Statistics for Data Analytics
Submitted to: Prof. Tony Delaney

Prediction of Heating Degree Day(HDD) Index using Time Series Forecasting Models

Jaswinder Singh[1]*

[1] *x19219997@student.ncirl.ie*
***Affiliation**: *Cohort 'B'*, MSc in Data Analytics, National College of Ireland, Dublin, Ireland

Abstract

This part of the project aims to forecast the value of *Heating Degree Day* or *HDD* index using various time series models. The data used for the purpose of this project is taken from the Eurostat climate change database. The models are evaluated on the basis of their AIC and log likelihood values, Mean Error(ME), Root Mean Squared Error(RMSE), Mean Percentage Error(MPE) and Mean Absolute percentage Error(MAPE). The models applied were chosen on the basis of the nature of the time series(i.e whether it has trend or seasonality or both). Before modelling, stationarity of the data was checked using the *Augmented Dickey-Fuller test*. The models applied are: *ARIMA/SARIMA*, *Holt Winter's Seasonal Method*(using both additive and multiplicative decomposition), *Naive and Seasonal Naive*(on STL decomposed series and the original time series. The forecasts for all the models were plotted with the original time series.

1. INTRODUCTION

Since the early 20th century, enormous advancements have been made in every technological field, especially engineering. More houses and buildings are built in a year now than in a decade in the early 1900s, all thanks to these technological advancements. But with great advancements comes various challenges too. One of these challenges is to understand and evaluate the energy requirements for a structure(like a building). Quantifying the heat requirements of a building is particularly tricky since it depends on a large number of external factors like humidity, amount of solar radiation reaching the building, wind speed, no. of electrical appliances being used and most importantly how well is a particular building insulated. The Heating Degree Day or HDD index is a parameter designed to aid in calculating the heat requirements of a building by using an algorithm which takes into account most if not all of the above factors. It is a weather- based index which can be used to quantify the heating requirements of a building. The HDD is defined relative to a base temperature(temperature above which the building does not needs any heating). The base temperature for countries in EU is defined to be 15.5 °C or 59.9 °F. The unit of measure for HDD is °C (temperature sums).

The data taken from the Eurostat database contains the average monthly values for HDD index for Dublin city. To forecast the future values, we will use several time series forecasting model to our aid and evaluate these models individually. We will also try to analyse and visualize the seasonality and trends in our data which will help us in better deciding the forecasting model. The analysis and modelling is done using R programming language.

2. TIME SERIES FORECASTING- A BRIEF INTRODUCTION

In simple terms, a time series can be defined as the set or a sequence of measurements of a variable measured at regular intervals of time(for e.g a day, week, month, year, etc). A time series is an example of a longitudinal data(since it contains the measurements of a single variable over successive period of time). The first step in identifying the underlying pattern in a time series is to plot it. Now, at first look a time series plot may appear to provide us with little or no information about our data. But it contains more information than it seems. To see that information, we have observe in our time series for patterns. In general, a time series has the following components:

- *Trend*: Trend is the overall general behaviour of a time series. The trend pattern tells us how the series is behaving generally over time.
- *Seasonality*: A seasonal pattern occurs when the time series is affected by the seasonal factors like a particular month(s) of the year or a particular day(s) of the week. The seasonal pattern has a fixed frequency.
- *Cyclic*: A cyclic pattern comes into play when there are rises an falls in the data which are not of fixed frequency.
- *Random Fluctuations*: Rapid and irregular changes in the data with no fixed frequency whatsoever.

Figure 1 illustrates these components more clearly.

Before applying the forecasting models to a time series, one has to perform the stationarity test and apply some decomposition methods(to gain more insights as to which models should be applied). The stationarity test is used to ensure that the statistical properties of a time series is independent of time. In other words, a stationary time series should have a time independent or constant mean and variance. The stationarity

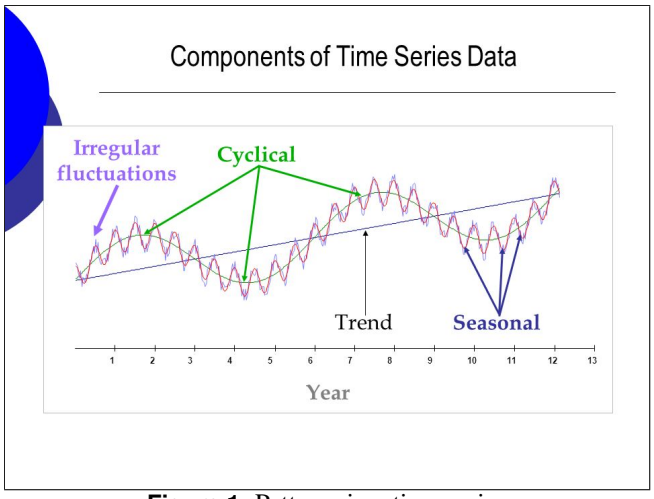

Figure 1. Patterns in a time series

can be checked by various methods which include both manual and software tests. To see if a series is stationary, we have to look at it's ACF(Auto- Correlation Function) plot. If the ACF plot is decaying rapidly as the lags increases, the series is non stationary. On the other hand, if it decays slowly, the series could be stationary. But this method is not reliable for obvious reasons. Therefore we employ the software tests to check the stationarity of the series. We will use the following three tests to conclude if our series is stationary or not.

- **Augmented Dickey-Fuller Test(ADF) test**: It is a unit root* test for stationarity. We have to specify the lag order in order to correctly check for the stationarity.
  - *Null hypothesis*: Non stationarity
  - *Alternative hypothesis*: Stationarity

  If we want to reject the null hypothesis, we want the p -value to be less than 0.05.

- **Phillips Perron test**: It is also a unit root test. It checks the series for the presence of a unit root.
  - *Null hypothesis*: Unit root is present(Non stationarity).
  - *Alternative hypothesis*: Unit root is not present (Stationarity).

  If we want to reject the null hypothesis, we want the p -value to be less than 0.05.

- **KPSS test**: It is a stationarity test which determines if the series is stationary around mean or if it is non stationary due to presence of a unit root.
  - *Null Hypothesis*: Series is stationary.
  - *Alternative Hypothesis*: Series is non-stationary.

  Therefore, if a series is stationary, the p-value for this test should be greater than 0.05. In other words, we want to be failed to reject the null hypothesis.

After the stationarity test, the next step is the smoothing of the time series to observe the trends and seasonality more clearly. We can smooth a time series by using a simple moving average method. The moving average method essentially reduces the no. of observations by replacing each observation with the mean of the specified no. of observations before and after it. For e.g. a centered moving average replaces each data point with the mean of that point, the one occurring before and after it.

The next step is the decomposition of time series into its individual components i.e *Trend*, *Seasonality*, etc There are several decomposition procedures available. The best practise to choose the best method is to plot the data along with the decomposed series and then decide which method best explains the trend in our time series. The main decomposition methods are:

- *Seasonal Decomposition*: Seasonal decomposition method can be either *additive* or *multiplicative*. In an additive model, the components sum give the values of the time series. While in the *multiplicative model*, the components multiplication give the resultant value of the time series. The additive model is used when the seasonal fluctuations do not depend on time and the multiplicative model is used when the seasonal fluctuations amplify with time.

$$\text{Additive Model: } Y_t = \text{Trend}_t + \text{Seasonal}_t + \text{Irregular}_t$$

$$\text{Multiplicative Model: } Y_t = \text{Trend}_t * \text{Seasonal}_t^* \text{Irregular}_t$$

- *X11 Decomposition* : This method basically overcomes the shortcomings of the classical decomposition. In particular, it handles the seasonal component very well by allowing for the very slow changes in it. It has also algorithms to handle the holiday effects and other unknown effects of the variable.

- *STL Decomposition* : The STL stands for Seasonal and Trend decomposition using Loess(Locally estimated scatterplot smoothing). It is more robust method as compared to the classical decomposition.

The final step is the application of the model to forecast the future values of a time series. The major methods used for forecasting are:

- *Random Walk Method*: In random walk, the variable takes an independent random step in either direction(up or down). The forecasts from random walk model are equal to the last observed value of the variable, since the future movements of such a variable is unpredictable. This method is used widely for non stationary series.

- *Naive Method*: For naive forecasts, we set the values of all the forecasts to be the value of the last observation in the the series. This method is not particularly useful in most of the cases but works well in many financial time series. The forecasts obtained are also called as the random walk forecasts since this method works best for the random walk data.

- *Seasonal Naive Method*: This method is particularly useful for the seasonal data. The underlying principle is same as the naive except that the value of the forecasts is set equal to the last value from the same season. For example if the data is monthly, the forecast value for the month of February will be last observed value for the February month.

- *Holt Winter's Seasonal Method*: The Holt Winter seasonal method is a type of an exponential smoothing model called triple exponential smoothing model since it fits a time series with level, trend and seasonal components. It uses three smoothing equations- one for each level(slope), trend and the seasonal components. The corresponding smoothing parameters are α, β, and γ. There are two variations in this method: *Additive*(preferred for constant seasonal variations) and *Multiplicative*(preferred for non constant seasonal variations).

- *ARIMA/SARIMA*: The ARIMA model is the most widely used method for forecasting of a time series. ARIMA models aim to describe the auto correlations in the data. ARIMA(Auto Regressive Integrated Moving Average) is the general model which combines the differencing with the auto regressive(AR) and moving average(MA) models. Lags of the stationary series in the equation are called AR(auto regressive) terms, forecast errors are called MA(Moving average) terms and the time series which need to differenced is called the integrated version of a stationary series. Random walk, AR models and exponential smoothing models are the special cases of ARIMA models. ARIMA models have three components:

  1. *AR(p)*: AR is the auto regressive part represented by the letter p. It's value is decided from the ACF plot.
  2. *d* : The order of differencing required to make the data stationary, represented by letter d.
  3. *MA(q)* : MA is the moving average part represented by letter q. Its value is decided from the PACF plot.

  The seasonal ARIMA or SARIMA also take into account the seasonality of the data and the seasonal indices are represented by (P,D,Q). the *auto.arima* function in R facilitates the automatic selection of the parameters (p,q,d)(P,D,Q) and outputs the model with best(lowest) AIC(Akaike's Information Criteria) value.

The *Ljung-Box* test is used to evaluate the ARIMA model. The null hypothesis(H_0) is that the model doesn't shows lack of fit. Therefore to be failed to reject the null hypothesis(which is required), the p- value should not be significant(i.e should be more than 0.05). The residuals and the Normal Q-Q plots are also analysed for judging the fit of the ARIMA model.

3. DATA SOURCE AND PREPARATION

The data used is taken from the Eurostat climate change database. The data was imported into R studio for cleaning and preparation. The filters were selected on the database webpage to allow for the selection of monthly data and year range of the variable. After importing the data in a csv file, a time series object was constructed for further analysis in R.

4. ANALYSIS, MODELLING AND FORECASTING

4.1 Visualizing the time series

For visualising the time series, we first plot the time series using the *autoplot()* function as shown in Figure 2. As we can see clearly, there is strong seasonality in our data. The trend is more or less constant over time with some random fluctuations as well. Now to see the seasonality more clearly, we can look at the seasonal plot of the time series(Refer to Figure 3)

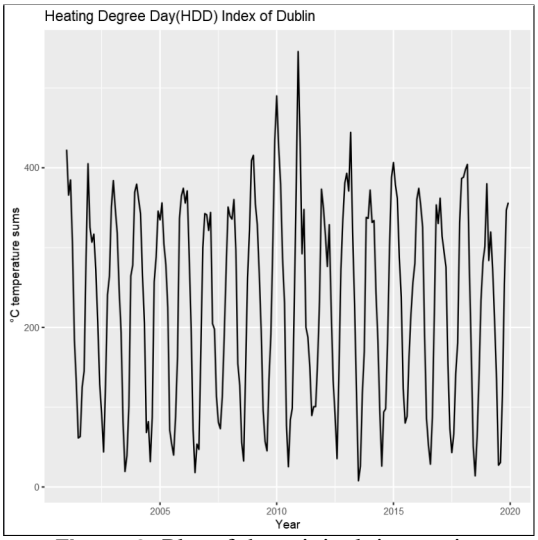

Figure 2. Plot of the original time series

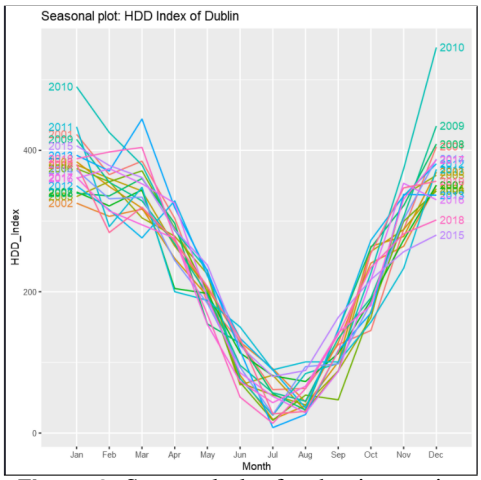

Figure 3. Seasonal plot for the time series

4.2 Stationarity Test for the time series

As discussed in Section 2, we know test the time series for the stationarity. We mentioned 3 tests for stationarity in Section 2. The figures 4 to 6 shows the outputs of these test done in R. As we can see, all the test suggest that our series is stationary.

```
Warning message in adf.test(heating_ts, k = 1):
"p-value smaller than printed p-value"

        Augmented Dickey-Fuller Test

data:  heating_ts
Dickey-Fuller = -9.6038, Lag order = 1, p-value = 0.01
alternative hypothesis: stationary
```

Figure 4. Augmented Dickey Fuller test for stationarity

```
Warning message in pp.test(heating_ts):
"p-value smaller than printed p-value"

        Phillips-Perron Unit Root Test

data:  heating_ts
Dickey-Fuller Z(alpha) = -80.135, Truncation lag parameter = 4, p-value
= 0.01
alternative hypothesis: stationary
```

Figure 5. Phillips Perron test for stationarity

```
Warning message in kpss.test(heating_ts):
"p-value greater than printed p-value"

        KPSS Test for Level Stationarity

data:  heating_ts
KPSS Level = 0.012799, Truncation lag parameter = 4, p-value = 0.1
```

Figure 6. KPSS test for stationarity

4.3 Decomposition of time series

In this section we decompose our time series into its various components using decomposition methods discussed in Section 2. As we see no increase in seasonal fluctuations with time in our data, we have plotted the classical decomposition(additive) as shown in Figure 7.

We have also plotted the X11 and the STL decomposition as depicted by figures 8 and 9 respectively. To understand the trend and seasonality more clearly, we have plotted the original time series with he seasonally adjusted and trend components.(Refer to figure 10)

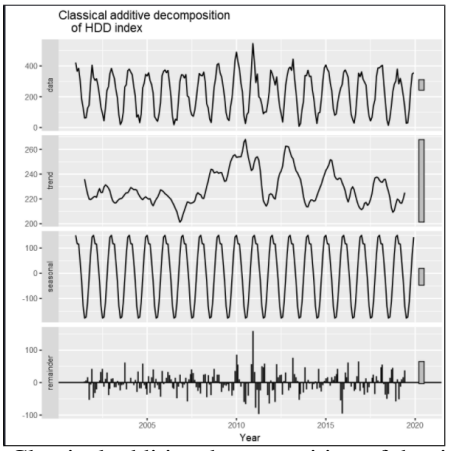

Figure 7. Classical additive decomposition of the time series

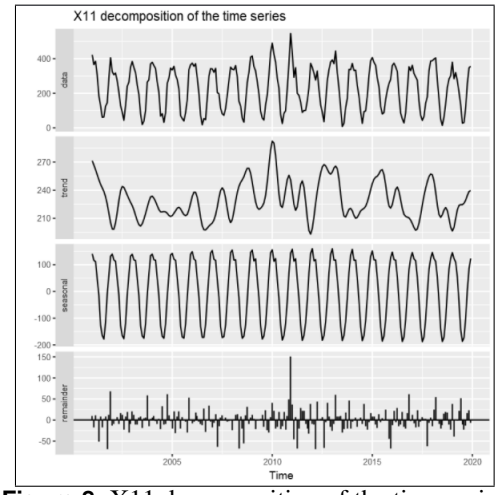

Figure 8. X11 decomposition of the time series

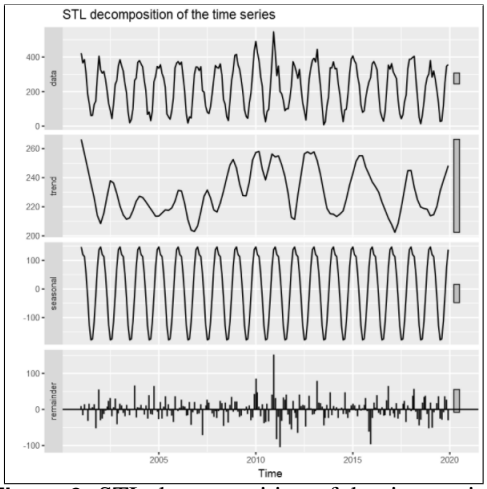

Figure 9. STL decomposition of the time series

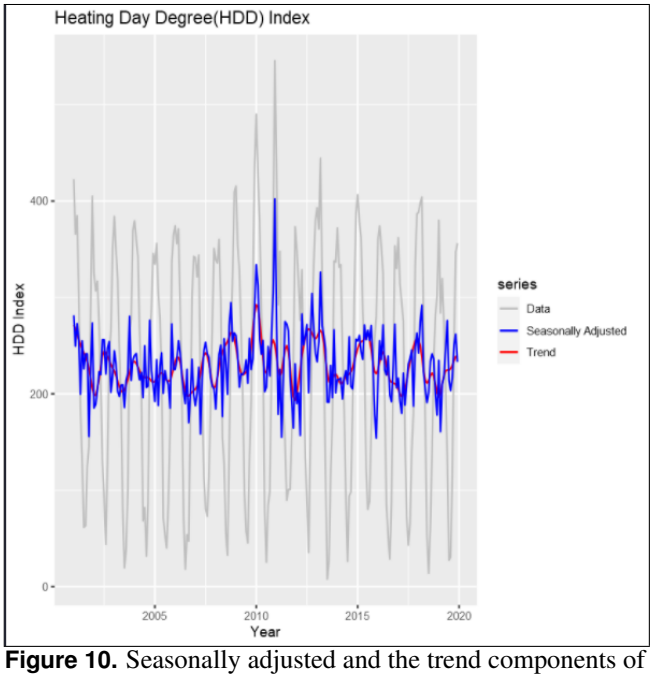

Figure 10. Seasonally adjusted and the trend components of the time series

4.4 Modelling and Forecasting

The only thing left now is to apply the different models to our time series and predict the future values. The models and their outputs are explained clearly below:

1. **Model 1-** *Naive and Seasonal Naive*: As explained in Section 2, the naive method sets the forecast value to be equal to the last observed value and the seasonal naive sets the forecast value equal to the last value observed in the same season. To get more insights into the forecasts, the data was divided into training and test data. The test data was then plotted to compare with the forecasted values.The forecasts obtained are shown in Figure 11

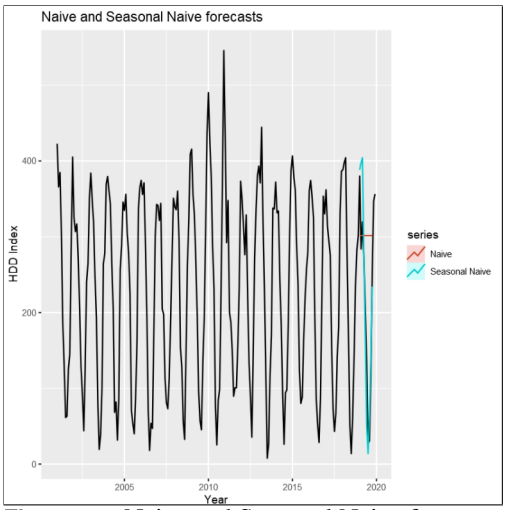

Figure 11. Naive and Seasonal Naive forecasts

2. **Model 2-** *Holt Winter's Seasonal Method:* Forecast from Holt Winter's method on both additive and multiplicative methods is shown in figure 12.
(*Note: The data plotted here starts from the year 2007 to show the forecasts more clearly*)

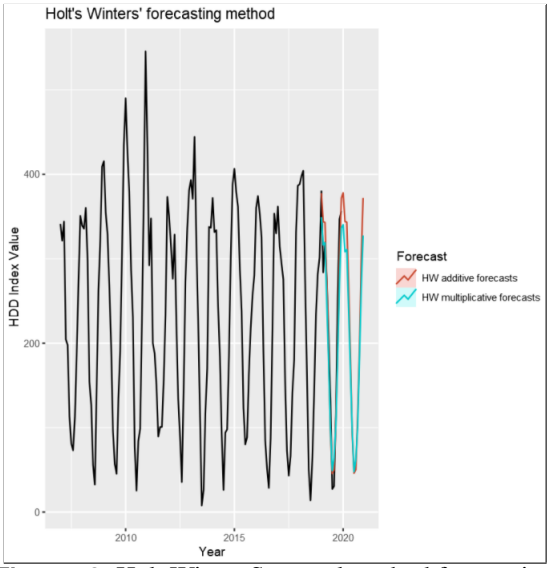

Figure 12. Holt Winter Seasonal method forecasting

3. **Model 3-** *ARIMA Model:* Forecasts from ARIMA Model are shown in figure 13. The values of (p,d,q) and (P,D,Q) as suggested by the auto arima function are (1,0,0) and (1,1,0). The values of p and q can also be decided by looking at the ACF(refer to figure 14) and PACF plots. respectively(Refer to figure 16)

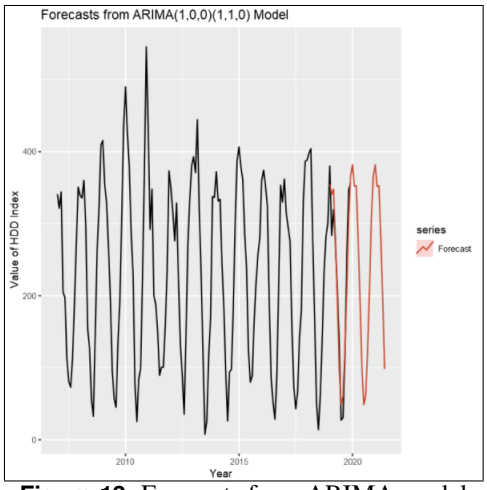

Figure 13. Forecasts from ARIMA model

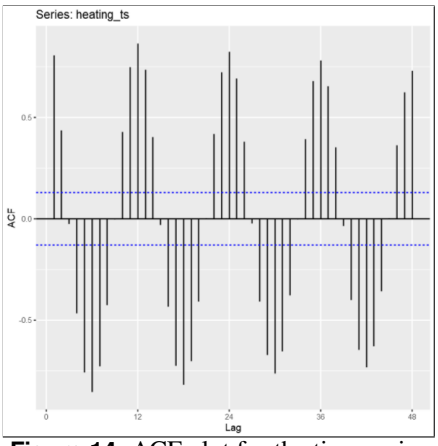

Figure 14. ACF plot for the time series

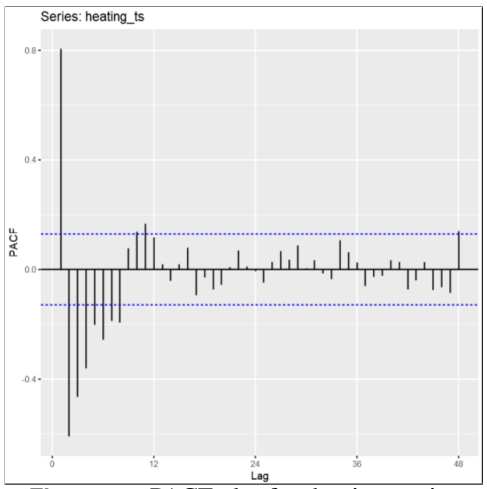

Figure 15. PACF plot for the time series

5. Evaluation of Models

Now we will evaluate each of the models by looking at various parameters.

1. *Naive and Seasonal Naive:* The accuracy parameters for Naive and Seasonal Naive are shown in figures 16 and 17 respectively. As we can see, Seasonal Naive method is better as it has less values for MPE and RMSE as compared to Naive method.

A matrix: 1 × 7 of type dbl

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	-0.5637209	76.1004	62.41833	-21.59108	51.08892	1.561551	0.4854461

Figure 16. Accuracy parameters for Naive method

A matrix: 1 × 7 of type dbl

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	-0.7863235	51.53242	39.97201	-12.87472	31.97836	1	0.381247

Figure 17. Accuracy parameters for Naive method

2. *Holt Winter Seasonal:* The accuracies for both the additive and multiplicative variations of the method are shown in figures 18 and 19 respectively. As we can see both the methods have almost same values for RMSE, while the MAE and MPE for multiplicative method are slightly higher than the additive. So to select as our final model, we would choose the additive variation of Holt Winters' method.

A matrix: 1 × 7 of type dbl

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	-0.08466687	38.46826	29.96959	-12.16935	26.81584	0.6537588	0.3697892

Figure 18. Accuracy parameters for Holt Winter additive method

A matrix: 1 × 7 of type dbl

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	-0.8384935	38.31467	30.4937	-13.06674	25.61046	0.6651917	0.4072229

Figure 19. Accuracy parameters for Holt Winter multiplicative method

3. *ARIMA:* We first test the ARIMA model for Ljung Box test(refer to figure). As we can see that the p value is non significant, hence the model is a good fit to our data. Next we look at the Normal Q-Q plot of the data(refer to figure). As we can see, the residuals can well be considered to be normally distributed. Next we look at the accuracy parameters of the model(refer too figure). We can safely say that ARIMA model fits the better than the other two models because of its low values for RMSE and MPE.

```
        Box-Ljung test

data:  model_arima$residuals
X-squared = 0.20658, df = 1, p-value = 0.6495
```

Figure 20. Ljung box test for ARIMA model

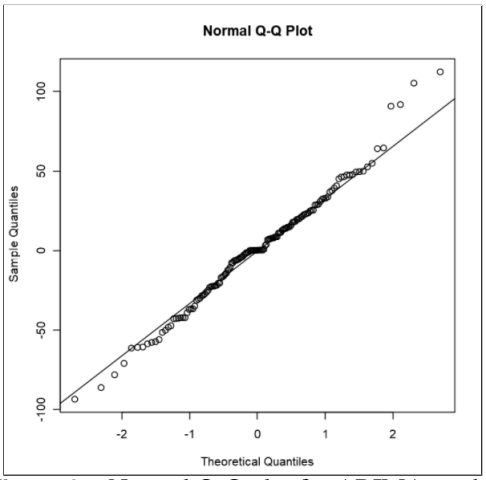

Figure 21. Normal Q-Q plot for ARIMA model

```
accuracy(model_arima)
```

A matrix: 1 × 7 of type dbl

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	0.4734442	36.29791	27.65096	-10.26378	22.86066	0.60318	0.03748495

Figure 22. Accuracy paramters for ARIMA model

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

Coefficients:
         ar1     sma1
      0.4182  -0.8837
s.e.  0.0801   0.1163

sigma^2 estimated as 1459:  log likelihood=-676.05
AIC=1358.09   AICc=1358.28   BIC=1366.74
```

Figure 23. ARIMA main output

```
        Ljung-Box test

data:  Residuals from ARIMA(1,0,0)(1,1,0)[12] with drift
Q* = 65.201, df = 21, p-value = 2.022e-06

Model df: 3.   Total lags used: 24
```

Figure 24. Ljung Box test for residuals

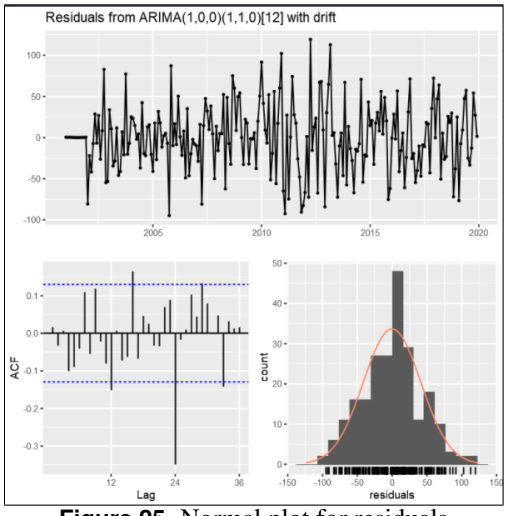

Figure 25. Normal plot for residuals