

ARQUIVO SUPLEMENTAR 1

heRcules: UM REPOSITÓRIO COM SCRIPTS PARA O APRENDIZADO DA ANÁLISE DE DADOS EM R

Hércules Rezende Freitas
Universidade Federal do Rio de Janeiro

Bloco 1. Pacotes utilizados no presente modelo.

```
library(psych)      #REVELLE, 2021
library(dplyr)       #WICKHAM et al., 2021
library(pastecs)     #GROSJEAN & IBANEZ, 2018
library(fBasics)      #WUERTZ et al., 2020
library(skimr)        #WARING et al., 2021
library(ggplot2)      #WICKHAM, 2016
library(xlsx)         #DRAGULESCU & ARENDT, 2020
library(reshape2)      #WICKHAM, 2007
library(rstatix)      #KASSAMBARA, 2021
library(kableExtra)    #ZHU, 2021
library(pwr)          #CHAMPELY, 2020
library(effsize)       #TORCHIANO, 2020
library(stats)         #R CORE TEAM, 2013
```

Bloco 2. Dados utilizados no modelo.

```
Controle <- c(95.2618454, 92.13381555, 97.84615385, 98.8475457, 90.50704942,
87.3190025, 90.77281689, 89.36114184, 95.00712872, 97.33020831, 90.82158097,
89.81937974, 97.75434077, 98.18070701)
```

```
Positivo <- c(0.67114094, 0.101112235, 0.343642612, 1.66768746, 0.622325536,
0.663438837, 1.29611535, 1.755941115, 0.567796205, 0.823841307, 1.050682223,
0.896305131, 1.319383896, 1.893025442)
```

```
Tratamento_I <- c(97.9301423, 88.39160839, 96.36363636, 92.36734536, 88.26833944,
97.31882091, 87.79539853, 95.4437578, 87.45515398, 91.02689562, 92.61273772,
91.05310723, 91.24194591, 92.28904403)
```

```
Tratamento_II <- c(76.4720108, 75.64001264, 76.49453384, 75.33877758, 75.45820278,  
78.72481174, 76.70422698, 76.20735939, 76.08017052, 75.49445089, 76.4321343,  
78.485223, 75.94288458, 76.91323186)  
  
df <- data.frame(Controle, Positivo, Tratamento_I, Tratamento_II)  
  
db <- df #Atribui "df" a "db"  
  
k <- kbl(caption = "Modelo I",  
         x = db) #Cria tabela kbl() e atribui a "k"  
  
kable_classic(full_width = F,  
               html_font = "Cambria",  
               kable_input = k) #Define o tema da tabela
```

Bloco 3. Estimativa de tamanho amostral para análise de variância.

```
avn <- pwr.anova.test(k = 4, #Número de grupos experimentais  
                      n = NULL, #Número de observações (tamanho amostral)  
                      f = 0.5, #Tamanho de efeito f  
                      sig.level = 0.05, #alfa (prob. de falso positivo)  
                      power = .95) #1-beta (1 - prob. de falso negativo)  
  
avn #Resultado da estimativa  
  
plot.power.htest(avn) #Resultado gráfico da estimativa
```

Bloco 4. Estimativa de poder estatístico para análise de variância.

```
avp <- pwr.anova.test(k = 4, #Número de grupos experimentais  
                      n = 15, #Número de observações (tamanho amostral)  
                      f = 0.5, #Tamanho de efeito f  
                      sig.level = 0.05, #alfa (prob. de falso positivo)  
                      power = NULL) #1-beta (1 - prob. de falso negativo)  
  
avp #Resultado da estimativa  
  
plot.power.htest(avp) #Resultado gráfico da estimativa
```

Bloco 5. Estimativa de tamanho amostral para testes t.

```
tt <- pwr.t.test(n = NULL,          #Número de observações (tamanho amostral)
                  d = 1.0,          #Tamanho de efeito d
                  sig.level = 0.05, #alfa (prob. de falso positivo)
                  power = .95,      #1-beta (1 - prob. de falso negativo)
                  type = "two.sample", #Duas amostras, não pareadas
                  alternative = "two.sided") #Duas caudas

tt                                #Resultado da estimativa

plot.power.htest(tt)             #Resultado gráfico da estimativa
```

Bloco 6. Estimativa de tamanho de efeito (Cohen D).

```
cohen.d(df$Controle, df$Tratamento_II) #Resultado
```

Bloco 7. Estatística descritiva com o comando “summary()”.

```
summary(df)                      #Resultado
```

Bloco 8. Estatística descritiva com o comando “stat.desc()”.

```
stat.desc(df)                      #Resultado
```

Bloco 9. Estatística descritiva com o comando “skim()” ou “describe()”.

```
skim(df)                          #Resultado
```

#OU

```
describe(df)                      #Resultado
```

Bloco 10. Diagramas de caixa (boxplots) das variáveis do modelo.

```
boxplot(df$Controle,
        df$Positivo,
        df$Tratamento_I,
        df$Tratamento_II,          #Determina as variáveis de df para o boxplot
        col = "lightblue",          #Determina a cor de preenchimento
        names = c("Controle",
                 "Positivo",
```

```
"Tratamento_I",  
"Tratamento_II")) #Determina os nomes de cada grupo
```

Bloco 11. Matriz de gráficos com sobreposição de elementos.

```
par(mfrow = c(2, 2)) #Duas linhas, duas colunas  
  
hist(df$Controle,  
      freq = F,  
      xlim = c(0,100),  
      col = "white",  
      main = "Controle",  
      xlab = "Intensidade")  
lines(density((df$Controle)),  
      col = "green",  
      lwd = 2)  
lines(x = c(mean(df$Controle), mean(df$Controle)),  
      y = c(mean(df$Controle), 0),  
      col = "green",  
      lty = 2,  
      lwd = 2)  
lines(x = c(min(df$Controle), min(df$Controle)),  
      y = c(min(df$Controle), 0),  
      col = "green",  
      lty = 2,  
      lwd = 2)  
  
hist(df$Tratamento_I,  
      freq = F,  
      xlim = c(0,100),  
      col = "white",  
      main = "Tratamento_I",  
      xlab = "Intensidade")  
lines(density((df$Tratamento_I)),  
      col = "blue",  
      lwd = 2)  
lines(x = c(mean(df$Tratamento_I), mean(df$Tratamento_I)),  
      y = c(mean(df$Tratamento_I), 0),  
      col = "blue",  
      lty = 2,  
      lwd = 2)  
lines(x = c(min(df$Tratamento_I), min(df$Tratamento_I)),  
      y = c(min(df$Tratamento_I), 0),  
      col = "blue",
```

#Histograma do grupo controle
#Sem contagem de frequências
#Limites do eixo x
#Cor desejada para o gráfico
#Título
#Legenda do eixo x
#Gráfico de densidade
#Cor desejada para o gráfico
#Espessura da linha
#xy da Linha
#xy da Linha
#Cor desejada para o gráfico
#Tipo de linha (pontilhada)
#Espessura da linha
#xy da Linha
#xy da Linha
#Cor desejada para o gráfico
#Tipo de linha (pontilhada)
#Espessura da linha

#Histograma do grupo controle

Hércules Rezende Freitas

```
lty = 2,  
lwd = 2)  
  
hist(df$Tratamento_II, #Histograma do grupo controle  
      freq = F,  
      xlim = c(0,100),  
      col = "white",  
      main = "Tratamento_II",  
      xlab = "Intensidade")  
lines(density((df$Tratamento_II)),  
      col = "orange",  
      lwd = 2)  
lines(x = c(mean(df$Tratamento_II), mean(df$Tratamento_II)),  
      y = c(mean(df$Tratamento_II), 0),  
      col = "orange",  
      lty = 2,  
      lwd = 2)  
lines(x = c(min(df$Tratamento_II), min(df$Tratamento_II)),  
      y = c(min(df$Tratamento_II), 0),  
      col = "orange",  
      lty = 2,  
      lwd = 2)  
  
hist(df$Positivo, #Histograma do grupo controle  
      freq = F,  
      xlim = c(0,100),  
      col = "white",  
      main = "Positivo",  
      xlab = "Intensidade")  
lines(density((df$Positivo)),  
      col = "red",  
      lwd = 2)  
lines(x = c(mean(df$Positivo), mean(df$Positivo)),  
      y = c(mean(df$Positivo), 0),  
      col = "red",  
      lty = 2,  
      lwd = 2)  
lines(x = c(min(df$Positivo), min(df$Positivo)),  
      y = c(min(df$Positivo), 0),  
      col = "red",  
      lty = 2,  
      lwd = 2)  
  
par(mfrow = c(1, 1)) #Uma linha, uma coluna
```

Bloco 12. Geração de gráfico de barras com ggplot2.

```
dtt <- melt(df, id.vars = 0) #formato wide para o formato long

dttt <- dtt %>%
  group_by(variable) %>%
  summarise(sd = sd(value,
                     na.rm = TRUE),
            len = mean(value)) #Agrupa as variáveis do modelo
                                         #Calcula os desvios padrão
                                         #Elimina valores ausentes
                                         #Calcula as médias

ggplot(dtt, aes(x = variable,
                 y = value)) +
  geom_bar(stat = "summary",
            fun = "mean",
            fill = "lightblue") +
  geom_jitter(position = position_jitter(0.2),
              color = "black",
              size = 1,
              shape = 21) + #Determina o tipo (barras)
                                         #Adiciona pontos individuais
  labs(x = " ",
        y = "Intensidade") + #Nomeia os eixos
  theme(panel.background = element_blank(),
        axis.line.y = element_line("black", linewidth = .25),
        axis.line.x = element_line("black", linewidth = .25)) + #Muda o tema
  geom_errorbar(aes(x = variable,
                     y = len,
                     ymin = len-sd,
                     ymax = len+sd),
                data = dttt,
                width = 0.2) #Adiciona as barras de erro
```

Bloco 13. Identificação de *outliers* e valores extremos em R.

```
dtt %>%
  group_by(variable) %>%
  identify_outliers(value) #Agrupa as variáveis do modelo
                                         #Calcula outliers
```

Bloco 14. Teste de normalidade por Shapiro-Wilk em R.

```
shapiroTest(x = df$Controle) #Resultado
shapiroTest(x = df$Positivo) #Resultado
shapiroTest(x = df$Tratamento_I) #Resultado
shapiroTest(x = df$Tratamento_II) #Resultado
```

Bloco 15. Criação e descrição do modelo ANOVA em R.

```
res.aov <- aov(dtt$value ~ dtt$variable)      #Cria o modelo ANOVA
summary(res.aov)                            #Resultado
```

Bloco 16. Teste *post-hoc* de Tukey em R.

```
TukeyHSD(res.aov,
           conf.level = .95)                      #Testa e apresenta os resultados
```

Bloco 17. Múltiplos testes *t* em R.

```
t.test(df$Controle, df$Positivo)          #Testa e apresenta os resultados
t.test(df$Tratamento_I, df$Positivo)        #Testa e apresenta os resultados
t.test(df$Tratamento_II, df$Positivo)        #Testa e apresenta os resultados
t.test(df$Controle, df$Tratamento_I)         #Testa e apresenta os resultados
t.test(df$Controle, df$Tratamento_II)         #Testa e apresenta os resultados
```