CH5350: Applied Time Series Analysis

Project Report

By:

Vishal Subbiah

MM12B035

Declaration:

I state that the following work has been done by me and only me without the assistance of any of the other students in this course (CH5350: Applied Time Series Analysis).

Sincerely,
Vishal Subbiah
(MM12B035)

Vishal Subbiah

1.R

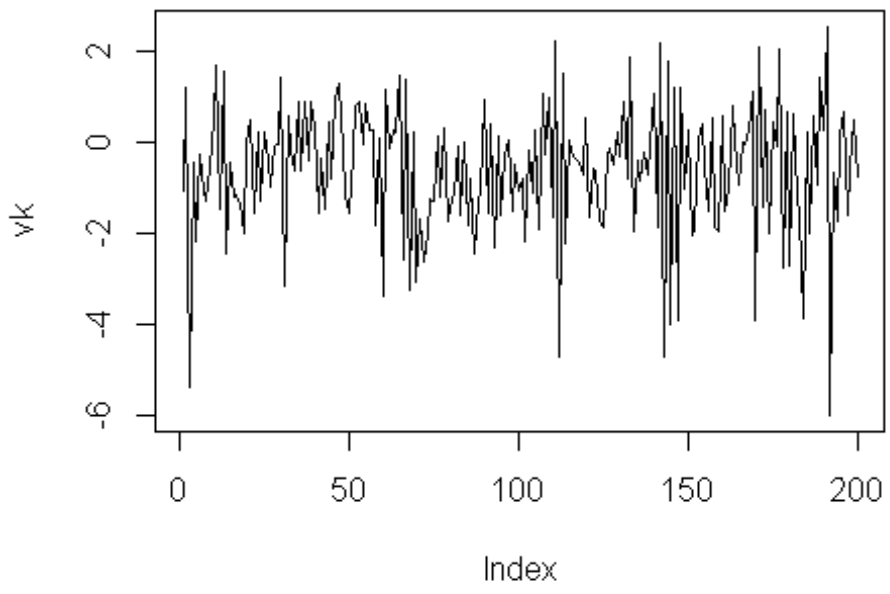
```
library(TSA)
```

```
## Loading required package: leaps
## Loading required package: locfit
## locfit 1.5-9.1    2013-03-22
## Loading required package: mgcv
## Loading required package: nlme
## This is mgcv 1.8-0. For overview type 'help("mgcv-package")'.
## Loading required package: tseries
##
## Attaching package: 'TSA'
##
## The following objects are masked from 'package:stats':
##
##     acf, arima
##
## The following object is masked from 'package:utils':
##
##     tar
```

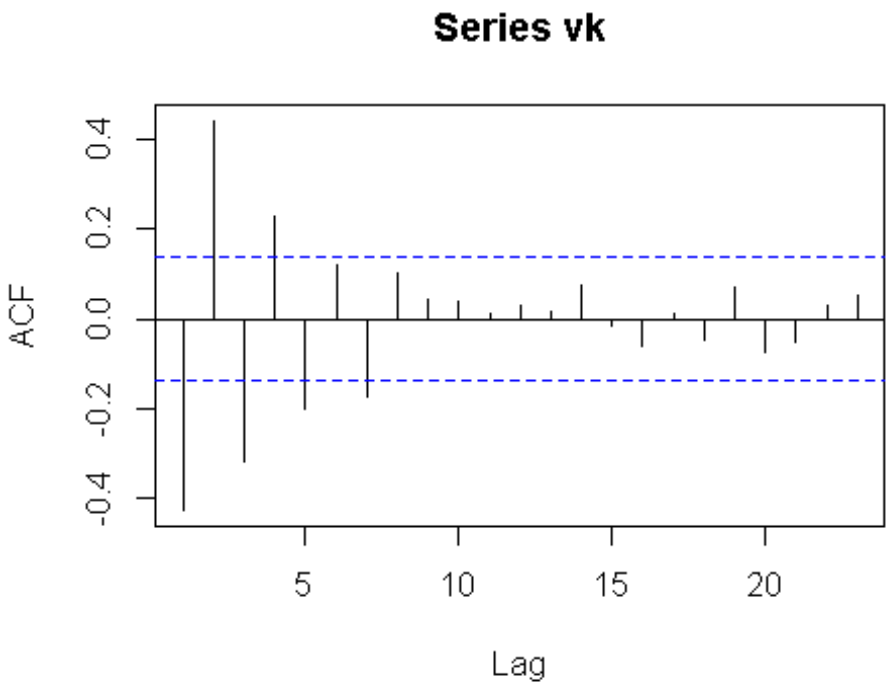
```
library(tsDyn)
```

```
## Warning: package 'tsDyn' was built under R version 3.1.2
```

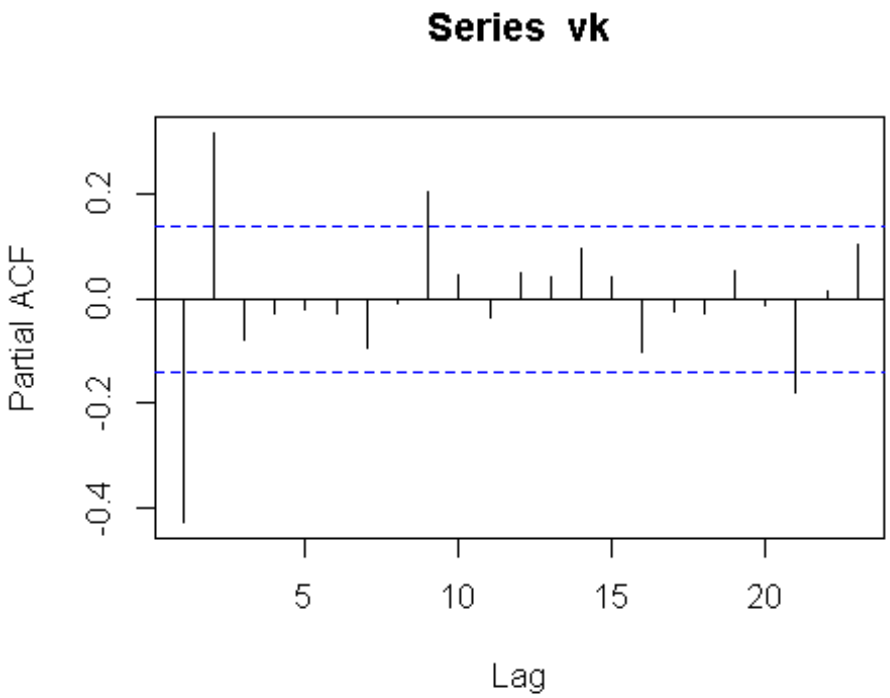
```
setwd("C:/Users/Toshiba/Desktop/vishal iit/5th sem/Applied time series analysis/assignments/project")
load("projq1a.Rdata")

plot(vk,type='l')
```



```
acf(vk)
```

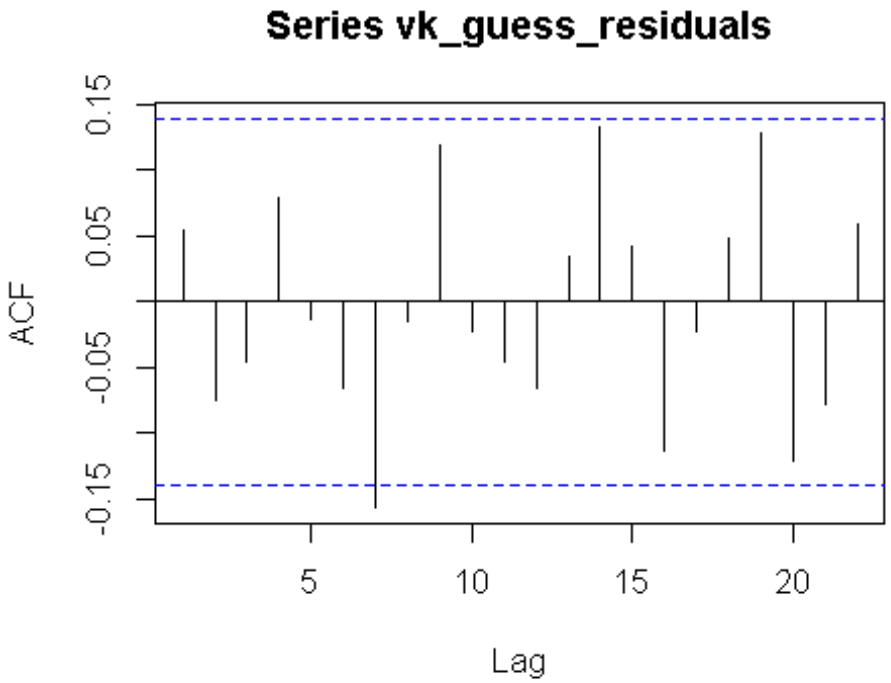



```
pacf(vk)
```

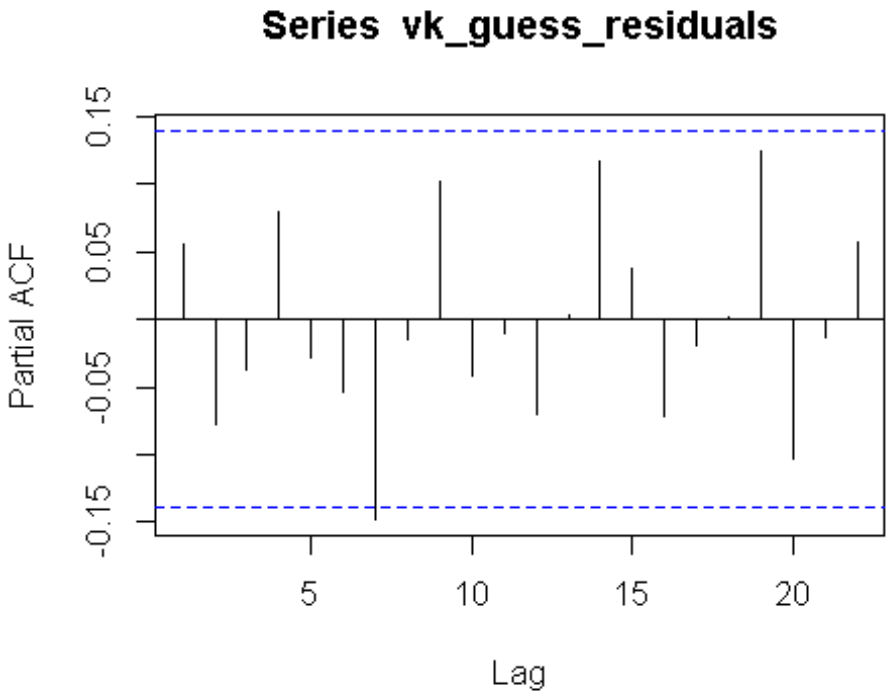



```
nonlinear_test=Tsay.test(vk,order=2)# non linear

vk_guess=tar(vk,2,1,1,method="MAIC",estimate.thd=TRUE)
```

```
vk_guess_residuals=vk_guess$residuals
acf(vk_guess_residuals)
```

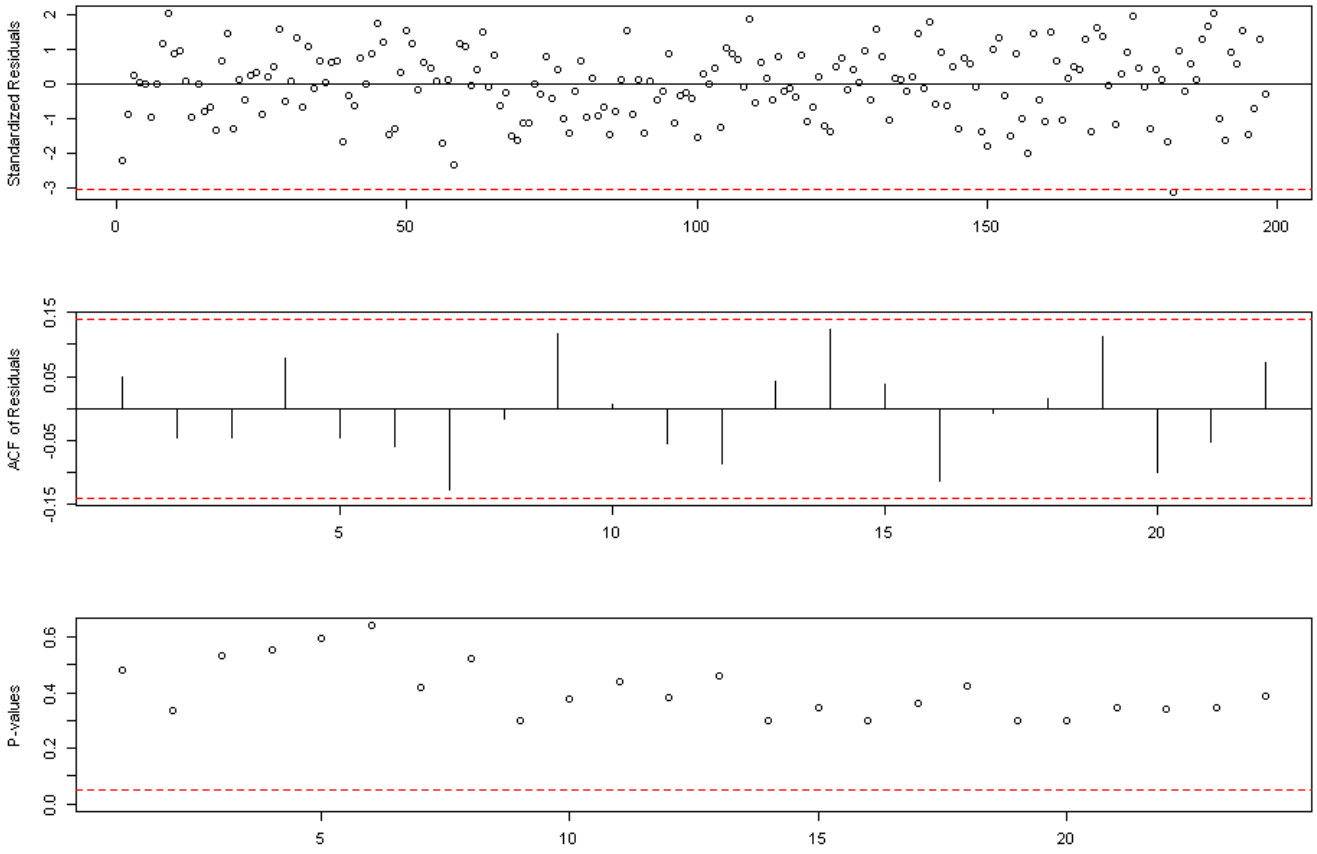



```
pacf(vk_guess_residuals)
```




```
Threshold=vk_guess$n1 # Threshold
vk_guess_1_coeff=vk_guess$qr1$coefficients # cofficients for first regime
vk_guess_2_coeff=vk_guess$qr2$coefficients # cofficients for second regime
```

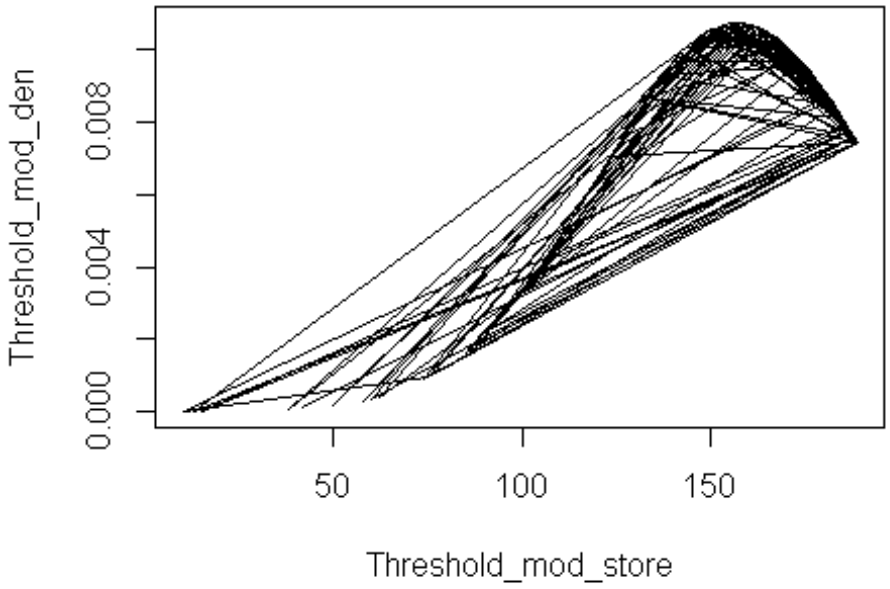
```
phi_intercept_1=vk_guess_1_coeff[1]
phi_1_1=vk_guess_1_coeff[2]
phi_2_1=vk_guess_1_coeff[3]

phi_intercept_2=vk_guess_2_coeff[1]
phi_1_2=vk_guess_2_coeff[2]

tsdiag.TAR(vk_guess)
```



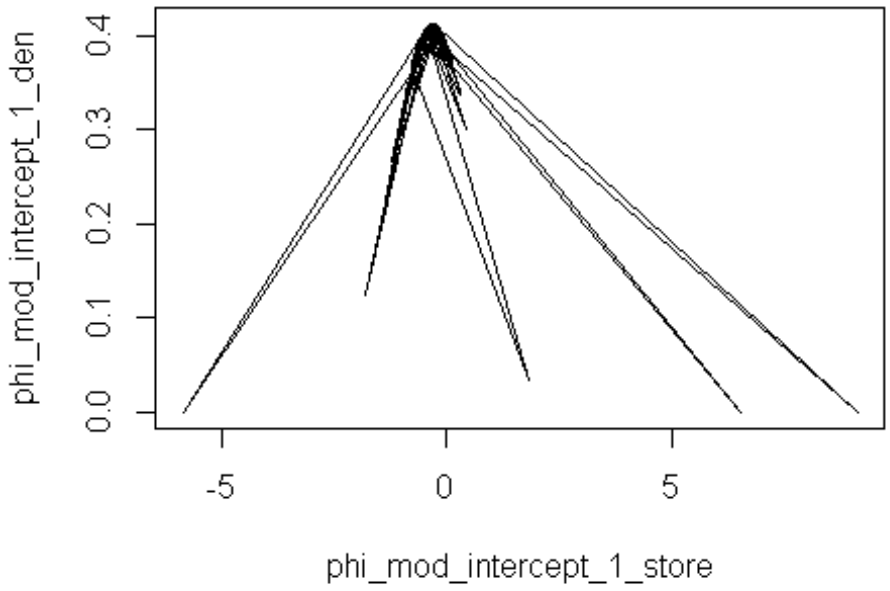

```
# part B

epsk=vk_guess_residuals

vk_guess_y=vk_guess$y

Threshold_mod_store=0
phi_mod_intercept_1_store=0
phi_mod_1_1_store=0
phi_mod_2_1_store=0

phi_mod_intercept_2_store=0
phi_mod_1_2_store=0
j=1
k=1
```

```
l=1
m=1
n=1
o=1
for (i in 1:200)
{

epskr1<-sample(epsk,size=200,replace=T)
vk_mod=vk_guess_y+epskr1
vk_guess_mod=tar(vk_mod,2,1,1,method="MAIC",estimate.thd=TRUE)

if(!is.na(vk_guess_mod$n1))
{
Threshold_mod_store[j]=vk_guess_mod$n1
j=j+1
}

vk_guess_mod_1_coeff=vk_guess_mod$qr1$coefficients # cofficients for first regime
vk_guess_mod_2_coeff=vk_guess_mod$qr2$coefficients # cofficients for second regime

if(!is.na(vk_guess_mod_1_coeff[1]))
{

phi_mod_intercept_1_store[k]=vk_guess_mod_1_coeff[1]
k=k+1
}

if(!is.na(vk_guess_mod_1_coeff[2]))
{
phi_mod_1_1_store[l]=vk_guess_mod_1_coeff[2]
l=l+1
}

if(!is.na(vk_guess_mod_1_coeff[3]))
{
phi_mod_2_1_store[m]=vk_guess_mod_1_coeff[3]
m=m+1
}

if(!is.na(vk_guess_mod_2_coeff[1]))
{
phi_mod_intercept_2_store[n]=vk_guess_mod_2_coeff[1]
n=n+1
}

if(!is.na(vk_guess_mod_2_coeff[2]))
{
phi_mod_1_2_store[o]=vk_guess_mod_2_coeff[2]
o=o+1
}

}

Threshold_mod_mean=mean(Threshold_mod_store)
```

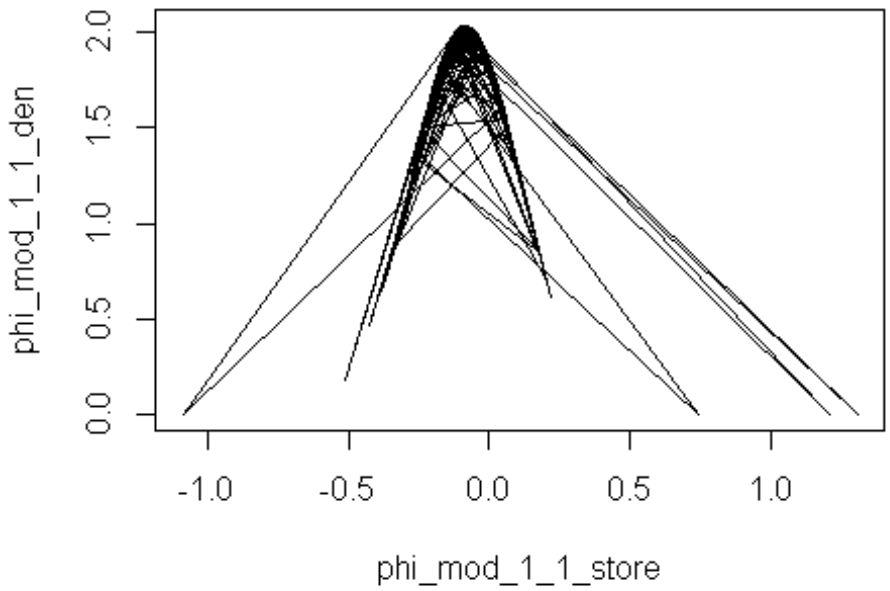
```
Threshold_mod_var=var(Threshold_mod_store)
Threshold_mod_den=dnorm(Threshold_mod_store,mean=Threshold_mod_mean,sd=sqrt(Threshold
_mod_var))
plot(Threshold_mod_store,Threshold_mod_den,type="l")
```




```
phi_mod_intercept_1_mean=mean(phi_mod_intercept_1_store)
phi_mod_intercept_1_var=var(phi_mod_intercept_1_store)
phi_mod_intercept_1_den=dnorm(phi_mod_intercept_1_store,mean=phi_mod_intercept_1_mean,sd
=sqrt(phi_mod_intercept_1_var))
plot(phi_mod_intercept_1_store,phi_mod_intercept_1_den,type='l')
```

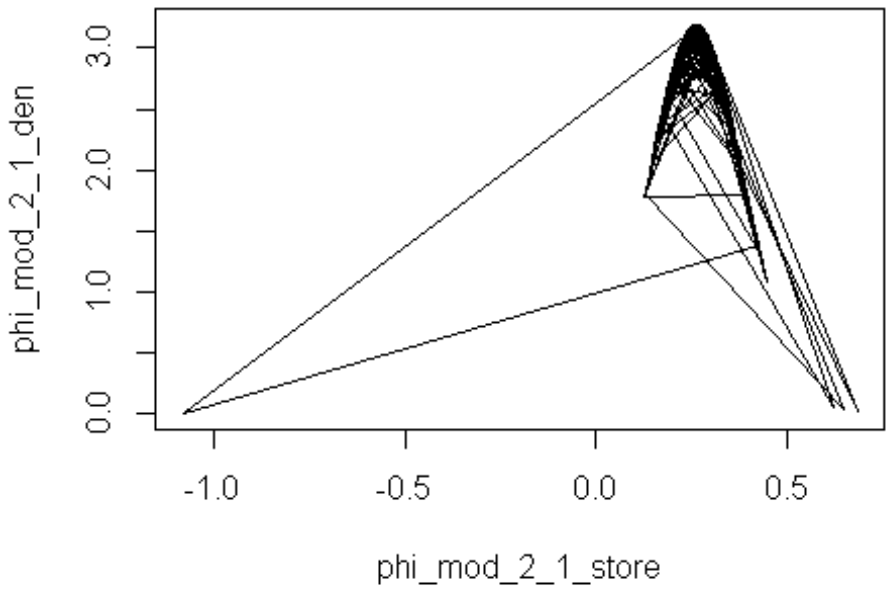



```
phi_mod_1_1_mean=mean(phi_mod_1_1_store)
phi_mod_1_1_var=var(phi_mod_1_1_store)
phi_mod_1_1_den=dnorm(phi_mod_1_1_store,mean=phi_mod_1_1_mean,sd=sqrt(phi_mod_1_1_v
ar))
plot(phi_mod_1_1_store,phi_mod_1_1_den,type='l')
```

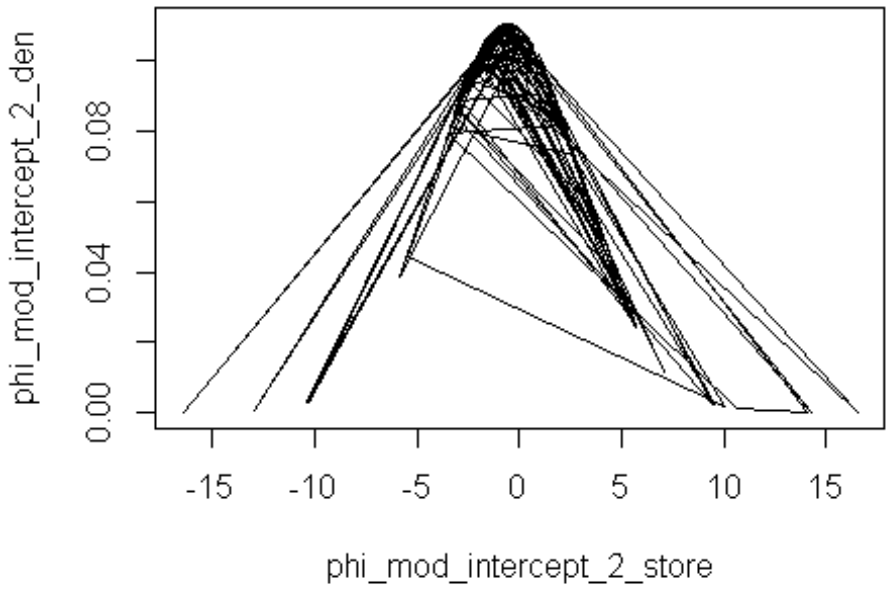



```
phi_mod_2_1_mean=mean(phi_mod_2_1_store)
phi_mod_2_1_var=var(phi_mod_2_1_store)
```

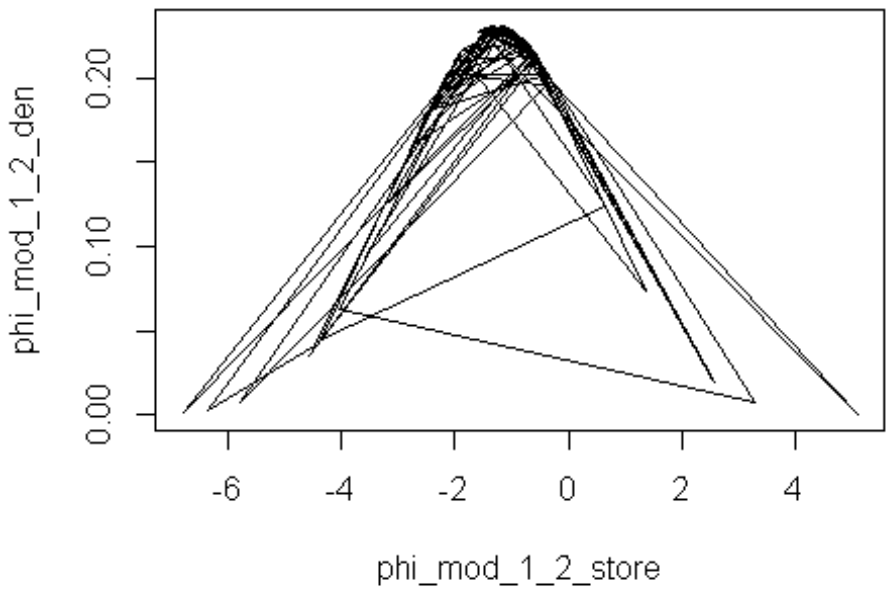
```
phi_mod_2_1_den=dnorm(phi_mod_2_1_store,mean=phi_mod_2_1_mean,sd=sqrt(phi_mod_2_1_v
ar))
plot(phi_mod_2_1_store,phi_mod_2_1_den,type='l')
```




```
phi_mod_intercept_2_mean=mean(phi_mod_intercept_2_store)
phi_mod_intercept_2_var=var(phi_mod_intercept_2_store)
phi_mod_intercept_2_den=dnorm(phi_mod_intercept_2_store,mean=phi_mod_intercept_2_mean,sd
=sqrt(phi_mod_intercept_2_var))
plot(phi_mod_intercept_2_store,phi_mod_intercept_2_den,type='l')
```




```
phi_mod_1_2_mean=mean(phi_mod_1_2_store)
phi_mod_1_2_var=var(phi_mod_1_2_store)
phi_mod_1_2_den=dnorm(phi_mod_1_2_store,mean=phi_mod_1_2_mean,sd=sqrt(phi_mod_1_2_v
ar))
plot(phi_mod_1_2_store,phi_mod_1_2_den,type='l')
```



Variable	From tar routine	Mean	Variance
Threshold	175	156.66	1290.74
Intercept^1	-0.165	-0.419	0.3845
$-d_1{}^1$	-0.0248	-0.107	0.0262
$-d_2{}^1$	0.422	0.272	0.0042
Intercept^2	0.366	-0.13	6.9465
$-d_1{}^2$	-1.88	-1.74	1.5018

The delay is 1.

This model satisfies since the residuals are white based on looking at the ACF and PACF and by running Box.test on the residuals.

2.R

```
library(TSA)
```

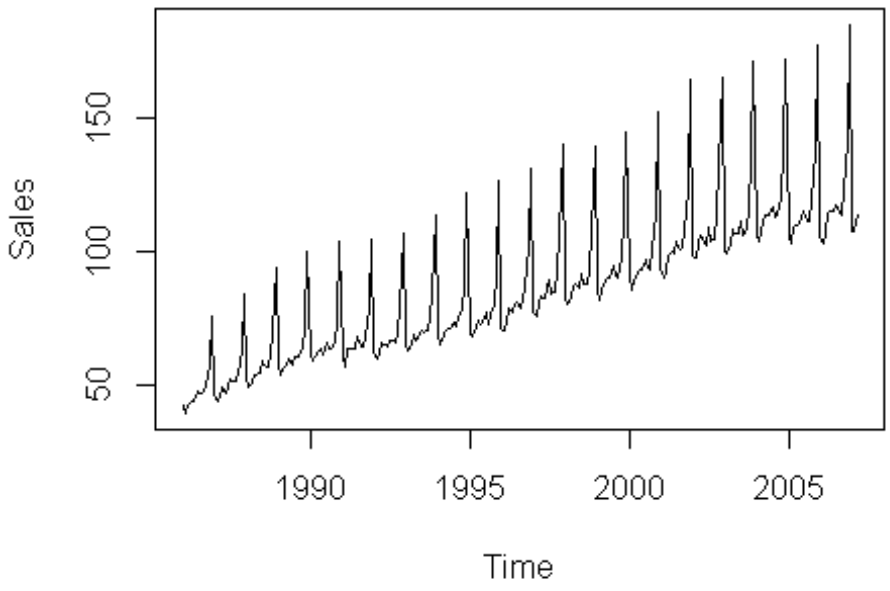
```
## Loading required package: leaps
## Loading required package: locfit
## locfit 1.5-9.1    2013-03-22
## Loading required package: mgcv
## Loading required package: nlme
## This is mgcv 1.8-0. For overview type 'help("mgcv-package")'.
## Loading required package: tseries
##
## Attaching package: 'TSA'
##
## The following objects are masked from 'package:stats':
##
##     acf, arima
##
## The following object is masked from 'package:utils':
##
##     tar
```

```
library(astsa)
library(forecast)
```

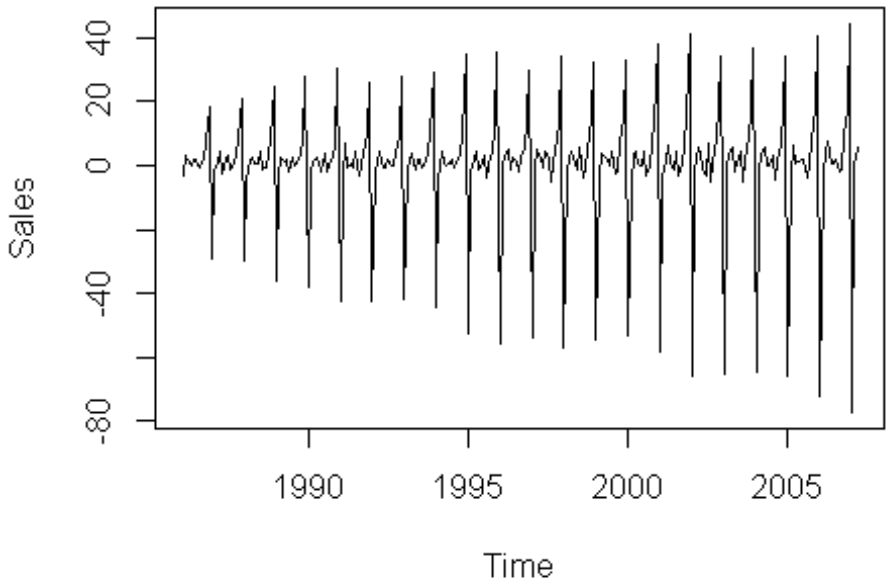
```
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
## Loading required package: timeDate
##
## Attaching package: 'timeDate'
##
## The following objects are masked from 'package:TSA':
##
##     kurtosis, skewness
##
## This is forecast 5.6
##
##
## Attaching package: 'forecast'
##
## The following object is masked from 'package:astsa':
##
##     gas
##
## The following object is masked from 'package:TSA':
##
```

```
##     fitted.Arima
##
## The following object is masked from 'package:nlme':
##
##     getResponse
```

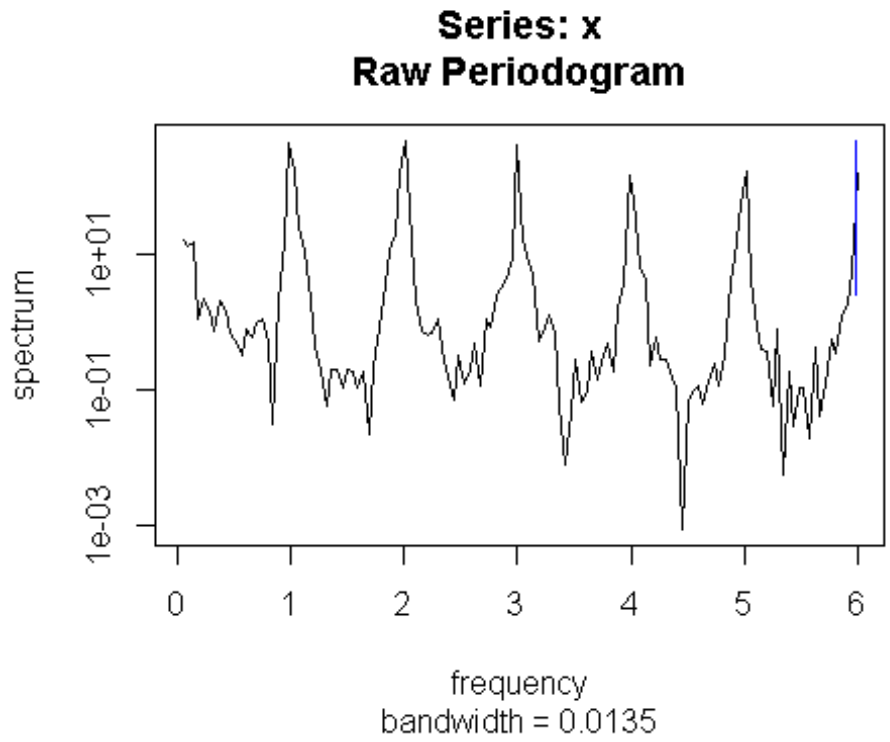
```
data(retail)
#retail_new=retail[1:255]
plot(retail,type='l')
```



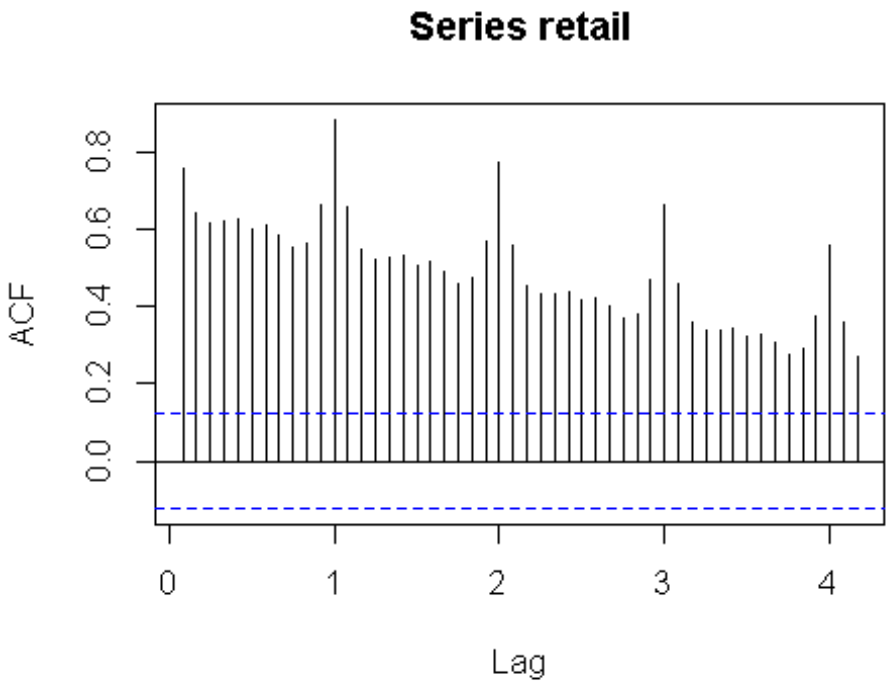

```
retail_diff=diff(retail)
plot(retail_diff)# d=1, D=1
```



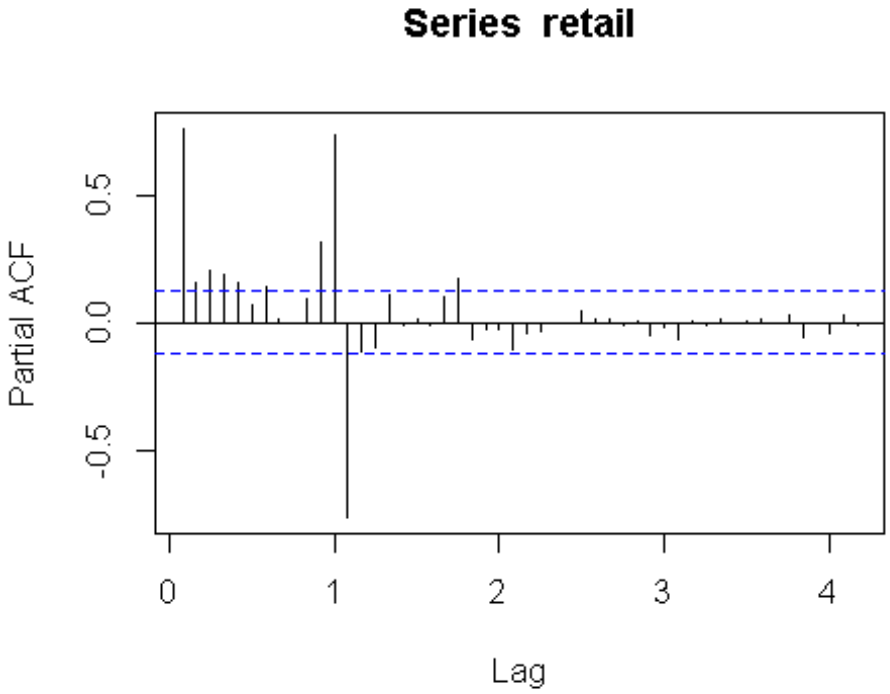

```
spectrum(retail)# S=12
```




```
retail_acf=acf(retail,lag.max=50)#q=2,Q=1
```



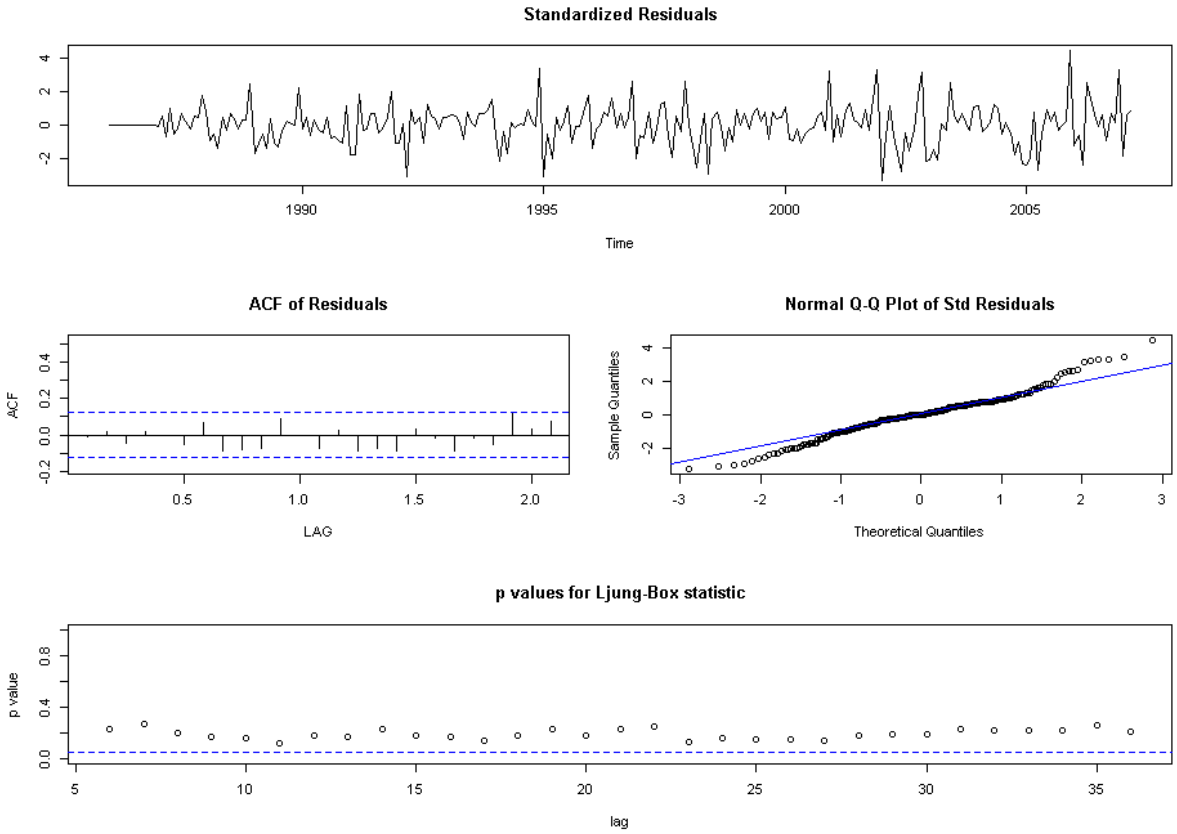

```
retail_pacf=pacf(retail,lag.max=50)#p=2,P=0
```



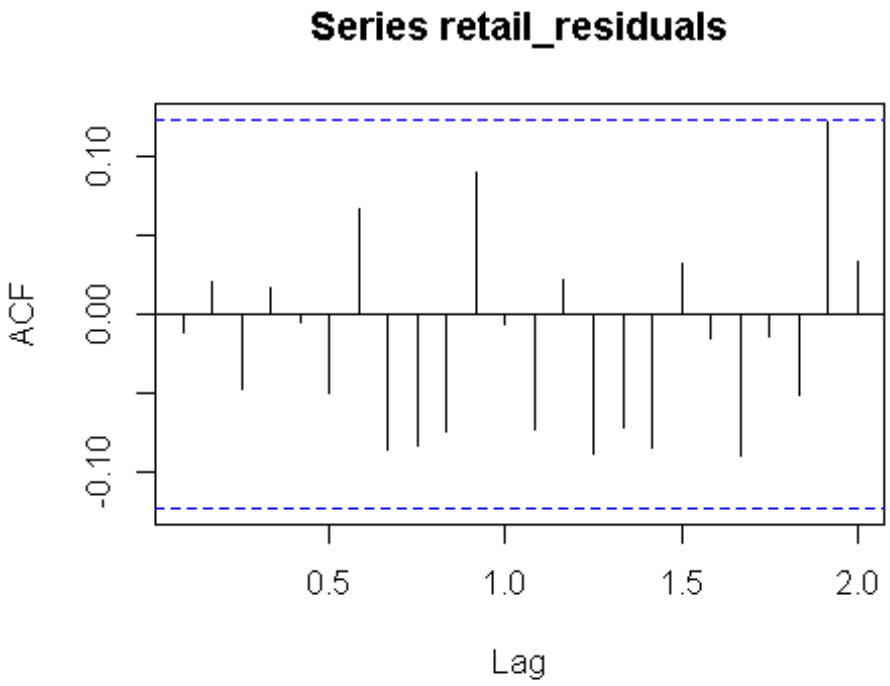

```
retail_test=sarima(retail,2,1,2,D=1,Q=1,S=12)
```

```
## initial  value 0.840627
## iter   2 value 0.685920
## iter   3 value 0.645495
## iter   4 value 0.645147
```

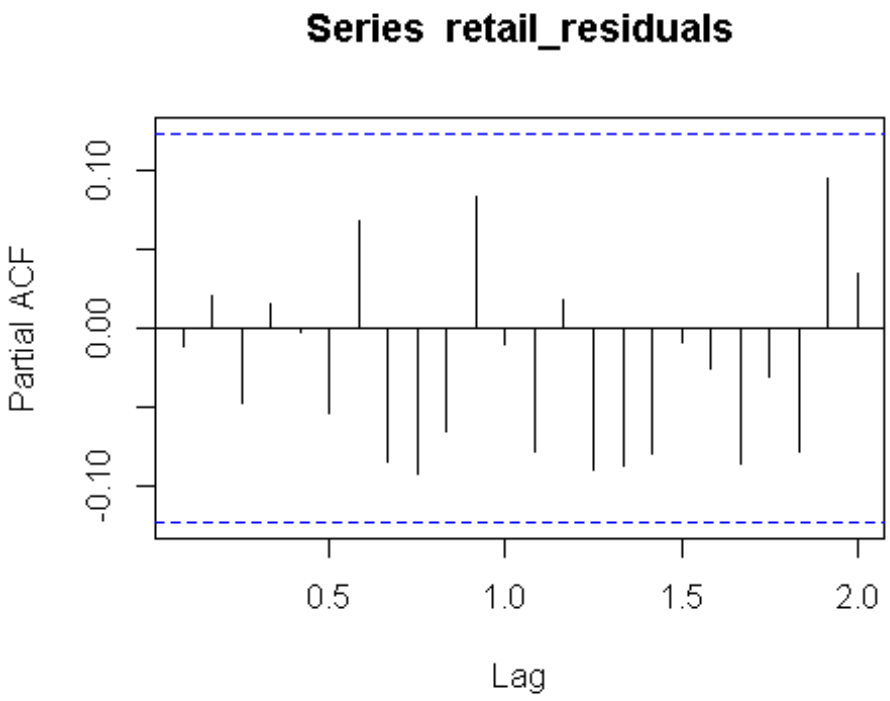
```
## iter   5 value 0.644846
## iter   6 value 0.644707
## iter   7 value 0.644628
## iter   8 value 0.644592
## iter   9 value 0.644499
## iter  10 value 0.644275
## iter  11 value 0.643740
## iter  12 value 0.642994
## iter  13 value 0.642226
## iter  14 value 0.641474
## iter  15 value 0.641163
## iter  16 value 0.641076
## iter  17 value 0.641016
## iter  18 value 0.640740
## iter  19 value 0.640226
## iter  20 value 0.639682
## iter  21 value 0.639121
## iter  22 value 0.638806
## iter  23 value 0.638802
## iter  24 value 0.638802
## iter  25 value 0.638802
## iter  26 value 0.638802
## iter  27 value 0.638802
## iter  28 value 0.638802
## iter  29 value 0.638802
## iter  30 value 0.638802
## iter  31 value 0.638801
## iter  32 value 0.638801
## iter  32 value 0.638801
## iter  32 value 0.638801
## final  value 0.638801
## converged
## initial  value 0.632797
## iter   2 value 0.632772
## iter   3 value 0.632748
## iter   4 value 0.632689
## iter   5 value 0.632626
## iter   6 value 0.632520
## iter   7 value 0.632494
## iter   8 value 0.632480
## iter   9 value 0.632477
## iter  10 value 0.632475
## iter  11 value 0.632475
## iter  11 value 0.632475
## iter  11 value 0.632475
## final  value 0.632475
## converged
```




```
retail_residuals=retail_test$fit$residuals
acf(retail_residuals)
```




```
pacf(retail_residuals)
```



The Model is ARIMA(2,1,2) x $(0,1,1)_{12}$ as seen based on the values and plots above.

This model satisfies since the residuals are white based on looking at the ACF and PACF and by running Box.test on the residuals.