

Universidade Federal da Bahia
Instituto de Matemática

Programa de Pós-graduação em Ciência da Computação

**CARACTERIZAÇÃO DA QUALIDADE
INTERNA DE FERRAMENTAS DE ANÁLISE
ESTÁTICA DE CÓDIGO FONTE**

Joenio Marques da Costa
joenio@joenio.me

QUALIFICAÇÃO DE MESTRADO

Salvador
11 de junho de 2016

Universidade Federal da Bahia
Instituto de Matemática

Joenio Marques da Costa
joenio@joenio.me

**CARACTERIZAÇÃO DA QUALIDADE INTERNA DE
FERRAMENTAS DE ANÁLISE ESTÁTICA DE CÓDIGO FONTE**

*Trabalho apresentado ao Programa de Pós-graduação em
Ciência da Computação do Instituto de Matemática da
Universidade Federal da Bahia como requisito parcial para
obtenção do grau de Mestre em Ciência da Computação.*

Orientadora: Profa. Dra. Christina von Flach G. Chavez
Co-orientador: Prof. Dr. Paulo Roberto Miranda Meirelles

Salvador
11 de junho de 2016

SUMÁRIO

| | |
|---|-----------|
| Capítulo 1—Introdução | 1 |
| 1.1 Motivação | 1 |
| 1.2 Contribuições esperadas | 1 |
| Capítulo 2—Fundamentação teórica | 3 |
| 2.1 Engenharia de Software | 3 |
| 2.1.1 Software científico | 3 |
| 2.1.2 Qualidade de software | 4 |
| 2.1.3 Métricas de código-fonte | 4 |
| 2.1.4 Ferramentas de análise estática de código-fonte | 5 |
| 2.2 Capítulo sobre estatística | 6 |
| Capítulo 3—Metodologia | 7 |
| 3.1 Questão de pesquisa e hipóteses | 7 |
| 3.2 Planejamento do estudo | 8 |
| 3.2.1 Seleção das métricas | 8 |
| 3.2.2 Seleção da ferramenta de análise estática de código-fonte | 9 |
| 3.2.3 Seleção das fontes de ferramentas de análise estática | 9 |
| 3.3 Coleta de dados | 9 |
| 3.3.1 Ferramentas da academia | 10 |
| 3.3.2 Ferramentas da indústria | 10 |
| 3.3.3 Caracterização dos artigos | 11 |
| 3.3.4 Caracterização das ferramentas | 11 |
| 3.4 Exemplo de uso | 12 |
| Capítulo 4—Cronograma | 13 |
| Capítulo 5—Análise | 15 |
| 5.1 Resultados preliminares | 15 |
| 5.1.1 Métricas dos softwares científicos | 20 |
| 5.1.2 Métricas das ferramentas da indústria | 26 |
| 5.1.3 Gráficos das métricas para os softwares científicos | 35 |
| 5.1.4 Gráficos das métricas para as ferramentas da indústria | 46 |
| 5.1.5 Gráfico resumido com distribuição de métricas | 62 |

| | | |
|-----------------------------|-------------------------|-----------|
| 5.1.6 | Regressão | 62 |
| Capítulo 6—Conclusão | | 71 |

LISTA DE FIGURAS

| | | |
|------|--|----|
| 1.1 | The reproducible research problem(VRIES, 2014) | 1 |
| 5.1 | distribuição das métricas para a ferramenta accessanalysis | 36 |
| 5.2 | distribuição das métricas para a ferramenta bakar-ali | 37 |
| 5.3 | distribuição das métricas para a ferramenta error-prone | 38 |
| 5.4 | distribuição das métricas para a ferramenta indus | 39 |
| 5.5 | distribuição das métricas para a ferramenta inputtracer | 40 |
| 5.6 | distribuição das métricas para a ferramenta jastadd | 41 |
| 5.7 | distribuição das métricas para a ferramenta source-meter | 42 |
| 5.8 | distribuição das métricas para a ferramenta srcml | 43 |
| 5.9 | distribuição das métricas para a ferramenta tackle | 44 |
| 5.10 | distribuição das métricas para a ferramenta wala | 45 |
| 5.11 | distribuição das métricas para a ferramenta boon | 47 |
| 5.12 | distribuição das métricas para a ferramenta clang | 48 |
| 5.13 | distribuição das métricas para a ferramenta closure-compiler | 49 |
| 5.14 | distribuição das métricas para a ferramenta cppcheck | 50 |
| 5.15 | distribuição das métricas para a ferramenta cqual | 51 |
| 5.16 | distribuição das métricas para a ferramenta findbugs | 52 |
| 5.17 | distribuição das métricas para a ferramenta findsecuritybugs | 53 |
| 5.18 | distribuição das métricas para a ferramenta jlint | 54 |
| 5.19 | distribuição das métricas para a ferramenta pixy | 55 |
| 5.20 | distribuição das métricas para a ferramenta pmd | 56 |
| 5.21 | distribuição das métricas para a ferramenta rats | 57 |
| 5.22 | distribuição das métricas para a ferramenta smatch | 58 |
| 5.23 | distribuição das métricas para a ferramenta splint | 59 |
| 5.24 | distribuição das métricas para a ferramenta uno | 60 |
| 5.25 | distribuição das métricas para a ferramenta wap | 61 |
| 5.26 | distribuição para todas as ferramenta | 62 |
| 5.27 | distribuição para todas as ferramenta | 63 |
| 5.28 | distribuição para todas as ferramenta | 64 |
| 5.29 | distribuição para todas as ferramenta da indústria | 65 |
| 5.30 | distribuição para todas as ferramenta | 66 |
| 5.31 | distribuição para todas as ferramenta | 67 |
| 5.32 | distribuição para todas as ferramenta | 68 |
| 5.33 | distribuição para todas as ferramenta | 69 |
| 5.34 | distribuição de percentis para todas as ferramentas | 70 |

LISTA DE TABELAS

| | | |
|------|---|----|
| 4.1 | Cronograma até a defesa | 13 |
| 4.2 | Cronograma até a qualificação | 13 |
| 5.1 | Total de artigos analisados por edições do SCAM | 16 |
| 5.2 | Lista de ferramentas do SAMATE - NIST com código fonte não disponível | 17 |
| 5.3 | Lista de ferramentas do SAMATE - NIST com código fonte disponível . . | 18 |
| 5.4 | Lista com total de ferramentas a serem analisadas | 19 |
| 5.5 | percentis da métrica acc | 20 |
| 5.6 | percentis da métrica accm | 20 |
| 5.7 | percentis da métrica amloc | 21 |
| 5.8 | percentis da métrica anpm | 21 |
| 5.9 | percentis da métrica cbo | 21 |
| 5.10 | percentis da métrica dit | 22 |
| 5.11 | percentis da métrica lcom4 | 22 |
| 5.12 | percentis da métrica loc | 22 |
| 5.13 | percentis da métrica mmloc | 23 |
| 5.14 | percentis da métrica noa | 23 |
| 5.15 | percentis da métrica noc | 23 |
| 5.16 | percentis da métrica nom | 24 |
| 5.17 | percentis da métrica npa | 24 |
| 5.18 | percentis da métrica npm | 24 |
| 5.19 | percentis da métrica rfc | 25 |
| 5.20 | percentis da métrica sc | 25 |
| 5.21 | percentis da métrica acc | 26 |
| 5.22 | percentis da métrica accm | 27 |
| 5.23 | percentis da métrica amloc | 27 |
| 5.24 | percentis da métrica anpm | 28 |
| 5.25 | percentis da métrica cbo | 28 |
| 5.26 | percentis da métrica dit | 29 |
| 5.27 | percentis da métrica lcom4 | 29 |
| 5.28 | percentis da métrica loc | 30 |
| 5.29 | percentis da métrica mmloc | 30 |
| 5.30 | percentis da métrica noa | 31 |
| 5.31 | percentis da métrica noc | 31 |
| 5.32 | percentis da métrica nom | 32 |
| 5.33 | percentis da métrica npa | 32 |
| 5.34 | percentis da métrica npm | 33 |

| | | |
|------|------------------------------------|----|
| 5.35 | percentis da métrica rfc | 33 |
| 5.36 | percentis da métrica sc | 34 |

1.1 MOTIVAÇÃO

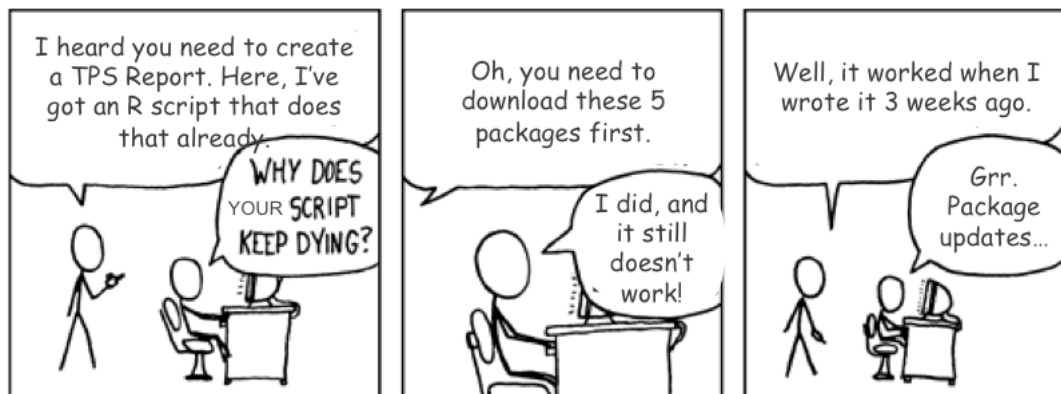


Figura 1.1: The reproducible research problem(VRIES, 2014)

Reprodutibilidade é a habilidade de replicar um experimento ou estudo em sua totalidade a fim de confirmar suas hipóteses e resultados, apesar de ser uma prática central do método científico ainda é um grande obstáculo em muitos estudos. Enquanto pesquisadores publicam artigos descrevendo e divulgando seus resultados, é raro que façam o mesmo com toda a produção gerada durante a pesquisa. A maioria dos componentes necessários para a reprodução dos resultados de uma pesquisa na engenharia de software – por exemplo, código-fonte e dados – usualmente permanecem não publicados.

Isto se configura como uma barreira para a reprodutibilidade, e consequentemente para a repetição, replicação e variação de estudos (FEITELSON, 2015) já que a disponibilidade de código-fonte é o mínimo necessário para que isto ocorra (PENG, 2011).

Dentro deste contexto, e considerando que muitos estudos ainda sofrem com dificuldades de repetição (KAZMAN, 2016), surge a preocupação de avaliar a qualidade dos softwares científicos a partir de métodos adequados, especialmente em relação a sua manutenibilidade e disponibilidade por serem problemas comuns enfrentados pelos pesquisadores (PRLÍ; PROCTER, 2012).

1.2 CONTRIBUIÇÕES ESPERADAS

1. Valores de referência de métricas de código-fonte para ferramentas de análise estática de código-fonte
2. Um guia de sugestões para refatoração de ferramentas de análise estática de código-fonte

3. Um método para avaliação e caracterização de ferramentas de análise estática de código-fonte
4. ...

CAPÍTULO 2

FUNDAMENTAÇÃO TEÓRICA

2.1 ENGENHARIA DE SOFTWARE

Sistemas de software são utilizados em praticamente todas as áreas do conhecimento humano e têm exercido um papel essencial em nossa sociedade (MAFRA; TRAVASSOS, 2006). A dependência crescente de serviços oferecidos por tais sistemas evidencia a necessidade de produzir software de qualidade, contornando os desafios relacionados a funcionalidades incompletas ou incorretas, custos acima do esperado ou prazos não cumpridos.

Diante desses desafios, surge a Engenharia de Software, uma disciplina centrada no desenvolvimento de sistemas de software através de uma abordagem sistemática, disciplinada, e quantificável para o desenvolvimento, operação e manutenção (SOCIETY, 2014).

Nas últimas décadas, o foco em estudos empíricos na área de Engenharia de Software tem crescido significativamente (STOL; FITZGERALD, 2015), resultando no uso crescente de métodos como surveys, estudos de caso, experimentos e revisões sistemáticas de literatura. Através destes estudos empíricos, pesquisadores transformam a Engenharia de Software em uma disciplina mais científica e controlável – a Engenharia de Software Experimental – provendo meios para avaliar e validar métodos, técnicas, linguagens e ferramentas.

Não raro, muitos destes estudos criam novos sistemas de software, tais sistemas costumam ser utilizados como meio para atingir os resultados da pesquisa ou, em alguns casos, são o próprio fim do estudo realizado. Neste trabalho, tais ferramentas de software são nosso objeto de pesquisa e serão chamados de “software científico” – Portillo-Rodríguez et al. (2012) utiliza o termo “research tool” para designar este mesmo tipo de software.

2.1.1 Software científico

Softwares científicos são ferramentas de software desenvolvidas no decorrer de pesquisas científicas como parte de um estudo, podem ser pequenos scripts, protótipos, ou mesmo produtos de software completos que demonstram ou refletem os resultados de uma pesquisa. Em Engenharia de Software este tipo de software desempenha um papel essencial e sua importância pode ser notada através do grande número de conferências com sessões específicas sobre publicação de ferramentas.

Kon et al. (2011) em um estudo sobre como pesquisas em Engenharia de Software podem se beneficiar do ecossistema de Software Livre faz uma análise de 10 edições do SBES¹ e conclui que apesar do aumento do interesse por parte dos pesquisadores em

¹Simpósio Brasileiro de Engenharia de Software

disponibilizar o código-fonte de suas ferramentas isto ainda é minoria. O que confirma a preocupação de Krishnamurthi e Vitek (2015) em um estudo sobre repetibilidade de pesquisas científicas, onde chamam atenção para o papel central que os artefatos de software possuem em pesquisas de ciência da computação e questionam: “Onde está o software nas pesquisas sobre linguagem de programação?”.

A partir daí podemos afirmar que softwares científicos são peça fundamental para que pesquisadores independentes possam reproduzir, validar ou expandir os resultados encontrados em estudos anteriores e assim aumentar o rigor e a qualidade científica de tais pesquisas (VITEK; KALIBERA, 2011).

2.1.2 Qualidade de software

Qualidade de software diz respeito à quão bem um software é projetado e o quanto este software está em conformidade com o projeto, embora existam inúmeras definições há um consenso de que existem duas dimensões básicas para medir a qualidade de um software, estas dimensões estão relacionadas à características de qualidade interna e de qualidade externa.

Segundo McConnell (2004), qualidade interna são aquelas características que preocupam o desenvolvedor, como: manutenibilidade, flexibilidade, portabilidade, reusabilidade, legibilidade, testabilidade e compreensão. São questões relacionadas ao código-fonte e em como o software foi construído, como design e boas práticas por exemplo.

Qualidade externa são aquelas características que afetam o usuário, como por exemplo: exatidão, usabilidade, eficiência, confiabilidade, integridade, adaptabilidade, acurácia e robustez. Estas características impactam extrinsecamente no uso do software e não em como o software foi construído.

Segundo a ISO/IEC 25010 (ISO, 2011) essas características podem ser divididas em subcaracterísticas. A característica usabilidade por exemplo é dividida nas seguintes subcaracterísticas: capacidade de compreensão, capacidade de aprendizado, operabilidade, atratividade e conformidade. Esta característica está relacionada, por exemplo, a facilidade com que usuários aprendem e utilizam um sistema, e passa por questões como facilidade de instalação, aprendizado, e uso.

Sabe-se que, em algum nível, as características de qualidade interna afetam as características de qualidade externa (MCCONNELL, 2004), softwares que não possuem boa manutenibilidade por exemplo afetam a habilidade de correção de defeitos, que por sua vez afetam as características de exatidão e confiabilidade.

Dito isso podemos perceber que é possível inferir características de qualidade externa de um software a partir de seus atributos de qualidade interna, o que pode ser realizado a partir da análise de suas métricas de código-fonte.

2.1.3 Métricas de código-fonte

Uma métrica, segundo a definição da ISO/IEC 25010 (ISO, 2011), é a composição de procedimentos para a definição de escalas e métodos para medidas, em engenharia de software estas métricas podem ser classificadas em três categorias: métricas de produto, métricas de processo e métricas de projeto.

Métricas de produto são aquelas que descrevem as características de artefatos do desenvolvimento, como documentos, diagramas, código-fonte e arquivos binários. Métricas de processo medem atributos relacionados ao ciclo de desenvolvimento do software. Métricas de projeto são aquelas que descrevem as características dos recursos disponíveis ao desenvolvimento.

Neste trabalho, nosso interesse estão nas métricas de produto, que podem ser classificadas entre internas ou externas, ou seja, aquelas que medem propriedades visíveis apenas aos desenvolvedores ou que medem propriedades visíveis aos usuários, respectivamente. Iremos extrair propriedades dos softwares científicos a fim de medir a sua qualidade interna através de um conjunto de métricas previamente definido, este conjunto tomará como base o trabalho realizado por Meirelles (2013) onde foi realizado um estudo associando qualidade de software à qualidade de código-fonte através da observação de métricas de código-fonte. E será realizado através de ferramentas de análise estática de código-fonte.

2.1.4 Ferramentas de análise estática de código-fonte

Ferramentas de análise estática de código-fonte são ferramentas capazes de realizar a leitura de código-fonte de um projeto de software de forma automatizada ou semi-automatizada e extrair daí informações sobre as entidades do software, como módulos, classes, funções, métodos, variáveis, seus relacionamentos, suas características e diversas outras informações possíveis de serem extraídas diretamente do código-fonte que seja útil ao engenheiro de software.

Estas informações costumam ser aplicadas à tarefas comuns da engenharia de software, como por exemplo, recuperação arquitetural, localização de falhas, manutenção, refatoração, compreensão, análise de performance, visualização, entre outras.

Segundo Kirkov e Agre (2010) estas ferramentas possuem uma anatomia comum, composta de quatro componentes básicos - construção de modelos; algoritmos de análise e reconhecimento de padrões; base de conhecimento de padrões; e representação final.

A construção de modelos de um programa é o primeiro passo e é feito por um parser de código-fonte (BINKLEY, 2007). A base de conhecimento de padrões é usada para representar e armazenar informações sobre potenciais problemas encontrados no código-fonte. O objetivo do algoritmo de análise e reconhecimento de padrões é classificar as informações encontradas no modelo a partir da base de conhecimento de padrões. A representação final é um relatório ou outro tipo de visualização apresentada através de uma interface de usuário apropriada.

Esta representação final pode ser por exemplo um relatório contendo os valores de métricas calculadas para o software sendo analisado, e é esta característica das ferramentas de análise estática de código-fonte que será utilizada neste trabalho para recuperar métricas de produto de software que reflitam os atributos de qualidade dos softwares científicos estudados.

(feedback de Paulo: faltou arrematar e contextualizar com o seu trabalho)

2.2 CAPÍTULO SOBRE ESTATÍSTICA

(ver TCC de Ronaldo e Kanashiro)

(falar que média não serve, distribuição é a solução... etc)

CAPÍTULO 3

METODOLOGIA

Visando então avaliar a qualidade de “softwares científicos” será feito um levantamento de artigos com publicação de ferramentas do domínio *análise estática de código-fonte*. Este domínio foi selecionado com base na experiência do pesquisador nesta área, o que facilita encontrar fontes sobre ferramentas, bem como analisar e estudar tais softwares. O foco em um domínio se justifica pela necessidade de reduzir o escopo do estudo a fim de viabilizar o trabalho dentro do tempo previsto em um programa de mestrado.

Além dos software científicos iremos também incluir ferramentas de análise estática desenvolvidas na indústria, o objetivo é aumentar o número de ferramentas analisadas visto que existe uma suspeita de encontrar um número pequeno de softwares científicos com disponibilidade de código-fonte.

Após o levantamento e seleção das ferramentas será feita a caracterização inicial e posteriormente análise estática do seu código-fonte, onde teremos métricas para cada ferramenta e possivelmente valores referência para ferramentas de análise estática, estes valores de referência darão origem a recomendações de refatoração para ferramentas deste mesmo domínio de aplicação.

Por final traçaremos um paralelo entre características de qualidade externa e valores de métricas de qualidade interna, especialmente em relação à portabilidade e usabilidade a fim de responder nossa questão de pesquisa e validar suas hipóteses.

3.1 QUESTÃO DE PESQUISA E HIPÓTESES

A questão de pesquisa **Q1** - *Quais características das ferramentas de análise estática de código fonte podem ser obtidas a partir da avaliação da sua qualidade interna?* será respondida através da validação das seguintes hipóteses:

- H1:** *Quanto mais antiga for a publicação maior a probabilidade do software científico não estar mais disponível nas fontes indicadas*
- H2:** *É possível calcular valores de referência de métricas de código-fonte para ferramentas de análise estática a partir de um conjunto de softwares científicos e da indústria*
- H3:** *Ferramentas da indústria possuem valores de métricas de código-fonte mais próximos aos valores de referência do que os softwares científicos*
- H4:** *É possível relacionar a subcaracterística de qualidade externa operabilidade à valores de métricas de qualidade interna para os softwares científicos*

A hipótese **H1** será validada com a avaliação dos artigos que publicam software científico com fontes para obtenção mas que não se encontram mais disponíveis.

A hipótese **H2** será validada (depende do capítulo 2.2) ...

A hipótese **H3** será validada a partir do cálculo das métricas de código-fonte das ferramentas da indústria e sua comparação aos valores de referência encontrados para as ferramentas de análise estática.

A hipótese **H4** será validada com a tentativa de instalação e o uso das funções mínimas de cada software científico avaliado, será feito a tentativa de instalar a ferramenta, instruções de instalação e uso serão pesquisados nos artigos relacionados bem como junto ao código-fonte do software. Espera-se que aqueles softwares com maior dificuldade de instalação e uso terão valores de métricas piores.

3.2 PLANEJAMENTO DO ESTUDO

3.2.1 Seleção das métricas

Meirelles (2013) realizou uma série de estudos onde associou características de qualidade de produto de software à características de qualidade de código-fonte, através de métricas de código-fonte, em um destes estudos utilizou-se métricas que medem aspectos relevantes à manutenibilidade do software, como a preocupação aqui também está na manutenibilidade das ferramentas iremos utilizar a mesma seleção de métricas utilizada por Meirelles (2013).

- **CBO** *Coupling Between Objects (Acoplamento entre objetos)*: mede o acoplamento entre objetos do software (CHIDAMBER; KEMERER, 1994).
- **LCOM4** *Lack of Cohesion in Methods (Ausência de coesão em métodos)*: mede o grau de falta de coesão em métodos (HITZ; MONTAZERI, 1995).
- **SC** *Structural Complexity (Complexidade estrutural)*: mede a complexidade do software (DARCY et al., 2005).
- **AMLOC** *Average Method LOC (Média do número de linhas de código por método)*: indica se o código está bem distribuído entre os métodos, quanto maior mais “pesados” são os métodos (??)
- **ACCM** *Average Cyclomatic Complexity per Method (Média de complexidade ciclomática por método)*: mede a complexidade do programa (MCCABE, 1976).
- **RFC** *Response For a Class (Resposta para uma classe)*: número de métodos dentre todos os métodos que podem ser invocados em resposta a uma mensagem enviada por um objeto de uma classe (SHARBLE; COHEN, 1993).
- **DIT** *Depth of Inheritance Tree (Profundidade da árvore de herança)*: mede o número de ancestrais de uma classe (SHIH et al., 1997).
- **NOC** *Number Of Children (Número de filhos)*: número total de filhos de uma classe (ROSENBERG; HYATT, 1997).

- **COF** *Coupling Factor (Fator de acoplamento)*: razão entre o número máximo possível de acoplamentos no sistema e o número atual de acoplamentos possíveis por herança (HARRISON; COUNSELL; NITHI, 1998).

Estas métricas serão coletadas para cada classe/módulo presente nas ferramentas de análise estática selecionadas e servirão de base para avaliar a qualidade das mesmas.

3.2.2 Seleção da ferramenta de análise estática de código-fonte

Para realizar a caracterização das ferramentas e extrair suas métricas será necessário uma ferramenta que permita automatizar este processo, neste trabalho utilizaremos a ferramenta de análise estática Analizo (TERCEIRO et al., 2010).

Analizo é um *toolkit* livre, multi-linguagem e extensível para análise de código-fonte, calcula uma grande quantidade de métricas, como CBO, LCOM4, RFC, LOC, entre outras, e suporta análise das linguagens de programação C, C++ e Java.

É uma ferramenta mantida constantemente, com desenvolvedores ativos, e atualizações frequentes, sua última versão 1.19.0 lançada em 18 de Fevereiro de 2016 será a versão utilizada neste estudo.

3.2.3 Seleção das fontes de ferramentas de análise estática

Para ser possível validar as hipóteses aqui levantadas é necessário realizar uma busca por ferramentas de análise estática desenvolvidas no contexto da academia e da indústria, para isso, será feito um planejamento detalhado para realizar a seleção de ferramentas em cada um destes contextos.

No contexto acadêmica a busca por ferramentas será feita através de artigos publicados em conferências que tenham histórico de publicação sobre ferramentas de análise estática de código fonte. Estes artigos serão analisados e aqueles com publicação de ferramenta de análise estática serão selecionados.

Na indústria, a busca por ferramentas será feita a partir de referências encontradas na internet, algumas organizações mantêm listas de ferramentas para análise de código-fonte, a Wikipedia por exemplo, mantêm uma lista de ferramentas, estas referências serão utilizadas como ponto de partida e cada ferramenta será analisada a fim de validar se são da indústria ou surgiram em contexto acadêmico.

Uma vez que as ferramentas tenham sido selecionadas iniciaremos a extração de seus atributos de qualidade interna a partir do cálculo de suas métricas com o Analizo.

3.3 COLETA DE DADOS

A partir das fontes selecionadas na etapa anterior serão realizadas duas atividades para identificar e mapear as ferramentas de análise estática com código-fonte disponível, uma atividade relacionada ao levantamento de ferramentas da academia, outra atividade relacionada ao levantamento de ferramentas da indústria.

3.3.1 Ferramentas da academia

A seleção de ferramentas será realizada através de uma revisão estruturada dos artigos selecionados a partir das seguintes conferências:

- ASE - Automated Software Engineering¹
- CSMR² - Conference on Software Maintenance and Reengineering³
- SCAM - Source Code Analysis and Manipulation Working Conference⁴
- ICSME - International Conference on Software Maintenance and Evolution⁵

Chamamos de revisão estruturada um processo disciplinado para seleção de artigos a partir de critérios bem definidos de forma que seja possível a reprodução do estudo por parte de pesquisadores interessados. Alguns resultados preliminares podem ser consultados na Tabela 5.1 da Seção 5.1.

3.3.2 Ferramentas da indústria

A seleção de ferramentas da indústria será feita de forma não estruturada a partir de uma busca livre e manual em fontes encontradas na Internet sobre ferramentas de análise estática. Nos estudos já realizados as seguintes fontes foram selecionadas:

- Projeto SAMATE⁶ - *Software Assurance Metrics and Tool Evaluation* disponível em NIST (2016) mantém uma lista de ferramentas de análise estática, detalhes sobre o projeto pode ser encontrado em Ribeiro (2015).
- O software Spin mantém em seu site uma lista de ferramentas comerciais e de pesquisa para análise estática de código-fonte para C em Spin (2016).
- O Instituto de Engenharia de Software do CERT mantém uma lista de ferramentas de análise estática em CERT (2016).
- O software Flawfinder oferece em seu site um link com referências para inúmeras ferramentas livres, proprietárias e gratuitas de ferramentas de análise estática e outros tipos de análise em Wheeler (2015).
- Uma outra fonte contendo uma relação expressiva de ferramentas é mantida na Wikipedia em Wikipedia (2016).

Estas fontes serão pesquisadas manualmente em busca de ferramentas de análise estática que tenham sido desenvolvidas no contexto da indústria, alguns resultados preliminares podem ser encontrados nas Tabelas 5.3 e 5.2.

¹<http://ase-conferences.org>

²A conferência CSMR tornou-se SANER - Software Analysis, Evolution, and Reengineering a partir da edição 2015.

³<http://ansymore.uantwerpen.be/csmr-wcre>

⁴<http://www.ieee-scam.org>

⁵<http://www.icsme.org>

⁶<http://samate.nist.gov>

3.3.3 Caracterização dos artigos

Os artigos selecionados a partir da revisão estruturada serão avaliados a fim de caracterizar se se tratam de publicação de ferramenta de análise estática de código-fonte, esta revisão será realizada de forma semi-automatizada, o primeiro passo será automatizado a partir de um script⁷ que busca os seguintes termos no conteúdo dos artigos:

```
"tool" OU "framework"; E  
"download" OU "available"; E  
"http" OU "ftp"; E  
"static analysis" OU "parser".
```

O segundo passo, manual, é realizar uma leitura do artigo a fim de identificar se realmente trata-se de um artigo com publicação de *software científico*, uma vez que se confirme que o artigo publica um software, identifica-se se o software é uma ferramenta de análise estática, softwares que sejam mais abrangentes do que apenas análise estática mas que contenham esta função em seu conjunto também será considerados.

Uma vez identificado os artigos que publicam *softwares científicos* de análise estática, procuramos no próprio artigo referências de onde encontrar o software, neste momento algumas ações serão tomadas a partir da situação encontrada.

- Aqueles autores que afirmam que a ferramenta está disponível mas o artigo não cita referências de onde encontrar serão contactados por email solicitando informações de onde obter o código-fonte.
- Os artigos que indicam onde obter o código-fonte mas o acesso ao local indicado não está disponível, ou está disponível mas o software não se encontra lá, os autores também serão contactados solicitando informações atualizadas de onde obter uma cópia do código-fonte da ferramenta
- Os demais artigos que indicam onde obter o código-fonte e a referência está correta, iremos fazer download da última versão disponível do software

Uma vez que os autores contactados por email respondam com informações de onde obter o software iremos adicionar estes softwares na lista de softwares a serem analisados.

3.3.4 Caracterização das ferramentas

As ferramentas da indústria e acadêmicas serão analisadas com o Analizo para extração das métricas de código-fonte previamente selecionadas, estas métricas serão relacionadas à característica e atributos de qualidade das ferramentas.

As métricas coletadas serão também utilizadas para identificar se existem valores de referência para métricas de ferramentas de análise estática, caso existam, estes valores de métricas serão utilizadas para calcular quão distante cada ferramenta analisada se encontra dos valores de referência, isto será feito com base no trabalho realizado Júnior (2015) onde o mesmo estudo foi feito para o sistema Android e seus aplicativos.

⁷<http://github.com/joenio/dissertacao-ufba-2016/blob/master/revisao-estruturada/filter>

3.4 EXEMPLO DE USO

Por fim, os valores de referência encontradas serão documentados e servirão de base para criar um guia de sugestões para refatoração de ferramentas de análise estática. Tomaremos como exemplo de uso a própria ferramenta Analizo, onde iremos calcular suas métricas e a partir de uma comparação com os valores de referência indicaremos refatorações que façam a ferramenta de aproximar dos valores de referência.

Estas indicações serão feitas utilizando o mesmo método de Almeida e Miranda (2010) onde foi feito um estudo mapeando boas práticas de programação em valores de métricas de código-fonte, estas boas práticas são baseadas nos trabalhos de (MARTIN; HAN, 2012) e (BECK, 2007) sobre *Clean Code*, onde sugerem práticas de desenvolvimento úteis para que um código tenham expressividade, simplicidade e flexibilidade. Neste trabalho os autores identificam melhorias de implementação através do uso de valores de métricas e oferecem aos desenvolvedores uma maneira de pensarem em melhorias para os seus códigos.

CAPÍTULO 4

CRONOGRAMA

Tabela 4.1: Cronograma até a defesa

| Atividade | Junho | Julho | Agosto | Setembro | Outubro | Novembro (?) |
|--------------|-------|-------|--------|----------|---------|--------------|
| Qualificação | | | | | | |
| ??? | | | | | | |
| ??? | | | | | | |
| ??? | | | | | | |

Tabela 4.2: Cronograma até a qualificação

| Data | Atividade |
|---------------------|---|
| 20 Junho | texto final para os orientadores |
| 22/23 Junho | feedback final dos orientadores |
| 23 à 27 Junho | correções finais |
| 27 Junho | envio para banca |
| 27 Junho à 04 Julho | compilação dos últimos resultados, inclusive o que não entraram no texto da |
| 6/7/8 Julho | (a depender do dia da sua viagem para SSA): Prévia/ensaio da apresentação |
| 11 Julho | Qualificação, na UFBA |

5.1 RESULTADOS PRELIMINARES

A Tabela 5.1 apresenta um resumo do número de artigos em cada edição do SCAM e quantos artigos trazem publicação de ferramenta de análise estática com código fonte disponível.

As Tabelas 5.3 e 5.2 apresentam ferramentas do NIST após avaliação inicial sobre disponibilidade do código-fonte. Das 54 ferramentas apenas 19 tinham código fonte disponível.

Assim, temos um total de 19 ferramentas da indústria com código-fonte disponível e ??? da academia com código fonte disponível, é preciso avaliar em qual linguagem de programação foi escrita cada ferramentas pois só iremos analisar aquelas em C, C++ ou Java que são suportadas pelo Analizo.

A ferramenta utilizada para identificar a linguagem de programação em que estes softwares foram escritos foi a sloccount, uma ferramenta livre para contagem de linhas de código fonte, onde se calcula em quais linguagens de programação um software foi escrito.

Após análise ficamos com um total de 25 ferramentas, 15 da indústria e 10 da academia, a Tabela 5.4 traz um resumo de todos as ferramentas analisadas.

Tabela 5.1: Total de artigos analisados por edições do SCAM

| Edição | Total de artigos | Script filter | Artigos com ferramenta |
|-----------|------------------|---------------|------------------------|
| SCAM 2001 | 23 | 6 | - |
| SCAM 2002 | 18 | 6 | - |
| SCAM 2003 | 21 | 7 | - |
| SCAM 2004 | 17 | 3 | - |
| SCAM 2005 | 19 | 7 | - |
| SCAM 2006 | 22 | 9 | 2 |
| SCAM 2007 | 23 | 7 | 1 |
| SCAM 2008 | 29 | 14 | - |
| SCAM 2009 | 20 | 10 | - |
| SCAM 2010 | 21 | 15 | 1 |
| SCAM 2011 | 21 | 9 | 1 |
| SCAM 2012 | 22 | 12 | 4 |
| SCAM 2013 | 24 | 12 | - |
| SCAM 2014 | 35 | 16 | 1 |
| SCAM 2015 | 30 | 18 | ? |
| Total | 315 | 133 | 10 |

Tabela 5.2: Lista de ferramentas do SAMATE - NIST com código fonte não disponível

| Ferramenta | Avaliacao |
|----------------------------------|--------------------------|
| ABASH | código não disponível |
| ApexSec Security Console | código não disponível |
| Astrée | código não disponível |
| bugScout | código não disponível |
| C/C++test® | código não disponível |
| dotTEST™ | código não disponível |
| Jtest® | código não disponível |
| HP Code Advisor (cadvice) | código não disponível |
| Checkmarx CxSAST | código não disponível |
| CodeCenter | código não disponível |
| CodePeer | código não disponível |
| CodeSecure | site offline |
| CodeSonar | código não disponível |
| Coverity SAVE™ | código não disponível |
| Csur | código não disponível |
| DoubleCheck | código não disponível |
| Fluid | código não disponível |
| Goanna Studio and Goanna Central | código não disponível |
| HP QAInspect | código não disponível |
| Insight | código não disponível |
| ObjectCenter | código não disponível |
| Parfait | código não disponível |
| PLSQLScanner 2008 | código não disponível |
| PHP-Sat | link para código offline |
| PolySpace | código não disponível |
| PREfix and PREfast | código não disponível |
| QA-C, QA-C++, QA-J | código não disponível |
| Qualitychecker | código não disponível |
| Rational AppScan Source Edition | código não disponível |
| Resource Standard Metrics (RSM) | código não disponível |
| SCA | código não disponível |
| SPARK tool set | código não disponível |
| TBmisra®, TBsecure® | código não disponível |
| PVS-Studio | código não disponível |
| xg++ | código não disponível |

Tabela 5.3: Lista de ferramentas do SAMATE - NIST com código fonte disponível

| Ferramenta | Avaliacao |
|---|-------------------|
| BOON | código disponível |
| Clang Static Analyzer | código disponível |
| Closure Compiler | código disponível |
| Cppcheck | código disponível |
| CQual | código disponível |
| FindBugs | código disponível |
| FindSecurityBugs | código disponível |
| Flawfinder | código disponível |
| Jlint | código disponível |
| LAPSE | código disponível |
| Pixy | código disponível |
| PMD | código disponível |
| pylint | codigo disponivel |
| RATS (Rough Auditing Tool for Security) | código disponível |
| Smatch | código disponível |
| Splint | código disponível |
| UNO | código disponível |
| Yasca | código disponível |
| WAP | código disponível |

Tabela 5.4: Lista com total de ferramentas a serem analisadas

| Ferramenta | Linguagem | Fonte |
|-----------------------|--------------------|-----------|
| BOON | ansic | industria |
| CQual | ansic | industria |
| RATS | ansic | industria |
| Smatch | ansic | industria |
| Splint | ansic | industria |
| UNO | ansic | industria |
| Clang Static Analyzer | cpp | industria |
| Cppcheck | cpp | industria |
| Jlint | cpp | industria |
| WAP | java | industria |
| Closure Compiler | java | industria |
| FindBugs | java | industria |
| FindSecurityBugs | java | industria |
| Pixy | java | industria |
| PMD | java | industria |
| Indus | java | academia |
| TACLE | java | academia |
| JastAdd | java | academia |
| WALA | java | academia |
| error-prone | java | academia |
| AccessAnalysis | java | academia |
| Bakar Alir | java/ada/python | academia |
| InputTracer | ansic | academia |
| Simplicissimus | ansic | academia |
| srcML | cpp/cs (cs = C# ?) | academia |
| Source Meter | java | academia |

5.1.1 Métricas dos softwares científicos

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|----------------|---------|-----|-----|-----|-----|-----|-----|------|------|-------|
| accessanalysis | 91 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 4.0 | 5.5 | 14.7 |
| bakarali | 611 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.9 |
| errorprone | 1703 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 3.0 | 20.0 |
| indus | 214 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 5.0 | 9.0 | 14.6 |
| inputtracer | 2045 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 13.0 | 29.0 | 215.9 |
| jastadd | 59 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 4.0 | 21.8 | 32.1 | 36.5 |
| sourcemeeter | 143 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.5 | 7.6 | 15.9 | 37.9 |
| srcml | 871 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 6.0 | 16.5 | 92.9 |
| tacle | 34 | 0.0 | 0.0 | 0.0 | 1.0 | 1.5 | 6.5 | 9.4 | 11.7 | 35.1 |
| wala | 2626 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 3.0 | 14.0 | 30.0 | 113.0 |

Tabela 5.5: percentis da métrica acc

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|----------------|---------|-----|-----|-----|-----|-----|-----|-----|-----|------|
| accessanalysis | 91 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 1.7 | 2.7 | 3.2 | 5.0 |
| bakarali | 611 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 2.0 |
| errorprone | 1703 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 2.2 | 3.5 | 6.5 |
| indus | 214 | 0.0 | 1.0 | 1.0 | 1.0 | 1.0 | 2.3 | 3.6 | 4.2 | 8.6 |
| inputtracer | 2045 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.1 | 3.0 | 5.0 | 12.1 |
| jastadd | 59 | 0.0 | 1.0 | 1.0 | 1.0 | 1.1 | 1.7 | 2.5 | 3.1 | 4.8 |
| sourcemeeter | 143 | 0.4 | 1.0 | 1.0 | 1.0 | 1.0 | 2.0 | 3.4 | 4.0 | 4.6 |
| srcml | 871 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 2.0 | 2.9 | 9.9 |
| tacle | 34 | 0.3 | 1.0 | 1.0 | 1.0 | 1.8 | 2.1 | 3.1 | 4.1 | 5.2 |
| wala | 2626 | 0.0 | 0.0 | 1.0 | 1.0 | 1.0 | 2.0 | 3.2 | 4.5 | 7.4 |

Tabela 5.6: percentis da métrica accm

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|----------------|---------|-----|-----|-----|-----|-----|------|------|------|--------|
| accessanalysis | 91 | 0.0 | 0.0 | 0.0 | 4.0 | 7.0 | 10.7 | 16.0 | 17.8 | 22.9 |
| bakarali | 611 | 3.0 | 3.0 | 3.0 | 3.0 | 3.0 | 3.0 | 3.0 | 3.1 | 10.0 |
| errorprone | 1703 | 0.0 | 0.0 | 0.0 | 0.0 | 3.0 | 7.1 | 14.3 | 19.9 | 35.0 |
| indus | 214 | 0.0 | 1.0 | 1.0 | 3.0 | 7.5 | 14.6 | 23.7 | 27.8 | 36.9 |
| inputtracer | 2045 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 14.5 | 31.5 | 49.0 | 156.4 |
| jastadd | 59 | 0.0 | 1.0 | 2.2 | 3.0 | 4.6 | 7.7 | 10.7 | 15.4 | 22.4 |
| sourcemeter | 143 | 0.8 | 3.0 | 3.0 | 3.0 | 5.7 | 12.4 | 23.3 | 29.8 | 1105.1 |
| srcml | 871 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 5.4 | 20.5 | 51.4 | 544.6 |
| tacle | 34 | 0.3 | 1.0 | 1.0 | 3.5 | 7.2 | 11.3 | 26.0 | 27.9 | 235.9 |
| wala | 2626 | 0.0 | 0.0 | 1.0 | 3.0 | 4.6 | 9.0 | 16.4 | 24.0 | 43.8 |

Tabela 5.7: percentis da métrica amloc

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|----------------|---------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| accessanalysis | 91 | 0.0 | 0.0 | 0.0 | 0.0 | 0.8 | 1.0 | 1.2 | 1.5 | 1.8 |
| bakarali | 611 | 0.0 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.7 |
| errorprone | 1703 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.3 | 2.0 | 2.0 | 3.0 |
| indus | 214 | 0.0 | 0.0 | 0.1 | 0.5 | 1.0 | 1.4 | 2.0 | 2.5 | 3.0 |
| inputtracer | 2045 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 2.0 | 2.4 | 4.0 |
| jastadd | 59 | 0.0 | 0.0 | 0.0 | 0.3 | 1.0 | 1.4 | 1.9 | 2.0 | 2.0 |
| sourcemeter | 143 | 0.0 | 0.0 | 0.0 | 0.3 | 1.0 | 2.0 | 2.7 | 3.0 | 3.5 |
| srcml | 871 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 2.0 | 2.0 | 4.0 |
| tacle | 34 | 0.0 | 0.0 | 0.0 | 0.5 | 0.9 | 1.2 | 1.7 | 1.9 | 2.5 |
| wala | 2626 | 0.0 | 0.0 | 0.0 | 0.4 | 0.9 | 1.3 | 2.0 | 2.7 | 4.2 |

Tabela 5.8: percentis da métrica anpm

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|----------------|---------|------|-------|-------|-------|-------|-------|-------|-------|--------|
| accessanalysis | 91 | 0.0 | 1.0 | 1.0 | 3.5 | 6.0 | 6.0 | 7.0 | 9.0 | 10.1 |
| bakarali | 611 | 4.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 | 8.0 |
| errorprone | 1703 | 5.0 | 78.0 | 86.0 | 130.0 | 162.0 | 168.0 | 196.0 | 197.0 | 198.0 |
| indus | 214 | 3.1 | 4.0 | 10.0 | 20.2 | 22.0 | 58.5 | 61.7 | 62.0 | 63.0 |
| inputtracer | 2045 | 54.0 | 123.0 | 203.2 | 376.0 | 503.0 | 584.0 | 643.0 | 660.0 | 672.0 |
| jastadd | 59 | 3.5 | 7.0 | 7.0 | 8.0 | 13.0 | 18.5 | 24.0 | 24.0 | 24.0 |
| sourcemeter | 143 | 1.3 | 5.2 | 8.0 | 15.5 | 32.0 | 38.0 | 39.0 | 39.0 | 39.0 |
| srcml | 871 | 0.0 | 5.0 | 5.0 | 9.0 | 94.0 | 105.0 | 148.0 | 166.0 | 176.0 |
| tacle | 34 | 0.0 | 0.0 | 0.3 | 5.8 | 14.5 | 17.0 | 23.0 | 23.0 | 23.0 |
| wala | 2626 | 40.0 | 52.0 | 128.0 | 369.2 | 615.0 | 873.0 | 986.0 | 998.0 | 1011.0 |

Tabela 5.9: percentis da métrica cbo

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|----------------|---------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| accessanalysis | 91 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 1.0 | 2.0 | 2.0 |
| bakarali | 611 | 0.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 |
| errorprone | 1703 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 2.0 | 2.0 | 3.0 |
| indus | 214 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 2.0 | 3.0 | 3.0 | 4.0 |
| inputtracer | 2045 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| jastadd | 59 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 2.0 | 3.0 | 3.0 |
| sourcemeter | 143 | 0.0 | 0.0 | 1.0 | 1.0 | 2.0 | 2.0 | 3.0 | 3.0 | 3.0 |
| srcml | 871 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 4.0 | 6.3 |
| tacle | 34 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 2.0 | 2.0 | 2.0 |
| wala | 2626 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 2.0 | 3.0 | 4.0 | 5.0 |

Tabela 5.10: percentis da métrica dit

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|----------------|---------|-----|-----|-----|-----|-----|------|------|------|------|
| accessanalysis | 91 | 0.0 | 0.0 | 0.0 | 1.0 | 2.0 | 4.0 | 9.0 | 19.0 | 27.3 |
| bakarali | 611 | 1.0 | 2.0 | 2.0 | 3.0 | 8.0 | 10.0 | 12.0 | 14.0 | 19.8 |
| errorprone | 1703 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 3.0 | 6.0 | 9.0 | 24.0 |
| indus | 214 | 0.0 | 1.0 | 1.0 | 1.0 | 2.0 | 4.0 | 7.0 | 13.0 | 25.6 |
| inputtracer | 2045 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 3.0 | 8.0 | 36.6 |
| jastadd | 59 | 0.0 | 1.0 | 1.0 | 1.0 | 2.0 | 6.0 | 20.2 | 25.2 | 96.9 |
| sourcemeter | 143 | 0.4 | 1.0 | 1.0 | 1.0 | 2.0 | 3.0 | 4.0 | 5.0 | 7.6 |
| srcml | 871 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 3.0 | 8.0 | 13.0 | 30.3 |
| tacle | 34 | 0.3 | 1.0 | 1.0 | 1.0 | 2.0 | 3.0 | 4.7 | 7.0 | 10.4 |
| wala | 2626 | 0.0 | 0.0 | 1.0 | 1.0 | 2.0 | 4.0 | 8.0 | 12.8 | 27.0 |

Tabela 5.11: percentis da métrica lcom4

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|----------------|---------|-----|-----|-----|------|------|-------|-------|-------|--------|
| accessanalysis | 91 | 0.0 | 0.0 | 0.0 | 6.5 | 24.0 | 72.5 | 135.0 | 160.0 | 365.9 |
| bakarali | 611 | 6.0 | 6.0 | 6.0 | 12.0 | 24.0 | 30.0 | 36.0 | 42.0 | 60.0 |
| errorprone | 1703 | 0.0 | 0.0 | 0.0 | 0.0 | 6.0 | 20.0 | 64.0 | 119.0 | 333.8 |
| indus | 214 | 0.0 | 1.0 | 2.0 | 6.0 | 34.0 | 107.8 | 199.6 | 280.3 | 584.6 |
| inputtracer | 2045 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 37.0 | 241.8 | 823.4 | 3712.7 |
| jastadd | 59 | 0.0 | 1.9 | 4.6 | 11.0 | 27.0 | 80.0 | 296.0 | 482.2 | 746.6 |
| sourcemeter | 143 | 1.3 | 3.0 | 3.0 | 9.0 | 17.0 | 49.0 | 138.6 | 232.1 | 1105.1 |
| srcml | 871 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 26.0 | 107.0 | 245.5 | 908.6 |
| tacle | 34 | 0.7 | 2.0 | 3.0 | 13.8 | 78.5 | 139.8 | 263.7 | 683.4 | 966.5 |
| wala | 2626 | 0.0 | 0.0 | 1.0 | 6.0 | 20.5 | 55.0 | 133.0 | 232.0 | 586.5 |

Tabela 5.12: percentis da métrica loc

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|----------------|---------|-----|-----|-----|-----|------|------|------|-------|--------|
| accessanalysis | 91 | 0.0 | 0.0 | 0.0 | 4.0 | 10.0 | 19.5 | 27.0 | 36.5 | 45.3 |
| bakarali | 611 | 3.0 | 3.0 | 3.0 | 3.0 | 3.0 | 3.0 | 3.0 | 3.5 | 15.0 |
| errorprone | 1703 | 0.0 | 0.0 | 0.0 | 0.0 | 3.0 | 10.0 | 24.0 | 36.0 | 60.0 |
| indus | 214 | 0.0 | 1.0 | 1.0 | 3.0 | 14.0 | 43.2 | 58.0 | 72.3 | 92.6 |
| inputtracer | 2045 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 23.0 | 87.0 | 191.6 | 1064.8 |
| jastadd | 59 | 0.0 | 1.0 | 3.0 | 3.0 | 9.0 | 18.5 | 31.4 | 40.0 | 91.7 |
| sourcemeter | 143 | 1.3 | 3.0 | 3.0 | 3.0 | 9.0 | 24.5 | 75.6 | 102.8 | 1105.1 |
| srcml | 871 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 9.0 | 38.0 | 92.0 | 671.8 |
| tacle | 34 | 0.3 | 1.0 | 1.0 | 7.0 | 16.0 | 35.8 | 53.8 | 106.5 | 366.9 |
| wala | 2626 | 0.0 | 0.0 | 1.0 | 3.0 | 8.0 | 20.0 | 44.0 | 67.0 | 161.8 |

Tabela 5.13: percentis da métrica mmloc

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|----------------|---------|-----|-----|-----|-----|-----|-----|------|------|-------|
| accessanalysis | 91 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 3.0 | 7.0 | 9.5 | 13.0 |
| bakarali | 611 | 0.0 | 1.0 | 1.0 | 2.0 | 4.0 | 5.0 | 6.0 | 7.0 | 10.0 |
| errorprone | 1703 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 2.0 | 4.0 | 9.0 |
| indus | 214 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 4.0 | 7.0 | 11.3 | 17.7 |
| inputtracer | 2045 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 4.0 | 8.0 | 13.8 | 50.0 |
| jastadd | 59 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 3.0 | 5.0 | 6.2 | 10.5 |
| sourcemeter | 143 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 4.0 | 9.0 | 26.6 | 131.7 |
| srcml | 871 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 3.0 | 6.5 | 22.2 |
| tacle | 34 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 7.5 | 10.0 | 10.3 | 15.0 |
| wala | 2626 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 2.0 | 5.0 | 7.0 | 23.2 |

Tabela 5.14: percentis da métrica noa

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|----------------|---------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| accessanalysis | 91 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 3.5 | 5.0 |
| bakarali | 611 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| errorprone | 1703 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 7.0 |
| indus | 214 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 2.0 | 3.0 |
| inputtracer | 2045 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| jastadd | 59 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 2.1 | 4.8 |
| sourcemeter | 143 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 4.0 | 4.0 | 4.6 |
| srcml | 871 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 4.0 |
| tacle | 34 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.8 | 1.7 | 2.0 | 3.3 |
| wala | 2626 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 3.0 | 9.8 |

Tabela 5.15: percentis da métrica noc

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|----------------|---------|-----|-----|-----|-----|-----|------|------|------|-------|
| accessanalysis | 91 | 0.0 | 0.0 | 0.0 | 1.5 | 3.0 | 7.0 | 17.0 | 21.5 | 27.3 |
| bakarali | 611 | 1.0 | 2.0 | 2.0 | 3.5 | 8.0 | 10.0 | 12.0 | 14.0 | 20.0 |
| errorprone | 1703 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 3.0 | 7.0 | 11.0 | 29.0 |
| indus | 214 | 0.0 | 1.0 | 1.0 | 2.0 | 4.0 | 8.8 | 13.7 | 18.0 | 31.2 |
| inputtracer | 2045 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 11.0 | 27.8 | 135.6 |
| jastadd | 59 | 0.0 | 1.0 | 1.0 | 2.0 | 6.0 | 16.0 | 32.6 | 83.6 | 157.1 |
| sourcemeter | 143 | 0.4 | 1.0 | 1.0 | 2.0 | 3.0 | 5.0 | 6.0 | 8.9 | 13.6 |
| srcml | 871 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 3.0 | 10.0 | 19.0 | 42.2 |
| tacle | 34 | 0.3 | 1.0 | 1.3 | 4.0 | 8.0 | 15.2 | 25.4 | 28.4 | 37.0 |
| wala | 2626 | 0.0 | 0.0 | 1.0 | 2.0 | 4.0 | 8.0 | 15.0 | 23.0 | 45.5 |

Tabela 5.16: percentis da métrica nom

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|----------------|---------|-----|-----|-----|-----|-----|-----|-----|------|-------|
| accessanalysis | 91 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 2.5 | 5.1 |
| bakarali | 611 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 |
| errorprone | 1703 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 3.0 |
| indus | 214 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 6.7 |
| inputtracer | 2045 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 4.0 | 8.0 | 13.8 | 50.0 |
| jastadd | 59 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 2.2 | 6.3 |
| sourcemeter | 143 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 4.6 | 26.6 | 130.7 |
| srcml | 871 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 2.0 | 4.0 | 14.6 |
| tacle | 34 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.7 |
| wala | 2626 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 2.8 | 10.0 |

Tabela 5.17: percentis da métrica npa

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|----------------|---------|-----|-----|-----|-----|-----|------|------|------|-------|
| accessanalysis | 91 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 4.0 | 10.0 | 17.5 | 24.9 |
| bakarali | 611 | 1.0 | 2.0 | 2.0 | 3.0 | 8.0 | 10.0 | 12.0 | 14.0 | 20.0 |
| errorprone | 1703 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 3.0 | 6.0 | 9.0 | 28.0 |
| indus | 214 | 0.0 | 0.0 | 1.0 | 1.0 | 3.0 | 6.0 | 11.0 | 13.3 | 23.7 |
| inputtracer | 2045 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 11.0 | 27.8 | 135.6 |
| jastadd | 59 | 0.0 | 0.9 | 1.0 | 2.0 | 5.0 | 11.0 | 29.4 | 79.1 | 157.1 |
| sourcemeter | 143 | 0.0 | 1.0 | 1.0 | 2.0 | 2.0 | 3.0 | 5.0 | 6.9 | 11.7 |
| srcml | 871 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 3.0 | 9.0 | 13.0 | 34.3 |
| tacle | 34 | 0.3 | 1.0 | 1.0 | 3.0 | 7.0 | 12.8 | 20.7 | 23.3 | 30.7 |
| wala | 2626 | 0.0 | 0.0 | 1.0 | 1.0 | 3.0 | 6.0 | 12.0 | 18.0 | 36.0 |

Tabela 5.18: percentis da métrica npm

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|----------------|---------|-----|-----|-----|-----|------|------|------|-------|-------|
| accessanalysis | 91 | 0.0 | 0.0 | 0.0 | 2.0 | 5.0 | 11.5 | 27.0 | 46.5 | 72.6 |
| bakarali | 611 | 1.0 | 2.0 | 2.0 | 4.0 | 8.0 | 10.0 | 12.0 | 14.0 | 20.0 |
| errorprone | 1703 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 6.0 | 13.0 | 25.0 | 61.0 |
| indus | 214 | 0.0 | 1.0 | 1.0 | 2.0 | 7.0 | 20.0 | 38.7 | 48.0 | 106.0 |
| inputtracer | 2045 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 8.0 | 47.6 | 145.0 | 647.6 |
| jastadd | 59 | 0.0 | 1.0 | 1.0 | 3.0 | 12.0 | 34.0 | 73.4 | 206.1 | 344.6 |
| sourcemeter | 143 | 0.4 | 1.0 | 2.0 | 3.0 | 7.0 | 17.5 | 35.6 | 51.3 | 200.3 |
| srcml | 871 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 9.0 | 27.0 | 41.5 | 177.4 |
| tacle | 34 | 0.3 | 1.7 | 2.0 | 5.5 | 21.5 | 56.8 | 82.1 | 112.4 | 169.9 |
| wala | 2626 | 0.0 | 0.0 | 1.0 | 2.0 | 8.0 | 20.0 | 46.0 | 79.8 | 198.8 |

Tabela 5.19: percentis da métrica rfc

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|----------------|---------|-----|------|------|-------|-------|--------|--------|--------|---------|
| accessanalysis | 91 | 0.0 | 0.0 | 0.0 | 4.0 | 8.0 | 20.0 | 54.0 | 114.0 | 163.8 |
| bakarali | 611 | 8.0 | 16.0 | 16.0 | 17.0 | 64.0 | 80.0 | 96.0 | 112.0 | 144.0 |
| errorprone | 1703 | 0.0 | 0.0 | 0.0 | 0.0 | 165.0 | 417.5 | 844.8 | 1348.2 | 4113.8 |
| indus | 214 | 0.0 | 4.7 | 20.0 | 36.0 | 62.0 | 120.0 | 298.2 | 370.2 | 668.7 |
| inputtracer | 2045 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 539.0 | 1348.0 | 3413.6 | 18666.0 |
| jastadd | 59 | 0.0 | 7.0 | 7.0 | 16.0 | 34.0 | 102.0 | 211.2 | 492.0 | 1014.2 |
| sourcemeter | 143 | 0.0 | 7.2 | 16.4 | 34.5 | 56.0 | 96.0 | 117.0 | 147.2 | 195.0 |
| srcml | 871 | 0.0 | 0.0 | 0.0 | 0.0 | 17.0 | 297.5 | 1008.0 | 1800.0 | 4654.1 |
| tacle | 34 | 0.0 | 0.0 | 0.0 | 5.2 | 18.0 | 45.8 | 68.1 | 87.4 | 100.7 |
| wala | 2626 | 0.0 | 0.0 | 53.0 | 391.0 | 998.0 | 2477.5 | 5059.0 | 8087.5 | 18411.5 |

Tabela 5.20: percentis da métrica sc

5.1.2 Métricas das ferramentas da indústria

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|------------------|---------|-----|-----|-----|-----|-----|------|------|-------|-------|
| boon | 12 | 0.0 | 0.0 | 0.0 | 0.0 | 0.5 | 3.2 | 4.9 | 5.9 | 6.8 |
| clang | 579 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 3.0 | 9.0 | 44.2 |
| closure | 35 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.9 | 4.0 |
| cppcheck | 338 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 18.0 | 51.7 | 264.5 |
| cqual | 78 | 0.0 | 0.0 | 0.0 | 0.0 | 3.0 | 24.0 | 85.7 | 110.0 | 187.6 |
| findbugs | 1486 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 3.0 | 11.0 | 21.8 | 129.5 |
| findsecuritybugs | 91 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 5.0 | 15.5 | 27.2 |
| jlint | 3 | 0.0 | 0.2 | 0.4 | 1.0 | 2.0 | 4.0 | 5.2 | 5.6 | 5.9 |
| pixy | 229 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 6.0 | 18.0 | 38.8 | 80.9 |
| pmd | 1340 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 9.0 | 19.0 | 82.6 |
| rats | 19 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 8.0 | 13.4 | 15.1 | 15.8 |
| smatch | 483 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 19.0 | 93.9 | 326.7 |
| splint | 681 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 7.0 | 34.0 | 96.0 | 544.6 |
| uno | 19 | 0.0 | 0.0 | 0.0 | 1.0 | 4.0 | 34.0 | 95.4 | 129.4 | 132.3 |
| wap | 338 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 4.0 | 17.1 | 45.8 |

Tabela 5.21: percentis da métrica acc

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|------------------|---------|------|------|------|------|------|-------|-------|-------|-------|
| boon | 12 | 0.1 | 0.6 | 1.0 | 1.4 | 4.1 | 5.4 | 7.3 | 8.0 | 8.5 |
| clang | 579 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 1.4 | 3.0 | 4.1 | 7.7 |
| closure | 35 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 1.6 | 2.3 | 5.0 |
| cppcheck | 338 | 0.0 | 0.0 | 0.0 | 1.0 | 1.2 | 3.0 | 6.8 | 9.4 | 17.3 |
| cqual | 78 | 0.0 | 0.0 | 1.0 | 1.0 | 1.2 | 2.6 | 6.3 | 8.7 | 38.4 |
| findbugs | 1486 | 0.0 | 0.0 | 1.0 | 1.0 | 1.3 | 2.5 | 4.8 | 7.0 | 12.4 |
| findsecuritybugs | 91 | 1.0 | 1.0 | 1.0 | 1.3 | 2.0 | 3.0 | 4.2 | 5.8 | 7.9 |
| jlint | 3 | 15.2 | 15.2 | 15.3 | 15.5 | 15.9 | 160.5 | 247.2 | 276.1 | 299.2 |
| pixy | 229 | 0.0 | 0.0 | 1.0 | 1.0 | 1.5 | 2.0 | 4.1 | 5.5 | 9.1 |
| pmd | 1340 | 0.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.9 | 3.0 | 4.0 | 7.0 |
| rats | 19 | 0.0 | 0.0 | 0.0 | 0.0 | 4.9 | 5.5 | 8.8 | 9.4 | 10.2 |
| smatch | 483 | 0.0 | 0.0 | 0.2 | 1.0 | 1.8 | 3.2 | 4.5 | 6.1 | 10.7 |
| splint | 681 | 0.0 | 0.0 | 1.0 | 1.0 | 1.0 | 2.0 | 4.0 | 6.0 | 15.0 |
| uno | 19 | 0.0 | 0.0 | 0.8 | 1.0 | 4.7 | 6.9 | 8.2 | 11.5 | 18.1 |
| wap | 338 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 2.0 | 5.6 | 23.9 |

Tabela 5.22: percentis da métrica accm

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|------------------|---------|------|------|------|------|------|------|------|-------|-------|
| boon | 12 | 0.4 | 2.2 | 4.0 | 5.6 | 18.3 | 31.1 | 38.8 | 43.2 | 46.9 |
| clang | 579 | 0.0 | 0.0 | 0.0 | 1.0 | 1.7 | 9.3 | 21.4 | 29.1 | 56.0 |
| closure | 35 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 4.0 | 7.8 | 17.3 |
| cppcheck | 338 | 0.0 | 0.0 | 0.0 | 4.0 | 9.6 | 20.9 | 30.9 | 45.5 | 83.0 |
| cqual | 78 | 0.0 | 0.0 | 1.0 | 4.2 | 9.2 | 18.6 | 48.6 | 86.2 | 234.9 |
| findbugs | 1486 | 0.0 | 0.0 | 1.0 | 3.0 | 6.0 | 12.3 | 21.7 | 30.0 | 59.9 |
| findsecuritybugs | 91 | 2.8 | 3.0 | 3.2 | 5.3 | 7.5 | 12.8 | 21.4 | 26.8 | 53.0 |
| jlint | 3 | 11.8 | 15.2 | 19.4 | 31.9 | 52.8 | 69.5 | 79.5 | 82.8 | 85.5 |
| pixy | 229 | 0.0 | 0.0 | 1.3 | 3.7 | 7.0 | 15.0 | 28.7 | 39.5 | 64.7 |
| pmd | 1340 | 0.0 | 1.0 | 3.0 | 3.0 | 4.6 | 9.6 | 16.0 | 22.0 | 37.9 |
| rats | 19 | 0.0 | 0.0 | 0.0 | 0.0 | 23.2 | 33.5 | 50.2 | 57.8 | 60.1 |
| smatch | 483 | 0.0 | 0.0 | 0.2 | 4.6 | 9.0 | 14.9 | 22.4 | 26.0 | 51.2 |
| splint | 681 | 0.0 | 0.0 | 1.0 | 4.5 | 9.0 | 15.0 | 28.0 | 42.7 | 129.6 |
| uno | 19 | 0.0 | 0.0 | 0.8 | 12.0 | 27.5 | 36.9 | 62.1 | 87.4 | 214.3 |
| wap | 338 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 7.0 | 27.9 | 119.7 | 642.3 |

Tabela 5.23: percentis da métrica amloc

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|------------------|---------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| boon | 12 | 0.1 | 0.4 | 0.8 | 1.0 | 1.4 | 2.0 | 2.0 | 2.1 | 2.3 |
| clang | 579 | 0.0 | 0.0 | 0.0 | 0.0 | 0.6 | 1.3 | 2.0 | 2.4 | 3.1 |
| closure | 35 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 2.0 | 2.0 |
| cppcheck | 338 | 0.0 | 0.0 | 0.0 | 0.0 | 0.6 | 1.3 | 2.0 | 2.4 | 3.0 |
| cqual | 78 | 0.0 | 0.0 | 0.9 | 1.0 | 1.6 | 2.0 | 2.4 | 2.8 | 4.8 |
| findbugs | 1486 | 0.0 | 0.0 | 0.0 | 0.5 | 1.0 | 1.3 | 2.0 | 2.3 | 4.0 |
| findsecuritybugs | 91 | 0.4 | 0.5 | 0.6 | 0.7 | 1.0 | 1.0 | 1.8 | 2.1 | 3.0 |
| jlint | 3 | 1.2 | 1.2 | 1.3 | 1.4 | 1.7 | 2.1 | 2.3 | 2.4 | 2.5 |
| pixy | 229 | 0.0 | 0.0 | 0.1 | 0.5 | 1.0 | 1.5 | 2.0 | 3.0 | 3.6 |
| pmd | 1340 | 0.0 | 0.0 | 0.0 | 0.2 | 1.0 | 1.5 | 2.0 | 2.0 | 3.1 |
| rats | 19 | 0.0 | 0.0 | 0.0 | 0.0 | 1.2 | 1.5 | 1.7 | 1.8 | 1.8 |
| smatch | 483 | 0.0 | 0.0 | 0.1 | 1.0 | 1.0 | 1.7 | 2.0 | 2.2 | 3.0 |
| splint | 681 | 0.0 | 0.0 | 0.5 | 1.0 | 1.0 | 1.4 | 2.0 | 2.2 | 3.9 |
| uno | 19 | 0.0 | 0.0 | 0.8 | 1.2 | 1.5 | 1.7 | 2.0 | 2.0 | 2.1 |
| wap | 338 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.7 | 1.0 | 2.8 | 7.6 |

Tabela 5.24: percentis da métrica anpm

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|------------------|---------|------|-------|-------|-------|-------|-------|-------|-------|-------|
| boon | 12 | 0.0 | 0.0 | 0.0 | 0.0 | 2.5 | 5.2 | 6.0 | 6.0 | 6.0 |
| clang | 579 | 10.0 | 21.8 | 35.8 | 60.0 | 94.0 | 125.0 | 139.0 | 140.0 | 145.0 |
| closure | 35 | 0.0 | 0.0 | 0.0 | 1.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 |
| cppcheck | 338 | 17.4 | 25.0 | 31.0 | 62.8 | 93.0 | 114.0 | 140.0 | 145.0 | 150.0 |
| cqual | 78 | 0.0 | 0.8 | 1.0 | 8.0 | 40.0 | 46.0 | 51.0 | 51.3 | 53.0 |
| findbugs | 1486 | 8.8 | 66.0 | 114.0 | 253.0 | 393.0 | 506.0 | 554.0 | 562.8 | 570.0 |
| findsecuritybugs | 91 | 0.0 | 0.0 | 0.0 | 3.0 | 5.0 | 10.0 | 15.0 | 15.0 | 16.0 |
| jlint | 3 | 0.0 | 0.1 | 0.2 | 0.5 | 1.0 | 1.5 | 1.8 | 1.9 | 2.0 |
| pixy | 229 | 17.0 | 36.4 | 42.8 | 68.0 | 81.0 | 89.0 | 108.0 | 115.0 | 117.0 |
| pmd | 1340 | 24.8 | 61.0 | 95.0 | 162.0 | 266.0 | 338.0 | 378.0 | 389.0 | 400.0 |
| rats | 19 | 0.7 | 3.6 | 4.0 | 5.5 | 6.0 | 8.0 | 8.0 | 8.0 | 8.0 |
| smatch | 483 | 16.0 | 44.1 | 49.0 | 68.0 | 109.0 | 116.0 | 122.0 | 123.0 | 123.0 |
| splint | 681 | 0.0 | 192.0 | 220.0 | 277.0 | 300.0 | 317.0 | 320.0 | 321.0 | 322.0 |
| uno | 19 | 3.0 | 3.0 | 3.0 | 3.5 | 7.0 | 12.5 | 14.0 | 14.0 | 14.0 |
| wap | 338 | 29.5 | 35.0 | 35.0 | 36.0 | 36.0 | 37.0 | 43.0 | 44.1 | 50.6 |

Tabela 5.25: percentis da métrica cbo

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|------------------|---------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| boon | 12 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| clang | 579 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 2.0 | 2.0 |
| closure | 35 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.6 | 1.0 | 1.0 |
| cppcheck | 338 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 2.0 | 2.0 | 2.0 |
| cqual | 78 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| findbugs | 1486 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 2.0 | 4.0 | 6.0 | 7.0 |
| findsecuritybugs | 91 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 1.0 | 5.0 | 5.0 | 5.0 |
| jlint | 3 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| pixy | 229 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 1.0 | 2.0 | 2.0 |
| pmd | 1340 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 4.0 | 4.0 | 5.0 | 6.0 |
| rats | 19 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| smatch | 483 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| splint | 681 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| uno | 19 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| wap | 338 | 0.0 | 0.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 3.0 |

Tabela 5.26: percentis da métrica dit

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|------------------|---------|-----|-----|-----|-----|-----|------|------|------|-------|
| boon | 12 | 0.1 | 0.6 | 1.0 | 1.0 | 1.0 | 1.0 | 2.8 | 3.4 | 3.9 |
| clang | 579 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 3.0 | 5.0 | 8.0 | 16.4 |
| closure | 35 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 1.3 | 2.0 |
| cppcheck | 338 | 0.0 | 0.0 | 0.0 | 1.0 | 2.0 | 5.0 | 9.0 | 14.3 | 50.4 |
| cqual | 78 | 0.0 | 0.0 | 1.0 | 1.0 | 1.0 | 3.8 | 12.3 | 17.1 | 208.3 |
| findbugs | 1486 | 0.0 | 0.0 | 1.0 | 1.0 | 2.0 | 4.0 | 7.0 | 11.0 | 27.5 |
| findsecuritybugs | 91 | 1.0 | 1.0 | 1.0 | 1.0 | 2.0 | 2.0 | 2.0 | 2.5 | 6.5 |
| jlint | 3 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 2.0 | 2.6 | 2.8 | 3.0 |
| pixy | 229 | 0.0 | 0.0 | 1.0 | 1.0 | 3.0 | 4.0 | 7.2 | 11.0 | 21.2 |
| pmd | 1340 | 0.0 | 1.0 | 1.0 | 1.0 | 2.0 | 4.0 | 6.0 | 8.0 | 22.6 |
| rats | 19 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 26.0 | 49.4 | 51.1 | 51.8 |
| smatch | 483 | 0.0 | 0.0 | 0.2 | 1.0 | 1.0 | 2.0 | 4.0 | 7.0 | 31.2 |
| splint | 681 | 0.0 | 0.0 | 1.0 | 1.0 | 2.0 | 4.0 | 8.0 | 13.0 | 65.2 |
| uno | 19 | 0.0 | 0.0 | 0.8 | 1.0 | 2.0 | 3.0 | 6.0 | 6.0 | 6.0 |
| wap | 338 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 3.0 | 5.0 | 11.6 | 86.0 |

Tabela 5.27: percentis da métrica lcom4

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|------------------|---------|------|-------|-------|-------|-------|--------|--------|--------|--------|
| boon | 12 | 0.4 | 2.2 | 4.2 | 8.2 | 59.0 | 139.0 | 403.0 | 1008.8 | 1573.8 |
| clang | 579 | 0.0 | 0.0 | 0.0 | 1.0 | 4.0 | 27.0 | 84.6 | 144.3 | 289.7 |
| closure | 35 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 4.4 | 7.8 | 17.3 |
| cppcheck | 338 | 0.0 | 0.0 | 0.0 | 8.0 | 54.5 | 302.8 | 1102.6 | 2527.9 | 5240.0 |
| cqual | 78 | 0.0 | 0.0 | 4.5 | 9.0 | 49.0 | 327.5 | 1469.2 | 2425.2 | 3372.7 |
| findbugs | 1486 | 0.0 | 0.0 | 2.0 | 8.0 | 28.0 | 77.0 | 170.5 | 307.5 | 705.3 |
| findsecuritybugs | 91 | 2.8 | 3.5 | 6.0 | 12.5 | 20.0 | 52.5 | 90.0 | 139.0 | 294.7 |
| jlint | 3 | 40.5 | 114.6 | 207.2 | 485.0 | 948.0 | 1028.0 | 1076.0 | 1092.0 | 1104.8 |
| pixy | 229 | 0.0 | 0.0 | 4.0 | 14.0 | 32.0 | 86.0 | 275.2 | 551.4 | 1443.0 |
| pmd | 1340 | 0.0 | 1.0 | 3.0 | 9.0 | 17.0 | 50.0 | 108.0 | 170.2 | 383.3 |
| rats | 19 | 0.0 | 0.0 | 0.0 | 0.0 | 470.0 | 1437.0 | 1578.6 | 1597.8 | 1603.6 |
| smatch | 483 | 0.0 | 0.0 | 0.2 | 8.0 | 22.0 | 90.0 | 326.8 | 798.9 | 2029.1 |
| splint | 681 | 0.0 | 0.0 | 2.0 | 9.0 | 22.0 | 85.0 | 343.0 | 1047.0 | 4479.2 |
| uno | 19 | 0.0 | 0.0 | 3.2 | 173.5 | 639.0 | 1196.5 | 1703.0 | 1960.0 | 3752.8 |
| wap | 338 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 32.5 | 180.5 | 641.7 | 9333.0 |

Tabela 5.28: percentis da métrica loc

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|------------------|---------|------|------|------|-------|-------|-------|-------|-------|--------|
| boon | 12 | 0.4 | 2.2 | 4.2 | 7.5 | 39.5 | 64.0 | 135.7 | 149.3 | 155.5 |
| clang | 579 | 0.0 | 0.0 | 0.0 | 1.0 | 3.0 | 19.0 | 44.2 | 69.2 | 123.9 |
| closure | 35 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 4.0 | 7.8 | 17.3 |
| cppcheck | 338 | 0.0 | 0.0 | 0.0 | 4.0 | 25.5 | 77.0 | 202.9 | 366.2 | 739.9 |
| cqual | 78 | 0.0 | 0.0 | 1.0 | 8.0 | 14.0 | 77.0 | 375.1 | 616.3 | 1055.5 |
| findbugs | 1486 | 0.0 | 0.0 | 1.0 | 3.0 | 11.0 | 32.0 | 67.5 | 104.0 | 247.3 |
| findsecuritybugs | 91 | 2.8 | 3.0 | 4.0 | 8.5 | 14.0 | 29.5 | 52.0 | 63.5 | 85.9 |
| jlint | 3 | 19.6 | 42.2 | 70.4 | 155.0 | 296.0 | 311.0 | 320.0 | 323.0 | 325.4 |
| pixy | 229 | 0.0 | 0.0 | 2.6 | 6.0 | 13.0 | 39.0 | 88.2 | 157.8 | 232.2 |
| pmd | 1340 | 0.0 | 1.0 | 3.0 | 3.0 | 7.0 | 18.0 | 36.0 | 55.0 | 108.6 |
| rats | 19 | 0.0 | 0.0 | 0.0 | 0.0 | 131.0 | 587.0 | 720.0 | 720.0 | 720.0 |
| smatch | 483 | 0.0 | 0.0 | 0.2 | 6.0 | 15.0 | 28.0 | 60.4 | 107.8 | 196.7 |
| splint | 681 | 0.0 | 0.0 | 1.0 | 6.0 | 13.0 | 28.0 | 84.0 | 273.0 | 777.2 |
| uno | 19 | 0.0 | 0.0 | 0.8 | 43.5 | 107.0 | 246.0 | 510.4 | 524.0 | 524.0 |
| wap | 338 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 11.0 | 86.6 | 404.4 | 2420.6 |

Tabela 5.29: percentis da métrica mmloc

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|------------------|---------|-----|-----|-----|-----|------|------|------|------|-------|
| boon | 12 | 0.0 | 0.0 | 0.0 | 0.8 | 2.5 | 5.8 | 8.9 | 22.9 | 36.6 |
| clang | 579 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 4.0 | 5.0 | 12.2 |
| closure | 35 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 |
| cppcheck | 338 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 3.0 | 6.3 | 9.1 | 20.5 |
| cqual | 78 | 0.0 | 0.0 | 0.0 | 0.0 | 2.5 | 15.8 | 70.3 | 90.1 | 268.4 |
| findbugs | 1486 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 3.0 | 7.0 | 11.0 | 25.2 |
| findsecuritybugs | 91 | 0.0 | 0.0 | 0.0 | 1.0 | 2.0 | 3.0 | 6.0 | 8.0 | 10.2 |
| jlint | 3 | 0.2 | 1.2 | 2.4 | 6.0 | 12.0 | 13.5 | 14.4 | 14.7 | 14.9 |
| pixy | 229 | 0.0 | 0.0 | 0.0 | 1.0 | 2.0 | 3.0 | 6.0 | 8.0 | 26.2 |
| pmd | 1340 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 2.0 | 4.1 | 7.0 | 17.0 |
| rats | 19 | 0.0 | 0.0 | 0.0 | 0.0 | 24.0 | 41.5 | 43.6 | 46.4 | 49.3 |
| smatch | 483 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 4.0 | 9.0 | 13.9 | 92.0 |
| splint | 681 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 4.0 | 9.0 | 20.0 | 98.6 |
| uno | 19 | 0.0 | 0.0 | 0.8 | 4.5 | 16.0 | 45.0 | 64.0 | 77.6 | 117.9 |
| wap | 338 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 331.7 |

Tabela 5.30: percentis da métrica noa

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|------------------|---------|-----|-----|-----|-----|-----|-----|-----|-----|------|
| boon | 12 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| clang | 579 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.4 |
| closure | 35 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.6 |
| cppcheck | 338 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 5.3 |
| cqual | 78 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| findbugs | 1486 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 2.0 | 7.0 |
| findsecuritybugs | 91 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 3.0 |
| jlint | 3 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| pixy | 229 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 5.7 |
| pmd | 1340 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 3.0 | 11.0 |
| rats | 19 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| smatch | 483 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| splint | 681 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| uno | 19 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| wap | 338 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 3.0 |

Tabela 5.31: percentis da métrica noc

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|------------------|---------|-----|-----|-----|-----|------|------|------|-------|-------|
| boon | 12 | 0.1 | 0.6 | 1.0 | 1.8 | 2.0 | 5.8 | 8.9 | 44.1 | 78.4 |
| clang | 579 | 0.0 | 0.0 | 0.0 | 1.0 | 2.0 | 3.0 | 6.0 | 10.0 | 20.4 |
| closure | 35 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 1.3 | 2.0 |
| cppcheck | 338 | 0.0 | 0.0 | 0.0 | 1.0 | 6.0 | 17.0 | 36.3 | 92.9 | 177.5 |
| cqual | 78 | 0.0 | 0.0 | 1.0 | 1.0 | 4.5 | 20.8 | 77.3 | 105.8 | 256.0 |
| findbugs | 1486 | 0.0 | 0.0 | 1.0 | 2.0 | 4.0 | 7.0 | 15.0 | 23.0 | 59.5 |
| findsecuritybugs | 91 | 1.0 | 1.0 | 1.0 | 2.0 | 2.0 | 4.0 | 7.0 | 13.0 | 29.8 |
| jlint | 3 | 2.2 | 2.9 | 3.8 | 6.5 | 11.0 | 16.0 | 19.0 | 20.0 | 20.8 |
| pixy | 229 | 0.0 | 0.0 | 1.0 | 2.0 | 4.0 | 8.0 | 18.2 | 29.0 | 59.2 |
| pmd | 1340 | 0.0 | 1.0 | 1.0 | 2.0 | 3.0 | 6.0 | 12.0 | 20.0 | 46.6 |
| rats | 19 | 0.0 | 0.0 | 0.0 | 0.0 | 11.0 | 40.5 | 49.4 | 51.1 | 51.8 |
| smatch | 483 | 0.0 | 0.0 | 0.2 | 1.0 | 2.0 | 7.0 | 21.8 | 43.0 | 99.5 |
| splint | 681 | 0.0 | 0.0 | 1.0 | 1.0 | 3.0 | 8.0 | 20.0 | 43.0 | 178.4 |
| uno | 19 | 0.0 | 0.0 | 0.8 | 6.5 | 17.0 | 40.5 | 53.8 | 70.5 | 138.9 |
| wap | 338 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 3.0 | 5.0 | 17.1 | 86.0 |

Tabela 5.32: percentis da métrica nom

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|------------------|---------|-----|-----|-----|-----|------|------|------|------|-------|
| boon | 12 | 0.0 | 0.0 | 0.0 | 0.8 | 2.5 | 5.8 | 8.9 | 22.9 | 36.6 |
| clang | 579 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 2.0 | 4.0 | 7.4 |
| closure | 35 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 |
| cppcheck | 338 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 4.0 | 7.0 | 15.3 |
| cqual | 78 | 0.0 | 0.0 | 0.0 | 0.0 | 2.5 | 15.8 | 70.3 | 90.1 | 268.4 |
| findbugs | 1486 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 4.0 | 12.0 |
| findsecuritybugs | 91 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 4.1 |
| jlint | 3 | 0.2 | 1.2 | 2.4 | 6.0 | 12.0 | 13.5 | 14.4 | 14.7 | 14.9 |
| pixy | 229 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.6 | 8.3 |
| pmd | 1340 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 2.0 | 7.0 |
| rats | 19 | 0.0 | 0.0 | 0.0 | 0.0 | 24.0 | 41.5 | 43.6 | 46.4 | 49.3 |
| smatch | 483 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 4.0 | 9.0 | 13.9 | 92.0 |
| splint | 681 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 4.0 | 9.0 | 20.0 | 98.6 |
| uno | 19 | 0.0 | 0.0 | 0.8 | 4.5 | 16.0 | 45.0 | 64.0 | 77.6 | 117.9 |
| wap | 338 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 323.7 |

Tabela 5.33: percentis da métrica npa

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|------------------|---------|-----|-----|-----|-----|------|------|------|-------|-------|
| boon | 12 | 0.1 | 0.6 | 1.0 | 1.8 | 2.0 | 5.8 | 8.9 | 44.1 | 78.4 |
| clang | 579 | 0.0 | 0.0 | 0.0 | 0.5 | 2.0 | 3.0 | 6.0 | 9.0 | 19.2 |
| closure | 35 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 1.3 | 2.0 |
| cppcheck | 338 | 0.0 | 0.0 | 0.0 | 1.0 | 2.0 | 8.0 | 15.0 | 22.1 | 50.6 |
| cqual | 78 | 0.0 | 0.0 | 1.0 | 1.0 | 4.5 | 20.8 | 77.3 | 105.8 | 256.0 |
| findbugs | 1486 | 0.0 | 0.0 | 1.0 | 2.0 | 3.0 | 6.0 | 13.0 | 18.0 | 54.0 |
| findsecuritybugs | 91 | 1.0 | 1.0 | 1.0 | 2.0 | 2.0 | 3.0 | 5.0 | 6.0 | 21.2 |
| jlint | 3 | 2.2 | 2.9 | 3.8 | 6.5 | 11.0 | 16.0 | 19.0 | 20.0 | 20.8 |
| pixy | 229 | 0.0 | 0.0 | 1.0 | 2.0 | 3.0 | 6.0 | 13.2 | 28.6 | 46.4 |
| pmd | 1340 | 0.0 | 1.0 | 1.0 | 1.0 | 3.0 | 5.0 | 10.0 | 16.0 | 45.6 |
| rats | 19 | 0.0 | 0.0 | 0.0 | 0.0 | 11.0 | 40.5 | 49.4 | 51.1 | 51.8 |
| smatch | 483 | 0.0 | 0.0 | 0.2 | 1.0 | 2.0 | 7.0 | 21.8 | 43.0 | 99.5 |
| splint | 681 | 0.0 | 0.0 | 1.0 | 1.0 | 3.0 | 8.0 | 20.0 | 43.0 | 178.4 |
| uno | 19 | 0.0 | 0.0 | 0.8 | 6.5 | 17.0 | 40.5 | 53.8 | 70.5 | 138.9 |
| wap | 338 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 3.0 | 5.0 | 17.0 | 84.0 |

Tabela 5.34: percentis da métrica npm

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|------------------|---------|------|------|------|------|------|-------|-------|-------|--------|
| boon | 12 | 0.1 | 0.6 | 1.0 | 2.5 | 8.0 | 21.8 | 42.0 | 355.7 | 661.6 |
| clang | 579 | 0.0 | 0.0 | 0.0 | 1.0 | 3.0 | 9.0 | 16.0 | 26.1 | 66.7 |
| closure | 35 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 2.0 | 2.0 | 3.3 |
| cppcheck | 338 | 0.0 | 0.0 | 0.0 | 2.0 | 15.0 | 67.0 | 180.5 | 375.5 | 707.8 |
| cqual | 78 | 0.0 | 0.0 | 1.0 | 4.2 | 14.0 | 122.5 | 629.3 | 745.4 | 1043.4 |
| findbugs | 1486 | 0.0 | 0.0 | 1.0 | 3.0 | 8.0 | 24.0 | 52.0 | 88.8 | 202.8 |
| findsecuritybugs | 91 | 1.0 | 1.5 | 3.0 | 5.0 | 6.0 | 10.0 | 17.0 | 38.0 | 105.3 |
| jlint | 3 | 14.7 | 17.7 | 21.4 | 32.5 | 51.0 | 83.5 | 103.0 | 109.5 | 114.7 |
| pixy | 229 | 0.0 | 0.0 | 2.0 | 5.0 | 9.0 | 22.0 | 71.4 | 100.0 | 284.9 |
| pmd | 1340 | 0.0 | 1.0 | 2.0 | 3.0 | 7.0 | 18.0 | 39.0 | 68.0 | 139.0 |
| rats | 19 | 0.0 | 0.0 | 0.0 | 0.0 | 65.0 | 399.0 | 542.6 | 551.0 | 565.4 |
| smatch | 483 | 0.0 | 0.0 | 0.2 | 3.0 | 7.0 | 41.5 | 121.6 | 281.9 | 763.5 |
| splint | 681 | 0.0 | 0.0 | 1.0 | 3.0 | 7.0 | 28.0 | 104.0 | 267.0 | 1151.6 |
| uno | 19 | 0.0 | 0.0 | 0.8 | 27.5 | 93.0 | 304.0 | 522.2 | 742.5 | 1401.3 |
| wap | 338 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 3.0 | 16.3 | 100.0 | 185.1 |

Tabela 5.35: percentis da métrica rfc

| | classes | 1% | 5% | 10% | 25% | 50% | 75% | 90% | 95% | 99% |
|------------------|---------|------|------|-------|-------|-------|--------|--------|--------|---------|
| boon | 12 | 0.0 | 0.0 | 0.0 | 0.0 | 1.5 | 5.2 | 14.1 | 19.0 | 23.0 |
| clang | 579 | 0.0 | 0.0 | 0.0 | 24.0 | 115.0 | 226.0 | 460.8 | 723.0 | 1678.3 |
| closure | 35 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2.0 | 2.0 | 2.0 | 3.3 |
| cppcheck | 338 | 0.0 | 0.0 | 0.0 | 90.2 | 237.0 | 409.0 | 704.0 | 950.8 | 5293.0 |
| cqual | 78 | 0.0 | 0.0 | 0.0 | 1.0 | 46.0 | 140.8 | 558.9 | 816.0 | 7501.3 |
| findbugs | 1486 | 0.0 | 0.0 | 90.5 | 322.5 | 605.0 | 1243.5 | 2507.5 | 3923.5 | 10572.4 |
| findsecuritybugs | 91 | 0.0 | 0.0 | 0.0 | 3.0 | 9.0 | 15.0 | 22.0 | 30.0 | 70.5 |
| jlint | 3 | 0.0 | 0.2 | 0.4 | 1.0 | 2.0 | 2.5 | 2.8 | 2.9 | 3.0 |
| pixy | 229 | 0.0 | 0.0 | 40.0 | 82.0 | 185.0 | 351.0 | 533.6 | 788.4 | 1471.6 |
| pmd | 1340 | 0.0 | 51.0 | 124.0 | 266.0 | 411.0 | 836.0 | 1514.4 | 1976.2 | 4876.0 |
| rats | 19 | 0.0 | 0.0 | 0.0 | 0.0 | 8.0 | 155.0 | 296.4 | 306.6 | 310.9 |
| smatch | 483 | 0.0 | 0.0 | 2.6 | 75.5 | 113.0 | 123.0 | 320.4 | 541.6 | 2032.9 |
| splint | 681 | 0.0 | 0.0 | 0.0 | 311.0 | 322.0 | 939.0 | 1935.0 | 2900.0 | 8942.4 |
| uno | 19 | 0.0 | 0.0 | 2.4 | 5.5 | 13.0 | 23.0 | 32.4 | 46.2 | 76.4 |
| wap | 338 | 33.1 | 35.0 | 35.0 | 36.0 | 37.0 | 108.0 | 211.5 | 503.2 | 3064.2 |

Tabela 5.36: percentis da métrica sc

5.1.3 Gráficos das métricas para os softwares científicos

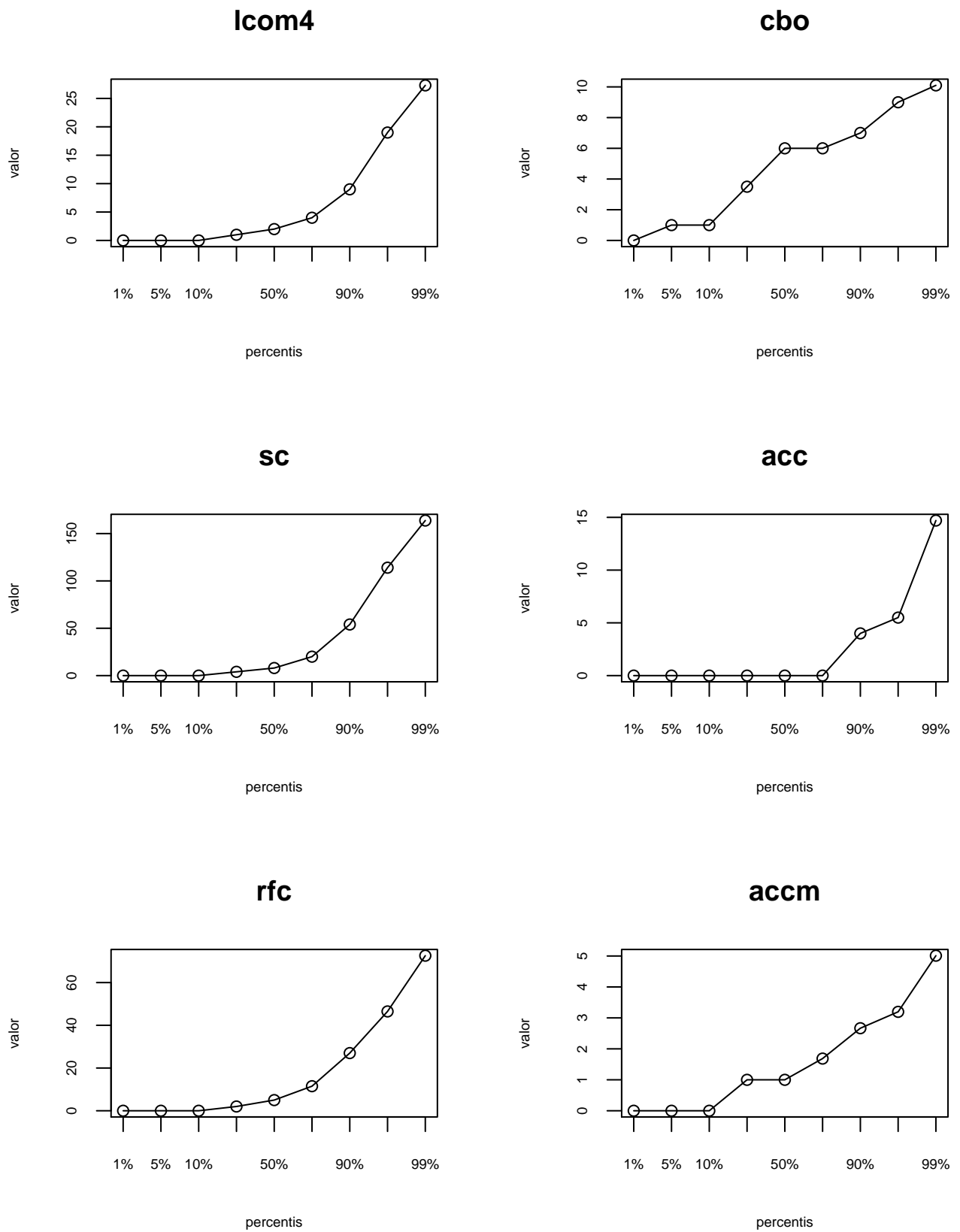


Figura 5.1: distribuição das métricas para a ferramenta accessanalysis

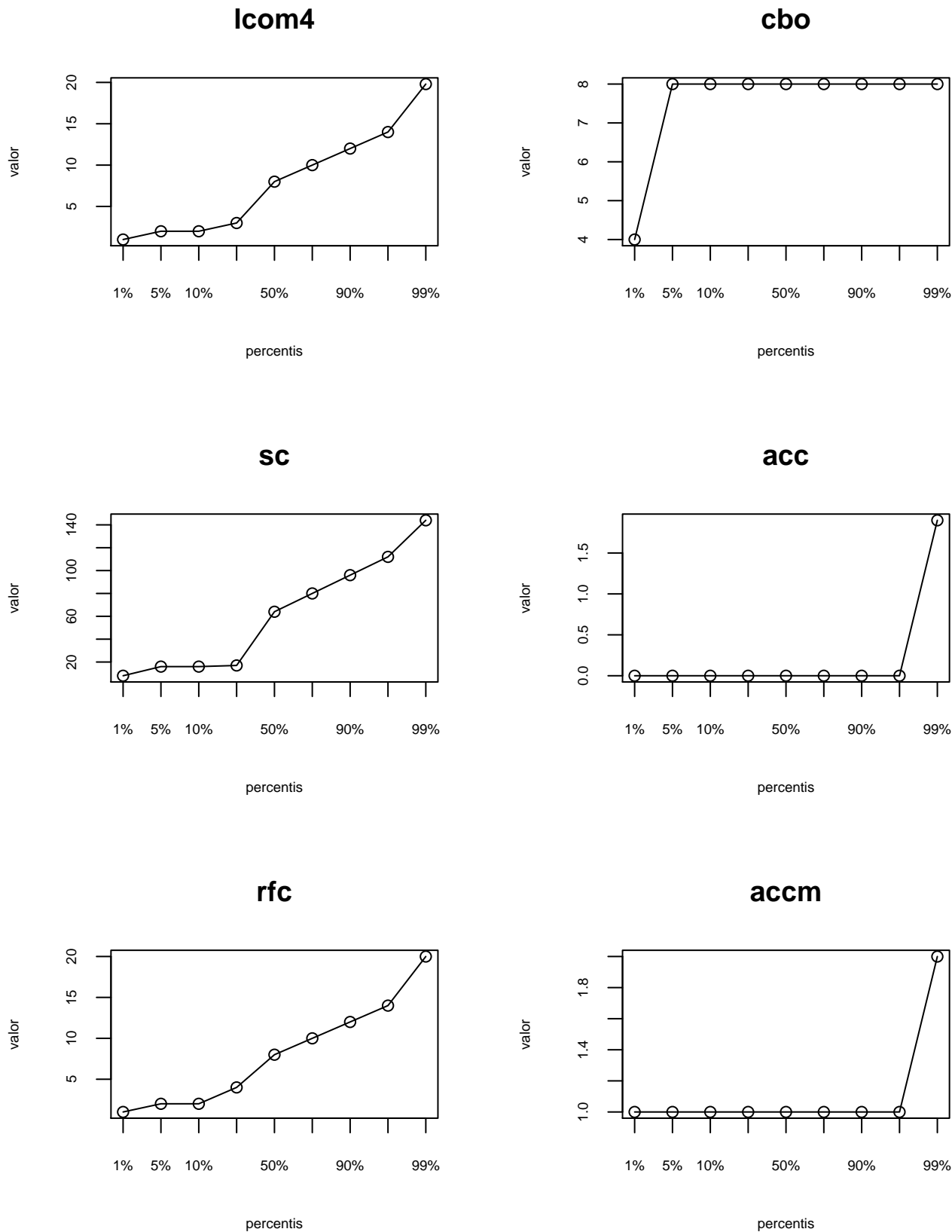


Figura 5.2: distribuição das métricas para a ferramenta bakar-ali

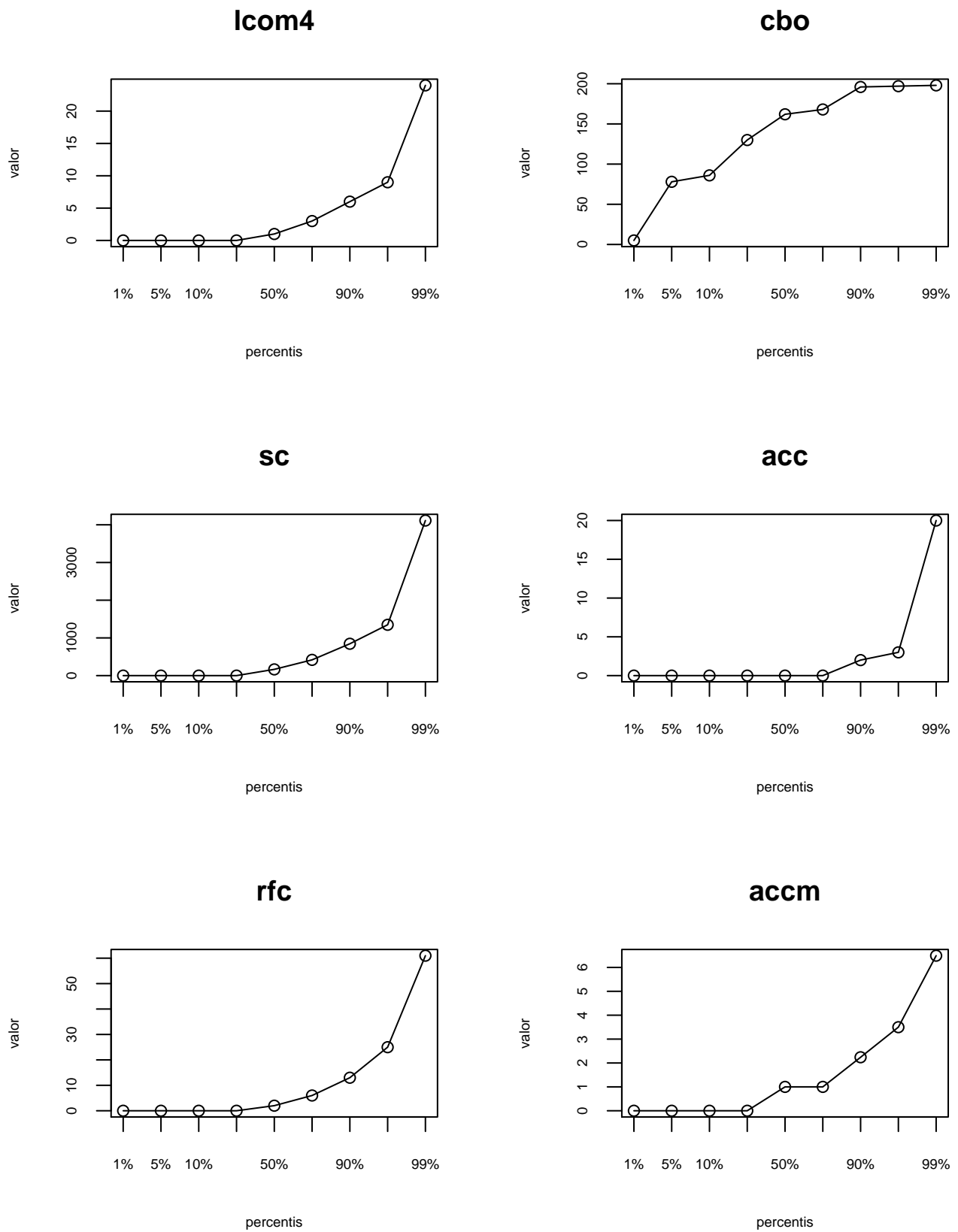


Figura 5.3: distribuição das métricas para a ferramenta error-prone

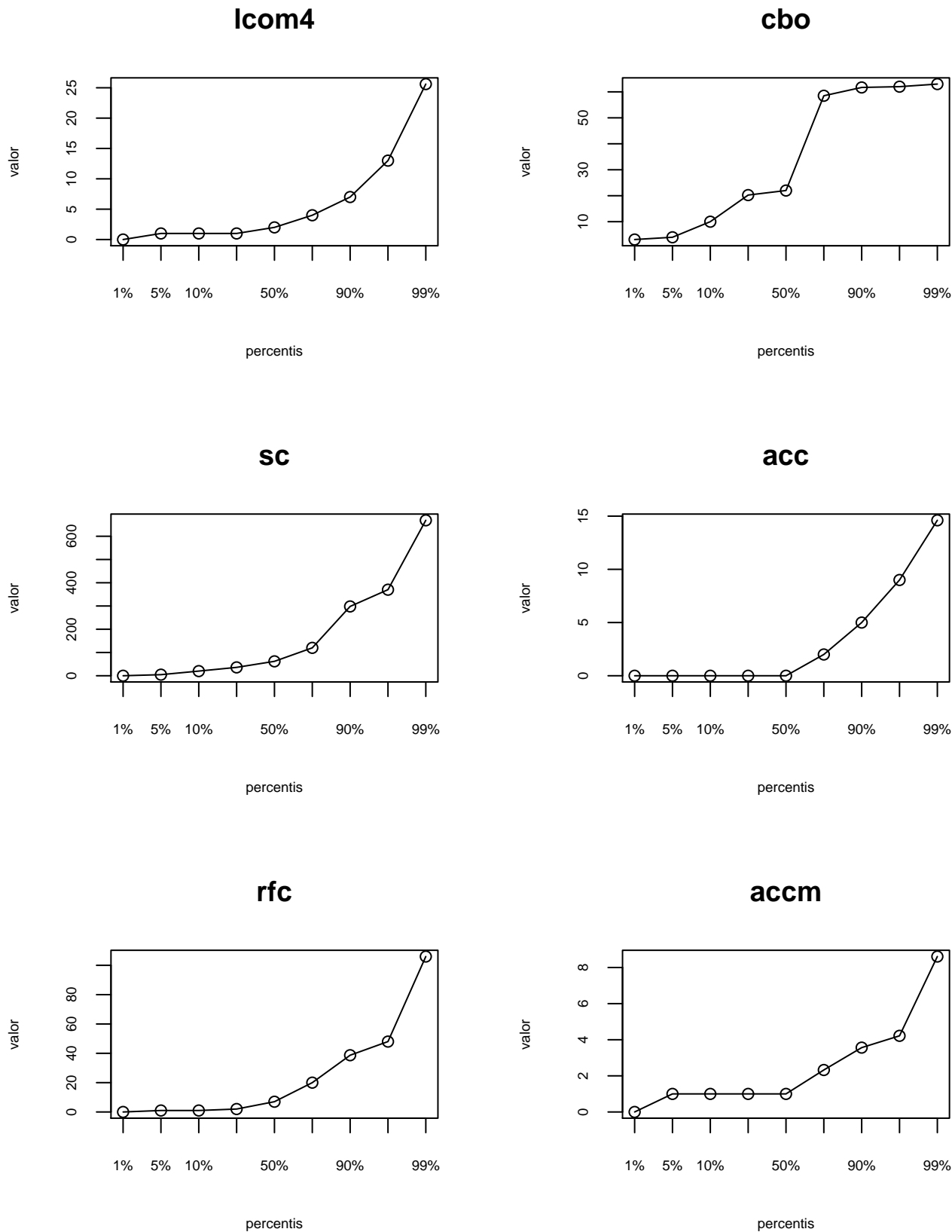


Figura 5.4: distribuição das métricas para a ferramenta indus

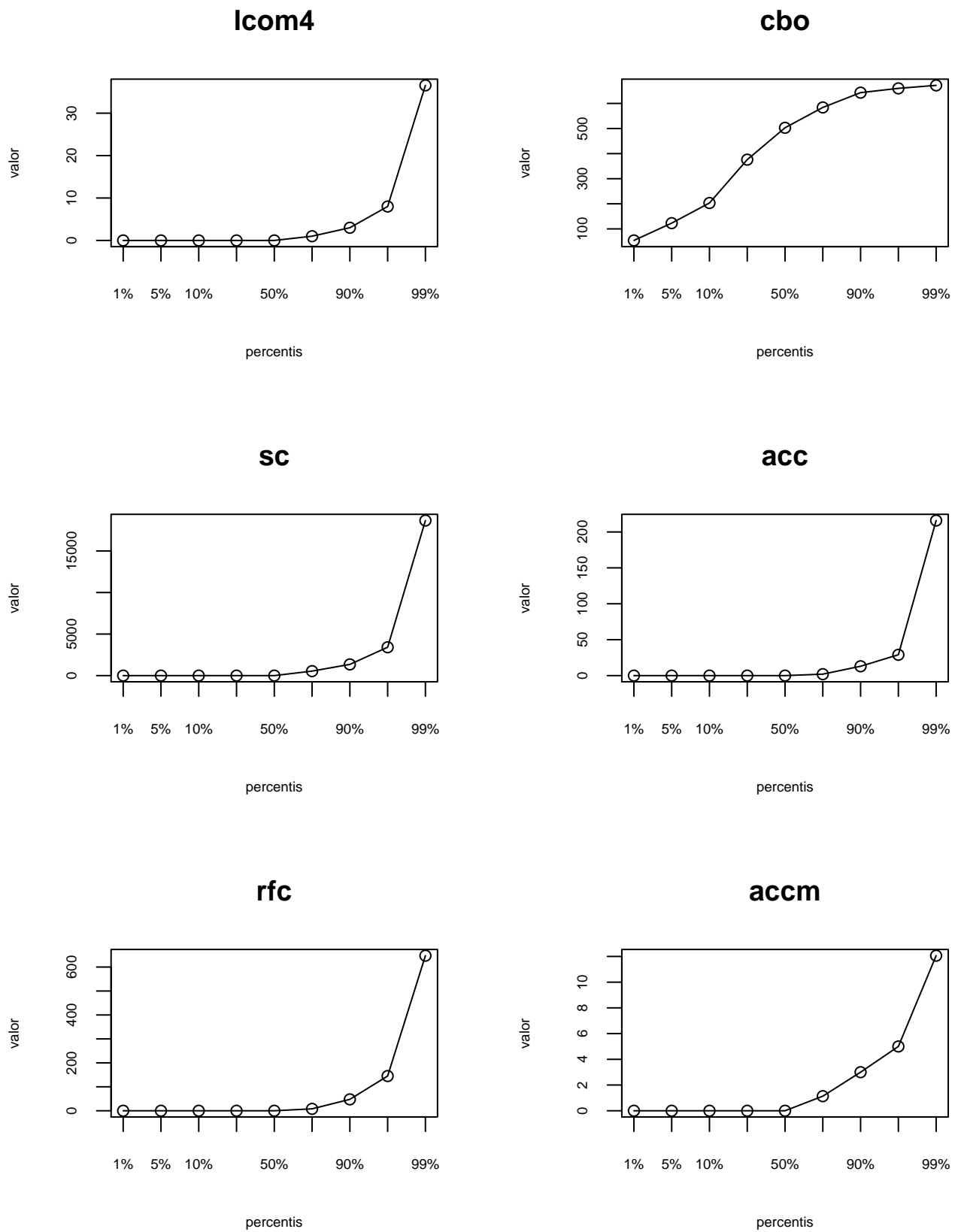


Figura 5.5: distribuição das métricas para a ferramenta inputtracer

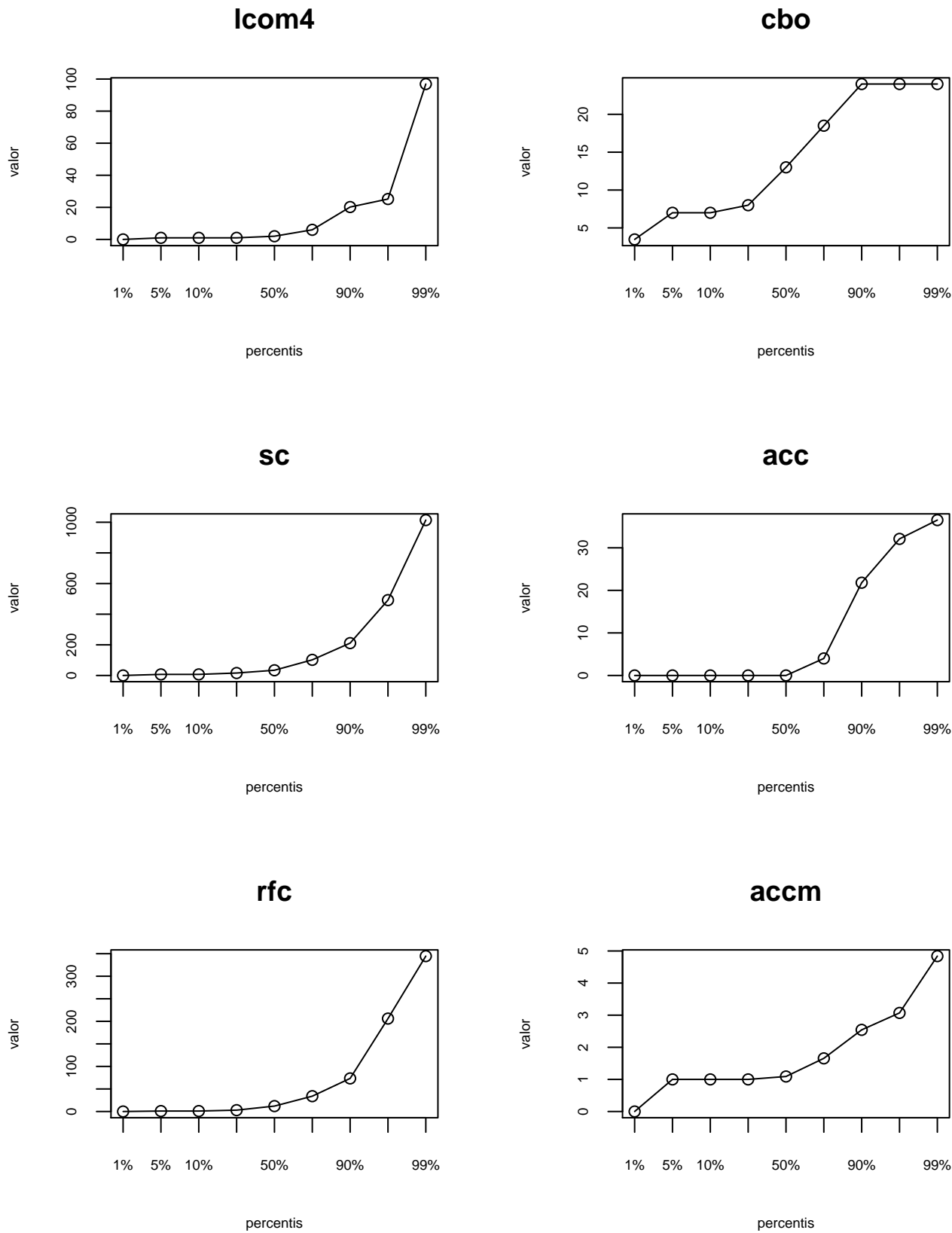


Figura 5.6: distribuição das métricas para a ferramenta jastadd

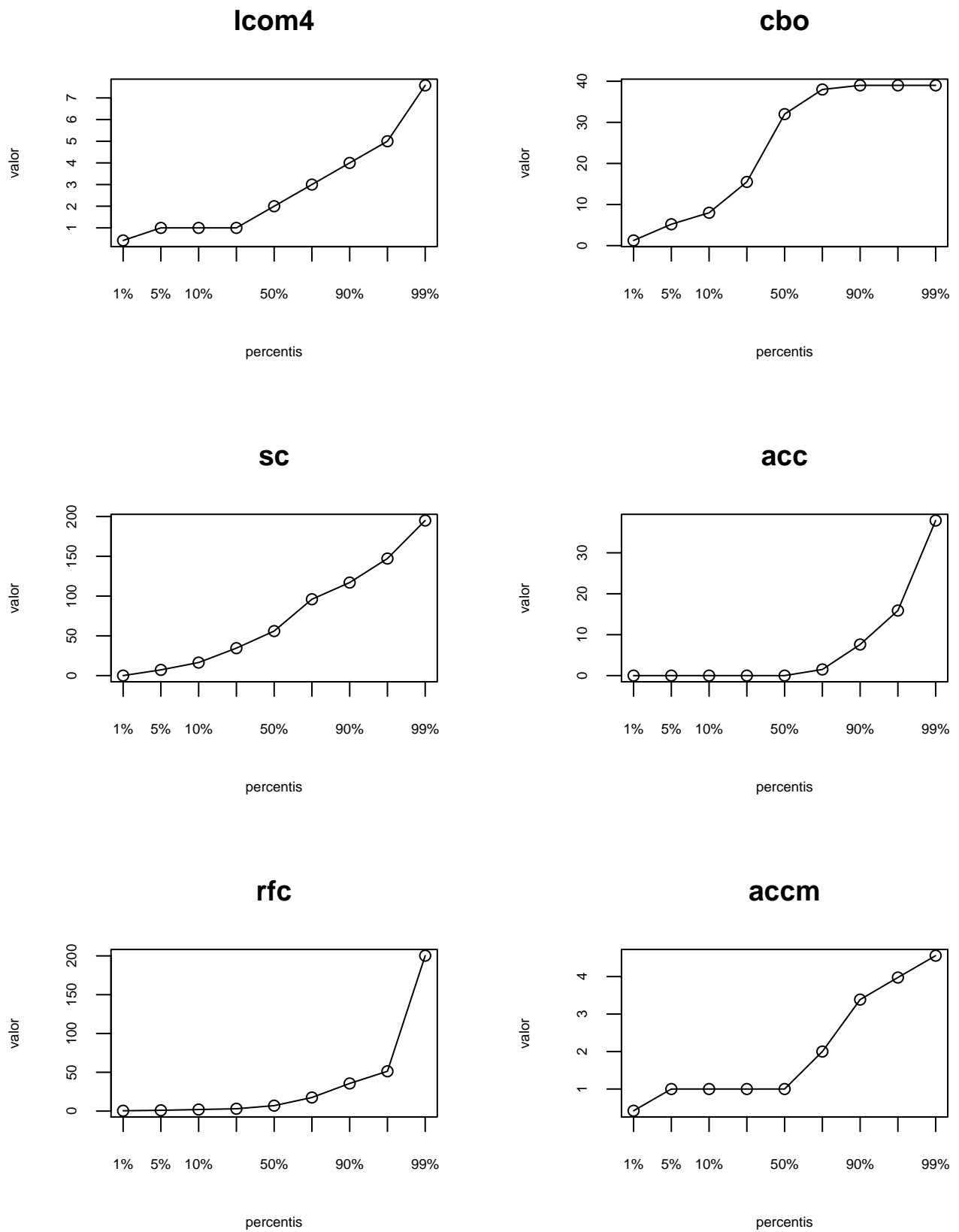


Figura 5.7: distribuição das métricas para a ferramenta source-meter

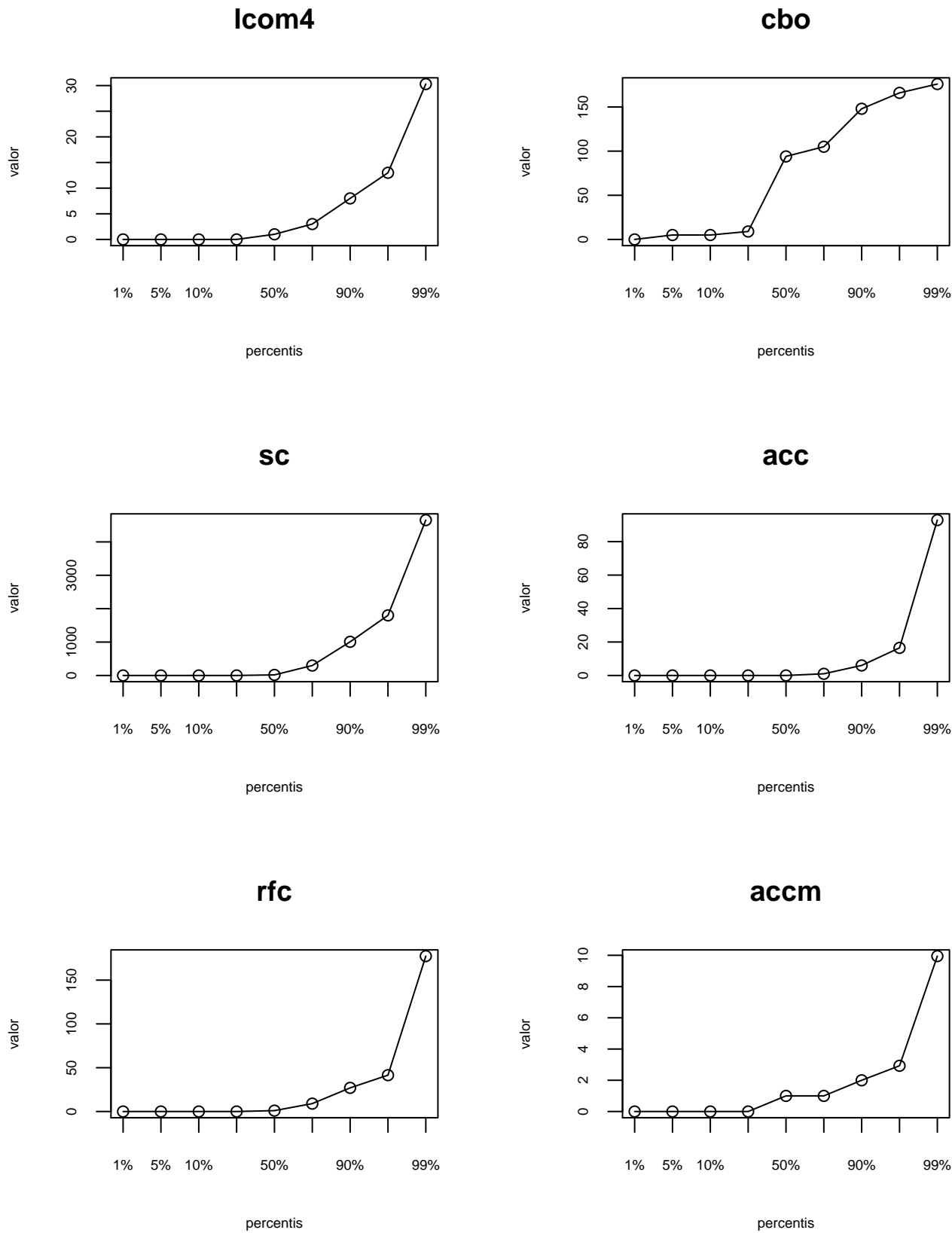


Figura 5.8: distribuição das métricas para a ferramenta srcml

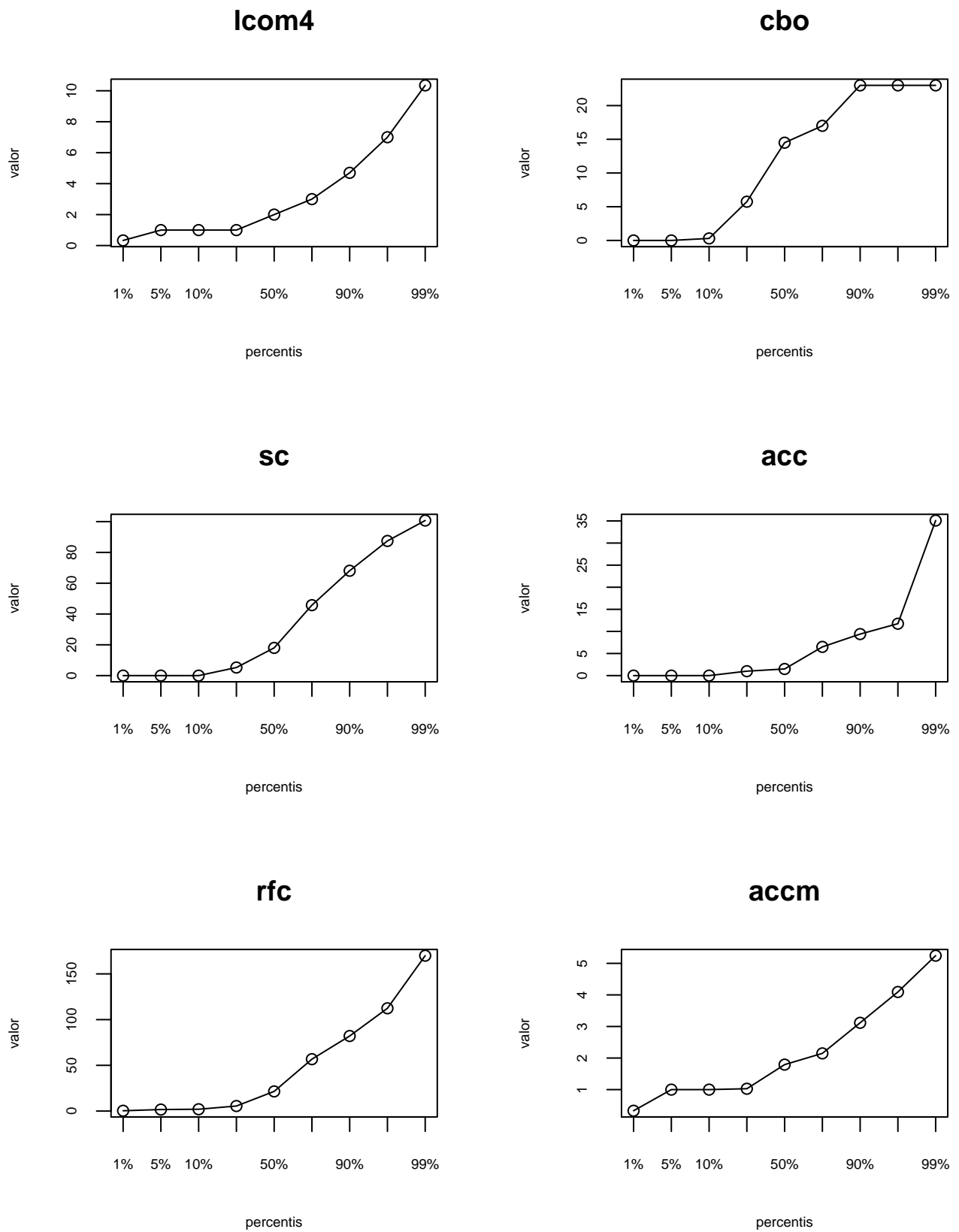


Figura 5.9: distribuição das métricas para a ferramenta tackle

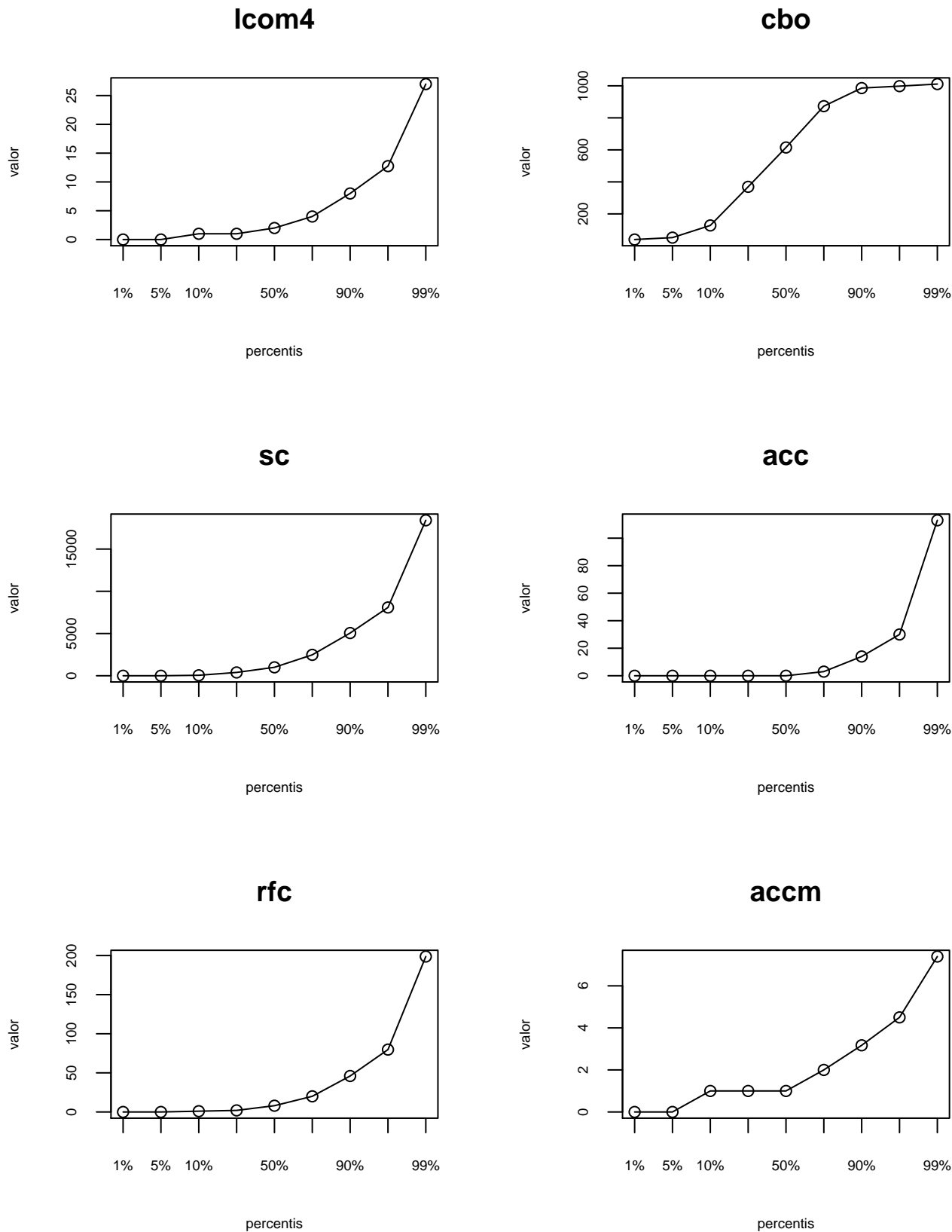


Figura 5.10: distribuição das métricas para a ferramenta wala

5.1.4 Gráficos das métricas para as ferramentas da indústria

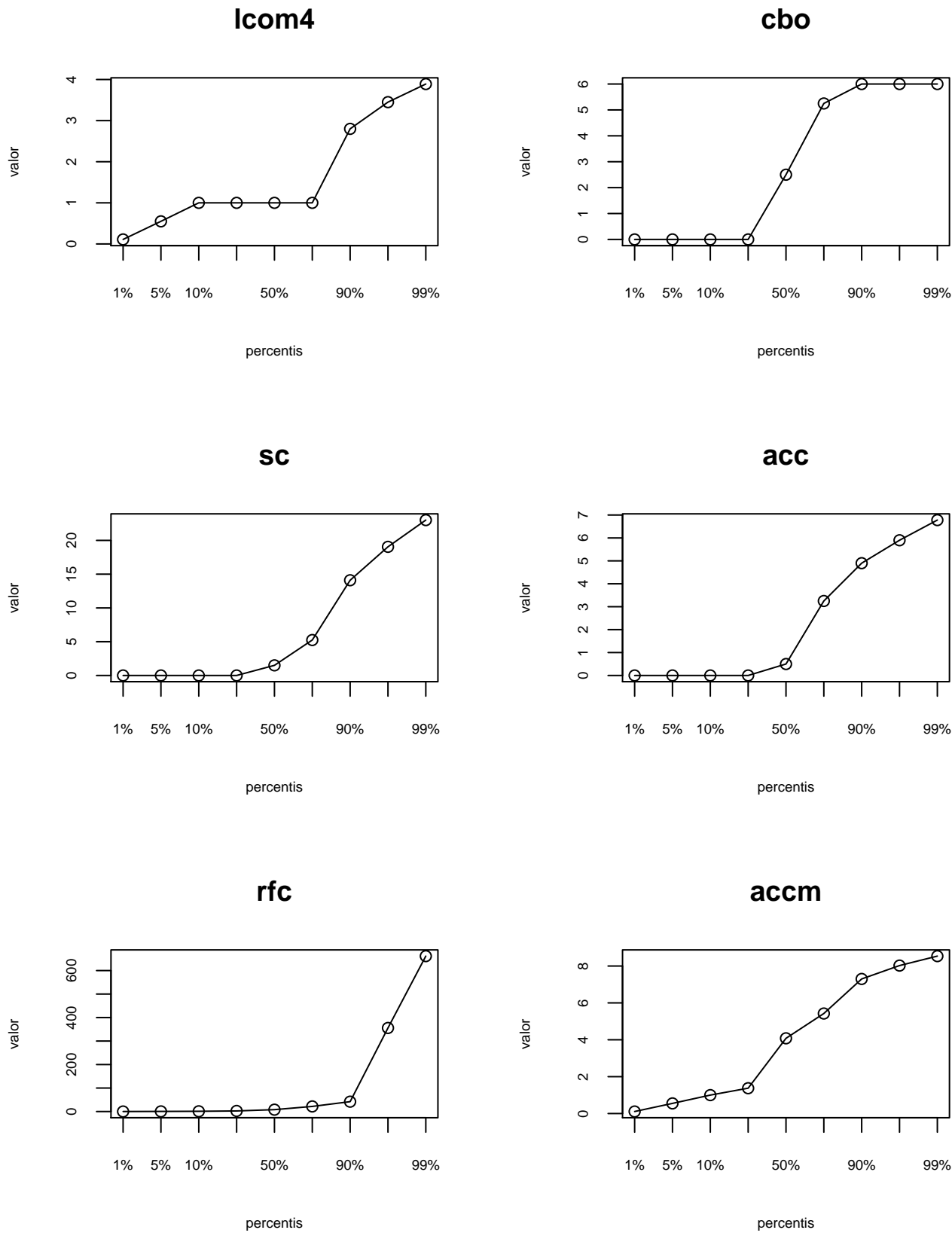


Figura 5.11: distribuição das métricas para a ferramenta boon

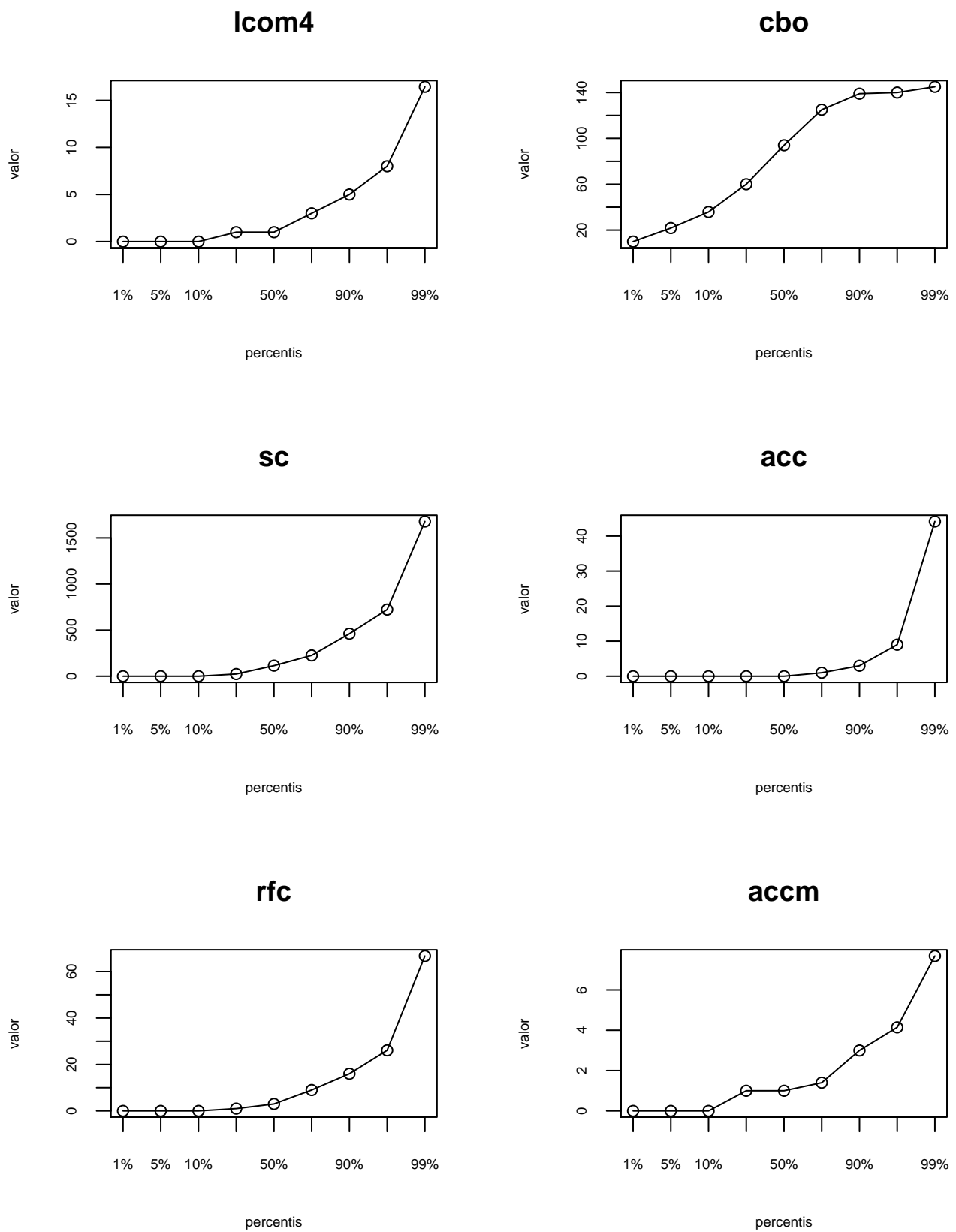


Figura 5.12: distribuição das métricas para a ferramenta clang

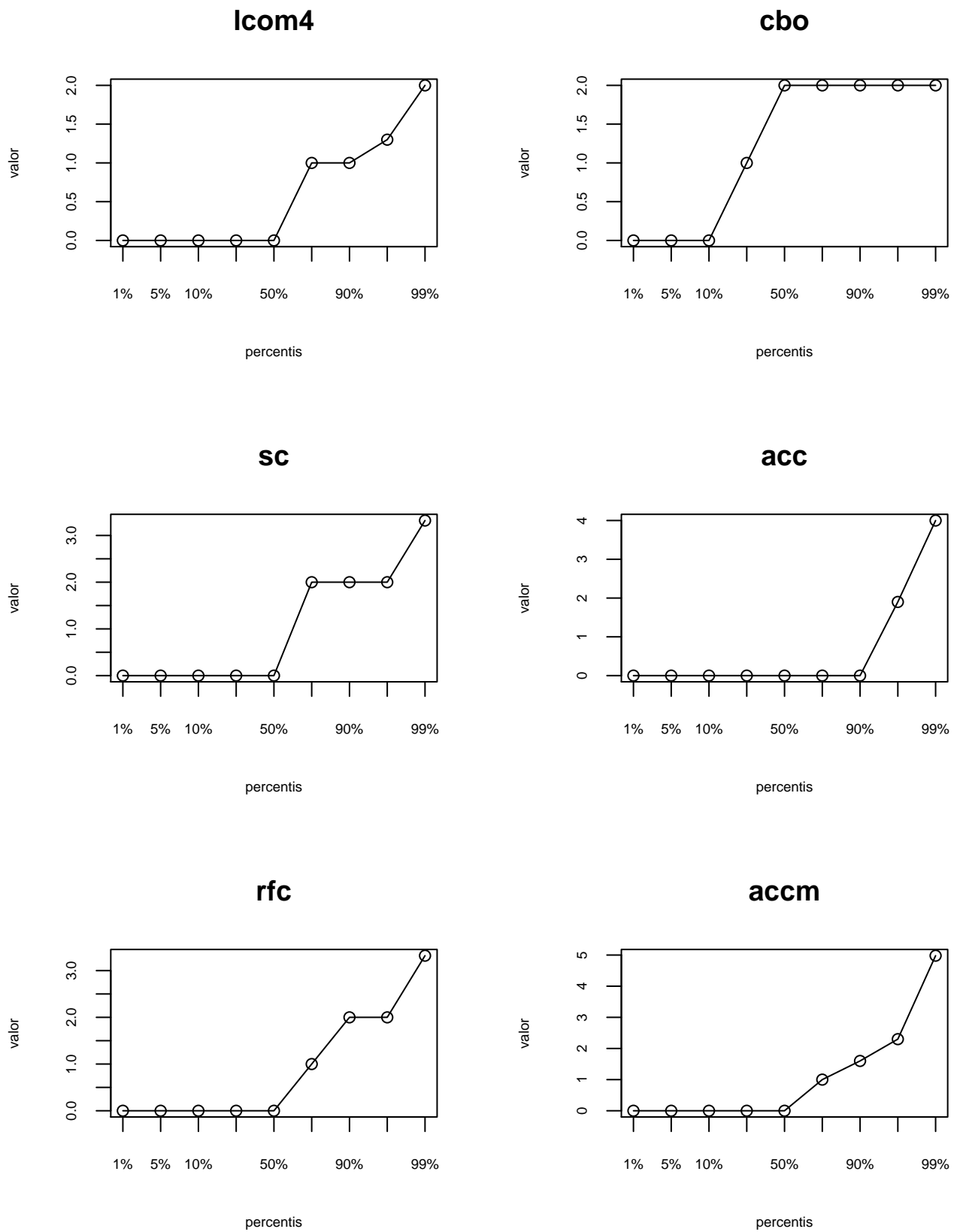


Figura 5.13: distribuição das métricas para a ferramenta closure-compiler

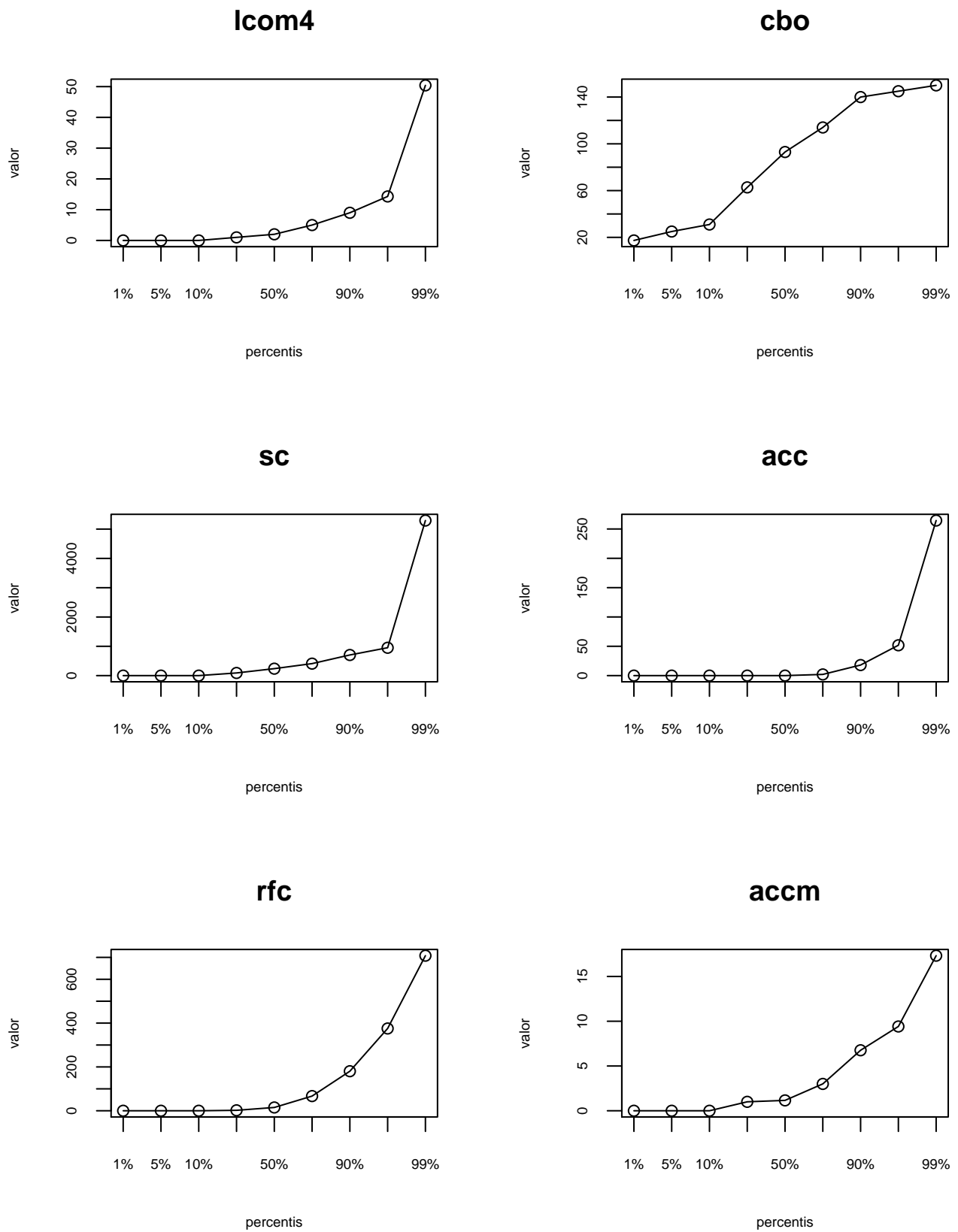


Figura 5.14: distribuição das métricas para a ferramenta cppcheck

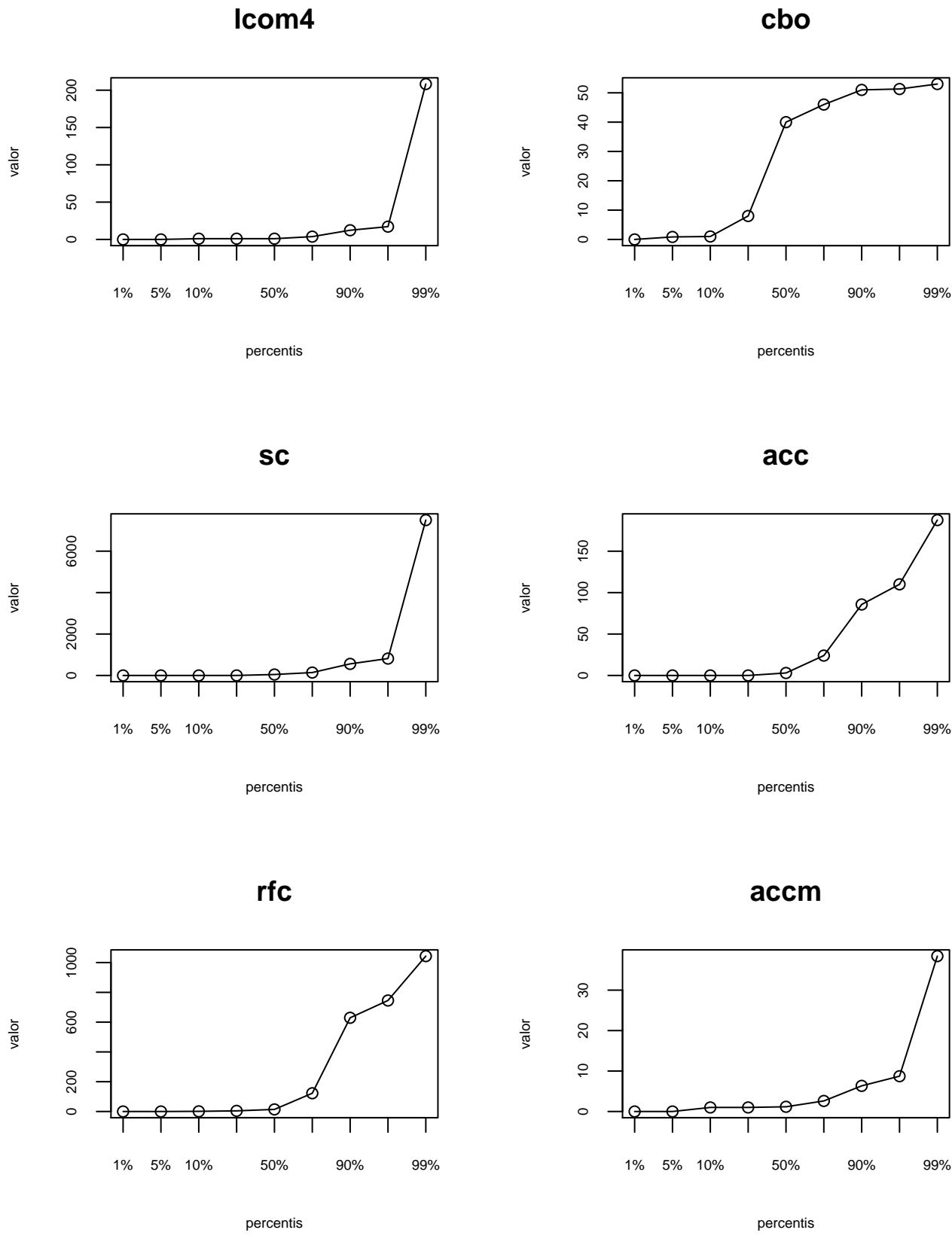


Figura 5.15: distribuição das métricas para a ferramenta equal

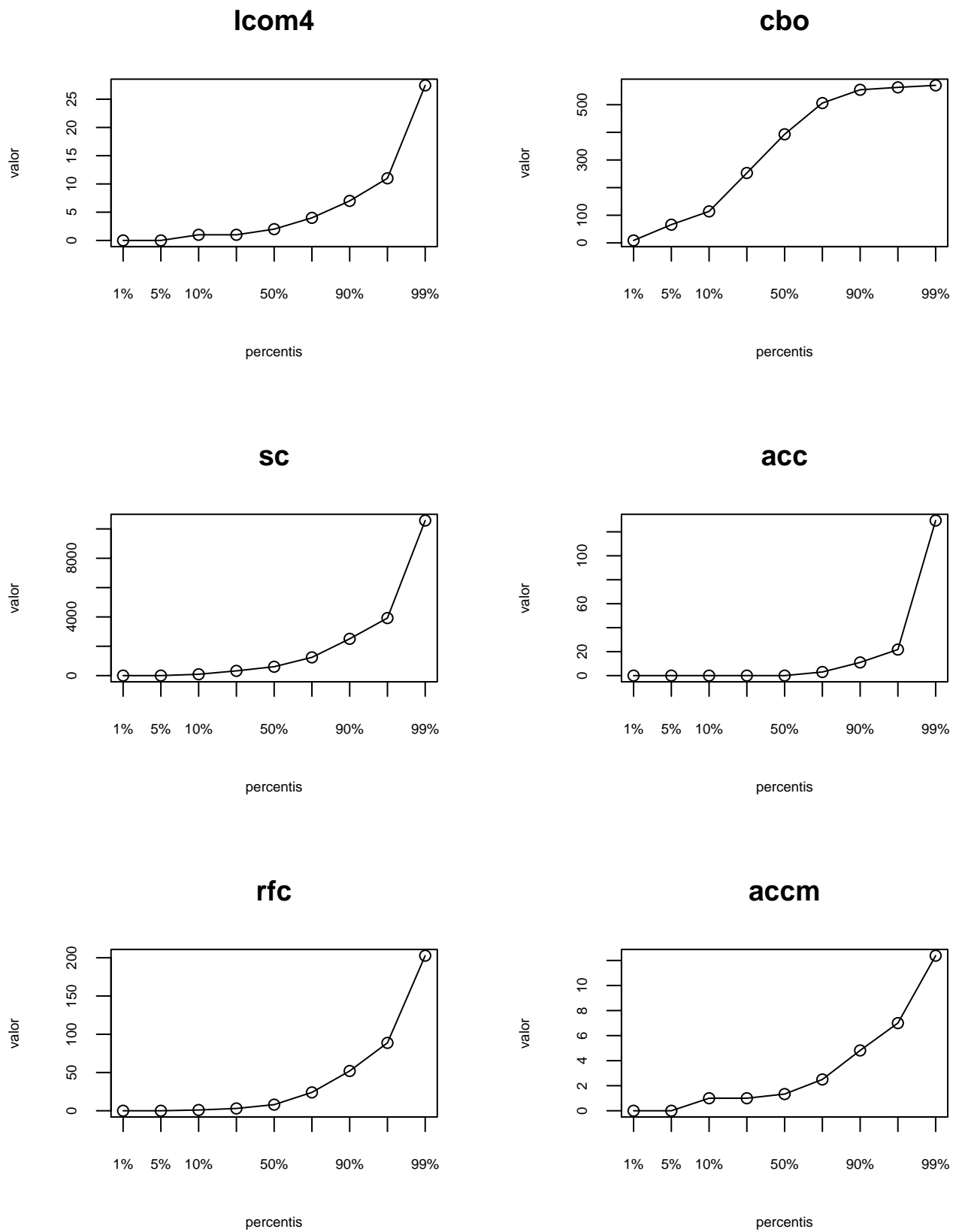


Figura 5.16: distribuição das métricas para a ferramenta findbugs

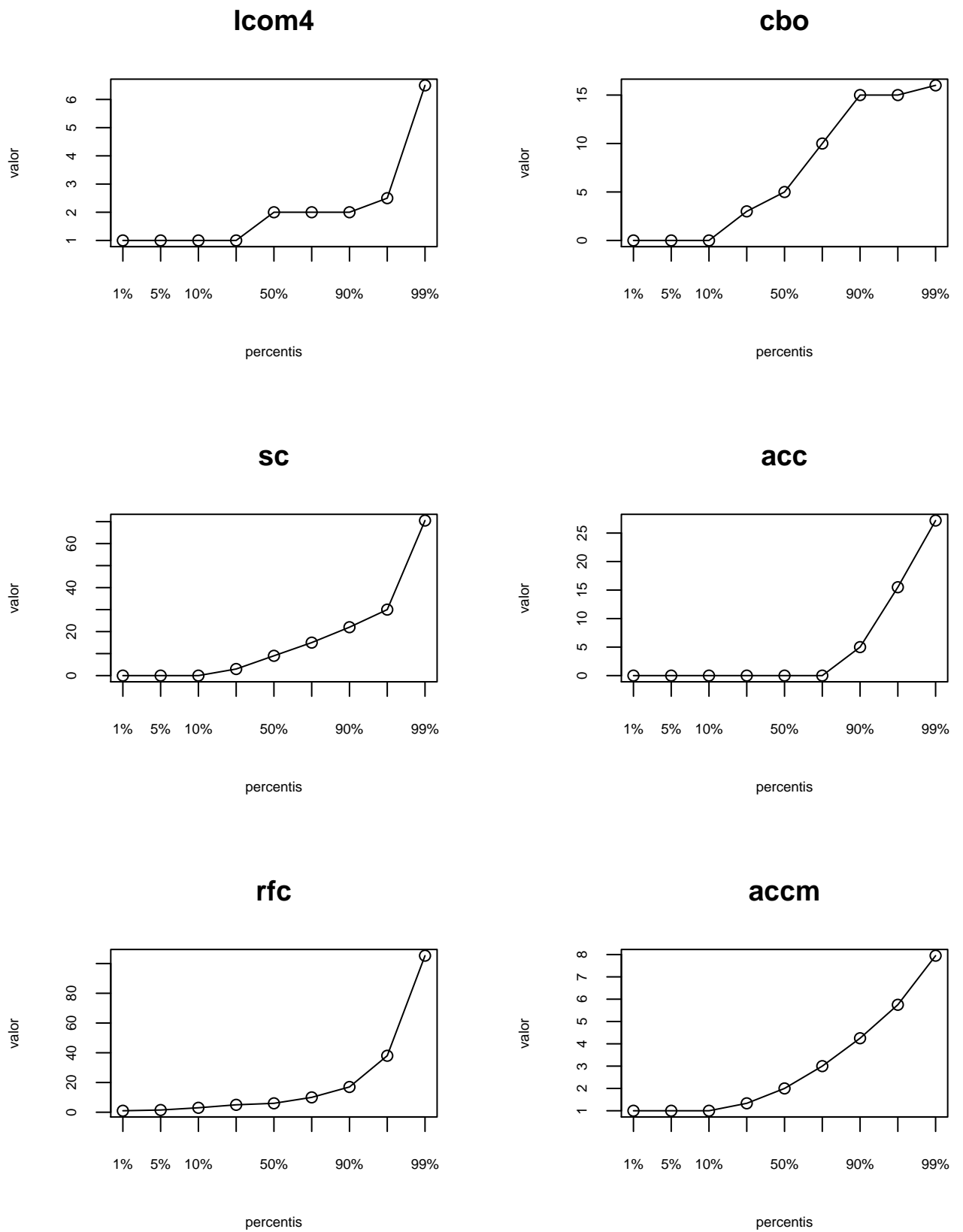


Figura 5.17: distribuição das métricas para a ferramenta findsecuritybugs

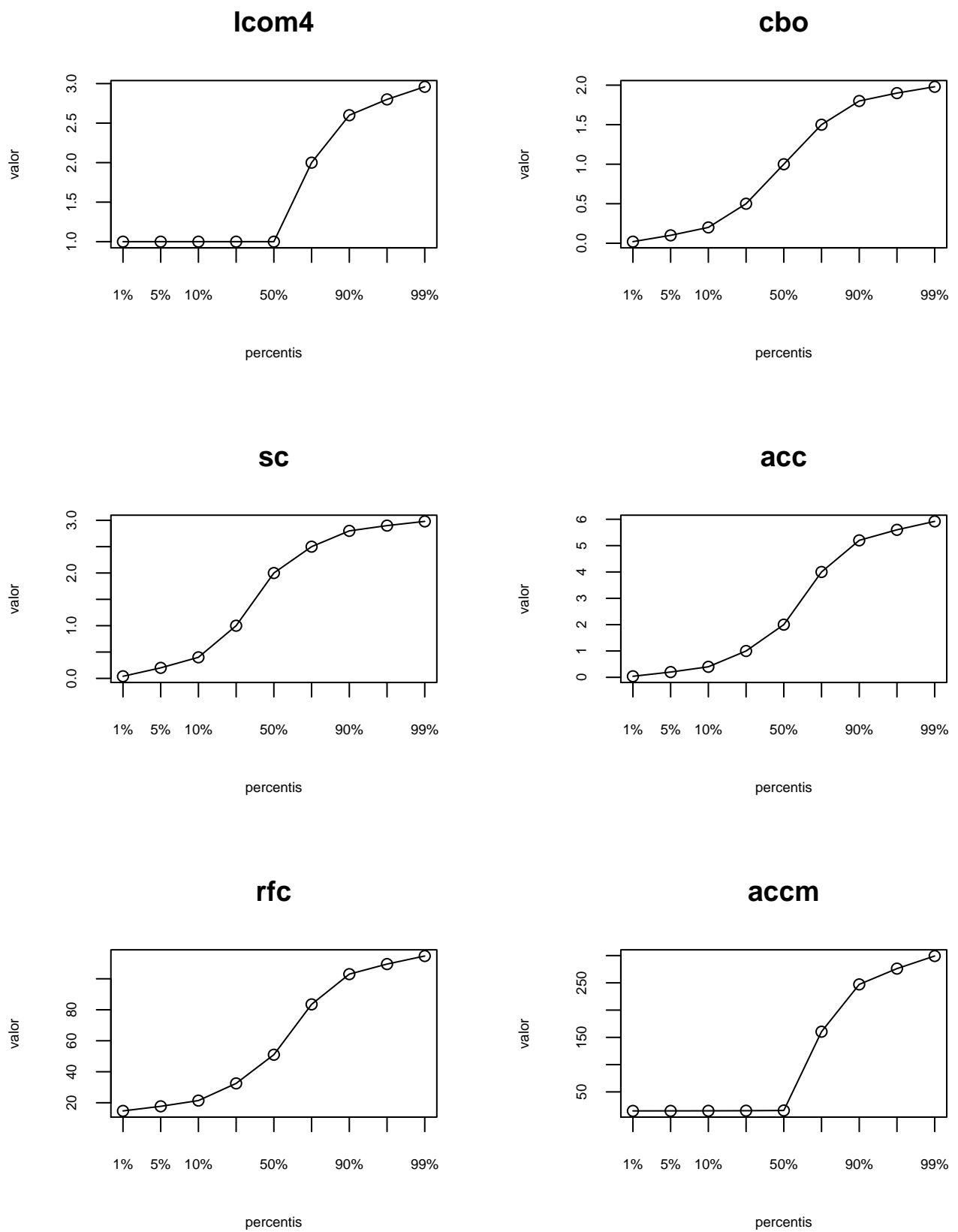


Figura 5.18: distribuição das métricas para a ferramenta jlint

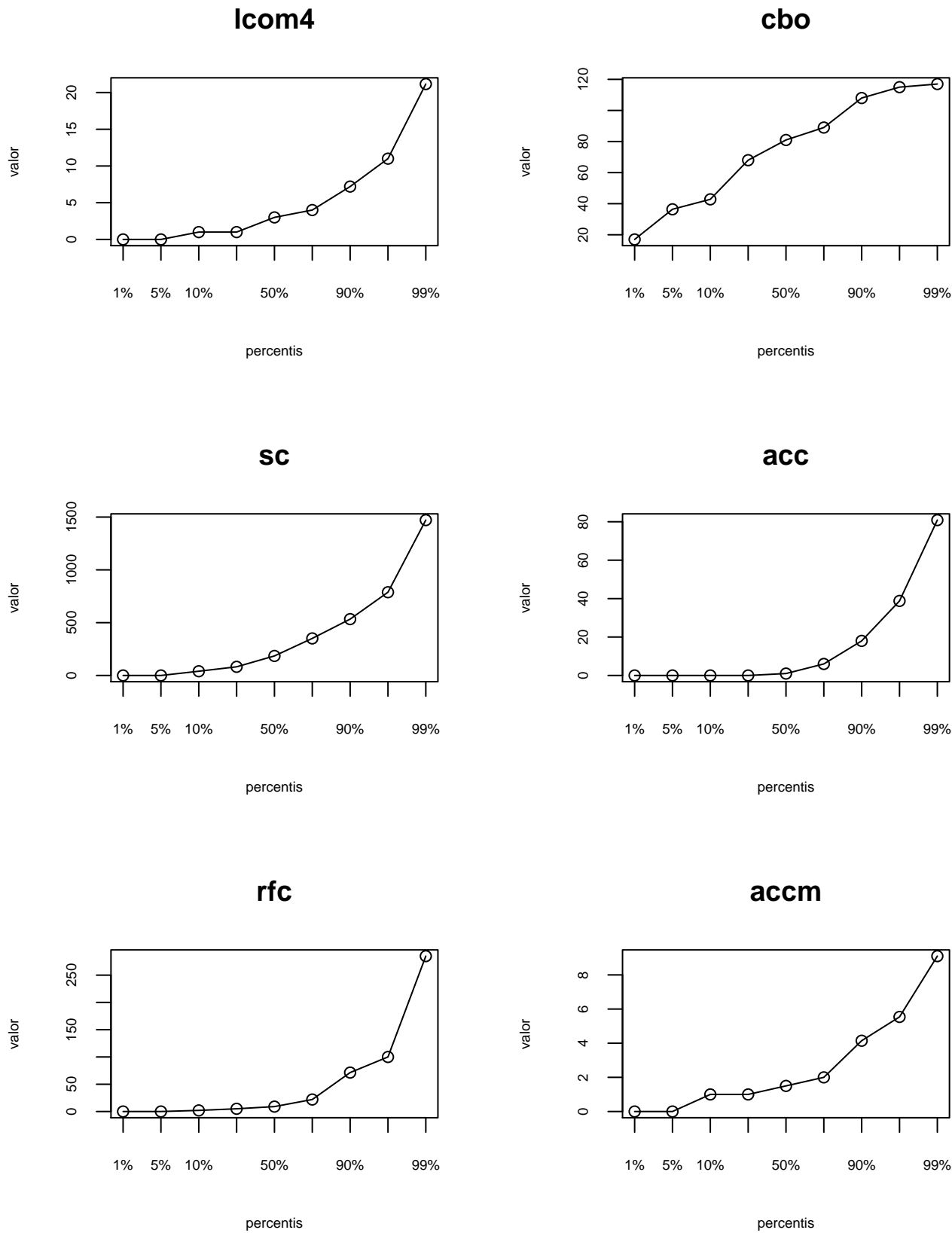


Figura 5.19: distribuição das métricas para a ferramenta pixy

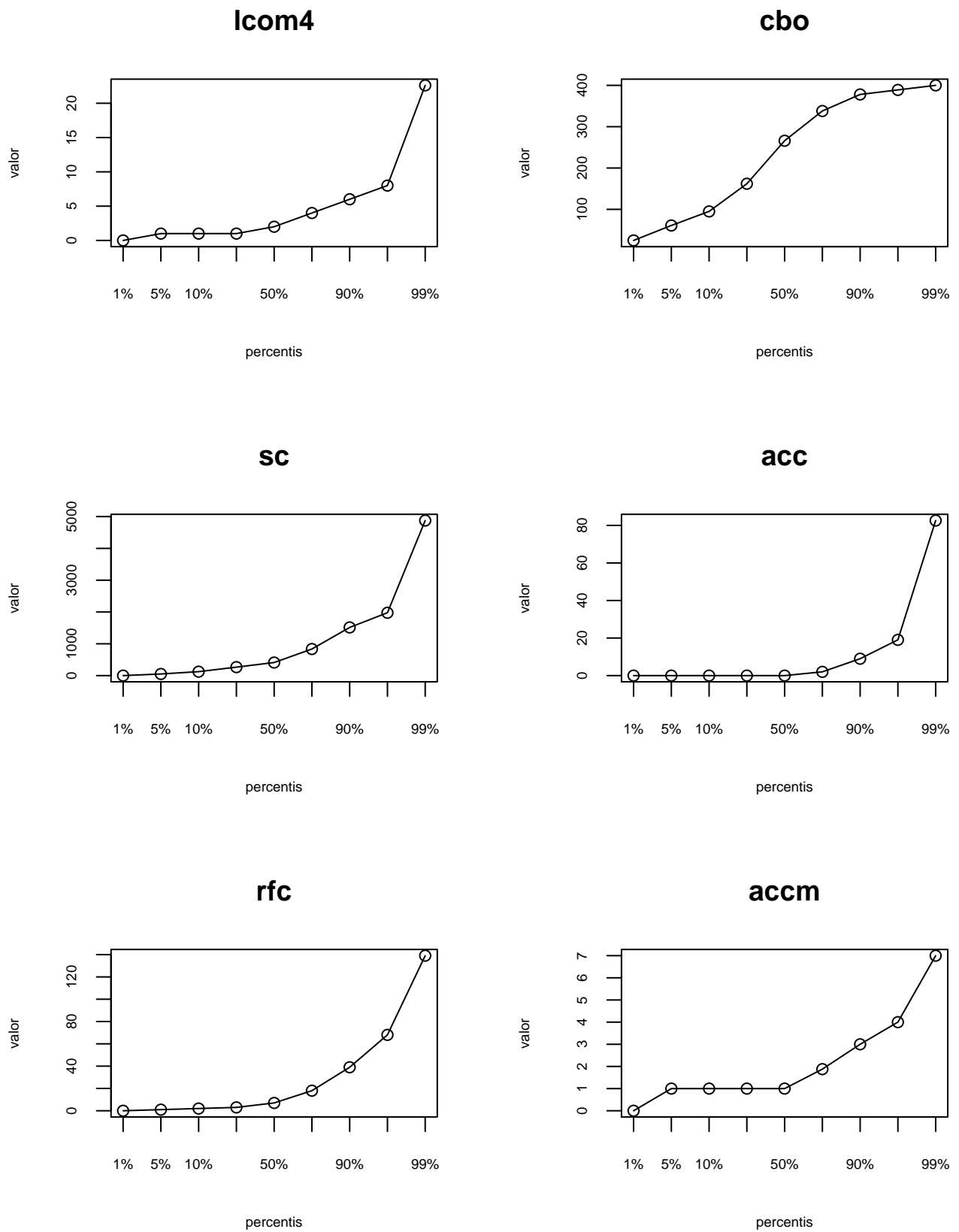


Figura 5.20: distribuição das métricas para a ferramenta pmd

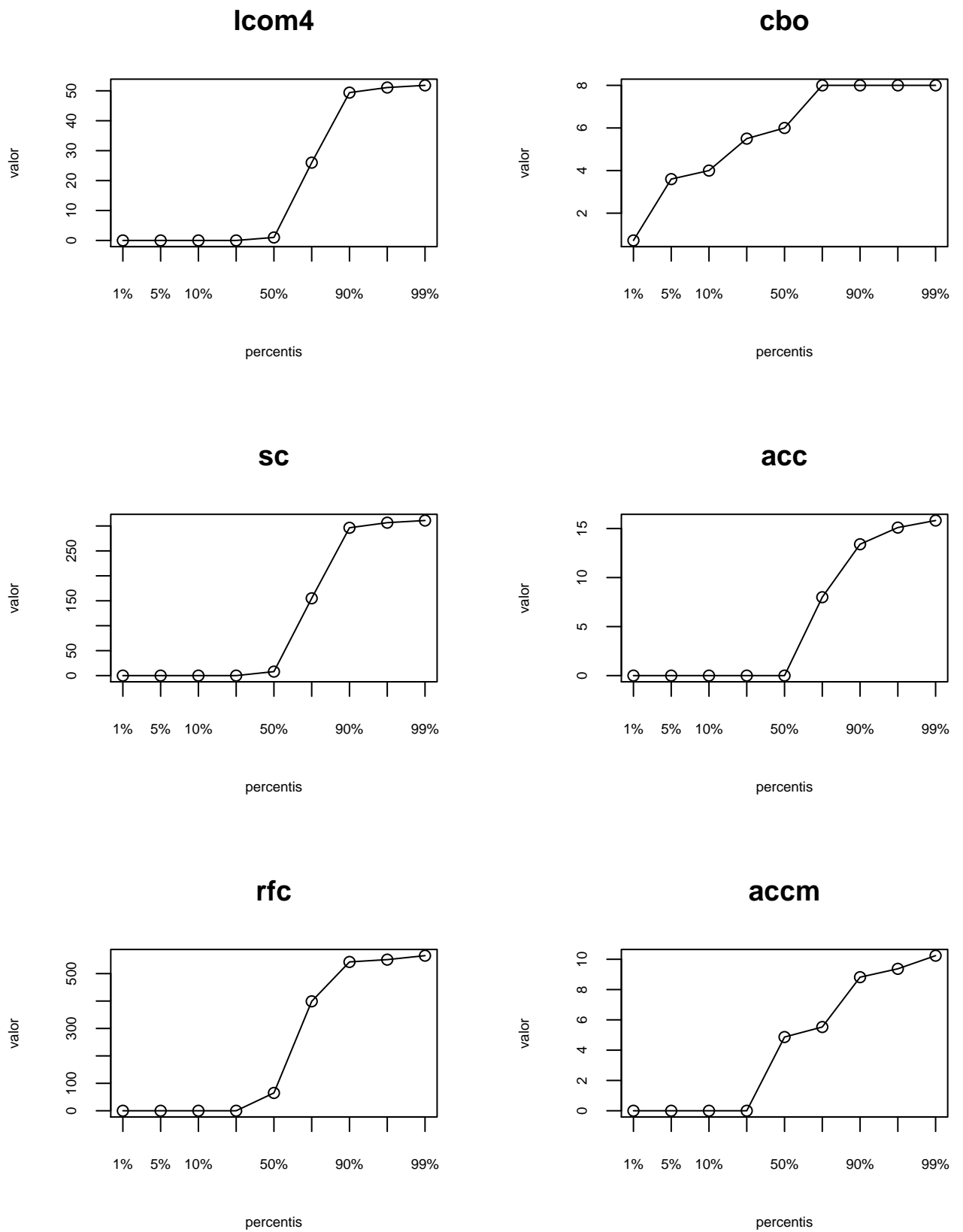


Figura 5.21: distribuição das métricas para a ferramenta rats

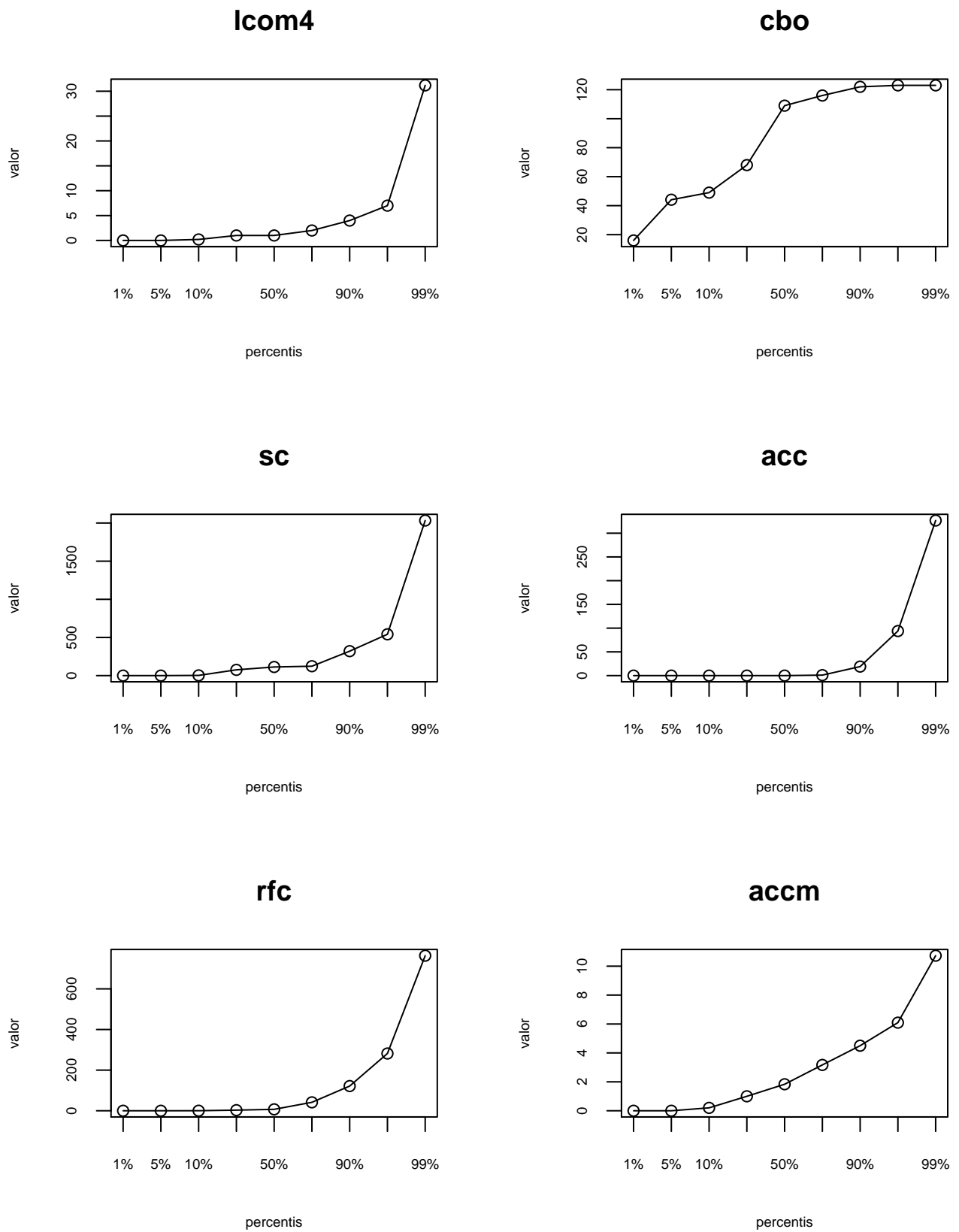


Figura 5.22: distribuição das métricas para a ferramenta smatch

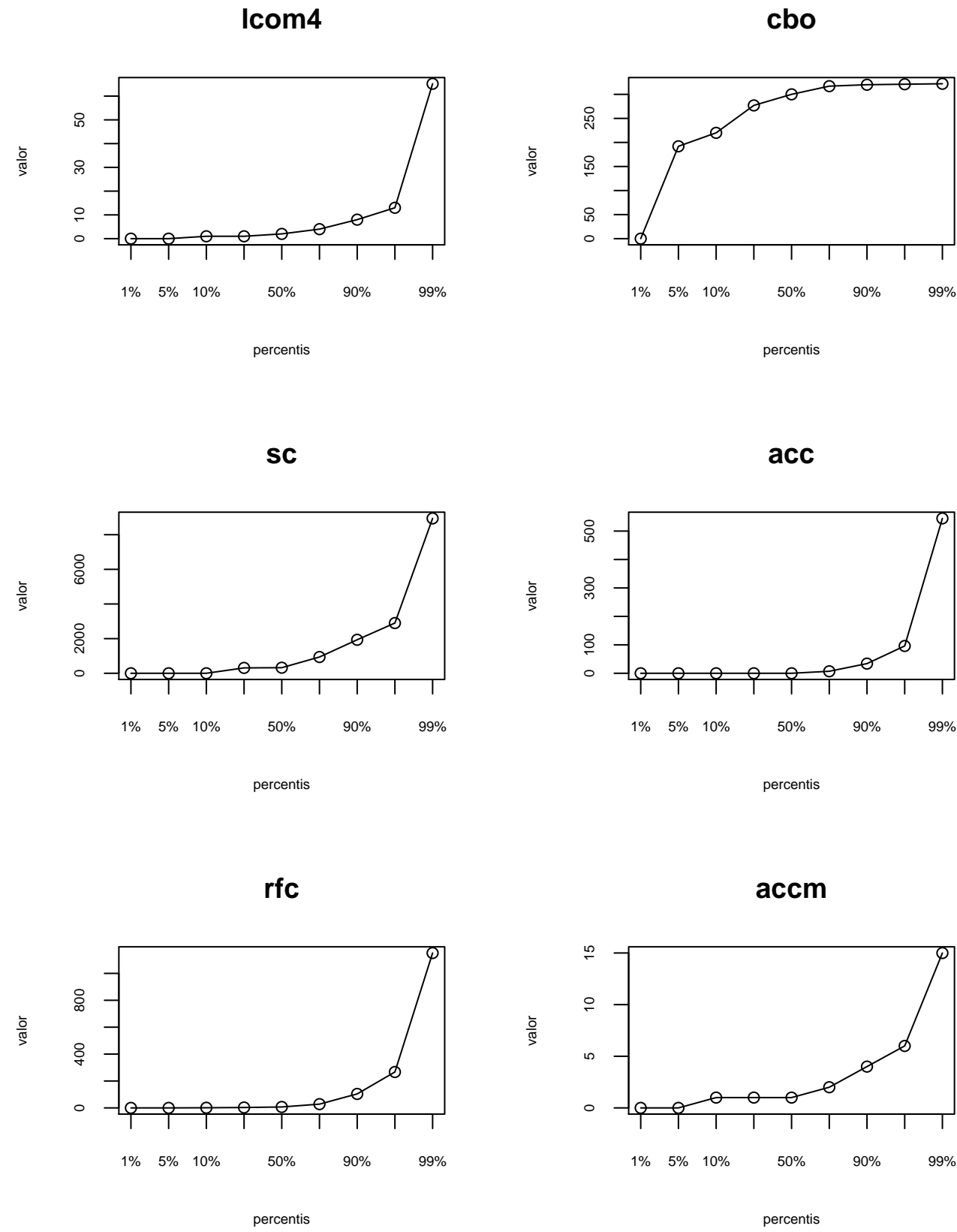


Figura 5.23: distribuição das métricas para a ferramenta splint

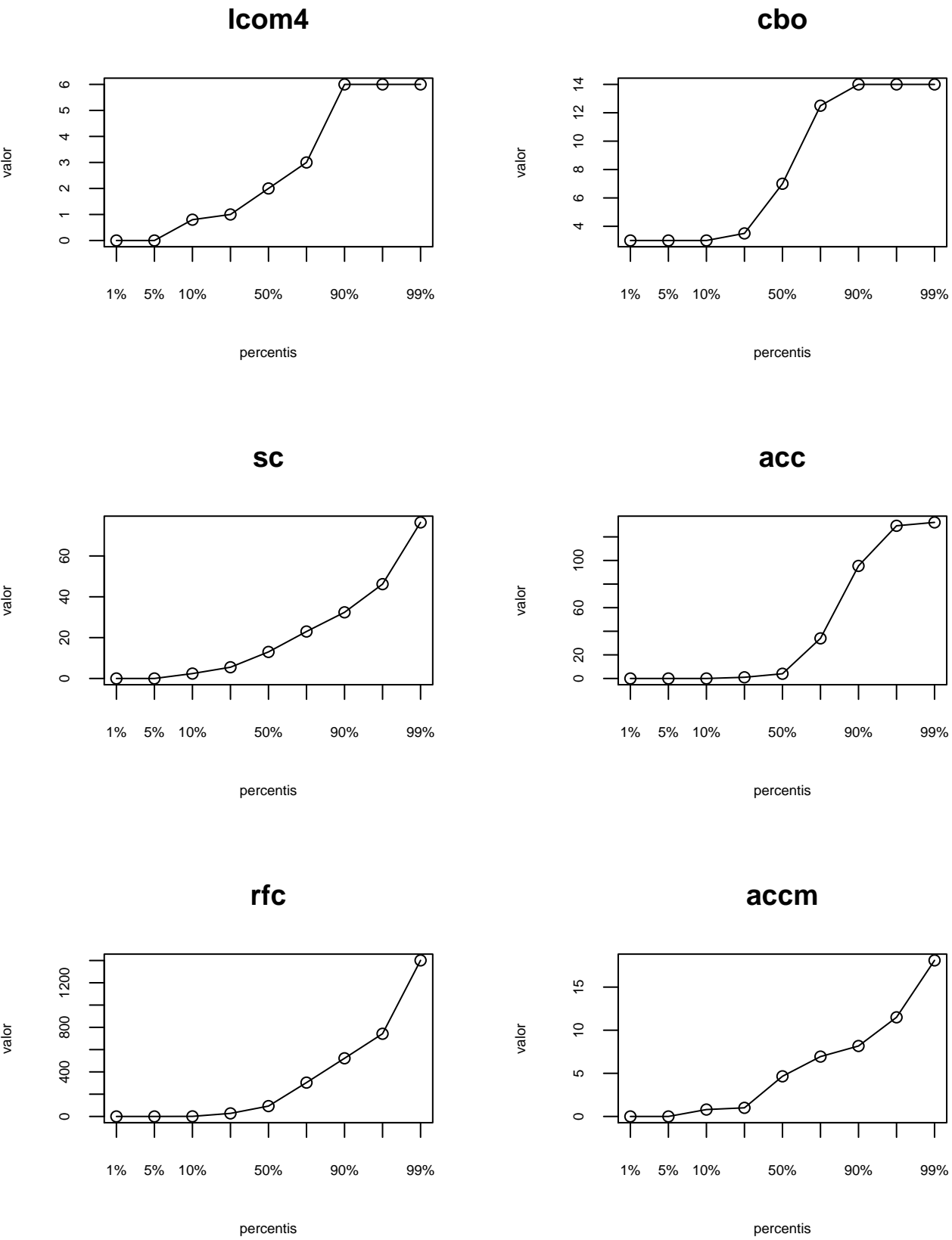


Figura 5.24: distribuição das métricas para a ferramenta uno

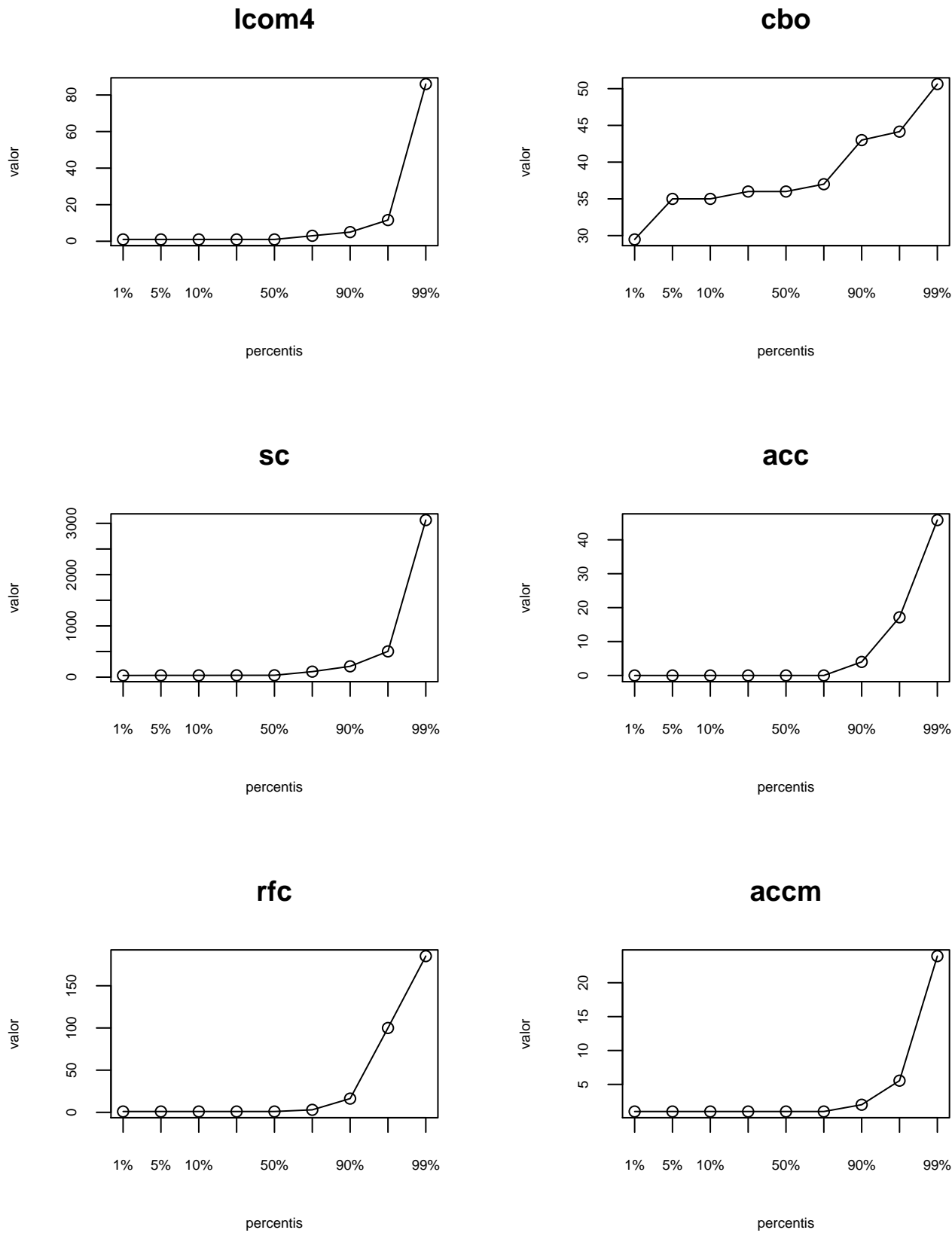


Figura 5.25: distribuição das métricas para a ferramenta wap

5.1.5 Gráfico resumido com distribuição de métricas

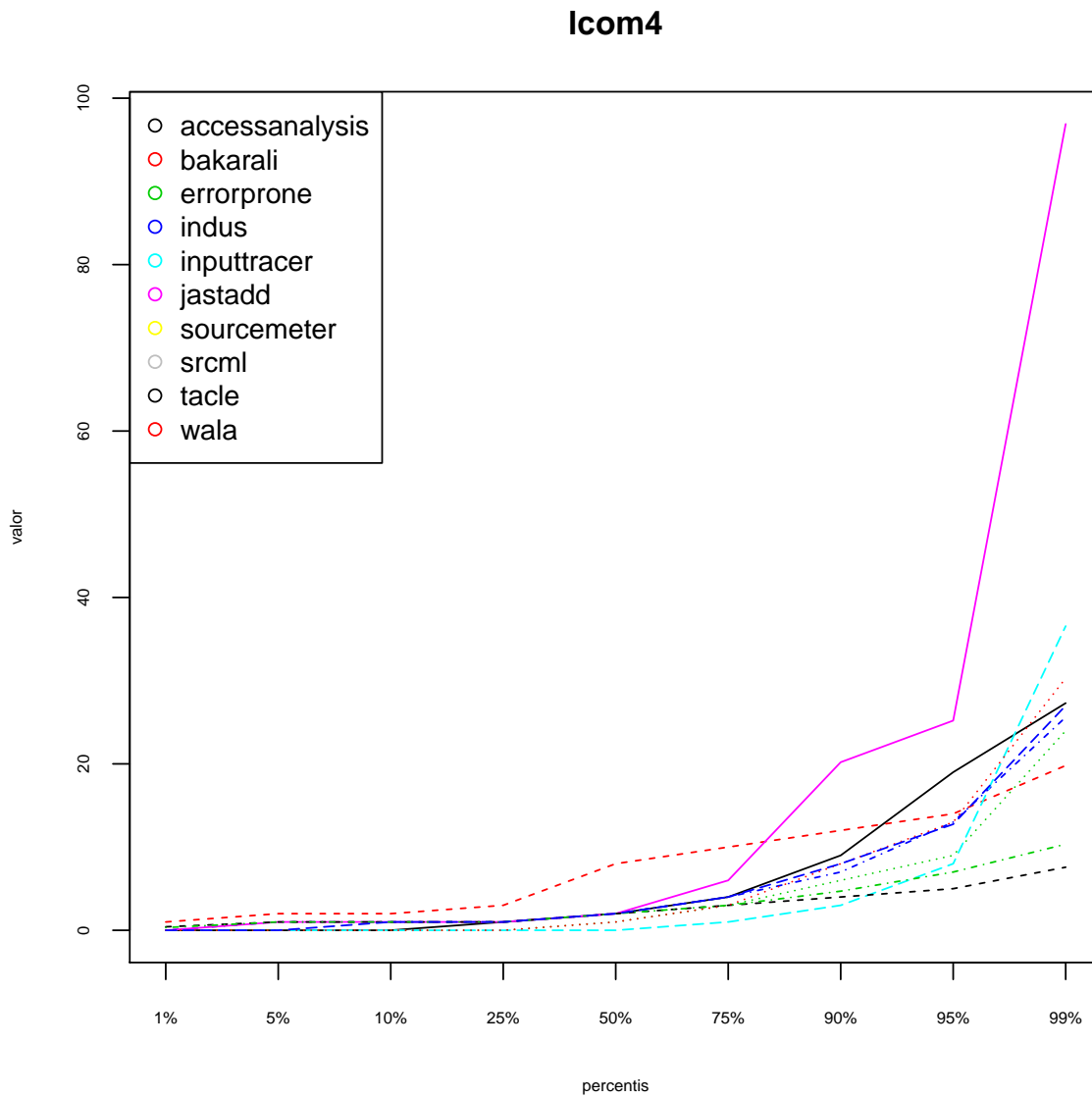


Figura 5.26: distribuição para todas as ferramenta

5.1.6 Regressão ...

...

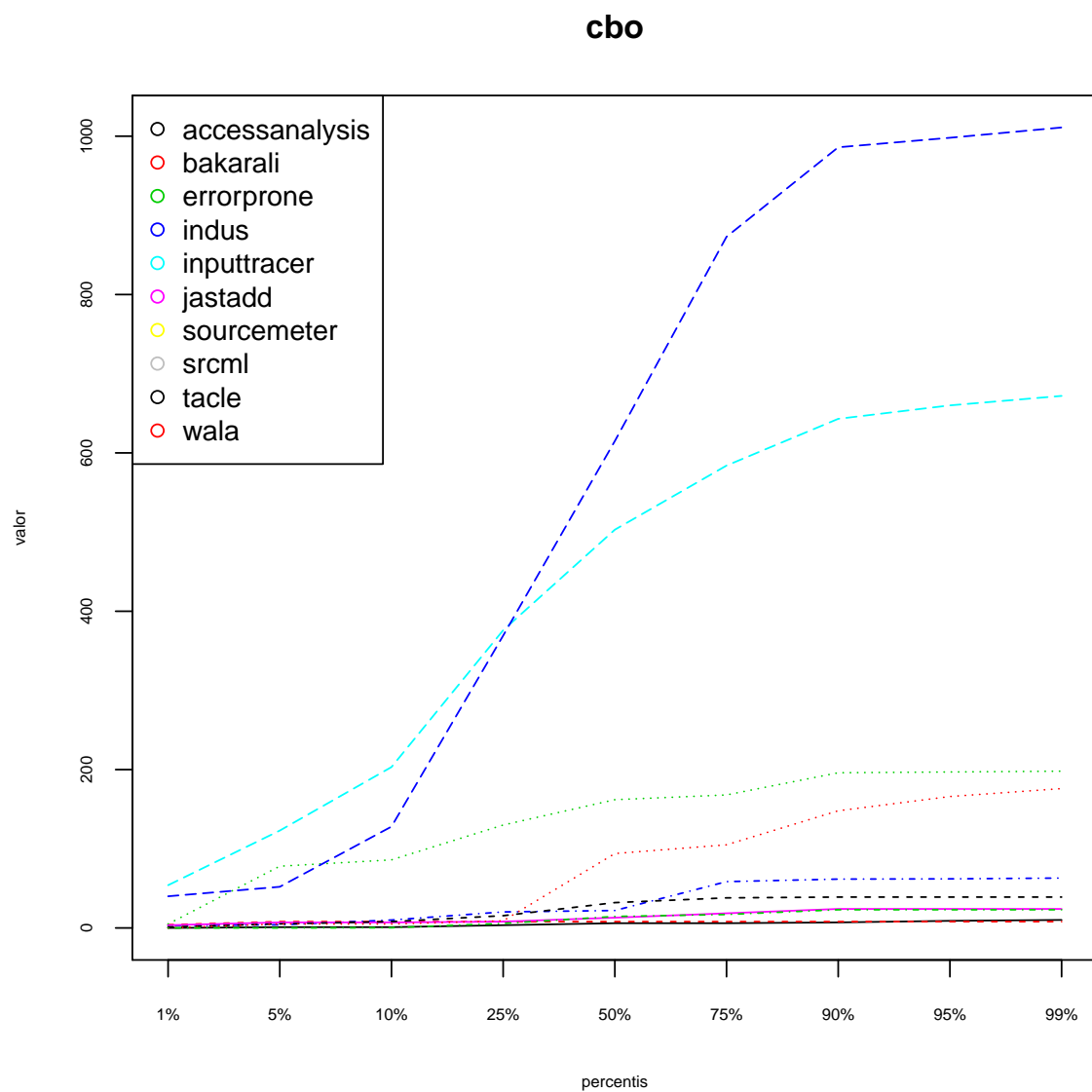


Figura 5.27: distribuição para todas as ferramenta

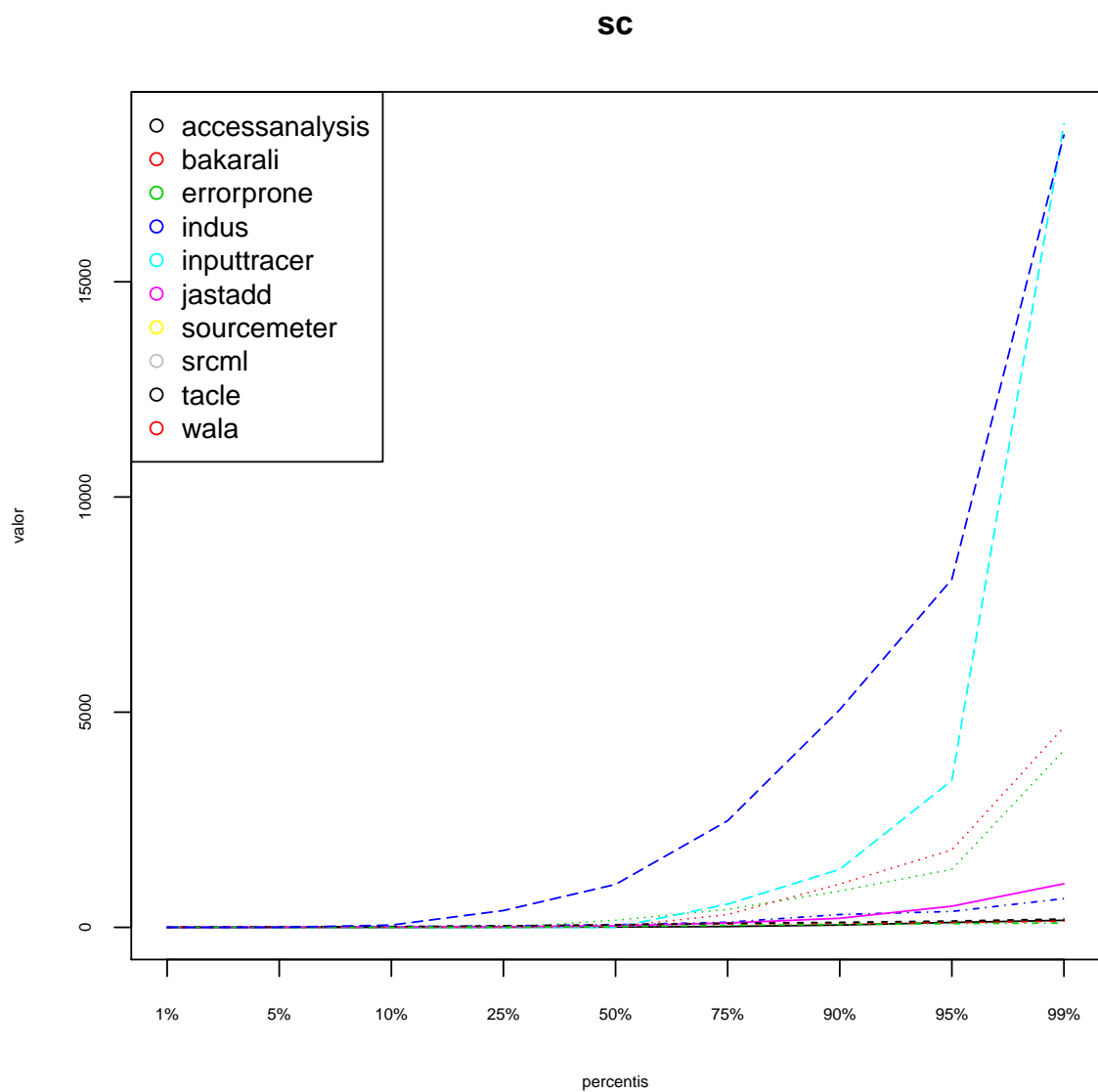


Figura 5.28: distribuição para todas as ferramenta

SC

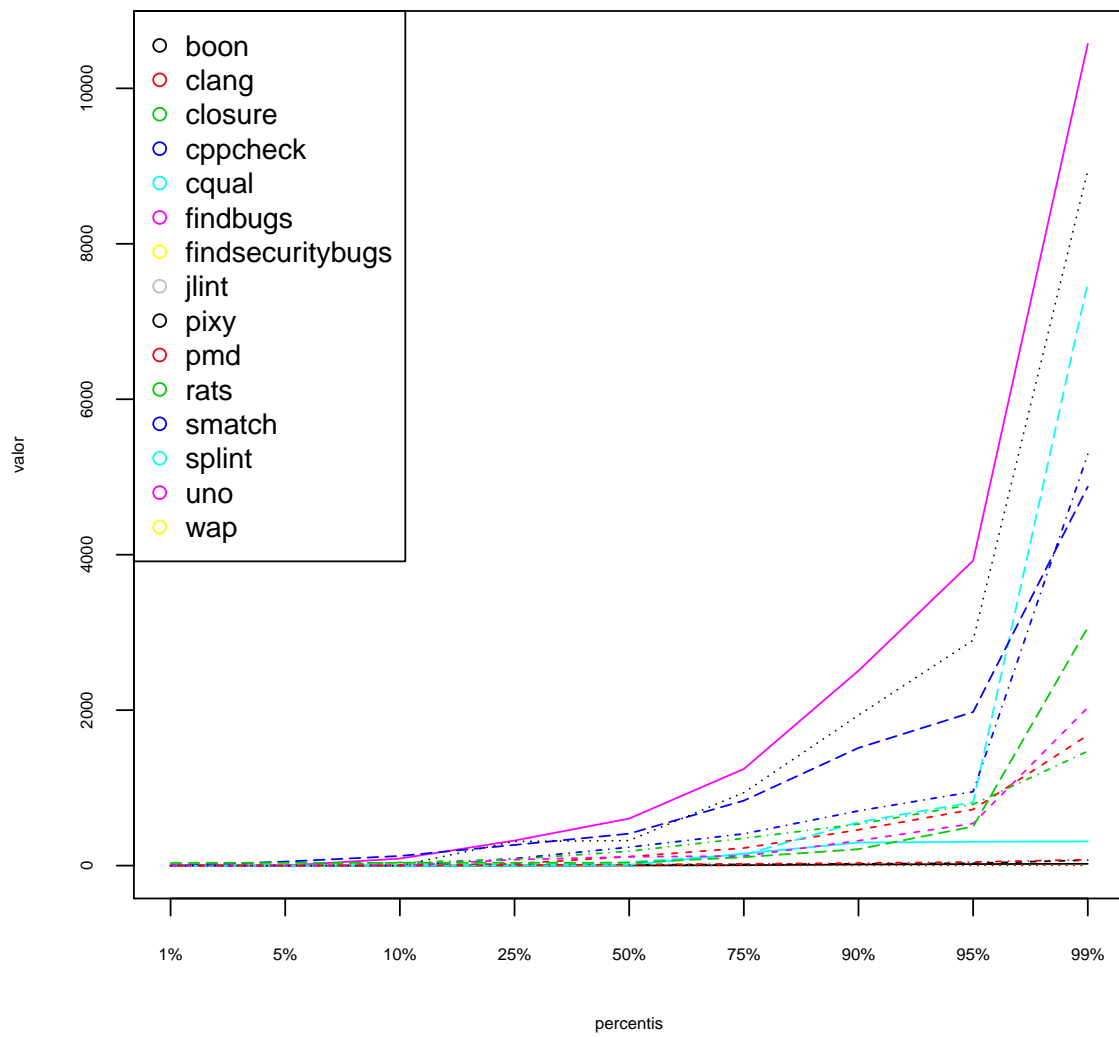


Figura 5.29: distribuição para todas as ferramenta da indústria

amloc

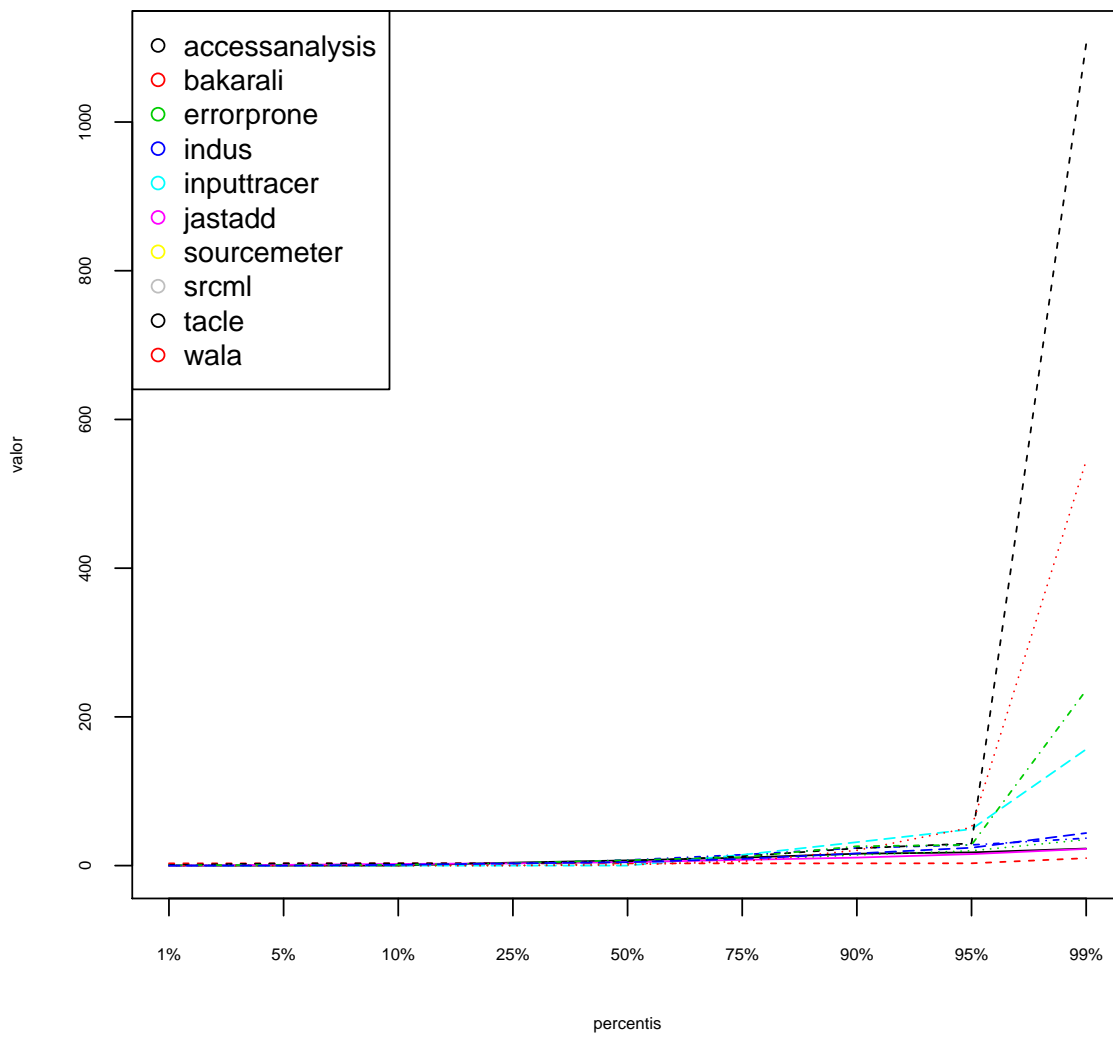


Figura 5.30: distribuição para todas as ferramenta

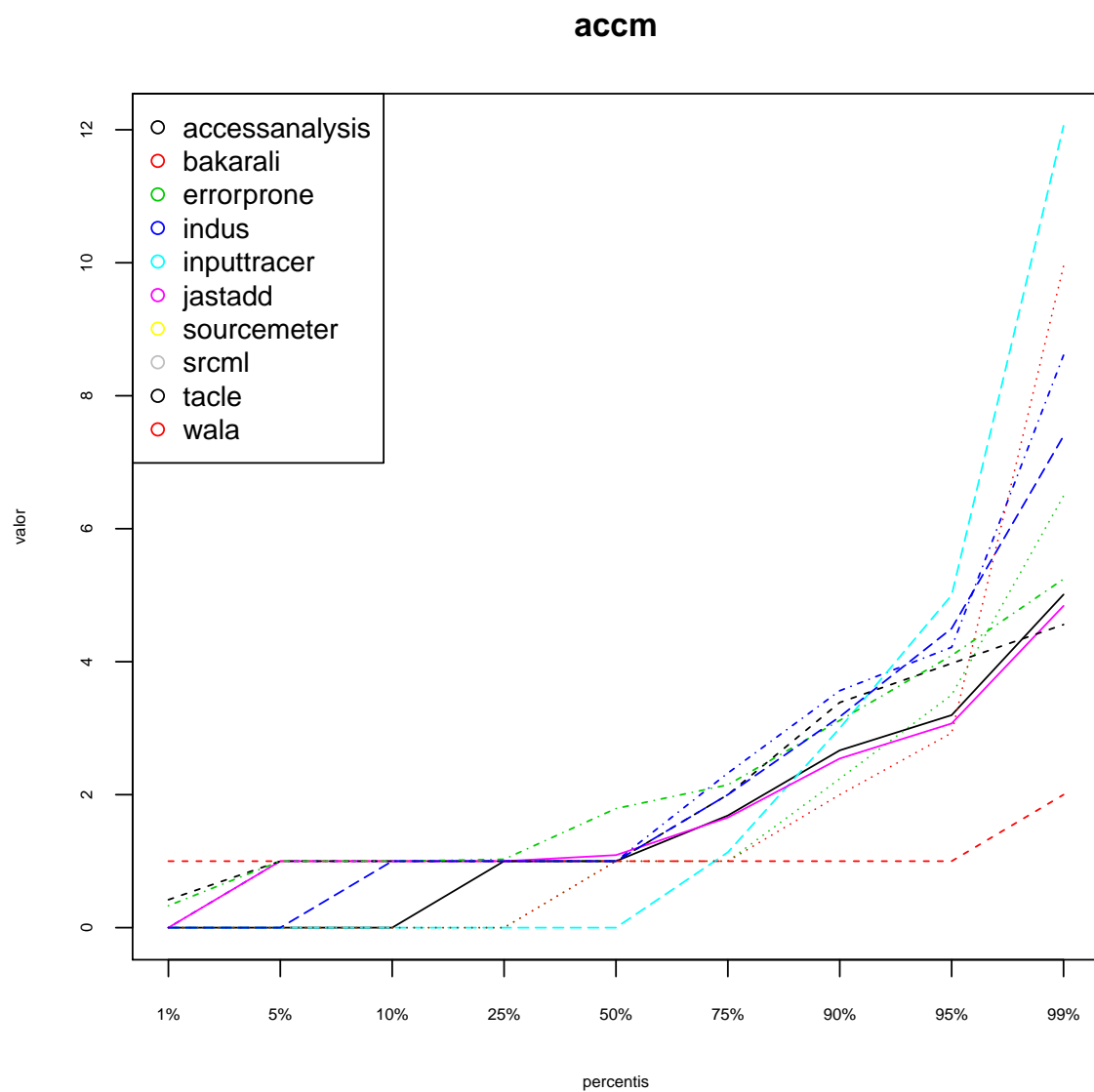


Figura 5.31: distribuição para todas as ferramenta

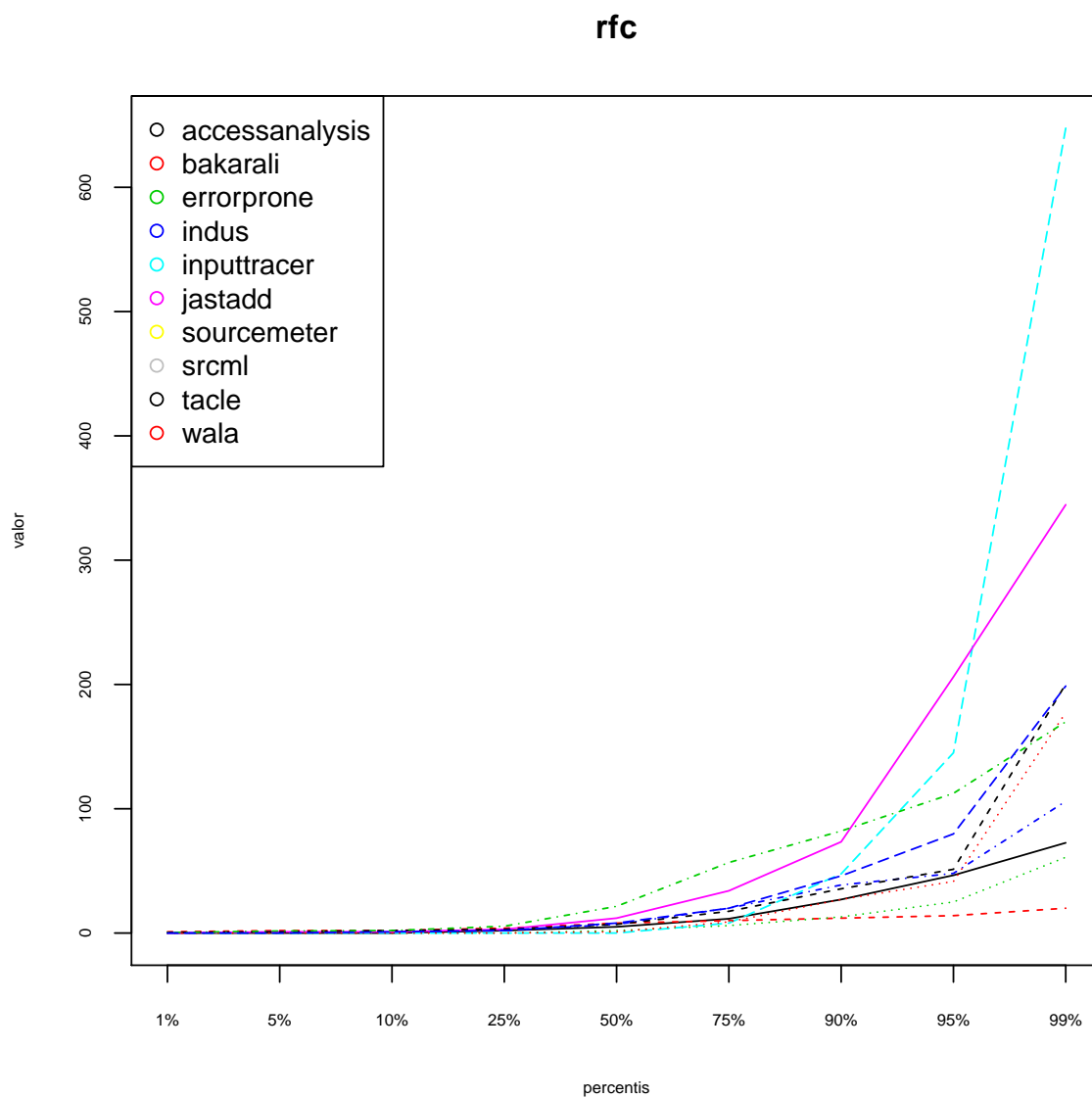


Figura 5.32: distribuição para todas as ferramenta

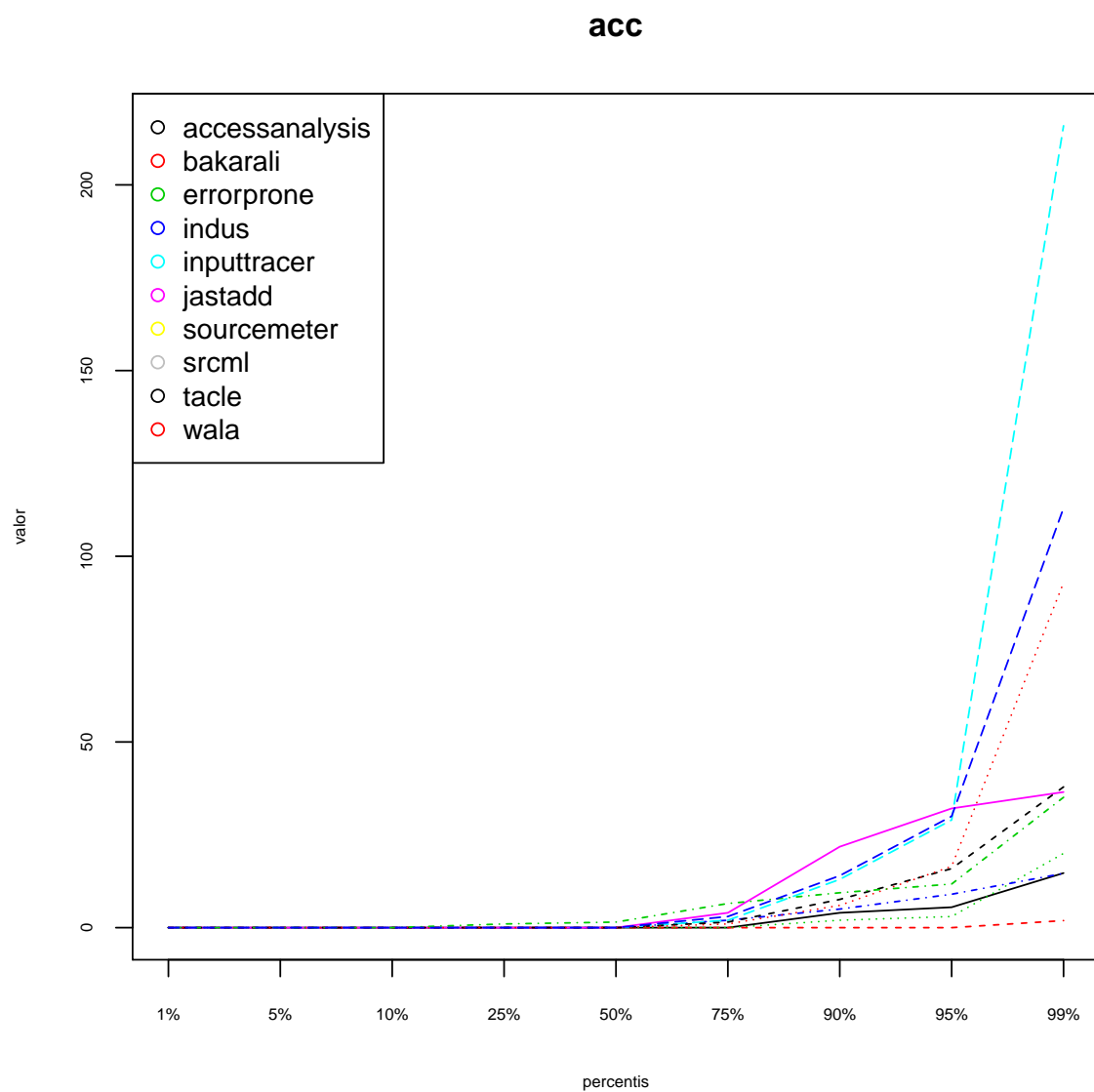


Figura 5.33: distribuição para todas as ferramenta

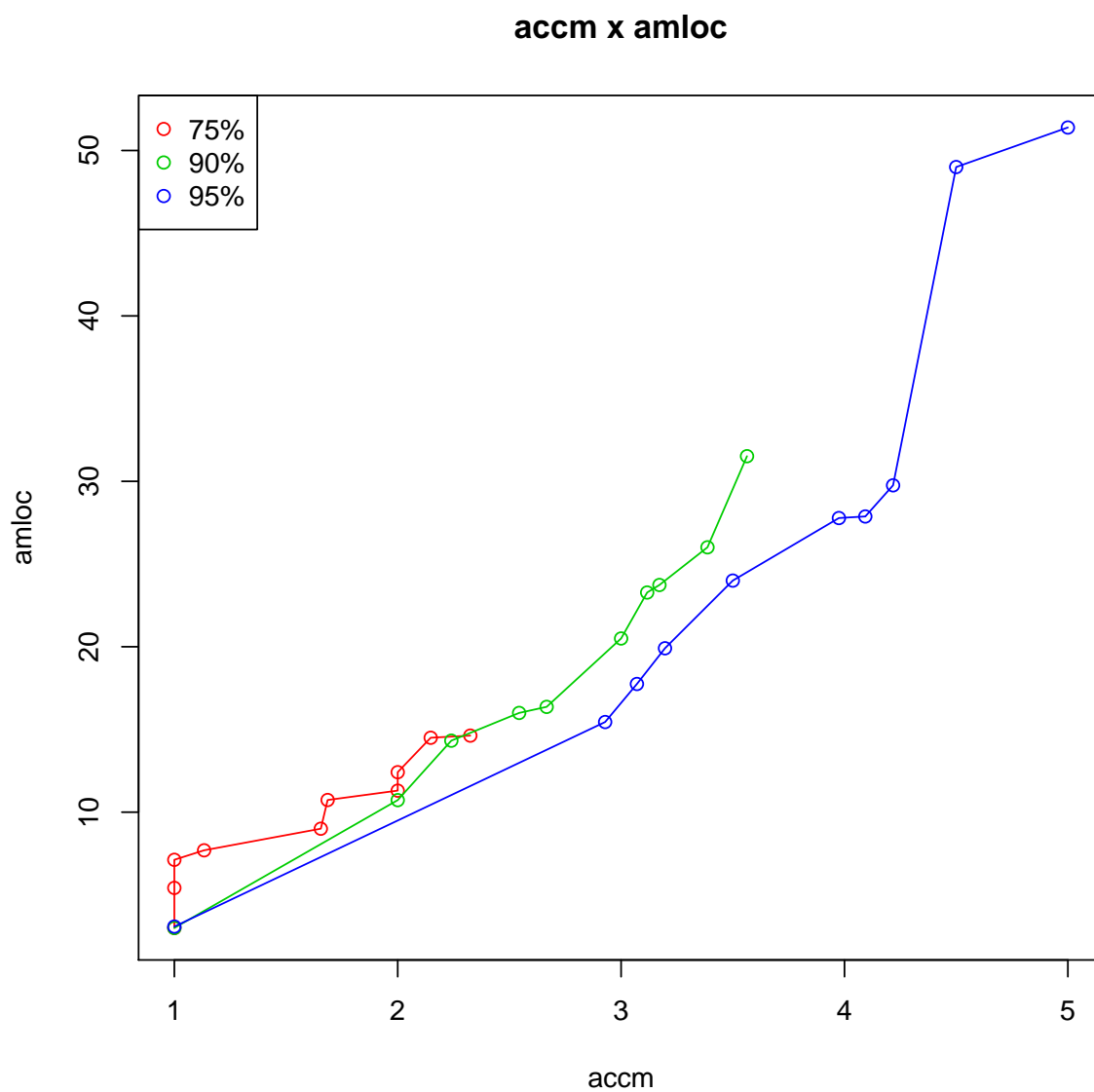


Figura 5.34: distribuição de percentis para todas as ferramentas

CAPÍTULO 6

CONCLUSÃO

(à fazer)

REFERÊNCIAS BIBLIOGRÁFICAS

- ALMEIDA, L. T.; MIRANDA, J. M. de. Código limpo e seu mapeamento para métricas de código fonte. *Monografia de Graduação em Ciência da Computação*, p. 74, 2010. Disponível em: <http://www.ime.usp.br/~cef/mac499-10/monografias/lucianna-joao/arquivos/monografia.pdf>.
- BECK, K. *Implementation Patterns*. [S.l.]: Pearson Education, 2007.
- BINKLEY, D. Source code analysis: A road map. In: IEEE COMPUTER SOCIETY. *2007 Future of Software Engineering*. [S.l.], 2007. p. 104–119.
- CERT. *Secure Coding Tools*. 2016. [Online; acessado 23 Abril de 2016]. Disponível em: <http://www.cert.org/secure-coding/tools/index.cfm>.
- CHIDAMBER, S. R.; KEMERER, C. F. A metrics suite for object oriented design. *Software Engineering, IEEE Transactions on*, IEEE, v. 20, n. 6, p. 476–493, 1994.
- DARCY, D. P. et al. The structural complexity of software an experimental test. *Software Engineering, IEEE Transactions on*, IEEE, v. 31, n. 11, p. 982–995, 2005.
- FEITELSON, D. G. From repeatability to reproducibility and corroboration. *ACM SIGOPS Operating Systems Review*, ACM, v. 49, n. 1, p. 3–11, 2015.
- HARRISON, R.; COUNSELL, S. J.; NITHI, R. V. An evaluation of the mood set of object-oriented software metrics. *Software Engineering, IEEE Transactions on*, IEEE, v. 24, n. 6, p. 491–496, 1998.
- HITZ, M.; MONTAZERI, B. *Measuring Coupling and Cohesion In Object-Oriented Systems*. [s.n.], 1995. Disponível em: <http://www.isys.uni-klu.ac.at/PDF/1995-0043-MHBM.pdf>.
- ISO, I. Iec25010: 2011 systems and software engineering—systems and software quality requirements and evaluation (square)—system and software quality models. *International Organization for Standardization*, p. 34, 2011.
- JÚNIOR, M. R. P. Estudo de métricas de código fonte no sistema android e seus aplicativos. p. 82, 2015. Disponível em: <https://fga.unb.br/tcc/software/tcc-2015.1-engenharia-de-software/marcos-ronaldo-pereira-junior/v3-tcc.pdf>.
- KAZMAN, A. T. R. On the worthiness of software engineering research. 2016. Disponível em: http://shidler.hawaii.edu/sites/shidler.hawaii.edu/files/users/kazman/se_research_worthiness.pdf.

- KIRKOV, R.; AGRE, G. Source code analysis - an overview. *Cybernetics and Information Technologies*, v. 10, n. 2, p. 60–77, 2010.
- KON, F. et al. Free and open source software development and research: Opportunities for software engineering. In: *SBES*. [s.n.], 2011. p. 82–91. Disponível em: <http://dblp.org/db/conf/sbes/sbes2011.html\#KonMLTCM11>.
- KRISHNAMURTHI, S.; VITEK, J. The real software crisis: Repeatability as a core value. *Communications of the ACM*, ACM, v. 58, n. 3, p. 34–36, 2015.
- MAFRA, S. N.; TRAVASSOS, G. H. Estudos primários e secundários apoiando a busca por evidência em engenharia de software. 2006.
- MARTIN, R. C.; HAN, L. *Clean Code*. [S.l.]: Publishing House of Electronics Industry, 2012.
- MCCABE, T. J. A complexity measure. *Software Engineering, IEEE Transactions on*, IEEE, n. 4, p. 308–320, 1976.
- MCCONNELL, S. *Code Complete*. 2nd. ed. [S.l.]: Microsoft Press, 2004.
- MEIRELLES, P. R. M. *Monitoramento de métricas de código-fonte em projetos de software livre*. Tese (Doutorado) — Universidade de São Paulo, São Paulo, Brazil, 2013.
- NIST. *SAMATE - Source Code Security Analyzers*. 2016. [Online; acessado 20 Abril de 2016]. Disponível em: http://samate.nist.gov/index.php/Source_Code_Security_Analyzers.html.
- PENG, R. D. Reproducible research in computational science. *Science (New York, Ny)*, NIH Public Access, v. 334, n. 6060, p. 1226, 2011.
- PORTILLO-RODRÍGUEZ, J. et al. Tools used in global software engineering: A systematic mapping review. p. 663–685, 2012. Disponível em: <http://dblp.org/db/journals/infsof/infsof54.html\#Portillo-RodriguezVPB12>.
- PRLÍČ, A.; PROCTER, J. B. Ten simple rules for the open development of scientific software. *PLoS Computational Biology*, vol. 8, issue 12, p. e1002802, v. 8, p. 2802, dec 2012.
- RIBEIRO, A. C. Análise estática de código-fonte com foco em segurança: Metodologia para avaliação de ferramentas. 2015.
- ROSENBERG, L. H.; HYATT, L. E. Software quality metrics for object-oriented environments. *Crosstalk journal*, v. 10, n. 4, 1997.
- SHARBLE, R. C.; COHEN, S. S. The object-oriented brewery: a comparison of two object-oriented development methods. *ACM SIGSOFT Software Engineering Notes*, ACM, v. 18, n. 2, p. 60–73, 1993.

SHIH, T. K. et al. Decomposition of inheritance hierarchy dags for object-oriented software metrics. In: *Engineering of Computer-Based Systems, 1997. Proceedings., International Conference and Workshop on*. [S.l.: s.n.], 1997. p. 238–245.

SOCIETY, I. C. *Guide to the Software Engineering Body of Knowledge*. Version 3.0. [S.l.], 2014.

SPIN. *Static Source Code Analysis Tools for C*. 2016. [Online; acessado 23 Abril de 2016]. Disponível em: <http://www.spinroot.com/static>.

STOL, K.-J.; FITZGERALD, B. A holistic overview of software engineering research strategies. In: *3rd International Workshop on Conducting Empirical Studies in Industry*. [S.l.: s.n.], 2015. p. 8.

TERCEIRO, A. et al. Analizo: an extensible multi-language source code analysis and visualization toolkit. p. 6, 2010.

VITEK, J.; KALIBERA, T. Repeatability, reproducibility, and rigor in systems research. In: ACM. *Proceedings of the ninth ACM international conference on Embedded software*. [S.l.], 2011. p. 33–38.

VRIES, A. de. *Introducing the Reproducible R Toolkit and the checkpoint package*. 2014. Internet. [Online; Acessado em 24 de Maio de 2016]. Disponível em: <http://blog.revolutionanalytics.com/2014/10/introducing-rrt.html>.

WHEELER, D. A. *Static analysis tools for security*. 2015. [Online; acessado 23 de Abril de 2016]. Disponível em: <http://www.dwheeler.com/essays/static-analysis-tools.html>.

WIKIPEDIA. *List of tools for static code analysis*. 2016. [Online; acessado 23 Abril de 2016]. Disponível em: https://en.wikipedia.org/wiki/List_of_tools_for_static_code_analysis.