<u>Exo1 Exemple de 1000 simulations du jet d'une pièce équilibrée pour estimer la probabilité d'obtenir "pile"</u>

```
set.seed(123)

# Simulation de 1000 jets (0 = face, 1 = pile)
jets <- rbinom(1000, 1, 0.5)

# Histogramme
hist(jets,
    breaks = c(-0.5, 0.5, 1.5),
    main = "Histogramme des résultats des jets",
    xlab = "Résultat",
    ylab = "Fréquence",
    col = "lightblue",
    xaxt = "n")

axis(1, at = c(0, 1), labels = c("Face", "Pile"))
```

<u>Exo2 Exemple de 1000 simulations du jet d'un dé à 6 faces équilibrées pour estimer la probabilité d'obtenir laface "4".</u>

```
set.seed(123)  # pour que les résultats soient reproductibles

# Nombre de simulations
n <- 1000

# Simulation des lancers (1 à 6)
jets <- sample(1:6, size = n, replace = TRUE)

# Estimation de la probabilité d'obtenir un "4"
prob_4 <- mean(jets == 4)

# Affichage du résultat
```

```
cat("Probabilité estimée d'obtenir la face 4 :", prob_4, "\n")

# Histogramme des résultats
hist(jets,
     breaks = 0.5:6.5,  # pour centrer les barres sur les faces
     col = "lightgreen",
     main = "Histogramme des lancers de dé",
     xlab = "Face du dé",
     ylab = "Fréquence")
```

<u>Exo3 Simuler la somme des valeurs des faces obtenues en lançant 2 dés à 6 faces</u>

équilibrées

```
set.seed(123)  # pour la reproductibilité

# Nombre de simulations
n <- 1000

# Simulation des deux dés
de1 <- sample(1:6, size = n, replace = TRUE)
de2 <- sample(1:6, size = n, replace = TRUE)

# Somme des deux dés
somme <- de1 + de2

# Affichage des premières valeurs pour vérifier
head(somme)

# Estimation de la fréquence de chaque somme possible (2 à 12)
table(somme) / n

# Histogramme des sommes
```

```
hist(somme,
    breaks = seq(1.5, 12.5, by = 1),
    col = "lightblue",
    main = "Histogramme des sommes de 2 dés",
    xlab = "Somme",
    ylab = "Fréquence")
```

Exo4 Simuler la distribution du rang de la première boule rouge tirée, tirage AVEC remise

Modèle d'urne : Une urne contient **m** boules dont **r** rouges. On tire, successivement avec remise **n** boules dans l'urne et on note leurs couleurs dans l'ordre. La variable aléatoire X étudiée est le rang de la première rouge tirée (=0 si aucune rouge tirée au bout de n fois).

```
set.seed(123)

# Paramètres
m <- 10   # nombre total de boules
r <- 3    # nombre de boules rouges
n <- 5    # nombre de tirages
Nsim <- 10000  # nombre de simulations

# Probabilité de tirer une boule rouge à chaque tirage
p <- r / m

# Simulation
X <- numeric(Nsim)

for (i in 1:Nsim) {
  tirages <- rbinom(n, size = 1, prob = p)  # 1 = rouge, 0 = non rouge
  if (any(tirages == 1)) {
    X[i] <- which(tirages == 1)[1]  # rang de la première rouge
  } else {
    X[i] <- 0
  }
```

```
}

# Distribution empirique
distribution <- table(X) / Nsim
distribution
# Visualisation de la distribution
barplot(distribution,
        col = "lightcoral",
        main = "Distribution du rang de la première boule rouge",
        xlab = "Rang de la première boule rouge (0 = aucune)",
        ylab = "Probabilité")
```

Exo5 Simuler le problème du QCM

Un QCM comprend 20 questions. Pour chacune de ces questions QUATRE propositions de réponse sont faites, dont UNE seule est la bonne. Une bonne réponse vaut 1 point, une mauvaise vaut 0. Quelle est la probabilité qu'un élève, répondant à toutes les questions au hasard, ait la moyenne? Élaborons une stratégie de simulation pour estimer cette probabilité.

Modélisation du problème

- 20 questions
- 4 propositions par question
- 1 seule bonne réponse
- Réponse au hasard ⇒

p=14p = \frac{1}{4}p=41

- Une bonne réponse = 1 point
- Note maximale = 20
- **Moyenne** = 10/20
- Variable aléatoire :

X=nombre de bonnes réponses X

Avoir la moyenne ⇔ X≥10X

Stratégie de simulation

1. Simuler un QCM :
   chaque question est un **succès** (bonne réponse) avec probabilité p=1/4p = 1/4p=1/4
2. Répéter l'expérience un grand nombre de fois (ex : 10 000)
3. Compter le nombre de QCM où la note est ≥ 10
4. Estimer la probabilité par une fréquence

```
set.seed(123)

# Paramètres
n_questions <- 20
p <- 1/4
Nsim <- 10000

# Simulation : nombre de bonnes réponses pour chaque QCM
notes <- rbinom(Nsim, size = n_questions, prob = p)

# Estimation de la probabilité d'avoir la moyenne
prob_moyenne <- mean(notes >= 10)

prob_moyenne
```


```
set.seed(123)

# Paramètres
n_questions <- 20
p <- 1/4
Nsim <- 10000

# Simulation : nombre de bonnes réponses pour chaque QCM
notes <- rbinom(Nsim, size = n_questions, prob = p)

# Estimation de la probabilité d'avoir la moyenne
prob_moyenne <- mean(notes >= 10)
```

```
prob_moyenne


#Visualisation des notes obtenues
hist(notes,
     breaks = seq(-0.5, 20.5, by = 1),
     col = "lightblue",
     main = "Distribution des notes au QCM (réponses au hasard)",
     xlab = "Nombre de bonnes réponses",
     ylab = "Fréquence")
```

Si on voulait calculer exactement :

$$P(X \geq 10) = \sum_{k=10}^{20} \binom{20}{k} \left(\frac{1}{4}\right)^k \left(\frac{3}{4}\right)^{20-k}$$

Import et calculs

```
# Création du tableau de données
ca <- data.frame(
  borne_inf = c(145,150,155,160,165,170,175,180,185,190,195,200,205),
  borne_sup = c(150,155,160,165,170,175,180,185,190,195,200,205,210),
  centre = c(147.5,152.5,157.5,162.5,167.5,172.5,177.5,
             182.5,187.5,192.5,197.5,202.5,207.5),
  effectif = c(2,2,5,14,38,52,76,50,40,13,5,2,1)
)
# Effectif total
N <- sum(ca$effectif)


# Fréquences
ca$frequence <- round(ca$effectif / N, 3)
```

```r
# Moyenne (données groupées)

moyenne <- sum(ca$centre * ca$effectif) / N


# Ecart-type (population)

variance <- sum(ca$effectif * (ca$centre - moyenne)^2) / N

ecart_type <- sqrt(variance)


# Affichage des résultats

ca

moyenne

ecart_type

barplot(

  ca$effectif,

  names.arg = ca$centre,

  col = "lightblue",

  border = "blue",

  xlab = "Chiffre d'affaires (k€)",

  ylab = "Nombre de jours",

  main = "Répartition des chiffres d'affaires observés"

)
```

Ou bien

```r
plot(
ca$centre, ca$frequence,
type = "o",
pch = 16,
col = "red",
xlab = "Chiffre d'affaires (k€)",
ylab = "Fréquence",
main = "Profil des chiffres d'affaires"
)
```

Import direct :

csv

```
# Définir le répertoire de travail
setwd("chemin/vers/le/dossier")

# Import CSV
ca <- read.csv("chiffre_affaires.csv", header = TRUE)

# Vérifier
head(ca)
```

Excel

```
install.packages("readxl") # à faire une seule fois
library(readxl)

ca <- read_excel("chiffre_affaires.xlsx", sheet = 1)

# Vérification
head(ca)

N <- sum(ca$effectif)

ca$frequence <- round(ca$effectif / N, 3)

moyenne <- sum(ca$centre * ca$effectif) / N
ecart_type <- sqrt(sum(ca$effectif * (ca$centre - moyenne)^2) / N)

moyenne
ecart_type
```

Code R pour tracer la courbe de densité

Soit une VA X qui suit une loi normale de moyenne 500 et d'écart-type 50 : **X → N(500 ; 50)**

```
# Paramètres
mu <- 500
sigma <- 50

# Axe des x (± 4 écarts-types autour de la moyenne)
x <- seq(mu - 4*sigma, mu + 4*sigma, length.out = 1000)

# Densité de la loi normale
y <- dnorm(x, mean = mu, sd = sigma)

# Tracé
plot(
x, y,
type = "l",
lwd = 2,
```

```
col = "blue",
main = "Densité de la loi normale N(500 ; 50)",
xlab = "x",
ylab = "f(x)"
)

# Ligne verticale pour la moyenne
abline(v = mu, col = "red", lwd = 2, lty = 2)
```

1- Calculer les probabilités suivantes : (les calculs pourront être faits dans un premier temps à la calculatrice puis en utilisant la loi normale centrée réduite)
   a. P(X<=550)
   b. P(X<=600)
   c. P(X<=650)
   d. P(X<=700)
   e. P(X>550)
   f. P(X<=450)
   g. P(450<X<=500)

```
mu <- 500
sigma <- 50

p1 <- pnorm(550, mean = mu, sd = sigma)
p2 <- pnorm(600, mean = mu, sd = sigma)
p3 <- pnorm(650, mean = mu, sd = sigma)
p4 <- pnorm(700, mean = mu, sd = sigma)
p5 <- 1 - pnorm(550, mean = mu, sd = sigma)
p6 <- pnorm(450, mean = mu, sd = sigma)
p7 <- pnorm(500, mean = mu, sd = sigma) - pnorm(450, mean = mu, sd = sigma)
round(c(p1, p2, p3, p4,p5,p6,p7), 4)
```