Ejercicios

Uriel Paluch

25/11/2021

3 Integración y Derivación Numérica (20 puntos)

Considere la siguiente función de densidad de la variable S:

$$f_S(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln(x)-\mu)^2}{2\sigma^2}}; \text{con } x > 0.$$

3.1 $E(S|\mu;\sigma)$

Calcule mediante una integral numérica, usando Simpson Compuesto con $n = 1000$, la esperanza matemática de la variable aleatoria S.

```
# Ingrese en este bloque todo el código necesario para resolver el ejercicio 3.1
```

3.2 Derivada respecto a μ

Estime numéricamente la derivada de $E(S|\mu;\sigma)$ respecto del parámetro μ.

```
# Ingrese en este bloque todo el código necesario para resolver el ejercicio 3.2
```

3.3 Derivada respecto a σ

Estime numéricamente la derivada de $E(S|\mu;\sigma)$ respecto del parámetro σ.

```
# Ingrese en este bloque todo el código necesario para resolver el ejercicio 3.3
```

Figure 1: Curva de Letras del Tesoro

Métodos

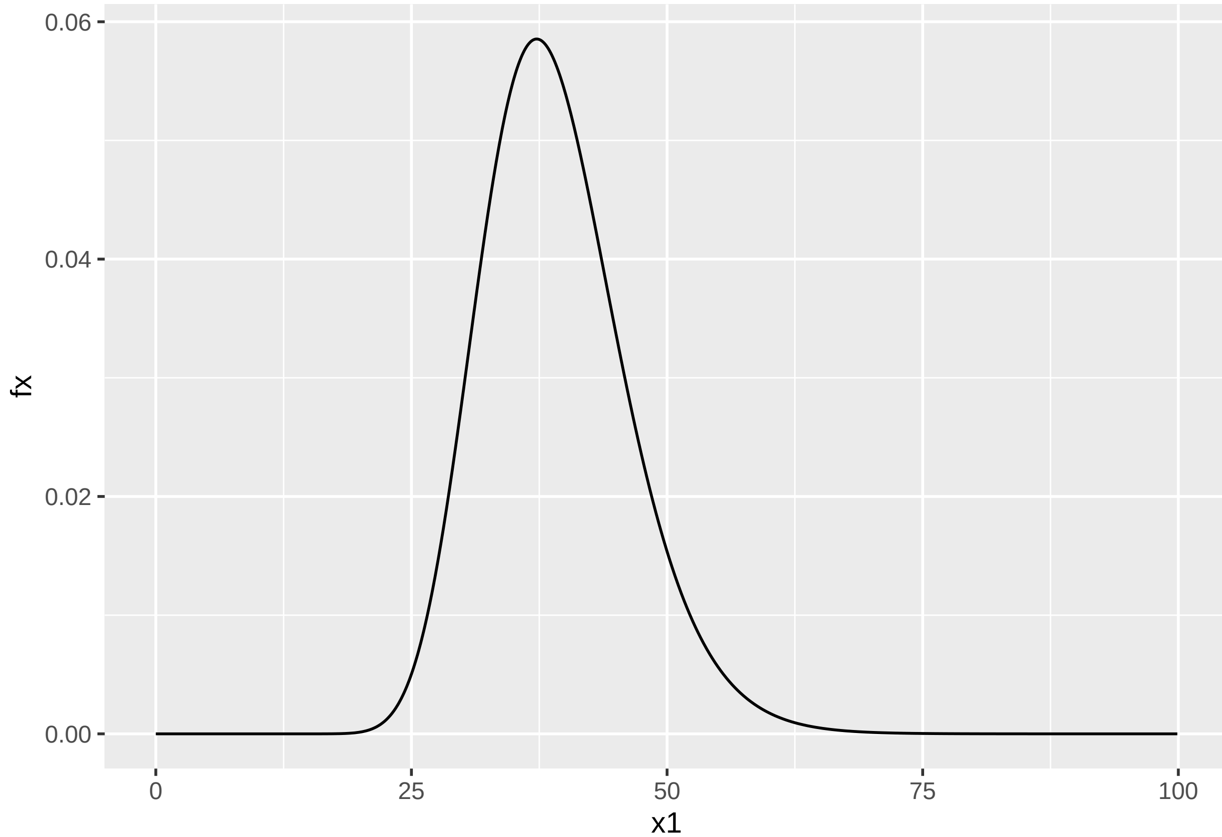
```
IntegracionCompuesta <- function(limiteInferior, limiteSuperior, funcion, n, cantIntervalos){

  #browser()
  if ((n == 2 || n == 0) && cantIntervalos%%2 != 0){
    return("cantIntervalos debe ser un entero par")
  }
```

```r
  cantIntervalos <- cantIntervalos/n


  crecimientoIntervalo <- (limiteSuperior-limiteInferior)/cantIntervalos

  fx <- rep(NA, times = (n+1))

  resultado <- 0

  for (i in 1:cantIntervalos) {
    limiteSuperior <- limiteInferior + crecimientoIntervalo

    if (n != 0){
      h <- (limiteSuperior - limiteInferior)/n
    }


    for (i in 1:(n+1)) {
      fx[i] <- eval(funcion, list(x = limiteInferior + (i-1)*h))
    }

    if(n == 2){
      resultado <- resultado + (h/3) * (fx[1] + 4*fx[2] + fx[3])
    }

    limiteInferior <- limiteSuperior
  }

  return(resultado)
}
```

```r
CompuestaSimpson <- function(limINF, limSUP, funcion, n){
  h <- (limSUP - limINF)/n
  fx0 <- eval(funcion, list(x = limINF)) #(f(a))
  fxn <- eval(funcion, list(x = limSUP)) #(f(b))

  suma1 <- 0 # la de X_2j
  suma2 <- 0 # la de X_2j-1

  for (j in 1:(n-1)) {
    xj <- limINF + j*h
    if ((j %% 2) == 0){
      suma1 <- suma1 + eval(funcion, list(x = xj))
    } else {
      suma2 <- suma2 + eval(funcion, list(x = xj))
    }
  }

  intCompuestaS <- h * (fx0 + 2 * suma1 + 4 * suma2 + fxn)/3
  return(intCompuestaS)

# n debe ser par si o si
}
```

Ejercicio 3.1

```r
mu <- 3.65
sigma <- 0.18

x1 <- seq(from = 0.0001, to = 100, by = 0.1)
fx <- eval(expression( ((1)/(x*sigma*sqrt(2*pi))) * exp(- (log(x)-mu)^2/(2*sigma^2)) ), list(x = x1))

ggplot() +
  geom_line(aes(x = x1, y = fx))
```




```r
tetas <- IntegracionCompuesta(limiteInferior = 0.0001, limiteSuperior = 1000, funcion = expression(x *

culo <- CompuestaSimpson(limINF = 0.0001, limSUP = 1000, funcion = expression(x * ((1)/(x*sigma*sqrt(2*p
```

Ejercicio 3.2

Vuelvo a la definición clásica de derivada:

$$\lim_{x \to \infty} f'(x) = \frac{f(x+h) - f(x)}{h}$$

```
fx <- IntegracionCompuesta(limiteInferior = 1, limiteSuperior = 100, funcion = expression(x * ((1)/(x*si

mu <- 3.65 + 10^-10

fx_h <- IntegracionCompuesta(limiteInferior = 1, limiteSuperior = 100, funcion = expression(x * ((1)/(x*

(fx_h - fx) / (10^-10)
```

```
## [1] 39.10287
```

Ejercicio 3.3

Vuelvo a la definición clásica de derivada:

$$\lim_{x \to \infty} f'(x) = \frac{f(x+h) - f(x)}{h}$$

```
fx <- IntegracionCompuesta(limiteInferior = 1, limiteSuperior = 100, funcion = expression(x * ((1)/(x*si

sigma <- 0.18 + 10^-10

fx_h <- IntegracionCompuesta(limiteInferior = 1, limiteSuperior = 100, funcion = expression(x * ((1)/(x*

(fx_h - fx) / (10^-10)
```

```
## [1] 7.037855
```