Estudo de Caso 01: Desempenho de uma nova versão de Software

André Boechat(Checker), Mateus Pongelupe(Coordinator), Samuel Leite(Recorder)

24 de Setembro de 2018

Resumo

Este trabalho consiste no primeiro estudo de caso da disciplina de Planejamento e Análise de Experimentos. Nele, foram executados testes para a avaliar se o desempenho de uma nova versão de software é superior ao da anterior. A média e a variância do custo de execução foram as variáveis escolhidas para fazer essa medida, sendo que a versão atual do software possui um custo de execução dado por uma distribuição conhecida. Os resultados alcançados não suportam a afirmativa que o desempenho da nova versão é superior em termos da média do custo de execução, mas suportam a hipótese de que a variância do custo de execução é menor para a nova versão.

Planejamento do Experimento

Nesse experimento, está sendo avaliado o desempenho de uma nova versão de software. É conhecido que a versão atual do software possui uma distribuição para seu custo de execução com média populacional $\mu = 50$ e variância populacional $\sigma^2 = 100$. Para a nova versão do software deseja-se investigar seus resultados quanto a melhorias de desempenho, isto é, menor custo médio de execução e/ou menor variância. Com esse intuito, foram desenvolvidos dois experimentos: um para avaliar a média e outro para avaliar a variância.

Teste da média

Para avaliar se o desempenho do novo software é melhor que a versão antiga, está sendo observado se a média do custo de execução é menor. Assim, ao definir a hipótese H_1, podemos fazer com que ela seja unidirecional, isto é, a região de interesse do teste está na direção em que a média de execução da nova versão seja menor que a média atual. Dessa forma, a hipótese nula H_0 e a hipótese H_1 podem ser definidas como:

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

Para esse teste, definiu-se um nível de significância de $\alpha = 0.01$, um efeito de relevância mínimo $\delta^* = 4$ e uma potência desejada de $\pi = 1 - \beta = 0.8$.

```r
h0.mean = 50
h0.sd = sqrt(100)

t1.alpha = 0.01
t1.delta = 4
t1.beta = 0.2
t1.power = 1 - t1.beta
```

Assumindo que a hipótese nula H_0 se comporte com uma distribuição de média populacional $\mu = 50$ e variância populacional $\sigma^2 = 100$, pode-se calcular o número de amostras mínimo a partir do teste Z, haja vista que a variância da hipótese nula é conhecida. Para fazer esse cálculo, foi usado o pacote *asbio*:

```r
library(asbio)
n <- power.z.test(power = t1.power, alpha = t1.alpha, effect = t1.delta,
                  sigma = h0.sd, test = "one.tail")$n
t1.N <- ceiling(n)
cat("N: ", t1.N)
```

```
## N:  63
```

Assim, fazendo uso do teste Z, precisaremos de uma amostra de tamanho $N = 63$ para executar o nosso teste com uma potência $\pi = 0.8$. Por nossa hipótese H_1 ser unidirecional, a região crítica do teste Z pode ser determinada como:

$$P(z_\alpha \leq Z_0 \mid H_0 \text{ seja verdadeira})$$

Isto é, para que a hipótese nula seja rejeitada com um nível de confiança de 99% é preciso que $z_\alpha > Z_0$.

Teste da variância

O teste da variância permite com que seja avaliado como está se comportando a nova versão do software em relação à variabilidade dos custos. Definem-se, portanto, duas hipóteses. A primeira, na qual a variância do novo processo é mantida constante e a segunda, na qual há.

$$\begin{cases} H_0 : \mu = 100 \\ H_1 : \mu < 100 \end{cases}$$

Para esse teste, definiu-se um nível de significância de $\alpha = 0.05$.

O teste utilizado para verificar essas hipóteses foi o χ^2.

De acordo com a Teórica, para fazer esse teste é necessário calcular o valor de χ^2 e se comparar de acordo com um valor tabelado.

$$\chi^2 = S^2(N-1)/\sigma^2$$

A Região Crítica, nessa distribuição, para o caso onde está se testando uma amostra de valor menor que a hipótese nula, é:

$$P(\chi^2 < \chi^2_c) = \alpha$$

Rejeita-se a hipótese nula se o valor da estatística pertencer à região crítica.

Dentro do pacote *EnvStats*, há a função *varTest*, que faz o teste da variância de acordo com três métodos: bilateral, menor ou maior (two-sided, less e greater).

Coleta de Dados

A coleta de dados foi simulada a partir da rotina sugerida no caso de uso, com uma pequena modificação: uma *seed* foi definida para a execução do programa, de forma a garantir sua reprodutibilidade.

```r
# Loading required package
library(ExpDE)
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))

# Setting seed so the program can be reproduced.
```

```
set.seed(1998)

# One sample
ExpDE(mpo, mmu, mre, mse, mst, mpr,
               showpars = list(show.iters = "none"))$Fbest
```

Em nossos experimentos, precisaremos coletar um número arbitrário **N** de amostras. Portanto, a partir das rotinas acima, foram criadas duas funções para essa coleta:

- *generate_sample* : Coleta uma única amostra.
- *generate_n_samples* : Coleta **n** amostras no formato de um *data.frame*.

Segue abaixo a codificação dessas funções, bem como um exemplo da chamada de *generate_n_samples* para $N = 10$:

```
#Generates one sample
generate_sample <- function() {
  return(ExpDE(mpo, mmu, mre, mse, mst, mpr,
               showpars = list(show.iters = "none"))$Fbest);
}

#Generates n samples on the data.frame format
generate_n_samples <- function(n) {
  cost <- replicate(n, generate_sample())
  return(data.frame(cost))
}

#Example for N=10
generate_n_samples(10)
```

Análise Estatística

Teste da Média

Dados os parâmetros definidos na seção *Planejamento do Experimento* para o teste da média, foram recolhidas $N = 63$ amostras e o teste foi executado nas linhas abaixo. O intervalo de confiança também foi calculado, considerando uma distribuição normal cuja variância populacional $\sigma^2 = 100$ é conhecida.

```
## Getting the samples
t1.samples <- generate_n_samples(t1.N)
## Writing samples to csv file
write.csv(t1.samples, 'test-one.csv')

## Test Z Execution
t1.mean <- mean(t1.samples$cost)
t1.sd <- sd(t1.samples$cost)
z0 <- (t1.mean - h0.mean)/(h0.sd/sqrt(t1.N))
t1.z_alpha <- qnorm(t1.alpha)

## Confidence interval
t1.error <- qnorm(1-(t1.alpha/2)) * h0.sd / sqrt(n)


cat("\n",
"Mean: ", t1.mean, "\n",
```

```
"Z0: ", z0 ,"\n",
"Zalpha: ",  t1.z_alpha ,"\n",
"Confidence Interval: ", t1.mean - t1.error, " <= ",
                         t1.mean, " <= ", t1.mean + t1.error, "\n")
```

```
##
##  Mean:  50.78709
##  Z0:  0.6247342
##  Zalpha:  -2.326348
##  Confidence Interval:  47.53475  <=  50.78709  <=  54.03943
```

Como $Z_\alpha < Z_0$, conclui-se que não há evidências suficientes para rejeitar H_0 a um nível de confiança de 99%.

Teste da Variância

Com os dados coletados e armazenados na variável *t1.samples*, é possível verificar se o novo software irá gerar dados com uma variância menor ou maior que aquela resultada no processo original.

O teste foi executado conforme explicado na seção *Teste da Variância*.

```
library(EnvStats)
varTest(unlist(t1.samples), alternative = "less", conf.level = 0.95, sigma.squared =100)
```

```
##
## Results of Hypothesis Test
## --------------------------
##
## Null Hypothesis:                 variance = 100
##
## Alternative Hypothesis:          True variance is less than 100
##
## Test Name:                       Chi-Squared Test on Variance
##
## Estimated Parameter(s):          variance = 62.01004
##
## Data:                            unlist(t1.samples)
##
## Test Statistic:                  Chi-Squared = 38.44622
##
## Test Statistic Parameter:        df = 62
##
## P-value:                         0.00814041
##
## 95% Confidence Interval:         LCL =  0.00000
##                                  UCL = 85.64727
```

```
## Foi utilizada a função unlist(t1.samples) para garantir que a variável t1.samples
## estivesse no data type correto.
```

Como o valor de P calculado é menor do que o valor de α, é possível afirmar que a hipótese nula está negada e a variância do novo teste é, portanto, inferior à variância do processo original.

Avaliando suposições do modelo

A validação das suposições de um experimento é um passo importante de uma análise de experimento. Não apenas permite verificá-las e como também identificar possíveis efeitos nos resultados encontrados, decorrentes de violações das premissas do planejamento experimental.

Ao fazer o teste da média, foi suposta uma distribuição normal das amostras. Para avaliar essa suposição, o teste de Shapiro-Wilk é uma boa alternativa. Trata-se de um teste de normalidade que assume uma hipótese nula de que a distribuição de um conjunto de dados é normal. O resultado do teste fornece um valor p que, se menor que o nível α desejado, permite rejeitar a hipótese nula. Para o método padrão disponível no R, o valor de $p < 0,05$ indica que não á uma distribuição normal.

Outro indicador interessante é o *qqplot* que é um gráfico em que se compara os quantis da distribuição das amostras aos quantis de uma distribuição normal. Ele fornece um bom indicativo do comportamento da distribuição das amostras em relação a uma normal, permitindo avaliar o quão próximo é de uma normal. Ambos indicadores foram calculados para as amostras colhidas, bem como um histograma e um gráfico de densidade.

```r
library(car)
car::qqPlot(t1.samples$cost,
       pch=16,
       cex=1.5,
       las=1,
       ylab = 'cost')
```

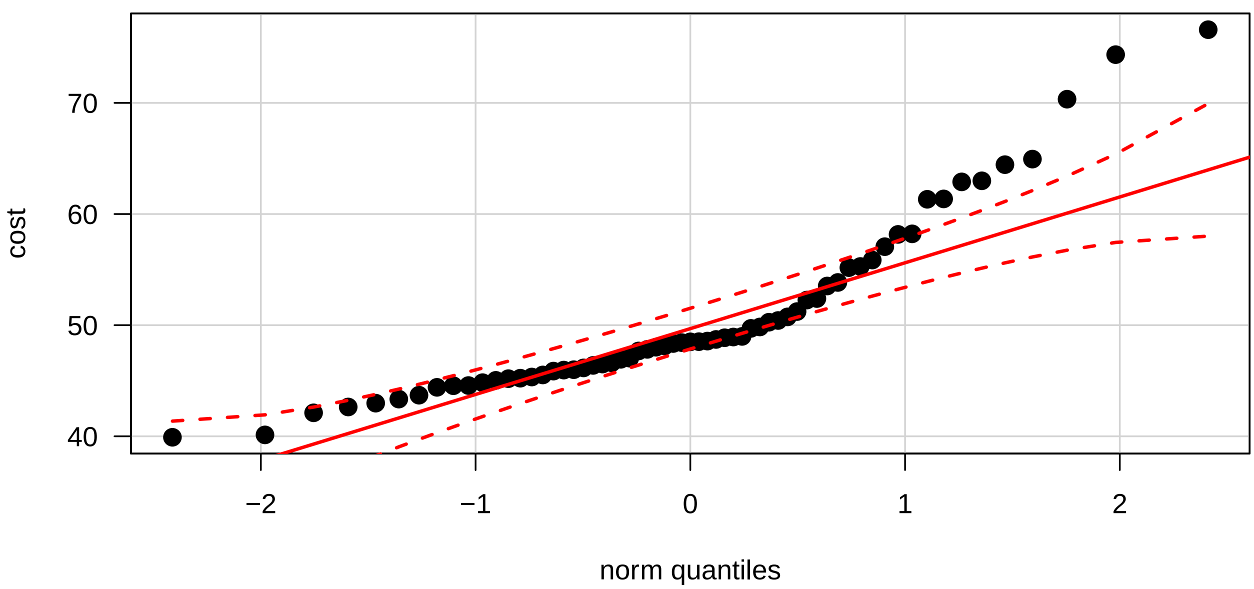

Figura 1: Comparação dos quantis da distribuição das amostras com os quantis de uma distribuição normal

```r
shapiro.test(t1.samples$cost)
```

```
##
## Results of Hypothesis Test
## --------------------------
##
## Alternative Hypothesis:
##
## Test Name:                       Shapiro-Wilk normality test
```

```
##
## Data:                            t1.samples$cost
##
## Test Statistic:                  W = 0.8611542
##
## P-value:                         4.223349e-06
```

```r
library(cowplot,warn.conflicts = FALSE)

theme_set(theme_cowplot(font_size=12))

plot.hist <- ggplot(t1.samples, aes(x=cost)) +
    geom_histogram(colour="black", fill="white") + background_grid(major = 'xy')

plot.dens <- ggplot(t1.samples, aes(x=cost)) +
    geom_density(alpha=.2, fill="#FF6666") +
    background_grid(major = 'xy')

plot_grid(plot.hist, plot.dens, labels = c('A','B'), ncol = 2)
```

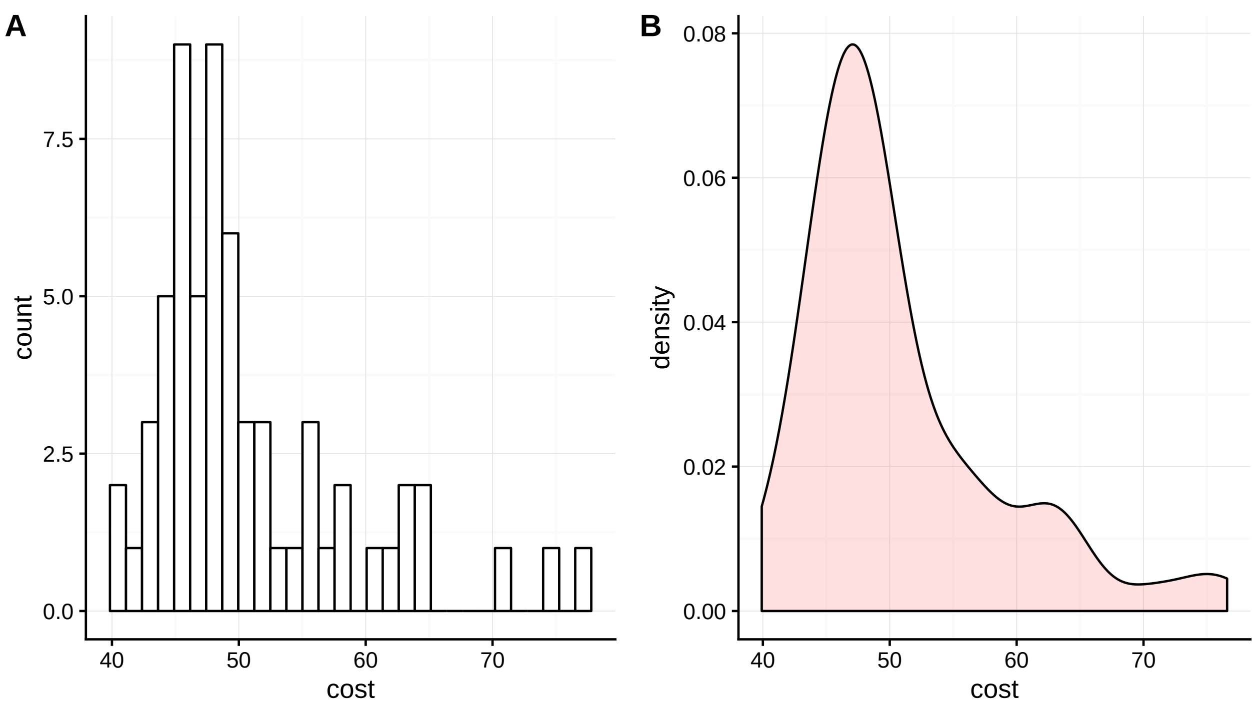

Figura 2: Histograma e gráfico de densidade das amostras colhidas para o teste da média.

Observando os resultados do teste de Saphiro-Wilk, verifica-se que $p = 4,22 \times 10^{-6} < 0,05$, isto é, o teste indica que a distribuição das amostras não segue uma distribuição normal. Isso também é observável no *qqplot*, em que é perceptível que os quantis da distribuição das amostras não estão próximos dos quantis normais em todo o intervalo. Contudo, uma boa parte dos quantis está em uma região quase normal, sendo que passa a fugir de um comportamento de uma normal quando o custo supera 60.

Os gráficos da figura seguinte, o histograma e o gráfico de densidade ajudam a ressaltar isso. No gráfico de densidade, percebe-se um comportamento próximo de uma normal até o custo atingir 60. A partir desse valor, a função de densidade apresenta dois picos que prejudicam bastante a premissa de normalidade.

Conclusões e Recomendações

O estudo conduzido nesse trabalho mirou avaliar o desempenho de uma nova versão de um software em comparação a sua versão anterior, cujo custo de execução é bem representado por uma distribuição populacional de média $\mu = 50$ e variância $\sigma^2 = 100$. Para tal, foram empregados métodos estatísticos provenientes das aulas da disciplina de Planejamento e Análise de Experimentos em ensaios acerca da média e da variância do custo de execução do novo software. No teste da média foi empregado o teste Z, haja vista a premissa que o comportamento da função era normal com variância semelhante ao da função comparada. Foi empregado o Teste de Hipóteses com base em uma distribuição χ^2 para analisar a alteração da variância no novo software.

Com relação ao teste da média, ele falhou em refutar a hipótese nula, isto é, ele foi incapaz de afirmar ao nível de significância de 99% que a nova versão do software possui um custo médio de execução mais baixo. Portanto, o teste executado não suporta a hipótese de que a nova versão do software possui um desempenho superior à versão anterior em termos da média do custo de execução. Apesar disso, a partir dos dados colhidos, foi estimado o intervalo de confiança para a média μ_1, com um grau de confiança de 99%: $\mu_1 \in [47.53, 54.04]$. Quanto a esse intervalo, observa-se que o seu centro é um pouco superior a 50 e que os limites do intervalo batem, aproximadamente, com o efeito de relevância mínima que este teste buscou detectar.

Posteriormente ao teste, em um momento de análise das premissas, verificou-se que a distribuição das amostras não era normal. A violação dessa premissa explica um pouco como o resultado do experimento pode ter sido distorcido, talvez pelo uso de procedimentos não adequados para o caso. Entre esses procedimentos, pode-se citar o teste Z, em que foi considerada a variância populacional da versão anterior, e o procedimento de seleção da amostra/cálculo da média.

Tendo ciência disso, uma alternativa seria a execução do teste T, que considera a variância da amostra retirada. Outra alternativa seria um tratamento/descarte de amostras espúrias ou que estão nas "pontas" da distribuição das amostras, de forma a atenuar o efeito que essas observações têm na distribuição das amostras.

O Teste de Variância foi realizado seguindo-se a premissa de que a distribuição segue uma tendência χ^2. Apesar de o teste afirmar que a variância da amostra pode ser considerada como inferior à variância original $\sigma^2 = 100$, a premissa de que a distribuição amostral pode ser considerada como normal foi refutada. Portanto, a modelagem dessa distribuição em um padrão qui-quadrático fica também contestada. Ainda, os intervalos de confiança calculados estão bem distante do valor da variância da amostra e esse valor é bem inferior ao valor original.

Referências

- R Man Pages - asbio package - https://rdrr.io/cran/asbio/man/power.z.test.html
- R Man Pages - car package - https://rdrr.io/cran/car/man/qqPlot.html
- Statistics R Tutorial - https://www.cyclismo.org/tutorial/R/confidence.html
- Montgomery, Douglas C. - Applied statistics and probabiliy for engineers (3ª Edição) - Capítulos 8,9
- Notas de Aula - https://github.com/fcampelo/Design-and-Analysis-of-Experiments
- Notas - https://edisciplinas.usp.br/pluginfile.php/2063723/mod_resource/content/0/Aula11-2016.pdf