Time Series Simulation and Cob-Web Model

11/20/2020

1)

Simulate the Cob-Web model in R:

A) Let the long run equilibrium price p^*=1.5. You can call this variable in R PStar.

B) Set the value of the persistence factor φ=-0.4. You can call this variable in R Phi.

C) Generate prices (p_t) by creating random shocks (ε_t) from a normal distribution with zero mean and variance of 0.25. To create the list of prices, you can Initialize the series at the equilibrium price. Generate 10,000 prices. You can call the list of prices generated in R Prices.

R-CODE:

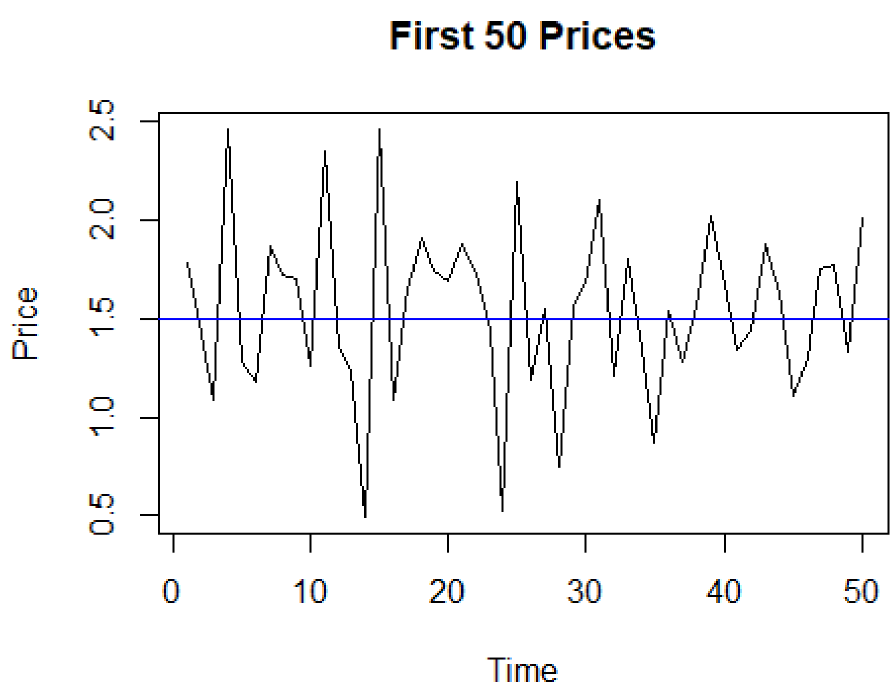
```
Pstar= 1.5 # A)
Phi= -.4 #B)
error= rnorm(10000,0,sqrt(0.25))
prices= c()
price.lag= 0
for(i in (seq(1:10000))){
  prices[i]<- Pstar * (1-Phi)+ (Phi*price.lag)+error[i]
  price.lag<- prices[i]
} # C
```

2)

Plot only the first 50 prices generated and include a horizontal line that illustrates the average price. Does this series look stationary? Discuss with your group how this process would change if φ=0 or φ=1. How effective can we be at predicting prices in these two special cases?

R-CODE:

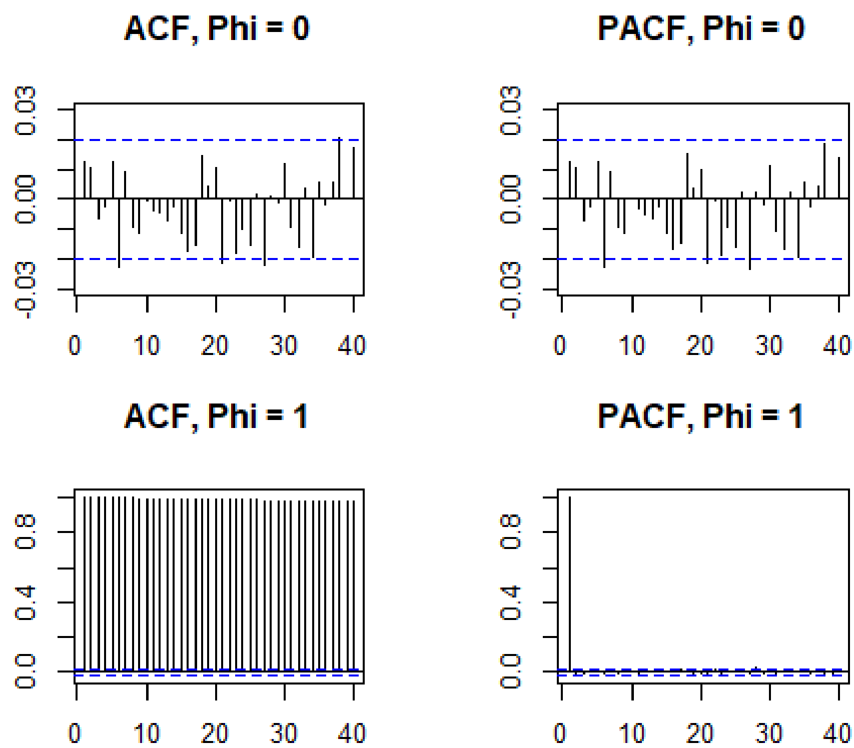
```
# plot first 50
avgprice= mean(prices)
ts.prices= ts(prices, frequency = 1)
plot( main = "First 50 Prices", ylab = "Price", window(ts.prices, end=
c(50)))
abline(h=(avgprice), col = "blue")
```




```
#Stationary Plot
print("Startionary Comparision for phi = -.4 Time series")

## [1] "Startionary Comparision for phi = -.4 Time series"

ts.prices %>% ur.kpss() %>% summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 12 lags.
##
## Value of test-statistic is: 0.0396
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
par(mfrow= c(2,2), mai= rep(0.5,4))

#When Phi = 0
price1 = c()
Phi1= 0
price.lag= 0
for(i in (seq(1:10000))){
  price1[i]<- Pstar * (1-Phi1)+ (Phi1*price.lag)+error[i]
  price.lag<- price1[i]
}

ts.price1= ts(price1,frequency = 1)
Acf(ts.price1, main = "ACF, Phi = 0")
Pacf(ts.price1, main = "PACF, Phi = 0")

#When Phi = 1
price2 = c()
Phi2= 1
price.lag= 0
for(i in (seq(1:10000))){
  price2[i]<- Pstar * (1-Phi2)+ (Phi2*price.lag)+error[i]
  price.lag<- price2[i]
}

ts.price2= ts(price2,frequency = 1)
Acf(ts.price2, main = "ACF, Phi = 1")
Pacf(ts.price2, main = "PACF, Phi = 1")
```




```
print("Startionary Comparision for phi = 0 Time series")

## [1] "Startionary Comparision for phi = 0 Time series"

ts.price1 %>% ur.kpss() %>% summary()

##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 12 lags.
##
## Value of test-statistic is: 0.0416
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739

print("Startionary Comparision for phi = 1 Time series")

## [1] "Startionary Comparision for phi = 1 Time series"

ts.price2 %>% ur.kpss() %>% summary()

##
## #######################
## # KPSS Unit Root Test #
```

```
## #######################
##
## Test is of type: mu with 12 lags.
##
## Value of test-statistic is: 47.4625
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
library(tseries)
adf.test(ts.prices) # p-value < 0.05 indicates the TS is stationary we have a pvalue of .01 thus stationary
```

```
## Warning in adf.test(ts.prices): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts.prices
## Dickey-Fuller = -22.473, Lag order = 21, p-value = 0.01
## alternative hypothesis: stationary
```

Answer:

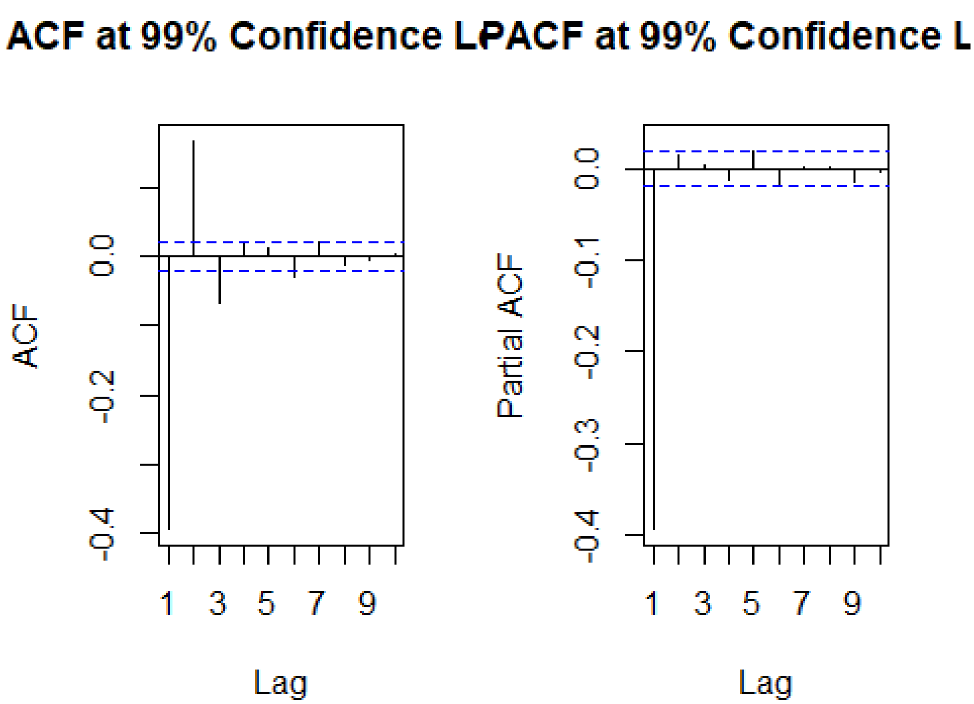
The time series that has been generated here is indeed stationary. When ϕ=0 the graph simply represents a white noise process and is even more stationary. When $\phi = 1$ the peaks of the graph stay at 1 and the graph is not stationary. Our effectiveness at predicting prices with $ϕ=0 $or 1 is pretty medicore at best. For instance, when phi is 0 it is resembling a white noise process indicating that observations have zero correlation, which is not useful when trying to predict future prices. Logic suggests that previous prices may have some impact on the decisions for setting future prices. While conversly a Phi of 1 results in the exact opposite effect suggesting that there is perfect correlation between our observations, thus all future predictions will be a positive value which may not be truly representative of the data and the reality of the world

3)

Load the library “forecast” and generate both the ACF and PACF plots at a 99% confidence level. Describe any patterns you observe in the ACF and PACF (How many correlations are statistically significant? Is there decaying behavior? Do the correlations alternate in sign?). Use the option lag.max=10 when creating your plots.

R-CODE:

```
par(mfrow=c(1,2))
Acf(prices, main ='ACF at 99% Confidence Level',  lag.max=10)
Pacf(prices, main ='PACF at 99% Confidence Level',  lag.max=10)
```



Answer:

The ACF returns 4 statistically significant lags at 1,2,3,6, this graph indicates decaying behavior. The PACF only returns 1 statistically significant lag at 1, this graph indicates decaying behavior as well. Some correlations have flipped signs at 3 & 4.

4)

Run a linear regression of the prices and the first lag. Start by loading the "dynlm" package in R. Then create a time series object with the Prices variable you created in 1). You can create a lag variable within your model by using the L() operator. What is the intercept? What is the slope? How do they relate to the parameters in the Cob-Web model?

R-CODE:

```
ts.prices.regression= dynlm(ts.prices~L(ts.prices,-1))
ts.prices.regression$coefficients

##      (Intercept) L(ts.prices, -1)
##        2.0866622       -0.3932441
```

Answer:

Our DYNLM Regression model's coeffiecients are as follows: $Intercept: 2.0866622 $ $Slope: -0.3932441$ While our Cob-web model Coefficients are: $Intercept: 1.5$
$Slope: -0.4$

5)

Load the avocado.csv data set and create a ts() object of "Avocado_Prices". Set the starting date to January, 2015 and the frequency of the ts() object to 52.

R-CODE:

```
avocado <- read.csv("avocado.csv")

Avocado_Prices= ts(data= avocado$Avocado_Prices, frequency = 52, start =
c(2015,1))
```

Answer:

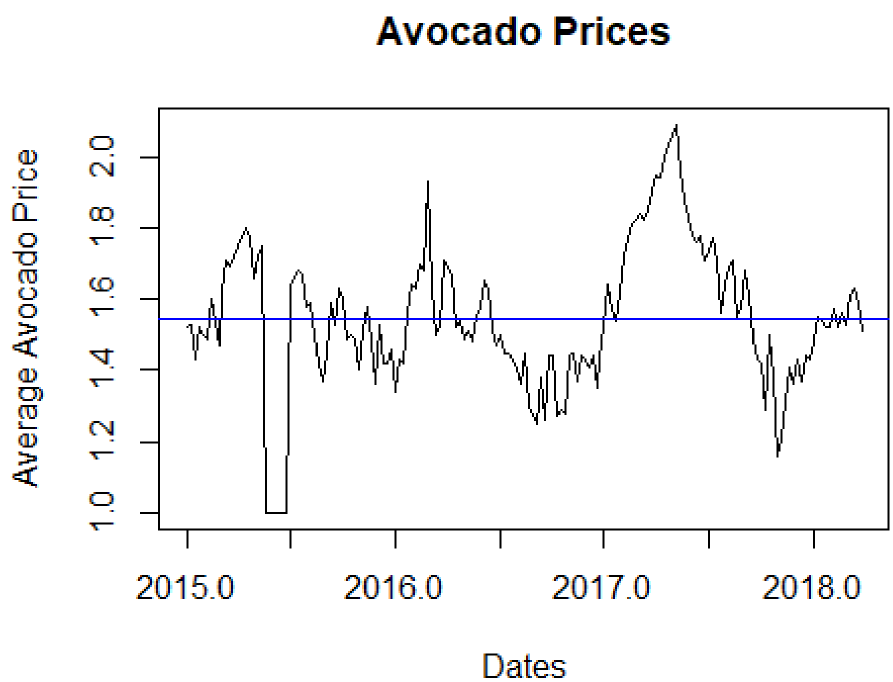
Is the R-CODE chunk related to (5)

6)

Plot the price of avocados and include the mean in your graph. What is the time mean of the series?

R-CODE:

```
avgprice= mean(Avocado_Prices)
plot(Avocado_Prices, main = "Avocado Prices", ylab= "Average Avocado Price",
xlab = "Dates")
abline(h = avgprice, col= "blue")
```



Answer:

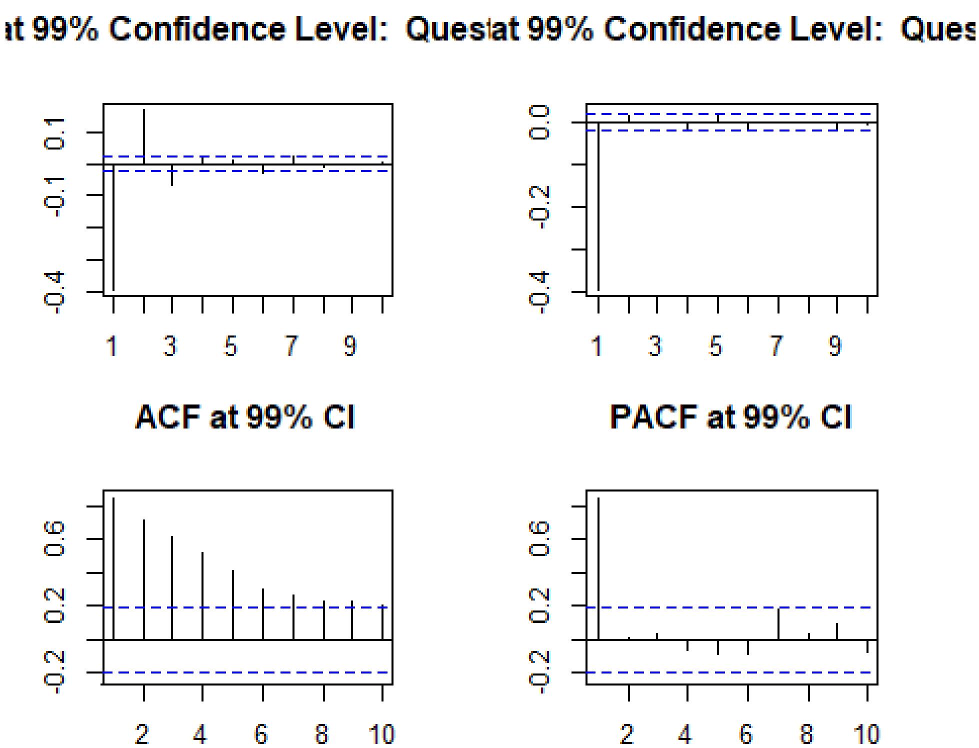
The time mean of the series is 1.546036 (rouded up from 1.5460355)

7)

Generate the ACF and PACF of the price of avocados (ci=0.99) by using the “forecast” package. Describe the differences/similarities between these plots and the ones generated by your simulation in 3).

R-CODE:

```r
par(mfrow= c(2,2), mai= rep(.5,4))

Acf(prices, main ='ACF at 99% Confidence Level:  Question 3',  lag.max=10)
Pacf(prices, main ='PACF at 99% Confidence Level:  Question 3',  lag.max=10)

Acf(Avocado_Prices, main ='ACF at 99% CI',  lag.max=10,ci=0.99)
Pacf(Avocado_Prices, main ='PACF at 99% CI',  lag.max=10,ci=0.99)
```



Answer:

Similarities between question 3’s graphs and question 7’s graphs: All have a large peak and statistically significant peak at 1, the difference is whether that peak is positive or negative in nature.

Differences between question 3’s graphs and question 7’s graphs: Many of the signs have been flipped in both the ACF & PACF at $ci = .99$ when comparing them to the graphs from question 3. The ACF at $ci = .99$ has only positive correlations that all appear to be statistically significant, yet the model does show clear decaying behavior.

8)

Run a regression of the price of avocados against its lag. This time let's use the "forecast" package and the function Arima(). Set the order to c(1,0,0) for an AR(1) process and the method to "CSS". What is the suggested value of Phi? What is the implied long run equilibrium price?

R-CODE:

```
Avocado.ar= Arima(Avocado_Prices, order = c(1,0,0), method = "CSS")
Avocado.ar

## Series: Avocado_Prices
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1    mean
##       0.8465  1.5461
## s.e.  0.0410  0.0541
##
## sigma^2 estimated as 0.01173:  part log likelihood=136.36

Avocadophi= 0.8465
Avocadoequ= 1.5461
```

Answer:

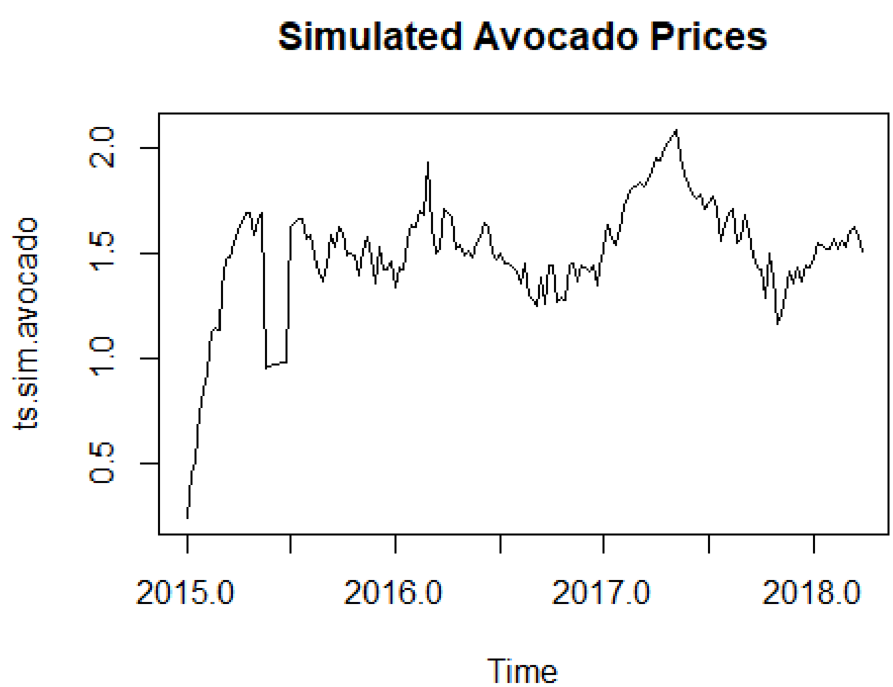
Suggested Value of Phi is 0.8465 & the new implied Long run Equilibrium Price is 1.5461

9)

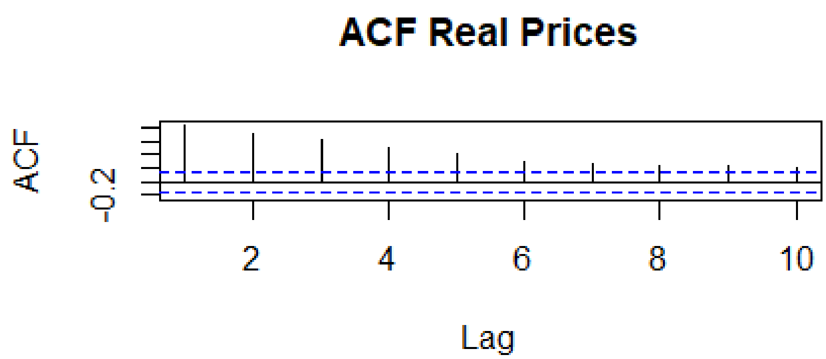
Run your simulation with the parameters that you have found in 8). In particular, create a ts() object of simulated prices with the same dates as the avocado data. Plot the new prices generated by your simulation. Explain why the plot looks less "zigzaggy" than the plot in 2). Continue by plotting the ACF and PACF of the new simulation. Are the plots like those found in 7)?

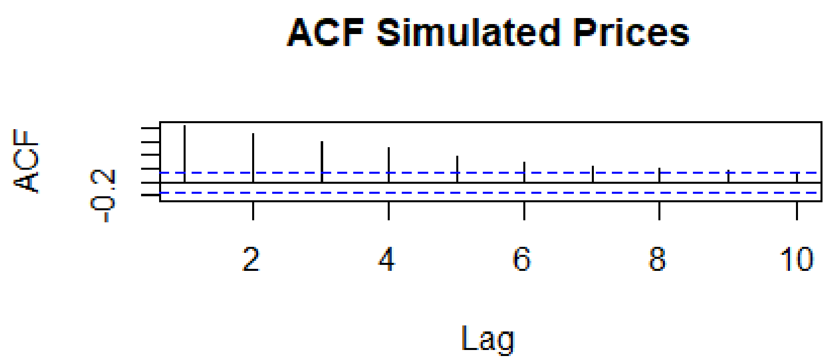
R-CODE:

```
simavocado= c()
avocadolag=0
avocadoerror= Avocado.ar$residuals
for(i in (seq(1:length(avocadoerror)))){
  simavocado[i]<- Avocadoequ * (1-Avocadophi)+
(Avocadophi*avocadolag)+avocadoerror[i]

  avocadolag<- simavocado[i]
}

ts.sim.avocado = ts(simavocado,frequency = 52, start = c(2015,1))
plot(ts.sim.avocado, main = "Simulated Avocado Prices")
```

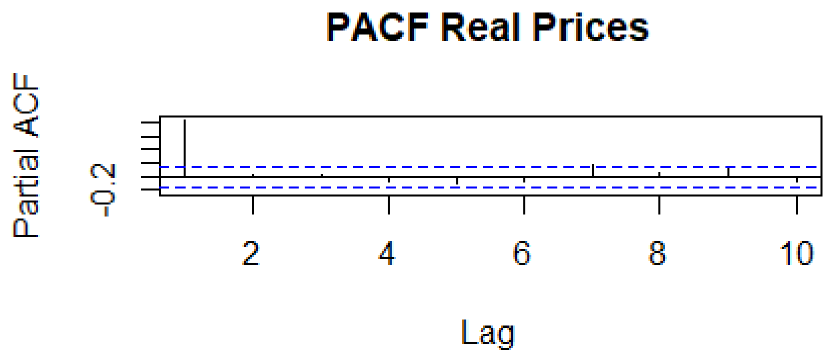


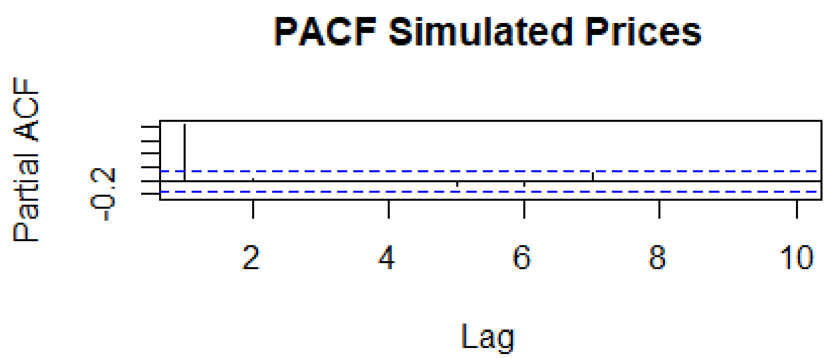
```
par(mfrow=c(2,1), mai = rep(.8,4))
Acf(Avocado_Prices, main = "ACF Real Prices", lag.max=10)
Acf(ts.sim.avocado, main = "ACF Simulated Prices", lag.max=10)
```






```
par(mfrow=c(2,1), mai = rep(.8,4))
Pacf(Avocado_Prices, main = "PACF Real Prices", lag.max=10)
Pacf(ts.sim.avocado, main = "PACF Simulated Prices", lag.max=10)
```





Answer:

The reason the new plot generated by our simulation looks "less Zigzaggy" than the graph from question 2 is because there is about twice (169) as many observation being plotted vs the 50 observations plotted by the graph in question 2. When comparing the real vs simulated ACF prices one should note that they are practically identical. When comparing the real vs simulated ACF prices one should note that they are practically identical. The only real differences being that the simulated data seems less statistically significant, my reasonging for this is that the peaks of the "Simulated Prices" graphs to not go as high or low as their real price counter part.

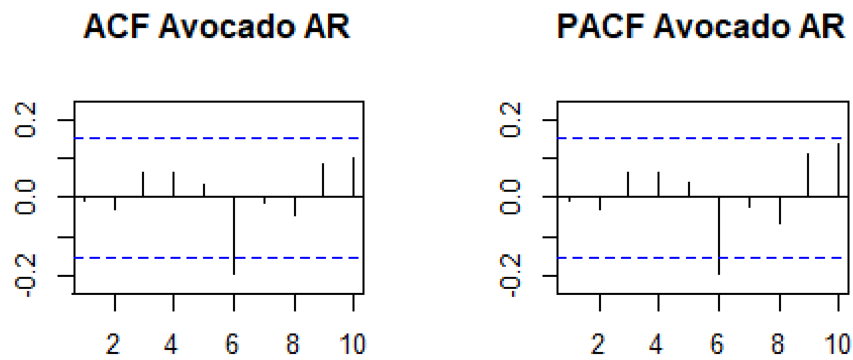
10)

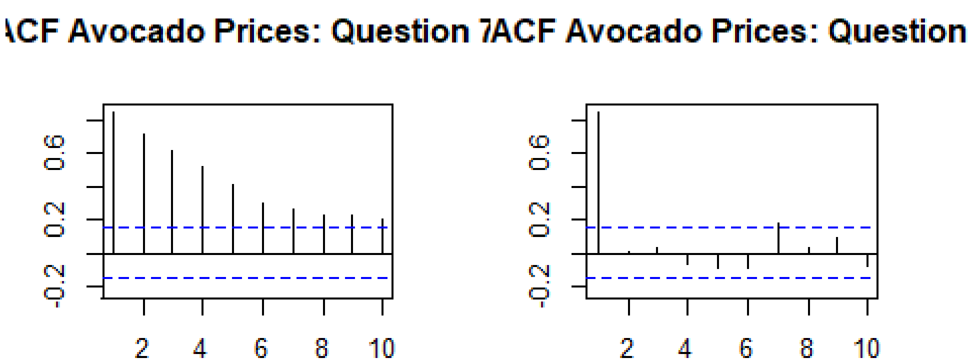
Retrieve the residuals from the regression you ran in 8) and generate the ACF and PACF plots for these residuals. Are the patterns found in 7) still there? Are there any significant lags? How would ACF and PACF plots compare to those of a white noise process?

R-CODE:

```
par(mfrow= c(2,2), mai = rep(.5,4))
Acf(Avocado.ar$residuals, main = "ACF Avocado AR",lag.max=10)
Pacf(Avocado.ar$residuals, main = "PACF Avocado AR",lag.max=10)
```

```
Acf(Avocado_Prices, main = "ACF Avocado Prices: Question 7", lag.max=10)
Pacf(Avocado_Prices, main = "PACF Avocado Prices: Question 7", lag.max=10)
```





Answer:

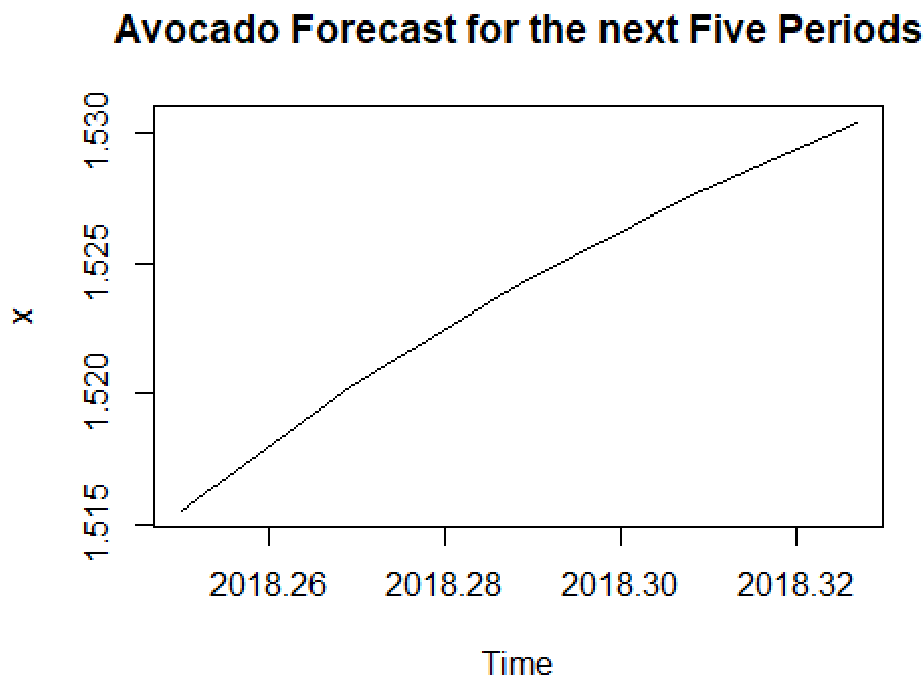
The patterns from 7) are not repeated in the ACF and PACF of the residuals from our regression. There seems to be one statisically significant lag at 6 in both the ACF & the PACF.
The ACF & the PACF of the residuals are not representative of a ACF & a PACF of a white noise process. This is because a white noise would indicate an autocorvarience of 0 something which our plots (the ACF & the PACF of the residuals) indicate is not true.

11)

What is your prediction for the next five periods? Hint: Use the forecast() function in r.

R-CODE:

```
avocado.forcast= forecast(Avocado.ar,h=5)
x=avocado.forcast$mean
plot(x, main = "Avocado Forecast for the next Five Periods")
```



Answer:

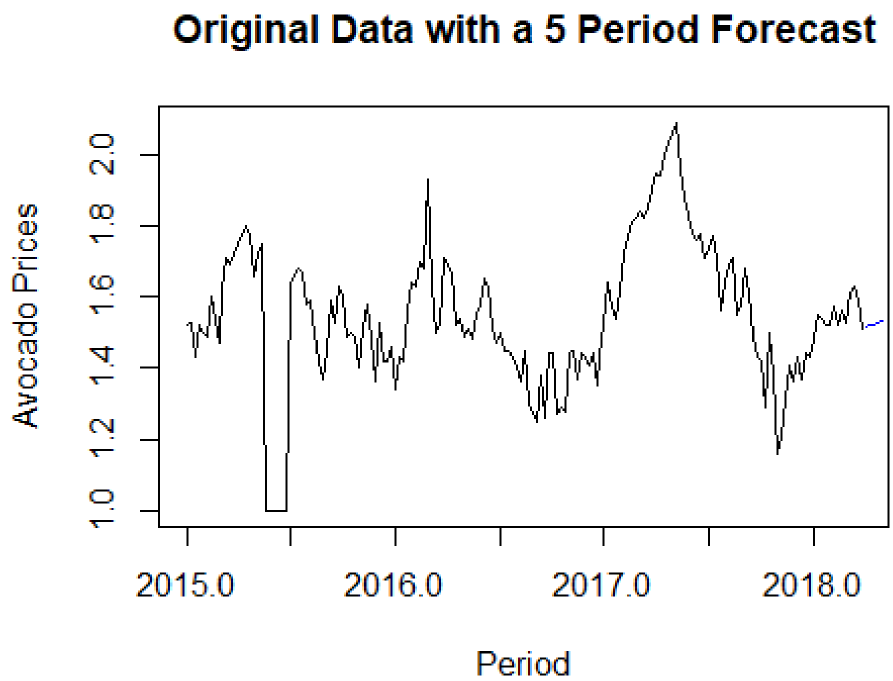
Is the Graph Labeled “Avocado Forecast for the next Five Periods”

12)

Create a plot with the original avocado data and the forecast.

R-CODE:

```
plot(window(Avocado_Prices), type = "l", ylab = "Avocado Prices", xlab=
"Period", main= "Original Data with a 5 Period Forecast")
lines(avocado.forcast$mean, col ="blue")
```



Answer:

Is the Graph labeled “Original Data with a 5 Period Forecast”