Forecasting and Analysis of ERCOT Electric Load Curves

Balaji Mediboina

Abstract

Analyzing electrical load curves is essential for efficient power plant operations and infrastructure planning by utility companies. Accurately forecasting power grid demand is critical to ensuring a stable and reliable energy supply. This project focuses on load curve data from the Electric Reliability Council of Texas (ERCOT), a non-profit organization under the Texas Legislature responsible for managing the state's power grid. The dataset, spanning from 2002 to 2023 and covering eight different zones within the grid, required preprocessing to transform it into weekly and monthly frequencies for analysis. Deterministic trend modeling and ARIMA modeling were applied to both time scales to enhance predictive accuracy. The project provides a two-year forecast, which is subsequently compared to actual load values. The findings highlight the effectiveness of the forecasting models and discuss the insights gained from examining the data at multiple temporal scales.

Contents

Introduction

Operations of generation, transmission and distribution of electricity are heavily dependent on the magnitude of demand faced by the system during different times of the day. In practice, this demand is tracked over designated areas of consumption by a central governing authority, which also settles the price per unit of energy in this market. Logging, processing, analysis as well as prediction of expected load on the system is of utmost importance for the central authority, utilities and the various commercial and private consumers in the region. (Mihai et al., 2010) and (Jia et al., 2018) show some of the techniques used to model this load curve by considering the kind of electrical appliances that are expected to operate in residential and commercial settings respectively.

A more time-series oriented approach is presented by (Maybee et al., 1979) using piece-wise linear approximations and forecast using Box-Jenkins (used interchangeably with ARIMA modelling in recent times) time series analysis is shown. Analysis was carried out for 10 years' worth of data given by Federal Power Commission on two regions each consisting of multiple states in the US. We are going to follow a similar approach, but instead of using seasonal differencing, our analysis will first try to fit the deterministic trend and then model the stationary residuals using ARIMA modelling.

Technically speaking, the load demand experienced by a power grid is the sum of all the electrical equipment connected at the user end, expressed in Watts. For a region spanning as wide as a state in a country, this power demand would be expressed in the order of Megawatts. Utilities can record this data for different designated zones within the vast grid, using time frequency of as high as 15 minutes to 1 hour. A daily load curve is aggregation of this data for a period of 24 hours. Figure 1 shows an example curve from our dataset at an hourly frequency.

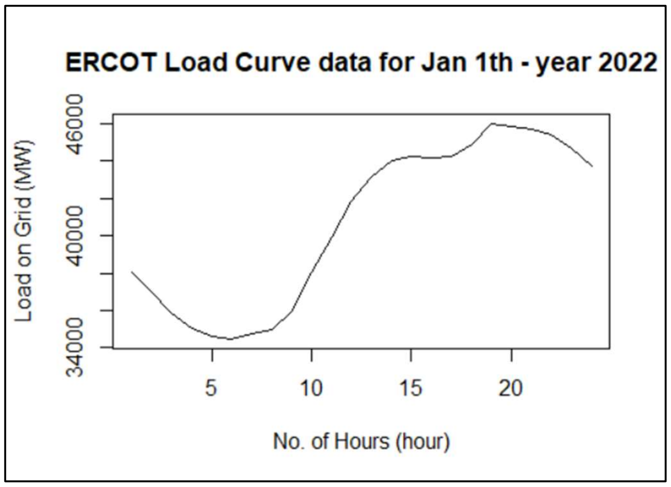


Figure 1. An example of daily load curve picked from the ERCOT dataset.

Our goal is to explore two research questions through this analysis – Firstly, How effective is this data modelling and prediction methodology for the given forecasting problem with univariate data? To answer this question, we are going to set aside the last two years of the known data in order to compare it with our forecast for this period. Metrics such as Root mean square error (RMSE) and Mean absolute percentage error (MAPE) help in quantifying results in way that can be compared with other approaches in literature. The second question asks about the effect of observing the same data at different time scales and its effect on the prediction. This would be answered on the basis of a comparison of the forecast results for the weekly and monthly adjusted frequency time series.

Time Series Data

Our time series data is taken from the Electric Reliability Council Of Texas (ERCOT) website, consisting of data from 8 different zones – North, North central, South, South central, East, Coast, West and Far west, within the state of Texas. (Figure 2)

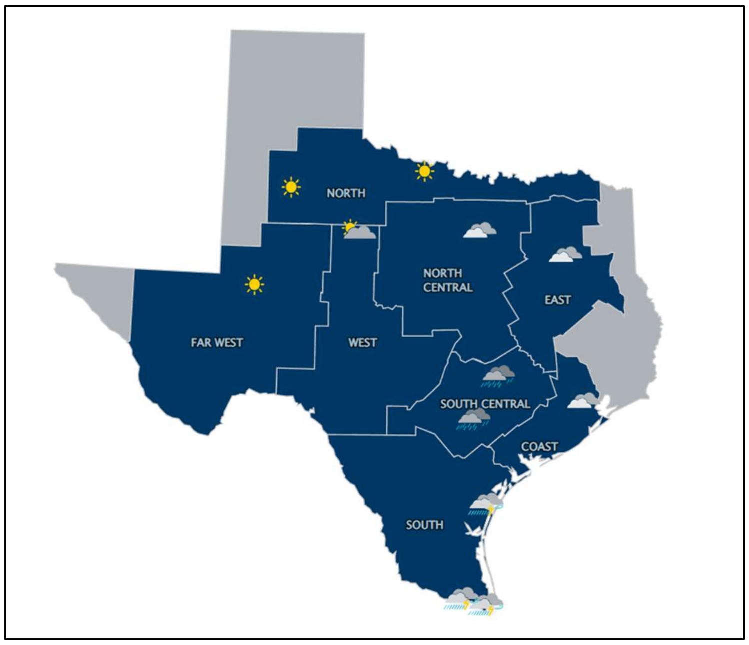


Figure 2. Areas under the ERCOT grid

Data was in the form of 'csv' files starting from 1995 all the way up to March 2023, as of writing this report. However, the older files did not have a clear indication of column headings and the year 2001 was missing entirely from the records. Therefore, we decided to consider the data from the year 2002 all the way up to the year 2022, which includes 23 years' worth of data. Each year consists of 8760 datapoints for a regular year and 24 additional data points if it is a leap year.

As the number of data points was too large to be handled using the tools covered in our course, we thought of processing the data to reduce the number of data points. First, the data was averaged over each week across all 23 years. This was done by identifying a starting point (for example, a Monday in the first week of January 2002) and then averaging every 24 x 7 = 168 datapoints from then on. The weekly dataset, therefore, has 1092 datapoints in total. It is plotted in figure 3, showing seasonality and a linear upward trend. The plot only shows data up to 2020 because the data from years 2021 and 2022 is kept aside for comparison against the forecast.

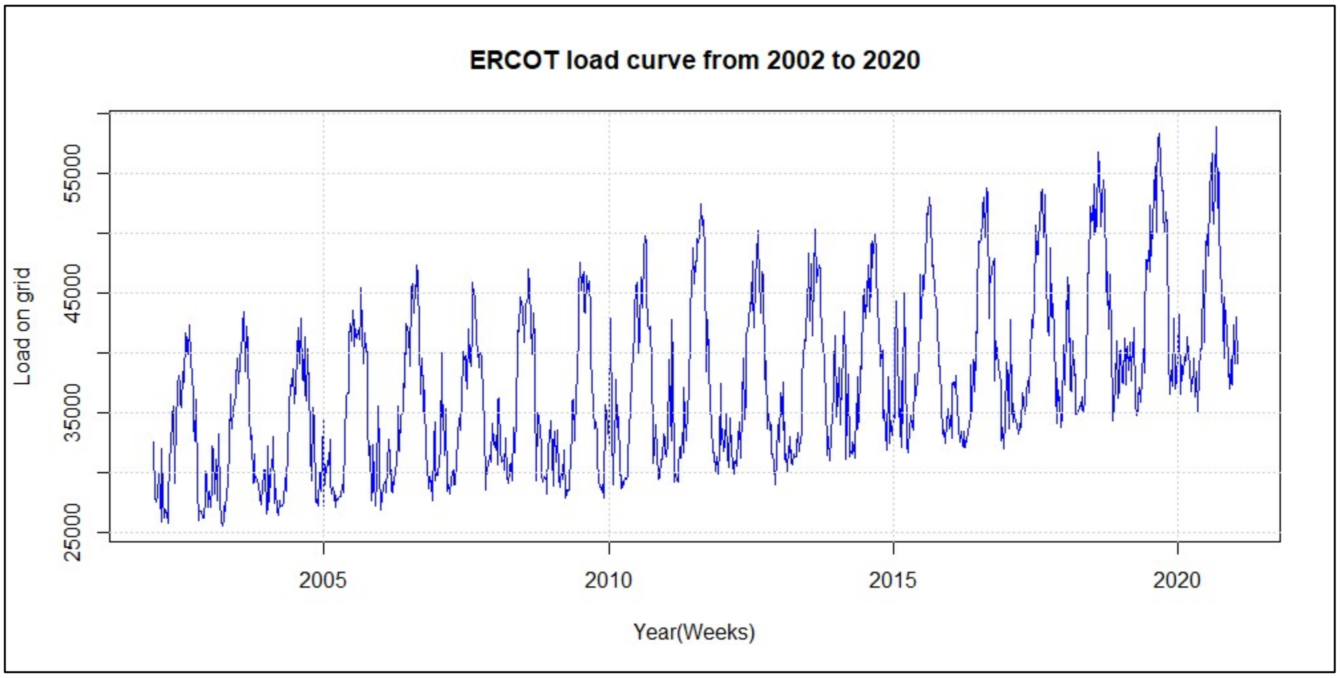


Figure 3. ERCOT dataset observed at weekly frequency.

Similarly, the data was averaged across each month for all the 23 years. This was not as straight forward because of the varying number of days in each month and considering leap years. With these considerations, the data was accurately averaged across proper intervals ending up 256 datapoints across the entire dataset. Again, figure 4 shows the data up to 2020, which shows a clear upward trend along with seasonality.

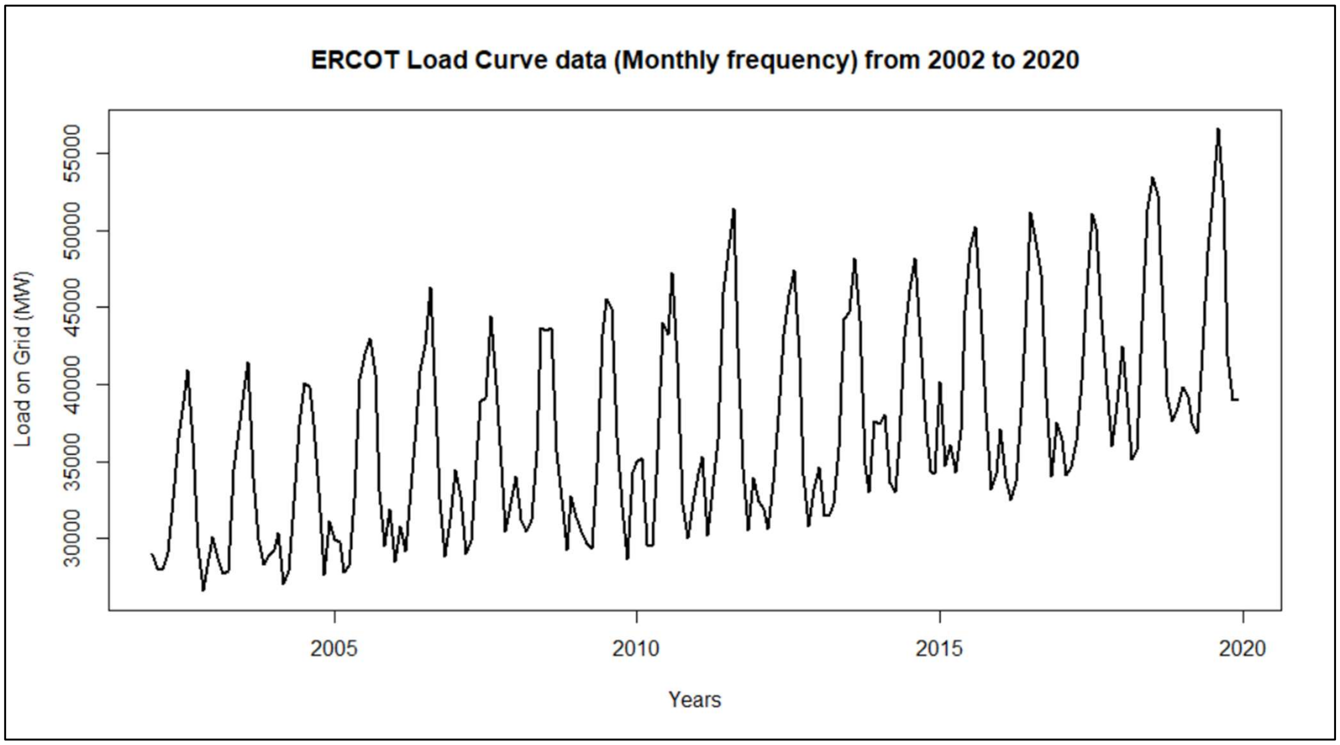


Figure 4. ERCOT dataset observed at monthly frequency.

Transformations

Considering the nature of both time series shown in Figure 5, it was deemed that some transformation was necessary to correct the non-normality seen in these distributions. Specifically, a right skew is seen in both instances of data.

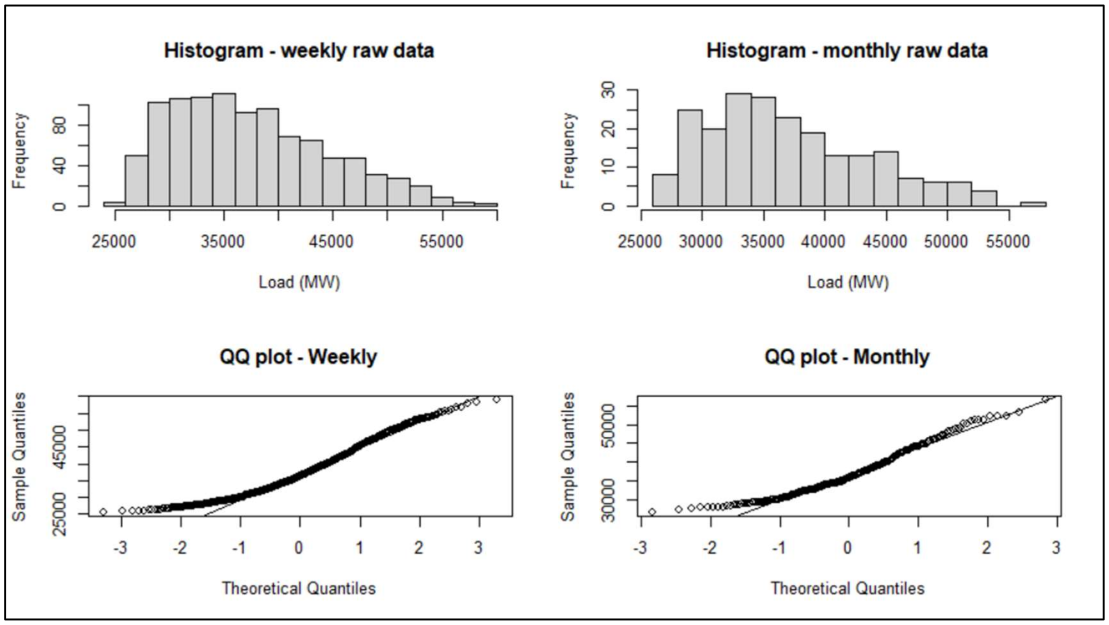


Figure 5. Histogram and QQ plots for Weekly and Monthly Data before transformation

For correcting these distributions, we use the Box-Cox Transformation, which is given by the following expression –

$$\boxed{y_i^{(\lambda)} = \begin{cases} \dfrac{y_i^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \ln(y_i) & \text{if } \lambda = 0, \end{cases}}$$

Where based on the value of λ the transformation can be interpreted as follows –

Value of λ	Transformed Data
-2	y^{-2}
-1	y^{-1}

-0.5	$1/\sqrt{y}$
0	Ln(y)
0.5	$\sqrt{y}$
1	y
2	y^2

Table 1. Interpretation of Box-Cox transform for different λ values.

We use the boxcox() function from MASS package in order to estimate a good value for λ for each of the time series. For weekly data, λ = -0.87 was obtained and for monthly data, λ = -1.23 was obtained. The transformation results along with the respective histograms are shown in figure 6. The transformation does not result in perfectly normal distribution, but it improves the p-value of Shapiro-Wilk's test for normality. Improvement of 5.788e-16 to 4.207e-09 for weekly data and 1.132e-06 to 0.003484 for monthly data is seen. Therefore, we proceed with this transformed data for deterministic trend modelling.

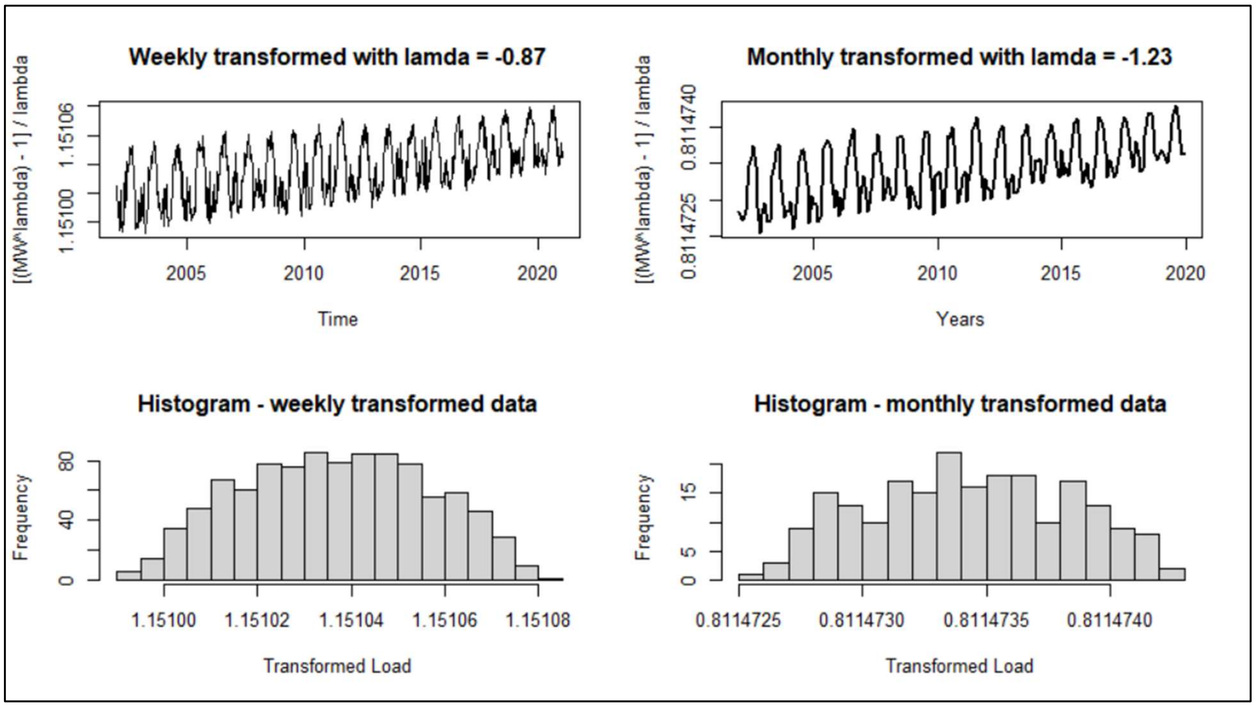


Figure 6. Box-Cox transform for weekly and monthly data and their respective histogram plots.

Deterministic Trend Model

Seasonal Means Model:

Seasonal means involve calculating the average value of a variable during specific periods, such as each month or quarter of the year. This can help identify seasonal patterns in the data, such as increased consumption of electricity due to higher temperatures during the summer months. For the weekly data, a seasonal frequency of 52 can be considered over a single year. For the monthly data, identifying the seasonal frequency of 12 is simple.

Linear Trend Model:

Linear trend models involve fitting a straight line to a set of data points over time. This allows for the identification of a trend, such as an increase or decrease in the variable being measured. The two values which characterize the fit are the intercept (starting reference) and the slope of the line (rate of change in the variable).

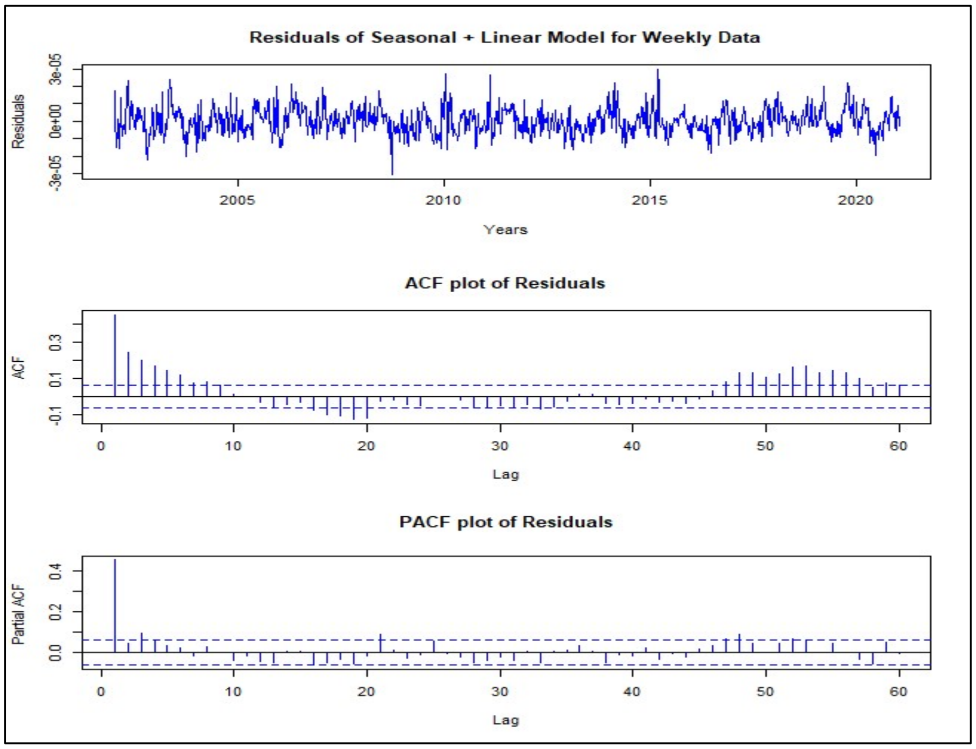


Figure 7. Residuals, ACF, and PACF of Seasonal + Linear Model for weekly data.

These models can also be used in combination to gain a more complete understanding of trends in data over time. For our analysis, we used these two elements to fit a deterministic model on the transformed weekly and monthly data. Figures 7 and 8 show the residuals, Auto Correlation Function (ACF), and Partial Auto Correlation Function (PACF) of both weekly and monthly data, respectively.

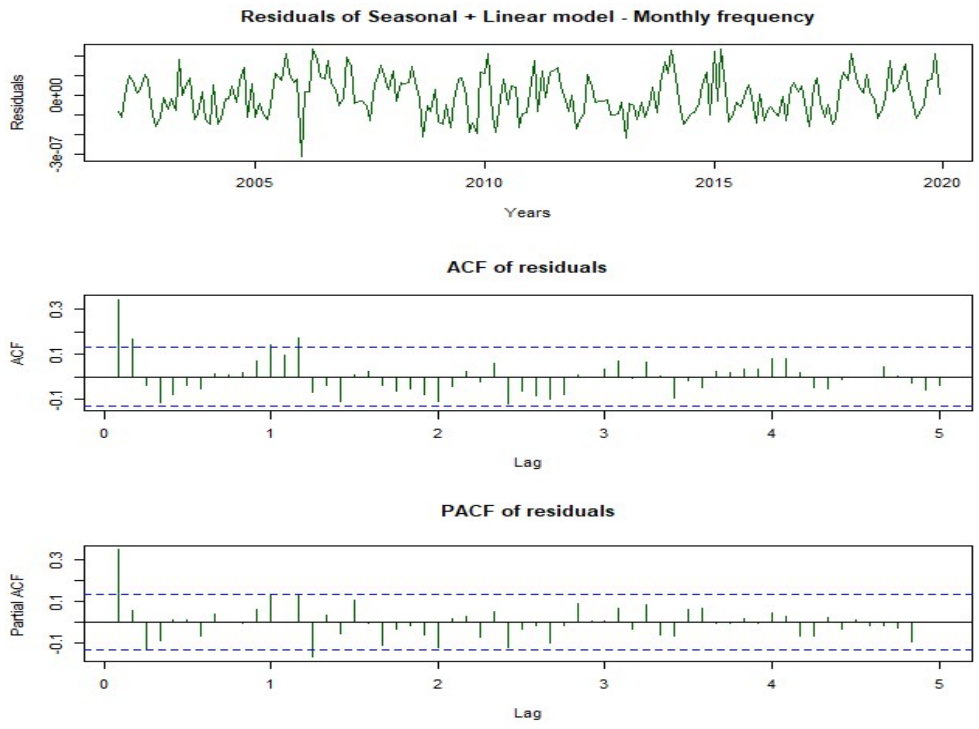


Figure 8. Residuals, ACF, and PACF of Seasonal + Linear Model for monthly data.

The resulting residuals in both cases look fairly stationary having mean centered around zero and constant variance throughout the entire time period observed. The ACF for weekly data (figure 7) dies out after around 6 lags, but a surge of significant values is seen near lag 52 which is concerning because that is the seasonal lag 1. This suggests there is still some amount of seasonality remaining in the residuals which is unaccounted for. However, considering that each week (7) does not perfectly align with the interval of one year (365/366), some unexpected behavior is natural. This issue is not observed in the case of residuals for monthly data (figure 8), so they can be safely declared as stationary.

In order to be completely sure, we performed the Augmented Dickey-Fuller Test, Phillips-Perron Test, and KPSS Test on both residuals to check for stationarity. From the table compiled below, we can see that all the test's p-values support the hypothesis that the weekly and monthly data are stationary. No stochastic trend is present, and differencing is not necessary.

	ADF Test	PP Test	KPSS Test
Weekly Data	p = 0.01	p = 0.01	p = 0.1
Monthly Data	p = 0.01	p = 0.01	p = 0.1
Hypothesis	Alternate	Alternate	Null
Unit Root	Absent	Absent	Absent

Table 2. Unit Root tests on the residuals of weekly and monthly data.

Now that both weekly and monthly data are modelled with a deterministic trend and the residuals obtained are stationary, we can proceed to ARIMA modelling of these residuals by selecting a few candidate models.

Candidate ARIMA Models for Stationary residuals

From figure 9, ACF and PACF of weekly residuals suggest ARMA (0, 8) and ARMA (1, 0) or ARMA (3, 0) respectively. From the monthly residuals, we can see that ACF suggests ARMA (0, 2) and PACF suggests ARMA (1, 0). These models are selected based on the lag at which ACF and PACF plots cut off for each data. The lag where ACF cuts off gives the potential MA order, while PACF cut off gives the potential AR order.

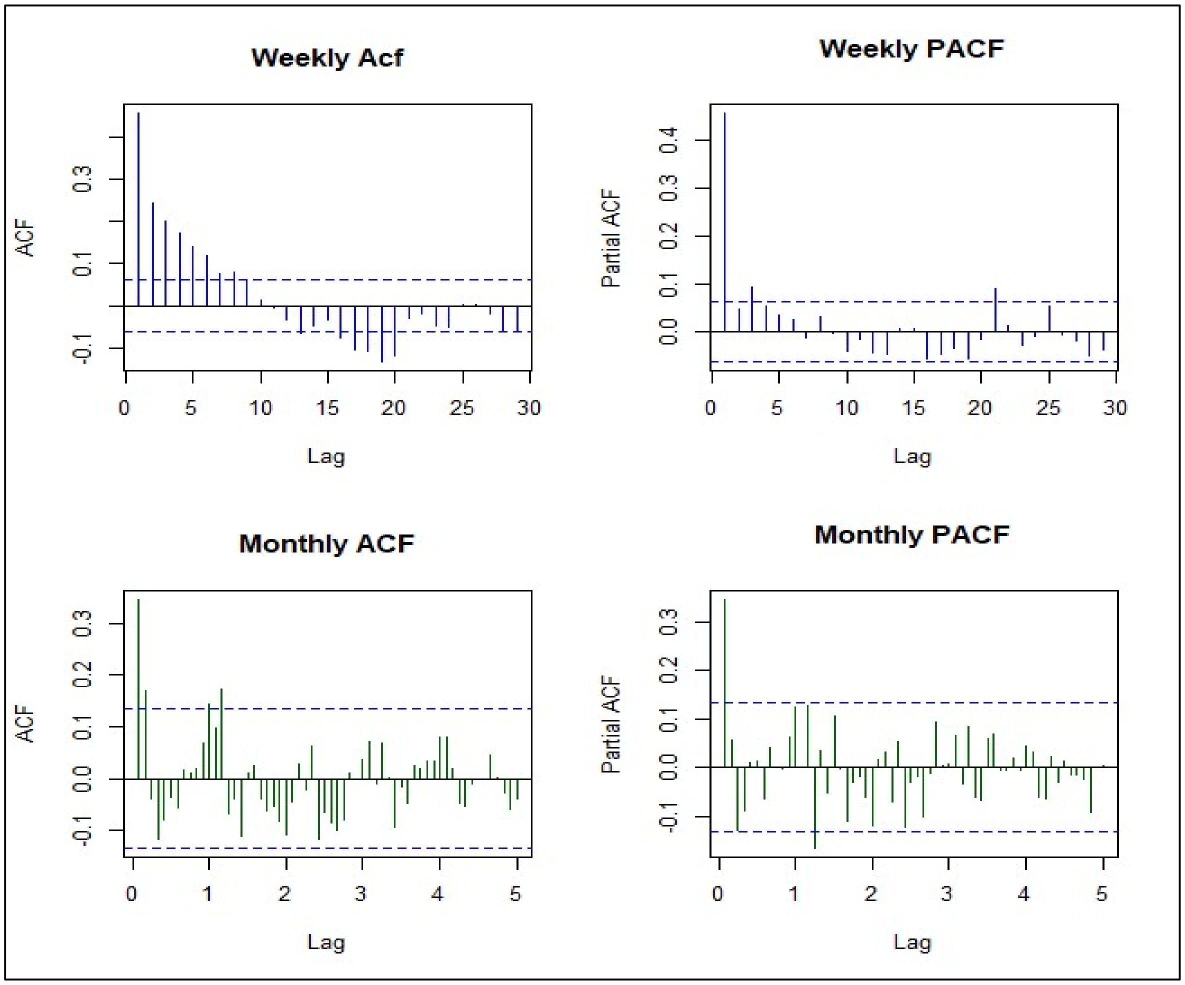
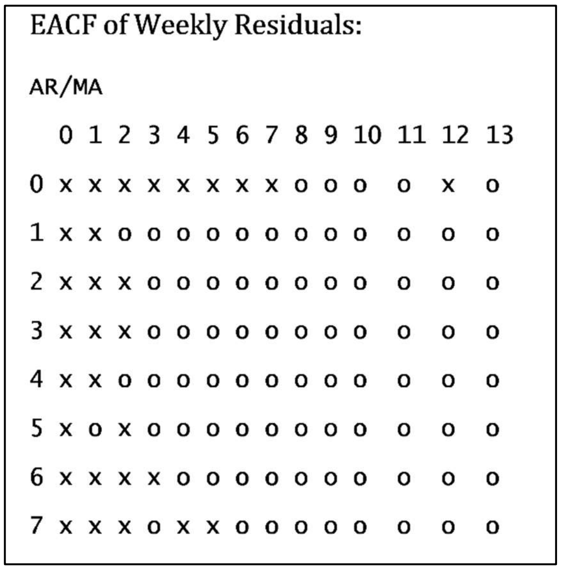


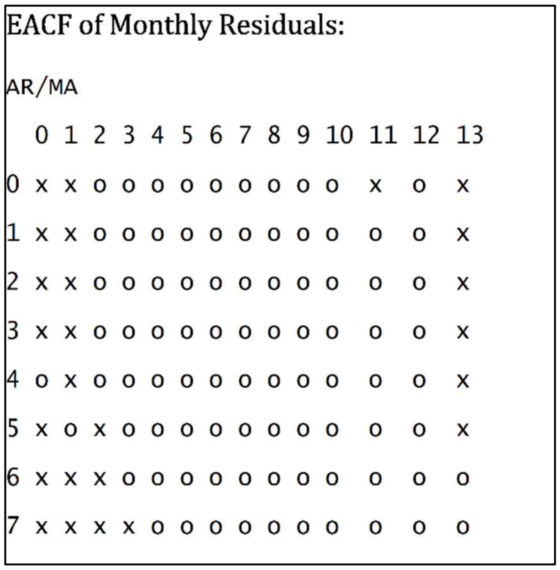
Figure 9. ACF and PACF of weekly and monthly residuals



EACF of Weekly Residuals:

AR/MA

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	x	x	x	x	x	x	x	x	o	o	o	o	x	o
1	x	x	o	o	o	o	o	o	o	o	o	o	o	o
2	x	x	x	o	o	o	o	o	o	o	o	o	o	o
3	x	x	x	o	o	o	o	o	o	o	o	o	o	o
4	x	x	o	o	o	o	o	o	o	o	o	o	o	o
5	x	o	x	o	o	o	o	o	o	o	o	o	o	o
6	x	x	x	x	o	o	o	o	o	o	o	o	o	o
7	x	x	x	o	x	x	o	o	o	o	o	o	o	o



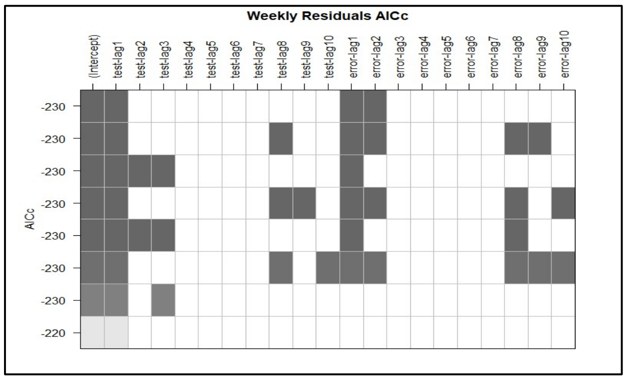
EACF of Monthly Residuals:

AR/MA

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	x	x	o	o	o	o	o	o	o	o	o	x	o	x
1	x	x	o	o	o	o	o	o	o	o	o	o	o	x
2	x	x	o	o	o	o	o	o	o	o	o	o	o	x
3	x	x	o	o	o	o	o	o	o	o	o	o	o	x
4	o	x	o	o	o	o	o	o	o	o	o	o	o	x
5	x	o	x	o	o	o	o	o	o	o	o	o	o	x
6	x	x	x	o	o	o	o	o	o	o	o	o	o	o
7	x	x	x	x	o	o	o	o	o	o	o	o	o	o

Figure 10. EACF of weekly and monthly residuals.

The Extended Auto-Correlation Function (EACF) plot can be used to identify ARMA order by selecting vertices forming triangular pattern. Using this rule, EACF suggests ARMA (1, 2), ARMA (4, 2), and ARMA (0, 8) for weekly residuals and ARMA (4, 0), ARMA (4, 2) for monthly residuals.



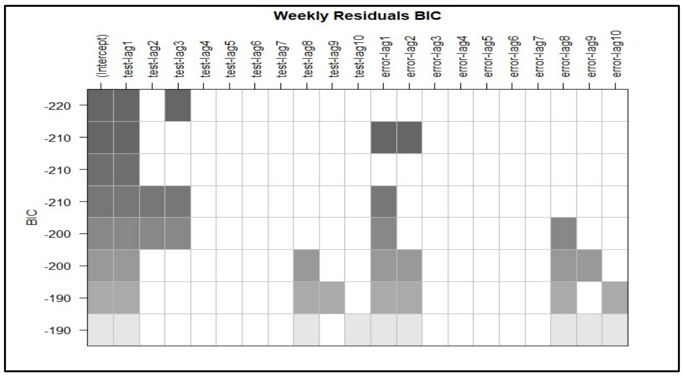


Figure 11. ARMA subsets - model rankings of weekly residuals by AICc and BIC

The top three candidates based on AICc and BIC were selected from the output of ARMA subsets. From figure 11, AICc gives the ARMA (1, 2), ARMA (8, 9), & ARMA (3, 1) models and BIC gives the ARMA (3, 0), ARMA (1, 2), and ARMA (1, 0) models for weekly data.

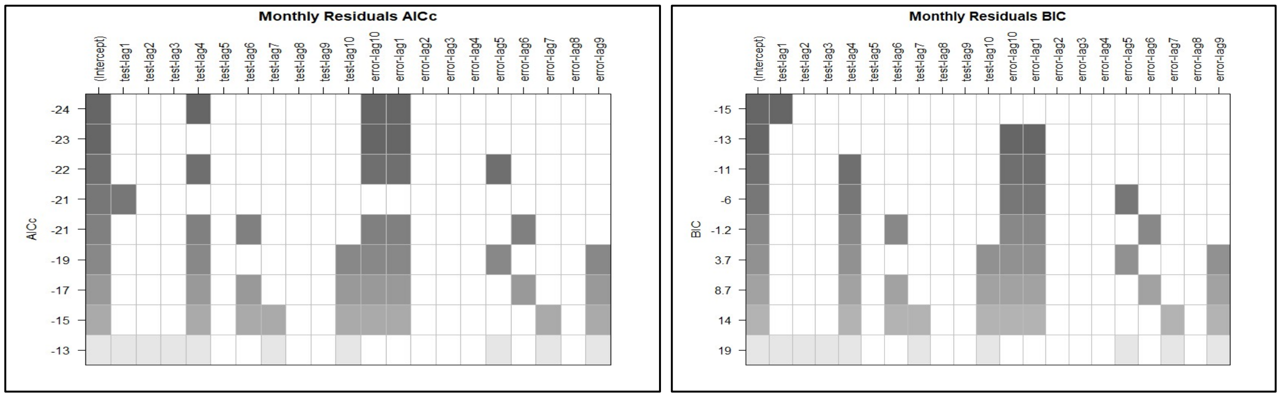


Figure 12. ARMA subsets - model rankings of monthly residuals by AICc and BIC

From figure 12, AICc gives the ARMA (4, 10), ARMA (1, 2), and ARMA (1, 0) models and BIC gives the ARMA (1, 0), ARMA (0, 10), and ARMA (4, 10) models for weekly data.

```
> auto.arima(weekly_resid) #AR(4)
Series: weekly_resid
ARIMA(4,0,0) with zero mean

Coefficients:
         ar1     ar2     ar3     ar4
      0.4265  0.0031  0.0710  0.0517
s.e.  0.0319  0.0346  0.0346  0.0319

sigma^2 = 4.55e-11:  log likelihood = 10
395.16
AIC=-20780.33   AICc=-20780.27   BIC=-20
755.83
```

```
> auto.arima(monthly_resid)
Series: monthly_resid
ARIMA(0,0,0) with zero mean

sigma^2 = 1.098e-14:  log likelihood = 3
164.95
AIC=-6327.91   AICc=-6327.89   BIC=-6324
.53
```

The auto.arima function gave ARMA (4, 0) for weekly residuals and White Noise for monthly residuals.

Model Selection

Weekly Data					Monthly Data				
Source	Models	AIC	AICc	BIC	Source	Models	AIC	AICc	BIC
ACF, PACF	ARMA(1,8)	-20774.0	-20773.8	-20725	ACF, PACF	ARMA(1,2)	-6353	-6352.8	-6339.5
EACF	ARMA(1,2)	-20783.2	-20783.2	-20763.6	EACF	ARMA(4,0)	-6353.5	-6353.2	-6336.6
	ARMA(4,2)	-20777.7	-20777.6	-20743.4		ARMA(2,4)	-6353.7	-6353.1	-6330
	ARMA(0.8)	-20772.2	-20772.1	-20728.2					
ARMA.subsets (AICc)	ARMA(1,2)	-20783.2	-20783.2	-20763.6	ARMA.subsets (AICc)	ARMA(4,10)	-6344.4	-6342	-6293.8
	ARMA(8,9)	-20785.4	-20784.7	-20697.2		ARMA(1,2)	-6353	-6352.8	-6339.5
	ARMA(3,3)	-20777.5	-20777.4	-20743.2		ARMA(1,0)	-6353.2	-6353.1	-6346.4
ARMA.subsets (BIC)	ARMA(3,0)	-20779.7	-20779.7	-20760.1	ARMA.subsets (BIC)	ARMA(1,0)	-6353.2	-6353.1	-6346.4
	ARMA(1,2)	-20783.2	-20783.2	-20763.6		ARMA(0,10)	-6343	-6341.7	-6305.8
	ARMA(1,0)	-20773.2	-20773.2	-20763.4		ARMA(4,10)	-6344.4	-6342	-6293.8
Auto.arima	ARMA(4,0)	-20780.3	-20780.2	-20755.8	Auto.arima	ARMA(0,0)	-6327.9	-6327.9	-6324.5

Table 3. Candidate Models' Performance comparison.

AIC, AICc, and BIC were used to evaluate the candidate models from all the methods and determine which model performed the best (Table 3). Since the primary objective is to predict and forecast, AICc is the main metric utilized to choose the model since it is preferred for forecasting over BIC. We can see from the above table, weekly data's AICc is low for ARMA (8, 9) and ARMA (1, 2) among other models. For the monthly data, AICc of ARMA (4, 0) and ARMA (1, 0) are almost same, so we choose ARMA (1, 0) for being simpler. Though the AICc of ARMA (8, 9) is better than ARMA (1, 2), again the values were pretty close and ARMA (1, 2) was much simpler than ARMA (8, 9), so we decided to go with ARMA (1, 2).

Therefore, we proceed with ARMA (1, 2) for the weekly data and ARMA (1, 0) for the monthly data as our final models.

Model Diagnostics

The tsdiag() function produces a time series plot of the standardized residuals of the given model, their ACF plot and the Ljung-Box test statistic for different value of lags.

Standardized Residuals: The plot shows the pattern of the residuals over time, and a good model should have residuals that are randomly distributed around zero and do not exhibit any discernible patterns. This is observed for the residuals in figures 13 and 14, showing no issue with zero mean or homoscedasticity.

ACF of residuals: Viewed same as the ACF plots in previous stages. The value should not be significant (above the blue-dashed line) for any lags except for being 1 at lag zero. Again, this criterion is satisfied for ACF plot of both the models, as there is no sign of any significant value after lag zero.

Ljung-Box test: It is a statistical test used to test the null hypothesis that the residuals of the time series model are independently and identically distributed. A good model should have residuals that are independently and identically distributed, and the test should not reject the null hypothesis. The p-values for the test up to lag 10 are way greater than 0.05, so the null hypothesis is not rejected. So, it can be safely concluded that the residuals are independent.

The model diagnostics of both weekly and monthly data tell us that the final models selected satisfy all the assumptions. Therefore, we can proceed with forecast using ARMA (1, 2) model for weekly and ARMA (1, 0) for monthly data.

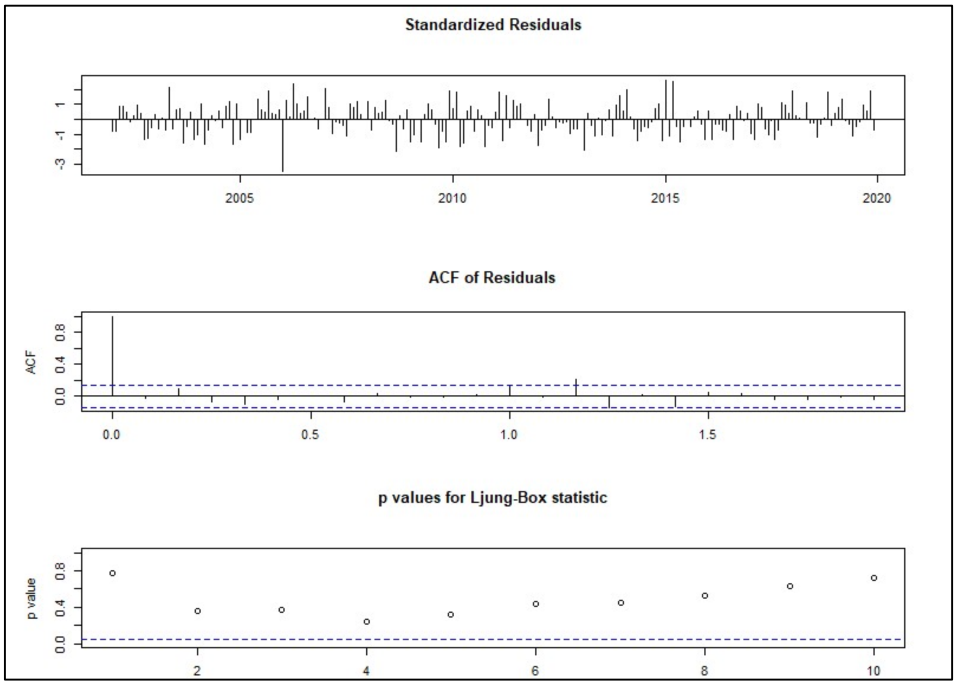


Figure 13. Model Diagnostics (std residuals, ACF and Ljung-Box test) of Linear plus Seasonal means detrended ARMA (1, 2) for Weekly Data.

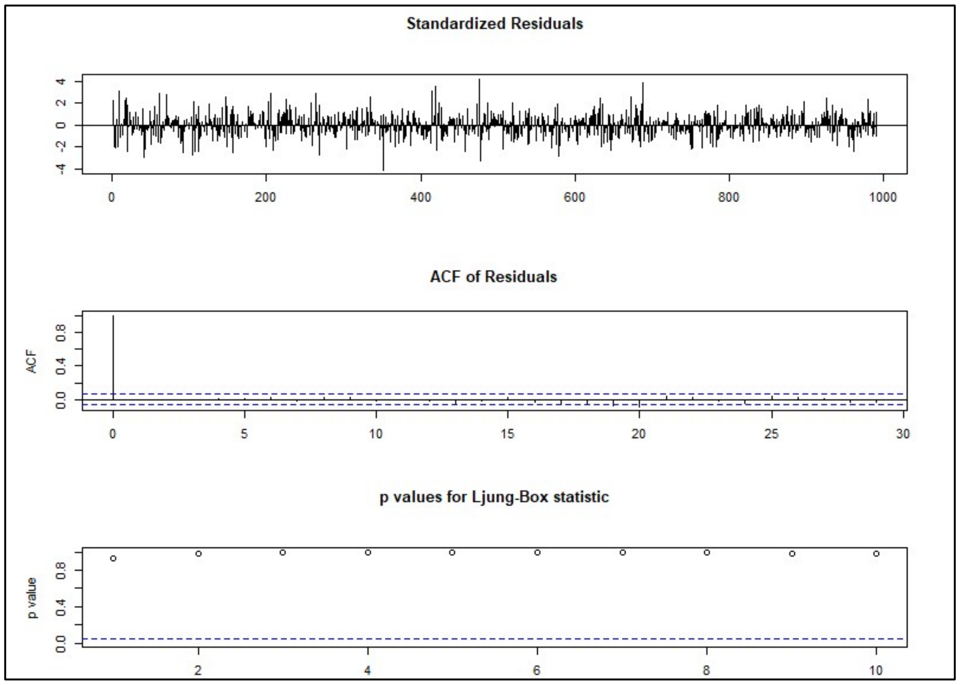


Figure 14. Model Diagnostics (std residuals, ACF and Ljung-Box test) of Linear plus Seasonal means detrended ARMA (1, 0) for Monthly Data.

Forecasting Results

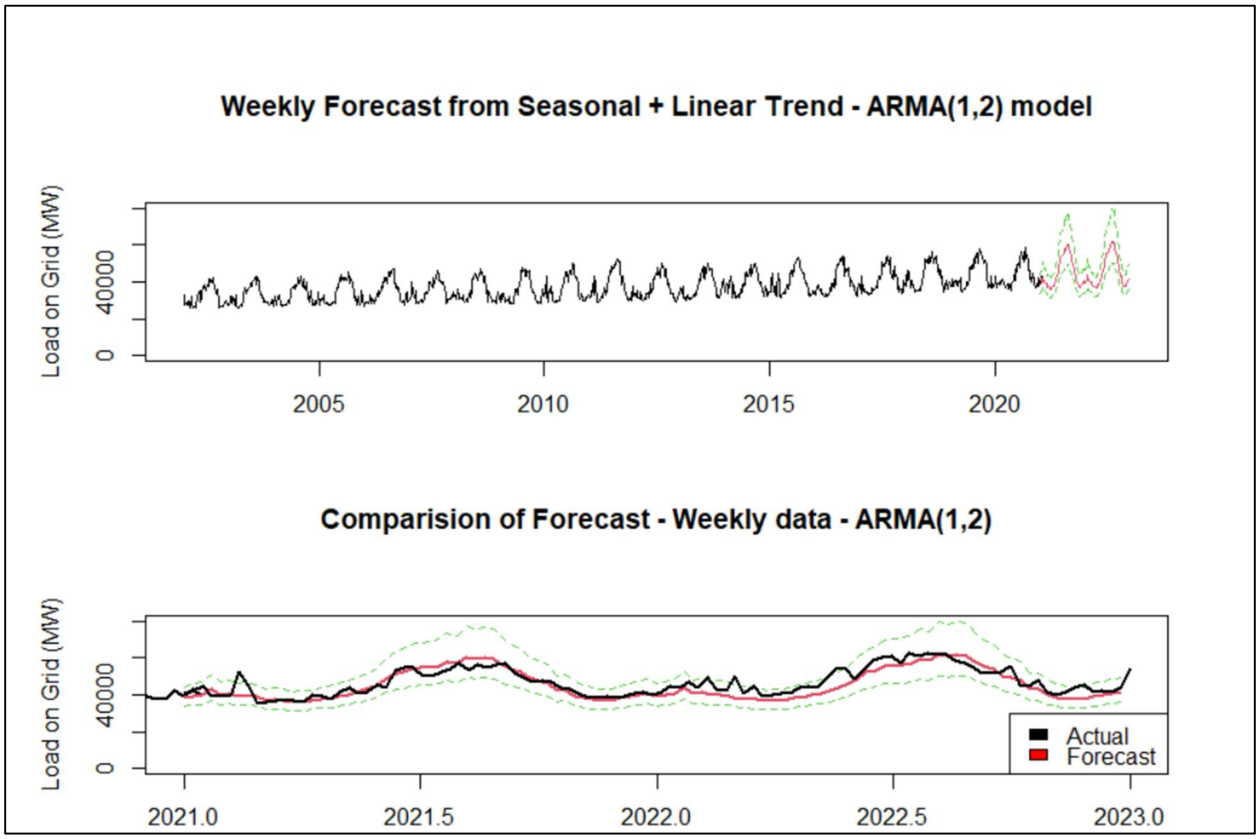


Figure 15. Forecasting of weekly data two years into the future with the 95% confidence interval in dashed green.

For weekly averaged data, ARMA (1, 2) with seasonal means and linear trend was used to forecast the values two years or 104 weeks into the future. As we also have the actual values for this period, we can plot the two curves together for clear visualization of the forecast accuracy. Figure 15 shows that the estimated value makes sense in context of the past data. When zoomed in and compared with the actual values, it is seen that except for a spike in the beginning almost all the actual values are within the 95% confidence interval set by the green-dashed lines. The results have a RMSE of 4435.86 MW and a MAPE of 7.25%.

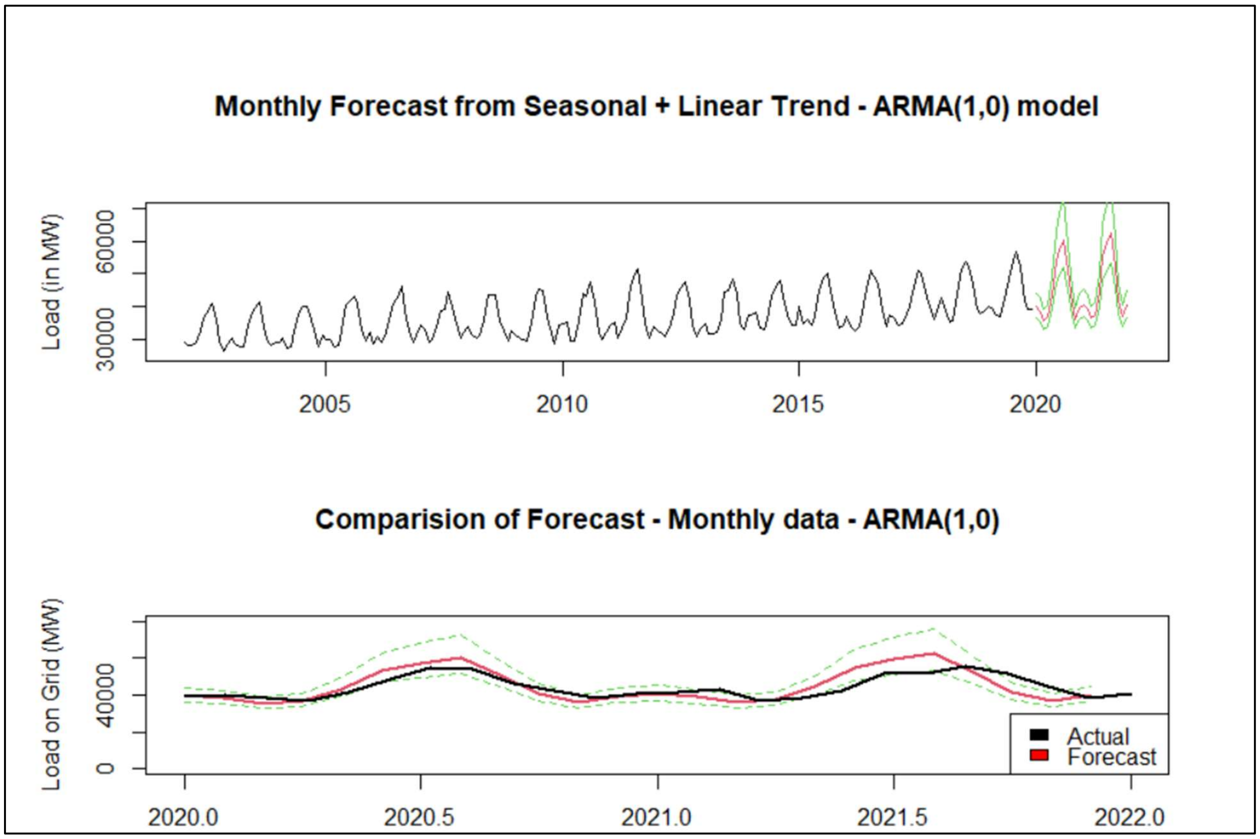


Figure 16. Forecasting of monthly data two years into the future with the 95% confidence interval in dashed green.

Figure 16 shows similar results for the monthly averaged data, which is forecasted by ARMA (1,0) using seasonal means and linear trend. The actual values begin to go outside the 95% confidence interval after 2021, which is expected as the prediction gets worse as time goes on. The actual value is first overestimated and then underestimated near the end of forecasting interval. The results have a RMSE of 3234.07 MW and a MAPE of 5.42%.

Conclusion

The weekly and monthly forecasts using the ERCOT dataset viewed at two different frequencies were obtained using deterministic trend modelling (seasonal means plus linear trend) and ARMA modelling of the detrended residuals. The forecasted results were successful in predicting the correct 95% confidence interval for at least the first year of prediction. That along with the fact that the model was able to achieve single-digit MAPE value is impressive given the simplicity of the methods involved.

The difference of viewing the data at different scales is evident when the results are zoomed in, in figures 15 and 16. Weekly frequency presents are much more erratic curve for actual value, while the monthly frequency actual values are pretty smooth and follow the prediction better. This suggests that the problem of weekly prediction is more difficult than the problem of month prediction. It becomes even more challenging to predict the data say on a daily or hourly frequency. Some literature (Marino et. al., 2016) has explored the use of deep neural networks for forecast at an hourly frequency and such models can give results with a MAPE of less than 1%. While the performance is impressive, methods involved use large datasets with high resolution and great computational power is expended in training these models. Additionally, neural networks are a black box solution to the problem which is hard to debug while training. The techniques applied in this project are perfect for modelling and understanding the data, possibly providing insights for preprocessing it before feeding it into a more complex Machine Learning architecture for prediction.

References

C. Mihai, I. Lepadat, E. Helerea and D. Călin, "Load curve analysis for an industrial consumer," 2010 12th International Conference on Optimization of Electrical and Electronic Equipment, Brasov, Romania, 2010, pp. 1275-1280, doi: 10.1109/OPTIM.2010.5510494.

Jia Li, Richard E. Just, Modeling household energy consumption and adoption of energy efficient technology, Energy Economics, Volume 72, 2018, Pages 404-415, ISSN 0140-9883, https://doi.org/10.1016/j.eneco.2018.04.019.

S. Maybee and N. D. Uri, "Time series forecasting of utility load duration curves," in Canadian Electrical Engineering Journal, vol. 4, no. 1, pp. 4-8, Jan. 1979, doi: 10.1109/CEEJ.1979.6592401.

D. L. Marino, K. Amarasinghe and M. Manic, "Building energy load forecasting using Deep Neural Networks," IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, Italy, 2016, pp. 7046-7051, doi: 10.1109/IECON.2016.7793413.

APPENDIX (R-Codes)

Weekly analysis

```
#necessary libraries
library(TSA)
library(tseries)
library(forecast)
library(MASS)
library(pracma)
#loading the data
weekly_data = read.csv("C:/Users/udayr/Documents/MA 5781/Final Project/ERCOT data/weekly_data.csv",
                        head = T)$ERCOT
#actual values to compare with forecasted values
actual = window(weekly_data, start = 992, end=1095)
#data for analysis


############################    Data   ########################################
series = window(weekly_data, end = 991)
#converting into time series data
data = ts(series, start=c(2002,1), freq=52)
#raising the plot so that xlab will be visible
par(mar = c(5, 4, 4, 2) + 0.1)
plot(data, xlab='Year(Weeks)', ylab='Load on grid',
     main = 'ERCOT load curve from 2002 to 2020', col='blue1')
grid(lwd = 0.5, lty = "dotted")
################################################################################


###################  check the normality of data   #############################
qqnorm(data);qqline(data)
hist(data, main='Histogram of Weekly Data', xlab='Load(MW)')
shapiro.test(data)
################################################################################

###########  Analysis(Transformation + Deterministic Modelling)     ############
#boxcox transformation
b = boxcox(lm(data~time(data)))
lambda = b$x[which.max(b$y)]
lambda

#Data after box-cox transformation
bc_data <- (data^lambda - 1) / lambda
plot(bc_data, main = bquote(paste(
  'Box-Cox Transformation with lamba: ', .(lambda), sep="")),
     ylab='Box-Cox Data')

#checking normality of box-cox data
hist(bc_data, main='Histogram of Box-Cox Data', xlab = 'Box-Cox Data')
shapiro.test(bc_data)

#Applying the Deterministic Trend Modelling
#seasonal means model + Linear Trend
month=season(bc_data)
model1=lm(bc_data~month+time(bc_data))
summary(model1)
#residuals
weekly_resid= resid(model1)

#converting residuals into timeseries
res = ts(weekly_resid
        , start=c(2002,1), freq=52)
#plotting residuals
plot(res, main='Residuals of Seasonal + Linear Model for Weekly Data',
     ylab = 'Residuals', xlab= 'Years')
```

```
################################################################################

#####################   Selection of Candidate Models ########################
#ACF plot of residuals
acf(weekly_resid, lag.max = 60, main= 'ACF plot of Residuals') #MA(8)
#PACF of residuals
pacf(weekly_resid, lag.max = 60, main= 'PACF plot of Residuals') #AR(1) AR(3)

#Unit root Tests
adf.test(resid_linear)
pp.test(resid_linear)
kpss.test(resid_linear)


#EACF plot of residuals
eacf(weekly_resid) # MA(8), ARMA(1,2), ARMA(2,3), ARMA(3,3)
auto.arima(weekly_resid) #AR(4)
res=armasubsets(y=rsd4,nar=10,nma=10,y.name='test',ar.method='ols')
par(mfrow=c(1,1))
plot(res)      # default is BIC
#ARMA(3,3)
#AR(4)
#ARMA(4,4)
plot(res,scale='AIC')#ARMA(9,8), ARMA(4,4)
plot(res,scale='AICc')#ARMA(9,8), ARMA(4,4)

#checking AIC, BIC, AICc score
ma8 = Arima(res, order = c(0,0,8), include.mean=F)
ar4 = Arima(res, order = c(4,0,0), include.mean=F)
arma33 = Arima(res, order = c(3,0,3), include.mean=F)
arma44 = Arima(res, order = c(4,0,4), include.mean=F)
#Selected AR(4) as a final model. Now, checking for Overfit
arma41 = Arima(weekly_resid, order = c(4,0,1), include.mean=F)
ar5 = Arima(weekly_resid, order = c(5,0,0), include.mean=F)
ma8
ar4  #We conclude this as it has best AIC and BIC scores
arma33
arma44
arma41
ar5
#Diagnostic plot of AR(4)
tsdiag(ar4)
################################################################################


################################    Forecast  ##################################
#forecasting for 2 years using ARMA(1,2)
arma12_xreg=Arima(bc_data,order=c(1,0,2),include.mean=F,xreg=model.matrix(model1))
pred_years = seq(from=2021,to=2023,length=104)
newpreddata=data.frame(month=as.factor(month[1:104]),tm=pred_years)
predx=predict(arma12_xreg,n.ahead=104,newxreg=
                cbind(1,rbind(diag(1,52)[,-1],diag(1,52)[,-1]),newpreddata[,2]))
pr=predx$pred
#95% Confidence Interval
uci=pr+2*predx$se
lci=pr-2*predx$se
#Converting prediction values and 95% CI into time series data
pr=ts(pr,start=2021,freq=52)
uci=ts(uci,start=2021,freq=52)
lci=ts(lci,start=2021,freq=52)

#calculating RMSE and MAPE
library(Metrics)
rmse(actual, InvBoxCox(pr, lambda = lambda))
mean(abs((actual- InvBoxCox(pr, lambda = lambda))/actual))*100
```

```
#Actual values which were kept aside to compare with predicted values
actual = ts(actual, start = c(2021,1), freq = 52)


#prediction on the original scale
par(mfrow=c(2,1))
plot(InvBoxCox(bc_data, lambda = lambda), ylab = "Load on Grid (MW)",
     main = "Weekly Forecast from Seasonal + Linear Trend - ARMA(1,2) model",
     xlim = c(2002, 2023), ylim=c(0, 80000))
lines(InvBoxCox(pr, lambda = lambda),col=2)
lines(InvBoxCox(uci, lambda = lambda),col=3, lty=2)
lines(InvBoxCox(lci, lambda = lambda),col=3, lty=2)


#Zoomed plot of forecast while comparing with actual values
plot(InvBoxCox(bc_data, lambda = lambda), ylab = "Load on Grid (MW)",
     main = "Comparision of Forecast - Weekly data - ARMA(1,2)",
     xlim = c(2021, 2023), ylim=c(0, 80000),lwd=2)
lines(InvBoxCox(pr, lambda = lambda),col=2, lwd=2)
lines(InvBoxCox(uci, lambda = lambda),col=3, lty=2)
lines(InvBoxCox(lci, lambda = lambda),col=3, lty=2)
lines(x=seq(from=2021,to=2023,length=104),y=actual,pch=3, lwd=2)
legend(x = "bottomright", legend=c("Actual", "Forecast"),
       fill = c("black","red", cex=0.7)
)

par(mfrow=c(1,1))
################################################################################
```

Monthly Analysis

```
library(TSA)
library(tseries)
library(forecast)
library(MASS)
library(pracma)

monthly_data = read.csv("C:/Users/udayr/Documents/MA 5781/Final Project/ERCOT
data/monthly_data.csv",head=T)
data = ts(monthly_data$ERCOT[1:216], start=c(2002,1), freq=12) # Monthly frequency

# Fit the model on data from 2002 to 2020 - 1:228
# Use remaining data to compare the forecast - 229:252

plot(data,lwd=2.0, xlab="Years", ylab="Load on Grid (MW)", main="ERCOT Load Curve data (Monthly
frequency) from 2002 to 2020")

hist(data,xlab="Load (MW)", main="Histogram - monthly raw data", breaks = 20)

qqnorm(data,main="QQ plot - Monthly")
qqline(data)


shapiro.test(data)

# Distribution seems right skewed

# Attempting transforms - Box-Cox
# Parameter estimation for using Box-Cox transformation
b = boxcox(lm(data~time(data)))
# Exact lambda
lambda = b$x[which.max(b$y)]
lambda
# Estimated parameter is somewhere near -1.23, which is around -1 corresponding to 1/x
tx_data = (data ^ lambda - 1) / lambda
```

```
plot(tx_data,lwd=2.0, xlab="Years", ylab="[(MW^lambda) - 1] / lambda", main="Monthly transformed with
lamda = -1.23")

hist(tx_data,xlab="Transformed Load", main="Histogram - monthly transformed data", breaks = 20)

shapiro.test(tx_data)


par(mfrow=c(3,1))
plot(tx_data,lwd=2.0, xlab="Years", ylab="[(MW^lambda) - 1] / lambda", main="Box-Cox tranformation with
lamda = -1.23")
hist(tx_data, xlab="Load (MW)", main="Histogram after transform - monthly raw data", breaks = 20)
qqnorm(tx_data,main="QQ plot of Box-Cox transform ")
qqline(tx_data)
par(mfrow=c(1,1))

shapiro.test(tx_data)
# Test says non-normal but at least the skew of the data has been corrected.

# Seasonal means plus linear trend on transformed series ---------------------------------------------
---------------------------------
tm = time(tx_data)
trend = lm(tx_data~season(tx_data) + tm)
summary(trend)
monthly_resid = ts(resid(trend))

par(mfrow=c(3,1))
plot(monthly_resid,lwd=2.0, ylab="Residuals",xlab="Years", main = sprintf("Residuals of Seasonal +
Linear model - Monthly frequency"))
acf(monthly_resid,lwd=2.0, main="ACF of residuals", lag.max = 100)
pacf(monthly_resid,lwd=2.0, main="PACF of residuals", lag.max = 100)
par(mfrow=c(1,1))
# Maybe MA(2) or AR(1) models based on acf and pacf

par(mfrow=c(1,2))
hist(monthly_resid, main="Histogram - Seasonal + Linear")
qqnorm(monthly_resid,main="QQ plot - Seasonal + Linear")
qqline(monthly_resid)
par(mfrow=c(1,1))

adf.test(monthly_resid)
pp.test(monthly_resid)
kpss.test(monthly_resid)
# No unit roots found

auto.arima(monthly_resid)
# Suggests ARIMA(0,0,0) with zero mean.

eacf(monthly_resid, ar.max = 13)
# Vertex found at --- ARMA(4,0), ARMA(4,2)

res=armasubsets(y=monthly_resid,nar=10,nma=10,y.name='test',ar.method='ols')
par(mfrow=c(1,1))
plot(res)      # default is BIC
# BIC rankings ---
# ARMA(1,0)
# ARMA(0,10)

plot(res,scale='AIC')
plot(res,scale='AICc')
# Both AIC and AICc rankings ---
# ARMA(4,10)
# ARMA(1,0)
par(mfrow=c(1,1))

# Using AR(1)
```

```
month = season(tx_data)
arma10_xreg=Arima(tx_data, order=c(1,0,0),include.mean=F,xreg=model.matrix(trend))
pred_years = seq(from=2020,to=2021.917,length=24)
newpreddata=data.frame(month=as.factor(month[1:12]),tm=pred_years)
predx=predict(arma10_xreg,n.ahead=24,newxreg=cbind(1,rbind(diag(1,12)[,-1],diag(1,12)[,-
1]),newpreddata[,2]))
pr=predx$pred
uci=pr+2*predx$se
lci=pr-2*predx$se

pr=ts(pr,start=2020,freq=12)
uci=ts(uci,start=2020,freq=12)
lci=ts(lci,start=2020,freq=12)

plot(tx_data,xlim=c(2002,2022), xlab="Years", main="Transformed Predictions - ARMA(1,0)")
lines(pr,col=2)
lines(uci,col=3)
lines(lci,col=3)


# Box test
Box.test(resid(arma10_xreg))

# Diagnostic plots
tsdiag(arma10_xreg)

# Checking overfit for AR(1) -- AR(2) and ARMA(1,1)
arma10_xreg=Arima(tx_data, order=c(1,0,0),include.mean=F,xreg=model.matrix(trend))
arma20_xreg=Arima(tx_data, order=c(2,0,0),include.mean=F,xreg=model.matrix(trend))
arma11_xreg=Arima(tx_data, order=c(1,0,1),include.mean=F,xreg=model.matrix(trend))

arma10_xreg
arma20_xreg
arma11_xreg

actual = ts(monthly_data$ERCOT[217:240], start = c(2020,1), freq = 12)


par(mfrow=c(2,1))

plot(InvBoxCox(tx_data,lambda = lambda), ylim=c(25000,70000), ylab="Load (in MW)", xlab="Years",
main="Monthly Forecast from Seasonal + Linear Trend - ARMA(1,0) model",xlim=c(2002,2022))
lines(InvBoxCox(pr,lambda = lambda),col=2)
lines(InvBoxCox(uci,lambda = lambda),col=3)
lines(InvBoxCox(lci,lambda = lambda),col=3)

plot(InvBoxCox(tx_data, lambda = lambda), ylab = "Load on Grid (MW)",
     main = "Comparision of Forecast - Monthly data - ARMA(1,0)",
     xlim = c(2020, 2022), ylim=c(0, 80000))
lines(InvBoxCox(pr, lambda = lambda),col=2, lwd='2.0')
lines(InvBoxCox(uci, lambda = lambda),col=3, lty=2)
lines(InvBoxCox(lci, lambda = lambda),col=3, lty=2)
lines(x=seq(from=2020,to=2022,length=24),y=actual,pch=3,lwd='2.0')
legend(x = "bottomright", legend=c("Actual", "Forecast"),
       fill = c("black","red")
)
par(mfrow=c(1,1))

# Getting the rmse and mape values
library(Metrics)
rmse(actual, InvBoxCox(pr, lambda = lambda))
mean(abs((actual- InvBoxCox(pr, lambda = lambda))/actual)) * 100
```