Fiche TD avec le logiciel R : tdr21

Lois de Probabilités

A.B. Dufour, D. Chessel & J.R. Lobry

---

d p q r, Loi binomiale, loi normale, loi hypergéométrique, loi de Poisson, loi uniforme, loi du Khi2, loi de Student, loi de Fisher, loi exponentielle, loi beta

1 d p q r

Donner toutes les valeurs de la loi binomiale de paramètres $n = 10$ et $p = 1/3$ ($\mathcal{B}(10, 1/3)$).

```
options(digits = 7)
dbinom(0:10, 10, 1/3)
 [1] 1.734153e-02 8.670765e-02 1.950922e-01 2.601229e-01 2.276076e-01 1.365645e-01
 [7] 5.690190e-02 1.625768e-02 3.048316e-03 3.387018e-04 1.693509e-05
```

```
options(digits = 4)
```

Rappeler l'ordre avec la touche ↑.

```
dbinom(0:10, 10, 1/3)
 [1] 1.734e-02 8.671e-02 1.951e-01 2.601e-01 2.276e-01 1.366e-01 5.690e-02 1.626e-02
 [9] 3.048e-03 3.387e-04 1.694e-05
```

Quelle est la probabilité d'obtenir 1 avec une loi binomiale $\mathcal{B}(10, 1/3)$?

```
dbinom(1, 10, 1/3)
[1] 0.0867
```

Quelle est la probabilité d'obtenir plus de 45 et moins de 55 avec une loi binomiale $\mathcal{B}(100, 1/2)$?
Taper :

```
?dbinom
```

La fenêtre de documentation de la fonction apparaît et vous pouvez lire en bas :

```
# Compute P(45 < X < 55) for X Binomial(100,0.5)
sum(dbinom(46:54, 100, 0.5))
```

Copier l'ordre et le coller dans la fenêtre de commande :

```
sum(dbinom(46:54, 100, 0.5))
```

```
[1] 0.6318
```

Quelle est la probabilité d'obtenir au plus 1 avec une loi binomiale pour $\mathcal{B}(10, 1/3)$?

```
pbinom(1, 10, 1/3)
[1] 0.1040
dbinom(0, 10, 1/3)
[1] 0.01734
dbinom(1, 10, 1/3)
[1] 0.0867
dbinom(0, 10, 1/3) + dbinom(1, 10, 1/3)
[1] 0.1040
pbinom(0.5, 10, 1/3)
[1] 0.01734
pbinom(-0.5, 10, 1/3)
[1] 0
```

Noter que `d` donne les valeurs $P(X = j)$ (densité, définie pour les valeurs possibles) et que `p` donne les valeurs $P(X \le x)$ (fonction de répartition, définie pour tout x).

Exercice

1. Quelle est la probabilité de dépasser strictement 4 pour une loi de Poisson de paramètre 2.7 ($\mathcal{P}(2.7)$) ?

   ```
   [1] 0.1371
   ```

2. Quelle est la probabilité de dépasser 1.96 pour une loi normale centrée réduite ($\mathcal{N}(0, 1)$) ?

   ```
   [1] 0.025
   ```

Quelle est la valeur x telle que $P(X \le x) = 0.975$ pour une loi normale (quantile) ?

```
qnorm(0.975)
[1] 1.96
```

Exercice. Quel est le quantile 1 % pour une loi $\mathcal{T}$ de Student à 5 ddl ?

```
[1] -3.365
```

Noter que `q` donne pour y *la plus grande valeur* de x telle que $P(X \le x) = y$ (quantile).

Donner un échantillon aléatoire simple de 10 valeurs d'une loi de Poisson $\mathcal{P}(2.7)$:

```
rpois(10, 2.7)
[1] 1 3 3 1 2 3 1 0 1 1
```

d'une loi normale réduite :

```
rnorm(10)
```

```
 [1]  0.17117  0.02731  0.23513  0.55858  2.46481 -0.61178 -0.84948 -1.06342  0.30373
[10] -1.46248
```

d'une loi du Khi2 à 2 ddl :

```
rchisq(10, 2)
[1] 2.1725 1.6271 0.8919 1.0711 1.6308 1.2386 5.0406 0.1879 1.3090 3.8177
```

d'une loi binomiale $n = 100$ et $p = 1/2$.

```
rbinom(10, 100, 0.5)
[1] 44 57 48 46 46 52 51 55 53 52
```

Noter que **r** donne des échantillons.

Les d, p, q et r disponibles dans la distribution standard de R sont donnés dans la table 1, ceux disponibles dans le package `SuppDists` sont donnés dans la table 2.

	d	p	q	r
beta	dbeta	pbeta	qbeta	rbeta
binom	dbinom	pbinom	qbinom	rbinom
cauchy	dcauchy	pcauchy	qcauchy	rcauchy
chisq	dchisq	pchisq	qchisq	rchisq
exp	dexp	pexp	qexp	rexp
f	df	pf	qf	rf
gamma	dgamma	pgamma	qgamma	rgamma
geom	dgeom	pgeom	qgeom	rgeom
hyper	dhyper	phyper	qhyper	rhyper
lnorm	dlnorm	plnorm	qlnorm	rlnorm
logis	dlogis	plogis	qlogis	rlogis
nbinom	dnbinom	pnbinom	qnbinom	rnbinom
norm	dnorm	pnorm	qnorm	rnorm
pois	dpois	ppois	qpois	rpois
signrank	dsignrank	psignrank	qsignrank	rsignrank
t	dt	pt	qt	rt
unif	dunif	punif	qunif	runif
weibull	dweibull	pweibull	qweibull	rweibull
wilcox	dwilcox	pwilcox	qwilcox	rwilcox

Table 1 – Les lois de probabilité définies dans R.

La figure 1 illustre l'utilisation de `d p q r` dans le cas d'une loi binomiale et de son approximation par une loi normale. Elle a été produite avec la fonction suivante, où `size` représente le nombre de tirages et `prob` la probabilité d'un succès :

```
dpqr <- function(size = 20, prob = 0.5) {
    opar <- par(no.readonly = TRUE)
    par(mfrow = c(2, 2), cex.main = 3)
    plot(0:size, dbinom(0:size, size, prob), type = "h", main = "d",
        las = 1, ylab = "probabilité", xlab = "Nombre de succès")
    mean = size * prob
    sd = sqrt(size * prob * (1 - prob))
    xseq <- seq(from = 0, to = size, length = 255)
```

	d	p	q	r
Friedman	dFriedman	pFriedman	qFriedman	rFriedman
ghyper	dghyper	pghyper	qghyper	rghyper
invGauss	dinvGauss	pinvGauss	qinvGauss	rinvGauss
Johnson	dJohnson	pJohnson	qJohnson	rJohnson
Kendall	dKendall	pKendall	qKendall	rKendall
KruskalWallis	dKruskalWallis	pKruskalWallis	qKruskalWallis	rKruskalWallis
maxFratio	dmaxFratio	pmaxFratio	qmaxFratio	rmaxFratio
NormScore	dNormScore	pNormScore	qNormScore	rNormScore
Pearson	dPearson	pPearson	qPearson	rPearson
Spearman	dSpearman	pSpearman	qSpearman	rSpearman

TABLE 2 – Les lois de probabilité définies dans le package SuppDists

```
    lines(xseq, dnorm(xseq, mean, sd), col = "red")
    legend("topleft", inset = 0.01, legend = c("dbinom", "dnorm"),
        lty = 1, col = c("black", "red"))
    plot(0:size, pbinom(0:size, size, prob), type = "s", main = "p",
        las = 1, ylab = "probabilité", xlab = "Nombre de succès")
    lines(xseq, pnorm(xseq, mean, sd), col = "red")
    legend("topleft", inset = 0.01, legend = c("pbinom", "pnorm"),
        lty = 1, col = c("black", "red"))
    pseq <- seq(from = 0, to = 1, length = 255)
    plot(pseq, qbinom(pseq, size, prob), type = "s", main = "q",
        las = 1, ylab = "Nombre de succès", xlab = "Probabilité")
    lines(pseq, qnorm(pseq, mean, sd), col = "red")
    legend("topleft", inset = 0.01, legend = c("qbinom", "qnorm"),
        lty = 1, col = c("black", "red"))
    n <- 500
    bin <- rbinom(n, size, prob)
    plot(table(bin)/n, xlim = c(0, size), main = "r", lwd = 1, las = 1,
        ylab = "probabilité", xlab = "Nombre de succès", xaxt = "n")
    axis(1, pretty(c(0, size)))
    nor <- rnorm(n, mean, sd)
    lines(density(nor), col = "red")
    legend("topleft", inset = 0.01, legend = c("rbinom", "rnorm"),
        lty = 1, col = c("black", "red"))
    par(opar)
}
```

Copiez/collez le code source de la fonction `dpqr()` dans votre console R, puis explorez l'espace des paramètres avec par exemple :

```
dpqr()
dpqr(30)
dpqr(50)
dpqr(100)
dpqr(50, 0.6)
dpqr(50, 0.7)
dpqr(50, 0.8)
dpqr(50, 0.9)
dpqr(50, 0.95)
dpqr(50, 0.99)
```

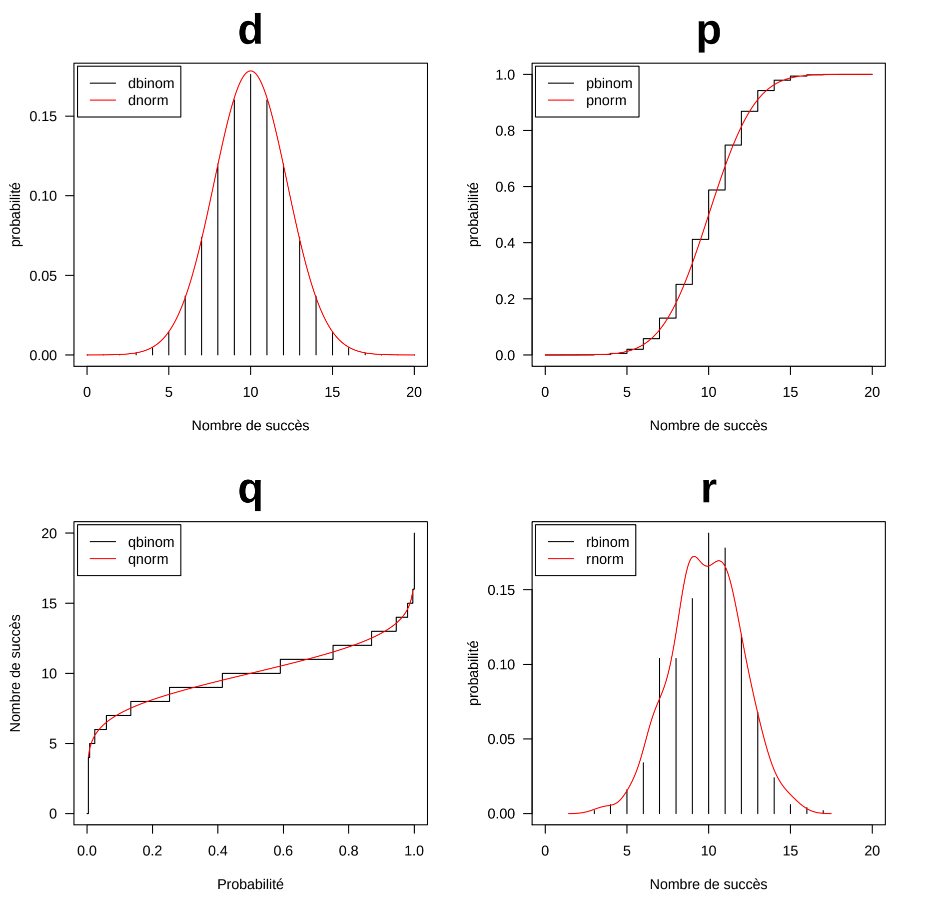


FIGURE 1 – Illustration de `d p q r` dans le cas d'une loi discrète, la loi binomiale avec $n = 20$ et $p = 0.5$ et de son approximation par une loi continue, la loi normale de paramètres $\mu = np$ et $\sigma = \sqrt{np(1-p)}$. `d` donne la fonction de densité de probabilité, c'est à dire les valeurs $P(X = j)$. Notez la différence entre la loi discrète et et la loi continue : dans le cas de la loi discrète la fonction n'est définie que pour les valeurs possibles $(0, 1, 2, \ldots, n)$. `p` donne la fonction de répartion, c'est à dire $P(X \leq j)$. Elle est définie partout, aussi bien pour la loi discrète que pour la loi continue. `q` donne les quantiles, c'est la fonction réciproque de la fonction de répartition. `r` donne un échantillon pseudo-aléatoire.

2 Loi binomiale

Soit X une loi binomiale. La probabilité d'obtenir k succès pour n essais indépendants avec une probabilité p de succès est :

$$P(X = k) = \left(\begin{array}{c} n \\ k \end{array} \right) p^k (1-p)^{n-k} \quad 0 \leq k \leq n$$

Sa moyenne et sa variance sont :

$$\mu = np \qquad \text{et} \qquad \sigma^2 = np\,(1-p)$$

La loi binomiale normalisée est définie par :

$$\Phi^\bullet = \left\{ \varphi_0^\bullet = \frac{0-\mu}{\sigma}, \varphi_1^\bullet = \frac{1-\mu}{\sigma}, ..., \varphi_n^\bullet = \frac{n-\mu}{\sigma} \right\}$$

et,

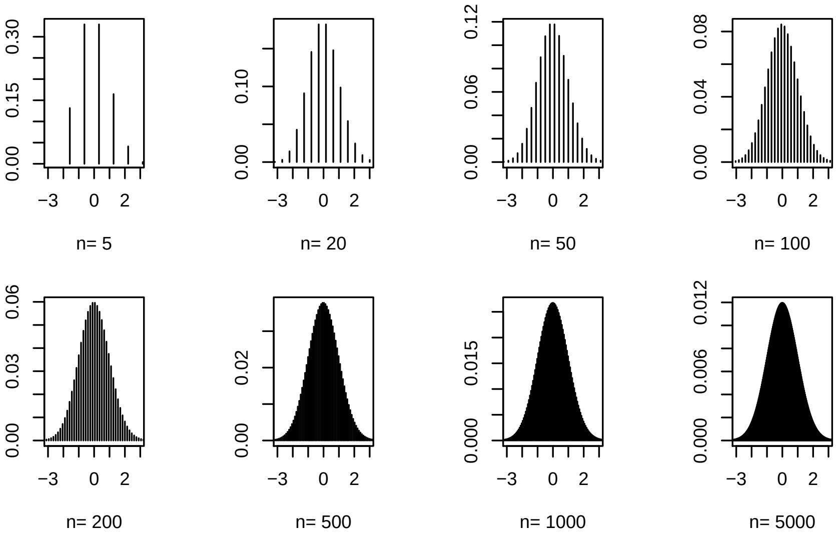
$$P\left(\varphi_k^\bullet\right) = \left(\begin{array}{c} n \\ k \end{array} \right) p^k \left(1-p\right)^{n-k} \quad 0 \leq k \leq n$$

L'espérance vaut 0 et la variance vaut 1.

Comparer les lois binomiales normalisées pour $p = 1/3$ et n = *5, 20, 50, 100, 200, 500, 1000, 5000.* Pour cela, écrire la fonction ci-dessous puis appliquer la aux différentes valeurs de n proposées.

```
loibin <- function(n, p) {
    y <- dbinom(0:n, n, p)
    x <- ((0:n) - n * p)/sqrt(n * p * (1 - p))
    etiq0 <- paste("n=", n)
    plot(x, y, xlab = etiq0, ylab = "", type = "h", xlim = c(-3,
        3))
}
```

```
old.par <- par(no.readonly = TRUE)
effectif <- c(5, 20, 50, 100, 200, 500, 1000, 5000)
par(mfrow = c(2, 4))
par(mar = c(5, 4, 1, 2))
sapply(effectif, loibin, p = 1/3)
par(old.par)
```



Commenter.

3 Loi normale

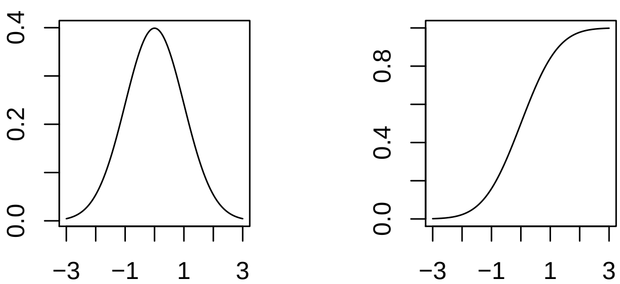
La fonction de répartition de la loi normale est définie comme suit :

$$P(X \leq x) = F_N(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$$

Sa moyenne et sa variance sont :

$$\mu = E(X) = \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} t e^{-\frac{t^2}{2}} dt \qquad \text{et} \qquad \sigma^2 = E\left(X^2\right) = \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} t^2 e^{-\frac{t^2}{2}} dt$$

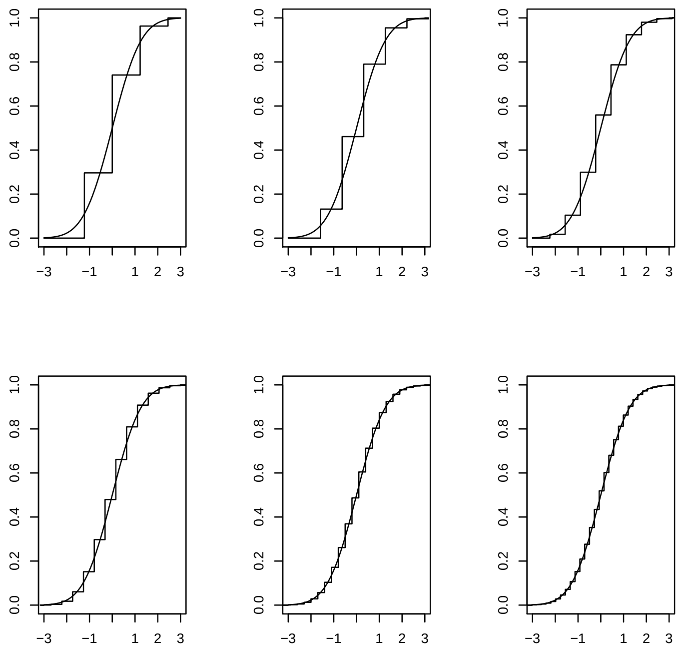
```
old.par <- par(no.readonly = TRUE)
par(mfrow = c(1, 2))
par(mar = c(5, 4, 3, 2))
w0 <- seq(-3, 3, length = 500)
plot(w0, dnorm(w0), type = "l", xlab = "", ylab = "")
plot(w0, pnorm(w0), type = "l", xlab = "", ylab = "")
par(old.par)
```



X suit une loi normale de moyenne μ et de variance σ^2 si $\frac{X-\mu}{\sigma}$ suit une loi normale.

Comparer la loi binomiale normalisée et la loi normale.

```
old.par <- par(no.readonly = TRUE)
g1 <- function(n, p) {
    mu <- n * p
    sigma <- sqrt(n * p * (1 - p))
    w0 <- (0:n - mu)/sigma
    x <- c(-3, rep(w0, rep(2, n + 1)), 3)
    z <- rep(c(-1, 0:n), rep(2, n + 2))
    y <- pbinom(z, n, p)
    etiq0 <- paste("Bin Nor n =", n)
    plot(c(0, 0), xlim = c(-3, 3), ylim = c(0, 1), type = "n", xlab = etiq0,
        ylab = "", ann = FALSE)
    lines(x, y)
    lines(seq(-3, 3, length = 100), pnorm(seq(-3, 3, length = 100)))
}
par(mfrow = c(2, 3))
par(mar = c(4, 4, 4, 2))
effectif <- c(3, 5, 10, 20, 50, 100)
sapply(effectif, g1, p = 1/3)
par(old.par)
```



4 Loi hypergéométrique

Une urne contient m boules blanches et n boules noires. Soit X une loi hypergéométrique. La probabilité d'obtenir j boules blanches pour k tirages sans remise dans l'urne vaut :

$$P(X = j) = \frac{\begin{pmatrix} m \\ j \end{pmatrix}\begin{pmatrix} n \\ k-j \end{pmatrix}}{\begin{pmatrix} m+n \\ k \end{pmatrix}}$$

Sa moyenne et sa variance sont :

$$\mu = \frac{km}{n+m} \qquad \text{et} \qquad \sigma^2 = \frac{nmk\,(n+m-k)}{(m+n)^2\,(m+n-1)}$$

Construire une urne contenant 30 boules blanches et 70 boules noires. Tirer sans remise 15 boules et compter les blanches. Recommencer 1000 fois et tabuler le résultat. Comparer les fréquences et les probabilités.

```
urne <- rep(c("b", "n"), c(30, 70))
urne
 [1] "b" "b" "b" "b" "b" "b" "b" "b" "b" "b" "b" "b" "b" "b" "b" "b" "b" "b" "b" "b"
[21] "b" "b" "b" "b" "b" "b" "b" "b" "b" "b" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n"
[41] "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n"
[61] "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n"
[81] "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n" "n"
sample(urne, 15, replace = FALSE)
[1] "n" "n" "n" "b" "n" "n" "n" "b" "n" "b" "b" "n" "n" "n" "n"
sample(urne, 15, replace = FALSE)
[1] "b" "b" "n" "n" "b" "n" "n" "b" "n" "n" "b" "n" "b" "n" "n"
w0 <- sample(urne, 15, replace = F)
w0
[1] "n" "b" "n" "b" "n" "n" "n" "n" "n" "b" "n" "b" "n" "b" "b"
sum(w0 == "b")
[1] 6
sum(sample(urne, 15, replace = F) == "b")
[1] 2
sum(sample(urne, 15, replace = F) == "b")
[1] 7
echa <- function(x) {
    sum(sample(urne, 15, replace = F) == "b")
}
echa()
[1] 6
echa()
[1] 4
tapply(1:10, as.factor(1:10), echa)
1  2  3  4  5  6  7  8  9 10
5  7  3  8  3  6  5  6  5  4
res <- tapply(1:1000, as.factor(1:1000), echa)
table(res)
res
  0   1   2   3   4   5   6   7   8   9  10
  3  28  79 189 227 213 153  64  33   9   2
table(res)/1000
res
    0     1     2     3     4     5     6     7     8     9    10
0.003 0.028 0.079 0.189 0.227 0.213 0.153 0.064 0.033 0.009 0.002
dhyper(0:10, 30, 70, 15)
 [1] 0.002848 0.022885 0.081502 0.170499 0.234075 0.223151 0.152426 0.075862 0.027696
[10] 0.007405 0.001435
```

5 Loi de Poisson

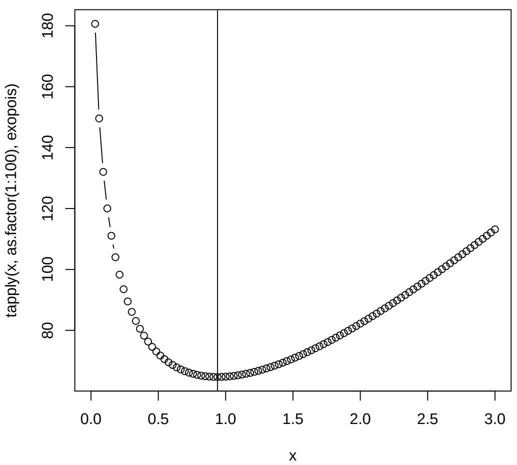
$$P(X = j) = e^{-\lambda}\frac{\lambda^j}{j!}$$

Probabilité d'obtenir j succès pour n essais indépendants avec une probabilité p de succès quand $n \to \infty \quad p \to 0 \quad np = \lambda$.
Sa moyenne et sa variance sont :

$$\mu = \lambda \qquad \text{et} \qquad \sigma^2 = \lambda$$

Exemple. On a compté les appels téléphoniques par unité de temps pendant une période de 50 unités. On a trouvé 0 (21 fois), 1 (16 fois), 2 (9 fois), 3 (3 fois) et 4 (1 fois). Tracer la fonction de vraisemblance de l'échantillon en supposant qu'il s'agit d'un échantillon aléatoire simple d'une loi de Poisson.

```
a <- c(21, 16, 9, 3, 1)
sum(a)
[1] 50
b <- rep(0:4, a)
b
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2
[41] 2 2 2 2 2 2 3 3 3 4
prod(dpois(b, 1.5))
[1] 1.907e-31
log(prod(dpois(b, 1.5)))
[1] -70.73
-log(prod(dpois(b, 1.5)))
[1] 70.73
-sum(dpois(b, 1.5, log = T))
[1] 70.73
exopois <- function(x) {
    -sum(dpois(b, x, log = T))
}
exopois(1.5)
[1] 70.73
x <- seq(0, 3, le = 100)
plot(x, tapply(x, as.factor(1:100), exopois), type = "b")
abline(v = mean(b))
```



Exercice. Démontrer que la moyenne de l'échantillon est l'estimation au maximum de vraisemblance du paramètre d'une loi de Poisson (*Non, il n'y a pas de fonction pour faire cela*).

6 Loi uniforme

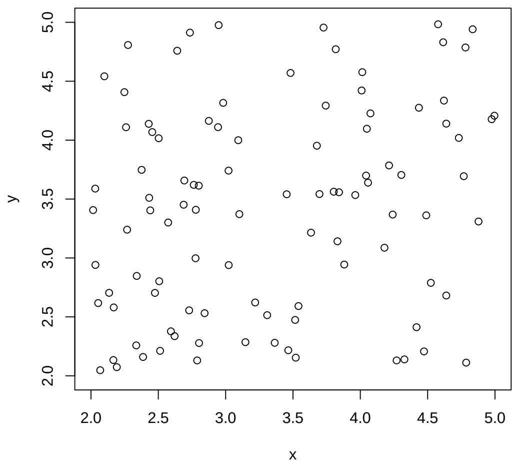
$$P(X \leq x) = F_{Uab}(x) = \begin{cases} 0 \; si \; x \leq a \\ \frac{x-a}{b-a} \; si \; x \leq a \\ 1 \; si \; x \geq b \end{cases}$$

Sa moyenne et sa variance sont :

$$\mu = E(X) = \frac{a+b}{2} \qquad \mathrm{et} qquad \sigma^2 = E\left((X - E(X))^2\right) = \frac{(b-a)^2}{12}$$

Placer 100 points au hasard dans un carré.

```
x <- runif(100, 2, 5)
y <- runif(100, 2, 5)
plot(x, y, xlim = c(2, 5), ylim = c(2, 5), pch = 21)
```



Ceci pourrait être remplacer par une seule expression :

```
plot(runif(100, 2, 5), runif(100, 2, 5), xlim = c(2, 5), ylim = c(2,
    5), pch = 21)
```

Exercice. Découper le carré en 100 parcelles élémentaires et compter le nombre de points par parcelle. Pour ce faire, utiliser les fonctions : `table`, `cut`. Comparer à une loi de Poisson de paramètre 1.

7 Loi du Khi2

Si $X_1, X_2, ..., X_p$ sont p lois normales centrées réduites, indépendantes, alors $X_1^2 + X_2^2 + ... + X_p^2$ suit une loi du Khi2 (ou loi du χ^2) à p degrés de liberté.

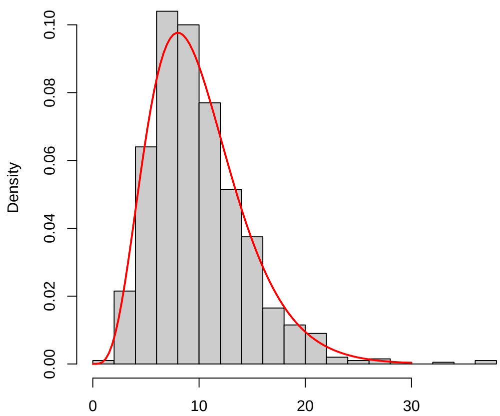
$$\chi^2 = \sum_{i=1}^{p} X_i^2$$

Sa moyenne et sa variance sont :

$$\mu = p \qquad \text{et} \qquad \sigma^2 = 2p$$

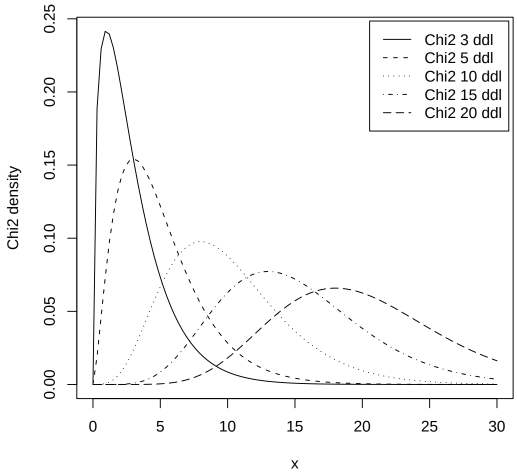
Illustrer la définition

```
exochi <- function(x) {
    sum((rnorm(10))^2)
}
hist(tapply(1:1000, as.factor(1:1000), exochi), proba = T, nclass = 20,
    main = "", col = grey(0.8), xlab = "")
x0 <- seq(0, 30, le = 100)
lines(x0, dchisq(x0, 10), col = "red", lwd = 2)
```



Tracer les densités pour différents degrés de liberté.

```
x0 <- seq(0, 30, le = 100)
ddl <- c(3, 5, 10, 15, 20)
maty <- matrix(nrow = 100, ncol = 5)
for (j in 1:5) maty[, j] = dchisq(x0, ddl[j])
{
    plot(x0, maty[, 1], type = "n", xlab = "x", ylab = "Chi2 density")
    for (j in 1:5) lines(x0, maty[, j], lty = j)
    legend0 <- c("Chi2 3 ddl", "Chi2 5 ddl", "Chi2 10 ddl", "Chi2 15 ddl",
        "Chi2 20 ddl")
}
legend("topright", inset = 0.01, legend0, lty = 1:5)
```



8 Loi t de Student

Si $X_1, X_2, ..., X_p$ sont p lois normales $\mathcal{N}(0,1)$ indépendantes et si :

$$\bar{X} = \frac{1}{p}\sum\nolimits_{j=1}^{p} X_j \quad S^2 = \frac{1}{p-1}\sum\nolimits_{j=1}^{p}\left(X_j - \bar{X}\right)^2$$

alors $Z = \frac{\sqrt{p}\bar{X}}{S}$ suit une loi de Student à p degrés de liberté.

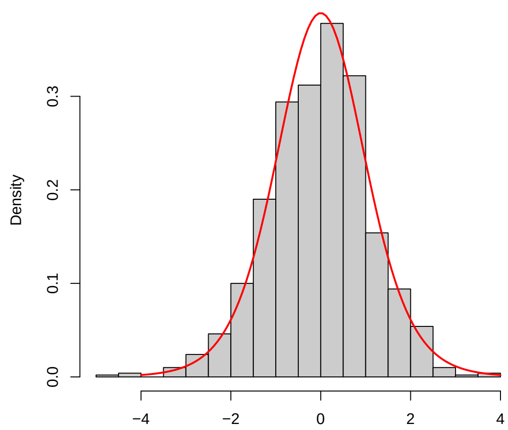
Sa moyenne et sa variance sont :

$$\mu = 0 \qquad \text{et} \qquad \sigma^2 = \frac{p}{p-2}$$

Illustrer la définition.

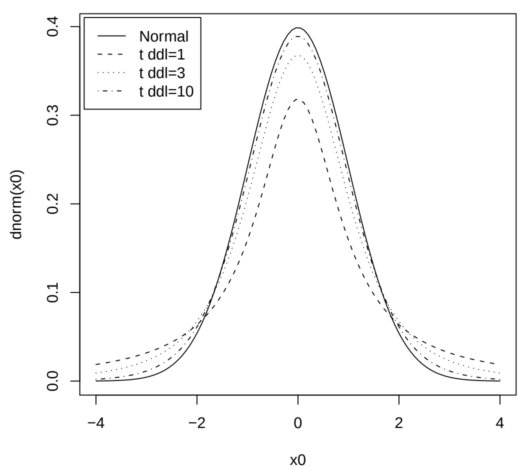
```
exot <- function(x) {
    x <- rnorm(10)
    m <- mean(x)
    s <- sqrt(var(x))
    sqrt(10) * m/s
}
```

```
hist(tapply(1:1000, as.factor(1:1000), exot), proba = T, nclass = 20,
    main = "", col = grey(0.8), xlab = "")
x0 <- seq(-4, 4, le = 100)
lines(x0, dt(x0, 10), col = "red", lwd = 2)
```



Tracer les densités pour différents degrés de liberté.

```
plot(x0, dnorm(x0), type = "l", lty = 1)
lines(x0, dt(x0, 1), type = "l", lty = 2)
lines(x0, dt(x0, 3), type = "l", lty = 3)
lines(x0, dt(x0, 10), type = "l", lty = 4)
legend0 <- c("Normal", "t ddl=1", "t ddl=3", "t ddl=10")
legend("topleft", inset = 0.01, legend0, lty = 1:4)
```



9 Loi de Fisher

Si X suit une loi du khi2 à n degrés de liberté (χ^2_n), si Y suit une loi du khi2 à p degrés de liberté (χ^2_p) et si X et Y sont indépendantes alors $Z = \frac{X/n}{Y/p}$ suit une loi de Fisher F à n et p degrés de liberté.

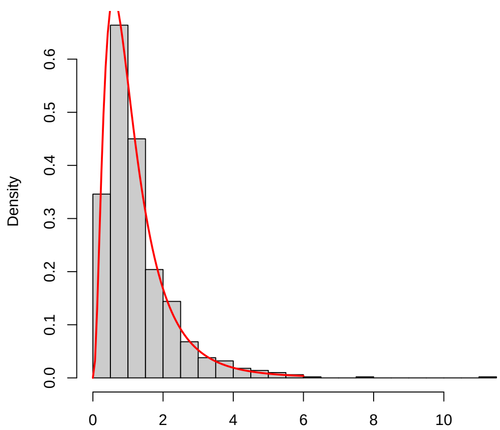
$$X \to \chi^2_n \,; Y \to \chi^2_p \,; X \text{ et } Y \text{ indépendantes} \Rightarrow Z = \frac{X/n}{Y/d} \to F_{n,p}$$

Sa moyenne et sa variance sont :

$$\mu = \frac{p}{p-2} \qquad \text{et} \qquad \sigma^2 = \frac{2p^2(n+p-2)}{n(p-4)(p-2)^2}$$

Illustrer la définition.

```
U <- rchisq(1000, 7)
V <- rchisq(1000, 10)
X <- (U/7)/(V/10)
hist(X, proba = T, nclass = 30, col = grey(0.8), main = "", xlab = "")
w <- seq(0, 6, le = 100)
lines(w, df(w, 7, 10), col = "red", lwd = 2)
```



10 Loi exponentielle

$$P\left(X < t\right) = \int_0^t \alpha e^{-\alpha x} dx = 1 - e^{-\alpha t}$$

Sa moyenne et sa variance sont :

$$\mu = E\left(X\right) = \frac{1}{\alpha} \qquad \text{et} \qquad \sigma^2 = E\left(\left(X - E\left(X\right)\right)^2\right) = \frac{1}{\alpha^2}$$

Exercice. Tirer 1000 points au hasard sur [0,1]. Compter ces points pour 1000 intervalles. Comparer à une loi de Poisson.

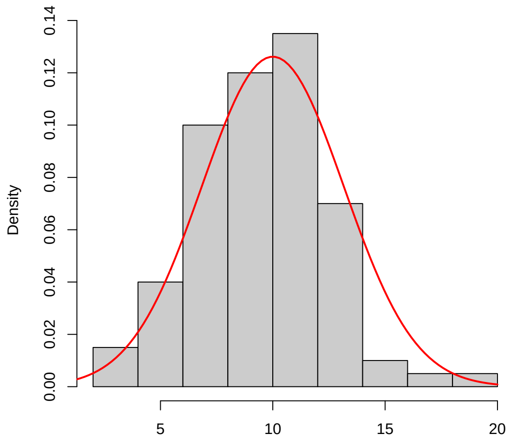
```
x <- runif(1000)
n1000 <- as.vector(table(cut(x, br = seq(0, 1, le = 1000))))
table(n1000)
n1000
  0   1   2   3   4   5   6
360 375 192  51  18   2   1
```

```
dpois(0:6, 1)
```

```
[1] 0.3678794 0.3678794 0.1839397 0.0613132 0.0153283 0.0030657 0.0005109
```

Compter ces points pour 100 intervalles. Comparer à une loi normale.

```
n100 <- as.vector(table(cut(x, br = seq(0, 1, le = 101))))
hist(n100, proba = T, main = "", col = grey(0.8), xlab = "")
x0 <- seq(0, 20, le = 100)
lines(x0, dnorm(x0, 10, sqrt(10)), col = "red", lwd = 2)
```



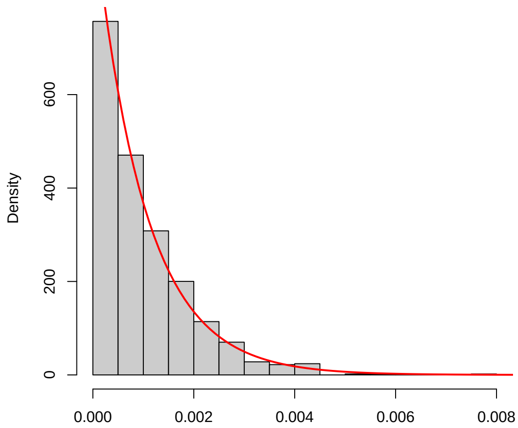
Compter ces points pour 10 intervalles.

```
n10 <- as.vector(table(cut(x, br = seq(0, 1, le = 11))))
n10
```

```
 [1] 110 100  98 102  87  98  98 100 116  91
```

Donner la distribution du temps d'attente du suivant. Comparer à une loi exponentielle.

```
delai <- diff(sort(x))
hist(delai, proba = T, nclass = 20, main = "", xlab = "", col = grey(0.8))
x0 <- seq(0, 0.01, le = 100)
lines(x0, dexp(x0, 1000), col = "red", lwd = 2)
```



11 Loi beta

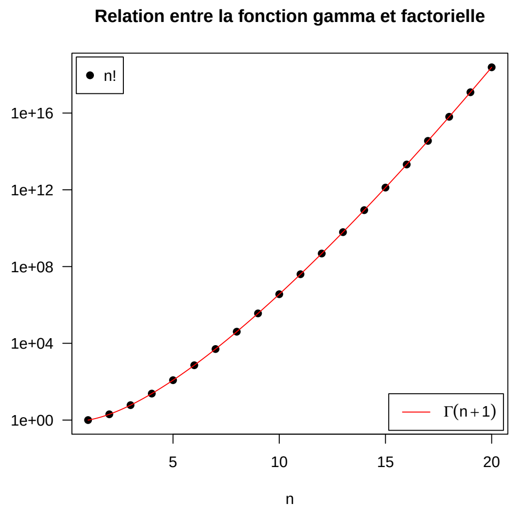
$$P\left(X \leq x\right) = F_{\beta_{ab}}\left(x\right) = \int_{-\infty}^{x} \varphi(t)dt = \int_{-\infty}^{x} \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} t^{a-1}(1-t)^{b-1}dt$$

Sa moyenne et sa variance sont :

$$E(X) = \frac{a}{a+b} \qquad \text{et} qquad V(X) = \frac{ab}{(a+b)^2(a+b+1)}$$

La fonction Γ représente ici la prolongation continue de la fonction factorielle avec $\Gamma(n+1) = n!$:

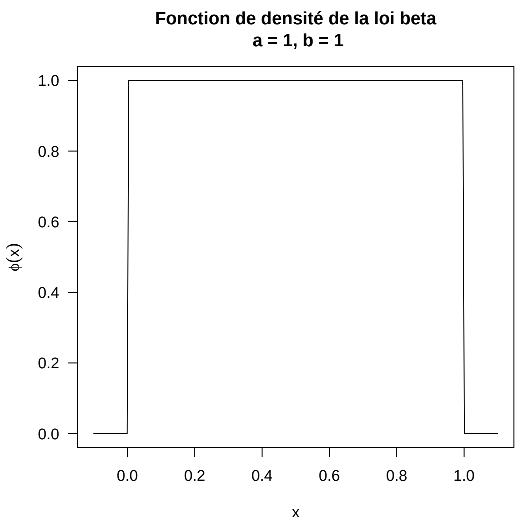
```
prod(1:5)
[1] 120
gamma(6)
[1] 120
nseq <- 1:20
nfac <- sapply(nseq, function(n) prod(1:n))
plot(nseq, nfac, log = "y", las = 1, xlab = "n", ylab = "", pch = 19,
    main = "Relation entre la fonction gamma et factorielle")
legend("topleft", inset = 0.01, legend = "n!", pch = 19)
xseq <- seq(from = 1, to = 20, length = 255)
lines(xseq, gamma(xseq + 1), col = "red")
legend("bottomright", inset = 0.01, legend = expression(Gamma(n +
    1)), lty = 1, col = "red")
```



La fonction de densité de la loi beta est définie entre 0 et 1, elle est très flexible.

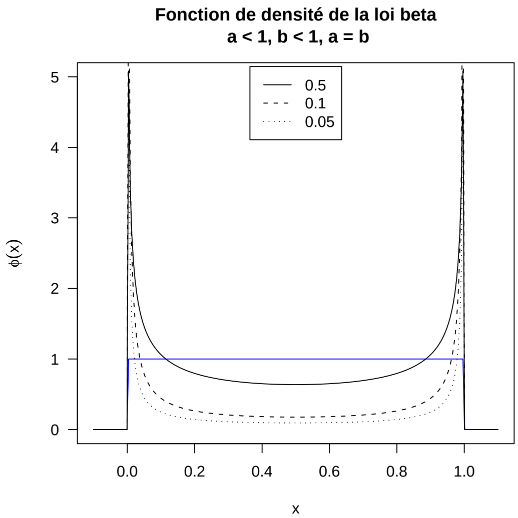
11.1 $a = 1, b = 1$

La fonction de densité de la loi beta est la même qu'une distribution uniforme entre 0 et 1 :

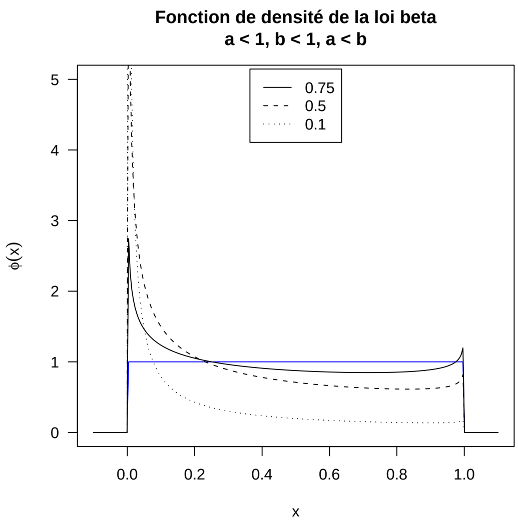


11.2 $a < 1, b < 1$

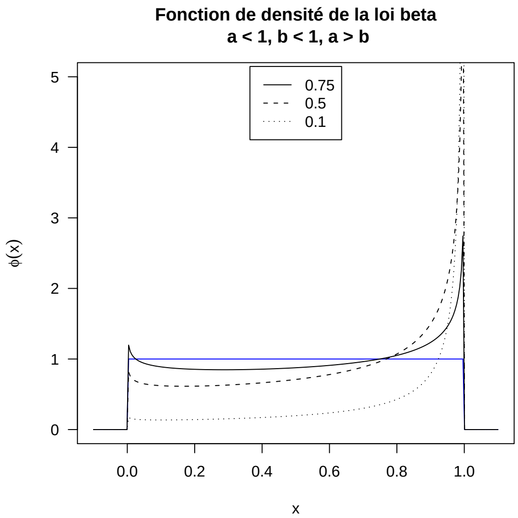
Quand $a = b$, la fonction de densité de la loi beta est en U :



Quand $a < b$, la fonction de densité de la loi beta prend progressivement la forme d'une courbe en L :

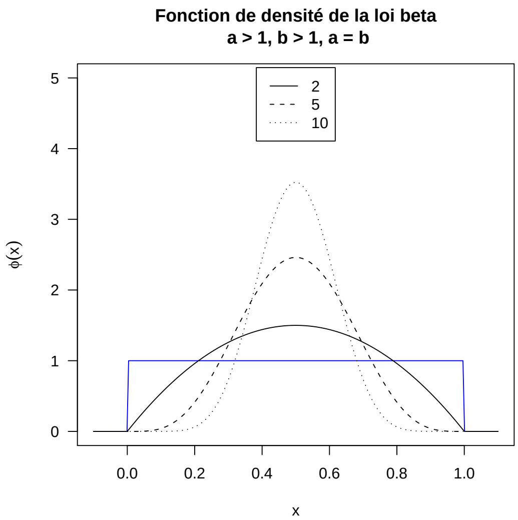


Quand $a > b$, la fonction de densité de la loi beta prend progressivement la forme d'une courbe en J :

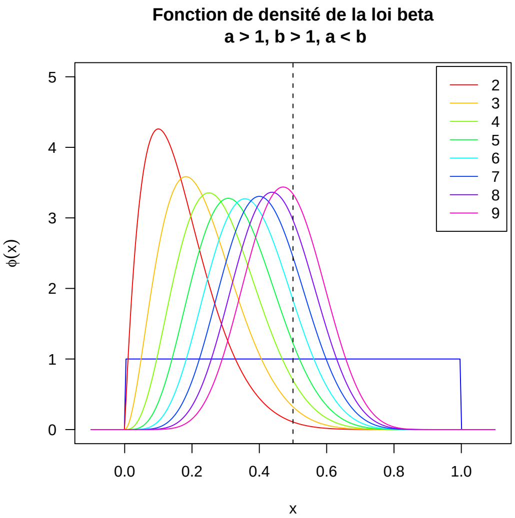


11.3 $a > 1, b > 1$

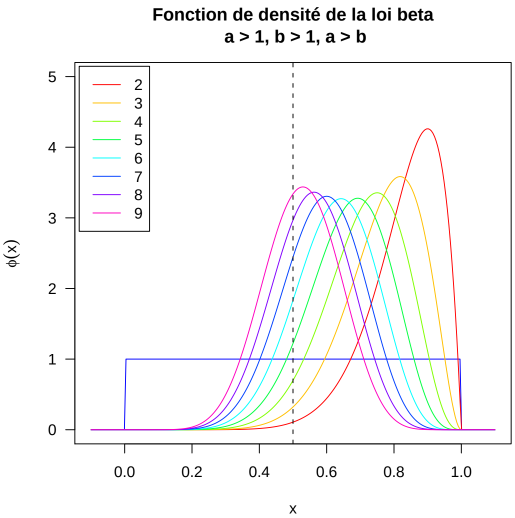
Quand $a > 1, b > 1$, la fonction de densité de la loi beta est unimodale, elle prend progressivement la forme d'une courbe en cloche, symétrique par rapport à 0.5 quand $a = b$:



Quand $a < b$, le mode est inférieur à 0.5 :

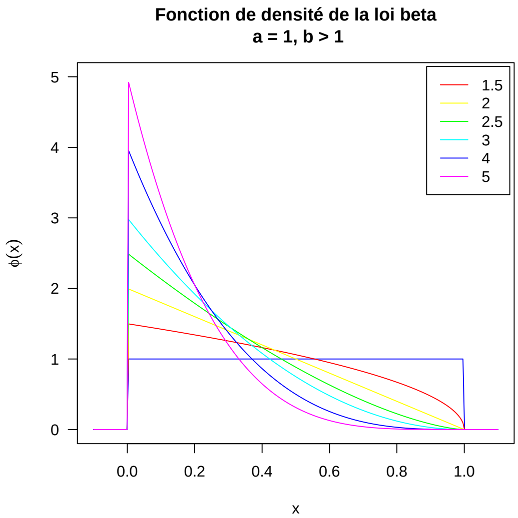


Quand $a > b$, le mode est supérieur à 0.5 :



11.4 $a < 1, b \geq 1$ ou $a = 1, b > 1$

Dans ce cas la fonction de densité est strictement monotone décroissante, par exemple :



11.5 $a > 1, b \leq 1$ ou $a = 1, b < 1$

Dans ce cas la fonction de densité est strictement monotone croissante, par exemple :