

Révision - Code R

Voici des exemples de calculs en R pour les mesures suivantes:

- Fonction de masse (densité) de probabilité
- Fonction de répartition
- Espérance
- Variance
- Espérance tronquée
- Fonction *stop-loss*
- Fonction quantile (mesure *VaR*)
- Mesure *TVaR*

On indique en commentaire la provenance des résultat.

```
#### Cas discrets
##
## Loi qu'on ne connaît pas
##
## ex.:  $P(x) = 0.3 + 0.15s^{100} + 0.4s^{400} + 0.1s^{1000} + 0.05s^{2000}$ 
##

k <- c(0, 100, 400, 1000, 2000)
fx <- c(0.3, 0.15, 0.4, 0.1, 0.05)

Esp <- sum(k * fx)           # Définition de l'espérance
Esp2 <- sum(k^2 * fx)        # Définition du second moment

Var <- Esp2 - Esp^2          # Formule pour la variance
# ou
Var <- sum((k - Esp)^2 * fx)  # Définition de la variance

Fx <- cumsum(fx)             # Définition de la répartition

Esstronc <- function(x) sum(k * fx * (k > x))    # Définition

SL <- function(d) sum(pmax(k - d, 0) * fx)        # Définition

VaR <- Fxinv <- function(u) k[min(which(Fx >= u))] # Définition

TVaR <- function(k) VaR(k) + SL(VaR(k))/(1 - k)  # Formule avec la SL
```

```

##
## Loi qu'on connait
##
## ex.: X suit une loi Binom(n = 100, q = 0.05)
##

n <- 100
k <- 0:n
q <- 0.05

fx <- dbinom(k, n, q) # Utiliser la fonction R d<loi>

Esp <- sum(k * fx)      # Vérification Esp == n * q TRUE

Fx <- pbinom(k, n, q)   # Utiliser la fonction R p<loi>

Esptronc <- function(x) sum(k * fx * (k > x)) # Définition

SL <- function(d) sum(pmax(k - d, 0) * fx)      # Définition

VaR <- Fxinv <- function(u) qbinom(u, n, q)      # Utiliser la fonction R
q<loi>

TVaR <- function(k) VaR(k) + SL(VaR(k))/(1 - k) # Formule avec la SL

#### Cas continu
##
## Loi qu'on connait
##
## ex.: X suit une loi Gamma(a = 1/2, b = 3)
##

al <- 1/2
be <- 3

fx <- function(x) dgamma(x, al, be)      # Utiliser la fonction R d<loi>

Esp <- al/be      # Formule de l'espérance dans l'annexe
Var <- al/be^2    # Formule de la variance dans l'annexe

Fx <- fonction(x) pgamma(x, al, be)      # Utiliser la fonction R p<loi>

Esptronc <- function(d) al/be * pgamma(d, al + 1, be, lower.tail = FALSE)

```

```

# Voir la formule dans l'annexe

SL <- function(d) Esptronc(d) - d * pgamma(d, al, be, lower.tail = FALSE)
# Voir la formule dans l'annexe

VaR <- Fxinv <- function(u) qgamma(u, al, be) # Utiliser la fonction R
q<loi>

TVaR <- function(k) al/be * pgamma(VaR(k), al + 1, be, lower.tail =
FALSE)/(1 - k)
# Voir la formule dans l'annexe


### Loi de mélange discret
##
## ex.: mélange de deux lois poissons
##

p <- c(0.2, 0.8)
al <- c(1, 4)

k <- 0:100          # Pas le vrai support, mais suffisant pour nos calculs

fx <- function(x) p[1] * dpois(x, al[1]) + p[2] * dpois(x, al[2])      #
d<loi>

Fx <- function(x) p[1] * ppois(x, al[1]) + p[2] * ppois(x, al[2])      #
p<loi>

SL <- function(d) sum(pmax(k - d, 0) * fx(k))          # Définition

VaR <- function(u) k[min(which(Fx(k) >= u))]          # Définition

TVaR <- function(k) VaR(k) + SL(VaR(k))/(1 - k)      # Formule avec la SL


### Loi de mélange continu
##
## ex.: mélange de deux lois exponentielles
##

p <- c(0.2, 0.8)
be <- c(1/10, 1/60)

```

```
fx <- function(x) p[1] * dexp(x, be[1]) + p[2] * dexp(x, be[2])      # d<loi>

Fx <- function(x) p[1] * pexp(x, be[1]) + p[2] * pexp(x, be[2])      # p<loi>

VaR <- function(u) optimize(function(x) abs(Fx(x) - u), c(0, 500))$min
# Optimisation
```