time_series_1

Soumarya Basak

24/05/2022

Problem 1

Data

```
library(readr)
library(ggplot2)
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.1.3

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

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.1.3
```

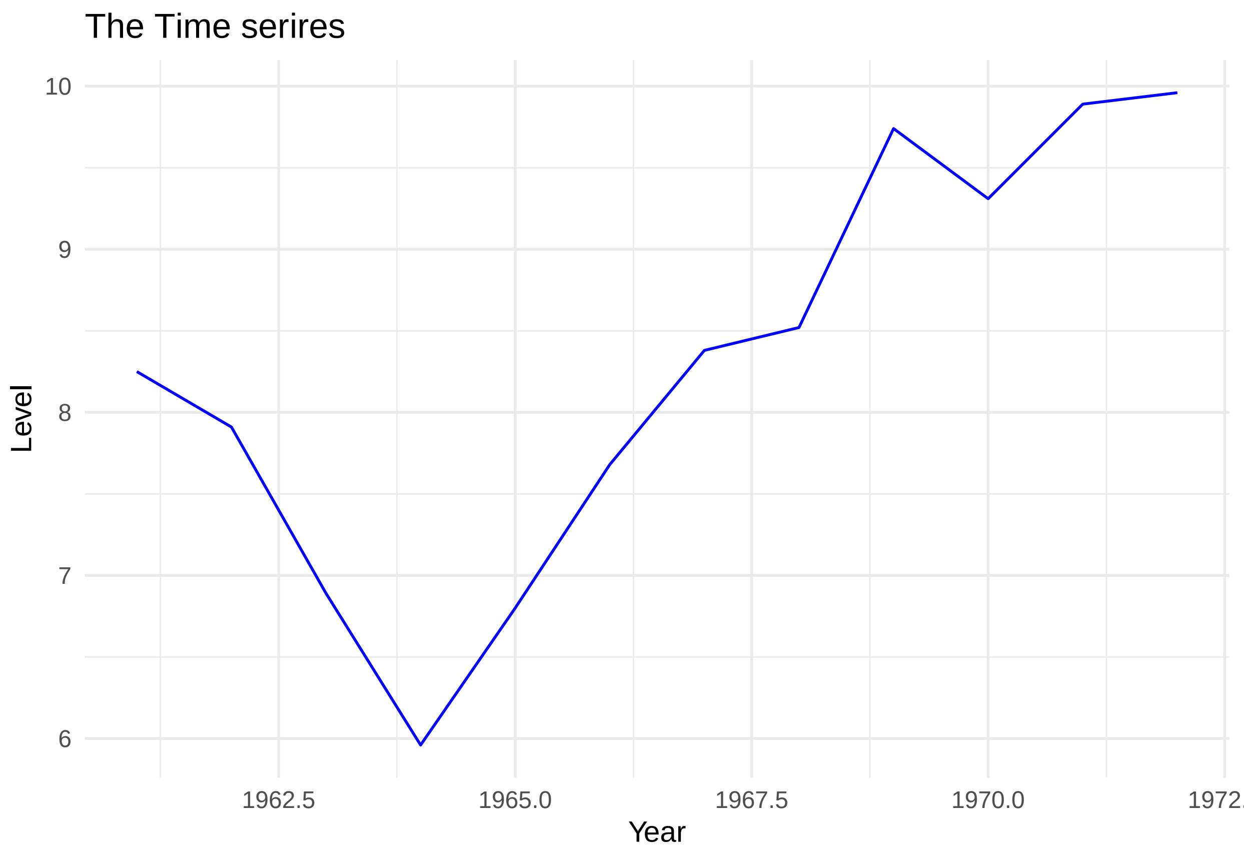
```
library(TSA)
```

```
## Warning: package 'TSA' was built under R version 4.1.3

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

##
## Attaching package: 'TSA'

## The following object is masked from 'package:readr':
##
##     spec

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

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

```
df<- read.csv("C:\\Users\\souma\\Dropbox\\Mstat_CU\\Sem 2\\Regression_analysis_1\\Data Sets\\ts_1.csv")
#df<- data.frame("Level"=df$Level,row.names = df$Year)
```

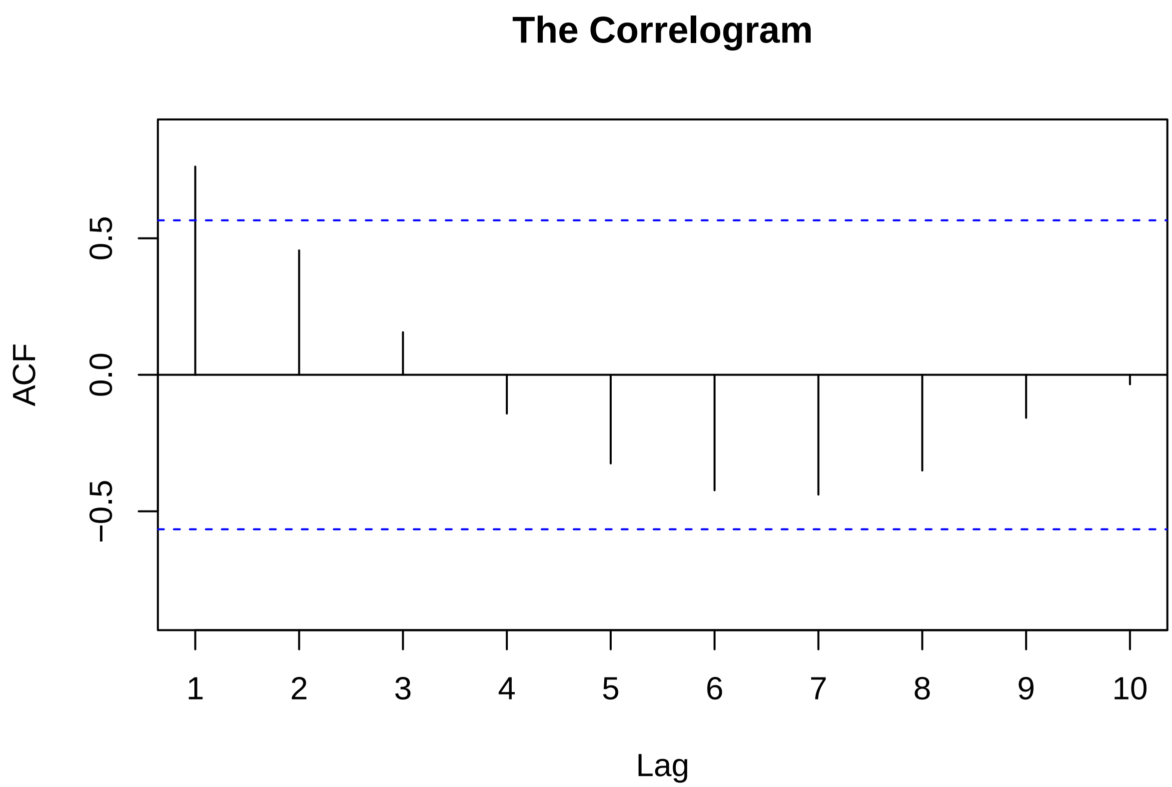
```
#plot(df$Level,type='l', main="The time series data", main)
ggplot(data=df)+
```

```
  geom_line(aes(Year,Level),col="blue")+
  ggtitle("The Time serires")+
  theme_minimal()
```

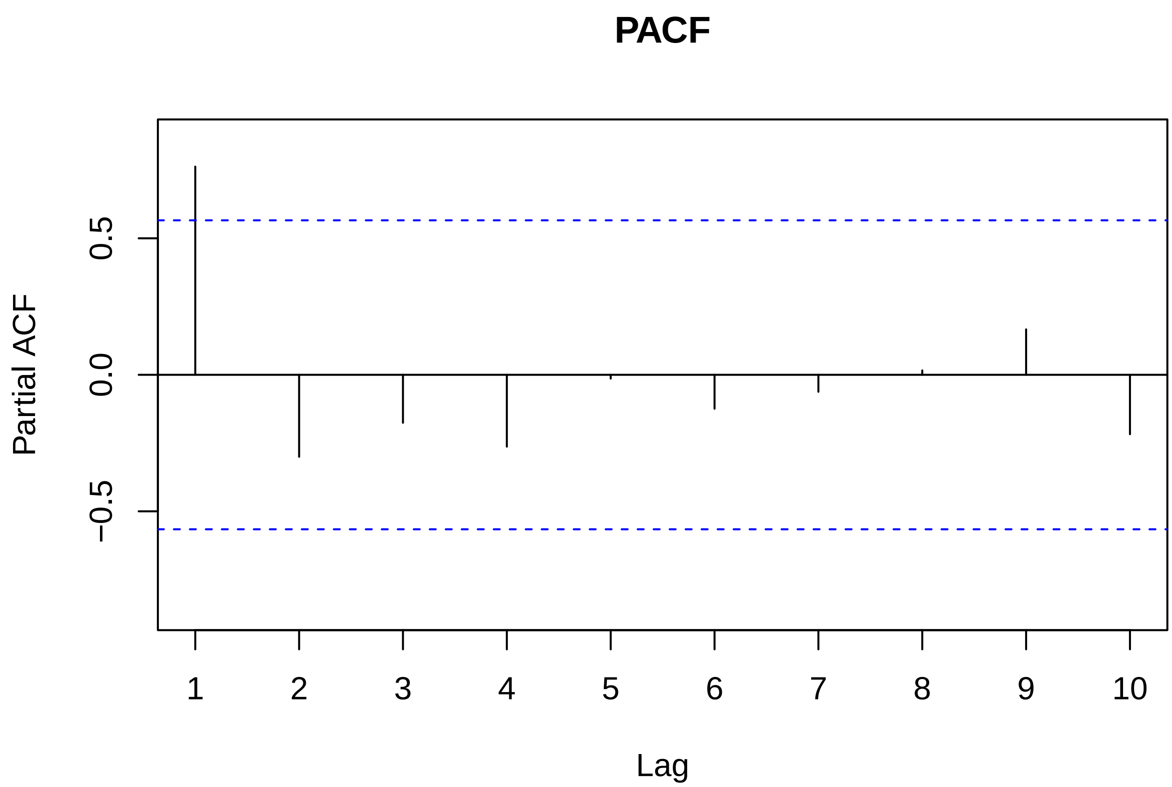


The Correlogram

```
Acf(ts(df$Level), main="The Correlogram")
```




```
Pacf(ts(df$Level),main="PACF")
```



Modeling

```
# ADF test for d
adf.test(df$Level,k=1)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  df$Level
## Dickey-Fuller = -3.5071, Lag order = 1, p-value = 0.0629
## alternative hypothesis: stationary
```

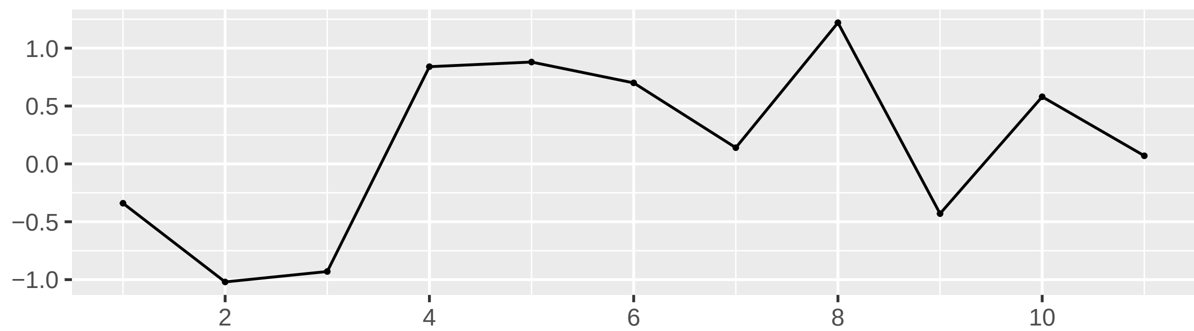
p value is more than 0.05 so the accept the null that the process is non-stationary

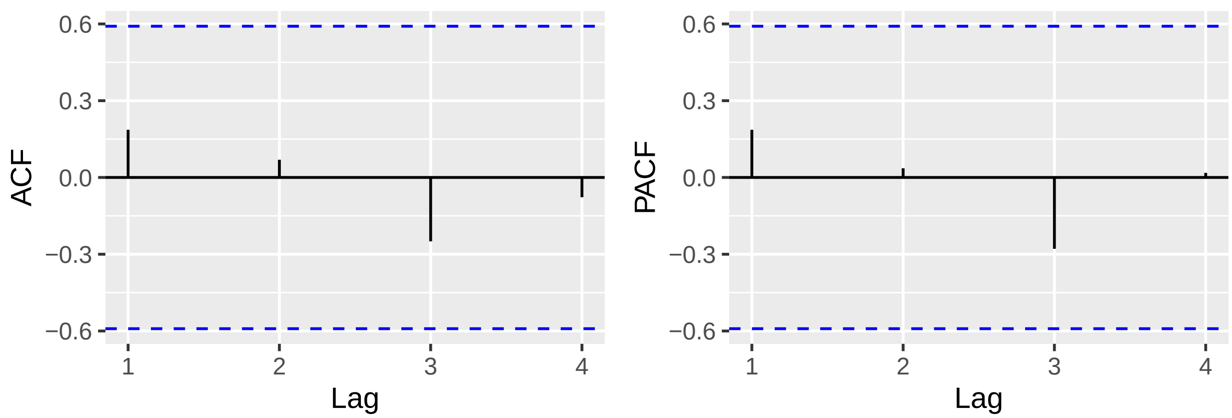
```
fst_diff<- diff(df$Level, differences = 1)

#adf test on first difference'
adf.test(fst_diff,k=1)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  fst_diff
## Dickey-Fuller = -1.5066, Lag order = 1, p-value = 0.7604
## alternative hypothesis: stationary
```

p value is larger than 0.05

```
# Acf and pacf plot of 1st diffreence
ggtsdisplay(fst_diff)
```



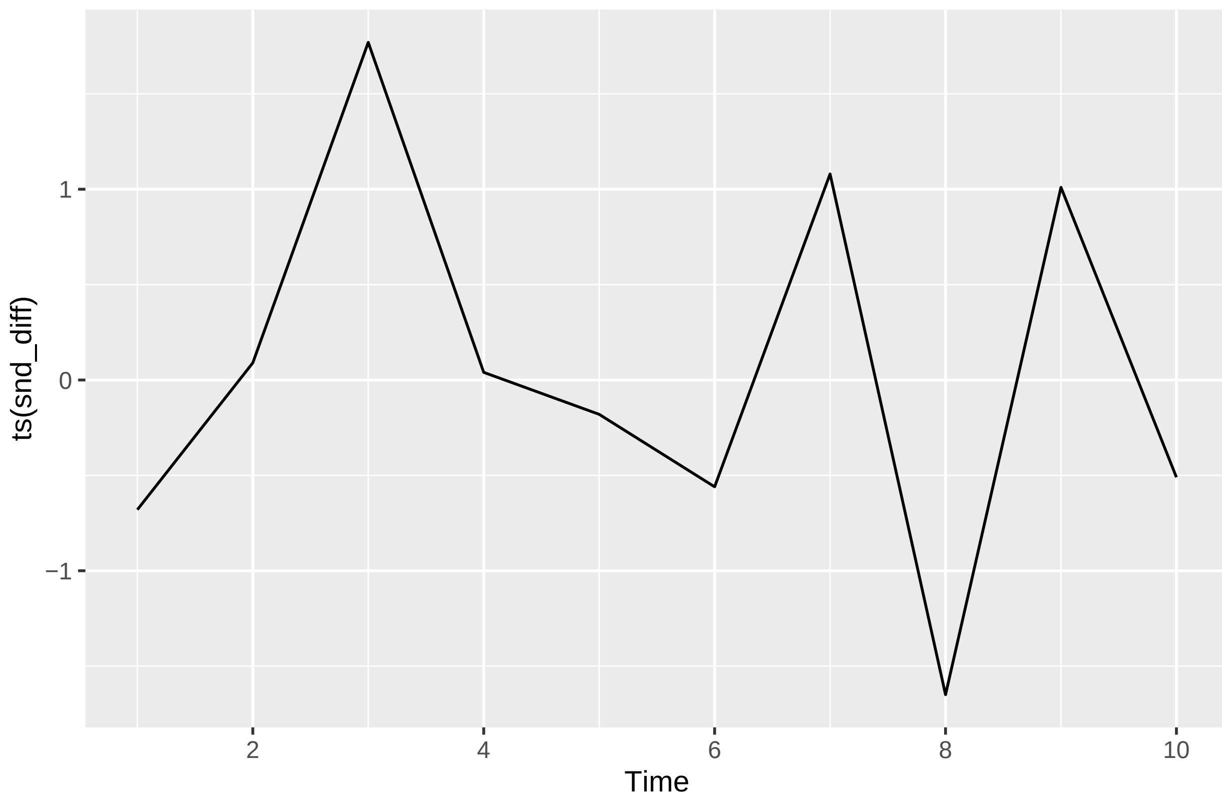



```
snd_diff<- diff(fst_diff,differences = 1)

adf.test(snd_diff,k=1)
```

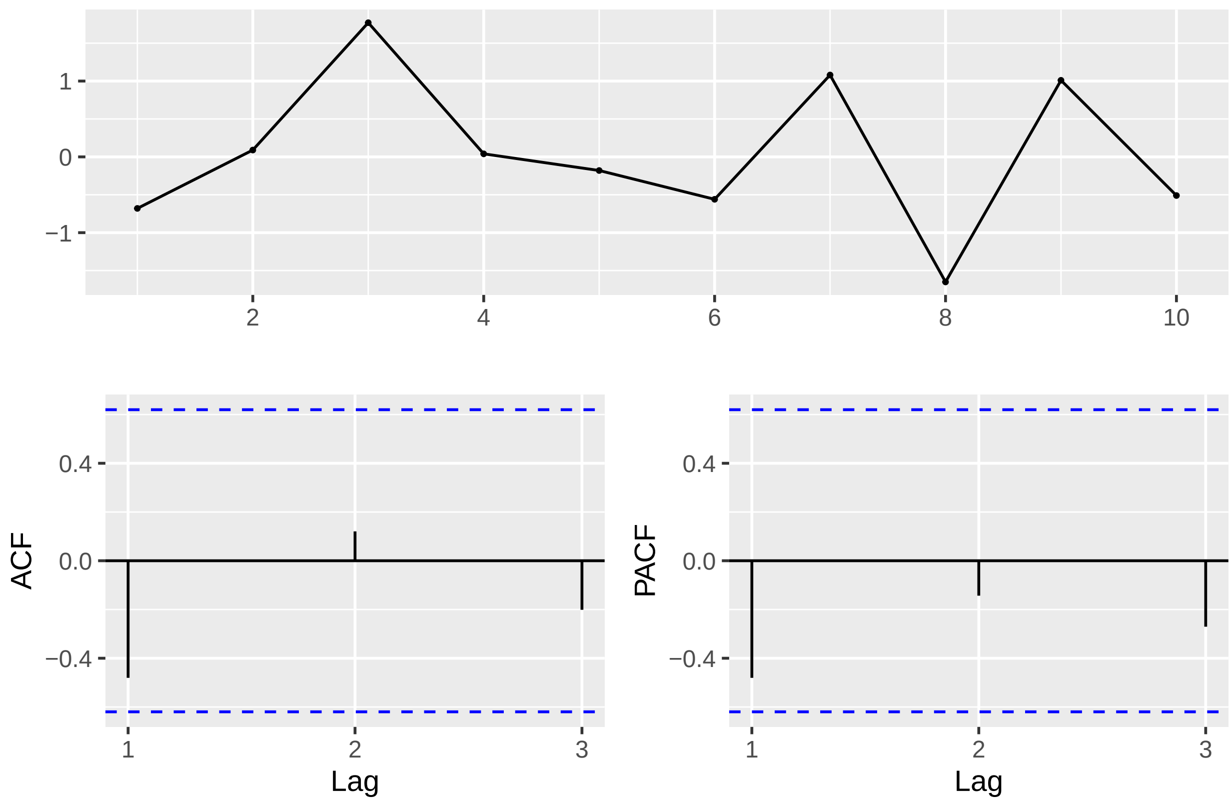
```
##
##  Augmented Dickey-Fuller Test
##
## data:  snd_diff
## Dickey-Fuller = -3.6427, Lag order = 1, p-value = 0.04695
## alternative hypothesis: stationary
```

note that, the p value is less than 0.05, so we reject the null hypothesis of non-stationary. So, after 2nd difference the series become stationary.

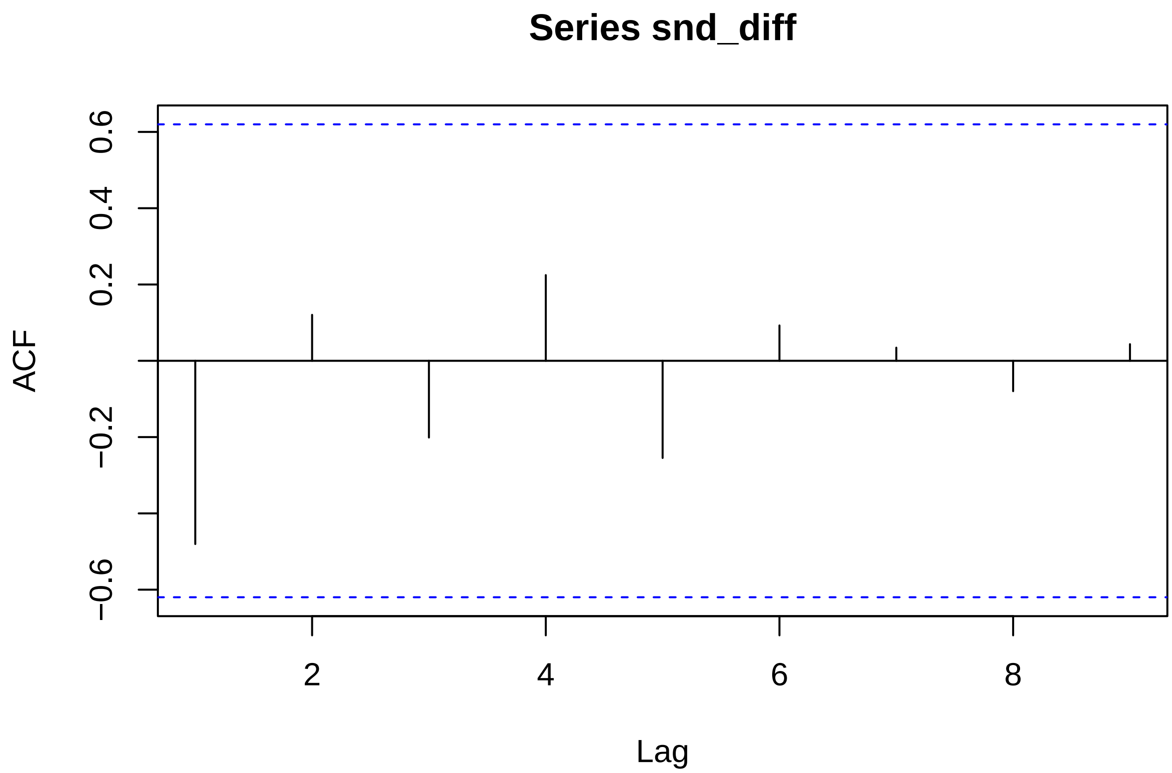
```
autoplot(ts(snd_diff))
```



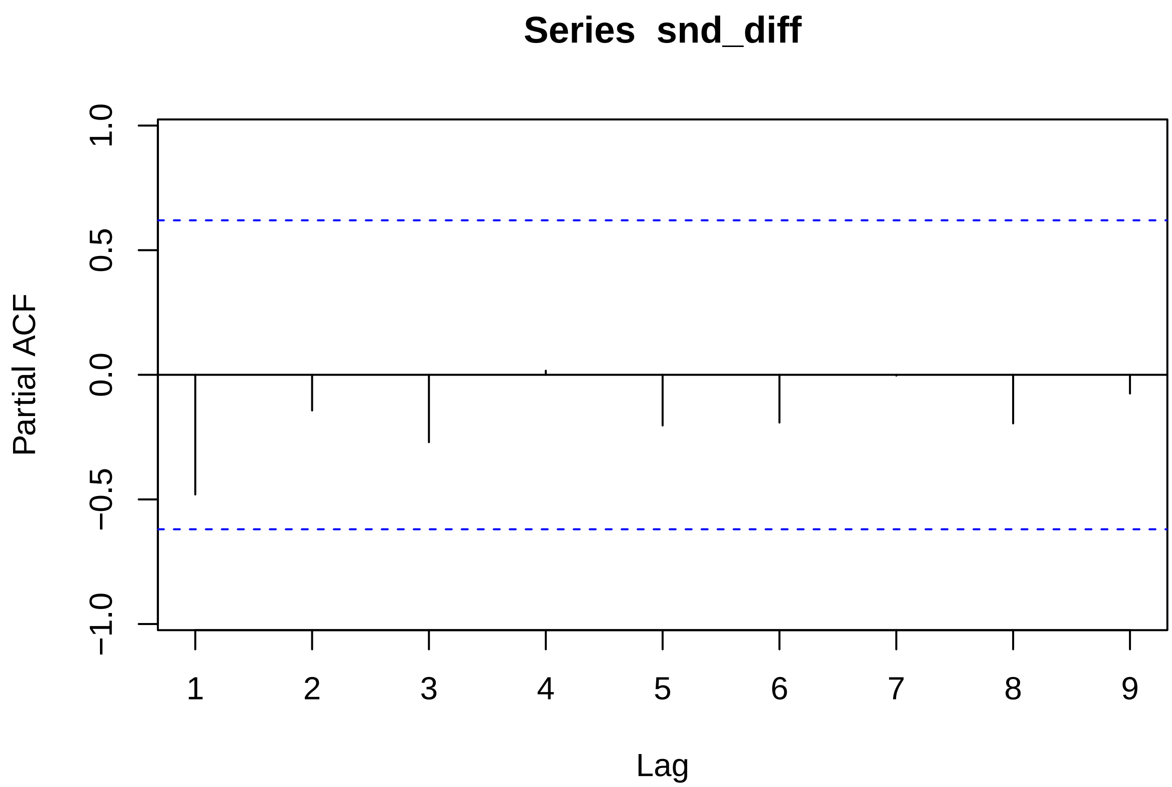

```
ggtsdisplay(ts(snd_diff))
```




```
acf(snd_diff)
```




```
Pacf(snd_diff)
```



MOdel 1

```
### very bad model
ts_mod<- Arima(y=ts(df$Level),order = c(0,2,0))
summary(ts_mod)
```

```
## Series: ts(df$Level)
## ARIMA(0,2,0)
##
## sigma^2 = 0.912:  log likelihood = -13.73
## AIC=29.46   AICc=29.96   BIC=29.76
##
## Training set error measures:
##                      ME      RMSE       MAE       MPE    MAPE      MASE
## Training set 0.0334884 0.8717919 0.6321265 0.6518871 7.58239 0.9725023
##                    ACF1
## Training set -0.475407
```

```
mod_1<- Arima(y=ts(df$Level),order=c(1,0,0))  # AR(1)
mod_2<- Arima(y=ts(df$Level),order=c(0,0,1))  # MA(1)
mod_3<- Arima(y=ts(df$Level),order=c(0,1,1))
mod_4<- Arima(y=ts(df$Level),order=c(1,1,0))
mod_5<- Arima(y=ts(df$Level),order=c(0,2,1))
mod_6<- Arima(y=ts(df$Level),order=c(1,2,0))
mod_7<- Arima(y=ts(df$Level),order=c(1,1,1))
mod_8<- Arima(y=ts(df$Level),order=c(1,2,1))
mod_9<- Arima(y=ts(df$Level),order=c(1,0,1))
```

```r
aic<- c(mod_1$aic,mod_2$aic,mod_3$aic,mod_4$aic,mod_5$aic,mod_6$aic,mod_7$aic,mod_8$aic,mod_9$aic)
model<- c(1:9)
cbind(model,aic)
```

```
##       model      aic
##  [1,]     1 32.60485
##  [2,]     2 37.33643
##  [3,]     3 28.14554
##  [4,]     4 28.02052
##  [5,]     5 28.59037
##  [6,]     6 28.95715
##  [7,]     7 29.97675
##  [8,]     8 30.44888
##  [9,]     9 33.82283
```

So, the 4th model has the least aic, so we will further go with the 4th model.

```r
#### to know the element of the model
names(ts_mod)
```

```
##  [1] "coef"      "sigma2"    "var.coef"  "mask"      "loglik"    "aic"
##  [7] "arma"      "residuals" "call"      "series"    "code"      "n.cond"
## [13] "nobs"      "model"     "aicc"      "bic"       "x"         "fitted"
```

```r
# the final model::::: ARIMA(1,1,0)
summary(mod_4)
```

```
## Series: ts(df$Level)
## ARIMA(1,1,0)
##
## Coefficients:
##          ar1
##       0.2179
## s.e.  0.2822
##
## sigma^2 = 0.5693:  log likelihood = -12.01
## AIC=28.02   AICc=29.52   BIC=28.82
##
## Training set error measures:
##                     ME      RMSE       MAE       MPE     MAPE      MASE
## Training set 0.1140929 0.6887984 0.5724599 0.9676481 7.298625 0.8807076
##                     ACF1
## Training set -0.03731791
```

The final model is

$$(Y_t - Y_{t-1}) = 0.2179 \times (Y_{t-1} - Y_{t-2})$$

```r
##### Model 2 ::::::::ARIMA(0,1,1)
summary(mod_3)
```

```
## Series: ts(df$Level)
## ARIMA(0,1,1)
##
## Coefficients:
##          ma1
##       0.1687
## s.e.  0.2412
```

```
##
## sigma^2 = 0.5769:  log likelihood = -12.07
## AIC=28.15   AICc=29.65   BIC=28.94
##
## Training set error measures:
##                     ME      RMSE       MAE      MPE     MAPE      MASE
## Training set 0.1223731 0.6933463 0.5802172 1.009252 7.406195 0.8926418
##                    ACF1
## Training set 0.01163234
```

```
### Model 3;;;;;;;;; ARIMA(0,2,1)
summary(mod_5)
```

```
## Series: ts(df$Level)
## ARIMA(0,2,1)
##
## Coefficients:
##          ma1
##      -0.5903
## s.e.   0.3515
##
## sigma^2 = 0.7288:  log likelihood = -12.3
## AIC=28.59   AICc=30.3   BIC=29.2
##
## Training set error measures:
##                     ME      RMSE       MAE      MPE     MAPE      MASE
## Training set 0.1419616 0.7393463 0.5674596 1.943772 7.104392 0.8730147
##                     ACF1
## Training set -0.06556938
```

```
### Model 4 :::: ARIMA (1,2,0)
summary(mod_6)
```

```
## Series: ts(df$Level)
## ARIMA(1,2,0)
##
## Coefficients:
##          ar1
##      -0.4639
## s.e.   0.2672
##
## sigma^2 = 0.7703:  log likelihood = -12.48
## AIC=28.96   AICc=30.67   BIC=29.56
##
## Training set error measures:
##                      ME     RMSE       MAE      MPE     MAPE      MASE
## Training set 0.07552466 0.760066 0.5480266 1.156868 6.975541 0.8431178
##                     ACF1
## Training set -0.08430298
```

Test for randomness of residuals

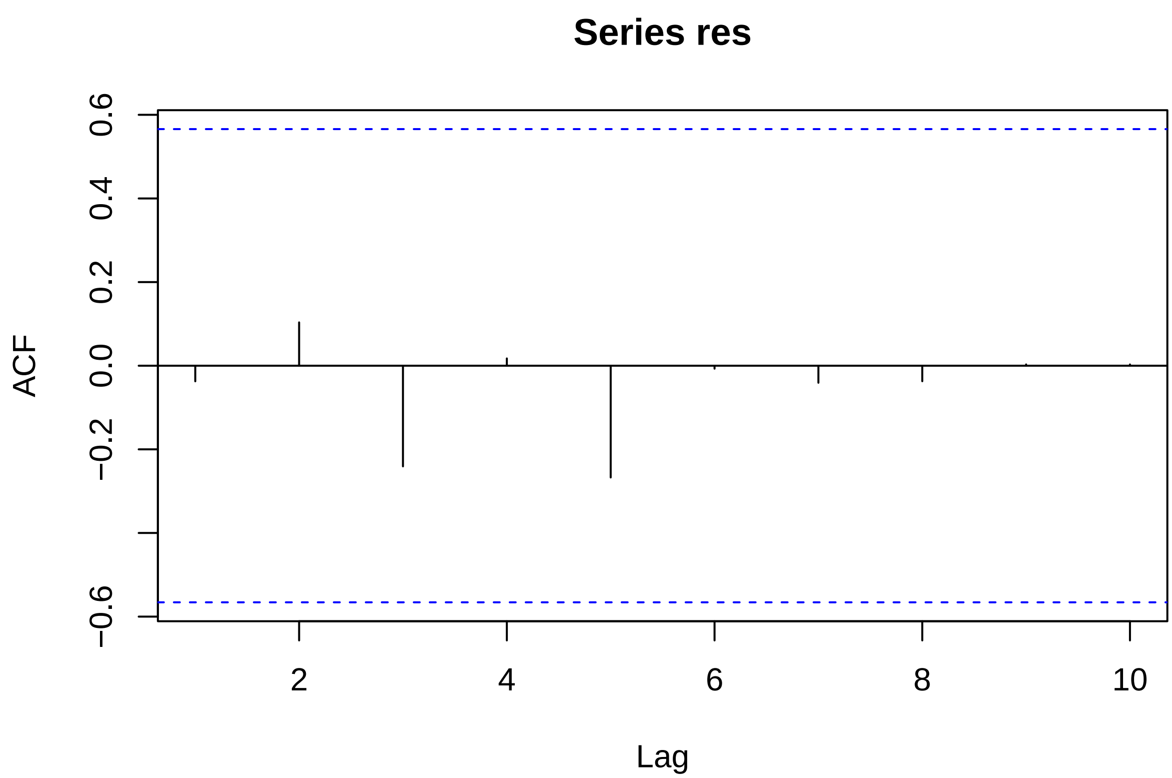
note that we sticking with ARIMA(1,1,0) :`mod_4`.

Portmanteau test

```
res<- mod_4$residuals
res
```

```
## Time Series:
## Start = 1
## End = 12
## Frequency = 1
##  [1]  0.008249996 -0.331834578 -0.945925439 -0.707776317  1.042615711
##  [6]  0.696992261  0.508277607 -0.012506449  1.189498710 -0.695796954
## [11]  0.673682533 -0.056362486
```

```
acf(res)
```




```
acf(res,plot=FALSE)
```

```
##
## Autocorrelations of series 'res', by lag
##
##      1      2      3      4      5      6      7      8      9     10
## -0.037  0.104 -0.241  0.017 -0.267 -0.007 -0.041 -0.037  0.003  0.003
```

The acf plot of the residuals shows that there is a randomness in the residuals.
But we have to test statistically.

Portmanteau test

```
Box.test(res,lag = 11, type="Box-Pierce",fitdf = 1)
```

```
##
##  Box-Pierce test
##
## data:  res
## X-squared = 1.7397, df = 10, p-value = 0.998
```

Ljung Box test

```
Box.test(res,lag=11,type = "Ljung-Box",fitdf =1)
```

```
##
##  Box-Ljung test
##
## data:  res
## X-squared = 3.1233, df = 10, p-value = 0.9784
```

p value > 0.05 so, we will accept that the residuals are random.

Forecast

```
forecast(mod_4,h=3) #h=lag
```

```
##    Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 13       9.975251 9.008268 10.94223 8.496378 11.45412
## 14       9.978573 8.454785 11.50236 7.648141 12.30901
## 15       9.979297 8.025067 11.93353 6.990560 12.96803
```