INTRODUCTION TO TIME SERIES ANALYSIS IN PYTHON

Describe Model

Rob Reider
Adjunct Professor, NYU-Courant
Consultant, Quantopian

Mathematical Decription of MA(1) Model

moving average

noise yesterday's noise

$$R_t = \mu + \epsilon_t 1 + \theta\, \epsilon_{t-1}$$

- Since only one lagged error on right hand side, this is called:
  - MA model of order 1, or
  - MA(1) model
- MA parameter is θ =0, --> white noise
- Stationary for all values of θ

Interpretation of MA(1) Parameter

$$R_t = \mu + \epsilon_t + \theta\, \epsilon_{t-1}$$

theta is -, a + shock last period, represented by epsilon t-1 would have caused last period's return to be +, but this periods return is more likely to be -.

- Negative θ: One-Period Mean Reversion
- Positive θ: One-Period Momentum
- Note: One-period autocorrelation is $\theta/(1+\theta^2)$, not θ

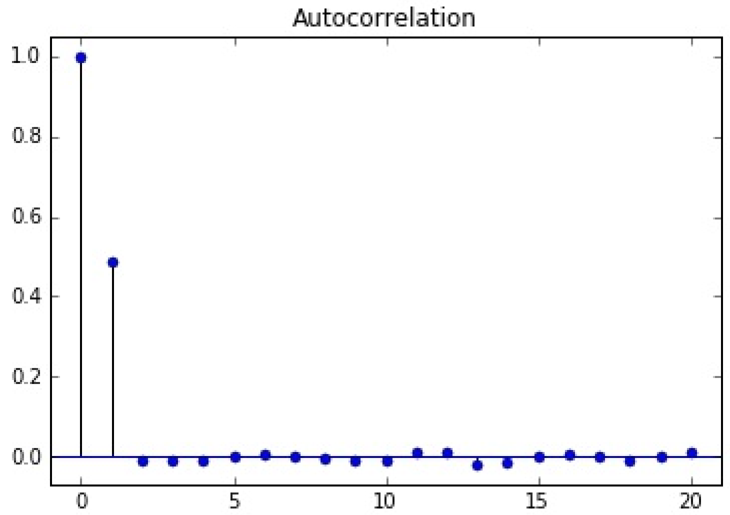
a shock 2 periods ago would have no effect on today's return - only the shock now and last period.

each case, there is zero autocorrelation for MA(1) beyond lag-1

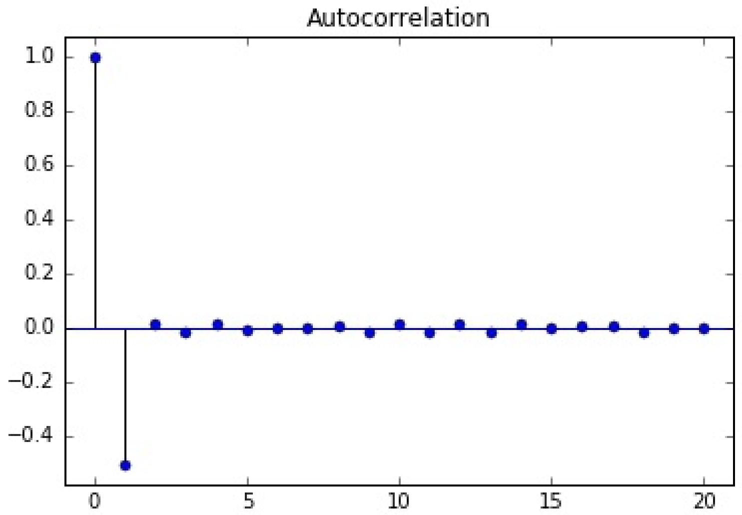
Comparison of MA(1) Autocorrelation Functions

theta > 0 , lag-1 autocorrelation is >0. vice versa

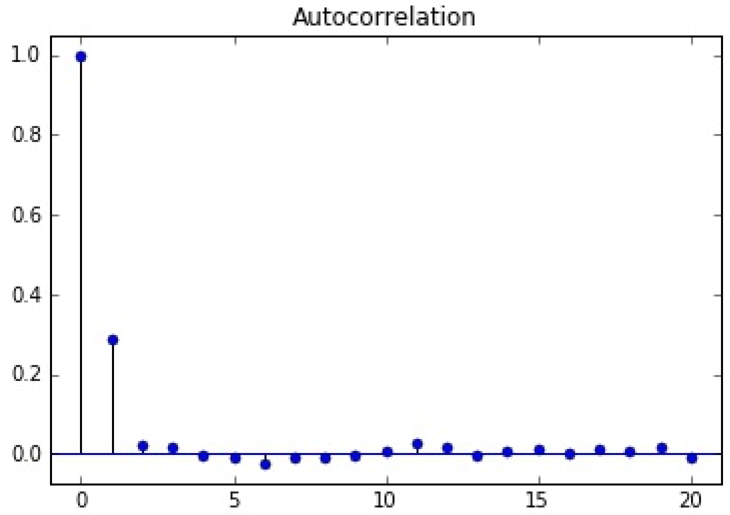
- $\theta = 0.9$



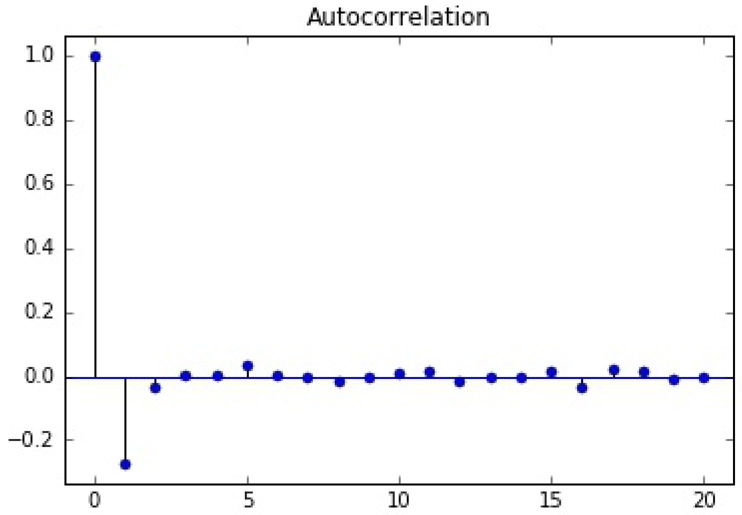
- $\phi = -0.9$



- $\phi = 0.5$

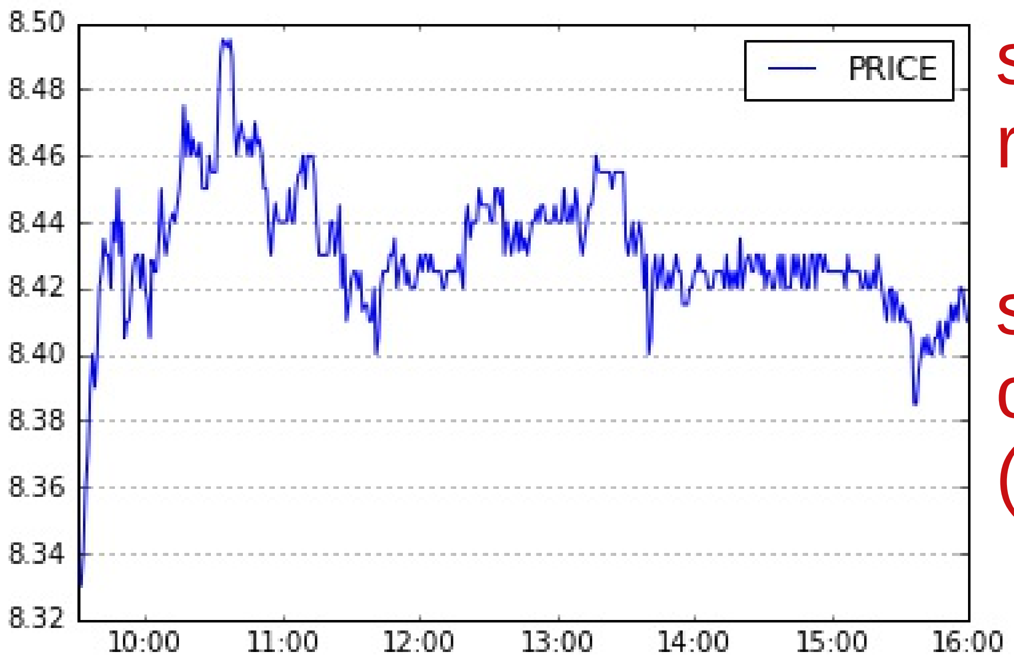


- $\phi = -0.5$



Example of MA(1) Process: Intraday Stock Returns

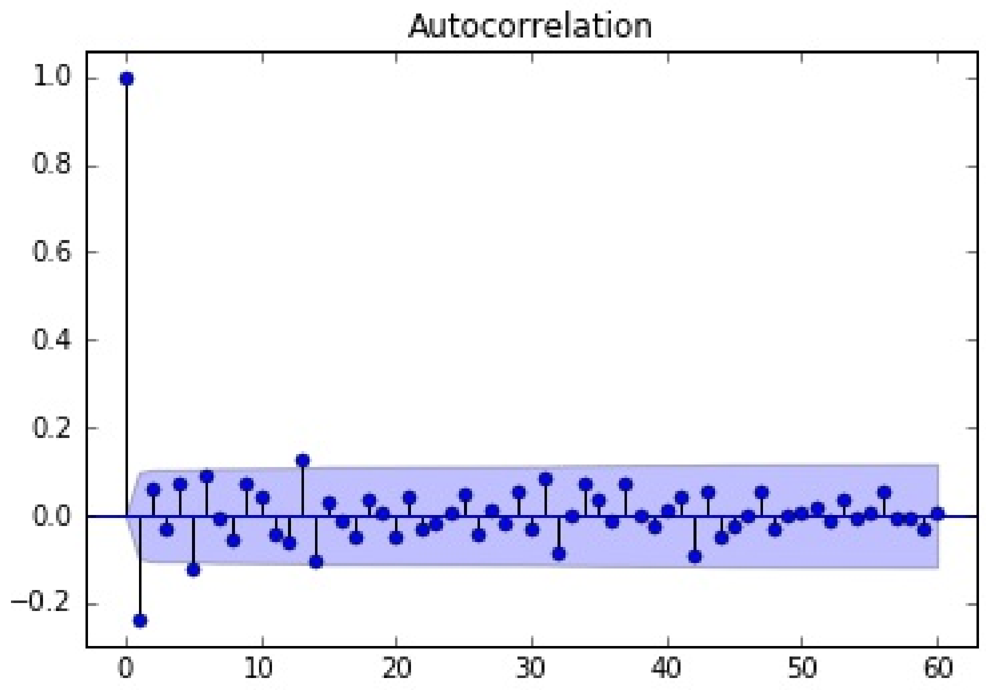
stock price for one day



stocks trade at discrete one-cent increments rather than at continuous prices

stock can bounce back and forth over a one cent range for long periods of time (sometime referred to as "bid/ask bounce"

Autocorrelation Function of Intraday Stock Returns



bid/ask bounce induces a negative lag-1 autocorrelation, but no autocorrelation beyond lag-1

Higher Order MA Models

- MA(1)

$$R_t = \mu + \epsilon_t - \theta_1 \, \epsilon_{t-1}$$

- MA(2)

$$R_t = \mu + \epsilon_t - \theta_1 \, \epsilon_{t-1} - \theta_2 \, \epsilon_{t-2}$$

- MA(3)

$$R_t = \mu + \epsilon_t - \theta_1 \, \epsilon_{t-1} - \theta_2 \, \epsilon_{t-2} - \theta_3 \, \epsilon_{t-3}$$

- ...

simulate a pure MA process

Simulating an MA Process

```
from statsmodels.tsa.arima_process import ArmaProcess
ar = np.array([1])
ma = np.array([1, 0.5])
AR_object = ArmaProcess(ar, ma)
simulated_data = AR_object.generate_sample(nsample=1000)
plt.plot(simulated_data)
```

ar order is just an array containing 1
ma order is an array containing 1 and MA(1) parameter theta
No need to reverse sign of theta

INTRODUCTION TO TIME SERIES ANALYSIS IN PYTHON

Let's practice!

INTRODUCTION TO TIME SERIES ANALYSIS IN PYTHON

Estimation and Forecasting an MA Model

Rob Reider
Adjunct Professor, NYU-Courant
Consultant, Quantopian

Estimating an MA Model

same module that u used to estimate the parameters of AR model can be used to estimate the parameters of an MA model

- Same as estimating an AR model (except order=(0,1))

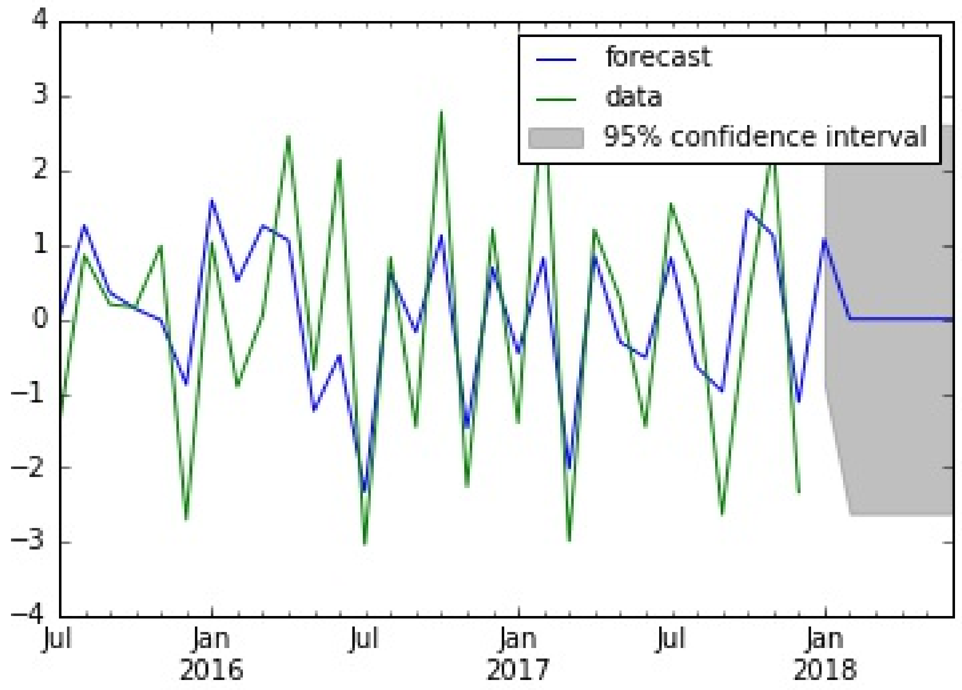
```
from statsmodels.tsa.arima_model import ARMA
mod = ARMA(simulated_data, order=(0,1))
result = mod.fit()
```

order is (0,1) for MA(1)
is (1,0) for AR(1)

Forecasting an MA Model

is the same as for an AR model

```
from statsmodels.tsa.arima_model import ARMA
mod = ARMA(simulated_data, order=(0,1))
res = mod.fit()
res.plot_predict(start='2016-07-01', end='2017-06-01')
plt.show()
```



note that with an MA(1) model, unlike an AR model, all forecasts beyond the one-step ahead forecast will be the same.

INTRODUCTION TO TIME SERIES ANALYSIS IN PYTHON

Let's practice!

INTRODUCTION TO TIME SERIES ANALYSIS IN PYTHON

ARMA models

an ARMA model is combination of an AR and MA model

Rob Reider
Adjunct Professor, NYU-Courant
Consultant, Quantopian

ARMA Model

- ARMA(1,1) model:

$$R_t = \mu + \phi\, R_{t-1} + \epsilon_t + \theta\, \epsilon_{t-1}$$

Converting Between ARMA, AR, and MA Models

- Converting AR(1) into an MA(infinity)

ARMA models can be converted to pure AR or pure MA models

$$R_t = \mu + \phi R_{t-1} + \epsilon_t$$ AR(1) model

$$R_t = \mu + \phi(\mu + R_{t-2} + \epsilon_{t-1}) + \epsilon_t$$ R_t-1 replace by AR(1)

$$R_t = \mu + \phi(\mu + \phi(\mu + R_{t-3} + \epsilon_{t-2}) + \epsilon_{t-1}) + \epsilon_t$$

$$\vdots$$

$$R_t = \mu + \epsilon_t + \phi\,\epsilon_{t-1} + \phi^2\,\epsilon_{t-2} + \phi^3\,\epsilon_{t-3} + \phi^4\,\epsilon_{t-4} + \cdots$$

INTRODUCTION TO TIME SERIES ANALYSIS IN PYTHON

Let's practice!