TP2 Theorèmes limites

Evan Voyles

15 Avril, 2022

Le code source pour ce document est disponible ici

```
library(tibble)
library(ggplot2)
library(tidyverse)
set.seed(1234) # For reproducible results
```

Exercice 1 Illustration de la LFGN.

1. Simuler un échantillon de taille 1000 de variables aléatoires X_i de loi exponentielle de paramètre 2.

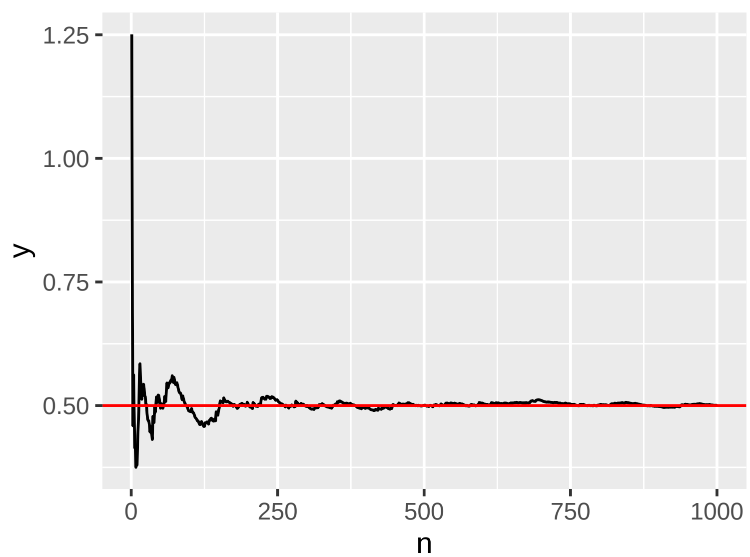
```
rate <- 2; n <- 1000

x <- rexp(n, rate)
```

2. En déduire les moyennes empiriques $\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$ pour $n = 1, \ldots, 1000$.

```
moyenne_empiriques <- cumsum(x) / seq(n)
```

3. Tracer une trajectoire de la moyenne empirique, i.e $n \mapsto \bar{X}_n$ pour $n = 1, \ldots, 1000$. Superposer la droite d'équation $y = 1/2$ (utiliser la fonction `abline`). D'où vient 1/2?

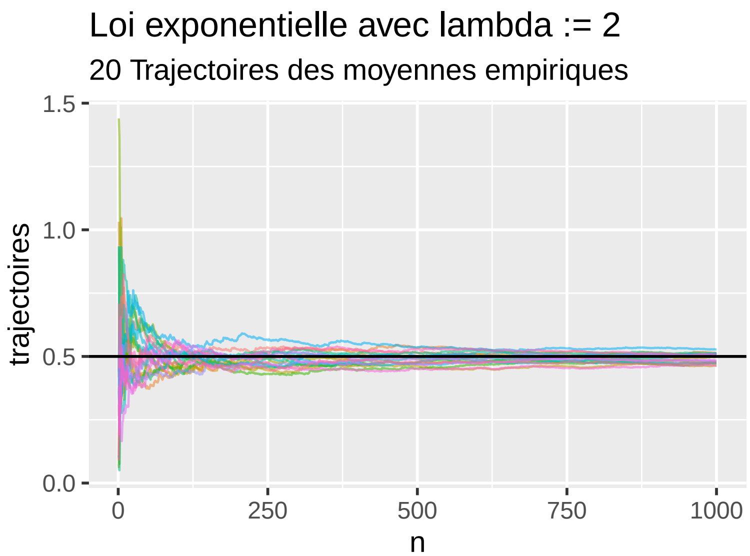
```
df <- tibble(n = seq(n), y = moyenne_empiriques)
df |> ggplot(aes(n, y)) + geom_line() + geom_hline(yintercept = 0.5, col = "red")
```



1/2 est l'espérance de notre loi exponentielle ($\frac{1}{\lambda}$)

4. Superposer 20 autres trajectoires de la moyenne empirique.

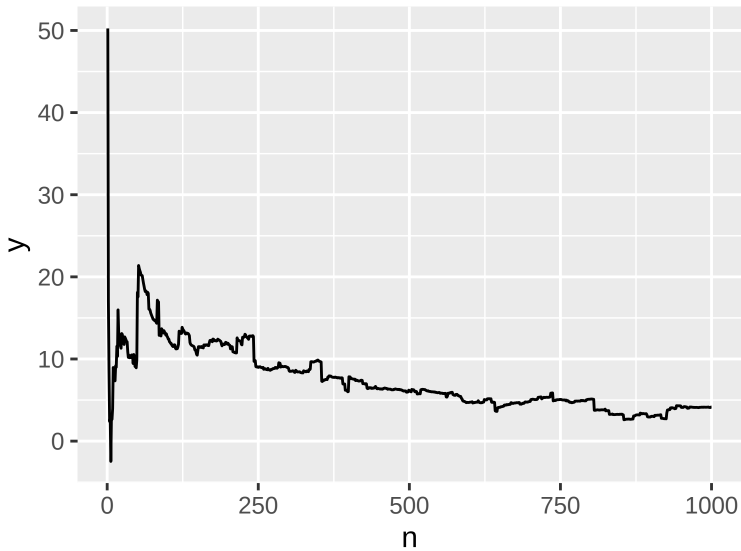
```r
sim <- function(n) { rexp(n, 2) }
# draw_traj est une fonction que j'ai définie, montrée en fin de document
draw_traj(sim,
          n_trials = 1000,
          n_traj = 20,
          title = "Loi exponentielle avec lambda := 2",
          subtitle = "20 Trajectoires des moyennes empiriques")
```



Tous les trajectoires tendent vers 0.5

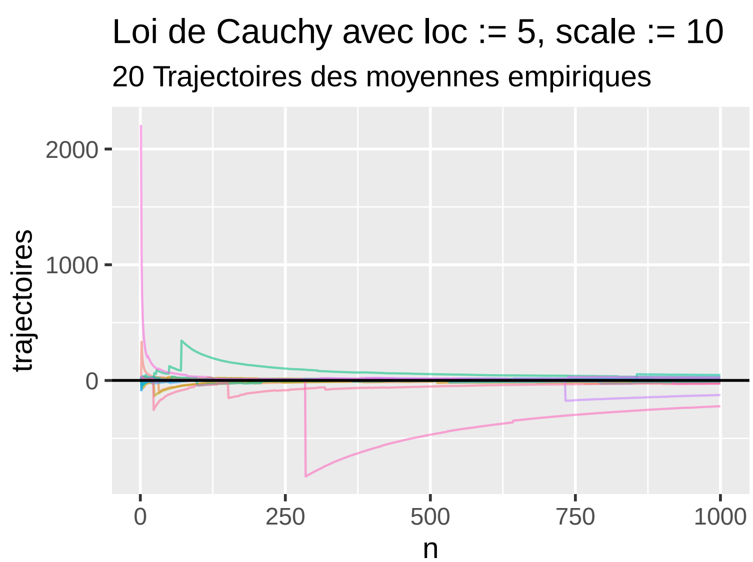
5. Reprendre les mêmes questions pour une loi de Cauchy. Que remarque-t-on?

```r
loc   <- 5
scale <- 10
n     <- 1000

cauch <- rcauchy(n, loc, scale)
moyenne_empiriques <- cumsum(cauch) / seq(n)

df.cauchy <- tibble(n = seq(n), y = moyenne_empiriques)
df.cauchy |> ggplot(aes(n, y)) + geom_line()
```



On suppose que lorsque $n \to \infty$, les moyennes empiriques tendront vers la "location" (endroit) de la loi de Cauchy.

```
sim <- function(n) { rcauchy(n, loc, scale) }
draw_traj(sim,
          n_trials = 1000,
          n_traj = 20,
          "Loi de Cauchy avec loc := 5, scale := 10",
          "20 Trajectoires des moyennes empiriques")
```

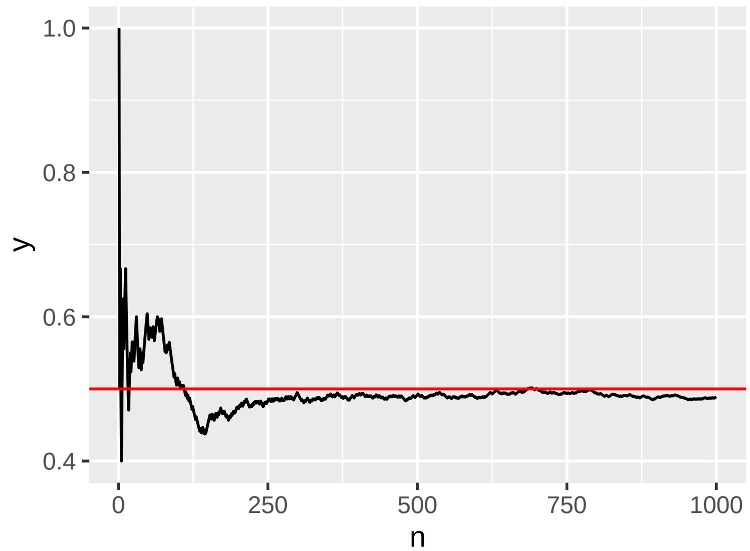


On observe la même tendance que les moyennes empiriques tendent vers l'endroit (5 dans ce cas). Cependant, on remarque que - grace au fait que la loi de Cauchy a des "queues" (tail) qui sont rélativement fort par rapport à des autres lois comme la loi normale - il est possible de tomber sur une quantité dont la valeur absolue est extrêment large. C'est pour cela qu'on peut observer des pics la ou la moyenne a été influencé par une valeur rélativement grande.

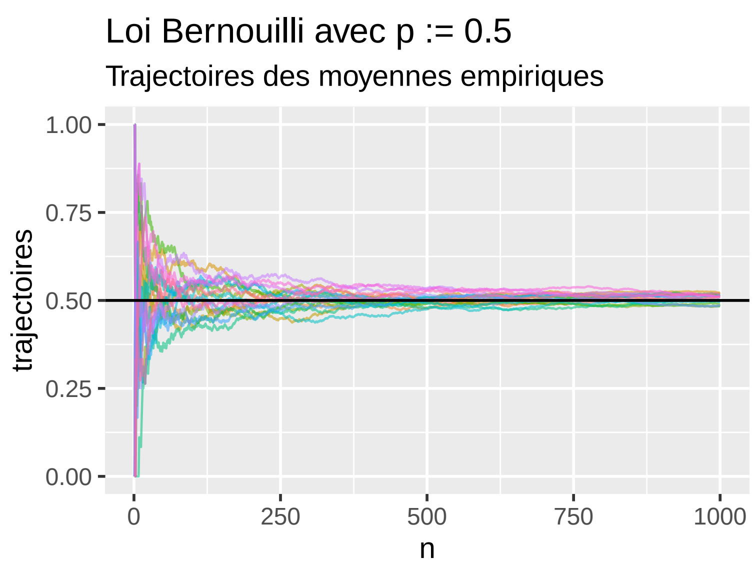
Exercice 2 Loi d'une variable aléatoire discrète.

1. Illustrer la LFGN lorsque les X_i sont i.i.d de loi de Bernouille $\mathcal{B}(p)$ avec $p = 0.5$

```
n <- 1000
p <- 0.5

bern <- rbinom(n, size = 1, prob = p)
moyenne_empiriques <- cumsum(bern) / seq(n)

df.bern <- tibble(n = seq(n), y = moyenne_empiriques)
df.bern |> ggplot(aes(n, y)) + geom_line() + geom_hline(yintercept = 0.5, col = "red")
```



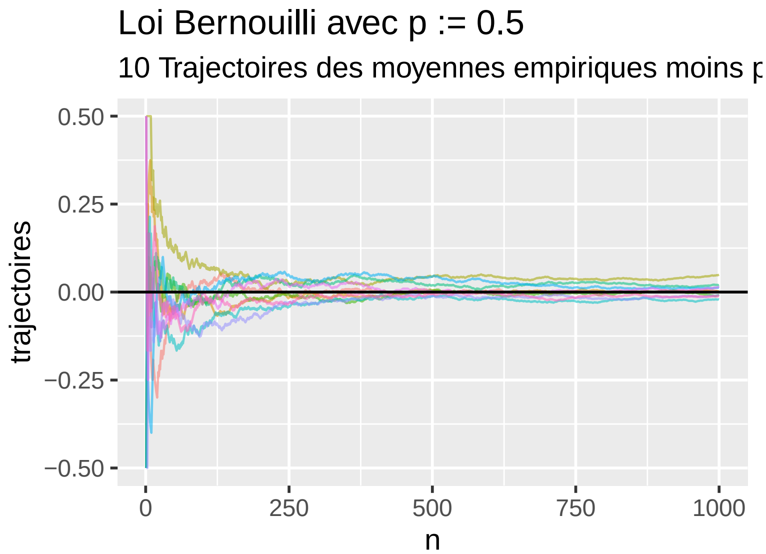

```
sim <- function(n) { rbinom(n, size = 1, prob = p) }
draw_traj(sim,
          n_trials = 1000,
          n_traj = 20,
          title = "Loi Bernouilli avec p := 0.5",
          subtitle = "Trajectoires des moyennes empiriques")
```



2. On s'intéresse maintenant à l'écart entre $\bar{X}_n$ et p pour étudier la vitesse de convergence.

(a) Représenter 10 trajectoires de $\bar{X}_n - p$.

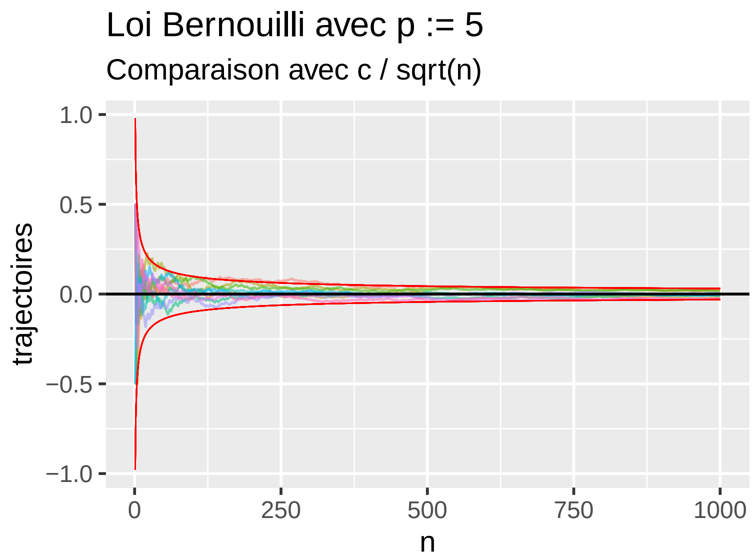
```
p <- 0.5

sim <- function(n) { rbinom(n, size = 1, prob = p) }
draw_traj_moins_p(sim, 1000, 10, p, "Loi Bernouilli avec p := 0.5",
                                    "10 Trajectoires des moyennes empiriques moins p")
```



(b) Superposer en rouge les courbes $n \mapsto 1.96\frac{\sqrt{p(1-p)}}{\sqrt{n}}$ et $n \mapsto -1.96\frac{\sqrt{p(1-p)}}{\sqrt{n}}$. A quoi correspond 1.96 et que signifie t'il?

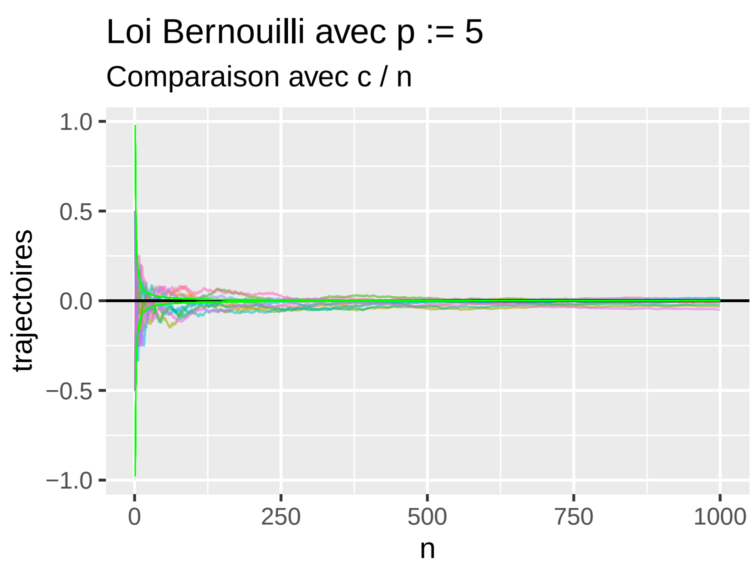
La valeur de 1.96 correspond à la valeur pour un intervalle de confiance de 95%. Très imprécisement, on suppose que 95% des moyennes empiriques seront entre les bornes de ces deux fonctions. On remarque que $\sqrt{p(1-q)}$ est l'écart type d'une loi normale approximant une loi de Bernouilli.

```
p <- 0.5
sqrt_pq_95 <- 1.96 * sqrt(p * (1 - p))
n <- 1000

fn_b <- function(n) { sqrt_pq_95 / sqrt(n) }
fn_c <- function(n) { sqrt_pq_95 / n }
fn_d <- function(n) { sqrt_pq_95 / log(n) }

draw_traj_moins_p(sim, n, 10, p) +
    geom_line(aes(x = n, y = fn_b(n)), col = "red", size = 0.25) +
    geom_line(aes(x = n, y = -fn_b(n)), col = "red", size = 0.25) +
    labs(title = "Loi Bernouilli avec p := 5", subtitle = "Comparaison avec c / sqrt(n)")
```

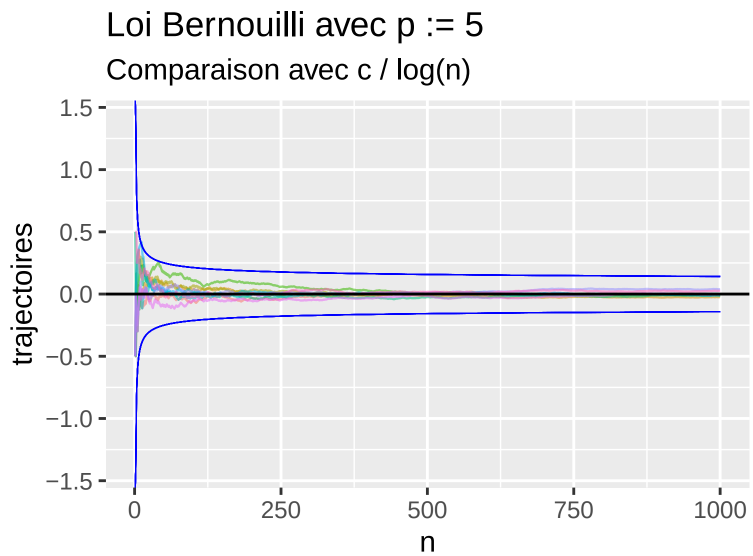


(c) Superposer en vert les courbes $n \mapsto 1.96\frac{\sqrt{p(1-p)}}{n}$ et $n \mapsto -1.96\frac{\sqrt{p(1-p)}}{n}$.

```
draw_traj_moins_p(sim, n, 10, p) +
    geom_line(aes(x = n, y = fn_c(n)), col = "green", size = 0.25) +
    geom_line(aes(x = n, y = -fn_c(n)), col = "green", size = 0.25) +
    labs(title = "Loi Bernouilli avec p := 5", subtitle = "Comparaison avec c / n")
```




```
draw_traj_moins_p(sim, n, 10, p) +
    geom_line(aes(x = n, y = fn_d(n)), col = "blue", size = 0.25) +
    geom_line(aes(x = n, y = -fn_d(n)), col = "blue", size = 0.25) +
    labs(title = "Loi Bernouilli avec p := 5", subtitle = "Comparaison avec c / log(n)")
```



Il est évident que la vitesse de $\frac{1}{n}$ est trop rapide, et que la vitesse de $\frac{1}{\log(n)}$ est trop lente. La vitesse de $\frac{1}{\sqrt{n}}$, par contre, n'est ni trop serrée, ni trop lâche.

Exercice 3 Illustration de la convergence en loi donnée par le TCL.

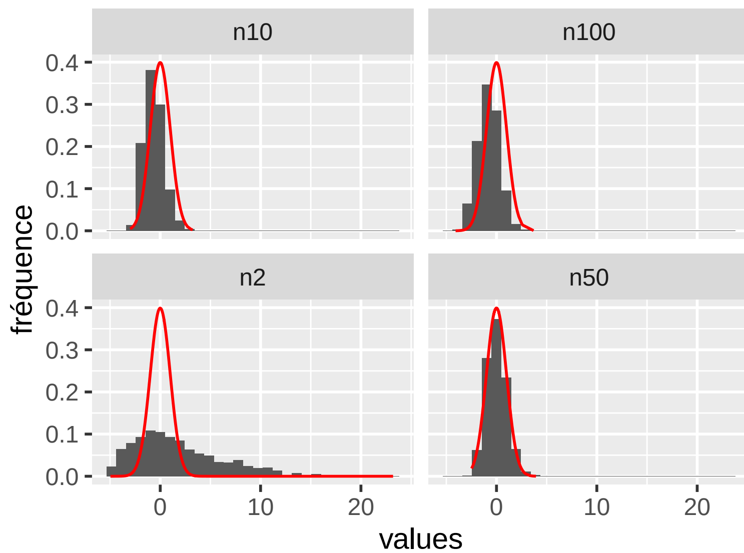
1. Ecrire une fonction `tclexpo` qui prend pour arguments n et le paramètre a de la loi exponentielle. Cette fonction simule 1000 vecteurs $(X_1, \ldots, X_n)$ où X_i suit une loi exponentielle de paramètre a et calculer pour chacun de ces vecteurs :

$$\sqrt{n}\frac{\frac{1}{n}\sum_{i=1}^{n} X_i - \mathbb{E}(X_1)}{\sqrt{\text{var}(X_1)}}$$

```
tclexpo <- function(n, a) {

    X1 <- rexp(n, a)
    mu <- mean(X1)
    sqv <- sqrt(var(X1))
    sqn <- sqrt(n)
    coeff <- force(sqn / (n * sqv)) # Calculate once, use 1000 times

    stat_Xi <- function(Xi) {
        coeff * sum(Xi - mu)
    }

    # Let's create a list so that we can easily map over it with lapply
    my_list <- list(X1)

    for (i in seq(999)) {
       my_list[[i + 1]] <- rexp(n, a)
    }

    unlist(lapply(my_list, stat_Xi)) # return a vector
}
```

2. Appliquer la fonction `tclexpo` pour $a = 2$ et $n \in \{2, 10, 50, 100\}$. Pour chaque valeur

```r
a <- 2
n <- c(2, 10, 50, 100)

out <- map(n, tclexpo, a) # Map tclexpo(n = ?, a) over the values of n

# Now I have a list of length 4 whose elements are vectors with size 1000.
df <- tibble(n2 = out[[1]], n10 = out[[2]], n50 = out[[3]], n100 = out[[4]])
df <- df |> pivot_longer(everything(), names_to = "names", values_to = "values")
# df$names <- factor(df$names, levels = "n10", "n2", "n50", "n100")


df |>
    ggplot(aes(values, after_stat(density))) +
    geom_histogram() +
    geom_line(aes(values, dnorm(values)), col = "red") +
    facet_wrap(~ names) +
    labs(y = "fréquence")
```



Etonnamment, pour $n = 50$ on a une loi qui suit la loi normale la plus loyalement.

Exercice 4 Illustration de l'inégalité de Tchebychev et de la LfGN

A faire.

Exercice 5 Illustration du théorème de Glivenko-Cantelli

Soit $X = (X_1, \dots, X_n)$ un vecteur de n variables aléatoires i.i.d de fonction de répartition F. On définit la fonction de répartition empirique par

$$F_n(t) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{X_i \leq t} \quad \forall t$$

```r
Fempirique <- function(X) {

    # First let's sort X
    X <- sort(X)
    n <- length(X)
```

```
    Fn_t <- function(t) {
        sum(X <= t) / n
    }

    # Then apply our function, returning a numeric vector
    right_column <- purrr::map_dbl(X, Fn_t)

    matrix(c(X, right_column), ncol = 2)
}
```

```
n <- 10 ** c(1, 2, 3, 4)

# first generate samples, then apply Fempirique to the samples
out <- lapply(map(n, runif), Fempirique) # Length 4 List

par(mfrow = c(2, 2))

for (i in seq(4)) {
    this_mat <- out[[i]]
    plot(this_mat[,1],
         this_mat[,2],
         xlab = paste("n =", n[i]),
         ylab = "F_n(t)")
    # title(xlab = paste(n[i]))
}
```

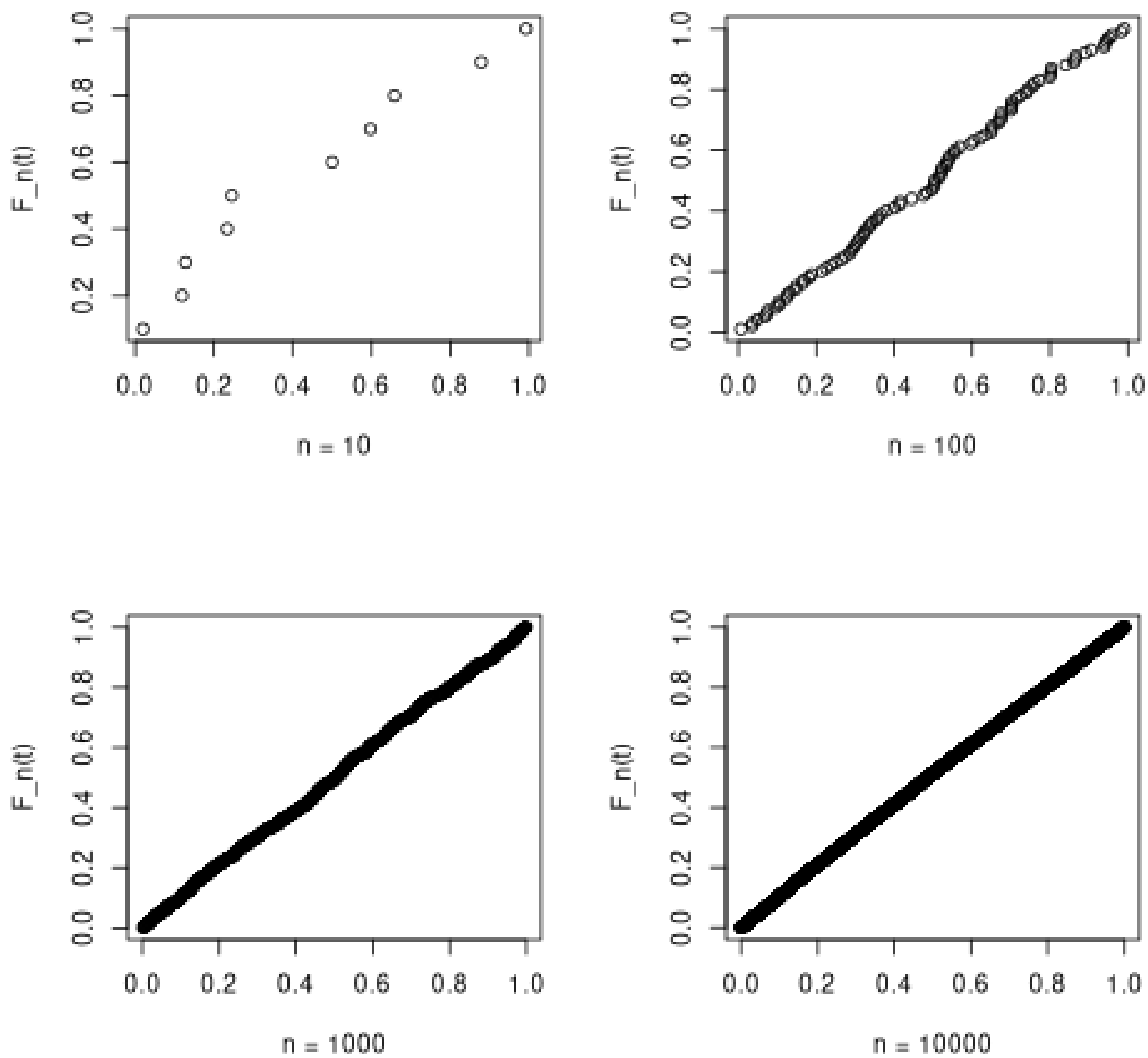


Figure 1: Fonction de répartition empirique pour une loi uniforme

Appendice
Ici les deux fonctions qui sont utilisées en haut (executé dans une cellule cachée).

```
draw_traj <- function(f_sim, n_trials, n_traj, title = NULL, subtitle = NULL) {

    df <- tibble(n = seq(n_trials))

    for (i in seq(n_traj)) {
        sim <- f_sim(n_trials) # Call the simulation function, like rnorm(n, 0, 1)
        cum_avg <- cumsum(sim) / seq(n_trials)
        df <- df |> add_column(cum_avg, .name_repair = c("unique"))
    }
```

```r
    df.long <- pivot_longer(df,
                            cols = !c(1), # Pivot everything BUT the first column
                            names_to = "data",
                            values_to = "traj")

    df.long |>
        ggplot(aes(n, traj, group = data)) +
        geom_line(aes(col = data, alpha = 1), size = 0.4) +
        theme(legend.position = "none") +
        labs(x = "n", y = "trajectoires", title = title, subtitle = subtitle) +
        geom_hline(yintercept = 0.5)

}

draw_traj_moins_p <- function(f_sim, n_trials, n_traj, p = 0.5, title = NULL, subtitle = NULL) {

    df <- tibble(n = seq(n_trials))

    for (i in seq(n_traj)) {
        sim <- f_sim(n_trials) # Call the simulation function, like rnorm(n, 0, 1)
        cum_avg <- cumsum(sim) / seq(n_trials)
        df <- df |> add_column(cum_avg - p, .name_repair = c("unique"))
    }

    df.long <- pivot_longer(df,
                            cols = !c(1), # Pivot everything BUT the first column
                            names_to = "data",
                            values_to = "traj")

    df.long |>
        ggplot(aes(n, traj, group = data)) +
        geom_line(aes(col = data, alpha = 1), size = 0.4) +
        theme(legend.position = "none") +
        labs(x = "n", y = "trajectoires", title = title, subtitle = subtitle) +
        geom_hline(yintercept = 0)

}
```