Lab 2, Checkpoint 1 - Mais variáveis, várias formas

```
library(tidyverse)
library(here)
library(knitr)
theme_set(theme_bw())
```

Dataset

Os dados utilizados na análise foram extraídos por meio Travis Torrent. Neles estão presentes informações sobre builds, commits, casos de testes e etc de diferentes projetos desenvolvidos sob as linguagens de programação Java ou Ruby hospedados no GitHub.

Com base dos dados do Travis Torrent, foi construído um novo dataset com a sumarização de valores e a criação de novas variáveis, abaixo são apresentadas as variáveis resultantes desse novo dataset.

```
projetos = read_csv(here::here("data/projetos.csv"))
```

```
## Parsed with column specification:
## cols(
##   gh_project_name = col_character(),
##   team = col_double(),
##   lang = col_character(),
##   sloc_end = col_integer(),
##   sloc_med = col_double(),
##   activity_period = col_integer(),
##   num_commits = col_integer(),
##   commits_per_month = col_double(),
##   tests_per_kloc = col_double(),
##   total_builds = col_integer(),
##   build_success_prop = col_double(),
##   builds_per_month = col_double(),
##   tests_added_per_build = col_double(),
##   tests_successful = col_double(),
##   test_density = col_double(),
##   test_size_avg = col_double()
## )
```

Nesse contexto, por meio dos dados, foi investigada a existência de uma ***relação (positiva ou negativa) entre número de commits mensais nos projetos e a sua densidade de testes e tempo de build com status sucesso, e ainda se a linguagem de programação utilizada podia influenciar nesse aspecto***. Para isso, as seguintes variáveis foram utilizadas:

- ***lang:*** principal linguagem de programação utilizada no projeto, variando entre Java, Ruby e Javascript;
- ***build__success__prop:*** sumarizaçao do número total de builds com status sucesso divido pelo número total de builds em cada projeto. Esta variável apresenta valores entre 0 e 1, onde quanto mais perto de 1, mais tempo o projeto ficou com build status sucesso;
- ***commits__per__month:*** número de commits divido pelo número de meses de atividade do projeto, identificado pelo intervalo de tempo em meses do primeiro até último commit realizado no projeto, variando entre as faixas de valores de 1 a 9716;
- ***test__density:*** mediana da densidade de testes do projeto. Esta variável apresenta valores nas faixas entre 0 a 2366.

Para esse estudo foram descartados alguns projetos da linguagem de programação Javascript existentes nos dados, devido a análise concentra-se na integração contínua de testes em projetos Java e Ruby.

```
projetos = projetos %>%
    filter(lang != "javascript")
```

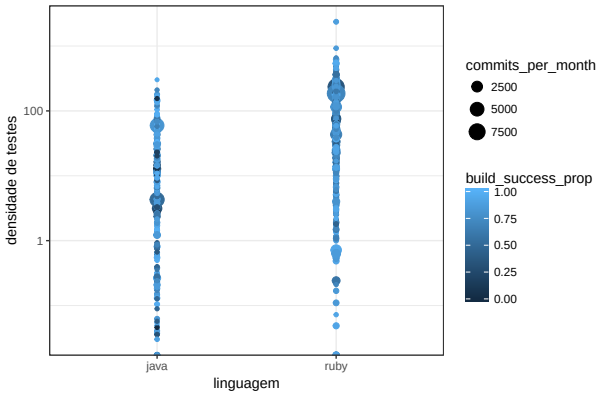
Investigando a relação entre número de commits mensais e densidade de testes e tempo de build com status sucesso.

Com o objetivo de investigar a relação entre o número de commits mensais com a densidade de testes e tempo médio de build com status sucessos dos projetos, a Figura 1 foi desenvolvida.

Nesse contexto, a fim de diferenciar o número de commits mensal, foi adotado o aspecto de dimensão do ponto, variando entre 3 faixa de valores entre os projetos com maior número de commits mensal (7500) a projetos com menor número de commits mensal (2500). Além disso, foi também aplicado na visualização tonalidades de azul para identificar a proporção de builds com status de sucesso, diversificando em 5 faixas de valores diferentes entre tonalidades mais escuras (0%) a tonalidades mais claras (100%).

```
    projetos %>%
    ggplot(aes(x =lang , y = test_density,)) +
    geom_point(aes(color = build_success_prop, size = comm
    scale_y_log10()+
    labs(y= "densidade de testes", x="linguagem", title="D
```

Figura 1 – Número de commits e densidade de testes/prop. de builds suce



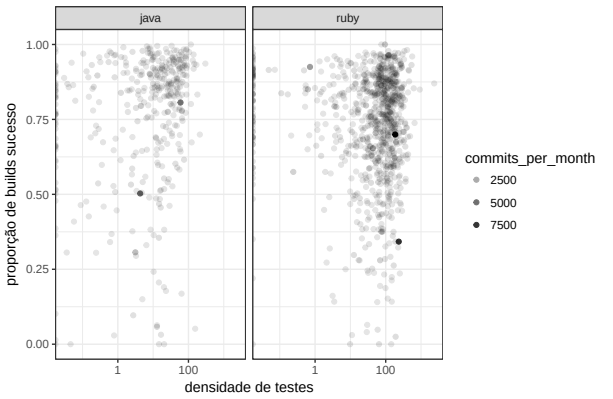
De acordo com a Figura 1, é possível observar que a maioria dos projetos apresenta a proporção de builds com status sucesso entre 0.75 (75%) e 1 (100%), esta observação não sofre alterações ao considerar o número de commits mensal do projeto ou ainda sua linguagem de programação. No mais, não é perceptível por meio desta visualização identificar uma relação entre a proporção de build com status sucesso e a densidade de testes do projetos, pois existem projetos com proporção de builds sucesso entre 0.75 a 1 que possuem sua densidade de teste entre 0 e 1.

Considerando a investigação proposta nesse estudo, é notável que alguns projetos possuem grande densidade de teste, ficando entre 5000 e 7500 commits mensais. Além, disso, levando em conta somente projetos com a linguagem Ruby, existem projetos com commits mensais acima de 100 e com densidade de teste de 7500. No entanto, ainda assim, é identificado na visualização uma grande variação (dispersão) neste aspecto, onde é apresentado projetos com densidade de testes entre 0 e 1 e comnpumero de commits entre 1 a 2500. Este número de commits pode também ser encontrado em projetos com grandes densidade de testes (acima de 100). Assim, somente por meio dessa visualização não é possível apontar uma relação entre as características observadas.

Diante do exposto, foi optado pela criação da Figura 2 para tentar confirmar ou refutar a existência dessa relação. Nela é mostrada em outra pespectiva a relação entre proporção de builds sucesso e a densidade de testes e também número de commits mensais dos projetos com auxilio do uso de brilho nos pontos, considerando tonalidades mais escuras para os projetos com maior número de commits mensais (7500).

```
projetos %>%
ggplot(aes(x = test_density, y = build_su
geom_point() +
facet_grid(. ~ lang) +
scale_x_log10()+
labs(y= "proporção de builds sucesso", x="densidade de
```

Figura 2 – Número de commits e densidade de testes/prop. de builds suc

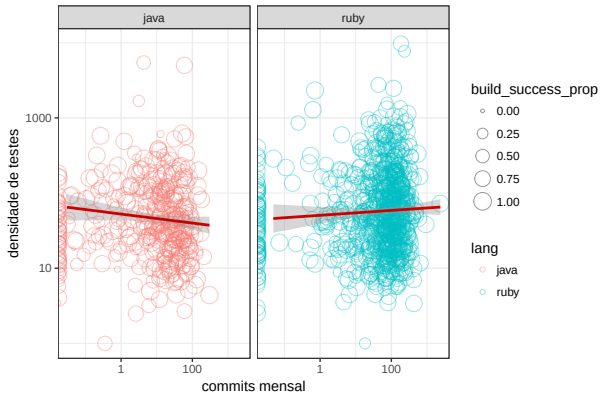


Com base na Figura 2, é possível identificar um número maior de projetos com alta densidade de testes (estando entre as faixas de 50 e 100) e maior proporção de builds sucesso entre os valores de 0.75 (75%) e 1 (100%) e ainda maiores quantidades de commits mensais, ficando mais evidente nesse aspecto projetos de linguagem Ruby. No entanto, no que diz respeito a proporção de builds com status sucesso e número de commits mensais, é observável também existência de projetos com proporção entre 0.75 e 1, mas com número de commits mensais entre 0 e 1, o que invalida por meio dessa visualização qualquer possibilidade de estabelecimento de uma relação entre elas. Já considerando a relação entre número de commits mensal e densidade de testes, é notável uma pequena tendência, onde alguns projetos com maior número de commits possuem maior densidade de testes em ambas linguagens. Contudo, esta relação por meio da visualização mostra-se fraca, não sendo possível confirmá-la.

A fim de identificar de forma mais clara a relação entre densidade de testes e número de commits mensal e também confirmar a não existência da relação de número de commits mensal e proporção de builds com status sucesso, a Figura 3 foi produzida.

```
projetos %>%
ggplot(aes(x = test_density, y = commits_per_month, co
geom_point(shape = 1, alpha = 0.4 ,aes( size = build_s
geom_smooth(method=lm, se=TRUE, color="red3") +
facet_grid(. ~ lang) +
scale_x_log10()+
scale_y_log10()+
labs(y= "densidade de testes", x="commits mensal", tit
```

Figura 3 – Número de commits e densidade de testes/prop. de builds suce

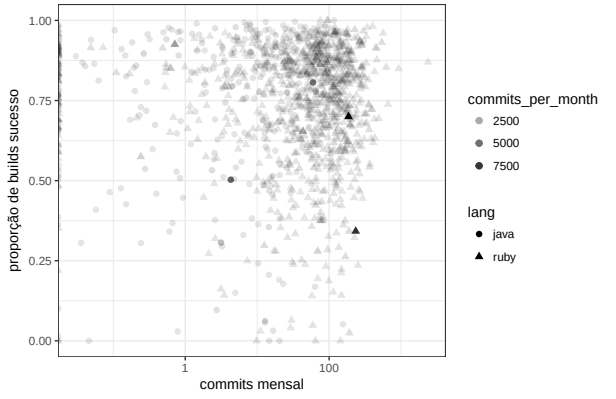


De acordo com a Figura 3, parece existir uma tendência entre número de commits mensal e densidade de testes no que diz respeito projetos Java, onde quanto maior o número de projetos menor a densidade de testes. Já considerando projetos Ruby, esta relação parece ser positiva, onde projetos com maior número de commits mensais possui maior densidades de testes. No entanto, ambas tendências são fracas (existência de uma grande dispersão). Assim, tais tendências não foram consideradas válidas para este análise.

Ainda nesse contexto, com o objetivo de observar as características investigadas em diferentes perspectivas, a Figura 5 foi produzida.

```
projetos %>%
ggplot(aes(x = test_density, y = build_succe
geom_point(size = 2, aes( shape = lang, alph
scale_x_log10()+
labs(y= "proporção de builds sucesso", x="cc
```

Figura 4 – Número de commits e densidade de testes/prop. de builds suce

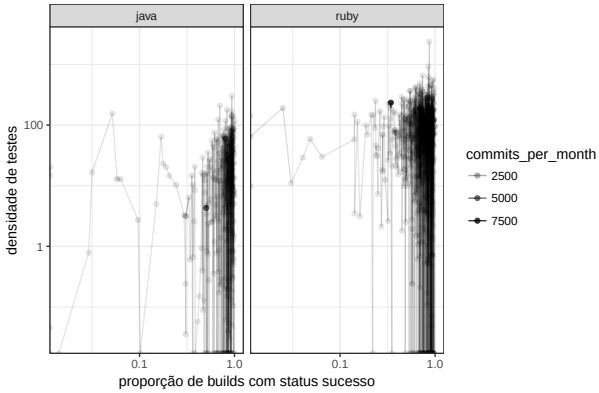


A figura 4, apresenta uma visualização que utiliza-se do brilho e formas geométricas para representar o número de commits mensal e a linguagem de programação utilizada nos projetos. Por meio dela, é perceptível uma maior concentração dos dados entre proporção de builds com status sucesso entre 0.75 e 1 e commits mensais entre 50 a 100. Neste contexto, é observado poucos possuem alto número de commits mensais (7500) e alta proporção de builds sucesso. Além disso, existem projetos com número de commits mensais entre 7500 e proporção de build sucesso abaixo de 0.50 (50%), o que invalida ainda mais a existência de uma relação entre proporção de builds sucesso e o número de commits mensais. Já considerando a relação entre densidade de testes e número de commits mensais, assim como as demais Figuras, é possível notar uma fraca relação, mas que não válida considerando os projetos e suas dispersões.

Diante das Figuras produzidas, surgiu a necessidade investigadas as relações propostas nessa pesquisa por meio de outros tipo de visualizações. Assim, a Figura 5 e 6 são apresentadas.

```
projetos %>%
ggplot(aes(x = build_success_prop  , y = test_density
geom_line()+
facet_grid(. ~ lang) +
geom_point() +
scale_x_log10()+
scale_y_log10()+
labs(y= "densidade de testes", x="proporção de builds
```

Figura 5 – Número de commits e densidade de testes/prop. de builds suce



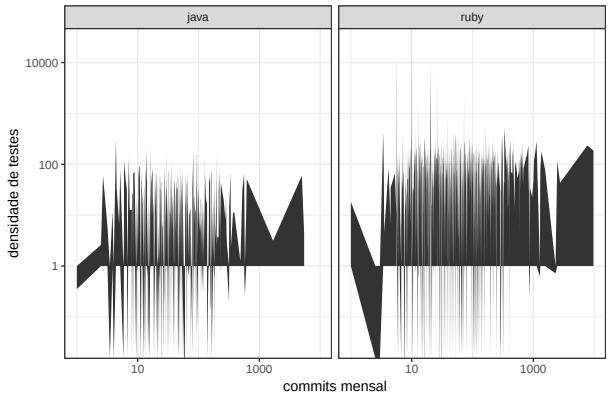
A Figura 5 apresenta a relação entre proporção de builds com status sucesso e densidade de testes. Para identificar os projetos com maior número de commits mensal, foi utilizado pontos com linhas com diferentes tons de preto, onde quanto mais escuro o ponto e linha for mais commits mensal o projeto possui.

Com base na figura, é possível observar que projetos maior proporção de build sucesso possuem mais commits mensais. No entanto, isto não invalidado em todas as demais visualizações, o que pode significar que esta visualização não é adequada para verificação dessa relação. Em outro contexto, também não possível de forma clara identificar se projetos com maior número de commits mensal possuem maior densidade de teste, devido a uma sobreposição entre pontos e

linhas na visualização. Assim, nenhuma das relações foram encontradas ou ainda validadas.

```
projetos %>%
ggplot(aes(x = commits_per_month, y = test_density, build_success_prop)) +
geom_area()+
facet_grid(. ~ lang) +
scale_x_log10()+
scale_y_log10()+
labs(y= "densidade de testes", x="commits mensal", title="Figura 6 - Número de commits e densidade
```

Figura 6 − Número de commits e densidade de testes/prop. de builds suc



Por meio da Figura 6, é possível verificar que os projetos de ambas linguagens possuem variações na densidade de testes entre 0 a 1000. Além disso, projetos Ruby parecem ter maior densidade de testes quando relacionado esse aspecto com número de commits mensais. Já projetos Java possuem maior constância nesse aspecto.

Avaliação das visualizações

Com base na análise realizada durante este trabalho, não foi possível identificar uma relação válida entre número de commits mensal e densidade de testes e/ou proporção de builds sucesso. Além disso, as linguagens de programação utilizadas nos projetos parece não interferir nesses resultados.

No que diz respeito a eficácia das visualizações apresentas durante esta análise, a que apresentou uma visualização mais clara para a identificação da existência de relações entre número de commits mensal e densidade de testes e/ou proporção de builds com status sucesso foi a Figura 1, onde por meio de tons de cores e uso de diferentes dimensões para os pontos é possível verificar os projetos com maiores e menores proporções de builds sucesso e os projetos com maior e menor número de commits. Desta forma, é possível apontar os projetos com maior densidade de testes, maior proporção de builds sucesso, maior número de commits mensal e ainda se existe diferenças nos dados para as diferentes linguagens estudadas. Assim, a investigação pode ser respondida de forma simples e clara. Em contrapartida, as visualizações apresentadas nas Figuras 5 e 6, parecem serem as menos indicadas para verificação da relação proposta.