Forecasting Assignment - 2

VISHAL PATNAIK DAMODARAPATRUNI - s3811521

18/09/2021

Table of Contents

Introduction

This analysis has two parts: Part 1: Forecasting Solar Radiation data. Part 2: Correlation Analysis.

Part 1:

Here, we will forecast the horizontal solar radiation data for the next two years using the best fit model. To get this best model we have three approaches. 1. Suitable DL model fitting 2. Smoothing methods 3. State Space models

The best model is the one that has the best MASE score as well as gives the best residual analysis.

Part 2:

The correlation strength is to be accessed among quarterly Residential Property Price Index (PPI) in Melbourne and quarterly population change over the previous quarter in Victoria and then check whether the relationship is spurious or not.

Method

Part 1

The following packages are used by both Part 1 and Part 2.

```
library(dplyr)
library(forecast) # Forecasting Functions for Time Series and Linear Models. [1]
library(dLagM) # Distributed lag model.
library(lmtest) # Testing Linear Regression Models. [2]- https://cran.r-project.org/web/packages/lmtest/index.html
library(tidyr)
library(tseries) # Time Series Analysis and Computational Finance. [3] - https://cran.r-project.org/web/packages/tseries/index.html
library(fUnitRoots) # To analyze trends and unit roots in financial time series. [4] - https://cran.r-project.org/web/packages/fUnitRoots/index.html
library(expsmooth) # Forecasting with Exponential Smoothing. [5] - https://cran.r-project.org/web/packages/expsmooth/index.html
library(TSA) # Time Series Analysis.
library(urca) # Unit Root and Cointegration Tests. [6] - https://cran.r-project.org/web/packages/urca/index.html
library(readr)
```

Data

The data here used is the monthly average horizontal solar radiation and the monthly precipitation series measured at the same points between January 1960 and December 2014.

```
v_Task1_data <- read.csv("data1.csv", header = TRUE)
head(v_Task1_data)

##       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

# Using str() to check the type of each column.
str(v_Task1_data)

## 'data.frame':    660 obs. of  2 variables:
##  $ solar: num  5.05 6.42 10.85 16.93 24.03 ...
##  $ ppt  : num  1.333 0.921 0.947 0.615 0.544 ...
```

Checking for Missing values.

```
colSums(is.na(v_Task1_data))

## solar   ppt
##     0     0
```

There are no missing values in the data.

Checking the class of v_solar_data. (It should be a data frame.)

```
class(v_Task1_data)

## [1] "data.frame"

v_solar_radiation_TS <- ts(v_Task1_data$solar, start = c(1960, 1), frequency
= 12)
v_precipitation_TS <- ts(v_Task1_data$ppt, start = c(1960, 1), frequency =
12)
```

Confirming the class of each time series object.

```
class(v_precipitation_TS)

## [1] "ts"

class(v_solar_radiation_TS)

## [1] "ts"
```

Now let us perform descriptive analysis on each time series object.

Descriptive Analysis

Solar radiation

```
plot(v_solar_radiation_TS, type = "b", xlab = "years", ylab = "Radiation
amount", main = "Time series plot for solar radiation from 1960-1 to 2014-12
(660 months)", pch = 1)
legend("topright", inset = .03, title = "Radiation amount", legend = "Solar
radiation series", horiz = TRUE, cex = 0.7, lty = 1, box.lty = 2, box.lwd =
2, pch = 1)
points(v_solar_radiation_TS, x = time(v_solar_radiation_TS), pch =
as.vector(season(v_solar_radiation_TS)))
```

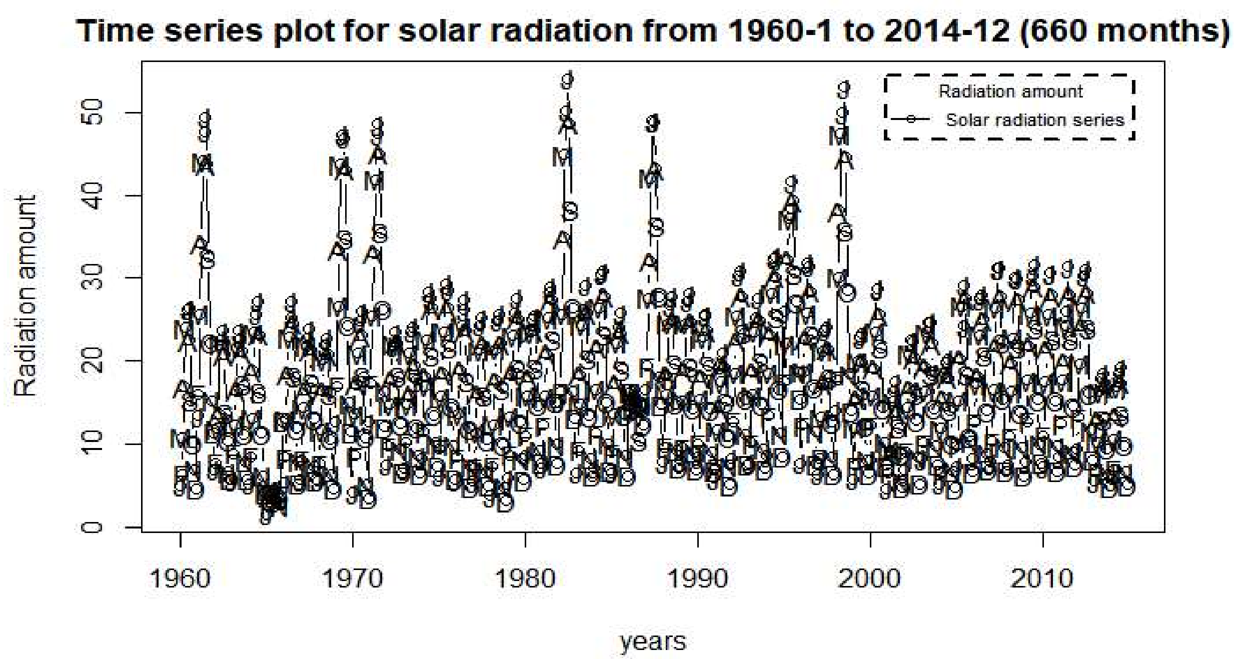


Fig 1.1: Solar radiation - Time series plot.

```
McLeod.Li.test(y = v_solar_radiation_TS, main = "McLeod-Li Test Statistics
for Solar radiation.")
```

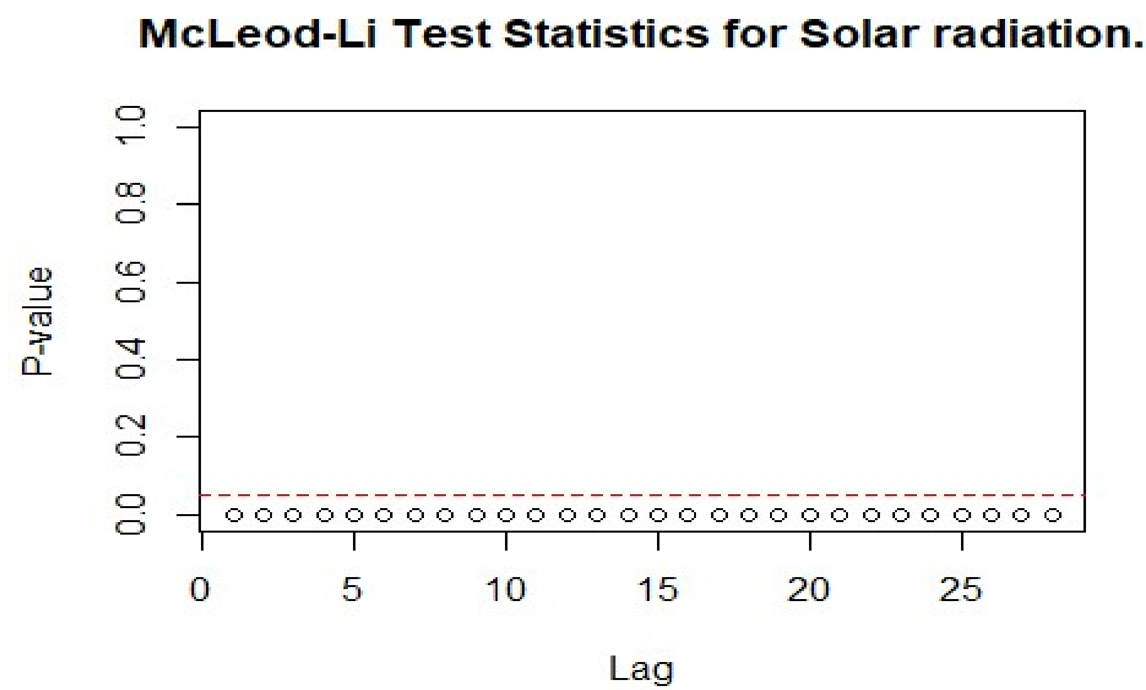


Fig 1.2: McLeod-Li Test Statistics for Solar radiation.

Descriptive analysis

1. From fig1, we can observe that there is no trend in the data.
2. There is an intervention around the years, 1965 and 1967.
3. From fig1, we can conclude that there is seasonality in the series.
4. It has lower values in the months of January and December, where as the higher values in the months of June and July. This shows that there is no consistency across the observed period of time.
5. Therefore, there is no change Autoregressive and moving average behaviour.
6. Also, we cannot see change in variance.

Precipitation

```
plot(v_precipitation_TS, type = "b", xlab = "years", ylab = "Precipitation",
main = "Time series plot for monthly precipitation from 1960-1 to 2014-12
(660 months)", pch = 1)
legend("topleft", inset = .03, title = "Precipitation", legend =
"Precipitation series", horiz = TRUE, cex = 0.8, lty = 1, box.lty = 2,
box.lwd = 2, pch = 1)
points(v_precipitation_TS, x = time(v_precipitation_TS), pch =
as.vector(season(v_precipitation_TS)))
```

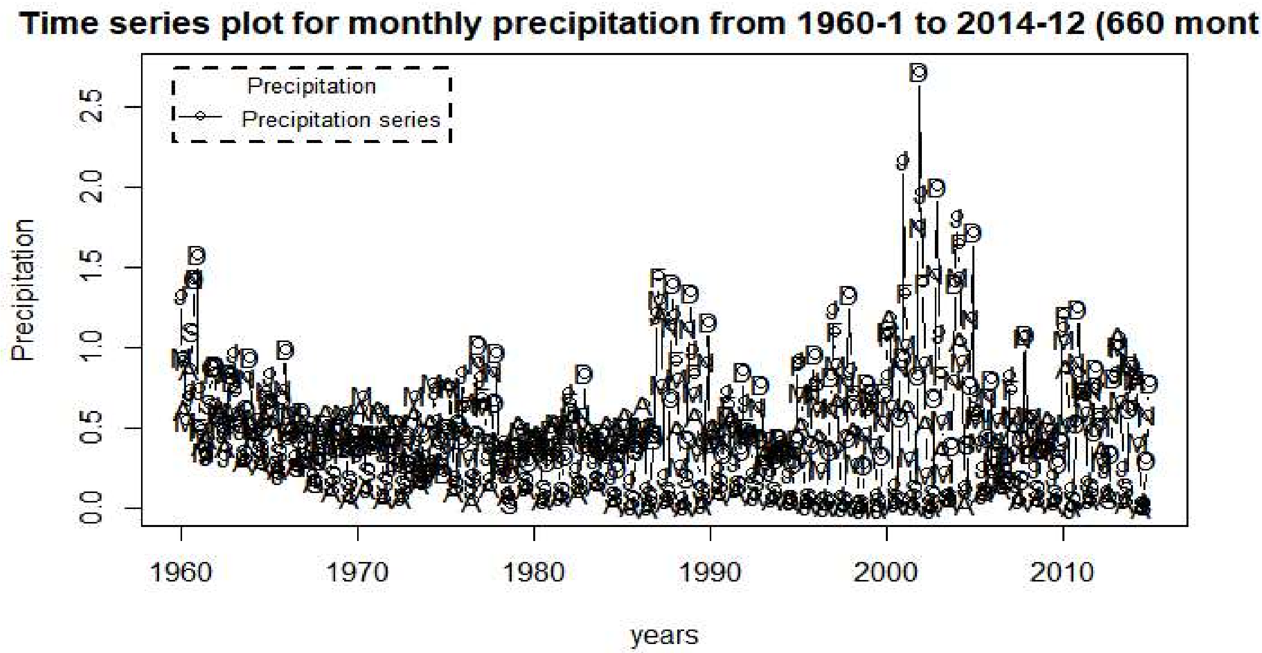


Fig 1.3: Solar radiation - Time series plot.

```
McLeod.Li.test(y = v_precipitation_TS, main = "McLeod-Li Test Statistics for Precipitation.")
```

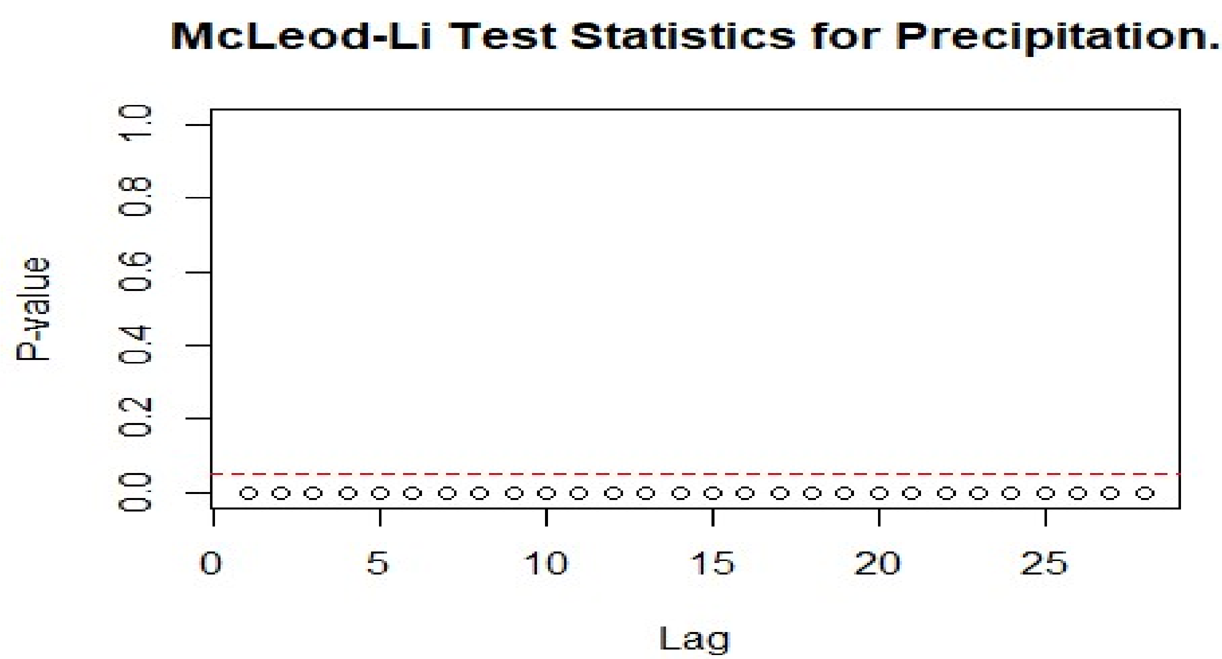


Fig 1.4: McLeod-Li Test Statistics for Precipitation.

Descriptive analysis

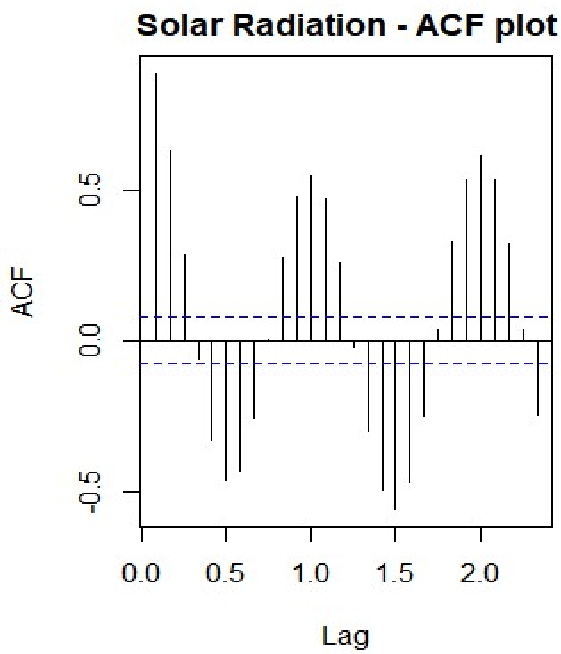
1. From fig1, we can observe that there is no trend in the data.
2. There are no obvious intervention points in the series.
3. From fig1, we can conclude that there is seasonality in the series.
4. It has lower values in the months of August and September, where as the higher values in the months of January and December. This shows that there is no consistency across the observed period of time.
5. Therefore, there is no change Autoregressive and moving average behaviour.
6. Also, we cannot see change in variance.

Checking for Stationary in the series

```
# Function to check Stationary on the series.
Stationary_Check <- function(x, m1, m2) {

  # Analysing trends by plotting ACF and PACF.
  par(mfrow = c(1,2))
  acf(x, main = m1)
  pacf(x, main = m2)

  # Lag for ADF test
  d = ar(x)$order

  # Conducting Augmented Dickey-Fuller test.
  adf.test(x, k = d)
}
```

Checking for Stationary on Solar Radiation series.

```
Stationary_Check(v_solar_radiation_TS, "Solar Radiation - ACF plot", "Solar
Radiation - PACF plot")

## Warning in adf.test(x, k = d): p-value smaller than printed p-value
```



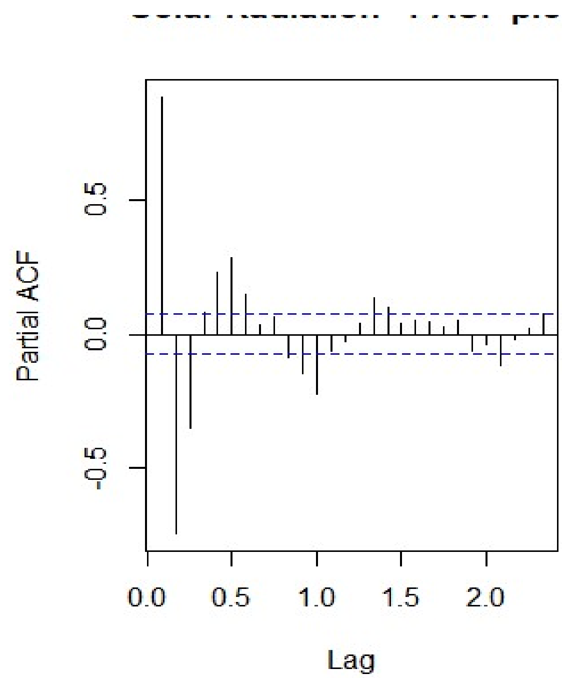


Fig 1.5: Solar Radiation - ACF

Fig 1.6: Solar Radiation - ACF

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

The seasonal pattern in the significant lags suggests that there is no trend in the series.

Hypotheses:

H_0: The data is not stationary.

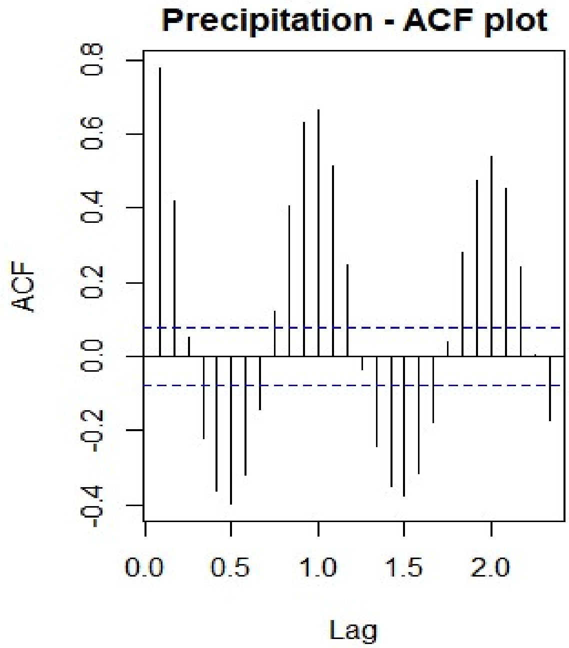
H_A: The data is stationary.

Interpretations:

- p - value: ~ 0.01 < 0.05
- p - value is less than 0.05 and hence the test is statistically significant. Therefore, we Null hypothesis can be rejected i.e., The data is stationary.
- Therefore, the Solar Radiation series is Stationary.

Checking for Stationary on Precipitation data.

```
Stationary_Check(v_precipitation_TS, "Precipitation - ACF plot",
"Precipitation - PACF plot")
```



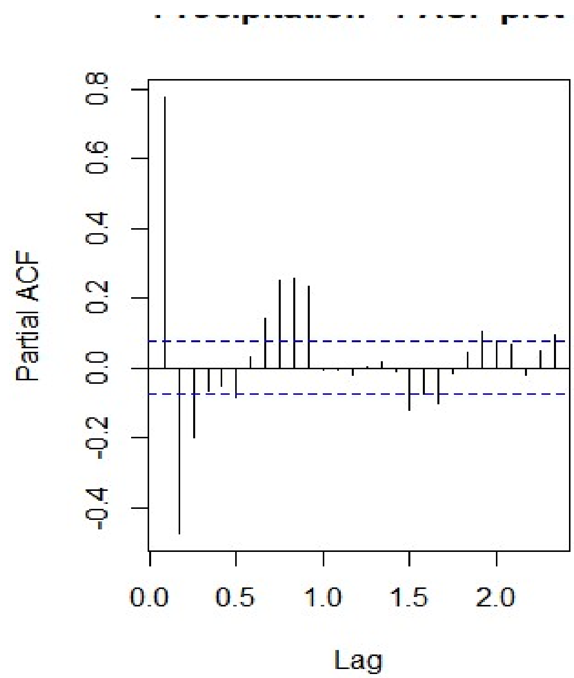


Fig 1.7: Precipitation - ACF

Fig 1.8: Precipitation - PACF

```
##
##  Augmented Dickey-Fuller Test
##
## data:  x
## Dickey-Fuller = -3.2594, Lag order = 28, p-value = 0.07769
## alternative hypothesis: stationary
```

Fig 1.7: Precipitation - ACF Fig 1.8: Precipitation - PACF

The seasonal pattern in the significant lags suggests that there is no trend in the series.

Hypotheses:

H_0: The data is not stationary.

H_A: The data is stationary.

Interpretations:

- p - value : 0.07769 > 0.05
- p - value is less than 0.05 and hence the test is statistically significant. Therefore, we Null hypothesis can be rejected i.e., The data is stationary.
- Therefore, the Precipitation series is Stationary.

Therefore, no differentiation is required. As the two series are stationary.

Suitable distributed lag model.

Before this let us find the correlation between the two series.

```
# Calculating the correlation coefficient between solar radiation and
precipitation.
cor(v_solar_radiation_TS, v_precipitation_TS)

## [1] -0.4540277
```

This suggests that there is a negative correlation between the series.

```
v_prep_fore_data <- read.csv("data.x.csv", header = TRUE)
head(v_prep_fore_data)

##           x
## 1 0.1890100
## 2 0.6972625
## 3 0.5952135
## 4 0.4873885
## 5 0.2616770
## 6 0.8086067

v_prep_fore_TS <- ts(v_prep_fore_data , start = c(2015, 1), frequency = 12)
v_prep_fore_TS

##             Jan        Feb        Mar        Apr        May        Jun
## 2015 0.18901000 0.69726252 0.59521349 0.48738853 0.26167702 0.80860665
## 2016 0.10986063 0.78146471 0.69685501 0.50241391 0.64938561 0.74596077
##             Jul        Aug        Sep        Oct        Nov        Dec
## 2015 0.94186202 0.90563633 1.05996468 0.34143878 0.52580532 0.60247106
## 2016 0.66304712 0.53377011 0.61542621 0.54606508 0.14267332 0.01365041
```

As we are going to forecast the solar radiation data our dependent variable "x" will be solar radiation series object and independent variable "y" will be precipitation data.

Finite distributed lag model

```
x = v_precipitation_TS # Independent variable
y = v_solar_radiation_TS # Dependent variable


for ( i in 1:10){
  model_1 = dlm(x = as.vector(x) , y = as.vector(y), q = i )
  cat("q = ", i, "AIC = ", AIC(model_1$model), "BIC = ", BIC(model_1$model),
"MASE =", MASE(model_1)$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
```

As we have the least AIC and BIC values at q = 10. Let us fit the finite distributed lag model with q = 10.

```
# Finite lag length based on AIC-BIC

finite_dlm_solar_rad = dlm( x = as.vector(x) , y = as.vector(y), q = 10)
summary(finite_dlm_solar_rad)

## 
## 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
```

Hypotheses:

H_0: The data doesn't fit the Finite distributed lag model.

HA: The data fits the Finite distributed lag model.

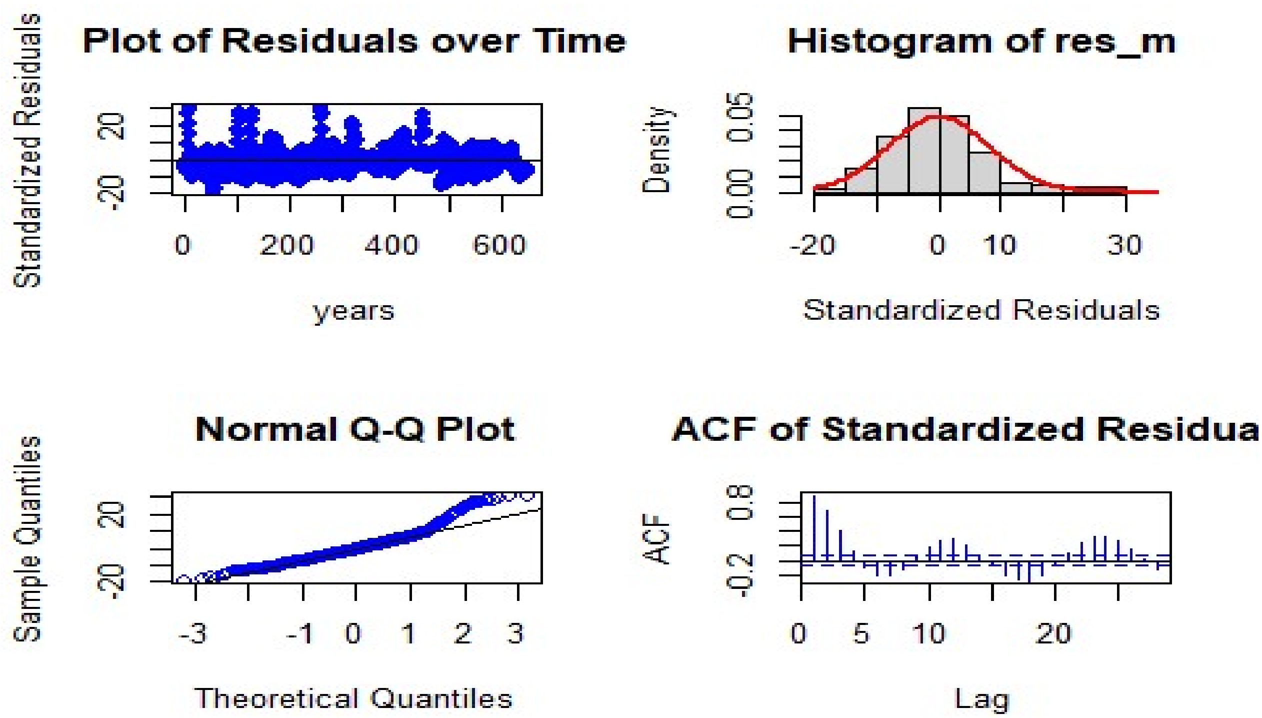
Interpretations:

- F - statistic is 25.82
- R - squared is 0.3081
- Adjusted R - squared is 0.2962
- Degrees of freedom - DF are (11, 638)
- p - value (~ 0.01) is < 0.05 and therefore, it is statistically significant. Therefore, Null hypothesis is rejected. Hence, the model fits the Finite distributed lag model.

This model suggests that there is only 29.62% of data variance. Suggesting that the model explains only 29.62% of the trend. Which implies that the model shows some trend.

Residual analysis

```
# Function for residual analysis.

res_analysis <- function(res_m) {

    par(mfrow = c(2, 2))
    # Scatter plot for model residuals
    plot(res_m, type = "b", pch = 19, col = "blue", xlab = "years", ylab = 
"Standardized Residuals", main = "Plot of Residuals over Time")

    abline(h = 0)

    # Standard distribution
    hist(res_m, xlab = 'Standardized Residuals', freq = FALSE)
    curve(dnorm(x, mean = mean(res_m), sd = sd(res_m)), col = "red", lwd = 2, 
add = TRUE, yaxt = "n")

    # QQplot for model residuals
    qqnorm(res_m, col = c("blue"))
    qqline(res_m)

    # Auto-Correlation Plot
    acf(res_m, main = "ACF of Standardized Residuals",col=c("blue"))

    # Shapiro Wilk test
    shapiro.test(res_m)

}

res_analysis(residuals(finite_dlm_solar_rad$model))
```




```
##
##  Shapiro-Wilk normality test
##
## data:  res_m
## W = 0.94643, p-value = 1.386e-14
```

Residual Analysis for Finite DLM:

1. The data points are below the line at the start and below the line at the end of the trend. Randomness is seen to some extent. So, we cannot decide anything at this stage. Further analysis is required.
2. From normal distribution curve, the distribution is almost symmetric.
3. The data at the tails is deviated more leaving some part on the line suggesting there is some normality in the trend.
4. There are significant lags in Autocorrelation plot suggesting that the stochastic component is not white noise. Also, ACF shows seasonality pattern.
5. p - value (~ 0.01) from Shapiro-Wilk normality test is < 0.05 and therefore, it is statistically significant. Therefore, Null hypothesis is rejected. Suggesting some normality.

Therefore, Further analysis is needed by adding polynomial to the lag model.

Polynomial distributed lag model

```
for (i in 1:3){
  model_2 <-  polyDlm(x = as.vector(x) , y = as.vector(y), q = i , k = i,
```

```
show.beta = FALSE)
  cat("q = ", i, "k = ", i, "AIC = ", AIC(model_2$model), "BIC = ",
BIC(model_2$model),"\n")
}

## q =  1 k =  1 AIC =  4728.713 BIC =  4746.676
## q =  2 k =  2 AIC =  4712.649 BIC =  4735.095
## q =  3 k =  3 AIC =  4688.551 BIC =  4715.478
```

Let us fit a polynomial model of order 3. Since least AIC and BIC scores.

```
# Ploynomial DLM

PolyDLM_model_solar = polyDlm(x = as.vector(x), y = as.vector(y), q = 3, k =
3, show.beta = TRUE)

## Estimates and t-tests for beta coefficients:
##        Estimate Std. Error t value  P(>|t|)
## beta.0  -11.400       1.77  -6.450 2.13e-10
## beta.1   -0.566       2.58  -0.219 8.26e-01
## beta.2   -2.490       2.57  -0.967 3.34e-01
## beta.3    7.820       1.76   4.450 1.01e-05

summary(PolyDLM_model_solar)

##
## Call:
## "Y ~ (Intercept) + X.t"
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.626  -5.831  -1.118   4.390  31.812
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  20.9176     0.7047  29.682  < 2e-16 ***
## z.t0        -11.4184     1.7694  -6.453 2.13e-10 ***
## z.t1         25.5737    13.1047   1.951   0.0514 .
## z.t2        -18.8879    12.0350  -1.569   0.1170
## z.t3          4.1669     2.6606   1.566   0.1178
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.533 on 652 degrees of freedom
## Multiple R-squared:  0.2479, Adjusted R-squared:  0.2433
## F-statistic: 53.72 on 4 and 652 DF,  p-value: < 2.2e-16
```

Hypotheses:

H_0: The data doesn't fit the Polynomial distributed lag model.

H_A: The data fits the Polynomial distributed lag model.

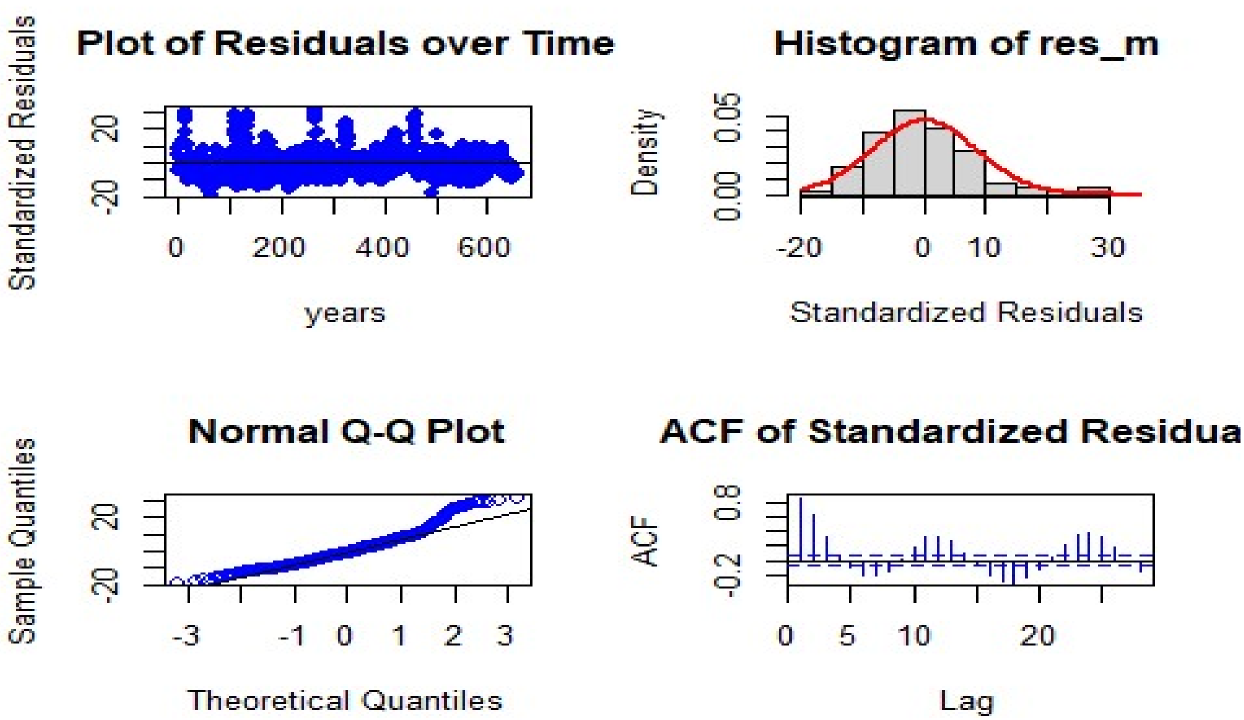
Interpretations:

- F - statistic is 53.72
- R - squared is 0.2479
- Adjusted R - squared is 0.2433
- Degrees of freedom - DF are (4, 652)
- p - value (~ 0.01) is < 0.05 and therefore, it is statistically significant. Therefore, Null hypothesis is rejected. Hence, the model fits the Polynomial distributed lag model.

This model suggests that there is only 24.33% of data variance. Suggesting that the model explains only 24.33% of the trend. Which implies that the model shows some trend.

Residual analysis

```
res_analysis(residuals(PolyDLM_model_solar$model))
```




```
##
##  Shapiro-Wilk normality test
##
## data:  res_m
## W = 0.94613, p-value = 1.013e-14
```

Residual Analysis for Polynomial DLM:

1. The data points are below the line at the start and below the line at the end of the trend. Randomness is seen to some extent. So, we cannot decide anything at this stage. Further analysis is required.

2. From normal distribution curve, the distribution is almost symmetric.
3. The data at the tails is deviated more leaving some part on the line suggesting there is some normality in the trend.
4. There are significant lags in Autocorrelation plot suggesting that the stochastic component is not white noise. Also, ACF shows seasonality pattern.
5. p - value (~ 0.01) from Shapiro-Wilk normality test is < 0.05 and therefore, it is statistically significant. Therefore, Null hypothesis is rejected. Suggesting some normality.

This analysis is not enough and we still require a better model than this. Therefore, let us fit Koyck model.

Koyck model

```
# Koyk DLM

Koyck_DLM_solar = koyckDlm(x = as.vector(x) , y = as.vector(y))
summary(Koyck_DLM_solar)

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

Hypotheses:

H_0: The data doesn't fit the Koyck distributed lag model.
H_A: The data fits the Koyck distributed lag model.

Interpretations:

Wald test statistic is 1104

- R - squared is 0.7598
- Adjusted R - squared is 0.7591
- Degrees of freedom - DF are (2, 656)
- p - value (~ 0.01) is < 0.05 and therefore, it is statistically significant. Therefore, Null hypothesis is rejected. Hence, the model fits the Koyck distributed lag model.

This model suggests that there is only 75.91% of data variance. Suggesting that the model explains only 75.91% of the trend. Which implies that the model performs better on the series data when compared to the former values.

Now let us perform residual analysis.

Residual analysis

```
res_analysis(residuals(Koyck_DLM_solar))

##           2           3           4           5           6 
7 
##  -1.24764058   1.70115896   5.19100459   6.67724783   1.09720481  -
1.26795959 
##           8           9          10          11          12 
13 
##  -5.90870316 -10.55198103 -10.27887277  -8.46692341  -8.44511063 
7.12209433 
##          14          15          16          17          18 
19 
##   2.33750859   9.51201375   8.99277568  10.73479855   4.64243658 
2.91748472 
##          20          21          22          23          24 
25 
##  -5.02564918 -11.79872346 -12.29939787  -8.93714088  -6.16910152  -
5.51149275 
##          26          27          28          29          30 
31 
##   0.27104782   3.25616886   0.95669801   8.19132959   0.66048723 
1.88539605 
##          32          33          34          35          36 
37 
##  -2.41247478  -7.28871401  -5.31102258  -5.89893720  -2.83201459  -
2.89508758 
##          38          39          40          41          42 
43 
##   1.61796574   2.59604609   3.42244512   4.76091472   1.96550418  -
0.01373900 
##          44          45          46          47          48 
49 
##  -0.63390787  -3.29269650  -6.79695740  -5.24732785  -4.78788118  -
1.13472596 
```

```
##          50          51          52          53          54
55
##  3.19823022  4.41325664  5.63406058  4.15649716  2.73254506
2.40894744
##          56          57          58          59          60
61
## -2.88521084 -6.39546652 -5.21662766 -6.58076877 -3.38995679 -
4.70779194
##          62          63          64          65          66
67
## -0.41896660  1.13371766 -1.30494698  1.12957909  1.28806919 -
0.10785645
##          68          69          70          71          72
73
##  1.27296577  0.85238128 -1.80205874 -2.53221608  7.12786589 -
8.19598433
##          74          75          76          77          78
79
##  3.09010555  4.42788724  5.20173278  3.77355596  2.37958244
2.83679702
##          80          81          82          83          84
85
## -1.49581120 -5.17684032 -6.04389141 -4.22318665 -3.50392728
0.64915110
##          86          87          88          89          90
91
##  3.25590543  6.90534880  1.88513537  4.96950498  1.40647677
2.65574193
##          92          93          94          95          96
97
## -0.81063107 -2.25768465 -3.55075914 -4.87504285 -1.61295729 -
0.33882723
##          98          99         100         101         102
103
##  2.43810871  4.07370031  3.95932802  3.48107270  2.35676873
2.50362183
##         104         105         106         107         108
109
##  0.02149958 -3.18972784 -5.12754657 -4.52237802 -1.95877535
8.15511005
##         110         111         112         113         114
115
##  4.88878409 10.05073393  6.28905815  9.86349490  4.35930343
2.01340135
##         116         117         118         119         120
121
## -1.31651837 -6.40072982 -9.78775865 -9.11114547 -3.61969072 -
6.71666595
##         122         123         124         125         126
```

```
127
##   4.86245843   5.14182848   2.73156123   2.48472635   4.06271516
3.09887072
##          128          129          130          131          132
133
##  -0.97002771  -3.56321183  -7.36352845  -5.77005287  -1.47151067
9.29564936
##          134          135          136          137          138
139
##   4.84420145   8.22344869   6.82375570   8.40404569   6.49298108
3.59072183
##          140          141          142          143          144
145
##  -0.90171282  -7.50822520  -8.45810893  -9.40274978  -4.37372626  -
3.48970748
##          146          147          148          149          150
151
##   2.93994886   5.21151355   2.52795143   3.87116465   0.84966204
3.69158588
##          152          153          154          155          156
157
##   0.95949602  -2.25092969  -5.67250353  -2.91447012  -1.91875711
1.70621433
##          158          159          160          161          162
163
##   4.25102841   5.26479415   0.41880618   3.86779208   2.88823616
1.84253538
##          164          165          166          167          168
169
##  -2.93070163  -0.29638632  -5.75123988  -2.70832261  -0.94359514
3.00634226
##          170          171          172          173          174
175
##   4.40865842   6.34981249   0.86599205   2.69118800   4.40686876  -
1.15080026
##          176          177          178          179          180
181
##  -0.69953468  -4.42552930  -7.04750301  -2.38468468  -1.66291980  -
0.78657346
##          182          183          184          185          186
187
##   2.99614122   6.01485845   1.02858809   3.60129842   5.70667570
0.87293280
##          188          189          190          191          192
193
##  -0.09942829  -2.81157261  -7.54809169  -4.85233304  -4.02901543  -
2.21361918
##          194          195          196          197          198
199
```

```
##   1.61942886   4.10407332   5.17823120   6.01738767   3.38196266
3.28446713
##          200          201          202          203          204
205
##  -1.37002953  -4.41606676  -5.84264837  -6.95643855  -5.64270783  -
1.38678800
##          206          207          208          209          210
211
##   0.30525373   3.14389878   3.59184457   5.46988005   3.77372417
2.52008421
##          212          213          214          215          216
217
##  -1.28841005  -5.76375016  -6.34455580  -5.99891282  -4.60170890  -
0.84445624
##          218          219          220          221          222
223
##   4.56263548   5.95590107   5.67567609   4.08223588   4.78374045
2.91518886
##          224          225          226          227          228
229
##  -1.40608778  -2.97002429  -5.48204108  -4.24329552  -1.33336620
2.82157437
##          230          231          232          233          234
235
##   3.34661709   4.99347677   4.67021919   4.62689311   3.61929071
3.55690507
##          236          237          238          239          240
241
##  -0.44857570  -4.20211676  -5.63720496  -4.10464480  -2.37719926
2.94549463
##          242          243          244          245          246
247
##   2.81086498   5.29134837   2.14766626   5.58610747   2.25601585
2.53836893
##          248          249          250          251          252
253
##   0.65317816  -2.49720337  -3.57248703  -5.04436005  -1.84934404  -
0.35007171
##          254          255          256          257          258
259
##   5.76594875   3.78660132   4.77111420   3.95446590   4.10539643
2.78918418
##          260          261          262          263          264
265
##   0.68548634  -2.70242170  -6.81673233  -5.11312982  -2.39957544
5.92758032
##          266          267          268          269          270
271
##   0.20534247   8.86125616   8.73872432  10.27676034   6.86469507
```

```
5.96915606
##         272         273         274         275         276
277
##  -2.86574926  -8.21223050 -11.25367342 -10.71765391  -5.17608160  -
6.37455266
##         278         279         280         281         282
283
##   3.10331802   5.60741978   3.39450276   6.04781111   2.86670579
4.47851947
##         284         285         286         287         288
289
##  -1.48633182  -3.70113326  -6.46567658  -4.73048029  -2.21369828
1.23307783
##         290         291         292         293         294
295
##   4.82624033   5.74301347   4.80141918   5.11985096   5.54173504
2.53465400
##         296         297         298         299         300
301
##  -0.43232071  -2.39335320  -6.99999398  -4.38568142  -3.04783156  -
0.14395181
##         302         303         304         305         306
307
##   2.79835518   4.25773572   3.01468918   4.89020712   3.34523546
4.30828020
##         308         309         310         311         312
313
##   0.08667086  -2.08118512  -4.98161986  -4.91486824  -2.21344459
9.53629172
##         314         315         316         317         318
319
##   1.11171697  -0.18249526  -1.69387522  -1.66832539   3.02743933
1.86836964
##         320         321         322         323         324
325
##   1.55290922  -1.67920958   4.16947143  -0.06960535  -2.22273710
0.22392724
##         326         327         328         329         330
331
##  -2.44399368   3.67846823   0.67474129   8.56832157   8.62783060
2.48723332
##         332         333         334         335         336
337
##  -2.99098348  -4.37981680  -9.58599315 -13.09257737  -8.25375242 -
10.96701299
##         338         339         340         341         342
343
##  -0.81023943   4.45118238   2.13138635   7.35377545   4.44413610
3.48827449
```

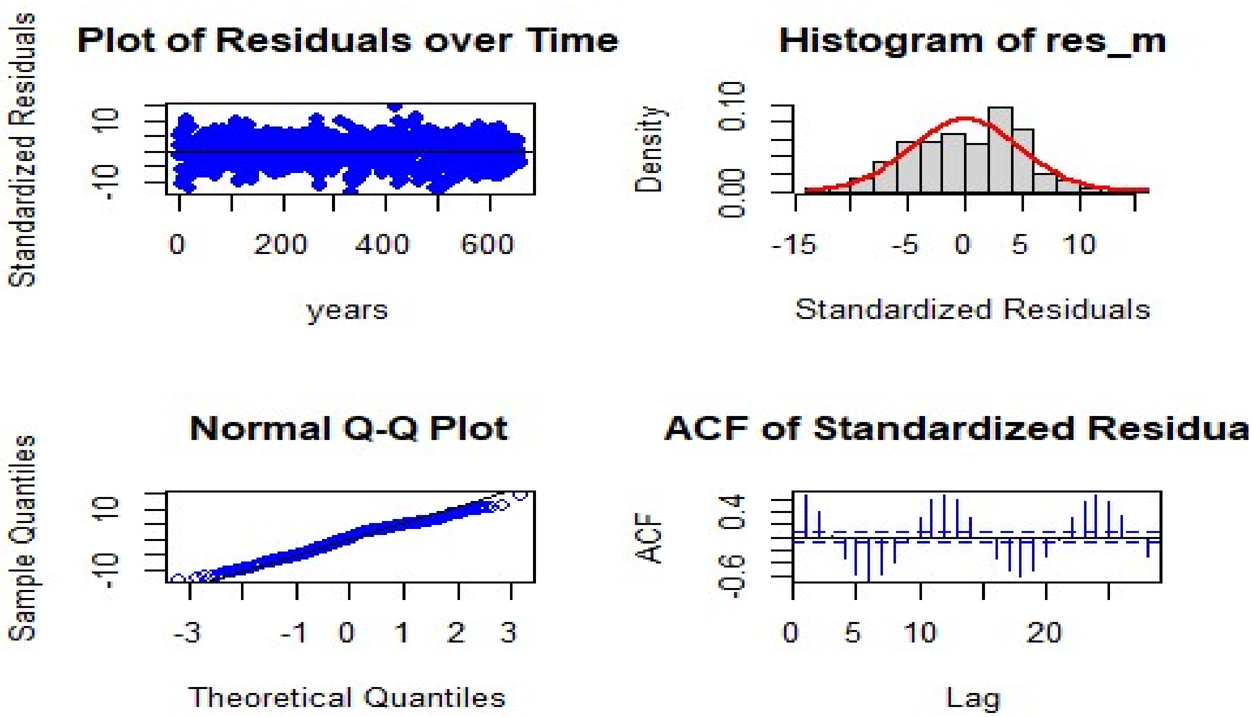
```
##          344          345          346          347          348
349
##  -0.27147623  -3.69274961  -4.62014483  -9.13856525  -6.95055437  -
2.06334845
##          350          351          352          353          354
355
##  -0.65394000   4.23588429   2.91299368   6.34406900   4.58417054
3.34640568
##          356          357          358          359          360
361
##  -0.88392724  -3.27847499  -4.93138283  -7.70722309  -5.86371963
1.11222174
##          362          363          364          365          366
367
##   1.99421057   5.32711862   2.40552182   5.14062507   3.76327429
1.93002067
##          368          369          370          371          372
373
##   0.26919199  -2.43128075  -4.67259972  -4.88112227  -3.01714985  -
2.50749039
##          374          375          376          377          378
379
##   0.13472488   3.93914568   2.08241204   2.87662449   3.13732698
3.11699949
##          380          381          382          383          384
385
##   0.30691182  -2.50474240  -3.72201358  -3.48429076  -4.55786739
1.85559434
##          386          387          388          389          390
391
##   4.18960245   5.62554460   3.89588094   3.32351454   6.22287620
2.79376578
##          392          393          394          395          396
397
##  -0.83466872  -2.72452725  -7.66261715  -5.19170545  -5.37168582
0.58193889
##          398          399          400          401          402
403
##   3.28753318   6.72940729   3.46133375   4.97200296   3.08014272
4.54493267
##          404          405          406          407          408
409
##  -0.31730267  -2.92655864  -4.40759248  -4.56954621  -2.44994561
1.67754277
##          410          411          412          413          414
415
##   5.30703950   5.71881601   5.12683512   5.57002284   5.73100848
2.59709234
##          416          417          418          419          420
```

```
421
## -0.06956093 -2.44673087 -7.59226259 -4.71048864 -2.74524978
14.83990519
##         422         423         424         425         426
427
## -10.66100250  3.31715535  9.89732891  6.05299287  3.54167256
5.96587595
##         428         429         430         431         432
433
##   0.31762551 -6.18654806 -2.96858147 -9.69032224 -5.81534057  -
9.45739213
##         434         435         436         437         438
439
##   2.90032655  5.44916922  4.02855901  4.40747869  8.33334268
2.31424276
##         440         441         442         443         444
445
##  -0.53082575 -2.67576404 -5.87948611 -6.11471132 -5.81895063  -
5.54224210
##         446         447         448         449         450
451
##  -1.62001893  2.44562640  2.88855452  4.53076281  4.59164257
2.97772284
##         452         453         454         455         456
457
##  -0.20654992 -1.11835068 -5.24681490 -5.96860348 -6.80377630
5.39417777
##         458         459         460         461         462
463
##   2.33751235 11.38848683  8.06234238 10.52636150  5.27015806
6.06217714
##         464         465         466         467         468
469
##  -5.53984181 -6.08220476 -5.93308837 -9.82312564 -6.16632322  -
9.21487544
##         470         471         472         473         474
475
##   0.10574786  3.14402221  2.54527325  4.85080655  4.93282389
2.97302713
##         476         477         478         479         480
481
##  -0.80903133 -0.90693636 -3.69573228 -4.68462084 -3.10539735  -
0.44100219
##         482         483         484         485         486
487
##   0.67624616  0.17118706 -0.59451507  3.47118958  7.04449450
2.30817505
##         488         489         490         491         492
493
```

```
##  -0.61085168  -1.60481644  -7.77904427  -6.42121805  -5.48454268 -
12.06000861
##          494          495          496          497          498
499
##  -2.92661294  -0.40617572  -1.32498271   6.81686191   3.26804157
3.49057436
##          500          501          502          503          504
505
##  -1.32025751  -0.03910850  -5.15336775 -10.37320133 -12.86165627  -
6.08164614
##          506          507          508          509          510
511
##  -3.08719549   2.88554497   1.92684951   5.48170690   3.91217550
2.53704518
##          512          513          514          515          516
517
##   0.48410922  -2.39291278  -4.42008908  -9.81021122 -11.56538804  -
0.50143132
##          518          519          520          521          522
523
##  -0.16777590   5.62191520   3.49844514   5.40969488   3.29161125
2.26633922
##          524          525          526          527          528
529
##  -0.20838031  -1.55246335  -3.62196126  -6.40112476  -7.32997764  -
9.65892822
##          530          531          532          533          534
535
##  -5.32070592   0.08754635   0.48158145   0.53406269   3.04179039
2.34106884
##          536          537          538          539          540
541
##   0.06975603  -3.53950717  -5.58841422  -7.33580174  -8.90073947
0.99857642
##          542          543          544          545          546
547
##   1.71358487   4.82649361   4.31009402   6.63474768  -1.13923131
7.04911649
##          548          549          550          551          552
553
##   0.44951015  -7.05282272  -7.35961776  -1.92313170  -5.56966104
3.06501565
##          554          555          556          557          558
559
##   5.59748473   5.42101945   1.24741631   4.54314520   4.64129597
1.93104467
##          560          561          562          563          564
565
##  -0.95735098  -2.58017101  -5.94015939  -3.97662412  -0.58101003  -
```

```
1.23202715
##         566         567         568         569         570
571
##  0.90411569  5.45188496  5.01186933  5.37554006  6.45312370
2.36819524
##         572         573         574         575         576
577
## -0.98806412 -3.78609465 -6.60297890 -9.01107902 -6.26816270
1.21465381
##         578         579         580         581         582
583
##  3.89358244  5.27640381  2.88154812  2.56741951  8.55566185
2.03538791
##         584         585         586         587         588
589
## -0.15771906 -3.72918475 -7.26374718 -3.76642267 -2.98942553
1.87258152
##         590         591         592         593         594
595
##  5.21344121  4.93561882  3.63916861  3.91761805  6.06514197
3.79010222
##         596         597         598         599         600
601
## -0.21331278 -2.86884370 -7.45936155 -4.82596700 -2.17012846  -
4.18815164
##         602         603         604         605         606
607
## -0.32729798  3.24809219  1.86426165  3.81709030  6.91328892
2.72123960
##         608         609         610         611         612
613
## -0.14385341 -2.91037307 -7.63955828 -7.92671528 -7.22441906  -
0.54401190
##         614         615         616         617         618
619
##  2.52265426  3.64100492  0.98388317  3.59380900  3.47601031
7.66571060
##         620         621         622         623         624
625
## -0.98035980 -3.41440016 -8.03136061 -7.08156062 -3.90008277
0.13942468
##         626         627         628         629         630
631
##  3.25277920  6.06889570  5.18480385  3.19907263  5.12075769
2.93553358
##         632         633         634         635         636
637
## -0.68525162 -1.97820215 -7.07368395 -5.05801043 -5.57057154  -
4.79465006
```

```
##          638          639          640          641          642
643
##  -1.82987686   0.43399120  -0.61790719   2.19003214   2.64656533
2.86967603
##          644          645          646          647          648
649
##   0.50780011  -1.94244395  -4.44776013  -5.56191964  -3.90385266  -
0.40028884
##          650          651          652          653          654
655
##  -0.83522144   1.72386295   0.92659680   3.70632712   3.89024711
3.08185927
##          656          657          658          659          660
##   0.64885503  -1.51016223  -2.87174571  -3.91206778  -3.39386776
```




```
##
##  Shapiro-Wilk normality test
##
## data:  res_m
## W = 0.98487, p-value = 2.468e-06
```

Residual Analysis for Koyck DLM:

1. The data points are below the line at the start and below the line at the end of the trend. Randomness is seen to some extent. So, we cannot decide anything at this stage. Further analysis is required.

2. From normal distribution curve, the distribution is almost symmetric.
3. The data at the tails is deviated more leaving some part on the line suggesting there is some normality in the trend.
4. There are significant lags in Autocorrelation plot suggesting that the stochastic component is not white noise. Also, ACF shows seasonality pattern.
5. p - value (~ 0.01) from Shapiro-Wilk normality test is < 0.05 and therefore, it is statistically significant. Therefore, Null hypothesis is rejected. Suggesting some normality.

So far this is the best model but let us fit ardlDlm model to check whether it fits better than Koyck model or not.

Autoregressive distributed lag model

```
for (i in 1:5){
  for(j in 1:5){
    model_4 = ardlDlm(x = as.vector(x) , y = as.vector(y), p = i , q = j )
    cat("p = ", i, "q = ", j, "AIC = ", AIC(model_4$model), "BIC = ",
BIC(model_4$model), "MASE =", MASE(model_4)$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
```

p = 3, 4, 5 and q = 5 has the least AIC, BIC and MASE scores.

```
# ARDLM model
AR_DLM_solar_35 = ardlDlm(x = as.vector(x) , y = as.vector(y), p = 3 , q = 5)
summary(AR_DLM_solar_35)

##
## 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
## -15.6649  -1.4447  -0.2663   1.0644  18.7430
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.20532    0.42237   5.221 2.40e-07 ***
## X.t         -0.60604    0.54924  -1.103 0.270258
## X.1          0.89253    0.77682   1.149 0.251001
## X.2          1.60774    0.77838   2.065 0.039276 *
## X.3         -0.38294    0.55738  -0.687 0.492308
## Y.1          1.27345    0.03859  32.999  < 2e-16 ***
## Y.2         -0.02859    0.06250  -0.457 0.647495
## Y.3         -0.40351    0.06036  -6.685 5.01e-11 ***
## Y.4         -0.22632    0.06234  -3.630 0.000305 ***
## Y.5          0.22116    0.03779   5.853 7.69e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.553 on 645 degrees of freedom
## Multiple R-squared:  0.9333, Adjusted R-squared:  0.9324
## F-statistic:  1003 on 9 and 645 DF,  p-value: < 2.2e-16
```

Hypotheses:

H0: The data doesn't fit the Autoregressive distributed lag model.

HA: The data fits the Autoregressive distributed lag model.

Interpretations:

- F - statistic is 1003
- R - squared is 0.9333
- Adjusted R - squared is 0.9324
- Degrees of freedom - DF are (9, 645)
- p - value (~ 0.01) is < 0.05 and therefore, it is statistically significant. Therefore, Null hypothesis is rejected. Hence, the model fits the Autoregressive distributed lag model.

This model suggests that there is only 93.24% of data variance. Suggesting that the model explains only 93.24% of the trend. Which implies that the model shows some trend.

Now let us perform residual analysis.

Residual analysis

```
res_analysis(residuals(AR_DLM_solar_35))

## Time Series:
## Start = 6
## End = 660
## Frequency = 1
##             6             7             8             9            10
## -1.998657e+00 -1.586839e+00 -2.482872e+00 -3.517808e+00 -7.967281e-01
##            11            12            13            14            15
## -3.521716e-01 -2.946419e+00  5.400921e+00 -3.758763e+00  3.896074e+00
##            16            17            18            19            20
##  3.921938e+00  6.826539e+00  7.769822e-01  3.340244e+00 -6.240802e-01
##            21            22            23            24            25
## -2.716544e+00  2.154105e-01  2.788442e+00  1.136823e+00 -4.124404e+00
##            26            27            28            29            30
## -3.328475e-01  2.334129e+00 -2.327892e+00  4.029274e+00 -3.058781e+00
##            31            32            33            34            35
## -1.036650e+00 -2.319721e+00 -3.513881e+00  7.721070e-01 -5.953526e-01
##            36            37            38            39            40
## -1.102231e+00 -2.092578e+00  2.557575e-01 -1.061511e+00 -1.022085e-01
##            41            42            43            44            45
##  1.542313e+00 -7.048249e-01 -2.319286e+00 -4.250721e-01 -9.156722e-01
##            46            47            48            49            50
## -2.733038e+00 -1.181621e-01 -5.348033e-01 -2.041560e+00  3.760030e-01
##            51            52            53            54            55
##  1.384024e+00  1.042157e+00  2.922450e-01 -5.784524e-01  8.228824e-01
##            56            57            58            59            60
## -2.723485e+00 -3.573437e+00  3.638122e-01 -6.076834e-01 -8.145395e-01
##            61            62            63            64            65
## -3.676539e+00 -1.389692e+00 -1.464462e+00 -4.869620e+00 -3.292950e+00
##            66            67            68            69            70
## -1.587808e+00 -2.996971e+00 -1.479258e+00 -9.087724e-01 -2.486808e+00
##            71            72            73            74            75
## -3.414276e+00  7.994560e+00 -1.304748e+01 -7.877550e-01  4.651397e+00
##            76            77            78            79            80
##  3.740989e+00 -1.392297e+00 -4.673754e-01  1.097342e+00 -1.800547e+00
##            81            82            83            84            85
## -3.637586e+00 -1.310928e+00  1.117128e+00 -5.144814e-01 -3.644480e-01
##            86            87            88            89            90
## -3.151102e-01  2.830552e+00 -2.353012e+00  1.913440e+00 -1.385509e+00
##            91            92            93            94            95
##  6.518042e-01 -1.709326e+00 -1.096656e+00 -9.867997e-01 -2.232768e+00
##            96            97            98            99           100
##  6.546425e-01  3.693129e-01 -2.663158e-01  3.736791e-01  1.046609e+00
##           101           102           103           104           105
## -3.234023e-01 -1.315221e+00 -8.362043e-02 -9.756481e-01 -2.087045e+00
```

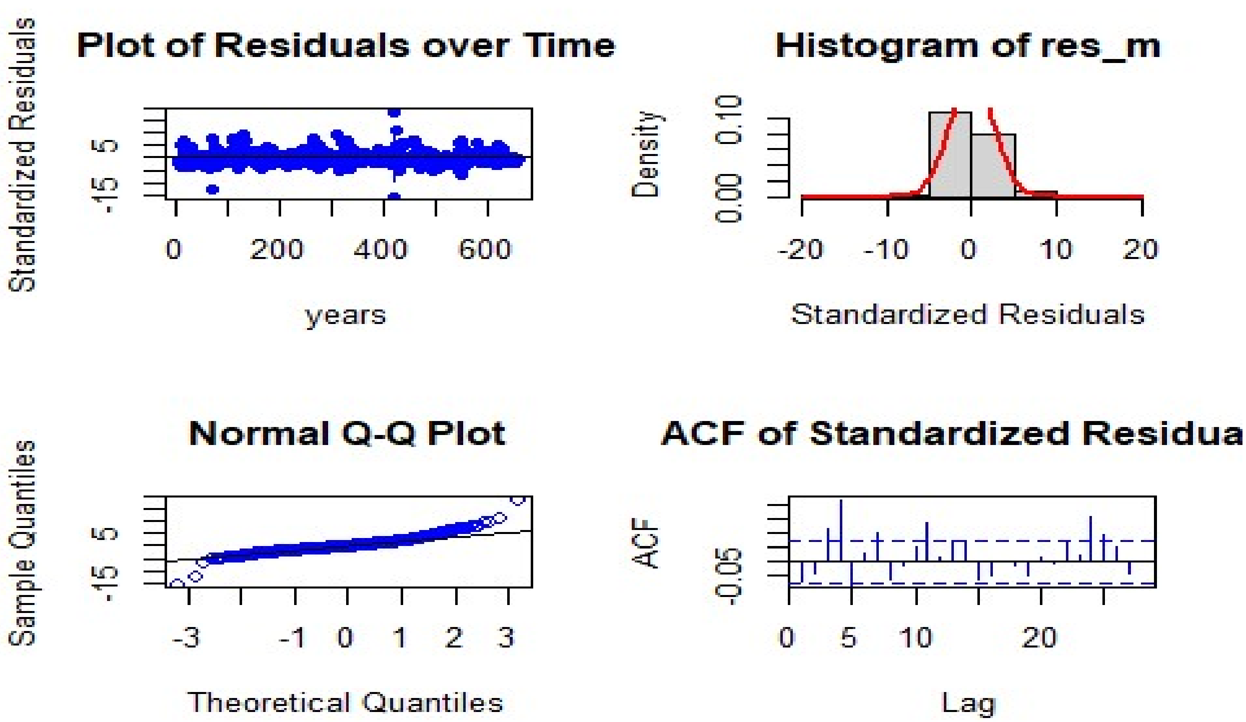
```
##           106           107           108           109           110 
## -1.965860e+00 -5.563978e-01  1.705312e-01  7.143727e+00 -5.507845e-01 
##           111           112           113           114           115 
##  3.715464e+00  2.065393e+00  7.737761e+00  1.044667e+00  5.936912e-01 
##           116           117           118           119           120 
##  1.260549e+00  6.500763e-01 -9.403982e-01  8.973695e-02  3.538409e+00 
##           121           122           123           124           125 
## -4.825225e+00  4.555335e+00  2.778564e+00 -1.526143e+00 -1.557072e+00 
##           126           127           128           129           130 
##  1.750326e+00 -2.764523e-01 -2.703049e+00 -1.707271e+00 -2.476365e+00 
##           131           132           133           134           135 
## -1.112659e+00  1.031885e+00  8.958461e+00 -1.147345e+00  1.229780e+00 
##           136           137           138           139           140 
##  3.088470e+00  6.670274e+00  3.074664e+00  1.443711e+00  6.912655e-01 
##           141           142           143           144           145 
## -1.222736e+00  1.061124e+00  2.273225e-01  2.902930e+00 -1.692397e+00 
##           146           147           148           149           150 
##  2.301956e+00  2.963500e+00 -1.297670e+00 -1.199869e-01 -2.359592e+00 
##           151           152           153           154           155 
##  1.537761e+00  5.274220e-02 -1.738510e+00 -2.633398e+00  8.785373e-01 
##           156           157           158           159           160 
## -8.887074e-01  2.702446e-01  1.524655e+00  1.502830e+00 -2.855226e+00 
##           161           162           163           164           165 
##  2.791592e+00  6.160989e-01 -1.054707e+00 -4.614100e+00  1.886379e+00 
##           166           167           168           169           170 
## -1.786819e+00 -7.531398e-01 -9.579059e-01  2.118500e+00  9.736975e-01 
##           171           172           173           174           175 
##  2.070379e+00 -2.651324e+00  2.327704e+00  4.614774e+00 -2.931266e+00 
##           176           177           178           179           180 
## -1.697345e+00 -2.499553e+00 -2.307553e+00  1.425499e+00  1.306652e-01 
##           181           182           183           184           185 
## -9.868574e-01  1.942011e+00  3.237957e+00 -2.365334e+00  1.132856e+00 
##           186           187           188           189           190 
##  3.400035e+00 -2.640768e+00 -1.792544e+00 -2.619755e-01 -1.610482e+00 
##           191           192           193           194           195 
##  6.338292e-02 -1.156910e+00  9.695304e-01  6.044228e-01  9.304281e-01 
##           196           197           198           199           200 
##  6.867079e-02  6.204584e-01 -9.331043e-01  1.088640e+00 -1.954579e+00 
##           201           202           203           204           205 
## -2.073733e+00 -5.024078e-01  3.544963e-01 -7.975669e-01 -8.095049e-01 
##           206           207           208           209           210 
## -1.783090e+00  8.824388e-02  2.061934e-01  1.002381e+00 -5.433142e-01 
##           211           212           213           214           215 
## -4.839785e-01 -2.328429e+00 -2.667293e+00 -2.998358e-01  7.475175e-02 
##           216           217           218           219           220 
## -7.892450e-01 -3.399592e+00  6.888852e-01  1.933480e+00  7.469409e-01 
##           221           222           223           224           225 
## -6.077273e-01  9.640503e-01  7.218628e-02 -2.437574e+00 -2.012616e+00 
##           226           227           228           229           230 
```

```
## -1.656097e+00 -4.146052e-01  9.112214e-01  2.216414e+00 -1.123983e+00
##           231           232           233           234           235
## -2.267135e-01  1.108395e+00  1.404956e+00 -9.294104e-02  7.937161e-01
##           236           237           238           239           240
## -1.336107e+00 -2.656975e+00 -1.565051e+00  7.556755e-01  2.961024e-01
##           241           242           243           244           245
##  2.797321e+00 -8.829990e-01  5.711884e-01 -1.526392e+00  3.348195e+00
##           246           247           248           249           250
## -3.847104e-01 -4.701838e-01 -4.716018e-01 -8.923507e-01 -5.738644e-01
##           251           252           253           254           255
## -1.576243e+00  1.132532e+00 -2.566211e-01  3.197437e+00 -4.021614e-01
##           256           257           258           259           260
##  1.158620e+00  1.018132e+00  1.262971e+00 -2.460597e-01  1.602700e-01
##           261           262           263           264           265
## -1.065591e+00 -2.681842e+00  3.655582e-01  1.500475e+00  7.988739e+00
##           266           267           268           269           270
## -3.227343e+00  3.953680e+00  5.020582e+00  6.401955e+00  1.627927e+00
##           271           272           273           274           275
##  4.842568e+00 -9.143687e-01 -1.741284e+00  6.758741e-02  2.031170e+00
##           276           277           278           279           280
##  4.859929e+00 -4.614741e+00  1.894711e+00  3.687398e+00 -8.960507e-01
##           281           282           283           284           285
##  7.911621e-01 -6.342050e-01  2.185914e+00 -2.549020e+00 -1.954858e+00
##           286           287           288           289           290
## -1.770111e+00  4.737650e-01  8.398221e-01  8.661766e-01  1.310749e+00
##           291           292           293           294           295
##  1.170026e+00  7.506194e-01  1.270037e+00  1.939994e+00 -1.708597e-01
##           296           297           298           299           300
## -9.802491e-01 -1.506967e-03 -2.451113e+00  6.267123e-01  7.170901e-01
##           301           302           303           304           305
##  5.343227e-01  3.016830e-01  1.339973e+00 -5.908430e-01  1.067702e+00
##           306           307           308           309           310
## -9.856360e-01  1.095619e+00 -1.263814e+00 -1.218097e+00 -1.151715e+00
##           311           312           313           314           315
## -5.431133e-01 -5.358770e-03  9.417346e+00 -3.604035e+00 -5.182923e+00
##           316           317           318           319           320
## -1.804319e+00  3.690326e-01  1.241520e+00 -7.999722e-01 -1.096538e+00
##           321           322           323           324           325
## -3.271750e+00  6.297508e+00 -5.725913e-01 -4.958351e+00  4.106825e+00
##           326           327           328           329           330
##  2.635588e+00  3.825055e+00 -7.252921e-01  6.543581e+00  3.924167e+00
##           331           332           333           334           335
## -5.257229e-01 -1.820124e+00  2.903748e+00  2.157005e+00 -8.850454e-01
##           336           337           338           339           340
##  2.781693e+00 -4.846003e+00  1.004638e+00  4.006118e+00 -2.956136e+00
##           341           342           343           344           345
##  6.645179e-01  4.064631e-02  7.771685e-01 -1.367425e+00 -1.072872e+00
##           346           347           348           349           350
##  2.076804e-01 -8.631973e-01  1.666882e-01 -3.010769e-01 -2.046185e+00
```

```
##           351           352           353           354           355
##  1.440270e+00 -1.029346e+00  1.773920e+00 -1.635601e-02  4.275366e-01
##           356           357           358           359           360
## -1.837850e+00 -1.018010e+00  2.246978e-01 -5.140241e-01  1.884274e-01
##           361           362           363           364           365
##  2.954709e-01 -1.836954e+00  1.896824e+00 -1.038357e+00  1.740679e+00
##           366           367           368           369           370
##  4.021842e-01 -3.887036e-01 -8.103976e-01 -7.299453e-01 -1.246032e+00
##           371           372           373           374           375
## -1.124070e+00  5.246545e-01 -5.250604e-01 -8.208021e-01  1.013872e+00
##           376           377           378           379           380
## -1.504442e+00 -7.917160e-01 -2.755419e-01  3.027853e-01 -1.685774e+00
##           381           382           383           384           385
## -2.459933e+00 -9.931267e-01 -2.277492e-01 -8.492708e-01  2.301905e+00
##           386           387           388           389           390
##  8.386736e-01  7.345202e-01 -7.670846e-01  3.687404e-01  3.940702e+00
##           391           392           393           394           395
##  5.572396e-01 -1.747386e+00 -3.334736e-01 -1.996618e+00  1.167243e+00
##           396           397           398           399           400
## -2.822680e-01  2.332753e-01  5.671448e-02  2.844604e+00 -1.470068e+00
##           401           402           403           404           405
##  1.711028e+00 -3.206064e-02  2.056051e+00 -1.699995e+00 -1.783823e+00
##           406           407           408           409           410
## -5.513772e-01 -4.406567e-01  3.191259e-01  2.114156e+00  2.403668e+00
##           411           412           413           414           415
##  6.574655e-01  2.587840e-01  2.245756e+00  2.939192e+00 -1.062399e-01
##           416           417           418           419           420
## -4.048134e-01  9.899397e-03 -2.633088e+00  9.053820e-01  1.704048e+00
##           421           422           423           424           425
##  1.874298e+01 -1.566487e+01 -9.392964e-01  1.080471e+01  4.564343e+00
##           426           427           428           429           430
## -4.016532e+00  5.501257e+00  2.560560e+00 -3.598928e+00  4.436585e+00
##           431           432           433           434           435
## -5.225061e-01  1.631393e+00 -5.746328e+00  6.070748e+00  4.291440e+00
##           436           437           438           439           440
## -1.800071e+00 -2.363588e+00  5.471511e+00 -1.379148e-01 -1.718597e+00
##           441           442           443           444           445
## -4.142131e-01  6.463034e-02  1.163865e-01 -4.265394e-01 -3.336058e-01
##           446           447           448           449           450
## -6.018256e-02  4.624872e-01 -1.505189e+00 -7.548395e-01 -2.603048e-01
##           451           452           453           454           455
##  5.286605e-02 -1.325935e+00 -1.553692e-01 -1.260359e+00 -4.696977e-01
##           456           457           458           459           460
##  6.212835e-04  6.653705e+00 -2.074317e+00  6.241145e+00  2.196480e+00
##           461           462           463           464           465
##  5.430187e+00  7.796422e-01  6.605052e+00 -2.662820e+00  7.508625e-01
##           466           467           468           469           470
##  4.413649e+00 -3.359051e-02  1.477538e+00 -4.983940e+00  3.072012e+00
##           471           472           473           474           475
```

```
##  3.022200e+00 -1.517536e+00 -1.498545e+00 -1.795509e-01 -6.383783e-01
##           476           477           478           479           480
## -2.353444e+00  1.794499e-01 -2.751188e-01 -1.097666e+00 -1.548998e-01
##           481           482           483           484           485
##  1.386805e+00  1.148444e+00 -1.297474e+00 -1.620297e+00  6.853838e-01
##           486           487           488           489           490
##  2.590622e+00 -1.569399e+00 -2.072716e+00  4.324528e-01 -1.484707e+00
##           491           492           493           494           495
##  7.212005e-01 -3.548158e-01 -2.425752e+00 -1.015643e+00 -2.711111e+00
##           496           497           498           499           500
## -5.479198e+00  2.333386e+00 -1.496542e+00 -1.134472e+00 -4.172210e+00
##           501           502           503           504           505
##  7.855515e-01 -4.259349e-01 -1.584229e+00 -1.207435e+00 -1.779355e+00
##           506           507           508           509           510
## -4.930312e+00 -7.064784e-01 -2.887426e+00  7.104066e-01 -1.759053e-01
##           511           512           513           514           515
##  4.354471e-02 -6.978218e-01 -1.403686e+00  6.596620e-01 -1.202799e+00
##           516           517           518           519           520
## -2.402570e+00  7.414144e-01 -2.755111e+00  1.192628e+00 -1.998064e+00
##           521           522           523           524           525
##  1.146594e+00 -3.152520e-01  8.343284e-01 -7.381751e-01 -4.063524e-01
##           526           527           528           529           530
## -1.317155e-01 -1.067565e+00 -1.952451e-01 -2.523810e+00 -1.674411e+00
##           531           532           533           534           535
##  7.461820e-01 -2.590296e+00 -3.066226e+00 -2.291837e+00 -2.151662e-01
##           536           537           538           539           540
## -1.113515e+00 -1.561030e+00 -7.466895e-01 -1.152285e+00 -1.655815e+00
##           541           542           543           544           545
## -1.256117e-01 -2.280585e+00  1.292042e+00  7.520874e-01  2.581620e+00
##           546           547           548           549           550
## -5.551725e+00  5.757414e+00  5.733630e-01 -6.755702e+00 -2.565328e+00
##           551           552           553           554           555
##  6.987547e+00 -1.966302e+00 -1.228032e+00  1.691782e+00  1.757891e+00
##           556           557           558           559           560
## -3.402261e+00  3.475566e+00  3.540563e+00 -1.207434e+00 -2.922079e+00
##           561           562           563           564           565
## -4.755097e-01 -1.513350e+00 -5.359651e-01  1.049038e+00  1.882215e+00
##           566           567           568           569           570
##  6.455625e-02  1.540887e+00  4.201898e-01  5.529202e-01  2.381241e+00
##           571           572           573           574           575
## -2.463932e-01 -1.447664e+00 -5.598640e-01 -5.444190e-01 -2.351989e-01
##           576           577           578           579           580
##  1.722455e-01 -2.097366e-01  3.746331e-01  1.729714e+00 -1.010363e+00
##           581           582           583           584           585
## -1.708439e-01  5.772990e+00 -1.683799e+00 -2.331676e+00 -1.545343e+00
##           586           587           588           589           590
## -1.205286e+00  1.217740e+00 -1.156070e-01  1.910982e+00  2.335123e+00
##           591           592           593           594           595
##  4.991477e-01 -6.720140e-01  1.096180e+00  3.201044e+00  8.954485e-01
```

```
##           596           597           598           599           600
## -1.489244e+00 -1.025581e+00 -2.429888e+00  9.096954e-01  1.905656e+00
##           601           602           603           604           605
##  5.138995e-01  1.223231e+00  1.362982e+00 -2.110169e+00 -8.085653e-01
##           606           607           608           609           610
##  2.957285e+00 -1.944285e-01 -1.148098e+00 -7.679775e-01 -1.070536e+00
##           611           612           613           614           615
##  1.474376e-01  4.027930e-01  1.124576e+00  1.126745e-01  3.643353e-01
##           616           617           618           619           620
## -2.750770e+00  1.382412e+00  3.621281e-01  5.255556e+00 -3.161284e+00
##           621           622           623           624           625
## -2.771056e+00 -2.310173e+00  1.392632e+00  1.830428e+00  1.404850e+00
##           626           627           628           629           630
## -1.532762e-01  1.561730e+00 -2.712833e-02 -1.050204e+00  3.505023e+00
##           631           632           633           634           635
##  1.916705e+00 -1.301108e+00 -5.675078e-01 -2.144993e+00  1.388530e+00
##           636           637           638           639           640
##  1.782176e-02 -2.061727e+00 -4.054509e-01  6.844244e-01 -2.198611e+00
##           641           642           643           644           645
## -1.747207e+00 -2.048940e+00 -7.328869e-01 -1.528461e+00 -1.897718e+00
##           646           647           648           649           650
## -6.715284e-01 -6.991226e-01 -1.441756e+00 -1.198299e+00 -1.641095e+00
##           651           652           653           654           655
##  1.344179e-01 -1.420299e+00 -5.409352e-01 -1.529818e+00 -1.562386e-01
##           656           657           658           659           660
## -1.234401e+00 -1.949843e+00 -6.750100e-01 -3.964842e-01 -6.667736e-01
```



```
##
##  Shapiro-Wilk normality test
##
## data:  res_m
## W = 0.90299, p-value < 2.2e-16
```

Residual Analysis for AR_DLM_solar_35:

1. The data points are below the line at the start and below the line at the end of the trend. Randomness is seen to some extent. So, we cannot decide anything at this stage. Further analysis is required.
2. From normal distribution curve, the distribution is almost symmetric.
3. The data at the tails is deviated more leaving some part on the line suggesting there is some normality in the trend.
4. There are significant lags in Autocorrelation plot suggesting that the stochastic component is not white noise.
5. p - value (~ 0.01) from Shapiro-Wilk normality test is < 0.05 and therefore, it is statistically significant. Therefore, Null hypothesis is rejected. Suggesting some normality.

Now let us fit AR_DLM_solar_45 model.

```
# ARDLM model
AR_DLM_solar_45 = ardlDlm(x = as.vector(x) , y = as.vector(y), p = 4 , q = 5)
summary(AR_DLM_solar_45)
```

```
##
## 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
## -15.5901  -1.3826  -0.2867   1.0526  18.5709
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.46103    0.43729   5.628 2.72e-08 ***
## X.t         -0.61439    0.54768  -1.122 0.262364
## X.1          0.78469    0.77617   1.011 0.312409
## X.2          1.28470    0.79026   1.626 0.104509
## X.3          0.80457    0.77946   1.032 0.302355
## X.4         -1.20750    0.55570  -2.173 0.030148 *
## Y.1          1.27195    0.03849  33.049  < 2e-16 ***
## Y.2         -0.01868    0.06249  -0.299 0.765112
## Y.3         -0.40480    0.06019  -6.725 3.87e-11 ***
## Y.4         -0.23201    0.06221  -3.729 0.000209 ***
```

```
## Y.5          0.21746    0.03772   5.766 1.26e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.546 on 644 degrees of freedom
## Multiple R-squared:  0.9338, Adjusted R-squared:  0.9328
## F-statistic: 908.5 on 10 and 644 DF,  p-value: < 2.2e-16
```

Hypotheses:

H_0: The data doesn't fit the Autoregressive distributed lag model.

H_A: The data fits the Autoregressive distributed lag model.

Interpretations:

- F - statistic is 908.5
  R - squared is 0.9338
- Adjusted R - squared is 0.9328
- Degrees of freedom - DF are (10, 644)
- p - value (~ 0.01) is < 0.05 and therefore, it is statistically significant. Therefore, Null hypothesis is rejected. Hence, the model fits the Autoregressive distributed lag model.

This model suggests that there is only 93.28% of data variance. Suggesting that the model explains only 93.28% of the trend. Which implies that the model shows some trend.

Now let us perform residual analysis.

Residual analysis

```
res_analysis(residuals(AR_DLM_solar_45))

## Time Series:
## Start = 6
## End = 660
## Frequency = 1
##            6            7            8            9           10
##  -2.066124094  -1.262968376  -2.381225729  -3.603476533  -0.599038712
##           11           12           13           14           15
##  -0.079953784  -2.683995905   5.546575423  -3.302787162   3.773099942
##           16           17           18           19           20
##   4.854843122   6.988046258   0.620595467   3.244204011  -0.724695761
##           21           22           23           24           25
##  -2.776599318   0.304591363   2.955370533   1.293367080  -4.028820843
##           26           27           28           29           30
##   0.003894827   2.431428961  -1.951022383   4.167971938  -3.148131431
##           31           32           33           34           35
##  -1.119208366  -2.303230614  -3.577865713   0.939323323  -0.514233278
##           36           37           38           39           40
##  -1.016646121  -2.239444849   0.440449828  -0.797511843  -0.244662697
##           41           42           43           44           45
##   1.814101694  -0.582650692  -2.360382470  -0.455073842  -0.954394715
```

```
##            46            47            48            49            50
##  -2.583980175  -0.045602769  -0.427217741  -2.130607492   0.358632003
##            51            52            53            54            55
##   1.277402397   1.497803341   0.316522066  -0.734214572   0.786532322
##            56            57            58            59            60
##  -2.960110262  -3.471979200   0.380839165  -0.466030992  -0.794817953
##            61            62            63            64            65
##  -3.631052104  -1.480478461  -1.232267318  -4.981204054  -3.076476952
##            66            67            68            69            70
##  -1.566923624  -3.029423245  -1.328024692  -1.003535484  -2.539475944
##            71            72            73            74            75
##  -3.508281400   7.923067812 -13.293312950  -0.796420308   4.356404035
##            76            77            78            79            80
##   4.210987922  -1.254732503  -0.613393133   1.023273437  -1.996626162
##            81            82            83            84            85
##  -3.604632652  -1.220659128   1.202884415  -0.409212144  -0.454413775
##            86            87            88            89            90
##  -0.290946413   2.670633093  -2.226405460   1.923751825  -1.519735304
##            91            92            93            94            95
##   0.365098273  -1.868208821  -1.057342074  -0.906177354  -2.255912184
##            96            97            98            99           100
##   0.657243874   0.227018247  -0.218633943   0.223277575   0.917352734
##           101           102           103           104           105
##  -0.243302002  -1.465421104  -0.359590474  -1.007715572  -2.078335717
##           106           107           108           109           110
##  -1.861381275  -0.536919688   0.093855000   7.036369865  -0.636078457
##           111           112           113           114           115
##   3.636862356   2.028678775   7.642041393   0.893486118   0.269717250
##           116           117           118           119           120
##   1.077080265   0.738577821  -0.714244320   0.192913413   3.567956525
##           121           122           123           124           125
##  -4.867552860   4.600114176   2.825092101  -1.401309680  -1.625324800
##           126           127           128           129           130
##   1.638913536  -0.528386283  -2.918108113  -1.592180100  -2.237270579
##           131           132           133           134           135
##  -1.117393130   0.996698311   8.784363579  -1.087755979   1.211424574
##           136           137           138           139           140
##   2.923130000   6.575444834   2.926560056   1.158699055   0.534074588
##           141           142           143           144           145
##  -1.133181583   1.263306749   0.303264387   2.917385229  -1.701802507
##           146           147           148           149           150
##   2.341322207   2.928863672  -1.087985038  -0.235053983  -2.560555844
##           151           152           153           154           155
##   1.312512736  -0.087102763  -1.635608987  -2.554962998   0.833092942
##           156           157           158           159           160
##  -0.919184603  -0.065096500   1.536708231   1.559804683  -2.878640218
##           161           162           163           164           165
##   2.606602831   0.535567850  -1.387458425  -4.859909441   2.079162387
##           166           167           168           169           170
```

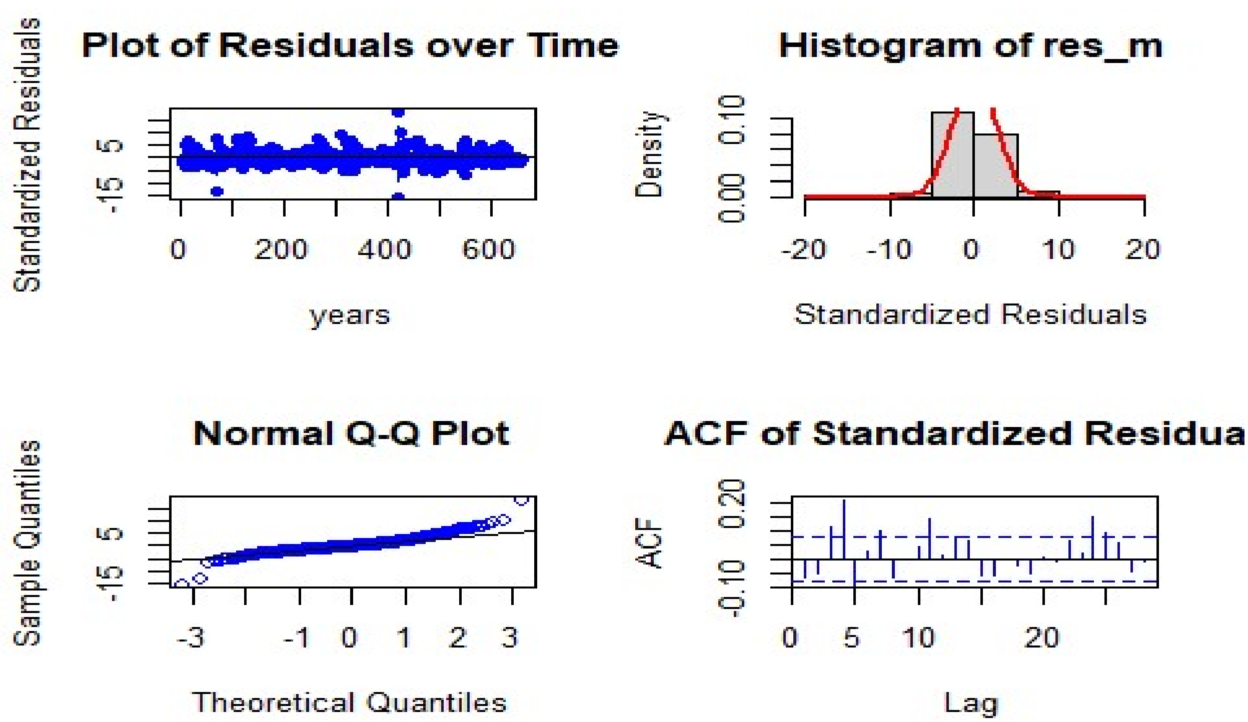
```
## -1.695339740 -0.667003276 -0.960837513  1.781974699  1.098100526
##          171          172          173          174          175
##  2.065173477 -2.769451882  2.147091878  4.513319458 -3.142561306
##          176          177          178          179          180
## -1.981394693 -2.474441183 -2.096738686  1.582526110  0.386152651
##          181          182          183          184          185
## -1.171966874  2.107491183  3.293437050 -2.551732996  0.923467312
##          186          187          188          189          190
##  3.447097289 -2.947824547 -1.897067322 -0.121280508 -1.383734989
##          191          192          193          194          195
##  0.137533688 -1.080138513  0.630027960  0.763016091  1.187061199
##          196          197          198          199          200
## -0.382442063  0.870966068 -1.089933314  1.121450685 -1.968143874
##          201          202          203          204          205
## -2.128485031 -0.545355735  0.433984947 -0.815528632 -0.839587136
##          206          207          208          209          210
## -2.021487511  0.096942161  0.530339405  1.161193641 -0.526852587
##          211          212          213          214          215
## -0.611497915 -2.231928504 -2.725720923 -0.230124743  0.211069658
##          216          217          218          219          220
## -0.897081112 -3.455530971  0.662574411  1.757520656  1.352800358
##          221          222          223          224          225
## -0.633254697  0.770435597 -0.190165785 -2.685350822 -1.995899241
##          226          227          228          229          230
## -1.663087624 -0.499379351  0.972015798  2.015005873 -1.166289979
##          231          232          233          234          235
## -0.416395494  0.966046197  1.376269153 -0.216943289  0.495955215
##          236          237          238          239          240
## -1.448798164 -2.655126057 -1.488892404  0.737613921  0.271352310
##          241          242          243          244          245
##  2.720651486 -0.933459620  0.489843906 -1.658026989  3.336043591
##          246          247          248          249          250
## -0.475527900 -0.694320150 -0.594084277 -0.979559100 -0.397924708
##          251          252          253          254          255
## -1.637060139  1.057691611 -0.402010316  3.148880963 -0.547967723
##          256          257          258          259          260
##  1.164970739  1.018511007  1.098514807 -0.488258901 -0.009938073
##          261          262          263          264          265
## -0.999272689 -2.566140602  0.310068155  1.510708503  7.848053402
##          266          267          268          269          270
## -3.268063844  4.012538782  4.720146486  6.412764293  1.631818537
##          271          272          273          274          275
##  4.660582860 -0.999658439 -1.687289407  0.074530679  2.167381113
##          276          277          278          279          280
##  5.025195096 -4.585888649  1.937032716  3.483287997 -0.389962662
##          281          282          283          284          285
##  0.780126621 -0.797181149  1.982573682 -2.656158515 -1.960417128
##          286          287          288          289          290
## -1.759102709  0.501659361  0.872550217  0.708182832  1.356074800
```

```
##           291           292           293           294           295
##   1.033716747   0.752969412   1.215201848   1.778614257  -0.492664944
##           296           297           298           299           300
##  -1.071837068   0.037030379  -2.447062986   0.594539390   0.690673044
##           301           302           303           304           305
##   0.403030762   0.258384346   1.251561161  -0.634418737   1.061468651
##           306           307           308           309           310
##  -1.207212198   0.895379048  -1.342513248  -1.109192200  -1.148033958
##           311           312           313           314           315
##  -0.630666027  -0.015446227   9.121318478  -3.636838119  -5.242815956
##           316           317           318           319           320
##  -1.854435098   0.295101355   1.184195815  -0.969650731  -1.108143374
##           321           322           323           324           325
##  -3.013706290   6.217439531  -0.649087320  -5.067814368   3.591347058
##           326           327           328           329           330
##   2.729722615   4.192127925  -1.287029228   6.535734207   4.297674894
##           331           332           333           334           335
##  -0.444750084  -1.532823983   3.365807710   2.223163206  -0.793505403
##           336           337           338           339           340
##   3.028881529  -5.017956987   1.005044185   4.119388745  -2.482739551
##           341           342           343           344           345
##   1.078825678   0.075525087   0.962542334  -1.351141916  -1.072639539
##           346           347           348           349           350
##   0.094181083  -0.871929894   0.107819005  -0.177862994  -2.456440140
##           351           352           353           354           355
##   1.419302203  -0.472198386   1.975262112   0.068531627   0.432452718
##           356           357           358           359           360
##  -1.803824392  -0.987068116   0.179139064  -0.546325840   0.216495992
##           361           362           363           364           365
##   0.235026952  -2.119324278   1.615710278  -0.312720481   1.811780807
##           366           367           368           369           370
##   0.236793881  -0.574259365  -0.863221027  -0.724648712  -1.274653892
##           371           372           373           374           375
##  -1.097275730   0.453776872  -0.637476897  -0.796837118   0.953967426
##           376           377           378           379           380
##  -1.614887725  -0.724862368  -0.194062089   0.199474103  -1.764327147
##           381           382           383           384           385
##  -2.405898908  -0.964137338  -0.267922954  -0.878547141   2.154412181
##           386           387           388           389           390
##   0.916815179   0.319525174  -0.596360819   0.466471194   3.872952709
##           391           392           393           394           395
##   0.438584047  -1.873141566  -0.293924159  -1.963352693   1.129120720
##           396           397           398           399           400
##  -0.256861910   0.107707109   0.006872407   2.641601696  -1.065100311
##           401           402           403           404           405
##   1.629743399  -0.153860955   1.847424234  -1.993058141  -1.759269265
##           406           407           408           409           410
##  -0.526520308  -0.500199398   0.276544202   1.922415912   2.438761566
##           411           412           413           414           415
```

```
##   0.497009146   0.180205536   2.186521702   2.795468846  -0.292526687
##           416           417           418           419           420
##  -0.605553033  -0.018832805  -2.594634179   0.826771240   1.750717636
##           421           422           423           424           425
##  18.570862426 -15.590138094  -0.981299915  10.326685879   4.687702861
##           426           427           428           429           430
##  -3.885637349   5.531287979   2.578239386  -3.515815516   4.311517783
##           431           432           433           434           435
##  -0.540425314   1.715526838  -5.863315330   6.010794920   4.268265509
##           436           437           438           439           440
##  -1.448997935  -2.341420446   5.485451000  -0.097697524  -1.744571753
##           441           442           443           444           445
##  -0.448845511  -0.039682873   0.047197285  -0.377427722  -0.494629254
##           446           447           448           449           450
##  -0.081737898   0.587924527  -1.742924063  -0.487652535  -0.048683076
##           451           452           453           454           455
##   0.179544940  -1.226871011  -0.038910544  -1.363173163  -0.557849224
##           456           457           458           459           460
##   0.055665498   6.461404738  -2.089624264   5.701388510   2.782821859
##           461           462           463           464           465
##   5.520887018   0.692080388   6.543280122  -2.688305067   0.816280551
##           466           467           468           469           470
##   4.338972412  -0.016095399   1.584672450  -5.046752793   3.111048919
##           471           472           473           474           475
##   2.866925313  -1.167192572  -1.550337993  -0.179244526  -0.710222074
##           476           477           478           479           480
##  -2.331971254   0.307782656  -0.361968930  -1.202140388  -0.172155217
##           481           482           483           484           485
##   1.155691016   1.078127998  -1.324293329  -1.536619163   0.601724421
##           486           487           488           489           490
##   2.746194021  -1.711705918  -1.641574973   0.780288826  -1.525715594
##           491           492           493           494           495
##   0.689991625  -0.209908042  -2.741408131  -0.969661723  -2.081372178
##           496           497           498           499           500
##  -6.585358566   3.489176471  -1.106786897  -0.824594476  -4.052785385
##           501           502           503           504           505
##   0.824348875  -0.542361787  -1.613561078  -1.027250774  -1.910339816
##           506           507           508           509           510
##  -5.103419797  -1.264259440  -1.646195392   1.462381784   0.345754494
##           511           512           513           514           515
##   0.294021457  -0.576419223  -1.428065201   0.528505235  -1.252111354
##           516           517           518           519           520
##  -2.257003376   0.585089954  -3.033796036   0.808586102  -0.810518860
##           521           522           523           524           525
##   1.390517345  -0.124804233   0.862083072  -0.795813767  -0.435746061
##           526           527           528           529           530
##  -0.287504097  -1.103429896  -0.149391049  -2.615640526  -1.648462889
##           531           532           533           534           535
##   0.644682896  -2.570763559  -2.533323763  -1.882100891   0.304946192
```

```
##          536          537          538          539          540 
## -1.154554838 -0.814151548 -0.867944442 -0.979563551 -1.867336332 
##          541          542          543          544          545 
## -0.137515749 -2.284468805  0.748184010  1.955128388  2.714297175 
##          546          547          548          549          550 
## -5.589121567  5.540794271  0.660269438 -6.759748497 -2.544842444 
##          551          552          553          554          555 
##  6.932240842 -1.807291568 -1.430610489  1.509251812  1.513695116 
##          556          557          558          559          560 
## -2.783957568  3.235185551  3.382491124 -1.381074162 -3.229998576 
##          561          562          563          564          565 
## -0.480743737 -1.388170777 -0.486616128  1.047443372  1.673893995 
##          566          567          568          569          570 
##  0.184297286  1.766518776 -0.286696032  0.617795163  2.458660922 
##          571          572          573          574          575 
## -0.306558718 -1.505903718 -0.552282361 -0.556540392 -0.236662583 
##          576          577          578          579          580 
##  0.188575969 -0.132205392 -0.106294030  1.696074269 -0.208005862 
##          581          582          583          584          585 
## -0.253752948  5.578978409 -1.851936188 -2.553139773 -1.393998184 
##          586          587          588          589          590 
## -1.162320637  1.218251243 -0.134575425  1.678413460  2.388561840 
##          591          592          593          594          595 
##  0.459237206 -0.780995034  1.013537621  3.052568921  0.679615813 
##          596          597          598          599          600 
## -1.616737155 -0.919350630 -2.421309931  0.854283503  1.897491330 
##          601          602          603          604          605 
##  0.373465816  1.274502464  1.612927736 -2.678546464 -0.749029626 
##          606          607          608          609          610 
##  3.182945213 -0.067735578 -0.976445405 -0.531640920 -1.097861274 
##          611          612          613          614          615 
##  0.089782851  0.540410348  0.904499737  0.051192220  0.138783219 
##          616          617          618          619          620 
## -2.197965975  1.558613431  0.342564386  5.261119104 -3.225157279 
##          621          622          623          624          625 
## -2.497996897 -2.370130986  1.364750229  1.897148841  1.251348663 
##          626          627          628          629          630 
## -0.226346649  1.519913895  0.091329052 -0.905523158  3.407517269 
##          631          632          633          634          635 
##  1.956748817 -1.381429876 -0.715681786 -2.117352890  1.336272340 
##          636          637          638          639          640 
##  0.131078310 -2.203117436 -0.459097527  0.655658576 -1.987628580 
##          641          642          643          644          645 
## -1.734012686 -1.912327996 -0.819282082 -1.214852328 -1.625459399 
##          646          647          648          649          650 
## -0.612233320 -0.729077117 -1.404630150 -1.549577748 -1.787790708 
##          651          652          653          654          655 
##  0.255078061 -0.997020050 -0.680401166 -1.468290017 -0.292976263 
```

```
##          656          657          658          659          660
##  -1.013598966  -1.744421978  -0.866645063  -0.490392222  -0.685137501
```




```
##
##  Shapiro-Wilk normality test
##
## data:  res_m
## W = 0.90356, p-value < 2.2e-16
```

Residual Analysis for AR_DLM_solar_45:

1. The data points are below the line at the start and below the line at the end of the trend. Randomness is seen to some extent. So, we cannot decide anything at this stage. Further analysis is required.
2. From normal distribution curve, the distribution is almost symmetric.
3. The data at the tails is deviated more leaving some part on the line suggesting there is some normality in the trend.
4. There are significant lags in Autocorrelation plot suggesting that the stochastic component is not white noise.
5. p - value (~ 0.01) from Shapiro-Wilk normality test is < 0.05 and therefore, it is statistically significant. Therefore, Null hypothesis is rejected. Suggesting some normality.

Now let us fit AR_DLM_solar_55 model.

```
# ARDLM model
AR_DLM_solar_55 = ardlDlm(x = as.vector(x) , y = as.vector(y), p = 5, q = 5)
summary(AR_DLM_solar_55)

##
## 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
```

Hypotheses:

H_0: The data doesn't fit the Autoregressive distributed lag model.

H_A: The data fits the Autoregressive distributed lag model.

Interpretations:

- F - statistic is 824.9
- R - squared is 0.9338
- Adjusted R - squared is 0.9327
- Degrees of freedom - DF are (11, 643)
- p - value (~ 0.01) is < 0.05 and therefore, it is statistically significant. Therefore, Null hypothesis is rejected. Hence, the model fits the Autoregressive distributed lag model.

This model suggests that there is only 93.27% of data variance. Suggesting that the model explains only 93.27% of the trend. Which implies that the model shows some trend.

Now let us perform residual analysis.

Residual analysis

```
res_analysis(residuals(AR_DLM_solar_55))

## Time Series:
## Start = 6
## End = 660
## Frequency = 1
##            6            7            8            9           10
##  -1.957909312  -1.277958659  -2.327089193  -3.588827147  -0.617244544
##           11           12           13           14           15
##  -0.046054584  -2.635863265   5.588563855  -3.270585941   3.846476573
##           16           17           18           19           20
##   4.837410086   7.153759519   0.658950240   3.215848586  -0.737392989
##           21           22           23           24           25
##  -2.795725194   0.291002448   2.970687513   1.327174606  -4.000595386
##           26           27           28           29           30
##   0.013196779   2.487193893  -1.932094655   4.230288554  -3.120423994
##           31           32           33           34           35
##  -1.138986183  -2.318712740  -3.579125906   0.925741403  -0.484295455
##           36           37           38           39           40
##  -1.001691368  -2.225791093   0.410797689  -0.764594340  -0.200805125
##           41           42           43           44           45
##   1.789092539  -0.533645055  -2.339643366  -0.465239680  -0.960161400
##           46           47           48           49           50
##  -2.590843970  -0.022885016  -0.414624884  -2.111926530   0.340847000
##           51           52           53           54           55
##   1.273623987   1.481027593   0.396844373  -0.730524601   0.758852930
##           56           57           58           59           60
##  -2.965421981  -3.516119858   0.394106808  -0.462220512  -0.769548016
##           61           62           63           64           65
##  -3.628550539  -1.478461987  -1.249795054  -4.943261976  -3.101736794
##           66           67           68           69           70
##  -1.533784120  -3.027068597  -1.335912704  -0.978818889  -2.556230346
##           71           72           73           74           75
##  -3.519835953   7.902591051 -13.294563974  -0.854623623   4.352782896
##           76           77           78           79           80
##   4.165669211  -1.163232022  -0.594552935   0.997352878  -2.007953923
##           81           82           83           84           85
##  -3.640601952  -1.219405904   1.216952475  -0.391406767  -0.436354096
##           86           87           88           89           90
##  -0.307555515   2.673451033  -2.251103820   1.941983919  -1.515851148
##           91           92           93           94           95
##   0.339903336  -1.916188934  -1.087722385  -0.899653935  -2.243327109
##           96           97           98           99          100
```

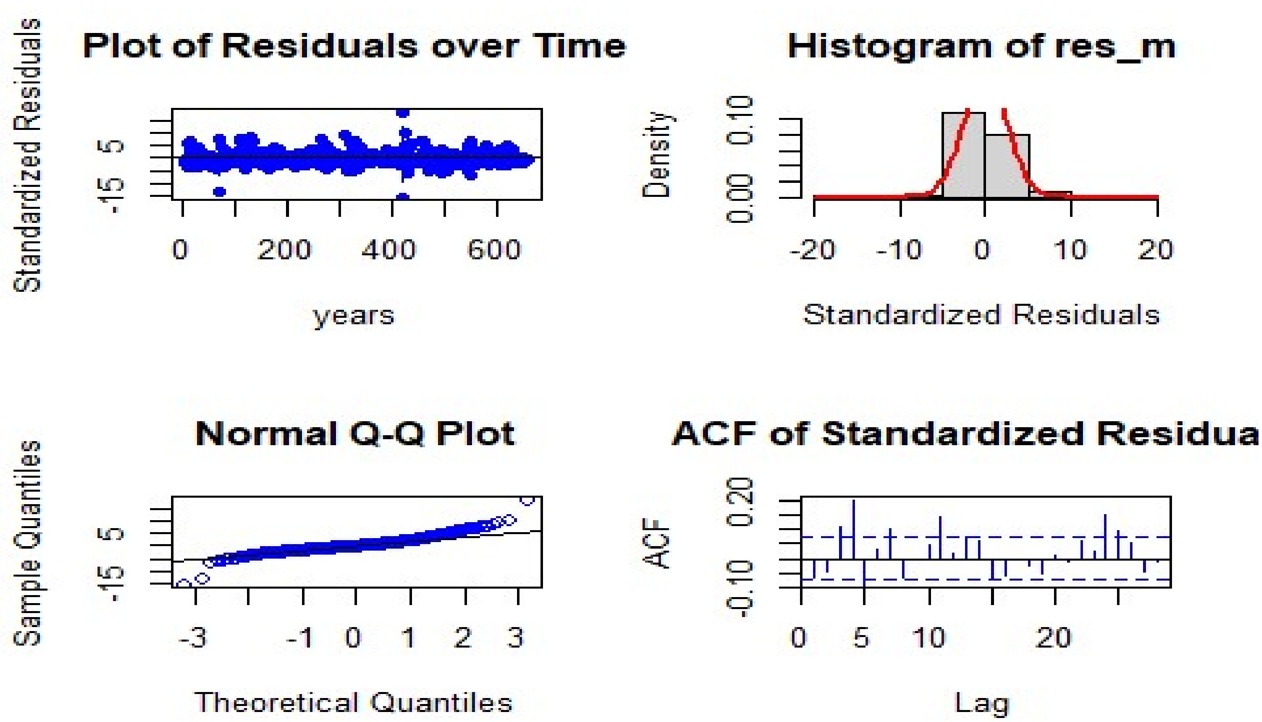
```
##   0.651242748   0.228205788  -0.242706988   0.231158955   0.890283376
##           101           102           103           104           105
##  -0.264603524  -1.452053762  -0.387657528  -1.055516712  -2.084634508
##           106           107           108           109           110
##  -1.862126358  -0.521034935   0.097315992   7.023869773  -0.645416587
##           111           112           113           114           115
##   3.620624359   2.018021829   7.636340429   0.887875005   0.243542473
##           116           117           118           119           120
##   1.021335637   0.706901153  -0.696966691   0.230650255   3.585739066
##           121           122           123           124           125
##  -4.858037808   4.585492087   2.836635682  -1.391118562  -1.606047872
##           126           127           128           129           130
##   1.621773538  -0.544671817  -2.961686759  -1.633441661  -2.219421966
##           131           132           133           134           135
##  -1.078168021   0.995513418   8.779917787  -1.106092177   1.219332445
##           136           137           138           139           140
##   2.919022665   6.548545999   2.920352669   1.135947747   0.485853740
##           141           142           143           144           145
##  -1.160777727   1.277375555   0.339855690   2.931062852  -1.695689515
##           146           147           148           149           150
##   2.336029138   2.936843060  -1.092168324  -0.201090618  -2.582873290
##           151           152           153           154           155
##   1.274393784  -0.123701539  -1.659821150  -2.538568514   0.842519624
##           156           157           158           159           160
##  -0.924832314  -0.070991282   1.478525336   1.562997464  -2.867020708
##           161           162           163           164           165
##   2.597622116   0.506660344  -1.399950209  -4.918721290   2.029420151
##           166           167           168           169           170
##  -1.658552021  -0.652582637  -0.945754404   1.779225935   1.043061844
##           171           172           173           174           175
##   2.087242219  -2.768336180   2.121899726   4.484584364  -3.153769717
##           176           177           178           179           180
##  -2.021051937  -2.527570177  -2.095457208   1.617533769   0.415610338
##           181           182           183           184           185
##  -1.126596497   2.073327758   3.323442253  -2.537988893   0.886872039
##           186           187           188           189           190
##   3.411117972  -2.934982505  -1.953023006  -0.142975619  -1.359320197
##           191           192           193           194           195
##   0.176209574  -1.066900095   0.641983352   0.704799478   1.214997519
##           196           197           198           199           200
##  -0.337444644   0.791668083  -1.046134250   1.092660716  -1.960723295
##           201           202           203           204           205
##  -2.133031646  -0.557127104   0.425929223  -0.800244323  -0.843518654
##           206           207           208           209           210
##  -2.028239879   0.052350960   0.531349517   1.217476618  -0.497689697
##           211           212           213           214           215
##  -0.609722062  -2.254471810  -2.711904123  -0.243273902   0.223554116
##           216           217           218           219           220
##  -0.872190398  -3.474939045   0.648205857   1.752783725   1.324642689
```

```
##         221          222          223          224          225 
##  -0.527004271  0.764214667 -0.222353954 -2.730553750 -2.041750670 
##         226          227          228          229          230 
##  -1.662612551 -0.502049070  0.957848507  2.026879243 -1.198376595 
##         231          232          233          234          235 
##  -0.426089606  0.931367910  1.352263490 -0.219506469  0.474148536 
##         236          237          238          239          240 
##  -1.499437813 -2.676286376 -1.491701471  0.749067542  0.270267909 
##         241          242          243          244          245 
##   2.717197443 -0.943622261  0.479016333 -1.672665902  3.309997251 
##         246          247          248          249          250 
##  -0.472769011 -0.710651774 -0.633120849 -1.002359078 -0.413212581 
##         251          252          253          254          255 
##  -1.607217413  1.045418609 -0.413336724  3.123164234 -0.552648109 
##         256          257          258          259          260 
##   1.137809661  1.020441579  1.098945295 -0.514461690 -0.052822190 
##         261          262          263          264          265 
##  -1.028613691 -2.555823564  0.326872617  1.501836348  7.852334091 
##         266          267          268          269          270 
##  -3.282356422  4.000111207  4.733336108  6.365209360  1.643276896 
##         271          272          273          274          275 
##   4.660780956 -1.025597468 -1.704027856  0.081448367  2.168071733 
##         276          277          278          279          280 
##   5.052738842 -4.551361953  1.934898593  3.490023327 -0.422629093 
##         281          282          283          284          285 
##   0.866920844 -0.801022256  1.953275426 -2.688262242 -1.982741488 
##         286          287          288          289          290 
##  -1.762007021  0.501053710  0.879636750  0.714934706  1.329638657 
##         291          292          293          294          295 
##   1.042133880  0.729623029  1.215684393  1.770287455 -0.518169296 
##         296          297          298          299          300 
##  -1.127977359  0.019672612 -2.440259745  0.592688807  0.686145663 
##         301          302          303          304          305 
##   0.399443705  0.236325239  1.242877955 -0.648989062  1.052357910 
##         306          307          308          309          310 
##  -1.207404449  0.855559175 -1.375663155 -1.124555854 -1.130054665 
##         311          312          313          314          315 
##  -0.631720410 -0.030213921  9.119657383 -3.676099631 -5.253902178 
##         316          317          318          319          320 
##  -1.873954323  0.281986533  1.175137304 -0.977510928 -1.138629555 
##         321          322          323          324          325 
##  -3.017274881  6.257381782 -0.654366685 -5.081012220  3.565178368 
##         326          327          328          329          330 
##   2.642937428  4.214467550 -1.218353390  6.435115730  4.305110273 
##         331          332          333          334          335 
##  -0.375832695 -1.519853508  3.411201867  2.307706064 -0.778693424 
##         336          337          338          339          340 
##   3.043379620 -4.972791741  0.968176126  4.119090269 -2.459674922 
##         341          342          343          344          345 
```

```
##   1.157642090   0.145325931   0.968326327  -1.317010132  -1.072455320
##           346           347           348           349           350
##   0.093628028  -0.891398056   0.105459982  -0.187494466  -2.435680884
##           351           352           353           354           355
##   1.344607642  -0.475025800   2.070504461   0.105785426   0.446658804
##           356           357           358           359           360
##  -1.801884864  -0.983990525   0.183830657  -0.553783565   0.210808006
##           361           362           363           364           365
##   0.240251747  -2.129364826   1.563044300  -0.360265080   1.936224911
##           366           367           368           369           370
##   0.251463668  -0.602920429  -0.895340551  -0.735401422  -1.274136157
##           371           372           373           374           375
##  -1.103492857   0.457251053  -0.649008858  -0.817116127   0.956623484
##           376           377           378           379           380
##  -1.624835510  -0.746330274  -0.183880132   0.213237542  -1.781042043
##           381           382           383           384           385
##  -2.421698652  -0.957652191  -0.263823620  -0.884862810   2.148427374
##           386           387           388           389           390
##   0.894455053   0.333781407  -0.668373296   0.493908240   3.890473126
##           391           392           393           394           395
##   0.432235740  -1.892492953  -0.318535068  -1.957554053   1.133103738
##           396           397           398           399           400
##  -0.261639547   0.111796689  -0.014352173   2.631145685  -1.097138698
##           401           402           403           404           405
##   1.697135480  -0.166483798   1.826159541  -2.026019813  -1.812950578
##           406           407           408           409           410
##  -0.524161105  -0.496816319   0.266872882   1.915311707   2.407960131
##           411           412           413           414           415
##   0.505738523   0.151820784   2.171902041   2.787539313  -0.313372893
##           416           417           418           419           420
##  -0.638147303  -0.054840652  -2.599545401   0.830459818   1.738357324
##           421           422           423           424           425
##  18.581237573 -15.595860615  -0.990187599  10.314317441   4.616354015
##           426           427           428           429           430
##  -3.851765697   5.543418822   2.589199790  -3.507988480   4.320689255
##           431           432           433           434           435
##  -0.558149676   1.712739518  -5.847017921   5.980887041   4.265570061
##           436           437           438           439           440
##  -1.448674906  -2.282976904   5.482327420  -0.087924278  -1.736920739
##           441           442           443           444           445
##  -0.456232745  -0.047018063   0.030361798  -0.389253068  -0.486956853
##           446           447           448           449           450
##  -0.110816399   0.583634525  -1.721288657  -0.531708418  -0.003847465
##           451           452           453           454           455
##   0.215871083  -1.204258506  -0.023150671  -1.342712388  -0.577158212
##           456           457           458           459           460
##   0.040163599   6.470780339  -2.113438854   5.694807135   2.695486495
##           461           462           463           464           465
##   5.624017252   0.715380697   6.526869253  -2.690005176   0.807594006
```

```
##          466          467          468          469          470 
##  4.351371408 -0.024347082  1.588983696 -5.028704810  3.092124157 
##          471          472          473          474          475 
##  2.876417118 -1.191743875 -1.491536944 -0.193237063 -0.710499439 
##          476          477          478          479          480 
## -2.344896044  0.308480689 -0.338772045 -1.216971304 -0.191079329 
##          481          482          483          484          485 
##  1.152060241  1.039813438 -1.335546701 -1.543802854  0.613272736 
##          486          487          488          489          490 
##  2.732968820 -1.680809758 -1.668439275  0.852114782 -1.464683434 
##          491          492          493          494          495 
##  0.681863537 -0.214045818 -2.717005346 -1.028547312 -2.074225572 
##          496          497          498          499          500 
## -6.480436646  3.289775994 -0.903329952 -0.757679799 -3.999812855 
##          501          502          503          504          505 
##  0.838740853 -0.533028324 -1.634375841 -1.033633209 -1.881090470 
##          506          507          508          509          510 
## -5.126774940 -1.301359430 -1.744442176  1.674812865  0.478018169 
##          511          512          513          514          515 
##  0.384566251 -0.531802941 -1.408113734  0.523013988 -1.274300041 
##          516          517          518          519          520 
## -2.266904285  0.607636047 -3.059532819  0.756030450 -0.876190705 
##          521          522          523          524          525 
##  1.593488262 -0.080409365  0.894398623 -0.788970472 -0.446830670 
##          526          527          528          529          530 
## -0.292650231 -1.130904066 -0.156835741 -2.608407154 -1.667838812 
##          531          532          533          534          535 
##  0.647066757 -2.587978677 -2.533011941 -1.794395108  0.373547413 
##          536          537          538          539          540 
## -1.063775473 -0.821830211 -0.739901018 -1.001261358 -1.838288816 
##          541          542          543          544          545 
## -0.176252172 -2.285670175  0.743422371  1.861969250  2.924009411 
##          546          547          548          549          550 
## -5.562193023  5.526467301  0.630251279 -6.743272864 -2.553283307 
##          551          552          553          554          555 
##  6.929985159 -1.805243578 -1.404053192  1.470837631  1.481774234 
##          556          557          558          559          560 
## -2.823279889  3.336610373  3.344721184 -1.402936336 -3.261289592 
##          561          562          563          564          565 
## -0.539688929 -1.389359986 -0.465825738  1.056005875  1.674800417 
##          566          567          568          569          570 
##  0.150066593  1.787095029 -0.246801196  0.494652821  2.470292353 
##          571          572          573          574          575 
## -0.290524788 -1.516259578 -0.564528112 -0.556042929 -0.238654650 
##          576          577          578          579          580 
##  0.187982458 -0.128605551 -0.093107816  1.611238737 -0.212532528 
##          581          582          583          584          585 
## -0.116428921  5.563774100 -1.877862645 -2.584024688 -1.436369520 
##          586          587          588          589          590 
```

```
## -1.138799612  1.226029795 -0.132482283  1.675241545  2.350189078
##          591          592          593          594          595
##  0.470220524 -0.788138357  0.992092742  3.039252742  0.658520450
##          596          597          598          599          600
## -1.652996288 -0.943905953 -2.404561582  0.853170845  1.889699250
##          601          602          603          604          605
##  0.374570928  1.250132347  1.622326659 -2.634409666 -0.851436766
##          606          607          608          609          610
##  3.191293748 -0.024545345 -0.953915960 -0.503682461 -1.058000242
##          611          612          613          614          615
##  0.084275922  0.530821261  0.929097310  0.014608042  0.126665207
##          616          617          618          619          620
## -2.237913550  1.650073700  0.374729678  5.258839297 -3.216890104
##          621          622          623          624          625
## -2.513722752 -2.326592463  1.350426114  1.896348110  1.265430362
##          626          627          628          629          630
## -0.251226637  1.505464527  0.084552696 -0.885670955  3.430695603
##          631          632          633          634          635
##  1.944135536 -1.371024903 -0.731452913 -2.144736068  1.338580887
##          636          637          638          639          640
##  0.124161302 -2.183478138 -0.486443204  0.644107315 -1.991860946
##          641          642          643          644          645
## -1.700643245 -1.912841563 -0.798184996 -1.230208413 -1.572542854
##          646          647          648          649          650
## -0.566711680 -0.719485509 -1.409703967 -1.544479379 -1.850576177
##          651          652          653          654          655
##  0.226998147 -0.975807138 -0.608705047 -1.492716131 -0.284361649
##          656          657          658          659          660
## -1.037111793 -1.707158226 -0.832851430 -0.524406784 -0.701154316
```




```
##
##  Shapiro-Wilk normality test
##
## data:  res_m
## W = 0.90353, p-value < 2.2e-16
```

Residual Analysis for AR_DLM_solar_55:

1. The data points are below the line at the start and below the line at the end of the trend. Randomness is seen to some extent. So, we cannot decide anything at this stage. Further analysis is required.
2. From normal distribution curve, the distribution is almost symmetric.
3. The data at the tails is deviated more leaving some part on the line suggesting there is some normality in the trend.
4. There are significant lags in Autocorrelation plot suggesting that the stochastic component is not white noise.
5. p - value (~ 0.01) from Shapiro-Wilk normality test is < 0.05 and therefore, it is statistically significant. Therefore, Null hypothesis is rejected. Suggesting some normality.

Among all the models ARDML models shown better performance and among them AR_DLM_solar_45 shows better trend.

Now let us calculate AIC, BIC and MASE scores and store them in a dataframe to check the better model based on MASE score.

```
attr(Koyck_DLM_solar$model, "class") = "lm"

v_model_name <- c("finite_dlm_solar_rad", "PolyDLM_model_solar",
"Koyck_DLM_solar", "AR_DLM_solar_35", "AR_DLM_solar_45", "AR_DLM_solar_55")

MASE <- MASE(finite_dlm_solar_rad$model, PolyDLM_model_solar$model,
Koyck_DLM_solar$model, AR_DLM_solar_35, AR_DLM_solar_45,
AR_DLM_solar_55)$MASE

aic <- AIC(finite_dlm_solar_rad$model, PolyDLM_model_solar$model,
Koyck_DLM_solar$model, AR_DLM_solar_35$model, AR_DLM_solar_45$model,
AR_DLM_solar_55$model)$AIC

bic <- BIC(finite_dlm_solar_rad$model, PolyDLM_model_solar$model,
Koyck_DLM_solar$model, AR_DLM_solar_35$model, AR_DLM_solar_45$model,
AR_DLM_solar_55$model)$BIC

v_score <- data.frame(v_model_name, MASE, aic, bic)
colnames(v_score) <- c("MODEL_NAME", "MASE", "AIC", "BIC")
v_score

##             MODEL_NAME      MASE      AIC      BIC
## 1 finite_dlm_solar_rad 1.5779955 4602.658 4660.858
## 2  PolyDLM_model_solar 1.6627033 4688.551 4715.478
## 3      Koyck_DLM_solar 1.0324829 3946.476 3964.439
## 4      AR_DLM_solar_35 0.4502885 3098.808 3148.139
## 5      AR_DLM_solar_45 0.4479481 3096.024 3149.839
## 6      AR_DLM_solar_55 0.4479311 3097.877 3156.177
```

Therefore, AR_DLM_solar_45 is the better model.

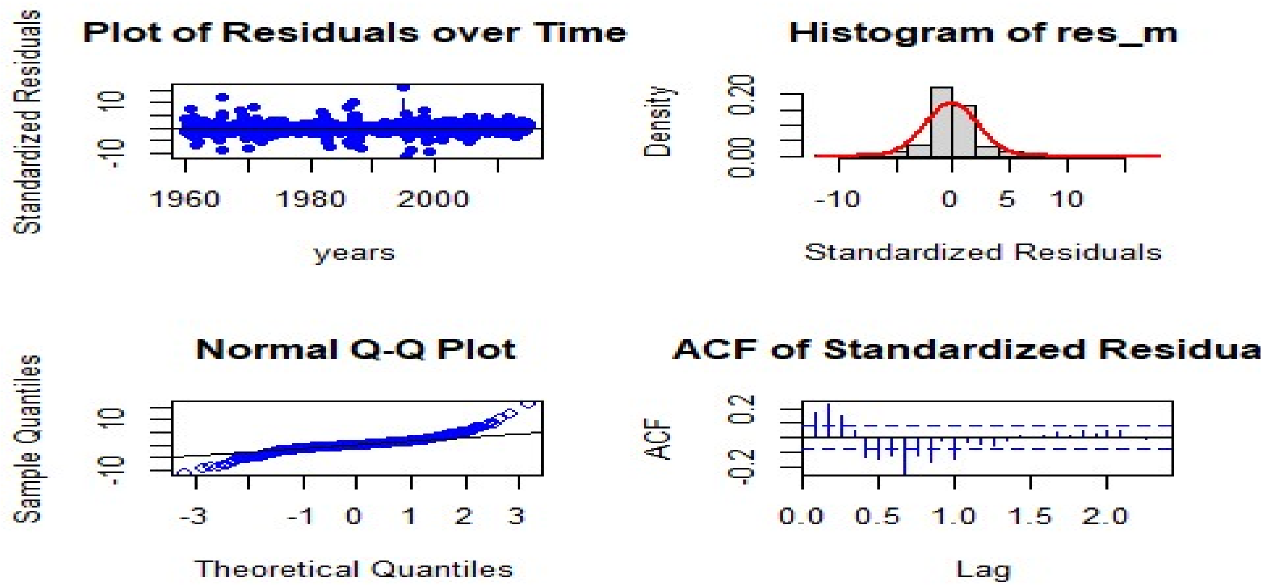
Exponential Smoothing

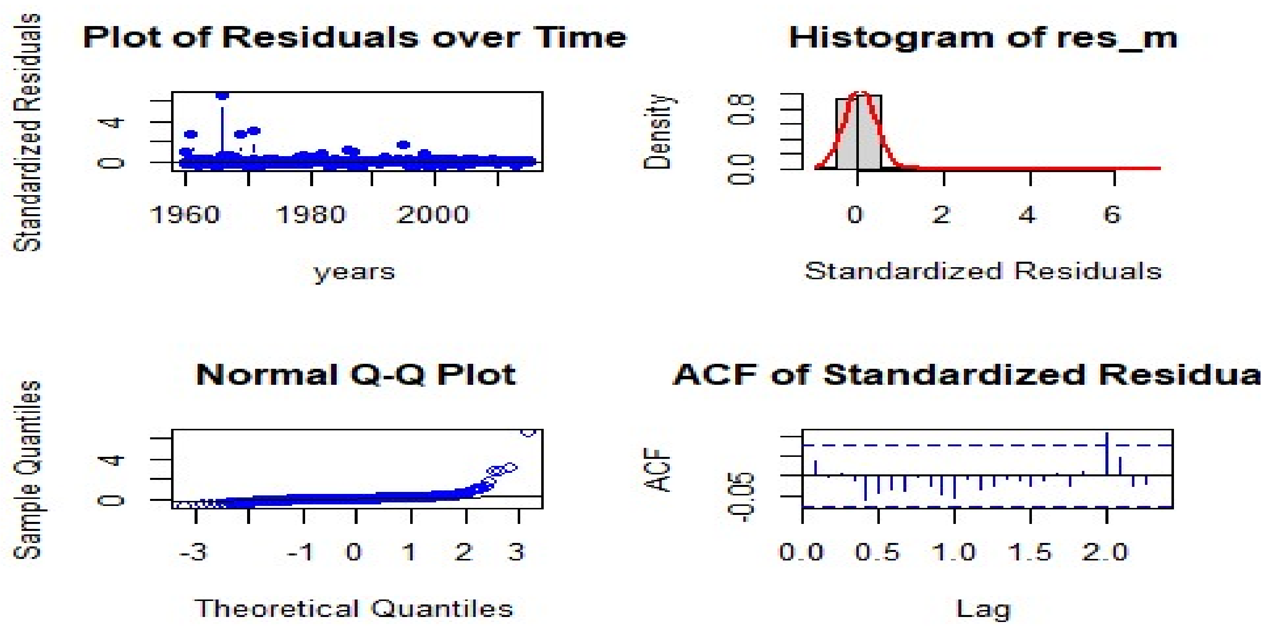
As there is a strong seasonal component in the solar radiation series. Let us consider models that have only additive or multiplicative seasonality.

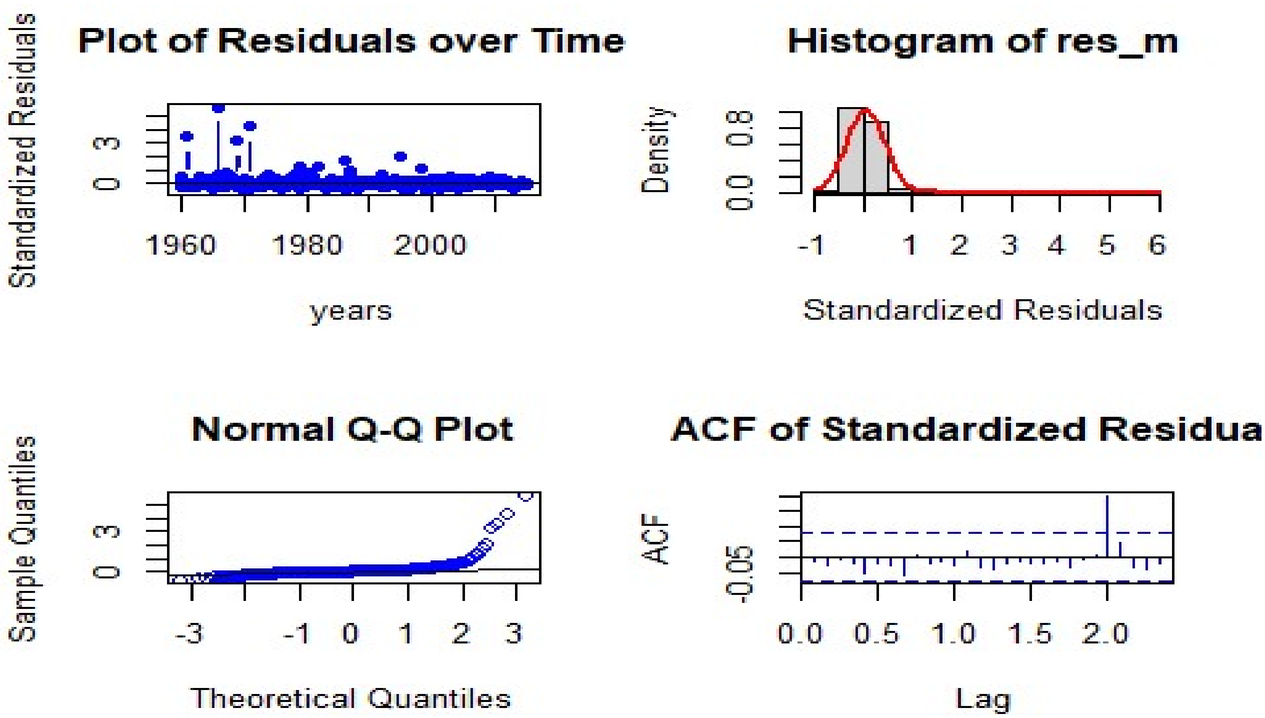
Here we have 6 smoothing methods.

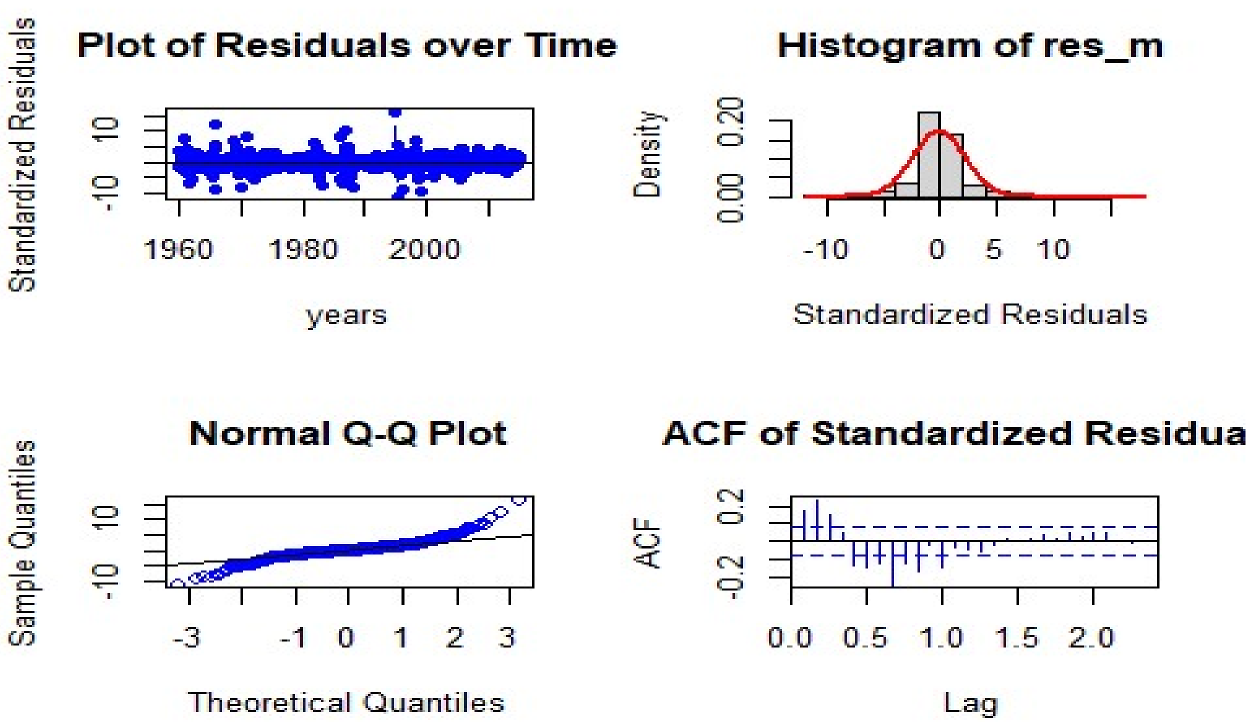
```
exponential = c(T,F)
seasonality <- c("additive", "multiplicative")
damped <- c(T,F)
exp <- expand.grid(exponential, seasonality, damped)
exp <- exp[-c(1,5),]
fit_aic <- array(NA, 6)
fit_bic <- array(NA, 6)
fit_mase <- array(NA, 6)
levels <- array(NA, dim=c(6,3))
for (i in 1:6){
  hw <- hw(v_solar_radiation_TS, exponential = exp[i,1], seasonal =
toString(exp[i,2], damped = exp[i,3]))
```

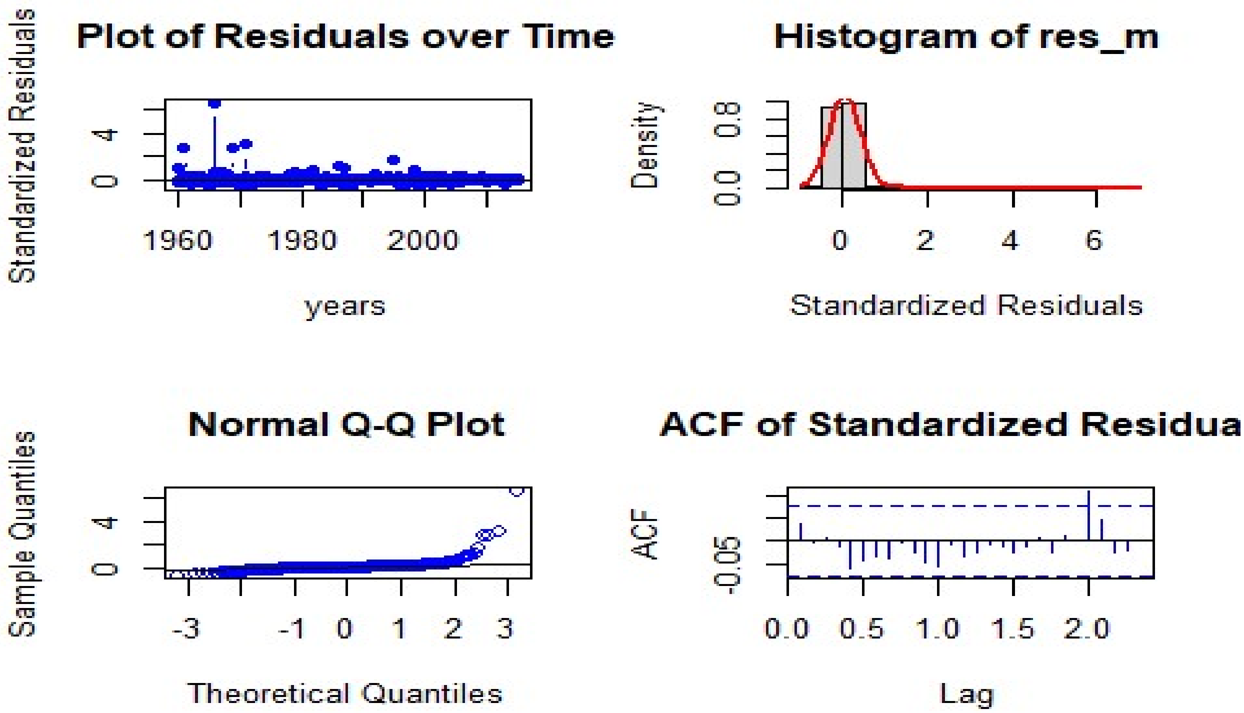
```
  fit_aic[i] <- hw$model$aic
  fit_bic[i] <- hw$model$bic
  fit_mase[i] <- accuracy(hw)[6]
  levels[i,1] <- exp[i,1]
  levels[i,2] <- toString(exp[i,2])
  levels[i,3] <- exp[i,3]
  res_analysis(residuals(hw))
}
```

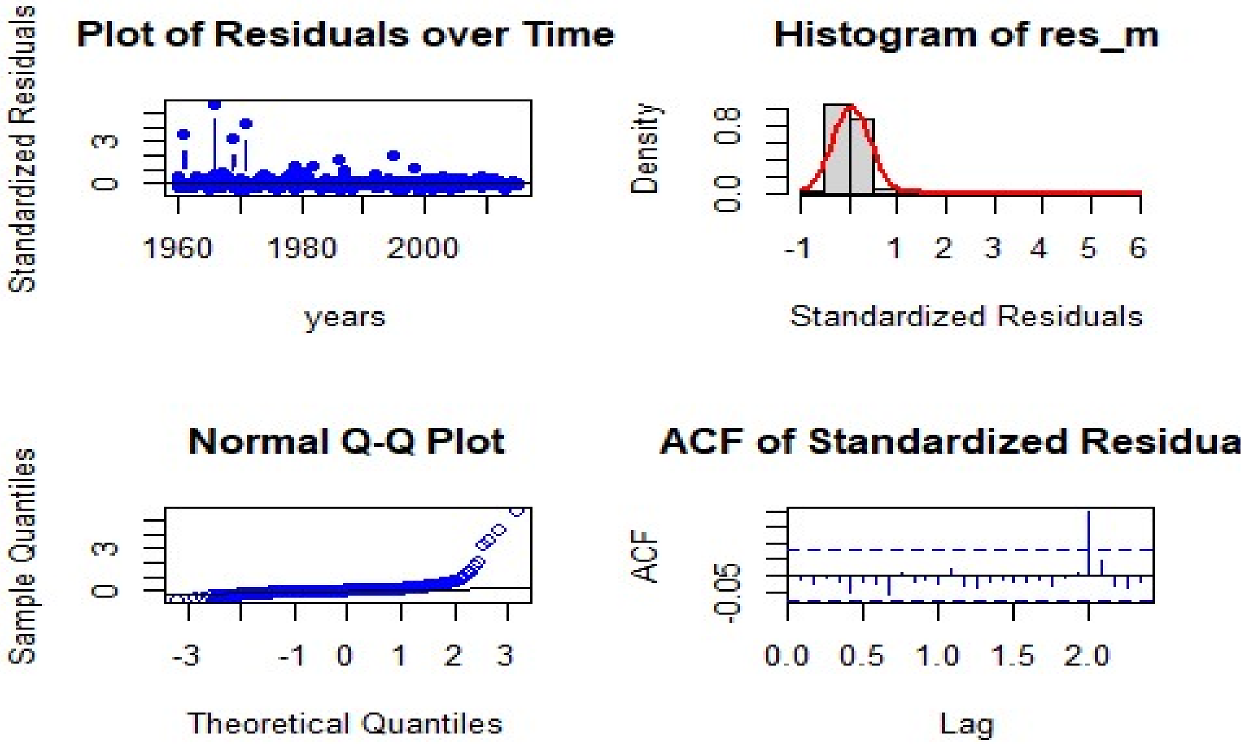





Plot of Residuals over Time
Standardized Residuals
years
Histogram of res_m
Density
Standardized Residuals
Normal Q-Q Plot
Sample Quantiles
Theoretical Quantiles
ACF of Standardized Residua
ACF
Lag


Plot of Residuals over Time
Standardized Residuals
years
Histogram of res_m
Density
Standardized Residuals
Normal Q-Q Plot
Sample Quantiles
Theoretical Quantiles
ACF of Standardized Residua
ACF
Lag


Plot of Residuals over Time
Standardized Residuals
0
4
1960
1980
2000
years
Histogram of res_m
Density
0.0
0.8
0
2
4
6
Standardized Residuals
Normal Q-Q Plot
Sample Quantiles
0
4
-3
-1
0
1
2
3
Theoretical Quantiles
ACF of Standardized Residua
ACF
-0.05
0.0
0.5
1.0
1.5
2.0
Lag


Plot of Residuals over Time
Standardized Residuals
0
3
1960
1980
2000
years
Histogram of res_m
Density
0.0
0.8
-1
1
2
3
4
5
6
Standardized Residuals
Normal Q-Q Plot
Sample Quantiles
0
3
-3
-1
0
1
2
3
Theoretical Quantiles
ACF of Standardized Residua
ACF
-0.05
0.0
0.5
1.0
1.5
2.0
Lag

Residual Analysis for each seasonality component method.

Residual Analysis Holt-Winters' Additive Method: (1)

1. The data points are below the line at the start and below the line at the end of the trend. Randomness is seen.
2. From normal distribution curve, the distribution is almost symmetric.
3. The data at the tails is deviated more leaving some part on the line suggesting there is some normality in the trend.
4. There are significant lags in Autocorrelation plot suggesting that the stochastic component is not white noise.
5. p - value (~ 0.01) from Shapiro-Wilk normality test is < 0.05 and therefore, it is statistically significant. Therefore, Null hypothesis is rejected. Suggesting some normality.

Residual Analysis Holt-Winters' multiplicative method with exponential trend: (2)

1. The data points are below the line at the start and below the line at the end of the trend. Randomness is not seen.
2. From normal distribution curve, the distribution is almost symmetric.
3. The data at the tails is deviated more leaving some part on the line suggesting there is some normality in the trend.
4. There is only one significant lag in Autocorrelation.
5. p - value (~ 0.01) from Shapiro-Wilk normality test is < 0.05 and therefore, it is statistically significant. Therefore, Null hypothesis is rejected. Suggesting some normality.

Residual Analysis Holt-Winters' multiplicative method: (3)

1. The data points are below the line at the start and below the line at the end of the trend. Randomness is not seen.
2. From normal distribution curve, the distribution is almost symmetric.
3. The data at the tails is deviated more leaving some part on the line suggesting there is some normality in the trend.
4. There is only one significant lag in Autocorrelation.
5. p - value (~ 0.01) from Shapiro-Wilk normality test is < 0.05 and therefore, it is statistically significant. Therefore, Null hypothesis is rejected. Suggesting some normality.

Residual Analysis Holt-Winters' additive method: (4)

1. The data points are below the line at the start and below the line at the end of the trend. Randomness is seen.
2. From normal distribution curve, the distribution is almost symmetric.
3. The data at the tails is deviated more leaving some part on the line suggesting there is some normality in the trend.

4. There are significant lags in Autocorrelation plot suggesting that the stochastic component is not white noise.
5. p - value (~ 0.01) from Shapiro-Wilk normality test is < 0.05 and therefore, it is statistically significant. Therefore, Null hypothesis is rejected. Suggesting some normality.

Residual Analysis Holt-Winters' multiplicative method with exponential trend: (5)

1. The data points are below the line at the start and below the line at the end of the trend. Randomness is not seen.
2. From normal distribution curve, the distribution is almost symmetric.
3. The data at the tails is deviated more leaving some part on the line suggesting there is some normality in the trend.
4. There is only one significant lag in Autocorrelation.
5. p - value (~ 0.01) from Shapiro-Wilk normality test is < 0.05 and therefore, it is statistically significant. Therefore, Null hypothesis is rejected. Suggesting some normality.

Residual Analysis Holt-Winters' multiplicative method: (6)

1. The data points are below the line at the start and below the line at the end of the trend. Randomness is not seen.
2. From normal distribution curve, the distribution is almost symmetric.
3. The data at the tails is deviated more leaving some part on the line suggesting there is some normality in the trend.
4. There is only one significant lag in Autocorrelation.
5. p - value (~ 0.01) from Shapiro-Wilk normality test is < 0.05 and therefore, it is statistically significant. Therefore, Null hypothesis is rejected. Suggesting some normality.

Therefore, this damped results show some changes from the previous analysis but not much. Overall, Holt-Winters' multiplicative methods shows better auto correlation and seasonality.

Let us now append the scores of these smoothing models with our previous scores data frame.

```
values <- data.frame(levels, fit_mase, fit_aic, fit_bic)
colnames(values) <- c("Trend", "Seasonality", "Damped", "MASE", "AIC", "BIC")
values$Trend <- factor(values$Trend, levels = c(T,F), labels =
c("multiplicative","additive"))
values$Damped <- factor(values$Damped, levels = c(T,F), labels =
c("damped","N"))
values <- unite(values, col = "MODEL_NAME",
c("Trend","Seasonality","Damped"))
```

```
v_score1 <- rbind(v_score, values)
v_score1

##                               MODEL_NAME      MASE      AIC      BIC
## 1                    finite_dlm_solar_rad 1.5779955 4602.658 4660.858
## 2                     PolyDLM_model_solar 1.6627033 4688.551 4715.478
## 3                         Koyck_DLM_solar 1.0324829 3946.476 3964.439
## 4                         AR_DLM_solar_35 0.4502885 3098.808 3148.139
## 5                         AR_DLM_solar_45 0.4479481 3096.024 3149.839
## 6                         AR_DLM_solar_55 0.4479311 3097.877 3156.177
## 7                  additive_additive_damped 0.2471600 5434.708 5511.076
## 8  multiplicative_multiplicative_damped 0.2320404 6584.208 6660.576
## 9        additive_multiplicative_damped 0.2233077 6648.746 6725.114
## 10                    additive_additive_N 0.2471600 5434.708 5511.076
## 11     multiplicative_multiplicative_N 0.2320404 6584.208 6660.576
## 12           additive_multiplicative_N 0.2233077 6648.746 6725.114
```

State Space Model Variations

Here we have 8 State Space Model Variations.

```
var <- c("AAA", "MAA", "MAM", "MMM")
damps <- c(T,F)
ets_models <- expand.grid(var, damps)
ets_aic <- array(NA, 8)
ets_mase <- array(NA,8)
ets_bic <- array(NA,8)
mod <- array(NA, dim=c(8,2))
for (i in 1:8){
  ets <- ets(v_solar_radiation_TS , model = toString(ets_models[i, 1]),
damped = ets_models[i,2])
  ets_aic[i] <- ets$aic
  ets_bic[i] <- ets$bic
  ets_mase[i] <- accuracy(ets)[6]
  mod[i,1] <- toString(ets_models[i,1])
  mod[i,2] <- ets_models[i,2]
}
```

Let us find the best ets model.

```
v_ets_fit <- ets(v_solar_radiation_TS)
summary(v_ets_fit)

## ETS(A,Ad,A)
##
## Call:
##  ets(y = v_solar_radiation_TS)
##
##   Smoothing parameters:
##     alpha = 0.9999
##     beta  = 1e-04
```

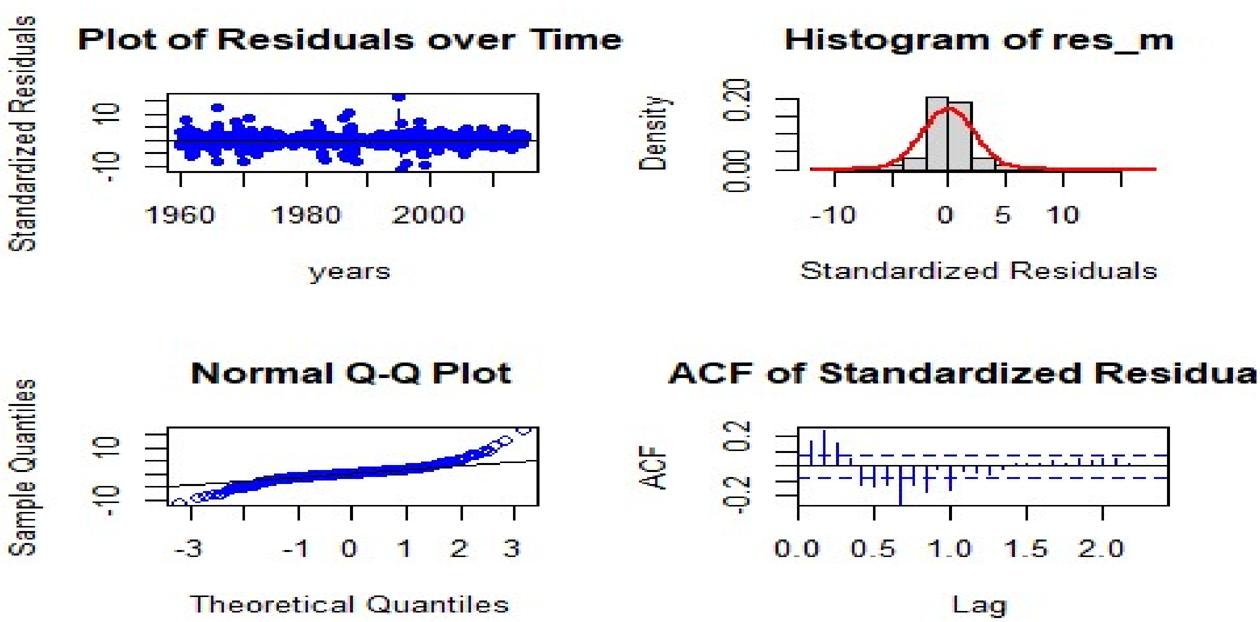
```
##     gamma = 1e-04
##     phi   = 0.9388
##
##   Initial states:
##     l = 11.154
##     b = 0.7632
##     s = -10.4919 -8.137 -3.348 2.5794 8.08 11.1219
##            9.9586 6.9916 1.9573 -1.8565 -7.1607 -9.6946
##
##   sigma:  2.3446
##
##      AIC     AICc      BIC
## 5428.422 5429.489 5509.282
##
## 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
```

ETS(A, AD, A)

- A - Additive errors
- Ad - Additive damped trend
- A - Additive seasonality.

Let us perform residual analysis on this ETS model.

```
res_analysis(residuals(v_ets_fit))
```



```
##
##  Shapiro-Wilk normality test
##
## data:  res_m
## W = 0.88756, p-value < 2.2e-16
```

Residual Analysis ETS(A, AD, A):

1. The data points are below the line at the start and below the line at the end of the trend. Randomness is seen to some extent. So, we cannot decide anything at this stage. Further analysis is required.
2. From normal distribution curve, the distribution is almost symmetric.
3. The data at the tails is deviated more leaving some part on the line suggesting there is some normality in the trend.
4. There are significant lags in Autocorrelation plot suggesting that the stochastic component is not white noise.
5. p - value (~ 0.01) from Shapiro-Wilk normality test is < 0.05 and therefore, it is statistically significant. Therefore, Null hypothesis is rejected. Suggesting some normality.

Let us now append the scores of these State Space Model with our previous scores data frame.

```
measures <- data.frame(mod, ets_mase, ets_aic, ets_bic)
measures$X2 <- factor(measures$X2, levels = c(T,F), labels = c("Damped","N"))
measures <- unite(measures, "MODEL_NAME", c("X1","X2"))
colnames(measures) <- c("MODEL_NAME", "MASE", "AIC", "BIC")
v_score2 <- rbind(v_score1, measures)

v_score2 <- arrange(v_score2, MASE)

v_score2

##                              MODEL_NAME      MASE      AIC      BIC
## 1        additive_multiplicative_damped 0.2233077 6648.746 6725.114
## 2             additive_multiplicative_N 0.2233077 6648.746 6725.114
## 3  multiplicative_multiplicative_damped 0.2320404 6584.208 6660.576
## 4       multiplicative_multiplicative_N 0.2320404 6584.208 6660.576
## 5                            AAA_Damped 0.2461797 5428.422 5509.282
## 6              additive_additive_damped 0.2471600 5434.708 5511.076
## 7                   additive_additive_N 0.2471600 5434.708 5511.076
## 8                                 AAA_N 0.2471600 5434.708 5511.076
## 9                            MMM_Damped 0.3201193 5995.550 6076.410
## 10                           MAM_Damped 0.3222574 5953.502 6034.363
## 11                                MAM_N 0.3721664 6105.959 6182.327
## 12                           MAA_Damped 0.3798095 6469.079 6549.940
## 13                      AR_DLM_solar_55 0.4479311 3097.877 3156.177
## 14                      AR_DLM_solar_45 0.4479481 3096.024 3149.839
## 15                      AR_DLM_solar_35 0.4502885 3098.808 3148.139
```

```
## 16                              MAA_N 0.4748561 7602.755 7679.123
## 17                              MMM_N 0.5292151 6670.168 6746.536
## 18                    Koyck_DLM_solar 1.0324829 3946.476 3964.439
## 19                finite_dlm_solar_rad 1.5779955 4602.658 4660.858
## 20                  PolyDLM_model_solar 1.6627033 4688.551 4715.478
```

The additive_multiplicative_damped model is better in terms of MASE score. Therefore let us

Forecasting

Let us forecast for the next 2 years on Solar radiation series. From 2016 to 2017.

```
fit <- hw(v_solar_radiation_TS, seasonal = "multiplicative", h =
2*frequency(v_solar_radiation_TS))

v_solar_forecasts <- ts.intersect(ts(fit$lower[, 2], start = c(2015, 1),
frequency = 12), ts(fit$mean, start = c(2015, 1), frequency = 12),
ts(fit$upper[, 2], start = c(2015, 1), frequency = 12))
colnames(v_solar_forecasts) <- c("Lower bound", "Point forecast", "Upper
bound")

v_solar_forecasts

##          Lower bound Point forecast Upper bound
## Jan 2015   1.2440033       5.610335    9.976666
## Feb 2015  -0.3306776       6.512806   13.356290
## Mar 2015  -2.6068190       8.786120   20.179058
## Apr 2015  -5.4366123      10.192785   25.822181
## May 2015  -9.6237347      12.512597   34.648928
## Jun 2015 -14.1890164      14.061609   42.312235
## Jul 2015 -18.2282011      14.502088   47.232377
## Aug 2015 -19.6235881      12.945620   45.514829
## Sep 2015 -18.5159294      10.352210   39.220350
## Oct 2015 -15.4143760       7.418279   30.250935
## Nov 2015 -11.9078451       4.989730   21.887305
## Dec 2015 -10.5204066       3.871776   18.263960
## Jan 2016 -12.9554236       4.199400   21.354224
## Feb 2016 -16.7893479       4.838892   26.467131
## Mar 2016 -25.1740619       6.477174   38.128410
## Apr 2016 -32.3437711       7.452633   47.249037
## May 2016 -43.8456686       9.069749   61.985166
## Jun 2016 -54.2836502      10.099485   74.482621
## Jul 2016 -61.5580226      10.315199   82.188420
## Aug 2016 -60.3285438       9.113765   78.556075
## Sep 2016 -52.8981594       7.208700   67.315559
## Oct 2016 -41.5237568       5.105869   51.735495
## Nov 2016 -30.5729044       3.391945   37.356795
## Dec 2016 -25.9541251       2.597256   31.148637
```

Now let us plot the forecast.

```
plot(fit, fcol = "white", main = "Forecast of Solar radiation series for the
next 2 years (2016, 2017)", ylab = "Solar Radiation")
lines(fitted(fit), col = "red")
lines(fit$mean, col = "blue", lwd = 2)
legend("bottom", inset = .03, cex = 0.9, box.lty = 2, box.lwd = 2, pch = 1,
lty = 1, col = c("red", "blue"), c("Data", "Forecasts"))
```

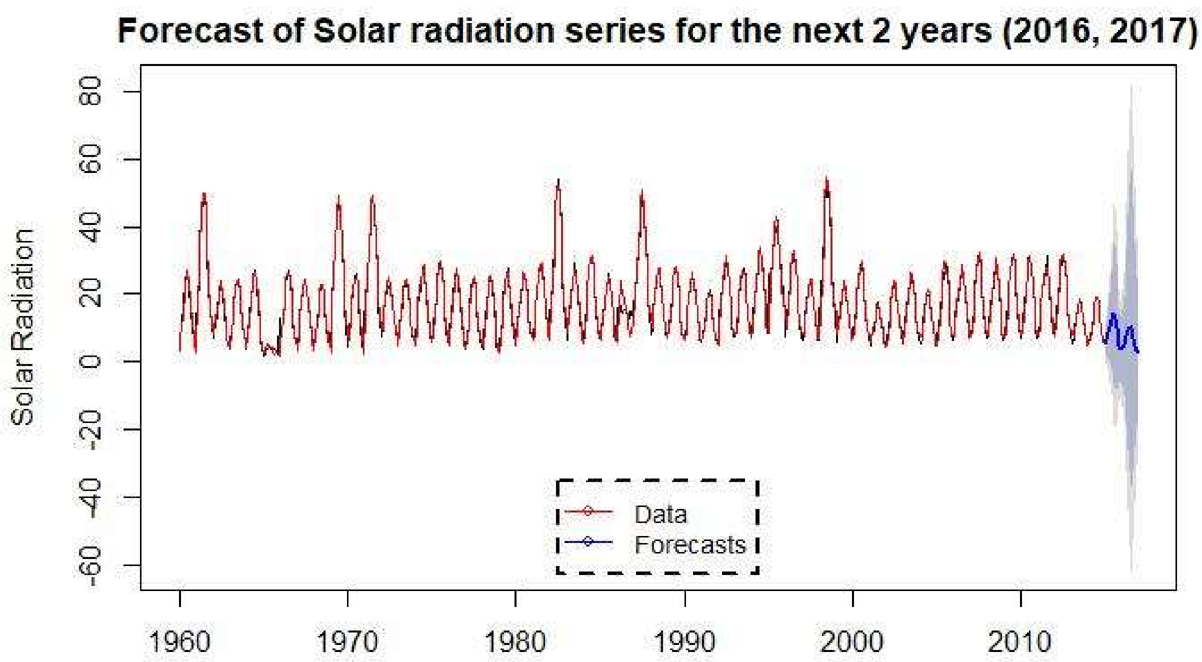


Fig 1.9: Next 2 years forecast on the Solar Radiation Series.

From the two year forecast results we can predict that there will be decrease in the Solar Radiation in the future.

Part 2

Data

The data here used is quarterly Residential Property Price Index (PPI) in Melbourne and quarterly population change over the previous quarter in Victoria between September 2003 and December 2016

```
v_Task2_data <- read.csv("data2.csv")
head(v_Task2_data)

##    Quarter price change
## 1 Sep-2003  60.7  14017
```

```
## 2 Dec-2003  62.1  12350
## 3 Mar-2004  60.8  17894
## 4 Jun-2004  60.9   9079
## 5 Sep-2004  60.9  16210
## 6 Dec-2004  62.4  13788
```

```
# Using str() to check the type of each column.
str(v_Task2_data)
```

```
## 'data.frame':    54 obs. of  3 variables:
##  $ Quarter: chr  "Sep-2003" "Dec-2003" "Mar-2004" "Jun-2004" ...
##  $ price  : num  60.7 62.1 60.8 60.9 60.9 62.4 62.5 63.2 63.1 64 ...
##  $ change : int  14017 12350 17894 9079 16210 13788 21195 10904 16995
16962 ...
```

Checking Missing values.

```
colSums(is.na(v_Task2_data))
```

```
## Quarter   price  change
##       0       0       0
```

There are no missing values in the data.

Checking the class of v_Task2_data. (It should be data frame.)

```
class(v_Task2_data)
```

```
## [1] "data.frame"
```

```
v_PPI_change_TS <- ts(v_Task2_data$price , start = c(2003, 3), frequency = 4)
v_population_change_TS <- ts(v_Task2_data$change, start = c(2003, 3),
frequency = 4)
```

Confirming the class of each time series object.

```
class(v_PPI_change_TS)
```

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

```
class(v_population_change_TS)
```

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

Now let us visualize each time series object.

Descriptive analysis

Property Price Index

```
plot(v_PPI_change_TS, type = "b", xlab = "years", ylab = "Population Price
index", main = "Residential Property Price Index from 2003-3 to 2016-4 (54
Quarters)", pch = 1)
legend("bottomright", inset = .03, title = "Population Price Index", legend =
```

```
"Population Price Index series", horiz = TRUE, cex = 0.8, lty = 1, box.lty =
2, box.lwd = 2, pch = 1)
```

Fig 2.1: Residential Property Price Index - Time series plot.

```
McLeod.Li.test(y = v_PPI_change_TS, main = "McLeod-Li Test Statistics for
Residential Population price index")
```

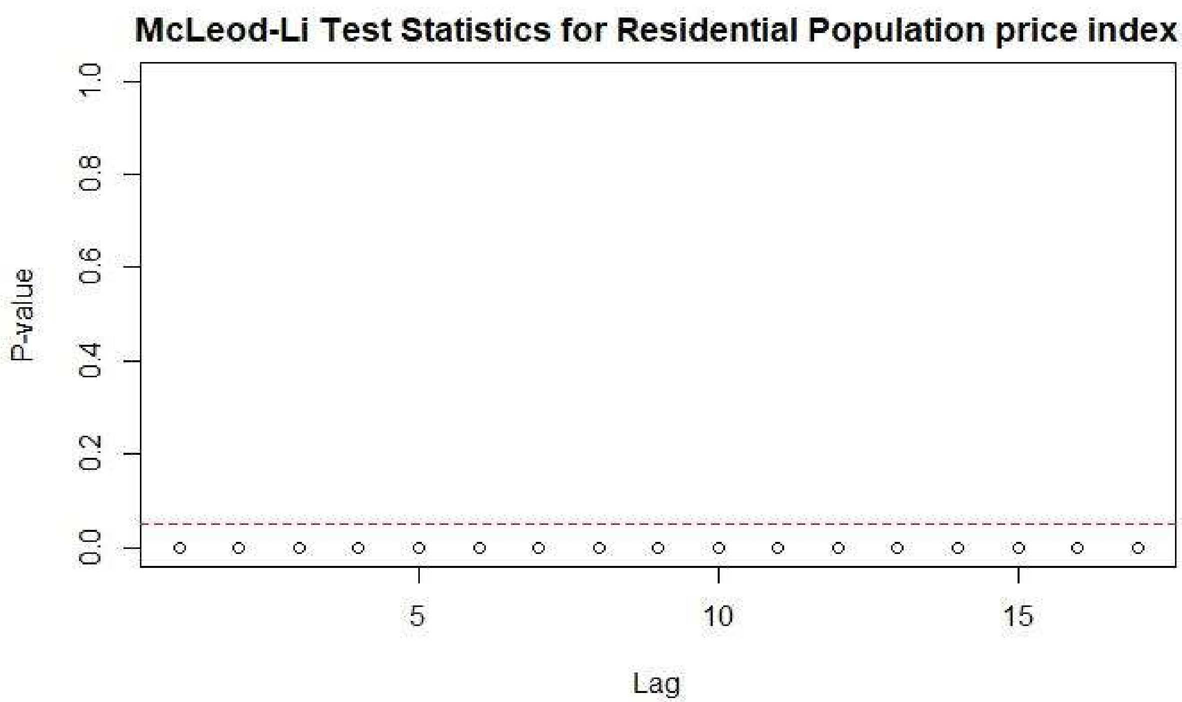


Fig 2.2: McLeod-Li Test Statistics for Residential Population price index.

Descriptive analysis

1. From fig1, we can observe an upward trend.
2. There is no seasonality in the series.
3. The series show Autoregressive and moving average behavior.
4. From the plot we cannot see a change in variance.
5. There is no obvious intervention in the data.

Population Change

```
plot(v_population_change_TS, type = "b", xlab = "years", ylab = "Population
Change", main = "Population Change from 2003-3 to 2016-4 (54 Quarters)", pch
= 1)
legend("bottomright", inset = .03, title = "Population Change", legend =
"Population Change series", horiz = TRUE, cex = 0.8, lty = 1, box.lty = 2,
box.lwd = 2, pch = 1)
```

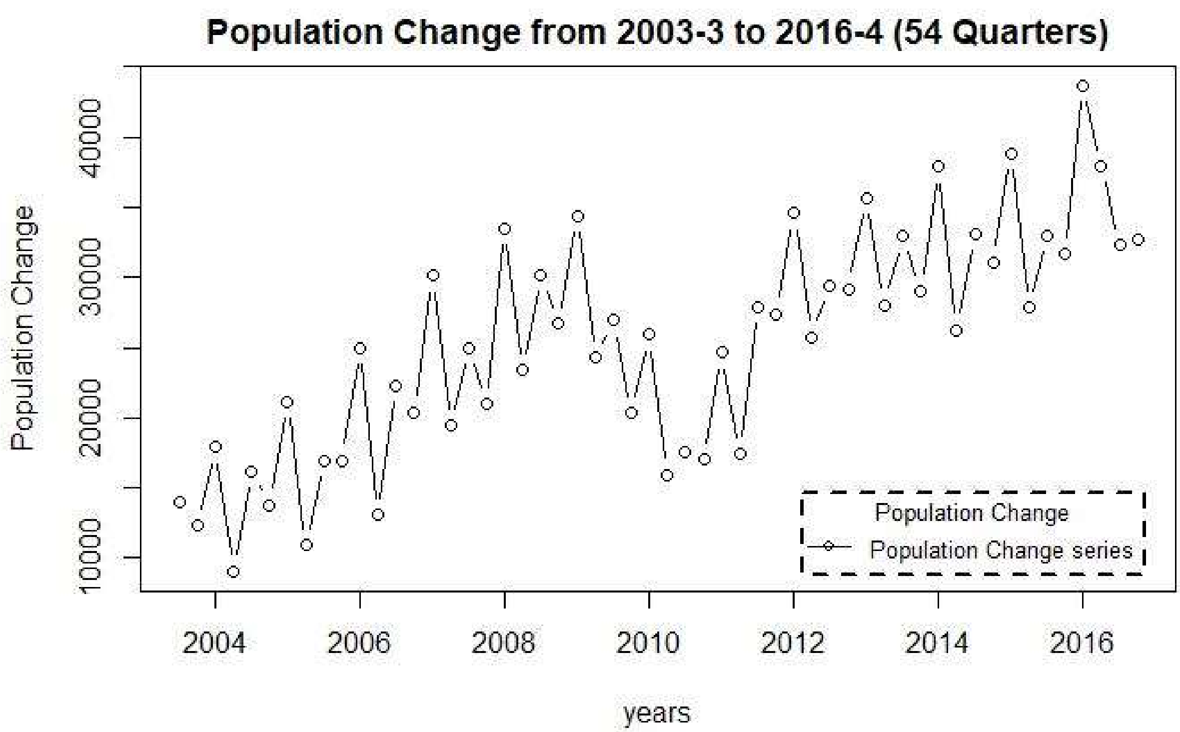


Fig 2.3: Population change - Time series plot.

```
McLeod.Li.test(y = v_population_change_TS, main = "McLeod-Li Test Statistics
for Population Change")
```

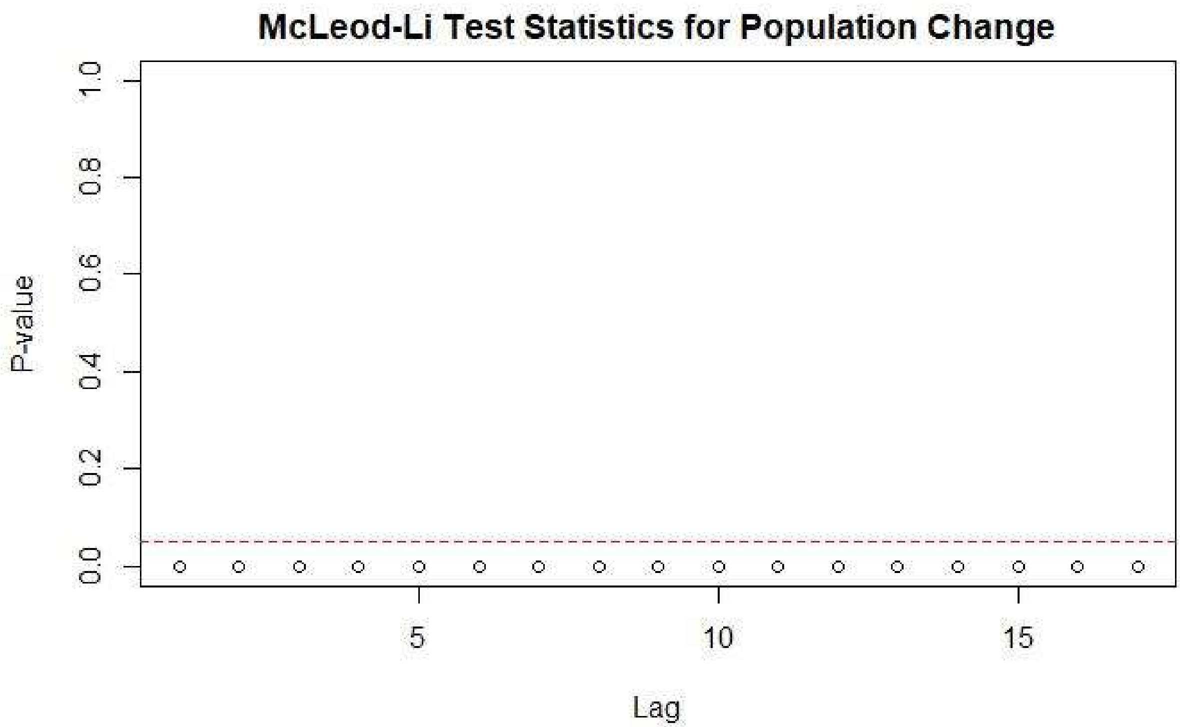


Fig 2.4: McLeod-Li Test Statistics for Population Change.

Descriptive analysis

1. From fig1, we can observe an upward trend.
2. There is seasonality in the series with higher values in the first quarter and lower values in the second quarter.
3. The series doesn't clearly show Autoregressive and moving average behavior.
4. We cannot see an change in variance.
5. There is some intervention in the data across the year.

From the two plots we can observe there is some correlation in the data. Let us analyze the correlation between both the series by plotting the series sample with cross correlation function (CCF).

Correlation Analysis

```
ccf(as.vector(v_PPI_change_TS), as.vector(v_population_change_TS), ylab =
"CCF", main = "PPI vs Population Change")
```

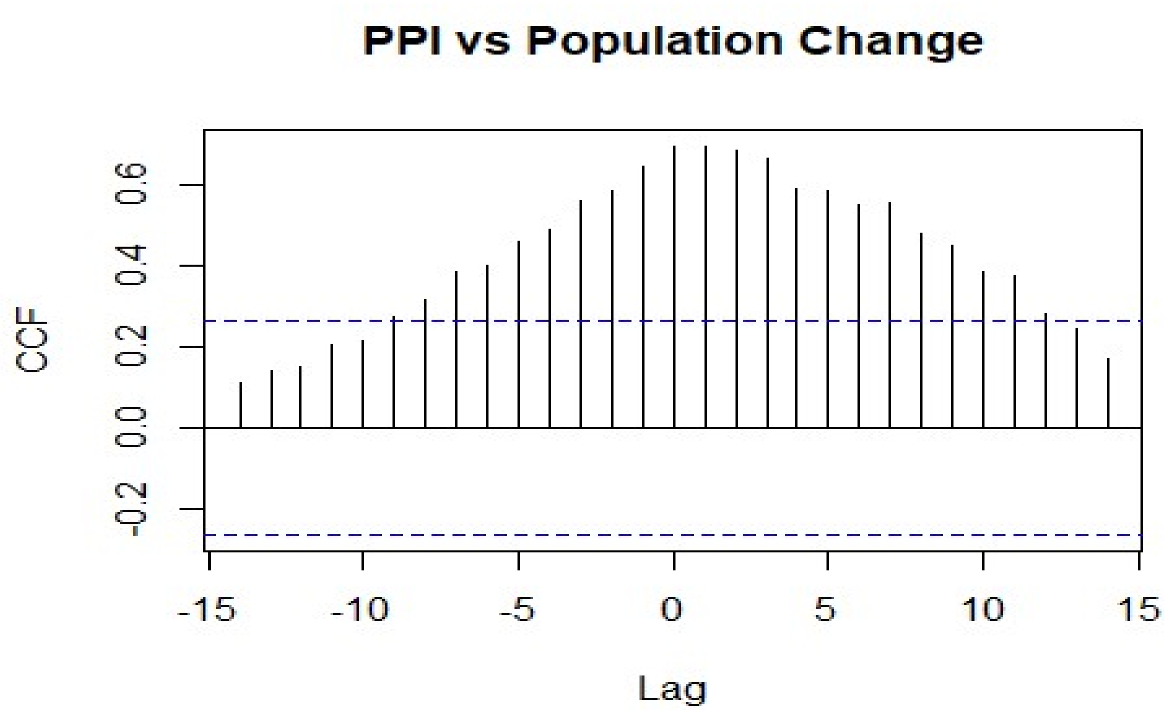


Fig 2.4: CCF plot

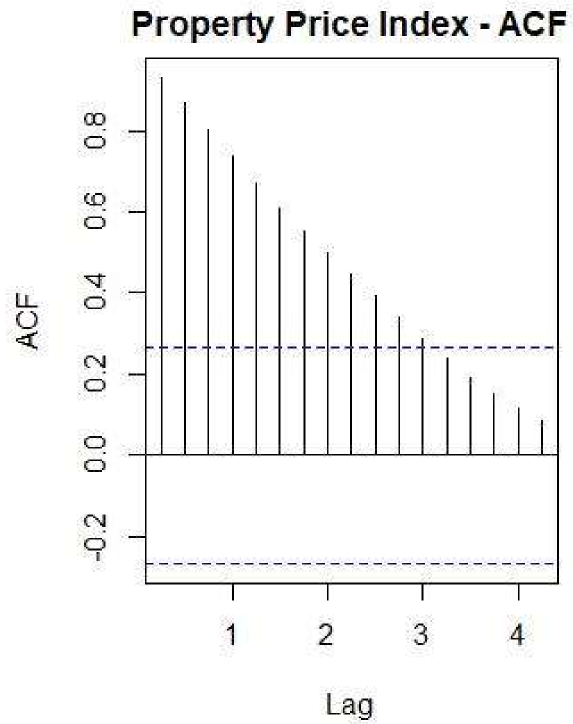
The plot suggests that there is a strong correlation between Residential Property Price Index (PPI) in Melbourne and population change Victoria. Also, We can clearly see that the lags are significantly different from zero based on 1.96/n−√ bounds.

Checking for Stationary in the series

```
# Function to check Stationary on the series.
Stationary_Check <- function(x, m1, m2) {

  # Analysing trends by plotting ACF and PACF.
  par(mfrow = c(1,2))
  acf(x, main = m1)
  pacf(x, main = m2)

  # Conducting Augmented Dickey-Fuller test.
  adf.test(x)
}
```

Checking for Stationary on Property Price Index

```
Stationary_Check(v_PPI_change_TS, "Property Price Index - ACF", "Property
Price Index - PACF")
```



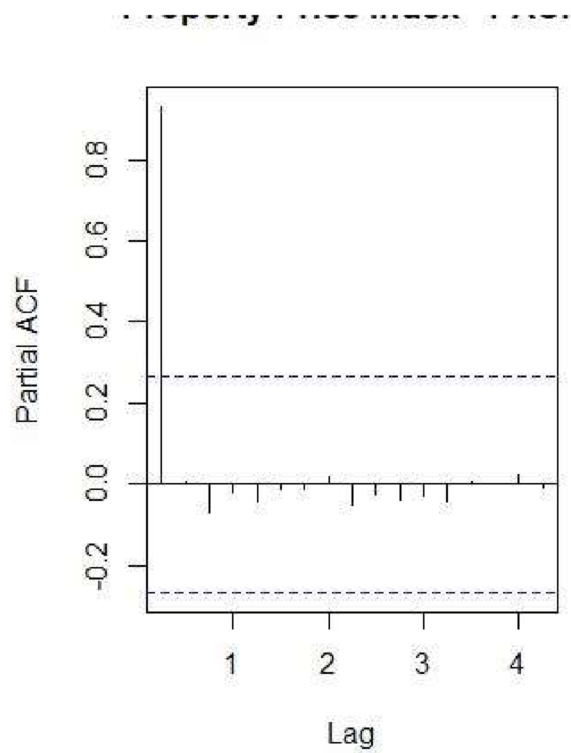


Fig 2.6: Population Price Index - ACF

Fig 2.7: Population Price Index - PACF

```
##
##  Augmented Dickey-Fuller Test
##
## data:  x
## Dickey-Fuller = -1.3264, Lag order = 3, p-value = 0.8458
## alternative hypothesis: stationary
```

The decrease in the ACF plot and a high peak in the PACF plot in the beginning, suggests that there is some pattern in the Property Price Index trend.

Hypotheses:

H_0: The data is not stationary.

H_A: The data is stationary.

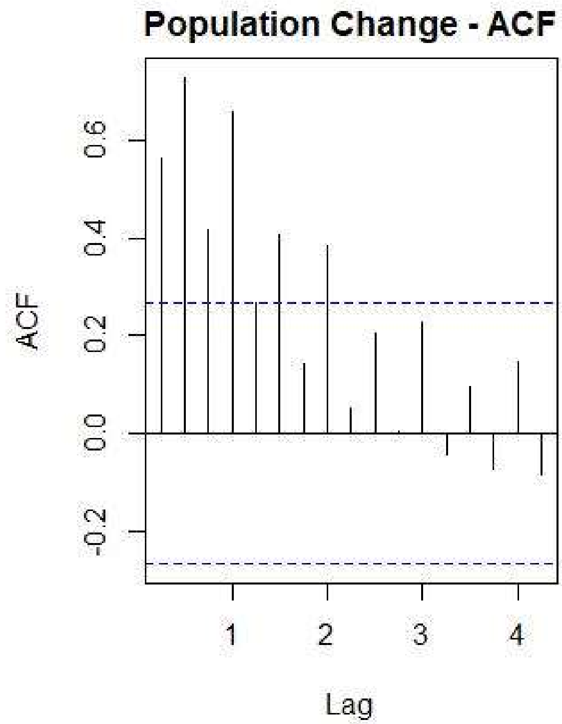
Interpretations:

- p - value: 0.8458 > 0.05
- p - value is greater than 0.05 and hence the test is not statistically significant. Therefore, we fail to reject Null hypothesis i.e., The data is not stationary.

Therefore, the property price index series is non - stationary.

Checking for Stationary on Population change

```
Stationary_Check(v_population_change_TS, "Population Change - ACF",
"Population Change - PACF")
```



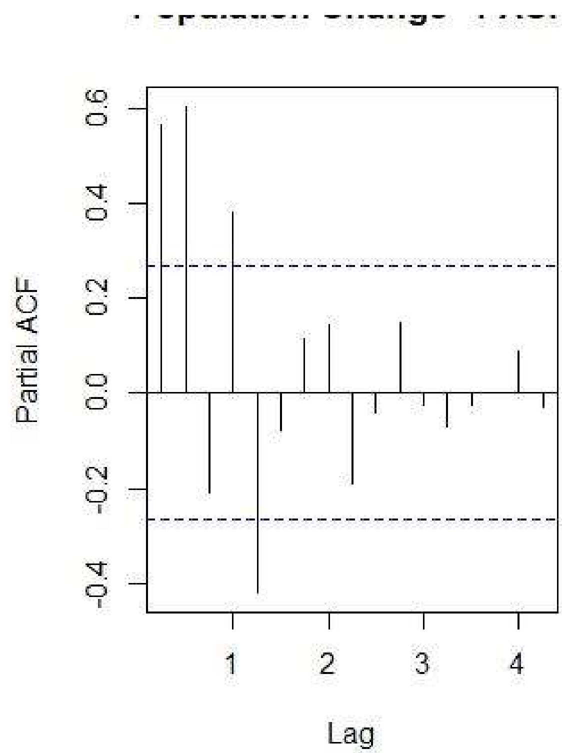


Fig 2.8: Population Change - ACF

Fig 2.9: Popolation Change - PACF

```
##
##  Augmented Dickey-Fuller Test
##
## data:  x
## Dickey-Fuller = -1.603, Lag order = 3, p-value = 0.7344
## alternative hypothesis: stationary
```

The decrease in the ACF plot and a high peak in the PACF plot in the beginning, suggests that there is some pattern in the GOLD price trend.

Hypotheses:

H_0: The data is not stationary.

H_A: The data is stationary.

Interpretations:

- p - value : 0.7344 > 0.05
- p - value is greater than 0.05 and hence the test is not statistically significant. Therefore, we fail to reject Null hypothesis i.e., The data is not stationary.

Therefore, the Population change series is non - stationary.

The two series are not stationary with high auto - correlation between them. This strong auto correlation makes it difficult in assessing the dependency between the series. To separate this strong correlationship between the series we will apply pre-whitening.

Prewhitening

Here, we will first make the data stationary by differencing the series. Our previous analysis suggests that we should do both the regular and seasonal differentiation to make sure that both the data has the same length.

```
v_diff <- ts.intersect(diff(diff(v_PPI_change_TS, 4)),
diff(diff(v_population_change_TS, 4)))
prewhiten(as.vector(v_diff[, 1]), as.vector(v_diff[, 2]), ylab = 'CCF', main
= "Prewhitened CFF")
```

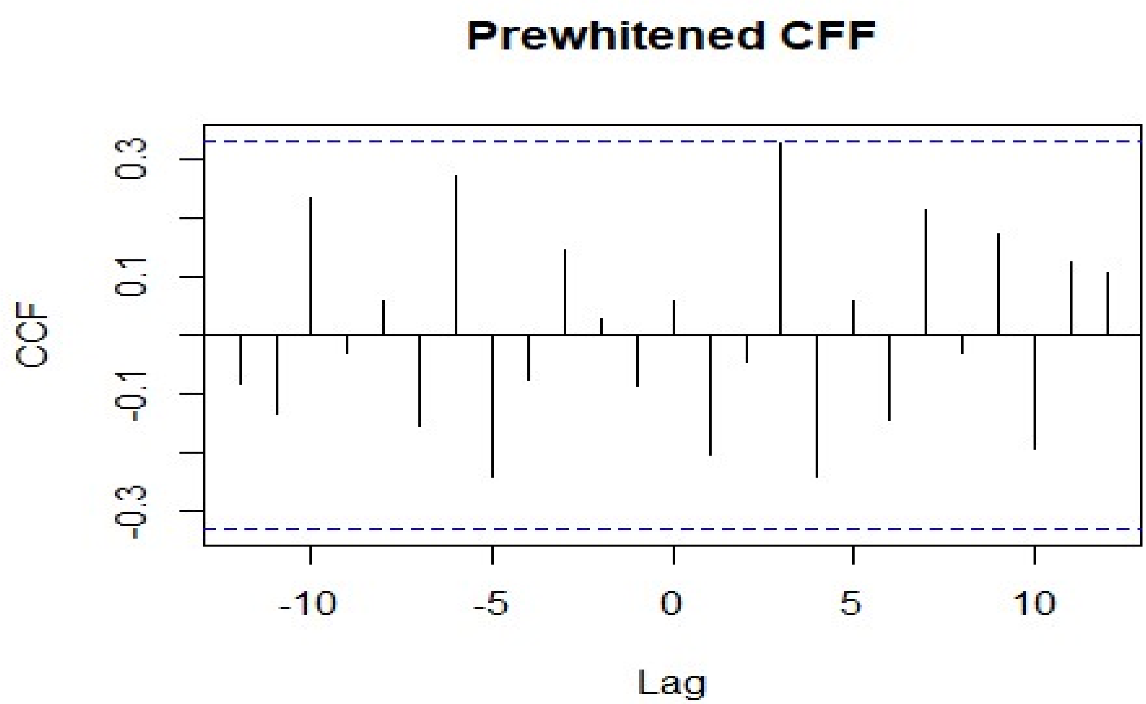


Fig 2.10: Pre Whitening

Now the data is stationary and there is no significant correlation between the two series. Therefore, we can conclude that the strong cross correlation is spurious.

Conclusion

Part 1

The data columns are converted into the time series objects of Solar Radiation and Precipitation respectively. Both have seasonality in their series with no obvious trend, behavior or change in variance.

Solar Radiation is stationary and hence it can me directly send to the models as it is a dependent variable. Similarly, Precipitation being the independent variable there is no need to make it stationary.

Also, we got negative correlation among the series.

Now to find the best model we have three approaches.

1. Suitable Dlag models: Here we got AR_DLM_solar_45 as the better model in terms of residual analysis as well as MASE scores. But it suffered with randomness, normality and correlation to some extent.

2. Smoothing methods: Here the better model method is Holt-Winters' multiplicative method in terms of residual analysis as well as MASE scores.

3. State space model variations: Here we got ETS(A, Ad, A) as the best model automatically. But it suffered with randomness, normality and correlation to some extent. But the MASE scores are shown least for Holt-Winters' multiplicative method.

So we considered this as the best model for forecasting the solar radiation data for the nest 2 years. From the two year forecast results we can predict that there will be decrease in the Solar Radiation in the future. But as we forecast with 95% confidence intervals we cannot consider this as accurate.

Part 2:

The data columns are converted into the time series objects of Property Price Index and Population Change respectively.

Population Change has seasonality in their series with no obvious trend, behavior or change in variance.

Both the series are Non - Stationary.

There is a strong correlation between the two series.

Two check whether this relationship is spurious or not we performed pre whitening. This resulted it there are no strong correlation bonds between the two series. Therefore, we acn conclude that the relationship between Residential Property Price Index (PPI) in

Melbourne and quarterly population change over the previous quarter in Victoria between September 2003 and December 2016 is spurious.

References

[1] [Online]. Available: https://cran.r-project.org/web/packages/forecast/index.html.

[2] lmtest, [Online]. Available: https://cran.r-project.org/web/packages/lmtest/index.html.

[3] tseries, [Online]. Available: https://cran.r-project.org/web/packages/tseries/index.html.

[4] fUnitRoots, [Online]. Available: https://cran.r-project.org/web/packages/fUnitRoots/index.html.

[5] expsmooth, [Online]. Available: https://cran.r-project.org/web/packages/expsmooth/index.html.

[6] urca, [Online]. Available: https://cran.r-project.org/web/packages/urca/index.html.

[7] Week 3, 4, 5 and 6 module tasks.

[8] My previous works for Time series Analysis in my Second year First Sem.