APPLIED TIME SERIES ANALYSIS

PROJECT REPORT

SUBMITTED BY

SHARA RHAGHA WARDHAN B

ME11B121

Honor Code

I have worked out the project in all honesty**,** i.e., without consulting any other person or referring to other resources for solutions, except the instructor for queries.

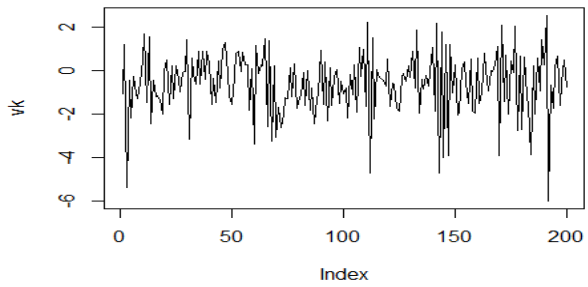
-Shara Rhagha Wardhan B

Project1.R

Shara

Mon Dec 01 19:47:42 2014

```
setwd("C:/Users/Shara/Desktop/Applied Time Series Analysis Project")
library("TSA", lib.loc="~/R/win-library/3.1")                    set.seed(1
)
rm(list=ls());
load('projq1a.Rdata')
plot(vk,type='l')
```



FIRST APPROACH OF DEVELOPING THE MODEL

We fix the delay as 1 since intuitively the least error will come when a sample has to depend on its immediate past rather than some instances in the past. Now we split the data into a test set and a training set. We use the training set to develop a model for the specified highest order model using the Minimize AIC method and find the MSE of how they work well in prediction of the test set. Now we find the models over a range of specified high order models and then we choose the model with the minimum MSE based on the prediction capabilities.

```
# Fixing the delay
delay = 1;
# Finding the optimal model
N = length(vk);
N_tr = ceiling(0.7*N);
N_ts = N - N_tr;
V_tr<-c(vk[1:N_tr]);
V_ts<-c(vk[(N_tr+1):N])
mse_ts<-c();
for (ord in 1:40)
{
  mod<-tar(V_tr,ord,ord,delay);
  vk_pred<-predict.TAR(mod,n.ahead=N_ts,n.sim=1000);
  vk_ts_pred<-vk_pred$fit;
```

```
  mse_<-mean((V_ts - vk_ts_pred)^2);
  mse_ts<-c(mse_ts,mse_);
}

odr_opt<-which (mse_ts==min(mse_ts))
mod<-tar(V_tr,odr_opt,odr_opt,delay,print=T)

## time series included in this analysis is:  V_tr
## SETAR(2, 2 , 2 ) model delay = 1
## estimated threshold =  -0.3648  from a Minimum AIC  fit with thresholds
## searched from the  49  percentile to the   50  percentile of all data.
## The estimated threshold is the  50.4  percentile of
## all data.
## lower regime:
## Residual Standard Error=0.8524
## R-Square=0.4379
## F-statistic (df=3, 54)=14.0233
## p-value=0
##
##                Estimate Std.Err t-value Pr(>|t|)
## intercept-V_tr   0.1234  0.2531  0.4878   0.6276
## lag1-V_tr        0.1349  0.1412  0.9553   0.3437
## lag2-V_tr        0.6197  0.1061  5.8434   0.0000
##
##
##
##
##  (unbiased) RMS
## 0.7266
##  with no of data falling in the regime being
## V_tr 57
##
##
##  (max. likelihood) RMS for each series (denominator=sample size in the reg
ime)
## V_tr 0.6884
##
##
##  upper regime:
## Residual Standard Error=1.1693
## R-Square=0.4547
## F-statistic (df=3, 53)=14.729
## p-value=0
##
##                Estimate Std.Err t-value Pr(>|t|)
## intercept-V_tr  -0.1963  0.2057 -0.9546   0.3441
## lag1-V_tr       -0.8015  0.2581 -3.1048   0.0031
## lag2-V_tr        0.3089  0.1282  2.4088   0.0195
##
##
```

```
##
##
##  (unbiased) RMS
## 1.367
##  with no of data falling in the regime being
## 56
##
##
##  (max. likelihood) RMS for each series (denominator=sample size in the reg
ime)
## 1.294
##
##  Nominal AIC is  327.8
```

```
# Printing the results of the optimal model
```

```
print(mod$n1)
```

```
## [1] 57
```

```
print(mod$n2)
```

```
## [1] 56
```

```
print(mod$p1)
```

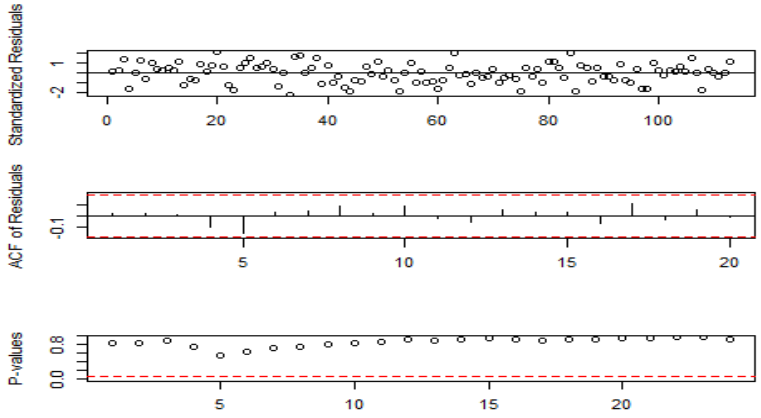
```
## [1] 2
```

```
print(mod$p2)
```

```
## [1] 2
```

The fit optimal order is AR(2) and AR(2) for the 1st and 2nd regimes respectively and the space of samples is equally distributed between the two regimes. But the standard error clearly shows that the estimates aren't good. We show the results of this model. And also this model satisfies the Whiteness test.

```
tsdiag(mod)
```



The model is optimal with respect to whiteness test, principle of parcemony and predictive capabilities but the model has issues with the parameter estimates. We build another model later without looking into the prediction capability.

```
#Generate the residuals from the best model
ep_k<-mod$residuals;
L<-length(ep_k);
offset = max(mod$p1,mod$p2)
V_tr_req<-c(V_tr[(1+offset):(L+offset)])
V_hat<-V_tr_req-ep_k;
```

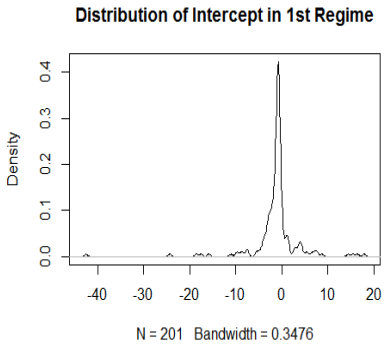
Now we try the robustness analysis by Bootstrapping method

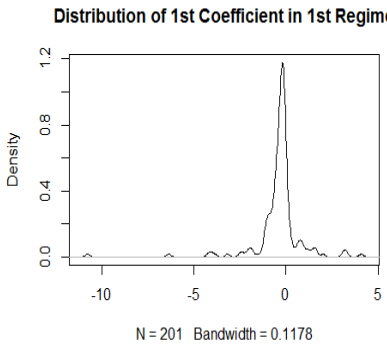
```
Coef1<-as.vector(c(mod$qr1$coefficients,mod$qr2$coefficients));

R = 200;
Coef_R=matrix(nrow=(R+1),ncol=length(Coef1))
Coef_R[1,]=as.vector(Coef1);
for(j in 1:R)
{
  epskr <- sample(ep_k,size=L,replace=T);
  V <- V_hat + epskr;
  modr<-tar(V,mod$p1[1],mod$p2[1],mod$d[1],order.select=FALSE);
  Coef_r<-as.vector(c(modr$qr1$coefficients,modr$qr2$coefficients));
  Coef_R[(1+j),]=as.vector(Coef_r);
}

Coef=matrix(nrow=(R+1),ncol=1);
m_C=c();
v_C=c();
for (l  in 1:length(Coef1))
{
  Coef <- Coef_R[,l];
  m_C<-c(m_C,mean(Coef));
  v_C<-c(v_C,var(Coef))
}
```

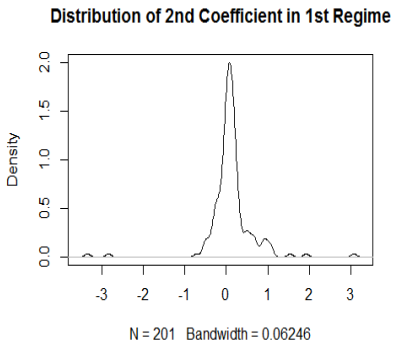
Results are presented

```
# Coefficient 1
C1 <- Coef_R[,1];
m_C1=mean(C1)
sd_C1 = (var(C1))^0.5;
cbind(c('','Mean','Standard
Deviation'),c('Actual',Coef[1],se_1[1]),c('Estimate',m_C1,sd_C1));
```

```
##                 [,1]                [,2]                [,3]
##                 ""                  "Actual"            "Estimate"
##                 "Mean"              "0.308905618466357" "-1.1545269858231"
## intercept-V_tr "Standard Deviation" "0.0640362543247204""5.33034317590942"


plot(density(C1),main='Distribution of Intercept in 1st Regime');
```








```
# Coefficient 2
C2 <- Coef_R[,2];
m_C2=mean(C2)
sd_C2 = (var(C2))^0.5;
cbind(c('','Mean','Standard Deviation'),c('Actual',Coef[2],se_1[2]),c('Estima
te',m_C2,sd_C2));
```

```
##           [,1]                [,2]                [,3]
##           ""                  "Actual"            "Estimate"
##           "Mean"              "0.28736935150986"  "-0.355465756908357"
## lag1-V_tr "Standard Deviation" "0.0199256028139817" "1.28632921827308"
```

```
plot(density(C2),main='Distribution of 1st Coefficient in 1st Regime');
```

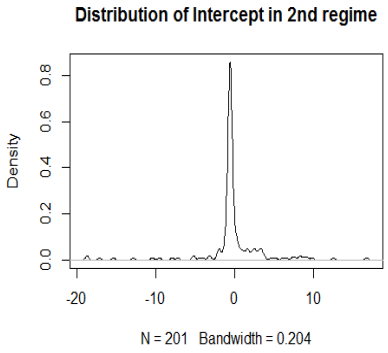
```
# Coefficient 3
C3 <- Coef_R[,3];
m_C3=mean(C3)
sd_C3 = (var(C3))^0.5;
cbind(c('','Mean','Standard Deviation'),c('Actual',Coef[3],se_1[3]),c('Estima
te',m_C3,sd_C3));
```

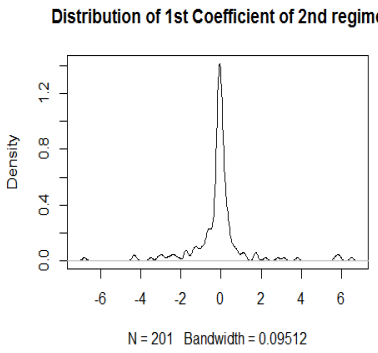
```
##           [,1]                [,2]                [,3]
##           ""                  "Actual"            "Estimate"
##           "Mean"              "0.220088421750601"  "0.116869604046567"
## lag2-V_tr "Standard Deviation" "0.0112466330027001" "0.521396653078834"
```

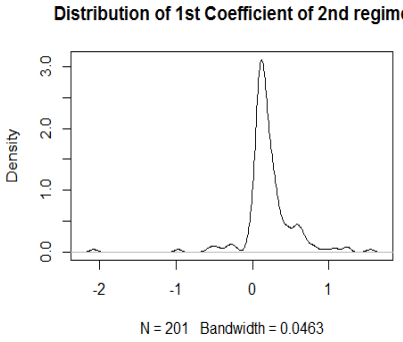
```
plot(density(C3),main='Distribution of 2nd Coefficient in 1st Regime');
```

```
# Coefficient 4
C4 <- Coef_R[,4];
m_C4=mean(C4)
sd_C4 = (var(C4))^0.5;
cbind(c('','Mean','Standard Deviation'),c('Actual',Coef[4],se_2[1]),c('Estima
te',m_C4,sd_C4));
```

```
##               [,1]                [,2]                [,3]
##               ""                  "Actual"            "Estimate"
##               "Mean"              "0.087498444135938" "-0.2988057865842"
## intercept-V_tr "Standard Deviation" "0.0423070085479862""3.95293131060593"
plot(density(C4),main='Distribution of Intercept in 2nd regime');
```








```
# Coefficient 5
C5 <- Coef_R[,5];
m_C5=mean(C5)
sd_C5 = (var(C5))^0.5;
cbind(c('','Mean','Standard Deviation'),c('Actual',Coef[5],se_2[2]),c('Estima
te',m_C5,sd_C5));
```

```
##           [,1]                [,2]                [,3]
##           ""                  "Actual"            "Estimate"
##           "Mean"              "0.314240423690341"  "-0.116209468297676"
## lag1-V_tr "Standard Deviation" "0.0666373737653331" "1.39304814797677"
```

```
plot(density(C5),main='Distribution of 1st Coefficient of 2nd regime');
```

```
# Coefficient 6
C6 <- Coef_R[,6];
m_C6=mean(C6)
sd_C6 = (var(C6))^0.5;
cbind(c('','Mean','Standard Deviation'),c('Actual',Coef[6],se_2[3]),c('Estima
te',m_C6,sd_C6));
```

```
##           [,1]                [,2]                [,3]
##           ""                  "Actual"            "Estimate"
##           "Mean"              "0.570650489940356"  "0.204415488367631"
## lag2-V_tr "Standard Deviation" "0.0164449510971871" "0.327224820161364"
```

```
plot(density(C6),main='Distribution of 1st Coefficient of 2nd regime');
```

The estimates have come out to be very poor. One possible reason could be because of the initial estimates in parameters were erroneous. So when we generate different realizations the error just propagated.

Here we fit the model by assuming the highest possible order and fitting a delay iteratively and the model with the minimum AIC is chosen as optimal.

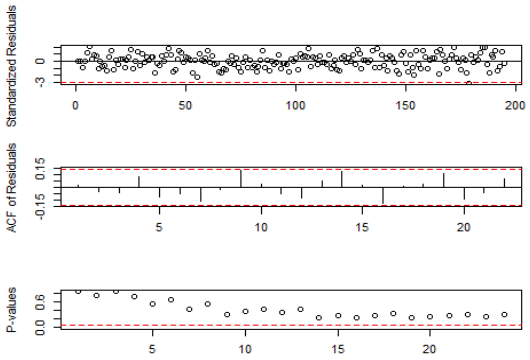
```
aic_no=c();
for (d in 1:5)
{
  mod<-tar(vk,p1=5,p2=5,d=d);
```

```
  aic_no = c(aic_no,mod$AIC);
}
odr_del<-which (aic_no==min(aic_no))
mod<-tar(vk,5,5,odr_del,print=T)
```

```
## time series included in this analysis is:  vk
## SETAR(2, 2 , 3 ) model delay = 1
## estimated threshold =  0.9466  from a Minimum AIC  fit with thresholds
## searched from the  6  percentile to the   94  percentile of all data.
## The estimated threshold is the  88.7  percentile of
## all data.
## lower regime:
## Residual Standard Error=1.044
## R-Square=0.2932
## F-statistic (df=3, 170)=23.5116
## p-value=0
##
##              Estimate Std.Err t-value Pr(>|t|)
## intercept-vk  -0.1713  0.1105 -1.5498   0.1231
## lag1-vk       -0.0385  0.0696 -0.5536   0.5806
## lag2-vk        0.4264  0.0669  6.3768   0.0000
##
##
##
##
##  (unbiased) RMS
## 1.09
##  with no of data falling in the regime being
## vk 173
##
##
##  (max. likelihood) RMS for each series (denominator=sample size in the reg
ime)
## vk 1.071
##
##
##  upper regime:
## Residual Standard Error=1.3048
## R-Square=0.8357
## F-statistic (df=4, 18)=22.8907
## p-value=0
##
##              Estimate Std.Err t-value Pr(>|t|)
## intercept-vk   0.1048  1.1541  0.0908   0.9287
## lag1-vk       -1.4423  0.7305 -1.9744   0.0639
## lag2-vk       -0.3887  0.2355 -1.6501   0.1163
## lag3-vk       -1.1385  0.5161 -2.2061   0.0406
##
##
##
```

```
##
##  (unbiased) RMS
## 1.702
##  with no of data falling in the regime being
## 22
##
##
##  (max. likelihood) RMS for each series (denominator=sample size in the reg
ime)
## 1.393
##
##  Nominal AIC is  588.6
```

```
tsdiag(mod)
```



The model passes the white noise tests. The estimates are at the border of the confidence interval but has the minimum AIC value.

```
#Generate the residuals from the best model
ep_k<-mod$residuals;
L<-length(ep_k);
offset = max(mod$p1,mod$p2)
V_tr_req<-c(vk[(1+offset):(L+offset)])
V_hat<-V_tr_req-ep_k;

Coef1<-as.vector(c(mod$qr1$coefficients,mod$qr2$coefficients));

R = 200;
Coef_R=matrix(nrow=(R+1),ncol=length(Coef1))
Coef_R[1,]=as.vector(Coef1);
for(j in 1:R)
{
  epskr <- sample(ep_k,size=L,replace=T);
  V <- V_hat + epskr;
  modr<-tar(V,mod$p1[1],mod$p2[1],mod$d[1],order.select=FALSE);
  Coef_r<-as.vector(c(modr$qr1$coefficients,modr$qr2$coefficients));
  Coef_R[(1+j),]=as.vector(Coef_r);
```

```
}

Coef=matrix(nrow=(R+1),ncol=1);
 m_C=c();
 v_C=c();
 sd_C=c();
for (l  in 1:length(Coef1))
{
  Coef <- Coef_R[,l];
#   m_C<-mean(Coef);
#   v_C=var(Coef);
#   sd_C=v_C^0.5;
m_C<-c(m_C,mean(Coef));
v_C<-c(v_C,var(Coef));
sd_C<-c(sd_C,v_C^0.5);
print('Coefficient parameters')
print(l)
print('mean')
print(m_C[l])
print('std.er')
print(sd_C[l])
}
```

```
## [1] "Coefficient parameters"
## [1] 1
## [1] "mean"
## [1] -1.051015        (ACTUAL -0.1713  )
## [1] "std.er"
## [1] 2.560391
## [1] "Coefficient parameters"
## [1] 2
## [1] "mean"
## [1] -0.289159         (ACTUAL -0.0385  )
## [1] "std.er"
## [1] 2.560391
## [1] "Coefficient parameters"
## [1] 3
## [1] "mean"
## [1] -0.1450375            (ACTUAL 0.4264  )
## [1] "std.er"
## [1] 0.4979975
## [1] "Coefficient parameters"
## [1] 4
## [1] "mean"
## [1] -0.3953807                  (ACTUAL 0.1048  )
## [1] "std.er"
## [1] 2.560391
## [1] "Coefficient parameters"
## [1] 5
## [1] "mean"
```

```
## [1] -0.5193459                    (ACTUAL -1.4423  )
## [1] "std.er"
## [1] 0.4979975
## [1] "Coefficient parameters"
## [1] 6
## [1] "mean"
## [1] -0.3106849              (ACTUAL -0.3887  )
## [1] "std.er"
## [1] 0.2040298
## [1] "Coefficient parameters"
## [1] 7
## [1] "mean"
## [1] -0.2386773                  (ACTUAL -1.1385  )
## [1] "std.er"
## [1] 2.560391
```

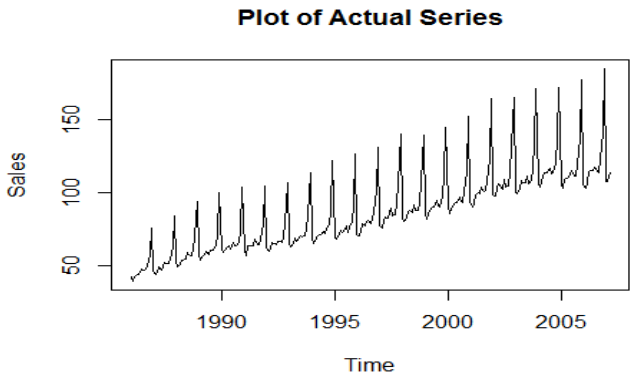
It is observed that the estimates are good where the initial standard errors are proper in the sense that the estimate lies in the confidence interval. Fo r others, the mean is closer but still is not in acceptable regime.

Project2.R

Shara

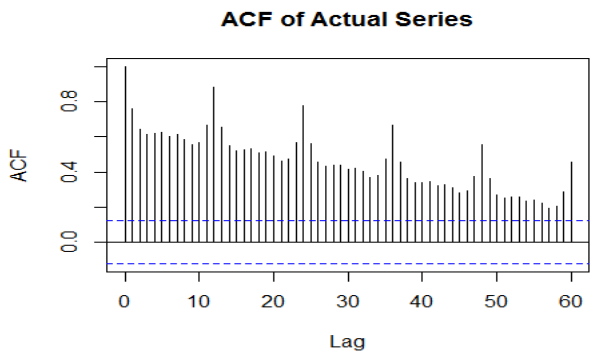
Mon Dec 01 12:05:05 2014

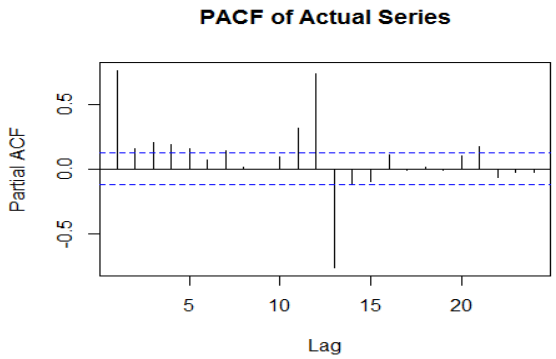
```
library("TSA", lib.loc="~/R/win-library/3.1")
data(retail)
plot(retail,main='Plot of Actual Series')
```



From the visual observation of the series it is clearly seen that there is non-stationarity of trend type combined with seasonal type. But it is not clear whether the trend is linear or non-linear. Our primary objective is to convert the non-stationary into a stationary process.

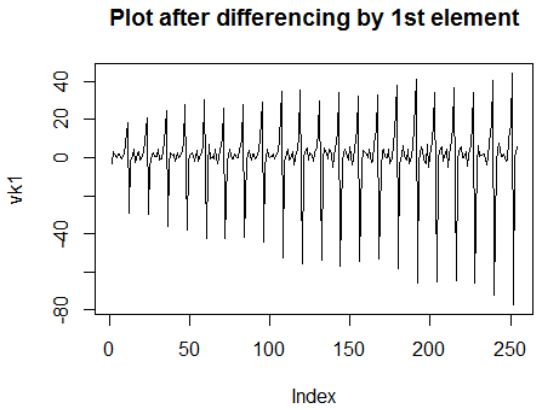
```
# ANALYSIS OF THE SERIES
vk = as.vector(retail);
acf(vk,lag.max=60,drop.lag.0=FALSE,main='ACF of Actual Series')
pacf(vk,,main='PACF of Actual Series')
```





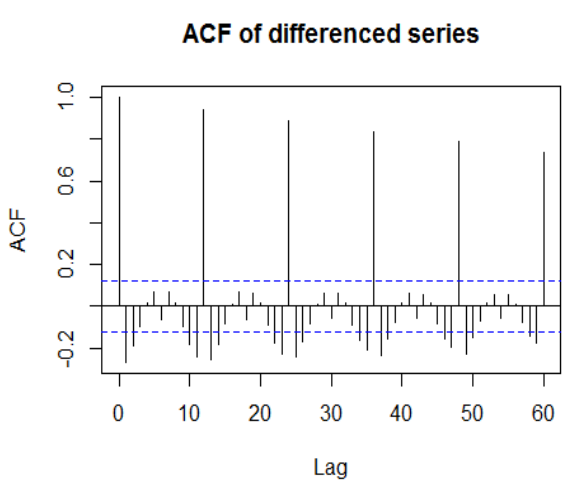
ACF shows clearly that there is a seasonal component with a period of 12. The PACF plot is an interesting one. It shows very strong linear dependence with the immediate sample and the trend repeats after the period of 12 (i.e. the 13th sample). Hence I first difference with the immediate sample.

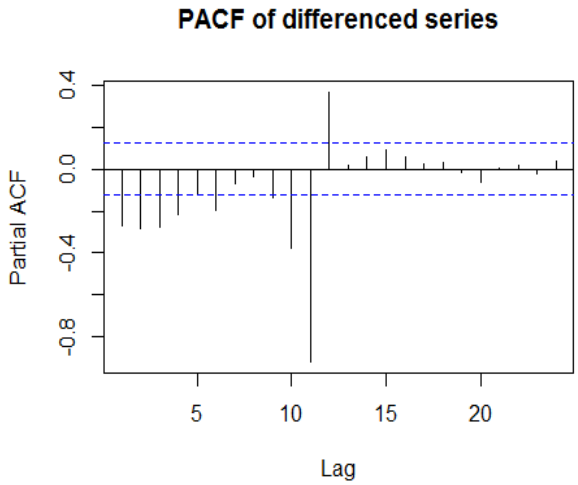
```
# WE DIFFERENCE BY THE 1st SAMPLE
vk1 <- diff(vk,1)
plot(vk1,type='l',main='Plot after differencing by 1st element')
```



From visual observation, we can conclude that the trend part is removed. But it is still a heteroskedastic process as the variance is a function of time.

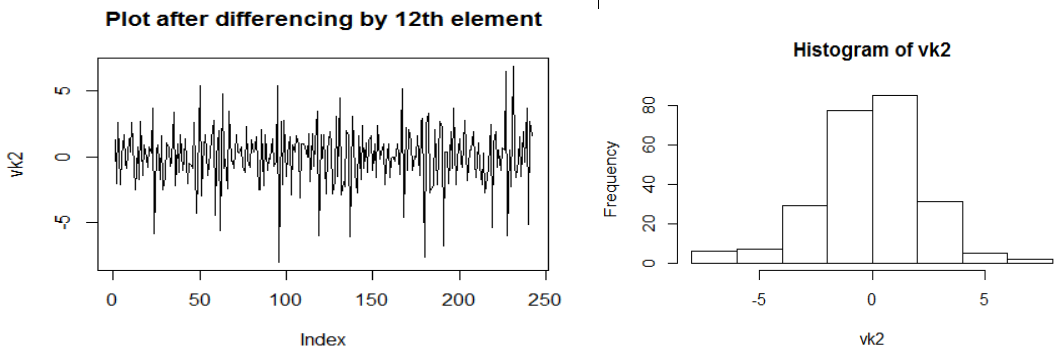
```
acf(vk1,lag.max=60,drop.lag.0=FALSE,main='ACF of differenced series')
pacf(vk1,main='PACF of differenced series')
```





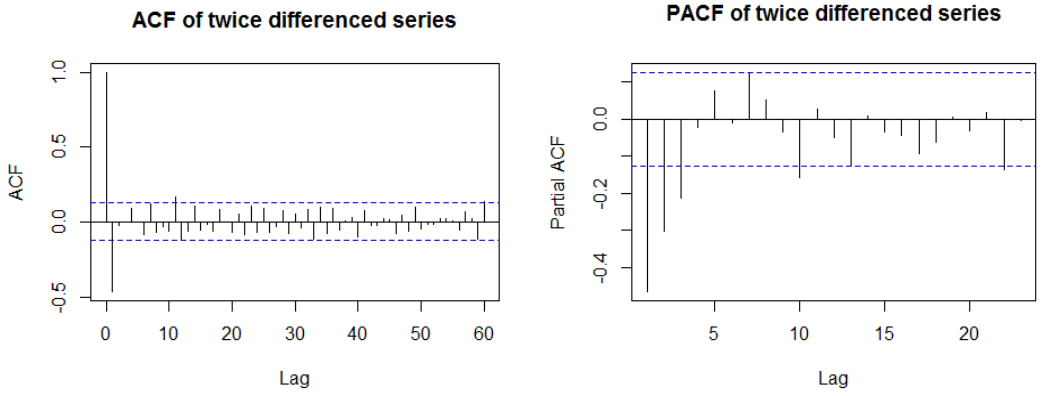
Now it is very clearly seen that the 12th sample period is dominant. Hence we go ahead and remove it by differencing the already differenced series by the 12th sample.

```r
# WE DIFFERENCE BY THE 12th SAMPLE
vk2 <- diff(vk1,12)
plot(vk2,type='l',main='Plot after differencing by 12th element')
hist(vk2)
```



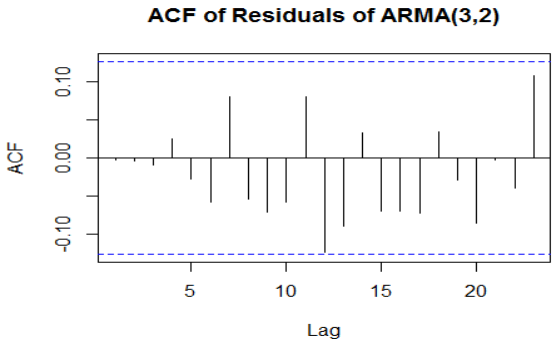
This is the beautiful awaited result. It looks mean stationary. From the histogram plot, it can be seen that the series is Gaussian.

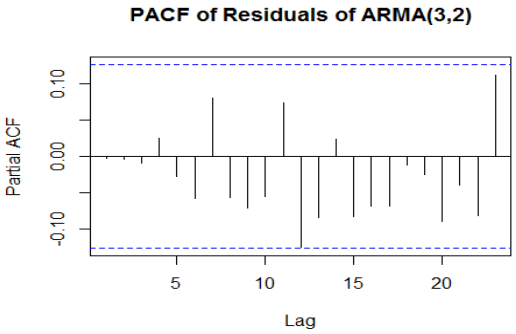
```r
acf(vk2,lag.max=60,drop.lag.0=FALSE,main='ACF of twice differenced series')
pacf(vk2,main='PACF of twice differenced series')
```



Both ACFs and PACFs die down with time suggesting Stationarity again. From the two plots, now we can be sure that we can start fitting an ARMA model. The ACf plot suggests that the MA order is 2. The PACF model suggests that the AR order is 3. So we go ahead and fit ARMA(3,2) model to the stationary remains of the series.

```r
# FITTING THE MODEL
#1 ARIMA(3,0,2) BASED ON ACF AND PACF
mod_arma32<-arima(vk2,order=c(3,0,2),include.mean=FALSE)
acf(mod_arma32$residuals,main='ACF of Residuals of ARMA(3,2)')
pacf(mod_arma32$residuals,main='PACF of Residuals of ARMA(3,2)')
```





The residuals are white and hence it passes the whiteness test. New we go ahead and look at the significance of the estimates.

```r
print(mod_arma32)
```

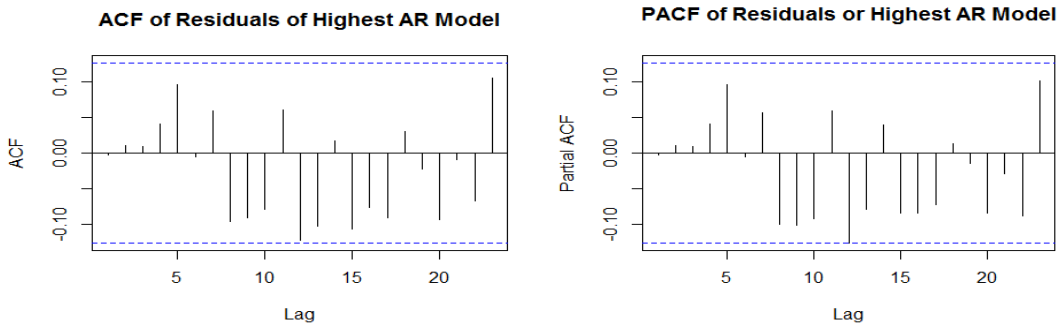
```
##
## Call:
## arima(x = vk2, order = c(3, 0, 2), include.mean = FALSE)
##
## Coefficients:
##          ar1      ar2      ar3      ma1     ma2
##       0.5029  -0.2293  -0.0750  -1.1916  0.6085
## s.e.  0.2033   0.1060   0.1012   0.1919  0.1263
##
## sigma^2 estimated as 3.572:  log likelihood = -497.77,  aic = 1005.55
```

```r
confint(mod_arma32)
```

```
##          2.5 %     97.5 %
## ar1  0.1044166  0.9013211
## ar2 -0.4370824 -0.0215253
## ar3 -0.2733539  0.1233869
## ma1 -1.5676563 -0.8154898
## ma2  0.3610572  0.8560418
```

After constructing the confidence intervals, it can be seen that all the parameters are acceptable. But we try other models to see if the number of parameters to be estimated can be reduced. So we first fit a high order AR model.

```
#2 HIGHEST AR MODEL
mod_ar<- ar(vk2,order.max = 50,method='ols')
acf(na.omit(mod_ar$resid),main='ACF of Residuals of Highest AR Model')
pacf(na.omit(mod_ar$resid),main='PACF of Residuals or Highest AR Model')
```



The whiteness test is satisfied by the AR(3) model. Next we look at the significance of the estimates by assuming that it is Gaussian.

```
mod_ar$ar
```

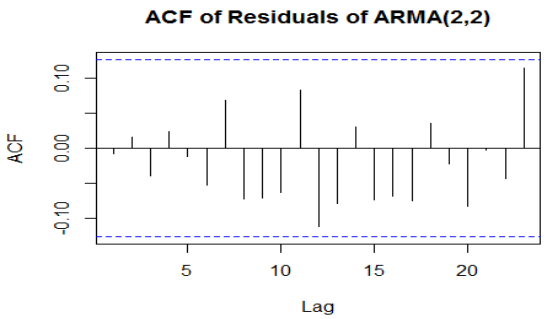
```
## , , 1
##
##             [,1]
## [1,] -0.6696089
## [2,] -0.4369002
## [3,] -0.2157548
```

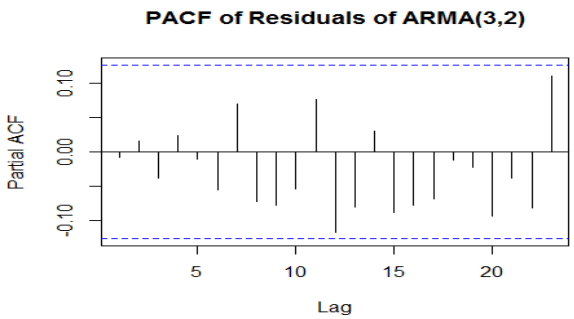
```
t(rep(mod_ar$ar,2)+cbind(1.96*mod_ar$asy.se.coef$ar,-1.96*mod_ar$asy.se.coef$
ar))
```

```
##             [,1]       [,2]        [,3]
## [1,] -0.5457831 -0.2967776 -0.09085167
## [2,] -0.7934347 -0.5770228 -0.34065789
```

The parameters lie in the confidence interval but the range is very less. Hence it may fail in prediction. Next we reduce the order of the ARMA model to see if we can fit a better model with lesser parameters.

```
#3 ARIMA(2,0,2) BASED ON ACF AND PACF
mod_arma22<-arima(vk2,order=c(2,0,2),include.mean=FALSE)
acf(mod_arma22$residuals,main='ACF of Residuals of ARMA(2,2)')
pacf(mod_arma22$residuals,main='PACF of Residuals of ARMA(3,2)')
```





Whiteness test is satisfied. We look at the significance of the parameters.

```
print(mod_arma22)
```

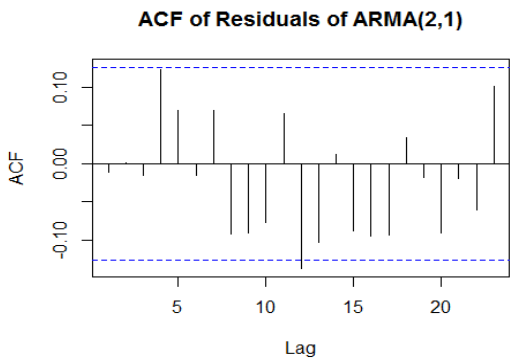
```
##
## Call:
## arima(x = vk2, order = c(2, 0, 2), include.mean = FALSE)
##
## Coefficients:
##          ar1      ar2      ma1     ma2
##       0.5276  -0.1981  -1.2107  0.5703
## s.e.  0.1705   0.1026   0.1582  0.1089
##
## sigma^2 estimated as 3.58:  log likelihood = -498.04,  aic = 1004.08
```

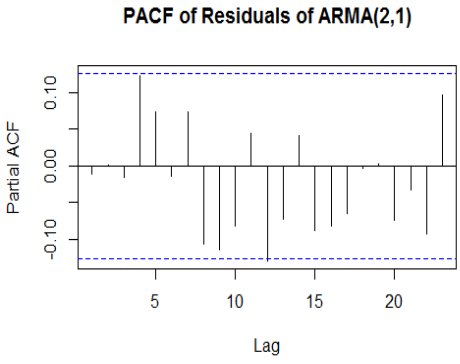
```
confint(mod_arma22)
```

```
##          2.5 %       97.5 %
## ar1  0.1933851  0.861796774
## ar2 -0.3991971  0.003022744
## ma1 -1.5207285 -0.900575421
## ma2  0.3568508  0.783790150
```

The parameters are within the 95% confidence limits. We go ahead and see if we can fit a lesser order.

```
#4 ARIMA(2,0,1) BASED ON ACF AND PACF
mod_arma21<-arima(vk2,order=c(2,0,1),include.mean=FALSE)
acf(mod_arma21$residuals,main='ACF of Residuals of ARMA(2,1)')
pacf(mod_arma21$residuals,main='ACF of Residuals of ARMA(2,1)')
```



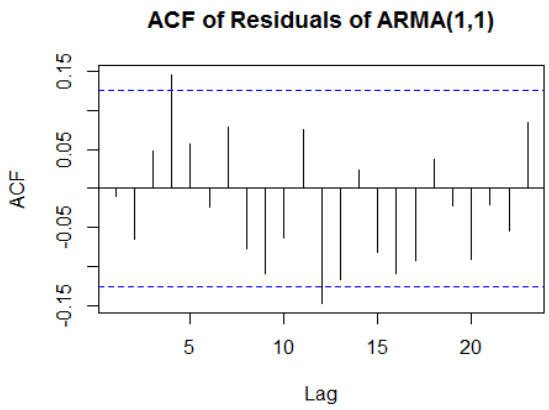


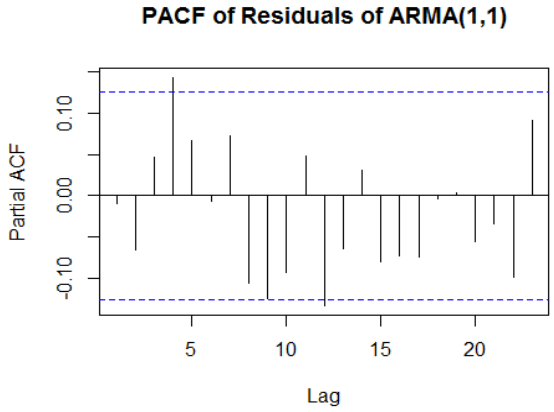
The whiteness test is satisfied. We look at the parameters.

```
print(mod_arma21)

##
## Call:
## arima(x = vk2, order = c(2, 0, 1), include.mean = FALSE)
##
## Coefficients:
##           ar1      ar2      ma1
##       -0.2615  -0.1590  -0.4047
## s.e.   0.1264   0.0895   0.1162
##
## sigma^2 estimated as 3.675:  log likelihood = -501.13,  aic = 1008.26

confint(mod_arma21)

##          2.5 %      97.5 %
## ar1 -0.5092397 -0.01382341
## ar2 -0.3343521  0.01641733
## ma1 -0.6323569 -0.17698652
```

The parameters are within the 95% confidence limits.

```
#5 ARIMA(1,0,1) BASED ON ACF AND PACF
mod_arma11<-arima(vk2,order=c(1,0,1),include.mean=FALSE)
acf(mod_arma11$residuals,main='ACF of Residuals of ARMA(1,1)')
pacf(mod_arma11$residuals,main='PACF of Residuals of ARMA(1,1)')
```





The whiteness test fails compared to the other models. Hence we don't look at the parameters and reject the model at this place itself.

A summary of the models to choose the optimal model:-

MODEL	NO OF PARAMETERS	AIC	SIGMA^2	LOG LIKELIHOOD
ARMA(3,2)	5	1005.55	3.572	-497.77
ARMA(2,2)	4	1004.08	3.58	-498.04
ARMA(2,1)	3	1008.26	3.675	-501.13

I choose ARMA(2,2) because the AIC and sigma^2 and log likelihood are comparable with the over fit ARMA(3,2) but the ARMA(2,1) is a little off. So by principle of parcemony, I choose ARMA(2,2) to fit the remaining series. After taking all the differencing into account, the final model is

$$v[k] - 1.5276.v[k-1] + 0.7257.v[k-2] - 0.1981.v[k-3] - v[k-12] + 1.5276.v[k-13] - 0.7257.v[k-14] + 0.1981.v[k-15] = e[k] - 1.2107e[k-1] + 0.5703.v[k-2]$$

_____END_OF_THE_COURSE_____