

# 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 de hipótese com nível de significância  $\alpha = 0.01$  e  $\alpha = 0.05$  para a média e variância, respectivamente. Após os testes verificou-se que o novo sistema possui variância significativamente inferior à versão atual, porém, não podemos afirmar o mesmo para a média com a significância especificada.

## 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:

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

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

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

Assim como no teste da média, neste caso adotou-se como hipótese alternativa ( $H_1$ ) a presença de melhoria do software, com valores menores que a variância da versão anterior ( $\sigma^2 = 100$ ). Enquanto a hipótese nula considera que não houve melhorias entre as versões, portanto, a variância é maior ou igual a 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.its = "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. É importante notar também que o método `get.n.samples` carrega os valores de um arquivo especificado, se existente, de forma a garantir a replicabilidade dos resultados.

## 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 menor que a média  $\mu$ , o cálculo do tamanho amostral pode ser

estimado com a função `power.t.test`:

```
# 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 invisible() 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 executados para a média e variância.

Antes de proceder com as análises estatísticas e realizar as inferências sobre o problema é importante realizar uma análise preliminar dos dados. 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 [3].

```
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
```

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

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

Pela análise dos dados verifica-se que a variância do novo software aparenta ser significativamente 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.

Outra análise importante no contexto da análise exploratória, é a 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.

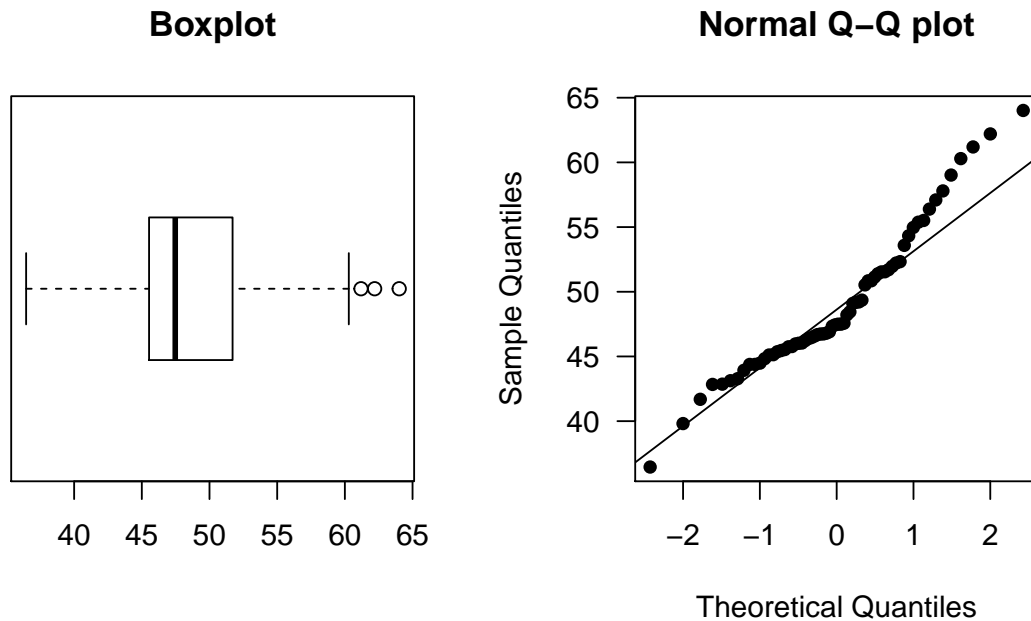


Figure 1: Boxplot e gráfico quantil-quantil das amostras

```
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 (p-value = 0.007161).

Sendo assim, partindo da premissa de não 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 Hipóteses

Mesmo havendo falha na aceitação da hipótese de normalidade dos dados, para testar a hipótese nula para a média utilizou-se o teste t (função `t.test`). É possível a utilização deste teste devido à normalidade alcançada por uma reamostragem maior, verificado pelo teorema do limite central, como será descrito na validação das premissas (seção 3.1.5).

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 está descrito a seguir.

```
#teste para custo médio:
(mean.t.teste <- t.test(data.mean.test,
```

```
mu=current_mu,          #hipótese nula
alternative = "less",   #hipótese alternativa
conf.level = ci_mean))
```

```
##
## One Sample t-test
##
## data: data.mean.test
## t = -1.4347, df = 65, p-value = 0.07808
## alternative hypothesis: true mean is less than 50
## 99 percent confidence interval:
##      -Inf 50.63846
## sample estimates:
## mean of x
## 49.03614
```

Analisando o p-valor retornado pelo teste t,  $p = 0.078078$ , verifica-se que é maior que o nível de significância  $\alpha = 0.01$  determinado, portanto não há evidências para rejeitar a hipótese nula. Sendo assim, com base na amostra não é possível concluir que houve melhoria em relação à média do custo entre as duas versões.

### 3.1.4 Cálculo do intervalo de confiança

O intervalo de confiança da hipótese nula é calculado conforme equação mostrada abaixo.

$$\bar{X} + \frac{t_{\alpha} s}{\sqrt{N}} \leq \mu$$

```
t_a<-2.6603
Lim_sup<-mean(data.mean.test)-sqrt((t_a*sd(data.mean.test)/sqrt(N))^2)
```

O intervalo de confiança da hipótese nula é de 47.2489313 a infinito. Sendo assim, como a média ( $\mu = 50$ ) está contida nesse intervalo, corrobora com o teste anterior, que a média não é inferior à da versão anterior.

### 3.1.5 Validação de premissas

Para validação das premissas utilizadas no teste t, realizou-se uma reamostragem dos dados para uma amostra maior, 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 média para a reamostragem é dada por:

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

```
## [1] 49.03318
```

Graficamente:

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

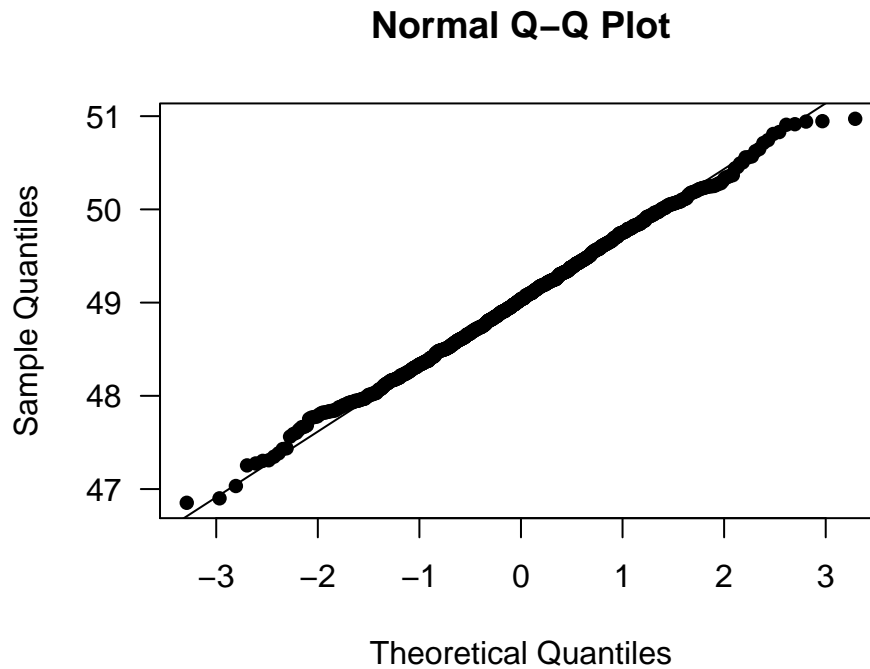


Figure 2: Gráfico quantil-quantil para reamostragem da média

Analisando o gráfico quantil-quantil para a reamostragem, é fácil atestar a normalidade da distribuição da média, atestando a normalidade dos dados com base no Teorema do Limite Central.

### 3.1.6 Potência do teste

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

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

O cálculo 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 possível realizar o teste de potência, utilizando a função `power.t.test`, conforme descrito abaixo.

```
#calcula da potencia
(potencia<-power.t.test(n = N, delta = delta, sd = sqrt(intervalo_conf_max),
  sig.level = 0.01, type = "one.sample",
  alternative = "one.sided"))
```

```
##
```

```
##      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 variância obtida para a versão anterior do software,  $\sigma^2 = 100$ . Isso bastaria para varificar que a potência do teste é grande o suficiente para validar as premissas apresentadas. Sendo assim, o 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 Hipóteses

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 logarítmica:

```
qqnorm(log(data.mean.test), pch = 16, las = 1)
qqline(log(data.mean.test))
```

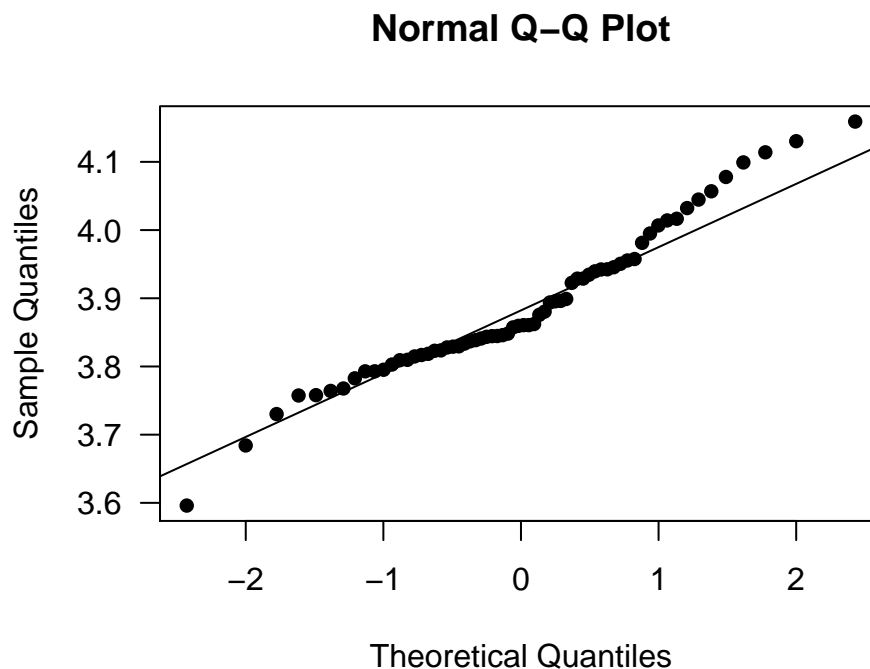


Figure 3: Gráfico quantil-quantil para transformação logarítmica dos dados

```
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, foi usado a técnica de bootstrapping para a estimativa do intervalo de confiança e execução do teste de hipóteses [4]. Foi utilizado o pacote *boot* [5] para a sua execução.

```
# run the bootstrapping
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 inferior 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 Conclusão

Neste trabalho foi feito um estudo estatístico dos ganhos de desempenho de uma nova versão de um software, em relação à média e variância dos custos. Através de uma análise exploratória, foi possível constatar que as amostras da nova versão não seguem uma distribuição normal. Partindo desta premissa, testes de hipóteses foram realizados e, para os níveis de significância especificados, é possível concluir que a variância é significativamente inferior, porém o mesmo não pode ser afirmado para a média. De acordo com os testes realizados pode-se concluir que o software atual não possui ganho de desempenho considerável em relação ao software anterior.

## 5 Divisão das Atividades

- Victor - Relator
- 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](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.