Assignment 2 - MATH1307 - Forecasting

Code

Student Details

- Saurabh Mallik (S3623575)

Introduction

TASK 1:
The objective of task 1 is to analyse and forecast the horizontal solar radiation reaching the ground at a particular location. We need to provide best 2 years forecasts using the following: a. DLM Models (DLM, POLYDLM, KOYCK etc.) b. Dynlm Models (SES, Holt's etc.) c. ETS Models (AAA, MAA etc.)

The purpose of task 1 research is to understand and analyse which model in each of the three categories best fits the series and projects the forecasts.

The data to undertake this task are 1. DATA1.CSV (2 variables ie Solar Radiation and Precipitation and 660 observations) and 2. Data.X.CSV (predictor series with 1 variable and 24 observations).

TASK 2:
The objective of this task is to analyse the correlation between quarterly residential property price index (PPI) and quartely population change over previous quarter in Victoria between September 2003 and December 2016. The dataset provided to carry out this investigation is Data2.csv.

Tha main aim of task 2 is to identify whether the correlation between the two is spurious or not.

Methodology

To undertake this research, forecasting methods on R Studio are being used to infer from the dataset.

Research and Inferences

1. Task 1

Reading in the Data and preparing for analysis and converting to time series.

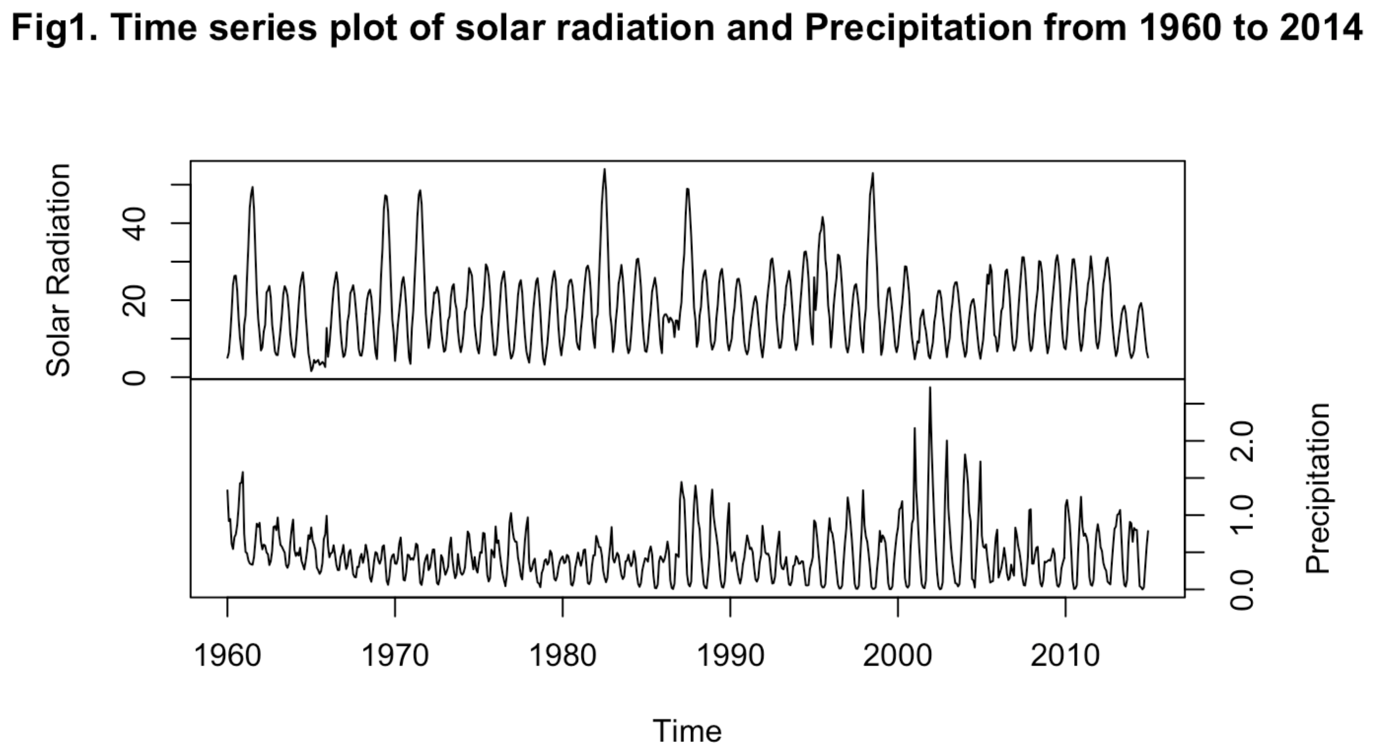
Hide

```
data1 <- read_csv("~/Desktop/Forecasting - Ass 2/data1.csv")
datax <- read_csv("~/Desktop/Forecasting - Ass 2/data.x.csv")
datax.ts = ts(datax, start = c(2015,1), end = c(2016,12), frequency = 12)
solar = ts(data1$solar,start = c(1960,1), end = c(2014,12), frequency = 12)
ppt = ts(data1$ppt,start = c(1960,1), end = c(2014,12), frequency = 12)
data1.ts = ts(data1[,1:2])
```

Plotting the time series and checking for correlation.

Hide

```
data.int = ts.intersect(solar , ppt)
colnames(data.int) =  c("Solar Radiation","Precipitation")
plot(data.int , yax.flip=T, main = "Fig1. Time series plot of solar radiation and Pre
cipitation from 1960 to 2014")
```

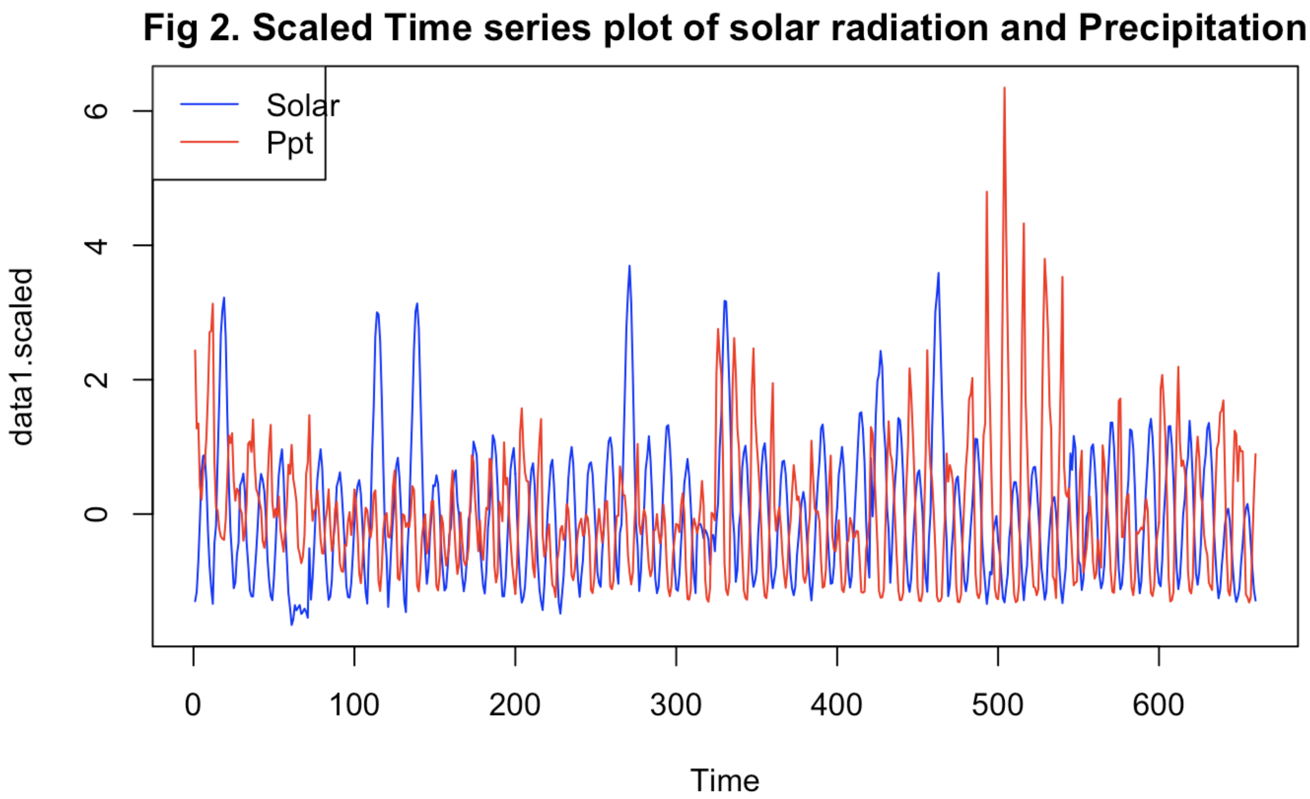


From Fig. 1 it is not easy to discern any correlation, hence we must try to scale and fit plots in one graph.

Plotting a scaled timeseries to check for correlation

Hide

```
data1.scaled = scale(data1.ts)
plot(data1.scaled, plot.type="s",col = c("blue", "red"), main = "Fig 2. Scaled Time s
eries plot of solar radiation and Precipitation")
legend("topleft",lty=1, text.width = 28, col=c("Blue","red"), c("Solar", "Ppt"))
```



The two series appear to follow each other in an inverse manner. We expect to see some negative correlation between the two variables.

Checking for correlation between the two variables.

Hide

```
cor(data1.ts)
```

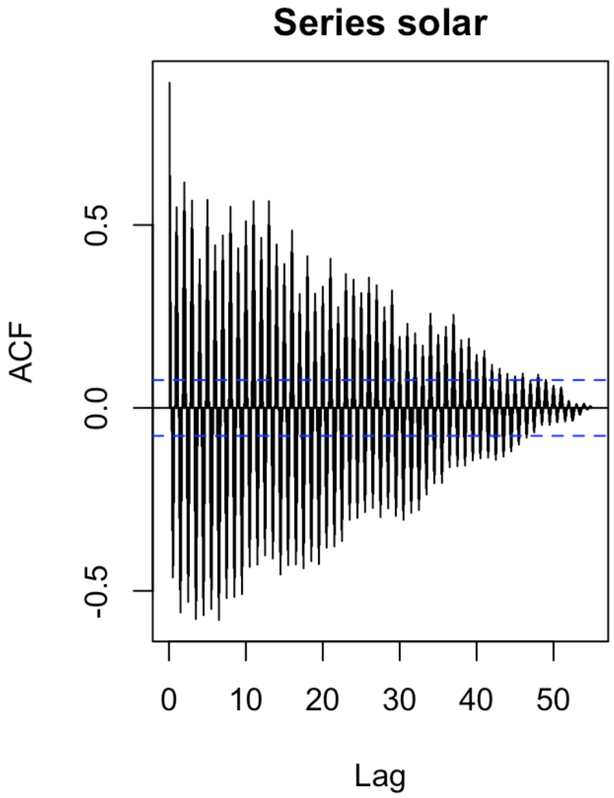
```
            solar        ppt
solar  1.0000000 -0.4540277
ppt   -0.4540277  1.0000000
```

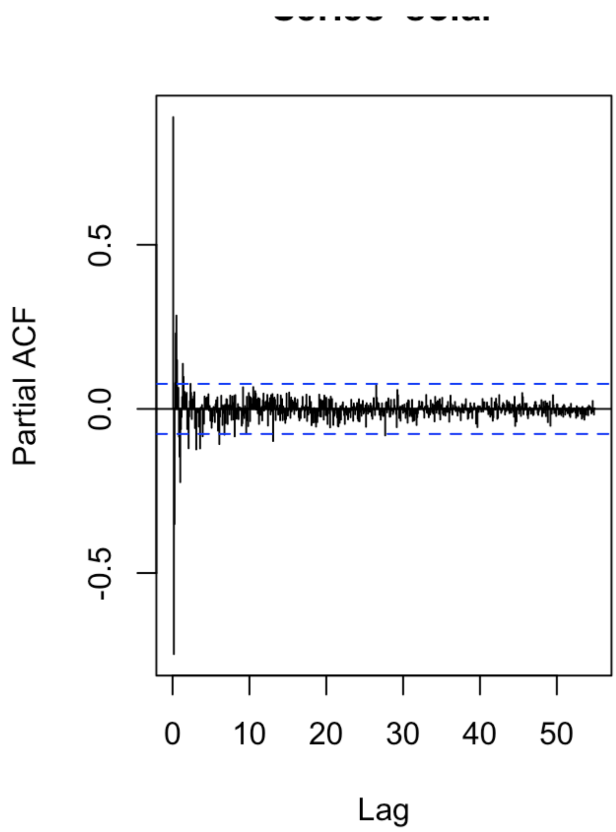
Since there is negative correlation, we can infer that the series are inversely related to each other.

We also conduct an ACF and PACF test to test for trend and seasonality.

Hide

```
par(mfrow = c(1,2))
acf(solar, "ACF for Solar Radiation")
pacf(solar, "PACF for Solar Radiation")
```





From the ACF we can see the slowly decomposing lags showing existence of trend and seasonality because of the presence of curves. The high 1st lag in PACF also shows evidence of trend.

In order to get the best fit forecasts, we need to model using DLM, Dynlm and ETS each, to find the best suitable fits.

1. DLM Fitting

We first attempt using dlm fitting.

Hide

```
model1p = dlm(x = as.vector(ppt) , y = as.vector(solar) , q = 12 , show.summary = TRU
E)
```

```

Call:
lm(formula = y.t ~ ., data = design)

Residuals:
    Min      1Q  Median      3Q     Max 
-18.563  -5.239  -0.796   4.137  32.430 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  19.5164     1.1151  17.501  < 2e-16 ***
x.t          -5.8876     1.9508  -3.018  0.00265 ** 
x.1           0.9993     2.5647   0.390  0.69694    
x.2           0.4343     2.5571   0.170  0.86520    
x.3           1.8763     2.5580   0.734  0.46352    
x.4           1.7459     2.5587   0.682  0.49529    
x.5           3.3279     2.5601   1.300  0.19410    
x.6           0.7751     2.5617   0.303  0.76230    
x.7           1.7937     2.5615   0.700  0.48402    
x.8           0.2827     2.5593   0.110  0.91207    
x.9          -1.1022     2.5615  -0.430  0.66712    
x.10         -1.9333     2.5508  -0.758  0.44880    
x.11         -0.5613     2.5532  -0.220  0.82605    
x.12         -5.3492     1.9216  -2.784  0.00553 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 8.181 on 634 degrees of freedom
Multiple R-squared:  0.3216,	Adjusted R-squared:  0.3077 
F-statistic: 23.12 on 13 and 634 DF,  p-value: < 2.2e-16

AIC and BIC values for the model:
       AIC      BIC
1 4578.787 4645.895
```

Hide

```
vif(model1p$model)
```

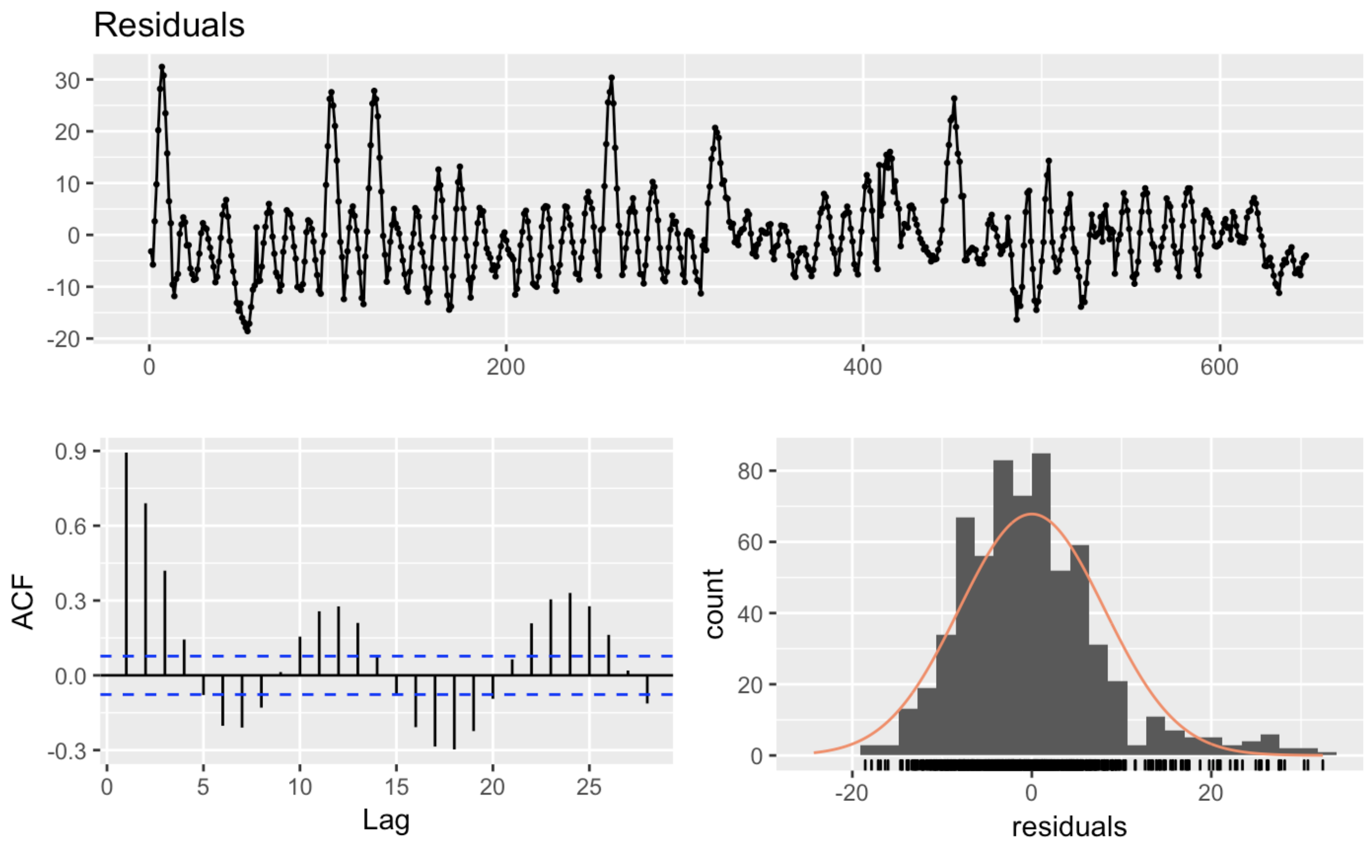
```
     x.t      x.1      x.2      x.3      x.4      x.5      x.6      x.7      x.8      
x.9     x.10     x.11     x.12 
4.432762 7.774629 7.820758 7.914873 7.941510 7.944820 7.943359 7.929999 7.916836 7.92
1508 7.867385 7.889225 4.508273 
```

We see that as we increase lags from 1 through 12, at q = 2 the R squared value is coming to .3077 and hence its not a very good fit. We also find that the lags are not affected by multicollinearity, as values of VIF test are below 10.

We continue to check residuals.

Hide

```
checkresiduals(model1p$model$residuals)
```



The residuals seem to be slightly random, but there are many extreme lags in the ACF test, suggesting evidence of serial correlation.

Hide

```
bgtest(model1p$model)
```

```


	Breusch-Godfrey test for serial correlation of order up to 1

data:  model1p$model
LM test = 527.93, df = 1, p-value < 2.2e-16
```

The Breusch-Godfrey test having a significant p value (< 2.2e-16) shows that the model is significant and fits the data well, although some coefficients are insignificant.

We next try the polydlm fitting.

Hide

```
model2p = polyDlm(x = as.vector(ppt) , y = as.vector(solar) , q = 12 , k = 2, show.be
ta = TRUE , show.summary = TRUE)
```

```

Call:
lm(formula = y.t ~ ., data = z)

Residuals:
    Min      1Q  Median      3Q     Max 
-18.689  -5.452  -0.686   4.129  32.797 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 19.22679    1.10784  17.355  < 2e-16 ***
z.t0        -1.81945    0.39287  -4.631  4.4e-06 ***
z.t1         1.57478    0.13106  12.015  < 2e-16 ***
z.t2        -0.15708    0.01019 -15.409  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.175 on 644 degrees of freedom
Multiple R-squared:  0.3119,    Adjusted R-squared:  0.3087 
F-statistic: 97.31 on 3 and 644 DF,  p-value: < 2.2e-16

Estimates and t-tests for beta coefficients:
        Estimate Std. Error t value  P(>|t|)
beta.0    -1.820      0.393   -4.63 4.41e-06
beta.1    -0.402      0.311   -1.29 1.97e-01
beta.2     0.702      0.261    2.69 7.42e-03
beta.3     1.490      0.240    6.22 9.28e-10
beta.4     1.970      0.237    8.31 5.71e-16
beta.5     2.130      0.239    8.89 6.28e-18
beta.6     1.970      0.240    8.22 1.18e-15
beta.7     1.510      0.237    6.36 3.90e-10
beta.8     0.726      0.232    3.13 1.84e-03
beta.9    -0.370      0.233   -1.58 1.14e-01
beta.10   -1.780      0.254   -7.01 5.94e-12
beta.11   -3.500      0.304  -11.50 4.37e-28
beta.12   -5.540      0.386  -14.30 1.31e-40
```

Hide

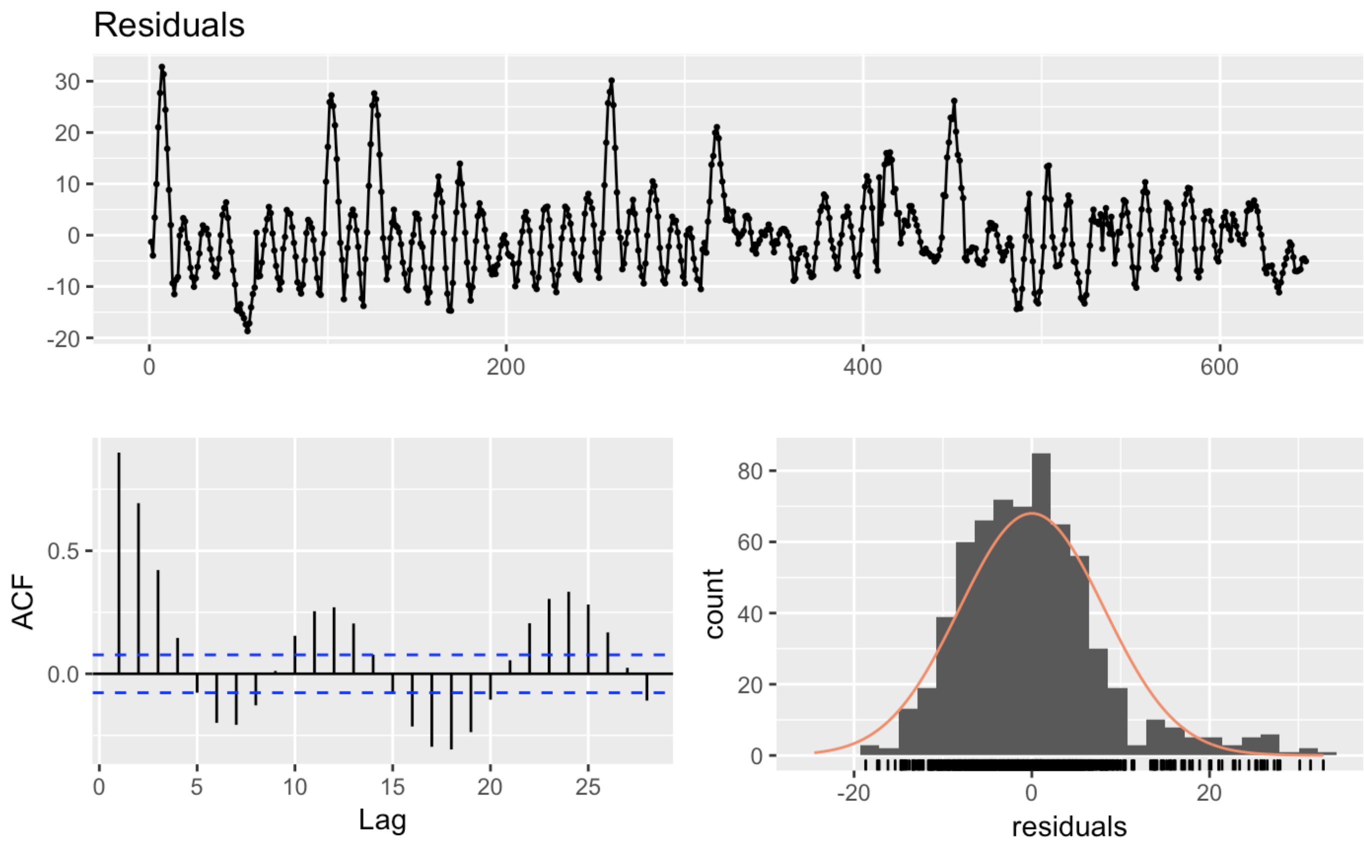
```
vif(model2p$model)
```

```
     z.t0      z.t1      z.t2 
 5.115134 28.849300 17.496603 
```

Model and all parameters are significant, but R squared value of fit is low ie. .3087.

Hide

```
checkresiduals(model2p$model$residuals)
```



The residuals randomness is still not good, and there are still many extreme lags in the ACF test suggesting serial correlation.

Hide

```
bgtest(model2p$model)
```

```

	Breusch-Godfrey test for serial correlation of order up to 1

data:  model2p$model
LM test = 525.11, df = 1, p-value < 2.2e-16
```

The test has a significant p value shows that the model is significant and fits the data well, despite the low R squared value.

We next use the Koyck fitting.

Hide

```
model3p = koyckDlm(x = as.vector(ppt) , y = as.vector(solar) , show.summary = TRUE)
```

```

Call:
ivreg(formula = y.t ~ Y.t_1 + X.t | Y.t_1 + X.t_1)

Residuals:
     Min       1Q   Median       3Q      Max 
-13.0926  -3.5961   0.3176   3.6103  14.8399 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -2.23925    0.76549  -2.925  0.00356 ** 
Y.t_1        0.98546    0.02424  40.650  < 2e-16 ***
X.t          5.34684    0.84383   6.336 4.37e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.814 on 656 degrees of freedom
Multiple R-Squared: 0.7598, Adjusted R-squared: 0.7591 
Wald test:  1104 on 2 and 656 DF,  p-value: < 2.2e-16 

                             alpha     beta       phi
Geometric coefficients:  -154.0203 5.346844 0.9854613
```

Hide

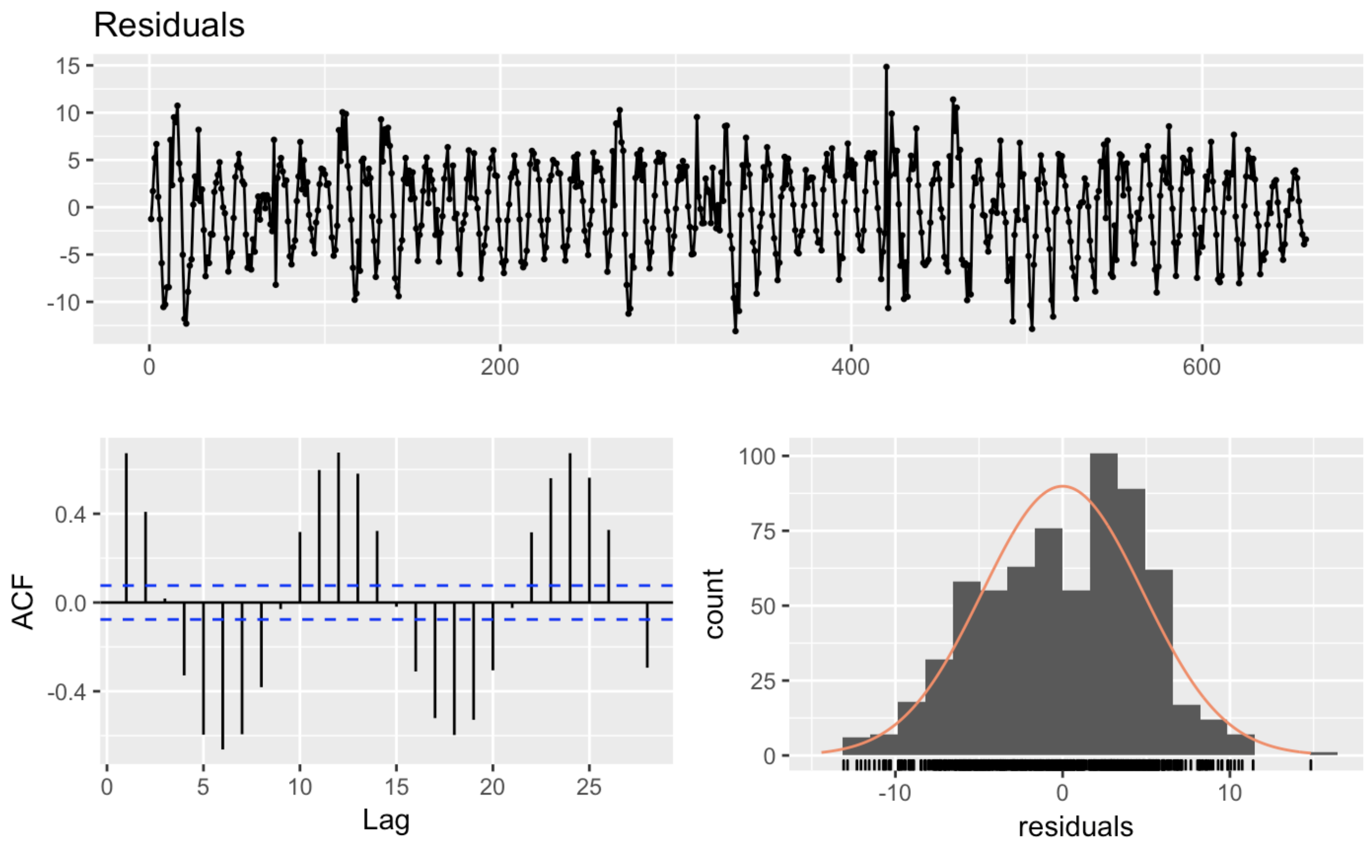
```
vif(model3p$model)
```

```
   Y.t_1      X.t 
1.605001 1.605001 
```

We see that the Koyck model shows a higher R squared value of .7591 and all lags and model are significant, hence it is a better fit than the previous models. We run diagnostic check.

Hide

```
checkresiduals(model3p$model$residuals)
```



The randomness is much better, but there are still some extreme values in the ACF plot showing serial correlation.

Hide

```
bgtest(model3p$model)
```

```

	Breusch-Godfrey test for serial correlation of order up to 1

data:  model3p$model
LM test = 387.66, df = 1, p-value < 2.2e-16
```

The model is statistically significant, since it has a very low p value.

We move on to the ardlm fitting.

Hide

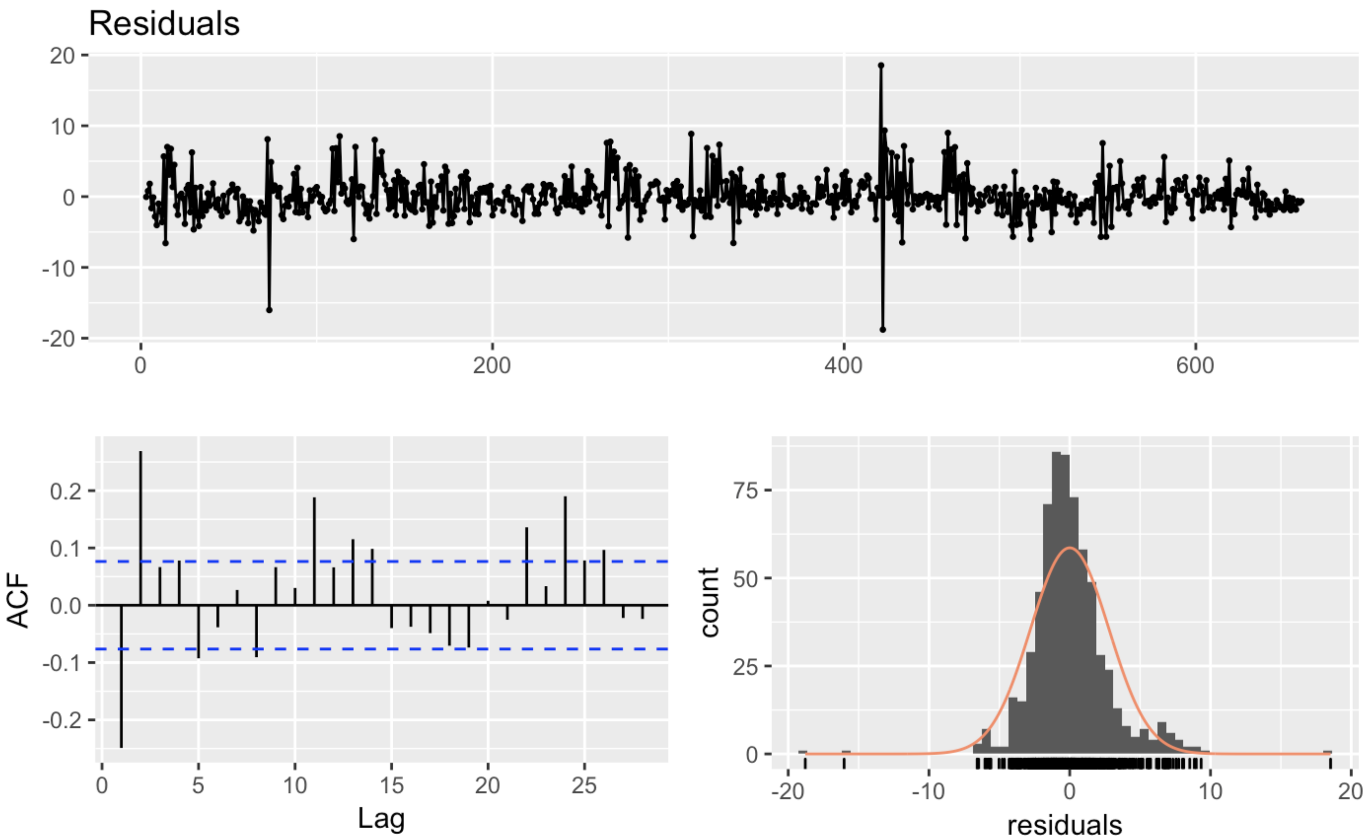
```
model4p = ardlDlm(x = as.vector(ppt) , y = as.vector(solar) , p = 2 , q = 2 , show.su
mmary = TRUE)
```

```
Time series regression with "ts" data:
Start = 3, End = 660

Call:
dynlm(formula = formula(model.text))

Residuals:
     Min       1Q   Median       3Q      Max
-18.7867  -1.5013  -0.2736   1.2345  18.5318

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.08758    0.39200   5.326 1.39e-07 ***
X.t         -0.96803    0.59464  -1.628 0.104022
L(X.t, 1)    0.70618    0.82880   0.852 0.394504
L(X.t, 2)    2.09832    0.59665   3.517 0.000467 ***
L(y.t, 1)    1.51119    0.02823  53.539  < 2e-16 ***
L(y.t, 2)   -0.67673    0.02840 -23.829  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.797 on 652 degrees of freedom
Multiple R-squared:  0.9192,    Adjusted R-squared:  0.9186
F-statistic:  1484 on 5 and 652 DF,  p-value: < 2.2e-16
```

The R squared value for the ardlm is coming to .9186 which shows a very good fit for the model. Intercept is significant and a lot of significant y,t lags. We move on to diagnostic testing.

Hide

```
checkresiduals(model4p$model$residuals)
```



We see that the randsomness is much better than the previous models and even less extreme lags showing low correlation and symmetric histogram.

Hide

```
bgtest(model4p$model)
```

```


    Breusch-Godfrey test for serial correlation of order up to 1

data:  model4p$model
LM test = 79.25, df = 1, p-value < 2.2e-16
```

With a p value < 2.2e-16 the model is statistically significant and the best DLM model fit so far.

Hide

```
model4p.forecasts = ardlDlmForecast(model = model4p, x = datax.ts, h = 24)$forecasts
plot(solar, type="o", xlim = c(1959, 2019),  ylab = "Solar Radiation", xlab = "Year",

     main="Fig 3. Solar Radiation Forecasts 2 years")
lines(ts(model4p.forecasts[1:24],start = 2015),col="Purple",type="o")
```

Hide

```
legend("topleft",lty=1, pch = 1, text.width = 20, col=c("black","purple"),
       c("Solar Radiation series","ARDL"))
```

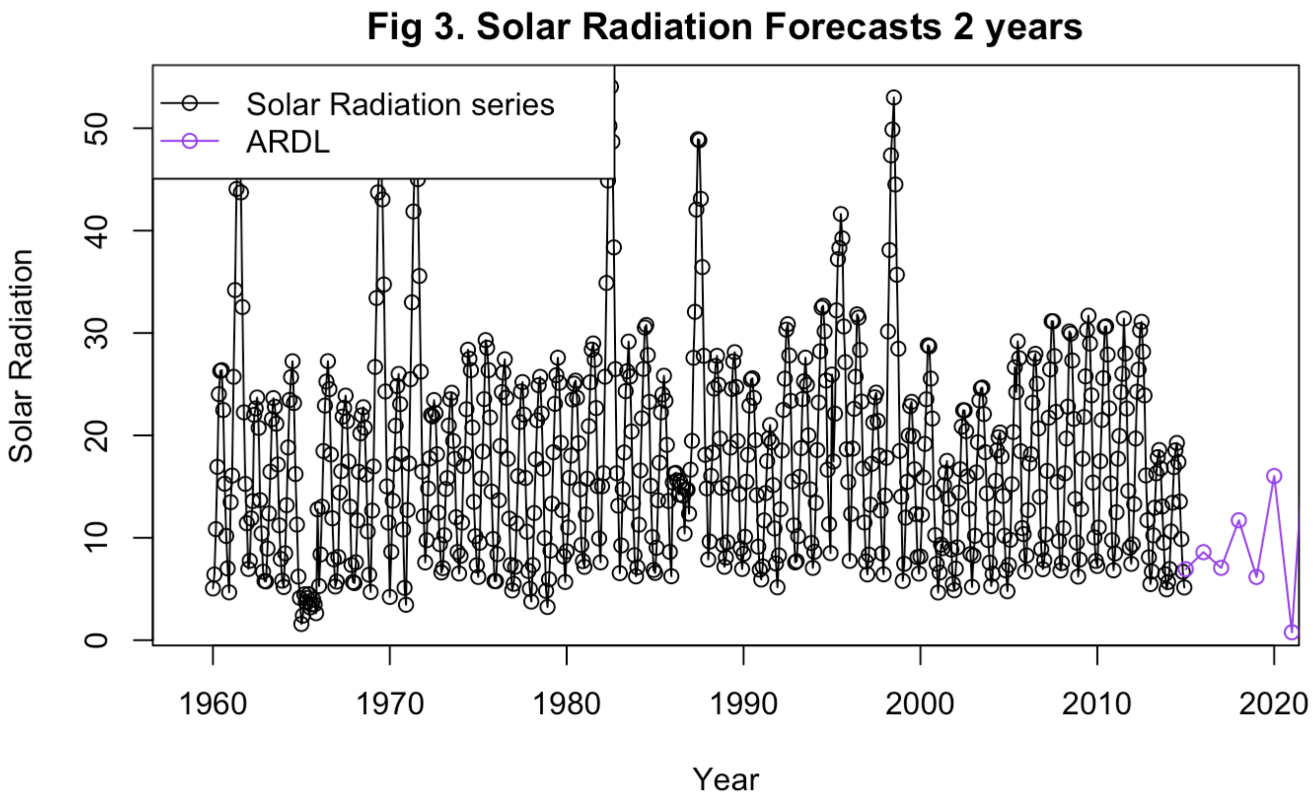


Figure 3 shows the ardl model forecast along with the original series.

2. DYNLM Fitting

For dynlm fitting we first start off with SES fitting.

Hide

```
fit1.ses = ses(solar, initial="simple", h=24)
summary(fit1.ses)
```

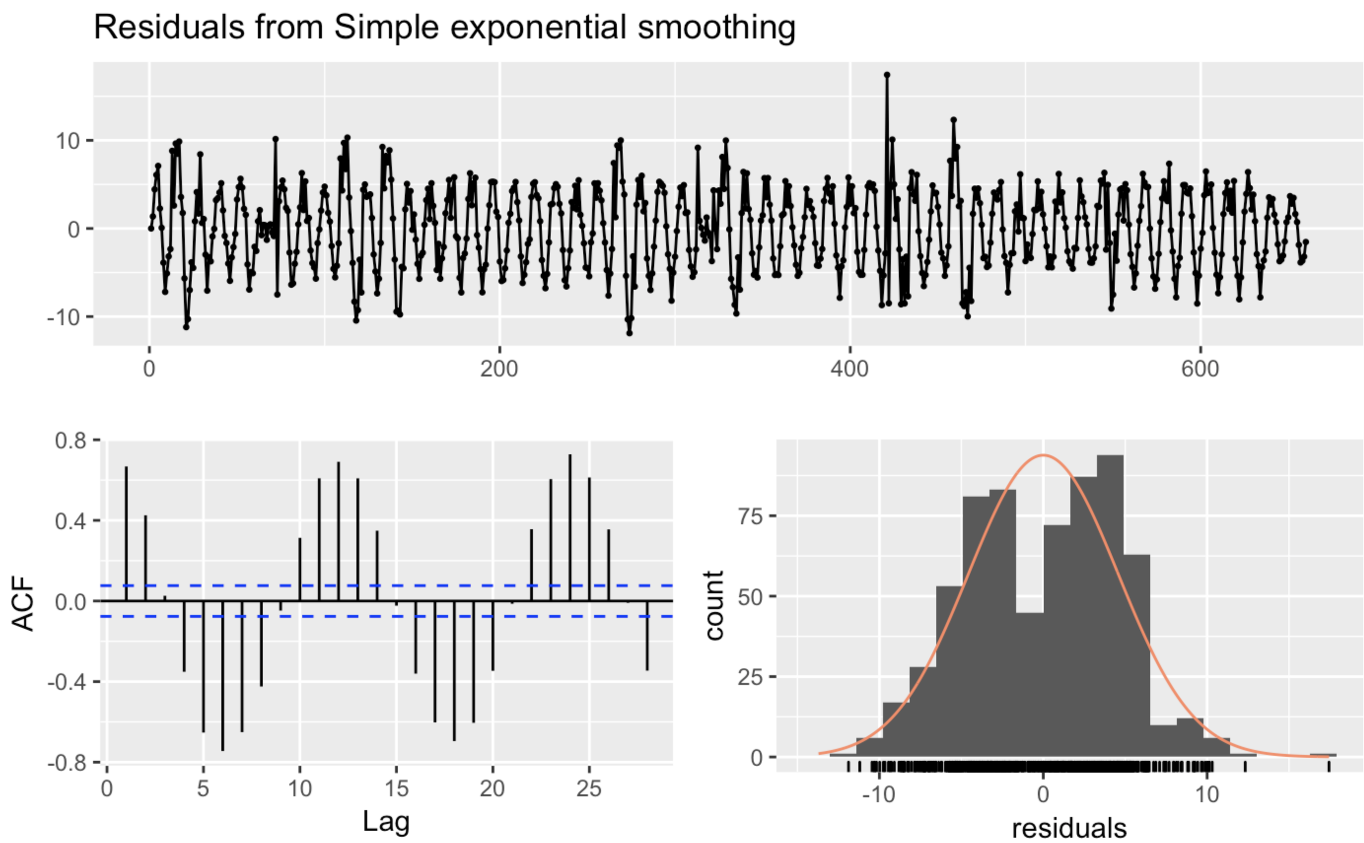
```
Forecast method: Simple exponential smoothing

Model Information:
Simple exponential smoothing

Call:
 ses(y = solar, h = 24, initial = "simple")

  Smoothing parameters:
    alpha = 1

  Initial states:
    l = 5.0517

  sigma:  4.5688
Error measures:
                       ME     RMSE      MAE       MPE     MAPE     MASE      ACF1
Training set 0.0001462894 4.568777 3.876091 -5.211851 27.29823 0.636771 0.6677846

Forecasts:
         Point Forecast       Lo 80    Hi 80      Lo 95    Hi 95
Jan 2015        5.14828  -0.7068426 11.00340  -3.806357 14.10292
Feb 2015        5.14828  -3.1321139 13.42867  -7.515490 17.81205
Mar 2015        5.14828  -4.9930900 15.28965 -10.361607 20.65817
Apr 2015        5.14828  -6.5619655 16.85853 -12.760995 23.05756
May 2015        5.14828  -7.9441725 18.24073 -14.874898 25.17146
Jun 2015        5.14828  -9.1937832 19.49034 -16.786013 27.08257
Jul 2015        5.14828 -10.3429188 20.63948 -18.543464 28.84002
Aug 2015        5.14828 -11.4125081 21.70907 -20.179260 30.47582
Sep 2015        5.14828 -12.4170884 22.71365 -21.715633 32.01219
Oct 2015        5.14828 -13.3672440 23.66380 -23.168771 33.46533
Nov 2015        5.14828 -14.2709655 24.56753 -24.550893 34.84745
Dec 2015        5.14828 -15.1344604 25.43102 -25.871495 36.16806
Jan 2016        5.14828 -15.9626655 26.25923 -27.138125 37.43469
Feb 2016        5.14828 -16.7595836 27.05614 -28.356906 38.65347
Mar 2016        5.14828 -17.5285132 27.82507 -29.532883 39.82944
Apr 2016        5.14828 -18.2722113 28.56877 -30.670271 40.96683
May 2016        5.14828 -18.9930099 29.28957 -31.772637 42.06920
Jun 2016        5.14828 -19.6929024 29.98946 -32.843030 43.13959
Jul 2016        5.14828 -20.3736087 30.67017 -33.884081 44.18064
Aug 2016        5.14828 -21.0366254 31.33319 -34.898077 45.19464
Sep 2016        5.14828 -21.6832636 31.97982 -35.887025 46.18359
Oct 2016        5.14828 -22.3146804 32.61124 -36.852694 47.14925
Nov 2016        5.14828 -22.9319027 33.22846 -37.796654 48.09321
Dec 2016        5.14828 -23.5358467 33.83241 -38.720306 49.01687
```

With a simple ses fitting we see that the MASE value is .636 which is still high. We do diagnostic check.

Hide

```
checkresiduals(fit1.ses)
```

```
    Ljung-Box test

data:  Residuals from Simple exponential smoothing
Q* = 1625.4, df = 8, p-value < 2.2e-16

Model df: 2.   Total lags used: 10
```



The randomness in the series is low, and extreme lags show existence of serial correlation.

We next try the holt model.

Hide

```
fit2.holt = holt(solar, initial="simple", h=24)
summary(fit2.holt)
```

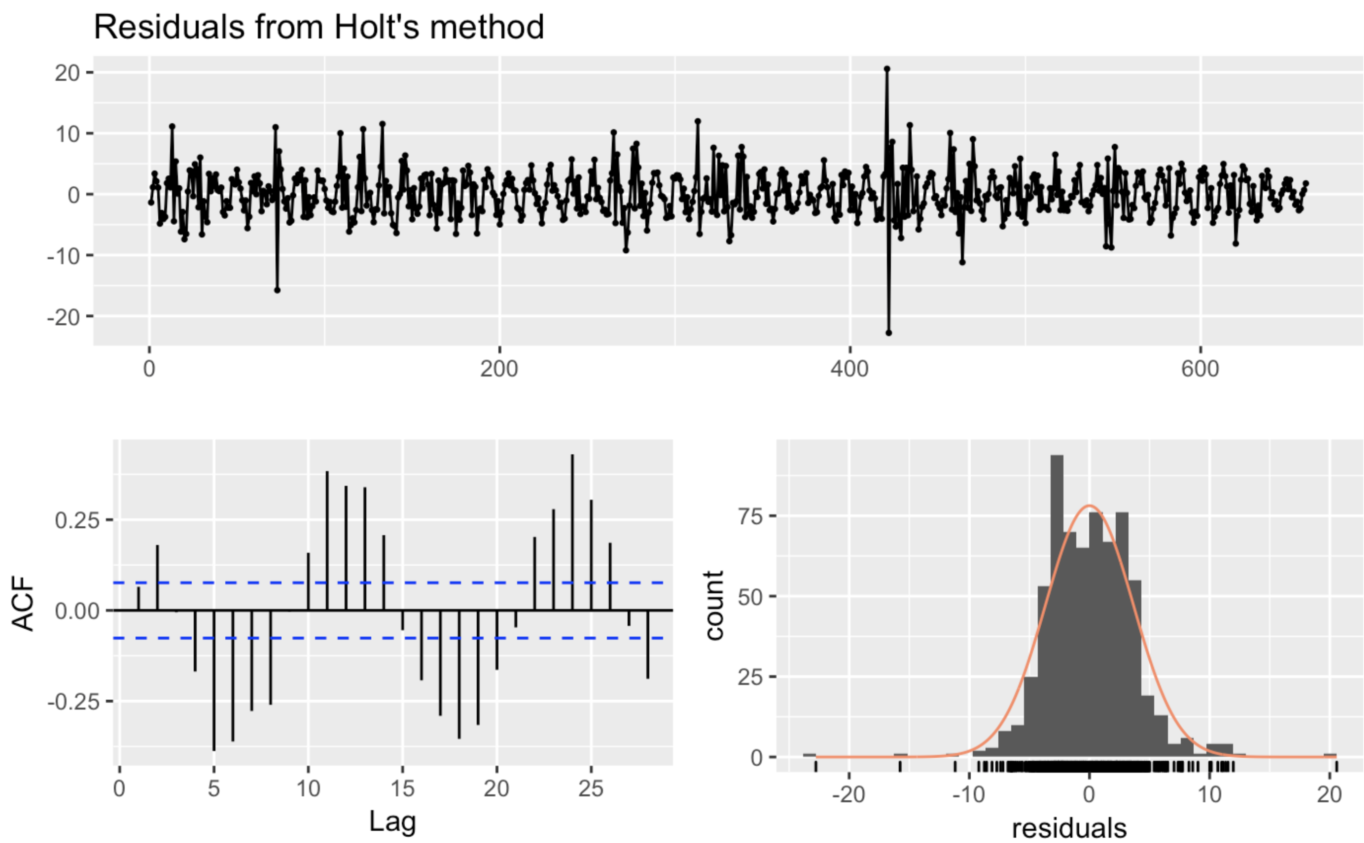
```
Forecast method: Holt's method

Model Information:
Holt's method

Call:
 holt(y = solar, h = 24, initial = "simple")

  Smoothing parameters:
    alpha = 0.9165
    beta  = 1

  Initial states:
    l = 5.0517
    b = 1.3641

  sigma:  3.6493
Error measures:
                       ME     RMSE      MAE      MPE     MAPE     MASE       ACF1
Training set -0.004941409 3.649286 2.806374 6.556496 21.13578 0.461036 0.06518485

Forecasts:
         Point Forecast       Lo 80      Hi 80      Lo 95     Hi 95
Jan 2015      3.3755387   -1.301209   8.052287   -3.77693  10.52801
Feb 2015      1.7507190   -8.358924  11.860362  -13.71065  17.21208
Mar 2015      0.1258992  -16.851847  17.103645  -25.83932  26.09112
Apr 2015     -1.4989205  -26.473565  23.475724  -39.69434  36.69650
May 2015     -3.1237402  -37.070990  30.823510  -55.04158  48.79410
Jun 2015     -4.7485600  -48.543955  39.046835  -71.72784  62.23072
Jul 2015     -6.3733797  -60.819126  48.072367  -89.64096  76.89420
Aug 2015     -7.9981994  -73.839431  57.843032 -108.69367  92.69727
Sep 2015     -9.6230191  -87.558671  68.312633 -128.81531 109.56928
Oct 2015    -11.2478389 -101.938399  79.442721 -149.94708 127.45140
Nov 2015    -12.8726586 -116.945936  91.200619 -172.03900 146.29368
Dec 2015    -14.4974783 -132.553050 103.558094 -195.04790 166.05294
Jan 2016    -16.1222981 -148.735022 116.490426 -218.93596 186.69136
Feb 2016    -17.7471178 -165.469968 129.975732 -243.66972 208.17549
Mar 2016    -19.3719375 -182.738335 143.994459 -269.21928 230.47541
Apr 2016    -20.9967573 -200.522511 158.528996 -295.55770 253.56419
May 2016    -22.6215770 -218.806525 173.563371 -322.66056 277.41741
Jun 2016    -24.2463967 -237.575806 189.083013 -350.50557 302.01278
Jul 2016    -25.8712164 -256.816987 205.074554 -379.07229 327.32986
Aug 2016    -27.4960362 -276.517751 221.525678 -408.34188 353.34981
Sep 2016    -29.1208559 -296.666696 238.424984 -438.29691 380.05520
Oct 2016    -30.7456756 -317.253232 255.761880 -468.92117 407.42982
Nov 2016    -32.3704954 -338.267484 273.526493 -500.19957 435.45858
Dec 2016    -33.9953151 -359.700219 291.709589 -532.11798 464.12735
```

The Holt model gives us a MASE value of .461036, which is lower than the SES model. We move on to diagnostic check.

Hide

```
checkresiduals(fit2.holt)
```

```
    Ljung-Box test

data:  Residuals from Holt's method
Q* = 344.48, df = 6, p-value < 2.2e-16

Model df: 4.   Total lags used: 10
```



The randomness in the data is much better, however there is still existence of serial correlation.

Hide

```
fit3.hw = hw(solar,seasonal="additive", h=2*frequency(solar))
summary(fit3.hw)
```

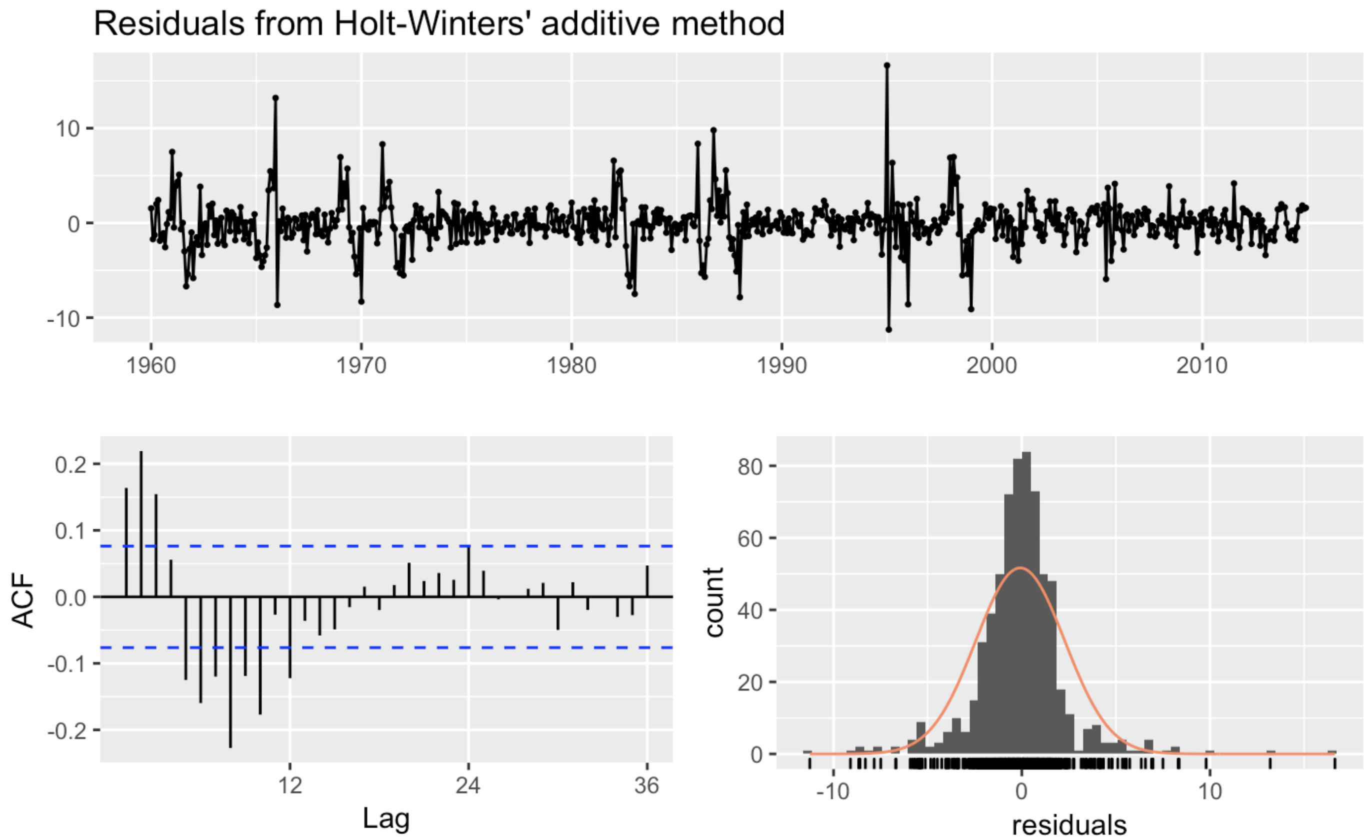
```
Forecast method: Holt-Winters' additive method

Model Information:
Holt-Winters' additive method

Call:
 hw(y = solar, h = 2 * frequency(solar), seasonal = "additive")

  Smoothing parameters:
    alpha = 0.9968
    beta  = 0.0079
    gamma = 0.0027

  Initial states:
    l = 12.813
    b = 0.4276
    s=-10.6349 -7.3748 -2.6593 2.7233 7.775 11.0058
           9.8199 6.1144 1.8544 -1.8065 -7.0856 -9.7316

  sigma:  2.3699

     AIC     AICc      BIC
5457.817 5458.770 5534.185

Error measures:
                      ME     RMSE      MAE       MPE     MAPE      MASE     ACF1
Training set -0.08375221 2.369864 1.547273 -1.615444 12.99165 0.2541887 0.163735

Forecasts:
         Point Forecast      Lo 80     Hi 80       Lo 95    Hi 95
Jan 2015       5.899303   2.862201  8.936406   1.2544557 10.54415
Feb 2015       8.536199   4.230998 12.841399   1.9519623 15.12043
Mar 2015      13.828280   8.537485 19.119075   5.7367070 21.91985
Apr 2015      17.502130  11.370374 23.633886   8.1244191 26.87984
May 2015      21.822830  14.941431 28.704228  11.2986391 32.34702
Jun 2015      25.314433  17.747444 32.881421  13.7417227 36.88714
Jul 2015      26.552786  18.348117 34.757454  14.0048274 39.10074
Aug 2015      23.394989  14.590069 32.199910   9.9290252 36.86095
Sep 2015      18.270816   8.895819 27.645813   3.9329954 32.60864
Oct 2015      12.811417   2.891257 22.731577  -2.3601587 27.98299
Nov 2015       8.147760  -2.296607 18.592126  -7.8255203 24.12104
Dec 2015       5.037795  -5.912884 15.988474 -11.7098227 21.78541
Jan 2016       5.789632  -5.654269 17.233532 -11.7123037 23.29157
Feb 2016       8.426527  -3.494615 20.347668  -9.8052858 26.65834
Mar 2016      13.718608   1.332116 26.105100  -5.2248969 32.66211
Apr 2016      17.392458   4.551168 30.233748  -2.2466003 37.03152
May 2016      21.713158   8.426495 34.999820   1.3929612 42.03335
Jun 2016      25.204761  11.481193 38.928329   4.2163742 46.19315
Jul 2016      26.443114  12.290280 40.595947   4.7982231 48.08800
Aug 2016      23.285317   8.710146 37.860488   0.9945162 45.57612
Sep 2016      18.161144   3.169938 33.152351  -4.7659279 41.08822
Oct 2016      12.701745  -2.699742 28.103232 -10.8527973 36.25629
Nov 2016       8.038088  -7.768410 23.844585 -16.1358645 32.21204
Dec 2016       4.928123 -11.278545 21.134792 -19.8578375 29.71408
```

The Holt-Winter's additive method gives us a MASE value of .2541887, which is the lowest we have seen amongst the dynlm models, hence we move on to diagnostic check

Hide

```
checkresiduals(fit3.hw)
```

```

	Ljung-Box test

data:  Residuals from Holt-Winters' additive method
Q* = 193.75, df = 8, p-value < 2.2e-16

Model df: 16.   Total lags used: 24
```



The randomness for the series is very good. Histogram is symmetric. Few extreme lags but much better than previous models, also the model is significant and has the lowest MASE calue of .254, hence we will use this for the forecast.

Hide

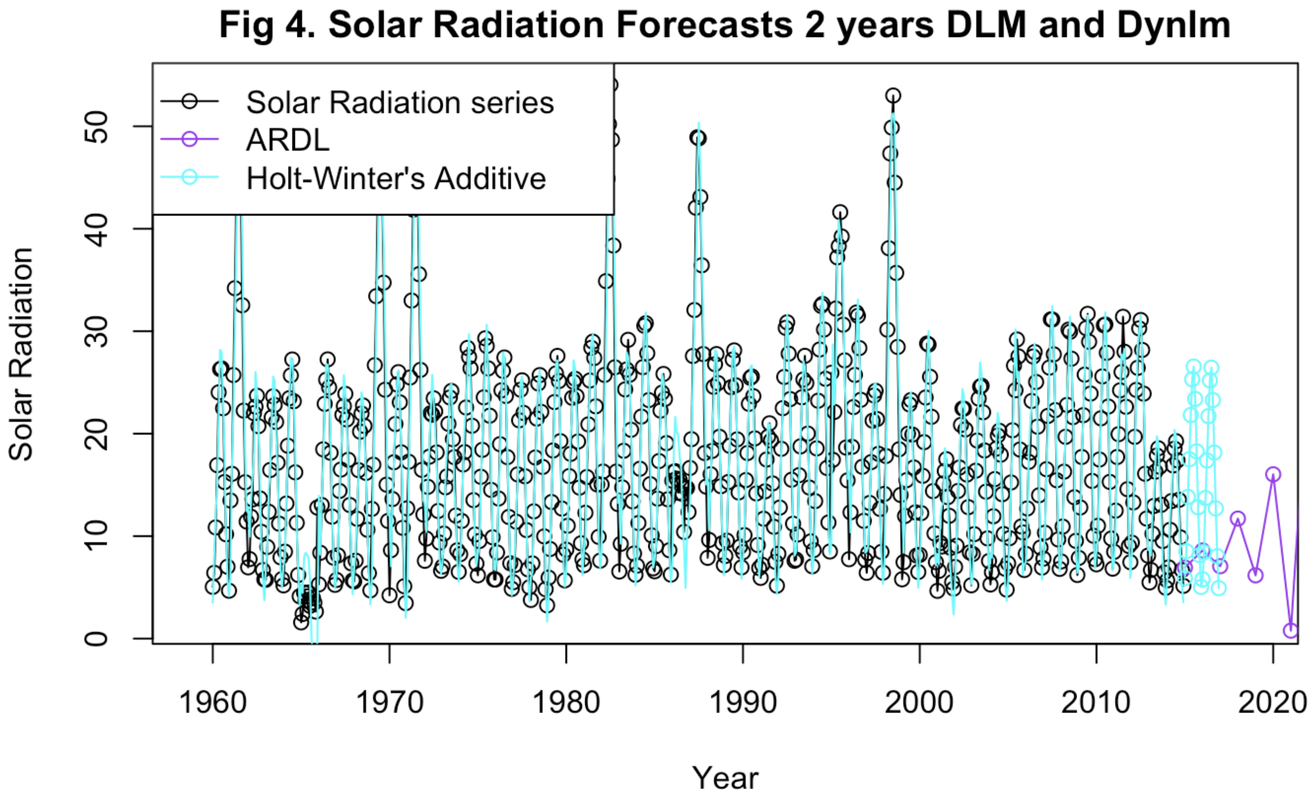
```
model4p.forecasts = ardlDlmForecast(model = model4p, x = datax.ts, h = 24)$forecasts
plot(solar, type="o", xlim = c(1959, 2019),  ylab = "Solar Radiation", xlab = "Year",

     main="Fig 4. Solar Radiation Forecasts 2 years DLM and Dynlm")

lines(ts(model4p.forecasts[1:24],start = 2015),col="Purple",type="o")
```

Hide

```
lines(fit3.hw$mean, type="o", col="cyan")
lines(fitted(fit3.hw), col="cyan")
```

Hide

```
legend("topleft",lty=1, pch = 1, text.width = 20, col=c("black","purple","cyan"),
       c("Solar Radiation series","ARDL","Holt-Winter's Additive"))
```



3. ETS Model fitting.

We beging ETS model fitting by using an ANN model.

Hide

```
fit1.etsA = ets(solar, model="ANN")
summary(fit1.etsA)
```

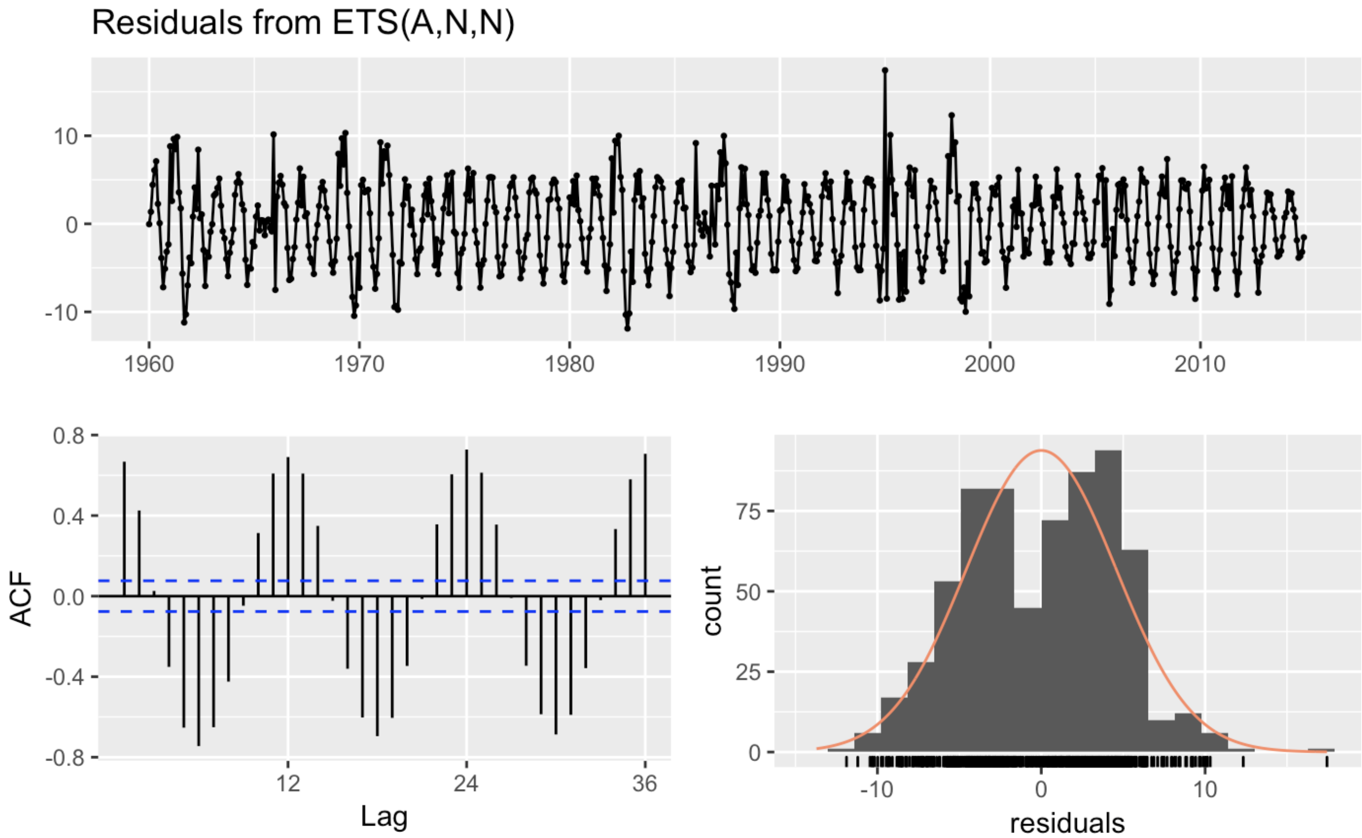
```
ETS(A,N,N)

Call:
 ets(y = solar, model = "ANN")

  Smoothing parameters:
    alpha = 0.9999

  Initial states:
    l = 5.0921

  sigma:  4.5691

     AIC     AICc      BIC
6296.371 6296.407 6309.847

Training set error measures:
                     ME     RMSE      MAE       MPE     MAPE      MASE      ACF1
Training set 8.54059e-05 4.569082 3.876443 -5.214167 27.30156 0.6368289 0.6678339
```

From the model we see that the MASE is very high at .6368, we continur to diagnostic check.

Hide

```
checkresiduals(fit1.etsA)
```

```

    Ljung-Box test

data:  Residuals from ETS(A,N,N)
Q* = 4227.6, df = 22, p-value < 2.2e-16

Model df: 2.   Total lags used: 24
```



Even though the Ljung-Bos test is significant, there is low randomness in the residuals and extreme lags which show signs of correlation.

We next try MNN model.

Hide

```
fit2.etsM = ets(solar, model="MNN")
summary(fit2.etsM)
```

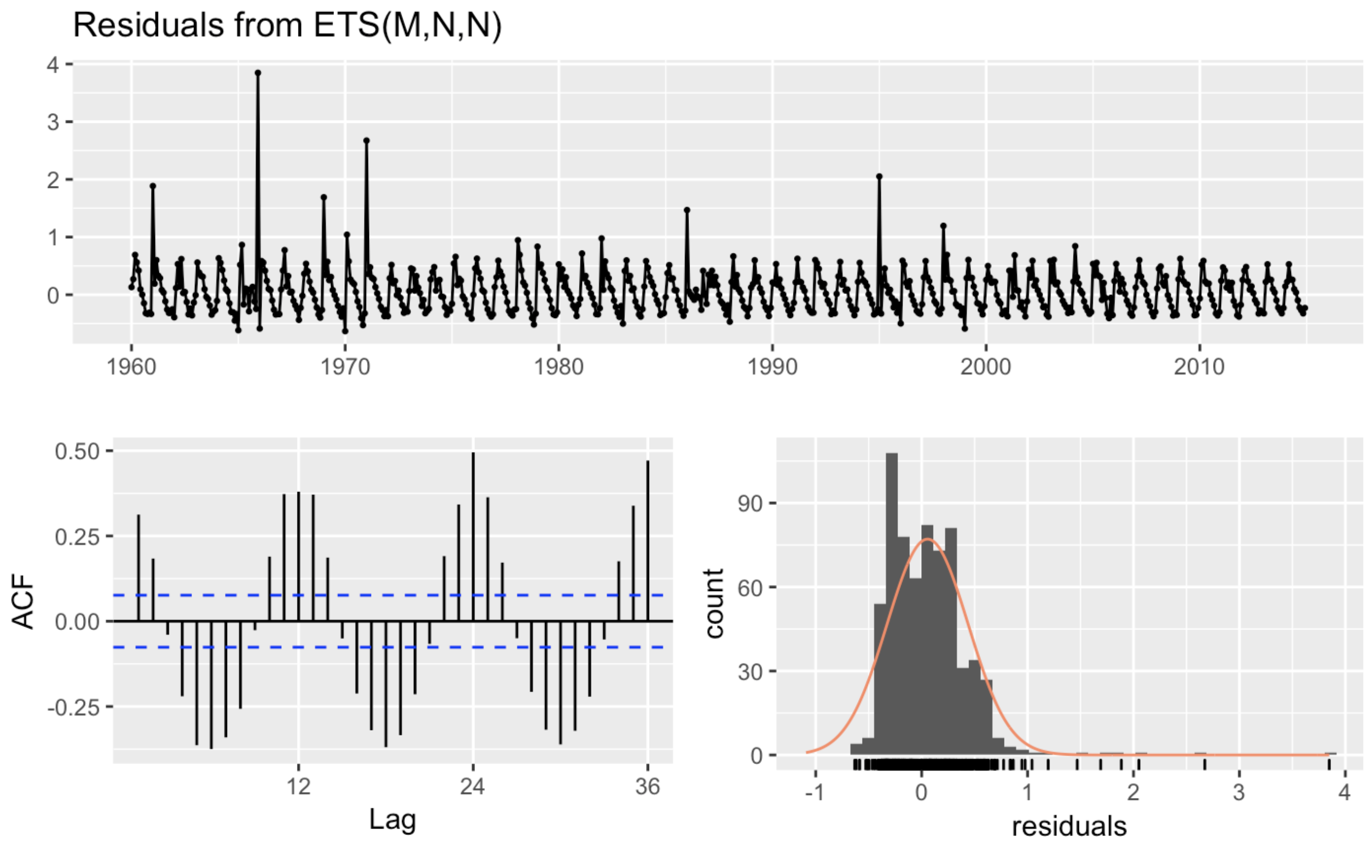
```
ETS(M,N,N)

Call:
 ets(y = solar, model = "MNN")

  Smoothing parameters:
    alpha = 0.9999

  Initial states:
    l = 4.4673

  sigma:  0.3862

     AIC     AICc      BIC
6619.776 6619.812 6633.252

Training set error measures:
                      ME     RMSE      MAE       MPE     MAPE      MASE      ACF1
Training set 0.001032069 4.569139 3.877268 -5.195429 27.31788 0.6369644 0.6678793
```

The MASE for the MNN model is also high at .66787 and it has increased from additive type.

Hide

```
checkresiduals(fit2.etsM)
```

```


    Ljung-Box test

data:  Residuals from ETS(M,N,N)
Q* = 1334.8, df = 22, p-value < 2.2e-16

Model df: 2.   Total lags used: 24
```



The series doesnt show randomness and the ACF extreme lags show serial correlation. Even though the Ljung-Box test is significant, this model is not very good.

We next try using AAN model

Hide

```
fit3.etsA = ets(solar, model="AAN")
summary(fit3.etsA)
```

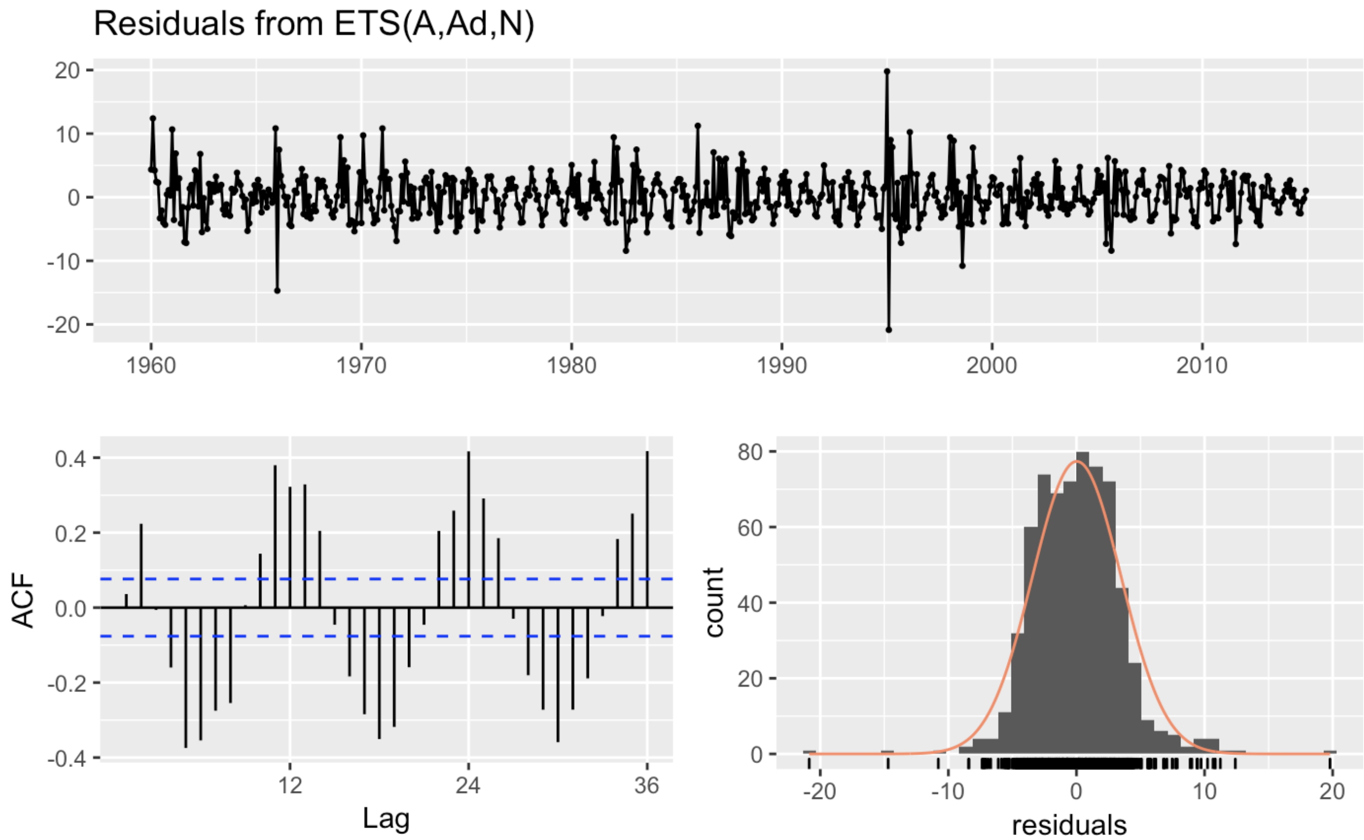
```
ETS(A,Ad,N)

Call:
 ets(y = solar, model = "AAN")

  Smoothing parameters:
    alpha = 0.9539
    beta  = 0.9539
    phi   = 0.8

  Initial states:
    l = 18.3958
    b = -22.1136

  sigma:  3.4572

     AIC     AICc      BIC
5934.274 5934.403 5961.228

Training set error measures:
                     ME     RMSE      MAE      MPE     MAPE      MASE       ACF1
Training set 0.02185466 3.457174 2.622583 4.237987 19.09486 0.4308425 0.03614568
```

The MASE for the AAN model is now .4308, which is lower than the previous two models. We move to diagnostic testing.

Hide

```
checkresiduals(fit3.etsA)
```

```

    Ljung-Box test

data:  Residuals from ETS(A,Ad,N)
Q* = 1049.9, df = 19, p-value < 2.2e-16

Model df: 5.   Total lags used: 24
```



The ACF plots still shows signs of seasonality, and the randomness is not very evident in the series. Hence, we move on to the next model.

We use the MAA model.

Hide

```
fit4.etsA = ets(solar, model="MAA")
summary(fit4.etsA)
```

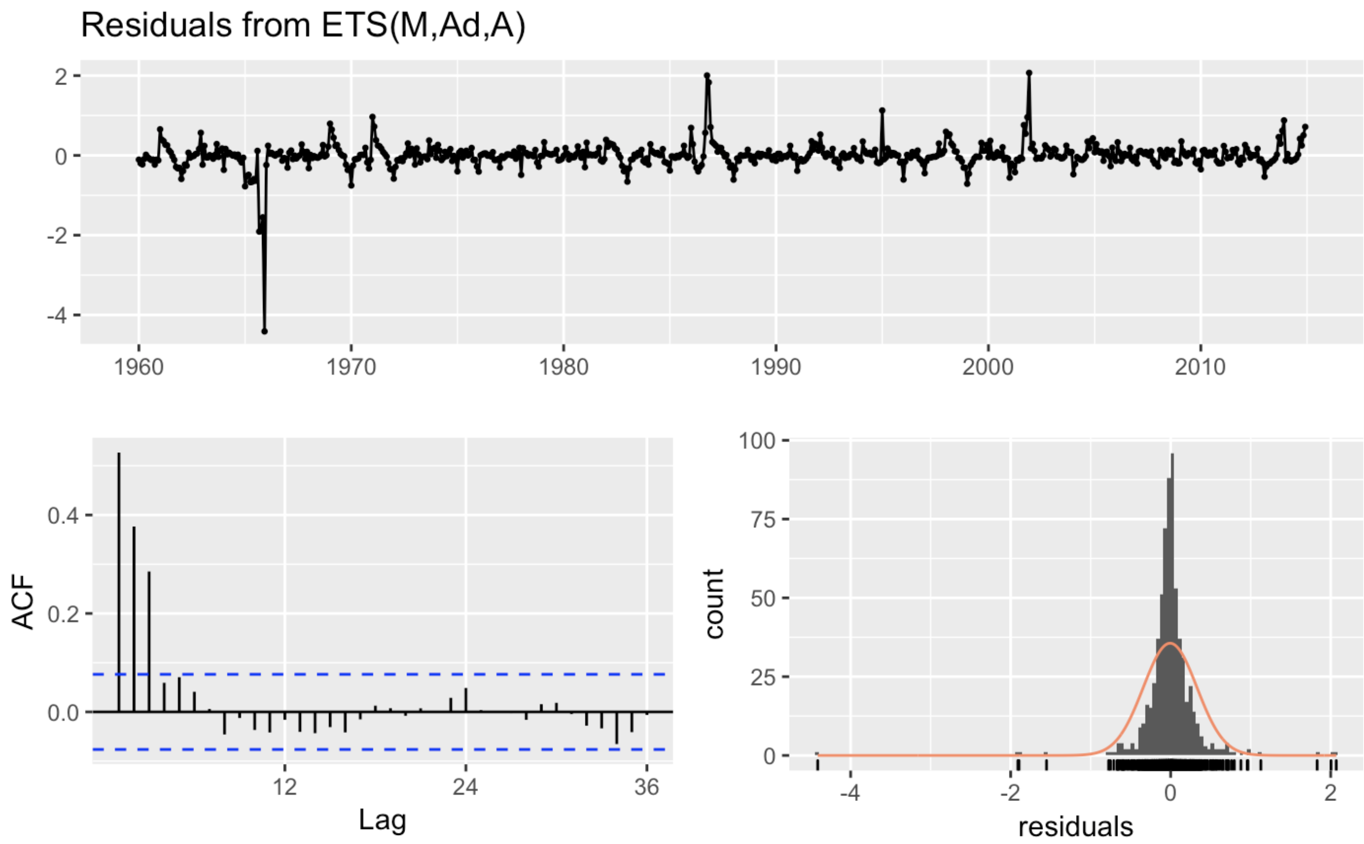
```
ETS(M,Ad,A)

Call:
 ets(y = solar, model = "MAA")

  Smoothing parameters:
    alpha = 0.478
    beta  = 8e-04
    gamma = 1e-04
    phi   = 0.8495

  Initial states:
    l = 10.7367
    b = 2.9076
    s=-10.3436 -7.8261 -3.4126 0.1089 7.7705 10.7246
           9.8295 7.1223 2.5865 -2.0162 -6.9922 -7.5514

  sigma:  0.335

     AIC     AICc      BIC
6492.852 6493.919 6573.712

Training set error measures:
                      ME     RMSE      MAE       MPE     MAPE      MASE      ACF1
Training set -0.04002889 3.335056 2.289546 -5.087836 19.54459 0.3761306 0.6061329
```

We see that the MASE for the MAA models has decreased from the previous models to .376, and we move on to diagnostic checking.

Hide

```
checkresiduals(fit4.etsA)
```

```


    Ljung-Box test

data:  Residuals from ETS(M,Ad,A)
Q* = 349.61, df = 7, p-value < 2.2e-16

Model df: 17.   Total lags used: 24
```



The variance and randomness of the series looks good, signs of seasonality in the ACF is gone, and Histogram looks symmetric. The MAA model seems to be catching the serial correlation structure in the series. Hence we will go ahead and forecast using this.

Hide

```
frc.MAA = forecast(fit4.etsA , h = 2* frequency(solar))
plot(solar, type="o", xlim = c(1959, 2019),  ylab = "Solar Radiation", xlab = "Year",

     main="Fig 5. Solar Radiation Forecasts 2 years DLM,Dynlm and ETS")

lines(ts(model4p.forecasts[1:24],start = 2015),col="Purple",type="o")
```

Hide

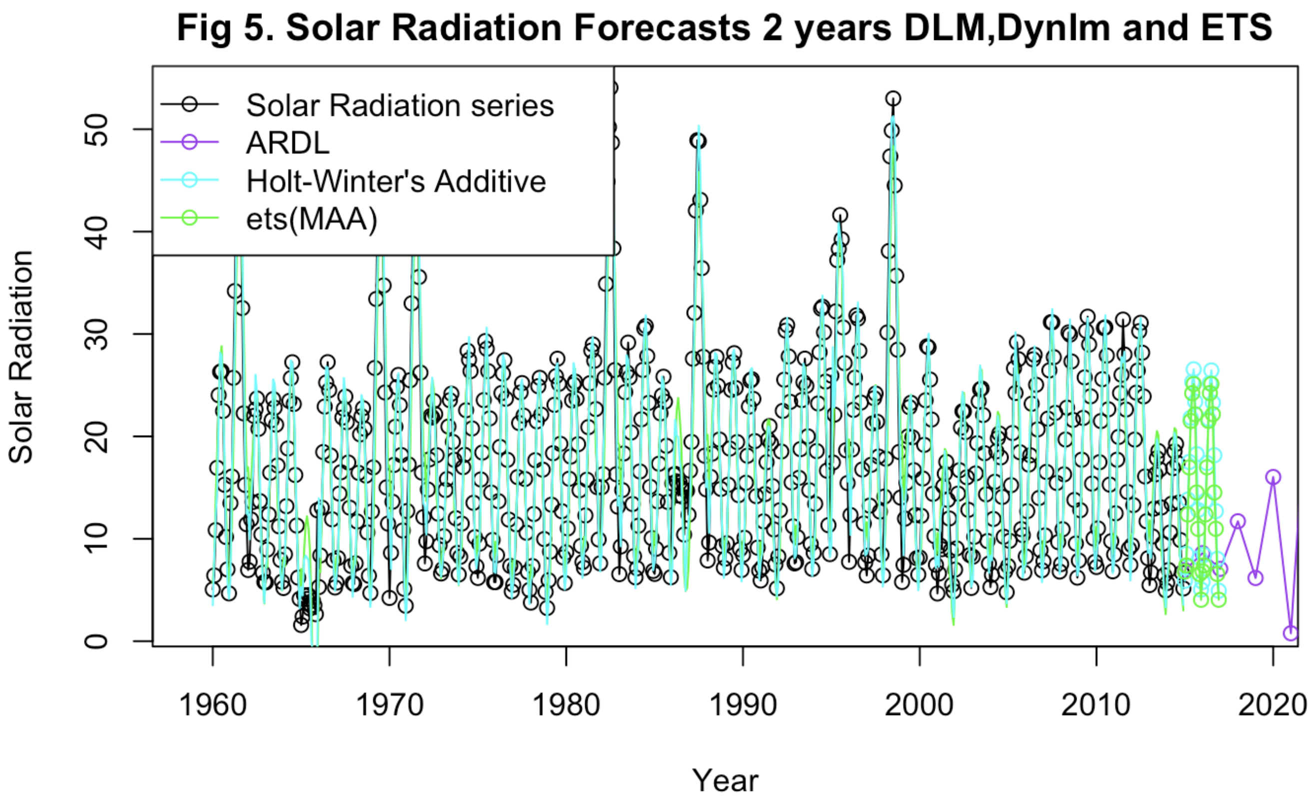
```
lines(fit3.hw$mean, type="o", col="cyan")
lines(fitted(fit4.etsA), col="green", lty=1)
```

Hide

```
lines(fitted(fit3.hw), col="cyan")
lines(frc.MAA$mean,col="green", type="o")
```

Hide

```
legend("topleft",lty=1, pch = 1, text.width = 20, col=c("black","purple","cyan","gree
n"),
       c("Solar Radiation series","ARDL","Holt-Winter's Additive", "ets(MAA)"))
```



In Fig 5. we can see the forecasts for DLM, DYNlm and ets Models that were chosen to be the best fit. Discussion on findings for task 1, is noted below in the conslusions section.

2. Task 2

Reading in the data and preparing for analysis

Hide

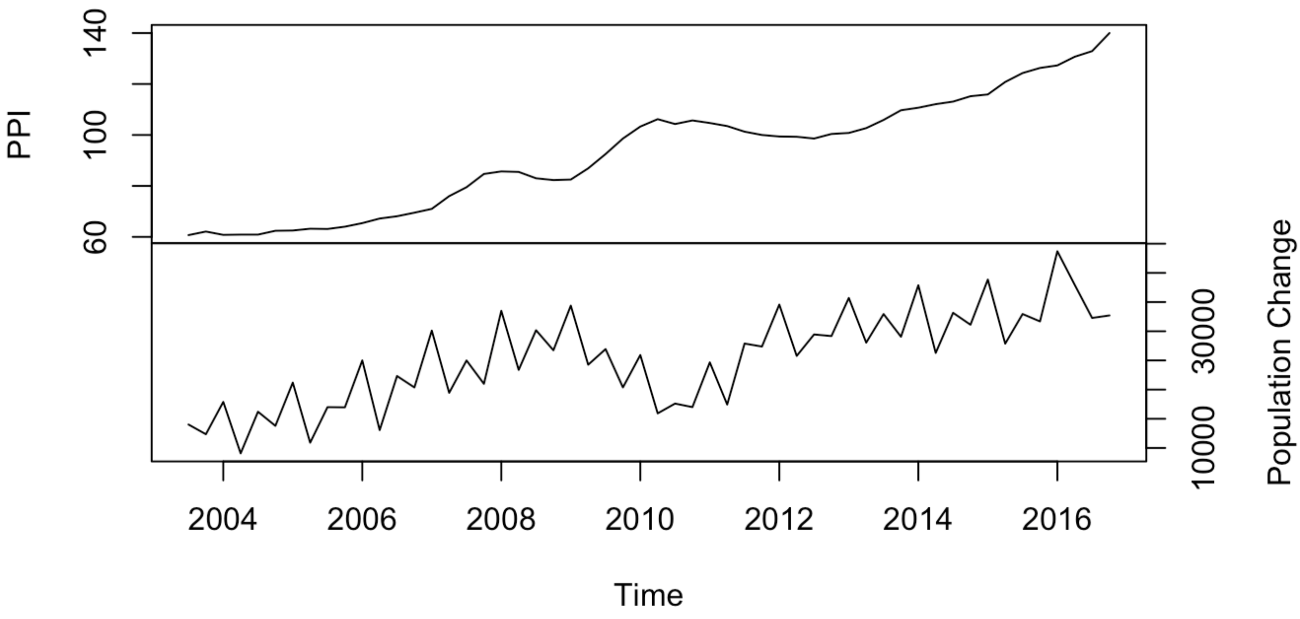
```
data2 <- read_csv("~/Desktop/Forecasting - Ass 2/data2.csv")
PPI = ts(data2$price, start = c(2003,3), frequency = 4)
change = ts(data2$change, start = c(2003,3), frequency = 4)
data2 = ts(data2[,2:3], start = c(2003,3), frequency = 4)
```

Plotting the two series.

Hide

```
data2.joint=ts.intersect(PPI,change)
colnames(data2.joint) =  c("PPI","Population Change")
plot(data2.joint , yax.flip=T, main = "Fig6. Timeseries plot of PPI and Population Ch
ange from Q3 2003 - Q4 2016")
```

Fig6. Timeseries plot of PPI and Population Change from Q3 2003 - Q4 2016

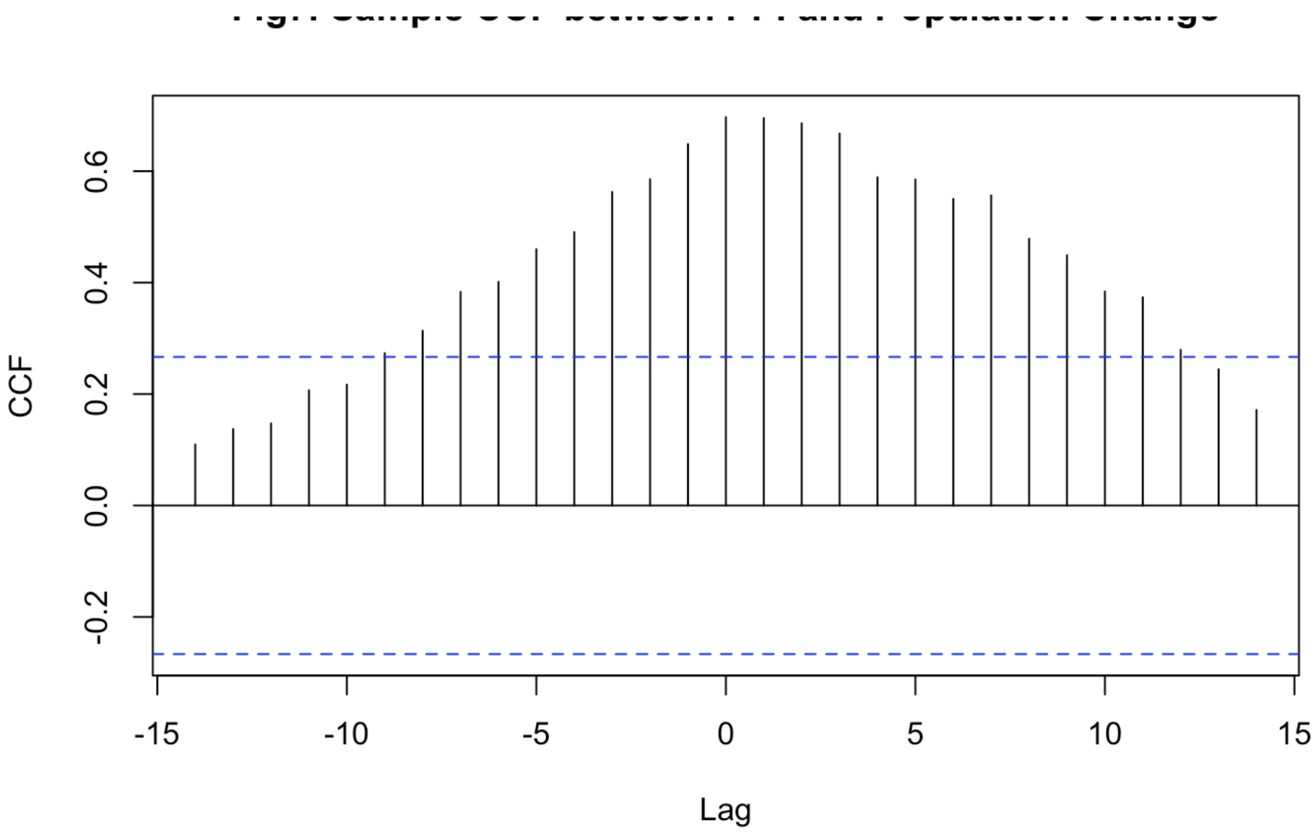


From Fig6. we can see in the two series, it seems that there might be a correlation between them, as PPI increases so does population change, as according to the visual trend in the two plots.

We will know display the CCF function to take another look at the correlation structure between the two series.

Hide

```
ccf(as.vector(data2.joint[,1]), as.vector(data2.joint[,2]),ylab='CCF', main = "Fig7.
 Sample CCF between PPI and Population Change")
```

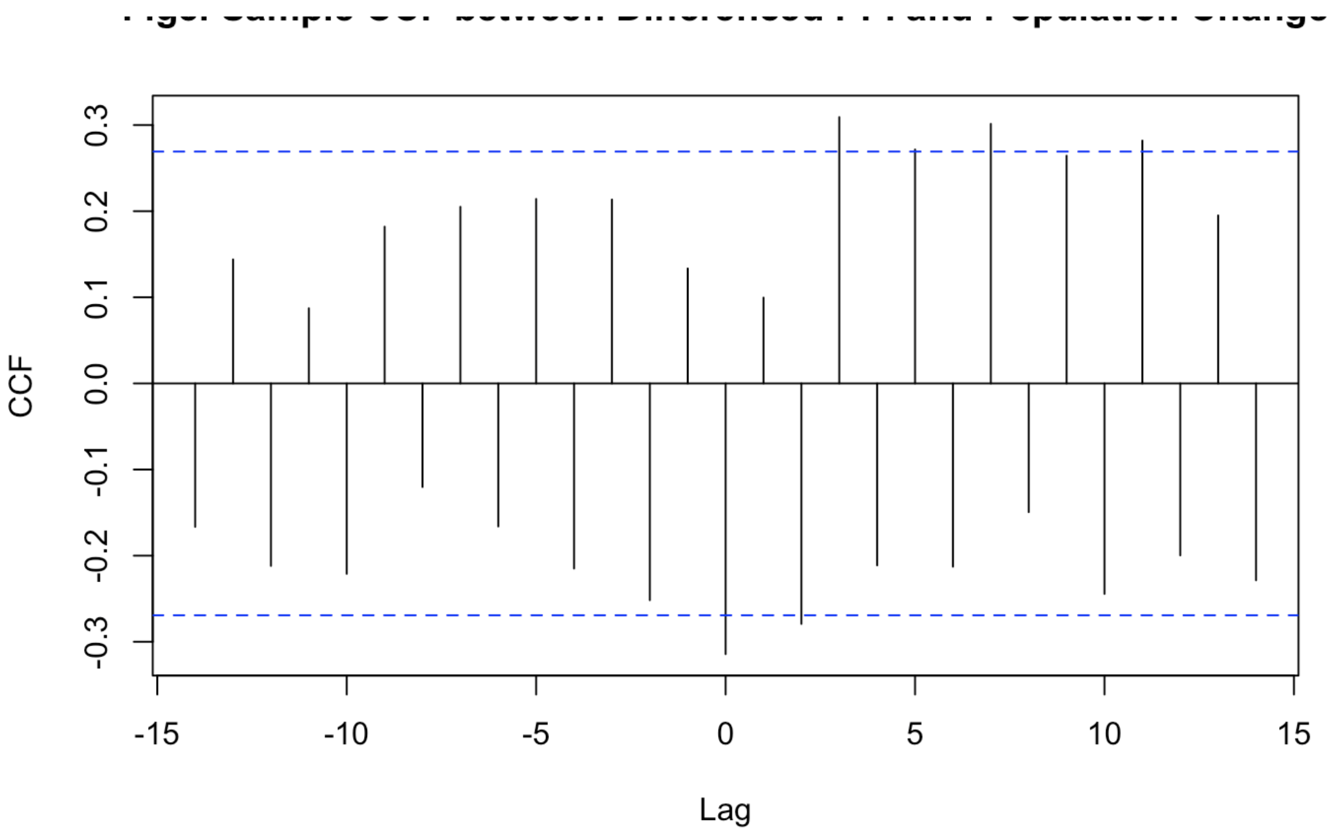


There appears to be a high correlation structure in the CCF plot, and a lot of cross -correlations are significantly different from zero.

We will now take the CCF of the differenced series to see if it has any variation to the previous CCF Plot.

Hide

```
ccf(as.vector(diff(data2.joint[,1])), as.vector(diff(data2.joint[,2])),ylab='CCF', main = "Fig8. Sample CCF between Differenced PPI and Population Change")
```



There appear to be some significant correlation in the CCF between the differenced time-series. The number of significant lags have reduced significantly.

This is not enough to say for certain, that there is no spurious correlation between the two series.

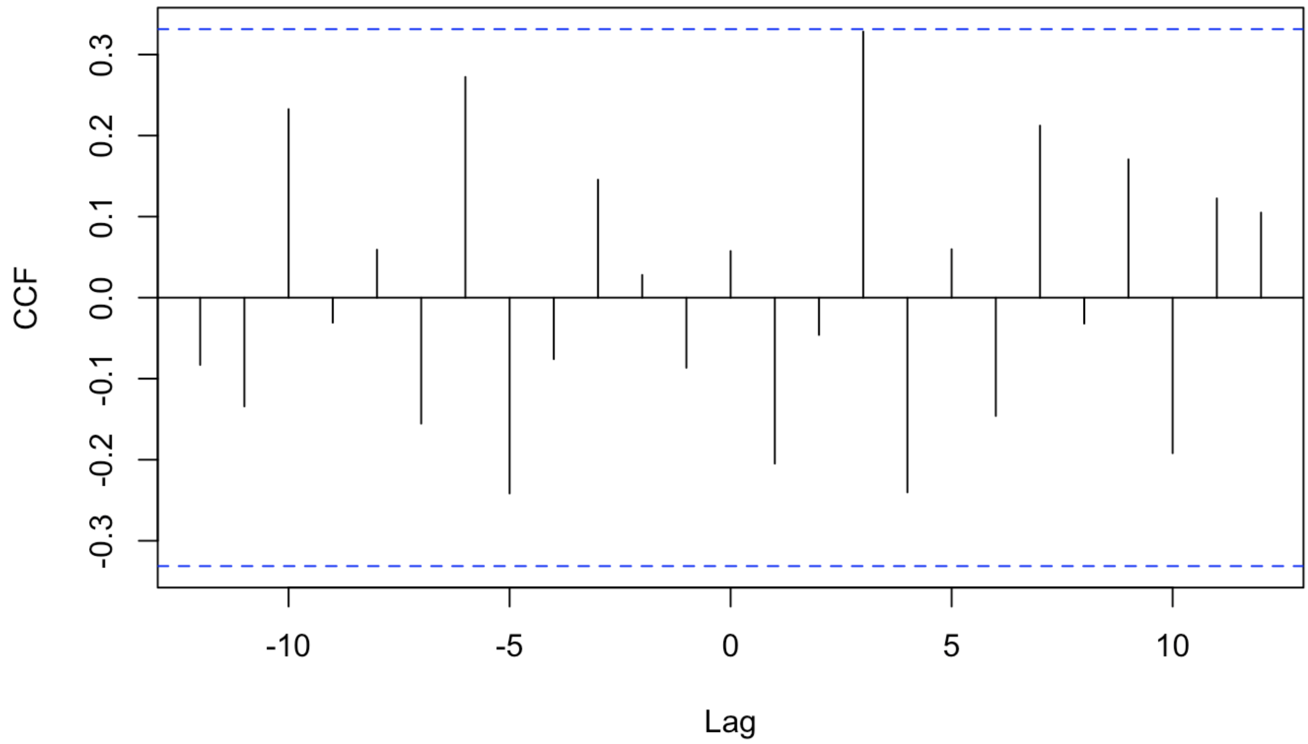
We will move on to prewhitening the series.

Hide

```
data2.dif=ts.intersect(diff(diff(PPI, lag = 4)),diff(diff(change, lag = 4)))
```

Hide

```
prewhiten(as.vector(data2.dif[,1]), as.vector(data2.dif[,2]), ylab='CCF', main="Fig9. Sample CFF after prewhitening")
```



From Fig9. it seems that there is no correlation between residential property price index (PPI) and quarterly poupulation change.

The significant correlations in Fig 7 and 8 can be said to be related to false alarm rate of CCF.

Therefore, it seems that the two series are uncorrelated, and the strong correlation pattern found between them in the dataset is indeed spurious.

Conclusion

Task 1:
From task 1, we found 1 model in each category ie DLM, DYNLM and ETS, to forecast 2 year ahead horizontal monthly solar radiation. The forecast plots in Fig5 below show the same.

Hide

```
frc.MAA = forecast(fit4.etsA , h = 2* frequency(solar))
plot(solar, type="o", xlim = c(1959, 2019),  ylab = "Solar Radiation", xlab = "Year",

     main="Fig 5. Solar Radiation Forecasts 2 years DLM,Dynlm and ETS")

lines(ts(model4p.forecasts[1:24],start = 2015),col="Purple",type="o")
```

Hide

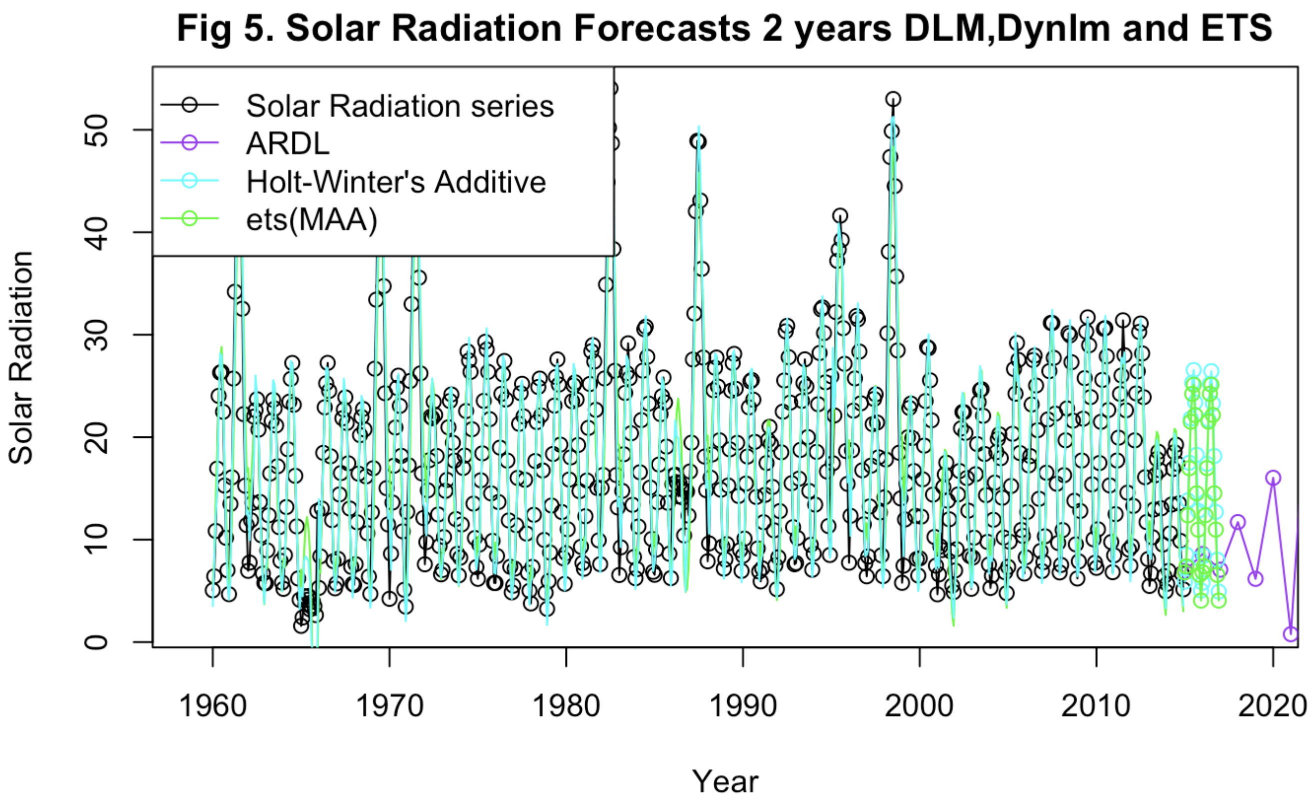
```
lines(fit3.hw$mean, type="o", col="cyan")
lines(fitted(fit4.etsA), col="green", lty=1)
```

Hide

```
lines(fitted(fit3.hw), col="cyan")
lines(frc.MAA$mean,col="green", type="o")
```

Hide

```
legend("topleft",lty=1, pch = 1, text.width = 20, col=c("black","purple","cyan","gree
n"),
       c("Solar Radiation series","ARDL","Holt-Winter's Additive", "ets(MAA)"))
```



Task 2:

The two time series residential property price index (PPI) and quarterly population change (change) are in fact uncorrelated as we saw in the research and inference stage, and from CCF and prewhitening inferences we can say that the two series are indeed spuriously correlated.