Project Report: Linear Time Series

Second year of the engineering program

ARIMA Modelling and Forecasting of the French Automobile Industry Production Index

---

Evan DJIEMON
Arold TOUBERT

Contents

1 The Data

1.1 Presentation of the Series

In this project, we choose to study the **French Industrial Production Index (IPI) for the automobile industry**, published by INSEE. The series is availiable here: INSEE Website. The IPI measures the monthly evolution of the production for every industry, and is calculated in base 100 (normalized so that a reference period equals 100, and other values show percentage changes relative to that base). Furthermore, we take it corrected from seasonal variations and working days (CVS-CJO). Here, we study the aggregate series for the automobile industry, which includes the production of automobile vehicles and automobile equipments.

Figure 1: Evolution of the Automobile Industry IPI over time

As we can see on Figure 1, the series exhibits two fairly distinct regimes, before and after the financial crisis in 2008, where an important spike can be seen. For this reason, we choose to only select data starting in January 2010 and ending in March 2025, in order to have homogeneous data and better results for our modeling.

An another important finding is the outlier in early 2020 due to the COVID Crisis, which importantly affected the automobile industry. Hence, we may have to correct this outlier later in the project.

1.2 Stationarity Analysis

Table 1: Linear Regression Results: IPI on dates_numeric

Variable	**Estimate**	**Std. Error**	**t value**	**Pr(>|t|)**
(Intercept)	3055.5852	551.3749	5.542	1.04e-07
dates_numeric	-1.4554	0.2733	-5.326	2.96e-07

Firstly, looking at Figure 1, we can already state that **the series does not seem to be stationary**, with a non stationary mean and variance, and we can also see **a clear downward linear trend** of the series starting in 2010. This is confirmed using a Linear Regression of the IPI on the dates converted into a numeric series. The results in Table 1 show that there is indeed a downward trend which seems to be significant, even if cannot confirm significancy because the test is not valid, where there are possibly autocorrelated residuals. However, it

indicates that we should include the parameter type="ct" into our Augmented Dickey Füller (ADF) tests, to include a constant and a linear trend into our tests. Moreover, if differentiation is needed to make the series stationary, it will probably correct this trend.

Then, before performing ADF tests, we need to ensure that the residuals from the ADF regression are free of autocorrelation. To do so, we incrementally search for the smallest lag order (up to 24 lags, i.e. 2 years), such that Ljung-Box tests on the residuals are all non-significant. The ADF test is rerun at each step until this condition is met, and then the optimal lag order is chosen.

Table 2: Augmented Dickey-Fuller Test Results on IPI

Parameter	**Value**
Lag Order	5
Dickey-Fuller Statistic	-3.0421
P-value	0.1405

Table 3: Augmented Dickey-Fuller Test Results on dIPI

Parameter	**Value**
Lag Order	4
Dickey-Fuller Statistic	-8.175
P-value	0.01

Table 2 shows that the optimal lag order found is 5, but that the **IPI series is not stationary**, with a p-value of 0.1405, which shows no significancy at every usual level. Hence, **we differentiate our series to obtain dIPI**. Following the same methodology to find the optimal lag order, we perform an ADF test on the differentiated series dIPI, resumed in Table 3. This time, the p-value is 0.01, which shows that **the differentiated series is stationary** at every usual level.

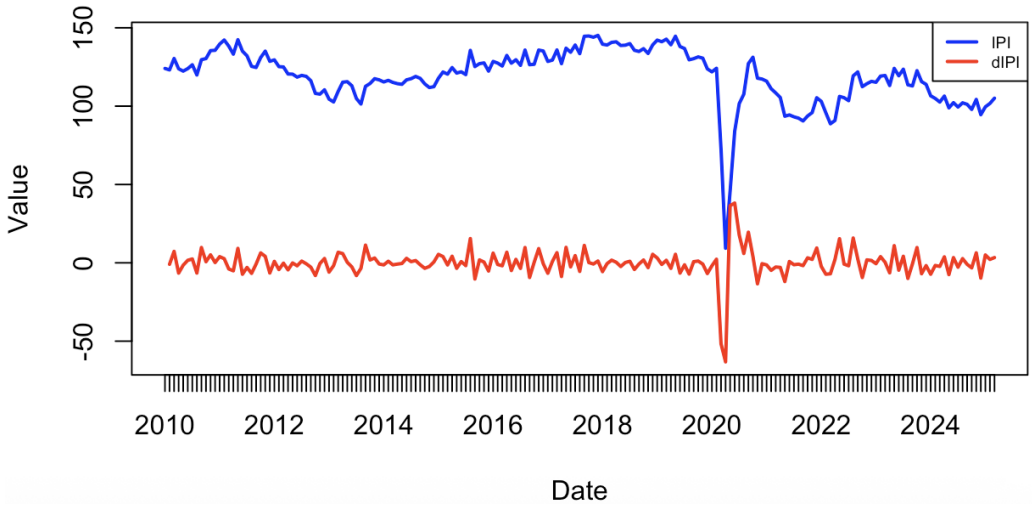

Figure 2: Evolution of IPI and dIPI over time

This is confirmed using Figure 2, where we can see that the series dIPI fluctuates around 0 (stationarity in mean), it is more stationary in variance than IPI, and it corrected the trend from IPI. Hence, IPI is $I(1)$.

2 ARMA Model Selection

In order to find the best ARMA(p, q) model for the differenciated series, we should first find the maximal orders p_{max} and q_{max}, using the plot of the ACF and the PACF of dIPI.

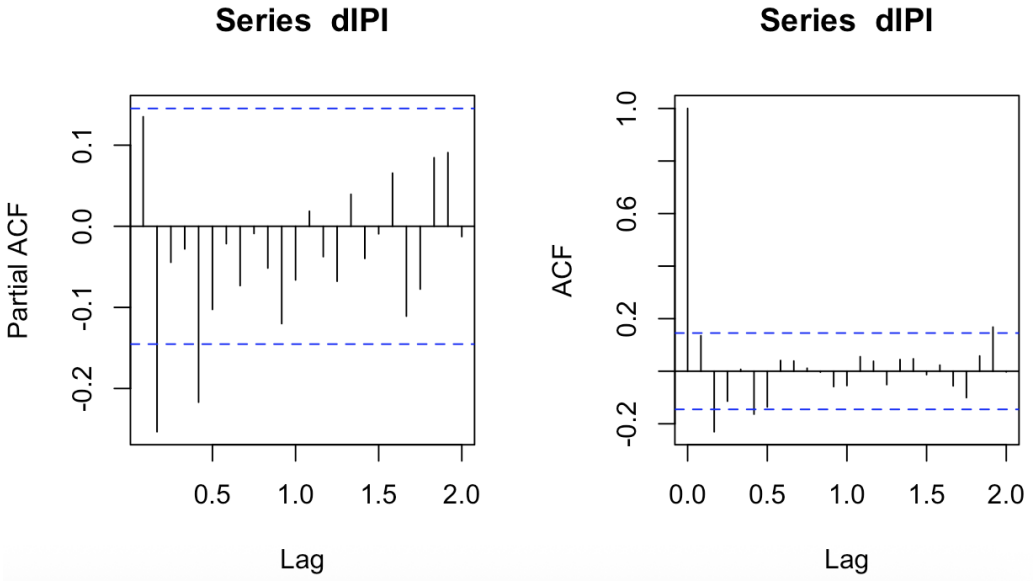

Figure 3: PACF and ACF of dIPI

Looking at Figure 3, we can see that for the PACF, there is no significant peak different from 0 after the fifth peak, which lead us to choose $p_{max} = 5$. For the ACF, there is no significant peak after the second peak, and, even if there are other peaks such as for q=5, we choose $q_{max} = 2$ for parcimony reasons.

To select the best (p, q) orders, we incrementally estimate all ARMA(p, q) models on dIPI with $p <= p_{max}$ and $q <= q_{max}$, and retain only those where all AR and MA coefficients are significant at a 5% level, and where residuals show no autocorrelation (using Ljung-Box tests, up to lag 24). Then, among valid and well-adjusted models, the one minimizing the AIC and the BIC criteria is selected.

Table 4: ARMA Models Comparison

Criterion	**ARMA(5, 0)**	**ARMA(1, 2)**
AIC	1307.916	1304.009
BIC	1330.344	1320.029

The results in Table 4 show that only two models were validated and well-adjusted: ARMA(5, 0) and ARMA(1, 2). We choose the model minimizing the criteria, which is ARMA(1, 2).

Then the ARIMA(1, 1, 2) adjusted for IPI is given by:

$$IPI_t - IPI_{t-1} - \phi_1(IPI_{t-1} - IPI_{t-2}) = \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2}$$

where ϕ_1 is the AR(1) parameter, θ_1, θ_2 are the MA(1) and MA(2) parameters, and $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$ is a white noise.

3 Prediction

Let T denote the length of the corrected time series. We suppose that the corrected series X_t follows an ARMA(1,2) model, and that the residuals (ε_t) are independent and identically distributed Gaussian variables, i.e., $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$. The model is given by the equation

$$X_t = \phi_1 X_{t-1} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2}.$$

3.1 Confidence region of level α

We aim to construct a confidence region of level α for the vector of future values (X_{T+1}, X_{T+2}), conditionally on all available information up to time T. Denote $\hat{X}_{T+1|T} = E[X_{T+1}|\mathcal{F}_T]$ and $\hat{X}_{T+2|T} = E[X_{T+2}|\mathcal{F}_T]$ the conditional expectations, which serve as point forecasts. Assuming that the past residuals ε_T, ε_{T-1}, etc., have been estimated, the forecast for time $T+1$ is

$$\hat{X}_{T+1|T} = \phi_1 X_T + \theta_1 \varepsilon_T + \theta_2 \varepsilon_{T-1}.$$

At time $T+2$, using the law of iterated expectations and the independence of future residuals, we obtain

$$\hat{X}_{T+2|T} = \phi_1 \hat{X}_{T+1|T} + \theta_2 \varepsilon_T.$$

Let us now introduce the vector of forecast errors:

$$e_{T+1} = X_{T+1} - \hat{X}_{T+1|T} = \varepsilon_{T+1}, \quad e_{T+2} = X_{T+2} - \hat{X}_{T+2|T} = (\phi_1 + \theta_1)\varepsilon_{T+1} + \varepsilon_{T+2}.$$

Let $a = \phi_1 + \theta_1$. Then the forecast error vector is

$$\begin{bmatrix} e_{T+1} \\ e_{T+2} \end{bmatrix} = \begin{bmatrix} \varepsilon_{T+1} \\ a\varepsilon_{T+1} + \varepsilon_{T+2} \end{bmatrix},$$

which is a linear transformation of two independent Gaussian variables. Its covariance matrix is

$$\Sigma = \sigma^2 \begin{bmatrix} 1 & a \\ a & 1 + a^2 \end{bmatrix}.$$

Before inverting this matrix, we must ensure it is invertible. We explicitly assume $\sigma^2 > 0$, which implies that the determinant of Σ is strictly positive:

$$\det(\Sigma) = \sigma^4(1 + a^2 - a^2) = \sigma^4 > 0.$$

This guarantees that Σ is positive definite and thus invertible. As a result, the vector (X_{T+1}, X_{T+2}) follows a bivariate normal distribution with mean vector $\hat{X} = \begin{bmatrix} \hat{X}_{T+1|T} \\ \hat{X}_{T+2|T} \end{bmatrix}$ and covariance matrix Σ. Therefore, $(X - \widehat{X})^\top \Sigma^{-1} (X - \widehat{X}) \sim \chi^2(2)$. The confidence region of level α is thus defined by the inequality:

$$\boxed{(X - \hat{X})' \Sigma^{-1} (X - \hat{X}) \leq \chi_2^2(1 - \alpha)},$$

where $\chi_2^2(1-\alpha)$ denotes the $(1-\alpha)$-quantile of the $\chi^2(2)$ distribution.

3.2 Hypotheses

To derive the confidence region for the future values (X_{T+1}, X_{T+2}), several assumptions are made throughout the reasoning. First, we assume that the time series (X_t) follows an ARMA(1,2) process with known parameters $\phi_1, \theta_1, \theta_2$, and that the series is observed up to time T. Second, we suppose that the residuals (ε_t) are independent and identically distributed Gaussian random variables with mean zero and variance σ^2, which ensures that the forecast errors are also Gaussian. In particular, we make the additional assumption that $\sigma^2 > 0$, so that the covariance matrix of the error vector is strictly positive definite and hence invertible. Furthermore, we assume that the past values of the process and the corresponding residuals have been estimated or are available, so that the conditional expectations $\hat{X}_{T+1|T}$ and $\hat{X}_{T+2|T}$ can be computed. Finally, the derivation relies on the fact that the conditional distribution of the future vector (X_{T+1}, X_{T+2}) given the past is multivariate normal, which allows us to use the Mahalanobis distance and the associated chi-squared distribution to define a confidence region.

To test the normality of the residuals, we use the Jarque Bera test (using the library tseries), which gives us **a p-value of 2.2e-16**, which validates the hypothesis at all levels.

3.3 Graphic representation

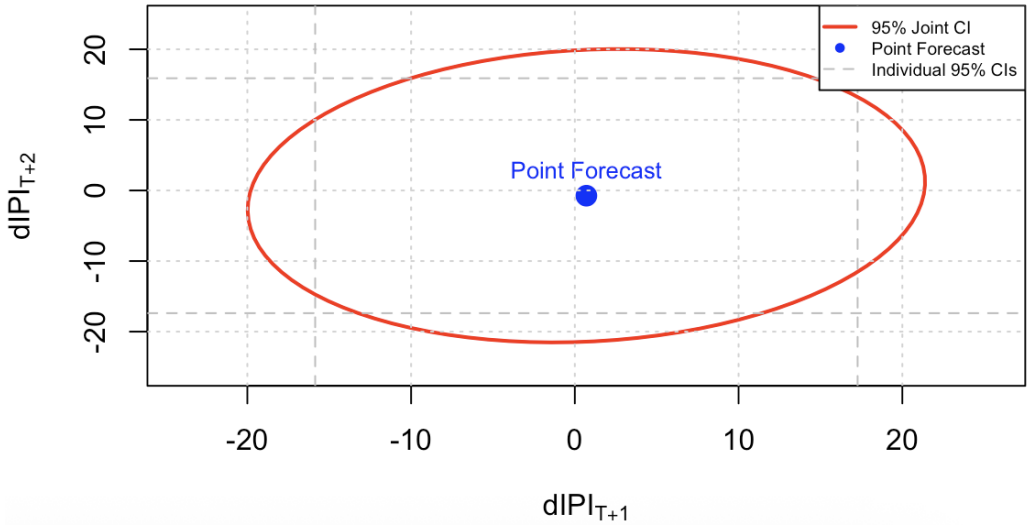

Figure 4: 95% confidence region for $dIPI_{t+1}$ and $dIPI_{t+2}$

If we look at Figure 4, the elliptical shape demonstrates that the 95% joint confidence region is more restrictive than the individual 95% confidence intervals (shown by the gray dashed lines). This is expected because joint confidence accounts for the simultaneous uncertainty in both forecasts, while individual intervals consider each forecast separately. The relatively wide confidence region indicates substantial uncertainty in the two-period-ahead forecasts of dIPI.

3.4 Open question

Let (X_t, Y_t) be a stationary bivariate process observed for $t = 1, \ldots, T$, and assume that the value of Y_{T+1} becomes available slightly before X_{T+1}. The question is whether this early observation of Y_{T+1} can help improve the forecast of X_{T+1}.

According to the course, this is possible if and only if there is **instantaneous causality from Y to X in the sense of Granger**. This means that, conditionally on the information

set $\mathcal{F}_T$ available at time T, the knowledge of Y_{T+1} provides additional predictive power for X_{T+1}. Formally, we require that:

$$E\left[X_{T+1} \mid \mathcal{F}_t, Y_{t+1}\right] \neq E\left[X_{T+1} \mid \mathcal{F}_T\right]$$

This condition is equivalent to:

$$\mathrm{Cov}(\varepsilon_{X,T+1}, \varepsilon_{Y,T+1}) \neq 0,$$

where $\varepsilon_{X,t}$ and $\varepsilon_{Y,t}$ are the reduced form innovations from the bivariate system:

$$\Phi(L) \begin{bmatrix} X_t \\ Y_t \end{bmatrix} = \begin{bmatrix} \varepsilon_{X,t} \\ \varepsilon_{Y,t} \end{bmatrix}, \quad \text{with} \quad \begin{bmatrix} \varepsilon_{X,t} \\ \varepsilon_{Y,t} \end{bmatrix} \sim \mathcal{N}(0, \Sigma).$$

This non-zero covariance Σ_{XY} between the contemporaneous innovations characterizes the presence of instantaneous Granger causality from Y to X. If $\Sigma_{XY} = 0$, then Y_{T+1} brings no new information for forecasting X_{T+1}, beyond what is already contained in $\mathcal{F}_T$.

To test this condition, the course proposes a Wald-type test. After estimating the reduced VAR system over $t = 1, \ldots, T$, one computes the residuals $\hat{\varepsilon}_{X,t}$ and $\hat{\varepsilon}_{Y,t}$, and the empirical covariance matrix $\hat{\Sigma}$. The null hypothesis of no instantaneous causality is:

$$H_0 : \Sigma_{XY} = 0.$$

The Wald statistic is then given by:

$$W = n \cdot \frac{\hat{\Sigma}_{XY}^2}{\hat{\Sigma}_{XX} \cdot \hat{\Sigma}_{YY}},$$

where n is the number of effective observations. Under H_0, this statistic asymptotically follows a chi-squared distribution with one degree of freedom:

$$W \xrightarrow{d} \chi_1^2.$$

We reject H_0 at level α if $W > \chi_1^2(1 - \alpha)$.

4 To Go Further: Correcting the Covid Outlier

To go further in our study, we decide to examine the outlier present in our series in early 2020 (March and April 2020). Not correcting this outlier could have a direct impact on our ARIMA models, as it leads to biases in the estimation of their parameters, and hence, poor predictions. One way to correct this effect is to use dummy variables as external regressor, to explicit the shock in the model's structure, avoiding biases in parameters estimation. This leads us to add two dummy variables as external regressor in our modeling: one for March 2020, and one for April 2020.

Table 5: ARIMA Model Comparison with COVID Correction

Criterion	**ARMA(5, 0)**	**ARMA(5, 1)**
AIC	1212.312	1212.244
BIC	1237.944	1241.080

Using the same methodology as before, we choose the valid (p, q) parameters iteratively, and find the ones minimizing the AIC and the BIC criteria. Here, we choose an ARMA(5, 0).

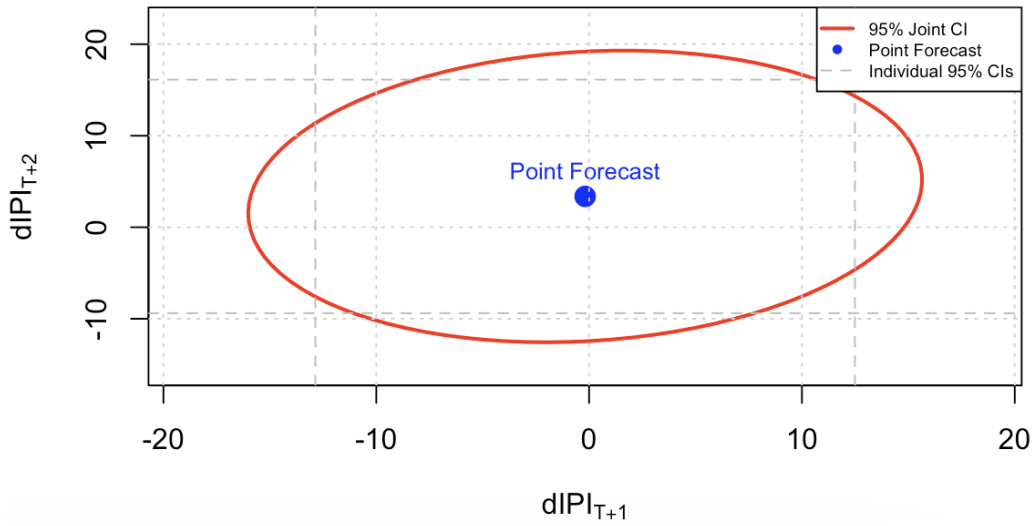

Figure 5: 95% confidence region for $dIPI_{t+1}$ and $dIPI_{t+2}$

Now, if we plot the new 95% confidence region for $dIPI_{t+1}$ and $dIPI_{t+2}$ (the Jarque Bera test gives a p-value of 2.2e-16), we can see that we have now more precision in our forecasting, as its size is smaller than before.

Table 6: RMSE Comparison - with and without COVID Correction

Modèle	**RMSE**
ARMA(1, 2) (without COVID)	6.942822
ARMA(5, 0) (with COVID)	6.072725

Finally, if we compare the RMSE between the best model found with and without the outlier correction, we see an important improvement in the RMSE with the use of dummy variables, highlighting the need of correcting outliers when modeling ARIMA models.

A R Code

```
path <- "/Users/aroldtoubert/Downloads/serie_010768301_30052025"
setwd(path)


# Loading the data
datafile <- "valeurs_mensuelles.csv"

#delete the headers
data <- read.csv(datafile, sep=";", skip = 4, header = FALSE)
data <- data[, 1:2]

# Rename the columns for simpler use
colnames(data) <- c("date", "IPI")

#ordering the data using date
data <- data[order(data$date), ]



require(zoo)
dates <- as.yearmon(data$date)
IPI <- zoo(data$IPI, order.by=dates)

#Plot the series before cutting
plot(index(IPI), IPI, type="l", col="blue", lwd=2,
     xlab="Date", ylab="IPI", main="Industrial Production Index over
        time")

# Filter to select only data starting in January 2010 after 2008
   financial crisis
IPI <- window(IPI, start = as.yearmon("Jan 2010"), end=as.yearmon("Mar
   2025"))

filtered_dates <- index(IPI)

#Plotting the IPI and its differenciated series
dIPI <- diff(IPI, 1)

y_range <- range(c(as.numeric(IPI), as.numeric(dIPI)), na.rm = TRUE)
plot(index(IPI), IPI, type="l", col="blue", lwd=2,
     xlab="Date", ylab="Value", main="IPI and dIPI over time",
     ylim = y_range)
lines(index(dIPI), dIPI, col="red", lwd=2)
legend("topright", legend=c("IPI", "dIPI"), col=c("blue", "red"), lwd
   =2, cex=0.6)


# Convert to numeric for regression to detect trends
dates_numeric <- as.numeric(filtered_dates)
```

```
#Regression to detect trends
summary(lm(IPI ~ dates_numeric))


require(fUnitRoots)
adf <- adfTest(IPI, lag=0, type="ct") #Using type=ct to include
   constant and linear trend

#First we need to test the residuals autocorrelation for lags until 24
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(adf@test$lm$residuals, 24, fitdf = length(adf@test$lm$
   coefficients))

#Find the minimum lag order without residual autocorrelation in ADF
series <- IPI; kmax <- 24; adftype="ct"
adfTest_valid <- function(series, kmax, adftype){
  k <- 0
  noautocorr <- 0
  while (noautocorr==0){
    cat(paste0("ADF with ",k," lags: residuals OK? "))
    adf <- adfTest(series, lags=k, type=adftype)
    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(IPI,24,adftype="ct")

#Run the ADF test ==> not stationary, we try the differenciated series
   dIPI
adf


filtered_dates_diff <- index(dIPI)
dates_numeric_diff <- as.numeric(filtered_dates_diff)

#Run the LR to test trend ==> no significancy ==> trend corrected by
   differenciation
```

```
summary(lm(dIPI ~ dates_numeric_diff))

#This time, we run with type="nc", as the trend is corrected with the
   differentiation
adf <- adfTest_valid(dIPI,24,"nc")
Qtests(adf@test$lm$residuals, 24, fitdf = length(adf@test$lm$
   coefficients))

#ADF test: stationary
adf


#Plot ACF and PACF on the stationary series
par(mfrow=c(1,2))
pacf(dIPI,24);acf(dIPI,24)
library(lmtest)
pmax=5; qmax=2

####### WITHOUT COVID CORRECTION #############

#Function to estimate ARMA(p, q) models, and check adjustment and
   validity
modelchoice <- function(p,q,data=dIPI, k=24){
  estim <- try(arima(data, c(p, 0,q),optim.control=list(maxit=20000)))
  if (class(estim)=="try-error") return(c("p"=p,"q"=q,"arsignif"=NA,"
     masignif"=NA,"resnocorr"=NA, "ok"=NA))
  arsignif <- if (p==0) NA else coeftest(estim)[1:p, 4][p] <= 0.05
  masignif <- if (q==0) NA else coeftest(estim)[(p+1):(p+q), 4][q] <=
     0.05
  resnocorr <- sum(Qtests(estim$residuals,24,length(estim$coef)-1)
     [,2]<=0.05,na.rm=T)==0
  checks <- c(arsignif,masignif,resnocorr)
  ok <- as.numeric(sum(checks,na.rm=T)==(3-sum(is.na(checks))))
  return(c("p"=p,"q"=q,"arsignif"=arsignif,"masignif"=masignif,"
     resnocorr"=resnocorr,"ok"=ok))
}


#Function to evaluate and validate all the arma(p, 0, q) for p<=pmax
   and q<=qmax on dIPI
armamodelchoice <- function(pmax,qmax){
  pqs <- expand.grid(0:pmax,0:qmax)
  t(apply(matrix(1:dim(pqs)[1]),1,function(row) {
    p <- pqs[row,1]; q <- pqs[row,2]
    cat(paste0("Computing ARMA(",p,",",q,") \n"))
    modelchoice(p,q)
  }))
}

armamodels <- armamodelchoice(pmax,qmax)
```

```
colnames(armamodels) <- c("p", "q", "arsignif", "masignif", "resnocorr"
   , "ok")
selec <- armamodels[armamodels[,"ok"]==1&!is.na(armamodels[,"ok"]),]
#Selec contains all the valid models
selec

#Among the valid models, choose the one minimizing the AIC and BIC
   criteria
pqs <- apply(selec,1,function(row) list("p"=as.numeric(row[1]),"q"=as.
   numeric(row[2])))
names(pqs) <- paste0("arima(",selec[,1],", 0,",selec[,2],")")
models <- lapply(pqs, function(pq) arima(dIPI, c(pq[["p"]], 0, pq[["q"
   ]])))
vapply(models, FUN.VALUE=numeric(2), function(m) c("AIC"=AIC(m),"BIC"=
   BIC(m)))

#Estimate the RMSE out of sample to evaluate the model
n_obs <- length(dIPI)
dIPI_train <- window(dIPI, end = index(dIPI)[n_obs-4])
dIPI_test <- tail(dIPI, 4)

models_train <- lapply(pqs, function(pq) arima(dIPI_train, c(pq[["p"]],
    0, pq[["q"]])))

forecasts <- lapply(models_train, function(m) as.zoo(predict(m,4)$pred)
   )

rmse <- vapply(forecasts, FUN.VALUE=numeric(1), function(forecast) {
  sqrt(mean((forecast - dIPI_test)^2))
})

rmse

#Best model according to the criteria: ARMA(1, 2)
selected_model_no_covid <- arima(dIPI, c(1, 0, 2))

library(tseries)

#Test of the normality of the residuals using Jarque Bera test
jb_test_no_covid <- jarque.bera.test(selected_model_no_covid$residuals)
print(jb_test_no_covid)


#Extract parameters
phi1 <- selected_model_no_covid$coef["ar1"]
theta1 <- selected_model_no_covid$coef["ma1"]
sigma_sq_no_covid <- selected_model_no_covid$sigma2

#Calculate a=      +      , as in the report
a <- phi1 + theta1
```

```
#Determine the covariance matrix
Sigma_no_covid <- sigma_sq_no_covid * matrix(c(1, a, a, 1 + a^2), nrow
   = 2)

#Estimate forecasts for dIPI without COVID correction
forecasts_no_covid <- predict(selected_model_no_covid, n.ahead = 2)
X_hat_no_covid <- forecasts_no_covid$pred

par(mfrow = c(1,1))
library(ellipse)

#Generate ellipse points using the correct covariance matrix
ellipse_points_no_covid <- ellipse(Sigma_no_covid, centre = as.numeric(
   X_hat_no_covid), level = 0.95)

#Plot the 95% confidence region
plot(ellipse_points_no_covid, type = 'l', lwd = 2, col = 'red',
     xlab = expression(dIPI[T+1]), ylab = expression(dIPI[T+2]),
     main = "95% Confidence Region for dIPI",
     xlim = range(ellipse_points_no_covid[,1]) + c(-0.1, 0.1)*diff(
        range(ellipse_points_no_covid[,1])),
     ylim = range(ellipse_points_no_covid[,2]) + c(-0.1, 0.1)*diff(
        range(ellipse_points_no_covid[,2])))

#Add the forecast
points(X_hat_no_covid[1], X_hat_no_covid[2], pch = 19, col = 'blue',
   cex = 1.5)
text(X_hat_no_covid[1], X_hat_no_covid[2], "Point Forecast", pos = 3,
   col = 'blue', cex = 0.8)

grid()

#Add individual confidence intervals using correct standard errors
se_1_corrected <- sqrt(Sigma_no_covid[1,1])
se_2_corrected <- sqrt(Sigma_no_covid[2,2])

ci_1_no_covid <- X_hat_no_covid[1] + c(-1, 1) * qnorm(0.975) * se_1_
   corrected
ci_2_no_covid <- X_hat_no_covid[2] + c(-1, 1) * qnorm(0.975) * se_2_
   corrected

abline(v = ci_1_no_covid, lty = 2, col = 'gray')
abline(h = ci_2_no_covid, lty = 2, col = 'gray')

legend("topright",
       legend = c("95% Joint CI", "Point Forecast", "Individual 95% CIs
          "),
       col = c("red", "blue", "gray"),
       lty = c(1, NA, 2),
       pch = c(NA, 19, NA),
       lwd = c(2, NA, 1),
```

```
       cex = 0.6)



######### WITH COVID (using dummy variables) ########

#Creation of dummy variables for March and April 2020, where covid
   effect is strong
covid_dates <- c(as.yearmon("Mar 2020"), as.yearmon("Apr 2020"))
covid_indicator_dIPI <- as.numeric(index(dIPI) %in% covid_dates)
covid_xreg_dIPI <- matrix(covid_indicator_dIPI, ncol=1) #We need to
   take into account the fewer amount of observation
colnames(covid_xreg_dIPI) <- "COVID"

#Modification to use the dummy variables as external regressors in our
   ARMA models for dIPI
modelchoice_covid <- function(p,q,data=dIPI, xreg=covid_xreg_dIPI, k
   =24){
  estim <- try(arima(data, c(p, 0,q), xreg=xreg, optim.control=list(
     maxit=20000)))
  if (class(estim)=="try-error") return(c("p"=p,"q"=q,"arsignif"=NA,"
     masignif"=NA,"resnocorr"=NA, "ok"=NA))
  n_xreg <- if(is.null(xreg)) 0 else ncol(xreg)
  coef_names <- names(estim$coef)
  arsignif <- if (p==0) NA else {
    ar_indices <- grep("^ar", coef_names)
    if(length(ar_indices) > 0) coeftest(estim)[ar_indices[p], 4] <=
       0.05 else NA
  }
  masignif <- if (q==0) NA else {
    ma_indices <- grep("^ma", coef_names)
    if(length(ma_indices) > 0) coeftest(estim)[ma_indices[q], 4] <=
       0.05 else NA
  }
  resnocorr <- sum(Qtests(estim$residuals,24,length(estim$coef)-1)
     [,2]<=0.05,na.rm=T)==0
  checks <- c(arsignif,masignif,resnocorr)
  ok <- as.numeric(sum(checks,na.rm=T)==(3-sum(is.na(checks))))
  return(c("p"=p,"q"=q,"arsignif"=arsignif,"masignif"=masignif,"
     resnocorr"=resnocorr,"ok"=ok))
}

#Modification to use the new modelchoice function for dIPI
armamodelchoice_covid <- function(pmax,qmax, xreg=covid_xreg_dIPI){
  pqs <- expand.grid(0:pmax,0:qmax)
  t(apply(matrix(1:dim(pqs)[1]),1,function(row) {
    p <- pqs[row,1]; q <- pqs[row,2]
    cat(paste0("Computing ARMA(",p,",0,",q,") with COVID indicators for
       dIPI \n"))
    modelchoice_covid(p,q,data=dIPI,xreg=xreg)
  }))
```

```
}

armamodels_covid <- armamodelchoice_covid(pmax,qmax, xreg=covid_xreg_
   dIPI)
colnames(armamodels_covid) <- c("p", "q", "arsignif", "masignif", "
   resnocorr", "ok")
selec_covid <- armamodels_covid[armamodels_covid[,"ok"]==1&!is.na(
   armamodels_covid[,"ok"]),]
selec_covid

pqs_covid <- apply(selec_covid,1,function(row) list("p"=as.numeric(row
   [1]),"q"=as.numeric(row[2])))
names(pqs_covid) <- paste0("arima(",selec_covid[,1],",0,",selec_covid
   [,2],")")

#Estimate the models with indicators for dIPI
models_covid <- lapply(pqs_covid, function(pq) arima(dIPI, c(pq[["p"]],
    0, pq[["q"]]), xreg=covid_xreg_dIPI))
vapply(models_covid, FUN.VALUE=numeric(2), function(m) c("AIC"=AIC(m),"
   BIC"=BIC(m)))

#RMSE out of sample evaluation for dIPI with COVID
n_obs_dIPI <- length(dIPI)
dIPI_train <- window(dIPI, end = index(dIPI)[n_obs_dIPI-4])
dIPI_test <- tail(dIPI, 4)

covid_xreg_dIPI_train <- covid_xreg_dIPI[1:(n_obs_dIPI-4), , drop=FALSE
   ]
covid_xreg_dIPI_test <- covid_xreg_dIPI[(n_obs_dIPI-3):n_obs_dIPI, ,
   drop=FALSE]

models_train_covid <- lapply(pqs_covid, function(pq) arima(dIPI_train,
   c(pq[["p"]], 0, pq[["q"]]), xreg=covid_xreg_dIPI_train))

forecasts_covid <- lapply(models_train_covid, function(m) as.zoo(
   predict(m, 4, newxreg=covid_xreg_dIPI_test)$pred))

rmse_covid <- vapply(forecasts_covid, FUN.VALUE=numeric(1), function(
   forecast) {
  sqrt(mean((forecast - dIPI_test)^2))
})

rmse_covid

#Fit the AR(5) model with COVID dummy
selected_model_covid <- arima(dIPI, order = c(5, 0, 0), xreg = covid_
   xreg_dIPI)

#Test of the normality of the residuals using Jarque Bera Test
jb_test_covid <- jarque.bera.test(selected_model_covid$residuals)
print(jb_test_covid)
```

```
#Extract AR coefficients
phi_coeffs <- selected_model_covid$coef[grep("^ar", names(selected_
   model_covid$coef))]
sigma_sq_covid <- selected_model_covid$sigma2

phi1 <- phi_coeffs[1]

#Covariance matrix for AR(5) 2-step forecasts (using same calculation
   as in the report but for AR(5))
Sigma_covid <- sigma_sq_covid * matrix(c(1, phi1, phi1, 1 + phi1^2),
   nrow = 2)

#Same code than without covid correction
future_covid_xreg_dIPI <- matrix(0, nrow = 2, ncol = 1)
colnames(future_covid_xreg_dIPI) <- "COVID"

forecasts_covid_final <- predict(selected_model_covid, n.ahead = 2,
   newxreg = future_covid_xreg_dIPI)
X_hat_covid <- forecasts_covid_final$pred

ellipse_points_covid <- ellipse(Sigma_covid, centre = as.numeric(X_hat_
   covid), level = 0.95)

plot(ellipse_points_covid, type = 'l', lwd = 2, col = 'red',
     xlab = expression(dIPI[T+1]), ylab = expression(dIPI[T+2]),
     main = "95% Confidence Region for dIPI\n(With COVID correction)",
     xlim = range(ellipse_points_covid[,1]) + c(-0.1, 0.1)*diff(range(
        ellipse_points_covid[,1])),
     ylim = range(ellipse_points_covid[,2]) + c(-0.1, 0.1)*diff(range(
        ellipse_points_covid[,2])))

points(X_hat_covid[1], X_hat_covid[2], pch = 19, col = 'blue', cex =
   1.5)
text(X_hat_covid[1], X_hat_covid[2], "Point Forecast", pos = 3, col = '
   blue', cex = 0.8)

grid()

se_1_corrected <- sqrt(Sigma_covid[1,1])
se_2_corrected <- sqrt(Sigma_covid[2,2])

ci_1_covid <- X_hat_covid[1] + c(-1, 1) * qnorm(0.975) * se_1_corrected
ci_2_covid <- X_hat_covid[2] + c(-1, 1) * qnorm(0.975) * se_2_corrected

abline(v = ci_1_covid, lty = 2, col = 'gray')
abline(h = ci_2_covid, lty = 2, col = 'gray')

legend("topright",
       legend = c("95% Joint CI", "Point Forecast", "Individual 95% CIs
          "),
```

```
       col = c("red", "blue", "gray"),
       lty = c(1, NA, 2),
       pch = c(NA, 19, NA),
       lwd = c(2, NA, 1),
       cex = 0.6)
```

Listing 1: R Program for the Project