# Working on MA(2) model

Assuming that x(t) follows a normal distribution with mean 0 and variance 1.

To generate 200 iid values of x(t) we use a random number generator in R. Code is attached with the submission. Below are the results from the code

## a). Plot on y(t) vs t

A picture containing graphical user interface

Description automatically generated

Stationary y(t) with added noise as per MA(2)

## (b). Compute Mean,Variance and ACF and plot ACF vs k

Computed Mean: -0.18374

Computed Variance: 1.150556

The mean and variance are pretty close to the original normal distribution of x(t)

ACF Computed:

[1,] 1.000000000

[2,] 0.358454254

[3,] 0.225643635

[4,] -0.095004746

[5,] 0.015711202

[6,] 0.024451217

[7,] 0.083726535

[8,] -0.007632661

Chart

Description automatically generated

As Visualized, the time series y(t) is dependent till a lag of 2, so y(t) approximately becomes the linear combination of previous two error points in the time series

## (c.) Compute the theoretical ACF value

The ACF of an MA(2) process with coefficients p1 and p2 is given by:

r(k) = E(y(t) \* y(t-k)) = (1 + p1^2 + p2^2) \* sigma^2 \* (delta(k,0) + p1 \* delta(k,1) + p2 \* delta(k,2))

where sigma^2 is the variance of the white noise x(t), and delta(i,j) is the Kronecker delta function (which equals 1 if i=j and 0 otherwise).

Substituting the values of p1 and p2 which were given in the question (i.e., p1 = 1/2 and p2 = 1/3), and assuming that x(t) is a white noise process with variance sigma^2 = 1, we get:

T(k) = (13/18) \* (delta(k,0) + (1/2) \* delta(k,1) + (1/3) \* delta(k,2))

This is the theoretical ACF of the given MA(2) process

## (d) Compare Theoretical ACF with Estimated ACF

Chart, histogram

Description automatically generated

The Theoretical ACF won’t show any dependence after k=2, while Estimated ACF in red does show some dependence but not significant enough to consider as the values don’t cross the threshold of 0.2 unlike k=1 and k=2, showing that the estimated ACF is good enough estimation of the population time series

# Working on AR(1) model

## (a) Compute the ACF of AR(1) model:

The autocorrelation function (ACF) for an AR(1) model can be calculated as follows:

ACF(k) = Corr(y(t), y(t-k))

= Corr(py(t-1) + E(t), py(t-1-k) + E(t-k))

= p^k

where Corr denotes the correlation between two variables, and we have used the fact that the error terms E(t) are independent and identically distributed (iid) with mean 0 and variance σ^2.

Since the ACF only depends on the lag k, and not on the time t, we can write the ACF as a function of k alone. Therefore, the ACF for this AR(1) model is:

ACF(k) = p^k

where h is the lag, p is the autoregressive coefficient (in this case, p is the only parameter of the model), and ACF(k) is the autocorrelation at lag k.

If we take lag(k)=10 with autoregressive coefficient p=0.8,following values of ACF is computed for different lags

1.0000000, 0.8000000, 0.6400000, 0.5120000, 0.4096000, 0.3276800, 0.2621440, 0.2097152, 0.1677722, 0.1342177, 0.1073742

## (b) Generate sample iid for y=1 to 200

To generate the time series in R, we can use the arima.sim() function, which simulates time series data from an ARIMA model. For an AR(1) model, we set the order parameter to c(1,0,0), and the ar parameter to p.

Here we have taken lag(k)=10 with autoregressive coefficient p=0.8

Chart, line chart

Description automatically generated

Generated samples for the AR(1) model with the set parameters

## ( c) Compare Estimated ACF with Theoretical

Chart, histogram

Description automatically generated

The Estimated ACF in red computes lower magnitude of dependence as compared to theoretical and also is only able to show dependence till lag 7