Live Session Assignment 10

Nikhil Gupta

2020-03-09 05:27:27

Table of Contents

# Setup

library(tswge)

## Warning: package 'tswge' was built under R version 3.5.3

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 3.5.3

## -- Attaching packages -------------------------------------------------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.0 v purrr 0.3.2  
## v tibble 2.1.3 v dplyr 0.8.3  
## v tidyr 0.8.3 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

## Warning: package 'ggplot2' was built under R version 3.5.3

## Warning: package 'tibble' was built under R version 3.5.3

## Warning: package 'tidyr' was built under R version 3.5.3

## Warning: package 'readr' was built under R version 3.5.2

## Warning: package 'purrr' was built under R version 3.5.3

## Warning: package 'dplyr' was built under R version 3.5.3

## Warning: package 'stringr' was built under R version 3.5.3

## Warning: package 'forcats' was built under R version 3.5.3

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

library(tswgewrapped)  
library(orcutt)

## Warning: package 'orcutt' was built under R version 3.5.3

## Loading required package: lmtest

## Warning: package 'lmtest' was built under R version 3.5.3

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 3.5.3

##   
## Attaching package: 'zoo'

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

# For pre-live

## Setup

electrictiy = read.csv("../Datasets/electricity\_usage\_nikhil\_house.csv")  
head(electrictiy)

## Date Usage  
## 1 12/31/2009 280  
## 2 1/31/2010 669  
## 3 2/28/2010 559  
## 4 3/31/2010 611  
## 5 4/30/2010 309  
## 6 5/31/2010 238

sum(is.na(electrictiy$Usage))

## [1] 6

* There are some missing values. We will need to impute these

data = ts(electrictiy$Usage, frequency = 12, start = c(2009, 12))  
  
library(imputeTS)

## Warning: package 'imputeTS' was built under R version 3.5.3

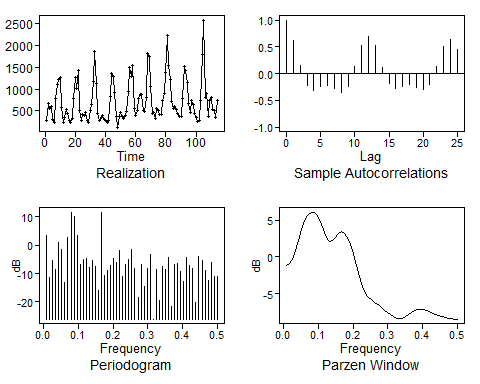
##   
## Attaching package: 'imputeTS'

## The following object is masked from 'package:zoo':  
##   
## na.locf

x = imputeTS::na\_interpolation(data, option = 'spline')  
x

## Jan Feb Mar Apr May Jun Jul  
## 2009   
## 2010 669.0000 559.0000 611.0000 309.0000 238.0000 775.0000 1113.0000  
## 2011 523.0000 440.0000 277.0000 234.0000 332.0000 779.0000 1274.0000  
## 2012 389.0000 462.0000 291.0000 247.0000 500.0000 648.0000 1167.0000  
## 2013 388.0000 300.0000 280.0000 232.0000 285.0000 830.0000 1358.0000  
## 2014 457.9237 381.3183 329.0000 400.0000 487.0000 942.0000 1493.0000  
## 2015 781.0000 873.0000 876.0000 507.0000 482.0000 804.0000 1817.0000  
## 2016 555.0000 510.0000 416.0000 408.0000 738.0000 890.9423 1378.0000  
## 2017 599.0000 530.0000 450.0000 366.0000 383.3351 792.0000 1508.0000  
## 2018 652.0000 437.0000 350.0000 249.9145 290.9522 727.0000 1784.0000  
## 2019 801.0000 532.0000 510.0000 342.0000 727.0000   
## Aug Sep Oct Nov Dec  
## 2009 280.0000  
## 2010 1214.0000 1273.0000 580.0000 245.0000 352.0000  
## 2011 1023.0000 1432.0000 503.0000 289.0000 430.0000  
## 2012 1863.0000 1133.0000 438.0000 291.0000 343.0000  
## 2013 1285.0000 927.0000 378.0000 130.0000 367.0000  
## 2014 1280.0000 1543.0000 548.0000 396.0000 504.0000  
## 2015 1733.0000 1063.0000 447.0000 468.0000 327.0000  
## 2016 2218.0000 1539.0000 1211.0000 720.0000 545.0000  
## 2017 1426.0000 1155.0000 660.0000 467.0000 749.0000  
## 2018 2570.0000 797.0000 891.0000 369.0000 733.0000  
## 2019

px = plotts.sample.wge(x)



# ARIMA Model

## Overfit table

* We see mild wandering behavior, but more than that data shows seasonality with s = 12. To verify this, we will use overfit table with s = 15

factor.wge.season(12)

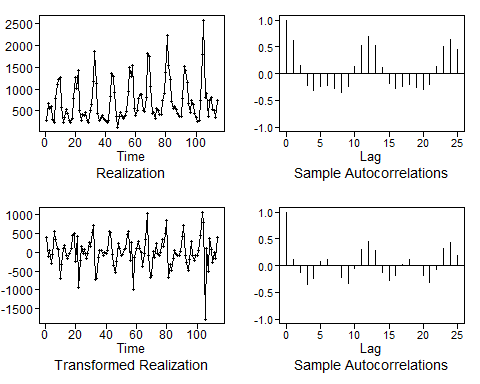
## --------------------------------------------------  
## Printing Factors for Seasonality 's' = 12  
## --------------------------------------------------  
##   
## Coefficients of Original polynomial:   
## 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000   
##   
## Factor Roots Abs Recip System Freq   
## 1-1.0000B+1.0000B^2 0.5000+-0.8660i 1.0000 0.1667  
## 1-1.0000B 1.0000 1.0000 0.0000  
## 1-1.7321B+1.0000B^2 0.8660+-0.5000i 1.0000 0.0833  
## 1+1.0000B+1.0000B^2 -0.5000+-0.8660i 1.0000 0.3333  
## 1-0.0000B+1.0000B^2 0.0000+-1.0000i 1.0000 0.2500  
## 1+1.7321B+1.0000B^2 -0.8660+-0.5000i 1.0000 0.4167  
## 1+1.0000B -1.0000 1.0000 0.5000  
##   
##

e.burg = est.ar.wge(x, p = 15, type = 'burg')

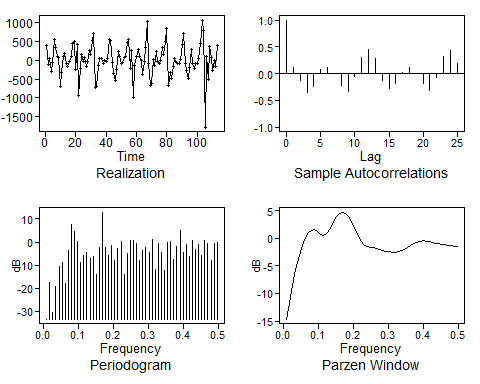
##   
## Coefficients of Original polynomial:   
## 0.3515 0.0333 -0.0236 0.0110 0.0647 -0.0780 -0.0270 -0.0469 -0.1439 0.1016 0.2480 0.2480 0.1343 -0.1363 -0.1349   
##   
## Factor Roots Abs Recip System Freq   
## 1-1.7198B+0.9858B^2 0.8723+-0.5035i 0.9929 0.0833  
## 1-0.9690B+0.9834B^2 0.4927+-0.8798i 0.9917 0.1688  
## 1-0.0691B+0.8364B^2 0.0413+-1.0927i 0.9145 0.2440  
## 1-0.9117B 1.0968 0.9117 0.0000  
## 1+1.2906B+0.7621B^2 -0.8467+-0.7715i 0.8730 0.3824  
## 1+0.5719B+0.7182B^2 -0.3982+-1.1108i 0.8475 0.3048  
## 1+1.4470B+0.5919B^2 -1.2223+-0.4420i 0.7694 0.4448  
## 1+0.7548B -1.3248 0.7548 0.5000  
## 1-0.7463B 1.3400 0.7463 0.0000  
##   
##

* Just considering ARIMA, we dont see a very strong 1-B factor. It is present with a Abs Reciprocal of 0.91, so we could still consider removing it.

x.d1 = artrans.wge(x, phi.tr = 1)



px = plotts.sample.wge(x.d1)



* THis shows a more sinusoidal behavior. It also lookks sort of stationary. We will try to model it using aic5

## ARMA Estimation

## Increased limit since lot of p = 4 and 5 were showing up in top 5.   
## Also used BIC for a smaller model  
aic5.wge(x.d1, p = 0:8, q = 0:3, type = 'bic')

## ---------WORKING... PLEASE WAIT...   
##   
##   
## Error in aic calculation at 8 2   
## Error in aic calculation at 8 3   
## Five Smallest Values of bic

## p q bic  
## 34 8 1 11.68706  
## 10 2 1 11.80833  
## 30 7 1 11.94985  
## 17 4 0 12.03088  
## 13 3 0 12.03682

e = est.arma.wge(x.d1, p = 8, q = 1)

##   
## Coefficients of Original polynomial:   
## 0.4833 -0.1660 -0.2849 -0.2112 -0.0093 -0.1058 -0.1334 -0.3555   
##   
## Factor Roots Abs Recip System Freq   
## 1-1.6869B+0.9362B^2 0.9009+-0.5064i 0.9676 0.0815  
## 1-0.8750B+0.8908B^2 0.4911+-0.9388i 0.9438 0.1733  
## 1+1.4843B+0.6649B^2 -1.1162+-0.5080i 0.8154 0.4320  
## 1+0.5943B+0.6411B^2 -0.4635+-1.1597i 0.8007 0.3105  
##   
##

factor.wge(e$theta)

##   
## Coefficients of Original polynomial:   
## 0.8984   
##   
## Factor Roots Abs Recip System Freq   
## 1-0.8984B 1.1131 0.8984 0.0000  
##   
##

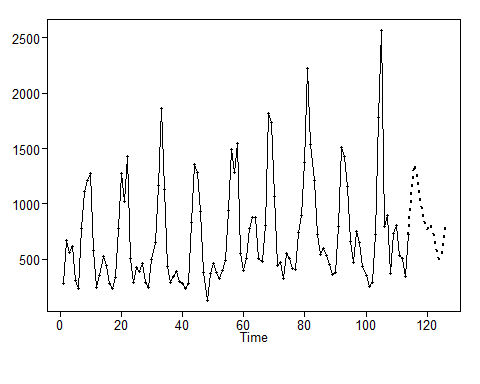
e$avar

## [1] 78331.4

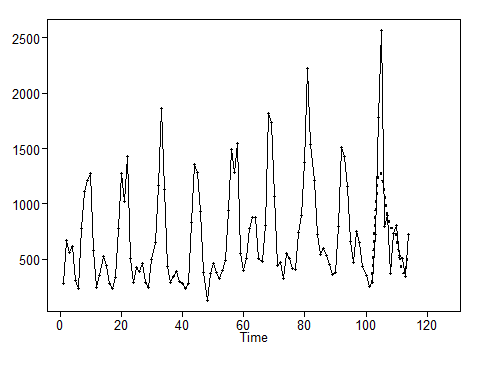
* Although there are a couple of weak factors, the 2 main ones match the frequency peaks that show up in the Spectral Density of the differenced data. Also, the theta value points to a dip at f = 0 which again matches with what we have in the Spectral Density.
* Hence we will stick to this model.
* (1-B)(1 -0.4833B +0.1660B2 +0.2849B3 +0.2112B4 +0.0093B5 +0.1058B6 +0.1334B7 +0.3555B8) Xt = (1-0.8984B)at
* (1-B)(1-1.6869B+0.9362B2)(1-0.8750B+0.8908B2)(1+1.4843B+0.6649B2)(1+0.5943B+0.6411B2) Xt = (1-0.8984B)at

## Prediction

f = fore.aruma.wge(x, phi = e$phi, theta = e$theta, d = 1, s = 0, n.ahead = 12, limits = FALSE)



f = fore.aruma.wge(x, phi = e$phi, theta = e$theta, d = 1, s = 0, n.ahead = 12, limits = FALSE, lastn = TRUE)



r = sliding\_ase(x, phi = e$phi, theta = e$theta, d = 1, s = 0, n.ahead = 12, batch\_size = 60)  
mean(r$ASEs)

## [1] 131793.9

sd(r$ASEs)

## [1] 65168.51

# Seasonal ARIMA Model

## Overfit Tables

* We see mild wandering behavior, but more than that data shows seasonality with s = 12. To verify this, we will use overfit table with s = 15

factor.wge.season(12)

## --------------------------------------------------  
## Printing Factors for Seasonality 's' = 12  
## --------------------------------------------------  
##   
## Coefficients of Original polynomial:   
## 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000   
##   
## Factor Roots Abs Recip System Freq   
## 1-1.0000B+1.0000B^2 0.5000+-0.8660i 1.0000 0.1667  
## 1-1.0000B 1.0000 1.0000 0.0000  
## 1-1.7321B+1.0000B^2 0.8660+-0.5000i 1.0000 0.0833  
## 1+1.0000B+1.0000B^2 -0.5000+-0.8660i 1.0000 0.3333  
## 1-0.0000B+1.0000B^2 0.0000+-1.0000i 1.0000 0.2500  
## 1+1.7321B+1.0000B^2 -0.8660+-0.5000i 1.0000 0.4167  
## 1+1.0000B -1.0000 1.0000 0.5000  
##   
##

e.burg = est.ar.wge(x, p = 15, type = 'burg')

##   
## Coefficients of Original polynomial:   
## 0.3515 0.0333 -0.0236 0.0110 0.0647 -0.0780 -0.0270 -0.0469 -0.1439 0.1016 0.2480 0.2480 0.1343 -0.1363 -0.1349   
##   
## Factor Roots Abs Recip System Freq   
## 1-1.7198B+0.9858B^2 0.8723+-0.5035i 0.9929 0.0833  
## 1-0.9690B+0.9834B^2 0.4927+-0.8798i 0.9917 0.1688  
## 1-0.0691B+0.8364B^2 0.0413+-1.0927i 0.9145 0.2440  
## 1-0.9117B 1.0968 0.9117 0.0000  
## 1+1.2906B+0.7621B^2 -0.8467+-0.7715i 0.8730 0.3824  
## 1+0.5719B+0.7182B^2 -0.3982+-1.1108i 0.8475 0.3048  
## 1+1.4470B+0.5919B^2 -1.2223+-0.4420i 0.7694 0.4448  
## 1+0.7548B -1.3248 0.7548 0.5000  
## 1-0.7463B 1.3400 0.7463 0.0000  
##   
##

* I only see a few of the factors show up from s = 12.
* The only factors that are appreciable are 1-1.7321B+1.0000B^2 and 1-1.0000B+1.0000B^2
* Hence only these should be used to transform the data

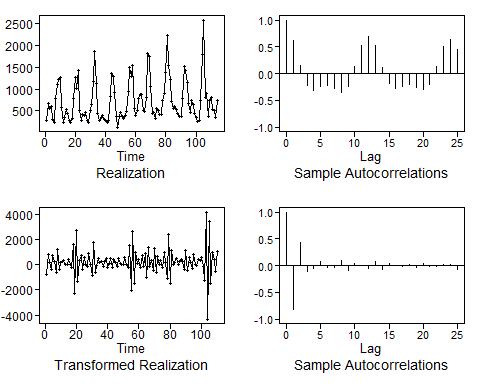
factors = mult.wge(fac1 = c(1.732, -1), fac2 = c(1, -1))  
factors$model.coef

## [1] 2.732 -3.732 2.732 -1.000

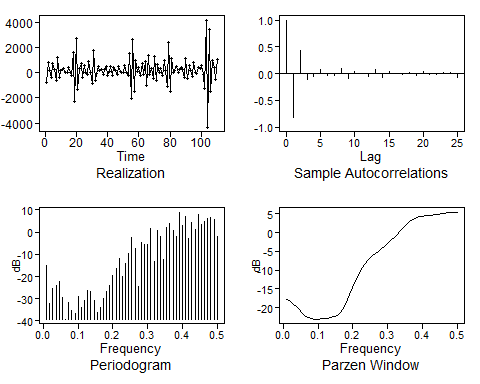
factor.wge(factors$model.coef)

##   
## Coefficients of Original polynomial:   
## 2.7320 -3.7320 2.7320 -1.0000   
##   
## Factor Roots Abs Recip System Freq   
## 1-1.7320B+1.0000B^2 0.8660+-0.5000i 1.0000 0.0833  
## 1-1.0000B+1.0000B^2 0.5000+-0.8660i 1.0000 0.1667  
##   
##

x.tr = artrans.wge(x, phi.tr = factors$model.coef)



px = plotts.sample.wge(x.tr)



* This looks very much stationary. Lets model this using AIC5

aic5.wge(x.tr, p = 0:10, q = 0:3, type = 'bic')

## ---------WORKING... PLEASE WAIT...   
##   
##   
## Five Smallest Values of bic

## p q bic  
## 27 6 2 11.34780  
## 31 7 2 11.36349  
## 33 8 0 11.38345  
## 41 10 0 11.41778  
## 35 8 2 11.41831

## ARMA Estimation

e = est.arma.wge(x.tr, p = 6, q = 2)

##   
## Coefficients of Original polynomial:   
## -1.1740 -0.5711 0.2901 0.8045 0.7885 0.3503   
##   
## Factor Roots Abs Recip System Freq   
## 1-0.9458B 1.0573 0.9458 0.0000  
## 1+0.1188B+0.7189B^2 -0.0826+-1.1765i 0.8479 0.2612  
## 1+1.2366B+0.6741B^2 -0.9173+-0.8013i 0.8210 0.3857  
## 1+0.7644B -1.3083 0.7644 0.5000  
##   
##

factor.wge(e$theta)

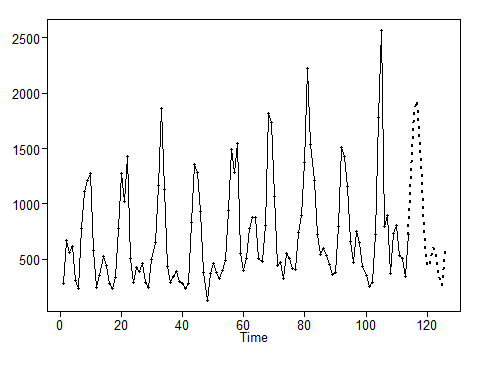
##   
## Coefficients of Original polynomial:   
## 1.2042 -0.6520   
##   
## Factor Roots Abs Recip System Freq   
## 1-1.2042B+0.6520B^2 0.9234+-0.8252i 0.8075 0.1161  
##   
##

e$avar

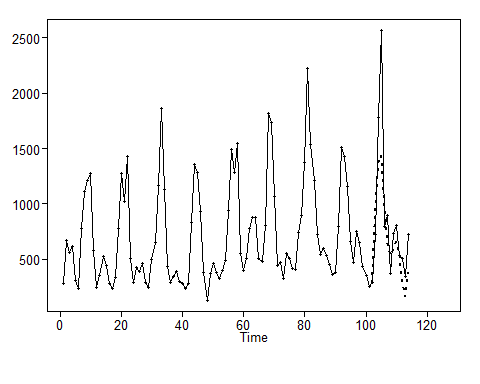
## [1] 57711.92

## Prediction

f = fore.aruma.wge(x, phi = e$phi, theta = e$theta, d = 0, s = 0, lambda = factors$model.coef, n.ahead = 12, limits = FALSE)



f = fore.aruma.wge(x, phi = e$phi, theta = e$theta, d = 0, s = 0, lambda = factors$model.coef, n.ahead = 12, limits = FALSE, lastn = TRUE)



r = sliding\_ase(x, phi = e$phi, theta = e$theta, d = 0, s = 0, n.ahead = 12, batch\_size = 60, lambda = factors$model.coef)  
mean(r$ASEs)

## [1] 88299.72

sd(r$ASEs)

## [1] 36460.07

# Signal Plus Noise Model

t = seq(1,length(x),1)  
df = data.frame(x = x, t= t)  
fit = lm(x~t, data = df)  
summary(fit)

##   
## Call:  
## lm(formula = x ~ t, data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -573.4 -333.3 -178.9 166.6 1737.3   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 585.919 87.831 6.671 1.01e-09 \*\*\*  
## t 2.351 1.326 1.773 0.0789 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 465.8 on 112 degrees of freedom  
## Multiple R-squared: 0.0273, Adjusted R-squared: 0.01862   
## F-statistic: 3.143 on 1 and 112 DF, p-value: 0.07895

* Does not show a linear trend.

cfit = cochrane.orcutt(fit)   
summary(cfit)

## Call:  
## lm(formula = x ~ t, data = df)  
##   
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 619.1313 185.7898 3.332 0.00117 \*\*  
## t 1.8810 2.7357 0.688 0.49315   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 369.4567 on 111 degrees of freedom  
## Multiple R-squared: 0.0042 , Adjusted R-squared: -0.0047  
## F-statistic: 0.5 on 1 and 111 DF, p-value: < 4.932e-01  
##   
## Durbin-Watson statistic   
## (original): 0.77529 , p-value: 1.086e-11  
## (transformed): 1.50544 , p-value: 2.985e-03

* Does not show a lineear trend in the data. Since the trend is not fouund, we will not continue the fitting process using this method.