

Relatório

Autores

Arthur Rocha Amaral

Guilherme Oliveira Antônio

Introdução

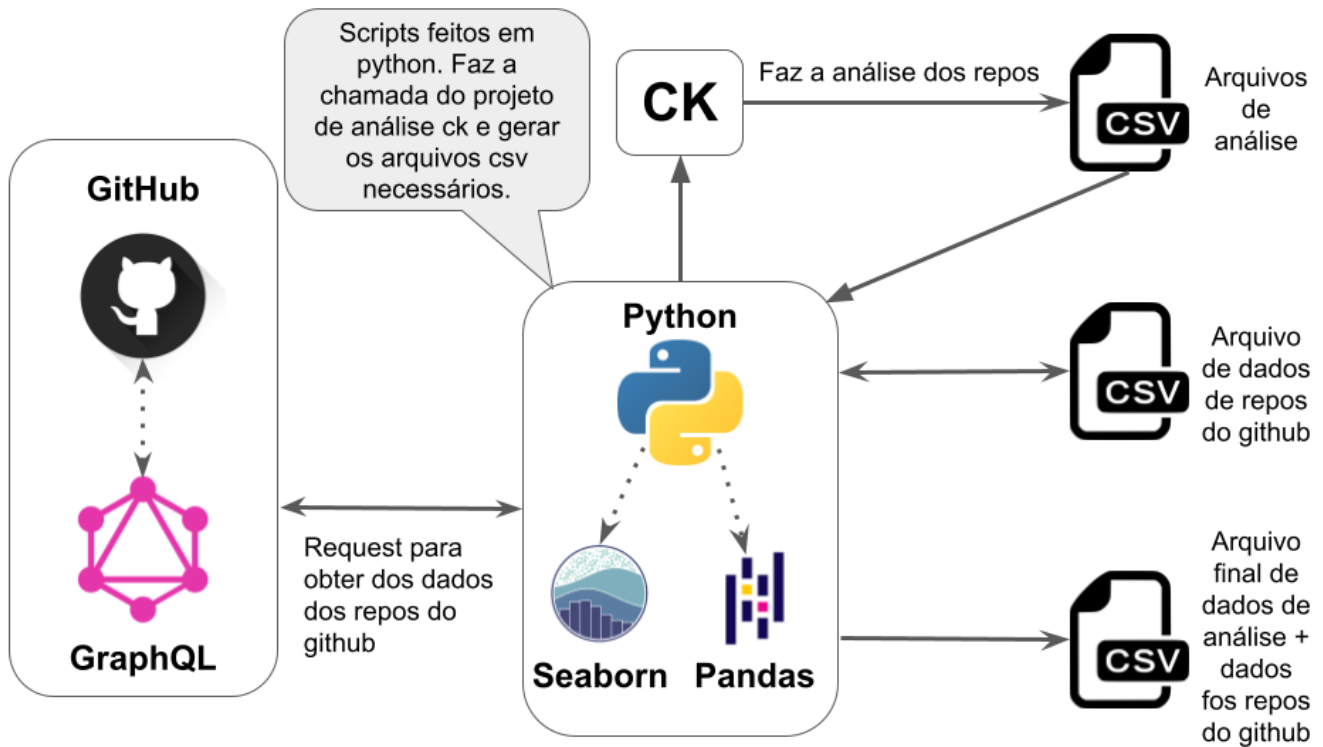
No processo de desenvolvimento de sistemas open-source, em que diversos desenvolvedores contribuem em partes diferentes do código, um dos riscos a serem gerenciados diz respeito à evolução dos seus atributos de qualidade interna. Isto é, ao se adotar uma abordagem colaborativa, corre-se o risco de tornar vulnerável aspectos como modularidade, manutenibilidade, ou legalidade do software produzido. Para tanto, diversas abordagens modernas buscam aperfeiçoar tal processo, através da adoção de práticas relacionadas à revisão de código ou à análise estática através de ferramentas de CI/CD.

Neste contexto, o objetivo deste laboratório é analisar aspectos da qualidade de repositórios desenvolvidos na linguagem Java, correlacionando-os com características do seu processo de desenvolvimento, sob a perspectiva de métricas de produto calculadas através da ferramenta [CK](#). Sendo que as métricas que serão usadas para esse trabalho serão o lcom, cbo e dit.

Metodologia

Para obtermos os dados a serem avaliados, buscamos da API GraphQL do GitHub os repositórios ordenados em ordem decrescente pela quantidade de estrelas da linguagem **java**, listados em páginas de 100 itens, assim, pegaremos as 10 primeiras páginas com os dados necessários para respondermos as questões de pesquisa. Após o recebimento dos dados, será filtrado os dados inconsistentes e salvo o resultado em um arquivo CSV. Dessa forma, será feito um clone de cada repositório java listado e salvo. Com isso, será rodado o projeto de análise de CK, o qual gera os arquivos de análise de métricas. Após esse passo, será construído um arquivo csv com uma mescla dos resultados analisados necessários para o trabalho e os dados necessários de cada repositório.

Por fim, após os dados necessários salvos, torna-se viável a construção dos gráficos construídos com as bibliotecas pandas e seaborn que ilustram os valores obtidos com os quais serão feitas as análises.



Métricas usadas

- **lcom**: É calculada considerando pares de métodos em uma classe. LCOM é a diferença entre o número de pares de método sem atributos compartilhados e o número de pares de método com atributos compartilhados.
- **cbo**: Conta as classes chamadas por uma classe. As classes são acopladas quando os métodos em uma classe usam métodos ou variáveis de instância definidos em uma classe diferente. CBO é uma medida do grau de acoplamento. Um valor alto de CBO significa que as classes são altamente dependentes.
- **dit**: Representam o número de níveis que uma classe herda métodos e atributos. Quanto mais profunda a árvore de herança, mais complexo é o projeto. Muitas classes podem ter de ser compreendidas para que seja possível compreender as classes nas folhas da árvore.

Coeficiente de correlação usado

- **Coeficiente de correlação de Spearman**: O coeficiente avalia com que intensidade a relação entre duas variáveis pode ser descrita pelo uso de uma função monótona.

Discussão/Hipóteses/Valores obtidos

RQ Q1 - Qual a relação entre a popularidade dos repositórios e as suas características de qualidade?

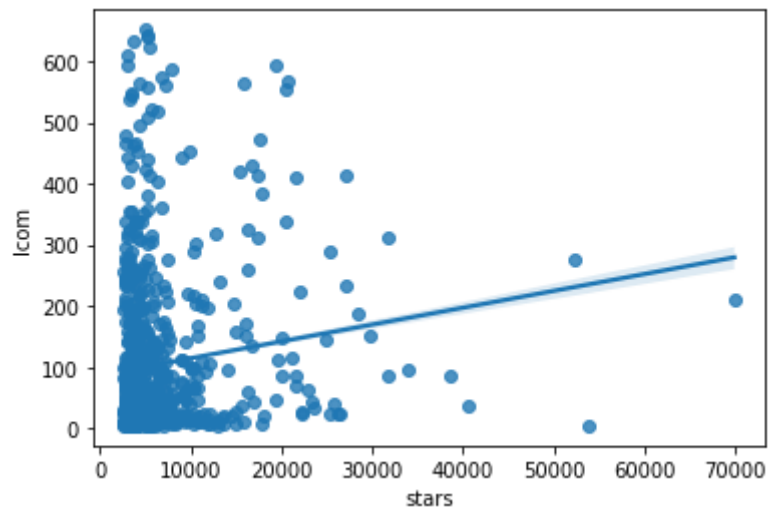
Estrelas x Lcom

- **Hipótese**:

Quanto mais popular, melhor é o LCOM dos repositórios, devido a melhor qualidade do código e melhor manutenibilidade e reusabilidade para mais facilitar a distribuição dos esforços entre os

colaboradores dos projetos.

- **Resultados:**



Coeficiente de correlação de Spearman para estrelas e lcom $r = 0.1580977350487473$

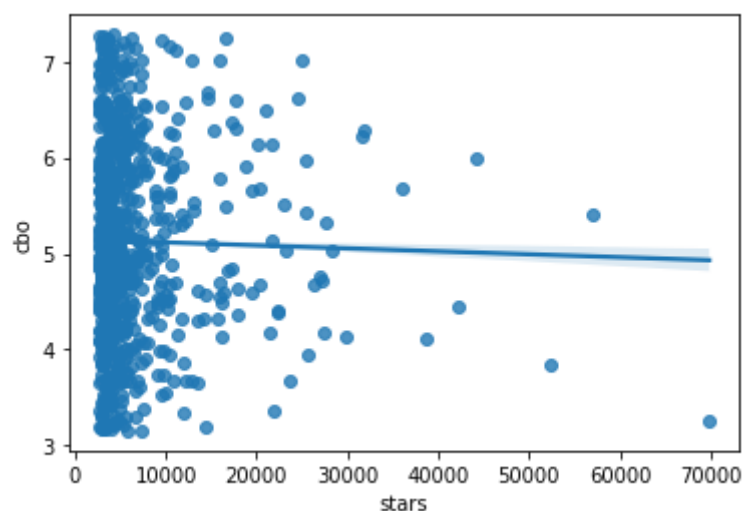
Podemos ver, pelo gráfico, uma tendência de aumento do LCOM em relação ao aumento da quantidade de estrelas, mas observando o coeficiente de correlação, vemos que essa correlação é frágil, pois temos muitos repositórios com quantidades iguais de estrelas e não existe uma certa definição de quantidade de LCOM para estes projetos.

Estrelas x CBO

- **Hipótese:**

Devido à popularidade do repositórios atrair mais colaboradores, podemos criar a seguinte hipótese: Para que mais pessoas possam atuar no mesmo código, de forma mais eficiente, esperamos que quanto mais popular, menos acoplado seja o código deste.

- **Resultados:**



Coeficiente de correlação de Spearman para estrelas e cbo $r = 0.01114985741684774$

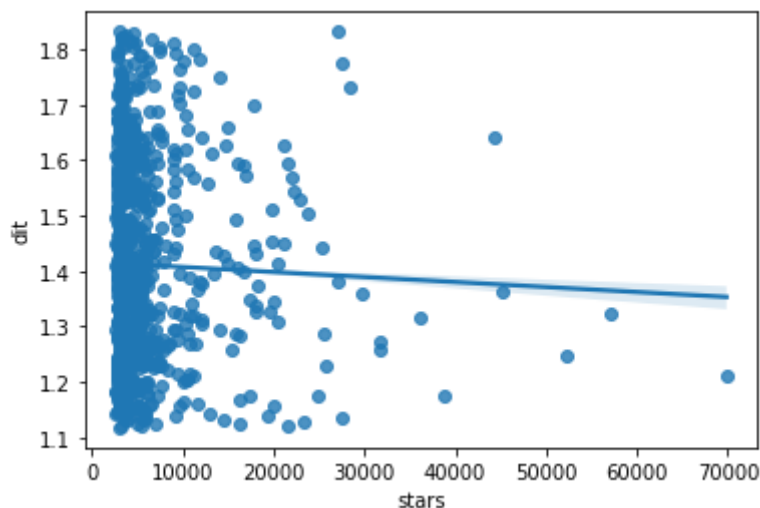
Como resultado, obtemos uma relação insignificante entre esses dois aspectos, mesmo tendo os 5 primeiros repositórios com mais estrelas, com CBO abaixo da média.

Estrelas x DIT

- **Hipótese:**

Quanto mais popular e colaborativo, o repositório deve ser menos complexo para que mais pessoas possam contribuir com mais facilidade.

- **Resultados:**



Coeficiente de correlação de Spearman para estrelas e dit $r = -0.010310920299837729$

Apesar do gráfico exibir uma reta decrescente, assim como na análise anterior com o COB, não podemos afirmar que esse gráfico ilustra uma correlação direta entre os eixos, mesmo mostrando os 5 repositórios mais populares abaixo da média de DTI. O coeficiente de correlação também aponta a desconexão dos dados apresentados.

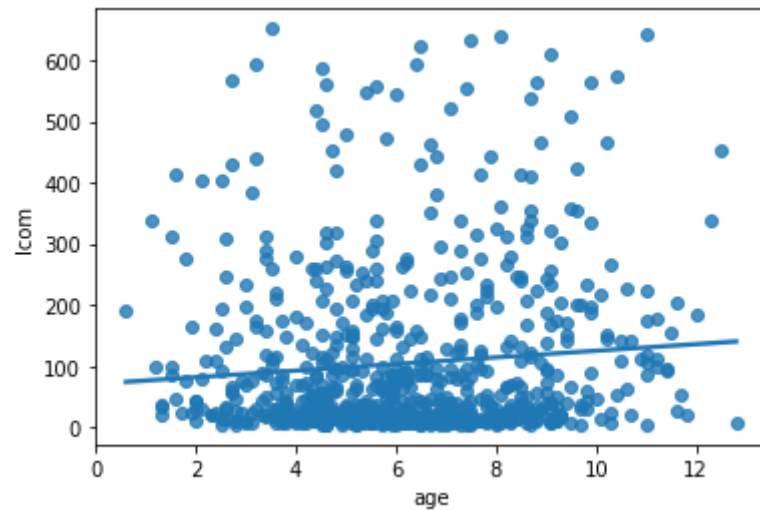
RQ Q2 - Qual a relação entre a maturidade dos repositórios e as suas características de qualidade ?

Idade x Lcom

- **Hipótese:**

Quanto mais o tempo passa mais coeso menos coeso os repositórios, pois a partir do momento que o projeto vai crescendo fica difícil a sua refatoração.

- **Resultados:**



Coeficiente de correlação de Spearman para estrelas e lcom $r = 0.05665888721051483$

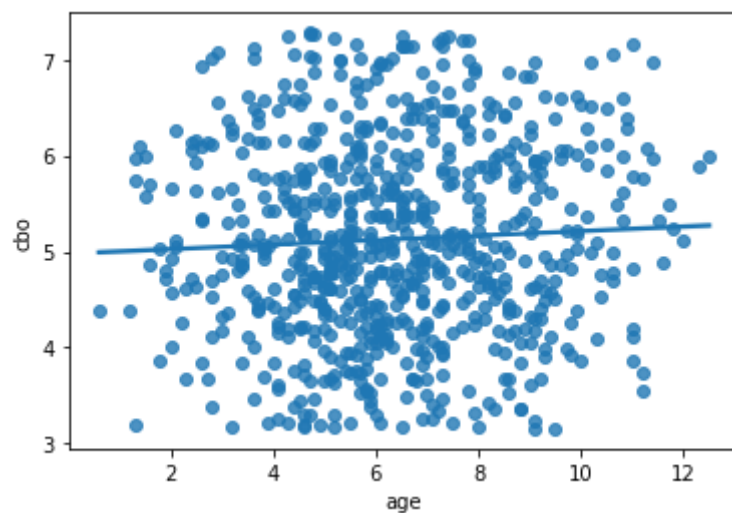
O resultado obtido mostra que há uma reta crescente, onde pode demonstrar que quanto mais o tempo passa mais coesos são os métodos dos projetos. Porém, com uma análise mais detalhada pode-se perceber que a maioria dos repositórios tem o lcom próximo de 0. Com isso, pode-se perceber uma desconexão entre os dados, podendo ser provado através do coeficiente de correlação calculado, o qual foi de 0.05.

Idade x CBO

- **Hipótese:**

Quanto maior a maturidade do projeto menos acoplado são as classes.

- **Resultados:**



Coeficiente de correlação de Spearman para estrelas e cbo $r = 0.032239205124179965$

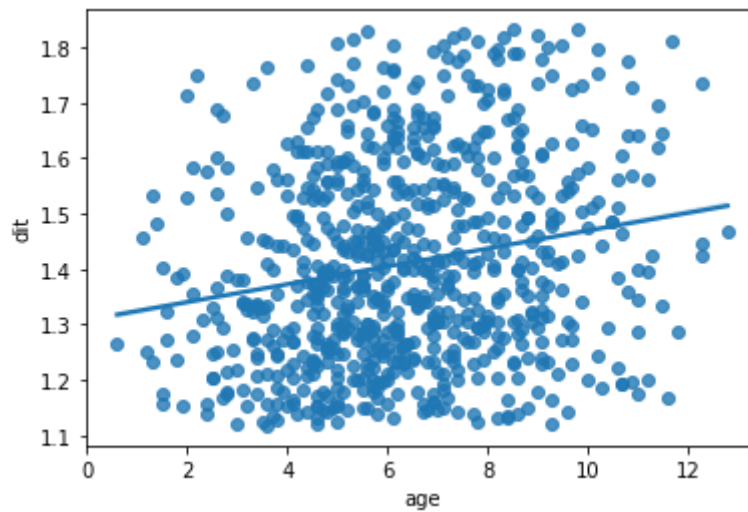
A hipótese não pôde ser comprovada através dos dados analisados. Ao observar o gráfico gerado, pode-se dizer que não existe uma relação entre a maturidade e a métrica observada. Isso pode ser confirmado ao se ter o coeficiente de correlação próximo de 0, onde o mesmo é de 0.032.

Idade x DIT

- **Hipótese:**

Quanto maior a maturidade do projeto maior o número de classes herdadas.

- **Resultados:**



Coeficiente de correlação de Spearman para age and dti $r = 0.19844914547241582$

É observado que a relação entre idade e dti é fraca, visto que seu coeficiente de correlação é de 0.19. Também é observado no gráfico que os dados estão bem dispersos, o que mostra que não é possível tirar uma conclusão dos dados analisados.

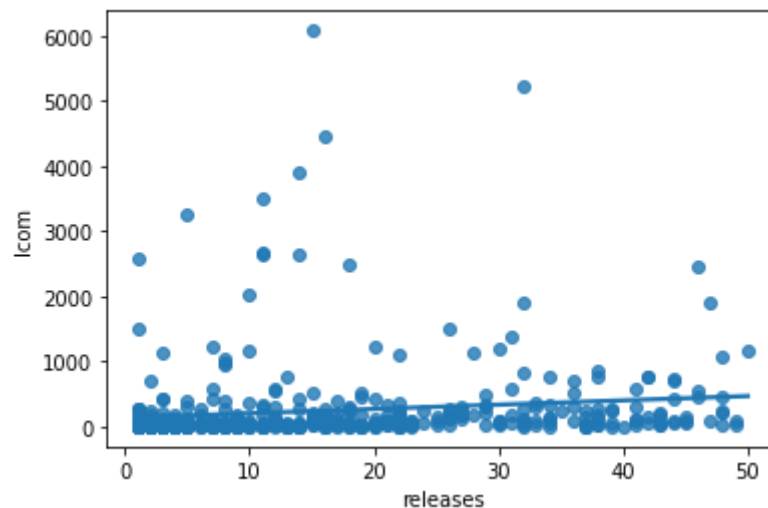
RQ Q3 - Qual a relação entre a atividade dos repositórios e as suas características de qualidade?

- **Hipótese:**

Acreditando que a métrica de release seja um valor que mostra a taxa de atualização de código, podemos esperar que todos os índices de qualidade de código reflitam essa melhoria, já que quanto mais o projeto é atualizado mais o código é melhorado.

- **Resultados:**

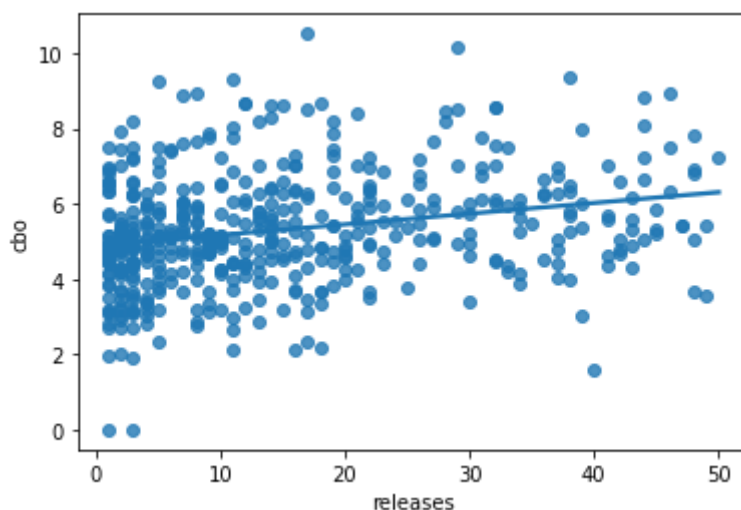
Releases x Lcom



Coeficiente de correlação de Spearman para releases and lcom $r = 0.3580470634144376$

Vemos que apesar de existir uma reta que aponta a tendência, temos um coeficiente de correlação baixo para dizer que a quantidade de releases interfere na métrica de LCOM.

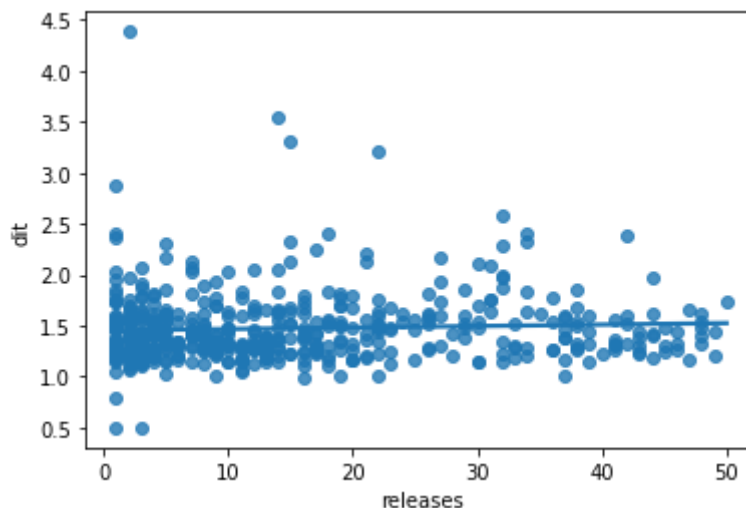
Releases x CBO



Coeficiente de correlação de Spearman para releases and cbo $r = 0.2657982980485922$

Apesar de existir uma leve tendência de aumento do CBO, não podemos afirmar que esteja correlacionado com a quantidade de releases de um repositório, devido ao coeficiente de correlação de Spearman.

Releases x DIT



Coeficiente de correlação de Spearman para releases and dit $r = 0.08469757952561427$

Com o resultado obtido não é possível correlacionar as métricas de quantidade de release e DIT.

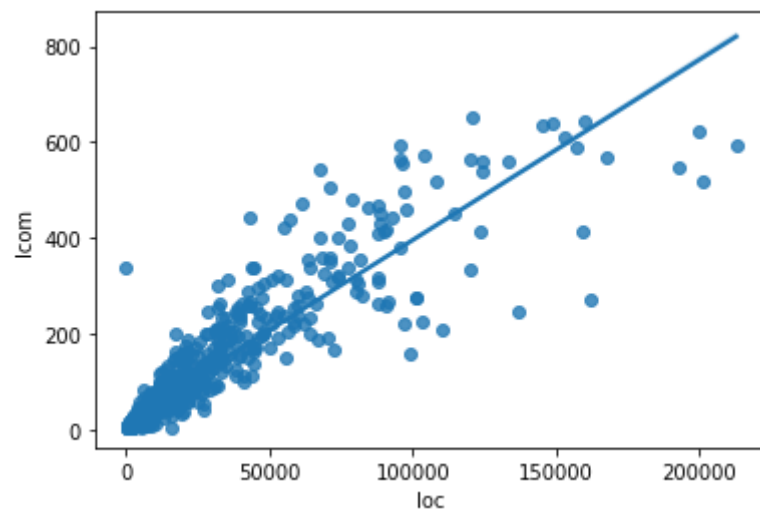
RQ Q4 - Qual a relação entre o tamanho dos repositórios e as suas características de qualidade?

LOC x Lcom

- **Hipótese:**

Quanto menos o número de linhas de código melhor a coesão dos metodos.

- **Resultados:**



Coeficiente de correlação de Spearman para loc and lcom $r = 0.9463379680074533$

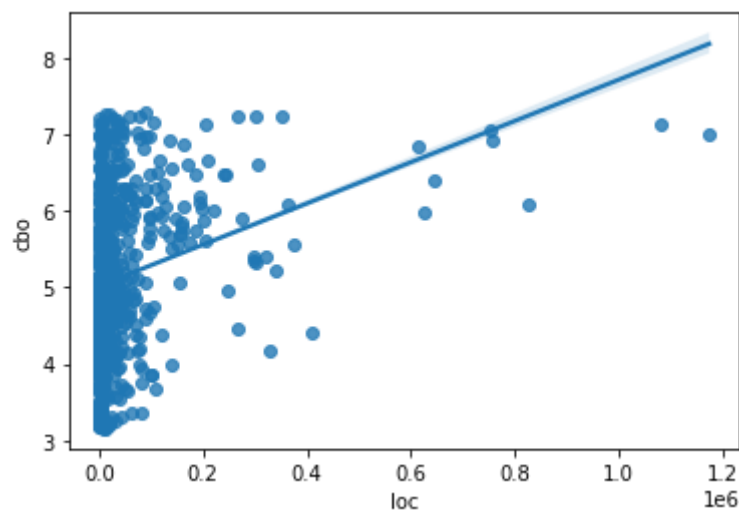
A hipótese apresentada realmente faz sentido, pois ao observar o gráfico é visível uma tendência, onde a linha é um crescente o que mostra que o numero de linhas de codigo influenciam na metrica lcom. Essa ideia é provado através do coeficiente de correlação, o qual mostra um valor proximo a 1, sendo que esse valor é valor é otimo para demonstrar a que a relação é valida.

LOC x CBO

- **Hipótese:**

Quanto menor a quantidade de linha de código, menor é quantidade de acoplamento entre as classes de um projeto.

- **Resultados:**



Coeficiente de correlação de Spearman para loc and cbo $r = 0.3074195157748057$

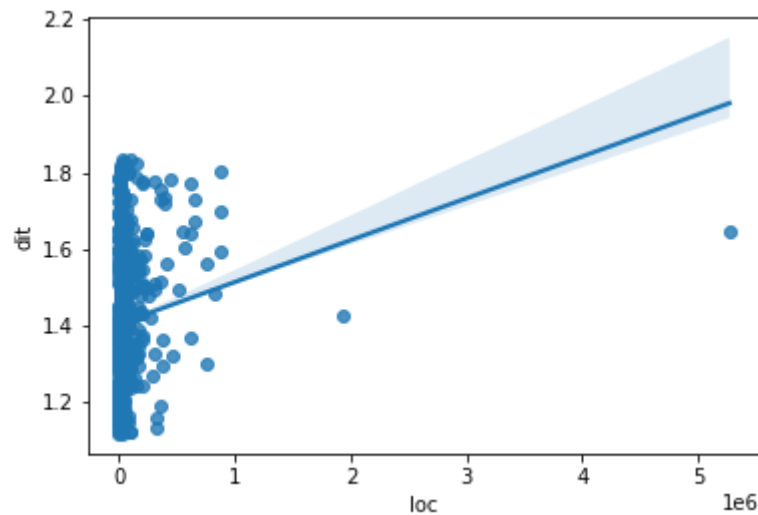
Como observado no grafico é no coeficiente de correlação, não há relação entre loc e cbo. Dessa forma, fica claro que a hipótese gerado não pode ser comprovada através desses dados.

LOC x DIT

- **Hipótese:**

Quanto menor a quantidade de linha de código, menor é quantidade de classes herdadas em um projeto.

- **Resultados:**



Coeficiente de correlação de Spearman para loc and dit $r = 0.2116342579005864$

Assim como na última comparação, a relação entre loc e dir é fraca. Sendo assim, não é possível provar a hipótese através dos dados que foram analisados.