Lab1 - Milestone 4

Bruno Dias - contato@diasbruno.com

27 de março de 2016

Esse relatório tem como objetivo analisar os dados do *github* dos anos de 2012-2015. Os pontos abordados serão quanto a produtividade dos usuários do *github*, além de entender como se comportaram as linguagens em termo de crescimento e declínio durante o período.

Perguntas sobre o *Github*

As perguntas que foram feitas são as seguintes:

1. Que linguagens que tiveram um crescimento e um declínio de eventos?
2. Quais os trimestres mais produtivos por linguagem?
3. Existe um aumento geral de eventos (todos os tipos de eventos e todos os tipos de linguagens) em um determinado trimestre?
4. Existe correlação entre a quantidade de *forkEvents* e *watchEvents*? Se sim, é baixa ou não?

Dados analisados

Para a análise dos dados foi utilizado o data frame disponibilizado que tem seis variáveis, sendo elas:

- **Linguagem**: linguagem que observada.
- **Tipo de evento**: Tipos de eventos do *github*, existem 5 na amostra(*Issues, Push, Fork, Watch, Create*).
- **Repositório ativo**: Repositórios ativos no momento da observação.
- **Quantidade de eventos**: Quantidade de eventos daquele tipo.
- **Ano**: Ano da observação.
- **Trimestre**: Trimestre observado - 1(Jan-Mar), 2(Abr-Jun), 3(Jul-Set), 4(Out-Dez).

Dessas variáveis só não será utilizada na análise a quantidade de repositórios ativos.

Conhecendo a amostra

Nossa amostra conta com 225 linguagens diferentes e também é possível perceber que o ano de 2015 só conta com o primeiro trimestre, isso dificulta as análises de algumas perguntas feitas. Sendo assim, para todas as análises o ano de 2015 foi ignorado.

Uso das linguagens no *Github*

Agora, iremos responder se existiu alguma linguagem que obteve um crescimento seguido de um declínio. Para ajudar na visualização dos dados, foram ignoradas as linguagens que tiveram pelo menos 1 ano sem atividades.

Primeiro passo, é agrupar os eventos das linguagens por ano.

```
# Agrupando os eventos das linguagens por ano e excluindo o ano de 2015
usoLinguagensAno <- githubDF %>%
  group_by(repository_language, year) %>%
  summarise(qtd_events= sum(events)) %>%
  filter(year < 2015)
```

Em seguida, é necessário retirar as linguagens que não tiveram atividades em pelo menos 1 ano.

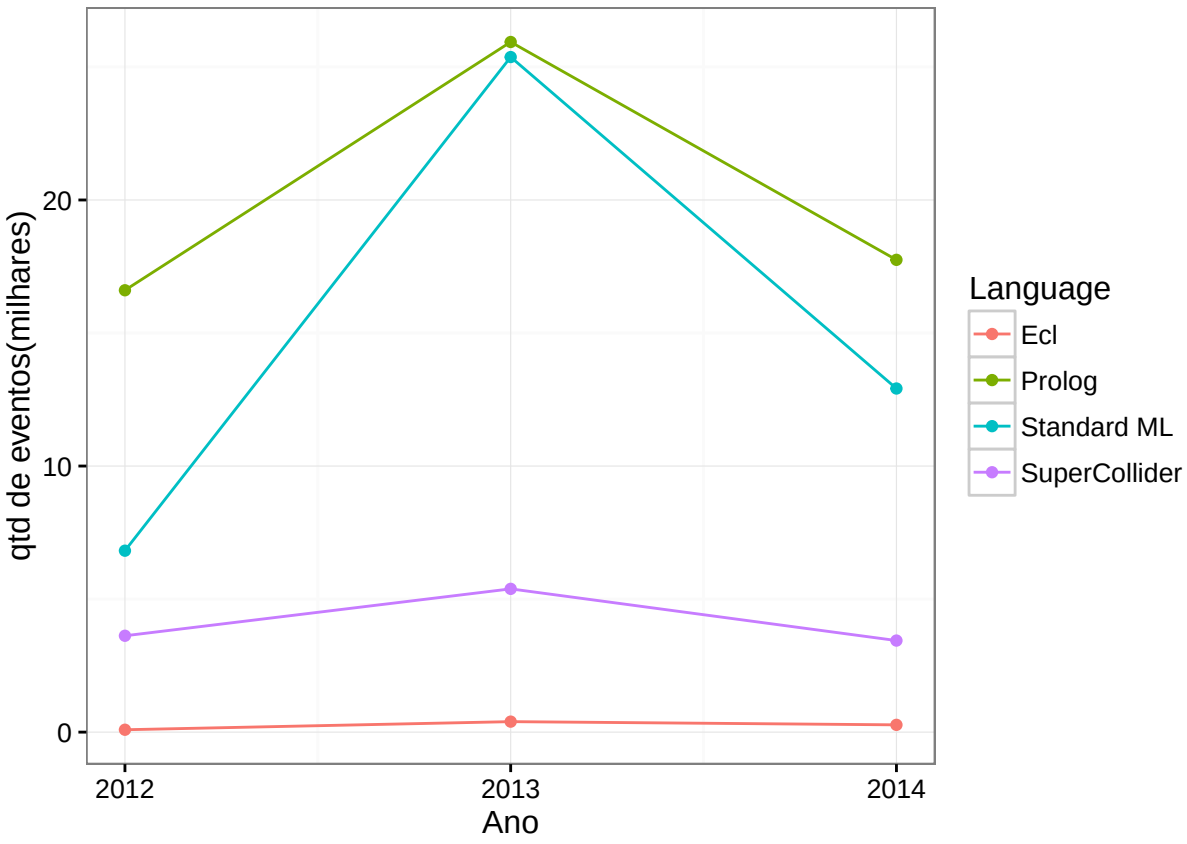
```
# Exclusão de linguagens que não tiveram atividades em pelo menos 1 ano
linguagensAtivas <- usoLinguagensAno %>%
  count(repository_language) %>%
  filter(n >= 3 )

# Reorganizando o dataset inicial para retirar as linguagens que não tiveram atividade
# no período
linguagensUsadas <-
  merge(x=usoLinguagensAno, y=linguagensAtivas, by="repository_language", all=TRUE) %>%
  na.omit(linguagensUsadas) %>% select(-n)
```

Dentre as 225 linguagens que existiam no *data frame* inicial , 91 linguagens tiveram eventos em todos os anos. Agora é preciso recuperar as linguagens que tiveram o crescimento maior em 2013.

```
# Recuperando somente as linguagens que tiveram de crescimento em 2013 e caíram em seguida
linguagensResult <-linguagensUsadas %>%
  group_by(repository_language) %>%
  arrange(desc(qtd_events)) %>%
  distinct(repository_language) %>%
  filter(year == "2013")
linguagensResult %>% select(repository_language)
```

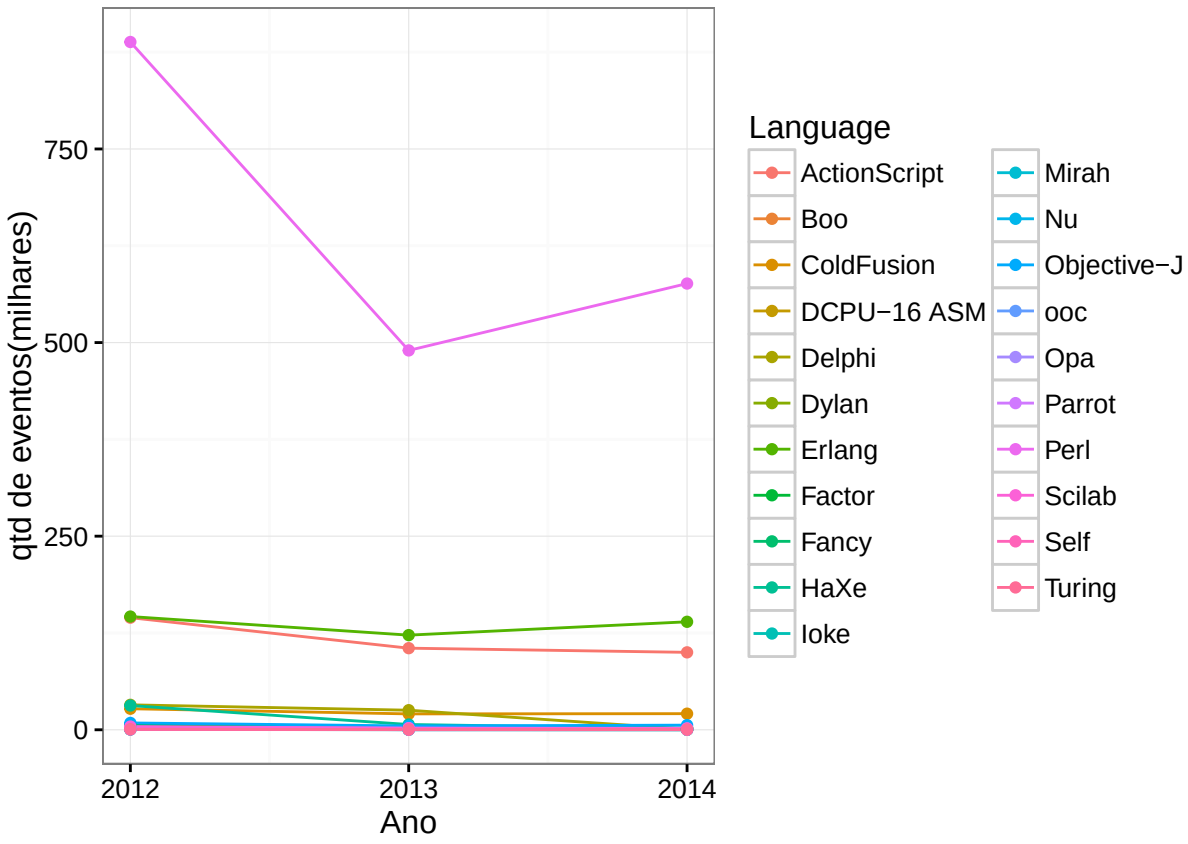
```
Source: local data frame [4 x 1]
Groups: repository_language [4]

  repository_language
               (fctr)
1                 Ecl
2              Prolog
3         Standard ML
4       SuperCollider
```

Em seguida, iremos exibir o gráfico de produtividade no período analisado.



Como vimos, poucas linguagens tiveram um crescimento seguido de um declínio no período. Isso mostra que houve um engajamento maior em 2013 de uso dessas linguagens e, em seguida, os programadores diminuíram o uso delas. Por fim, como houve poucas linguagens que tiveram essa oscilação, podemos concluir que isso é algo incomum dentro do nosso conjunto de dados.

Para complementar a análise, vamos analisar as linguagens que decaíram no período.



Como podemos ver, ao total 21 linguagens tiveram um declínio no período, apesar de algumas delas terem se recuperado um pouco entre 2013 e 2014, ainda sim sofreram uma diminuição de uso no geral. *Perl* é uma das linguagens que tiveram uma queda considerável entre os anos de 2012 e 2013 e seguida de uma pequena recuperação em 2014.

Por fim, da nossa amostra de 91 linguagens, 25 delas sofreram um declínio ao longo do período, algumas tiveram uma recuperação, mas pode ser que estejam começam a entrar em desuso. Para isso, poderíamos investigar os eventos do tipo *Create* e *Fork*, o qual deixamos para um trabalho futuro.

Trimestres mais produtivos do ano por linguagem

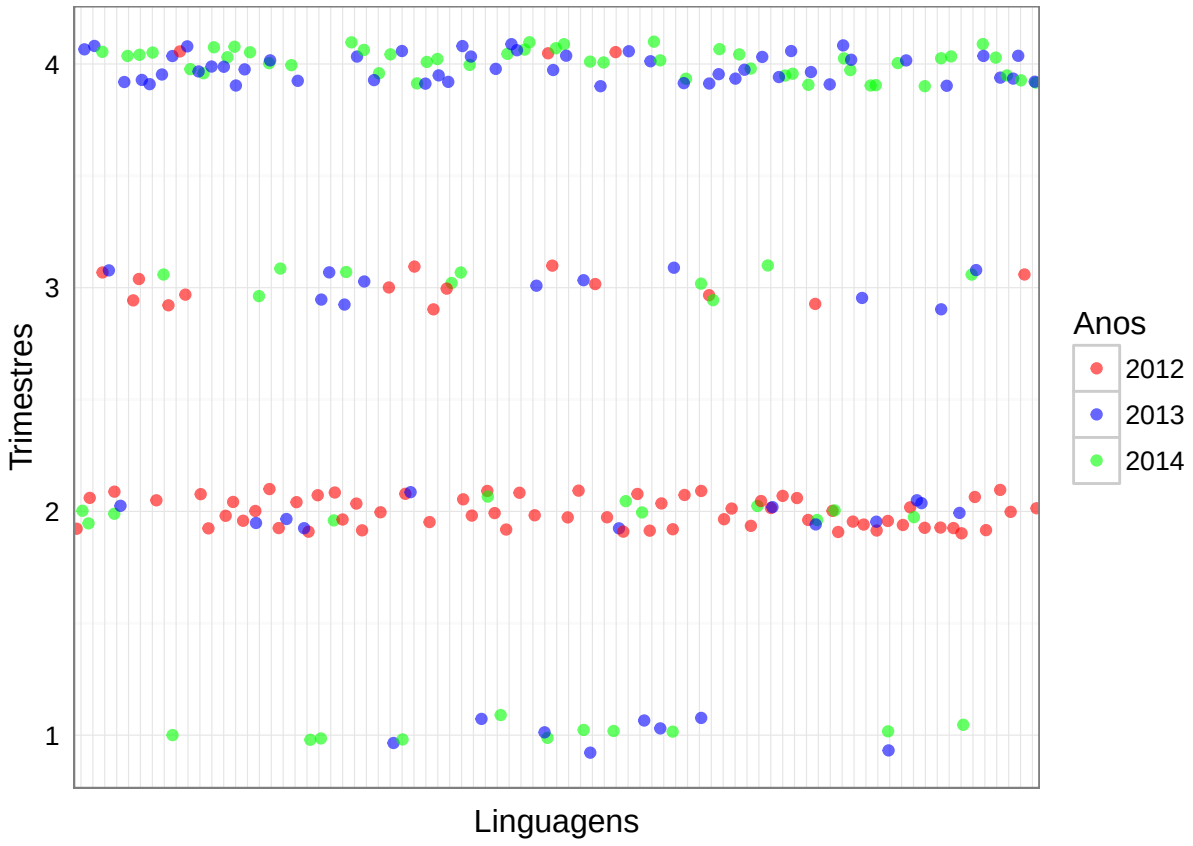
Para analisarmos os trimestres mais produtivos devemos primeiro agrupar os eventos por trimestre. Em seguida, excluímos as linguagens que não tiverem pelo menos 1 evento por trimestre.

```
# Agrupando os eventos das linguagens por trimestre e excluindo o ano de 2015
usoLinguagensTri <- githubDF %>%
  group_by(repository_language, year, quarter) %>%
  summarise(qtd_events= sum(events)) %>%
  filter(year < 2015)

# Exclusão de linguagens que não tiveram atividades em pelo menos 1 trimestre no período.
linguagensAtivasTri <- usoLinguagensTri %>%
  count(repository_language) %>%
  filter(n >= 12)

# Reorganizando o dataset inicial para retirar as linguagens que não tiveram atividade
# no período
```

```
linguagensUsadasTri <-
  merge(x=usoLinguagensTri, y=linguagensAtivasTri, by="repository_language", all=TRUE) %>%
  na.omit(linguagensUsadas) %>% select(-n)

# Definindo os trimestres mais produtivos no período por linguagem.
trimestresMaisProdutivos <- linguagensUsadasTri %>%
  group_by(repository_language) %>%
  arrange(desc(qtd_events)) %>%
  distinct(year)
```

Agora, temos os dados necessários para visualizarmos quais foram os trimestres mais produtivos por linguagens, segue abaixo o gráfico que mostra os trimestres que tiveram mais eventos por linguagem.

```
ggplot(trimestresMaisProdutivos, aes(x=repository_language, y=quarter)) +
  xlab("Linguagens") + ylab("Trimestres") +
  geom_jitter(height=0.25, alpha=0.6, aes(color=as.character(year))) +
  scale_color_manual(name="Anos", labels=c("2012","2013","2014"),
                     values=c("red","blue","green"))+
  theme_bw() +
  theme(axis.text.x=element_blank(),
        axis.ticks=element_blank())
```



No gráfico, percebemos que no ano de 2012 o segundo trimestre foi o trimestre em que o *Github* foi mais utilizado pelos usuários nas linguagens da amostra. Curiosamente, nesse mesmo ano, não existiu nenhuma linguagem que teve o primeiro trimestre como o mais produtivo.

Também podemos perceber que os anos seguintes houve uma concentração maior de produtividade nas linguagens no quarto trimestre.

Por fim, podemos afirmar que houve uma concentração muito maior de eventos nos 2 e 4 trimestres dos anos avaliados da amostra. Isso contraria a suposição inicial que tinha sido feita na atividade, onde dizia que os trimestres mais produtivos seriam 2 e 3.

É interessante ver também que o primeiro trimestre do ano tem uma produtividade muito menor que os demais.

Trimestre com mais eventos do período

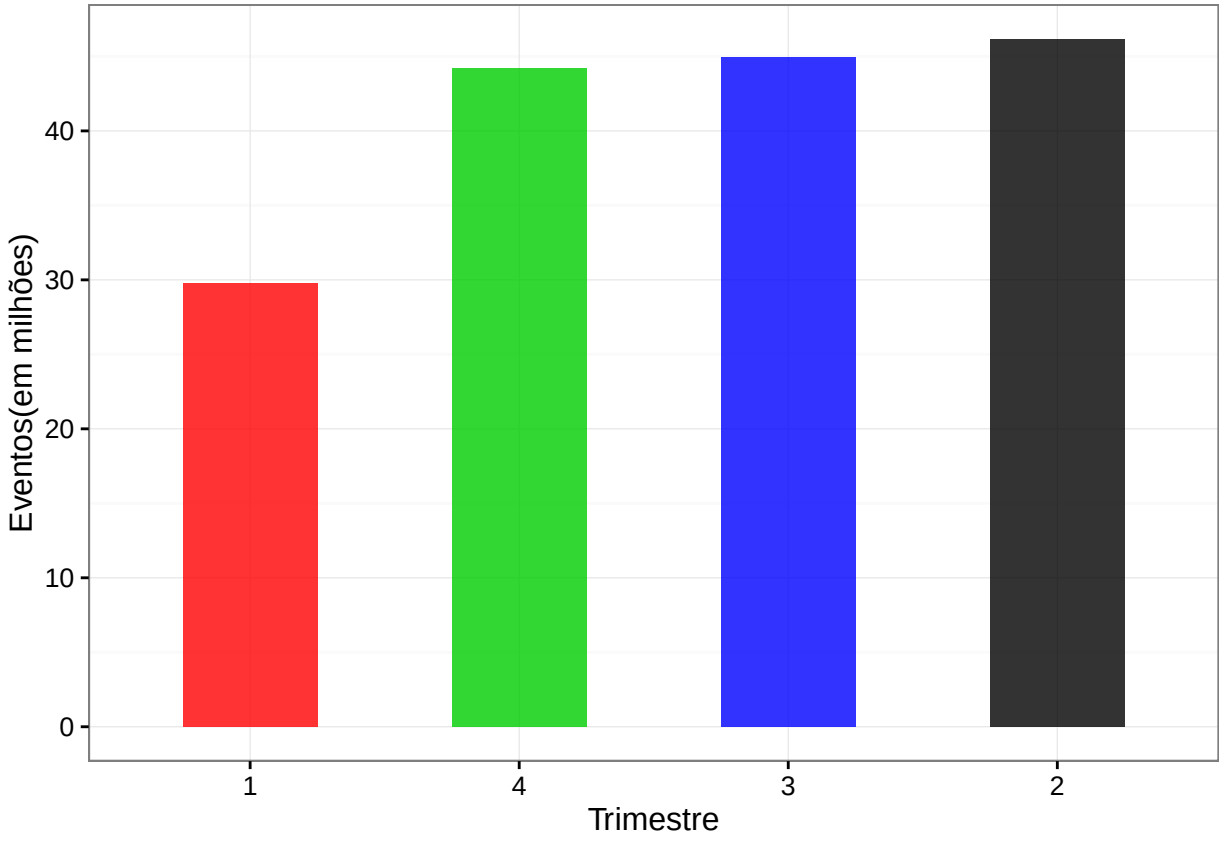
Agora iremos analisar qual o trimestre que o github foi mais utilizado. Para isso, precisamos agrupar os dados por trimestre, somando todos os eventos de cada semestre, assim temos:

```
## Source: local data frame [4 x 2]
##
##   quarter   events
##     (int)    (int)
## 1       2 46170524
## 2       3 44920333
## 3       4 44202284
## 4       1 29783606
```

Podemos perceber a grande diferença entre o primeiro trimestre e os demais, assim confirmamos que o primeiro trimestre tem uma produtividade muito menor que os demais.

Por outro lado, curiosamente, o terceiro trimestre foi o segundo que teve mais eventos, algo que não foi possível perceber na análise de pontos que foi feita anteriormente. Abaixo temos um gráfico ordenado que mostra a quantidade de eventos por trimestre.

```
ggplot(EventosTotaisTri, aes(reorder(quarter, events), y=events/1e+6)) +
  xlab("Trimestre") +
  ylab("Eventos(em milhões)") +
  geom_bar(size=.2, stat="identity",
           alpha=0.8, width = 0.5,
           fill=EventosTotaisTri$quarter) +
  theme_bw()
```



4. Existe correlação entre a quantidade de *forkEvents* e *watchEvents*? Se sim, é baixa ou não?

Relação entre eventos do tipo *Fork* e *Watch*

Por fim, iremos analisar se existe correlação entre os eventos do tipo *Fork* e *Watch* nos trimestres e por linguagem. Para isso, foi necessário agrupar esses eventos por trimestre e por linguagem. Em seguida tivemos que dividir a quantidade de eventos por repositórios ativos, assim temos a quantidade de eventos normalizada. Foi preciso também retirar as linguagens que não tiveram eventos em todos os trimestres.

```
# Agrupando os a quantidades de eventos Fork e Watch por trimestre e por linguagem
EventosPorTrimestre <- githubDF %>%
  group_by(repository_language, year, quarter, events/active_repos_by_url) %>%
  filter(year < 2015) %>%
  filter(type == "ForkEvent" | type =="WatchEvent" ) %>%
  arrange(desc(repository_language))

# Excluindo linguagens que não tem eventos em pelo menos 1 trimestre
linguagensAtivasTri <- EventosPorTrimestre %>%
  count(repository_language) %>%
  filter(n >= 24)

# Reorganizando o dataset inicial para retirar as linguagens que não tiveram atividade
# no periodo
linguagensUsadasTri <-
  merge(x=EventosPorTrimestre, y=linguagensAtivasTri,
        by="repository_language", all=TRUE) %>%
```

```
  na.omit(linguagensUsadas) %>% select(-n)

# Agrupando os tipos de eventos por repositorio e normalizando
Qtd_eventosTrimestre <- linguagensUsadasTri %>%
  group_by(repository_language, year, quarter, type) %>%
  summarise(events=sum(events/active_repos_by_url))
```

Por fim, faremos o cálculo da correlação entre os dois tipos de eventos por linguagem, como temos variáveis contínuas será usado o método de *Pearson*. Como podemos ver, existe uma correlação entre os dois eventos, porém é uma correlação moderada.

```
[1] 0.6814071
```

Essa correlação não nos mostra uma associação de causa e efeito entre os eventos analisados, porém mais uma vez confirma que ao longo dos anos o *Github* vem crescendo e existe uma relação de crescimento dentro os eventos. Em um trabalho futuro, seria interessante correlacionar os outros eventos para tentarmos entender como se comporta a comunidade de usuários do *Github*

Esse relatório pode ser visualizado no próprio ***Github*** e no ***rPubs***