DS 6373: Time Series: Unit 7 HW Solutions

Below are the homework (HW) problems for this Unit. You do not need to submit the solutions rather double check your solutions to the solutions posted. Solutions will be posted to the Wall a few days after the release of the HW. This is intended to let the student think about the problem and attempt it without the temptation to first look at the solution. Please write any questions to the Wall or in an email to myself and/or bring them up during office hours or even in the next Live Session. Remember that the concepts covered below are fundamental to the course and are fair game for the midterm and final.

Have a blast!

Problems from Chapter 6 of the Textbook:
6.1

Problem 6.1

$$\hat{X}_{10}(1) = -1.1\hat{X}_{10}(0) - 1.78\hat{X}_{10}(-1) - .88\hat{X}_{10}(-2) - .64\hat{X}_{10}(-3) - .2\hat{a}_{10} - .9\hat{a}_9 + (1+1.1+1.18+.88+.64)\bar{X}$$
$$= -1.1\hat{X}_{10}(0) - 1.78X_9 - .88X_8 - .64X_7 - .2\hat{a}_{10} - .9\hat{a}_9 + 5.4\bar{X} \text{ where } \bar{X} = 39.6$$

Note, the constant $5.4\bar{X} = 213.84$ and we need $\hat{a}_{10}$ and $\hat{a}_9$

Set $\hat{a}_t = 0, t = 0,1,\ldots,4$

$\hat{a}_5 = 29.3 + 1.1(38.3) + 1.78(49.8) + .88(30.5) + .64(40) - 213.84 = -1.326$

$\hat{a}_6 = 48.7 + 1.1(29.3) + 1.78(38.3) + .88(49.8) + .64(30.5) + .2(-1.326) - 213.84 = -1.6572$
$\hat{a}_7 = 39.2 + 1.1(48.7) + 1.78(29.3) + .88(38.3) + .64(49.8) + .2(-1.326) - .9(-1.6572) - 213.84 = -2.47804$

Continuing in this manner we get $\hat{a}_8 = -1.042128$, $\hat{a}_9 = 0.535704$, and $\hat{a}_{10} = 2.405$

$\hat{X}_{10}(1) = -1.1X_{10} - 1.78X_9 - .88X_8 - .64X_7 - .2\hat{a}_{10} + .9\hat{a}_9 + 213.84 = 32.159$
$\hat{X}_{10}(2) = -1.1\hat{X}_{10}(1) - 1.78X_{10} - .88X_9 - .64X_8 + .9\hat{a}_{10} + 213.84 = 44.3016$
$\hat{X}_{10}(3) = -1.1\hat{X}_{10}(2) - 1.78\hat{X}_{10}(1) - .88X_{10} - .64X_9 - 213.84 = 41.04922$
$\hat{X}_{10}(4) = -1.1\hat{X}_{10}(3) - 1.78\hat{X}_{10}(2) - .88\hat{X}_{10}(1) - .64X_{10} - 213.84 = 34.393$

ψ -weights:

$\psi_0 = 1, \psi_1 = -1.3, \psi_2 = .55, \psi_3 = .829$

$\hat{\sigma}_a^2 = \frac{1}{6}\sum_{k=1}^{10}\hat{a}_k^2 = 2.967, \quad \hat{\sigma}_a = 1.722$

95% prediction limits: Note that (1.96)(1.722)=3.376
ℓ=1: 32.1592 ± 3.376 or (28.7832, 35.5352)

ℓ=2: $44.302 \pm 3.376\sqrt{1+1.3^2} = 44.302 \pm 5.537$ or (38.765, 49.839)

ℓ=3: $41.049 \pm 3.376\sqrt{1+1.3^2+.55^2} = 41.049 \pm 5.84$ or (35.209, 46.889)

ℓ=4: $34.393 \pm 3.376\sqrt{1+1.3^2+.55^2+.829^2} = 34.393 \pm 6.476$ or (27.917, 40.869)

(b) Using the code

```
x=c(40,30.5,49.8,38.3,29.3,48.7,39.2,31.7,46.1,42.4)
ff=fore.arma.wge(x, phi=c(-1.1,-1.78,-.88,-.64), theta=c(.2,-.9),
   n.ahead=4,lastn=FALSE)
```

The output below matches the hand-calculations above.

```
ff$f
[1] 32.15921 44.30143 41.04902 34.39342

ff$ll
[1] 28.78304 38.76410 35.20864 27.91702

ff$ul
[1] 35.53539 49.83877 46.88941 40.86981

ff$resid
 [1]  0.0000000  0.0000000  0.0000000  0.0000000 -1.3260000 -1.6572000
     -2.4780400
 [8] -1.0421280  0.5358104  2.4050773

ff$wnv
[1] 2.967132

$psi
[1] -1.3000  0.5500  0.8290 -1.3869
```

6.3 (For part c include the ASE)

Problem 6.3 Using

`mult.gwe(fac1=.8,fac2=c(1,-.9))` we see that $(1-.8B)(1-B+.9B^2)=(1-1.8B+1.7B^2-.72B^3)$

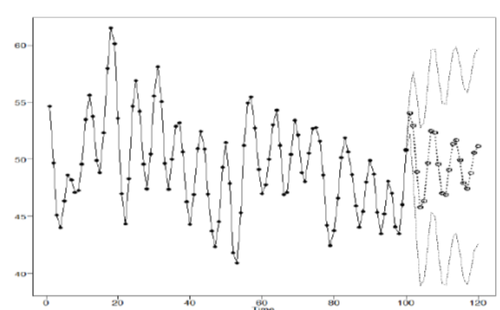
```
x=gen.arma.wge(n=100, phi=c(1.8,-1.7,.72),theta=-.5)
x50=x+50
```

`x50` is the desired data set.

(a-b) We forecast using the command

```
ff=fore.arma.wge(x50, phi=c(1.8,-1.7,.72),theta=-.5,n.ahead=20,lastn=FALSE)
```

The following is a plot of the forecasts and 95% prediction limits.



The desired forecasts are, and they have the appearance of a damped sinusoid. The effect of the factor $(1-.8B)$ is not apparent in the forecasts.

```
ff$f
```

[1] 54.07549 52.97511 48.91969 45.83055 46.37203 49.67835 52.48500 52.30612

[9] 49.59337 47.03529 46.91366 49.09025 51.37308 51.69440 49.95910 47.93296

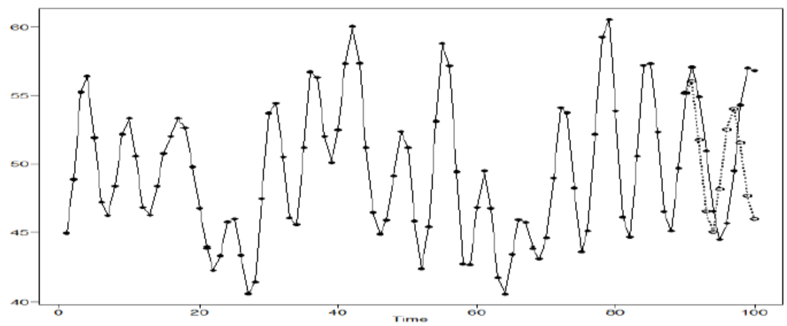
[17] 47.46727 48.82404 50.59910 51.15237

(c-d) We forecast the last 10 values and then forecast the last 10 values showing 95% limits with the following two commands:

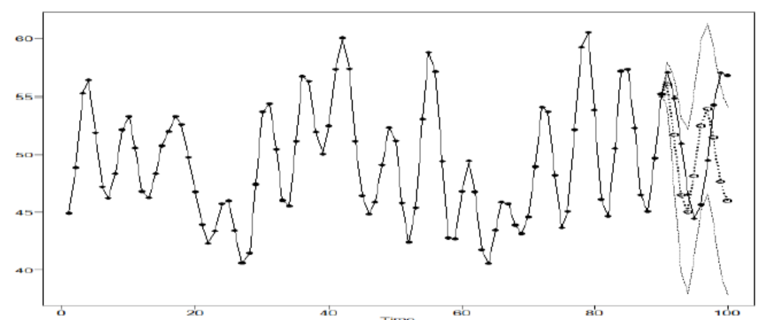
```
ff=fore.arma.wge(x50, phi=c(1.8,-1.7,.72),theta=-.5,n.ahead=100,lastn=FALSE,
  limits=FALSE)
```

```
ff=fore.arma.wge(x50, phi=c(1.8,-1.7,.72),theta=-.5,n.ahead=100,lastn=FALSE,
  limits=TRUE)
```

The plots are shown below where it appears that the forecasts track fairly well with the actual data although the last 2 values are poorly forecast.



The forecasts showing 95% prediction limits are shown below:



Visually, it appears that all 10 forecasts stay within limits.

Forecast	Lower	Upper	Actual
56.02	54.10	57.94	57.05
51.74	46.91	56.56	54.91
46.52	39.79	53.26	50.94
45.04	37.92	52.16	46.55
48.14	41.10	55.29	44.45
52.50	45.16	59.84	45.65
54.00	46.65	61.35	49.50
51.52	43.94	59.11	54.30
47.66	39.62	55.70	57.00
45.99	37.82	54.15	56.80

In the table it can be seen that the last two actual values fall outside the forecast limits.

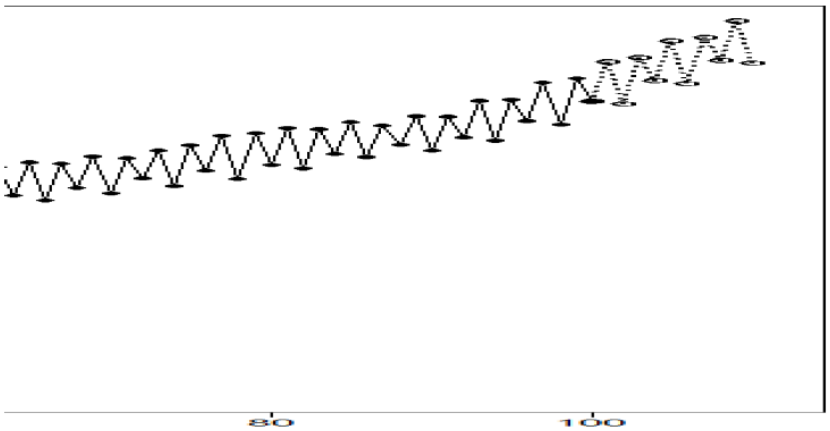
6.4

Problem 6.4

We generate the series using `gen.aruma.wge`:

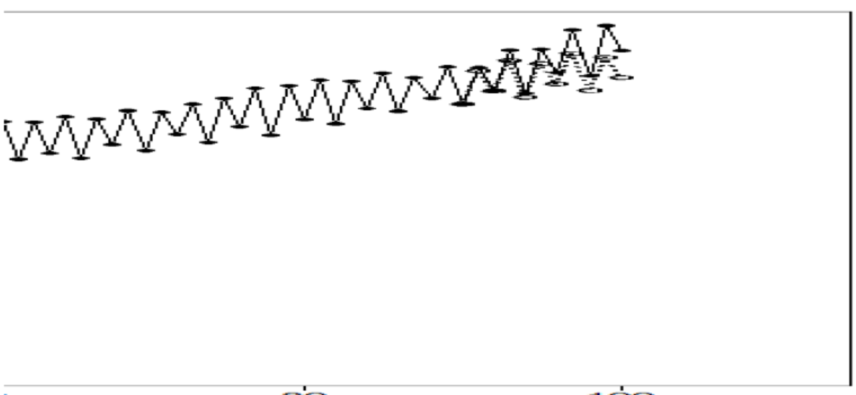
```
x=gen.aruma.wge(n=100,phi=-.5,d=1,s=4)
x50=x+50
```

(a) `fore.aruma.wge(x50,phi=-.5,d=1,s=4,n.ahead=10,lastn=FALSE,limits=FALSE)`



The forecasts look good and track both the seasonal and upward trending behavior.

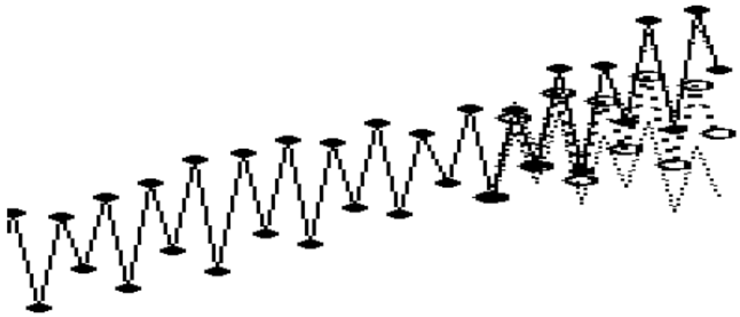
(b) `fore.aruma.wge(x50,phi=-.5,d=1,s=4,n.ahead=10,lastn=TRUE,limits=FALSE)`



The forecasts track the seasonal component of the data but do not show an upward trend in the last 4 values well.

(c) `fore.aruma.wge(x50,phi=-.5,d=1,s=4,n.ahead=10,lastn=TRUE,limits=TRUE)`

It appears that the last 4 values fall very close to the upper limits. The actual values are given in the table below where it can be seen that only 6 of the actual values fall within the limits. This seems to be because the series took an upward turn that was predicted by the last few data values prior to the forecasts.



Forecast	Lower	Upper	Actual
239.75	237.65	241.84	240.73
233.08	230.74	235.42	233.28
243.19	240.37	246.01	246.44
231.01	227.90	234.12	232.36
242.01	237.30	246.72	246.94
235.35	230.06	240.65	239.20
245.45	239.38	251.53	253.12
233.28	226.63	239.92	238.11
244.28	236.00	252.56	254.64
237.62	228.51	246.74	246.31

6.7

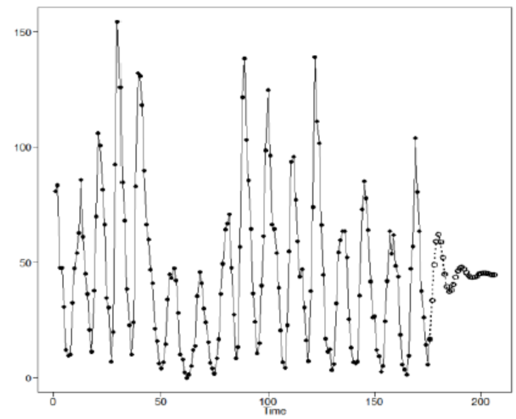
Problem 6.7

We use the following code:

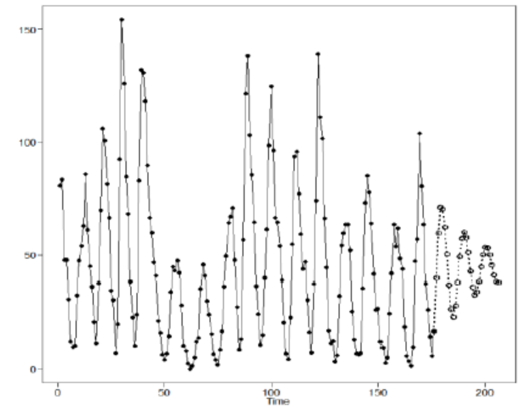
```
data(sunspot.classic)
phiA=c(1.42,-.73)
phiB=c(1.23,-.47,-.14,.16,-.14,.07,-.13,.21)
phiC=c(-.37,-.11,.03,.29,.31,.27)
lamC=c(1.646,-1)
fore.arma.wge(sunspot.classic,phi=phiA,n.ahead=30,lastn=FALSE,limits=FALSE)
fore.arma.wge(sunspot.classic,phi=phiB,n.ahead=30,lastn=FALSE,limits=FALSE)
fore.aruma.wge(sunspot.classic,phi=phiC,lambda=lamC,n.ahead=30,lastn=FALSE,limits=FALSE)
```

(a) (i) The forecasts from the three models are shown below:

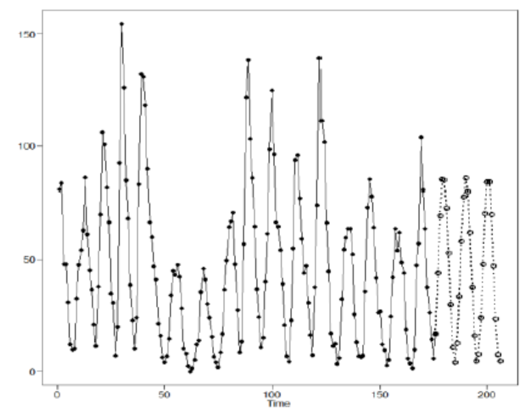
Model A



Model B



Model C



(ii) As we have discussed throughout the text, the way to understand ARMA models is to use the factor table. The factor tables for the three models are given below:

MODEL A

Factor	Roots	Abs Recip	System Freq
1-1.4200B+0.7300B^2	0.9726+-0.6511i	0.8544	0.0939

MODEL B

Factor	Roots	Abs Recip	System Freq
1-1.5574B+0.8966B^2	0.8685+-0.6008i	0.9469	0.0963
1-0.8811B	1.1349	0.8811	0.0000
1-0.4157B+0.6579B^2	0.3160+-1.1917i	0.8111	0.2088
1+0.7981B	-1.2530	0.7981	0.5000
1+0.8262B+0.5063B^2	-0.8159+-1.1443i	0.7115	0.3486

MODEL C

```
Factor                    Roots                 Abs Recip    System Freq
1-1.6460B+1.0000B^2       0.8230+-0.5680i       1.0000       0.0961
1-0.8960B                 1.1160                0.8960       0.0000
1-0.4123B+0.6928B^2       0.2975+-1.1640i       0.8324       0.2102
1+0.8046B                -1.2428                0.8046       0.5000
1+0.8737B+0.5405B^2      -0.8082+-1.0940i       0.7352       0.3513
```

All three models have a dominant factor associated with a frequency of about $f_0 = .096$ (a period of about 10.5 years). For Model A this factor is associated with roots not very close to the unit circle ($|r^{-1}| = .854$). Consequently, this model will be associated with autocorrelations (and forecasts) that damp rather quickly. In Model B, the roots associated with this factor are much closer to the unit circle ($|r^{-1}| = .947$) and consequently the cyclic behavior in the forecasts (and autocorrelations) damps more slowly. Models A and B are stationary models. Model C is a nonstationary model where the factor associated with the cyclic behavior has roots on the unit circle. Thus, the forecasts do not damp and essentially predict the last cycle to continue.

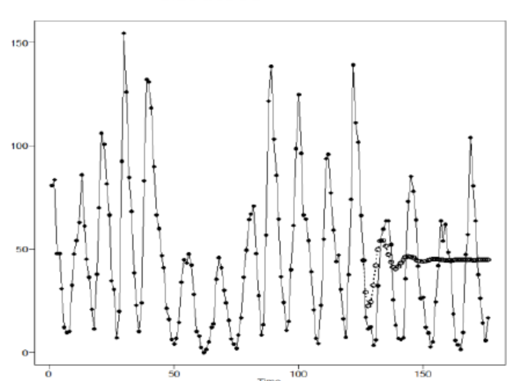
(iii) From the forecasts above, it appears that Model C followed by Model B do the best job of forecasting. They seem to be more desirable because of the known pseudo-cyclic behavior of the sunspot data. The forecasts for Model A tend to damp too quickly and the forecasts seem unreasonable.

(b) We use the following code:

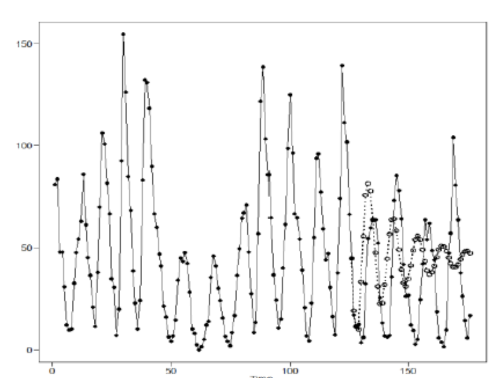
```
data(sunspot.classic)
phiA=c(1.42,-.73)
phiB=c(1.23,-.47,-.14,.16,-.14,.07,-.13,.21)
phiC=c(-.37,-.11,.03,.29,.31,.27)
lamC=c(1.646,-1)
fore.arma.wge(sunspot.classic,phi=phiA,n.ahead=50,lastn=TRUE,limits=FALSE)
fore.arma.wge(sunspot.classic,phi=phiB,n.ahead=50,lastn=TRUE,limits=FALSE)
fore.aruma.wge(sunspot.classic,phi=phiC,lambda=lamC,n.ahead=50,lastn=TRUE,limits=FALSE)
```

(i) The forecasts are plotted below

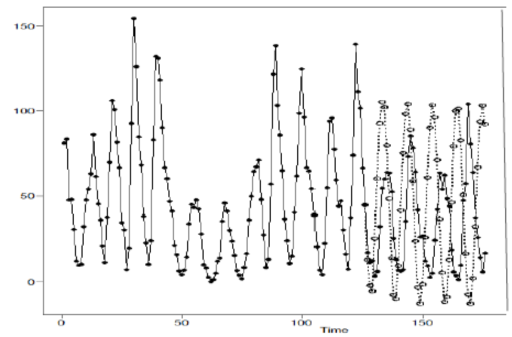
Model A



Model B



Model C



(ii) While the forecasts from Model A (give up) and begin forecasting the mean, it can be seen that using either of the other two models to forecast too many steps ahead can be dangerous. This is because of the pseudo-periodic nature of the data. While near term forecasts with Models B and C seem to be preferred to those from Model A, we see that, for example, forecasts from the other two models are completely out of phase with the sunspot data (i.e. they forecast a peak where there is a trough and vice versa).

(iii) Models B and C seem preferable for shorter term forecasts, but for longer steps ahead, it is obvious that the best strategy might be to just forecast the mean (as does Model A).

6.8

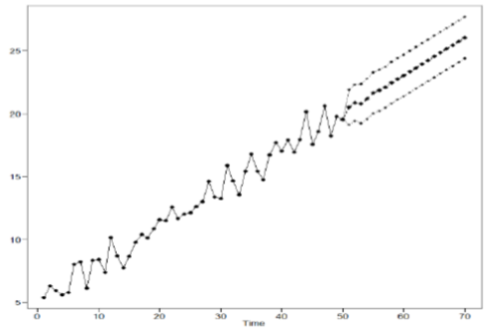
Problem 6.8

(a) We generate and forecast based on the model in (a) using the following code:

```
spna=gen.sigplusnoise.wge(n=50,b0=5,b1=.3,coef=c(0,0),freq=c(0,0),
    psi=c(0,0),phi=c(-.2,-.6),vara=.5,plot=TRUE,sn=0)
fore.sigplusnoise.wge(x=spna,linear=TRUE,freq=0,max.p=5,n.ahead=20,
    lastn=FALSE,plot=TRUE,limits=TRUE)
```

The plot of the data, forecasts and limits are shown below. Note that the fitted line and the model fit to the residuals from the line are assumed to be true values when computing the forecasts. We allowed a model up to order AR(5) and the model correctly chose an AR(2) for the residuals. Note that there is very little expansion of the lengths of the forecast limits since the model fit to the residuals is $Z_t + .31Z_{t-1} + .51Z_{t-2} = a_t$ which has a factor table shown below that has a pair of complex roots quite far removed from the unit circle. It should also be noted that these limits assume the existence of a linear trend. If the data were actually from an AR model, for example, with a root quite close to +1, then the forecasts and limits shown below would be "overly optimistic."

```
Factor                  Roots                Abs Recip    System Freq
1+0.3100B+0.5100B^2   -0.3039+-1.3669i       0.7141       0.2848
```



(b) We generate and forecast based on the model in (b) using the following code:

```
spnb=gen.sigplusnoise.wge(n=100,b0=50,b1=0,coef=c(5,0),freq=c(.1,0),
    psi=c(2.5,0),phi=-.7,vara=1,plot=TRUE,sn=0)
fore.sigplusnoise.wge(x=spnb,linear=FALSE,freq=.10,max.p=5,n.ahead=20,
    lastn=FALSE,plot=TRUE,limits=TRUE)
```

The plot of the data, forecasts and limits are shown below. Note that we assume that we know $f_0 = .1$. Given cyclic data, a good estimate of f_0 could be obtained from a spectral estimate or factor table. If the data truly are cyclic (and do not have phase shifts) then the forecasts of the last k values should be much better than that which was seen for the `sunspot.classic` data. Again, this is assuming the cyclic behavior is actually due to a cyclic component.