UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
ESCOLA DE INFORMÁTICA APLICADA

Github - Análise dos projetos mais famosos

Cecilia Soares
Diogo Ferreira da Silva
Felipe Arruda Pontes
João Marcello Calil

Trabalho de Estatística - 2015.1

Prof.Bruno Simões

Introdução

Neste trabalho vamos analisar o comportamento e utilização do GitHub[1] que é uma rede social colaborativa para projetos que usam o controle de versionamento Git (DABBISH, et al, 2012). Nele usuários de todo mundo e empresas podem compartilhar seus projetos, códigos, oferecer e receber suporte da comunidade, além procurar soluções para seus problemas em outros projetos, tudo para contribuir para o melhor desenvolvimento de softwares e repositórios.

Para este trabalho foram analisados os dados dos 1843 projetos melhor ranqueados do GitHub, com diversos dados relevantes como por exemplo: Linguagem utilizada; Numero de Estrelas, Idade do projeto e outros.

Para coletar estes dados, no dia 28 de Abril, foi acessado a API pública do Github[2], utilizando scripts em Python. Devido as limitações de acesso impostas pelo Github, não foi possível coletar mais dados, pois para tal seria necessário mais tempo para recuperar os dados sem infringir os limites definidos pelo Github, conforme explicado melhor em sua documentação (GITHUB, 2015).

Objetivo

O objetivo deste trabalho é analisar os dados dos mais famosos projetos do Github e identificar como estes dados se relacionam, de forma que possamos estabelecer possíveis padrões na utilização da ferramenta, bem como entender melhor o comportamento dos seus usuários.

Como dito anteriormente o critério de escolha dos projetos é a quantidade de estrelas que o mesmo possui, em outras palavras, os mais bem votados.

Contextualização das Variáveis

Neste tópico será descrito e contextualizado de forma mais detalhada o que representa cada varíavel coletada.

Nome do projeto (name)

O nome do projeto, exemplos: Linux, Rails, Django.

Tipo de dono (owner_type)

Indica qual o tipo de usuário é responsável pelo projeto em questão, podendo este ser uma organização (uma empresa, ou grupo) ou usuário comum.

[1] Endereço do site Github: https://github.com/

[2] Endereço da API pública do Github: https://api.github.com/

Data de criação (created_at)

Representa a data em que um determinado projeto foi criado no Github. Essa váriavel foi subdividida em 3 outras, uma representando o dia, mês e ano (Created at Day/Month/Year).

Data de atualização (updated_at)

Essa variável define a ultima dada em que foi feito alguma alteração no, levando em consideração que os dados foram coletados em 28 de Abril de 2015. Assim como a data de criação, esta variável foi dividida em dia, mês e ano;

Tamanho em KBs (size)

Define o espaço em disco ocupado pelo projeto, em KBs.

Estrelas (stargazers_count)

Representa o número de estrelas que o projeto possui. Qualquer usuário no Github pode dar uma estrela ou não para qualquer projeto. Marcar um projeto com uma estrela é uma forma do usuário indicar que um projeto em questão merece um destaque. Assim, projetos com maior numero de estrelas são os projetos que tem maior destaque no Github.

Linguagem utilizada (language)

Define a linguagem principal[3] usada no projeto. Essa linguagem não é necessariamente uma linguagem de programação, podendo ser uma linguagem de marcação (como o HTML, ou XML), linguagem de folha de estilos (ex: CSS) e etc. Quando a linguagem é None, isso indica que nenhuma linguagem foi utilizada no projeto, sendo o repositório composto apenas por textos e/ou imagens e outros artefatos.
Como o conceito de escrita colaborativa de livros tem se tornado mais popular, conforme podemos observar em diversos sites como GitBook, Penflip e outros, o Github também se tornou palco para manter estes projetos de livros. Com isso

Se tem ou não wiki (has_wiki)

Booleano que indica se o projeto utiliza a Wiki do próprio Github como forma de documentar o projeto em questão.

Numero de commits por dia da semana (Num. Cmts.)

Um *commmit* é uma alteração feita por um usuário em um projeto, sendo que está pode representar uma alteração em uma linha em determinado arquivo texto, e/ou alterações em imagens ou outros artefatos do projeto.
Existe uma variável indicando o numero total de commits em um projeto para cada dia da semana, desde a criação do projeto, até o momento (Num. Cmts. Seg, Num. Cmts. Ter, e etc).

[3] Linguagem Principal: a linguagem que tem maior porcentagem de uso dentre todas as utilizadas no mesmo projeto, segundo a análise do Github.

Numero de commits total (Total Commits)

Indica o total de commits realizados no projeto, desde sua criação até o momento (essa variável é o somatório da variáveis Num. Cmts. para cada dia da semana).

Idade do projeto (Age)

Quantos anos possui o projeto, considerando o dia, mes e ano em que ele foi criado, até a data que foi coletado os dados. Vale ressaltar que o calculo no calculo da idade foi utilizado a biblioteca Python *dateutil* considerado o valor correto para o mesmo, considerando o numero de dias correto de cada mês, considerando os bisexto (LEEUW, 2015).

Análise de Resultados

Nesta seção serão analisados os dados e apresentados os resultados que foram obtidos.

Análise Descritiva

Stars

Conforme pode ser visto na Figura 1, existe uma grande quantidade de projetos com um numero próximo a 4 mil Stars e apenas um projeto (Bootstrap[4]) com 80mil Stars, e o projeto com menor numero de Stars com 302. Por conta disto demos para essa variável um valor **médio de 4013** Stars, e com um CV(Coeficiente de Variação) de **1.05**.

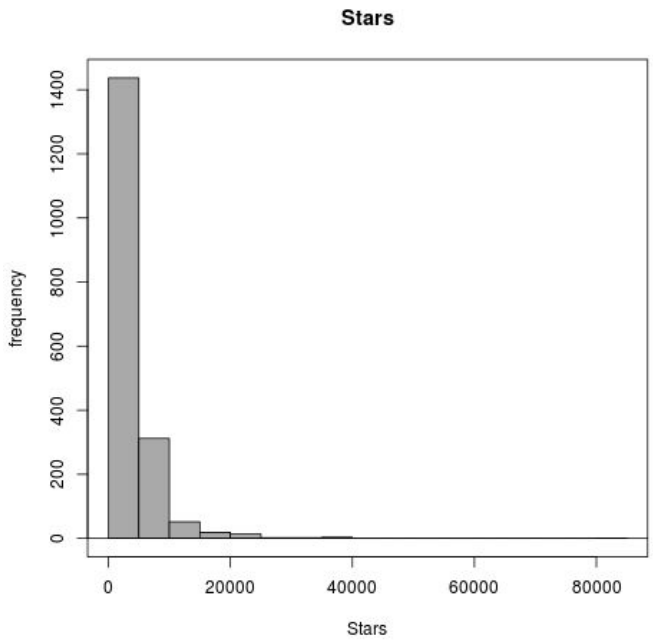


***Figura 1**: Frequência de Stars*

Total de Commits

Temos também nessa variável um grande grande frequência dos projetos reunidos no primeiro quartil, conforme ilustrado na **Figura 2**. Temos também valores interessantes para

[4] Link para repositório do projeto Bootstrap: https://github.com/twbs/bootstrap

o total de commits, visto que o **minimo** para o mesmo é de 1[5] , e seu máximo de 20 mil, e uma média de 1472 (com CV de 2.23).
Tal caso em que temos apenas 1 commit, é dado pelo fato de que o projeto foi realocado para outro repositorio, de mesmo nome, mas sob responsabilidade de outra organização (usuário).

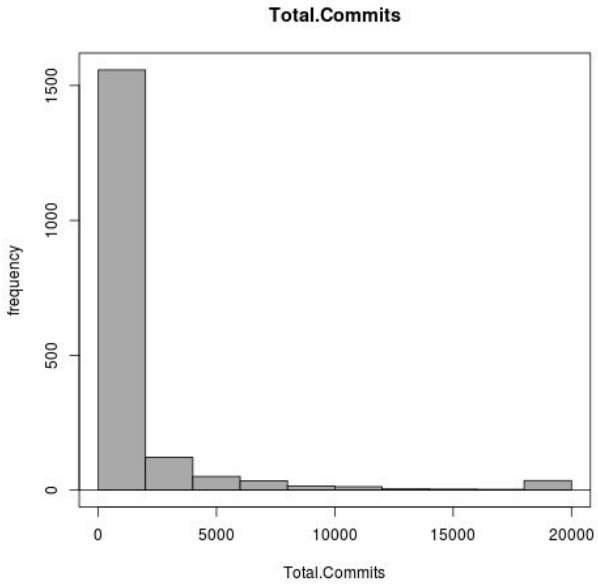


***Figura 2**: Frequencia do Total de Commits*

Forks

Nesta variável temos um valor médio de 813, com CV de 2.29, um valor minimo de 4 e máximo 56170, conforme mostra a **Figura 3**.

[5] Projeto libuv com apenas 1 commit: https://github.com/joyent/libuv

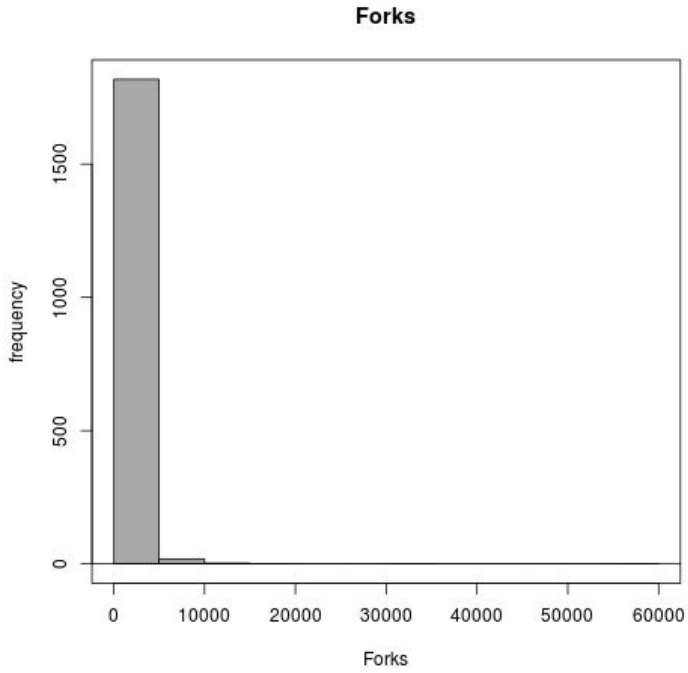


***Figura 3**: Frequencia do Numero de Forks*

Created at Year

Essa variável foi muito interessante de se analisar, pois deixa mostra que existem muitos projetos criados a muito tempo. Ela também mostra a “explosão” que houve no uso do Github entre 2011 e 2013 (que detém 57,9% de todos os projetos), e que agora parece estar diminuindo, conforme mostra a **Figura 4**.

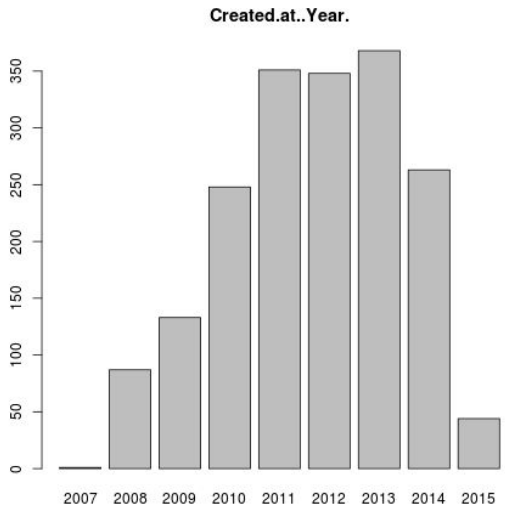


***Figura 4:** Frequencia de Create at Year*

Last Update Year

Esta variável demonstra como projetos colaborativos são capazes de mesmo após muitos anos, continuarem ativos, principalmente se formos levar em consideração que existem **221**

projetos criados entre **2007 e 2009**, e que apenas **5** projetos foram atualizados pela ultima vez em **2014**. Vide **Figura 5**.

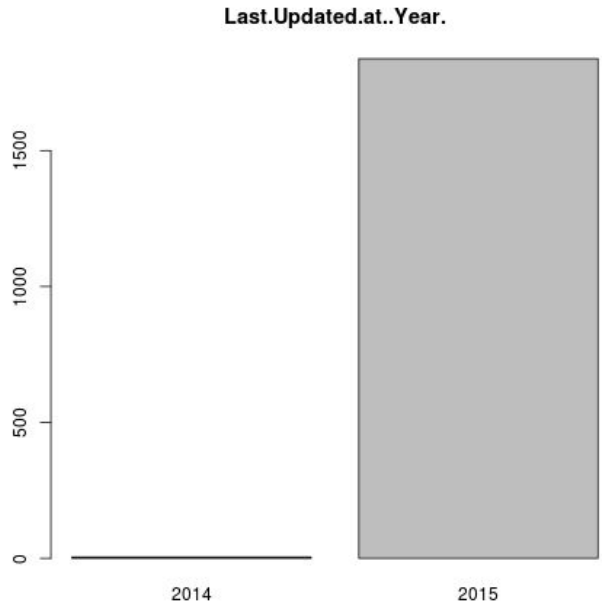


Figura 5: *Frequencia de Last Updated at Year*

Languages

Para facilitar a visualização desta variável (**Figura 6**), separamos apenas as 10 linguagens mais utilizadas nestes 1843 projetos estudados, e o restante juntamos num conjunto denominado “Outros”. Conforme podemos observar, a linguagem predominante é o **JavaScript** (com 34%), isso ocorre por que muitos dos projetos feitos nesta linguagem acabam sendo usados na interface de sistemas web de diferentes linguagens. Assim, sua importância é exaltada. O mesmo ocorre com o **CSS** e **HTML**.
Já a linguagem definida como **None** (que não possui nenhuma linguagem na verdade), tem aparecido em grande quantidade, dentre esses 1843 projetos mais bem ranqueados, pois muitos destes projetos são grandes repositórios de links, textos, artigos e livros sobre diversos assuntos. Esta prática de utilizar o Github para livros colaborativos e afins tem crescido bastante, conforme será mostrado mais na frente.

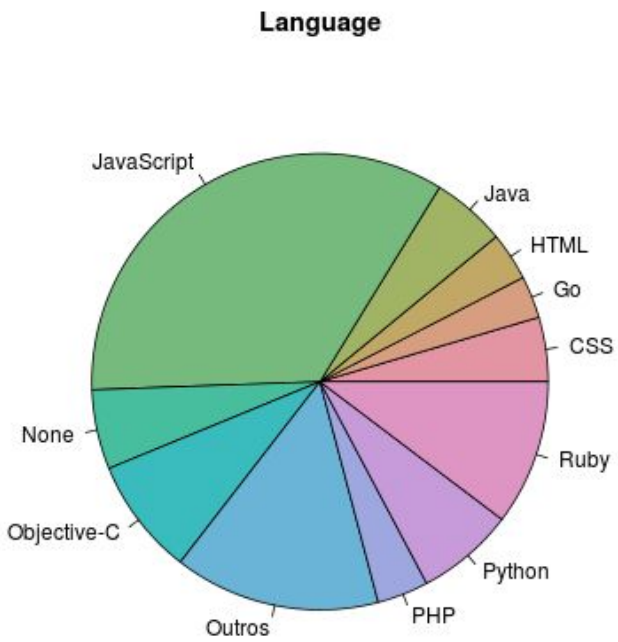


Figura 6: *Gráfico das Linguagens dos projetos*

Has Wiki

Como podemos observar, muitos dos projetos estudados (78% ao todo) utilizam a funcionalidade de Wiki presente no Github. Porém vimos também que isto não tem uma relação direta com o número de estrelas, provavelmente pelo fato de que mesmo projetos que não usam Wiki do Github costumam utilizar outros meios de documentação externos.

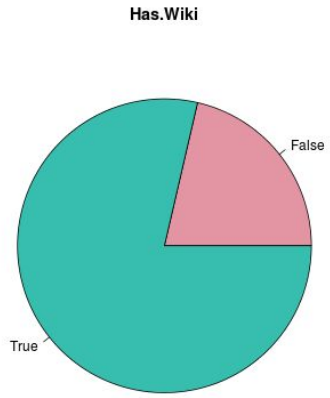


Figura 7*: Gráfico do uso de Wikis*

Age

Esta variável foi calculada a partir do ano de criação de cada projeto e a data no qual os dados foram coletados (28 de Abril de 2015).

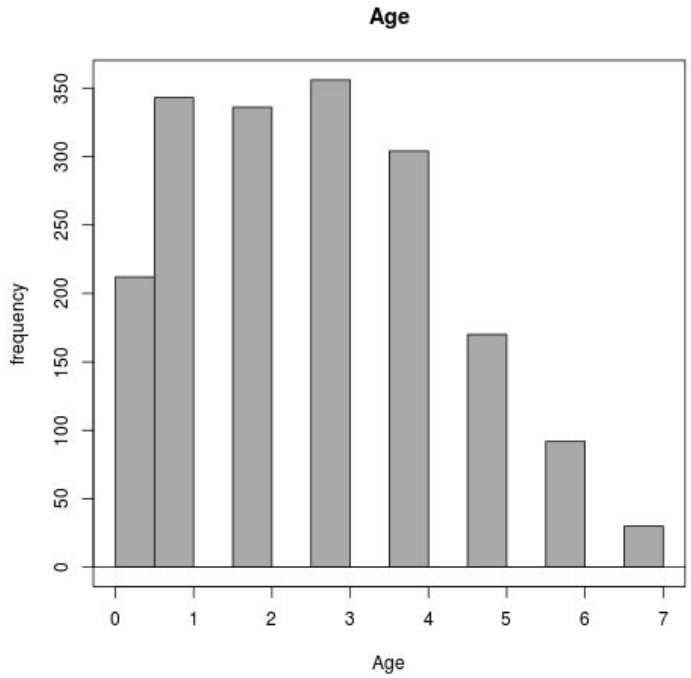


***Figura 8**: Frequência da variável Age*

Análise de Correlação

Neste tópico iremos detalhar algumas das correlações importantes encontradas durante a análise.

Variáveis Quantitativas

Primeiramente mostrar quais correlações existem e quais não existem entre as variáveis que pudemos identificar, como por exemplo:

O fato de ter correlação entre o numero total de commits e o tipo de usuário responsável pelo projeto. Isso pôde ser visto quando separamos os dados em 2 grupos (um para usuários e outro para organização) e verificamos a distribuição do Total de Commits em cada um dos grupos. Tal fato é melhor visto na tabela a seguir e na **Figura 9**.

Tipo de Usuário	Média (Total de Commits)	CV (Total de Commits)
Organização	2570	1.74
Usuário	555	2.08

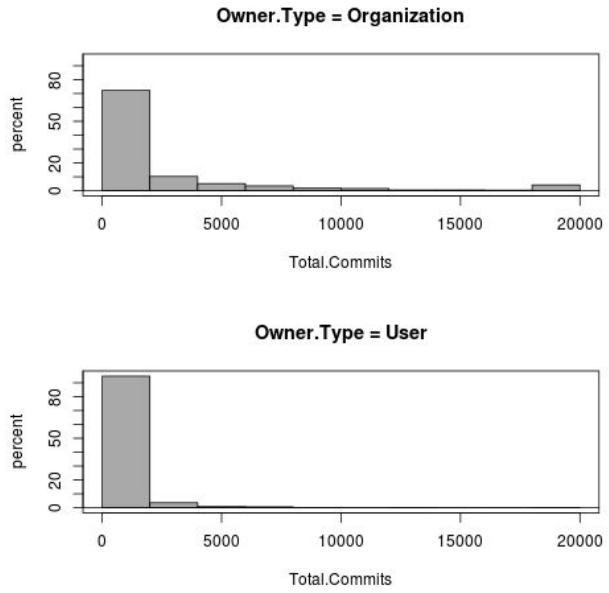


Figura 9: *Total de Commits por Tipo de Usuario*

Outro ponto analisado, foi a correlação do Total de Commits com o ano de criação do projeto. Ficou claro que apenas a idade de um projeto não determina que ele terá mais alterações (commits), mesmo que ele tenha tido mais tempo, conforme pode ser visto na **Figura 10**.

Uma possível explicação para esse acontecimento, seria que um projeto mais antigo, como os de 2007 e 2008 não tem mais tanta alteração (mesmo que tenham sido alterados recentemente), em seguida temos um crescimento em 2009-2010, e por fim, voltamos ao comportamento esperado, que seria de que quanto mais recente o projeto, menos commits ele teria (já que não teria tido tempo de terem sido feitas tantas alterações).

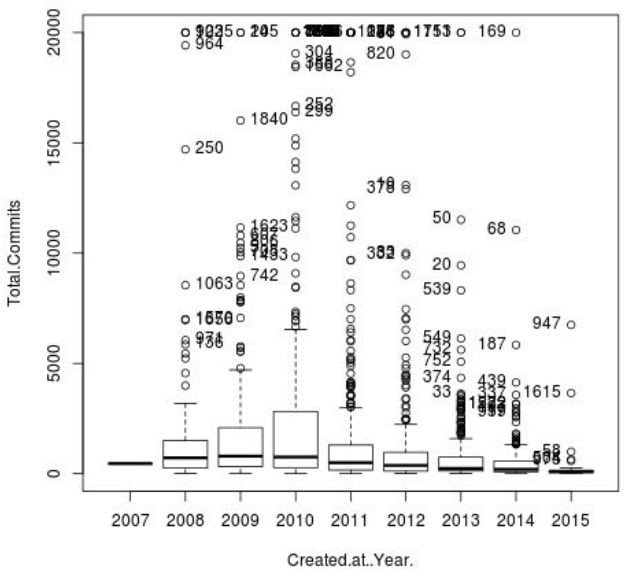


Figura 10: *Total de Commits por Created at Year.*

Em seguida, analisamos como os commits ao longo dos dias da semana se relacionam, e pudemos perceber que os commits feitos no Sábado e no Domingo seguem um mesmo padrão, com uma correlação de **0.981**, conforme pode ser visto na **Figura 11**.

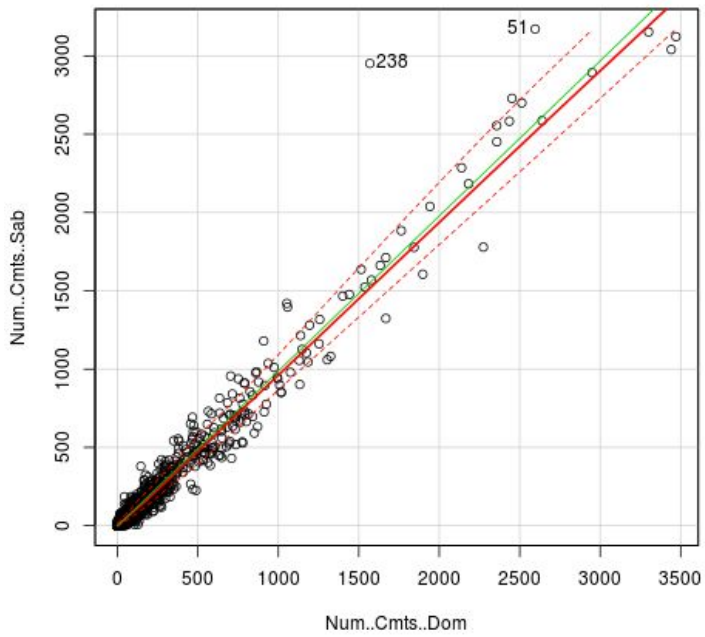


Figura 11: *Correlação entre commits no Domingo e Sabado*

Já os outros dias da semana, os dias úteis, também possuem relação entre si. Entretanto ao comparar qualquer dia útil com um dia de final de semana, tal correlação se torna bem fraca, conforme podemos ver pela tabela a seguir e pela **Figura 12**. Por conta desta correlação, decidimos tentar verificar se havia de fato 2 fatores que poderiam representar estas variáveis. Assim, mais à frente falaremos sobre o fator de Dia de Semana, e o de Final de Semana.

	Num..Cmts..Dom	Num..Cmts..Qua	Num..Cmts..Qui	Num..Cmts..Sab	Num..Cmts..Seg	Num..Cmts..Sex	Num..Cmts..Ter
Num..Cmts..Dom	1.0000000	0.7609127	0.7774553	**0.9814794**	0.7955426	0.7939936	0.7700988
Num..Cmts..Qua	0.7609127	1.0000000	**0.9909539**	0.7904662	**0.9837513**	**0.9841419**	**0.9800930**
Num..Cmts..Qui	0.7774553	0.9909539	1.0000000	0.8002759	**0.9918765**	**0.9937085**	**0.9908228**
Num..Cmts..Sab	0.9814794	0.7904662	0.8002759	1.0000000	0.8105627	0.8155841	0.7828633

Num..Cmts.. Seg	0.7955426	0.9837513	0.9918765	0.8105627	1.0000000	**0.9908058**	0.9917477
Num..Cmts.. Sex	0.7939936	0.9841419	0.9937085	0.8155841	**0.9908058**	1.0000000	**0.9904784**
Num..Cmts.. Ter	0.7700988	0.9800930	0.9908228	0.7828633	**0.9917477**	0.9904784	1.0000000

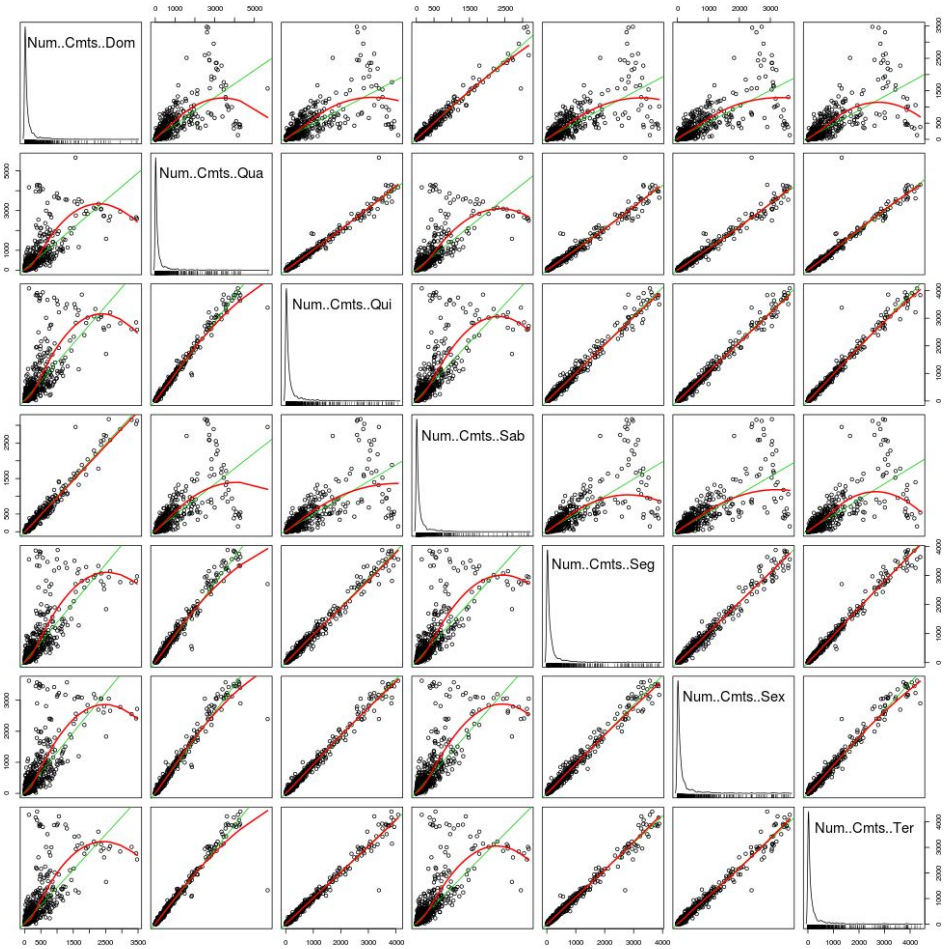


Figura 12: *Matriz de Correlação entre todos os dias da semana*

Variáveis Qualitativas

Foi possível identificar também algumas relações entre variáveis qualitativas presentes no estudo. Neste caso mostraremos a relação entre a variável Language e Created at Year,

conforme pode ser visto na **Figura 13**.

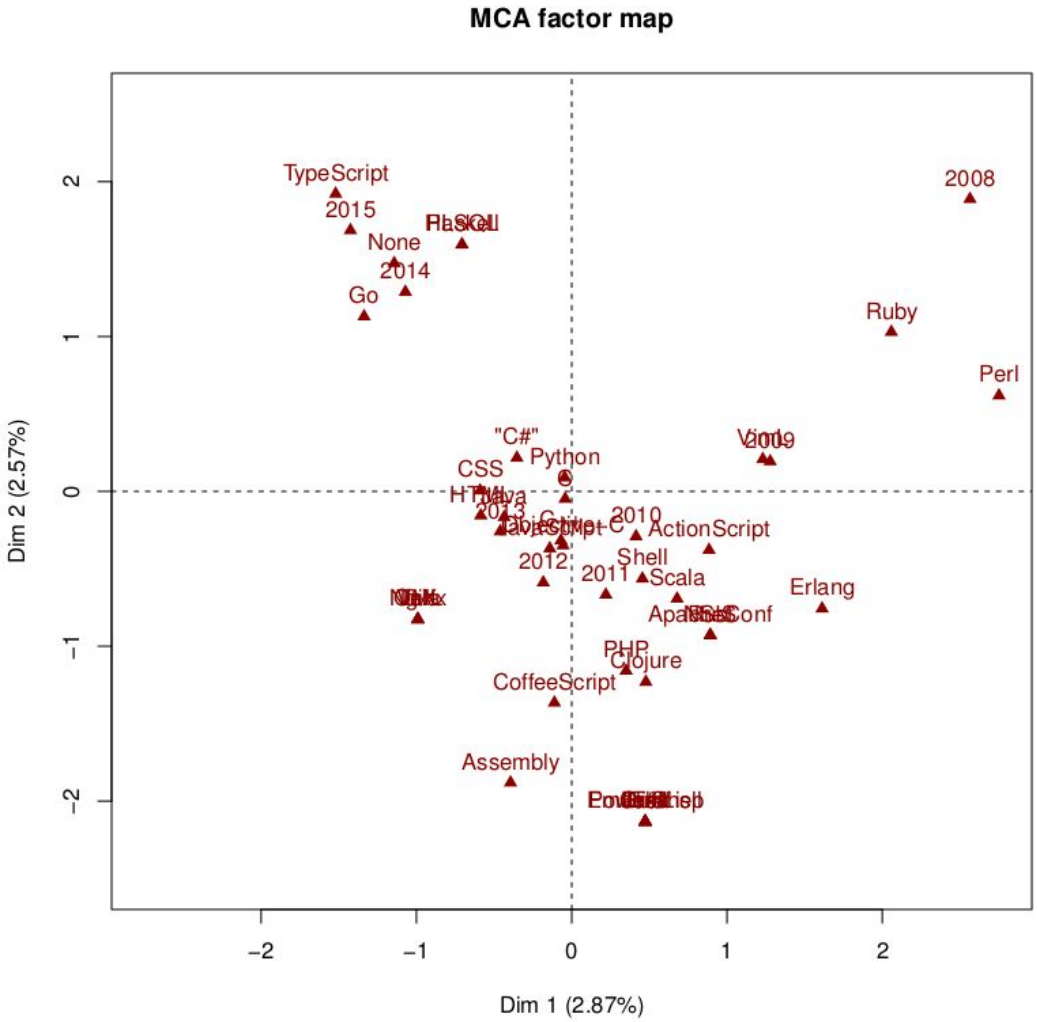


Figura 13: *Correlação entre Language e Created at Year*

Neste caso é possível observar que linguagens como o Python, Java, C e C# possuem um peso maior nos anos de 2010-2013, que é justamente o mesmo período em que houve uma maior quantidade de projetos sendo criados (vide **Figura 4**). Enquanto a linguagem Ruby é maior entre os anos de 2009 e 2008, que bate justamente com os anos iniciais do site Github, onde este era mais utilizado pela comunidade Ruby.
Outro ponto importante de ressaltar é a correlação entre das linguagem None e Go e os anos de 2014 e 2015. No que diz respeito ao GO isto se deve por conta linguagem Go ser relativamente nova, tendo sua primeira versão lançada apenas em 2012 (GOLANG, 2015), já com relação a caso de None é provavelmente por conta da popularização do conceito de escrita colaborativa de livros, conforme podemos observar em diversos sites como GitBook[6] , Penflip[7] e outros, que tornou o Github palco para manter estes projetos de livros.

[6] Site do GitBook: https://www.gitbook.com/
[7] Site do Penflip: https://www.penflip.com/

Testes de Normalidade

Durante a analise foi observado que nenhuma das variáveis possuem um comportamento de reta Normal então esta analise não entrará das deliberações deste trabalho. Pode-se ter uma noção deste fato observando-se os seguintes resultados :

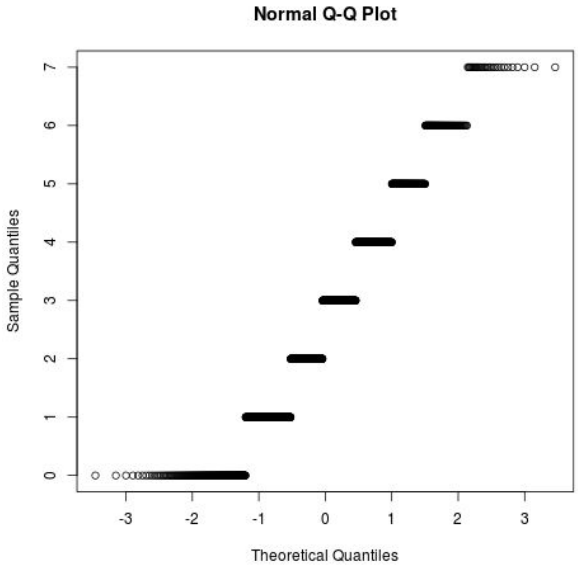


Figura 14: *QQNorm da Variável Age*

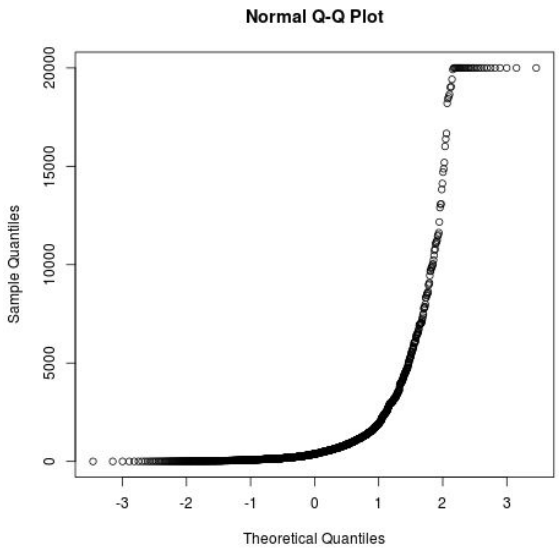


Figura 15: *QQNorm da Variável Total de Commits*

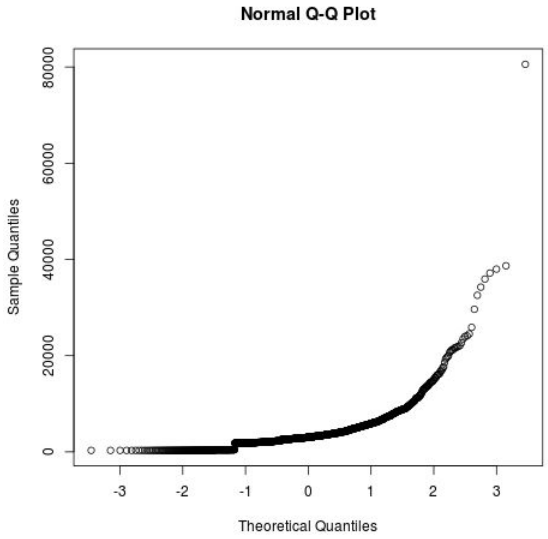


***Figura 16**: QQNorm da Variável Stars*

E com base nos **p-valores** baixos considerando-se o teste de normalidade de Shapiro-Wilk podemos dizer que nenhuma das variáveis segue uma distribuição normal (SHAPIRO & WILK, 1965):

	statistic	p.value
Size	0.1109658	1.334735e-68
Stars	0.5924987	2.152607e-54
Watchers	0.5924987	2.152607e-54
Forks	0.284992	1.833763e-64
Open.Issues	0.4090312	5.945727e-61
Num..Cmts..Dom	0.4380682	4.849562e-60
Num..Cmts..Seg	0.4384998	5.006677e-60
Num..Cmts..Ter	0.4352295	3.93423e-60
Num..Cmts..Qua	0.4263426	2.055576e-60
Num..Cmts..Qui	0.43328	3.409515e-60
Num..Cmts..Sex	0.4381348	4.873476e-60
Num..Cmts..Sab	0.4222538	1.52919e-60
Total.Commits	0.4451797	8.222452e-60

Age	0.9465493	2.458655e-25

Análise Fatorial

Neste tópico serão descritas algumas analises fatoriais aplicadas ao longo do projeto, dentre as que testamos, as que tiveram melhor resultado foram os Fatores Dia de Semana e Final de Semana, conseguidos através das variares de numero de commmits ao longo da semana.
Conforme a **Figura 17** nos mostra, ficou bem claro a separação entre os dias de semana e os dias úteis.

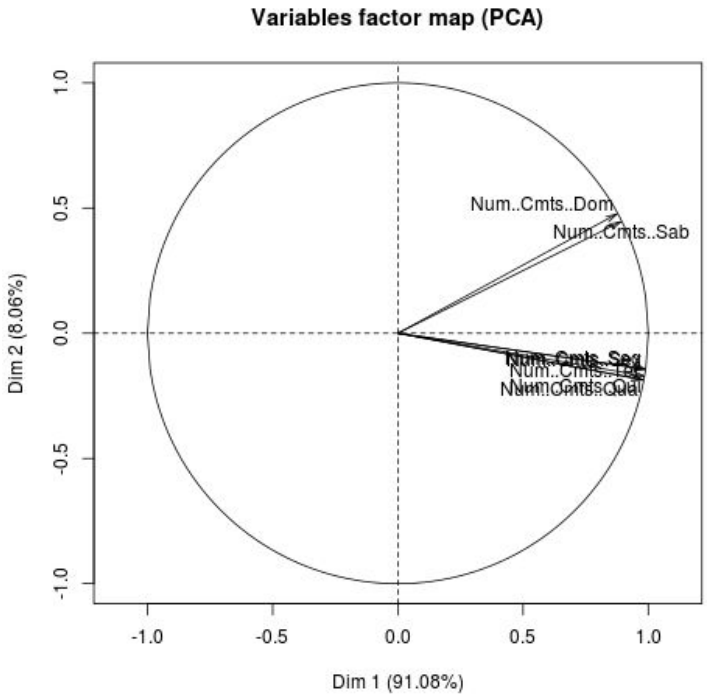


Figura 17: *Mapa dos fatores para as variaveis de Numero de Commits ao longo da semana.*

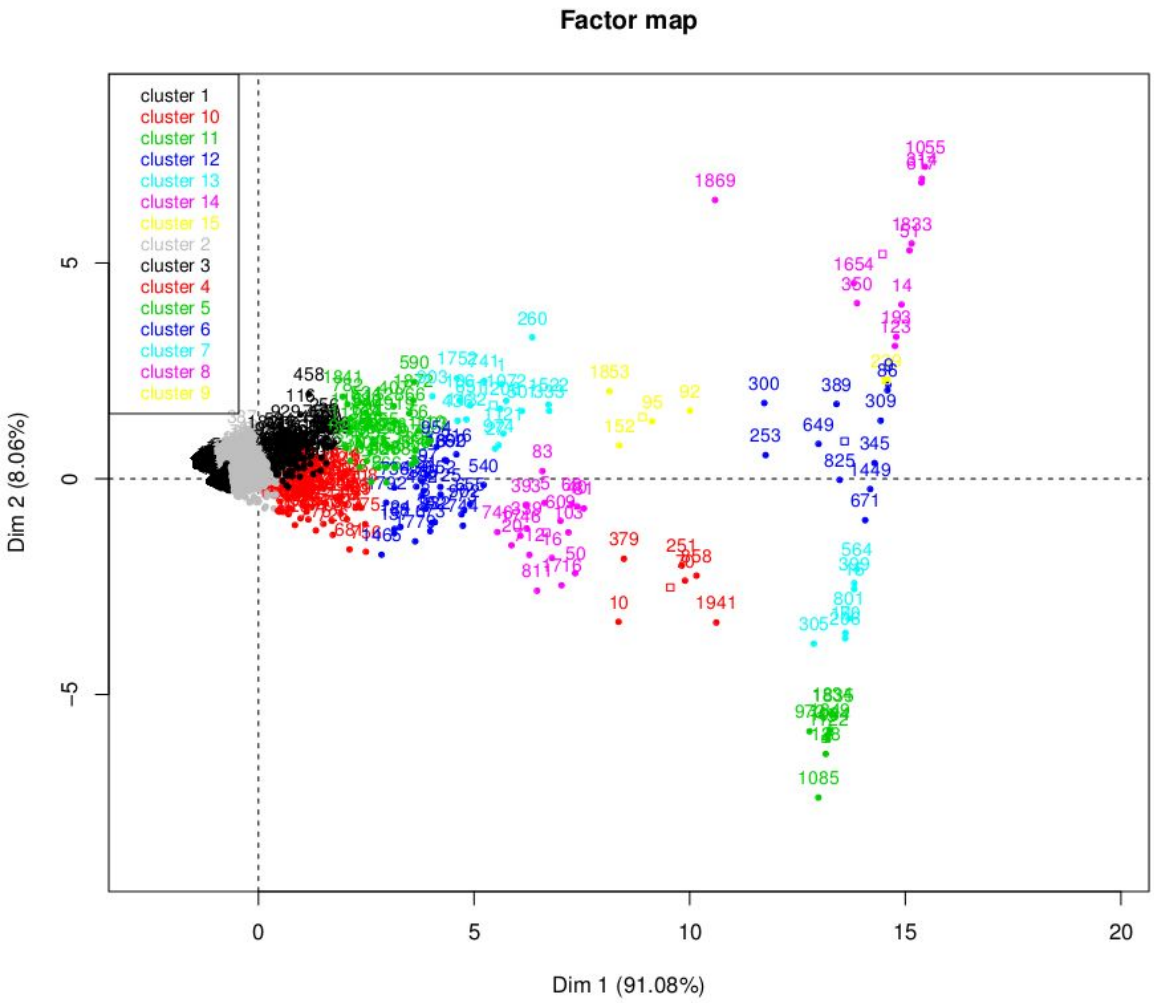
O uso destes dois fatores se mostra muito promissor, visto que juntos representam 98,7% de todas essas 7 variáveis, conforme podemos ver na tabela abaixo (onde o Fator 1 representa Dias Úteis, e o Fator 2 os Finais de Semana):

	Factor1 (Dias Úteis)	Factor2 (Finais de Semana)
Num..Cmts..Dom	0.462	0.869
Num..Cmts..Qua	0.903	0.405
Num..Cmts..Qui	0.907	0.415
Num..Cmts..Sab	0.482	0.873
Num..Cmts..Seg	0.895	0.435
Num..Cmts..Sex	0.894	0.440
Num..Cmts..Ter	0.913	0.394

SS loadings	4.518	2.393
Proportion Var	0.645	0.342
Cumulative Var	0.645	**0.987**

Análise Cluster

Usando os fatores identificados anteriomente para representar os Dias Úteis e os Finais de Semana, foi feito uma analise cluster hierarquica dos mesmos, onde dividimos em 15 clusters.



Selecionando os seguintes clusters:

- Cluster 1: Poucos Commits;
- Cluster 14: muitos Commits no Final de Semana;
- Cluster 11: muitos commits em dias de semana;

- Cluster 12 e 13: muitos commits tanto nos finais de semana quanto nos dias de semana.

Temos essa tabela com os seguintes comparativos:

Cluster	**Média de Forks**	**Média de Stars**	**Média de Total.Commits**
Cluster 1	574	3223	227
Cluster 14	3327	6949	19166
Cluster 11	1297	4478	19935
Cluster 12 e 13	3051	7387	19325

Isso nos faz ter uma noção que projetos que tem seus commits distribuídos mais igualmente ao longo da semana(Cluster 12 e 13), tem uma maior quantidade de Stars do que aqueles que fazer mais nos finais de semana ou em diais úteis.
E podemos ver também que projetos que tem uma maior quantidade de commits no final de semana tem, em média, mais Forks e mais Stars que aqueles que focam mais em dias de semana.

Conclusões

Com base nos dados coletados e analisados, foi possível determinar diversos correlações entre os mesmos, bem como entender melhor o comportamento dos usuários do site Github e ter uma ideia melhor de quais fatores levam um projeto a se tornar um projeto de renome. Desta forma, foi possível ver que o tipo de usuário não tem uma grande influência na quantidade de Stars de um projeto.
Da mesma forma, foi interessante identificar que projetos nas linguagens Java Script, CSS e None tinham uma quantidade de estrelas maior que a maioria. Tal analise em conjunto da demonstração de que projetos sem nenhuma linguagem especifica (sejam eles livros, ou acervos colaborativos de informação) tem tido um grande crescimento desde 2014, serve para ratificar o fato de que cada vez mais pessoas tem utilizado o Github para projetos que não sejam unicamente de programação.
Futuramente seria interessante analisar mais repositórios além dos 1843 coletados, talvez ampliando a analise para um numero maior de repositórios seja possível encontrar uma normal entre as variáveis. Além disso, uma nova analise após esta, em um diferente período temporal, pode nos proporcionar novos comportamentos entre as variáveis. Por exemplo hoje temos algumas linguagens em maior foco, daqui a três anos essas linguagens podem não estar mais em foco assim como podem haver novas entrando em destaque. Seria possível ver também se o site teve alguma evolução, no sentido de criar novas variáveis a serem analisadas.

Referências Bibliográficas

Dabbish, L., Stuart, C., Tsay, J., & Herbsleb, J. (2012, February). Social coding in GitHub: transparency and collaboration in an open software repository. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work* (pp. 1277-1286). ACM.

GoLang (2015). “Frequently Asked Questions (FAQ) - The Go Programming Language.” https://golang.org/doc/faq. Acessado em: 25-06-2014.

Github (2015). “GitHub API v3”. https://developer.github.com/v3/. Acessado em: 25-06-2014.

Leeuw, Y. (2015). “dateutil - powerful extensions to datetime - dateutil 2.4.2 documentation.” https://dateutil.readthedocs.org/en/latest/index.html. Acessado em: 25-06-2014.

Shapiro, S. S., & Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 591-611.