MH4500 Lab 2 Report

Chen Zeyi (U2040386J)

Contents

We first import the library "forecast".

```
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

1 Question 1: "masim" Function

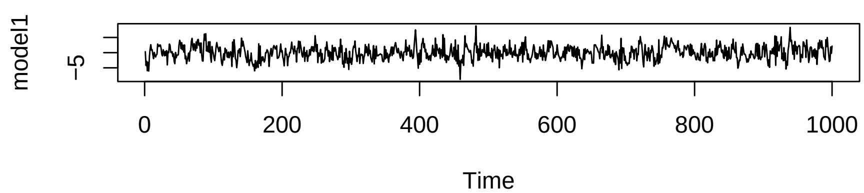
Setting: 1. "para" the coefficient vector. 2. "sigsq" the variance of each noise term. 3. "T" the length of the generated time series.

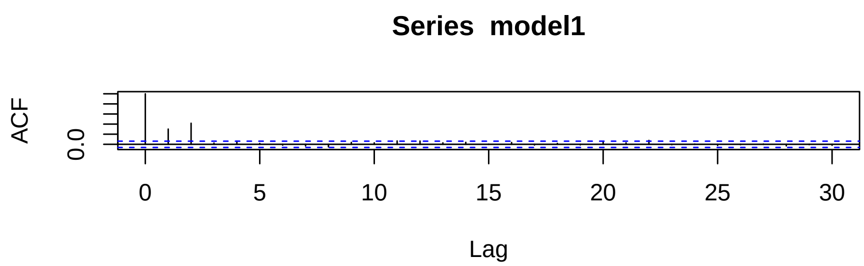
```
masim <- function(para, sigsq, T){
  q = length(para)
  noise = rnorm(T + q, mean = 0, sd = 1)
  x = c(noise[1:q], rep(0, T))
  for (i in (q + 1):(q + T)){
    x[i] =  para %*% noise[i - (1:q)] + noise[i] # Theta0 = 1
  }
  x = x[(q + 1):(q + T)]
  x
}
```

2 Question 2: Simulation

2.1 Model1: para = c(0.5, 2), sigsq = 1, T = 1000

```
set.seed(2022)
par(mfrow= c(2,1))
model1 <- masim(c(0.5, 2), 1, 1000)
plot.ts(model1)
acf(model1)
```



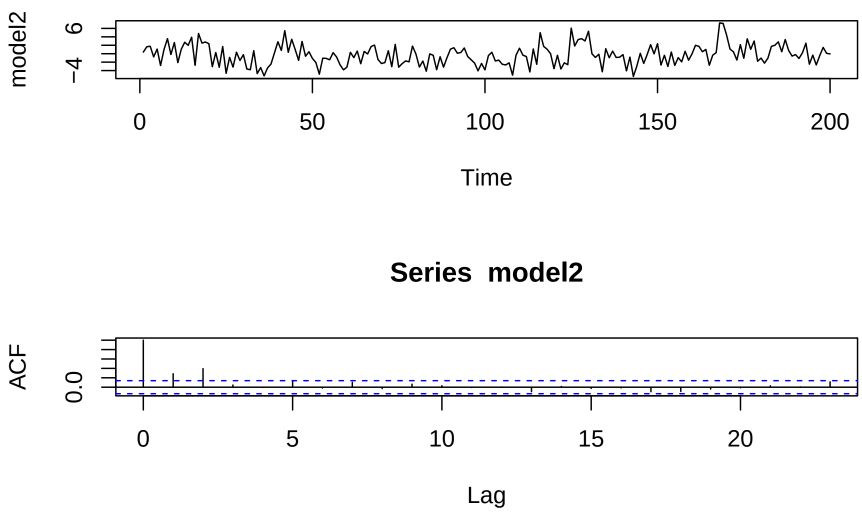


Remark:

1. From ACF, the ACF cuts-off after lag 2, which is consistent with the feature of MA(2).
2. The MA(2) process is stationary, because the ACF cuts-off rapidly and the time plot fluctuates with constant variation around the mean.
3. The vertical line at lag 22 subtly breaks the bound line because there is still 5% probability that the ACF value is outside the 95% confidence interval.

2.2 Model2: para = c(0.5, 2), sigsq = 1, T = 200

```
set.seed(2021)
par(mfrow= c(2,1))
model2 <- masim(c(0.5, 2), 1, 200)
plot.ts(model2)
acf(model2)
```

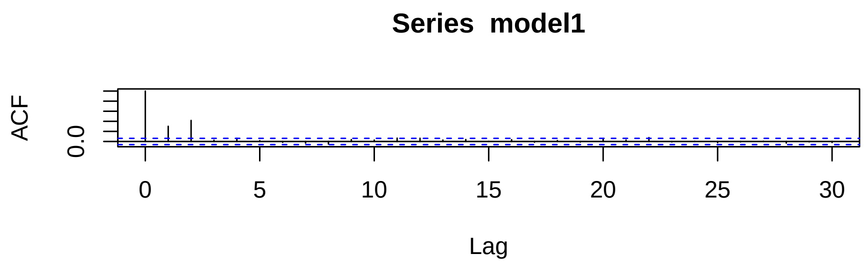


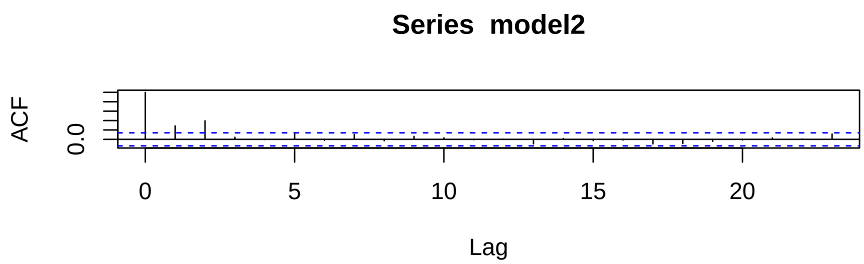
Remark:

1. From ACF, the ACF cuts-off after lag 2, which is consistent with the feature of MA(2).
2. The MA(2) process is stationary, because the ACF cuts-off rapidly and the time plot fluctuates with constant variation around the mean.

2.3 Compare the two models:

```
par(mfrow = c(2, 1))
acf(model1)
acf(model2)
```





Remark:

1. The interval between the dotted blue lines is wider for small T (T = 200) and narrower for large T(T = 1000). This observation can be explained by |r(h)| <= 1.96/sqrt(n), which is smaller for large n.

2. The ACF, generally, is larger for model2 after cut-off, compared to model1.

3 Question 3: ARIMA Model Fitting

We fit the simulated data using ARIMA(0,0,2) model as they are known to be MA(2) process by our simulation algorithm. (Alternatively, one may use auto.arima function to detect ARIMA mode automatically.)

```r
fit1 <- Arima(model1, c(0, 0, 2))
fit1 # The fitted model is X_t = Z_t + 0.2804*Z_t-1 + 0.5356*Z_t-2 + 0.0039
```

```
## Series: model1
## ARIMA(0,0,2) with non-zero mean
##
## Coefficients:
##          ma1     ma2    mean
##       0.2804  0.5356  0.0039
## s.e.  0.0264  0.0283  0.1144
##
## sigma^2 = 3.985:  log likelihood = -2109.05
## AIC=4226.1   AICc=4226.14   BIC=4245.73
```

```r
fit2 <- Arima(model2, c(0, 0, 2))
fit2 # The fitted model is X_t = Z_t + 0.2538*Z_t-1 + 0.5517*Z_t-2 - 0.3154
```

```
## Series: model2
## ARIMA(0,0,2) with non-zero mean
##
## Coefficients:
##          ma1     ma2     mean
##       0.2538  0.5517  -0.3154
## s.e.  0.0597  0.0587   0.2642
##
## sigma^2 = 4.381:  log likelihood = -430.37
## AIC=868.74   AICc=868.95   BIC=881.94
```

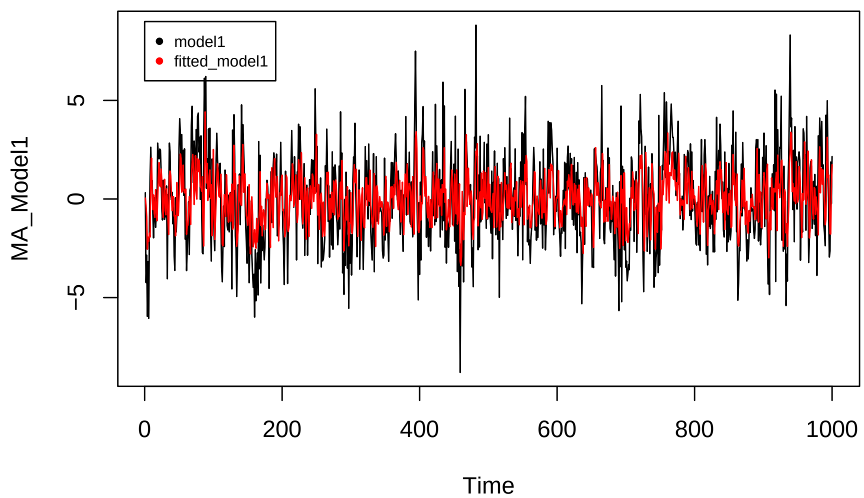
Remark:

The intercept term detected in both fitted models may suggest non-zero mean property of the MA process. Both of them are small enough to be negligible, so we may believe that both models are with zero mean.

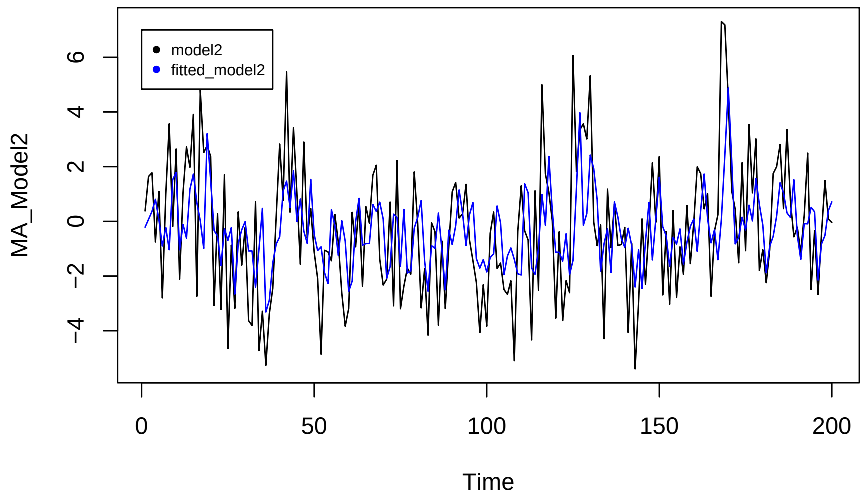
Commpare the fitted model with the original ones:

```r
par(mfrow = c(1, 1))
plot.ts(model1, ylab = "MA_Model1")
```

```r
lines(fitted(fit1), col = "red")
legend(0, 9, legend = c("model1", "fitted_model1"), pch = c(16, 16),
       col = c("black", "red"), cex = 0.7)
```




```r
plot.ts(model2, ylab = "MA_Model2")
lines(fitted(fit2), col = "blue")
legend(0, 7, legend = c("model2", "fitted_model2"), pch = c(16, 16),
       col = c("black", "blue"), cex = 0.7)
```



Store the images into one PDF document:

```
pdf("Lab2_HW_Plots.pdf")
par(mfrow = c(2, 1))
plot.ts(model1)
acf(model1)
plot.ts(model2)
acf(model2)
acf(model1)
acf(model2)
par(mfrow = c(1, 1))
plot.ts(model1, ylab = "MA_Model1")
lines(fitted(fit1), col = "red")
legend(0, 9, legend = c("model1", "fitted_model1"), pch = c(16, 16),
       col = c("black", "red"), cex = 0.7)

plot.ts(model2, ylab = "MA_Model2")
lines(fitted(fit2), col = "blue")
legend(0, 7, legend = c("model2", "fitted_model2"), pch = c(16, 16),
       col = c("black", "blue"), cex = 0.7)
dev.off()
```

```
## pdf
##   2
```

4 Potential Improvement:

1. Some unexpected trends and stationarities may arise due to the randomness of white noise. Though no obvious trend and stationarity are not observed here, there may exist long-term stationary property. Therefore, longer time series can be generated for analysis. Once detecting trend or stationarity, difference operator with lag may be used.

2. One may find non-constant seasonal fluctuation after generating long-term time series. Operations like Box-Cox Transformation may be applied to modify the original data somehow.