Assignment4

Team 2: Riti Dabas, Anoushka Mahar, Dylan Koury

12/2/2021

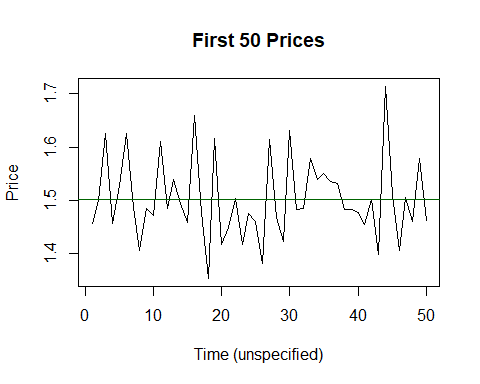
## Part 1

PStar = 1.5 #Long run equilibrium price  
Phi = -0.4 #Persistence factor  
n = 10000 #number of prices  
meanEr = 0 #Mean error  
stdDvEr = 0.08 #Std Deviation error  
set.seed(123)  
error = rnorm(n=n, mean = meanEr, sd=stdDvEr)  
PriceSimulator <- function(PStar, Phi, error) {  
 Prices = c() #List of prices  
 P0 = PStar #Setting initial price to equilibrium price  
 for (i in 1:n) {  
 pt = PStar\*(1-Phi) + Phi\*P0 + error[i]  
 Prices = c(Prices, pt)  
 P0 = pt  
 }  
 return(Prices)  
}  
Prices <- PriceSimulator(PStar,Phi,error)

Generated 10,000 Prices by creating random shocks from a normal distribution with given parameters.

## Part 2

plot(ts(Prices[1:50]), main="First 50 Prices", ylab="Price", xlab="Time (unspecified)")  
abline(h=mean(Prices[1:50]), col="dark green")

 Looking at the graph, the prices appear stationary as there is an even distributon around the mean across all 50 temporal points.

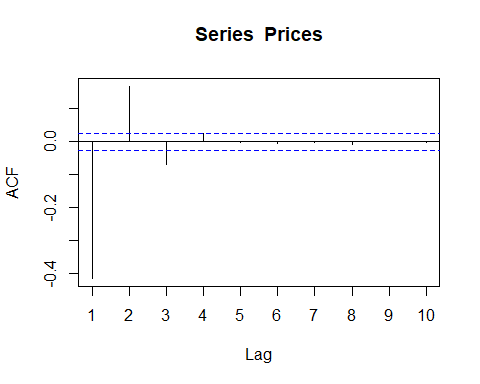
When phi = 0: the equation simplifies to PStar plus the error term, meaning that each year we can predict that the price will be generally around the long run equilibrium price. In fact, when we changed the value to zero, we saw the data centered around our long equilibrium price of 1.5 with many back and forth fluxations, it does not stray too far for too long. When phi = 1: The graph is still stationary but not at the long run equilibrium price. In fact when we changed the value of Phi we saw there was a mean of about 1.6 with a higher standard deviation and fewer fluxuations over the mean when compared to the Phi = 0 model, indicating this model is more likely to stray further from the mean. Therefore, Phi=0 is a better predictor and more effective on predicting the prices than when Phi=1, as we can use the long run equilibrium price as a fairly effective predictor. In fact, the long run equilibrium price is completely eliminated from the model when Phi = 1, and therefore cannot truly be used to accurately predict prices in this scenario.

## Part 3

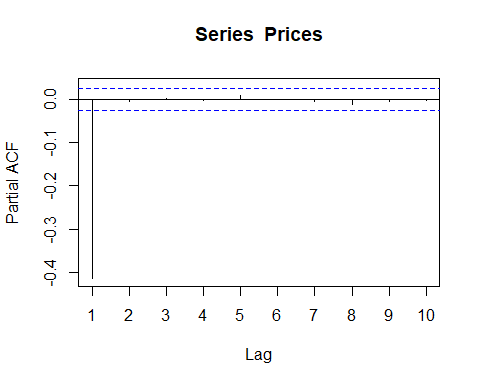
library(forecast)

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

Acf(Prices,lag.max = 10, plot=TRUE, type="correlation", ci=0.99)



Pacf(Prices, lag.max=10, plot=TRUE, ci=0.99)

 The ACF shows us that there are 3 statistically significant correlations, with one borderline correlation. As the lag progresses, we see the spikes oscillate between negative and positive. This is an MA 3 process and the decaying pattern of PACF shows it’s quick decay to zero after one negative spike.

## Part 4

library(dynlm)

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

options(scipen=999)  
PricesTS <- dynlm(Prices~L(Prices,-1))  
summary(PricesTS)

##   
## Time series regression with "numeric" data:  
## Start = 1, End = 10000  
##   
## Call:  
## dynlm(formula = Prices ~ L(Prices, -1))  
##   
## Residuals:  
## Min 1Q Median   
## -0.0000000000000000521 -0.0000000000000000022 -0.0000000000000000003   
## 3Q Max   
## 0.0000000000000000016 0.0000000000000032127   
##   
## Coefficients:  
## Estimate Std. Error  
## (Intercept) -0.000000000000000568434 0.000000000000000005584  
## L(Prices, -1) 1.000000000000000444089 0.000000000000000003716  
## t value Pr(>|t|)   
## (Intercept) -101.8 <0.0000000000000002 \*\*\*  
## L(Prices, -1) 269079021715113792.0 <0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.00000000000000003261 on 9998 degrees of freedom  
## Multiple R-squared: 1, Adjusted R-squared: 1   
## F-statistic: 7.24e+34 on 1 and 9998 DF, p-value: < 0.00000000000000022

INTERCEPT: 0

SLOPE: 1

## Part 5

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.3 v dplyr 1.0.7  
## v tidyr 1.1.3 v stringr 1.4.0  
## v readr 2.0.0 v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

avocado <- read\_csv("avocado.csv")

## Rows: 143 Columns: 2

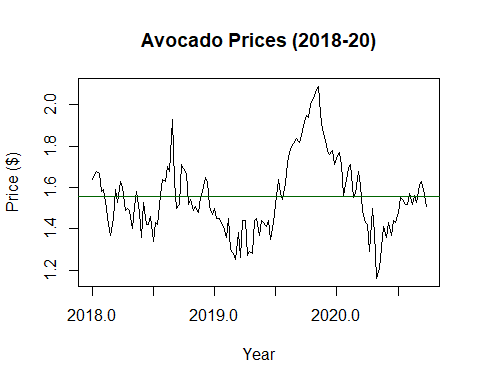
## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## dbl (2): Time, Avocado\_Prices

##   
## i Use `spec()` to retrieve the full column specification for this data.  
## i Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

Avocado\_Prices <- ts(avocado$Avocado\_Prices,frequency=52,start=c(2018,1))

## Part 6

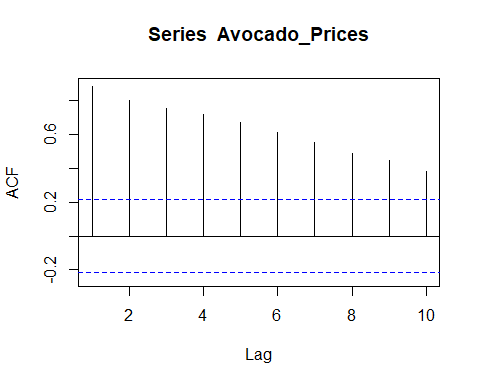
plot(Avocado\_Prices, xlab="Year", ylab="Price ($)", main="Avocado Prices (2018-20)")  
TimeMean = mean(Avocado\_Prices) #Time Mean  
abline(h=TimeMean,col="dark green")



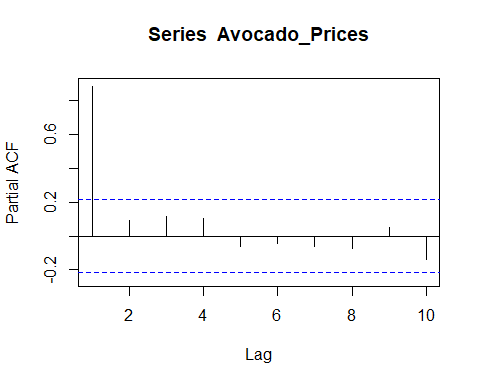
Time Mean: 1.5574126

## Part 7

Acf(Avocado\_Prices,lag.max = 10, plot=TRUE, type="correlation", ci=0.99)



Pacf(Avocado\_Prices,lag.max = 10, plot=TRUE, ci=0.99)



In part 3, both ACF and PACF plots showed an alternating decaying pattern; with three spikes in part 3 ACF plot. The decaying pattern was more visible in part 3 ACF whereas it is less prominent in part 7 ACF which consists of all ten spikes. PACF has one spike for both part 3 and 7; where part 3 PACF lacked a prominent decaying pattern. However the decaying pattern was more visible in part 7 even though it has at least 4 spikes.

## Part 8

ar\_1<-Arima(Avocado\_Prices,order=c(1,0,0))  
(ar\_1)

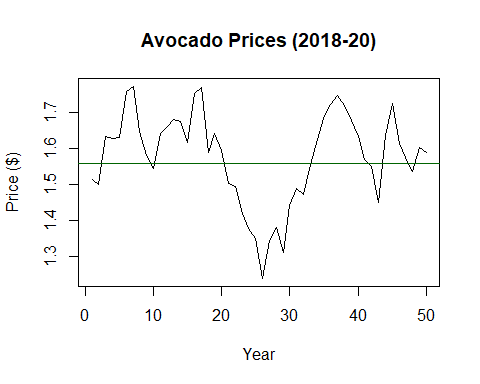
## Series: Avocado\_Prices   
## ARIMA(1,0,0) with non-zero mean   
##   
## Coefficients:  
## ar1 mean  
## 0.8822 1.5591  
## s.e. 0.0376 0.0564  
##   
## sigma^2 estimated as 0.007081: log likelihood=151.29  
## AIC=-296.59 AICc=-296.42 BIC=-287.7

AvocadoPhi <- ar\_1$coef[1] #Phi  
AvocadoPStar <- ar\_1$coef[2] #/(1-AvocadoPhi) #Long Run Equilibrium Price

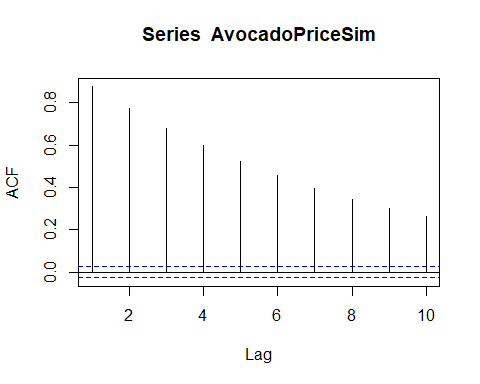
The suggested of Phi is 0.8821806 and the implied long run equilibrium price is 1.5590716

## Part 9

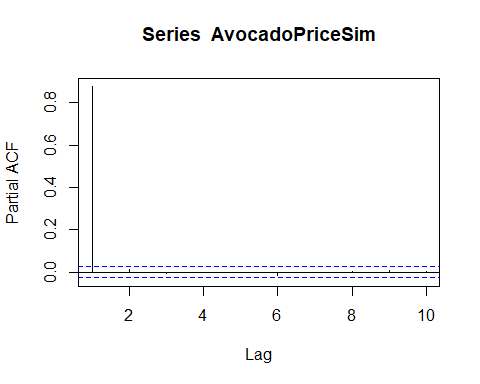
AvocadoPriceSim <- ts(PriceSimulator(PStar=AvocadoPStar, Phi=AvocadoPhi, error=error),  
 frequency=52, start=c(2018,1))  
plot(ts(AvocadoPriceSim[1:50]), xlab="Year", ylab="Price ($)", main="Avocado Prices (2018-20)")  
TimeMeanSim = mean(AvocadoPriceSim) #Time Mean  
abline(h=TimeMeanSim,col="dark green")



Acf(AvocadoPriceSim,lag.max = 10, plot=TRUE, type="correlation", ci=0.99)



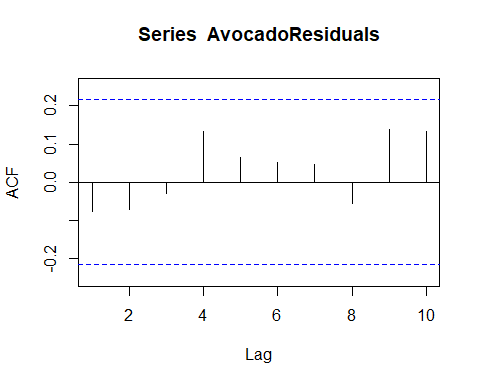
Pacf(AvocadoPriceSim,lag.max = 10, plot=TRUE, ci=0.99)



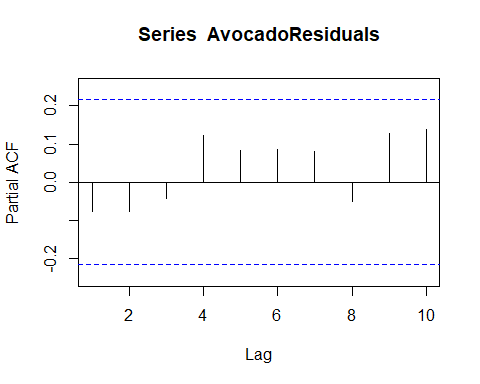
The ACF plot looks almost identical to that in part 7. The PACF plot for both have only one spike, negative for part 7 and positive for part 9. Another difference in the PACF plots is that the sticks are much closer to 0 for part 9.

## Part10

AvocadoResiduals <- ar\_1$residuals  
Acf(AvocadoResiduals,lag.max = 10, plot=TRUE, type="correlation", ci=0.99)



Pacf(AvocadoResiduals,lag.max = 10, plot=TRUE, ci=0.99)

 In ACF, the patterns shown in part 7 are not shown here. Part 7 had a lot of high spikes whereas there is a more distinct noisy pattern in part 10 ACF. Similarly, PACF for both parts are different and same patterns are not present in part 10 compared to part 7. There are no significant lags that are noticeable. A white noise process is a sequence of uncorrelated random variables with no visible pattern. The underlying difference is that we see some sort of patterns in ACF and PACF while white noise is more noisy with little to no observable pattern.

## Part 11

forecasts <- forecast(Avocado\_Prices)$mean[1:5]

In the next 5 periods we predict prices of: 1: 1.6293327

2: 1.6158403

3: 1.5729952

4: 1.5989453

5: 1.590317

## Part 12

plot(ts(c(Avocado\_Prices,forecasts),frequency=52,start=c(2018,1)),ylab="Price ($)",main="Avocado Prices and Forecast")  
lines(ts(forecasts,frequency=52,start=c(2020,40)), col="red")  
abline(h=mean(c(Avocado\_Prices,forecasts)),col="darkgreen")

