Homework 6

Nikhil Gupta

2020-02-23 17:46:02

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")

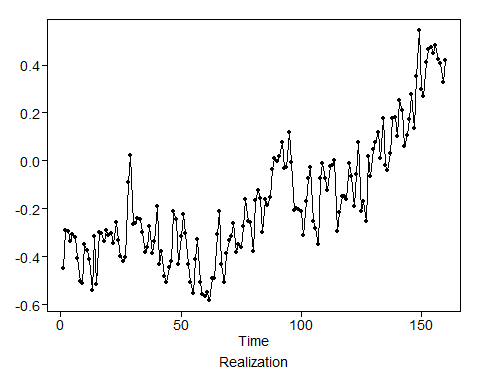
# Problem 5.1

data("hadley")  
  
hadley %>%   
 glimpse()

## num [1:160] -0.447 -0.292 -0.294 -0.337 -0.307 -0.321 -0.406 -0.503 -0.513 -0.349 ...

## a

px = plotts.wge(hadley)



## b

data = data.frame(temp = hadley)  
data = data %>%  
 mutate(time = as.numeric(rownames(.)))  
str(data)

## 'data.frame': 160 obs. of 2 variables:  
## $ temp: num -0.447 -0.292 -0.294 -0.337 -0.307 -0.321 -0.406 -0.503 -0.513 -0.349 ...  
## $ time: num 1 2 3 4 5 6 7 8 9 10 ...

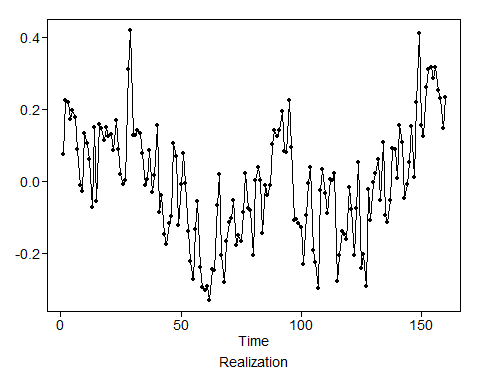
model\_lm = glm(formula = temp ~ time, data = data)  
model\_lm

##   
## Call: glm(formula = temp ~ time, data = data)  
##   
## Coefficients:  
## (Intercept) time   
## -0.525737 0.004438   
##   
## Degrees of Freedom: 159 Total (i.e. Null); 158 Residual  
## Null Deviance: 10.67   
## Residual Deviance: 3.947 AIC: -132.3

a\_hat = -0.525737 b\_hat = 0.0044378

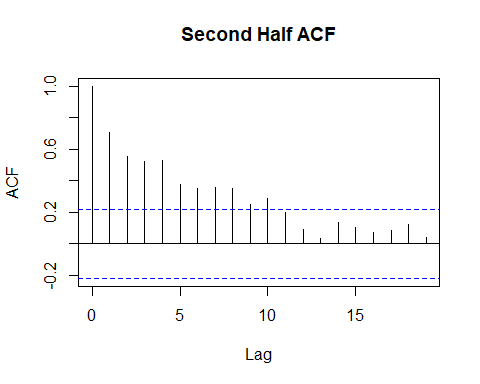
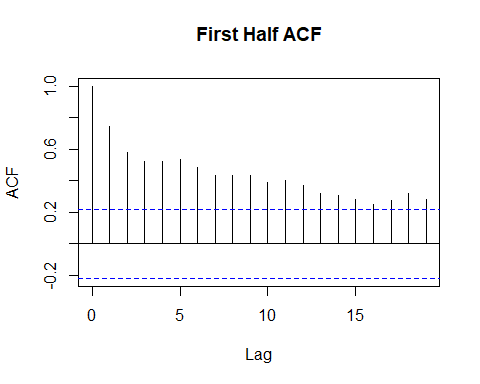
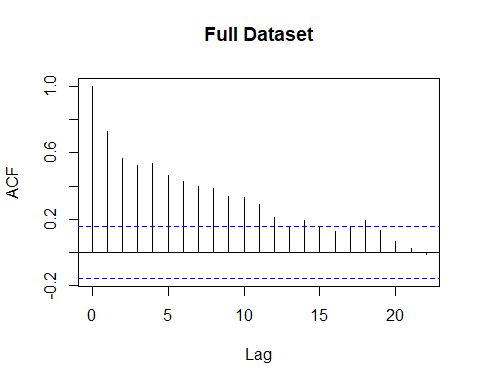
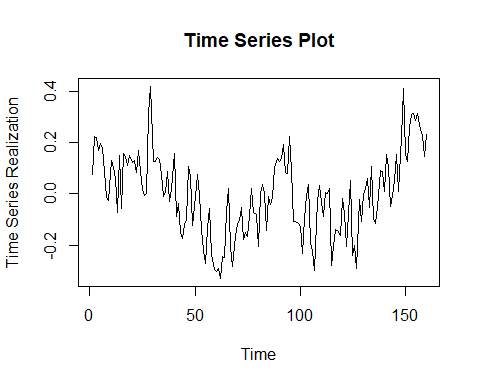
## c

px = plotts.wge(model\_lm$residuals)



## d

check\_stationarity(model\_lm$residuals)



### Condition 1:

Mean does not depend on time: It is hard to pin point from a single realization. It looks like the mean might be varying with time, but it could just be wandering behavior similar to what is seen in a stationary AR process with real positive root.

### Condition 2:

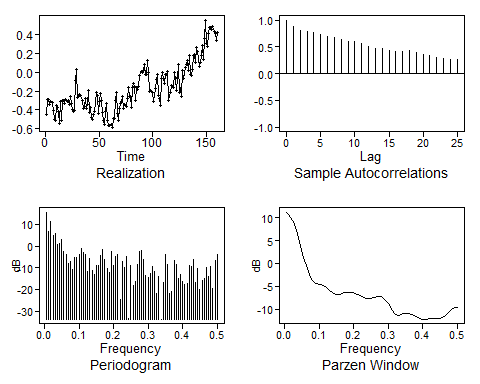
Variance does not depend on time: Again it is hard to say from just a single realization whether the variance is indepent of time. But given just this dataset, there is no reason to doubt it.

### Condition 3

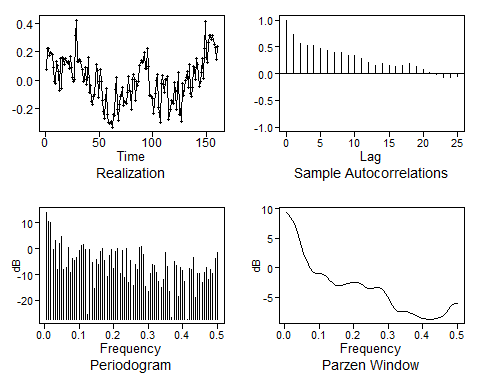
The ACF for the 1st half looks similar to that of the second half upto a lag of about 12. There is a slight mismatch after that, but not too appreciable to warrant suspicion of non-stationarity.

## e

px = plotts.sample.wge(data$temp)



px = plotts.sample.wge(model\_lm$residuals)



### Original Data

The original data ACF plots show extended autocorrelations pointing to an ARIMA type of model with root at 1. This manifests as wandering behavior in the realizations which can also be seen in the plot.

### Residuals

The residual shows an exponentially decaying ACF, quite similar to what we would expect from a stationary AR process with a real and positive root. This is also reinforced by he Spectral Density plot which shows a peak at 0 (indicative of wandering behavior)

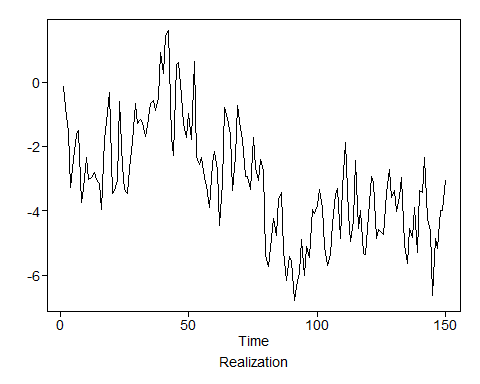
# Problem 5.2

## a

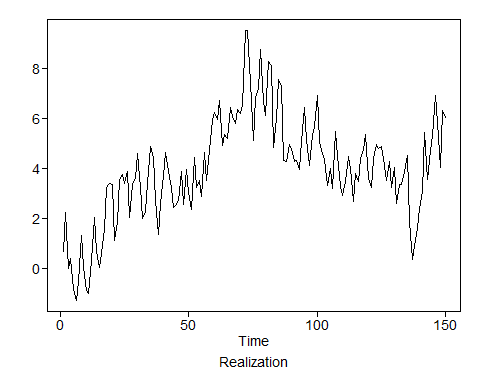
# (1 - 0.66B + 0.02B^2 -0.1B^3 -0.24B^4) \* Xt = at  
  
phi = c(0.66, -0.02, 0.1, 0.24)  
factor.wge(phi = phi)

##   
## Coefficients of Original polynomial:   
## 0.6600 -0.0200 0.1000 0.2400   
##   
## Factor Roots Abs Recip System Freq   
## 1-0.9896B 1.0105 0.9896 0.0000  
## 1-0.2037B+0.4548B^2 0.2240+-1.4658i 0.6744 0.2259  
## 1+0.5333B -1.8752 0.5333 0.5000  
##   
##

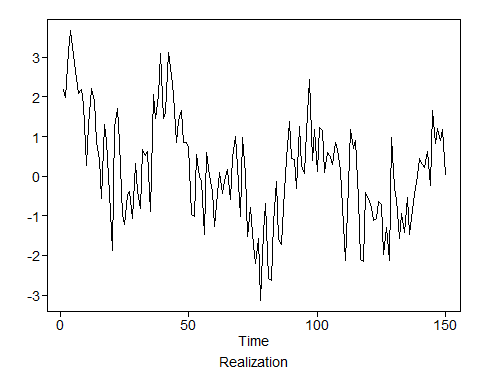
x1 = gen.arma.wge(n = 150, phi = phi)



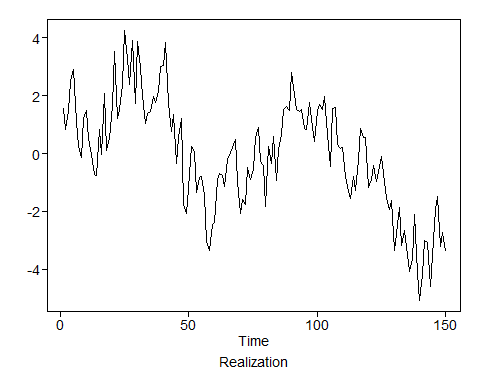
x2 = gen.arma.wge(n = 150, phi = phi)



x3 = gen.arma.wge(n = 150, phi = phi)



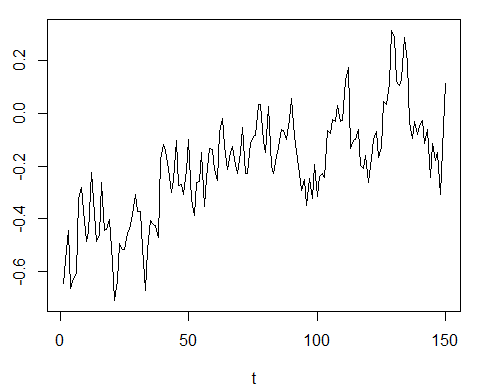
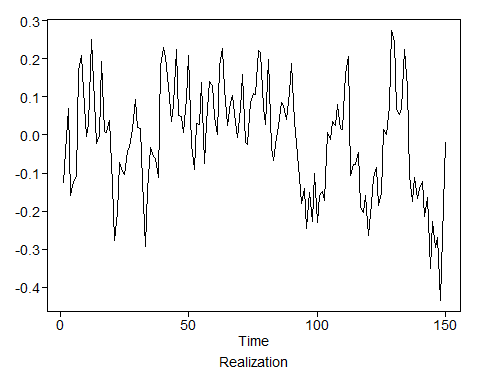
x4 = gen.arma.wge(n = 150, phi = phi)



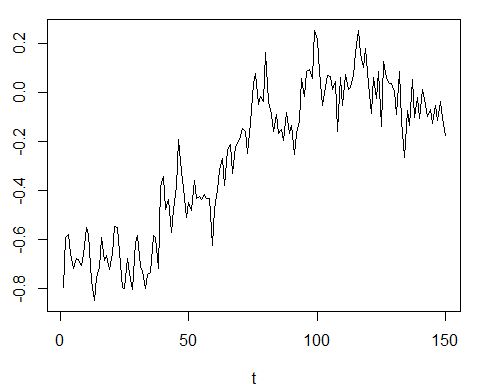
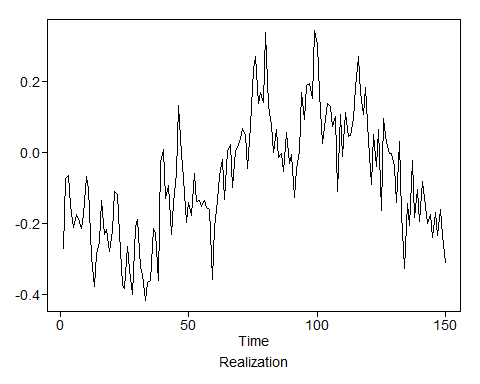
Yes, These do tend to show a linear trend, even though the original model was stationary.

## b

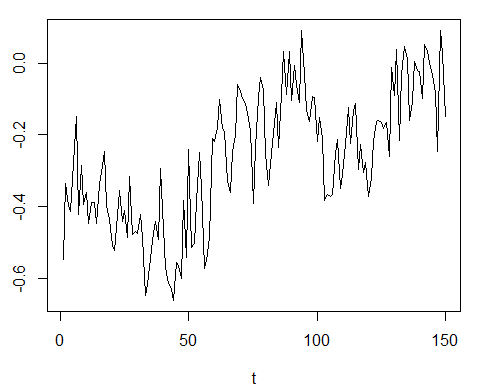
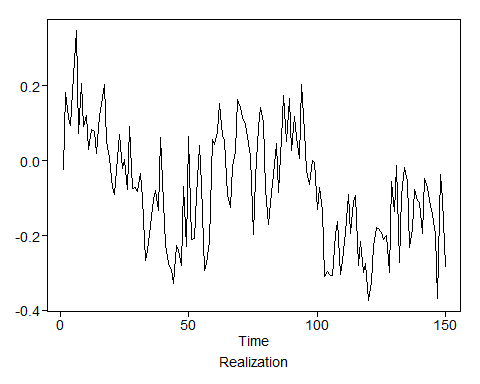
phi = c(0.614, -0.044, 0.077, 0.206)  
x1 = gen.sigplusnoise.wge(n = 150, b0 = -0.526, b1 = 0.0044, phi = phi, vara = 0.01)



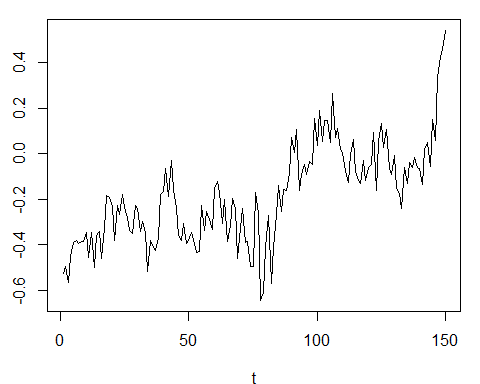
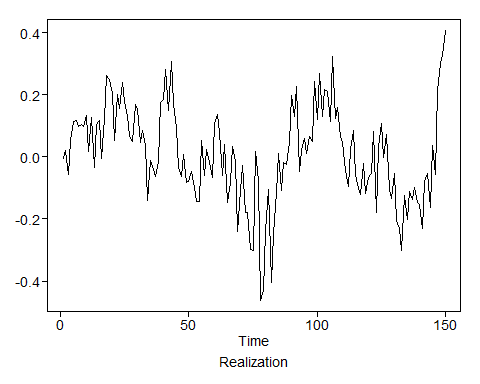
x2 = gen.sigplusnoise.wge(n = 150, b0 = -0.526, b1 = 0.0044, phi = phi, vara = 0.01)



x3 = gen.sigplusnoise.wge(n = 150, b0 = -0.526, b1 = 0.0044, phi = phi, vara = 0.01)



x4 = gen.sigplusnoise.wge(n = 150, b0 = -0.526, b1 = 0.0044, phi = phi, vara = 0.01)



Yes, some of these realization do have an upward trend and tend to look similar to the global temp data.

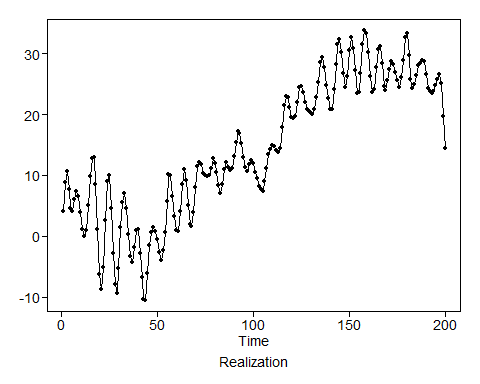
# 5.3

## a

phi = c(1.2, -0.8)  
factor.wge(phi = phi)

##   
## Coefficients of Original polynomial:   
## 1.2000 -0.8000   
##   
## Factor Roots Abs Recip System Freq   
## 1-1.2000B+0.8000B^2 0.7500+-0.8292i 0.8944 0.1330  
##   
##

x = gen.arima.wge(n = 200, phi = phi, d = 1)

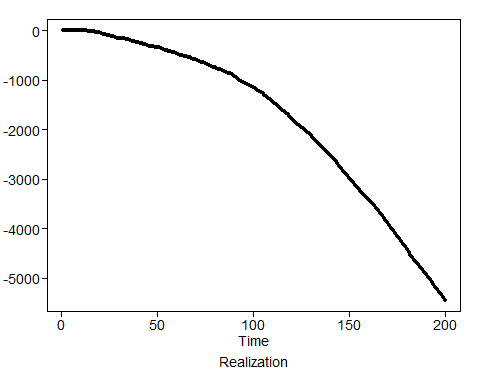


This is a non-stationary model and does not have a seasonal component.

The wandering behavior due to the ARIMA factor can be seen in this dataset. The cyclical behavior due to the AR(2) complex root is also superimposed on top of the wandering behavior due to the ARIMA term.

## b

phi = c(1.2, -0.8)  
x = gen.arima.wge(n = 200, phi = phi, d = 2)

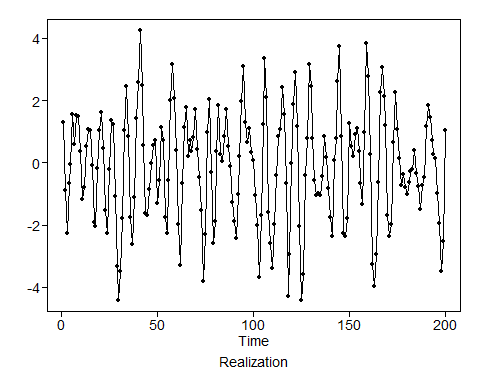


The wandering behavior due to the ARIMA factor (d = 2) is clearly visible in this data and this completely drowns out the cyclical behavior due to the AR(2) complex root.

This is a non-stationary model and does not have a seasonal component.

## c

# There might be a typo in the book. It should be (1-B^4) not (1-B)^4  
phi = c(1.2, -0.8)  
x = gen.arima.wge(n = 200, phi = phi, d = 0, s = 4)



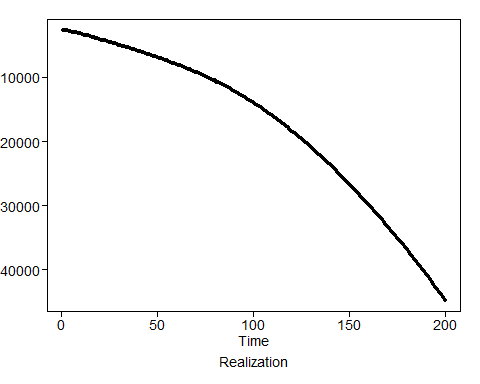
This has even more wandering compared to ‘b’. This is a non-stationary model and does not have a seasonal component.

## d

# There might be a typo in the book. It should be (1-B^4) not (1-B)^4  
phi = c(1.2, -0.8)  
lambda = c(-1, -1)  
  
factor.wge(phi = lambda)

##   
## Coefficients of Original polynomial:   
## -1.0000 -1.0000   
##   
## Factor Roots Abs Recip System Freq   
## 1+1.0000B+1.0000B^2 -0.5000+-0.8660i 1.0000 0.3333  
##   
##

x = gen.aruma.wge(n = 200, phi = phi, d = 2, s = 4, lambda = lambda)



The ARIMA factor with d = 2 is drowning out all the seasonality in the data.

# 5.5

## a

phi = c(3, -4.5, 5, -4, 2, -0.5)  
theta = c(1.7, -0.8)  
  
factor.wge(phi = phi)

##   
## Coefficients of Original polynomial:   
## 3.0000 -4.5000 5.0000 -4.0000 2.0000 -0.5000   
##   
## Factor Roots Abs Recip System Freq   
## 1-1.0000B 1.0000 1.0000 0.0000  
## 1+0.0000B+1.0000B^2 0.0000+-1.0000i 1.0000 0.2500  
## 1-1.0000B 1.0000 1.0000 0.0000  
## 1-1.0000B+0.5000B^2 1.0000+-1.0000i 0.7071 0.1250  
##   
##

factor.wge(phi = theta)

##   
## Coefficients of Original polynomial:   
## 1.7000 -0.8000   
##   
## Factor Roots Abs Recip System Freq   
## 1-1.7000B+0.8000B^2 1.0625+-0.3480i 0.8944 0.0504  
##   
##

* p = 2
* d = 4
* q = 2
* ARUMA(2,4,2) # Note this is ARUMA not ARIMA and there is no seasonal component

## b

phi = c(0.5, -0.3, 0.95, -0.3, 0.35, -0.2)  
  
factor.wge(phi = phi)

##   
## Coefficients of Original polynomial:   
## 0.5000 -0.3000 0.9500 -0.3000 0.3500 -0.2000   
##   
## Factor Roots Abs Recip System Freq   
## 1-1.0000B 1.0000 1.0000 0.0000  
## 1+1.0000B+0.8000B^2 -0.6250+-0.9270i 0.8944 0.3444  
## 1+0.0000B+0.5000B^2 0.0000+-1.4142i 0.7071 0.2500  
## 1-0.5000B 2.0000 0.5000 0.0000  
##   
##

* p = 5
* d = 1
* q = 0
* ARUMA(5,1,0)

## c

phi = c(1.5, -1.3, -0.35, 1, -1.35, 0.7, -0.4)  
theta = c(0, -0.9)  
  
factor.wge(phi = phi)

##   
## Coefficients of Original polynomial:   
## 1.5000 -1.3000 -0.3500 1.0000 -1.3500 0.7000 -0.4000   
##   
## Factor Roots Abs Recip System Freq   
## 1+1.0000B -1.0000 1.0000 0.5000  
## 1-1.5000B+1.0000B^2 0.7500+-0.6614i 1.0000 0.1150  
## 1-1.0000B+0.8000B^2 0.6250+-0.9270i 0.8944 0.1556  
## 1+0.0000B+0.5000B^2 0.0000+-1.4142i 0.7071 0.2500  
##   
##

factor.wge(phi = theta)

##   
## Coefficients of Original polynomial:   
## 0.0000 -0.9000   
##   
## Factor Roots Abs Recip System Freq   
## 1+0.0000B+0.9000B^2 0.0000+-1.0541i 0.9487 0.2500  
##   
##

* p = 4
* d = 3
* q = 2
* ARUMA(4,3,2)

## d

phi = c(0.5, 0.5, 0, 1, -0.5, -0.5)  
theta = c(0, 0.81)  
  
factor.wge(phi = phi)

##   
## Coefficients of Original polynomial:   
## 0.5000 0.5000 0.0000 1.0000 -0.5000 -0.5000   
##   
## Factor Roots Abs Recip System Freq   
## 1-1.0000B 1.0000 1.0000 0.0000  
## 1+1.0000B -1.0000 1.0000 0.5000  
## 1+0.0000B+1.0000B^2 0.0000+-1.0000i 1.0000 0.2500  
## 1-1.0000B 1.0000 1.0000 0.0000  
## 1+0.5000B -2.0000 0.5000 0.5000  
##   
##

factor.wge(phi = theta)

##   
## Coefficients of Original polynomial:   
## 0.0000 0.8100   
##   
## Factor Roots Abs Recip System Freq   
## 1+0.9000B -1.1111 0.9000 0.5000  
## 1-0.9000B 1.1111 0.9000 0.0000  
##   
##

* p = 1
* d = 5
* q = 2
* ARUMA(1,5,2)

# 5.7

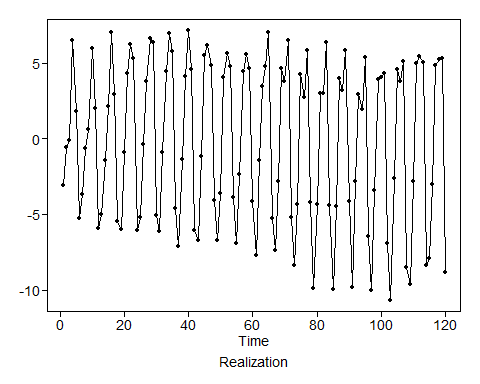
## a

phi = c(rep(0,5),1)  
factor.wge(phi = phi)

##   
## Coefficients of Original polynomial:   
## 0.0000 0.0000 0.0000 0.0000 0.0000 1.0000   
##   
## Factor Roots Abs Recip System Freq   
## 1+1.0000B -1.0000 1.0000 0.5000  
## 1+1.0000B+1.0000B^2 -0.5000+-0.8660i 1.0000 0.3333  
## 1-1.0000B+1.0000B^2 0.5000+-0.8660i 1.0000 0.1667  
## 1-1.0000B 1.0000 1.0000 0.0000  
##   
##

## b

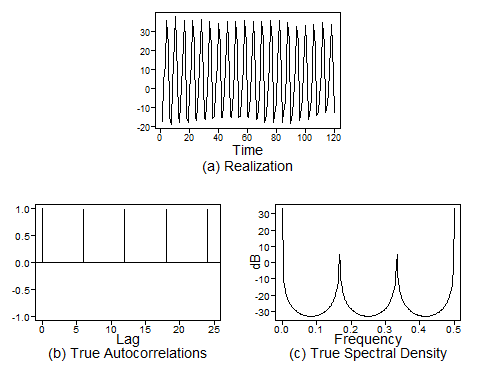
x = gen.aruma.wge(n = 120, s = 6)



We can clearly see the seasonal behavior in this data. The pattern repeats every 6 cycles.

## c

# Since we can not plot a non-stationary model here, we will replace this with a stationary (almost non-stationary) counterpart  
pt = plotts.true.wge(n = 120, phi = c(rep(0, 5), 0.999))

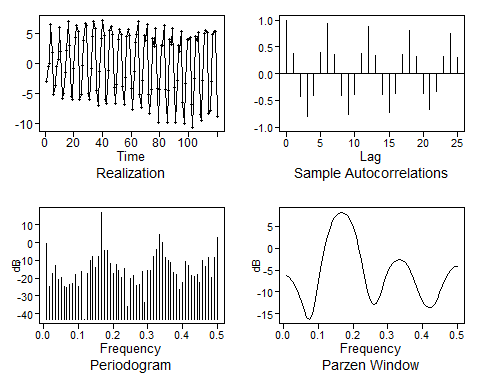


The ACF shows that there is a seasonal behavior with s = 6 since the ACF at lag = 6 is almost perfectly correlated with the ACF at lag = 0.

We also see peaks in the Spectral Density at f = 0, 0.16, 0.33 and 0.5

## d

px = plotts.sample.wge(x)



These results are very much similar to what we saw in c. The ACF shows the seasonal behavior wth s = 6 and the spectral density shows peaks at roughly the same points (with the peak at f = 0 being slghtly depressed compared to the true spectral density).