

PEC 7 Aproximación de funciones y regresión(I)

Alan Coila Bustinza

2022-05-08

Pregunta 1

Comenzamos cargando las la libreria necesaria

```
library(polynom)
```

Sabemos que cada polinomio de legendre se puede expresar con la formula de rodriguez

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} [(x^2 - 1)^n].$$

Crearemos una funcion auxiliar para generar la recursion de la derivacion

```
# recibe n , la derivada enesima solicitada
nth_diff <- function(n){
  # polinomio base
  pol <- c(-1,0,1)
  #elevamos a la potencia de enesima
  a <- polynomial(pol)^n
  # derivamos recursivamente segun n
  for(k in 1:n){
    a <- deriv(a)
  }
  return(a)
}
```

Ahora calcularemos la matrix kxk con los resultados de las evaluaciones del polinomio de legendre para los valores del vector $x \in [-1,1]$, para un grado n

```
my_legendre = function(x, n){

  m = length(x)
  #p <- c(-1,0,1)
  L = matrix(1, n+1, m)
  if (n > 0){
    L[2, ] = x# Coeficiente L1
  }
  if (n > 1)
    for (k in 1:(n-1)){
      pol <- as.function(nth_diff(k+1)*(2^(k+1)*factorial(k+1))^-1)
```

```

    L[k+2, ] = pol(x)
  }
  return(L)
}

# Creamos otra funcion auxiliar que nos permitira realizar el cambio de variable y evaluar la aproximaci
myf_Legendre = function(t, ck){

  x = (2*t-(a+b))/(b-a)
  # Cambio de variables
  Lk = my_legendre(x, n)

  return(colSums(ck*Lk))
}

```

Ahora cargaremos os datos solicitados en le pregunta

```

# definimos la funcion para cargar los datos en un rango de fechas, asi como la columna que nos interes
myReadData_byDate = function(file, date_ini, date_fin, column){

  df = read.csv(file, sep = ',', row.names = 1)
  idx_Dates = as.character( seq(as.Date(date_ini, format = '%d/%m/%Y'), as.Date(date_fin, format = '%
  return( na.omit(df[idx_Dates, column]) )
}

# cargamos los nuevos casos registrados entre las fechas 01/12/2020 y 31/03/2021
Y = myReadData_byDate('WHO-COVID-19-global-data-SPAIN.csv', '01/12/2020', '31/03/2021', 'New_cases') #
m = length(Y)
t = 1:m
plot(t,Y)
# primer dia o dia 1
a = 1
# ultimo dia
b = length(Y)
# Ahora para nuestro ejercicio necesitaremos realizar un cambio de variable ya que los polinomio de leg

x = (2*t-(a+b))/(b-a)

plot(t,Y)
N = c(2,4,6,8,10)
color = c('cyan', 'yellow', 'magenta', 'blue', 'red')
#plot(NA, ylim = c(2000, 20000), xlim = c(0, 200))
for(i in 1:length(N)) {

  n = N[i]

  ## Polinomios de Legendre
  Lk = my_legendre(x, n)
  # creamos un vector con los terminos de la ortogonalidad
  Ak <- c()
  for(w in 1:(n+1)){
    o <- 2/(2*(w-1)+1)

```

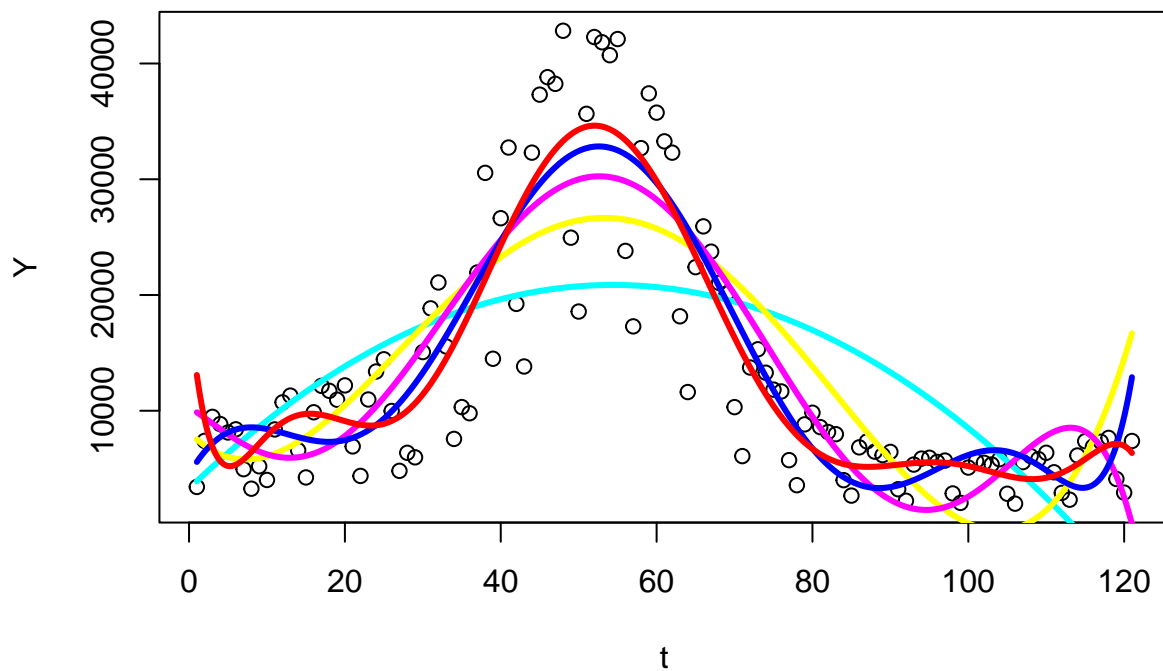
```

    Ak <- c(Ak,o)
  }
  fLk = as.vector( Lk%*%Y )
  # obtenemos los coeficientes
  ckL = (2/(b - a))*fLk/Ak

  hh = 0.01
  tt = seq(a, b, hh)

  # Evaluamos la aproximacio de grado 'n' en los puntos 'tt'
  fL = myf_Legendre(tt, ckL)
  # graficamos para cada grado
  lines(tt, fL, col=color[i], lwd=3)
}

```



La suma progresiva de los polinomios de Legendre permiten una mayor adaptabilidad a hechos de la naturaleza como es el comportamiento de una curva de contagios, en nuestro caso la curva se fue ajustando mejor a nuestro gráfico de puntos mientras iba aumentando el grado.