Time Series Lab 1

Yubo Rasmussen

## Time Series Lab 1

In this computer lab we will use simulation to explore two common models for time series moving average (MA) and autoregressive (AR) processes. By generating simulated time series according to the two models, we can examine the statistical properties, and compare them to theoretical results.

## Key R Commands

To explore these processes we will rely on 3 functions from the R stats package. A brief introduction is provided here, but for more information please consult the R Documentation.

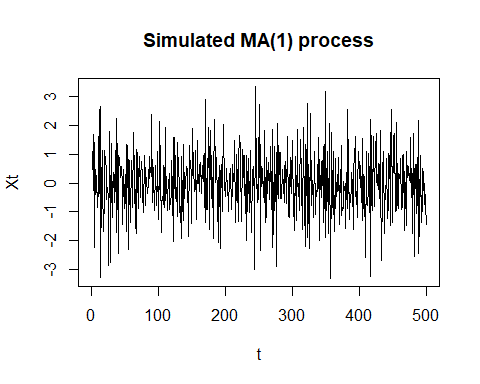
1. To generate data according to MA and AR models, we use the arima.sim() function. For our purposes, the function should be called as arima.sim(model, n) where model is a list containing the AR and/or MA coeffcients, and n is the length of the series to be generated. As a simple example, the following generates a series of length 10 from an AR(2) process:

n <- 10  
ar <- c(0.1, 0.2)  
model <- list(ar = ar)  
generated\_series <- arima.sim(model, n)

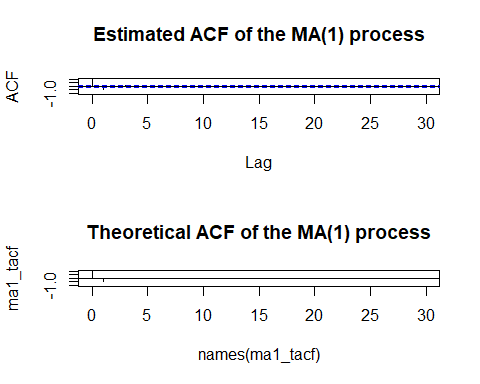
1. To estimate the ACF of an observed time series, the function acf() should be used. For our purposes this should simply be called as acf(series, lag.max=k) where k is the maximum lag for which the ACF should be estimated.
2. Finally, the theoretical ACF of ARMA processes can be computed using the function ARMAacf(). ARMAacf(ar=ar, ma=ma, lag.max=k) where ar and ma are numeric vectors containing the AR and MA coefficients respectively

## Exercises

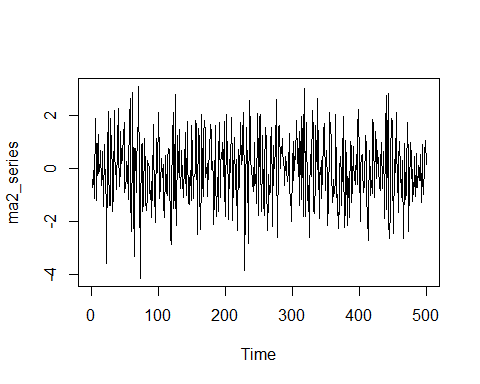
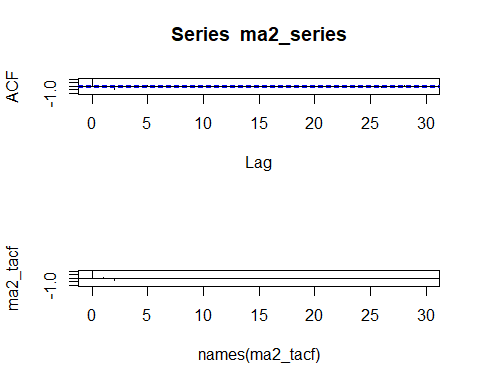
1. Generate a time series of length from an process, where , with coeffcient of your choice. Plot the time series, with an appropriate title and axis labels.

* # MA(p) process  
    
  # Xt = e\_t + phi1 \* e\_(t-1)  
    
    
  n = 500 # Length of series  
  p = 1  
    
  # Define coefficients  
  phi1 = -0.5  
    
  #Define MA model  
  ma1 = c(phi1)  
  ma1\_model = list(ma = ma1)  
    
  # Generate series  
  ma1\_series = arima.sim(model = ma1\_model, n=n)  
    
  # Plot series  
  plot(ma1\_series, xlab="t", ylab="Xt", main=paste0("Simulated MA(", p ,") process"))
* 

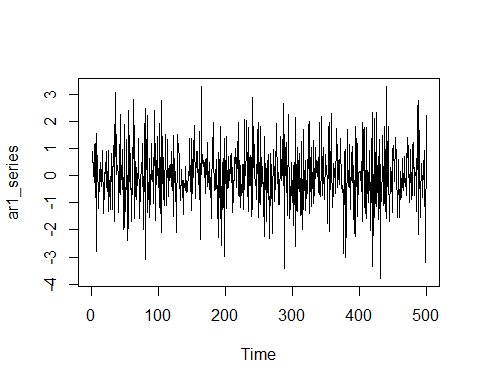
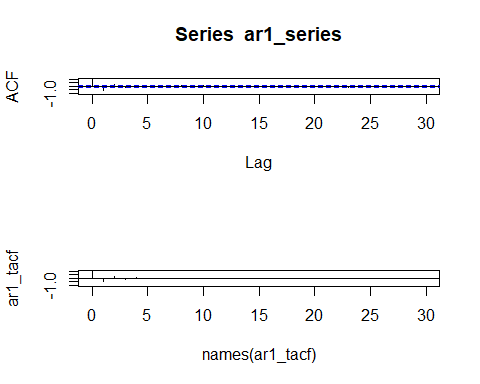
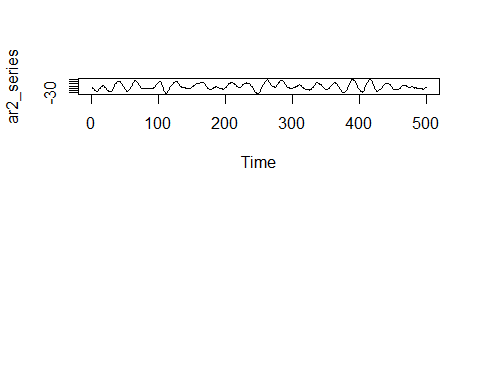
1. Calculate an estimate of the ACF for this data, and compare to the theoretical ACF for the model, with maximum lag . Produce a plot to visually compare the estimated and theoretical ACFs

* # ACF plots  
  # Estimate acf from series  
  ma1\_acf = acf(ma1\_series, lag.max = 30, plot = FALSE)  
  # Theoretical acf  
  ma1\_tacf = ARMAacf(ma = ma1, lag.max=30)  
    
  par(mfrow=c(2,1))  
    
  plot(ma1\_acf, ylim=c(-1,1), main = paste0("Estimated ACF of the MA(", p ,") process"))  
  plot(x=names(ma1\_tacf), y=ma1\_tacf, ylim=c(-1,1), type="h", main = paste0("Theoretical ACF of the MA(", p ,") process"))  
  abline(h=0) # Add line at 0 to tacf plot
* 

1. Explore different values for the coefficient. How do changes affect the data, and ACF?
2. Repeat for and , exploring how the behaviour changes as you alter the coefficients.

* # MA(2)  
    
  p = 2  
    
  phi1 = 0.5  
  phi2 = -0.5 # Observe effect of changing sign of parameters  
    
  ma2 = c(phi1, phi2)  
    
  ma2\_model = list(ma = ma2)  
    
  ma2\_series = arima.sim(model = ma2\_model, n=n)  
    
  plot(ma2\_series)
* 
* ma2\_acf = acf(ma2\_series, lag.max = 30, plot = FALSE)  
  ma2\_tacf = ARMAacf(ma = ma2, lag.max=30)  
    
  par(mfrow=c(2,1))  
    
    
  plot(ma2\_acf, ylim=c(-1,1))  
    
  plot(x=names(ma2\_tacf), y=ma2\_tacf, ylim=c(-1,1), type="h")  
  abline(h=0)
* 

1. Repeat the above for an process. For , explore the behavior observed in different regions of the stationarity triangle by altering coefficient values.

* #AR(1)  
  # X\_t = phi1 \* X\_(t-1) + e\_t  
    
    
  p = 1  
    
  phi1 = -0.5 # -1 < phi1 < 1 for stationarity  
    
    
  ar1 = c(phi1)  
    
  ar1\_model = list(ar = ar1)  
    
  ar1\_series = arima.sim(model = ar1\_model, n=n)  
    
  plot(ar1\_series)
* 
* ar1\_acf = acf(ar1\_series, lag.max = 30, plot = FALSE)  
  ar1\_tacf = ARMAacf(ar = ar1, lag.max=30)  
    
  par(mfrow=c(2,1))  
    
  plot(ar1\_acf, ylim=c(-1,1))  
    
  plot(x=names(ar1\_tacf), y=ar1\_tacf, ylim=c(-1,1), type="h")  
  abline(h=0)
* 
* # AR(2)  
    
  p = 2  
    
  phi1 = 1.9 # -1 < phi2 < 1, phi1 + phi2 < 1, phi2 - phi1 < 1  
  phi2 = -0.95 # Observe behaviour in different regions of stationarity triangle  
    
  ar2 = c(phi1, phi2)  
    
  ar2\_model = list(ar = ar2)  
    
  ar2\_series = arima.sim(model = ar2\_model, n=n)  
    
  plot(ar2\_series)  
    
  ar2\_acf = acf(ar2\_series, lag.max = 30, plot = FALSE)  
  ar2\_tacf = ARMAacf(ar = ar2, lag.max=30)  
    
  par(mfrow=c(2,1))
* 
* plot(ar2\_acf, ylim=c(-1,1))  
    
  plot(x=names(ar2\_tacf), y=ar2\_tacf, ylim=c(-1,1), type="h")  
  abline(h=0)
* 