Correlation PDF

Vinod Kumar Ahuja

November 21, 2017

Correlation between Closed issues and number of commits

```
# Loading Libraries
library(RODBC) # for database connection
library(sqldf) # for sql query
```

```
## Loading required package: gsubfn
```

```
## Loading required package: proto
```

```
## Loading required package: RSQLite
```

```
library(tidyverse) # for joins
```

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr
```

```
## Conflicts with tidy packages ----------------------------------------------
```

```
## filter(): dplyr, stats
## lag():    dplyr, stats
```

```
library(ggpubr) # for correlation plotting
```

```
## Loading required package: magrittr
```

```
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:purrr':
##
##     set_names
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(PerformanceAnalytics) # for correlation plotting
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
##
## Attaching package: 'xts'
```

```
## The following objects are masked from 'package:dplyr':
##
##     first, last
```

```
##
## Attaching package: 'PerformanceAnalytics'
```

```
## The following object is masked from 'package:graphics':
##
##     legend
```

```
# Connecting to GHData MySQL database
connect <- odbcConnect("ghtorrent")

#Select Project
Project_Name <- "rust"
Project_Owner <- "rust-lang"

#Finding Project ID from GHTorrent data
project_id <- sqlQuery(connect, paste("SELECT projects.id FROM projects INNER JOIN users ON pr
ojects.owner_id = users.id WHERE projects.name ='",Project_Name,"'AND users.login = '",Project
_Owner,"';",sep = ""))

# Selecting number of commits
query <- sqlQuery(connect, paste("select id as ID, created_at as 'Date' from commits where pro
ject_id =",project_id,";"),as.is=T)
```

```
# Sorting by Date
number_of_commits <- query[order(query$Date),]

# Converting date format to character
number_of_commits$Date <- as.character(number_of_commits$Date)

# Counting number of commits
number_of_commits$Commit_Count <- seq.int(nrow(number_of_commits))

# selecting maximum values and one date from the repeated dates
selected_commits <- sqldf(' select Date, Commit_Count from number_of_commits Group by Date')

# Selecting number of closed issues
query <- sqlQuery(connect, paste("SELECT issue_events.event_id as 'ID', issue_events.action as
 'Issue_status', issue_events.created_at as 'Date' FROM issue_events INNER JOIN issues ON issu
e_events.issue_id = issues.id WHERE issues.repo_id =", project_id, "AND issue_events.action =
'closed';"))

# Sorting by Date
Closed_Issues <- query[order(query$Date),]

# Converting date format to character
Closed_Issues$Date <- as.character(Closed_Issues$Date)

# Counting number of commits
Closed_Issues$Issues_Count <- seq.int(nrow(Closed_Issues))

# selecting maximum values and one date from the repeated dates
selected_closed_issues <- sqldf(' select Date, Issues_Count from Closed_Issues Group by Date')

# Merge closed issues count and number of commits count by dates into one data frame for corre
lation
join <- full_join(selected_closed_issues,selected_commits)
```

```
## Joining, by = "Date"
```

```
# Sort join by date
join <- join[order(join$Date),]

# Various correlation matrix

# Pearson
Cor_matrix_pearson <- cor.test(join$Issues_Count, join$Commit_Count,
                method = "pearson")
Cor_matrix_pearson
```

```
##
##  Pearson's product-moment correlation
##
## data:  join$Issues_Count and join$Commit_Count
## t = 72.62, df = 29, p-value < 2.2e-16
```

```
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9942648 0.9986936
## sample estimates:
##       cor
## 0.9972617
```

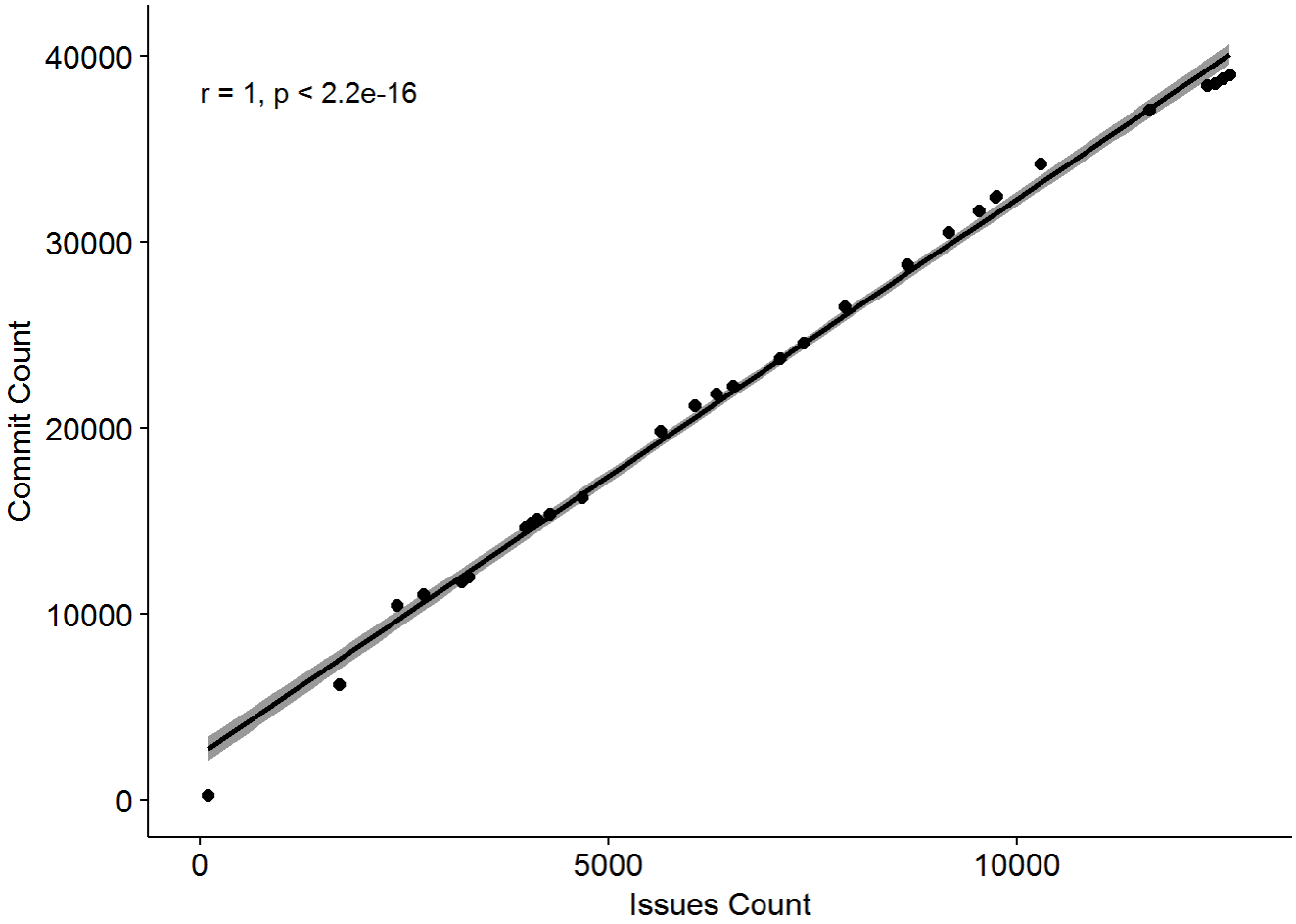
```
#Kendall
Cor_matrix_kendall <- cor.test(join$Issues_Count, join$Commit_Count,
                               method = "kendall")
Cor_matrix_kendall
```

```
##
##  Kendall's rank correlation tau
##
## data:  join$Issues_Count and join$Commit_Count
## T = 465, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
## tau
##   1
```

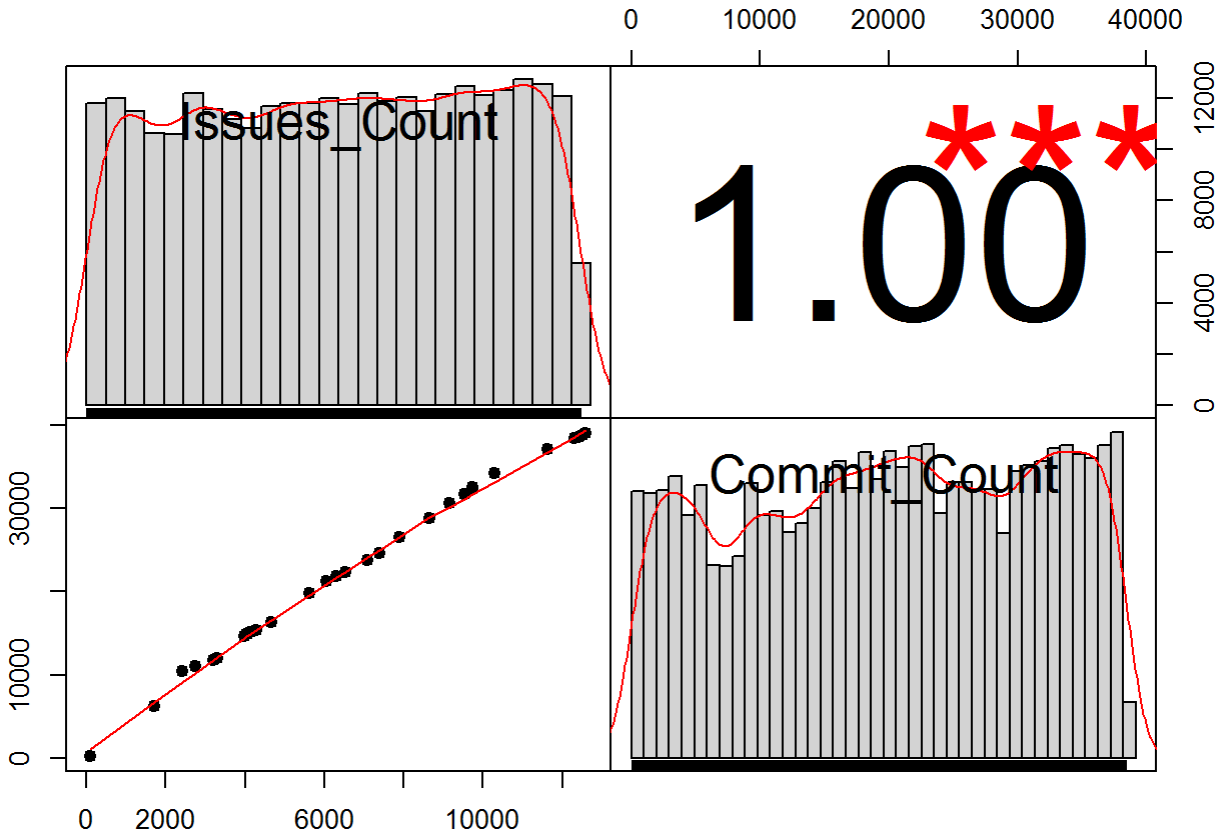
```
#Spearman
Cor_matrix_spearmen <- cor.test(join$Issues_Count, join$Commit_Count,
                               method = "spearman")
Cor_matrix_spearmen
```

```
##
##  Spearman's rank correlation rho
##
## data:  join$Issues_Count and join$Commit_Count
## S = 0, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
## rho
##   1
```

```
# Ploting correlation
ggscatter(join, x = "Issues_Count", y = "Commit_Count",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "pearson",
          xlab = "Issues Count", ylab = "Commit Count")
```




```
# Another Plot using performance analysis library
join1 <- join[, c(2,3)]
chart.Correlation(join1, histogram=TRUE, pch=19)
```


Issues_Count
1.00***
Commit_Count
0
10000
20000
30000
40000
0
4000
8000
12000
0
10000
30000
0
2000
6000
10000