Diferenciación numérica

Uriel Paluch

31/10/2021

Dada cualquier función continua definida dentro de un intervalo cerrado, existe un polinomio que esta arbitrariamente cerca de la función en cada punto del intervalo. Ademas las derivadas e integrales de los polinomio se obtienen con facilidad.

Derivación numérica

La derivada de la función f en x_0 es:

$$f = \lim_{h \to 0} \frac{f(x_0 + h) - f(x_0)}{h}$$

Se conoce como fórmula de diferencia hacia adelante si $h > 0$ y diferencia hacia atras si $h < 0$

Se asume que todos los valores están separados por una h constante

```
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)
}

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
```

Formula de 3 puntos

```
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)
}
```

Pero si se consideran argumentos equiespaciados. Luego de realizar las sustituciones convenientes, se pueden calcular el punto medio y el punto extremo.

```
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)) {
    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)
}

print(Cinco_puntos(x = c(0.2, 0.4, 0.6, 0.8, 1), fx = c(0.9798652, 0.9177710, 0.8080348, 0.6386093, 0.3
```

```
##     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
```

```
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)

}
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
```

```
##      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(2.31, 3.51, 4.71,5.91), fx = c(3.8915, 1.4308, 0.0994, 2.4595)))
```

```
##      x     fx    fprima
## 1 2.31 3.8915        NA
## 2 3.51 1.4308 0.7842361
## 3 4.71 0.0994 2.5635417
## 4 5.91 2.4595        NA
```