

Extra integrales

Diego Hernández Jiménez

Objetivo: construir función que permita obtener la integral definida de una función sobre un intervalo dado.

`integral_definida()`

- Descripción: aproxima la integral definida de una función sobre [a,b], dado un número de particiones.
- Argumentos:
 - **funcion**: class=function. Función matemática que queremos integrar.
 - **a**: class=numeric. Límite inferior de integración.
 - **b**: class=numeric. Límite superior de integración.
 - **n**: class=numeric. Número de particiones.
 - **aprox**: class=character. Técnica de integración elegida. La integral definida se puede aproximar mediante rectángulos ('**rect**'), la regla de los trapecios ('**trap**') o mediante la regla de Simpson ('**simp**'). Ésta última proporciona las mejores aproximaciones, pero cuando n es un número par.

Aproximación por rectángulos: $\int_a^b f(x) dx = \lim_{n \rightarrow \infty} I_n = \lim_{n \rightarrow \infty} S_n$.

$$I_n = \sum_{i=1}^n f(m_i) \cdot \left(\frac{b-a}{n}\right) ; S_n = \sum_{i=1}^n f(M_i) \cdot \left(\frac{b-a}{n}\right)$$

Regla de los trapecios: $\int_a^b f(x) dx \approx \frac{b-a}{2n} [f(x_0) + 2f(x_1) + 2f(x_2) + 2f(x_3) + \dots + 2f(x_{n-2}) + 2f(x_{n-1}) + f(x_n)]$

Regla de Simpson: $\int_a^b f(x) dx \approx \frac{b-a}{3n} [f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \dots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)]$

- Devuelve:
 - Si **aprox**='rect': lista con dos elementos class=numeric que representan la suma inferior (In) y la suma superior (Sn)
 - Si **aprox**='trap' o **aprox**='simp': vector de longitud uno que representa la estimación de la integral definida sobre a y b. Class=numeric

```
integral_definida <- function(funcion,a,b,n,aprox='rect'){  
  
  interv <- (b-a)/n  
  
  seg <- seq(from=a,to=b,by=interv)
```

```

if (aprox=='trap'){

  strt_end <- c(function(seg[1]),function(seg[n+1]))
  mid <- 2*function(seg[2:n])

  return(( (b-a)/(2*n) ) *sum(strt_end,mid))
}
else if (aprox=='simp'){

  strt_end <- c(function(seg[1]),function(seg[n+1]))
  seg_cut <- seg[2:n]

  indices <- 1:length(seg_cut)
  mid <- c(4*function(seg_cut[indices%%2 != 0]),
          2*function(seg_cut[indices%%2 == 0]))

  return(( (b-a)/(3*n) ) *sum(strt_end,mid))
}
else{

  In <- double(1)
  Sn <- double(1)

  for (i in 1:n){
    fmi <- min(function(seg[i]:seg[i+1]))
    In <- In+fmi*interv
    fMi <- max(function(seg[i]:seg[i+1]))
    Sn <- Sn+fMi*interv
  }
  return(list(In=In,Sn=Sn))
}
}

integral_definida(dnorm,0,1.5,10)

```

```

## $In
## [1] 0.4530353
##
## $Sn
## [1] 0.4530353

```

Estos resultados parecen corresponderse con lo esperado, la discrepancia se debe tan solo al bajo número de particiones utilizadas

```
pnorm(1.5)-pnorm(0)
```

```
## [1] 0.4331928
```

Podemos ver como convergen las sumas inferior y superior a medida que aumenta el número de particiones

```

vector_In <- double(0)
vector_Sn <- double(0)

for (n in 1:100){
  vector_In <- append(vector_In,integral_definida(dnorm,0,1.5,n)$In)
  vector_Sn <- append(vector_Sn,integral_definida(dnorm,0,1.5,n)$Sn)
}

df <- data.frame(n=1:100,
                 Suma_inf=vector_In,
                 Suma_sup=vector_Sn)
tail(df)

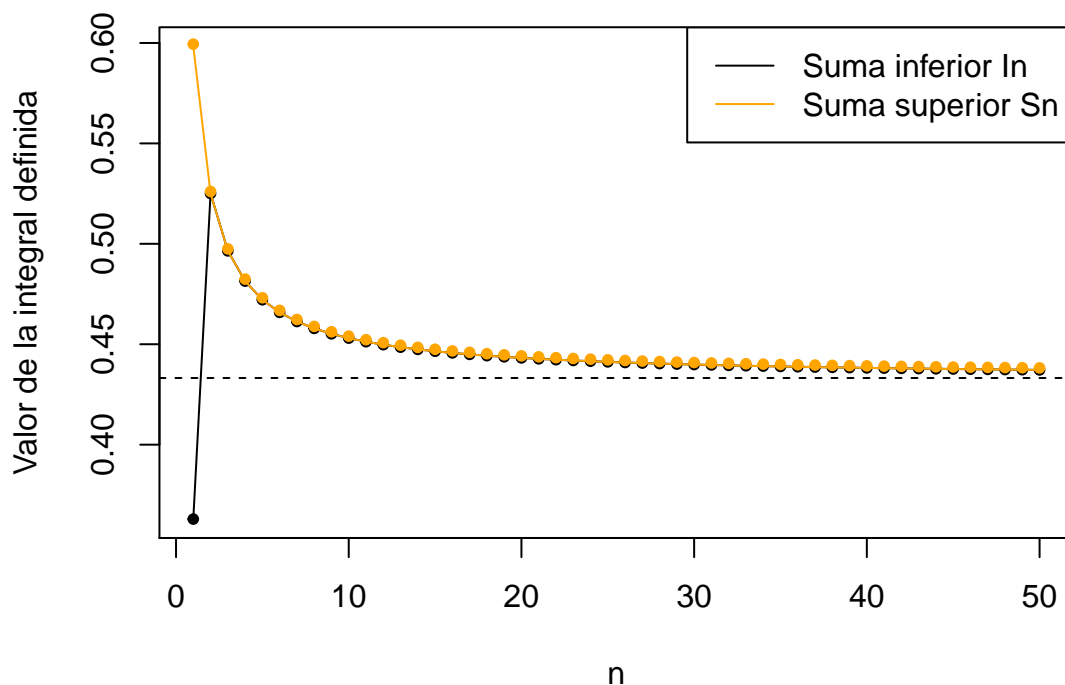
```

```

##      n Suma_inf Suma_sup
## 95  95 0.4353158 0.4353158
## 96  96 0.4352937 0.4352937
## 97  97 0.4352721 0.4352721
## 98  98 0.4352509 0.4352509
## 99  99 0.4352302 0.4352302
## 100 100 0.4352098 0.4352098

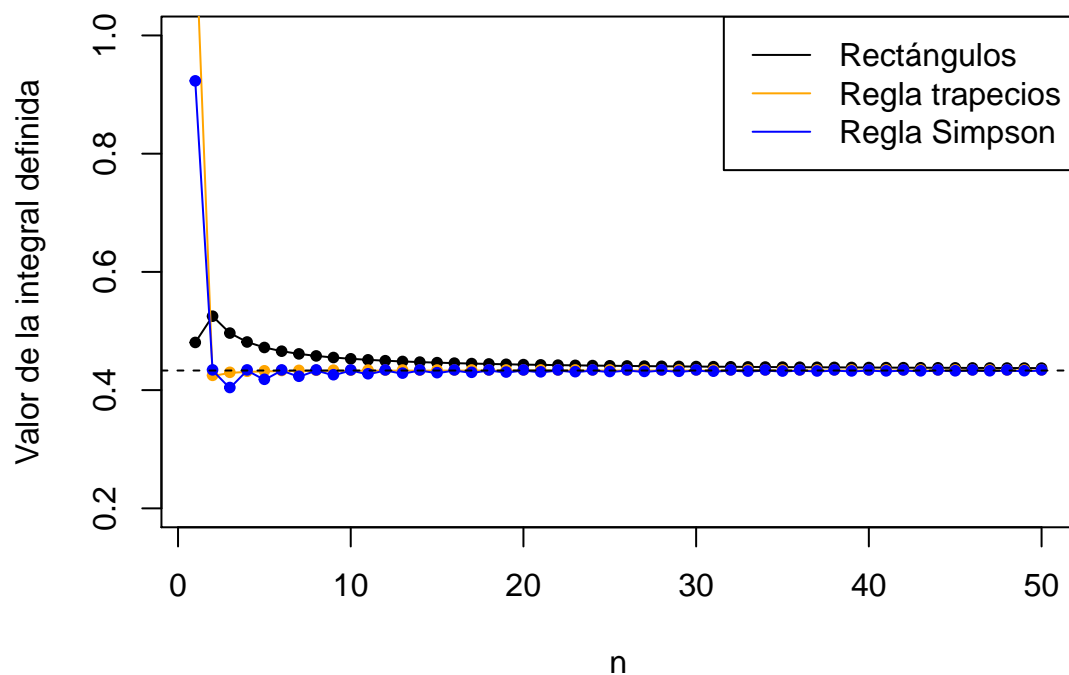
```

De hecho, gráficamente se ve que convergen mucho antes de que se llegue a las 100 particiones



La línea discontinua representa el valor exacto de la integral. Los puntos correspondientes a la suma superior e inferior realmente se acaban solapando, pero para que se apreciase en el gráfico los he separado un poco.

Solo por curiosidad también puede verse cómo funciona cada método de integración numérica. Para compararlos adecuadamente, en el caso de la aproximación utilizando rectángulos se utiliza como estimación el promedio de la suma superior e inferior.



Si hacemos ‘zoom’ se ve que la regla de los trapecios y la regla de Simpson son mejores aproximaciones. Sin embargo, por algún motivo que no entiendo bien, la regla de Simpson produce un comportamiento extraño cuando el número de particiones es impar. Ofrece estimaciones peores.

