Exercicio Regressao Multipla Sala

Davi Wentrick Feijó - 200016806

2023-06-02

Regressao Linear Multipla

Vamos usar esses dados:

```
##    Area Safra de Trigo Fertilizante Chuva
## 1     1             40          100    10
## 2     2             38          150    10
## 3     3             50          200    20
## 4     4             49          250    20
## 5     5             50          300    10
## 6     6             55          350    20
## 7     7             70          400    30
## 8     8             55          410    20
## 9     9             45          450    10
## 10   10             65          500    20
## 11   11             72          550    20
## 12   12             70          600    30
## 13   13             65          650    20
## 14   14             80          700    30
## 15   15             75          800    30
```

```
n <- nrow(dados) # Number of observations
n
```

Especificar os elementos dos vetores e matrizes do problema

```
## [1] 15
```

Nosso Y será a safra de trigo:

```
##       [,1]
##  [1,]   40
##  [2,]   38
##  [3,]   50
##  [4,]   49
##  [5,]   50
##  [6,]   55
##  [7,]   70
```

```
##  [8,]   55
##  [9,]   45
## [10,]   65
## [11,]   72
## [12,]   70
## [13,]   65
## [14,]   80
## [15,]   75
```

E o X será o fertilizante e o indice de chuvas:

```
##       [,1] [,2]
##  [1,]  100   10
##  [2,]  150   10
##  [3,]  200   20
##  [4,]  250   20
##  [5,]  300   10
##  [6,]  350   20
##  [7,]  400   30
##  [8,]  410   20
##  [9,]  450   10
## [10,]  500   20
## [11,]  550   20
## [12,]  600   30
## [13,]  650   20
## [14,]  700   30
## [15,]  800   30
```

podemos adicionar na matriz X o vetor de 1 para ser nosso intecepto.

```
X <- cbind(rep(1,n), X)
X
```

```
##       [,1] [,2] [,3]
##  [1,]    1  100   10
##  [2,]    1  150   10
##  [3,]    1  200   20
##  [4,]    1  250   20
##  [5,]    1  300   10
##  [6,]    1  350   20
##  [7,]    1  400   30
##  [8,]    1  410   20
##  [9,]    1  450   10
## [10,]    1  500   20
## [11,]    1  550   20
## [12,]    1  600   30
## [13,]    1  650   20
## [14,]    1  700   30
## [15,]    1  800   30
```

Vamos definir nossa matris J que sera com composta inteiramente por 1 e sera $n \times n$ (n sendo o numero de observacoes)

```
J = matrix(data = 1, nrow = n, ncol = n)
head(J)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
## [1,]    1    1    1    1    1    1    1    1    1     1     1     1     1     1
## [2,]    1    1    1    1    1    1    1    1    1     1     1     1     1     1
## [3,]    1    1    1    1    1    1    1    1    1     1     1     1     1     1
## [4,]    1    1    1    1    1    1    1    1    1     1     1     1     1     1
## [5,]    1    1    1    1    1    1    1    1    1     1     1     1     1     1
## [6,]    1    1    1    1    1    1    1    1    1     1     1     1     1     1
##      [,15]
## [1,]     1
## [2,]     1
## [3,]     1
## [4,]     1
## [5,]     1
## [6,]     1
```

Agora podemos encontrar nosso vetor de Betas sabendo que ele pode ser calculado da seguinte forma:

$$\beta = (X^T X)^{-1} X^T Y$$

```
#encontrando XTX

XTX = t(X) %*% X
XTX
```

```
##      [,1]    [,2]   [,3]
## [1,]   15    6410    300
## [2,] 6410 3343100 143200
## [3,]  300  143200   6800
```

```
#encontrando XTY

XTY = t(X) %*% Y
XTY
```

```
##        [,1]
## [1,]    879
## [2,] 409350
## [3,]  18800
```

```
#encontrando a inversa de XTX

XTX_inv = solve(XTX)
XTX_inv
```

```
##               [,1]          [,2]          [,3]
## [1,]  0.5751552282 -1.622018e-04 -2.195872e-02
## [2,] -0.0001622018  3.099398e-06 -5.811371e-05
## [3,] -0.0219587160 -5.811371e-05  2.339632e-03
```

```r
#encnontrando a matriz de parametros Beta (ela contem o Beta 0 e o Beta 1 ou todos os beta no caso de um

beta = XTX_inv %*% t(X) %*% Y #em partes

beta = solve(t(X) %*% X) %*% t(X) %*% Y #direto

beta
```

```
##             [,1]
## [1,] 26.34027254
## [2,]  0.03362537
## [3,]  0.89452440
```

Com a matriz de betas podemos ajustar os valores esperados do modelo $\hat{Y}$:

```r
#valores estimados pelo modelo (y chapeu)
Yhat <- X %*% beta # Fitted Values
Yhat
```

```
##           [,1]
##  [1,] 38.64805
##  [2,] 40.32932
##  [3,] 50.95583
##  [4,] 52.63710
##  [5,] 45.37313
##  [6,] 55.99964
##  [7,] 66.62615
##  [8,] 58.01716
##  [9,] 50.41693
## [10,] 61.04344
## [11,] 62.72471
## [12,] 73.35122
## [13,] 66.08725
## [14,] 76.71376
## [15,] 80.07630
```

Em seguida podemos calcular nossos residuos

```r
#residuos do modelo
e <- Y - Yhat # Residuals
e
```

```
##             [,1]
##  [1,]  1.3519469
##  [2,] -2.3293213
##  [3,] -0.9558336
##  [4,] -3.6371019
##  [5,]  4.6268738
##  [6,] -0.9996384
##  [7,]  3.3738493
##  [8,] -3.0171603
##  [9,] -5.4169310
```

```
## [10,]  3.9565568
## [11,]  9.2752885
## [12,] -3.3512237
## [13,] -1.0872480
## [14,]  3.2862397
## [15,] -5.0762968
```

Podemos realizar a ANOVA do modelo para isso precisaremos calcular as seguintes somas de quadrados

- Soma de quadrados da regressao (SSR ou SQR)
  - $\hat{\beta}^T X^T Y - \frac{1}{n} Y^T J Y$
- Soma de quadrados do residuo (SSE ou SQE)
  - $\varepsilon^T \varepsilon$ ou $Y^T Y - \hat{\beta}^T X^T Y$
- Soma de quadrados total (SST ou SQT)
  - $SSE + SSR$

```
sse = t(e) %*% e #soma de quadrados do residuo
sse = t(Y) %*% Y - t(beta) %*% t(X) %*% Y #segunda forma de calcular
sse
```

```
##          [,1]
## [1,] 244.2984
```

```
ssr = t(beta) %*% t(X) %*% Y - (1/n) * (t(Y) %*% J %*% Y) #soma de quadrados da regressao
ssr
```

```
##          [,1]
## [1,] 2225.302
```

```
sst = ssr +sse  #soma de quadrados total
sst
```

```
##        [,1]
## [1,] 2469.6
```

Em seguida temos que encontrar os quadrados medios:

- Quadrado Médio da regressao (MSR ou QMR)
  - $\frac{SSR}{p-1}$
- Quadrado Médio do residuo (MSE ou QME)
  - $\frac{SSE}{n-p}$

Onde n é o número de observacoes e p o número de parametros (variaveis) do modelo

```
p = length(beta)
```

Vamos obter os graus de liberdade da regressao, residuo e total

```
glreg = p-1
glres = n-p
gltot = n-1
```

```
## A quantiade de graus de liberdade relacionado a regressao é: 2
```

```
## A quantiade de graus de liberdade relacionado aos residuos é: 12
```

```
## A quantiade de graus de liberdade relacionado ao total é: 14
```

```
msr = ssr/(p-1)
msr
```

```
##          [,1]
## [1,] 1112.651
```

```
mse = sse/(n-p)
mse
```

```
##         [,1]
## [1,] 20.3582
```

Vamos calcular nosso R^2:

```
r2 = ssr / sst
r2
```

```
##           [,1]
## [1,] 0.9010777
```

Vamos calcular a soma de quadrados extra, para isso vamos ajustar um modelo somente com a variavel fertilizante e ver quanto ela explica por si só e quato que adicionar a variavel chuva vai incrementar nessa explicação. Para agilizar vamos calcular usando o a função lm()

```
modelo_1 = lm(safra_de_trigo ~ fertilizante ,data = dados)
ssr1 <- sum((fitted(modelo_1) - mean(dados$safra_de_trigo))^2)

ssrx1x2 = ssr - ssr1
```

Em seguida podemos calcular os valores F observados:

```
f_value_reg = msr/mse
f_value_reg
```

```
##          [,1]
## [1,] 54.65369
```

```r
f_value_x1 = ssr1/mse
f_value_x1
```

```
##          [,1]
## [1,] 92.50784
```

```r
f_value_x1x2 = ssrx1x2/mse
f_value_x1x2
```

```
##          [,1]
## [1,] 16.79954
```

Vamos calcular os p-valores do teste F

Podemos apresentar nossa tabela da anova

```
##   Fonte_de_variacao GL        SS      MQ F_Value
## 1         Regressao  2 2225.3016 1112.65   54.65
## 2                X1  1 1883.2931 1883.29   92.51
## 3             X1|X2  1  342.0084  342.01    16.8
## 4          Residuos 12  244.2984   20.36
## 5             Total 14 2469.6000
```

Vamos comparar com a anova do R:

```r
modelo_completo = lm(safra_de_trigo ~ fertilizante + chuva ,data = dados)
anova(modelo_completo)
```

```
## Analysis of Variance Table
##
## Response: safra_de_trigo
##              Df  Sum Sq Mean Sq F value    Pr(>F)
## fertilizante  1 1883.29 1883.29  92.508 5.446e-07 ***
## chuva         1  342.01  342.01  16.799  0.001476 **
## Residuals    12  244.30   20.36
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```