

Ejercicio

Uriel Paluch

27/11/2021

```
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)
}

SplineNatural <- function(x, y){
  #browser()
  n <- length(x)

  # Paso 1
  h <- rep(NA, times = (n-1))
  for (i in 1:(n-1)) {
    h[i] <- x[i+1] - x[i]
  }; rm(i)
```

```

# Paso 2
alfa <- rep(NA, times = (n-2))
for (i in 2:(n-1)) {
  alfa[i] <- (3/h[i]) * (y[i+1] - y[i]) - (3/h[i-1]) * (y[i] - y[i-1])
}

# Paso 3
mu <- rep(NA, times = n)
zeta <- rep(NA, times = n)
l <- rep(NA, times = n)

mu[1] <- 0
zeta[1] <- 0
l[1] <- 1

# Paso 4
for (i in 2:(n-1)) {
  l[i] <- 2 * (x[i+1] - x[i-1]) - h[i-1] * mu[i-1]
  mu[i] <- h[i]/l[i]
  zeta[i] <- (alfa[i] - h[i-1] * zeta[i-1])/l[i]
}

# Paso 5
l[n] <- 1
zeta[n] <- 0
c <- rep(NA, times = n)
c[n] <- 0

# Paso 6
b <- rep(NA, times = (n-1))
d <- rep(NA, times = (n-1))
for (j in (n-1):1) {
  c[j] <- zeta[j] - mu[j] * c[j+1]
  b[j] <- (y[j+1] - y[j]) / h[j] - h[j] * (c[j+1] + 2 * c[j])/3
  d[j] <- (c[j+1] - c[j]) / (3*h[j])
}

# Paso 7
resultados <- matrix(rep(NA, 4*(n-1)), nrow = (n-1), ncol = 4, byrow = F)
for (k in 1:(n-1)) {
  resultados[k, 1] <- y[k]
  resultados[k, 2] <- b[k]
  resultados[k, 3] <- c[k]
  resultados[k, 4] <- d[k]
}

print(resultados)

#Construyo el polinomio
polinomios <- rep(NA, times = nrow(resultados))
for (i in 1:nrow(resultados)) {
  polinomios[i] <- glue::glue(resultados[i,1])
}

```

```

    for(j in 2:ncol(resultados)){
      polinomios[i] <- polinomios[i] + glue::glue(" + ", resultados[i,j], " * (x - ", x[i], ")^", (j-1))
    }
  }

  return(polinomios)
}

SplineCondicionado <- function(x, y, fpo, fpn){
  #browser()
  n <- length(x)

  # Paso 1
  h <- rep(NA, times = (n-1))
  for (i in 1:(n-1)) {
    h[i] <- x[i+1] - x[i]
  }; rm(i)

  # Paso 2
  alfa <- rep(NA, times = n)
  alfa[1] <- 3 * (y[2] - y[1])/h[1] - 3 * fpo
  alfa[n] <- 3 * fpn - 3 * (y[n] - y[n-1]) / h[n-1]

  # Paso 3
  for (i in 2:(n-1)) {
    alfa[i] <- (3/h[i]) * (y[i+1] - y[i]) - (3/h[i-1]) * (y[i] - y[i-1])
  }; rm(i)

  # Paso 4
  mu <- rep(NA, times = n)
  zeta <- rep(NA, times = n)
  l <- rep(NA, times = n)

  l[1] <- 2 * h[1]
  mu[1] <- 0.5
  zeta[1] <- alfa[1]/l[1]

  # Paso 5
  for (i in 2:(n-1)) {
    l[i] <- 2 * (x[i+1] - x[i-1]) - h[i-1] * mu[i-1]
    mu[i] <- h[i]/l[i]
    zeta[i] <- (alfa[i] - h[i-1] * zeta[i-1])/l[i]
  }

  # Paso 6
  l[n] <- h[n-1] * (2 - mu[n-1])
  zeta[n] <- (alfa[n] - h[n-1] * zeta[n-1]) / l[n]
  c <- rep(NA, times = n)
  c[n] <- zeta[n]

  # Paso 7
  b <- rep(NA, times = (n-1))

```

```

d <- rep(NA, times = (n-1))
for (j in (n-1):1) {
  c[j] <- zeta[j] - mu[j] * c[j+1]
  b[j] <- (y[j+1] - y[j]) / h[j] - h[j] * (c[j+1] + 2 * c[j])/3
  d[j] <- (c[j+1] - c[j]) / (3*h[j])
}

#Paso 7
resultados <- matrix(rep(NA, 4*(n-1)), nrow = (n-1), ncol = 4, byrow = F)
for (k in 1:(n-1)) {
  resultados[k, 1] <- y[k]
  resultados[k, 2] <- b[k]
  resultados[k, 3] <- c[k]
  resultados[k, 4] <- d[k]
}

print(resultados)

#Construyo el polinomio
polinomios <- rep(NA, times = nrow(resultados))
for (i in 1:nrow(resultados)) {
  polinomios[i] <- glue::glue(resultados[i,1])
  for(j in 2:ncol(resultados)){
    polinomios[i] <- polinomios[i] + glue::glue(" + ", resultados[i,j], " * (x - ", x[i], ")^", (j-1))
  }
}

return(polinomios)
}

```

Métodos

```

# Cargo el df
df <- data.frame("Tasa" = seq(from = 0, to = 0.1, by = 0.01), "Precio" = c(120, 114.56029, 109.42692, 105.38571, 101.84286, 98.79846, 96.25000, 94.19643, 92.63673, 91.56000, 90.96619, 90.84444, 91.18556, 91.98889, 93.25455, 94.97222, 97.14286, 99.75781, 102.81818, 106.32407, 110.27556, 114.67222, 119.51429, 124.80000, 130.52941, 136.70222, 143.31818, 150.37778, 157.88111, 165.82857, 174.22000, 183.05556, 192.33333, 202.05263, 212.21333, 222.81556, 233.85926, 245.34444, 257.27111, 269.63889, 282.44778, 295.69778, 309.38889, 323.52000, 338.09111, 353.10222, 368.55444, 384.44778, 400.78333, 417.56000, 434.77778, 452.43556, 470.53333, 489.07111, 508.04889, 527.46667, 547.32444, 567.62222, 588.36000, 609.53778, 631.15556, 653.21333, 675.71111, 698.64889, 722.02667, 745.84444, 770.10222, 794.80000, 819.93778, 845.51556, 871.53333, 898.00000, 924.91111, 952.26889, 980.07111, 1008.31889, 1037.01111, 1066.14889, 1095.73333, 1125.76444, 1156.24222, 1187.16667, 1218.53778, 1250.35556, 1282.61889, 1315.32667, 1348.48000, 1382.07778, 1416.12000, 1450.60778, 1485.54000, 1520.91667, 1556.73333, 1592.99000, 1629.68889, 1666.83333, 1704.42222, 1742.45556, 1780.93333, 1819.85556, 1859.22222, 1899.03333, 1939.28889, 1979.98889, 2021.13333, 2062.72222, 2104.75556, 2147.23333, 2190.15556, 2233.52222, 2277.33333, 2321.58889, 2366.28889, 2411.43333, 2457.02222, 2503.05556, 2549.53333, 2596.45556, 2643.82222, 2691.63333, 2739.88889, 2788.58889, 2837.73333, 2887.32222, 2937.35556, 2987.83333, 3038.75556, 3089.12222, 3140.03333, 3191.48889, 3243.48889, 3296.03333, 3349.12222, 3402.75556, 3456.93333, 3511.65556, 3566.92222, 3622.73333, 3679.08889, 3735.98889, 3793.43333, 3851.42222, 3909.95556, 3969.03333, 4028.65556, 4088.82222, 4149.53333, 4210.78889, 4272.58889, 4334.93333, 4397.82222, 4461.25556, 4525.23333, 4589.75556, 4654.82222, 4720.43333, 4786.58889, 4853.28889, 4920.53333, 4988.33333, 5056.68889, 5125.58889, 5195.03333, 5265.02222, 5335.55556, 5406.63333, 5478.25556, 5550.42222, 5623.13333, 5696.38889, 5770.18889, 5844.53333, 5919.42222, 5994.85556, 6070.83333, 6147.35556, 6224.42222, 6302.03333, 6380.18889, 6458.88889, 6538.13333, 6617.92222, 6698.25556, 6779.13333, 6860.55556, 6942.52222, 7025.03333, 7108.08889, 7191.68889, 7275.83333, 7360.52222, 7445.75556, 7531.53333, 7617.85556, 7704.72222, 7792.13333, 7880.08889, 7968.58889, 8057.63333, 8147.22222, 8237.35556, 8328.03333, 8419.25556, 8511.02222, 8603.33333, 8696.18889, 8789.58889, 8883.53333, 8978.02222, 9073.05556, 9168.63333, 9264.75556, 9361.42222, 9458.63333, 9556.38889, 9654.68889, 9753.53333, 9852.92222, 9952.85556, 10053.33333, 10154.35556, 10255.92222, 10358.03333, 10460.68889, 10563.88889, 10667.63333, 10771.92222, 10876.75556, 10982.12222, 11088.03333, 11194.48889, 11301.48889, 11409.03333, 11517.12222, 11625.75556, 11734.92222, 11844.62222, 11954.85556, 12065.62222, 12176.92222, 12288.75556, 12401.12222, 12514.03333, 12627.48889, 12741.48889, 12856.03333, 12971.12222, 13086.75556, 13202.92222, 13319.62222, 13436.85556, 13554.62222, 13672.92222, 13791.75556, 13911.12222, 14031.03333, 14151.48889, 14272.48889, 14394.03333, 14516.12222, 14638.75556, 14761.92222, 14885.62222, 15009.85556, 15134.62222, 15259.92222, 15385.75556, 15512.03333, 15638.75556, 15765.92222, 15893.58889, 16021.75556, 16150.42222, 16279.58889, 16409.25556, 16539.42222, 16669.98889, 16800.98889, 16932.48889, 17064.48889, 17196.98889, 17329.98889, 17463.48889, 17597.48889, 17731.98889, 17866.98889, 18002.48889, 18138.48889, 18274.98889, 18411.98889, 18549.48889, 18687.48889, 18825.98889, 18964.98889, 19104.48889, 19244.48889, 19384.98889, 19525.98889, 19667.48889, 19809.48889, 19951.98889, 20094.98889, 20238.48889, 20381.48889, 20524.98889, 20668.98889, 20813.48889, 20958.48889, 21103.98889, 21249.98889, 21396.48889, 21543.98889, 21691.48889, 21839.98889, 21988.48889, 22137.98889, 22287.48889, 22437.98889, 22588.48889, 22739.98889, 22891.48889, 23043.98889, 23196.48889, 23349.98889, 23503.98889, 23657.48889, 23811.98889, 23966.48889, 24121.98889, 24277.48889, 24433.98889, 24590.48889, 24747.98889, 24905.98889, 25064.48889, 25223.48889, 25382.98889, 25542.98889, 25703.48889, 25864.48889, 26025.98889, 26187.98889, 26350.48889, 26513.98889, 26677.98889, 26842.48889, 27007.48889, 27172.98889, 27338.98889, 27505.48889, 27672.98889, 27840.48889, 28008.98889, 28177.98889, 28347.48889, 28517.48889, 28687.98889, 28858.98889, 29030.48889, 29202.48889, 29374.98889, 29547.98889, 29721.48889, 29895.48889, 30069.98889, 30244.98889, 30420.48889, 30595.98889, 30771.98889, 30948.48889, 31124.98889, 31301.98889, 31479.48889, 31656.98889, 31834.98889, 32013.48889, 32191.98889, 32370.48889, 32549.98889, 32729.48889, 32909.98889, 33090.48889, 33271.98889, 33453.48889, 33635.98889, 33818.48889, 34001.98889, 34185.48889, 34369.98889, 34554.48889, 34739.98889, 34925.48889, 35111.98889, 35298.48889, 35485.98889, 35673.48889, 35861.98889, 36050.48889, 36239.98889, 36429.48889, 36619.98889, 36810.48889, 37001.98889, 37193.48889, 37385.98889, 37578.48889, 37771.98889, 37965.48889, 38159.98889, 38354.48889, 38549.98889, 38745.48889, 38941.98889, 39138.48889, 39335.98889, 39533.48889, 39731.98889, 39930.48889, 40129.98889, 40329.48889, 40529.98889, 40730.48889, 40931.98889, 41133.48889, 41335.98889, 41538.48889, 41741.98889, 41945.48889, 42149.98889, 42354.48889, 42559.98889, 42765.48889, 42971.98889, 43178.48889, 43385.98889, 43593.48889, 43801.98889, 44010.48889, 44219.98889, 44429.48889, 44639.98889, 44850.48889, 45061.98889, 45273.48889, 45485.98889, 45698.48889, 45911.98889, 46125.48889, 46339.98889, 46554.48889, 46769.98889, 46985.48889, 47201.98889, 47418.48889, 47635.98889, 47853.48889, 48071.98889, 48290.48889, 48509.98889, 48729.48889, 48949.98889, 49170.48889, 49391.98889, 49613.48889, 49835.98889, 50058.48889, 50281.98889, 50505.48889, 50729.98889, 50954.48889, 51179.98889, 51405.48889, 51631.98889, 51858.48889, 52085.98889, 52313.48889, 52541.98889, 52770.48889, 53000.48889, 53230.98889, 53461.48889, 53692.98889, 53924.48889, 54156.98889, 54389.48889, 54621.98889, 54854.48889, 55087.98889, 55321.48889, 55555.98889, 55790.48889, 56025.98889, 56261.48889, 56497.98889, 56734.48889, 56971.98889, 57209.48889, 57447.98889, 57686.48889, 57925.98889, 58165.48889, 58405.98889, 58646.48889, 58887.98889, 59129.48889, 59371.98889, 59614.48889, 59857.98889, 60101.48889, 60345.98889, 60590.48889, 60835.98889, 61081.48889, 61327.98889, 61574.48889, 61821.98889, 62069.48889, 62317.98889, 62566.48889, 62815.98889, 63065.48889, 63315.98889, 63566.48889, 63817.98889, 64069.48889, 64321.98889, 64574.48889, 64827.98889, 65081.48889, 65335.98889, 65590.48889, 65845.98889, 66101.48889, 66357.98889, 66614.48889, 66871.98889, 67129.48889, 67387.98889, 67646.48889, 67905.98889, 68165.48889, 68425.98889, 68686.48889, 68947.98889, 69209.48889, 69471.98889, 69734.48889, 70000.48889, 70265.98889, 70531.48889, 70797.98889, 71064.48889, 71331.98889, 71600.48889, 71868.98889, 72137.98889, 72407.48889, 72677.98889, 72948.48889, 73219.98889, 73491.48889, 73763.98889, 74036.48889, 74309.98889, 74583.48889, 74857.98889, 75132.48889, 75407.98889, 75683.48889, 75959.98889, 76236.48889, 76513.98889, 76791.48889, 77069.98889, 77348.48889, 77627.98889, 77907.48889, 78187.98889, 78468.48889, 78749.98889, 79031.48889, 79313.98889, 79596.48889, 79879.98889, 80163.48889, 80447.98889, 80732.48889, 81017.98889, 81303.48889, 81589.98889, 81876.48889, 82163.98889, 82451.48889, 82739.98889, 83028.48889, 83317.98889, 83607.48889, 83897.98889, 84188.48889, 84479.98889, 84771.48889, 85063.98889, 85356.48889, 85649.98889, 85943.48889, 86237.98889, 86532.48889, 86827.98889, 87123.48889, 87419.98889, 87716.48889, 88013.98889, 88311.48889, 88609.98889, 88908.48889, 89207.98889, 89507.48889, 89807.98889, 90108.48889, 90409.98889, 90711.48889, 91013.98889, 91316.48889, 91619.98889, 91923.48889, 92227.98889, 92532.48889, 92837.98889, 93143.48889, 93449.98889, 93756.48889, 94063.98889, 94371.48889, 94679.98889, 94988.48889, 95297.98889, 95607.48889, 95917.98889, 96228.48889, 96539.98889, 96851.48889, 97163.98889, 97476.48889, 97789.98889, 98103.48889, 98417.98889, 98732.48889, 99047.98889, 99363.48889, 99679.98889, 99995.98889, 100311.48889, 100627.98889, 100944.48889, 101261.98889, 101579.48889, 101897.98889, 102216.48889, 102535.98889, 102855.48889, 103175.98889, 103496.48889, 103817.98889, 104139.48889, 104461.98889, 104784.48889, 105107.98889, 105431.48889, 105755.98889, 106080.48889, 106405.98889, 106731.48889, 107057.98889, 107384.48889, 107711.98889, 108039.48889, 108367.98889, 108696.48889, 109025.98889, 109355.48889, 109685.98889, 110016.48889, 110347.98889, 110679.48889, 111011.98889, 111344.48889, 111677.98889, 112011.48889, 112345.98889, 112680.48889, 113015.98889, 113351.48889, 113687.98889, 114024.48889, 114361.98889, 114699.48889, 115037.98889, 115376.48889, 115715.98889, 116055.48889, 116395.98889, 116736.48889, 117077.98889, 117419.48889, 117761.98889, 118104.48889, 118447.98889, 118791.48889, 119135.98889, 119480.48889, 119825.98889, 120171.48889, 120517.98889, 120864.48889, 121211.98889, 121559.48889, 121907.98889, 122256.48889, 122605.98889, 122955.48889, 123305.98889, 123656.48889, 124007.98889, 124359.48889, 124711.98889, 125064.48889, 125417.98889, 125771.48889, 126125.98889, 126480.48889, 126835.98889, 127191.48889, 127547.98889, 127904.48889, 128261.98889, 128619.48889, 128977.98889, 129336.48889, 129695.98889, 130055.48889, 130415.98889, 130776.48889, 131137.98889, 131499.48889, 131861.98889, 132224.48889, 132587.98889, 132951.48889, 133315.98889, 133680.48889, 134045.98889, 134411.48889, 134777.98889, 135144.48889, 135511.98889, 135879.48889, 136247.98889, 136616.48889, 136985.98889, 137355.48889, 137725.98889, 138096.48889, 138467.98889, 138839.48889, 139211.98889, 139584.48889, 140000.48889, 140416.48889, 140832.48889, 141248.48889, 141664.48889, 142080.48889, 142496.48889, 142912.48889, 143328.48889, 143744.48889, 144160.48889, 144576.48889, 145000.48889, 145424.48889, 145848.48889, 146272.48889, 146696.48889, 147120.48889, 147544.48889, 147968.48889, 148392.48889, 148816.48889, 149240.48889, 149664.48889, 150088.48889, 150512.48889, 150936.48889, 151360.48889, 151784.48889, 152208.48889, 152632.48889, 153056.48889, 153480.48889, 153904.48889, 154328.48889, 154752.48889, 155176.48889, 155600.48889, 156024.48889, 156448.48889, 156872.48889, 157296.48889, 157720.48889, 158144.48889, 158568.48889, 158992.48889, 159416.48889, 159840.48889, 160264.48889, 160688.48889, 161112.48889, 161536.48889, 161960.48889, 162384.48889, 162808.48889, 163232.48889, 163656.48889, 164080.48889, 164504.48889, 164928.48889, 165352.48889, 165776.48889, 166200.48889, 166624.48889, 167048.48889, 167472.48889, 167896.48889, 168320.48889, 168744.48889, 169168.48889, 169592.48889, 170016.48889, 170440.48889, 170864.48889, 171288.48889, 171712.48889, 172136.48889, 172560.48889, 172984.48889, 173408.48889, 173832.48889, 174256.48889, 174680.48889, 175104.48889, 175528.48889, 175952.48889, 176376.48889, 176800.48889, 177224.48889, 177648.48889, 178072.48889, 178496.48889, 178920.48889, 179344.48889, 179768.48889, 180192.48889, 180616.48889, 181040.48889, 181464.48889, 181888.48889, 182312.48889, 182736.48889, 183160.48889, 183584.48889, 184008.48889, 184432.48889, 184856.48889, 185280.48889, 185704.48889, 186128.48889, 186552.48889, 186976.48889, 187400.48889, 187824.48889, 188248.48889, 188672.48889, 189096.48889, 189520.48889, 189944.48889, 190368.48889, 190792.48889, 191216.48889, 191640.48889, 192064.48889, 192488.48889, 192912.48889, 193336.48889, 193760.48889, 194184.48889, 194608.48889, 195032.48889, 195456.48889, 195880.48889, 196304.48889, 196728.48889, 197152.48889, 197576.48889, 198000.48889, 198424.48889, 198848.48889, 199272.48889, 199696.48889, 200120.48889, 200544.48889, 200968.48889, 201392.48889, 201816.48889, 202240.48889, 202664.48889, 203088.48889, 2
```

Tasa	Precio
0.00	120.00000
0.01	114.56029
0.02	109.42692
0.03	104.57971
0.04	100.00000
0.05	95.67052
0.06	91.57527
0.07	87.69941
0.08	84.02916
0.09	80.55174
0.10	77.25528

DERIVACIÓN NUMÉRICA E INTERPOLACIÓN

Considere la tabla de tasas de interés (en tanto por uno) y Precio, suponiendo que el Precio es una función de la tasa: $\text{Precio} = P(r)$.

- Aproxime la derivada $P'(0.00)$ y $P'(0.10)$ usando el método que considere más preciso (justifique).
- Estime $P(0.095)$ mediante un Cubic Spline Natural.
- Estime $P(0.095)$ mediante un Cubic Spline Sujeto usando las derivadas estimadas en a).

[Escriba a continuación la respuesta, y cargue en el zip que entrega mediante campus el Script de R con la resolución.]

(15 puntos)

Figure 1: Consigna

```
trazadores <- SplineNatural(x = df$Tasa, y = df$Precio)
```

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] 120.00000 -550.5363    0.0000 65653.257
## [2,] 114.56029 -530.8403 1969.5977 -21926.283
## [3,] 109.42692 -498.0263 1311.8092  1871.876
## [4,] 104.57971 -471.2285 1367.9655 -4221.221
## [5,] 100.00000 -445.1356 1241.3289 -2256.993
## [6,]  95.67052 -420.9861 1173.6191 -2750.806
## [7,]  91.57527 -398.3390 1091.0949 -1579.784
## [8,]  87.69941 -376.9910 1043.7014 -4710.058
## [9,]  84.02916 -357.5300  902.3996  7640.015
## [10,] 80.55174 -337.1900 1131.6001 -37720.003
```

```
eval(parse(text = trazadores[10]), list(x = 0.095))
```

```
## [1] 78.88936
```

C

```
trazadores_condicionados <- SplineCondicionado(x = df$Tasa, y = df$Precio, fpo = (120 - 114.56029)/0.1,
```

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] 120.00000   54.3971 -104777.34927 4494053.927
## [2,] 114.56029 -692.9337  30044.26854 -1208459.781
## [3,] 109.42692 -454.5863  -6209.52489  319605.198
## [4,] 104.57971 -482.8952   3378.63104  -88621.009
## [5,] 100.00000 -441.9089    720.00075  17608.840
## [6,]  95.67052 -422.2262   1248.26596   2185.649
## [7,]  91.57527 -396.6052   1313.83542  -41191.435
## [8,]  87.69941 -382.6859    78.09236  148800.092
## [9,]  84.02916 -336.4841   4542.09513 -566788.934
## [10,] 80.55174 -415.6788 -12461.57290 2106485.645
```

```
eval(parse(text = trazadores_condicionados[10]), list(x = 0.095))
```

```
## [1] 78.42512
```