Time Series HW # 2

Liam Flaherty

Professor Martin

NCSU: ST546-001

September 13, 2024

1) Consider the time series

$$\textbf{Model 1: } Z_t = a_t + 0.5a_{t-1} + 0.24a_{t-2}$$
$$\textbf{Model 2: } Z_t = 0.8Z_{t-1} + a_t - 0.3a_{t-1}$$

a. Simulate data of lengths 50 and 1000 for the models. Use a burn-in period of length 101 ($t = -100$ to 0) before outputting data from the model.

We can use the `arima.sim()` command from R. The below script gives us our simulation:

```r
#####1. Load Required Packages#####
library(stats)


#####2. Specify Parameters#####
length1=50
length2=1000
burnin=101

#m1=a_t+0.5a_{t-1}+0.24a_{t-2}#
m1coefma=c(0.5, 0.24)      #ARIMA(0,0,2) process#

#m2=0.8Z_{t-1}+a_t-0.3a_{t-1}#
m2coefma=-0.3
m2coefar=0.8               #ARIMA(1,0,1) process#


#####3. Specify Models#####
set.seed(534)               #To make reproducible#
m1_short=arima.sim(model=list(ma=m1coefma),
             n=length1,
             n.start=burn)

m1_long=arima.sim(model=list(ma=m1coef),
                  n=length2,
                  n.start=burn)

m2_short=arima.sim(model=list(ar=m2coefar, ma=m2coefma),
                   n=length1,
                   n.start=burn)

m2_long=arima.sim(model=list(ar=m2coefar, ma=m2coefma),
                   n=length2,
                  n.start=burn)

```

And this code results in output like the below:

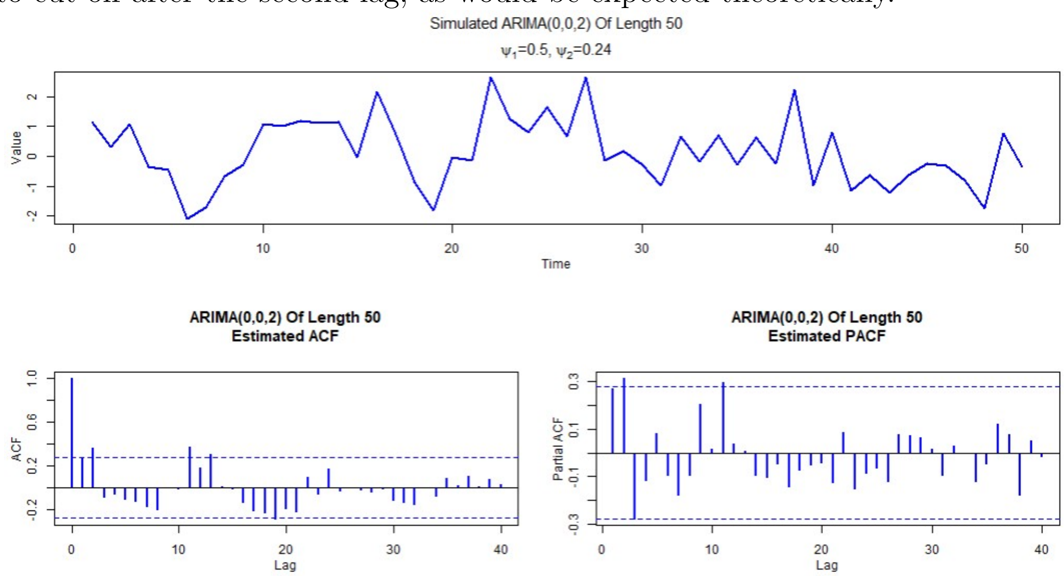
```
> m1_short
Time Series:
Start = 1
End = 50
Frequency = 1
 [1]  0.47869805  0.21060373 -1.18648917 -0.90270106 -1.70181202  1.37558369 -0.43062264 -1.53659288 -1.58523395 -1.05513791  0.08008187
[12] -1.15878363 -0.21750940 -1.05108398  0.53885936 -0.74126957 -2.24613275 -2.11626422 -1.96868241 -0.26191589  1.21242265  0.14918433
[23]  0.89292468  0.10197492 -0.11404904 -1.40130243 -1.76749607 -0.68300524 -0.05418151  1.11855449  1.50300725  2.33500825  1.46505142
[34]  0.60403725 -0.32464004 -0.28893189 -1.35669185 -0.27787101  0.68879004 -0.23163217  0.20071319  1.47539302 -0.45619082  0.38454498
[45] -0.55748997  0.60319607  0.93271552  0.02528023 -0.65376996 -1.54489792
```

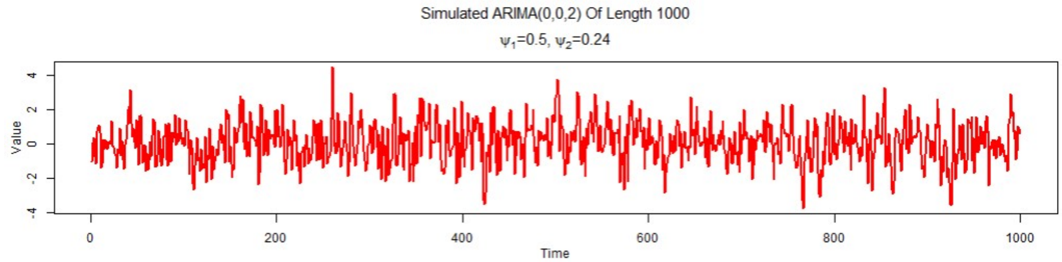
b. Use software to produce plots of the simulated time series, and the estimated autocorrelation (ACF) and partial autocorrelation (PACF) functions.

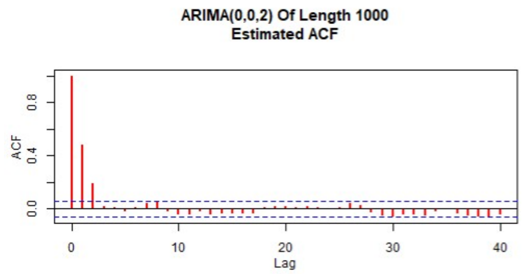
We can plot the simulated data and estimated ACF and PACF functions. The full script is in the appendix, but the gist is below.

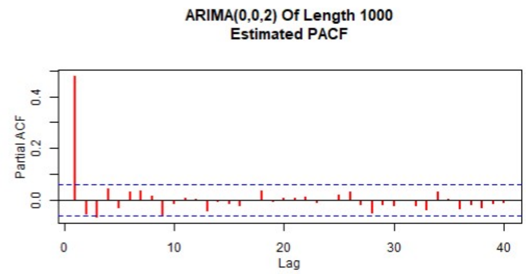
```
#####4. Plot Results Along With ACF and PACF#####
###4a. ARIMA(0,0,2) short###
layout(matrix(c(1,1,2,3), nrow=2, ncol=2, byrow=TRUE))
plot(m1_short,
     main=bquote(atop(
       paste("Simulated ARIMA(0,0,2) Of Length ", .(length1)),
       paste(psi[1], "=", .(m1coefma[1]), ", ", psi[2], "=", .(m1coefma[2])))),
     xlab="Time",
     ylab="Value",
     lwd=2,
     lty=1,
     col="blue")

acf(m1_short, lag.max=40,
    main=paste0("ARIMA(0,0,2) Of Length ", length1, "\n", "Estimated ACF"),
    ci.col="blue",
    col="blue",
    lwd=2)
pacf(m1_short, lag.max=40,
     main=paste0("ARIMA(0,0,2) Of Length ", length1, "\n", "Estimated PACF"),
     ci.col="blue",
     col="blue",
     lwd=2)
```

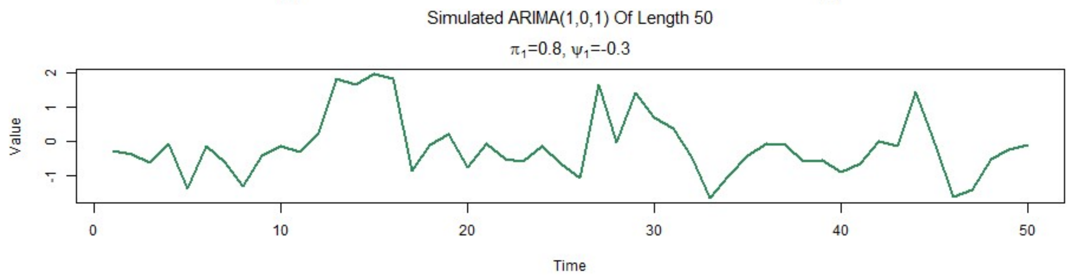
The output for the four models are shown below. Note that with smaller sample sizes, the estimated ACF and PACF is not as reliable. For example, the ACF of the MA(2) model spikes at lags out to 10 when there is only 50 data points, whereas with more data, the ACF seems to cut off after the second lag, as would be expected theoretically.

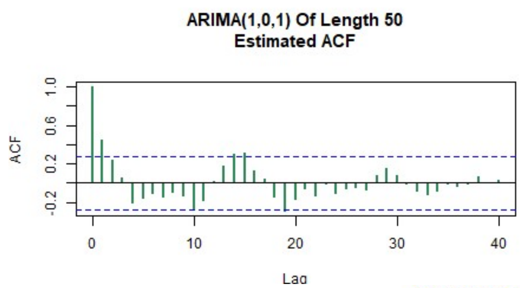


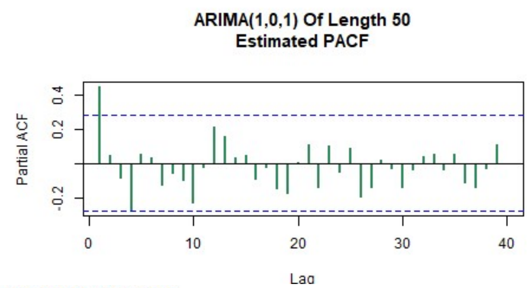

Simulated ARIMA(0,0,2) Of Length 1000
ψ_1=0.5, ψ_2=0.24
Value
Time

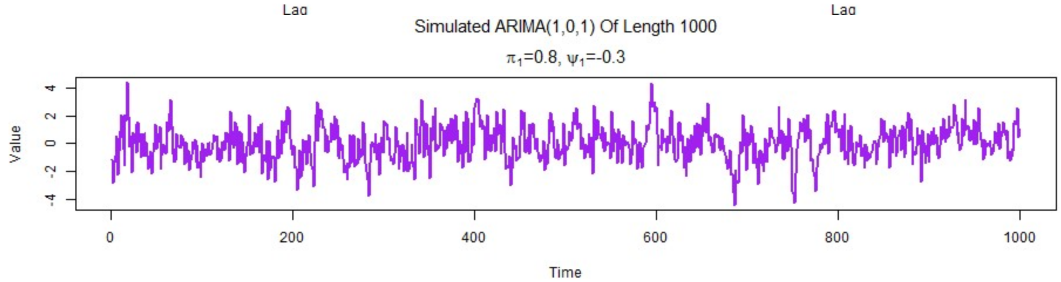

ARIMA(0,0,2) Of Length 1000
Estimated ACF
ACF
Lag

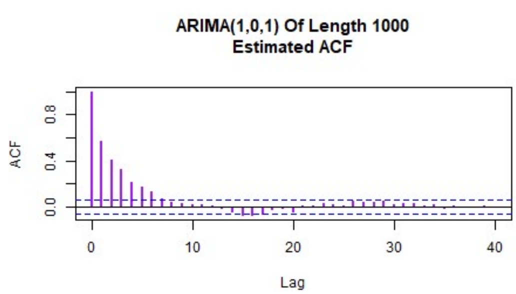

ARIMA(0,0,2) Of Length 1000
Estimated PACF
Partial ACF
Lag

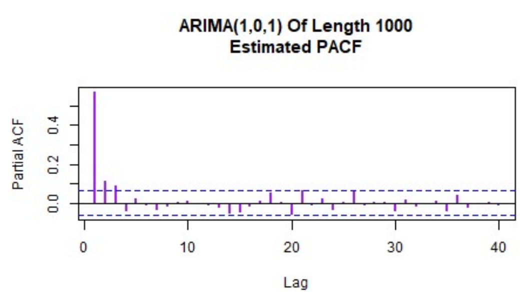

Simulated ARIMA(1,0,1) Of Length 50
π_1=0.8, ψ_1=-0.3
Value
Time


ARIMA(1,0,1) Of Length 50
Estimated ACF
ACF
Lag


ARIMA(1,0,1) Of Length 50
Estimated PACF
Partial ACF
Lag


Simulated ARIMA(1,0,1) Of Length 1000
π_1=0.8, ψ_1=-0.3
Value
Time


ARIMA(1,0,1) Of Length 1000
Estimated ACF
ACF
Lag


ARIMA(1,0,1) Of Length 1000
Estimated PACF
Partial ACF
Lag

c. Show whether or not the model is stationary, and whether or not the model is invertible. If the model is stationary, do the following:

The MA representation of a model, say $\tilde{Z}_t = \sum_{j=0}^{\infty} \psi_j a_{t-j} = \psi(B)a_t$, is stationary when $\sum_{j=0}^{\infty} |\psi_j| < \infty$. It is invertible when the roots of $\psi(B)$ lie outside the unit circle. Similarly, the AR representation of a model, say $\tilde{Z}_t = a_t + \sum_{1=0}^{\infty} \pi_j Z_{t-j} \implies \pi(B)\tilde{Z}_t = a_t$, is invertible when $\sum_{1=0}^{\infty} |\pi_j| < \infty$. It is stationary when the roots of $\pi(B)$ lie outside the unit circle. Logically, an ARMA model $\pi(B)\tilde{Z}_t = \psi(B)a_t$ is stationary when the roots of $\psi(B)$ lie outside the unit circle, and invertible when the roots of $\pi(B)$ lie outside the unit circle.

The first model, $\tilde{Z}_t = a_t + 0.5a_{t-1} + 0.24a_{t-2} = (1 + 0.5B + 0.24B^2)a_t$, is an MA(2) and so is automatically stationary. It is also invertible, since it's roots are $\frac{-0.5 \pm \sqrt{0.5^2 - 4(0.24)(1)}}{2(0.24)}$ or $\frac{-0.5 \pm i\sqrt{0.71}}{0.48} = \frac{-0.5}{0.48} \pm \frac{\sqrt{0.71}}{0.48}i$, and the complex modulus of this is $\sqrt{\left(\frac{-0.5}{0.48}\right)^2 + \left(\frac{\sqrt{0.71}}{0.48}\right)^2}$ or better yet $\sqrt{\frac{0.25}{0.48^2} + \frac{0.71}{0.48^2}}$. The numerator of the second term in the square root is larger than its denominator, so the entire value in the square root is greater than one and thus the entire expression is greater than one.

The second model, $\tilde{Z}_t = 0.8Z_{t-1} + a_t - 0.3a_{t-1} \implies (1 - 0.8B)\tilde{Z}_t = (1 - 0.3B)a_t$, is both stationary and invertible since the roots of $\pi(B)$ and $\psi(B)$ both lie inside the unit circle. To see this, see that the constant in both functions is 1, and the coefficient to the B terms being subtracted are both below 1 (so the root must be above 1).

i. Determine theoretical autocorrelations (the true model values) $\boldsymbol{\rho_1, \rho_2,}$ and $\boldsymbol{\rho_3}$ as well as partial autocorrelations $\boldsymbol{\phi_{11}, \phi_{22},}$ and $\boldsymbol{\phi_{33}}$ for each of the two models.

The covariance function for the MA(2) is given by:

$$\begin{aligned}
\gamma_1 &= \mathbb{E}(\tilde{Z}_t \tilde{Z}_{t-1}) && \text{Definition}\\
&= \mathbb{E}\left((a_t + 0.5a_{t-1} + 0.24a_{t-2})\left(a_{t-1} + 0.5a_{t-2} + 0.24a_{t-3}\right)\right) && \text{Substitution}\\
&= \mathbb{E}(0.5a_{t-1}^2 + (0.24)(0.5)a_{t-2}^2) && \text{Ignore off diagonal terms}\\
&= 0.5\sigma_a^2 + 0.12\sigma_a^2 && \text{Expectations are linear}
\end{aligned}$$

and $\gamma_2 = \mathbb{E}\left((a_t + 0.5a_{t-1} + 0.24a_{t-2})\left(a_{t-2} + 0.5a_{t-3} + 0.24a_{t-4}\right)\right) = \mathbb{E}(0.24a_{t-2}^2) = 0.24\sigma_a^2$. Note that we can ignore the off-diagonal terms since $\mathbb{E}(a_t a_{t+k}) = 0$ when $k \neq 0$ by the definition of white-noise. The variance of model is:

$$\begin{aligned}
\gamma_0 = \mathbb{V}(\tilde{Z}_t) &= \mathbb{V}(a_t + 0.5a_{t-1} + 0.24a_{t-2}) && \text{Substitution}\\
&= \mathbb{V}(a_t) + 0.5^2\mathbb{V}(a_{t-1}) + 0.24^2\mathbb{V}(a_{t-1}) && \text{No covariance between terms}\\
&= \sigma_a^2(1 + 0.5^2 + 0.24^2) && \text{Constant variance assumption}
\end{aligned}$$

In general, the autocorrelations at lag k are $\rho_k = \frac{\gamma_k}{\gamma_0}$ and thus $\rho_1 = \frac{0.5\sigma_a^2 + 0.12\sigma_a^2}{\sigma_a^2(1+0.5^2+0.24^2)} \approx 0.47$, $\rho_2 = \frac{0.24\sigma_a^2}{\sigma_a^2(1+0.5^2+0.24^2)} \approx 0.18$, and, since the process is an MA(2), $\rho_3 = 0$.

In general, we can compute the partial autocorrelations as $\phi_{k,k} = \dfrac{\left|\begin{bmatrix} 1 & \rho_1 & \cdots & \rho_{k-2} & \rho_1 \\ \rho_1 & 1 & \cdots & \rho_{k-3} & \rho_2 \\ \vdots & \vdots & \cdots & \ddots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \cdots & \rho_1 & \rho_k \end{bmatrix}\right|}{\left|\begin{bmatrix} 1 & \rho_1 & \cdots & \rho_{k-2} & \rho_{k-1} \\ \rho_1 & 1 & \cdots & \rho_{k-3} & \rho_{k-2} \\ \vdots & \vdots & \cdots & \ddots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \cdots & \rho_1 & 1 \end{bmatrix}\right|}$.

Here, we have:

$$\phi_{1,1} = \rho_1 \approx 0.47$$

$$\phi_{2,2} = \frac{\left|\begin{bmatrix} 1 & \rho_1 \\ \rho_1 & \rho_2 \end{bmatrix}\right|}{\left|\begin{bmatrix} 1 & \rho_1 \\ \rho_1 & 1 \end{bmatrix}\right|} = \frac{\rho_2 - \rho_1^2}{1 - \rho_1^2} \approx \frac{0.18 - (0.47^2)}{1 - 0.47^2} \approx -0.05$$

$$\phi_{3,3} = \frac{\left|\begin{bmatrix} 1 & \rho_1 & \rho_1 \\ \rho_1 & 1 & \rho_2 \\ \rho_2 & \rho_1 & \rho_3 \end{bmatrix}\right|}{\left|\begin{bmatrix} 1 & \rho_1 & \rho_2 \\ \rho_1 & 1 & \rho_1 \\ \rho_2 & \rho_1 & 1 \end{bmatrix}\right|} = \frac{(1)(\rho_3 - \rho_1\rho_2) - (\rho_1)(\rho_1\rho_3 - \rho_2^2) + (\rho_1)(\rho_1^2 - \rho_2)}{(1)(1 - \rho_1^2) - (\rho_1)(\rho_1 - \rho_1\rho_2) + (\rho_2)(\rho_1^2 - \rho_2)} \approx -0.09$$

We can double check our calculations with the below:

```
> macoef=c(.5,.24)
> round(ARMAacf(ma=macoef, lag.max=5),2)
   0    1    2    3    4    5
1.00 0.47 0.18 0.00 0.00 0.00
> round(ARMAacf(ma=macoef, lag.max=5, pacf=TRUE),2)
[1]  0.47 -0.05 -0.09  0.06 -0.01
```

We compute the theoretical autocovariance of the ARMA(1,1) as follows:

$$
\begin{aligned}
\gamma_k &= \mathbb{E}(\tilde{Z}_t\tilde{Z}_{t-k}) && \text{Definition}\\
&= \mathbb{E}\left(0.8\tilde{Z}_{t-1}\tilde{Z}_{t-k} + \tilde{Z}_{t-k}a_t - 0.3\tilde{Z}_{t-k}a_{t-1}\right) && \text{Multiplying through by } \tilde{Z}_{t-k}\\
&= 0.8\mathbb{E}(\tilde{Z}_{t-1}\tilde{Z}_{t-k}) + \mathbb{E}(\tilde{Z}_{t-k}a_t) - 0.3\mathbb{E}(\tilde{Z}_{t-k}a_{t-1}) && \text{Expectations are linear}\\
&= \begin{cases} 0.8\gamma_0 - 0.3\sigma_a^2, & k=1\\ 0.8\gamma_1, & k=2\\ 0.8\gamma_2, & k=3\end{cases} && \text{Properties of white noise}
\end{aligned}
$$

We can compute the variance as follows:

$$
\begin{aligned}
\mathbb{V}(\tilde{Z}_t) &= \mathbb{V}\left(0.8\tilde{Z}_{t-1} + a_t - 0.3a_{t-1}\right) && \text{Substitution}\\
&= 0.8^2\mathbb{V}(\tilde{Z}_{t-1}) + \sigma_a^2 + 0.3^2\sigma_a^2 + 2\left((0.8)(-0.3)\sigma_a^2)\right)\\
&= \frac{(1+0.3^2-0.48)\sigma_a^2}{1-0.8^2} = \frac{0.61\sigma_a^2}{0.36} && \text{Weak Stationarity}
\end{aligned}
$$

On the second line, we use the general formula for the variance of a linear combination, $\mathbb{V}\left(\sum_{i=1}^n c_iX_i\right) = \sum_{i=1}^n c_i\mathbb{V}(X_i) + 2\sum_{i=1}^n\sum_{j:j>i}^n c_ic_j\text{Cov}(X_i,X_j)$, and the observation that the covariance between the other two random variables in the double sum are zero by the properties of white-noise.

The autocorrelations are $\rho_1 = \frac{\gamma_k}{\gamma_0}$, so $\rho_1 = \frac{0.8\left(\frac{0.61\sigma_a^2}{0.36}\right)-0.3\sigma_a^2}{\frac{0.61\sigma_a^2}{0.36}} = 0.8 - \frac{0.3\cdot 0.36}{.61} \approx 0.62$, $\rho_2 \approx 0.8(0.62) \approx 0.50$, and $\rho_3 \approx 0.8(0.5) \approx 0.4$.

To find the partial autocorrelations, we use the same process as before:

$$
\begin{aligned}
\phi_{1,1} &= \rho_1 \approx 0.62\\
\phi_{2,2} &= \frac{\left|\begin{bmatrix}1 & \rho_1\\ \rho_1 & \rho_2\end{bmatrix}\right|}{\left|\begin{bmatrix}1 & \rho_1\\ \rho_1 & 1\end{bmatrix}\right|} = \frac{\rho_2-\rho_1^2}{1-\rho_1^2} \approx \frac{0.5-(0.62^2)}{1-0.62^2} \approx 0.18\\
\phi_{3,3} &= \frac{\left|\begin{bmatrix}1 & \rho_1 & \rho_1\\ \rho_1 & 1 & \rho_2\\ \rho_2 & \rho_1 & \rho_3\end{bmatrix}\right|}{\left|\begin{bmatrix}1 & \rho_1 & \rho_2\\ \rho_1 & 1 & \rho_1\\ \rho_2 & \rho_1 & 1\end{bmatrix}\right|} = \frac{(1)(\rho_3-\rho_1\rho_2)-(\rho_1)(\rho_1\rho_3-\rho_2^2)+(\rho_1)(\rho_1^2-\rho_2)}{(1)(1-\rho_1^2)-(\rho_1)(\rho_1-\rho_1\rho_2)+(\rho_2)(\rho_1^2-\rho_2)} \approx 0.05
\end{aligned}
$$

We verify our calculations with the below:

```
> ARmacoef=0.8
> arMAcoef=-0.3
> arma11acf=round(ARMAacf(ar=ARmacoef, ma=arMAcoef, lag.max=10),2)
> arma11pacf=round(ARMAacf(ar=ARmacoef, ma=arMAcoef, lag.max=10, pacf=TRUE),2)
> arma11acf
   0    1    2    3    4    5    6    7    8    9   10
1.00 0.62 0.50 0.40 0.32 0.26 0.20 0.16 0.13 0.10 0.08
> arma11pacf
 [1] 0.62 0.18 0.05 0.02 0.00 0.00 0.00 0.00 0.00 0.00
```

ii. Discuss what you observe as far as the proximity of the estimated autocorrelations values to the true values.

We see that with an increased sample, the estimated ACF moves closer to the theoretical ACF. The comparison between the two models at the two different sample sizes are shown below.

```
> ######6. Comparison#####
> acf_comparison=data.frame(
+   mylag=1:10,
+   Estimated_MA_50=as.numeric(m1_shortacf[1:10]$acf),
+   Estimaged_MA_1000=as.numeric(m1_longacf[1:10]$acf),
+   Theoretical_MA=as.numeric(ma2acf[2:11]),
+   Estimated_ARMA_50=as.numeric(m2_shortacf[1:10]$acf),
+   Estimaged_ARMA_1000=as.numeric(m2_longacf[1:10]$acf),
+   Theoretical_ARMA=as.numeric(arma11acf[2:11])
+ )
>
> round(acf_comparison,2)
   mylag Estimated_MA_50 Estimaged_MA_1000 Theoretical_MA Estimated_ARMA_50 Estimaged_ARMA_1000 Theoretical_ARMA
1      1            0.27              0.48           0.47              0.45                0.57             0.62
2      2            0.36              0.19           0.18              0.23                0.40             0.50
3      3           -0.08              0.02           0.00              0.05                0.33             0.40
4      4           -0.05              0.01           0.00             -0.21                0.22             0.32
5      5           -0.11             -0.01           0.00             -0.15                0.17             0.26
6      6           -0.12              0.02           0.00             -0.10                0.12             0.20
7      7           -0.17              0.04           0.00             -0.15                0.07             0.16
8      8           -0.20              0.05           0.00             -0.09                0.04             0.13
9      9            0.00             -0.01           0.00             -0.13                0.02             0.10
10    10            0.00             -0.04           0.00             -0.27                0.02             0.08
```

2) Let $\tilde{Z}_t = 0.4\tilde{Z}_{t-1} + 0.21\tilde{Z}_{t-2} + a_t + 0.7a_{t-1} + 0.12a_{t-2}$. Is the model in its reduced form? If not, write the model in reduced form.

The above model can be written as $\tilde{Z}_t - 0.4\tilde{Z}_{t-1} - 0.21\tilde{Z}_{t-2} = a_t + 0.7a_{t-1} + 0.12a_{t-2}$ or equivalently $(1 - 0.4B - 0.21B^2)\,\tilde{Z}_t = (1 + 0.7B + 0.12B^2)\,a_t$.

We can compute the roots of $\pi(B) = (1 - 0.4B - 0.21B^2)$ as:

$$\frac{-(-0.4) \pm \sqrt{(-0.4^2) - 4(-0.21)(1)}}{2(-0.21)} = \frac{0.4 \pm i\sqrt{0.68}}{-0.42}$$

And the roots of $\psi(B) = (1 + 0.7B + 0.12B^2)$ as:

$$\frac{-(0.7) \pm \sqrt{(0.7^2) - 4(0.12)(1)}}{2(.12)} = \frac{-0.7 \pm \sqrt{.01}}{0.24} = \frac{\frac{-7}{10} \pm \frac{1}{10}}{0.24} = \left(\frac{-5}{2}, \frac{-10}{3}\right)$$

In general, we can tell if a model is in reduced form if there are no common roots in the $\psi(B)$ and $\pi(B)$ polynomials. Since the constants in both functions above are 1, the roots completely determine the factoring. And by the above, there are no common roots, and therefore no common factors; the model is in reduced form.

3) For the following two time series models, determine if $W_t = (1 - B)\tilde{Z}_t$ is stationary and if it is invertible. $(1 - B)\tilde{Z}_t = a_t - a_{t-1}$ and $(1 - B)^2\tilde{Z}_t = a_t - 0.81a_{t-1} + 0.38a_{t-2}$.

The first model is $W_t = (1 - B)\tilde{Z}_t = a_t - a_{t-1}$, which can be written $W_t = \psi(B)a_t = (1-B)a_t$. This is an MA(2), and since all finite MA processes are stationary, W_t is stationary. The process is invertible if the roots of $\psi(B)$ lie outside the unit circle. Since the root is 1, the process is not invertible.

The second model is $W_t = (1 - B)\tilde{Z}_t = (a_t - 0.81a_{t-1} + 0.38a_{t-2})\,(1 - B)^{-1}$ which can be written $W_t = \psi(B)a_t = \frac{(1-0.81B+0.38B^2)}{(1-B)}a_t$. The denominator has a root of 1, so $\psi(B)$ is infinite in extent and fails to be absolutely summable; W_t is non-stationary. The function $\psi(B)$ has a root at $B = x$ if and only if the numerator of the function has a root at the same $B = x$. So we can use the quadratic formula and identify the roots as $\frac{0.81 \pm \sqrt{0.81^2 - 4(.38)(1)}}{2(.38)} = \frac{0.81 \pm i\sqrt{0.8639}}{0.76}$. The complex modulus is then $\sqrt{\left(\frac{0.81}{0.76}\right)^2 + \left(\frac{.8639}{0.76^2}\right)} > \sqrt{\frac{.8639}{0.76^2}} > 1$, and so the process is invertible.

1 Appendix

```
#################Written By Liam Flaherty For ST534################################
#####1. Load Required Packages#####
library(stats)


#####2. Specify Parameters#####
length1=50
length2=1000
burnin=101

#m1=a_t+0.5a_{t-1}+0.24a_{t-2}#
m1coefma=c(0.5, 0.24)       #ARIMA(0,0,2) process#

#m2=0.8Z_{t-1}+a_t-0.3a_{t-1}#
m2coefma=-0.3
m2coefar=0.8                #ARIMA(1,0,1) process#


#####3. Specify Models#####
set.seed(534)                #To make reproducible#
m1_short=arima.sim(model=list(ma=m1coefma),
             n=length1,
             n.start=burn)

m1_long=arima.sim(model=list(ma=m1coef),
                  n=length2,
                  n.start=burn)

m2_short=arima.sim(model=list(ar=m2coefar, ma=m2coefma),
                   n=length1,
m2_short=arima.sim(model=list(ar=m2coefar, ma=m2coefma),
                   n=length1,
                   n.start=burn)

m2_long=arima.sim(model=list(ar=m2coefar, ma=m2coefma),
                   n=length2,
                  n.start=burn)




#####4. Plot Results Along With ACF and PACF#####
###4a. ARIMA(0,0,2) short###
layout(matrix(c(1,1,2,3), nrow=2, ncol=2, byrow=TRUE))
plot(m1_short,
     main=bquote(atop(
       paste("Simulated ARIMA(0,0,2) Of Length ", .(length1)),
       paste(psi[1], "=", .(m1coefma[1]), ", ", psi[2], "=", .(m1coefma[2])))),
     xlab="Time",
     ylab="Value",
     lwd=2,
     lty=1,
     col="blue")

m1_shortacf=acf(m1_short, lag.max=40,
    main=paste0("ARIMA(0,0,2) Of Length ", length1, "\n", "Estimated ACF"),
    ci.col="blue",
    col="blue",
    lwd=2)
m1_shortacf
```

```
m1_shortpacf=pacf(m1_short, lag.max=40,
     main=paste0("ARIMA(0,0,2) Of Length ", length1, "\n", "Estimated PACF"),
     ci.col="blue",
     col="blue",
     lwd=2)
m1_shortpacf



###4b. ARIMA(0,0,2) long###
layout(matrix(c(1,1,2,3), nrow=2, ncol=2, byrow=TRUE))
plot(m1_long,
     main=bquote(atop(
       paste("Simulated ARIMA(0,0,2) Of Length ", .(length2)),
       paste(psi[1], "=", .(m1coefma[1]), ", ", psi[2], "=", .(m1coefma[2])))),
     xlab="Time",
     ylab="Value",
     lwd=2,
     lty=1,
     col="red")

m1_longacf=acf(m1_long, lag.max=40,
    main=paste0("ARIMA(0,0,2) Of Length ", length2, "\n", "Estimated ACF"),
    ci.col="blue",
    col="red",
    lwd=2)
m1_longacf


m1_longpacf=pacf(m1_long, lag.max=40,
m1_longpacf=pacf(m1_long, lag.max=40,
     main=paste0("ARIMA(0,0,2) Of Length ", length2, "\n", "Estimated PACF"),
     ci.col="blue",
     col="red",
     lwd=2)
m1_longpacf



###4c. ARIMA(1,0,1) Short###
layout(matrix(c(1,1,2,3), nrow=2, ncol=2, byrow=TRUE))
plot(m2_short,
     main=bquote(atop(
       paste("Simulated ARIMA(1,0,1) Of Length ", .(length1)),
       paste(pi[1], "=", .(m2coefar[1]), ", ", psi[1], "=", .(m2coefma[1])))),
     xlab="Time",
     ylab="Value",
     lwd=2,
     lty=1,
     col="seagreen4")

m2_shortacf=acf(m2_short, lag.max=40,
    main=paste0("ARIMA(1,0,1) Of Length ", length1, "\n", "Estimated ACF"),
    ci.col="blue",
    col="seagreen4",
    lwd=2)
m2_shortacf

m2_shortpacf=pacf(m2_short, lag.max=40,
     main=paste0("ARIMA(1,0,1) Of Length ", length1, "\n", "Estimated PACF"),
     ci.col="blue",
```

```
      ci.col="blue",
      col="seagreen4",
      lwd=2)
m2_shortpacf



###4d. ARIMA(1,0,1) long###
plot(m2_long,
      main=bquote(atop(
        paste("Simulated ARIMA(1,0,1) of Length ", .(length2)),
        paste(pi[1], "=", .(m2coefar[1]), ", ", psi[1], "=", .(m2coefma[1])))),
      xlab="Time",
      ylab="Value",
      lwd=2,
      lty=1,
      col="purple")

m2_longacf=acf(m2_long, lag.max=40,
     main=paste0("ARIMA(1,0,1) of Length ", length2, "\n", "Estimated ACF"),
     ci.col="blue",
     col="purple",
     lwd=2)
m2_longacf

m2_longpacf=pacf(m2_long, lag.max=40,
      main=paste0("ARIMA(1,0,1) of Length ", length2, "\n", "Estimated PACF"),
      ci.col="blue",
      col="purple",
      lwd=2)
m2_longpacf
```

m2_longpacf

```




######5. Theoretical P/ACF######
macoef=c(.5,.24)
ma2acf=round(ARMAacf(ma=macoef, lag.max=10),2)
ma2pacf=round(ARMAacf(ma=macoef, lag.max=10, pacf=TRUE),2)
ma2acf
ma2pacf

ARmacoef=0.8
arMAcoef=-0.3
arma11acf=round(ARMAacf(ar=ARmacoef, ma=arMAcoef, lag.max=10),2)
arma11pacf=round(ARMAacf(ar=ARmacoef, ma=arMAcoef, lag.max=10, pacf=TRUE),2)
arma11acf
arma11pacf





######6. Comparison#####
acf_comparison=data.frame(
  mylag=1:10,
  Estimated_MA_50=as.numeric(m1_shortacf[1:10]$acf),
  Estimaged_MA_1000=as.numeric(m1_longacf[1:10]$acf),
  Theoretical_MA=as.numeric(ma2acf[2:11]),
  Estimated_ARMA_50=as.numeric(m2_shortacf[1:10]$acf),
```

```
######6. Comparison#####
acf_comparison=data.frame(
  mylag=1:10,
  Estimated_MA_50=as.numeric(m1_shortacf[1:10]$acf),
  Estimaged_MA_1000=as.numeric(m1_longacf[1:10]$acf),
  Theoretical_MA=as.numeric(ma2acf[2:11]),
  Estimated_ARMA_50=as.numeric(m2_shortacf[1:10]$acf),
  Estimaged_ARMA_1000=as.numeric(m2_longacf[1:10]$acf),
  Theoretical_ARMA=as.numeric(arma11acf[2:11])
)

round(acf_comparison,4)
```