DA_ASSIGNMENT_3

TIME SERIES ANALYSIS OF UK DRIVER DEATHS

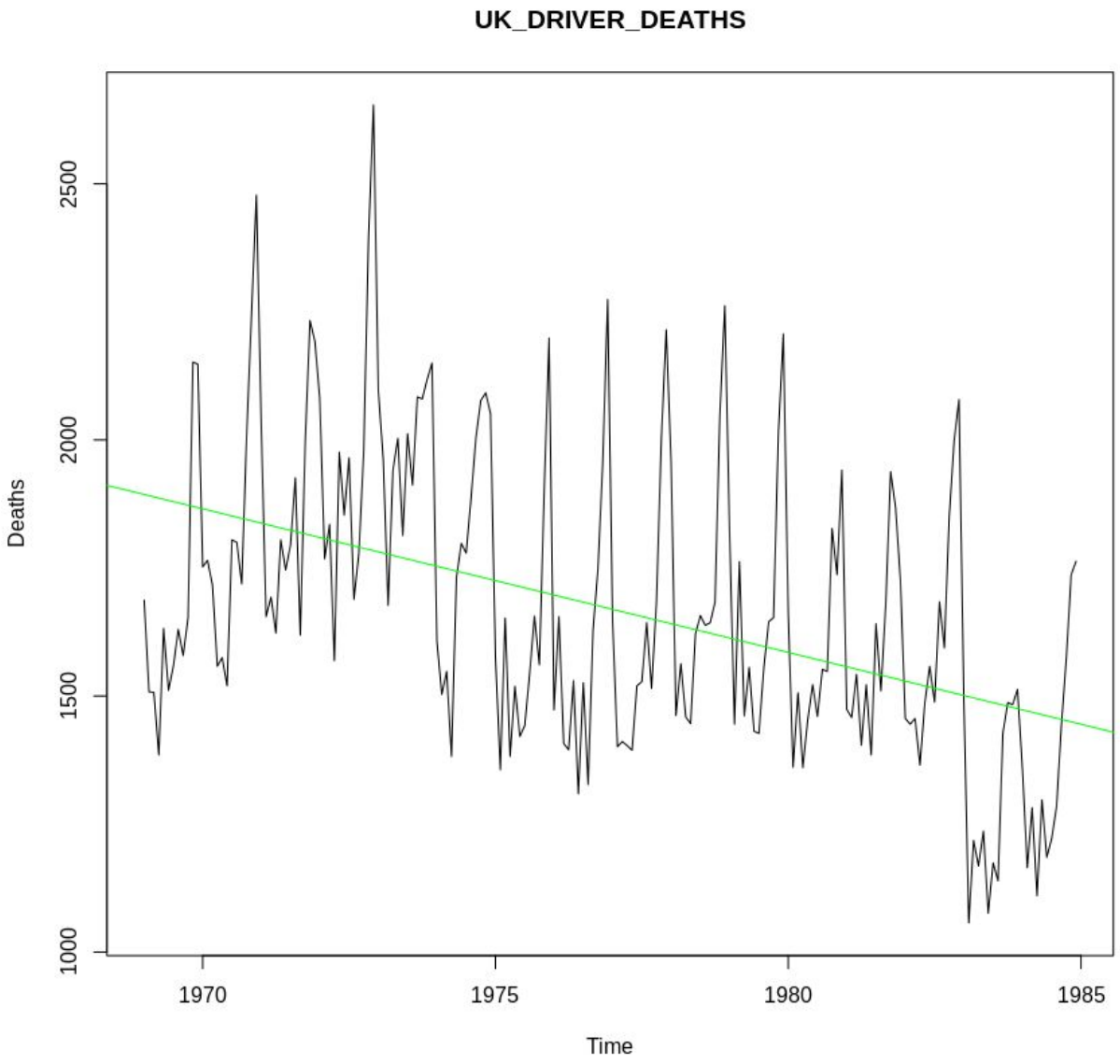
Name : Somyadeep Shrivastava
Roll No: 17BCS028

```
> ####CHOOSING DATASET
> library(ggplot2)
> library(Metrics)
> library(forecast)
> library(reshape)
> data("UKDriverDeaths")
> UKDriverDeaths
     Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
1969 1687 1508 1507 1385 1632 1511 1559 1630 1579 1653 2152 2148
1970 1752 1765 1717 1558 1575 1520 1805 1800 1719 2008 2242 2478
1971 2030 1655 1693 1623 1805 1746 1795 1926 1619 1992 2233 2192
1972 2080 1768 1835 1569 1976 1853 1965 1689 1778 1976 2397 2654
1973 2097 1963 1677 1941 2003 1813 2012 1912 2084 2080 2118 2150
1974 1608 1503 1548 1382 1731 1798 1779 1887 2004 2077 2092 2051
1975 1577 1356 1652 1382 1519 1421 1442 1543 1656 1561 1905 2199
1976 1473 1655 1407 1395 1530 1309 1526 1327 1627 1748 1958 2274
1977 1648 1401 1411 1403 1394 1520 1528 1643 1515 1685 2000 2215
1978 1956 1462 1563 1459 1446 1622 1657 1638 1643 1683 2050 2262
1979 1813 1445 1762 1461 1556 1431 1427 1554 1645 1653 2016 2207
1980 1665 1361 1506 1360 1453 1522 1460 1552 1548 1827 1737 1941
1981 1474 1458 1542 1404 1522 1385 1641 1510 1681 1938 1868 1726
1982 1456 1445 1456 1365 1487 1558 1488 1684 1594 1850 1998 2079
1983 1494 1057 1218 1168 1236 1076 1174 1139 1427 1487 1483 1513
1984 1357 1165 1282 1110 1297 1185 1222 1284 1444 1575 1737 1763
>
> ###################################
>
>
> #start end time frequency cycle - The deltat() function returns the fixed time interval between
observations and the frequency() function returns the number of observations per unit time.
Finally, the cycle() function returns the position in the cycle of each observation
> start(UKDriverDeaths)
[1] 1969    1
> end(UKDriverDeaths)
[1] 1984   12
> time(UKDriverDeaths)
       Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep      Oct      Nov      Dec
```

1969 1969.000 1969.083 1969.167 1969.250 1969.333 1969.417 1969.500 1969.583 1969.667
1969.750 1969.833 1969.917
1970 1970.000 1970.083 1970.167 1970.250 1970.333 1970.417 1970.500 1970.583 1970.667
1970.750 1970.833 1970.917
1971 1971.000 1971.083 1971.167 1971.250 1971.333 1971.417 1971.500 1971.583 1971.667
1971.750 1971.833 1971.917
1972 1972.000 1972.083 1972.167 1972.250 1972.333 1972.417 1972.500 1972.583 1972.667
1972.750 1972.833 1972.917
1973 1973.000 1973.083 1973.167 1973.250 1973.333 1973.417 1973.500 1973.583 1973.667
1973.750 1973.833 1973.917
1974 1974.000 1974.083 1974.167 1974.250 1974.333 1974.417 1974.500 1974.583 1974.667
1974.750 1974.833 1974.917
1975 1975.000 1975.083 1975.167 1975.250 1975.333 1975.417 1975.500 1975.583 1975.667
1975.750 1975.833 1975.917
1976 1976.000 1976.083 1976.167 1976.250 1976.333 1976.417 1976.500 1976.583 1976.667
1976.750 1976.833 1976.917
1977 1977.000 1977.083 1977.167 1977.250 1977.333 1977.417 1977.500 1977.583 1977.667
1977.750 1977.833 1977.917
1978 1978.000 1978.083 1978.167 1978.250 1978.333 1978.417 1978.500 1978.583 1978.667
1978.750 1978.833 1978.917
1979 1979.000 1979.083 1979.167 1979.250 1979.333 1979.417 1979.500 1979.583 1979.667
1979.750 1979.833 1979.917
1980 1980.000 1980.083 1980.167 1980.250 1980.333 1980.417 1980.500 1980.583 1980.667
1980.750 1980.833 1980.917
1981 1981.000 1981.083 1981.167 1981.250 1981.333 1981.417 1981.500 1981.583 1981.667
1981.750 1981.833 1981.917
1982 1982.000 1982.083 1982.167 1982.250 1982.333 1982.417 1982.500 1982.583 1982.667
1982.750 1982.833 1982.917
1983 1983.000 1983.083 1983.167 1983.250 1983.333 1983.417 1983.500 1983.583 1983.667
1983.750 1983.833 1983.917
1984 1984.000 1984.083 1984.167 1984.250 1984.333 1984.417 1984.500 1984.583 1984.667
1984.750 1984.833 1984.917
> cycle(UKDriverDeaths)
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
1969 1 2 3 4 5 6 7 8 9 10 11 12
1970 1 2 3 4 5 6 7 8 9 10 11 12
1971 1 2 3 4 5 6 7 8 9 10 11 12
1972 1 2 3 4 5 6 7 8 9 10 11 12
1973 1 2 3 4 5 6 7 8 9 10 11 12
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

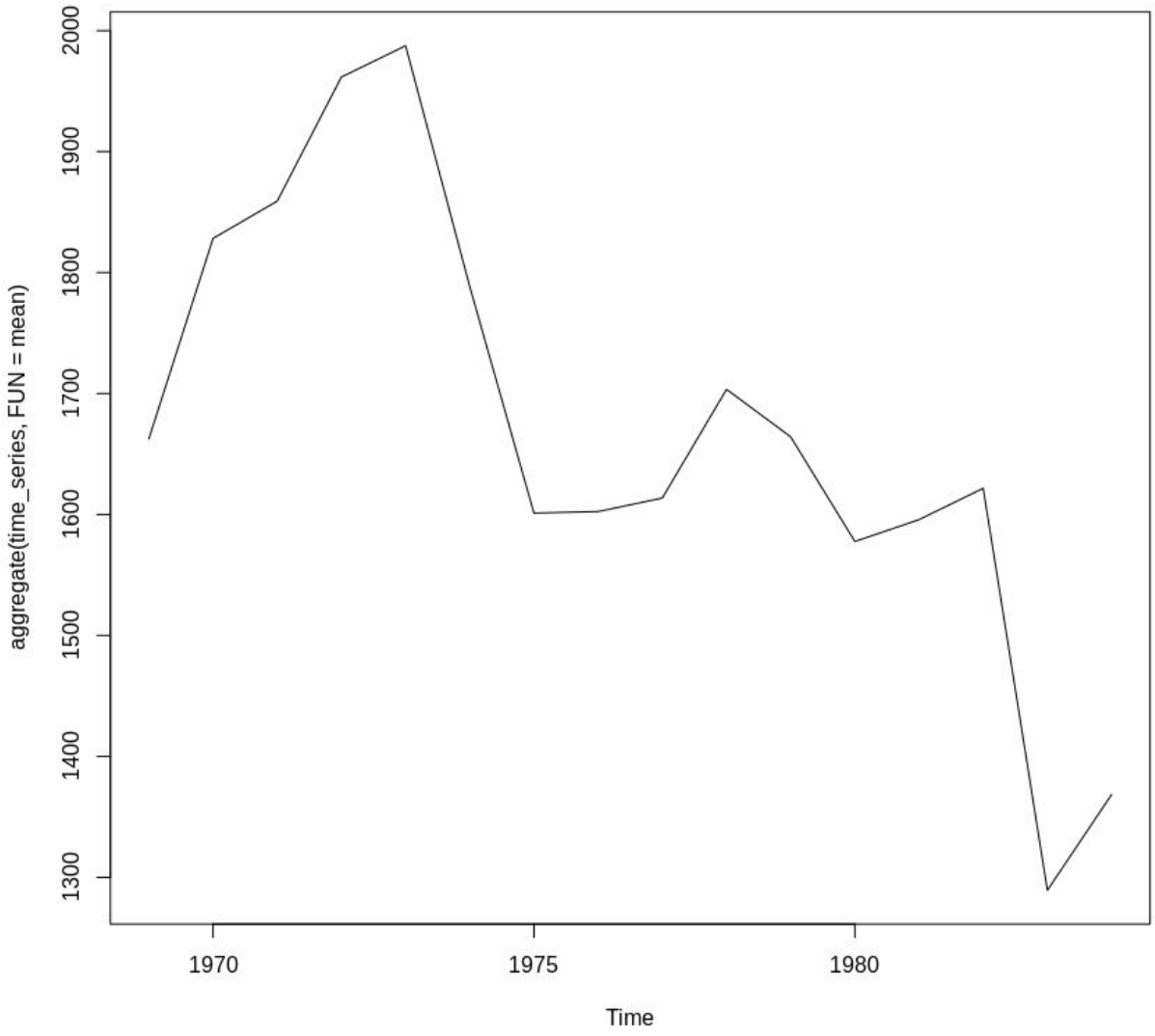
```
1978   1   2   3   4   5   6   7   8   9  10  11  12
1979   1   2   3   4   5   6   7   8   9  10  11  12
1980   1   2   3   4   5   6   7   8   9  10  11  12
1981   1   2   3   4   5   6   7   8   9  10  11  12
1982   1   2   3   4   5   6   7   8   9  10  11  12
1983   1   2   3   4   5   6   7   8   9  10  11  12
1984   1   2   3   4   5   6   7   8   9  10  11  12
> deltat(UKDriverDeaths)
[1] 0.08333333
>
>
> # Q2 - time series object
>
> time_series <- ts(UKDriverDeaths, start=1969 ,frequency = 12)
> #check whether it is a ts object
> is.ts(time_series)
[1] TRUE
> ts.plot(time_series, main="UK_DRIVER_DEATHS",ylab ="Deaths")
> abline(reg = lm(time_series~time(time_series)),col="green")
```

>



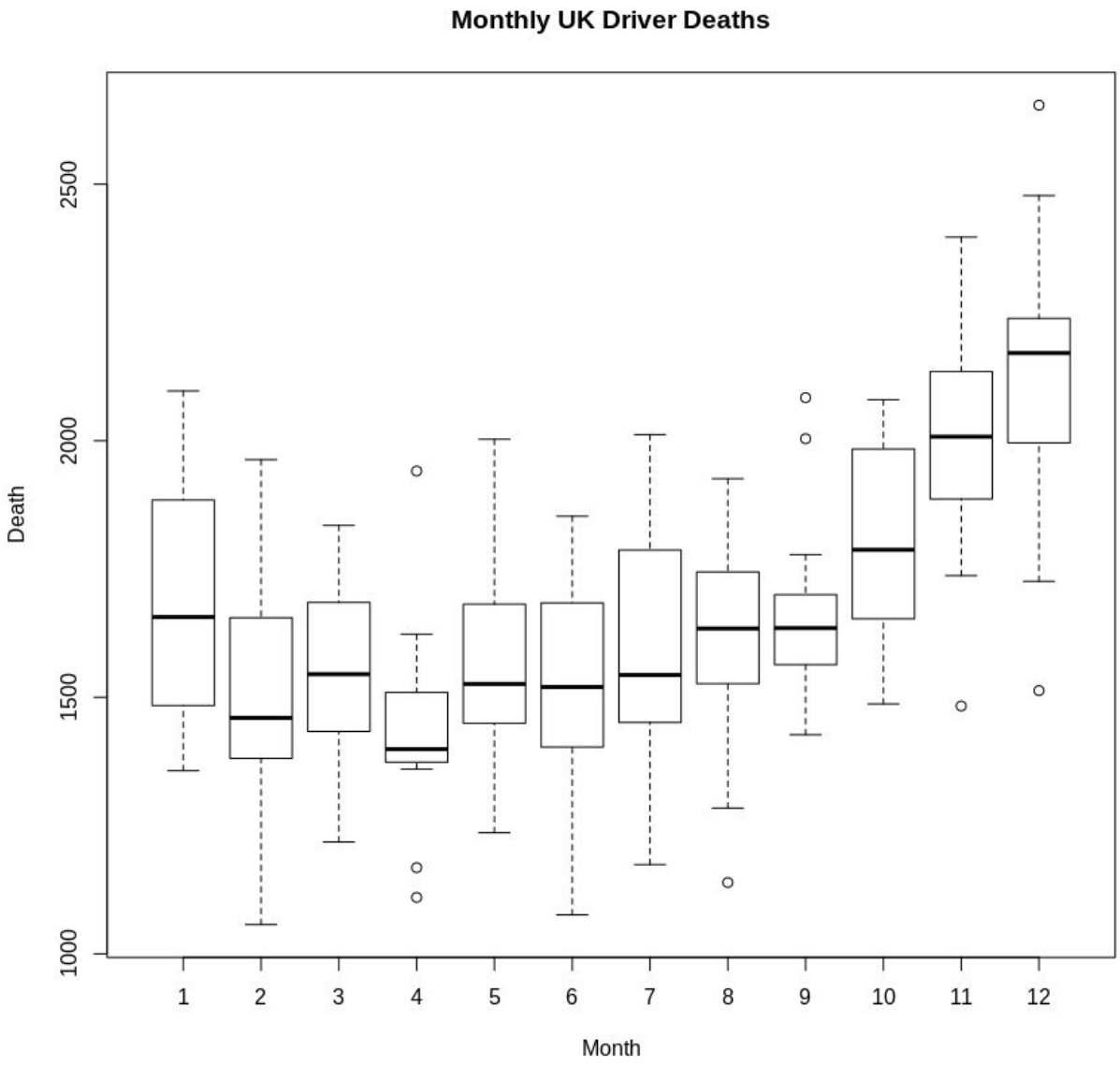
> #This property can stabilize variability when a series exhibits increasing variability over time. It may also be used to linearize a rapid growth pattern over time
>
> linear_growth <- log(time_series)
> ts.plot(linear_growth)
>
>
> #Q3 - Monthly mean values
>
> plot(aggregate(time_series,FUN = mean))

>



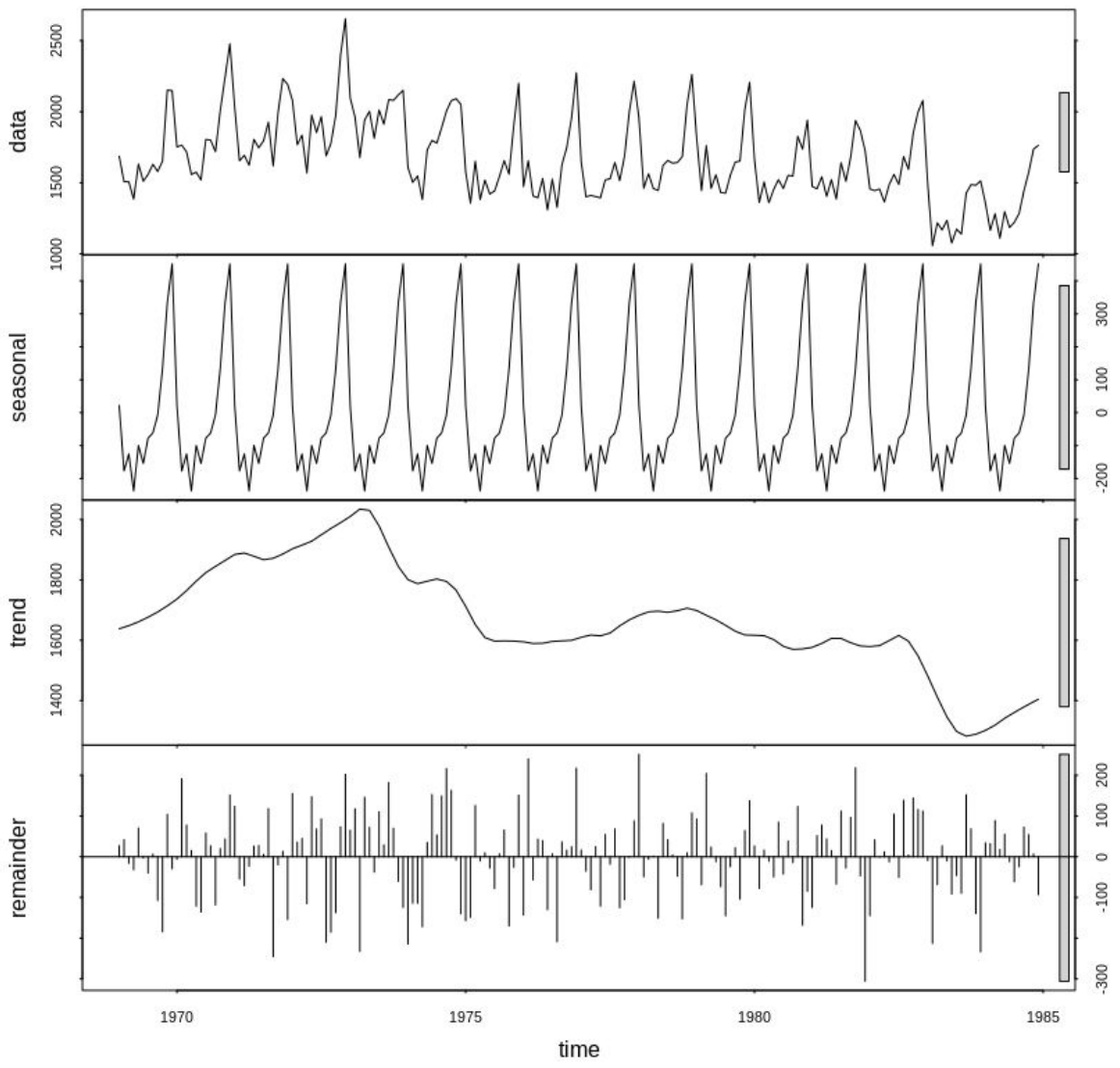
>
> #Q4 - Boxplot Monthly
>
> boxplot(time_series~cycle(time_series),xlab="Month",ylab = "Death",main = "Monthly UK Driver Deaths")

>



> #Q5 - decompose using stl and find TREND
>
>
> stlts <- stl(time_series, s.window = "periodic")
> plot(stlts)

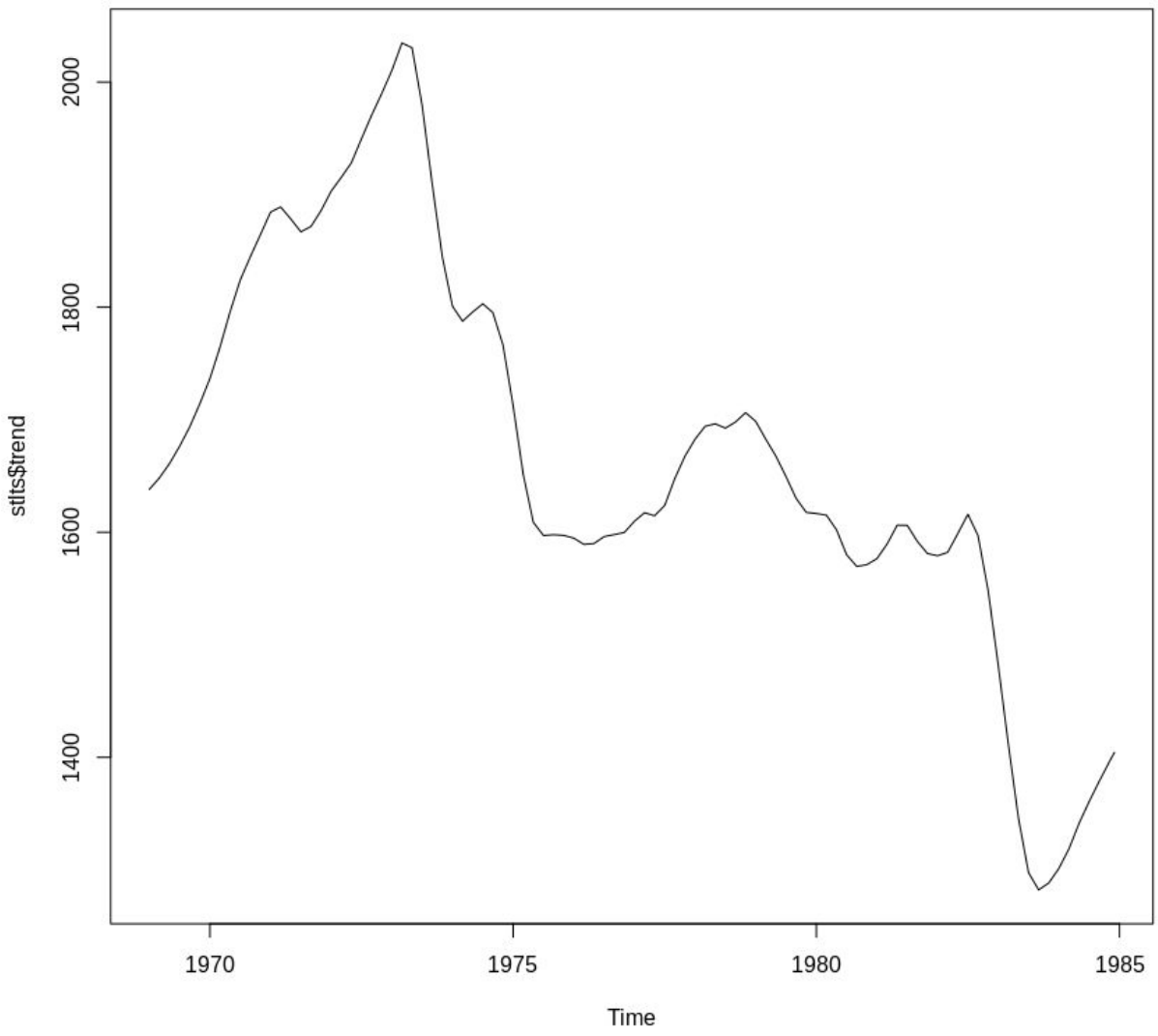
>



>

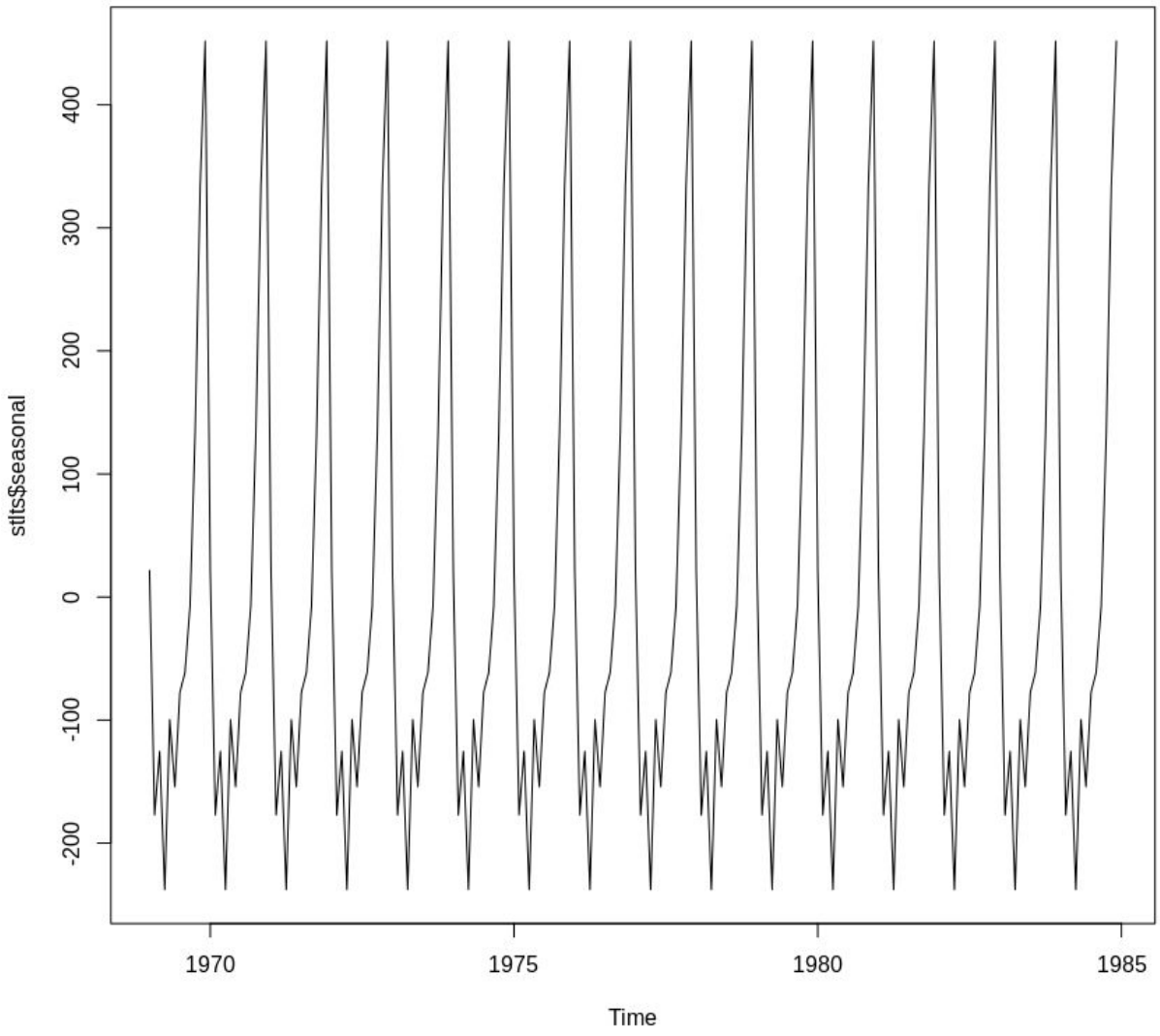
```
> stlts$trend <- stlts$time.series[,2]
> plot(stlts$trend)
```

>

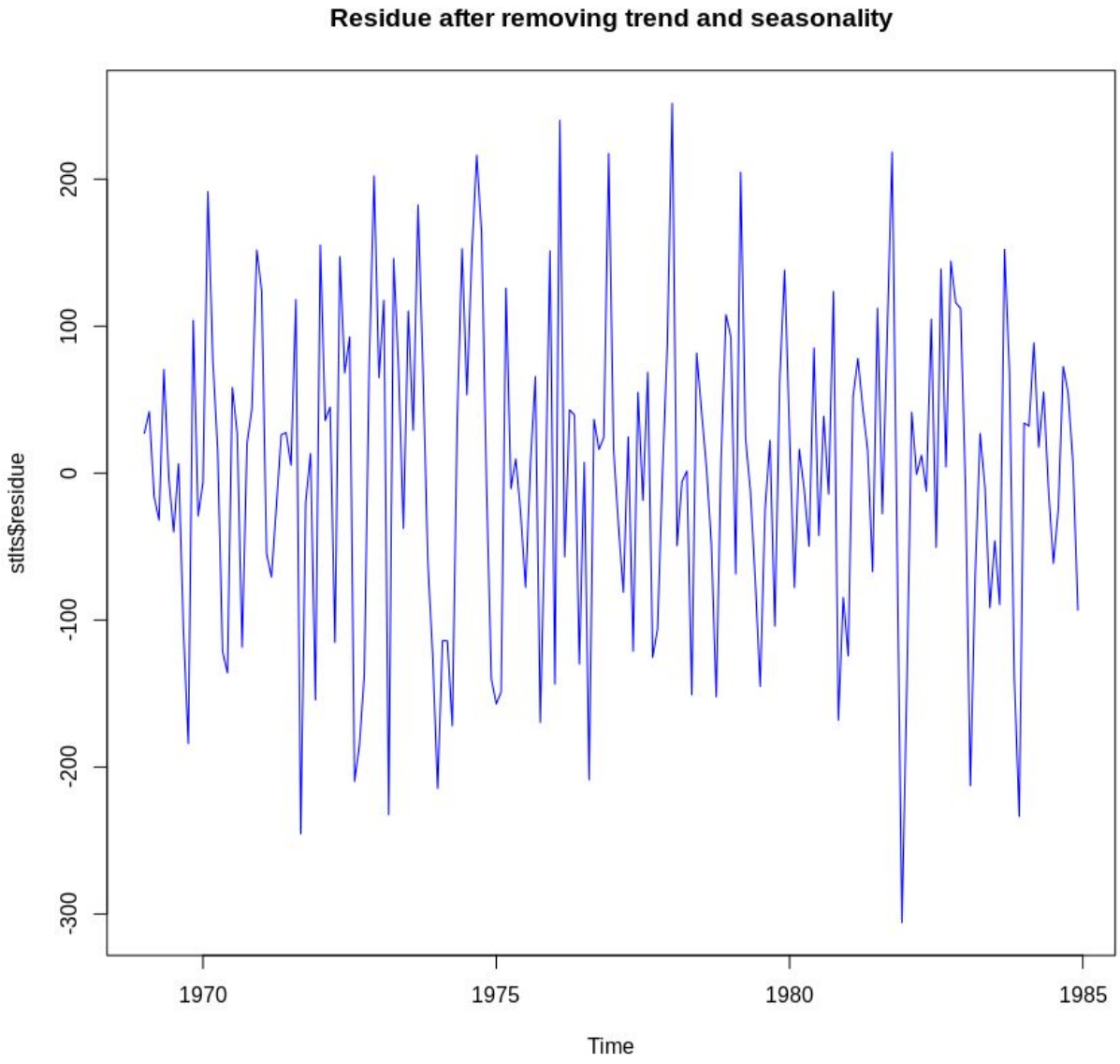


> # Trend is downward
>
> # Q6 - Seasonality
>
> stlts$seasonal <- stlts$time.series[,1]
> plot(stlts$seasonal)

>



> # Yearly quite uniform patterns
>
> #Q7 Residuals
>
>
> stlts$residue <- (time_series -(stlts$trend + stlts$seasona))
>
> plot(stlts$residue,main = "Residue after removing trend and seasonality",col = "blue")



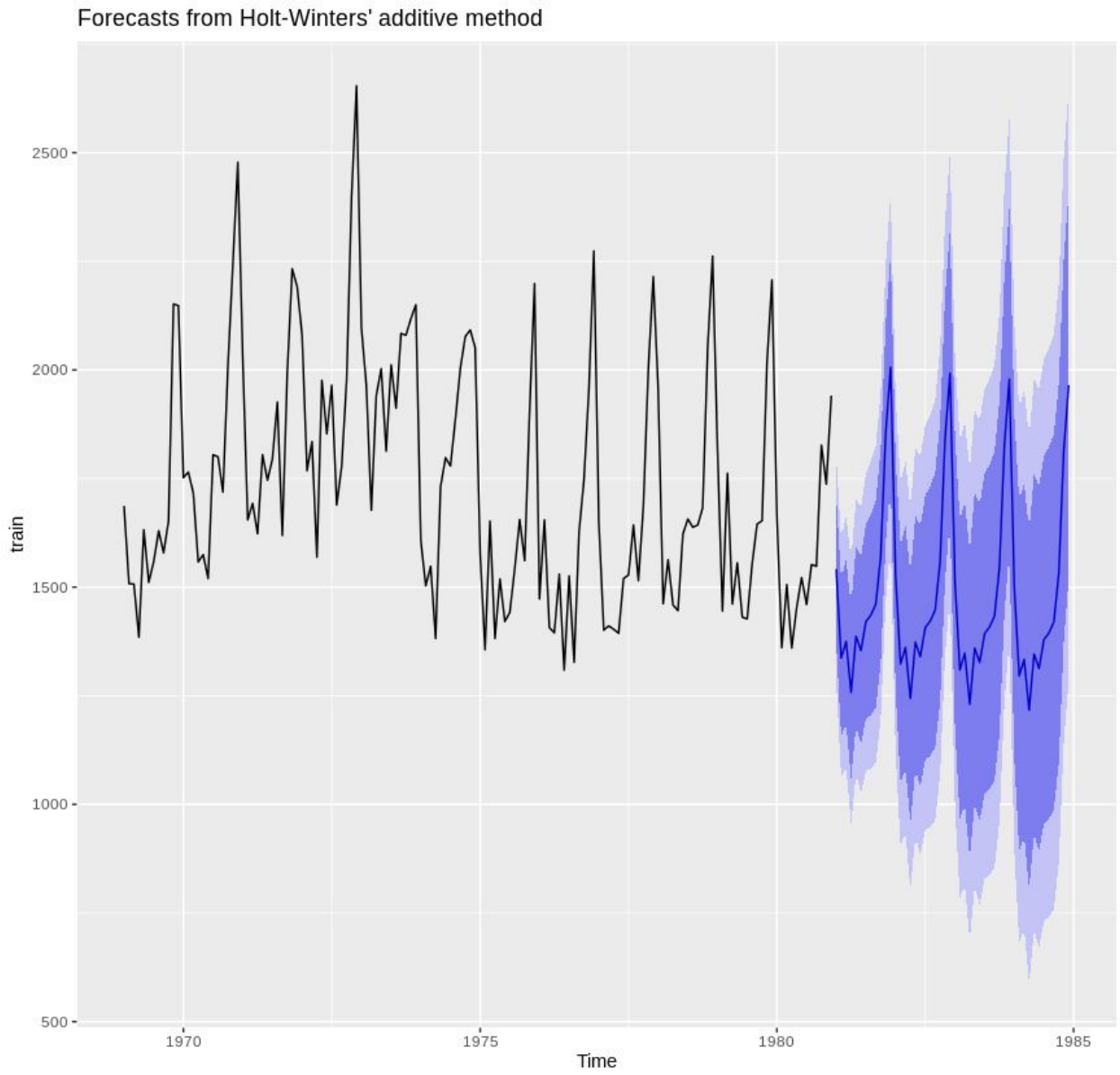

```
> #Q8 & Q9 75% trained holtwinter model and predicting 25%
>
> train <- window(UKDriverDeaths,start = c(1969,1) ,end=c(1980,12))
> train
     Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
1969 1687 1508 1507 1385 1632 1511 1559 1630 1579 1653 2152 2148
1970 1752 1765 1717 1558 1575 1520 1805 1800 1719 2008 2242 2478
1971 2030 1655 1693 1623 1805 1746 1795 1926 1619 1992 2233 2192
1972 2080 1768 1835 1569 1976 1853 1965 1689 1778 1976 2397 2654
1973 2097 1963 1677 1941 2003 1813 2012 1912 2084 2080 2118 2150
1974 1608 1503 1548 1382 1731 1798 1779 1887 2004 2077 2092 2051
1975 1577 1356 1652 1382 1519 1421 1442 1543 1656 1561 1905 2199
```

```
1976 1473 1655 1407 1395 1530 1309 1526 1327 1627 1748 1958 2274
1977 1648 1401 1411 1403 1394 1520 1528 1643 1515 1685 2000 2215
1978 1956 1462 1563 1459 1446 1622 1657 1638 1643 1683 2050 2262
1979 1813 1445 1762 1461 1556 1431 1427 1554 1645 1653 2016 2207
1980 1665 1361 1506 1360 1453 1522 1460 1552 1548 1827 1737 1941
> # here train data for 75% and test is 25% that is 48 months , fchw is prediction , summary give
whole detail about model ,rms values ,traindata ,testdata ,alpha,beta,gammavalues
> fchw <- hw(train, seasonal = "additive", h = 48)
> summary(fchw)

Forecast method: Holt-Winters' additive method

Model Information:
Holt-Winters' additive method

Call:
 hw(y = train, h = 48, seasonal = "additive")

  Smoothing parameters:
    alpha = 0.3525
    beta  = 1e-04
    gamma = 1e-04

  Initial states:
    l = 1760.89
    b = -1.1134
    s = 512.2312 359.2779 76.4419 -36.6632 -62.779 -79.1582
          -146.8413 -114.9296 -245.1559 -129.1682 -168.4493 35.1938

  sigma:  131.8163

     AIC     AICc      BIC
2138.538 2143.395 2189.025

Error measures:
                    ME     RMSE      MAE        MPE     MAPE      MASE       ACF1
Training set -1.879834 124.2776 99.58413 -0.5150968 5.723996 0.6750426 0.03779911

Forecasts:
         Point Forecast     Lo 80    Hi 80     Lo 95    Hi 95
Jan 1981       1541.770 1372.8403 1710.699 1283.4145 1800.125
Feb 1981       1336.974 1157.8503 1516.098 1063.0278 1610.921
Mar 1981       1375.113 1186.3384 1563.887 1086.4072 1663.819
```

Apr 1981	1257.989	1060.0286	1455.950	955.2344	1560.744
May 1981	1387.089	1180.3437	1593.833	1070.8997	1703.277
Jun 1981	1354.012	1138.8363	1569.187	1024.9293	1683.094
Jul 1981	1420.563	1197.2702	1643.856	1079.0660	1762.061
Aug 1981	1435.794	1204.6630	1666.924	1082.3099	1789.277
Sep 1981	1460.768	1222.0526	1699.484	1095.6842	1825.852
Oct 1981	1572.748	1326.6770	1818.820	1196.4148	1949.082
Nov 1981	1854.390	1601.1725	2107.608	1467.1271	2241.654
Dec 1981	2006.217	1746.0441	2266.389	1608.3172	2404.116
Jan 1982	1528.052	1261.0976	1795.005	1119.7808	1936.322
Feb 1982	1323.256	1049.6885	1596.824	904.8706	1741.642
Mar 1982	1361.395	1081.3657	1641.424	933.1273	1789.662
Apr 1982	1244.271	957.9227	1530.620	806.3388	1682.204
May 1982	1373.370	1080.8348	1665.906	925.9758	1820.765
Jun 1982	1340.294	1041.6957	1638.892	883.6274	1796.960
Jul 1982	1406.845	1102.3017	1711.389	941.0861	1872.604
Aug 1982	1422.075	1111.6968	1732.454	947.3922	1896.759
Sep 1982	1447.050	1130.9403	1763.160	963.6019	1930.498
Oct 1982	1559.030	1237.2881	1880.772	1066.9681	2051.092
Nov 1982	1840.672	1513.3913	2167.953	1340.1392	2341.205
Dec 1982	1992.498	1659.7674	2325.229	1483.6303	2501.367
Jan 1983	1514.333	1176.2338	1852.433	997.2547	2031.412
Feb 1983	1309.538	966.1535	1652.922	784.3768	1834.699
Mar 1983	1347.677	999.0843	1696.269	814.5507	1880.803
Apr 1983	1230.553	876.8265	1584.280	689.5748	1771.532
May 1983	1359.652	1000.8615	1718.443	810.9291	1908.375
Jun 1983	1326.576	962.7882	1690.363	770.2107	1882.940
Jul 1983	1393.127	1024.4077	1761.846	829.2195	1957.035
Aug 1983	1408.357	1034.7681	1781.947	837.0020	1979.713
Sep 1983	1433.332	1054.9326	1811.731	854.6200	2012.044
Oct 1983	1545.312	1162.1601	1928.464	959.3316	2131.292
Nov 1983	1826.954	1439.1048	2214.803	1233.7898	2420.118
Dec 1983	1978.780	1586.2870	2371.274	1378.5137	2579.047
Jan 1984	1500.615	1103.5269	1897.704	893.3211	2107.909
Feb 1984	1295.820	894.1888	1697.451	681.5782	1910.061
Mar 1984	1333.959	927.8329	1740.084	712.8430	1955.074
Apr 1984	1216.835	806.2613	1627.409	588.9167	1844.754
May 1984	1345.934	930.9572	1760.911	711.2816	1980.587
Jun 1984	1312.857	893.5209	1732.194	671.5375	1954.177
Jul 1984	1379.409	955.7550	1803.063	731.4861	2027.332
Aug 1984	1394.639	966.7088	1822.570	740.1761	2049.102
Sep 1984	1419.614	987.4466	1851.781	758.6711	2080.556
Oct 1984	1531.594	1095.2286	1967.959	864.2308	2198.957

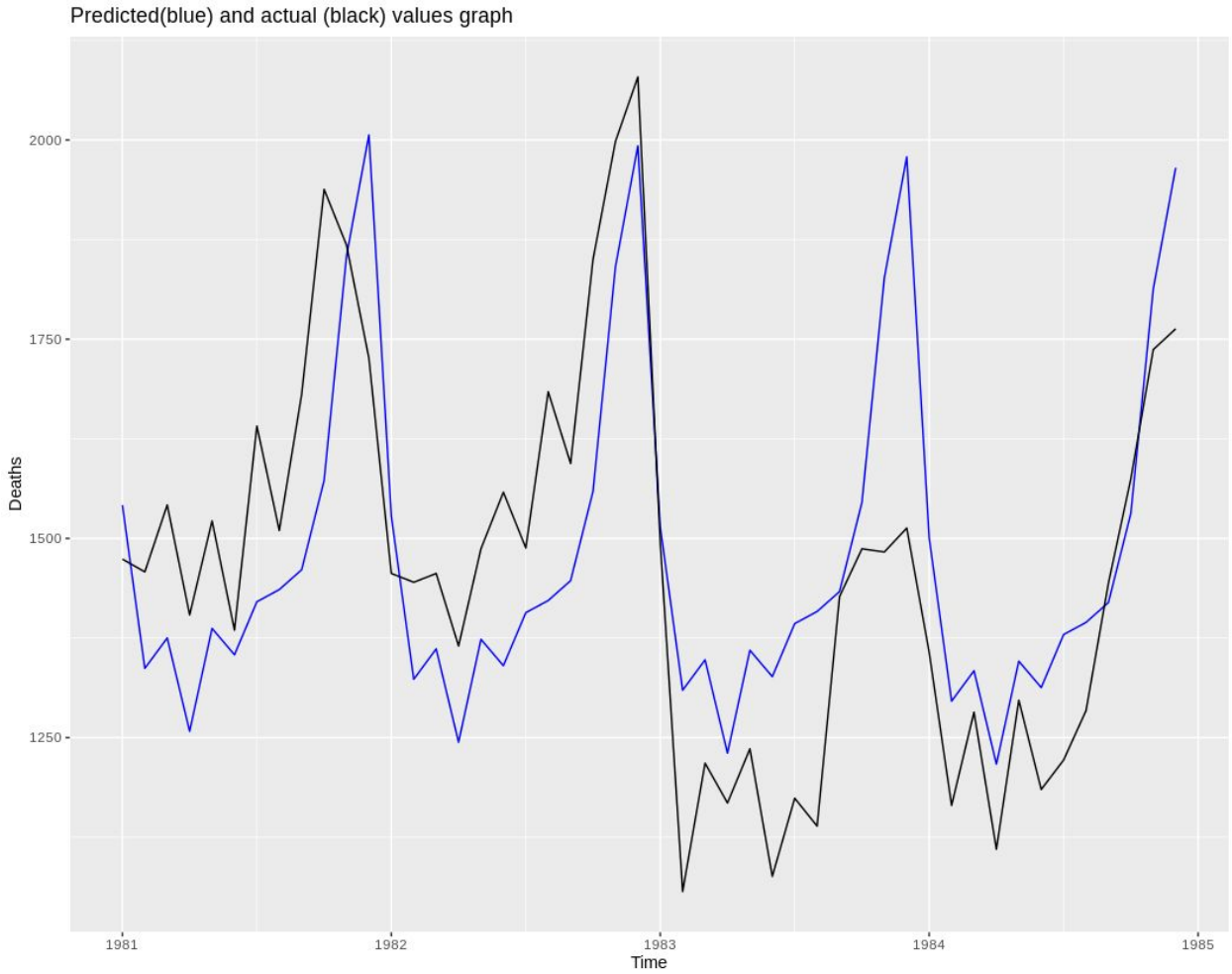
Nov 1984 1813.236 1372.7100 2253.762 1139.5096 2486.962
Dec 1984 1965.062 1520.4118 2409.713 1285.0281 2645.096

```
> autoplot(fchw)
>
```




```
>  #Q10 PLOT predicted and actual values
>
> act_value = tail(time_series,48)
>
>
>
> df = data.frame( fchw , tail(time_series,48))
>
```

```
>
> X = time(act_value)
> dfplt = as.data.frame(data.frame(df$Point.Forecast,df$tail.time_series..48.))
> ggplot(dfplt,aes(X))+
+  geom_line(aes(y=dfplt$df.Point.Forecast),colour = "blue")+
+  geom_line(aes(y=dfplt$df.tail.time_series..48.),colour = "black") + xlab("Time") +
ylab("Deaths") +
+  ggtitle("Predicted(blue) and actual (black) values graph")
```

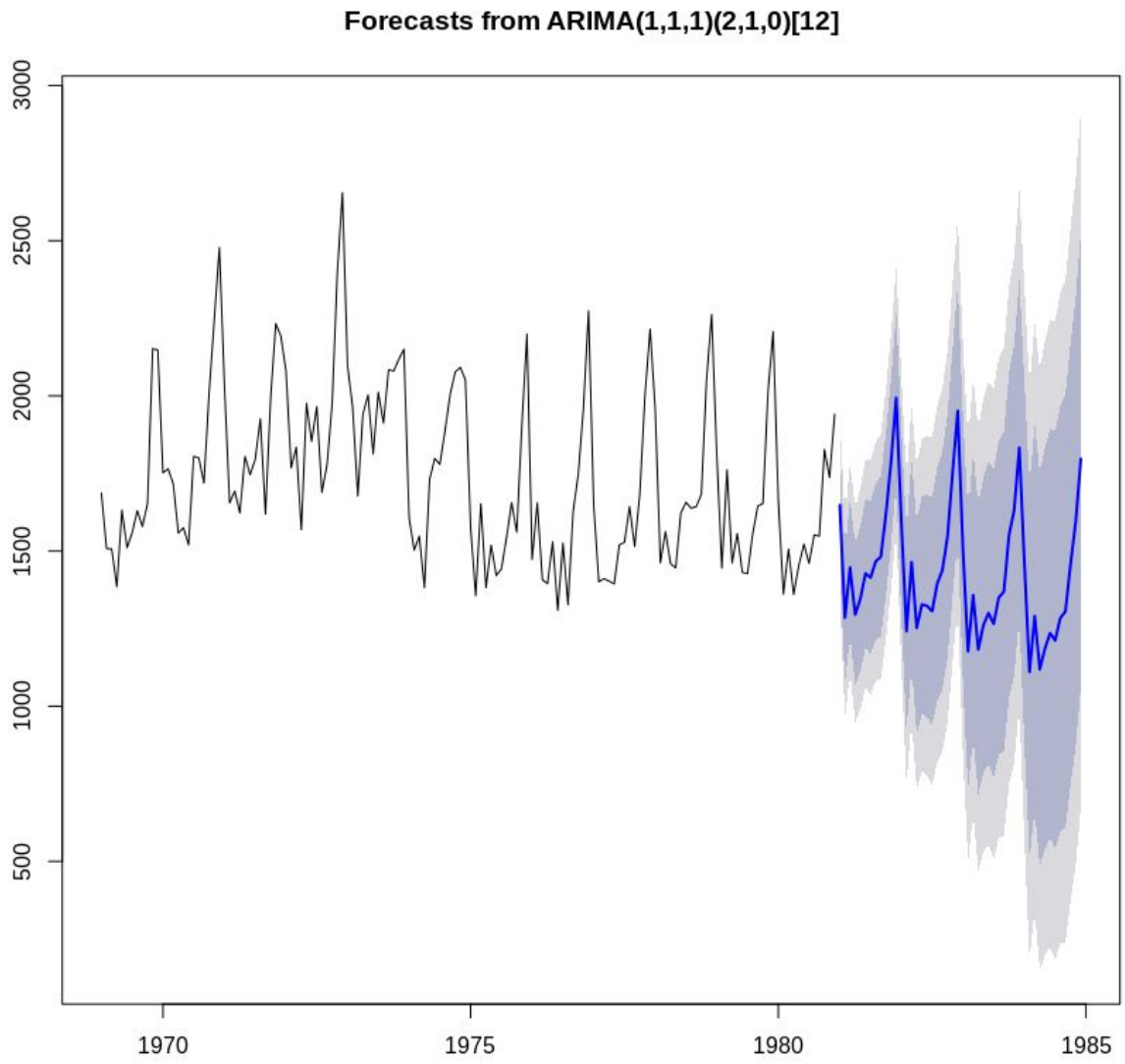



```
>
> #Q11 - RMS(predicted,actual)
>
> rmse(df$Point.Forecast,df$tail.time_series..48.)
[1] 176.7067
>
>
> #Q12  - Tuning Model
>
```

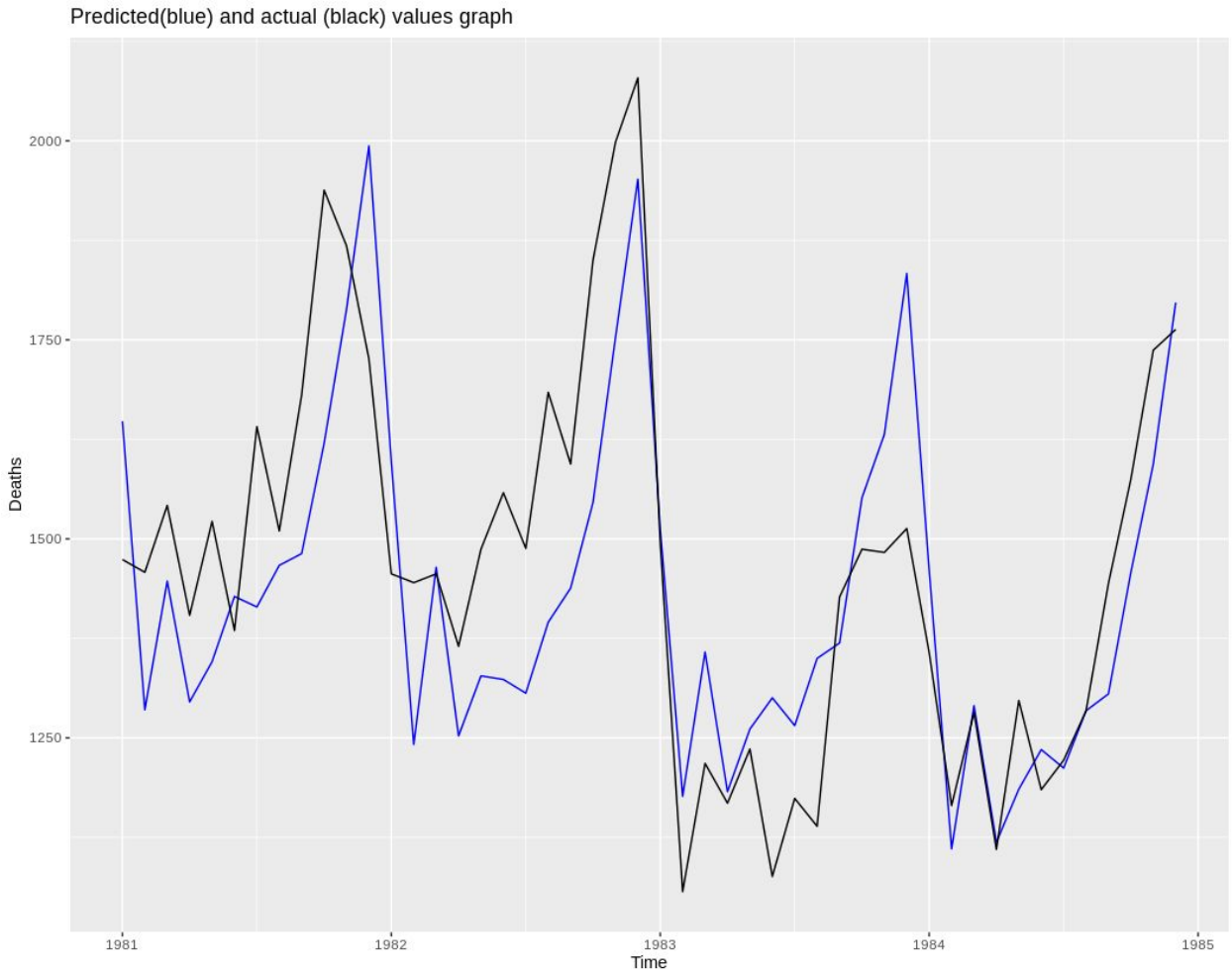
```
> #improved
>
> hw_modelt <- HoltWinters(train,alpha = "0.22" ,beta = "0.32" ,gamma = "0.82" )
>
>
> model.predict <- predict(hw_modelt,n.ahead = 48)
> round(model.predict)
     Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
1981 1593 1304 1511 1366 1462 1488 1427 1502 1494 1661 1667 1891
1982 1518 1229 1436 1291 1387 1413 1352 1427 1419 1586 1591 1816
1983 1442 1154 1361 1216 1312 1337 1277 1351 1344 1511 1516 1741
1984 1367 1078 1285 1140 1237 1262 1202 1276 1268 1436 1441 1665
> p_values= model.predict
>
> act_value = tail(time_series,48)
>
> rmse(act_value,p_values)
[1] 156.6058
> # SINCE RMSE LESS HENCE IMPROVED
>
>
> #Q13 - ARIMA MODEL - 75% train
>
> train <- window(UKDriverDeaths,start = c(1969,1) ,end=c(1980,12))
> train
     Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
1969 1687 1508 1507 1385 1632 1511 1559 1630 1579 1653 2152 2148
1970 1752 1765 1717 1558 1575 1520 1805 1800 1719 2008 2242 2478
1971 2030 1655 1693 1623 1805 1746 1795 1926 1619 1992 2233 2192
1972 2080 1768 1835 1569 1976 1853 1965 1689 1778 1976 2397 2654
1973 2097 1963 1677 1941 2003 1813 2012 1912 2084 2080 2118 2150
1974 1608 1503 1548 1382 1731 1798 1779 1887 2004 2077 2092 2051
1975 1577 1356 1652 1382 1519 1421 1442 1543 1656 1561 1905 2199
1976 1473 1655 1407 1395 1530 1309 1526 1327 1627 1748 1958 2274
1977 1648 1401 1411 1403 1394 1520 1528 1643 1515 1685 2000 2215
1978 1956 1462 1563 1459 1446 1622 1657 1638 1643 1683 2050 2262
1979 1813 1445 1762 1461 1556 1431 1427 1554 1645 1653 2016 2207
1980 1665 1361 1506 1360 1453 1522 1460 1552 1548 1827 1737 1941
> model = auto.arima(train)
> model
Series: train
ARIMA(1,1,1)(2,1,0)[12]
```

```
Coefficients:
         ar1      ma1     sar1     sar2
      0.2467  -0.7842  -0.5668  -0.4080
s.e.  0.1329   0.0882   0.0824   0.0814

sigma^2 estimated as 22213:  log likelihood=-842.97
AIC=1695.94   AICc=1696.42   BIC=1710.32
>
>
> # Q14 - Predict for next 25% data
> p <- forecast(model, h = 48)
> p
         Point Forecast     Lo 80    Hi 80     Lo 95    Hi 95
Jan 1981       1647.760 1456.7582 1838.761 1355.6482 1939.871
Feb 1981       1285.321 1074.8800 1495.763  963.4791 1607.163
Mar 1981       1446.909 1227.2366 1666.582 1110.9490 1782.869
Apr 1981       1295.201 1068.3132 1522.089  948.2061 1642.196
May 1981       1345.713 1112.2017 1579.224  988.5884 1702.837
Jun 1981       1427.656 1187.7912 1667.520 1060.8147 1794.496
Jul 1981       1414.469 1168.4367 1660.502 1038.1948 1790.744
Aug 1981       1466.751 1214.7059 1718.796 1081.2812 1852.221
Sep 1981       1481.516 1223.5997 1739.433 1087.0670 1875.965
Oct 1981       1619.960 1356.3037 1883.617 1216.7324 2023.188
Nov 1981       1788.367 1519.0928 2057.641 1376.5476 2200.187
Dec 1981       1993.566 1718.7885 2268.343 1573.3303 2413.801
Jan 1982       1597.262 1290.0234 1904.500 1127.3811 2067.143
Feb 1982       1241.838  920.8377 1562.838  750.9104 1732.766
Mar 1982       1464.195 1132.8837 1795.506  957.4982 1970.891
Apr 1982       1252.486  911.8162 1593.156  731.4764 1773.496
May 1982       1327.898  978.2697 1677.526  793.1878 1862.608
Jun 1982       1323.357  965.0314 1681.683  775.3451 1871.370
Jul 1982       1306.164  939.3554 1672.974  745.1784 1867.151
Aug 1982       1395.239 1020.1406 1770.337  821.5756 1968.902
Sep 1982       1438.125 1054.9170 1821.332  852.0591 2024.190
Oct 1982       1545.680 1154.5308 1936.829  947.4691 2143.890
Nov 1982       1752.425 1353.4930 2151.357 1142.3111 2362.539
Dec 1982       1951.640 1545.0744 2358.207 1329.8513 2573.430
Jan 1983       1512.269 1080.7548 1943.783  852.3249 2172.213
Feb 1983       1176.711  731.6359 1621.785  496.0276 1857.394
Mar 1983       1357.854  901.6287 1814.080  660.1175 2055.591
Apr 1983       1182.485  715.8344 1649.135  468.8047 1896.164
May 1983       1261.116  784.3786 1737.854  532.0090 1990.223
Jun 1983       1300.317  813.7275 1786.906  556.1427 2044.491
```

```
Jul 1983       1265.480  769.2412 1761.720  506.5480 2024.413
Aug 1983       1349.903  844.2000 1855.607  576.4968 2123.310
Sep 1983       1369.194  854.2005 1884.187  581.5794 2156.808
Oct 1983       1551.601 1027.4821 2075.719  750.0305 2353.171
Nov 1983       1631.192 1098.1040 2164.279  815.9044 2446.479
Dec 1983       1833.310 1291.4013 2375.218 1004.5324 2662.087
Jan 1984       1460.397  876.6153 2044.179  567.5798 2353.215
Feb 1984       1110.717  507.2805 1714.153  187.8406 2033.593
Mar 1984       1290.429  671.6638 1909.195  344.1092 2236.749
Apr 1984       1118.940  486.0508 1751.829  151.0195 2086.861
May 1984       1185.588  539.0813 1832.095  196.8413 2174.335
Jun 1984       1235.278  575.4821 1895.073  226.2074 2244.348
Jul 1984       1212.077  539.2663 1884.887  183.1020 2241.052
Aug 1984       1284.126  598.5504 1969.702  235.6286 2332.623
Sep 1984       1305.319  607.2119 2003.425  237.6566 2372.980
Oct 1984       1457.899  747.4824 2168.315  371.4107 2544.387
Nov 1984       1593.924  871.4072 2316.441  488.9300 2698.918
Dec 1984       1796.838 1062.4202 2531.255  673.6431 2920.032
> plot(UKDriverDeaths)
> plot(p)
```

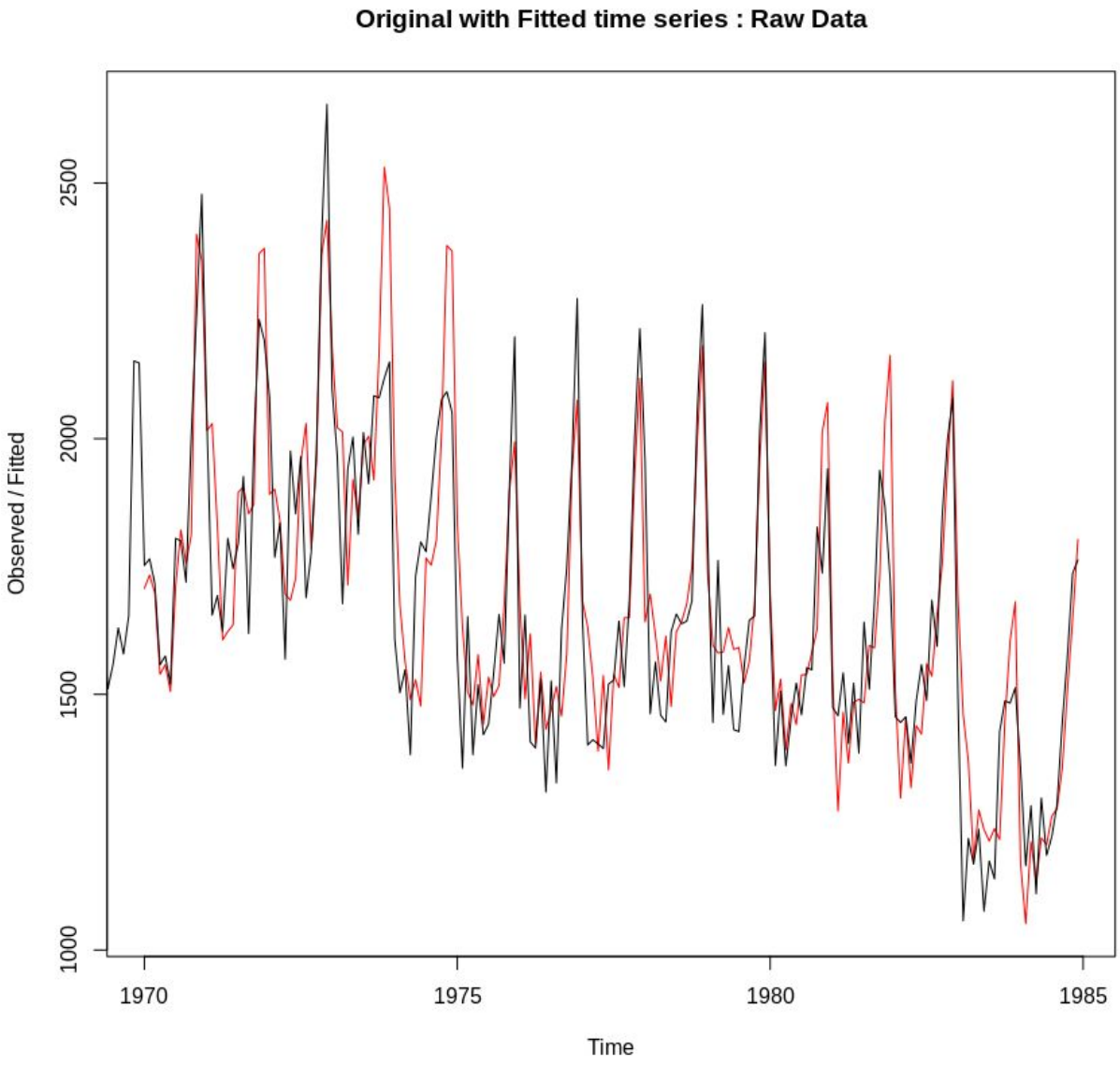



```
> predicted = data.frame(p)
> arima_act_values = tail(time_series,48)
>
> #Q15 Plotting predicted and actual
> X = time(arima_act_values)
> arima_df <- as.data.frame(data.frame(X,predicted$Point.Forecast,arima_act_values))
>
> ggplot(arima_df,aes(X))+
+  geom_line(aes(y=predicted$Point.Forecast),colour="blue")+
+  geom_line(aes(y=arima_act_values),colour = "black") + xlab("Time") + ylab("Deaths")+
+  ggtitle("Predicted(blue) and actual (black) values graph")
```

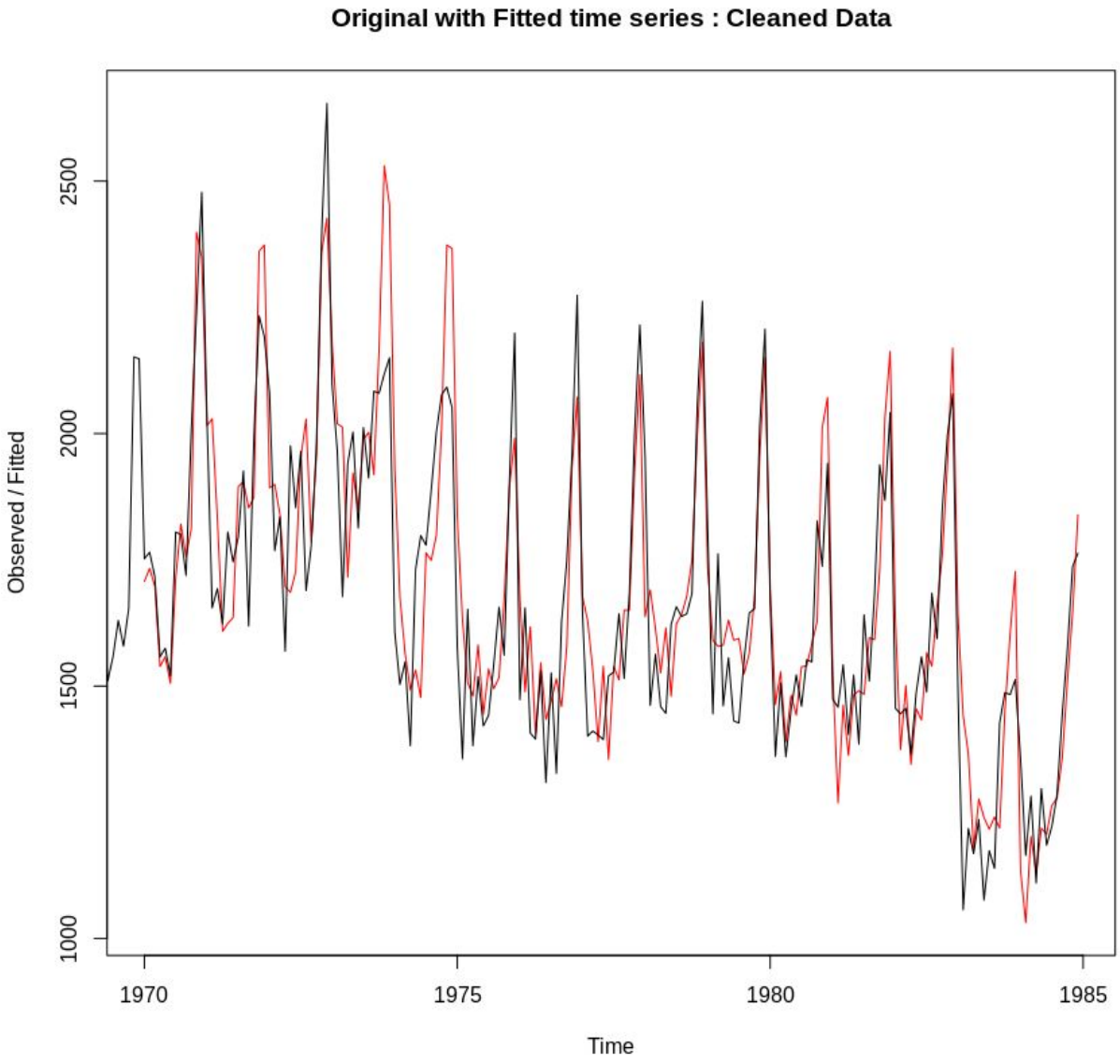



```
>
>
>
>
>
> # Q16 -  RMSE
> #TWO WAYS TO FIND
> rmse(arima_act_values,predicted$Point.Forecast)
[1] 158.3609
> forecast::accuracy(p,UKDriverDeaths)[,'RMSE']
Training set     Test set
   139.9655     158.3609
>
> # Q17 - Tuning MOdel
>
> #NOT POSSIBLE IN THIS CASE  AS AUTOTUNING DIDN't HELP
> #ORIGINAL VALUES WERE P,D,Q = 1,1,1
> #Tried others like - (1,0,1) ,(0,1,0) ,(0,0,0) etc.
```

```
>
> # Q18 - Which model better
>
> # ARIMA IS BETTER ,AS LESS RMS ERROR THEN HOLTWINTER
>
>
> # Q19 Cleaning data
> tscl <- tsclean(time_series)
> modelcl <- HoltWinters(tscl)
> model_without_cleaning <- HoltWinters(time_series)
> plot(model_without_cleaning, main = "Original with Fitted time series : Raw Data")
```




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




```
> modelcl$SSE
[1] 3752961
> model_without_cleaning$SSE
[1] 3904794
> # hence model after cleaning had less sum of squared error as compared to model without cleaning , also plot little bit  different
```