Time Series Midterm 2020 Take Home

Chad Madding

February 26, 2020

### Take-Home Portion:

Due 11:59pm CST Saturday February 29. Please Submit to 2DS in addition to Emailing it to Dr. Sadler.

### Question about the realization.

1. Do you think the data come from a stationary process? Defend your thoughts using the 3 conditions of stationarity. Provide acf plots for condition 3.

#load the midterm data  
midterm2020 = read.csv("midterm2020.csv",header = TRUE)  
#Convert to a Time Series  
midterm2020 = ts(midterm2020$x)

Description

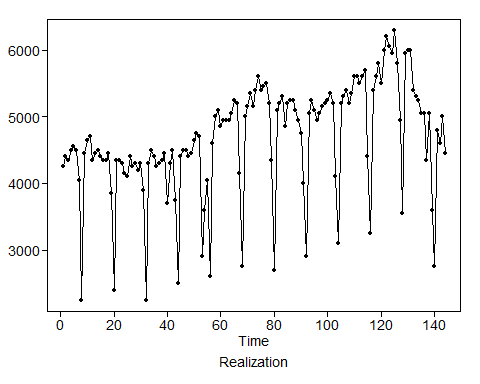
**Midterm2020 is a dataset in the form of a CSV provided for the take-home portion of the Spring 2020 Midterm in MSDS 6373 Time Series.**

Additional Realization

**We do not know the origin of the Midterm2020 dataset other than it was provided for one portion of the midterm. Therefore, we cannot obtain additional realizations.**

Condition 1 - Subpopulations of have the same mean for each . Restated, the mean does not depend on time ().

#Visualize the midterm data  
plotts.wge(midterm2020)



**The Midterm2020 dataset appears to oscillate with some seasonality. The series trends up slightly, then declines during the last unit, which does appear to represent a year. It seems the series cannot be stationary because there is a level of dependency between the mean and time.**

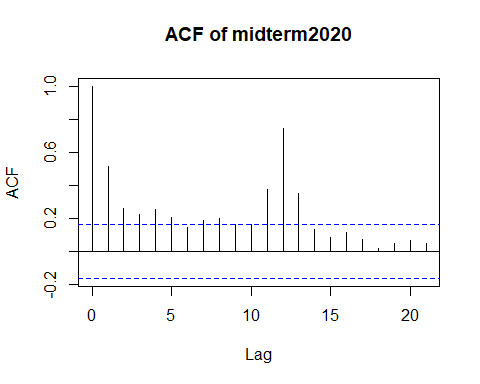
Condition 2 - Subpopulations of for a given time have a finite and constant variance for all . Restated, the variance does not depend on time.

**We can not accurately assess the homoscedasticity of the Midterm2020 dataset since the data is dependent on time (and therefore not stationary). Nonetheless, variance seems to be greater earlier in the measurements and smaller later in the measurements.**

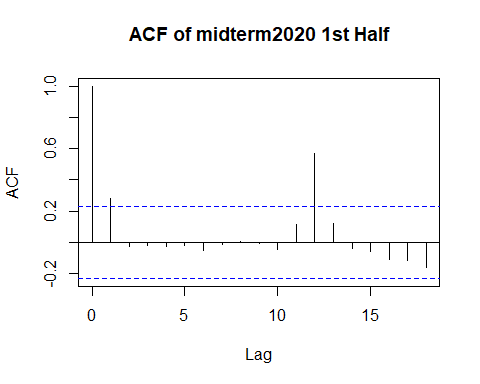
Condition 3 - The correlation between and depends only on . That is, the correlation between data points depends only on how far apart they are in time, not where they are in time.

**Based on the first ACF chart (ACF of midterm2020), there appears to be a strong seasonal component represented in the sinusoidal degradation. Autocorrelation cycles are almost identical across similar volumes of lags, evenly spaced.**

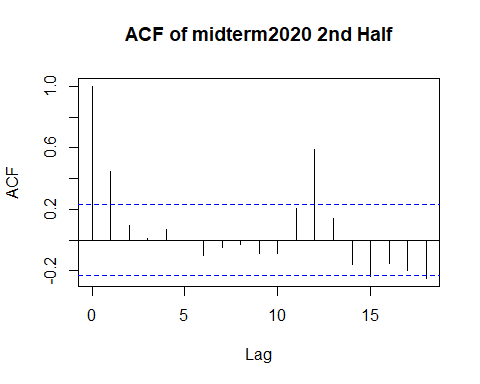
acf(midterm2020, main="ACF of midterm2020")



acf(midterm2020[1:72],plot=TRUE, main="ACF of midterm2020 1st Half")



acf(midterm2020[72:144],plot=TRUE, main="ACF of midterm2020 2nd Half")



**In analyzing the ACFs of the first and second halves of the series, the autocorrelations seem to mirror themselves (when comparing the first half to the second half). Therefore, the data seems dependent on position in time, not just on the distance between each pair of points.**

Conclusion

**Because the three conditions of a stationary time series cannot be confirmed, we must conclude that this is not a stationary time series and that there is a dependency on time driving the position of each successive data point.**

### The Models:

#### Consider these two models of the data in the realization in Midterm2020.csv:

* Model 1:
* Model 2:

#### Questions about Model 1:

1. Write this model in GLP form up to 4 terms.

psi.weights.wge(phi = c(0.5380, 0.0606, 0.1923), lag.max = 4)

## [1] 0.5380000 0.3500440 0.4132265 0.3469859

#### Questions about Model 2:

1. Is Model 2 Invertible? Provide evidence for or against.

**Model 2 is invertible, in that both absolute reciptricals are less than 1.**

factor.wge(phi = c(1.0507, -0.0756))

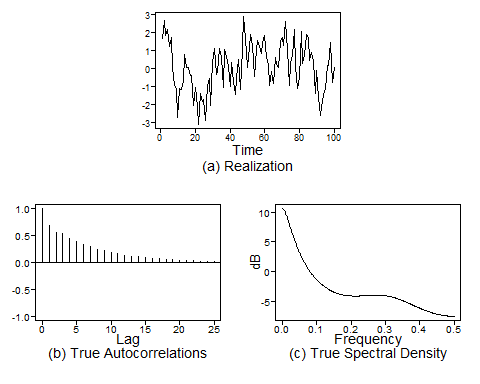
##   
## Coefficients of Original polynomial:   
## 1.0507 -0.0756   
##   
## Factor Roots Abs Recip System Freq   
## 1-0.9730B 1.0277 0.9730 0.0000  
## 1-0.0777B 12.8704 0.0777 0.0000  
##   
##

#### Questions for each model:

1. Provide acfs and spectral densities for each model.

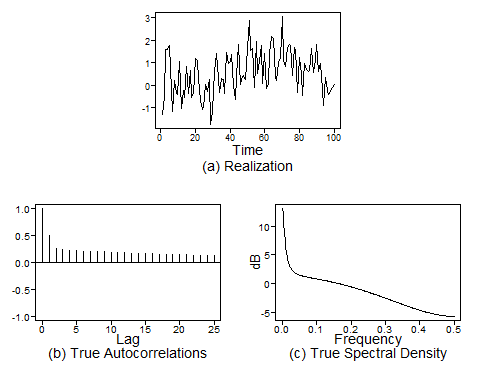
**Here are the ACF and spectral density for Model 1.**

#Model 1  
#plotts.sample.wge(x = midterm2020, phi = c(0.5380, 0.0606, 0.1923),s=12)  
plotts.true.wge(phi = c(0.5380, 0.0606, 0.1923))



**Here are the ACF and spectral density for Model 2.**

#Model 2  
plotts.true.wge(phi = c(1.0507,-0.0756), theta = c(0.5927, 0.2751))



1. Provide a factor table for each model.

**Here is the factor table for Model 1.**

#Model 1 factor table  
factor.wge(phi = c(0.5380, 0.0606, 0.1923))

##   
## Coefficients of Original polynomial:   
## 0.5380 0.0606 0.1923   
##   
## Factor Roots Abs Recip System Freq   
## 1-0.8650B 1.1560 0.8650 0.0000  
## 1+0.3270B+0.2223B^2 -0.7356+-1.9893i 0.4715 0.3064  
##   
##

**Here is the factor table for Model 2.**

#Model 2 factor table  
factor.wge(phi = c(1.0507,-0.0756))

##   
## Coefficients of Original polynomial:   
## 1.0507 -0.0756   
##   
## Factor Roots Abs Recip System Freq   
## 1-0.9730B 1.0277 0.9730 0.0000  
## 1-0.0777B 12.8704 0.0777 0.0000  
##   
##

1. Calculate the ASE for the last 12 months of the data set. (This will be only 1 ASE per model.).

#Get the length of the dataset  
lengthMT2020=length(midterm2020)  
#Model 1 Forcast  
model1Q6f = fcst(aruma, midterm2020, s = 12, phi = c(0.5380, 0.0606, 0.1923), n.ahead = 12, lastn = T, plot = F)  
#Model 1 ASE  
model1Q6\_ase = ase(midterm2020, model1Q6f)  
  
#Model 2 Forcast  
model2Q6f = fcst(arma, midterm2020, phi = c(1.0507,-0.0756), theta = c(0.5927,0.2751), n.ahead = 12, lastn = T, plot = F)  
#Model 2 ASE  
model2Q6\_ase = ase(midterm2020, model2Q6f)

**The ASE for Model 1 for the last 12 months of the data set: 1301829**

model1Q6\_ase

## [1] 1301829

**The ASE for Model 2 for the last 12 months of the data set: 836289**

model2Q6\_ase

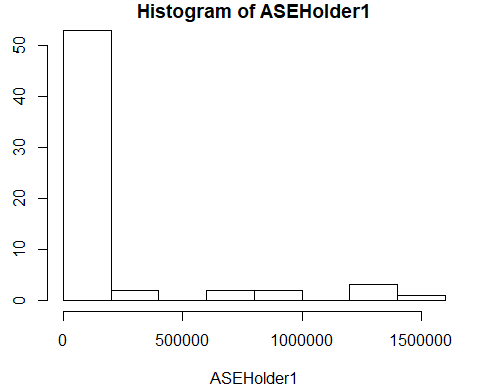
## [1] 836289.1

1. Calculate at least 10 ASEs across the data set and find their average (the rolling window ASE).

#Model 1  
phis1 = c(0.5380, 0.0606, 0.1923)  
thetas1 = 0  
s1 = 12  
d1 = 0  
  
trainingSize = 70  
horizon = 12  
ASEHolder1 = numeric()  
dataLength=length(midterm2020)  
i=0  
for( i in 1:(dataLength-(trainingSize + horizon) + 1))  
{  
   
 forecasts1 = fore.aruma.wge(midterm2020[i:(i+(trainingSize-1))],phi = phis1, theta = thetas1, s = s1, d = d1,n.ahead = horizon, plot = F)  
   
 ASE1 = mean((midterm2020[(trainingSize+i):(trainingSize+ i + (horizon) - 1)] - forecasts1$f)^2)  
   
 ASEHolder1[i] = ASE1  
  
}  
  
ASEHolder1

## [1] 62151.67 49493.80 51547.38 39834.38 57418.98 63021.43  
## [7] 81029.90 96712.73 90059.87 112142.56 91342.26 103495.29  
## [13] 98359.05 91144.05 56163.90 53471.08 39212.14 44274.56  
## [19] 36140.13 57318.58 87993.35 91163.85 44907.78 48249.58  
## [25] 45475.88 62613.31 52417.91 74309.88 90632.58 105686.83  
## [31] 78196.15 42693.73 48163.76 60902.04 58609.89 65562.07  
## [37] 79966.91 59950.80 59143.63 73644.86 75917.25 81205.85  
## [43] 98479.48 133801.64 101860.29 125121.80 141870.24 163328.35  
## [49] 150228.36 117176.41 114385.08 105394.04 186167.90 283486.74  
## [55] 363189.07 737157.60 733698.95 958978.04 984575.34 1218370.87  
## [61] 1388842.84 1423357.87 1301829.30

hist(ASEHolder1)



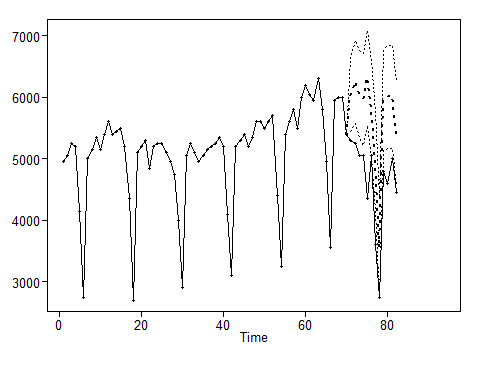
summary(ASEHolder1)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 36140 58014 87993 217350 121149 1423358

WindowedASE1 = mean(ASEHolder1)  
WindowedASE1

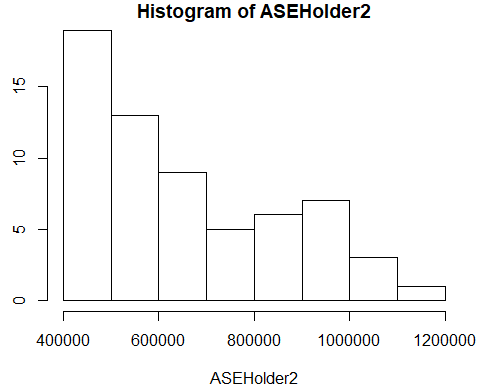
## [1] 217349.9

# Visualization  
  
i = length(ASEHolder1)  
fs1 = fore.aruma.wge(midterm2020[i:(i+(trainingSize+horizon)-1)],phi = phis1, theta = thetas1, s = s1, d = d1,n.ahead = 12, lastn = T, plot = T)



ASE1 = mean((midterm2020[(i+trainingSize):(i+(trainingSize+horizon)-1)] - fs1$f )^2)

#Model 2  
phis2 = c(1.0507,-0.0756)  
thetas2 = c(0.5927, 0.2751)  
s2 = 0  
d2 = 0  
  
ASEHolder2 = numeric()  
i=0  
for( i in 1:(dataLength-(trainingSize + horizon) + 1))  
{  
   
 forecasts2 = fore.arma.wge(midterm2020[i:(i+(trainingSize-1))],phi = phis2, theta = thetas2, n.ahead = horizon, plot = F)  
   
 ASE2 = mean((midterm2020[(trainingSize+i):(trainingSize+ i + (horizon) - 1)] - forecasts2$f)^2)  
   
 ASEHolder2[i] = ASE2  
  
}  
  
#ASEHolder2 = ASEHolder  
hist(ASEHolder2)



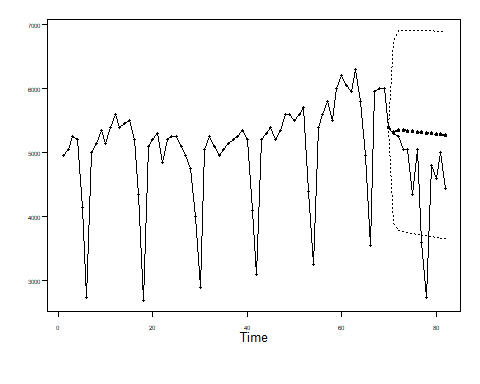
WindowedASE2 = mean(ASEHolder2)  
  
summary(ASEHolder2)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 412399 485372 588553 654834 814259 1126142

WindowedASE2

## [1] 654834.2

# Visualization  
  
i = length(ASEHolder2)  
fs2 = fore.arma.wge(midterm2020[i:(i+(trainingSize+horizon)-1)],phi = phis2, theta = thetas2, n.ahead = horizon, lastn = TRUE, plot = T)



ASE2 = mean((midterm2020[(i+trainingSize):(i+(trainingSize+horizon)-1)] - fs2$f )^2)

1. Compare the single ASE to the rolling window ASE. Are they roughly the same, is one significantly larger? Does it provide evidence as to which model is more useful?

**Now, let’s compare the single ASE from question 6 to the “windowed” results in question 7. For Model one, we have the original ASE at 1301829 and the windowed at 217350. The original ASE from the second model is 836289 and the windowed model 2 ASE is 654834. Both models ASE’s improved significantly, but when the original ASE’s were calculated, model 2 had the better ASE. After applying a rolling window to both models, the first model outperformed the second one. Applying an average to ASE’s taken in windowed sections provide evidence that model one is more useful.**

#Original model 1 ase  
model1Q6\_ase

## [1] 1301829

#Windowed model 1 ase  
WindowedASE1

## [1] 217349.9

#Original model 2 ase  
model2Q6\_ase

## [1] 836289.1

#Windowed model 2 ase  
WindowedASE2

## [1] 654834.2

#### Final Question:

1. Given your analysis, which model do you feel is more useful in making 12-month forecasts?

**Given all the information above, we feel model one will outperform the second one over time. The ASE obtained on the first model using the rolling window method proved to be better than the second model.**

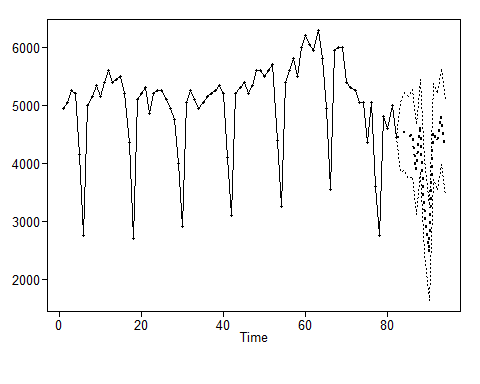
#### BONUS (up to 3 points):

Create an interesting, descriptive and useful plot to visualize the forecasts that the rolling window ASE was based on. This would help the analyst diagnose why the ASE is large or small and/or where it is fitting relatively well and relatively poorly. In addition, it may add confidence to the client that the model is performing adequately.

**To help visualize how much better model one could perform over model two, we have provided a few plots of both models. Each chart shows a forecast for the coming year.**

**With the results collected from the rolling window ASE model, one will outperform the second model. The Forecast Model One plot below shows a more natural trend seen over the past several years.**

## Forecast Model One



**Model two is trending to the mean. This type of data has too much seasonality in it for model two, and the ASE seemed to point that out.**

## Forecast Model Two

