INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
DHARWAD

Lab Report

Course: Data Analytics in R (CS6E23L)

Course Instructor: Dr. Kavi Mahesh

Lab Instructor: Pragya Verma

By:

Paka Sravan Kumar Yadav

6^{th} Semester

3rd Year

16CS11

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
DHARWAD

Lab – 03 (3rd Feb 2019)

Time Series Analysis

```
library(timeSeries)

## Warning: package 'timeSeries' was built under R version 3.5.3

## Loading required package: timeDate

library(xts)

## Warning: package 'xts' was built under R version 3.5.3

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 3.5.3

##
## Attaching package: 'zoo'

## The following object is masked from 'package:timeSeries':
##
##     time<-

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

library(zoo)
library(forecast)

## Warning: package 'forecast' was built under R version 3.5.3

library(fUnitRoots)

## Warning: package 'fUnitRoots' was built under R version 3.5.3

## Loading required package: fBasics

## Warning: package 'fBasics' was built under R version 3.5.3

library(lmtest)
library(FitAR)

## Warning: package 'FitAR' was built under R version 3.5.3

## Loading required package: lattice

## Loading required package: leaps
```

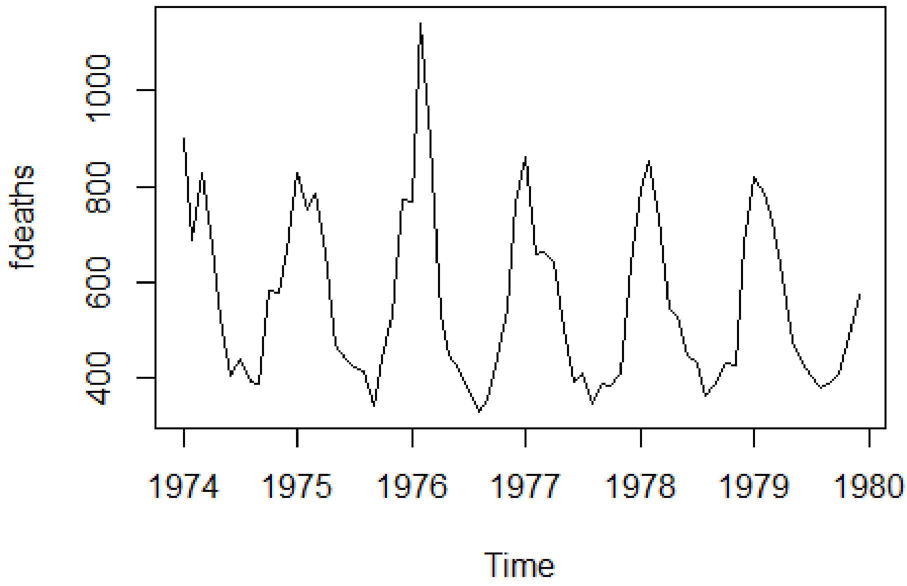
```
## Loading required package: ltsa

## Loading required package: bestglm

## Warning: package 'bestglm' was built under R version 3.5.3

##
## Attaching package: 'FitAR'

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

1. Build a timeSeries object with the data.

```
data("fdeaths")

## Warning in data("fdeaths"): data set 'fdeaths' not found

fdeaths

##      Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
## 1974  901  689  827  677  522  406  441  393  387  582  578  666
## 1975  830  752  785  664  467  438  421  412  343  440  531  771
## 1976  767 1141  896  532  447  420  376  330  357  445  546  764
## 1977  862  660  663  643  502  392  411  348  387  385  411  638
## 1978  796  853  737  546  530  446  431  362  387  430  425  679
## 1979  821  785  727  612  478  429  405  379  393  411  487  574
```

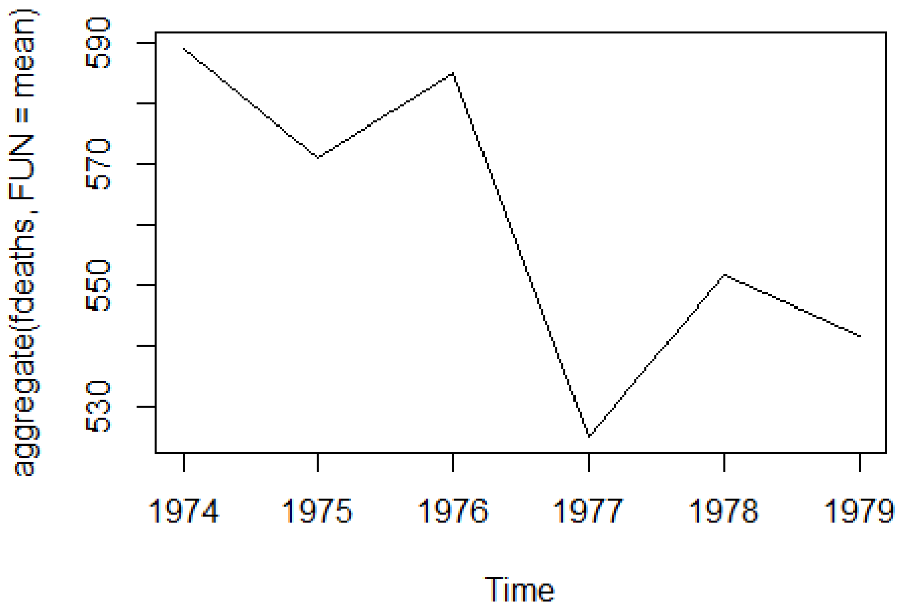
2. Plot the yearly (or other suitable periodic) mean values

```
plot(fdeaths)
```



3. **Plot the monthly (or other suitable periodic) boxplots**

```
plot(aggregate(fdeaths, FUN = mean))
```



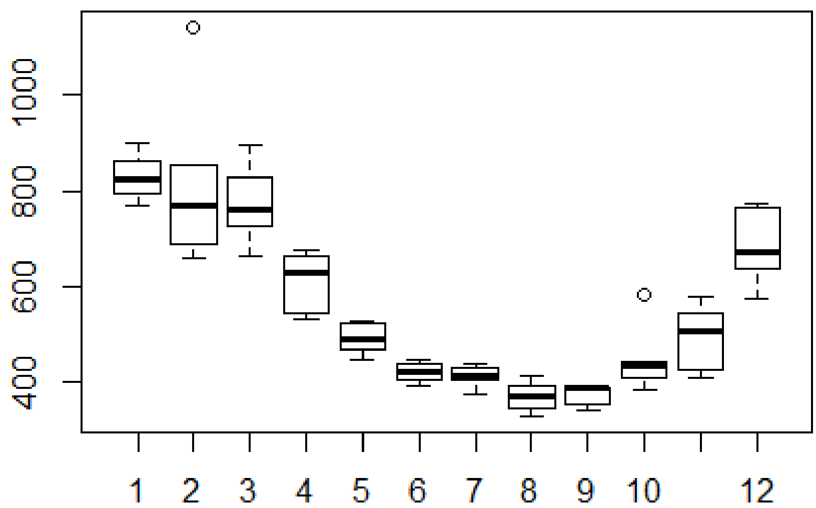

```
class(fdeaths)
## [1] "ts"
frequency(fdeaths)
```

```
## [1] 12
```

```r
cycle(fdeaths)
```

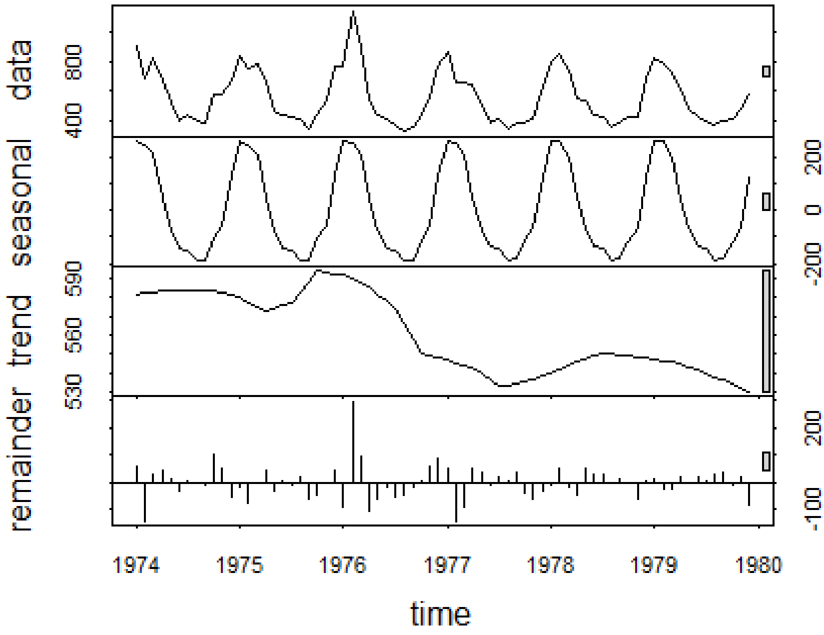
```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 1974   1   2   3   4   5   6   7   8   9  10  11  12
## 1975   1   2   3   4   5   6   7   8   9  10  11  12
## 1976   1   2   3   4   5   6   7   8   9  10  11  12
## 1977   1   2   3   4   5   6   7   8   9  10  11  12
## 1978   1   2   3   4   5   6   7   8   9  10  11  12
## 1979   1   2   3   4   5   6   7   8   9  10  11  12
```

```r
boxplot(fdeaths ~ cycle(fdeaths))
```



4. Decompose the time series using the stl function. What type of trend does it show

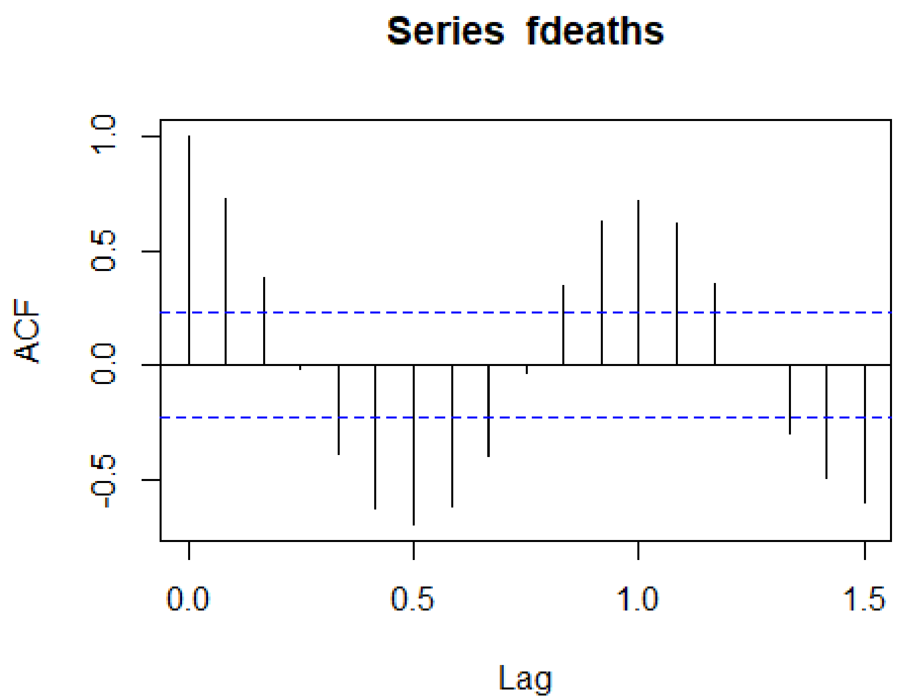
```r
t1 <- stl(fdeaths, s.window = 12)
plot(t1)
```



5. **What type of seasonality?**

   Yearly

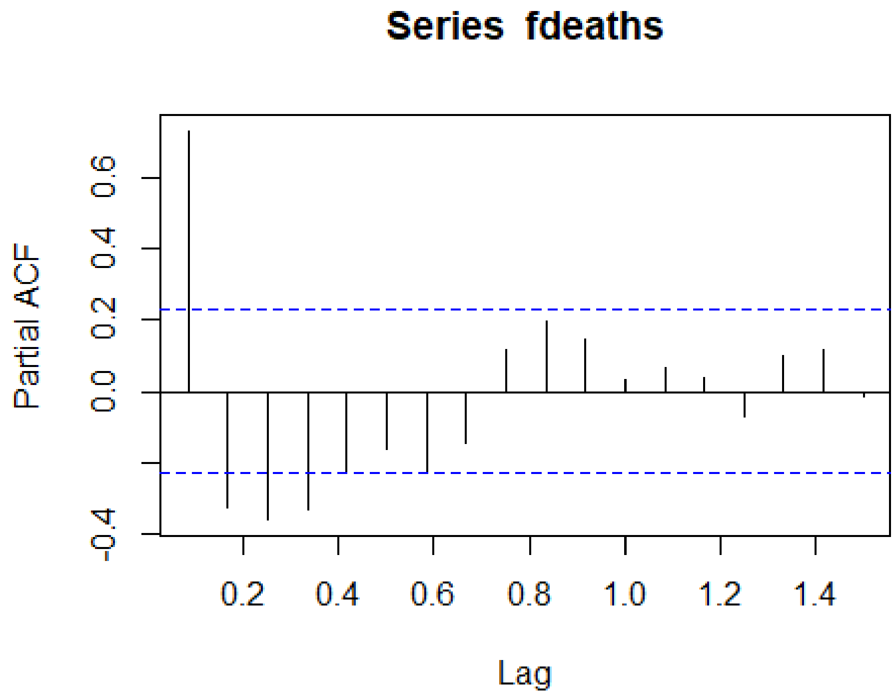
```
acf(fdeaths)
```




```
pacf(fdeaths)
```

6. How is the residue after you remove trend and seasonality?

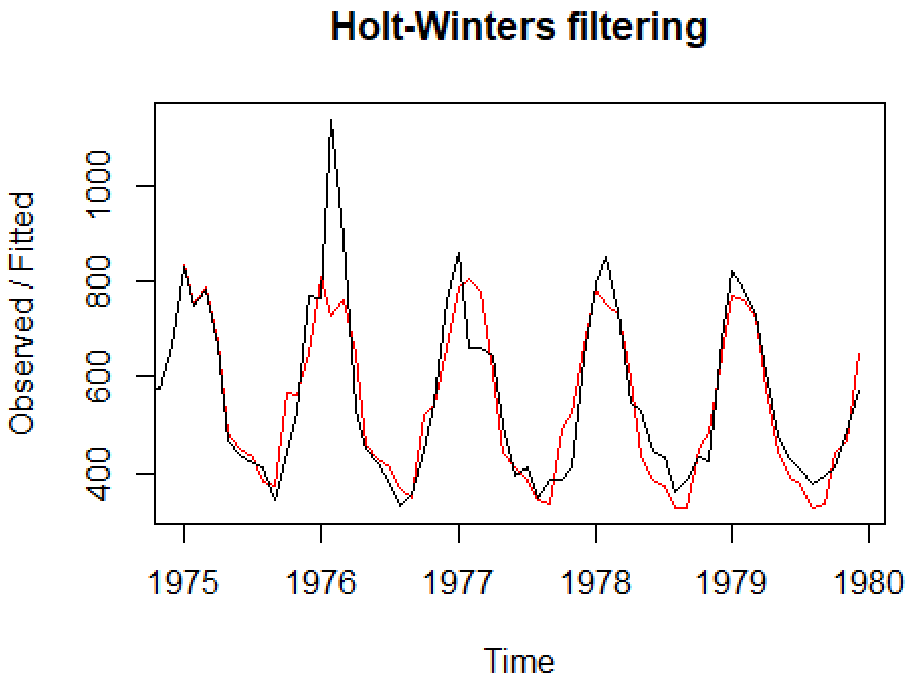
   Noise.



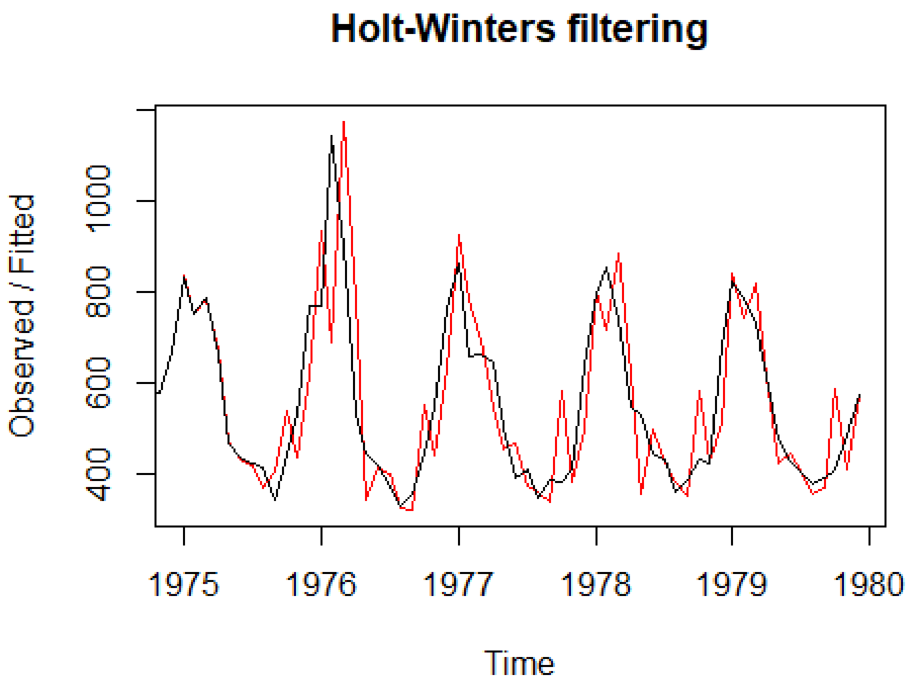

```
Box.test(fdeaths)

##
## Box-Pierce test
##
## data: fdeaths
## X-squared = 38.318, df = 1, p-value = 6.009e-10

head(fdeaths)

##      Jan Feb Mar Apr May Jun
## 1974 901 689 827 677 522 406

tail(fdeaths)

##      Jul Aug Sep Oct Nov Dec
## 1979 405 379 393 411 487 574
```

7. **Build a model of the data using the HoltWinters method for the period u pto about 75% of the data (e.g., up to December 2015 if it were for the CO2 data set). Use suitable values of alpha, beta and gamma.**

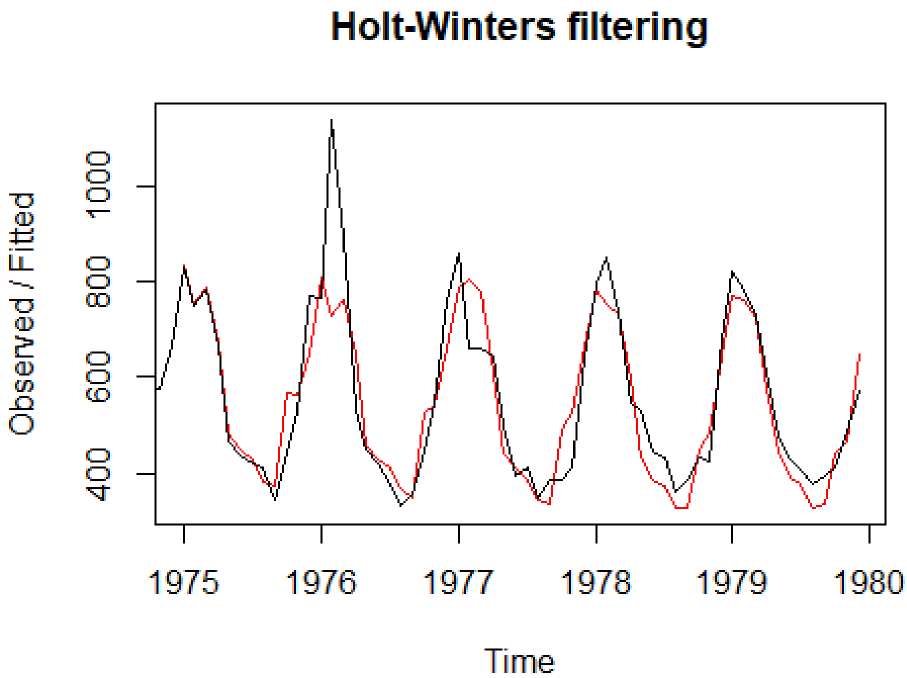
```
hw = HoltWinters(fdeaths)
plot(hw)
```



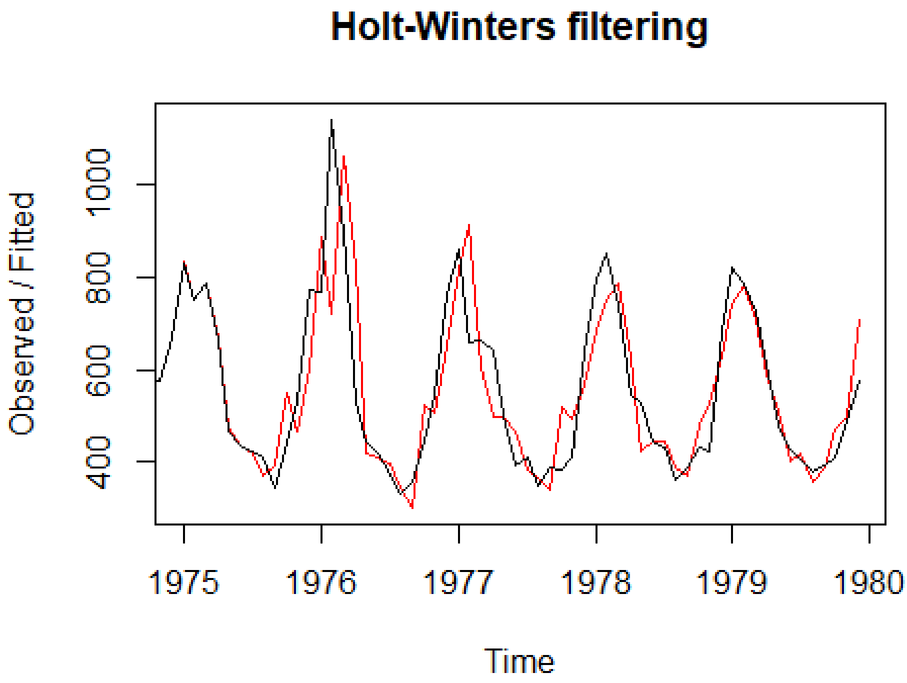

```
hw = HoltWinters(fdeaths, alpha = 1)
plot(hw)
```



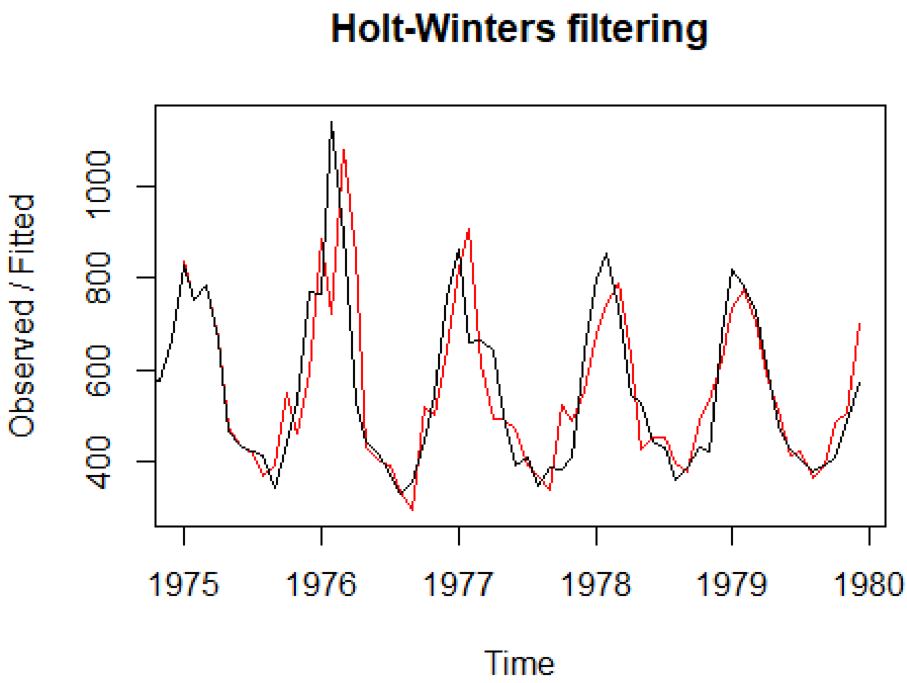

```r
hw = HoltWinters(fdeaths, beta = 1)
plot(hw)
```



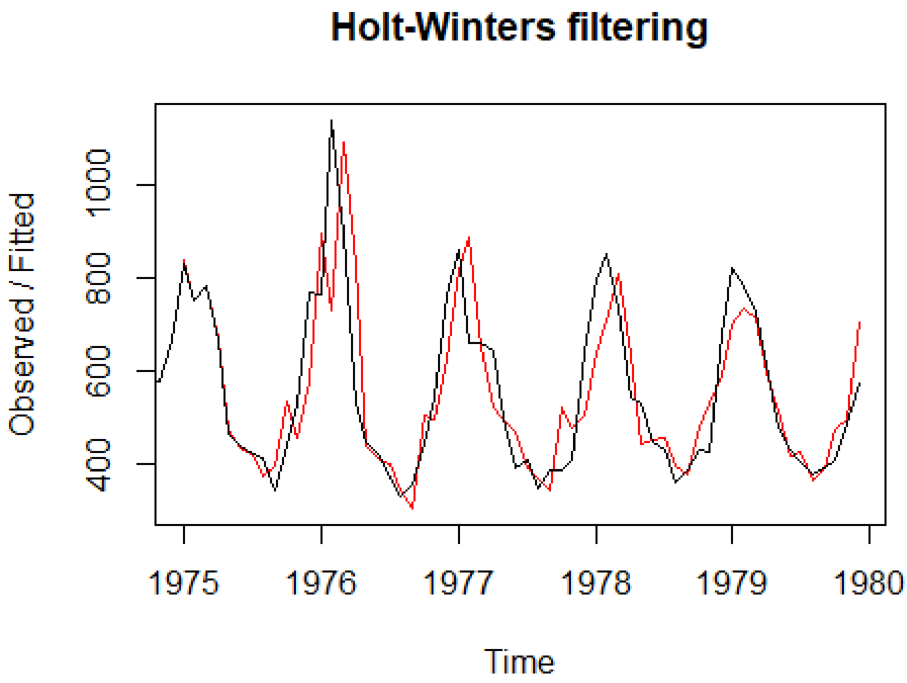
```
hw = HoltWinters(fdeaths, gamma = 1)
plot(hw)
```




```
hw = HoltWinters(fdeaths, alpha = 0.7 ,beta = 0.1, gamma = 0.8)
plot(hw)
```




```
hw = HoltWinters(fdeaths, alpha = 0.7 ,beta = 0.1, gamma = 0.8, seasonal = '
multiplicative')
plot(hw)
```



```
rollmean(fdeaths, 2)

##        Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct
## 1974  795.0  758.0  752.0  599.5  464.0  423.5  417.0  390.0  484.5  58
0.0
## 1975  791.0  768.5  724.5  565.5  452.5  429.5  416.5  377.5  391.5  48
5.5
## 1976  954.0 1018.5  714.0  489.5  433.5  398.0  353.0  343.5  401.0  4
95.5
## 1977  761.0  661.5  653.0  572.5  447.0  401.5  379.5  367.5  386.0  39
8.0
## 1978  824.5  795.0  641.5  538.0  488.0  438.5  396.5  374.5  408.5  42
7.5
## 1979  803.0  756.0  669.5  545.0  453.5  417.0  392.0  386.0  402.0  44
9.0
##        Nov    Dec
## 1974  622.0  748.0
## 1975  651.0  769.0
## 1976  655.0  813.0
## 1977  524.5  717.0
## 1978  552.0  750.0
## 1979  530.5

#HOLTWINTER'S METHOD

#Since time series data is atomic, it cannot be divided.
#So library Zoo is used to convert the time series data, divided and again conv
erted to timeseries
```

8. **Build a model of the data using the HoltWinters method for the period upto about 75% of the data (e.g., up to December 2015 if it were for the CO2 data set). Use suitable values of alpha, beta and gamma.**

```
fz = as.zoo(fdeaths)
length(fdeaths)

## [1] 72

fz_75 = fz[1:54]
fz_25 = fz[55:72]

fdeaths_75 = as.ts(fz_75)
fdeaths_25 = as.ts(fz_25)

fdeaths_25
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 1978                     431 362 387 430 425 679
## 1979 821 785 727 612 478 429 405 379 393 411 487 574
```

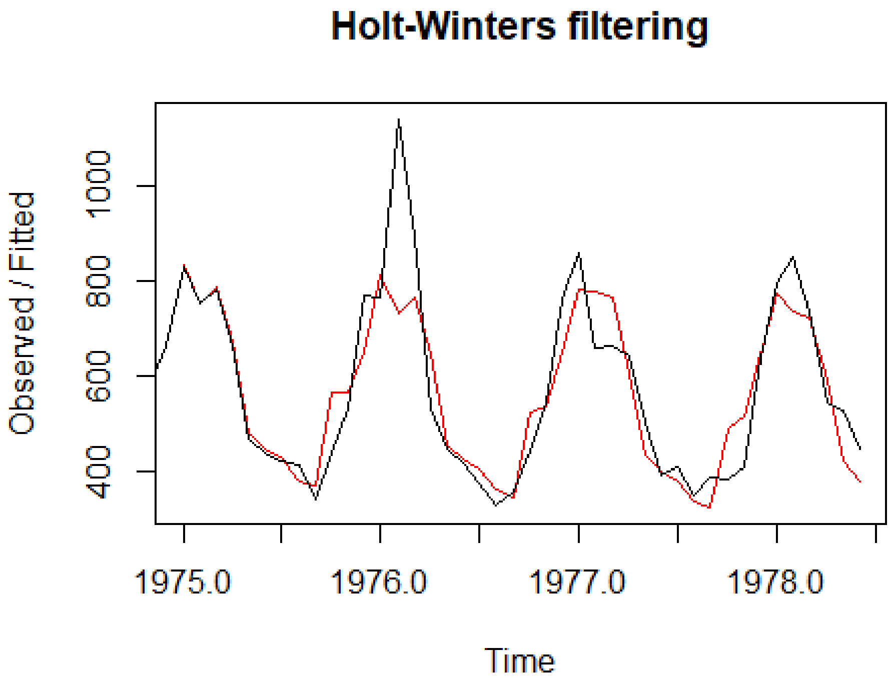
```
fdeaths_75
```

```
##       Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
## 1974  901  689  827  677  522  406  441  393  387  582  578  666
## 1975  830  752  785  664  467  438  421  412  343  440  531  771
## 1976  767 1141  896  532  447  420  376  330  357  445  546  764
## 1977  862  660  663  643  502  392  411  348  387  385  411  638
## 1978  796  853  737  546  530  446
```

```
hw_75 = HoltWinters(fdeaths_75, alpha = NULL, beta = NULL, gamma = 0.1
699511)
hw_75
```

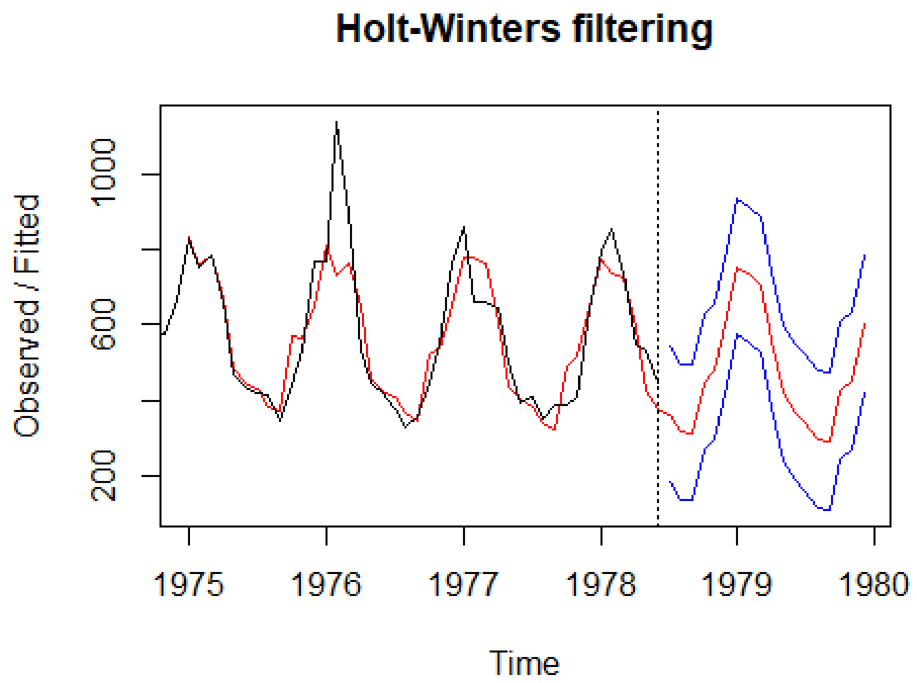
```
## Holt-Winters exponential smoothing with trend and additive seasonal com
ponent.
##
## Call:
## HoltWinters(x = fdeaths_75, alpha = NULL, beta = NULL, gamma = 0.16
99511)
##
## Smoothing parameters:
##  alpha: 0
##  beta : 0
##  gamma: 0.1699511
##
## Coefficients:
##           [,1]
## a    513.530303
## b     -1.850379
## s1  -149.245252
## s2  -193.962954
## s3  -194.745018
## s4   -57.692010
## s5   -28.245002
## s6   120.855387
## s7   254.814095
## s8   236.182973
## s9   208.542830
## s10   66.091368
## s11  -74.423026
## s12 -124.137802
```

```
plot(hw_75)
```

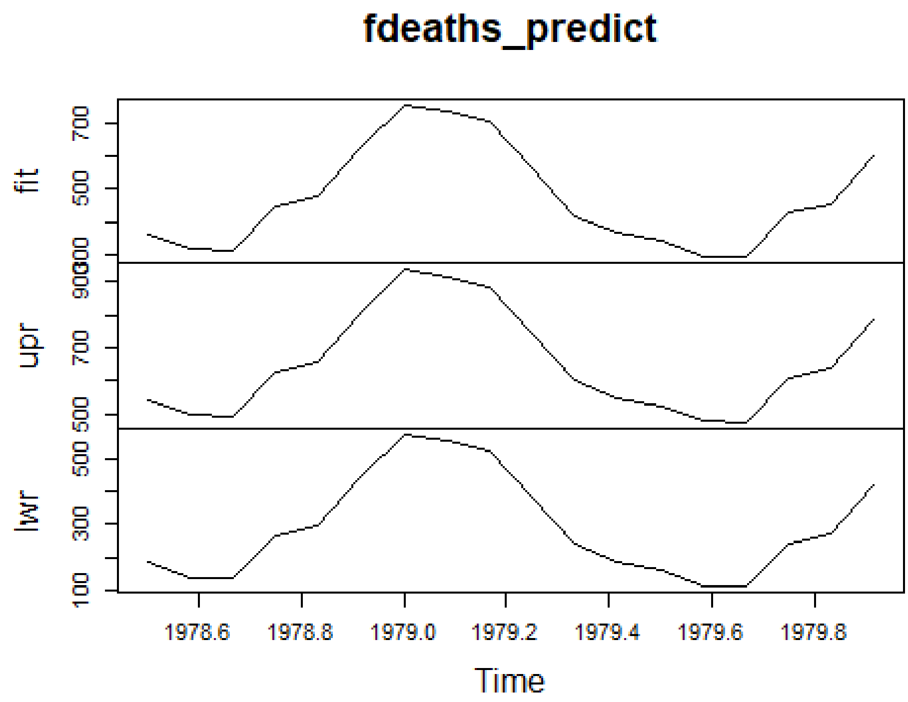


9. **Predict the values for the next 25% of the time (e.g., for the CO2 data set, all of 2016 and the first 3 months of 2017).**

```
fdeaths_predict = predict(hw_75, 18, prediction.interval = TRUE)
plot(hw_75, fdeaths_predict)
```




```
predict_zoo = as.zoo(fdeaths_predict)

predict_fitted <- fdeaths_predict[, 'fit']

class(predict_fitted)

## [1] "ts"

plot(fdeaths_predict)
```




```
fitted <- matrix(predict_fitted, ncol = 12, byrow = FALSE)

## Warning in matrix(predict_fitted, ncol = 12, byrow = FALSE): data length
## [18] is not a sub-multiple or multiple of the number of columns [12]

#fitted

matrix_fd_25 <- as.matrix(fz_25)
#matrix_df_25
actual <- matrix(matrix_fd_25, ncol = 12, byrow = FALSE)

## Warning in matrix(matrix_fd_25, ncol = 12, byrow = FALSE): data length
[18]
## is not a sub-multiple or multiple of the number of columns [12]

#actual
```

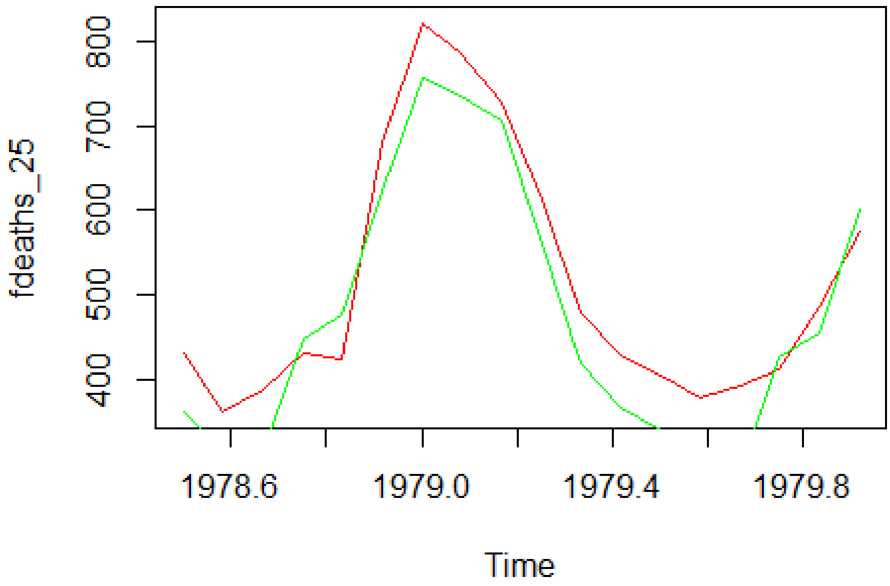
10. **Compute the rms error between the predicted and actual values.**

```
RMSE = function(fitted, actual){
  sqrt(mean((fitted - actual)^2))
}

RMSE(fitted, actual)
```

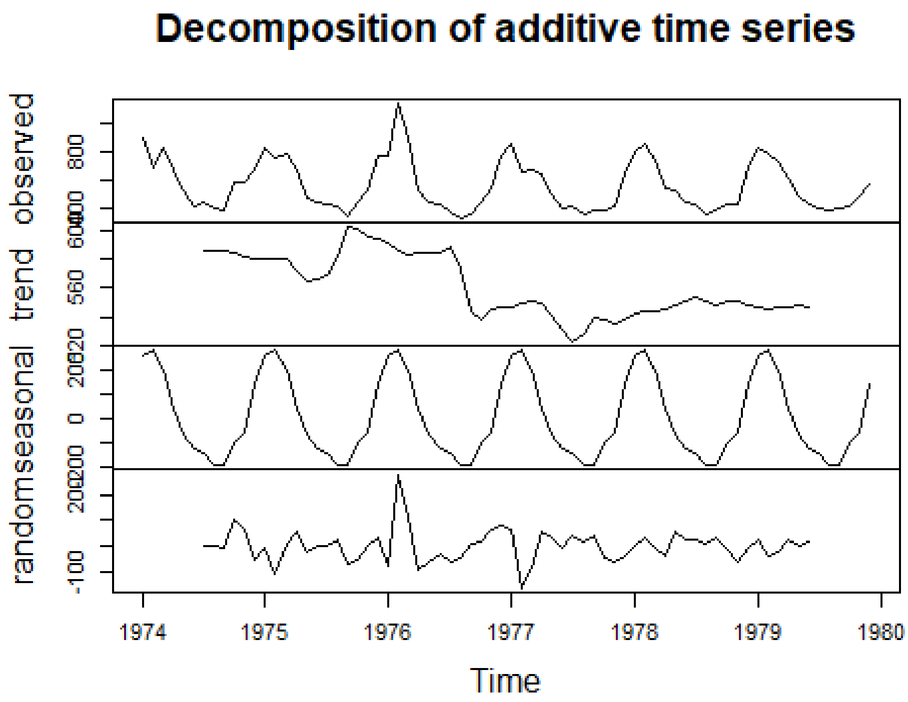
```
## [1] 56.92811
```

11. **Plot the predicted values along with the actual values to compare them**

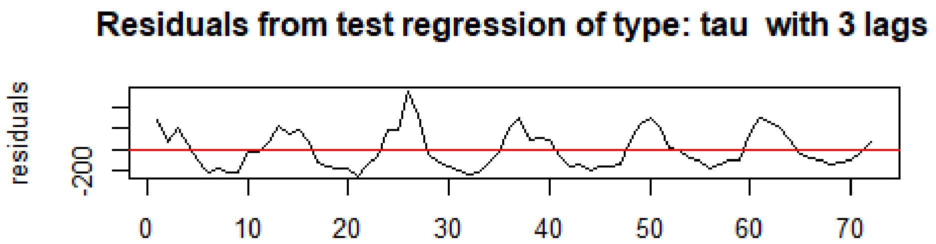
```
plot(fdeaths_25, col = "red")
lines(predict_zoo$fit, col = "green")
```

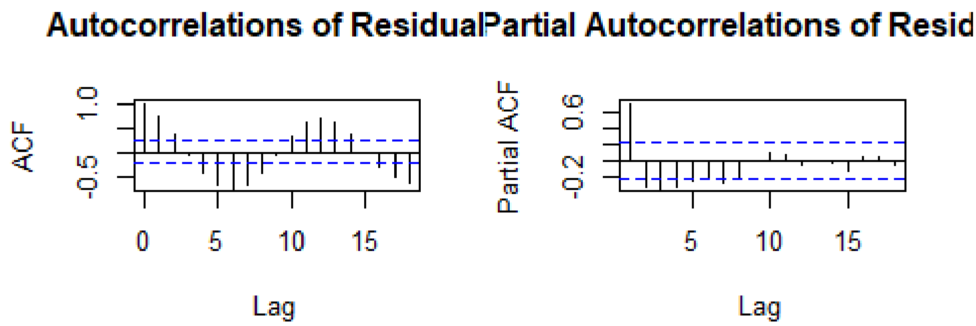



```
class(fdeaths_25)

## [1] "ts"

#ARIMA

components.ts = decompose(fdeaths)
plot(components.ts)
```



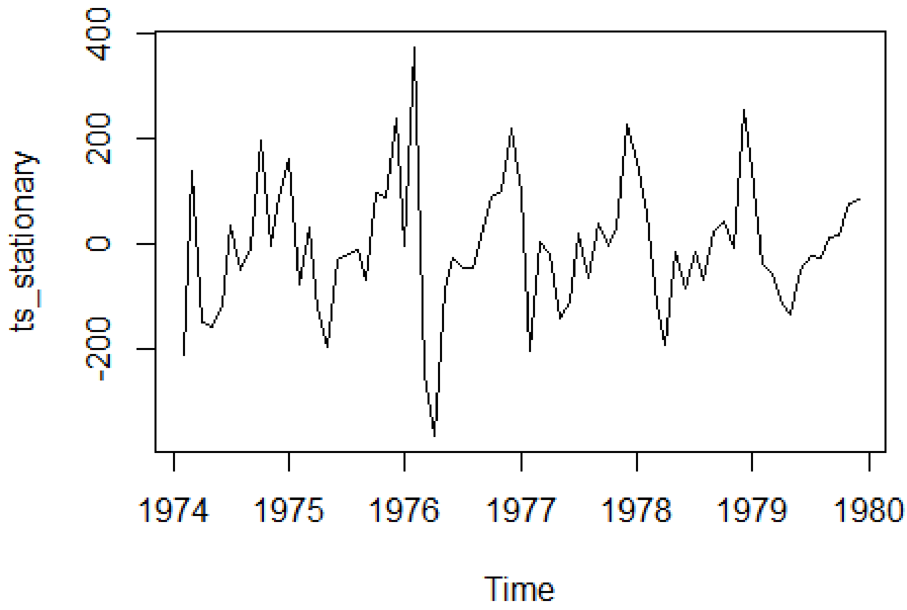

```
urkpssTest(fdeaths, type = c("tau"), lags = c("short"),use.lag = NULL, doplot = TRUE)
```



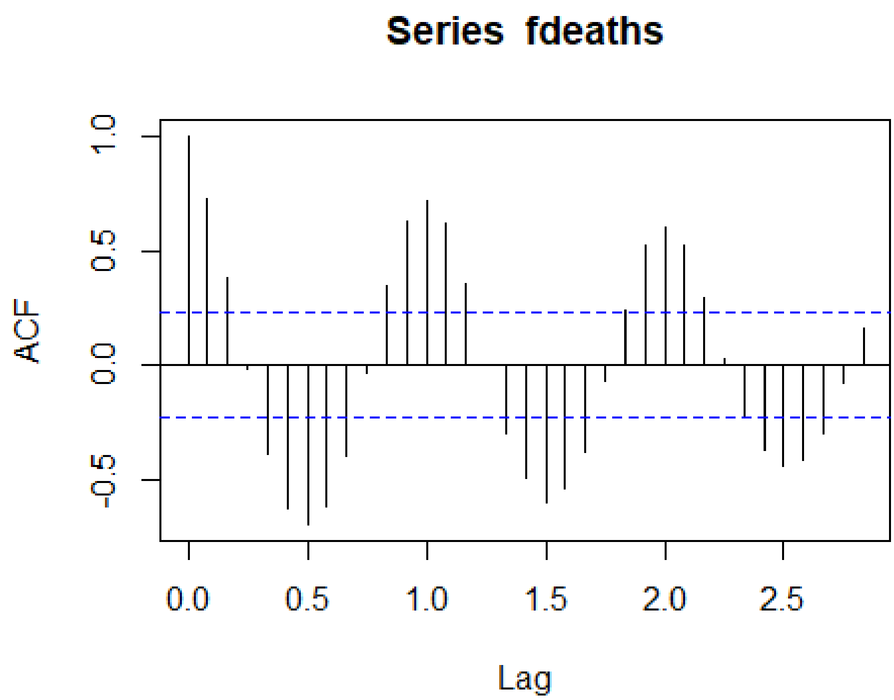


```
##
## Title:
##  KPSS Unit Root Test
##
## Test Results:
##   NA
##
## Description:
##  Fri May 03 16:26:33 2019 by user: kumar
```

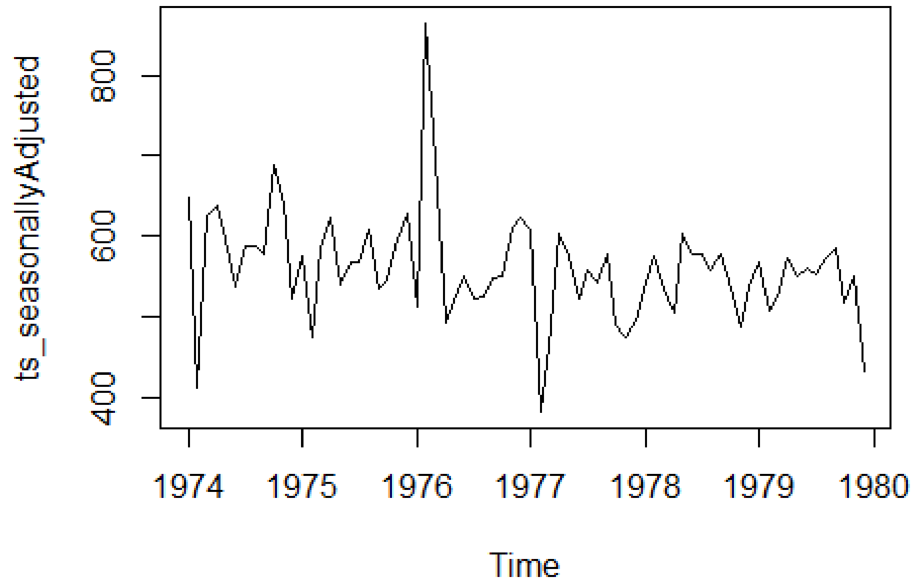
```
ts_stationary = diff(fdeaths, differences=1)
plot(ts_stationary)
```



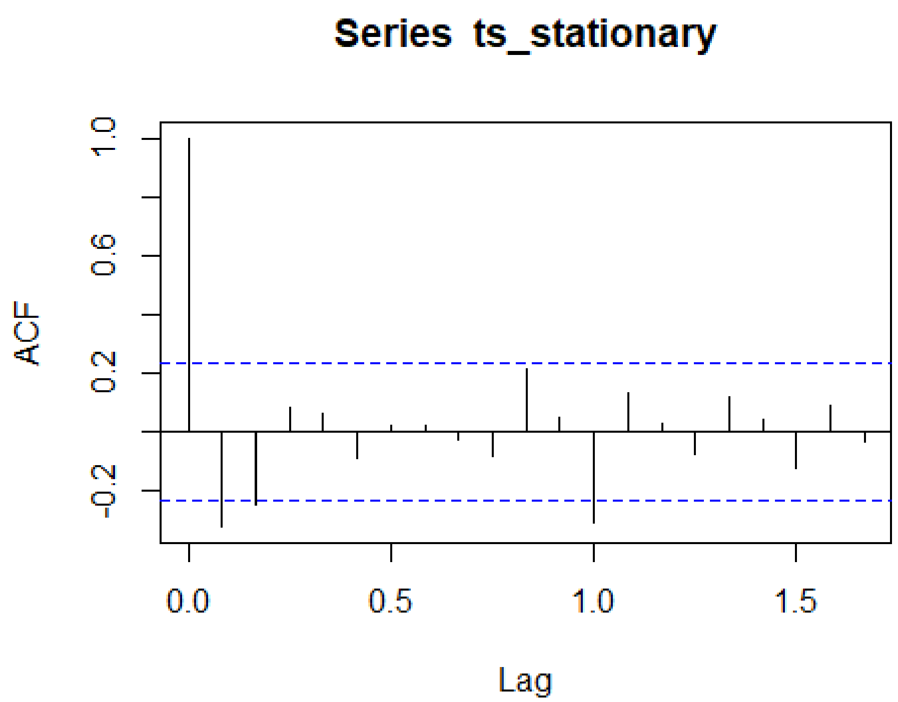

```
acf(fdeaths,lag.max = 34)
```



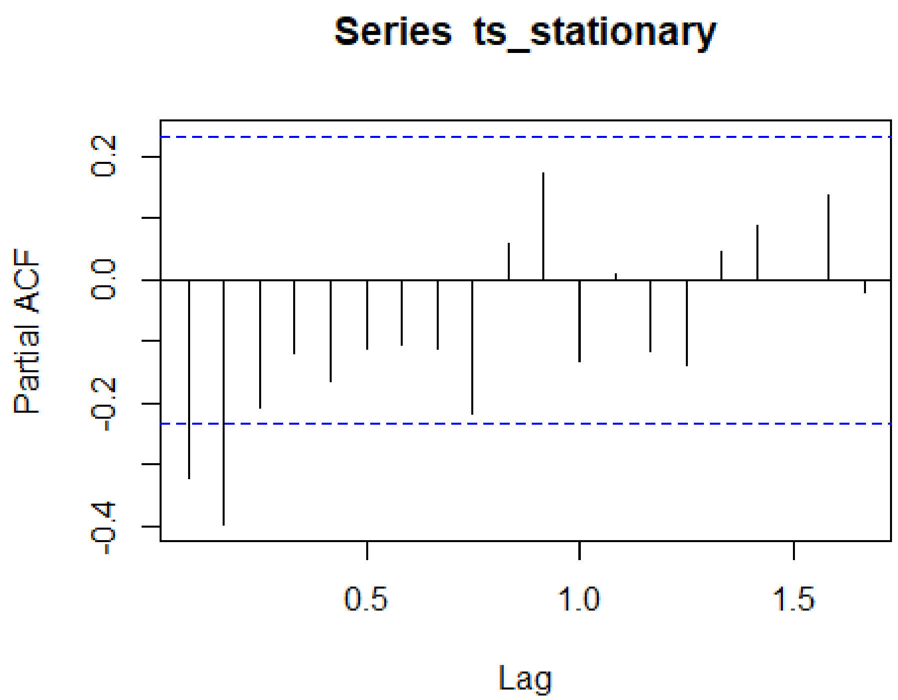

```
ts_seasonallyAdjusted <- fdeaths- components.ts$seasonal
ts_stationary <- diff(ts_seasonallyAdjusted, differences=1)

plot(ts_seasonallyAdjusted)
```



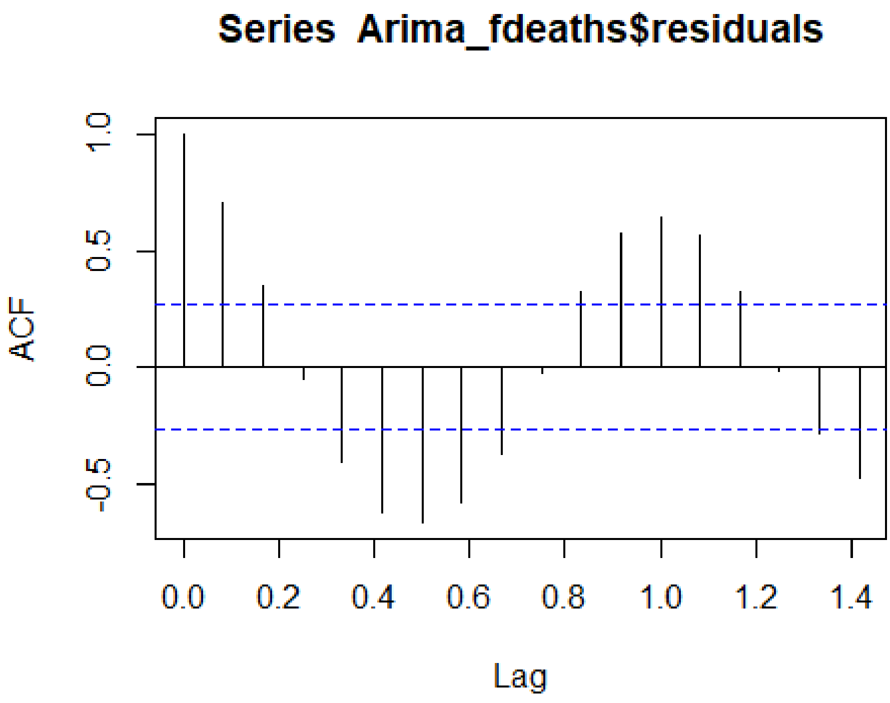
```
acf(ts_stationary, lag.max = 20)
```



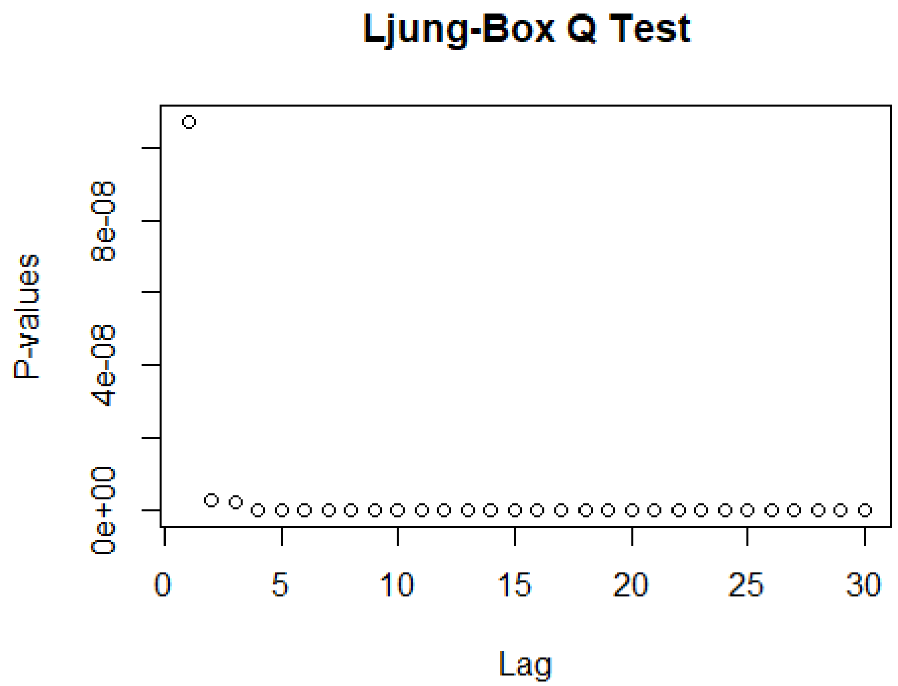

```
pacf(ts_stationary, lag.max = 20)
```



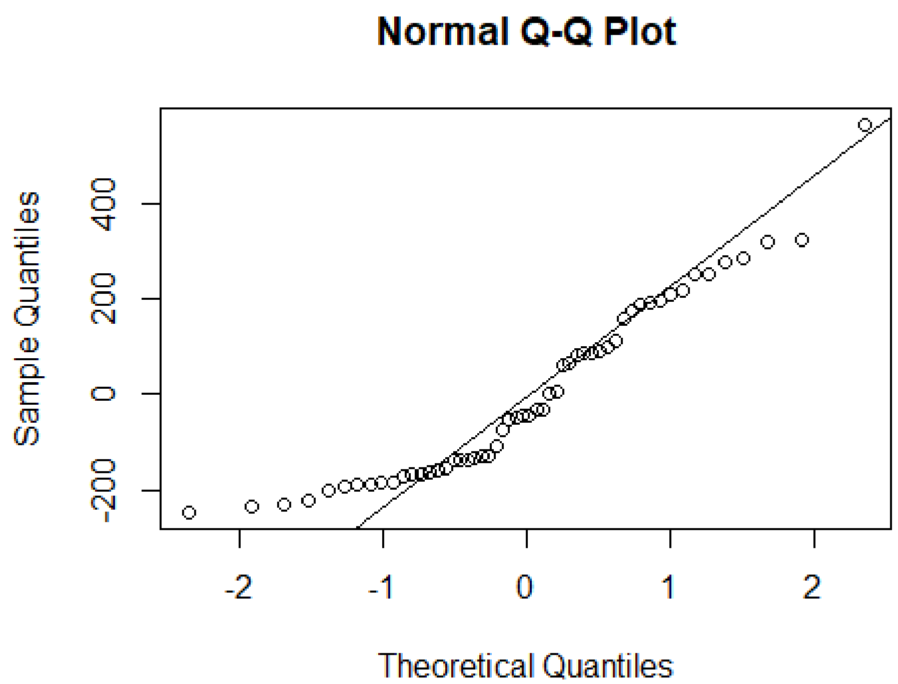
```
Arima_fdeaths <- arima(fdeaths_75, order=c(0,0,0.1699511),seasonal = list(
order = c(0,0,0.1699511), period = 12),method="ML")

coeftest(Arima_fdeaths)

##
## z test of coefficients:
##
##           Estimate Std. Error z value  Pr(>|z|)
## intercept  576.926     25.236  22.861 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

confint(Arima_fdeaths)

##             2.5 %   97.5 %
## intercept 527.4635 626.3884

acf(Arima_fdeaths$residuals)
```




```
boxresult<-LjungBoxTest (Arima_fdeaths$residuals,k=2,StartLag=1)
plot(boxresult[,3],main= "Ljung-Box Q Test", ylab= "P-values", xlab= "Lag")
```




```
qqnorm(Arima_fdeaths$residuals)
qqline(Arima_fdeaths$residuals)
```



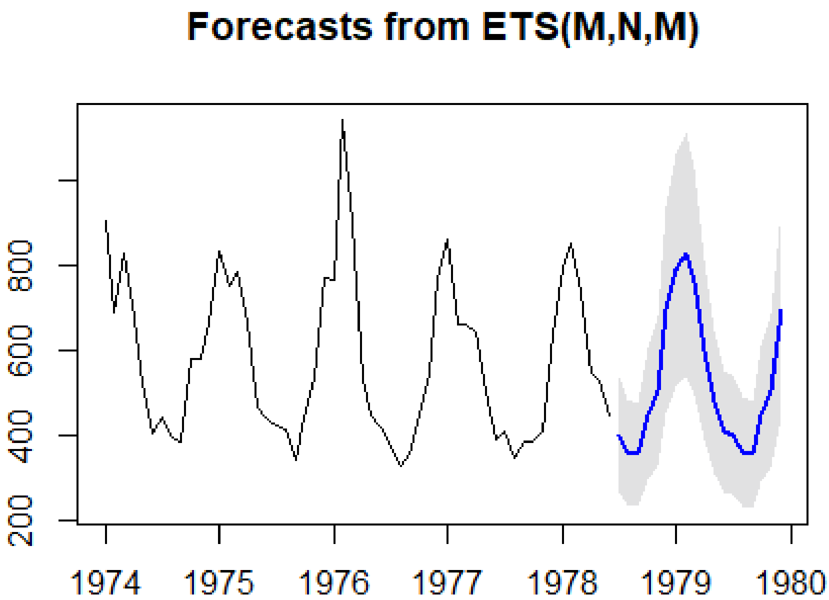
12. Build an ARIMA model for the period up to about 75% of the data (e.g., for the CO2 data, up to December 2015) using auto.arima()

```
auto.arima(fdeaths_75, trace=TRUE)

##
##  ARIMA(2,0,2)(1,1,1)[12] with drift        : Inf
##  ARIMA(0,0,0)(0,1,0)[12] with drift        : 528.6231
##  ARIMA(1,0,0)(1,1,0)[12] with drift        : 518.8591
##  ARIMA(0,0,1)(0,1,1)[12] with drift        : Inf
##  ARIMA(0,0,0)(0,1,0)[12]                   : 526.9322
##  ARIMA(1,0,0)(0,1,0)[12] with drift        : 530.5452
##  ARIMA(1,0,0)(1,1,1)[12] with drift        : Inf
##  ARIMA(1,0,0)(0,1,1)[12] with drift        : Inf
##  ARIMA(0,0,0)(1,1,0)[12] with drift        : 516.6078
##  ARIMA(0,0,0)(1,1,1)[12] with drift        : Inf
##  ARIMA(0,0,0)(0,1,1)[12] with drift        : Inf
##  ARIMA(0,0,1)(1,1,0)[12] with drift        : 518.6871
##  ARIMA(1,0,1)(1,1,0)[12] with drift        : 520.487
##  ARIMA(0,0,0)(1,1,0)[12]                   : 515.8743
##  ARIMA(0,0,0)(1,1,1)[12]                   : Inf
##  ARIMA(0,0,0)(0,1,1)[12]                   : Inf
##  ARIMA(1,0,0)(1,1,0)[12]                   : 517.7598
##  ARIMA(0,0,1)(1,1,0)[12]                   : 517.5001
##  ARIMA(1,0,1)(1,1,0)[12]                   : 519.2975
##
##  Best model: ARIMA(0,0,0)(1,1,0)[12]

## Series: fdeaths_75
## ARIMA(0,0,0)(1,1,0)[12]
##
## Coefficients:
##          sar1
##       -0.5069
## s.e.   0.1176
##
## sigma^2 estimated as 10738:  log likelihood=-255.78
## AIC=515.57   AICc=515.87   BIC=519.04
```

13. Predict the values for the next 15 months (e.g., for the CO2 data, all of 2016 and the first 3 months of 2017).

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
DHARWAD

```
arimaPred <- predict(Arima_fdeaths,n.ahead = 708)
futurVal <- forecast(fdeaths_75,h=18, level=c(99.5))
plot(futurVal)
```




```
futurVal

##          Point Forecast  Lo 99.5   Hi 99.5
## Jul 1978       401.3461 264.2365  538.4557
## Aug 1978       362.4939 238.2996  486.6882
## Sep 1978       355.1583 233.1279  477.1887
## Oct 1978       447.3793 293.2235  601.5352
## Nov 1978       504.1969 329.9699  678.4239
## Dec 1978       693.9247 453.4596  934.3898
## Jan 1979       788.2058 514.3028 1062.1089
## Feb 1979       826.0859 538.2179 1113.9538
## Mar 1979       760.2726 494.6031 1025.9422
## Apr 1979       600.7350 390.2345  811.2355
## May 1979       479.1535 310.7946  647.5124
## Jun 1979       408.9998 264.8980  553.1016
## Jul 1979       401.3461 259.5556  543.1367
## Aug 1979       362.4939 234.0832  490.9046
## Sep 1979       355.1584 229.0080  481.3087
## Oct 1979       447.3794 288.0476  606.7111
## Nov 1979       504.1970 324.1522  684.2417
## Dec 1979       693.9248 445.4739  942.3757
```

```
#Arima RMSE
diffArima <- (sqrt(mean((fdeaths_25 - arimaPred$pred)^2)))
diffArima

## [1] 159.0115
```