Análise Estatística e Modelagem Preditiva com Shiny

1. Introdução

Este trabalho utiliza o dataset `KC1_classlevel_numdefect.xlsx`, que contém métricas de qualidade de classes de um sistema orientado a objetos. O objetivo é aplicar técnicas estatísticas, criar um modelo preditivo e construir uma aplicação web interativa com Shiny.

2. Carregamento do Dataset

O dataset foi carregado com a biblioteca `readxl`:

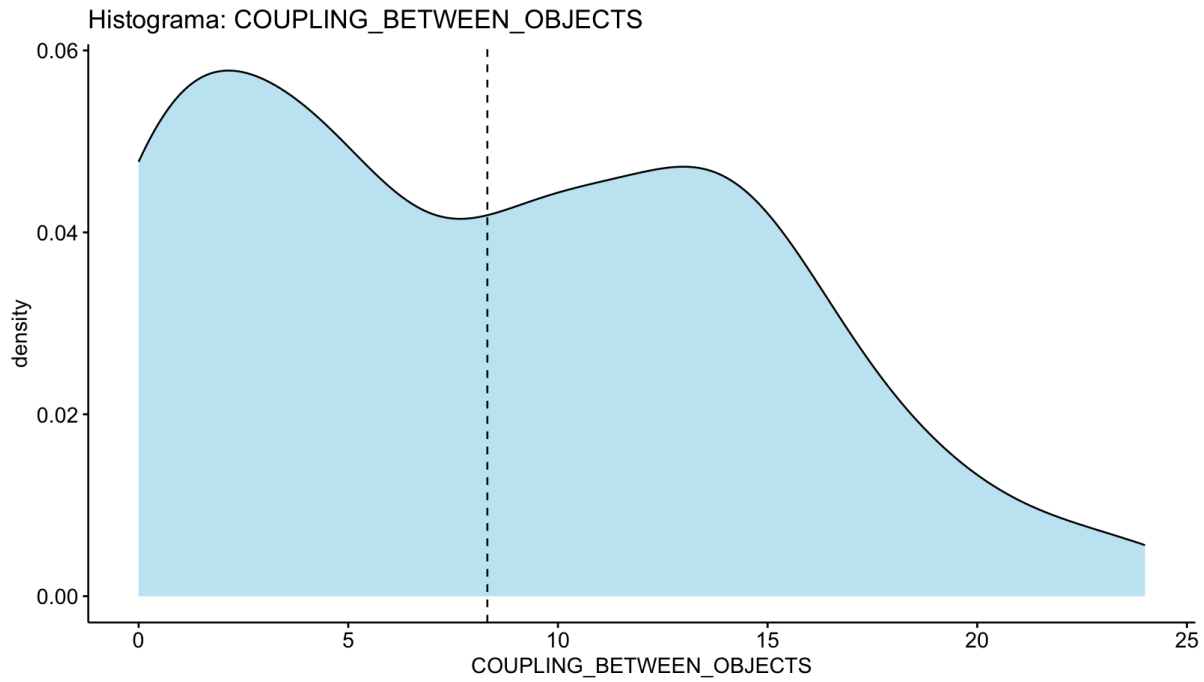
```
library(readxl)

dados <-
read_excel("/Users/luisgustavobrito/Downloads/dataset_KC1_classlevel_numdefect.xlsx")
```

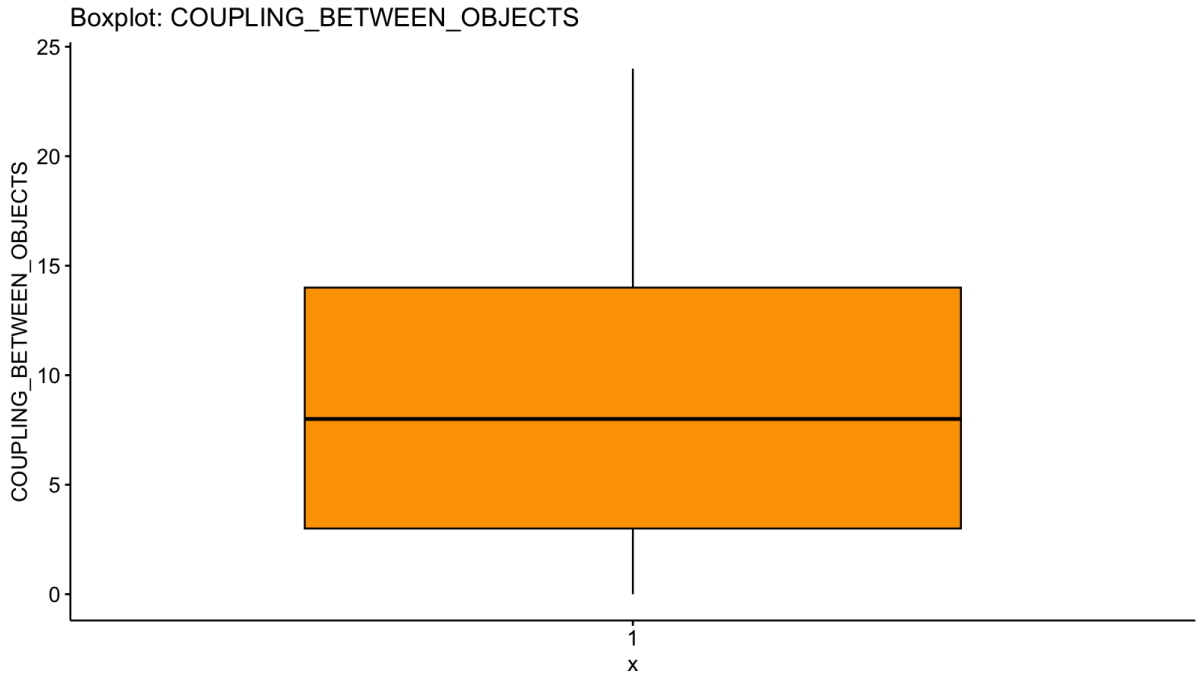
3. Estatística Descritiva

3.1 Medidas Calculadas para Cada Variável:

COUPLING_BETWEEN_OBJECTS

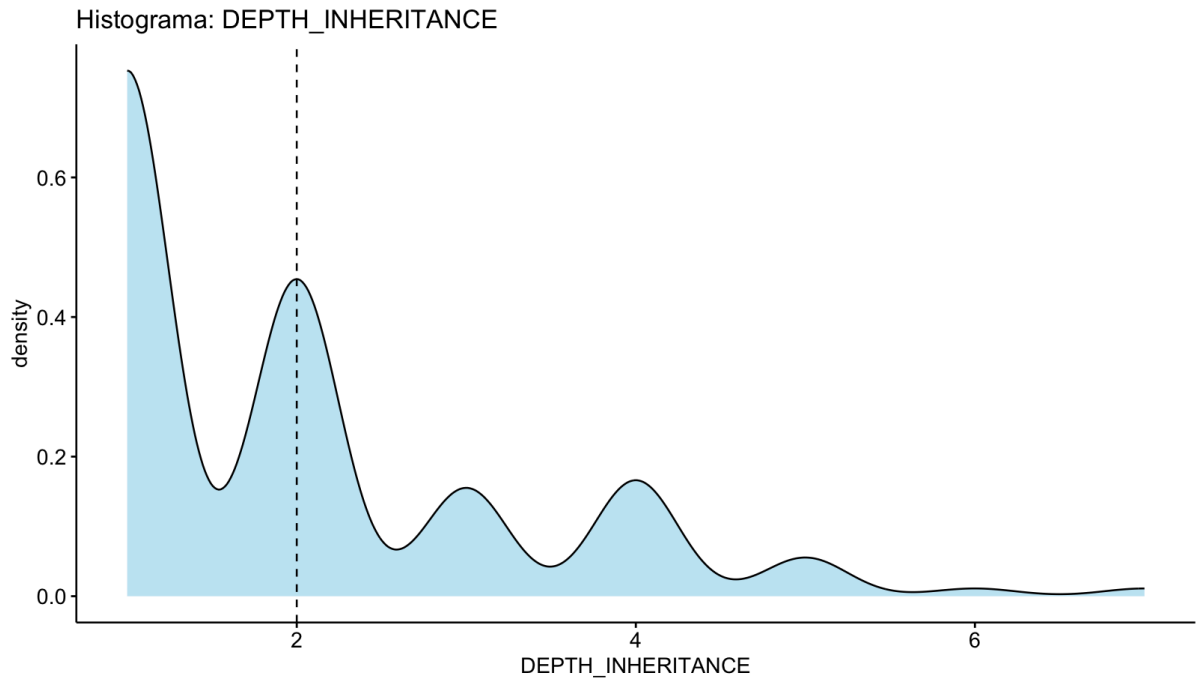
- Média: 8.317241 — levemente acima da mediana (8), indicando leve assimetria à direita.
- Moda: 0 — muitas classes com zero acoplamento.
- Alta dispersão (DP = 6.38) e p-valor < 0.05 no teste de normalidade: **não normal**.

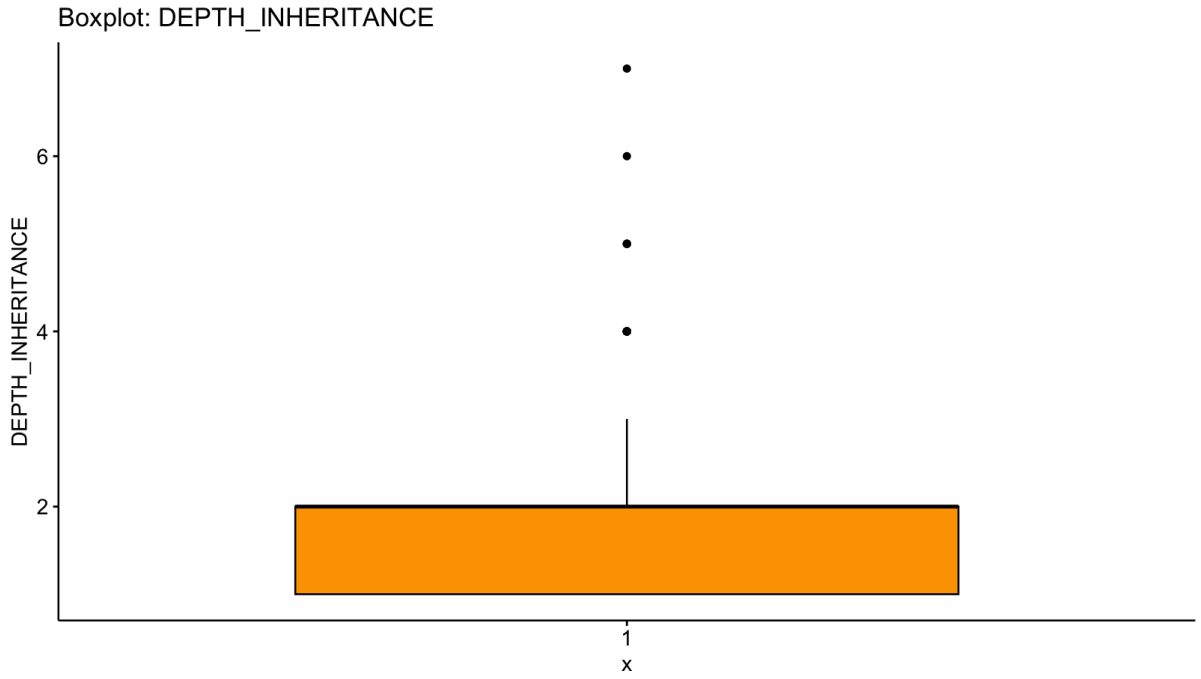




DEPTH_INHERITANCE

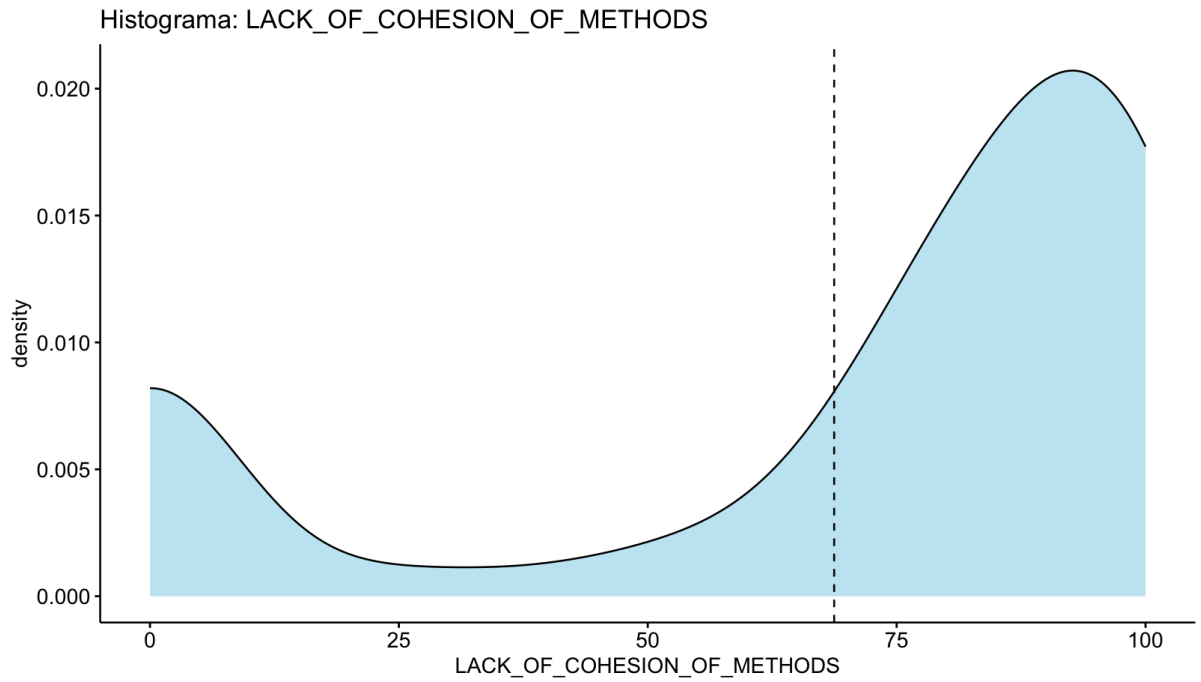
- Média e mediana iguais (2), mas moda em 1 — tendência leve à esquerda.
- Baixa dispersão (DP = 1.26) e p-valor muito baixo: **não normal**.

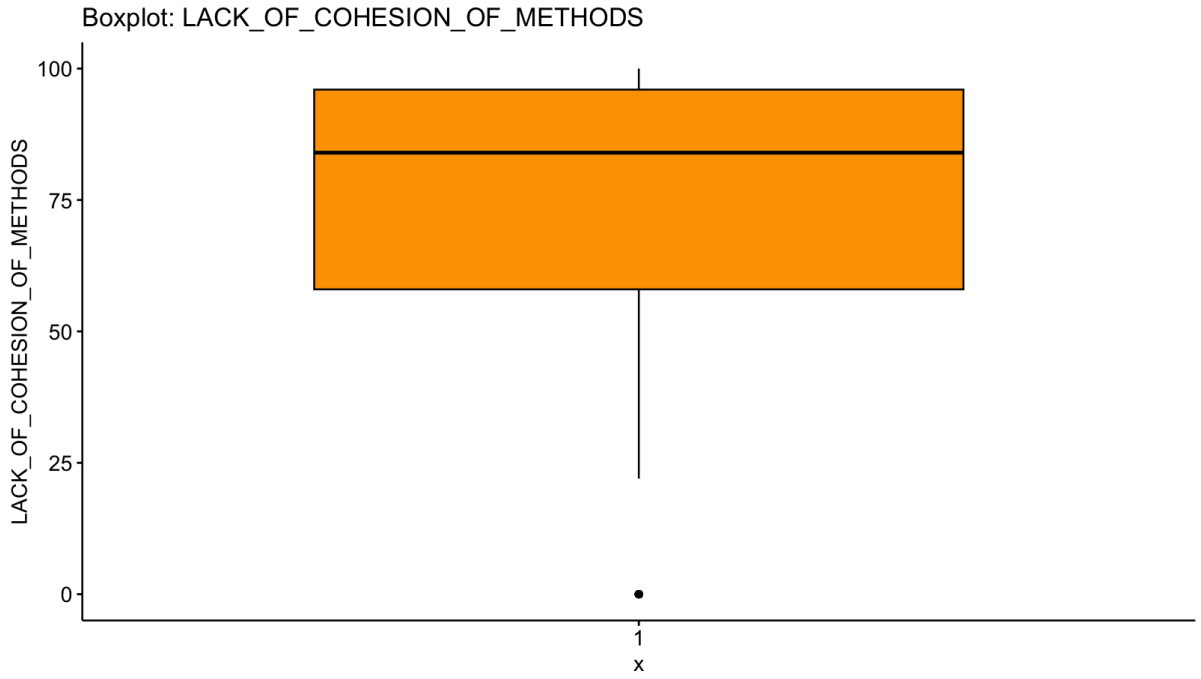




LACK_OF_COHESION_OF_METHODS

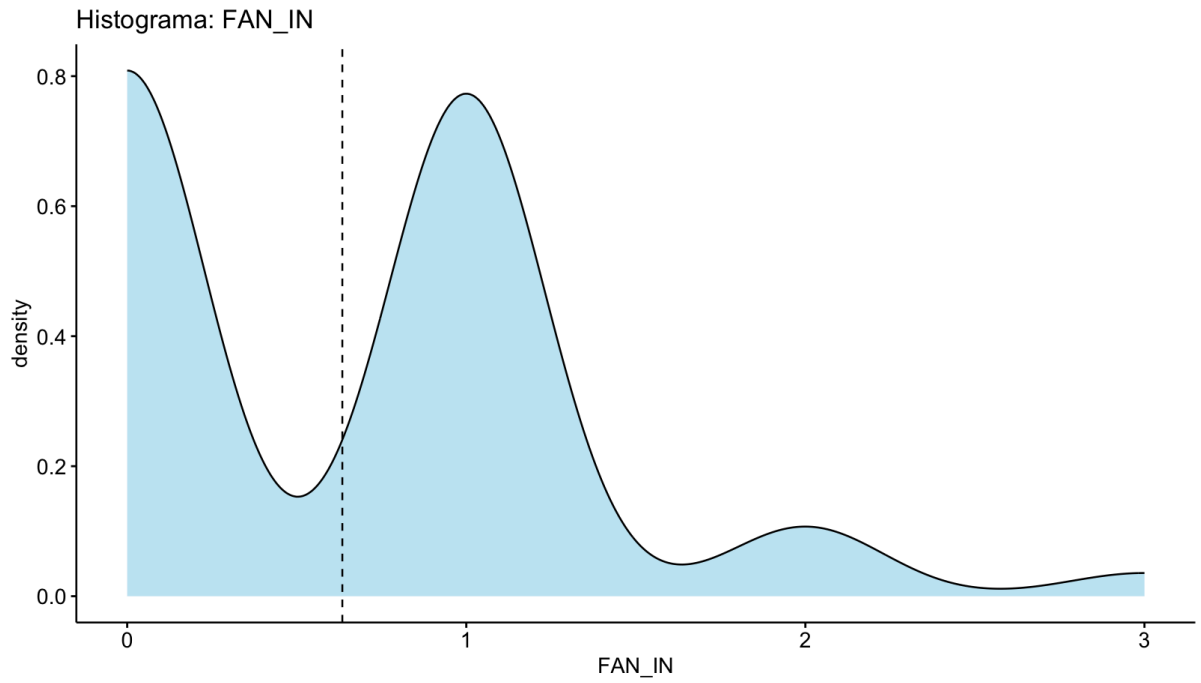
- Média: 68.72, mas mediana maior (84) — assimetria à esquerda.
- Moda: 100 — muitas classes pouco coesas.
- Alta dispersão e **não normal**.

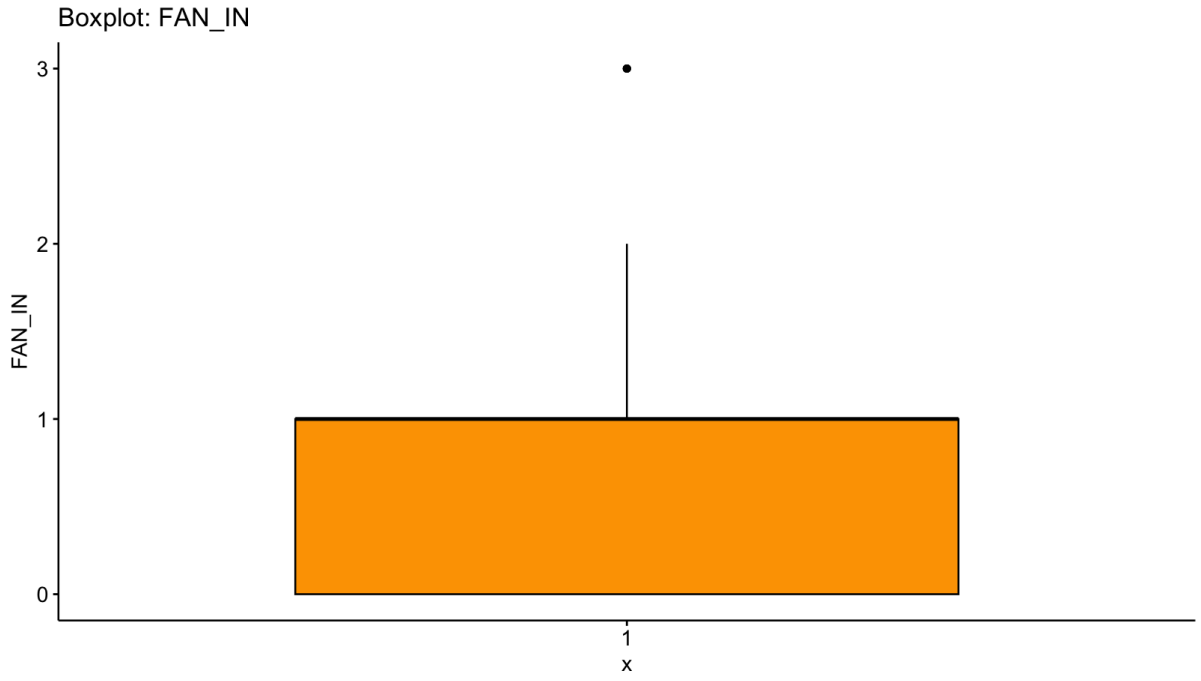




FAN_IN

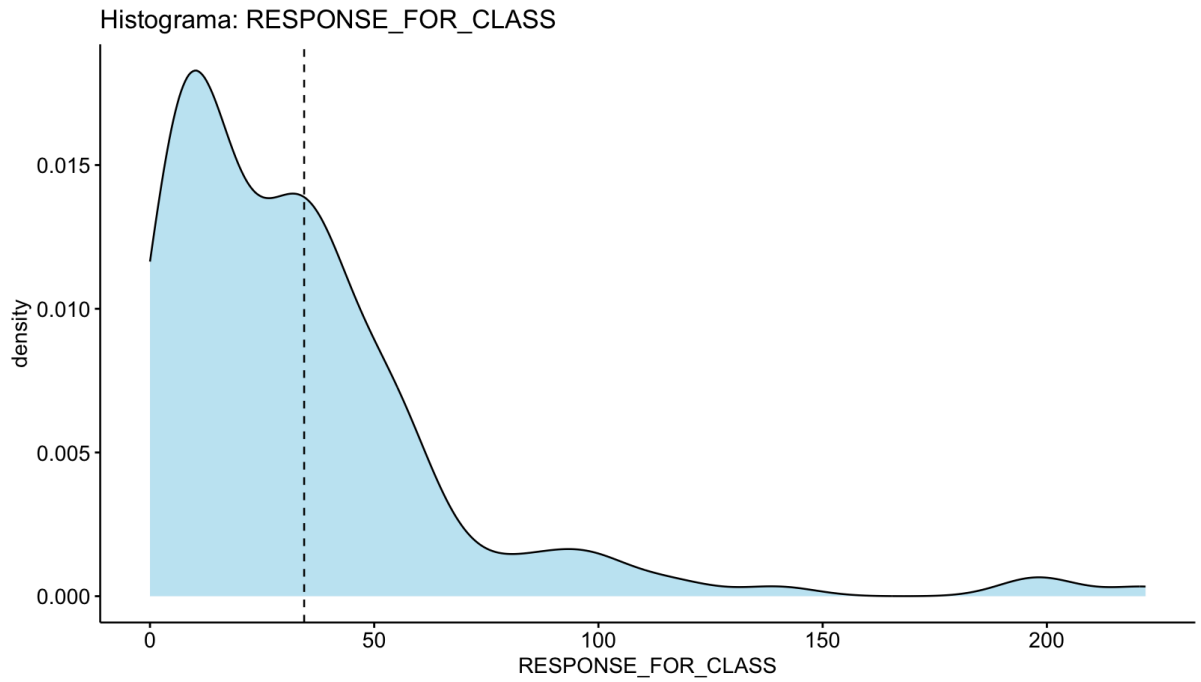
- Média < Mediana (0.63 vs 1), moda 0 — assimetria à esquerda.
- Baixa dispersão (DP = 0.69) e distribuição **não normal**.

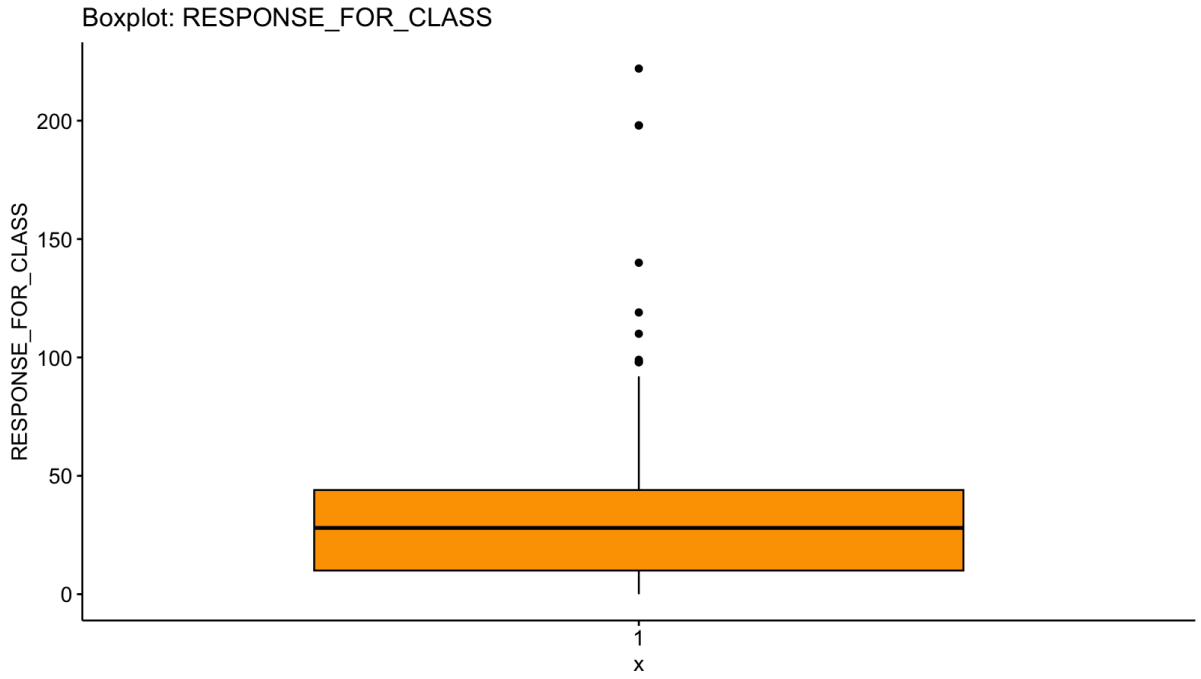




RESPONSE_FOR_CLASS

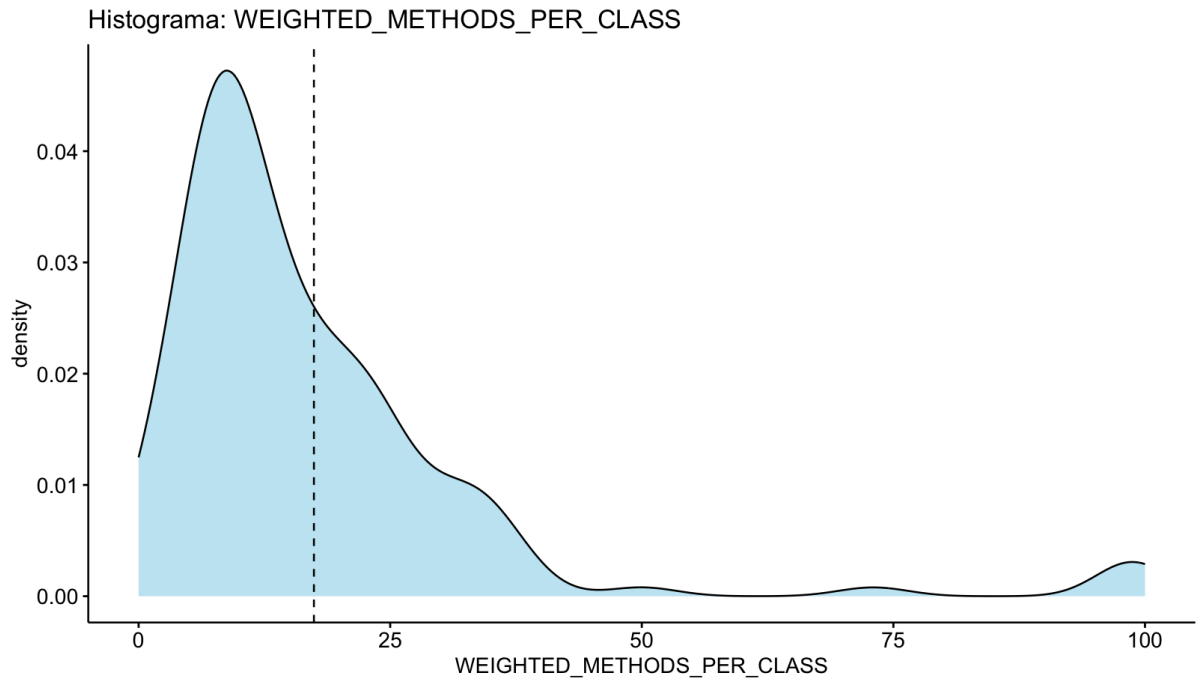
- Média: 34.37, mediana: 28 — assimetria leve à direita.
- Moda: 38 — valor mais recorrente.
- Grande amplitude (0 a 222) e **não normal**.

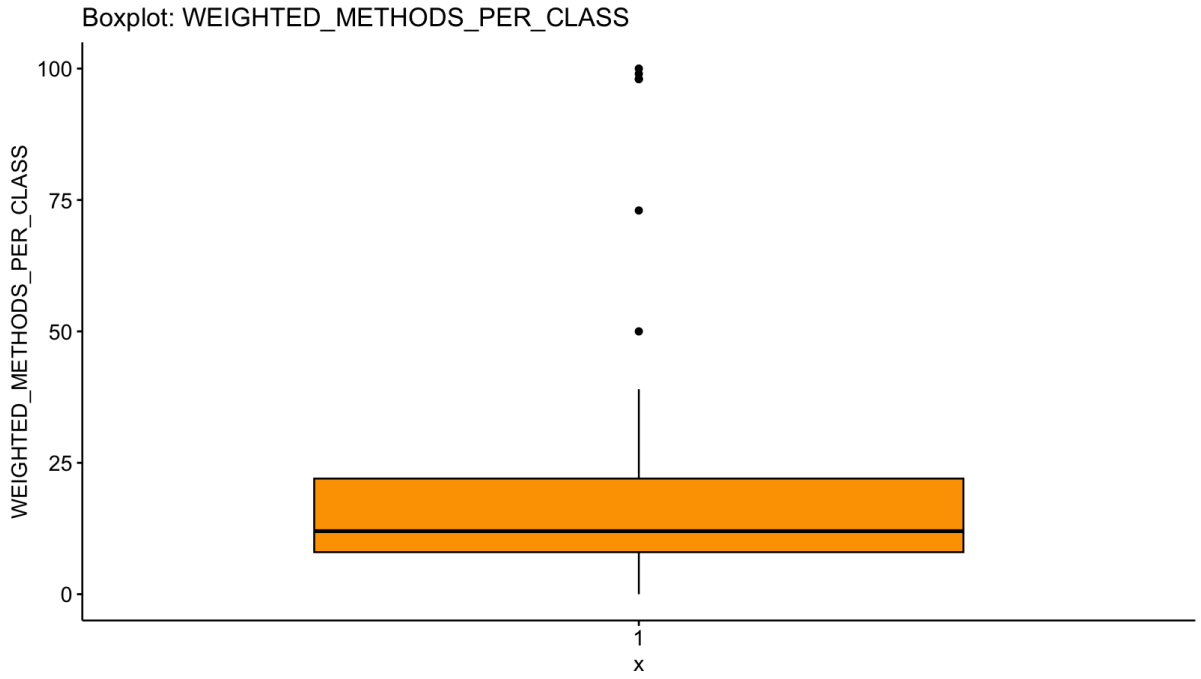




WEIGHTED_METHODS_PER_CLASS

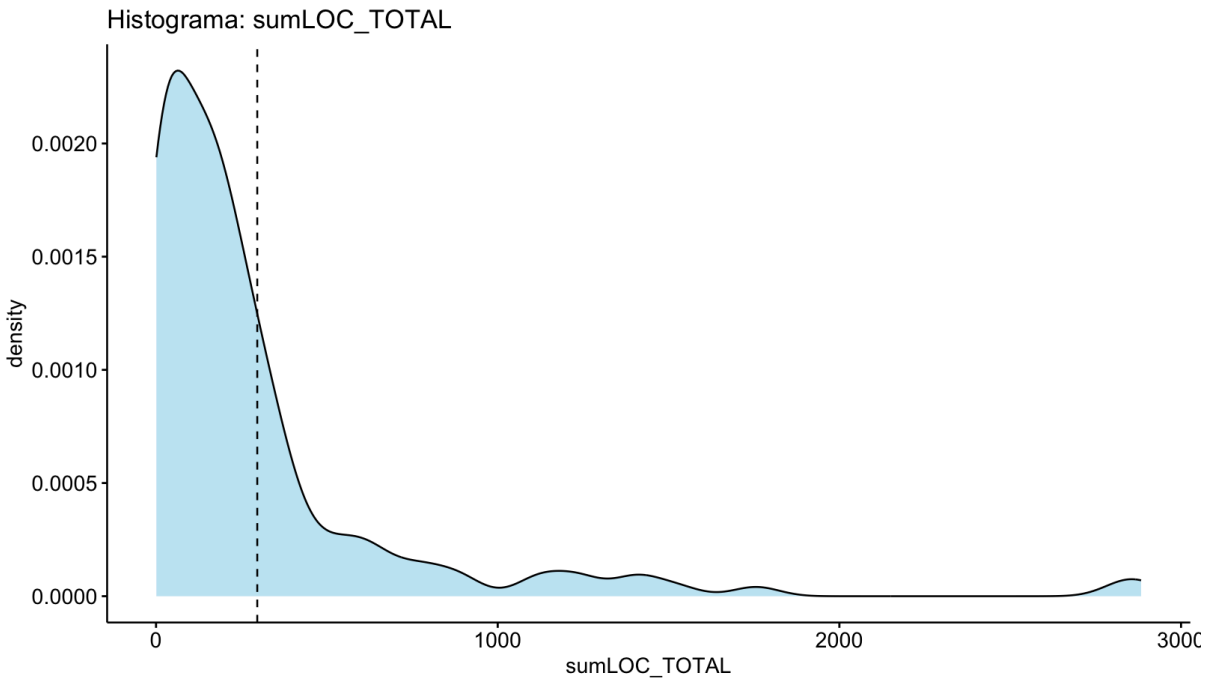
- Média e desvio padrão elevados (17.42 ± 17.45), moda em 8.
- Assimetria à direita e distribuição **não normal**.

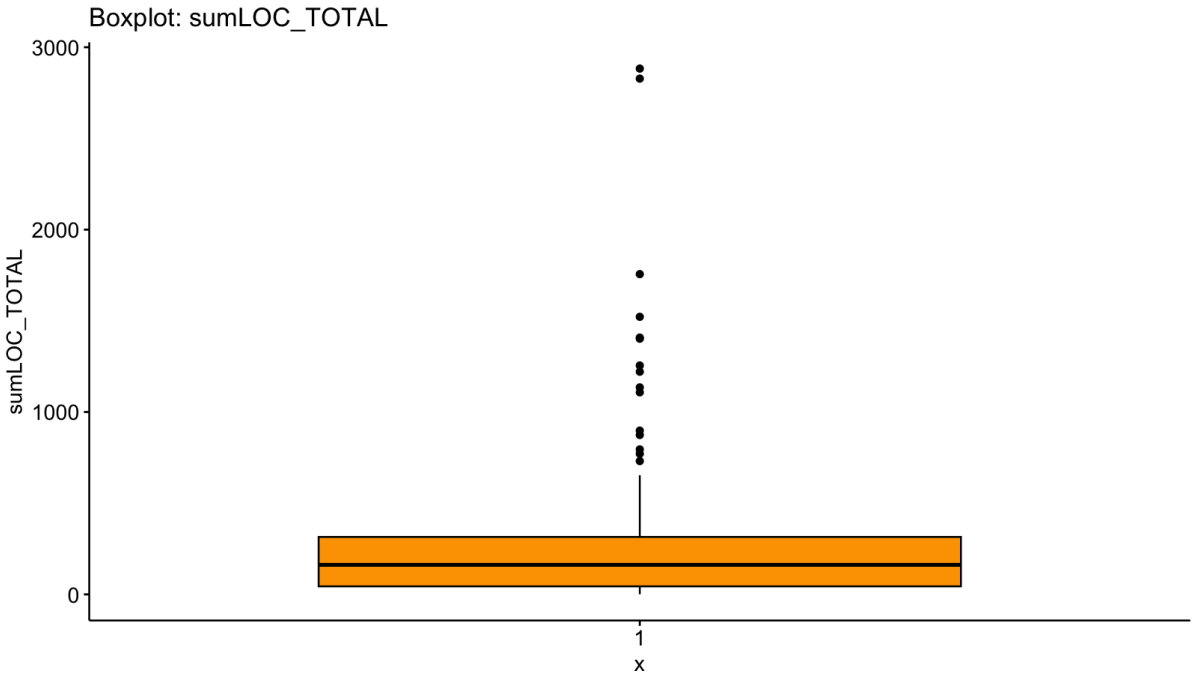




sumLOC_TOTAL

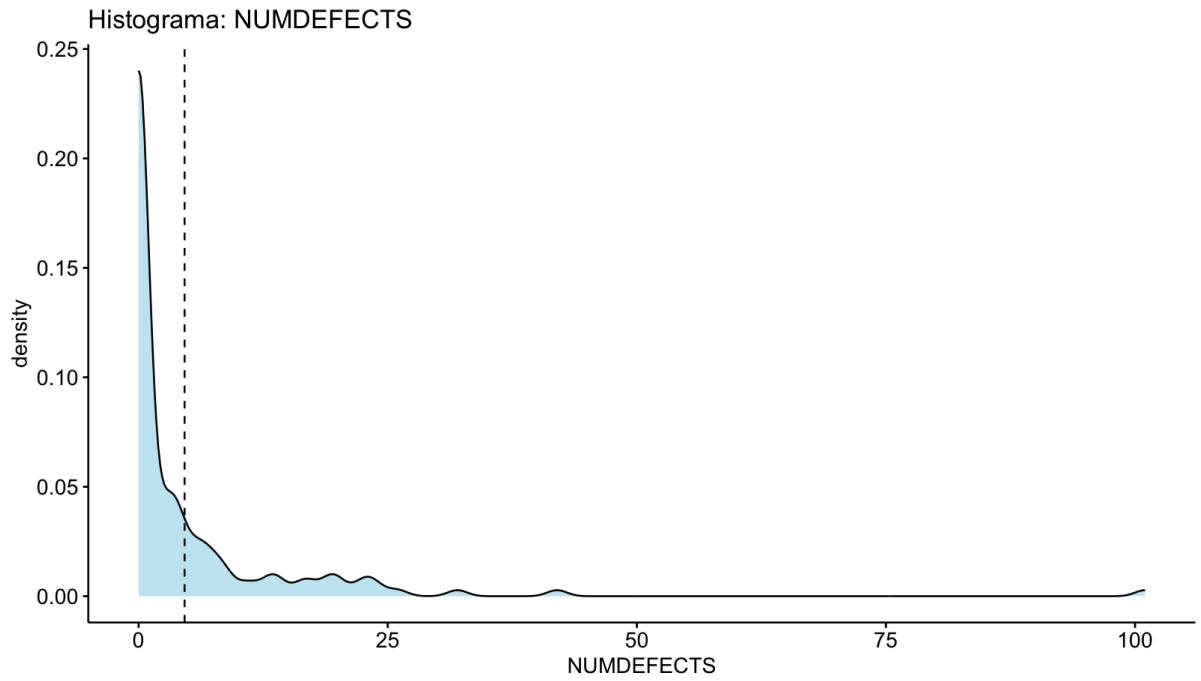
- Média: 296, mediana: 162 — distribuição altamente assimétrica (direita).
- Moda muito baixa (1) e máximo muito alto (2883).
- Altamente dispersa e **não normal**.

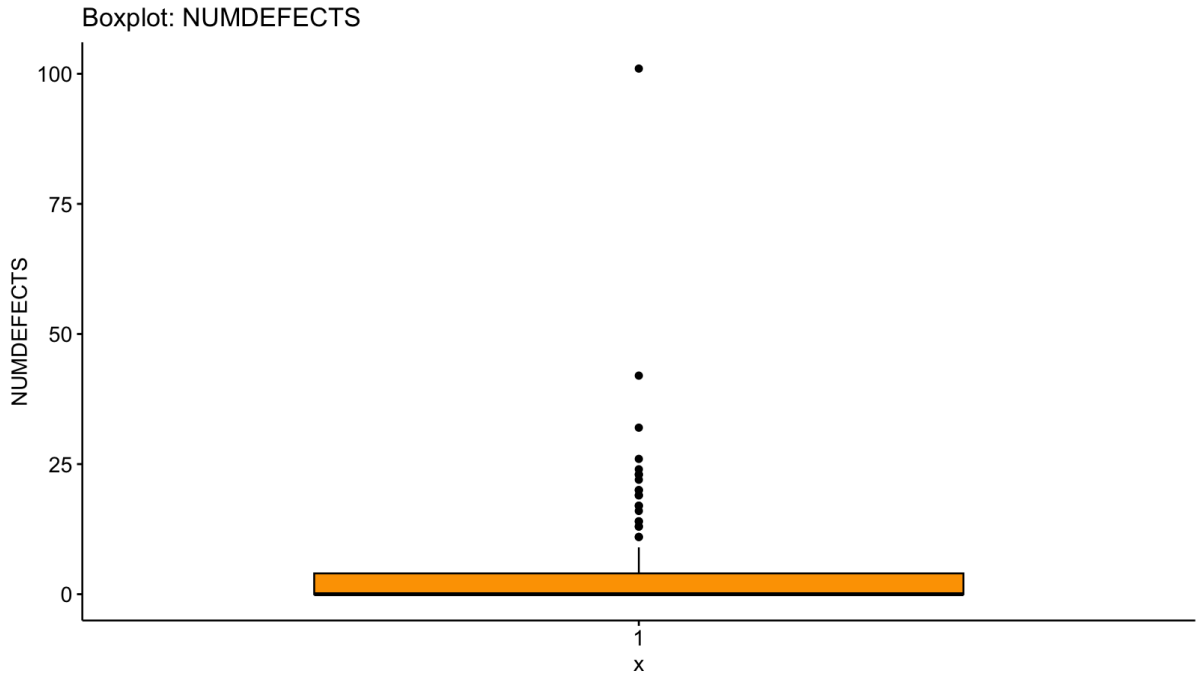




NUMDEFECTS

- Média: 4.61, mediana e moda 0 — maioria sem defeitos.
- Amplitude muito alta (0 a 101).
- Distribuição fortemente assimétrica e **não normal**.





3.2 Teste de Normalidade — Shapiro-Wilk

Para verificar se as variáveis numéricas seguem uma distribuição normal, foi aplicado o teste estatístico **Shapiro-Wilk Para** cada uma das oito variáveis do dataset. Este teste avalia a hipótese nula de que os dados são provenientes de uma população com distribuição normal.

A tabela apresenta os valores da estatística W, os p-values obtidos e a conclusão sobre a normalidade para cada variável.

Variável	Estatística W	p-valor	Normalidade
COUPLING_BETWEEN_OBJECTS	0.9390	< 0.001	Não
DEPTH_INHERITANCE	0.7763	< 0.001	Não
LACK_OF_COHESION_OF_METHODS	0.7418	< 0.001	Não
FAN_IN	0.7506	< 0.001	Não
RESPONSE_FOR_CLASS	0.7385	< 0.001	Não
WEIGHTED_METHODS_PER_CLASS	0.6735	< 0.001	Não
sumLOC_TOTAL	0.6110	< 0.001	Não
NUMDEFECTS	0.4547	< 0.001	Não

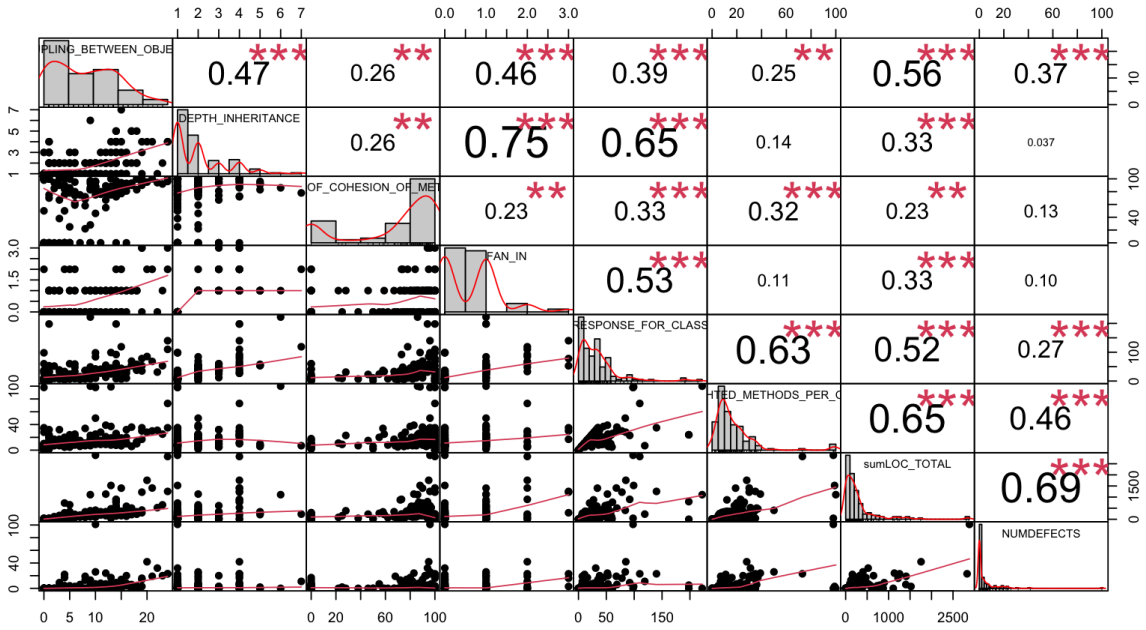
Resultados do teste Shapiro-Wilk para as variáveis numéricas do dataset.

Como os p-values para todas as variáveis foram menores que o nível de significância adotado (0,05), rejeitamos a hipótese nula de normalidade em todos os casos. Portanto, as distribuições das variáveis analisadas não podem ser consideradas normais.

Essa não normalidade justifica a atenção na escolha dos métodos estatísticos subsequentes, podendo ser necessário utilizar técnicas que não assumam normalidade, ou considerar transformações nos dados para atender a pressupostos de modelos estatísticos.

4. Análise de Correlação

A matriz de correlação gerada revela a força das relações entre as variáveis numéricas.

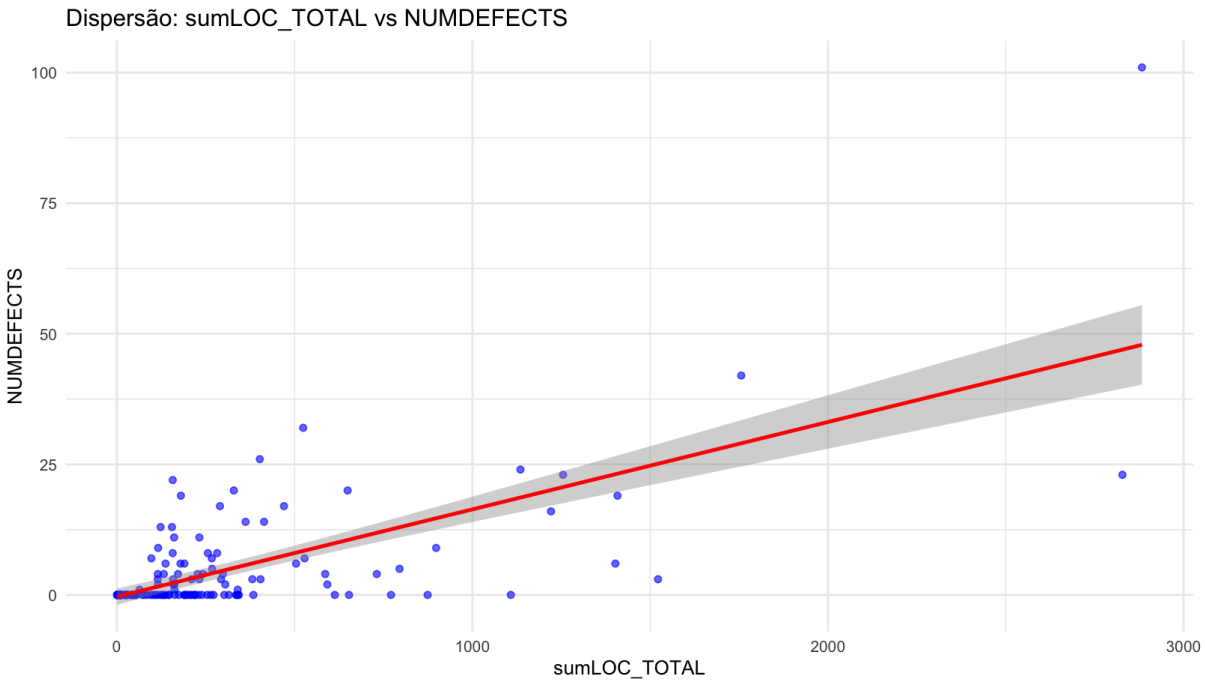


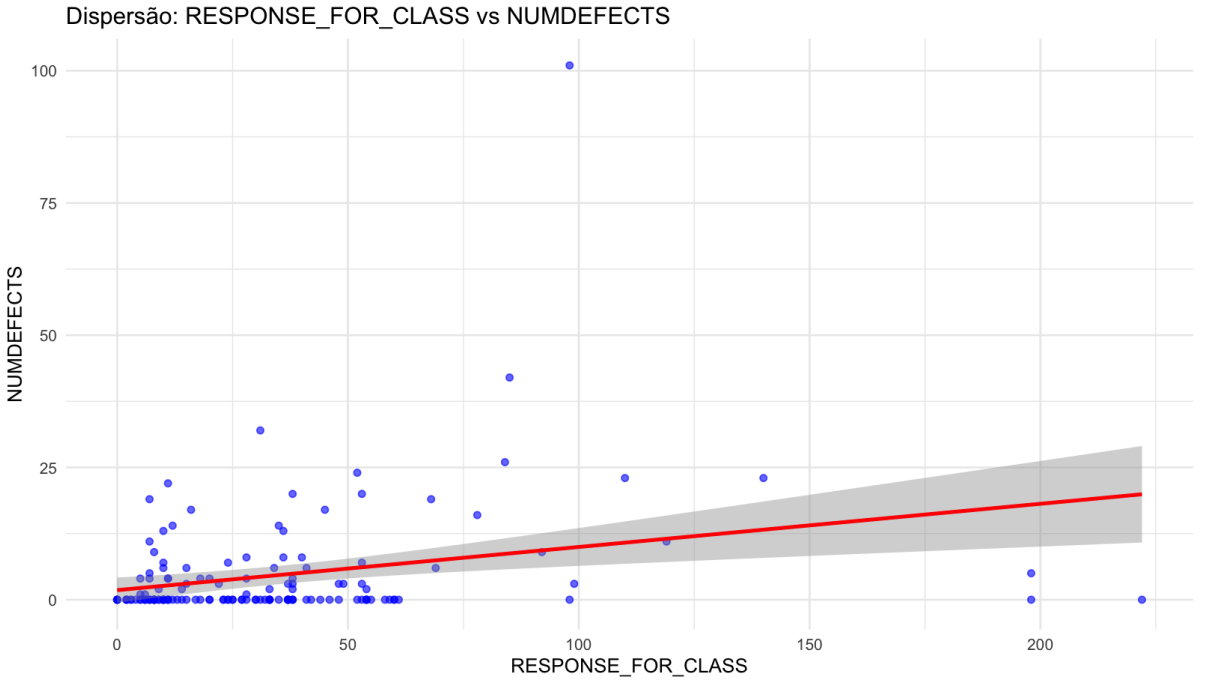
Interpretação:

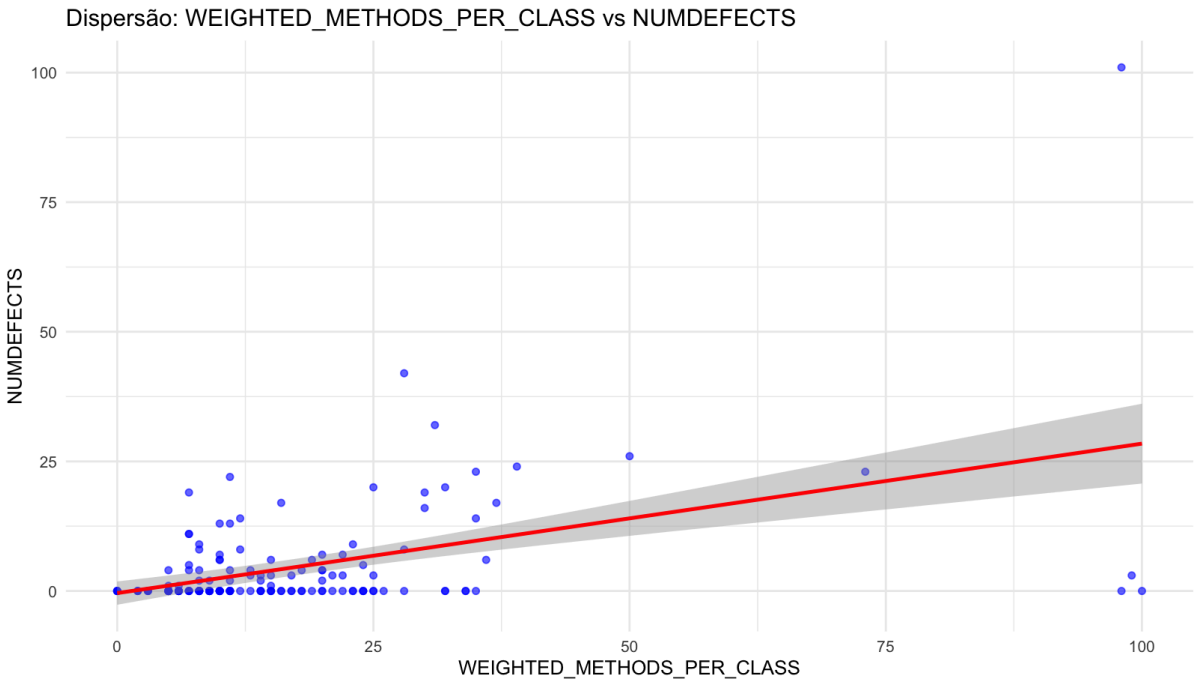
As variáveis com maior correlação com `NUMDEFECTS` são:

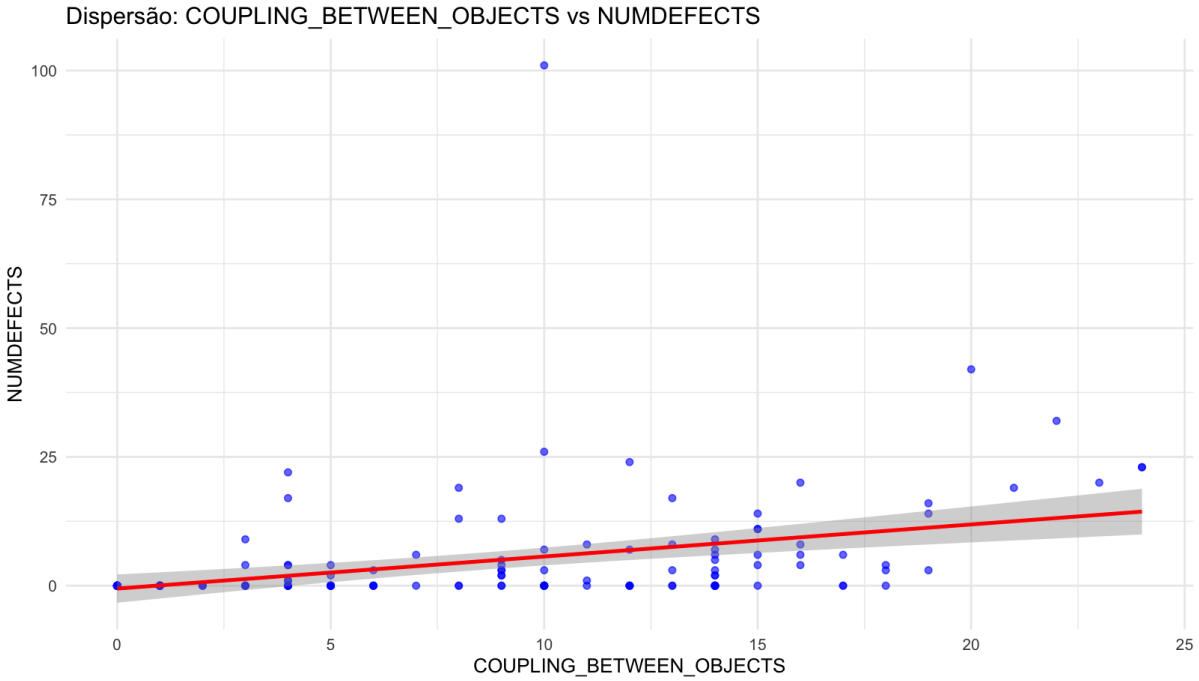
- `sumLOC_TOTAL`: **0.69** — Mais linhas de código tendem a resultar em mais defeitos.
- `RESPONSE_FOR_CLASS`: **0.63** — Classes que respondem a muitos estímulos têm mais defeitos.
- `WEIGHTED_METHODS_PER_CLASS`: **0.65** — Classes com mais métodos ponderados também apresentam mais defeitos.
- `COUPLING_BETWEEN_OBJECTS`: **0.56** — Acoplamento elevado está associado a mais defeitos.

Essas variáveis devem ser priorizadas na modelagem preditiva.


Dispersão: sumLOC_TOTAL vs NUMDEFECTS
NUMDEFECTS
sumLOC_TOTAL
0
25
50
75
100
0
1000
2000
3000


Dispersão: RESPONSE_FOR_CLASS vs NUMDEFECTS
NUMDEFECTS
RESPONSE_FOR_CLASS
0
25
50
75
100
0
50
100
150
200





Os gráficos reforçam que características relacionadas ao tamanho, complexidade e interdependência das classes influenciam diretamente a ocorrência de defeitos. Classes com maior quantidade de linhas de código, maior número e complexidade de métodos, além de maior acoplamento com outras classes, tendem a apresentar mais falhas.

Isso evidencia que métricas como **sumLOC_TOTAL**, **RESPONSE_FOR_CLASS**, **WEIGHTED_METHODS_PER_CLASS** e **COUPLING_BETWEEN_OBJECTS** são indicadores importantes para a previsão de defeitos, justificando sua inclusão prioritária no modelo preditivo.

5. Regressão Linear

Modelo Simples:

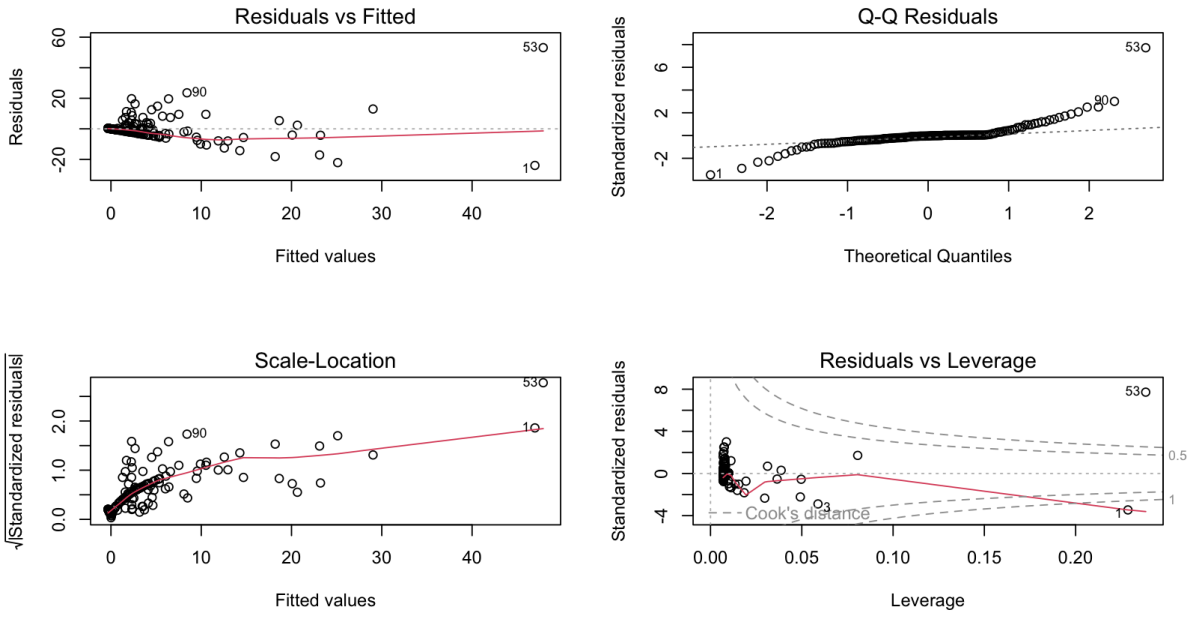
modelo <- lm(NUMDEFECTS ~ sumLOC_TOTAL, data = dados)

summary(modelo)

Saída:

- `sumLOC_TOTAL` foi utilizada por ser a mais correlacionada.
- Coeficientes indicam que um aumento em LOC impacta positivamente os defeitos.
- R^2 relevante e valores-p significativos.

Diagnóstico dos Resíduos:



Interpretação dos Gráficos:

- **Residuals vs Fitted**: indica leve tendência de heterocedasticidade (resíduos mais dispersos conforme aumenta o valor previsto).
- **Normal Q-Q**: há desvio dos resíduos das linhas teóricas, especialmente nas caudas — evidência de não normalidade.
- **Scale-Location**: confirma a heterocedasticidade crescente.
- **Residuals vs Leverage**: alguns pontos com alta influência (Cook's distance próximos de 1), mas poucos em número.

Esses gráficos indicam que, embora o modelo explique parte da variabilidade dos defeitos, há violação das suposições clássicas da regressão, sugerindo que uma abordagem mais robusta ou transformações possam melhorar o ajuste.

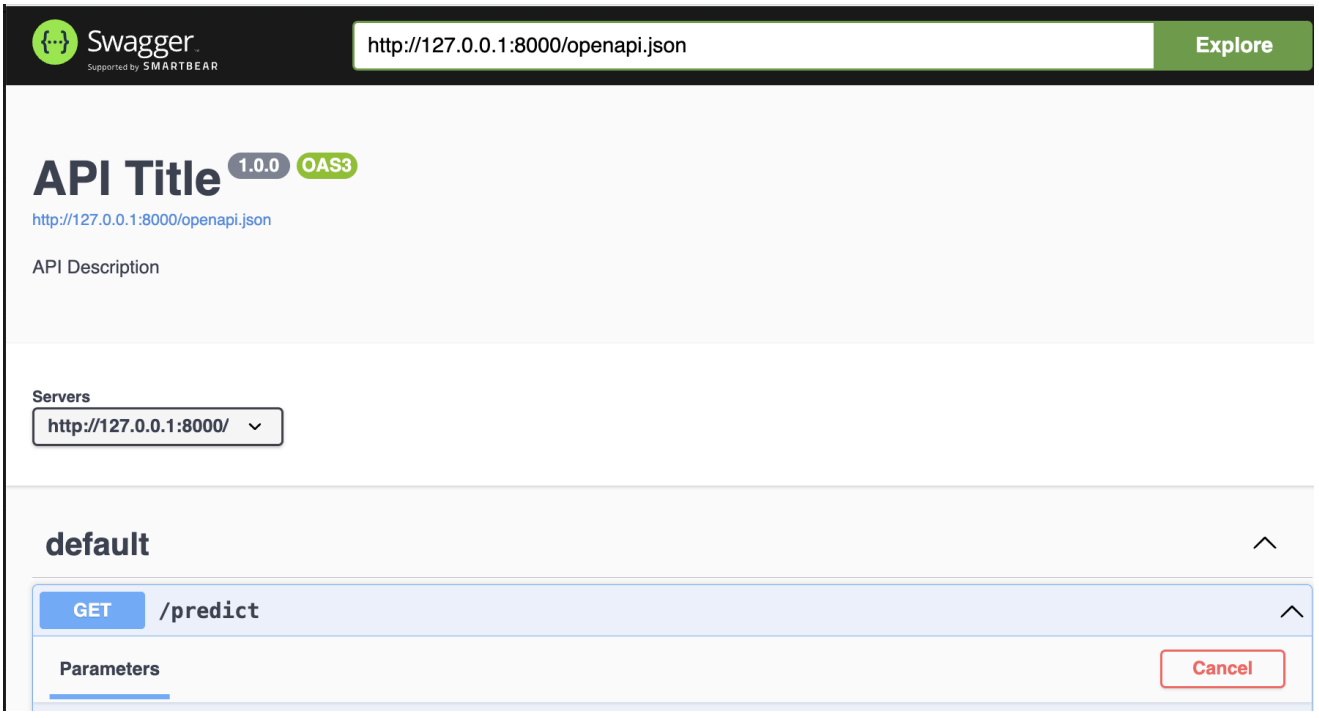
6. API REST com Plumber

Criei uma API REST usando o pacote **plumber** para expor o modelo de regressão linear que prevê o número de defeitos (`NUMDEFECTS`) a partir da métrica `sumLOC_TOTAL`. A API recebe o valor de `sumLOC_TOTAL` via requisição GET, realiza a previsão e retorna o resultado em JSON.

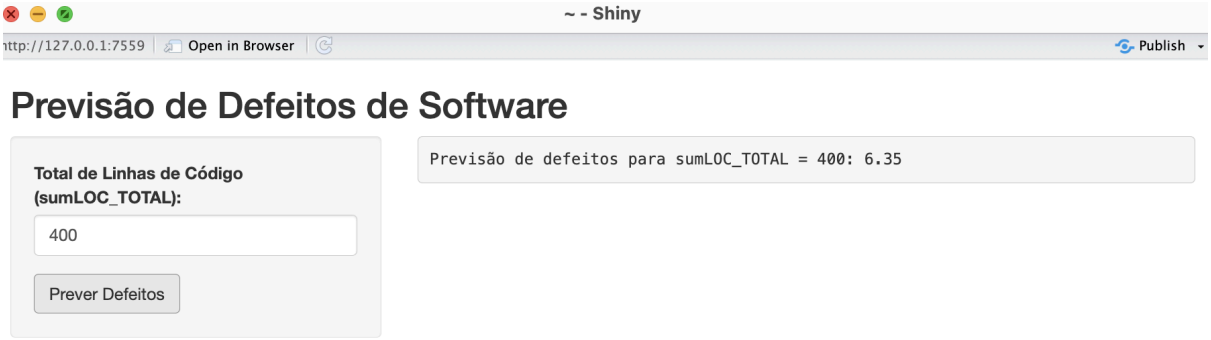
Durante a implementação, enfrentei desafios com o carregamento do modelo dentro do ambiente da API e com a atualização da sintaxe do pacote `plumber`. Também precisei validar os parâmetros de entrada para evitar erros. A execução simultânea da API e da aplicação Shiny exigiu cuidados na configuração das portas.

No final, a API funcionou corretamente, permitindo a integração com a aplicação Shiny e facilitando a modularização do projeto.

Segue imagem da evidência de que a API rodou corretamente.



Logo abaixo a imagem do shiny conectado e rodando o mesmo numero



7. Aplicação Shiny

Após finalizar o desenvolvimento da aplicação Shiny e da API REST, procedi com a publicação no serviço shinyapps.io. Preparei o pacote para o deploy, incluindo o arquivo `app.R` e o dataset necessário.

Durante o processo de implantação, ocorreu um erro relacionado à versão do R utilizada pelo servidor da plataforma. O shinyapps.io não suporta ainda a versão 4.5.1 do R, que foi a versão usada localmente para desenvolver o projeto. A mensagem de erro exibida foi:

```
Unhandled Exception: Unsupported R version 4.5.1 for operating syste
```

Devido a essa limitação, não foi possível concluir a publicação da aplicação nesse ambiente. Para contornar esse problema, seria necessário utilizar uma versão do R compatível com o shinyapps.io, o que ficou fora do escopo deste trabalho.

Segue print do erro obtido na tentativa de publicação.

```
>
> rsconnect::deployApp()

── Preparing for deployment ──────────────────────────────────────────────────
✔ Deploying "shinydefeitos" using "server: shinyapps.io / username: lgbrito"
ℹ Creating application on server...
✔ Created application with id 15010546
ℹ Bundling 2 files: app.R and dataset_KC1_classlevel_numdefect.xlsx
ℹ Capturing R dependencies
✔ Found 42 dependencies
✔ Created 38,378b bundle
ℹ Uploading bundle...
✔ Uploaded bundle with id 10540273
── Deploying to server ───────────────────────────────────────────────────────

Waiting for task: 1562325957
  building: Parsing manifest
  error: Building image: 12968827
## Begin Task Log ##################################################################
## End Task Log ####################################################################

                                                                Show Traceback
                                                                Rerun with Debug
Error: Unhandled Exception: child_task=1562325959 child_task_status=error: Unhandled Exception: Unsupported R version 4.5.1 for operating system jammy.
```

8. Conclusão

Este trabalho explorou a análise estatística e a modelagem preditiva aplicadas a métricas de qualidade de software, com base no dataset `KC1_classlevel_numdefect.xlsx`. Foram aplicadas técnicas de estatística descritiva, análise de correlação e regressão linear, revelando que métricas como `sumLOC_TOTAL`, `RESPONSE_FOR_CLASS` e `WEIGHTED_METHODS_PER_CLASS` estão fortemente associadas ao número de defeitos nas classes.

O modelo gerado foi integrado a uma API REST desenvolvida com o pacote **plumber**, permitindo acesso externo à previsão. Além disso, foi criada uma aplicação interativa com **Shiny**, que conecta-se à API e permite ao usuário realizar previsões de forma simples.

Apesar de dificuldades técnicas, como a execução simultânea da API e do app, e a incompatibilidade da versão do R com o shinyapps.io, os objetivos principais foram alcançados com sucesso. O resultado final demonstra a viabilidade de utilizar ferramentas do R para análise de dados e construção de soluções web interativas baseadas em modelos estatísticos.