FORECASTING FINAL PROJECT 2023

COURSED CODE: MATH1307

STUDENT ID: S3914638

Table of Contents

TASK 1

Introduction

The effects of pollution and climate factors on mortality are intricate and complicated problems. Numerous studies have investigated the connection between these elements and mortality in various circumstances. Infectious illness transmission can be influenced by climatic conditions including temperature and humidity. It takes a lot of investigation and data analysis to comprehend how mortality, climatic conditions, and pollution interact. In order to lessen the detrimental effects on health caused by these variables, public health policies and interventions are informed by this study.

The purpose of the research is to analyze disease-specific mortality between the years 2010 and 2020 that has been impacted by both pollution and climatic conditions, and then to review the findings to draw conclusions. The dataset incorporates the weekly mortality average in Paris, France, along with information about the local environment within the city (Fahrenheit temperature), the concentration of contaminants, and the number of detrimental substances released from vehicles and industries, all recorded at the same locations between 2010 and 2020.

T1_mort.csv contains data for each of the five periods, including mortality, temperature, pollutants particle size, and two chemical emissions (chem1, chem2). The 508 time points between 2010 and 2020 are covered by this statistics. Estimation concerning mortality for the following four weeks will be made using this data. This assignment provides most accurate forecasts for the mortality series four weeks prior to the event in terms of R squared, AIC, BIC, MASE, etc. (as appropriate), as well as point forecasts, confidence intervals, and an appropriate plot for the best model for each technique (DLM, ARDL, polyck, koyck, dynamic, exponential smoothing, and state-space model).

The brief summary of the data set is given below:

```
> summary(mort_data)
       X           mortality          temp           chem1
 Min.   :  1.0   Min.   :142.1   Min.   :50.91   Min.   : 2.520
 1st Qu.:127.8   1st Qu.:159.6   1st Qu.:67.23   1st Qu.: 4.970
 Median :254.5   Median :166.7   Median :74.06   Median : 6.865
 Mean   :254.5   Mean   :169.0   Mean   :74.26   Mean   : 7.909
 3rd Qu.:381.2   3rd Qu.:176.4   3rd Qu.:81.49   3rd Qu.:10.080
 Max.   :508.0   Max.   :231.7   Max.   :99.88   Max.   :22.390
     chem2        particle.size
 Min.   : 21.57   Min.   :20.25
 1st Qu.: 40.21   1st Qu.:35.85
 Median : 48.23   Median :44.25
 Mean   : 50.48   Mean   :47.41
 3rd Qu.: 59.69   3rd Qu.:57.54
 Max.   :100.12   Max.   :97.94
```

Our goal during the modelling phase is to identify the model that will work well across all ordinaries benchmark rates. On the basis of the connections, they have alongside the variable that is dependent and among themselves, the aforementioned determinants were chosen.

Implement Finite Distributed Lag Model (DLM)

We will investigate the use of randomly distributed lag models, that require including a different explanatory series of data along with its lagged values. With the eventual goal of finding an effective framework for forecasting, our method seeks to more fully capture the overall variability and correlation structure in our dependent time series.

To do this, a loop that generates several reliability measures, including AIC/BIC and MASE across models with varied lag durations, has been created. In order to establish the ideal lag duration for our model, we will subsequently select the model with the lowest values for these criteria. The fitting of a finite Distributed Lag Model (DLM) with a total of 10 lagged data was chosen after careful consideration. The finding that information criteria values and MASE tend to decline as lag duration (q) grows supports this decision.

```
q =  1 AIC =  3812.934 BIC =  3855.219 Mase = 0.9196282
q =  2 AIC =  3769.907 BIC =  3829.078 Mase = 0.8853702
q =  3 AIC =  3755.807 BIC =  3831.85 Mase = 0.8688271
q =  4 AIC =  3741.636 BIC =  3834.533 Mase = 0.8544125
q =  5 AIC =  3729.364 BIC =  3839.099 Mase = 0.8387617
q =  6 AIC =  3726.178 BIC =  3852.736 Mase = 0.8382967
q =  7 AIC =  3713.873 BIC =  3857.237 Mase = 0.8244978
q =  8 AIC =  3710.06 BIC =  3870.215 Mase = 0.8215737
q =  9 AIC =  3704.335 BIC =  3881.265 Mase = 0.8174129
q =  10 AIC =  3696.208 BIC =  3889.895 Mase = 0.8145466
```

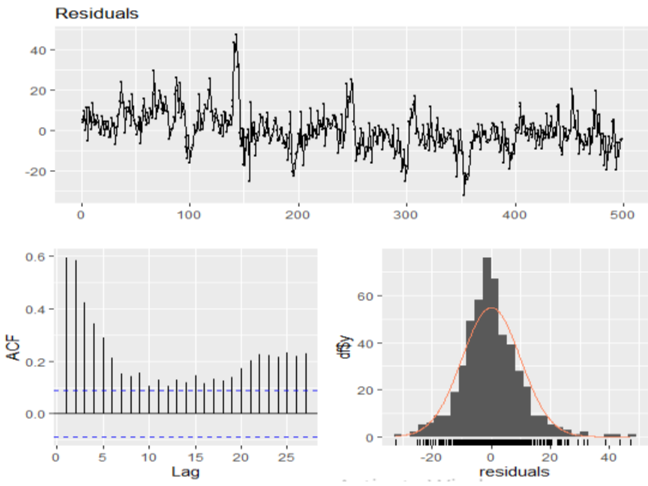
Below is the summary DLM of multiple predictors for all indexes representing point forecast and confidence interval with AIC=3744.47, BIC =3845.524and multiple R- square=0.5142. The corresponding plot for finite DLM is also presented below. Here we use q=10 for minimum values of AIC, BIC and MASE.

```
Call:
lm(formula = as.formula(model.formula), data = design)

Residuals:
    Min      1Q  Median      3Q     Max 
-32.126  -5.717  -0.682   5.390  47.296 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) 170.024985   8.667381  19.617  < 2e-16 ***
temp.t        0.215883   0.098813   2.185  0.02939 *  
temp.1       -0.306224   0.102021  -3.002  0.00283 ** 
temp.2       -0.130899   0.105455  -1.241  0.21512    
temp.3       -0.037292   0.106386  -0.351  0.72610    
temp.4        0.071821   0.109455   0.656  0.51203    
temp.5       -0.171849   0.109725  -1.566  0.11797    
temp.6       -0.040155   0.108529  -0.370  0.71155    
temp.7       -0.002236   0.105279  -0.021  0.98307    
temp.8        0.030891   0.105385   0.293  0.76955    
temp.9       -0.019568   0.102362  -0.191  0.84848    
temp.10      -0.091033   0.100694  -0.904  0.36642    
X3.t          0.178672   0.059111   3.023  0.00264 ** 
X3.1          0.085758   0.059680   1.437  0.15138    
X3.2          0.090882   0.061297   1.483  0.13883    
X3.3         -0.041179   0.061915  -0.665  0.50631    
X3.4          0.043195   0.063203   0.683  0.49467    
X3.5          0.081258   0.063647   1.277  0.20233    
X3.6          0.024181   0.063533   0.381  0.70367    
```

```
X3.7          0.096313   0.062143   1.550  0.12184
X3.8          0.033600   0.061299   0.548  0.58386
X3.9          0.079559   0.059192   1.344  0.17956
X3.10         0.058284   0.057673   1.011  0.31273
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.14 on 475 degrees of freedom
Multiple R-squared:  0.5142,    Adjusted R-squared:  0.4917
F-statistic: 22.85 on 22 and 475 DF,  p-value: < 2.2e-16

AIC and BIC values for the model:
      AIC      BIC
1 3744.47 3845.524
```

```
        Shapiro-Wilk normality test

data:  x$residuals
W = 0.96789, p-value = 5.628e-09
```

```
        Breusch-Godfrey test for serial correlation of order up to 26

data:  Residuals
LM test = 232.94, df = 26, p-value < 2.2e-16
```



Variance Inflation Factor

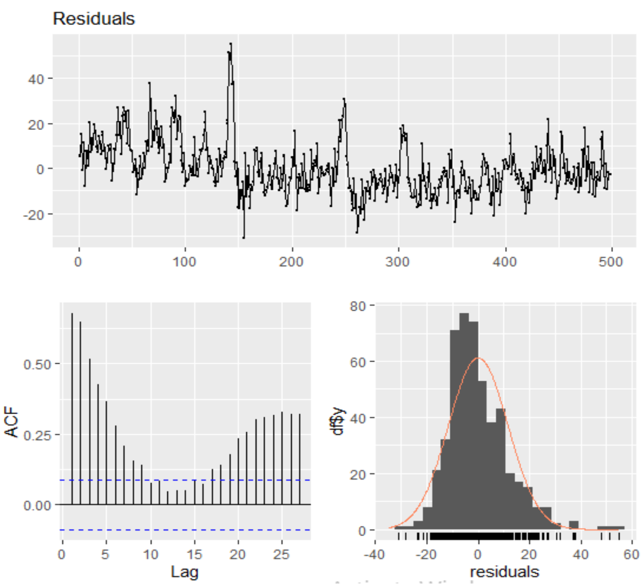
```
> VIF_m1
  temp.t   temp.1   temp.2   temp.3   temp.4   temp.5   temp.6   temp.7 
3.878845 4.138613 4.433041 4.511505 4.775051 4.803352 4.691717 4.411295 
  temp.8   temp.9  temp.10     X3.t     X3.1     X3.2     X3.3     X3.4 
4.406664 4.124785 3.963967 3.894308 3.963149 4.195181 4.260122 4.439046 
    X3.5     X3.6     X3.7     X3.8     X3.9    X3.10 
4.501028 4.479929 4.288435 4.173196 3.887738 3.703222 
> VIF_m1 > 10
 temp.t  temp.1  temp.2  temp.3  temp.4  temp.5  temp.6  temp.7  temp.8  temp.9 
  FALSE   FALSE   FALSE   FALSE   FALSE   FALSE   FALSE   FALSE   FALSE   FALSE 
temp.10    X3.t    X3.1    X3.2    X3.3    X3.4    X3.5    X3.6    X3.7    X3.8 
  FALSE   FALSE   FALSE   FALSE   FALSE   FALSE   FALSE   FALSE   FALSE   FALSE 
   X3.9   X3.10 
  FALSE   FALSE 
```

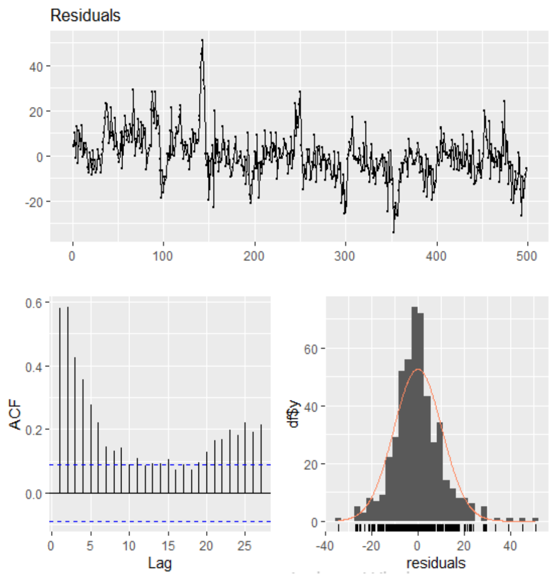
DLM for temperature and pollutant particle size index

Point forecast and confidence interval with AIC= 3880.743, BIC =3935.481 and multiple R-square=0.3324. The corresponding plot for residuals is also presented below.

```
Call:
lm(formula = model.formula, data = design)

Residuals:
    Min      1Q  Median      3Q     Max
-30.895  -7.865  -1.896   6.551  54.902

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 230.44497    5.96788  38.614  < 2e-16 ***
x.t           0.09809    0.08348   1.175 0.240546
x.1          -0.51663    0.08510  -6.071 2.57e-09 ***
x.2          -0.26483    0.08818  -3.003 0.002808 **
x.3          -0.30029    0.08849  -3.394 0.000746 ***
x.4          -0.05048    0.09017  -0.560 0.575875
x.5          -0.20227    0.09011  -2.245 0.025235 *
x.6          -0.08642    0.09009  -0.959 0.337918
x.7           0.10527    0.08841   1.191 0.234366
x.8           0.08518    0.08789   0.969 0.332983
x.9           0.16858    0.08512   1.980 0.048212 *
x.10          0.13595    0.08378   1.623 0.105286
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11.75 on 486 degrees of freedom
Multiple R-squared:  0.3324,	Adjusted R-squared:  0.3173
F-statistic:    22 on 11 and 486 DF,  p-value: < 2.2e-16

AIC and BIC values for the model:
       AIC      BIC
1 3880.743 3935.481
```

```
Call:
lm(formula = model.formula, data = design)

Residuals:
    Min      1Q  Median      3Q     Max
-34.054  -5.873  -0.709   5.064  51.024

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 128.44861    2.11266  60.799  < 2e-16 ***
x.t           0.23480    0.04349   5.400 1.05e-07 ***
x.1          -0.03930    0.04409  -0.891  0.37317
x.2           0.03466    0.04545   0.762  0.44617
x.3          -0.04439    0.04628  -0.959  0.33795
x.4           0.10265    0.04691   2.188  0.02913 *
x.5           0.02556    0.04694   0.544  0.58637
x.6           0.04844    0.04697   1.031  0.30298
x.7           0.15047    0.04647   3.238  0.00129 **
x.8           0.12310    0.04577   2.690  0.00740 **
x.9           0.13668    0.04413   3.097  0.00207 **
x.10          0.08239    0.04348   1.895  0.05868 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.49 on 486 degrees of freedom
Multiple R-squared:  0.4675,	Adjusted R-squared:  0.4555
F-statistic: 38.79 on 11 and 486 DF,  p-value: < 2.2e-16

AIC and BIC values for the model:
       AIC      BIC
1 3768.114 3822.852
```

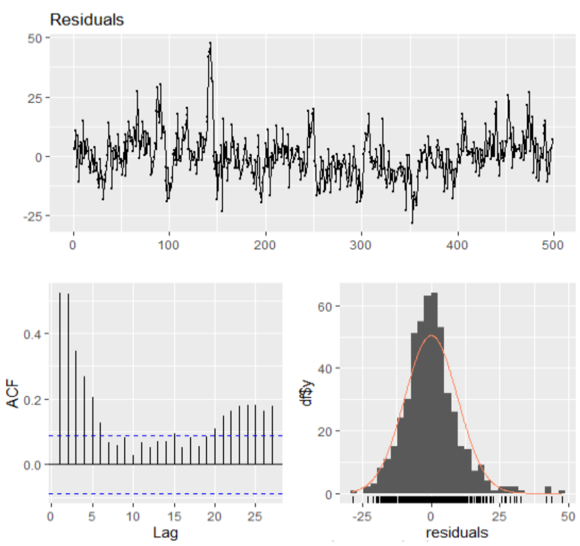


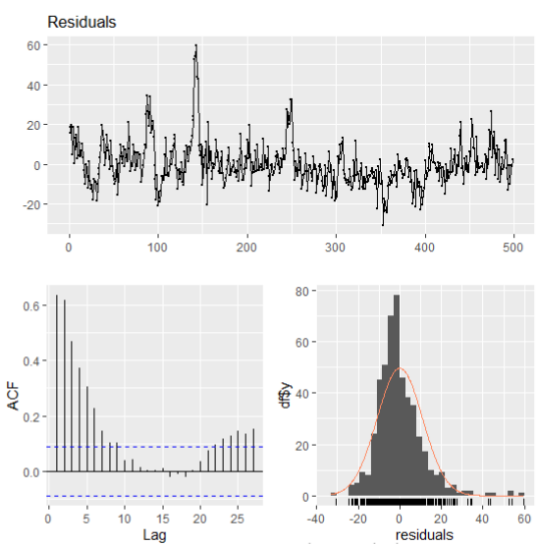


DLM function for Chemical 1 and Chemical 2 index

```
Call:
lm(formula = model.formula, data = design)

Residuals:
    Min      1Q  Median      3Q     Max
-28.258  -5.462  -0.547   4.347  47.536

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 141.94242    1.27083 111.693  < 2e-16 ***
x.t           0.86498    0.17997   4.806 2.05e-06 ***
x.1          -0.26184    0.18135  -1.444 0.149430
x.2           0.15725    0.19242   0.817 0.414221
x.3          -0.01975    0.19617  -0.101 0.919859
x.4           0.56747    0.19730   2.876 0.004203 **
x.5           0.11239    0.19725   0.570 0.569093
x.6           0.30151    0.19747   1.527 0.127448
x.7           0.70386    0.19642   3.583 0.000373 ***
x.8           0.47447    0.19262   2.463 0.014112 *
x.9           0.45753    0.18159   2.520 0.012067 *
x.10          0.04697    0.18013   0.261 0.794365
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.744 on 486 degrees of freedom
Multiple R-squared:  0.5405,	Adjusted R-squared:  0.5301
F-statistic: 51.97 on 11 and 486 DF,  p-value: < 2.2e-16

AIC and BIC values for the model:
     AIC      BIC
1 3694.7 3749.438
```

```
Call:
lm(formula = model.formula, data = design)

Residuals:
    Min      1Q  Median      3Q     Max
-30.744  -6.283  -1.566   4.703  59.445

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 120.339199   2.816910  42.720  < 2e-16 ***
x.t           0.226553   0.044858   5.050 6.25e-07 ***
x.1          -0.043932   0.044962  -0.977  0.32900
x.2           0.039548   0.046370   0.853  0.39415
x.3          -0.002763   0.046804  -0.059  0.95294
x.4           0.128075   0.047207   2.713  0.00690 **
x.5           0.029783   0.047119   0.632  0.52762
x.6           0.064553   0.047185   1.368  0.17192
x.7           0.164653   0.046838   3.515  0.00048 ***
x.8           0.133390   0.046360   2.877  0.00419 **
x.9           0.135941   0.044947   3.024  0.00262 **
x.10          0.083554   0.044849   1.863  0.06306 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11.06 on 486 degrees of freedom
Multiple R-squared:  0.4083,	Adjusted R-squared:  0.3949
F-statistic: 30.49 on 11 and 486 DF,  p-value: < 2.2e-16

AIC and BIC values for the model:
      AIC      BIC
1 3820.629 3875.367
```





Interpretation of Model-1

- The VIF results make it clear that multicollinearity is a problem with the estimations of DLM estimates. To solve this particular problem, we use a constrained least squares method for figuring out the parameters.
- The predictive model results included in the summary encountered significance tests, and it is clear that not all lag weights of predictors reach statistical significance at 5% level.

- F-test is also used to evaluate the modeling's overall statistical significance, and results show that the model does not attain statistical significance because the p-value is too high.
- As a result, we may conclude that the model does not adequately fit the data. The stated VIF values reveal multicollinearity has a significant impact.
- The screening check plots in all Figures show that the residuals are clearly trending and are not scattered randomly. Beusch-Godfrey test, with a 5% threshold of significance, confirms the serial correlation in the residuals that is shown by the ACF plot. The histogram and Shapiro-Wilk test show that the residuals' normality is not maintained (p-value 0.05). Ultimately, we may infer that further investigation of the lag 10 restricted DLM might not be required.

MODEL -2

Implement the Polynomial DLM model

Model for polynomial distributed lags we will give lag distribution a polynomial form to lessen the negative effects of multicollinearity. A balanced polynomial pattern for the lag weights is assumed. In honor of Shirley Almon, who first presented this idea, the final model is additionally referred to as a polynomial dispersed lag model or the Almon distributed lag framework.

By using polynomial curves to limit lag weights, we attempt to mitigate the multicollinearity problem in the distributed Lag Model (DLM). We use a function that can fit finite dlm with lag durations ranging from 1 to 10. The best lag length is then determined by ranking the models on the basis of AIC values.

A summary of the polyDLM model for multiple predictors of all indexes is given below. The multiple R-square is 0.3809 which means that 38.09 % of the variation is explained by independent variables in the model.

```
Estimates and t-tests for beta coefficients:
        Estimate Std. Error t value  P(>|t|)
beta.0    0.0045    0.00761   0.591 5.55e-01
beta.1    0.0111    0.00624   1.780 7.50e-02
beta.2    0.0178    0.00490   3.620 3.25e-04
beta.3    0.0244    0.00367   6.650 8.01e-11
beta.4    0.0310    0.00267  11.600 8.57e-28
beta.5    0.0376    0.00223  16.900 1.47e-50
beta.6    0.0443    0.00267  16.600 2.32e-49
beta.7    0.0509    0.00367  13.900 4.57e-37
beta.8    0.0575    0.00491  11.700 4.07e-28
beta.9    0.0642    0.00624  10.300 1.44e-22
beta.10   0.0708    0.00762   9.290 5.09e-19
```

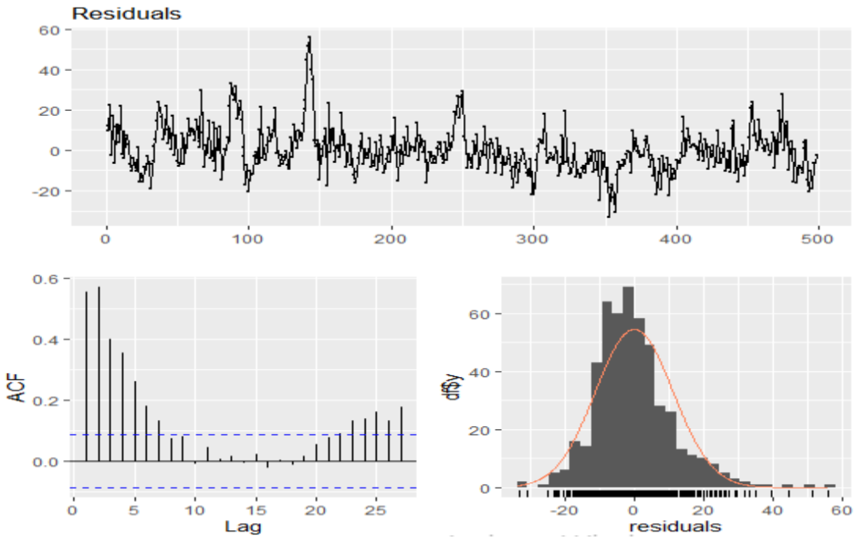
```
Call:
"Y ~ (Intercept) + X.t"

Residuals:
    Min      1Q  Median      3Q     Max
-33.084  -7.029  -1.371   5.217  55.845

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 94.353812   4.450571  21.200  < 2e-16 ***
z.t0         0.004500   0.007614   0.591    0.555
z.t1         0.006628   0.001456   4.551 6.71e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11.21 on 495 degrees of freedom
Multiple R-squared:  0.3809,    Adjusted R-squared:  0.3784
F-statistic: 152.3 on 2 and 495 DF,  p-value: < 2.2e-16
```

```
> vif(Model2.AllIndexes$model)
    z.t0     z.t1
11.62013 11.62013
```

```
        Shapiro-Wilk normality test

data:  x$residuals
W = 0.95049, p-value = 7.323e-12
```

```
        Breusch-Godfrey test for serial correlation of order up to 10

data:  Residuals
LM test = 208.15, df = 10, p-value < 2.2e-16
```

Variance inflation factor is greater than 10 which indicate that multicollinearity is high in model. Also the data is normal because Shapiro-wilk test gives p value is 7.323e-12<0.05.



The above figure represents the graphical form of model-2 for multiple predictors of all indexes.

PolyDLM for Chemical and particle size

Since particle size and chemical 1 has highest correlation. The F-statistic value for both chemical 1 and particle size gives p-value of < 2.2e-16 indicates that the overall model is statistically significant, and at least one of the independent variables is significantly related to dependent variable.

```
Call:
"Y ~ (Intercept) + X.t"

Residuals:
    Min      1Q  Median      3Q     Max
-28.459  -6.130  -0.489   4.694  48.855

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.420e+02  1.258e+00 112.836  < 2e-16 ***
z.t0        2.659e-01  5.457e-02   4.872 1.49e-06 ***
z.t1        8.688e-03  1.057e-02   0.822    0.412
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.936 on 495 degrees of freedom
Multiple R-squared:  0.5134,	Adjusted R-squared:  0.5114
F-statistic: 261.1 on 2 and 495 DF,  p-value: < 2.2e-16
```

```
Call:
"Y ~ (Intercept) + X.t"

Residuals:
    Min      1Q  Median      3Q     Max
-33.675  -6.288  -0.899   5.031  48.906

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.294e+02  2.084e+00  62.105  < 2e-16 ***
z.t0        5.002e-02  1.374e-02   3.642 0.000299 ***
z.t1        5.172e-03  2.636e-03   1.962 0.050303 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.7 on 495 degrees of freedom
Multiple R-squared:  0.4353,	Adjusted R-squared:  0.4331
F-statistic: 190.8 on 2 and 495 DF,  p-value: < 2.2e-16
```

```
> vif(model2.c1$model)>10
z.t0 z.t1
TRUE TRUE
> vif(model2.p$model)>10
z.t0 z.t1
TRUE TRUE
```

Interpretation of Model-2

Similar to finite DLM fitting, q = 10 has the lowest AIC and BIC values in the specified limit. We choose to set this polynomial order to 1 because it significantly reduces the amount of information needed. The predictive analysis shows that all lag weights (p-value > 0.05) are significant at the 5% level.

The general reliability assessment indicates that the predicted outcome is statistically noteworthy at the 5% level. The multicollinearity consequence of this model persists to be visible when VIF values are more than 10. The residuals' evaluation demonstrates that they are not dispersed arbitrarily. There are several fairly substantial delays on the ACF plot, which suggests autocorrelation in residuals.

Beusch-Godfrey test results with a 5% significance level back this up. The Shapiro-Wilk normality test result (p-value 0.05) and histogram both suggest that the residuals' normality is likewise broken. In conclusion, we conclude that further investigation into the polynomial dlm of lag 10 may not be necessary.

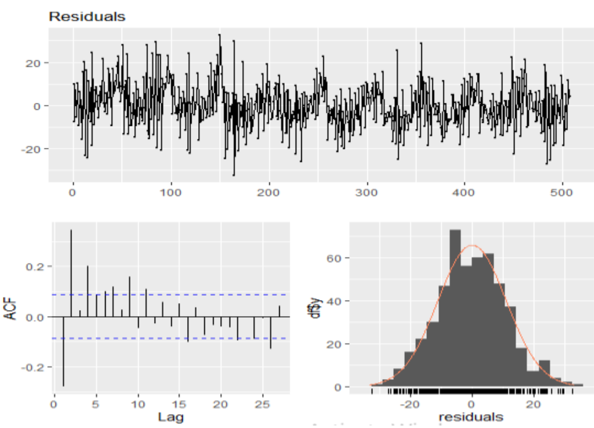
MODEL-3

KOYCK Transformation DL modeling

The summary of multiple indicators for all indexes is presented.

```
Call:
"Y ~ (Intercept) + Y.1 + X.t"

Residuals:
     Min       1Q   Median       3Q      Max
-32.5229  -7.2279  -0.1483   7.3809  32.5759

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 56.15359    8.44400   6.650 7.64e-11 ***
Y.1          0.76465    0.04130  18.513  < 2e-16 ***
X.t         -0.09106    0.05448  -1.671   0.0953 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11.11 on 504 degrees of freedom
Multiple R-Squared: 0.3892,     Adjusted R-squared: 0.3868
Wald test: 220.2 on 2 and 504 DF,  p-value: < 2.2e-16

Diagnostic tests:
NULL

                              alpha        beta       phi
Geometric coefficients:  238.5965 -0.09106318 0.7646504
```

```
        Shapiro-Wilk normality test

data:  x$residuals
W = 0.99735, p-value = 0.5989

> checkresiduals(Model3.AllIndexes$model)

        Ljung-Box test

data:  Residuals
Q* = 151.6, df [redacted] 10, p-value < 2.2e-16

Model df: 0.   Total lags used: 10
```




```
> vif(Model3.AllIndexes$model)
     Y.1      X.t
1.410584 1.410584
> vif(Model3.AllIndexes$model)>10
  Y.1   X.t
FALSE FALSE
```

The VIF indicates that there is no multicollinearity in this model, since results of VIF lie below lag value 10. And residuals plot also represents that no multicollinearity exists.

For Chemical 1 and Particle size

Chemical 1 and Particle size has no multicollinearity for this model since values lie below lag 10

```
> vif(model3.c1$model)
     Y.1      X.t
1.752114 1.752114
> vif(model3.c1$model)>10
  Y.1   X.t
FALSE FALSE
```

```
> vif(model3.p$model)
     Y.1      X.t
1.541246 1.541246
> vif(model3.p$model)>10
  Y.1   X.t
FALSE FALSE
```

```
> attr(model1.c1$model, "class") ="lm"
> AIC(model1.c1$model)
[1] 3694.7
> attr(model2.c1$model, "class") ="lm"
> AIC(model2.c1$model)
[1] 3705.251
> attr(model3.c1$model, "class") ="lm"
> AIC(model3.c1$model)
[1] 3673.683
```

Interpretation of Model-3

The paper states that the AIC measure for Model 3 is 3673.683, which is lower than both the finite and polynomial dlms. The ACF plot's lack of significant delays indicates that there is no serial correlation in the residuals. The Shapiro-Wilk normality test results show that the residuals do not correspond to normality (p-value 0.05), and the residuals' histogram is left-skew.

MODEL-4

AutoRegressive DLM

Neither polynomial nor Koyck DLMs offer satisfactory solutions, autoregressive DLMs come to our aid. The autoregressive DLM, which is essentially an infinite DLM with adaptability and efficiency, is explored. In our quest to replace Koyck model with a more suitable alternative, we proceed to fit autoregressive DLMs.

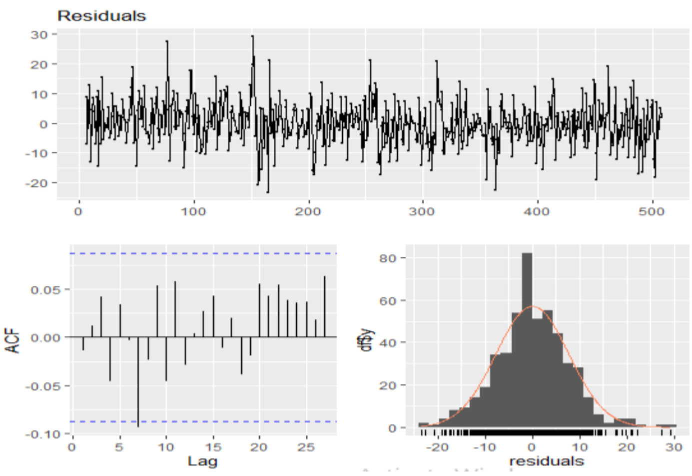
We use an iterative strategy to fit autoregressive DLMs with various lag durations and orders of autoregressive (AR) process. To determine the ARDL (p, q) parameters, models are chosen based on criteria that minimize information. Based on the information requirements, we decide to use the following models: ARDL (1, 5), ARDL (3, 5), ARDL (3, 3), ARDL (4, 5), and ARDL (5, 5).

```
p=  1 q=  1 AIC = 3600.181 BIC = 3629.781 Mase= 0.7738067
p=  1 q=  2 AIC = 3515.837 BIC = 3549.649 Mase= 0.7155269
p=  1 q=  3 AIC = 3511.638 BIC = 3549.66 Mase= 0.7155207
p=  1 q=  4 AIC = 3507.66 BIC = 3549.885 Mase= 0.7168988
p=  1 q=  5 AIC = 3502.287 BIC = 3548.713 Mase= 0.7155448
p=  2 q=  1 AIC = 3587.757 BIC = 3625.796 Mase= 0.7699201
p=  2 q=  2 AIC = 3517.355 BIC = 3559.62 Mase= 0.7124933
p=  2 q=  3 AIC = 3513.008 BIC = 3559.479 Mase= 0.7125774
p=  2 q=  4 AIC = 3509.048 BIC = 3559.718 Mase= 0.7141449
p=  2 q=  5 AIC = 3503.462 BIC = 3558.329 Mase= 0.7126694
p=  3 q=  1 AIC = 3582.97 BIC = 3629.44 Mase= 0.7681325
p=  3 q=  2 AIC = 3511.921 BIC = 3562.616 Mase= 0.7099816
p=  3 q=  3 AIC = 3513.823 BIC = 3568.742 Mase= 0.7103593
p=  3 q=  4 AIC = 3509.856 BIC = 3568.972 Mase= 0.7119373
p=  3 q=  5 AIC = 3504.101 BIC = 3567.41 Mase= 0.7103955
p=  4 q=  1 AIC = 3569.697 BIC = 3624.59 Mase= 0.7587092
p=  4 q=  2 AIC = 3500.931 BIC = 3560.047 Mase= 0.7063966
p=  4 q=  3 AIC = 3502.912 BIC = 3566.251 Mase= 0.7062284
p=  4 q=  4 AIC = 3504.195 BIC = 3571.756 Mase= 0.7069019
p=  4 q=  5 AIC = 3497.719 BIC = 3569.469 Mase= 0.7038418
p=  5 q=  1 AIC = 3559.026 BIC = 3622.335 Mase= 0.7528236
p=  5 q=  2 AIC = 3491.958 BIC = 3559.488 Mase= 0.7022435
p=  5 q=  3 AIC = 3493.94 BIC = 3565.69 Mase= 0.7025439
p=  5 q=  4 AIC = 3494.167 BIC = 3570.137 Mase= 0.7028216
p=  5 q=  5 AIC = 3495.741 BIC = 3575.932 Mase= 0.701704
```

By taking into account the previous discovery regarding model estimations, we might attempt to reduce latency for predictor series. As a result, a diagnostic checking and conclusion for ARDL (1, 5) has been performed below.

```
Time series regression with "ts" data:
Start = 6, End = 508

Call:
dynlm(formula = as.formula(model.text), data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-23.3680  -4.3568  -0.2709   4.6167  29.1267

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 48.8318550  8.5816092   5.690 2.18e-08 ***
temp.t       0.2345620  0.0648784   3.615 0.000331 ***
temp.1      -0.4009341  0.0648000  -6.187 1.29e-09 ***
X3.t         0.1508772  0.0350104   4.309 1.98e-05 ***
X3.1         0.0400318  0.0373657   1.071 0.284536
mortality.1  0.3843707  0.0437040   8.795  < 2e-16 ***
mortality.2  0.3454568  0.0442202   7.812 3.41e-14 ***
mortality.3 -0.0111228  0.0467470  -0.238 0.812030
mortality.4 -0.0009001  0.0431968  -0.021 0.983384
mortality.5  0.0127614  0.0406322   0.314 0.753600
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.773 on 493 degrees of freedom
Multiple R-squared:  0.7048,	Adjusted R-squared:  0.6994
F-statistic: 130.8 on 9 and 493 DF,  p-value: < 2.2e-16
```

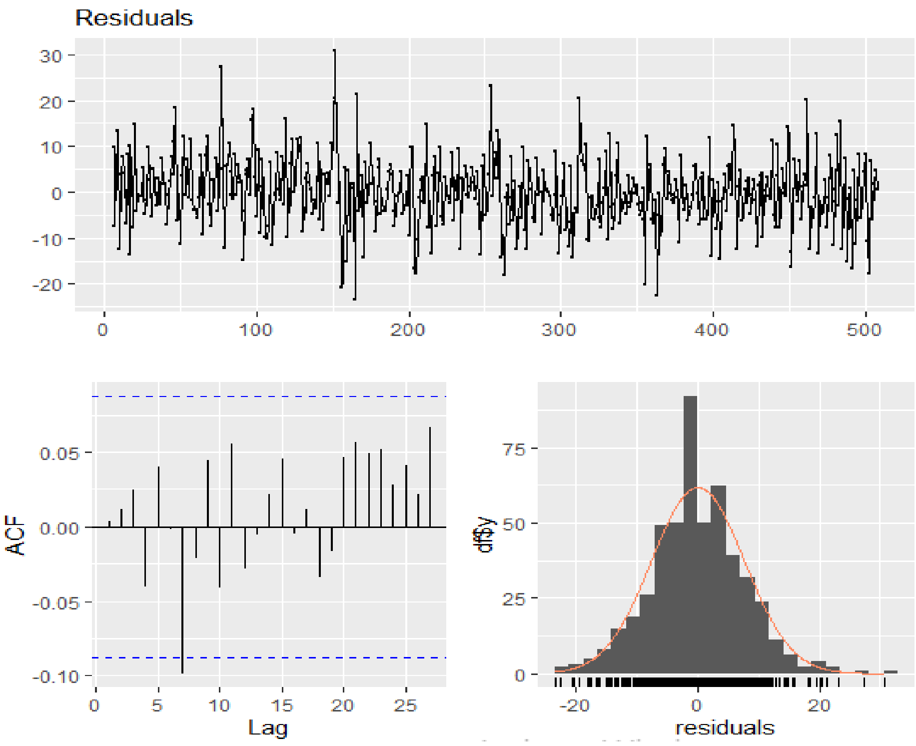
```
> residualcheck(model4_1)

	Shapiro-Wilk normality test

data:  x$residuals
W = 0.99236, p-value = 0.01126

> checkresiduals(model4_1)

	Breusch-Godfrey test for serial correlation of order up to 13

data:  Residuals
LM test = 22.475, df = 13, p-value = 0.04843
```



ARDL (3, 5) summary and **diagnostic** checking is performed below.

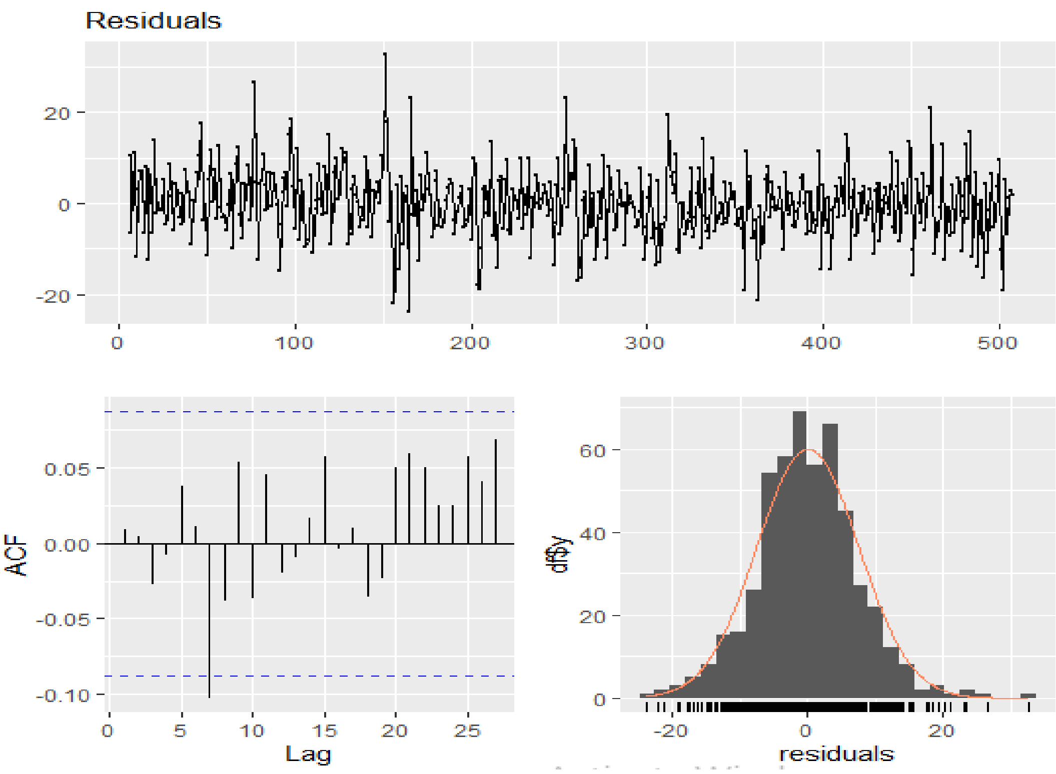
```
Time series regression with "ts" data:
Start = 6, End = 508

Call:
dynlm(formula = as.formula(model.text), data = data)

Residuals:
     Min       1Q   Median       3Q      Max 
-23.3883  -4.7313  -0.5274   4.6396  30.7928 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 49.902323   9.394500   5.312 1.65e-07 ***
temp.t       0.257809   0.071916   3.585 0.000371 ***
temp.1      -0.391851   0.076784  -5.103 4.79e-07 ***
temp.2      -0.159586   0.077322  -2.064 0.039552 *  
temp.3       0.112510   0.071889   1.565 0.118217    
X3.t         0.146039   0.040186   3.634 0.000309 ***
X3.1         0.040009   0.042019   0.952 0.341482    
X3.2         0.065371   0.043356   1.508 0.132262    
X3.3        -0.070694   0.043067  -1.641 0.101340    
mortality.1  0.373650   0.045459   8.219 1.85e-15 ***
mortality.2  0.362654   0.048441   7.486 3.32e-13 ***
mortality.3 -0.019020   0.049908  -0.381 0.703295    
mortality.4  0.003915   0.043647   0.090 0.928568    
mortality.5  0.012370   0.040887   0.303 0.762369    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.757 on 489 degrees of freedom
Multiple R-squared:  0.7084,	Adjusted R-squared:  0.7006 
F-statistic: 91.38 on 13 and 489 DF,  p-value: < 2.2e-16
```

```
> residualcheck(model4_3)

	Shapiro-Wilk normality test

data:  x$residuals
W = 0.99118, p-value = 0.00429

> checkresiduals(model4_3)

	Breusch-Godfrey test for serial correlation of order up to 17

data:  Residuals
LM test = 23.157, df = 17, p-value = 0.1442
```



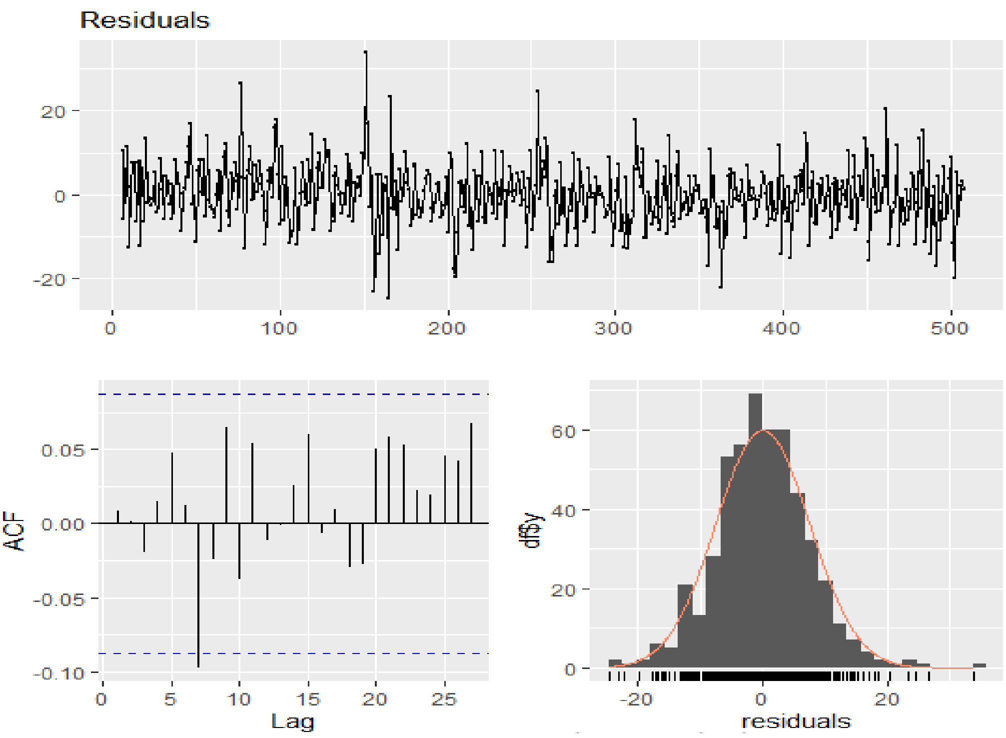
```
Time series regression with "ts" data:
Start = 6, End = 508

Call:
dynlm(formula = as.formula(model.text), data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-23.796  -4.699  -0.135   4.515  32.772

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 45.12282    9.51404   4.743 2.77e-06 ***
temp.t       0.24000    0.07193   3.336 0.000913 ***
temp.1      -0.37906    0.07664  -4.946 1.04e-06 ***
temp.2      -0.16432    0.08032  -2.046 0.041300 *
temp.3       0.05333    0.07699   0.693 0.488839
temp.4       0.11945    0.07203   1.658 0.097881 .
X3.t         0.13495    0.04013   3.363 0.000832 ***
X3.1         0.01955    0.04243   0.461 0.645153
X3.2         0.04145    0.04474   0.927 0.354642
X3.3        -0.05975    0.04392  -1.360 0.174327
X3.4         0.05555    0.04307   1.290 0.197769
mortality.1  0.37378    0.04528   8.255 1.44e-15 ***
mortality.2  0.37034    0.04823   7.678 8.93e-14 ***
mortality.3  0.01591    0.05095   0.312 0.755018
mortality.4 -0.04520    0.04595  -0.984 0.325787
mortality.5  0.02177    0.04103   0.531 0.595853
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.693 on 487 degrees of freedom
Multiple R-squared:  0.7143,    Adjusted R-squared:  0.7055
F-statistic: 81.19 on 15 and 487 DF,  p-value: < 2.2e-16
```

```
> residualcheck(model4_4)

        Shapiro-Wilk normality test

data:  x$residuals
W = 0.98976, p-value = 0.001412

> checkresiduals(model4_4)

        Breusch-Godfrey test for serial correlation of order up to 19

data:  Residuals
LM test = 25.434, df = 19, p-value = 0.1468
```



```
Time series regression with "ts" data:
Start = 6, End = 508

Call:
dynlm(formula = as.formula(model.text), data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-24.581  -4.662  -0.103   4.512  33.750

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 47.413653   9.557779   4.961 9.73e-07 ***
temp.t       0.234956   0.072235   3.253  0.00122 **
temp.1      -0.350011   0.077496  -4.516 7.90e-06 ***
temp.2      -0.147893   0.080307  -1.842  0.06614 .
temp.3       0.111423   0.080523   1.384  0.16707
temp.4       0.184527   0.077607   2.378  0.01781 *
temp.5      -0.170871   0.073385  -2.328  0.02030 *
X3.t         0.154439   0.041214   3.747  0.00020 ***
X3.1         0.009170   0.042517   0.216  0.82933
X3.2         0.030056   0.045080   0.667  0.50527
X3.3        -0.080912   0.045500  -1.778  0.07598 .
X3.4         0.029117   0.044302   0.657  0.51133
X3.5         0.078351   0.043189   1.814  0.07028 .
mortality.1  0.376073   0.045390   8.285 1.16e-15 ***
mortality.2  0.378775   0.048268   7.847 2.74e-14 ***
mortality.3  0.004942   0.051036   0.097  0.92290
mortality.4 -0.069436   0.047768  -1.454  0.14671
mortality.5  0.027875   0.043492   0.641  0.52188
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.663 on 485 degrees of freedom
Multiple R-squared:  0.7177,    Adjusted R-squared:  0.7078
F-statistic: 72.54 on 17 and 485 DF,  p-value: < 2.2e-16
```

```
> residualcheck(model4_5)

        Shapiro-Wilk normality test

data:  x$residuals
W = 0.98964, p-value = 0.001288

> checkresiduals(model4_5)

        Breusch-Godfrey test for serial correlation of order up to 21

data:  Residuals
LM test = 20.456, df = 21, p-value = 0.4926
```



Variance Inflation Factor

```
> vif(model4_1)>10
            temp        L(temp, 1)                X3         L(X3, 1) L(mortality, 1)
           FALSE             FALSE             FALSE             FALSE             FALSE
L(mortality, 2) L(mortality, 3) L(mortality, 4) L(mortality, 5)
           FALSE             FALSE             FALSE             FALSE
> vif(model4_2)>10
Error: object 'model4_2' not found
> vif(model4_3)>10
            temp        L(temp, 1)        L(temp, 2)        L(temp, 3)                X3
           FALSE             FALSE             FALSE             FALSE             FALSE
        L(X3, 1)          L(X3, 2)          L(X3, 3) L(mortality, 1) L(mortality, 2)
           FALSE             FALSE             FALSE             FALSE             FALSE
L(mortality, 3) L(mortality, 4) L(mortality, 5)
           FALSE             FALSE             FALSE
> vif(model4_4)>10
            temp        L(temp, 1)        L(temp, 2)        L(temp, 3)        L(temp, 4)
           FALSE             FALSE             FALSE             FALSE             FALSE
              X3          L(X3, 1)          L(X3, 2)          L(X3, 3)          L(X3, 4)
           FALSE             FALSE             FALSE             FALSE             FALSE
L(mortality, 1) L(mortality, 2) L(mortality, 3) L(mortality, 4) L(mortality, 5)
           FALSE             FALSE             FALSE             FALSE             FALSE
> vif(model4_5)>10
            temp        L(temp, 1)        L(temp, 2)        L(temp, 3)        L(temp, 4)
           FALSE             FALSE             FALSE             FALSE             FALSE
      L(temp, 5)                X3          L(X3, 1)          L(X3, 2)          L(X3, 3)
           FALSE             FALSE             FALSE             FALSE             FALSE
        L(X3, 4)          L(X3, 5) L(mortality, 1) L(mortality, 2) L(mortality, 3)
           FALSE             FALSE             FALSE             FALSE             FALSE
L(mortality, 4) L(mortality, 5)
           FALSE             FALSE
```

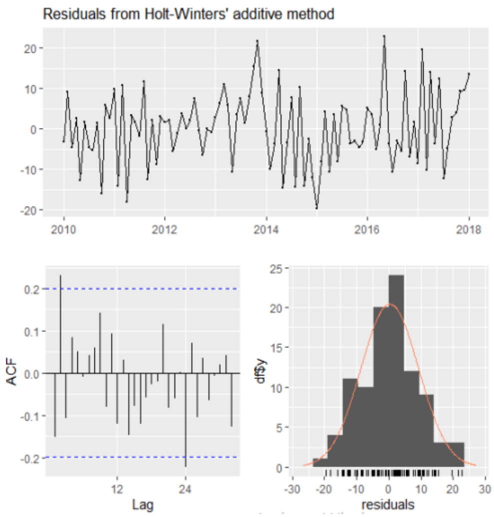
Interpretation of Model-04

According to reports, all constructed ARDL models are considered significant at a level of 5%. ARDL (1, 5) was a slightly superior model. ACF and Beusch-Godfrey test results show that there is no autocorrelation in residuals, indicating that the suggested model is suitable. The error terms are randomly distributed because show no discernable pattern. According to the histogram and Shapiro, the normality assumption wasn't evident. All ARDL models fitted, however, have no multicollinearity when VIFs are less than 10. In terms of multicollinearity, we were able to locate an ARDL model overall.

MODEL 5

Exponential Smoothing

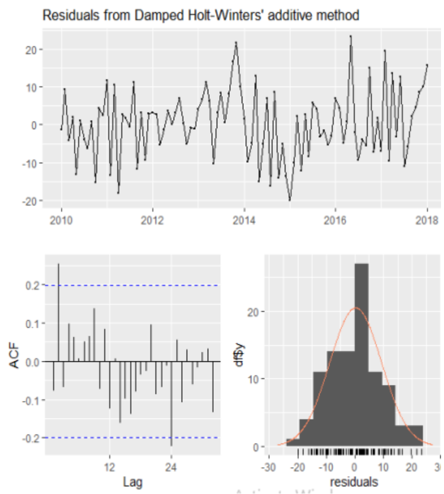
```
Forecast method: Holt-Winters' additive method

Model Information:
Holt-Winters' additive method

Call:
 hw(y = Mort1)

  Smoothing parameters:
    alpha = 0.3745
    beta  = 2e-04
    gamma = 3e-04

  Initial states:
    l = 184.2616
    b = -0.0536
    s = -2.2954 1.6241 1.5959 1.0561 3.6228 -4.9819
           1.4077 0.8779 -2.5386 -1.7245 -1.1711 2.527

  sigma:  9.7528

     AIC     AICc      BIC
902.1062 909.8531 945.8763

Error measures:
                    ME     RMSE      MAE         MPE     MAPE      MASE
Training set 0.2677826 8.912181 7.183698 -0.05164846 4.016544 0.5333901
                   ACF1
Training set -0.1512674
```




```
        Ljung-Box test

data:  Residuals from Holt-Winters' additive method
Q* = 21.61, df = 19, p-value = 0.3041

Model df: 0.   Total lags used: 19
```

The above summary represents Residuals from Holt-Winters' additive method indicates that residuals are insignificant with p-value= 0.3041. The above summary gives values of AIC, BIC and MASE for exponential smoothing model.

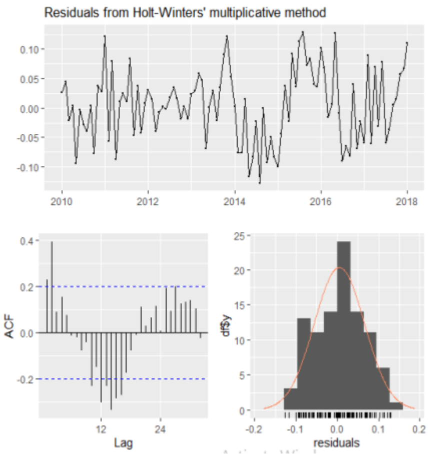
```
Forecast method: Damped Holt-Winters' additive method

Model Information:
Damped Holt-Winters' additive method

Call:
 hw(y = Mortl, h = 5 * frequency(Mortl), seasonal = "additive",

 Call:
     damped = TRUE)

  Smoothing parameters:
    alpha = 0.316
    beta  = 0.0073
    gamma = 0.0148
    phi   = 0.9673

  Initial states:
    l = 183.7259
    b = -0.0779
    s = -2.0229 2.399 1.0386 1.3208 4.3478 -5.6764
           1.5674 1.2689 -2.1265 -1.8105 -1.5492 1.243

  sigma:  9.9419

     AIC     AICc      BIC
906.6279 915.3971 952.9727

Error measures:
                    ME     RMSE      MAE         MPE     MAPE      MASE
Training set 0.2763798 9.028779 7.251837 -0.04298571 4.050838 0.5384495
                    ACF1
Training set -0.07685818
```




```
> checkresiduals(hw3)

        Ljung-Box test

data:  Residuals from Damped Holt-Winters' additive method
Q* = 22.378, df = 19, p-value = 0.2658

Model df: 0.   Total lags used: 19
```

The above summary represents Residuals from Damped Holt-Winters' additive method gives p-value = 0.2658 and the AIC, BIC, and MASE values.

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

Model Information:
Holt-Winters' multiplicative method

Call:
 hw(y = Mortl, seasonal = "multiplicative")

  Smoothing parameters:
    alpha = 0.1346
    beta  = 0.0538
    gamma = 0.1487

  Initial states:
    l = 183.8001
    b = -0.3035
    s = 0.9883 1.0033 1.0058 1.0092 1.0286 0.9722
           1.0093 1.0317 0.9932 0.9909 0.993 0.9744

  sigma:  0.0664

     AIC     AICc      BIC
938.1489 945.8957 981.9190
```

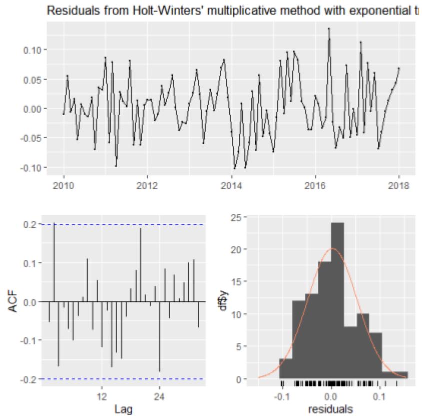



```
> checkresiduals(hw2)

        Ljung-Box test

data:  Residuals from Holt-Winters' multiplicative method
Q* = 86.366, df = 19, p-value = 1.451e-10

Model df: 0.   Total lags used: 19
```

The above summary represents Residuals from Holt-Winters' multiplicative method with p-value = 1.451e-10 and the values of AIC, BIC, and MASE.

```
Forecast method: Holt-Winters' multiplicative method with exponential trend

Model Information:
Holt-Winters' multiplicative method with exponential trend

Call:
 hw(y = Mortl, h = 5 * frequency(Mortl), seasonal = "multiplicative",

 Call:
     exponential = TRUE)

  Smoothing parameters:
    alpha = 0.2196
    beta  = 0.1042
    gamma = 1e-04

  Initial states:
    l = 184.2529
    b = 0.9942
    s = 0.9925 1.0159 1.0022 1.0041 1.0145 0.9666
           1.0129 0.9992 0.9942 0.9876 0.9975 1.0128

  sigma:  0.0556

     AIC     AICc      BIC
903.8814 911.6283 947.6515
```

```
> checkresiduals(hw5)

        Ljung-Box test

data:  Residuals from Holt-Winters' multiplicative method with exponential trend
Q* = 22.109, df = 19, p-value = 0.2789

Model df: 0.   Total lags used: 19
```



The above summary represents residuals from Holt-Winters' trend with p-value = 0.2789 which indicates that residuals are insignificant and the values of AIC, BIC and MASE are also listed.

Interpretation of Model 6

We update the accuracy data frame with the measurements of accuracy from the exponentially smoothed approaches.

Model 7

State-space variation

```
ETS(A,N,N)

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

  Smoothing parameters:
    alpha = 0.5081

  Initial states:
    l = 184.6573

  sigma:  9.074

     AIC     AICc      BIC
5409.782 5409.830 5422.474

Training set error measures:
                      ME     RMSE      MAE        MPE     MAPE      MASE
Training set -0.06480644 9.056138 7.121406 -0.2446192 4.196617 0.7025894
                    ACF1
Training set -0.08930403
```

The brief summary of Artificial Neural Network (ANN) is presented which gives the values of AIC, BIC and MASE.

```
ETS(M,N,N)

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

  Smoothing parameters:
    alpha = 0.4818

  Initial states:
    l = 184.0437

  sigma:  0.0526

     AIC     AICc      BIC
5386.809 5386.857 5399.500

Training set error measures:
                      ME     RMSE      MAE        MPE     MAPE      MASE
Training set -0.06720288 9.061271 7.121783 -0.2494414 4.194287 0.7026266
                    ACF1
Training set -0.05776727
```

```
ETS(A,A,N)

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

  Smoothing parameters:
    alpha = 0.5151
    beta  = 1e-04

  Initial states:
    l = 191.6977
    b = -0.0657

  sigma:  9.1

     AIC     AICc      BIC
5414.671 5414.790 5435.823

Training set error measures:
                     ME     RMSE      MAE        MPE     MAPE      MASE
Training set 0.03893265 9.064061 7.135536 -0.1804909 4.203038 0.7039834
                    ACF1
Training set -0.09692901
```

TASK 2

Introduction

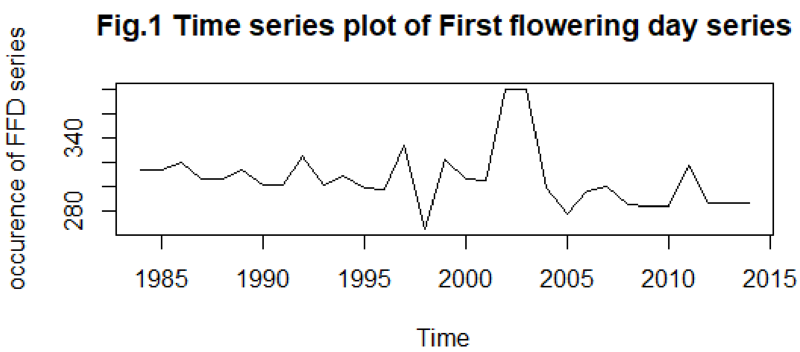
Hudson and Keatley (2021) examined whether climate variables like Temperature (temp), Rainfall (rain), Relative humidity (RH), and Radiation level (rad) have an impact on the day of occurrence of a species' first flowering (first flowering day, FFD, a number between 1 and 365).

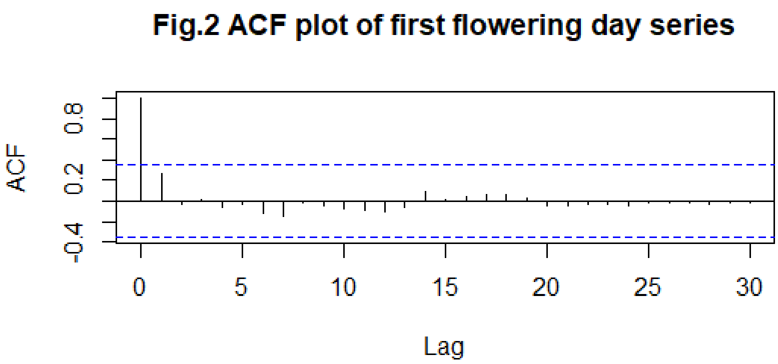
Problem Statement

Using the annual FFD series as an explanatory series, we will fit distributed lag models using the time series regression approach for this assignment. Best-performing model for each approach will then be used to produce point forecasts, build confidence intervals, and visually display outcomes. In addition to these strategies, we will also use state-space models and exponential smoothing to predict solar radiation data. Then, we will evaluate these alternative approaches by gauging quality-of-fit indicators and residual assumptions.

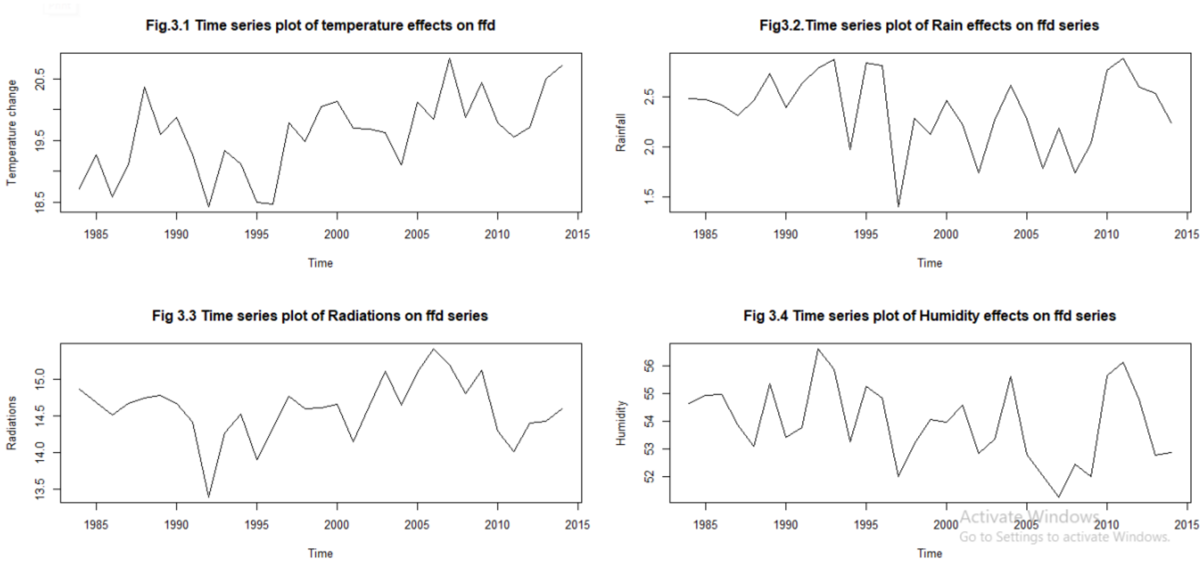
Using the model with the lowest mean absolute scaled error (MASE) value, forecasts for the following four years are the primary objective of this study. Data comprises 5-time series, the FFD time series of the selected plant species, and the contemporaneous annual averaged climatic variables observed from 1984 to 2014 (31 years). The data is specific to one species (out of the 81 species). Here, you may get all of the series in “T2_FFD.csv"

Data exploration and visualization



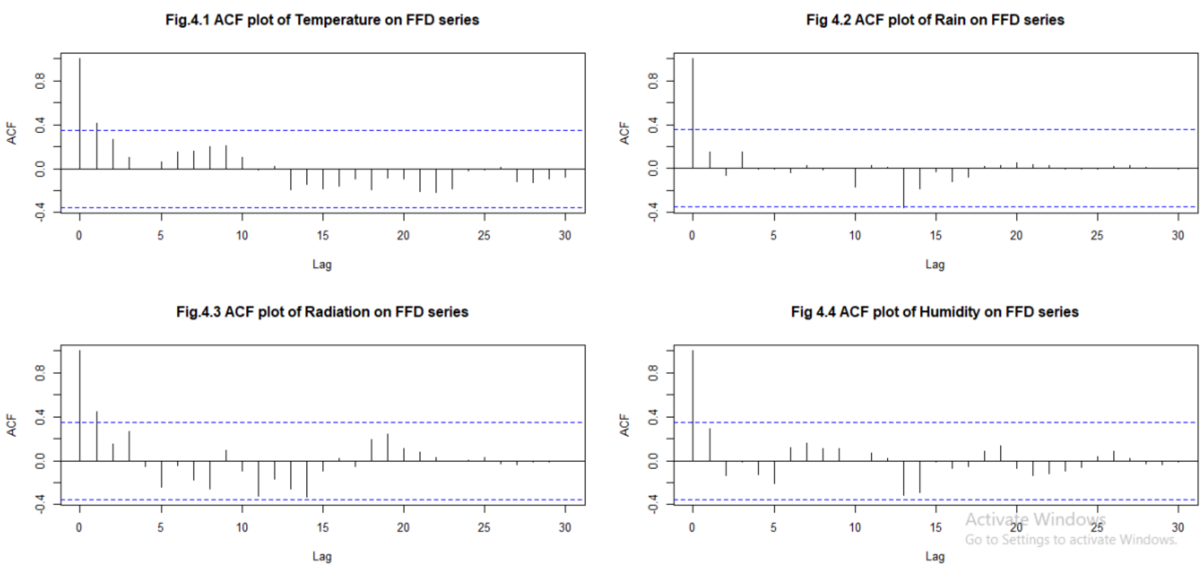


Below is a series of time graphic of variables influencing FFD that will be used as a predictive sequence for dispersed lag representations.



Interpretation

- Particularly in the start of the series, there may be a modest downward tendency.
- • Despite the fact that the pattern changes with time, we can still infer that higher values are seen in December and January and lower values are seen in July and August. It is challenging to identify the presence of shifting variation and series behavior because of seasonality.
- No obvious areas for action exist.
- We will make an instance of an ACF plot and run an ADF test across the series to more completely explore the pattern and seasonality of elements in precipitation data.

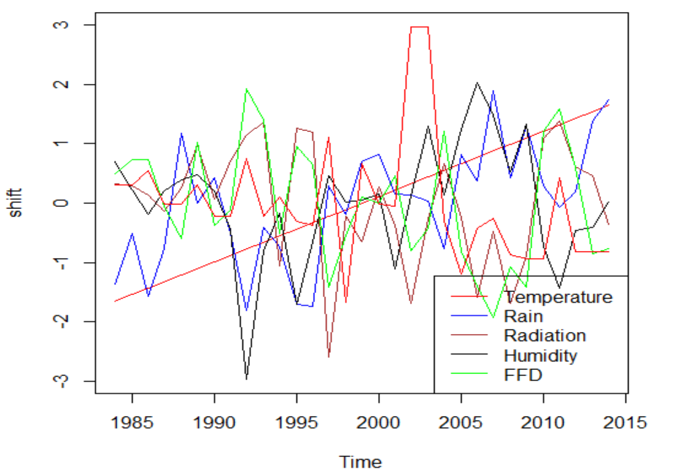


A small seasonal pattern in temperature, a pattern of declining rainfall, and diminishing seasonal delays in radiation and humidity are all signs that a trend is likely to exist. The Augmented Dickey-Fuller test (ADF) results are as follows:

- The test results indicate that the pattern of data is nonstationary at a 5% level of significance with a lag of 2 and a p-value of 0.1902.
- The test results at a 5% level of significance show that the pattern is stationary with a lag of 0 and a p-value of 0.01.
- The test indicates that, at a 5% level of significance, the pattern of data is nonstationary with a lag of 4 and a p-value of 0.2911.
- At a 5% level of significance, a lag of 0, and a p-value of 0.01789, the test confirms that the data pattern is stationary.

We will standardize results such that dependent solar series and explanatory precipitation series may be easily seen side by side on the same display. The diagram of time series made using the code below looks at the connections between series.

Fig.5 FFD rate versus factor affecting ffd wrt time(Scaled)



The dependent and independent series are more likely to have a negative correlation, as seen in the preceding Figure. Low precipitation values are correlated with high radiation levels, and vice versa.

To confirm link, we compute correlation coefficient.

```
> cor(ffdata_ts)
                  Year Temperature   Rainfall    Radiation RelHumidity
Year         1.0000000   0.5944486 -0.1752091  0.118818288 -0.29949678
Temperature  0.5944486   1.0000000 -0.3915072  0.519357599 -0.66219798
Rainfall    -0.1752091  -0.3915072  1.0000000 -0.581316101  0.79110786
Radiation    0.1188183   0.5193576 -0.5813161  1.000000000 -0.73540867
RelHumidity -0.2994968  -0.6621980  0.7911079 -0.735408669  1.00000000
FFD         -0.2476550  -0.1984473 -0.2203478  0.003593172  0.06017675
                    FFD
Year        -0.247654950
Temperature -0.198447254
Rainfall    -0.220347796
Radiation    0.003593172
RelHumidity  0.060176754
FFD          1.000000000
```

The inference made from the plot in Figure’s legend is supported by the correlation coefficient, which shows that FFD has r=-0.19844 w.r.t. temperature. This suggests that there is a generally negative association between the series. We go on to the modeling stage after examining the traits of several series and identifying any indications of a relationship between them.

TIME SERIES REGRESSION TECHNIQUES

MODEL 1

Model for finite distributed lag (DLM)

With the eventual objective of finding a reliable model for **forecasting** solar radiation levels, this method seeks to improve our understanding of the general variance and correlation structure within the time series.

We use a methodical approach to doing this. For models with varied lag durations, a loop computing several accuracy measures was built, including AIC/BIC and the value of MASE. Then, in order to find the ideal lag time for our model, we choose the model with the most favorable metrics, namely with the shortest values.

```
q = 1 AIC = 285.3891 BIC = 290.9939 MASE = 0.8692428
q = 2 AIC = 278.8658 BIC = 285.7023 MASE = 0.8414928
q = 3 AIC = 271.9603 BIC = 279.9535 MASE = 0.8436984
q = 4 AIC = 265.5399 BIC = 274.6107 MASE = 0.8449204
q = 5 AIC = 250.7426 BIC = 260.8074 MASE = 0.7604314
q = 6 AIC = 243.4304 BIC = 254.4003 MASE = 0.7638545
q = 7 AIC = 236.8679 BIC = 248.6485 MASE = 0.7578434
q = 8 AIC = 230.7683 BIC = 243.2588 MASE = 0.7758607
q = 9 AIC = 222.6794 BIC = 235.7719 MASE = 0.7489874
q = 10 AIC = 209.0075 BIC = 222.5863 MASE = 0.6442552
```

We shall fit a finite DLM with a number of delays equal to 10 because it has been demonstrated that scores of the information criteria and MASE decline as q grows.

Finite DLM

1) Temperature

```
Call:
lm(formula = model.formula, data = design)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.38057 -0.05091  0.20664  0.26992  0.51763 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)  
(Intercept) 24.5432126  9.3888138   2.614   0.0281 *
x.t         -0.0010231  0.0060331  -0.170   0.8691  
x.1         -0.0060863  0.0064391  -0.945   0.3692  
x.2         -0.0041677  0.0064212  -0.649   0.5325  
x.3         -0.0001096  0.0068612  -0.016   0.9876  
x.4          0.0003766  0.0066847   0.056   0.9563  
x.5         -0.0004193  0.0068845  -0.061   0.9528  
x.6          0.0042511  0.0071537   0.594   0.5670  
x.7         -0.0035732  0.0070779  -0.505   0.6258  
x.8          0.0009756  0.0071723   0.136   0.8948  
x.9         -0.0104321  0.0068725  -1.518   0.1633  
x.10         0.0046656  0.0078937   0.591   0.5690  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7396 on 9 degrees of freedom
Multiple R-squared:  0.3618,	Adjusted R-squared:  -0.4183 
F-statistic: 0.4638 on 11 and 9 DF,  p-value: 0.8852

AIC and BIC values for the model:
       AIC      BIC
1 55.13358 68.71237
```

```
> vif(ftem_dlm$model)
     x.t      x.1      x.2      x.3      x.4      x.5      x.6      x.7 
1.174353 1.311746 1.300256 1.448418 1.370890 1.403890 1.452937 1.365903 
     x.8      x.9     x.10 
1.398261 1.261904 1.514276 
```

We derived Adjusted R-squared: -0.4183, p-value: 0.8852 > 0.05, and AIC: 55.13358 from temperature dataset.

2) Rain

```
> summary(frain_dlm)

Call:
lm(formula = model.formula, data = design)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.58612 -0.10785  0.02393  0.09722  0.55713 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)  
(Intercept)  1.638e+00  4.552e+00   0.360   0.7272  
x.t         -5.448e-03  2.925e-03  -1.863   0.0954 .
x.1          4.354e-03  3.122e-03   1.395   0.1965  
x.2         -4.801e-05  3.113e-03  -0.015   0.9880  
x.3         -1.449e-03  3.326e-03  -0.436   0.6734  
x.4          1.604e-03  3.241e-03   0.495   0.6324  
x.5         -5.210e-03  3.337e-03  -1.561   0.1530  
x.6         -2.769e-03  3.468e-03  -0.798   0.4452  
x.7         -2.411e-05  3.431e-03  -0.007   0.9945  
x.8          6.375e-03  3.477e-03   1.834   0.0999 .
x.9          5.255e-04  3.332e-03   0.158   0.8782  
x.10         4.012e-03  3.827e-03   1.048   0.3218  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3585 on 9 degrees of freedom
Multiple R-squared:  0.6522,	Adjusted R-squared:  0.227 
F-statistic: 1.534 on 11 and 9 DF,  p-value: 0.265

AIC and BIC values for the model:
       AIC      BIC
1 24.72299 38.30178
> vif(frain_dlm$model)
     x.t      x.1      x.2      x.3      x.4      x.5      x.6      x.7 
1.174353 1.311746 1.300256 1.448418 1.370890 1.403890 1.452937 1.365903 
     x.8      x.9     x.10 
1.398261 1.261904 1.514276 
```

We found Adjusted R-squared: 0.227, p-value: 0.265 > 0.05, and AIC: 24.7229 from rainfall dataset.

3) Radiation

```
Call:
lm(formula = model.formula, data = design)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.79729 -0.03552  0.06608  0.12004  0.26464 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)  
(Intercept)  7.0266674  4.1943660   1.675   0.1282  
x.t          0.0022792  0.0026952   0.846   0.4197  
x.1          0.0013272  0.0028766   0.461   0.6555  
x.2          0.0004104  0.0028686   0.143   0.8894  
x.3          0.0065195  0.0030652   2.127   0.0623 .
x.4          0.0040333  0.0029863   1.351   0.2098  
x.5          0.0024292  0.0030756   0.790   0.4499  
x.6          0.0065009  0.0031959   2.034   0.0724 .
x.7          0.0018960  0.0031620   0.600   0.5636  
x.8         -0.0014503  0.0032041  -0.453   0.6615  
x.9         -0.0023894  0.0030702  -0.778   0.4564  
x.10         0.0031291  0.0035264   0.887   0.3980  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3304 on 9 degrees of freedom
Multiple R-squared:  0.6858,	Adjusted R-squared:  0.3018 
F-statistic: 1.786 on 11 and 9 DF,  p-value: 0.1965

AIC and BIC values for the model:
       AIC      BIC
1 21.29096 34.86975
> vif(frad_dlm$model)
     x.t      x.1      x.2      x.3      x.4      x.5      x.6      x.7 
1.174353 1.311746 1.300256 1.448418 1.370890 1.403890 1.452937 1.365903 
     x.8      x.9     x.10 
1.398261 1.261904 1.514276 
```

We found Adjusted R-squared: 0.3018, p-value: 0.1965>0.05, and AIC: 21.2906 using rain dataset.

4) **Humidity**

```
Call:
lm(formula = model.formula, data = design)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.2147 -0.4170 -0.1015  0.1246  1.7356 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)   
(Intercept) 54.608499  12.513137   4.364  0.00181 **
x.t         -0.006891   0.008041  -0.857  0.41370   
x.1          0.011317   0.008582   1.319  0.21983   
x.2          0.006507   0.008558   0.760  0.46653   
x.3         -0.009202   0.009144  -1.006  0.34055   
x.4         -0.005796   0.008909  -0.651  0.53157   
x.5         -0.009687   0.009175  -1.056  0.31862   
x.6         -0.023408   0.009534  -2.455  0.03645 * 
x.7          0.009532   0.009433   1.010  0.33868   
x.8          0.012853   0.009559   1.345  0.21167   
x.9          0.020321   0.009159   2.219  0.05369 . 
x.10        -0.008966   0.010520  -0.852  0.41618   
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9857 on 9 degrees of freedom
Multiple R-squared:  0.7752,	Adjusted R-squared:  0.5005 
F-statistic: 2.822 on 11 and 9 DF,  p-value: 0.06515

AIC and BIC values for the model:
       AIC      BIC
1 67.19851 80.7773
> vif(fhum_dlm$model)
     x.t      x.1      x.2      x.3      x.4      x.5      x.6      x.7 
1.174353 1.311746 1.300256 1.448418 1.370890 1.403890 1.452937 1.365903 
     x.8      x.9     x.10 
1.398261 1.261904 1.514276 
```

We achieved Adjusted R-squared: 0.5005, p-value: 0.06515 > 0.05, and AIC: 67.19851 from the humidity dataset.

Interpretation of Model 1

According to the results of the tests of significance performed on coefficients of model produced from the summary, the majority of lag weights for the predictor series do not reach statistical significance at the 5% level. This finding leads us to the conclusion that the model does not adequately fit data, mainly as a result of the inclusion of non-significant components and its low explanatory power. It should be noted that the model has a low level of multicollinearity, as shown by VIF values of 10.

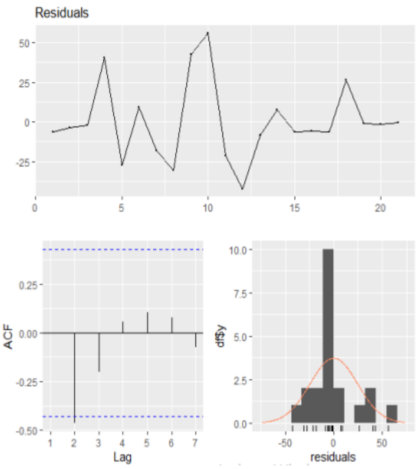
MODEL 2

Polynomial distributed lag model

Polynomial modeling on univariate

1) Temperature

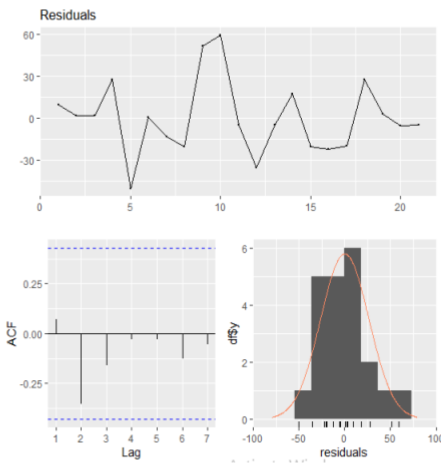
```
> summary(temp_polyd)

Call:
"Y ~ (Intercept) + X.t"

Residuals:
    Min      1Q  Median      3Q     Max
-42.389  -8.563  -3.514   7.604  55.874

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.233e+03  3.929e+02   3.138    0.006 **
z.t0         5.676e+00  5.915e+00   0.960    0.351
z.t1        -2.033e+00  2.709e+00  -0.750    0.463
z.t2         3.899e-03  2.516e-01   0.016    0.988
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 26.5 on 17 degrees of freedom
Multiple R-squared:  0.3236,    Adjusted R-squared:  0.2042
F-statistic: 2.711 on 3 and 17 DF,  p-value: 0.07751

> residualcheck(temp_polyd$model)

        Shapiro-Wilk normality test

data:  x$residuals
W = 0.92374, p-value = 0.1031

> checkresiduals(temp_polyd$model)

        Breusch-Godfrey test for serial correlation of order up to 7

data:  Residuals
LM test = 9.3808, df = 7, p-value = 0.2265
```



We achieved Adjusted R-squared: 0.2042, p-value: 0.07751 >0.05 from temperature dataset.

2) Rainfall

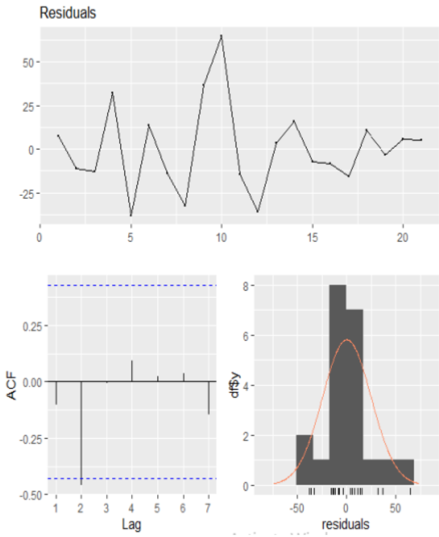
```
> summary(rain_polyd)

Call:
"Y ~ (Intercept) + X.t"

Residuals:
    Min      1Q  Median      3Q     Max
-50.305 -19.946  -4.664   9.431  59.330

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 250.8970   113.7004   2.207   0.0414 *
z.t0         -7.8772    12.2458  -0.643   0.5286
z.t1         -0.7171     6.1659  -0.116   0.9088
z.t2          0.3830     0.6432   0.595   0.5594
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 28.63 on 17 degrees of freedom
Multiple R-squared:  0.2107,    Adjusted R-squared:  0.07137
F-statistic: 1.512 on 3 and 17 DF,  p-value: 0.2473

> residualcheck(rain_polyd$model)

        Shapiro-Wilk normality test

data:  x$residuals
W = 0.95603, p-value = 0.4401

> checkresiduals(rain_polyd$model)

        Breusch-Godfrey test for serial correlation of order up to 7

data:  Residuals
LM test = 9.4203, df = 7, p-value = 0.2239
```



From rainfall data, we derived Adjusted R-squared: 0.07137, p-value: 0.2473 >0.05.

3) Radiation

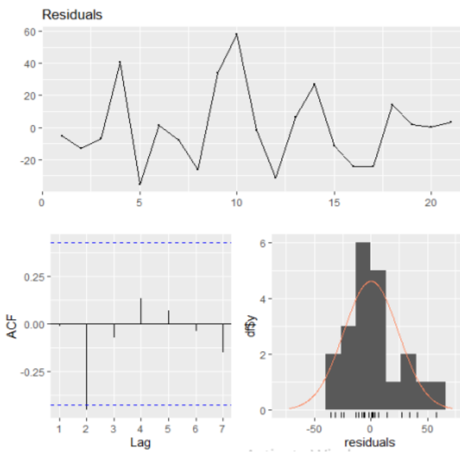
```
> summary(rad_polyd)

Call:
"Y ~ (Intercept) + X.t"

Residuals:
    Min      1Q  Median      3Q     Max 
-38.108 -14.250  -3.279  10.779  64.505 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)   
(Intercept) 1599.7417   501.1976   3.192  0.00534 **
z.t0         -10.5586     8.5329  -1.237  0.23276   
z.t1           4.5066     4.1083   1.097  0.28796   
z.t2          -0.5731     0.4160  -1.378  0.18619   
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 26.69 on 17 degrees of freedom
Multiple R-squared:  0.3138,	Adjusted R-squared:  0.1927 
F-statistic: 2.592 on 3 and 17 DF,  p-value: 0.08654

> residualcheck(rad_polyd$model)

	Shapiro-Wilk normality test

data:  x$residuals
W = 0.94754, p-value = 0.3058

> checkresiduals(rad_polyd$model)

	Breusch-Godfrey test for serial correlation of order up to 7

data:  Residuals
LM test = 9.0497, df = 7, p-value = 0.2491
```



From radiation dataset, Adjusted R-squared: 0.1927, p-value: 0.08654 >0.05.

4) Humidity

```
Call:
"Y ~ (Intercept) + X.t"

Residuals:
    Min      1Q  Median      3Q     Max 
-35.282 -12.993  -1.455   6.480  57.835 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)  
(Intercept) -1059.8458   673.6140  -1.573   0.1341  
z.t0            0.1301     2.6695   0.049   0.9617  
z.t1           -1.1694     1.2029  -0.972   0.3446  
z.t2            0.2287     0.1309   1.747   0.0986 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 26.19 on 17 degrees of freedom
Multiple R-squared:  0.3395,	Adjusted R-squared:  0.2229 
F-statistic: 2.912 on 3 and 17 DF,  p-value: 0.06448

> residualcheck(hum_polyd$model)

	Shapiro-Wilk normality test

data:  x$residuals
W = 0.94248, p-value = 0.2438

> checkresiduals(hum_polyd$model)

	Breusch-Godfrey test for serial correlation of order up to 7

data:  Residuals
LM test = 6.04, df = 7, p-value = 0.5351
```



From humidity data, we achieved Adjusted R-squared: 0.2229, p-value: 0.06448 >0.05.

Interpretation of model 2

The following conclusions are drawn from an analysis of the polynomial model's residuals:

- • Errors do not occur at random intervals. Notably, the ACF plot displays numerous highly significant lags as well as a distinctive wavy pattern at seasonal delays, demonstrating that residuals are still auto-correlative and seasonal. This is further supported by the Beusch-Godfrey test, which identifies serial correlation in residuals at a 5% level of significance with a p-value less than 0.05.
- Additionally, it is discovered that residuals' normality assumption is false. This is clear from both the histogram's shape and the Shapiro-Wilk normality test's findings, which show a p-value less than 0.05.
- As a result of its inability to fully account for the autocorrelation and seasonality in the series, the second-order polynomial model with a lag of 10 can be considered to have a finite amount of explanatory power.

MODEL 3

Koyck transformation

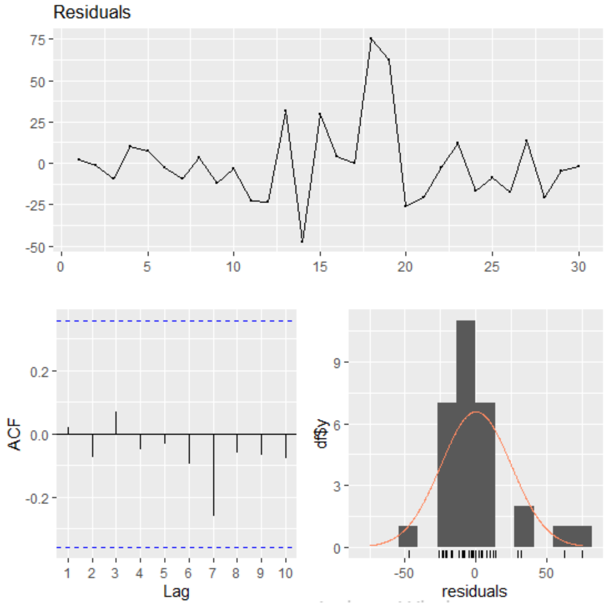
The Koyck transformation model will be applied to the precipitation predictor series in the manner described below.

For each parameter, we first create a multivariate model and then a single-variate model.

```
Call:
"Y ~ (Intercept) + Y.1 + X.t"

Residuals:
    Min      1Q  Median      3Q     Max 
-44.862 -14.803  -2.075   7.780  79.064 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -35.8807  1444.9196  -0.025    0.980
Y.1            0.2423     0.2321   1.044    0.306
X.t            2.9569    16.3858   0.180    0.858

Diagnostic tests:
                 df1 df2 statistic p-value
Weak instruments   1  27     1.839   0.186
Wu-Hausman         1  26     0.169   0.685
Sargan             0  NA        NA      NA

Residual standard error: 26.04 on 27 degrees of freedom
Multiple R-Squared: 0.00755,    Adjusted R-squared: -0.06596 
Wald test: 0.9551 on 2 and 27 DF,  p-value: 0.3974

> vif(K_trans$model)
     Y.1      X.t 
1.436777 1.436777 
```

1) Temperature

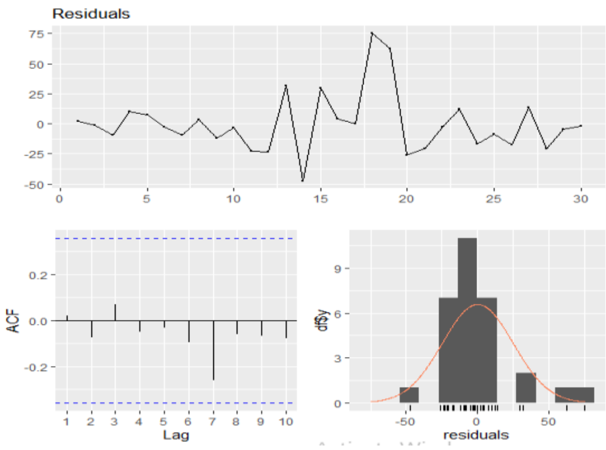
```
Call:
"Y ~ (Intercept) + Y.1 + X.t"

Residuals:
    Min      1Q  Median      3Q     Max
-47.451 -15.487  -2.648   6.757  75.055

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 526.5488   401.9192   1.310    0.201
Y.1           0.1585     0.2372   0.668    0.510
X.t         -13.7092    18.0543  -0.759    0.454

Residual standard error: 25.66 on 27 degrees of freedom
Multiple R-Squared: 0.03631,    Adjusted R-squared: -0.03508
Wald test: 1.255 on 2 and 27 DF,  p-value: 0.3011

Diagnostic tests:
NULL

                             alpha      beta       phi
Geometric coefficients:  625.7113 -13.70923 0.1584797
> vif(temp_Koyck$model, diagnostics =T)
    Y.1     X.t
1.54426 1.54426
> checkresiduals(temp_Koyck$model)

        Ljung-Box test

data:  Residuals
Q* = 0.86475, df = 6, p-value = 0.9902

Model df: 0.   Total lags used: 6
```



From temperature series, Adjusted R-squared: -0.03508,p-value: 0.311>0.05.

2) Rainfall

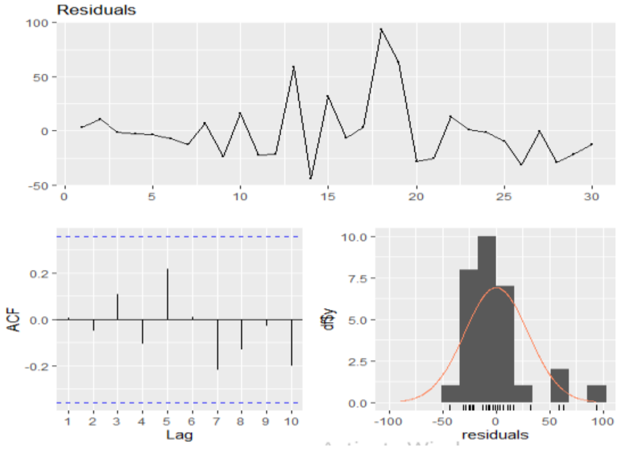
```
Residuals:
    Min      1Q  Median      3Q     Max
-43.776 -21.430  -3.028   6.055  93.212

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 177.6796   136.5337   1.301    0.204
Y.1           0.1850     0.3036   0.609    0.547
X.t          30.2749    75.4565   0.401    0.691

Residual standard error: 30.69 on 27 degrees of freedom
Multiple R-Squared: -0.3779,    Adjusted R-squared: -0.48
Wald test: 0.7567 on 2 and 27 DF,  p-value: 0.4789

Diagnostic tests:
NULL

                             alpha      beta       phi
Geometric coefficients:  218.0165 30.27487 0.1850179
> vif(rain_Koyck$model,diagnostics =T)
     Y.1      X.t
1.770534 1.770534
> residualcheck(rain_Koyck$model)

        Shapiro-Wilk normality test

data:  x$residuals
W = 0.8642, p-value = 0.001248

> checkresiduals(temp_Koyck$model)

        Ljung-Box test

data:  Residuals
Q* = 0.86475, df = 6, p-value = 0.9902

Model df: 0.   Total lags used: 6
```



From rainfall series, Adjusted R-squared: -0.48,p-value: 0.4789>0.05.

3) Radiation

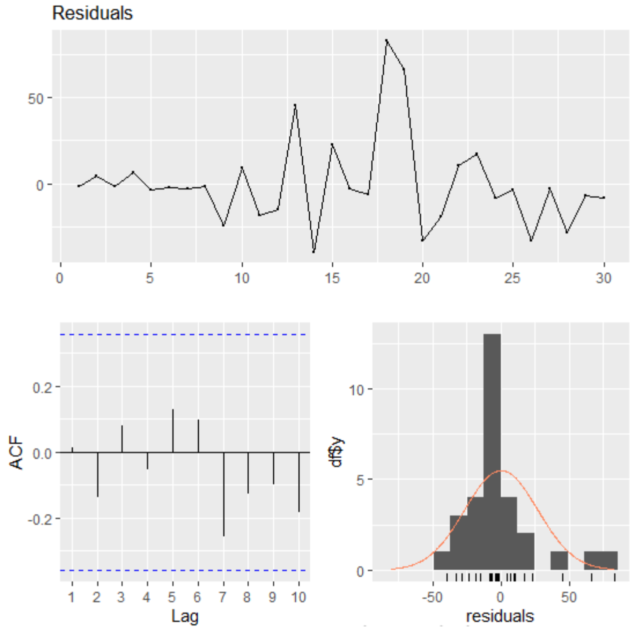
```
"Y ~ (Intercept) + Y.1 + X.t"

Residuals:
    Min      1Q  Median      3Q     Max
-48.615 -13.816  -3.619   9.597  76.084

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 633.4417   404.8050   1.565    0.129
Y.1           0.2965     0.2085   1.422    0.166
X.t         -28.6891    28.0566  -1.023    0.316

Residual standard error: 27.73 on 27 degrees of freedom
Multiple R-Squared: -0.1254,    Adjusted R-squared: -0.2088
Wald test: 1.351 on 2 and 27 DF,  p-value: 0.276

Diagnostic tests:
NULL

                              alpha      beta       phi
Geometric coefficients:  900.4256 -28.68906 0.2965086
> vif(rad_Koyck$model)
     Y.1      X.t
1.021775 1.021775
> residualcheck(rad_Koyck$model)

        Shapiro-Wilk normality test

data:  x$residuals
W = 0.90897, p-value = 0.01401

> checkresiduals(rain_Koyck$model)

        Ljung-Box test

data:  Residuals
Q* = 2.7366, df = 6, p-value = 0.8411
```



From radiation series, we achieved Adjusted R-squared: -0.2088,p-value: 0.276>0.05.

4) Humidity

```
"Y ~ (Intercept) + Y.1 + X.t"

Residuals:
    Min      1Q  Median      3Q     Max
-39.742 -13.465  -3.038   6.024  82.953

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -181.7506   639.8825  -0.284    0.779
Y.1            0.1695     0.2552   0.664    0.512
X.t            8.0820    12.6627   0.638    0.529

Residual standard error: 27.73 on 27 degrees of freedom
Multiple R-Squared: -0.1248,    Adjusted R-squared: -0.2081
Wald test: 1.032 on 2 and 27 DF,  p-value: 0.3699

Diagnostic tests:
NULL

                              alpha      beta       phi
Geometric coefficients:  -218.8323 8.082039 0.1694529
> vif(hum_Koyck$model)
     Y.1      X.t
1.531457 1.531457
> residualcheck(hum_Koyck$model)

        Shapiro-Wilk normality test

data:  x$residuals
W = 0.85813, p-value = 0.0009221

> checkresiduals(hum_Koyck$model)

        Ljung-Box test

data:  Residuals
Q* = 2.0008, df = 6, p-value = 0.9196
```



From humidity series, we derived Adjusted R-squared: -0.2081,p-value: 0.3699>0.05.

Interpretation of model 3

- It is clear from reading model summary that none of the Koyck model's terms reach statistical significance at 5% level. In addition, adjusted R-squared value is negative, suggesting that model accounts for a negative amount of variability in FFD, and model is determined to be statistically insignificant overall at 5% level (with a p-value surpassing 0.05).

- Weak Instruments test indicates that the model's first least-squares estimate lacks statistical significance at a 5% level, as shown by a p-value greater than 0.05.

- Based on the results of the Wu-Hausman test, where the p-value is bigger than 0.05, we come to the conclusion that there isn't much of a relationship between the explanatory variable and error term at the 5% level. Furthermore, since all Variance Inflation Factors (VIFs) are lower than 10, there is no sign of multicollinearity
- **Diagnostic checks** indicate wave-like patterns, with each lag in Autocorrelation Function (ACF) plot indicating relevance. This trend implies that residuals have both serial correlation and seasonality.
- Furthermore, the nature of the flaws is clearly abnormal. Both the shape of the histogram and the Shapiro-Wilk normality test, which results in a 0.05 p-value, demonstrate that the residuals do not follow a normal distribution.
- Overall, it is reasonable to conclude that the Koyck model likewise falls short of accurately capturing the seasonality and autocorrelation present in the series.

MODEL 4

Autoregressive distributed lag models

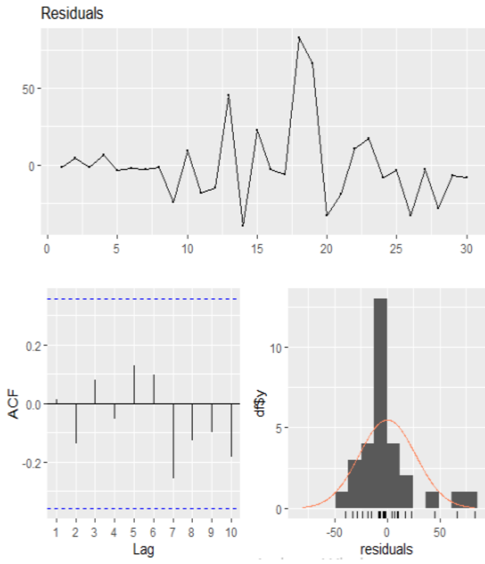
Neither polynomial nor Koyck DLMs offer satisfactory solutions, autoregressive DLMs come to our aid. The autoregressive DLM, which is essentially an infinite DLM with adaptability and efficiency, is explored. In our quest to replace Koyck model with a more suitable alternative, we proceed to fit autoregressive DLMs.

We use an iterative strategy to fit autoregressive DLMs with various lag durations and orders of autoregressive (AR) process. To calculate the ARDL (p, q) parameters, models are chosen based on criteria that minimise information. Based on the information requirements, we decide to use the following models: ARDL (1, 5), ARDL (3, 5), ARDL (3, 3), ARDL (4, 5), and ARDL (5, 5).

```
p = 1 q = 1 AIC = 284.6546 BIC = 291.6606 MASE = 0.873158
p = 1 q = 2 AIC = 277.8466 BIC = 286.0504 MASE = 0.8649917
p = 1 q = 3 AIC = 271.4002 BIC = 280.7257 MASE = 0.8667862
p = 1 q = 4 AIC = 264.796 BIC = 275.1627 MASE = 0.8606255
p = 1 q = 5 AIC = 258.4917 BIC = 269.8145 MASE = 0.8643187
p = 2 q = 1 AIC = 278.0962 BIC = 286.3 MASE = 0.8448059
p = 2 q = 2 AIC = 279.3831 BIC = 288.9542 MASE = 0.8589978
p = 2 q = 3 AIC = 272.8805 BIC = 283.5382 MASE = 0.865483
p = 2 q = 4 AIC = 266.2796 BIC = 277.9421 MASE = 0.8535675
p = 2 q = 5 AIC = 259.8228 BIC = 272.4038 MASE = 0.8669796
p = 3 q = 1 AIC = 271.2323 BIC = 280.5577 MASE = 0.850414
p = 3 q = 2 AIC = 272.2636 BIC = 282.9212 MASE = 0.8656056
p = 3 q = 3 AIC = 273.9221 BIC = 285.912 MASE = 0.8833988
p = 3 q = 4 AIC = 267.0333 BIC = 279.9917 MASE = 0.8727281
p = 3 q = 5 AIC = 260.5733 BIC = 274.4124 MASE = 0.8806191
p = 4 q = 1 AIC = 264.996 BIC = 275.3627 MASE = 0.8428691
p = 4 q = 2 AIC = 266.0656 BIC = 277.7281 MASE = 0.8599371
p = 4 q = 3 AIC = 267.6818 BIC = 280.6402 MASE = 0.8801673
p = 4 q = 4 AIC = 268.7382 BIC = 282.9925 MASE = 0.8751267
p = 4 q = 5 AIC = 262.3909 BIC = 277.4881 MASE = 0.8827498
p = 5 q = 1 AIC = 245.7741 BIC = 257.0969 MASE = 0.7270182
p = 5 q = 2 AIC = 247.7312 BIC = 260.3121 MASE = 0.728233
p = 5 q = 3 AIC = 249.2628 BIC = 263.1018 MASE = 0.7454119
p = 5 q = 4 AIC = 251.0579 BIC = 266.1551 MASE = 0.7440115
p = 5 q = 5 AIC = 250.4073 BIC = 266.7625 MASE = 0.7040422
```

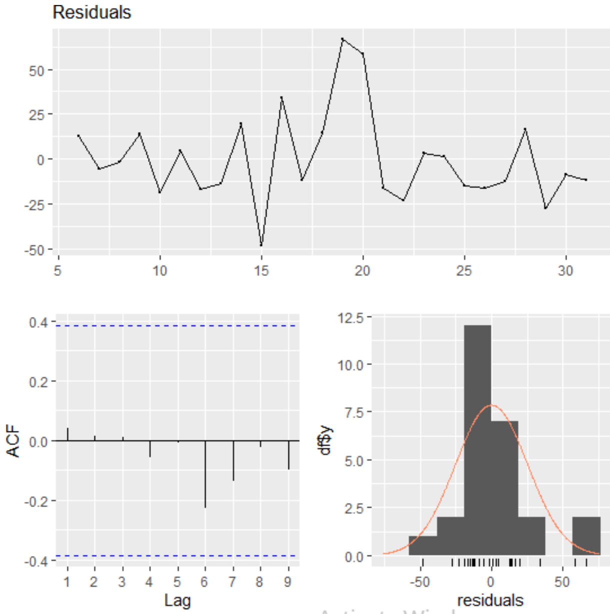
Based on previous research involving model estimations, we might strive to reduce delays for predictor series. We'll perform the ARDL (1, 5) and **diagnostic** exams.

```
Time series regression with "ts" data:
Start = 6, End = 31

Call:
dynlm(formula = as.formula(model.text), data = data, start = 1)

Residuals:
    Min      1Q  Median      3Q     Max
-50.699 -14.992  -3.891  14.851  62.850

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 518.31615  636.62738   0.814    0.426
X.t          -4.63807    5.47789  -0.847    0.408
X.1           1.90610    5.07051   0.376    0.711
Y.1           0.34908    0.24101   1.448    0.165
Y.2          -0.15769    0.24736  -0.637    0.532
Y.3           0.06685    0.25175   0.266    0.794
Y.4          -0.10498    0.24927  -0.421    0.679
Y.5          -0.04056    0.26948  -0.151    0.882

Residual standard error: 29.66 on 18 degrees of freedom
Multiple R-squared:  0.129,	Adjusted R-squared:  -0.2097
F-statistic: 0.381 on 7 and 18 DF,  p-value: 0.9015

> residualcheck(ardl_15$model)

	Shapiro-Wilk normality test

data:  x$residuals
W = 0.93875, p-value = 0.1254
```

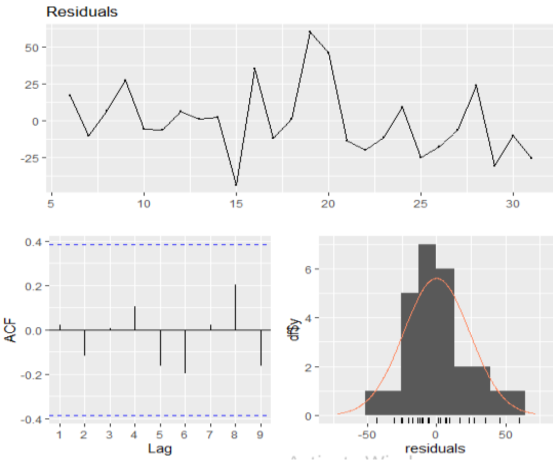


1) Temperature

```
Time series regression with "ts" data:
Start = 6, End = 31

Call:
dynlm(formula = as.formula(model.text), data = data, start = 1)

Residuals:
    Min      1Q  Median      3Q     Max
-48.073 -15.763  -6.921  13.594  66.479

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 445.72276  294.08252   1.516    0.147
X.t          -2.18918   11.20077  -0.195    0.847
X.1          -7.08787   11.39914  -0.622    0.542
Y.1           0.28599    0.24215   1.181    0.253
Y.2          -0.18723    0.24857  -0.753    0.461
Y.3           0.09967    0.24848   0.401    0.693
Y.4          -0.12819    0.24849  -0.516    0.612
Y.5           0.06691    0.24768   0.270    0.790

Residual standard error: 29.73 on 18 degrees of freedom
Multiple R-squared:  0.1247,	Adjusted R-squared:  -0.2157
F-statistic: 0.3664 on 7 and 18 DF,  p-value: 0.9101

> residualcheck(ardl_temp15$model)

	Shapiro-Wilk normality test

data:  x$residuals
W = 0.91536, p-value = 0.03505

> checkresiduals(ardl_temp15$model)

	Breusch-Godfrey test for serial correlation of order up to 11

data:  Residuals
LM test = 11.434, df = 11, p-value = 0.4077
```



```
Call:
dynlm(formula = as.formula(model.text), data = data, start = 1)

Residuals:
    Min      1Q  Median      3Q     Max
-43.535 -12.977  -5.806   8.713  60.054

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 305.55695  156.78101   1.949   0.0671 .
X.t         -24.94031   15.90947  -1.568   0.1344
X.1          12.91423   15.25396   0.847   0.4083
Y.1           0.41782    0.23817   1.754   0.0964 .
Y.2          -0.20041    0.23829  -0.841   0.4114
Y.3           0.05176    0.23995   0.216   0.8316
Y.4          -0.07210    0.23758  -0.303   0.7650
Y.5          -0.10391    0.25594  -0.406   0.6895
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 28.19 on 18 degrees of freedom
Multiple R-squared:  0.2134,	Adjusted R-squared:  -0.09251
F-statistic: 0.6976 on 7 and 18 DF,  p-value: 0.6738

> residualcheck(ardl_rain15$model)

	Shapiro-Wilk normality test

data:  x$residuals
W = 0.95919, p-value = 0.3758

> checkresiduals(ardl_rain15$model)

	Breusch-Godfrey test for serial correlation of order up to 11

data:  Residuals
LM test = 16.879, df = 11, p-value = 0.1115
```



Below is a frame of information with the AIC, BIC, and MASE values for the temperature ARDL (3, 5), ARDL (4, 5), and ARDL (5, 5).

```
> colnames(accuracy_1) <- c("Model", "MASE", "AIC", "BIC")
> head(accuracy_1)
                   Model MASE       AIC      BIC       NA
ftem_dlm      FFtemp_DLM   21 0.7066020 55.13358 68.71237
temp_polyd    temp_PolyD   21 0.7268353 55.13358 68.71237
temp_Koyck    temp_Koyck   30 0.8454533 55.13358 68.71237
ardl_temp15 ARDL_temp15    26 0.8645458 55.13358 68.71237
ardl_temp35 ARDL_temp35    26 0.8763482 55.13358 68.71237
ardl_temp45 ARDL_temp45    26 0.8596000 55.13358 68.71237
```

Below is a frame of information with the AIC, BIC, and MASE values for rainfall ARDL (3, 5), ARDL (4, 5), and ARDL (5, 5).

```
> colnames(accuracy_2) <- c("Model", "MASE", "AIC", "BIC")
> head(accuracy_2)
                   Model MASE       AIC      BIC       NA
frain_dlm     FFrain_DLM   21 0.3848341 24.72299 38.30178
rain_polyd    rain_PolyD   21 0.8045022 24.72299 38.30178
rain_Koyck    rain_Koyck   30 1.0169315 24.72299 38.30178
ardl_rain15 ARDL_rain15    26 0.8328321 24.72299 38.30178
ardl_rain35 ARDL_rain35    26 0.8298604 24.72299 38.30178
ardl_rain45 ARDL_rain45    26 0.8296785 24.72299 38.30178
```

Below is a frame of information with the AIC, BIC, and MASE values for radiation ARDL (3, 5), ARDL (4, 5), and ARDL (5, 5).

```
> colnames(accuracy_3) <- c("Model", "MASE", "AIC", "BIC")
> head(accuracy_3)
                 Model MASE       AIC      BIC       NA
frad_dlm     FFrad_DLM   21 0.3976538 21.29096 34.86975
rad_polyd    rad_PolyD   21 0.7781601 21.29096 34.86975
rad_Koyck    rad_Koyck   30 0.9751565 21.29096 34.86975
ardl_rad15 ARDL_rad15    26 0.8468247 21.29096 34.86975
ardl_rad35 ARDL_rad35    26 0.8409705 21.29096 34.86975
ardl_rad45 ARDL_rad45    26 0.8428853 21.29096 34.86975
```

Below is a frame of information with the AIC, BIC, and MASE values for humidity ARDL (3, 5), ARDL (4, 5), and ARDL (5, 5).

```
> colnames(accuracy_4) <- c("Model", "MASE", "AIC", "BIC")
> head(accuracy_4)
                    Model MASE       AIC      BIC      NA
fhum_dlm       Fhum_DLM   21 0.3536045 67.19851 80.7773
hum_polyd     hum_PolyD   21 0.7461637 67.19851 80.7773
hum_Koyck     hum_Koyck   30 0.8921521 67.19851 80.7773
ardl_hum15 ARDL_hum15   26 0.8642572 67.19851 80.7773
ardl_hum35 ARDL_hum35   26 0.8989016 67.19851 80.7773
ardl_hum45 ARDL_hum45   26 0.8959271 67.19851 80.7773
```

Interpretation of model 4

- The ARDL (1,5) model does not have statistical significance at the 5% level, according to the p-value of the test for overall significance, which is bigger than 0.05. The **diagnostic checks** display a consistent pattern, resulting in same observations as those made for the previously fitted models.
- On the whole, none of models generated through time series regression approach have proven effective in capturing autocorrelation and seasonal patterns within radiation series.
- To keep track of accuracy measures such as AIC/BIC and MASE for the models fitted thus far, we have established a data frame named "accuracy." Subsequent models will contribute their accuracy measures to this data frame.

MODEL 5

Exponential smoothing methods

As a **forecasting** method, exponential smoothing may also be tested. Since the frequency is only based on one year and there is a little seasonal component in the FFD series, exponential smoothing is not an option.

```
	Ljung-Box test

data:  Residuals from Holt-Winters' additive method
Q* = 1.5608, df = 6, p-value = 0.9554

Model df: 0.   Total lags used: 6

	Ljung-Box test

data:  Residuals from Holt-Winters' multiplicative method
Q* = 2.2408, df = 6, p-value = 0.8963

Model df: 0.   Total lags used: 6

	Ljung-Box test

data:  Residuals from Holt-Winters' multiplicative method
Q* = 2.2408, df = 6, p-value = 0.8963

Model df: 0.   Total lags used: 6
```

```
	Ljung-Box test

data:  Residuals from Holt-Winters' additive method
Q* = 1.5608, df = 6, p-value = 0.9554

Model df: 0.   Total lags used: 6

	Ljung-Box test

data:  Residuals from Holt-Winters' multiplicative method
Q* = 2.2408, df = 6, p-value = 0.8963

Model df: 0.   Total lags used: 6

	Ljung-Box test

data:  Residuals from Holt-Winters' multiplicative method
Q* = 2.2408, df = 6, p-value = 0.8963

Model df: 0.   Total lags used: 6
```

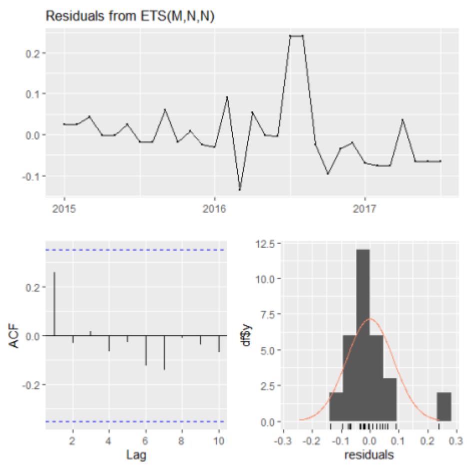
```
> accuracy_T <- rbind(accuracy_1, accuracy_2,accuracy_3,accuracy_4)
> accuracy_T
                    Model MASE       AIC      BIC       NA
ftem_dlm     FFtemp_DLM   21 0.7066020 55.13358 68.71237
temp_polyd   temp_PolyD   21 0.7268353 55.13358 68.71237
temp_Koyck   temp_Koyck   30 0.8454533 55.13358 68.71237
ardl_temp15 ARDL_temp15   26 0.8645458 55.13358 68.71237
ardl_temp35 ARDL_temp35   26 0.8763482 55.13358 68.71237
ardl_temp45 ARDL_temp45   26 0.8596000 55.13358 68.71237
ardl_temp55 ARDL_temp55   26 0.7198996 55.13358 68.71237
frain_dlm    FFrain_DLM   21 0.3848341 24.72299 38.30178
rain_polyd   rain_PolyD   21 0.8045022 24.72299 38.30178
rain_Koyck   rain_Koyck   30 1.0169315 24.72299 38.30178
ardl_rain15 ARDL_rain15   26 0.8328321 24.72299 38.30178
ardl_rain35 ARDL_rain35   26 0.8298604 24.72299 38.30178
ardl_rain45 ARDL_rain45   26 0.8296785 24.72299 38.30178
ardl_rain55 ARDL_rain55   26 0.7385166 24.72299 38.30178
frad_dlm      FFrad_DLM   21 0.3976538 21.29096 34.86975
rad_polyd     rad_PolyD   21 0.7781601 21.29096 34.86975
rad_Koyck     rad_Koyck   30 0.9751565 21.29096 34.86975
ardl_rad15   ARDL_rad15   26 0.8468247 21.29096 34.86975
ardl_rad35   ARDL_rad35   26 0.8409705 21.29096 34.86975
ardl_rad45   ARDL_rad45   26 0.8428853 21.29096 34.86975
ardl_rad55   ARDL_rad55   26 0.8283986 21.29096 34.86975
fhum_dlm      Fhum_DLM    21 0.3536045 67.19851 80.77730
hum_polyd     hum_PolyD   21 0.7461637 67.19851 80.77730
hum_Koyck     hum_Koyck   30 0.8921521 67.19851 80.77730
ardl_hum15   ARDL_hum15   26 0.8642572 67.19851 80.77730
ardl_hum35   ARDL_hum35   26 0.8989016 67.19851 80.77730
ardl_hum45   ARDL_hum45   26 0.8959271 67.19851 80.77730
ardl_hum55   ARDL_hum55   26 0.9144586 67.19851 80.77730
```

MODEL 6

State-space models variations

There are two equivalent state-space models (with additive or multiplicative mistakes) for any exponential smoothing method. In R, we are allowed to create 8 state-space variations, including seasonality (certain combinations are prohibited owing to stability difficulties). To fit these models and assess their accuracy for future comparison, we build a loop.

```
> auto_ets <- ets(ffd_ts)
> summary(auto_ets)
ETS(M,N,N)

Call:
 ets(y = ffd_ts)

  Smoothing parameters:
    alpha = 1e-04

  Initial states:
    l = 306.3826

  sigma:  0.0825

     AIC     AICc      BIC
310.6132 311.5021 314.9152

Training set error measures:
                        ME     RMSE      MAE        MPE     MAPE      MASE
Training set -0.0009438502 24.43768 16.75958 -0.5805143 5.329407 0.5686286
                  ACF1
Training set 0.2589561
```




```
> checkresiduals(auto_ets)

        Ljung-Box test

data:  Residuals from ETS(M,N,N)
Q* = 3.1718, df = 6, p-value = 0.787

Model df: 0.   Total lags used: 6
```

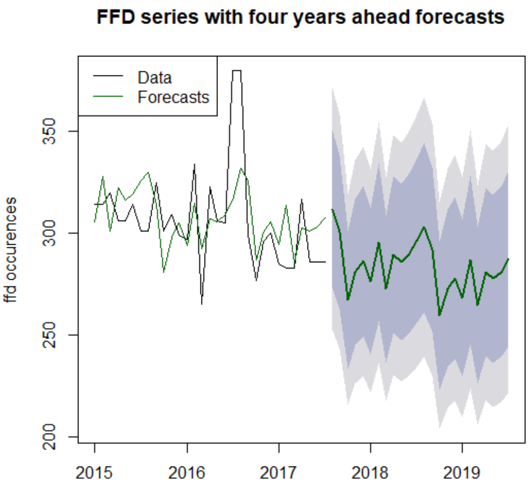
The residual analysis indicates that this model is generally unsuccessful in constructing autocorrelation and seasonality in dataset.

We add state-space model metrics to the accuracy data frame.

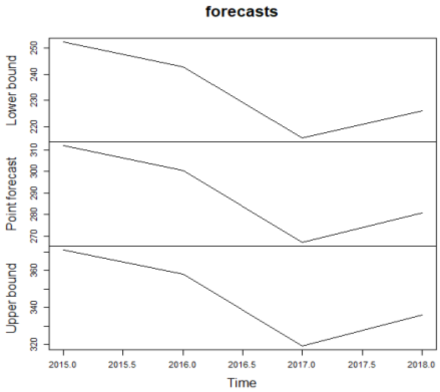
```
> accuracy_T <- rbind(accuracy_1, accuracy_2,accuracy_3,accur
> accuracy_T
                    Model MASE       AIC      BIC       NA
ftem_dlm     FFtemp_DLM   21 0.7066020 55.13358 68.71237
temp_polyd   temp_PolyD   21 0.7268353 55.13358 68.71237
temp_Koyck   temp_Koyck   30 0.8454533 55.13358 68.71237
ardl_temp15 ARDL_temp15   26 0.8645458 55.13358 68.71237
ardl_temp35 ARDL_temp35   26 0.8763482 55.13358 68.71237
ardl_temp45 ARDL_temp45   26 0.8596000 55.13358 68.71237
ardl_temp55 ARDL_temp55   26 0.7198996 55.13358 68.71237
frain_dlm    FFrain_DLM   21 0.3848341 24.72299 38.30178
rain_polyd   rain_PolyD   21 0.8045022 24.72299 38.30178
rain_Koyck   rain_Koyck   30 1.0169315 24.72299 38.30178
ardl_rain15 ARDL_rain15   26 0.8328321 24.72299 38.30178
ardl_rain35 ARDL_rain35   26 0.8298604 24.72299 38.30178
ardl_rain45 ARDL_rain45   26 0.8296785 24.72299 38.30178
ardl_rain55 ARDL_rain55   26 0.7385166 24.72299 38.30178
frad_dlm      FFrad_DLM   21 0.3976538 21.29096 34.86975
rad_polyd     rad_PolyD   21 0.7781601 21.29096 34.86975
rad_Koyck     rad_Koyck   30 0.9751565 21.29096 34.86975
ardl_rad15   ARDL_rad15   26 0.8468247 21.29096 34.86975
ardl_rad35   ARDL_rad35   26 0.8409705 21.29096 34.86975
ardl_rad45   ARDL_rad45   26 0.8428853 21.29096 34.86975
ardl_rad55   ARDL_rad55   26 0.8283986 21.29096 34.86975
fhum_dlm       Fhum_DLM   21 0.3536045 67.19851 80.77730
hum_polyd     hum_PolyD   21 0.7461637 67.19851 80.77730
hum_Koyck     hum_Koyck   30 0.8921521 67.19851 80.77730
ardl_hum15   ARDL_hum15   26 0.8642572 67.19851 80.77730
ardl_hum35   ARDL_hum35   26 0.8989016 67.19851 80.77730
ardl_hum45   ARDL_hum45   26 0.8959271 67.19851 80.77730
ardl_hum55   ARDL_hum55   26 0.9144586 67.19851 80.77730
```

The accuracy table will be used to evaluate every strategy we looked into throughout the modelling phase in terms of its Mean Absolute Scaled Error (MASE). With the damping value remaining constant across all models, the Holt-Winters multiplicative technique with an additive trend produces the lowest MASE of these. ETS (A,Ad,A), which has additive errors, additive damped trend, and additive seasonality, stands out as the best option for state-space models in terms of achieving the lowest MASE. It's interesting to observe that the model that was automatically recommended was also ETS (A, Ad, A).

The table clearly demonstrates that techniques outperform time series regression methods in MASE. However, it's essential to note that the latter approach yields lower AIC/BIC values




```
Time Series:
Start = 2015
End = 2018
Frequency = 1
     Lower bound Point forecast Upper bound
2015    252.4308       311.8282    371.2255
2016    242.7493       300.4809    358.2126
2017    215.4651       267.2555    319.0460
2018    226.0974       281.0226    335.9478
> plot(forecasts)
```



Forecasting

- To choose the model that will be utilized to forecast the FFD value for the ensuing four years, we evaluate the projections from three different models.
- The multiplicative Holt-Winters method, which captures the series' autocorrelation and seasonality the best and has the lowest MASE.
- The Holt-Winters multiplicative method has a multiplicative trend, placing second in MASE and successfully capturing autocorrelation and seasonality within the series.
- The ETS (A, Ad, A) model, which has the lowest MASE of all the state-space models, was recommended by an automatic software. It should be noted that this method does not effectively capture the autocorrelation within series.

TASK 03 (A)

Introduction

The dataset looks at 81 plant species' relative blooming orders from 1983 to 2014 for similarities. The Rank-based Order similarity metric (RBO), which compares the yearly blooming order to the flowering order from 1983, was used to determine changes in the species' flowering order. The earliest flowering species for the year under consideration is ranked 1, and the last is placed 81. RBO values are therefore integers between 0 and 1. The 81 species' first blooming occurrence orders from 1983 and each of the 31 years that followed, from 1984 to 2014, were more similar according to higher RBO values.

More recently, particularly during the Millennium Drought (1997–2009), flowering orders have diversified, showing that Australian flora is adjusting to environmental changes. Australia's drought, according to the BoM, lasted from 1996 to 2009.

Problem Statement

The main dataset "T3_RBO.csv" contains four dependent variables: temperature, precipitation, radiation, and humidity. An order value in regard to FFD is provided by the target variable, RBO. The offered secondary dataset "Covariate x-values for Task 3.csv" contains future values of the dependent variables.

TIME SERIES REGRESSION METHODS

MODEL 1

Model for finite distributed lag

With eventual objective of finding a reliable model for **forecasting** solar radiation levels, this method seeks to improve our understanding of the general variance and correlation inside time series.

We use a methodical approach to doing this. For models with varied lag durations, a loop that computes several accuracy measures was built, such as AIC/BIC and MASE. Then, in order to find the ideal lag time for our model, we choose the model with most favorable metrics, namely with the shortest values.

```
q = 1 AIC = -97.52442 BIC = -91.91963 MASE = 1.005622
q = 2 AIC = -91.57332 BIC = -84.73684 MASE = 1.147422
q = 3 AIC = -88.84678 BIC = -80.85356 MASE = 1.18155
q = 4 AIC = -82.35556 BIC = -73.28471 MASE = 1.249185
q = 5 AIC = -81.62553 BIC = -71.56076 MASE = 1.094436
q = 6 AIC = -77.12304 BIC = -66.15316 MASE = 0.9869733
q = 7 AIC = -76.88467 BIC = -65.10413 MASE = 0.9137492
q = 8 AIC = -78.3504 BIC = -65.85997 MASE = 0.7126978
q = 9 AIC = -79.75895 BIC = -66.66644 MASE = 0.6298166
q = 10 AIC = -85.18937 BIC = -71.61058 MASE = 0.523527
```

At q=1, temperature data have a lower AIC than rainfall data.

```
q = 1 AIC = -100.898 BIC = -95.29319 MASE = 0.9417954
q = 2 AIC = -96.70956 BIC = -89.87308 MASE = 0.9993747
q = 3 AIC = -97.19966 BIC = -89.20643 MASE = 0.9796852
q = 4 AIC = -90.46187 BIC = -81.39101 MASE = 1.038827
q = 5 AIC = -87.24242 BIC = -77.17765 MASE = 0.925677
q = 6 AIC = -82.31788 BIC = -71.348 MASE = 0.8543964
q = 7 AIC = -77.98405 BIC = -66.20351 MASE = 0.829337
q = 8 AIC = -76.81922 BIC = -64.32879 MASE = 0.7233794
```

Finite DLM of each variate

1) **Temperature**

```
Call:
lm(formula = model.formula, data = design)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.02910 -0.01175 -0.00495  0.01596  0.02903 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.301149   0.398691   0.755    0.469
x.t         -0.011599   0.012936  -0.897    0.393
x.1          0.024259   0.013847   1.752    0.114
x.2         -0.007121   0.012630  -0.564    0.587
x.3         -0.010736   0.012412  -0.865    0.410
x.4          0.010868   0.012314   0.883    0.400
x.5         -0.001881   0.012441  -0.151    0.883
x.6          0.022337   0.012282   1.819    0.102
x.7          0.001619   0.011577   0.140    0.892
x.8         -0.006598   0.011773  -0.560    0.589
x.9         -0.007787   0.012477  -0.624    0.548
x.10         0.007677   0.012192   0.630    0.545

Residual standard error: 0.02618 on 9 degrees of freedom
Multiple R-squared:  0.5452,	Adjusted R-squared:  -0.0107 
F-statistic: 0.9808 on 11 and 9 DF,  p-value: 0.5205

AIC and BIC values for the model:
       AIC       BIC
1 -85.18937 -71.61058
> vif(temp_dlm$model)
     x.t      x.1      x.2      x.3      x.4      x.5      x.6      x.7      x.8 
1.882817 1.925625 1.781877 1.739787 1.730817 1.757883 1.682573 1.517274 1.373584 
     x.9     x.10 
1.508769 1.414382 
```

From temperature series, we derived Adjusted R-squared: -0.0107,p-value: 0.5205 >0.05 and AIC:-85.18937.

2) Rain

```
Call:
lm(formula = model.formula, data = design)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.050274 -0.013229 -0.001445  0.015071  0.039030 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.709414   0.129781   5.466 0.000397 ***
x.t          0.016449   0.019478   0.845 0.420260    
x.1          0.009100   0.017831   0.510 0.622079    
x.2          0.014628   0.018404   0.795 0.447163    
x.3         -0.006321   0.018174  -0.348 0.735980    
x.4         -0.006181   0.020176  -0.306 0.766285    
x.5          0.004570   0.020010   0.228 0.824453    
x.6         -0.007054   0.019391  -0.364 0.724424    
x.7         -0.011836   0.021444  -0.552 0.594424    
x.8          0.004407   0.020473   0.215 0.834366    
x.9         -0.021710   0.021728  -0.999 0.343817    
x.10         0.006802   0.022752   0.299 0.771745    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.03187 on 9 degrees of freedom
Multiple R-squared:  0.3261,	Adjusted R-squared:  -0.4975 
F-statistic: 0.3959 on 11 and 9 DF,  p-value: 0.9251

AIC and BIC values for the model:
        AIC       BIC
1 -76.93255 -63.35376
> vif(rain_dlm$model)
     x.t      x.1      x.2      x.3      x.4      x.5      x.6      x.7      x.8 
1.242349 1.147054 1.282021 1.257098 1.420090 1.381663 1.279639 1.407959 1.274726 
     x.9     x.10 
1.277607 1.399164 
```

From rainfall series, Adjusted R-squared: -0.4975,p-value: 0.9251 >0.05 and AIC:-76.93255 are derived.

3) Radiation

```
Call:
lm(formula = model.formula, data = design)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.036879 -0.014268 -0.000611  0.013122  0.040675 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.428718   0.623265   0.688    0.509
x.t         -0.036410   0.037502  -0.971    0.357
x.1          0.048104   0.044770   1.074    0.311
x.2         -0.021862   0.026884  -0.813    0.437
x.3          0.002035   0.025673   0.079    0.939
x.4          0.002219   0.026890   0.083    0.936
x.5          0.002622   0.025560   0.103    0.921
x.6          0.027607   0.026512   1.041    0.325
x.7         -0.014768   0.025996  -0.568    0.584
x.8          0.008387   0.026420   0.317    0.758
x.9          0.010569   0.032643   0.324    0.754
x.10        -0.008962   0.032801  -0.273    0.791

Residual standard error: 0.03102 on 9 degrees of freedom
Multiple R-squared:  0.3616,	Adjusted R-squared:  -0.4187 
F-statistic: 0.4634 on 11 and 9 DF,  p-value: 0.8854

AIC and BIC values for the model:
       AIC    BIC
1 -78.06879 -64.49
> vif(rad_dlm$model)
     x.t      x.1      x.2      x.3      x.4      x.5      x.6      x.7      x.8 
4.571066 6.783792 3.502952 3.193621 3.262380 2.898270 2.938461 2.800495 2.625446 
     x.9     x.10 
3.207576 3.013446 
```

From rainfall series, we achieved Adjusted R-squared: -0.4187,p-value: 0.8854 >0.05 and AIC:-78.06879.

4) **Humidity**

```
Call:
lm(formula = model.formula, data = design)

Residuals:
      Min        1Q    Median        3Q       Max
-0.0305118 -0.0079610 -0.0002443  0.0159831  0.0256032

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.3474469  2.8680763  -1.167   0.2731
x.t          0.0090536  0.0094493   0.958   0.3630
x.1         -0.0067710  0.0098800  -0.685   0.5104
x.2          0.0214895  0.0097786   2.198   0.0556 .
x.3         -0.0116432  0.0094885  -1.227   0.2509
x.4         -0.0052385  0.0092682  -0.565   0.5857
x.5          0.0177998  0.0087535   2.033   0.0725 .
x.6          0.0005203  0.0081145   0.064   0.9503
x.7          0.0091601  0.0080459   1.138   0.2843
x.8          0.0074338  0.0079817   0.931   0.3760
x.9         -0.0031835  0.0081450  -0.391   0.7050
x.10         0.0043326  0.0084918   0.510   0.6222
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02378 on 9 degrees of freedom
Multiple R-squared:  0.6249,    Adjusted R-squared:  0.1663
F-statistic: 1.363 on 11 and 9 DF,  p-value: 0.3263

AIC and BIC values for the model:
        AIC       BIC
1 -89.23348 -75.65468
> vif(hum_dlm$model)
     x.t      x.1      x.2      x.3      x.4      x.5      x.6      x.7      x.8
1.925010 2.083771 1.978215 2.200686 1.986807 1.787024 1.553067 1.526096 1.503189
     x.9     x.10
1.573971 1.745843
```

From t rainfall series, we derived Adjusted R-squared: 0.1663,p-value: 0.3263 >0.05 and AIC:-89.23348.

Interpretation of Model 1

- The significance tests conducted on the model coefficients produced from the summary indicate that the bulk of the lag weights for the predictor series do not reach statistical significance at the 5% level.
- With an adjusted R-squared value of 0.1663, the Distributed Lag Model (DLM) can only account for 16.63% of the variation in radiation. It is plausible to infer that the model does not adequately fit the data given the existence of non-significant components and a low degree of explanatory power.
- Variance Inflation Factor (VIF) values below 10 show that multicollinearity is not a problem in this model, which is significant to note.

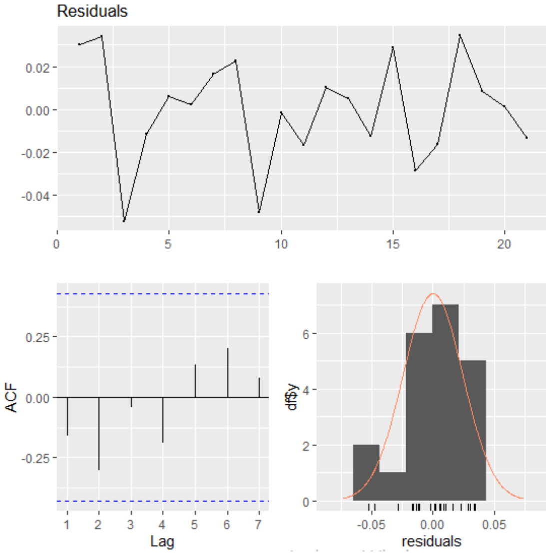
MODEL 2

Polynomial distributed lag model

Polynomial modelling on univariate

1) Temperature

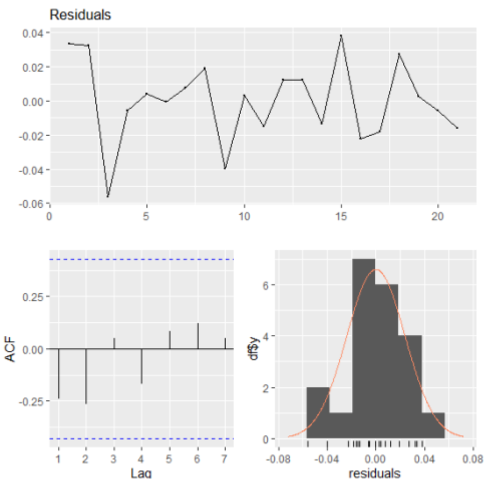
```
Call:
"Y ~ (Intercept) + X.t"

Residuals:
      Min        1Q    Median        3Q       Max
-0.052142 -0.013569  0.002131  0.016534  0.034370

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.3831305  0.3931805   0.974    0.344
z.t0        -0.0033362  0.0059193  -0.564    0.580
z.t1         0.0031363  0.0027112   1.157    0.263
z.t2        -0.0003088  0.0002518  -1.226    0.237

Residual standard error: 0.02652 on 17 degrees of freedom
Multiple R-squared:  0.1185,	Adjusted R-squared:  -0.03702
F-statistic: 0.762 on 3 and 17 DF,  p-value: 0.5308

> residualcheck(Temp_polyd3$model)

	Shapiro-Wilk normality test

data:  x$residuals
W = 0.94982, p-value = 0.3381

> checkresiduals(Temp_polyd3$model)

	Breusch-Godfrey test for serial correlation of order up to 7

data:  Residuals
LM test = 9.291, df = 7, p-value = 0.2324
```



From temperature series, Adjusted R-squared: -0.03702 ,p-value: 0.5308>0.05 derived.

2) Rainfall

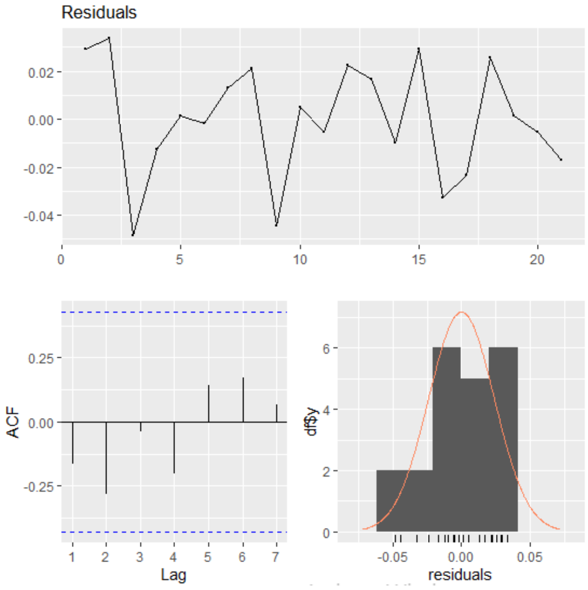
```
Call:
"Y ~ (Intercept) + X.t"

Residuals:
      Min        1Q    Median        3Q       Max
-0.056127 -0.014819  0.002352  0.012277  0.038053

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.7166742  0.1029308   6.963 2.29e-06 ***
z.t0         0.0152725  0.0110859   1.378    0.186
z.t1        -0.0060829  0.0055819  -1.090    0.291
z.t2         0.0004315  0.0005823   0.741    0.469
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02592 on 17 degrees of freedom
Multiple R-squared:  0.1584,	Adjusted R-squared:  0.009865
F-statistic: 1.066 on 3 and 17 DF,  p-value: 0.3894

> residualcheck(Rain_polyd3$model)

	Shapiro-Wilk normality test

data:  x$residuals
W = 0.97008, p-value = 0.7348

> checkresiduals(Rain_polyd3$model)

	Breusch-Godfrey test for serial correlation of order up to 7

data:  Residuals
LM test = 9.9973, df = 7, p-value = 0.1887
```



From rainfall series, Adjusted R-squared: 0.009865, p-value: 0.3894>0.05 derived.

3) Radiation

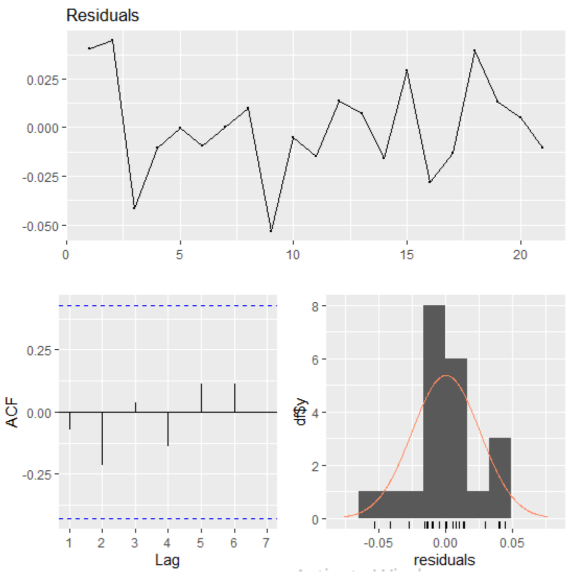
```
Call:
"Y ~ (Intercept) + X.t"

Residuals:
      Min        1Q    Median        3Q       Max
-0.048403 -0.012315  0.001545  0.021358  0.033699

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.2975083  0.4879633   0.610    0.550
z.t0        -0.0057426  0.0083076  -0.691    0.499
z.t1         0.0038059  0.0039999   0.952    0.355
z.t2        -0.0003054  0.0004050  -0.754    0.461

Residual standard error: 0.02599 on 17 degrees of freedom
Multiple R-squared:  0.1538,	Adjusted R-squared:  0.004456
F-statistic:  1.03 on 3 and 17 DF,  p-value: 0.4043

> residualcheck(Rad_polyd3$model)

	Shapiro-Wilk normality test

data:  x$residuals
W = 0.951, p-value = 0.3557

> checkresiduals(Rad_polyd3$model)

	Breusch-Godfrey test for serial correlation of order up to 7

data:  Residuals
LM test = 7.121, df = 7, p-value = 0.4164
```



From radiation series, Adjusted R-squared: 0.004456, p-value: 0.4043>0.05 derived.

4) Humidity

```
> summary(Humi_polyd3)

Call:
"Y ~ (Intercept) + X.t"

Residuals:
      Min        1Q    Median        3Q       Max
-0.053351 -0.013376 -0.000361  0.013300  0.044624

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.6065641  3.1981382  -0.502    0.622
z.t0         0.0022052  0.0058503   0.377    0.711
z.t1         0.0004784  0.0025645   0.187    0.854
z.t2        -0.0000676  0.0002504  -0.270    0.790

Residual standard error: 0.0276 on 17 degrees of freedom
Multiple R-squared:  0.04517,	Adjusted R-squared:  -0.1233
F-statistic: 0.2681 on 3 and 17 DF,  p-value: 0.8475

> residualcheck(Humi_polyd3$model)

	Shapiro-Wilk normality test

data:  x$residuals
W = 0.96765, p-value = 0.6806

> checkresiduals(Humi_polyd3$model)

	Breusch-Godfrey test for serial correlation of order up to 7

data:  Residuals
LM test = 5.2541, df = 7, p-value = 0.629
```



From the humidity series, Adjusted R-squared: -0.1233, p-value: 0.8475>0.05 derived.

Interpretation of model 2

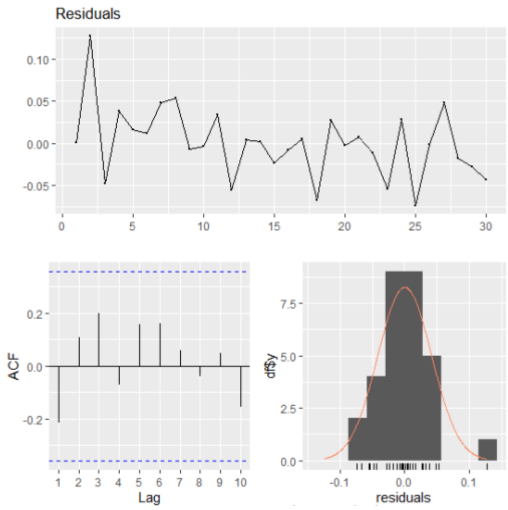
- Since the p-value is greater than 0.05, the Beusch-Godfrey test finds a serial correlation in the residuals at a 5% level of significance.
- The Shapiro-Wilk normality test results, which reveal a p-value less than 0.05, and the shape of the histogram both demonstrate that the normality assumption for the residuals is flawed.
- In conclusion, the second-order polynomial model with a lag of 10 fails to adequately account for the autocorrelation and seasonality inherent in the series, which has a low degree of explanatory power.

MODEL 3

Koyck transformation

1) Temperature

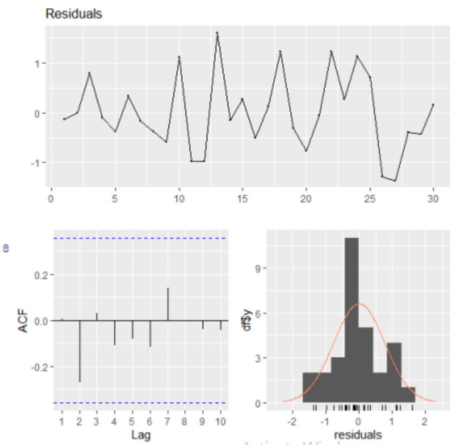
```
Call:
"Y ~ (Intercept) + Y.1 + X.t"

Residuals:
      Min         1Q     Median         3Q        Max
-0.0741656 -0.0225173 -0.0006794  0.0240622  0.1270971

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.20775    0.83741  -0.248   0.8059
Y.1          0.68547    0.25559   2.682   0.0123 *
X.t          0.02235    0.03523   0.634   0.5312
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.04319 on 27 degrees of freedom
Multiple R-Squared: 0.1517,     Adjusted R-squared: 0.08891
Wald test: 5.309 on 2 and 27 DF,  p-value: 0.01136

Diagnostic tests:
                 df1 df2 statistic    p-value
Weak instruments   1  27  4.634891 0.04042345
Wu-Hausman         1  26  1.347013 0.25634707

                                alpha       beta       phi
Geometric coefficients:  -0.6605142 0.02234675 0.6854665
> vif(Temp_Koyck3$model, diagnostics =T)
     Y.1      X.t
2.059958 2.059958
```



Results obtained were R-squared:0.0889, p-value: 0.01136 <0.05.

2) Rainfall

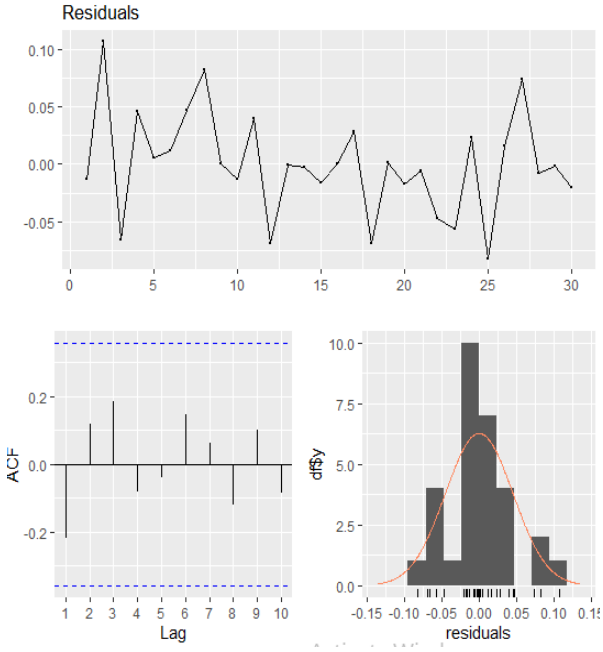
```
Call:
"Y ~ (Intercept) + Y.1 + X.t"

Residuals:
    Min      1Q  Median      3Q     Max
-1.3665 -0.4155 -0.1142  0.3241  1.6012

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.3207     2.4302   0.132    0.896
Y.1          -6.5147   243.8216  -0.027    0.979
X.t           2.2101    76.0635   0.029    0.977

Residual standard error: 0.7951 on 27 degrees of freedom
Multiple R-Squared: -286.5,     Adjusted R-squared: -307.8
Wald test: 0.01549 on 2 and 27 DF,  p-value: 0.9846

Diagnostic tests:
                 df1 df2    statistic   p-value
Weak instruments   1  27 0.0008275768 0.9772615
Wu-Hausman         1  26 0.3602689549 0.5535531

                              alpha    beta       phi
Geometric coefficients:  0.04267914 2.21011 -6.514689
> vif(Rain_Koyck3$model,diagnostics =T)
     Y.1      X.t
5531.807 5531.807
```



From rainfall series, Adjusted R-squared: -307.8, p-value: 0.9846 >0.05 derived.

3) Radiation

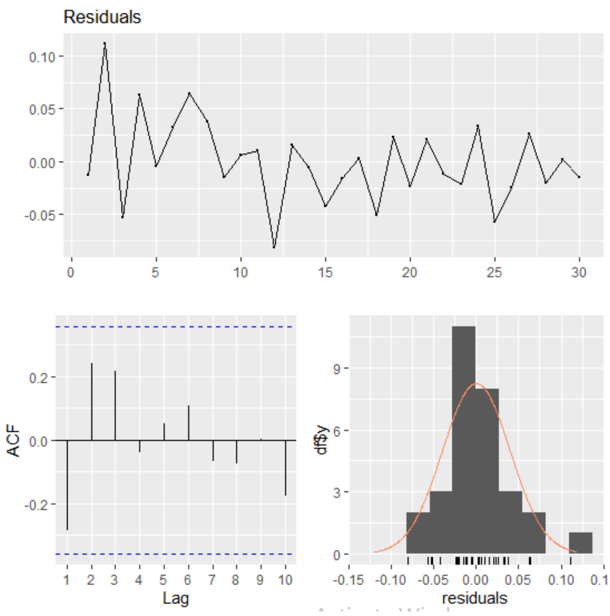
```
Call:
"Y ~ (Intercept) + Y.1 + X.t"

Residuals:
      Min        1Q    Median        3Q       Max
-0.082255 -0.017008 -0.001036  0.021424  0.106984

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.48011    0.94819  -0.506   0.6167
Y.1          0.69801    0.24502   2.849   0.0083 **
X.t          0.04812    0.05661   0.850   0.4028
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0467 on 27 degrees of freedom
Multiple R-Squared: 0.008467,   Adjusted R-squared: -0.06498
Wald test: 4.731 on 2 and 27 DF,  p-value: 0.01732

Diagnostic tests:
                 df1 df2 statistic    p-value
Weak instruments   1  27  4.941539 0.03478221
Wu-Hausman         1  26  2.764873 0.10836470

                              alpha       beta       phi
Geometric coefficients:  -1.589802 0.04811971 0.6980071
> vif(Rad_Koyck3$model,diagnostics =T)
     Y.1      X.t
1.619594 1.619594
```



From radiation series Adjusted R-squared: -0.06498, p-value: 0.1732 <0.05 derived.

4) Humidity

```
Call:
"Y ~ (Intercept) + Y.1 + X.t"

Residuals:
      Min        1Q    Median        3Q       Max
-0.080897 -0.021103 -0.004676  0.022673  0.111041

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.16679    8.04941  -0.145   0.8858
Y.1          0.62503    0.34753   1.798   0.0833 .
X.t          0.01525    0.08274   0.184   0.8551
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.04127 on 27 degrees of freedom
Multiple R-Squared: 0.2256,     Adjusted R-squared: 0.1682
Wald test: 5.612 on 2 and 27 DF,  p-value: 0.009161

Diagnostic tests:
                 df1 df2  statistic   p-value
Weak instruments   1  27 0.39261559 0.5361901
Wu-Hausman         1  26 0.05497691 0.8164556

                              alpha       beta       phi
Geometric coefficients:  -3.111733 0.01525207 0.6250339
> vif(Humi_Koyck3$model,diagnostics =T)
     Y.1      X.t
4.171591 4.171591
```



From Humidity series, Adjusted R-squared: 0.1682, p-value: 0.009161 <0.05 derived.

Interpretation of model 3

- We derived Adjusted R-squared: 0.0889, p-value: 0.01136 0.05 from the temperature series. Results are superior to those from previous Koyck models.
- We may conclude from the model summary that, aside from rainfall, all Koyck model components are significant at the 5% level. The model is shown to be statistically significant overall at a 5% level (p-value 0.05), and its adjusted R2 is negative, indicating that it explains roughly negative variability in RBO.

- The Weak Instruments Test indicates that the model in the initial stage of least-squares estimation is not significant at the 5% level (p-value > 0.05).
- The explanatory variable and error term do not substantially correlate at the 5% level, according to the Wu-Hausman test (p-value > 0.05). Due to the fact that all VIFs are smaller than 10, multicolinearity has no effect.
- • The ACF plot's wave-like form and the statistical importance of each lag show that serial correlation and seasonality still exist in the residuals.

MODEL 4

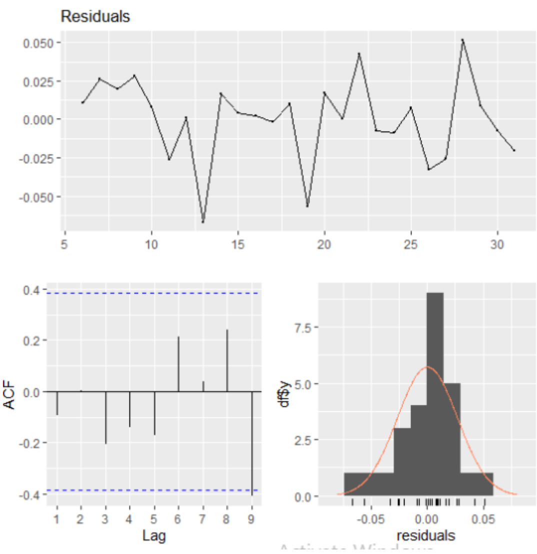
Autoregressive distributed lag models

Neither polynomial nor Koyck DLMs offer satisfactory solutions, autoregressive DLMs come to our aid. The autoregressive DLM, which is essentially an infinite DLM with adaptability and efficiency, is explored. In our quest to replace Koyck model with a more suitable alternative, we proceed to fit autoregressive DLMs.

We employ an iterative technique. To calculate the ARDL (p, q) parameters, models are chosen based on criteria that minimise information. Based on the information requirements, we decide to use the following models: ARDL (1, 5), ARDL (3, 5), ARDL (3, 3), ARDL (4, 5), and ARDL (5, 5).

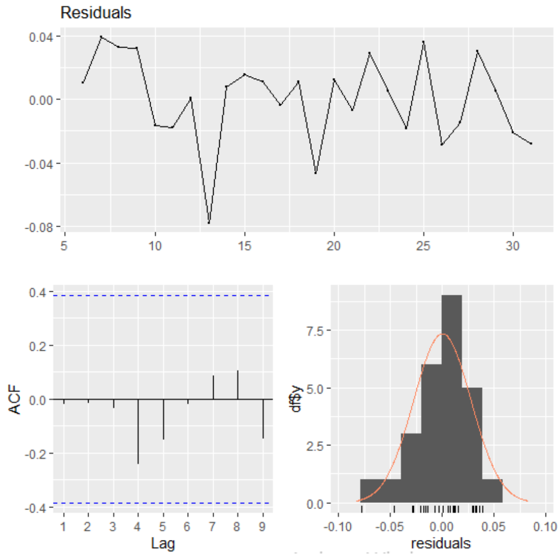
1) Temperature

```
Call:
dynlm(formula = as.formula(model.text), data = data, start = 1)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.067028 -0.008625  0.003178  0.015027  0.051371 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept) -0.07455    0.51710  -0.144   0.8870  
X.t         -0.01239    0.01497  -0.827   0.4189  
X.1          0.02394    0.01600   1.496   0.1520  
Y.1          0.47283    0.22081   2.141   0.0462 *
Y.2          0.27948    0.24197   1.155   0.2632  
Y.3         -0.00975    0.23679  -0.041   0.9676  
Y.4         -0.06098    0.23349  -0.261   0.7969  
Y.5          0.10531    0.20951   0.503   0.6213  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.03165 on 18 degrees of freedom
Multiple R-squared:  0.5828,	Adjusted R-squared:  0.4206 
F-statistic: 3.592 on 7 and 18 DF,  p-value: 0.01348

> residualcheck(ardl3_Temp15$model)

	Shapiro-Wilk normality test

data:  x$residuals
W = 0.95746, p-value = 0.344

> checkresiduals(ardl3_Temp15$model)

	Breusch-Godfrey test for serial correlation of order up to 11

data:  Residuals
LM test = 19.815, df = 11, p-value = 0.04795
```



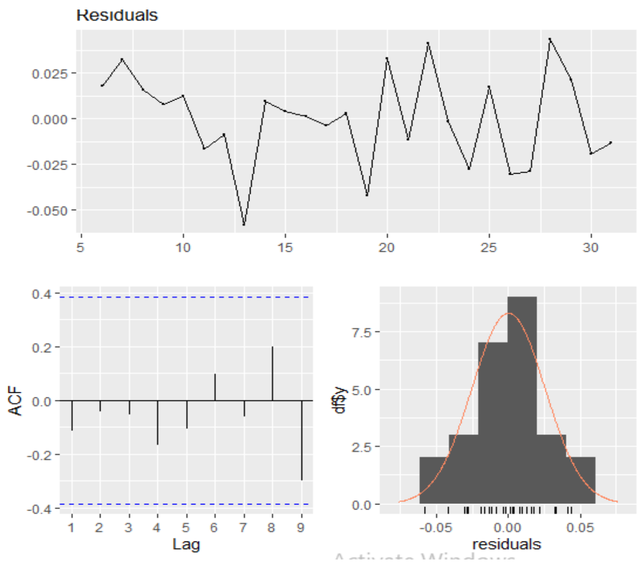
2) Rainfall

```
Call:
dynlm(formula = as.formula(model.text), data = data, start = 1)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.078127 -0.017340  0.005343  0.014866  0.039246 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.228993   0.134816   1.699    0.107
X.t          0.019647   0.018162   1.082    0.294
X.1          0.007268   0.018640   0.390    0.701
Y.1          0.372964   0.248591   1.500    0.151
Y.2          0.260567   0.244988   1.064    0.302
Y.3          0.174384   0.214252   0.814    0.426
Y.4         -0.203452   0.196005  -1.038    0.313
Y.5         -0.008021   0.196922  -0.041    0.968

Residual standard error: 0.0326 on 18 degrees of freedom
Multiple R-squared:  0.5575,	Adjusted R-squared:  0.3854 
F-statistic: 3.239 on 7 and 18 DF,  p-value: 0.02091

> residualcheck(ardl3_Rain15$model)

	Shapiro-Wilk normality test

data:  x$residuals
W = 0.94165, p-value = 0.147

> checkresiduals(ardl3_Rain15$model)

	Breusch-Godfrey test for serial correlation of order up to 11

data:  Residuals
LM test = 15.27, df = 11, p-value = 0.1705
```



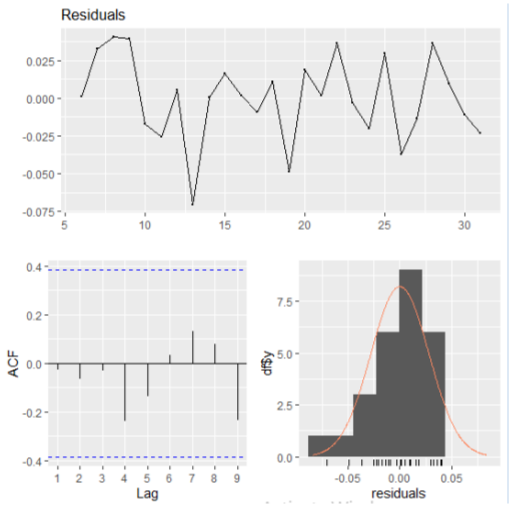
3) Radiation

```
dynlm(formula = as.formula(model.text), data = data, start = 1)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.058184 -0.015578  0.002216  0.017187  0.043461 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept)  0.12704    0.37571   0.338   0.7392  
X.t         -0.02852    0.01654  -1.724   0.1017  
X.1          0.03195    0.01691   1.889   0.0751 .
Y.1          0.54526    0.21425   2.545   0.0203 *
Y.2          0.24093    0.21721   1.109   0.2819  
Y.3          0.06064    0.19795   0.306   0.7629  
Y.4         -0.21453    0.18353  -1.169   0.2577  
Y.5          0.12092    0.18766   0.644   0.5275  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02995 on 18 degrees of freedom
Multiple R-squared:  0.6264,	Adjusted R-squared:  0.4811 
F-statistic: 4.311 on 7 and 18 DF,  p-value: 0.005806

> residualcheck(ardl3_Rad15$model)

	Shapiro-Wilk normality test

data:  x$residuals
W = 0.98447, p-value = 0.9519

> checkresiduals(ardl3_Rad15$model)

	Breusch-Godfrey test for serial correlation of order up to 11

data:  Residuals
LM test = 16.97, df = 11, p-value = 0.1088
```



4) Humidity

```
dynlm(formula = as.formula(model.text), data = data, start = 1)

Residuals:
     Min       1Q   Median       3Q      Max
-0.07050 -0.01625  0.00118  0.01782  0.04034

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.831753   1.210326  -0.687   0.5007
X.t          0.006641   0.009130   0.727   0.4764
X.1          0.003776   0.009215   0.410   0.6868
Y.1          0.446855   0.235475   1.898   0.0739 .
Y.2          0.312764   0.264204   1.184   0.2519
Y.3          0.211856   0.234960   0.902   0.3791
Y.4         -0.188857   0.198286  -0.952   0.3535
Y.5          0.002712   0.198356   0.014   0.9892
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.03306 on 18 degrees of freedom
Multiple R-squared:  0.5449,	Adjusted R-squared:  0.3679
F-statistic: 3.079 on 7 and 18 DF,  p-value: 0.02568

> residualcheck(ardl3_Hum15$model)

	Shapiro-Wilk normality test

data:  x$residuals
W = 0.95958, p-value = 0.3833

> checkresiduals(ardl3_Hum15$model)

	Breusch-Godfrey test for serial correlation of order up to 11

data:  Residuals
LM test = 14.675, df = 11, p-value = 0.1978
```



Interpretation of model 4

- The p-value from the test for overall significance, which is greater than 0.05, indicates that the ARDL(1,5) model does not have statistical significance at a 5% level. The model's modified R-squared value is 0.4811, meaning it accounts for 48.11% of the variation in radiation.

- The **diagnostic plots** show patterns that correspond to those seen during earlier diagnostic assessments. The same findings and remarks so hold true as for prior fitted models.
- In summary, it can be said that no time series regression-based model has been able to accurately capture the seasonal and autocorrelation patterns present in radiation series. To record the accuracy metrics, such as AIC/BIC and MASE from the models fitted thus far, we establish a data frame accuracy. This data frame will be supplemented with the accuracy metrics for further models.

Below is a frame of information with AIC, BIC, and MASE values for temperature ARDL (3, 5), ARDL (4, 5), and ARDL (5, 5).

```
> colnames(accuracy_a) <- c("Model", "MASE", "AIC", "BIC")
> head(accuracy_a)
                     Model MASE       AIC       BIC        NA
temp_dlm          Temp_DLM3   21 0.5235270 -85.18937 -71.61058
Temp_polyd3      Temp_PolyD3   21 0.6714874 -85.18937 -71.61058
Temp_Koyck3      Temp_Koyck3   30 0.9535116 -85.18937 -71.61058
ardl3_Temp15 ARDL3_temp15   26 0.7402468 -85.18937 -71.61058
ardl3_Temp35 ARDL3_temp35   26 0.7628024 -85.18937 -71.61058
ardl3_Temp45 ARDL3_temp45   26 0.7788240 -85.18937 -71.61058
```

Below is a frame of information with AIC, BIC, and MASE values for rainfall ARDL (3, 5), ARDL (4, 5), and ARDL (5, 5).

```
> colnames(accuracy_b) <- c("Model", "MASE", "AIC", "BIC")
> head(accuracy_b)
                    Model MASE        AIC       BIC        NA
rain_dlm         Rain_DLM   21  0.5856170 -76.93255 -63.35376
Rain_polyd3   Rain_PolyD3   21  0.6451804 -76.93255 -63.35376
Rain_Koyck3   Rain_Koyck3   30 19.1057647 -76.93255 -63.35376
ardl3_Rain15 ARDL3_Rain15   26  0.8152025 -76.93255 -63.35376
ardl3_Rain35 ARDL3_Rain35   26  0.8095313 -76.93255 -63.35376
ardl3_Rain45 ARDL3_Rain45   26  0.7390848 -76.93255 -63.35376
```

Below is a frame of information with AIC, BIC, and MASE values for radiation ARDL (3, 5), ARDL (4, 5), and ARDL (5, 5).

```
> colnames(accuracy_c) <- c("Model", "MASE", "AIC", "BIC")
> head(accuracy_c)
                  Model MASE       AIC       BIC     NA
rad_dlm         Rad_DLM   21 0.6037290 -78.06879 -64.49
Rad_polyd3  Rain_PolyD3   21 0.6711964 -78.06879 -64.49
Rad_Koyck3  Rain_Koyck3   30 1.0314227 -78.06879 -64.49
ardl3_Rad15 ARDL3_Rain15  26 0.7627672 -78.06879 -64.49
ardl3_Rad35 ARDL3_Rain35  26 0.7009101 -78.06879 -64.49
ardl3_Rad45 ARDL3_Rain45  26 0.7052516 -78.06879 -64.49
```

Below is a frame of information with AIC, BIC, and MASE values for humidity ARDL (3, 5), ARDL (4, 5), and ARDL (5, 5).

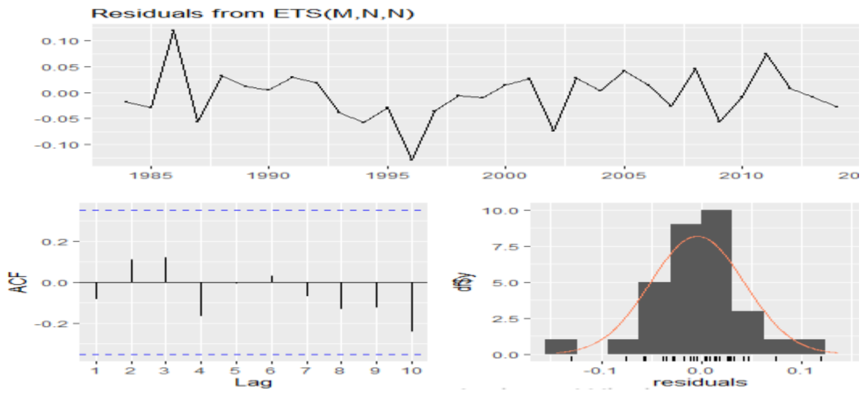
```
> colnames(accuracy_d) <- c("Model", "MASE", "AIC", "BIC")
> head(accuracy_d)
                    Model MASE       AIC       BIC        NA
hum_dlm           Hum_DLM   21 0.4518490 -89.23348 -75.65468
Humi_polyd3   Humi_PolyD3   21 0.6821919 -89.23348 -75.65468
Humi_Koyck3   Humi_Koyck3   30 0.9559618 -89.23348 -75.65468
ardl3_Hum15   ARDL3_Hum15   26 0.8156142 -89.23348 -75.65468
ardl3_Humi35 ARDL3_Humi35   26 0.8095313 -89.23348 -75.65468
ardl3_Humi45 ARDL3_Humi45   26 0.7390848 -89.23348 -75.65468
```

MODEL 05

Exponential Smoothing

Before choosing the final model to generate three-year projections of solar radiation, we evaluate the predictions from the other two models:

```
> f1.etsM = ets(RBO_ts, model="MNN")
> summary(f1.etsM)
ETS(M,N,N)

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

  Smoothing parameters:
    alpha = 0.4421

  Initial states:
    l = 0.7685

  sigma:  0.0479

      AIC      AICc       BIC
-96.69180 -95.80291 -92.38984

Training set error measures:
                       ME       RMSE        MAE        MPE     MAPE      MASE
Training set -0.003529451 0.03466016 0.02607253 -0.6459794 3.549133 0.8460683
                   ACF1
Training set -0.0753464
> checkresiduals(f1.etsM)

	Ljung-Box test

data:  Residuals from ETS(M,N,N)
Q* = 2.2852, df = 6, p-value = 0.8917

Model df: 0.   Total lags used: 6
```



- We analyze the results of alternative models prior to selecting the final one to use for **forecasting** the FFD value over the following four years:
- The multiplicative Holt-Winters approach, which has the lowest Mean Absolute Scaled Error (MASE) and is the best at capturing autocorrelation as well as seasonality.
- There is a multiplicative trend in the Holt-Winters multiplicative approach, which is ranked second in MASE and excels at capturing autocorrelation and seasonality within series.

- The ETS(M,N,N) model, with the lowest MASE among all state-space models, was suggested by an automated system. It's crucial to remember that this model fails to accurately depict autocorrelation present in series.

1) simple exponential forecast

```
Forecast method: Simple exponential smoothing

Model Information:
Simple exponential smoothing

Call:
 ses(y = RBO_ts, h = 3, initial = "simple", alpha = 0.1)

  Smoothing parameters:
    alpha = 0.1

  Initial states:
    l = 0.755

  sigma:  0.0406
Error measures:
                       ME       RMSE        MAE       MPE     MAPE     MASE
Training set -0.009995034 0.0406462 0.03161539 -1.640631 4.337625 1.025937
                  ACF1
Training set 0.4122565

Forecasts:
     Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
2015      0.7240242 0.6719340 0.7761144 0.6443591 0.8036893
2016      0.7240242 0.6716742 0.7763742 0.6439618 0.8040866
2017      0.7240242 0.6714157 0.7766327 0.6435664 0.8044820
> checkresiduals(f1)

        Ljung-Box test

data:  Residuals from Simple exponential smoothing
Q* = 17.699, df = 6, p-value = 0.007029

Model df: 0.   Total lags used: 6
```

2) Holts simple forecast

```
Forecast method: Holt's method

Model Information:
Holt's method

Call:
 holt(y = RBO_ts, h = 3, initial = "simple")

  Smoothing parameters:
    alpha = 0.5678
    beta  = 0.0888

  Initial states:
    l = 0.755
    b = -0.0143

  sigma:  0.0376
Error measures:
                      ME       RMSE        MAE       MPE     MAPE      MASE
Training set 0.008262872 0.03758621 0.02896041 0.9541908 3.905009 0.9397818
                   ACF1
Training set -0.1449597

Forecasts:
     Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
2015      0.7162102 0.6680415 0.7643789 0.6425426 0.7898778
2016      0.7148677 0.6572445 0.7724909 0.6267407 0.8029948
2017      0.7135252 0.6456320 0.7814184 0.6096915 0.8173589
> checkresiduals(f2)

        Ljung-Box test

data:  Residuals from Holt's method
Q* = 2.6105, df = 6, p-value = 0.8559

Model df: 0.   Total lags used: 6
```

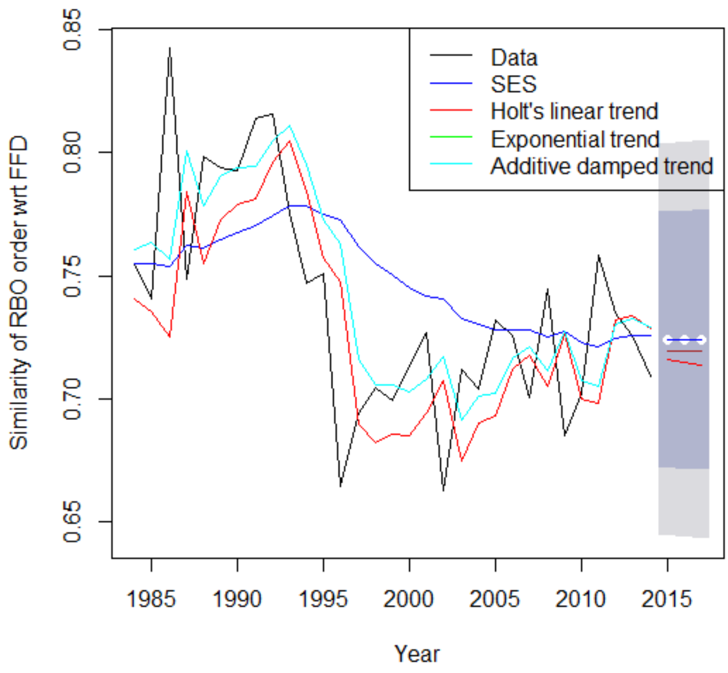
3) Holts with exponential trend

```
Forecast method: Holt's method with exponential trend

Model Information:
Holt's method with exponential trend

Call:
 holt(y = RBO_ts, h = 3, initial = "simple", exponential = TRUE)

  Smoothing parameters:
    alpha = 0.5667
    beta  = 0.0845

  Initial states:
    l = 0.755
    b = 0.9811

  sigma:  0.0514
Error measures:
                      ME       RMSE        MAE       MPE     MAPE      MASE
Training set 0.008053192 0.03737566 0.02865753 0.9221333 3.861994 0.9299532
                   ACF1
Training set -0.1461854

Forecasts:
     Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
2015      0.7164020 0.6694961 0.7646153 0.6444499 0.7912975
2016      0.7151840 0.6591996 0.7730208 0.6334674 0.8037453
2017      0.7139681 0.6475410 0.7813773 0.6167553 0.8198834
> checkresiduals(f33)

        Ljung-Box test

data:  Residuals from Holt's method with exponential trend
Q* = 2.6109, df = 6, p-value = 0.8559

Model df: 0.   Total lags used: 6
```

4) Additive damped holts method

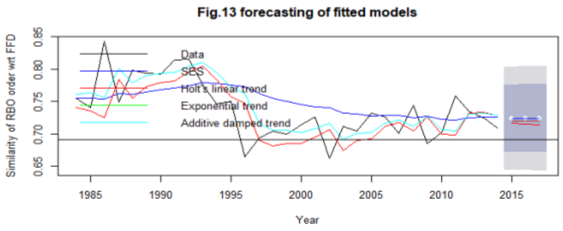
```
Call:
 holt(y = RBO_ts, h = 3, damped = TRUE, initial = "simple")

  Smoothing parameters:
    alpha = 0.4773
    beta  = 1e-04
    phi   = 0.8

  Initial states:
    l = 0.7542
    b = 0.0082

  sigma:  0.0377

      AIC      AICc       BIC
-90.15879 -86.65879 -81.55487

Error measures:
                       ME       RMSE        MAE        MPE     MAPE      MASE
Training set -0.004553134 0.03457183 0.02527824 -0.7722916 3.450154 0.8202932
                   ACF1
Training set -0.1237046

Forecasts:
     Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
2015      0.7195752 0.6711967 0.7679537 0.6455866 0.7935638
2016      0.7195801 0.6659717 0.7731885 0.6375931 0.8015670
2017      0.7195840 0.6612112 0.7779567 0.6303106 0.8088574
> checkresiduals(f4)

        Ljung-Box test

data:  Residuals from Damped Holt's method
Q* = 2.7586, df = 6, p-value = 0.8385

Model df: 0.   Total lags used: 6
```

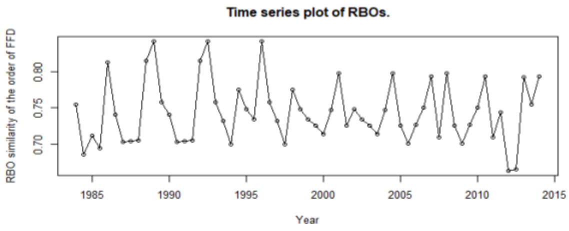
Fig.13 forecasting of fitted models

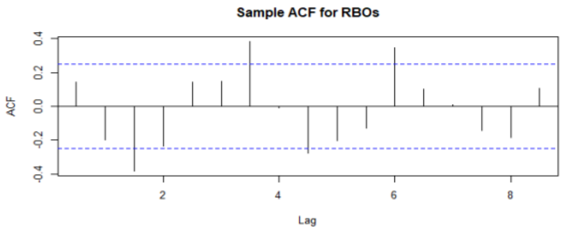


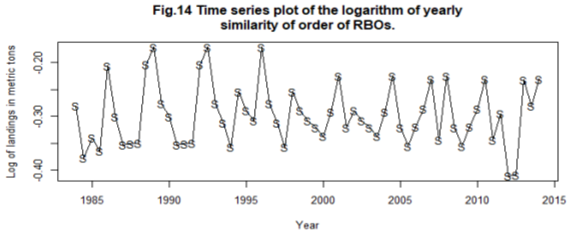
However, we can observe that the 95% confidence intervals for the forecasts from selected approach are very precise and provide reliable forecasts.

TASK 3(B)







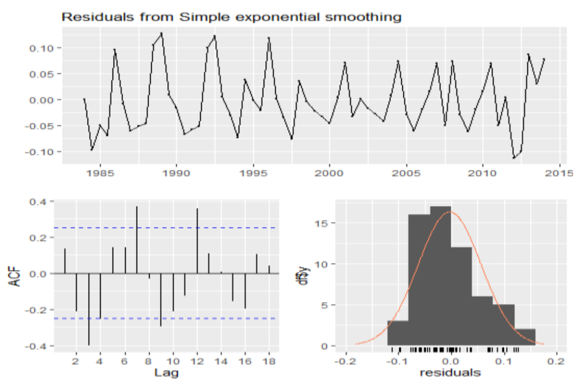


Dynlm Modelling univariately

```
Time series regression with "ts" data:
Start = 1984(2), End = 2014(1)

Call:
dynlm(formula = Y.t ~ L(Y.t, k = 1) + S.t + trend(Y.t) + season(Y.t))

Residuals:
      Min        1Q    Median        3Q       Max
-0.122950 -0.043817 -0.009939  0.030096  0.119378

Coefficients: (1 not defined because of singularities)
                Estimate Std. Error t value Pr(>|t|)
(Intercept)   -2.343e-01  4.350e-02  -5.387 1.47e-06 ***
L(Y.t, k = 1)  1.730e-01  1.331e-01   1.300    0.199
S.t                   NA         NA      NA       NA
trend(Y.t)    -9.609e-05  8.560e-04  -0.112    0.911
season(Y.t)2  -2.037e-02  1.494e-02  -1.363    0.178
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.05729 on 56 degrees of freedom
Multiple R-squared:  0.05307,	Adjusted R-squared:  0.002347
F-statistic: 1.046 on 3 and 56 DF,  p-value: 0.3793
```

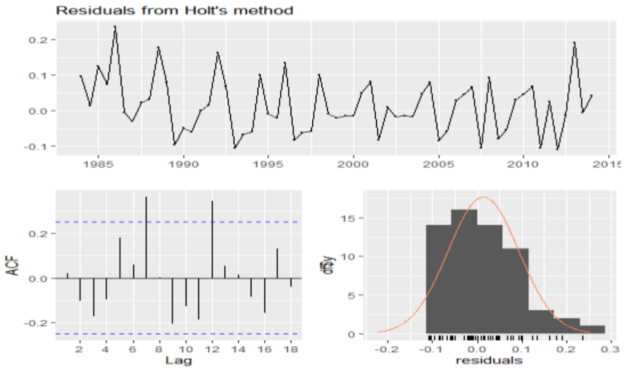
1) Simple exponential forecasting

```
Forecast method: Simple exponential smoothing

Model Information:
Simple exponential smoothing

Call:
 ses(y = RBO.tr, h = 3, initial = "simple", alpha = 0.1)

  Smoothing parameters:
    alpha = 0.1

  Initial states:
    l = -0.281

  sigma:  0.0591
Error measures:
                       ME       RMSE        MAE       MPE     MAPE      MASE
Training set -0.003149581 0.05912707 0.04786464 -3.448812 17.99458 0.6576632
                 ACF1
Training set 0.135882

Forecasts:
        Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95
2014.50     -0.3002383 -0.3760127 -0.2244464 -0.4161253 -0.1843514
2015.00     -0.3002383 -0.3763907 -0.2240860 -0.4167033 -0.1837734
2015.50     -0.3002383 -0.3767667 -0.2237100 -0.4172784 -0.1831983
> checkresiduals(f31)

	Ljung-Box test

data:  Residuals from Simple exponential smoothing
Q* = 19.144, df = 4, p-value = 0.0007364

Model df: 0.   Total lags used: 4
```



2) Holts simple

```
Forecast method: Holt's method

Model Information:
Holt's method

Call:
 holt(y = RBO.tr, h = 3, initial = "simple")

  Smoothing parameters:
    alpha = 0.7249
    beta  = 0.1837

  Initial states:
    l = -0.281
    b = -0.0969

  sigma:  0.0796
Error measures:
                     ME       RMSE        MAE       MPE     MAPE     MASE
Training set 0.01355766 0.07956267 0.06302001 -8.828896 24.08236 0.865899
                   ACF1
Training set 0.01998612

Forecasts:
        Point Forecast      Lo 80       Hi 80      Lo 95        Hi 95
2014.50     -0.2288297 -0.3307933 -0.12686599 -0.3847696 -0.072889690
2015.00     -0.2155438 -0.3533143 -0.07777332 -0.4262456 -0.004842037
2015.50     -0.2022579 -0.3794222 -0.02509370 -0.4732073  0.068691395
> checkresiduals(f32)

        Ljung-Box test

data:  Residuals from Holt's method
Q* = 3.327, df = 4, p-value = 0.5047

Model df: 0.   Total lags used: 4
```



3) Holts with exponential trend

```
Forecast method: Holt's method with exponential trend

Model Information:
Holt's method with exponential trend

Call:
 holt(y = RBO.tr, h = 3, initial = "simple", exponential = TRUE)

  Smoothing parameters:
    alpha = 0.7417
    beta  = 0.2581

  Initial states:
    l = -0.281
    b = 1.3447

  sigma:  0.2658
Error measures:
                     ME       RMSE        MAE       MPE     MAPE      MASE
Training set 0.01935805 0.08295608 0.06495641 -10.75976 24.88312 0.8925052
                     ACF1
Training set -0.002607442

Forecasts:
        Point Forecast      Lo 80       Hi 80      Lo 95       Hi 95
2014.50     -0.2301822 -0.3084845 -0.15318670 -0.3480332 -0.11163240
2015.00     -0.2186152 -0.3344795 -0.12240103 -0.4128748 -0.08459783
2015.50     -0.2076295 -0.3706626 -0.09485428 -0.4939048 -0.06208476
> checkresiduals(f33)

        Ljung-Box test

data:  Residuals from Holt's method with exponential trend
Q* = 4.6137, df = 4, p-value = 0.3293

Model df: 0.   Total lags used: 4
```

4) Additive Damped holts method

```
Call:
 holt(y = RBO.tr, h = 3, damped = TRUE, initial = "simple")

  Smoothing parameters:
    alpha = 1e-04
    beta  = 1e-04
    phi   = 0.98

  Initial states:
    l = -0.2809
    b = -6e-04

  sigma:  0.0593

      AIC      AICc       BIC
-87.03774 -85.48219 -74.37250

Error measures:
                       ME       RMSE        MAE       MPE    MAPE      MASE
Training set -0.004210585 0.05685667 0.04609657 -2.873118 17.1346 0.6333698
                  ACF1
Training set 0.1502685

Forecasts:
        Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95
2014.50     -0.3013437 -0.3773918 -0.2252956 -0.4176492 -0.1850381
2015.00     -0.3015122 -0.3775603 -0.2254641 -0.4178177 -0.1852066
2015.50     -0.3016773 -0.3777254 -0.2256292 -0.4179829 -0.1853718
> checkresiduals(f34)

        Ljung-Box test

data:  Residuals from Damped Holt's method
Q* = 17.72, df = 4, p-value = 0.0014

Model df: 0.   Total lags used: 4
```

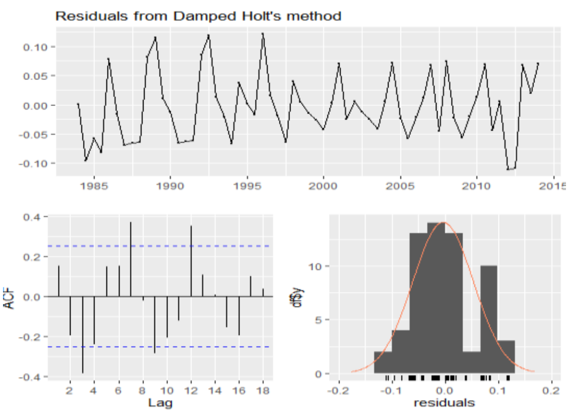
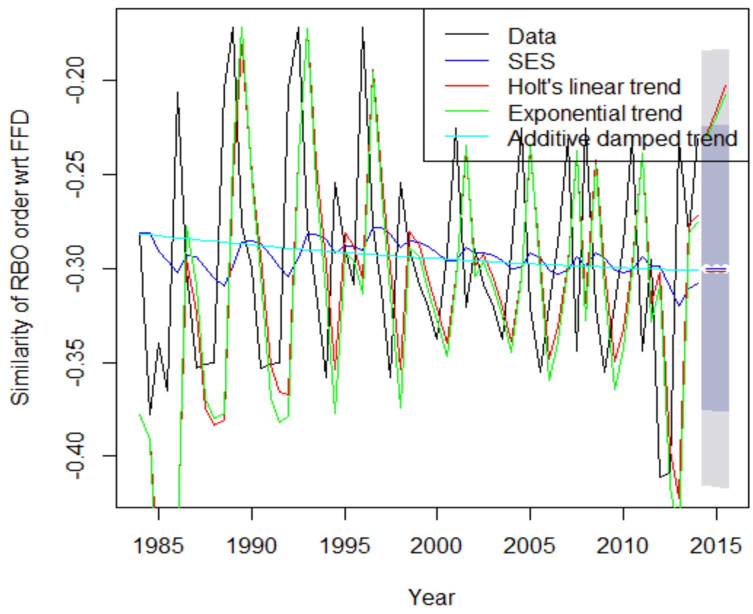


Fig.15 Forecasting of RBO wrt FFd values



CONCLUSION:

We have achieved the forecasts for the upcoming three years 2015,2016 and 2017 using a variety of time series analysis and modelling methodologies.

R CODE:

```
rm(list=ls())
############### Task # 01 ############
############### Data generation #######
mort_data<-
read.csv("C:/Users/mohamedmanzoor/desktop/Forecasting.zip/MM_Final_Project/csv/T1_mort.csv
")
colnames(mort_data)
head(mort_data,5)
summary(mort_data)
class(mort_data)
tail(mort_data)
t1_ts<-ts(mort_data[,2:6], start=c(2010,7), frequency=52)
class(t1_ts)
mortal_ts<-ts(mort_data$mortality, start=c(2010,7),frequency=52)
temp_ts<-ts(mort_data$temp, start=c(2010,7),frequency=52)
temp_ts
############## Finite Dynamic Linear Model (DLM)################
```

```
dataf = mort_data
dataf=dataf[,-1]
colnames(dataf) <- c("mortality", "temp", "X1", "X2","X3")

library(dLagM)
library(forecast)

for ( i in 1:10){
  model1.1 = dlm(formula = mortality ~ temp + +X1+X2+X3, data = data.frame(dataf), q = i )
  cat("q = ", i, "AIC = ", AIC(model1.1$model), "BIC = ", BIC(model1.1$model),"Mase
=",MASE(model1.1)$MASE, "\n")
}
############# Multiple Predictors for all indexes ##########
Model1.AllIndexes = dlm(formula = mortality ~ temp + X3, data = data.frame(dataf), q=10)
summary(Model1.AllIndexes)

residualcheck=function(x){
  shapiro.test(x$residuals)
}
######## Normality Test########

residualcheck(Model1.AllIndexes$model)

checkresiduals(Model1.AllIndexes$model)
########### Variance in flation Factor ###########
library(car)
VIF_m1 = vif(Model1.AllIndexes$model)
VIF_m1
VIF_m1 > 10
################### For Temperature############
#Temp
model1.temp <- dlm(x=as.vector(dataf$temp), y=as.vector(dataf$mortality), q=10)
summary(model1.temp)
checkresiduals(model1.temp)

############### For Chemical 1###############
model1.c1 <- dlm(x=as.vector(dataf$X1), y=as.vector(dataf$mortality), q=10)
summary(model1.c1)
checkresiduals(model1.c1)
############### For Chemical 2 ##############
model1.c2 <- dlm(x=as.vector(dataf$X2), y=as.vector(dataf$mortality), q=10)
summary(model1.c2)
checkresiduals(model1.c2)
################ For particle space ##############
model1.part <- dlm(x=as.vector(dataf$X3), y=as.vector(dataf$mortality), q=10)
```

```
summary(model1.part)
checkresiduals(model1.part)
finiteDLMauto(x=
as.vector(dataf$temp)+as.vector(dataf$X1)+as.vector(dataf$X2)+as.vector(dataf$X3), y=
as.vector(dataf$mortality),q.min = 1,q.max =10, k.order =1, model.type ="poly", error.type="AIC",
trace= TRUE)
################# As chemiclal 1 and partcle space have high correlation#########
finiteDLMauto(x= as.vector(dataf$X1), y= as.vector(dataf$mortality),q.min = 1,q.max =10, k.order
=1, model.type ="poly", error.type="AIC", trace= TRUE)
finiteDLMauto(x= as.vector(dataf$X3), y= as.vector(dataf$mortality),q.min = 1,q.max =10, k.order
=1, model.type ="poly", error.type="AIC", trace= TRUE)
attr(model1.c1$model, "class") ="lm"
AIC(model1.c1$model)
################## Polynomial DLM ################
################## Multiple predictors for all indexes#########3
Model2.AllIndexes <- polyDlm(x=
as.vector(dataf$temp)+as.vector(dataf$X1)+as.vector(dataf$X2)+as.vector(dataf$X3), y=
as.vector(dataf$mortality),q=10,k=1, show.beta = T)
summary(Model2.AllIndexes)
########### Variance inflation factor ############3
vif(Model2.AllIndexes$model)
residualcheck(Model2.AllIndexes$model)
checkresiduals(Model2.AllIndexes$model)
vif(Model2.AllIndexes$model)>10

################### For temperature ###########
pmodel1 = polyDlm(x = as.vector(dataf$temp) , y = as.vector(dataf$mortality),q=2,k = 2 , show.beta
= TRUE)
################### For Chemical 1 ##########
model2.c1 =polyDlm(x=as.vector(dataf$X1),y= as.vector(dataf$mortality),q=10,k=1, show.beta = T)
model2.p =polyDlm(x=as.vector(dataf$X3),y= as.vector(dataf$mortality),q=10,k=1, show.beta = T)
summary(model2.c1, diagnostics=T)
#################p value is large adjusted r square is lower#########
summary(model2.p, diagnostics=T)
########## pvalue is lowest and adjusted r square is acceptable low########
vif(model2.c1$model)>10
vif(model2.p$model)>10
attr(model2.c1$model, "class") ="lm"
AIC(model2.c1$model)

########KOYCK transformation#########
####One way to deal with this infinite DLM is to use Koyck transformation####
####### For all indexes #########
```

```
Model3.AllIndexes <-
koyckDlm(x=as.vector(dataf$temp)+as.vector(dataf$X1)+as.vector(dataf$X2)+as.vector(dataf$X3),
y= as.vector(dataf$mortality))
Model3.AllIndexes
summary(Model3.AllIndexes)
vif(Model3.AllIndexes$model)
vif(Model3.AllIndexes$model)>10
residualcheck(Model3.AllIndexes$model)
checkresiduals(Model3.AllIndexes$model)
vif(Model3.AllIndexes$model)>10
########### For Chemcial 1#########
model3.c1 =koyckDlm(x=as.vector(dataf$X1),y= as.vector(dataf$mortality))
model3.c1
summary(model3.c1, diagnostics=T)
vif(model3.c1$model)
vif(model3.c1$model)>10
#######3 For Particle Space####
model3.p =koyckDlm(x=as.vector(dataf$X3),y= as.vector(dataf$mortality))
#####p value is smaller adjusted r square is higher####
summary(model3.p, diagnostics=T)
vif(model3.p$model)
vif(model3.p$model)>10
####chemical 1 has no multicolinearity since values lie below lag value 10###
#### pvalue is lowest and adjusted r square is high####
#particle size contains no multicolinearity

#changed attribute of model to obtain aic value
attr(model3.c1$model, "class") ="lm"
AIC(model3.c1$model)

##############################AutoRegressive Dynamic Linear Model
(ARDL)#################################
for(i in 1:5){
  for (j in 1:5) {
    model4.allIndexes = ardlDlm(formula = mortality ~ temp + X3, data = data.frame(dataf), p= i, q=j)
    cat("p= ", i, "q= ", j,"AIC =", AIC(model4.allIndexes$model), "BIC =",
BIC(model4.allIndexes$model),"Mase=", MASE(model4.allIndexes)$MASE,"\n")

  }
}
for (i in c(3,4,5)){
  model4_ardl <- ardlDlm(formula = mortality ~ temp + X3, data = data.frame(dataf), p
              = i, q = 5)
  summary(model4_ardl)
```

```
  residualcheck(model4_ardl$model)

}
checkresiduals(model4_ardl$model)
####Based on the observation about model estimates made earlier, we can try to decrease the
####number of lags for predictor series. We will fit ARDL(1,5) and perform diagnostic checking.

####for p=1, q=5 ########
model4_1 = ardlDlm(formula = mortality ~ temp + X3, data = data.frame(dataf),p=1 ,q =5)$model
summary(model4_1)
residualcheck(model4_1)
checkresiduals(model4_1)
#####for p=3, q=5#######
model4_3 = ardlDlm(formula = mortality ~ temp + X3, data = data.frame(dataf), p =3, q=5)$model
summary(model4_3)
residualcheck(model4_3)
checkresiduals(model4_3)
#######for p=4, q=5#######
model4_4 = ardlDlm(formula = mortality ~ temp + X3, data = data.frame(dataf), p =4, q=5)$model
summary(model4_4)
residualcheck(model4_4)
checkresiduals(model4_4)
########for p=5, q=5#######
model4_5 = ardlDlm(formula = mortality ~ temp + X3, data = data.frame(dataf), p =5, q=5)$model
summary(model4_5)
residualcheck(model4_5)
checkresiduals(model4_5)
vif(model4_1)>10
vif(model4_2)>10
vif(model4_3)>10
vif(model4_4)>10
vif(model4_5)>10
####################################### Exponential Smoothing ###################
Mort1<-ts(mort_data$mortality,start = 2010,end =2018,frequency = 12)
hw1 <- hw(Mort1)
summary(hw1,)
checkresiduals(hw1)
hw2 <- hw(Mort1,seasonal="multiplicative")
summary(hw2)
checkresiduals(hw2)
hw3 <- hw(Mort1,seasonal="additive",damped = TRUE, h=5*frequency(Mort1))
summary(hw3)
checkresiduals(hw3)
hw4 <- hw(Mort1,seasonal="multiplicative",damped = TRUE, h=5*frequency(Mort1))
summary(hw4)
```

```
checkresiduals(hw4)
hw5 <- hw(Mort1,seasonal="multiplicative",exponential = TRUE, h=5*frequency(Mort1))
summary(hw5)
checkresiduals(hw5)
############################ finding best fit for each attribute###########
fit.expo = ets(mortal_ts, model="ZZZ", ic ="bic")
fit.expo$method
#### for temperature#####
fit.tem = ets(temp_ts, model="ZZZ", ic ="bic")
fit.tem$method
###### for Chemical 1 ####
fit.ch1 = ets(chem1_ts, model="ZZZ", ic ="bic")
fit.ch1$method
####### for chemical 2 #####
fit.ch2 = ets(chem2_ts, model="ZZZ", ic ="bic")
fit.ch2$method
#### for particle state######
fit.par = ets(part_ts, model="ZZZ", ic ="bic")
fit.par$method
################################State-space models variation#####################
ssmodel1=ets(mortal_ts,model = "ANN")
summary(ssmodel1)
ssmodel2=ets(mortal_ts,model = "MNN")
summary(ssmodel2)
ssmodel3=ets(mortal_ts,model = "AAN")
summary(ssmodel3)
ssmodel4=ets(mortal_ts,model = "MAN",damped = TRUE)
summary(ssmodel4)
ssmodel5=ets(mortal_ts,model = "MAN")
summary(ssmodel5)
vlist <- c("AAA", "MAA", "MAM", "MMM")
damp <- c(T,F)
ets_models <- expand.grid(vlist, damp)
ets_aic <- array(NA, 8)
ets_mase <- array(NA,8)
ets_bic <- array(NA,8)


auto_ets <- ets(head(mortal_ts,50))
summary(auto_ets)
checkresiduals(auto_ets)
```

```
############## Task # 02 ############
############## Data generation #######
read.csv("C:/Users/mohamedmanzoor/desktop/Forecasting.zip/MM_Final_Project/csv/._T2_FFD.csv
")
ffdata<-
read.csv("C:/Users/mohamedmanzoor/desktop/Forecasting.zip/MM_Final_Project/csv/T2_FFD.csv")
ffdata
library(dLagM)
library(forecast)
library(car)

##########################Converting into timeseries####################
ffdata_ts <- ts(ffdata, start=c(1984,1), frequency= 1)
head(ffdata_ts)
tail(ffdata_ts)
Ts_tempo<- ts(ffdata$Temperature, start =c(1984,1), frequency = 1)
head(Ts_tempo)
Ts_Rain <- ts(ffdata$Rainfall, start =c(1984,1), frequency = 1)
head(Ts_Rain)
Ts_Rad<- ts(ffdata$Radiation, start =c(1984,1), frequency = 1)
head(Ts_Rad)
Ts_Hum <- ts(ffdata$RelHumidity, start =c(1984,1), frequency = 1)
head(Ts_Hum)
Ts_FFD <- ts(ffdata$FFD, start =c(1984,1), frequency = 1)
head(Ts_FFD)
####################Data exploration and visualisation#################
par(mfrow=c(2,1))
plot(Ts_FFD,  main = "Fig.1 Time series plot of First flowering day series", ylab = "occurence of FFD
series", xlab = "Time")
acf(Ts_FFD, lag.max = 48, main="Fig.2 ACF plot of first flowering day series")
################## Augmented Dicky Fuller test#############
library(tseries)
adf.test(Ts_FFD, k=ar(Ts_FFD)$order)
par(mfrow=c(2,2))
plot(Ts_tempo, main ="Fig.3.1 Time series plot of temperature effects on ffd", ylab="Temperature
change", xlab = "Time")

plot(Ts_Rain, main ="Fig3.2.Time series plot of Rain effects on ffd series", ylab="Rainfall", xlab =
"Time")

plot(Ts_Rad, main ="Fig 3.3 Time series plot of Radiations on ffd series", ylab="Radiations", xlab =
"Time")

plot(Ts_Hum, main ="Fig 3.4 Time series plot of Humidity effects on ffd series", ylab="Humidity",
xlab = "Time")
```

```
###################ACF for 5 series##################
par(mfrow=c(2,2))
acf(Ts_tempo,lag.max = 48, main = "Fig.4.1 ACF plot of Temperature on FFD series")
adf.test(Ts_tempo,k=ar(Ts_tempo)$order)
acf(Ts_Rain,lag.max = 48, main = "Fig 4.2 ACF plot of Rain on FFD series")
adf.test(Ts_Rain,k=ar(Ts_Rain)$order)
acf(Ts_Rad,lag.max = 48, main = "Fig.4.3 ACF plot of Radiation on FFD series")
adf.test(Ts_Rad,k=ar(Ts_Rad)$order)
acf(Ts_Hum,lag.max = 48, main = "Fig 4.4 ACF plot of Humidity on FFD series")
adf.test(Ts_Hum,k=ar(Ts_Hum)$order)
##################scaling of data###################
shift<- scale(ffdata_ts)
plot(shift, plot.type="s",col=c("Red", "Blue", "Brown","Black","Green"),main= "Fig.5 FFD rate versus
factor affecting ffd wrt time(Scaled)")
legend("bottomright", lty=1, text.width = 7, col = c("Red", "Blue", "Brown","Black","Green"),
c("Temperature", "Rain", "Radiation", "Humidity","FFD"))
#######We also calculate the correlation coefficient to check the relationship.

cor(ffdata_ts)

for (i in 1:10){
  model1 <- dlm(x =
as.vector(ffdata$Temperature)+as.vector(ffdata$Rainfall)+as.vector(ffdata$Radiation)+as.vector(ffd
ata$RelHumidity), y = ffdata$FFD, q = i)
  cat("q =", i, "AIC =", AIC(model1$model), "BIC =", BIC(model1$model), "MASE =",
MASE(model1)$MASE, "\n")
}
###############1) Temperature
ftem_dlm <- dlm(x = ffdata$FFD, y = ffdata$Temperature, q=10)
summary(ftem_dlm)
vif(ftem_dlm$model)
################2)Rain

frain_dlm <- dlm(x = ffdata$FFD, y = ffdata$Rainfall, q=10)
summary(frain_dlm)
vif(frain_dlm$model)
##############3) radiation
frad_dlm <- dlm(x = ffdata$FFD, y = ffdata$Radiation, q=10)
summary(frad_dlm)
vif(frad_dlm$model)
###############4)Humidity

fhum_dlm <- dlm(x = ffdata$FFD, y = ffdata$RelHumidity, q=10)
summary(fhum_dlm)
vif(fhum_dlm$model)
```

```
#############Polynomial distributed lag model
########Polynomial modelling on univariate
#########1)     Temperature
residualcheck=function(x){
  shapiro.test(x$residuals)
}
temp_polyd <- polyDlm(x=as.vector(ffdata$Temperature), y=as.vector(ffdata$FFD), q=10,k=2)
summary(temp_polyd)
residualcheck(temp_polyd$model)
checkresiduals(temp_polyd$model)
#######2)Rainfall

rain_polyd <- polyDlm(x=as.vector(ffdata$Rainfall), y=as.vector(ffdata$FFD), q=10,k=2)
summary(rain_polyd)
residualcheck(rain_polyd$model)
checkresiduals(rain_polyd$model)
########3)Radiation

rad_polyd <- polyDlm(x=as.vector(ffdata$Radiation), y=as.vector(ffdata$FFD), q=10,k=2)
summary(rad_polyd)
residualcheck(rad_polyd$model)
checkresiduals(rad_polyd$model)
######### 4)Humidity

hum_polyd <- polyDlm(x=as.vector(ffdata$RelHumidity), y=as.vector(ffdata$FFD), q=10,k=2)
summary(hum_polyd)
residualcheck(hum_polyd$model)
checkresiduals(hum_polyd$model)

################################Koyck transformation###################
##########3We will implement Koyck transformation model with precipitation predictor series as
follows.

###########First we design multivariate model and then univariate models for each parameter

K_trans =
koyckDlm(x=as.vector(ffdata$Temperature)+as.vector(ffdata$Rainfall)+as.vector(ffdata$Radiation)+
as.vector(ffdata$RelHumidity), y=as.vector(ffdata$FFD))
summary(K_trans$model, diagnostics=T)
vif(K_trans$model)
############1)Temperature

temp_Koyck <- koyckDlm(x=as.vector(ffdata$Temperature), y=as.vector(ffdata$FFD))
summary(temp_Koyck)
```

```
vif(temp_Koyck$model, diagnostics =T)
checkresiduals(temp_Koyck$model)

###############2)Rain

rain_Koyck <- koyckDlm(x=as.vector(ffdata$Rainfall), y=as.vector(ffdata$FFD))
summary(rain_Koyck)
vif(rain_Koyck$model,diagnostics =T)
residualcheck(rain_Koyck$model)
checkresiduals(temp_Koyck$model)
############## 3)Radiation

rad_Koyck <- koyckDlm(x=as.vector(ffdata$Radiation), y=as.vector(ffdata$FFD))
summary(rad_Koyck)
vif(rad_Koyck$model)
residualcheck(rad_Koyck$model)
checkresiduals(rain_Koyck$model)
##########################humidity

hum_Koyck <- koyckDlm(x=as.vector(ffdata$RelHumidity), y=as.vector(ffdata$FFD))
summary(hum_Koyck)
vif(hum_Koyck$model)
residualcheck(hum_Koyck$model)
checkresiduals(hum_Koyck$model)
#################### Autoregressive DLM##############
for (i in 1:5){
  for(j in 1:5){
    model2 = ardlDlm(x = 
as.vector(ffdata$Temperature)+as.vector(ffdata$Rainfall)+as.vector(ffdata$Radiation)+as.vector(ffd
ata$RelHumidity), y = as.vector(ffdata$FFD), p = i , q = j)
    cat("p =", i, "q =", j, "AIC =", AIC(model2$model), "BIC =", BIC(model2$model), "MASE =", 
MASE(model2)$MASE, "\n")
  }
}
ardl_15 <- ardlDlm(x = 
as.vector(ffdata$Temperature)+as.vector(ffdata$Rainfall)+as.vector(ffdata$Radiation)+as.vector(ffd
ata$RelHumidity), y = as.vector(ffdata$FFD), p=1, q=5)
summary(ardl_15)
residualcheck(ardl_15$model)
#temperature
ardl_temp15 <- ardlDlm(x = as.vector(ffdata$Temperature), y = as.vector(ffdata$FFD), p=1, q=5)
summary(ardl_temp15)
residualcheck(ardl_temp15$model)
checkresiduals(ardl_temp15$model)
#rainfall
```

```
ardl_rain15 <- ardlDlm(x = as.vector(ffdata$Rainfall), y = as.vector(ffdata$FFD), p=1, q=5)
summary(ardl_rain15)
residualcheck(ardl_rain15$model)
checkresiduals(ardl_rain15$model)
#radiation
ardl_rad15 <- ardlDlm(x = as.vector(ffdata$Radiation), y = as.vector(ffdata$FFD), p=1, q=5)
summary(ardl_rad15)
residualcheck(ardl_rad15$model)
checkresiduals(ardl_rad15$model)
#humidity
ardl_hum15 <- ardlDlm(x = as.vector(ffdata$RelHumidity), y = as.vector(ffdata$FFD), p=1, q=5)
summary(ardl_hum15)
residualcheck(ardl_hum15$model)
checkresiduals(ardl_hum15$model)
attr(K_trans$model,"class") = "lm"

#temperature
ardl_temp35 <- ardlDlm(x = (ffdata$Temperature), y = (ffdata$FFD), p=3, q=5)
ardl_temp45 <- ardlDlm(x = as.vector(ffdata$Temperature), y = as.vector(ffdata$FFD), p=4, q=5)
ardl_temp55 <- ardlDlm(x = as.vector(ffdata$Temperature), y = as.vector(ffdata$FFD), p=5, q=5)

models <- c("FFtemp_DLM", "temp_PolyD", "temp_Koyck", "ARDL_temp15", "ARDL_temp35",
"ARDL_temp45", "ARDL_temp55")
aic_1 <- AIC( ftem_dlm,temp_polyd, temp_Koyck, ardl_temp15, ardl_temp35, ardl_temp45,
ardl_temp55)
bic_1 <- BIC(ftem_dlm, temp_polyd, temp_Koyck, ardl_temp15, ardl_temp35, ardl_temp45,
ardl_temp55)
MASE_1 <- MASE(ftem_dlm, temp_polyd, temp_Koyck, ardl_temp15, ardl_temp35, ardl_temp45,
ardl_temp55)
accuracy_1 <- data.frame(models, MASE_1, aic_1, bic_1 )
colnames(accuracy_1) <- c("Model", "MASE", "AIC", "BIC")
head(accuracy_1)
2)rainfall

ardl_rain35 <- ardlDlm(x = (ffdata$Rainfall), y = (ffdata$FFD), p=3, q=5)
ardl_rain45 <- ardlDlm(x = as.vector(ffdata$Rainfall), y = as.vector(ffdata$FFD), p=4, q=5)
ardl_rain55 <- ardlDlm(x = as.vector(ffdata$Rainfall), y = as.vector(ffdata$FFD), p=5, q=5)
#better compared to others
models <- c("FFrain_DLM", "rain_PolyD", "rain_Koyck", "ARDL_rain15", "ARDL_rain35",
"ARDL_rain45", "ARDL_rain55")
aic_2 <- AIC(frain_dlm, rain_polyd, rain_Koyck, ardl_rain15, ardl_rain35, ardl_rain45, ardl_rain55)
bic_2 <- BIC(frain_dlm, rain_polyd, rain_Koyck, ardl_rain15, ardl_rain35, ardl_rain45, ardl_rain55)
MASE_2 <- MASE(frain_dlm, rain_polyd, rain_Koyck, ardl_rain15, ardl_rain35, ardl_rain45,
ardl_rain55)
accuracy_2 <- data.frame(models, MASE_2, aic_2, bic_2 )
```

```
colnames(accuracy_2) <- c("Model", "MASE", "AIC", "BIC")
head(accuracy_2)
#radiation
ardl_rad35 <- ardlDlm(x = as.vector(ffdata$Radiation), y = as.vector(ffdata$FFD), p=3, q=5)
ardl_rad45 <- ardlDlm(x = as.vector(ffdata$Radiation), y = as.vector(ffdata$FFD), p=4, q=5)
ardl_rad55 <- ardlDlm(x = as.vector(ffdata$Radiation), y = as.vector(ffdata$FFD), p=5, q=5)

models <- c("FFrad_DLM", "rad_PolyD", "rad_Koyck", "ARDL_rad15", "ARDL_rad35", "ARDL_rad45",
"ARDL_rad55")
aic_3 <- AIC(frad_dlm, rad_polyd, rad_Koyck, ardl_rad15, ardl_rad35, ardl_rad45, ardl_rad55)
bic_3 <- BIC(frad_dlm, rad_polyd, rad_Koyck, ardl_rad15, ardl_rad35, ardl_rad45, ardl_rad55)
MASE_3 <- MASE(frad_dlm, rad_polyd, rad_Koyck, ardl_rad15, ardl_rad35, ardl_rad45, ardl_rad55)
accuracy_3 <- data.frame(models, MASE_3, aic_3, bic_3 )
colnames(accuracy_3) <- c("Model", "MASE", "AIC", "BIC")
head(accuracy_3)
#humidity
ardl_hum35 <- ardlDlm(x = as.vector(ffdata$RelHumidity), y = as.vector(ffdata$FFD), p=3, q=5)
ardl_hum45 <- ardlDlm(x = as.vector(ffdata$RelHumidity), y = as.vector(ffdata$FFD), p=4, q=5)
ardl_hum55 <- ardlDlm(x = as.vector(ffdata$RelHumidity), y = as.vector(ffdata$FFD), p=5, q=5)

#worst accuracy
models <- c("Fhum_DLM", "hum_PolyD", "hum_Koyck", "ARDL_hum15", "ARDL_hum35",
"ARDL_hum45", "ARDL_hum55")
aic_4 <- AIC(fhum_dlm, hum_polyd, hum_Koyck, ardl_hum15, ardl_hum35, ardl_hum45,
ardl_hum55)
bic_4 <- BIC(fhum_dlm, hum_polyd, hum_Koyck, ardl_hum15, ardl_hum35, ardl_hum45,
ardl_hum55)
MASE_4 <- MASE(fhum_dlm, hum_polyd, hum_Koyck, ardl_hum15, ardl_hum35, ardl_hum45,
ardl_hum55)
accuracy_4 <- data.frame(models, MASE_4, aic_4, bic_4 )
colnames(accuracy_4) <- c("Model", "MASE", "AIC", "BIC")
head(accuracy_4)
######################Exponential Smoothing################
ffd_ts <- ts(Ts_FFD, start=c(2015,1), frequency= 12)
ffd_ts
ES = c(T,F)
seasonality <- c("additive","multiplicative")
damped <- c(T,F)
expa <- expand.grid(ES, seasonality, damped)
expa <- expa[-c(1,5),]
f_aic <- array(NA, 6)
f_bic <- array(NA, 6)
f_mase <- array(NA, 6)
levels <- array(NA, dim=c(6,3))
for (i in 1:6){
```

```
 holt_w <- hw(ffd_ts, ES = expa[i,1], seasonal = toString(expa[i,2], damped = expa[i,3]))
 f_aic[i] <- holt_w$model$aic
 f_bic[i] <- holt_w$model$bic
 f_mase[i] <- accuracy(holt_w)[6]
 levels[i,1] <- expa[i,1]
 levels[i,2] <- toString(expa[i,2])
 levels[i,3] <- expa[i,3]
 checkresiduals(holt_w)
}
library(tidyr)
newvalues <- data.frame(levels, f_mase, f_aic, f_bic, NA)
colnames(newvalues) <- c("Trend", "Seasonality", "damped", "MASE", "AIC", "BIC","NA")
newvalues$Trend <- factor(newvalues$Trend, levels = c(T,F), labels = c("multiplicative","additive"))
newvalues$damped <- factor(newvalues$damped, levels = c(T,F), labels = c("damped","N"))

newvalues <- unite(newvalues, col = "Model", c("Trend","Seasonality","damped"))
accuracy_T <- rbind(accuracy_1, accuracy_2,accuracy_3,accuracy_4)
accuracy_T
########################State-space models variations###############
vlist <- c("AAA", "MAA", "MAM", "MMM")
damp <- c(T,F)
ets_models <- expand.grid(vlist, damp)
ets_aic <- array(NA, 8)
ets_mase <- array(NA,8)
ets_bic <- array(NA,8)
mod <- array(NA, dim=c(8,2))
#Auto ETS fitted to see what the software automatically suggested model is

auto_ets <- ets(ffd_ts)
summary(auto_ets)
checkresiduals(auto_ets)
library(tidyr)
calculate <- data.frame(mod, ets_mase, ets_aic, ets_bic,"NA")
calculate$X2 <- factor(calculate$X2, levels = c(T,F), labels = c("Damped","N"))
calculate <- unite(calculate, "Model", c("X1","X2"))
colnames(calculate) <- c("Model", "MASE", "AIC", "BIC","NA")
accuracy_T <- rbind(accuracy_1, accuracy_2,accuracy_3,accuracy_4)
accuracy_T
##############ues and 4 year forecasts are displayed in Figure 13.
fitm1 <- hw(ffd_ts, seasonal = "multiplicative", h = 2*frequency(ffd_ts))
fitm2 <- hw(ffd_ts, seasonal = "multiplicative", exponential = T, h = 2*frequency(ffd_ts))
fitm3 <- ets(ffd_ts,model="AAA", damped=T)
#class(fit3)
#methods(forecast())
for_fit3 <- forecast.ets(fitm3)
```

```
plot(for_fit3, fcol = "black", main = "FFD occurences series with four years ahead forecasts", ylab =
"ffd", ylim = c(-10,55))
lines(fitted(fitm1), col = "darkgreen")
lines(fitm1$mean, col = "darkgreen", lwd = 2)
lines(fitted(fitm2), col = "brown2")
lines(fitm2$mean, col = "brown2", lwd = 2)
lines(fitted(fitm3), col = "dodgerblue3")
lines(for_fit3$mean, col = "dodgerblue3", lwd = 2)
legend("bottomleft", lty = 1, col = c("black", "darkgreen", "brown2", "dodgerblue3"), c("Data", "Holt-
Winters' Multiplicative", "Holt-Winters' Multiplicative Exponential", "ETS(M,N,N)"))
plot(fitm1, fcol = "white", main = "FFD series with four years ahead forecasts", ylab = "ffd
occurences")
lines(fitted(fitm1), col = "darkgreen")
lines(fitm1$mean, col = "darkgreen", lwd = 2)
legend("topleft", lty = 1, col = c("black", "darkgreen"), c("Data", "Forecasts"))
#The solar radiation 2 years ahead point forecast values with corresponding 95% confidence
intervals are as follows:

forc <- fitm1$mean
ub <- fitm1$upper[,2]
lb <- fitm1$lower[,2]
forecasts <- ts.intersect(ts(lb, start = c(2015,1),end =c(2018,1) , frequency = 1), ts(forc,start =
c(2015,1),end =c(2018,1), frequency = 1), ts(ub,start = c(2015,1),end =c(2018,1), frequency = 1))
colnames(forecasts) <- c("Lower bound", "Point forecast", "Upper bound")
forecasts
plot(forecasts)




############## Task # 02 ############
############## Data generation #######
ffdata<-
read.csv("C:/Users/mohamedmanzoor/desktop/Forecasting.zip/MM_Final_Project/csv/ffdata.csv")
ffdata
library(dLagM)
library(forecast)
library(car)

##########################Converting into timeseries###################
ffdata_ts <- ts(ffdata, start=c(1984,1), frequency= 1)
head(ffdata_ts)
tail(ffdata_ts)
```

```
Ts_tempo<- ts(ffdata$Temperature, start =c(1984,1), frequency = 1)
head(Ts_tempo)
Ts_Rain <- ts(ffdata$Rainfall, start =c(1984,1), frequency = 1)
head(Ts_Rain)
Ts_Rad<- ts(ffdata$Radiation, start =c(1984,1), frequency = 1)
head(Ts_Rad)
Ts_Hum <- ts(ffdata$RelHumidity, start =c(1984,1), frequency = 1)
head(Ts_Hum)
Ts_FFD <- ts(ffdata$FFD, start =c(1984,1), frequency = 1)
head(Ts_FFD)
####################Data exploration and visualisation#################
par(mfrow=c(2,1))
plot(Ts_FFD,  main = "Fig.1 Time series plot of First flowering day series", ylab = "occurence of FFD series", xlab = "Time")
acf(Ts_FFD, lag.max = 48, main="Fig.2 ACF plot of first flowering day series")
################### Augmented Dicky Fuller test############
library(tseries)
adf.test(Ts_FFD, k=ar(Ts_FFD)$order)
par(mfrow=c(2,2))
plot(Ts_tempo, main ="Fig.3.1 Time series plot of temperature effects on ffd", ylab="Temperature change", xlab = "Time")

plot(Ts_Rain, main ="Fig3.2.Time series plot of Rain effects on ffd series", ylab="Rainfall", xlab = "Time")

plot(Ts_Rad, main ="Fig 3.3 Time series plot of Radiations on ffd series", ylab="Radiations", xlab = "Time")

plot(Ts_Hum, main ="Fig 3.4 Time series plot of Humidity effects on ffd series", ylab="Humidity", xlab = "Time")
###################ACF for 5 series###################
par(mfrow=c(2,2))
acf(Ts_tempo,lag.max = 48, main = "Fig.4.1 ACF plot of Temperature on FFD series")
adf.test(Ts_tempo,k=ar(Ts_tempo)$order)
acf(Ts_Rain,lag.max = 48, main = "Fig 4.2 ACF plot of Rain on FFD series")
adf.test(Ts_Rain,k=ar(Ts_Rain)$order)
acf(Ts_Rad,lag.max = 48, main = "Fig.4.3 ACF plot of Radiation on FFD series")
adf.test(Ts_Rad,k=ar(Ts_Rad)$order)
acf(Ts_Hum,lag.max = 48, main = "Fig 4.4 ACF plot of Humidity on FFD series")
adf.test(Ts_Hum,k=ar(Ts_Hum)$order)
##################scaling of data###################
shift<- scale(ffdata_ts)
plot(shift, plot.type="s",col=c("Red", "Blue", "Brown","Black","Green"),main= "Fig.5 FFD rate versus factor affecting ffd wrt time(Scaled)")
```

```
legend("bottomright", lty=1, text.width = 7, col = c("Red", "Blue", "Brown","Black","Green"),
c("Temperature", "Rain", "Radiation", "Humidity","FFD"))
#######We also calculate the correlation coefficient to check the relationship.

cor(ffdata_ts)

for (i in 1:10){
  model1 <- dlm(x =
as.vector(ffdata$Temperature)+as.vector(ffdata$Rainfall)+as.vector(ffdata$Radiation)+as.vector(ffd
ata$RelHumidity), y = ffdata$FFD, q = i)
  cat("q =", i, "AIC =", AIC(model1$model), "BIC =", BIC(model1$model), "MASE =",
MASE(model1)$MASE, "\n")
}
###############1) Temperature
ftem_dlm <- dlm(x = ffdata$FFD, y = ffdata$Temperature, q=10)
summary(ftem_dlm)
vif(ftem_dlm$model)
################2)Rain

frain_dlm <- dlm(x = ffdata$FFD, y = ffdata$Rainfall, q=10)
summary(frain_dlm)
vif(frain_dlm$model)
##############3) radiation
frad_dlm <- dlm(x = ffdata$FFD, y = ffdata$Radiation, q=10)
summary(frad_dlm)
vif(frad_dlm$model)
###############4)Humidity

fhum_dlm <- dlm(x = ffdata$FFD, y = ffdata$RelHumidity, q=10)
summary(fhum_dlm)
vif(fhum_dlm$model)

#############Polynomial distributed lag model
########Polynomial modelling on univariate
#########1)     Temperature
residualcheck=function(x){
  shapiro.test(x$residuals)
}
temp_polyd <- polyDlm(x=as.vector(ffdata$Temperature), y=as.vector(ffdata$FFD), q=10,k=2)
summary(temp_polyd)
residualcheck(temp_polyd$model)
checkresiduals(temp_polyd$model)
#######2)Rainfall

rain_polyd <- polyDlm(x=as.vector(ffdata$Rainfall), y=as.vector(ffdata$FFD), q=10,k=2)
```

```
summary(rain_polyd)
residualcheck(rain_polyd$model)
checkresiduals(rain_polyd$model)
########3)Radiation

rad_polyd <- polyDlm(x=as.vector(ffdata$Radiation), y=as.vector(ffdata$FFD), q=10,k=2)
summary(rad_polyd)
residualcheck(rad_polyd$model)
checkresiduals(rad_polyd$model)
######### 4)Humidity

hum_polyd <- polyDlm(x=as.vector(ffdata$RelHumidity), y=as.vector(ffdata$FFD), q=10,k=2)
summary(hum_polyd)
residualcheck(hum_polyd$model)
checkresiduals(hum_polyd$model)

################################Koyck transformation####################
##########3We will implement Koyck transformation model with precipitation predictor series as follows.

###########First we design multivariate model and then univariate models for each parameter

K_trans =
koyckDlm(x=as.vector(ffdata$Temperature)+as.vector(ffdata$Rainfall)+as.vector(ffdata$Radiation)+
as.vector(ffdata$RelHumidity), y=as.vector(ffdata$FFD))
summary(K_trans$model, diagnostics=T)
vif(K_trans$model)
############1)Temperature

temp_Koyck <- koyckDlm(x=as.vector(ffdata$Temperature), y=as.vector(ffdata$FFD))
summary(temp_Koyck)
vif(temp_Koyck$model, diagnostics =T)
checkresiduals(temp_Koyck$model)

###############2)Rain

rain_Koyck <- koyckDlm(x=as.vector(ffdata$Rainfall), y=as.vector(ffdata$FFD))
summary(rain_Koyck)
vif(rain_Koyck$model,diagnostics =T)
residualcheck(rain_Koyck$model)
checkresiduals(temp_Koyck$model)
############## 3)Radiation

rad_Koyck <- koyckDlm(x=as.vector(ffdata$Radiation), y=as.vector(ffdata$FFD))
summary(rad_Koyck)
```

```
vif(rad_Koyck$model)
residualcheck(rad_Koyck$model)
checkresiduals(rain_Koyck$model)
#########################humidity

hum_Koyck <- koyckDlm(x=as.vector(ffdata$RelHumidity), y=as.vector(ffdata$FFD))
summary(hum_Koyck)
vif(hum_Koyck$model)
residualcheck(hum_Koyck$model)
checkresiduals(hum_Koyck$model)
#################### Autoregressive DLM##############
for (i in 1:5){
  for(j in 1:5){
    model2 = ardlDlm(x =
as.vector(ffdata$Temperature)+as.vector(ffdata$Rainfall)+as.vector(ffdata$Radiation)+as.vector(ffd
ata$RelHumidity), y = as.vector(ffdata$FFD), p = i , q = j)
    cat("p =", i, "q =", j, "AIC =", AIC(model2$model), "BIC =", BIC(model2$model), "MASE =",
MASE(model2)$MASE, "\n")
  }
}
ardl_15 <- ardlDlm(x =
as.vector(ffdata$Temperature)+as.vector(ffdata$Rainfall)+as.vector(ffdata$Radiation)+as.vector(ffd
ata$RelHumidity), y = as.vector(ffdata$FFD), p=1, q=5)
summary(ardl_15)
residualcheck(ardl_15$model)
#temperature
ardl_temp15 <- ardlDlm(x = as.vector(ffdata$Temperature), y = as.vector(ffdata$FFD), p=1, q=5)
summary(ardl_temp15)
residualcheck(ardl_temp15$model)
checkresiduals(ardl_temp15$model)
#rainfall
ardl_rain15 <- ardlDlm(x = as.vector(ffdata$Rainfall), y = as.vector(ffdata$FFD), p=1, q=5)
summary(ardl_rain15)
residualcheck(ardl_rain15$model)
checkresiduals(ardl_rain15$model)
#radiation
ardl_rad15 <- ardlDlm(x = as.vector(ffdata$Radiation), y = as.vector(ffdata$FFD), p=1, q=5)
summary(ardl_rad15)
residualcheck(ardl_rad15$model)
checkresiduals(ardl_rad15$model)
#humidity
ardl_hum15 <- ardlDlm(x = as.vector(ffdata$RelHumidity), y = as.vector(ffdata$FFD), p=1, q=5)
summary(ardl_hum15)
residualcheck(ardl_hum15$model)
checkresiduals(ardl_hum15$model)
```

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

#temperature
ardl_temp35 <- ardlDlm(x = (ffdata$Temperature), y = (ffdata$FFD), p=3, q=5)
ardl_temp45 <- ardlDlm(x = as.vector(ffdata$Temperature), y = as.vector(ffdata$FFD), p=4, q=5)
ardl_temp55 <- ardlDlm(x = as.vector(ffdata$Temperature), y = as.vector(ffdata$FFD), p=5, q=5)

models <- c("FFtemp_DLM", "temp_PolyD", "temp_Koyck", "ARDL_temp15", "ARDL_temp35",
"ARDL_temp45", "ARDL_temp55")
aic_1 <- AIC( ftem_dlm,temp_polyd, temp_Koyck, ardl_temp15, ardl_temp35, ardl_temp45,
ardl_temp55)
bic_1 <- BIC(ftem_dlm, temp_polyd, temp_Koyck, ardl_temp15, ardl_temp35, ardl_temp45,
ardl_temp55)
MASE_1 <- MASE(ftem_dlm, temp_polyd, temp_Koyck, ardl_temp15, ardl_temp35, ardl_temp45,
ardl_temp55)
accuracy_1 <- data.frame(models, MASE_1, aic_1, bic_1 )
colnames(accuracy_1) <- c("Model", "MASE", "AIC", "BIC")
head(accuracy_1)
2)rainfall

ardl_rain35 <- ardlDlm(x = (ffdata$Rainfall), y = (ffdata$FFD), p=3, q=5)
ardl_rain45 <- ardlDlm(x = as.vector(ffdata$Rainfall), y = as.vector(ffdata$FFD), p=4, q=5)
ardl_rain55 <- ardlDlm(x = as.vector(ffdata$Rainfall), y = as.vector(ffdata$FFD), p=5, q=5)
#better compared to others
models <- c("FFrain_DLM", "rain_PolyD", "rain_Koyck", "ARDL_rain15", "ARDL_rain35",
"ARDL_rain45", "ARDL_rain55")
aic_2 <- AIC(frain_dlm, rain_polyd, rain_Koyck, ardl_rain15, ardl_rain35, ardl_rain45, ardl_rain55)
bic_2 <- BIC(frain_dlm, rain_polyd, rain_Koyck, ardl_rain15, ardl_rain35, ardl_rain45, ardl_rain55)
MASE_2 <- MASE(frain_dlm, rain_polyd, rain_Koyck, ardl_rain15, ardl_rain35, ardl_rain45,
ardl_rain55)
accuracy_2 <- data.frame(models, MASE_2, aic_2, bic_2 )
colnames(accuracy_2) <- c("Model", "MASE", "AIC", "BIC")
head(accuracy_2)
#radiation
ardl_rad35 <- ardlDlm(x = as.vector(ffdata$Radiation), y = as.vector(ffdata$FFD), p=3, q=5)
ardl_rad45 <- ardlDlm(x = as.vector(ffdata$Radiation), y = as.vector(ffdata$FFD), p=4, q=5)
ardl_rad55 <- ardlDlm(x = as.vector(ffdata$Radiation), y = as.vector(ffdata$FFD), p=5, q=5)

models <- c("FFrad_DLM", "rad_PolyD", "rad_Koyck", "ARDL_rad15", "ARDL_rad35", "ARDL_rad45",
"ARDL_rad55")
aic_3 <- AIC(frad_dlm, rad_polyd, rad_Koyck, ardl_rad15, ardl_rad35, ardl_rad45, ardl_rad55)
bic_3 <- BIC(frad_dlm, rad_polyd, rad_Koyck, ardl_rad15, ardl_rad35, ardl_rad45, ardl_rad55)
MASE_3 <- MASE(frad_dlm, rad_polyd, rad_Koyck, ardl_rad15, ardl_rad35, ardl_rad45, ardl_rad55)
accuracy_3 <- data.frame(models, MASE_3, aic_3, bic_3 )
colnames(accuracy_3) <- c("Model", "MASE", "AIC", "BIC")
```

```
head(accuracy_3)
#humidity
ardl_hum35 <- ardlDlm(x = as.vector(ffdata$RelHumidity), y = as.vector(ffdata$FFD), p=3, q=5)
ardl_hum45 <- ardlDlm(x = as.vector(ffdata$RelHumidity), y = as.vector(ffdata$FFD), p=4, q=5)
ardl_hum55 <- ardlDlm(x = as.vector(ffdata$RelHumidity), y = as.vector(ffdata$FFD), p=5, q=5)

#worst accuracy
models <- c("Fhum_DLM", "hum_PolyD", "hum_Koyck", "ARDL_hum15", "ARDL_hum35",
"ARDL_hum45", "ARDL_hum55")
aic_4 <- AIC(fhum_dlm, hum_polyd, hum_Koyck, ardl_hum15, ardl_hum35, ardl_hum45,
ardl_hum55)
bic_4 <- BIC(fhum_dlm, hum_polyd, hum_Koyck, ardl_hum15, ardl_hum35, ardl_hum45,
ardl_hum55)
MASE_4 <- MASE(fhum_dlm, hum_polyd, hum_Koyck, ardl_hum15, ardl_hum35, ardl_hum45,
ardl_hum55)
accuracy_4 <- data.frame(models, MASE_4, aic_4, bic_4 )
colnames(accuracy_4) <- c("Model", "MASE", "AIC", "BIC")
head(accuracy_4)
######################Exponential Smoothing#################
ffd_ts <- ts(Ts_FFD, start=c(2015,1), frequency= 12)
ffd_ts
ES = c(T,F)
seasonality <- c("additive","multiplicative")
damped <- c(T,F)
expa <- expand.grid(ES, seasonality, damped)
expa <- expa[-c(1,5),]
f_aic <- array(NA, 6)
f_bic <- array(NA, 6)
f_mase <- array(NA, 6)
levels <- array(NA, dim=c(6,3))
for (i in 1:6){
 holt_w <- hw(ffd_ts, ES = expa[i,1], seasonal = toString(expa[i,2], damped = expa[i,3]))
 f_aic[i] <- holt_w$model$aic
 f_bic[i] <- holt_w$model$bic
 f_mase[i] <- accuracy(holt_w)[6]
 levels[i,1] <- expa[i,1]
 levels[i,2] <- toString(expa[i,2])
 levels[i,3] <- expa[i,3]
 checkresiduals(holt_w)
}
library(tidyr)
newvalues <- data.frame(levels, f_mase, f_aic, f_bic, NA)
colnames(newvalues) <- c("Trend", "Seasonality", "damped", "MASE", "AIC", "BIC","NA")
newvalues$Trend <- factor(newvalues$Trend, levels = c(T,F), labels = c("multiplicative","additive"))
newvalues$damped <- factor(newvalues$damped, levels = c(T,F), labels = c("damped","N"))
```

```
newvalues <- unite(newvalues, col = "Model", c("Trend","Seasonality","damped"))
accuracy_T <- rbind(accuracy_1, accuracy_2,accuracy_3,accuracy_4)
accuracy_T
########################State-space models variations###############
vlist <- c("AAA", "MAA", "MAM", "MMM")
damp <- c(T,F)
ets_models <- expand.grid(vlist, damp)
ets_aic <- array(NA, 8)
ets_mase <- array(NA,8)
ets_bic <- array(NA,8)
mod <- array(NA, dim=c(8,2))
#Auto ETS fitted to see what the software automatically suggested model is

auto_ets <- ets(ffd_ts)
summary(auto_ets)
checkresiduals(auto_ets)
library(tidyr)
calculate <- data.frame(mod, ets_mase, ets_aic, ets_bic,"NA")
calculate$X2 <- factor(calculate$X2, levels = c(T,F), labels = c("Damped","N"))
calculate <- unite(calculate, "Model", c("X1","X2"))
colnames(calculate) <- c("Model", "MASE", "AIC", "BIC","NA")
accuracy_T <- rbind(accuracy_1, accuracy_2,accuracy_3,accuracy_4)
accuracy_T
##############ues and 4 year forecasts are displayed in Figure 13.
fitm1 <- hw(ffd_ts, seasonal = "multiplicative", h = 2*frequency(ffd_ts))
fitm2 <- hw(ffd_ts, seasonal = "multiplicative", exponential = T, h = 2*frequency(ffd_ts))
fitm3 <- ets(ffd_ts,model="AAA", damped=T)
#class(fit3)
#methods(forecast())
for_fit3 <- forecast.ets(fitm3)
plot(for_fit3, fcol = "black", main = "FFD occurences series with four years ahead forecasts", ylab =
"ffd", ylim = c(-10,55))
lines(fitted(fitm1), col = "darkgreen")
lines(fitm1$mean, col = "darkgreen", lwd = 2)
lines(fitted(fitm2), col = "brown2")
lines(fitm2$mean, col = "brown2", lwd = 2)
lines(fitted(fitm3), col = "dodgerblue3")
lines(for_fit3$mean, col = "dodgerblue3", lwd = 2)
legend("bottomleft", lty = 1, col = c("black", "darkgreen", "brown2", "dodgerblue3"), c("Data", "Holt-
Winters' Multiplicative", "Holt-Winters' Multiplicative Exponential", "ETS(M,N,N)"))
plot(fitm1, fcol = "white", main = "FFD series with four years ahead forecasts", ylab = "ffd
occurences")
lines(fitted(fitm1), col = "darkgreen")
lines(fitm1$mean, col = "darkgreen", lwd = 2)
```

```
legend("topleft", lty = 1, col = c("black", "darkgreen"), c("Data", "Forecasts"))
#The solar radiation 2 years ahead point forecast values with corresponding 95% confidence
intervals are as follows:

forc <- fitm1$mean
ub <- fitm1$upper[,2]
lb <- fitm1$lower[,2]
forecasts <- ts.intersect(ts(lb, start = c(2015,1),end =c(2018,1) , frequency = 1), ts(forc,start =
c(2015,1),end =c(2018,1), frequency = 1), ts(ub,start = c(2015,1),end =c(2018,1), frequency = 1))
colnames(forecasts) <- c("Lower bound", "Point forecast", "Upper bound")
forecasts
plot(forecasts)

############################# TASK 03####################
######Load Data##############
RBOdata <-
read.csv("C:/Users/mohamedmanzoor/desktop/Forecasting.zip/MM_Final_Project/csv/R.csv")
RBOdata
#Converting into timeseries

RBO_ts<- ts(RBOdata$RBO, start =c(1984,1), frequency = 1)
head(RBO_ts)
Temperature_ts <- ts(RBOdata$Temperature, start =c(1984,1), frequency = 1)
head(Temperature_ts)
RainFall_ts <-ts(RBOdata$Temperature, start =c(1984,1), frequency = 1)
head(RainFall_ts)
Radiation_ts <-ts(RBOdata$Radiation, start =c(1984,1), frequency = 1)
head(Radiation_ts)
##Finite distributed lag model
for (i in 1:10){
  model1 <- dlm(x = RBOdata$Temperature, y = RBOdata$RBO, q = i)
  cat("q =", i, "AIC =", AIC(model1$model), "BIC =", BIC(model1$model), "MASE =",
MASE(model1)$MASE, "\n")
}
#Temperature has lowest Aic at q=1 than rainfall data
for (i in 1:10){
  model1_r <- dlm(x = RBOdata$Rainfall, y = RBOdata$RBO, q = i)
  cat("q =", i, "AIC =", AIC(model1_r$model), "BIC =", BIC(model1_r$model), "MASE =",
MASE(model1_r)$MASE, "\n")
}
Finite dlm of each variate

1)Temperature
temp_dlm <- dlm(x = RBOdata$Temperature, y = RBOdata$RBO, q=10)
summary(temp_dlm)
```

```
vif(temp_dlm$model)
2)Rain

rain_dlm <- dlm(x = RBOdata$Rainfall, y = RBOdata$RBO, q=10)
summary(rain_dlm)
vif(rain_dlm$model)
3)Radiation

rad_dlm <- dlm(x = RBOdata$Radiation, y = RBOdata$RBO, q=10)
summary(rad_dlm)
vif(rad_dlm$model)
4)humidity

hum_dlm <- dlm(x = RBOdata$RelHumidity, y = RBOdata$RBO, q=10)
summary(hum_dlm)
vif(hum_dlm$model)

#################Polynomial distributed lag model
#################Polynomial modelling on univariate

##############1)Temperature
Temp_polyd3 <- polyDlm(x=as.vector(RBOdata$Temperature), y=as.vector(RBOdata$RBO),
q=10,k=2)
summary(Temp_polyd3, diagnostics=T)
residualcheck(Temp_polyd3$model)
checkresiduals(Temp_polyd3$model)
#########2)Rain

#better than others
Rain_polyd3 <- polyDlm(x=as.vector(RBOdata$Rainfall), y=as.vector(RBOdata$RBO), q=10,k=2)
summary(Rain_polyd3)
residualcheck(Rain_polyd3$model)
checkresiduals(Rain_polyd3$model)
########3)Radiation

Rad_polyd3 <- polyDlm(x=as.vector(RBOdata$Radiation), y=as.vector(RBOdata$RBO), q=10,k=2)
summary(Rad_polyd3)
residualcheck(Rad_polyd3$model)
checkresiduals(Rad_polyd3$model)
############4)Humidity

Humi_polyd3 <- polyDlm(x=as.vector(RBOdata$RelHumidity), y=as.vector(RBOdata$RBO), q=10,k=2)
summary(Humi_polyd3)
residualcheck(Humi_polyd3$model)
checkresiduals(Humi_polyd3$model)
```

```
########################Koyck transformation##################
########We will implement Koyck transformation model with precipitation predictor series as follows

K_total =
koyckDlm(x=as.vector(RBOdata$Temperature)+as.vector(RBOdata$Rainfall)+as.vector(RBOdata$Radiation)+as.vector(RBOdata$RelHumidity), y=as.vector(RBOdata$RBO))
summary(K_total$model, diagnostics=T)
vif(K_total$model)
######1)Temperature

Temp_Koyck3 <- koyckDlm(x=as.vector(RBOdata$Temperature), y=as.vector(RBOdata$RBO))
summary(Temp_Koyck3, diagnostics=T)
vif(Temp_Koyck3$model, diagnostics =T)
###########2)Rainfall

Rain_Koyck3 <- koyckDlm(x=as.vector(RBOdata$Rainfall), y=as.vector(RBOdata$RBO))
summary(Rain_Koyck3,diagnostics=T)
vif(Rain_Koyck3$model,diagnostics =T)

#########3)Radiation

Rad_Koyck3 <- koyckDlm(x=as.vector(RBOdata$Radiation), y=as.vector(RBOdata$RBO))
summary(Rad_Koyck3, diagnostics=T)
vif(Rad_Koyck3$model,diagnostics =T)
#########4)humidity

Humi_Koyck3 <- koyckDlm(x=as.vector(RBOdata$RelHumidity), y=as.vector(RBOdata$RBO))
summary(Humi_Koyck3,diagnostics=T)
vif(Humi_Koyck3$model,diagnostics =T)
par(mfrow=c(2,2))
residualcheck(Temp_Koyck3$model)
checkresiduals(Temp_Koyck3$model)
residualcheck(Rain_Koyck3$model)
checkresiduals(Rain_Koyck3$model)
residualcheck(Rad_Koyck3$model)
checkresiduals(Rad_Koyck3$model)
residualcheck(Humi_Koyck3$model)
checkresiduals(Humi_Koyck3$model)
############### Arutoregressive Distribution lag model#########
#########1)Temperature

ardl3_Temp15 <- ardlDlm(x = as.vector(RBOdata$Temperature), y = as.vector(RBOdata$RBO), p=1, q=5)
summary(ardl3_Temp15)
```

```
residualcheck(ardl3_Temp15$model)
checkresiduals(ardl3_Temp15$model)
########2)Rainfall

ardl3_Rain15 <- ardlDlm(x = as.vector(RBOdata$Rainfall), y = as.vector(RBOdata$RBO), p=1, q=5)
summary(ardl3_Rain15)
residualcheck(ardl3_Rain15$model)
checkresiduals(ardl3_Rain15$model)
#########3)Radiation

#best results
ardl3_Rad15 <- ardlDlm(x = as.vector(RBOdata$Radiation), y = as.vector(RBOdata$RBO), p=1, q=5)
summary(ardl3_Rad15)
residualcheck(ardl3_Rad15$model)
checkresiduals(ardl3_Rad15$model)
###4)Humidity

ardl3_Hum15 <- ardlDlm(x = as.vector(RBOdata$RelHumidity), y = as.vector(RBOdata$RBO), p=1,
q=5)
summary(ardl3_Hum15)
residualcheck(ardl3_Hum15$model)
checkresiduals(ardl3_Hum15$model)
##########Univariate Ardl modelling 1) Temperature

ardl3_Temp35 <- ardlDlm(x = as.vector(RBOdata$Temperature), y = as.vector(RBOdata$RBO), p=3,
q=5)
ardl3_Temp45 <- ardlDlm(x = as.vector(RBOdata$Temperature), y = as.vector(RBOdata$RBO), p=4,
q=5)
ardl3_Temp55 <- ardlDlm(x = as.vector(RBOdata$Temperature), y = as.vector(RBOdata$RBO), p=5,
q=5)

models <- c("Temp_DLM3", "Temp_PolyD3", "Temp_Koyck3", "ARDL3_temp15", "ARDL3_temp35",
"ARDL3_temp45", "ARDL3_temp55")
aic_a <- AIC(temp_dlm, Temp_polyd3, Temp_Koyck3, ardl3_Temp15, ardl3_Temp35, ardl3_Temp45,
ardl3_Temp55)

bic_a <- BIC(temp_dlm, Temp_polyd3, Temp_Koyck3, ardl3_Temp15, ardl3_Temp35, ardl3_Temp45,
ardl3_Temp55)

MASE_a <- MASE(temp_dlm, Temp_polyd3, Temp_Koyck3, ardl3_Temp15, ardl3_Temp35,
ardl3_Temp45, ardl3_Temp55)
accuracy_a <- data.frame(models, MASE_a, aic_a, bic_a )
colnames(accuracy_a) <- c("Model", "MASE", "AIC", "BIC")
head(accuracy_a)
########2)Rainfall
```

```
ardl3_Rain35 <- ardlDlm(x = as.vector(RBOdata$Rainfall), y = as.vector(RBOdata$RBO), p=3, q=5)
ardl3_Rain45 <- ardlDlm(x = as.vector(RBOdata$Rainfall), y = as.vector(RBOdata$RBO), p=4, q=5)
ardl3_Rain55 <- ardlDlm(x = as.vector(RBOdata$Rainfall), y = as.vector(RBOdata$RBO), p=5, q=5)

models <- c("Rain_DLM", "Rain_PolyD3", "Rain_Koyck3", "ARDL3_Rain15", "ARDL3_Rain35",
"ARDL3_Rain45", "ARDL3_Rain55")
aic_b <- AIC(rain_dlm, Rain_polyd3, Rain_Koyck3, ardl3_Rain15, ardl3_Rain35, ardl3_Rain45,
ardl3_Rain55)

bic_b <- BIC(rain_dlm, Rain_polyd3, Rain_Koyck3, ardl3_Rain15, ardl3_Rain35, ardl3_Rain45,
ardl3_Rain55)

MASE_b <- MASE(rain_dlm, Rain_polyd3, Rain_Koyck3, ardl3_Rain15, ardl3_Rain35, ardl3_Rain45,
ardl3_Rain55)
accuracy_b <- data.frame(models, MASE_b, aic_b, bic_b )
colnames(accuracy_b) <- c("Model", "MASE", "AIC", "BIC")
head(accuracy_b)

########3)Radiation

ardl3_Rad35 <- ardlDlm(x = as.vector(RBOdata$Radiation), y = as.vector(RBOdata$RBO), p=3, q=5)
ardl3_Rad45 <- ardlDlm(x = as.vector(RBOdata$Radiation), y = as.vector(RBOdata$RBO), p=4, q=5)
ardl3_Rad55 <- ardlDlm(x = as.vector(RBOdata$Radiation), y = as.vector(RBOdata$RBO), p=5, q=5)

models <- c("Rad_DLM", "Rain_PolyD3", "Rain_Koyck3", "ARDL3_Rain15", "ARDL3_Rain35",
"ARDL3_Rain45", "ARDL3_Rain55")
aic_c <- AIC(rad_dlm, Rad_polyd3, Rad_Koyck3, ardl3_Rad15, ardl3_Rad35, ardl3_Rad45,
ardl3_Rad55)

bic_c <- BIC(rad_dlm, Rad_polyd3, Rad_Koyck3, ardl3_Rad15, ardl3_Rad35, ardl3_Rad45,
ardl3_Rad55)

MASE_c <- MASE(rad_dlm, Rad_polyd3, Rad_Koyck3, ardl3_Rad15, ardl3_Rad35, ardl3_Rad45,
ardl3_Rad55)
accuracy_c <- data.frame(models, MASE_c, aic_c, bic_c )
colnames(accuracy_c) <- c("Model", "MASE", "AIC", "BIC")
head(accuracy_c)
#########4)Humidity

ardl3_Humi35 <- ardlDlm(x = as.vector(RBOdata$Rainfall), y = as.vector(RBOdata$RBO), p=3, q=5)
ardl3_Humi45 <- ardlDlm(x = as.vector(RBOdata$Rainfall), y = as.vector(RBOdata$RBO), p=4, q=5)
ardl3_Humi55 <- ardlDlm(x = as.vector(RBOdata$Rainfall), y = as.vector(RBOdata$RBO), p=5, q=5)
```

```
models <- c("Hum_DLM", "Humi_PolyD3", "Humi_Koyck3", "ARDL3_Hum15", "ARDL3_Humi35",
"ARDL3_Humi45", "ARDL3_Humi55")
aic_d <- AIC(hum_dlm, Humi_polyd3, Humi_Koyck3, ardl3_Hum15, ardl3_Humi35, ardl3_Humi45,
ardl3_Humi55)
## [1] -89.23348
bic_d <- BIC(hum_dlm, Humi_polyd3, Humi_Koyck3, ardl3_Hum15, ardl3_Humi35, ardl3_Humi45,
ardl3_Humi55)
## [1] -75.65468
MASE_d <- MASE(hum_dlm, Humi_polyd3, Humi_Koyck3, ardl3_Hum15, ardl3_Humi35,
ardl3_Humi45, ardl3_Humi55)
accuracy_d <- data.frame(models, MASE_d, aic_d, bic_d )
colnames(accuracy_d) <- c("Model", "MASE", "AIC", "BIC")
head(accuracy_d)
###Exonential Smoothing
library(forecast)
####For deciding on the final model to give three years ahead forecasts of solar radiation, we
compare forecasts from three models:
RBO_ts<- ts(RBOdata$RBO, start =c(1984,1), frequency = 1)
head(RBO_ts)
fit.auto =ets(RBO_ts,model="ZZZ",ic="bic")
fit.auto$method
f1.etsM = ets(RBO_ts, model="MNN")
summary(f1.etsM)
checkresiduals(f1.etsM)
#########1)simple exponential forecast

f1 <- ses(RBO_ts, alpha=0.1, initial="simple", h=3) # Set alpha to a small value
summary(f1)
checkresiduals(f1)
#####2)Holts simple forecast

f2 <- holt(RBO_ts,initial = "simple",h=3)
summary(f2)
checkresiduals(f2)
#########3)Holts with exponential trend
RBO_ts<- ts(RBOdata$RBO, start =c(1984,1), frequency = 1)
head(RBO_ts)

f33 <- holt(RBO_ts, initial="simple", exponential=TRUE, h=3)
# Fit with exponential trend
summary(f33)
checkresiduals(f33)
########4)Additive damped holts method

f4 <- holt(RBO_ts, damped=TRUE, initial="simple", h=3)
```

```
# Fit with additive damped trend
summary(f4)
checkresiduals(f4)
plot(f1, type="l", ylab="Similarity of RBO order wrt FFD", xlab="Year",main="Fig.13 forecasting of
fitted models",
    fcol="white", plot.conf=FALSE)
lines(fitted(f1), col="blue")
lines(fitted(f2), col="red")
lines(fitted(f3), col="green")
lines(fitted(f4), col="cyan")
lines(f1$mean, col="blue", type="l")
lines(f2$mean, col="red", type="l")
lines(f3$mean, col="green", type="l")
lines(f4$mean, col="brown", type="l")
legend("topright", lty=1, col=c("black","blue","red","green","cyan"),c("Data","SES", "Holt's linear
trend", "Exponential trend","Additive damped trend"))

################################ TASK 3(b)
library(TSA)
library(car)
library(dynlm)
library(Hmisc)
library(forecast)
library(stats)
RBO.ts = matrix(RBOdata$RBO, nrow = 25, ncol = 12)
RBO.ts = as.vector(t(RBO.ts))
RBO.ts = ts(RBO.ts,start=c(1984,1), end=c(2014,1), frequency=2)
class(RBO.ts)
plot(RBO.ts,ylab='RBO similarity of the order of FFD',xlab='Year',type='o',
    main = "Time series plot of RBOs.")
acf(RBO.ts,max.lag = 48, main="Sample ACF for RBOs")
RBO.tr = log(RBO.ts)
plot(RBO.tr,ylab='Log of landings in metric tons',xlab='Year',
    main = "Fig.14 Time series plot of the logarithm of yearly
similarity of order of RBOs.")
points(y=RBO.tr,x=time(RBO.tr), pch=as.vector(season(RBO.tr)))
Y.t = RBO.tr
T = 96
S.t = 1*(seq(Y.t) >= T)
S.t.1 = Lag(S.t,+1)
model31 = dynlm(Y.t ~ L(Y.t , k = 1 ) + S.t + trend(Y.t) + season(Y.t))
summary(model31)
model31.2 = dynlm(Y.t ~ L(Y.t , k = 1 ) + S.t + season(Y.t))
summary(model31.2)
model31.3 = dynlm(Y.t ~ L(Y.t , k = 1 ) + S.t + trend(Y.t) )
```

```
summary(model31.3)
aic = AIC(model31, model31.2, model31.3)
bic = BIC(model31, model31.2, model31.3)
aic
model32 = dynlm(Y.t ~ L(Y.t , k = 2 ) + S.t + trend(Y.t) + season(Y.t))
summary(model32)
model33 = dynlm(Y.t ~ L(Y.t , k = 1 ) + S.t + S.t.1 + trend(Y.t) + season(Y.t))
summary(model33)

##########1) Simple exponential forecasting

f31 <- ses(RBO.tr, alpha=0.1, initial="simple", h=3) # Set alpha to a small value
summary(f31)
checkresiduals(f31)
#####Holts simple
f32 <- holt(RBO.tr,initial = "simple",h=3)
summary(f32)
checkresiduals(f32)
###########Holts with exponential trend
f33 <- holt(RBO.tr, initial="simple", exponential=TRUE, h=3)
# Fit with exponential trend
summary(f33)
checkresiduals(f33)
####4)Additive Damped holts method

#
f34 <- holt(RBO.tr, damped=TRUE, initial="simple", h=3)
# Fit with additive damped trend
summary(f34)
checkresiduals(f34)
plot(f31, type="l", ylab="Similarity of RBO order wrt FFD", xlab="Year",main="Fig.15 Forecasting of
RBO wrt FFd values",
    fcol="white", plot.conf=FALSE)
lines(fitted(f31), col="blue")
lines(fitted(f32), col="red")
lines(fitted(f33), col="green")
lines(fitted(f34), col="cyan")
lines(f31$mean, col="blue", type="l")
lines(f32$mean, col="red", type="l")
lines(f33$mean, col="green", type="l")
lines(f34$mean, col="brown", type="l")
legend("topright", lty=1, col=c("black","blue","red","green","cyan"),c("Data","SES", "Holt's linear
trend", "Exponential trend","Additive damped trend"))
```