Forecasting Assignment 2

Ratnak Saha S3973869

2023-09-23

```
library(TSA)
library(urca)
library(readr)
library(magrittr)
library(tseries)
library(x12)
library(forecast)
library(seasonal)
library(dplyr)
library(lubridate)
library(tidyr)
library(car)
library(dlm)
library(dLagM)
library(dynlm)
library(lmtest)
library(Hmisc)
library(xts)
```

Introduction

- In this task, our objective is to analyze and forecast the monthly average horizontal solar radiation reaching the ground at various global locations. We will be working with data spanning from January 1960 to December 2014, which includes both the monthly average horizontal solar radiation and monthly precipitation measurements taken at the same points.

The primary goal is to generate the best two-year ahead forecasts for the solar radiation series, focusing on minimizing the Mean Absolute Scaled Error (MASE).

Task 1

Forecasting the amount of horizontal solar radiation reaching the ground at a particular location over the globe.

Importing dataset

```
getwd()
```

```
## [1] "F:/2nd semester/Forecasting"
```

```
setwd("F:/2nd semester/Forecasting/Assignment_2")
data_1<- read.csv("data1.csv")
head(data_1)
```

```
##       solar   ppt
## 1  5.051729 1.333
## 2  6.415832 0.921
## 3 10.847920 0.947
## 4 16.930264 0.615
## 5 24.030797 0.544
## 6 26.298202 0.703
```

Converting variables into separate time series

```
solar_ts <- ts(data_1$solar,start= c(1960,1),frequency = 12)
head(solar_ts)
```

```
##            Jan       Feb       Mar       Apr       May       Jun
## 1960  5.051729  6.415832 10.847920 16.930264 24.030797 26.298202
```

```
precipitation_ts <- ts(data_1$ppt,start= c(1960,1),frequency = 12)
head(precipitation_ts)
```

```
##        Jan   Feb   Mar   Apr   May   Jun
## 1960 1.333 0.921 0.947 0.615 0.544 0.703
```

```
class(solar_ts)
```

```
## [1] "ts"
```

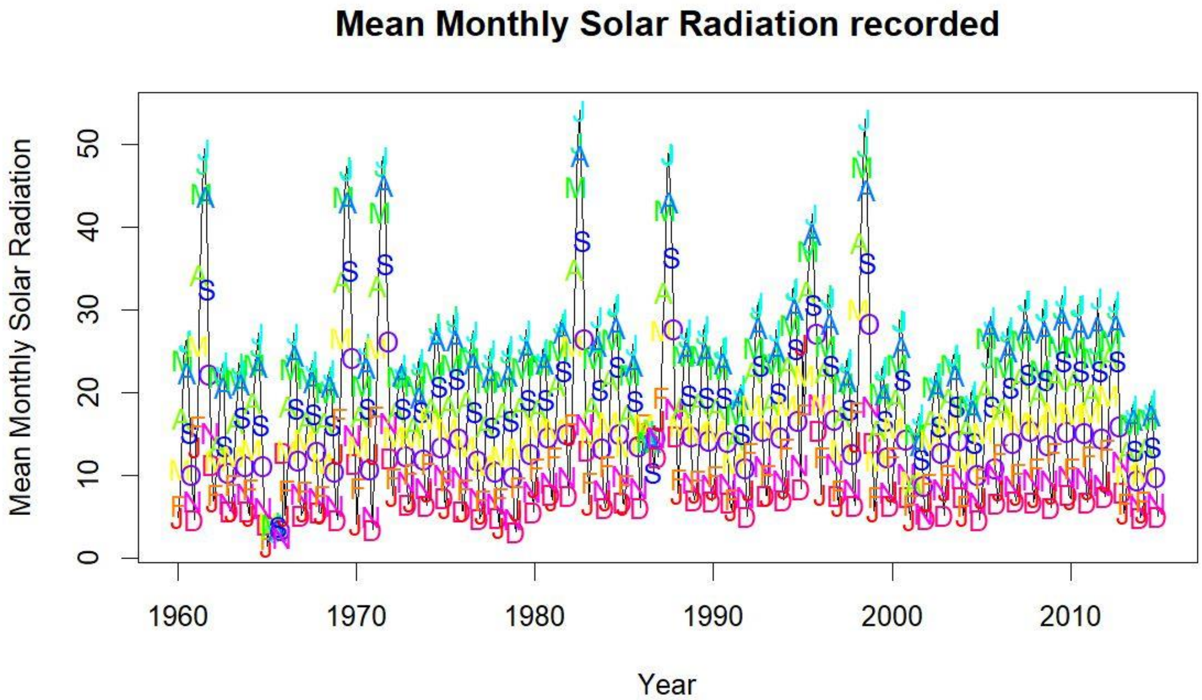
```
class(precipitation_ts)
```

```
## [1] "ts"
```

Determining the nature of series

Time-series plot of solar radiation

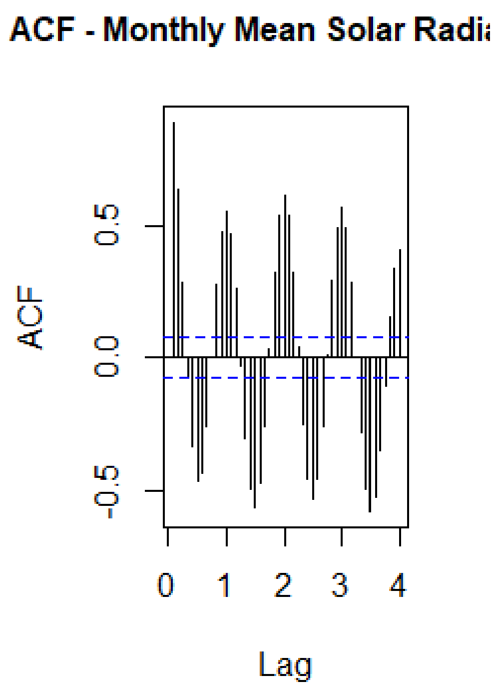
```
par(mfrow=c(1,1))
plot(solar_ts, ylab="Mean Monthly Solar Radiation", xlab = "Year", main =
"Mean Monthly Solar Radiation recorded")
points(y=solar_ts,x=time(solar_ts), pch=as.vector(season(solar_ts)),
col=rainbow(12))
```

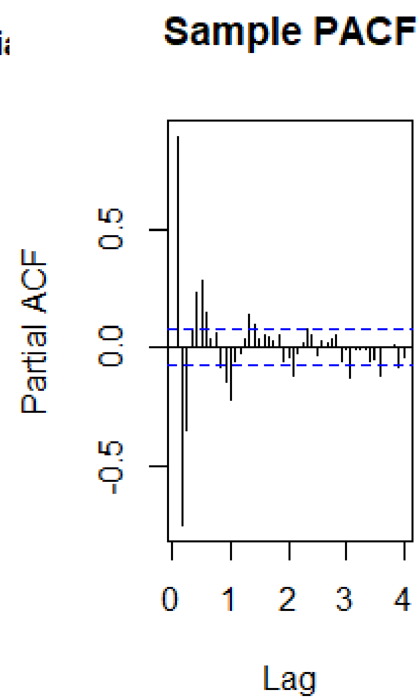


* This time series exhibits distinct seasonal patterns, with elevated values observed in June and July but lower values in December. There is no clear indication of a trend, and the variance of the data doesn't appear to change significantly over time. Upon closer examination, it becomes evident that there have been multiple interventions or significant events around the years 1962, 1982, and 1999.

To conduct a more comprehensive analysis of stationarity, seasonality, and trend in this time series, we will explore Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots, along with the Augmented Dickey-Fuller (ADF).

```
par(mfrow=c(1,2))
acf(solar_ts, lag.max = 48, main = "ACF - Monthly Mean Solar Radiation",
cex.main = 1.0)
pacf(solar_ts, lag.max = 48, main = "Sample PACF", cex.main = 0.5)
```





* Analyzing the ACF plot reveals a pronounced seasonal pattern in the data. However, it's also evident that there is a gradual decline in the series, indicating the presence of some underlying trend. This trend has the potential to dominate the correlation properties of the series.

Examining the PACF plot, we observe that the first lag is highly significant, this finding provides evidence that the series is non-stationary.

```
adf.test(solar_ts, k =ar(solar_ts)$order)

## Warning in adf.test(solar_ts, k = ar(solar_ts)$order): p-value smaller
than
## printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  solar_ts
## Dickey-Fuller = -4.557, Lag order = 25, p-value = 0.01
## alternative hypothesis: stationary
```

* Solar Radiation: With a lag order of 25 and a p-value of 0.01, we have substantial evidence to reject the null hypothesis. Rejecting the null hypothesis implies that the time series is stationary. This means that the series doesn't display patterns dependent on time, and its variance remains stable over time.

Time series regression methods

Finite distributed Lag Model

```
for (i in 1:10){
  model1 <- dlm(x = data_1$ppt, y = data_1$solar, q = i)
  cat("q =", i, "AIC =", AIC(model1$model), "BIC =", BIC(model1$model), "MASE
=", MASE(model1)$MASE, "\n")
}

## q = 1 AIC = 4728.713 BIC = 4746.676 MASE = 1.688457
## q = 2 AIC = 4712.649 BIC = 4735.095 MASE = 1.675967
## q = 3 AIC = 4688.551 BIC = 4715.478 MASE = 1.662703
## q = 4 AIC = 4663.6 BIC = 4695.003 MASE = 1.646357
## q = 5 AIC = 4644.622 BIC = 4680.499 MASE = 1.613848
## q = 6 AIC = 4637.489 BIC = 4677.837 MASE = 1.607532
## q = 7 AIC = 4632.716 BIC = 4677.532 MASE = 1.607042
## q = 8 AIC = 4625.986 BIC = 4675.267 MASE = 1.604806
## q = 9 AIC = 4615.084 BIC = 4668.827 MASE = 1.593121
## q = 10 AIC = 4602.658 BIC = 4660.858 MASE = 1.577996
```

* After conducting a thorough analysis that considered factors such as MASE, AIC, and BIC, it is determined that a lag length of 10 is the best choice. It's clear that as it increases the lag length, these metrics consistently show improvement. With this information, it can determined with the implementation of a finite Dynamic Linear Model (DLM).

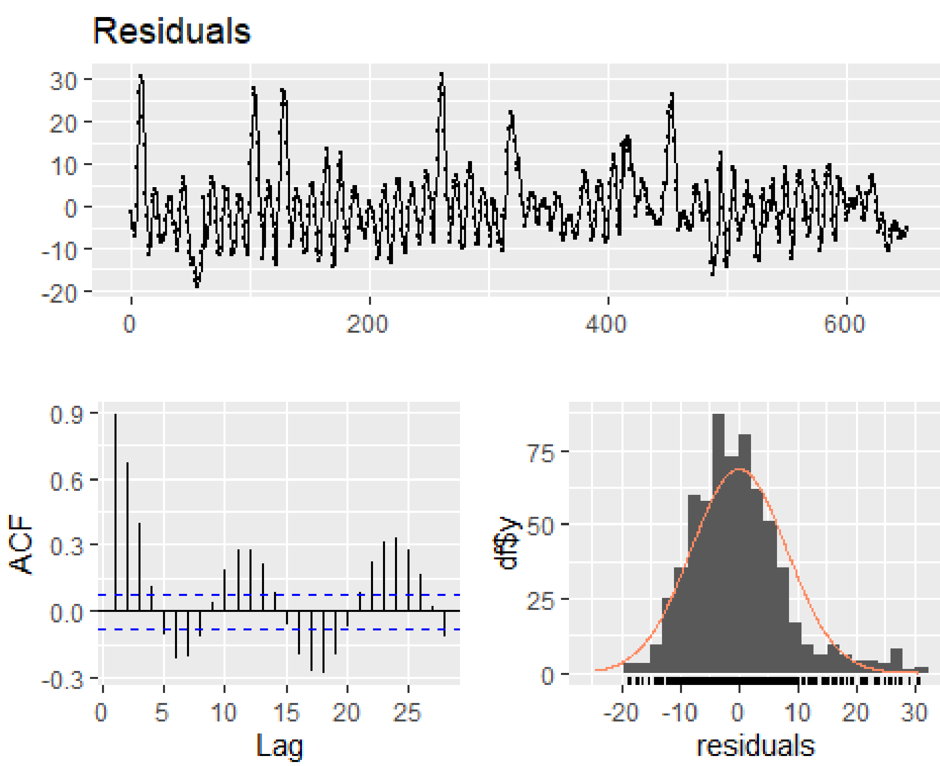
```
model1_finite = dlm(x = as.vector(precipitation_ts), y = as.vector(solar_ts),
q =10)
summary(model1)

##
## Call:
## lm(formula = model.formula, data = design)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.9353  -5.4124  -0.7911   4.0184  30.8900
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  19.0105     1.0942  17.374  < 2e-16 ***
## x.t          -7.3843     1.8995  -3.887 0.000112 ***
## x.1          -0.4763     2.5395  -0.188 0.851288
## x.2          -0.1324     2.5734  -0.051 0.958980
## x.3           1.7902     2.5781   0.694 0.487691
## x.4           1.9686     2.5808   0.763 0.445877
## x.5           3.4928     2.5807   1.353 0.176402
## x.6           0.5243     2.5787   0.203 0.838943
## x.7           1.6762     2.5797   0.650 0.516088
## x.8           0.9282     2.5673   0.362 0.717817
## x.9           0.3754     2.5338   0.148 0.882272
```

```
## x.10          -5.3798     1.8760  -2.868 0.004272 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.256 on 638 degrees of freedom
## Multiple R-squared:  0.3081, Adjusted R-squared:  0.2962
## F-statistic: 25.82 on 11 and 638 DF,  p-value: < 2.2e-16
##
## AIC and BIC values for the model:
##        AIC      BIC
## 1 4602.658 4660.858
```

```r
vif(model1$model)> 10
```

```
##   x.t   x.1   x.2   x.3   x.4   x.5   x.6   x.7   x.8   x.9  x.10
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```r
checkresiduals(model1$model)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 15
##
## data:  Residuals
## LM test = 588.43, df = 15, p-value < 2.2e-16
```

- model1_finite does not exhibit a strong fit, and most of its lag variables do not appear to be statistically significant, except for x.10. The lower Adjusted R-squared value suggests that it can only explain approximately 29.62% of the variability in the dependent variable. Interestingly, despite having multiple lag variables, Model1 does achieve statistical significance at the 5% level, as demonstrated by the F-Test for the overall significance of the model.
- As evident from the previous data, all VIF values for the lags within model1_finite are well below the threshold of 10. This suggests that there is no pronounced issue of multicollinearity in the model
- The results of the Breusch-Godfrey test reveal that the p-value is below the 5% significance threshold. Consequently, we can reject the null hypothesis, which signifies the presence of serial correlation in the residuals. A closer examination of the residuals obtained from the finite dynamic linear model (DLM) fit demonstrates that they exhibit a relatively random distribution. Notably, the autocorrelation function (ACF) reveals patterns of seasonality and serial correlation, thus confirming the existence of serial correlation within the residuals. This observation aligns with the findings of the Breusch-Godfrey test.

Polynomial DLM

```
finiteDLMauto(x = as.vector(precipitation_ts), y = as.vector(solar_ts), q.min
= 1, q.max = 10, k.order = 2,
model.type = "poly", error.type ="MASE", trace = TRUE)

##      q - k    MASE      AIC      BIC   GMRAE     MBRAE R.Adj.Sq Ljung-Box
## 10 10 - 2 1.59255 4591.904 4614.289 1.40716  -0.02753  0.29924         0
## 9   9 - 2 1.60880 4607.861 4630.253 1.42573   0.36108  0.28915         0
## 8   8 - 2 1.61903 4620.470 4642.870 1.38665   0.27726  0.28205         0
## 7   7 - 2 1.61909 4627.974 4650.382 1.38654   0.56541  0.28073         0
## 6   6 - 2 1.61999 4634.526 4656.942 1.38760   0.34815  0.28104         0
## 5   5 - 2 1.63194 4645.250 4667.673 1.43173 -91.88214  0.27675         0
## 4   4 - 2 1.65329 4664.741 4687.171 1.47467   0.97378  0.26226         0
## 3   3 - 2 1.66635 4689.018 4711.457 1.45895   0.29153  0.24157         0
## 2   2 - 2 1.67597 4712.649 4735.095 1.42249   0.14593  0.22173         0
## 1   1 - 2 1.68846 4728.713 4746.676 1.43103  -1.07642  0.21036         0
```

* Taking into account the consistent reduction in MASE, AIC, and BIC values with an increasing lag length (q), as demonstrated in the previous code, a lag length of 10 will be employed. Furthermore, adjustments to the polynomial order will be made by setting it to 2.

```
model1_poly <- polyDlm(x=as.vector(precipitation_ts), y=as.vector(solar_ts),
q=10, k = 2, show.beta = TRUE)

## Estimates and t-tests for beta coefficients:
##         Estimate Std. Error t value  P(>|t|)
## beta.0    -5.040      0.435 -11.600 2.39e-28
## beta.1    -2.320      0.314  -7.400 4.33e-13
```

```
## beta.2    -0.188      0.251  -0.749 4.54e-01
## beta.3     1.370      0.237   5.770 1.23e-08
## beta.4     2.340      0.243   9.600 1.74e-20
## beta.5     2.720      0.248  11.000 7.32e-26
## beta.6     2.530      0.243  10.400 1.40e-23
## beta.7     1.750      0.235   7.430 3.39e-13
## beta.8     0.387      0.249   1.560 1.20e-01
## beta.9    -1.560      0.312  -4.990 7.68e-07
## beta.10   -4.090      0.433  -9.440 7.08e-20
```

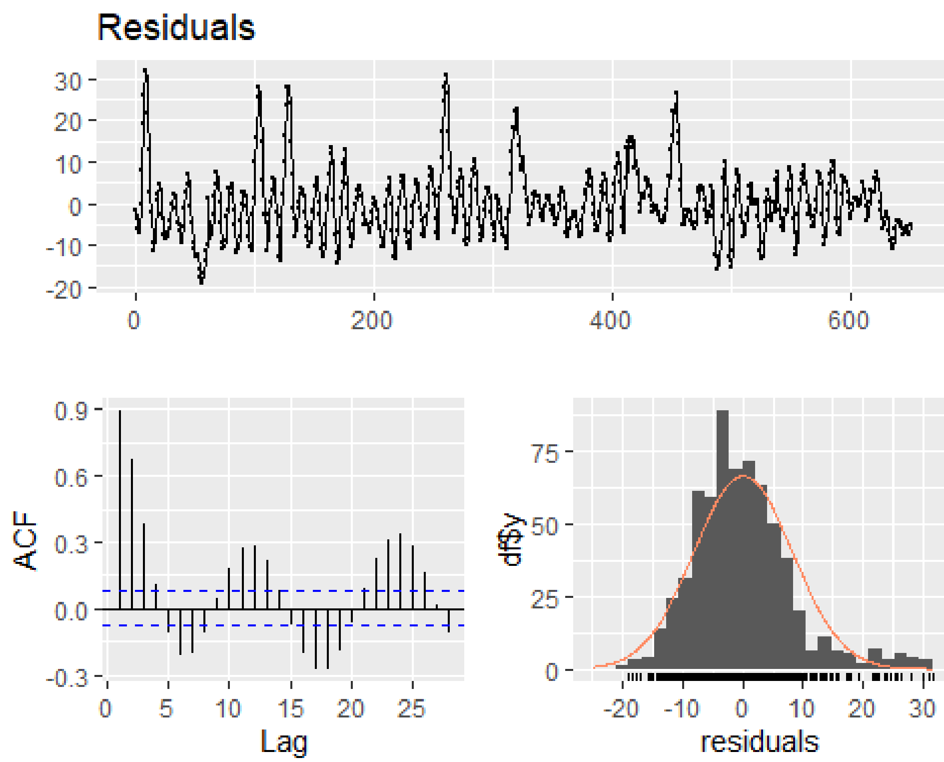
```r
summary(model1_poly)
```

```
## 
## Call:
## "Y ~ (Intercept) + X.t"
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -19.012  -5.343  -1.090   4.097  31.641 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 18.77806    1.08588   17.29   <2e-16 ***
## z.t0        -5.04398    0.43502  -11.60   <2e-16 ***
## z.t1         3.01112    0.18300   16.45   <2e-16 ***
## z.t2        -0.29153    0.01761  -16.56   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 8.237 on 646 degrees of freedom
## Multiple R-squared:  0.3025, Adjusted R-squared:  0.2992 
## F-statistic: 93.38 on 3 and 646 DF,  p-value: < 2.2e-16
```

```r
vif(model1_poly$model)>10
```

```
##  z.t0  z.t1  z.t2 
## FALSE  TRUE  TRUE
```

```r
checkresiduals(model1_poly$model)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 580.37, df = 10, p-value < 2.2e-16
```

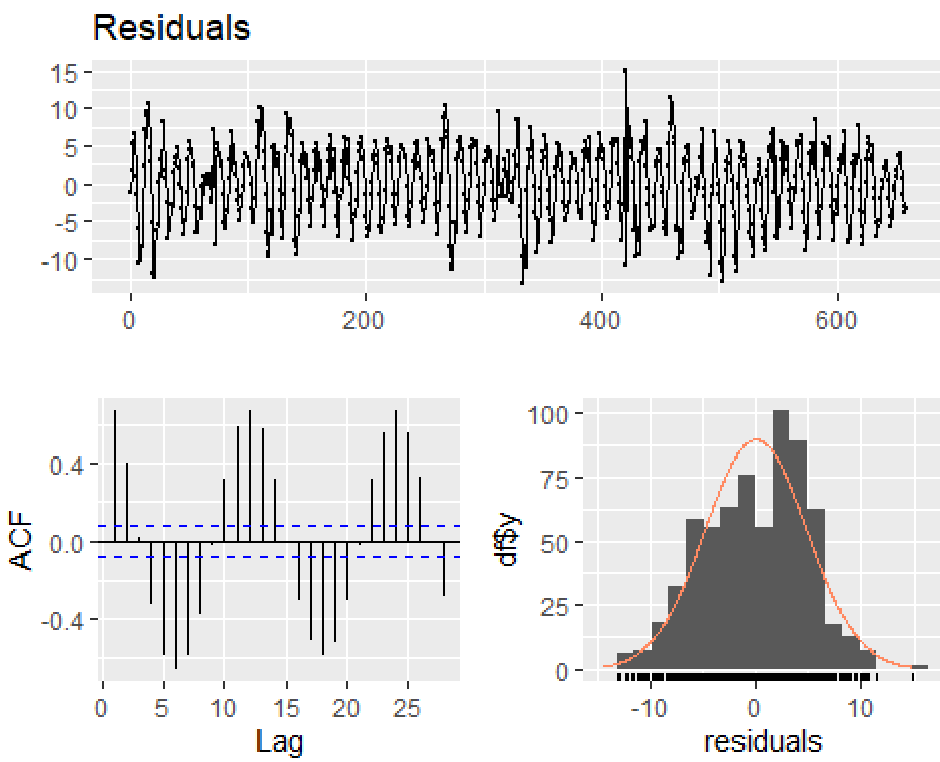
- The model1_poly displays significance in nearly all of its lag variables. However, it has a relatively low Adjusted R-squared value, suggesting that it can only account for approximately 30% of the variability in the dependent variable. The model1_poly attains statistical significance at the 5% significance level, as demonstrated by the F-Test for overall model significance.

- The model exhibits some limitations due to the presence of multicollinearity, as indicated by VIF values exceeding the threshold of 10. This multicollinearity issue can impact the model's performance and interpretation.

- As indicated by the results of the Breusch-Godfrey test, the p-value falls below the 5% significance threshold, which leads us to reject the null hypothesis. This rejection is an indication that serial correlation exists within the residuals. Furthermore, upon closer examination of the residuals derived from the model, it becomes evident that they follow a random pattern. Particularly noteworthy is the presence of seasonal lags in the autocorrelation function (ACF), which serves as confirmation of the existence of serial correlation. This observation aligns with the conclusions drawn from the Breusch-Godfrey test and points to the continued presence of seasonality in the residuals.

Koyck Transformation DLM

```
model_1_koyck <- koyckDlm(x=as.vector(precipitation_ts), y =
as.vector(solar_ts))
summary(model_1_koyck, diagnostics = TRUE)

##
## Call:
## "Y ~ (Intercept) + Y.1 + X.t"
##
## 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.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
##
## Diagnostic tests:
##                  df1 df2 statistic       p-value
## Weak instruments   1 656  710.7209 1.191744e-106
## Wu-Hausman         1 655  146.8017  1.248856e-30
##
##                               alpha     beta       phi
## Geometric coefficients:  -154.0203 5.346844 0.9854613

checkresiduals(model_1_koyck$model)
```




```
##
##  Ljung-Box test
##
## data:  Residuals
## Q* = 1413.2, df = 10, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 10
```

```r
vif(model_1_koyck$model) > 10
```

```
##   Y.1   X.t
## FALSE FALSE
```

- The summary results indicate that model_1_koyck outperforms the other two models, as evidenced by a more favorable F-test p-value and a higher Adjusted R-squared value. This signifies the significance of the explanatory variable, with the model explaining approximately 75.91% of the variance in the dependent variable.Furthermore, the Wu-Hausman test results, conducted at a 5% significance level, provide strong evidence of a meaningful relationship between the independent variable and the error term. This underscores the importance of the chosen independent variable in explaining variations in the dependent variable.
- Analyzing the residuals obtained from the polynomial Dynamic Linear Model (DLM), it becomes clear that they do not follow a random pattern. Particularly striking is the presence of seasonal lags in the autocorrelation function (ACF). Importantly,

there is no indication of multicollinearity in the model, as indicated by VIF values that do not exceed the threshold of 10. This absence of multicollinearity enhances the reliability of the model's parameter estimates and interpretations. It's worth noting that the histogram of residuals doesn't display signs of normality. However, when compared to other models, this model stands out as it doesn't exhibit multicollinearity issues.

Auto Regressive DLM

```
for (i in 1:5){
  for(j in 1:5){
    model_auto = ardlDlm(x=
as.vector(precipitation_ts),y=as.vector(solar_ts), p = i , q = j )
    cat("p =", i, "q =", j, "AIC =", AIC(model_auto$model), "BIC =",
BIC(model_auto$model), "MASE =", MASE(model_auto)$MASE, "\n")
  }
}

## p = 1 q = 1 AIC = 3712.311 BIC = 3734.765 MASE = 0.8392434
## p = 1 q = 2 AIC = 3239.416 BIC = 3266.352 MASE = 0.4971918
## p = 1 q = 3 AIC = 3143.522 BIC = 3174.936 MASE = 0.4740063
## p = 1 q = 4 AIC = 3138.399 BIC = 3174.288 MASE = 0.4697571
## p = 1 q = 5 AIC = 3100.283 BIC = 3140.644 MASE = 0.450425
## p = 2 q = 1 AIC = 3639.223 BIC = 3666.159 MASE = 0.7834855
## p = 2 q = 2 AIC = 3229.051 BIC = 3260.476 MASE = 0.4951319
## p = 2 q = 3 AIC = 3137.634 BIC = 3173.535 MASE = 0.4738939
## p = 2 q = 4 AIC = 3132.962 BIC = 3173.337 MASE = 0.4702773
## p = 2 q = 5 AIC = 3097.288 BIC = 3142.134 MASE = 0.4503599
## p = 3 q = 1 AIC = 3608.793 BIC = 3640.207 MASE = 0.7572489
## p = 3 q = 2 AIC = 3226.623 BIC = 3262.524 MASE = 0.4955334
## p = 3 q = 3 AIC = 3139.409 BIC = 3179.798 MASE = 0.4737144
## p = 3 q = 4 AIC = 3134.777 BIC = 3179.638 MASE = 0.4701162
## p = 3 q = 5 AIC = 3098.808 BIC = 3148.139 MASE = 0.4502885
## p = 4 q = 1 AIC = 3602.664 BIC = 3638.553 MASE = 0.7580664
## p = 4 q = 2 AIC = 3224.285 BIC = 3264.66 MASE = 0.4959949
## p = 4 q = 3 AIC = 3131.289 BIC = 3176.15 MASE = 0.4695096
## p = 4 q = 4 AIC = 3131.424 BIC = 3180.772 MASE = 0.4665123
## p = 4 q = 5 AIC = 3096.024 BIC = 3149.839 MASE = 0.4479481
## p = 5 q = 1 AIC = 3599.402 BIC = 3639.764 MASE = 0.7572617
## p = 5 q = 2 AIC = 3221.853 BIC = 3266.699 MASE = 0.4954501
## p = 5 q = 3 AIC = 3127.103 BIC = 3176.434 MASE = 0.4675479
## p = 5 q = 4 AIC = 3127.868 BIC = 3181.684 MASE = 0.4651969
## p = 5 q = 5 AIC = 3097.877 BIC = 3156.177 MASE = 0.4479311
```

- The above output shows p = 5 and Q =5 with lowest MASE, so further exploration will be taken into consideration based on the lowest Mase.

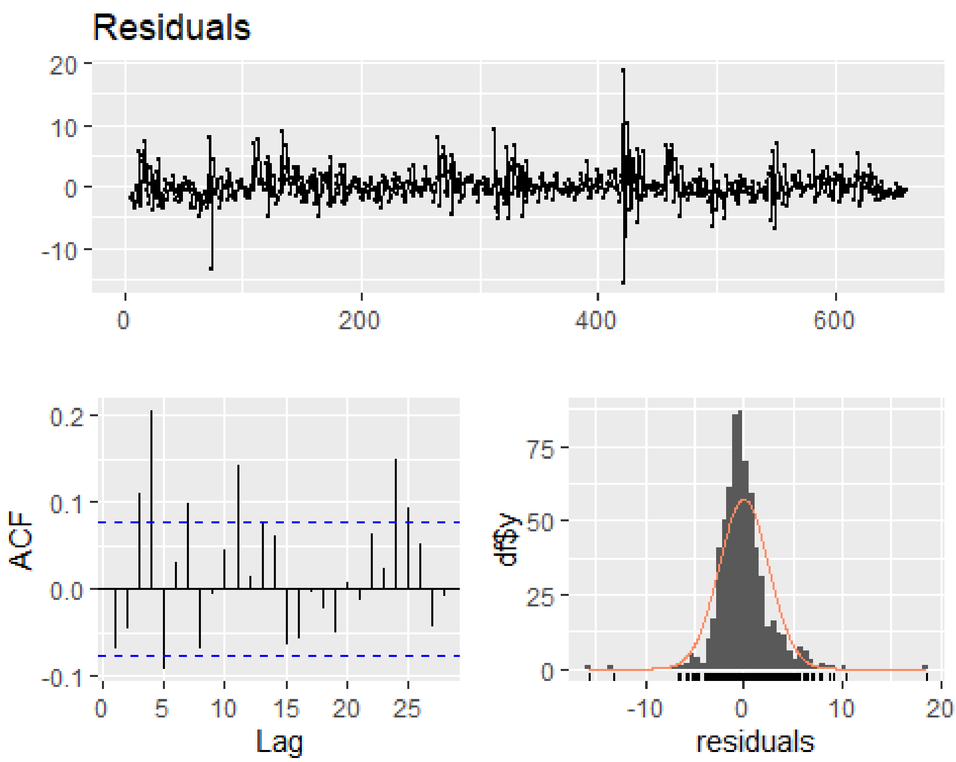
```
model_auto_1 <- ardlDlm(x=as.vector(precipitation_ts),
y=as.vector(solar_ts),p = 5, q = 5)
summary(model_auto_1)
```

```
##
## Time series regression with "ts" data:
## Start = 6, End = 660
##
## Call:
## dynlm(formula = as.formula(model.text), data = data, start = 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.5959  -1.3825  -0.2646   1.0410  18.5812
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.50740    0.45434   5.519 4.96e-08 ***
## X.t         -0.61416    0.54804  -1.121 0.262863
## X.1          0.78299    0.77670   1.008 0.313788
## X.2          1.26543    0.79241   1.597 0.110772
## X.3          0.75184    0.79227   0.949 0.342998
## X.4         -1.00181    0.77678  -1.290 0.197617
## X.5         -0.21024    0.55439  -0.379 0.704639
## Y.1          1.27063    0.03867  32.861  < 2e-16 ***
## Y.2         -0.01727    0.06264  -0.276 0.782907
## Y.3         -0.40297    0.06043  -6.669 5.56e-11 ***
## Y.4         -0.23273    0.06229  -3.737 0.000203 ***
## Y.5          0.21571    0.03802   5.673 2.12e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.548 on 643 degrees of freedom
## Multiple R-squared:  0.9338, Adjusted R-squared:  0.9327
## F-statistic: 824.9 on 11 and 643 DF,  p-value: < 2.2e-16
```

```
vif(model_auto_1$model)>10
```

```
##       X.t L(X.t, 1) L(X.t, 2) L(X.t, 3) L(X.t, 4) L(X.t, 5) L(y.t, 1)
L(y.t, 2)
##     FALSE     FALSE     FALSE     FALSE     FALSE     FALSE      TRUE
TRUE
## L(y.t, 3) L(y.t, 4) L(y.t, 5)
##      TRUE      TRUE      TRUE
```

```
checkresiduals(model_auto_1$model)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 15
##
## data:  Residuals
## LM test = 107.98, df = 15, p-value = 3.937e-16
```

br>

- The above summary shows few significant lags. With the highest Adjusted R-squared value that explains 93.27% of the variation of the dependent variable. The p-value is below the 5% significance threshold, indicating the significance of the model. This model provided be the best summary statistics compared to the previous models.

- The p-value is less than 0.05 indicates there is presence of serial correlation among the residuals. The residual plot. The residuals plot is not non-random and seasonal pattern can be observed. ACF plot shows significant lags that are will above the 95% confidence interval meaning the there is presence of auto-correlation in the residuals. Histogram seems to be normally distributed.

- The model exhibits multicollinearity as many of its elements have VIF values exceeding 10.

```
all_mase_values <- MASE(model1, model1_poly,model_1_koyck, model_auto_1)
all_mase_values
```

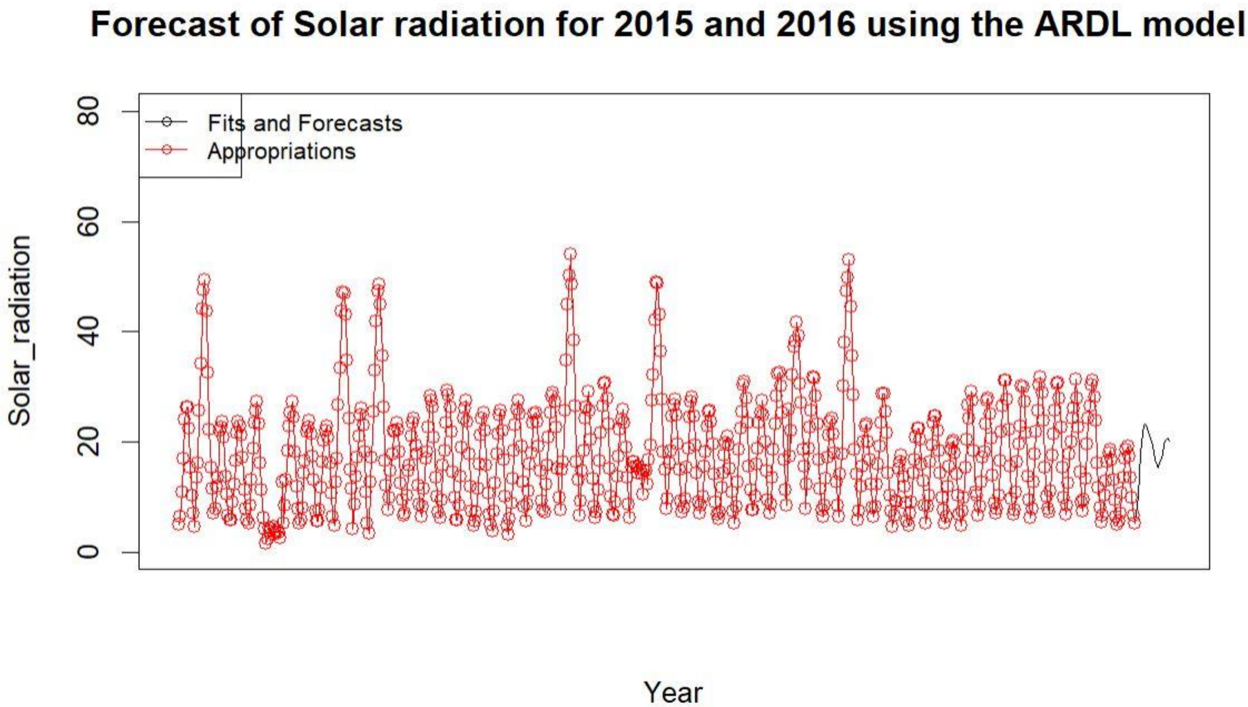
```
##                   n      MASE
## model1        650 1.5779955
## model1_poly   650 1.5925546
## model_1_koyck 659 1.0324829
## model_auto_1  655 0.4479311
```

- We can conclude that model_auto_1 of the time series regression models can be a good fit cause it has lower MASE values and higher adjusted R squared value of 0.9327 compared to any Distributed Lag Models.Model_auto_1 has the best summary statistics compared the the other models but the presence of multicollinearity raises a concern about the reliability of the forecast.

```
forecasts.ardldlm = dLagM::forecast(model = model_auto_1 ,  x = 
c(0.189009998,0.697262522,0.595213491,0.487388526,0.261677017,0.808606651,0.9
4186202,0.905636325,1.059964682,0.341438784,0.525805322,0.602471062,0.1098606
32,0.781464707,0.69685501,0.502413906,0.649385609,0.745960773,0.663047123,0.5
33770112,0.61542621,0.54606508,0.142673325,0.013650407) , h = 24)$forecasts
forecasts.ardldlm
```

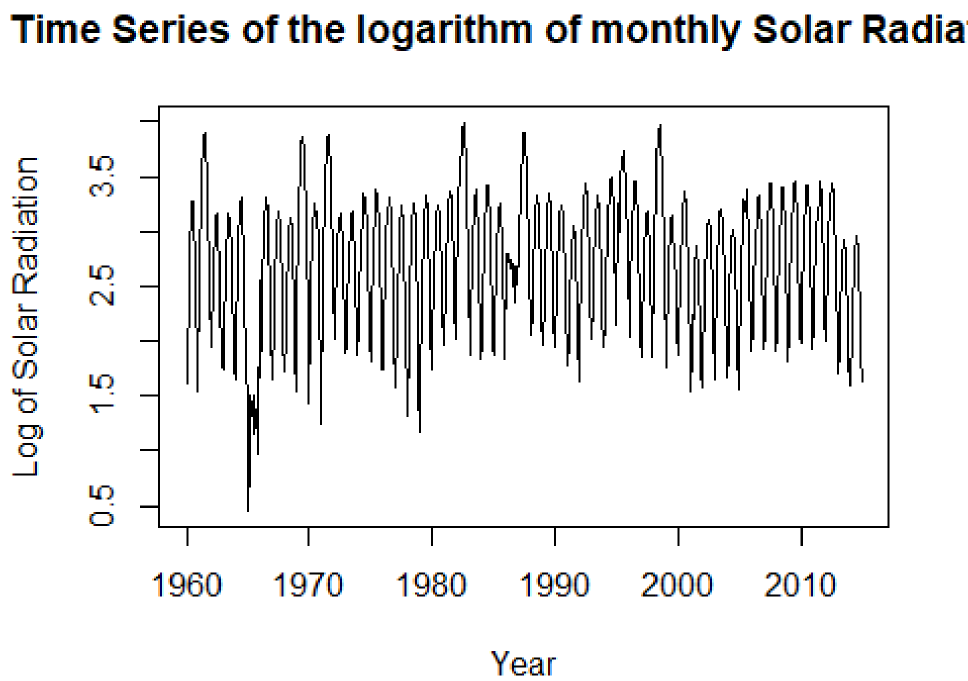
```
##  [1]  6.992061 10.069235 14.045730 17.888778 21.564606 23.138383 23.239398
##  [8] 22.601168 21.833247 21.370450 20.612553 19.449640 18.121295 16.825834
## [15] 15.726348 15.411218 16.242858 16.939873 18.024494 19.351433 20.186320
## [22] 20.557309 20.639830 20.210939
```

```
y.extended = c(data_1$solar, forecasts.ardldlm)
y.extented = ts(y.extended, start = c(1960,1), frequency = 12)
plot(y.extended,type="l",xaxt="n",ylim= c(0, 80), ylab = "Solar_radiation", 
xlab = "Year", main="Forecast of Solar radiation for 2015 and 2016 using the 
ARDL model.")
lines(data_1$solar,col="Red",type="o")
legend("topleft",lty=1, pch = 1, text.width = 16, col=c("black","red"),
       c("Fits and Forecasts", "Appropriations"), cex = 0.8)
```



Dynamic Linear Modelling

```
log_solar_ts <- log(solar_ts)

par(mfrow=c(1,1))
plot(log_solar_ts,ylab='Log of Solar Radiation',xlab='Year',
     main = "Time Series of the logarithm of monthly Solar Radiation")
```



* The log transformation has evened out the data and the data looks more stable in terms of mean and variance and it can be observed that there is an intervention in the year 1966.

```
Y.t <- log(solar_ts)
t <- 69
S.t <- 1*(seq(log(solar_ts))==T)
S.t.1 <- lag(S.t, +1)

dynlm_1<-  dynlm(Y.t~ L(Y.t, k=1) + S.t + trend(Y.t)+ season(Y.t))

summary(dynlm_1)

##
## Time series regression with "ts" data:
## Start = 1960(2), End = 2014(12)
##
## Call:
## dynlm(formula = Y.t ~ L(Y.t, k = 1) + S.t + trend(Y.t) + season(Y.t))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.09371 -0.05078 -0.00037  0.05472  1.68104
##
## Coefficients: (1 not defined because of singularities)
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.3368955  0.0474146   7.105 3.19e-12 ***
## L(Y.t, k = 1)   0.8562733  0.0203497  42.078  < 2e-16 ***
```

```
## S.t                      NA         NA      NA       NA    
## trend(Y.t)      0.0002171  0.0004588   0.473 0.636274    
## season(Y.t)Feb  0.2553540  0.0356002   7.173 2.02e-12 ***
## season(Y.t)Mar  0.4088850  0.0363697  11.242  < 2e-16 ***
## season(Y.t)Apr  0.2589237  0.0390495   6.631 7.07e-11 ***
## season(Y.t)May  0.3038046  0.0409952   7.411 3.96e-13 ***
## season(Y.t)Jun  0.2196540  0.0434474   5.056 5.60e-07 ***
## season(Y.t)Jul  0.1544582  0.0447472   3.452 0.000593 ***
## season(Y.t)Aug  0.0258825  0.0450880   0.574 0.566138    
## season(Y.t)Sep -0.1094457  0.0438069  -2.498 0.012724 *  
## season(Y.t)Oct -0.2627235  0.0413002  -6.361 3.79e-10 ***
## season(Y.t)Nov -0.3817767  0.0382079  -9.992  < 2e-16 ***
## season(Y.t)Dec -0.3012671  0.0359980  -8.369 3.61e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.1857 on 645 degrees of freedom
## Multiple R-squared:  0.9052, Adjusted R-squared:  0.9033 
## F-statistic: 473.7 on 13 and 645 DF,  p-value: < 2.2e-16
```

* Analyzing dynlm_1, it's evident that the first lag of Y.t is notably significant. While the trend component doesn't show significance, several seasonal components exhibit significance. Additionally, there remains some serial correlation in the residuals. This model displays a higher R-squared value and a p-value that indicates significance.

```
dynlm_2 <-  dynlm(Y.t~ L(Y.t, k=2) + S.t + trend(Y.t)+ season(Y.t))
summary(dynlm_2)

## 
## Time series regression with "ts" data:
## Start = 1960(3), End = 2014(12)
## 
## Call:
## dynlm(formula = Y.t ~ L(Y.t, k = 2) + S.t + trend(Y.t) + season(Y.t))
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.24390 -0.07164  0.00412  0.07839  1.65531 
## 
## Coefficients: (1 not defined because of singularities)
##                  Estimate Std. Error t value Pr(>|t|)    
## (Intercept)     0.3258449  0.0662511   4.918 1.11e-06 ***
## L(Y.t, k = 2)   0.7530208  0.0259322  29.038  < 2e-16 ***
## S.t                    NA         NA      NA       NA    
## trend(Y.t)      0.0003415  0.0005856   0.583  0.55995    
## season(Y.t)Feb  0.5204552  0.0460675  11.298  < 2e-16 ***
## season(Y.t)Mar  0.8892350  0.0456500  19.479  < 2e-16 ***
## season(Y.t)Apr  0.8645651  0.0453695  19.056  < 2e-16 ***
## season(Y.t)May  0.7727101  0.0472316  16.360  < 2e-16 ***
## season(Y.t)Jun  0.7228309  0.0490502  14.737  < 2e-16 ***
```

```
## season(Y.t)Jul  0.5811291  0.0515648  11.270  < 2e-16 ***
## season(Y.t)Aug  0.3945853  0.0529622   7.450 3.01e-13 ***
## season(Y.t)Sep  0.1486189  0.0533342   2.787  0.00548 **
## season(Y.t)Oct -0.1184529  0.0519476  -2.280  0.02292 *
## season(Y.t)Nov -0.3642872  0.0493521  -7.381 4.86e-13 ***
## season(Y.t)Dec -0.3788571  0.0465264  -8.143 2.00e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2365 on 644 degrees of freedom
## Multiple R-squared:  0.846,  Adjusted R-squared:  0.8429
## F-statistic: 272.2 on 13 and 644 DF,  p-value: < 2.2e-16
```

- Observing dynlm_2, it becomes apparent that the second lag of Y.t holds significant importance. Several seasonal components also show significance. However, it's worth noting that dynlm_2 exhibits a lower Adjusted R-squared value when compared to dynlm_1.

```
dynlm_3<-  dynlm(Y.t~ L(Y.t, k=1) + S.t + S.t.1 +trend(Y.t)+ season(Y.t))
summary(dynlm_3)
```

```
##
## Time series regression with "ts" data:
## Start = 1960(2), End = 2014(12)
##
## Call:
## dynlm(formula = Y.t ~ L(Y.t, k = 1) + S.t + S.t.1 + trend(Y.t) +
##     season(Y.t))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.09438 -0.05054  0.00000  0.05466  1.68005
##
## Coefficients: (1 not defined because of singularities)
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.3383136  0.0474848   7.125 2.80e-12 ***
## L(Y.t, k = 1)   0.8558122  0.0203709  42.012  < 2e-16 ***
## S.t                    NA         NA      NA       NA
## S.t.1          -0.1233791  0.1880510  -0.656 0.511998
## trend(Y.t)      0.0001980  0.0004599   0.431 0.666961
## season(Y.t)Feb  0.2576160  0.0357824   7.200 1.69e-12 ***
## season(Y.t)Mar  0.4090491  0.0363866  11.242  < 2e-16 ***
## season(Y.t)Apr  0.2592833  0.0390706   6.636 6.83e-11 ***
## season(Y.t)May  0.3042627  0.0410192   7.418 3.78e-13 ***
## season(Y.t)Jun  0.2202174  0.0434750   5.065 5.33e-07 ***
## season(Y.t)Jul  0.1550731  0.0447768   3.463 0.000569 ***
## season(Y.t)Aug  0.0265118  0.0451181   0.588 0.557001
## season(Y.t)Sep -0.1088632  0.0438353  -2.483 0.013265 *
## season(Y.t)Oct -0.2622432  0.0413249  -6.346 4.17e-10 ***
```

```
## season(Y.t)Nov -0.3814545  0.0382280  -9.978  < 2e-16 ***
## season(Y.t)Dec -0.3011348  0.0360145  -8.361 3.83e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1858 on 644 degrees of freedom
## Multiple R-squared:  0.9053, Adjusted R-squared:  0.9032
## F-statistic: 439.5 on 14 and 644 DF,  p-value: < 2.2e-16
```

```
dynlm_3<-  dynlm(Y.t~ L(Y.t, k=1) + S.t + S.t.1 +trend(Y.t)+ season(Y.t))
summary(dynlm_3)
```

```
##
## Time series regression with "ts" data:
## Start = 1960(2), End = 2014(12)
##
## Call:
## dynlm(formula = Y.t ~ L(Y.t, k = 1) + S.t + S.t.1 + trend(Y.t) +
##     season(Y.t))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.09438 -0.05054  0.00000  0.05466  1.68005
##
## Coefficients: (1 not defined because of singularities)
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.3383136  0.0474848   7.125 2.80e-12 ***
## L(Y.t, k = 1)   0.8558122  0.0203709  42.012  < 2e-16 ***
## S.t                    NA         NA      NA       NA
## S.t.1          -0.1233791  0.1880510  -0.656 0.511998
## trend(Y.t)      0.0001980  0.0004599   0.431 0.666961
## season(Y.t)Feb  0.2576160  0.0357824   7.200 1.69e-12 ***
## season(Y.t)Mar  0.4090491  0.0363866  11.242  < 2e-16 ***
## season(Y.t)Apr  0.2592833  0.0390706   6.636 6.83e-11 ***
## season(Y.t)May  0.3042627  0.0410192   7.418 3.78e-13 ***
## season(Y.t)Jun  0.2202174  0.0434750   5.065 5.33e-07 ***
## season(Y.t)Jul  0.1550731  0.0447768   3.463 0.000569 ***
## season(Y.t)Aug  0.0265118  0.0451181   0.588 0.557001
## season(Y.t)Sep -0.1088632  0.0438353  -2.483 0.013265 *
## season(Y.t)Oct -0.2622432  0.0413249  -6.346 4.17e-10 ***
## season(Y.t)Nov -0.3814545  0.0382280  -9.978  < 2e-16 ***
## season(Y.t)Dec -0.3011348  0.0360145  -8.361 3.83e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1858 on 644 degrees of freedom
## Multiple R-squared:  0.9053, Adjusted R-squared:  0.9032
## F-statistic: 439.5 on 14 and 644 DF,  p-value: < 2.2e-16
```

```
dynlm_4 <-  dynlm(Y.t~ L(Y.t, k=1) + S.t + trend(Y.t))
summary(dynlm_4)
```

```
##
## Time series regression with "ts" data:
## Start = 1960(2), End = 2014(12)
##
## Call:
## dynlm(formula = Y.t ~ L(Y.t, k = 1) + S.t + trend(Y.t))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.15036 -0.26782  0.07499  0.23506  1.32204
##
## Coefficients: (1 not defined because of singularities)
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.988e-01  5.931e-02   6.725 3.83e-11 ***
## L(Y.t, k = 1)  8.534e-01  2.039e-02  41.855  < 2e-16 ***
## S.t                   NA         NA      NA       NA
## trend(Y.t)    -1.212e-05  7.675e-04  -0.016    0.987
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3117 on 656 degrees of freedom
## Multiple R-squared:  0.7283, Adjusted R-squared:  0.7275
## F-statistic: 879.4 on 2 and 656 DF,  p-value: < 2.2e-16
```

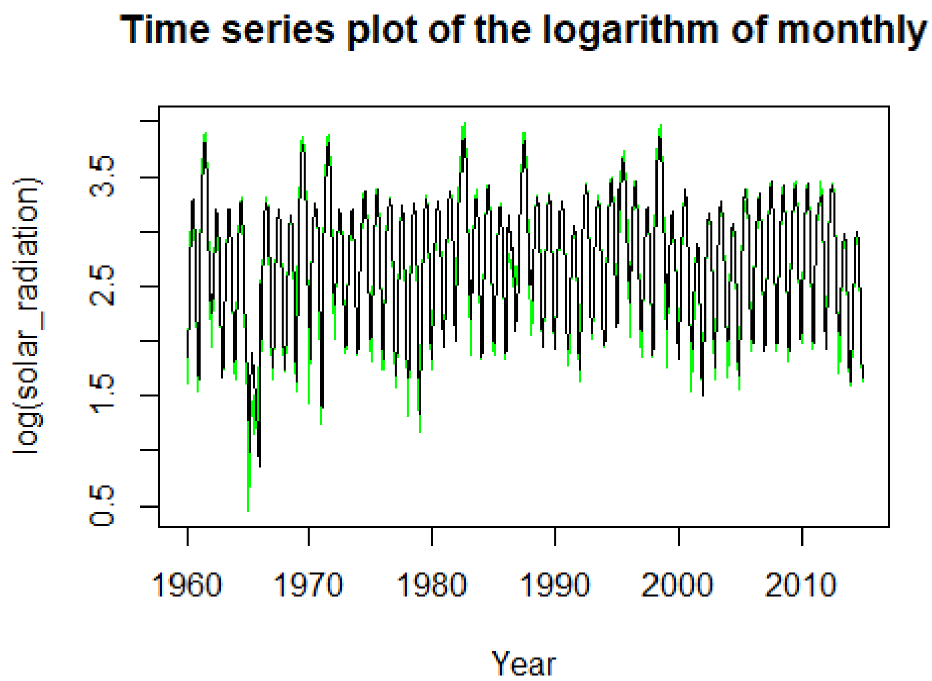
- Looking at the dynlm_3 we can say that there are several season components that are significant. The model dynlm_4 has lower Adjusted R-squared compared to any Dynamic Linear Models, so we will proceed with model dynlm_3.

```
dynamic_mase<- MASE(lm(dynlm_1),lm(dynlm_2),lm(dynlm_3),lm(dynlm_4))
dynamic_mase
```

```
##               n      MASE
## lm(dynlm_1) 659 0.3641351
## lm(dynlm_2) 658 0.5212544
## lm(dynlm_3) 659 0.3635303
## lm(dynlm_4) 659 0.9684986
```

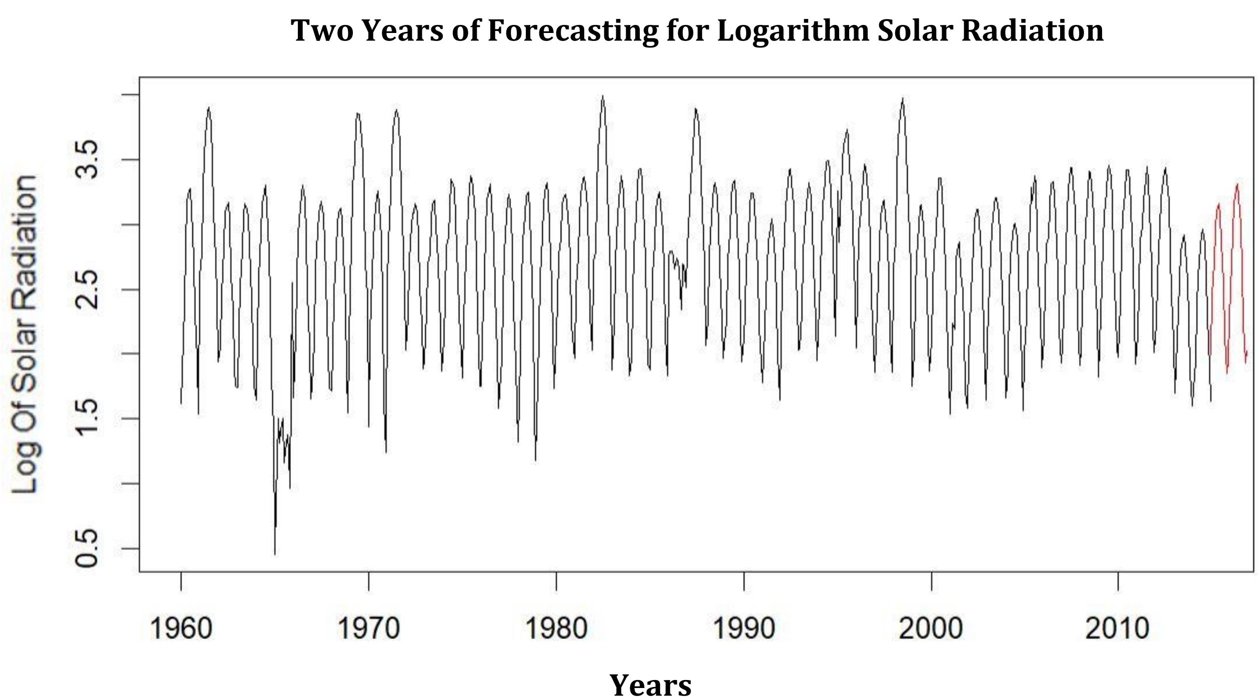
* Out of all the models dynlm_3 has the lowest MASE with seasonality and higher adjusted R-Squared value that captures 90% of the variability of the dependent variable and it is considered as the best model to conduct summary and residual analysis for the model.

```
plot(log(solar_ts),main="Time series plot of the logarithm of monthly", ylab
= "log(solar_radiation)",
,type="l",col="green", xlab = "Year")
lines(dynlm_3$fitted.values)
```



- Forecast of dynlm_4 model

```
q = 24
n = nrow(dynlm_3$model)
rad.frc = array(NA , (n + q))
rad.frc[1:n] = Y.t[2:length(Y.t)]
trend = array(NA,q)
trend.start = dynlm_3$model[n,"trend(Y.t)"]
trend = seq(trend.start , trend.start + q/12, 1/12)
for (i in 1:q){
months = array(0,11)
months[(i+0)%%12] = 1
# Data ends in May, to start the new forecast from JUNE, put i + 4.
data.new = c(1,rad.frc[n-1+i], S.t.1[n] , S.t[n] ,trend[i],months)
rad.frc[n+i] = as.vector(dynlm_3$coefficients) %*% data.new
}

par(mfrow=c(1,1))
plot(Y.t,xlim=c(1960,2015),ylab='Log Of Solar Radiation',xlab='Year',main =
"Two years forecasting for lograithm solar radiation)")
lines(ts(rad.frc[(n+1):(n+q)],start=c(2015,0),frequency = 12),col="green")
```

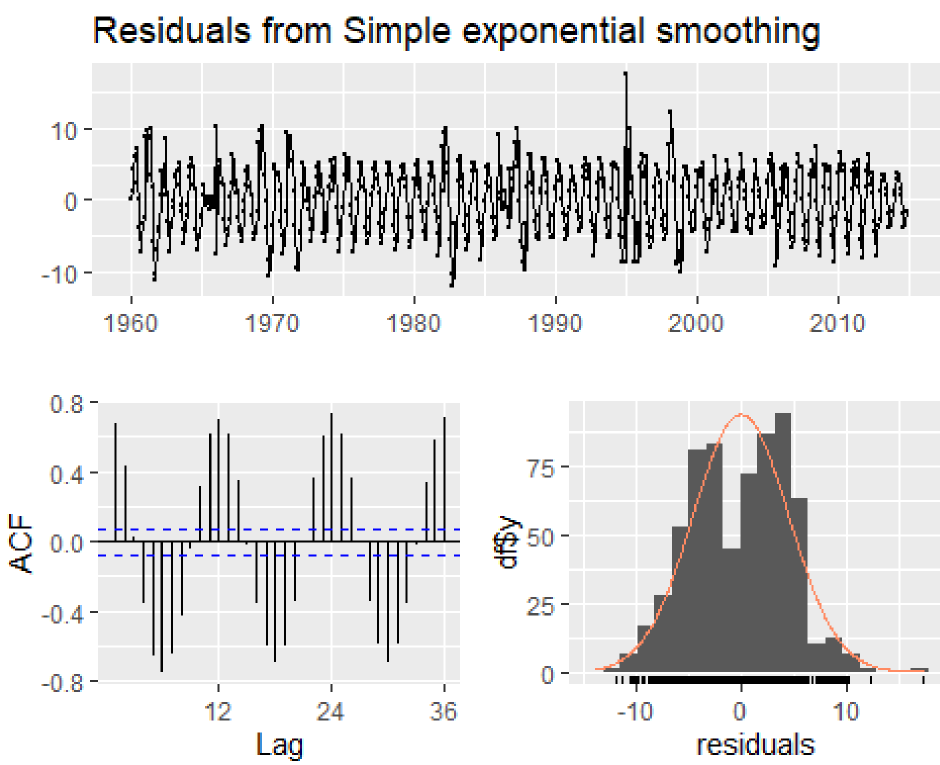


Exponential Smoothing

```
fit1_ses <- ses(solar_ts, initial="simple", h=2)
summary(fit1_ses)

##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
##  ses(y = solar_ts, h = 2, initial = "simple")
##
##   Smoothing parameters:
##     alpha = 1
##
##   Initial states:
##     l = 5.0517
##
##   sigma:  4.5688
## Error measures:
##                        ME     RMSE      MAE       MPE     MAPE     MASE
## Training set 0.0001462894 4.568777 3.876091 -5.211851 27.29823 0.636771
##                   ACF1
## Training set 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
```

```r
checkresiduals(fit1_ses)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Simple exponential smoothing
## Q* = 4227.3, df = 24, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 24
```

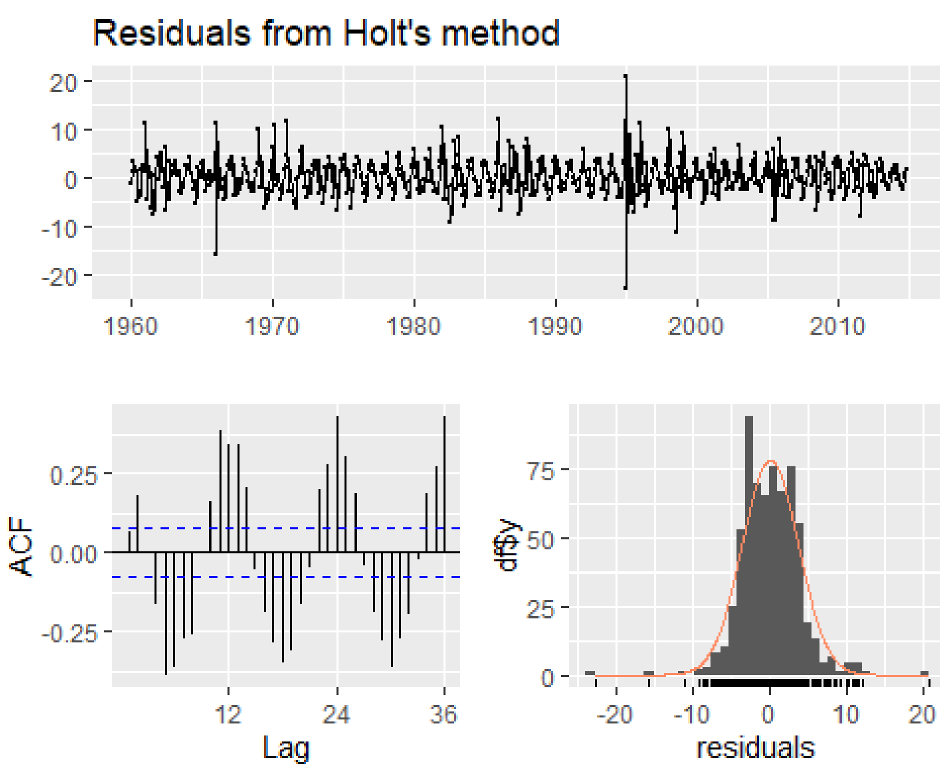
* Considering the residuals, lags are significant and the data does not follow the normal distribution and it is skewed to the right.The p-value of Ljung-Box test is less than 0.05 which means the null hypothesis of the independence of the error terms that is rejected and serial correlation is taking place. The point forecast are all the same based on exponential of nature.

```r
fit2_holt <- holt(solar_ts, initial="simple", h=2)
summary(fit2_holt)
```

```
##
## Forecast method: Holt's method
##
## Model Information:
## Holt's method
```

```
##
## Call:
##  holt(y = solar_ts, h = 2, 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
## Training set -0.004941411 3.649286 2.806374 6.556498 21.13578 0.4610361
##                    ACF1
## Training set 0.06518525
##
## Forecasts:
##          Point Forecast     Lo 80     Hi 80     Lo 95    Hi 95
## Jan 2015       3.375538 -1.301210  8.052286  -3.77693 10.52801
## Feb 2015       1.750718 -8.358924 11.860360 -13.71065 17.21208
```

```
checkresiduals(fit2_holt)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from Holt's method
## Q* = 1096.3, df = 24, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 24
```

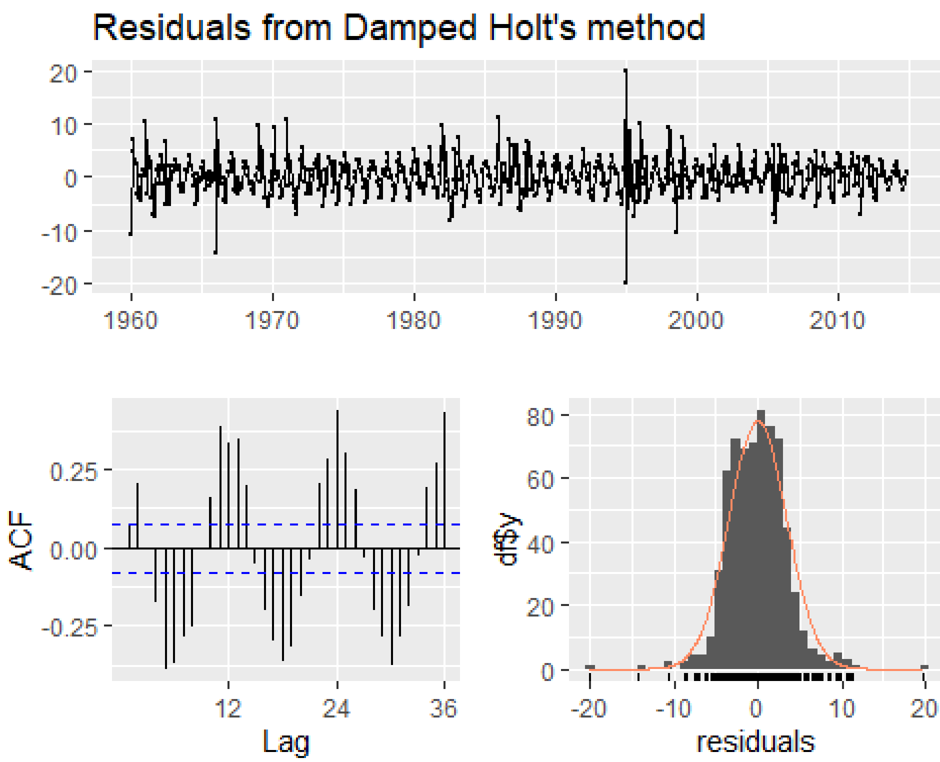
* Looking into the residuals from the Holt's method, we can infer that the series is non-stationary and the MASE is 0.461 which is lower than the fit1.ses, so we will continue to test with the holt model.

```
fit3_holt <- holt(solar_ts,seasonal="additive", damped=TRUE,
initial="simple", h=2)
summary(fit3_holt)
```

```
##
## Forecast method: Damped Holt's method
##
## Model Information:
## Damped Holt's method
##
## Call:
##  holt(y = solar_ts, h = 2, damped = TRUE, initial = "simple",
##
##  Call:
##      seasonal = "additive")
##
##   Smoothing parameters:
##     alpha = 0.9278
##     beta  = 0.9278
##     phi   = 0.8
##
##   Initial states:
##     l = 13.7194
##     b = 2.5785
##
##   sigma:  3.4657
##
##      AIC     AICc      BIC
## 5932.524 5932.652 5959.477
##
## Error measures:
##                       ME     RMSE      MAE      MPE     MAPE     MASE
## Training set -0.00831309 3.452592 2.638613 3.791232 19.25658 0.433476
##                    ACF1
## Training set 0.07287524
##
## Forecasts:
##          Point Forecast     Lo 80    Hi 80      Lo 95    Hi 95
```

```
## Jan 2015       3.774689 -0.6668422  8.21622  -3.018047 10.56742
## Feb 2015       2.735869 -5.9098357 11.38157 -10.486595 15.95833
```

```r
checkresiduals(fit3_holt)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Damped Holt's method
## Q* = 1134.4, df = 24, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 24
```

* Looking into the residuals from the Damped Holts'method, we can infer that the series is non-stationary and the MASE is 0.433 which did not change much compared to the previous model.

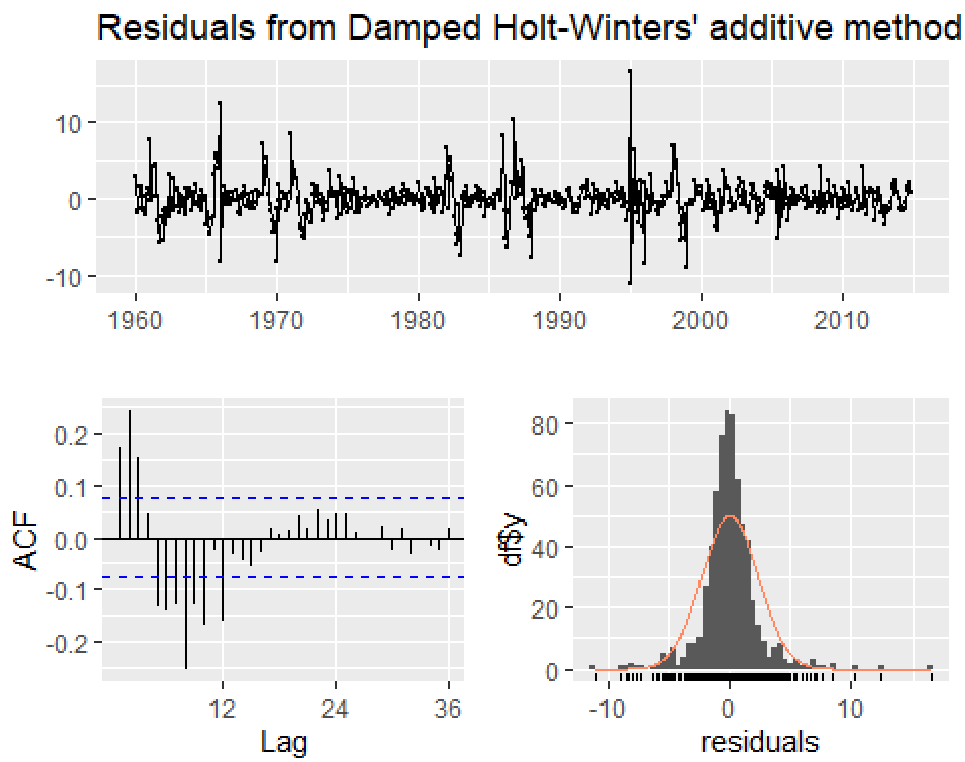
```r
fit4_hw <- hw(solar_ts,seasonal="additive",damped = TRUE,
h=2*frequency(solar_ts))
summary(fit4_hw)
```

```
##
## Forecast method: Damped Holt-Winters' additive method
##
## Model Information:
## Damped Holt-Winters' additive method
##
```

```
## Call:
##  hw(y = solar_ts, h = 2 * frequency(solar_ts), seasonal = "additive",
##
##  Call:
##      damped = TRUE)
##
##   Smoothing parameters:
##     alpha = 0.9999
##     beta  = 1e-04
##     gamma = 1e-04
##     phi   = 0.9388
##
##   Initial states:
##     l = 11.154
##     b = 0.7632
##     s = -10.4919 -8.137 -3.348 2.5794 8.08 11.1219
##            9.9586 6.9916 1.9573 -1.8565 -7.1607 -9.6946
##
##   sigma:  2.3446
##
##      AIC     AICc      BIC
## 5428.422 5429.489 5509.282
##
## Error measures:
##                        ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.01091357 2.314163 1.498521 -1.468083 12.44796 0.2461797
##                   ACF1
## Training set 0.1700724
##
## Forecasts:
##          Point Forecast      Lo 80     Hi 80       Lo 95    Hi 95
## Jan 2015       5.945198  2.940530  8.949866   1.3499548 10.54044
## Feb 2015       8.479778  4.230548 12.729009   1.9811413 14.97842
## Mar 2015      13.784309  8.579902 18.988716   5.8248548 21.74376
## Apr 2015      17.598568 11.588776 23.608360   8.4073847 26.78975
## May 2015      22.632258 15.912803 29.351713  12.3557383 32.90878
## Jun 2015      25.599150 18.238024 32.960277  14.3412786 36.85702
## Jul 2015      26.761775 18.810497 34.713053  14.6013444 38.92221
## Aug 2015      23.722251 15.221610 32.222892  10.7216429 36.72286
## Sep 2015      18.222624  9.205955 27.239293   4.4328191 32.01243
## Oct 2015      12.293259  2.788471 21.798047  -2.2430606 26.82958
## Nov 2015       7.504219 -2.464877 17.473315  -7.7421977 22.75064
## Dec 2015       5.150488 -5.262286 15.563263 -10.7744758 21.07545
## Jan 2016       5.947275 -4.891175 16.785726 -10.6287044 22.52326
## Feb 2016       8.481728 -2.766251 19.729707  -8.7205712 25.68403
## Mar 2016      13.786139  2.142988 25.429291  -4.0205242 31.59280
## Apr 2016      17.600287  5.574906 29.625668  -0.7909464 35.99152
## May 2016      22.633871 10.238009 35.029734   3.6760353 41.59171
## Jun 2016      25.600665 12.845046 38.356283   6.0926298 45.10870
## Jul 2016      26.763197 13.657667 39.868726   6.7200186 46.80637
```

```
## Aug 2016      23.723586 10.277223 37.169950   3.1591478 44.28802
## Sep 2016      18.223877  4.445085 32.002669  -2.8489663 39.29672
## Oct 2016      12.294435 -1.808972 26.397843  -9.2748652 33.86374
## Nov 2016       7.505324 -6.915414 21.926061 -14.5492911 29.55994
## Dec 2016       5.151525 -9.579726 19.882776 -17.3779790 27.68103
```

```r
checkresiduals(fit4_hw)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Damped Holt-Winters' additive method
## Q* = 210.76, df = 24, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 24
```

* Observing the seasonal additive damped model, it can be inferred that the series is non-stationary base on the p-value of Ljung-Box test. MASE is 0.246 which was 0.433 in the previous model thus a significant improvement. Residuals from Damped Holt-Winters' Additive method, shows presence of non-randomness and ACF plot showing significant lags that demonstrates the presence of serial correlation. The histogram is normally distributed.

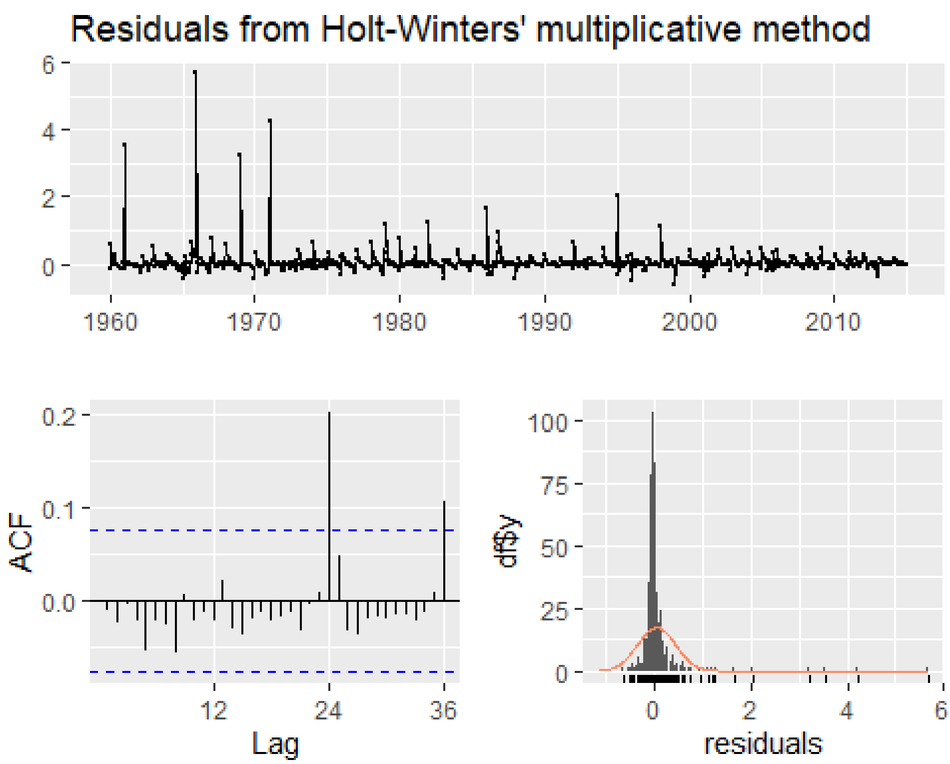
```r
fit5_hw <- hw(solar_ts,seasonal="multiplicative", h=2*frequency(solar_ts))
summary(fit5_hw)
```

```
##
## Forecast method: Holt-Winters' multiplicative method
```

```
##
## Model Information:
## Holt-Winters' multiplicative method
##
## Call:
##  hw(y = solar_ts, h = 2 * frequency(solar_ts), seasonal =
"multiplicative")
##
##   Smoothing parameters:
##     alpha = 0.8251
##     beta  = 1e-04
##     gamma = 0.0259
##
##   Initial states:
##     l = 10.5659
##     b = -0.2371
##     s = 0.5024 0.6292 0.9319 1.2229 1.4923 1.6309
##            1.5606 1.3672 1.0278 0.8128 0.5106 0.3114
##
##   sigma:  0.3971
##
##      AIC     AICc      BIC
## 6648.746 6649.699 6725.114
##
## Error measures:
##                      ME    RMSE      MAE        MPE     MAPE      MASE
## Training set 0.04281136 2.12539 1.359297 -0.4896895 10.87401 0.2233077
##                    ACF1
## Training set 0.06551473
##
## Forecasts:
##          Point Forecast         Lo 80     Hi 80       Lo 95     Hi 95
## Jan 2015       5.610335   2.75534413  8.465326   1.2440033  9.976666
## Feb 2015       6.512806   2.03809262 10.987520  -0.3306776 13.356290
## Mar 2015       8.786120   1.33667747 16.235562  -2.6068190 20.179058
## Apr 2015      10.192785  -0.02672876 20.412298  -5.4366123 25.822181
## May 2015      12.512597  -1.96157245 26.986766  -9.6237347 34.648928
## Jun 2015      14.061609  -4.41048190 32.533701 -14.1890164 42.312235
## Jul 2015      14.502088  -6.89909775 35.903274 -18.2282011 47.232377
## Aug 2015      12.945620  -8.35024040 34.241481 -19.6235881 45.514829
## Sep 2015      10.352210  -8.52365174 29.228072 -18.5159294 39.220350
## Oct 2015       7.418279  -7.51119146 22.347750 -15.4143760 30.250935
## Nov 2015       4.989730  -6.05900058 16.038460 -11.9078451 21.887305
## Dec 2015       3.871776  -5.53876625 13.282319 -10.5204066 18.263960
## Jan 2016       4.199400  -7.01753622 15.416336 -12.9554236 21.354224
## Feb 2016       4.838892  -9.30305408 18.980837 -16.7893479 26.467131
## Mar 2016       6.477174 -14.21845674 27.172805 -25.1740619 38.128410
## Apr 2016       7.452633 -18.56883703 33.474103 -32.3437711 47.249037
## May 2016       9.069749 -25.52978297 43.669280 -43.8456686 61.985166
## Jun 2016      10.099485 -31.99838423 52.197355 -54.2836502 74.482621
```

```
## Jul 2016      10.315199 -36.68017471 57.310572 -61.5580226 82.188420
## Aug 2016       9.113765 -36.29212001 54.519651 -60.3285438 78.556075
## Sep 2016       7.208700 -32.09306283 46.510463 -52.8981594 67.315559
## Oct 2016       5.105869 -25.38360449 35.595343 -41.5237568 51.735495
## Nov 2016       3.391945 -18.81647619 25.600367 -30.5729044 37.356795
## Dec 2016       2.597256 -16.07148862 21.266001 -25.9541251 31.148637
```

```
checkresiduals(fit5_hw)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Holt-Winters' multiplicative method
## Q* = 38.585, df = 24, p-value = 0.03017
##
## Model df: 0.   Total lags used: 24
```

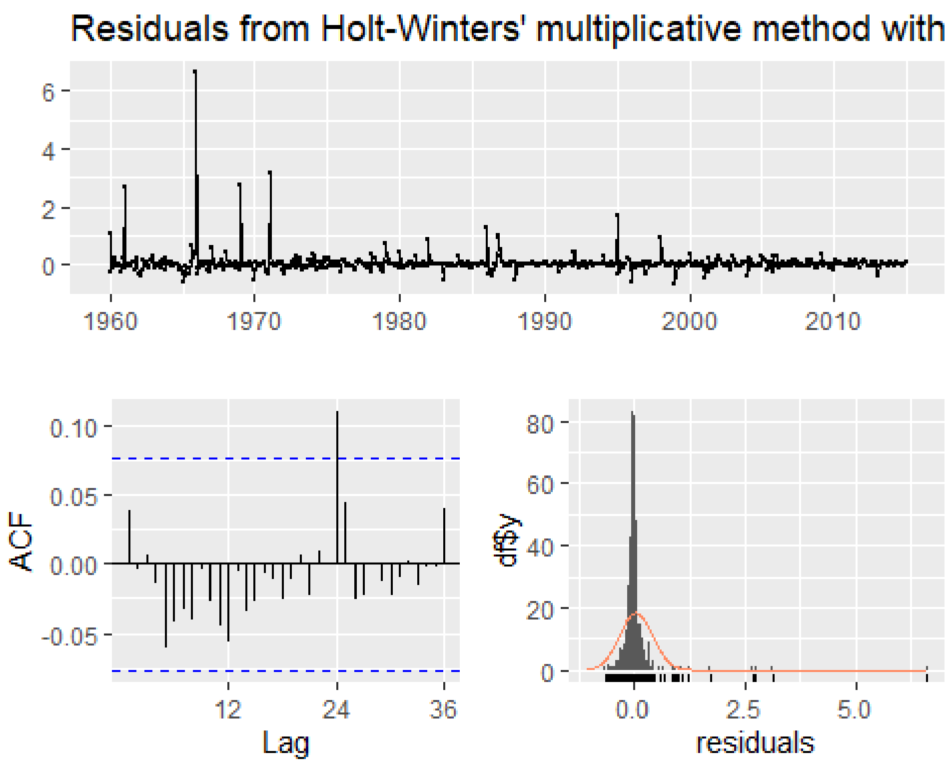
 * Observing the Holt Winter's multiplicative method, we can infer that the series is non-stationary base on the p-value of Ljung-Box test. MASE is 0.223 which was 0.246 in the previous model thus a significant improvement.Residuals from Holt Winter's multiplicative method, shows presence of less randomness from the previous model and ACF plot showing 2 significant lags that demonstrates the presence of serial correlation but a much improved model compared to the previous ones. The histogram is not normally distributed.

```
fit6_hw <- hw(solar_ts,seasonal="multiplicative",exponential = TRUE,
h=2*frequency(solar_ts))
summary(fit6_hw)
```

```
##
## Forecast method: Holt-Winters' multiplicative method with exponential
trend
##
## Model Information:
## Holt-Winters' multiplicative method with exponential trend
##
## Call:
##  hw(y = solar_ts, h = 2 * frequency(solar_ts), seasonal =
"multiplicative",
##
##  Call:
##      exponential = TRUE)
##
##   Smoothing parameters:
##     alpha = 0.6499
##     beta  = 1e-04
##     gamma = 0.0597
##
##   Initial states:
##     l = 9.3759
##     b = 0.9889
##     s = 0.3552 0.4789 0.952 1.0553 1.4608 1.7926
##            1.6746 1.4529 1.1145 0.8672 0.5333 0.2627
##
##   sigma:  0.3755
##
##      AIC     AICc      BIC
## 6584.208 6585.161 6660.576
##
## Error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MASE
ACF1
## Training set 0.06388152 2.208727 1.412454 -1.303792 11.28449 0.2320404
0.153871
##
## Forecasts:
##          Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
## Jan 2015       5.797870 2.8852976  8.544780 1.3373003  9.993484
## Feb 2015       7.513359 3.3885755 12.092392 1.6178697 14.895076
## Mar 2015      10.758212 4.2394234 18.179044 2.1621012 24.012712
## Apr 2015      12.694087 4.4380272 22.815905 2.2063044 30.883870
## May 2015      15.340074 4.9506809 28.830224 2.5743273 41.596602
## Jun 2015      17.239431 5.0535263 32.879690 2.4716554 49.560191
## Jul 2015      18.004700 4.7985240 36.594145 2.3473627 55.726753
## Aug 2015      16.206596 3.9790177 32.848007 1.8042339 51.946987
## Sep 2015      13.011256 2.8556627 27.194447 1.3080383 43.608989
## Oct 2015       9.215869 1.8858387 19.274631 0.8191275 33.538207
## Nov 2015       6.214762 1.1652839 13.378701 0.4942809 24.224815
## Dec 2015       4.565667 0.7903985 10.169511 0.3460751 18.052769
```

```
## Jan 2016       5.220661 0.8293840 11.698323 0.3190152 21.507416
## Feb 2016       6.765364 1.0078247 15.234698 0.3821093 28.736007
## Mar 2016       9.687175 1.3147214 22.553160 0.4717281 43.876651
## Apr 2016      11.430324 1.4594613 27.417186 0.5660426 52.507448
## May 2016      13.812889 1.6019284 32.613583 0.5884780 66.535502
## Jun 2016      15.523155 1.6774780 37.386142 0.6133504 73.870011
## Jul 2016      16.212237 1.6697776 38.401838 0.6086061 80.094850
## Aug 2016      14.593143 1.3722738 34.909740 0.5118141 70.281633
## Sep 2016      11.715916 1.0475750 29.141882 0.3702521 60.831682
## Oct 2016       8.298381 0.7067339 19.206006 0.2479080 43.555785
## Nov 2016       5.596049 0.4377307 13.407517 0.1395887 31.459929
## Dec 2016       4.111130 0.3017424  9.660859 0.1069407 24.559981
```

```
checkresiduals(fit6_hw)
```



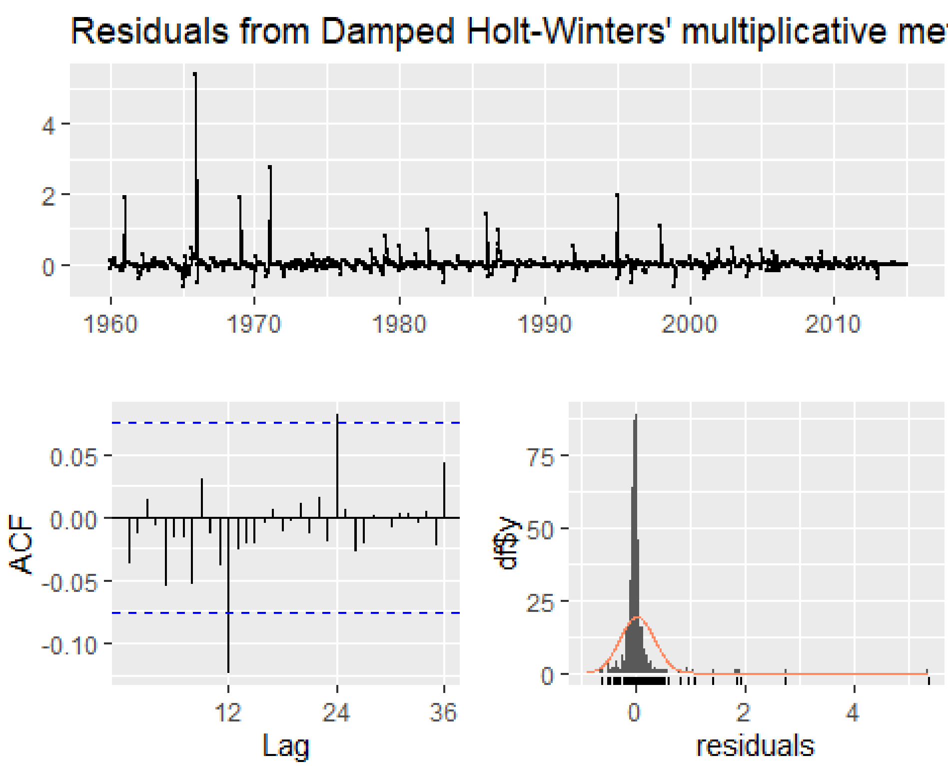

```
##
##  Ljung-Box test
##
## data:  Residuals from Holt-Winters' multiplicative method with exponential
trend
## Q* = 21.246, df = 24, p-value = 0.6242
##
## Model df: 0.   Total lags used: 24
```

* Observing the Holt Winter's multiplicative method with exponential trend, we can infer that the series is stationary based on the p-value of Ljung-Box test. MASE is 0.232 a slight increase from the previous model thus a significant improvement in terms of stationarity.

Residuals shows presence of less randomness from the previous model and ACF plot showing 1 significant lags that demonstrates the presence of serial correlation but a much improved model compared to the previous ones. The histogram is not normally distributed.

```
fit7_hw <- hw(solar_ts,seasonal="multiplicative",exponential = TRUE,damped =
TRUE,h=2*frequency(solar_ts))
summary(fit7_hw)

##
## Forecast method: Damped Holt-Winters' multiplicative method with
exponential trend
##
## Model Information:
## Damped Holt-Winters' multiplicative method with exponential trend
##
## Call:
##  hw(y = solar_ts, h = 2 * frequency(solar_ts), seasonal =
"multiplicative",
##
##  Call:
##      damped = TRUE, exponential = TRUE)
##
##   Smoothing parameters:
##     alpha = 0.8468
##     beta  = 1e-04
##     gamma = 0.0111
##     phi   = 0.8363
##
##   Initial states:
##     l = 9.3778
##     b = 1.1479
##     s = 0.4388 0.5653 0.8294 1.1487 1.449 1.6183
##            1.5604 1.3968 1.1149 0.8758 0.5804 0.4221
##
##   sigma:  0.3143
##
##      AIC     AICc      BIC
## 6365.329 6366.396 6446.190
##
## Error measures:
##                       ME     RMSE      MAE       MPE    MAPE      MASE
## Training set -0.03713562 2.041516 1.236905 -2.113647 9.98074 0.2032009
##                      ACF1
## Training set -0.003606111
##
## Forecasts:
##          Point Forecast     Lo 80     Hi 80      Lo 95     Hi 95
## Jan 2015       5.562434 3.3825844  7.852082 2.27341405  9.106885
## Feb 2015       7.022873 3.5360437 11.076511 2.15972549 13.683763
## Mar 2015      10.401147 4.4391155 17.608554 2.51340621 22.660431
```

```
## Apr 2015      12.948609 4.8768879 23.379129 2.55642162 31.370057
## May 2015      16.301555 5.1366419 31.031209 2.75349074 43.923377
## Jun 2015      18.277245 5.0816752 35.745207 2.51788745 52.699227
## Jul 2015      18.904626 4.6152460 37.631096 2.37089122 58.931853
## Aug 2015      16.978636 3.7913972 35.143734 1.87291368 55.840425
## Sep 2015      13.530464 2.7535249 28.267905 1.27260167 48.575258
## Oct 2015       9.679038 1.7617791 20.953408 0.80433700 36.849379
## Nov 2015       6.541756 1.0952479 14.437987 0.45409201 25.230611
## Dec 2015       5.166293 0.7583887 11.930694 0.31040274 21.272747
## Jan 2016       5.562371 0.7457058 12.492877 0.30108687 24.599156
## Feb 2016       7.022805 0.8873454 15.815293 0.33569079 31.768844
## Mar 2016      10.401064 1.2020173 24.149316 0.47070494 48.687659
## Apr 2016      12.948522 1.3482758 30.218589 0.49659033 61.873715
## May 2016      16.301464 1.5649454 37.705691 0.54846198 83.041858
## Jun 2016      18.277160 1.6014590 42.020050 0.54589633 94.943787
## Jul 2016      18.904552 1.5118464 44.460175 0.50458429 98.981697
## Aug 2016      16.978581 1.2856457 40.460942 0.42365805 97.300782
## Sep 2016      13.530427 0.9300846 32.353154 0.30280964 80.387441
## Oct 2016       9.679016 0.5919862 23.077282 0.19328468 57.322041
## Nov 2016       6.541743 0.3784627 15.451995 0.12360043 39.387378
## Dec 2016       5.166285 0.2829269 12.505903 0.08528641 30.255753

checkresiduals(fit7_hw)
```




```
##
##  Ljung-Box test
##
```

```
## data:  Residuals from Damped Holt-Winters' multiplicative method with
exponential trend
## Q* = 23.862, df = 24, p-value = 0.4695
##
## Model df: 0.   Total lags used: 24
```

* Examining the Damped Holt Winter's multiplicative method with an exponential trend, it becomes apparent that the series is stationary, as indicated by the p-value of the Ljung-Box test. The MASE value, which is 0.20, shows a slight decrease compared to the previous model, suggesting a modest improvement in terms of stationarity and MASE. However, similar to the previous model, the residuals still exhibit some non-random patterns, and the ACF plot indicates the presence of serial correlation with two significant lags. Overall, this model doesn't represent a substantial improvement in addressing serial correlation compared to the previous ones. Additionally, it's worth noting that the histogram does not follow a normal distribution.

```
mase_hw_1 <- accuracy(fit1_ses)[,"MASE"]
mase_hw_2 <- accuracy(fit2_holt)[,"MASE"]
mase_hw_3 <- accuracy(fit3_holt)[,"MASE"]
mase_hw_4 <- accuracy(fit4_hw)[,"MASE"]
mase_hw_5 <- accuracy(fit5_hw)[,"MASE"]
mase_hw_6 <- accuracy(fit6_hw)[,"MASE"]
mase_hw_7 <- accuracy(fit7_hw)[,"MASE"]
mase_hw_df <- data.frame(Model =
c("fit1_ses","fit2_holt","fit3_holt","fit4_hw","fit5_hw","fit6_hw",
"fit7_hw"),
                      MASE =
c(mase_hw_3,mase_hw_3,mase_hw_3,mase_hw_4,mase_hw_5,mase_hw_6,mase_hw_7))
mase_hw_df

##       Model      MASE
## 1  fit1_ses 0.4334760
## 2 fit2_holt 0.4334760
## 3 fit3_holt 0.4334760
## 4   fit4_hw 0.2461797
## 5   fit5_hw 0.2233077
## 6   fit6_hw 0.2320404
## 7   fit7_hw 0.2032009
```

- After evaluating the Mean Absolute Scaled Error (MASE) values for all the exponential smoothing models, it is evident that the "fit7_hw" model stands out with the lowest MASE value of 0.203. Furthermore, the p-value suggests that the series demonstrates stationarity. Based on these findings, it is reasonable to conclude that the "fit7_hw" model is likely to offer more precise forecasts for the solar series data.

```
variable <- c("AAA", "MAA", "MAM", "MMM")
damped <- c(TRUE, FALSE)
models <- expand.grid(variable, damped)
aic <- array(NA,8)
```

```r
bic <- array(NA,8)
mase <- array(NA,8)
new_model<- array(NA,dim =c(8,2))
for(i in 1:8){
  ets_1<- ets(solar_ts, model = toString(models[i,1]), damped = models[i,2])
  aic[i] <- ets_1$aic
  bic[i] <- ets_1$bic
  mase[i] <- accuracy(ets_1)[6]
  new_model[i,1] <- toString(models[i,1])
  new_model[i,2] <- models[i,2]
}

new_1 <- data.frame(new_model, mase, aic,bic)
new_1$X2 <- factor(new_1$X2, levels = c(T,F), labels = c("Damped","N"))
new_1 <- unite(new_1, "ETS_Model", c("X1","X2"))
colnames(new_1) <- c("ETS_Model", "MASE", "AIC", "BIC")

accuracy <- arrange(new_1, MASE)

accuracy

##    ETS_Model      MASE      AIC      BIC
## 1 AAA_Damped 0.2461797 5428.422 5509.282
## 2      AAA_N 0.2471600 5434.708 5511.076
## 3 MMM_Damped 0.3201193 5995.550 6076.410
## 4 MAM_Damped 0.3222574 5953.502 6034.363
## 5      MAM_N 0.3721664 6105.959 6182.327
## 6 MAA_Damped 0.3798095 6469.079 6549.940
## 7      MAA_N 0.4748561 7602.755 7679.123
## 8      MMM_N 0.5292151 6670.168 6746.536
```

* Comparing the MASE values with Hotel Winters' MASE values, it does not demonstrate lower MASE values. Therefore we will look into the auto ETS values.

```r
new_1 <- ets(solar_ts,model = "ANN")
new_2 <- ets(solar_ts,model = "AAN")
new_3 <- ets(solar_ts,model = "AAN", damped = TRUE)
new_4 <- ets(solar_ts,model = "AAA")
new_5 <- ets(solar_ts,model = "AAA", damped = TRUE)
new_6 <- ets(solar_ts, model="MNN", beta = 0.0001)
new_7 <- ets(solar_ts, model="MAN", damped = TRUE)
summary(new_1)

## ETS(A,N,N)
##
## Call:
##  ets(y = solar_ts, model = "ANN")
##
##   Smoothing parameters:
##     alpha = 0.9999
```

```
##
##   Initial states:
##     l = 5.0575
##
##   sigma:  4.576
##
##      AIC     AICc      BIC
## 6296.371 6296.407 6309.847
##
## Training set error measures:
##                        ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 0.0001378357 4.569082 3.876391 -5.213129 27.30052 0.6368203
##                   ACF1
## Training set 0.6678374
```

```
summary(new_2)
```

```
## ETS(A,Ad,N)
##
## Call:
##  ets(y = solar_ts, model = "AAN")
##
##   Smoothing parameters:
##     alpha = 0.9279
##     beta  = 0.9279
##     phi   = 0.8
##
##   Initial states:
##     l = 13.7196
##     b = 2.5786
##
##   sigma:  3.4657
##
##      AIC     AICc      BIC
## 5932.524 5932.653 5959.478
##
## Training set error measures:
##                        ME     RMSE      MAE      MPE     MAPE      MASE
## Training set -0.008312543 3.452593 2.638571 3.790894 19.25631 0.4334691
##                    ACF1
## Training set 0.07272876
```

```
summary(new_3)
```

```
## ETS(A,Ad,N)
##
## Call:
##  ets(y = solar_ts, model = "AAN", damped = TRUE)
##
##   Smoothing parameters:
##     alpha = 0.9279
```

```
##     beta  = 0.9279
##     phi   = 0.8
##
##   Initial states:
##     l = 13.7196
##     b = 2.5786
##
##   sigma:  3.4657
##
##      AIC     AICc      BIC
## 5932.524 5932.653 5959.478
##
## Training set error measures:
##                        ME     RMSE      MAE      MPE     MAPE      MASE
## Training set -0.008312543 3.452593 2.638571 3.790894 19.25631 0.4334691
##                    ACF1
## Training set 0.07272876
```

```
summary(new_4)
```

```
## ETS(A,Ad,A)
##
## Call:
##  ets(y = solar_ts, model = "AAA")
##
##   Smoothing parameters:
##     alpha = 0.9999
##     beta  = 1e-04
##     gamma = 1e-04
##     phi   = 0.9388
##
##   Initial states:
##     l = 11.154
##     b = 0.7632
##     s = -10.4919 -8.137 -3.348 2.5794 8.08 11.1219
##            9.9586 6.9916 1.9573 -1.8565 -7.1607 -9.6946
##
##   sigma:  2.3446
##
##      AIC     AICc      BIC
## 5428.422 5429.489 5509.282
##
## Training set error measures:
##                       ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.01091357 2.314163 1.498521 -1.468083 12.44796 0.2461797
##                   ACF1
## Training set 0.1700724
```

```
summary(new_5)
```

```
## ETS(A,Ad,A)
##
## Call:
##  ets(y = solar_ts, model = "AAA", damped = TRUE)
##
##   Smoothing parameters:
##     alpha = 0.9999
##     beta  = 1e-04
##     gamma = 1e-04
##     phi   = 0.9388
##
##   Initial states:
##     l = 11.154
##     b = 0.7632
##     s = -10.4919 -8.137 -3.348 2.5794 8.08 11.1219
##            9.9586 6.9916 1.9573 -1.8565 -7.1607 -9.6946
##
##   sigma:  2.3446
##
##      AIC     AICc      BIC
## 5428.422 5429.489 5509.282
##
## Training set error measures:
##                       ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.01091357 2.314163 1.498521 -1.468083 12.44796 0.2461797
##                   ACF1
## Training set 0.1700724
```

```
summary(new_6)
```

```
## ETS(M,N,N)
##
## Call:
##  ets(y = solar_ts, model = "MNN", beta = 1e-04)
##
##   Smoothing parameters:
##     alpha = 0.9999
##
##   Initial states:
##     l = 4.4856
##
##   sigma:  0.3871
##
##      AIC     AICc      BIC
## 6619.776 6619.812 6633.253
##
## Training set error measures:
##                       ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 0.001004352 4.569136 3.877241 -5.195978 27.31733 0.6369599
```

```
##                      ACF1
## Training set 0.6678785
```

```
summary(new_7)
```

```
## ETS(M,Ad,N)
##
## Call:
##  ets(y = solar_ts, model = "MAN", damped = TRUE)
##
##   Smoothing parameters:
##     alpha = 0.9815
##     beta  = 1e-04
##     phi   = 0.98
##
##   Initial states:
##     l = 9.9304
##     b = 5.8356
##
##   sigma:  0.35
##
##      AIC     AICc      BIC
## 6540.866 6540.995 6567.819
##
## Training set error measures:
##                        ME     RMSE       MAE       MPE      MAPE       MASE
ACF1
## Training set -0.4464292 4.75387 4.007741 -9.78188 29.60799 0.6583987
0.6870915
```

- The initial time series plot makes it evident that the Linear anmd the Non Linear State Space Models and does not exhibit multiple trends or changing variance. The presence of seasonality is clearly observable in the time series plot. As a result, we can conclude that there is no evidence of volatility clustering in this series. Consequently, we anticipate that the ANN model might outperform the multiplicative error, multiplicative seasonality, MNN model in this scenario.

```
auto_ets <- ets(solar_ts)
```

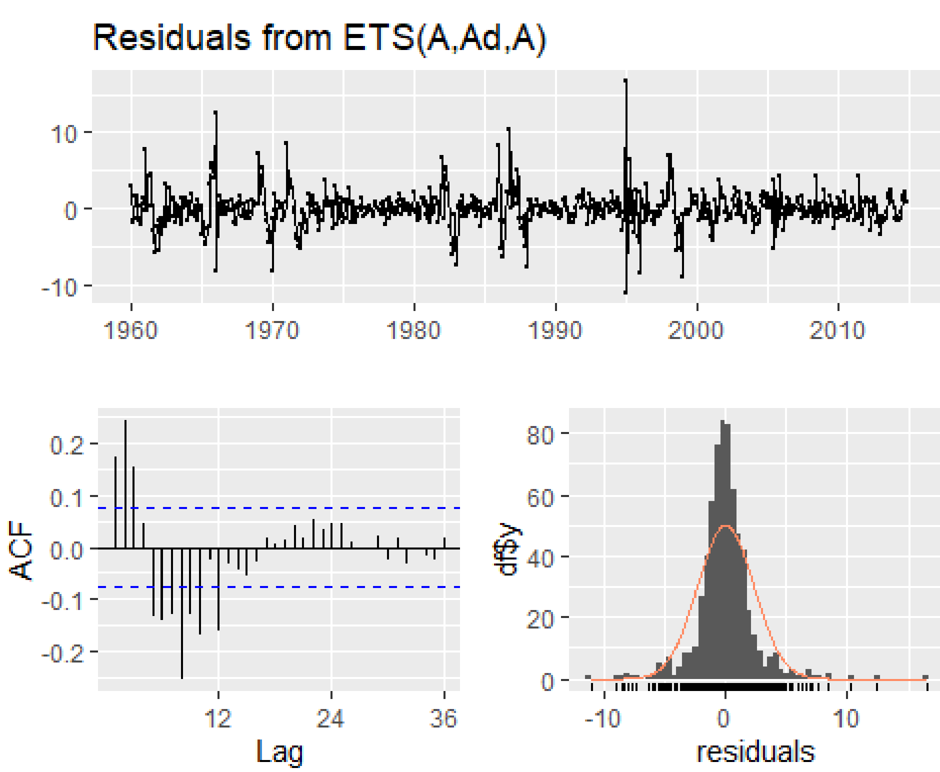
- The automated model fitting process is recommending the use of ETS(A,Ad,A), which signifies a model comprising an additive error component, an additive trend component, and an additive seasonal component. This recommendation is made because, in this configuration, the specific parameter values for smoothing and seasonality factors should be estimated or determined based on the unique characteristics of the time series data under analysis.

```
summary(auto_ets)
```

```
## ETS(A,Ad,A)
##
## Call:
##  ets(y = solar_ts)
```

```
##
##   Smoothing parameters:
##     alpha = 0.9999
##     beta  = 1e-04
##     gamma = 1e-04
##     phi   = 0.9388
##
##   Initial states:
##     l = 11.154
##     b = 0.7632
##     s = -10.4919 -8.137 -3.348 2.5794 8.08 11.1219
##            9.9586 6.9916 1.9573 -1.8565 -7.1607 -9.6946
##
##   sigma:  2.3446
##
##      AIC     AICc      BIC
## 5428.422 5429.489 5509.282
##
## Training set error measures:
##                       ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.01091357 2.314163 1.498521 -1.468083 12.44796 0.2461797
##                   ACF1
## Training set 0.1700724
```

```
checkresiduals(auto_ets)
```

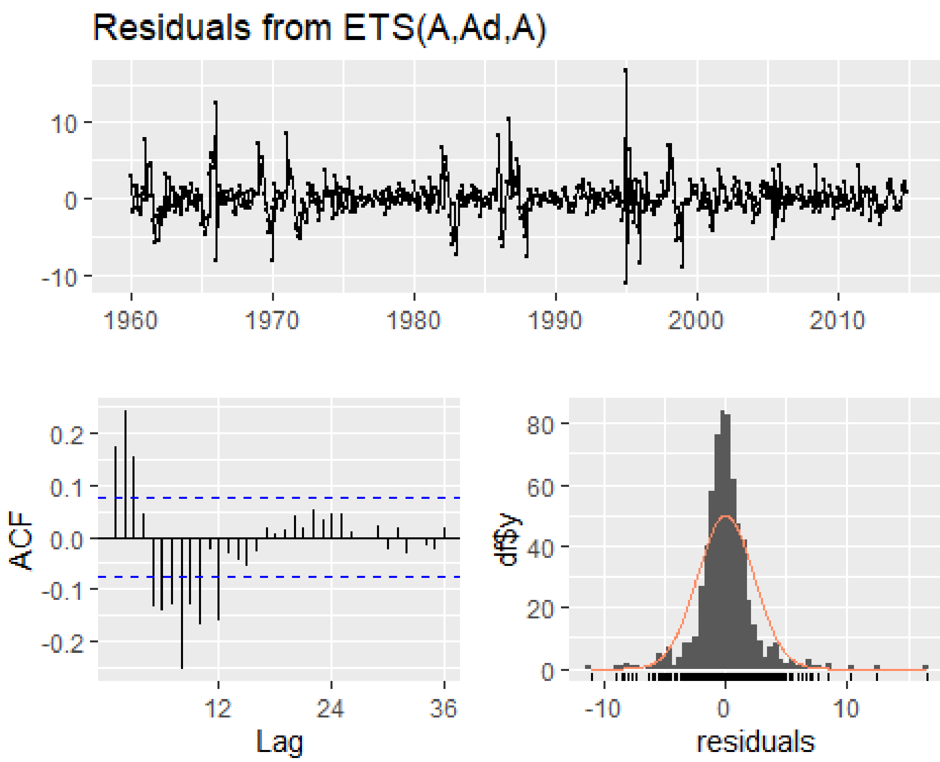


```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,Ad,A)
## Q* = 210.76, df = 24, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 24
```

```r
fit.AAdA = ets(solar_ts, model = "AAA", damped = TRUE)
summary(fit.AAdA)
```

```
## ETS(A,Ad,A)
##
## Call:
##  ets(y = solar_ts, model = "AAA", damped = TRUE)
##
##   Smoothing parameters:
##     alpha = 0.9999
##     beta  = 1e-04
##     gamma = 1e-04
##     phi   = 0.9388
##
##   Initial states:
##     l = 11.154
##     b = 0.7632
##     s = -10.4919 -8.137 -3.348 2.5794 8.08 11.1219
##            9.9586 6.9916 1.9573 -1.8565 -7.1607 -9.6946
##
##   sigma:  2.3446
##
##      AIC     AICc      BIC
## 5428.422 5429.489 5509.282
##
## Training set error measures:
##                         ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.01091357 2.314163 1.498521 -1.468083 12.44796 0.2461797
##                   ACF1
## Training set 0.1700724
```

```r
checkresiduals(fit.AAdA)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,Ad,A)
## Q* = 210.76, df = 24, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 24
```

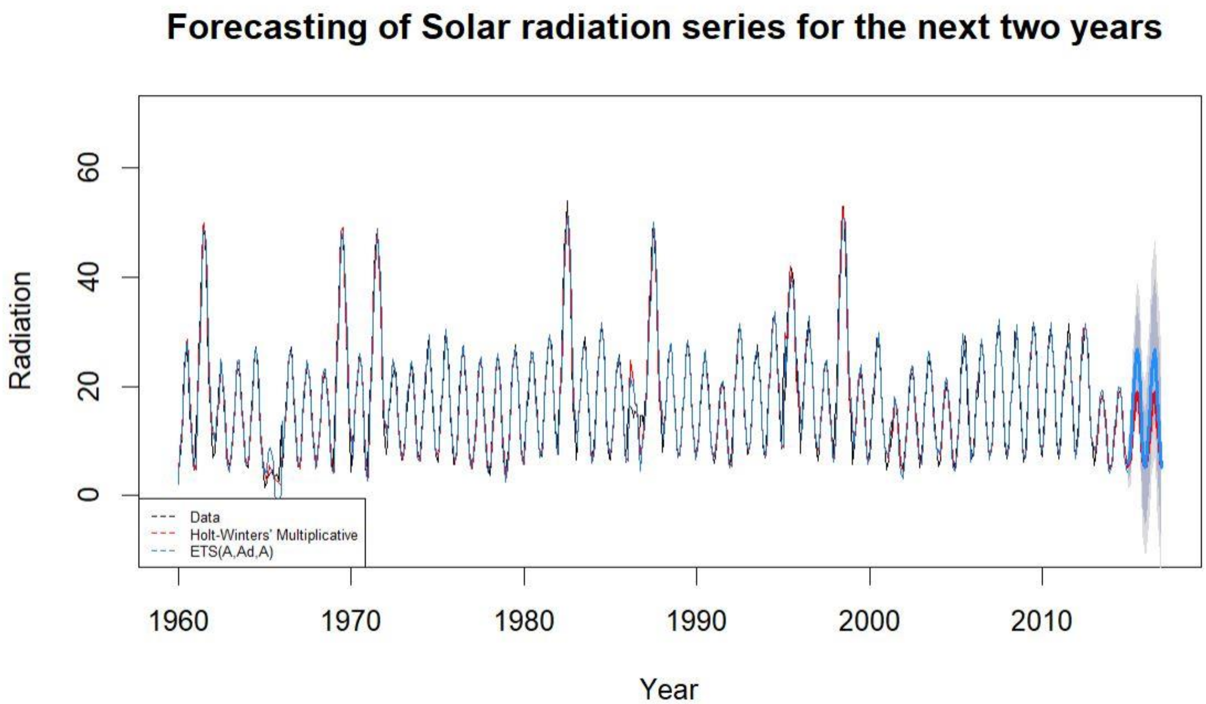
 * Observing the summary statistics, it can be demonstrated that series is non-stationary but has the lowest MASE values among all the state space models. The residuals show signs of more randomness compared to the previous ones, ACF plot shows presence of significant lags that demonstrates the presence of serial correlation and some seasonal effects. The histogram is normally distributed.

- When comparing all the different models, the model that minimizes MASE is Multiplicative Exponential Damped Model and the best state space model is the ETS(A,Ad,A)

Forecasting

```
frc_fit3 <- forecast(fit.AAdA)

plot(frc_fit3, fcol = "white", main = "Forecasting of Solar radiation series for the next two years ",
ylab = "Radiation", xlab="Year", ylim = c(-10,70))

lines(fitted(fit7_hw), col = "red")
```

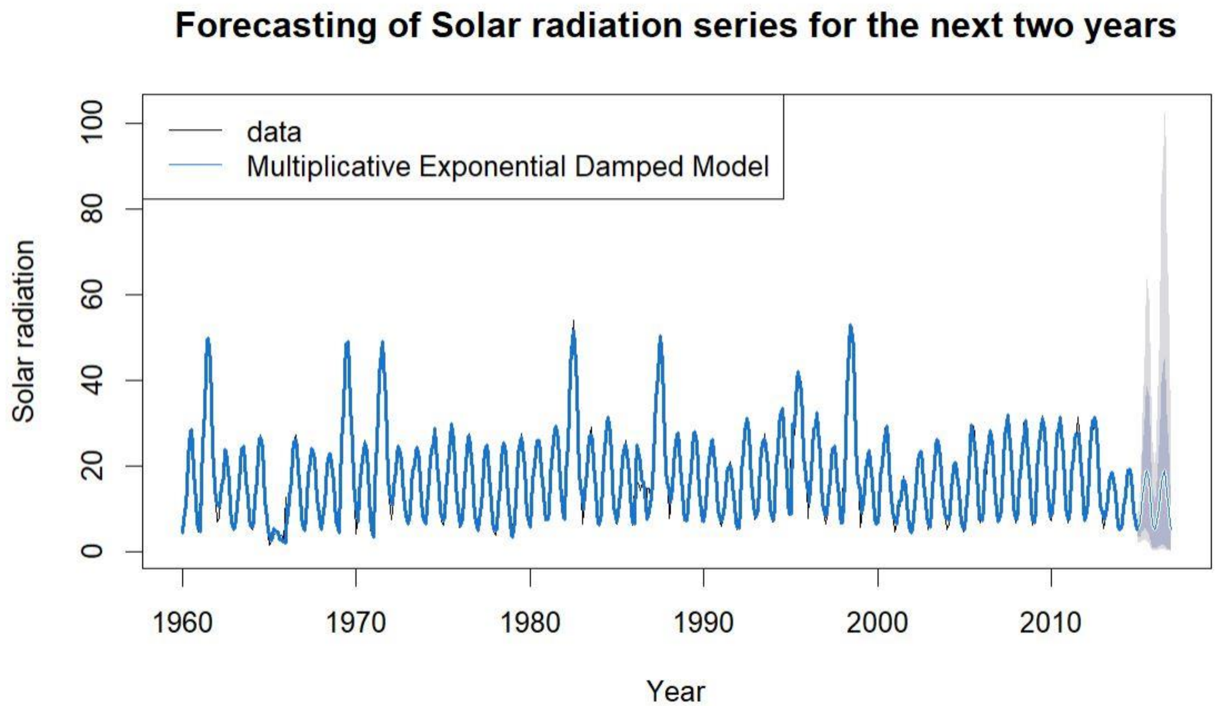
```
lines(fit7_hw$mean, col = "red", lwd = 2)

lines(fitted(fit.AAdA), col = "dodgerblue3")

lines(frc_fit3$mean, col = "dodgerblue", lwd = 2)

legend("bottomleft", lty = 2, col = c("black", "red", "dodgerblue3"), c("Data", "Holt-Winters'
Multiplicative", "ETS(A,Ad,A)"), , cex = 0.5)
```



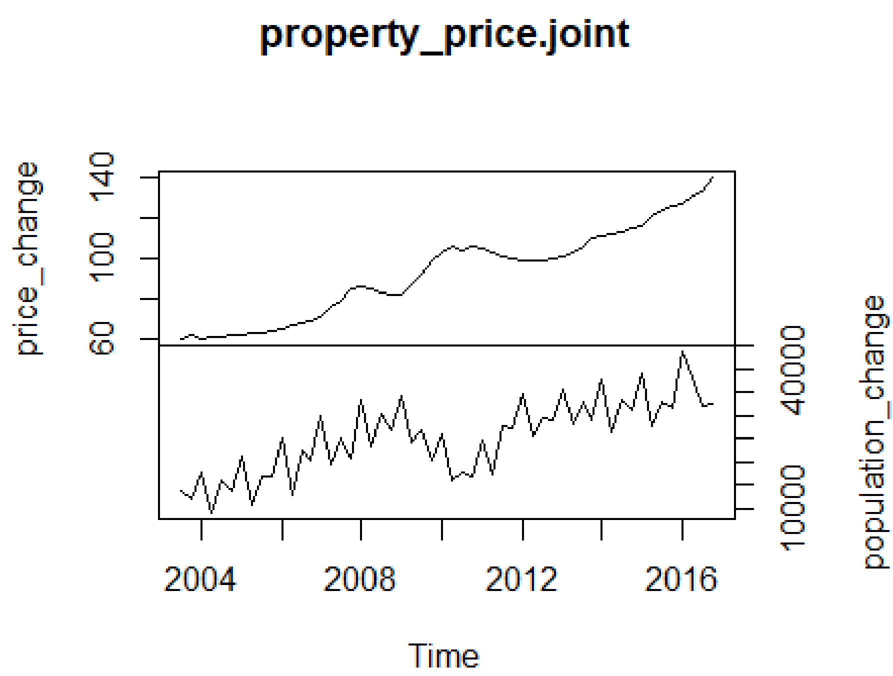
* The models are showing similar trends when compared with the fitted values. The Multiplicative exponential Damped model seemed to better fit the original data and the state space model seems to have higher deviation. Comparing to the original data, the multiplicative model suggest decrease in in temperature but the state space model suggest increase in temperature.

- Multiplicative Model has the lower Mase values in comparison, therefore, we are considering multiplicative model to predict the next two years forecast for solar radiation reaching the ground at a particular location over the globe.

```
plot(fit7_hw, type="l", main = "Forecasting of Solar radiation series for the
next two years", ylab="Solar radiation", xlab="Year", fcol="white",
plot.conf=FALSE)
lines(fit7_hw$mean, col="red", type="l")
lines(fitted(fit7_hw), col="red", lwd=2)
legend("topleft", lty=1, col=c("black","red"),
c("data","Multiplicative Exponential Damped Model"))
```

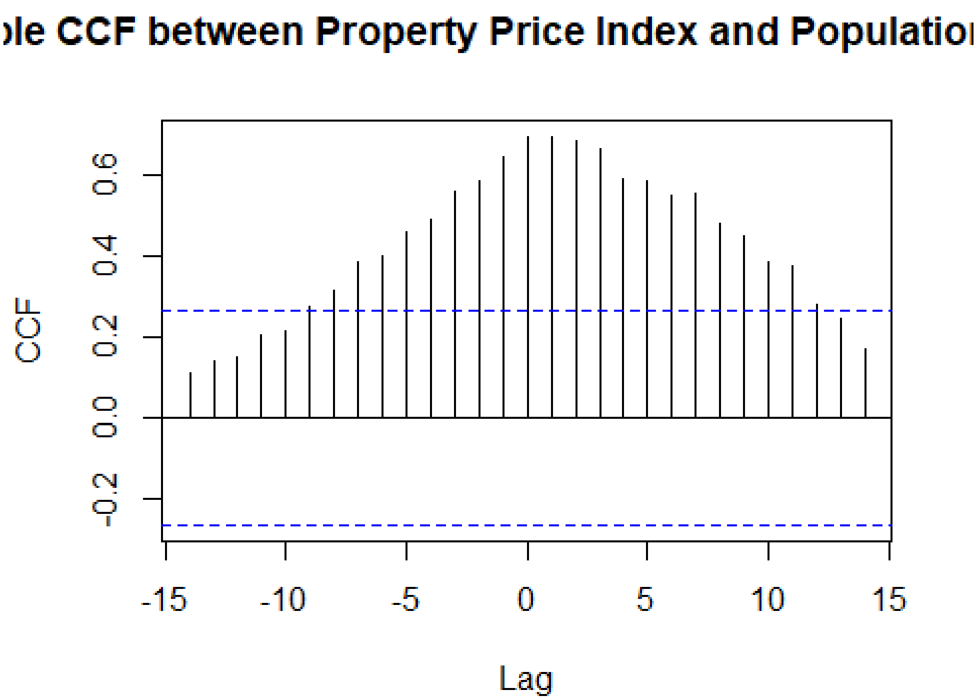



```r
getwd()

## [1] "F:/2nd semester/Forecasting"

setwd("F:/2nd semester/Forecasting/Assignment_2")
property_price <- read.csv("data2.csv")
property_price <- ts(property_price, start = c(2003,3), frequency = 4)
price_change <- ts(property_price[,2], start = c(2003,3), frequency = 4)
population_change <- ts(property_price[,3], start = c(2003,3), frequency = 4)
property_price.joint<- ts.intersect(price_change,population_change)
plot(property_price.joint,yax.flip = T)
```



- The time series plot reveals several noteworthy patterns. Firstly, there is a noticeable simultaneous increase in both series over time, indicating a consistent upward trend in both. The visualization strongly implies a positive correlation between the two series, as they exhibit synchronized upward trends and potential seasonality patterns that coincide over the observed time period. This suggests that increase in population in victoria might be linked to the observed increase in Residential Property Price Index in Melbourne, potentially indicating a causal relationship or shared underlying factors between the two series. This give some evidence of spurious correlation.

```
ccf(as.vector(property_price.joint[,1]), as.vector(property_price.joint[,2]),
ylab='CCF', main = "Sample CCF between Property Price Index and Population
Change")
```

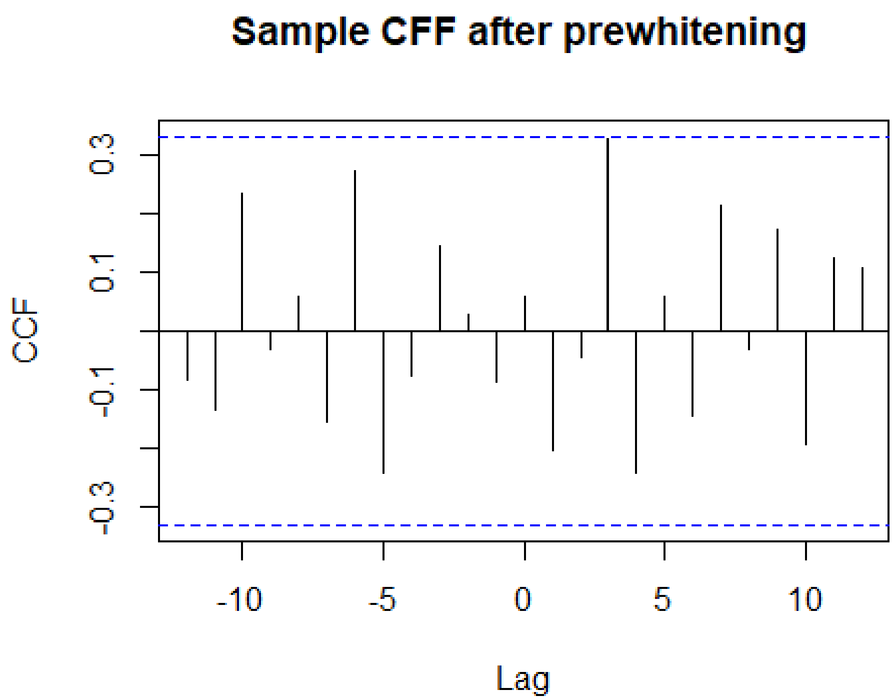


* Looking at the cross correlation function has lot of significant values above the 95% confidence interval, which significantly gives evidence of existence of spurious correlation.

Prewhitening

- We've noticed that when dealing with highly autocorrelated data, it becomes challenging to evaluate the relationship between the two processes. To effectively separate the linear connection between the two variables from the autocorrelation present in the series, prewhitening is a valuable approach.

```
new.dif=ts.intersect(diff(diff(price_change,4)),diff(diff(population_change,4
)))
prewhiten(as.vector(new.dif[,1]),as.vector(new.dif[,2]),ylab='CCF',
main="Sample CFF after prewhitening ")
```



* It appears that the correlation observed between Residential PPI and population change is misleading or spurious because of the difference in correlation structure between the two variables

Conclusion

- After using the prewhitening technique and examining the Cross-Correlation Function (CCF) plot, it's clear that there isn't a statistically significant correlation between the Residential Property Price Index and population change in Victoria. This means that the strong connection we initially thought existed between these two variables when not considering factors like autocorrelation was actually not true.The relationship between the Property Price Index and population change, which seemed strong without considering autocorrelation, doesn't hold up when we account for the underlying autocorrelation in the data. The prewhitening technique was crucial in revealing the true nature of this relationship, showing that it isn't significant once we consider autocorrelation properly.

References:

Hudson, I. Time Series Regression Models I - Distributed lag models - By using dLagM R-package [Lab solutions]. MATH1307: Forecasting. MATH1307 Forecasting RMIT University, School of Science, Mathematical Sciences.

Hudson, I. Time Series Regression Models II - Dynamic Models. MATH1307: Forecasting. MATH1307 Forecasting. RMIT University, School of Science, Mathematical Sciences.

Hudson, I. Exponential Smoothing Methods. MATH1307: Forecasting. MATH1307 Forecasting. RMIT University, School of Science, Mathematical Sciences.

Hudson, I. Linear Innovations State Space Models. MATH1307: Forecasting. MATH1307 Forecasting. RMIT University, School of Science, Mathematical Sciences.

Hudson, I. Nonlinear and Heteroscedastic Innovations State Space Models. MATH1307: Forecasting. MATH1307 Forecasting. RMIT University, School of Science, Mathematical Sciences.

G.G. Judge, W.E. Griffiths. Undergraduate Econometrics. Wiley, 2000.