
# Projet de Séries Temporelles

2ème année cycle d'ingénieur

Kacim Younsi
Mira Maamari

Mai 2023

# 1 Part 1 - The Data

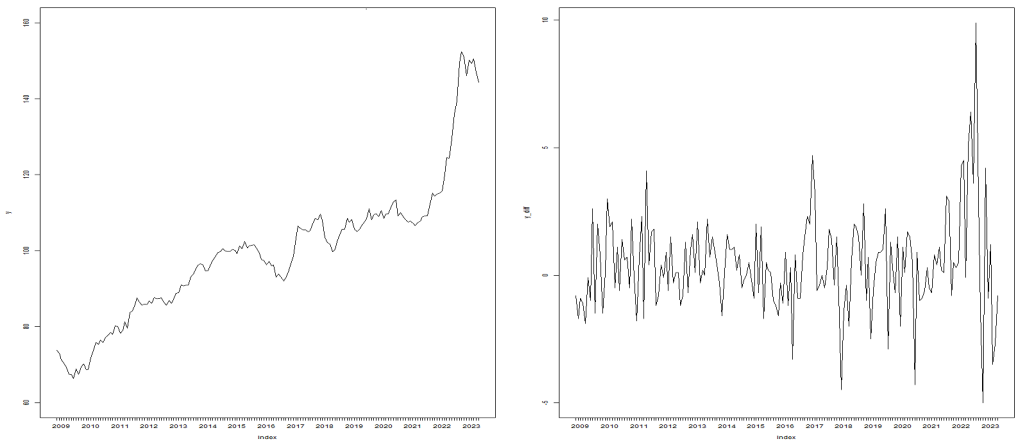
## 1.1 Presentation of the chosen time series

The series we chose represents the industrial producer price index (PPI) for the French dairy products sold on the foreign markets. This index measures the rate of change in the price of French dairy products at the moment they are first sold by the producer, without taking into account any intermediary cost. The dataset contains monthly values ranging from October 2008 to March 2023 (around 150 values).

## 1.2 Stationarity

Before modeling our series with ARMA model, we need to check that it is stationary, and to make it stationary if it is not.

### 1.2.1 First observations and hypothesis




Graph 1 – Plots of the series (left) and its first difference series (right)

The series in level seems to present a linear ascending trend, and is thus not stationary. Yet, it seems quite stable around its trend, which makes us want to check for the first difference series.

The first difference series looks stable around a null constant and may thus be stationary. We guess that the initial series is probably I(1).

### 1.2.2 Validation of the hypothesis

**Non stationarity of the level series**

In order to check the validity of our hypothesis, we first check for the non stationarity of $Z_t$

We check if there is an intercept or a non linear trend, by regressing $Z_t$ on its dates :

$$Z_t = \alpha_0 + \beta_0 t + \varepsilon_0$$

We find that the coefficients for the intercept and the linear trend (dates) are significative and non null, $\beta_0$ being indeed positive.

We thus study the case of unit root tests with intercept and possibly non zero trends. We chose to perform and augmented Dickey-Fuller test (ADF), and had to consider 6 lags in order to erase residual autocorrelation.

In our ADF test, we thus suppose that $Z_t$ follows :

$$Z_t = \alpha + \beta t + \pi Z_{t-1} + \sum_{l=1}^{6} \phi_l \Delta Z_{t-l} + \varepsilon_0$$

The unit root is not rejected at the 95% level. The series $(Z_t)$ is thus at least I(1).

**Stationarity of the first difference series**

We now test the stationarity of the first difference series.

As before, we first check the presence of any trend or intercept by the regression of $(\Delta Z_t)$ on dates :

$$\Delta Z_t = \alpha_0 + \beta_0 t + \varepsilon_0$$

We find no significative constant nor trend.

We thus perform the ADF test in the no-constant and no-trend case. When controling for the non correlation of the residuals, we have to include one lag, which gives us the following model (we denote $X_t := \Delta Z_t$) :

$$X_t = \pi X_{t-1} + \phi_1 \Delta X_{t-1} + \varepsilon_0$$

The test rejects the unit root hypothesis (p-value<0.05), and thus, does not reject the stationarity of $(X_t)$. We therefore accept the hypothesis that $(Z_t)$ **is I(1)** , as guessed by the plot.

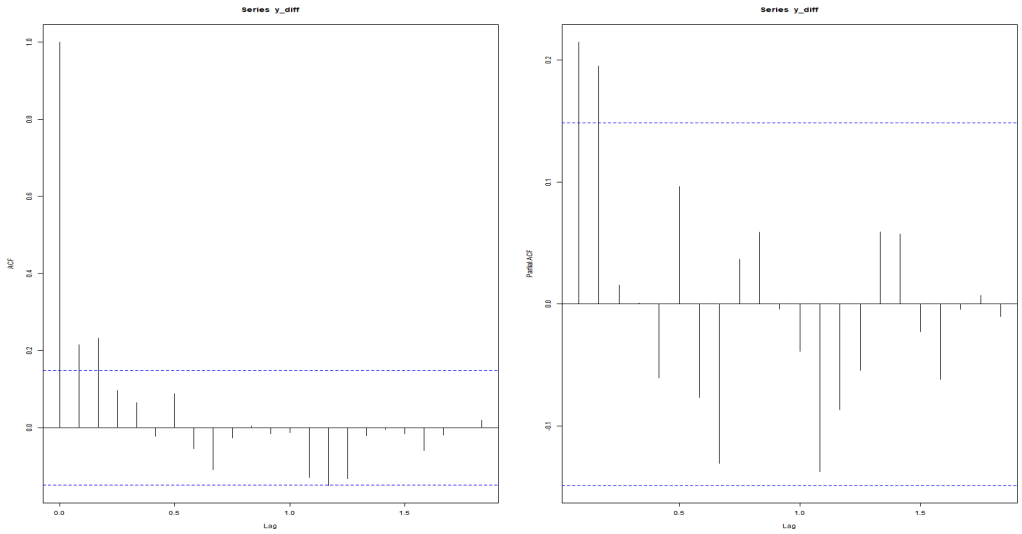
# 2 Part 2 - ARMA models

## 2.1 ARMA(p, q) model for the corrected time series

The corrected time series $(X_t)$ being stationary, we can model it with an ARMA(p, q). We chose the model using the Box-Jenkins method.

**Step 1- A priori identification of p and q**

The autocorrelations and partial autocorrelations for $(X_t)$ are both non-null until h = 2, and we do not reject their nullity starting for any h > 2. Thus, possible models for our corrected series are MA(2), AR(2) and mixed models.




Graph 2 – acf (left) and pacf (right)

**Step 2 - Estimation of parameters**

For each of the potential models, we estimate the parameters using the arima method, which essentially uses the maximum likelihood estimation.

**Step 3 - Diagnostic checking (tests on parameters and residuals)**

In order to test if each model is valid and well adjusted, we check the significance of the parameters, and the non-autocorrelation of the residuals.

AR(2), MA(2), ARMA(1, 1) and ARMA(2, 2) are found to be valid and well adjusted.

**Step 4 - Model choice using information criteria**

We then compute the AIC and BIC for each model, in order to chose the "best", in terms of

information.

| | ar2 | ma2 | ar1ma1 | ar2ma2 |
|---|---|---|---|---|
| AIC | 704.27 | 706.08 | 706.09 | 708.07 |
| BIC | 716.86 | 718.72 | 718.73 | 727.03 |

AR(2) having the smallest AIC and BIC, we chose it as our model for the corrected series $(X_t)$.

### 2.2 ARIMA(p, d, q) for the series $(Z_t)$

Thus, the chosen model for our initial series is an ARIMA(2, 1, 0).

$$\Phi(B)(1-B)Z_t = \varepsilon_{t0} \text{ with } (\varepsilon_{t0}) \text{ white noise}$$

$$\Leftrightarrow \Delta Z_t = \phi_1 \Delta Z_{t-1} + \phi_2 \Delta Z_{t-2} + \varepsilon_t \text{ with } (\varepsilon_t) \text{ white noise}$$

After fitting the model on our initial series, we obtain the estimated parameters : $\phi_1 = 0.1958$ and $\phi_2 = 0.2262$.

## 3 Part 3 - Prediction

### 3.1 Confidence region for $(X_{T+1}, X_{T+2})$

We denote T the length of the differenciated series, and we suppose that the residuals $(\varepsilon_t)$ follow a $\mathcal{N}(0, \sigma^2)$ in the ARMA(2, 0) equation of $(X_t)$.

We have :

$$\begin{cases} X_{T+1} = \phi_1 X_T + \phi_2 X_{T-1} + \varepsilon_{T+1} \\ X_{T+2} = \phi_1 X_{T+1} + \phi_2 X_T + \varepsilon_{T+2} \end{cases} \tag{1}$$

$$\Leftrightarrow \begin{cases} X_{T+1} = \phi_1 X_T + \phi_2 X_{T-1} + \varepsilon_{T+1} \\ X_{T+2} = \phi_1(\phi_1 X_T + \phi_2 X_{T-1} + \varepsilon_{T+1}) + \phi_2 X_T + \varepsilon_{T+2} \end{cases} \tag{2}$$

$$\Leftrightarrow \begin{cases} X_{T+1} = \phi_1 X_T + \phi_2 X_{T-1} + \varepsilon_{T+1} \\ X_{T+2} = (\phi_1^2 + \phi_2) X_T + \phi_1 \phi_2 X_{T-1} + \phi_1 \varepsilon_{T+1} + \varepsilon_{T+2} \end{cases} \tag{3}$$

Since X follows a causal ARMA, $\varepsilon_t$ is the innovation. The predictions at time T of $X_{T+1}$ and $X_{T+2}$ are thus given by :

$$\begin{cases} X_{T+1}^T = \phi_1 X_T + \phi_2 X_{T-1} \\ X_{T+2}^T = (\phi_1^2 + \phi_2) X_T + \phi_1 \phi_2 X_{T-1} \end{cases} \tag{4}$$

The prediction errors are given by :

$$\begin{cases} X_{T+1} - X_{T+1}^T = \varepsilon_{T+1} \\ X_{T+2} - X_{T+2}^T = \phi_1 \varepsilon_{T+1} + \varepsilon_{T+2} \end{cases} \tag{5}$$

Thus, we can compute the variance-covariance matrix of the error of estimation for $(X_{T+1}, X_{T+2})$ :

$$\Sigma := \mathbb{V}(\begin{pmatrix} \varepsilon_{T+1} \\ \phi_1 \varepsilon_{T+1} + \varepsilon_{T+2} \end{pmatrix}) = \sigma^2 \begin{pmatrix} 1 & \phi_1 \\ \phi_1 & \phi_1^2 + 1 \end{pmatrix} \tag{6}$$

Since $\text{Det}(\Sigma) = \sigma^2$, it is non null. $\Sigma$ is thus invertible and we have :

$$\begin{pmatrix} \varepsilon_{T+1} \\ \phi_1 \varepsilon_{T+1} + \varepsilon_{T+2} \end{pmatrix} \sim \mathcal{N}(0, \Sigma) \tag{7}$$

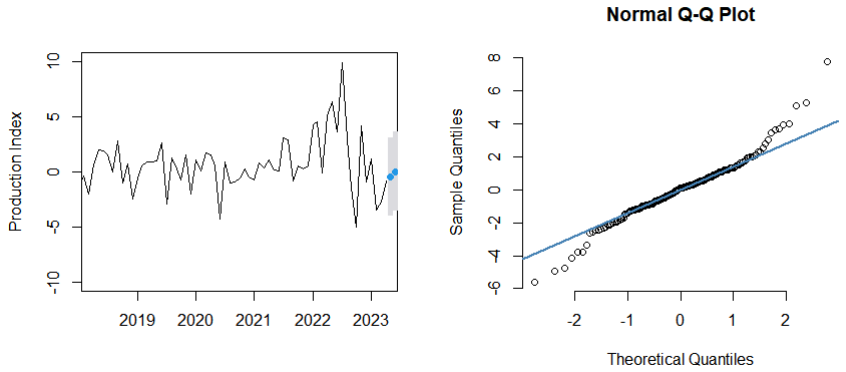
$$\begin{pmatrix} \varepsilon_{T+1} \\ \phi_1 \varepsilon_{T+1} + \varepsilon_{T+2} \end{pmatrix}' \Sigma^{-1} \begin{pmatrix} \varepsilon_{T+1} \\ \phi_1 \varepsilon_{T+1} + \varepsilon_{T+2} \end{pmatrix} \sim \chi^2(2) \tag{8}$$

We denote $q_{\chi^2(2)}^{1-\alpha}$ the quantile at level $1-\alpha$ of the $\chi^2(2)$ law. The confidence region for X := $(X_{T+1}, X_{T+2})$ is thus given by :

$$R_\alpha = \{X | \begin{pmatrix} \varepsilon_{T+1} \\ \phi_1 \varepsilon_{T+1} + \varepsilon_{T+2} \end{pmatrix}' \Sigma^{-1} \begin{pmatrix} \varepsilon_{T+1} \\ \phi_1 \varepsilon_{T+1} + \varepsilon_{T+2} \end{pmatrix} \leqslant q_{\chi^2(2)}^{1-\alpha} \} \tag{9}$$

In order to obtain this interval, we have supposed that the residuals are Gaussians and that the chosen model (AR(2)) and estimated coefficients are correct.

Here is the graphical representation of the confidence region :




Graph 3 – Prediction and confidence intervals and normal QQ-Plot of residuals

What should be noted is that the length of the interval of the second prediction seems similar to the first one but is in fact slightly larger when we look at the numerical values. However, these intervals rely on the hypothesis of gaussians residuals which is rejected, as indicated by the QQ-plot (we confirmed it with a Shapiro-Wilk normality test).

## 3.2 Open question

Let $(Y_t)$ a stationary time series available from t = 1 to T such that $(Y_{T+1})$ is available faster than $(X_{T+1})$. This information allows to improve the prediction of $X_{T+1}$ when $(Y_t)$ causes $(X_t)$ instantaneously in the Granger sense.

We can test this hypothesis using a Wald test on our series.

# 1 R Code

```
#######################################
###             Packages            ###
#######################################

require(forecast)
require(zoo)
require(tseries)
require(fUnitRoots)

#######################################
###      Cleaning the workspace     ###
#######################################
rm(list=(objects()))

#######################################
### Import the data and set the path ###
#######################################
# path <- "C:\\Users\\youns\\Documents\\GitHub\\LTS_Forecasting_Project_ENSAE_2022-2023"
path <- "C:\\Users\\mira_\\Documents\\GitHub\\LTS_Forecasting_Project_ENSAE_2022-2023"
setwd(path) #definit l'espace de travail (working directory ou "wd")
getwd() #affiche le wd
list.files() #liste les elements du wd

datafile <- "valeurs_mensuelles.csv" #definit le fichier de donnees
data <- read.csv(datafile,sep=";")
#importe un fichier .csv dans un objet de classe data.frame


# Define time and interest variable #

dates_char <- as.character(data[[1]])
T <- length(dates_char)
dates_char <- dates_char[4:T]
dates_char[1]
tail(dates_char,1)
dates <- as.yearmon(seq(from=2008+10/12, to=2023+3/12, by=1/12))
y <-zoo(data[[2]][4:T], order.by=dates)
y_num = as.numeric(y)
y_num = zoo(y_num, order.by=dates)
y_diff = diff(y_num, differences = 1)#first difference
y_diff = zoo(y_diff, order.by=dates)

#Q2)

#######################################
### Plot the data ###
#######################################

plot(y, ylim =c(60,160))
graphics.off()

#The series in level seems to have a increasing linear trend.

plot(y_diff)
graphics.off()


#The first difference series seems relatively stable around a null constant
#and could be stationary.

#We guess that the series is probably I(1)


#######################################
### Stationary ###
#######################################
# Before modeling our series with ARMA model we need to check that it is stationary
# If it is not, we need to correct it by differentiating it or deseasonalizing it.

### Useful function ###
adfTest_valid <-
  function(series,kmax,type){ #ADF tests until no more autocorrelated residuals
    k <- 0
    noautocorr <- 0
    while (noautocorr==0){
      cat(paste0("ADF with ",k, " lags: residuals OK? "))
```

```
      adf <- adfTest(series,lags=k,type=type)
      pvals <- Qtests(adf@test$lm$residuals,24,fitdf=length(adf@test$lm$coefficients))[,2]
      if (sum(pvals<0.05,na.rm=T) == 0) {
        noautocorr <- 1; cat("OK \n")}
      else cat("nope \n")
      k <- k + 1
    }
    return(adf)
}

Qtests <- function(series, k, fitdf=0) {
  pvals <- apply(matrix(1:k), 1, FUN=function(l) {
    pval <- if (l<=fitdf) NA else Box.test(series,
    lag=l, type="Ljung-Box", fitdf=fitdf)$p.value
    return(c("lag"=l,"pval"=pval))
  })
  return(t(pvals))
}


#Before performing the unit root tests to check stationarity, we need to check if there is an
#intercept and / or a non null linear trend.
#The graph representation of the series showed that the trend is
#probably linear and increasing.


# Let's regress y on its dates to check :
summary(lm( formula = y ~ dates))

#The coefficient associated with the linear trend (dates) is indeed positive, thus we
#need to study the case of unit root tests with intercept and possibly non zero trends

adf <- adfTest(y_num, lag=0, type="ct") # ct here take into account the fact that y
#has an intercept and non zero trend.
#Before interpreting the test, let's check that the model's residuals are not autocorrelated,
#otherwise the test would not be valid.

Qtests(adf@test$lm$residuals,24,length(adf@test$lm$coefficients))

#We reject the absence of residual autocorrelation for every lag, thus invalidating the ADF
#test without lags. Let's add lags of Xt until the residuals are no longer autocorrelated.

adf <- adfTest_valid(y_num,24, type="ct") # ct here take into account the fact that y
#has an intercept and non zero trend.
#We have had to consider 6 lags on the ADF test to erase residual autocorrelation.

adf
#The unit root is not rejected at the 95% - level for the series in levels, the series
#is thus at least I(1).


#Let's now test the unit root for the first differenciated series. The previous graph
#representation seems to show the absence of a constant and non zero trend.
#Let's check with a regression :

summary(lm( formula = y_diff ~ dates[-1]))

#There isn't any constant or significant trend. Let's perform the ADF test in the no-constant
#and no-trend case, and control for the absence of residual autocorrelation.

adf <- adfTest_valid(y_diff,24, type="nc") # nc here take into account the fact that y has no
#intercept and zero trend.

#It was necessary to include 1 lags in the ADF test
adf

#The test rejects the unit root hypothesis (p-value<0.05), we will thus say that the
#differenciated series is "stationary". y is therefore I(1) as guessed by the plot.


#Q3)
#we plot the series before and after transforming it

plot(cbind(y_num,y_diff))
graphics.off()
```

```r
#####################################
### ARMA modelization ###
#####################################

#Q4)
#Since our data is now stationary, we can try to model it with an ARMA(p,q) model.

### Identification of p and q ###
par(mfrow=c(1,2)) #puts the graphs into 1 column and 2 lines
acf(y_diff)
pacf(y_diff)
graphics.off()

#Since the series is stationary, it is integrated of order d = 0.
#The complete autocorrelation functions are statistically significant
#(i.e. bigger than the bounds ±1, 96/n of
#the confidence interval of a null test of the autocorrelation at the 95% level)
#until q = 2 and the partial autocorrelation until p = 2.
#If the differienciated series follows an ARMA(p,q),
#it follows at most an ARMA(p =2, q = 2), which we can estimate.

### Model selection  ###

# We know that our model is at most an ARMA(2,2)
#The potential models are all the ARMA(p,q) for spread where p  2 and q  2.
#We are looking for a model that is :
# | well adjusted : the estimated coefficients
#(notably the coefficients of the higher AR and MA orders) are statistically significant.
# | valid : the residuals are not correlated.


#Function that tests the significance of coefficients
#Allow us to test if the model is well adjusted

signif <- function(estim){
  coef <- estim$coef
  se <- sqrt(diag(estim$var.coef))
  t <- coef/se
  pval <- (1-pnorm(abs(t)))*2
  return(rbind(coef,se,pval))
}
#Function that apply "signif" function and tests if the residuals are autocorrelated
#Allow us to test if the model is valid and well adjusted.

arimafit <- function(estim){
  adjust <- round(signif(estim),3)
  pvals <- Qtests(estim$residuals,24,length(estim$coef)-1)
  pvals <- matrix(apply(matrix(1:24,nrow=6),2,function(c) round(pvals[c,],3)),nrow=6)
  colnames(pvals) <- rep(c("lag", "pval"),4)
  cat("Nullity test of the coefficients :\n")
  print(adjust)
  cat("\n Test of absence of residuals autocorrelation : \n")
  print(pvals)
}


# First we want to apply this function on every simpler model that respect p  2 and q  2
# to test if there exist some simpler model that are both valid and well adjusted.

estim <- Arima(y_diff,c(1,0,0)); arimafit(estim) # Nope:The model is not valid

estim <- Arima(y_diff,c(2,0,0)); arimafit(estim) # OK:The model is well adjusted and valid
ar2 <- estim

estim <- Arima(y_diff,c(0,0,1)); arimafit(estim) # Nope:The model is not valid

estim <- Arima(y_diff,c(0,0,2)); arimafit(estim) # OK:The model is well adjusted and valid
ma2 <- estim

estim <- Arima(y_diff,c(1,0,1)); arimafit(estim) # OK:The model is well adjusted and valid
ar1ma1 <- estim

estim <- Arima(y_diff,c(1,0,2)); arimafit(estim) # Nope:The model is not properly adjusted

estim <- Arima(y_diff,c(2,0,1)); arimafit(estim) # Nope:The model is not properly adjusted

estim <- Arima(y_diff,c(2,0,2)); arimafit(estim) # OK:The model is well adjusted and valid
ar2ma2 <- estim
```

```r
# To choose between all the models that are well adjusted and valid we compute AIC and BIC
# of each model
# We will probably select both models with minimum AIC and BIC.

### Compute the AIC BIC matrix of valid models #####
models <- c("ar2","ma2",'ar1ma1',"ar2ma2"); names(models) <- models
apply(as.matrix(models),1, function(m) c("AIC"=AIC(get(m)), "BIC"=BIC(get(m))))

#We can see that the model AR(2) has both minimum AIC and BIC, that's why we select it
# Thus our final model for the first difference series is an ARIMA(2,0,0)


#Q5

#plotting the corrected series and its modelisation
plot(ar2$x, col = "red")
lines(fitted(ar2), col = "blue")
graphics.off() #cleaning the graph window


#fitting the ARIMA(2, 1, 0) to the initial series
estim <- Arima(y_num,c(2,1,0)); arimafit(estim) # OK:The model is well adjusted and valid
ar2i1ma0 <- estim

#plotting the initial series and its modelisation
plot(ar2i1ma0$x, col = "red")
lines(fitted(ar2i1ma0), col = "blue")
graphics.off()

#Q8
#obj <-forecast(ar2, h = 2, level = 95)
#autoplot(obj, 50)

qqnorm(ar2$residuals, pch = 1, frame = FALSE)
qqline(ar2$residuals, col = "steelblue", lwd = 2)

shapiro.test(ar2$residuals)
#H0: The residuals are gaussian
#p_value << 0.01 we reject H0

#Given the QQ-plot and the Shapiro WIlk normality test
#we can reject the hypothesis of normality of residuals
#So we keep in mind that the hypothesis of the test is not verified

dates2 <-append(dates, c(as.yearmon(2023+1/12), as.yearmon(2023+2/12)))


forecast_result <- ar2 %>% forecast(h = 2, level = c(95))
forecast_result
# Set the plot range and other parameters
plot(forecast_result, xlim = c(max(time(ar2$x)) - 5, max(time(ar2$x))),
     ylim = c(-10, 10), main = "", ylab = "Production Index")

# Create a sequence of dates for custom x-axis tick positions
custom_dates <- seq.Date(max(time(ar2$x)) - 5, max(time(ar2$x)), by = "1 month")

# Add custom x-axis tick positions and labels
axis.Date(1, at = custom_dates, format = "%b %Y")
```