

Linguagens de Programação

Alisson Rosa

5/22/2021

1 Introdução

Linguagens de programação nós permitem fornecer regras a quais os computadores entendem, ao longo do tempo foram surgindo inúmeras linguagens com propósitos e estruturas diferentes, aqui vamos analisar o comportamento dessas linguagens no GitHub (gh), que hoje é uma das principais plataformas para versionamento de códigos, assim vamos avaliar:

- Quantidade de *issues* em repositórios que contém tais linguagens ao longo do tempo
- Quantidade de *pull requests* (pr) em repositórios que contém tais linguagens ao longo do tempo
- Quantidade totais de repositórios das linguagens.

Todo código a ser desenvolvido aqui será usando a linguagem Julia.

```
include("utils.jl")
using DataFrames
import CSV
using Plots
default(formatter=identity, tickfontsize = 7, titlefontsize=12,
legend=:topleft)
#theme(:vibrant)
using Statistics
using Pipe: @pipe

pr = DataFrame(CSV.File("data/prs.csv"));
issues = DataFrame(CSV.File("data/issues.csv"));
total = DataFrame(CSV.File("data/repos.csv"));
```

Precisamos primeiro notar que até ‘ existiam linguagens disponíveis no gh, assim aqui vamos avaliar as top 5 linguagens e R¹ em quantidade de repositórios

¹R, pois certas pessoas curtem R, certo.. Certo??

Linguagem	Total de Repositórios
JavaScript	1100421
CSS	813443
HTML	779549
Shell	638068
Python	548870

Tabela 1: Top 5 Linguagens em quantidade de repositórios.

Notamos que as top 3 linguagens, são de desenvolvimento web, sendo elas JavaScript, CSS, HTML e no top 4 está o famoso shell script, e por último Python. No seguinte gráfico vamos ver como se comporta R em frente a esses gigantes.

```
@pipe total |>
  sort(_, order(:num_repos, rev=true)) |>
  first(_, 5) |>
  select(_, :language => :Linguagem, :num_repos => :N) |>
  df -> bar(df.Linguagem, df.N, fill=["#f3ff33", "#337aff",
    "#ff5e33", "#55ff33", "#233cad"],
    series_annotation=df.N, legend=false)

@pipe total |>
  filter(:language == ("R"), _) |>
  df -> bar!(df[:, :language], df[:, :num_repos], legend=false)
```

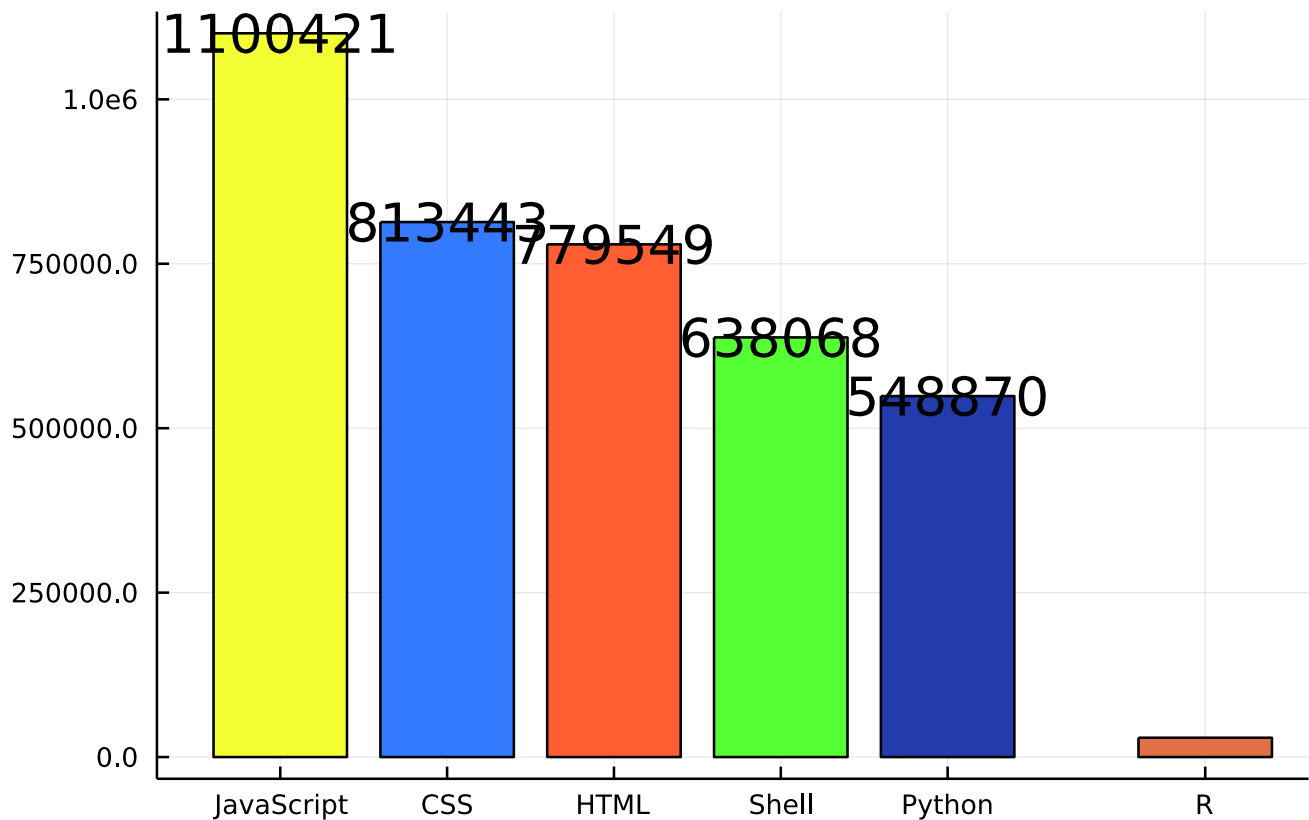


Figura 1: Top 5 Linguagens em quantidade de repositórios e R

2 Análise descritiva

Primeiro vamos começar avaliando o comportamento médio de prs e issues no gh ao longo dos anos. Notamos alguns pontos fundamentais, tanto prs e issues crescem até atingir seu ápice em 2017, mas as issues diferentes dos prs não descresem de forma monótona depois de 2017.

```
p1 = @pipe top_5(issues,:year) |>
      areaplot(_[:,:year],_[:,:media], title="pr", label= "prs", legend=:topleft)
p2 = @pipe top_5(pr,:year) |>
      areaplot(_[:,:year],_[:,:media], title="issues", color = "#ff4933",
      label= "issues", legend=:topleft)

plot(p1,p2,layout= (1,2))
```

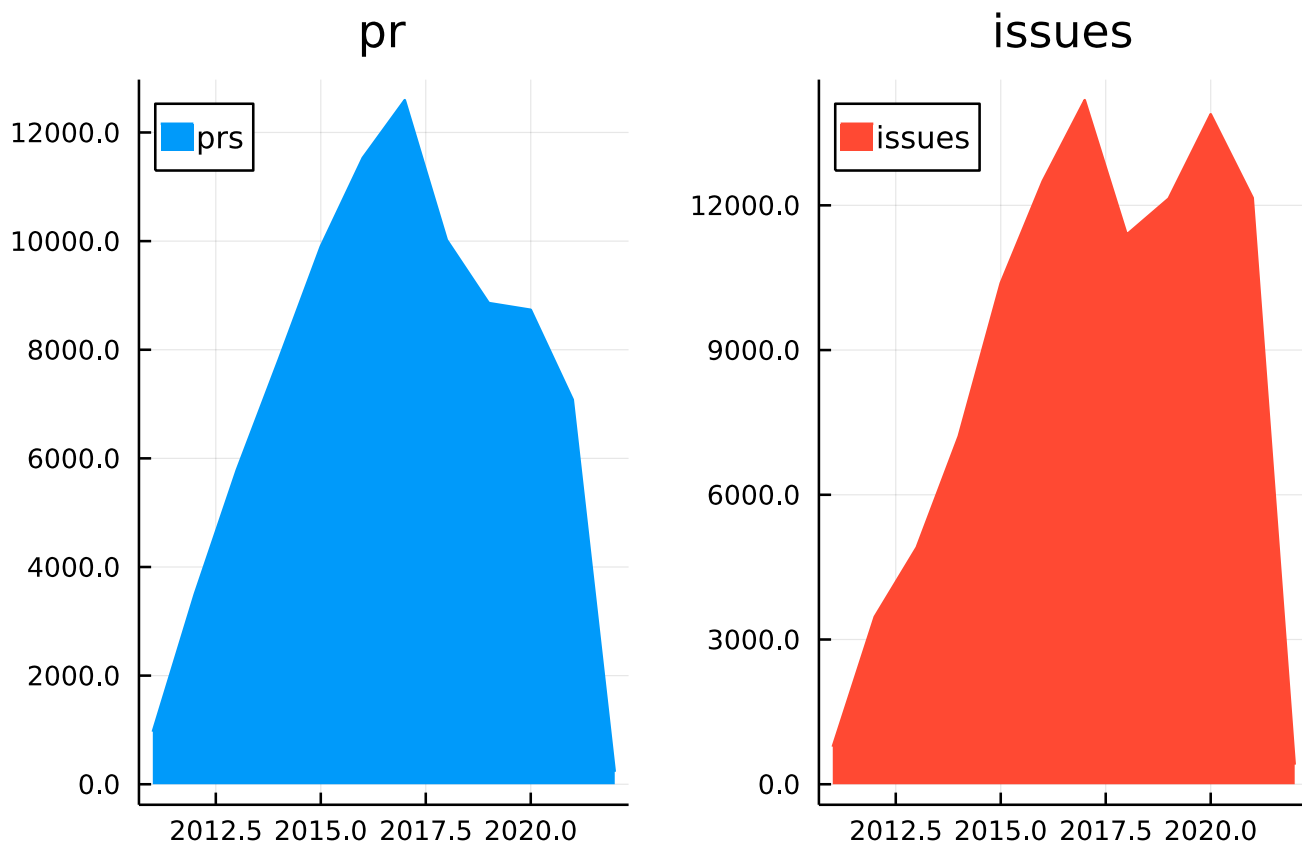


Figura 2: Comportamento médio ao longo dos anos

Mas no gráfico ref, avaliamos o comportamento das issues e prs no gh de forma global, independente da linguagem, será que as linguagens individualmente se comportam da mesma maneira? É isso que veremos nas seções seguintes

3 Linguagens Globalmente

Aqui vamos avaliar o comportamento de 5 linguagens: Java, JavaScript, PHP, Python, Ruby. Pelo gráfico xxx, podemos ter um vislumbre do comportamento de prs para essas linguagens ao longo do tempo

```
fav_langs(pr, ["Java","Python", "JavaScript", "Ruby", "PHP"]) |>
  df -> plot(df[:,year], df[:,media], g = df[:,name])
```

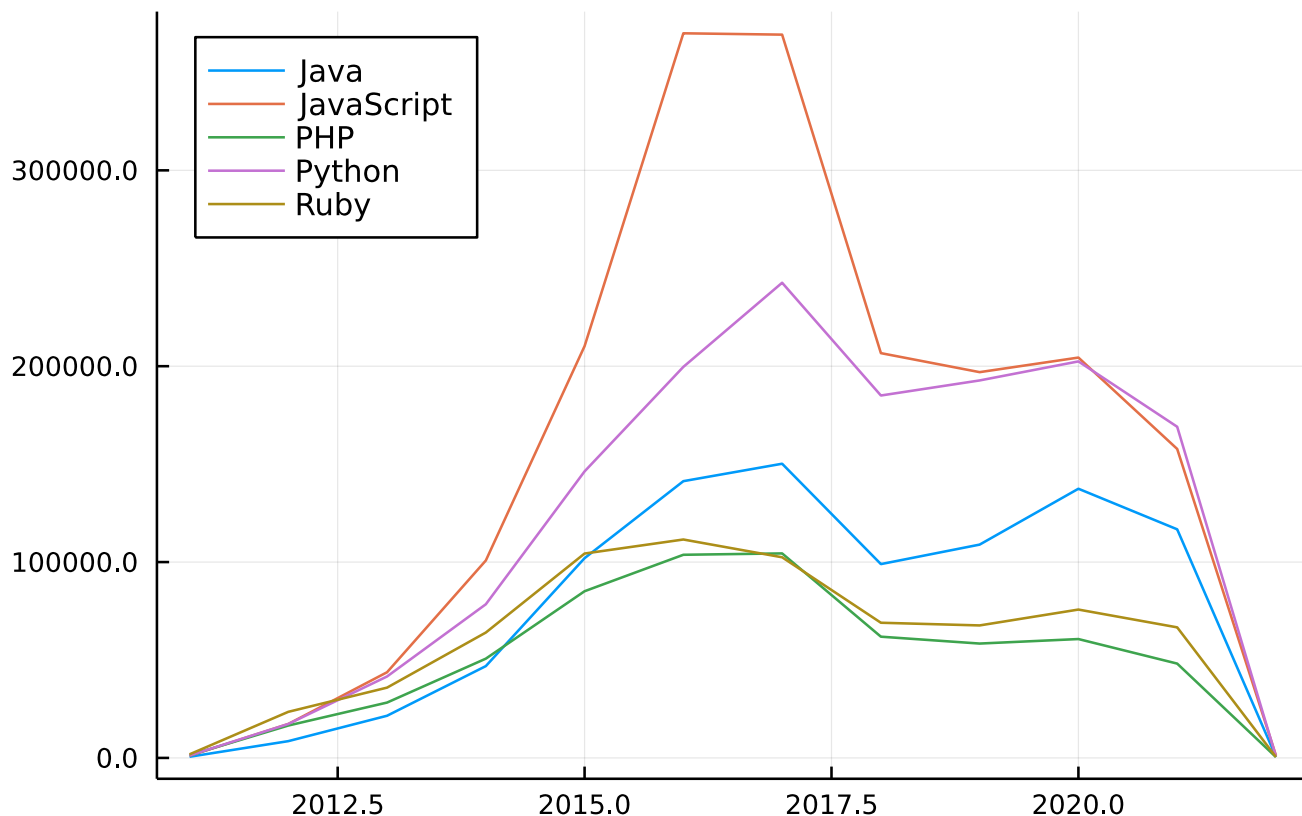


Figura 3: Comportamento médio de prs

Note que JavaScript domina desde o começo, porém Python avançou muito nos últimos anos, e em 2021 finalmente ultrapassou JavaScript, outra linguagem que andou tendo um decréscimo considerável é PHP. Vale ressaltar o pico de todas as linguagens em 2017, e depois um leve decréscimo, mas somente PHP e Ruby continuaram caindo consideravelmente nos últimos anos.

Pelo gráfico, nota-se que Ruby tem um comportamento diferente em termos de issues que em prs, pois fica abaixo de todas as linguagens praticamente em todos os anos, Java para issues está mais próximo de Python do que estava em termos de prs, e Java depois de 2017 só decresce em issues, algo que não aconteceu em prs.

```
fav_langs(issues, ["Java","Python", "JavaScript", "Ruby", "PHP"]) |>
  df -> plot(df[:,year], df[:,media], g = df[:,name])
```

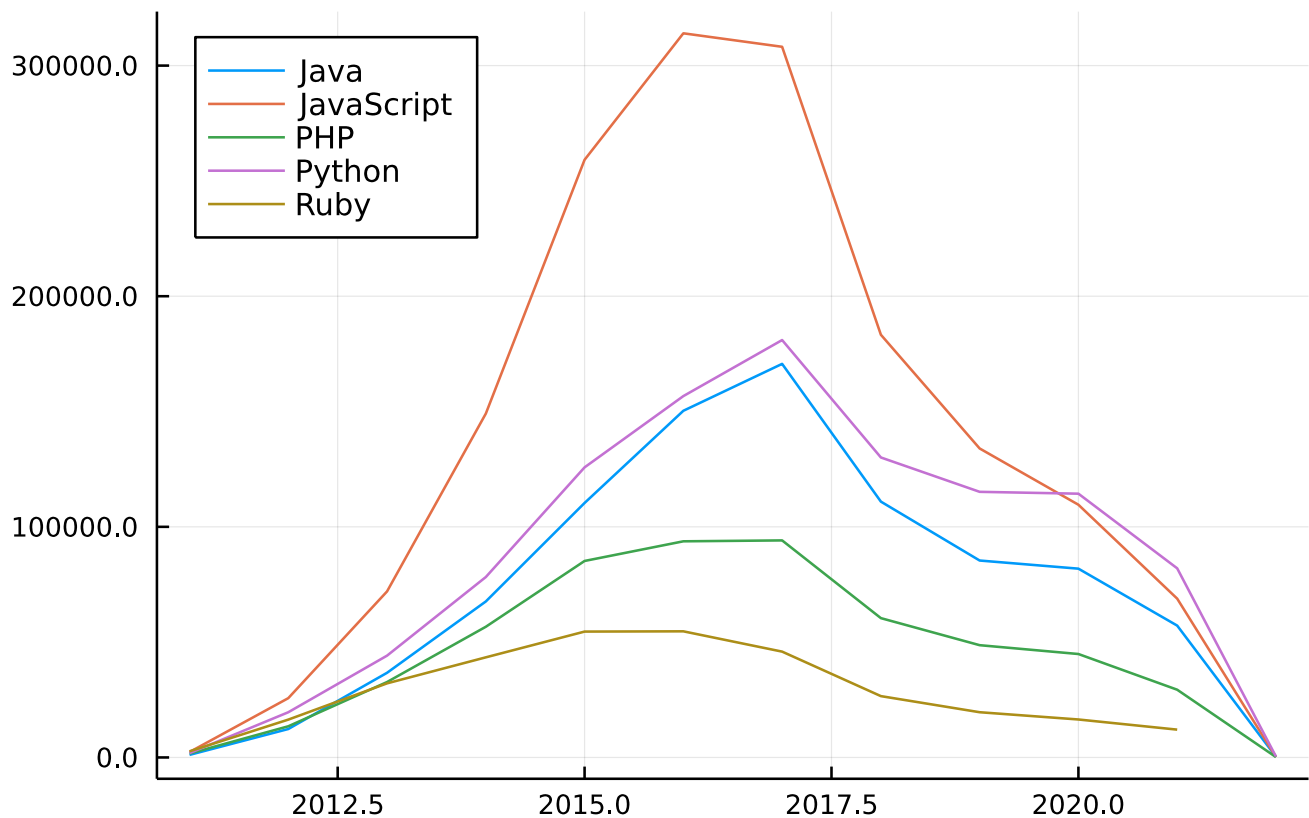


Figura 4: Comportamento médio de issues

Notamos portanto um comportamento bem claro, prs cresceram até atingir seu apogeu em 2017, aonde depois, em geral tende a cair, a única das linguagens das estudadas que recuperou crescimento foi Java. As issues em uma análise global tinham crescimento mesmo após 2017, porém nas linguagens estudadas nessa seção, não foi o que aconteceu.

4 Linguagens para Dados

Nessa seção vamos avaliar o comportamento especificamente de R e Julia, que são duas das três principais linguagens para trabalhar com dados no momento.

4.1 Breve Introdução

R é uma implementação open-source da Linguagem S teve seu lançamento oficial em 1995, sendo desenvolvida por [Ross Ihaka](#) e [Robert Gentleman](#), sendo assim seu nome “R” vem da letra inicial do seus criadores e também do dialeto com “S”. R tem suas raízes na estatística computacional, passou seus 10 primeiros anos sendo primariamente uma linguagem acadêmica mas depois do desenvolvimento de certos pacotes se tornou um ambiente propicio para ciência de dados.

O projeto da linguagem Julia teve seu início nos meados de 2009, onde foi lançada oficialmente em 2012, sendo assim, no momento que o autor vos escreve Julia está fazendo 10 anos, é uma linguagem que tem a característica de ser **rápida** e **fácil** de usar, mesmo com somente 10 anos de idade Julia já é usada pela NASA e também pelo INPE. Vamos começar avaliando a quantidade total de repositórios para cada uma das duas linguagens.

Podemos ver que R possui muito mais repositórios disponíveis no gh, porém R é uma linguagem que está a bastante tempo no mercado, Julia possui apenas 10 anos de idade..

Agora vamos ver o comportamento das duas linguagens ao longo dos últimos anos

```
p1 = fav_langs(pr, ["R","Julia"]) |>
  df -> plot(df[:,year], df[:,media], g = df[:,name])
p2 = fav_langs(issues, ["R","Julia"]) |>
  df -> plot(df[:,year], df[:,media], g = df[:,name])

plot(p1,p2,layout=(1,2))
```

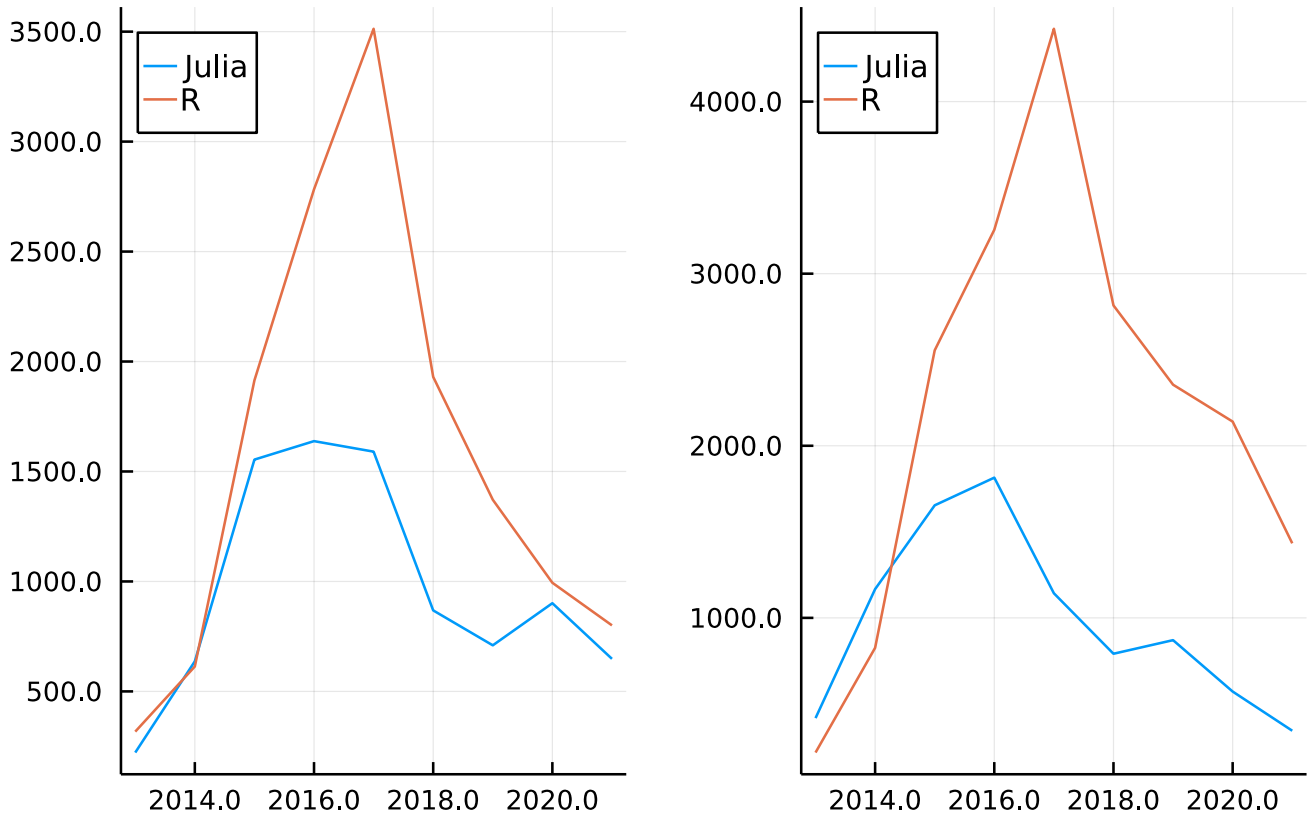


Figura 5: Média de prs ao longo dos anos

Nota-se que em 2014 Julia teve mais prs que R e ambas as linguagens atingem seu pico em 2017, um comportamento global como vimos anteriormente, porém para issues Julia teve mais de 2012 até 2014 e

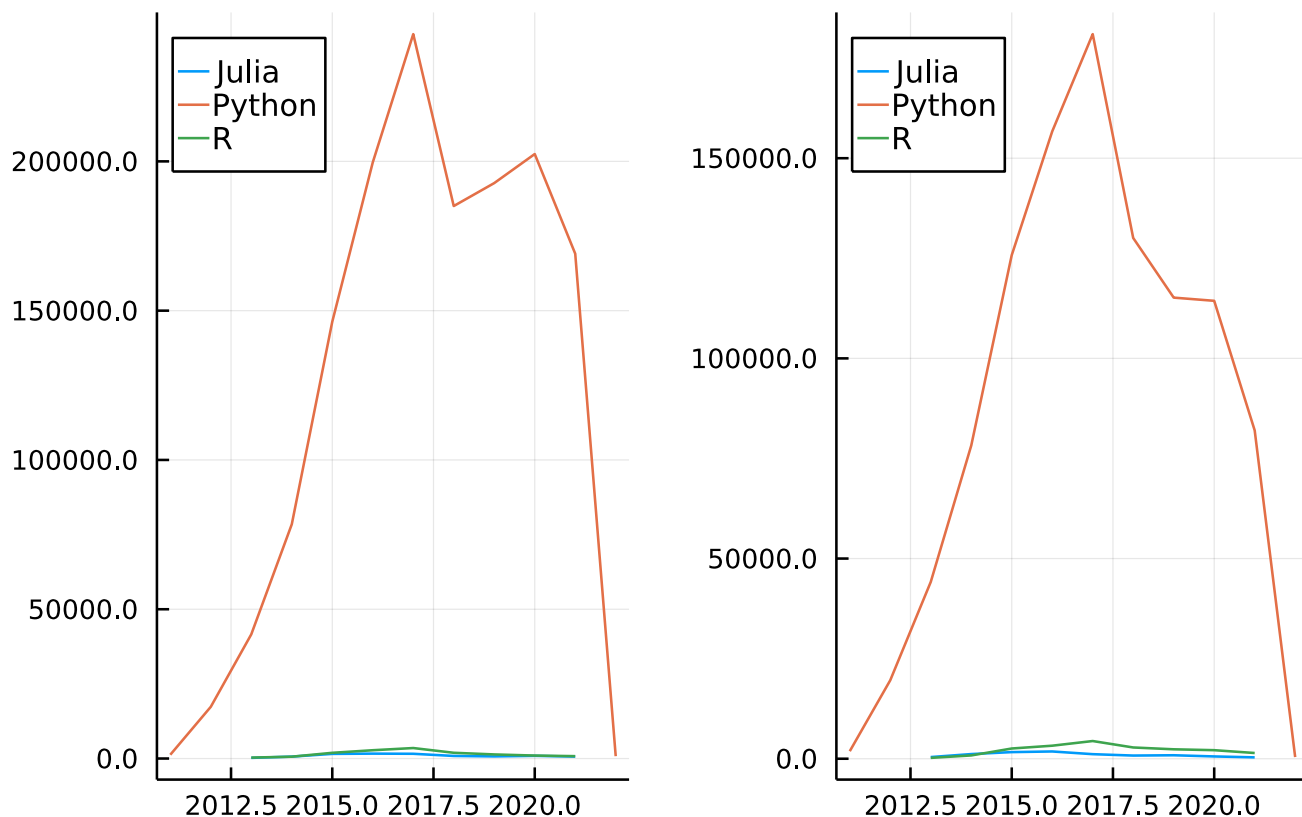
também seu pico de issues foi em 2016 e não 2017 como R.

4.2 Python entra no game

Evidentemente não podemos esquecer de Python, pois como vimos está no top5 de linguagens mais usadas e também ———,

```
p1 = fav_langs(pr, ["R","Julia","Python"]) |>
  df -> plot(df[:,year], df[:,media], g = df[:,name])
p2 = fav_langs(issues, ["R","Julia", "Python"]) |>
  df -> plot(df[:,year], df[:,media], g = df[:,name])

plot(p1,p2,layout=(1,2))
```



Assim pela figura notamos que: Uma imagem vale mais que mil palavras

5 Conclusão momentânea

Referências