Timeseries

Manda Sai Prabha Reddy

01/12/2020

```
# Time Series
# 1.Generate a simulated two-class data set with 100 observations and two features in which there
# is a visible but non-linear seperation between the classes. Show that in this setting, a support
# vector machine with a polynomial kernal (with degree greater than 1) or a radial kernal will
# outperform a support vector classifier on the training data. Which technique performs best
# on the test data? Make plots and report training and test error rates in order to back up your
# assertions.
#a.TSD1
library('TSA')
```

```
## Warning: package 'TSA' was built under R version 3.6.3
```

```
##
## Attaching package: 'TSA'
```

```
## The following objects are masked from 'package:stats':
##
##     acf, arima
```

```
## The following object is masked from 'package:utils':
##
##     tar
```

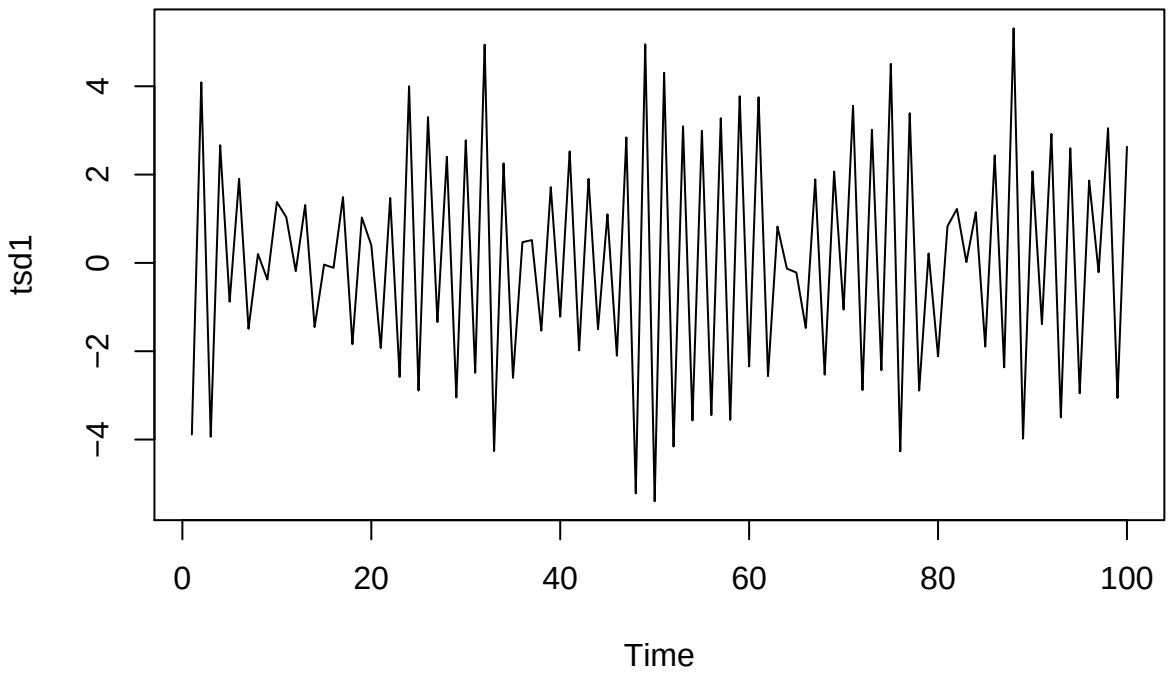
```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 3.6.3
```

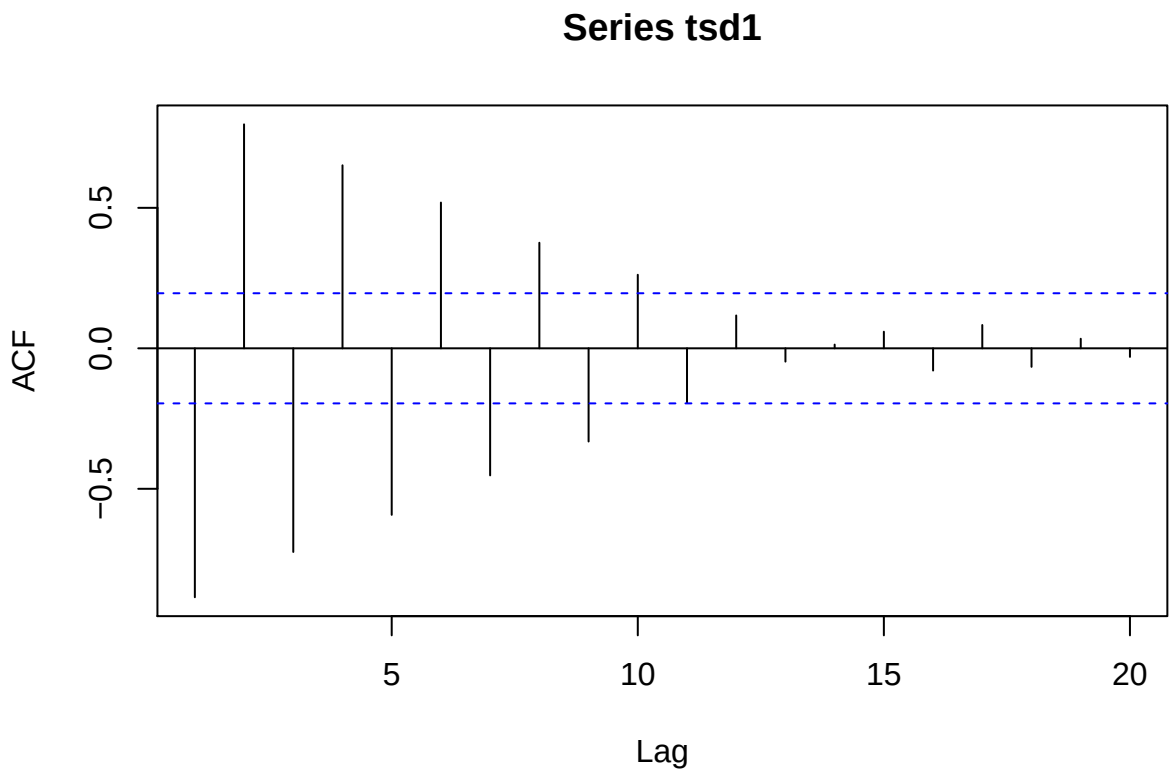
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## Registered S3 methods overwritten by 'forecast':
##   method       from
##   fitted.Arima TSA
##   plot.Arima   TSA
```

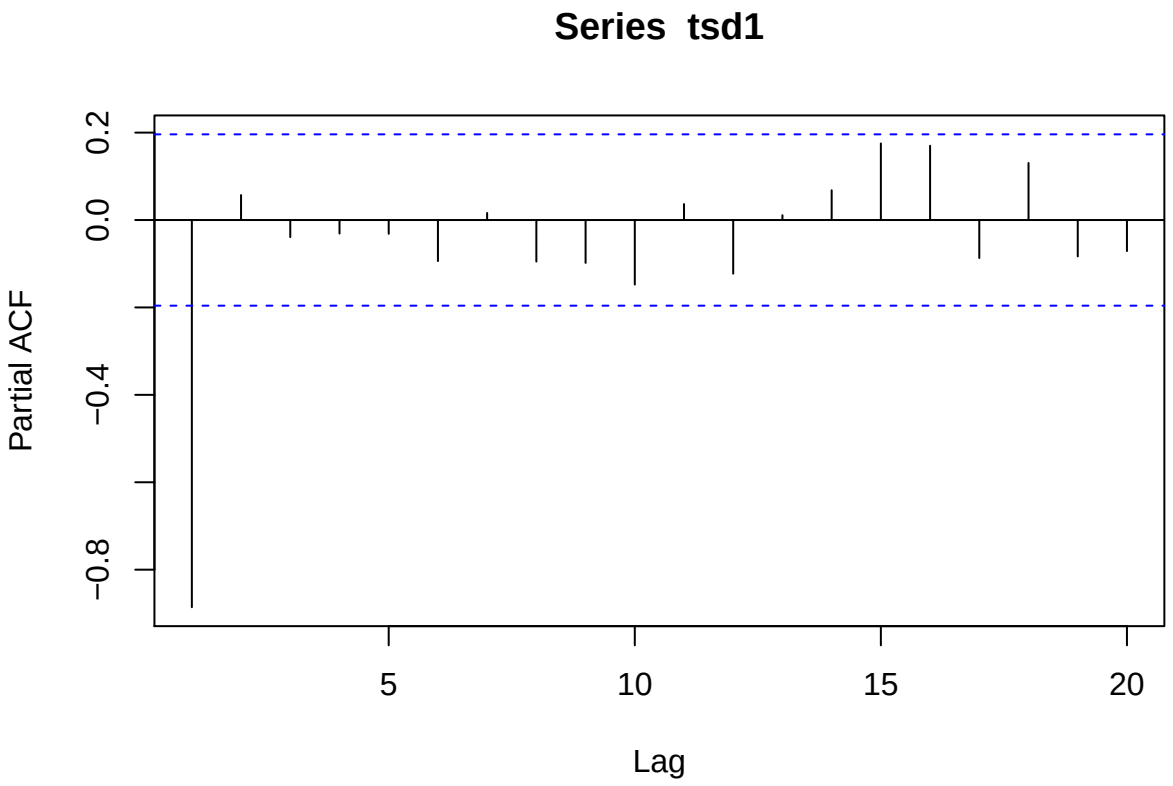
```
#TSD1
tsd1<-read.table("F:/TSD1.txt")
tsd1<-as.matrix(tsd1)
ts.plot(tsd1)
```




```
#to estimate p and q in arima(p,d,q) we require pacf,acf and eacf
#acf of tsd1
acf(tsd1)
```




```
pacf(tsd1)
```




```
eacf(tsd1)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x x x x x x o  o  o  o
## 1 o o o o o o o o o o o  o  o  o
## 2 x o o o o o o o o o o  o  o  o
## 3 x x o o o o o o o o o  o  o  o
## 4 o x o o o o o o o o o  o  o  o
## 5 o x o o o o o o o o o  o  o  o
## 6 o x o o o o o o o o o  o  o  o
## 7 o x x x o o o o o o o  o  o  o
```

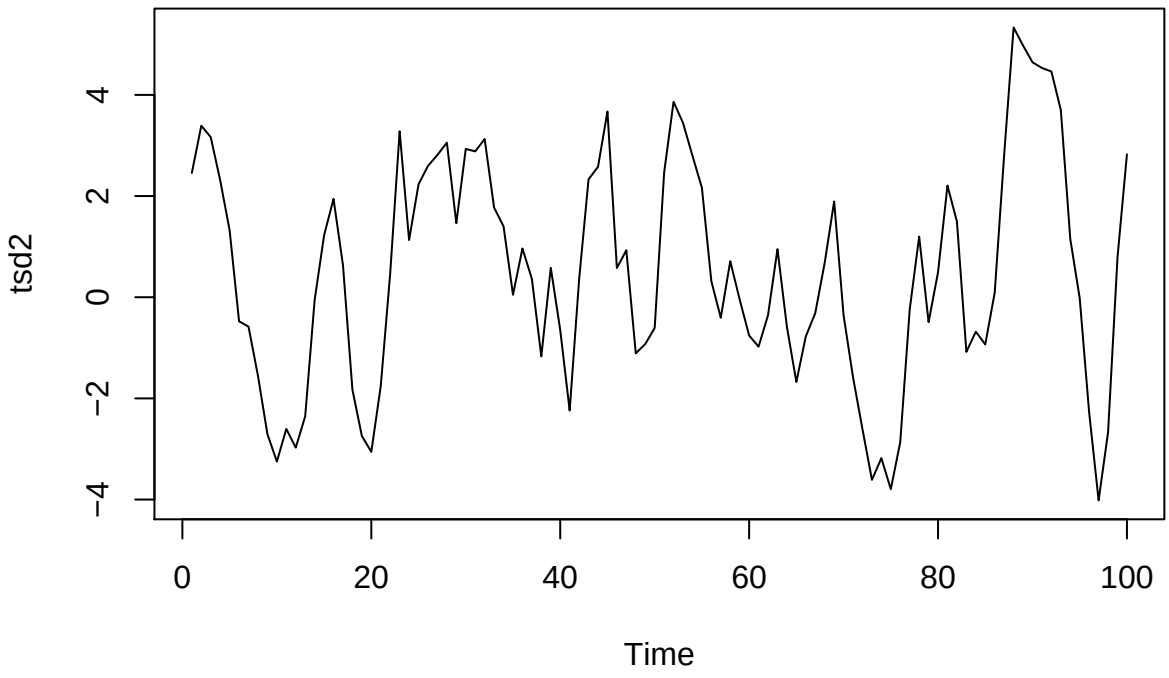
```
arima(tsd1,order=c(1,0,0))
```

```
##
## Call:
## arima(x = tsd1, order = c(1, 0, 0))
##
## Coefficients:
##           ar1  intercept
##       -0.9043     0.0681
## s.e.   0.0422     0.0617
##
## sigma^2 estimated as 1.369:  log likelihood = -158.46,  aic = 320.93
```

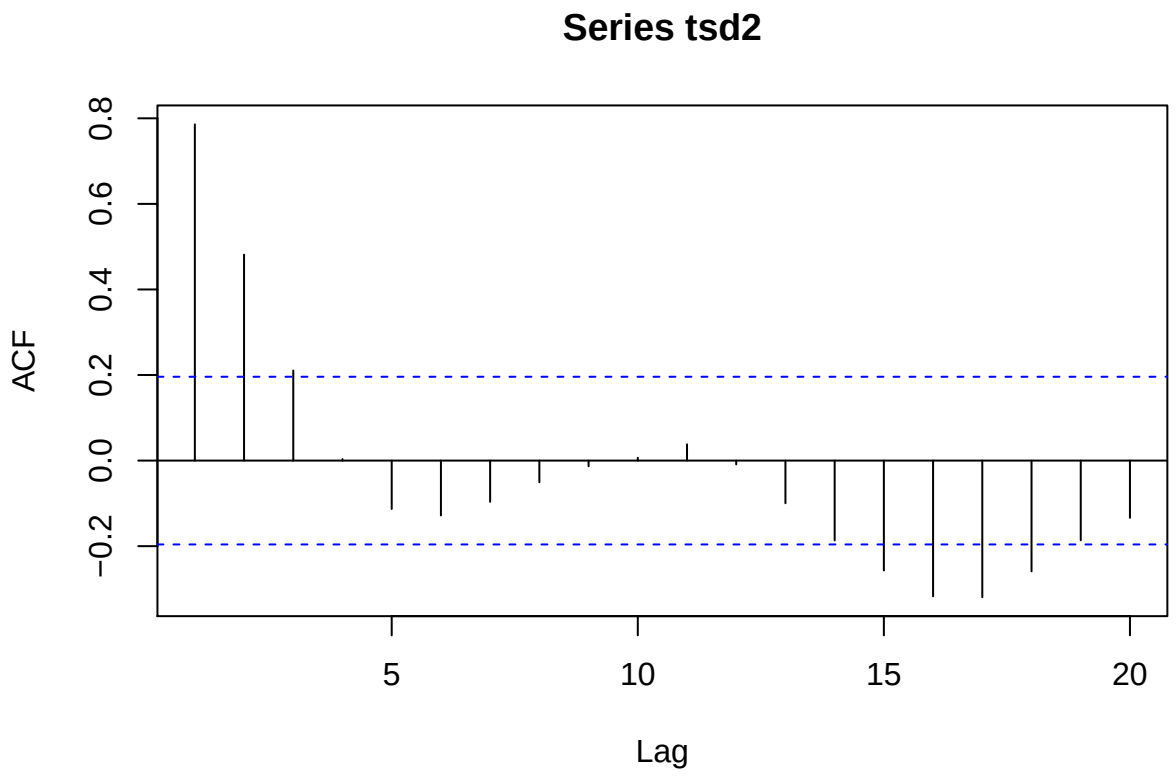
```r
auto.arima(tsd1)
```

```
## Series: tsd1
## ARIMA(1,0,0) with zero mean
##
## Coefficients:
##           ar1
##       -0.9030
## s.e.   0.0424
##
## sigma^2 estimated as 1.4:  log likelihood=-159.07
## AIC=322.13   AICc=322.26   BIC=327.35
```

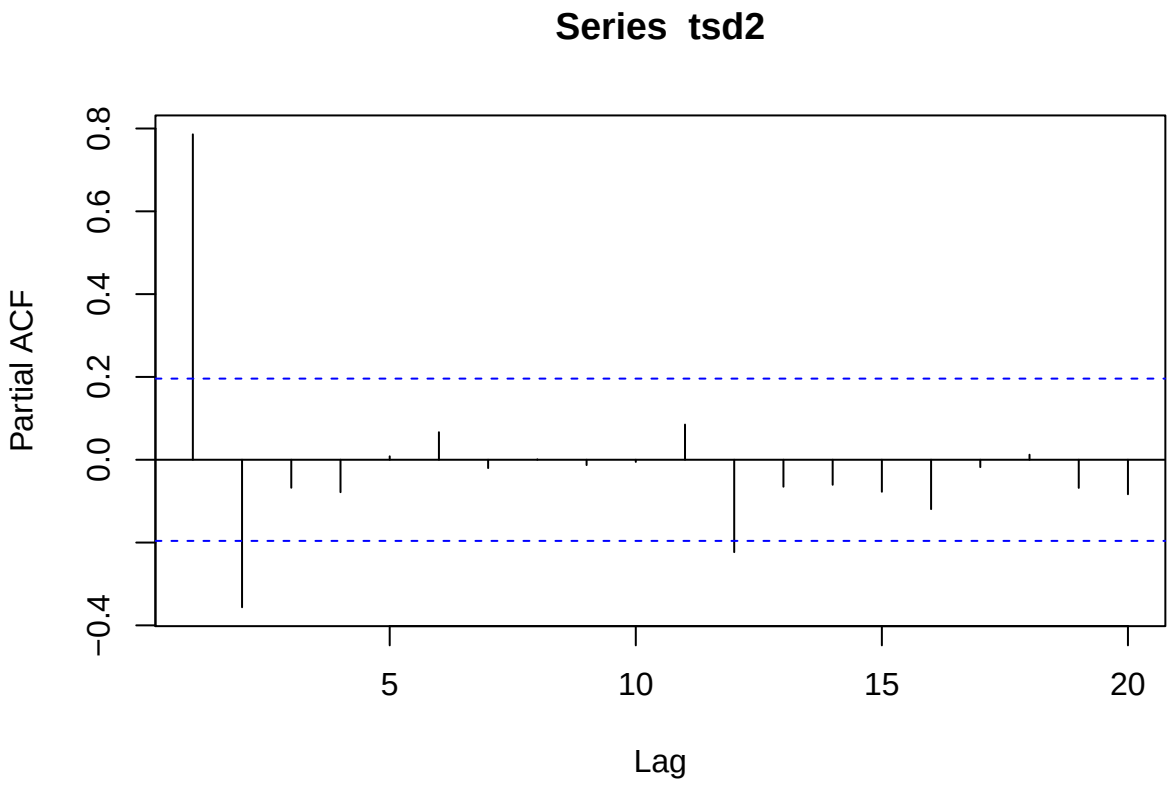
```r
#TSD2
tsd2<-read.table("F:/TSD2.txt")
tsd2<-as.matrix(tsd2)
ts.plot(tsd2)
```




```r
#to estimate p and q in arima(p,d,q) we require pacf,acf and eacf
#acf of tsd1
acf(tsd2)
```




```
pacf(tsd2)
```




```
eacf(tsd2)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x o o o o o o o o  o  o  o
## 1 x x x o x o o o o o o  o  o  o
## 2 x o o o o o o o o o o  o  o  o
## 3 x x o o o o o o o o o  o  o  o
## 4 x x o o o o o o o o o  o  o  o
## 5 x x o x o o o o o o o  o  o  o
## 6 o o o o o o o o o o o  o  o  o
## 7 x o o o o o o o o o o  o  o  o
```

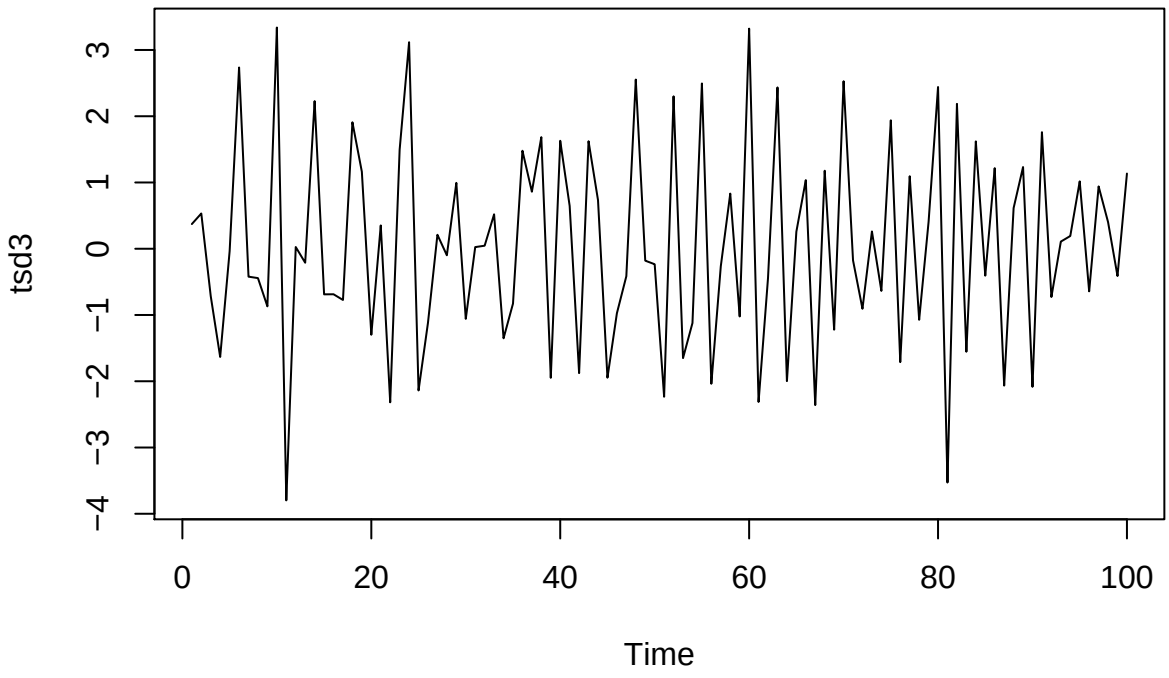
```
arima(tsd2)
```

```
##
## Call:
## arima(x = tsd2)
##
## Coefficients:
##       intercept
##          0.4906
## s.e.     0.2220
##
## sigma^2 estimated as 4.929:  log likelihood = -221.65,  aic = 445.3
```

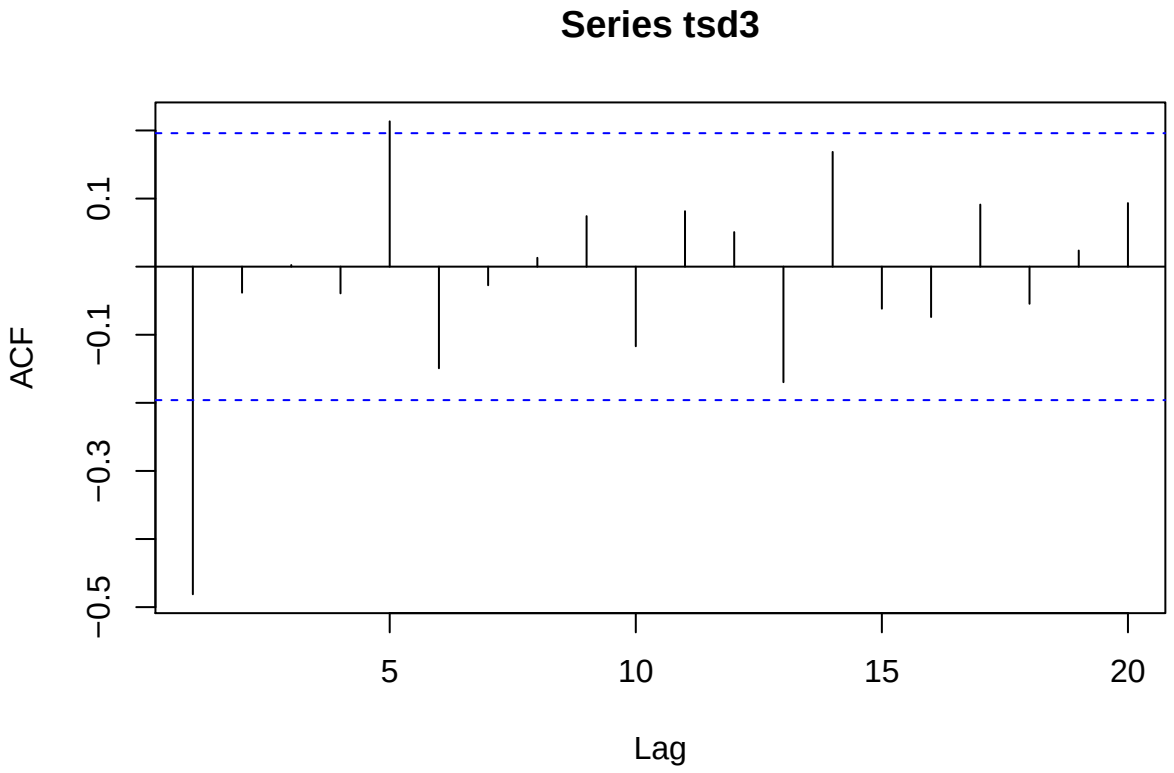
```r
auto.arima(tsd2)
```

```
## Series: tsd2
## ARIMA(2,0,0) with zero mean
##
## Coefficients:
##          ar1      ar2
##       1.1051  -0.3747
## s.e.  0.0932   0.0939
##
## sigma^2 estimated as 1.619:  log likelihood=-165.64
## AIC=337.28   AICc=337.53   BIC=345.09
```

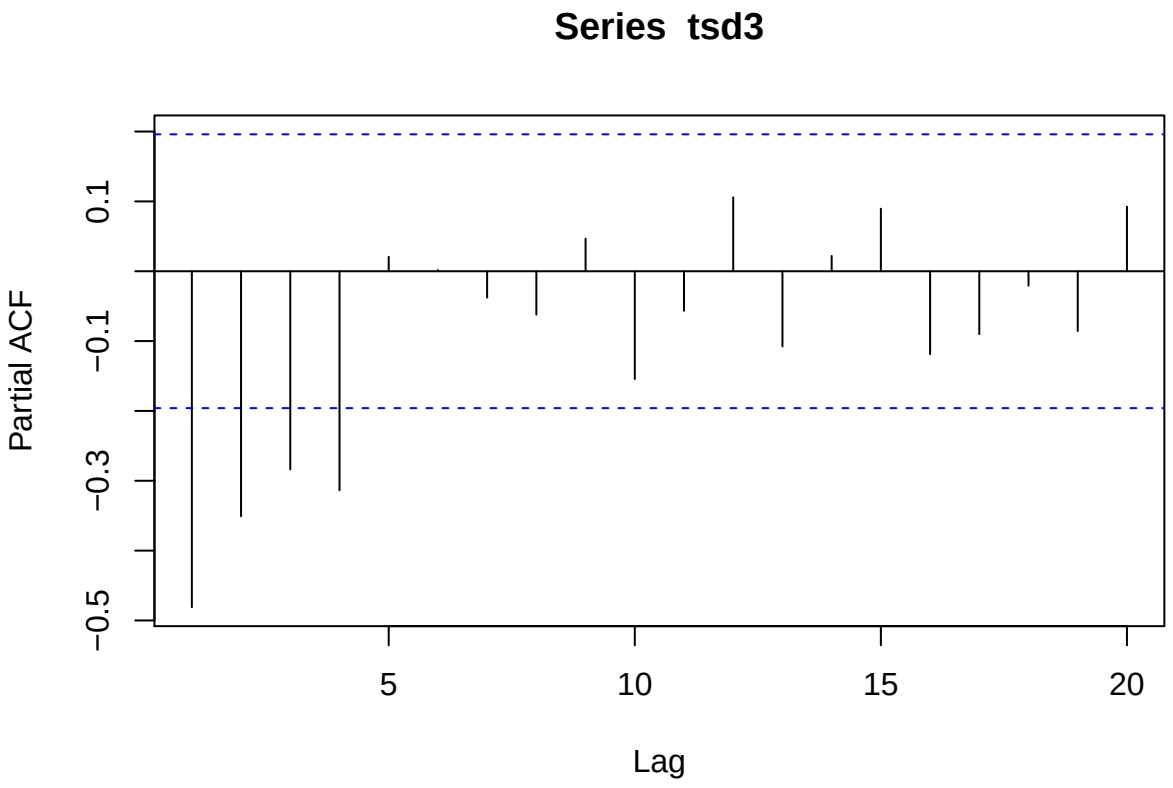
```r
#TSD3
tsd3<-read.table("F:/TSD3.txt")
tsd3<-as.matrix(tsd3)
ts.plot(tsd3)
```




```r
#to estimate p and q in arima(p,d,q) we require pacf,acf and eacf
#acf of tsd1
acf(tsd3)
```




```
pacf(tsd3)
```




```
eacf(tsd3)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x o o o x o o o o o o  o  o  o
## 1 x o o o x o o o o o o  o  o  o
## 2 x o o o x o o o o o o  o  o  o
## 3 x o o o x o o o o o o  o  o  o
## 4 o o o x o o o o o o o  o  o  o
## 5 x o o x o x o o o o o  o  o  o
## 6 o o o o o x o o o o o  o  o  o
## 7 x x o o o x x o o o o  o  o  o
```

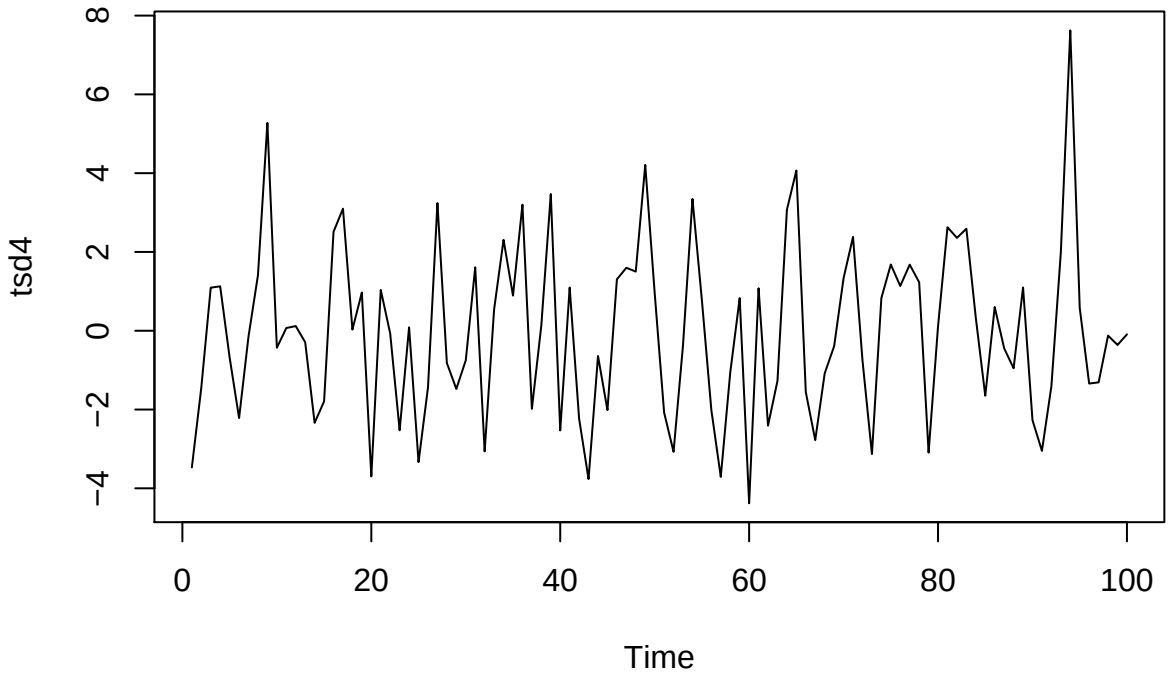
```
arima(tsd3)
```

```
##
## Call:
## arima(x = tsd3)
##
## Coefficients:
##       intercept
##          0.0457
## s.e.     0.1541
##
## sigma^2 estimated as 2.374:  log likelihood = -185.12,  aic = 372.23
```

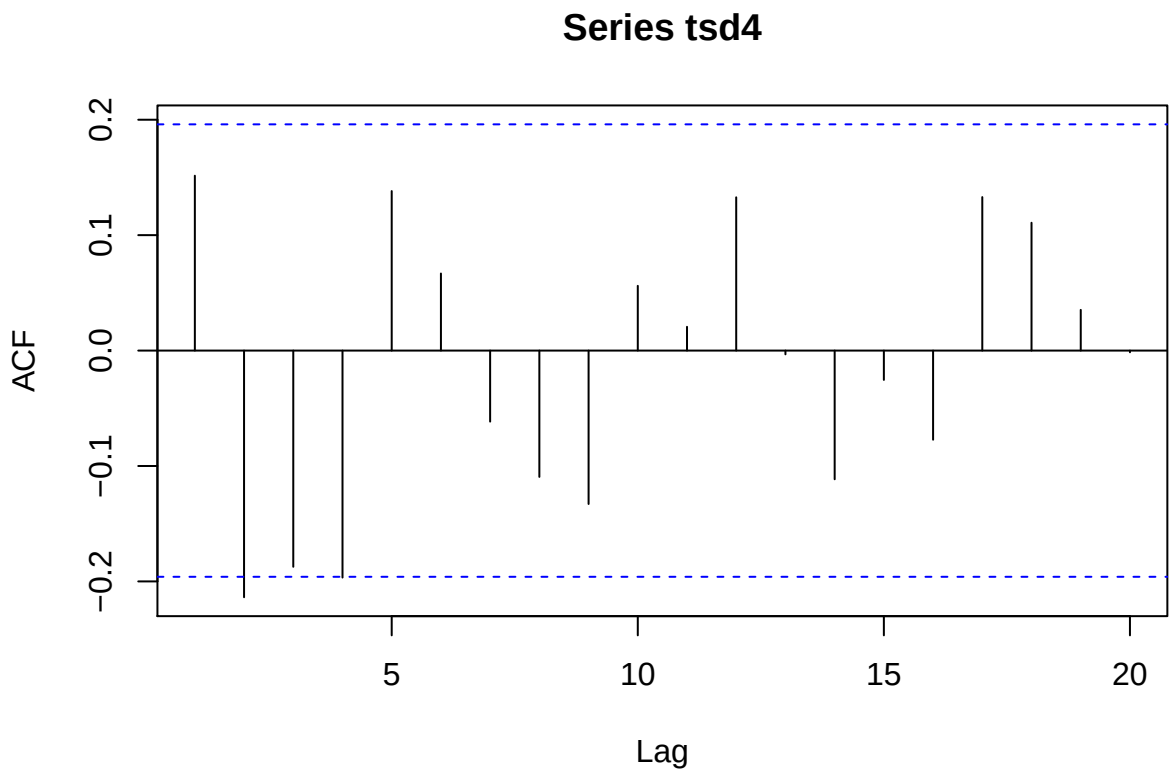
```r
auto.arima(tsd3)
```

```
## Series: tsd3
## ARIMA(0,0,1) with non-zero mean
##
## Coefficients:
##           ma1    mean
##       -0.8675  0.0312
## s.e.   0.0666  0.0168
##
## sigma^2 estimated as 1.373:  log likelihood=-157.44
## AIC=320.89   AICc=321.14   BIC=328.7
```

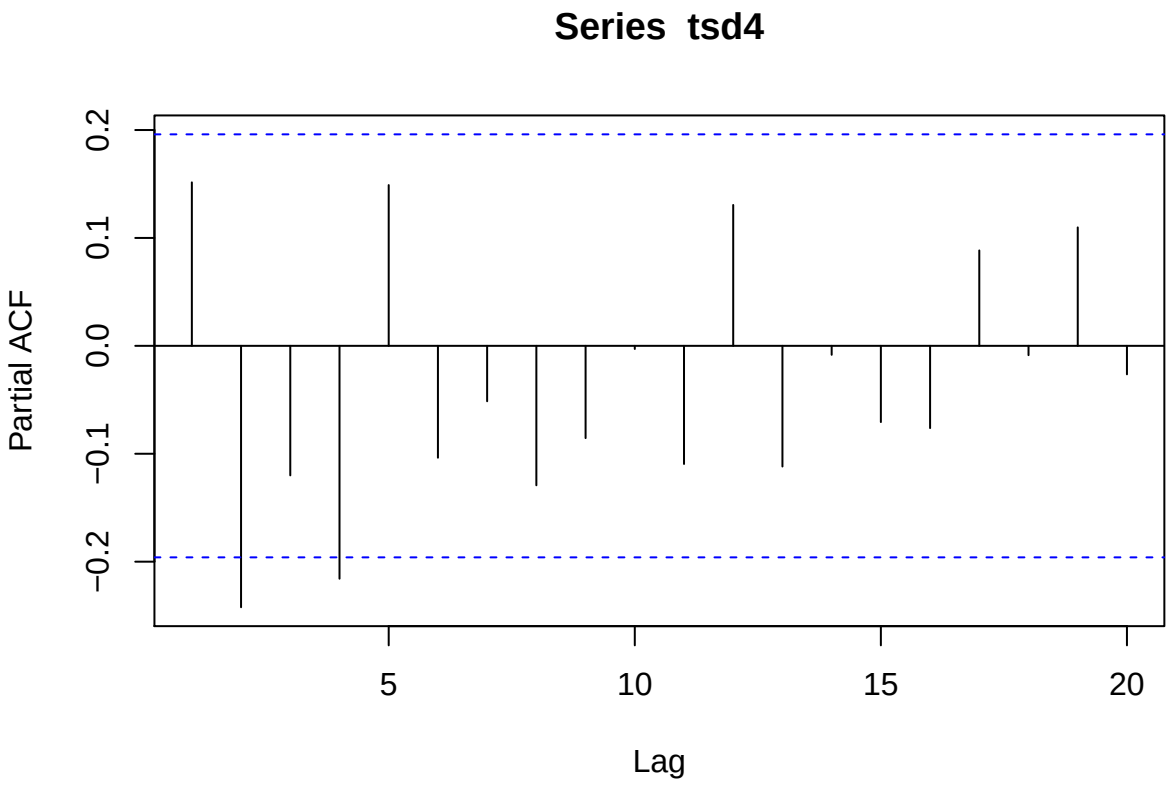
```r
#TSD4
#TSD2
tsd4<-read.table("F:/TSD4.txt")
tsd4<-as.matrix(tsd4)
ts.plot(tsd4)
```




```r
#to estimate p and q in arima(p,d,q) we require pacf,acf and eacf
#acf of tsd1
acf(tsd4)
```




```
pacf(tsd4)
```




```
eacf(tsd4)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 o x o o o o o o o o o  o  o  o
## 1 x x o o x o o o o o o  o  o  o
## 2 x x o x o o o o o o o  o  o  o
## 3 x x x x o o o o o o o  o  o  o
## 4 x o x x o o o o o o o  o  o  o
## 5 x o o o x o o o o o o  o  o  o
## 6 x o o o x o o o o o o  o  o  o
## 7 x x o o o o x o o o o  o  o  o
```

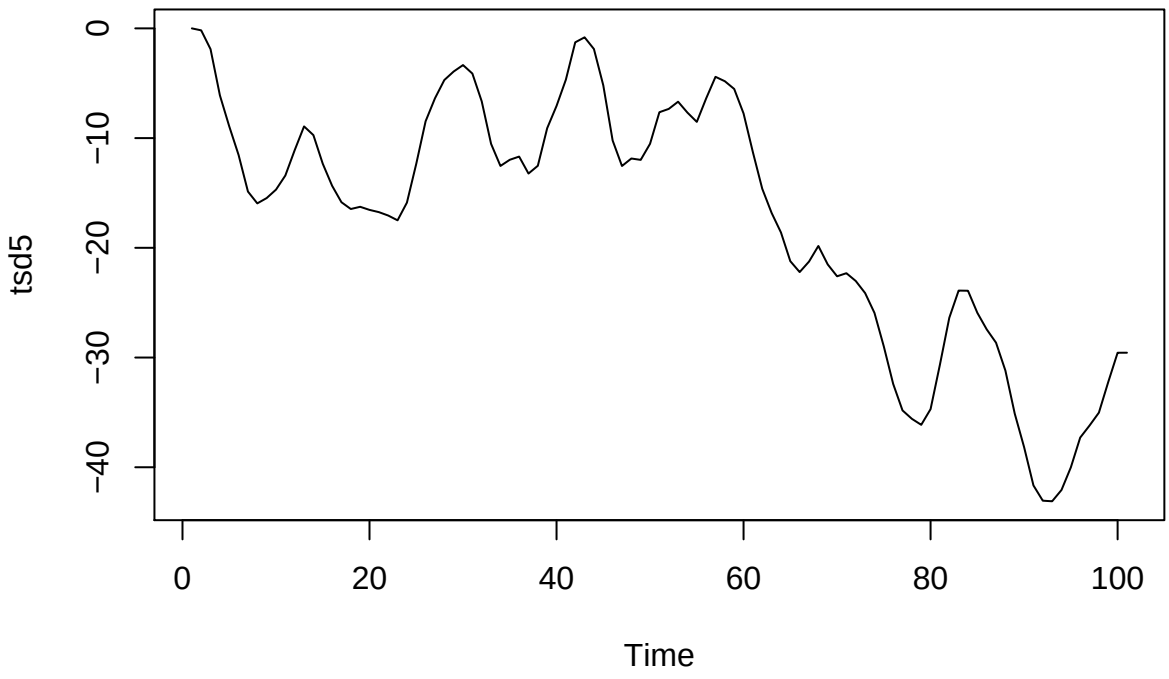
```
arima(tsd4)
```

```
##
## Call:
## arima(x = tsd4)
##
## Coefficients:
##       intercept
##         -0.0305
## s.e.     0.2193
##
## sigma^2 estimated as 4.811:  log likelihood = -220.44,  aic = 442.88
```

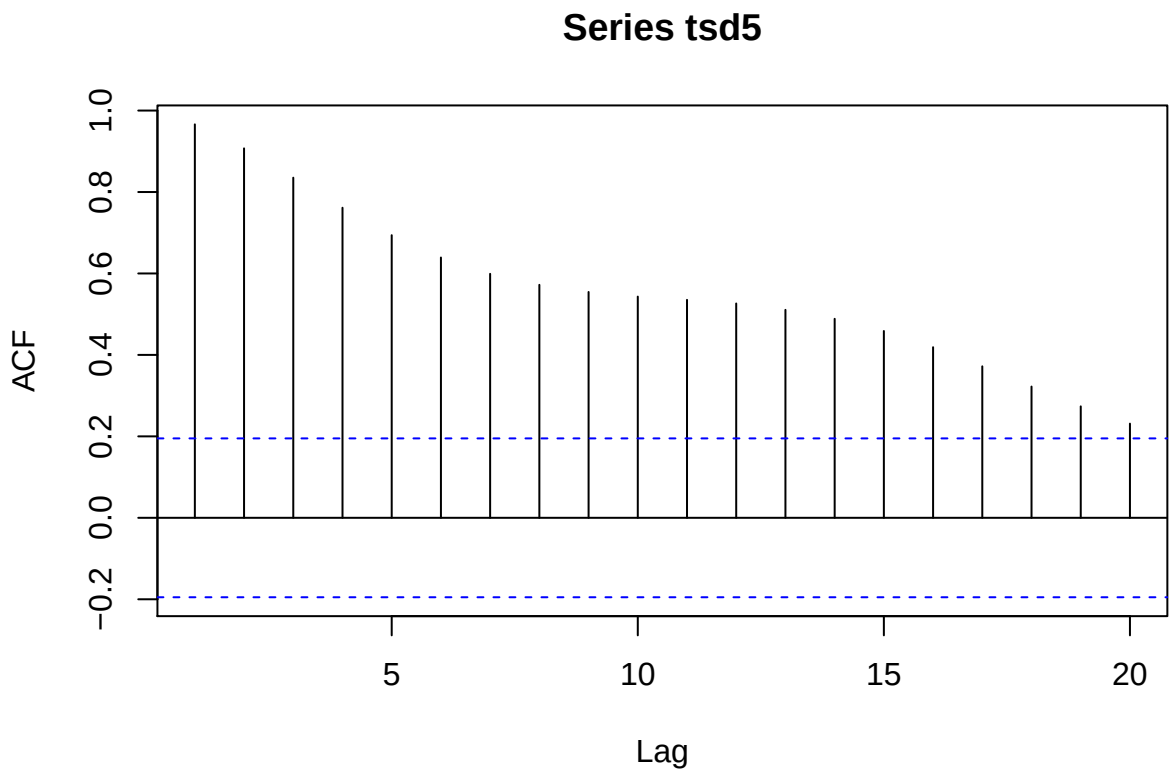
```r
auto.arima(tsd4)
```

```
## Series: tsd4
## ARIMA(1,0,2) with zero mean
##
## Coefficients:
##          ar1      ma1      ma2
##       0.5705  -0.4299  -0.3929
## s.e.  0.2110   0.2212   0.1175
##
## sigma^2 estimated as 4.301:  log likelihood=-213.56
## AIC=435.12   AICc=435.54   BIC=445.54
```

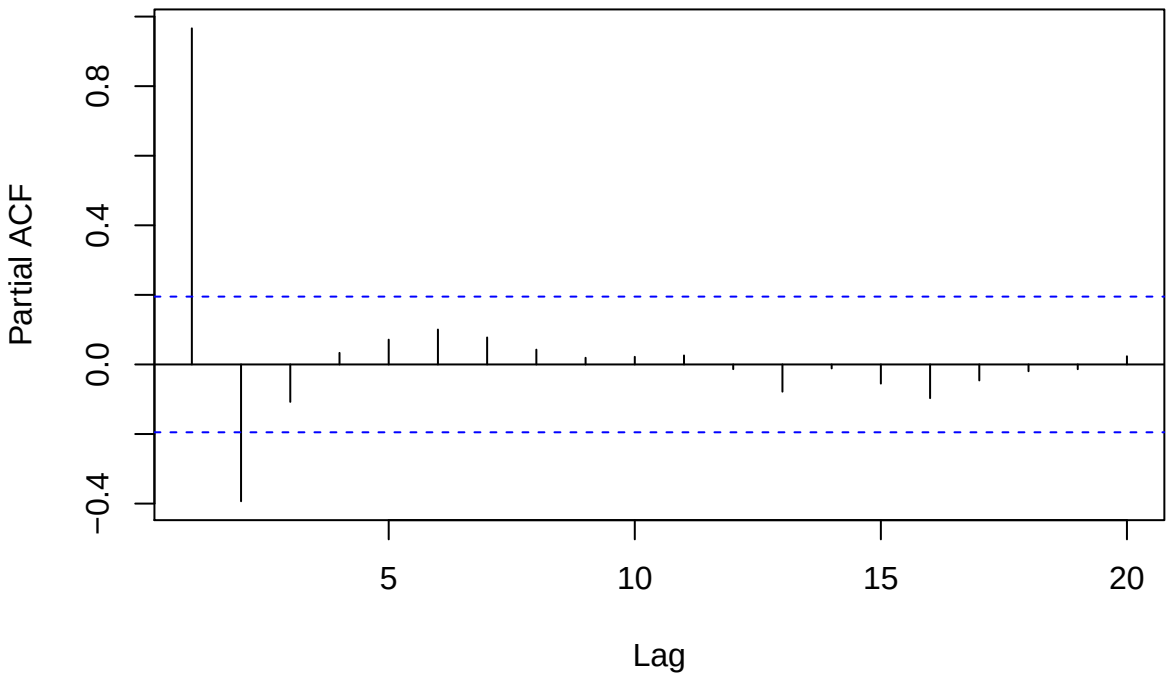
```r
#TSD5
#TSD2
tsd5<-read.table("F:/TSD5.txt")
tsd5<-as.matrix(tsd5)
ts.plot(tsd5)
```




```r
#to estimate p and q in arima(p,d,q) we require pacf,acf and eacf
#acf of tsd1
acf(tsd5)
```




```
pacf(tsd5)
```




```
eacf(tsd5)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x x x x x x x  x  x  x
## 1 x x x o x x x x x o o  o  x  x
## 2 x o x o o o o o o o o  o  o  o
## 3 x o x o o o o o o o o  o  o  o
## 4 x x x o o o o o o o o  o  o  o
## 5 o x x o o o o o o o o  o  o  o
## 6 o x o x o o o o o o o  o  o  o
## 7 x x x o o o o o o o o  o  o  o
```

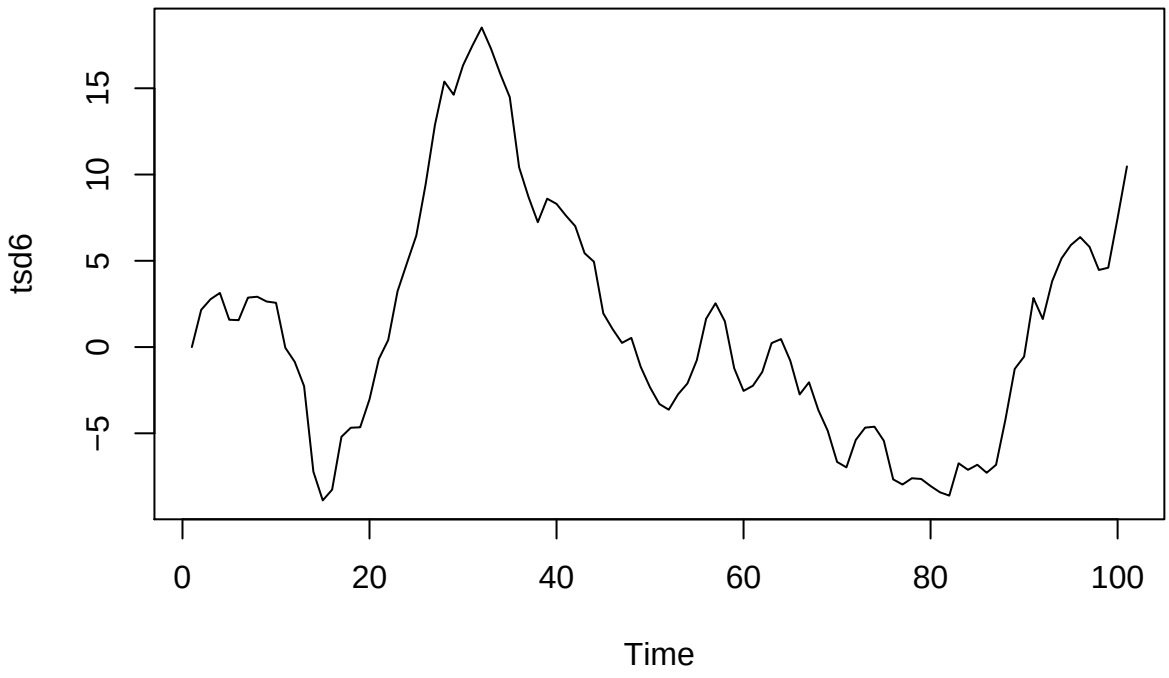
```
arima(tsd5)
```

```
##
## Call:
## arima(x = tsd5)
##
## Coefficients:
##       intercept
##        -17.3970
## s.e.     1.1423
##
## sigma^2 estimated as 131.8:  log likelihood = -389.82,  aic = 781.64
```

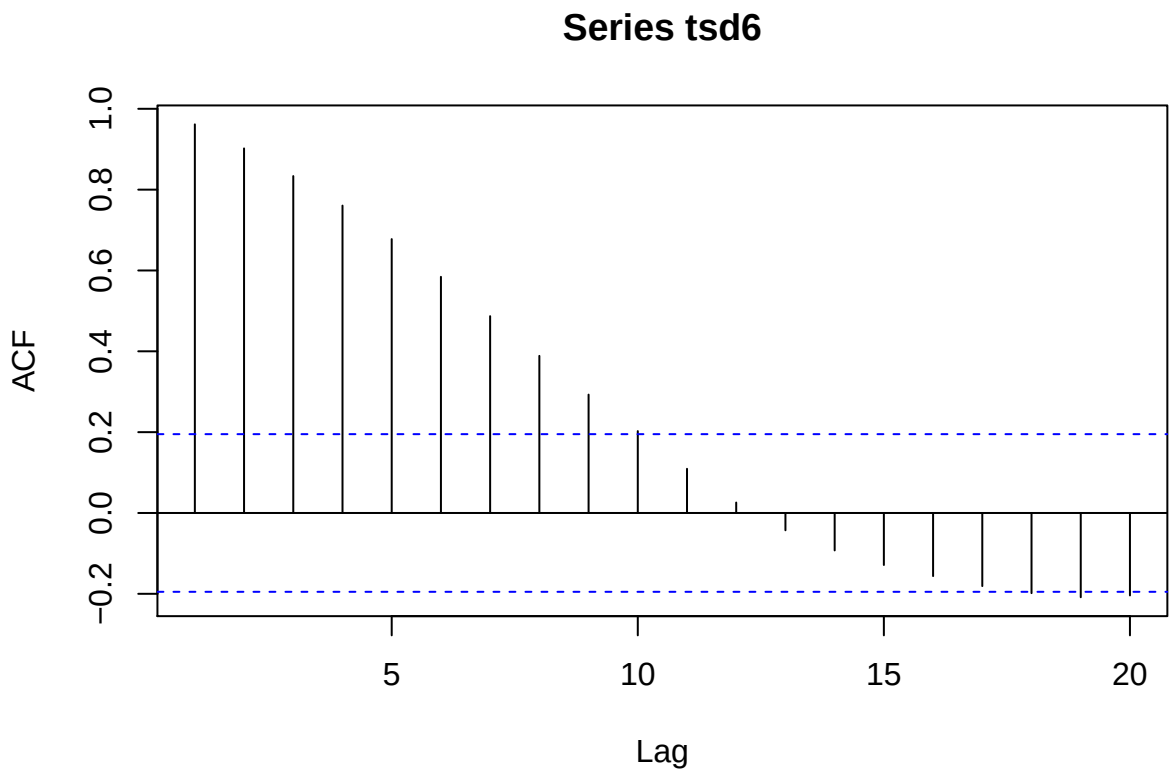
```r
auto.arima(tsd5)
```

```
## Series: tsd5
## ARIMA(2,1,2) with drift
##
## Coefficients:
##          ar1      ar2      ma1      ma2    drift
##       1.4339  -0.5827  -0.2787  -0.5223  -0.3061
## s.e.  0.1060   0.0941   0.1174   0.1132   0.1692
##
## sigma^2 estimated as 1.52:  log likelihood=-161.13
## AIC=334.26   AICc=335.16   BIC=349.89
```

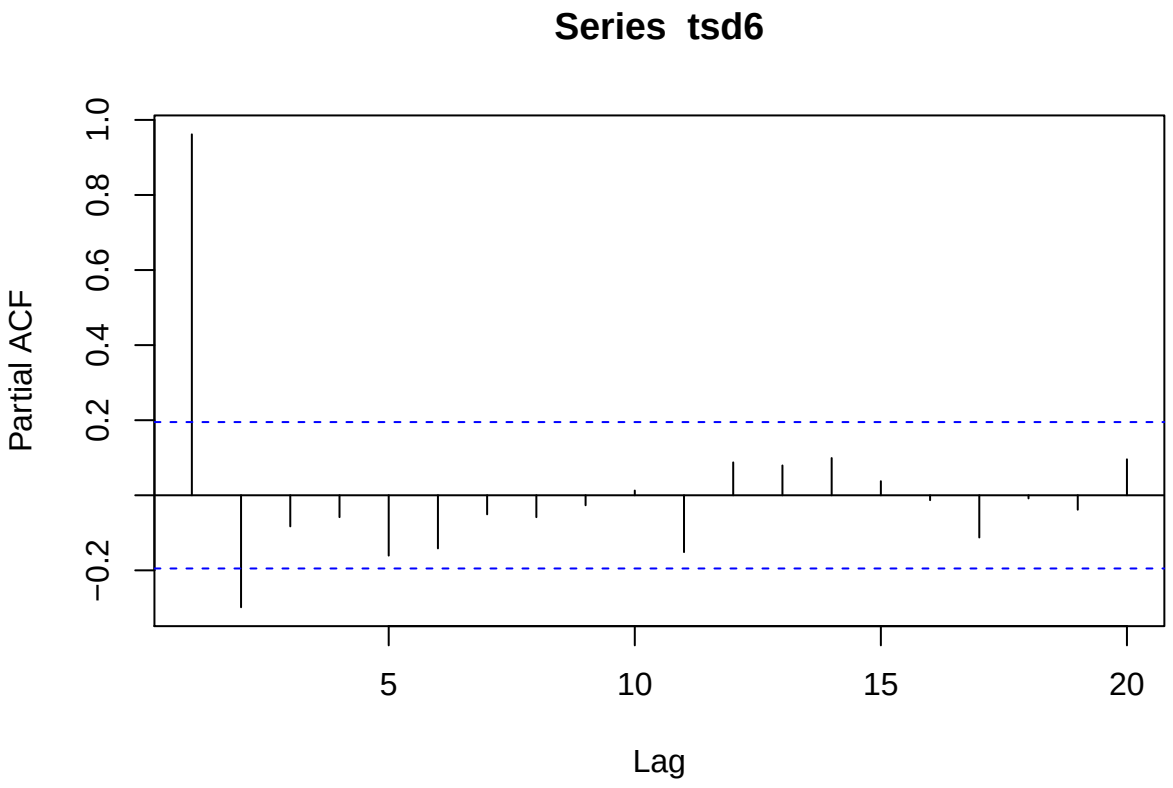
```r
#TSD6
tsd6<-read.table("F:/TSD6.txt")
tsd6<-as.matrix(tsd6)
ts.plot(tsd6)
```




```r
#to estimate p and q in arima(p,d,q) we require pacf,acf and eacf
#acf of tsd1
acf(tsd6)
```




```
pacf(tsd6)
```




```
eacf(tsd6)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x x x x x o o  o  o  o
## 1 x x o x x x x o o o o  o  o  o
## 2 o o o o o o o o o o o  o  o  o
## 3 x o o o o o o o o o o  o  o  o
## 4 x x x o o o o o o o o  o  o  o
## 5 x x x x o o o o o o o  o  o  o
## 6 o x x o o o o o o o o  o  o  o
## 7 o o o o o o o o o o o  o  o  o
```

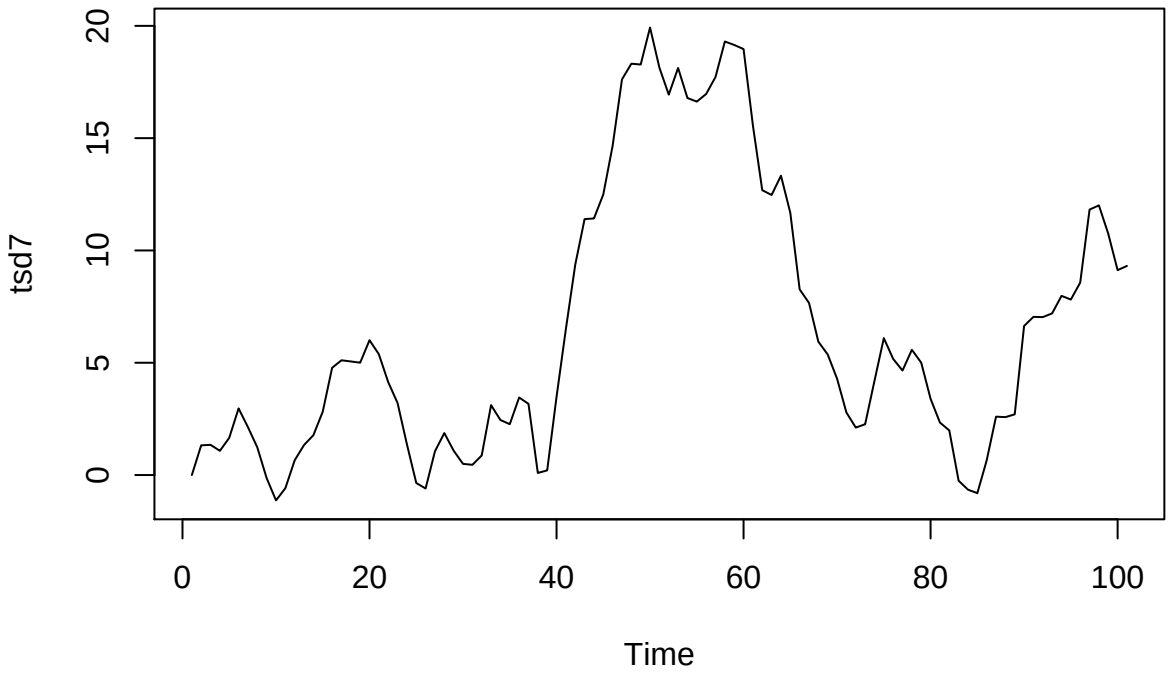
```
arima(tsd6)
```

```
##
## Call:
## arima(x = tsd6)
##
## Coefficients:
##       intercept
##          1.1125
## s.e.     0.6723
##
## sigma^2 estimated as 45.65:  log likelihood = -336.27,  aic = 674.54
```

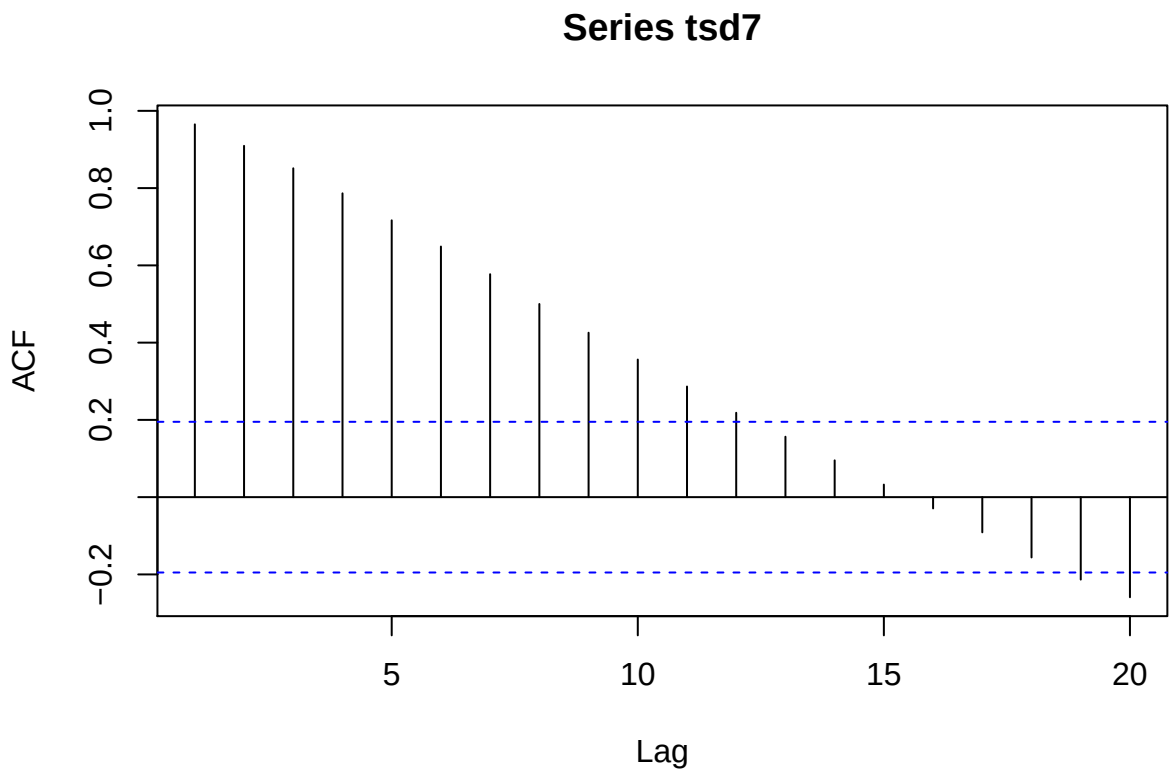
```r
auto.arima(tsd6)
```

```
## Series: tsd6
## ARIMA(1,1,0)
##
## Coefficients:
##          ar1
##       0.4993
## s.e.  0.0886
##
## sigma^2 estimated as 2.037:  log likelihood=-177.11
## AIC=358.23   AICc=358.35   BIC=363.44
```

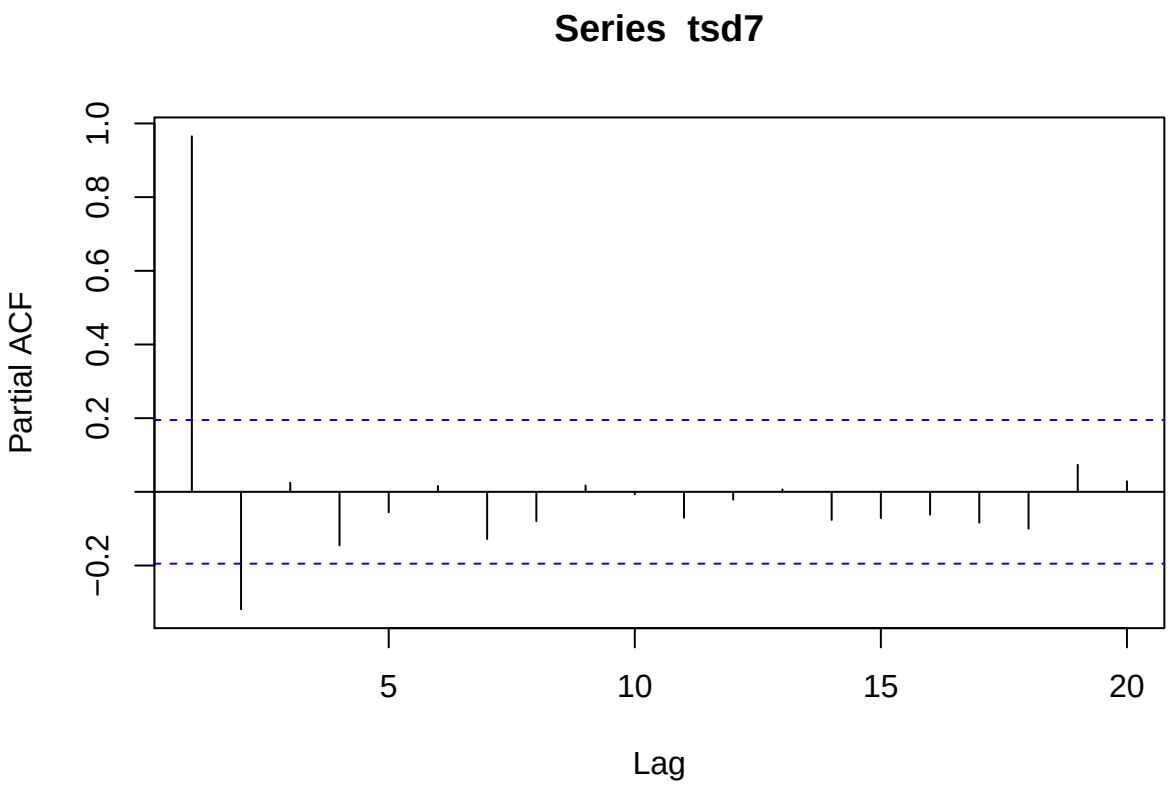
```r
#TSD7
tsd7<-read.table("F:/TSD7.txt")
tsd7<-as.matrix(tsd7)
ts.plot(tsd7)
```




```r
#to estimate p and q in arima(p,d,q) we require pacf,acf and eacf
#acf of tsd1
acf(tsd7)
```




```
pacf(tsd7)
```




```
eacf(tsd7)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x x x x x x x  x  o  o
## 1 x o o o o o o o o o o  o  o  o
## 2 x x o o o o o o o o o  o  o  o
## 3 x x x o o o o o o o o  o  o  o
## 4 o x o o o o o o o o o  o  o  o
## 5 x o o o o o o o o o o  o  o  o
## 6 x x o o o o o o o o o  o  o  o
## 7 x o x x o x o o o o o  o  o  o
```

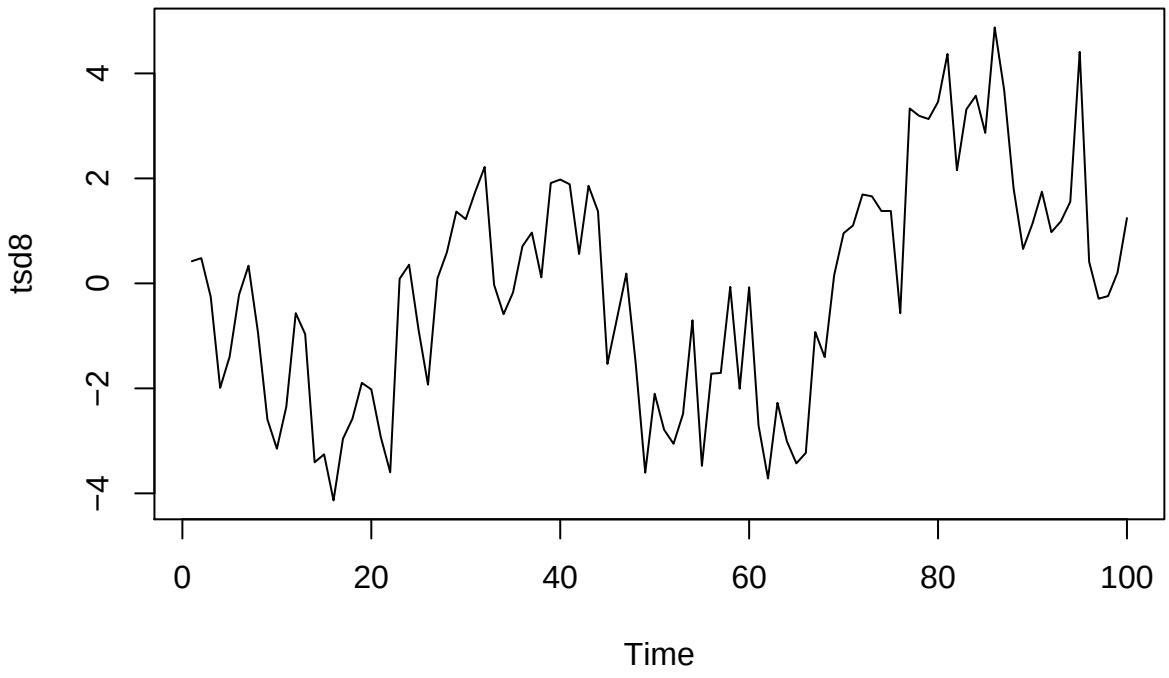
```
arima(tsd7)
```

```
##
## Call:
## arima(x = tsd7)
##
## Coefficients:
##       intercept
##          6.4674
## s.e.     0.6002
##
## sigma^2 estimated as 36.38:  log likelihood = -324.82,  aic = 651.63
```

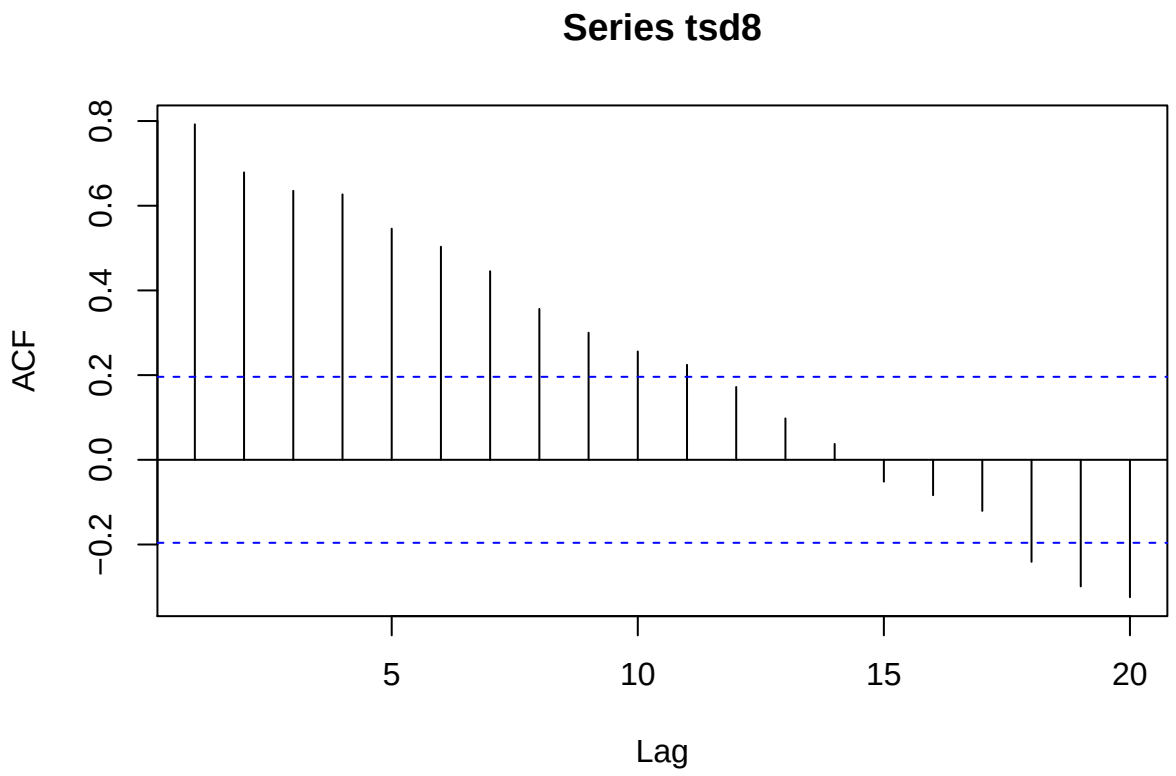
```r
auto.arima(tsd7)
```

```
## Series: tsd7
## ARIMA(0,1,1)
##
## Coefficients:
##          ma1
##       0.5397
## s.e.  0.0875
##
## sigma^2 estimated as 1.63:  log likelihood=-165.98
## AIC=335.96   AICc=336.09   BIC=341.18
```

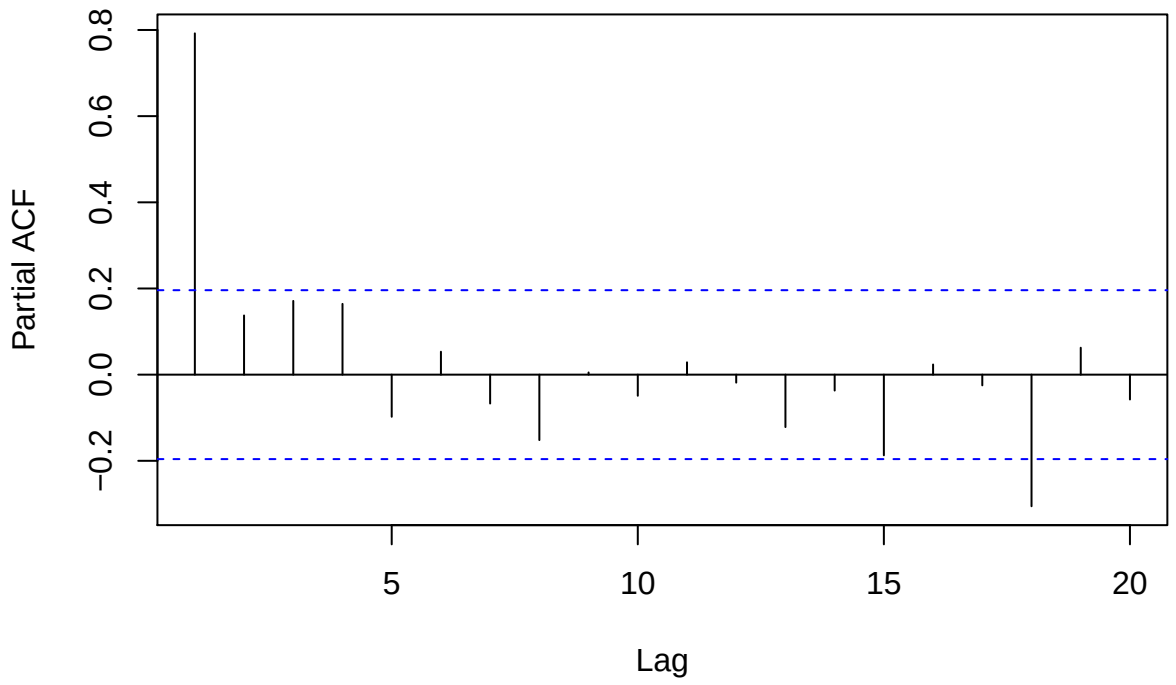
```r
#TSD8
tsd8<-read.table("F:/TSD8.txt")
tsd8<-as.matrix(tsd8)
ts.plot(tsd8)
```




```r
#to estimate p and q in arima(p,d,q) we require pacf,acf and eacf
#acf of tsd1
acf(tsd8)
```




```
pacf(tsd8)
```




```
eacf(tsd8)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x x x x x x x  o  o  o
## 1 o o o o o o o o o o o  o  o  o
## 2 x o o o o o o o o o o  o  o  o
## 3 x o x o o o o o o o o  o  o  o
## 4 x o o o o o o o o o o  o  o  o
## 5 x o o x o o o o o o o  o  o  o
## 6 x o o x o o o o o o o  o  o  o
## 7 x x x x o x x o o o o  o  o  o
```

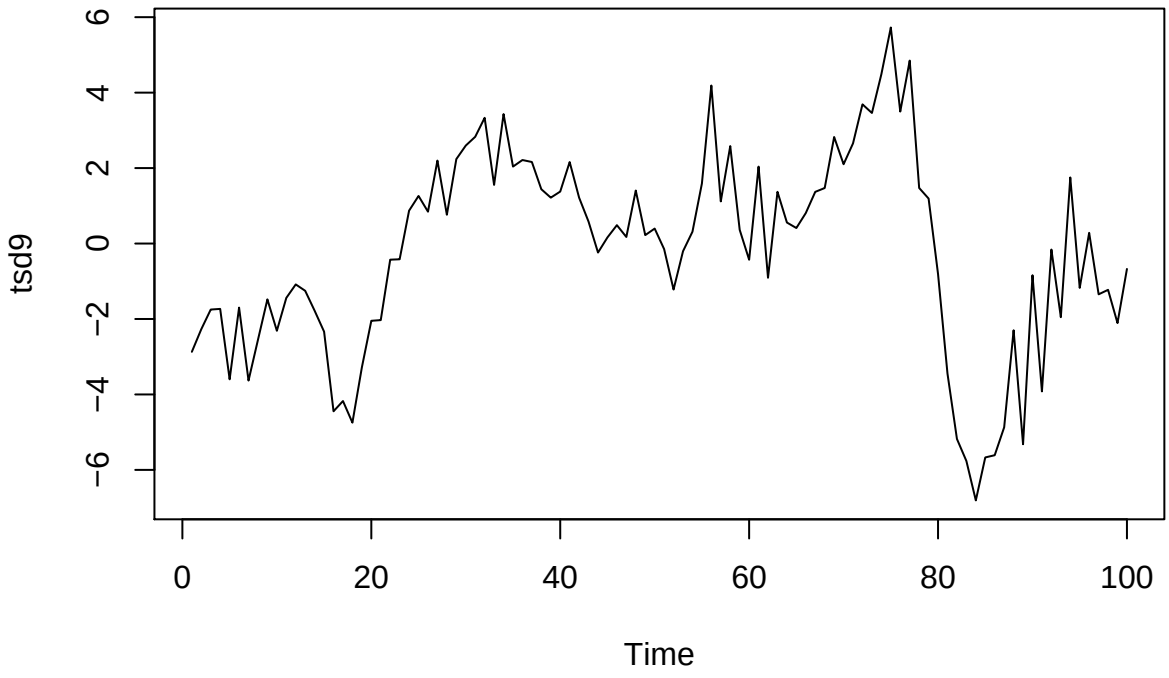
```
arima(tsd8)
```

```
##
## Call:
## arima(x = tsd8)
##
## Coefficients:
##       intercept
##         -0.1206
## s.e.     0.2143
##
## sigma^2 estimated as 4.592:  log likelihood = -218.11,  aic = 438.22
```

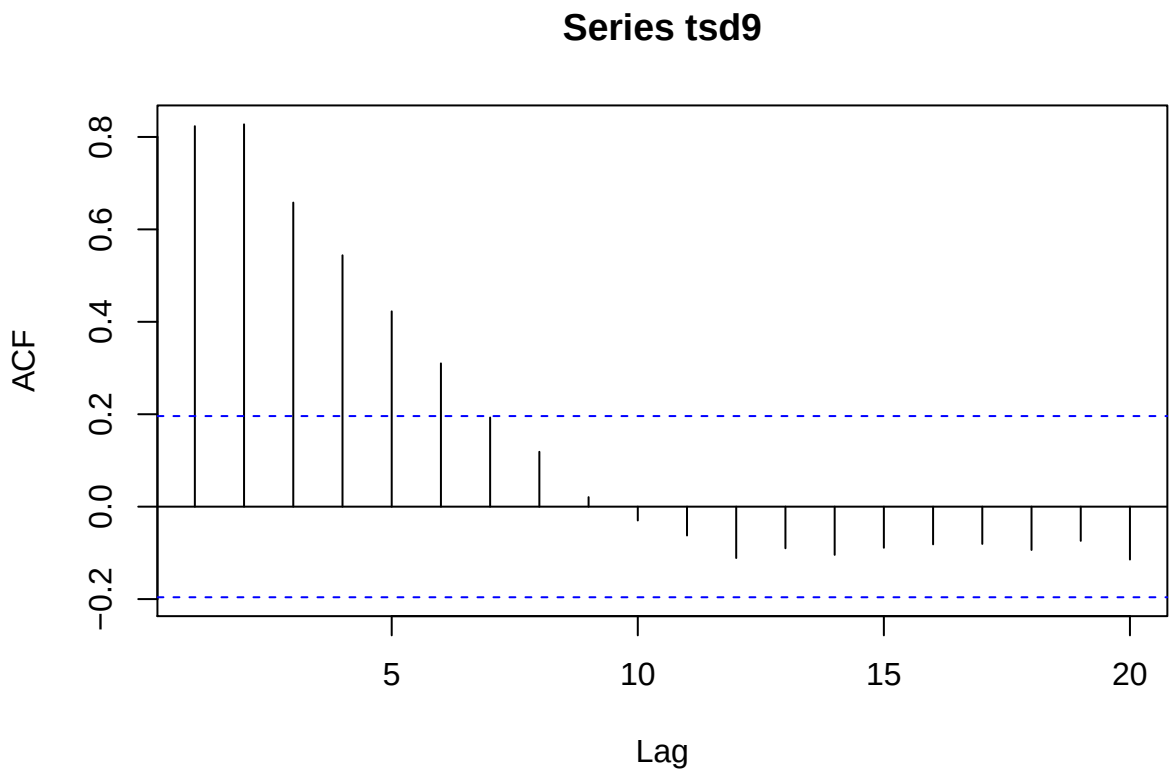
```r
auto.arima(tsd8)
```

```
## Series: tsd8
## ARIMA(0,1,2)
##
## Coefficients:
##           ma1      ma2
##       -0.3202  -0.1596
## s.e.   0.0962   0.0882
##
## sigma^2 estimated as 1.716:  log likelihood=-166.31
## AIC=338.62   AICc=338.87   BIC=346.41
```

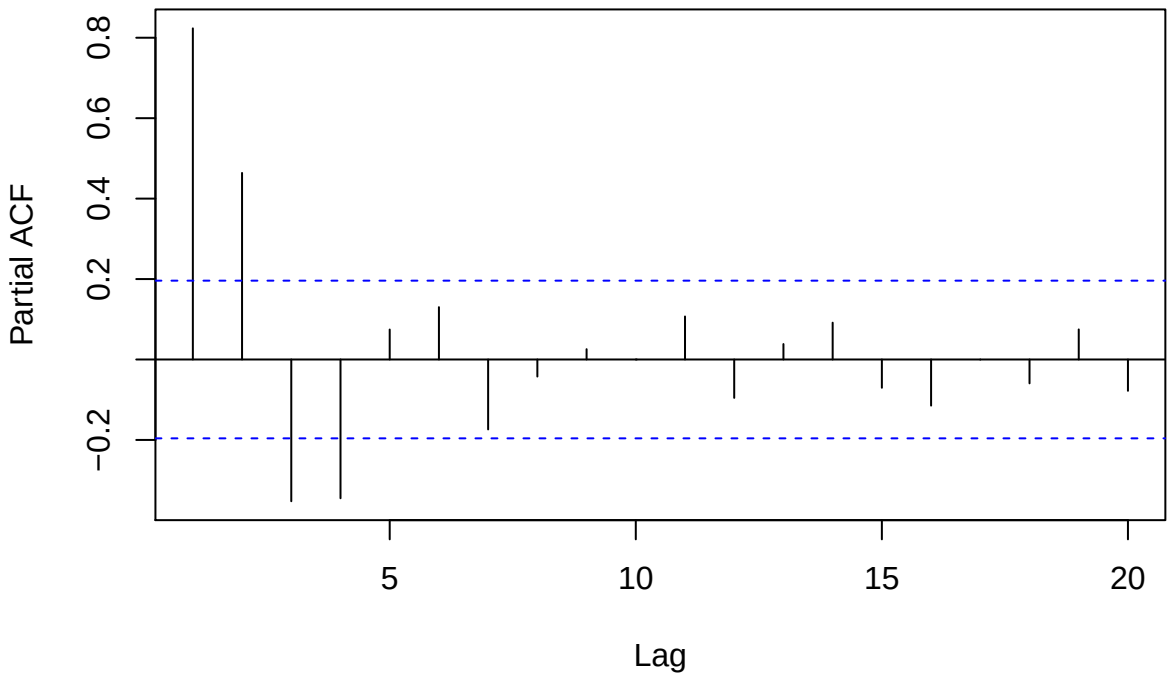
```r
#TSD9
tsd9<-read.table("F:/TSD9.txt")
tsd9<-as.matrix(tsd9)
ts.plot(tsd9)
```




```r
#to estimate p and q in arima(p,d,q) we require pacf,acf and eacf
#acf of tsd1
acf(tsd9)
```




```
pacf(tsd9)
```




```
eacf(tsd9)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x x o o o o o  o  o  o
## 1 x x o o o o o o o o o  o  o  o
## 2 x x x o o o o o o o o  o  o  o
## 3 x x x o o o o x o o o  o  o  o
## 4 x x o o o o o o o o o  o  o  o
## 5 x x o x o o o o o o o  o  o  o
## 6 x x o x o o o o o o o  o  o  o
## 7 o x o x o o x o o o o  o  o  o
```

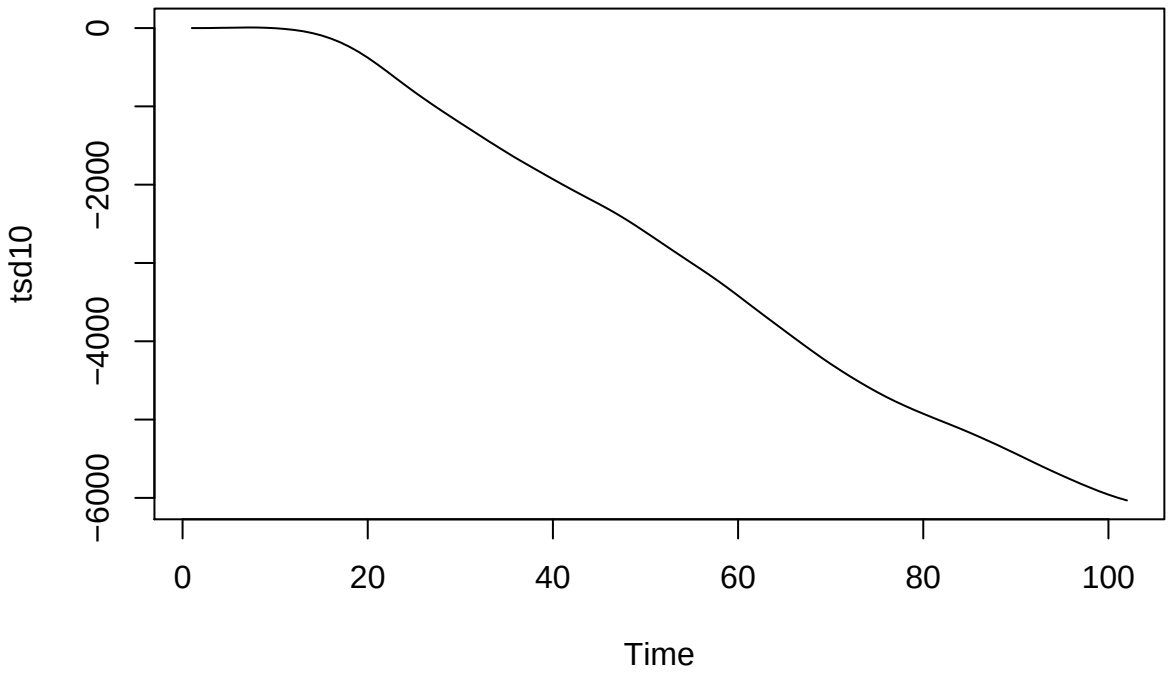
```
arima(tsd9)
```

```
##
## Call:
## arima(x = tsd9)
##
## Coefficients:
##       intercept
##         -0.2035
## s.e.     0.2616
##
## sigma^2 estimated as 6.846:  log likelihood = -238.08,  aic = 478.15
```

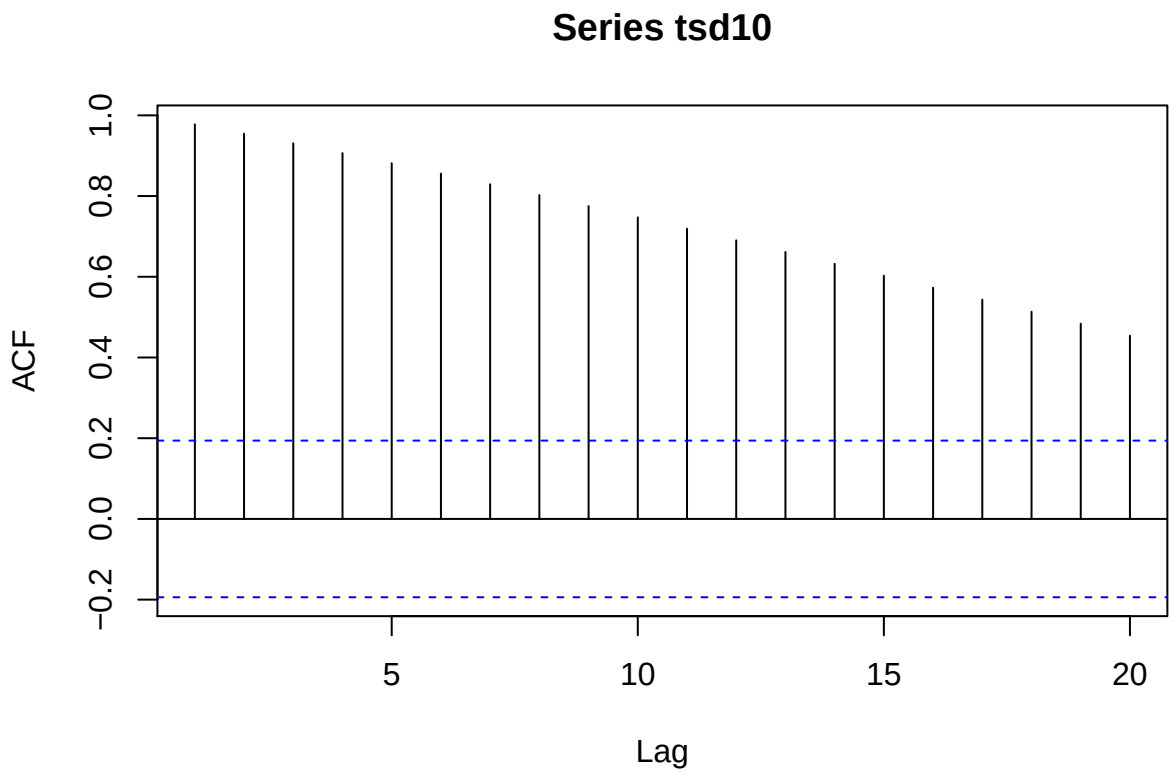
```r
auto.arima(tsd9)
```

```
## Series: tsd9
## ARIMA(1,0,2) with zero mean
##
## Coefficients:
##          ar1      ma1     ma2
##       0.7837  -0.1952  0.9232
## s.e.  0.0621   0.0405  0.0620
##
## sigma^2 estimated as 1.088:  log likelihood=-147.34
## AIC=302.69   AICc=303.11   BIC=313.11
```

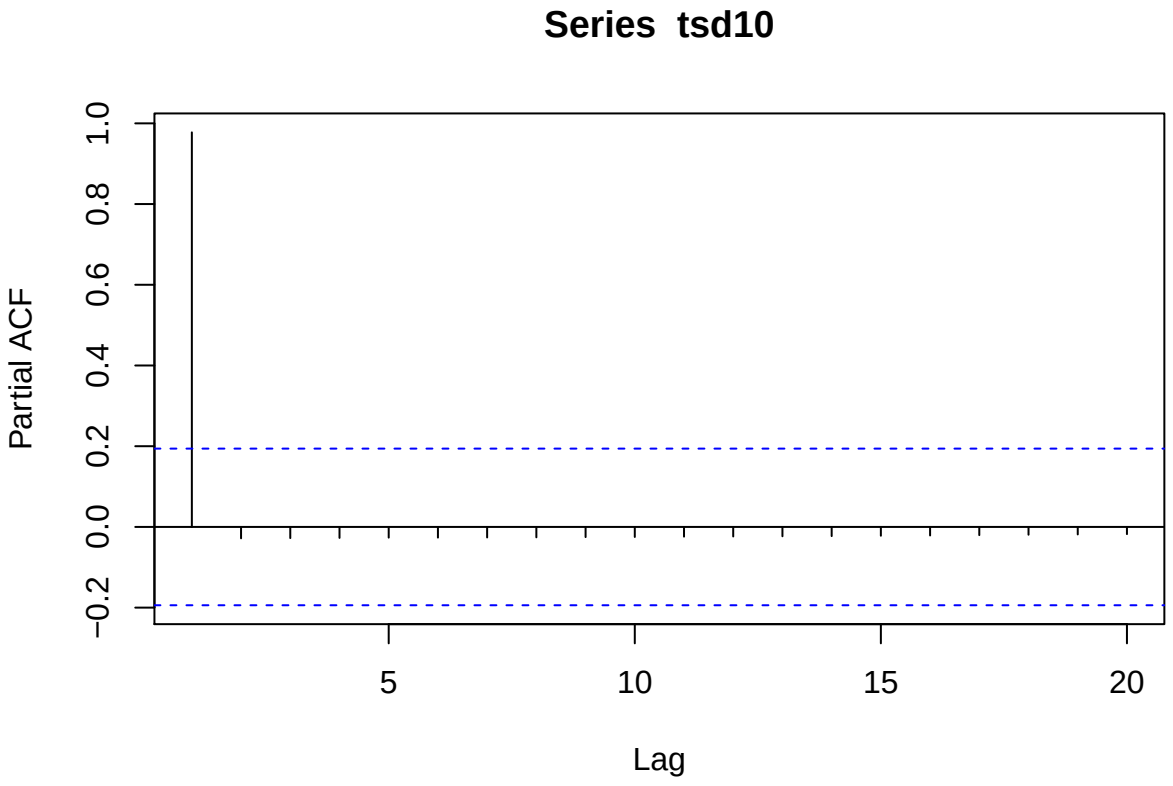
```r
#TSD10
tsd10<-read.table("F:/TSD10.txt")
tsd10<-as.matrix(tsd10)
ts.plot(tsd10)
```




```r
#to estimate p and q in arima(p,d,q) we require pacf,acf and eacf
#acf of tsd1
acf(tsd10)
```




```
pacf(tsd10)
```




```
arima(tsd10)
```

```
##
## Call:
## arima(x = tsd10)
##
## Coefficients:
##        intercept
##       -2808.5121
## s.e.    200.7082
##
## sigma^2 estimated as 4108940:  log likelihood = -921.39,  aic = 1844.79
```

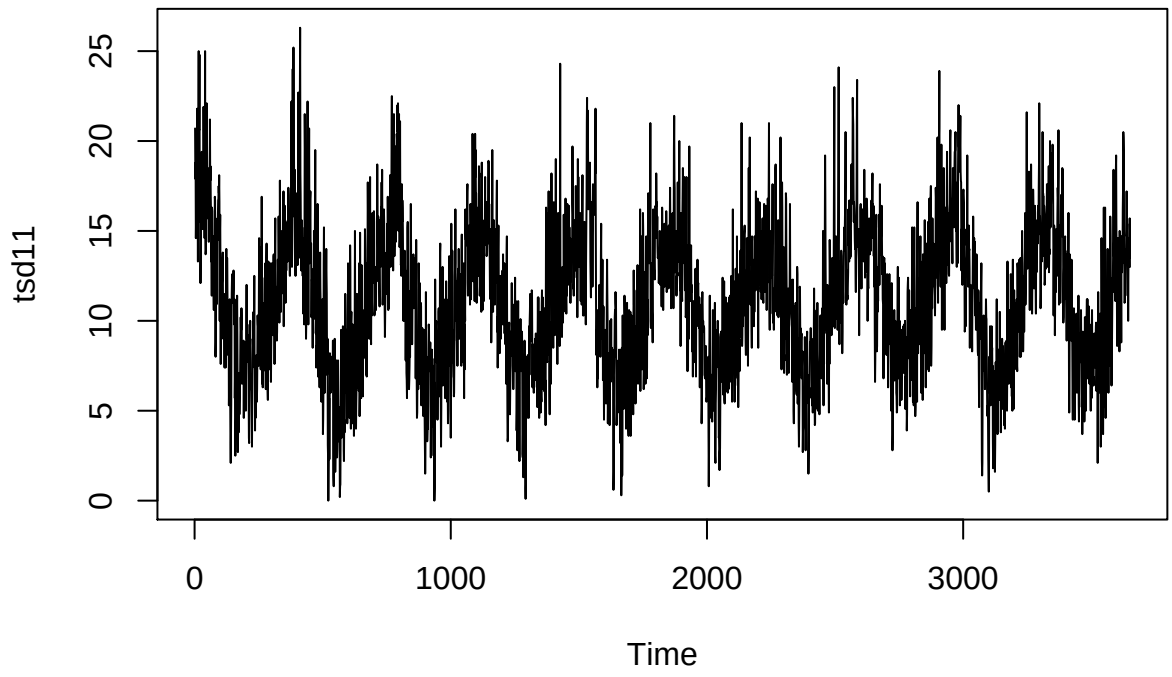
```
auto.arima(tsd10)
```

```
## Series: tsd10
## ARIMA(1,2,3)
##
## Coefficients:
##          ar1      ma1     ma2     ma3
##       0.8097  -0.1776  0.8333  0.1985
## s.e.  0.0691   0.1144  0.0702  0.1120
##
## sigma^2 estimated as 1.346:  log likelihood=-157.45
## AIC=324.89   AICc=325.53   BIC=337.92
```

```
#TSD11
tsd11<-read.delim("F:/TSD11.txt",header =FALSE,sep = ",")
tsd11<-tsd11[-c(1)]
tsd11<-as.matrix(tsd11)
ts.plot(tsd11)
```

```
## Warning in xy.coords(x = matrix(rep.int(tx, k), ncol = k), y = x, log = log, :
## NAs introduced by coercion
```

```
## Warning in xy.coords(x, y): NAs introduced by coercion
```




```
#to estimate p and q in arima(p,d,q) we require pacf,acf and eacf
#acf of tsd1
#acf(tsd11)
```

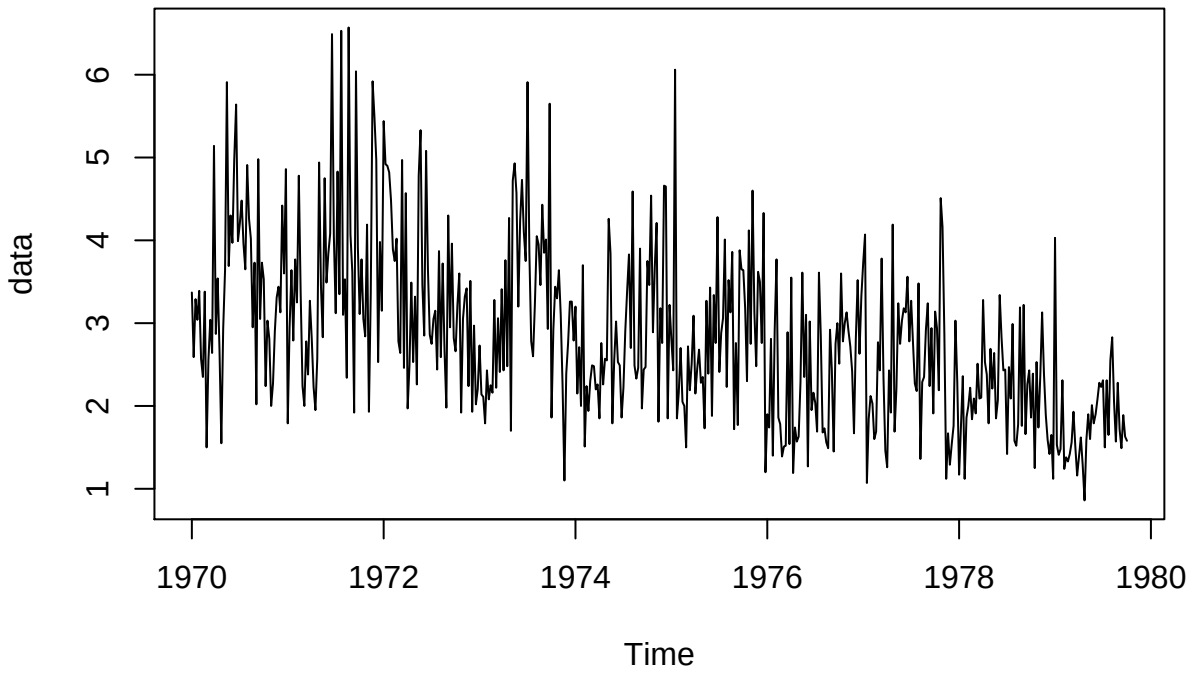
```
library('astsa')
```

```
## Warning: package 'astsa' was built under R version 3.6.3
```

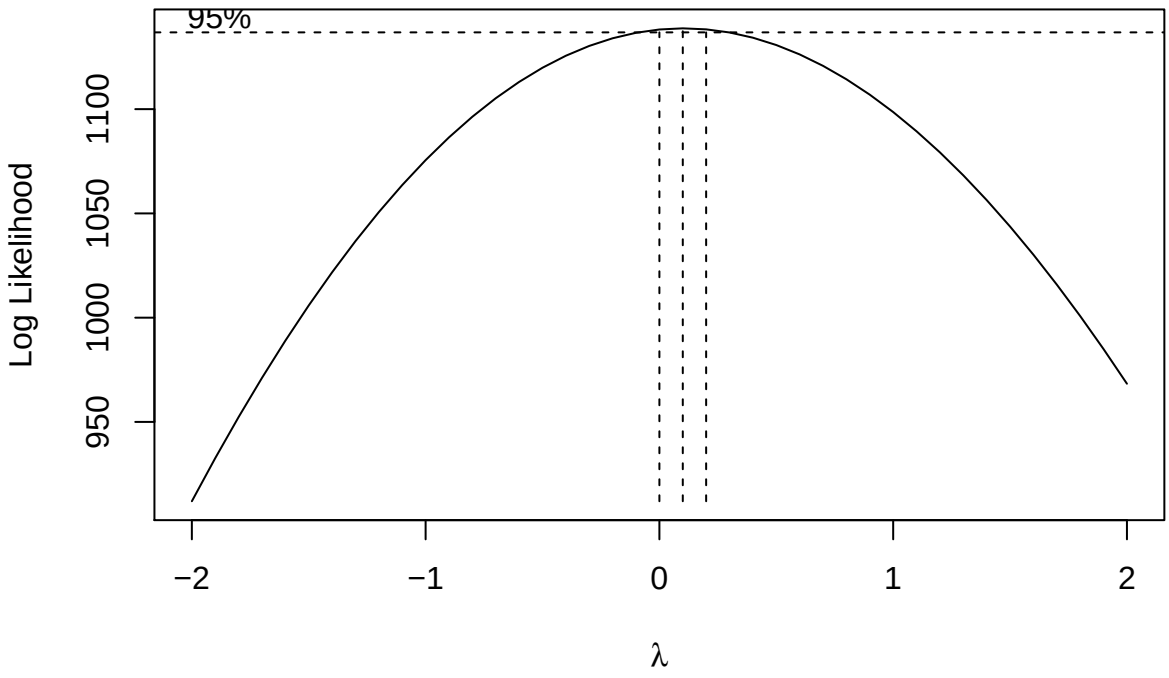
```
##
## Attaching package: 'astsa'
```

```
## The following object is masked from 'package:forecast':
##
##     gas
```

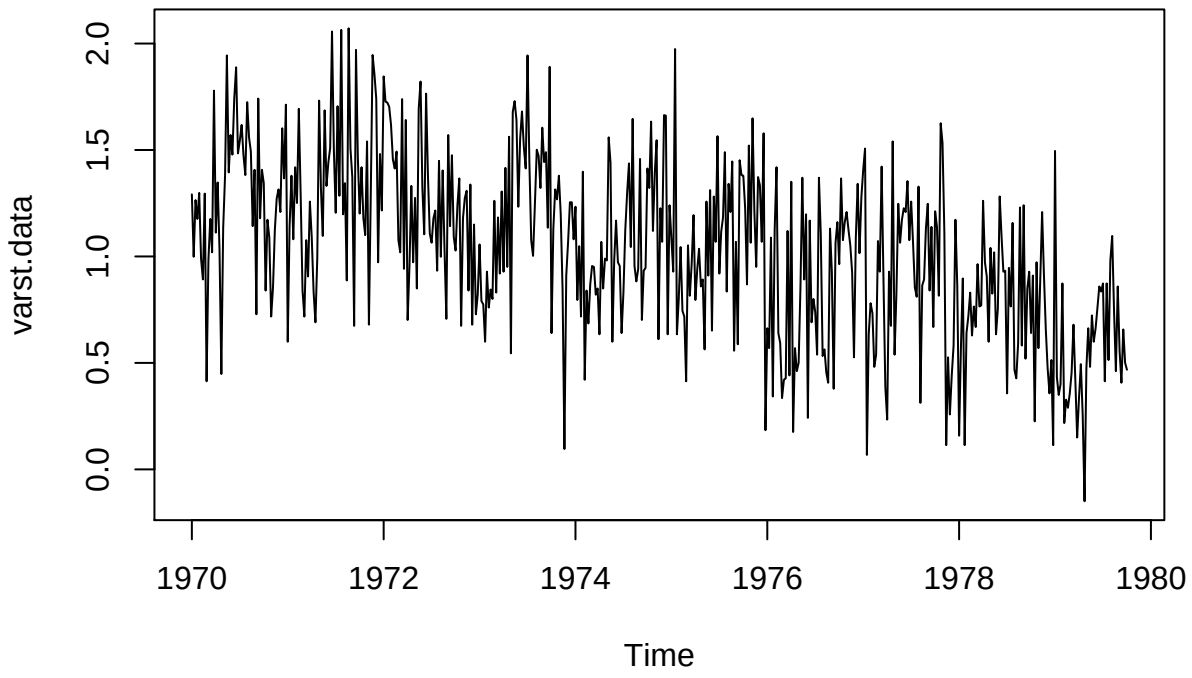
```r
library("TSA")
library("MASS")

data <-so2
ts.plot(data)
```



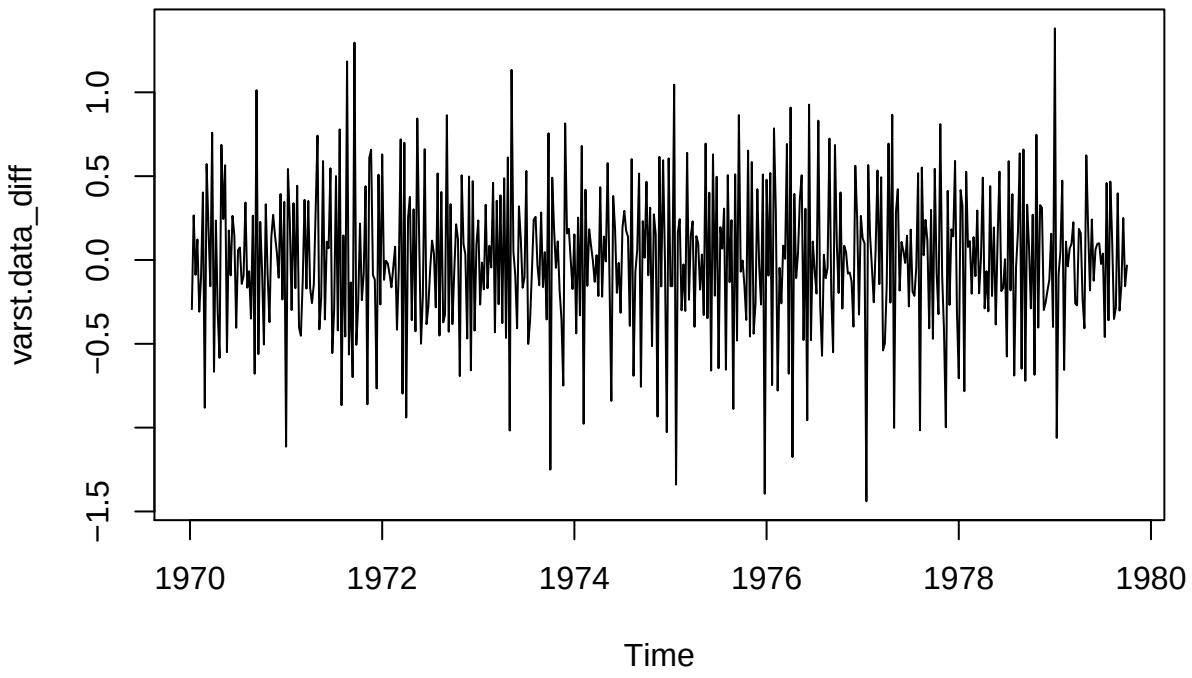

```r
library(forecast)
#we can see from the plot that there is not any variance stabilizing thing.
#lets Use box cox for variance stabilization.
boxcox <-BoxCox.ar(data)
```



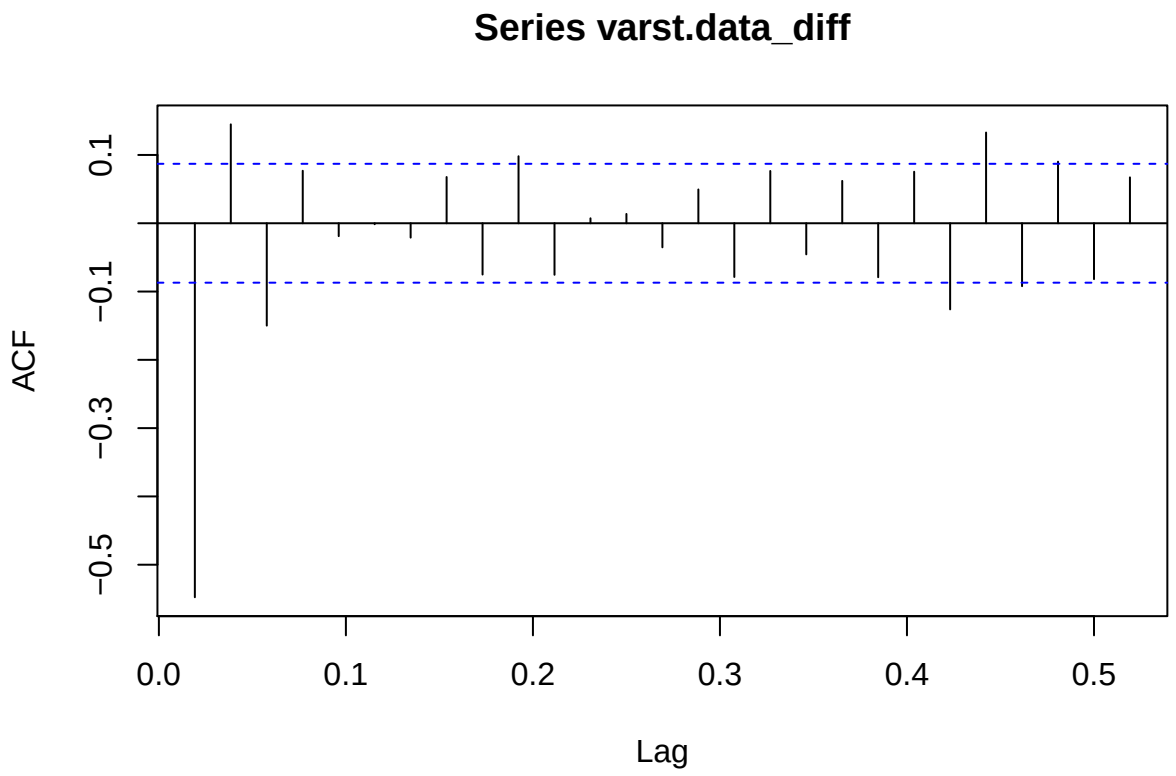

```
#we use that lambda as mle for vaiance stabilization using box cox transformation
varst.data <- BoxCox(data,lambda =boxcox $mle)
ts.plot(varst.data)
```



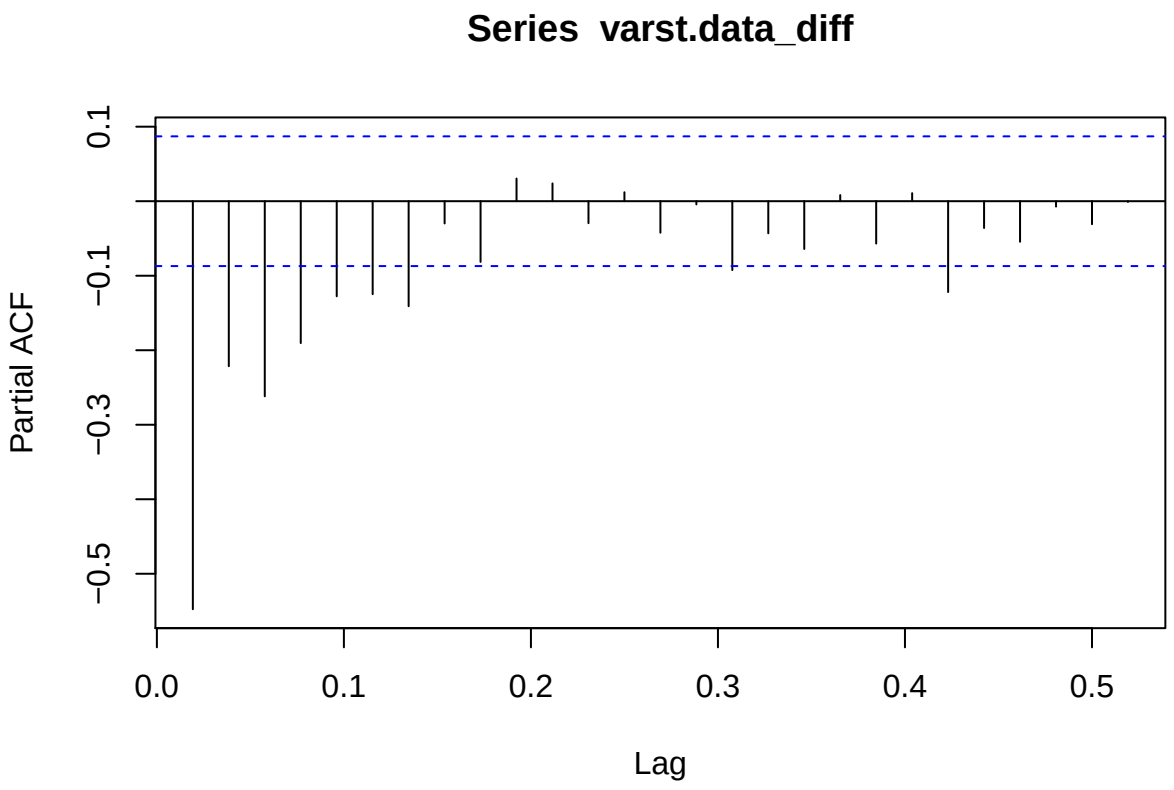

```
#we see variance is stabilized but we see that there is a linear trend in negative direction
varst.data_diff<-diff(varst.data)
ts.plot(varst.data_diff)
```




```
#we see that trend is removed after once differencing.
acf(varst.data_diff)
```




```
pacf(varst.data_diff)
```




```
eacf(varst.data_diff)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x o o o o o o x o  o  o  o
## 1 x x x o o o o o o o o  o  o  o
## 2 x x x o o o o o o o o  o  o  o
## 3 x o x o o o o o o o o  o  o  o
## 4 x x x x o o o o o o o  o  o  o
## 5 x x x x o x o o o o o  o  o  o
## 6 x o x x x x o o o o o  o  o  o
## 7 x x x x x x x o o o o  o  o  o
```

```
# from the Plot We can see that it is ARMA(1,3)Process and one unit root as we appliled 1 differnecing
model<-arima(varst.data,order = c(1,1,3))
model$x<-data
print(model)
```

```
##
## Call:
## arima(x = varst.data, order = c(1, 1, 3))
##
## Coefficients:
##           ar1      ma1      ma2      ma3
##       -0.5199  -0.3254  -0.3376  -0.1137
```

```
## s.e.   0.2360   0.2345   0.2114   0.0558
##
## sigma^2 estimated as 0.1134:  log likelihood = -168.31,  aic = 344.62
```

```
#forecast for 4 levels
forecast.data<-forecast(model,h=4,levels=95)
```

```
## Warning in forecast.Arima(model, h = 4, levels = 95): The non-existent levels
## arguments will be ignored.
```

```
print(forecast.data)
```

```
##          Point Forecast     Lo 80    Hi 80       Lo 95    Hi 95
## 1979.769      0.5868064 0.1551639 1.018449 -0.07333383 1.246947
## 1979.788      0.6025216 0.1657424 1.039301 -0.06547452 1.270518
## 1979.808      0.6151187 0.1645173 1.065720 -0.07401666 1.304254
## 1979.827      0.6085696 0.1562997 1.060840 -0.08311760 1.300257
```

```
#The 4 forecasts with confindece 95% are
#1
CI_1<-paste("The first forecast mean value",as.character(forecast.data$mean[1]),"with 95% Confidence","
print(CI_1)
```

```
## [1] "The first forecast mean value 0.58680639446961 with 95% Confidence ( 0.15516390401314 , 1.01844
```

```
CI_2<-paste("The second forecast mean value",as.character(forecast.data$mean[2]),"with 95% Confidence",
print(CI_2)
```

```
## [1] "The second forecast mean value 0.602521624107291 with 95% Confidence ( 0.16574242079633 , 1.039
```

```
CI_3<-paste("The third forecast mean value",as.character(forecast.data$mean[3]),"with 95% Confidence","
print(CI_3)
```

```
## [1] "The third forecast mean value 0.615118663038797 with 95% Confidence ( 0.16451729226916 , 1.0657
```

```
CI_4<-paste("The first forecast mean value",as.character(forecast.data$mean[4]),"with 95% Confidence","
print(CI_4)
```

```
## [1] "The first forecast mean value 0.608569624169606 with 95% Confidence ( 0.15629965782298 , 1.0608
```