

Correction TP : Manipulation de séries temporelles avec R

Séries Temporelles avec R - Initiation

Anna Smyk, Tanguy Barthelemy

Insee - Département des Méthodes Statistiques



Installation packages

```
install.packages(c("zoo", "xts", "dplyr", "tsibble", "lubridate", "XLConnect"))
```

Loading packages

```
library("dplyr")  
library("zoo")  
library("xts")  
library("tsibble")  
library("tsbox")  
library("imputeTS")  
library("lubridate")
```

Exercice 1: Import de data frame et creation d'objets TS

```
# importer ipi_nace4.csv du repertoire
ipi <- read.csv2(file.path("../..", "Data", "IPI_nace4.csv"))

# quels formatages sont necessaires ?
str(ipi)
ipi$date <- as.Date(ipi$date, format = "%d/%m/%Y")
ipi[, -1] <- sapply(ipi[, -1], as.numeric)
str(ipi)
# creer un objet TS avec la serie RF3030 (attention au start)
y_raw <- ts(ipi[, "RF3030"], start = c(1990, 1), frequency = 12)
#y_raw

### graphique simple en attendant de voir plus sophistique
plot.ts(y_raw)
# # ajout autre courbe
```

Exercice 3: Séquences de dates et création data frames

```
# creer une serie de dates mensuelles entre fevrier 2019 et novembre 2023

d <- seq(from = as.Date("2019-02-01"),
         to = as.Date("2023-11-01"),
         by = "month")

d

# extraire les valeurs de l'année 2020 de la serie Ipi RF3030 (yraw)

y_2020 <- ts_span(y_raw, "2020") # attention faux
y_2020
y_2020 <- ts_span(y_raw, start = "2020-1", end = "2020-12")
y_2020

# creer un data frame serie Ipi RF3030 pour l'année 2020
```

Exercice 4: Serie temporelle avec valeurs ad hoc

Créer une série temporelle mensuelle qui commence en 2000, qui se termine en janvier 2020, qui vaut 1 en avril 2009 et 0 à toutes les autres dates.

Option 1 : utiliser la fonction `window()`

```
ts_zeros <- ts(0, start = 2000, end = 2020, frequency = 12)
window(ts_zeros, start = c(2009, 4), end = c(2009, 4)) <- 1
ts_zeros
```

Option 2 : utiliser `time()` directement

```
ts_zeros <- ts(0, start = 2000, end = 2020, frequency = 12)
#ts_zeros[as.Date(time(y_raw))="2009-04-01"]<-1
#ts_zeros
```

```
indicatrice <- ts(0, start = 2000, end = 2020, frequency = 12)
# Donne un vecteur de booléens
```

Exercice 5: Jointures

```
# créer le TS correspondant à la serie ipi RF1011
y_1011 <- ts(ipi[, "RF1011"], start = c(1990, 1), frequency = 12)
y_1011
end(y_1011)
```

```
# créer un mts avec RF3030 et RF1011 entre 2020 et 2023
y_raw1 <- window(y_raw, start = c(2020, 1), end = c(2023, 12))
y_1011_1 <- window(y_1011, start = c(2020, 1), end = c(2023, 12))
mts_u <- ts.union(y_raw1, y_1011_1)
mts_u
class(mts_u)
```

```
# créer UNE serie ayant les valeurs de RF3030 entre 2010 et 2020
# et les valeurs de RF1011 entre 2021 et la fin de la serie
y_raw1 <- window(y_raw, start = c(2010, 1), end = c(2020, 12))
```

Exercice 6: Taux de variation

```
# ecrire une fonction taux de variation par rapport à la période précédente  
# mensuelle ou trimestrielle
```

```
### package stats
```

```
ev <- function(x) {  
  result <- (x / stats::lag(x, k = -1) - 1) * 100  
  return(result)  
}
```

```
# Ou de manière équivalente :
```

```
ev2 <- function(x) {  
  # Attention ici c'est bien k = 1 dans la fonction diff  
  # et k = -1 dans la fonction lag  
  result <- (diff(x, lag = 1) / lag(x, k = -1)) * 100  
  return(result)  
}
```

Exercice 7: Fonctions statistiques

```
# calculer moyenne/ mediane/ecart type de la serie IPI RF3030
mean(y_raw)
median(y_raw)
sd(y_raw)

# calculer moyenne annuelle / trimestrielle de la serie IPI RF3030
# en utilisant le package xts
# stocker les resultats dans un data frame
library(xts) # voir cheat sheet
moy_an <- apply.yearly(as.xts(y_raw), mean)
moy_an
class(moy_an)
moy_an_mts <- ts_ts(moy_an)
class(moy_an_mts)
df_moy_an <- ts_df(moy_an)
```


Exercice 8: Fonctions statistiques sur tsibble

```
## transformer le mts avec RF3030 et RF1011 entre 2020 et 2023 en tsibble
library(tsibble)
mts_u
tsib_u <- as_tsibble(mts_u)
class(tsib_u)
View(tsib_u) # empilement
table(tsib_u$key)
str(tsib_u)
#format de la date
class(tsib_u$index) # date en year month

# calculer les moyennes timestrielles de RF3030 et RF1011
# Indice: il faut utiliser à la fois group_by_key()` (identifiant série)
# et `index_by` qui permet de regrouper des dates
```

Subsection 1

Exercice 9: Jointures avec tsibble

Exercice 9: Jointures avec tsibble

```
# creer un tsibble (tsi3) contenant les series de l'ipi commençant par RF301
# pas de def de key or index: pas d'emplacement ..
ipi3 <- ipi %>% select(date | starts_with("RF301"))
tsi3 <- as_tsibble(ipi3)
class(tsi3)
View(ipi3)
View(tsi3) ## pas de key, ni d'index
tsi3 <- as_tsibble(ipi3, key = NULL, index = date)
tsi3 <- as_tsibble(ipi3)

# avec `dplyr::full_join()` joindre tsi3 et tsib_u (de l'exercice 8)

# Comparer les résultats avec `dplyr::left_join()` et `dplyr::right_join()`

# Quelle serait l'équivalent de `ts.intersect()` et `ts.union()` ?
```

Exercice 9: Valeurs manquantes

```
serie_avec_NA <- ts(c(rep(NA, 12), rep(0, 24), rep(NA, 24),  
                    rep(1, 24), rep(NA, 12)), start = 2000, frequency = 12)  
serie_avec_NA
```

```
# Reperer les positions des valeurs manquantes
```

```
p_na <- which(is.na(serie_avec_NA))  
p_na
```

```
# Enlever les valeurs manquantes au début de la série
```

```
etape_1 <- zoo::na.trim(serie_avec_NA, sides = "left")  
etape_1
```

```
# Interpoler de manière linéaire les valeurs manquantes entre les 0 et les 1
```