Homework 7

Nikhil Gupta

2020-02-22 10:26:33

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()

source("../Code/common\_functions.R")

# Problem 6.1

x = c(40, 30.5, 49.8, 38.3, 29.3, 48.7, 39.2, 31.7, 46.1, 42.4)  
phi = c(-1.1, -1.78, -0.88, -0.64)  
theta = c(0.2, -0.9)

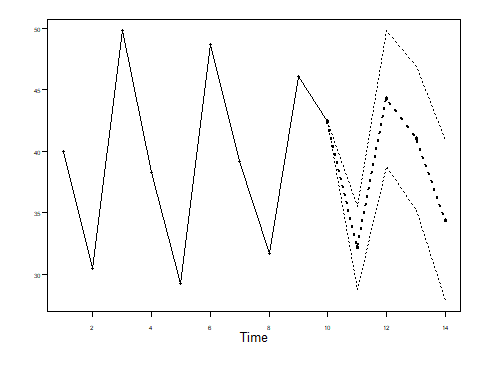
## a

compute\_a = function(x, phi, theta, index){  
   
 ## Limit 1  
 get\_a\_i\_lesseq\_p = function(index){  
 all\_a = rep(0,index) # Return all 0s till the index  
 return(all\_a)  
 }  
   
 ## Limit 2  
 get\_a\_i\_lesseq\_lenX = function(x, p, q, phi, theta, index){  
 all\_a = get\_a\_i\_lesseq\_p(p) ## get all 0's till p  
   
 ## Then compute the values one by one till needed  
 limit = min(index, length(x))  
 for (i in (p+1):limit){  
 a = x[i] - sum(phi \* x[(i-1):(i-p)]) + sum(theta \* all\_a[(i-1):(i-q)]) - (1 - sum(phi)) \* mean(x)  
 all\_a = c(all\_a, a)  
 }  
   
 return(all\_a)  
 }  
   
 ## Limit 3  
 get\_a\_i\_more\_lenX = function(x, p, q, phi, theta, index){  
 n = length(x)  
 all\_a = get\_a\_i\_lesseq\_lenX(x, p, q, phi, theta, n) ## Get all values till len(x)  
 all\_a = c(all\_a, rep(0, (index-n))) ## Then append 0s after that  
 return(all\_a)  
 }  
   
   
 p = length(phi)  
 q = length(theta)  
  
 all\_a = c()  
   
 if (index <= p){  
 all\_a = get\_a\_i\_lesseq\_p(index)  
 }  
 else if (index <= length(x)){  
 all\_a = get\_a\_i\_lesseq\_lenX(x, p, q, phi, theta, index)  
 }  
 else{  
 all\_a = get\_a\_i\_more\_lenX(x, p, q, phi, theta, index)  
 }  
   
 return(all\_a)  
}  
  
compute\_vara = function(all\_a){  
 subset = all\_a[all\_a != 0]  
 len\_subset = length(subset)  
 vara = sum(subset^2)/length(subset)  
 return(vara)  
}  
  
compute\_stda = function(all\_a){  
 return(sqrt(compute\_vara(all\_a)))  
}  
  
return\_all\_a\_calc = function(x, phi, theta, index){  
 all\_a = compute\_a(x = x, phi = phi, theta = theta, index = index)  
 vara = compute\_vara(all\_a)  
 stda = compute\_stda(all\_a)  
 return(list(all\_a = all\_a, vara = vara, stda = stda))  
}  
  
rv = return\_all\_a\_calc(x, phi, theta, 11)  
rv

## $all\_a  
## [1] 0.0000000 0.0000000 0.0000000 0.0000000 -1.3260000 -1.6572000  
## [7] -2.4780400 -1.0421280 0.5358104 2.4050773 0.0000000  
##   
## $vara  
## [1] 2.967132  
##   
## $stda  
## [1] 1.722536

## b

f = fore.arma.wge(x, phi = phi, theta = theta, n.ahead = 4)



f

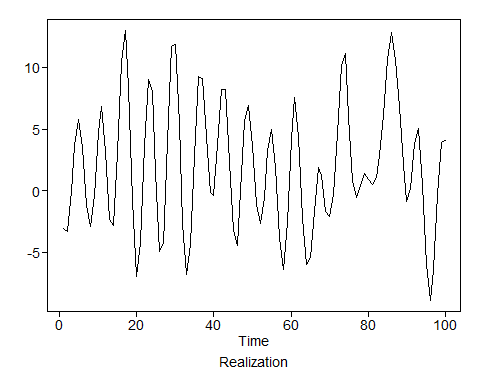
## $f  
## [1] 32.15921 44.30143 41.04902 34.39342  
##   
## $ll  
## [1] 28.78304 38.76410 35.20864 27.91702  
##   
## $ul  
## [1] 35.53539 49.83877 46.88941 40.86981  
##   
## $resid  
## [1] 0.0000000 0.0000000 0.0000000 0.0000000 -1.3260000 -1.6572000  
## [7] -2.4780400 -1.0421280 0.5358104 2.4050773  
##   
## $wnv  
## [1] 2.967132  
##   
## $se  
## [1] 1.722536 2.825170 2.979789 3.304282  
##   
## $psi  
## [1] -1.3000 0.5500 0.8290 -1.3869

# Problem 6.3

factors = mult.wge(fac1 = 0.8, fac2 = c(1,-0.9))  
phi = factors$model.coef  
theta = -0.5  
  
factor.wge(phi = phi) # Check

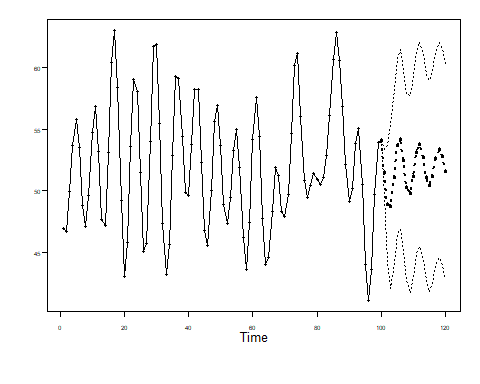
##   
## Coefficients of Original polynomial:   
## 1.8000 -1.7000 0.7200   
##   
## Factor Roots Abs Recip System Freq   
## 1-1.0000B+0.9000B^2 0.5556+-0.8958i 0.9487 0.1616  
## 1-0.8000B 1.2500 0.8000 0.0000  
##   
##

x = 50 + gen.arma.wge(n = 100, phi = phi, theta = theta)



## a

f = fore.arma.wge(x, phi = phi, theta = theta, n.ahead = 20)



f$f

## [1] 51.45545 48.84643 48.74201 51.10169 53.64815 54.14512 52.40967  
## [8] 50.27445 49.73915 51.15594 53.07884 53.74608 52.69828 51.06242  
## [15] 50.37954 51.17691 52.59524 53.30105 52.73443 51.53586

f$ul

## [1] 53.36976 53.64750 55.44035 58.18423 60.75853 61.45133 59.72200  
## [8] 57.82102 57.73965 59.27716 61.22698 62.05965 61.05425 59.45808  
## [15] 58.94737 59.81310 61.23918 62.03807 61.51632 60.32297

f$ll

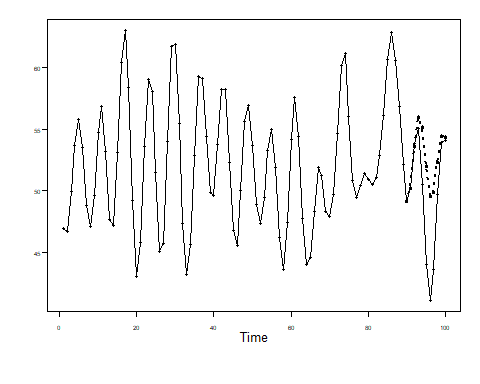
## [1] 49.54114 44.04536 42.04367 44.01915 46.53776 46.83891 45.09734  
## [8] 42.72789 41.73865 43.03473 44.93070 45.43251 44.34231 42.66675  
## [15] 41.81172 42.54072 43.95131 44.56402 43.95254 42.74874

## b

* The forecasts are damped sinusoidals and are due to the complex root in the characteristic equation (dominant root).
* The effect of the real root is not very prominent since it is further away from the unit circle.

## c

n = length(x)  
n.ahead = 10  
f = fore.arma.wge(x, phi=phi, theta = theta, n.ahead = n.ahead, limits=FALSE, lastn = TRUE)



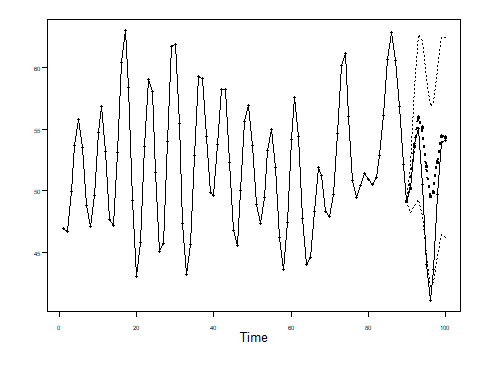
ase = mean((x[(n-n.ahead+1):n] - f$f)^2)  
ase

## [1] 20.3031

* The forecasts show sinusoidal behavior similar to the realization, but are regressing to the mean of the realization (mean = 50). This is not true of the actual realization for the same points (in that short period, they are not moving towards the mean, even though they may do that in the long run.)

## d

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



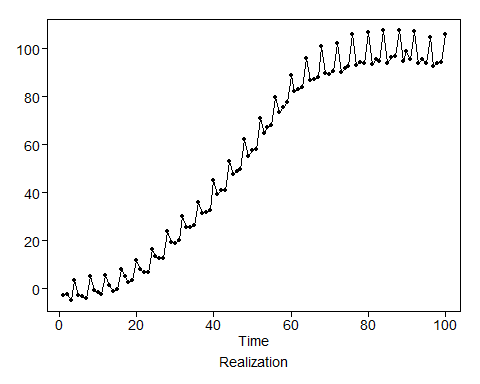
# results = cbind(f$f, f$ll, f$ul, x[(n-n.ahead+1):n])  
# results = data.frame(results)  
# names(results) = c("Forecast", "Lower Limit", "Upper Limit", "Actual")  
#   
# results = results %>%   
# mutate(ll\_satisfied = ifelse(Actual >= `Lower Limit`, TRUE, FALSE),  
# ul\_satisfied = ifelse(Actual <= `Upper Limit`, TRUE, FALSE),  
# within\_limits = ifelse(ll\_satisfied & ul\_satisfied, TRUE, FALSE))  
#   
# results  
  
comparison = cbind(f$f, f$ll, f$ul, x[(n-n.ahead+1):n])  
comparison = data.frame(comparison)  
names(comparison) = c("Forecast", "Lower Limit", "Upper Limit", "Actual")  
comparison = comparison %>%   
 mutate(ll\_satisfied = ifelse(Actual >= `Lower Limit`, TRUE, FALSE),  
 ul\_satisfied = ifelse(Actual <= `Upper Limit`, TRUE, FALSE),  
 within\_limits = ifelse(ll\_satisfied & ul\_satisfied, TRUE, FALSE))  
  
comparison

## Forecast Lower Limit Upper Limit Actual ll\_satisfied ul\_satisfied  
## 1 50.14133 48.22702 52.05564 50.17344 TRUE TRUE  
## 2 53.60330 48.80223 58.40436 53.84857 TRUE TRUE  
## 3 55.97430 49.27596 62.67264 55.05104 TRUE TRUE  
## 4 55.08698 48.00444 62.16952 50.50711 TRUE TRUE  
## 5 51.95171 44.84133 59.06210 43.96097 FALSE TRUE  
## 6 49.52380 42.21759 56.83001 41.09995 FALSE TRUE  
## 7 49.84464 42.53231 57.15697 43.57753 TRUE TRUE  
## 8 52.29221 44.74564 59.83877 49.69448 TRUE TRUE  
## 9 54.40431 46.40381 62.40481 53.89006 TRUE TRUE  
## 10 54.27623 46.15501 62.39745 54.07724 TRUE TRUE  
## within\_limits  
## 1 TRUE  
## 2 TRUE  
## 3 TRUE  
## 4 TRUE  
## 5 FALSE  
## 6 FALSE  
## 7 TRUE  
## 8 TRUE  
## 9 TRUE  
## 10 TRUE

* However, if we look at the 95% confidence interval, we see that the prediction is much more reasonable in that case. Still the further out we go, the actual values tend to be outside the 95% forecast limits.

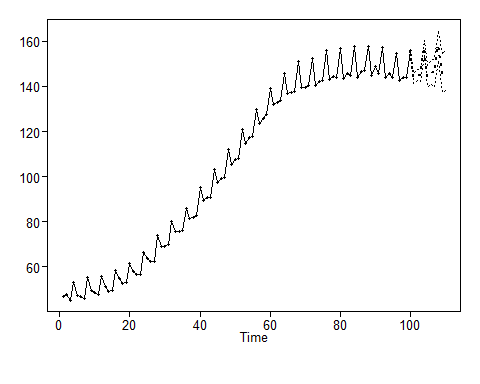
# Problem 6.4

phi = -0.5  
x = 50 + gen.aruma.wge(n = 100, phi = phi, d = 1, s = 4)



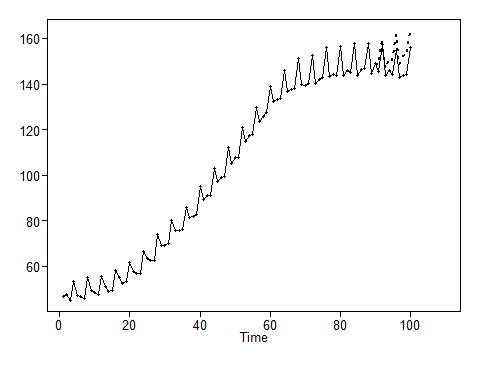
## a

n.ahead = 10  
f = fore.aruma.wge(x, phi = phi, d = 1, s = 4, n.ahead = n.ahead)



## b

n = length(x)  
n.ahead = 10  
  
f = fore.aruma.wge(x, phi = phi, d = 1, s = 4, n.ahead = n.ahead, lastn = TRUE, limits = FALSE)



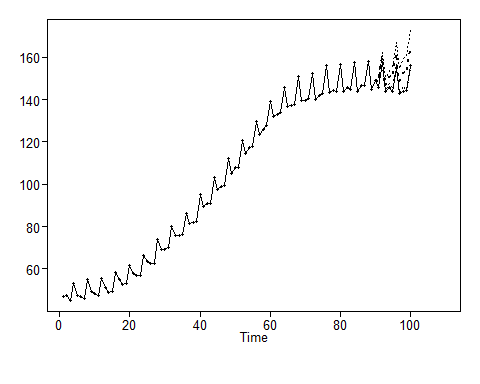
ase = mean((x[(n-n.ahead+1):n] - f$f)^2)  
ase

## [1] 37.54438

* The forecasts are fairly accurate in this case, especially for the 1st few points.
* As we move away from the last known point (90 in this case), the forecast error tends to increase.

## c

n = length(x)  
n.ahead = 10  
  
f = fore.aruma.wge(x, phi = phi, d = 1, s = 4, n.ahead = n.ahead, lastn = TRUE, limits = TRUE)



# results = cbind(f$f, f$ll, f$ul, x[(n-n.ahead+1):n])  
# results = data.frame(results)  
# names(results) = c("Forecast", "Lower Limit", "Upper Limit", "Actual")  
# results = results %>%   
# mutate(ll\_satisfied = ifelse(Actual >= `Lower Limit`, TRUE, FALSE),  
# ul\_satisfied = ifelse(Actual <= `Upper Limit`, TRUE, FALSE),  
# within\_limits = ifelse(ll\_satisfied & ul\_satisfied, TRUE, FALSE))  
#   
# results  
  
  
comparison = cbind(f$f, f$ll, f$ul, x[(n-n.ahead+1):n])  
comparison = data.frame(comparison)  
names(comparison) = c("Forecast", "Lower Limit", "Upper Limit", "Actual")  
comparison = comparison %>%   
 mutate(ll\_satisfied = ifelse(Actual >= `Lower Limit`, TRUE, FALSE),  
 ul\_satisfied = ifelse(Actual <= `Upper Limit`, TRUE, FALSE),  
 within\_limits = ifelse(ll\_satisfied & ul\_satisfied, TRUE, FALSE))  
  
comparison

## Forecast Lower Limit Upper Limit Actual ll\_satisfied ul\_satisfied  
## 1 148.5591 146.4487 150.6695 145.6867 FALSE TRUE  
## 2 159.9085 157.5490 162.2680 157.4362 FALSE TRUE  
## 3 146.6989 143.8577 149.5400 143.8618 TRUE TRUE  
## 4 150.7865 147.6541 153.9189 145.7814 FALSE TRUE  
## 5 150.4451 145.7022 155.1880 144.0157 FALSE TRUE  
## 6 161.8148 156.4810 167.1486 154.6604 FALSE TRUE  
## 7 148.5950 142.4751 154.7149 142.6778 TRUE TRUE  
## 8 152.6877 145.9904 159.3850 143.8953 FALSE TRUE  
## 9 152.3438 144.0033 160.6843 144.1983 TRUE TRUE  
## 10 163.7148 154.5303 172.8992 156.1789 TRUE TRUE  
## within\_limits  
## 1 FALSE  
## 2 FALSE  
## 3 TRUE  
## 4 FALSE  
## 5 FALSE  
## 6 FALSE  
## 7 TRUE  
## 8 FALSE  
## 9 TRUE  
## 10 TRUE

* Most of the values are outside the limit. This may be due to the fact that the realization had an upward trend just before the forecast started. Hence we see that most of the points that lie outside the limits are violating the lower limit since the forecast is higher than the actual.

# Problem 6.7

data("sunspot.classic")  
data %>%glimpse()

## function (..., list = character(), package = NULL, lib.loc = NULL,   
## verbose = getOption("verbose"), envir = .GlobalEnv)

x = sunspot.classic  
  
n = length(x)  
n.ahead = 30

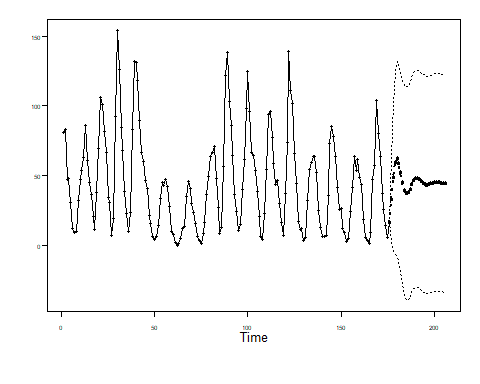
## a

### Model 1

phi = c(1.42, -0.73)  
factor.wge(phi)

##   
## Coefficients of Original polynomial:   
## 1.4200 -0.7300   
##   
## Factor Roots Abs Recip System Freq   
## 1-1.4200B+0.7300B^2 0.9726+-0.6511i 0.8544 0.0939  
##   
##

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

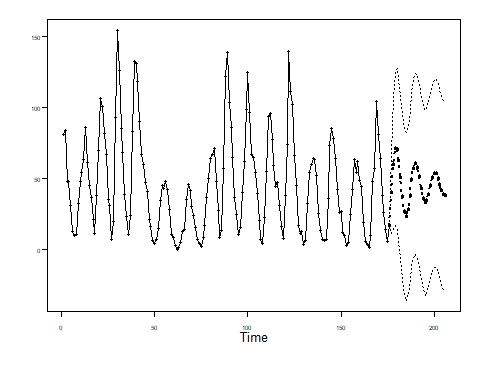


### Model 2

phi = c(1.23, -0.47, -0.14, 0.16, -0.14, 0.07, -0.13, 0.21)  
factor.wge(phi)

##   
## Coefficients of Original polynomial:   
## 1.2300 -0.4700 -0.1400 0.1600 -0.1400 0.0700 -0.1300 0.2100   
##   
## 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  
##   
##

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



### Model 3

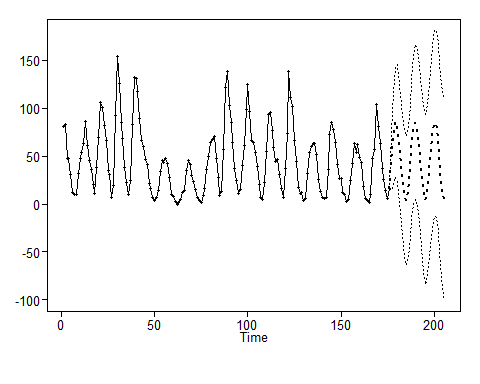
lambda = c(1.646, -1)  
phi = c(-0.37, -0.11, 0.03, 0.29, 0.31, 0.27)  
  
factor.wge(lambda)

##   
## Coefficients of Original polynomial:   
## 1.6460 -1.0000   
##   
## Factor Roots Abs Recip System Freq   
## 1-1.6460B+1.0000B^2 0.8230+-0.5680i 1.0000 0.0961  
##   
##

factor.wge(phi)

##   
## Coefficients of Original polynomial:   
## -0.3700 -0.1100 0.0300 0.2900 0.3100 0.2700   
##   
## Factor Roots Abs Recip System Freq   
## 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  
##   
##

f = fore.aruma.wge(x = x, phi = phi, lambda = lambda, n.ahead = n.ahead)



### Conclusions

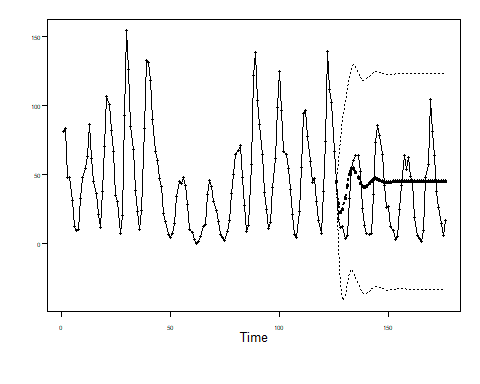
* 1. Model 3 seems to be the best fit for the data.
  2. All 3 models have a system frequency at ~0.09 which is the frequency of the sunspot data. However, in the 1st model, the root is furthest away from the unit circle (0.85). Hence, the sinusoidal behavior associated with it is no sustained. In the 2nd model, the root associated with this frequency is closer to the unit circle (~0.95), hence its effect is more sustained, though it eventually begins to damp out. In the 3rd model, the root associated with this frequency is right on the unit circle, hence this behavior is sustained over many time points (and this matched with the realization well)
  3. Based on this, we would prefer the 3rd model. The 1st model is the poorest.

## b

n = length(x)  
n.ahead = 50  
  
results = tribble(~model, ~ASE)

### Model 1

phi = c(1.42, -0.73)  
f = fore.arma.wge(x = x, phi = phi, n.ahead = n.ahead, lastn = TRUE)



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

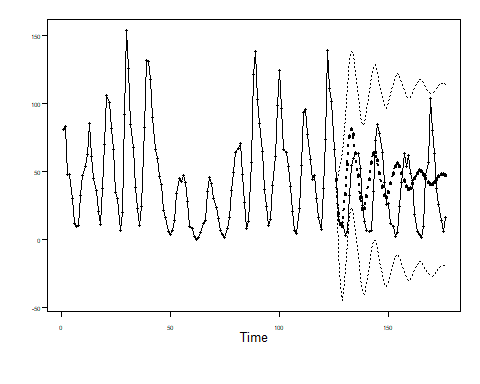
## [1] 709.9071

results = results %>% add\_row(model = "Model 1", ASE = ase)  
  
  
comparison = cbind(f$f, f$ll, f$ul, x[(n-n.ahead+1):n])  
comparison = data.frame(comparison)  
names(comparison) = c("Forecast", "Lower Limit", "Upper Limit", "Actual")  
comparison = comparison %>%   
 mutate(ll\_satisfied = ifelse(Actual >= `Lower Limit`, TRUE, FALSE),  
 ul\_satisfied = ifelse(Actual <= `Upper Limit`, TRUE, FALSE),  
 within\_limits = ifelse(ll\_satisfied & ul\_satisfied, TRUE, FALSE))  
  
comparison

## Forecast Lower Limit Upper Limit Actual ll\_satisfied ul\_satisfied  
## 1 29.03107 -1.398116 59.46025 17.0 TRUE TRUE  
## 2 22.47619 -30.372573 75.32494 11.3 TRUE TRUE  
## 3 24.60657 -41.160077 90.37322 12.4 TRUE TRUE  
## 4 32.41678 -37.606476 102.44004 3.4 TRUE TRUE  
## 5 41.95210 -28.291868 112.19608 6.0 TRUE TRUE  
## 6 49.79080 -21.112852 120.69446 32.3 TRUE TRUE  
## 7 53.96097 -19.134099 127.05605 54.3 TRUE TRUE  
## 8 54.16036 -21.161823 129.48255 59.7 TRUE TRUE  
## 9 51.39927 -25.010992 127.80954 63.7 TRUE TRUE  
## 10 47.33297 -29.238991 123.90493 63.5 TRUE TRUE  
## 11 43.57442 -33.032600 120.18144 52.2 TRUE TRUE  
## 12 41.20567 -35.713369 118.12471 25.4 TRUE TRUE  
## 13 40.58580 -36.762393 117.93399 13.1 TRUE TRUE  
## 14 41.43476 -36.186184 119.05570 6.8 TRUE TRUE  
## 15 43.09280 -34.597940 120.78353 6.3 TRUE TRUE  
## 16 44.82746 -32.863305 122.51823 7.1 TRUE TRUE  
## 17 46.08032 -31.650789 123.81144 35.6 TRUE TRUE  
## 18 46.59308 -31.216973 124.40314 73.0 TRUE TRUE  
## 19 46.40661 -31.467093 124.28031 85.1 TRUE TRUE  
## 20 45.76750 -32.129664 123.66466 78.0 TRUE TRUE  
## 21 44.99610 -32.902185 122.89438 64.0 TRUE TRUE  
## 22 44.36725 -33.535176 122.26967 41.8 TRUE TRUE  
## 23 44.03741 -33.878422 121.95325 26.2 TRUE TRUE  
## 24 44.02810 -33.901522 121.95772 26.7 TRUE TRUE  
## 25 44.25566 -33.680721 122.19204 12.1 TRUE TRUE  
## 26 44.58559 -33.351752 122.52294 9.5 TRUE TRUE  
## 27 44.88798 -33.049622 122.82558 2.7 TRUE TRUE  
## 28 45.07651 -32.863139 123.01617 5.0 TRUE TRUE  
## 29 45.12349 -32.818934 123.06592 24.4 TRUE TRUE  
## 30 45.05257 -32.891596 122.99675 42.0 TRUE TRUE  
## 31 44.91757 -33.027031 122.86218 63.5 TRUE TRUE  
## 32 44.77764 -33.166962 122.72225 53.8 TRUE TRUE  
## 33 44.67749 -33.267384 122.62237 62.0 TRUE TRUE  
## 34 44.63743 -33.307966 122.58282 48.5 TRUE TRUE  
## 35 44.65365 -33.292159 122.59945 43.9 TRUE TRUE  
## 36 44.70592 -33.240030 122.65188 18.6 TRUE TRUE  
## 37 44.76832 -33.177642 122.71428 5.7 TRUE TRUE  
## 38 44.81876 -33.127233 122.76474 3.6 TRUE TRUE  
## 39 44.84483 -33.101249 122.79091 1.4 TRUE TRUE  
## 40 44.84503 -33.101133 122.79120 9.6 TRUE TRUE  
## 41 44.82629 -33.119919 122.77250 47.4 TRUE TRUE  
## 42 44.79953 -33.146689 122.74574 57.1 TRUE TRUE  
## 43 44.77520 -33.171014 122.72142 103.9 TRUE TRUE  
## 44 44.76020 -33.186028 122.70643 80.6 TRUE TRUE  
## 45 44.75666 -33.189592 122.70291 63.6 TRUE TRUE  
## 46 44.76257 -33.183687 122.70883 37.6 TRUE TRUE  
## 47 44.77356 -33.172700 122.71983 26.1 TRUE TRUE  
## 48 44.78485 -33.161415 122.73111 14.2 TRUE TRUE  
## 49 44.79285 -33.153413 122.73912 5.8 TRUE TRUE  
## 50 44.79598 -33.150290 122.74225 16.7 TRUE TRUE  
## within\_limits  
## 1 TRUE  
## 2 TRUE  
## 3 TRUE  
## 4 TRUE  
## 5 TRUE  
## 6 TRUE  
## 7 TRUE  
## 8 TRUE  
## 9 TRUE  
## 10 TRUE  
## 11 TRUE  
## 12 TRUE  
## 13 TRUE  
## 14 TRUE  
## 15 TRUE  
## 16 TRUE  
## 17 TRUE  
## 18 TRUE  
## 19 TRUE  
## 20 TRUE  
## 21 TRUE  
## 22 TRUE  
## 23 TRUE  
## 24 TRUE  
## 25 TRUE  
## 26 TRUE  
## 27 TRUE  
## 28 TRUE  
## 29 TRUE  
## 30 TRUE  
## 31 TRUE  
## 32 TRUE  
## 33 TRUE  
## 34 TRUE  
## 35 TRUE  
## 36 TRUE  
## 37 TRUE  
## 38 TRUE  
## 39 TRUE  
## 40 TRUE  
## 41 TRUE  
## 42 TRUE  
## 43 TRUE  
## 44 TRUE  
## 45 TRUE  
## 46 TRUE  
## 47 TRUE  
## 48 TRUE  
## 49 TRUE  
## 50 TRUE

### Model 2

phi = c(1.23, -0.47, -0.14, 0.16, -0.14, 0.07, -0.13, 0.21)  
f = fore.arma.wge(x = x, phi = phi, n.ahead = n.ahead, lastn = TRUE)



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

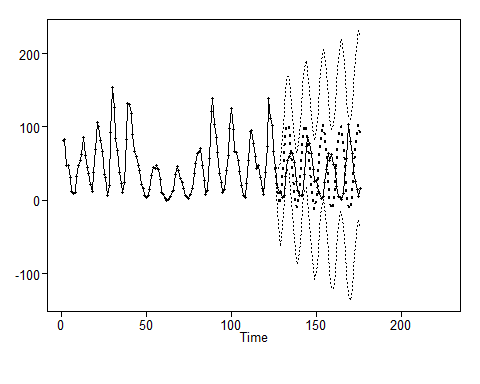
## [1] 881.8741

results = results %>% add\_row(model = "Model 2", ASE = ase)  
  
  
comparison = cbind(f$f, f$ll, f$ul, x[(n-n.ahead+1):n])  
comparison = data.frame(comparison)  
names(comparison) = c("Forecast", "Lower Limit", "Upper Limit", "Actual")  
comparison = comparison %>%   
 mutate(ll\_satisfied = ifelse(Actual >= `Lower Limit`, TRUE, FALSE),  
 ul\_satisfied = ifelse(Actual <= `Upper Limit`, TRUE, FALSE),  
 within\_limits = ifelse(ll\_satisfied & ul\_satisfied, TRUE, FALSE))  
  
comparison

## Forecast Lower Limit Upper Limit Actual ll\_satisfied ul\_satisfied  
## 1 19.204659 -9.4422426 47.85156 17.0 TRUE TRUE  
## 2 11.443390 -33.9680489 56.85483 11.3 TRUE TRUE  
## 3 9.817839 -44.5399151 64.17559 12.4 TRUE TRUE  
## 4 33.143555 -23.5699977 89.85711 3.4 TRUE TRUE  
## 5 55.547519 -1.4326118 112.52765 6.0 TRUE TRUE  
## 6 75.777429 18.6967418 132.85812 32.3 TRUE TRUE  
## 7 81.267571 23.7389134 138.79623 54.3 TRUE TRUE  
## 8 77.591630 19.0497878 136.13347 59.7 TRUE TRUE  
## 9 63.517857 4.5159145 122.51980 63.7 TRUE TRUE  
## 10 47.480675 -11.5225556 106.48391 63.5 TRUE TRUE  
## 11 31.125050 -28.6125812 90.86268 52.2 TRUE TRUE  
## 12 22.560643 -38.8057679 83.92705 25.4 TRUE TRUE  
## 13 22.680854 -40.3223510 85.68406 13.1 TRUE TRUE  
## 14 31.825399 -31.9971911 95.64799 6.8 TRUE TRUE  
## 15 44.489649 -19.5055037 108.48480 6.3 TRUE TRUE  
## 16 56.606435 -7.4033015 120.61617 7.1 TRUE TRUE  
## 17 63.480345 -0.8043925 127.76508 35.6 TRUE TRUE  
## 18 64.072583 -0.6880645 128.83323 73.0 TRUE TRUE  
## 19 58.307098 -6.7494604 123.36366 85.1 TRUE TRUE  
## 20 48.966508 -16.1264510 114.05947 78.0 TRUE TRUE  
## 21 39.230873 -25.9129430 104.37469 64.0 TRUE TRUE  
## 22 32.707876 -32.7179903 98.13374 41.8 TRUE TRUE  
## 23 31.128113 -34.7163716 96.97260 26.2 TRUE TRUE  
## 24 34.618849 -31.5322345 100.76993 26.7 TRUE TRUE  
## 25 41.281090 -24.9649187 107.52710 12.1 TRUE TRUE  
## 26 48.595518 -17.6505086 114.84154 9.5 TRUE TRUE  
## 27 53.954796 -12.3646145 120.27421 2.7 TRUE TRUE  
## 28 55.803397 -10.6821010 122.28889 5.0 TRUE TRUE  
## 29 53.804473 -12.8236261 120.43257 24.4 TRUE TRUE  
## 30 49.044142 -17.6294675 115.71775 42.0 TRUE TRUE  
## 31 43.383901 -23.2899045 110.05771 63.5 TRUE TRUE  
## 32 38.863459 -27.8580637 105.58498 53.8 TRUE TRUE  
## 33 36.874787 -29.9585929 103.70817 62.0 TRUE TRUE  
## 34 37.832685 -29.1102347 104.77561 48.5 TRUE TRUE  
## 35 41.084451 -25.9073961 108.07630 43.9 TRUE TRUE  
## 36 45.296330 -21.6981780 112.29084 18.6 TRUE TRUE  
## 37 48.932033 -18.0755333 115.93960 5.7 TRUE TRUE  
## 38 50.820526 -16.2357854 117.87684 3.6 TRUE TRUE  
## 39 50.490905 -16.6220207 117.60383 1.4 TRUE TRUE  
## 40 48.283822 -18.8580344 115.42568 9.6 TRUE TRUE  
## 41 45.137168 -22.0071323 112.28147 47.4 TRUE TRUE  
## 42 42.216681 -24.9336052 109.36697 57.1 TRUE TRUE  
## 43 40.485098 -26.6929818 107.66318 103.9 TRUE TRUE  
## 44 40.405472 -26.8094234 107.62037 80.6 TRUE TRUE  
## 45 41.830693 -25.4069336 109.06832 63.6 TRUE TRUE  
## 46 44.121751 -23.1199632 111.36347 37.6 TRUE TRUE  
## 47 46.410296 -20.8326081 113.65320 26.1 TRUE TRUE  
## 48 47.919704 -19.3353175 115.17473 14.2 TRUE TRUE  
## 49 48.216749 -19.0577634 115.49126 5.8 TRUE TRUE  
## 50 47.325565 -19.9628761 114.61401 16.7 TRUE TRUE  
## within\_limits  
## 1 TRUE  
## 2 TRUE  
## 3 TRUE  
## 4 TRUE  
## 5 TRUE  
## 6 TRUE  
## 7 TRUE  
## 8 TRUE  
## 9 TRUE  
## 10 TRUE  
## 11 TRUE  
## 12 TRUE  
## 13 TRUE  
## 14 TRUE  
## 15 TRUE  
## 16 TRUE  
## 17 TRUE  
## 18 TRUE  
## 19 TRUE  
## 20 TRUE  
## 21 TRUE  
## 22 TRUE  
## 23 TRUE  
## 24 TRUE  
## 25 TRUE  
## 26 TRUE  
## 27 TRUE  
## 28 TRUE  
## 29 TRUE  
## 30 TRUE  
## 31 TRUE  
## 32 TRUE  
## 33 TRUE  
## 34 TRUE  
## 35 TRUE  
## 36 TRUE  
## 37 TRUE  
## 38 TRUE  
## 39 TRUE  
## 40 TRUE  
## 41 TRUE  
## 42 TRUE  
## 43 TRUE  
## 44 TRUE  
## 45 TRUE  
## 46 TRUE  
## 47 TRUE  
## 48 TRUE  
## 49 TRUE  
## 50 TRUE

### Model 3

lambda = c(1.646, -1)  
phi = c(-0.37, -0.11, 0.03, 0.29, 0.31, 0.27)  
f = fore.aruma.wge(x = x, phi = phi, lambda = lambda, n.ahead = n.ahead, lastn = TRUE)



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

## [1] 3039.52

results = results %>% add\_row(model = "Model 3", ASE = ase)  
  
  
comparison = cbind(f$f, f$ll, f$ul, x[(n-n.ahead+1):n])  
comparison = data.frame(comparison)  
names(comparison) = c("Forecast", "Lower Limit", "Upper Limit", "Actual")  
comparison = comparison %>%   
 mutate(ll\_satisfied = ifelse(Actual >= `Lower Limit`, TRUE, FALSE),  
 ul\_satisfied = ifelse(Actual <= `Upper Limit`, TRUE, FALSE),  
 within\_limits = ifelse(ll\_satisfied & ul\_satisfied, TRUE, FALSE))  
  
comparison

## Forecast Lower Limit Upper Limit Actual ll\_satisfied ul\_satisfied  
## 1 12.951342 -16.3899516 42.29263 17.0 TRUE TRUE  
## 2 -2.077089 -49.6441665 45.48999 11.3 TRUE TRUE  
## 3 -5.528269 -63.4633694 52.40683 12.4 TRUE TRUE  
## 4 25.505978 -35.3974554 86.40941 3.4 TRUE TRUE  
## 5 60.052705 -1.0790254 121.18443 6.0 TRUE TRUE  
## 6 92.629726 31.0904957 154.16896 32.3 TRUE TRUE  
## 7 105.324267 42.2096321 168.43890 54.3 TRUE TRUE  
## 8 101.954026 35.9400694 167.96798 59.7 TRUE TRUE  
## 9 79.642669 12.1810687 147.10427 63.7 TRUE TRUE  
## 10 48.304568 -19.1626110 115.77175 63.5 TRUE TRUE  
## 11 13.829305 -55.2293812 82.88799 52.2 TRUE TRUE  
## 12 -7.743033 -81.1400434 65.65398 25.4 TRUE TRUE  
## 13 -9.888443 -88.0922606 68.31538 13.1 TRUE TRUE  
## 14 9.553441 -71.2815264 90.38841 6.8 TRUE TRUE  
## 15 41.436403 -39.8758708 122.74868 6.3 TRUE TRUE  
## 16 75.167483 -6.3298722 156.66484 7.1 TRUE TRUE  
## 17 98.301634 14.9560815 181.64719 35.6 TRUE TRUE  
## 18 103.788756 17.3452014 190.23231 73.0 TRUE TRUE  
## 19 88.818483 0.2406373 177.39633 85.1 TRUE TRUE  
## 20 58.748576 -30.2100229 147.70718 78.0 TRUE TRUE  
## 21 23.789435 -65.3910726 112.96994 64.0 TRUE TRUE  
## 22 -3.193764 -94.3101687 87.92264 41.8 TRUE TRUE  
## 23 -12.815241 -107.2637683 81.63329 26.2 TRUE TRUE  
## 24 -1.618883 -98.7509660 95.51320 26.7 TRUE TRUE  
## 25 26.104005 -71.8763003 124.08431 12.1 TRUE TRUE  
## 26 60.673390 -37.3158444 158.66263 9.5 TRUE TRUE  
## 27 89.826035 -9.2827996 188.93487 2.7 TRUE TRUE  
## 28 103.322032 1.5960818 205.04798 5.0 TRUE TRUE  
## 29 96.236573 -7.9722210 200.44537 24.4 TRUE TRUE  
## 30 71.073534 -34.0870336 176.23410 42.0 TRUE TRUE  
## 31 36.706399 -68.4553259 141.86812 63.5 TRUE TRUE  
## 32 5.361286 -100.6518877 111.37446 53.8 TRUE TRUE  
## 33 -11.909044 -120.3270275 96.50894 62.0 TRUE TRUE  
## 34 -9.007751 -120.0714472 102.05595 48.5 TRUE TRUE  
## 35 12.999889 -99.3665967 125.36637 43.9 TRUE TRUE  
## 36 46.343515 -66.0961131 158.78314 18.6 TRUE TRUE  
## 37 79.212470 -33.6989304 192.12387 5.7 TRUE TRUE  
## 38 99.968251 -14.8701204 214.80662 3.6 TRUE TRUE  
## 39 101.237734 -16.1016332 218.57710 1.4 TRUE TRUE  
## 40 82.570595 -36.2433798 201.38457 9.6 TRUE TRUE  
## 41 50.572023 -68.4236056 169.56765 47.4 TRUE TRUE  
## 42 16.572430 -102.6842543 135.82911 57.1 TRUE TRUE  
## 43 -7.403316 -128.2599235 113.45329 103.9 TRUE TRUE  
## 44 -12.873125 -136.1710250 110.42478 80.6 TRUE TRUE  
## 45 2.094525 -122.9153079 127.10436 63.6 TRUE TRUE  
## 46 32.202787 -93.1712449 157.57682 37.6 TRUE TRUE  
## 47 66.789988 -58.6786787 192.25865 26.1 TRUE TRUE  
## 48 93.609003 -33.0802269 220.29823 14.2 TRUE TRUE  
## 49 103.161520 -25.7776377 232.10068 5.8 TRUE TRUE  
## 50 92.065397 -38.7225709 222.85336 16.7 TRUE TRUE  
## within\_limits  
## 1 TRUE  
## 2 TRUE  
## 3 TRUE  
## 4 TRUE  
## 5 TRUE  
## 6 TRUE  
## 7 TRUE  
## 8 TRUE  
## 9 TRUE  
## 10 TRUE  
## 11 TRUE  
## 12 TRUE  
## 13 TRUE  
## 14 TRUE  
## 15 TRUE  
## 16 TRUE  
## 17 TRUE  
## 18 TRUE  
## 19 TRUE  
## 20 TRUE  
## 21 TRUE  
## 22 TRUE  
## 23 TRUE  
## 24 TRUE  
## 25 TRUE  
## 26 TRUE  
## 27 TRUE  
## 28 TRUE  
## 29 TRUE  
## 30 TRUE  
## 31 TRUE  
## 32 TRUE  
## 33 TRUE  
## 34 TRUE  
## 35 TRUE  
## 36 TRUE  
## 37 TRUE  
## 38 TRUE  
## 39 TRUE  
## 40 TRUE  
## 41 TRUE  
## 42 TRUE  
## 43 TRUE  
## 44 TRUE  
## 45 TRUE  
## 46 TRUE  
## 47 TRUE  
## 48 TRUE  
## 49 TRUE  
## 50 TRUE

### Conclusions

results

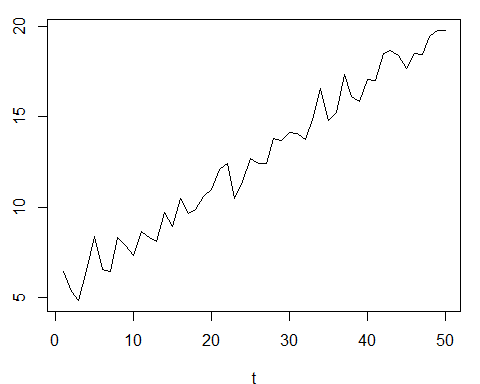
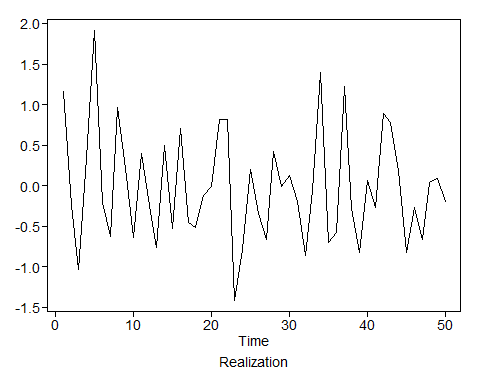
## # A tibble: 3 x 2  
## model ASE  
## <chr> <dbl>  
## 1 Model 1 710.  
## 2 Model 2 882.  
## 3 Model 3 3040.

* From the ASE calculations, we see that the 2nd and 3rd models dont perform that well and their prediction is very bad compred to model 1.
* This is due to the fact that although the cyclical behavior, the phase is not aligned (peak’s dont align) and this leads to the larger error.
* On this basis, we would choose Model 1 since it has the best ASE. However, the prediction may only be good for the next few time points (and then settles at the mean).

# Problem 6.8

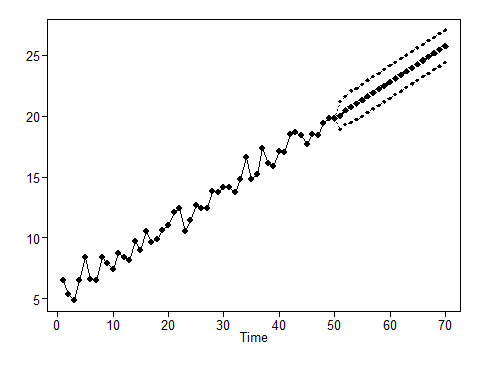
## a

x = gen.sigplusnoise.wge(n = 50, b0 = 5, b1 = 0.3, phi = c(-0.2, -0.6), vara = 0.5)

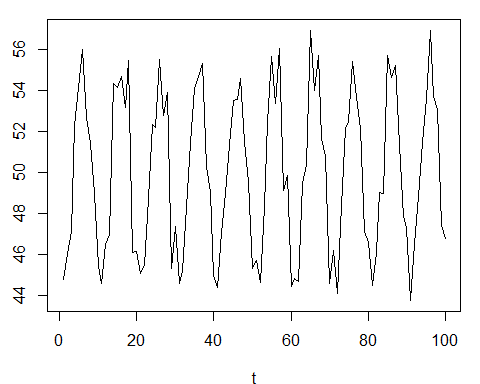
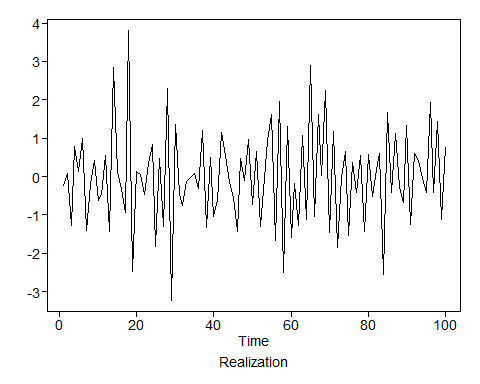


n.ahead = 20  
f = fore.sigplusnoise.wge(x = x, linear = TRUE, n.ahead = 20)

##   
## Coefficients of Original polynomial:   
## -0.1370 -0.4963   
##   
## Factor Roots Abs Recip System Freq   
## 1+0.1370B+0.4963B^2 -0.1380+-1.4128i 0.7045 0.2655  
##   
##



x = gen.sigplusnoise.wge(n = 100,  
 b0 = 50,  
 coef = c(5, 0), freq = c(0.1, 0), psi = c(2.5, 0),  
 phi = c(-0.7), vara = 1)



n.ahead = 20  
f = fore.sigplusnoise.wge(x = x, linear = FALSE, freq = c(0.1, 0), n.ahead = 20)

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x1[t] <- cos(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

## Warning in x2[t] <- sin(2 \* pi \* freq \* t): number of items to replace is  
## not a multiple of replacement length

##   
## Coefficients of Original polynomial:   
## -0.6212   
##   
## Factor Roots Abs Recip System Freq   
## 1+0.6212B -1.6098 0.6212 0.5000  
##   
##

