

Estudo de caso: Grupo D 3

Gilmar and Maressa Nunes R. Tavares and Victor

3 de Setembro, 2019

1 Summary

O presente trabalho tem realizado 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 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....

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 ???.

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 $\mu = 50$, considerou-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 $\alpha = 0.01$; nível de confiança $1 - \alpha = 0.99$; o tamanho mínimo de efeito $\delta^* = 4$; e a 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: $\alpha = 0.05$ e $1 - \alpha = 0.95$

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* ????. 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(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.its = "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, sendo elas para coletar uma única amostra ou n amostras, respectivamente.

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.

3 Análise Exploratória dos Dados

Antes de proceder com as análises estatísticas 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.

4 Resultados

4.1 Validação das premissas

4.2 Teste do custo médio

4.2.1 Teste da variância do custo

5 Análise Estatística

5.1 Teste sobre a média do custo

5.1.1 Cálculo do tamanho amostral

5.1.2 Teste de Hipoteses

5.1.3 Calculo do intervalo de confianca

5.1.4 Validacao das premissas

5.2 Teste sobre a variância do custo

5.2.1 Teste de Hipoteses

5.2.2 Calculo do intervalo de confianca

5.2.3 Validacao das premissas

6 Discussão e Conclusões

7 Divisão das Atividades

Victor - Reporter Maressa - Coordenadora Gilmar - Verificador e Monitor

8 Exploratory Data Analysis

The first step is to load and preprocess the data. For instance,

```
data(mtcars)
fc<-c(2,8:11)
for (i in 1:length(fc)){mtcars[,fc[i]]<-as.factor(mtcars[,fc[i]])}
levels(mtcars$am) <- c("Automatic","Manual")
```

To get an initial feel for the relationships between the relevant variables of your experiment it is frequently interesting to perform some preliminary (exploratory) analysis. This is frequently referred to as *getting a feel* of your data, and can suggest procedures (such as outlier investigation or data transformations) to experienced experimenters.

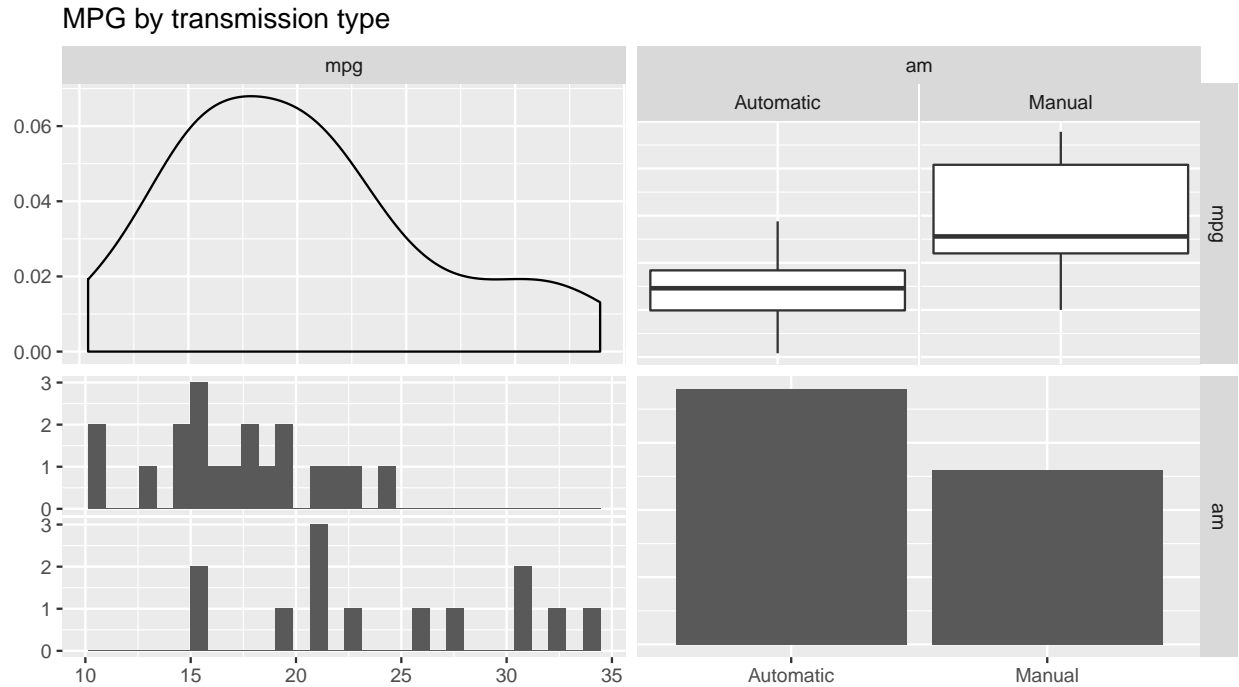


Figure 1: Exploring the effect of car transmission on mpg values

```
library(GGally,quietly = T, warn.conflicts = F) # This is just me getting fancy.
# There are much simpler ways ;-)
ggpairs(data=mtcars,columns=c(1,9),title="MPG by transmission type",
  upper=list(combo="box"),lower=list(combo="facethist"),
  diag=list(continuous="density",discrete="bar"))
```

```
## Warning in check_and_set_ggpairs_defaults("diag", diag, continuous =
## "densityDiag", : Changing diag$continuous from 'density' to 'densityDiag'
```

```
## Warning in check_and_set_ggpairs_defaults("diag", diag, continuous =
## "densityDiag", : Changing diag$discrete from 'bar' to 'barDiag'
```

Your preliminary analysis should be described together with the plots. In this example, two facts are immediately clear from the plots: first, **mpg** tends to correlate well with many of the other variables, most intensely with **drat** (positively) and **wt** (negatively). It is also clear that many of the variables are highly correlated (e.g., **wt** and **disp**). Second, it seems like manual transmission models present larger values of **mpg** than the automatic ones. In the next section a linear model will be fit to the data in order to investigate the significance and magnitude of this possible effect.

9 Statistical Analysis

Your statistical analysis should come here. This is the place where you should fit your statistical model, get the results of your significance test, your effect size estimates and confidence intervals.

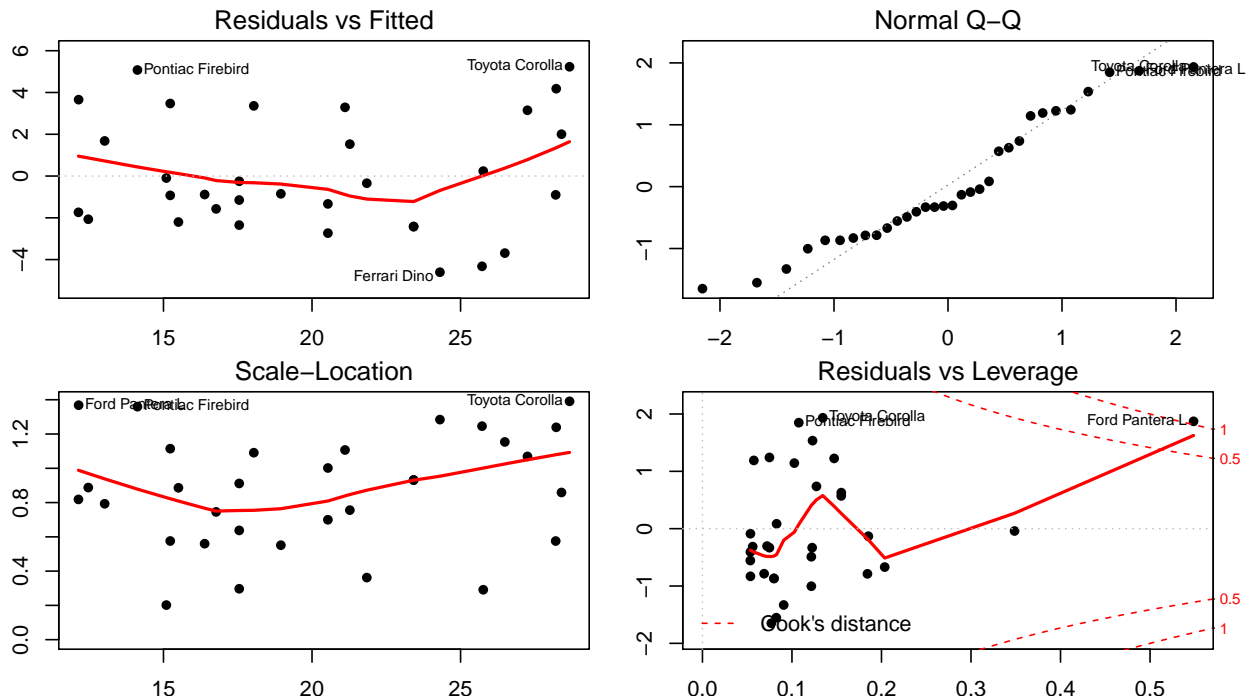


Figure 2: Residual plots for the anova model

```
model<-aov(mpg~am*disp,data=mtcars)
summary(model)
```

```
##           Df Sum Sq Mean Sq F value    Pr(>F)
## am          1  405.2   405.2   47.948 1.58e-07 ***
## disp        1  420.6   420.6   49.778 1.13e-07 ***
## am:disp      1   63.7    63.7    7.537  0.0104 *
## Residuals   28  236.6     8.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

9.1 Checking Model Assumptions

The assumptions of your test should also be validated, and possible effects of violations should also be explored.

```
par(mfrow=c(2,2), mai=.3*c(1,1,1,1))
plot(model,pch=16,lty=1,lwd=2)
```

10 Referências