On Automating Github Topic-based Data Analysis

PGC303E Análise de Dados Aplicada - Prof. Marcelo de Almeida Maia

Students: Mohamed A. Fouad, Antonio Livio Cruz de Mendonca

Abstract

Github offers an API for searching popular *topics* based on its opensource projects classifications. We adopt a notion of a software domain as the interesection between a number of github topics. We question the effectiveness of automating github's topic-based data analysis via a toolkit of small-in-size software utilities in accordance with the unix philopshy, that provide summary statistics regarding software domains. We report our developement progress of the required utilities as well as our preliminary results, showcasing an exemplary topic-based analysis.

Introduction

Github offers an API for searching popular *topics* based on its opensource projects classifications, e.g. *https://api.github.com/search/repositories?q=topic:game+topic:go* for searching projects under the joint intersection of the topics game and and go. We question the effectiveness of automating github's topic-based data analysis via a toolkit of small-in-size software utilities in accordance with the unix philopshy, for providing summary statistics regarding software domains. We report our developement progress of the required utilities as well as our preliminary results, showcasing an exemplary topic-based analysis.

We adopt a notion of a software domain as the interesection between a number of github topics and consider summary statistics of such topics and as an example consider the topics "spaceinvaders, game, c" and "pacman, game, c".

Toolkit Development

We report our developement of three utilities that are managed via a Makefile.

domain(1)

A utility for searching github according to a list of topics that constitute a software domain and download a selection of the finding for further investigation locally.

```
domain $domain_name $topicA $topicB $topicC
```

stats(1)

A utility for summary statistics of local git repositories, it accepts a directory of git repositories and the produce a csv.

```
stats $git_repos_dir
```

plot(1)

A utility for plotting stats(1) csv.

```
plot $csv_fullpath
```

Full sourcecode is available at *github.com/moresearch/PGC303E*

Data Sample

Table 1: Table continues below

domain	project	total.files.changed
pacman	aman-txt/PacMan-using-c	6
pacman	iasebsil83/Pacman	30
pacman	MatheusBueno/PacMan	26
pacman	MurphyMc/xchomp	142
pacman	shamiul94/Pacman-Game-using-iGraphics	51
pacman	sobotat/Pacman	770
pacman	yuredev/ringo-game	199
space-invaders	awave1/space-invaders	1817
space-invaders	GoranTopic/Terminal-Space-Adventure	1189
space-invaders	hippopotamus-prime/interlopers	689
space-invaders	rnsavinelli/swbss	102

Table 2: Table continues below

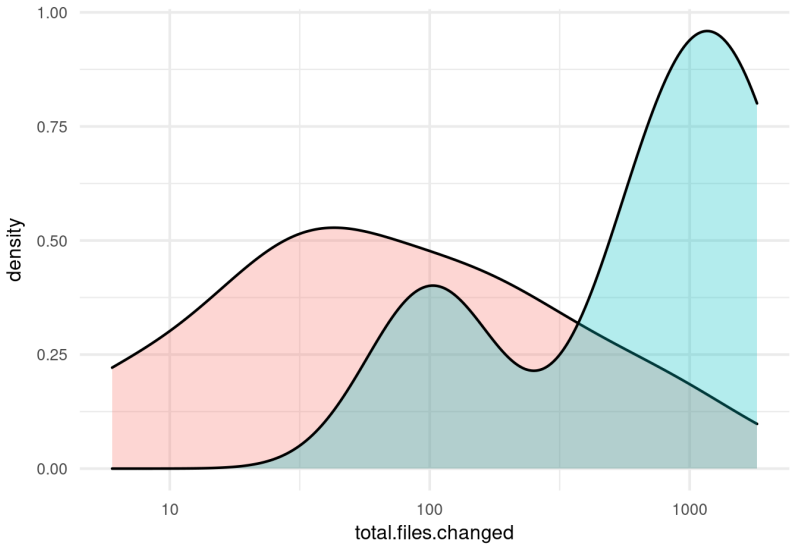
total.lines.added	total.lines.deleted	total.lines
1015	0	1015
2342	2314	28
1911	0	1911
4860	352	4508
11367	2	11365
23071	4797	18274
6578	1860	4718
53306	35653	17653
4875307	4873215	2092
12567	457	12110
5322	1173	4149

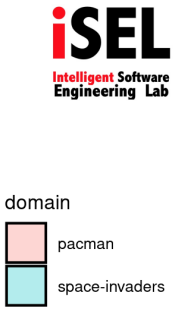
lines.deleted.added.ratio
0
0.988
0
0.07243
0.0001759
0.2079
0.2828
0.6688
0.9996
0.03637
0.2204

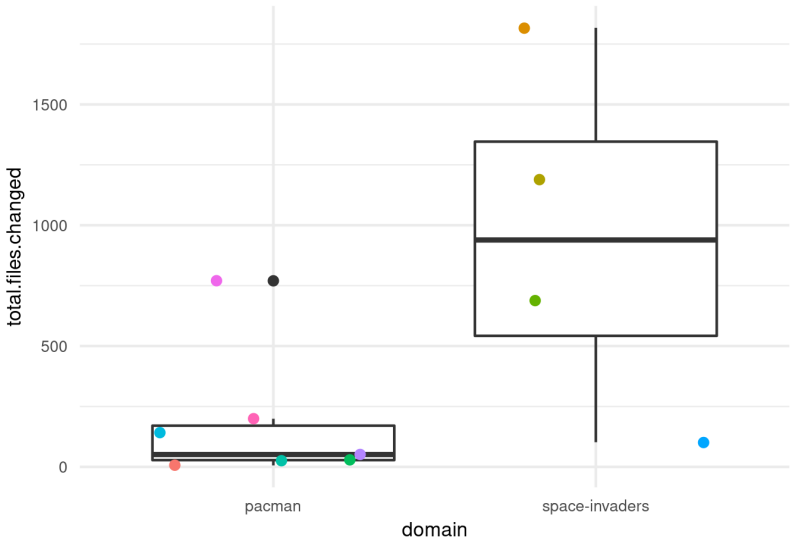
Visualization

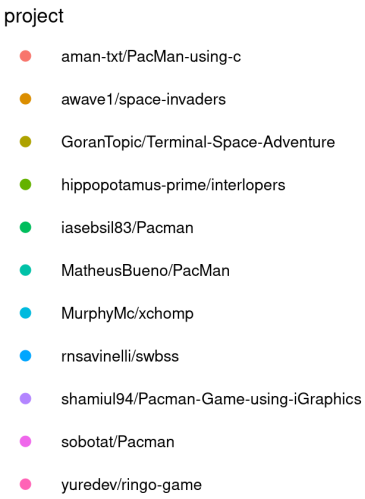
```
# generate a boxplot
boxplot <- function(input){
    newdata<-arrange(data, !!sym(input))
    plot<- newdata%>%
        ggplot(aes(domain, !!sym(input)))+
        geom_boxplot()+
        labs(title = "",subtitle = "", caption = "", x = "domain",y = input) +
            theme_minimal() +
            theme(text=element_text(size = 8))+
            geom_jitter(aes(color=project))
}

# generate a density plot
denplot <- function(input){
    newdata<-arrange(data, !!sym(input))
    plot <- newdata%>%
        ggplot(aes(x=!!sym(input), fill=domain)) +
        geom_density(alpha=0.3)+
        scale_x_log10()+
        theme_minimal() +
        theme(text=element_text(size = 8))
}

# Generate a single plot
plotpage <- function(input){
    path<- paste("FIG/",input,".png",sep="")
    ggsave(filename=path,
           plot=(denplot(input) / boxplot(input)),
           device = png,
           units = "mm")
    plot_with_logo <- add_logo(
                               plot_path = path,
                               logo_path = "FIG/logo.jpg",
                               logo_position = "top right",
                               logo_scale = 10
    )
    magick::image_write(plot_with_logo, path)
}

plotpage("total.files.changed")
plotpage("total.lines.added")
plotpage("total.lines.deleted")
plotpage("total.lines")
```

Total Files Changed

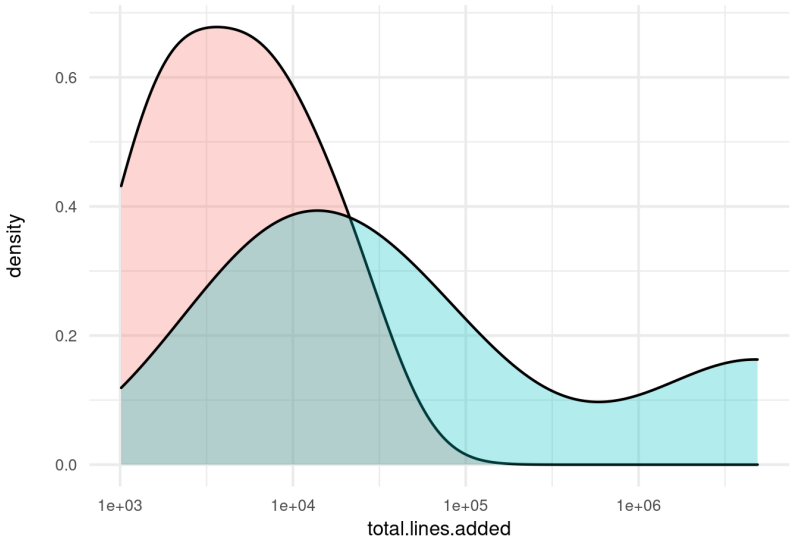


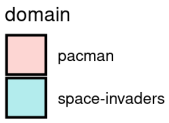


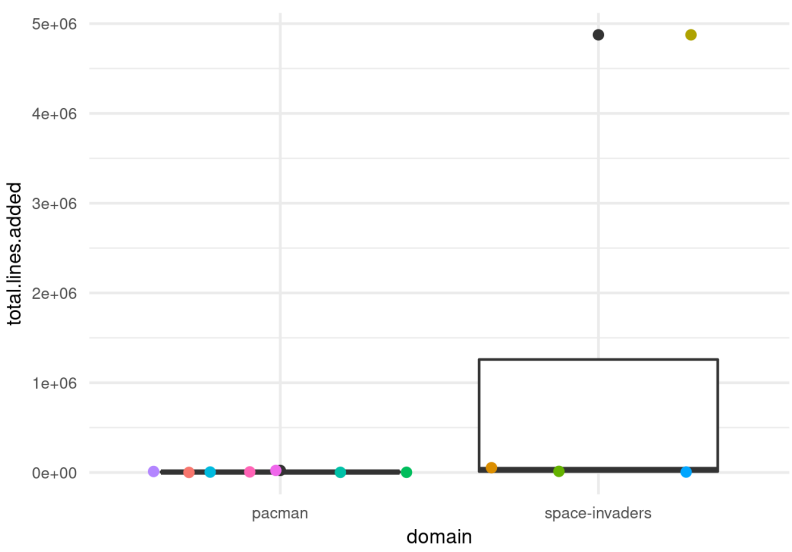


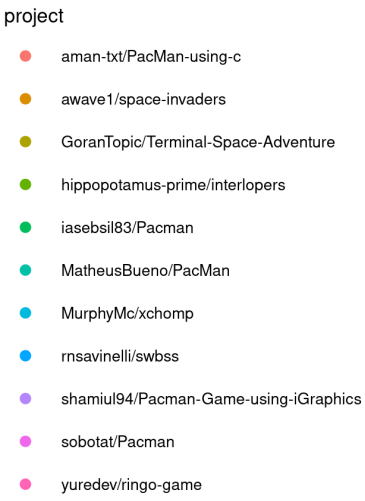


Total Lines Added

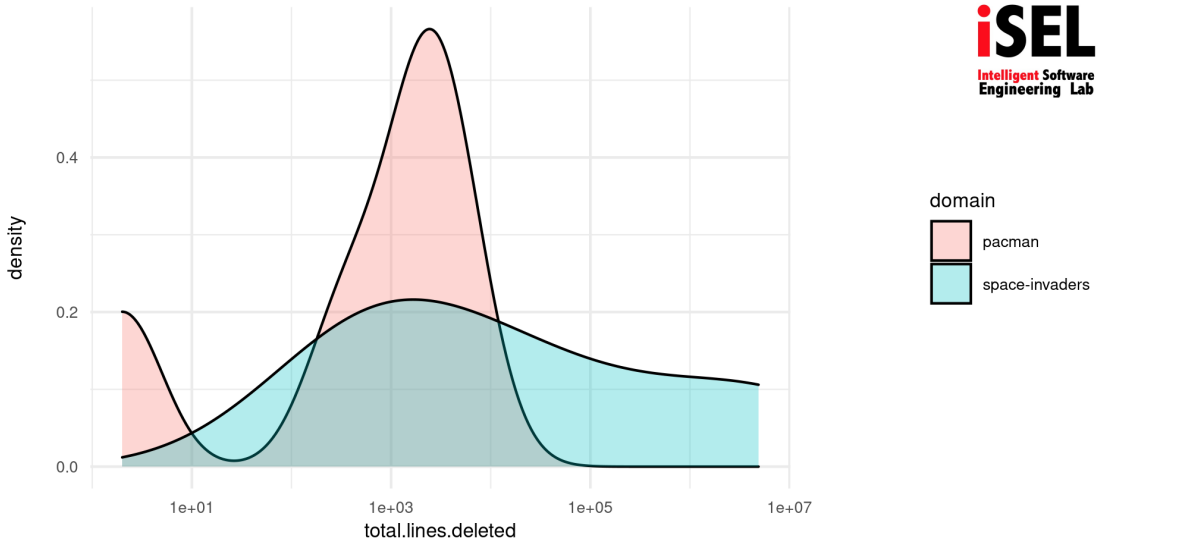


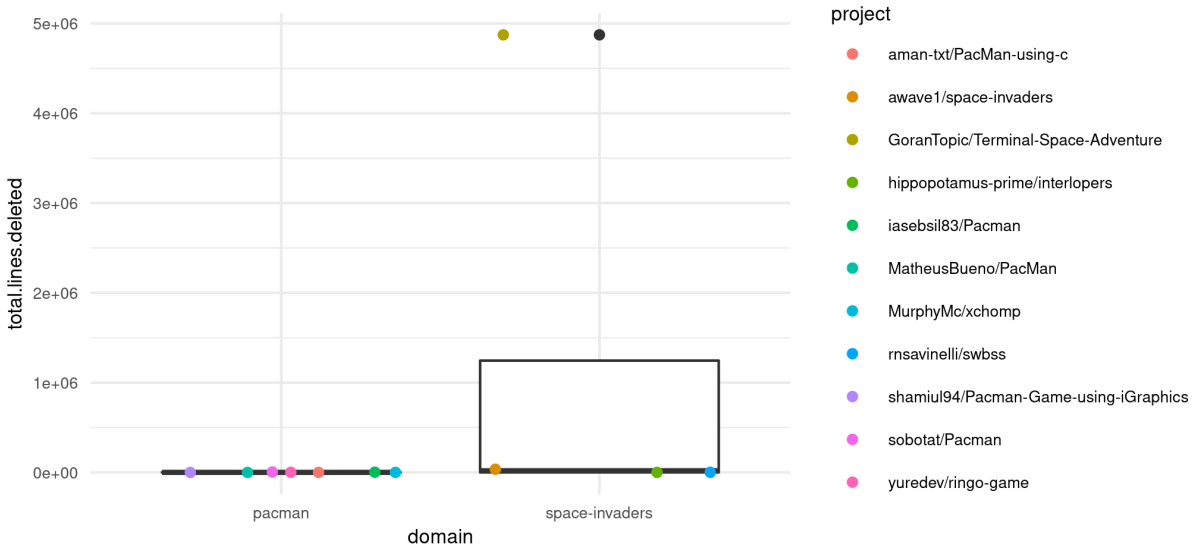




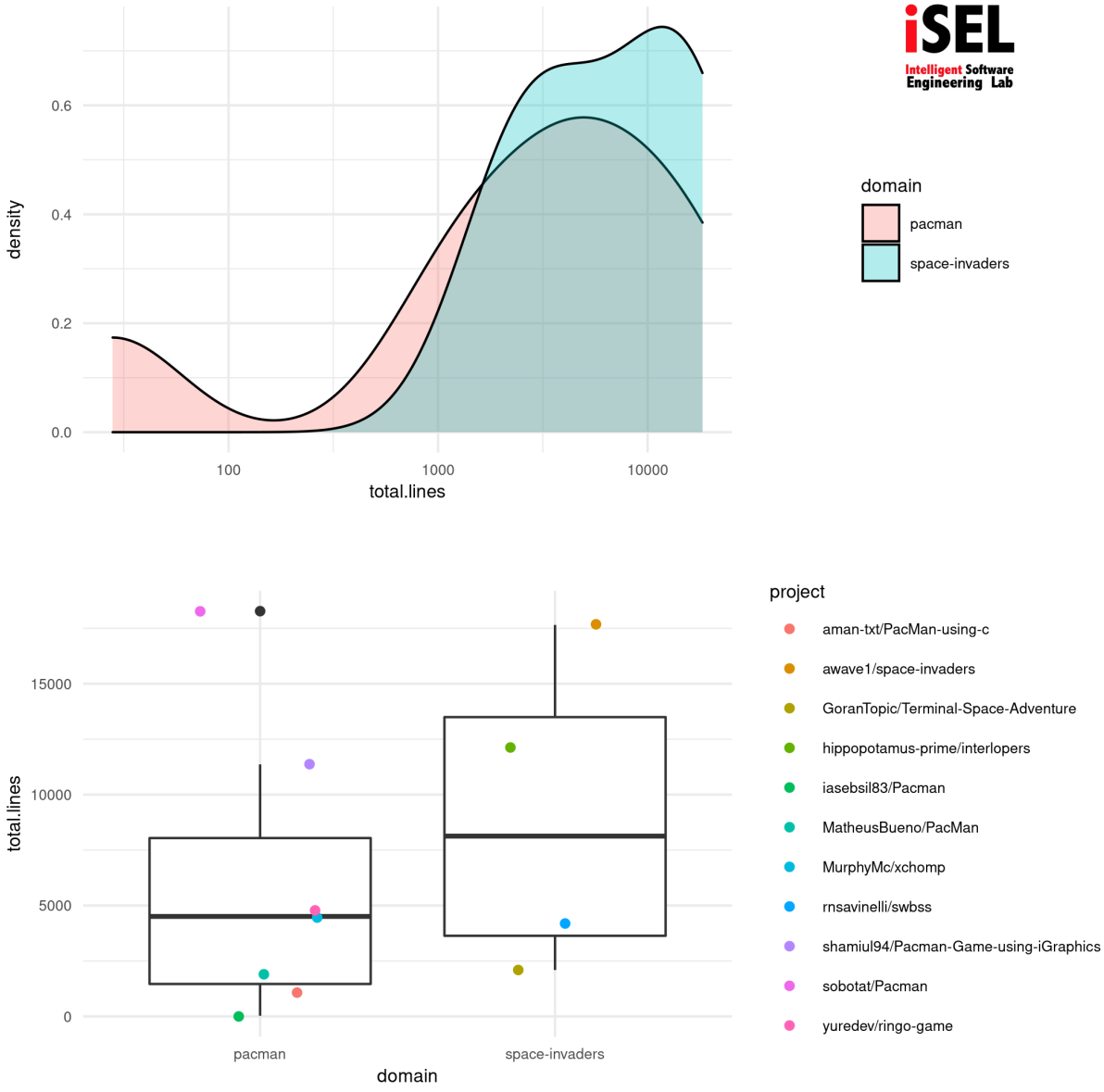


Total Lines Deleted





Total Lines


density
0.0
0.2
0.4
0.6
100
1000
10000
total.lines
domain
pacman
space-invaders
total.lines
0
5000
10000
15000
pacman
space-invaders
domain
project
aman-txt/PacMan-using-c
awave1/space-invaders
GoranTopic/Terminal-Space-Adventure
hippopotamus-prime/interlopers
iasebsil83/Pacman
MatheusBueno/PacMan
MurphyMc/xchomp
rnsavinelli/swbss
shamiul94/Pacman-Game-using-iGraphics
sobotat/Pacman
yuredev/ringo-game

Future work

Further development of toolkit as more usecases arise.

Bibliography

- https://linearb.io/blog/git-statistics
- https://gitential.com/