TP3 Practice - Monte Carlo Integration

Evan Voyles

21 Avril, 2022

To estimate the integral, just sample along X uniformly along the interval, calculate the area of the rectangle whose height is X_i and whose width if $b - a$. Then averae a ton of these samples.

Let's try computing

$$h(x) = (\cos(50x) + \sin(20x))^2$$

via Monte Carlo integration.

```
set.seed(123)

library(purrr)

id <- function(x) { x }

# accept a function and bounds.
#
# @param f A real function f : R -> R to be integrated by Monte Carlo integration
#          over a uniform distribution
mc_integrate_unif <- function(h = function(x) { id(x) }, a = 0, b = 1, n = 1000) {

    # sample from a to b, n times
    draws <- runif(n, min = a, max = b)

    # Now go ahead and map the function over this.
    h_Xs <- map_dbl(draws, h)

    # Take the cumsum and then average it by dividing by n
    cumsum(h_Xs) / seq(n)
}

mc_integrate_norm <- function(h = function(x) { id(x) }, n = 1000) {

    draws <- rnorm(n)
    h_Xs <- map_dbl(draws, h)
    cumsum(h_Xs) / seq(n)
}

h_x <- function(x) { (cos(50 * x) + sin(20 * x))**2 }

# 2 dimensional version where we determine the x AND y bounds
mc_integrate_rec <- function(f = id, min_x, max_x, min_y, max_y, n = 1000) {

    x_width <- max_x - min_x
    y_width <- max_y - min_y
```

```
    # Randomly sample between x and y
    x_rand <- runif(n, min_x, max_x)
    y_rand <- runif(n, min_y, max_y)

    sum(map2_lgl(x_rand, y_rand, function(x, y) { x <= y })) / n
}

rnorm_bounded_single <- function(a, b, mean = 0, sd = 1) {
    while (TRUE) {
        x <- rnorm(1, mean, sd)
        if (x >= a & x <= b) {return(x)}
    }
}

rnorm_bounded <- function(n, a, b, mean = 0, sd = 1) {
    x <- vector("numeric", n)
    for (i in seq_len(n)) {
        x[i] = rnorm_bounded_single(a, b, mean, sd)
    }
    x
}

mc_integrate_norm_bounded <- function(h = id, a = 0, b = 1, n = 1000) {

    draws <- rnorm_bounded(n, a, b)
    h_Xs <- map_dbl(draws, h)
    cumsum(h_Xs) / seq(n)

}
```

Awesome, now let's move on to the TP.

Ecercice 1 Soit l'intégrale élémentaire

$$I = \int_0^2 x^2 dx$$

.

1. Dans cet exercice, I est facilement calculable. Calculer sa valeur.

$$I = \int_0^2 x^2 dx = \left[\frac{x^3}{3} + c\right]_0^2 = \frac{8}{3}$$

2. On souhaire estimer I à l'aide d'un échantillon de loi uniforme.
   (a) Ecrire I sous la forme $\mathbb{E}[(h(X))]$ où X suit une loi uniforme sur $[0, 2]$, et h est à expliciter.

$$E[h(X)] = \int_0^2 h(x) f(x) \mathrm{d}x,$$

où $f(x)$ est la densité de la loi uniforme, c'est-à-dire $f(x) = \frac{1}{2}$ si $0 \leq x \leq 2$, 0 sinon. Donc,

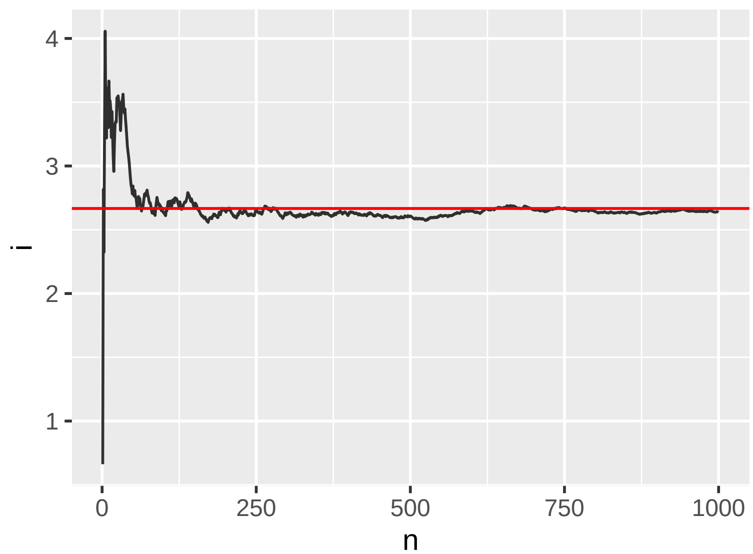
$$I = \int_0^2 x^2 dx = E[h(X)] = \int_0^2 h(x)\frac{1}{2} \implies h(x) = 2x^2$$

.

(b) En déduire une estimation $\hat{I}_n$ de I par la méthode de Monte Carlo pour un nombre n de simulations.

$$\hat{I}_n = \frac{1}{n}\sum_{i=1}^{n} h(X_i)$$

```
h_x <- function(x) { 2 * x ** 2 }

MCtraj <- function(n) { mc_integrate_unif(h_x, 0, 2, n) }

traj <- MCtraj(1000)

df <- tibble(n = seq(1000), i = traj)

df |>
    ggplot(aes(n, i)) +
    geom_line(alpha = 0.8) +
    geom_hline(yintercept = 8/3, col = "red")
```



Exercice 2 On cherche à estimer l'intégrale

$$I = \int_0^2 e^{-x^2} dx$$

1. On peut estimer I à partir d'un échantillon de loi uniforme.
    (a) Ecrire I sous la form $\mathbb{E}[h(U)]$ où U sui une loi uniform dont vous choisirez le support, est h est une fonction à expliciter.

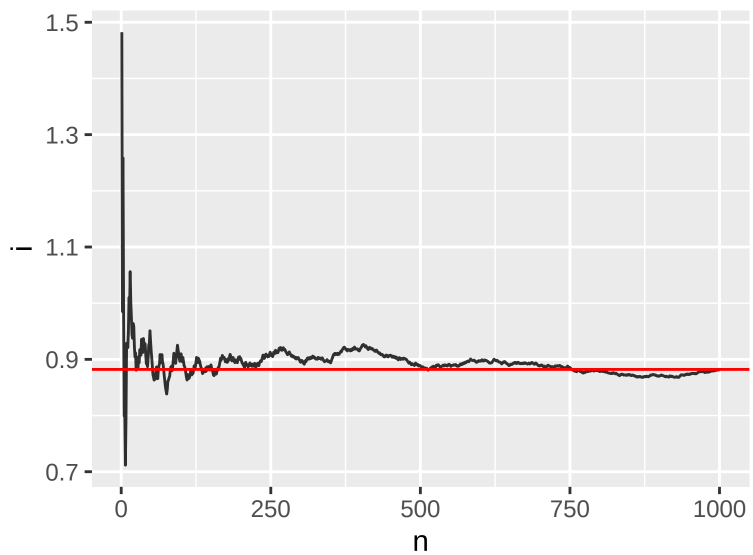
La valeur vraie de I peut être calculer à partir de la fonction d'erreur, `erf`, définie comme:

$$\text{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt,$$

qui est implémenté sur plusiers logiciels tels que Mathematica ou Matlab.

$$I = \frac{\sqrt{\pi}}{2}\text{erf}(2) = 0.882081390762422$$

```r
h_x <- function(x) { 2 * exp(- (x * x)) }

a <- 0
b <- 2
n <- 1000

MCtraj.unif <- function(n) { mc_integrate_unif(h_x, a, b, n) }


traj <- MCtraj.unif(n)

df <- tibble(n = seq(n), i = traj)

I_true <- 0.882081390762422


df |>
    ggplot(aes(n, i)) +
    geom_line(alpha = 0.8) +
    geom_hline(yintercept = I_true, col = "red")
```



2. On peut également estimer I à partir d'un échantillon de loi gaussienne.
    (a) Ecrire I sous forme $\mathbb{E}[(h_2(X)]$ où $X \sim \mathcal{N}(0,1)$ et h_2 est à expliciter.

$$\mathbb{E}_{\mathcal{N}}[h_2(X)] = \int_{-\infty}^{\infty} h_2(x) \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}} \tag{1}$$

o'u la densité de $\mathcal{N}$ est la gaussienne de la loi normale centrée réduite. On veut finir par calculer l'intégrale de e^{-x^2} entre 0 et 2 donc on postule

$$h(x) = \sqrt{2\pi} e^{\frac{-x^2}{2}}$$

```r
a <- 0
b <- 2
n <- 1000

h_x <- function(x) { sqrt(2 * pi) * exp(- (x * x) / 2 ) }
```

```r
MCtraj.norm <- function(n) { mc_integrate_norm_bounded(h_x, a, b, n) }

# I really have no clue how to procede since the domain of N(0, 1) is -\infty to +\infty
```

Exercice 3 L'objectif de l'exercice est de calculer l'intégrale

$$I = \int_{\mathbb{R}} \exp(-u^2) f_d(u) du,$$

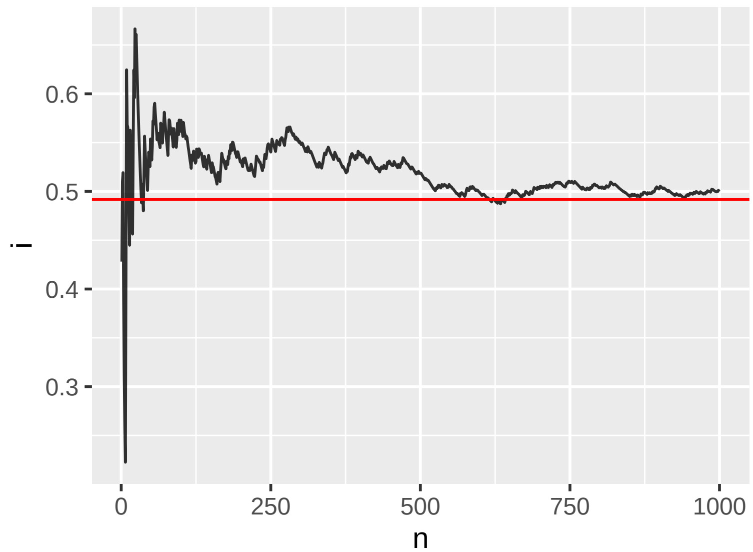
où f_d est la densité de la loi de Student $\mathcal{S}t(d)$ à $d > 1$ degrés de liberté.

1. **Approximation directe via la loi de Student**

```r
mc_integrate_student <- function(h = id, degrees_freedom = 2, n = 1000) {

    draws <- rt(n, degrees_freedom)
    h_Xs <- map_dbl(draws, h)
    cumsum(h_Xs) / seq(n)
}

h_x <- function(x) { exp(- (x * x))}

MCtraj.student <- function(n, d) { mc_integrate_student(h_x, d, n) }


mc_integrate_cauchy <- function(h = id, location = 0, scale = 1) {

    draws <- rcauchy(n, location, scale)
    h_Xs <- map_dbl(draws, h)
    cumsum(h_Xs) / seq(n)
}

# We want to calculate the integral of exp(-u^2) * f_d, where f_d is the students t. So exp(-u^2) * fd
# our E[h(x)], so h(x) * dcauchy = exp(-u^2) * fd. Thus, h(x) = (exp(-u^2) * fd) / dcauchy.
MCtraj.cauchy <- function(n, d) {
    mc_integrate_cauchy(h = function(x) {(exp(-x**2) * dt(x, d)) / dcauchy(x, 0, d)},
                        0,
                        d)
}
```

```r
t_cauchy <- function(n, d) {
    x <- MCtraj.cauchy(n, d)
    x[n]
}

t_student <- function(n, d) {
    x <- MCtraj.student(n, d)
    x[n]
}

# Cauchy Trajectory
traj <- MCtraj.cauchy(n, 2)
df <- tibble(n = seq(n), i = traj)
df |>
    ggplot(aes(n, i)) +
```

```
    geom_line(alpha = 0.8)+
    geom_hline(yintercept = 0.4916598, col = "red")
```




```
# Student Trajectory
traj <- MCtraj.student(n, 2)
df <- tibble(n = seq(n), i = traj)
df |>
    ggplot(aes(n, i)) +
    geom_line(alpha = 0.8) +
    geom_hline(yintercept = 0.4916598, col = "red")
```