ENSAE Paris

Linear time series project render

Time series analysis of shipbuilding in France

Corentin Pla and Lucas Degeorge

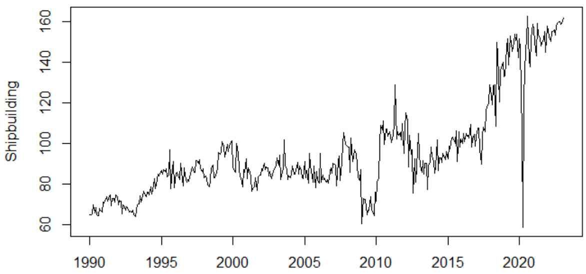
May 16th, 2023

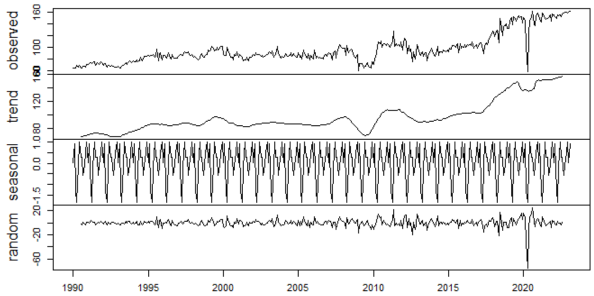
1 The data

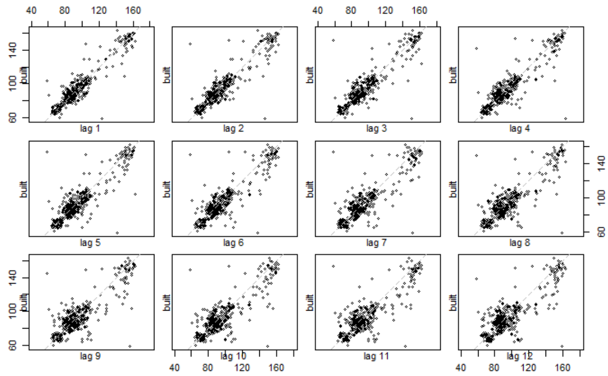
1.1 What do the data represent?

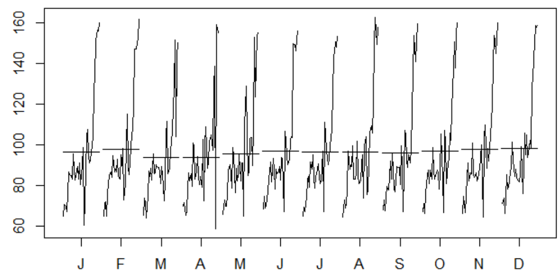
The series we study in this report represents the index of industrial production related to shipbuilding. The industrial production indices make it possible to monitor the monthly evolution of industrial activity in France and in construction. This index is computed with a Laspeyres' formula, with a fixed weighting corresponding to the added values of the various branches in the base year.

The series studied has 398 observations from January 1990 to February 2023 with a monthly frequency.





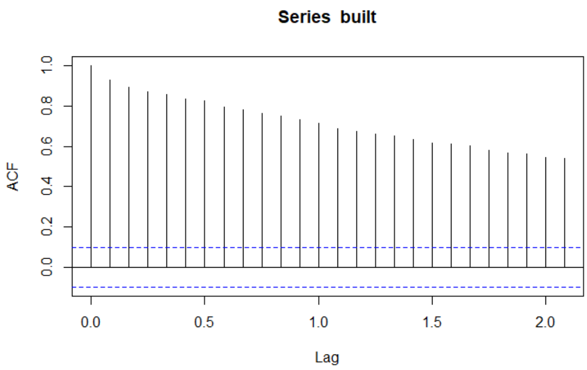


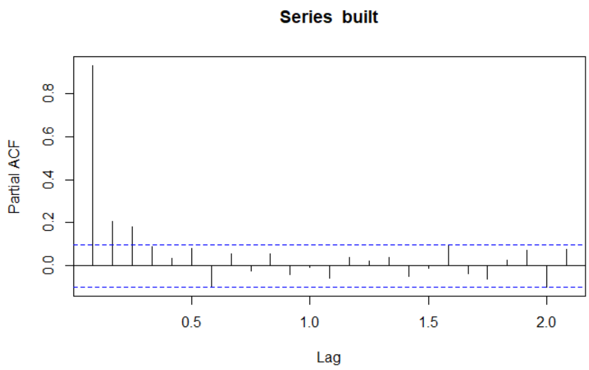


From these figures, we can see and conjecture that:

- The representation of the series suggests an increasing linear trend, but not necessarily a seasonality.
- The error represented by the decomposition appears to be constant over time, which confirms the additive model.
- The monthplot shows 12 similar monthly patterns, suggesting a lack of seasonality.
- The lagplot shows a strong correlation between the variables.

The next figure shows the auto-correlation (ACF) and the partial auto-correlation (PACF) functions:





Here, we notice that:

- The PACF does not show a repeated pattern. Thus, the series does not seem to show seasonality

- The auto-correlations decrease very gradually and the partial auto-correlation of order 1 is close to 1. Then, the series doesn't seem to be stationary.

In order to test our hypothesis, we run the KPSS and the Augmented Dickey–Fuller (ADF) tests. The results are reported in the next table

Test	Stats	Lag	p-value
KPSS	0.8752	5	≤ 0.01
ADF	-1.2717	21	0.8849

Table 1: Results of different tests on the series

The KPSS allows testing the stationarity of the series (the null hypothesis). Here, the results show a small p-value. Then, we reject the stationarity hypothesis at 5% level. The ADF test allows us to show the existence of a unit root in the case with a trend, and thus the non-stationarity of the series. The large p-value does not allow us to reject the hypothesis of a unit root at 5% level.

1.2 Make the series stationary again

We use the first difference method to stationarize our series: $X_t = \Delta S_t = S_t - S_{t-1}$, where S_t is our initial series. To verify that our new series is indeed stationary, we rerun the two tests mentioned above, as well as a trend test using linear regression on t. The results are reported in the next table.

Test	Stats	Lag	p-value
KPSS	0.066976	5	≥ 0.1
ADF	-5.9108	21	0.01
Regression on t			0.729

Table 2: Results of different tests on the differentiated series

The p-value of the KPSS test is well above 0.05, which allows us not to reject the hypothesis of stationarity with 95% confidence. Moreover, the p-value of the ADF test is well below 0.05, which allows us to reject the unit root hypothesis with 95% confidence. Finally, the p-value of the coefficient of the linear regression of X_t on t is about 0.73. This coefficient is therefore not significant. The hypothesis of a trend can also be rejected.

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

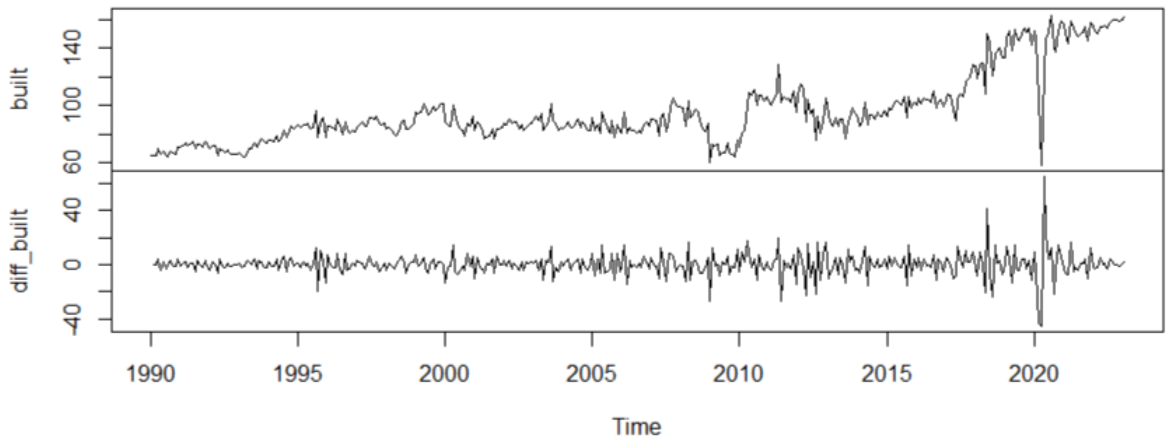


Figure 1: The series before and after differenciation

2 ARMA and ARIMA models

2.1 ARMA model

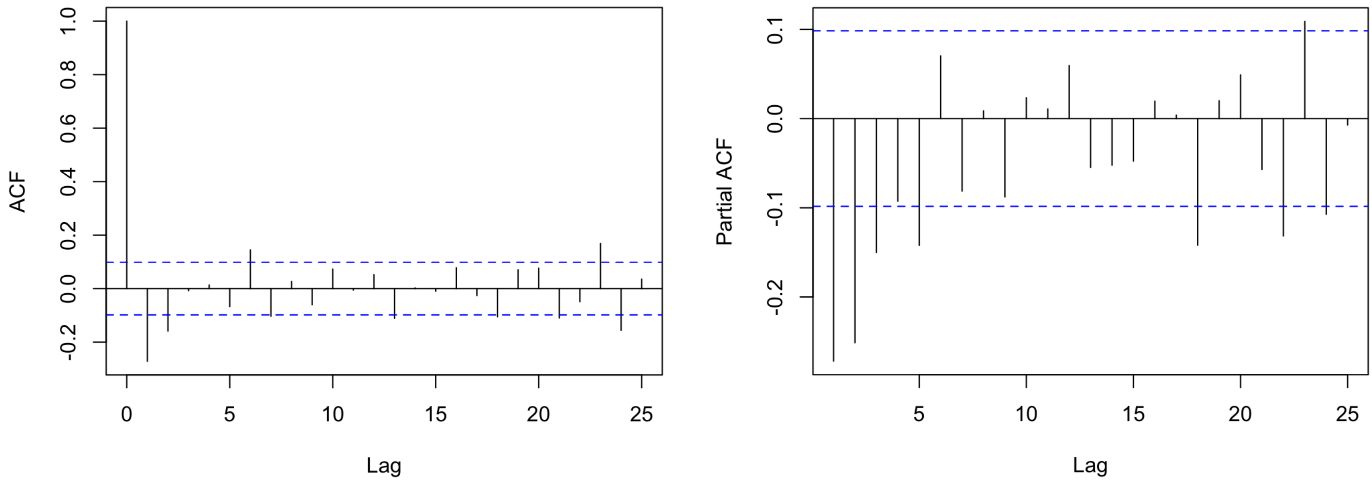


Figure 2: ACF and PACF of the order 1 differentiated serie

According to the figure above, ACF is significant until lag 3 and PACF until lag 5 so we set $q_{max} = 3$ and $p_{max} = 5$.

To determine which parameters to choose, we minimize the two well knowed information critierions :

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

with : $\hat{\sigma}^2 = \sum_{i=1}^{n} \hat{\epsilon}_t^2$

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

	q=0	q=1	q=2
p=0	2833.188	2778.022	2766.189
p=1	2804.996	2767.761	2768.182
p=2	2781.584	2768.080	2770.077
p=3	2774.892	2770.076	2768.357
p=4	2773.743	2772.039	2768.907
p=5	2768.212	2763.932	2765.851

	q=0	q=1	q=2
p=0	2837.172	2785.990	2778.141
p=1	2812.964	2779.713	2784.118
p=2	2793.536	2784.016	2789.996
p=3	2790.828	2789.995	2792.260
p=4	2793.662	2795.943	2796.795
p=5	2792.115	2791.820	2797.722

Figure 3: AIC and BIC of the order 1 differentiated serie

lag	p − value
1	NA
2	NA
3	NA
4	NA
5	NA
6	0.011987089
7	0.009721263
8	0.024845126
9	0.031765500
10	0.033426867
11	0.058765517
12	0.090077634
13	0.018666355
14	0.027411777
15	0.037596358
16	0.048190163
17	0.057065823
18	0.026002433
19	0.034498043
20	0.043828733
21	0.020278112
22	0.018197666
23	0.007912331
24	0.001228481

lag	p − value
1	NA
2	NA
3	NA
4	NA
5	NA
6	0.33843611
7	0.61047937
8	0.76030636
9	0.85339862
10	0.85350898
11	0.91860474
12	0.95661682
13	0.60365536
14	0.61027231
15	0.69290200
16	0.72099110
17	0.71828541
18	0.49665483
19	0.48008570
20	0.54035180
21	0.43011240
22	0.38195148
23	0.18725813
24	0.06120193

Figure 4: Test of autocorrelation of residuals for ARMA(0,2) and ARMA(5,1)

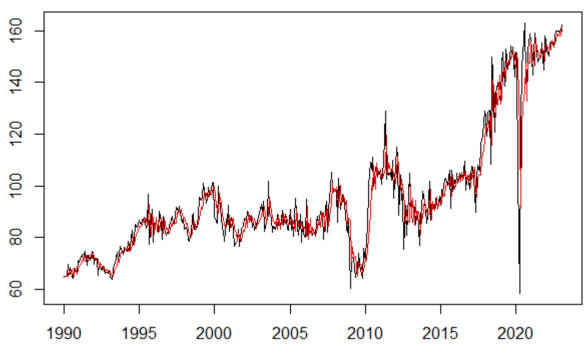
The values obtained for ARMA(0,2) show that the absence of autocorrelation of the residuals is always rejected, whereas the ones obtained for ARMA(5,1) show that the lack of autocorrelation of the residuals is never rejected. That's why we choose ARMA(5,1)

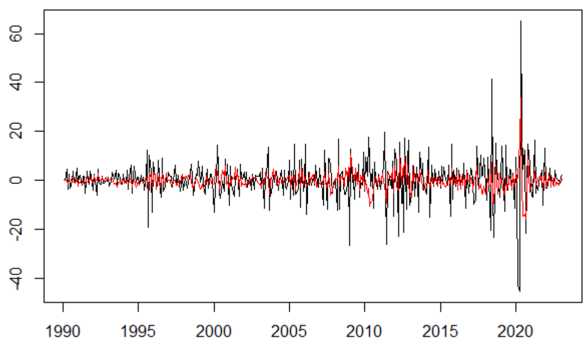
As for the adjusted R^2 we find 0.172216.

2.2 ARIMA model

We have differentiated the initial series once to obtain the series X_t . So d = 1. Thus, the model corresponding to the series we initially chose is the ARIMA(5,1,1) model.

The two following figures show the observed series (in black) and the series predicted by the model. On the left, this is the series against the ARIMA model prediction and on the right, the differentiated series against the ARMA model prediction.





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 $\epsilon_t \sim \mathcal{N}\left(0, \sigma_\epsilon^2\right)$. We have a model ARMA$(5,1)$ which is written :

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \phi_3 X_{t-3} + \phi_4 X_{t-4} + \phi_5 X_{t-5} + \epsilon_t - \theta_1 \epsilon_{t-1}$$

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

$$\begin{cases} \hat{X}_{T+1|T} = \phi_1 X_T + \phi_2 X_{T-1} + \phi_3 X_{T-2} + \phi_4 X_{T-3} + \phi_5 X_{T-4} - \theta_1 \epsilon_T \\ \hat{X}_{T+2|T} = \phi_1 \hat{X}_{T+1|T} + \phi_2 X_T + \phi_3 X_{T-1} + \phi_4 X_{T-2} + \phi_5 X_{T-3} \end{cases}$$

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

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

$X - \hat{X}$ thus follows a normal distribution with parameter $\mu = 0$ et Σ, i.e $X - \hat{X} \sim \mathcal{N}(0, \Sigma)$ où Σ is the variance-covariance matrix such that :

$$\Sigma = \sigma_\epsilon^2 \begin{pmatrix} 1 & \theta_1 + \phi_1 \\ \theta_1 + \phi_1 & 1 + (\theta_1 + \phi_1)^2 \end{pmatrix}$$

As $\text{Det}(\Sigma) = \sigma_\epsilon^2$, the variance covariance matrix is invertible if and only if $\sigma_\epsilon^2 > 0$, what we have assumed to be true.

According to the course, we finally get ${}^t(X - \hat{X})\Sigma^{-1}(X - \hat{X}) \sim \chi^2(2)$. which allows us to directly deduce the confidence region of level α. We thus get $\forall \alpha \in [0,1]$:

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

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

3.2 Hypothetis

We have assumed that our residuals are Gaussian, let's check this assumption using the figure 5 below. The blue curve represents a normal distribution of mean and variance, followed by our residuals. The black curve represents the density of our residuals. The two curves have a similar trend, but they do not merge. The assumption is therefore probably a bit strong for our model, even though the black curve looks like a normal distribution. In order to get a clearer picture, we performed the Jarque Bera test, which is more suitable for a number of observations greater than 50 data. Unfortunately, the p-value of our test is very low (p-value $\leq 2,2 \cdot 10^{16}$), which does not allow us to confirm the normality hypothesis.

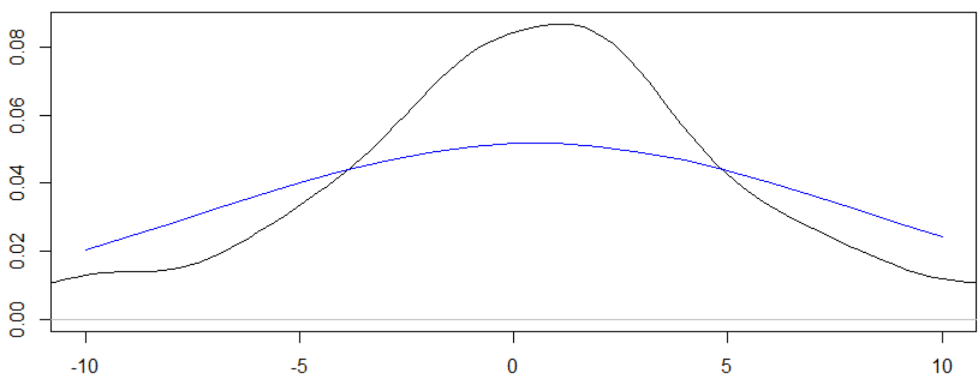


Figure 5: Density of the residuals

Moreover, we have considered the errors as innovations, uncorrelated to each other and to the past values of our series. This hypothesis is verified only if the polynomial in canonical writing does not admit a root inside the unit circle. In our case,

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \phi_3 X_{t-3} + \phi_4 X_{t-4} + \phi_5 X_{t-5} + \epsilon_t + \theta_1 \epsilon_{t-1}$$

The different values of the coefficients are reported in the next table :

Coefficients	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	θ_1
Values	-1.098637	-0.6195126	-0.4494547	-0.2645472	-0.1909255	-0.7243604

Table 3: Values of the coefficients of the ARMA(5,1) model

Thus, the roots (reported in the next table) of the polynomials Φ and Θ are well outside the unit circle.

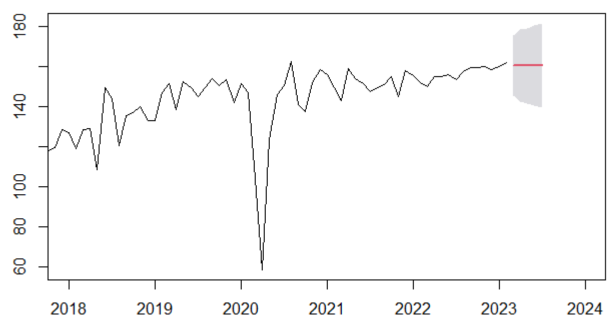
Coefficients	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	θ_1
Roots	1.513487	1.116450	1.431100	1.513487	1.431100	1.380528

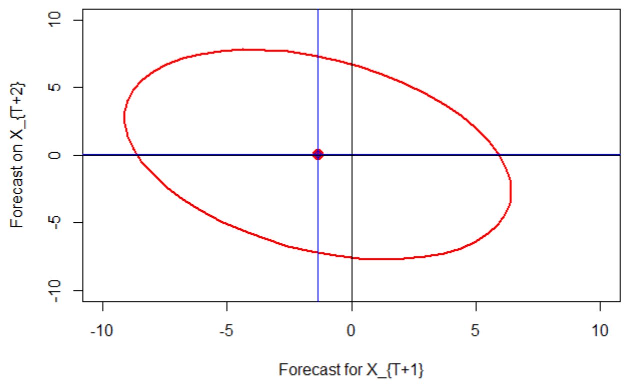
Table 4: Roots of the coefficients of the ARMA(5,1) model

Finally, it is necessary that the specification of our model is adapted and that the parameters are correctly estimated. Here, the comparison of the two curves above shows that the Gaussian model overestimates the variance of our effective residuals. our effective residuals.

3.3 Results and graphs

The two following figures show the predictions and their confidence regions.





The figure on the left shows the univariate prediction of values of (S_{t+1}, S_{T+2}) with an ARIMA(5,1,1) model and a 95% confidence interval. The other figure represents the bivariate elliptical confidence region (the variance-covariance matrix is symmetric positive-definite), at 95% of (X_{t+1}, X_{T+2}).

The region of confidence is rather large. The prediction is therefore not very accurate. One hypothesis that could explain this is the peak in 2020, probably due to the Covid-19 crisis.

3.4 Open question

Knowledge of Y_{t+1} can improve the prediction of X_{t+1} if Y_t instantaneously causes X_t in the sense of Granger sense, i.e. :

$$\hat{X}_{t=1|\{X_u, Y_u; u \leq t\} \cap \{T_{T+1}\}} \neq \hat{X}_{t=1|\{X_u, Y_u; u \leq t\}}$$

This condition is characterised by the correlation between the two residuals of a VAR model of (X, Y), and is testable by a Wald test.

4 Appendix

Here is the R code we use during our work.

```r

require(zoo)
require(tseries)

library(readr)
library(tidyverse)
library(plyr)
library(questionr)
library(corrplot)
library(Hmisc)
library(lmtest)
library(margins)
library(psych)

library(fUnitRoots)

library(forecast)

require(ellipse)
require(ellipsis)
# require(car)
library(ellipse)

path <- "C:/Users/lucas/Documents/GitHub/Linear_time_series_electricity"
setwd(path)
getwd()

# Loading of data

datafile <- "valeurs_mensuelles.csv"
data <- as.data.frame(read.csv(datafile,sep=";"))

data <- data[,-3]
data <- apply(data, 2, rev)

rownames(data) <- 1:dim(data)[1]
built <- ts(as.numeric(data[,2]), start=1990, frequency=12)
n <- length(built)
plot(built, xlab="Date", ylab="Shipbuilding", main = "Shipbuilding")
monthplot(built)

## Part I ##

# Question 1

plot(built, xlab="Date", ylab="Shipbuilding", main = "Shipbuilding")
monthplot(built)
lag.plot(built, lags=12, layout=c(3,4), do.lines=FALSE)
fit1 <- decompose(built)
plot(fit1)

# Plot ACF and PACF
acf(built)
pacf(built)

summary(lm(built~seq(1,n)))

# KPSS test
kpss.test(built, null="Trend")
```

```r

# ADF test

# Function Q_tests for testing th autocorrelation of residuals
Qtests <- function(series, k, fitdf=0) {
  aux <- function(l){
    pval <- if (l<=fitdf) NA else Box.test(series, lag=l, type="Ljung", fitdf=fitdf)$p.value
    return (c("lag"=l, "pval"=pval))
  }
  pvals <- apply(matrix(1:k), 1, FUN=aux)
  return (t(pvals))
}

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 = TRUE) == 0) {
      noautocorr <- 1
      cat("OK \n")
    } else {
      cat("nope \n")
    }
    k <- k + 1
  }
  return(adf)
}

adf <- adfTest_valid(built, 24, "ct")
adf

# Question 2

diff_built = diff(built,1)
plot(diff_built)

# calculate autocorrelation
acf(diff_built, pl=TRUE)

summary(lm(diff_built ~ seq(1, length(diff_built))))
kpss.test(diff_built, null="Level")

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))
}

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 = TRUE) == 0) {
      noautocorr <- 1
      cat("OK \n")
```

```r
    } else {
      cat("nope \n")
    }
    k <- k + 1
  }
  return(adf)
}

adf <- adfTest_valid(diff_built, 24, "ct")
adf


# Question 3

# Representation before and after
plot(cbind(built,diff_built))

## Part II ##

# Question 4

# calculate autocorrelation
acf(as.numeric(diff_built), pl=TRUE)
q_max <- 2

# calculate partial autocorrelation
pacf(as.numeric(diff_built), pl=TRUE)
p_max <- 5

# Matrix of AICs and BICs
mat <- matrix (NA, nrow=p_max+1, ncol=q_max+1) # empty matrix
rownames(mat) <- paste0("p=",0:p_max)
colnames(mat) <- paste0("q=",0:q_max)
AICs <- mat # AIC matrix
BICs <- mat # BIC matrix
pqs <- expand.grid(0:p_max, 0:q_max)
for (row in 1:dim(pqs)[1]){
  p <- pqs[row, 1]
  q <- pqs[row, 2]
  # try to estimate the ARIMA
  estim <- try(arima(diff_built, 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)
}

# display AICs
AICs
AICs==min(AICs)
# display BICs
BICs
BICs==min(BICs)

# Interpretation: we choose ARMA(0,2) and ARMA(5,1)

arma02 <- arima(diff_built, c(0, 0, 2), include.mean=F)
arma02

arma51 <- arima(diff_built, c(5, 0, 1), include.mean=F)
arma51

Qtests(arma02$residuals, 24, fitdf=5)
Qtests(arma51$residuals, 24, fitdf=5)

```

```r
# Function adj_r2 for computing the adjusted R square
adj_r2 <- function(model){
  ss_res <- sum(model$residuals^2) # sum of squared residuals
  p <- model$arma[1]
  q <- model$arma[2]
  ss_tot <- sum(diff_built[-c(1:max(p, q))]^2)
  n <- model$nobs-max(p, q)
  adj_r2 <- 1-(ss_res/(n-p-q-1)) / (ss_tot/(n-1)) #ajusted R square
  return (adj_r2)
}
adj_r2(arma51)

# Question 5
arima012 <- arima(built, c(0, 1, 2), include.mean=F)
arima012

arima511 <- arima(built, c(5, 1, 1), include.mean=F)
arima511

plot(built, xlab="Date" , ylab="Indice", main = "Observed vs. Predicted" )
lines(fitted(arima511), col = "red")

plot(diff_built, xlab="Date", ylab="Indice", main="Observed vs. Predicted" )
lines(fitted(arma51), col = "red")

## Part III ##

# Question 7

tsdiag(arma51)
jarque.bera.test(arima511$residuals)
qqnorm(arma51$residuals)
plot(density(arma51$residuals, lwd=0.5), xlim=c(-10,10), main="Density of residuals")
mu <- mean(arma51$residuals)
sigma <- sd(arma51$residuals)
x <- seq(-10,10)
y <- dnorm(x,mu,sigma)
lines(x, y, lwd=0.5, col="blue")

# Question 8

arma51$coef
phi_1 <- as.numeric(arma51$coef[1])
phi_2 <- as.numeric(arma51$coef[2])
phi_3 <- as.numeric(arma51$coef[3])
phi_4 <- as.numeric(arma51$coef[4])
phi_5 <- as.numeric(arma51$coef[5])
theta <- as.numeric(arma51$coef["ma1"])
sigma2 <- as.numeric(arma51$sigma)
phi_1
phi_2
phi_3
phi_4
phi_5
theta
sigma2

# We check the roots :
ar_coefs <- c(phi_1, phi_2, phi_3, phi_4, phi_5)
ma_coefs <- c(theta)

# Check if roots are outside the unit circle
ar_roots <- polyroot(c(1, -ar_coefs))
```

```r
ma_roots <- polyroot(c(1, ma_coefs))

abs(ar_roots)
abs(ma_roots)

all(abs(ar_roots) > 1)
all(abs(ma_roots) > 1)


# Prediction

XT1 = predict(arma51, n.ahead=2)$pred[1]
XT2 = predict(arma51, n.ahead=2)$pred[2]
XT1
XT2

# Prediction for the serie built
fore = forecast(arima511, h=5, level=95)
par(mfrow=c(1,1))
plot(fore, xlim=c(2018,2024), col=1, fcol=2, shaded=TRUE, xlab="Time" , ylab="Value",
     main="Forecast for the serie built")

arma <- arima0(diff_built, order = c(5, 1, 1))
sigma2 <- arma$sigma2
phi <- arma$coef[-1]

Sigma <- matrix(c(sigma2, phi[1] * sigma2, phi[2] * sigma2, phi[3] * sigma2, phi[4] * sigma2, phi[5] * sigma2,
                  phi[1] * sigma2, sigma2, 0, 0, 0, 0,
                  phi[2] * sigma2, 0, sigma2, 0, 0, 0,
                  phi[3] * sigma2, 0, 0, sigma2, 0, 0,
                  phi[4] * sigma2, 0, 0, 0, sigma2, 0,
                  phi[5] * sigma2, 0, 0, 0, 0, sigma2), ncol = 6)

plot(XT1, XT2, xlim = c(-10, 10), ylim = c(-10, 10), xlab = "Forecast for X_{T+1}", ylab = "Forecast on X_{T+2}", main = "95% b
ellipse(Sigma[1:2, 1:2], center = c(XT1, XT2), type = "l", col = "red", radius = c(1, 1))
points(XT1, XT2, col = "blue")
abline(h=XT2,v=XT1, col="blue")
abline(h=0,v=0)
```