Optimize

Jérémie Barde

07 October 2022

Résumé

Le présent document contient des exemples de codes R sur les fonctions de probabilité incluses dans R et, plus précisément, la fonction `optimize`.

Table des matières

1 Lois de probabilité et fonctions R **2**

2 Fonction `optimize` **3**

2.1 Exemple 1 3

2.2 Exemple 2 4

2.3 Exemple 3 4

2.4 Exemple 4 5

1 Lois de probabilité et fonctions R

R contient la majorité des lois que nous voyons au baccalauréat. Avec les fonctions suivantes, il est possible d'évaluer certaines quantités de base directement. Voici un exemple avec $X \sim Exp\left(\frac{1}{10}\right)$. Trouver $\Pr(X \leq 5)$:

```
pexp(5, 0.1)
```

```
## [1] 0.3934693
```

Trouver $\Pr(X \geq 5)$:

```
# Deux choix possibles
pexp(5, 0.1, lower.tail = FALSE)
```

```
## [1] 0.6065307
```

```
1 - pexp(5, 0.1)
```

```
## [1] 0.6065307
```

Trouver $F_X^{-1}(0.45)$:

```
qexp(0.45, 0.1)
```

```
## [1] 5.97837
```

Trouver $E[X]$, $Var(X)$ et $E[X^3]$:

```
EX <- mexp(1, 0.1)
VarX <- mexp(2, 0.1) - (mexp(1, 0.1))^2
EX3 <- mexp(3, 0.1)
cbind(EX, EX3, VarX)
```

```
##      EX  EX3 VarX
## [1,] 10 6000  100
```

Trouver $\pi_x(5)$ et $E\left[X \times 1_{\{X>5\}}\right]$ et $E\left[X \times 1_{\{X<5\}}\right]$, on sait :

$$\begin{aligned} E[X] &= E[\max(X-d,0)] + E[\min(X,d)] \\ \pi_X &= E\left[X \times 1_{\{X>d\}} - d\bar{F}_X(d)\right] \\ E[X] &= E\left[X \times 1_{\{X>d\}}\right] + E\left[X \times 1_{\{X<d\}}\right] \end{aligned}$$

```
SL <- mexp(1, 0.1) - levexp(5, 0.1)
EXTrD <- SL + 5 * pexp(5, 0.1, lower.tail = FALSE)
EXTrU <- mexp(1, 0.1) - EXTrD
cbind(SL, EXTrD, EXTrU)
```

```
##            SL   EXTrD     EXTrU
## [1,] 6.065307 9.09796 0.9020401
```

On peut aussi vouloir simuler des réalisations de X :

```
set.seed(2022)
rexp(5, 0.1)
```

```
## [1]  6.319553  2.945187 22.074866  2.715817 22.682255
```

Exemple pour $X \sim Po(5)$:

```
# P(X = 3), on utilise dpois
dpois(3, 5)
```

```
## [1] 0.1403739
```

On peut utiliser la rubrique d'aide de R pour avoir plus d'informations sur les fonctions et leurs paramètres.

2 Fonction `optimize`

Il n'est pas toujours simple d'inverser une fonction de répartition à la main. Dans le cas de la loi exponentielle, on trouve facilement :

$$F_X^{-1}(u) = -\frac{1}{\lambda}\ln(1-u)$$

Par contre, pour une loi Erlang, il n'est pas possible de le faire à la main. On a donc recours à la fonction `qgamma`. Cependant, ce ne sont pas toutes les lois qui sont programmées en R.

2.1 Exemple 1

On a la fonction de répartition suivante :

$$F_X(x) = 0.85 \times \left(1 - e^{-0.1x}\right) + 0.1 \times \left(1 - e^{-0.5x}\right) + 0.05 \times \left(1 - e^{-0.8x}\right)$$

Ce qu'on cherche, c'est :

$$F_X(x) = u$$
$$F_X(x) - u = 0$$

On veut $F_X(x)$ pour que l'équation soit égale à 0. Il faut donc avoir recours à un outil d'optimisation pour inverser cette fonction de répartition; on utilisera `optimize`.

```r
P <- c(0.85, 0.1, 0.05)
lam <- c(0.1, 0.5, 0.8)
k <- 0.99

Fx <- function(x) {
    P[1] * (1 - exp(-lam[1] * x)) + P[2] * (1 - exp(-lam[2] * x)) + P[3] * (1 - exp(-lam[3] *
        x))
}

# En reconnaissant les exponentielles
Fx <- function(x) {
    P[1] * pexp(x, lam[1]) + P[2] * pexp(x, lam[2]) + P[3] * pexp(x, lam[3])
}

Fx(20)
```

```
## [1] 0.8849605
```

```r
Fx(300)
```

```
## [1] 1
```

```r
Fx_inver <- function(u) optimize(function(x) abs(Fx(x) - u), c(0, 300))$min

# Vérification
verif <- cbind(c(Fx_inver(k), Fx(Fx_inver(k))))
row.names(verif) <- c("Fx_inver", "Fx(Fx_inver)")
colnames(verif) <- "Résultats"
round(verif, 4)
```

```
##                Résultats
## Fx_inver       44.4265
## Fx(Fx_inver)    0.9900
```

2.2 Exemple 2

On a la fonction de répartition suivante :

$$F_Y(y) = 0.4 \times H(y; 3, 0.1) + 0.6 \times B(y; 4, 5)$$

Encore une fois impossible à la main et aucune fonction préprogrammée en R, on doit utiliser `optimize`.

```
G <- c(3, 0.1)
B <- c(4, 5)
k <- 0.99

Fy <- function(y) 0.4 * pgamma(y, G[1], G[2]) + 0.6 * pbeta(y, B[1], B[2])

Fy(30)
```

```
## [1] 0.830724
```

```
Fy(300)
```

```
## [1] 1
```

```
Fy_inver <- function(u) optimize(function(y) abs(Fy(y) - u), c(0, 300))$min

# Vérification
verif <- cbind(c(Fy_inver(k), Fy(Fy_inver(k))))
row.names(verif) <- c("Fy_inver", "Fy(Fy_inver)")
colnames(verif) <- "Résultats"
verif
```

```
##                Résultats
## Fy_inver       72.24688
## Fy(Fy_inver)    0.99000
```

2.3 Exemple 3

On a $X \sim BinComp(n = 2, q = 0.3, F_B)$, $B \sim Exp(\frac{1}{10})$, dont la fonction de répartition est :

$$F_X(x) = \Pr(N = 0) + \Pr(N = 1)F_{B_1} + \Pr(N = 2)F_{B_1+B_2}(x)$$

```
n <- 2
q <- 0.3
lam <- 0.1
k <- 0.99
Fv <- function(x) {
    dbinom(0, n, q) + dbinom(1, n, q) * pexp(x, lam) + dbinom(2, n, q) * pgamma(x,
        2, lam)
}

Fv(25)
```

```
## [1] 0.9396675
```

```r
Fv(100)
```

```
## [1] 0.999936
```

```r
Fv_inver <- function(u) optimize(function(x) abs(Fv(x) - u), c(0, 100))$min

# Vérification
verif <- cbind(c(Fv_inver(k), Fv(Fv_inver(k))))
row.names(verif) <- c("Fv_inver", "Fv(Fv_inver)")
colnames(verif) <- "Résultats"
verif
```

```
##              Résultats
## Fv_inver      45.18121
## Fv(Fv_inver)   0.99000
```

2.4 Exemple 4

On a $X \sim PoComp(\lambda = 1, F_B),\ B \sim Exp(\frac{1}{10})$, dont la fonction de répartition est :

$$F_X(x) = \Pr(X = 0) + \sum_{k=1}^{\infty} \Pr(X = k) H(x, k, 0.1)$$

```r
lam <- 1
b <- 0.1
k <- 0.99

Fn <- function(x) {
    dpois(0, lam) + sum(dpois(1:100, lam) * pgamma(x, 1:100, b))
}
sum(dpois(0:100, 1))
```

```
## [1] 1
```

```r
Fn(500)
```

```
## [1] 1
```

```r
Fn_inver <- function(k) optimize(function(x) abs(Fn(x) - k), c(0, 500))$min

# Vérification
verif <- cbind(c(Fn_inver(k), Fn(Fn_inver(k))))
row.names(verif) <- c("Fn_inver", "Fn(Fn_inver)")
colnames(verif) <- "Résultats"
verif
```

```
##              Résultats
## Fn_inver      61.77124
## Fn(Fn_inver)   0.99000
```