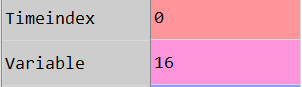
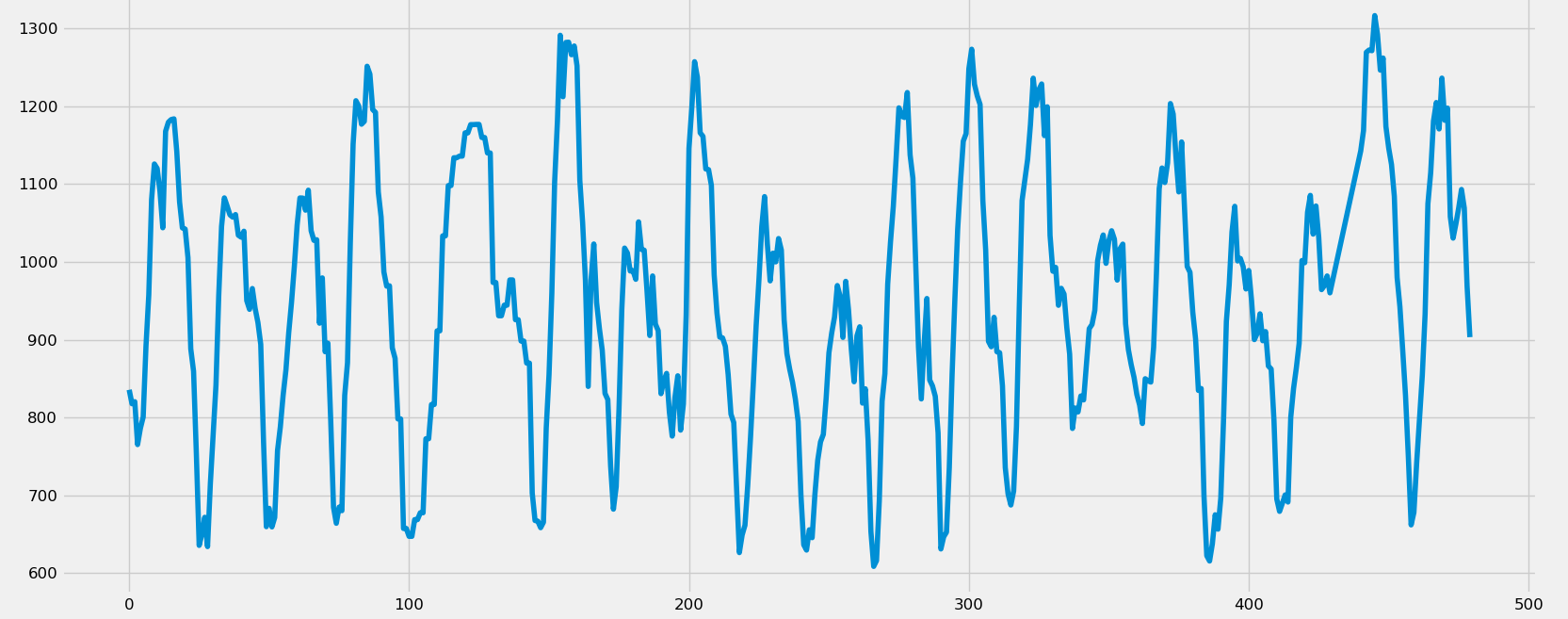
**Time Series Analysis**

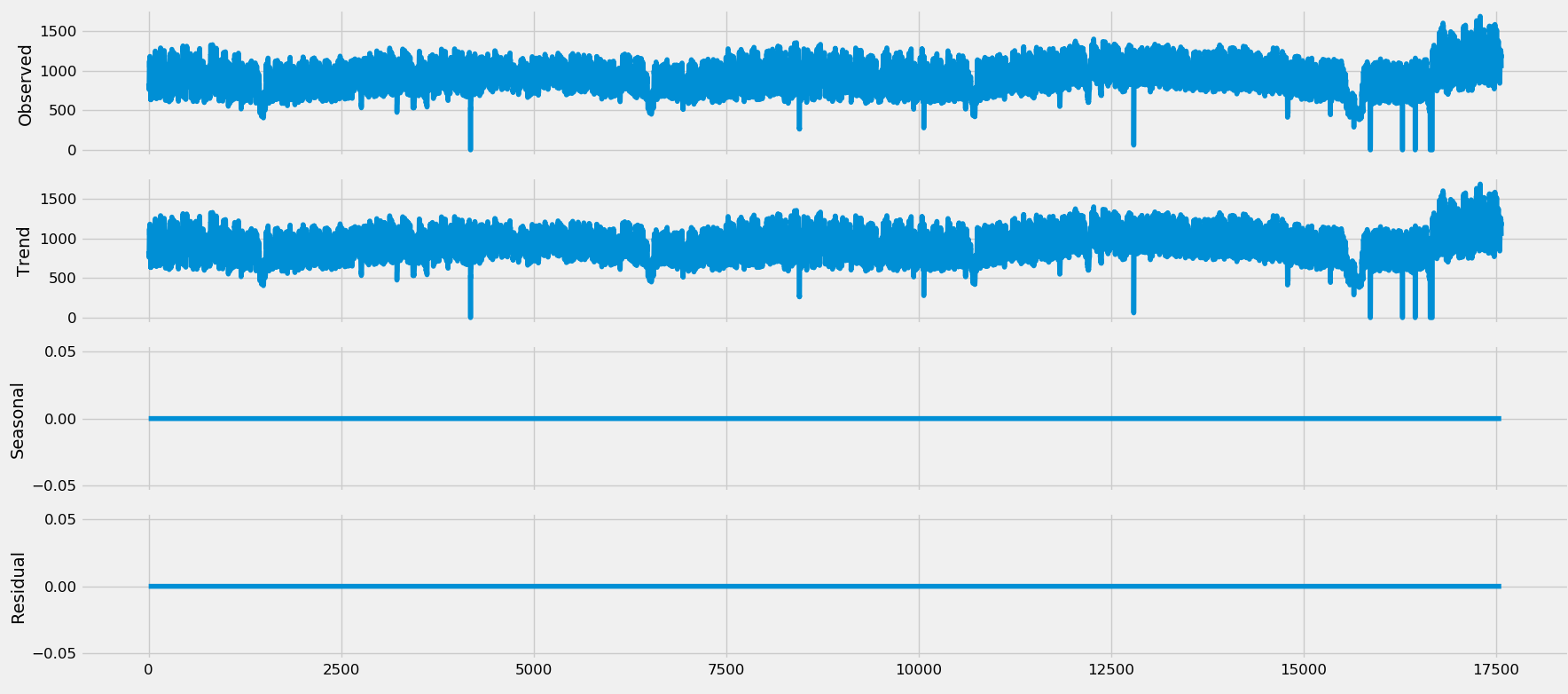
These are the steps followed in the analysis:

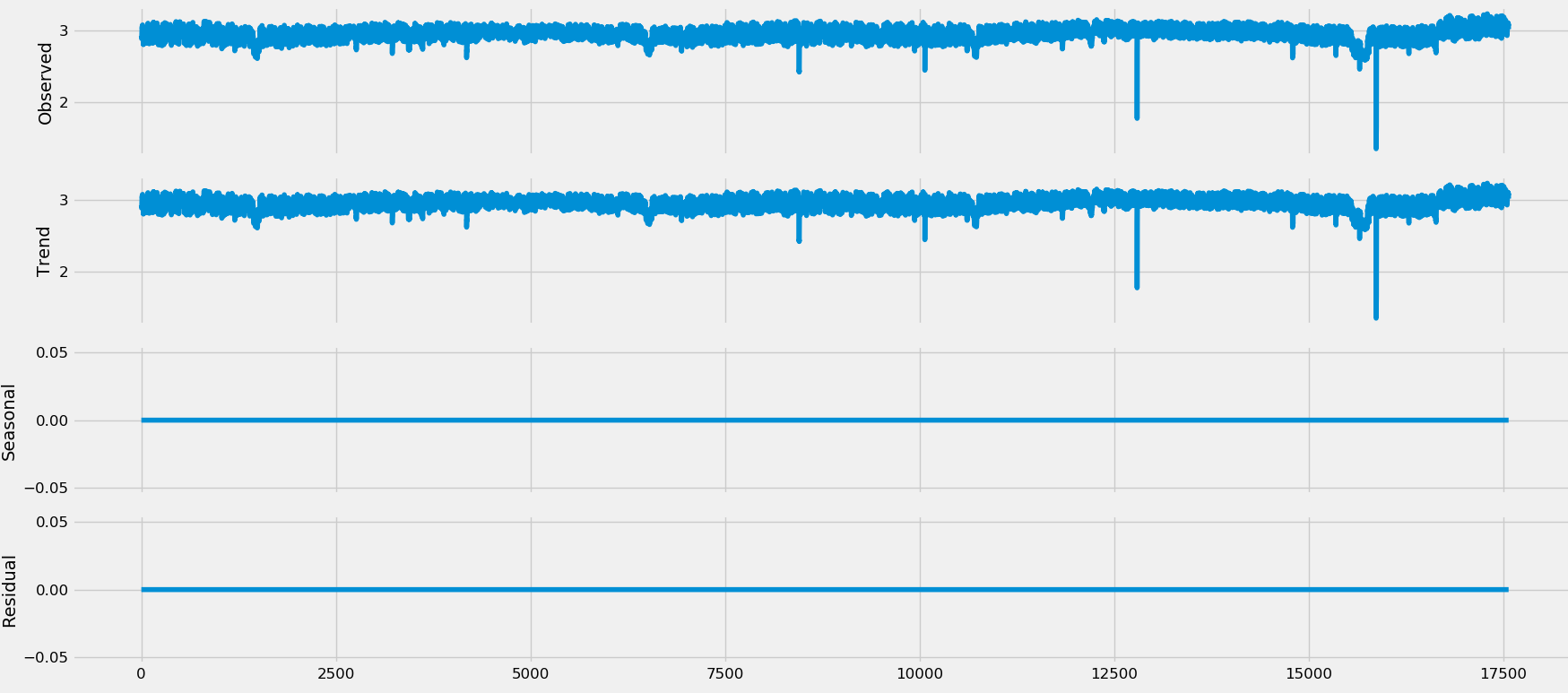
1. The python code to be run is “time\_series\_analysis\_code\_3”. You may have to install the necessary python libraries.
2. Create a time index as the hours elapsed starting from “0” as an integer since we have hourly data.
3. Checking min and max of time index: (0, 17567)
4. Took the log of “Variable” column to transform the data. This to check if it will be better to fit the model on the transformed data. It is also helpful if the original data has “multiplicative” seasonality.
5. There are missing values in the “Variable” column as shown below. We used linear interpolation to fill in the missing values.



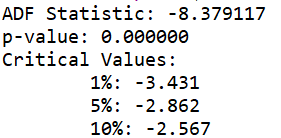
1. From visual inspection at granular level, the time series appears to be cyclic. There is no regular seasonality that is apparent in this time series.



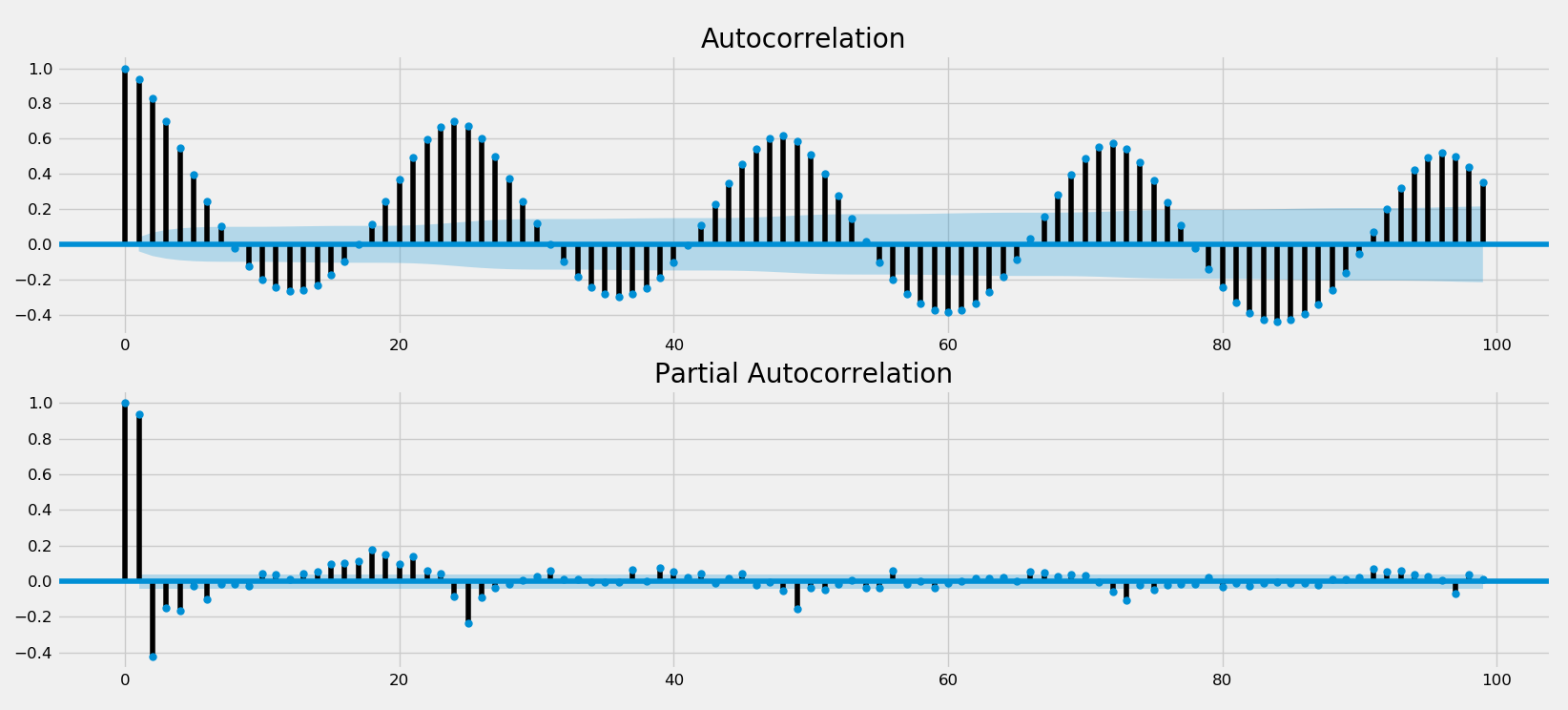
1. To be sure, we decomposed the original series and the log transformed. We didn’t see any apparent seasonality.
2. 



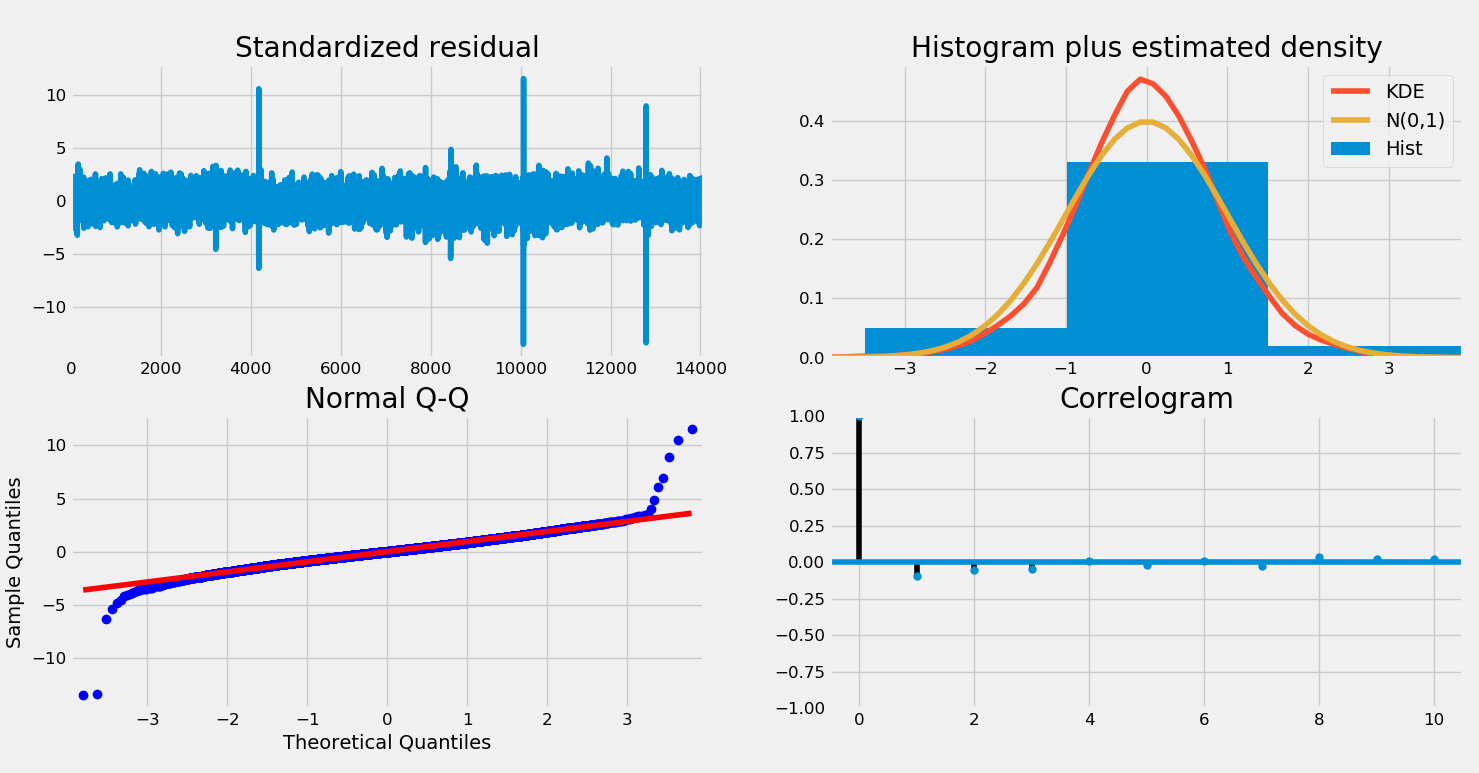
1. Check for stationarity using Augmented Dicky Fuller Test. We can see that the calculated ADF statistic is less than critical values, rejecting the null hypothesis which means that the process has no unit root, and in turn that the time series is stationary or does not have time-dependent structure. This means we don’t have to difference the series to make it stationary. That means d = 0 for ARIMA model.



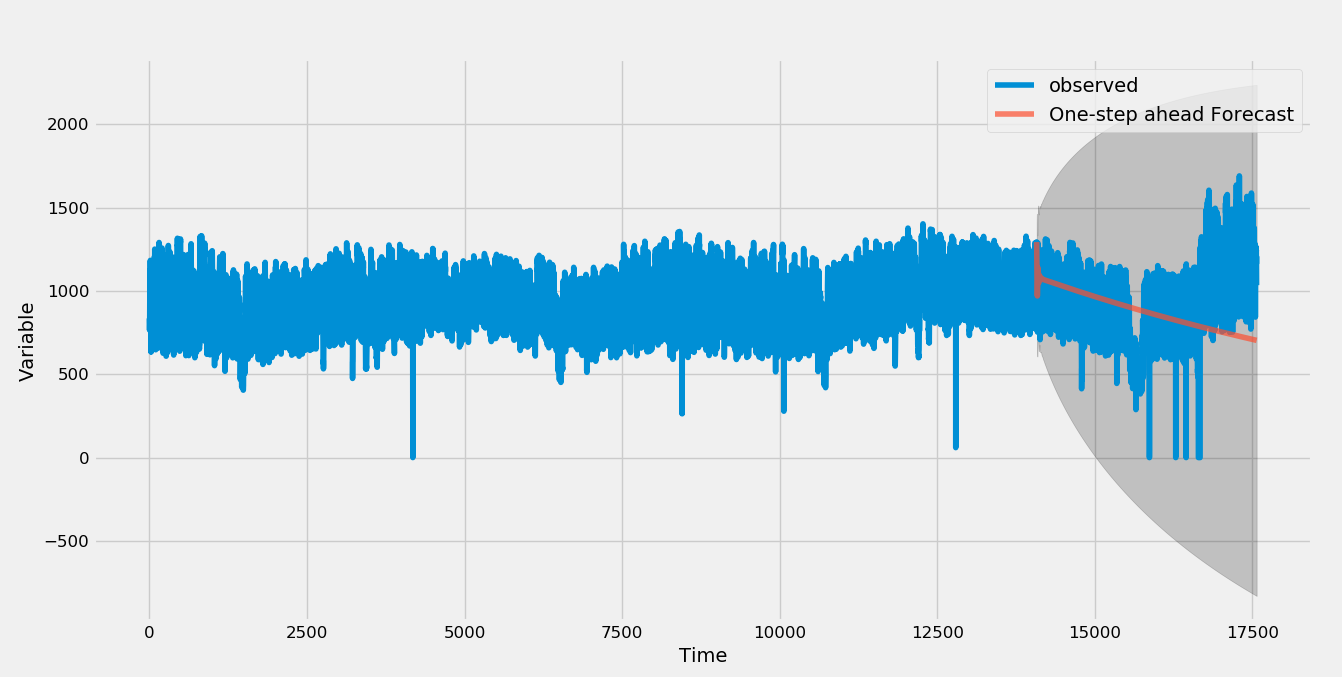
1. In order to determin p (AR) and q (MA) values, we plot the autocorrelation (“ACF) and partial autocorrelation (“PACF”) graphs for the original series as shown below. We can see that “ACF” is decaying geometrically and shows a sinusoidal trend which is indicative of an AR-2 process. We can also see that first two lags of PACF are prominent which indicates AR-2 process. We also see that PACF has still some significant values at higher lags which indicates some MA components.



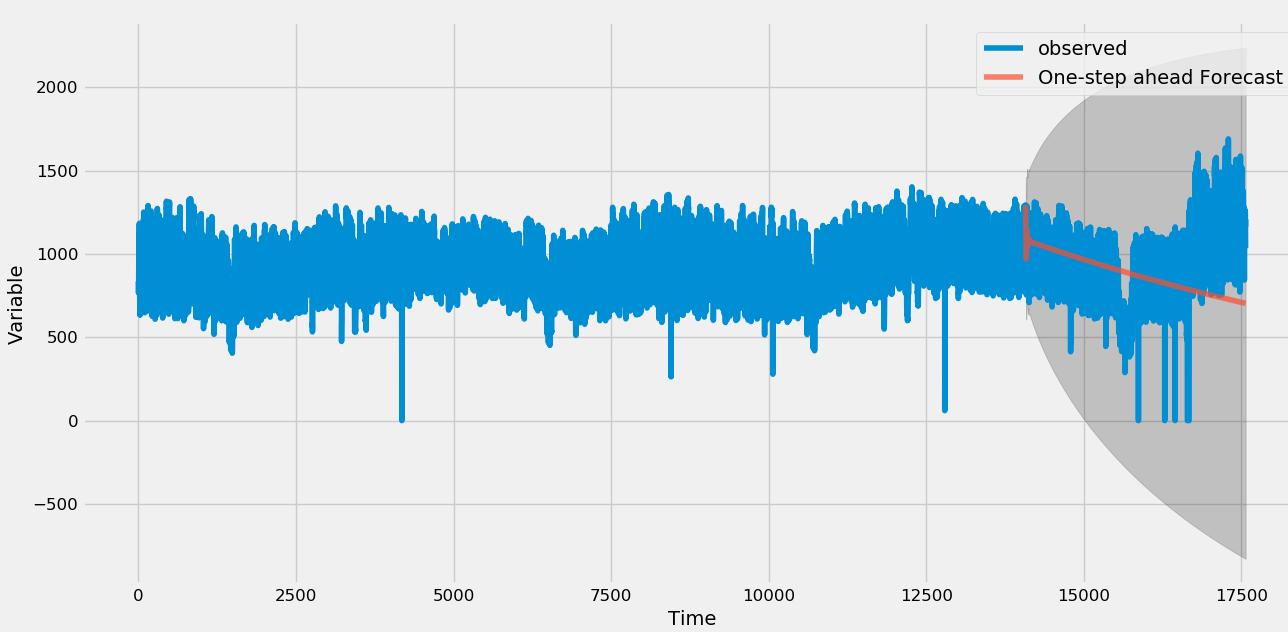
1. We will do a grid search of p and q values and check the RMSE to determine which values give the least error. A better statistic is AIC which is recommended if possible.
2. After, the grid search we found that that p = 4 and q =3 gave the least RMSE so that’s what we will use for our ARIMA model.
3. We split the data for 80% training and 20% for testing. The testing data is the most recent data.
4. The performance of the fitted ARIMA model on training data is shown below. We can see that the errors are normally distributed and uncorrelated.



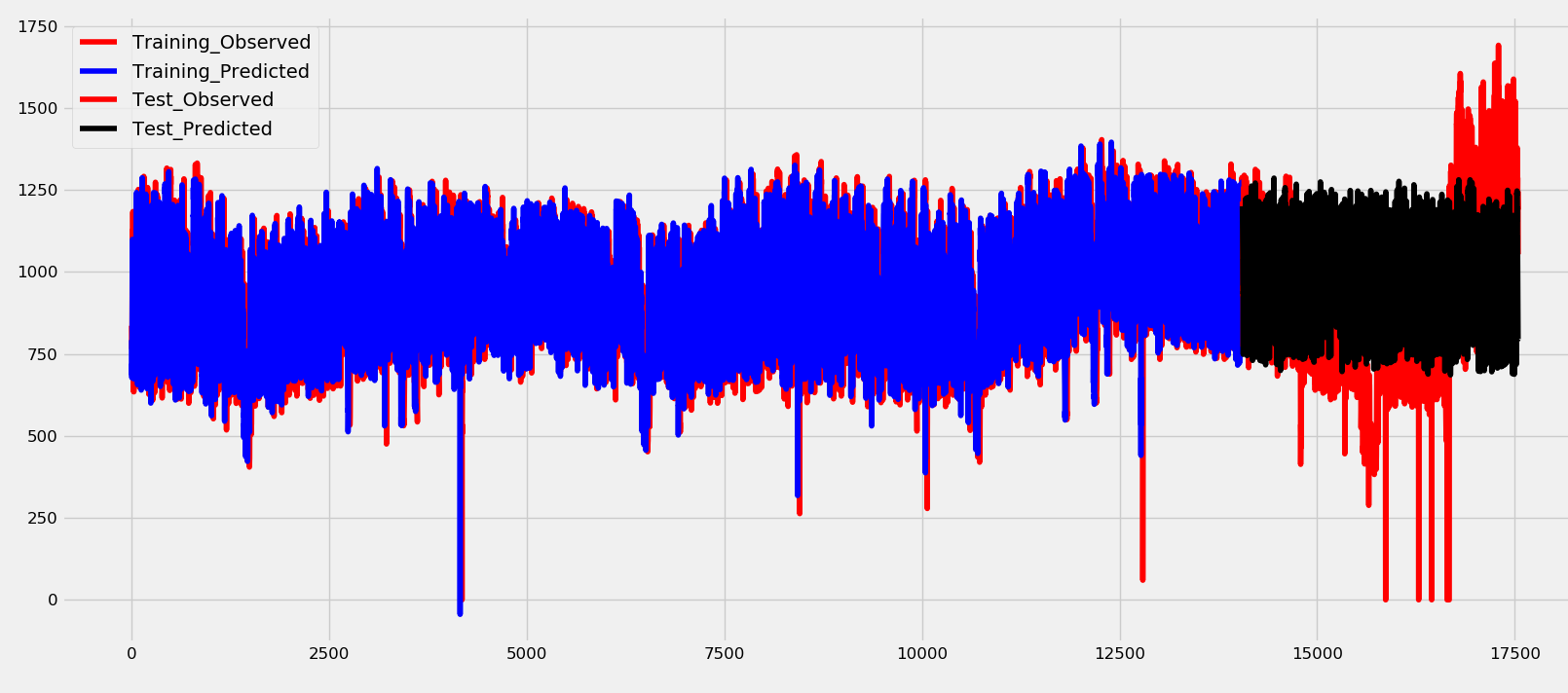
1. We can see the prediction from the fitted ARIMA model along with the 95% confidence interval. This prediction is done using the original test data is input for each “one-step ahead” prediction.



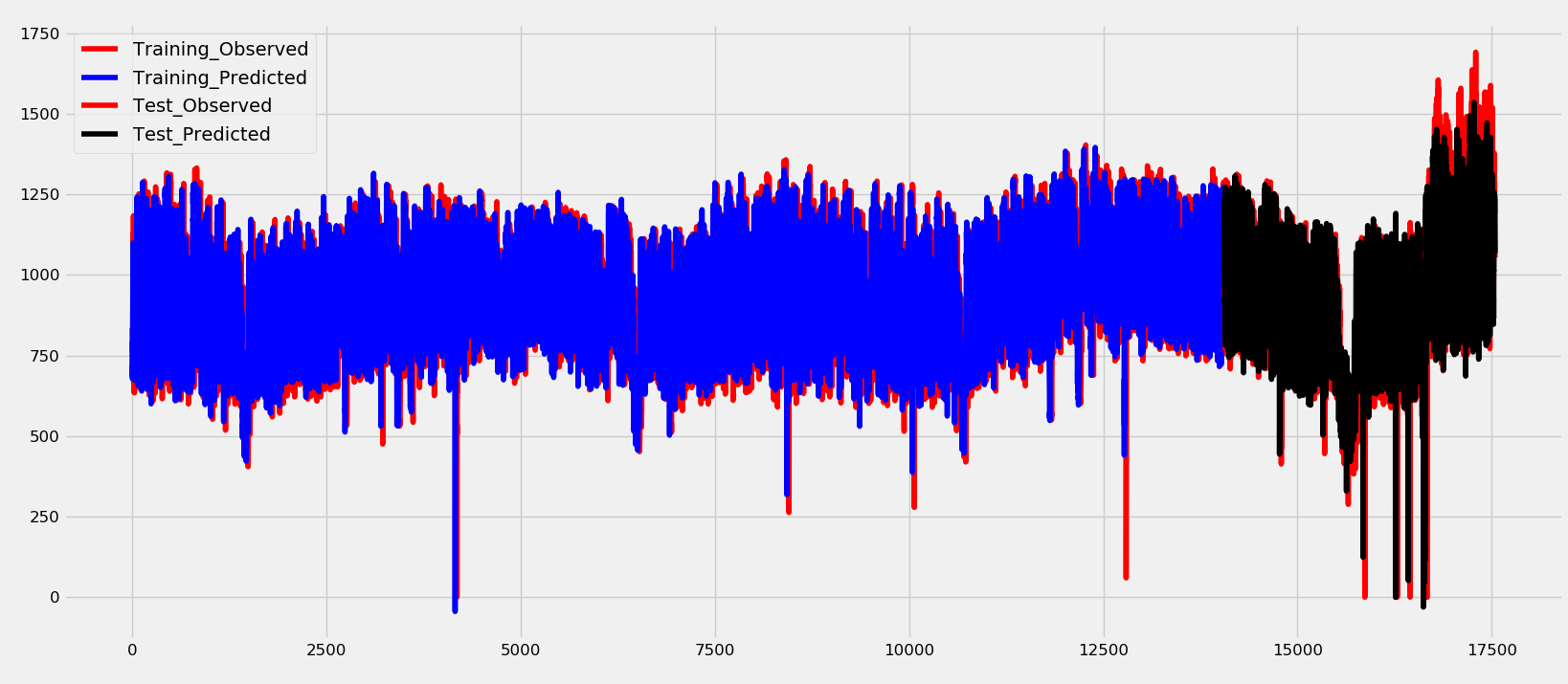
1. We also performed dynamic prediction using the predicted values of ARIMA model as the input for the next time step. We can see the results below:



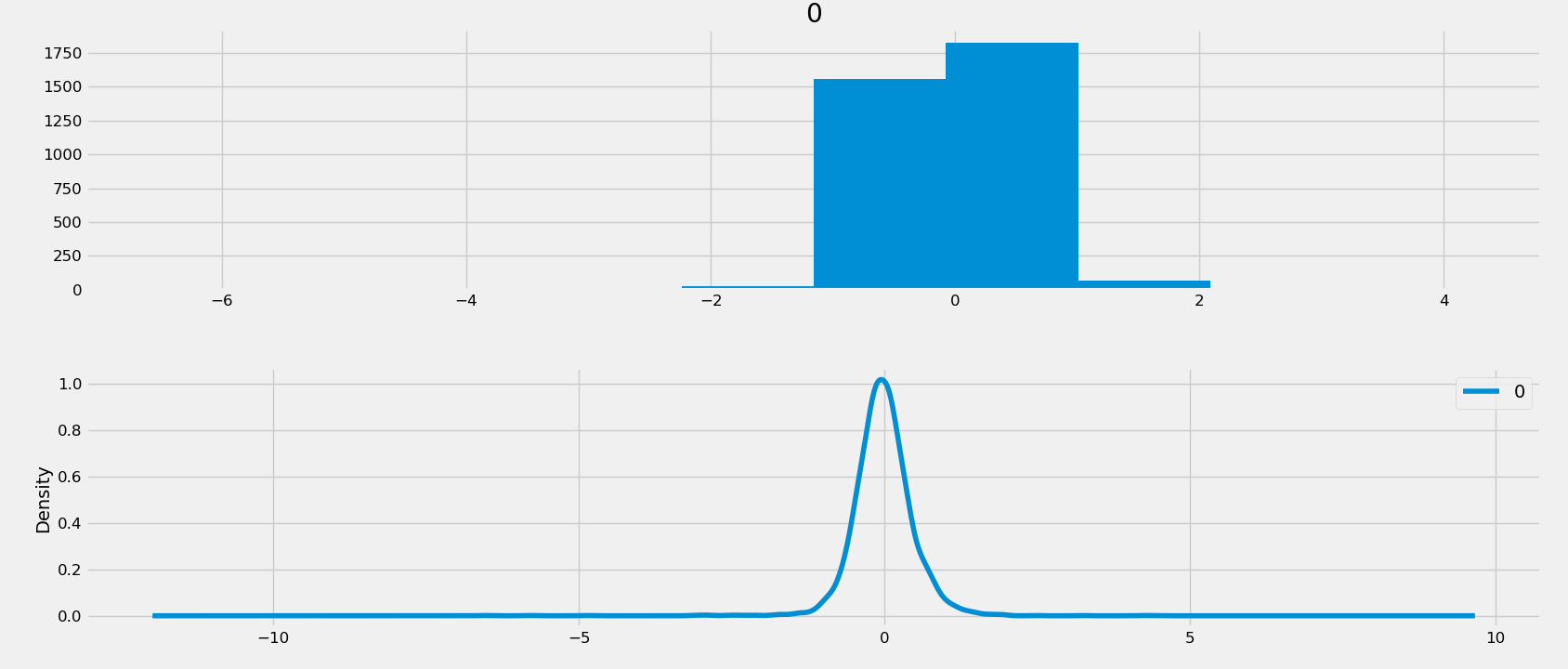
1. The second approach is the Deep Stateful LSTM neural network which takes the previous 24 hours of values as inputs to predict the next value. Below is the result of “dynamic” prediction for test data. We can see that the model has learnt the general trend of the series but is not able to predict some unexpected signals as seen in the end.



1. We also used the trained LSTM model for non-dynamic prediction. In this case, we used the actual values of inputs to predict the output at next time step. As we can see, in this set up, the model performs really well.



1. Checking the distribution of errors for the prediction shown result in point 17. We can see that the errors are centered around 0 which means that the model results are not biased.



1. By checking the ACF and PACF of test prediction errors, we can see that there is some correlation in errors at different lags which means that the model can be further improved.

