

Forecasting Stock Returns using ARIMA model ()

On March 9, 2017

 4 Comments (<https://www.quantinsti.com/blog/forecasting-stock-returns-using-arima-model/#comments>)

G+



21



11



47

79
SHARESBy Milind Paradkar (<https://www.linkedin.com/in/milind-paradkar-b37292107/>)

"Prediction is very difficult, especially about the future". Many of you must have come across this famous quote by Neils Bohr, a Danish physicist. Prediction is the theme of this blog post. In this post, we will cover the popular ARIMA forecasting model to predict returns on a stock and demonstrate a step-by-step process of ARIMA modeling using R programming.

What is a forecasting model in Time Series?

Forecasting involves predicting values for a variable using its historical data points or it can also involve predicting the change in one variable given the change in the value of another variable. Forecasting approaches are primarily categorized into qualitative forecasting and quantitative forecasting. Time series forecasting falls under the category of quantitative forecasting wherein statistical principals and concepts are applied to a given historical data of a variable to forecast the future values of the same variable. Some time series forecasting techniques used include:

- Autoregressive Models (AR)
- Moving Average Models (MA)
- Seasonal Regression Models
- Distributed Lags Models

Search for posts

Search

WANT TO LEARN ALGO TRADING?

Name *

Email *

Phone *

Country *

Are you human?

Y WB 5 T

[I'm Interested](#)

Write a
Guest Post
Here

[\(https://www.quantinsti.com/guest-post/\)](https://www.quantinsti.com/guest-post/)

CATEGORIES

Career Advice

[\(https://www.quantinsti.com/category/career-advice/\)](https://www.quantinsti.com/category/career-advice/)
 (11)

Downloadables

[\(https://www.quantinsti.com/category/downloadables/\)](https://www.quantinsti.com/category/downloadables/)
 (18)

Getting Started

[\(https://www.quantinsti.com/category/quantitative-trading-getting-started/\)](https://www.quantinsti.com/category/quantitative-trading-getting-started/) (86)

Guest Posts

[\(https://www.quantinsti.com/category/guest-posts/\)](https://www.quantinsti.com/category/guest-posts/) (1)

Do you want to learn Algorithmic Trading?

IF YES, CLICK HERE

(<https://www.quantinsti.com/epat/>)

What is Autoregressive Integrated Moving Average (ARIMA)?

ARIMA stands for Autoregressive Integrated Moving Average. ARIMA is also known as Box-Jenkins approach. Box and Jenkins claimed that non-stationary data can be made stationary by differencing the series, Y_t . The general model for Y_t is written as,

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \dots + \theta_q \epsilon_{t-q}$$

Where, Y_t is the differenced time series value, ϕ and θ are unknown parameters and ϵ are independent identically distributed error terms with zero mean. Here, Y_t is expressed in terms of its past values and the current and past values of error terms.

The ARIMA model combines three basic methods:

- AutoRegression (AR) – In auto-regression the values of a given time series data are regressed on their own lagged values, which is indicated by the “p” value in the model.
- Differencing (I-for Integrated) – This involves differencing the time series data to remove the trend and convert a non-stationary time series to a stationary one. This is indicated by the “d” value in the model. If $d = 1$, it looks at the difference between two time series entries, if $d = 2$ it looks at the differences of the differences obtained at $d = 1$, and so forth.
- Moving Average (MA) – The moving average nature of the model is represented by the “q” value which is the number of lagged values of the error term.

This model is called Autoregressive Integrated Moving Average or ARIMA(p,d,q) of Y_t . We will follow the steps enumerated below to build our model.

Step 1: Testing and Ensuring Stationarity

To model a time series with the Box-Jenkins approach, the series has to be stationary. A stationary time series means a time series without trend, one having a constant mean and variance over time, which makes it easy for predicting values.

Testing for stationarity – We test for stationarity using the Augmented Dickey-Fuller unit root test. The p-value resulting from the ADF test has to be less than 0.05 or 5% for a time series to be stationary. If the p-value is greater than 0.05 or 5%, you conclude that the time series has a unit root which means that it is a non-stationary process.

Differencing – To convert a non-stationary process to a stationary process, we apply the differencing method. Differencing a time series means finding the differences between consecutive values of a time series data. The differenced values form a new time series dataset which can be tested to uncover new correlations or other interesting statistical properties.

We can apply the differencing method consecutively more than once, giving rise to the “first differences”, “second order differences”, etc.

We apply the appropriate differencing order (d) to make a time series stationary before we can proceed to the next step.

Step 2: Identification of p and q

In this step, we identify the appropriate order of Autoregressive (AR) and Moving average (MA) processes by using the Autocorrelation function (ACF) and Partial Autocorrelation function (PACF). Please refer to our blog, “Starting out with Time Series” (<https://www.quantinsti.com/blog/starting-time-series/>) for an explanation of ACF and PACF functions.

Identifying the p order of AR model

For AR models, the ACF will dampen exponentially and the PACF will be used to identify the order (p) of the AR model. If we have one significant spike at lag 1 on the PACF, then we have an AR model of the order 1, i.e. AR(1). If we have significant spikes at lag 1, 2, and 3 on the PACF, then we have an AR model of the order 3, i.e. AR(3).

Identifying the q order of MA model

News (<https://www.quantinsti.com/category/news/>) (46)

Events

(<https://www.quantinsti.com/category/news/events-news/>) (30)

Press Releases

(<https://www.quantinsti.com/category/press-releases/>) (3)

Programming and Trading Tools

(<https://www.quantinsti.com/category/programming-and-trading-tools/>) (80)

Other Languages

(<https://www.quantinsti.com/category/programming-and-trading-tools/other-languages/>) (11)

Python

(<https://www.quantinsti.com/category/programming-and-trading-tools/python/>) (28)

R Programming

(<https://www.quantinsti.com/category/programming-and-trading-tools/r-programming/>) (37)

Trading Platforms

(<https://www.quantinsti.com/category/programming-and-trading-tools/trading-platforms/>) (5)

Project Work EPAT

(<https://www.quantinsti.com/category/project-work-epat/>) (14)

Trading Strategies

(<https://www.quantinsti.com/category/quantitative-trading-strategies/>) (65)

Webinars

(<https://www.quantinsti.com/category/events/>) (28)

Previous Webinars

(<https://www.quantinsti.com/category/events/previous>) (28)

HELPFUL SOURCES

Quantocracy (http://quantocracy.com/?utm_source=quantinsti.com)

Quantsportal (http://www.quantsportal.com/?utm_source=quantinsti.com)

Quantpedia (http://quantpedia.com/?utm_source=quantinsti.com)

KDnuggets (http://www.kdnuggets.com/?utm_source=quantinsti.com)

R-bloggers (http://www.r-bloggers.com/?utm_source=quantinsti.com)

The Financial Hacker (http://www.financial-hacker.com/?utm_source=quantinsti.com)

Wall Street Oasis (http://www.wallstreetoasis.com/?utm_source=quantinsti.com)

For MA models, the PACF will dampen exponentially and the ACF plot will be used to identify the order of the MA process. If we have one significant spike at lag 1 on the ACF, then we have an MA model of the order 1, i.e. MA(1). If we have significant spikes at lag 1, 2, and 3 on the ACF, then we have an MA model of the order 3, i.e. MA(3).

Step 3: Estimation and Forecasting

Once we have determined the parameters (p,d,q) we estimate the accuracy of the ARIMA model on a training data set and then use the fitted model to forecast the values of the test data set using a forecasting function. In the end, we cross check whether our forecasted values are in line with the actual values.

Building ARIMA model using R programming

Now, let us follow the steps explained to build an ARIMA model in R. There are a number of packages available for time series analysis and forecasting. We load the relevant R package for time series analysis and pull the stock data from yahoo finance.

```
library(quantmod);library(tseries);
library(timeSeries);library(forecast);library(xts);

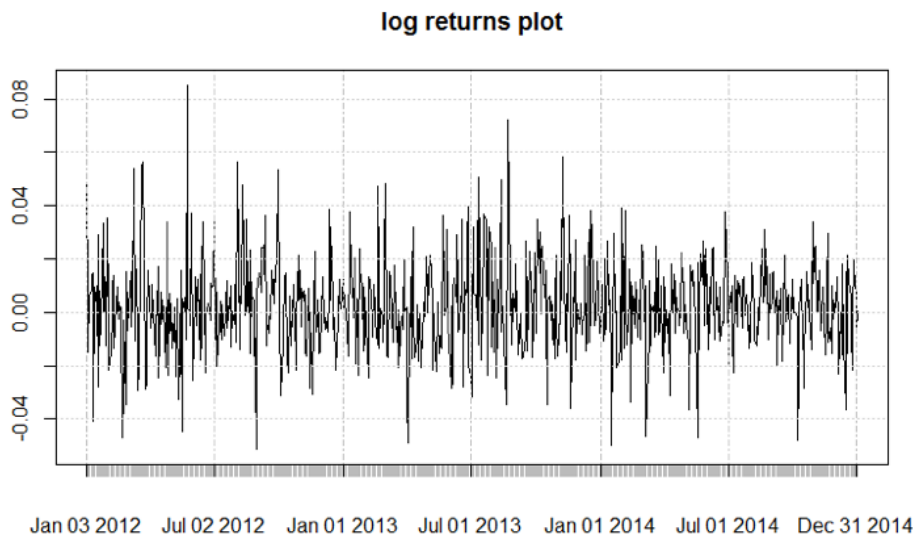
# Pull data from Yahoo finance
getSymbols('TECHM.NS', from='2012-01-01', to='2015-01-01')

# Select the relevant close price series
stock_prices = TECHM.NS[,4]
```

In the next step, we compute the logarithmic returns of the stock as we want the ARIMA model to forecast the log returns and not the stock price. We also plot the log return series using the plot function.

```
# Compute the log returns for the stock
stock = diff(log(stock_prices),lag=1)
stock = stock[!is.na(stock)]

# Plot log returns
plot(stock,type='l', main='log returns plot')
```



Next, we call the ADF test on the returns series data to check for stationarity. The p-value of 0.01 from the ADF test tells us that the series is stationary. If the series were to be non-stationary, we would have first differenced the returns series to make it stationary.

```
# Conduct ADF test on log returns series
print(adf.test(stock))
```

Robot Wealth (http://robotwealth.com/?utm_source=quantinsti.com)

Turing Finance (http://www.turingfinance.com/?utm_source=quantinsti.com)

Augmented Dickey-Fuller Test

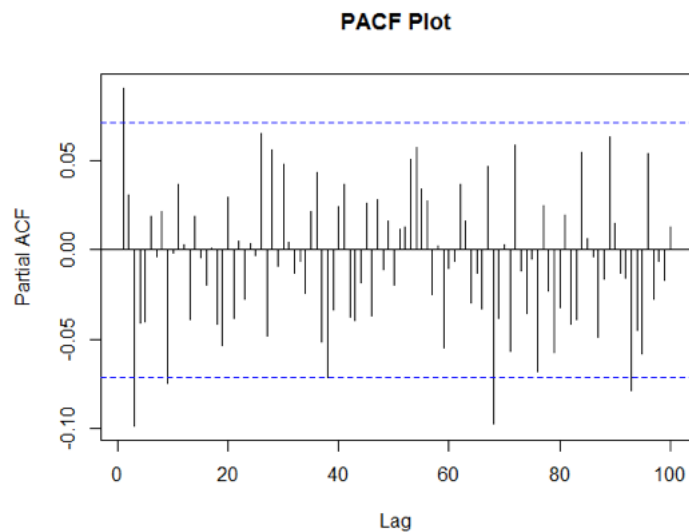
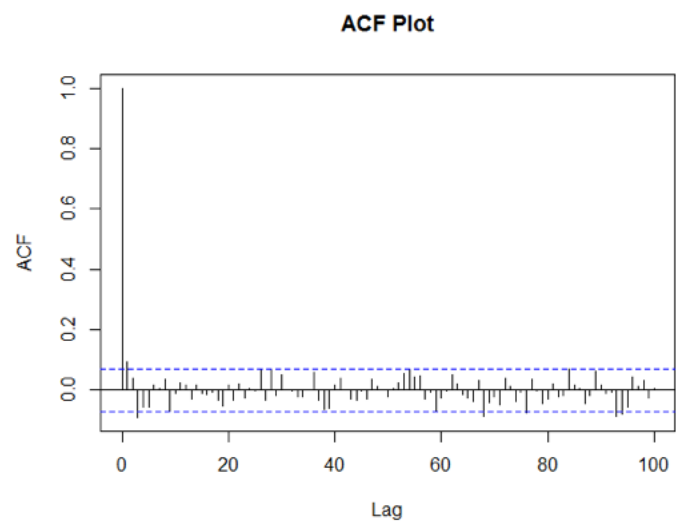
```
data: stock
Dickey-Fuller = -9.5228, Lag order = 9, p-value = 0.01
alternative hypothesis: stationary
```

In the next step, we fixed a breakpoint which will be used to split the returns dataset in two parts further down the code.

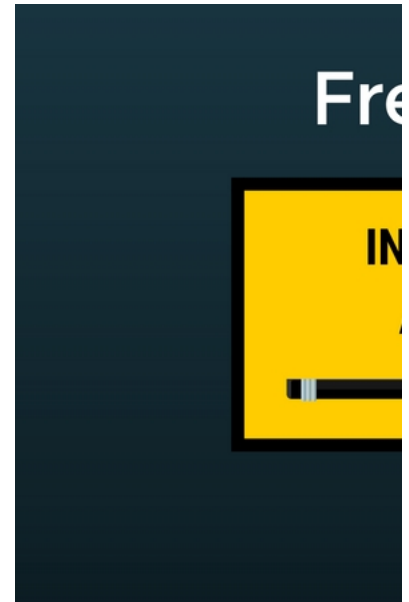
```
# Split the dataset in two parts - training and testing
breakpoint = floor(nrow(stock)*(2.9/3))
```

We truncate the original returns series till the breakpoint, and call the ACF and PACF functions on this truncated series.

```
# Apply the ACF and PACF functions
par(mfrow = c(1,1))
acf.stock = acf(stock[c(1:breakpoint)],, main='ACF Plot', lag.max=100)
pacf.stock = pacf(stock[c(1:breakpoint)],, main='PACF Plot', lag.max=100)
```



Download E-book

**Name *****Email *****Phone *****Country ***

We can observe these plots and arrive at the Autoregressive (AR) order and Moving Average (MA) order.

We know that for AR models, the ACF will dampen exponentially and the PACF plot will be used to identify the order (p) of the AR model. For MA models, the PACF will dampen exponentially and the ACF plot will be used to identify the order (q) of the MA model. From these plots let us select AR order = 2 and MA order = 2. Thus, our ARIMA parameters will be (2,0,2).

Our objective is to forecast the entire returns series from breakpoint onwards. We will make use of the For Loop statement in R and within this loop we will forecast returns for each data point from the test dataset.

In the code given below, we first initialize a series which will store the actual returns and another series to store the forecasted returns. In the For Loop, we first form the training dataset and the test dataset based on the dynamic breakpoint.

We call the arima function on the training dataset for which the order specified is (2, 0, 2). We use this fitted model to forecast the next data point by using the forecast.Arima function. The function is set at 99% confidence level. One can use the confidence level argument to enhance the model. We will be using the forecasted point estimate from the model. The "h" argument in the forecast function indicates the number of values that we want to forecast, in this case, the next day returns.

We can use the summary function to confirm the results of the ARIMA model are within acceptable limits. In the last part, we append every forecasted return and the actual return to the forecasted returns series and the actual returns series respectively.

```
# Initializing an xts object for Actual log returns
Actual_series = xts(0,as.Date("2014-11-25", "%Y-%m-%d"))

# Initializing a dataframe for the forecasted return series
forecasted_series = data.frame(Forecasted = numeric())

for (b in breakpoint:(nrow(stock)-1)) {

  stock_train = stock[1:b, ]
  stock_test = stock[(b+1):nrow(stock), ]

  # Summary of the ARIMA model using the determined (p,d,q) parameters
  fit = arima(stock_train, order = c(2, 0, 2),include.mean=FALSE)
  summary(fit)

  # plotting a acf plot of the residuals
  acf(fit$residuals,main="Residuals plot")

  # Forecasting the log returns
  arima.forecast = forecast.Arima(fit, h = 1,level=99)
  summary(arima.forecast)

  # plotting the forecast
  par(mfrow=c(1,1))
  plot(arima.forecast, main = "ARIMA Forecast")

  # Creating a series of forecasted returns for the forecasted period
  forecasted_series = rbind(forecasted_series,arima.forecast$mean[1])
  colnames(forecasted_series) = c("Forecasted")

  # Creating a series of actual returns for the forecasted period
  Actual_return = stock[(b+1),]
  Actual_series = c(Actual_series,xts(Actual_return))
  rm(Actual_return)

  print(stock_prices[(b+1),])
  print(stock_prices[(b+2),])

}
```

Before we move to the last part of the code, let us check the results of the ARIMA model for a sample data point from the test dataset.

```

Model Information:

Call:
arima(x = stock_train, order = c(2, 0, 2), include.mean = FALSE)

Coefficients:
      ar1      ar2      ma1      ma2
    0.6072 -0.8818 -0.5447  0.8972
s.e.  0.0504  0.0739  0.0533  0.0687

sigma^2 estimated as 0.00029:  log likelihood = 2075.27,  aic = -4140.54

Error measures:
      ME      RMSE      MAE  MPE  MAPE      MASE      ACF1
Training set 0.001821791 0.01702845 0.01258089 NaN   Inf  0.7060578 0.01843719

Forecasts:
      Point Forecast      Lo 99      Hi 99
783  -0.001326978 -0.04518937  0.04253541

```

From the coefficients obtained, the return equation can be written as:

$$Y_t = 0.6072 \cdot Y_{(t-1)} - 0.8818 \cdot Y_{(t-2)} - 0.5447 \epsilon_{(t-1)} + 0.8972 \epsilon_{(t-2)}$$

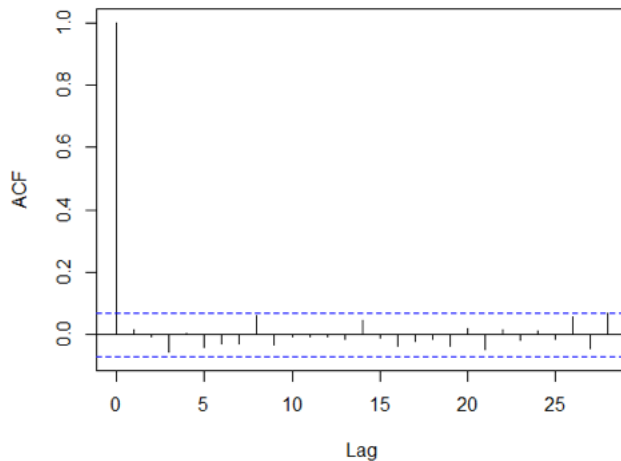
The standard error is given for the coefficients, and this needs to be within the acceptable limits. The Akaike information criterion (AIC) score is a good indicator of the ARIMA model accuracy. Lower the AIC score better the model. We can also view the ACF plot of the residuals; a good ARIMA model will have its autocorrelations below the threshold limit. The forecasted point return is -0.001326978, which is given in the last row of the output.

Become an Algo Trader from your home!

SIGN UP TO KNOW HOW

(<https://www.quantinsti.com/epat/>)

Residuals plot



Let us check the accuracy of the ARIMA model by comparing the forecasted returns versus the actual returns. The last part of the code computes this accuracy information.

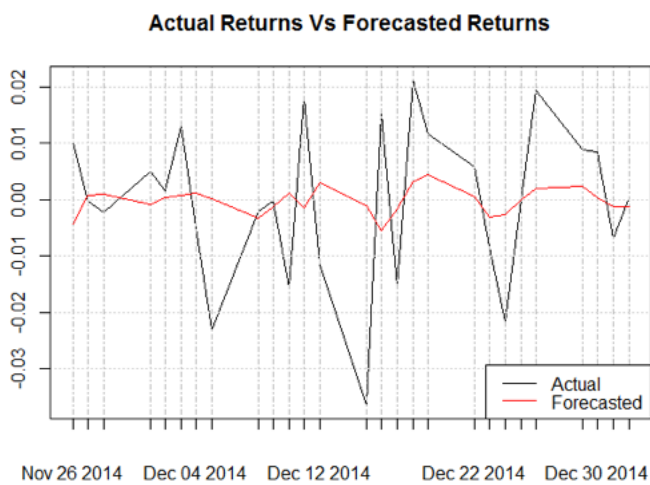
```
# Adjust the length of the Actual return series
Actual_series = Actual_series[-1]

# Create a time series object of the forecasted series
forecasted_series = xts(forecasted_series,index(Actual_series))

# Create a plot of the two return series - Actual versus Forecasted
plot(Actual_series,type='l',main='Actual Returns Vs Forecasted Returns')
lines(forecasted_series,lwd=1.5,col='red')
legend('bottomright',c("Actual", "Forecasted"),lty=c(1,1),lwd=c(1.5,1.5),col=c('black', 'red'
))

# Create a table for the accuracy of the forecast
comparision = merge(Actual_series,forecasted_series)
comparision$Accuracy = sign(comparision$Actual_series)==sign(comparision$Forecasted)
print(comparision)

# Compute the accuracy percentage metric
Accuracy_percentage = sum(comparision$Accuracy == 1)*100/length(comparision$Accuracy)
print(Accuracy_percentage)
```



	Actual_series	Forecasted	Accuracy
2014-11-26	0.0101377233	-0.0043090616	0
2014-11-27	-0.0003219135	0.0007807852	0
2014-11-28	-0.0023441737	0.0010150580	0
2014-12-01	0.0049120615	-0.0009070759	0
2014-12-02	0.0016375258	0.0003435073	1
2014-12-03	0.0131590684	0.0008073216	1
2014-12-04	-0.0049711114	0.0011228101	0
2014-12-05	-0.0230689605	0.0001129290	0
2014-12-08	-0.0020447544	-0.0032902765	1
2014-12-09	-0.0002495719	-0.0012435422	1
2014-12-10	-0.0156834851	0.0012498747	0
2014-12-11	0.0182084822	-0.0014586629	0
2014-12-12	-0.0115992224	0.0029173522	0
2014-12-15	-0.0365085594	-0.0010131041	1
2014-12-16	0.0154268090	-0.005505608	0
2014-12-17	-0.0154268090	-0.0015995673	1
2014-12-18	0.0212679508	0.0031471618	1
2014-12-19	0.0116987764	0.0043570056	1
2014-12-22	0.0058169635	0.0005555327	1
2014-12-23	-0.0083976341	-0.0030760477	1
2014-12-24	-0.0216700763	-0.0026127616	1
2014-12-25	0.0000000000	-0.0001160376	0
2014-12-26	0.0193149604	0.0019859033	1
2014-12-29	0.0089410911	0.0024058764	1
2014-12-30	0.0084193215	0.0002595307	1
2014-12-31	-0.0069530750	-0.0013568468	1
2015-01-01	0.0003854973	-0.0013269785	0
[1]	55.55556		

If the sign of the forecasted return equals the sign of the actual returns we have assigned it a positive accuracy score. The accuracy percentage of the model comes to around 55% which looks like a decent number. One can try running the model for other possible combinations of (p,d,q) or instead use the auto.arima function which selects the best

optimal parameters to run the model.

Conclusion

To conclude, in this post we covered the ARIMA model and applied it for forecasting stock price returns using R programming language. We also cross checked our forecasted results with the actual returns. In our upcoming posts, we will cover other time series forecasting techniques and try them in Python/R programming languages.

Next Step

If you want to learn various aspects of Algorithmic trading then check out the Executive Programme in Algorithmic Trading (EPAT™) (<https://www.quantinsti.com/epat/>). The course covers training modules like Statistics & Econometrics, Financial Computing & Technology, and Algorithmic & Quantitative Trading. EPAT™ equips you with the required skill sets to build a promising career in algorithmic trading. Enroll now! (<https://www.quantinsti.com/epat/>)

Related Posts:



Starting Out with Time Series

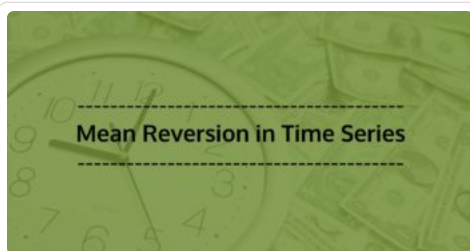


with...

<https://www.quantinsti.com/blog/epat-project->



Build Technical Indicators in Python



Mean Reversion in Time Series

G+



21

f

11

in

47



79
SHARES

4 thoughts on “Forecasting Stock Returns using ARIMA model”



March 11, 2017

Thomas Haas

Reply

(<https://www.quantinsti.com/blog/forecasting-stock-returns-using-arima-model/?replytocom=7651#respond>)

Thanks for posting this. Very interesting article, and I will be waiting for the follow-ons.

I think the return equation should be ...

$$Y_t = 0.6072 Y_{t-1} - 0.8818 Y_{t-2} - 0.5447 \epsilon_{t-1} + 0.8972 \epsilon_{t-2}$$

The present equation has $0.8972 \epsilon_{t-1}$.



March 12, 2017

Blanco saco

Reply

(<https://www.quantinsti.com/blog/forecasting-stock-returns-using-arima-model/?replytocom=7663#respond>)

you can do practically all the steps with only one with the package forecast. Have you read the instructions?



July 24, 2017

baljinder

Reply

(<https://www.quantinsti.com/blog/forecasting-stock-returns-using-arima-model/?replytocom=13126#respond>)

hello sir can u please tell me forecast.arimr in for loop



August 2, 2017

admin

Reply

(<https://www.quantinsti.com/blog/forecasting-stock-returns-using-arima-model/?replytocom=13738#respond>)

We are using the “For loop” because it adds latest value to the stock_train data set in every iteration. This updated stock_train data set is used in the arima function.

```
fit = arima(stock_train, order = c(2, 0, 2), include.mean=FALSE)
```

The resultant object of class "Arima" is used as an input to the forecast.Arima function which given as forecast.Arima(fit, h = 1, level = 99).

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

I'm not a robot

reCAPTCHA
Privacy - Terms

Post Comment

Quick Links

About Us (<https://www.quantinsti.com/about-us/>)
 EPAT™ (<https://www.quantinsti.com/epat/>)
 Alumni (<https://www.quantinsti.com/alumni/>)
 Testimonials (<https://www.quantinsti.com/testimonials/>)
 Faculty (<https://www.quantinsti.com/faculty/>)
 Careers (<https://www.quantinsti.com/careers/>)
 Quantra™ (<https://quantra.quantinsti.com/>)
 Placement Cell (<https://www.quantinsti.com/placement-cell/>)
 BTRM (<https://www.quantinsti.com/btrm/>)
 Contact Us (<https://www.quantinsti.com/contact-us/>)
 Disclaimer Policy (<https://www.quantinsti.com/disclaimer-policy/>)
 Privacy Policy (<https://www.quantinsti.com/privacy-policy/>)
 Terms & Conditions (<https://www.quantinsti.com/terms-conditions/>)
 Pay Your Fees (<https://www.quantinsti.com/payment/>)
 Student Login (<http://quantinsti.talentlms.com/>)

India

QuantInsti Quantitative Learning Pvt Ltd
 A-309, Boomerang,
 Chandivali Farm Road, Powai,
 Mumbai – 400 072
 Toll Free: 1800-266-5401
 Phone: +91-22-61691400

Singapore

30 Cecil Street, #19-08,
 Prudential Tower,
 Singapore – 049712
 Phone: +65-9057-8301

Connect with us...



(<https://www.facebook.com/quantinsti>) (<https://twitter.com/quantinsti>)



(<http://quantocracy.com/>) Show us

QUANTOCRACY
POWER TO THE QUANTS

some love on Quantocracy.

Copyright © 2017 QuantInsti.com All Rights Reserved.

