Ejercicios

Uriel Paluch

16/10/2021

Guía Práctica 4: Interpolación

```
PolinomioLagrange <- function(x, fx, y){

  n <- length(x)

  l <- rep("", times = n)

  resultado <- 0

  for (i in 1:n) {
    l[i] <- fx[i]
    for (j in 1:n) {
      if (j != i){
        l[i] <- l[i] + glue::glue("*(x-",x[j],")/(",x[i],"-",x[j],")")
      }
    }
  }

  for(i in 1:n){
    resultado <- resultado + eval(parse(text=l[i]), y)
  }
  return(paste("El resultado es: ", resultado))
}
```

```
# Metodo de Neville ------------------------------------------------------

Neville <- function(x, y, interpolar){
  #cantidad de iteraciones que voy a hacer
  n <- length(x)-1

  #Hago un vector vacio para llenar el df
  empty_vec <- rep(0, times = length(x))

  df <- data.frame(x, y)

  for (i in 1:n) {
    df[glue::glue("Q",i)] <- empty_vec

    for (j in (i+1):(n+1)) {

      df[j, (i+2)] <- ( (interpolar-x[(j-i)]) * df[j,(i+1)] - (interpolar-x[j]) * df[(j-1),(i+1)] )  /
```

```
    }
  }

  return(df)
}
```

Ejercicio 1:

Use los polinomios interpolantes de Lagrange apropiados de grado uno, dos y tres para aproximar lo siguiente:

```
# x es una lista con todos los valores
# fx es la funcion que hay que aproximar
# y es el valor donde se desea aproximar la función
polinomioLagrangeGrado3 <- PolinomioLagrange(x = c(0, 0.25, 0.5, 0.75), fx = c(1, 1.64872, 2.71828, 4.48
print(polinomioLagrangeGrado3)
```

a. f (0.43) si f (0) = 1, f (0.25) = 1.64872, f (0.5) = 2.71828, f (0.75) = 4.48169

```
## [1] "El resultado es:  2.36060473408"
```

```
polinomioLagrangeGrado2 <- PolinomioLagrange(x = c(0, 0.25, 0.5), fx = c(1, 1.64872, 2.71828), y = list
print(polinomioLagrangeGrado2)
```

```
## [1] "El resultado es:  2.376382528"
```

```
polinomioLagrangeGrado2 <- PolinomioLagrange(x = c(0.25, 0.5, 0.75), fx = c(1.64872, 2.71828, 4.48169),
print(polinomioLagrangeGrado2)
```

```
## [1] "El resultado es:  2.34886312"
```

```
#Este es el que va
#Creo que es porque es el intervalo mas pequeño
polinomioLagrangeGrado1 <- PolinomioLagrange(x = c(0.25, 0.5), fx = c(1.64872, 2.71828), y = list(x = 0
print(polinomioLagrangeGrado1)
```

```
## [1] "El resultado es:  2.4188032"
```

```
polinomioLagrangeGrado3 <- PolinomioLagrange(x = c(-0.5, -0.25, 0.25, 0.5), fx = c(1.93750, 1.33203, 0.8
print(polinomioLagrangeGrado3)
```

b. f (0) si f (-0.5) = 1.93750, f (-0.25) = 1.33203, f (0.25) = 0.800781, f (0.5) = 0.687500

```
## [1] "El resultado es:  0.984374"
```

```
polinomioLagrangeGrado2 <- PolinomioLagrange(x = c(-0.5, -0.25, 0.25), fx = c(1.93750, 1.33203, 0.80078
print(polinomioLagrangeGrado2)
```

```
## [1] "El resultado es:  0.953123666666667"
```

```
polinomioLagrangeGrado2 <- PolinomioLagrange(x = c(-0.25, 0.25, 0.5), fx = c(1.33203, 0.800781, 0.68750
print(polinomioLagrangeGrado2)
```

```
## [1] "El resultado es:  1.01562433333333"
```

```
polinomioLagrangeGrado1 <- PolinomioLagrange(x = c(-0.25, 0.25), fx = c(1.33203, 0.800781), y = list(x =
print(polinomioLagrangeGrado1)
```

```
## [1] "El resultado es:  1.0664055"
```

```
polinomioLagrangeGrado3 <- PolinomioLagrange(x = c(0.1, 0.2, 0.3, 0.4), fx = c(-0.29004986, -0.56079734
print(polinomioLagrangeGrado3)
```

c. f (0.18) sif (0.1) = -0.29004986, f (0.2) = -0.56079734, f (0.3) = -0.81401972, f (0.4) =-1.0526302

```
## [1] "El resultado es:  -0.5081430744"
```

```
polinomioLagrangeGrado2 <- PolinomioLagrange(x = c(0.1, 0.2, 0.3), fx = c(-0.29004986, -0.56079734, -0.8
print(polinomioLagrangeGrado2)
```

```
## [1] "El resultado es:  -0.508049852"
```

```
polinomioLagrangeGrado1 <- PolinomioLagrange(x = c(0.1, 0.2), fx = c(-0.29004986, -0.56079734), y = lis
print(polinomioLagrangeGrado1)
```

```
## [1] "El resultado es:  -0.506647844"
```

Ejercicio 4:

Aplique el método de Neville para obtener las aproximaciones del ejercicio 1.

```
# x es la preimagen
# y es la imagen
# interpolar es el número que se desea interpolar
print(Neville(x = c(0, 0.25, 0.5, 0.75), y = c(1, 1.64872, 2.71828, 4.48169), interpolar = 0.43))
```

Ejercicio a:

```
##      x       y       Q1       Q2       Q3
## 1 0.00 1.00000 0.000000 0.000000 0.000000
## 2 0.25 1.64872 2.115798 0.000000 0.000000
## 3 0.50 2.71828 2.418803 2.376383 0.000000
## 4 0.75 4.48169 2.224525 2.348863 2.360605
```

```
# x es la preimagen
# y es la imagen
# interpolar es el número que se desea interpolar
print(Neville(x = c(-0.5, -0.25, 0.25, 0.5), y = c(1.93750, 1.33203, 0.800781, 0.687500), interpolar = 
```

Ejercicio b:

```
##       x        y       Q1        Q2       Q3
## 1 -0.50 1.937500 0.000000 0.0000000 0.000000
## 2 -0.25 1.332030 0.726560 0.0000000 0.000000
## 3  0.25 0.800781 1.066406 0.9531237 0.000000
## 4  0.50 0.687500 0.914062 1.0156243 0.984374
```

```
# x es la preimagen
# y es la imagen
# interpolar es el número que se desea interpolar
print(Neville(x = c(0.1, 0.2, 0.3, 0.4), y = c(-0.29004986, -0.56079734, -0.81401972, -1.0526302), inte
```

Ejercicio c:

```
##     x          y         Q1         Q2         Q3
## 1 0.1 -0.2900499  0.0000000  0.0000000  0.0000000
## 2 0.2 -0.5607973 -0.5066478  0.0000000  0.0000000
## 3 0.3 -0.8140197 -0.5101529 -0.5080499  0.0000000
## 4 0.4 -1.0526302 -0.5276871 -0.5083994 -0.5081431
```