

Script Aplicando o teste $F(H_0)$ de Graybill e graficos de analise

*Sollano Rabelo Braga
Marcio Leles R. de Oliveira*

2015

Contents

1) Baixar e carregar os pacotes necessários, e importar dados	2
2) Cálculo do FH_0 , F-Crítico e p-valor	2
3) Teste de Hipótese	3
4) Tabela de Resultados Simples	3
5) Tabela de Resultados com medidas de tendência	4
6) Tabela de Resultados completa	5
7) Gráfico de dispersao	5
8) Exportar os resultados	6

```
#
```

1) Baixar e carregar os pacotes necessários, e importar dados

faz-se o download das bibliotecas necessárias, caso ainda não estejam instaladas na máquina. Este passo deve ser feito apenas uma vez. Obs: Remover “#” dos pacotes que não estiverem instalados para instalar o pacote readxl, para poder importar dados providos do excel, sem a necessidade de conversão.

```
# install.packages('readxl', dependencies = T)
```

instala-se o pacote ggplot2, para a criação dos gráficos.

```
# install.packages('ggplot2', dependencies = T)
# install.packages('GridExtra') install.packages('grid')
```

Carregar os Pacotes

```
library(readxl)
library(ggplot2)
```

Para carregar o script funcione corretamente, a padronização de nomes é necessária. Foi utilizado Y1 para nomear a variável padrão, e Yj para a variável proposta. Portanto estes nomes devem ser utilizados na planilha que contém os dados, ou alterados após a importação utilizando a função names().

Para importar os dados utiliza-se a função read_excel:

```
dados <- read_excel("graybill_dados.xlsx")
```

2) Cálculo do FH0, F-Crítico e p-valor

A estatística do teste é a seguinte:

$$F(H_0) = \frac{(\hat{\beta} - \theta)'(y_1' y_1)(\hat{\beta} - \theta)}{2QMRes} \sim F_{\alpha}(2, n - 2 \text{ g.l.})$$

Para calculá-la, primeiro ajusta-se um modelo Linear Simples composto do valor proposto como y, e do valor padrão como x:

```
fit <- lm(Yj ~ Y1, dados)
```

Cálcula-se o Quadrado médio do resíduo, obtido pela razão entre a soma de quadrados de resíduos e os graus de liberdade do resíduo da regressão:

```
QMRes <- sum(residuals(fit)^2)/fit$df.residual
```

A seguir a criação da matriz 2x1 beta - theta, subtraindo os coeficientes da regressão pela matriz theta [0 1]:

```
beta_theta <- coef(fit) - c(0, 1)
```

Cálculo da Matriz Y1'Y1:

esta matriz é composta por n, Somatório de Y1, Somatório de Y1, e o somatório quadrático de Y1.

Utiliza-se a função length para calcular n, e a função sum para calcular os somatórios concatenando-se n e Somatório de Y1 em um vetor, e Somatório de Y1 e somatório quadrático de Y1 em outro e une-se os dois com a função cbind

```
Y1linha_Y1 <- cbind(c(length(dados$Y1), sum(dados$Y1)),
  c(sum(dados$Y1), sum(dados$Y1^2)))
```

Cálculo de FH0

Multiplica-se da matriz beta_theta transposta, pela matriz Y1'Y1, e multiplicamos isto pela matriz beta_theta;

Então divide-se tudo isto por 2*Quadrado médio do resíduo.

Arredonda-se este cálculo para 4 casas decimais com a funcao round():

```
FH0 <- round((t(beta_theta) %*% Y1linha_Y1 %*% beta_theta)/(2 *
  QMRes), 4)
```

Definicao do alpha:

```
alpha <- 0.05
```

Cálculo do F Critico utilizando os graus de liberdade do residuo, e arredondado para 4 casas decimais:

```
Ftab <- round(qf(p = alpha, df1 = 2, df2 = fit$df.residual,
  lower.tail = FALSE), 4)
```

Cálculo do p-valor arredondado para 6 casas decimais:

```
pvalor <- signif(pf(FH0, 2, fit$df.residual, lower = F),
  4)
```

3) Teste de Hipótese

O resultado do teste de hipótese é realizado utilizando lógica condicional; cria-se um objeto denominado resultado com valor "*" caso FH0 > 0, e caso contrario, "ns":

```
if (FH0 > Ftab) {
  Resultado <- "*"
} else {
  Resultado <- "ns"
}
```

Lógica condicional semelhante a anterior

```
if (FH0 > Ftab) {
  Conclusao <- "Yj e estatisticamente diferente de Y1, para o alpha estabelecido"
} else {
  Conclusao <- "Yj e estatisticamente igual a Y1, para o alpha estabelecido"
}
```

4) Tabela de Resultados Simples

A seguir é criada uma tabela de resultados simples, reunindo os resultados essenciais do teste:

```
Tab_Res_Simp <- data.frame(F_H0 = FH0, F_crit = Ftab, P_valor = pvalor,
  Alpha = alpha, Teste = Resultado)
Tab_Res_Simp
```

```
##      F_H0 F_crit P_valor Alpha Teste
## 1 0.4511 3.8056 0.6465 0.05      ns
```

5) Tabela de Resultados com medidas de tendência

A seguir é criada uma tabela um pouco mais detalhada, com valores de tendência e composta apenas por valores numéricos:

```
Tab_Res_Med <- data.frame(
  Resultado = rbind( mean(dados$Y1),
                     mean(dados$Yj),
                     var(dados$Y1),
                     var(dados$Yj),
                     sd(dados$Y1),
                     sd(dados$Yj),
                     length(dados$Y1),
                     2,
                     fit$df.residual,
                     Ftab,
                     FH0,
                     alpha,
                     pvalor),

  row.names = c("Media_Y1",
                "Media_Yj",
                "Variancia_Y1",
                "Variancia_Yj",
                "Desvio_Padrazo_Y1",
                "Desvio_Padrazo_Yj",
                "Observacoes", "g.l.1",
                "g.l.2", "F_crit",
                "F_H0",
                "alpha",
                "p-valor") )
```

Tab_Res_Med

```
##      Resultado
## Media_Y1      0.25351281
## Media_Yj      0.25719061
## Variancia_Y1  0.02055523
## Variancia_Yj  0.02207194
## Desvio_Padrazo_Y1 0.14337094
## Desvio_Padrazo_Yj 0.14856628
## Observacoes    15.00000000
## g.l.1          2.00000000
## g.l.2         13.00000000
## F_crit         3.80560000
## F_H0           0.45110000
## alpha          0.05000000
## p-valor        0.64650000
```

6) Tabela de Resultados completa

A seguir é criada uma tabela ainda mais detalhada, com valores de tendência de ambas as variáveis, e resultado do teste de hipótese; São criados dois objetos separados, que são unidos em um dataframe em seguida:

```
aux1 <- c(round(mean(dados$Y1), 2), round(var(dados$Y1),
2), round(sd(dados$Y1), 2), length(dados$Y1), 2, Ftab,
FH0, alpha, pvalor, Resultado, Conclusao)
aux2 <- c(round(mean(dados$Yj), 2), round(var(dados$Yj),
2), round(sd(dados$Yj), 2), length(dados$Yj), fit$df.residual,
" ", " ", " ", " ", " ", " ", " "))

Tab_Res_Comp <- data.frame(Valor_Padrao = aux1, Valor_Proposto = aux2,
row.names = c("Media", "Variancia", "Desvio_Padrao",
"Observacoes", "g.l.", "F_Critico", "F_H0", "Alpha",
"P-valor", "Teste", "Conclusao"))

Tab_Res_Comp
```

##	Valor_Padrao
## Media	0.25
## Variancia	0.02
## Desvio_Padrao	0.14
## Observacoes	15
## g.l.	2
## F_Critico	3.8056
## F_H0	0.4511
## Alpha	0.05
## P-valor	0.6465
## Teste	ns
## Conclusao	Yj e estatisticamente igual a Y1, para o alpha estabelecido
##	Valor_Proposto
## Media	0.26
## Variancia	0.02
## Desvio_Padrao	0.15
## Observacoes	15
## g.l.	13
## F_Critico	
## F_H0	
## Alpha	
## P-valor	
## Teste	
## Conclusao	

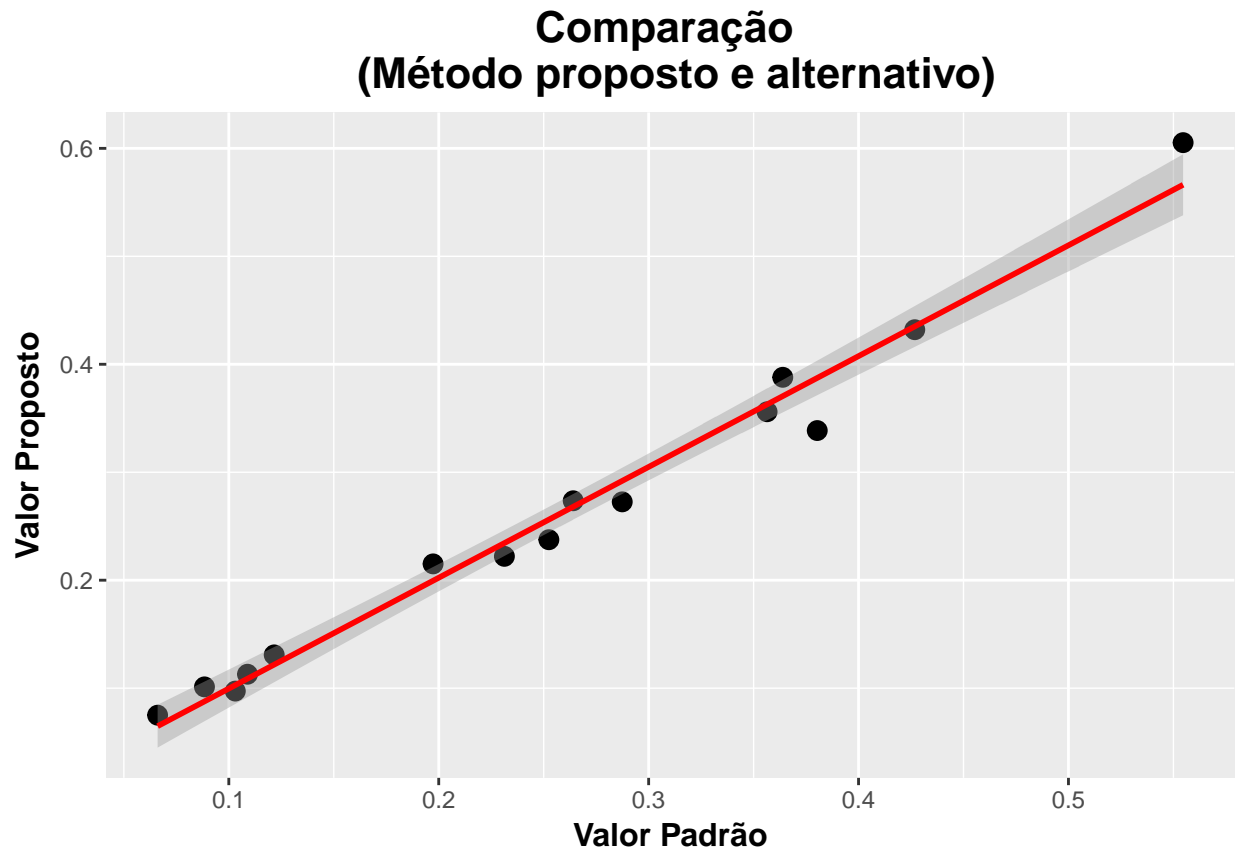
7) Gráfico de dispersao

A seguir Gera-se um gráfico de dispersao Y1 x Yj, para avaliação de tendencia:

```
graph <- ggplot(data = dados, aes(x = Y1, y = Yj)) + geom_point(size = 3) +
labs(x = "Valor Padrão", y = "Valor Proposto", title = "Comparação \n (Método proposto e alternativo)") +
geom_smooth(method = "lm", colour = "red") + theme(axis.title = element_text(size = 12,
face = "bold"), plot.title = element_text(size = 16,
```

```
face = "bold", hjust = 0.5)) #+
# coord_cartesian(xlim = c(0, max(dados$Y1 + 0.3)) , #
# tamanho dos eixos pode ylim = c(0, max(dados$Yj +
# 0.3))) # ser alterado com estes comandos
```

graph



8) Exportar os resultados

Grafico

```
ggsave("graph.png", graph, width = 12, height = 6)
```

Tabela com resultados completa

```
write.csv2(Tab_Res_Comp, "Tab_Res_Comp.csv", row.names = F)
```

Tabela Numerica

```
write.csv2(Tab_Res_Med, "Tab_Res_Med.csv", row.names = F)
```

Tabela simples

```
write.csv2(Tab_Res_Simp, "Tab_Res_Simp.csv", row.names = F)
```