

# Dates sous Windows

*Ewen Gallic*

*Novembre 2014*

Lorsque l'on a parlé des dates en R, nous avons rencontré deux problèmes sur un exemple, qui tentait de convertir une chaîne de caractères en objet de mode `Date` :

- le format de date français dans l'exemple donné n'est pas reconnu sous Windows ;
- le format de date anglais non plus.

Ces deux problèmes ont deux origines distinctes. Pour l'exemple en français, il s'agit d'un problème de respect de normes, tandis que le second est juste une histoire de paramètres locaux.

## L'exemple de la date en français

Créons un objet contenant une chaîne de caractères avec une date en français.

```
d_char_fr <- "Mer 04 Fév 2015"  
as.Date(d_char_fr, format = c("%a %d %b %Y"))
```

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

Il s'avère que les conventions d'abréviations ne sont pas respectées ici. En effet, les jours de la semaine, doivent être abrégés comme suit (ce qui est bien respecté sous Windows, mais pas sous Mac OS) :

```
(d <- seq(as.Date("2014-01-01"), as.Date("2014-12-01"), by = "month"))  
[1] "2014-01-01" "2014-02-01" "2014-03-01" "2014-04-01"  
[5] "2014-05-01" "2014-06-01" "2014-07-01" "2014-08-01"  
[9] "2014-09-01" "2014-10-01" "2014-11-01" "2014-12-01"  
format(d, "%b")  
[1] "janv." "févr." "mars" "avr." "mai" "juin" "juil."  
[8] "août" "sept." "oct." "nov." "déc."
```

Alors, comment faire, si on reçoit un document qui contient des dates dans ce format non conventionnel ? Il n'y a pas de solution (à ma connaissance) élégante. Je vous propose néanmoins un moyen de contourner le problème, en créant une fonction qui va remplacer les occurrences des jours abrégés par leur bonne abréviations. Avant de lire plus bas, je vous conseille d'essayer vous même de créer cette fonction, puis de regarder la solution.

Voici la solution que je vous soumetts :

```
# Weekdays  
jours_ab <- c("Lun", "Mar", "Mer", "Jeu", "Ven", "Sam", "Dim")  
jours_ab_correcste <- c("lun.", "mar.", "mer.", "jeu.", "ven", "sam.", "dim.")  
# Months  
mois_ab <- c("jan", "fév", "mar", "avr", "mai", "jui", "jul",  
            "aou", "sep", "oct", "nov", "déc")  
mois_ab_correcste <- c("janv.", "févr.", "mars", "avr.", "mai", "juin", "juil.",  
                    "août", "sept.", "oct.", "nov.", "déc.")
```

```

library(stringr)
# Replaces the incorrect abbreviation by a correct one
# @string:      input character string
# @abbreviation: string vector of abbreviations to replace
# @correct_abb: string vector of corresponding correct abbreviations
replace_abb <- function(string, abbreviation, correct_abb){
  ind <- str_detect(string, ignore.case(abbreviation))
  if(any(ind)){
    ind <- which(ind)
    str_replace(string, ignore.case(abbreviation[ind]), correct_abb[ind])
  }else{
    string
  }
}

```

Ce qui, appliqué à notre objet `d_char_fr`, donne :

```

d_char_fr <- replace_abb(d_char_fr, jours_ab, jours_ab_correcte)
d_char_fr <- replace_abb(d_char_fr, mois_ab, mois_ab_correcte)
d_char_fr

```

```
## [1] "mer. 04 févr. 2015"
```

On peut alors, sous Windows, le convertir à l'aide de la fonction `as.Date()` :

```
as.Date(d_char_fr, format = c("%a %d %b %Y"))
```

```
## [1] "2015-02-04"
```

## Convertir une date anglaise

Lorsque le nom des jours de la semaine ou des mois est écrit en anglais, il suffit de changer les paramètres locaux.

```

d_char <- "Wed 04 Feb 2015"
as.Date(d_char, format = c("%a %d %b %Y"))

```

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

Cela retourne un objet non-disponible (NA), parce que R n'a pas réussi à interpréter correctement la date. Il est alors nécessaire de changer les paramètres locaux de temps. Vous pouvez afficher les paramètres locaux de votre machine à l'aide de l'instruction suivante (vous obtiendrez un résultat différent du mien si vous êtes sous Windows) :

```
Sys.getlocale()
```

```
## [1] "fr_FR.UTF-8/fr_FR.UTF-8/fr_FR.UTF-8/C/fr_FR.UTF-8/fr_FR.UTF-8"
```

Sous Windows, pour changer les paramètres de temps au format américain, il faut évaluer l'instruction suivante :

```
Sys.setlocale("LC_TIME", "English_United States")
```

Sous Mac OS, cela donne :

```
Sys.setlocale("LC_TIME", "en_US")
```

À présent, l'instruction pour convertir l'objet `d_char` en `Date` fonctionne :

```
as.Date(d_char, format = c("%a %d %b %Y"))
```

```
## [1] "2015-02-04"
```

En revanche, on ne peut plus transformer `d_char_fr` :

```
as.Date(d_char_fr, format = c("%a %d %b %Y"))
```

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

Il faut alors changer les paramètres locaux à nouveau, soit en précisant qu'on veut du français :

```
# Windows  
Sys.setlocale('LC_TIME', "French_France")  
# Mac OS  
Sys.setlocale('LC_TIME', "fr_FR")
```

Ou bien restaurer les paramètres par défaut :

```
Sys.setlocale('LC_TIME', "")
```

```
invisible(Sys.setlocale('LC_TIME', ""))
```

```
as.Date(d_char_fr, format = c("%a %d %b %Y"))
```

```
## [1] "2015-02-04"
```

Enfin, on peut noter qu'avec le package `lubridate`, l'opération est moins pénible :

```
library(lubridate)  
  
# Windows  
dmy("mer. 04 févr. 2015", locale = "french_france")  
dmy("Wed 04 Feb 2015", locale = "english_us")  
  
# Mac OS  
dmy("Mer 04 Fév. 2015", locale = "fr_FR")  
dmy("Wed 04 Feb 2015", locale = "en_US")
```