Linear Time Series Assignment

By : Kanupriya Jain, Soumodeep Hoodaty

Contents

1 PART-I

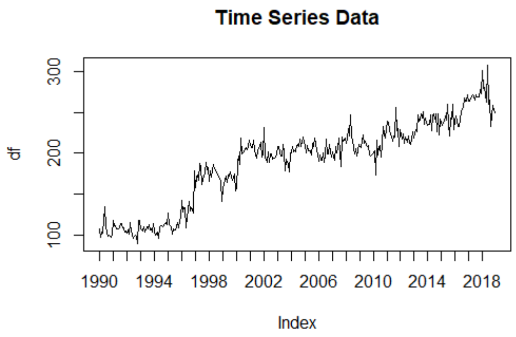
1.1 Question 1

The chosen series represents the industrial production indices for the extraction of ornamental and construction stones, industrial limestone, gypsum, chalk, and slate. These indices are calculated in accordance with the European Regulation on short-term statistics (Regulation (EU) 2019/2152) and are vital for monitoring monthly changes in industrial and construction activities in France. They serve as crucial indicators for tracking the business cycle and identifying early turning points, complementing other macroeconomic indicators such as employment statistics, price indices, services production index, and foreign trade statistics. Additionally, these indices contribute to the elaboration of French quarterly accounts, including the GDP flash estimate.

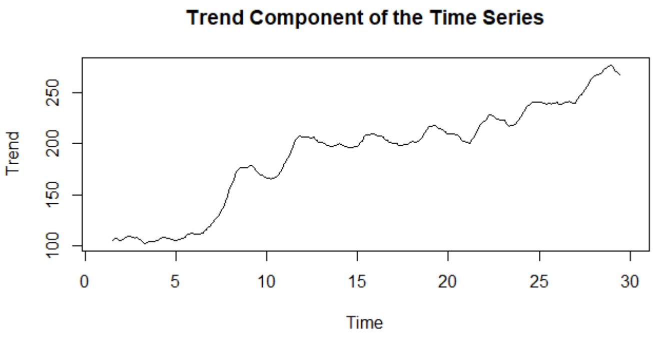
The data for these indices can be accessed from the following link: CVS-CJO index of industrial production

1.2 Question 2

The following figure (a) represents the series over 28 years. We have taken the data from the year 1990 to 2018. We are not checking for the seasonality of the series since we chose a seasonally corrected series as mentioned. We feel that there is a presence of an increasing trend in the series, so we decomposed it using `decompose()` function and extracted the trend component. Decomposing the time series involves trying to separate the time series into these individual components. Figure (b) shows the trend component of the time series and confirms our observation.



((a)) Illustration Of Series



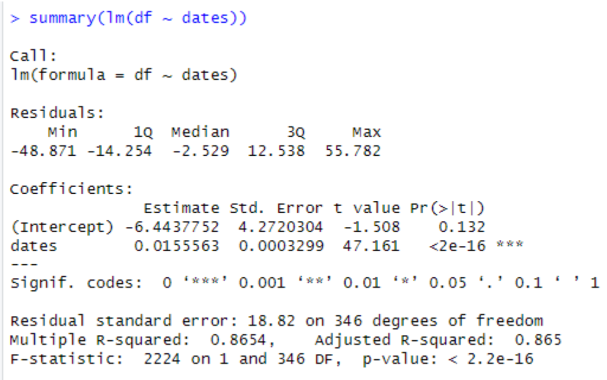
((b)) Decomposition Of Series

<u>Unit Root Tests</u>

We will perform mainly two Unit Root Tests to confirm our hypothesis for the chosen series-

Augmented Dickey–Fuller test (ADF) Test

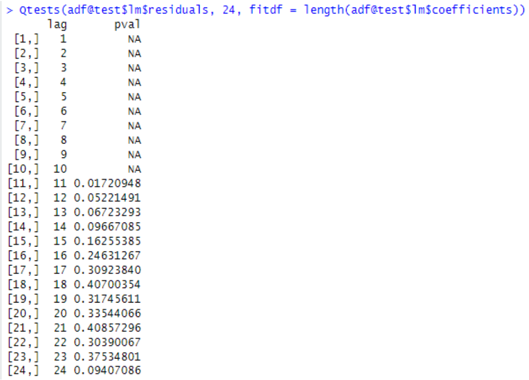
Before performing the ADF Unit root test, we need to check if there is an intercept and/or a non-null linear trend.



```
> summary(lm(df ~ dates))

Call:
lm(formula = df ~ dates)

Residuals:
    Min      1Q  Median      3Q     Max 
-48.871 -14.254  -2.529  12.538  55.782 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -6.4437752  4.2720304  -1.508    0.132    
dates        0.0155563  0.0003299  47.161   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 18.82 on 346 degrees of freedom
Multiple R-squared:  0.8654,	Adjusted R-squared:  0.865 
F-statistic:  2224 on 1 and 346 DF,  p-value: < 2.2e-16
```

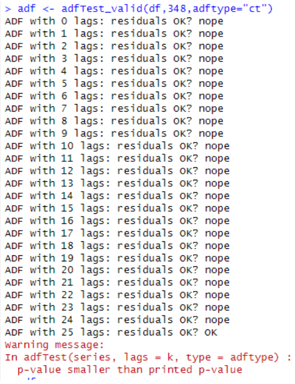
The coefficient associated with the linear trend (dates) is positive, and maybe statistically significant (which cannot be confirmed because the test is not valid where there are possibly autocorrelated residuals). Before proceeding with the test, we will check that the model's residuals are not autocorrelated by running Q tests, otherwise, ADF test would not be valid.

We reject the absence of residual autocorrelation at least once (Q(11)), thus invalidating the ADF test without lags. We added lags of ΔX_t until the residuals were no longer autocorrelated.



```
> Qtests(adf@test$lm$residuals, 24, fitdf = length(adf@test$lm$coefficients))
      lag       pval
 [1,]   1         NA
 [2,]   2         NA
 [3,]   3         NA
 [4,]   4         NA
 [5,]   5         NA
 [6,]   6         NA
 [7,]   7         NA
 [8,]   8         NA
 [9,]   9         NA
[10,]  10         NA
[11,]  11 0.01720948
[12,]  12 0.05221491
[13,]  13 0.06723293
[14,]  14 0.09667085
[15,]  15 0.16255385
[16,]  16 0.24631267
[17,]  17 0.30923840
[18,]  18 0.40700354
[19,]  19 0.31745611
[20,]  20 0.33544066
[21,]  21 0.40857296
[22,]  22 0.30390067
[23,]  23 0.37534801
[24,]  24 0.09407086
```

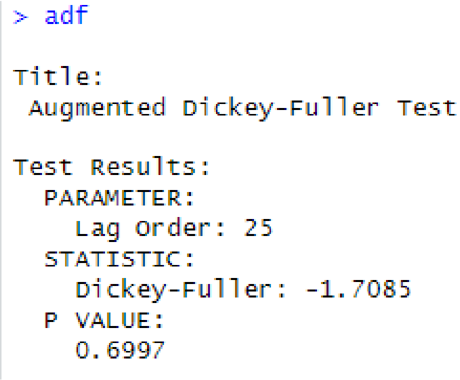
((a)) Q Tests



```
> adf <- adfTest_valid(df,348,adftype="ct")
ADF with 0 lags: residuals OK? nope
ADF with 1 lags: residuals OK? nope
ADF with 2 lags: residuals OK? nope
ADF with 3 lags: residuals OK? nope
ADF with 4 lags: residuals OK? nope
ADF with 5 lags: residuals OK? nope
ADF with 6 lags: residuals OK? nope
ADF with 7 lags: residuals OK? nope
ADF with 8 lags: residuals OK? nope
ADF with 9 lags: residuals OK? nope
ADF with 10 lags: residuals OK? nope
ADF with 11 lags: residuals OK? nope
ADF with 12 lags: residuals OK? nope
ADF with 13 lags: residuals OK? nope
ADF with 14 lags: residuals OK? nope
ADF with 15 lags: residuals OK? nope
ADF with 16 lags: residuals OK? nope
ADF with 17 lags: residuals OK? nope
ADF with 18 lags: residuals OK? nope
ADF with 19 lags: residuals OK? nope
ADF with 20 lags: residuals OK? nope
ADF with 21 lags: residuals OK? nope
ADF with 22 lags: residuals OK? nope
ADF with 23 lags: residuals OK? nope
ADF with 24 lags: residuals OK? nope
ADF with 25 lags: residuals OK? OK
Warning message:
In adfTest(series, lags = k, type = adftype) :
  p-value smaller than printed p-value
```

((b)) Valid ADF test for chosen series

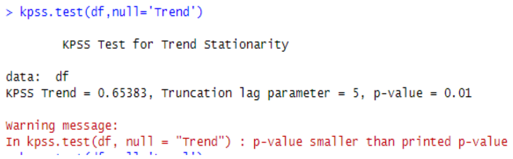
After adding lags, we can observe in the following valid ADF Test that the p-value is greater than 0.05 indicating that we are not rejecting the null hypothesis which says that the series is not stationary. So, by ADF test, we can conclude that the series is not stationary.



```
> adf

Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 25
  STATISTIC:
    Dickey-Fuller: -1.7085
  P VALUE:
    0.6997
```

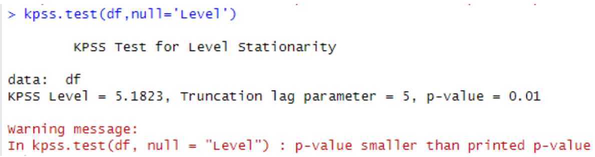
Figure 3: Result for ADF test for chosen series

Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test



```
> kpss.test(df,null='Trend')

        KPSS Test for Trend Stationarity

data:  df
KPSS Trend = 0.65383, Truncation lag parameter = 5, p-value = 0.01

Warning message:
In kpss.test(df, null = "Trend") : p-value smaller than printed p-value
```

((a)) KPSS Test for Trend Stationarity



```
> kpss.test(df,null='Level')

        KPSS Test for Level Stationarity

data:  df
KPSS Level = 5.1823, Truncation lag parameter = 5, p-value = 0.01

Warning message:
In kpss.test(df, null = "Level") : p-value smaller than printed p-value
```

((b)) KPSS Test for Level Stationarity

We are getting the following result indicating that the series is not stationary. The null hypothesis of the KPSS test is that the series is stationary and we are rejecting the null hypothesis.

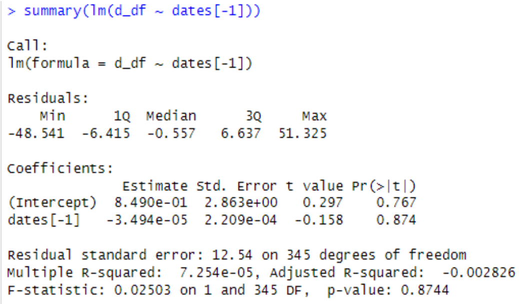
Now,we will use `diff()` to differentiate the series once and then check if the trend and non- stationarity have been removed. We are calling differentiated series as `d_df`. We can use the same tests again to check the stationarity of the differentiated series.

Augmented Dickey–Fuller test (ADF) Test for Differentiated Series

There is not any constant or significant trend. So, we performed the ADF test in the no-constant and no-trend case.

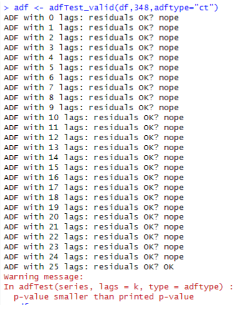
We ran the following command and found that 26 lags needed to be included -

```
adf1 <- adfTest_valid(d_df,348,adftype="nc")
```



```
> summary(lm(d_df ~ dates[-1]))

Call:
lm(formula = d_df ~ dates[-1])

Residuals:
    Min      1Q  Median      3Q     Max 
-48.541  -6.415  -0.557   6.637  51.325 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.490e-01  2.863e+00   0.297    0.767
dates[-1]   -3.494e-05  2.209e-04  -0.158    0.874

Residual standard error: 12.54 on 345 degrees of freedom
Multiple R-squared:  7.254e-05,	Adjusted R-squared:  -0.002826 
F-statistic: 0.02503 on 1 and 345 DF,  p-value: 0.8744
```

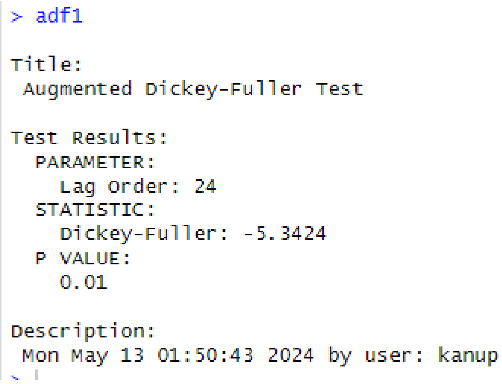
((a)) To check trend and constant



```
> adf <- adfTest_valid(df,348,adftype="ct")
ADF with 0 lags: residuals OK? nope
ADF with 1 lags: residuals OK? nope
ADF with 2 lags: residuals OK? nope
ADF with 3 lags: residuals OK? nope
ADF with 4 lags: residuals OK? nope
ADF with 5 lags: residuals OK? nope
ADF with 6 lags: residuals OK? nope
ADF with 7 lags: residuals OK? nope
ADF with 8 lags: residuals OK? nope
ADF with 9 lags: residuals OK? nope
ADF with 10 lags: residuals OK? nope
ADF with 11 lags: residuals OK? nope
ADF with 12 lags: residuals OK? nope
ADF with 13 lags: residuals OK? nope
ADF with 14 lags: residuals OK? nope
ADF with 15 lags: residuals OK? nope
ADF with 16 lags: residuals OK? nope
ADF with 17 lags: residuals OK? nope
ADF with 18 lags: residuals OK? nope
ADF with 19 lags: residuals OK? nope
ADF with 20 lags: residuals OK? nope
ADF with 21 lags: residuals OK? nope
ADF with 22 lags: residuals OK? nope
ADF with 23 lags: residuals OK? nope
ADF with 24 lags: residuals OK? nope
ADF with 25 lags: residuals OK? OK
Warning message:
In adfTest(series, lags = k, type = adftype) :
  p-value smaller than printed p-value
```

((b)) Result for ADF test for differentiated series

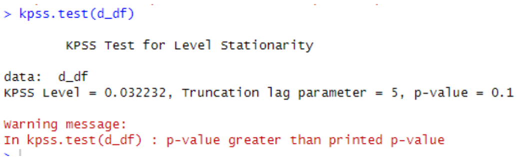
We can observe that the p-value is less than 0.05 indicating that we are rejecting the null hypothesis which says that the series is not stationary. So, by ADF test, we can conclude that this new series is stationary.



```
> adf1

Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 24
  STATISTIC:
    Dickey-Fuller: -5.3424
  P VALUE:
    0.01 

Description:
 Mon May 13 01:50:43 2024 by user: kanup
```

Figure 6: Result for ADF test for differentiated series

Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test for Differentiated Series



```
> kpss.test(d_df)

        KPSS Test for Level Stationarity

data:  d_df
KPSS Level = 0.032232, Truncation lag parameter = 5, p-value = 0.1

Warning message:
In kpss.test(d_df) : p-value greater than printed p-value
```

Figure 7: Result for KPSS test for differentiated series

We are getting the following result indicating that the new series is stationary. The null hypothesis of the KPSS test is that the series is stationary and we are not rejecting the null hypothesis.

1.3 Question 3

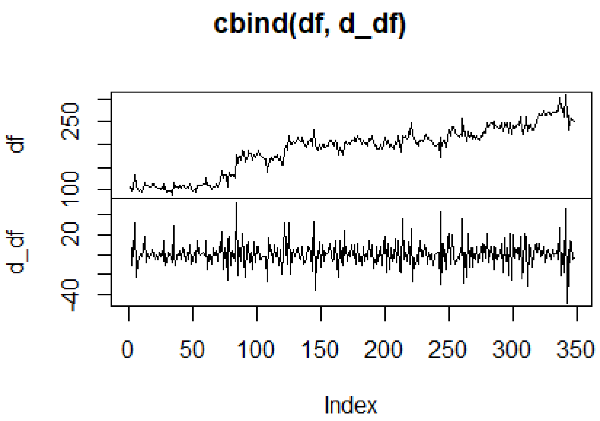


Figure 8: Illustration of Chosen series and differentiated series

2 PART - II

2.1 Question 4

In order to select the right ARMA model for the new series, we will observe the ACF and PACF of the series.

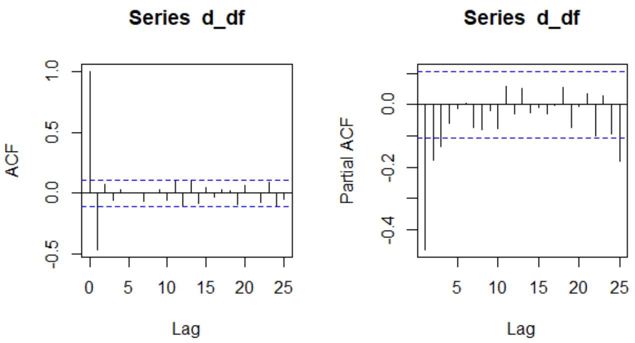
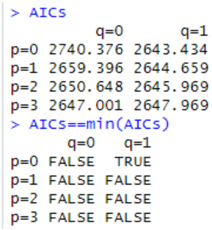


Figure 9: ACF and PACF of differentiated series

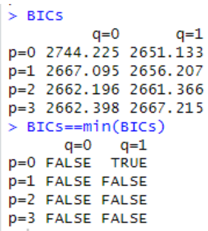
The ACF is statistically significant in the first order maximum, we will therefore choose $q* = 1$. The PACF is also statistically significant in the third order maximum, we will pick $p* = 3$. The potential models are all the ARIMA(p,1,q) for `df` and ARIMA(p,0,q) for `d_df` where $p \leq 3$ and $q \leq 1$. Let's compute the AIC *(Akaike information criterion)* and BIC *(Bayesian information criterion)* for each of those models.

The following figure displays the values for AIC's and BIC's for values of $p \leq 3$ and $q \leq 1$. The value TRUE and FALSE indicate the values of p and q for which model is attaining minimum AIC and BIC value.



```
> AICs
         q=0      q=1
p=0 2740.376 2643.434
p=1 2659.396 2644.659
p=2 2650.648 2645.969
p=3 2647.001 2647.969
> AICs==min(AICs)
      q=0   q=1
p=0 FALSE  TRUE
p=1 FALSE FALSE
p=2 FALSE FALSE
p=3 FALSE FALSE
```

((a)) AIC's



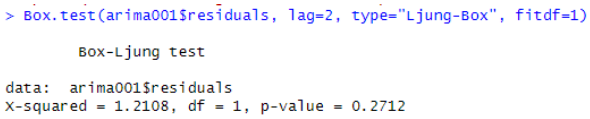
```
> BICs
         q=0      q=1
p=0 2744.225 2651.133
p=1 2667.095 2656.207
p=2 2662.196 2661.366
p=3 2662.398 2667.215
> BICs==min(BICs)
      q=0   q=1
p=0 FALSE  TRUE
p=1 FALSE FALSE
p=2 FALSE FALSE
p=3 FALSE FALSE
```

((b)) BIC'c

The ARIMA(0,0,1) minimizes the AIC and BIC both, for d_df. We thus keep it.

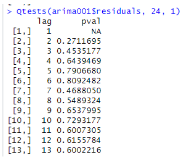
Now, we would like to check the validity of the model. The validity of the model hinges on the absence of autocorrelation within the residuals. This can be assessed through the Ljung-Box test which examines whether there is joint nullity of autocorrelations within the residuals up to the given order k, indicating the absence of autocorrelation.

Following figure gives the results. We are taking `fitdf = p+q = 1`. Since, the p-value $>$ 0.05, we will not reject the null hypothesis.



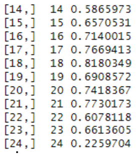
```
> Box.test(arima001$residuals, lag=2, type="Ljung-Box", fitdf=1)

	Box-Ljung test

data:  arima001$residuals
X-squared = 1.2108, df = 1, p-value = 0.2712
```

To further ensure the absence of autocorrelation, we performed the Q test for two periodicities and concluded the absence of auto correlations and thus the model is ***"valid"***.



```
> Qtests(arima001$residuals, 24, 1)
      lag      pval
 [1,]   1        NA
 [2,]   2 0.2711695
 [3,]   3 0.4535177
 [4,]   4 0.6439469
 [5,]   5 0.7906680
 [6,]   6 0.8092482
 [7,]   7 0.4688050
 [8,]   8 0.5489324
 [9,]   9 0.6537995
[10,]  10 0.7293177
[11,]  11 0.6007305
[12,]  12 0.6155784
[13,]  13 0.6002216
```

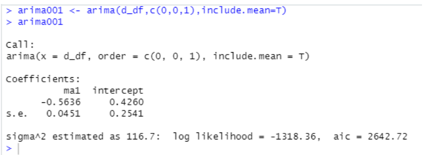
((a))



```
[14,]  14 0.5865973
[15,]  15 0.6570531
[16,]  16 0.7140015
[17,]  17 0.7669413
[18,]  18 0.8180349
[19,]  19 0.6908572
[20,]  20 0.7418367
[21,]  21 0.7730173
[22,]  22 0.6078118
[23,]  23 0.6613605
[24,]  24 0.2259704
```

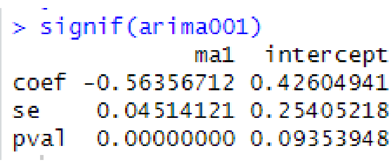
((b))

We also need to check if the model is well-adjusted. Following image shows the coefficient of the equation for the differentiated series. We have not centered the series before fitting it so we are including mean while fitting.



```
> arima001 <- arima(d_df,c(0,0,1),include.mean=T)
> arima001

Call:
arima(x = d_df, order = c(0, 0, 1), include.mean = T)

Coefficients:
          ma1  intercept
      -0.5636     0.4260
s.e.   0.0451     0.2541

sigma^2 estimated as 116.7:  log likelihood = -1318.36,  aic = 2642.72
> 
```

((a))



```
> signif(arima001)
             ma1  intercept
coef -0.56356712 0.42604941
se    0.04514121 0.25405218
pval  0.00000000 0.09353948
```

((b))

The coefficient of MA(1) shows statistical significance (with the ratio between the estimated coefficient and the standard error which is 12.4966 which exceeds 1.96 in absolute value), indicating that the ARIMA(0,0,1) model is ***"well-adjusted"***. As for the statistical significance of coefficients, we can check that we do have indeed a ratio between the estimated coefficient and the estimated variance that is well above 1.96 in absolute value(or if the corresponding p-value is lower than 0.05 which is true in our case).

2.2 Question 5

We observed that we differentiated series once and `d_df` became stationary. We also observed that the differentiated series was effectively modeled by $ARIMA(0,0,1)$ so we can choose $ARIMA(0,1,1)$ for chosen series.

We can write the equation of our model for differentiated series as per the convention of parameterization followed by the R language as follows-

$$X_t = 0.4260 + \epsilon_t - 0.5636\epsilon_{t-1}$$

Note that $\phi(z)$ and $\psi(z)$ are polynomials and have no common roots and $\phi(z) \neq 0 \; \forall |z| \leq 1$ and hence, we have a causal stationary solution (by definitions done in class of ARMA(p,q)) model and causal stationary solution).

Using the above equation, we can write the equation of our chosen series as follows (denoted by Y_t) -

$$Y_t = Y_{t-1} + 0.4260 + \epsilon_t - 0.5636\epsilon_{t-1}$$

3 PART-III

3.1 Question 6

Here, we denote T as the length of the series, and assume that the residuals of the series are Gaussian, i.e., $\epsilon_t \sim N(0, \sigma^2)$ where $\sigma^2 > 0$. Here, we can observe that our differentiated series X_t admits canonical $ARMA(0,1)$ and ϵ_t is the linear innovation of X_t. The best linear prediction of X_{T+1} is $\hat{X}_{T+1|T}$ which is given by -

$$\begin{aligned} \hat{X}_{T+1|T} &= EL[X_{T+1}|X_T, X_{T-1}, \ldots] \\ &= EL[0.4260 + \epsilon_{T+1} - 0.5636\epsilon_T|X_T, X_{T-1}, \ldots] \\ &= 0.4260 - 0.5636\epsilon_T \end{aligned} \tag{1}$$

Similarly, we can find $\hat{X}_{T+2|T}$ as follows-

$$\begin{aligned} \hat{X}_{T+2|T} &= EL[X_{T+2}|X_T, X_{T-1}, \ldots] \\ &= EL[0.4260 + \epsilon_{T+2} - 0.5636\epsilon_{T+1}|X_T, X_{T-1}, \ldots] \\ &= 0.4260 \end{aligned} \tag{2}$$

Let's call the vector of predicted values $\hat{\mathbf{X}} = \begin{pmatrix} \hat{X}_{T+1|T} \\ \hat{X}_{T+2|T} \end{pmatrix} = \begin{pmatrix} 0.4260 - 0.5636\epsilon_T \\ 0.4260 \end{pmatrix}$

and $\mathbf{X} = \begin{pmatrix} X_{T+1} \\ X_{T+2} \end{pmatrix}$. We can observe that $\mathbf{X} - \hat{\mathbf{X}} = \begin{pmatrix} \epsilon_{T+1} \\ \epsilon_{T+2} - 0.5636\epsilon_{T+1} \end{pmatrix}$

We know $\epsilon_t \sim N(0, \sigma^2)$ with $\sigma^2 > 0$, so we can say $X - \hat{X} \sim N(0, \Sigma)$ where Σ is the covariance matrix given by -

$$\Sigma = \sigma^2 \begin{pmatrix} 1 & -0.5636 \\ -0.5636 & (1 + 0.5636^2) \end{pmatrix} = \sigma^2 \begin{pmatrix} 1 & -0.5636 \\ -0.5636 & 1.31764496 \end{pmatrix}$$

Clearly, $det(\Sigma) = \sigma^4 > 0$. So, Σ is invertible. It is also real and symmetric. We can apply the transformation $Y = \Sigma^{-1/2}(X - \hat{X})$ resulting in Y following bivariate standard Normal distribution i.e. $Y \sim N_2(0, Id)$ where Id is a 2x2 Identity matrix. Then we can say squared norm of Y will follow chi-squared distribution with 2 degrees of freedom. $||Y||^2 = ||\Sigma^{-1/2}(X - \hat{X})||^2 = (X - \hat{X})^T \Sigma^{-1}(X - \hat{X}) \sim \chi^2_2$.Therefore, confidence region of any level $\alpha \in [0,1]$ can be given by-

$$R_\alpha = \{X \in \mathcal{R}^2 : (X - \hat{X})^T \Sigma^{-1}(X - \hat{X}) \leq q^{\chi^2_2}_{1-\alpha}\}$$

where $q^{\chi^2_2}_{1-\alpha}$ is the $(1-\alpha)^{th}$ quantile of chi-squared distribution with two degrees of freedom

3.2 Question 7

We work on the assumptions that:

1. The ARIMA model is perfectly verified, i.e., the coefficients that were estimated are the actual coefficients.
2. The innovations ϵ_t follows $N(0, \sigma^2)$, where $\sigma^2 > 0$. However, there is a problem with this assumption which can be checked in the Appendix Figure 15 where we have the QQ plot which suggests that the residuals are not Gaussian. We have also performed the Jarque-Bera and the Shapiro-Wilk test to confirm this notion, where we discard the null hypothesis of the tests which claim the normality of the residuals.

3.3 Question 8

The following figures represent the forecast and 95% confidence region for the two forecasts of differentiated series X_t. Figure 12 represents the prediction by blue dots and grey strips represent the individual 95% confidence intervals for each of the predictions. Assuming that the forecast errors are normally distributed, a 95% prediction interval for the h-step forecast is given by $\hat{X}_{T+h|T} \pm 1.96\hat{\sigma_h}$, where $\hat{\sigma_h}$ is an estimate of the standard deviation of the h-step forecast. More generally, a prediction interval can be written as $\hat{X}_{T+h|T} \pm c\hat{\sigma_h}$ where c is the coverage probability.

Figure 13 represents the bivariate 95% confidence region.

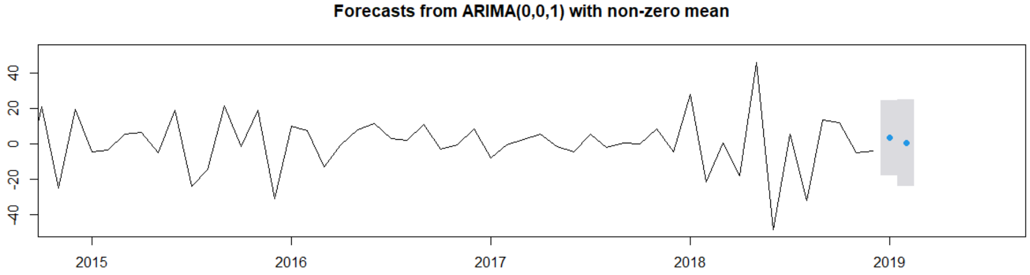


Figure 13: Time Series with Forecasts and 95% Confidence Region

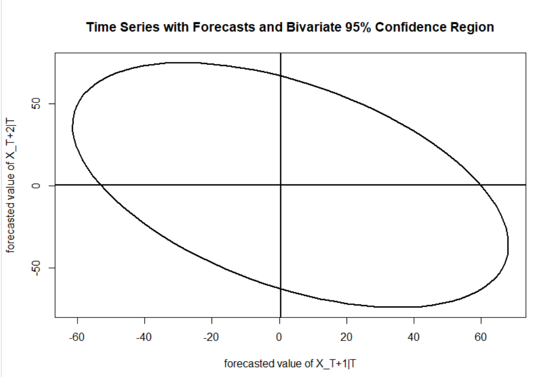


Figure 14: Time Series with Forecasts and 95% Bivariate Confidence Region

The elliptical shape of the confidence region indicates the joint distribution of the forecast errors for X_{T+1} and X_{T+2}. The larger the region, more is the uncertainty in the predicted values. The negative tilt suggests a negative correlation between X_{T+1} and X_{T+2}. The center (point of intersection of the axes) represents the point forecasts for X_{T+1} and X_{T+2} which is (3.3569545,0.4260494).

3.4 Question 9

Given a stationary time series Y_t available up to time T and the assumption that Y_{T+1} is available before X_{T+1} can help us improve the prediction for X_{T+1} iff X_t is instantaneously Granger-caused by Y_t. Hence, if we utilise the value of Y_{T+1} the prediction should be different from the one when we do not:

$$\hat{X}_{T+1|\{X_s,Y_s;s\leq T\}} \neq \hat{X}_{T+1|\{X_s,Y_s;s\leq T\}\cup Y_{T+1}}$$

This can be checked using the Granger-Causality test:

1. (a) H_0: Y_t does not instantaneously Granger-cause X_t
   (b) H_1: Y_t instantaneously Granger-causes X_t
2. Estimate the Restricted Model (without Y_{T+1}) and the unrestricted model (with Y_{T+1}). This is done by fitting the Vector Auto Regressive (VAR) model accordingly.
3. We need to test if Y_t instantaneously Granger-causes X_t, so we need to determine if any lags of Y are statistically significant in our model, hence we perform the Wald test where we estimate both the models and obtain the coefficients of the residuals and finally compute the test statistic for the Wald test.
4. In the end, we use the Wald test statistic to decide whether to reject the null hypothesis or not.

4 Appendix

4.1 Proof for Univariate 95% Confidence Interval

We know $\epsilon_t \sim N(0, \sigma^2)$ with $\sigma^2 > 0$ and we have already shown above that -

$$X_{T+1} - X_{T+1|T} = \epsilon_{T+1} \tag{3}$$

$$X_{T+2} - X_{T+2|T} = \epsilon_{T+2} - 0.5636\epsilon_{T+1} \tag{4}$$

Now, we can say that

$$X_{T+1} - X_{T+1|T} \sim N(0, \sigma^2) \tag{5}$$

$$X_{T+2} - X_{T+2|T} \sim N(0, \sigma^2(1 + 0.5636^2)) \tag{6}$$

i.e

$$\frac{1}{\sigma}X_{T+1} - X_{T+1|T} \sim N(0, 1) \tag{7}$$

$$\frac{1}{\sigma\sqrt{(1.31764496)}}X_{T+2} - X_{T+2|T} \sim N(0, 1) \tag{8}$$

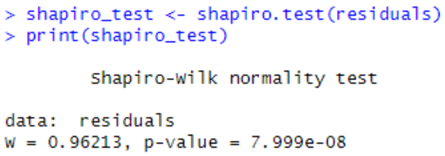
We also know that if $Z \sim N(0, 1)$ then $\mathcal{P}(|Z| \leq q_{1-\frac{\alpha}{2}}) = 1 - \alpha$ where $q_{1-\frac{\alpha}{2}}$ is the $(1 - \frac{\alpha}{2})^{th}$ quantile.

Therefore, Univariate Confidence Interval with level α is given by $[X_{T+1|T} - \sigma q_{1-\frac{\alpha}{2}}, X_{T+1|T} + \sigma q_{1-\frac{\alpha}{2}}]$ and $[X_{T+2|T} - q_{1-\frac{\alpha}{2}} * \sqrt{1.31764496} * \sigma, X_{T+2|T} + q_{1-\frac{\alpha}{2}} * \sqrt{1.31764496} * \sigma]$ respectively.

In particular, 95% Confidence Interval is given by $[X_{T+1|T} - 1.96\sigma, X_{T+1|T} + 1.96\sigma]$ and $[X_{T+2|T} - 1.96 * \sqrt{1.31764496} * \sigma, X_{T+2|T} + 1.96 * \sqrt{1.31764496} * \sigma]$ respectively.

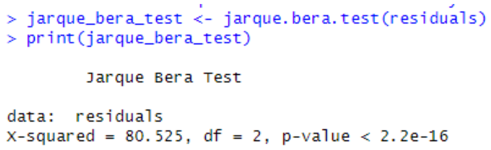
4.2 Checking Hypothesis on residuals

We want to check if our residuals are actually Gaussian or not. Therefore, we performed Shapiro-Wilk test and Jarque-Bera test for normality. We got the following results which indicate that the residuals are not normally distributed.



```
> shapiro_test <- shapiro.test(residuals)
> print(shapiro_test)

        Shapiro-Wilk normality test

data:  residuals
W = 0.96213, p-value = 7.999e-08
```

((a))



```
> jarque_bera_test <- jarque.bera.test(residuals)
> print(jarque_bera_test)

        Jarque Bera Test

data:  residuals
X-squared = 80.525, df = 2, p-value < 2.2e-16
```

((b))

Following figure represents the Q-Q plot (quantile-quantile) plot which compares the quantiles of the residuals to the quantiles of a standard normal distribution. The Q-Q plot confirms the results from the Shapiro-Wilk and Jarque-Bera tests. The residuals do not appear to be normally distributed, as evidenced by the significant deviations in the tails of the Q-Q plot.

0.5

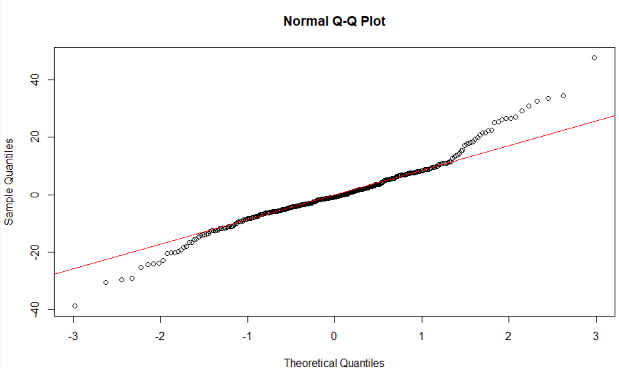


Figure 16

4.3 Code in R

```r
datafile <- "C:/Users/kanup/Downloads/data.csv"
data <- read.csv(datafile, header = FALSE, sep=",")

require(zoo) # practical and easy-to-use time series format (but larger)
require(tseries) # various functions on time series

#### Q1 ####

data[, 2] <- as.numeric(data[, 2])
df.source <- zoo(data[, 2]) # convert the second column and name it as df.source into a time series
T <- length(df.source) # calculate the length of the time series
df <- df.source[1:T] # selecting the rows for plotting and storing them in df

#### Plotting ###
plot(df, xaxt="n", main = "Time Series Data") # represents df without the x-axis
# Define the sequence of dates from January 1990 to December 2018
dates <- seq(as.Date("1990-01-01"), by = "month", length.out = T)
# Set the x-axis ticks for year-wise plotting
axis(side = 1, at = seq(1, T, by = 12), labels = format(dates[seq(1, T, by = 12)], "%Y"))

### Q2 ####
d_df <- diff(df, 1) # Differentiating to make it stationary
plot(cbind(df, d_df))
acf(d_df)

# decomposing to check trend
df_s <- ts(df, frequency = 12) # converts a column from a data frame to a simple time series object.

# Decompose the time series
decomposed_ts <- decompose(df_s)
# Extract the trend component
trend_component <- decomposed_ts$trend
# Plot the trend component
plot(trend_component, main = "Trend Component of the Time Series", ylab = "Trend", xlab = "Time")

# checking stationarity using ADF Test
install.packages('fUnitRoots')
library('fUnitRoots')
require(fUnitRoots)

```

```r
summary(lm(df ~ dates)) # to check coefficients of dates and intercept

# ADF test for the original series
adf <- adfTest(df, lag = trunc((length(df) - 1)^(1/3)), type = "ct")
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))

## Adding lags
series <- df; kmax <- 348; 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, 348, 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(df, 348, adftype = "ct")
Qtests(adf@test$lm$residuals, 48, fitdf = length(adf@test$lm$coefficients))
adf

# ADF test for differentiated series
summary(lm(d_df ~ dates[-1])) # to check coefficients of dates and intercept
adf1 <- adfTest(d_df, lag = trunc((length(d_df) - 1)^(1/3)), type = "nc")
Qtests(adf1@test$lm$residuals, 72, fitdf = length(adf1@test$lm$coefficients))

# Valid test after adding lags
adf1 <- adfTest_valid(d_df, 348, adftype = "nc")
adf1

# KPSS Test
kpss.test(df, null = ’Trend’)
kpss.test(df, null = ’Level’)
kpss.test(d_df)

##################### Q3 ############
plot(cbind(df, d_df))

############ PART-II ###################

########## Q4 ########

# study of acf and pacf
x <- d_df
par(mfrow = c(1, 2))
acf(d_df); pacf(d_df)

# choosing q = 1 and p = 3
pmax = 3; qmax = 1
```

```r

# Checking for AIC and BIC to choose the right model
mat <- matrix(NA, nrow = pmax + 1, ncol = qmax + 1) # empty matrix to fill
rownames(mat) <- paste0("p=", 0:pmax) # renames lines
colnames(mat) <- paste0("q=", 0:qmax) # renames columns
AICs <- mat # AIC matrix not filled
BICs <- mat # BIC matrix not filled
pqs <- expand.grid(0:pmax, 0:qmax) # all possible combinations of p and q
for (row in 1:dim(pqs)[1]) { # loop for each (p, q)
  p <- pqs[row, 1] # gets p
  q <- pqs[row, 2] # gets q
  estim <- try(arima(x, c(p, 0, q), include.mean = F)) # tries ARIMA estimation
  AICs[p + 1, q + 1] <- if (class(estim) == "try-error") NA else estim$aic # assigns the AIC
  BICs[p + 1, q + 1] <- if (class(estim) == "try-error") NA else BIC(estim) # assigns the BIC
}
AICs
AICs == min(AICs)
BICs
BICs == min(BICs)

# checking if well-adjusted
arima001 <- arima(d_df, c(0, 0, 1), include.mean = T)

# Checking Validity
Qtests(arima001$residuals, 24, fitdf = 3)
arima001
Box.test(arima001$residuals, lag = 2, type = "Ljung-Box", fitdf = 1)

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)$p.value
    return(c("lag" = l, "pval" = pval))
  })
  return(t(pvals))
}
Qtests(arima001$residuals, 24, 5)

# Checking Statistical Significance of coefficients
signif <- function(estim) {
  coef <- estim$coef
  se <- sqrt(diag(estim$var.coef))
  t <- coef / se
  pval <- (1 - pnorm(abs(t))) * 2
  return(rbind(coef, se, pval))
}

signif(arima001)

## function to print the tests for the ARIMA model selection
arimafit <- function(estim) {
  adjust <- round(signif(estim), 3)
  pvals <- Qtests(estim$residuals, 24, length(estim$coef) - 1)
  pvals <- matrix(apply(matrix(1:24, nrow = 6), 2, function(c) round(pvals[c,], 3)), nrow = 6)
  colnames(pvals) <- rep(c("lag", "pval"), 4)
  cat("coefficients nullity tests :\n")
  print(adjust)
  cat("\n tests of autocorrelation of the residuals : \n")
  print(pvals)
}

```

```r
arimafit(arima001)

# Plotting residuals
plot(arima001$residuals)

arima001 <- arima(d_df, c(0, 0, 1), include.mean = T)
arima001

########### PART-III #########

dates2 <- seq(as.Date("1990-01-01"), by = "month", length.out = T + 2)
future_values <- forecast(arima001, h = 2, level = 0.95)

plot(future_values, xlim = c(300, 355), xaxt = 'n')
axis(side = 1, at = seq(1, T + 2, by = 12), labels = format(dates2[seq(1, T + 2, by = 12)], "%Y"))

future_values

####### Q8 #########

# Print the future values
future_values
# confidence region

sigma2 <- arima001$sigma2
# Define the covariance matrix
Sigma <- sigma2 * matrix(c(1, -0.5636, -0.5636, 1.31764496), nrow = 2)

# Extract point forecasts and standard errors for X_{T+1} and X_{T+2}
point_forecasts <- future_values$mean[1:2]
se <- forecasted_values$se[1:2]

# Define the covariance matrix
Sigma <- sigma2 * matrix(c(1, -0.5636, -0.5636, 1.31764496), nrow = 2)

# Define the confidence level
alpha <- 0.05
chisq_val <- qchisq(1 - alpha, df = 2)

# Compute the confidence ellipse centered at the forecasted values
ellipse_points <- ellipse(Sigma, centre = point_forecasts, level = 1 - alpha, t = chisq_val)

# Plotting the results
plot(ellipse_points, type = 'l', main = "Time Series with Forecasts and Bivariate 95% Confidence Region",
     xlab = "forecasted value of X_T+1|T", ylab = "forecasted value of X_T+2|T", lwd = 2)

abline(h = point_forecasts[2], v = point_forecasts[2], lwd = 2)

################### APPENDIX ################

## Test on Residuals

# Extract the residuals from the fitted ARIMA model
residuals <- residuals(arima001)

# Plot the residuals to visually inspect them
plot(residuals, main = "Residuals of ARIMA Model", ylab = "Residuals")

# Plot the histogram of the residuals
hist(residuals, breaks = 20, main = "Histogram of Residuals", xlab = "Residuals")
```

```

# Perform the Shapiro-Wilk test for normality
shapiro_test <- shapiro.test(residuals)
print(shapiro_test)

# Perform the Jarque-Bera test for normality
jarque_bera_test <- jarque.bera.test(residuals)
print(jarque_bera_test)

# Plot a Q-Q plot to visually check for normality
qqnorm(residuals)
qqline(residuals, col = "red")
```

5 References

RPUBS - Time series Analysis in R - Decomposing Time Series. (n.d.).

3.5 Prediction intervals | Forecasting: Principles and Practice (2nd ed). (n.d.).

Franq, C., & Wandji, J. N. (n.d.). Linear Time Series [Slide show].