ARIMA Model (Auto-Regressive Integrated Moving Average Model)

An ARMA (p,q) process combines both AR and MA terms and is defined succinctly using the Backward operator by the equation ***[1]***



Equation can also be written as ***[1]***:

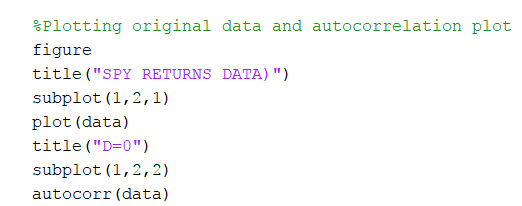


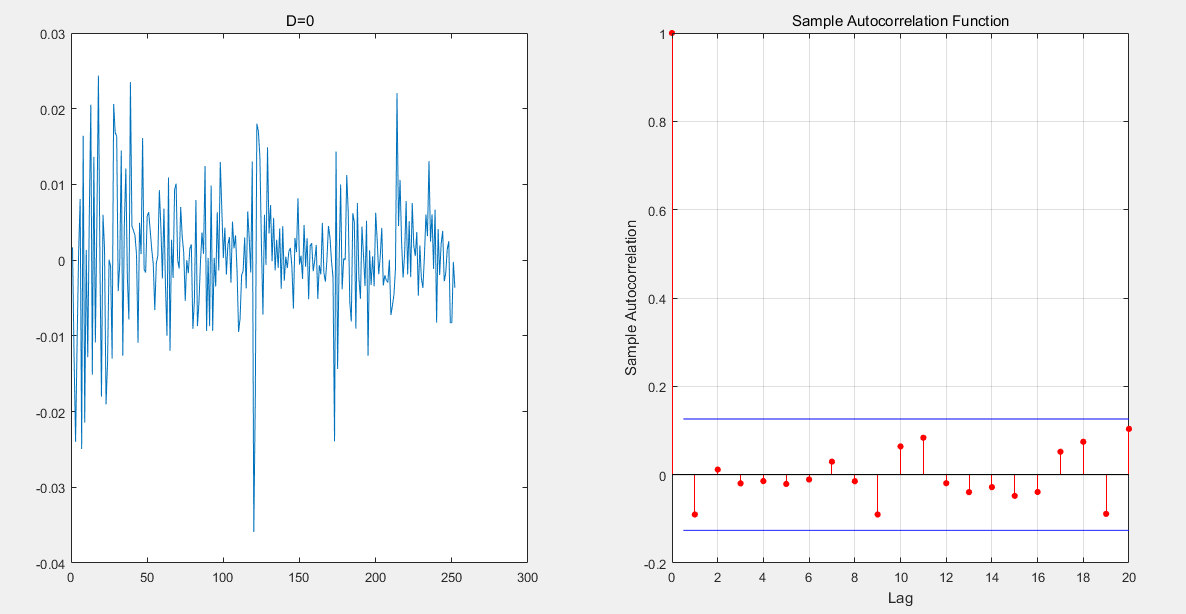
RANDOM WALK (when p = 0 , q = 0)



Our first time series is SPY ETF Returns for the financial year 2016, which we used in our RETURNS Project.

The following code plots original data and its auto correlation function for both time series

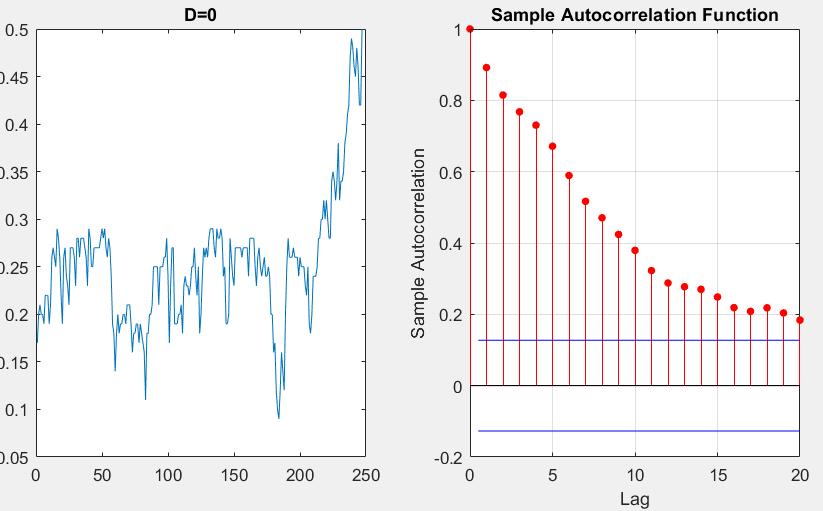


For SPY Returns, data is stationary and following figure is the plot and autocorrelation function of SPY returns

**SPY ETF RETURNS - 2016 ORIGINAL DATA PLOTS**

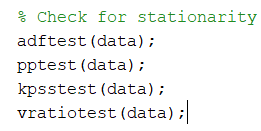
Our second financial time series is the US Treasury 1 MONTH interest rate time series.

For US Treasury data, data is not stationary which is evident from the following plot and auto correlation plot.



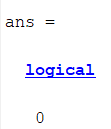
**US TREASURY 1 MONTH - ORIGINAL DATA PLOTS**

This is verified by passing data to the following tests:



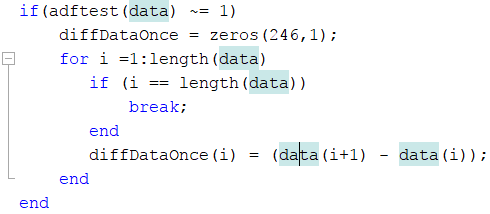
ADFTest is Augmented Dickey Fuller test and if it returns logical ‘0’, then data is not-stationary.

Following is the result on US Treasury 1 MONTH Interest rate data:

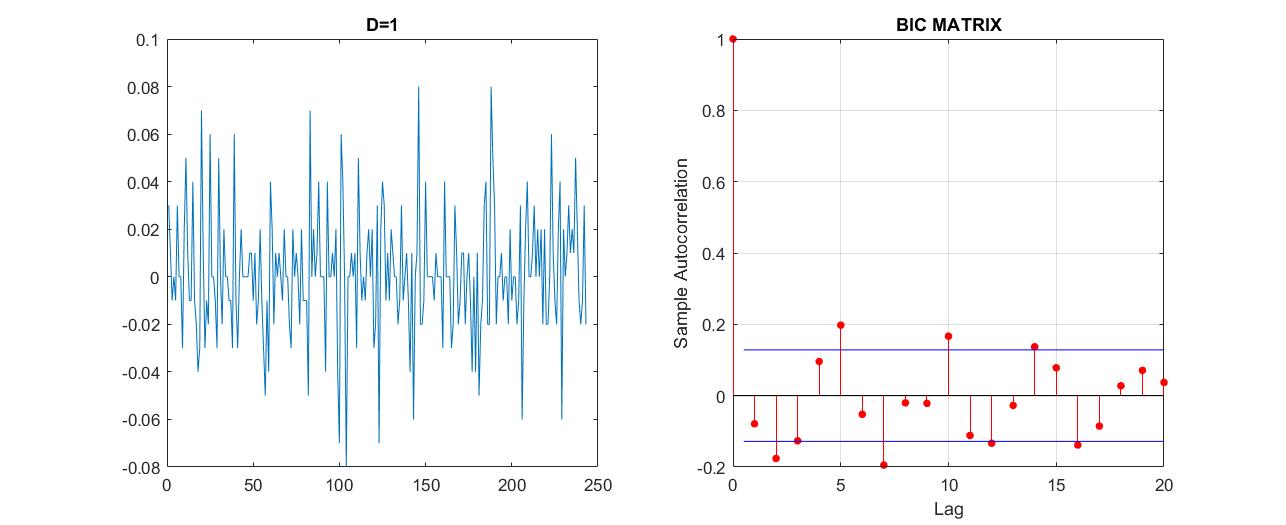


Therefore we make the data stationary by differencing and obtaining 





Following is the plot after differentiating data once and data appears stationary and passes the adftest also, which returns logical ‘1’.

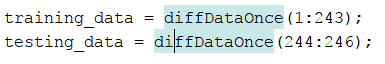


**US TREASURY 1 MONTH TRANSFORMED DATA PLOTS**

Then we split the data into training and testing data set. We remove the last 3 points from differentiated data for US Treasuries and last 3 data points from SPY ETF Returns data set. If we fit the model to entire dataset and make predictions on the latest data set, the model will be biased as it already knows the last 3 data points.

For SPY ETF RETURNS dataset, we have

252 data points, so we split : For US TREASURY 1 MONTH Interest Rates we have 246 data points, so we split:

Next, we figure out best models for our both data set using the following snapshot of code.

We iterate over 2 for loops to identify the appropriate lags *p & q*for our ARIMA Model and select the model with lowest BIC (BAYESIAN INFORMATION CRITERION) and AIC (AKAIKE INFORMAION CRITERION).

AIC (AKAIKE’s INFORMATION CRITERION)

-2\*log(L) + 2\*(p+q)

BIC (BAYESIAN Information CRITERION)

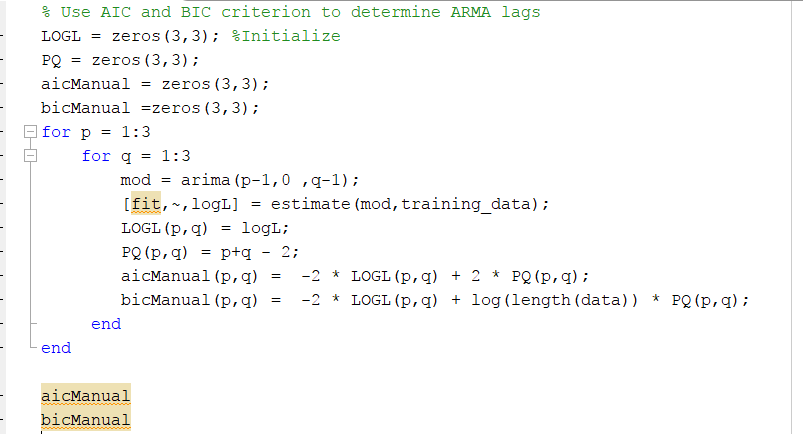
-2\*log(L) + log(n)(p+q)

The term 2(p+q) in AIC or log(n) in BIC is penalty on having too many parameters (lack of parsimony) . AIC tends to choose models with more parameters than BIC as log(n) > (p+q) ***[1]***. And n >= 8 for both our datasets

In case of SPY ETF returns our *d=0.* In case of US Treasury 1 MONTH interest rates *d=1*

But to the model we pass *d=0*  as we are fitting the model to differentialted data for US Treasury 1 MONTH interest rates.

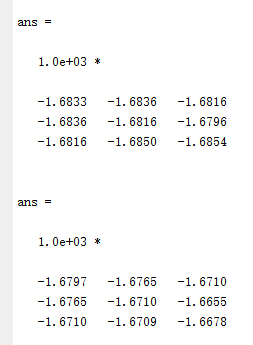
Aicmanual and bicmanual array calculates the aic and bic using the formulas.



The *estimate()*  function uses the maximum likelihood to estimate the parameters of the Arima(p,D,q) given the the observed univariate time series. *[2]*

So for each data-set we get 9 models for different lags *{p, q}* in {0,1,2}.

For SPY ETF Returns data, AIC/BIC Matrix for all 9 models is as followed. First Matrix is AIC and second matrix is BIC. The smallest value in AIC Matrix is at (2,2) index ( - 1.6854\*1.0e+03) and for BIC is (0,0) -1.6797 \*1.0e+03. As mentioned above, AIC selects models with more parameters than BIC.

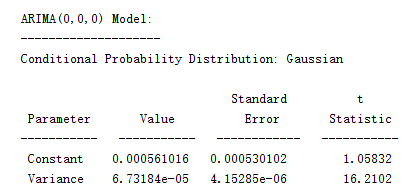


1. **AIC 2) BIC Matrix - SPY ETF**

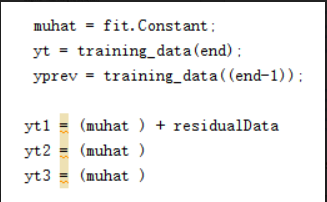
So our best model is ARIMA(0,0,0). Our lags *p=0, q=0* mean that our process has no correlation with past data and does not include residual information for more than 1 lag.

ARIMA (0,0,0) = RANDOM WALK.

We do not have any AR, MA coefficients and just have estimated mean which is referred as constant by the model.

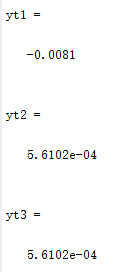


Once we have ARIMA (0,0,0) model fit to data, we make predictions using the coefficients obtained from training\_data.



**FORECASTING ARIMA (0,0,0)**

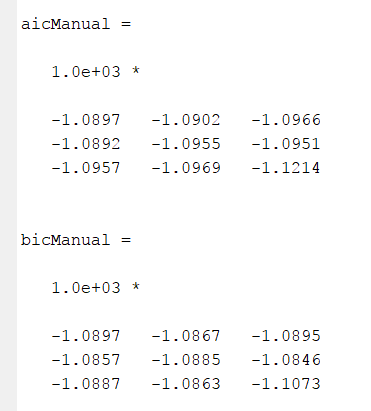
Our predicted values for testing data using the above equations are as follows:

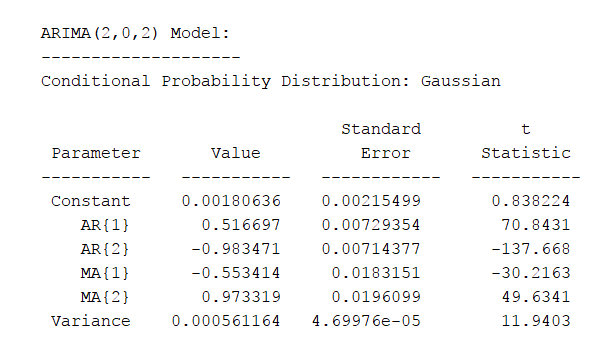
**Forecasted Values Actual Data**

The forecasted values aren’t as accurate to the testing data. yt1 = y\_t+1 is equivalent to our 1st testing data point. The other two data points are

For US Treasury 1 MONTH Interest rates, we find the AIC/BIC Matrix for all 9 models. The AIC and BIC Error metric give the same results. They have lowest values of -1.214 \* 1.0e+03 (aic) and1.1073\*1.0e+03 (bic)



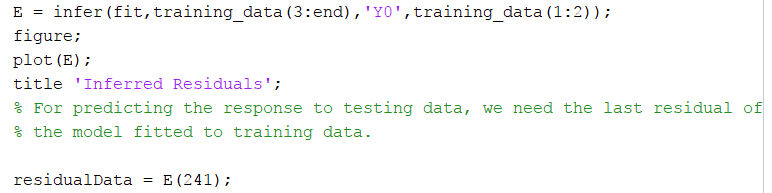
Our best model by both AIC and BIC criterion is ARIMA (2,1,2). Following is the picture of model coefficients from matlab command window:

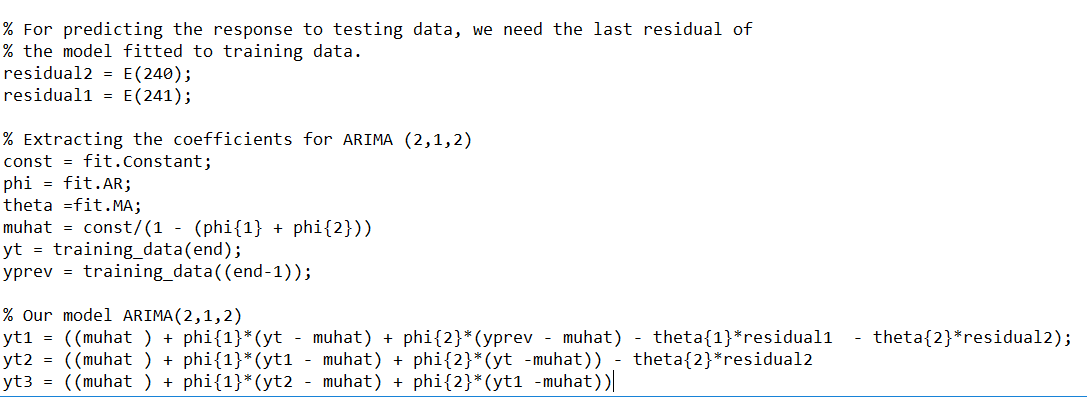


**ARIMA (2,1,2) - US TREASURY 1 MONTH Interest Rate**

The data given to the model is differentiated once already and hence we are passing d=0 as parameter to *arima (p-1, 0, q-1)* function.

The arima function fits the model to training\_data and estimates y\_t for all data points. The infer function calculates the residuals and we extract the last residual as it is required in predicting the next data point.





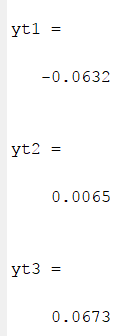
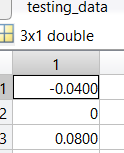
For ARIMA(2,1,2) we require AR(1), AR(2), MA(1) and MA(2) coefficients, which we get from the model. Using these coefficients, we have developed above equations for predicting testing\_data.



As we are predicting we do not have the residual at time (t+1), but we have previous two residuals for time (t) and (t-1).

Similarly, for (t+2), we do not have residuals for (t+2) and (t+1), we just have it for time t.

Lastly, for time (t+3), we have no residuals, so it just behaves like AR(2) model

**Forecasted Data Actual Data**

So we can see that for ARIMA (2,1,2), our model predicts approximately all data points near to the actual data.

**Conclusion:**

Our prediction for SPY returns is accurate for first data point using ARIMA (0,0,0)/RANDOM WALK Model.

Our prediction for US TREASURY 1 MONTH Interest rates is modelled using ARIMA (2,1,2) is approximately near to the exact data points.