Universidade de São Paulo
Instituto de Matemática e Estatística

Centro de Estatística Aplicada

Relatório de Análise Estatística

RAE-CEA–22P05

RELATÓRIO DE ANÁLISE ESTATÍSTICA SOBRE O PROJETO:

"Influência das métricas de software no engajamento de contribuidores em repositórios de código aberto"

Airlane Pereira Alencar
Ana Cristina Vieira de Melo
Diego Cardozo Sandrim
Francisco Marcelo Monteiro da Rocha
Tereza Cristina de Oliveira Lacerda

São Paulo, julho de 2022

CENTRO DE ESTATÍSTICA APLICADA - CEA – USP

TÍTULO: Relatório de Análise Estatística sobre o Projeto: “Influência das métricas de software no engajamento de contribuidores em repositórios de código aberto”.

PESQUISADOR: Diego Cardozo Sandrim

ORIENTADORA: Prof. Dra. Ana Cristina Vieira de Melo

INSTITUIÇÃO: Instituto de Matemática e Estatística da USP

FINALIDADE DO PROJETO: Mestrado

RESPONSÁVEIS PELA ANÁLISE: Airlane Pereira Alencar
Francisco Marcelo Monteiro da Rocha
Tereza Cristina de Oliveira Lacerda

FICHA TÉCNICA

REFERÊNCIAS BIBLIOGRÁFICAS:

CHARRAD, M.; GHAZZALI, N.; BOITEAU, V.; NIKNAFS, A. (2014). NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set. **Journal of Statistical Software**, **61**, 1–36. DOI: 10.18637/jss.v061.i06. Disponível em: <https://www.jstatsoft.org/index.php/jss/article/view/v061i06>. Acesso em: 19 de maio de 2022.

SINGER, J.M.; ROCHA, F.M.M.; NOBRE, J.S.N. (2017). Graphical Tools for Detecting Departures from Linear Mixed Model Assumptions and Some Remedial Measures. **International Statistical Review, 85, 2, 290 – 324**

SINGER, J.M.; NOBRE, J.S.; ROCHA F.M.M. (2018). **Análise de dados longitudinais**. Versão parcial preliminar, em produção.

MAECHLER, M.; ROUSSEEUW, P.; STRUYF, A.; HUBERT, M.; HORNIK, K. (2013). **cluster: Cluster Analysis Basics and Extensions. R package version 1.14.4**. Disponível em: <https://cran.r-project.org/web/packages/cluster/cluster.pdf>. Acesso em: 19 de maio de 2022.

PROGRAMAS COMPUTACIONAIS UTILIZADOS:

Microsoft Word for Windows (versão 2019)

R (versão 4.1.1)

RStudio (versão 1.4.1717)

TÉCNICAS ESTATÍSTICAS UTILIZADAS

Análise Descritiva Unidimensional (03:010)

Análise Descritiva Multidimensional (03:020)

Análise de Conglomerados (06:120)

Análise de Regressão Linear Misto (07:990)

ÁREA DE APLICAÇÃO

Ciência da Computação e Autômatos (14:150)

Resumo

Diversos softwares são criados e mantidos no formato de código fonte aberto. Nesse formato, qualquer desenvolvedor é bem-vindo a contribuir na construção e na melhoria desse software, desenvolvendo novas funcionalidades, corrigindo bugs, melhorando a documentação, entre outras tarefas relacionadas.

Existem algumas métricas que podem ser avaliadas estaticamente nos softwares, entre elas: quantidade de más práticas de código, quantidade de comentários, quantidade de linhas duplicadas, quantidade de linhas de código por função e por arquivo. A análise estatística realizada nesse relatório tem como objetivo avaliar a influência dessas métricas na quantidade de contribuidores de código.

Para isso, foram selecionados projetos da plataforma Github, escritos na linguagem de programação Go, e que se incluíssem em uma das seguintes categorias: Full stack web frameworks; Middlewares; Libraries for creating HTTP middlewares; Routers. Tais projetos foram avaliados no período de 2011 a 2022, e as variáveis de interesse foram coletadas semestralmente.

Para a modelagem dos dados, foi proposto um modelo linear misto gaussiano (Singer et al., 2018), o qual se adequou bem aos dados coletados.

Com base no modelo ajustado, obteve-se como resultado que a quantidade de contribuidores está associada de maneira negativa com a densidade de más práticas de código, e associada de maneira positiva com a densidade de funções no código.

Sumário

1. Introdução

Diversos softwares são criados e mantidos no formato de código fonte aberto. Nesse formato, qualquer desenvolvedor é bem-vindo a contribuir na construção e na melhoria desse software, desenvolvendo novas funcionalidades, corrigindo bugs, melhorando a documentação, entre outras tarefas relacionadas.

Geralmente esse trabalho é voluntário, e quem contribuiu possui interesse de aprender sobre o tema de software, melhorar a sua reputação como desenvolvedor, resolver um problema em outro sistema que depende do software em questão, entre outros.

Existe uma grande disparidade no número de contribuidores, alguns projetos possuem milhares de contribuidores, e outros projetos apenas poucas pessoas. Essa diferença pode estar relacionada com diversos fatores, entre eles: quantas pessoas usam o software, maturidade do software, clima entre os colaboradores. Há também evidências de que a complexidade de código pode estar relacionada a essa disparidade.

Além da complexidade, existem outras métricas que podem ser avaliadas estaticamente no software, entre elas: quantidade de más práticas de código, quantidade de comentários, quantidade de linhas duplicadas, quantidade de linhas de código por função e por arquivo, as quais podem ou não ter influência na quantidade de contribuidores de código.

2. Objetivos

Nesse projeto, tem-se como principais objetivos:

- Entender o relacionamento entre a quantidade de colaboradores e as métricas de código;
- Verificar quais métricas influenciam na quantidade de contribuidores com o passar do tempo;
- Verificar a intensidade da influência dessas métricas na quantidade dos colaboradores.

3. Descrição do estudo

Para o estudo, foram selecionados projetos da plataforma Github, escritos na linguagem de programação Go, e que se incluíssem em uma das seguintes categorias:

- Full stack web frameworks;
- Middlewares;
- Libraries for creating HTTP middlewares;
- Routers.

Para cada projeto, foram coletadas métricas de código semestralmente (o Github é um sistema de controle de versão, então é possível acessar o passado do código desde a data em que foi criado, até o seu estado atual). Como cada projeto tem uma duração diferente, trata-se de um estudo longitudinal desbalanceado, no período de 2011 a 2022.

Os critérios de exclusão utilizados foram: projetos que não tiveram contribuidor em algum semestre; projetos que possuem menos de 2 anos de duração.

Aplicados os critérios, restaram 32 projetos para a análise.

4. Descrição das variáveis

4.1 Variáveis explicativas

- **Data**: data da coleta da métrica.
- **Tempo**: número de semestres decorridos desde a coleta da primeira métrica (01/01/2013). É uma transformação bijetora da variável data

As variáveis data e tempo representam a mesma coisa.

- **Linhas de código**: quantidade de linhas de código do projeto
- **Más práticas**: quantidade de más práticas de software
- **Complexidade cognitiva**: complexidade cognitiva do código
- **Linhas de comentário**: quantidade de linhas de comentário
- **Complexidade ciclomática**: complexidade ciclomática do código

- **Linhas duplicadas**: quantidade de linhas de código duplicadas
- **Arquivos**: quantidade de arquivos
- **Funções:** quantidade de funções
- **Classes**: quantidade de classes
- **Índice de qualidade**: índice de qualidade definido pelo padrão SQALE

4.2 Variável resposta

- **Contribuidores:** quantidade de colaboradores distintos no semestre

4.3 Padronização das variáveis explicativas

Como todas as métricas de código não possuem um limite superior (ou seja, podem crescer indefinidamente conforme o crescimento do projeto), optou-se por fazer uma padronização pelo número de linhas de código, ou seja, cada uma das variáveis explicativas foi dividida pelo número de linhas de código do projeto na respectiva data. Dessa maneira, tem-se a densidade de cada métrica por linha de código, tornando os projetos mais comparáveis entre si.

Pela Figura B.1, pode-se ver que sem aplicar a padronização, todas as variáveis explicativas apresentam um comportamento muito semelhante. Já na Figura B.2 (após feita a padronização), observa-se que as variáveis já apresentam comportamentos mais distintos.

Todas as análises descritivas e inferenciais serão feitas com base nessa padronização.

5. Análise descritiva

Nesta seção, é apresentada a análise descritiva dos dados, que nos permite ter uma visão inicial dos resultados do estudo.

5.1 Agrupamento dos projetos

Como os projetos selecionados para o estudo apresentavam comportamentos muito distintos entre si, foi aplicada uma análise de agrupamentos a fim de separá-los em grupos mais homogêneos, e dessa maneira, obter conclusões específicas para cada grupo.

Esse agrupamento não será levado em conta na análise inferencial, entretanto, uma parte da análise descritiva será feita com base nesses agrupamentos, por interesse do pesquisador.

O agrupamento foi feito utilizando-se todas as variáveis explicativas numéricas (para cada projeto, foram calculadas a média, mediana, máximo e mínimo de cada variável explicativa), a distância euclidiana, e o método hierárquico Ward, obtendo uma correlação cofenética de 0,88, indicando uma boa adequação do método de agrupamento. O número de grupos escolhido foi três. O agrupamento foi feito com auxílio do programa R, e das bibliotecas "cluster" e "NbClust" (Charrad, 2014; Maechler, 2013).

Fazendo agora um comparativo dos grupos formados, pela Tabela A.1, pode-se ver que:

- O grupo 1 contém 17 dos 32 projetos, os quais possuem uma quantidade baixa de semestres observados (ou seja, uma duração mais baixa em relação aos demais grupos).
- O grupo 2 é composto por 11 projetos e o grupo 3 por 4 projetos, com uma duração média e mediana parecidas

Agora, observando a Tabela A.2, nota-se que entre as 105 observações dos projetos do grupo 1, entre as 129 do grupo 2 e as 49 do grupo 3, a média e a mediana do número de contribuidores são menores no grupo 1, seguidas pelo grupo 2, que

mostram um pouco mais de contribuidores. Já o grupo 3 é caracterizado pela maior quantidade de contribuidores em relação aos demais grupos.

Essa distinção entre o número de contribuidores em cada grupo é muito importante para a análise, uma vez que essa variável não foi levada em conta para fazer o agrupamento, ou seja, apenas a combinação das variáveis explicativas em cada grupo já foi capaz de separar os projetos em três grupos distintos em relação à nossa variável de interesse, indicando que as métricas de código estão relacionadas com o número de contribuidores.

A análise descritiva que segue tem como principal objetivo fazer a caracterização dos grupos com base nas variáveis do estudo e no número total de observações em cada grupo.

5.1.1 Histogramas suavizados

Na Figura B.3, pode-se ver que a média e a variância do número de contribuidores cresce com o grupo, ou seja, os grupos 1 e 3 possuem menor e maior média e variância, respectivamente.

Pelas Figuras B.4 e B.5, pode-se notar uma clara distinção do grupo 3 em relação às variáveis complexidade ciclomática e complexidade cognitiva: esse grupo se caracteriza por apresentar maiores valores dessas variáveis, em relação aos demais.

Pelas Figuras B.6, B.7, B.9 e B.12, nota-se que não é possível encontrar uma distinção clara entre os grupos em relação às variáveis classes, más práticas, linhas duplicadas e índice de qualidade.

Já na Figura B.8, percebe-se um destaque no grupo 2 em relação aos demais: esse grupo possui menos linhas de comentário por linha de código.

Analisando as Figuras B.10 e B.11, parece que o grupo 3 tende a ter menores quantidades de funções e de arquivos por linha de código, em relação aos demais grupos.

5.1.2 *Box plots*

Os *box plots* têm uma interpretação muito parecida com os histogramas suavizados.

Pela Figura B.13, fica muito clara a distinção dos grupos em relação à variável resposta. Como já havia sido comentado, o grupo 1 possui menor média e menor variância do número de contribuidores, enquanto o grupo 3 possui maiores valores dessas medidas-resumo e o grupo 2 apresenta valores intermediários dessas medidas.

As variáveis que caracterizam o grupo 3 podem ser vistas nas Figuras B.14, B.15, B.20 e B.21: esse grupo se destaca por possuir maiores valores de complexidade cognitiva e ciclomática por linha de código, e por possuir menores valores de arquivos e funções por linha de código.

A variável que caracteriza o grupo 2 é o número de linhas de comentário por linha de código, que pode ser visto na Figura B.18. Esse grupo é caracterizado por valores menores dessa variável.

Já para as demais variáveis, não há uma distinção clara entre os grupos, isso pode ser visto nas Figuras B.16, B.17, B.19 e B.22

5.1.3 Gráficos de dispersão

Fazendo agora uma análise bidimensional, temos que:

Na Figura B.23, nota-se uma associação negativa entre o número de classes por linha de código e o número de colaboradores, no grupo 2.

Pela Figura B.24, vê-se que nenhum dos grupos parece ter muita associação entre as más práticas de código e o número de contribuidores.

O número de contribuidores parece estar associado de forma negativa com a complexidade cognitiva e com a complexidade ciclomática no grupo 2 (Figuras B.25 e B.27).

Pela Figura B.28, observa-se uma possível associação positiva no grupo 3 entre o número de contribuidores e a quantidade de linhas duplicadas por linha de código.

Nota-se uma associação negativa entre o número de contribuidores e a quantidade de arquivos por linha de código no grupo 3 (Figura B.29).

Nas Figuras B.26, B.30 e B.31, vê-se que a quantidade de funções por linha de código, a quantidade de linhas de comentário por linha de código e o índice de qualidade não parecem estar associados com o número de contribuidores em nenhum dos grupos.

5.1.4 Correlações

Na Figura B.32, pode-se ver a correlação entre as variáveis levando em conta todos os 32 projetos. É possível notar que as correlações entre as variáveis explicativas e resposta são muito baixas (com um máximo de 0,23, em módulo).

Fazendo os correlogramas por grupo, observa-se que as correlações aumentam bastante, chegando a um máximo de 0,43 (em módulo).

Pelas Figuras B.33, B.34 e B.35, temos que as variáveis explicativas mais correlacionadas com a variável resposta em cada grupo são:

- No grupo 1: quantidade de classes por linha de código, quantidade de arquivos por linha de código, e quantidade de más práticas por linha de código.
- No grupo 2: complexidade cognitiva e complexidade ciclomática, e número de classes por linha de código.
- No grupo 3: quantidade de arquivos, linhas duplicadas e linhas comentadas por linha de código.

5.1.5 Gráficos de perfis

Pela Figura B.36, pode-se ver um destaque do grupo 3 nas variáveis contribuidores e complexidade cognitiva.

Nas Figuras B.37, B.38 e B.39, pode-se ver o comportamento distinto e desbalanceado dos projetos mesmo dentro de um mesmo grupo.

5.2 Análise descritiva de perfis

Na Figura B.40 pode-se ver que não existe uma tendência na variância da variável resposta, ou seja, ela parece se manter constante ao longo do tempo. O mais importante

a ser analisado nesse gráfico são os instantes que possuem maior número de observações (Tabela A.3), então, por mais que os instantes iniciais possuam variâncias destoantes das demais, eles não merecem muita atenção por terem poucas observações.

Pelas Figuras B.41 e B.42, pode-se notar que há correlação entre a quantidade de contribuidores de um mesmo projeto em diferentes instantes.

Pelas Figuras B.43 e B.44, tem-se que os projetos apresentam comportamentos distintos em relação à variável de interesse, principalmente em relação à variabilidade.

6. Análise inferencial

Pelo fato de se ter várias medidas associadas a um mesmo projeto ao longo do tempo, e de maneira desbalanceada, decidiu-se utilizar o modelo linear misto gaussiano (Singer et al., 2018), considerando:

- Variável resposta: quantidade de colaboradores distintos no semestre
- Efeito aleatório: efeito de cada projeto
- Variáveis explicativas**:** data, más práticas, linhas de comentário, complexidade ciclomática, linhas duplicadas, arquivos, funções, classes, índice de qualidade

A variável complexidade cognitiva não será incluída no modelo por apresentar alta correlação com a variável complexidade ciclomática (uma variável é calculada a partir da outra).

Inicialmente foi considerado o modelo misto gaussiano com efeito aleatório de projeto e erros condicionais independentes. Após análise de resíduos (Singer et al., 2017), constatou-se no gráfico de Lesaffre-Verbeke que alguns projetos tinham estruturas de covariâncias inadequadas. Para esses projetos, foi considerada uma variância diferente no erro condicional. A partir de nova análise de resíduos, constatou-se no gráfico de autocorrelação a presença de correlação serial, sugerindo processo autorregressivo para o erro aleatório. Considerou-se o modelo misto gaussiano com efeito aleatório de projeto, alguns projetos com variâncias diferentes e o erro

condicional com estrutura de correlação autorregressiva de ordem 1 (AR(1)). O ajuste dos modelos foi feito usando o pacote nlme do R com a função lme.

Pelas Figuras B.45 e B.46, nota-se que estrutura de covariância sugerida está adequada para o modelo saturado (contém todas as variáveis explicativas) ajustado. Na Figura B.47, pode-se ver que a suposição de distribuição normal adotada para os efeitos aleatórios também está adequada. Já na Figura B.48, verifica-se que as suposições do modelo misto gaussiano adotado estão satisfeitas.

Verificada a adequabilidade do modelo ajustado, foram aplicados os métodos *backward* e *stepwise* para a seleção de variáveis explicativas do modelo. Na Tabela A.4 pode-se ver as variáveis que saíram do modelo, uma a uma, e seus respectivos p-valores, quando aplicado o método *backward*. Já na tabela A.5 estão as variáveis que entraram no modelo, uma a uma, e seus respectivos p-valores, quando aplicado o método *stepwise*. A escolha entre os dois modelos foi baseada em critérios de informação de Akaike (AIC) e bayesiano (BIC), além da função de verossimilhança. O modelo mais adequado é aquele que possui menor AIC e BIC, e maior valor do logaritmo da função de verossimilhança. Na Tabela A.6 é possível ver que o modelo selecionado pelo método *stepwise* mostrou ser o mais indicado. Desse modo, serão retiradas do modelo as seguintes variáveis: data, linhas de comentário, complexidade ciclomática, linhas duplicadas, arquivos, classes e índice de qualidade.

Dessa maneira, para o modelo final foram mantidas apenas as seguintes variáveis de efeitos fixos: más práticas e funções; apenas com efeitos principais.

A Tabela A.7 contém as estimativas, erros padrões e valores-p dos efeitos fixos obtidos pelo modelo. Nota-se que as duas variáveis de efeitos fixos são significantes a um nível de 6%. A Tabela A.8 mostra as estimativas das variâncias para cada grupo de projetos; nota-se que elas condizem com a Figura B.44.

É possível notar que o coeficiente estimado associado à variável más práticas é negativo (-22,17), indicando que existe uma associação negativa entre o número de contribuidores e a quantidade de más práticas por linha de código. Já o coeficiente estimado associado à variável funções é positivo (14,55), indicando que existe uma

associação positiva entre o número de contribuidores e a quantidade de funções por linha de código.

Interpretando esses coeficientes estimados, tem-se que: com o acréscimo de 0,1 na quantidade de más práticas por linha de código, é esperado que o número de contribuidores caia, em média, em 2,22 (mantendo fixas as demais variáveis). Além disso, com o acréscimo de 0,1 na quantidade de funções por linha de código, é esperado que o número de contribuidores aumente, em média, em 1,45 (mantendo fixas as demais variáveis).

7. Conclusões

7.1. Caracterização dos grupos

Pela análise descritiva dado o agrupamento, tem-se que cada grupo é caracterizado da seguinte maneira:

Grupo 1

O grupo 1 é caracterizado por projetos com menor período duração (média de 6,18 semestres), e menor número de contribuidores (média de 2,97 por semestre).

Dadas essas características, temos que nesse grupo, o número de contribuidores por semestre está mais associado às variáveis:

- quantidade de arquivos por linha de código (correlação de -0,16)
- complexidade ciclomática (correlação de 0,18)
- complexidade cognitiva (correlação de 0,21)

Grupo 2

O grupo 2 é caracterizado por projetos com período médio de duração (média de 11,73 semestres) e número médio de contribuidores (média de 10,47 por semestre).

Além disso, esse grupo se destaca por projetos com menor quantidade de linhas de comentário por linha de código.

Dadas essas características, temos que nesse grupo, o número de contribuidores por semestre está mais associado às variáveis:

- quantidade de classes por linha de código (correlação de -0,27)
- complexidade cognitiva (correlação de -0,27)
- complexidade ciclomática (correlação de -0,3)

Grupo 3

O grupo 3 é caracterizado por projetos com maior período de duração (média de 12,25 semestres), e maior número de contribuidores (média de 18,51 por semestre).

Além disso, esse grupo se destaca por projetos com maior complexidade cognitiva e ciclomática, e menor quantidade de arquivos e funções por linha de código.

Dadas essas características, temos que nesse grupo, o número de contribuidores por semestre está mais associado às variáveis:

- linhas de comentário por linha de código (correlação de 0,24)
- linhas de duplicadas por linha de código (correlação de 0,37)
- quantidade de arquivos por linha de código (correlação de -0,43)

7.2. Modelo ajustado

Pelo modelo ajustado, tem-se que a quantidade de colaboradores distintos no semestre é explicada pela densidade de más práticas no código (associação negativa), e pela densidade de funções no código (associação positiva).

Não foi encontrada associação significativa entre as demais métricas de código consideradas e a quantidade de contribuidores de um projeto.

Além disso, o efeito da variável tempo (da maneira que foi definida) também não foi significativo para o modelo, indicando que a quantidade média de contribuidores se mantém constante ao longo do tempo.

APÊNDICE A

Tabelas

Tabela A.1 Tabela comparativa entre os grupos, em relação ao número de observações por projeto

	Grupo 1	Grupo 2	Grupo 3
Mínimo de Observações por projeto	4,00	6,00	6,00
Média de Observações por projeto	6,18	11,73	12,25
Mediana de Observações por projeto	5,00	11,00	12,00
Máximo de Observações por projeto	14,00	22,00	19,00
Desvio Padrão de Observações por projeto	2,88	4,73	5,56
Total de Observações no grupo	105,00	129,00	49,00
Número de Projetos	17,00	11,00	4,00

Tabela A.2 Tabela comparativa entre os grupos, em relação ao número de contribuidores por semestre

	Grupo 1	Grupo 2	Grupo 3
Mínimo	1,00	1,00	1,00
1 Quartil	1,00	3,00	3,00
Mediana	2,00	8,00	14,00
Média	2,97	10,47	18,51
3 Quartil	3,00	16,00	34,00
Máximo	17,00	35,00	49,00

Tabela A.3 Tabela do número de observações em cada tempo

Tempo	Número de observações
1	1
2	1
3	2
4	4
5	4
6	5
7	8
8	9
9	12
10	14
11	18
12	18
13	18
14	18
15	17
16	15
17	18
18	19
19	22
20	21
21	20
22	19

Tabela A.4 Variáveis retiradas do modelo quando aplicado o método *backward,* utilizando 0,05 como corte de saída

Passo	Varável	valor-p
1	Classes	0,9158
2	Complexidade	0,8131
3	Data	0,6943
4	Linhas duplicadas	0,5677
5	Arquivos	0,4612
6	Funções	0,1399
7	Más práticas	0,1284

Tabela A.5 Variáveis incluídas no modelo quando aplicado o método *stepwise,* utilizando 0,15 como corte de saída, e 0,15 como corte de entrada

Passo	Varável	valor-p
1	Más Práticas	0,014
2	Funções	0,060

Tabela A.6 Comparação entre o AIC e BIC dos modelos selecionados pelos métodos *stepwise* e *backward*

Método	AIC	BIC	Log Verossimilhança
stepwise	1500,104	1558,26	-734,05
backward	1507,589	1565,74	-737,79

Tabela A.7 Estimativa, erro padrão e p-valor dos efeitos fixos do modelo final, apenas com efeitos principais, com respectivos intervalos de confiança.

	Estimativa	Erro Padrão	valor-p	Intervalo confiança 95%
Intercepto	1,91	0,58	0,001	[0,77; 3,06]
Más práticas	-22,17	7,37	0,002	[-36,71; -7,64]
Funções	14,55	7,70	0,060	[-0,61; 29,71]

Tabela A.8 Estimativas das variâncias para cada grupo de projetos, com respectivos intervalos de confiança.

Projetos	Variância estimada	Intervalo confiança 95%
13	14,30	[7,95; 25,71]
5	11,92	[8,37; 16,98]
9, 11, 18, 25	7,99	[6,38; 10,00]
4, 10	6,73	[4,88; 9,27]
17, 20	5,54	[4,03; 7,61]
12	4,86	[3,27; 7,22]
23	2,91	[1,87; 4,53]
31	1,88	[0,92; 3,85]
7	1,65	[0,86; 3,13]
14	1,59	[0,89; 2,86]
Demais	1,00	

APÊNDICE B

Figuras

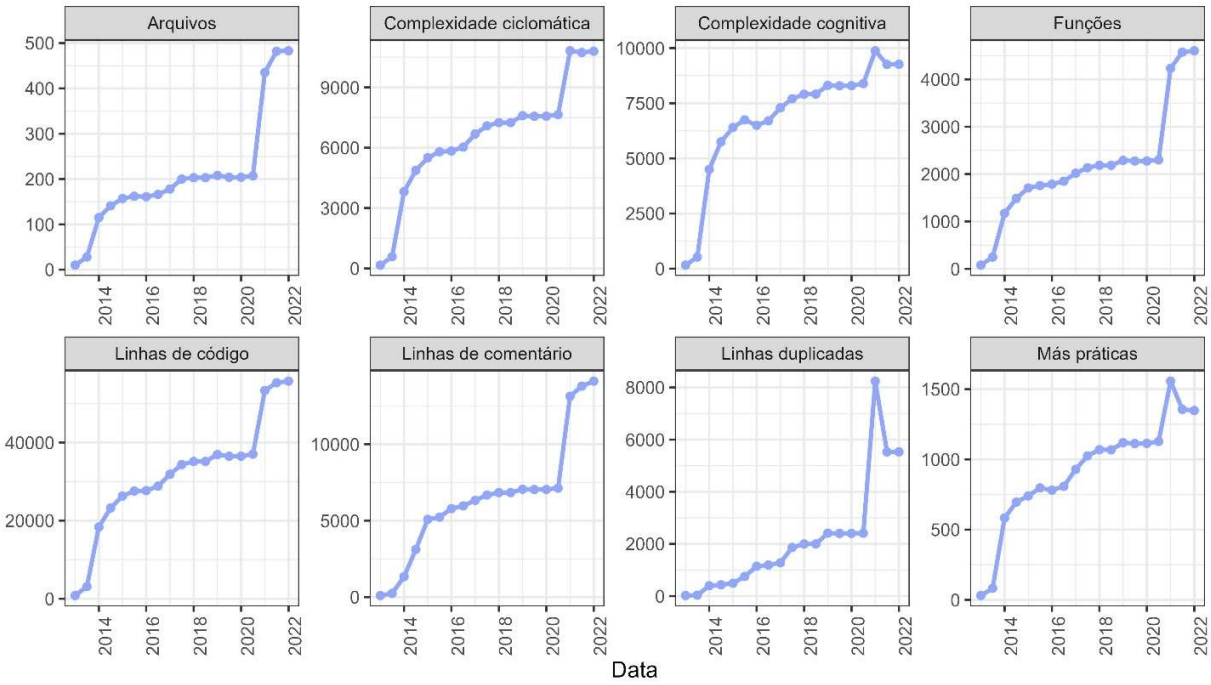

Figura B.1 Gráficos de linhas do projeto "beego:beego" antes da padronização

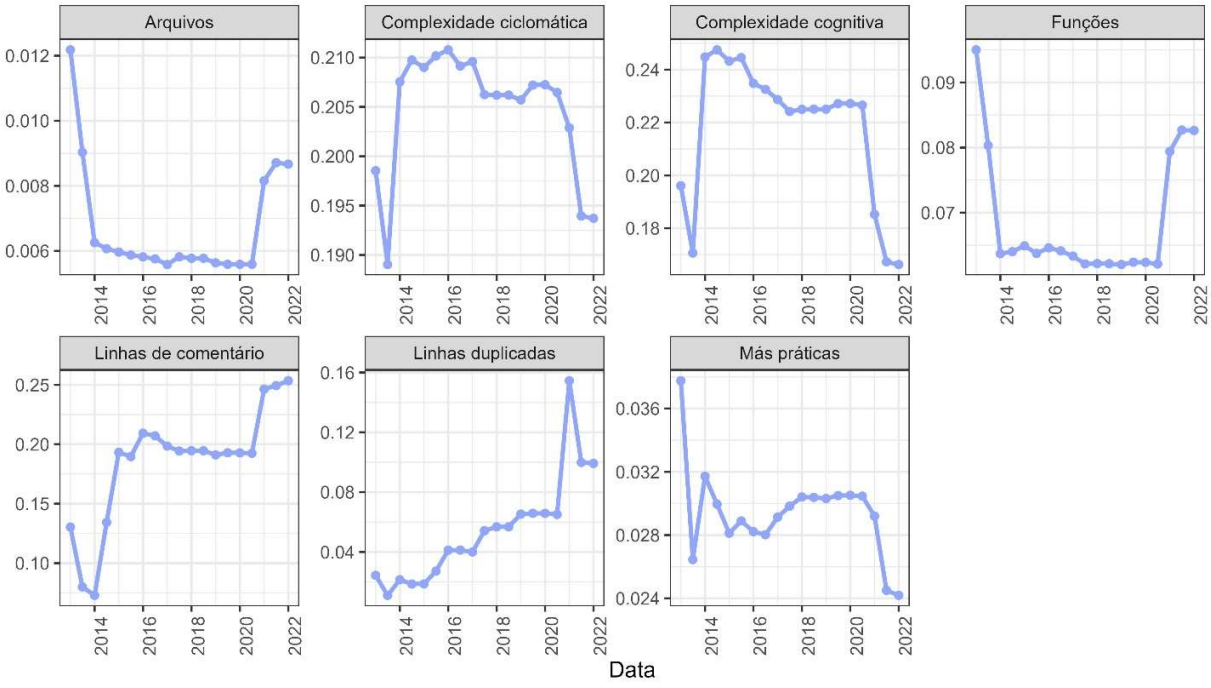

Figura B.2 Gráficos de linhas do projeto "beego:beego" depois da padronização

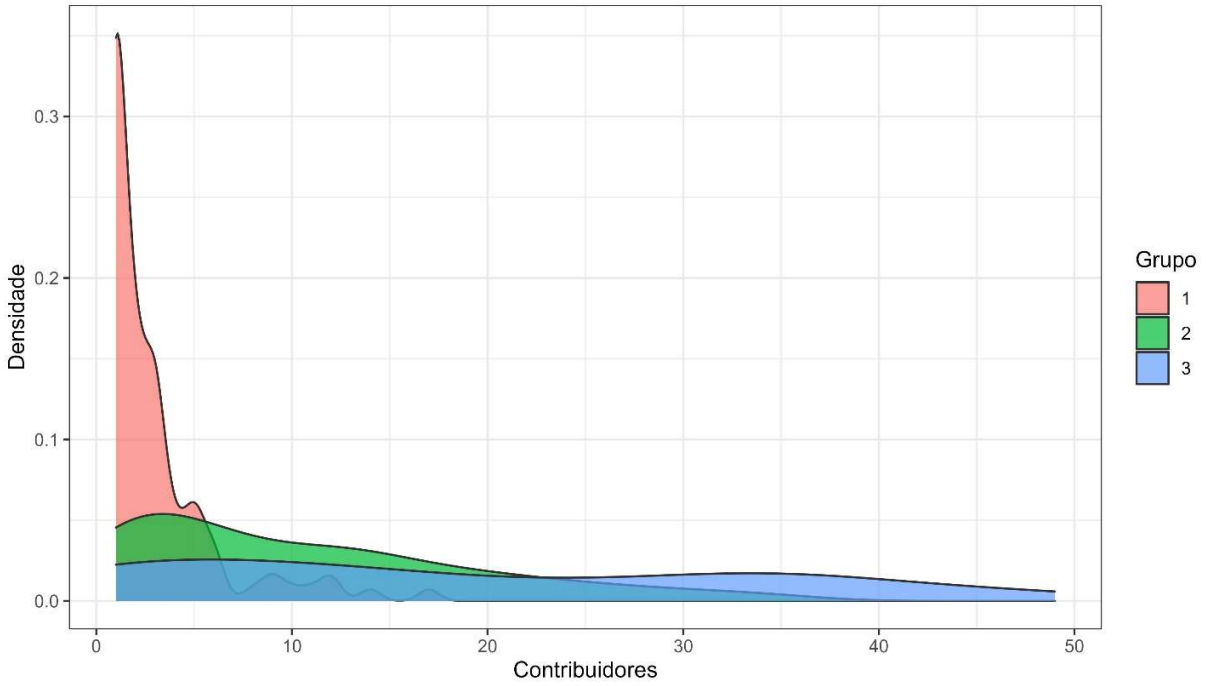

Figura B.3 Histograma suavizado da quantidade de contribuidores por semestre, segundo os grupos

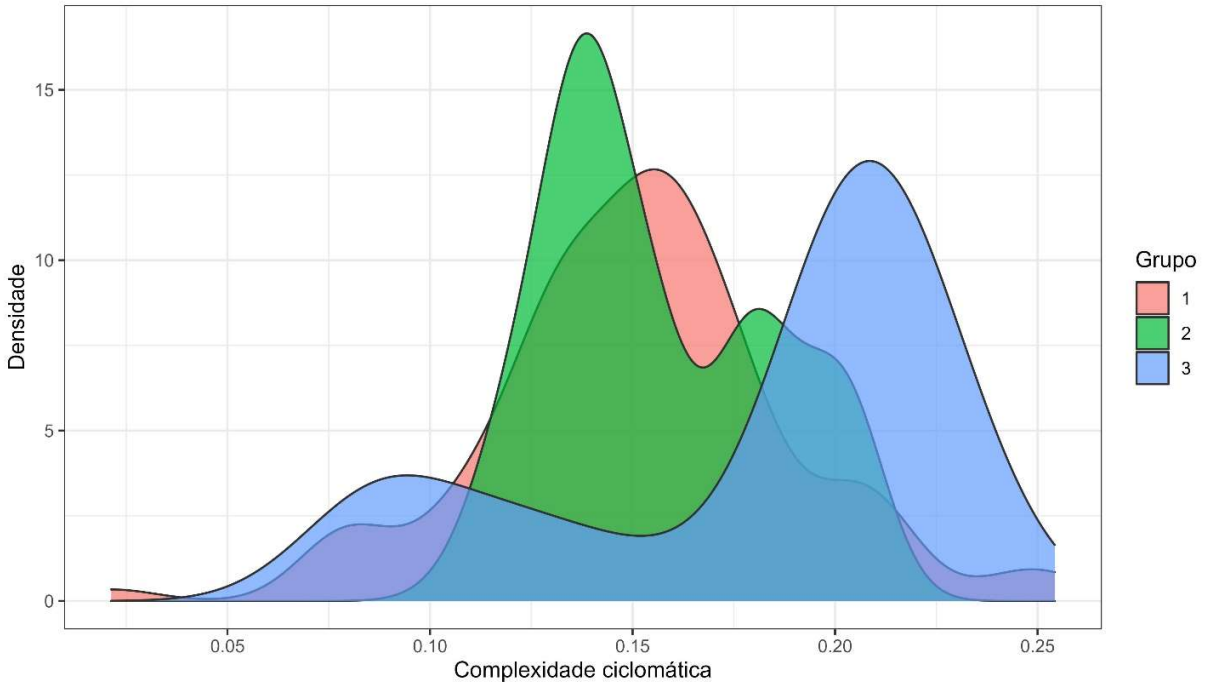

Figura B.4 Histograma suavizado da complexidade ciclomática (padronizada pelo número de linhas de código), segundo os grupos

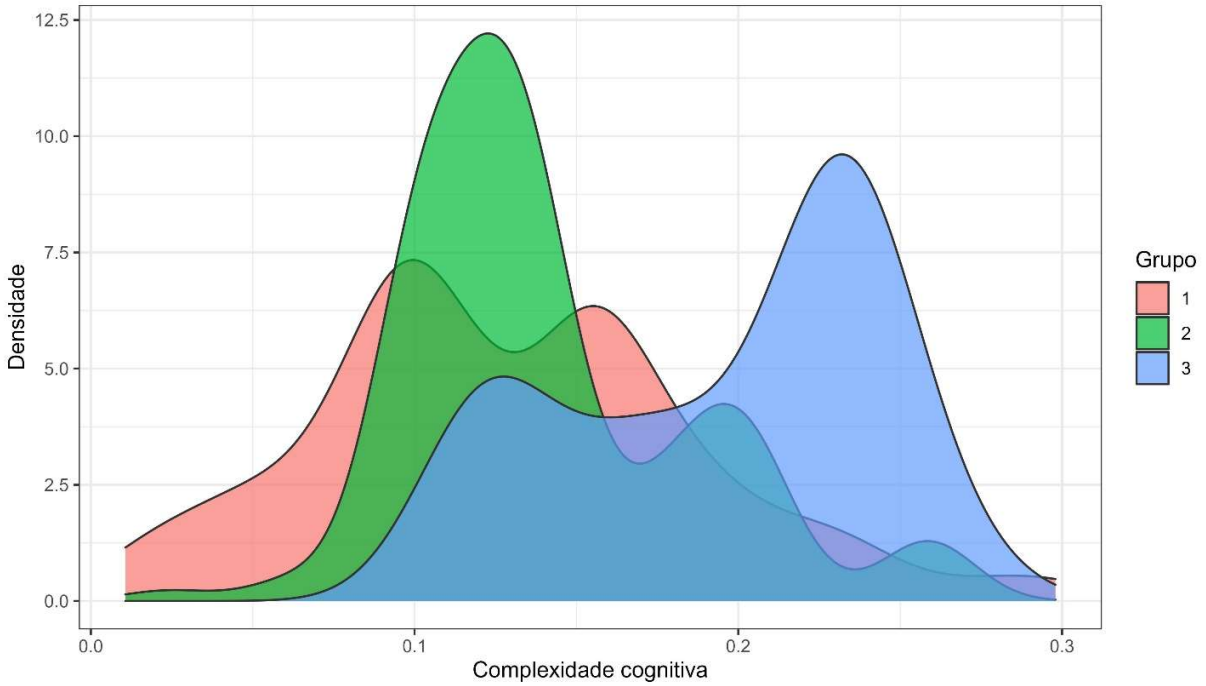

Figura B.5 Histograma suavizado da complexidade cognitiva (padronizada pelo número de linhas de código), segundo os grupos

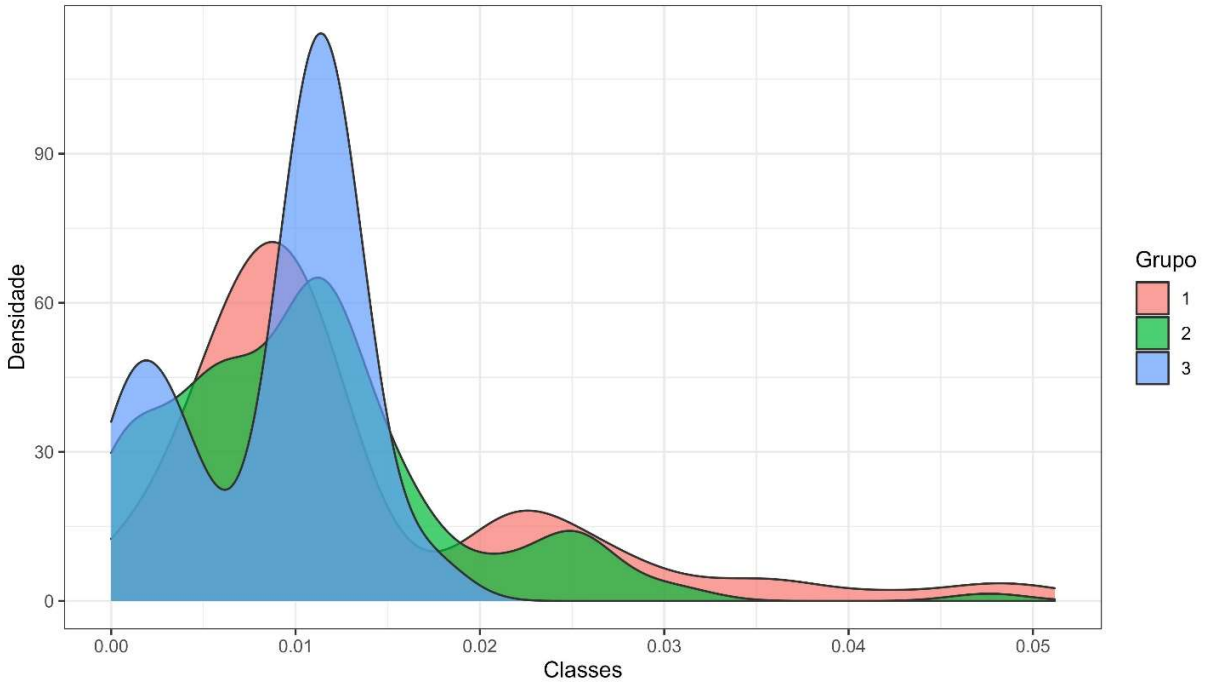

Figura B.6 Histograma suavizado da quantidade de classes por linha de código, segundo os grupos

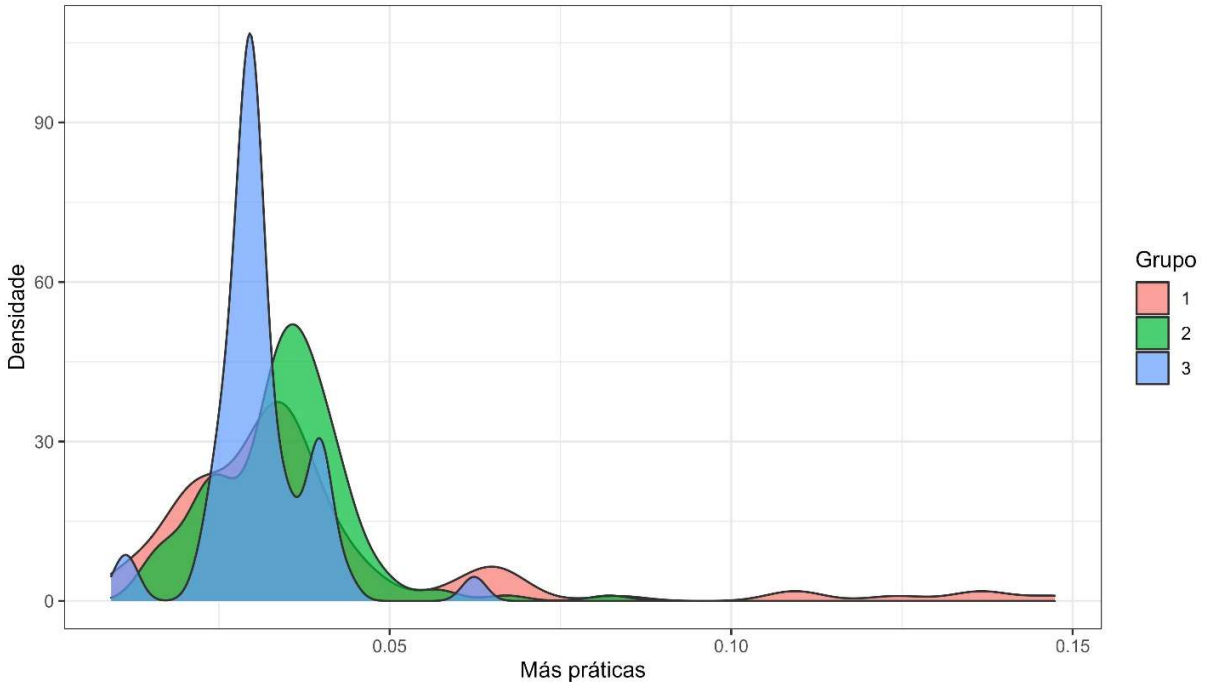

Figura B.7 Histograma suavizado da quantidade de más práticas por linha de código, segundo os grupos

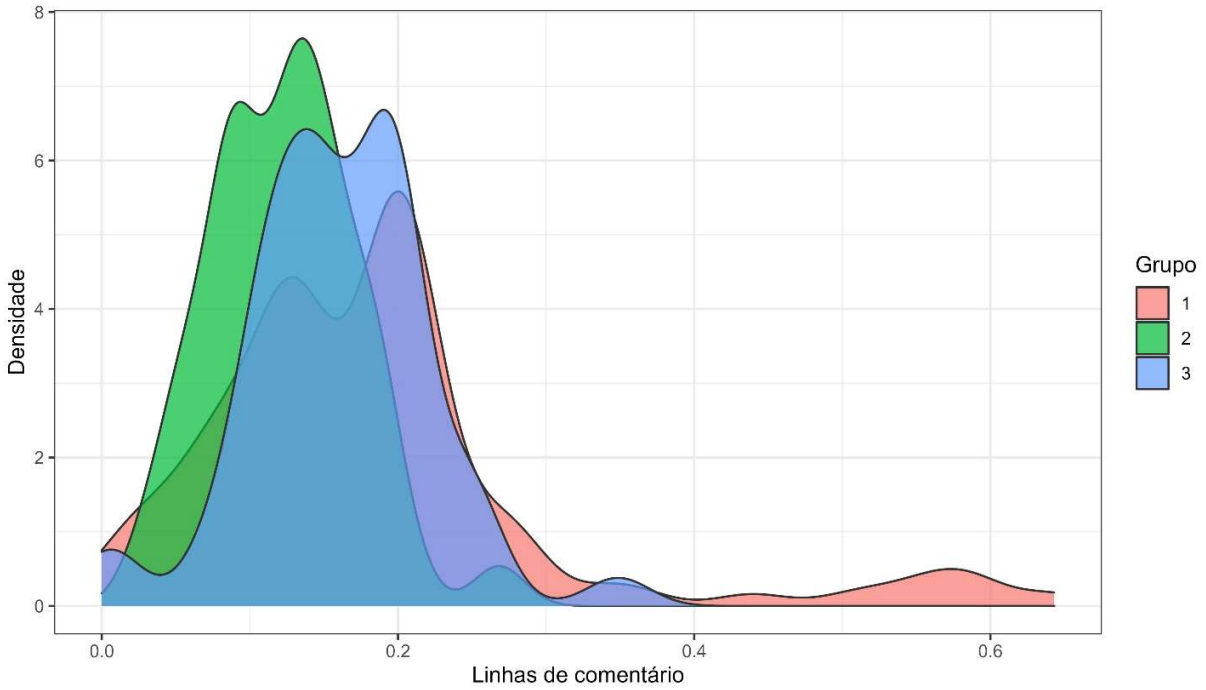

Figura B.8 Histograma suavizado da quantidade de linhas de comentário por linha de código, segundo os grupos

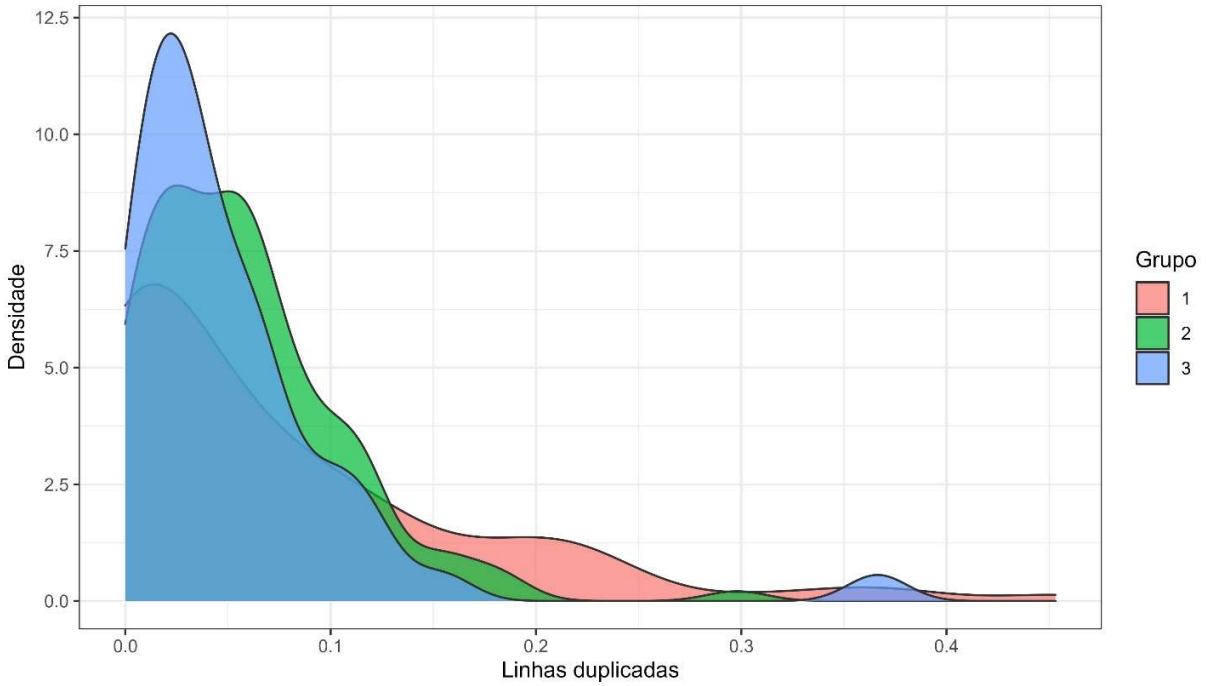

Figura B.9 Histograma suavizado da quantidade de linhas de código duplicadas por linha de código, segundo os grupos

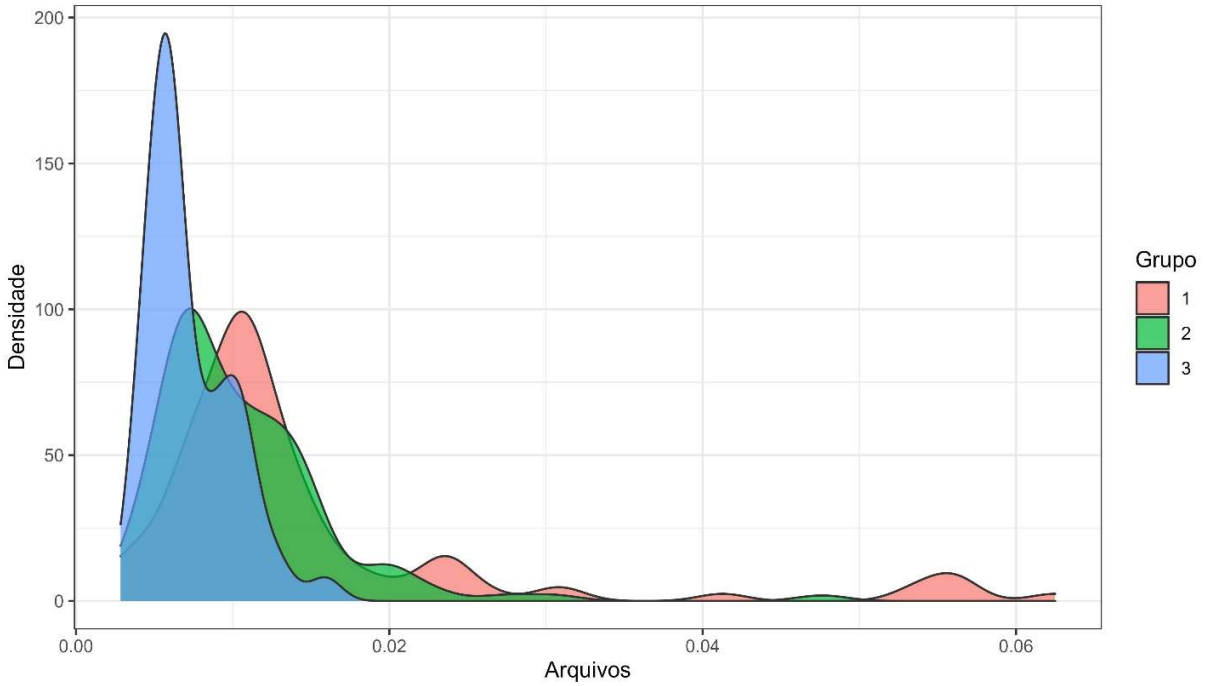

Figura B.10 Histograma suavizado da quantidade de arquivos por linha de código, segundo os grupos

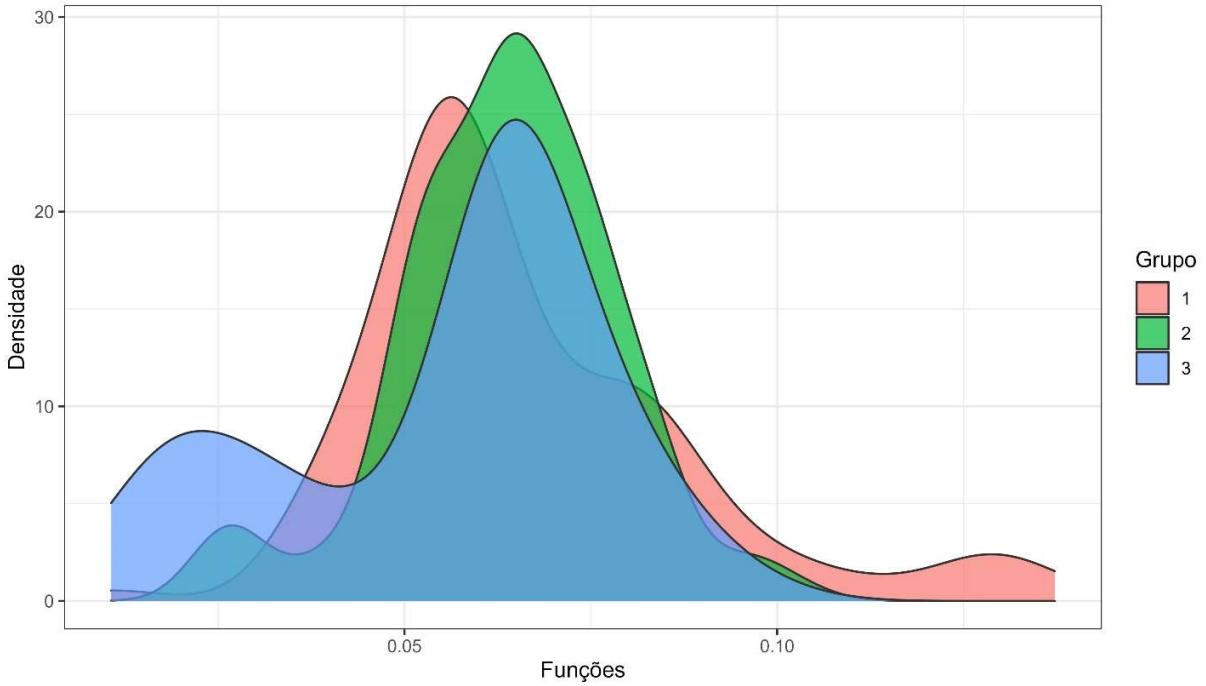

Figura B.11 Histograma suavizado da quantidade de funções por linha de código, segundo os grupos

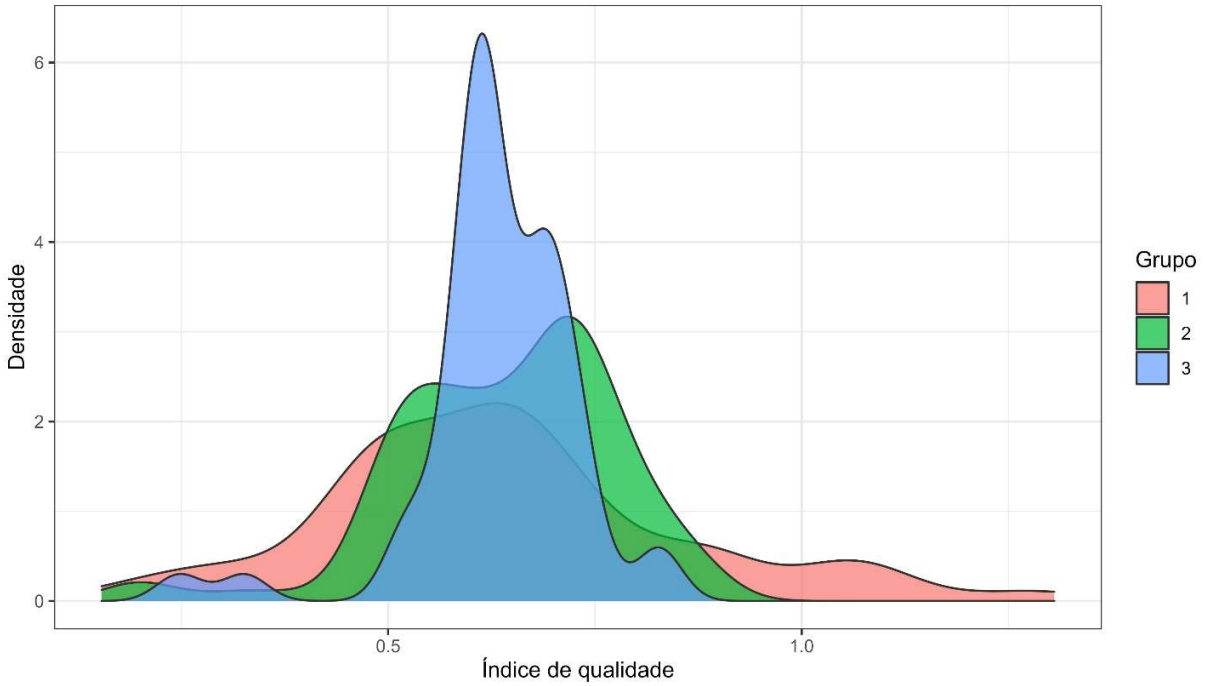

Figura B.12 Histograma suavizado do índice de qualidade (padronizado pelo número de linhas de código), segundo os grupos

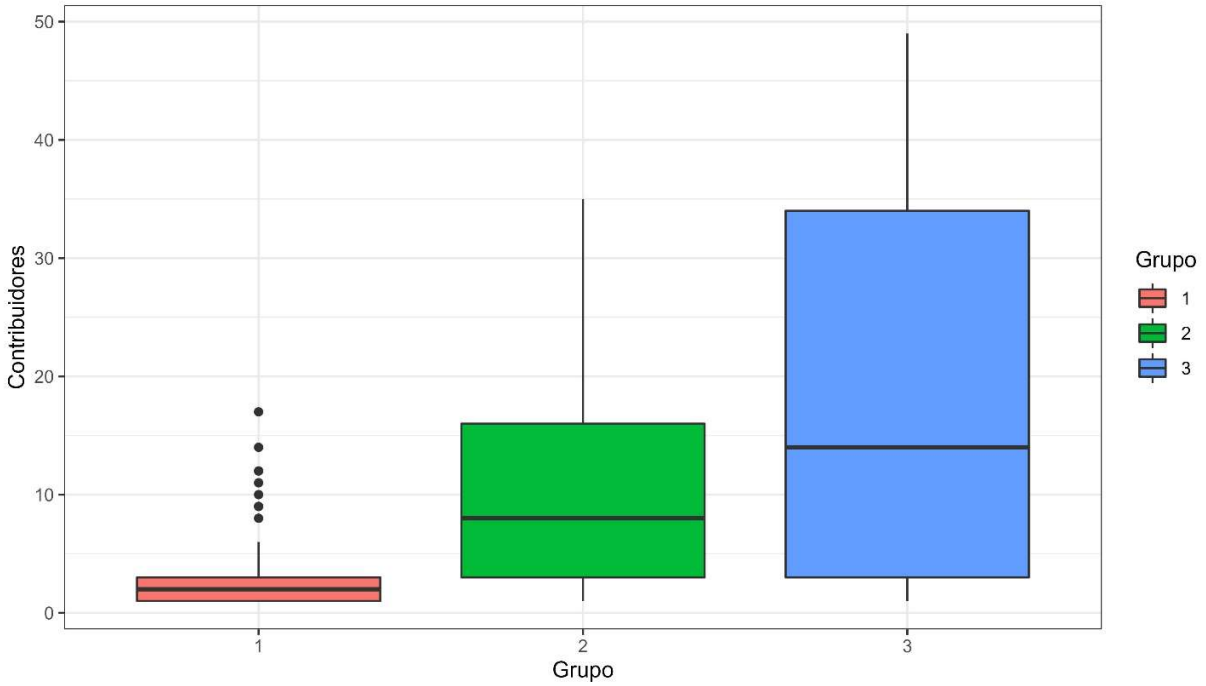

Figura B.13 *Box plots* da quantidade de contribuidores por semestre, segundo os grupos

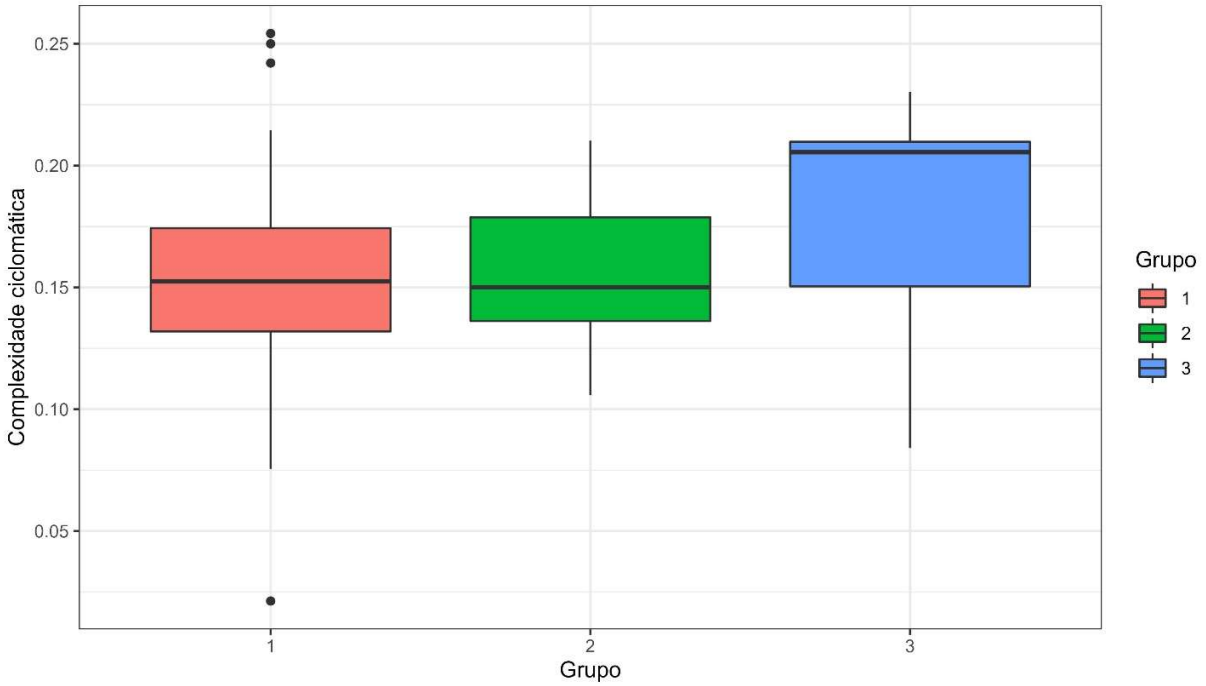

Figura B.14 *Box plots* da complexidade ciclomática (padronizada pelo número de linhas de código), segundo os grupos

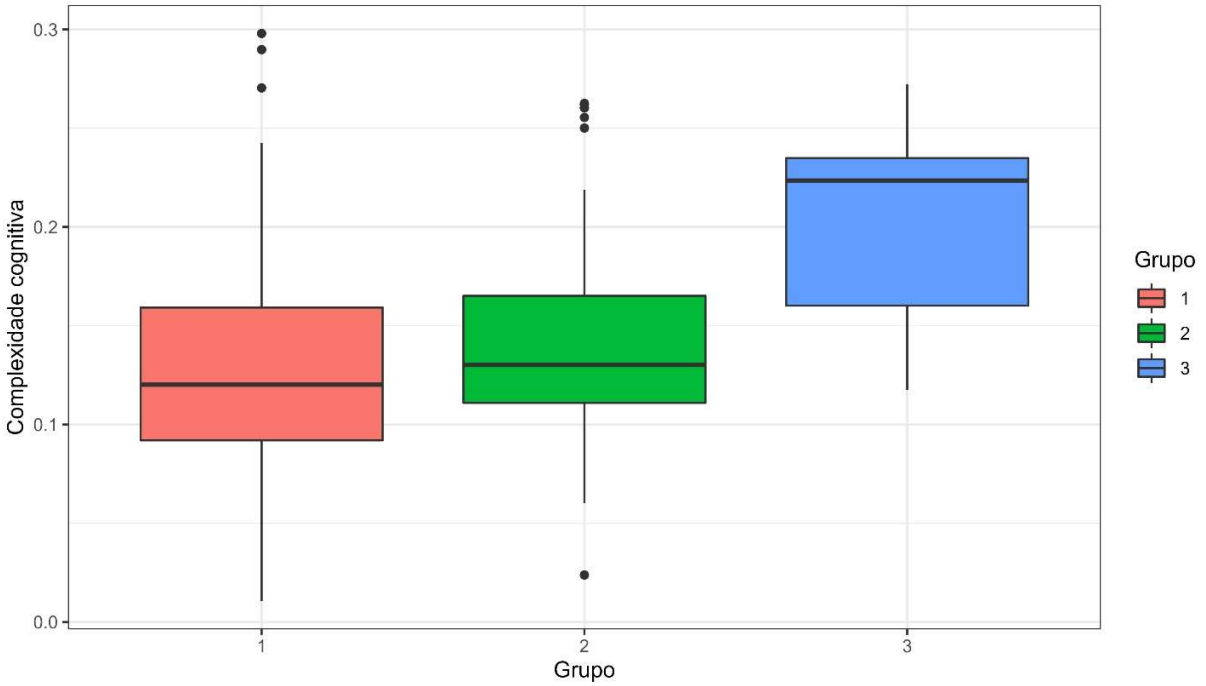

Figura B.15 Box plots da complexidade cognitiva (padronizada pelo número de linhas de código), segundo os grupos

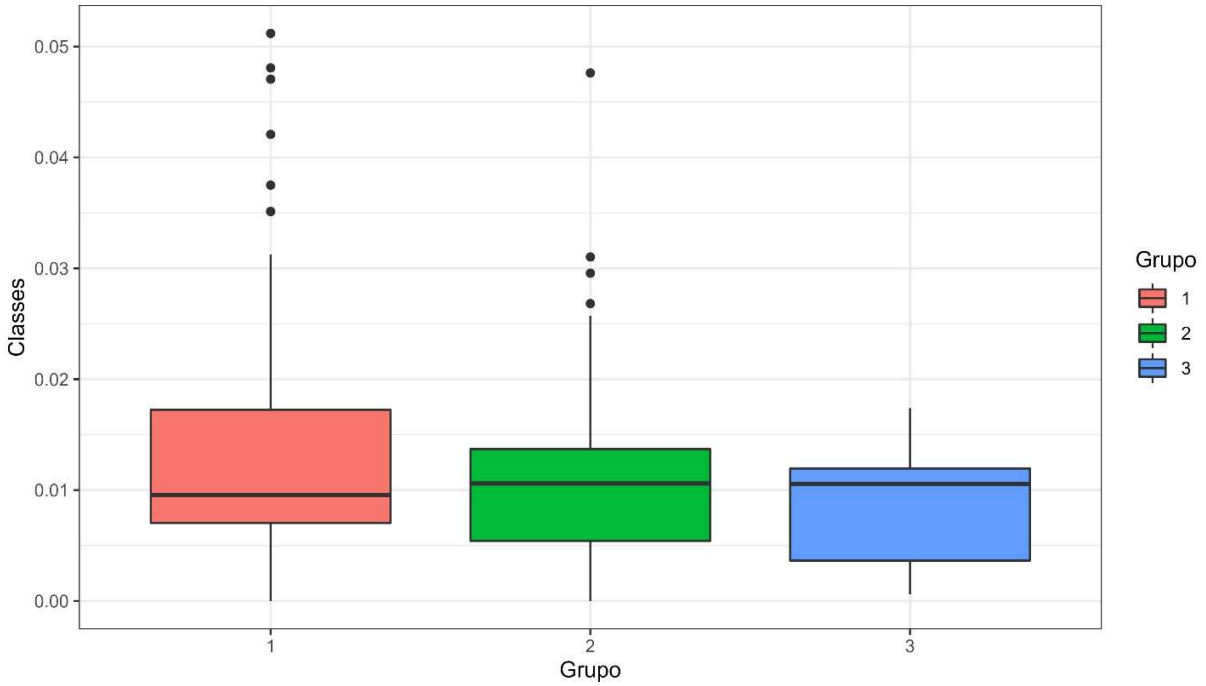

Figura B.16 *Box plots* da quantidade de classes por linha de código, segundo os grupos

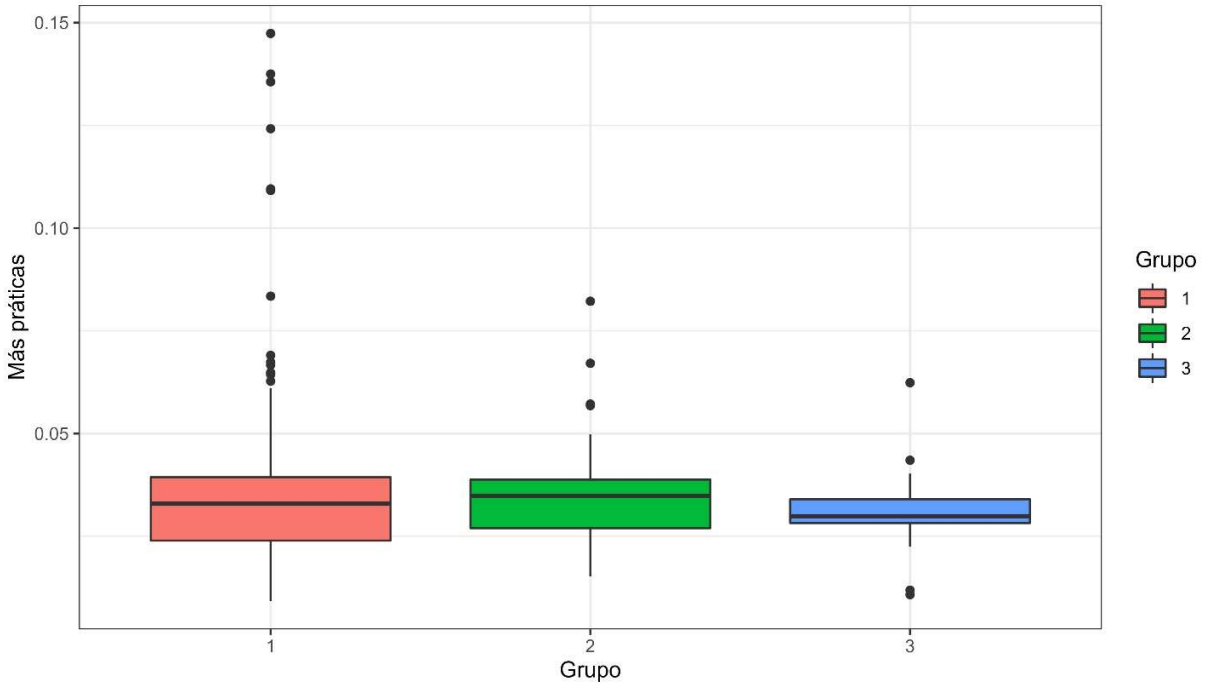

Figura B.17 *Box plots* da quantidade de más práticas por linha de código, segundo os grupos

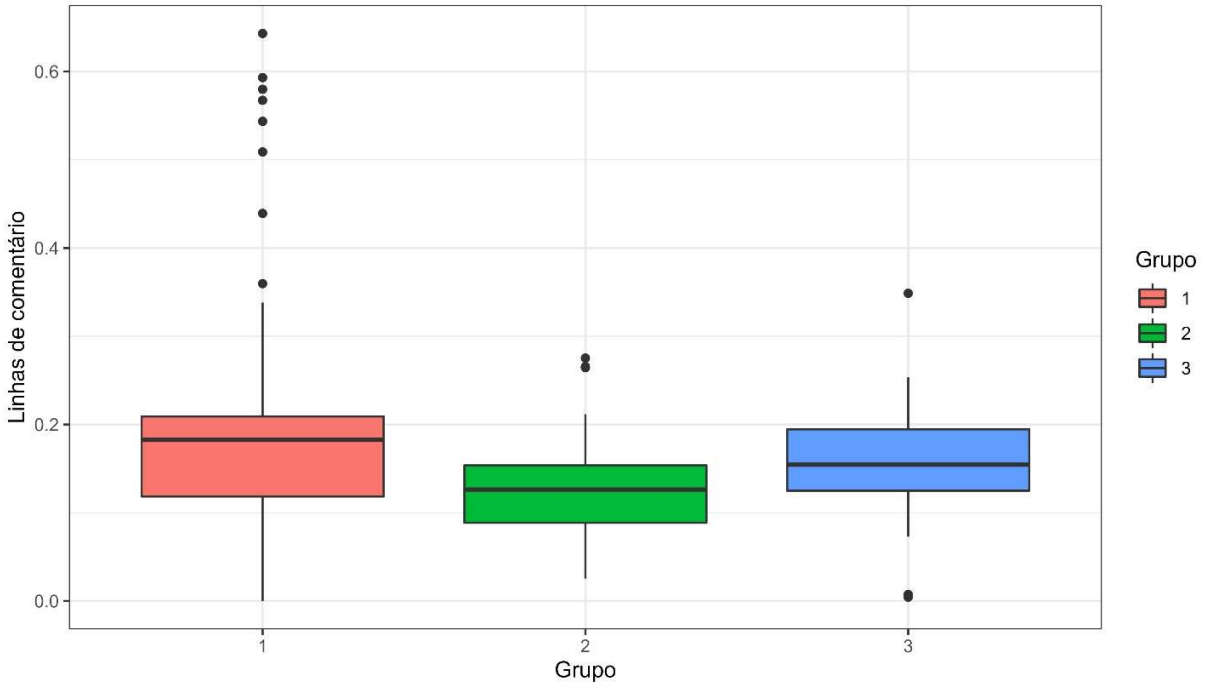

Figura B.18 *Box plots* da quantidade de linhas de comentário por linha de código, segundo os grupos

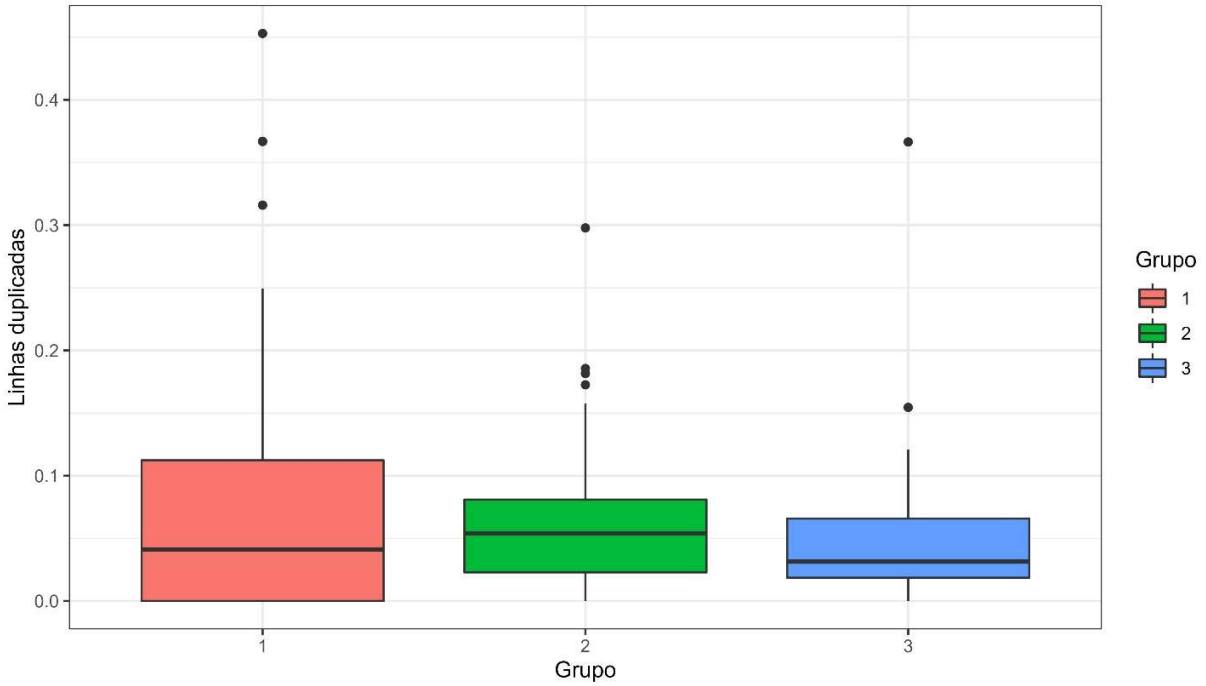

Figura B.19 *Box plots* da quantidade de linhas de código duplicadas por linha de código, segundo os grupos

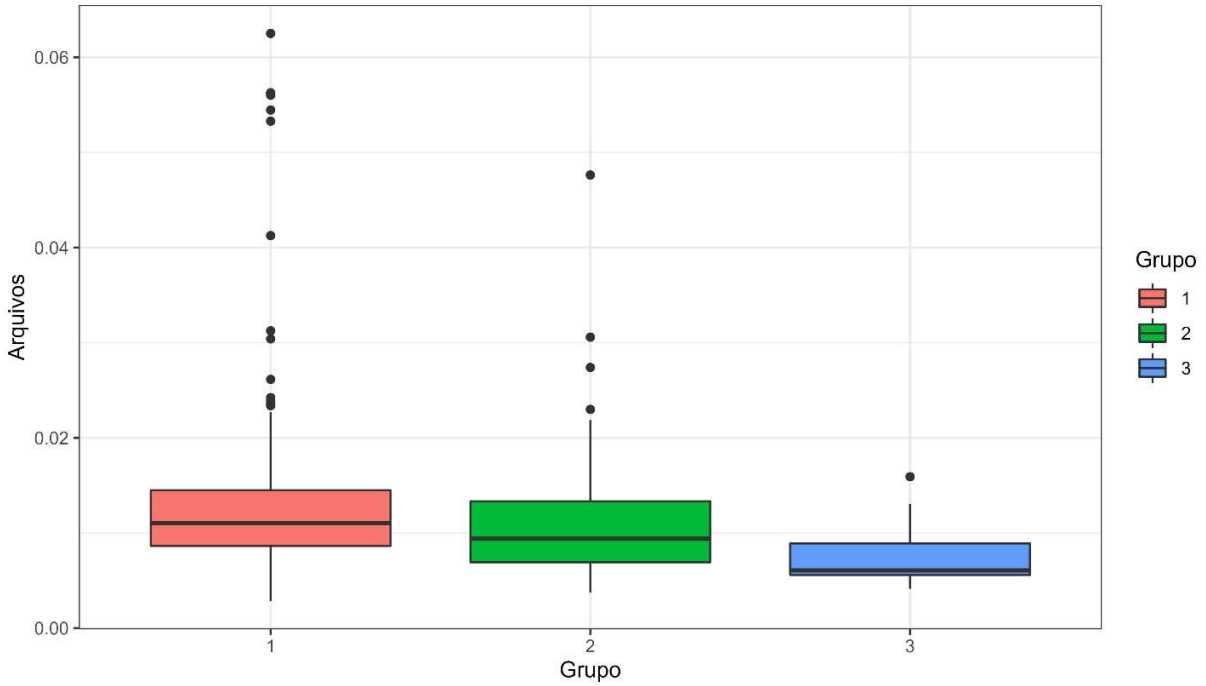

Figura B.20 *Box plots* da quantidade de arquivos por linha de código, segundo os grupos

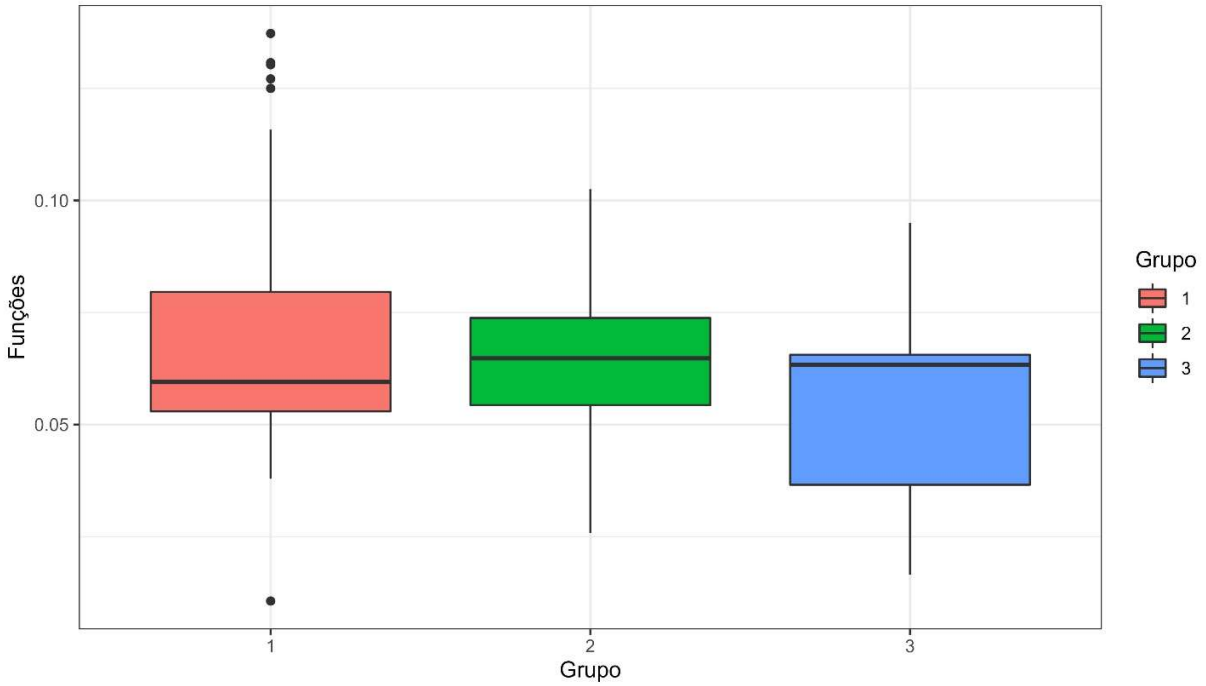

Figura B.21 *Box plots* da quantidade de funções por linha de código, segundo os grupos

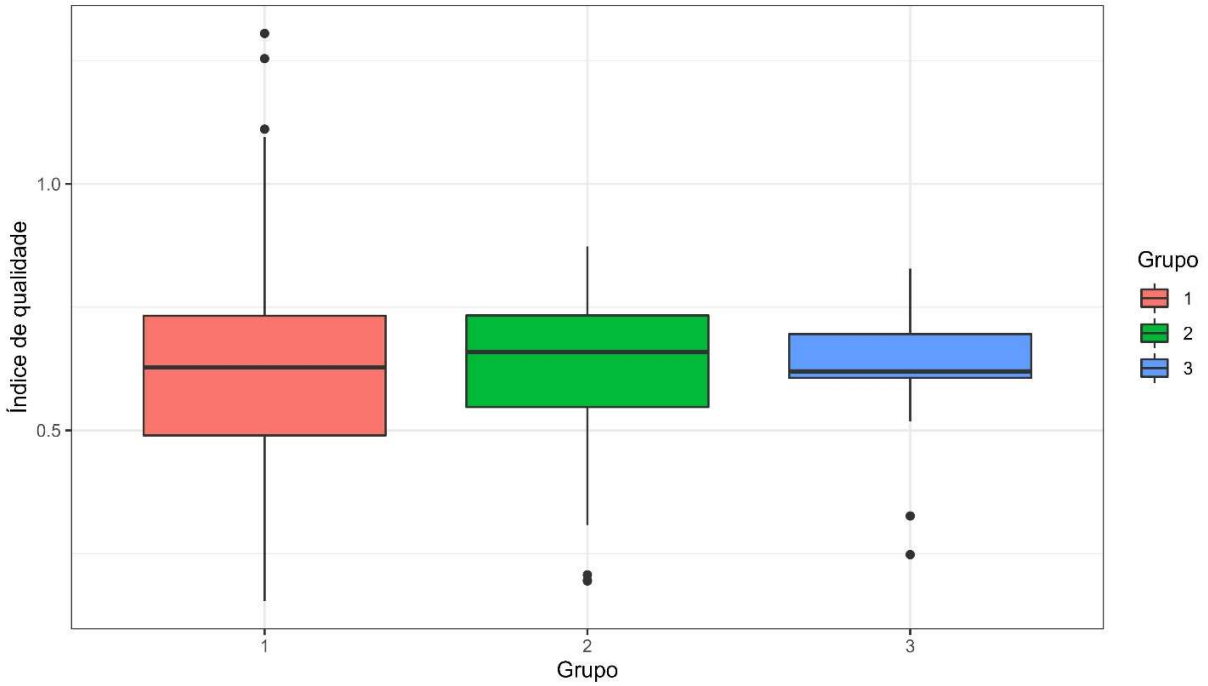

Figura B.22 *Box plots* do índice de qualidade (padronizado pelo número de linhas de código), segundo os grupos

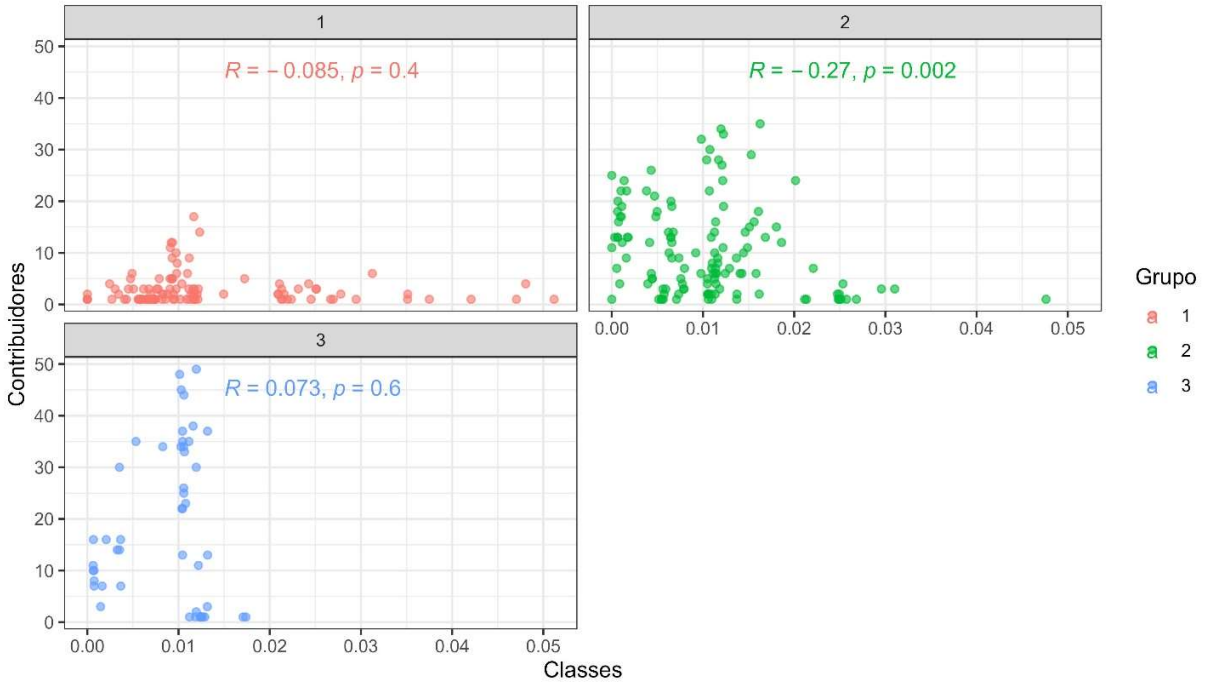

Figura B.23 Gráficos de dispersão entre a quantidade de contribuidores por semestre e a quantidade de classes por linha de código, segundo os grupos

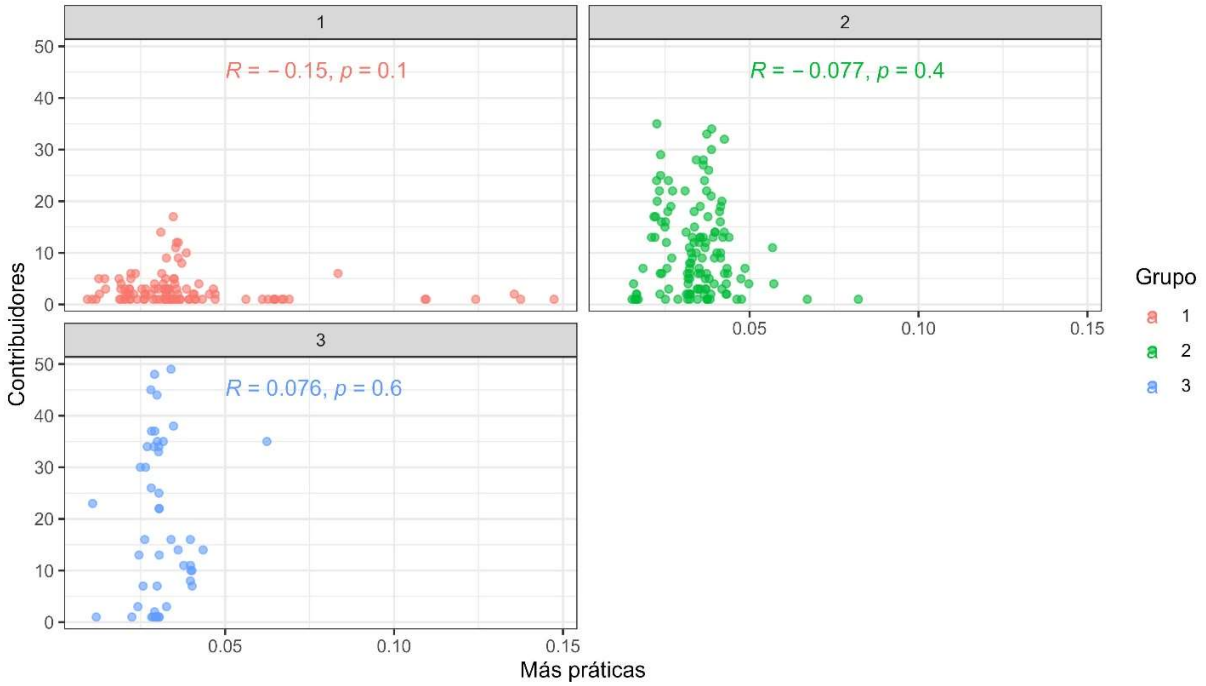

Figura B.24 Gráficos de dispersão entre a quantidade de contribuidores por semestre e a quantidade de más práticas por linha de código, segundo os grupos

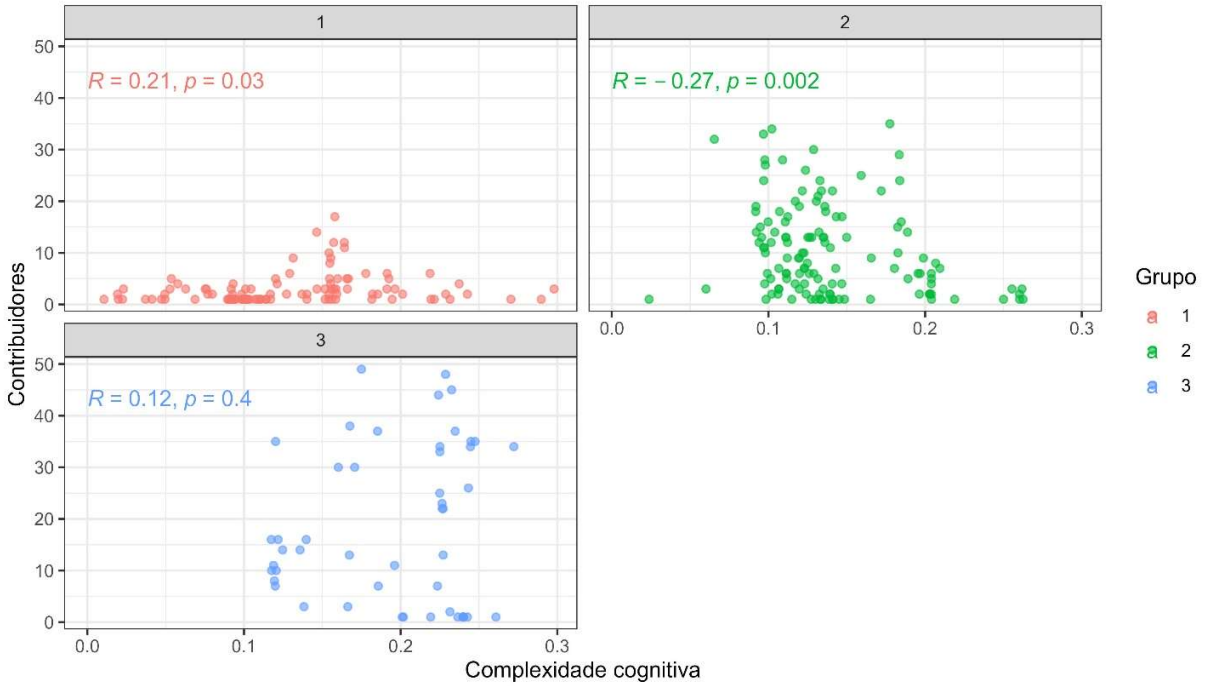

Figura B.25 Gráficos de dispersão entre a quantidade de contribuidores por semestre e a complexidade cognitiva (padronizada pelo número de linhas de código), segundo os grupos

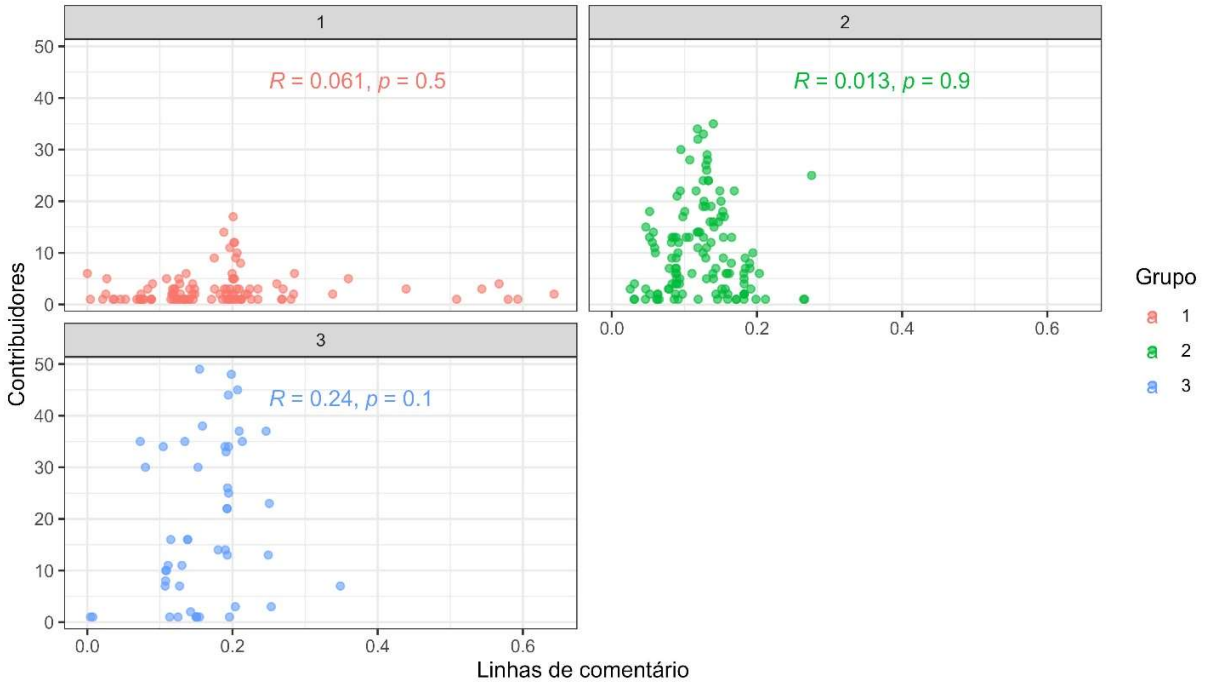

Figura B.26 Gráficos de dispersão entre a quantidade de contribuidores por semestre e a quantidade de linhas de comentário por linha de código, segundo os grupos

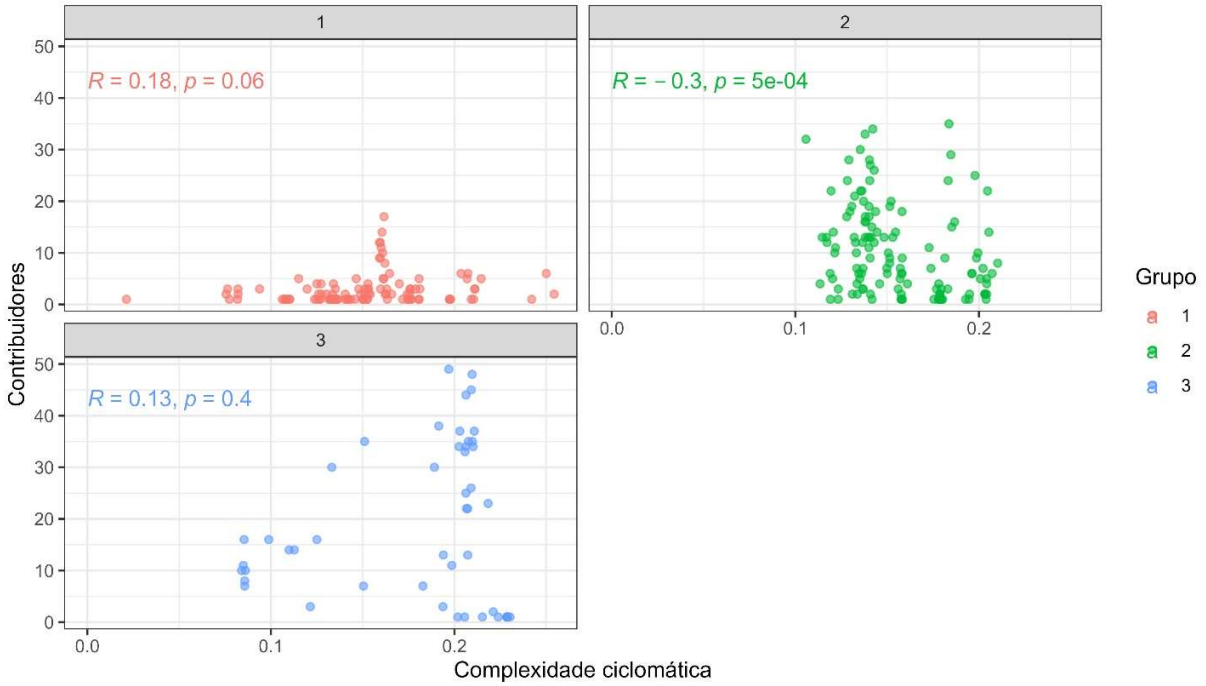

Figura B.27 Gráficos de dispersão entre a quantidade de contribuidores por semestre e a complexidade ciclomática (padronizada pelo número de linhas de código), segundo os grupos

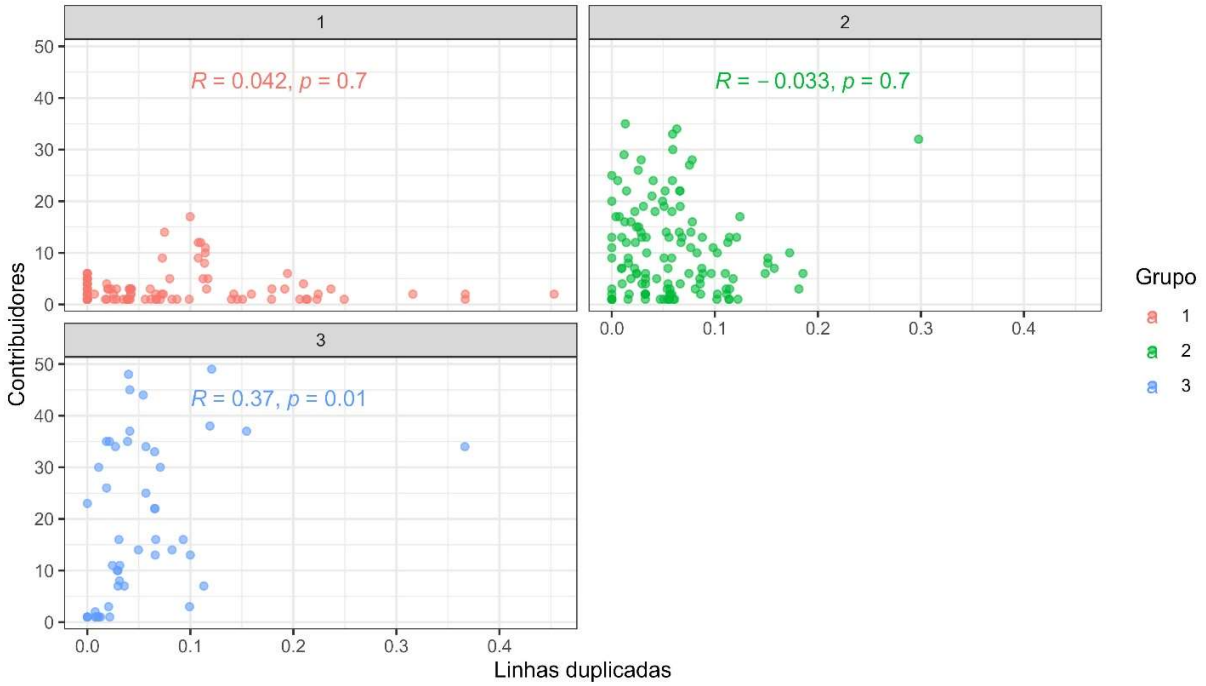

Figura B.28 Gráficos de dispersão entre a quantidade de contribuidores por semestre e a quantidade de linhas duplicadas por linha de código, segundo os grupos

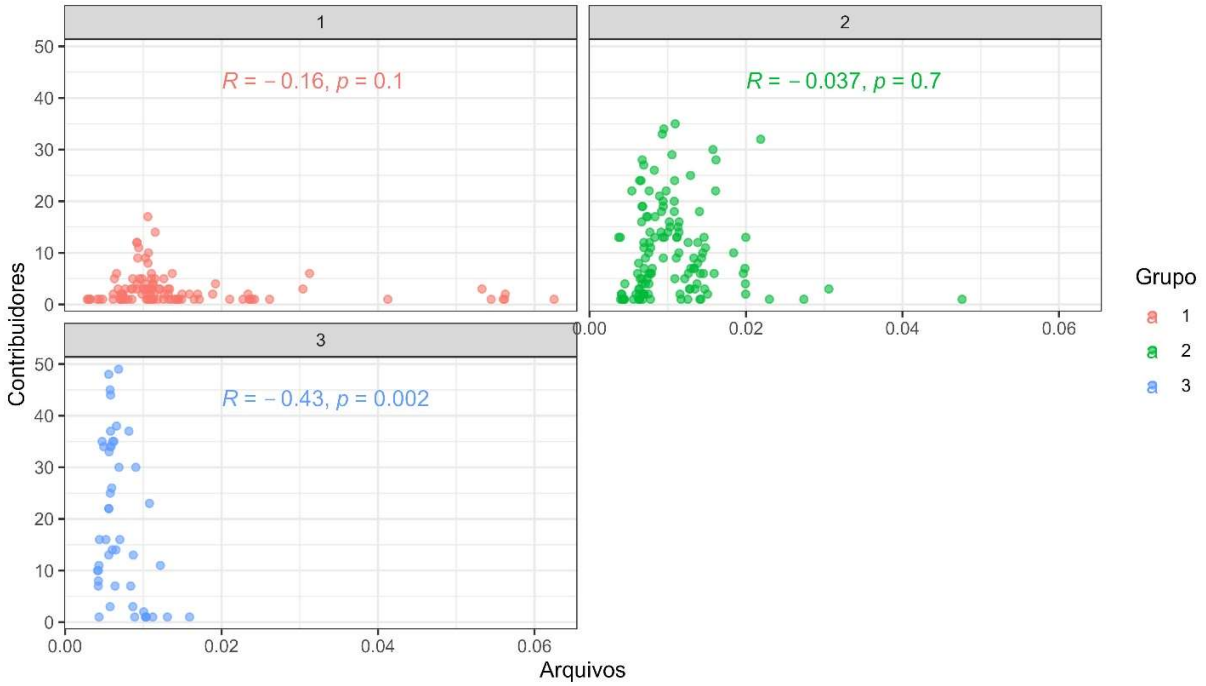

Figura B.29 Gráficos de dispersão entre a quantidade de contribuidores por semestre e a quantidade de arquivos por linha de código, segundo os grupos

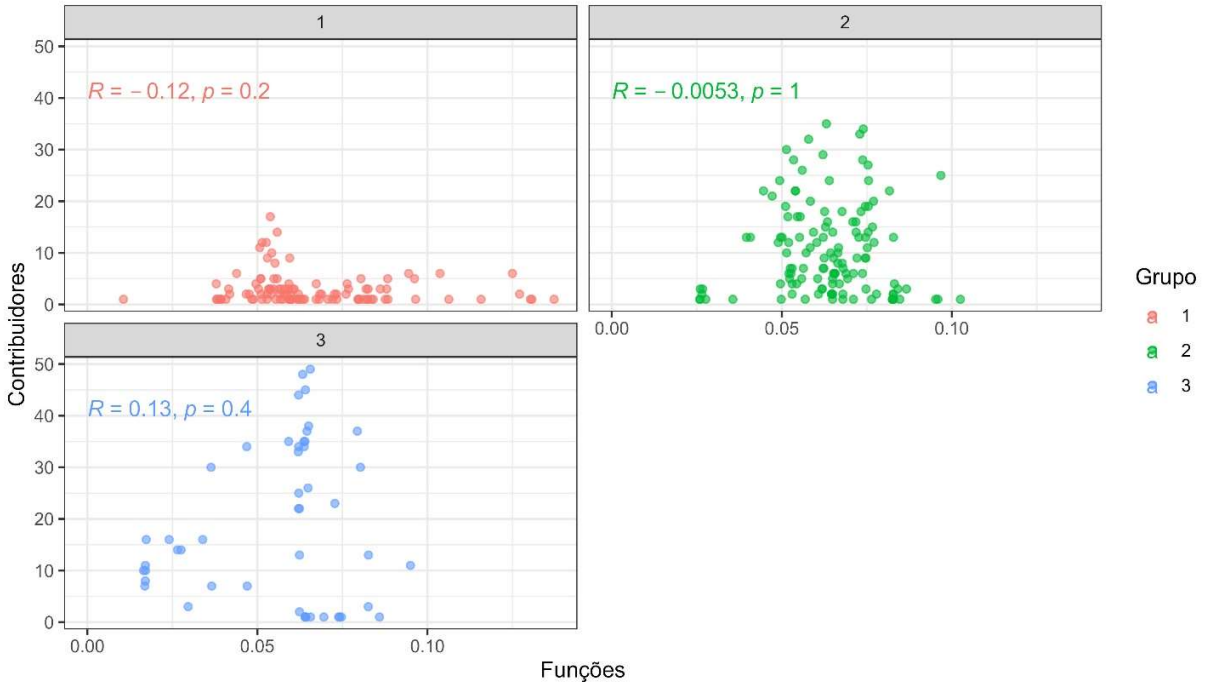

Figura B.30 Gráficos de dispersão entre a quantidade de contribuidores por semestre e a quantidade de funções por linha de código, segundo os grupos

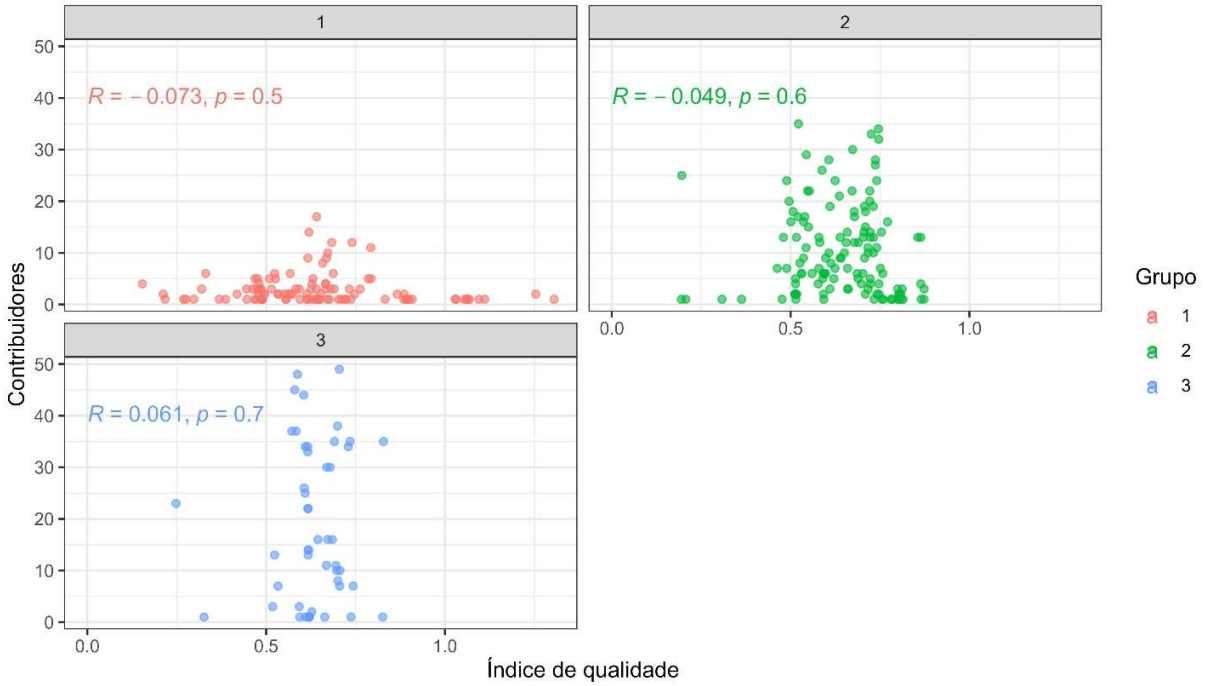

Figura B.31 Gráficos de dispersão entre a quantidade de contribuidores por semestre e o índice de qualidade (padronizado pelo número de linhas de código), segundo os grupos

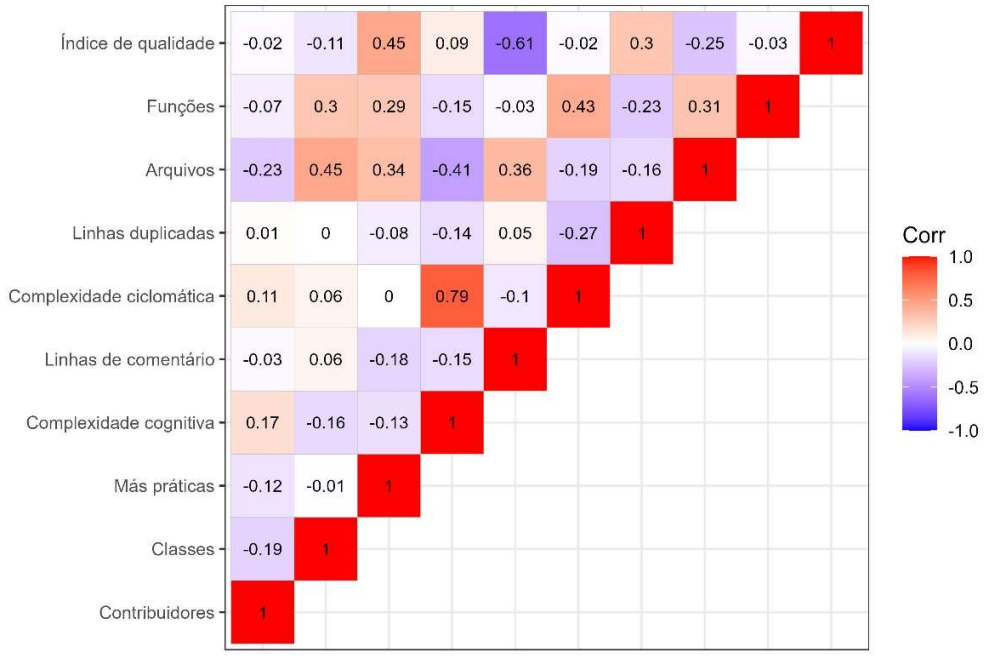

Figura B.32 Correlograma considerando observações de todos os projetos

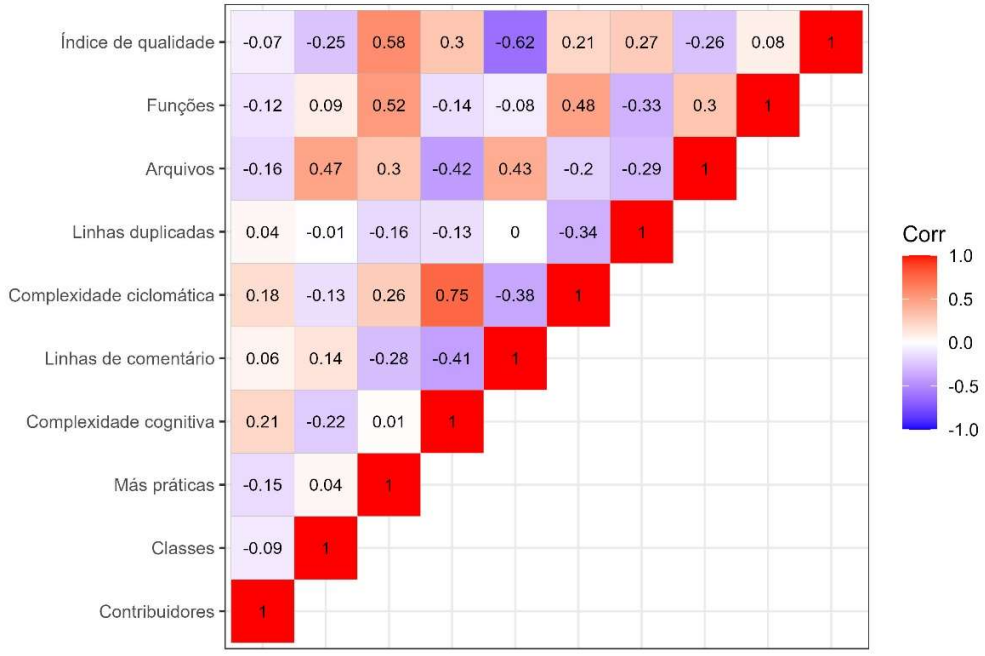

Figura B.33 Correlograma considerando apenas observações dos projetos do grupo 1

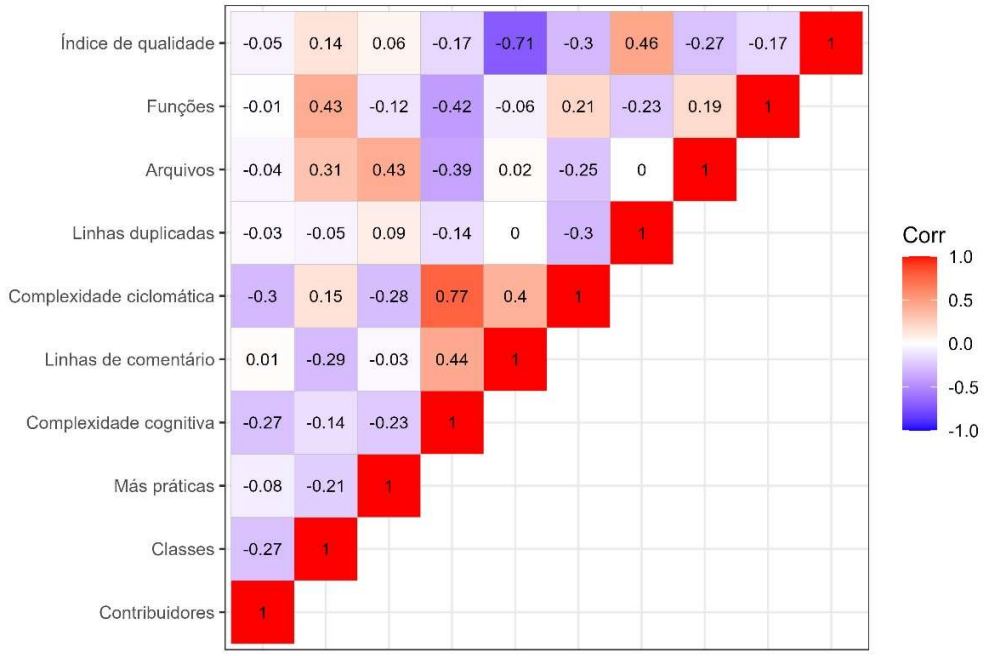

Figura B.34 Correlograma considerando apenas observações dos projetos do grupo 2

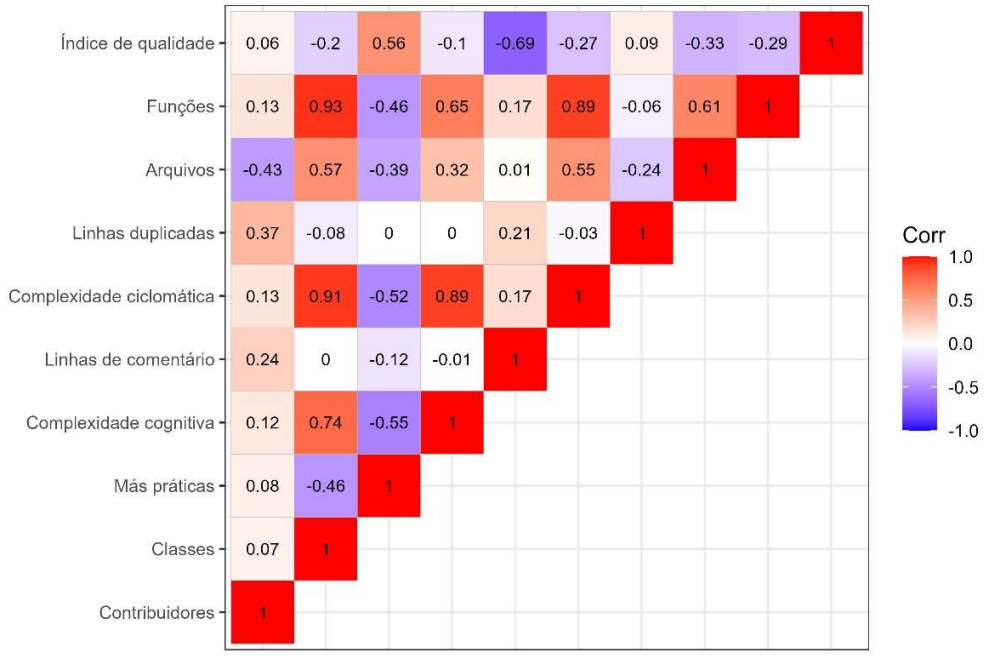

Figura B.35 Correlograma considerando apenas observações dos projetos do grupo 3

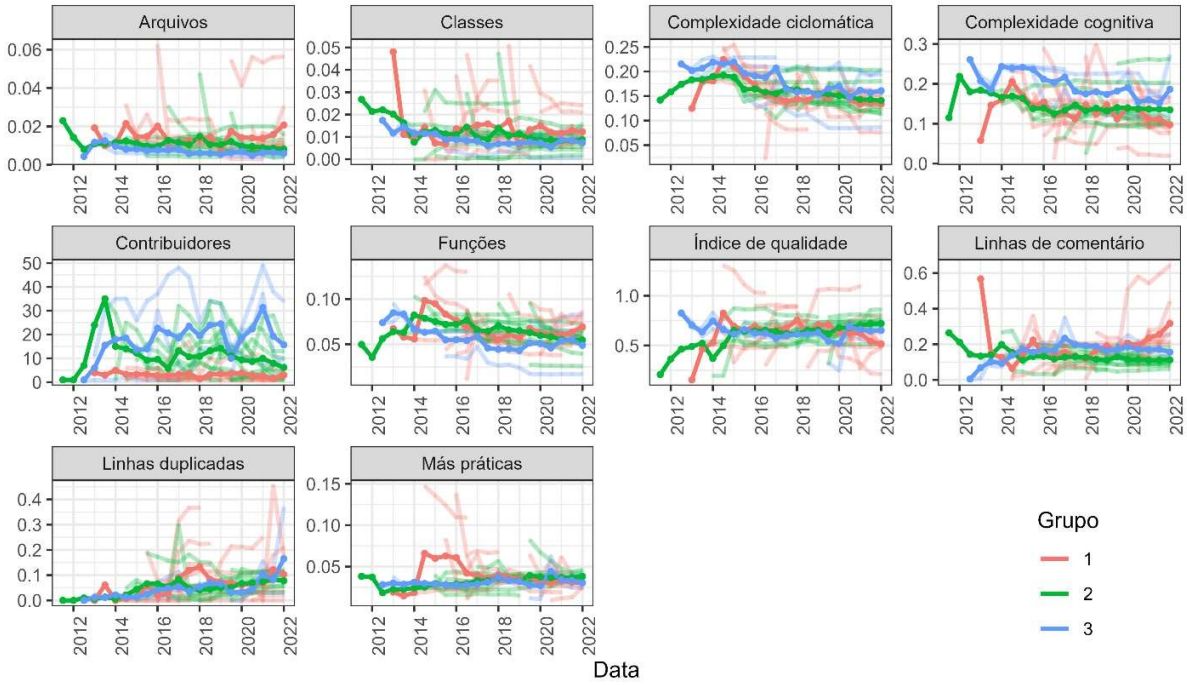

Figura B.36 Gráficos de perfis médios, segundo os grupos

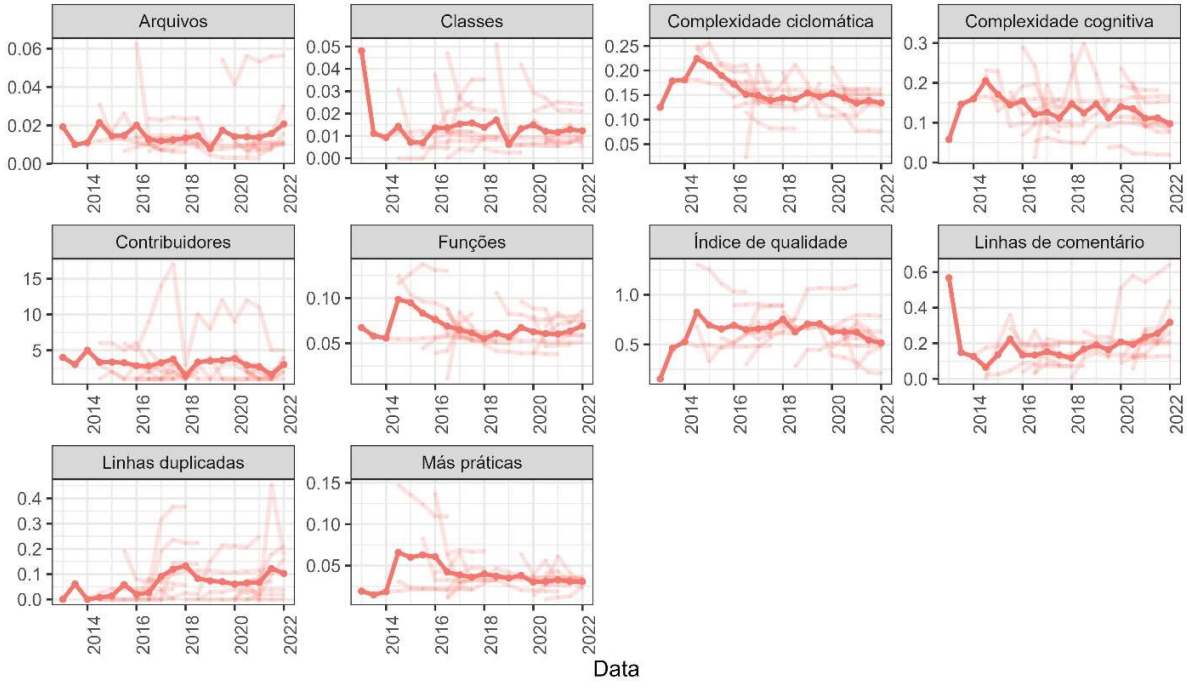

Figura B.37 Gráficos de perfis médios considerando apenas projetos do grupo 1

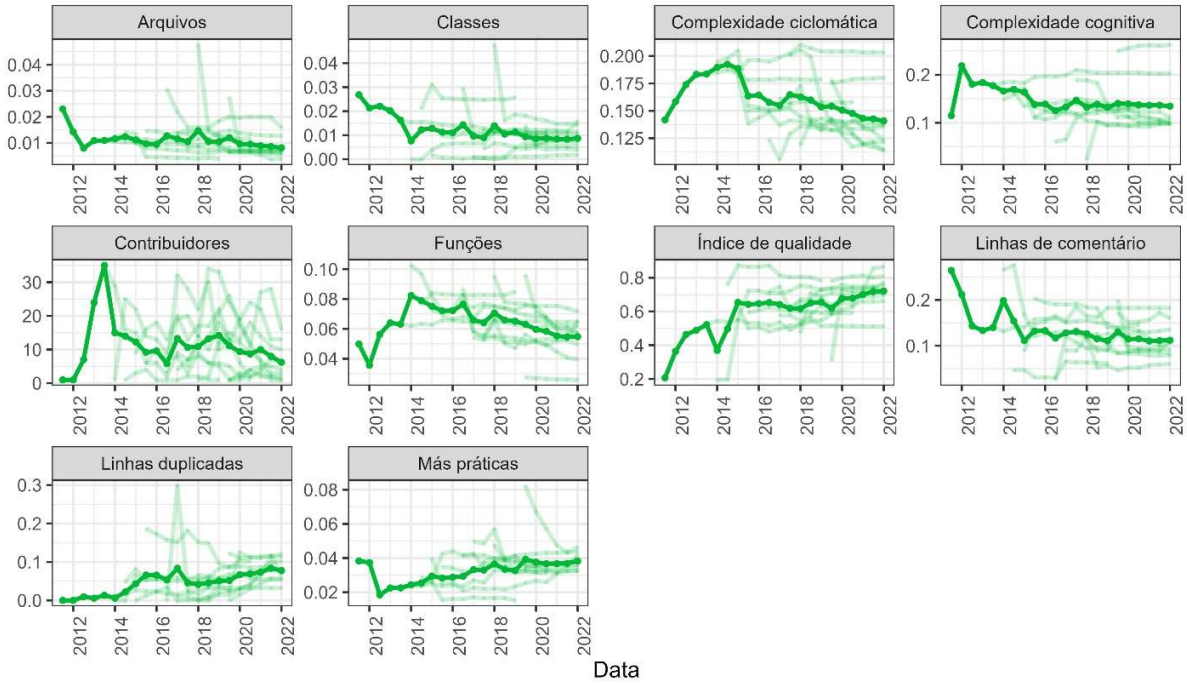

Figura B.38 Gráficos de perfis médios considerando apenas projetos do grupo 2

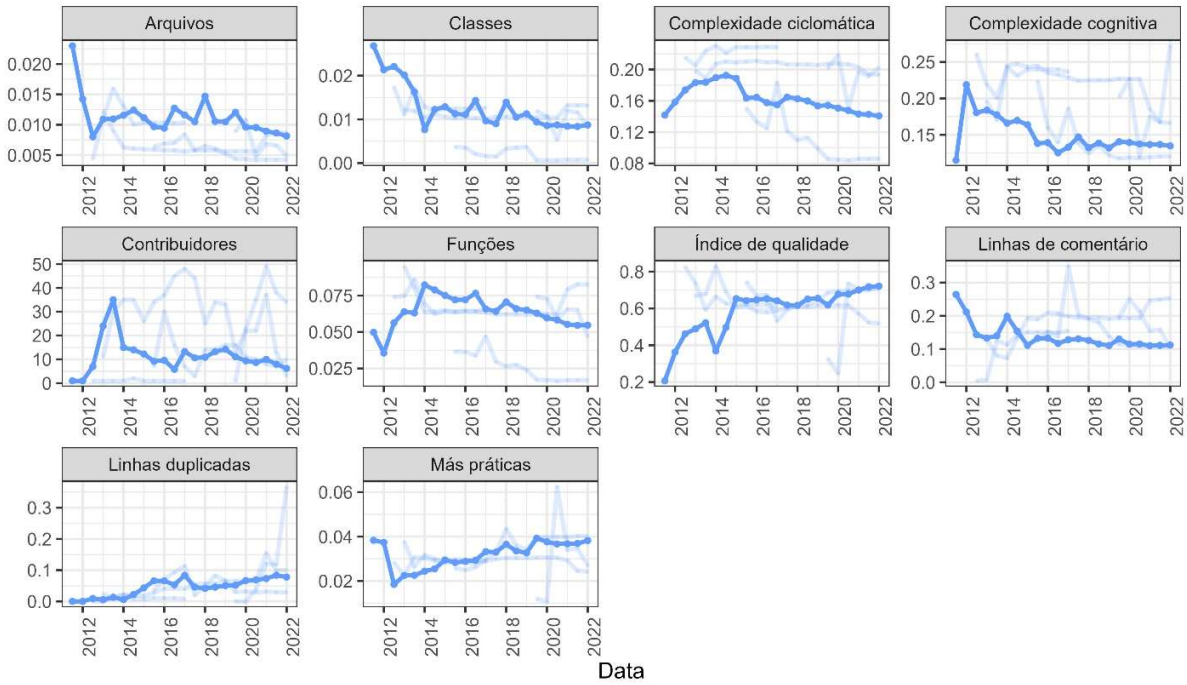

Figura B.39 Gráficos de perfis médios considerando apenas projetos do grupo 3

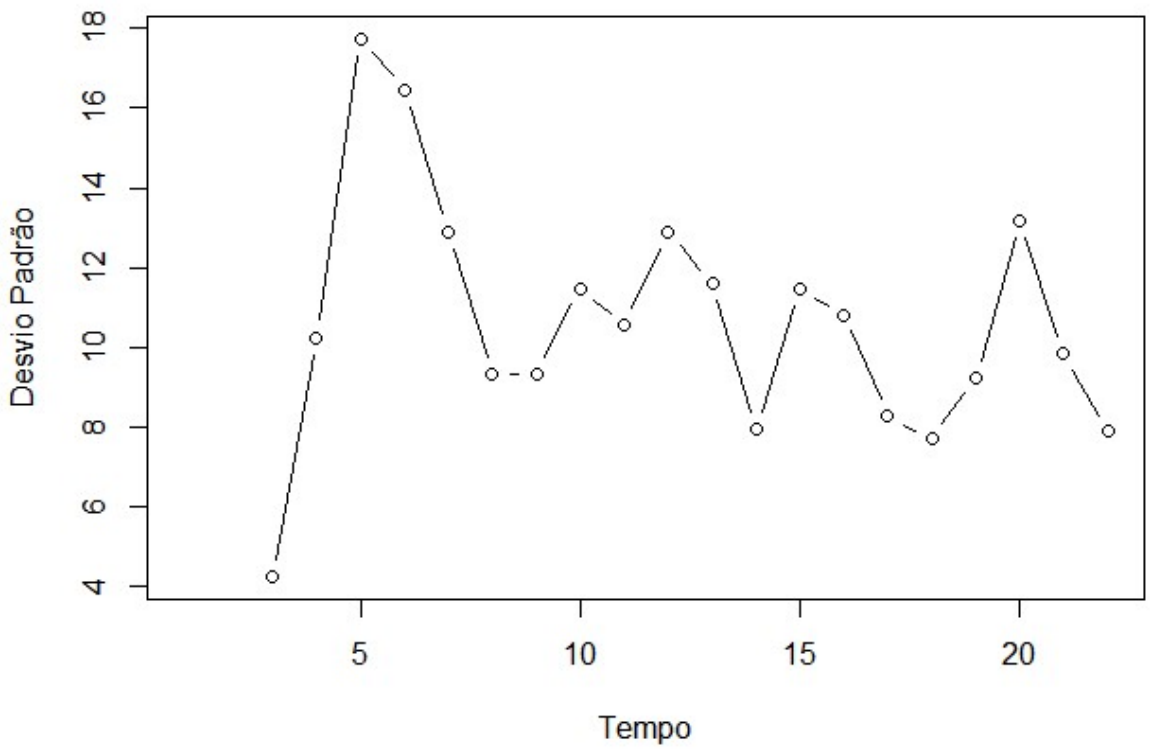

Figura B.40 Gráfico dos desvios padrões do número de contribuidores ao longo do tempo.

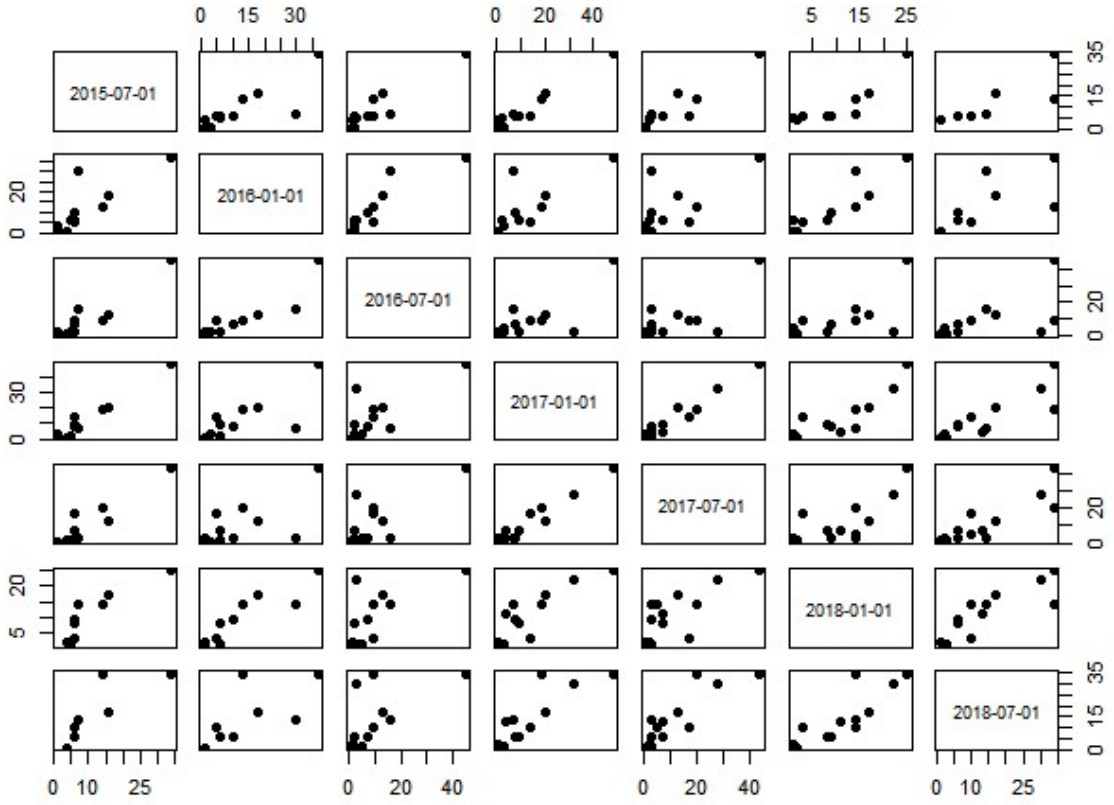

Figura B.41 Gráficos de dispersão do número de contribuidores em diferentes instantes (de julho de 2015 a julho de 2018).

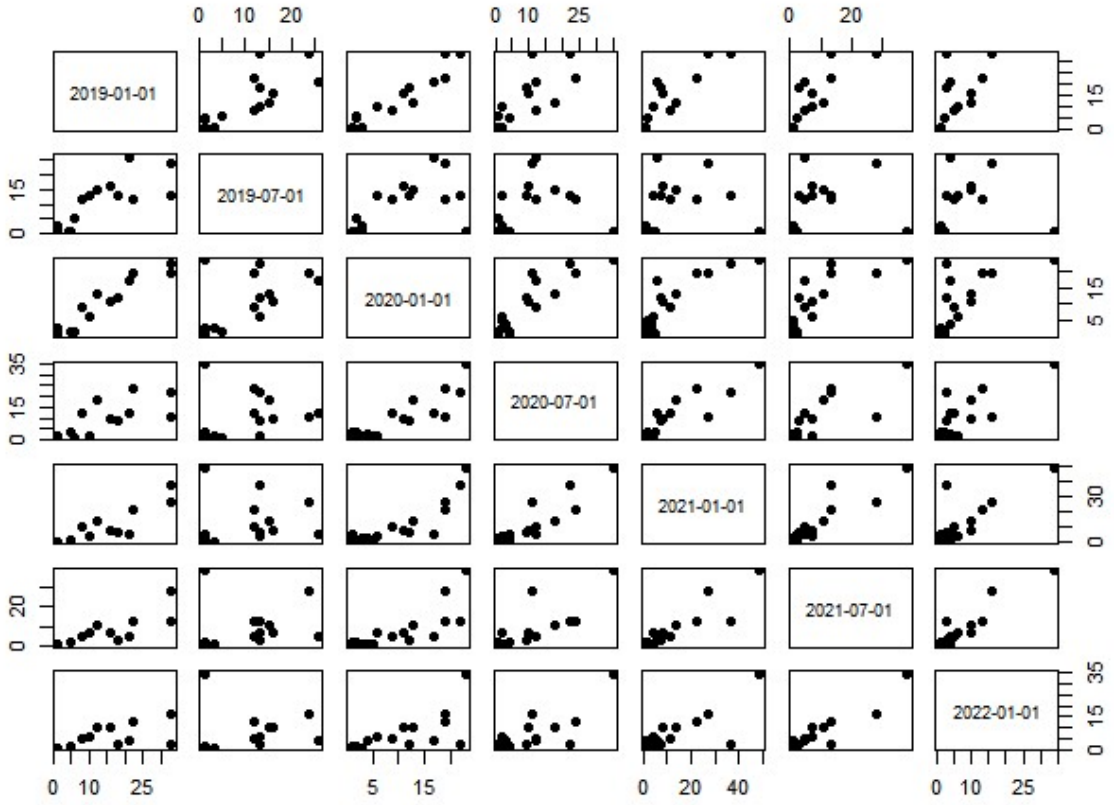

Figura B.42 Gráficos de dispersão do número de contribuidores em diferentes instantes (de janeiro de 2019 a janeiro de 2022).

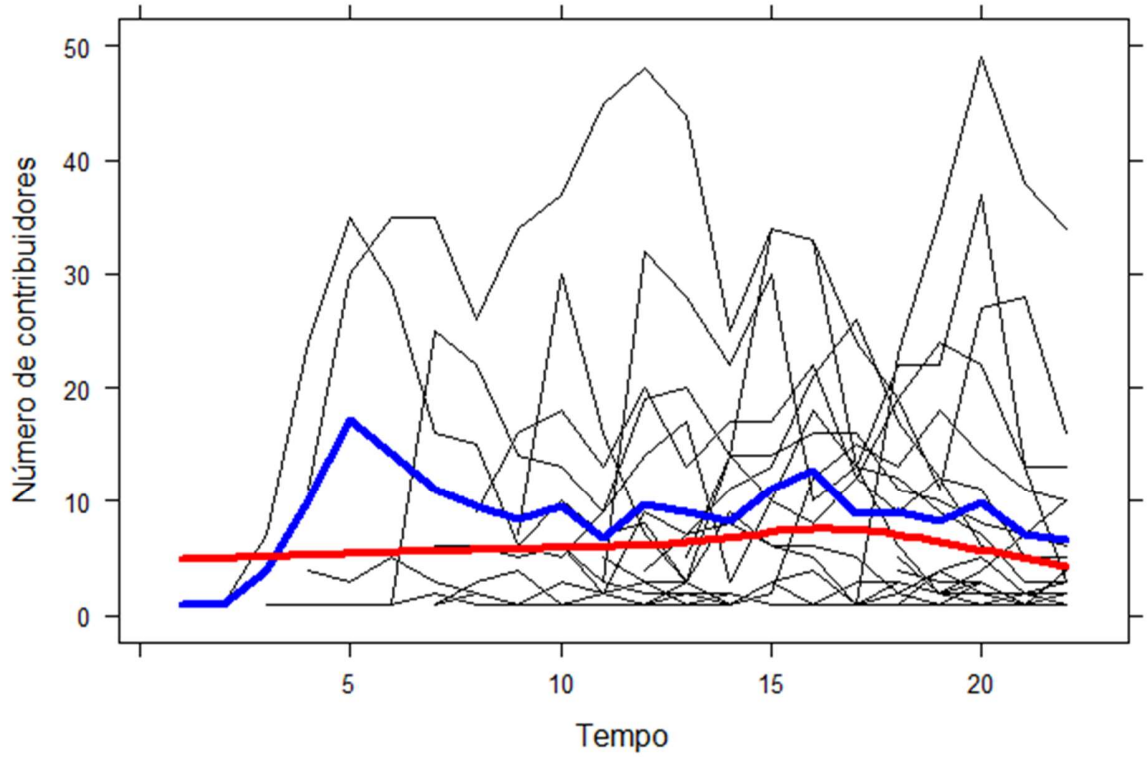

Figura B.43 Gráfico de perfis do número de contribuidores, com perfil médio destacado em azul, e a curva lowess destacada em vermelho (Tempo em semestres).

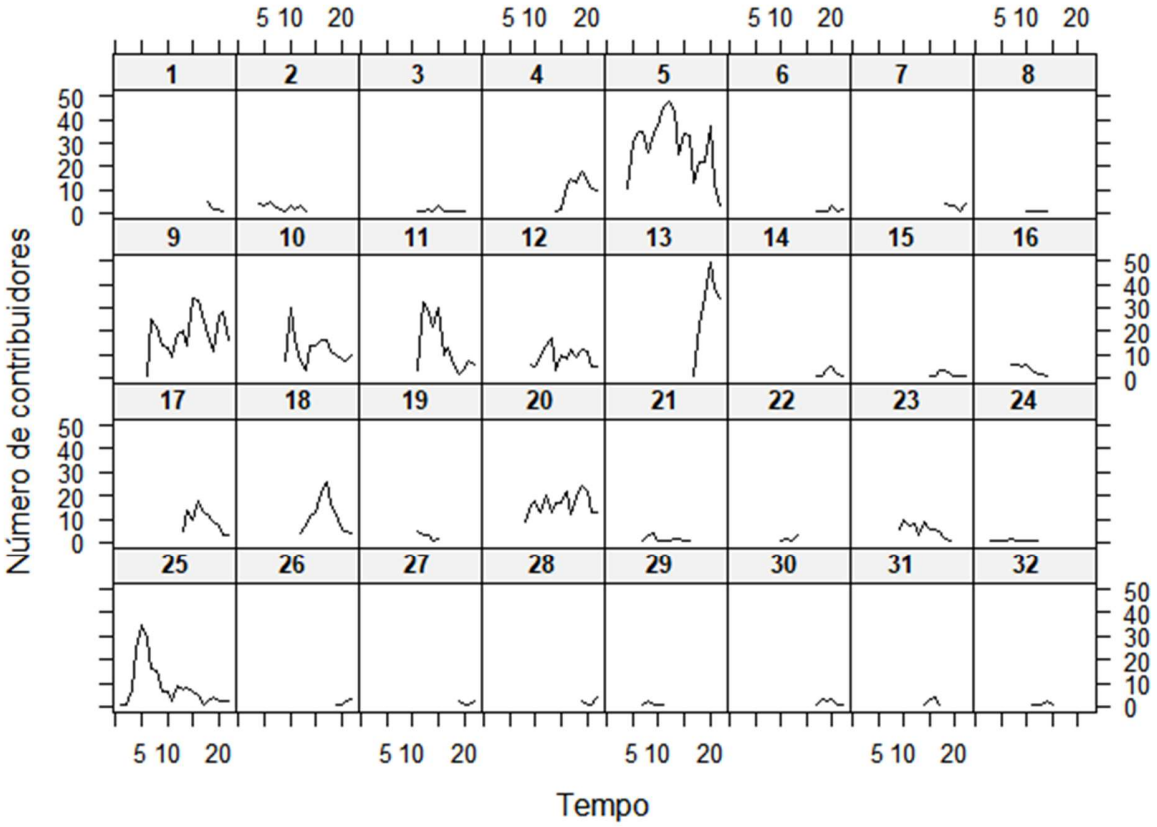

Figura B.44 Gráfico de perfis separados do número de contribuidores (Tempo em semestres).

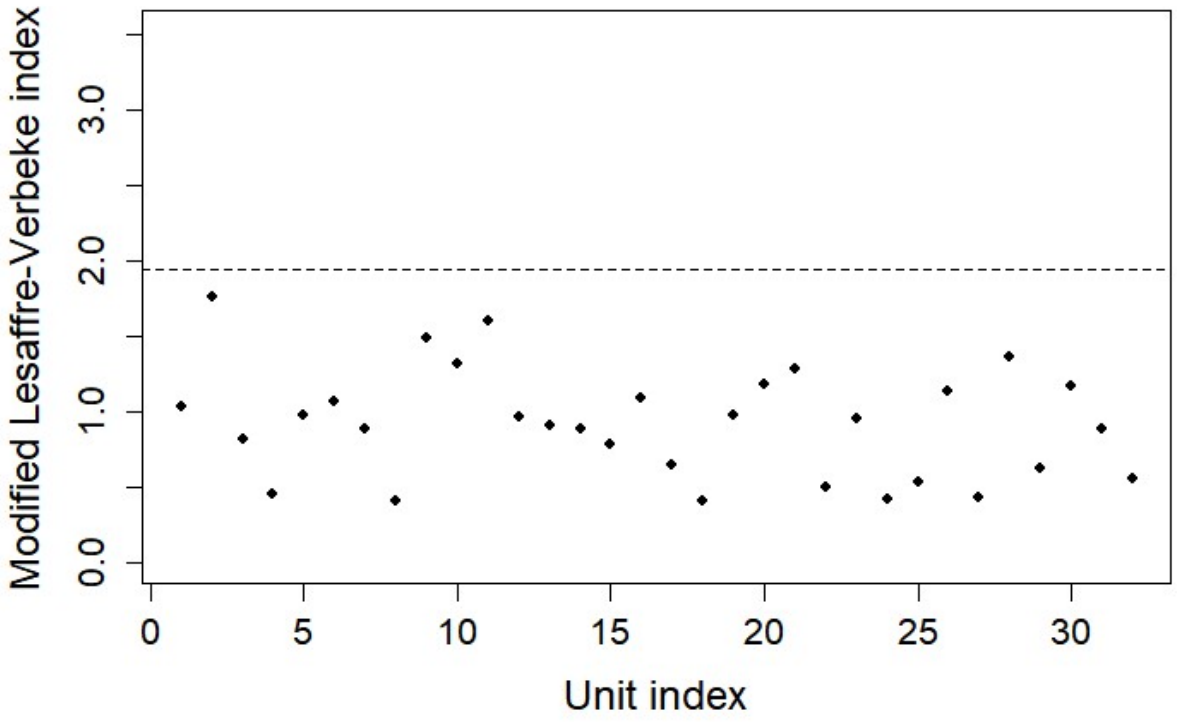

Figura B.45 Gráfico de resíduos Lesaffre-Verbeke, para o modelo saturado

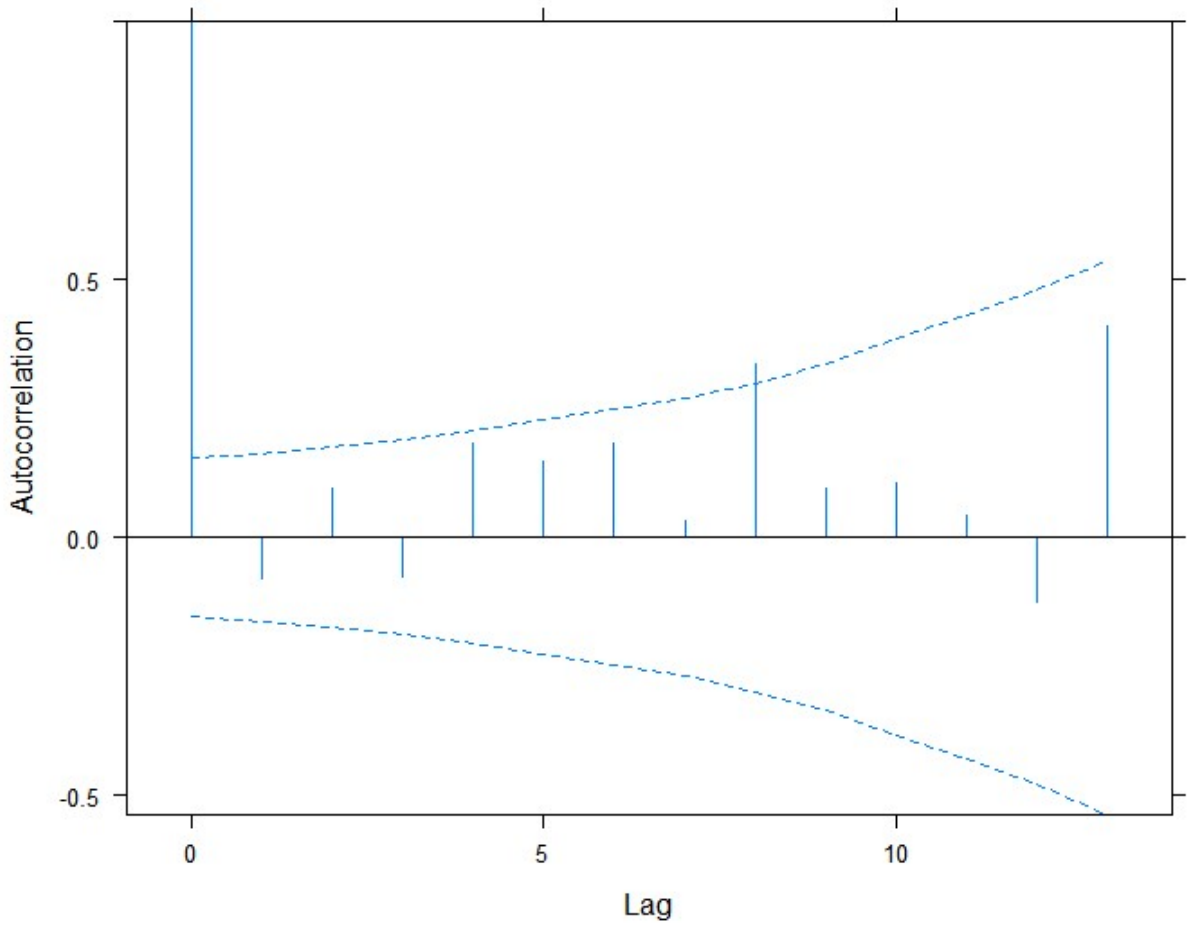

Figura B.46 Gráfico dos autocorrelação dos resíduos padronizados, para o modelo saturado

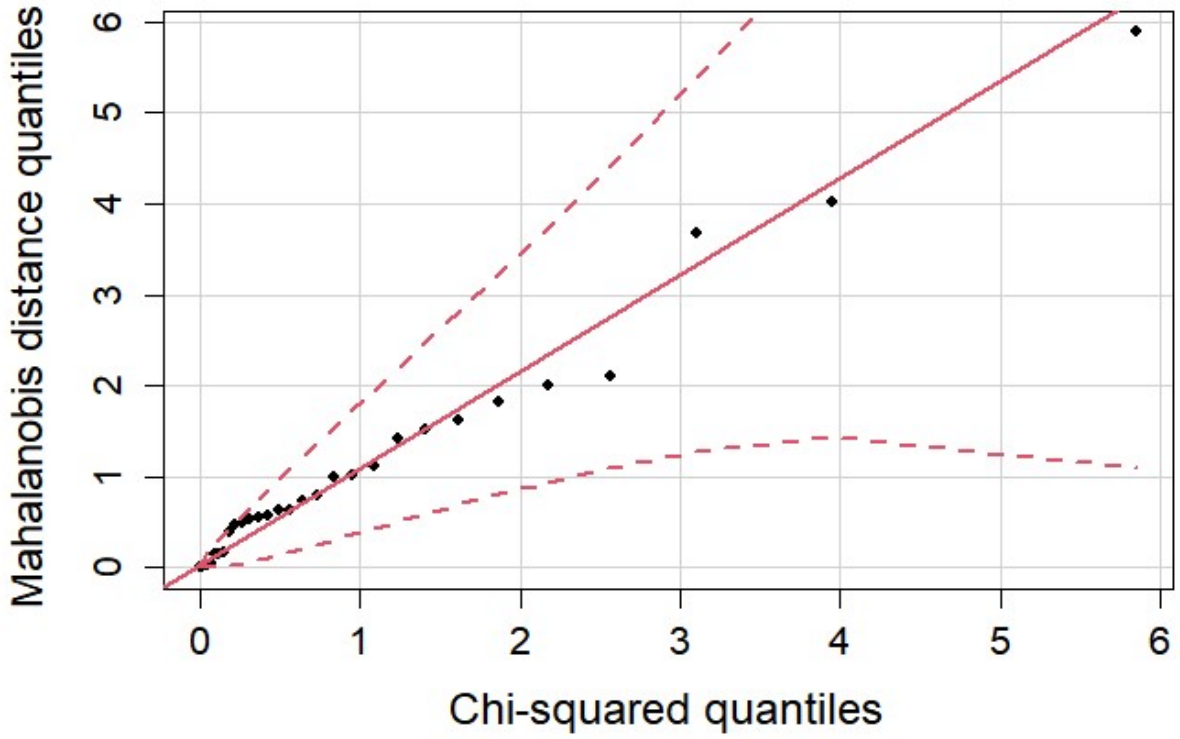

Figura B.47 QQ-plot dos preditores dos efeitos aleatórios, para o modelo saturado

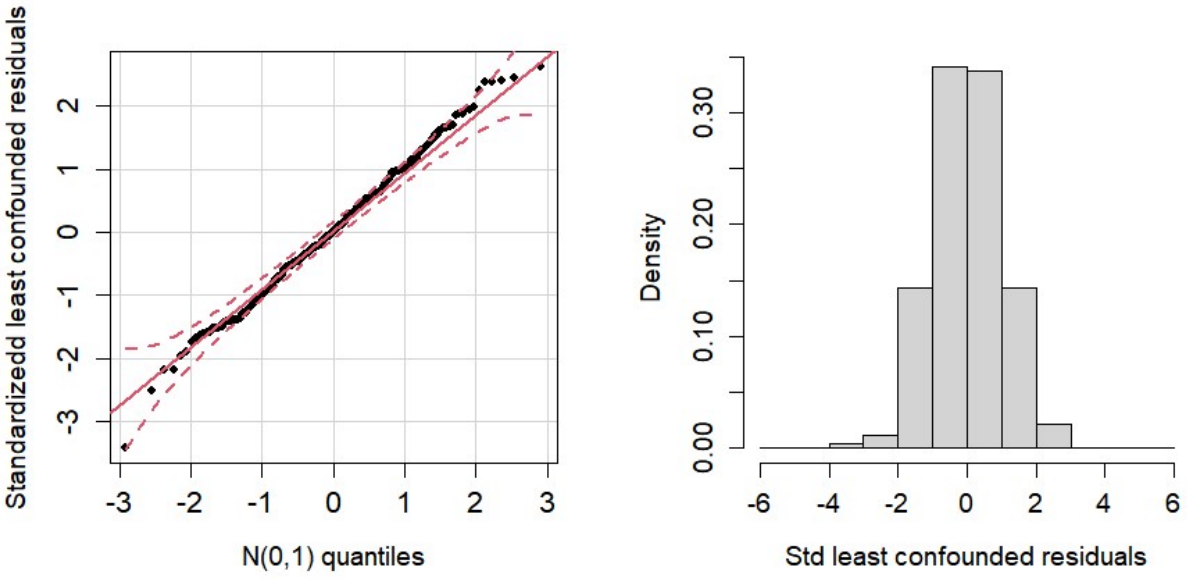

Figura B.48 Gráfico dos resíduos minimamente confundidos, para o modelo saturado