Assignment\_5

Shritej Chavan (BE14B004)

November 1, 2018

#### Q.1 Sample simulations of ARIMA models

library(forecast)  
  
aic\_mod = matrix(data = NA, nrow = 100, ncol = 3)   
#aic values of 3 models of all 100 different noise realizations   
  
order\_mod = matrix(data = NA, nrow = 100, ncol = 4)   
#order of 3 models of all 100 different noise realizations   
  
best\_mod = {}  
  
for (i in 1:100) {  
vk = arima.sim(list(ar=c(-0.2, 0, -0.1)), 600) #Generating data  
  
#AR model estimation  
  
aic\_ar = {}  
  
for (j in 1:10) {  
 armod = arima(vk, order = c(j,0,0))  
 aic\_ar[j] = armod$aic  
   
   
}  
aic\_mod[i,1] = min(aic\_ar)  
order\_mod[i,1] = which.min(aic\_ar)  
  
  
# MA model estimation  
  
aic\_ma={}  
for (k in 1:10){  
  
 mamod=arima(vk,order = c(0,0,k))  
 aic\_ma[k]=mamod$aic  
}  
  
aic\_mod[i,2]=min(aic\_ma)  
order\_mod[i,2]=which.min(aic\_ma)  
  
  
  
# ARMA model estimation  
  
armamod <- auto.arima(vk,seasonal = FALSE,d=0,D=0,max.p=10,max.q=10,start.p = 1,start.q = 1)   
# auto.arima() searches through all the possible arma models and chooses the one with minimum aic value  
  
  
aic\_mod[i,3] = armamod$aic  
  
order\_mod[i,3] = armamod$arma[1]  
  
order\_mod[i,4] = armamod$arma[2]  
  
best\_mod[i] = which.min(aic\_mod[i,])  
}  
  
ar\_ = which(best\_mod == 1) ## No of times AR model is selected   
print(ar\_)

## [1] 1 4 6 7 9 11 15 24 27 28 29 32 34 40 41 43 48 52 55 57 60 61 62  
## [24] 64 65 66 69 76 77 79 85 89 90

ma\_ = which(best\_mod == 2) ## No of times MA model is selected  
print(ma\_)

## [1] 8 21 26 33 38 42 45 46 53 54 58 59 72 80 97

arma\_ = which(best\_mod == 3) ##No of times ARMA model is selected  
print(arma\_)

## [1] 2 3 5 10 12 13 14 16 17 18 19 20 22 23 25 30 31  
## [18] 35 36 37 39 44 47 49 50 51 56 63 67 68 70 71 73 74  
## [35] 75 78 81 82 83 84 86 87 88 91 92 93 94 95 96 98 99  
## [52] 100

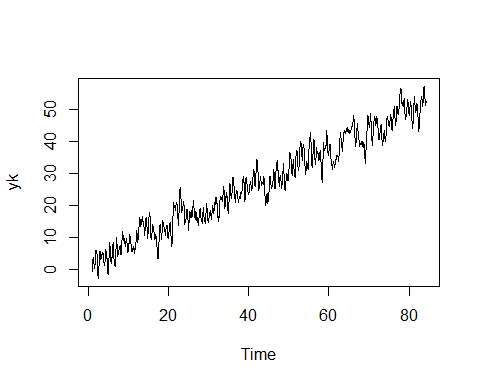
ar3 = which(order\_mod[,1]==3) ## No of times AR(3) model is selected  
  
true\_process = length(intersect(ar3,ar\_))  
  
print(true\_process) ##No of times the model matches with AR(3) process

## [1] 21

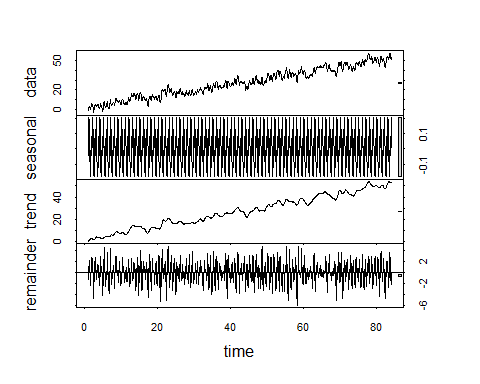
Similarly for N =100 sample size, 5 number of times the model matches with the true process

### Q2 Fitting Seasonal Models

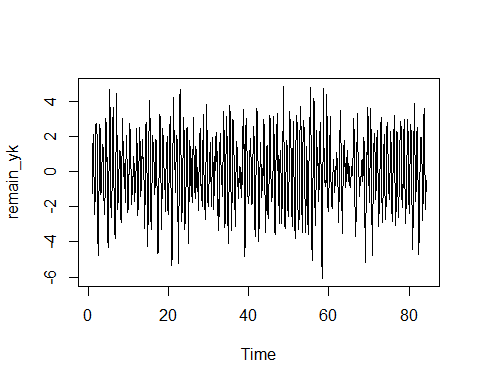
library(forecast)  
  
load('sarima\_data.Rdata')  
  
plot(yk)



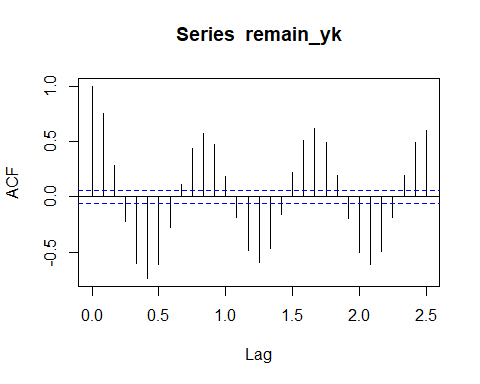
yk\_stl = stl(yk,s.window = "periodic")  
  
plot(yk\_stl)



remain\_yk = remainder(yk\_stl)  
  
plot(remain\_yk)



acf(remain\_yk)

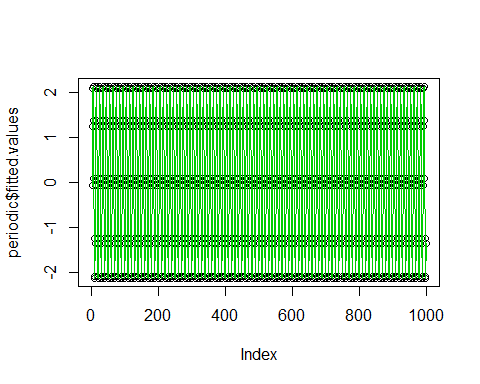


We can observe that acf of the remainder series is periodic with a period of 10.

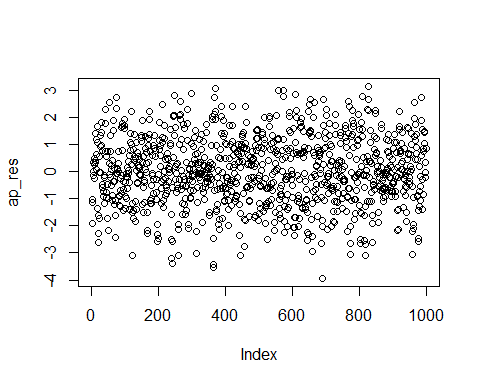
lin = 1:length(yk)  
  
periodic = lm(remain\_yk~sin(2\*pi\*lin/10)+I(cos(2\*pi\*lin/10)))  
  
summary(periodic)

##   
## Call:  
## lm(formula = remain\_yk ~ sin(2 \* pi \* lin/10) + I(cos(2 \* pi \*   
## lin/10)))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.9377 -0.8288 -0.0292 0.8523 3.1360   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.002798 0.039502 0.071 0.944   
## sin(2 \* pi \* lin/10) 1.756465 0.055865 31.441 <2e-16 \*\*\*  
## I(cos(2 \* pi \* lin/10)) -1.374654 0.055865 -24.607 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.249 on 997 degrees of freedom  
## Multiple R-squared: 0.6152, Adjusted R-squared: 0.6144   
## F-statistic: 797 on 2 and 997 DF, p-value: < 2.2e-16

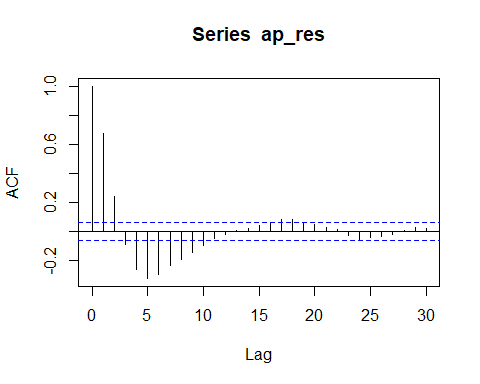
plot(periodic$fitted.values)  
  
lines(fitted(periodic)~lin,col=3)



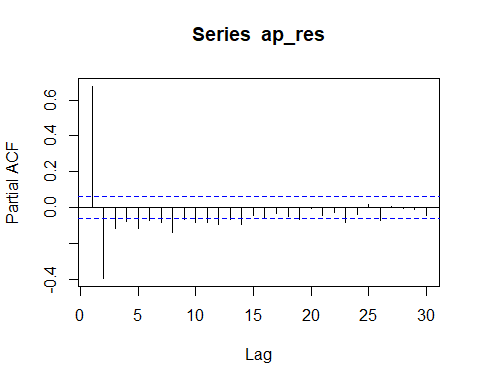
ap\_res = periodic$residuals  
  
plot(ap\_res)



acf(ap\_res)



pacf(ap\_res)



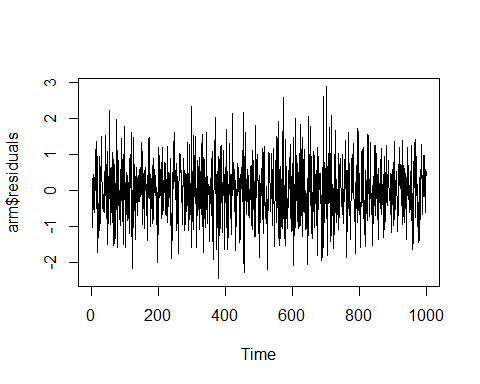
Since, PACF abruptly vanishes after lag = 2, we can choose to fit a AR(2) model on the remainder series,

arm = arima(ap\_res, order = c(2,0,0))  
summary(arm)

##   
## Call:  
## arima(x = ap\_res, order = c(2, 0, 0))  
##   
## Coefficients:  
## ar1 ar2 intercept  
## 0.9434 -0.3940 0.0010  
## s.e. 0.0290 0.0291 0.0592  
##   
## sigma^2 estimated as 0.7116: log likelihood = -1249.31, aic = 2506.63  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 3.698348e-05 0.8435815 0.6733389 85.52833 227.3035 0.8435731  
## ACF1  
## Training set -0.04540371

The coefficients are inspected for overfitting.

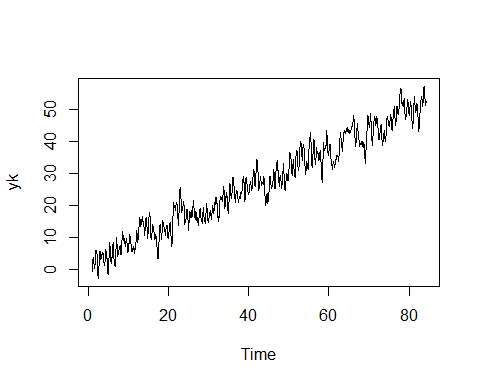
plot(arm$residuals)



The residuals shows a white noise characteristics.

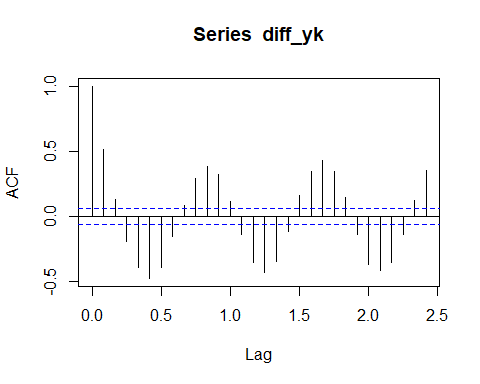
So to conclude, the remainder series obtained by stl method was fitted by periodic function of sin and cos of period 10 with lm method. Then we found out that the AR(2) is appropriate model for the then residual series.

plot(yk)



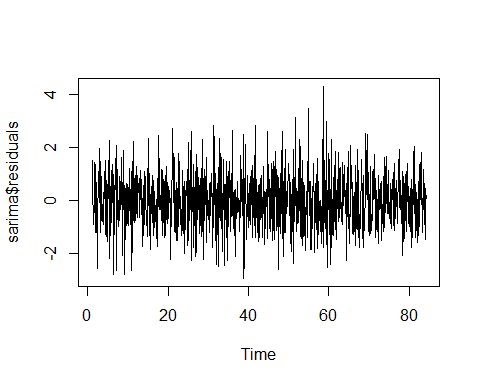
Since we see a almost linear trend in the given data, we can choose to difference it once.

diff\_yk = diff(yk)  
acf(diff\_yk)

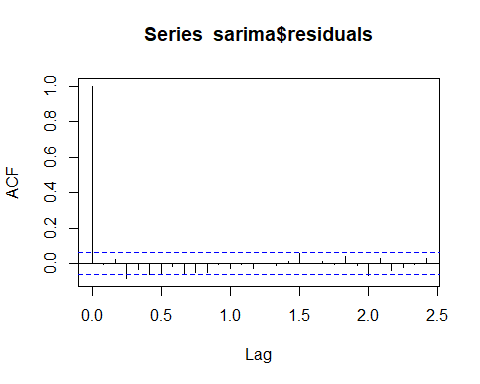


We observe that the ACF plot shows a periodicity of 10 so by trial and error and satisfying the criteria for overfitting and underfitting it was concluded that the AR(2) and SARIMA(1,1) is the best model.

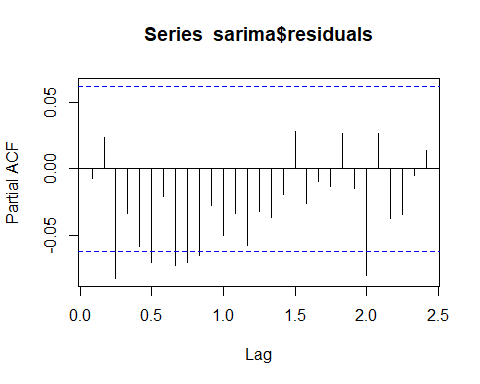
sarima = arima(diff\_yk, order = c(2,0,0), seasonal = list(order = c(1,0,1), period = 10), include.mean = TRUE)  
  
plot(sarima$residuals)



acf(sarima$residuals)



pacf(sarima$residuals)



Since the residuals show the WN characteristics, it satisfies the Underfitting criteria

### Q.3 Maximum Likelihood

For a uniform distribution over interval [a,b], p.d.f. is equal to

For independent RV Xn

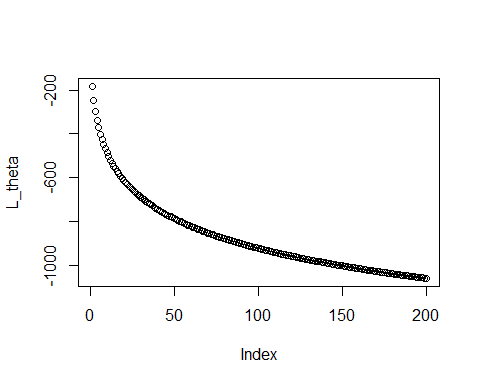
b)

The log likelihood function is a decreasing function , therefore $$ should greater than or equal to the maximum value among all the observations. Let’s say is the maximum value, therefore

load('mle\_unif.Rdata')  
  
max(xk)

## [1] 1.966975

theta = seq(max(xk),201, by=1)  
  
L\_theta={}  
  
for (i in 1:200) {  
   
 L\_theta[i] = -200\*(log(theta[i]))-(sum(xk)/theta[i])  
  
}  
plot(L\_theta)



1. Bias we know that
2. Hence,

### Q.4 Fisher’s Characteristics

According to Cramer Rao inequality the variance of an unbaised estimator of a single parameter with regular p.d.f. , is bounded by and the most efficient estimator exists if and only if

is independent of and only dependent on the observations y

#### a) - parameter

According to the given exponential distribution of parameter ,

Therefore,

The also called efficient estimator does not satisfy the sufficiency condition and is dependent on the parameter itself.

Therefore we can say that there exists no efficient estimator for estimating the parameter

#### b) k = parameter

Therefore substituting the above equation in the efficient estimator equation, we get

Since the above efficient estimator is purely a function of y, we can say that it passses the sufficiency test and also the CR inequality, hence we can say that the most efficient estimator exists for parameter .

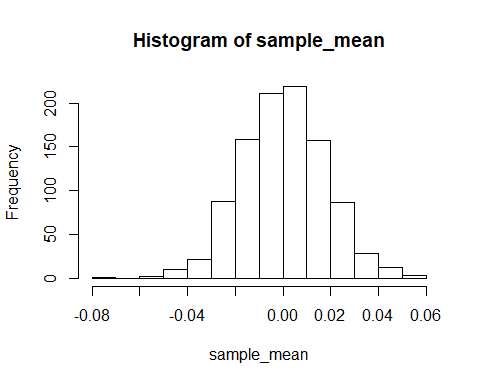
### Q5 Variability of Sample Mean

where, m and n are not equal

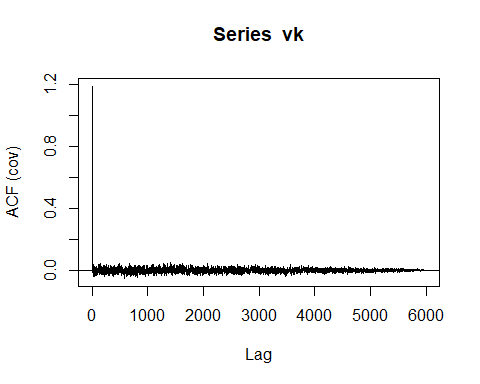
we can write the second term as autocovariance function with lag () = |n-m|

Also, for stationary process

# No. of samples  
N=1000  
  
#Sample size  
n = 6000  
  
#Conventional method for calculation of variance of sample mean estimator  
  
sample\_mean={}  
  
for (i in 1:N){  
   
 vk<-arima.sim(model=list(ma=0.4,order=c(0,0,1)),n=6000)  
 sample\_mean[i]=mean(vk)  
}  
  
hist(sample\_mean)



var\_es ={}  
for (i in 1:N){  
 var\_es[i]=(mean(sample\_mean)-sample\_mean[i])^2  
}  
  
var\_bar=sum(var\_es)/(N-1)  
  
#Calculation using the proved expression  
  
acvf\_vk = acf(vk,lag.max = n, type='cov')



acvf\_vector = acvf\_vk$acf  
  
second\_term=0  
  
for (l in 2:n){  
   
 second\_term = second\_term+ ((1-(l/n))\*acvf\_vector[l])  
}  
  
var\_2 = (1/n)\*(acvf\_vector[1]+(2\*second\_term))

The variance of the sample mean is being calculated using both the conventional and the using the proved expression. For sample size of n = 6000 and 1000 number of samples the variance using the conventional method comes out to be 1.002e-04 and using the above expression we get 3.34e-04. They can be compared