Impact of Transformations in Modeling and Forecasting with ARIMA

Course Title: Time Series Analysis and Forecasting
Course Code: WM-ASDS10, Fall 2023
Course Teacher: Prof. Dr. Rumana Rois

Submitted by
Dipayan Bhadra
Roll: 20231006
Batch: 10th Sec: A
PM-ASDS, JU

Contents

1. Introduction:

This assignment is prepared to fulfill the requirements of the course on Time Series Analysis and Forecasting (WM-ASDS-10), Fall-2023 semester. Last two digits of the ID =06, mod(06,06)=0, so nottem dataset is selected for this assignment.

2. Dataset:

"nottem" dataset is a time series object containing average monthly air temperatures at Nottingham Castle in degrees Fahrenheit for 20 years (1920–1939). There are 240 data points in the dataset arranged in monthly basis.

Table 01: nottem Dataset

Year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1920	40.6	40.8	44.4	46.7	54.1	58.5	57.7	56.4	54.3	50.5	42.9	39.8
1921	44.2	39.8	45.1	47	54.1	58.7	66.3	59.9	57	54.2	39.7	42.8
1922	37.5	38.7	39.5	42.1	55.7	57.8	56.8	54.3	54.3	47.1	41.8	41.7
1923	41.8	40.1	42.9	45.8	49.2	52.7	64.2	59.6	54.4	49.2	36.3	37.6
1924	39.3	37.5	38.3	45.5	53.2	57.7	60.8	58.2	56.4	49.8	44.4	43.6
1925	40	40.5	40.8	45.1	53.8	59.4	63.5	61	53	50	38.1	36.3
1926	39.2	43.4	43.4	48.9	50.6	56.8	62.5	62	57.5	46.7	41.6	39.8
1927	39.4	38.5	45.3	47.1	51.7	55	60.4	60.5	54.7	50.3	42.3	35.2
1928	40.8	41.1	42.8	47.3	50.9	56.4	62.2	60.5	55.4	50.2	43	37.3
1929	34.8	31.3	41	43.9	53.1	56.9	62.5	60.3	59.8	49.2	42.9	41.9
1930	41.6	37.1	41.2	46.9	51.2	60.4	60.1	61.6	57	50.9	43	38.8
1931	37.1	38.4	38.4	46.5	53.5	58.4	60.6	58.2	53.8	46.6	45.5	40.6
1932	42.4	38.4	40.3	44.6	50.9	57	62.1	63.5	56.3	47.3	43.6	41.8
1933	36.2	39.3	44.5	48.7	54.2	60.8	65.5	64.9	60.1	50.2	42.1	35.8
1934	39.4	38.2	40.4	46.9	53.4	59.6	66.5	60.4	59.2	51.2	42.8	45.8
1935	40	42.6	43.5	47.1	50	60.5	64.6	64	56.8	48.6	44.2	36.4
1936	37.3	35	44	43.9	52.7	58.6	60	61.1	58.1	49.6	41.6	41.3
1937	40.8	41	38.4	47.4	54.1	58.6	61.4	61.8	56.3	50.9	41.4	37.1
1938	42.1	41.2	47.3	46.6	52.4	59	59.6	60.4	57	50.7	47.8	39.2
1939	39.4	40.9	42.4	47.8	52.4	58	60.7	61.8	58.2	46.7	46.6	37.8

After close inspection of the dataset, it is clear that there is no missing data in the dataset. Also it seems there is no outlier in the dataset-that can be confirmed by plotting the data and tsoutliers(nottem) function. So forecast::tsclean() function is not required in this case.

tsoutliers(nottem) command can check outlier which gives the following output indication the absence of outliers in the dataset.

output:
```
$index
integer(0)

$replacements
numeric(0)
```

3. Plotting the data:

Time plot, ACF plot, PACF plot, sub0series plot and Boxplot have been used to investigate the main features of the time series data. The plots and the features of the salient features are given below.

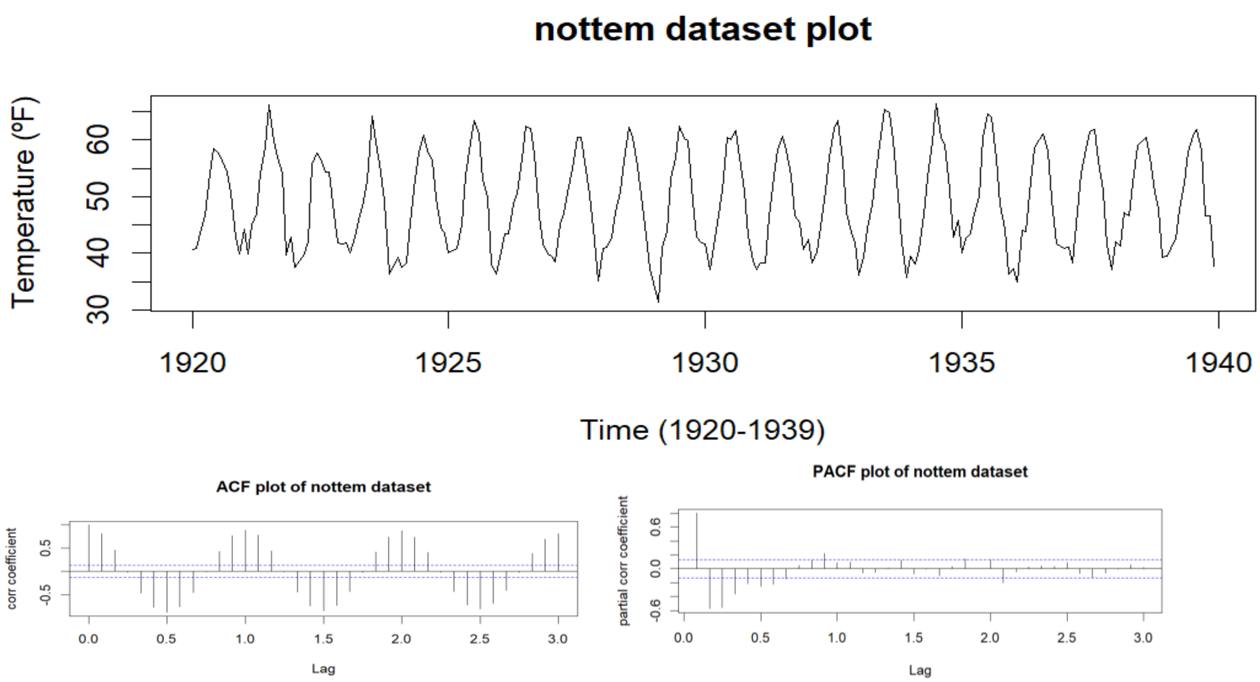


Figure 01: Time plot, ACF plot and PACF plot of nottem dataset

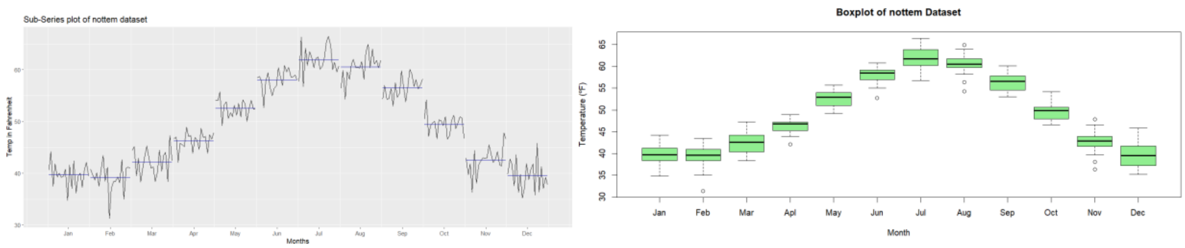


Figure 02: Sub-Series plot and Boxplot of nottem dataset

Main Features of the nottem dataset:

3.1. From the time-plot of the nottem dataset, we can see that, there is no trend in the dataset. Plot also in-consistent with this observation, because there is no slow decreasing correlations values in the ACF plot, rather very quick decrease in values indicates the possibility of absence of trend in the dataset.

3.2. Time-plot indicates additive seasonality. ACF plot confirms there is seasonality with frequency S=12.

3.3. ACF plot indicates that the series is alternating in nature.

3.4. the autocorrelations between the observations decrease quickly to zero indicates the possibility of the series being stationary.

3.5. The absence of sudden peak in ACF plot indicates the possibility of absence of outliers in the series. But Boxplot indicates the presence of outliers for some months (Mostly in August and November, single outlier in February, April and June) , but they are not outliers for the overall data.

3.6. Sub-Series plot and Boxplot show that the highest temperature occurs in July month with mean temperature around 62^{o}F. The lowest temperature occurs in February with lowest variations with mean around 39^{o}F. December to February is the winter showing almost same mean temperature. April-June and September-November exhibit the rapid changes in temperature.

3.7. ACF and PACF plot indicate Seasonal ARIMA model with p=2, q=2, s=12.

3.8. The Dataset needs to be tested for stationarity. If non-seasonal differencing is required, d value will be set accordingly.

3.9. Figure 3 shows mean temperature varies from month to month. So seasonal differencing (D=12) is required to be examined to get more appropriate seasonal adjusted model. Seasonal differencing removes seasonal trend and can also get rid of a seasonal random walk type of non-stationarity.

4. Split the dataset into training and test dataset:

There are 240 data pints (1920-1939, 20 years and 12 months in each year). The dataset is divided into training and test. 70% of the total data i.e. 192 data points (first 16 years) are considered as training and rest amount is considered as test (48 data points, last 4 years).

Table 02: Dataset split

	Total Dataset (nottem)	Training Dataset	Test Dataset
Length	240	168	72
No of years	20 years	14 years	6 years
Duration	Jan, 1920- Dec, 1939	Jan, 1920- Dec, 1933	Jan, 1934 – Dec, 1939
%	100%	70%	30%

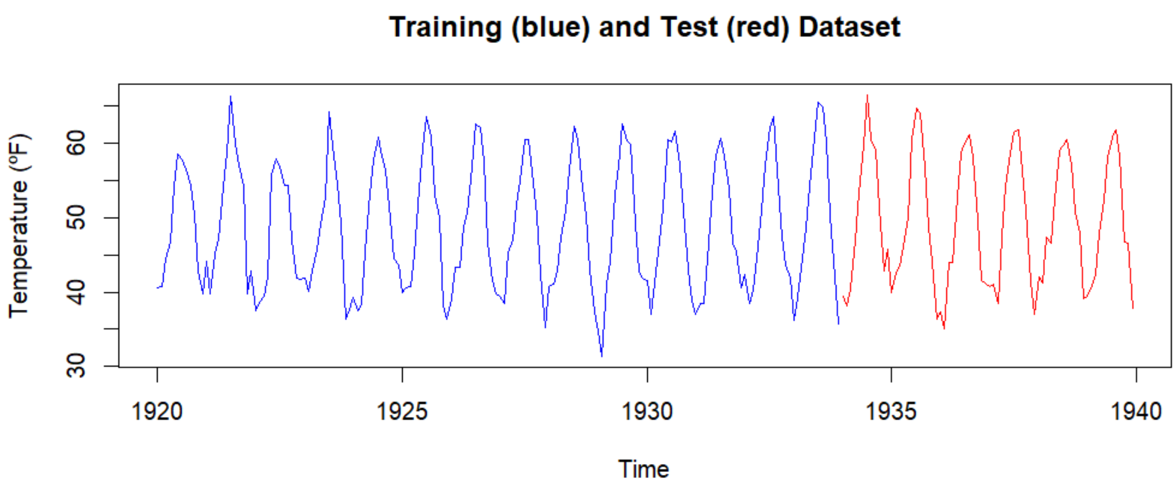


Figure: 03: Time-plot of Training and Test data

5. Stationarity Checking and the first model (M1):

5.1 Stationarity checking of train dataset

ADF and KPSS tests have been carried out to check the stationarity of the training dataset by adf.test() and kpss.test() functions.

Table 03: Stationarity Tests

Sl No	Name of the test	Hypothesis	Test Statistic value	P-value	Decision
1	ADF test	Ho: Non-Stationary H1: Stationary	-11.021	0.01	Ho may be rejected, **Stationary series**
2	KPSS test	Ho: Stationary H1: Non-Stationary	0.016241	0.1	Ho may not be rejected, **Stationary series**

Both tests indicate the **stationary series**. So, non-seasonal differencing is not needed as per the test results. **d=0.**

Since seasonality is present, so as per 3.9 seasonal differencing need to be examined and the stationarity of the seasonal differenced data need to be tested.

5.2 Stationarity checking after taking seasonal difference:

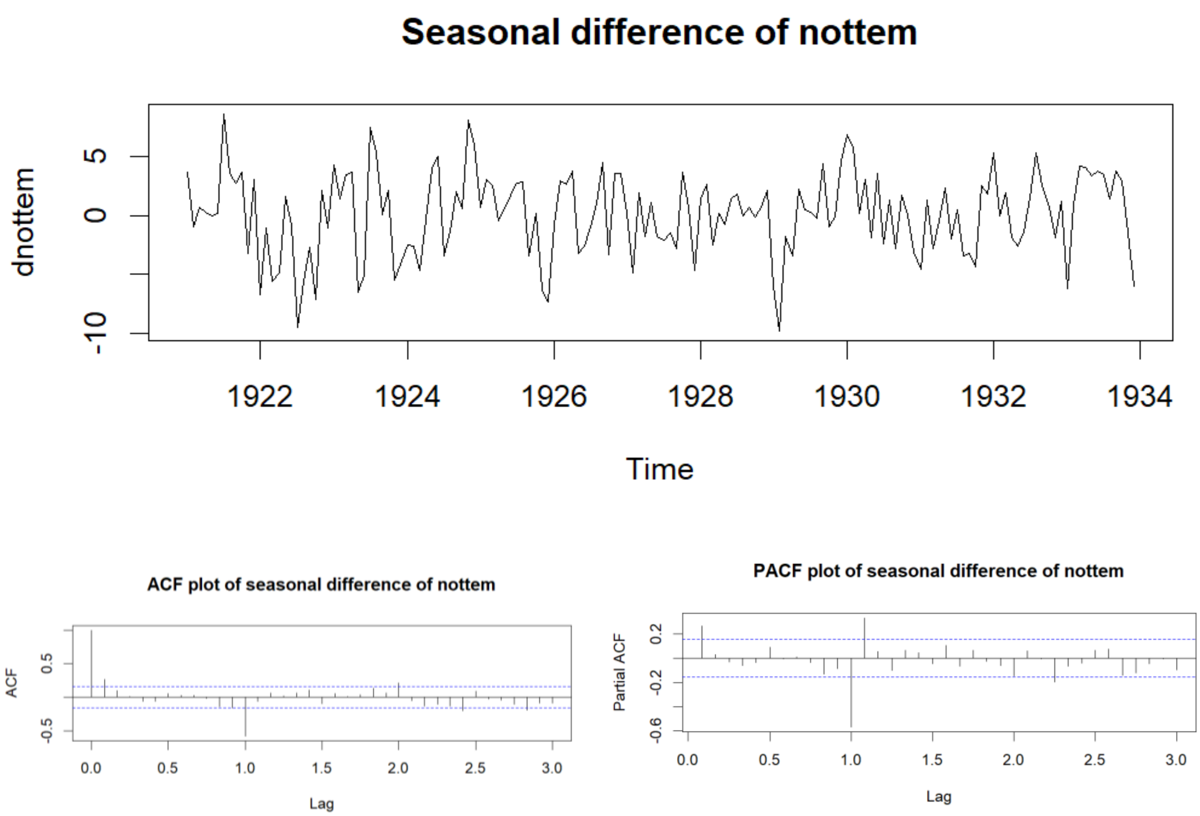


Figure 04: Seasonal Difference plot, ACF and PACF plot of nottem dataset

ADF and KPSS tests have been done to check the stationarity of the training dataset.

Table 04: Stationarity Test on the seasonal difference data

Sl No	Name of the test	Hypothesis	Test Statistic value	P-value	Decision
1	ADF test	Ho: Non-Stationary H1: Stationary	-4.5604	0.01	Ho may be rejected, **Stationary series**

2	KPSS test	Ho: Stationary H1: Non-Stationary	0.071104	0.1	Ho may not be rejected, **Stationary series**

Both tests indicate the **stationary series**. Moreover, the ACF and PACF plot indicates stationary series. So, seasonal differencing is needed as per the test results. **D=1.** ACF and PACF plot of seasonal difference data indicate Seasonal ARIMA model with P=1, Q=1, s=12 i.e. proposed model is **SARIMA(p=2, d=0, q=2)(P=1, D=1, Q=1)[s=12]. Some other variation were tested (table 05). SARIMA(p=2, d=0, q=2)(P=1, D=1, Q=1)[s=12] gives the best result among them.**

Table 05: Model selection for training data

Model for Log Transformed Training Data	AIC	AICc	BIC	σ^2	Remarks
SARIMA (3,0,2)(1,1,1)[12] ar1 ar2 ar3 ma1 ma2 sar1 sma1 1.3038 -1.2185 0.2323 -1.0109 1.0000 -0.3921 -0.6272	721.39	722.37	745.79	4.966	Some co-eff is larger than 1
SARIMA (2,0,2)(1,1,1)[12] ar1 ar2 ma1 ma2 sar1 sma1 0.8050 -0.4937 -0.4920 0.4196 -0.3437 -0.6577	729.81	730.57	751.16	5.497	**Selected as M1 (Lowest AIC, AICc)**
SARIMA (3,0,1)(1,1,1)[12] ar1 ar2 ar3 ma1 sar1 sma1 0.4920 0.0016 -0.0828 -0.1581 -0.3437 -0.6538	730.61	731.37	751.96	5.531	Higher AIC and BIC
SARIMA (3,0,1)(2,1,0)[12] ar1 ar2 ar3 ma1 sar1 sar2 0.4268 0.0283 -0.0813 -0.0701 -0.8713 -0.3013	738.95	739.7	760.29	5.972	Higher AIC and BIC

5.3 Model 1

So the selected model is SARIMA(p=2, d=0, q=2)(P=1, D=1, Q=1)[s=12]

```
M1 =Arima(trainData, order=c(2,0,2), seasonal = list(order=c(1,1,1), period=12))
> M1
Series: trainData
ARIMA(2,0,2)(1,1,1)[12]

Coefficients:
         ar1      ar2      ma1     ma2     sar1     sma1
      0.8050  -0.4937  -0.4920  0.4196  -0.3437  -0.6577
s.e.  0.3637   0.3528   0.4041  0.3110   0.1003   0.0962

sigma^2 = 5.497:  log likelihood = -357.91
AIC=729.81   AICc=730.57   BIC=751.16
```

Mathematical expression of ARIMA(2,0,2)(1,1,1)[12] is given below:

$$(1-\varphi_1 B-\varphi_2 B^2)(1-B^{12})(1-\Phi_1 B^{12})Y_t=(1+\theta_1 B+\theta_2 B^2)(1+\Theta_1 B^{12})Z_t,\ \ Z_t \sim N(0,\sigma^2)$$

Putting the values of the co-efficient from the expression becomes:

$$(1 - 0.8050B + 0.4937B^2)(1 - B^{12})(1 + 0.3437B^{12})Y_t$$
$$= (1 - 0.4920B + 0.4196B^2)(1 - 0.6577B^{12})Z_t, \;\; Z_t \sim N(0, 5.497)$$

Residual Diagnostic Checking:

```
> test(M1$residuals)
Null hypothesis: Residuals are iid noise.
Test                        Distribution Statistic   p-value
Ljung-Box Q                 Q ~ chisq(20)     16.03    0.7147
McLeod-Li Q                 Q ~ chisq(20)     11.71    0.9257
Turning points T  (T-110.7)/5.4 ~ N(0,1)       103    0.1584
Diff signs S       (S-83.5)/3.8 ~ N(0,1)        83     0.894
Rank P          (P-7014)/364.5 ~ N(0,1)      7670    0.0719
> shapiro.test(M1$residuals)

         Shapiro-Wilk normality test

data:  M1$residuals
W = 0.98788, p-value = 0.1574
```

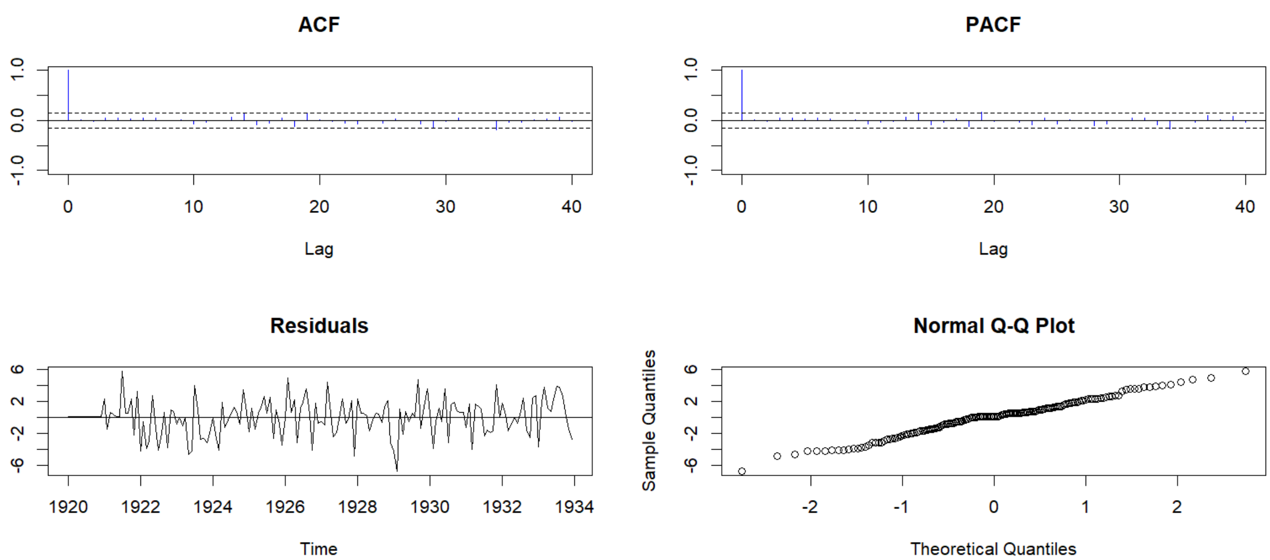


Figure 05: Residual Diagnostic plot of model 1 (M1)

Decision: All the iid noise tests of residuals (L-Jung Box,Q, McLeod-Li Q, Turning points T, Diff Sign S, Rank P) indicate that the p-value is greater than the level of significance, α = 0.05, so that the Null Hypothesis may not be rejected, that is the residuals are iid noise. The Shaprio-Wilk Normality test indicate that residuals are normally distributed.

6. Transformation and the Second Model (M2):

In this step, among three transformations, log, square root and cubic root, a single transformation need to be selected. Since the series does not have any trend, only stationary seasonal pattern, hence to select the series, a comparison of the test statistics need to be evaluated first:

Table 06: Transformations and the stationarity test

Serial No	Transformation	ADF Test	KPSS test	Stationarity Decision	Remarks
1	Log transform	ADF=-11.26, p-Val= 0.01 Ho rejected	KPSS level=0.013822 p-Val= 0.1 Ho not-rejected	Stationary series	Lowest ADF value and KPSS level
2	Square-root transform	ADF=-11.139, p-Val= 0.01 Ho rejected	KPSS level=0.014865 p-Val= 0.1 Ho not-rejected	Stationary series	
3	Cubic Root transform	ADF=-11.182, p-Val= 0.01 Ho rejected	KPSS level=0.01448 p-Val= 0.1 Ho not-rejected	Stationary series	

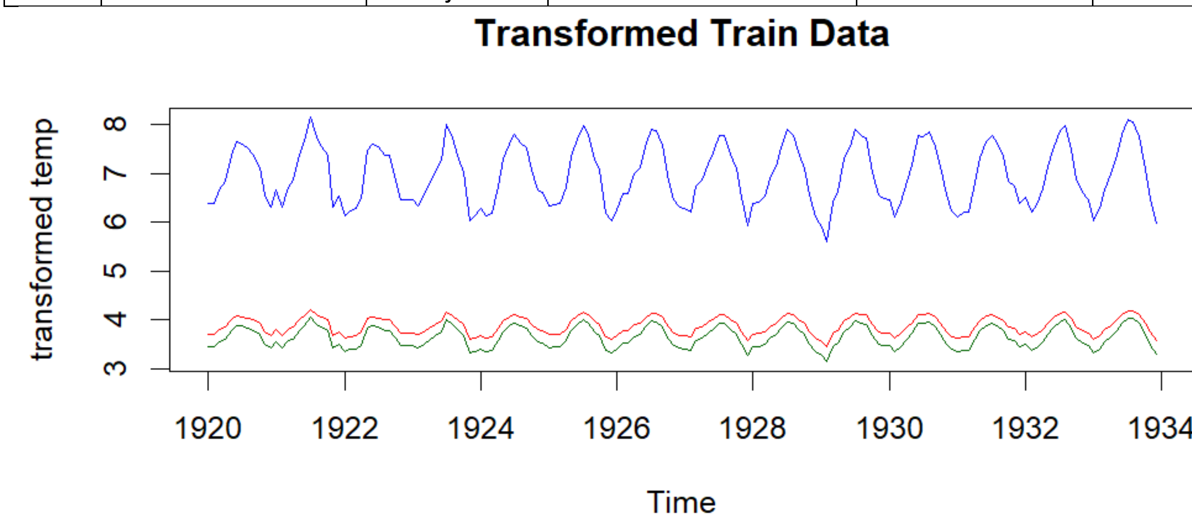


Figure 06: Log transformation (red), Cubic-root (green) and square-root transformation (blue)

From the graph more constant variation of values can be observed from log transformation data. Also Table 4 shows that ADF and KPSS statistic value is the most extreme for log transformed data. **So log transformation is considered for model 2**.

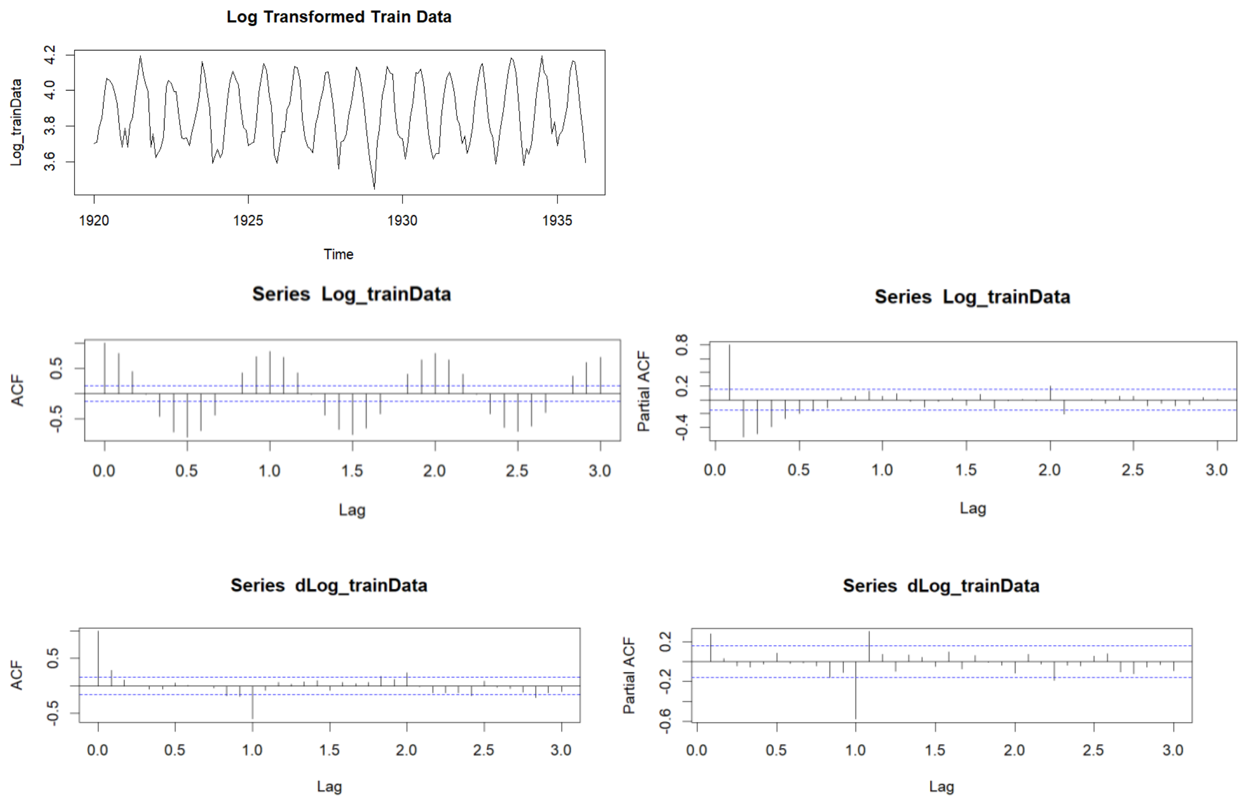


Figure 07: Seasonal Difference plot, ACF and PACF plot of log transformed training data (nottem dataset)

Due to the seasonality, D=1. 3.7. ACF and PACF plot indicate Seasonal ARIMA model with p=2, q=2, s=12. ACF and PACF of log transformed data ACF and PACF plot of seasonal difference data indicate Seasonal ARIMA model with P=1, Q=1, s=12. But the model, SARIMA(p=2, d=0, q=2)(P=1, D=1, Q=1)[s=12] gives some co-efficients larger than 1, and very close to 1. SARIMA(p=2, d=0, q=1)(P=1, D=1, Q=1)[s=12] model provides good co-efficient values while maintaining the AIC and BIC values. **Hence SARIMA(p=2, d=0, q=1)(P=1, D=1, Q=1)[s=12] model has been chosen as M2 for the log transformed train data.**

Table 07: Model selection for Log-Transformed training data

Model for Log Transformed Training Data	AIC	AICc	BIC	σ^2	Remarks
SARIMA (3,0,2)(1,1,1)[12] ar1 ar2 ar3 ma1 ma2 sar1 sma1 1.2740 -1.1929 0.2146 -0.9998 1.0000 -0.3706 -0.6442	-465.07	-464.09	-440.67	0.002474	Some co-eff is larger than 1
SARIMA (2,0,2)(1,1,1)[12] ar1 ar2 ma1 ma2 sar1 sma1 1.0591 -0.9529 -0.9874 1.0000 -0.3460 -0.6733	-460.35	-459.60	-439.00	0.002541	Some co-eff is larger than 1
SARIMA (2,0,1)(1,1,1)[12] ar1 ar2 ma1 sar1 sma1 -0.1980 0.2164 0.5072 -0.3093 -0.6790	-457.82	-457.26	-439.52	0.002734	Selected as M2

```
M2=Arima(Log_trainData, order=c(2,0,1), seasonal = list(order=c(1,1,1), period=12))
>M2
```

```
Series: Log_trainData
ARIMA(2,0,1)(1,1,1)[12]

Coefficients:
          ar1     ar2     ma1     sar1     sma1
      -0.1980  0.2164  0.5072  -0.3093  -0.6790
s.e.   0.6217  0.1975  0.6267   0.1061   0.1039

sigma^2 = 0.002734:  log likelihood = 234.91
AIC=-457.82   AICc=-457.26   BIC=-439.52
```

Mathematical expression of ARIMA(2,0,1)(1,1,1)[12] is given below:

$$(1-\varphi_1 B-\varphi_2 B^2)(1-B^{12})(1-\Phi_1 B^{12})Y_t=(1+\theta_1 B)(1+\Theta_1 B^{12})Z_t,\ \ Z_t \sim N(0,\sigma^2)$$

Putting the values of the co-efficient from the expression becomes:

$$\begin{aligned}(1+0.1980B-0.2164B^2)(1-B^{12})(1+0.3093B^{12})Y_t \\ =(1+0.5072B)(1-0.6790B^{12})Z_t,\ \ Z_t \sim N(0,0.002734)\end{aligned}$$

Residual Diagnostic Checking:

```
> test(M2$residuals) ##p-value = 2.431e-05
Null hypothesis: Residuals are iid noise.
Test                         Distribution Statistic   p-value
Ljung-Box Q                 Q ~ chisq(20)     22.54    0.3121
McLeod-Li Q                 Q ~ chisq(20)     23.59     0.261
Turning points T  (T-110.7)/5.4 ~ N(0,1)       103    0.1584
Diff signs S       (S-83.5)/3.8 ~ N(0,1)        83     0.894
Rank P          (P-7014)/364.5 ~ N(0,1)       7631    0.0905
> shapiro.test(M2$residuals)

        Shapiro-Wilk normality test

data:  M2$residuals
W = 0.97123, p-value = 0.001454
```

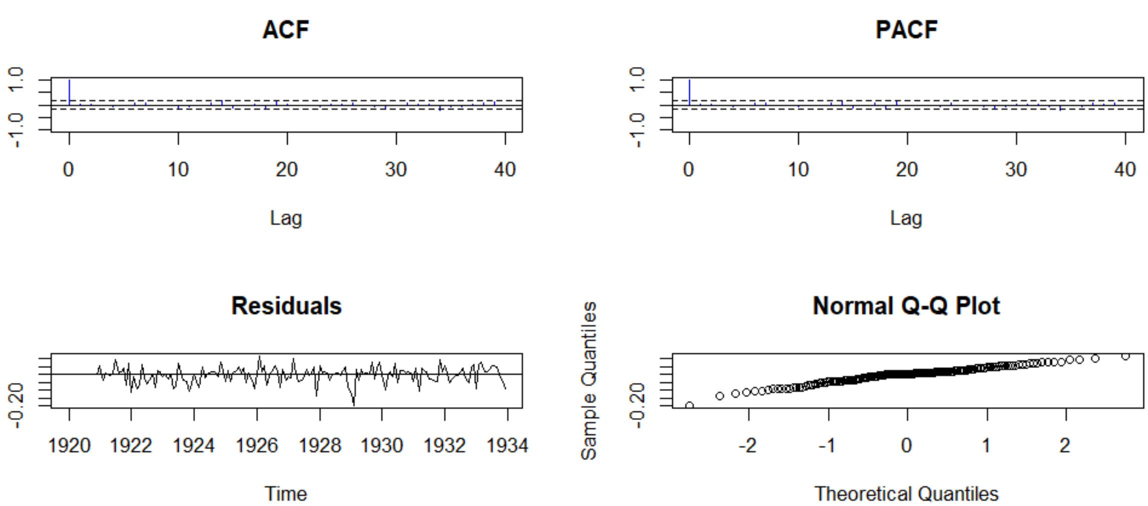


Figure 08: Residual Diagnostic plot of model 2 (M2)

Decision: All the iid noise tests of residuals (L-Jung Box,Q, McLeod-Li Q, Turning points T, Diff Sign S, Rank P) indicate that the p-value is greater than the level of significance, α = 0.05, so that the Null Hypothesis may not be rejected, that is the residuals are iid noise.

7. BoxCox Transformation and Third Model (M3)

In this step, the Boxcox transformation need to be applied with the optimum lambda, then a suitable model needs to be selected after making the series stationary. Thereafter, model (M3) need to be chosen using the ACF & PACF plots of the stationary series.

BoxCox.lambda(trainData) gives the optimal lambda.

```
    Lambda (optimal) = -0.544016

Boxcox_trainData=BoxCox(trainData, lambda = lamda)
plot(Boxcox_trainData)

> adf.test(Boxcox_trainData)

    Augmented Dickey-Fuller Test

data:  Boxcox_trainData
Dickey-Fuller = -11.332, Lag order = 5, p-value = 0.01
alternative hypothesis: stationary


> kpss.test(Boxcox_trainData)

    KPSS Test for Level Stationarity

data:  Boxcox_trainData
KPSS Level = 0.013088, Truncation lag parameter = 4, p-value = 0.1
```

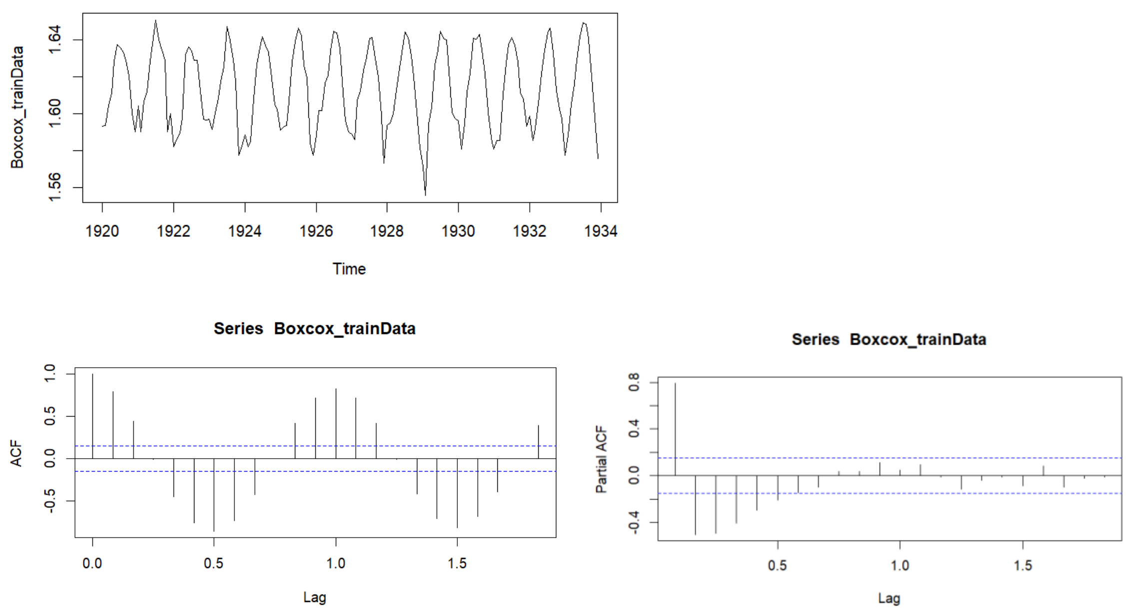


Figure 09: Seasonal Difference plot, ACF and PACF plot of BoxCox transformed training data

Due to the seasonality, D=1. 3.7. ACF and PACF plot indicate Seasonal ARIMA model with p=2, q=2, s=12. ACF and PACF of log transformed data ACF and PACF plot of seasonal difference data indicate Seasonal ARIMA model with P=1, Q=1, s=12. But the model, SARIMA(p=2, d=0, q=2)(P=1, D=1, Q=1)[s=12] gives some co-efficients larger than 1, and very close to 1. SARIMA(p=2, d=0, q=1)(P=1, D=1, Q=1)[s=12] model provides good co-efficient values while maintaining the AIC and BIC values. **Hence SARIMA(p=2, d=0, q=1)(P=1, D=1, Q=1)[s=12] model has been chosen as M3 for the BoxCox transformed train data.**

Table 8: Model selection for automated BoxCox transformation

Model for Log Transformed Training Data	AIC	AICc	BIC	σ^2	Remarks
SARIMA (2,0,2)(1,1,1)[12] ar1 ar2 ma1 ma2 sar1 sma1 1.0508 -0.9508 -0.979 0.9987 -0.3391 -0.6822	-1097.59	-1096.84	-1076.24	4.3×10^{-05}	Some co-eff is larger than 1
SARIMA (2,0,1)(1,1,1)[12] ar1 ar2 ma1 sar1 sma1 -0.1980 0.2164 0.5072 -0.3093 -0.6790	-1095.46	-1094.89	-1077.16	4.602×10^{-05}	Selected as M3

```
M3 =Arima(Boxcox_trainData, order=c(2,0,2), seasonal = list(order=c(1,1,1), period=12))
>M3

Series: Boxcox_trainData
ARIMA(2,0,1)(1,1,1)[12]

Coefficients:
          ar1     ar2     ma1     sar1     sma1
      -0.2121  0.2175  0.5139  -0.2901  -0.6954
s.e.   0.5888  0.1834  0.5933   0.1066   0.1036

sigma^2 = 4.602e-05:  log likelihood = 553.73
AIC=-1095.46   AICc=-1094.89   BIC=-1077.16
```

Mathematical expression of ARIMA(2,0,1)(1,1,1)[12] is given below:

$$(1-\varphi_1 B-\varphi_2 B^2)(1-B^{12})(1-\Phi_1 B^{12})Y_t=(1+\theta_1 B)(1+\Theta_1 B^{12})Z_t,\ \ Z_t \sim N(0,\sigma^2)$$

Putting the values of the co-efficient from the expression becomes:

$$(1+0.2121B-0.2175B^2)(1-B^{12})(1+0.2901B^{12})Y_t$$
$$=(1+0.5139B)(1-0.6954B^{12})Z_t,\ \ Z_t \sim N(0,4.062\times 10^{-05})$$

Residual Diagnostic Checking:

```
> test(M3$residuals)
Null hypothesis: Residuals are iid noise.
Test                        Distribution Statistic   p-value
Ljung-Box Q                  Q ~ chisq(20)     21.52     0.367
McLeod-Li Q                  Q ~ chisq(20)     31.02    0.0549
Turning points T  (T-110.7)/5.4 ~ N(0,1)        99    0.0318 *
Diff signs S        (S-83.5)/3.8 ~ N(0,1)        83     0.894
Rank P           (P-7014)/364.5 ~ N(0,1)      7446     0.236
> shapiro.test(M3$residuals)

         Shapiro-Wilk normality test

data:  M3$residuals
W = 0.95092, p-value = 1.361e-05
```

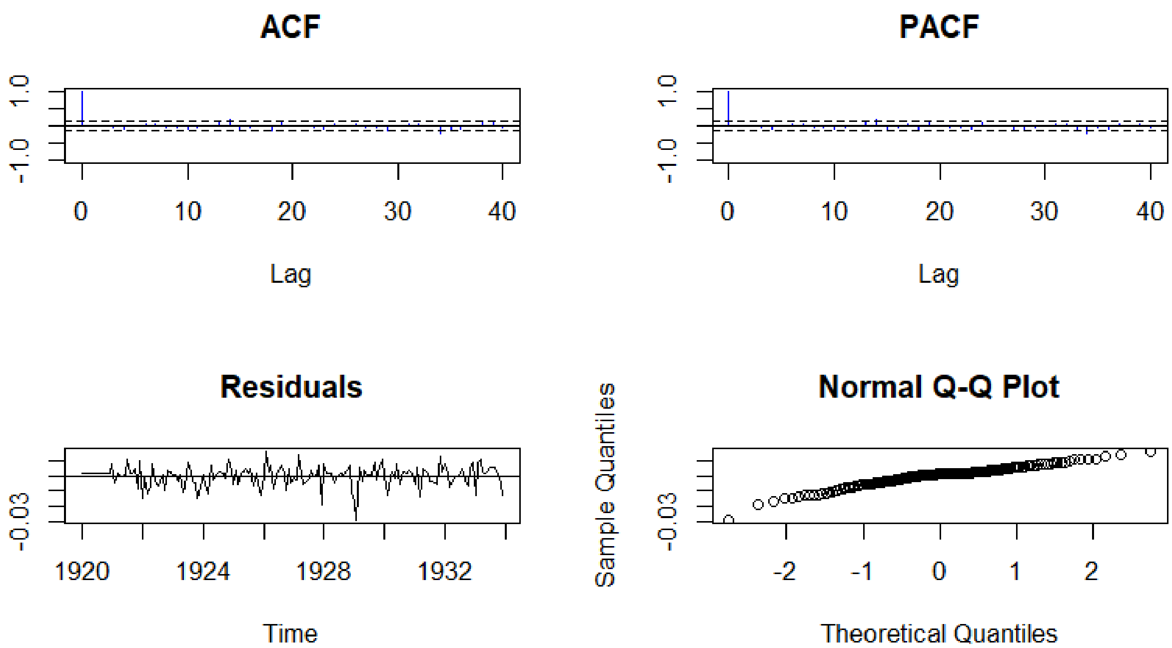


Figure 10: Residual Diagnostic plot of model 3 (M3)

Decision: All the iid noise tests of residuals (L-Jung Box,Q, McLeod-Li Q, Diff Sign S, Rank P) except , Turning points T indicate that the p-value is greater than the level of significance, α = 0.05, so that the Null Hypothesis may not be rejected, that is the residuals are iid noise.

8. Using Automated function (auto.arima) of R to get best Model:

In this step, auto.arima function will be used to get the Model M4 based on the training dataset. M4=auto.arima(trainData) provides the following M4 model:

```
>M4=auto.arima(trainData)
>M4
Series: trainData
ARIMA(1,0,0)(2,1,0)[12]

Coefficients:
         ar1      sar1      sar2
      0.3634   -0.8649   -0.2887
s.e.  0.0757    0.0804    0.0845

sigma^2 = 5.9:  log likelihood = -363
AIC=733.99   AICc=734.26   BIC=746.19
```

Mathematical expression of ARIMA(1,0,0)(2,1,1)[12] is given below:

$$(1-\varphi_1 B)(1-B^{12})(1-\Phi_1 B^{12}-\Phi_2 B^{24})Y_t = Z_t,\ \ Z_t \sim N(0,\sigma^2)$$

Putting the values of the co-efficient from the expression becomes:

$$(1-0.3634B)(1-B^{12})(1+0.8649B^{12}+0.2887B^{24})Y_t = Z_t,\ \ Z_t \sim N(0, 5.9)$$

Residual Diagnostic Checking:

```
> test(M4$residuals)
Null hypothesis: Residuals are iid noise.
Test                        Distribution Statistic   p-value
Ljung-Box Q                 Q ~ chisq(20)      20.73    0.4131
McLeod-Li Q                 Q ~ chisq(20)      12.97    0.8788
Turning points T  (T-110.7)/5.4 ~ N(0,1)        107    0.4999
Diff signs S       (S-83.5)/3.8 ~ N(0,1)         80     0.351
Rank P           (P-7014)/364.5 ~ N(0,1)       7419    0.2665
> shapiro.test(M4$residuals)

         Shapiro-Wilk normality test

data:  M4$residuals
W = 0.99169, p-value = 0.4417
```

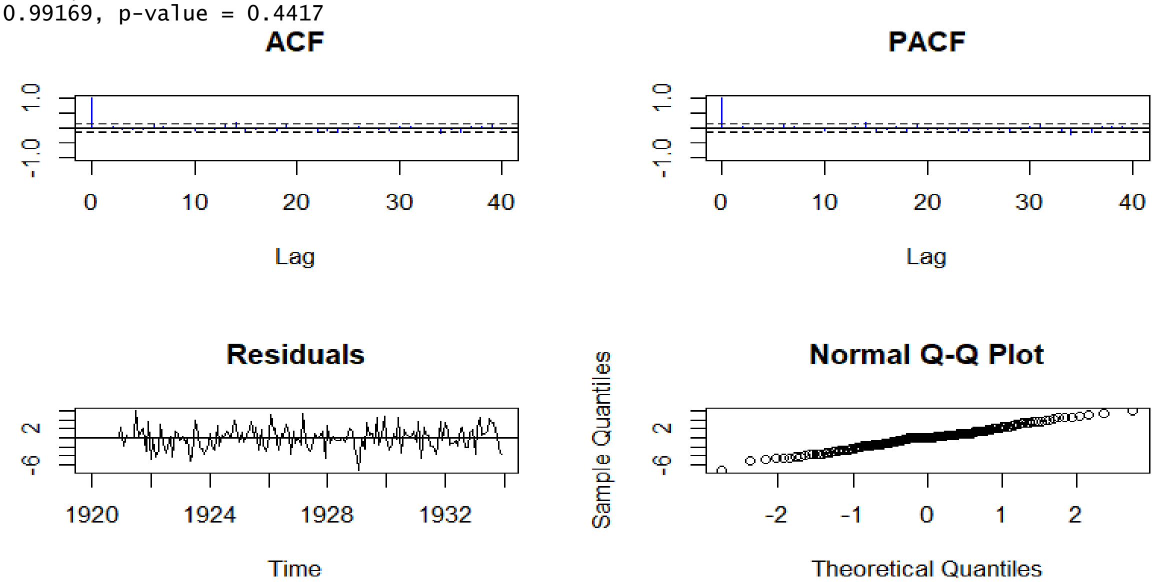


Figure 11: Residual Diagnostic plot of model 4 (M4)

Decision: All the iid noise tests of residuals (L-Jung Box,Q, McLeod-Li Q, Turning points T, Diff Sign S, Rank P) indicate that the p-value is greater than the level of significance, α = 0.05, so that the Null Hypothesis may not be rejected, that is the residuals are iid noise. The Shaprio-Wilk Normality test indicate that residuals are normally distributed.

9. Model comparison based on training dataset:

Following table represents AIC, AICc, BIC values for the models of this assignment based on training dataset.

Table 9: Model performance analysis based on AIC, AICc and BIC values

Model	AIC	AICc	BIC	σ^2	Assumptions fulfillment	Remarks
M1: SARIMA (2,0,2)(1,1,1)[12]	729.81	730.57	751.16	5.497347	Residuals iid noise and normally distributed	Best Model
M2: Log Transformation + SARIMA (2,0,1)(1,1,1)[12]	-457.82	-457.26	-439.52	0.002734252 (5.543032)	Residuals iid noise but not normally distributed	Transformation gives wrong AIC, BIC, σ^2 values. Corrected σ^2 are given inside bracket.
M3: BoxCox Transform+ SARIMA (2,0,1)(1,1,1)[12]	-1095.46	-1094.89	-1077.16	4.601902e-05 (5.595308)	Residuals iid noise but not normally distributed	
M4: SARIMA (1,0,0)(2,1,0)[12]	733.99	734.26	746.19	5.900475	Residuals iid noise and normally distributed	

Decision: Based on the AIC, AICc and BIC values on the training dataset, model 1 (M1) is the best model among the four models because of its lower AIC and AICc values. Note that AIC, AICc and BIC values of M2 and M3 are based on the transformed data, so the lower values are misleading and can not be used for this comparison. But corrected SSE (σ^2) shows that M1 performs better than all other models.

Note: To get corrected SSE (σ^2) following codes has been used

```
>sum((M1$fitted-trainData)^2)/150 or M1$sigma2
>sum((exp(M2$fitted)-trainData)^2)/151
>sum(((lamda*M3$fitted+1)^(1/lamda)-trainData)^2)/151
>sum((M4$fitted-trainData)^2)/153 or M4$sigma2
Where M2$sigma2 gives 0.002734252
And M3$sigma2 gives 4.601902e-05
```

10. Forecasting test data

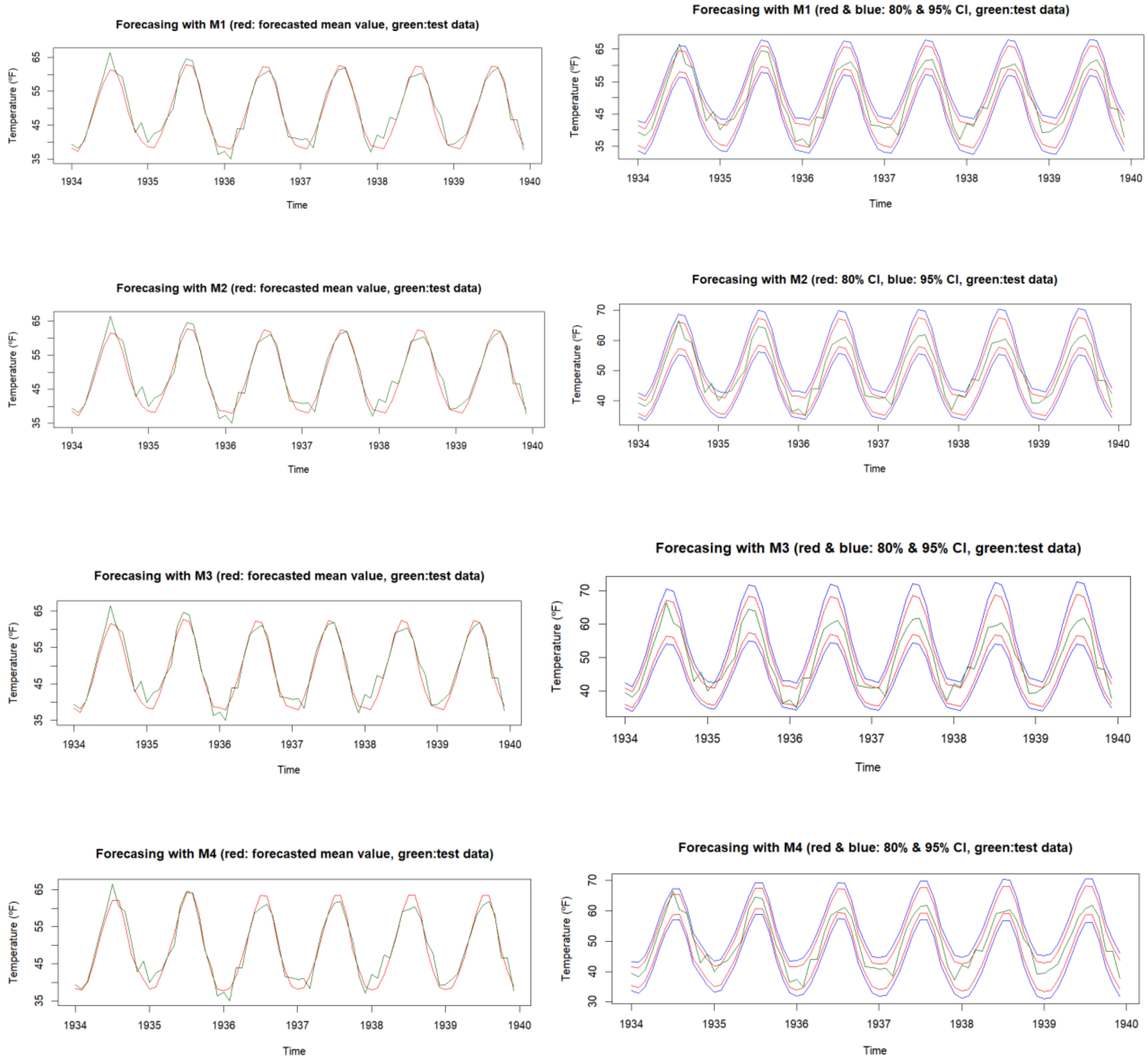
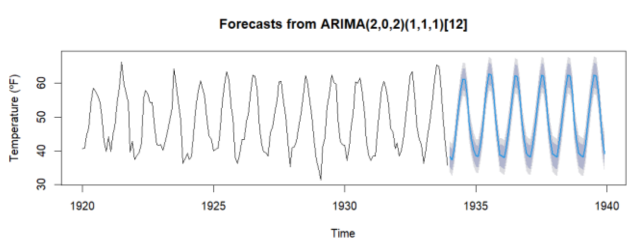
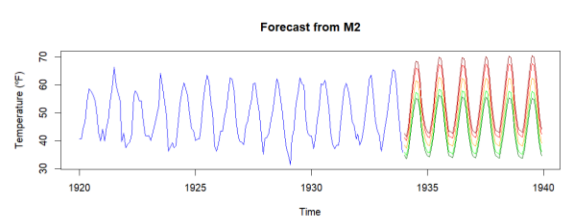


Figure 12: Forecasted and actual test data using the M1,M2,M3 and M4 models and 80% and 95% CI





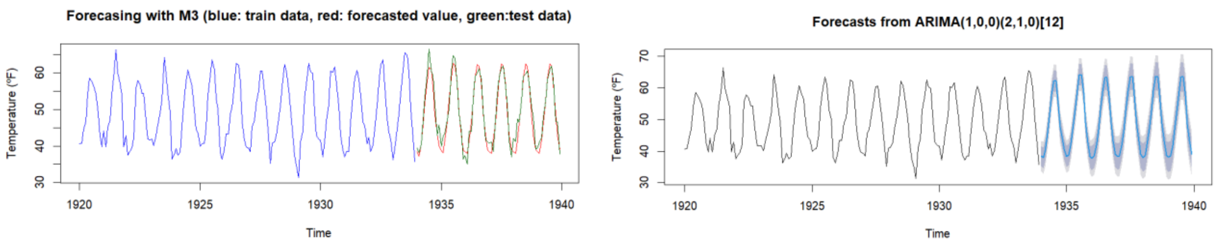
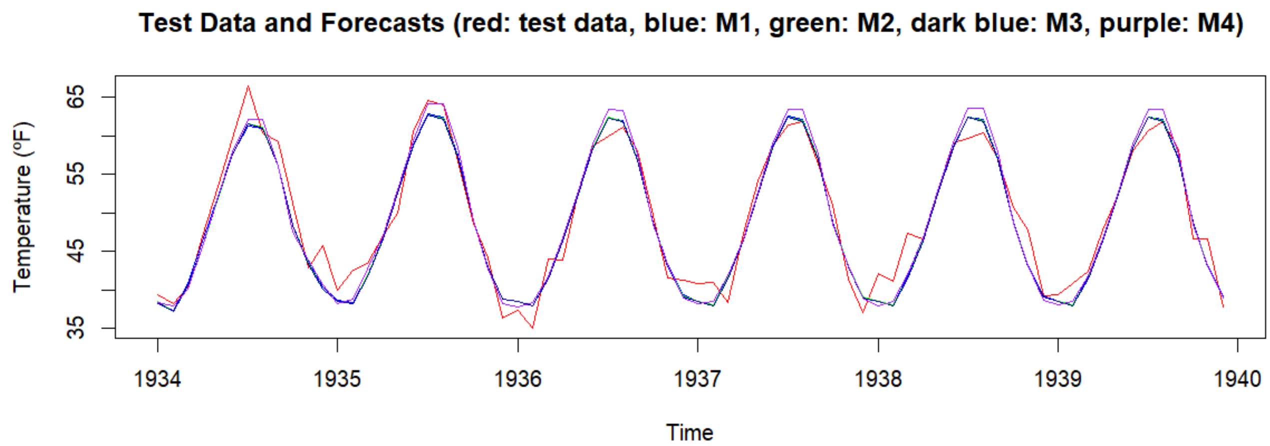


Figure 12: Forecasted and actual test data using the M1,M2,M3 and M4 models and 80% and 95% CI

Table 10: Test Data and forecasted values (2years out of 6 years) based on the models M1, M2, M3, and M4

Month	Test Data	Forecasts using M1	Forecasts using M2	Forecasts using M3	Forecasts using M4
Jan, 1934	39.4	38.30	38.46	38.40	38.42
Feb, 1934	38.2	37.31	37.27	37.26	37.93
Mar, 1934	40.4	40.89	40.48	40.54	40.11
Apr, 1934	46.9	46.28	45.67	45.67	45.62
May, 1934	53.4	52.17	51.90	51.92	52.07
Jun, 1934	59.6	57.66	57.69	57.66	57.91
Jul, 1934	66.5	61.27	61.49	61.50	62.12
Aug, 1934	60.4	60.99	61.05	60.98	62.16
Sep, 1934	59.2	55.98	56.02	56.02	56.09
Oct, 1934	51.2	48.34	48.35	48.39	47.49
Nov, 1934	42.8	43.59	43.42	43.33	43.95
Dec, 1934	45.8	40.37	40.18	40.09	40.64
Jan, 1935	40	38.63	38.59	38.58	38.29
Feb, 1935	42.6	38.36	38.28	38.22	38.86
Mar, 1935	43.5	41.91	41.80	41.74	42.69
Apr, 1935	47.1	46.75	46.66	46.61	47.10
May, 1935	50	52.69	52.62	52.58	52.96
Jun, 1935	60.5	58.78	58.65	58.57	59.31
Jul, 1935	64.6	62.84	62.70	62.62	64.06
Aug, 1935	64	62.41	62.22	62.09	64.13
Sep, 1935	56.8	57.40	57.25	57.16	58.46
Oct, 1935	48.6	48.95	48.92	48.91	49.00
Nov, 1935	44.2	43.05	43.01	42.97	42.78
Dec, 1935	36.4	38.79	38.77	38.76	38.19



11. Model accuracy comparison based on test dataset

Table 11: Model selection based on RMSE on test dataset

Model		MSE	RMSE	MAE	MPE	MAPE
M1:	SARIMA (2,0,2)(1,1,1)[12]	4.6162	2.1485	1.6922	1.1749	3.5999
M2:	Log Transform + SARIMA (2,0,1)(1,1,1)[12]	4.7000	2.1679	1.7094	1.3565	3.6330
M3:	BoxCox Transform+ SARIMA (2,0,1)(1,1,1)[12]	4.7418	2.1775	1.7123	1.4333	3.6415
M4:	SARIMA (1,0,0)(2,1,0)[12]	4.9608	2.2273	1.8015	0.7333	3.7730

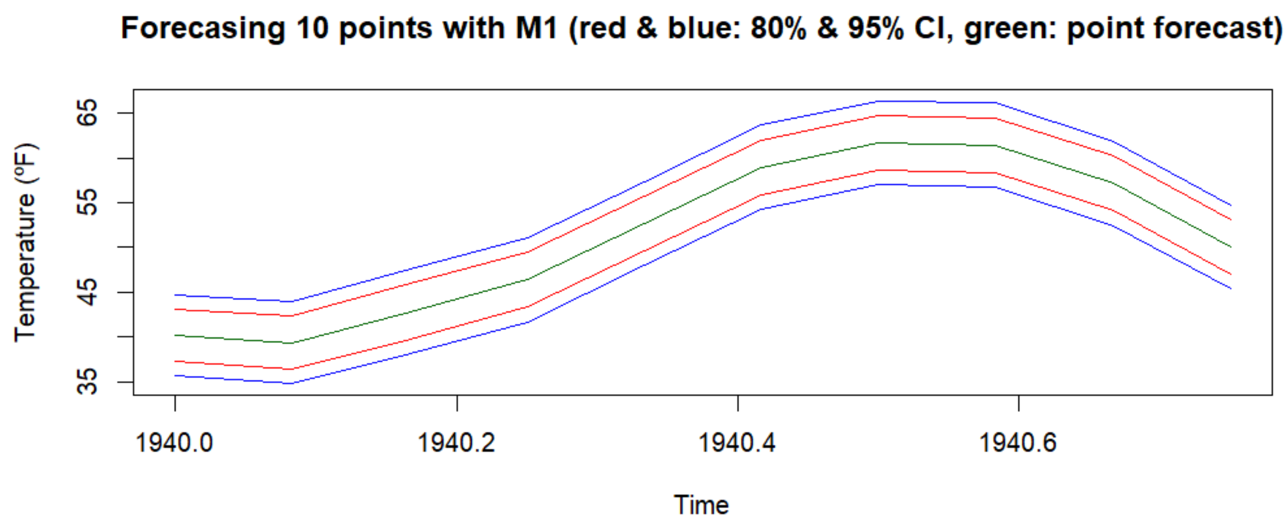
Decision: Based on the MSE, RMSE, MAE, MPE and MAPE values on the test dataset, model 1 (M1) is the best model among the four models because of its lowest values except MPE.

12. Final Model and Forecasting 10 points based on whole dataset:

10 points forecast based on the total dataset of nottem is given below:

Table 12: 10 points forecast on the nottem dataset by Model 1 (SARIMA (2,0,2)(1,1,1)[12])

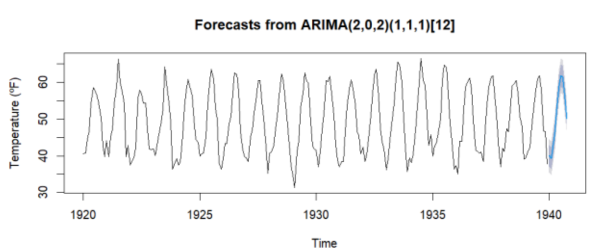
	Point Forecast (Mean Value of Forecast)	Low Value 80% CI	Low Value 95% CI	Hi Value 80% CI	Hi Value 95% CI
Jan 1940	40.13	37.19	35.64	43.08	44.63
Feb 1940	39.36	36.33	34.73	42.39	44.00
Mar 1940	42.86	39.79	38.17	45.92	47.55
Apr 1940	46.39	43.32	41.70	49.45	51.08
May 1940	52.66	49.59	47.97	55.73	57.35
Jun 1940	59.00	55.93	54.30	62.06	63.69
Jul 1940	61.74	58.67	57.04	64.81	66.43
Aug 1940	61.47	58.40	56.77	64.54	66.16
Sep 1940	57.16	54.09	52.47	60.23	61.85
Oct 1940	50.10	47.03	45.41	53.17	54.80

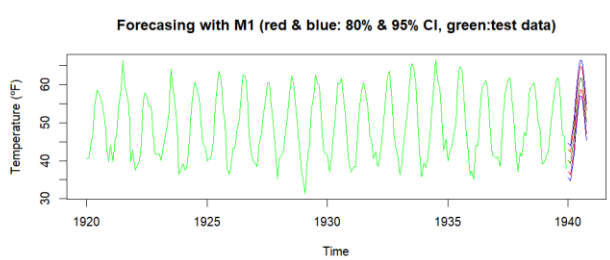



```
> M1
Series: nottem
ARIMA(2,0,2)(1,1,1)[12]

Coefficients:
         ar1      ar2      ma1     ma2     sar1     sma1
      0.5732  -0.2783  -0.3241  0.2923  -0.2857  -0.7378
s.e.  1.1576   0.6737   1.1658  0.3987   0.0823   0.0767

sigma^2 = 5.268:  log likelihood = -517.48
AIC=1048.96   AICc=1049.47   BIC=1072.97
> frcst_M1 <- forecast::forecast(M1, h=10)
> plot(frcst_M1)
> frcst_M1
> frcst_M1
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Jan 1940       40.13455 37.19302 43.07608 35.63587 44.63323
Feb 1940       39.36216 36.33079 42.39352 34.72608 43.99823
Mar 1940       42.85737 39.79115 45.92360 38.16799 47.54676
Apr 1940       46.38670 43.31988 49.45352 41.69641 51.07700
May 1940       52.65932 49.59108 55.72757 47.96684 57.35181
Jun 1940       58.99540 55.92635 62.06446 54.30168 63.68913
Jul 1940       61.73774 58.66865 64.80683 57.04397 66.43151
Aug 1940       61.46873 58.39961 64.53784 56.77492 66.16253
Sep 1940       57.15994 54.09081 60.22907 52.46611 61.85377
Oct 1940       50.10335 47.03421 53.17248 45.40951 54.79718
```





Annexure A (Code)

```
ID=06
#Average Monthly Temperatures at Nottingham, 1920–1939
library(tseries)
library(forecast)
library(itsmr)

N=length(nottem)

#outliers checking
tsoutliers(nottem)

#k=tsclean(nottem)
#k-nottem

decompose(nottem, type="additive")
plot(decompose(nottem, type="additive"))


### Time Plot, ACF plot, Seasonal Sub series plot
ts.plot(nottem, main="nottem dataset plot", xlab="Time (1920-1939)", ylab="Temperature (ºF)")
#plot(nottem)
acf(nottem, lag.max=36, main="ACF plot of nottem dataset",  xlab="Lag", ylab="corr coefficient")
pacf(nottem, lag.max=36,  main="PACF plot of nottem dataset",  xlab="Lag", ylab="partial corr coefficient")

ggsubseriesplot(nottem,main="Sub-Series plot of nottem dataset",  xlab="Months", ylab="Temperature (ºF)")




#ggplot(nottem)+geom_boxplot()
Col<- c('Jan','Feb','Mar','Apl','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec')
boxplot(as.vector(nottem)~cycle(nottem), data=nottem, names=Col, col="lightgreen", border="black",main = "Boxplot of nottem Dataset", xlab="Month", ylab = "Temperature (ºF)")
#boxplot(as.vector(nottem)~cycle(nottem), nottem, main = "Boxplot of nottem Dataset", ylab = "Temperature (ºF)")
```

```
# Convert the dataset to a data frame
#nottem_df <- data.frame(month = cycle(nottem), temperature = as.vector(nottem))

# Create a boxplot of temperature by month
#boxplot(temperature ~ month, data = nottem_df,
#       main = "Month-wise Boxplot of nottem Dataset",
#       xlab = "Month", ylab = "Temperature (ºF)")



## Training and test data split


n_train=ceiling(N*0.7)
n_train
n_test= N - n_train

trainData=ts(nottem[1:n_train], start=c(1920,1), frequency=12)
acf(trainData)
plot(trainData, main="training dataset", ylab="Temperature (ºF)")
length(trainData)
idx=n_train + 1

testData=ts(nottem[idx:N],start=c(1934,1), frequency =12)
ts.plot(trainData,testData, col=c('blue','red'),main="Training (blue) and Test (red) Dataset",
ylab="Temperature (ºF)")
#legend(5,5, legend=c("Train Dataset", "Test Dataset"),
#      fill = c("blue","red")
#)
length(testData)

####   stationarity check
library(tseries)
adf.test(trainData)     # p-value = 0.01 < 0.05, null hypo is rejected, hence stationary
kpss.test(trainData)    #p-value = 0.1 >0.05, hence null hypo not rejected, stationary


#seasonal differencing and stationarity checking of the difference data

ddnottem=diff(trainData, differences = 1)
plot(ddnottem)
acf(ddnottem)
```

```
pacf(ddnottem)
adf.test(ddnottem)
kpss.test(ddnottem)



dnottem=diff(trainData, lag  =12)
ts.plot(dnottem, main="Seasonal difference of nottem")

acf(dnottem, lag.max=36, main="ACF plot of seasonal difference of nottem")
pacf(dnottem, lag.max=36,main="PACF plot of seasonal difference of nottem")
adf.test(dnottem)
kpss.test(dnottem)



#### Model selection

acf(trainData)
pacf(trainData)


model <- function(data)
{ library(tseries)
  order=c(2,0,1)
  seasonal= list(order=c(2,1,2), period=12)
  M=Arima(data, order=order, seasonal=seasonal)
  return (M)

}


#M1=Arima(trainData, =Arima(trainData, order=c(3,0,1), seasonal = list(order=c(0,0,0), period=12))
M1 =Arima(trainData, order=c(2,0,2), seasonal = list(order=c(1,1,1), period=12))

M1
test(M1$residuals)
shapiro.test(M1$residuals)

library(forecast)
plot(forecast::forecast(M1, h=n_test))
```

```
#### Transform no 1

plot(trainData, main="Train Data")

Log_trainData=log(trainData)
plot(Log_trainData, main="Log Transformed Train Data")
adf.test(Log_trainData)     #Dickey-Fuller = -11.885,Lag order = 5, p-value = 0.01
kpss.test(Log_trainData)    #KPSS Level = 0.025818, lag parameter = 4, p-value = 0.1

sqrt_trainData=sqrt(trainData)
plot(sqrt_trainData, main="Square root Transformed Train Data")
adf.test(sqrt_trainData)     # Dickey-Fuller = -11.796, Lag order = 5, p-value = 0.01
kpss.test(sqrt_trainData)     #KPSS Level = 0.02829, Truncation lag parameter = 4, p-value = 0.1



cubicrt_trainData=(trainData)^(1/3)
plot(cubicrt_trainData, main="Cubic root Transformed Train Data")
adf.test(cubicrt_trainData)  #Dickey-Fuller = -11.824, Lag order = 5, p-value = 0.01
kpss.test(cubicrt_trainData)  # KPSS Level = 0.027419, Truncation lag parameter = 4, p-value = 0.1

ts.plot(Log_trainData, sqrt_trainData, cubicrt_trainData, col=c("red", "blue","darkgreen"),
main="Transformed Train Data", ylab="transformed temp")

##Log transform is selected

acf(Log_trainData, lag.max=36)
pacf(Log_trainData,  lag.max=36)

dLog_trainData=diff(Log_trainData, lag=12)
acf(dLog_trainData, lag.max=36)
pacf(dLog_trainData, lag.max=36)


M2=Arima(Log_trainData, order=c(2,0,1), seasonal = list(order=c(1,1,1), period=12))
#M2=model(Log_trainData)
M2  #sigma^2 estimated as 8.445:  log likelihood = -479.38,  aic = 970.77
  `#install.packages("itsmr")
test(M2$residuals) ##p-value = 2.431e-05
shapiro.test(M2$residuals)

AIC(M2)
```

```
#### Forecast
n=length(testData)
plot(forecast::forecast(M2, h=n))


### M3 after BoxCox transformation

lamda <- BoxCox.lambda(trainData)
lamda
Boxcox_trainData=BoxCox(trainData, lambda = lamda)
plot(Boxcox_trainData)
adf.test(Boxcox_trainData)
kpss.test(Boxcox_trainData)

acf(Boxcox_trainData)
pacf(Boxcox_trainData)


M3=Arima(Boxcox_trainData, order=c(2,0,1), seasonal = list(order=c(1,1,1), period=12))
#M3=model(Boxcox_trainData)
M3
test(M3$residuals)
shapiro.test(M3$residuals)

### M4

M4=auto.arima(trainData)
M4   #  Arima(1,0,2)(1,1,2)[12] with drift sigma^2 = 5.389:  log likelihood = -410.74  AIC=837.49
AICc=838.33   BIC=863.03
test(M4$residuals)
shapiro.test(M4$residuals)

#### Model evaluation based on training

sum((M1$fitted-trainData)^2)/150
M1$sigma2

M2_sigma2_corrected <- sum((exp(M2$fitted)-trainData)^2)/151

M3_sigma2_corrected <- sum(((lamda*M3$fitted+1)^(1/lamda)-trainData)^2)/151
```

```
sum((M4$fitted-trainData)^2)/153
M4$sigma2

M1_evaluation <- c(M1$aic, M1$aicc, M1$bic, M1$sigma2)
M2_evaluation <- c(M2$aic, M2$aicc, M2$bic, M3$sigma2, M2_sigma2_corrected)
M3_evaluation <- c(M3$aic, M3$aicc, M3$bic, M3$sigma2, M3_sigma2_corrected)
M4_evaluation <- c(M4$aic, M4$aicc, M4$bic, M4$sigma2)

M1_evaluation
M2_evaluation
M3_evaluation
M4_evaluation


#### Forecasting using the models
n=length(testData)


## M1
frcst_M1 <- forecast::forecast(M1, h=n)
plot(forecast::forecast(M1, h=n_test), xlab="Time", ylab="Temperature (ºF)")
ts.plot(frcst_M1$mean, testData, col=c("red","darkgreen"),main="Forecasing with M1 (red: forecasted
mean value, green:test data)", ylab="Temperature (ºF)")
frcst_M1
#ts.plot(frcst_M1$lower, frcst_M1$upper, testData, col =
c("red","darkred","blue","darkblue","darkgreen"),main="Forecasing with M1 (red & blue: CI, green:test
data)", ylab="Temperature (ºF)")
ts.plot(frcst_M1$lower, frcst_M1$upper, testData, col =
c("red","blue","red","blue","darkgreen"),main="Forecasing with M1 (red & blue: 80% & 95% CI,
green:test data)", ylab="Temperature (ºF)")


## M2
scaled_frcst_M2 <- forecast::forecast(M2, h=n)
scaled_frcst_M2
frcst_M2=exp(scaled_frcst_M2$mean)
frcst_M2
plot(frcst_M2)
ts.plot(frcst_M2, testData, col=c("red","darkgreen"),main="Forecasing with M2 (red: forecasted mean
value, green:test data)", ylab="Temperature (ºF)")
#ts.plot(exp(scaled_frcst_M2$lower), exp(scaled_frcst_M2$upper), testData, col =
c("red","darkred","blue","darkblue","darkgreen"))
```

```
ts.plot(exp(scaled_frcst_M2$lower), exp(scaled_frcst_M2$upper), testData, col =
c("red","blue","red","blue","darkgreen"),main="Forecasing with M2 (red: 80% CI, blue: 95% CI,
green:test data)", ylab="Temperature (ºF)")

frcst_M2Lo80=exp(scaled_frcst_M2$lower[,1])
frcst_M2Lo95=exp(scaled_frcst_M2$lower[,2])
frcst_M2Hi80=exp(scaled_frcst_M2$upper[,1])
frcst_M2Hi95=exp(scaled_frcst_M2$upper[,2])

#plot(frcst_M2)
#frcst_M2

ts.plot(trainData, frcst_M2,frcst_M2Lo80,frcst_M2Lo95, frcst_M2Hi80, frcst_M2Hi95,
col=c('blue','orange','green', 'darkgreen','red','darkred'),main="Forecast from M2", ylab="Temperature
(ºF)")

## M3
scaled_frcst_M3 <- forecast::forecast(M3, h=n_test)
frcst_M3=InvBoxCox(scaled_frcst_M3$mean, lambda=lamda)
frcst_M3

ts.plot(trainData, frcst_M3, testData, col=c("blue","red","darkgreen"),main="Forecasing with M3 (blue:
train data, red: forecasted value, green:test data)", ylab="Temperature (ºF)")
ts.plot(frcst_M3, testData, col=c("red","darkgreen"),main="Forecasing with M3 (red: forecasted mean
value, green:test data)", ylab="Temperature (ºF)")

#ts.plot(InvBoxCox(scaled_frcst_M3$lower, lambda=lamda), InvBoxCox(scaled_frcst_M3$upper,
lambda=lamda), testData, col = c("red","darkred","blue","darkblue","darkgreen"))
ts.plot(InvBoxCox(scaled_frcst_M3$lower, lambda=lamda), InvBoxCox(scaled_frcst_M3$upper,
lambda=lamda), testData, col = c("red","blue","red","blue","darkgreen"),main="Forecasing with M3 (red
& blue: 80% & 95% CI, green:test data)", ylab="Temperature (ºF)")

frcst_M3Lo80=InvBoxCox(scaled_frcst_M3$lower[,1],  lambda=lamda)
frcst_M3Lo95=InvBoxCox(scaled_frcst_M3$lower[,2],  lambda=lamda)
frcst_M3Hi80=InvBoxCox(scaled_frcst_M3$upper[,1],  lambda=lamda)
frcst_M3Hi95=InvBoxCox(scaled_frcst_M3$upper[,2],  lambda=lamda)

#plot(frcst_M2)
#frcst_M2

ts.plot(trainData, frcst_M3,frcst_M3Lo80,frcst_M3Lo95, frcst_M3Hi80, frcst_M3Hi95,
col=c('blue','orange','green', 'darkgreen','red','darkred'))
```

```
#legend(2,10, legend=c('blue','orange','green', 'darkgreen','red','darkred'))


##M4

frcst_M4 <- forecast::forecast(M4, h=n)
plot(frcst_M4)
frcst_M4
plot(forecast::forecast(M4, h=n_test), xlab="Time", ylab="Temperature (ºF)")

#ts.plot(frcst_M4$lower, frcst_M4$upper, testData, col =
c("red","darkred","blue","darkblue","darkgreen",)main="Forecasing with M4 (red: forecasted mean
value, green:test data)", ylab="Temperature (ºF)")
ts.plot(frcst_M4$mean, testData, col=c("red","darkgreen"),main="Forecasing with M4 (red: forecasted
mean value, green:test data)", ylab="Temperature (ºF)")

ts.plot(trainData, frcst_M4$mean, testData, col=c("blue","red","darkgreen"),main="Forecasing with M4
(blue: train data, red: forecasted value, green:test data)", ylab="Temperature (ºF)")

ts.plot(frcst_M4$lower, frcst_M4$upper, testData, col =
c("red","blue","red","blue","darkgreen"),main="Forecasing with M4 (red & blue: 80% & 95% CI,
green:test data)", ylab="Temperature (ºF)")



ts.plot(testData,frcst_M1$mean,frcst_M2,frcst_M3,frcst_M4$mean,col =
c("red","blue","green","darkblue","purple"),main="Test Data and Forecasts (red: test data, blue: M1,
green: M2, dark blue: M3, purple: M4)", ylab="Temperature (ºF)")


### Accuracy Measures

AccuracyMeasures <- function(predicted,testData){
  n=length(testData)
  error = testData - predicted
  ME    = sum(error)/n
  MSE   = sum(error^2)/n
  RMSE  = sqrt(MSE)
  MAE   = sum(abs(error))/n
  PE    = (error/testData)*100
  MPE   = sum (PE)/n
  MAPE  = sum(abs(PE))/n
```

```
  return (c(MSE, RMSE, MAE, MPE, MAPE))
  }

AM_M1 <- AccuracyMeasures(frcst_M1$mean,testData)
AM_M2 <- AccuracyMeasures(frcst_M2,testData)
AM_M3 <- AccuracyMeasures(frcst_M3,testData)
AM_M4 <- AccuracyMeasures(frcst_M4$mean,testData)

AM_M1
AM_M2
AM_M3
AM_M4

#best model is M1 model

## Forecasting 10 points ahead and plotting

#M4=auto.Arima(nottem)
#M4

M1 =Arima(nottem, order=c(2,0,2), seasonal = list(order=c(1,1,1), period=12))

M1

frcst_M1 <- forecast::forecast(M1, h=10)
plot(frcst_M1, xlab="Time", ylab="Temperature (ºF)")
frcst_M1
#last_6_years=ts(nottem[180:240], start=c(1935,1), frequency=12)
#ts.plot(frcst_M1$lower, frcst_M1$upper, frcst_M1$mean, nottem, col =
c("red","darkred","blue","darkblue","darkgreen","blue"))
ts.plot(frcst_M1$lower, frcst_M1$upper, frcst_M1$mean, nottem, col =
c("red","blue","red","blue","darkgreen","green"),main="Forecasing with M1 (red & blue: 80% & 95% CI,
green:test data)", ylab="Temperature (ºF)")
ts.plot(frcst_M1$lower, frcst_M1$upper, frcst_M1$mean, col =
c("red","blue","red","blue","darkgreen"),main="Forecasing 10 points with M1 (red & blue: 80% & 95%
CI, green: point forecast)", ylab="Temperature (ºF)")

testData
```