ENSAE Paris

Linear Time Series Project

Index of Industrial Production in Aeronautical and Space Construction

Antoine GILSON, Valentin SMAGUE

May 23rd 2024

1 The data

1.1 What does the data represent ?

The series we study in this report represents the index of industrial production in the aeronautical and space sectors in France. The statistics are delivered by INSEE and allow us to monitor the monthly evolution of industrial activity in France. The series studied has 411 observations from January 1990 to March 2024, with a monthly frequency, but we decided to cut our series before the COVID 19 to ensure propers results (last value is February then 2020), leaving us with 362 observations.

Here are first some figures to help visualize our data.

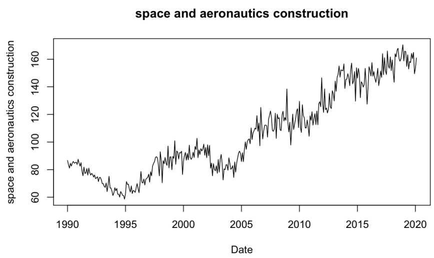

Figure 1 – Original series

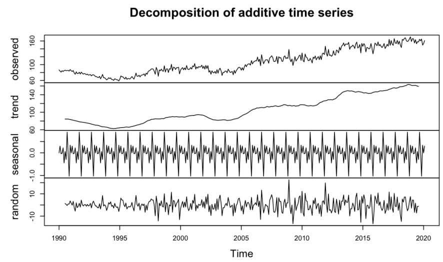

Figure 2 – Decomposition of the series

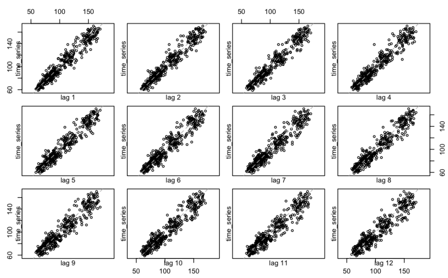

Figure 3 – Lagplot

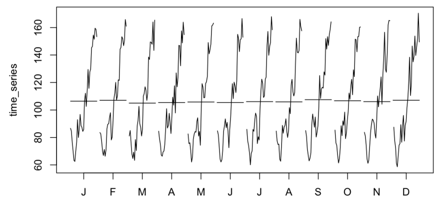

Figure 4 – Monthplot

- The representation of the series (Figure 1) indicates an increasing linear trend, but without any clear signs of seasonality.
- The monthplot (Figure 4) shows 12 similar monthly patterns, also suggesting a lack of seasonality in the year.
- Eventually, the lagplot (Figure 3) shows a strong correlation between the variables since lag 1.

Next, we analyze two new figures to show the auto-correlation (ACF) and the partial auto-correlation (PACF), in order to verify if there is a seasonality or if the series is stationnary.

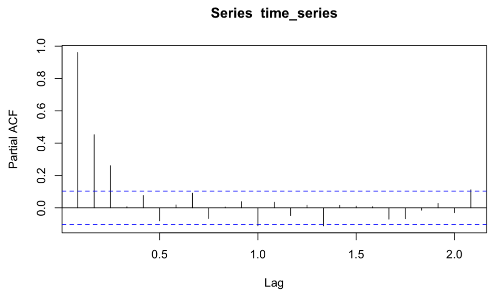

FIGURE 5 – PACF on the series

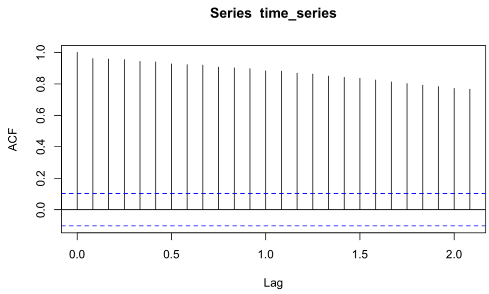

FIGURE 6 – ACF on the series

— Firstly, the PACF does not display a repeated pattern, indicating that the series likely lacks seasonality, as suggested earlier in the figures.
— On the other hand, the auto-correlations decrease very gradually and the partial auto-correlation of order 1 is close to 1. Then, the series does not seem to be stationary either.

Then, we perform the unit root tests to confirm that the series is not stationnary. To do so, we first need to check if there is a non-null linear trend as expected, based on Figures 1 and 2. Thus, we regress the series on its dates.

	Estimate	Std. Error	t value	Pr($\geq\|t\|$)
Intercept	56.454723	1.213863	46.51	$\leq 2e-16$
date	0.274107	0.005796	47.29	$\leq 2e-16$

TABLE 1 – Coefficients of the Linear Model

The coefficients here show evidence for a trend. Thus, we need to study the case of unit root tests with intercept and possibly non zero trends. The augmented Dickey-Fuller test (ADF) in the intercept and trend case consists in the following regression, with a given S variable :

$$\Delta S_t = c + bt + \beta S_{t-1} + \sum_{l=1}^{k} \phi_l \Delta S_{t-l} + \epsilon_t$$

where $\beta + 1$ is the autocorrelation of order 1 of S and k the number of lags needed to render our residuals non autocorrelated. We run this test until we find the right value of k. Since we have a monthly series, we test residual autocorrelation up to order 24 (2 years). We have had to consider 21 lags on the ADF test to erase residual autocorrelation. We also run a KPSS test to complete the analysis.

Test	Stats	Lag	p-value
ADF	-3.3026	21	0.07081
KPSS	0.77428	5	≤ 0.01

TABLE 2 – Results of various tests on the original series

These results ensure that the series is not stationnary :
— The p-value of the ADF test, where H_0 means "our series isn't stationnary", is above 0.05, which does not allow us to reject the hypothesis with 95% confidence.
— On the other hand, the p-value of the KPSS test, where H_0 is "our series is stationnary" is below 0.01 which allows us to reject the hypothesis of stationarity with 99% confidence.

As a result, our series is indeed neither stationnary nor seasonal.

1.2 Making the series stationary

As we showed that the series wasn't stationnary, we have to stationarize it to exploit it later. Where S_t is our initial series, we use the first difference method to stationarize our series : $X_t = \Delta S_t = S_t - S_{t-1}$.

To verify that our new series is indeed stationary, we first check the trend using a linear regression on t.

	Estimate	Std. Error	t value	Pr($\geq|t|$)
Intercept	-0.088576	0.854343	-0.104	0.917
date	0.001626	0.004091	0.398	0.691

TABLE 3 – Coefficients of the Linear Model

There is not any constant or significant trend. We now perform the ADF test in the no-constant and no-trend case, and control for the absence of residual autocorrelation. We also run a KPSS test.

Test	Stats	Lag	p-value
ADF	-14.6696	3	≤ 0.01
KPSS	0.098491	5	≥ 0.1

TABLE 4 – Results of various tests on the differenciated series

— The p-value of the ADF test, is well below 0.01, which allows us to reject the hypothesis with 99% confidence.

— On the other hand, the p-value of the KPSS test is well above 0.1, which does not allows us to reject the hypothesis of stationarity with 95% confidence.

The next figure shows the series before and after the application of the first difference method.

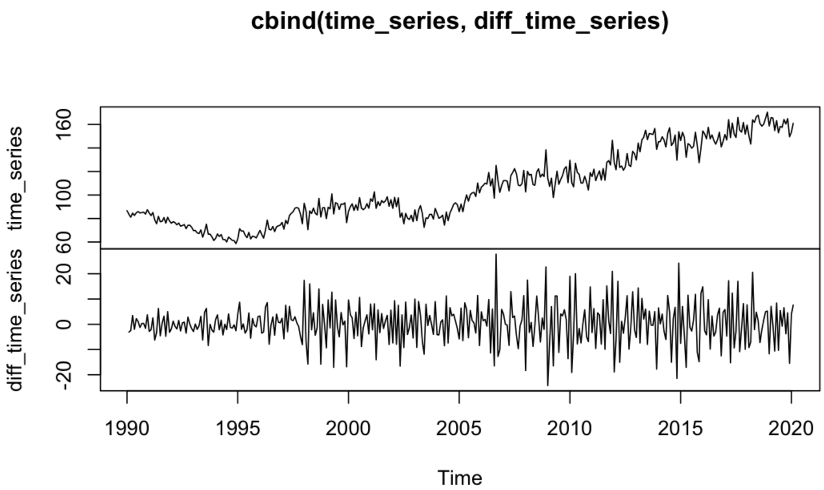

FIGURE 7 – Comparaison before and after the first difference method

We have succeeded in stationnarizing our series.

2 ARMA and ARIMA models

2.1 ARMA model

We compute the ACF and the PACF on the differentiated series to find p_{max} and q_{max}.

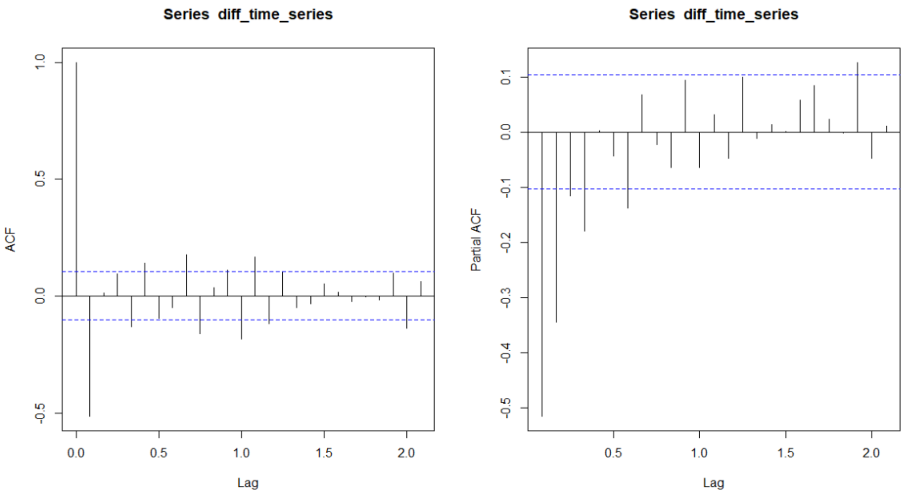

FIGURE 8 – ACF and PACF on the differenciated series

According to the figure above, ACF is significant only until lag 1 and PACF until lag 4 so we set $q_{max} = 1$ and $p_{max} = 4$. We need now to check each combination (p, q) for $p \leq p_{max}$ and $q \leq q_{max}$. To do so, we check for each model the significance of the coefficients and the residuals' autocorrelations. The only model acceptable is AR(4). In fact, the p-value of the 24th residual is under 0.05 but we will not take it into account.

	ar1	ar2	ar3	ar4	intercept
coef	-0.754	-0.501	-0.245	-0.178	0.203
se	0.052	0.064	0.064	0.052	0.125
pval	0.000	0.000	0.000	0.001	0.105

TABLE 5 – Coefficients Nullity Tests for AR(4)

lag	1	2	3	4	5	6	7	8	9	10	11	12
pval	NA	NA	NA	NA	0.225	0.189	0.342	0.170	0.172	0.259	0.147	0.085
lag	13	14	15	16	17	18	19	20	21	22	23	24
pval	0.060	0.086	0.053	0.069	0.094	0.089	0.057	0.078	0.105	0.137	0.164	0.047

TABLE 6 – Lag and p-values for AR(4)

We also compute the two information critierions (AIC and BIC) for each model. The results are presented in the appendix. The AR(4) model minimize the AIC. Finally, the adjusted R^2 we find is 0.172216.

2.2 ARIMA model

We have differentiated the initial series once to obtain the series X_t. So, d is equal to 1. Thus, the model corresponding to the series we initially chose is the ARIMA(4,1,0) model.

The two following figures show the observed series (in black) and the series predicted by the model (in red).

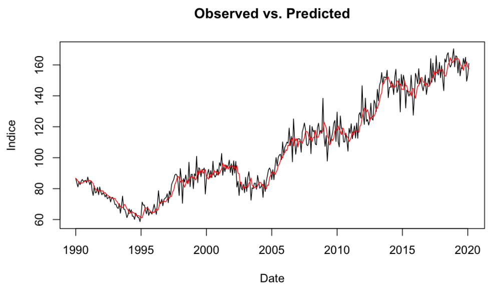

FIGURE 9 – Series against ARIMA(4,1,0) model

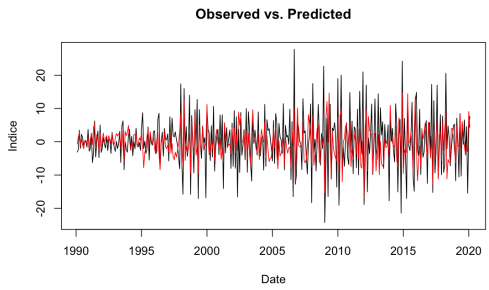

FIGURE 10 – Differenciated series against AR(4) model

3 Prediction

3.1 Confidence regions of level α

We will assume for the following that the residuals of the series are Gaussian, i.e. that $\epsilon_t \sim N(0, \sigma_\epsilon^2)$. We have a model AR(4) which is written :

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \phi_3 X_{t-3} + \phi_4 X_{t-4} + \epsilon_t \tag{1}$$

Knowing that $E[\epsilon_{T+h}|X_T, X_{T-1}, \ldots] = 0 \ \forall h > 0$, by the course, we know that the optimal forecast in T are given by :

$$\begin{cases} \hat{X}_{T+1|T} = \phi_1 X_T + \phi_2 X_{T-1} + \phi_3 X_{T-2} + \phi_4 X_{T-3} \\ \hat{X}_{T+2|T} = \phi_1 \hat{X}_{T+1|T} + \phi_2 X_T + \phi_3 X_{T-1} + \phi_4 X_{T-2} \end{cases}$$

Let's compute the prediction errors $X_{T+1} - \hat{X}_{T+1|T}$ and $X_{T+2} - \hat{X}_{T+2|T}$. We have :

$$\hat{X} = \begin{pmatrix} \hat{X}_{T+1|T} \\ \hat{X}_{T+2|T} \end{pmatrix} \quad \text{and} \quad X = \begin{pmatrix} X_{T+1} \\ X_{T+2} \end{pmatrix}$$

Thus :

$$X - \hat{X} = \begin{pmatrix} X_{T+1} - \hat{X}_{T+1|T} \\ X_{T+2} - \hat{X}_{T+2|T} \end{pmatrix} = \begin{pmatrix} \epsilon_{T+1} \\ \epsilon_{T+2} + \phi_1 \epsilon_{T+1} \end{pmatrix}$$

Thus, we have $X - \hat{X} \sim \mathcal{N}(0, \Sigma)$ where Σ is the variance-covariance matrix such that :

$$\Sigma = \sigma_\epsilon^2 \begin{pmatrix} 1 & \phi_1 \\ \phi_1 & 1 + \phi_1^2 \end{pmatrix}$$

As $\det(\Sigma) = \sigma_\epsilon^2$, the variance-covariance matrix is invertible if and only if $\sigma_\epsilon^2 > 0$, which we have assumed to be true. According to the course, we finally get $(X - \hat{X})^\top \Sigma^{-1} (X - \hat{X}) \sim \chi^2(2)$, which allows us to directly deduce the confidence region of level α. We thus get $\forall \alpha \in [0, 1]$:

$$\left\{ X \in \mathbb{R}^2 \mid (X - \hat{X})^\top \Sigma^{-1} (X - \hat{X}) \leq q_{1-\alpha}^{\chi^2(2)} \right\}$$

Where $q_{1-\alpha}^{\chi^2(2)}$ is the quantile of order $1 - \alpha$ of the $\chi^2(2)$ distribution.

3.2 Hypothesis used

Firstly, we check that our residuals are Gaussian using the figures below. On the left, the blue curve represents a normal distribution of mean and variance followed by our residuals, while the black curve represents the density of our residuals. The two curves have a similar trend. On the right, we plot the normal Q-Q plot of the residuals. It seems to fit well with the red line. Eventually, we performed the Jarque Bera test. The p-value of our test is 0.1066, so we can not reject the Gaussian assumption at 10%. All in all, the Gaussian assumption is acceptable, eventhough a bit strong.

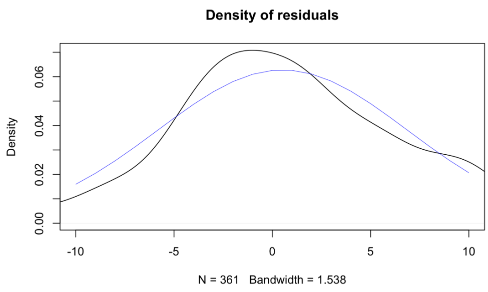

FIGURE 11 – Density of the residuals

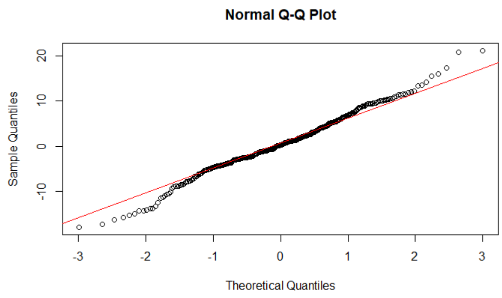

FIGURE 12 – Normal Q-Q plot

Secondly, during the previous computations, we have considered the errors as linear innovations. This hypothesis is verified only if the polynomial in canonical writing does not admit a root inside the unit circle. In our case, this hypothesis is verified [1].

3.3 Graphic representations

The two following figures show the predictions for X_{T+1} and X_{T+2} and their confidence regions of level 95%. On the left, we plot the forecast in red and the confidence region is in grey. On the left, we plot the elliptical bivariate confidence region thanks to the writting of 3.1. However, as we do not known the true value of σ_ϵ^2, we use the value estimated by the model. We get rather large confidence region, therefore the predictions is not good. The assumption about the residuals is finally too strong.

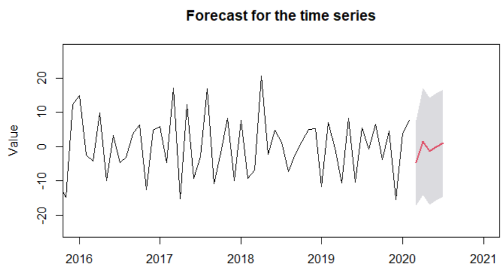

FIGURE 13 – Forecast for the differenciated series

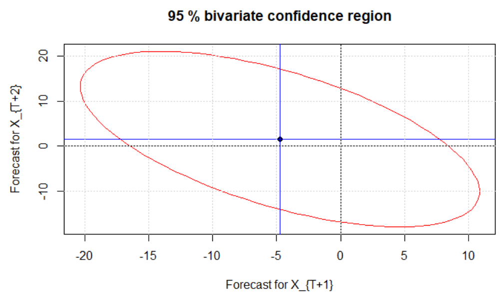

FIGURE 14 – 95% bivariate confidence region

3.4 Open question

We saw during the 8th tutorial that the knowledge of the past Y_t improves the optimal prediction of X_t knowing the past if Y_t instantaneously Granger-cause X_t. For that, we need the assumptions that

$$\begin{pmatrix} X_t \\ Y_t \end{pmatrix} \text{ is a VAR with } \Phi(L)\begin{pmatrix} X_t \\ Y_t \end{pmatrix} = \eta_t = \begin{pmatrix} u_t \\ v_t \end{pmatrix} \text{ a white noise and } \Phi(O) = I_n$$

In our case, it may hold since X_t follows an AR(4) model. Then, Y_t instantaneously Granger-cause X_t if and only if $\mathrm{Cov}(u_t, v_t) \neq 0$, which can be the alternative hypothesis of a Wald test.

1. The modulus of the roots are 1.451935 and 1.662084 (both twice)

4 Appendix

We plot first the shape of the data including data after the Covid19 to illustrate why we decided not to keep this period.

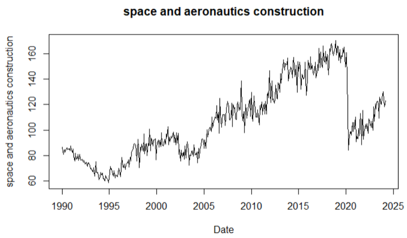

FIGURE 15 – Comparaison before and after the first difference method

We here present the information criterions AIC and BIC used in our project and the values obtained for each model :

$$\text{AIC}(p,q) = \log\left(\hat{\sigma}^2\right) + 2\frac{(p+q)}{n} \text{ where } \hat{\sigma}^2 = \frac{\sum_{i=1}^{n} \hat{\epsilon}_t^2}{n}$$

$$\text{BIC}(p,q) = \log\left(\hat{\sigma}^2\right) + \frac{(p+q)\log(n)}{n}$$

	AIC		BIC	
	q=0	q=1	q=0	q=1
p=0	2535.134	2374.309	2539.023	2382.087
p=1	2426.394	2372.894	2434.171	2384.561
p=2	2382.795	2373.762	2394.462	2389.317
p=3	2380.508	2374.895	2396.064	2394.339
p=4	2371.751	2373.666	2391.196	2396.999

TABLE 7 – AIC et BIC of differenciated series

Finally, we show our R code below.

```r
library(ellipse)
library(tseries)
library(dplyr)
library(forecast)
require(fUnitRoots)

path <- "C:/Users/Valentin/Documents/Travail/ENSAE/valeurs_mensuelles.csv"
setwd(path)
getwd()

# Loading of data
data <- as.data.frame(read.csv(path, sep = ";", header = TRUE, stringsAsFactors = FALSE))
data <- slice(data, -(1:3))
data <- rename(data, date := Libellé)
data <- rename(data, 'Indice brut de la production industrielle aéoronautique et spatiale' := !! colnames(data)[2] )
data <- data[, 1:2]
data <- arrange(data, date)

## Part I ##

# Question 1

# first visualisation
time_series <- ts(as.numeric(data[,2]), start=c(1990, 1), frequency=12)
plot(time_series, xlab="Date", ylab="space and aeronautics construction", main = "space and aeronautics construction")
monthplot(time_series)

# before Covid19
data <- data %>% filter(as.Date(paste0(date, "-01")) < as.Date("2020-03-01"))
time_series <- ts(as.numeric(data[,2]), start=c(1990, 1), frequency=12)
plot(time_series, xlab="Date", ylab="space and aeronautics construction", main = "space and aeronautics construction")
monthplot(time_series)
lag.plot(time_series, lags=12, layout=c(3,4), do.lines=FALSE)
decomp <- decompose(time_series)
plot(decomp)

# acf and pacf
acf(time_series)
pacf(time_series)
n <- length(time_series)

# linear model
summary(lm(time_series~seq(1,n)))

# adf and kpss test
adfTest_valid <- function(series,kmax,type){
  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)
}
adf <- adfTest_valid(time_series,24,"ct")
adf
kpss.test(time_series, null="Trend")

# Question 2

# differantiating the time series
diff_time_series <- diff(time_series, 1)
plot(diff_time_series)

# linear model
summary(lm(diff_time_series ~ seq(1, length(diff_time_series))))

# adf and kpss test
adf <- adfTest_valid(diff_time_series,24, type="nc")
adf
kpss.test(diff_time_series)

# Question 3

plot(cbind(time_series, diff_time_series))

```

```
## Part II ##

# Question 4

# acf and pacf
par(mfrow=c(1,2))
acf(diff_time_series);pacf(diff_time_series)

q_max <- 1
p_max <- 4

#LB tests for orders 1 to 24
arima401 <- arima(diff_time_series,c(4,0,1))
Box.test(arima401$residuals, lag=6, type="Ljung-Box", fitdf=5)

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)$p.value
    return(c("lag"=l,"pval"=pval))
  })
  return(t(pvals))
}
Qtests(arima401$residuals, 24, 5)

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))
}
signif(arima401)

#test of all the possible models
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("coefficients nullity tests :\n")
  print(adjust)
  cat("\n tests of autocorrelation of the residuals : \n")
  print(pvals)
}

estim <- arima(diff_time_series,c(1,0,0)); arimafit(estim)
estim <- arima(diff_time_series,c(2,0,0)); arimafit(estim)
estim <- arima(diff_time_series,c(3,0,0)); arimafit(estim)
estim <- arima(diff_time_series,c(4,0,0)); arimafit(estim)
estim <- arima(diff_time_series,c(0,0,1)); arimafit(estim)
estim <- arima(diff_time_series,c(1,0,1)); arimafit(estim)
estim <- arima(diff_time_series,c(2,0,1)); arimafit(estim)
estim <- arima(diff_time_series,c(3,0,1)); arimafit(estim)

# AIC and BIC
mat <- matrix (NA, nrow=p_max+1, ncol=q_max+1)
rownames(mat) <- paste0("p=",0:p_max)
colnames(mat) <- paste0("q=",0:q_max)
AICs <- mat #
BICs <- mat
pqs <- expand.grid(0:p_max, 0:q_max)
for (row in 1:dim(pqs)[1]){
  p <- pqs[row, 1]
  q <- pqs[row, 2]
  estim <- try(arima(diff_time_series, c(p, 0, q), include.mean = F))
  AICs[p+1,q+1] <- if (class(estim)=="try-error") NA else estim$aic
  BICs[p+1,q+1] <- if (class(estim)=="try-error") NA else BIC(estim)
}

AICs
AICs==min(AICs)
BICs
BICs==min(BICs)

```

```r
# final model
arma40 <- arima(diff_time_series, c(4, 0, 0), include.mean=F)
arma40

#adjusted R2
adj_r2 <- function(model){
  ss_res <- sum(model$residuals^2)
  p <- model$arma[1]
  q <- model$arma[2]
  ss_tot <- sum(diff_time_series[-c(1:max(p, q))]^2)
  n <- model$nobs-max(p, q)
  adj_r2 <- 1-(ss_res/(n-p-q-1)) / (ss_tot/(n-1))
  return (adj_r2)
}
adj_r2(arma40)

# Question 5

arima410 <- arima(time_series, c(4, 1, 0), include.mean=F)
arima410

plot(time_series, xlab="Date" , ylab="Indice", main = "Observed vs. Predicted" )
lines(fitted(arima410), col = "red")
plot(diff_time_series, xlab="Date", ylab="Indice", main="Observed vs. Predicted" )
lines(fitted(arma40), col = "red")

## Part III ##

# Question 7

# discussion on the gaussian hypothesis
tsdiag(arma40)
jarque.bera.test(arma40$residuals)
qqnorm(arma40$residuals)
qqline(arma40$residuals, col = "red")
plot(density(arma40$residuals), xlim=c(-10,10), main="Density of residuals")
mu <- mean(arma40$residuals)
sigma <- sd(arma40$residuals)
x <- seq(-10,10)
y <- dnorm(x,mu,sigma)
lines(x, y, lwd=0.5, col="blue")

arma40$coef
phi_1 <- as.numeric(arma40$coef[1])
phi_2 <- as.numeric(arma40$coef[2])
phi_3 <- as.numeric(arma40$coef[3])
phi_4 <- as.numeric(arma40$coef[4])
sigma2 <- as.numeric(arma40$sigma)
phi_1
phi_2
phi_3
phi_4
sigma2

# checking of the roots
ar_coefs <- c(phi_1, phi_2, phi_3, phi_4)
ar_roots <- polyroot(c(1, -ar_coefs))
abs(ar_roots)
all(abs(ar_roots) > 1)

# Question 8

# prediction
XT1 = predict(arma40, n.ahead=2)$pred[1]
XT2 = predict(arma40, n.ahead=2)$pred[2]
XT1
XT2

fore = forecast(arma40, h=5, level=95)
par(mfrow=c(1,1))
plot(fore, xlim=c(2016,2021), col=1, fcol=2, shaded=TRUE, xlab="Time" , ylab="Value", main="Forecast for the time series")

# Bivariate confidence region
mean <- c(XT1, XT2)
sigma2 <- arma40$sigma2
phi_1 <- arma40$coef[1]
cov_matrix <- matrix(c(sigma2, sigma2*phi_1, sigma2*phi_1, sigma2*(1+phi_1*phi_1)), nrow=2)

alpha <- 0.05
chi2_val <- qchisq(1 - alpha, df = 2)

ellipse_points <- ellipse(cov_matrix, centre = mean, level = 1 - alpha)
plot(ellipse_points, type = 'l', xlab = 'Forecast for X_{T+1}', ylab = 'Forecast for X_{T+2}', main = '95 % bivariate confidence region', col = 'red')
points(mean[1], mean[2], pch = 19)
abline(h=XT2,v=XT1, col="blue")
abline(h=0,v=0)
grid()
```