

moodle_304-06-11-24

Alan Coila Bustinza

2022-06-04

```
library(knitr)      # For knitting document and include_graphics function
library(ggplot2)    # For plotting
library('png')
```

pregunta 1

```
img1_path <- "p1_2022-06-04_112730.png"
include_graphics(img1_path)
```

Dado este conjunto de puntos:

$$\begin{bmatrix} X & 0 & 2 & 4 & 5 & 7 \\ Y & -1 & 3 & -5 & -5 & 1 \end{bmatrix}$$

¿Cuáles son las ecuaciones normales que determinan el polinomio de regresión $y = c_2 \cdot x^2 + c_1 \cdot x + c_0$

Seleccione una:

☒ a.
{ $5 \cdot c_0 + 18 \cdot c_1 + 94 \cdot c_2 = -7$, $18 \cdot c_0 + 94 \cdot c_1 + 540 \cdot c_2 = -32$, $94 \cdot c_0 + 540 \cdot c_1 + 3298 \cdot c_2 = -144$ }



Correcto!

☐ b.
{ $5 \cdot c_0 + 18 \cdot c_1 + 94 \cdot c_2 = -7$, $94 \cdot c_0 + 540 \cdot c_1 + 3298 \cdot c_2 = -32$, $540 \cdot c_0 + 3298 \cdot c_1 + 20988 \cdot c_2 = -144$ }

☐ c.
{ $5 \cdot c_0 + 18 \cdot c_1 + 94 \cdot c_2 = -7$, $18 \cdot c_0 + 94 \cdot c_1 + 94 \cdot c_2 = -7$, $94 \cdot c_0 + 540 \cdot c_1 + 3298 \cdot c_2 = -7$ }

```
x1 <- c(0,2,4,5,7)
y1 <- c(-1,3,-5,-5,1)
grado <- 2

myPhi <- function(x, n) {
  Phi <- matrix(1, length(x), n + 1)
  for (i in 1:n) {
    Phi[, i + 1] <- x^i # funciones base para el ajuste polinómico, segun el grado
  }
}
```



```

x2 <- c(0,1,3,6,9)
y2 <- c(-1,0,5,5,4)
grado <- 2
ec_normal <- function(x,y,n){
  A <- myPhi(x,n)
  B <- t(A)%*%A
  C <- t(A)%*%y
  return (cbind(B,C))
}
ec_normal(x2,y2,grado)

```

```

##      [,1] [,2] [,3] [,4]
## [1,]    5   19  127   13
## [2,]   19  127  973   81
## [3,]  127  973 7939  549

```

pregunta 3

```

img1_path <- "p3_2022-06-04_113148.png"
include_graphics(img1_path)

```

Consideramos la recta de regresión lineal $y = 0.057x - 0.364$ asociada al conjunto de datos de la tabla. Calcula el coeficiente de determinación de la regresión.

$$\begin{bmatrix} x_i & 10 & 30 & 60 & 90 & 120 \\ y_i & 0.5 & 1 & 3 & 5 & 6.5 \end{bmatrix}$$

Seleccione una:

☒ a.
0.9899 ✓

Muy bien, Sigue así!

☐ b.
0.264

☐ c.
0.29699

```

x3 <- c(10,30,60,90,120)
y3 <- c(0.5,1,3,5,6.5)
grado <- 1 #lineal

myPhi <- function(x, n) {
  Phi <- matrix(1, length(x), n + 1)
  for (i in 1:n) {
    Phi[, i + 1] <- x^i # funciones base para el ajuste polinómico, segun el grado
  }
}

```

```

    return(Phi)
}

mylssolve <- function(A, y) {
  AT <- t(A)
  return((solve(AT %*% A)) %*% AT %*% y) # la función solve nos devuelve la inversa de la matriz
}

mypolyfit <- function(x, y, n) {
  Phi <- myPhi(x, n) # construimos la matriz con myPhi
  c <- mylssolve(Phi, y) # resolvemos el sistema de ecuaciones normales con la función mylssolve
  return(c)
}

myeval <- function(x, c) {
  f <- 0
  for (i in 1:length(c)) {
    f <- f + c[i] * x^(i - 1)
  }
  return(f)
}

cof_det <- function(x,y,grado){
  cof <- mypolyfit(x,y,grado)
  # Cálculo de discrepancias de los datos con respecto a la media
  St <- sum((y - mean(y))**2)
  # Sr_L = # sumatorio de los residuos para el ajuste exponencial
  Sr_L <- sum((y - myeval(x, cof))**2)
  r_L <- (St - Sr_L) / St
  return(r_L)
}
cof_det(x3,y3,grado)

```

```
## [1] 0.9901807
```

pregunta 4

```

img1_path <- "p4_2022-06-04_113538.png"
include_graphics(img1_path)

```

Dado este conjunto de puntos:

$$\begin{bmatrix} X & 1 & 4 & 5 & 6 & 8 \\ Y & 1 & 17 & 12 & 9 & 16 \end{bmatrix}$$

Usa la recta de regresión para calcular el error estándar.

Respuesta:

5.278088

La respuesta correcta es: $\frac{\sqrt{500222}}{134}$

```
x4 <- c(1,4,5,6,8)
y4 <- c(1,17,12,9,16)
myPhi <- function(x, n) {
  Phi <- matrix(1, length(x), n + 1)
  for (i in 1:n) {
    Phi[, i + 1] <- x^i # funciones base para el ajuste polinómico, según el grado
  }
  return(Phi)
}

mylssolve <- function(A, y) {
  AT <- t(A)
  return((solve(AT %*% A)) %*% AT %*% y) # la función solve nos devuelve la inversa de la matriz
}

mypolyfit <- function(x, y, n) {
  Phi <- myPhi(x, n) # construimos la matriz con myPhi
  c <- mylssolve(Phi, y) # resolvemos el sistema de ecuaciones normales con la función mylssolve
  return(c)
}

err_st <- function(x,y,n){
  cof <- mypolyfit(x,y,n)
  s <- sum((y - myeval(x, cof))**2)
  elm <- length(x)
  e <- sqrt(s/(elm-2))
  return(e)
}
```

```
}  
err_st(x4,y4,1)
```

```
## [1] 5.278088
```