

chap1ex4

Yannick Le Pen

2023-06-26

Chapitre I, exercice 4 : Etude préliminaire du CAC 40

Importation des données

Option 1 : à partir d'un fichier de données

Source : Datastream Datastream est une base de données payante accessible à la BU Dauphine. Elle présente l'avantage de fournir des données complètes. Le séparateur de décimale est le point “.”

```
CAC40_df_1<-read.csv2("C:\\users\\ylepen\\Documents\\COURS dauphine\\econometrie II\\data exercice\\cac40_datastream.csv")
```

Autre méthode possible :

```
masource<-"C:\\users\\ylepen\\Documents\\COURS dauphine\\econometrie II\\data exercice\\"
CAC40_df_1<-read.csv2(file=paste0(masource,"cac40_datastream.csv"),dec=".",col.names = c('Date','CAC40'))
#CAC40_df_1$Date<-as.Date(CAC40_df_1$Date,format = "%d/%m/%y")
```

On convertit la colonne Date au format date Year/Month/Day

```
CAC40_df_1$Date<-as.Date(CAC40_df_1$Date,format = "%d/%m/%Y")
tail(CAC40_df_1$Date)
```

```
## [1] "2023-06-19" "2023-06-20" "2023-06-21" "2023-06-22" "2023-06-23"
## [6] "2023-06-26"
```

```
#library(lubridate)
#CAC40_df_1$Date<-dmy(CAC40_df_1$Date)
```

On vérifie qu'il n'y a pas de données manquantes

```
sapply(CAC40_df_1,function(x) sum(is.na(x)))
```

```
## Date CAC40
##      0      0
```

Les données commencent en “1987-07-09”. On décide d’étudier la série à partir de “1990-01-01” en partie pour avoir le même échantillon que les autres sources de données.

```
CAC40_df_1<-CAC40_df_1[CAC40_df_1$Date>="1990-01-01",]
```

à partir d'un fichier csv provenant de Yahoo Finance Les données de Yahoo contiennent des données manquantes représentées par le terme “null”. Avec R, “null” n’est pas le symbole habituel des données manquantes qui est “na”. Dans l’importation des données à partir de la fonction read.csv, on ajoute l’option na.strings=c(“null”) pour lui indiquer que le symbole des données manquantes dans le fichier ^FCHI.csv.

```
library(readr)
```

```
## Warning: le package 'readr' a été compilé avec la version R 4.2.2
```

```
CAC40_df_2<-read.csv("C:\\users\\ylepen\\Documents\\COURS dauphine\\econometrie II\\data exercice\\`FCH
class(CAC40_df_2)
```

```
## [1] "data.frame"
```

```
str(CAC40_df_2)
```

```
## 'data.frame': 8605 obs. of 7 variables:
## $ Date : chr "1990-03-01" "1990-03-02" "1990-03-05" "1990-03-06" ...
## $ Open : num 1836 1831 1866 1869 1874 ...
## $ High : num 1838 1860 1874 1875 1881 ...
## $ Low : num 1827 1831 1862 1866 1874 ...
## $ Close : num 1832 1860 1874 1872 1880 ...
## $ Adj.Close: num 1832 1860 1874 1872 1880 ...
## $ Volume : int 0 0 0 0 0 0 0 0 0 0 ...
```

Décompte des données manquantes :

```
sapply(CAC40_df_2,function(x) sum(is.na(x)))
```

```
##      Date      Open      High      Low      Close Adj.Close      Volume
##      0         147        147        147        147        147        147
```

Il existe plusieurs manières de gérer les données manquantes. On décide ici de supprimer les dates ayant des données manquantes car elles peuvent empêcher l'exécution de certaines fonctions de R.

```
CAC40_df_2<-na.omit(CAC40_df_2)
sapply(CAC40_df_2,function(x) sum(is.na(x)))
```

```
##      Date      Open      High      Low      Close Adj.Close      Volume
##      0         0         0         0         0         0         0
```

Rappel : Les fonctions class() : affiche la classe d'un objet str() : affiche la structure d'un objet head() : affiche les premières lignes d'un objet tail() : affiche les dernières lignes d'un objet

```
class(CAC40_df_2)
```

```
## [1] "data.frame"
```

```
str(CAC40_df_2)
```

```
## 'data.frame': 8458 obs. of 7 variables:
## $ Date : chr "1990-03-01" "1990-03-02" "1990-03-05" "1990-03-06" ...
## $ Open : num 1836 1831 1866 1869 1874 ...
## $ High : num 1838 1860 1874 1875 1881 ...
## $ Low : num 1827 1831 1862 1866 1874 ...
## $ Close : num 1832 1860 1874 1872 1880 ...
## $ Adj.Close: num 1832 1860 1874 1872 1880 ...
## $ Volume : int 0 0 0 0 0 0 0 0 0 0 ...
## - attr(*, "na.action")= 'omit' Named int [1:147] 32 33 44 49 61 68 120 176 213 214 ...
## ..- attr(*, "names")= chr [1:147] "32" "33" "44" "49" ...
```

```
head(CAC40_df_2)
```

```
##      Date Open High Low Close Adj.Close Volume
## 1 1990-03-01 1836 1838 1827 1832 1832 0
## 2 1990-03-02 1831 1860 1831 1860 1860 0
## 3 1990-03-05 1866 1874 1862 1874 1874 0
## 4 1990-03-06 1869 1875 1866 1872 1872 0
## 5 1990-03-07 1874 1881 1874 1880 1880 0
```

```
## 6 1990-03-08 1891 1923 1891 1917 1917 0
```

```
tail(CAC40_df_2)
```

```
##           Date      Open      High      Low      Close Adj.Close      Volume
## 8600 2023-06-19 7353.31 7362.44 7302.63 7314.05 7314.05          0
## 8601 2023-06-20 7303.24 7322.57 7281.12 7294.17 7294.17 55853400
## 8602 2023-06-21 7273.89 7297.85 7246.86 7260.97 7260.97 48414000
## 8603 2023-06-22 7194.26 7206.89 7145.06 7203.28 7203.28 53704500
## 8604 2023-06-23 7165.38 7193.78 7128.64 7163.42 7163.42 51938700
## 8605 2023-06-26 7175.88 7203.46 7105.72 7184.35 7184.35 52606600
```

Option 2 : Importation des données à partir d'un site internet

```
library(TT)
```

librarie quantmod

```
## Warning: le package 'TT' a été compilé avec la version R 4.2.3
```

```
library(quantmod)
```

```
## Warning: le package 'quantmod' a été compilé avec la version R 4.2.3
```

```
## Le chargement a nécessité le package : xts
```

```
## Warning: le package 'xts' a été compilé avec la version R 4.2.3
```

```
## Le chargement a nécessité le package : zoo
```

```
## Warning: le package 'zoo' a été compilé avec la version R 4.2.3
```

```
##
```

```
## Attachement du package : 'zoo'
```

```
## Les objets suivants sont masqués depuis 'package:base':
```

```
##
```

```
##      as.Date, as.Date.numeric
```

```
##
```

```
## ##### WARNING #####
```

```
## # We noticed you have dplyr installed. The dplyr lag() function breaks how #
```

```
## # base R's lag() function is supposed to work, which breaks lag(my_xts). #
```

```
## # #
```

```
## # If you call library(dplyr) later in this session, then calls to lag(my_xts) #
```

```
## # that you enter or source() into this session won't work correctly. #
```

```
## # #
```

```
## # All package code is unaffected because it is protected by the R namespace #
```

```
## # mechanism. #
```

```
## # #
```

```
## # Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning. #
```

```
## # #
```

```
## # You can use stats::lag() to make sure you're not using dplyr::lag(), or you #
```

```
## # can add conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop #
```

```
## # dplyr from breaking base R's lag() function. #
```

```
## ##### WARNING #####
```

```
## Le chargement a nécessité le package : TTR
```

```
## Warning: le package 'TTR' a été compilé avec la version R 4.2.3
```

```
## Registered S3 method overwritten by 'quantmod':
##   method      from
##   as.zoo.data.frame zoo
```

```
CAC40 <-getSymbols('^FCHI',auto.assign = FALSE,from = '1990-01-01')
```

```
## Warning: ^FCHI contains missing values. Some functions will not work if objects
## contain missing values in the middle of the series. Consider using na.omit(),
## na.approx(), na.fill(), etc to remove or replace them.
```

```
class(CAC40) # an xts object
```

```
## [1] "xts" "zoo"
```

```
CAC40_df<-data.frame(date=index(CAC40),coredata(CAC40))# conversion xts en dataframe avec les dates com
colnames(CAC40_df)<-c("Date","Open","High","Low","Close","Vol","Adj.Close")
sapply(CAC40_df,function(x) sum(is.na(x)))
```

```
##      Date      Open      High      Low      Close      Vol Adj.Close
##      0         147         147         147         147         147         147
```

```
CAC40_df<-na.omit(CAC40_df)
sapply(CAC40_df,function(x) sum(is.na(x)))
```

```
##      Date      Open      High      Low      Close      Vol Adj.Close
##      0          0          0          0          0          0          0
```

```
class(CAC40_df)
```

```
## [1] "data.frame"
```

Utilisation de la librairie yahoofinancer

Il existe aussi la librairie yahoofinancer mais elle ne semble pas capable d'importer des données dès qu'elles contiennent des observations manquantes. On importe ici des données mensuelles et non quotidiennes.

```
library(yahoofinancer)
```

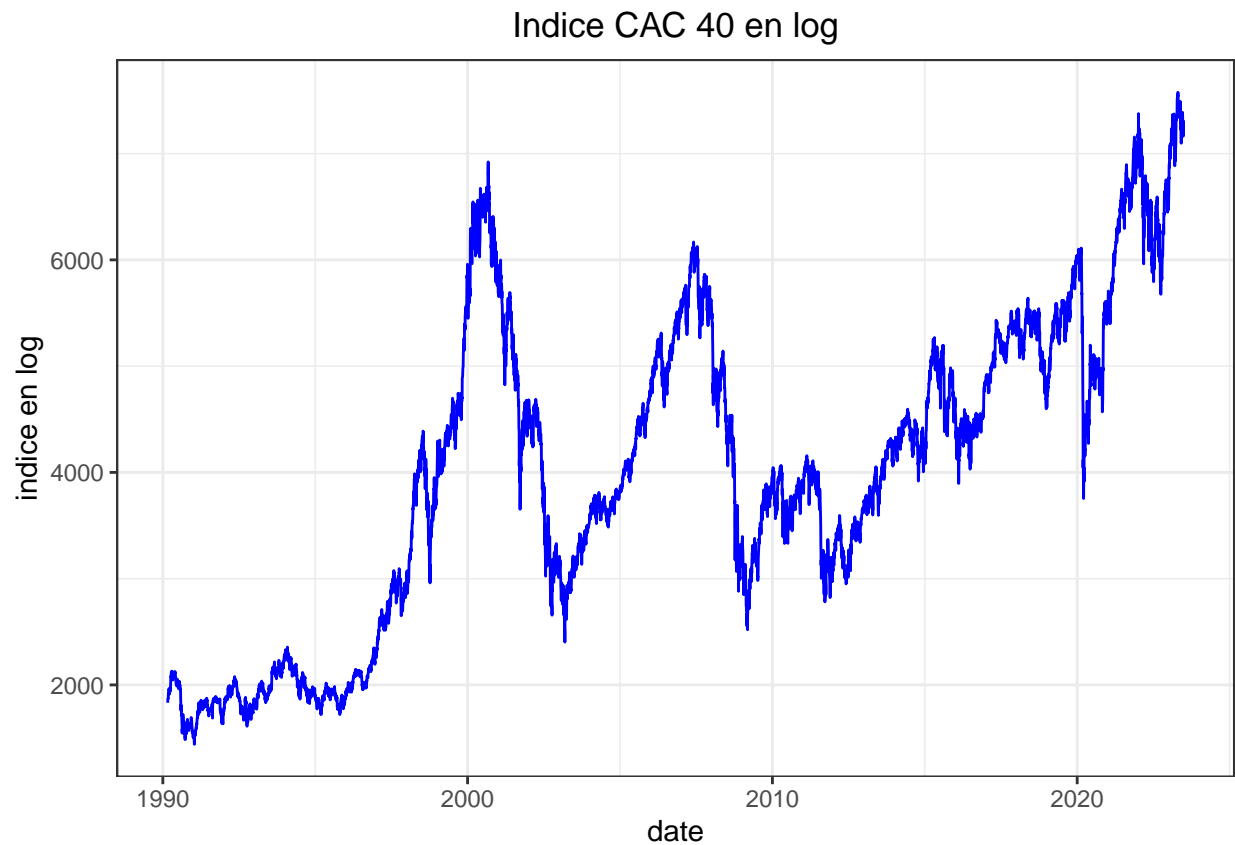
```
## Warning: le package 'yahoofinancer' a été compilé avec la version R 4.2.3
```

```
cac40<-Index$new('^FCHI')
cac40_data<-cac40$get_history(start = '1990-01-01',interval = '1mo')
```

Représentation graphique de l'indice CAC40

On utilise les données importées via quantmod.

```
## Warning: le package 'ggplot2' a été compilé avec la version R 4.2.2
```



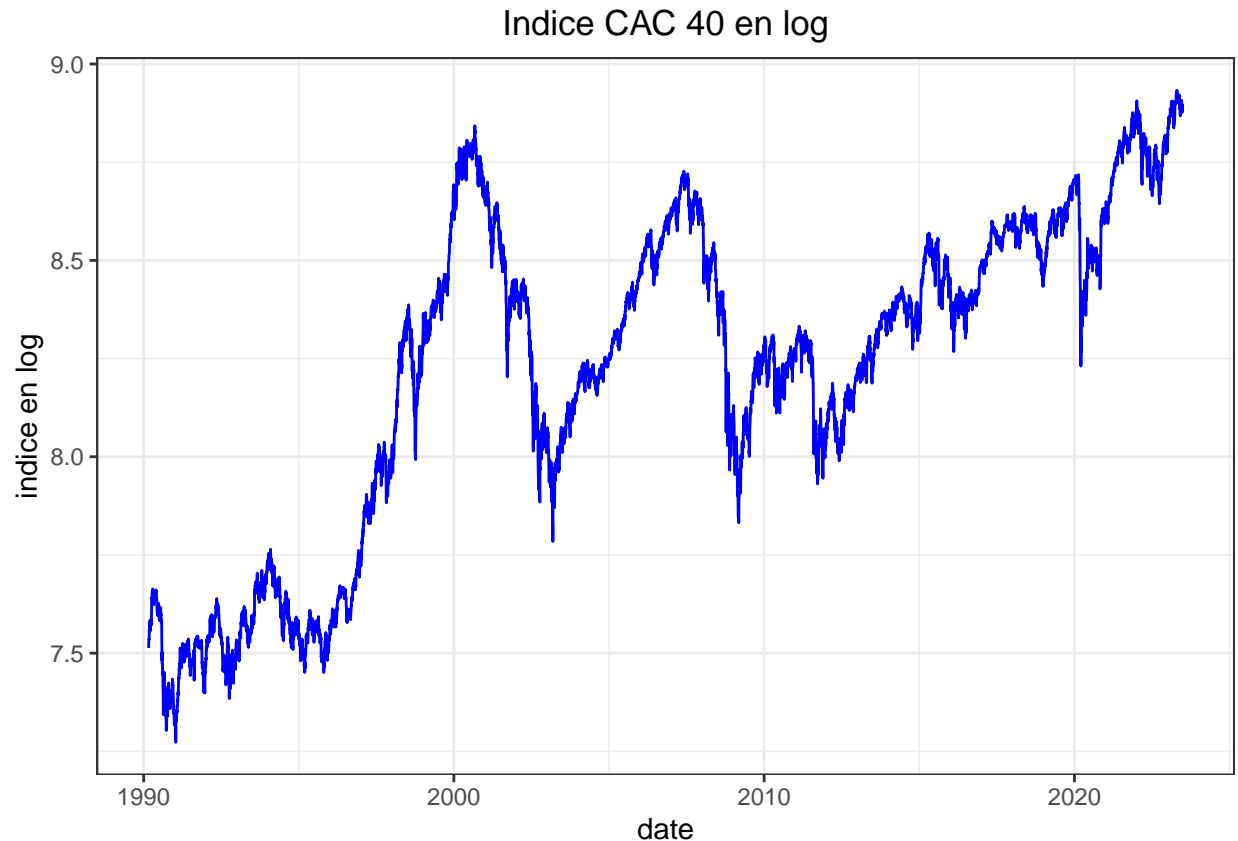
Calcul de l'indice en log

On ajoute l'indice en log au dataframe cac40

```
CAC40_df$logI<-log(CAC40_df$Adj.Close)
```

```
library(ggplot2)
```

```
p<-ggplot(data=CAC40_df,aes(x=Date,y=logI))+geom_line(colour='blue')+ggtitle('Indice CAC 40 en log')+xlab('date')+ylab('indice en log')+p
```



Calcul de l'indice en différence première

On note I_t l'indice CAC40, $i_t = \log(I_t)$ l'indice en logarithme, alors la différence première de l'indice en logarithme est $i_t - i_{t-1} = \log(P_t) - \log(P_{t-1}) = \log(\frac{P_t}{P_{t-1}})r_t$.

En calculant la différence première, on perd la première observation. On ne peut donc stocker la nouvelle variable créée dans le dataframe précédent.

```
r_CAC40_df<-data.frame(CAC40_df$Date[-1],diff(CAC40_df$logI))
colnames(r_CAC40_df)<-c('Date','rdt')
```

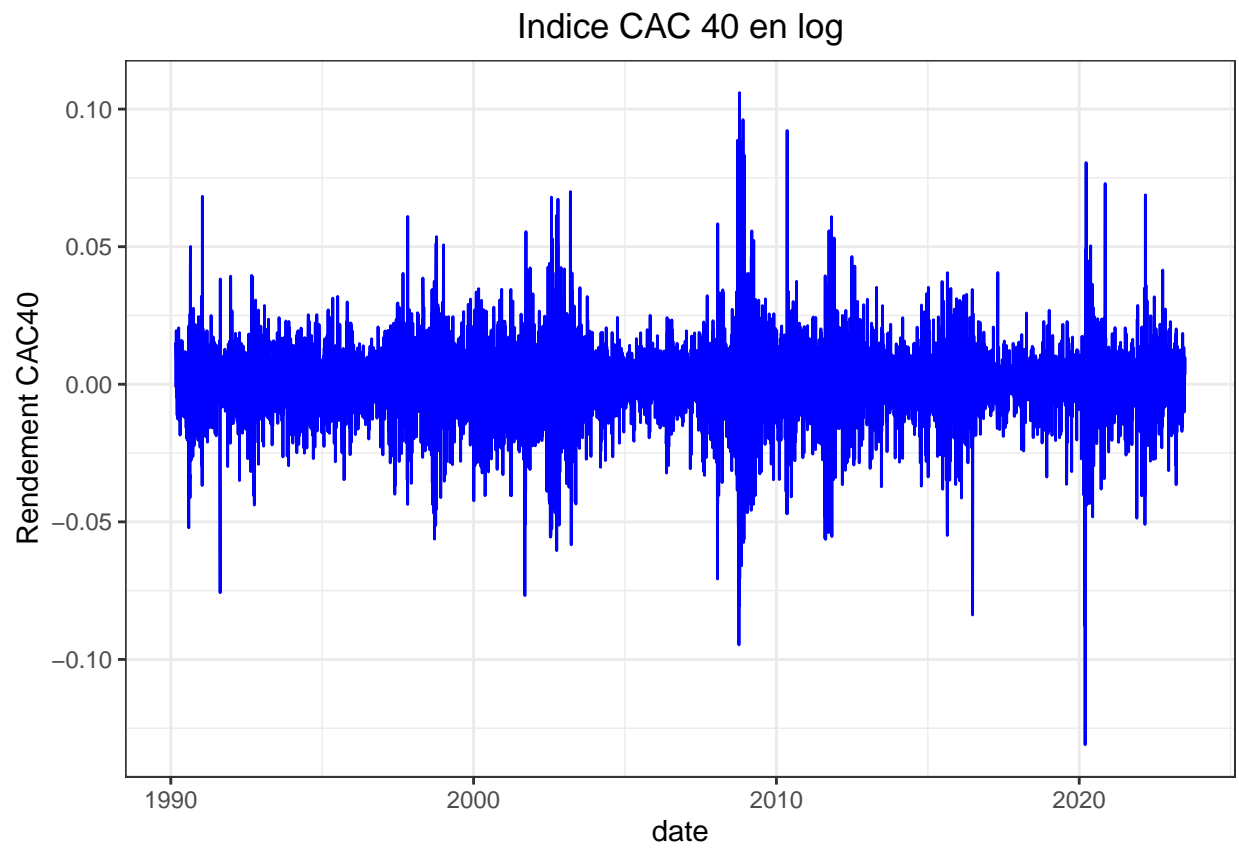
```
str(r_CAC40_df)
```

```
## 'data.frame': 8460 obs. of 2 variables:
## $ Date: Date, format: "1990-03-02" "1990-03-05" ...
## $ rdt : num 0.01517 0.0075 -0.00107 0.00426 0.01949 ...
```

```
head(r_CAC40_df)
```

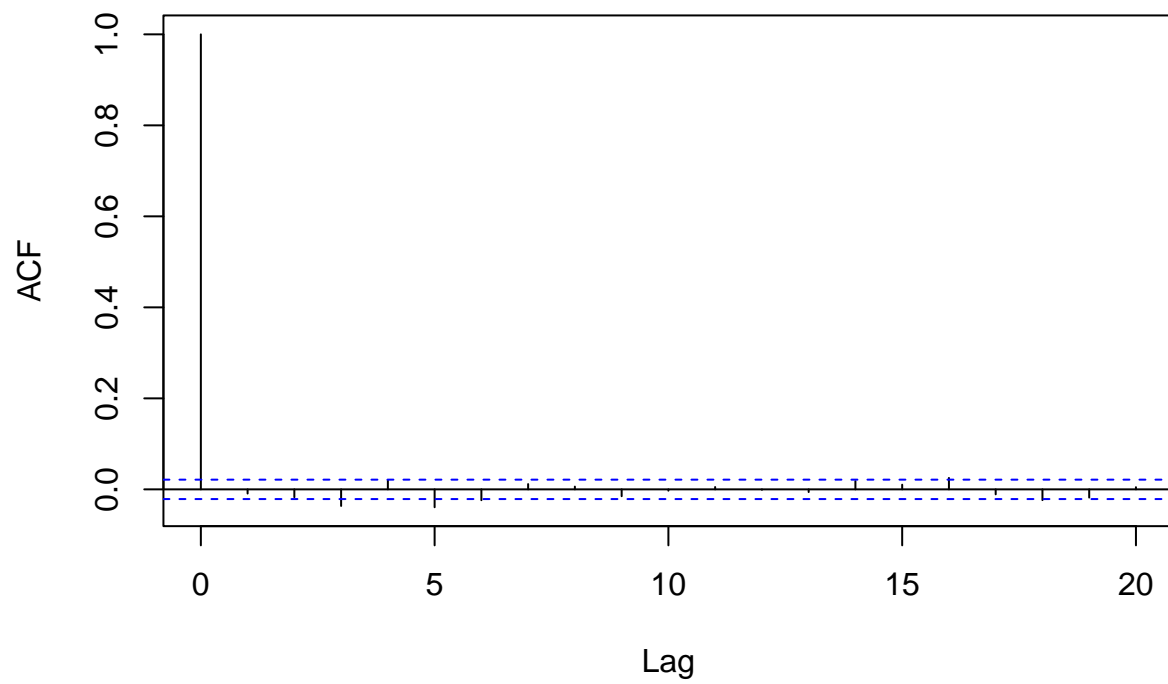
```
##      Date      rdt
## 1 1990-03-02 0.015168221
## 2 1990-03-05 0.007498696
## 3 1990-03-06 -0.001067806
## 4 1990-03-07 0.004264399
## 5 1990-03-08 0.019489687
## 6 1990-03-09 0.002084420
```

```
p<-ggplot(data=r_CAC40_df,aes(x=Date,y=rdt))+geom_line(colour='blue')+ggtitle('Indice CAC 40 en log')+xlab('date')
p
```



```
acf(x = r_CAC40_df$rdt,lag.max = 20)
```

Series r_CAC40_df\$rdt



```
pacf(x = r_CAC40_df$rdt, lag.max = 20, main='Autocorrelations partielles')
```


Autocorrelations partielles

