20217334-MATH3026-CW1

Zihang Yu

2023-3-10

Q1(a)

$\{X_t : t \in \mathbb{Z}\}$ is a ARMA(2,1) process

Q1(b)

$\{X_t : t \in \mathbb{Z}\}$ is a (weakly) stationary process.

Since

$$X_t = \frac{5}{6}X_{t-1} - \frac{1}{6}X_{t-2} + Z_t + \frac{1}{4}Z_{t-1}$$

$$X_t - \frac{5}{6}BX_t + \frac{1}{6}B^2X_t = Z_t + \frac{1}{4}BZ_t$$

Where B is the backward shift operator

$$X_t(1 - \frac{5}{6}B + \frac{1}{6}B^2) = Z_t(1 + \frac{1}{4}B)$$

Thus,

$$\phi(B) = (1 - \frac{5}{6}B + \frac{1}{6}B^2)$$
$$\theta(B) = (1 + \frac{1}{4}B)$$

Solve,

$$\phi(B) = (1 - \frac{5}{6}B + \frac{1}{6}B^2) = 0$$

We get,

$$\frac{1}{6}(B^2 - 5B + 6) = 0$$

$$\frac{1}{6}(B-2)(B-3) = 0$$

Thus the roots of $\phi(B)$ are $B = 2$ or $B = 3$. Which are all lie outside the unit circle. From the lecture notes we know that if the roots of $\phi(B)$ lie outside the unit circle then the ARMA(2,1) process $\{X_t : t \in \mathbb{Z}\}$ is a (weakly) stationary process.

Q1(c)

From Q1(b) we get,

$$\theta(B) = (1 + \frac{1}{4}B)$$

From the lecture notes we know that the ARMA(2,1) process is invertable if the roots of $\theta(B) = (1 + \frac{1}{4}B) = 0$ lie outside the unit circle.

Solve,

$$\theta(B) = (1 + \frac{1}{4}B) = 0$$

We get, B = -4 which lie outside the unit circle, thus the ARMA(2,1) process $\{X_t : t \in \mathbb{Z}\}$ is invertable.

Q1(d)

The $MA(\infty)$ form is $X_t = \sum_{j=0}^{\infty} \alpha_j Z_{t-j}$

Rearranging X_t we get,

$$(1 - \frac{5}{6}B + \frac{1}{6}B^2)X_t = (1 + \frac{1}{4}B)Z_t$$

$$(\frac{1}{2}B - 1)(\frac{1}{3}B - 1)X_t = (1 + \frac{1}{4}B)Z_t$$

$$X_t = \frac{(1+\frac{1}{4}B)}{(1-\frac{1}{2}B)(1-\frac{1}{3}B)}Z_t$$

Let,

$$\frac{(1+\frac{1}{4}B)}{(1-\frac{1}{2}B)(1-\frac{1}{3}B)}Z_t = (\frac{A}{(1-\frac{1}{2}B)} + \frac{C}{(1-\frac{1}{3}B)})Z_t$$

We get,

$$A = \frac{9}{2},\ C = -\frac{7}{2}$$

Thus,

$$X_t = (\frac{\frac{9}{2}}{(1-\frac{1}{2}B)} - \frac{\frac{7}{2}}{(1-\frac{1}{3}B)})Z_t$$

Take Taylor expansions at 0:

$$(1 - \frac{1}{2}B)^{-1} = \sum_{j=0}^{\infty}(\frac{1}{2})^j B^j$$

$$(1 - \frac{1}{3}B)^{-1} = \sum_{j=0}^{\infty}(\frac{1}{3})^j B^j$$

Thus,

$$X_t = (\frac{9}{2}\sum_{j=0}^{\infty}(\frac{1}{2})^j B^j - \frac{7}{2}\sum_{j=0}^{\infty}(\frac{1}{3})^j B^j)Z_t$$

$$X_t = (\frac{9}{2}\sum_{j=0}^{\infty}(\frac{1}{2})^j - \frac{7}{2}\sum_{j=0}^{\infty}(\frac{1}{3})^j)Z_{t-j}$$

$$X_t = (\sum_{j=0}^{\infty} \frac{9}{2}(\frac{1}{2})^j - \frac{7}{2}(\frac{1}{3})^j)Z_{t-j}$$

Above all, the process in $MA(\infty)$ form is:

$$X_t = \sum_{j=0}^{\infty} \alpha_j Z_{t-j}$$

where,

$$\alpha_j = \frac{9}{2}(\frac{1}{2})^j - \frac{7}{2}(\frac{1}{3})^j \ ,$$

Q1(e)

$$X_t = \frac{5}{6}X_{t-1} - \frac{1}{6}X_{t-2} + Z_t + \frac{1}{4}Z_{t-1} \quad (1)$$

Multiplying both sides of (1) by Z_t and taking expectations,

$$E(X_t Z_t) = \frac{5}{6}E(X_{t-1}Z_t) - \frac{1}{6}E(X_{t-2}Z_t) + E(Z_t^2) + \frac{1}{4}E(Z_{t-1}Z_t)$$

By causality assumption $E(X_{t-1}Z_t)=0,\ E(X_{t-2}Z_t)=0$
Since Z_{t-1} and Z_t are independent, $E(Z_{t-1}Z_t)=0$

we get,

$$E(X_tZ_t)=\sigma_z^2 \ \ (2)$$

Multiplying both sides of (1) by X_t and taking expectations,

$$E(X_t^2)=\frac{5}{6}E(X_{t-1}X_t)-\frac{1}{6}E(X_{t-2}X_t)+E(Z_tX_t)+\frac{1}{4}E(Z_{t-1}X_t)$$

We get,

$$\gamma(0)=\frac{5}{6}\gamma(1)-\frac{1}{6}\gamma(2)+\sigma_z^2+\frac{1}{4}E(X_tZ_{t-1}) \ \ (3)$$

Multiplying both sides of (1) by Z_{t-1} and taking expectations,

$$E(X_tZ_{t-1})=\frac{5}{6}E(X_{t-1}Z_{t-1})-\frac{1}{6}E(X_{t-2}Z_{t-1})+E(Z_tZ_{t-1})+\frac{1}{4}E(Z_{t-1}^2)$$

Thus,

$$E(Z_{t-1}X_t)=\frac{5}{6}\sigma_z^2+\frac{1}{4}\sigma_z^2=\frac{13}{12}\sigma_z^2 \ \ (4)$$

Substitude (4) into (3) we get,

$$\gamma(0)=\frac{5}{6}\gamma(1)-\frac{1}{6}\gamma(2)+\frac{61}{48}\sigma_z^2 \ \ (5)$$

Multiplying both sides of (1) by X_{t-1} and taking expectations,

$$E(X_tX_{t-1})=\frac{5}{6}E(X_{t-1}X_{t-1})-\frac{1}{6}E(X_{t-2}X_{t-1})+E(Z_tX_{t-1})+\frac{1}{4}E(Z_{t-1}X_{t-1})$$

We get,

$$\gamma(1)=\frac{5}{6}\gamma(0)-\frac{1}{6}\gamma(1)+\frac{1}{4}\sigma_z^2 \ \ (6)$$

Multiplying both sides of (1) by X_{t-2} and taking expectations,

$$E(X_tX_{t-2})=\frac{5}{6}E(X_{t-1}X_{t-2})-\frac{1}{6}E(X_{t-2}^2)+E(Z_tX_{t-2})+\frac{1}{4}E(Z_{t-1}X_{t-2})$$

We get,

$$\gamma(2)=\frac{5}{6}\gamma(1)-\frac{1}{6}\gamma(0) \ \ (7)$$

By solving (5), (6), (7)

$$\gamma(0)=\frac{477}{160}\sigma_z^2$$

$$\gamma(1)=\frac{75}{32}\sigma_z^2$$

Multiplying both sides of (1) by X_{t-h} and taking expectations,

$$E(X_tX_{t-h})=\frac{5}{6}E(X_{t-1}X_{t-h})-\frac{1}{6}E(X_{t-2}X_{t-h})+E(Z_tX_{t-h})+\frac{1}{4}E(Z_{t-1}X_{t-h})$$

We get,

$$\gamma(h)=\frac{5}{6}\gamma(h-1)-\frac{1}{6}\gamma(h-2) \ \ (8)$$

The auxiliary equation is,

$$\lambda^2-\frac{5}{6}\lambda+\frac{1}{6}=0$$

$$(\lambda-\frac{1}{2})(\lambda-\frac{1}{3})=0$$

roots are $\lambda_1=\frac{1}{2}$, and $\lambda_2=\frac{1}{2}$

Thus the general solution is of the form:

$$\gamma(h) = C_1\left(\frac{1}{2}\right)^{|h|} + C_2\left(\frac{1}{3}\right)^{|h|}$$

To find C_1 and C_2,

When h = 0,

$$\gamma(0) = C_1 + C_2 = \frac{477}{160}\sigma_z^2$$

When h = 1,

$$\gamma(1) = C_1\left(\frac{1}{2}\right) + C_2\left(\frac{1}{3}\right) = \frac{75}{32}\sigma_z^2$$

Solve for C_1, C_2, we get,

$$C_1 = \frac{81}{10}\sigma_z^2,\ C_2 = -\frac{819}{160}\sigma_z^2$$

Above all,

The autocovariance function for the process is,

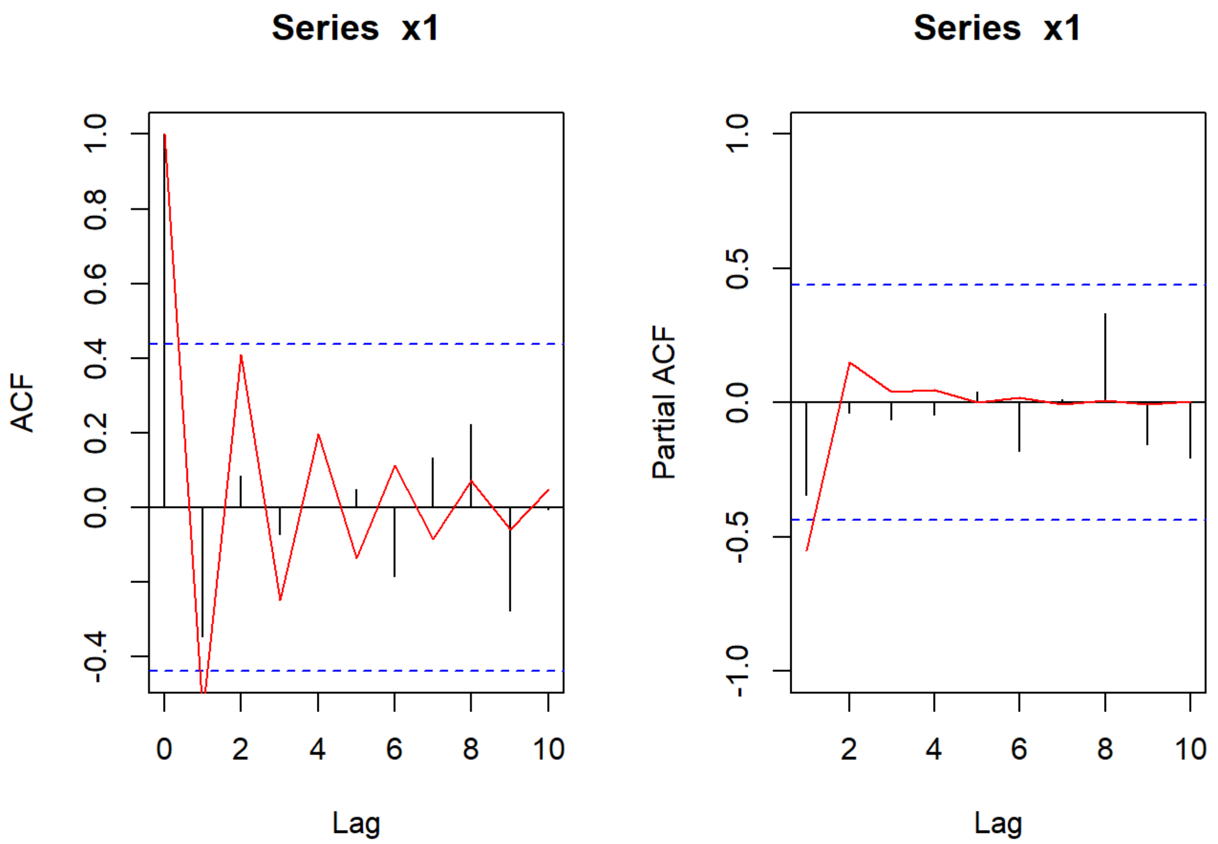
$$\gamma(h) = \left(\frac{81}{10}\left(\frac{1}{2}\right)^{|h|} - \frac{819}{160}\left(\frac{1}{3}\right)^{|h|}\right)\sigma_z^2$$

Q2

For all the plots in Q2, the red lines are theoretical ACF/PACF.

Case 1, n = 20

The plot of theoretical and sample ACF and PACF when simulating data from the given process of sample size n = 20.



To measure the theoretical and sample ACF and PACF numerically, we can calculate the mean square error
which is:
$$mean\ square\ error\ ACF = \frac{1}{n+1}\sum(theoretical\ ACF - sample\ ACF)^2$$
$$mean\ square\ error\ PACF = \frac{1}{n}\sum(theoretical\ PACF - sample\ PACF)^2$$
Where n is the maximum lags.

In this case, the values of the sample ACF when simulating data from the given process of sample size n = 20, with maximum lag 10 are,

```
##  [1]  1.000000000 -0.345252873  0.084250054 -0.071031681  0.003630778
##  [6]  0.047688455 -0.184906532  0.133186117  0.223934612 -0.275759715
## [11] -0.006067153
```

The theoretical ACF values are,

```
## [1]  1.00000000 -0.55251528  0.41064410 -0.24952139  0.19812017 -0.13511146
## [7]  0.11248082 -0.08459319  0.07213344 -0.05779978  0.04971525
```

Hence, the mean square error for ACF is,

```
## [1] 0.04201092
```

The values of the sample PACF when simulating data from the given process of sample size n = 20, with maximum lag 10 are,

```
## [1] -0.34525287 -0.03967924 -0.06196095 -0.04362399  0.04106670 -0.17887996
## [7]  0.01002258  0.33098788 -0.15392283 -0.20501794
```

The theoretical PACF values are,

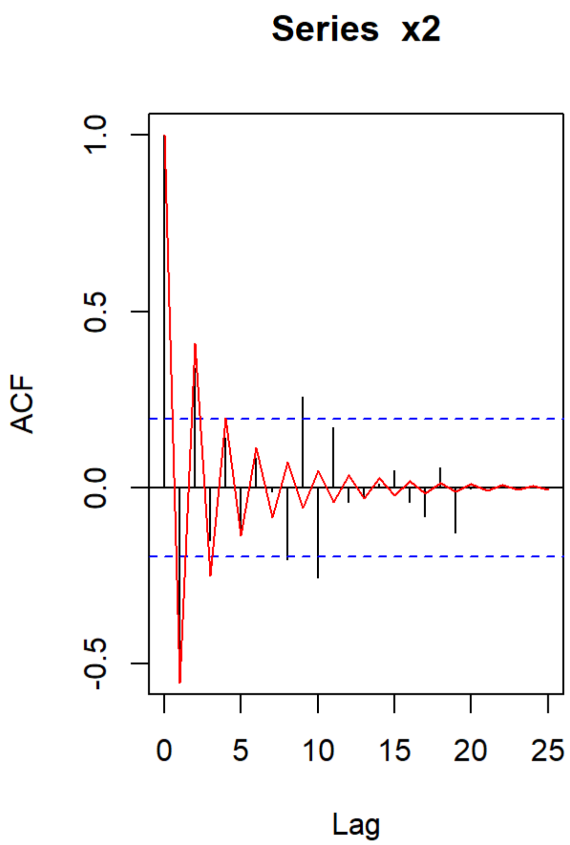
```
## [1] -0.5525152797  0.1516725069  0.0394177266  0.0466035893  0.0003752874
## [6]  0.0181841538 -0.0057360819  0.0090011706 -0.0051671261  0.0052096374
```

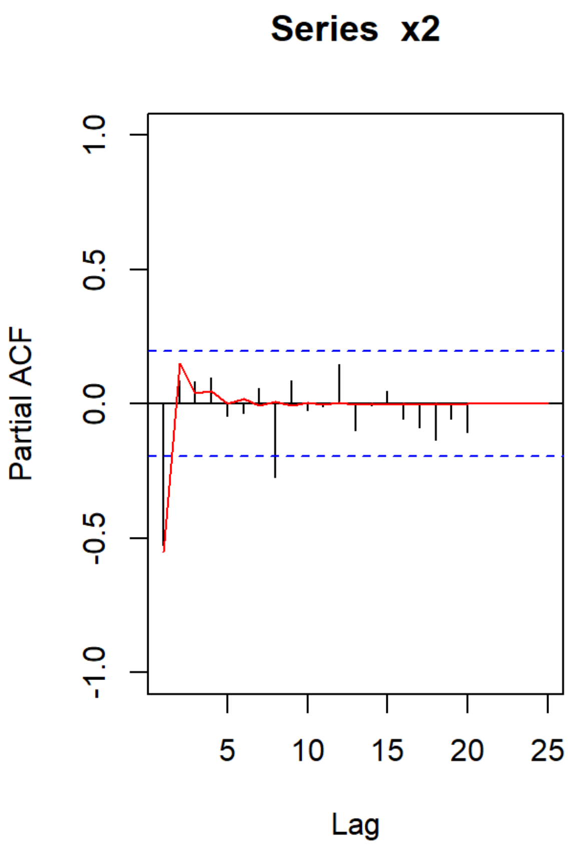
Hence, the mean square error for PACF is,

```
## [1] 0.03087296
```

Case 2, n = 100

The plot of theoretical and sample ACF and PACF when simulating data from the given process of sample size n = 100.





Same as above, to measure the theoretical and sample ACF and PACF numerically, we can calculate the mean square errors

The mean square error for ACF is,

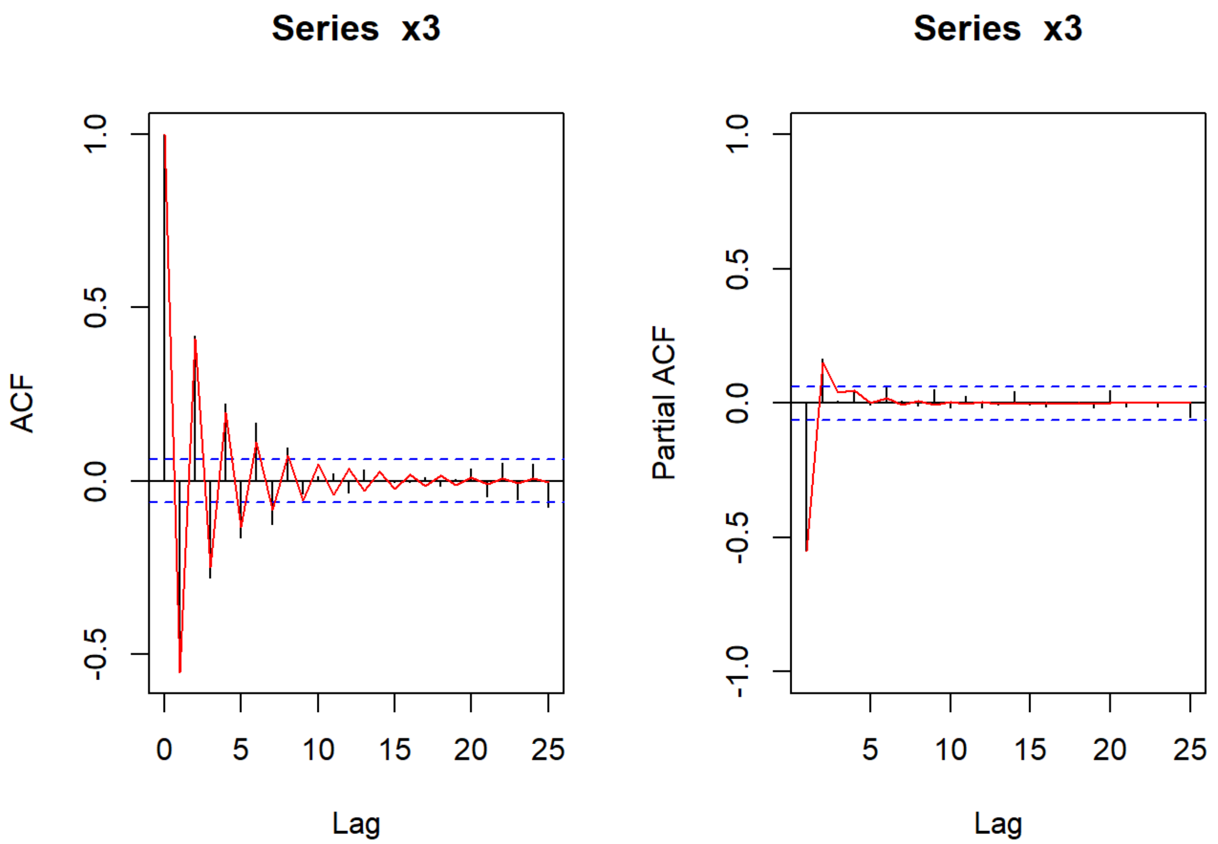
```
## [1] 0.01708503
```

The mean square error for PACF is,

```
## [1] 0.008975989
```

Case 3, n = 1000

The plot of theoretical and sample ACF and PACF when simulating data from the given process of sample size n = 1000.



Same as above, to measure the theoretical and sample ACF and PACF numerically, we can calculate the mean square errors

The mean square error for ACF is,

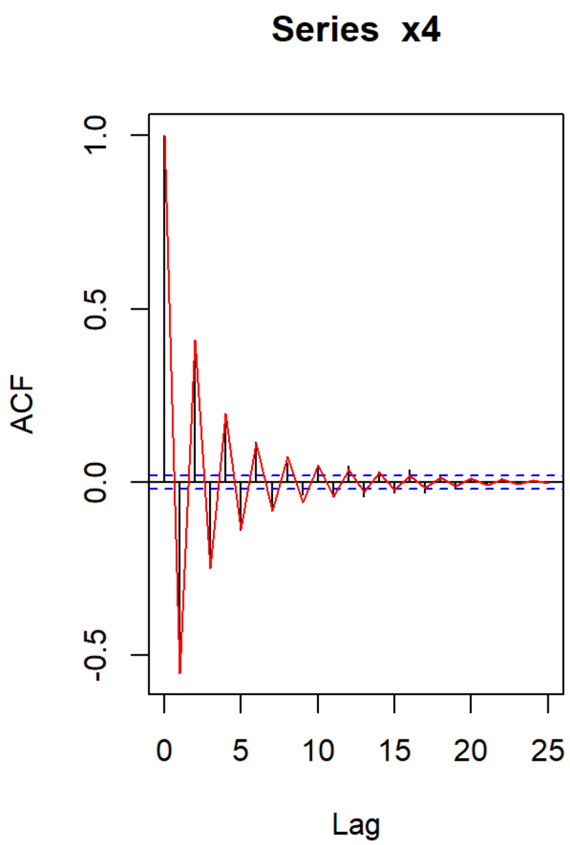
```
## [1] 0.00144035
```

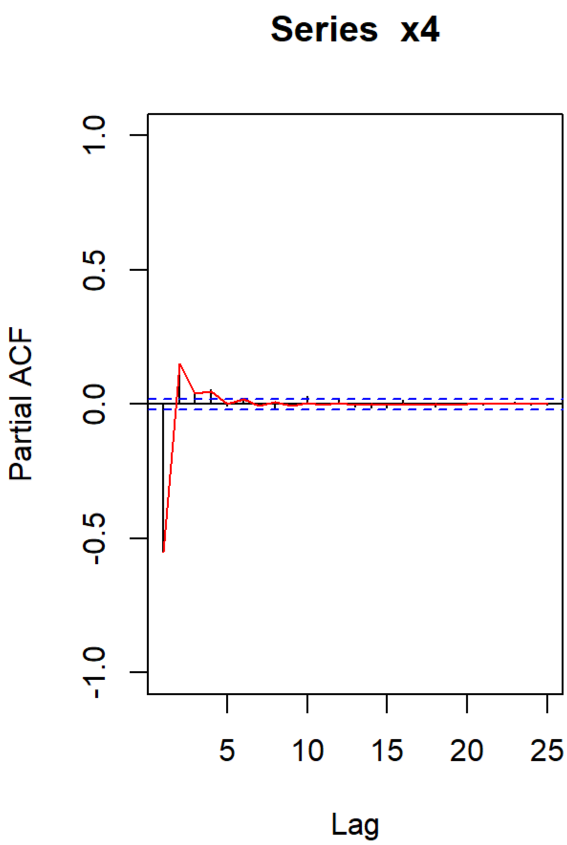
The mean square error for PACF is,

```
## [1] 0.0006801876
```

Case 4, n = 10000

The plot of theoretical and sample ACF and PACF when simulating data from the given process of sample size n = 10000.





Same as above, to measure the theoretical and sample ACF and PACF numerically, we can calculate the mean square errors

The mean square error for ACF is,

```
## [1] 7.279064e-05
```

The mean square error for PACF is,

```
## [1] 9.928432e-05
```

Comments:

From the plots above,

When n = 20, the sample size is small, thus the sample ACF/PACF are more variable and may not provide a reliable estimate of the theoretical ACF/PACF.

When n = 100, the sample ACF/PACF provide a better estimate of the theoretical ACF/PACF. However, there may still be some variability in the sample ACF/PACF.

When n = 1000, With a larger sample size of 1000, the sample ACF/PACF converge to the theoretical ACF/PACF more closely. The sample ACF/PACF provide a more reliable estimate of the properties of the process.

When n = 10000, the sample ACF/PACF converge even more closely to the theoretical ACF/PACF. The sample ACF/PACF should provide a highly reliable estimate of the properties of the given process.

In general, it is obvious that the peaks of the red lines are getting dense, and the theoretical and the sample ACF/PACF are getting closer to each other as the simulated sample size n increases.

Numerically, the result of mean square errors for the theoretical and the sample ACF/PACF are getting smaller as the simulated samples size increased.

Mean square errors for different sample size

sample_size	MSE_ACF	MSE_PACF
20	0.0420109	0.0308730
100	0.0170850	0.0089760
1000	0.0014403	0.0006802
10000	0.0000728	0.0000993

Therefore, we could conclude that the larger the sample size is, the quicker the sample ACF and PACF converge to their theoretical values.

```
# Appendix with R code

#Set the seed
set.seed(7334)
#Simulate 20 observations from the ARMA(3,2) process with
#phi1 = -0.8, phi2 = 0.4, phi3 = 0.3, theta1 = 0.75, theta2 = -0.6 and sigma_z = 1
x1 <- arima.sim(n=20,model=list(ar=c(-0.8,0.4,0.3),ma=c(0.75,-0.6)), sd = 1)

#Simulate 100 observations from the process above
x2 <- arima.sim(n=100,model=list(ar=c(-0.8,0.4,0.3),ma=c(0.75,-0.6)), sd = 1)

#Simulate 1000 observations from the process above
x3 <- arima.sim(n=1000,model=list(ar=c(-0.8,0.4,0.3),ma=c(0.75,-0.6)), sd = 1)

#Simulate 10000 observations from the process above
x4 <- arima.sim(n=10000,model=list(ar=c(-0.8,0.4,0.3),ma=c(0.75,-0.6)), sd = 1)

#Calculate the theoretical ACF and PACF for the process above with lags up to 10
t_acf1 <- ARMAacf(ar=c(-0.8,0.4,0.3),ma=c(0.75,-0.6),lag.max=10)
t_pacf1 <- ARMAacf(ar=c(-0.8,0.4,0.3),ma=c(0.75,-0.6),lag.max=10, pacf=TRUE)

#Calculate the theoretical ACF and PACF for the process above with lags up to 25
t_acf2 <- ARMAacf(ar=c(-0.8,0.4,0.3),ma=c(0.75,-0.6),lag.max=25)
t_pacf2 <- ARMAacf(ar=c(-0.8,0.4,0.3),ma=c(0.75,-0.6),lag.max=25, pacf=TRUE)

#Set the two plots with 1 row and 2 columns
par(mfrow=c(1,2))

#Plot the sample ACF of x1
acf(x1,xlim=c(0,10))
#Add the theoretical ACF to this plot as a red line
lines(c(0:10),t_acf1,col="red")

#Plot the sample PACF of x1
#ylim=c(-1,1) ensures that the vertical axis runs from -1 to 1.
pacf(x1,xlim=c(1,10),ylim=c(-1,1))
#Add the theoretical PACF to this plot as a red line
lines(c(1:10),t_pacf1,col="red")
#Calculate the sample acf of x1 with maximum lag 10
y_acf1 <- acf(x1,lag.max = 10)

# store the sample acf as a vector
y_sample_acf1 <- as.vector(y_acf1$acf[,,1])

#Calculate the sample pacf of x1 with maximum lag 10
y_pacf1 <- pacf(x1,lag.max = 10)

# store the sample pacf as a vector
y_sample_pacf1 <- as.vector(y_pacf1$acf[,,1])

unname(t_acf1)
#theoretical ACF values

y_sample_acf1
(sum((y_sample_acf1 - t_acf1)^2))/11
# calculate the sum of the square of the errors

t_pacf1
```

```
#theoretical PACF values

y_sample_pacf1
(sum((y_sample_pacf1 - t_pacf1)^2))/10
# calculate the sum of the square of the errors

#Set the two plots with 1 row and 2 columns
par(mfrow=c(1,2))

#Plot the sample ACF of x2
acf(x2,xlim=c(0,25))
#Add the theoretical ACF to this plot as a red line
lines(c(0:25),t_acf2,col="red")

#Plot the sample PACF of x2
#ylim=c(-1,1) ensures that the vertical axis runs from -1 to 1.
pacf(x2,xlim=c(1,25),ylim=c(-1,1))
#Add the theoretical PACF to this plot as a red line
lines(c(1:25),t_pacf2,col="red")

#Calculate the sample acf of x2 with maximum lag 25
y_acf2 <- acf(x2,lag.max = 25)

# store the sample acf as a vector
y_sample_acf2 <- as.vector(y_acf2$acf[,,1])

#Calculate the sample pacf of x2 with maximum lag 25
y_pacf2 <- pacf(x2,lag.max = 25)

# store the sample pacf as a vector
y_sample_pacf2 <- as.vector(y_pacf2$acf[,,1])

(sum((y_sample_acf2 - t_acf2)^2))/26
# calculate the sum of the square of the errors


(sum((y_sample_pacf2 - t_pacf2)^2))/25
# calculate the sum of the square of the errors

#Set the two plots with 1 row and 2 columns
par(mfrow=c(1,2))

#Plot the sample ACF of x3
acf(x3,xlim=c(0,25))
#Add the theoretical ACF to this plot as a red line
lines(c(0:25),t_acf2,col="red")

#Plot the sample PACF of x3
#ylim=c(-1,1) ensures that the vertical axis runs from -1 to 1.
pacf(x3,xlim=c(1,25),ylim=c(-1,1))
#Add the theoretical PACF to this plot as a red line
lines(c(1:25),t_pacf2,col="red")

#Calculate the sample acf of x3 with maximum lag 25
y_acf3 <- acf(x3,lag.max = 25)

# store the sample acf as a vector
y_sample_acf3 <- as.vector(y_acf3$acf[,,1])
```

```
#Calculate the sample pacf of x3 with maximum lag 25
y_pacf3 <- pacf(x3,lag.max = 25)

# store the sample pacf as a vector
y_sample_pacf3 <- as.vector(y_pacf3$acf[,,1])

(sum((y_sample_acf2 - t_acf2)^2))/26
# calculate the sum of the square of the errors
(sum((y_sample_pacf2 - t_pacf2)^2))/25

#Set the two plots with 1 row and 2 columns
par(mfrow=c(1,2))

#Plot the sample ACF of x4
acf(x4,xlim=c(0,25))
#Add the theoretical ACF to this plot as a red line
lines(c(0:25),t_acf2,col="red")

#Plot the sample PACF of x4
#ylim=c(-1,1) ensures that the vertical axis runs from -1 to 1.
pacf(x4,xlim=c(1,25),ylim=c(-1,1))
#Add the theoretical PACF to this plot as a red line
lines(c(1:25),t_pacf2,col="red")


#Calculate the sample acf of x4 with maximum lag 25
y_acf4 <- acf(x4,lag.max = 25)

# store the sample acf as a vector
y_sample_acf4 <- as.vector(y_acf4$acf[,,1])

#Calculate the sample pacf of x4 with maximum lag 25
y_pacf4 <- pacf(x4,lag.max = 25)

# store the sample pacf as a vector
y_sample_pacf4 <- as.vector(y_pacf4$acf[,,1])

(sum((y_sample_acf4 - t_acf2)^2))/26
# calculate the sum of the square of the errors

(sum((y_sample_pacf4 - t_pacf2)^2))/25
# calculate the sum of the square of the errors

df <- data.frame(
  sample_size = c("20", "100", "1000" , "10000"),
  MSE_ACF = c(0.04201092, 0.01708503, 0.00144035, 7.279064e-05),
  MSE_PACF = c(0.03087296, 0.008975989, 0.0006801876, 9.928432e-05)
)
library(knitr)
kable(df, caption = "Mean square errors for different sample size")
```