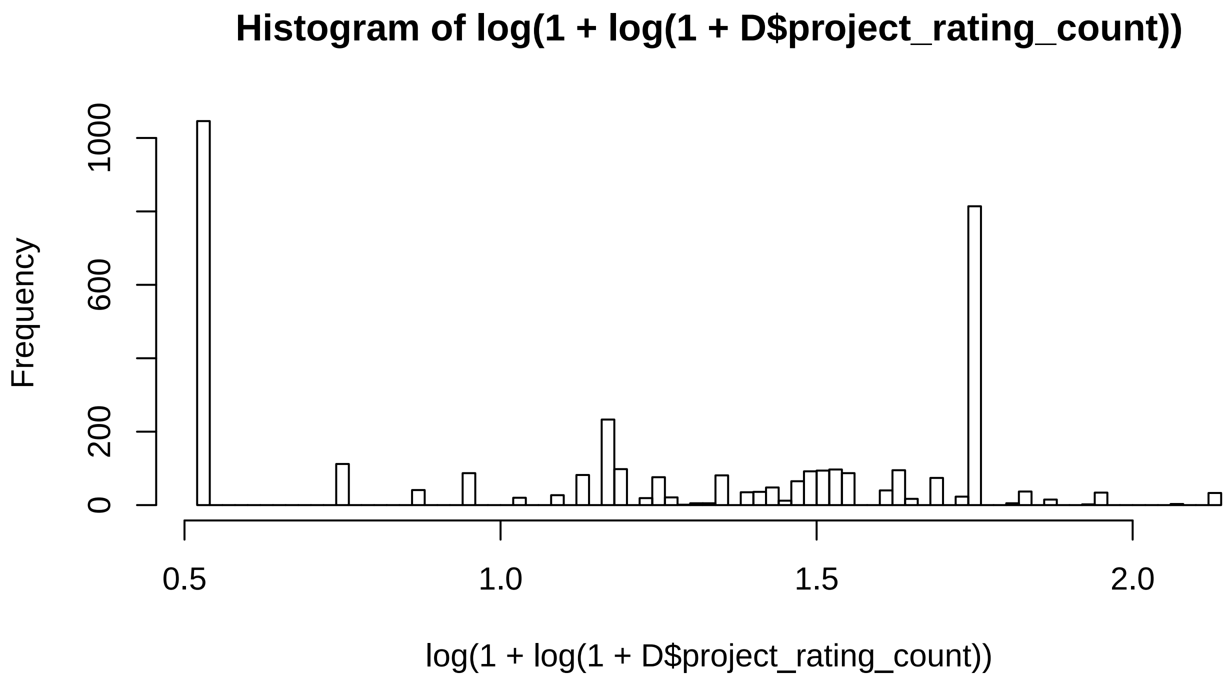
```
hist(log(1+D$project_rating_count), breaks=100)
```



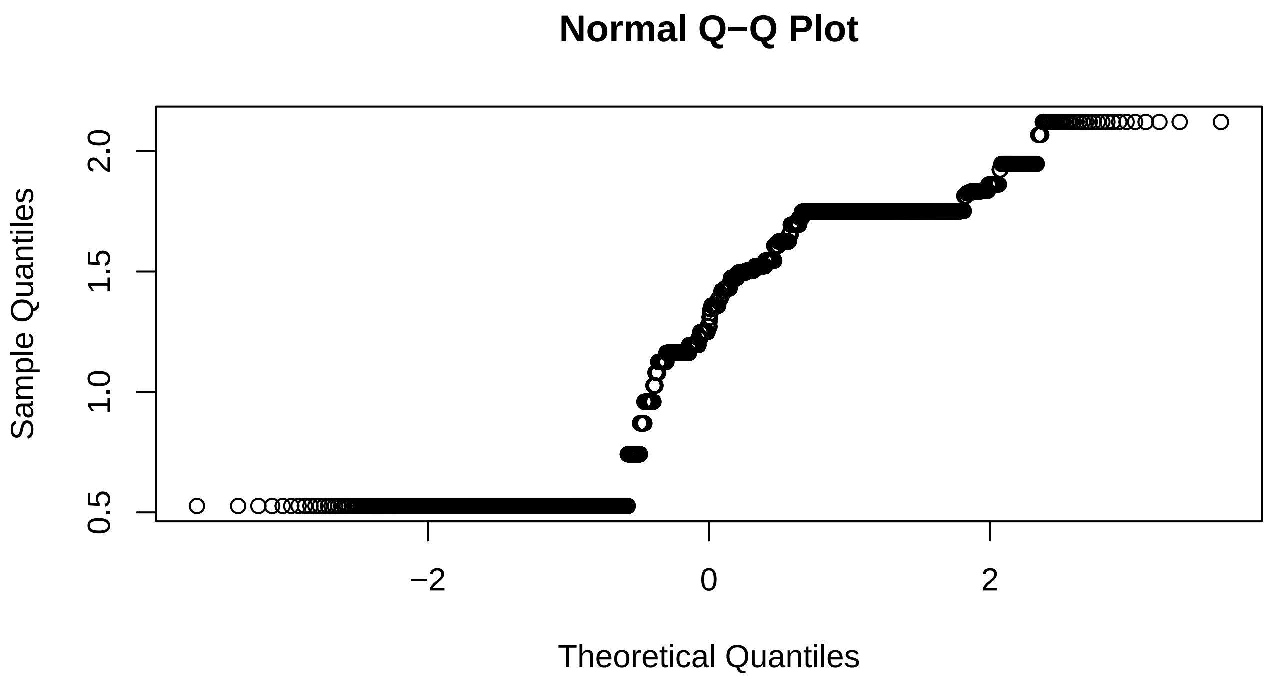
```
qqnorm(log(1+D$project_rating_count))
```



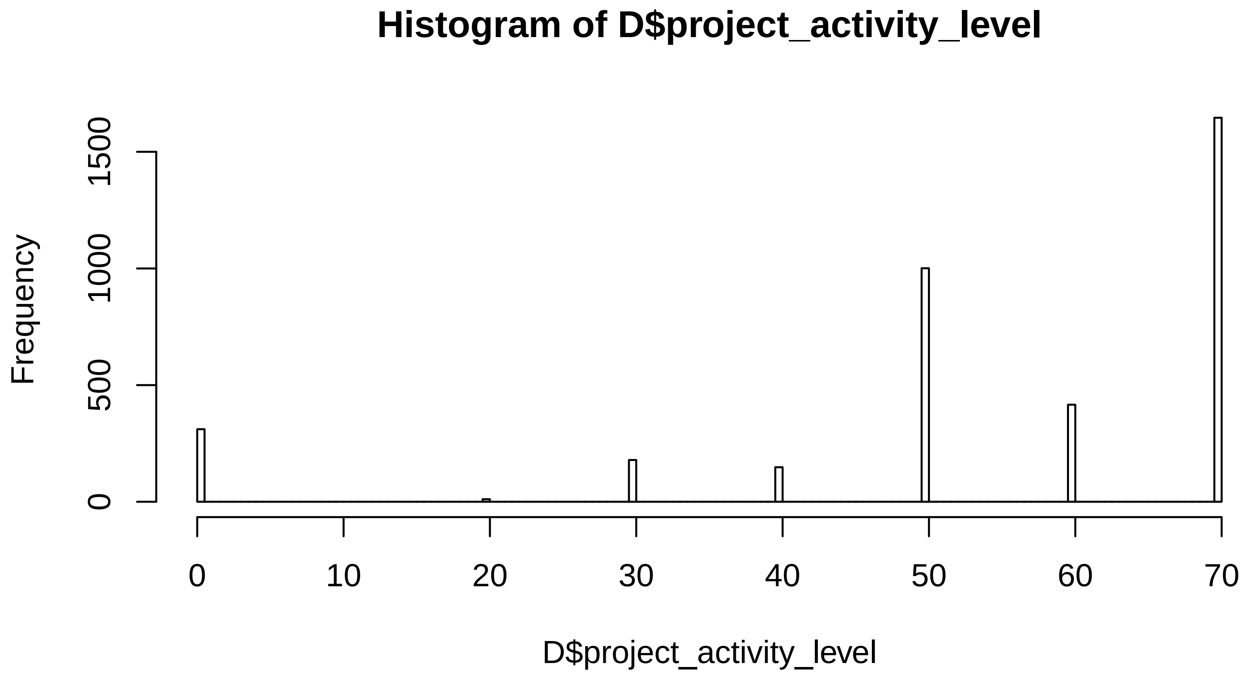
```
hist(log(1+log(1+D$project_rating_count)), breaks=100)
```



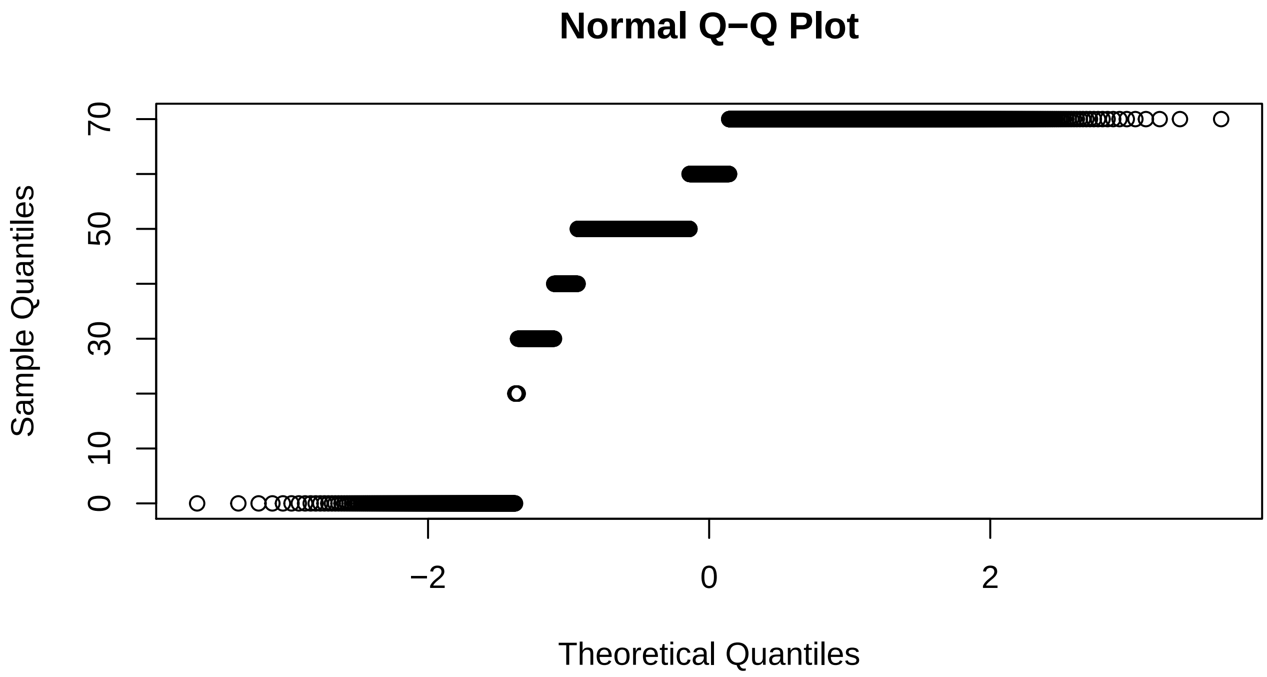
```
qqnorm(log(1+log(1+D$project_rating_count)))
```



```
summary(D$project_rating_count)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    1.00   12.00   57.95  114.00 1541.00

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



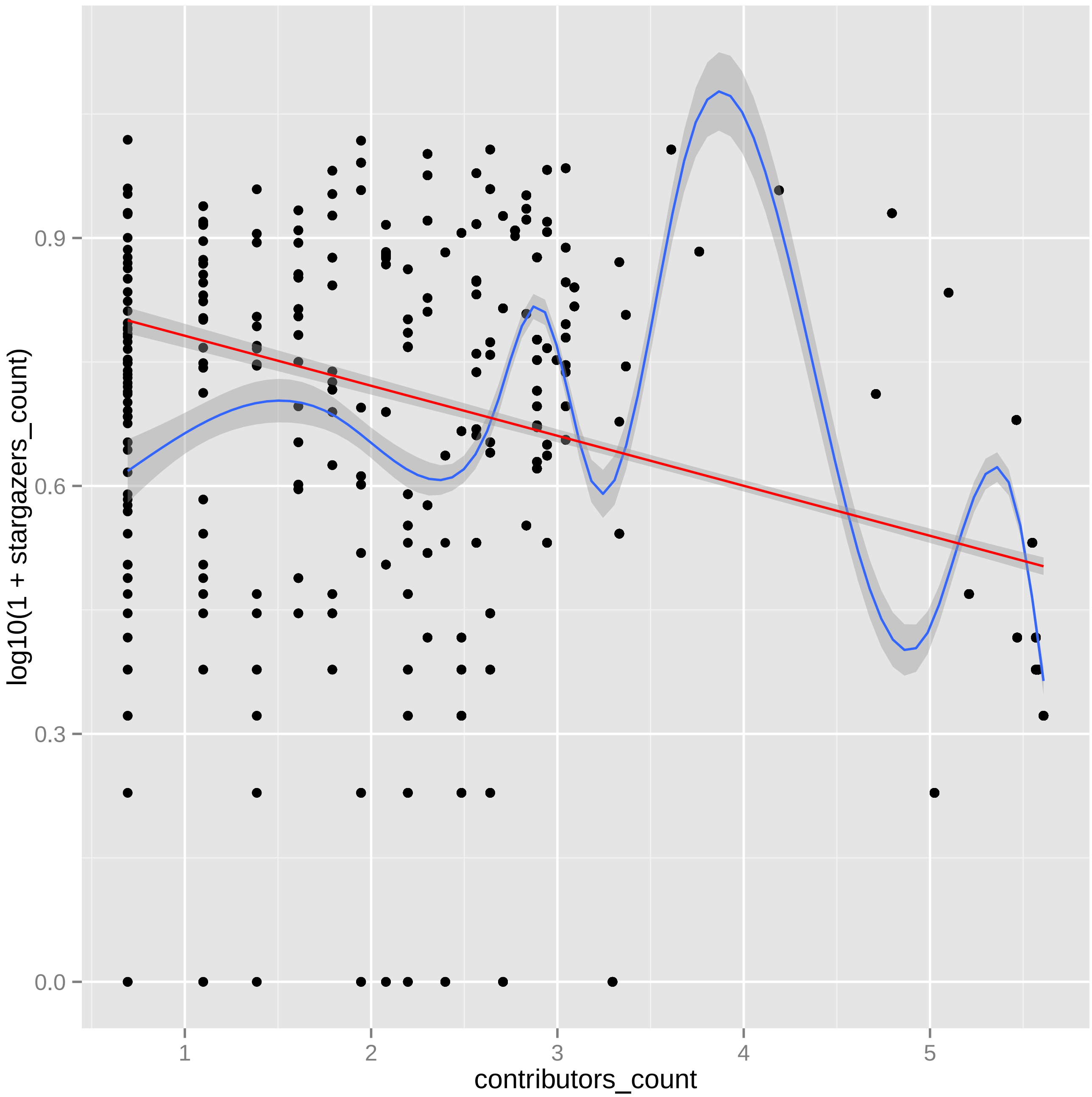
```
qqnorm(D$project_activity_level)
```



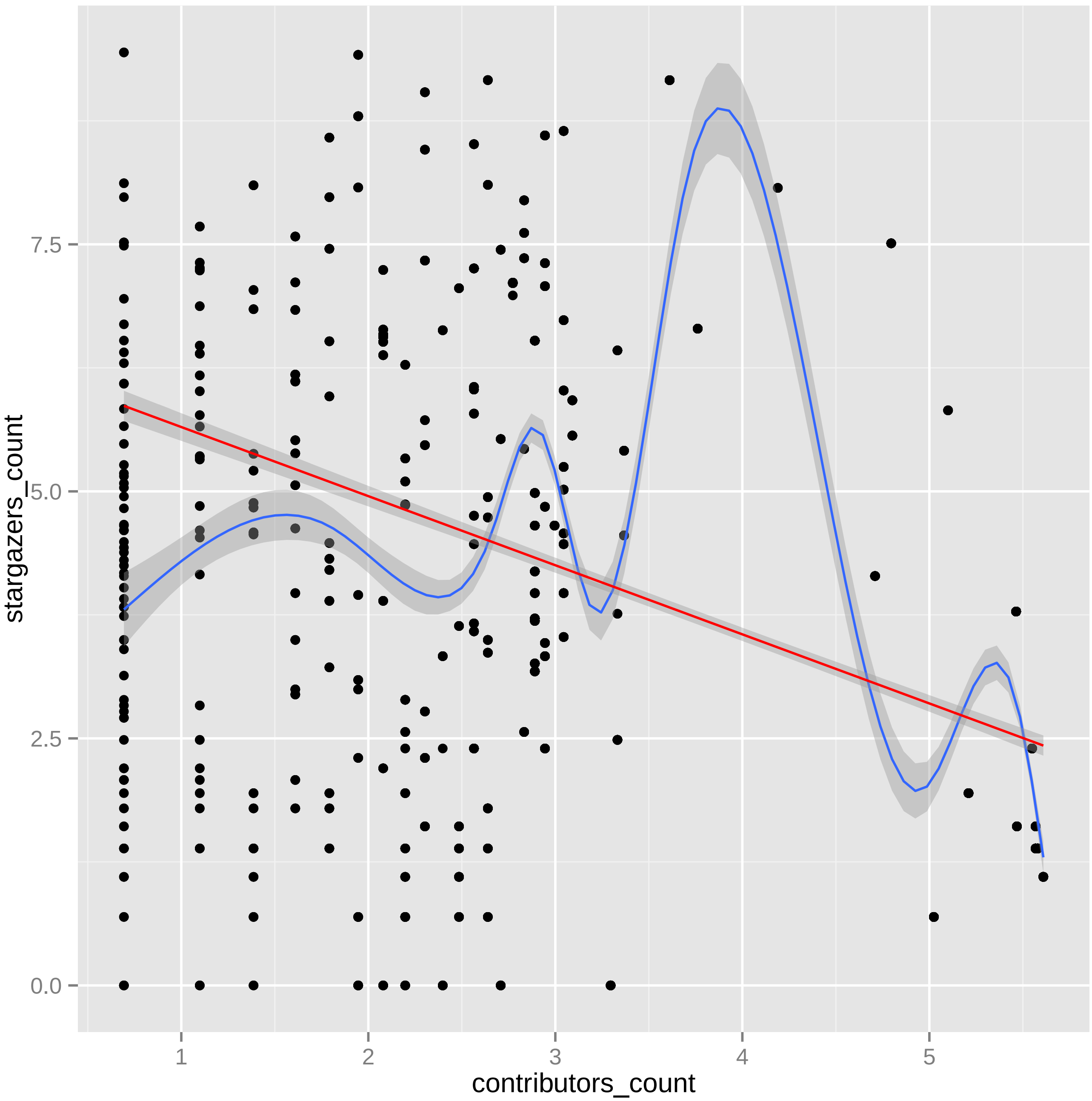
```
summary(D$project_activity_level)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    0.00   50.00   60.00   54.35   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)
```

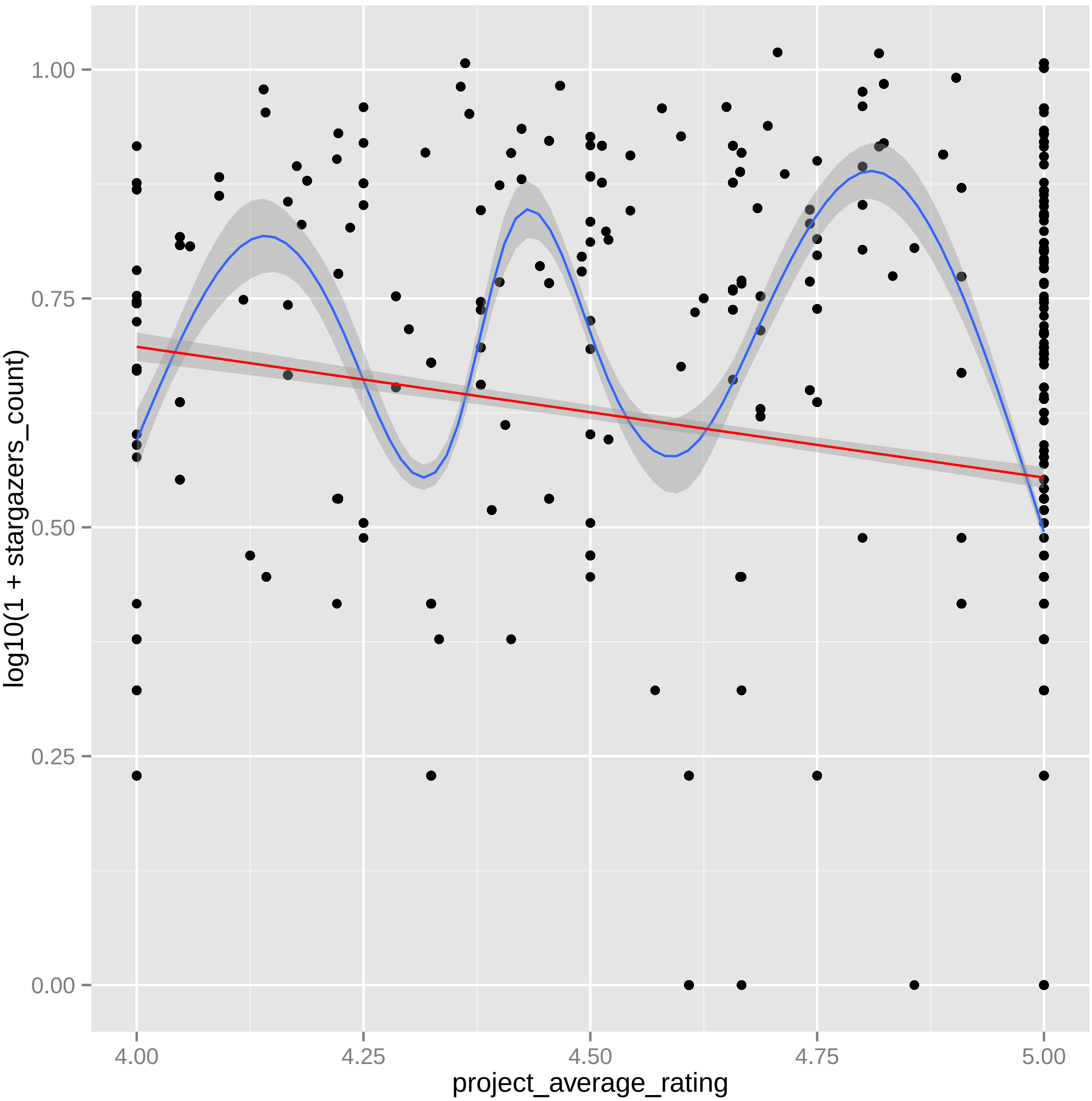
```
ggplot(D, aes(x=contributors_count, y=log10(1+stargazers_count))) + geom_point() + geom_smooth() + geom_smoot

## 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.
```



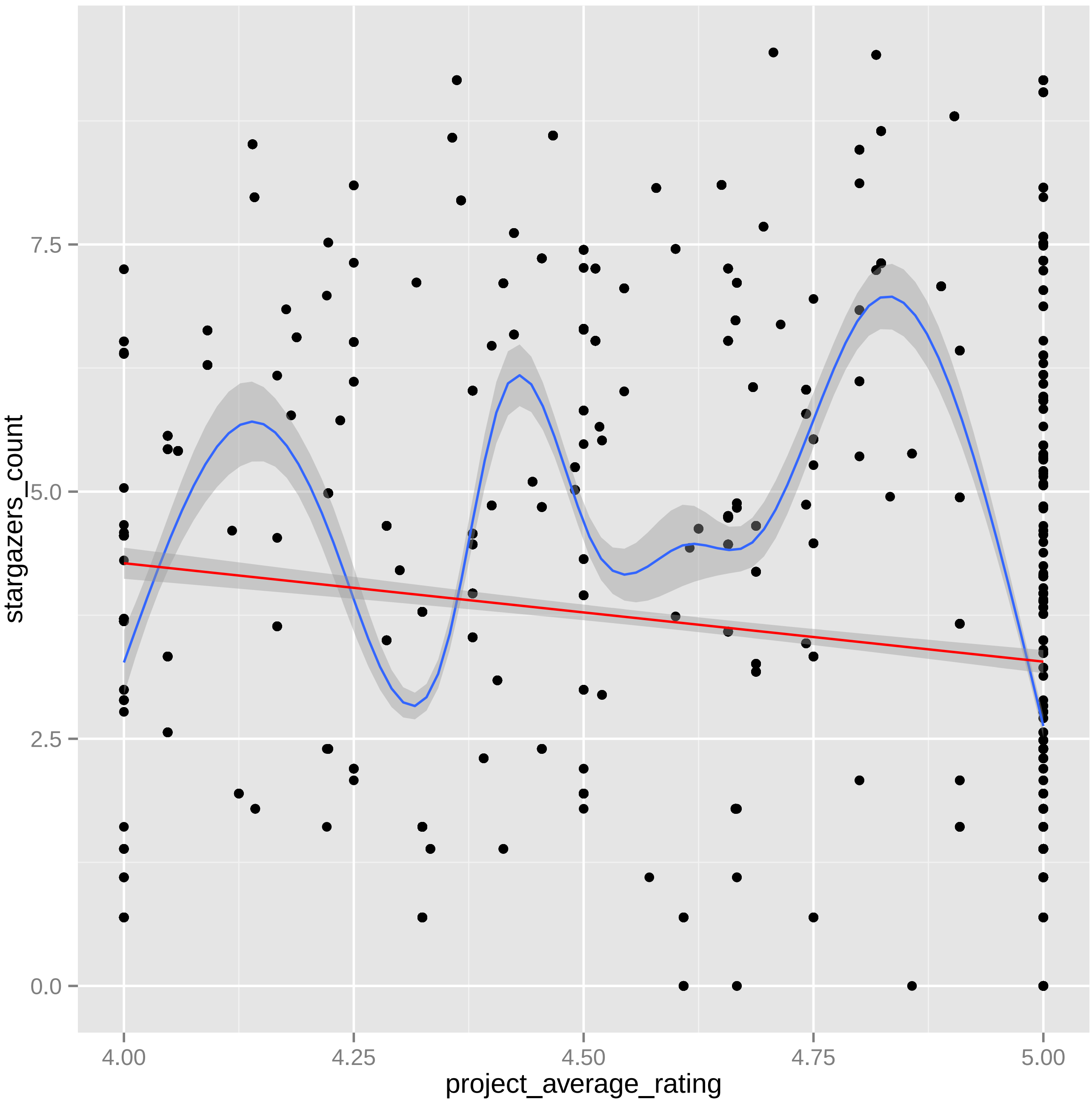
```
ggplot(D, aes(x=contributors_count, y=stargazers_count)) + geom_point() + geom_smooth() + geom_smooth(method=

## 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.
```



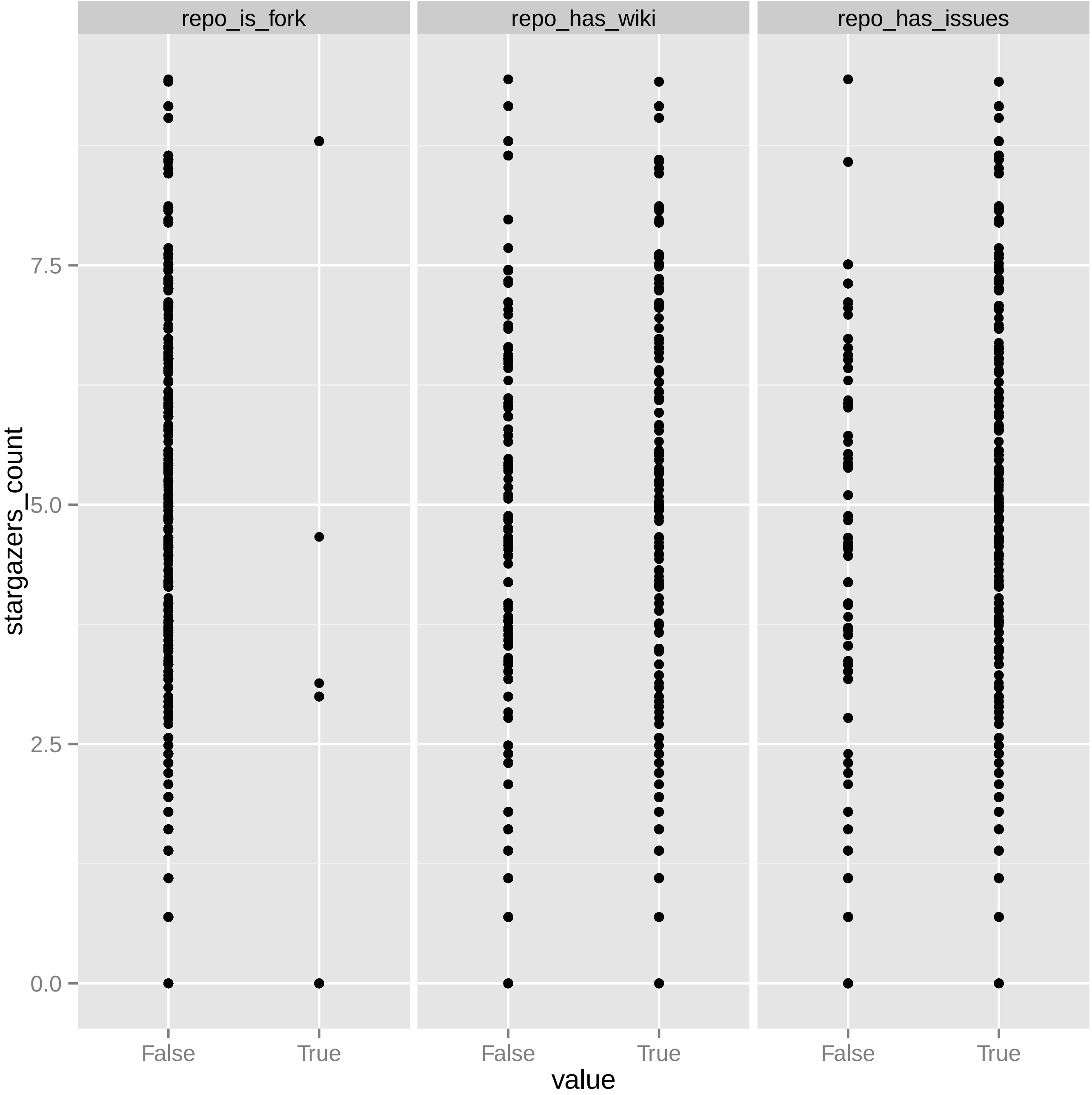
```
ggplot(D, aes(x=project_average_rating, y=log10(1+stargazers_count))) + geom_point() + geom_smooth() + geom_s

## 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.
```



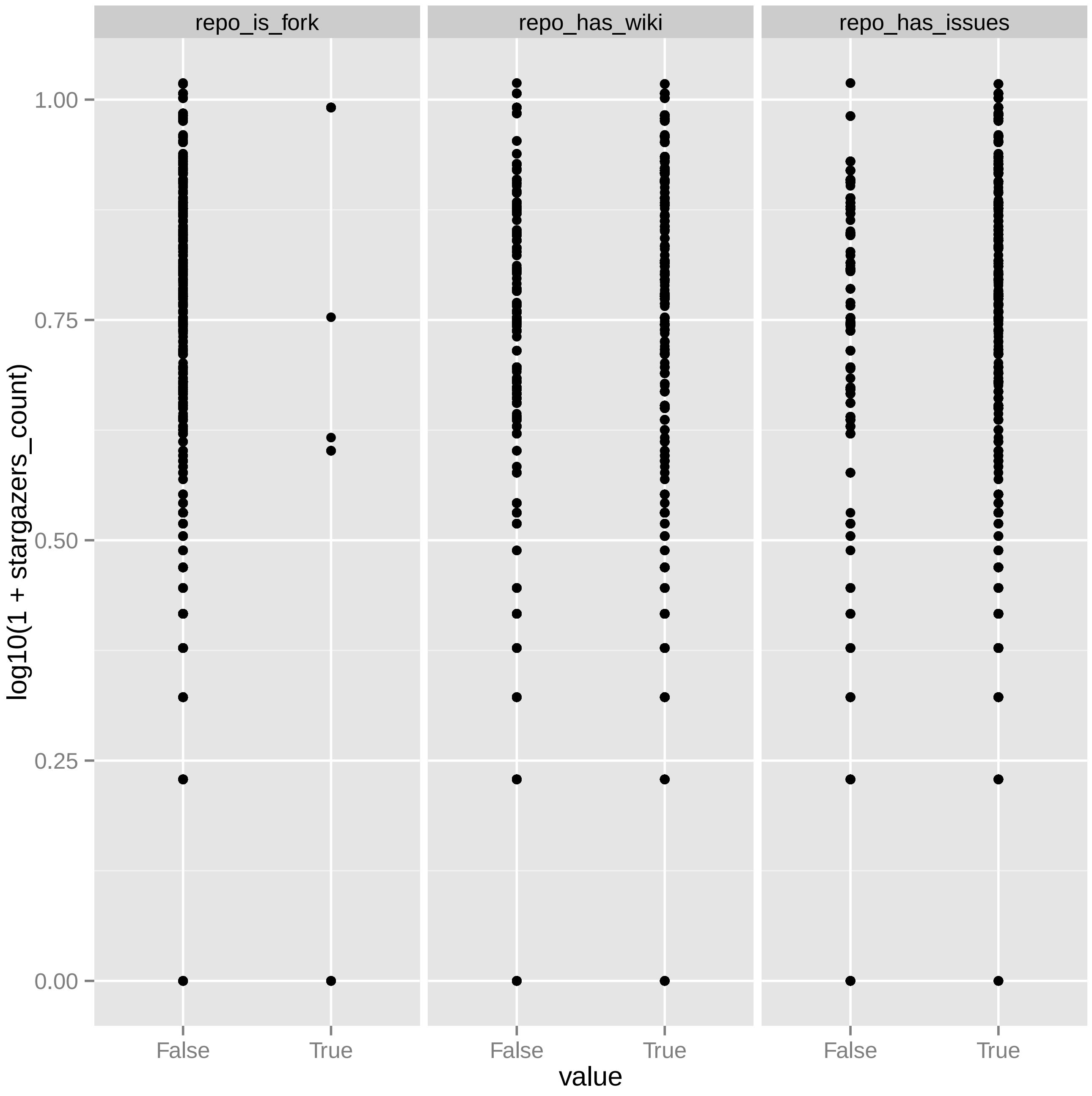
```
ggplot(D, aes(x=project_average_rating, y=stargazers_count)) + geom_point() + geom_smooth() + geom_smooth(met

## 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.
```



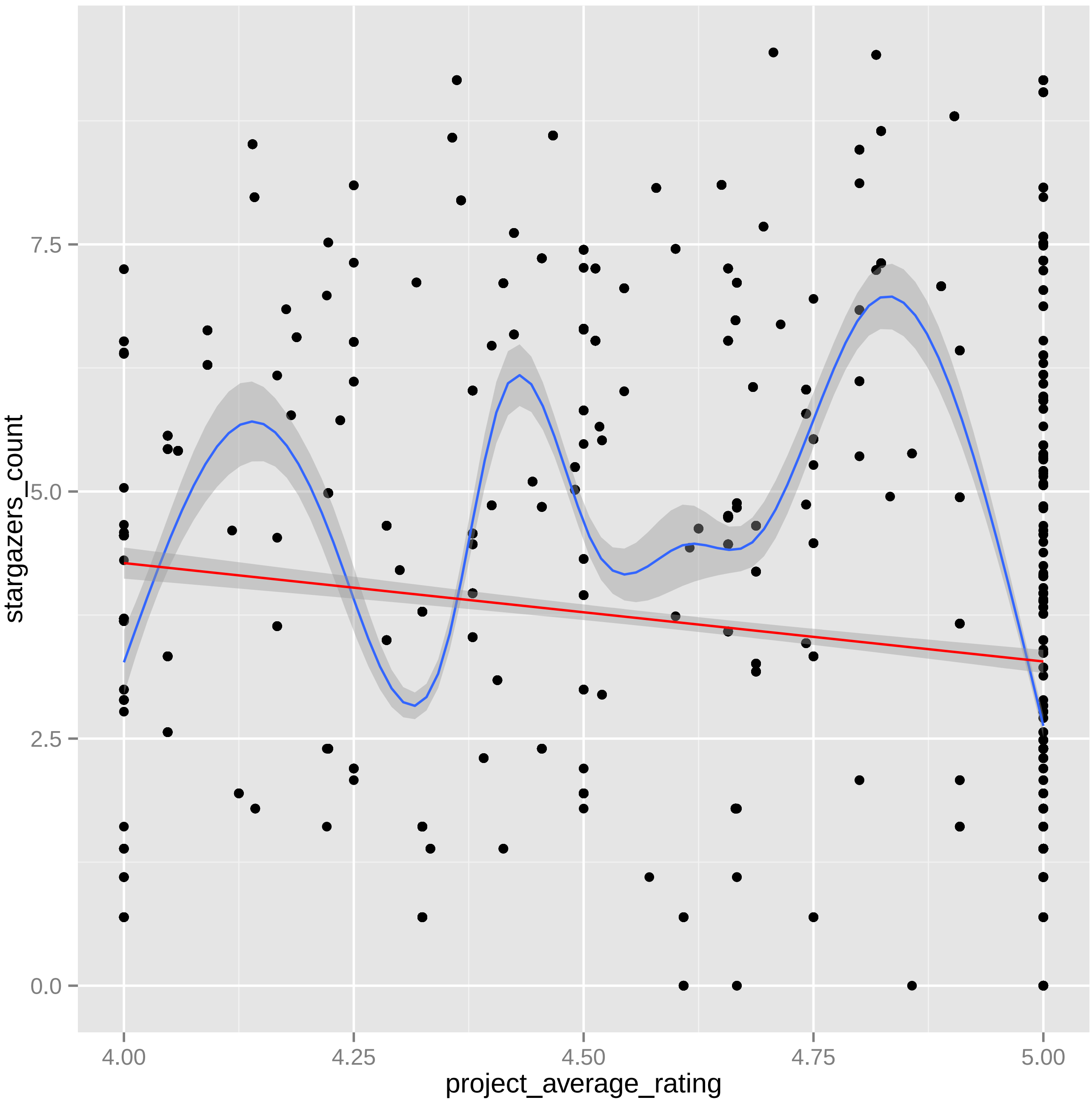
```
attrs <- c("repo_is_fork",
                  "repo_has_wiki", "repo_has_issues")
d <- cbind(melt(D[,attrs], id.vars=c()), stargazers_count=D$stargazers_count)
ggplot(d,aes(x = value, y=stargazers_count)) +
       facet_wrap(~variable, scales = "free_x") +
       geom_point() + geom_smooth() + geom_smooth(method=lm,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



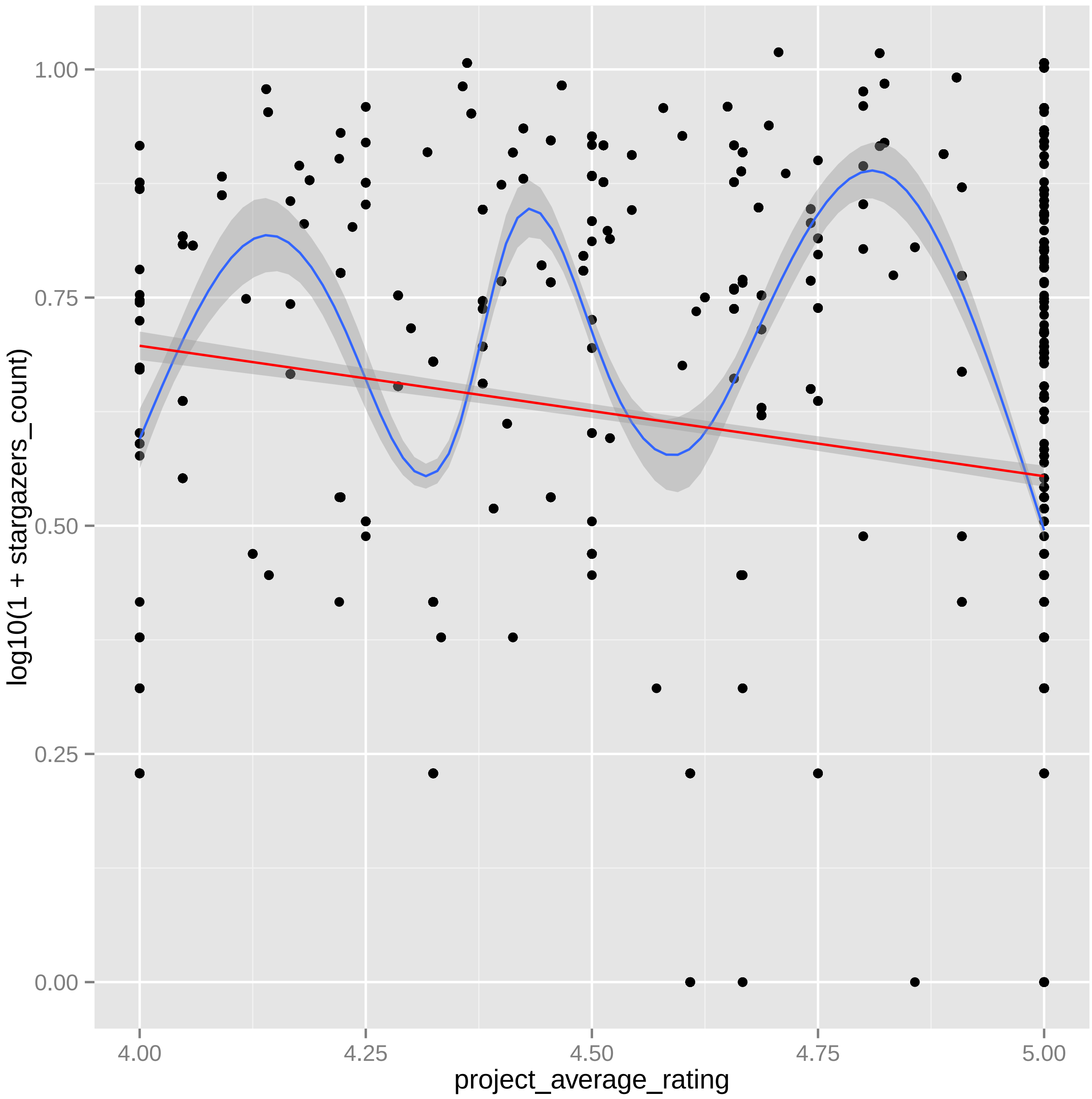
```
ggplot(d,aes(x = value, y=log10(1+stargazers_count))) +
       facet_wrap(~variable, scales = "free_x") +
       geom_point() + geom_smooth() + geom_smooth(method=lm, color="red")

## 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)?
```



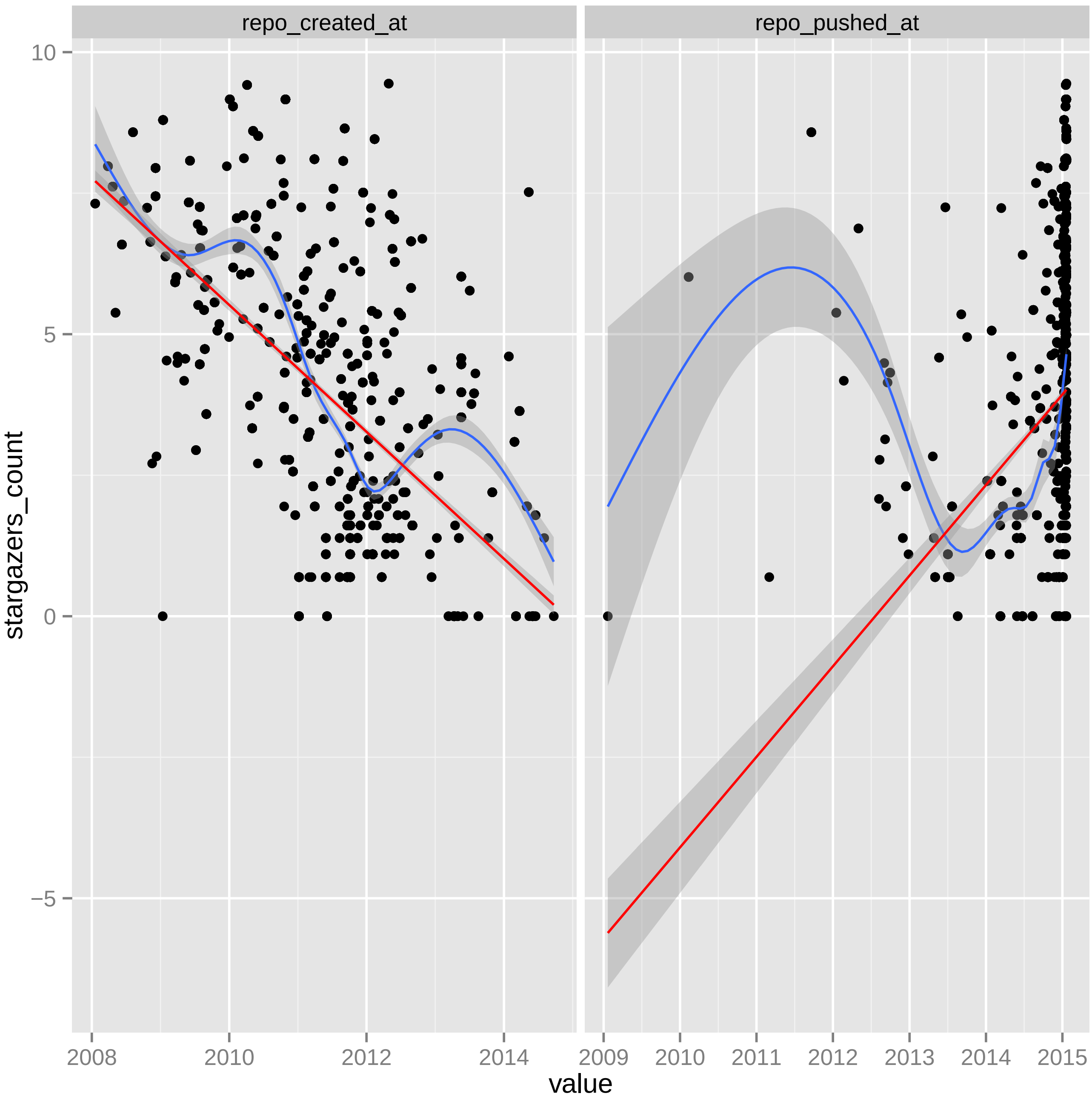
```
ggplot(D,aes(x = project_average_rating, y=stargazers_count)) +
       geom_point() + geom_smooth() + geom_smooth(method=lm, color="red")

## 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.
```



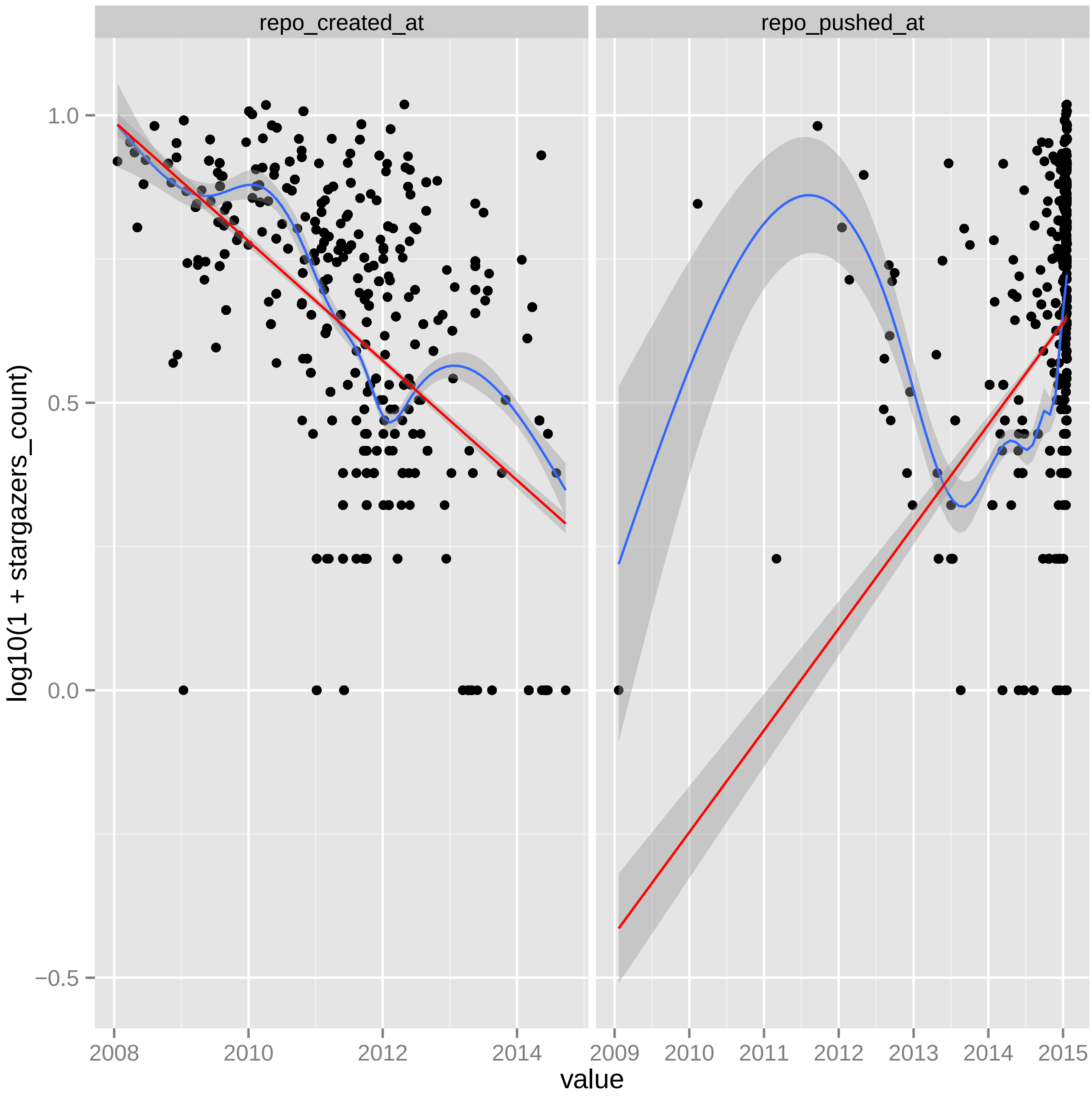
```
ggplot(D,aes(x = project_average_rating, y=log10(1+stargazers_count))) +
       geom_point() + geom_smooth() + geom_smooth(method=lm, color="red")

## 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.
```



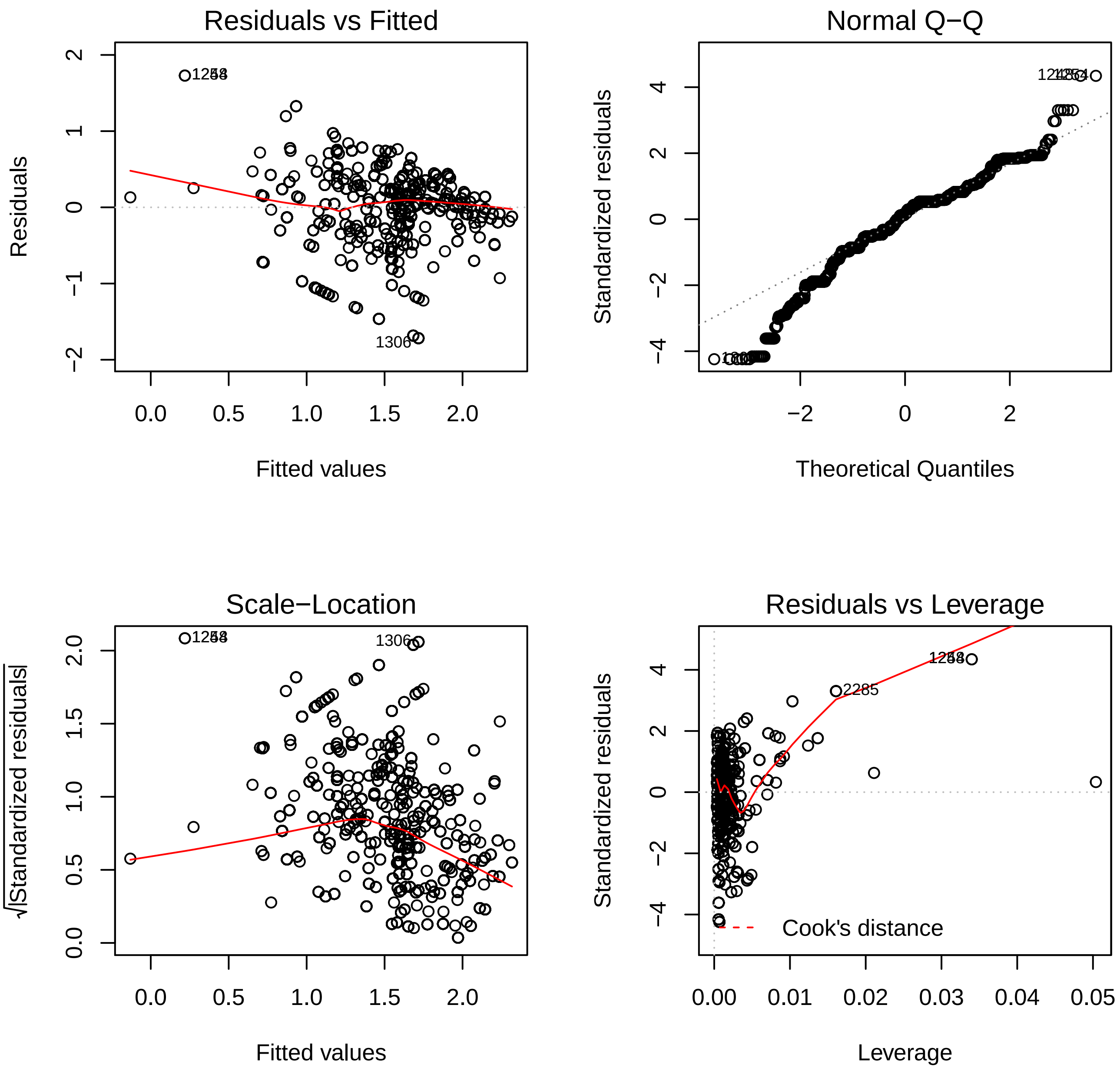
```
ggplot(d,aes(x = value, y=log10(1+stargazers_count))) +
       facet_wrap(~variable, scales = "free_x") +
       geom_point() + geom_smooth() + geom_smooth(method=lm, color="red")

## 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.
```



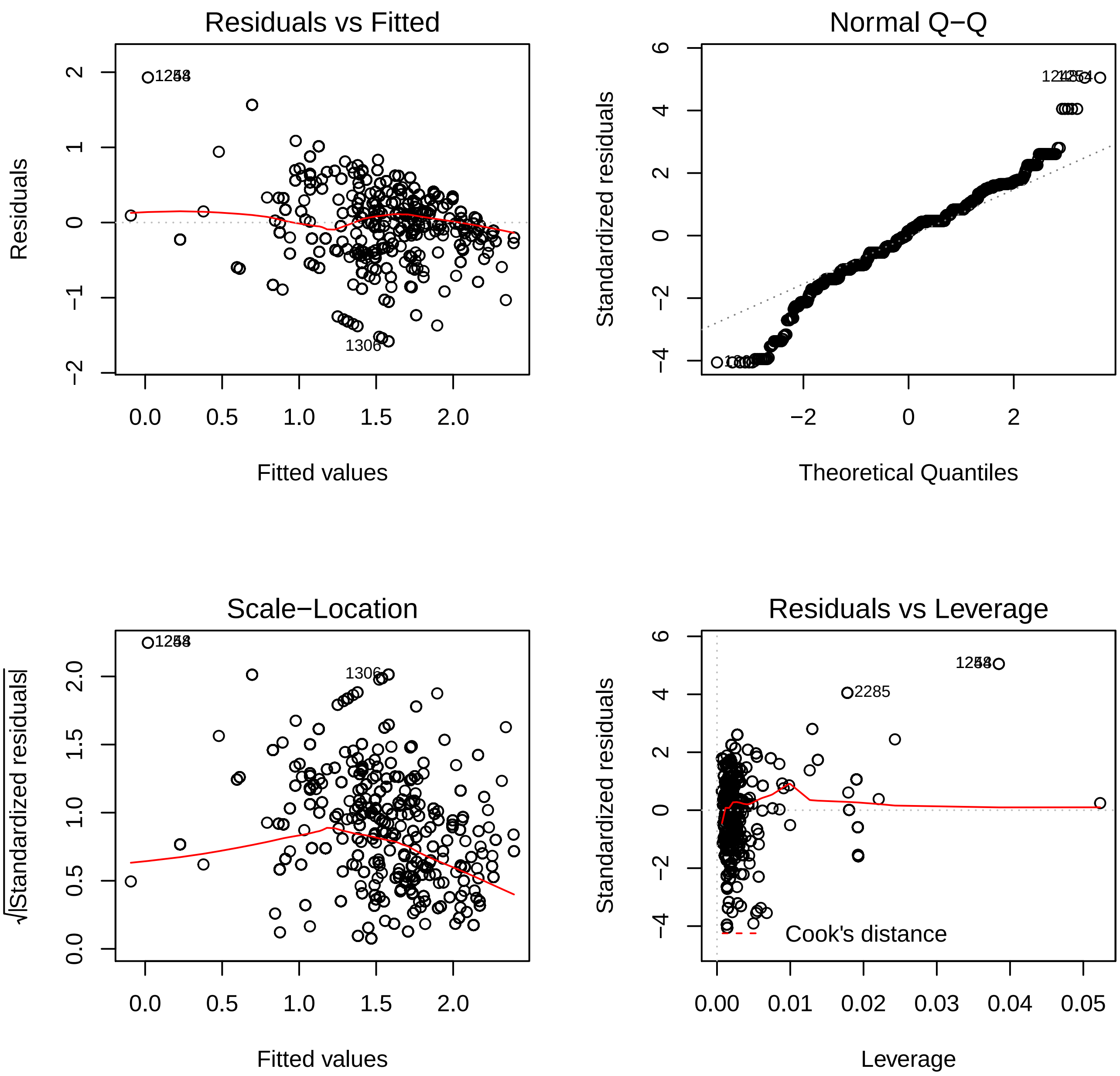
```
D$stargazers_count <- log(1+D$stargazers_count)
```

```
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
## -1.71749 -0.21145  0.06645  0.23770  1.72907
##
```

```
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -7.682e+00  7.280e-01  -10.55   <2e-16 ***
## contributors_count -7.343e-02  4.684e-03  -15.68   <2e-16 ***
## repo_pushed_at      1.087e-03  4.204e-05   25.86   <2e-16 ***
## repo_created_at    -5.545e-04  1.614e-05  -34.37   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.405 on 3708 degrees of freedom
## Multiple R-squared:  0.4375,Adjusted R-squared:  0.437
## F-statistic: 961.2 on 3 and 3708 DF,  p-value: < 2.2e-16

plot(m)
```

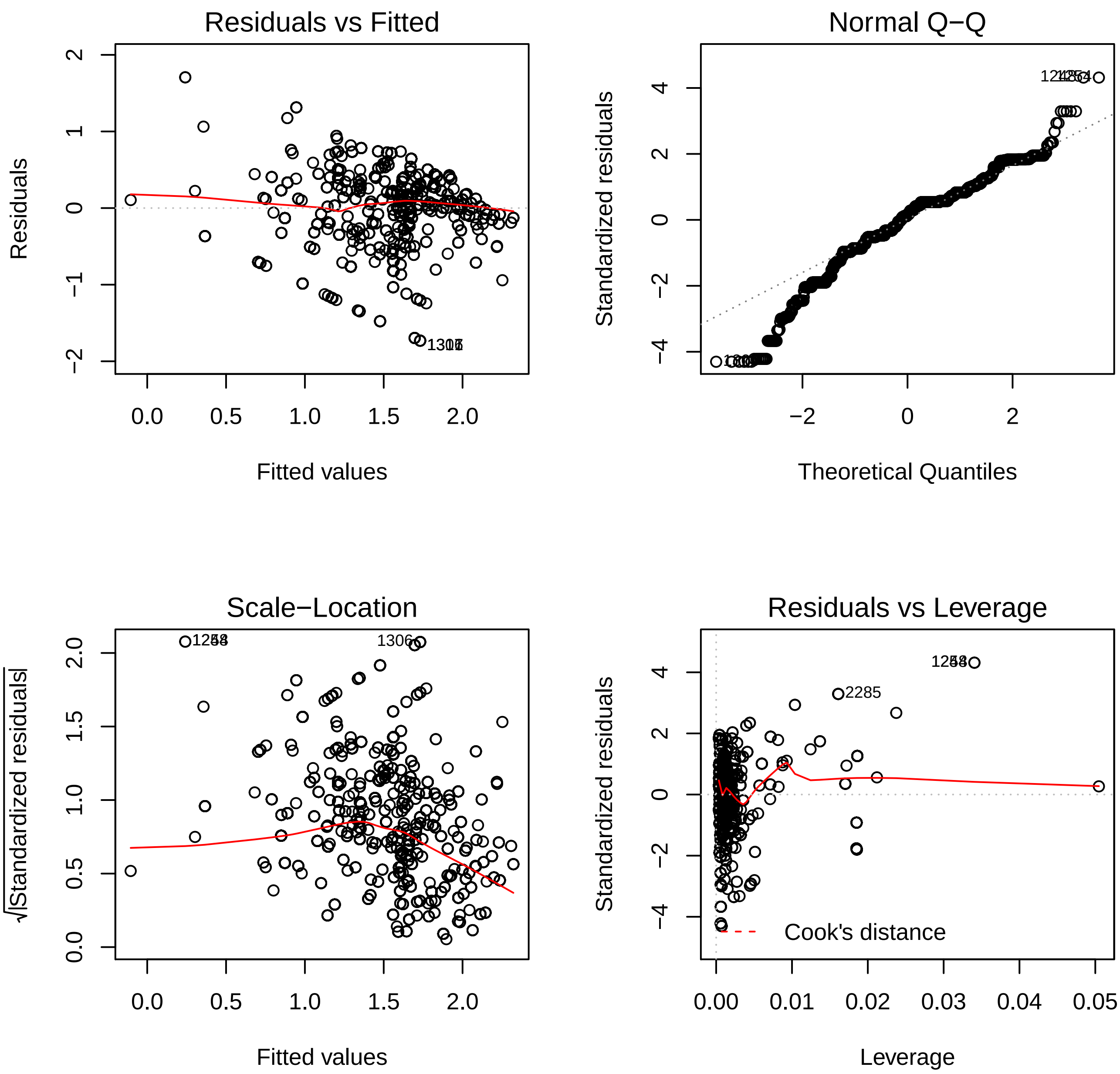


```
par(mfrow=c(2,2))
m2 <- lm(stargazers_count ~ contributors_count + repo_pushed_at + repo_created_at + repo_is_fork + repo_has_w
summary(m2)

##
## 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
## -1.58039 -0.21326  0.05862  0.18339  1.92994
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -9.075e+00  7.202e-01 -12.600  < 2e-16 ***
## contributors_count  -1.050e-01  4.883e-03 -21.495  < 2e-16 ***
## repo_pushed_at       1.118e-03  4.210e-05  26.566  < 2e-16 ***
## repo_created_at     -5.010e-04  1.617e-05 -30.978  < 2e-16 ***
## repo_is_forkTrue    -3.607e-01  5.187e-02  -6.955 4.16e-12 ***
## repo_has_wikiTrue   -2.132e-02  1.469e-02  -1.451    0.147
## repo_has_issuesTrue  2.635e-01  1.763e-02  14.944  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3898 on 3705 degrees of freedom
## Multiple R-squared:  0.4793,Adjusted R-squared:  0.4784
## F-statistic: 568.3 on 6 and 3705 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   3708 608.17
## 2   3705 562.97  3    45.195 99.145 < 2.2e-16 ***
## ---
## 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
## -1.73119 -0.20961  0.05708  0.23077  1.70676
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -7.700e+00  7.234e-01 -10.643  < 2e-16 ***
## contributors_count -7.877e-02  4.718e-03 -16.697  < 2e-16 ***
## repo_pushed_at      1.081e-03  4.178e-05  25.882  < 2e-16 ***
## repo_created_at    -5.456e-04  1.608e-05 -33.919  < 2e-16 ***
## repo_is_forkTrue   -3.725e-01  5.354e-02  -6.958 4.08e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4024 on 3707 degrees of freedom
## Multiple R-squared:  0.4447,Adjusted R-squared:  0.4441
## F-statistic: 742.2 on 4 and 3707 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   3708 608.17
## 2   3707 600.33  1    7.8393 48.407 4.076e-12 ***
## ---
## 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")
```