AR and MA predictions

Predictions with AR and MA models

In this short note, we show how to simulate from AR and MA processes. In addition, we highlight that long-term forecasts of both models are equal to their unconditional expectation. In particular, we shall do the following:

1. Simulate from MA and AR models given the model parameters.
2. Compute short-term and long-term predictions.

We remind that the AR(p) model is given by

$$y_t = \beta_0 + \sum_{i=1}^{p} \beta_i y_{t-i} + \epsilon_t.$$

We set $\epsilon_t \sim i.N(0, \sigma^2)$ in this document. The unconditional expectation for stationary AR models is given by

$$E(y_t) = \frac{\beta_0}{1 - \sum_{i=1}^{p} \beta_i}.$$

Regarding the moving average process, the MA(q) model is specified as

$$y_t = \mu + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t,$$

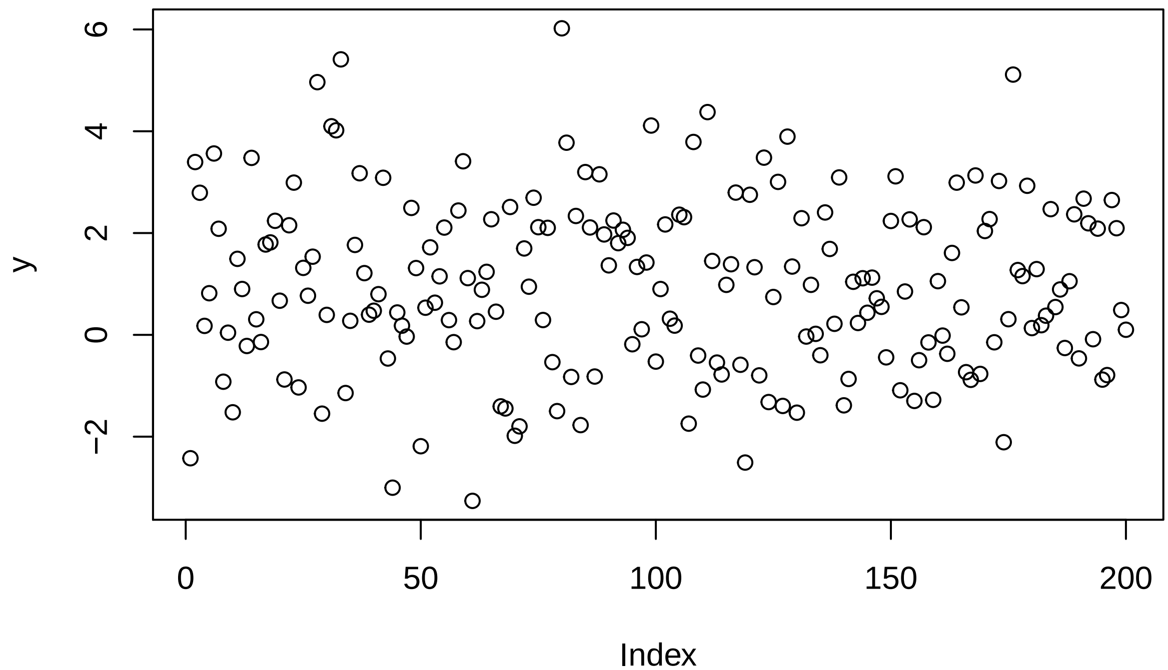
Again, we assume that $\epsilon_t \sim i.N(0, \sigma^2)$. So, our MA(q) models are always strictly stationary. The unconditional expectation is given by

$$E(y_t) = \mu.$$

Simulation and prediction with a MA(q) process

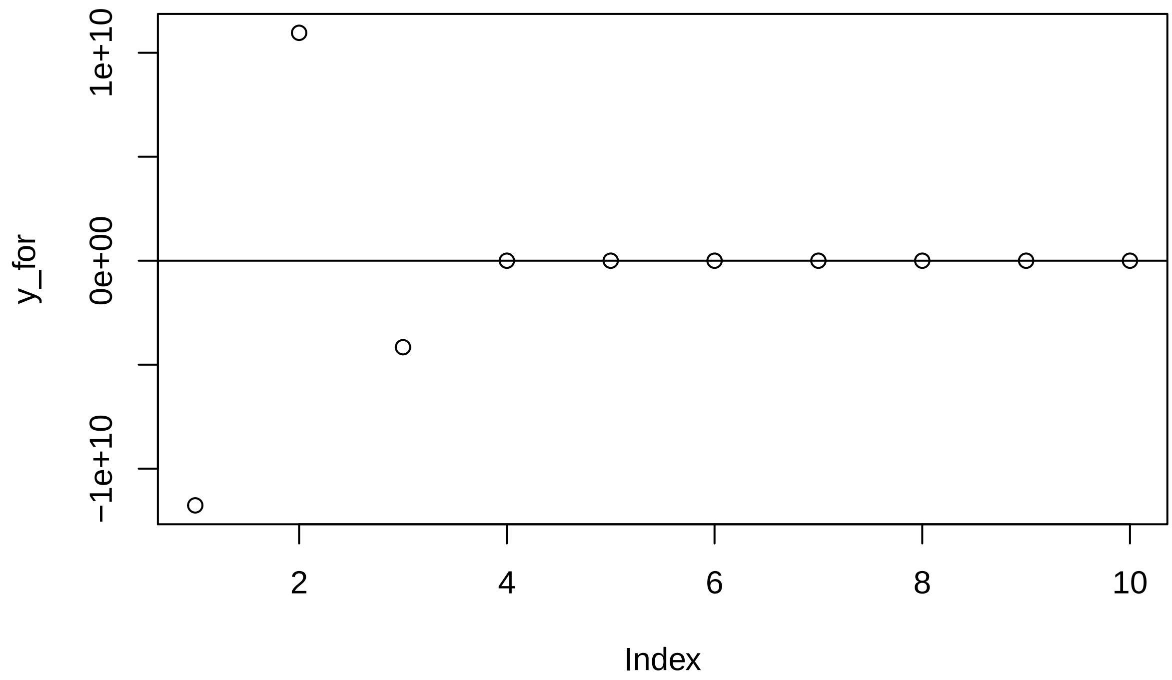
We first simulate from a MA(q) model. We fix the number of lags to three (i.e. $q = 3$).

```
T = 200 # Sample size
mu = 1
theta = c(0.2,-0.7,0.4)
sigma_sq = 2
MA_q = length(theta)
y = numeric(T)
epsilon = rnorm(T)*sqrt(sigma_sq)
y[1:MA_q] = mu + rnorm(MA_q)*sqrt(sigma_sq)
for(t in (MA_q+1):T){
  y[t] = mu + sum(theta*epsilon[(t-1):(t-MA_q)]) + epsilon[t]
}
plot(y)
```



We now predict from the last observation. To do so, we need to recursively compute the error term.

```
## prediction
error_term = numeric(T) ## To do as if we do not know the error term (as in practice)
for (t in (MA_q+1):T){
  error_term[t] = y[t] - mu - sum(theta*error_term[(t-1):(t-MA_q)]) ## MA(q) model must be invertible
}
nb_forecast = 10
y_for = numeric(nb_forecast)
eps_for = error_term[T:(T-MA_q+1)]
for(h in 1:nb_forecast){
    eps_cur = sum(theta*eps_for)
    y_for[h] = mu + eps_cur
    if(h<=MA_q){
      for(i in (MA_q-1):1){
        eps_for[i+1] = eps_for[i]
      }
      eps_for[h] = 0
    }
}
plot(y_for)
abline(a=mu,b=0)
```

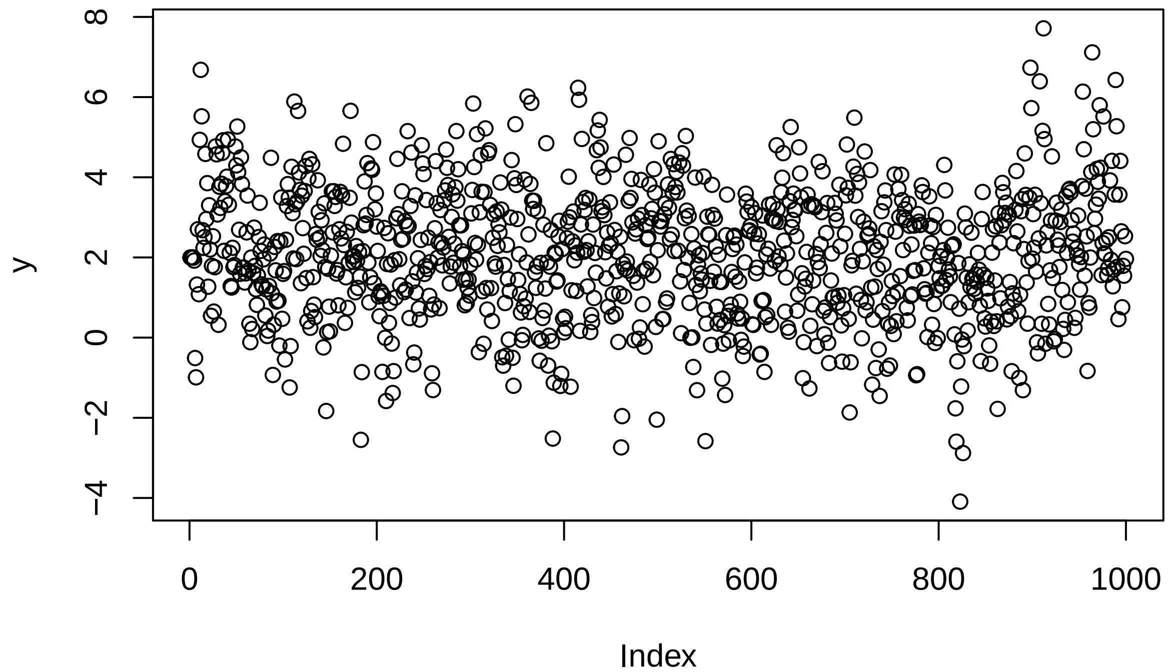


We observe that after MA_q lags, the prediction of the MA model is equal to its unconditional expectation.

Simulation and predictions with an AR(p) model

We now turn to the simulation of an AR(p) model.

```
T = 1000
beta_0 = 1
beta_lags = c(0.6,-0.4,0.3)
AR_p = length(beta_lags)
sigma_sq = 2
y = numeric(T)
unc_mean = beta_0/(1-sum(beta_lags))
y[1:AR_p] = unc_mean
for(t in (AR_p+1):T){
  y[t] = beta_0 + sum(beta_lags*y[(t-1):(t-AR_p)]) + rnorm(1)*sqrt(sigma_sq)
}
plot(y)
```




```
(mean(y))
```

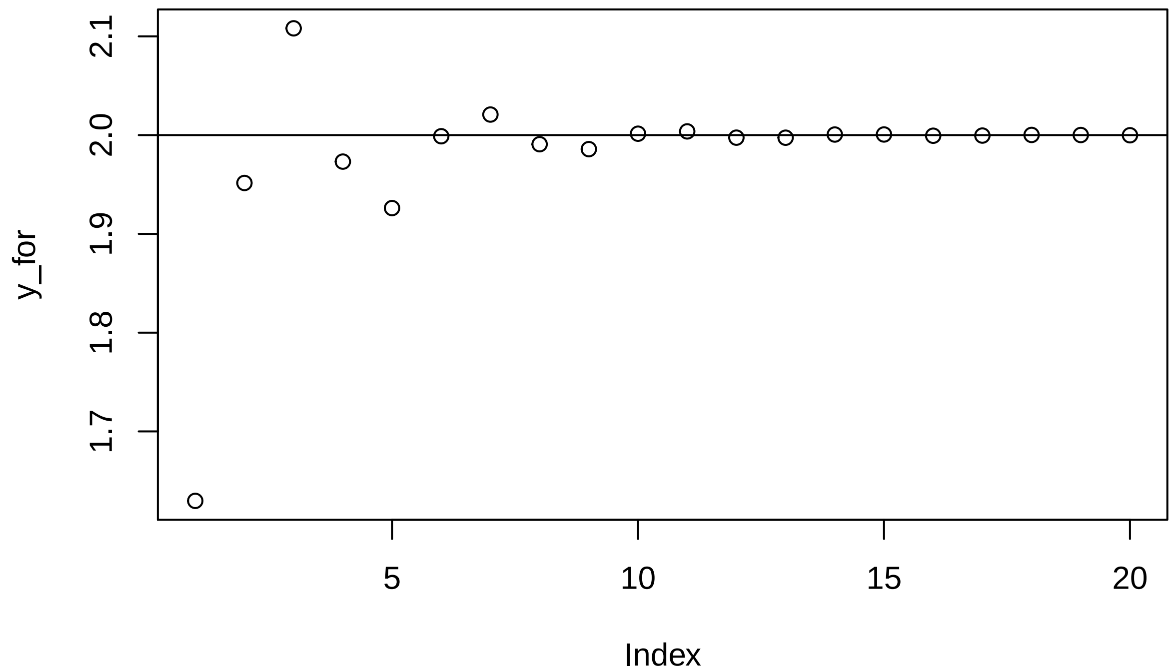
```
## [1] 2.070375
```

```
(unc_mean)
```

```
## [1] 2
```

The unconditional expecation is equal to 2 which is close to the empirical mean of the series given by 2.0703749. We forecast the series.

```
## Predictions with an AR(p) model
nb_forecast = 20
y_for = numeric(nb_forecast)
y_prev = y[T:(T-AR_p+1)]
for(h in 1:nb_forecast){
  y_for[h] = beta_0 + sum(beta_lags*y_prev)
  for(i in (AR_p-1):1){
    y_prev[i+1] = y_prev[i]
  }
  y_prev[1] = y_for[h]
}
plot(y_for)
abline(a=unc_mean,b=0)
```



We observe that the predictions slowly converge to the unconditional expectation of the model.