

Notes sur les mises à jour

Les numéros de référence entre parenthèses peuvent être insérés dans l'ouvrage à l'endroit indiqué.

Les numéros indiqués en bleu signalent que les modifications ont pu être effectuées dans l'ouvrage disponible en ligne (NOTO).

Corrigenda

(C1) 20180202 page 19: Changement de version du *package coin*. En milieu de page, remplacer "*coin_1.2-1.zip*" par "*coin_1.2-2.zip*", et "*coin_1.2-1.tgz*" par "*coin_1.2-2.tgz*".

(C2) 20180914 page 927 section `uplicated()`

Remplacer : "Attention ! `uplicated()` n'indique pas les positions des éléments uniques. Pour connaître les positions uniques dans le vecteur `x`, on peut écrire `x %in% x[uplicated(x)]`"

Par :

"Pour extraire ce qui est unique, voir la fonction `unique()`, ou employer `x[! duplicated(x)]`. Pour connaître ce qui est répété dans le vecteur `x`, on peut écrire `x %in% x[duplicated(x)]` et `! x %in% x[duplicated(x)]` pour ce qui n'est jamais répété"

(C3) 20180914 page 930 section `unique()`

Remplacer la phrase incorrecte : "Pour connaître les positions uniques dans le vecteur `x` : `x %in% x[duplicated(x)]`"

Par :

"Voir également la fonction `duplicated()`, pour connaître ce qui est répété et non répété dans un vecteur"

(C4) 20190502 page 906

Remplacer:

`\\s` un espace quelque part

`\\S` tout sauf une chaîne de caractères formée uniquement d'espaces et rien d'autre

Par :

`\\s` un caractère "blanc" quelque part: espace, tabulation (`\\t`), fin de ligne (`\\n`) ou retour chariot (`\\r`)

`\\S` tout sauf une chaîne de caractères formée uniquement de caractères "blancs" et rien d'autre

(C5) 20190502 page 906

Remplacer:

`\\t` identifie strictement `\\t` dans une chaîne de caractères, soit l'introduction d'une tabulation (voir l'aide `?Quotes`)

\\n voir \\t
 \\r voir \\t

Par :

\\t identifie strictement \t dans une chaîne de caractères, soit la présence d'une tabulation (voir l'aide ?Quotes)
 \\n identifie strictement \n dans une chaîne de caractères, soit la présence d'une fin de ligne (voir l'aide ?Quotes)
 \\r identifie strictement \r dans une chaîne de caractères, soit la présence d'un retour chariot (équivalent de fin de ligne mais pas le même caractère, voir l'aide ?Quotes)

(C6) 20190502 page 916

Remplacer:

Mais cela ne fonctionne que dans l'intervalle de valeurs +/- 2³¹, soit +/- 2 147 483 648.
 En effet, R utilise le format 32 bits pour coder les entiers (1 bit pour le signe + ou -, et donc 2³¹ combinaisons de 0 et 1 pour le nombre entier)

Par :

Mais cela ne fonctionne que dans l'intervalle de valeurs +/- 2³¹ - 1, soit +/- 2 147 483 647. En effet, R utilise le format 32 bits pour coder les entiers (1 bit pour le signe + ou -, et donc 2³¹ - 1 combinaisons de 0 et 1 pour le nombre entier)

(C7) 20191022 page 872-873

Remplacer:

$$SCE = Y'Y - C_{b,a} X'Y$$

$$SCT = Y'Y - C_{red} X'_{red} Y$$

Par :

$$SCE = Y'Y - (C_{b,a})' X'Y$$

$$SCT = Y'Y - (C_{red})' X'_{red} Y$$

Addenda

(A1) 20190502

Une mise à jour des *packages* de la version 3.3.3 de R, postérieure à la publication du livre, semble avoir généré quelques problèmes d'installation. Ainsi, le *package coin* ne peut être installé que de façon manuelle (comme expliquée page 19 du livre). L'installation par le menu de R ne fonctionne pas. Par ailleurs, l'installation de *ggplot2* n'est plus possible, à cause de la version 0.2.1 du *package* dépendant *rlang* qui n'est plus disponible. Tous ces problèmes semblent inexistantes dans les versions récentes de R.

(A2) 20191022 fin de page 917

Pour terminer, noter que l'instruction `.Machine` affiche toutes les limites numériques de l'environnement de travail (utiliser `?Machine` pour l'explication des valeurs). Par ailleurs, pour

contrôler les problèmes de virgule flottante dans un test d'égalité, il est préférable d'employer

`isTRUE(all.equal())` plutôt que `==`. Exemple :

```
> 0.3 / 3 == 0.1
```

```
[1] FALSE
```

```
> isTRUE(all.equal(0.3 / 3, 0.1))
```

```
[1] TRUE
```