

Evaluation des dossiers sur la Programmation en R

Les « 12 travaux d'AstéRix »

Daif Hakim

31/01/2021

Package Lubridat

1. Critères d'évaluation

Dans ce dossier, nous allons parcourir et faire une évaluation des travaux de mes camarades sur la Programmation en R:

Nos critères seront :

1. Critères d'évaluation

Dans ce dossier, nous allons parcourir et faire une Auto-evaluation de notre travail sur la Programmation en R qui porte que la P-Value.

Nos critères seront :

1. Visuel et l'organisation du dossier: Es ce que la mise en page est propre ? Le travail est-il organisé (chapitres, sections, sous-sections, Parties) ?

2. Qualité de la rédaction du dossier: le document est-il agréable à lire? es ce que l'auteur a respecté les consignes de rédaction d'un article scientifique ou d'une thèse... ?

3. Compréhension de l'idée générale:Les Auteurs arrivent-ils à transmettre l'idée générale ?

4. lisibilité et fonctionnement du code (explication et commentaires): Es ce que le code fonctionne parfaitement sans erreurs ? Es ce que l'auteur a mis des commentaires pour expliquer chaque ligne de commande ?

5. L'utilisation d'exemples et de cas concrets (cas pratique): l'Auteur a t'il utilisé des exemples et des cas pratique lors de sa rédaction ?

2. Lien vers le document commenté

Pour ce dossier, nous avons étudié le travail de Gaspard Palay qui porte sur le Package Lubridat.

le lien (github) vers son dossier est :

<https://github.com/GaspardPalay/PSBX>

Le travail en question est le dossier **TutoLubridate.pdf**

3. Synthèse du document

Ce document présente le package : Lubridate.

Celui-ci a été spécialement conçu pour gérer les données de temps (Date) et les rendre facilement manipulables. Gaspard compte nous expliquer comment travailler les données indiquant le temps, avec des dates, ou des heures (et des minutes et des secondes), ou les deux à la fois.

Tout d'abord l'auteur nous expose dans son introduction les problèmes récurrents qu'on rencontre lors de manipulation de ces données, par exemple le fait que les dates ne sont pas toujours écrites de la même façon.

Il va principalement travailler sur le format date et le format POSIXct

lubridate est disponible avec la collection de package TidyVerse en open source, à cet effet, l'auteur nous explique comment installer la totalité du package tidyverse

Gaspard commence directement à nous expliquer l'une des fonctionnalités les plus importantes de ce package qui consiste en la transformation d'une chaîne de caractère en date ou bien en heure.

Ensuite, Gaspard va nous montrer comment manipuler les dates et les heures. On va voir comment arrondir une date ou bien comment calculer une période de temps écoulé entre deux dates.

Après cela, il va nous montrer comment calculer des intervalles entre deux dates,

Gaspard consacrera sa dernière partie pour nous montrer ce qu'est l'heure POSIX (aussi appelé l'heure Unix).

L'heure POSIX est une mesure de temps qui se base sur le nombre de seconde écoulé depuis le premier Janvier 1970. Grâce aux fonctions POSIX, il est possible d'obtenir une date ou heure extrêmement précise sur cet intervalle de temps. On peut aussi vérifier qu'un objet possède bien le type date ou bien renvoyer à l'heure ou la date actuelle sous forme POSIX.

4. Extrait commenté des parties de code

Nous allons maintenant étudier le code de Gaspard.

```
class("05 may 2020")
```

On voit que cet élément est une chaîne de caractère, Gaspard va donc nous montrer comment transformer cela en date, en format day, month and year:

```
jourJ <- lubridate::dmy("30 may 2020")  
class(jourJ)
```

Maintenant, cette chaîne de caractère est devenue une date. Pour continuer, nous allons voir des vecteurs qui regroupent la date et l'heure :

```
#J'ai ajouté le library(lubridate) pour faire fonctionner  
library(lubridate)  
ymd("2019/04_11")  
ymd_hm("2019.04.11 14h37")  
ymd_hms("20190407143752")  
hms("14h37min52s")
```

Maintenant, ces dates et heures sont lisibles par tout le monde. Elles sont compréhensibles.

Gaspard nous montre aussi comment arrondir une date. Dans cet exemple, on va arrondir vers le plus proche.

```
t <- ymd_hms("2020.11.09_17.56.32")
round_date(t, "hour")
```

Pour finir, voilà comment calculer le temps écoulé entre deux dates grâce à lubridate :

```
t1 <- dmy("12/09/2020")
t2 <- dmy("30/01/2016")
diff <- t1-t2
as.duration(diff)
as.period(diff)
```

On voit que le résultat apparaît en années (4,62) et en jour (1687).

5. Evaluation du travail suivant les 5 critères précités

- 1. Visuel et l'organisation du dossier :** c'est un dossier très propre, Visuel agréable, organisation très bonne rien qu'à travers les titres (sections et parties) on peut déjà tirer une idée générale de son travail
- 2. Qualité de la rédaction du dossier :** Des phrases simples, de courtes lignes introductives de chaque partie sont rédigées, agréables à lire.
- 3. Compréhension de l'idée générale :** Gaspard semble avoir compris l'idée générale du package Lubridate, il nous a fait de bons exemples et il a bien détaillé les fonctionnalités les plus importantes.
- 4. Lisibilité, explication (commentaires) et fonctionnement du code :** le code est parfaitement lisible, L'exécution du RMarkdown se réalise parfaitement aussi, les commandes sont très bien expliquées, on trouve aussi des commentaires très utiles pour mieux comprendre l'enchaînement des commandes, et c'est très bien fait.
- 5. L'utilisation d'exemples et de cas concrets (cas pratique) :** on trouve des exemples simples et vu la simplicité de ce thème (sujet non compliqué, à la portée de tout le monde), donc c'est largement suffisant

6. Conclusion

Cet article a été bien agréable et rapide à lire. C'est un outil très utile car les données de temps sont extrêmement importantes en analyse de données.

Le document est très bien réalisé avec des expressions très simples, une rédaction de qualité, après tout la non-complexité de ce package a sûrement facilité le travail de l'auteur, néanmoins il est impossible de nier l'importance de ce package dans le traitement des données ou dans le monde de data.