Nannan Wang

Project #2

STAT 878

Spring 2022

Complete the following problems below. Within each part, include your R program output with code inside of it and any additional information needed to explain your answer. Your R code and output should be formatted in the exact same manner as in the course notes.

1. (15 total points) The purpose of this problem is to continue the radon data problem from project #1.
   1. (3 points) Plot the estimated ACF and the PACF for lags 1 to 20. Put both plots in the same graphics window (1 row, 2 columns) with the y-axis limits between -1 and 1.

> dev.new(width=8,height=6,pointsize=10)

NULL

> par(mfcol=c(1,2))

> acf(x=radon,lag.max = 20,ylim=c(1,-1),main="ACF of Radon time series")

> pacf(x=radon,lag.max = 20,ylim=c(1,-1),main="PACF of Radon time series")

Chart, box and whisker chart

Description automatically generated

* 1. (3 points) Are there any true partial autocorrelations jhh that are significantly different from zero (a = 0.05)? Explain.

Yes, from the PACF plot, is significantly different from zero because the first line in the PACF plot representing is across the blue dotted line. The dotted blue line represents the critical value of the t-test for .If the length of each is larger than the distance of the dotted line from x-axis, then the corresponding is significantly different from 0.

* 1. (3 points) Based on the estimated ACF and PACF plots, suggest “possible” model(s) for the data. Justify your suggestions using the terminology given in the table on p. 2 of CompareTrueEstimatedACF\_PACF-6.docx.

According to the ACF plot, since ACF estimates have the trend of tailing off to 0, there would be an AR part in the model. Because only is significantly different from 0, I would suggest q=1. I would also have q=2 as a candidate value for q because the autocorrelation at lag 2 is relatively larger than those at other lags and it is not specifically clear where at which lag the autocorrelation begins to tail off to 0.

According to the PACF plot, since PACF estimates have the trend of tailing off to 0, there would be an MA part in the model. Because only is significantly different from 0, I would suggest p=1. To be safe, it would be wise to include p=2 in our consideration to ensure that the tailing off begins at lag p=1

In summary, I would suggest a ARMA(1,1), ARMA(1,2), ARMA(2,1), and ARMA(2,2) models as candidate models.

* 1. (3 points) Plot the true ACF and PACF for an AR(1) model with j1 = 0.3013. Put both plots in the same graphics window (1 row, 2 columns) with the y-axis limits between -1 and 1. I recommend using ARMAacf() to find the ACF and PACF. Compare these plots with those estimated from the data in part a).

> # plot true ACF and PACF plots with phi1=0.3013

> dev.new(width=8,height=6,pointsize=10)

NULL

> par(mfcol=c(1,2))

> plot(y=ARMAacf(ar=0.3013,lag.max = 20),x=0:20,type = "h",ylim = c(-1,1),xlab = "h",ylab = expression(rho(h)),main=expression(paste("TRUE ACF of AR(1) with",phi1[1]==0.3013)))

> abline(h=0)

> plot(x=ARMAacf(ar=0.3013,lag.max = 20,pacf=TRUE),type = "h",ylim = c(-1,1),xlab = "h",ylab=expression(phi1[hh]),main=expression(paste("TRUE PACF of AR(1) with",phi1[1]==0.3013)))

> abline(h=0)

Chart, box and whisker chart

Description automatically generated

In the True ACF plot, tails off to 0 and are 0 after lag 5, while in the ACF plot in part a, although there’s a trend that tail off to 0, those do not tail off to 0 as fast as those in the true ACF plot. There is difference in the PACF plot as well. In the true PACF plot, cuts off to 0 after lag 1, while in the PACF in part a, tails off to 0 instead of cutting off to 0.

T

hese true ACF and PACF plots with j1 = 0.3013 is very different from the ACF and PACF plots estimated from the radon time series data. In the true ACF and PACF plots, there is not a and having negative values, while from the ACF and PACF estimated from radon time series in part a, there are negative values. In addition, the true PACF plot with j1 = 0.3013 cuts off to 0 after h=1, while the PACF estimated from the radon time series tails off to 0.

* 1. (3 points) Plot the true ACF and PACF for an MA(1) model with q1 = 0.2592. Put both plots in the same graphics window (1 row, 2 columns) with the y-axis limits between -1 and 1. I recommend using ARMAacf() to find the ACF and PACF. Compare these plots with those estimated from the data in part a).

> dev.new(width=8,height=6,pointsize=10)

NULL

> par(mfcol=c(1,2))

> plot(y=ARMAacf(ma=0.2592,lag.max = 20),x=0:20,type = "h",ylim = c(-1,1),xlab = "h",ylab = expression(rho(h)),main=expression(paste("TRUE ACF of MA(1) with",theta[1]==0.2592)))

> abline(h=0)

> plot(x=ARMAacf(ma=0.2592,lag.max = 20,pacf=TRUE),type = "h",ylim = c(-1,1),xlab = "h",ylab=expression(phi1[hh]),main=expression(paste("TRUE PACF of MA(1) with",theta[1]==0.2592)))

> abline(h=0)

Chart, box and whisker chart

Description automatically generated

The true ACF and PACF plots from the MA(1) model = 0.2592 are different from the estimated ACF and PACF plots from the radon time series data. The most obvious difference is in the ACF plots. In the estimated ACF plot, tails off to 0, while in the true ACF of the MA(1) model cuts off to 0 after h=1. There is also difference in the two PACF plots. In the estimated PACF plot, it takes more lags for the estimated tails off to 0, while in the true PACF of the MA(1) model, it only takes 3 lags for the to tail off to 0.

1. (15 total points) For an ARMA(1,2) and  = 1, complete the following:
   1. (6 points) Find g(0), g(1), and r(1) with the help of the  result from the course notes.

Text, letter

Description automatically generated

Finding the:

Text, letter

Description automatically generated

In general, (i=2,3,4,…) can be expressed as:

Text

Description automatically generated

Using

Text, letter

Description automatically generated

Text

Description automatically generated

A picture containing diagram

Description automatically generated

* 1. (3 points) Using j1 = 0.3, q1 = 0.1, and q2 = 0.2, find the values of g(0), g(1), and r(1) with the expressions you found in part a).

Using the expressions found in part a):

Text, letter

Description automatically generated

Recall,

Diagram

Description automatically generated with medium confidence

So,

A picture containing text

Description automatically generated

A picture containing text

Description automatically generated

Thus,

Text

Description automatically generated

* 1. (3 points) Using j1 = 0.3, q1 = 0.1, and q2 = 0.2, find the values of g(0) and g(1) with the help of ARMAtoMA() and  (use an upper bound of 20 in the sum). Of course, these should match what you got in part b)!

ARMAtoMA function returns values as follows:

> # 2.c

> ARMAtoMA <- ARMAtoMA(ar=0.3,ma=c(0.1,0.2),lag.max = 100)

> ARMAtoMA

[1] 4.000000e-01 3.200000e-01 9.600000e-02 2.880000e-02 8.640000e-03 2.592000e-03 7.776000e-04 2.332800e-04

[9] 6.998400e-05 2.099520e-05 6.298560e-06 1.889568e-06 5.668704e-07 1.700611e-07 5.101834e-08 1.530550e-08

These values match what we calculated in part b.

can be calculated by summing the inter product of two same vectors including all . I defined the vector as “psiVector” in the following code chunk:

> psiVector <- c(1,ARMAtoMA)

> head(psiVector)

[1] 1.00000 0.40000 0.32000 0.09600 0.02880 0.00864

> gamma.0 <- sum(psiVector\*psiVector)

> gamma.0

[1] 1.272527

To calculate , I created two vectors and find the inner product of the two vectors. The sum of the vector is the estimated value.

> psiVector99 <- psiVector[1:99]

> psiVector100 <- psiVector[2:100]

> gamma.1 <- sum(psiVector99\*psiVector100)

> gamma.1

[1] 0.5617582

It turns out that the and calculated in this part are the same in part b.

* 1. (3 points) Using j1 = 0.3, q1 = 0.1, and q2 = 0.2, find the value of r(1) with the help of ARMAacf(). Of course, this should match what you got in part b)!

Using the result from part c):

rho1 <- gamma.1/gamma.0

> rho1

[1] 0.4414508

Using the ARMAacf function:

> head(ARMAacf(ar=0.3,ma=c(0.1,0.2),lag.max = 20))

0 1 2 3 4 5

1.000000000 0.441450777 0.289602763 0.086880829 0.026064249 0.007819275

It turns out that calculated in this part is the same as the true value calculated from the ARMAacf function.

1. (19 points) The problem is a continuation of the previous problem for an ARMA(1, 2) with j1 = 0.3, q1 = 0.1, and q2 = 0.2.
   1. (3 points) Plot the ACF and PACF with the help of the plot() and ARMAacf() functions. Put both plots in the same graphics window (1 row, 2 columns) with the y-axis limits between -1 and 1.

> par(mfcol=c(1,2))

> plot(y=ARMAacf(ar=0.3,ma=c(0.1,0.2),lag.max = 20),x=0:20,type = "h",ylim = c(-1,1),xlab = "h",ylab = expression(rho(h)))

> abline(h=0)

> plot(x=ARMAacf(ar=0.3,ma=c(0.1,0.2),lag.max = 20,pacf=TRUE),type = "h",ylim = c(-1,1),xlab = "h",ylab=expression(phi1[hh]))

> abline(h=0)

Chart, box and whisker chart

Description automatically generated

* 1. For the sample sizes of n = 20, 100, and 10,000, complete the following:
     1. (3 points) Simulate n observations from this model using the arima.sim() function with  = 1. Set a seed number (1287 for n = 20, 9198 for n = 100, 8712 for n = 10,000) right before you use arima.sim()*.* Print the first 6 and last 6 observations of the data you simulated for each sample size

> # n=20

> set.seed(1287)

> n.20 <- arima.sim(model=list(order=c(1,0,2),ar=0.3,ma=c(0.1,0.2)),n=20,rand.gen = rnorm,sd=1)

> head(n.20)

[1] -1.3947662 -0.4348349 1.1530043 0.7052666 1.4753558 1.1231884

> tail(n.20)

[1] -0.1676370 -1.4973930 -0.3330622 1.1252211 0.4242377 0.3574755

> #n=100

> set.seed(9198)

> n.100 <- arima.sim(model=list(order=c(1,0,2),ar=0.3,ma=c(0.1,0.2)),n=100,rand.gen = rnorm,sd=1)

> head(n.100)

[1] -1.9165311 -0.9439305 -0.5526135 -0.5539728 -1.0233593 -0.7631167

> tail(n.100)

[1] -0.4911244 -1.0913517 -0.9524217 -0.9941773 0.1612366 0.4145669

#n=10000

> set.seed(8712)

> n.10000 <- arima.sim(model=list(order=c(1,0,2),ar=0.3,ma=c(0.1,0.2)),n=10000,rand.gen = rnorm,sd=1)

> head(n.10000)

[1] 1.27903237 -0.09534493 0.56465636 0.08733939 1.26040575 0.49122601

> tail(n.10000)

[1] -0.58597270 -1.78330390 -0.04082442 -1.56330815 0.41673879 0.18684316

* + 1. (5 points) Construct plots of the estimated ACF and PACF for each sample. Put both plots in the same graphics window (1 row, 2 columns) with the y-axis limits between -1 and 1. Compare the plots here with those from part a).

> # n=20

> dev.new(width=8,height=6,pointsize=10)

NULL

> par(mfcol=c(1,2))

> acf(x=n.20,type = "correlation",lag.max = 20,ylim=c(-1,1),main="ACF for sample size 20")

> pacf(x=n.20,lag.max = 20,ylim=c(-1,1),main="PACF for sample size 20")

Chart, box and whisker chart

Description automatically generated

> # n=100

> dev.new(width=8,height=6,pointsize=10)

NULL

> par(mfcol=c(1,2))

> acf(x=n.100,type = "correlation",lag.max = 20,ylim=c(-1,1),main="ACF for sample size 100")

> pacf(x=n.100,lag.max = 20,ylim=c(-1,1),main="PACF for sample size 100")

Chart

Description automatically generated

> # n=10000

> dev.new(width=8,height=6,pointsize=10)

NULL

> par(mfcol=c(1,2))

> acf(x=n.10000,type = "correlation",lag.max = 20,ylim=c(-1,1),main="ACF for sample size 10000")

> pacf(x=n.10000,lag.max = 20,ylim=c(-1,1),main="PACF for sample size 10000")

Chart, box and whisker chart

Description automatically generated

From the above three plots of estimated ACF and PACF from sample sizes of 20, 100, and 10000, we could tell that the ACF and PACF plot from sample size 10000 looks almost identical to the true ACF and PACF, while plots from sample size 20 and 100 are not as close as the true ACF and PACF plot. The result matches with my expectation because with sample size goes up, the variance of the estimator decreases and so the estimates are more accurate.

* + 1. (5 points) Simulate 1,000 different data sets for each sample size beginning with the same seed number as in i). For each data set, compute . Find the mean of these 1,000  values and compare it to . Using these averaged  values, does it appear that  is a consistent estimator of ? Explain.

The true value of  is 0. 4414508 (code could be found below) using the ARMAacf function.

In summary, mean of  of sample size 20 is 0.2983119, mean of  of sample size 100 is 0.4080574, and mean of  of sample size 10000 is 0.4408854. The true value is 0.4414508 (code and result could be found below), with sample size increases, we could observe that  gets really closer to the true value. So, I think  is a consistant estimator of (0.4414508).

> ACF.true <- ARMAacf(ar=0.3,ma=c(0.1,0.2),lag.max = 20)

> ACF.true[2]

1

0.4414508

This above chuck of code shows the true value of  from the ARMAacf function.

> # n=20

> n20.summary <- matrix(data=NA,nrow = 1000,ncol=3)

> set.seed(1287)

> for (i in 1:1000){

+ x <- arima.sim(model=list(order=c(1,0,2),ar=0.3,ma=c(0.1,0.2)),n=20,rand.gen = rnorm,sd=1)

+ acfs <- acf(x=x,lag.max = 20,type = "correlation")

+ acf1 <- acfs$acf[2]

+ z <- acf1\*sqrt(20)

+ if (z>qnorm(p=0.975) | z< -qnorm(p=0.975)){

+ reject=1

+ }

+ else{

+ reject=0

+ }

+ n20.summary[i,] <- c(acf1,reject,z)

+ }

> mean(n20.summary[,1])

[1] 0.2983119

> # n=100

> n100.summary <- matrix(data=NA,nrow = 1000,ncol=3)

> set.seed(9198)

> for (i in 1:1000){

+ x <- arima.sim(model=list(order=c(1,0,2),ar=0.3,ma=c(0.1,0.2)),n=100,rand.gen = rnorm,sd=1)

+ acfs <- acf(x=x,lag.max = 20,type = "correlation")

+ acf1 <- acfs$acf[2]

+ z <- acf1\*sqrt(100)

+ if (z>qnorm(p=0.975) | z< -qnorm(p=0.975)){

+ reject=1

+ }

+ else{

+ reject=0

+ }

+ n100.summary[i,] <- c(acf1,reject,z)

+ }

> mean(n100.summary[,1])

[1] 0.4080574

> # n=10000 8712

> n10000.summary <- matrix(data=NA,nrow = 1000,ncol=3)

> set.seed(8712)

> for (i in 1:1000){

+ x <- arima.sim(model=list(order=c(1,0,2),ar=0.3,ma=c(0.1,0.2)),n=10000,rand.gen = rnorm,sd=1)

+ acfs <- acf(x=x,lag.max = 20,type = "correlation")

+ acf1 <- acfs$acf[2]

+ z <- acf1\*sqrt(10000)

+ if (z>qnorm(p=0.975) | z< -qnorm(p=0.975)){

+ reject=1

+ }

+ else{

+ reject=0

+ }

+ n10000.summary[i,] <- c(acf1,reject,z)

+ }

> mean(n10000.summary[,1])

[1] 0.4408854

* + 1. (3 points) Using the simulated data from iii), compute the estimated power for the test of Ho: r(1) = 0 vs. Ha: r(1) ¹ 0 with a type I error rate of 0.05.

Since the true  value is 0.4414508, under the null hypothesis that =0, the reject rate would be the power. In summary, the estimated power using sample size of 20 is 0.293, the estimated power using sample size of 100 is 0.976, and the estimated power using sample size of 10000 is 1. This result matches my expectation that with sample size increasing, power increases while keeping unchanged.

> # n=20

> n20.summary <- matrix(data=NA,nrow = 1000,ncol=3)

> set.seed(1287)

> for (i in 1:1000){

+ x <- arima.sim(model=list(order=c(1,0,2),ar=0.3,ma=c(0.1,0.2)),n=20,rand.gen = rnorm,sd=1)

+ acfs <- acf(x=x,lag.max = 20,type = "correlation")

+ acf1 <- acfs$acf[2]

+ z <- acf1\*sqrt(20)

+ if (z>qnorm(p=0.975) | z< -qnorm(p=0.975)){

+ reject=1

+ }

+ else{

+ reject=0

+ }

+ n20.summary[i,] <- c(acf1,reject,z)

+ }

> mean(n20.summary[,2])

[1] 0.293

> # n=100

> n100.summary <- matrix(data=NA,nrow = 1000,ncol=3)

> set.seed(9198)

> for (i in 1:1000){

+ x <- arima.sim(model=list(order=c(1,0,2),ar=0.3,ma=c(0.1,0.2)),n=100,rand.gen = rnorm,sd=1)

+ acfs <- acf(x=x,lag.max = 20,type = "correlation")

+ acf1 <- acfs$acf[2]

+ z <- acf1\*sqrt(100)

+ if (z>qnorm(p=0.975) | z< -qnorm(p=0.975)){

+ reject=1

+ }

+ else{

+ reject=0

+ }

+ n100.summary[i,] <- c(acf1,reject,z)

+ }

> mean(n100.summary[,2])

[1] 0.976

> # n=10000 8712

> n10000.summary <- matrix(data=NA,nrow = 1000,ncol=3)

> set.seed(8712)

> for (i in 1:1000){

+ x <- arima.sim(model=list(order=c(1,0,2),ar=0.3,ma=c(0.1,0.2)),n=10000,rand.gen = rnorm,sd=1)

+ acfs <- acf(x=x,lag.max = 20,type = "correlation")

+ acf1 <- acfs$acf[2]

+ z <- acf1\*sqrt(10000)

+ if (z>qnorm(p=0.975) | z< -qnorm(p=0.975)){

+ reject=1

+ }

+ else{

+ reject=0

+ }

+ n10000.summary[i,] <- c(acf1,reject,z)

+ }

> mean(n10000.summary[,2])

[1] 1