ARMA Simulation

Aron Hartvig

09/03/2020

Create the function

ARMA processes has two parts, the AR(p) and the MA(q) part. We can simply simulate an ARMA process by multiplying the matrix of coefficients and the lags of Ys and errors.

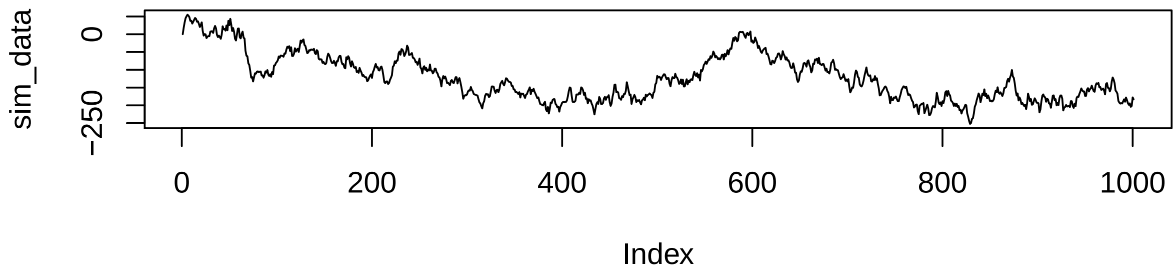
```
ARMASimulation <- function( N = 1000, ar = c( 0 ), ma = c( 0 ), sd ){
  ar_lag <- length( ar ) # p
  ma_lag <- length( ma ) # q
  max_lag <- max( ar_lag, ma_lag )
  sim_data = rep( NA, ( N + max_lag ) )
  # Create empty vector
  sim_data[ 1:max_lag ] <- 0
  # Set first max(p,q) items 0
  epsilon <- rnorm( N + max_lag + 1, mean = 0, sd = sd )
  # Vector of errors

  for( i in (max_lag + 1):(N + max_lag)){
    sim_data[ i ] <- ar %*% sim_data[ c( (i - 1):(i - ar_lag) )] +
      epsilon[ i ] + ma %*% epsilon[ c( (i - 1):(i - ma_lag) )]
  }
  plot( sim_data, type = "l" )
  return( sim_data[ -c( 1:max_lag ) ] )
}
```

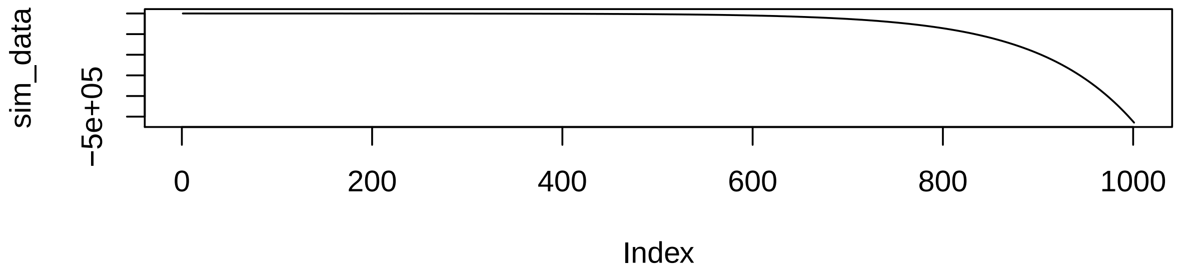
Run the function

Try it with specific values, to see if it works.

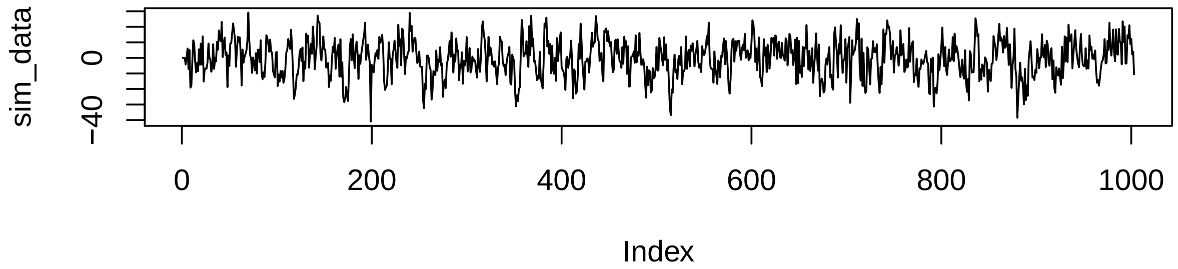
```
N <- 1000
ar <- c( 1 )
sd <- 10
ARMA <- ARMASimulation( N, ar, sd = sd ) # Random walk
```



```
ar <- c( 1.01 )
ARMA <- ARMASimulation( N, ar, sd = sd ) # Exponential
```



```
ar <- c( 0.3, 0.2, 0.02 )
ma <- c( 0.1, 0.03 )
ARMA <- ARMASimulation( N, ar, ma, sd ) # Stationer
```


sim_data
Index