Linear Time Series Project

Authors

CIANFARANI Ana-Sofia
BENABDESADOK Nayel

May 2025

Contents

1 Data

1.1 What does the chosen series represent

The chosen series is a French Industrial Production Index (IPI) Series from INSEE[1]. It represents construction as a sector (Section F of the NAF classification). Its base year is 2021. Data is available on a monthly basis from January 1990 to February 2025, thus containing 422 observations. The series is corrected from seasonal variations and working days (CVS-CJO). The goal is to facilitate conjunctural analysis of industrial production over time (see Appendix B for further detail).

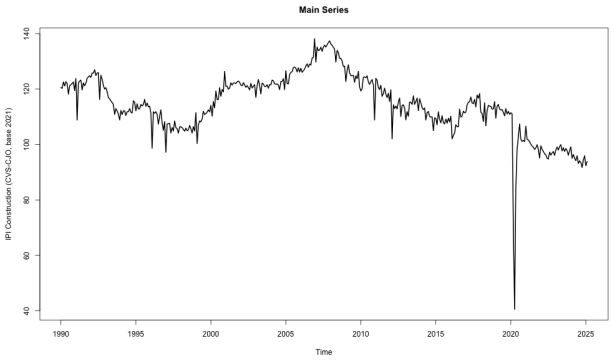


Figure 1: Plot of the Original Series (as a function of time)

The series evolves in an irregular manner, with three main phases. From 1990 to 2000, it decreases, with the level of production reducing by about 20 basis points over this period. This decrease reflects a period of economic difficulties for the construction sector as a whole, which struggled against poor economic conditions such as rising unemployment, high real interest rates, and public debt reduction to prepare for the introduction of the euro. Moreover, a wave of office construction in the early 1990s set off a real estate crisis due to the saturation of the market[INSEE, 2000].

Then, from 2000 to 2008, a net increase of industrial production can be seen, gaining about 30 basis points , thanks to generally favorable economic conditions. From 2008 onwards, the series has decreased regularly.

A slight increase was seen from early 2017, however these gains were wiped out in early 2020, and production has continued to decrease since, with production now at its lowest recorded level (excluding the Covid-era restrictions period) : it has lost 40 basis points since 2008. March & April 2020 stand out as remarkable values in our series, as construction was halved in the span of a month : this was due to the partial shutdown of economic activity linked to the pandemic. This outlier value particularly stands out; however, one can also notice many smaller outliers throughout the series, which will require specific treatment in our analysis so as to ensure they do not overly influence our model.

1.2 Making the series stationary

The series seems to have a deterministic trend and suffer from heteroscedasticity, which is confirmed by analysis of the ACF and the decomposition of the series into trend-cycle, seasonal variations and residuals. Classic stationarity tests also confirm this intuition. The detailed results are available in Appendixes A and C.

As the series has already been corrected, there should not be any seasonality : this is confirmed by a number of graphic tests presented in Appendix B .

To correct for non-stationarity, we take the first difference of the series. It now appears stationary, and this is confirmed by all three classic stationary tests (see Appendix C). We conclude that the series is of integration order 1, $I(1)$.

[1]Full series available at : https://www.insee.fr/fr/statistiques/serie/010767635.

1.3 Graphic Representation

These graphs represent the original time series X_t and its stationary version $Y_t = X_t - X_{t-1}$.

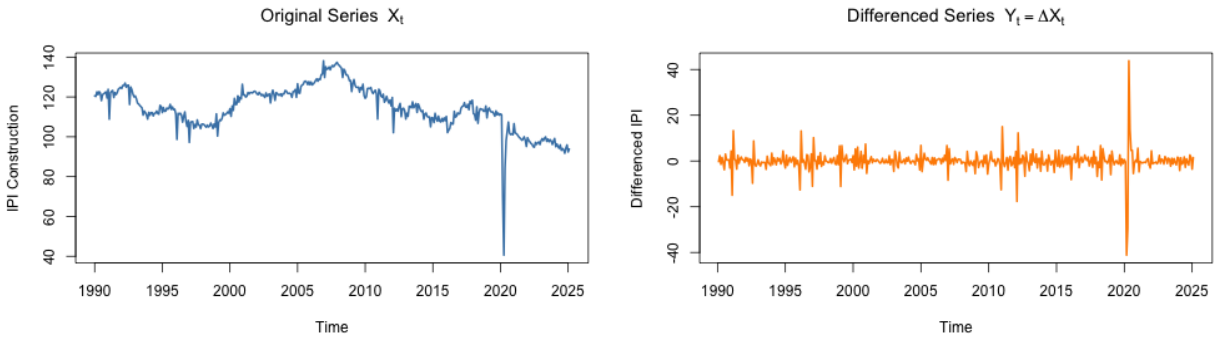


Figure 2: Plot of the Original Series (left) and the Differenced Series (right)

2 ARMA Models

2.1 Choosing the Appropriate ARMA(p, q) Model

After transforming the original series into a stationary one through first-order differencing (i.e., $Y_t = X_t - X_{t-1}$), we aim to identify the optimal AR and MA orders p and q for the ARMA(p,q) model. This step relies on examining the autocorrelation function (ACF) and the partial autocorrelation function (PACF).

From Figure 3, we can assess the statistical significance of the autocorrelation and partial autocorrelation coefficients. Values lying outside the bounds $\pm 1.96/\sqrt{n}$ exceed the 95% confidence interval under the null hypothesis of zero correlation, and are thus considered statistically significant.

- The **ACF** is used to identify the MA order q: for an MA(q) process, autocorrelations drop to zero after lag q.
- The **PACF** is used to identify the AR order p: for an AR(p) process, partial autocorrelations drop to zero after lag p.

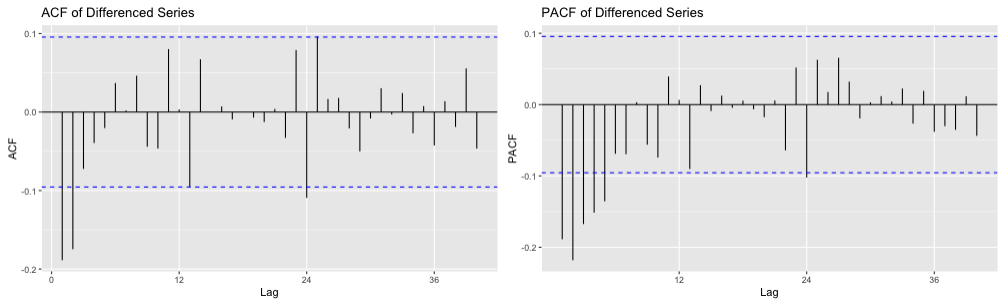


Figure 3: ACF (left) and PACF(right) of the differenced series Y_t.

Note: The dashed blue line represents the 95% significance bounds.

Based on Figure 3, we identify a maximum MA order $q_{\max} = 2$ (cutoff after lag 2) and a maximum AR order $p_{\max} = 5$ (cutoff after lag 5). Some higher-order lags exceed the confidence bounds, but they occur after long sequences of non-significant lags and are therefore disregarded according to standard practice[2].

[2]Some lags beyond $p_{\max}$ and $q_{\max}$ exceed the confidence bounds, but we ignore them since they appear after many insignificant lags.

We now aim to select an ARMA(p,q) model such that $p \leq 5$ and $q \leq 2$. The selection process follows these steps:

Significance of Coefficients We first assess the statistical significance of the estimated AR coefficients ϕ_i and MA coefficients θ_j. This involves testing the null hypothesis H_0: "the coefficient equals zero" against the alternative hypothesis H_1: "the coefficient is significantly different from zero."

If any coefficient in a given ARMA(p,q) model is not statistically significant (typically, if its p-value exceeds 5%), this may indicate model overfitting. In such cases, a more parsimonious model—excluding the non-significant terms—is generally preferred, both for interpretability and model performance.

After this step, the following candidate models remain: ARMA(0,1), ARMA(0,2), ARMA(1,0), ARMA(1,1), ARMA(2,0), ARMA(3,0), ARMA(4,0), and ARMA(5,0).

Ljung-Box Test The Ljung-Box test is used to assess whether the residuals from a model are free of autocorrelation. The null hypothesis H_0 states that "the residuals are not autocorrelated" (i.e., they are white noise), whereas the alternative hypothesis H_1 states that "the residuals are autocorrelated." We reject any model for which the p-value of the Ljung-Box test is below 5%, as this indicates statistically significant evidence of autocorrelation in the residuals, thereby violating the white noise assumption. In other words, we retain only those models for which the Ljung-Box test fails to reject the null hypothesis H_0: the residuals are independently distributed. At this stage, only the following models remain viable: ARMA(0,2), ARMA(1,1), ARMA(4,0), and ARMA(5,0). These models satisfy both the requirement of coefficient significance and the Ljung-Box diagnostic for uncorrelated residuals.

Model Selection via AIC and BIC Minimization To finalize our choice, we compare the remaining models using the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC), both of which reward goodness of fit while penalizing model complexity. The ARMA(1,1) model exhibits the lowest values for both AIC and BIC, indicating the best trade-off between accuracy and simplicity. We therefore select ARMA(1,1), with $p = 1$ and $q = 1$.

A comprehensive summary Table 3 in the Appendix reports the results of coefficient significance tests, Ljung-Box statistics, and the corresponding AIC and BIC values for each ARMA model fitted to the series Y_t.

2.2 Expression of X_t

As concluded in the previous section, the process X_t can be modeled as an ARIMA(1,1,1). Here, $\epsilon_t \sim \text{WN}(0, \sigma^2)$. The standard formulation of such a process is given by:

$$\phi(B)(1 - B)X_t = \theta(B)\epsilon_t, \tag{1}$$

where B denotes the backshift operator ($BX_t = X_{t-1}$). The autoregressive and moving average polynomials are defined as:

$$\phi(B) = 1 - 0.4764B,$$
$$\theta(B) = 1 + 0.8387B.$$

However, as highlighted in Section 1.1, several outliers are present in the series X_t and must be explicitly accounted for. Table 4 lists the identified outliers along with their type, estimated magnitude, and associated t-statistic.

A total of 10 outliers were detected using our R procedure, as shown in Figure 10. However, the last one corresponds to extreme values observed in 2020, which likely reflect the extraordinary shock caused by the COVID-19 pandemic. As these may not be representative of the usual dynamics of the series, we restrict our analysis to the first nine outliers. In particular, the last outlier does not appear significant when visually inspecting Figure 10.

Using the specification framework provided in Appendix E, we can now incorporate these shocks directly into the ARIMA(1,1,1) model. Starting from equation (1), and applying the appropriate structure for each outlier type as listed in Table 4, while adjusting the estimated coefficients ϕ_1 and θ_1, the resulting model becomes:

$$Y_t - 0.1781 \cdot Y_{t-1} = \epsilon_t - 0.7285 \cdot \epsilon_{t-1} - \sum_{i=1}^{7} \omega_i \cdot \mathbf{1}_{\{t=t_i\}} - \omega_8 \cdot \delta^{t-t_{\text{Feb.2016}}} \cdot \mathbf{1}_{\{t \geq t_{\text{Feb.2016}}\}} - \omega_9 \cdot \delta^{t-t_{\text{Mar.2020}}} \cdot \mathbf{1}_{\{t \geq t_{\text{Mar.2020}}\}}. \tag{2}$$

X_t is therefore :

$$\begin{aligned} X_t = {} & 1.1781 \cdot X_{t-1} - 0.1781 \cdot X_{t-2} + \epsilon_t - 0.7285 \cdot \epsilon_{t-1} \\ & - \sum_{i=1}^{7} \omega_i \cdot \mathbf{1}_{\{t=t_i\}} - \omega_8 \cdot \delta^{t-t_{\text{Feb.2016}}} \cdot \mathbf{1}_{\{t \geq t_{\text{Feb.2016}}\}} - \omega_9 \cdot \delta^{t-t_{\text{Mar.2020}}} \cdot \mathbf{1}_{\{t \geq t_{\text{Mar.2020}}\}}. \end{aligned}$$

3 Prediction

3.1 Forecasting X_{T+1} and X_{T+2}

We aim to forecast the values of X_{T+1} and X_{T+2}, where $T = 422$ corresponds to the final observed time point.

Note that regardless of the value chosen for the damping parameter δ associated with transient change (TC) outliers, we have $\delta^{t-t_0} \to 0$ as $t \to \infty$. Therefore, the effect of such outliers becomes negligible in the long run ($t_0 = 314$ or $t_0 = 363$).

Under this assumption, for $t \geq T$, the dynamics of the series Y_t effectively reduce to a standard ARMA(1,1) model, which can be written as:

$$Y_t - \phi_1 \cdot Y_{t-1} = \epsilon_t - \theta_1 \cdot \epsilon_{t-1} \text{ where the estimated coefficients are } \phi_1 = 0.1781, \theta_1 = -0.7285.$$

According to the theory (Chapter 2), for any forecast horizon $h > 0$, $\mathbb{E}[\epsilon_{T+h} \mid Y_T, Y_{T-1}, \ldots, Y_0] = 0$ as the residuals ϵ_{T+h} are orthogonal to the $Y_t, t \leq T$.

Using the equation of Y_t and recursively substituting Y_{T+1} into the expression for Y_{T+2}, we obtain the following system:

$$\begin{cases} Y_{T+1} = \phi_1 \cdot Y_T + \epsilon_{T+1} - \theta_1 \cdot \epsilon_T \\ Y_{T+2} = \phi_1^2 \cdot Y_T - \phi_1 \theta_1 \cdot \epsilon_T + (\phi_1 - \theta_1) \cdot \epsilon_{T+1} + \epsilon_{T+2} \end{cases}$$

From linearity of the conditional expectation and using the result above :

$$\begin{cases} \hat{Y}_{T+1|T} = \phi_1 \cdot \mathbb{E}[Y_T \mid Y_t,\, t \leq T] + \mathbb{E}[\epsilon_{T+1} \mid Y_t,\, t \leq T] - \theta_1 \cdot \mathbb{E}[\epsilon_T \mid Y_t,\, t \leq T] \\ \hat{Y}_{T+2|T} = \phi_1^2 \cdot \mathbb{E}[Y_T \mid Y_t,\, t \leq T] - \phi_1 \theta_1 \cdot \mathbb{E}[\epsilon_T \mid Y_t,\, t \leq T] + (\phi_1 - \theta_1) \cdot \mathbb{E}[\epsilon_{T+1} \mid Y_t,\, t \leq T] + \mathbb{E}[\epsilon_{T+2} \mid Y_t,\, t \leq T] \end{cases}$$

We obtain the following point forecasts based on the ARMA(1,1) model:

$$\begin{cases} \hat{Y}_{T+1|T} = \phi_1 \cdot Y_T - \theta_1 \cdot \epsilon_T, \\ \hat{Y}_{T+2|T} = \phi_1 \cdot (\phi_1 Y_T - \theta_1 \cdot \epsilon_T). \end{cases}$$

Our goal is now to derive the confidence region at level $1 - \alpha$ for the future values (X_{T+1}, X_{T+2}). We are particularly interested in the forecast error vector, denoted $X - \hat{X} \in \mathbb{R}^2$. By reintroducing the original series X_t using the relation $Y_t = X_t - X_{t-1}$, and recalling that:

$$\begin{cases} \hat{Y}_{T+1|T} = \hat{X}_{T+1|T} - X_T, \\ \hat{Y}_{T+2|T} = \hat{X}_{T+2|T} - \hat{X}_{T+1|T}, \end{cases}$$

We calculate : $\begin{cases} X_{T+1} - \hat{X}_{T+1|T} = Y_{T+1} - \hat{Y}_{T+1|T} = \epsilon_{T+1}, \\ X_{T+2} - \hat{X}_{T+2|T} = Y_{T+2} - \hat{Y}_{T+2|T} + Y_{T+1} - \hat{Y}_{T+1|T} = (1 + \phi_1 - \theta_1) \cdot \epsilon_{T+1} + \epsilon_{T+2}. \end{cases}$

We thus obtain the forecast error vector:

$$X - \hat{X} = \begin{pmatrix} X_{T+1} - \hat{X}_{T+1|T} \\ X_{T+2} - \hat{X}_{T+2|T} \end{pmatrix} = \begin{pmatrix} \epsilon_{T+1} \\ (1 + \phi_1 - \theta_1) \cdot \epsilon_{T+1} + \epsilon_{T+2} \end{pmatrix}. \tag{3}$$

Assuming the normality of the residuals : $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$, so $X - \hat{X} \sim \mathcal{N}(0, \Sigma)$ where Σ is the variance-covariance matrix of the forecast errors $\in \mathcal{M}_{\in}(\mathbb{R})$. We assume that Σ is invertible.

The computations leading to the expression of Σ are provided in appendix F. We ultimately obtain:

$$\Sigma = \begin{pmatrix} 1 & 1 + \phi_1 - \theta_1 \\ 1 + \phi_1 - \theta_1 & 1 + (1 + \phi_1 - \theta_1)^2 \end{pmatrix} \sigma^2. \tag{4}$$

We are then interested in the test statistic $(X - \hat{X})^T \Sigma^{-1} (X - \hat{X}) \sim \chi^2(2)$. For a risk level α, and thus a confidence level $1 - \alpha$: $\mathbb{P}\left((X - \hat{X})^T \Sigma^{-1} (X - \hat{X}) \leq \chi^2_{1-\alpha}(2)\right) = 1 - \alpha$ where $\chi^2_{1-\alpha}(2)$ is the quantile of order $1 - \alpha$ of the chi-squared distribution with 2 degrees of freedom.

The $(1-\alpha)$-level confidence region for the bivariate forecast vector X is given by (see [Banks and Fienberg, 2003]):

$$R_{1-\alpha}(X) = \left\{ x \in \mathbb{R}^2 \,\middle|\, (x - \hat{X})^\top \Sigma^{-1} (x - \hat{X}) \leq {\chi^2_{1-\alpha}}^{(2)} \right\}$$

More concretely, we can derive the individual 95% confidence intervals for the two forecasts as follows:

$$\begin{cases} IC_{95\%}(X_{T+1}) = \left[\hat{X}_{T+1|T} - 1.96 \cdot \hat{\sigma} \,,\, \hat{X}_{T+1|T} + 1.96 \cdot \hat{\sigma} \right] \\ IC_{95\%}(X_{T+2}) = \left[\hat{X}_{T+2|T} - 1.96 \cdot \hat{\sigma} \cdot \sqrt{1 + (1 + \hat{\phi}_1 - \hat{\theta}_1)^2} \,,\, \hat{X}_{T+2|T} + 1.96 \cdot \hat{\sigma} \cdot \sqrt{1 + (1 + \hat{\phi}_1 - \hat{\theta}_1)^2} \right] \end{cases} \tag{5}$$

3.2 Main hypotheses

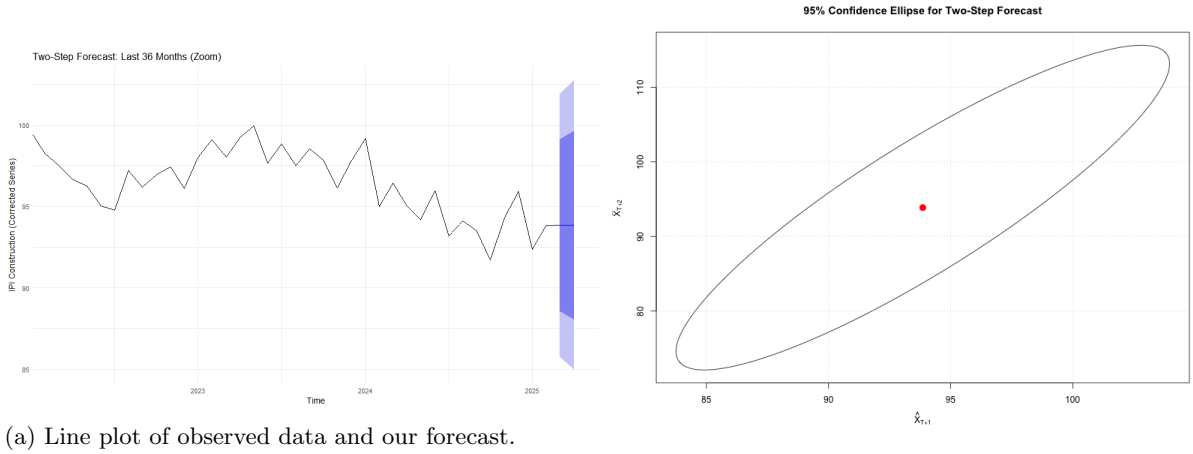
The main hypotheses used in estimating the confidence interval are:

- The consistency of all estimators, meaning our model is identifiable.
- Zero mean for future innovations (for any $h > 0$, $\mathbb{E}[\epsilon_{T+h} \mid Y_T, Y_{T-1}, \ldots, Y_0] = 0$).
- The residuals are independent and identically distributed across a Gaussian distribution : $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$. This also means the ϵ_t are strong white noise and homoskedastic.
- The variance-covariance matrix Σ is invertible and positive-definite ($\sigma^2_\epsilon > 0$).
- For any damping parameter δ associated with transient change (TC) outliers, we have $\delta^{t-t_0} \to 0$ as $t \to \infty$.

3.3 Graphic Representation

The predicted variables for March and April 2025 are 93.8607 and 93.8662 respectively. Their respective confidence intervals are $[85.7689; 101.9524]$ and $[84.9941; 102.7382]$. Figure 4 shows these predictions and confidence intervals in both a line plot (including observations over the last three years) and a joint confidence interval prediction ellipse.

The confidence intervals on our line plot seem quite large : our maximal and minimal values are, respectively, larger and smaller than any recent observations. The elongated shape of the confidence ellipse and its orientation show relatively strong positive correlation between our two predictions, which makes sense.



(a) Line plot of observed data and our forecast.
Note : 80% confidence interval - dark blue ; 95% - light blue.

(b) Prediction ellipse of our forecast

Figure 4: Forecasts and prediction intervals for March and April 2025

3.4 Improving the prediction of X_{T+1}

In this section, let Z_t a stationary time series available from t = 1 to T. We assume that Z_{T+1} is available faster than X_{T+1}. We will discuss how this information may allow us to improve the prediction X_{T+1}.

Saying Z_T is useful to predict X_T is equivalent to saying it causes X_T in the Granger sense [Granger, 1969]. Here, as we are specifically asking if Z_{T+1} improves the prediction of X_{T+1}, we are checking if there is instantaneous causality. This is true if and only if:

$$\hat{X}_{T+1|\{X_u, Z_u, u \leq T\} \cup \{Z_{T+1}\}} \neq \hat{X}_{T+1|\{X_u, Z_u, u \leq T\}}$$

This can also be represented in vector form.

Assume a bivariate vector autoregressive model of order p for the stationary time series (X_t, Z_t):

$$\begin{pmatrix} X_{T+1} \\ Z_{T+1} \end{pmatrix} = \sum_{i=1}^{p} A_i \begin{pmatrix} X_{T+1-i} \\ Z_{T+1-i} \end{pmatrix} + \begin{pmatrix} \epsilon_{1,T+1} \\ \epsilon_{2,T+1} \end{pmatrix}, \quad \begin{pmatrix} \epsilon_{1,T+1} \\ \epsilon_{2,T+1} \end{pmatrix} \sim \mathcal{N}(\mathbf{0}, \Sigma),$$

where $\Sigma = \mathbb{E}[\epsilon_{T+1}\epsilon'_{T+1}]$ is the variance-covariance matrix of the innovation vector ϵ_{T+1}. We assume Σ is positive definite.

We have instantaneous causality from Z_{T+1} to X_{T+1} if and only if the covariance $\sigma_{12} \neq 0$, where $\Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{pmatrix}$. For T+1, that means $\text{Cov}(\epsilon_{1,T+1}, \epsilon_{2,T+1}) \neq 0$.

To test this formally, one could use a **Wald Test** for the null hypothesis:

$$H_0 : \sigma_{12} = 0 \quad \text{(no instantaneous causality from } Z_t \text{ to } X_t\text{)}.$$

As we are testing a single parameter, the Wald statistic takes the form :

$$W = \frac{(\hat{\sigma_{12}} - \sigma_{12})^2}{Var(\hat{\sigma_{12}})}$$

Under the null hypothesis, this follows a χ^2_1 distribution. We then reject H_0 at level α if:

$$W > \chi^2_1(1 - \alpha),$$

where $\chi^2_1(1 - \alpha)$ is the $(1 - \alpha)$-quantile of χ^2_1.

A Study of the original series

When decomposing the series into its trend, seasonal, and random components, we used a multiplicative model. The trend confirms the patterns discussed in Section 1.1. Residuals appear stationary apart from the outliers of early 2020, which we will treat later.

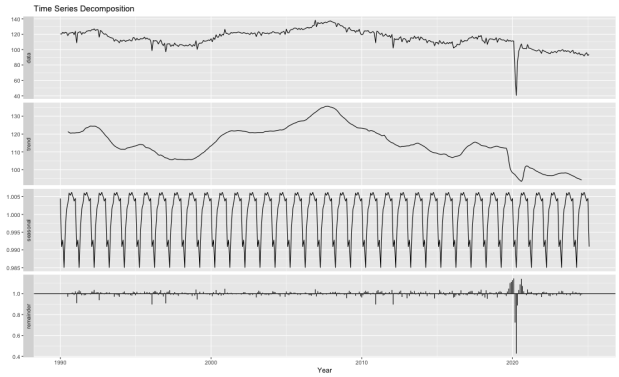


Figure 5: Decomposition of the original series into its trend / cycle / seasonality / residual components

We then study the ACF and PACF.

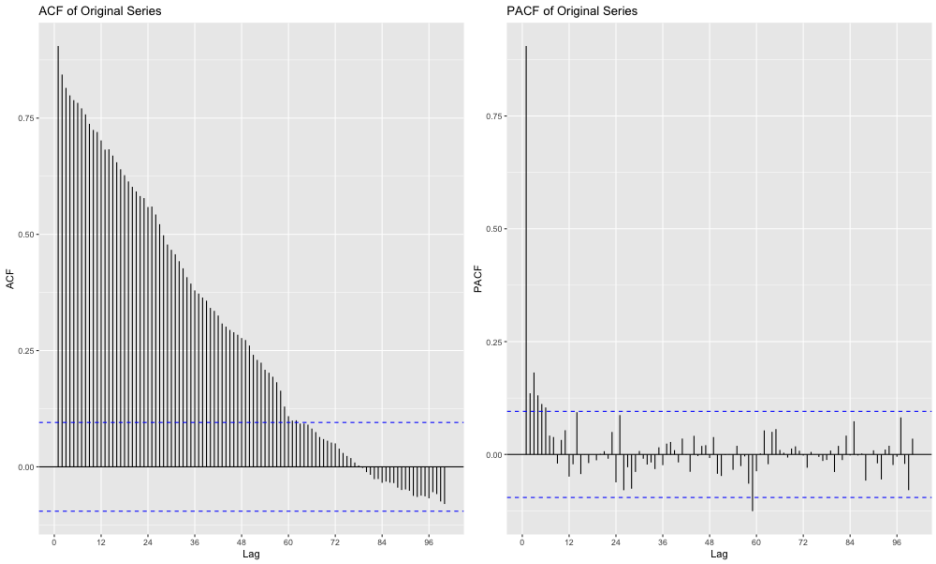


Figure 6: ACF (left) and PACF(right) of the Original Series

Note: The dashed blue line represents the 95% significance bounds.

The ACF decays slowly, and the PACF behaves like white noise : both of these suggest non-stationarity.

These hypotheses are confirmed by the three classic stationarity tests (see Appendix C for further detail).

B Seasonality

The original series is corrected from seasonal variations and working days (CVS-CJO) by INSEE.

The correction process seeks to break down the raw series into multiple components:

$$X_t = TC_t + S_t + WD_t + I_t$$

which correspond to the trend-cycle, the seasonal component, the calendar/working days effect, and the irregular component.

It then seeks to identify S_t and WD_t to correct the series of these variations. Today, the method used is JDemetra+, as recommended by Eurostat. It relies on an X13-ARIMA. The method itself is broken down into two principal modules[3]:

1. **RegArima** pre-adjusts the series by detecting additive outliers, transitory changes, level shifts, and seasonal outliers. It also corrects for working days thanks to regressors that reproduce the characteristics of the French calendar. Once identified, these effects are estimated by a Reg-Arima model to "linearise" the series and prolong it at the edges to facilitate seasonal adjustment.

2. **X11** does the actual seasonal adjustment through iterative smoothing using moving averages and decomposes the linearized series from the first part into orthogonal components: the trend-cycle, the seasonality, and the irregular component.

Coefficients are re-estimated each month, and model specifications each year.

Thus, our series should exhibit no seasonality. We verify this assumption graphically with the following plots.

As all the boxplots have similar profiles, no seasonality can be detected. As both the linear and polar plots of seasonality show similar behaviour for all months of the year (except the visible outlier of April 2020 due to Covid), we can once again conclude there is no seasonality.

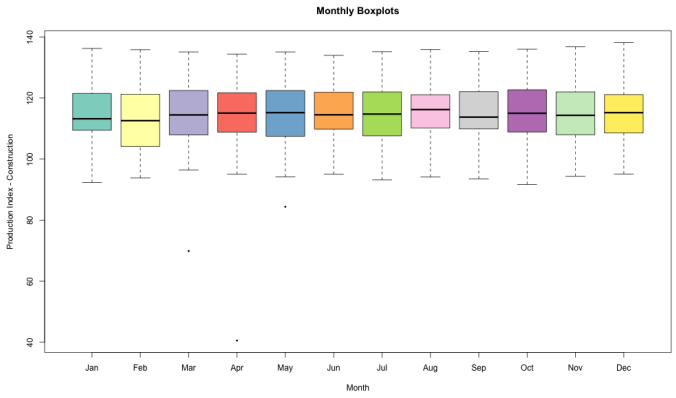


Figure 7: Monthly Boxplots of the Original Series

Note: Each colored box represents a different month. Both median (centre of box) and extreme values (ends of whiskers) are similar across each month.

[3]For more detail, see: https://www.insee.fr/fr/statistiques/fichier/4186908/imet133-g.pdf

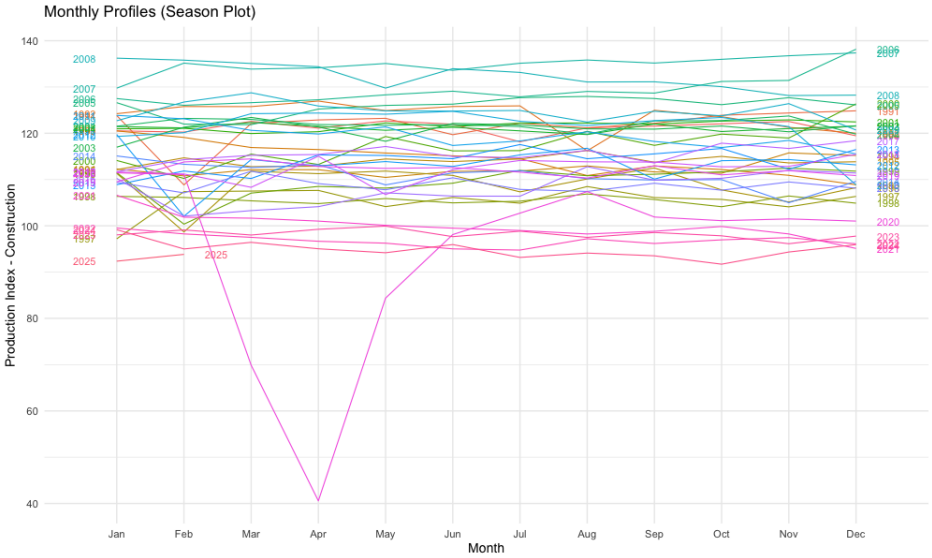


Figure 8: Seasonality of the Original Series

Note: The x-axis is months of the year, the y-axis the value of production.
Each line represents a year : except changes in level, behaviour each year is very similar.
No variations between months can be noted (except for the 2020 outlier).

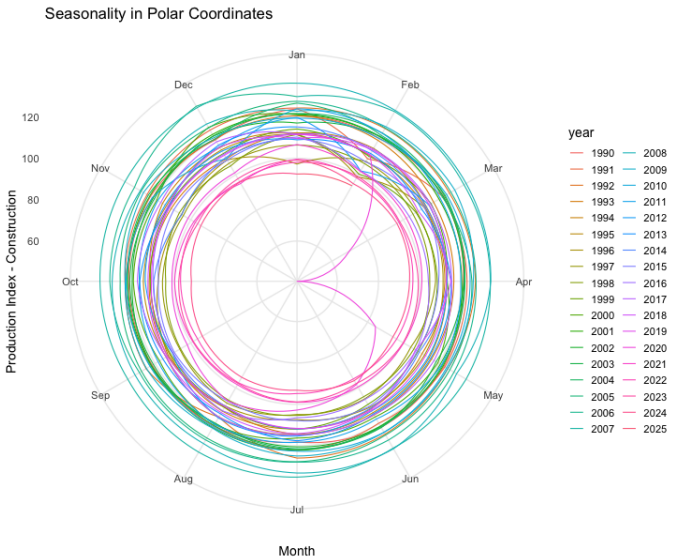


Figure 9: Seasonality of the Original Series in Polar Coordinates

Note: Each month represents a polar coordinate, and each line represents a year.
No variations between months can be noted (except for the 2020 outlier).

C Stationarity tests

Three classic tests were used to ensure stationarity:

- Augmented Dickey-Fuller (ADF) based on the following regression:

$$\Delta X_t = \alpha + \beta t + \gamma X_{t-1} + \sum_{i=1}^{p} \delta_i \Delta X_{t-i} + \varepsilon_t$$

  where the null hypothesis is $H_0 : \gamma = 0$ (non-stationarity).

- Phillips-Perron (PP), similar to the ADF but without lagged difference terms and using a semi-parametric model of the form:

$$\Delta X_t = \alpha + \beta t + \gamma X_{t-1} + \varepsilon_t$$

  Here too, the null hypothesis is $H_0 : \gamma = 0$ (non-stationarity).

- Kwiatkowski, Phillips, Schmidt and Shin (KPSS), which assumes the series can be decomposed into a deterministic trend, a random walk, and a stationary error:

$$y_t = r_t + \beta t + \varepsilon_t, \quad \text{where } r_t = r_{t-1} + u_t$$

  The null hypothesis is that the random walk component u_t has zero variance: $H_0 : \sigma_u^2 = 0$, implying stationarity.

The first two tests are unit root tests, meaning their null hypothesis is non-stationarity, whereas KPSS is a stationarity test, so its null hypothesis is stationarity.

Lag for the ADF test was chosen according to the method proposed by Ng and Perron (1995) : first, set an upper bound pmax for p, then estimate the ADF test regression with $p = p_{max}$. If the absolute value of the t-statistic for testing the significance of the last lagged difference is greater than 1.6, set $p = p_{max}$ and perform the unit root test. Otherwise, reduce the lag length by one and repeat the process. p_{max} is chosen according to Schwert's formula [Schwert, 1989]:

$$p_{max} = \lfloor 12 \cdot (\frac{T}{100})^{0.25} \rfloor$$

For X_t , $T = 422$, so $p_{max} = 17$.

For the original series, this process brings us to lag 8, as this is the highest amount of lags under 17 for which the test statistic is above the 1.6 threshold in absolute values.

For the differenced series, as the absolute value for the test with lag 17 is $5.42 > 1.6$, we can indeed conduct our ADF test with lag 17.

For our original series, two out of three tests (ADF and KPSS) indicate non-stationarity. Although the PP test suggests stationarity, the overall evidence leans toward non-stationarity of the series. Furthermore, Phillips-Perrron performs less well in finite samples than other unit root tests [Schwert, 1989].

Test Type	**Test Statistic**	**Lag Used**	**p-value**
ADF (Augmented Dickey-Fuller)	-1.387	8	0.7114
PP (Phillips-Perron)	-30.185 (Z-alpha)	5	< 0.01
KPSS	2.2324	5	0.01

Table 1: Unit root test results for the original series

We can easily conclude our differenced series is stationary, as all three tests concur:

Test Type	Test Statistic	Lag Used	p-value
ADF (Augmented Dickey-Fuller)	-5.42	17	< 0.01
PP (Phillips-Perron)	-386.85 (Z-alpha)	5	< 0.01
KPSS	0.049	5	> 0.1

Table 2: Unit root test results for the differenced series

D Choose p and q

This table summarises the steps involved in our selection of our ARMA(p,q) model:

- The outputs linked to our initial test of the significance of the coefficients are contained in the columns "Std Error", "t-value", "p-value S" : we reject the null hypothesis of $\beta = 0$ if our p-value is below 0.05. We are running a classic Student's test.
- The next two columns ,"Ljung-Box Stat" and "p-value L" contian the outptus linked to our Ljung-Box tests on the residuals: we reject the null hypothesis of residual autocorrelation if the p-value is below 0.05.
- The two final columns, "AIC" and "BIC", contain the AIC and BIC used to select an optimal model : we want to minimize both.

Table 3: Overall Summary of the selection of p and q

Model	Coeff	Value	Std Error	t-value	p-value S	Ljung-Box stat	p-value L	AIC	BIC
ARMA(0,1)	ma1	-0.4008	0.0739	-5.4260	0.0000	40.0360	0.0000	2466.548	2478.676
ARMA(0,2)	ma1	-0.3565	0.0452	-7.8945	0.0000	6.8582	0.7388	2433.739	2449.910
	ma2	-0.2863	0.0448	-6.3896	0.0000				
ARMA(1,0)	ar1	-0.1883	0.0478	-3.9387	0.0001	36.0879	0.0001	2481.413	2493.541
ARMA(1,1)	ar1	0.4764	0.0649	7.3378	0.0000	5.5339	0.8528	2432.691	2448.861
	ma1	-0.8387	0.0380	-22.0435	0.0000				
ARMA(1,2)	ar1	0.2917	0.1386	2.1049	0.0359	2.8734	0.9842	2431.968	2452.181
	ma1	-0.6218	0.1385	-4.4895	0.0000				
	ma2	-0.1515	0.0879	-1.7236	0.0855				
ARMA(2,0)	ar1	-0.2293	0.0475	-4.8249	0.0000	27.6341	0.0021	2462.963	2479.134
	ar2	-0.2174	0.0475	-4.5775	0.0000				
ARMA(2,1)	ar1	0.4667	0.0697	6.6960	0.0000	2.6036	0.9893	2431.644	2451.857
	ar2	-0.0987	0.0559	-1.7655	0.0782				
	ma1	-0.7945	0.0530	-14.9906	0.0000				
ARMA(2,2)	ar1	0.8687	0.4284	2.0277	0.0432	2.3787	0.9925	2433.201	2457.457
	ar2	-0.2963	0.2003	-1.4795	0.1398				
	ma1	-1.2002	0.4363	-2.7508	0.0062				
	ma2	0.3462	0.3650	0.9486	0.3434				
ARMA(3,0)	ar1	-0.2658	0.0480	-5.5351	0.0000	20.5814	0.0242	2453.081	2473.294
	ar2	-0.2556	0.0481	-5.3139	0.0000				
	ar3	-0.1664	0.0479	-3.4718	0.0006				
ARMA(3,1)	ar1	0.4462	0.0844	5.2838	0.0000	2.4840	0.9911	2433.424	2457.680
	ar2	-0.0954	0.0565	-1.6870	0.0923				
	ar3	-0.0281	0.0598	-0.4706	0.6381				
	ma1	-0.7759	0.0700	-11.0902	0.0000				

Model	Coeff	Value	Std Error	t-value	p-value S	Ljung-Box stat	p-value L	AIC	BIC
ARMA(3,2)*	ar1	-0.0544	NaN	NaN	NaN	2.5800	0.9897	2435.576	2463.874
	ar2	0.1423	NaN	NaN	NaN				
	ar3	-0.0636	NaN	NaN	NaN				
	ma1	-0.2741	NaN	NaN	NaN				
	ma2	-0.4070	NaN	NaN	NaN				
ARMA(4,0)	ar1	-0.2912	0.0482	-6.0468	0.0000	13.0093	0.2232	2445.394	2469.650
	ar2	-0.2943	0.0491	-5.9875	0.0000				
	ar3	-0.2064	0.0491	-4.2066	0.0000				
	ar4	-0.1504	0.0480	-3.1307	0.0019				
ARMA(4,1)	ar1	0.4053	0.1132	3.5804	0.0004	2.0884	0.9956	2435.002	2463.301
	ar2	-0.1106	0.0615	-1.7978	0.0729				
	ar3	-0.0299	0.0611	-0.4888	0.6252				
	ar4	-0.0409	0.0628	-0.6514	0.5152				
	ma1	-0.7356	0.1037	-7.0936	0.0000				
ARMA(4,2)	ar1	0.5478	0.8873	0.6173	0.5373	2.0743	0.9957	2436.974	2469.315
	ar2	-0.1767	0.4031	-0.4384	0.6613				
	ar3	-0.0167	0.1022	-0.1636	0.8701				
	ar4	-0.0386	0.0686	-0.5623	0.5742				
	ma1	-0.8782	0.8872	-0.9899	0.3228				
	ma2	0.1132	0.6895	0.1642	0.8697				
ARMA(5,0)	ar1	-0.3117	0.0483	-6.4566	0.0000	5.7229	0.8380	2439.674	2467.972
	ar2	-0.3224	0.0497	-6.4833	0.0000				
	ar3	-0.2462	0.0507	-4.8594	0.0000				
	ar4	-0.1894	0.0496	-3.8198	0.0002				
	ar5	-0.1342	0.0481	-2.7919	0.0055				
ARMA(5,1)	ar1	0.3736	0.1728	2.1621	0.0312	2.0252	0.9961	2436.921	2469.262
	ar2	-0.1209	0.0740	-1.6342	0.1030				
	ar3	-0.0415	0.0763	-0.5447	0.5863				
	ar4	-0.0449	0.0662	-0.6773	0.4986				
	ar5	-0.0197	0.0707	-0.2783	0.7810				
	ma1	-0.7042	0.1666	-4.2258	0.0000				
ARMA(5,2)	ar1	-0.5734	0.1132	-5.0658	0.0000	2.7268	0.9871	2438.360	2474.744
	ar2	0.2976	0.1420	2.0951	0.0368				
	ar3	-0.1338	0.0847	-1.5799	0.1149				
	ar4	-0.0689	0.0892	-0.7729	0.4400				
	ar5	-0.0300	0.0634	-0.4735	0.6361				
	ma1	0.2448	0.1032	2.3719	0.0182				
	ma2	-0.7329	0.1008	-7.2726	0.0000				

Note: *The NaNs returned by R for this model mean that the parameters are near the edge of the stationarity region: the standard errors cannot be computed because the sum of the coefficients is close to 1.

E Outlier Types and Their Treatment in Time Series Model

Outliers are observations that deviate significantly from the expected behavior of a time series. In time series analysis, especially when using ARIMA models, uncorrected outliers can bias parameter estimation and impair forecasting accuracy. It is therefore important to detect them and model their impact appropriately. Though INSEE's CVS-CJO treatment methodology should address most of the issues related to outliers (see Appendix C), we still found values that needed correction.

Let X_t be the observed time series, ω the magnitude of the outlier, and Z_t the adjusted series after removing the outlier's effect at time $t = t_{\text{outlier}}$. Four common types of outliers are typically distinguished in time series analysis, each associated with a different dynamic impact on the series.

1. **Additive Outlier (AO)**: A sudden, isolated disturbance that affects only a single time point without altering the underlying dynamics. These are typically associated with recording errors or temporary shocks. The corrected series is:
$$Z_t = X_t + \omega \cdot \mathbf{1}_{\{t=t_{outlier}\}}.$$

2. **Innovative Outlier (IO)**: A shock that enters through the innovation term and propagates according to the model dynamics. Unlike AO, the effect of an IO is filtered by the ARIMA process itself. The correction must take into account the full ARIMA structure:
$$Z_t = X_t + \psi(B)\omega \cdot \mathbf{1}_{\{t=t_{outlier}\}},$$
where $\psi(B) = \frac{\theta(B)}{\phi(B)}$ is the impulse response function of the ARIMA model.

3. **Level Shift (LS)**: A permanent change in the mean level of the series starting at time $t = S$. This could represent a structural break or regime change. The effect persists indefinitely:
$$Z_t = X_t + \omega \cdot \sum_{k=0}^{\infty} \mathbf{1}_{\{t=t_{outlier}+k\}} = X_t + \omega \cdot \mathbf{1}_{\{t \geq t_{outlier}\}}.$$

4. **Transient Change (TC)**: A temporary deviation that decays gradually over time. This type of outlier reflects a temporary disturbance that fades exponentially. The standard correction is modeled by a geometric decay:
$$Z_t = X_t + \omega \cdot \delta^{t-t_{outlier}} \cdot \mathbf{1}_{\{t \geq t_{outlier}\}},$$
where $0 < \delta < 1$ is a damping parameter governing the speed of decay.

Table 4: Detected Outliers in the Series X_t (the last one not taken in the adjusted series)

Observation	**Time**	**Type**	**Estimate** ($\hat{\beta}$)	t**-statistic**
14	February 1991	AO	-13.1768	-7.39
32	August 1992	AO	-8.3201	-4.66
74	February 1996	AO	-13.5640	-7.61
85	January 1997	AO	-10.0067	-5.61
110	February 1999	AO	-7.3368	-4.09
252	December 2010	AO	-13.7539	-7.72
266	February 2012	AO	-13.5115	-7.54
314	February 2016	TC	-7.9013	-4.36
363	March 2020	TC	-41.1391	-21.10
364	April 2020	AO	-41.0581	-21.08

Coefficients of the Outliers ω_i The estimated outlier effects ω_i incorporated in equation (2) are provided below:

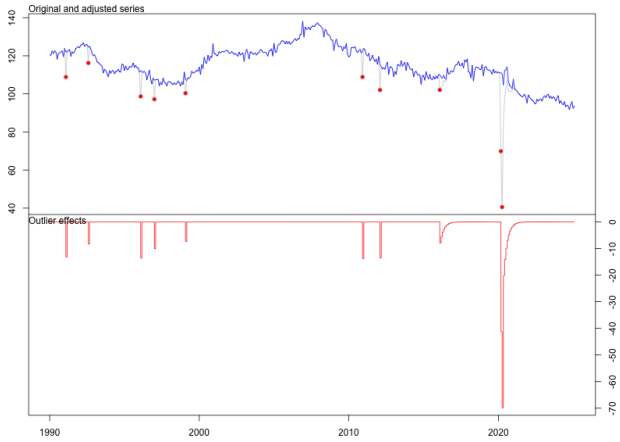


Figure 10: Outliers

Note: The downward red bars indicate substantial negative shocks to the series, which are often associated with exogenous events such as unexpected macroeconomic disturbances.

$\omega_1 = -13.1768$	(February 1991, AO)
$\omega_2 = -8.3201$	(August 1992, AO)
$\omega_3 = -13.5640$	(February 1996, AO)
$\omega_4 = -10.0067$	(January 1997, AO)
$\omega_5 = -7.3368$	(February 1999, AO)
$\omega_6 = -13.7539$	(December 2010, AO)
$\omega_7 = -13.5115$	(February 2012, AO)
$\omega_8 = -7.9013$	(February 2016, TC)
$\omega_9 = -41.1391$	(March 2020, TC)
$\delta \in]0, 1[$	(Damping parameter for TC effects)

F Calculus

$$\text{Var}(X_{T+1} - \hat{X}_{T+1|T}) = \text{Var}(\epsilon_{T+1}) = \sigma^2, \quad \text{(by assumption)}$$

$$\begin{aligned}\text{Var}(X_{T+2} - \hat{X}_{T+2|T}) &= \text{Var}\left((1 + \phi_1 - \theta_1) \cdot \epsilon_{T+1} + \epsilon_{T+2}\right) \\ &= (1 + \phi_1 - \theta_1)^2 \cdot \text{Var}(\epsilon_{T+1}) + \text{Var}(\epsilon_{T+2}) \quad \text{(since the residuals are independent)} \\ &= \left(1 + (1 + \phi_1 - \theta_1)^2\right) \cdot \sigma^2,\end{aligned}$$

$$\begin{aligned}\text{Cov}(X_{T+1} - \hat{X}_{T+1|T},\, X_{T+2} - \hat{X}_{T+2|T}) &= \text{Cov}\left((1 + \phi_1 - \theta_1) \cdot \epsilon_{T+1} + \epsilon_{T+2},\, \epsilon_{T+1}\right) \\ &= (1 + \phi_1 - \theta_1) \cdot \text{Var}(\epsilon_{T+1}) + \text{Cov}(\epsilon_{T+2},\, \epsilon_{T+1}) \\ &= (1 + \phi_1 - \theta_1) \cdot \sigma^2 \quad \text{(by bilinearity and independence of the residuals).}\end{aligned}$$

G Code

```r
#########################################################
################## Linear Time Series ###################
###### BENABDESADOK Nayel, CIANFARANI Ana-Sofia #######
#########################################################

####### - Part 0 : Installations & Data Download - ######

# Load required packages
library(fUnitRoots)
library(zoo)
library(tseries)
library(forecast)
library(ggplot2)
library(scales)
library(RColorBrewer)
library(gridExtra)
library(astsa)
library(tsoutliers)
library(ellipse)

# # Load the data on personal computer (Sofia)
# path <- "C:/Users/cianf/OneDrive/Documents/STL"
# path <- "C:/Users/cianf/OneDrive/Documents/STL/data.csv"
# # Load the data on personnal computer (Nayel)
path_donnees <- "/Users/nayelbenabdesadok/GitProjects/Time_Series_ENSAE/reprise/data.csv"
path <- "/Users/nayelbenabdesadok/GitProjects/Time_Series_ENSAE/reprise"
setwd(path)
getwd()


# Load the data
data <- read.csv(file = path_donnees, sep = ";")

# Clean the table
data_all <- data[4:425, ]
data_all$Codes <- NULL
colnames(data_all) <- c("Date_MY", "IPI")


# Converting date to the correct format
data_all$Date <- as.Date(paste(data_all$Date_MY, 1, sep = "-"), format = "%Y-%m-%d")
data_all$Date_MY <- NULL

# Extracting month and year from date
Year <- as.numeric(format(data_all$Date, format = "%Y"))
Month <- format(data_all$Date, format = "%m")
data_all <- cbind(data_all, Year, Month)
data_all$Year <- as.numeric(data_all$Year)
data_all$IPI <- as.numeric(data_all$IPI)
sort(data_all$IPI)
data_all <- data_all[order(data_all$Date), ]
rownames(data_all) <- seq(length = nrow(data_all))


# Create a time series for easier analysis, and plot
Xt.ts <- ts(data_all$IPI, start = c(1990, 1), end = c(2025, 2), frequency = 12)
par(mfrow = c(1, 1))
plot.ts(Xt.ts, xlab = "Years", ylab = "IPI Construction (CVS-CJO, base 2021)")

png("Xt.png", width = 1000, height = 600)
plot.ts(Xt.ts,
        main = "Main Series",
        ylab = "IPI Construction (CVS-CJO, base 2021)",
        xlab = "Time",
        lwd = 2)
```

```r
dev.off()
#============================================================================#
# The dates range from January 1990 (1990m1) to February 2025 (2025m2) as shown above.
# The series represents the Index of Industrial Production (IPI) in the construction sector,
# seasonally and working-day adjusted (CVS-CJO), with a base of 100 in 2021.
# At first glance, the series appears non-stationary, with multiple structural breaks.
# A sharp drop is clearly visible around April 2020, likely linked to the COVID-19 crisis.
# Before applying any transformations, we will examine the overall trend and variability.
#============================================================================#



####### - Part I : The Data - ######

####### - Part 1.1 : Trend/Cycle/Seasonal Decomposition (What does the chosen series
    represent ?)- ######
# Decomposition of the time series to analyze the trend component.
decompose <- decompose(Xt.ts,type="multiplicative")
png("decompose.png", width = 1000, height = 600)
autoplot(decompose, main = "Time Series Decomposition", xlab ="Year")
dev.off()


#============================================================================#
# Trend Analysis: the series displays three distinct phases
# a decline from 1990 to 2000, a growth period from 2000 to 2008,
# and a significant downward trend from 2008 to 2025.
# A sharp outlier is observed in early 2020, likely due to the COVID-19 crisis.
#============================================================================#

#============================#
#  Is there Seasonality ?   #
#============================#
n_years <- length(unique(floor(time(Xt.ts))))
colors <- scales::hue_pal()(n_years)

# Standard season plot (monthly profile)
png("season_plot.png", width = 1000, height = 600)
ggseasonplot(Xt.ts,
             year.labels = TRUE,
             year.labels.left = TRUE) +
  ylab("Production Index - Construction") +
  xlab("Month") +
  ggtitle("Monthly Profiles (Season Plot)") +
  scale_color_manual(values = colors) +
  theme_minimal(base_size = 14)
dev.off()

# Polar season plot
png("season_polar_plot.png", width = 1000, height = 600)
ggseasonplot(Xt.ts,
             polar = TRUE) +
  ylab("Production Index - Construction") +
  xlab("Month") +
  ggtitle("Seasonality in Polar Coordinates") +
  scale_color_manual(values = colors) +
  theme_minimal(base_size = 14)
dev.off()

# Monthly boxplots
png("monthly_boxplots.png", width = 1000, height = 600)
cols <- RColorBrewer::brewer.pal(12, "Set3")
boxplot(Xt.ts ~ cycle(Xt.ts),
        col = cols,
        pch = 20,
        cex = 0.5,
        main = "Monthly Boxplots",
        ylab = "Production Index - Construction",
```

```
        xlab = "Month",
        names = month.abb)
dev.off()
#=============================================================================#
# Regardless of the plot displayed, we observe no influence of the month
# on the behavior of the series.
# Absence of seasonality.
#=============================================================================#



####### - Part 1.2 : Stationary ######
#===========================#
#  Is Xt.ts Stationary ?    #
#===========================#
# ACF
acf_plot_orig <- ggAcf(Xt.ts, lag.max = 100, plot = TRUE) +
  ggtitle("ACF of Original Series")
#PACF
pacf_plot_orig <- ggPacf(Xt.ts, lag.max = 100, plot = TRUE) +
  ggtitle("PACF of Original Series")

png("ACF_PACF_Xt.png", width = 1000, height = 600)
gridExtra::grid.arrange(acf_plot_orig, pacf_plot_orig, ncol = 2)
dev.off()

#=============================================================================#
# ACF decays slowly + PACF behaves like white noise.
# Suggests non-stationarity     Let's run some tests: ADF, PP, KPSS.
#=============================================================================#

# Center the series
Xt_centered <- Xt.ts - mean(Xt.ts)
ggAcf(Xt_centered,lag.max=100, plot=T)

#=============================================================================#
# We decided to center the series but this is optional :
# the series X_t and X_t_centered are the same in our analysis !
#=============================================================================#


#=============================================================================#
# To determine the lag to use for our ADF test, we refer to Schwert (1989).
# The code below implements their lag selection algorithm.
# For more details see our report, Appendix : Stationarity tests.
# We set type = ct as there is a trend.
#=============================================================================#

# Initialize the lag value at pmax = 17
lag <- 17

# Store the ADF test result
adf_result <- NULL

# Loop until the t-statistic is above 1.6 in absolute value
while (lag > 0) {
  # Perform the ADF test
  adf_result <- adfTest(Xt_centered, lags = lag, type = "ct")

  # Check the t-statistic
  if (abs(adf_result@test$statistic) > 1.6) {
    break
  }

  # Decrease the lag value
  lag <- lag - 1
}
cat("Final lag value:", lag, "\n")
```

```r
print(adf_result)

# Phillips-Perron Test (PP)
library(tseries)
cat("\nPhillips-Perron Test Results:\n")
print(pp.test(Xt_centered))

# KPSS Test
cat("\nKPSS Test Results:\n")
kpss_result <- kpss.test(Xt_centered)
print(kpss_result)

#=============================================================================#
# Stationarity Tests Summary:
# - ADF Test (lag = 8, type = "ct")     p-value = 0.7114     FAIL to reject H0
# - PP Test     p-value < 0.01     REJECT H0 (suggests stationarity but less reliable for
    finite samples)
# - KPSS Test     p-value = 0.01     REJECT H0 of stationarity
#
# Conclusion: Two out of three tests (ADF and KPSS) indicate non-stationarity.
# Although the PP test suggests stationarity, the overall evidence leans toward
# non-stationarity of the centered series, likely due to trend or structural changes.
#=============================================================================#

#=============================================================================#
# Stationarity Tests on the Differenced Centered Series
#=============================================================================#

diff_Xt_centered <- diff(Xt_centered)

png("diff_Xt_centered.png", width = 1000, height = 600)
plot.ts(diff_Xt_centered,
        main = "Differenced Centered Series",
        ylab = "Differenced IPI",
        xlab = "Time",
        lwd = 2)
dev.off()

#=============================================================================#
# This time, our ADF test works with the max lag immediately.
# We set type = nc as there is the series is centered and stationary.
#=============================================================================#

# Augmented Dickey-Fuller Test (ADF)
adf_result_diff <- adfTest(diff_Xt_centered, lags = 17, type = "nc")
cat("ADF Test Results (Differenced Series):\n")
print(adf_result_diff)

# Phillips-Perron Test (PP)
cat("\nPhillips-Perron Test Results (Differenced Series):\n")
print(pp.test(diff_Xt_centered))

# KPSS Test
cat("\nKPSS Test Results (Differenced Series):\n")
kpss_result_diff <- kpss.test(diff_Xt_centered)
print(kpss_result_diff)

#=============================================================================#
## Stationarity Tests Summary:
# - ADF Test (lag = 17, type = "nc")     p-value = 0.01     REJECT H0 - Stationarity
# - PP Test     p-value < 0.01     REJECT H0 - Stationarity
# - KPSS Test     p-value > 0.1     DO NOT REJECT H0 - Stationarity
#
# All three stationarity tests (ADF, PP, KPSS) applied to the differenced
# centered series confirm stationarity (p-values all in favor).
# Conclusion: The original series is integrated of order 1 (I(1)).
#=============================================================================#
####### - Part 1.3 : Before/after Stationarity ######
```

```r

png("Xt_vs_diffXt.png", width = 1000, height = 300)
par(mfrow = c(1, 2))

plot.ts(Xt.ts,
        main = expression("Original Series " ~ X[t]),
        ylab = "IPI Construction",
        xlab = "Time",
        lwd = 2,
        col = "steelblue")

plot.ts(diff(Xt.ts),
        main = expression("Differenced Series " ~ Y[t] == Delta * X[t]),
        ylab = "Differenced IPI",
        xlab = "Time",
        lwd = 2,
        col = "darkorange")

dev.off()
####### - Part II : ARMA Models ######

####### - Part 2.1 : Pick p and q ######


acf_diff_plot <- ggAcf(diff_Xt_centered, lag.max = 40, plot = TRUE) +
  ggtitle("ACF of Differenced Series")
pacf_diff_plot <- ggPacf(diff_Xt_centered, lag.max = 40, plot = TRUE) +
  ggtitle("PACF of Differenced Series")

png("ACF_PACF_diff_Xt_centered.png", width = 1000, height = 300)
gridExtra::grid.arrange(acf_diff_plot, pacf_diff_plot, ncol = 2)
dev.off()

#=============================================================================#
# Property of MA(2) models: if the autocorrelation function (ACF) becomes
# zero for all lags h > 2, the process can be identified as MA(2).
# Similarly, for AR(5) models: the partial autocorrelation function (PACF)
# cuts off after lag 5.
#
# p_max=5 and q_max=2. Let's test all ARMA(p,q) such that p<=pmax and q<qmax
#=============================================================================#

# we call the differents models arma_p_q
#=============================================================================#
# We test all ARIMA(p,1,q) models for the construction series Xt.ts,
# where p     5 and q     2. We compute AIC and BIC for each combination.
#=============================================================================#
arma_0_1 <- sarima(Xt.ts, 0, 1, 1)
arma_0_2 <- sarima(Xt.ts, 0, 1, 2)

arma_1_0 <- sarima(Xt.ts, 1, 1, 0)
arma_1_1 <- sarima(Xt.ts, 1, 1, 1)
arma_1_2 <- sarima(Xt.ts, 1, 1, 2)

arma_2_0 <- sarima(Xt.ts, 2, 1, 0)
arma_2_1 <- sarima(Xt.ts, 2, 1, 1)
arma_2_2 <- sarima(Xt.ts, 2, 1, 2)

arma_3_0 <- sarima(Xt.ts, 3, 1, 0)
arma_3_1 <- sarima(Xt.ts, 3, 1, 1)
arma_3_2 <- sarima(Xt.ts, 3, 1, 2)

arma_4_0 <- sarima(Xt.ts, 4, 1, 0)
arma_4_1 <- sarima(Xt.ts, 4, 1, 1)
arma_4_2 <- sarima(Xt.ts, 4, 1, 2)

arma_5_0 <- sarima(Xt.ts, 5, 1, 0)
arma_5_1 <- sarima(Xt.ts, 5, 1, 1)
```

```r
arma_5_2 <- sarima(Xt.ts, 5, 1, 2)


#=============================================================================#
# To eliminate unsuitable models, we first exclude models with at least
# one non-significant coefficient (p-value > 5%), based on hypothesis testing (H0 vs H1).
#=============================================================================#


arma_0_1$ttable
arma_0_2$ttable

arma_1_0$ttable
arma_1_1$ttable
arma_1_2$ttable # out

arma_2_0$ttable
arma_2_1$ttable # out
arma_2_2$ttable # out

arma_3_0$ttable
arma_3_1$ttable # out
arma_3_2$ttable # out

arma_4_0$ttable
arma_4_1$ttable # out
arma_4_2$ttable # out

arma_5_0$ttable
arma_5_1$ttable # out
arma_5_2$ttable # out

#=============================================================================#
# We still keep the excluded models ("out models") to compute the Ljung-Box
# statistics on their residuals.
# We will later exclude models whose Ljung-Box p-value is less than 5%.
#=============================================================================#
models <- c("arma_0_1", "arma_0_2",
            "arma_1_0", "arma_1_1", "arma_1_2",
            "arma_2_0", "arma_2_1", "arma_2_2",
            "arma_3_0", "arma_3_1", "arma_3_2",
            "arma_4_0", "arma_4_1", "arma_4_2",
            "arma_5_0", "arma_5_1", "arma_5_2")

for (m in models) {
  cat("\n--- Ljung-Box Test for", m, "---\n")
  print(Box.test(residuals(get(m)$fit), lag = 10, type = "Ljung-Box"))
}
# The remaining models are ARMA(1,1), ARMA(0,2), and ARMA(5,0).
# To select the final model, we choose the one that minimizes both the AIC and BIC criteria.


info_criteria <- data.frame(
  Model = character(),
  AIC = numeric(),
  BIC = numeric(),
  stringsAsFactors = FALSE
)

for (m in models) {
  fit <- get(m)$fit
  model_aic <- AIC(fit)
  model_bic <- BIC(fit)

  info_criteria <- rbind(info_criteria, data.frame(
    Model = m,
    AIC = model_aic,
    BIC = model_bic
```

```
  ))
}
print(info_criteria)

# ARMA (1,1) is choosen !

####### - Part 2.2 : Taking outliers into account  #######

# Detect the outliers
arima111 <- arima(Xt.ts, order = c(1, 1, 1))
outlier_result <- tso(Xt.ts, types = c("AO", "LS", "TC", "IO"),
                      tsmethod = "arima", args.tsmethod = list(order = c(1, 1, 1)))
all_outliers <- outlier_result$outliers
print(all_outliers)

#=============================================================================#
# A total of 10 outliers are detected: 8 additive outliers (AO) and 2 transient changes (TC).
# The last outlier, however, does not correspond to any visually identifiable shock in the series.
# Therefore, we retain only the first 9 outliers for the adjustment.
# We then correct the series accordingly using the corresponding estimated effects.
#=============================================================================#

selected_indices <- head(all_outliers$ind, 9)
adjusted_effects <- outlier_result$effects
keep_positions <- rep(FALSE, length(adjusted_effects))
keep_positions[selected_indices] <- TRUE
adjusted_effects[!keep_positions] <- 0

yadj_9out <- Xt.ts - adjusted_effects
# Adjusted coefficients : phi1 = 0.1781 , theta1 = -0.7285


final_model_9out <- arima(yadj_9out, order = c(1, 1, 1))
summary(final_model_9out)
# AIC = 2395.14 < 2432.691  (AIC of the initial model) --> improvement confirmed

png("outliers_plot_full.png", width = 800, height = 600)
plot(outlier_result)
dev.off()



####### - Part III     Forecasting ######

####### - Part 3.1 : X_T+1 and X_T+2  #######
forecast_2 <- forecast(final_model_9out, h = 2)
print(forecast_2)

#=============================================================================#
# Point Forecast     80% CI                  95% CI
#-----------------------------------------------------------------------------#
# Mar 2025:  93.86069   [88.56977 ; 99.15160]   [85.76894 ; 101.95240]
# Apr 2025:  93.86615   [88.06500 ; 99.66730]   [84.99406 ; 102.73820]
#=============================================================================#


#===========================#
# Zoom Forecast Plot (Last 36 Months)
#===========================#

start_zoom <- time(yadj_9out)[length(yadj_9out) - 35]
end_zoom <- time(forecast_2$mean)[2]

recent_values <- window(yadj_9out, start = start_zoom)
forecast_values <- forecast_2$mean
forecast_lower <- forecast_2$lower[,2]  # 95% CI
```

```r
forecast_upper <- forecast_2$upper[,2]  # 95% CI

y_min <- min(recent_values, forecast_lower)
y_max <- max(recent_values, forecast_upper)

# Get the last observed point and first forecast point
# Then create a line between them, as autoplot does not do this automatically.
last_obs_time <- time(tail(recent_values, 1))
last_obs_value <- tail(recent_values, 1)
first_fc_time <- time(forecast_values)[1]
first_fc_value <- forecast_values[1]
connect_df <- data.frame(
  Time = c(last_obs_time, first_fc_time),
  Value = c(last_obs_value, first_fc_value)
)

# Plot, including the  connecting line
png("forecast_zoom_adjusted.png", width = 1000, height = 600)
autoplot(forecast_2, series = "Forecast") +
  autolayer(recent_values, series = "Observed", color = "black") +
  geom_line(data = connect_df, aes(x = Time, y = Value), linetype = "solid", color = "black"
    ) +
  ggtitle("Two-Step Forecast: Last 36 Months (Zoom)") +
  ylab("IPI Construction (Corrected Series)") +
  xlab("Time") +
  coord_cartesian(xlim = c(start_zoom, end_zoom), ylim = c(y_min, y_max)) +
  scale_color_manual(name = "Series",
                     values = c("Observed" = "black", "Forecast" = "blue")) +
  theme_minimal(base_size = 14)
dev.off()

####### - Part 3.2 : Confidence Region  #######

phi <- coef(final_model_9out)["ar1"]
theta <- coef(final_model_9out)["ma1"]
sigma2 <- final_model_9out$sigma2

sigma_g1 <- sqrt(sigma2)
sigma_g2 <- sqrt(sigma2 * (1 + (1 + phi - theta)^2))
rho <- sigma2 * (1 + phi - theta)

Sigma <- matrix(c(sigma_g1^2, rho,
                  rho, sigma_g2^2), nrow = 2)

centre <- c(forecast_2$mean[1], forecast_2$mean[2])

ell <- ellipse(Sigma, centre = centre, level = 0.95, npoints = 1000)

png("confidence_ellipse_forecast.png", width = 800, height = 600)
plot(ell,
     type = 'l',
     xlab = expression(hat(X)[T+1]),
     ylab = expression(hat(X)[T+2]),
     main = "95% Confidence Ellipse for Two-Step Forecast")
points(x = centre[1], y = centre[2], pch = 19, col = "red", cex = 1.5)
grid()
dev.off()
```

References

[Banks and Fienberg, 2003] Banks, D. L. and Fienberg, S. E. (2003). *Encyclopedia of Physical Science and Technology (Third Edition)*, chapter II. Statistics, Multivariate. Academic Press.

[Granger, 1969] Granger, C. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*.

[INSEE, 2000] INSEE (2000). La construction entre 1990 et 1997 : une crise profonde avant la reprise. Technical report, INSEE.

[Schwert, 1989] Schwert, W. (1989). Test for unit roots: A monte carlo investigation. *Journal of Business and Economic Statistics*.