

Devoir 4 : Mon premier package

Patrick Fournier

30 septembre 2022

L’objectif de ce devoir est la création d’un package R. Celui-ci devra respecter les exigences suivantes :

- Chacune des fonctions doit être documentée. La documentation doit comprendre :

Titre : Description en une ligne de la fonction.

Description : Description plus détaillée de la fonction.

Paramètres : Description de chacun des arguments acceptés par la fonction.

Retour : Description de ce qui est retourné par la fonction.

Exemple : Au moins un exemple d’utilisation de la fonction.

- Le package lui-même doit être documenté.
- Votre package doit être mis en ligne sur GitHub de manière à pouvoir être installé à l’aide de la fonction `devtools::install_github`.
- *Notez que chacune des fonctions disponibles doit être fonctionnelle.*
- *Votre package sera testé à l’aide de la commande R `CMD check`.* Des pénalités seront attribuées pour chaque “erreur” et chaque “avertissement” (“warning”). Assurez-vous donc de passer chacun des checks avant la date limite.

L’objectif de votre package est d’implémenter la class S3 `melange_dist` représentant un mélange de distributions. Bien entendu, afin que votre classe soit utile, il sera nécessaire de définir des méthodes et des fonctions génériques. De plus, afin de la rendre facile à utiliser, vous devrez implémenter un constructeur et documenter l’ensemble des objets exportés.

Exercice 1

Définissez un constructeurs pour la classe `melange_dist`. Celui-ci devra prendre les 2 arguments suivants :

1. Une liste de fonctions. Chacune doit être une fonction *d’un seul argument*, disons `n`, permettant de générer `n` réalisations d’une variable aléatoire ;
2. Un vecteur contenant les poids du mélange. N’oubliez pas de le normaliser à 1 !

De plus, un argument optionel devra permettre à l’utilisateur de donner une brève description. À titre d’exemple, les appels suivants devraient retourner une instance :

```
melange_dist(list(rnorm, rexp, \n) rgamma(n, 10)), c(0.1, 0.2, 0.7))
melange_dist(list(\n) rnorm(n, 10, 5), \n) rnorm(n, 3, 2)), c(3, 4))
melange_dist(list(\n) rexp(n, 10), \n) rexp(n, 20)),
              c(3, 4),
              "Mélange de deux exponentielles")
```

Les appels ci-dessous devraient mener à une erreur :

```
melange_dist(list(rnorm, rexp, \n) rgamma(n, 10)), c(0.1, 0.9))
melange_dist(list(rnorm, rexp, gamma), c(0.1, 0.2, 0.7))
```

Exercice 2

Implémentez une méthode `print` pour votre classe. Celle-ci devrait retourner des informations utiles sur l’instance sur laquelle elle est appelée et ce, de manière concise. Laissez aller votre créativité !

Exercice 3

Implémentez une méthode `rand` pour `melange_dist`. En plus d’une instance, celle-ci devra prendre un argument `n` et retourner un échantillon de taille `n` provenant du mélange représenté par l’instance.

Exercice 4

Implémentez une méthode `plot` fournissant une représentation graphique de la densité correspondant à une instance. Pour ce faire, la méthode devra générer un échantillon “représentatif” de la distribution, estimer sa densité (voir `density`) et faire un graphique. Notez que l’échantillon est pour usage interne uniquement : il ne doit pas être retourné.

Exercice 5

En utilisant la même stratégie que pour l’exercice précédent, implémentez les méthodes `mean` et `quantile`.

Exercice 6

Définissez une nouvelle classe `melange_normal`. Le constructeur de celle-ci devra prendre deux arguments obligatoires et un argument optionel. Les deux arguments obligatoires sont les suivants :

1. Un vecteur de moyennes ;
2. Un vecteur d’écart-types.

Comme pour la classe `melange_dist`, l’argument optionel sert à permettre à l’utilisateur de fournir une courte description. N’oubliez pas de valider les arguments fourni par l’utilisateur !

Par ailleurs, *toutes* les méthodes définies aux exercices précédents pour `melange_dist` doivent fonctionner pour `melange_normal`. Codez intelligemment et vous n’aurez pas à travailler en double !

Exercice 7

Finalement, écrivez :

- 5 tests pour le constructeur de `melange_dist`
- 5 tests pour le constructeur de `melange_normal`
- 1 test pour votre méthode `plot`