**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE**

Assignment prepared in partial fulfillment of the **course MATH F432** (**Applied Statistical Methods**)

|  |  |
| --- | --- |
| **Name** | RISHAV DATTA |
| **ID No.** | 2018A3PS0910H |
| **Assignment Number** | 8 (OPEC Oil Prices) |
| **Course Code** | Math F432 |

**Contents:**

|  |  |
| --- | --- |
| Glossary | 1-2 |
| Introduction | 3 |
| Analysis | 4-15 |
| Conclusion | 16-17 |

Glossary

**Time Series:** Time Series forecasting is an important area of machine learning. A time series is a sequence of observations taken sequentially in time. **AR**, **MA**, **ARMA**, and **ARIMA** models are used to forecast the observation at (t+1) based on the historical data of previous time spots recorded for the same observation. However, it is necessary to make sure that the time series is stationary over the historical data of observation overtime period. If the time series is not stationary then we could apply the differencing factor on the records and see if the graph of the time series is a stationary overtime period.  
**Examples of Time Series:** Sales Trend, Oil Price etc.  
  
**White Noise:** A series which is purely random in nature is called White Noise. Average is the best forecast we can make for such series.  
  
**Autoregressive (AR) Model:** Yt depends only on past values **Yt = a + b1Yt-1 + b2yt-2 + ... + bpYt-p**

**Moving Average (MA) Model:** Yt depends random error terms  
**Yt = a + et + c1et-1 + c2et-2 + ... + cpet-p**

**ARMA Model:** Combines AR and MA   
**Yt = a + b1Yt-1 + b2yt-2 + ... + bpYt-p + et + c1et-1 + c2et-2 + ... + cpet-p**

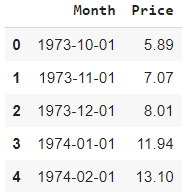
**Stationarity of Time Series:** A series is said to be **strictly stationary**, if the mean, variance and covariance is time variant. A series which is not stationary can be made stationary after **differencing**. After differencing once, series is called **integrated of order 1**.Most models assume data is stationary. Standard techniques are invalid if data is non-stationary.

**ARIMA Model:** Auto-Regressive Integrated Moving average. It combines all three methods.

**Autocorrelation:** It is the similarity between observations as a function of time lag between them.

**Introduction**

For this assignment I have chosen Project number 8. In this project we were given data on price of crude oil per barrel (in $) OPEC countries from 1973 to 2019(month wise). Below is a given sample of the data:

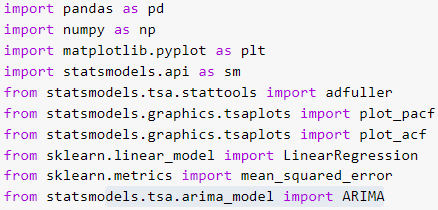


**Figure 1: First 5 elements of the data**

This is a classic example of a time series problem. In these types of problems we have to predict some future value based on the past values of the data. These examples include stock prices, Crude oil prices etc.  
So for this project I have chosen to apply various time series analysis. These models have been described in the following sections.   
The complete study and analysis of the oil prices has been done in python **(Google Colab** has been used for this purpose**)**. All the python notebooks have been uploaded alongside this report.

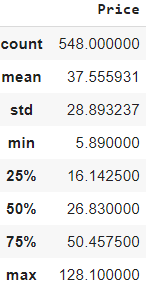
Analysis of Data

Firstly I imported the following modules in python:



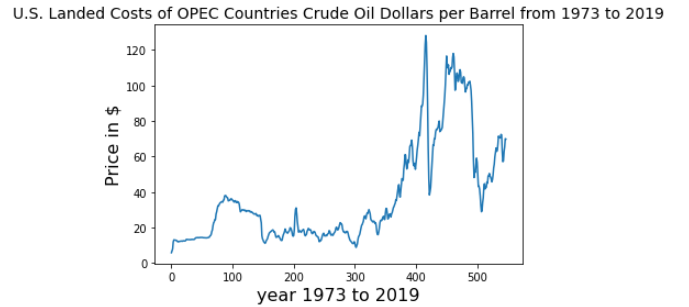
**Figure 2: List of modules**

Let us provide some details regarding the data:



**Figure 3: Some details regarding the Price of oil barrel**

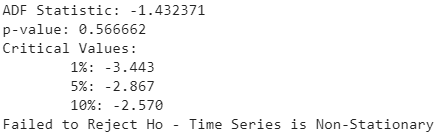
Now let us plot the Price of OPEC Countries Crude oil per barrel from 1973 to 2019.



**Figure 4**

**Stationary test of the Data:**

Our next task is to find out if the data is stationary or non-stationary. It will be important as we will use this result to predict which is the best model we can code for this data.  
Even before conducting tests we can comment on this just by looking at the plot in figure 4. In the plot we observe that the mean of the data is time variant. So it must be non-stationary.  
  
Augmented Dickey Fuller test (ADF Test) is a common statistical test used to test whether a given Time series is stationary or not. It is one of the most commonly used statistical tests when it comes to analyzing the stationary of a series. The **statsmodel** package provides a reliable implementation of the ADF test via the **adfuller()** function in Python. Given below is the result of ADF test:



**Figure 5: Result of ADF test**

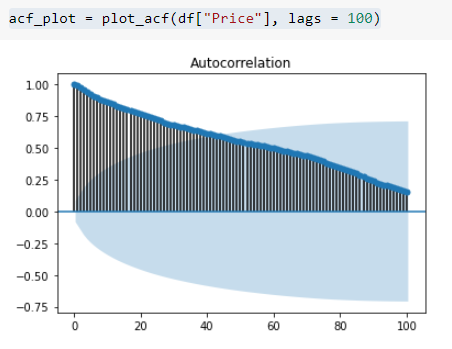
From the result of the test we conclude that the data is non-stationary. To convert a stationary data into non-stationary data we use the technique of differencing.  
As we have to use differencing AR, MA and ARMA models won’t give satisfactory results. So the ARIMA model will give us the best results. But for the sake of this project I have analyzed each of the above mentioned models.  
  
**AutoRegression (AR) Model:**

An autoregressive (AR) model **predicts future behavior based on past behavior**. It’s used for forecasting when there is some correlation between values in a time series and the values that precede and succeed them.

**The AR(p) model is defined by the equation:**  
**yt = δ + φ1yt-1 + φ2yt-2 + … + φpyt-1 + At**  
Where:

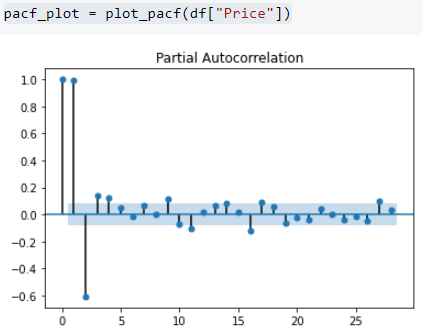
* yt-1, yt-2…yt-p are the past series values (lags),
* At is white noise (i.e. randomness),

In this study we have considered white noise as 0.  
  
Let us plot Autocorrelation and Partial AutoCorrelation plot.



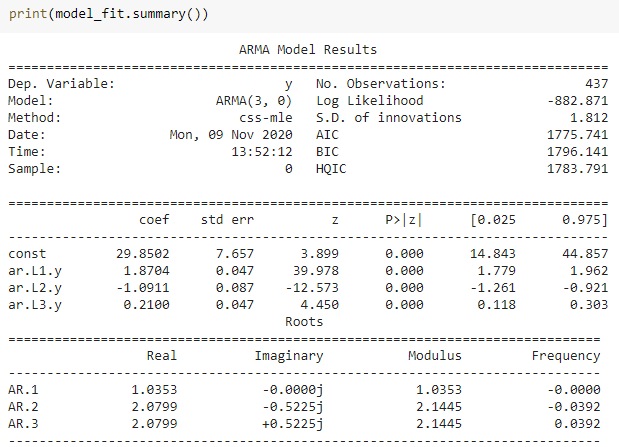
**Figure 6: AutoCorrelation Plot**

An autocorrelation plot is designed to show whether the elements of a time series are positively correlated, negatively correlated, or independent of each other.



**Figure 7: Partial AutoCorrelation Plot**

**There are significant spikes up to lags 1, 2 and 3. Based on the PACF Plot we should start with an Auto Regressive model with lags 1, 2, 3. The code will be uploaded and here we discuss the results and analyze it.**



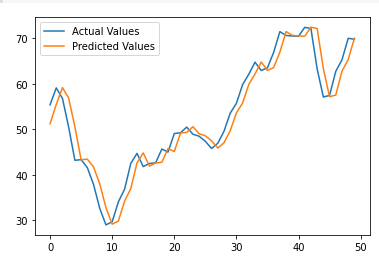
**Figure 8: ARMA model results**

Let us analyze the results. Firstly we see the model name is **ARMA(3,0)**. **3** indicates number of lags we found from **PACF** plot and **0** is the coefficient of **MA**. **MA=0** means that it is an **AR** model. In figure 8 we can also observe the P>|z| values. From the table we see that it is zero for all the lags and the constant. So all of them are quite significant and can’t be ignored.

Therefore the model is,

**Yt = 29.8502 + 1.8704Yt-1 -1.0911Yt-2 + 0.21Yt-3**

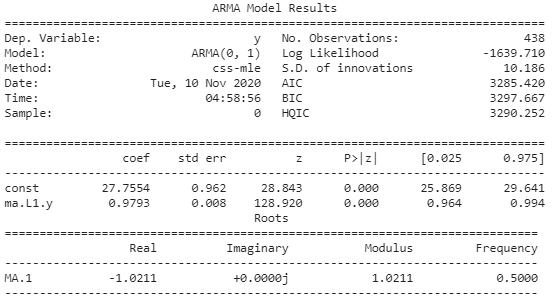
**Akaike Information Criterion** value is observed to be 1775.741.



**Figure 9: Actual vs. Predicted Values**

**Moving Average (MA) Model:**Moving averages can smooth time series data, reveal underlying trends, and identify components for use in statistical modeling.  
A **moving average** term in a time series model is a past error (multiplied by a coefficient).

The code of MA model and AR model is exactly the same except one part. In this model we will write ARMA(0,1) : 0 is for AR model and 1 is for the error term in MA model.



**Figure 10: Results for MA =1 test.**

Let us analyze the results. Firstly we see the model name is **ARMA(0,1)**. **0** indicates number of lags we found from **PACF** plot and **1** is the coefficient of **MA**. **MA=1** means that it is an **MA** model. In figure 10 we can also observe the P>|z| values. From the table we see that it is zero for all the terms. So all of them are quite significant and can’t be ignored.

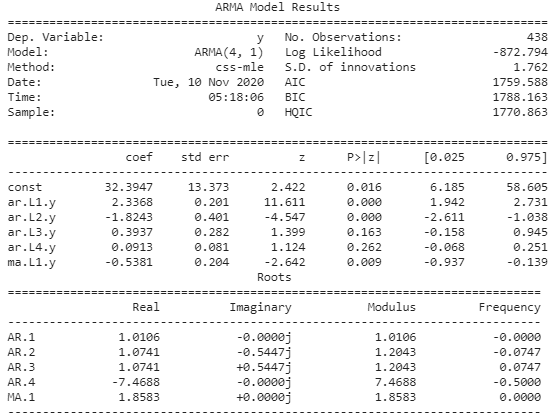
The model is, **Yt = 27.7554 + 0.9793et   
Akaike Information Criterion is 3285.420**This extremely large value shows that the data is very unsuitable for a pure Moving Averages model.

**Auto-Regressive Moving Averages Model:**An ARMA model, or Autoregressive Moving Average model, is used to describe weakly [stationary](https://www.statisticshowto.com/stationarity/)[stochastic](https://www.statisticshowto.com/stochastic-model/)[time series](https://www.statisticshowto.com/timeplot/) in terms of two [polynomials](https://calculushowto.com/polynomial-function-degrees/). The first of these polynomials is for [auto regression](https://www.statisticshowto.com/autoregressive-model/), the second for the [moving average](https://www.statisticshowto.com/moving-average/).

Often this model is referred to as the **ARMA(p,q) model**; where:

* p is the order of the autoregressive polynomial,
* q is the order of the moving average polynomial.

ARMA model is used only for stationary data. Hence we can’t expect good results from this test.



**Figure 11: ARMA(4,1) Results**

Let us analyze the results. Firstly we see the model name is **ARMA(4,1)**. **4** indicates number of lags we found from **PACF** plot and **1** is the coefficient of **MA**. In figure 10 we can also observe the P>|z| values. From the table we see lag 3 and lag 4 values can be ignored. So the model is,

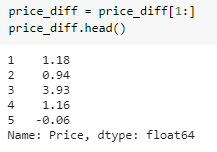
**Yt = 32.3947+2.3368Yt-1 -2.3368Yt-2 -0.5381e1**

**Akaike Information Criterion = 1759.588**

**Auto-Regressive Integrated Moving Average (ARIMA) Model:**ARIMA model is a generalization of the ARMA Model. ARIMA Model is applied whenever there is some evidence of non-stationarity of data. The initial differencing step can be applied once or more than once to remove the non-stationarity.

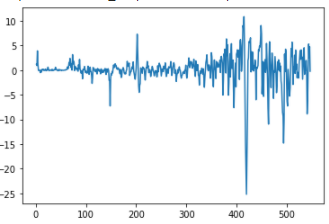
* AR indicates that the variable of interest is regressed on its lagged values.
* MA indicates regression error is a linear combination of past error terms.
* I indicates that the data values have been replaced with the difference between their values and previous values.

From figure 4 we have inferred that the data provided is non-stationary. Hence the AR,MA and ARMA models which we did gave poor results. **That is the reason we are applying ARIMA model.**First let us apply **differencing of order 1.**



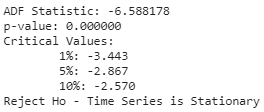
**Figure 12: Data after applying differencing once**

Now let us check again if the data is stationary or not.



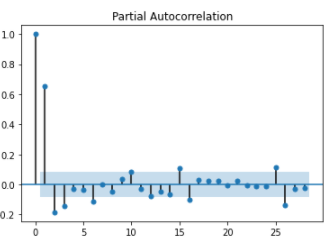
**Figure 13: Price(after differencing once) vs Time**

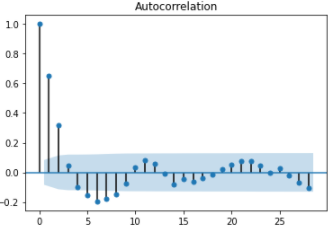
From the plot itself we can clearly see that mean and variance has become constant. Let us use the Augmented Dickey Fuller Test once again to prove that the data is indeed stationary.



**Figure 14: Result of the ADF test**

Hence the data is stationary after differencing once. Now, we plot the new ACF and PACF Plot.

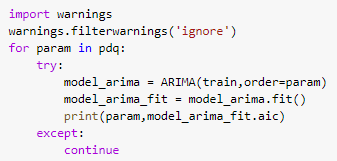




**Figure 16: New PACF Plot**

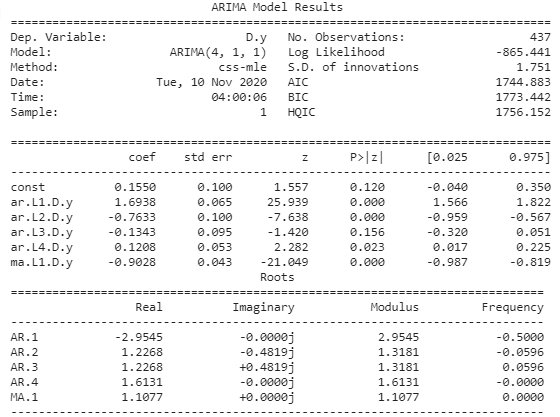
**Figure 15: New ACF Plot**

AR values will be obtained from PACF Plot and the MA values from ACF plot. Rather than doing it manually I have written a code to optimize the best possible AR and MA values. The value of ‘I’ is 1 as we applied differencing once.



**Figure 17: Code snippet to optimize AR and MA values.**

After optimizing it we get the best results for AR=4 and MA = 1. Therefore, our model is ARIMA(4,1,1).



**Figure 18: ARIMA Model Results**

Let us analyze the results. Firstly we see the model name is **ARIMA (4,1,1)**. **4** indicates number of lags we found from **PACF** plot and **1** is the coefficient of **MA**. In figure 8 we can also observe the P>|z| values. From the table we see that it is less than 0.05 for lags 1, 2, 4 and error 1. For the constant value and lag 3 P>|z| value is much greater than 0.05. So we neglect them.

**Yt = 1.6938Yt-1 -0.7633Yt-2 + 0.1208Yt-4 -0.9028e1**

**Akaike Information Criterion = 1744.883**

Conclusion

In this project we had to deal with non-stationary data. In order to make the data stationary we must make it stationary by using differencing. Because of this reason ARIMA model works better   
than AR, MA and ARMA model. We come to this conclusion by using Akaike Information Criterion (AIC). In this table we provide a sample of some models and their AIC values. Lower the AIC values better will be the model.

|  |  |  |
| --- | --- | --- |
| **Model Name** | **Order** | **AIC** |
| AR | (3,0) | 1775.741 |
| MA | (0,1) | 3285.420 |
| ARMA | (4,1) | 1759.588 |
| ARIMA | (4,1,1) | 1744.883 |

In the code provided, I have considered many other cases.

**Order AIC**

(0, 0, 0) 3859.8099896502054

(0, 0, 1) 3285.4201014383584

(0, 1, 0) 2112.9723861549996

(0, 1, 1) 1881.581824406106

(0, 1, 3) 1771.2450079569073

(0, 1, 4) 1767.9882954715517

(0, 2, 0) 1854.1315776705899

(0, 2, 1) 1854.6854028086836

(0, 2, 2) 1856.406921183022

(0, 2, 3) 1790.0112444110819

(0, 2, 4) 1774.091659226898

(1, 0, 0) 2125.38331002219

(1, 0, 1) 1892.729296889819

(1, 0, 2) 1796.2374534750695

(1, 0, 3) 1779.162228143302

(1, 0, 4) 1774.4303291393535

(1, 1, 0) 1794.824250996653

(1, 1, 1) 1782.3800687315493

(1, 1, 2) 1767.4244380323498

(1, 1, 3) 1768.0693610582537

(1, 2, 0) 1854.6549599668529

(1, 2, 1) 1797.1213535427946

(1, 2, 2) 1784.8601259512625

(1, 2, 3) 1770.033162532765

(1, 2, 4) 1770.6860304111387

(2, 0, 0) 1796.344361285764

(2, 0, 1) 1786.3442791132697

(2, 0, 2) 1772.8800389183955

(2, 0, 3) 1773.7028600995654

(2, 0, 4) 1785.5925217781764

(2, 1, 0) 1773.3641674629991

(2, 1, 1) 1746.13404920525

(2, 1, 2) 1748.0411536320976

(2, 2, 0) 1856.6549018955438

(2, 2, 1) 1776.035495468692

(2, 2, 2) 1749.5832766704073

(2, 2, 3) 1751.4894777841978

(2, 2, 4) 1769.329217998131

(3, 0, 0) 1779.3762650508486

(3, 0, 1) 1758.2026324287976

(3, 0, 2) 1759.8797912486546

(3, 1, 0) 1753.0268414814802

(3, 1, 1) 1747.9818250944588

(3, 1, 2) 1749.7289086086184

(3, 2, 0) 1847.484212245935

(3, 2, 1) 1755.9881249415155

(3, 2, 2) 1751.4200979330287

(3, 2, 3) 1751.6324304865657

(3, 2, 4) 1753.61844016358

(4, 0, 0) 1761.8143119823335

(4, 0, 1) 1759.5875965576652

(4, 0, 2) 1761.0281995231326

(4, 1, 0) 1752.1711079857464

(4, 1, 1) 1744.8825799589667

(4, 1, 2) 1746.8635676398455

(4, 1, 3) 1745.4403526774029

(4, 2, 0) 1838.993060429121

(4, 2, 1) 1755.2205621960395

(4, 2, 2) 1748.077470801029

(4, 2, 3) 1753.3328445955453

(4, 2, 4) 1748.8569222504566