Live Session Assignment 7

Nikhil Gupta

2020-02-17 03:18:58

Table of Contents

# Setup

library(tswge)

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

library(dplyr)

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

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

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 readr 1.3.1  
## v tibble 2.1.3 v purrr 0.3.2  
## v tidyr 0.8.3 v stringr 1.4.0  
## v ggplot2 3.2.0 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 '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()

source("common\_functions.R")

# For pre-live

## 2

factors = mult.wge(fac1 = 0.9, fac2 = 0.8)  
phi = factors$model.coef  
# factor.wge(phi = phi) # Cross check  
  
psi\_wts = psi.weights.wge(phi = phi, lag.max = 5)  
cat("\nPsi Weights: \n")

##   
## Psi Weights:

print(psi\_wts)

## [1] 1.70000 2.17000 2.46500 2.62810 2.69297

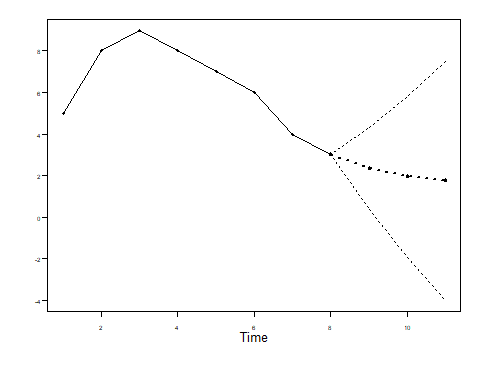
l = 3  
# For 95% interval of X\_t0(3), we need lags will l-1  
multiplier = 1.96 \* sqrt(1 + sum(psi\_wts[1:l-1]^2))  
cat("\nMultiplier: ")

##   
## Multiplier:

cat(multiplier)

## 5.747481

# We are still missing the a\_t term. How do we compute this?  
# Maybe get from the forecast function  
  
x = c(5, 8, 9, 8, 7, 6, 4, 3)  
f = fore.arma.wge(x, phi = phi, n.ahead = 3, limits = TRUE)



vara = f$wnv  
cat("\nvara: ")

##   
## vara:

cat(vara)

## 0.9958917

half\_width = sqrt(vara) \* multiplier  
cat("\nHalf Width: ")

##   
## Half Width:

cat(half\_width)

## 5.735662

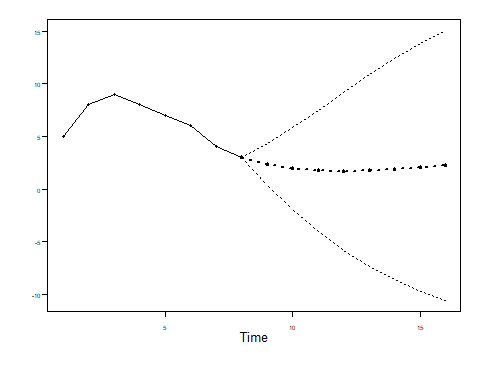
# Cross Check  
half\_width\_computed = f$f[3] - f$ll[3]  
cat("\nHalf Width (computed tswge): ")

##   
## Half Width (computed tswge):

cat(half\_width\_computed)

## 5.735662

n.ahead = 8  
f = fore.arma.wge(x, phi = phi, n.ahead = 8, limits = TRUE)



## 3

data = read.csv("../MSDS-6373-Time-Series/Unit 7/AmtrakPassengersMonthly.csv")  
data %>% glimpse()

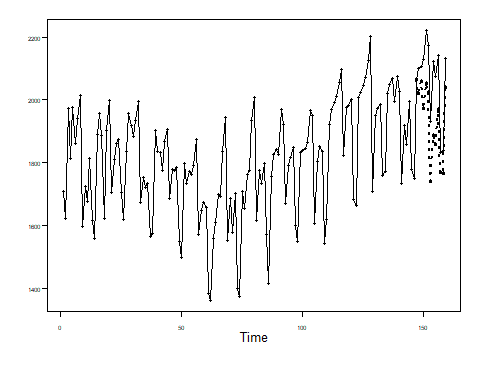
## Observations: 159  
## Variables: 5  
## $ Month <fct> Jan-91, Feb-91, Mar-91, Apr-91, May-91, Jun-91, Jul-...  
## $ Ridership <int> 1709, 1621, 1973, 1812, 1975, 1862, 1940, 2013, 1596...  
## $ X <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...  
## $ X.1 <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...  
## $ X.2 <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...

n = nrow(data)  
n.ahead = 12  
data = data$Ridership

results = tribble(~model, ~ASE)

### Model 1

phi = c(0.5511, 0.1680, -0.0145, 0.0651, 0.1388, -0.2966, 0.1539, 0.1270, -0.1815, 0.0364, 0.1456, 0.6287, -0.3832, -0.0199, -0.1679)   
theta = 0  
  
f = fore.arma.wge(data, phi=phi, theta = theta, n.ahead=n.ahead, limits=FALSE, lastn = TRUE)



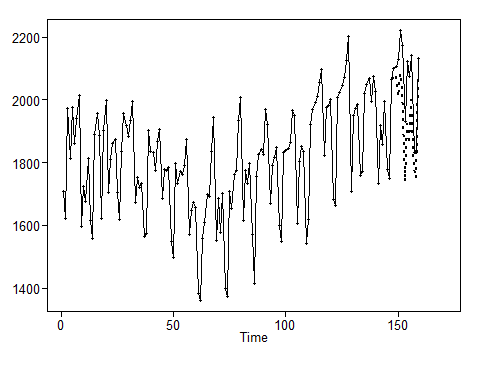
ase = mean((data[(n-n.ahead+1):n] - f$f)^2)  
ase

## [1] 22588.56

results = results %>% add\_row(model = "Model 1", ASE = ase)

### Model 2

phi = c(-0.02709541, 0.74213105)  
theta = c(-0.5844596, 0.3836931)  
  
f = fore.aruma.wge(data, phi=phi, theta = theta, s = 12, n.ahead = n.ahead, limits=FALSE, lastn = TRUE)



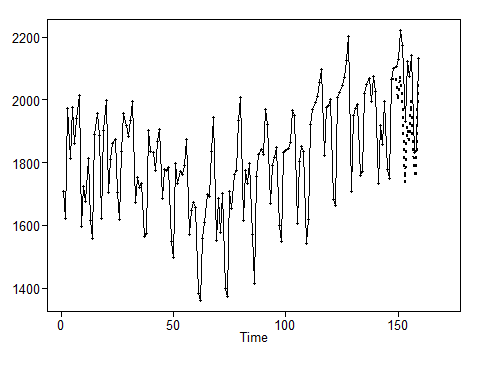
ase = mean((data[(n-n.ahead+1):n] - f$f)^2)  
ase

## [1] 17197.77

results = results %>% add\_row(model = "Model 2", ASE = ase)

### Model 3

phi = c(0.306943)  
theta = c(0.7431719)  
  
f = fore.aruma.wge(data, phi=phi, theta = theta, d = 1, s = 12, n.ahead = n.ahead, limits=FALSE, lastn = TRUE)



ase = mean((data[(n-n.ahead+1):n] - f$f)^2)  
ase

## [1] 18399.97

results = results %>% add\_row(model = "Model 3", ASE = ase)

### Results

results

## # A tibble: 3 x 2  
## model ASE  
## <chr> <dbl>  
## 1 Model 1 22589.  
## 2 Model 2 17198.  
## 3 Model 3 18400.