

# Ejercicios

Uriel Paluch

31/10/2021

## Algoritmos

```
DerivadaPorDefinicion <- function(x, fx){
  fprima <- rep(NA, times = length(x))

  #Se asume que todos los valores estan separados por un h constante
  h <- x[2] - x[1]

  #Diferencia progresiva
  for (i in 1:(length(x)-1)){
    fprima[i] <- (fx[i+1] - fx[i]) / h
  }

  #Diferencia regresiva
  for (i in (length(x):2)) {
    fprimaReg <- (fx[i-1] - fx[i]) / (-h)

    if (!is.na(fprima[i])){
      if(fprimaReg != fprima[i]){
        aux <- fprima[i]
        fprima[i] <- glue::glue(aux, " (P)",
                                " o ",
                                fprimaReg, " (R)" )
      }
    } else{
      fprima[i] <- fprimaReg
    }
  }

  resultado <- data.frame(x, fx, fprima)

  return(resultado)
}
```

```
Tres_puntos <- function(x, fx){
  n <- length(x)

  fprima <- rep(NA, times = n)

  h <- x[2] - x[1]
```

```

#Punto extremo
fprima[1] <- (1/(2*h))*(-3*fx[1]+4*fx[2]-fx[3])
fprima[n] <- (1/(2*(-h)))*(-3*fx[n]+4*fx[n-1]-fx[n-2])

#Punto medio
for (i in 2:(n-1)) {
  fprima[i] <- (1/(2*h))*(-fx[i-1]+fx[i+1])
}

tabla <- data.frame(x, fx, fprima)

return(tabla)
}

Cinco_puntos <- function(x, fx){
  n <- length(x)

  fprima <- rep(NA, times = n)

  h <- x[2] - x[1]

  #Punto extremo
  fprima[1] <- (1/(12*h))*(-25*fx[1]+48*fx[2]-36*fx[3]+16*fx[4]-3*fx[5])
  fprima[n] <- (1/(12*(-h)))*(-25*fx[n]+48*fx[n-1]-36*fx[n-2]+16*fx[n-3]-3*fx[n-4])

  #Punto medio
  for (i in 3:(n-2)) {
    print(i)
    fprima[i] <- (1/(12*h))*(fx[i-2]-8*fx[i-1]+8*fx[i+1]-fx[i+2])
  }

  tabla <- data.frame(x, fx, fprima)

  return(tabla)
}

Cinco_puntos_modificado <- function(x, fx){
  n <- length(x)

  fprima <- rep(NA, times = n)

  h <- x[2] - x[1]

  #Punto extremo progresivo
  for (i in 1:(n-4)) {
    fprima[i] <- (1/(12*h))*(-25*fx[i]+48*fx[i+1]-36*fx[i+2]+16*fx[i+3]-3*fx[i+4])
  }

  #Punto extremo regresivo
  for (i in n:5) {
    fprima[i] <- (1/(12*(-h)))*(-25*fx[i]+48*fx[i-1]-36*fx[i-2]+16*fx[i-3]-3*fx[i-4])
  }
}

```

```

    tabla <- data.frame(x, fx, fprima)

    return(tabla)
}

SegundaDerivada <- function (x, fx){
  n <- length(x)

  fprima <- rep(NA, times = n)

  h <- x[2] - x[1]

  #Punto medio
  for (i in 2:(n-1)) {
    fprima[i] <- (1/(h^2))*(fx[i-1]-2*fx[i]+fx[i+1])
  }

  tabla <- data.frame(x, fx, fprima)

  return(tabla)
}

```

## Ejercicio 1

A

```
print(DerivadaPorDefinicion(x = c(0.5, 0.6, 0.7), fx = c(0.4794, 0.5646, 0.6442)))
```

```
##      x      fx      fprima
## 1 0.5 0.4794      0.852
## 2 0.6 0.5646 0.796 (P) o 0.852 (R)
## 3 0.7 0.6442      0.796
```

B

```
print(DerivadaPorDefinicion(x = c(0, 0.2, 0.4), fx = c(0, 0.7414, 1.3718)))
```

```
##      x      fx      fprima
## 1 0.0 0.0000      3.707
## 2 0.2 0.7414 3.152 (P) o 3.707 (R)
## 3 0.4 1.3718      3.152
```

## Ejercicio 2

A

```
print(Tres_puntos(x = c(2.9, 3.0, 3.1, 3.2), fx = c(-4.827866, -4.240058, -3.496909, -2.596792)))
```

```
##      x      fx      fprima
## 1 2.9 -4.827866 5.101375
## 2 3.0 -4.240058 6.654785
## 3 3.1 -3.496909 8.216330
## 4 3.2 -2.596792 9.786010
```

## B

```
print(Tres_puntos(x = c(8.1, 8.3, 8.5, 8.7), fx = c(16.9441, 17.56492, 18.19056, 18.82091)))

##      x      fx    fprima
## 1 8.1 16.94410 3.092050
## 2 8.3 17.56492 3.116150
## 3 8.5 18.19056 3.139975
## 4 8.7 18.82091 3.163525
```

## Ejercicio 3

```
print(Cinco_puntos(x = c(0.2, 0.4, 0.6, 0.8, 1), fx = c(0.9798652, 0.9177710, 0.8080348, 0.6386093, 0.3843735)))

## [1] 3
##      x      fx    fprima
## 1 0.2 0.9798652 -0.1951027
## 2 0.4 0.9177710          NA
## 3 0.6 0.8080348 -0.6824175
## 4 0.8 0.6386093          NA
## 5 1.0 0.3843735 -1.5414152
```

## Ejercicio 4

i

**A y B** P: Progresiva R: Regresiva No se puede calcular el primero con regresiva porque no hay algo mas atras que el primero, de manera análoga, no se puede calcular el último con progresiva porque no hay un elemento mas allá del último.

```
print(DerivadaPorDefinicion(x = c(6.41, 6.76, 7.11, 7.46, 7.81, 8.16, 8.51), fx = c(4.0951, 5.1989, 6.1303, 6.789, 7.1079, 7.0591, 6.6598)))

##      x      fx    fprima
## 1 6.41 4.0951          3.15371428571429
## 2 6.76 5.1989    2.66114285714286 (P) o 3.15371428571429 (R)
## 3 7.11 6.1303    1.88285714285714 (P) o 2.66114285714286 (R)
## 4 7.46 6.7893    0.910285714285715 (P) o 1.88285714285714 (R)
## 5 7.81 7.1079 -0.139428571428571 (P) o 0.910285714285715 (R)
## 6 8.16 7.0591 -1.14085714285714 (P) o -0.139428571428571 (R)
## 7 8.51 6.6598    -1.14085714285714
```

**C** No tendría que tener en cuenta los extremos

```
print(Tres_puntos(x = c(6.41, 6.76, 7.11, 7.46, 7.81, 8.16, 8.51), fx = c(4.0951, 5.1989, 6.1303, 6.789, 7.1079, 7.0591, 6.6598)))

##      x      fx    fprima
## 1 6.41 4.0951    3.4000000
## 2 6.76 5.1989    2.9074286
## 3 7.11 6.1303    2.2720000
## 4 7.46 6.7893    1.3965714
## 5 7.81 7.1079    0.3854286
## 6 8.16 7.0591   -0.6401429
## 7 8.51 6.6598   -1.6415714
```

```
print(Cinco_puntos_modificado(x = c(6.41, 6.76, 7.11, 7.46, 7.81, 8.16, 8.51), fx = c(4.0951, 5.1989, 6.1303, 6.789, 7.1079, 7.0591, 6.6598)))
```

## D

```
##      x      fx      fprima
## 1 6.41 4.0951 3.2819048
## 2 6.76 5.1989 2.9562381
## 3 7.11 6.1303 2.3120714
## 4 7.46 6.7893      NA
## 5 7.81 7.1079 0.3820952
## 6 8.16 7.0591 -0.6607143
## 7 8.51 6.6598 -1.5941190
```

## ii

**A y B** P: Progresiva R: Regresiva No se puede calcular el primero con regresiva porque no hay algo mas atras que el primero, de manera análoga, no se puede calcular el último con progresiva porque no hay un elemento mas allá del último.

```
print(DerivadaPorDefinicion(x = c(2.31, 2.91, 3.51, 4.11, 4.71, 5.31, 5.91), fx = c(3.8915, 2.8249, 1.4308, 0.3549, 1.2661666666666667, 2.6673333333333333, 2.6673333333333333)))
```

```
##      x      fx      fprima
## 1 2.31 3.8915      -1.7776666666666667
## 2 2.91 2.8249      -2.3235 (P) o -1.7776666666666667 (R)
## 3 3.51 1.4308      -1.7931666666666667 (P) o -2.3235 (R)
## 4 4.11 0.3549 -0.4258333333333333 (P) o -1.7931666666666667 (R)
## 5 4.71 0.0994 1.2661666666666667 (P) o -0.4258333333333333 (R)
## 6 5.31 0.8591 2.6673333333333333 (P) o 1.2661666666666667 (R)
## 7 5.91 2.4595      2.6673333333333333
```

**C** No tendría que tener en cuenta los extremos

```
print(Tres_puntos(x = c(2.31, 2.91, 3.51, 4.11, 4.71, 5.31, 5.91), fx = c(3.8915, 2.8249, 1.4308, 0.3549, 1.2661666666666667, 2.6673333333333333, 2.6673333333333333)))
```

```
##      x      fx      fprima
## 1 2.31 3.8915 -1.5047500
## 2 2.91 2.8249 -2.0505833
## 3 3.51 1.4308 -2.0583333
## 4 4.11 0.3549 -1.1095000
## 5 4.71 0.0994 0.4201667
## 6 5.31 0.8591 1.9667500
## 7 5.91 2.4595 3.3679167
```

```
print(Cinco_puntos_modificado(x = c(2.31, 2.91, 3.51, 4.11, 4.71, 5.31, 5.91), fx = c(3.8915, 2.8249, 1.4308, 0.3549, 1.2661666666666667, 2.6673333333333333, 2.6673333333333333)))
```

## D

```
##      x      fx      fprima
## 1 2.31 3.8915 -1.0862361
## 2 2.91 2.8249 -2.1815833
## 3 3.51 1.4308 -2.2147361
## 4 4.11 0.3549      NA
## 5 4.71 0.0994 0.4770417
## 6 5.31 0.8591 2.0923056
## 7 5.91 2.4595 3.1170972
```

## Ejercicio 5

```
print(SegundaDerivada(x = c(2.31, 2.91, 3.51, 4.11, 4.71, 5.31, 5.91), fx = c(3.8915, 2.8249, 1.4308, 0.3549, 0.0994, 0.8591, 2.4595)))
```

##		x	fx	fprima
## 1	2.31	3.8915		NA
## 2	2.91	2.8249	-0.9097222	
## 3	3.51	1.4308	0.8838889	
## 4	4.11	0.3549	2.2788889	
## 5	4.71	0.0994	2.8200000	
## 6	5.31	0.8591	2.3352778	
## 7	5.91	2.4595		NA

```
print(SegundaDerivada(x = c(2.91, 4.11, 5.31), fx = c(2.8249, 0.3549, 0.8591)))
```

##		x	fx	fprima
## 1	2.91	2.8249		NA
## 2	4.11	0.3549	2.065417	
## 3	5.31	0.8591		NA