Estudo de caso: Grupo D 3

Gilmar Pereira, Maressa Tavares e Victor Ruela

3 de Setembro, 2019

1 Summary

O presente trabalho realizou o delineamento e executou os testes estatísticos para avaliar uma nova versão de um software, em relação aos resultados obtidos na versão anterior. Tendo em vista que a última versão possui uma distribuição de custos com média $\mu = 50$ e variância $\sigma^2 = 100$, dados da população, objetiva-se verificar se a nova versão apresenta resultados melhores para tais características. Para tanto, utilizou-se o teste t com nível de significância $\alpha = 0.01$ e $\alpha = 0.05$, para os testes de média e variância, respectivamente. Após os testes verificou-se que.....

2 Planejamento do experimento

Nesta seção são apresentados os detalhes do delineamento dos testes que foram executados para comparar o desempenho das duas versões do software em relação à média e à variância do custo de execução. Essa etapa é fundamental, pois trata-se de uma abordagem que fornece resultados importantes em análises de sistemas complexos, além disso, os testes servem para validar a teoria que está por trás dele [1].

2.1 Objetivo do experimento

Para a versão atual de um dado sistema, sabe-se que sua distribuição de custos de execução possui média populacional de $\mu = 50$ e variância $\sigma^2 = 100$. Uma nova versão desse software foi desenvolvida, portanto realizou-se uma análise estatística para investigar os ganhos de desempenho obtidos em relação à versão atual.

Inicialmente o teste foi executado para as médias do custo, assim, para verificar se a nova versão é melhor que a anterior, foram formuladas as seguintes hipóteses:

$$\begin{cases} H_0 : \mu < 50 \\ H_1 : \mu \geq 50 \end{cases}$$

Como a média da população para a primeira versão é $\mu = 50$, considerou-se como hipótese alternativa (H_1) a ausência de melhoria do software, isto é, a segunda versão apresenta uma performance pior ou igual à versão anterior, com média maior ou igual a 50, $\mu \geq 50$. Por outro lado, a hipótese nula, complementar à alternativa, considera que houve melhorias entre as versões, portanto, a média é menor que 50 (H_0).

Além disso, para o teste da média foram definidos os seguintes objetivos:

- Nível de significância desejado $\alpha = 0.01$. Logo, o nível de confiança desejado é $1 - \alpha = 0.99$
- Efeito relevante mínimo de $\delta^* = 4$
- Potência desejada $\pi = 1 - \beta = 0.8$

Por outro lado, para a variância o experimento foi realizado com base nas seguintes hipóteses:

$$\begin{cases} H_0 : \sigma^2 < 100 \\ H_1 : \sigma^2 \geq 100 \end{cases}$$

Assim como no teste da média, neste caso adotou-se como hipótese alternativa (H_1) a ausência de melhoria do software, mantendo os resultados de variância da versão anterior ($\sigma^2 = 100$). Enquanto a hipótese nula considera que houve melhorias entre as versões, portanto, a variância é menor que 100 (H_0).

Em relação aos objetivos, o teste da variância considerou:

- $\alpha = 0.05$
- $1 - \alpha = 0.95$

Os dois testes foram realizados com os mesmos dados coletados conforme a descrição da próxima seção.

2.1.1 Descrição da coleta de dados

Para coletar os dados referente à nova versão do software, foi executada uma simulação no software R utilizando a biblioteca *ExpDE* [2]. A coleta de dados foi declarada da seguinte forma:

```
# Set-up the data generating procedure
mre <- list(name = "recombination_bin", cr = 0.9)
mmu <- list(name = "mutation_rand", f = 2)
mpo <- 100
mse <- list(name = "selection_standard")
mst <- list(names = "stop_maxeval", maxevals = 10000)
mpr <- list(name = "sphere", xmin = -seq(1, 20), xmax = 20 + 5 * seq(5, 24))

#set.seed(1235) # to generate always the same results

# define functions for data generation
get.single.sample <- function(mpo, mmu, mre, mse, mst, mpr){
  generator <- ExpDE(mpo, mmu, mre, mse, mst, mpr, showpars = list(show.iters = "none"))
  return(generator$Fbest)
}

get.n.samples <- function(mpo, mmu, mre, mse, mst, mpr, N){
  if(!file.exists('CS01data.csv')){
    my.sample <- numeric(N)
    for (i in seq(N)){
      my.sample[i] <- get.single.sample(mpo, mmu, mre, mse, mst, mpr)
    }

    write.csv(my.sample, file = 'CS01data.csv', row.names = FALSE)
    return(my.sample)
  }
  else{
      return(read.csv('CS01data.csv')$x)
  }
}
```

As funções `get.single.sample` e `get.n.samples` foram criadas para facilitar o entendimento da função de geração de dados, sendo elas para coletar uma única amostra ou n amostras, respectivamente.

3 Resultados

3.1 Teste sobre a média do custo

3.1.1 Cálculo do tamanho amostral

Baseado nas informações preliminares do problema, $\sigma^2 = 100$, $\delta^* = 4$ e $\pi = 0.8$, e dado que estamos considerando uma hipótese alternativa maior ou igual a média μ, o cálculo do tamanho amostral pode ser estimado com a função `power.t.test`:

```r
# define current system parameters
current_mu <- 50
current_var <- 100

# define mean cost test parameters
sig_level_mean <- 0.01
delta <- 4
beta <- 0.2
pi <- 1 - beta
ci_mean <- 1 - sig_level_mean

# use the function inivisble() to supress the function console output
invisible(sample_size_calc <- power.t.test(delta = delta,
                           sd = sqrt(current_var),
                           sig.level = sig_level_mean,
                           power = pi,
                           alternative = "one.sided",
                           type = "one.sample"))
# round to the next integer
N <- ceiling(sample_size_calc$n)
```

Resultando em um tamanho amostral de 66.

Definido o tamanho da amostra, procedeu-se a coleta da amostra que foi utilizada em todos os testes nas próximas seções.

3.1.2 Análise Exploratória dos Dados

Com base nas amostras coletadas referente à segunda versão do software, foi realizada uma análise exploratória dos dados a fim de validar as premissas dos testes que foram para a média e variância.

Antes de proceder com as análises estatíticas e realizar as inferências sobre o problema, é importante realizar uma análise preliminar dos dados. [3] destaca que a análise exploratória tem o papel de extrair informações dos dados antes de realizar as inferências estatísticas, a fim de obter os modelos plausíveis para cada estudo.

```r
data.mean.test <- get.n.samples(mpo, mmu, mre, mse, mst, mpr, N)
summary(data.mean.test)
```

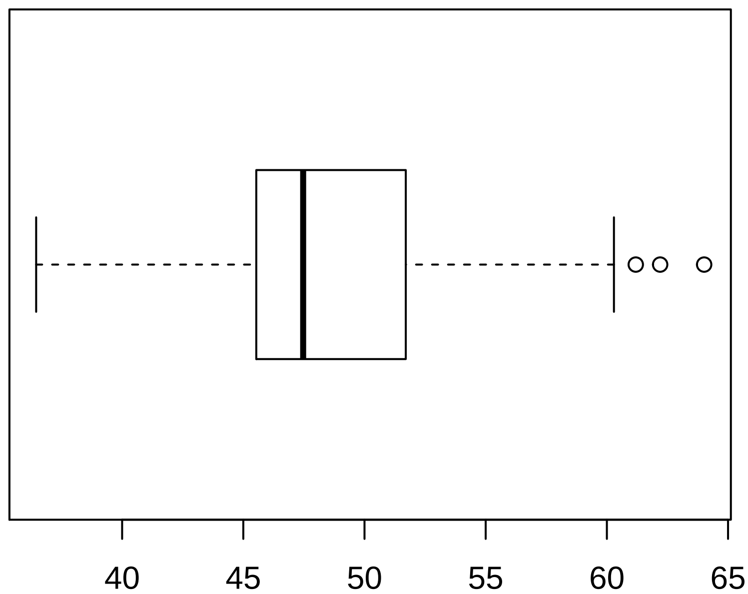
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   36.45   45.59   47.47   49.04   51.66   64.01
```

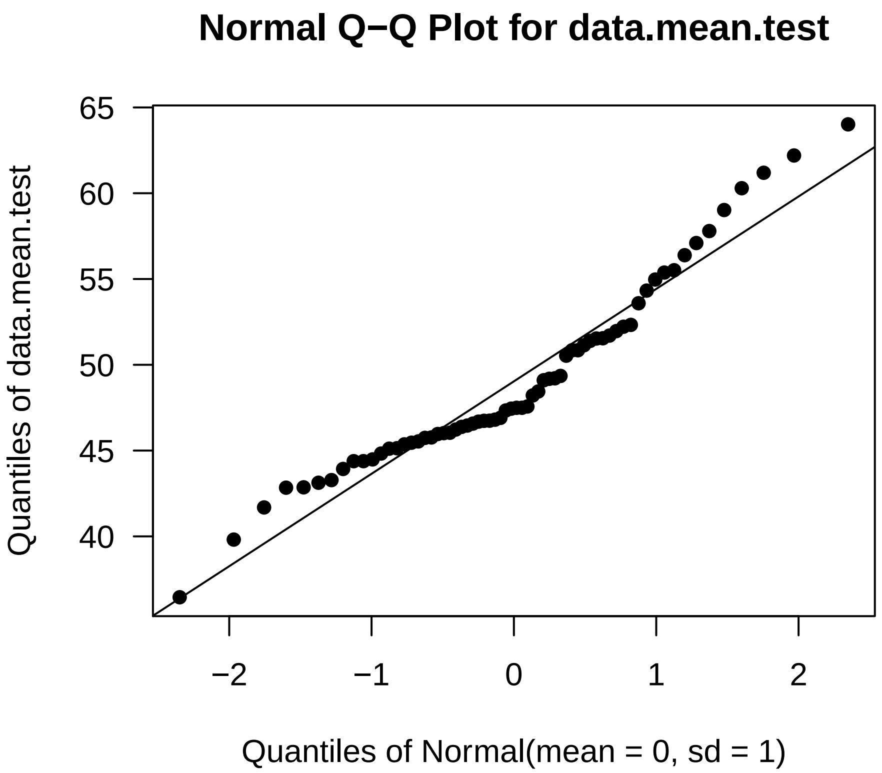
```r
var(data.mean.test)
```

```
## [1] 29.7873
```

Pela análise dos dados verifica-se que a variância do novo software aparenta ser significamente menor que a versão anterior, por outro lado, em relação à média a diferença foi mais discreta. Para concluir com maior precisão sobre os dados, os testes estatísticos serão apresentados com detalhes nas seções seguintes.

Outras análises importantes no contexto da análise exploratória, é de outliers, através de um boxplot, e o teste da normalidade dos dados por meio do gráfico quantil x quantil e do teste de Shapiro Wilk.


40
45
50
55
60
65

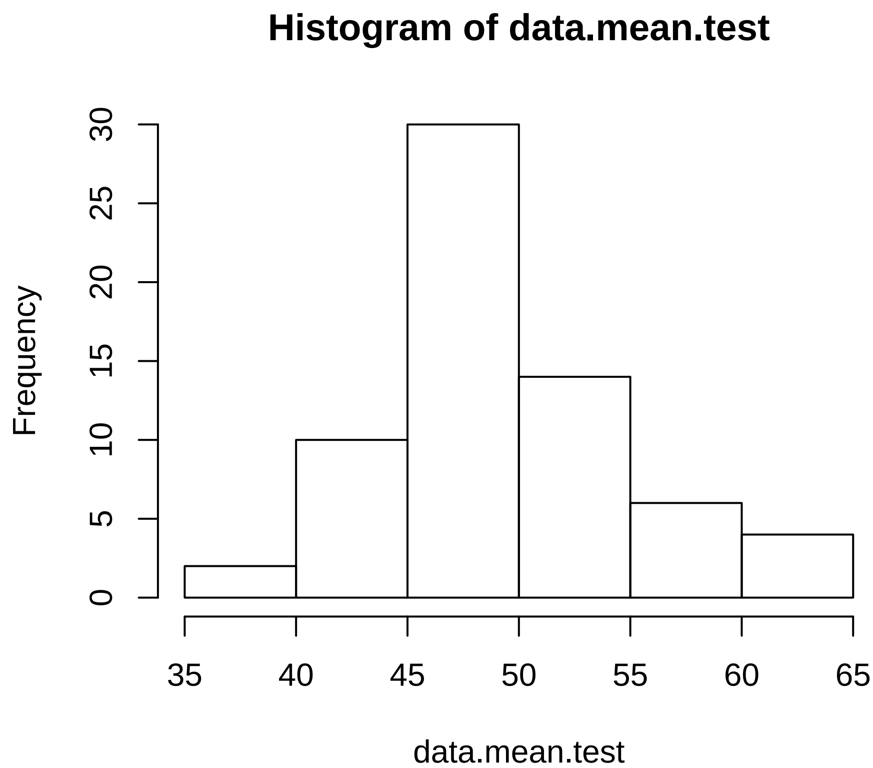



```r
shapiro.test(data.mean.test)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data.mean.test
## W = 0.94725, p-value = 0.007161
```

Pela análise do boxplot verificou-se que os dados são assimétricos positivos, pois há um deslocamento da mediana em direção ao primeiro quartil, além disso, observam-se três pontos considerados como outliers da amostra.

Com base no gráfico quantil X quantil e no teste de Shapiro Wilk verifica-se que o teste falhou em aceitar a hipótese de normalidade dos dados, pois o p-valor do teste é menor que 0.05 (0.007161).



Sendo assim, partindo da premissa de normalidade dos dados relacionados ao custo médio do software é possível prosseguir com as análises estatísticas que serão apresentadas nas seções a seguir.

3.1.3 Teste de Hipoteses

Mesmo havendo falha na aceitação da hipotese de normalidade dos dados, para testar a hipotese nula para a média utilizou-se a função t.teste. É possivel a utilização deste teste devido normalidade alcançada por uma reamostragem, verificado pelo o teorema do limite central.

A análise foi realizada sobre os dados amostrados usando a função get.n.samples, descrita na seção 2.1.1, para 66 amostras. O teste esta descrito a seguir.

```r
#teste para custo médio:
(mean.t.teste <- t.test(data.mean.test,
                  mu=current_mu,            #hipotese nula
                  alternative = "greater",  #hipotese alternativa
                  conf.level = ci_mean))
```

```
##
##  One Sample t-test
##
## data:  data.mean.test
## t = -1.4347, df = 65, p-value = 0.9219
## alternative hypothesis: true mean is greater than 50
## 99 percent confidence interval:
##  47.43381      Inf
## sample estimates:
## mean of x
```

```
## 49.03614
```

O t teste retorna que a hipotese alternativa é verdadeira para uma média maior que 50. Porém, pode-se verifcar que o p-valor é maior do que o nível de significância estabelecido ($\alpha = 0.01$). Desta maneira não há evidência suficiente para se rejeitar a hipótese nula.

3.1.4 Calculo do intervalo de confianca

O intervalo de confiança é retornado pela função t.test, conforme mostrado a seguir.

```
mean.t.teste$conf.int
```

```
## [1] 47.43381      Inf
## attr(,"conf.level")
## [1] 0.99
```

O valor da media retornado pelo t teste é facilmente verificado dentro do intervalo de confiança calculado, reafirmando, desta maneira, que é provavel a não rejeição da hipotese nula.

3.1.5 Validação de premissas

Para validação das premissas utilizadas no t teste, realizou-se uma reamostragem dos dado, conforme descrito a seguir.

```
#reamostragem:
N2<-999
means.re<-numeric(N2)
for(i in seq(N2)){
  means_sample<-sample(data.mean.test, replace = TRUE)
  means.re[i]<-mean(means_sample)
}
```

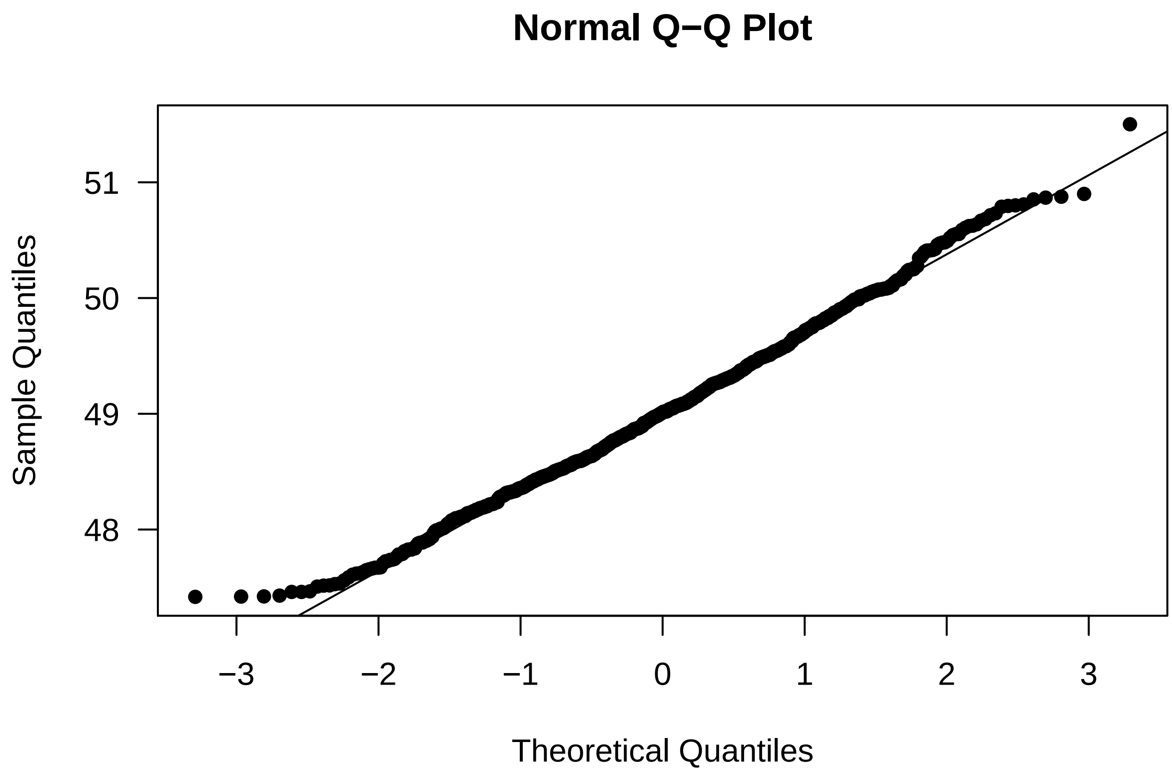
A media para a reamostragem é dada por:

```
mean(means.re)
```

```
## [1] 49.02243
```

Graficamente:

```
#normal
qqnorm(means.re, las = 1, pch = 16)
qqline(means.re)
```



3.1.6 Potencia do teste

Para determinar a potência do teste, primeiramente definiu-se o intervalo superior de confiança, a partir do χ^2, para a variância, conforme equação mostrada.

$$\sigma^2 \leq \frac{(N-1)s^2}{\chi_{\alpha}^{2(N-1)}}$$

O calculo do intervalo superior foi realizado da seguinte maneira.

```
#intervalo superior unilateral de confiança
intervalo_conf_max<-(N-1)*var(data.mean.test)/qchisq(p = 0.01, df = N-1)
intervalo_conf_max
```

```
## [1] 46.71828
```

Com o valor retornado, foi possivel realizar o teste de potência, utilizando a função power.t.test, conforme descrito abaixo.

```
#calculo da potencia
(potencia<-power.t.test(n = N, delta = delta, sd = sqrt(intervalo_conf_max), sig.level = 0.01, type = "o
```

```
##
##      One-sample t test power calculation
##
##               n = 66
##           delta = 4
##              sd = 6.835077
```

```
##       sig.level = 0.01
##           power = 0.9900339
##     alternative = one.sided
```

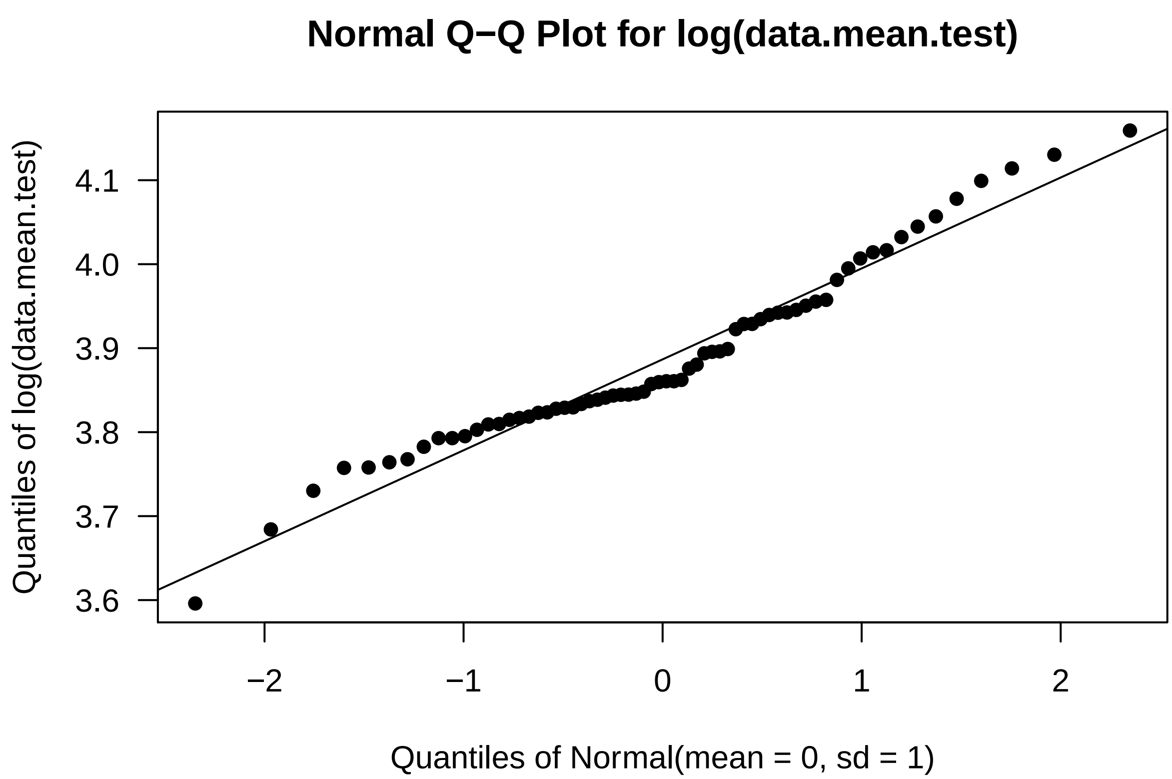
A variância encontrada na análise exploratória é substancialmente menor que a variancia determinada para a versão do software anterior, $\sigma = 100$. Isso bastaria para varificar que a potência do teste é grande o suficiente para validar as premissas apresentadas. No resultado da potência do teste retorna, como esperado, um valor de 0.9900339, confirmando assim a veracidade dos testes realizados.

3.2 Teste sobre a variância do custo

3.2.1 Teste de Hipoteses

Para a variância, dado que a população não é modelada por uma distribuição normal (vide análise exploratória), a estatística de teste não irá seguir uma distribuição chi-quadrado, logo é necessário aplicar uma transformação que leve os dados à normalidade ou utilizar técnicas não-paramétricas. Uma transformação possível é a logarítimica:

```
qqPlot(log(data.mean.test), pch = 16, las = 1, add.line = TRUE)
```




```
shapiro.test(log(data.mean.test))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  log(data.mean.test)
## W = 0.96592, p-value = 0.06649
```

Pelo gráfico quantil-quantil e p-valor baixo obtido no teste de Shapiro, conclui-se que esta transformação não é capaz de levar os dados à normalidade. Logo, neste trabalho, será usado a técnica de bootstraping para a estimativa do intervalo de confiança e execução do teste de hipóteses [4]. Será utilizado o pacote *boot* [5] para a sua execução.

```
# run the booootstrapping
set.seed(12345) # set a fixed seed to yield the same results for bootstrapping always
data.var.test.boot <- boot(data.mean.test, statistic = function(x, i){var(x[i])}, R=1000)

# define the desired significance level and CI
sig_level_sd <- 0.05
ci_sd <- 1 - 2 * sig_level_sd
(test.boot.var <- boot.ci(data.var.test.boot, conf = ci_sd, type = "bca"))
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = data.var.test.boot, conf = ci_sd, type = "bca")
##
## Intervals :
## Level       BCa
## 90%   (21.48, 41.39 )
## Calculations and Intervals on Original Scale
```

É importante notar que o método acima calcula o intervalo de confiança para uma hipótese bilateral. Portanto, foi necessário ajustar o nível de confiança para 90%, de forma a ter uma taxa de erro de 0.05 em cada extremidade do intervalo. Como o interesse é somente no intervalo superior, podemos ignorar a extremidade interior e assumir que a nova versão do software possui variância inferior a 41.3914099 com 95% de confiança. Logo, a hipótese nula é rejeitada, pois a variância é significativamente inferior à versão atual do software.

4 Discussão e Conclusões

5 Divisão das Atividades

Victor - Reporter Maressa - Coordenadora Gilmar - Verificador e Monitor

Referências

[1] D. C. Montgomery and G. C. Runger, *Applied statistics and probability for engineers, (with cd)*. John Wiley & Sons, 2007.

[2] M. B. Felipe Campelo, "CRAN - package expde - modular differential evolution for experimenting with operators." https://cran.r-project.org/web/packages/ExpDE/index.html, Jan-2018.

[3] W. Medri, "Análise exploratória de dados." http://www.uel.br/pos/estatisticaquantitativa/textos_didaticos/especializacao_estatistica.pdf, 2011.

[4] A. C. Davison and D. V. Hinkley, *Bootstrap methods and their applications*. Cambridge: Cambridge University Press, 1997.

[5] A. Canty and B. D. Ripley, *Boot: Bootstrap r (s-plus) functions*. 2019.