Estudo de caso: Grupo D 3

Gilmar and Maressa Nunes R. Tavares and Victor

3 de Setembro, 2019

0.1 Descrição do Problema

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 uma análise estatística deve ser feita para investigar os ganhos de desempenho obtidos em relação à versão atual. Espera-se que sejam testados a média e variância dos custos de execução O presente trabalho tem como objetivo delinear e executar 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 do custo computacional com média $\mu = 50 evariância\ \sigma = 100$, dados da população, objetiva-se verificar se a nova versão apresenta resultados melhores para tais características. Para tanto, utilizou-se o teste z 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. . . ..

0.2 Planejamento do Experimento

0.2.1 Geraçao dos dados

Para simular a geração de dados da nova versão, a biblioteca *ExpDE* [1] será utilizada. Ela é declarada da seguinte forma:

```r
# 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(1234) # 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){
  my.sample <- numeric(N)
  for (i in seq(N)){
    my.sample[i] <- get.single.sample(mpo, mmu, mre, mse, mst, mpr)
  }
  return(my.sample)
}
```

As funções `get.single.sample` e `get.n.samples` foram criadas para facilitar o entendimento da função de geração de dados.

0.2.2 Teste do custo médio

Para este teste, são estabelecidos 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 $\pi = 1 - \beta = 0.8$

Como estamos interessados em saber se existem ganhos em termos do custo médio, e dado que a média populacional da versão atual é $\mu_0 = 50$, define-se a seguinte hipótese nula e alternativa:

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

0.2.3 Teste da variância do custo

Para este teste, são estabelecidos os seguintes objetivos:

- Nível de significância desejado $alpha = 0.01$. Logo, o nível de confiança desejado é $1 - \alpha = 0.99$
- Usar as mesmas observações coletadas para o teste da média.

Como estamos interessados em saber se existem ganhos em termos de variância média, e dado que a variância populacional da versão atual é $\sigma^2 = 100$, define-se a seguinte hipótese nula e alternativa:

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

0.3 Análise Exploratória dos Dados

0.4 Análise Estatística

0.4.1 Teste sobre a média do custo

0.4.1.1 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`:

```
# 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"))
```

```r
# round to the next integer
N <- ceiling(sample_size_calc$n)
```

Resultando em um tamanho amostral de:

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

0.4.1.2 Teste de Hipoteses

0.4.1.3 Calculo do intervalo de confianca

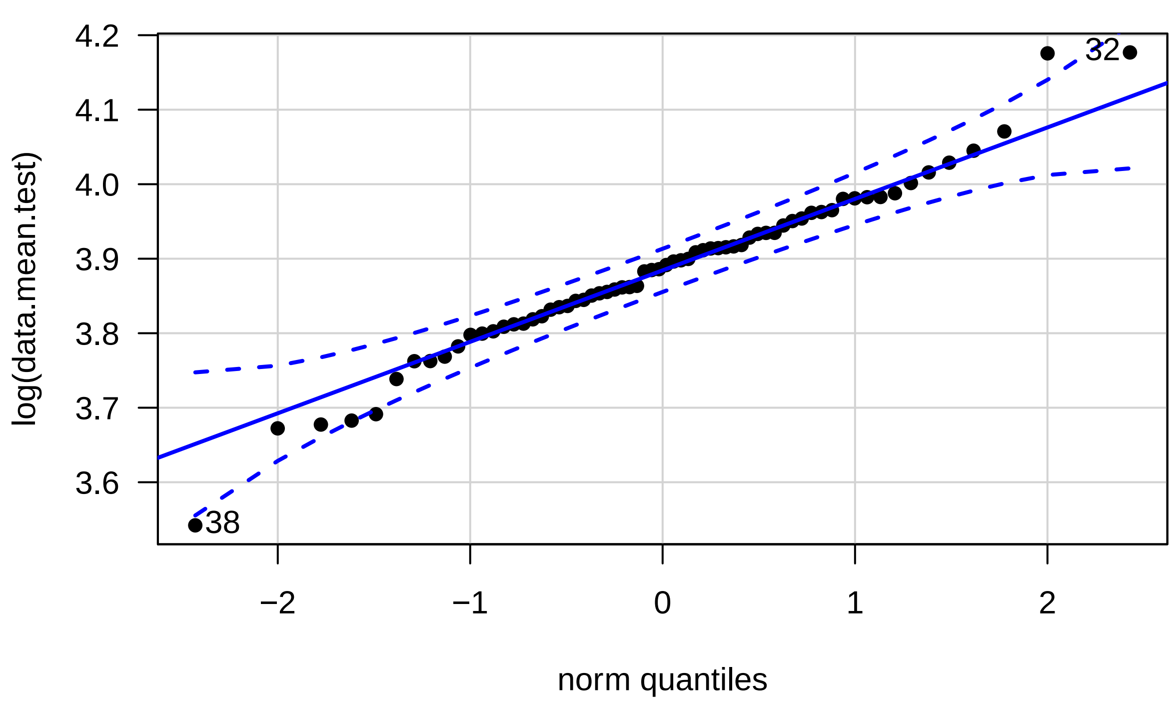
0.4.1.4 Validaçao das premissas

0.4.2 Teste sobre a variância do custo

0.4.2.1 Teste de Hipoteses

Para a variância, e assumindo que a população é modelada por uma distribuição normal, a estatística de teste irá seguir uma distribuição chi-quadrado com $N-1$ graus de liberdade. Entretanto, conforme visto na análise exploratória, a nova versão do sistema não segue uma distribuição normal. Portanto, uma alternativa é aplicar alguma transformação aos dados que os leve à normalidade. Uma possibilidade é o uso da transformação logarítimica:

```r
data.mean.test <- get.n.samples(mpo, mmu, mre, mse, mst, mpr, N)
# Plot a Normal qq-plot (ggplot2)
qqPlot(log(data.mean.test), las = 1, pch = 16)
```




```
## [1] 38 32
```

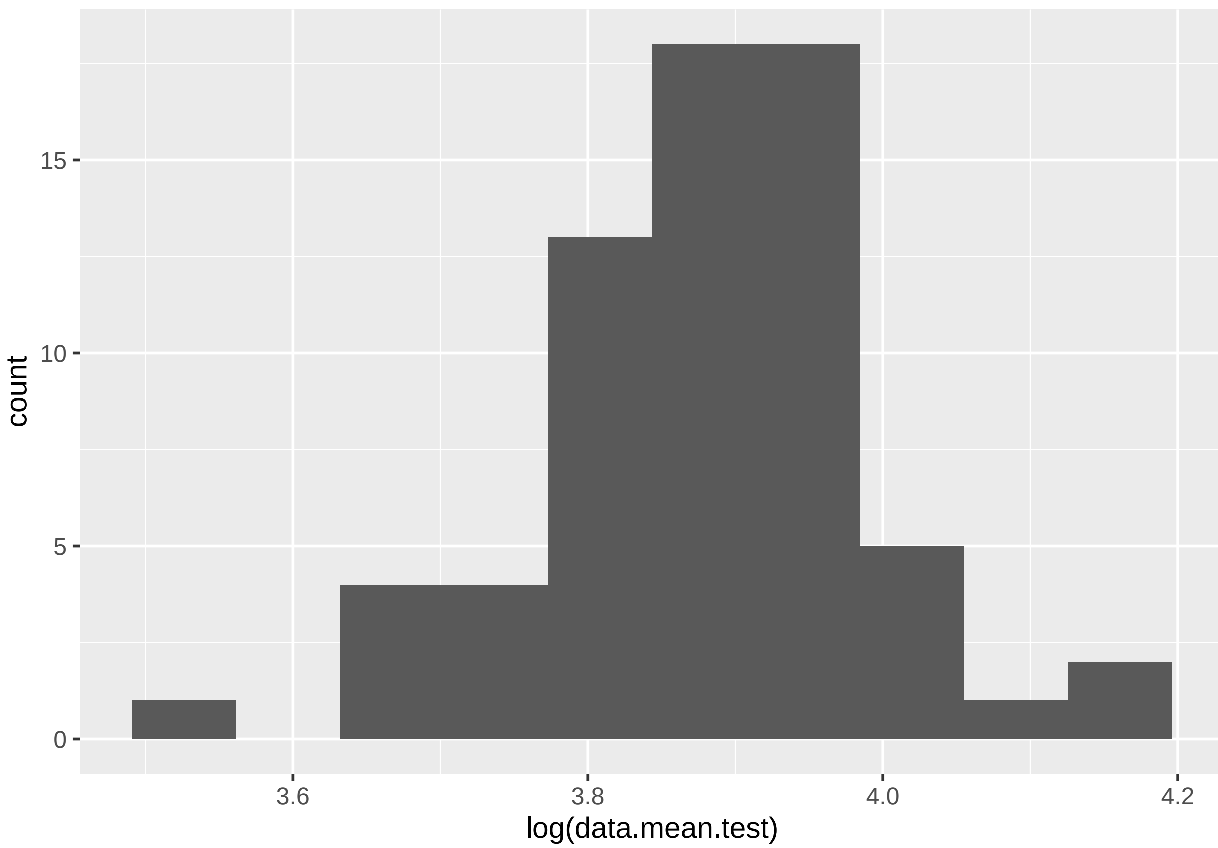
```r
# Perform shapiro-test on the transformed data, to validate the normal distribution hypothesis
shapiro.test(data.mean.test)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data.mean.test
## W = 0.96988, p-value = 0.1084
```

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

```
##
##  Shapiro-Wilk normality test
##
## data:  log(data.mean.test)
## W = 0.97713, p-value = 0.2617
```

```r
p2 <- ggplot(as.data.frame(log(data.mean.test)), aes(x = log(data.mean.test)))
p2 + geom_histogram(bins = 10)
```



Logo, como o p-valor do teste Shapiro é maior que 0.1 para a transformação logarítimca, podemos afirmar que a população pode ser modelada por uma distribuição normal. Portanto, podemos aplicar o teste usual descrito anteriormente:

```r
var.chi.squared <- (N-1)*var(data.mean.test)/current_var
print(var.chi.squared)
```

```
## [1] 19.10404
```

A hipótese nula é rejeitada se o `var.chi.squared` está na região crítica.

0.4.2.2 Calculo do intervalo de confianca

0.4.2.3 Validaçao das premissas

0.4.3 Discussão e Conclusões

0.5 Divisão das Atividades

Victor - Reporter Maressa - Coordenadora Gilmar - Verificador e Monitor

Referências

[1] 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.