Entrega Simulacion

Arturo Gonzalez Moya

07 abril, 2021

Contents

1 Estimar π mediante el método de Monte Carlo a través del ejemplo (explicado en clase) de las gotas de lluvia que caen dentro del círculo tras realizar el experimento $N = 10^4$ veces. Dar el intervalo de confianza con un nivel de confianza del 97 %. Analizar los resultados.

El método de las gotas de lluvia consiste en considerar un círculo de radio unidad circunscrito en un cuadrado y generar puntos y ver cuantos han caido dentro del círculo y cuanto fuera. Estos puntos vienen de la generación aleatoria de una variable uniforme entre -1 y 1.

```
N = 10^4
accept = c()

set.seed(123)
sample.x <- runif(N,-1,1)
set.seed(1234)
sample.y <- runif(N,-1,1)
```

Consideramos ahora la función de la circunferencia y vemos cuales de los puntos generados han caido dentro del circulo.

```
F = function(x,y){x^2+y^2}

for(i in 1:length(sample.x)){
  if(F(sample.x[i],sample.y[i]) <= 1){
    accept[i] = 1
  }
  else if(F(sample.x[i],sample.y[i]) > 1){
    accept[i] = 0
  }
}
```

Para calcular la estimación de π, hemos de utilizar la siguiente formula.

$$\hat{\pi} = 4 * \frac{puntos\_dentro\_del\_circulo}{puntos\_totales\_generados}$$

```
Zhat = 4*sum(accept)/N

phat = sum(accept)/N
```

Vemos que la estimación de π que obtenemos mediante el método de las gotas de lluvia es de 3.134 y la probabilidad estimada es de 0.7835. Pasemos a observar cuál es el intervalo de confianza al 97%.

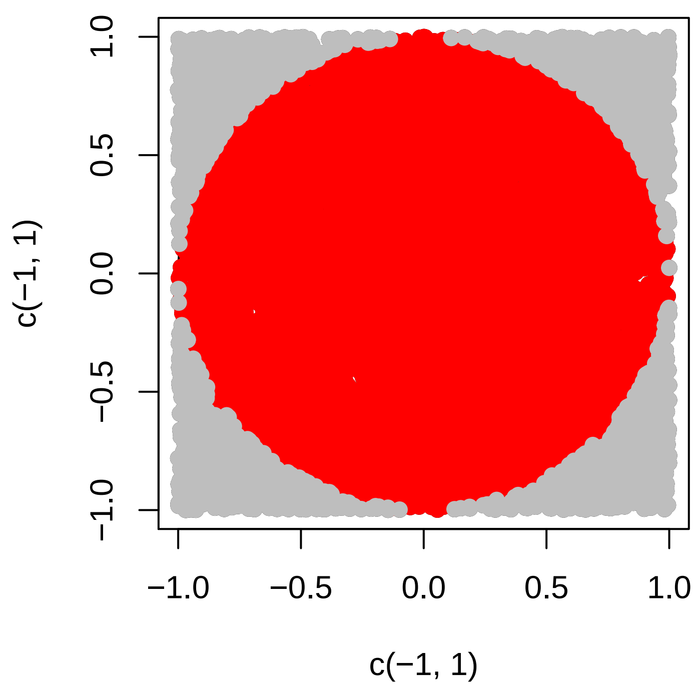
```
## Intervalo 97%

x = qnorm(1-(1-0.97)/2)

CI = c(phat - x*sqrt(phat*(1-phat)/N),phat + x*sqrt(phat*(1-phat)/N))
CI
```

```
## [1] 0.7745623 0.7924377
```

Vemos que el intervalo de confianza al 97% para la probabilidad es de (0.7745623, 0.7924377), por lo tanto, el intervalo de confianza para π sería (3.0982492, 3.1697508).

Por último, veamos la representació gráfica de este método de las gotas de agua.

```r
plot(c(-1, 1), c(-1, 1), type = "n")

radius <- 1
theta <- seq(0, 2 * pi, length = 200)

lines(x = radius * cos(theta), y = radius * sin(theta))

points(sample.x,sample.y)

T = data.frame(sample.x,sample.y, accept = factor(accept, levels= c('Yes','No')))

sample.accept.x = T[accept == "1", 1]
sample.noaccept.x = T[accept == "0", 1]

sample.accept.y = T[accept == "1", 2]
sample.noaccept.y = T[accept == "0", 2]

points(sample.accept.x,sample.accept.y,col="red",pch = 19)
points(sample.noaccept.x,sample.noaccept.y,col="grey",pch = 19)
```



2 Simúlese una variable aleatoria X ~ Beta(3, 6) mediante otra variable beta Y ~ Beta(2.7, 5.6) como envolvente.

Lo que haremos para resolver este ejercicio será, primero, generar 10^6 muestras de una variable Beta(3, 6).

```
x = seq(0,1,by=0.001)
set.seed(123)
sample.x = rbeta(10^6,2.7,5.6)
```

Ahora necesitamos calcular la cota de la función $f(x)/g(x)$ donde $f(x)$ es la función de distribución de una variable Beta(3, 6) y $g(x)$ es la función de distribución de una variable Beta(2.7, 5.6) que será nuestra envolvente. Esta cota la hemos calculado mediante wolframalpha y hemos obtenido que es de 1.067. Pasamos a aplicar el método general de aceptación-rechazo.

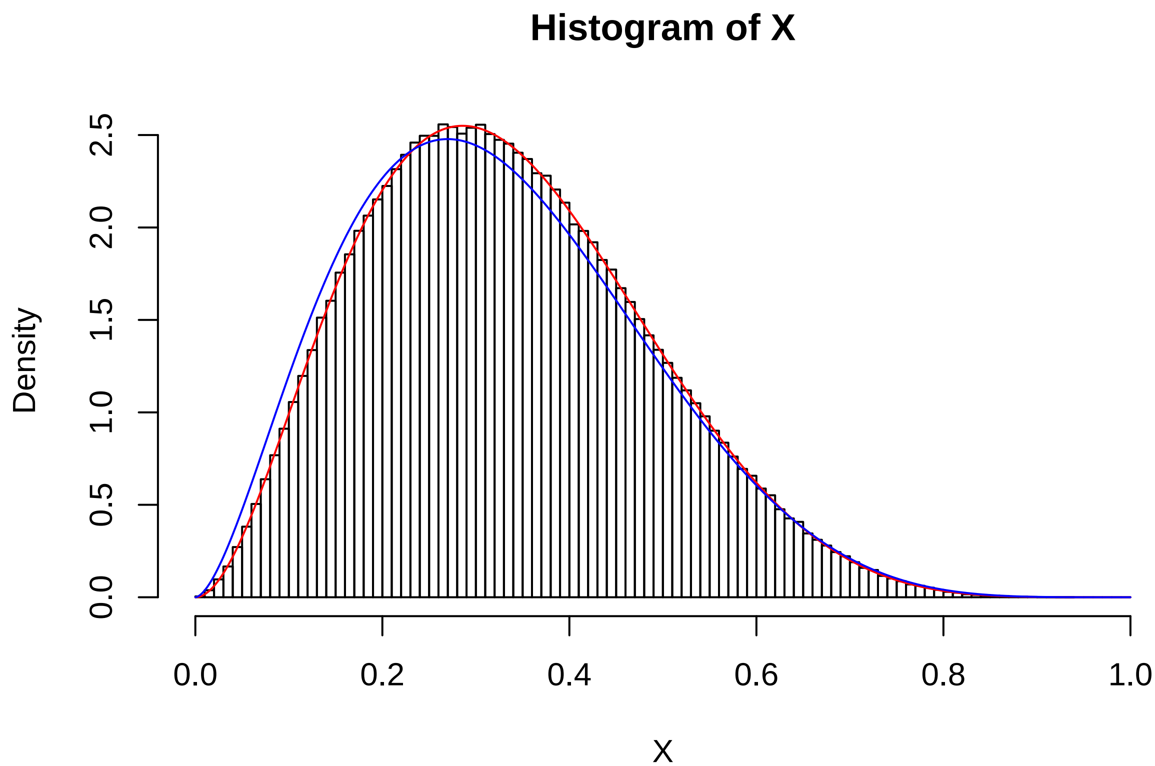
```
accept = c()

for(i in 1:length(sample.x)){
  U = runif(1, 0, 1)
  if(dbeta(sample.x[i], 2.7, 5.6)*(1.067)*U <= dbeta(sample.x[i], 3, 6)) {
    accept[i] = 'Yes'
  }
  else if (dbeta(sample.x[i], 2.7, 5.6)*(1.067)*U > dbeta(sample.x[i], 3, 6)){
    accept[i] = 'No'
  }
}

T = data.frame(sample.x, accept = factor(accept, levels= c('Yes','No')))

summary(T)
```

```
##     sample.x         accept
##  Min.   :0.002003   Yes:936995
##  1st Qu.:0.208126   No : 63005
##  Median :0.310548
##  Mean   :0.325242
##  3rd Qu.:0.427796
##  Max.   :0.952122
```

Podemos ver que de las 10^6 muestras generadas, aceptamos 936995 y rechazamos 63005. Esto se debe a que las funciones de las variables aleatorias X e Y no son muy diferentes.

Hagamos una representación gráfica del resultado.

```
hist(T[,1][T$accept=='Yes'], breaks = seq(0,1,0.01), freq = FALSE, main = 'Histogram of X', xlab = 'X')
lines(x, dbeta(x,3,6), col = "red")
lines(x, dbeta(x,2.7,5.6), col = "blue")
```



En este gráfico podemos ver la función de densidad de la variable Beta(2.7, 5.6) (que es la linea azul), la función de densidad de la variable Beta(3, 6) (que es la linea roja) y lo que sería nuestra aproximación de la variable Beta(3, 6) (que corresponde con las barras del histograma). Podemos ver que esta aproximación es bastante exacta.

3 Mediante el método Hit and Miss determínese el valor de la siguiente integral:

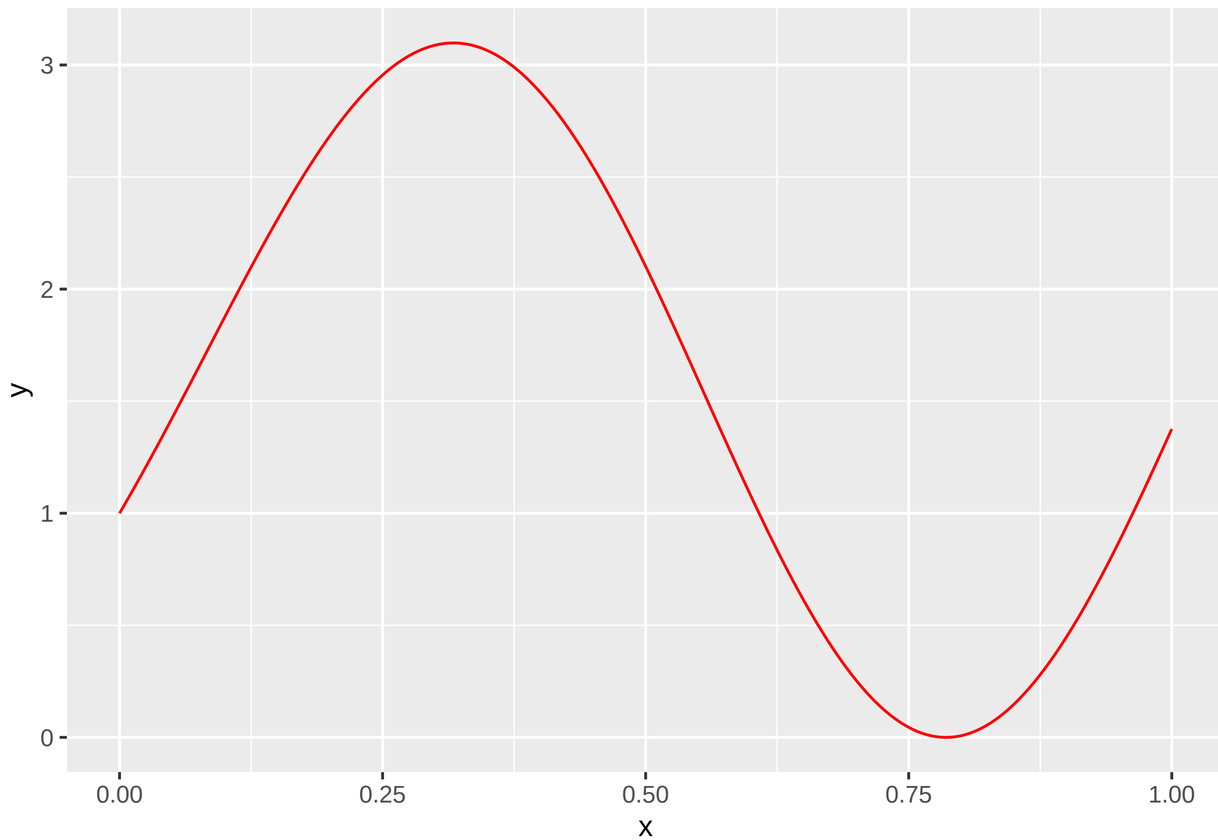
$$\int_0^1 (\cos(2x) + \sin(4x))^2 dx$$

Para utilizar el método Hit and Miss, hemos de ver que nuestra integral se encuentra en un intervalo acotado (que lo está ya que se encuentra entre 0 y 1), hemos de ver que la función que se integra es positiva en el dominio de integración (que lo es ya que está elevada al cuadrado) y hemos de calcular el máximo de la función $(\cos(2x) + \sin(4x))^2$ en ese intervalo. Utilizando la pagina web de wolframalpha, se ha obtenido que el valor máximo de la función en el intervalo $[0, 1]$ es, aproximadamente, de 3.09821.

Ahora podemos pasar al método Hit and Miss. Lo primero que haremos será dibujar nuestra función.

```
x = seq(0,1,by=0.001)
f = function(x){(cos(2*x)+ sin(4*x))^2}
y = f(x)
df <- data.frame(x,y)
```

```r
ggplot(df, aes(x))+
  geom_line(aes(y=y), colour="red")
```



Ahora generaremos 10^6 valores de una distribución uniforme $[0, 1]$ y vamos generando valores con una distribución uniforme $[0, 3.09821]$ formando pares *(x,u)* donde x viene de la uniforme $[0, 1]$ y u viene de la uniforme $[0, 3.09821]$. Entonces vemos cuales de estos pares estan debajo de la función que se integra y el valor de la integral sería el siguiente:

$$\hat{I} = 3.09821 * (1 - 0) * \frac{puntos\_debajo\_de\_la\_curva}{puntos\_totales\_generados}$$

```r
sample.x = runif(10^6,0,1)
accept = c()
sample.accept = c()

for(i in 1:length(sample.x)){
  U = runif(1, 0, 1)
  if(dunif(sample.x[i], 0, 1)*(3.09821)*U <= f(sample.x[i])) {
    accept[i] = 'Yes'
    sample.accept[i] = 1
  }
  else if(dunif(sample.x[i],0,1)*(3.09821)*U > f(sample.x[i])) {
    accept[i] = 'No'
    sample.accept[i] = 0
  }
```

```
}

T = data.frame(sample.x, accept = factor(accept, levels= c('Yes','No')), sample.accept)

phat = sum(sample.accept)/10^6
I = (3.09821)*(1-(0))*phat
```

El valor estimado de la integral es de 1.5575631. Calculemos ahora el valor real de esta integral.

```
Ireal = integrate(f,0,1)
```

Vemos que el valor real de esta integral es de 1.558277 con un error absoluto menor a $1.7 * 10^{-14}$. Por ultimo, calculemos el error cometido con la aproximación.

```
SE = (3.09821)*sqrt(phat*(1-phat))/sqrt(10^6)
```

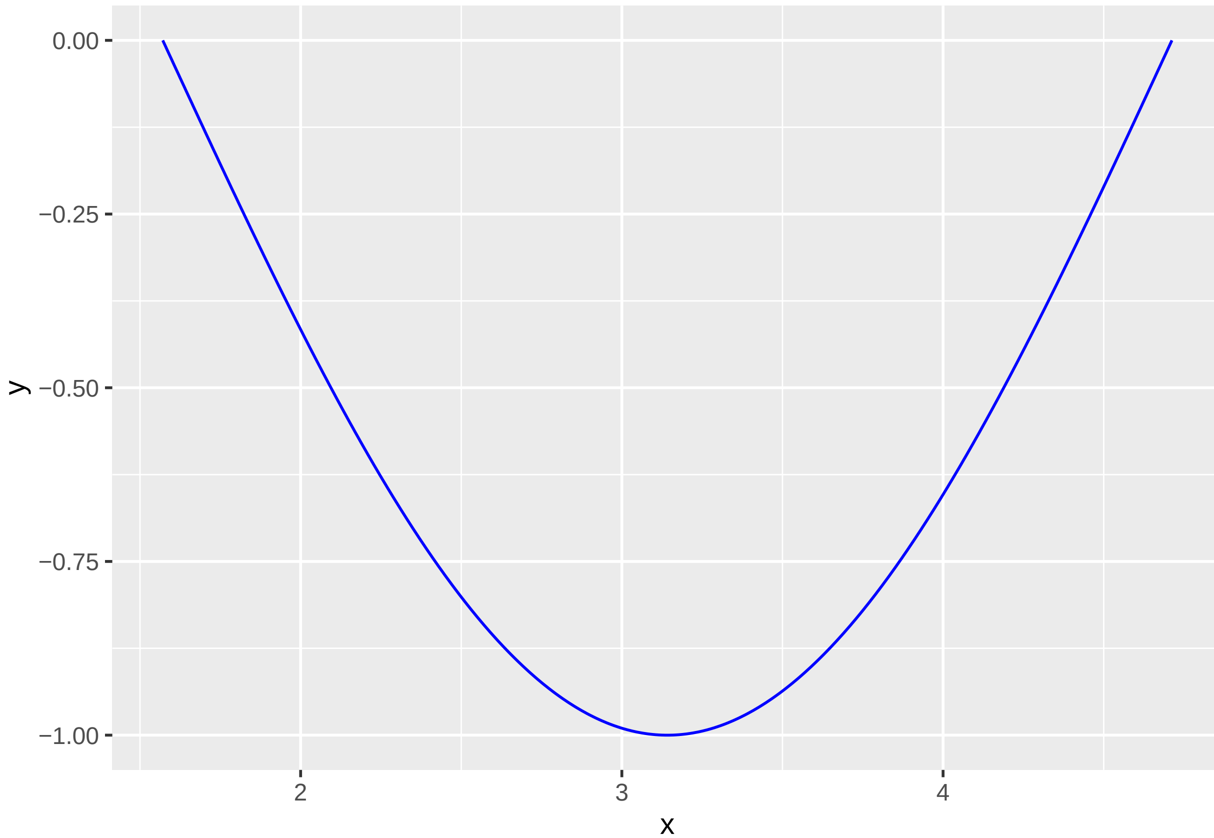
Vemos que el error es de 0.0015491, que es un error muy pequeño.

4 Dada la función $f(x) = \cos(x)$

4.1 Dibujar la gráfica entre $\pi/2$ y $3\pi/2$.

Para dibujar la función lo hacemos igual que en el ejercicio anterior, 1000 puntos equidistantes en intervlo $[\pi/2, 3\pi/2]$ y dibujamos la función con `ggplot`.

```
x = seq(pi/2,3*pi/2,by=(3*pi/2-pi/2)/1000)
f = function(x){cos(x)}
y = f(x)
df <- data.frame(x,y)

ggplot(df, aes(x))+
  geom_line(aes(y=y), colour="blue")
```



4.2 Obtener mediante alguna técnica de simulación de Monte Carlo el valor de la siguiente integral:

$$\int_{\pi/2}^{3\pi/2} \cos(x)dx$$

Lo que haremos será resolver esta ecuación por el método general de integración Monte Carlo utilizando la función uniforme, es decir:

$$\int_{\pi/2}^{3\pi/2} \cos(x)dx = \int_{\pi/2}^{3\pi/2} (3\pi/2 - \pi/2)\cos(x) f_U(x)dx$$

siendo $G(x) = (3\pi/2 - \pi/2)\cos(x)$ y $f_U(x)$ es la función de densidad de la uniforme en $[\pi/2, 3\pi/2]$. Lo que hacemos ahora es simular 10^6 muestras de la uniforme $U[\pi/2, 3\pi/2]$ y la aproximación del valor de la integral se calcula de la siguiente forma:

$$G(U)_{10^6} = \sum_{i=1}^{10^6} \frac{G(U_i)}{N}$$

Utilizaremos la siguiente función para el cálculo de la integral

```
integral <- function(N,a,b,f){
  set.seed(123)
```

```
  X <- runif(N, a, b)
  Y <- (b-a)*f(X)
  Int <- sum(Y)/N
  return(Int)
}
```

Pasemos a calcular la integral mediante una aproximación y su valor real.

```
g <- function(x){cos(x)}

I <- integral(10^6, pi/2,3*pi/2, g)
Ireal = integrate(g,pi/2,3*pi/2)
```

Observamos que el valor de la integral aproximandola es -1.9989661 y el valor real de la integral es -2, por lo que podemos decir que la aproximación es bastante buena.

4.3 ¿Podríamos integrar mediante el método de Hit and Miss? Razona la respuesta.

Para ver si podemos integrar mediante el método de Hit and Miss, hemos de ver si se satisfacen las hipótesis. Primero se ha de cumplir que el intervalo de integración sea acotado que en este caso se cumple ya que el intervalo es $[\pi/2, 3\pi/2]$. Lo siguiente que hemos de ver es que esta función en el intervalo de integración es positiva. Esto no ocurre ya que por ejemplo $\cos(\pi) = -1$. Por lo tanto en un principio no podría aplicarse el método de Hit and Miss.

Para solucionar este problema (ya que al ser una función continua en un intervalo acotado, esta tiene un máximo, que es la hipótesis que faltó comprobar antes), lo que podemos hacer es integrar la función $-\cos(x)$ en el mismo intervalo. Esta nueva función es positiva en todo el intervalo y satisface las demás propiedades, entoces podemos aplicar el método Hit and Miss, lo único que al resultado de la estimación habría que cambiarle el signo. Veamoslo.

```
f <- function(x){-cos(x)}
set.seed(123)
sample.x = runif(10^6,pi/2,3*pi/2)
accept = c()
sample.accept = c()

for(i in 1:length(sample.x)){
  a = runif(1,0,1)
  if(a <= f(sample.x[i])) {
    accept[i] = 'Yes'
    sample.accept[i] = 1
  }
  else if(a  > f(sample.x[i])) {
    accept[i] = 'No'
    sample.accept[i] = 0
  }
}

T = data.frame(sample.x, accept = factor(accept, levels= c('Yes','No')), sample.accept)

phat = sum(sample.accept)/10^6
I = (3*pi/2-(pi/2))*phat
```

Hemos obtenido una aproximación de 1.9997525, que si le cambiamos el signo es muy parecido a -2 que es el valor real de la integral.

Otra opción que podemos hacer para aplicar el método de Hit and Miss sobre la integral del coseno es, ya que la función es periódica, cambiar el intervalo de integración a por ejemplo $[-\pi/2, \pi, 2]$ y cambiarle el signo al resultado.

5 Determínese cual de las siguientes descomposiciones sería más adecuada para el cálculo de la integral (mediante el método de integración por importancia):

$$\int_0^1 2\pi\sqrt{2-2x^4}dx$$

5.1 $G(x) = \frac{2\pi\sqrt{2-2x^4}}{\frac{5(2-2x^4)}{8}}$ y $f(x) = \frac{5(2-2x^4)}{8}$

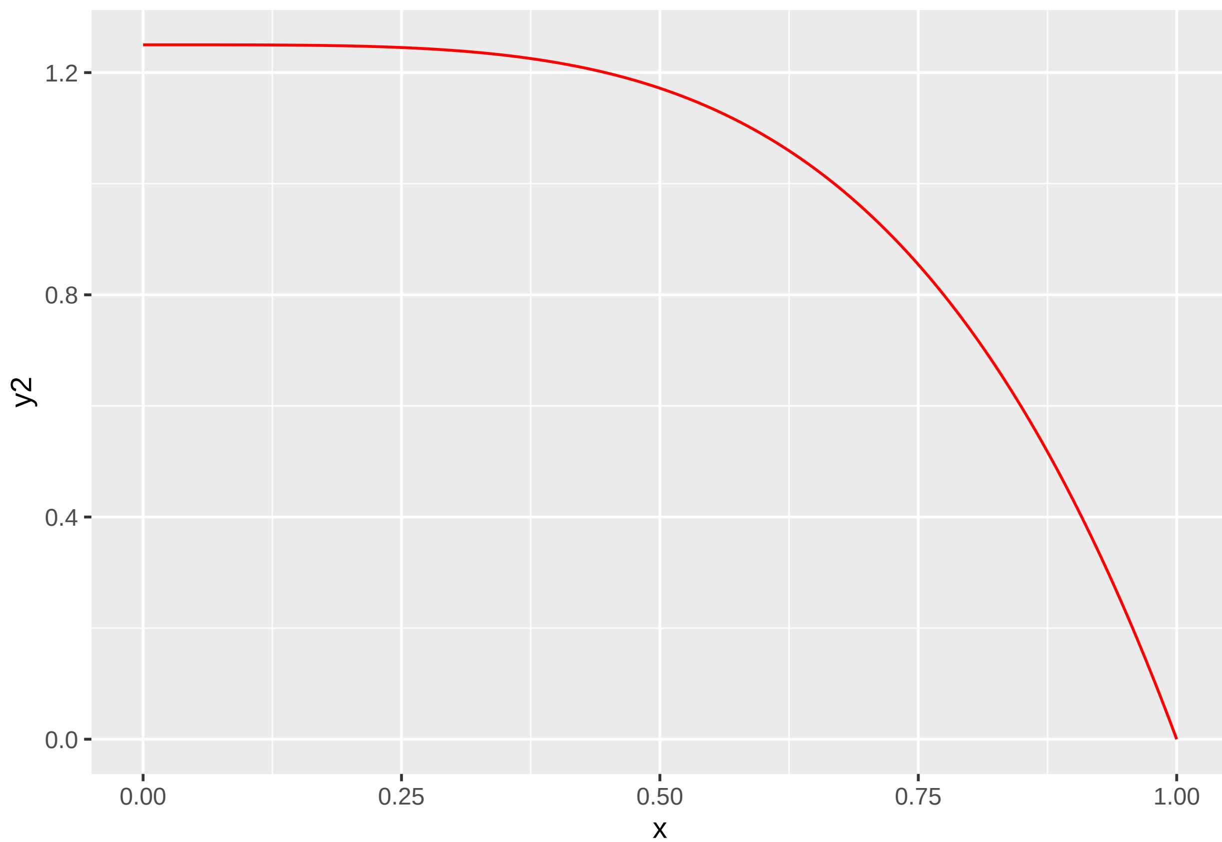
Comenzamos con calcular el valor aproximado de la integral con las funciones $G(x)$ y $f(x)$ que nos indican en este apartado. Primero veremos que $f(x)$ es una función de densidad. Para ello calcularemos la integral entre 0 y 1.

```
f <- function(x){5*(2-2*x^4)/8}

Ireal = integrate(f,0,1)
Ireal
```

```
## 1 with absolute error < 1.1e-14
```

El valor de la integral es 1, por lo que es función de densidad. Vamos a dibujar esta función $f(x)$ para ver como es.

```
x = seq(0,1,by=0.001)

y2 = f(x)
df <- data.frame(x,y2)

ggplot(df, aes(x))+
  geom_line(aes(y=y2), colour="red")
```



En el intervalo $[0, 1]$ esta función es decreciente, por lo que su máximo se toma en 0 y tiene un valor de 1.25. Además, esta función es siempre positiva en el intervalo de integración. Pasamos a calcular el valor de la integral. Para ello utilizaremos 10^6 muestras de una distribución uniforme en $[0, 1]$.

```
set.seed(123)
sample.x = runif(10^6,0,1)
accept = c()
sample.accept = c()

for(i in 1:length(sample.x)){
  U = runif(1, 0, 1)
  if(dunif(sample.x[i], 0, 1)*(5/4)*U <= f(sample.x[i])) {
    accept[i] = 'Yes'
    sample.accept[i] = sample.x[i]
  }
  else if(dunif(sample.x[i],0,1)*(5/4)*U > f(sample.x[i])) {
    accept[i] = 'No'
    sample.accept[i] = 0
  }
}

T = data.frame(sample.x, accept = factor(accept, levels= c('Yes','No')),sample.accept)

Taccept = T[accept == 'Yes',]

G = function(x){(2*pi*sqrt(2-2*x^4))/((5/8)*(2-2*x^4))}
```

```r
Naccept = length(which(T$accept == 'Yes'))

I = sum(G(Taccept$sample.accept))/Naccept
```

El valor aproximado de la integral que hemos obtenido es 7.7662549, mientras que el valor real de la integral es el siguiente.

```r
F = function(x){(2*pi*sqrt(2-2*x^4))}
Ireal = integrate(F,0,1)
Ireal
```

```
## 7.76633 with absolute error < 0.00049
```

```r
Ir <- 7.76633
```

Vemos que la aproximación ha sido muy cercana. Por último, calcularemos la varianza obtenida con la función $f(x)$ escogida.

```r
S = sqrt(sum((G(Taccept$sample.accept)-sum(G(Taccept$sample.accept))/Naccept)^2)/(Naccept - 1))
S
```

```
## [1] 1.674111
```

La varianza obtenida es de 1.6741106. Esta varianza servirá para saber la función $f(x)$ que hemos de seleccionar, ya que el método de selección por importacia nos dice de elegir aquella con varianza menor.

5.2 $G(x) = \frac{2\pi\sqrt{2-2x^4}}{\frac{3(2-2x^2)}{4}}$ y $f(x) = \frac{3(2-2x^2)}{4}$

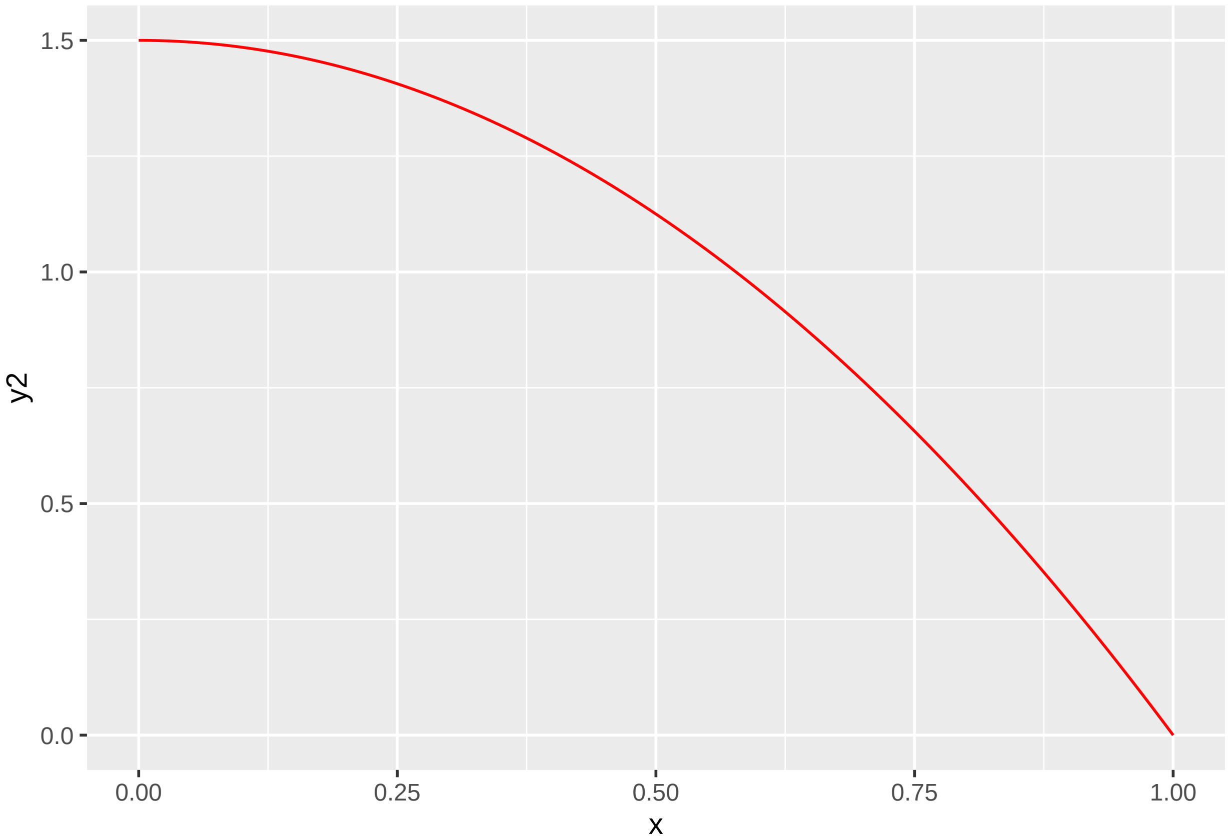
Como antes, para poder calcular el valor de la integral que nos piden utilizando esta función $f(x)$ lo primero que hemos de hacer es ver si es una función de densidad en el intervalo $[0, 1]$.

```r
f <- function(x){(3/4)*(2-2*x^2)}

Ireal = integrate(f,0,1)
Ireal
```

```
## 1 with absolute error < 1.1e-14
```

Vemos que la integral vale 1, por lo que es función de densidad. Pasamos a dibujar esta función $f(x)$.

```r
x = seq(0,1,by=0.001)

y2 = f(x)
df <- data.frame(x,y2)

ggplot(df, aes(x))+
  geom_line(aes(y=y2), colour="red")
```



Igual que en el apartado anterior, esta función es decreciente en el intervalo $[0,1]$, por lo que el máximo se da en 0 y su valor es 1.5. Además, esta función es siempre positiva en el intervalo de integración. Pasamos a calcular el valor de la integral y para ello utilizaremos 10^6 muestras de una distribución uniforme en $[0,1]$.

```
set.seed(123)
sample.x = runif(10^6,0,1)
accept = c()
sample.accept = c()

for(i in 1:length(sample.x)){
  U = runif(1, 0, 1)
  if(dunif(sample.x[i], 0, 1)*(3/2)*U <= f(sample.x[i])) {
    accept[i] = 'Yes'
    sample.accept[i] = sample.x[i]
  }
  else if(dunif(sample.x[i],0,1)*(3/2)*U > f(sample.x[i])) {
    accept[i] = 'No'
    sample.accept[i] = 0
  }
}

T = data.frame(sample.x, accept = factor(accept, levels= c('Yes','No')),sample.accept)

Taccept = T[accept == 'Yes',]

G = function(x){(2*pi*sqrt(2-2*x^4))/((3/4)*(2-2*x^2))}
```

```r
Naccept = length(which(T$accept == 'Yes'))

I = sum(G(Taccept$sample.accept))/Naccept
```

El valor aproximado de la integral que hemos obtenido es 7.7625077 mientras que el valor real de esta es 7.76633, por lo que podemos decir que la aproximación ha sido buena. Veamos ahora cual es la varianza.

```r
S = sqrt(sum((G(Taccept$sample.accept)-sum(G(Taccept$sample.accept))/Naccept)^2)/(Naccept - 1))
S
```

```
## [1] 3.112682
```

En este caso, la varianza es mayor que en el apartado anterior, por lo que descartamos la función $f(x)$ elegida en este apartado.

5.3 $G(x) = \frac{2\pi\sqrt{2-2x^4}}{\frac{5(1-2x^4)}{3}}$ y $f(x) = \frac{5(1-2x^4)}{3}$

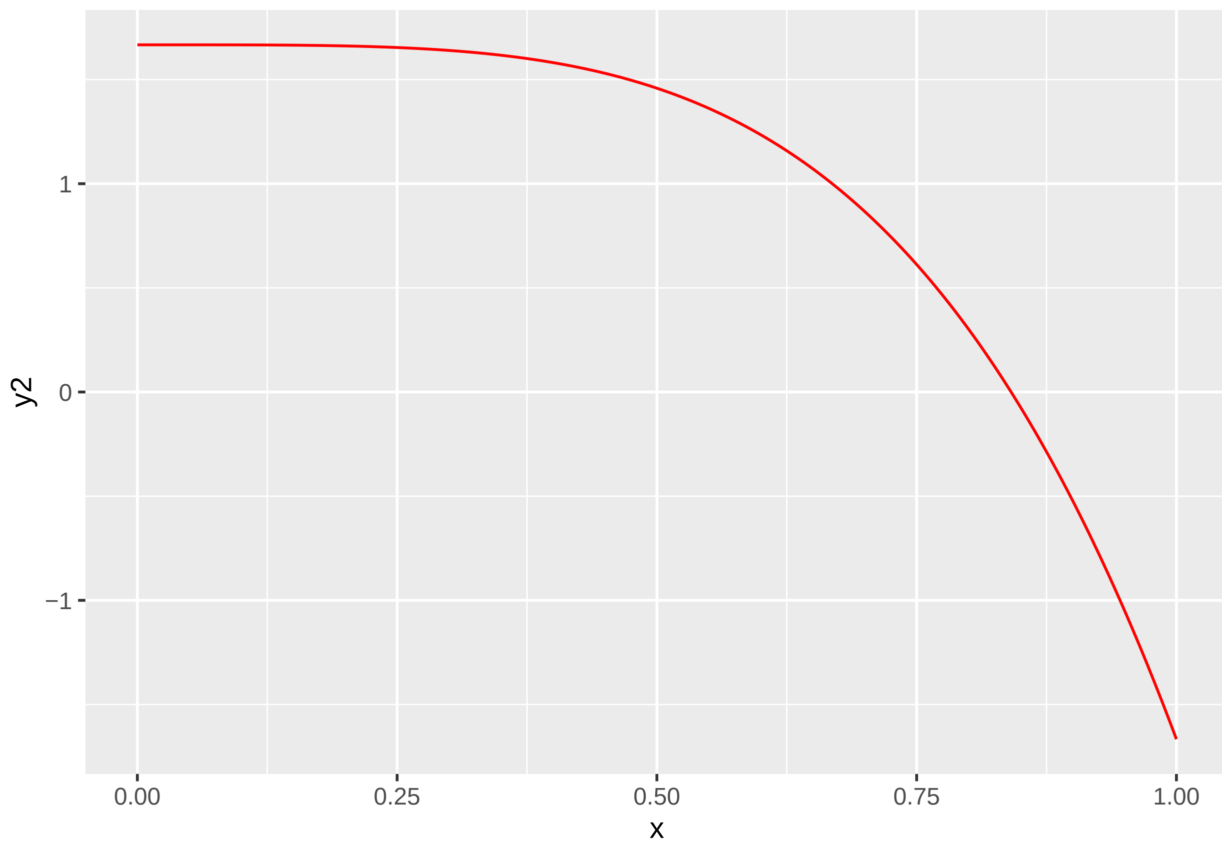
Por último, veamos que ocurre con la función $f(x)$ que nos dan en este apartado. Veamos primero si es función de densidad.

```r
f <- function(x){(5/3)*(1-2*x^4)}

Ireal = integrate(f,0,1)
Ireal
```

```
## 1 with absolute error < 1.4e-14
```

Vemos que la integral vale 1, por lo que es función de densidad. Pasamos a dibujar esta función $f(x)$.

```r
x = seq(0,1,by=0.001)

y2 = f(x)
df <- data.frame(x,y2)

ggplot(df, aes(x))+
  geom_line(aes(y=y2), colour="red")
```



Igual que en el apartado anterior, esta función es decreciente en el intervalo $[0, 1]$, por lo que el máximo se da en 0 y su valor es 1.6666667. Además, esta función es siempre positiva en el intervalo de integración. Pasamos a calcular el valor de la integral y para ello utilizaremos 10^6 muestras de una distribución uniforme en $[0, 1]$.

```
# REVISAR ESTO
set.seed(123)
sample.x = runif(10^6,0,1)
accept = c()
sample.accept = c()

for(i in 1:length(sample.x)){
  U = runif(1, 0, 1)
  if(dunif(sample.x[i], 0, 1)*5/3*U <= f(sample.x[i])) {
    accept[i] = 'Yes'
    sample.accept[i] = sample.x[i]
  }
  else if(dunif(sample.x[i],0,1)*5/3*U > f(sample.x[i])) {
    accept[i] = 'No'
    sample.accept[i] = 0
  }
}

T = data.frame(sample.x, accept = factor(accept, levels= c('Yes','No')),sample.accept)

Taccept = T[accept == 'Yes',]
```

```r
G = function(x){(2*pi*sqrt(2-2*x^4))/((5/3)*(1-2*x^4))}
Naccept = length(which(T$accept == 'Yes'))

I = sum(G(Taccept$sample.accept))/Naccept
```

El valor aproximado de la integral que hemos obtenido es 6.2919445 mientras que el valor real de esta es 7.76633. Esta aproximación de la integral no es muy buena. Veamos cual es su varianza.

```r
S = sqrt(sum((G(Taccept$sample.accept)-sum(G(Taccept$sample.accept))/Naccept)^2)/(Naccept - 1))
S
```

```
## [1] 5.612324
```

La varianza es la más grande de todas las obtenidas, por lo que esta $f(x)$ no es la mejor.

Concluimos diciendo que entre las 3 opciones de funciones $f(x)$ que hemos tenido, la primera (que era $f(x) = \frac{5(2-2x^4)}{8}$) es la que menor varianza tiene y es la que debemos escoger.