# Electricity Production forecasting using Time Series Analytics

**Dheeraj Kashyap Varanasi – wt3631**
**Siddhharth Reddy – fo4203**
**Mohan Srinivas Vendra – us2205**
**Rajashekar Reddy Ayluri – nv7709**
**Sai Rohith Reddy Nagella – tx8218**

## Summary

In this project, we utilized the Electricity Production dataset obtained from Kaggle, which contains yearly data from 1985 to 2017. The goal was to predict electricity production using time series techniques. To begin, we visually analyzed the data using a historical plot and observed a consistent upward trend from 1985 to 2017. Additionally, we discovered that the data is strongly correlated, as evidenced by the significant autocorrelation coefficients for all 12 lags. We performed basic predictability tests using both the normal approach and the first lag differencing approach and found that the data was predictable.

We then applied several time series techniques to forecast the data, including Two-Level Forecasting (Quadratic trend + seasonality and MA trailing method), Holt-Winter's model with automated selection of error/level, trend, and seasonality, and ARIMA (Autoregressive Integrated Moving Average). We calculated the accuracy measures for each method and determined that ARIMA was the most effective approach for forecasting the data. We compared the accuracy measures for ARIMA with those of the seasonal naïve and naïve forecasts, ultimately selecting ARIMA as the most suitable method for forecasting.

To compare the accuracy of the different methods, we utilized four measures: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Error Percentage (MAPE). We primarily evaluated the models based on their MAPE and RMSE comparisons.


## Introduction

Modern society is highly reliant on electricity production, which powers everything from homes and businesses to transportation and communication infrastructure. In the United States, electricity is generated using a variety of energy sources, including fossil fuels, nuclear energy, and renewable energy. According to the U.S. Energy Information Administration (EIA), the country produced 4.01 trillion kilowatt-hours (kWh) of energy in 2020, with most of it coming from fossil fuels.

The dataset used for this project contains electricity production data from 1985 to 2017, which was obtained from the Kaggle website. Generally, electricity production has increased over time, with production defined as the kilowatt-hours (kWh) generated in a day, month, or year.

Time series analysis is a valuable tool for forecasting various types of outcome variables, such as sales or stock volume, using historical time-based data. This is also known as prescriptive analysis. The focus of this project is to forecast electricity production using various time series analysis techniques applied to the historical data collected from Kaggle. This could provide a useful solution for countries to manage their electricity production based on available funds and plan accordingly.

**8- steps of forecasting**

**Step -1: Define Goal:**

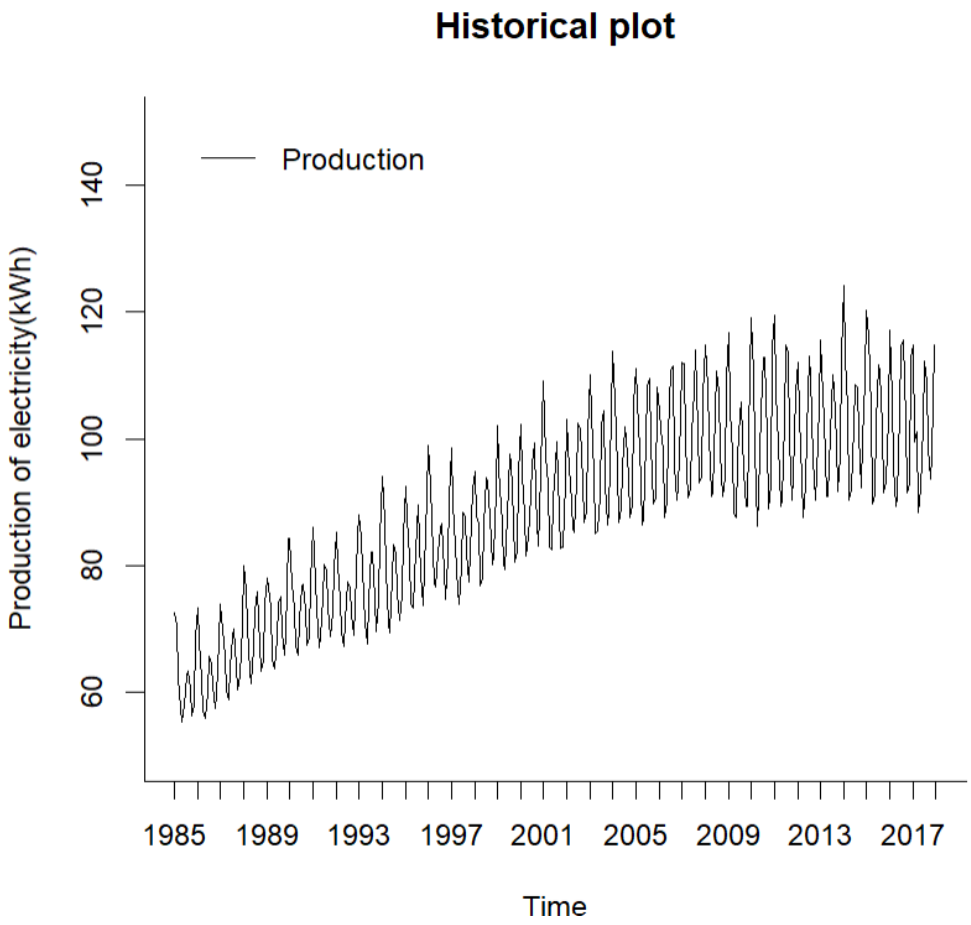
The main objective of this research is to predict the power output for the upcoming fiscal year, specifically for the year 2018, using historical data from the dataset. The goal is to develop a prediction model that accurately forecasts data for the next 12 months, while taking into account seasonality and trend patterns. The preferred model is the one with the highest level of accuracy, as it will aid in future data forecasting. To implement the strategies, R was chosen as the programming language for this research.

**Step -2: Get data:**

The dataset used in our project was obtained from Kaggle.com, an online community for sharing and discovering datasets. It covers a period from January 1985 to December 2017 and consists of 396 data points. Our goal is to use this data to forecast the electricity production for the next 12 months, from January 2018 to December 2018.

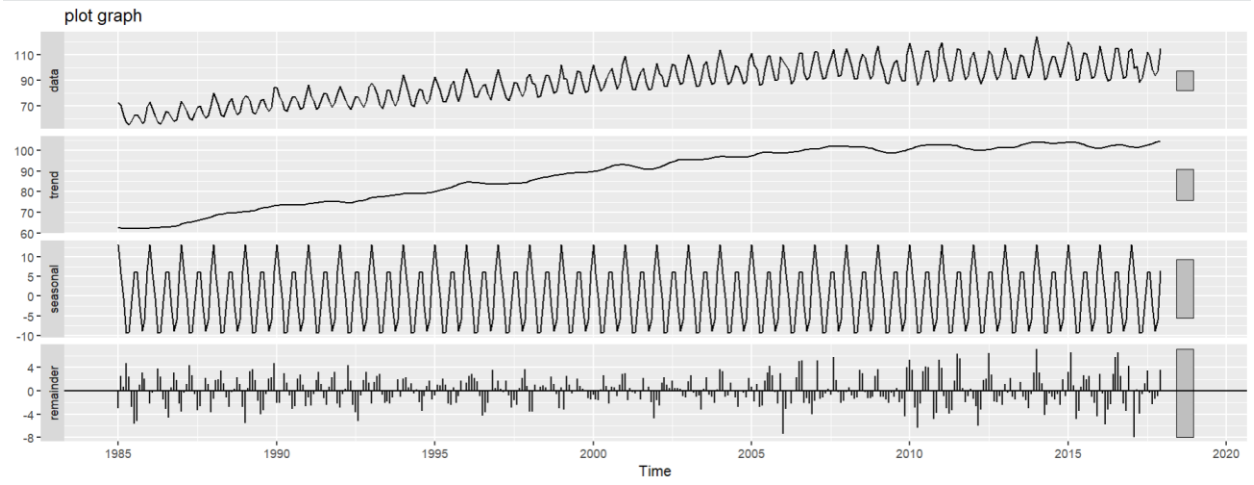
**Step -3: Explore and visualize series:**

The historical plot of the data indicates an upward trend, with no noticeable decrease in production over the years. However, there is observable seasonality in the data, with repeating patterns, and the peak production value occurred in 2014.



**Trend, seasonality and level plot :**

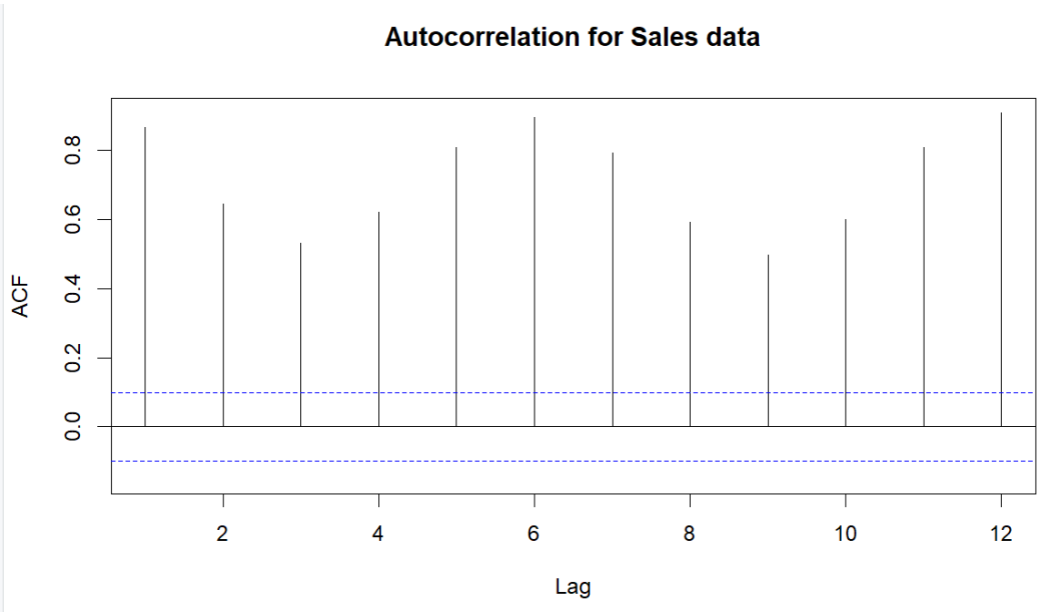
From the below plot we can understand that there is an upwards trend in the data hence, there is a trend component. Which can be true from the below stl() plot aswell.




From the seasonal section we can infer that there is an additive seasonality and there is also a level component in the data. The noise is present in all the data but it is quite high in the end of the data.

**Autocorrelation plot:**

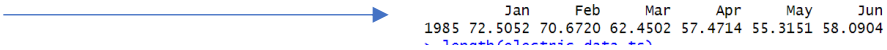
The plot below displays the correlation coefficients for up to 12 lags. It is observed that the autocorrelation coefficients are significantly high for all the lags. Lag 1 and lag 12 have the highest autocorrelation coefficients, both with a value of approximately 0.9, indicating the presence of trend and seasonality components in the data.




**Step -4: Data Preprocessing:**

In the data we have two columns, one is date column and other is value one which contains production of electricity. We have used ts() function to convert the whole data into time series data.

| DATE | Value |
|---|---|
| 1/1/1985 | 72.5052 |
| 2/1/1985 | 70.672 |
| 3/1/1985 | 62.4502 |
| 4/1/1985 | 57.4714 |
| 5/1/1985 | 55.3151 |




396 observations are contained in the time series data electric.data.ts file.

**Checking the predictability of data:**

We need to check whether the data is predictable or a random walk for this we followed two approaches first is the general approach with the null hypothesis and Ar(1).

➔ **Approach – ARIMA – AR(1):**

ar1 <- 0.9993
s.e. <- 0.0010
null_mean <- 1
alpha <- 0.05
z.stat <- (ar1-null_mean)/s.e.
z.stat p.value <- pnorm(z.stat)

p.value if (p.value < alpha){
"Reject null hypothesis"
} else {
"Accept null hypothesis"
}

```
> ar1 <- 0.8734
> s.e. <- 0.0245
> null_mean <- 1
> alpha <- 0.05
> z.stat <- (ar1-null_mean)/s.e.
> z.stat
[1] -5.167347
> p.value <- pnorm(z.stat)
> p.value
[1] 1.187201e-07
> if (p.value<alpha) {
+   "Reject null hypothesis"
+ } else {
+   "Accept null hypothesis"
+ }
[1] "Reject null hypothesis"
```

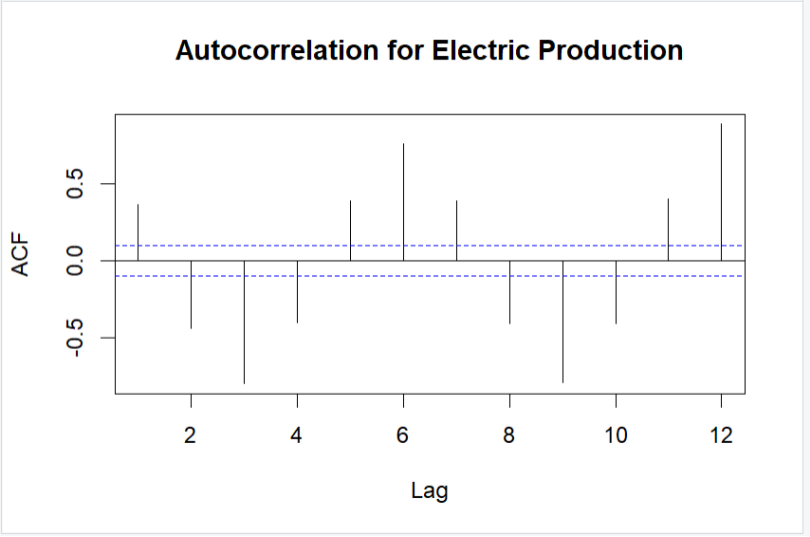
Based on this approach we got p-vale < 0.05 and this results in

**"Reject null hypothesis."**

This means that the data is predictable. In addition, we can go to another approach which is first lag differencing approach.

➔ **Approach – 2 ACF with differencing lag 1:**

Below graph represents the auto correlation of the first lag differencing where all the autocorrelation coefficients are significant and are above the horizontal dashed lines, hence the graph shows that the data is predictable.



**Step -5: Partition series:**

We get 396 data points in total, and when we divide the time series data into two parts, we receive 308.6 as 80% of the training data and 79.2 as 20% of the validation data out of the 396 data points we have. As it is preferred to divide the yearly data into the proper ratio of years , we got the training data of 324 records(26 years) for the training period and 72 records(6 years) for the validation period.

**Training data:** train.ts

| | |
|---|---|
| train.ts | Time-Series [1:324] from 1985 to 2012: |

**Validation data:** valid.ts

| | |
|---|---|
| valid.ts | Time-Series [1:72] from 2012 to 2018: |

**Step 6 & 7 : Apply forecasting and comparing performance:**

1) **Two level forecasts (quadratic trend and seasonality with moving average for residuals):**

The trailing MA may be used to data with trend or seasonality using two-level forecasting, which combines two forecasting models.

**Level1**: A quadratic trend in regression with seasonality. Additionally, it may be used to remove seasonality and/or patterns from historical data.

Finding residuals (errors): The differences between the forecast using the regression trend with seasonality and actual data points for different time periods.

**Level2**: The residuals (errors) of the regression can be predicted using trailing MA.

We combine these Regression with quadratic trend with seasonality and trailing MA to bring the overall forecasts of the data.

To improve the linear trend and seasonality regression model and anticipate model residuals, a trailing moving average was used. After combining these components, a two-level model and a model over all the data were created and used to anticipate the upcoming 12 months.

**Model trained over training data:**

```
Call:
tslm(formula = train.ts ~ trend + I(trend^2) + season)

Residuals:
    Min      1Q  Median      3Q     Max
-7.6608 -1.8429 -0.1331  1.9076  8.3580

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.279e+01  6.910e-01 105.336  < 2e-16 ***
trend        2.033e-01  6.643e-03  30.599  < 2e-16 ***
I(trend^2)  -2.095e-04  1.980e-05 -10.585  < 2e-16 ***
season2     -6.746e+00  7.588e-01  -8.891  < 2e-16 ***
season3     -1.285e+01  7.588e-01 -16.932  < 2e-16 ***
season4     -2.126e+01  7.588e-01 -28.018  < 2e-16 ***
season5     -2.155e+01  7.588e-01 -28.404  < 2e-16 ***
season6     -1.423e+01  7.588e-01 -18.759  < 2e-16 ***
season7     -7.297e+00  7.588e-01  -9.617  < 2e-16 ***
season8     -6.832e+00  7.589e-01  -9.002  < 2e-16 ***
season9     -1.530e+01  7.589e-01 -20.165  < 2e-16 ***
season10    -2.131e+01  7.589e-01 -28.080  < 2e-16 ***
season11    -1.873e+01  7.589e-01 -24.682  < 2e-16 ***
season12    -6.557e+00  7.590e-01  -8.639 3.06e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.788 on 310 degrees of freedom
Multiple R-squared:  0.9655,    Adjusted R-squared:  0.9641
F-statistic: 667.6 on 13 and 310 DF,  p-value: < 2.2e-16
```

Looking at Adjusted R – Squared value 0.9641 (96%), we can say that the model is a good fit. Considering overall p – value is less than 0.05 hence the model is significant, and all the coefficients are also significant. May be applied for the forecasting.

**Model Equation: (Regression model with quadratic trend and seasonality)**

$$Y_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 D_2 + \beta_4 D_3 + \beta_5 D_4 + \beta_6 D_5 + \beta_7 D_6 + \beta_8 D_7 + \cdots + \beta_{12} D_{11} + \varepsilon$$

**For our case:**

$$Y_t = 72.97 + 0.2033t - 0.000295t^2 - 6.746D_2 - 0.1285D_3 - 0.212D_4 - 0.2155D_5 - 0.1423D_6 - 7.297D_7 + \cdots - 6.577\, D_{12} + \varepsilon$$

***where, t = 1,2,3……n (n=number of time periods/trends)***

***D2 = binary (1,0), it is 1 if Feb and 0 if otherwise***

***D3 = binary (1,0), it is 1 if Mar and 0 if otherwise***

.

.

***D12 = binary (1,0), it is 1 if Dec and 0 if otherwise***

***If D2 , D3 ,…, D12 are 0 then it is Jan***

**Selecting K (window width):**

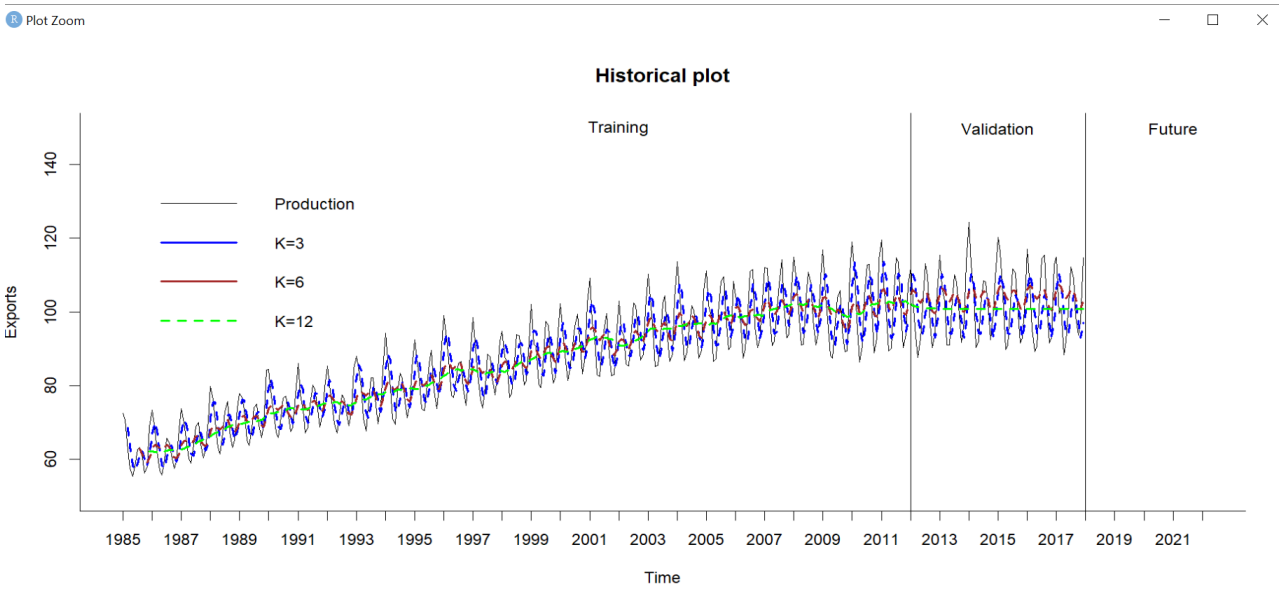
We want to apply the trailing Moving Average method to several window sizes for the training data and select the best window size with statistical backing (by contrasting accuracy metrics for all selected window sizes). This approach will help us predict better.

**To select the window width from all the selected window sizes using the moving average for the residuals, we compare the accuracy measures for various sizes as well:**

```
> round(accuracy(ma.trail_3.pred$mean, valid.ts),3)
             ME  RMSE  MAE   MPE  MAPE  ACF1 Theil's U
Test set 1.761 8.493 7.27 1.148 7.007 0.479     0.914
> round(accuracy(ma.trail_8.pred$mean, valid.ts),3)
             ME  RMSE   MAE    MPE  MAPE  ACF1 Theil's U
Test set -1.66 8.384 7.032 -2.337 7.029 0.487     0.913
> round(accuracy(ma.trail_12.pred$mean, valid.ts),3)
             ME  RMSE  MAE   MPE MAPE  ACF1 Theil's U
Test set 1.437 9.304 7.87 0.613 7.63 0.471     0.982
```

From the common accuracy measure the moving average with window width 3 and 8 have the similar MAPE and RMSE values for both window widths we can take any one of them, we selected window width = 3.

Window width Graph for all three window widths



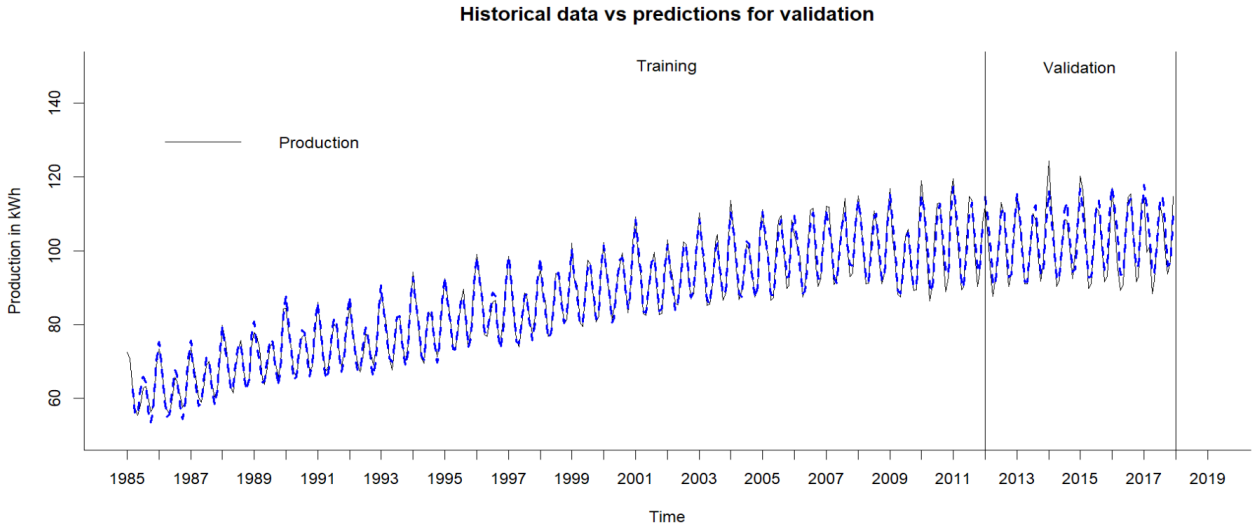

From the above graph we are choosing either window width as 3 or 8 but, in the project, we are using 3 as the window width in the trailing MA.

**Two-level predictions = regression model with linear trend and seasonality + Trailing MA for residual**

**<u>Two level forecasting for validation data</u>**

```
> fst.2level <- trend.seas.pred$mean + ma.trail.res.pred$mean
> fst.2level
           Jan       Feb       Mar       Apr       May       Jun
2012 114.74061 108.46175 100.56177  90.75498  90.83019 100.42056
2013 115.51549 109.23160 101.32660  91.51478  91.58496 101.17030
2014 116.23003 109.94111 102.03108  92.21423  92.27938 101.85969
2015 116.88422 110.59027 102.67521  92.85333  92.91346 102.48874
2016 117.47807 111.17910 103.25901  93.43210  93.48719 103.05745
2017 118.01158 111.70758 103.78246  93.95052  94.00059 103.56582
           Jul       Aug       Sep       Oct       Nov       Dec
2012 110.03886 111.65799 101.44320  92.82135  93.67369 107.01054
2013 110.78356 112.39767 102.17785  93.55098  94.39829 107.73010
2014 111.46793 113.07701 102.85216  94.22026  95.06254 108.38932
2015 112.09195 113.69601 103.46612  94.82919  95.66645 108.98820
2016 112.65563 114.25466 104.01974  95.37779  96.21001 109.52674
2017 113.15897 114.75297 104.51302  95.86604  96.69323 110.00493
```

**Predictions Graph:**




**Forecasted values for validation data:**

```
> fst.2level <- trend.seas.pred$mean + ma.trail.res.pred$mean
> fst.2level
           Jan       Feb       Mar       Apr       May       Jun       Jul
2012 114.74061 108.46175 100.56177  90.75498  90.83019 100.42056 110.03886
2013 115.51549 109.23160 101.32660  91.51478  91.58496 101.17030 110.78356
2014 116.23003 109.94111 102.03108  92.21423  92.27938 101.85969 111.46793
2015 116.88422 110.59027 102.67521  92.85333  92.91346 102.48874 112.09195
2016 117.47807 111.17910 103.25901  93.43210  93.48719 103.05745 112.65563
2017 118.01158 111.70758 103.78246  93.95052  94.00059 103.56582 113.15897
           Aug       Sep       Oct       Nov       Dec
2012 111.65799 101.44320  92.82135  93.67369 107.01054
2013 112.39767 102.17785  93.55098  94.39829 107.73010
2014 113.07701 102.85216  94.22026  95.06254 108.38932
2015 113.69601 103.46612  94.82919  95.66645 108.98820
2016 114.25466 104.01974  95.37779  96.21001 109.52674
2017 114.75297 104.51302  95.86604  96.69323 110.00493
```

**Just to check the comparison we used only quadratic trend without seasonality as well for the level – 1:**

```
> summary(trend)

Call:
tslm(formula = train.ts ~ trend + I(trend^2))

Residuals:
     Min       1Q   Median       3Q      Max
-16.0967  -6.1807  -0.3687   5.2425  18.1477

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.020e+01  1.248e+00  48.228  < 2e-16 ***
trend        2.024e-01  1.774e-02  11.410  < 2e-16 ***
I(trend^2)  -2.090e-04  5.285e-05  -3.955 9.43e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.443 on 321 degrees of freedom
Multiple R-squared:  0.7455,    Adjusted R-squared:  0.7439
F-statistic: 470.1 on 2 and 321 DF,  p-value: < 2.2e-16
```

**The Adjusted – R square value is 0.7439 which is around 74% the output is reliable on the input variables. Comparatively the quadratic trend with seasonality is having high adjusted R – Square value which is a good fit.**

To measure them correctly we are going to use the common accuracy measurements below for the Two – level forecasting.

```
> round(accuracy(trend.seas.pred$mean + ma.trail.res.pred$mean, valid.ts), 3)
             ME  RMSE  MAE   MPE MAPE  ACF1 Theil's U
Test set -1.229 3.626 2.89 -1.34 2.84 0.478     0.375
> round(accuracy(trend.pred$mean + ma.trail.trend.res.pred$mean, valid.ts), 3)
             ME  RMSE   MAE    MPE  MAPE  ACF1 Theil's U
Test set -1.283 7.932 6.896 -1.785 6.814 0.453     0.846
> round(accuracy((snaive(valid.ts))$fitted, valid.ts), 3)
            ME  RMSE   MAE  MPE  MAPE  ACF1 Theil's U
Test set 0.389 4.093 3.093 0.31 2.941 0.449     0.392
```

The Two – Level forecasting with quadratic trend and seasonality with MA trailing for residual has LOW MAPE and RMSE values. Hence, it is a good fit among other two models i.e., seasonal naïve and Two – level with quadratic trend and MA trailing.

**<u>For Entire Dataset (Two – level Forecasting):</u>**

Firstly, we applied quadratic trend with seasonality for entire data and the observations are below from the R – code:

```
Call:
tslm(formula = electric.data.ts ~ trend + I(trend^2) + season)

Residuals:
    Min      1Q  Median      3Q     Max
-9.7939 -2.0740 -0.0358  1.9489  8.7433

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.232e+01  6.787e-01 106.567   <2e-16 ***
trend        2.235e-01  5.333e-03  41.911   <2e-16 ***
I(trend^2)  -2.820e-04  1.301e-05 -21.681   <2e-16 ***
season2     -7.292e+00  7.449e-01  -9.789   <2e-16 ***
season3     -1.360e+01  7.449e-01 -18.256   <2e-16 ***
season4     -2.248e+01  7.449e-01 -30.178   <2e-16 ***
season5     -2.231e+01  7.449e-01 -29.954   <2e-16 ***
season6     -1.442e+01  7.449e-01 -19.362   <2e-16 ***
season7     -7.006e+00  7.449e-01  -9.404   <2e-16 ***
season8     -6.892e+00  7.449e-01  -9.251   <2e-16 ***
season9     -1.569e+01  7.450e-01 -21.058   <2e-16 ***
season10    -2.207e+01  7.450e-01 -29.629   <2e-16 ***
season11    -1.926e+01  7.450e-01 -25.846   <2e-16 ***
season12    -6.742e+00  7.450e-01  -9.050   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.026 on 382 degrees of freedom
Multiple R-squared:  0.962,     Adjusted R-squared:  0.9607
F-statistic: 744.6 on 13 and 382 DF,  p-value: < 2.2e-16
```

Looking at the image above the adjusted – R square is around 0.9607 (96%). Hence, we can conclude that it is a good fit for the data as the p – value is less than 0.05, the model is significant, and the coefficients are significant as well. Hence, this model may be applied for the time series forecasting.

**Model Equation:**

$$Y_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 D_2 + \beta_4 D_3 + \beta_5 D_4 + \beta_6 D_5 + \beta_7 D_6 + \beta_8 D_7 + \cdots + \beta_{12} D_{11} + \varepsilon$$

For this entire dataset based on coefficients model equation is:

$$Y_t = 72.32 + 22.35t - 0.000282t^2 - 7.292D_2 - 0.13601D_3 - 0.2248D_4 - 0.2231D_5 - 0.1442D_6 - 7.006D_7 + \cdots - 6.742D_{12} + \varepsilon$$

***where, t = 1,2,3……n (n=number of time periods/trends)***

***D2 = binary (1,0), it is 1 if Feb and 0 if otherwise***

***D3 = binary (1,0), it is 1 if Mar and 0 if otherwise***

.

.

***D12 = binary (1,0), it is 1 if Dec and 0 if otherwise***

***If D2, D3 ,…, D12 are 0 then it is Jan***

**Forecast of 12 months after applying two – level forecast:**

After getting the quadratic trend and seasonality predicted mean and residuals, we take the residuals and run it through the MA trailing then we combine both the levels to predict the future values of year 2018 as shown in the image below.

```
> tot.fst.2level <- tot.trend.seas.pred$mean + tot.ma.trail.res.pred$mean
> tot.fst.2level
           Jan       Feb       Mar       Apr       May       Jun       Jul
2018 117.91383 110.62145 104.31354  95.43029  95.59507 103.48153 110.89546
           Aug       Sep       Oct       Nov       Dec
2018 111.00532 102.20479  95.81444  98.62613 111.13278
```

**Accuracy Measures:**

```
> ######## accuracy measures #########
> round(accuracy(tot.trend.seas.pred$fitted + tot.ma.trail.res.pred$fitted, electri
c.data.ts), 3)
             ME  RMSE   MAE    MPE  MAPE  ACF1 Theil's U
Test set -0.004 3.273 2.638 -0.055 3.063 0.386     0.455
> round(accuracy((snaive(electric.data.ts))$fitted, electric.data.ts), 3)
             ME  RMSE   MAE  MPE  MAPE  ACF1 Theil's U
Test set 1.243 3.584 2.823 1.47 3.127 0.535     0.462
> round(accuracy((naive(electric.data.ts))$fitted, electric.data.ts), 3)
             ME  RMSE   MAE    MPE  MAPE  ACF1 Theil's U
Test set 0.107 7.731 6.566 -0.235 7.295 0.367         1
>
```

From the above picture it is evident that the MAPE and RMSE values are quite low for Two – level Forecasting with quadratic trend + seasonality and MA trailing method. Hence, we can say that two – level Forecasting with Quadratic trend+ seasonality and MA trailing is better as compared to Seasonal naive and naive methods.

**Holt-winter's:**

The next technique utilized for the time series analysis is the Holt-Winters model of advanced exponential smoothing. The Holt-Winter's (HW) or simply Winter's model is used for time series that exhibit trend and seasonality. By including a seasonal component, the goal is to enhance Holt's model. This model is appropriate for making predictions since it creates estimates that take into consideration both the trend and seasonality components.

**Advanced Exponential Smoothing:**

The Holt-Winters model of advanced exponential smoothing is the next method used for the time series analysis. Since this model makes projections considering both the trend and seasonality components, it is suitable. The model was initially tested using the training and validation partitions before being run on the complete dataset.

**Automated Holt-Winter Model (Z, Z, Z):**

Ets() function uses model=ZZZ and chooses the best possible parameters for alpha, beta, gamma.

**where:**

**α = smoothing constant for exponential smoothing**

**β = smoothing constant for trend estimate**

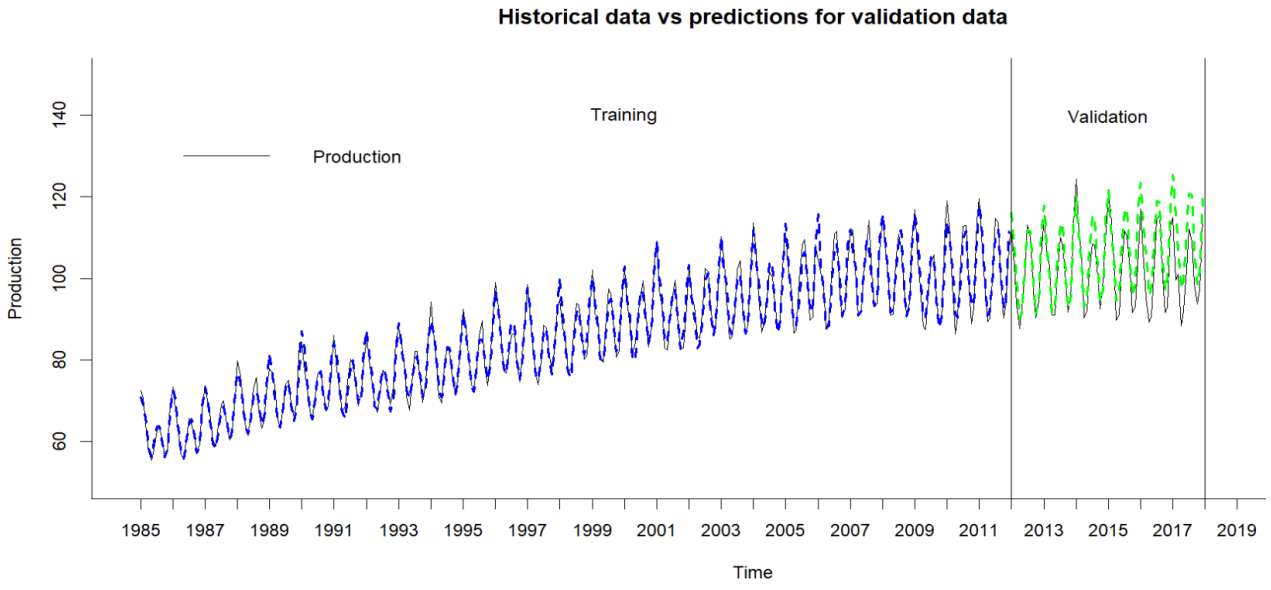
**$\gamma\gamma$ = smoothing constant for seasonality estimate**

**k = periods to be forecasted into future**

**M = number of seasons**

```
> HW.ZZZ <- ets(train.ts, model = "ZZZ")
> HW.ZZZ
ETS(M,A,M)

Call:
 ets(y = train.ts, model = "ZZZ")

  Smoothing parameters:
    alpha = 0.3778
    beta  = 2e-04
    gamma = 0.2052

  Initial states:
    l = 61.6489
    b = 0.1383
    s = 1.076 0.9324 0.9045 0.9691 1.0365 1.0298
           0.963 0.8984 0.9269 1.0135 1.103 1.1469

  sigma:  0.0251

     AIC     AICc      BIC
2378.036 2380.036 2442.308
```

## Plot for training and validation data:



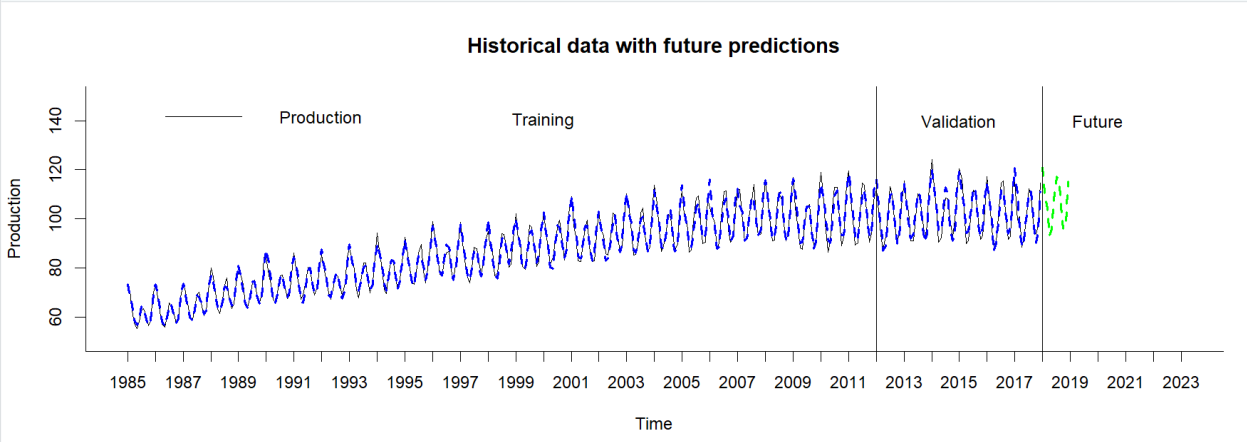
## Holts winter Model built over entire data.

```
> HW.ZZZ.entire <- ets(electric.data.ts, model = "ZZZ")
> HW.ZZZ.entire
ETS(M,A,M)

Call:
 ets(y = electric.data.ts, model = "ZZZ")

  Smoothing parameters:
    alpha = 0.4161
    beta  = 1e-04
    gamma = 0.1865

  Initial states:
    l = 63.0864
    b = 0.1429
    s = 1.0628 0.9363 0.9096 0.9612 1.0278 1.0469
           0.9603 0.9037 0.9297 1.0113 1.0923 1.158

  sigma:  0.026

     AIC     AICc      BIC
3038.513 3040.132 3106.197
```

## Forecast for 12 months in 2018:

```
> HW.ZZZ.entire.pred <- forecast(HW.ZZZ.entire, h = 12 , level = 0)
> HW.ZZZ.entire.pred
         Point Forecast      Lo 0      Hi 0
Jan 2018      121.12644 121.12644 121.12644
Feb 2018      111.04476 111.04476 111.04476
Mar 2018      104.48143 104.48143 104.48143
Apr 2018       93.29755  93.29755  93.29755
May 2018       95.80191  95.80191  95.80191
Jun 2018      107.34365 107.34365 107.34365
Jul 2018      117.06649 117.06649 117.06649
Aug 2018      115.80005 115.80005 115.80005
Sep 2018      104.54608 104.54608 104.54608
Oct 2018       95.94523  95.94523  95.94523
Nov 2018       99.25199  99.25199  99.25199
Dec 2018      115.07688 115.07688 115.07688
```

## Plot of holt's winter model over entire data:




From the above plot we can understand that the holt's winter model is perfectly fitting into the data for entire dataset and is also able to forecast the results of the future year from January 2018 to December 2018.

**Accuracy measure of this model compared with naïve, seasonal naïve and holts winter:**

```
> round(accuracy((snaive(electric.data.ts))$fitted, electric.data.ts), 3)
             ME  RMSE   MAE  MPE  MAPE  ACF1 Theil's U
Test set 1.243 3.584 2.823 1.47 3.127 0.535     0.462
> round(accuracy((naive(electric.data.ts))$fitted, electric.data.ts), 3)
             ME  RMSE   MAE    MPE  MAPE  ACF1 Theil's U
Test set 0.107 7.731 6.566 -0.235 7.295 0.367         1
> round(accuracy(HW.ZZZ.entire.pred$fitted, electric.data.ts), 3)
              ME  RMSE   MAE    MPE  MAPE  ACF1 Theil's U
Test set -0.073 2.421 1.842 -0.131 2.018 0.198     0.307
```

By comparing the MAPE and RMSE values the Holt's winter model is quite better than seasonal naïve and naïve model. Hence, we are using the Holt's winter for the time series forecasting as the measures are better for the data.

**ARIMA Model:**

ARIMA (Autoregressive Integrated Moving Average) model is a widely used statistical model for time series forecasting. It is a combination of autoregression (AR), differencing (I), and moving average (MA) models.

The AR component represents the linear dependence of the current value on past values, where the number of past values to consider is controlled by the order of the model (p). The I component represents the degree of differencing needed to make the series stationary, where the order of differencing is controlled by (d). The MA component represents the linear dependence of the current value on past errors, where the order of the model is controlled by (q).

ARIMA models are capable of capturing trends, seasonality, and other complex patterns in the data. They are widely used in various industries for forecasting applications, such as predicting sales, stock prices, weather patterns, and more. The accuracy of ARIMA models depends on the appropriate selection of model parameters (p, d, q), which can be done using statistical techniques or by trial and error.

**Auto.arima():**

`auto.arima` is a function in the R programming language that automatically selects the best ARIMA model for a given time series data. The function uses a stepwise approach to evaluate a large number of possible ARIMA models and selects the one with the lowest Akaike Information Criterion (AIC) value.

The `auto.arima` function is used because selecting the best ARIMA model for a time series can be a difficult and time-consuming process. The function simplifies this process by automating the selection process and choosing the optimal ARIMA model. This can save time and effort for analysts and help ensure that the best model is selected for forecasting.

**Arima model for training data:**

```
> summary(train.auto.arima)
Series: train.ts
ARIMA(3,0,0)(0,1,1)[12] with drift

Coefficients:
         ar1      ar2     ar3     sma1   drift
      0.6128  -0.0986  0.1543  -0.6676  0.1307
s.e.  0.0577   0.0664  0.0569   0.0440  0.0113

sigma^2 = 4.953:  log likelihood = -693.58
AIC=1399.16   AICc=1399.43   BIC=1421.62

Training set error measures:
                     ME     RMSE      MAE         MPE     MAPE      MASE
Training set 0.02739305 2.166434 1.615992 0.01997704 1.820758 0.5831288
                    ACF1
Training set 0.003350867
```

**In this model:**

ARIMA (3,0,0) (0,1,1) [12] with drift

p = 3, order 3 autoregressive model AR(3)

d = 0, order 0 differencing to remove linear trend

q = 0, order 0 moving average MA(0) for error lags

P = 0, order 0 autoregressive model no AR() for seasonality

D = 1, order 1 differencing to remove linear trend

Q = 1, order 1 moving average MA(1) for error lags

m = 12, for monthly seasonality

**Model Equation:**

$$Y_t - Y_{t-1} = \mathbf{0.1307} + \mathbf{0.6128}(Y_{t-1}) + (\mathbf{-0.0986})(Y_{t-2}) + \mathbf{0.1543}(Y_{t-3}) + (\mathbf{-0.6676})\rho_{t-1}$$

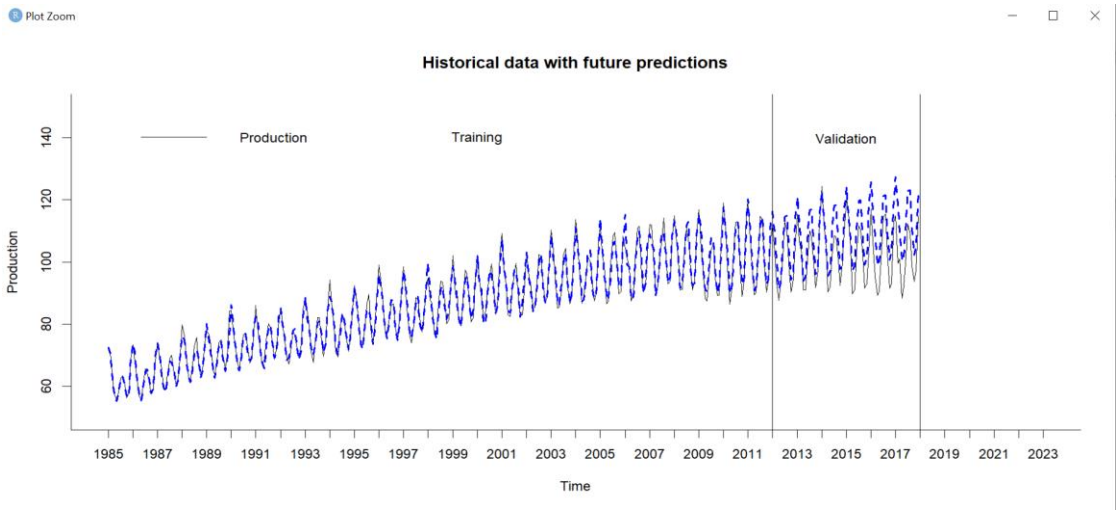
**Forecast for validation:**

| | Point Forecast | Lo 0 | Hi 0 |
|---|---|---|---|
| Jan 2012 | 116.38161 | 116.38161 | 116.38161 |
| Feb 2012 | 109.26790 | 109.26790 | 109.26790 |
| Mar 2012 | 100.93798 | 100.93798 | 100.93798 |
| Apr 2012 | 91.15399 | 91.15399 | 91.15399 |
| May 2012 | 93.12512 | 93.12512 | 93.12512 |
| Jun 2012 | 105.21819 | 105.21819 | 105.21819 |
| Jul 2012 | 114.54627 | 114.54627 | 114.54627 |
| Aug 2012 | 114.85080 | 114.85080 | 114.85080 |
| Sep 2012 | 102.54712 | 102.54712 | 102.54712 |
| Oct 2012 | 94.19282 | 94.19282 | 94.19282 |
| Nov 2012 | 96.85455 | 96.85455 | 96.85455 |
| Dec 2012 | 113.55812 | 113.55812 | 113.55812 |
| Jan 2013 | 121.00911 | 121.00911 | 121.00911 |
| Feb 2013 | 112.47947 | 112.47947 | 112.47947 |
| Mar 2013 | 103.93041 | 103.93041 | 103.93041 |
| Apr 2013 | 93.90505 | 93.90505 | 93.90505 |
| May 2013 | 95.53147 | 95.53147 | 95.53147 |
| Jun 2013 | 107.40330 | 107.40330 | 107.40330 |
| Jul 2013 | 116.59256 | 116.59256 | 116.59256 |
| Aug 2013 | 116.78067 | 116.78067 | 116.78067 |
| Sep 2013 | 104.38521 | 104.38521 | 104.38521 |
| Oct 2013 | 95.96474 | 95.96474 | 95.96474 |

| | | | |
|---|---|---|---|
| Nov 2013 | 98.57701 | 98.57701 | 98.57701 |
| Dec 2013 | 115.24264 | 115.24264 | 115.24264 |
| Jan 2014 | 122.66504 | 122.66504 | 122.66504 |
| Feb 2014 | 114.11400 | 114.11400 | 114.11400 |
| Mar 2014 | 105.54880 | 105.54880 | 105.54880 |
| Apr 2014 | 95.51124 | 95.51124 | 95.51124 |
| May 2014 | 97.12848 | 97.12848 | 97.12848 |
| Jun 2014 | 108.99339 | 108.99339 | 108.99339 |
| Jul 2014 | 118.17744 | 118.17744 | 118.17744 |
| Aug 2014 | 118.36162 | 118.36162 | 118.36162 |
| Sep 2014 | 105.96320 | 105.96320 | 105.96320 |
| Oct 2014 | 97.54050 | 97.54050 | 97.54050 |
| Nov 2014 | 100.15109 | 100.15109 | 100.15109 |
| Dec 2014 | 116.81545 | 116.81545 | 116.81545 |
| Jan 2015 | 124.23690 | 124.23690 | 124.23690 |
| Feb 2015 | 115.68514 | 115.68514 | 115.68514 |
| Mar 2015 | 107.11939 | 107.11939 | 107.11939 |
| Apr 2015 | 97.08143 | 97.08143 | 97.08143 |
| May 2015 | 98.69836 | 98.69836 | 98.69836 |
| Jun 2015 | 110.56304 | 110.56304 | 110.56304 |
| Jul 2015 | 119.74692 | 119.74692 | 119.74692 |
| Aug 2015 | 119.93096 | 119.93096 | 119.93096 |
| Sep 2015 | 107.53244 | 107.53244 | 107.53244 |
| Oct 2015 | 99.10967 | 99.10967 | 99.10967 |
| Nov 2017 | 104.85808 | 104.85808 | 104.85808 |
| Dec 2017 | 121.52240 | 121.52240 | 121.52240 |

| | | | |
|---|---|---|---|
| Nov 2015 | 101.72020 | 101.72020 | 101.72020 |
| Dec 2015 | 118.38452 | 118.38452 | 118.38452 |
| Jan 2016 | 125.80594 | 125.80594 | 125.80594 |
| Feb 2016 | 117.25415 | 117.25415 | 117.25415 |
| Mar 2016 | 108.68839 | 108.68839 | 108.68839 |
| Apr 2016 | 98.65041 | 98.65041 | 98.65041 |
| May 2016 | 100.26733 | 100.26733 | 100.26733 |
| Jun 2016 | 112.13200 | 112.13200 | 112.13200 |
| Jul 2016 | 121.31587 | 121.31587 | 121.31587 |
| Aug 2016 | 121.49992 | 121.49992 | 121.49992 |
| Sep 2016 | 109.10139 | 109.10139 | 109.10139 |
| Oct 2016 | 100.67861 | 100.67861 | 100.67861 |
| Nov 2016 | 103.28915 | 103.28915 | 103.28915 |
| Dec 2016 | 119.95346 | 119.95346 | 119.95346 |
| Jan 2017 | 127.37488 | 127.37488 | 127.37488 |
| Feb 2017 | 118.82310 | 118.82310 | 118.82310 |
| Mar 2017 | 110.25733 | 110.25733 | 110.25733 |
| Apr 2017 | 100.21935 | 100.21935 | 100.21935 |
| May 2017 | 101.83627 | 101.83627 | 101.83627 |
| Jun 2017 | 113.70094 | 113.70094 | 113.70094 |
| Jul 2017 | 122.88481 | 122.88481 | 122.88481 |
| Aug 2017 | 123.06886 | 123.06886 | 123.06886 |
| Sep 2017 | 110.67033 | 110.67033 | 110.67033 |
| Oct 2017 | 102.24755 | 102.24755 | 102.24755 |

## Plot for the auto.arima():




## Auto Arima model for entire data:

```
> entire.auto.arima <- auto.arima(electric.data.ts)
> summary(entire.auto.arima)
Series: electric.data.ts
ARIMA(2,1,1)(0,1,2)[12]

Coefficients:
         ar1      ar2      ma1     sma1     sma2
      0.5321  -0.0734  -0.9446  -0.6651  -0.1064
s.e.  0.0536   0.0542   0.0195   0.0649   0.0619

sigma^2 = 5.628:  log likelihood = -877.9
AIC=1767.81   AICc=1768.03   BIC=1791.5

Training set error measures:
                     ME     RMSE      MAE        MPE     MAPE      MASE
Training set -0.1652422 2.317733 1.772619 -0.2224851 1.930074 0.6278169
                     ACF1
Training set -0.002681408
```

**In this model:**

ARIMA (2,1,1) (0,1,2) [12]

p = 2, order 2 autoregressive model AR(2)

d = 1, order 1 differencing to remove linear trend

q = 1, order 1 moving average MA(1) for error lags

P = 0, order 0 autoregressive model no AR() for seasonality

D = 1, order 1 differencing to remove linear trend

Q = 2, order 2 moving average MA(2) for error lags

m = 12, for monthly seasonality

**Model Equation:**

$$Y_t - Y_{t-1} = \mathbf{0.5321}(Y_{t-1}) + (\mathbf{-0.0734})(Y_{t-2}) + (\mathbf{-0.9446})\,\varepsilon_{t-1} + (\mathbf{-0.6651})\rho_{t-1} + (\mathbf{-0.1064})\rho_{t-2}$$

**Forecast:**

| | Point Forecast | Lo 0 | Hi 0 |
|---|---|---|---|
| Jan 2018 | 119.63932 | 119.63932 | 119.63932 |
| Feb 2018 | 107.67361 | 107.67361 | 107.67361 |
| Mar 2018 | 101.74493 | 101.74493 | 101.74493 |
| Apr 2018 | 90.30881 | 90.30881 | 90.30881 |
| May 2018 | 92.53142 | 92.53142 | 92.53142 |
| Jun 2018 | 103.13300 | 103.13300 | 103.13300 |
| Jul 2018 | 112.52901 | 112.52901 | 112.52901 |
| Aug 2018 | 111.19553 | 111.19553 | 111.19553 |
| Sep 2018 | 100.65442 | 100.65442 | 100.65442 |
| Oct 2018 | 93.17218 | 93.17218 | 93.17218 |
| Nov 2018 | 96.52175 | 96.52175 | 96.52175 |
| Dec 2018 | 111.92159 | 111.92159 | 111.92159 |

**Arima (3,0,0) (0,1,1) [12] model for entire data:**

Summary:

```
Series: electric.data.ts
ARIMA(3,0,0)(0,1,1)[12]

Coefficients:
         ar1      ar2     ar3     sma1
      0.7061  -0.0249  0.2368  -0.7001
s.e.  0.0510   0.0617  0.0497   0.0423

sigma^2 = 6.295:  log likelihood = -900.53
AIC=1811.05   AICc=1811.21   BIC=1830.81

Training set error measures:
                     ME     RMSE      MAE       MPE     MAPE      MASE
Training set 0.3466508 2.457786 1.879018 0.3574635 2.044139 0.6655006
                     ACF1
Training set -0.03250925
```

**In this model:**

ARIMA (3,0,0) (0,1,1) [12] with drift

p = 3, order 3 autoregressive model AR(3)

d = 0, order 0 differencing to remove linear trend

q = 0, order 0 moving average MA(0) for error lags

P = 0, order 0 autoregressive model no AR() for seasonality

D = 1, order 1 differencing to remove linear trend

Q = 1, order 1 moving average MA(1) for error lags

m = 12, for monthly seasonality

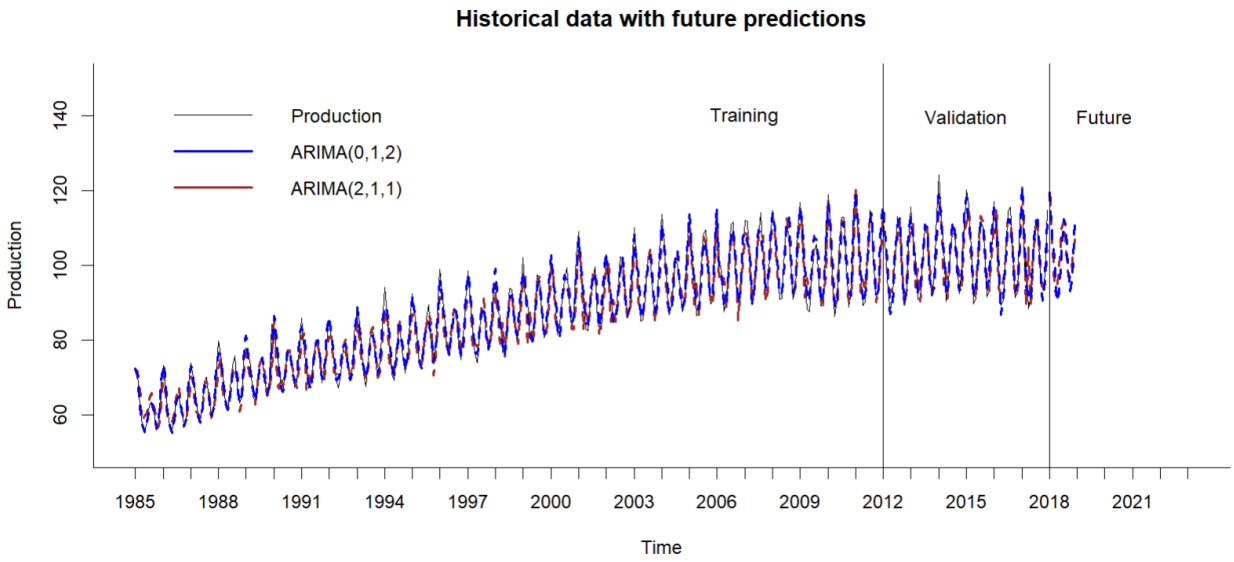
**Model Equation:**

$$Y_t - Y_{t-1} = \mathbf{0.7061}(Y_{t-1}) + (\mathbf{-0.0249})(Y_{t-2}) + \mathbf{0.2368}(Y_{t-3}) + (\mathbf{-0.7001})\boldsymbol{\rho}_{t-1}$$

**Forecast**:

| | Point Forecast | Lo 0 | Hi 0 |
|---|---|---|---|
| Jan 2018 | 121.54860 | 121.54860 | 121.54860 |
| Feb 2018 | 110.14052 | 110.14052 | 110.14052 |
| Mar 2018 | 104.00312 | 104.00312 | 104.00312 |
| Apr 2018 | 92.75131 | 92.75131 | 92.75131 |
| May 2018 | 94.68125 | 94.68125 | 94.68125 |
| Jun 2018 | 105.50746 | 105.50746 | 105.50746 |
| Jul 2018 | 114.90251 | 114.90251 | 114.90251 |
| Aug 2018 | 113.49789 | 113.49789 | 113.49789 |
| Sep 2018 | 102.74008 | 102.74008 | 102.74008 |
| Oct 2018 | 94.53969 | 94.53969 | 94.53969 |
| Nov 2018 | 97.67099 | 97.67099 | 97.67099 |
| Dec 2018 | 113.07578 | 113.07578 | 113.07578 |

## Plot for future prediction:




## Step – 8 Implement Forecast:

```
> round(accuracy(tot.trend.seas.pred$fitted + tot.ma.trail.res.pred$fitted, electric.data.ts), 3)
             ME  RMSE   MAE    MPE  MAPE  ACF1 Theil's U
Test set -0.004 3.273 2.638 -0.055 3.063 0.386     0.455
> round(accuracy(HW.ZZZ.entire.pred$fitted, electric.data.ts), 3)
              ME  RMSE   MAE    MPE  MAPE  ACF1 Theil's U
Test set -0.073 2.421 1.842 -0.131 2.018 0.198     0.307
> round(accuracy(entire.auto.arima.pred$fitted, electric.data.ts), 3)
              ME  RMSE   MAE    MPE MAPE   ACF1 Theil's U
Test set -0.165 2.318 1.773 -0.222 1.93 -0.003     0.292
> round(accuracy(entire.300.arima.pred$fitted, electric.data.ts), 3)
             ME  RMSE   MAE   MPE  MAPE   ACF1 Theil's U
Test set 0.347 2.458 1.879 0.357 2.044 -0.033     0.313
> round(accuracy((snaive(electric.data.ts))$fitted, electric.data.ts), 3)
             ME  RMSE   MAE  MPE  MAPE  ACF1 Theil's U
Test set 1.243 3.584 2.823 1.47 3.127 0.535     0.462
> round(accuracy((naive(electric.data.ts))$fitted, electric.data.ts), 3)
             ME  RMSE   MAE    MPE  MAPE  ACF1 Theil's U
Test set 0.107 7.731 6.566 -0.235 7.295 0.367         1
```

**Below is the comparison table of all methods:**

| | MAPE | RMSE |
|---|---|---|
| **Two-level (MA.Trailing Residuals)** | 3.063 | 3.273 |
| **HoltWinter's.zzz** | 7.295 | 7.731 |
| **auto.arima** | 1.93 | 2.318 |
| **ARIMA(3,0,0)(0,1,1)[12]** | 2.044 | 2.458 |
| **Seasonal Naïve** | 3.127 | 3.584 |

Auto.Arima() is having lowest MAPE value so, we are selecting auto.arima() is our best model to forecast the result.

**Forecast auto.arima() for next 12 months:**

```
> entire.auto.arima.pred
         Point Forecast      Lo 0      Hi 0
Jan 2018      119.63932 119.63932 119.63932
Feb 2018      107.67361 107.67361 107.67361
Mar 2018      101.74493 101.74493 101.74493
Apr 2018       90.30881  90.30881  90.30881
May 2018       92.53142  92.53142  92.53142
Jun 2018      103.13300 103.13300 103.13300
Jul 2018      112.52901 112.52901 112.52901
Aug 2018      111.19553 111.19553 111.19553
Sep 2018      100.65442 100.65442 100.65442
Oct 2018       93.17218  93.17218  93.17218
Nov 2018       96.52175  96.52175  96.52175
Dec 2018      111.92159 111.92159 111.92159
```

**Forecast of ARIMA (3,0,0)(0,1,1)[12] for 12 months:**

```
> entire.300.arima.pred
         Point Forecast      Lo 0      Hi 0
Jan 2018      121.54860 121.54860 121.54860
Feb 2018      110.14052 110.14052 110.14052
Mar 2018      104.00312 104.00312 104.00312
Apr 2018       92.75131  92.75131  92.75131
May 2018       94.68125  94.68125  94.68125
Jun 2018      105.50746 105.50746 105.50746
Jul 2018      114.90251 114.90251 114.90251
Aug 2018      113.49789 113.49789 113.49789
Sep 2018      102.74008 102.74008 102.74008
Oct 2018       94.53969  94.53969  94.53969
Nov 2018       97.67099  97.67099  97.67099
Dec 2018      113.07578 113.07578 113.07578
```

## CONCLUSION

In this study, we used a number of data analytics approaches (Time series) to acquire some understanding of power production. To analyze its association with the data, we used a variety of models from Regression (two-level), Holt's winter, and Auto Arima. This concluding analysis showed that the ARIMA models with the lowest MAPE and RMSE are the Auto.Arima() model, which comes in first, followed by the ARIMA (3,0,0) (0,1,1) [12] model, and finally, the forecast for the next year of 2023 utilizing the Auto Arima model.

**Limitations on study:**

- Limitation in study regarding the electric production and economics of them to understand about the trend.
- This analysis is only based on the historical data and do not include factors that impact the electricity production such as weather conditions or low in manpower, anything else.

**Reference for the data:**

https://www.kaggle.com/datasets/kandij/electric-production.