

# Analyse de Lubridate par Gaspard Palay

Wenjun ZHAO

12/20/2020

## I. Introduction

Vous pouvez retrouver mon travail sur mon Github, <https://github.com/claudia0524/PSBX>.

Pour ce dossier, nous avons étudié le travail de Gaspard Palay, étudiant en M2DM au sein de PSB.

Nous avons trouvé son dossier sur son Github : <https://github.com/GaspardPalay/PSBX/tree/main/TutorielLubridate>

Maintenant, nous allons établir mes 5 critères d'évaluations, qui seront les mêmes pour tout les dossiers que j'ai vus étudier pour ce devoir:

### Les cinq critères d'évaluation

1. *Rmd se comporte bien à l'exécution*
2. *Les aspects intéressants, didactiques, complets*
3. *La qualité Rmarkdown, la qualité de l'écriture*
4. *Didactique, conformité aux exigences vues plus haut et comporte du calcul symbolique et du calcul numérique*
5. *La qualité du LaTeX et des illustrations, la qualité de l'écriture, le choix des ressources internet, la compréhension personnelle des concepts*

## II. Synthèse du travail en question

Si vous avez déjà dû travailler sur des fichiers comportant des données indiquant le temps, avec des dates, ou des heures (et des minutes et des secondes), ou les deux à la fois, vous savez que ce n'est pas toujours facile à gérer. D'abord, parce que les dates ne sont pas toujours écrites de la même façon. Par exemple, pour le premier mars 2019, on peut trouver "2019-03-01" ou encore 1-mars-2019, ou encore 01/03/2019, etc... La même remarque peut être faite pour les heures, les minutes et les secondes. Et ensuite, parce qu'il faut convertir ces données, qui sont généralement considérées comme des chaînes de caractères, ou des modalités d'une variable catégorielle à l'issue de l'importation dans R, dans un format (ou classe) reconnu comme étant du temps dans R. Ces formats (ou classes) sont notamment le format **POSIXct** et le format **Date**.

Une fois que ces données indiquant le temps, sont correctement reconnues par R, il est alors possible de les manipuler, de réaliser des analyses descriptives (par année, mois, jour, heure, etc...), de les représenter visuellement, ou encore de réaliser des calculs. Le package lubridate a été spécialement conçu pour gérer ces données de temps et les rendre facilement manipulables.

Nous allons maintenant étudier le code de Gaspard Palay dans le chapitre suivant.

## III. Extrait commenté des parties de code

si vous souhaitez installer la totalité du package tidyverse (très utile) prenez la fonction suivante(Palay, n.d.):

```
install.packages("tidyverse")
```

ou simplement le package

```
install.packages("lubridate")
```

## 1.1 Transformer une chaîne de caractères en date

### 1.1.1 transformation d'un vecteur de classe chaîne de caractères en un vecteur de classe date

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

05 Mai 2020 est donc une chaîne de caractère, je vais lui indiquer que les éléments composants cette chaîne de caractère sont des Jours des Mois et des Années (DMY)

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

```
## [1] "Date"
```

grâce au package Lubridate R transforme la chaîne de caractère en date, si j'exécute jourJ il le transformera :

```
jourJ
```

```
## [1] "2020-05-30"
```

### 1.1.2 transformation d'un vecteur de classe chaîne de caractères en un vecteur de classe heure

quand on a des vecteurs qui regroupent à la fois la date et l'heure (comme "11/04/2019 14h37" par exemple) on parle de "date-time". Plusieurs fonctions sont disponibles dans le package :

```
ymd("2019/04_11")  
ymd_hm("2019.04.11 14h37")  
ymd_hms("20190407143752")  
hms("14h37min52s")
```

Lubridate nous permet aussi de récupérer seulement certains éléments compris dans une date grâce aux plusieurs fonctions suivantes :

```
t <- ymd_hms("2020.11.09_17.56.32")  
date(t)  
hour(t)  
minute(t)  
second(t)
```

## 1.2 Manipulation des dates/heures

### 1.2.1 arrondir une date

Arrondir vers le haut avec la fonction (**ceiling\_date()**),

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

vers le bas (**floor\_date()**):

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

ou vers le plus proche (**round\_date()**):

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

choisir à quelle unité se fait cet arrondi en précisant ainsi:

```
t <- ymd_hms("2019.04.11 14h37min52s")
round_date(t, "minute")
round_date(t, "hour")
round_date(t, "day")
round_date(t, "month")
round_date(t, "year")
```

### 1.2.2 Calcul des périodes de temps écoulés ou de durée

“diff” nous renseigne sur la “différence de temps” entre t1 et t2. Il s’agit d’un objet de classe difftime

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

si l’on veut connaître la date d’un événement dont on connaît la récurrence (tous les 38 ans par exemple) on utilisera la fonction suivante :

```
t0 <- dmy_h("12/09/2020 22h")
t0+years(38)
```

Vous remarquez que le résultat est inexacte. Il prédit un événement le 12/09 sans compter les années bissextiles. Pour obtenir le bon résultat on fait :

```
t0+dyears(38)
```

si vous souhaitez savoir si une année est bissextile, la fonction suivante le permet :

```
leap_year(2016)
leap_year(2017)
```

## 1.3 Calculs arithmétiques

### 1.3.1 calculs arithmétiques sur des périodes ou des durées

Non pouvons inclure des calculs arithmétiques sur des périodes ou des durées. Les périodes correspondent aux fonctions **xxx()** (par exemple **days()** ou **months()**) tandis que les durées correspondent aux fonctions **dxxx()** (par exemple **ddays()** ou **dyears()**)

```
t1+months(9) # t1 + 9 mois
t1+ddays(287) # t1 + exactement 287 jours
ddays(287)/dweeks(1) # combien de semaines (exactement) pour 287 jours?
t2-dweeks(7) # t2 - 7 semaines
```

### 1.3.2 Intervalles

Les intervalles sont des intervalles de temps qui commencent et finissent à des instants bien déterminés (parce qu’ils sont liés à des dates spécifiques).

Les intervalles conservent des informations complètes sur un intervalle de temps. Les fonctions pour travailler avec l’objet intervals sont **is.interval**, **as.interval**, **interval**, **int\_start**, **int\_end**, **int\_shift**, **int\_flip**, **int\_aligns**, **int\_overlaps** et **%within%**.

Les intervalles peuvent également être manipulés avec les fonctions suivantes :

## 1.4 L'Heure POSIX ou l'heure Unix

L'heure Unix ou heure Posix (aussi appelée Unix Timestamp) est une mesure du temps basée sur le nombre de secondes écoulées depuis le 1er janvier 1970 00:00:00 UTC, hors secondes intercalaires. Elle est utilisée principalement dans les systèmes qui respectent la norme POSIX ISO 8601, dont les systèmes de type Unix, d'où son nom. C'est la représentation POSIX du temps.

### 1.4.1 Lubridate et Instants

Lubridate dispose d'un analyseur POSIX très rapide intégré. La plupart des formats `strptime()` et des diverses extensions sont pris en charge

Les instants sont des moments précis du temps. ***Date***, ***POSIXct*** et ***POSIXlt*** sont les trois classes d'objets que la base R reconnaît comme des instants.

Chaque objet `POSIXct` est enregistré comme le nombre de secondes où il s'est produit après le 1970-01-01 00:00:00.

***is.Date()*** vérifie si un objet hérite de la classe `Date`.

```
is.Date(as.Date("2009-08-03"))
```

***is.POSIXt()*** vérifie si un objet hérite des classes `POSIXlt` ou `POSIXct`. ***is.instant()*** vérifie si un objet hérite de l'une des trois classes.

```
is.POSIXt(as.Date("2009-08-03"))
is.POSIXt(as.POSIXct("2009-08-03"))
```

***now()*** renvoie l'heure système actuelle sous la forme d'un objet `POSIXct`.

```
now(tzone = "")
```

***today()*** renvoie la date système actuelle.

```
today(tzone = "")
```

## IV. Evaluation du travail en question

1. *Rmd se comporte bien à l'exécution (4/4)*
2. *Les aspects intéressant, didactique, complet (3/4)*
3. *La qualité Rmarkdown, la qualité de l'écriture (4/4)*
4. *Didactique, conformité aux exigences vues plus haut et comporte du calcul symbolique et du calcul numérique (3/4)*
5. *La qualité du LaTeX et des illustrations, la qualité de l'écriture, le choix des ressource internet, la compréhension personnelle des concepts (3/4)*

## V. Conclusion

En général, ce travail exécuse bien dans l'environnement de R. Et il nous explique clairement comment fonctionner le package `Dplyr` dans un `rmd`, On comprend rapidement le but principal de ce package. L'aspect est intéressant, complet et propre. D'ailleurs, il a expliqué chaque code, on peut les comprendre facilement. Il a présenté des fonctions qui peut être utile pour analyser des données dans les rapports. La qualité `Rmarkdown`, la qualité de l'écriture et La qualité du `LaTeX` sont ainsi bien.

## VI. Bibliographie

Palay, Gaspard. n.d. "Tutoriel Lubridate." <https://github.com/GaspardPalay/PSBX/tree/main/TutorielLubridate>.