Forecasting Stock Prices using AR model

In the previous example we have shown how to identify the AR(p) model and its order p for specific economic time series. In this example it is assumed that the AR(p) model is already identified and p is known. We also assume that the data is stationary and non-seasonal. An example of the AR(2) model is given below, where n_t is Additive White Gaussian Noise - AWGN of variance σ^2.

$$x_t = f_0 + f_1 x_{t-1} + f_2 x_{t-2} + n_t$$

Estimation of parameters

In MatDeck, we have implemented the method of moments to estimate the parameters of AR(p) model. The model is also known as the Yule-Walker estimation, which is based on recursive formula known as the Durbin-Levinson algorithm. The recursive formula is implemented iteratively and thus this method becomes efficient. In MatDeck, the function yulewalker() is used for AR(p) parameter estimation. The function yulewalker() takes training data for parameter estimation, and p order of AR(p) as arguments, and returns the vector of coefficients f_i and variance σ^2 of AWGN according to the equation above. The obtained values are used within arforecast() function to make a prediction of future values.

As an example, for our analysis we take the data set which consists of n = 105 values which are the closing stock price of a share of Google stock between 7-2-2005 to 7-7-2005. The value against time is illustrated below. In order to test the performance of the algorithm, the data set is divided into a training set and a testing set. The length of the training set is 90 samples, and the testing set contains 15 samples.

```
1  time := excel read("googledata.xlsx", "Sheet1", "C2:C106", false)
2  price := excel read("googledata.xlsx", "Sheet1", "B2:B106", false)
3  train := subset(price, 0, 0, 89, 0)
```

We estimate the parameters using yulewalker(), and order p=1.

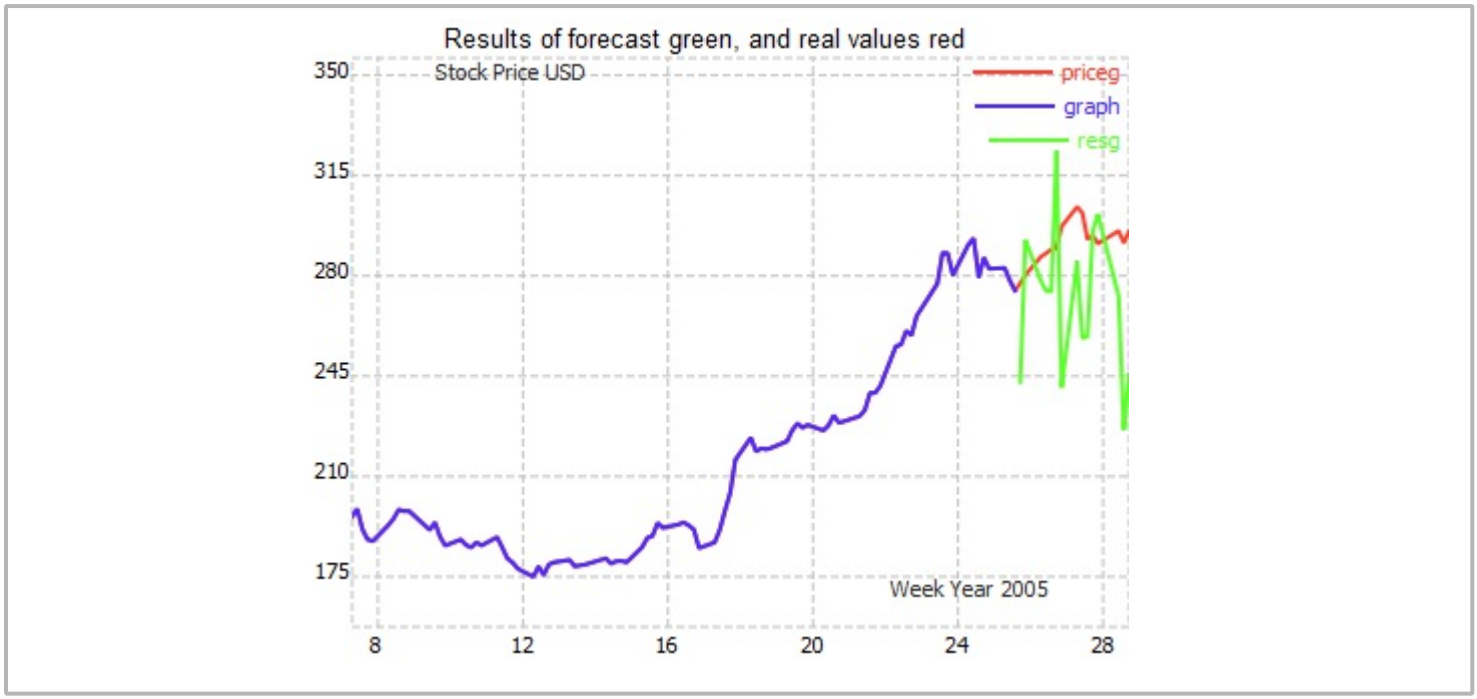
```
4  yw := yulewalker(train, 1)
```

Forecast

Next, the function arforecast() is used to predict the "future" values of the stock price. We predict 15 steps ahead. The result is compared to the testing set in order to test and evaluate the performance. The results are displayed graphically.

```
5  priceg := join mat cols(time, price)
6  graph := join mat cols(subset(time, 0, 0, 89, 0),subset(price, 0, 0, 89,
   0))
7  res := arforecast(yw,subset(price, 0, 0, 89, 0), 15)
8  time1 := subset(time, 90, 0, 104, 0)
9  test := subset(price, 90, 0, 104, 0)
10 resg := join mat cols(time1, res)
```

The remaining task is to estimate the quality of estimation by using the functions for calculation of means square error - mse(), mean absolute error -mae(), mean percentage error -mpe(), and mean absolute percentage error - mape().



Quality Performance	Value
Mean Square Error	$\mathrm{mse}(\mathrm{test}, \mathrm{res}) = 1127.389$
Mean Percentage Error	$\mathrm{mpe}(\mathrm{test}, \mathrm{res}) = 6.879$
Mean Absolute Error	$\mathrm{mae}(\mathrm{test}, \mathrm{res}) = 27.97$
Mean Absolute Percentage Error	$\mathrm{mape}(\mathrm{test}, \mathrm{res}) = 9.556$