Atividade 2 - Modelos de Regressão

Renan de Oliveira da Cruz

23/04/2022

Abaixo temos o enunciado da atividade

- 1) Gere 5 covariáveis, x1, x2, x3, x4 e x5, n = 50 vezes. Use uma distribuição diferente para cada covariável (por exemplo, N(18, 6), binomial(20, 0,6), Poisson(8) etc);
- 2) Atribua valores para os coeficientes do modelo, considerando $\beta_1 = 3,97$ e $\beta_3 = 4,04$;
- 3) Gere n = 50 erros seguindo $N(0, \sigma^2)$;
- 4) Determine os valores de Y usando o modelo linear nas 5 covariaveis:

$$Y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_5 x_{i5} + erro_i$$

- 5) Usando os dados gerados, ajuste um modelo linear sujeito à restrição $\beta_1 = \beta_3$;
- 6) Usando os dados gerados, ajuste um modelo linear e teste Ho: $\beta_1 = \beta_3$.

Vamos então ao comprimento de cada um dos itens propostos pelo professor.

1) Gere 5 covariáveis, x1, x2, x3, x4 e x5, n = 50 vezes. Use uma distribuição diferente para cada covariável

Vamos gerar dados de tal forma que:

- $x_1 \sim N(20, 25)$;
- $x_2 \sim Gamma(10, 9)$;
- $x_3 \sim Unif(1, 60)$;
- $x_4 \sim Binomial(50, 0.6)$;
- $x_5 \sim Poisson(38)$.

Simulamos esses valores 50 vezes ($n = 50$, tamanho amostral).

```
set.seed(21)

n <- 50
x1 <- rnorm(n, 20, 5)
x2 <- rgamma(n, 10, 9)
x3 <- runif(n, 1, 60)
x4 <- rbinom(n, 50, 0.6)
x5 <- rpois(n, 38)
```

2) Atribua valores para os coeficientes do modelo, considerando $\beta_1 = 3,97$ e $\beta_3 = 4,04$

Agora, vamos a atribuição dos valores dos coeficientes do modelo.

```r
beta0 <- 10
beta1 <- 3.97
beta2 <- -9
beta3 <- 4.04
beta4 <- 1.2
beta5 <- 7
```

3) Gere n = 50 erros seguindo $N(0, \sigma^2)$

Temos para o erro: $\epsilon \sim N(0, 25)$.

```r
sigma2 <- 25
erro <- rnorm(n, 0, sqrt(sigma2))
```

4) Determine os valores de Y usando o modelo linear nas 5 covariaveis:

$$Y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_5 x_{i5} + erro_i$$

Para $i = 1, 2, \cdots, n$.

Por fim, a variável resposta Y é dada por:

```r
y <- beta0 + beta1 * x1 + beta2 * x2 + beta3 * x3 + beta4 * x4 + beta5 * x5 + erro
```

Isto posto, temos: $Y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_5 x_{i5} + erro_i$

5) Usando os dados gerados, ajuste um modelo linear sujeito à restrição $\beta_1 = \beta_3$

Antes de tudo definiremos funções que constrõem a matriz do modelo, realizam a estimação e a estimação sujeita a restrição.

```r
#' Constroe a matriz do modelo
#'
#' @param ... recebe os vetores de dados
#'
matrixX <- function(...) {
  out <- cbind(1, ...)
  n_var <- ncol(out) - 1
  colnames(out) <- c("Intercept", paste0("x", seq(1, n_var)))
  return(out)
}

#' Estimação de beta sem retrição
#'
#' @param X - matriz do modelo
#' @param y - vetor resposta
```

```r
#'
estbeta <- function(X, y) solve(t(X) %*% X) %*% t(X) %*% y

#' Estimação de beta com restrição
#'
#' @param X - matriz do modelo
#' @param y - vetor resposta
#' @param A - matriz refente a restrição
#' @param c - vetor referente a restrição
#'
estbetaRestr <- function(X, y, A, c) {
  beta_hat <- estbeta(X, y )

  beta_hat_h <- beta_hat +
    solve(t(X) %*% X) %*% t(A) %*%
    solve(A %*% solve(t(X) %*% X) %*% t(A)) %*%
    (c - A %*% beta_hat)
  return(beta_hat_h)
}
```

A restrição pedida no exercício é: $\beta_1 = \beta_3$. Essa restrição pode ser escrita como:

$$A\boldsymbol{\beta} = \mathbf{c}$$

em que $A = \begin{bmatrix} 0 & 1 & 0 & -1 & 0 & 0 \end{bmatrix}$ e $c = 0$

ou seja,

$$\begin{aligned} A\beta &= c \\ \begin{bmatrix} 0 & 1 & 0 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \beta 0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \end{bmatrix} &= 0 \\ \beta_1 - \beta_3 &= 0 \\ \beta_1 &= \beta_3 \end{aligned}$$

Abaixo temos a matriz do modelo X:

```r
X <- matrixX(x1, x2, x3, x4, x5)
head(X) # mostrando apenas as 6 primeiras linhas
```

```
##      Intercept       x1        x2        x3 x4 x5
## [1,]         1 23.96507 0.8656018  5.017537 28 36
## [2,]         1 22.61126 1.3832050 30.005475 28 47
## [3,]         1 28.73111 1.0171024 19.486933 29 38
## [4,]         1 13.64332 1.2654699 53.751855 28 27
## [5,]         1 30.98695 1.4312528 13.642574 36 40
## [6,]         1 22.16565 1.0478404 12.101594 32 27
```

Vamos encontrar as estimativas do modelo linear sujeito à restrição $\beta_1 = \beta_3$.

```
A <- matrix(c(0, 1, 0, -1, 0, 0), nrow = 1)
c <- 0
beta_res <- estbetaRestr(X, y, A, c)
beta_res
```

```
##                 [,1]
## Intercept  29.3314825
## x1          3.9968475
## x2        -13.5491556
## x3          3.9968475
## x4          0.7984622
## x5          6.9721608
```

6) Usando os dados gerados, ajuste um modelo linear e teste Ho: $\beta_1 = \beta_3$

Primeiramente, vamos ajustar o modelo: $Y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_5 x_{i5} + erro_i$, para $i = 1, 2, \cdots, n$.

```
beta_hat <- estbeta(X, y)
beta_hat
```

```
##                [,1]
## Intercept  29.153619
## x1          4.034075
## x2        -13.721423
## x3          3.993382
## x4          0.793003
## x5          6.968862
```

No item anterior já ajustamos o modelo com a restrição $\beta_1 = \beta_3$, a estimativa dos parâmetros sob a restrição é:

```
beta_res
```

```
##                 [,1]
## Intercept  29.3314825
## x1          3.9968475
## x2        -13.5491556
## x3          3.9968475
## x4          0.7984622
## x5          6.9721608
```

Abaixo definimos funções para o cálculo dos resíduos, para a soma de quadrados dos resíduos e para a estatística F.

```
#' Calculo dos residuos
#'
#' @param X - matriz do modelo
#' @param y - vetor resposta
#' @param betah - vetor com a estimativa dos parâmetros
```

```r
#'
doResiduals <- function(X, y, betah) y - X %*% betah

#' Calculo da soma de quadrados dos resíduos
#'
#' @param erro - vetor de residuos
#'
doRSS <- function(erro) t(erro) %*% erro

#' Calculo da estatística F
#'
#' @param RSSh - Soma de quadrados dos resíduos para o modelo com restrição
#' @param RSS - Soma de quadrados dos residuos para o modelo completo
#' @param n - tamanho amostral
#' @param q - ordem da restrição. Número de linhas da matriz A
#' @param p - numero de parâmetros
#'
doEstF <- function(RSSh, RSS, n, q, p) {
  return(((RSSh - RSS) / q) / (RSS / (n - p)))
}
```

Abaixo temos os calculos.

```r
residuals_c <- doResiduals(X, y, beta_hat) # residuos do modelo completo
RSS <- doRSS(residuals_c) # Residuals Sum Square modelo completo
print(paste("Soma de quadrados dos resíduos para o modelo completo:", RSS))
```

```
## [1] "Soma de quadrados dos resíduos para o modelo completo: 920.363276457889"
```

```r
residuals_r <- doResiduals(X, y, beta_res) # residuos do modelo sob restrição
RSSh <- doRSS(residuals_r) # Residuals Sum Square modelo sob restrição
print(paste("Soma de quadrados dos resíduos para o modelo com restrição:", RSSh))
```

```
## [1] "Soma de quadrados dos resíduos para o modelo com restrição: 922.410452963572"
```

```r
estF <- doEstF(RSSh, RSS, n = 50, q = 1, p = 6) # estatistica F
estF
```

```
##           [,1]
## [1,] 0.0978698
```

Vamos agora realizar o cálculo do valor-p.

```r
valorp <- pf(estF, df1 = 1, df2 = 50 - 6, lower.tail = FALSE)
valorp
```

```
##           [,1]
## [1,] 0.7558791
```

Como valor-p é "alto", ou melhor dizendo, como valor-p é maior do um nível de significância 0.05, não rejeitamos a hipótese nula e concluímos a favor de $H_0 : \beta_1 = \beta_3$.

Conferindo com a função pronta do R, temos:

```
model <- lm(y ~ 1 + x1 + x2 + x3 + x4 + x5) # modelo ajustado
print(model)
```

```
##
## Call:
## lm(formula = y ~ 1 + x1 + x2 + x3 + x4 + x5)
##
## Coefficients:
## (Intercept)           x1           x2           x3           x4           x5
##      29.154        4.034      -13.721        3.993        0.793        6.969
```

```
library(car)
```

```
## Carregando pacotes exigidos: carData
```

```
linearHypothesis(model, c("x1 = x3"))
```

```
## Linear hypothesis test
##
## Hypothesis:
## x1 - x3 = 0
##
## Model 1: restricted model
## Model 2: y ~ 1 + x1 + x2 + x3 + x4 + x5
##
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     45 922.41
## 2     44 920.36  1    2.0472 0.0979 0.7559
```

No output acima vemos que os resultados foram iguais.