Estudo de caso: Grupo D 3

Gilmar Pereira, Maressa Tavares e Victor Ruela

3 de Setembro, 2019

1 Summary

O presente trabalho tem 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 $\sigma = 100$, 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, formulou-se as seguintes hipóteses:

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

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

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:

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

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

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 os dados de acordo com 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 unilateral para a média amostral, 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:

```
## [1] N =  66
```

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 realizados em seguida, conforme apresentado na seção seguinte.

Antes de proceder com as análises estatíticas e realizar as inferências sobre o problema, é importante realizar uma análise preliminar dos dados **???**. 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)
```

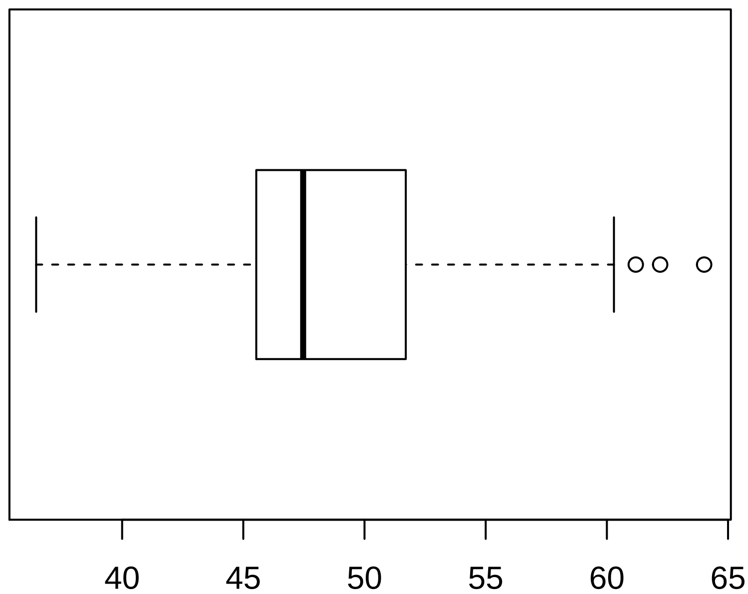
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   36.45   45.59   47.47   49.04   51.66   64.01
```

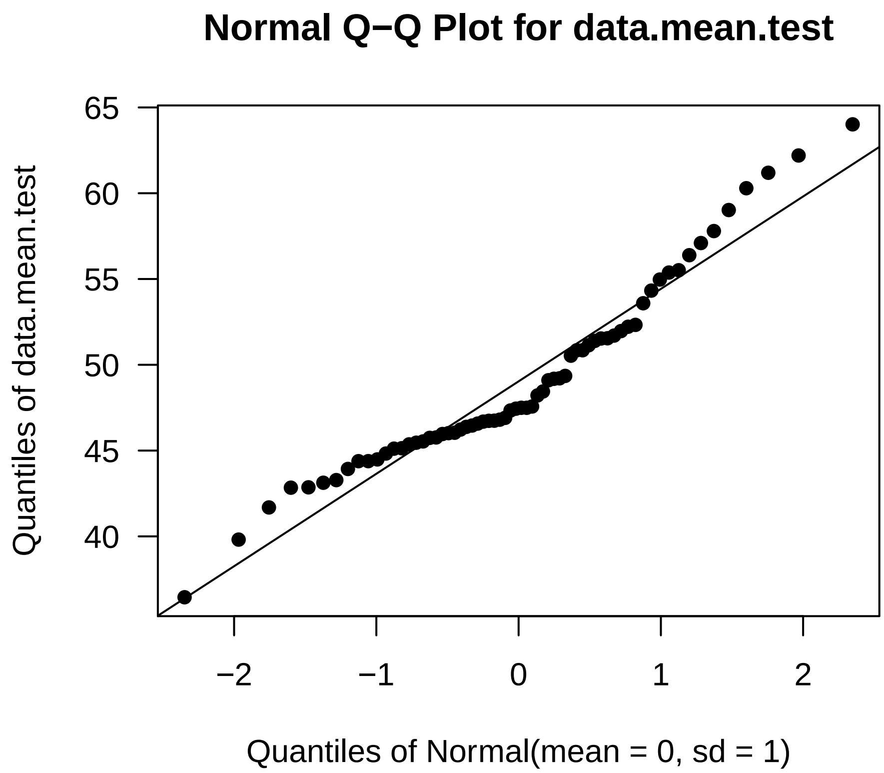
```r
var(data.mean.test)
```

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

Pela análise dos dados verifica-se que a variância do novo software aparenta ser significamente menor que a versão anterior, enquanto 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.

No contexto da análise exploratória, é válido realizar o teste de outliers, através de um boxplot, e o teste da normalidade dos dados por meio do gráfico quantil x quantil.

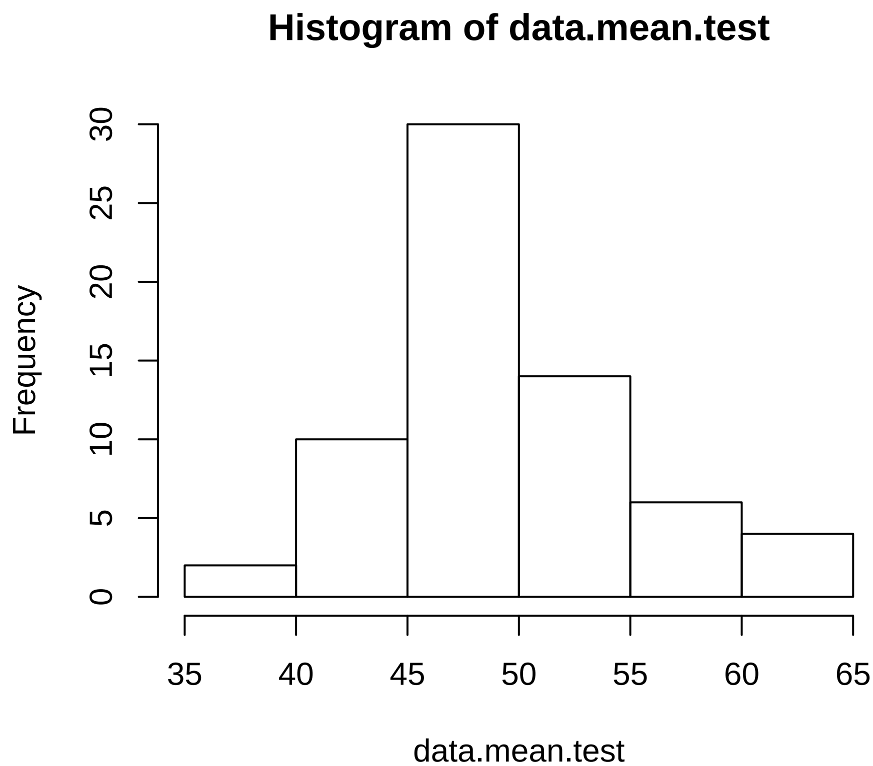

40
45
50
55
60
65




```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 e do gráfico quantil X quantil, verifica-se que, embora o valor máximo observado (65.08) apresenta-se como um outlier no boxplot, ele não interfere na normalidade dos dados. Além da análise gráfica, o teste de Shapiro-Wilk também comprova a normalidade dos dados, pois o p-valor do teste é maior que 0.05, logo, o teste falhou em rejeitar a hipótese nula de que os dados têm distribuição normal, o histograma dos dados apresentado a seguir também coorrobora com essa conclusão.



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

Para testar a hipotese nula para o custo médio utilizou-se a função t.teste, sobre os dados amostrados usando a função get.n.samples, descrita na seção 2.1.1, para N amostras. O teste esta descrito a seguir.

```
#teste para custo médio:
(mean.t.teste <- t.test(data.mean.test,
                 mu=current_mu,                 #hipotese nula
                 alternative = "less",  #hipotese 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
```

O teste da média resultou em:

```
mean.t.teste
```

```
##
##  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
```

3.1.4 Calculo do intervalo de confianca

Para o cálculo do intervalo de confiança utilizou-se a função t.test para o teste da hipotese nula, conforme mostrado a seguir.

```
mean.intervalo.ci <- t.test(data.mean.test,
                    mu = current_mu,
                    conf.level=ci_mean)$conf.int
```

A função resultou no intervalo:

```
mean.intervalo.ci
```

```
## [1] 47.25343 50.81884
## attr(,"conf.level")
## [1] 0.99
```

Como o p-valor é maior do que o nível de significância estabelecido ($\alpha = 0.01$), não há evidência suficiente para se rejeitar a hipótese nula.

3.1.5 Validação de premissas

3.2 Teste sobre a variância do custo

3.2.1 Teste de Hipoteses

Para a variância, e dado que a população é modelada por uma distribuição normal (vide análise exploratória), a estatística de teste irá seguir uma distribuição chi-quadrado com $n-1$ graus de liberdade [1]. O teste da variância pode ser executado com a função `varTest` do pacote *EnvStats* [3].

```
sig_level_sd <- 0.05
ci_sd <- 1 - sig_level_sd

varTest(x = data.mean.test, alternative = "less",
        sigma.squared =  current_var, conf.level = ci_sd)
```

```
##
##  Chi-Squared Test on Variance
##
## data:  data.mean.test
## Chi-Squared = 19.362, df = 65, p-value = 6.401e-09
## alternative hypothesis: true variance is less than 100
## 95 percent confidence interval:
##   0.00000 40.80488
```

```
## sample estimates:
## variance
##  29.7873
```

Logo, a hipótese nula é rejeitada, sendo possível declarar que a variância da nova versão do software é significativamente inferior à atual (com um intervalo de confiança de 95%).

3.2.2 Calculo do intervalo de confianca

Como temos uma hipótese unilateral, precisamos definir somente um intervalo superior para o nível de significância de desejadado $\alpha = 0.95$. De acordo com [1], para $n-1$ graus de liberdade e significância α, ele pode ser calculado por:

$$\sigma^2_{Upper} = \frac{(n-1)S^2}{\chi^2{}_{(\alpha,n-1)}} \tag{1}$$

```
sig_level_sd <- 0.05
ci_sd <- 1 - sig_level_sd
var.upper.ci <- (N-1)*var(data.mean.test)/qchisq(sig_level_sd, N-1)
print(var.upper.ci)
```

```
## [1] 40.80488
```

Logo, o intervalo de confiança é [0, 40.26176], para um nível de confiança de 95%. ### Validaçao das premissas

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] S. P. Millard, *EnvStats: An r package for environmental statistics.* New York: Springer, 2013.