DA Lab 3

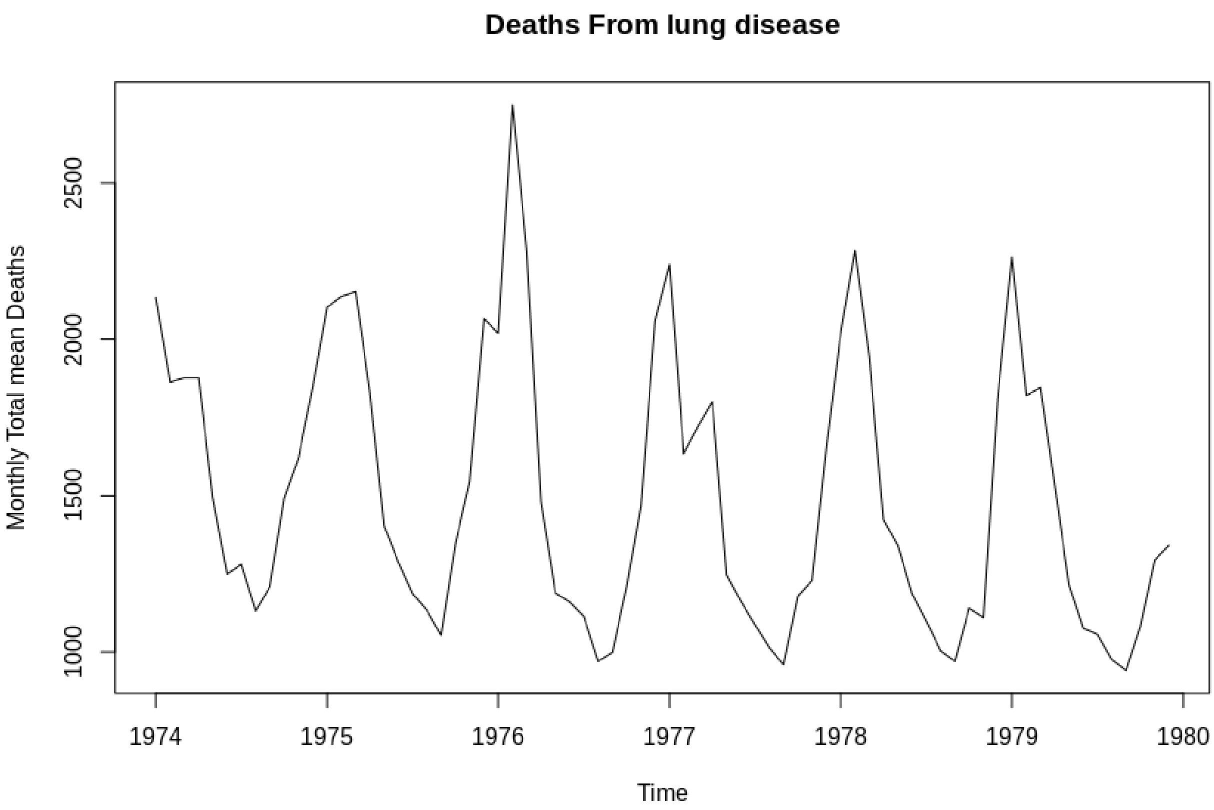
Time Series Analysis of **Death Due to Lung Cancer** Dataset

Name: Dheeraj Chaudhary

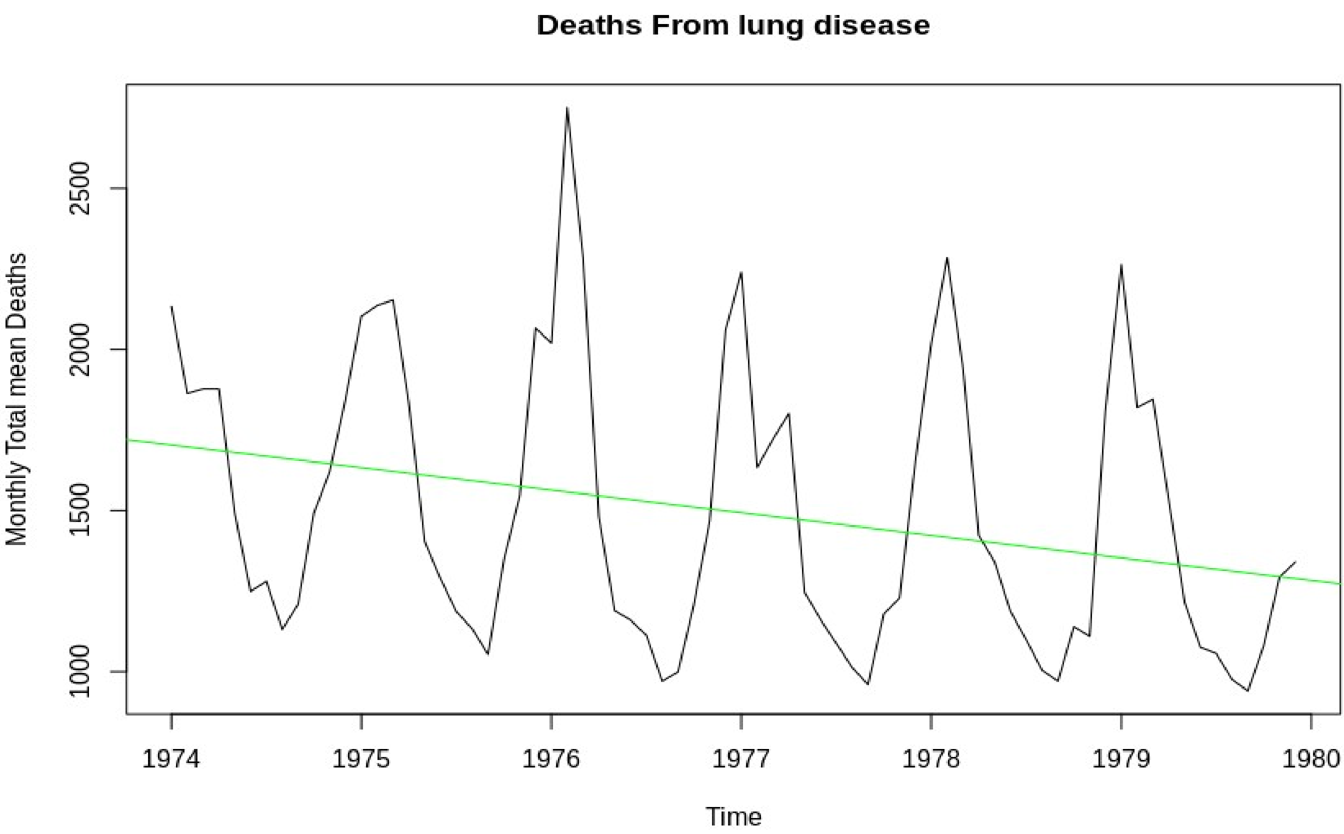
Roll: 17BCS009


```
library(ggplot2)

library(Metrics)

library(forecast)

library(reshape)

data("mdeaths")

mdeaths

######## OUTPUT  > mdeaths

      Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
1974 2134 1863 1877 1877 1492 1249 1280 1131 1209 1492 1621 1846
1975 2103 2137 2153 1833 1403 1288 1186 1133 1053 1347 1545 2066
1976 2020 2750 2283 1479 1189 1160 1113  970  999 1208 1467 2059
1977 2240 1634 1722 1801 1246 1162 1087 1013  959 1179 1229 1655
1978 2019 2284 1942 1423 1340 1187 1098 1004  970 1140 1110 1812
1979 2263 1820 1846 1531 1215 1075 1056  975  940 1081 1294 1341


start(mdeaths)

######## OUTPUT

       1974    1


end(mdeaths)

######## OUTPUT

       1979   12


############ Q.2) TIME SERIES OBJECT OF THE DATA ###############

my_Object <- ts(mdeaths, start=1974 ,frequency = 12)

Object

######  Check whether it is an object or not ############

is.ts(my_Object)

###### OUTPUT

[1] TRUE
```

```
###### PLOT OF MY DATA ########
ts.plot(my_Object, main="Deaths From lung disease",ylab ="Monthly Total mean Deaths")
```

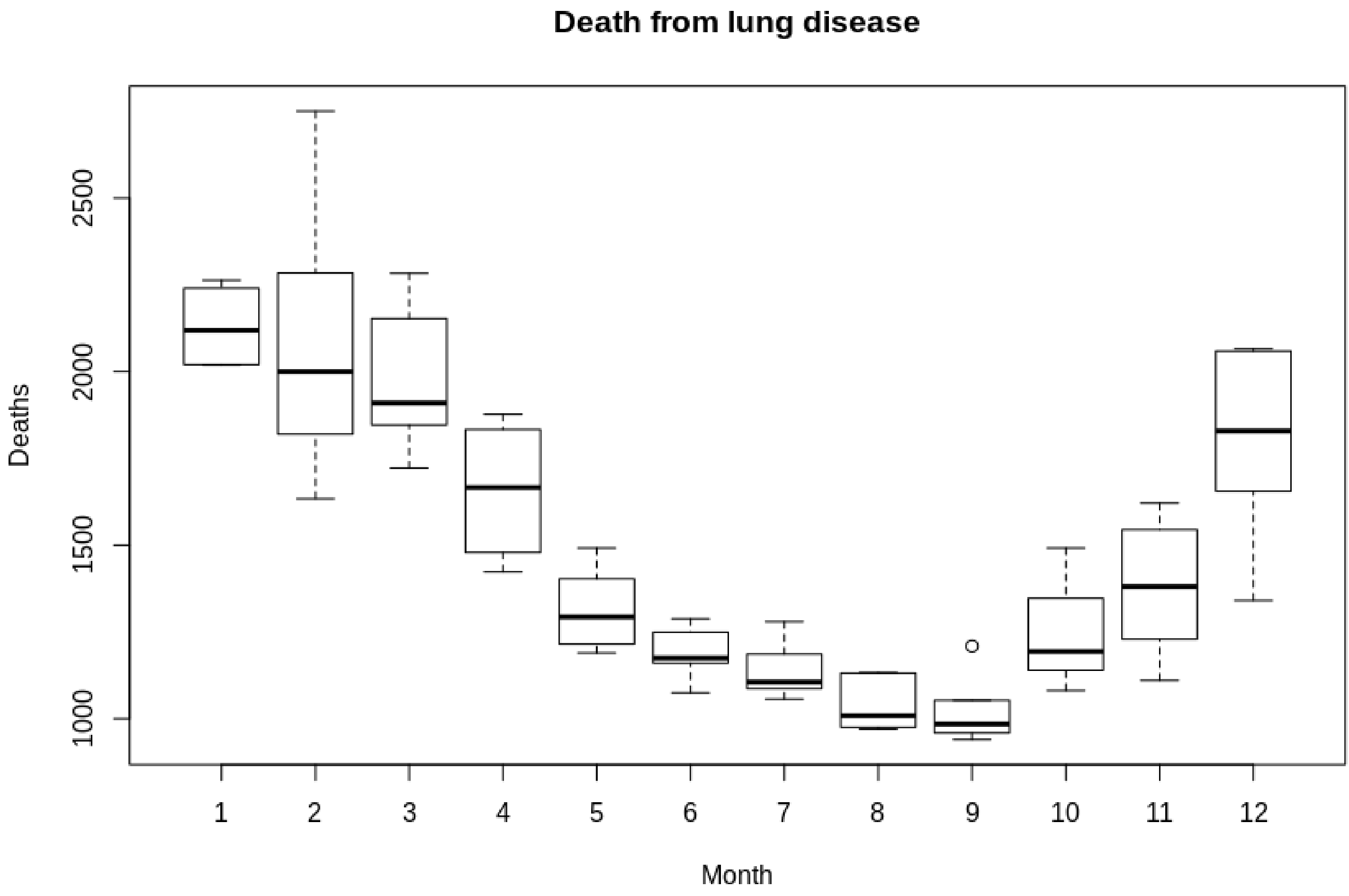



```
###### PLOT OF MY DATA with adding a horizontal regression line in the plot ########
abline(reg = lm(my_Object~time(my_Object)),col="green")
```

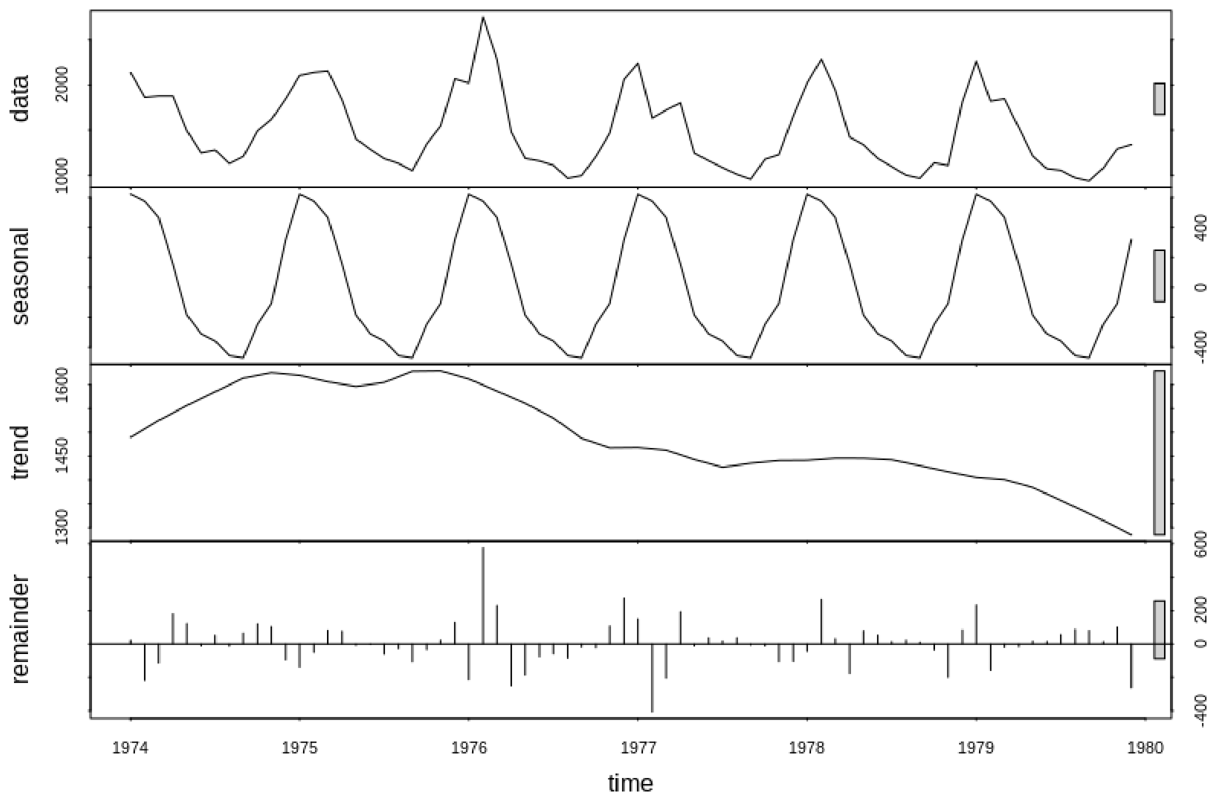


```
#################### Q.3) PLOT OF YEARLY MEAN VALUES ARE #############
plot(aggregate(my_Object,FUN = mean))
```

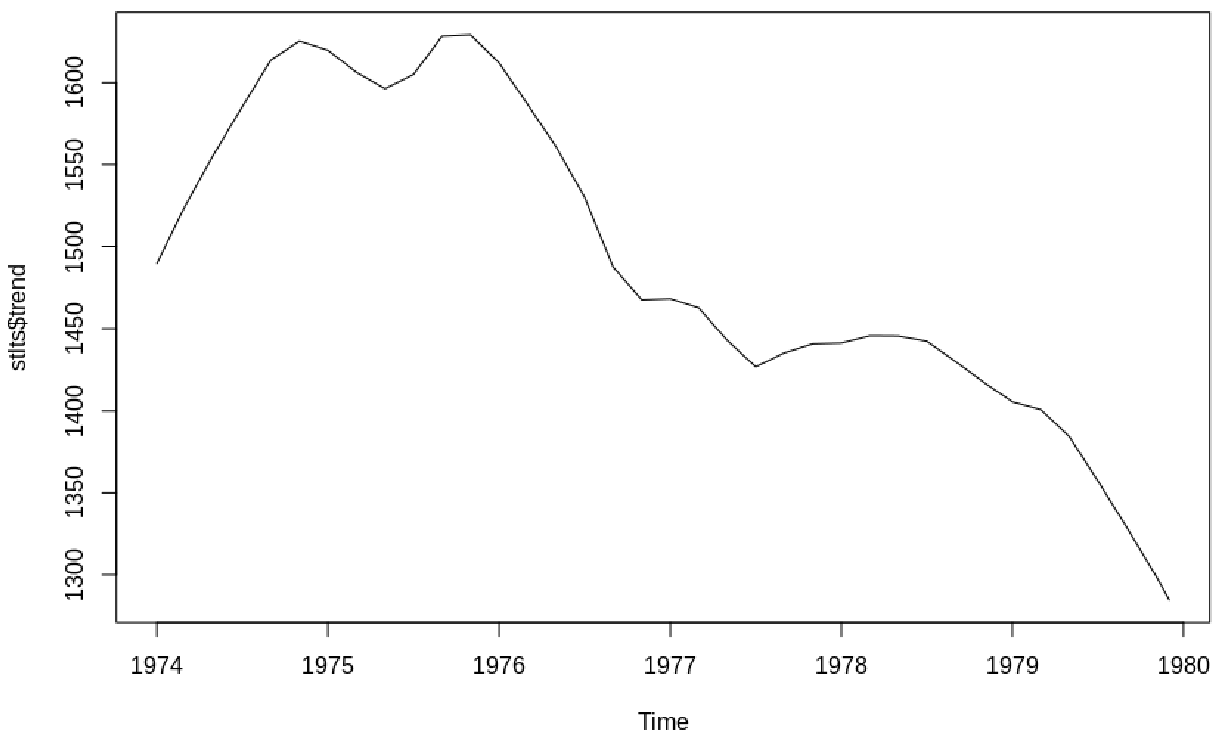

```
################### Q.4) PLOT OF MONTHLY BOX PLOTS ARE #################
boxplot(my_Object~cycle(my_Object),xlab="Month",ylab = "Deaths",main = "Death from lung
disease")
```



```
################ Q.5) DECOMPOSING THE ABOVE TIME SERIES USING STL FUNCTIONS ##########

decomp_STL <- stl(my_Object, s.window = "periodic")
plot(decomp_STL)
```

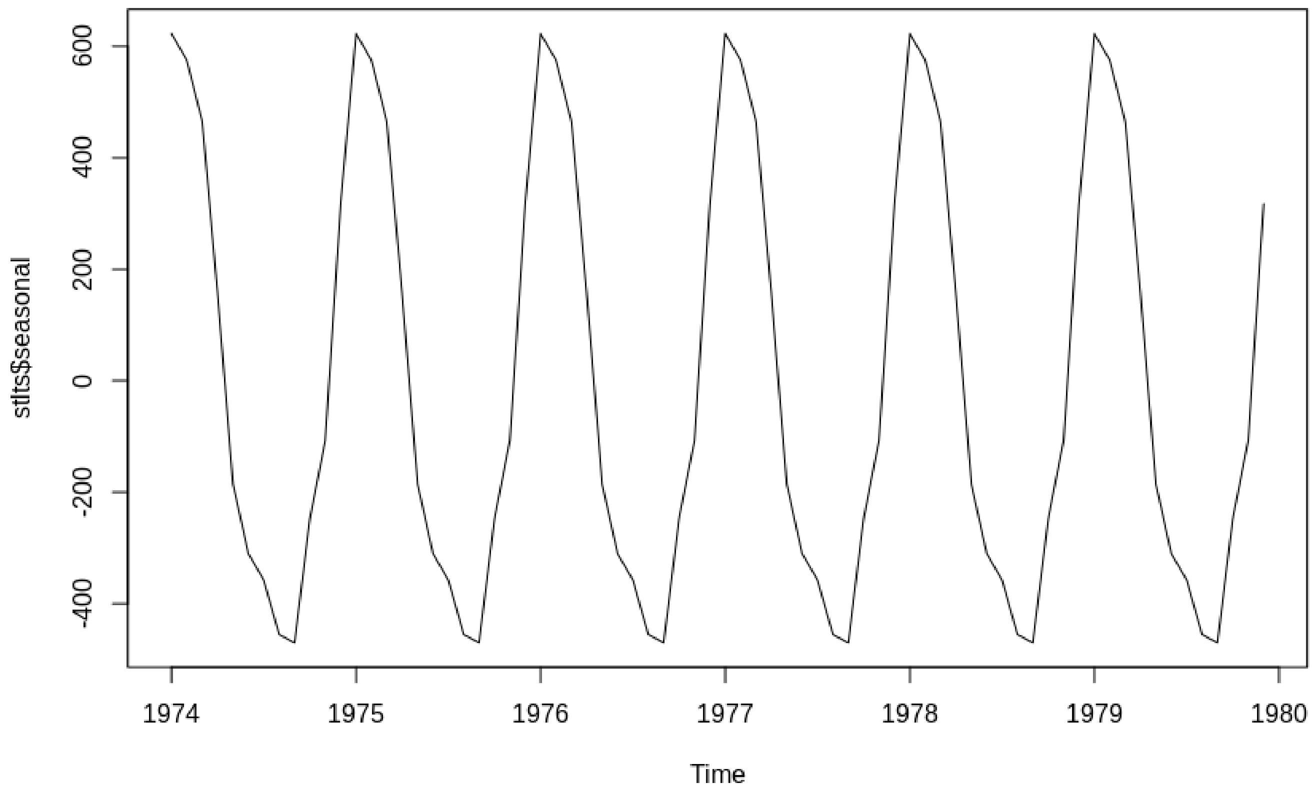



```
############### PLOTTING TREND GRAPH SEPARATELY #################
decomp_STL$trend <- decomp_STL$time.series[,2]
plot(decomp_STL$trend)                ## We can see Trend is Decreasing
```

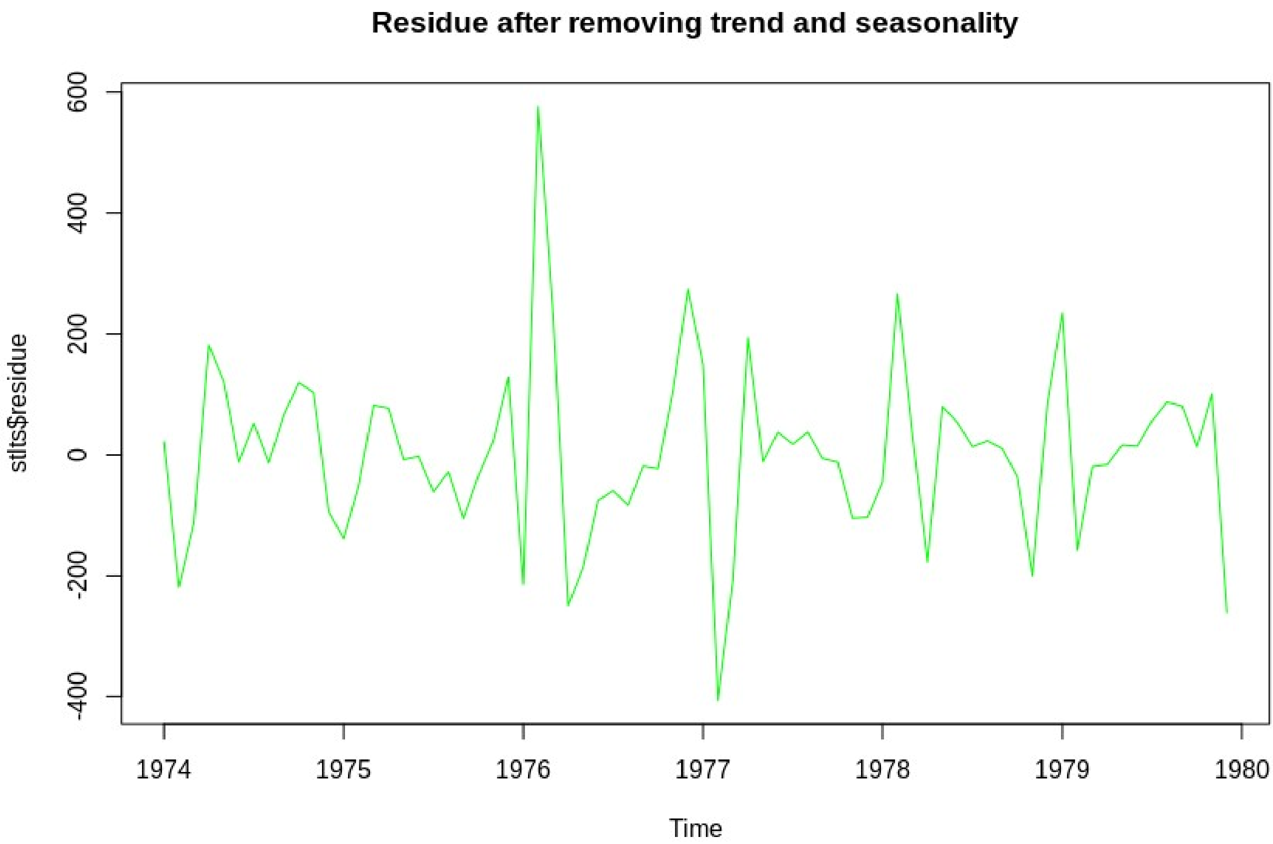



```
################ Q.6) TO CHECK SEASONALITY ######################
```

```
decomp_STL$seasonal <- decomp_STL$time.series[,1]
plot(decomp_STL$seasonal)          # We can see seasonality is Uniform in the plot
```



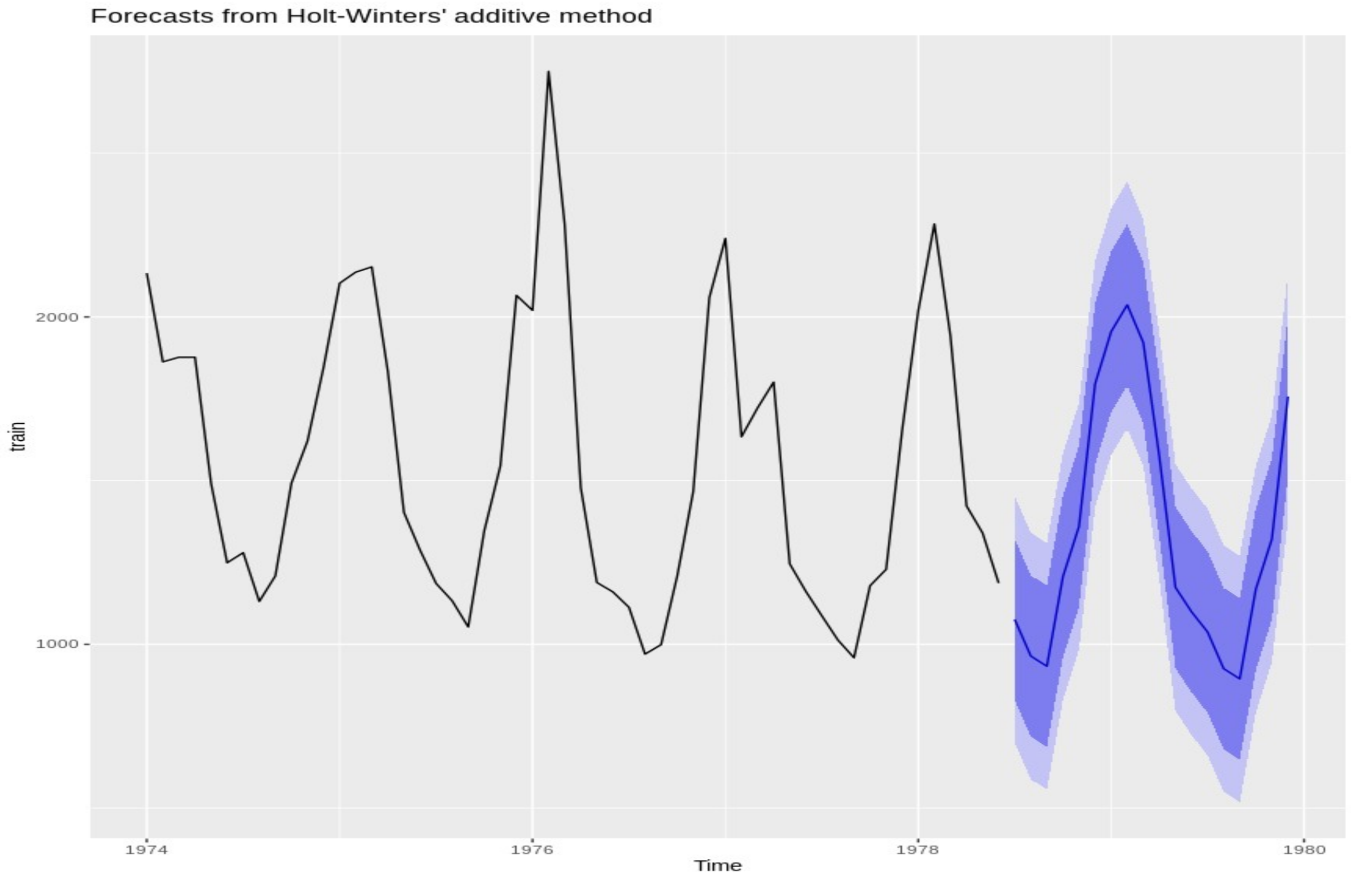

```
############## Q.7) RESIDUE AFTER REMOVING THE TREND AND SEASONALITY ##########

decomp_STL$residue <- (my_Object-(decomp_STL$trend + decomp_STL$seasonal))
plot(decomp_STL$residue,main = "Residue after removing trend and seasonality",col = "green")
```



```
############## Q.8) MODEL OF THE DATA USING HOLTWINTER METHOD FOR 75% OF DATA ########
holt_model <- window(mdeaths,start = c(1974,1) ,end=c(1978,6))
holt_model

###### OUTPUT
      Jan  Feb Mar  Apr May Jun  Jul Aug Sep Oct  Nov Dec
1974 2134 1863 1877 1877 1492 1249 1280 1131 1209 1492 1621 1846
1975 2103 2137 2153 1833 1403 1288 1186 1133 1053 1347 1545 2066
1976 2020 2750 2283 1479 1189 1160 1113  970 999 1208 1467 2059
1977 2240 1634 1722 1801 1246 1162 1087 1013  959 1179 1229 1655
1978 2019 2284 1942 1423 1340 1187

########### SUMMARY of MODEL AND PLOT FOR NEXT 25% DATA FOR ABOVE HOLTWINTER MODEL ######

Pred_data <- hw(holt_model, seasonal = "additive", h = 18)

summary(Pred_data)

###### OUTPUT

Forecast method: Holt-Winters' additive method


Model Information:
Holt-Winters' additive method


Call:
 hw(y = train, h = 18, seasonal = "additive")


  Smoothing parameters:
    alpha = 1e-04
    beta  = 1e-04
    gamma = 2e-04


  Initial states:
    l = 1618.6168
    b = -3.2349
    s = 369.3278 -68.9199 -225.7441 -502.8133 -474.9254 -366.2004
           -307.6114 -236.3283 158.7177 504.9388 617.2867 532.2718


  sigma:  192.1291


     AIC     AICc   BIC
798.3117 815.3117 832.1245


Error measures:
                   ME     RMSE   MAE   MPE MAPE     MASE ACF1
Training set 4.543167 161.1714 110.868 -0.3603639 6.606394 0.572891 0.192809
```

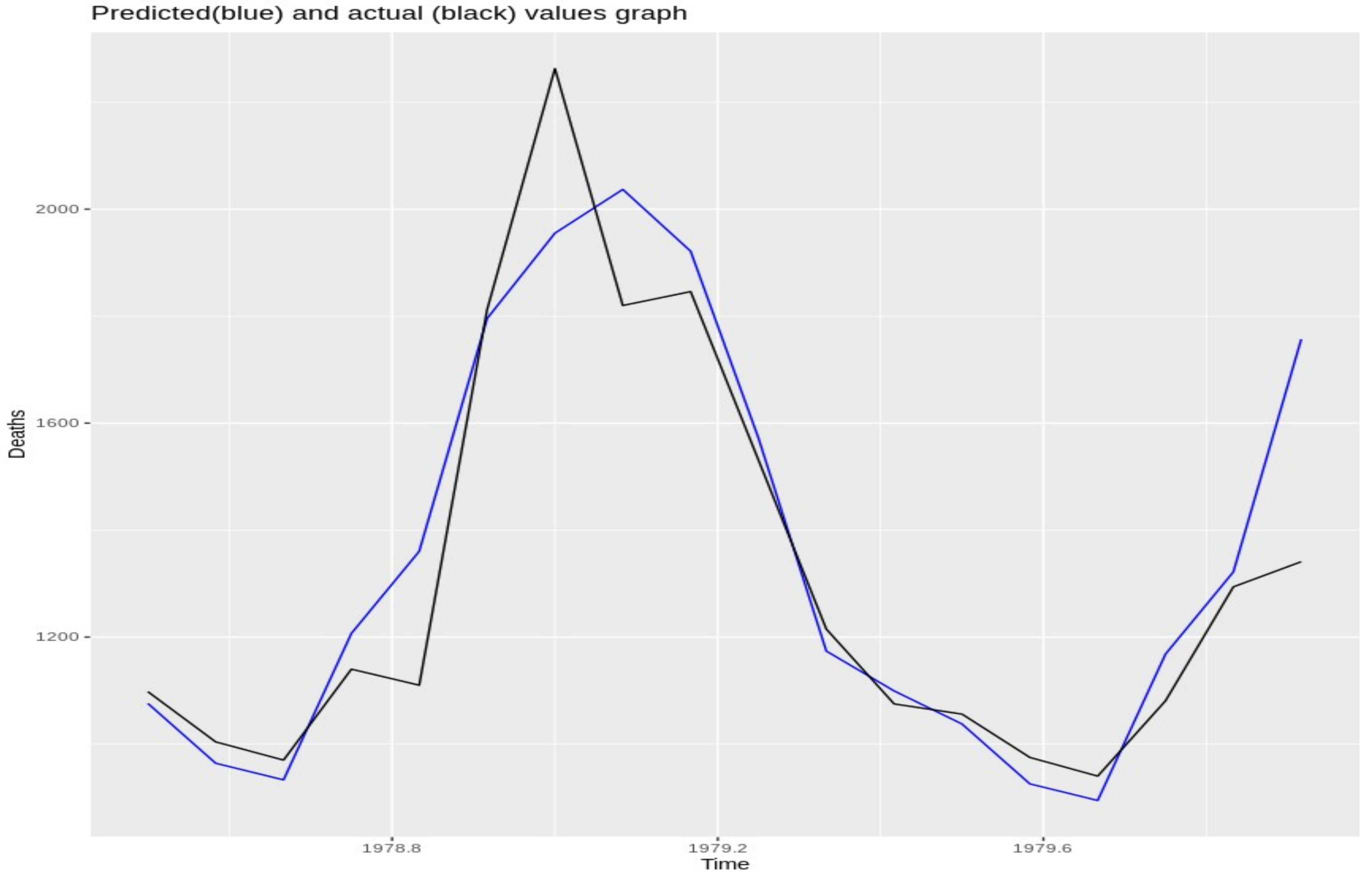
```
Forecasts:
         Point Forecast     Lo 80    Hi 80     Lo 95    Hi 95
Jul 1978      1076.0085  829.7852 1322.232  699.4424 1452.575
Aug 1978       964.0788  717.8555 1210.302  587.5127 1340.645
Sep 1978       932.9994  686.7760 1179.223  556.4332 1309.565
Oct 1978      1206.8411  960.6177 1453.064  830.2749 1583.407
Nov 1978      1360.4591 1114.2357 1606.683  983.8929 1737.025
Dec 1978      1795.5011 1549.2776 2041.725 1418.9348 2172.067
Jan 1979      1955.2547 1709.0312 2201.478 1578.6884 2331.821
Feb 1979      2037.0099 1790.7863 2283.234 1660.4434 2413.576
Mar 1979      1921.4300 1675.2063 2167.654 1544.8634 2297.997
Apr 1979      1572.0339 1325.8100 1818.258 1195.4670 1948.601
May 1979      1173.8253  927.6013 1420.049  797.2582 1550.392
Jun 1979      1099.2835  853.0594 1345.508  722.7162 1475.851
Jul 1979      1037.4852  791.2608 1283.710  660.9174 1414.053
Aug 1979       925.5555  679.3308 1171.780  548.9874 1302.124
Sep 1979       894.4761  648.2511 1140.701  517.9075 1271.045
Oct 1979      1168.3178  922.0925 1414.543  791.7488 1544.887
Nov 1979      1321.9358 1075.7102 1568.161  945.3663 1698.505
Dec 1979      1756.9778 1510.7518 2003.204 1380.4076 2133.548
```

```
############### Q.9) FORECAST PLOT FOR NEXT 25% DATA ################
autoplot(Pred_data)
```



```
############### PREDICTED Plot And VALUE ALONG WITH THE ACTUAL VALUE #########

act_value = tail(my_Object,18)

df = data.frame( Pred_data , tail(my_Object,18))

X = time(act_value)

fore_cast = df$Point.Forecast

fore_cast

##### OUTPUT

 [1] 1076.0085  964.0788 932.9994 1206.8411 1360.4591 1795.5011 1955.2547 2037.0099 1921.4300 1572.0339 1173.8253 1099.2835 1037.4852

[14]  925.5555  894.4761 1168.3178 1321.9358 1756.9778


actu_data = df$tail.my_Object..18.

actu_data

###### OUTPUT

      Jan  Feb Mar  Apr May Jun  Jul Aug Sep Oct  Nov Dec

1978                          1098 1004 970 1140 1110 1812

1979 2263 1820 1846 1531 1215 1075 1056  975 940 1081 1294 1341


Pred_actual = as.data.frame(data.frame(fore_cast,actu_data))

Pred_actual

###### OUTPUT

        fore actu

1  1076.0085 1098
2   964.0788 1004
3   932.9994  970
4  1206.8411 1140
5  1360.4591 1110
6  1795.5011 1812
7  1955.2547 2263
8  2037.0099 1820
9  1921.4300 1846
10 1572.0339 1531
11 1173.8253 1215
12 1099.2835 1075
13 1037.4852 1056
14  925.5555  975
15  894.4761  940
16 1168.3178 1081
17 1321.9358 1294
18 1756.9778 1341
```

```
ggplot(Pred_actual,aes(X))+
  geom_line(aes(y=dfpPred_actuallt$fore_cast),colour = "blue")+
  geom_line(aes(y=Pred_actual$actu_data),colour = "black") + xlab("Time") + ylab("Deaths") +
  ggtitle("Predicted(blue) and actual (black) values graph")
```




```
############### Q.11) RMS ERROR BETWEEN PREDICTED AND ACTUAL VALUE #############
rmse(df$Point.Forecast,df$tail.my_Object..18.)
#### OUTPUT
 [1] 150.6795

#########  Q.12 )TUNING THE MODEL BY MODIFYING THE VALUE OF ALPHA, BETA, AND GAMMA ######
hw_modelt <- HoltWinters(train,alpha = "0.1" ,beta = "0.1" ,gamma = "0.4" )
model.predict <- predict(hw_modelt,n.ahead = 18)
round(model.predict)
##### OUTPUT
```

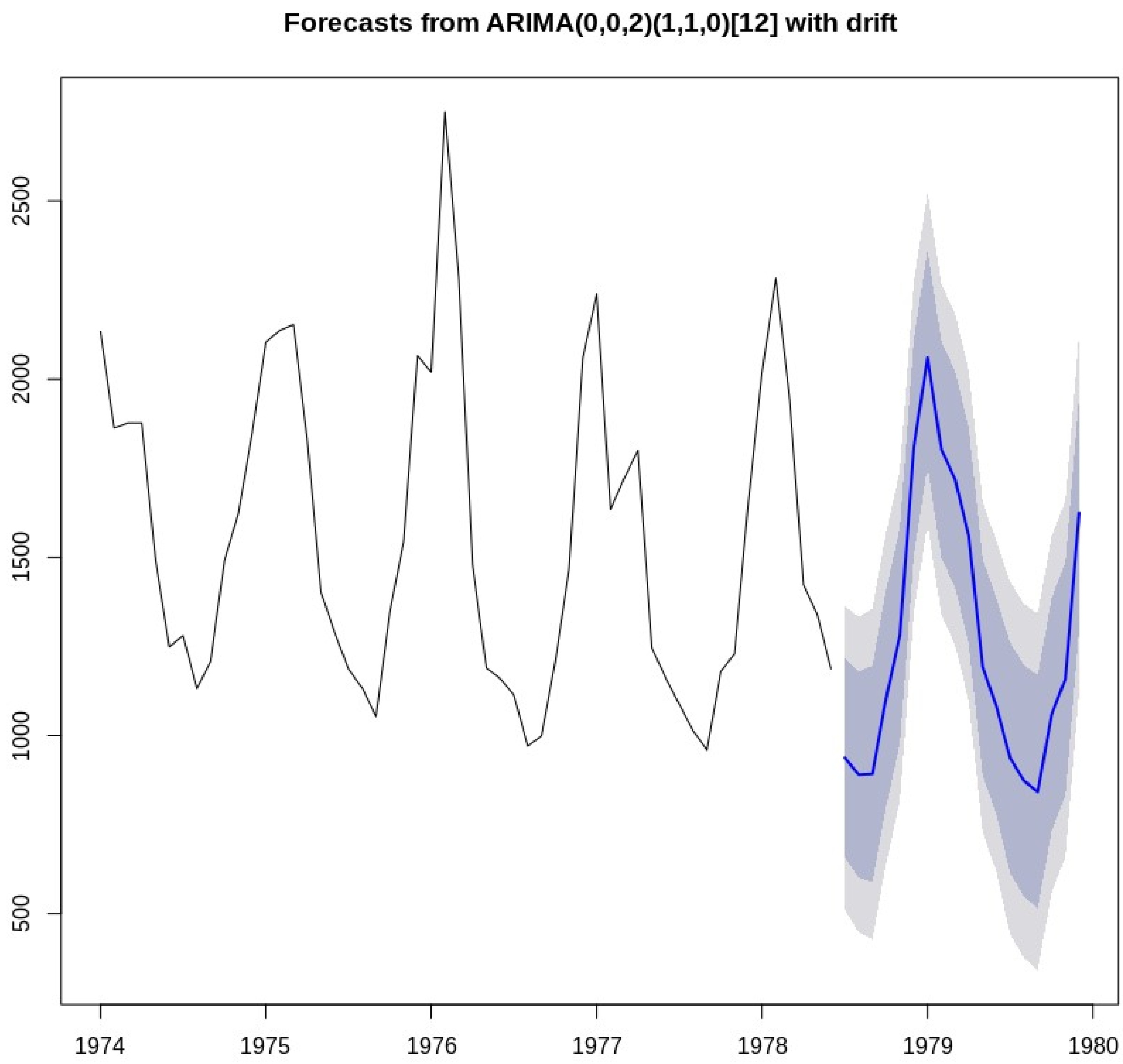
	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1978							1032	927	917	1164	1313	1749
1979	1982	2048	1873	1508	1203	1088	995	889	880	1126	1276	1712

```
p_values= model.predict
act_value = tail(time_series1,18)
rmse(act_value,p_values)
#### OUTPUT
```

```
[1] 138.5872


############# Q.13) MODEL OF THE DATA USING ARIMA METHOD FOR 75% OF DATA ########

arima_model <- window(mdeaths,start = c(1974,1) ,end=c(1978,6))
arima_model
######## OUTPUT
      Jan  Feb Mar  Apr May Jun  Jul Aug Sep Oct  Nov Dec
1974 2134 1863 1877 1877 1492 1249 1280 1131 1209 1492 1621 1846
1975 2103 2137 2153 1833 1403 1288 1186 1133 1053 1347 1545 2066
1976 2020 2750 2283 1479 1189 1160 1113  970 999 1208 1467 2059
1977 2240 1634 1722 1801 1246 1162 1087 1013  959 1179 1229 1655
1978 2019 2284 1942 1423 1340 1187


new_model = auto.arima(arima_model)
new_model

######## OUTPUT
Series: train

ARIMA(0,0,2)(1,1,0)[12] with drift


Coefficients:
        ma1      ma2 sar1    drift
     0.2757  -0.3298 -0.5996  -4.7405
s.e.  0.1750   0.1997 0.1195   1.8286


sigma^2 estimated as 47385:  log likelihood=-286.47
AIC=582.95   AICc=584.62 BIC=591.64



############# Q.14) PREDICTED OUTPUT FOR NEXT 25% DATA #######################

pred_arima <- forecast(new_model, h = 18)

pred_arima

######## OUTPUT

Point Forecast     Lo 80 Hi 80 Lo 95    Hi 95

Jul 1978      937.5533 658.5834 1216.523  510.9057 1364.201
Aug 1978       889.9089 600.5321 1179.286  447.3453 1332.472
Sep 1978       891.9903 588.3411 1195.639  427.5990 1356.382
Oct 1978      1105.3951 801.7460 1409.044  641.0038 1569.786
Nov 1978      1280.7029 977.0538 1584.352  816.3116 1745.094
Dec 1978      1806.2297 1502.5805 2109.879 1341.8384 2270.621
Jan 1979      2060.5104 1756.8613 2364.160 1596.1191 2524.902
Feb 1979      1803.2946 1499.6455 2106.944 1338.9033 2267.686
Mar 1979      1719.1050 1415.4558 2022.754 1254.7137 2183.496
Apr 1979      1558.6412 1254.9920 1862.290 1094.2499 2023.032
```

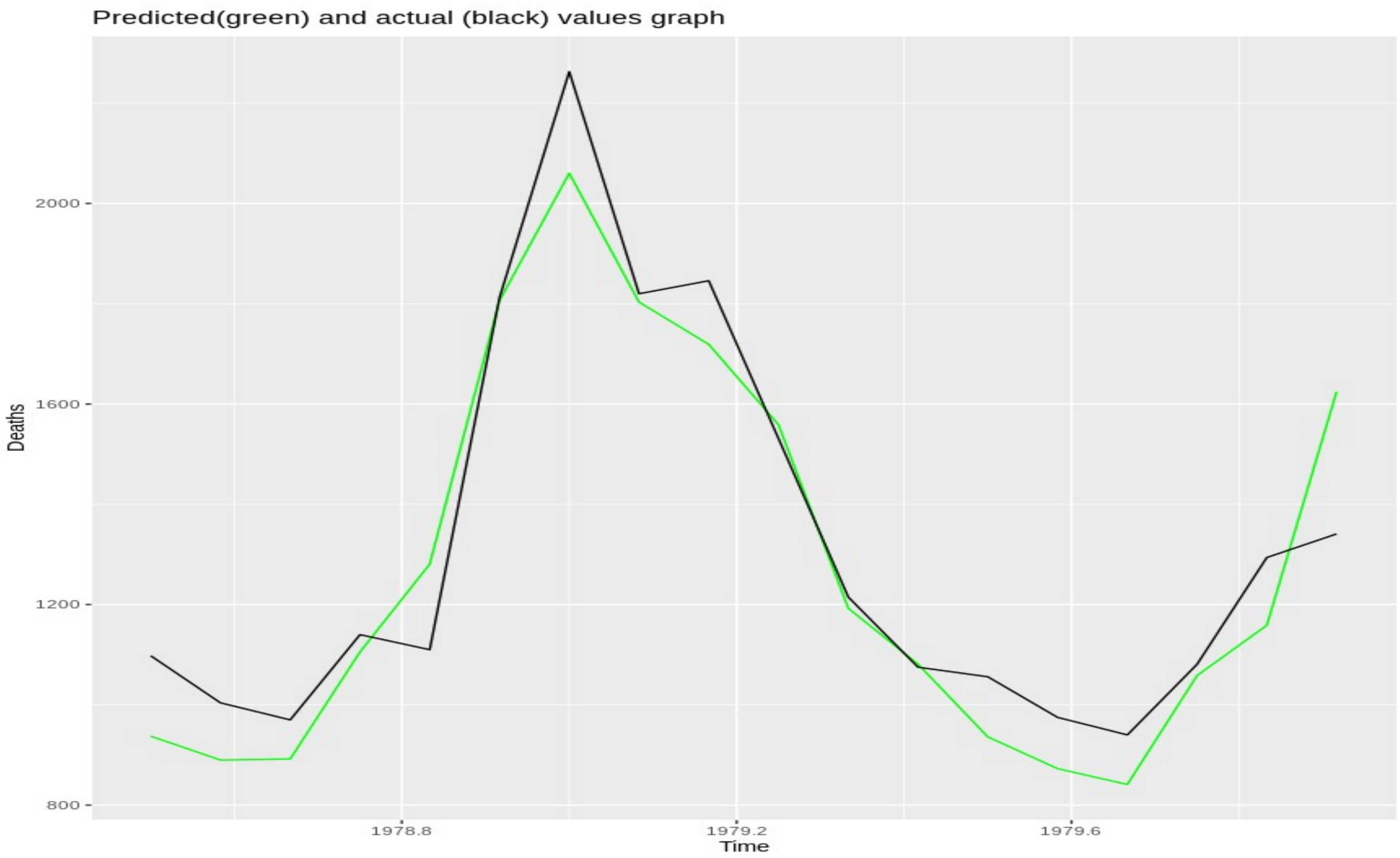
```
May 1979      1192.6494 889.0002 1496.299  728.2581 1657.041
Jun 1979      1081.0189 777.3698 1384.668  616.6276 1545.410
Jul 1979       936.1633 612.6171 1259.710  441.3421 1430.985
Aug 1979       872.7172 547.7086 1197.726  375.6594 1369.775
Sep 1979       841.1745 514.0845 1168.264  340.9336 1341.415
Oct 1979      1058.5335 731.4436 1385.623  558.2926 1558.774
Nov 1979      1158.7119 831.6220 1485.802  658.4710 1658.953
Dec 1979      1624.5665 1297.4766 1951.656 1124.3256 2124.807
```

```
plot(mdeaths)
plot(pred_arima)
predicted = data.frame(pred_arima)
arima_act_values = tail(my_Object,18)
```




```
################# Q.15) PLOTTING PREDICTED AND ACTUAL VALUES ##############
X = time(arima_act_values)
```

```
arima_df <- as.data.frame(data.frame(X,predicted$Point.Forecast,arima_act_values))
ggplot(arima_df,aes(X))+
  geom_line(aes(y=predicted$Point.Forecast),colour="green")+
  geom_line(aes(y=arima_act_values),colour = "black") + xlab("Time") + ylab("Deaths")+
  ggtitle("Predicted(green) and actual (black) values graph")
```

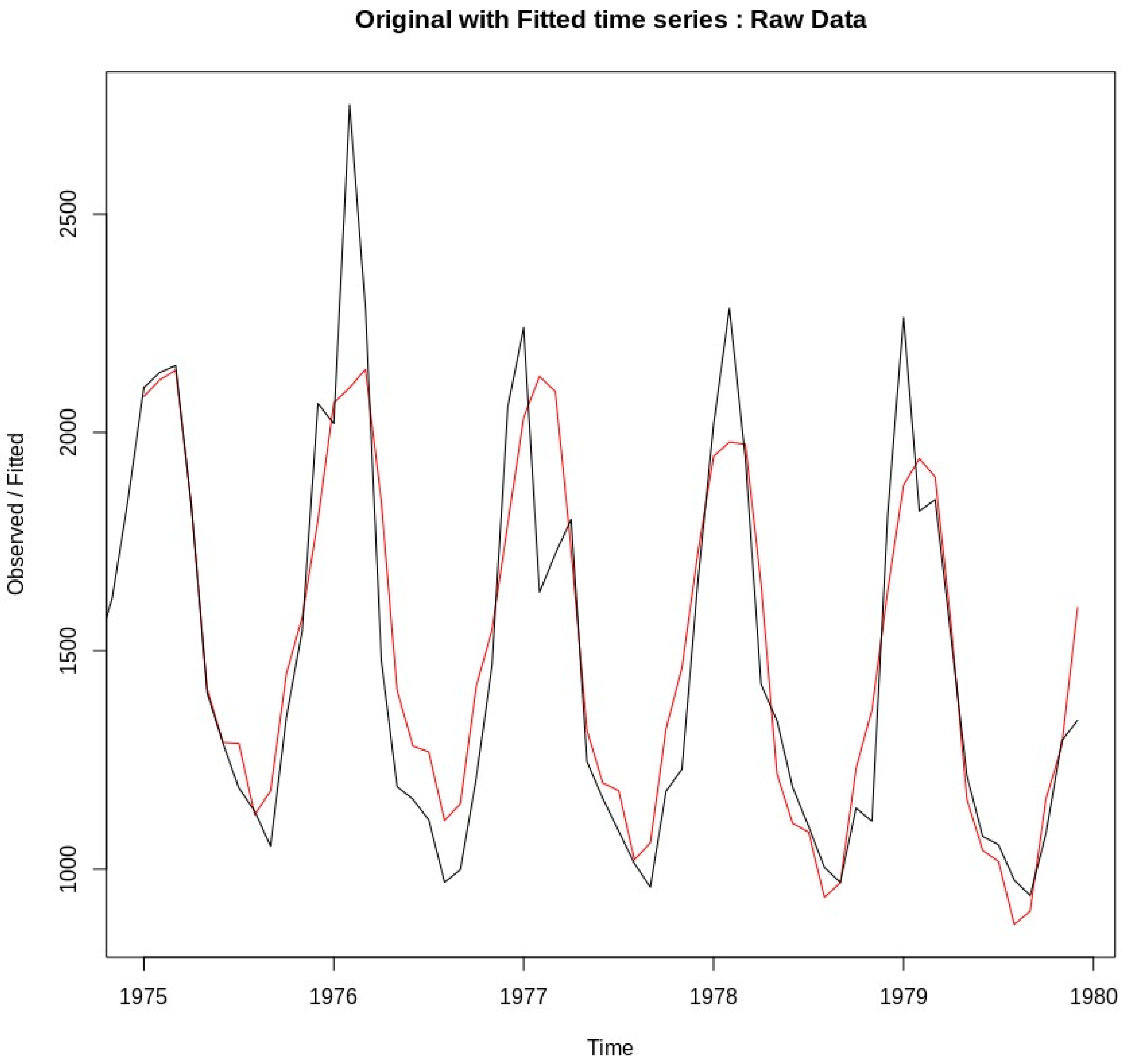



```
############ Q.16) RMS ERROR BETWEEN PREDICTED AND ACTUAL VALUE USING ARIMA MODEL ###########
rmse(arima_act_values,predicted$Point.Forecast)
###### OUTPUT
```

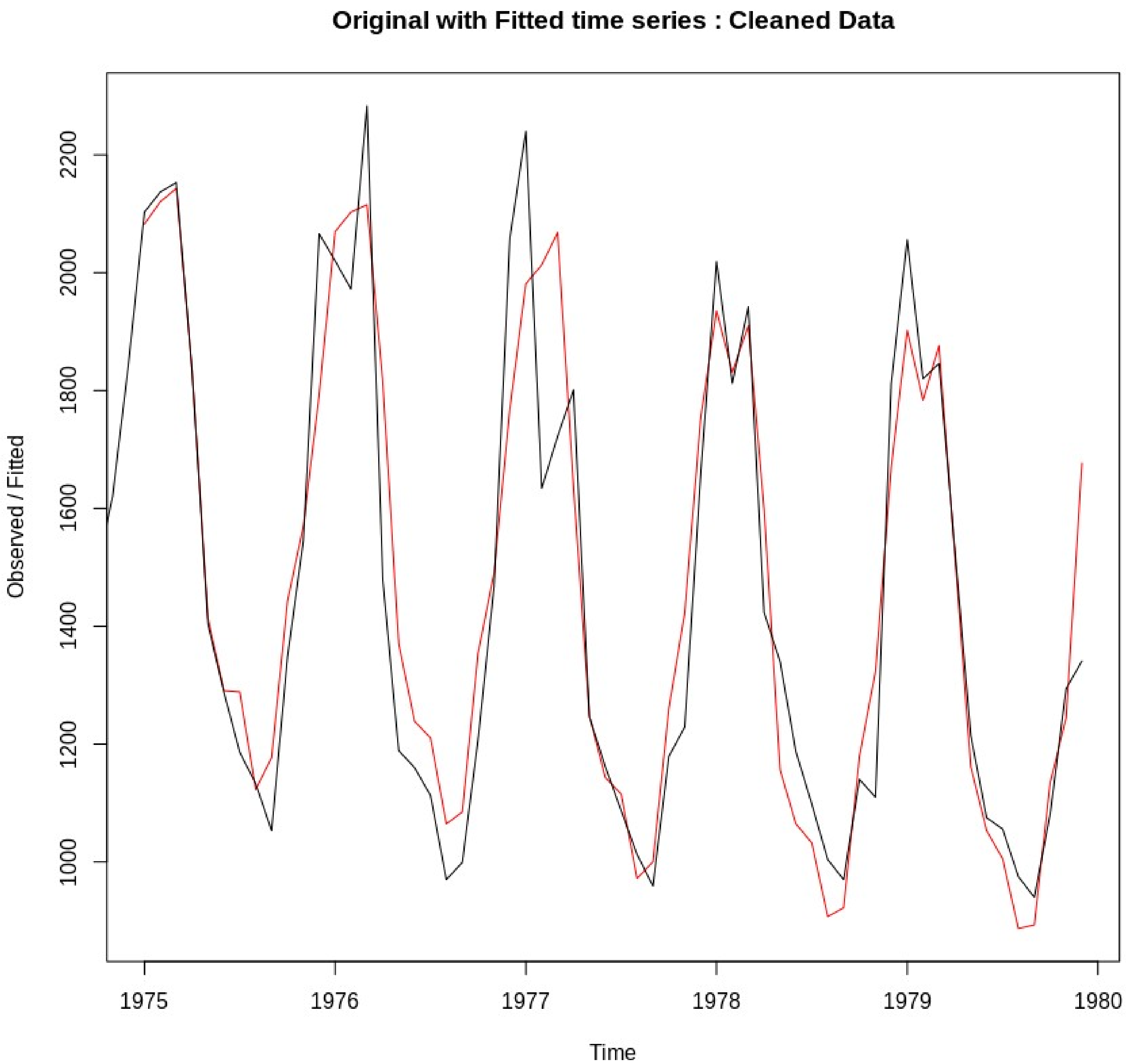
[1] 121.953

```
####### Q.17) NOT POSSIBLE IN THIS MODEL

####### Q.18) ARIMA MODEL IS GOOD AS ARIMA HAVE RMS ERROR VALUE OF (121.95) whereas HOTWINTER
MODEL HAVE ERROR VALUE OF (138.58)

####### Q.19) BELOW ARE THE PLOTS WITH AND WITHOUT CLEANING THE DATA
```

```
################# Q.20) CLEANING THE DATA ###############
Cleaned <- tsclean(my_Object)
modelcl <- HoltWinters(Cleaned)
model_without_cleaning <- HoltWinters(my_Object)
plot(model_without_cleaning, main = "Original with Fitted time series : Raw Data")
```



```
plot(modelcl, main = "Original with Fitted time series : Cleaned Data")
```




```
modelcl$SSE
######## OUTPUT
[1] 1186927

model_without_cleaning$SSE
######## OUTPUT
[1] 2005186
```