Financial Time Series Analysis & Update: ARIMA Time Series Models

Bharathi Srinivasan

David Berscheid

Ladislaus von Bortkiewicz Chair of Statistics
Humboldt-Universität zu Berlin
lvb.wiwi.hu-berlin.de

Financial Time Series Analysis in Python

Time series analysis attempts to understand the past and predict the future
- Michael Halls Moore [Quantstart.com]

Why is stationarity such an important concept?

- time series becomes easier to predict
- assumption that future statistical properties are the same or proportional to current statistical properties
- most of the models use covariance-stationarity

Another key concept - Serial correlation:

- Critical for the validity of our model predictions
- Intrinsically related to stationarity

Intro:

By developing our time series analysis (TSA) skillset we are better able to understand what has already happened, and make better, more profitable, predictions of the future.

Example applications : predicting future asset returns, future correlations/covariances, and future volatility.

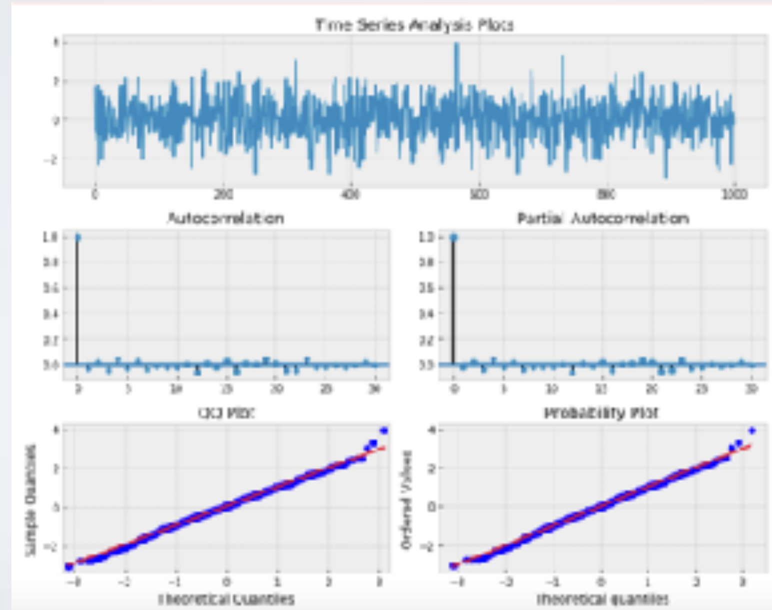
Serial Correlation/ Autocorrelation:

Essentially when we model a time series we decompose the series into three components: trend, seasonal/cyclical, and random. The random component is called the residual or error. It is simply the difference between our predicted value(s) and the observed value(s). Serial correlation is when the residuals (errors) of our TS models are correlated with each other.

Recall that the residuals (errors) of a stationary TS are serially uncorrelated by definition! If we fail to account for this in our models the standard errors of our coefficients are underestimated, inflating the size of our T-statistics. The result is too many Type-1 errors, where we reject our null hypothesis even when it is True! In layman's terms, ignoring autocorrelation means our model predictions will be bunk, and we're likely to draw incorrect conclusions about the impact of the independent variables in our model.

White Noise

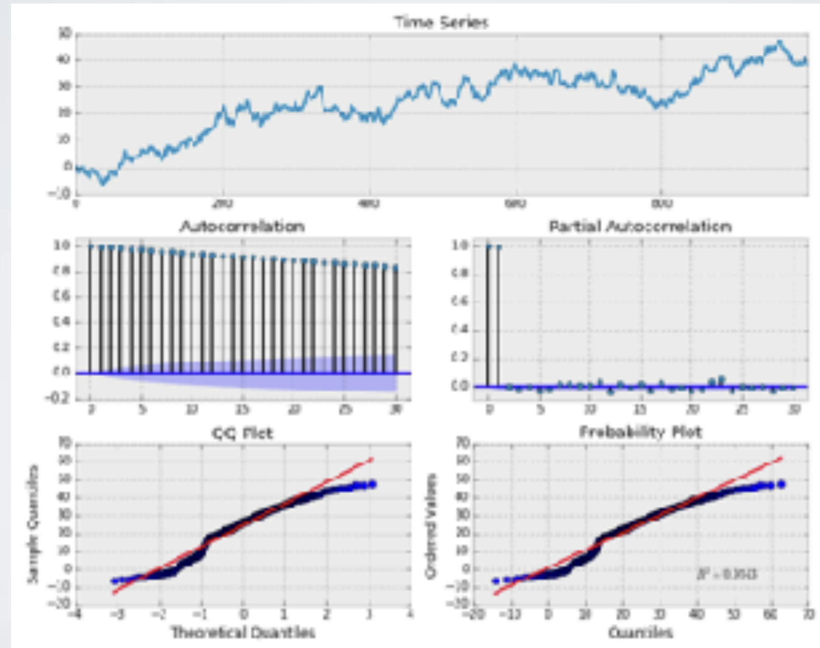
- White noise process has serially uncorrelated errors and expected mean of the errors is zero
- Importance: If the time series model successfully captures the underlying model, residuals will resemble a white noise process



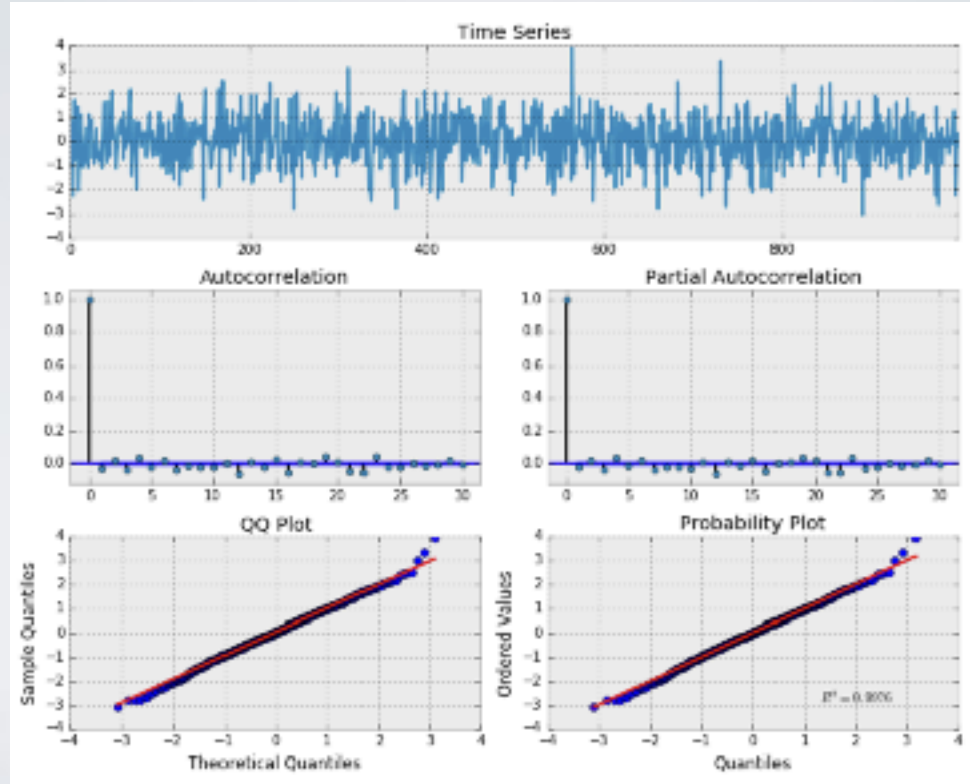
We can see that process appears to be random and centered about zero. The autocorrelation (ACF) and partial autocorrelation (PACF) plots also indicate no significant serial correlation. Keep in mind we should see approximately 5% significance in the autocorrelation plots due to pure chance as a result of sampling from the Normal distribution. Below that we can see the QQ and Probability Plots, which compares the distribution of our data with another theoretical distribution. In this case, that theoretical distribution is the standard normal distribution. Clearly our data is distributed randomly, and appears to follow Gaussian (Normal) white noise, as it should.

Random Walk

- non stationary: covariance between observations is time dependent

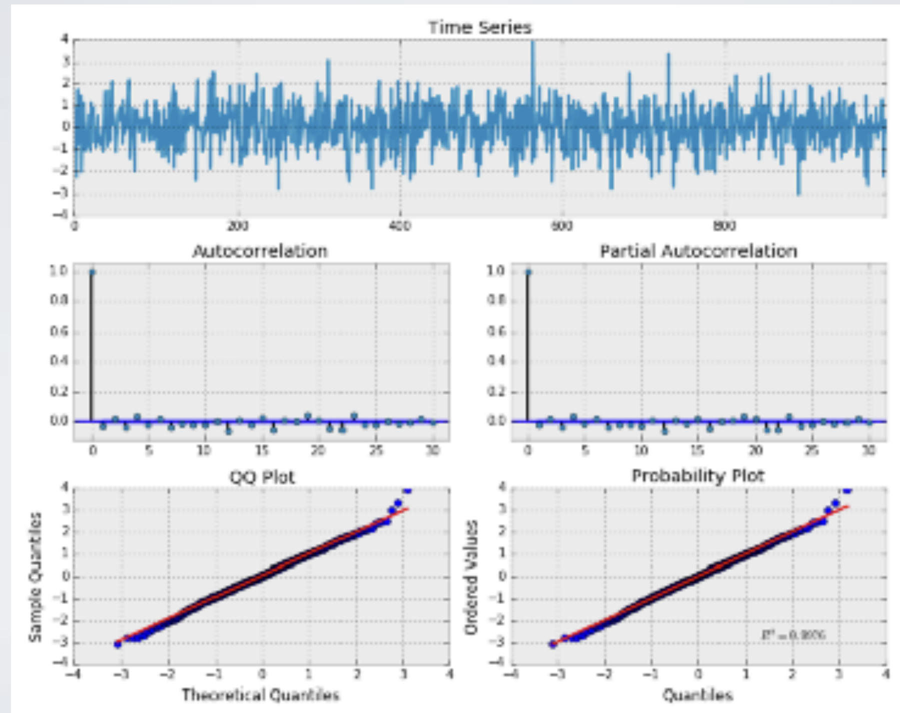


First difference of Random Walk



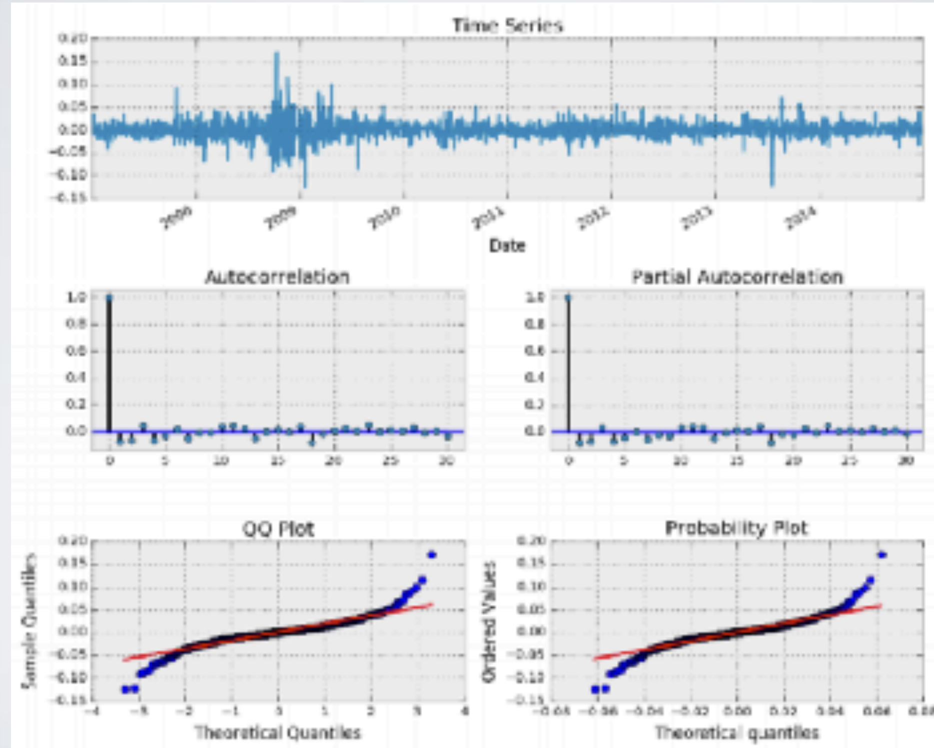
Clearly our TS is not stationary. Let's find out if the random walk model is a good fit for our simulated data. Recall that a random walk is xt = xt-1 + wt. Using algebra we can say that xt - xt-1 = wt. Thus the first differences of our random walk series should equal a white noise process! We can use the "np.diff()" function on our TS and see if this holds.

Fitting a Random Walk to S&P 500 ETF Prices



Wow, it's quite similar to white noise. However, notice the shape of the QQ and Probability plots. This indicates that the process is close to normality but with 'heavy tails'. There also appears to be some significant serial correlation in the ACF, and PACF plots around lags 1, 5?, 16?, 18 and 21. This means that there should be better models to describe the actual price change process.

How does AR process fit MSFT log returns?



The best order is 23 lags or 23 parameters! Any model with this many parameters is unlikely to be useful in practice. Clearly there is more complexity underlying the returns process than this model can explain.

"Reproduce Figures 6 through 8 for more data points"

Example: MA(1) process

$$X_t = \beta\varepsilon_{t-1} + \varepsilon_t$$

Comparison of conditional and unconditional likelihood functions

$$\beta = 0.5 \text{ and } \varepsilon_t \sim \mathrm{N}(0,1) \text{ with } n = 10 \text{ and } n = 20$$

The matrix Γ is band diagonal with elements $1+\beta^2$ on the main diagonal and β on the lower and higher diagonal.

Since the process contains only one parameter we may display the likelihood functions on a grid.

Example: MA(1) process

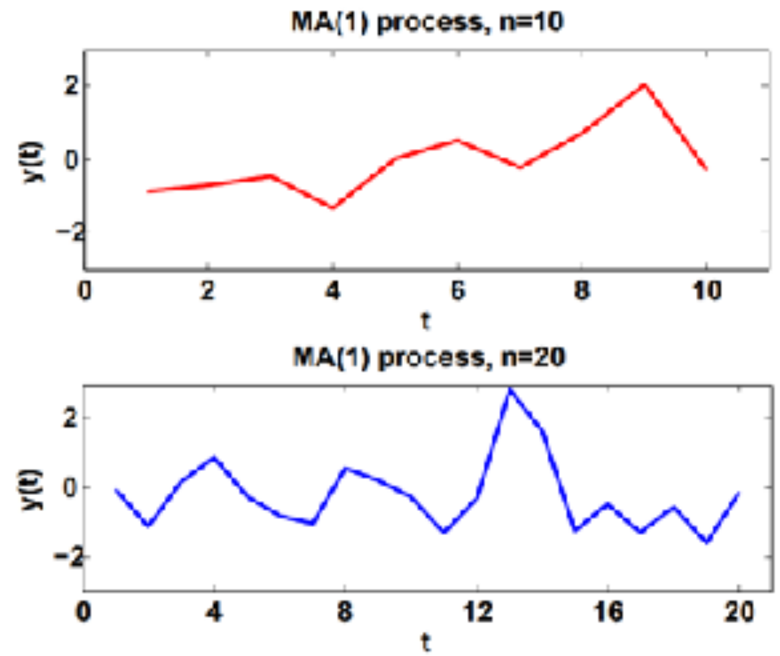


Figure 6: β = 0.5 and εt ~ N(0, 1), n = 10 (red) and n = 20 (blue)

Example: MA(1) process

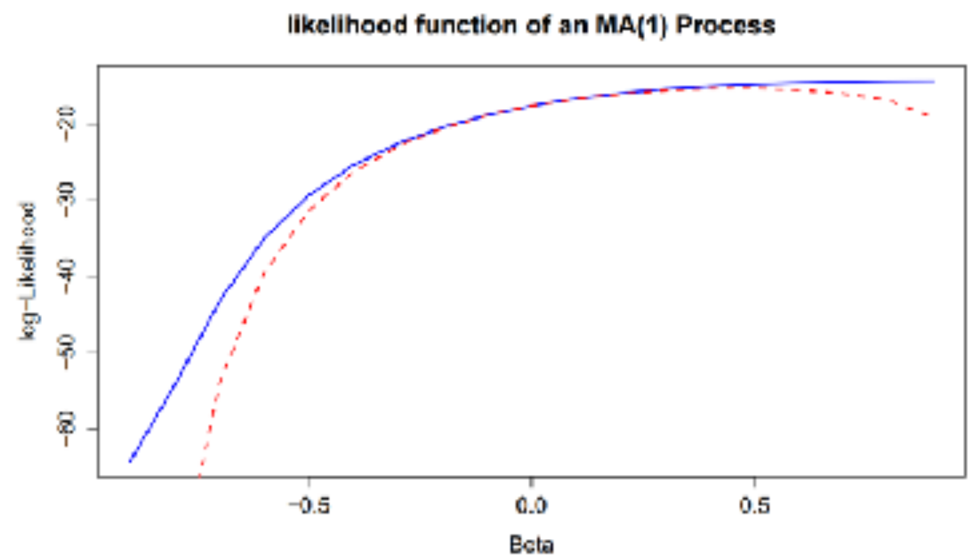


Figure 7: Exact (blue) and conditional (red) likelihood function of an MA(1) process from Figure 6 with $n = 10$. The true parameter is $\beta = 0.5$.

Example: MA(1) process

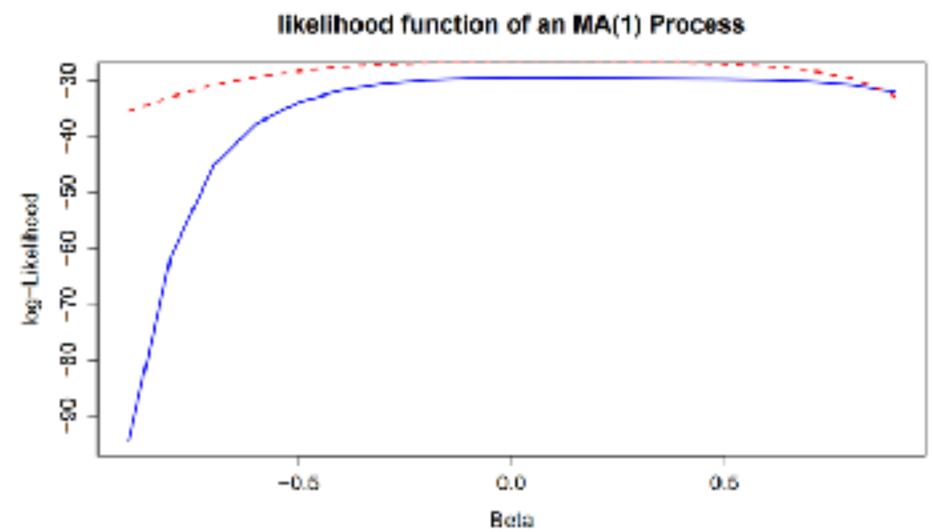


Figure 8: Exact (blue) and conditional (red) likelihood function of an MA(1) process from Figure 6 with n = 20. The true parameter is β = 0.5.

MA(1) process - increased number of data points

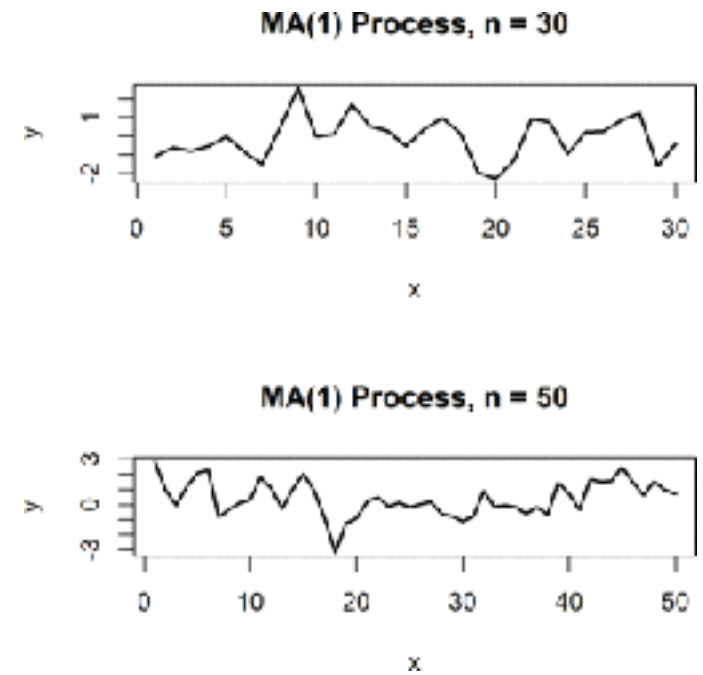


Figure 6a: $\beta = 0.5$ and $\varepsilon_t \sim N(0, 1)$, $n = 30$ (upper) and $n = 50$ (lower)

MA(1) process - increased number of data points

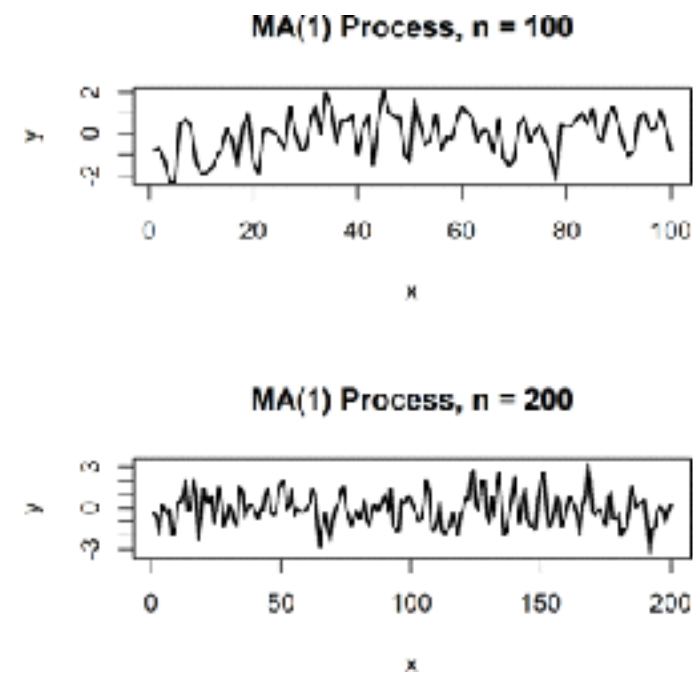


Figure 6b: β = 0.5 and εt ~ N(0, 1), n = 100 (upper) and n = 200 (lower)

Example: MA(1) process

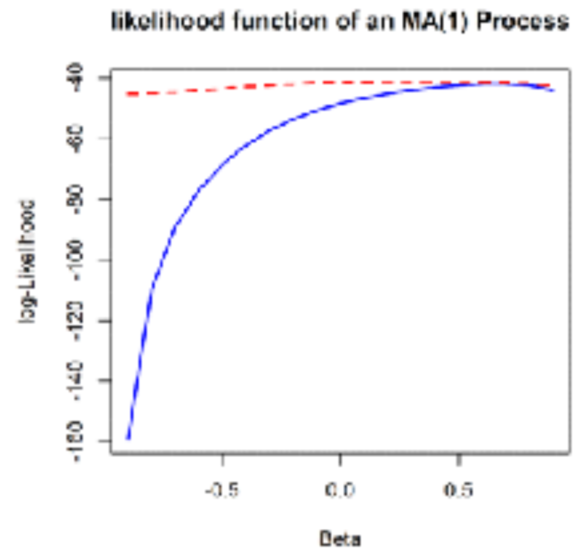


Figure 7a: Exact (blue) and conditional (red) likelihood function of an MA(1) process from Figure 6 with n = 30 . The true parameter is β = 0.5.

Example: MA(1) process

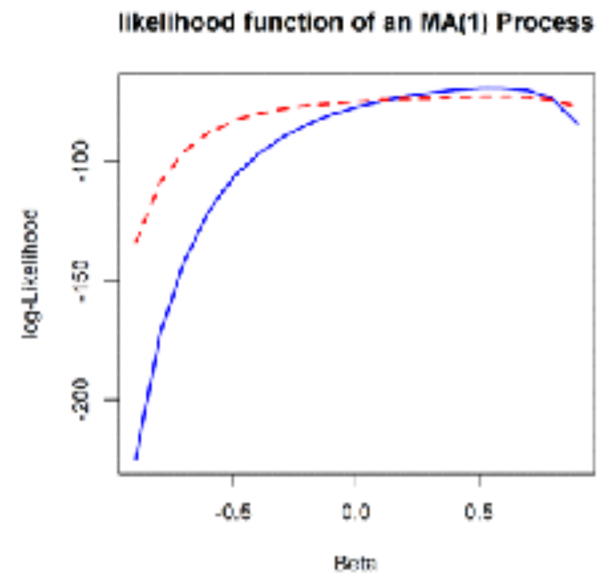


Figure 7a: Exact (blue) and conditional (red) likelihood function of an MA(1) process from Figure 6 with $n = 50$. The true parameter is $\beta = 0.5$.

Example: MA(1) process

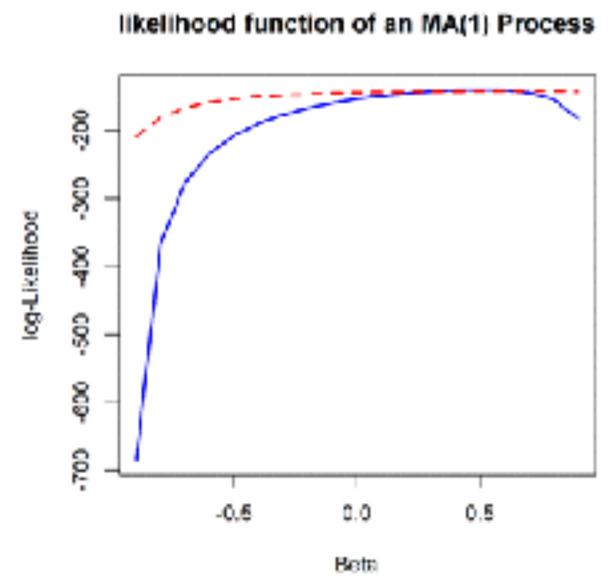


Figure 7a: Exact (blue) and conditional (red) likelihood function of an MA(1) process from Figure 6 with n = 100 . The true parameter is β = 0.5.

Example: MA(1) process

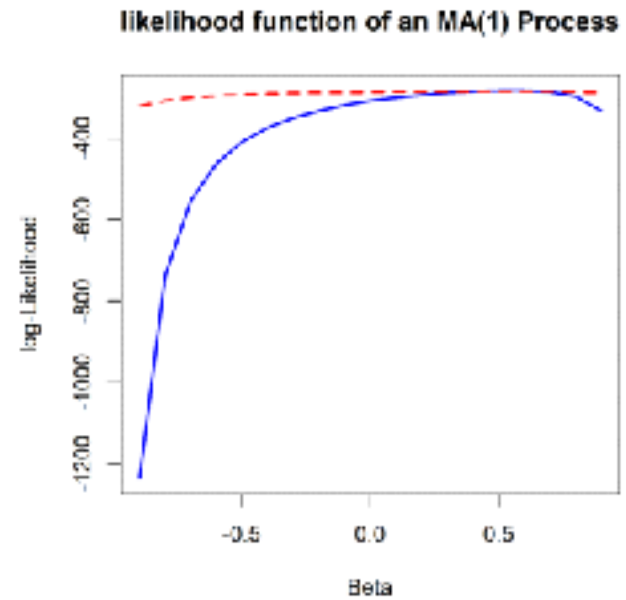


Figure 7a: Exact (blue) and conditional (red) likelihood function of an MA(1) process from Figure 6 with n = 200 . The true parameter is β = 0.5.

Can MA(3) explain SPY log returns?