Assignment\_3

Shritej Chavan (BE14B004)

September 26, 2018

### Q.1 ARIMA process

1. Parametrization of h[.]
2. AR(2) model :

Factorizing the denominator , we get

Therefore

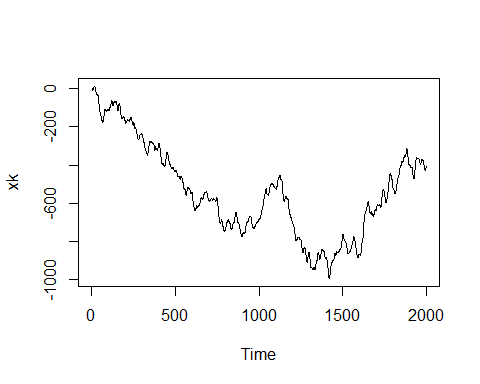
From binomial expansion, we get

Therefore,

1. ARMA(1,1) process
2. After binomial expansion of the given transfer function operator, we get

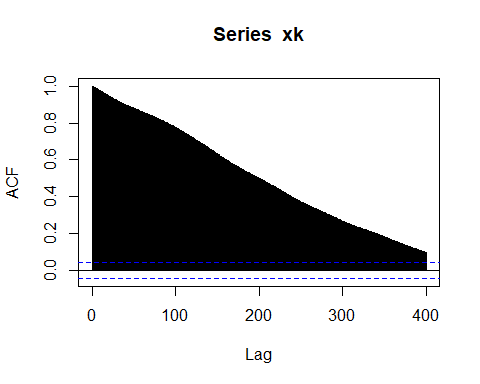
Therfore

#Loading the given time series   
load("a3\_q1.Rdata")  
#Plotting the time series  
plot(xk)

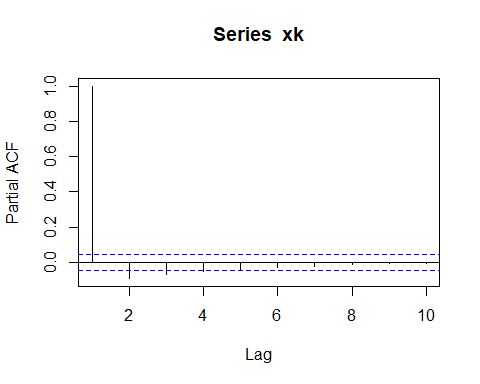


In the above plot of the given series we can clearly see that the series is not stationary

#Plotting the ACF and PACF of the given time series   
  
acf(xk, lag.max = 400)



pacf(xk,lag.max = 10)



Now, observing the ACF and PACF of the given series and ASSUMING it to be STATIONARY, we will fit an AR(4) model given the abrupt change in the PACF after lag - 4.

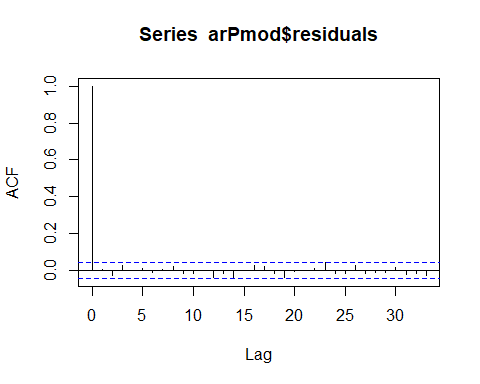
arPmod <- arima(xk,order=c(4,0,0), include.mean = FALSE)  
print(arPmod)

##   
## Call:  
## arima(x = xk, order = c(4, 0, 0), include.mean = FALSE)  
##   
## Coefficients:  
## ar1 ar2 ar3 ar4  
## 2.6742 -2.5958 1.0960 -0.1743  
## s.e. 0.0220 0.0588 0.0588 0.0220  
##   
## sigma^2 estimated as 1.007: log likelihood = -2847.35, aic = 5704.7

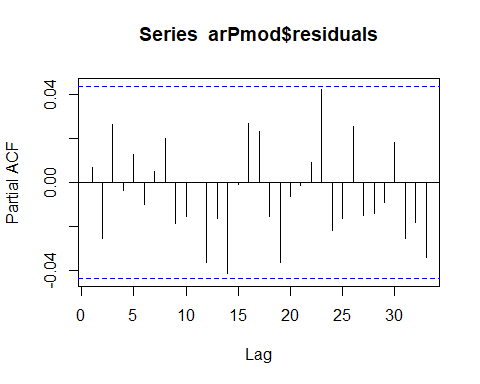
Now checking the coefficients of the above AR(4) model and it’s S.E we can clearly see that [ar\_i - 3(se\_i), ar\_i + 3(se\_i)] does not include zero. Test for Overfitting passed.

Checking for underfitting, we plot the residuals

acf(arPmod$residuals)



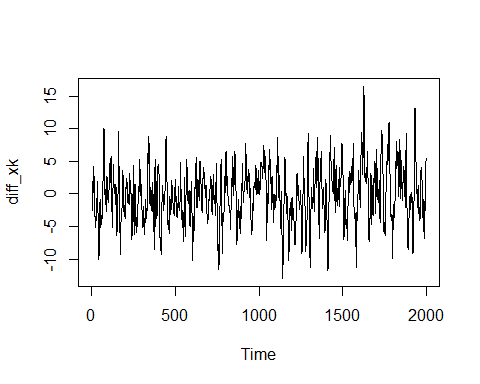
pacf(arPmod$residuals)



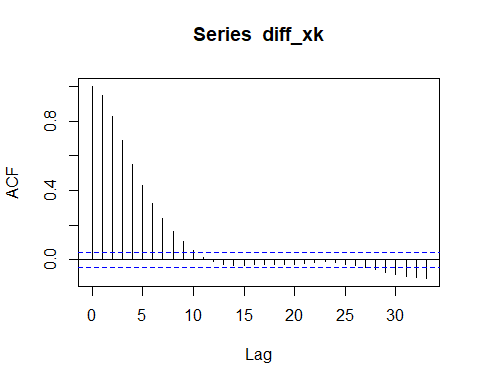
Since the residuals clearly shows the white-noise characteristics we can say the model has passed the underfitting test

Now since we assumed the above series which clearly non - stationary (by observation) as a stationary process we will filter out non-stationarity by differencing

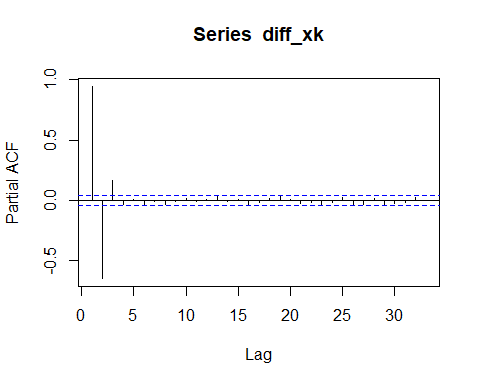
diff\_xk <- diff(xk)  
plot(diff\_xk)



acf(diff\_xk)



pacf(diff\_xk)



Observing the ACF and PACF of the differenced series we can either assume MA(10) or AR(3) process for the differenced series

ma10mod <- arima(diff\_xk,c(0,0,10))  
print(ma10mod)

##   
## Call:  
## arima(x = diff\_xk, order = c(0, 0, 10))  
##   
## Coefficients:  
## ma1 ma2 ma3 ma4 ma5 ma6 ma7 ma8  
## 1.6839 1.8658 1.7602 1.4967 1.1871 0.8722 0.5992 0.3912  
## s.e. 0.0224 0.0438 0.0598 0.0685 0.0694 0.0657 0.0621 0.0574  
## ma9 ma10 intercept  
## 0.2039 0.0555 -0.1952  
## s.e. 0.0444 0.0234 0.2498  
##   
## sigma^2 estimated as 1.013: log likelihood = -2850.68, aic = 5725.37

Looking at the coefficients we can clearly see that the MA(10) model has overfitted the parameters, Therefore MA(10) model is rejected

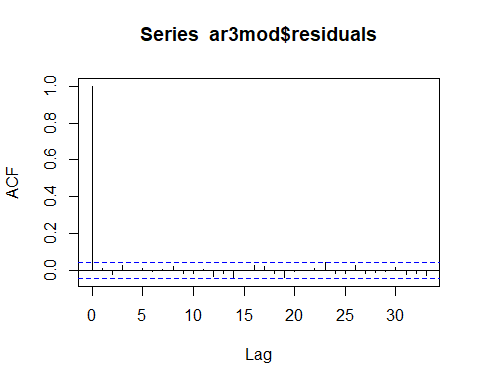
Therefore moving on to AR(3) process

ar3mod <- arima(diff\_xk,c(3,0,0))  
print(ar3mod)

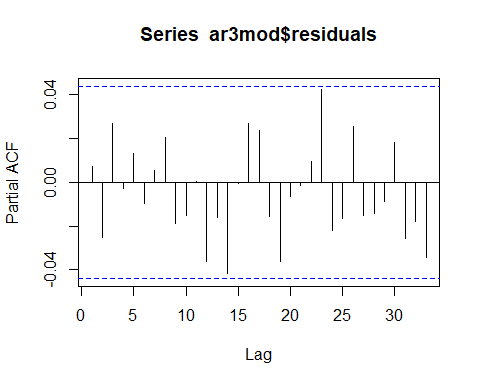
##   
## Call:  
## arima(x = diff\_xk, order = c(3, 0, 0))  
##   
## Coefficients:  
## ar1 ar2 ar3 intercept  
## 1.674 -0.9213 0.174 -0.1897  
## s.e. 0.022 0.0382 0.022 0.3053  
##   
## sigma^2 estimated as 1.008: log likelihood = -2845.78, aic = 5701.57

Since the coefficients passes the overfitting test we can now check the residuals

acf(ar3mod$residuals)



pacf(ar3mod$residuals)



Since the residuals clearly shows the white-noise characteristics we can say the model has passed the underfitting test

Now comparing the sigma of the residuals and the aic of the process

(slightly) and

Now the test of parsimony comes to the rescue, showing that the AR(3) model on the differenced series is the appropriate fit with coefficients shown by the var above.

Therefore, to conclude the given series can be fitted by ARIMA(3,1,0) model with coefficients same as where 1 stands for ARMA(3,0) model on the series that is differenced 1 time.

1. The given process

In the above equation substituting the value of and subsequently, we get

Therefore

Therefore

###Q.2 Computing correlations

The given process is

where, u[k] and e[k] both are Gaussian WN processes with zero mean.

Therefore, $E(y[k]) = 0 $

Substituting the values for lag = 0 and 1, we get

From equation (4) & (5), we get

From equation (6), we get

Solving for (1) and (2) we get

Also,

From equation (3), substituting the value of lag = 3, we get

Similarly for lag = 4, we get

Therefore we can see that,

Since, therefore above correlation we keep decreasing with lag. Hence it will be maximum at lag = 2.

We know that the delay is observed at maximum correlation, therefore we can say that the delay between and is 2.

### Q.3 Process with Trend

Given series is of the type

where, is stationary ARMA process

For the above type of series two different ways a model can be build 1. Filtering approach 2. Differencing approach

The filtering approach relies on estimation of trend followed by the development of an ARMA model for the stationary residuals. The differencing approach, in contrast, eliminates the trend implicitly and fits an ARMA model.

Another difference is that an ARIMA model fixes one of the poles of the model to the unit circle irrespective of whether the actual process has a pole on the unit circle or not.

The above process has its own merits and demerits:

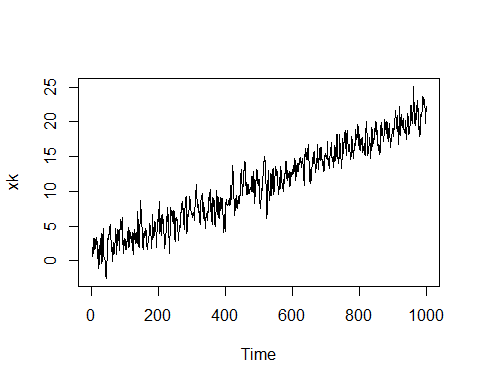
The merit is that the estimation of “stationary” poles, but close to unit circle, can result in models with confidence regions containing non-stationary models.By forcing the pole to the unit circle a priori, this situation is avoided.

The demerits are however that a stationary process with slowly decaying process acquires a non-stationary representation.

In system identification, differencing has to be carried out with caution since it amplifies the noise levels in the differenced data thereby decreasing the SNR

However single degree of differencing introduces a non-inverible zero in the differenced series

load("a3\_q3.Rdata")  
plot(xk)

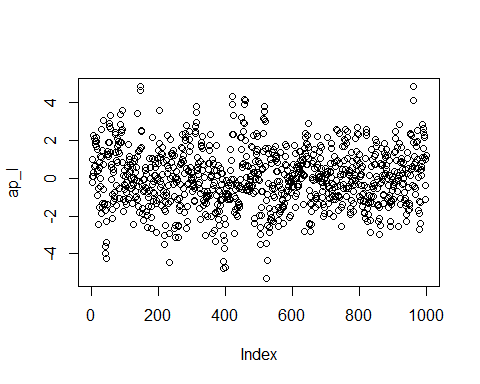


lin = 1:1000  
ap <- lm(xk~lin)  
print(ap)

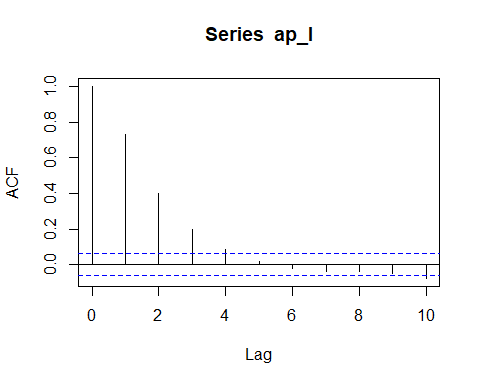
##   
## Call:  
## lm(formula = xk ~ lin)  
##   
## Coefficients:  
## (Intercept) lin   
## 0.87462 0.02012

Therefore checking the residuals after eliminating the linear trend we get

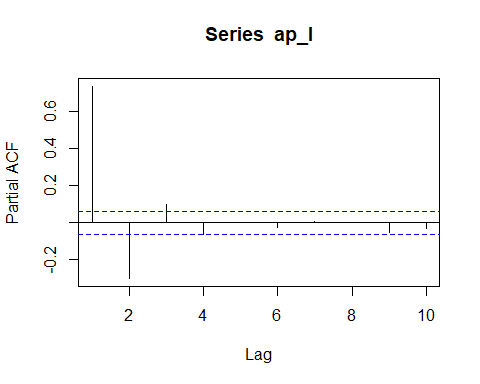
ap\_l = ap$residuals  
plot(ap\_l)



acf(ap\_l, lag.max = 10)



pacf(ap\_l, lag.max = 10)



Observing the ACF and PACF of the linearly fitted (for trend) series we can either assume MA(4) or AR(3) process.

Starting with MA(4) process

ma4mod <- arima(ap\_l,c(0,0,4))  
print(ma4mod)

##   
## Call:  
## arima(x = ap\_l, order = c(0, 0, 4))  
##   
## Coefficients:  
## ma1 ma2 ma3 ma4 intercept  
## 0.9898 0.5448 0.2648 0.0925 0.0043  
## s.e. 0.0315 0.0423 0.0407 0.0319 0.0904  
##   
## sigma^2 estimated as 0.9788: log likelihood = -1408.72, aic = 2829.44

Observing the coefficients and their standard deviations we can say that the above model fails test for overfitting.

Now checking the fit of AR(3) model,

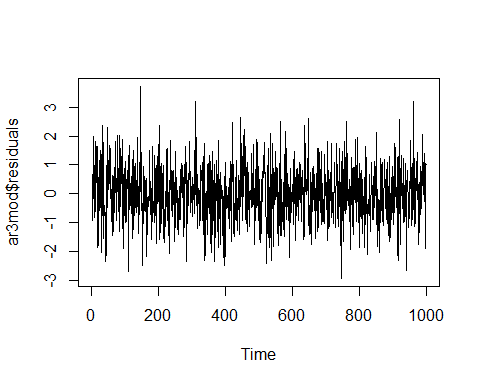
ar3mod = arima(ap\_l, c(3,0,0))  
print(ar3mod)

##   
## Call:  
## arima(x = ap\_l, order = c(3, 0, 0))  
##   
## Coefficients:  
## ar1 ar2 ar3 intercept  
## 0.9858 -0.3988 0.1001 0.0051  
## s.e. 0.0315 0.0425 0.0315 0.0999  
##   
## sigma^2 estimated as 0.9799: log likelihood = -1409.27, aic = 2828.53

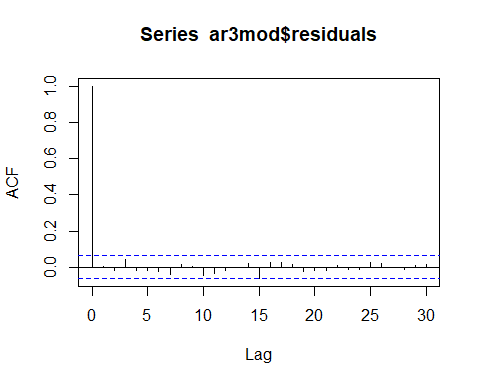
Model passes the test for overfitting

Now checking the underfitting

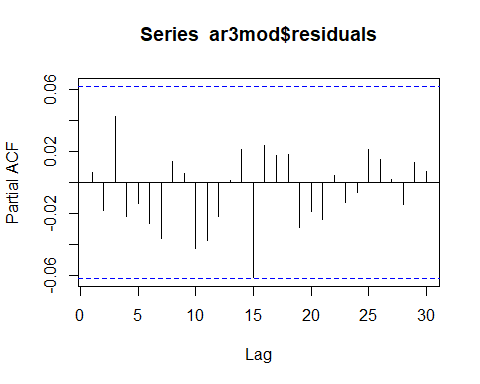
plot(ar3mod$residuals)



acf(ar3mod$residuals)



pacf(ar3mod$residuals)

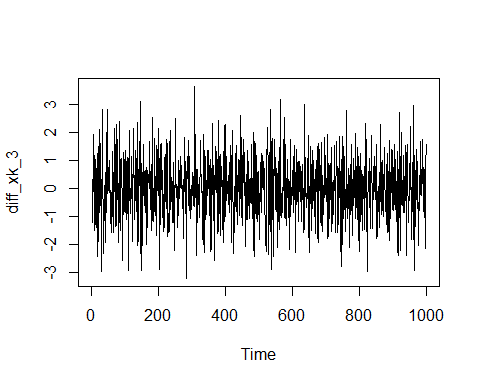


Since the residuals show the WN characteristics we can say that the series passes the underfitting test

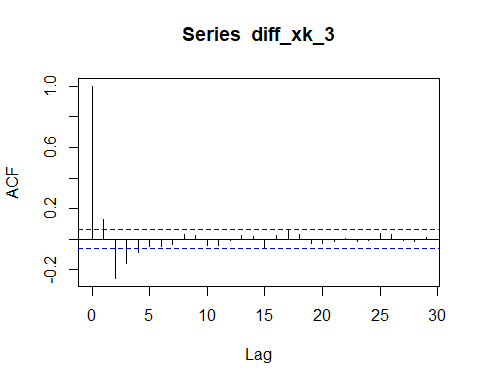
So the given series can be build by below model after fitting a linear trend

b)

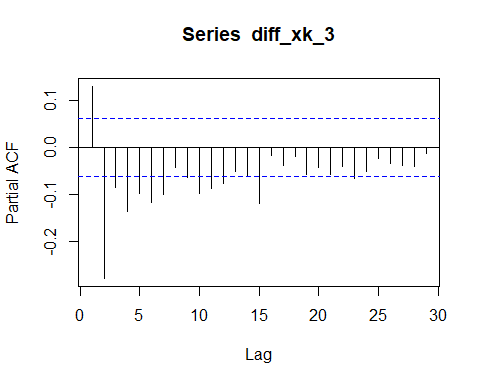
#Differencing the given series in order to remove the linear polynomial trend   
diff\_xk\_3 = diff(xk)   
plot(diff\_xk\_3)



acf(diff\_xk\_3)



pacf(diff\_xk\_3)



By observing the ACF and PACF of the differenced series, we can say that the series cannot model by purely AR or MA process

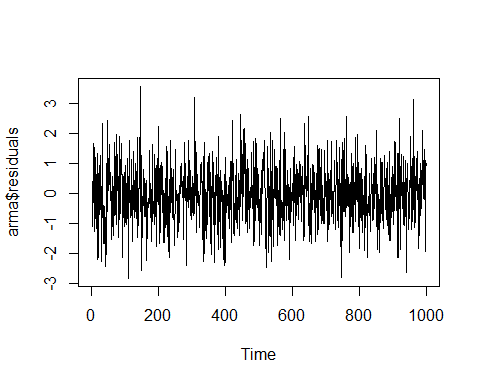
Therefore by trial and error ARMA(1,2) model is chosen

arma = arima(diff\_xk\_3, order = c(1,0,2))  
print(arma)

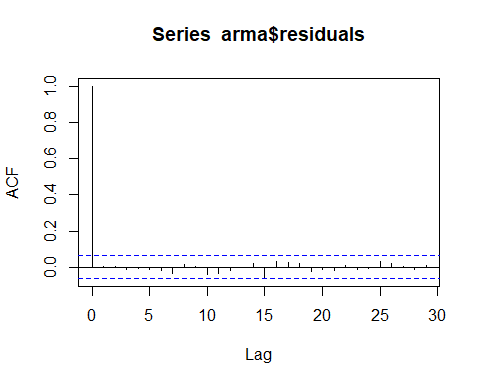
##   
## Call:  
## arima(x = diff\_xk\_3, order = c(1, 0, 2))  
##   
## Coefficients:  
## ar1 ma1 ma2 intercept  
## 0.5575 -0.5685 -0.4315 0.0201  
## s.e. 0.0327 0.0352 0.0350 0.0004  
##   
## sigma^2 estimated as 0.9781: log likelihood = -1409.24, aic = 2828.48

Coefficients and respective standard deviations checked Test for overfitting passed

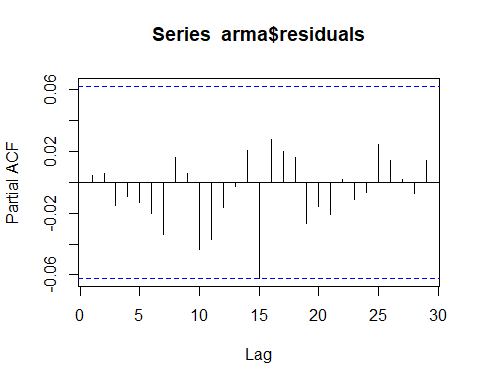
plot(arma$residuals)



acf(arma$residuals)



pacf(arma$residuals)



Residuals showing the WN characteristics Test for underfitting passed

Therefore, to conclude the given series can be fitted by ARIMA(2,1,1) model with coefficients same as where 1 stands for ARMA(2,1) model on the series that is differenced 1 time.

The model can be written as follows

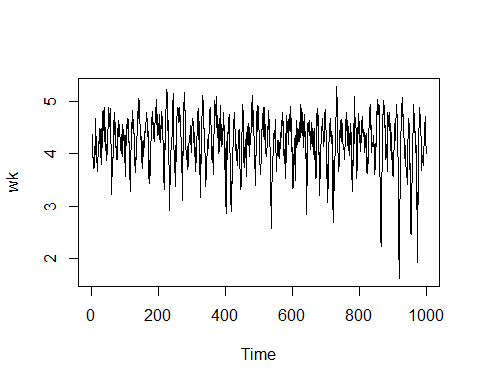
### Q.4 Variance Non - Stationarity

library(MLmetrics)

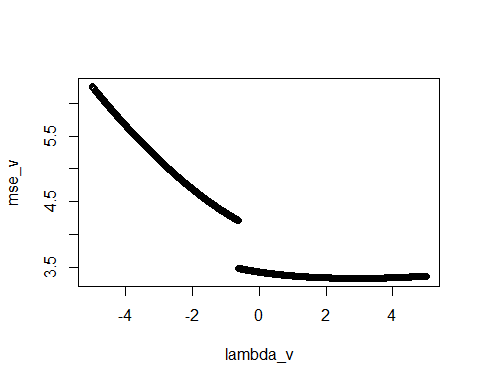
##   
## Attaching package: 'MLmetrics'

## The following object is masked from 'package:base':  
##   
## Recall

# Loading and Plotting the given data   
  
load("a3\_q4.Rdata")  
  
plot(wk)



train = wk[1:800] # Training purpose   
  
test = wk[801:1000] #Test data   
  
#List of the Box - Cox transformation parameters   
lambda\_v = seq(-5,5,0.001)  
  
#Empty matrix for list mse values   
mse\_v = {}  
  
index = 0   
  
  
for (lambda in lambda\_v) {  
   
 index = index + 1  
   
 if (lambda == 0){  
   
 bxcx = log(wk)  
 }  
   
 else{  
   
 bxcx = ((wk)^lambda -1)/lambda  
   
 #Fitting an AR model on the first 800 observations  
 armod = ar(bxcx[1:800], aic = TRUE, order.max = 7)  
   
 #One step ahead prediction using filter routine   
 test\_filter = filter(test, armod$ar, method = "convolution", sides =1 )  
   
 #Trimming the first 'NA' values appearing due to filtering   
 mse\_v[index] = MSE(test\_filter[(1+armod$order):200], test[(1+armod$order):200])  
   
 }  
}  
  
#Plotting Mean Squared error on each lambda wrt lambda  
plot( lambda\_v, mse\_v)

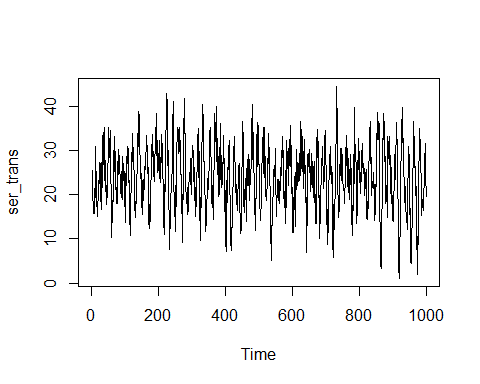


#Finding the optimal lambda   
  
opt\_lambda = lambda\_v[which.min(mse\_v)]  
print(opt\_lambda)

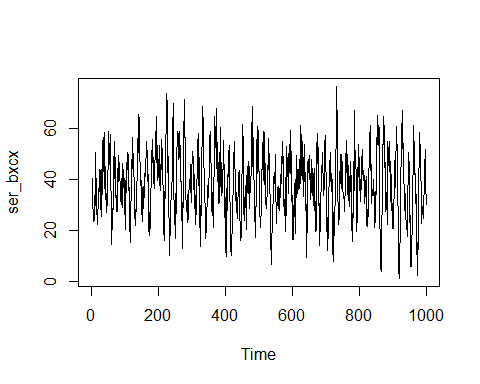
## [1] 2.928

The above value is the value of the optimal corresponding to the case where MSE is minimum.

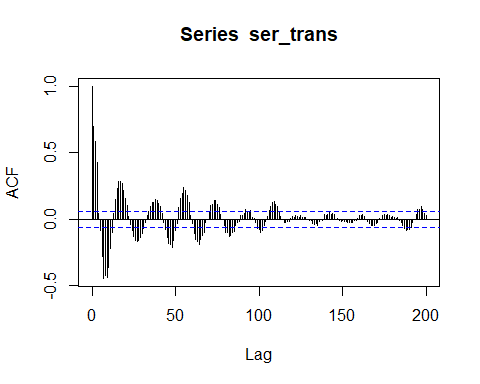
library(forecast)  
  
lambda\_bxcx = BoxCox.lambda(wk, method = 'guerrero', lower=-5,upper=5)  
  
#Transformed Series with lambda from the BoxCox routine   
ser\_bxcx = BoxCox(wk,lambda\_bxcx)  
  
#Series with lambda obtained from the robust code   
ser\_trans = ((wk)^opt\_lambda - 1)/opt\_lambda   
  
plot(ser\_trans)



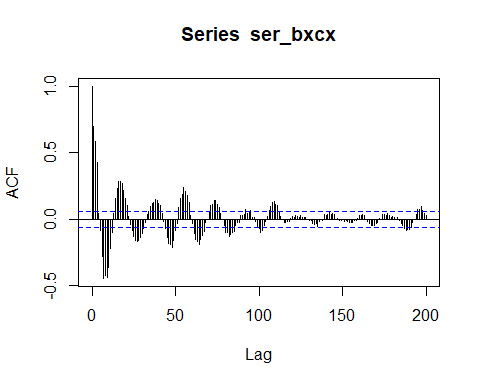
plot(ser\_bxcx)



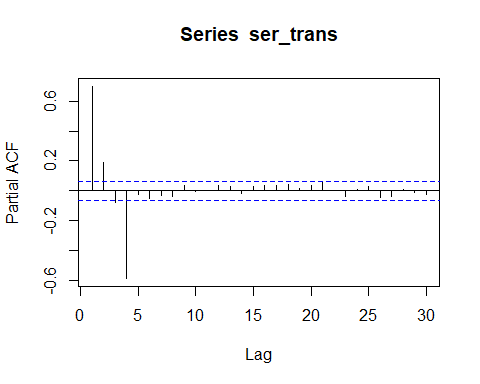
acf(ser\_trans, lag.max = 200)



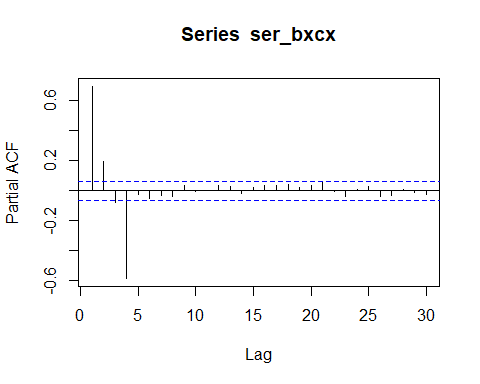
acf(ser\_bxcx, lag.max = 200)



pacf(ser\_trans)



pacf(ser\_bxcx)



From PACF of the both transformed series we can try building an AR(4) model for both and check whether it’s an optimal fit

armod\_tr = arima(ser\_trans, c(4,0,0))  
armod\_bx = arima(ser\_bxcx, c(4,0,0))  
print(armod\_tr)

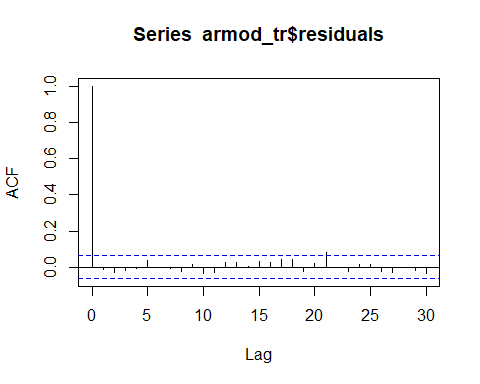
##   
## Call:  
## arima(x = ser\_trans, order = c(4, 0, 0))  
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 intercept  
## 0.5321 0.3768 0.2616 -0.5882 23.9859  
## s.e. 0.0255 0.0295 0.0295 0.0256 0.3037  
##   
## sigma^2 estimated as 16.06: log likelihood = -2808.43, aic = 5628.86

print(armod\_bx)

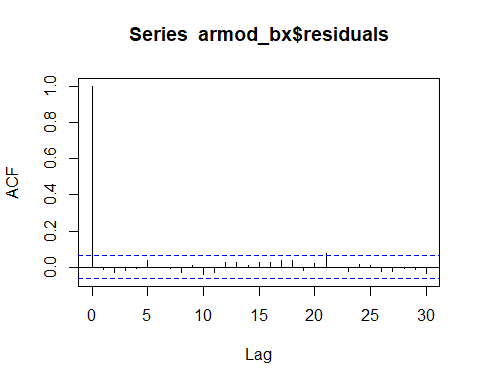
##   
## Call:  
## arima(x = ser\_bxcx, order = c(4, 0, 0))  
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 intercept  
## 0.5306 0.3752 0.2623 -0.5867 38.2313  
## s.e. 0.0256 0.0295 0.0295 0.0256 0.5385  
##   
## sigma^2 estimated as 50.76: log likelihood = -3383.68, aic = 6779.35

Looking at the coefficients we can say that, there is no overfitting in both cases

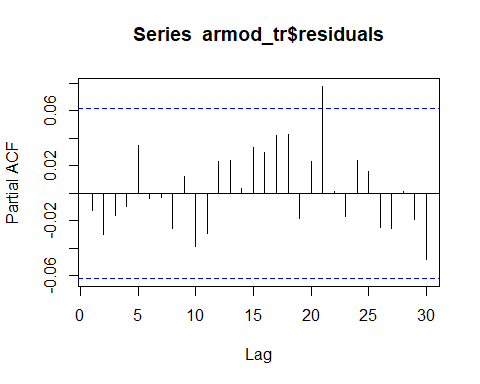
acf(armod\_tr$residuals)



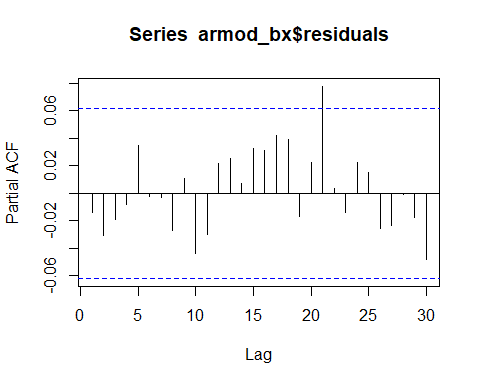
acf(armod\_bx$residuals)



pacf(armod\_tr$residuals)



pacf(armod\_bx$residuals)



Also we can roughly say that there is no underfitting as well

After fitting an AR(4) for both transformed series we can observe that the coefficients come out to be almost equal. Therefore writing an appropriate model for the given data

where, and obtain from the Box-Cox routine with AIC = 6779.35

or the other model can be written as

where, and obtain from the code written with AIC = 5634.61