1. Introduction

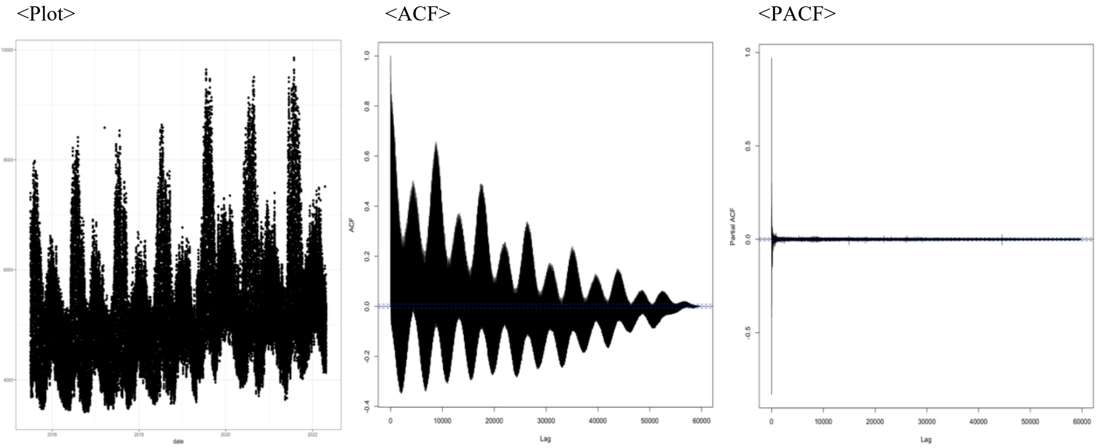
Forecasting of power demand plays an essential role in the electric industry, as it provides the basis for making decisions in power system planning and operation. Accurate forecasts of electricity demand inform investment decisions about power generation and supporting network infrastructure. Of major interest to energy policymakers, power utilities, and private investors alike, forecasts are also essential for development professionals. Inaccurate forecasts, whether they over- or under predict demand, can have dire social and economic consequences. Underestimating demand results in supply shortages and forced power outages, with serious consequences for productivity and economic growth. Overestimating demand can lead to overinvestment in generation capacity, possible financial distress, and, ultimately, higher electricity prices. In line with this importance of forecasting of power demand, this project aims to understand the overall electricity demand data and find the best model to predict the future data, specifically electric energy demand in Colorado on Sunday May 1st from 5-6 PM.

2. Dataset

In mid-2015, the EIA began collecting hourly energy demand data across the contiguous United States. The Energy demand data are freely available and constantly updated by the government EIA website. The data for this project are imported from the EIA website using the EIA R package. It provides utilities to convert dates and provides tools to update a dataset whenever new data is available. Once the data are imported, I go over the dataset for data preprocessing. I use IQR (Interquartile Range) rule to find outliers. I multiply the interquartile range (IQR) by 1.5 and add it to the third quartile. Any number greater than this is a suspected outlier. Likewise, subtract 1.5 x IQR value from the first quartile, and any number smaller than this is a suspected outlier as well. After the outliers are identified, I replace them with the mean of the previous and the next value.

3. Methodology

Once data preprocessing is done, I plot the dataset (x = date, y = demand) as well as ACF (Auto Correlation Function) and PACF (Partial ACF). The plots show strong seasonality, and ACF decays slowly, and this means that the time series is not stationary.



<Plot> <ACF> <PACF>

Figure.1. Plots

The fact that ACF and PACF decay exponentially as lag gets larger suggests that ARMA model to be selected. Since the time series is non-stationary and it has seasonality in it, I decide to use SARIMA model. In addition, the time series has multiple seasonality, daily seasonality, weekly seasonality, and annual seasonality as indicated in the Figure.2. All this seasonality should be considered, when we choose the model, so the msts() function is selected to make the time series with multiple seasonality, daily(24), weekly(24*7), and yearly(24 * 7 * 52), and use auto.arima() function to find the best model.

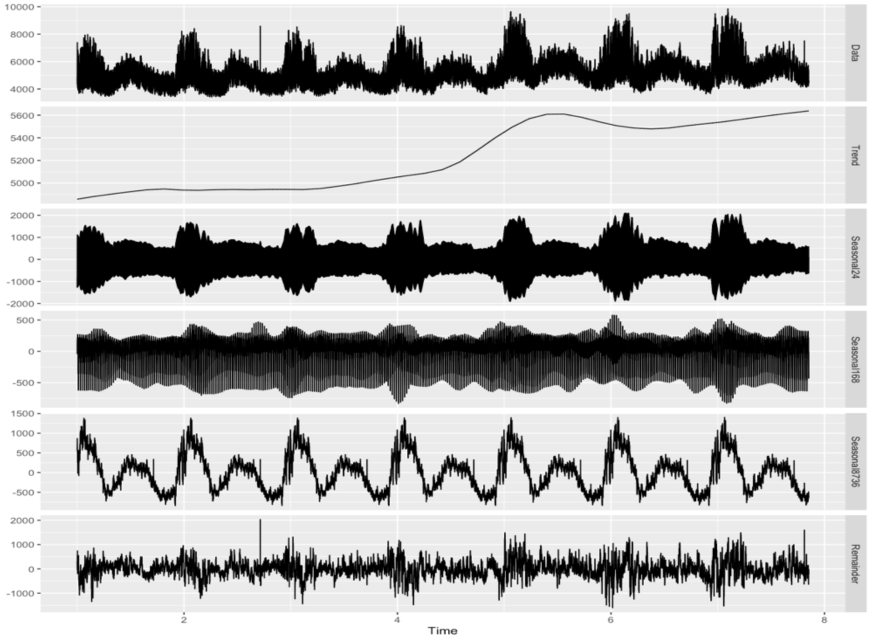


Figure.2. Multiple Seasonality

However, it takes significant amount of time to find the best model using auto.arima() with annual seasonality. So, considering the value we are predicting is only less than 36 lags away from the present and the trend is not very significant in this dataset, I decide not to consider the annual seasonality, but consider only daily and weekly seasonality to predict the electric energy demand in Colorado on Sunday May 1st from 5-6 PM.

I split the data, the first 80% of the dataset as training dataset and the rest and the latest 20% of the dataset as test dataset. First, I use auto.arima() with the training dataset to find the best model, and it returns SARIMA(3,0,0)(0,1,0)[168] model. I test 'test data' using this model to find out whether this model fits the dataset. However, since the model doesn't consider the annual seasonality, it doesn't fit the model very well. Secondly, I use auto.arima() with the test data just to see the difference, but it returns the exact same SARIMA model. This does not fit either as expected, because it also doesn't consider the annual seasonality. So, to avoid the impact caused by the annual seasonality, I consider only the 1,000 recent data. I think 1,000 data is big enough to predict the next 36 (29 to be exact) data and it is also small enough to ignore the trend effect as well as the impact due to the annual seasonality. I use auto.arima() and the result is SARIMA(2,1,1)(0,1,0)[168].

4. Result and Discussion

The result using SARIMA(2,1,1,)(0,1,0)[168] is summarized as below figure.3. ACF plot shows that the residuals are white noise, and it is also normally distributed, and all the p values of Ljung-Box statistic are greater than 0.05, which suggests that we fail to reject the null hypothesis that the residuals are independently distributed.

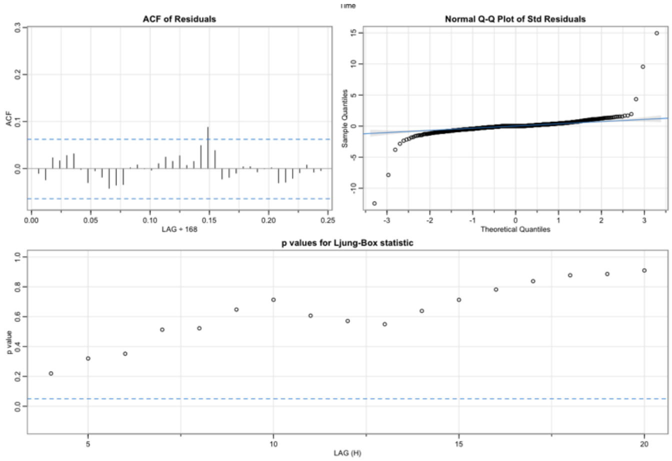


Figure.3. Result of SARIMA(2,1,1)(0,1,0)[168] Model

After I find that the model fits the data well, I used sarima.for() function to predict the next values, and the result is in the Figure.4 and the table.1 as below.

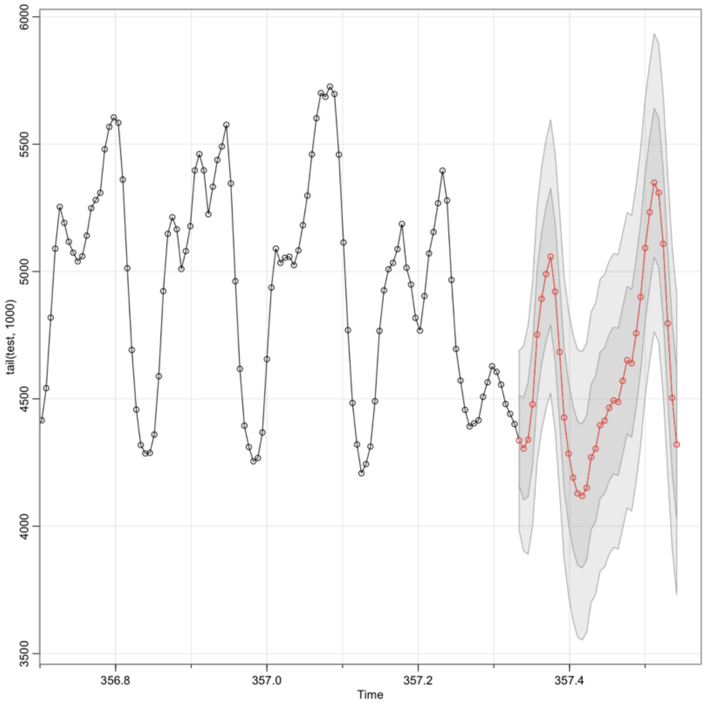


Figure.4. Prediction Result

UTC	MST	MWh (Megawatt hour)
5/1/2022 22:00:00	5/1/2022 16:00:00	4757.26
5/1/2022 23:00:00	5/1/2022 17:00:00	4900.44
5/2/2022 00:00:00	5/1/2022 18:00:00	**5092.59**
5/2/2022 01:00:00	5/1/2022 19:00:00	5232.71
5/2/2022 02:00:00	5/1/2022 20:00:00	5348.82

Table.1. Electric energy demand in Colorado on Sunday May 1st from 5-6 PM

Using SARIMA(2,1,1)(0,1,0)[168] model, the Electric Energy Demand in Colorado on Sunday May 1st from 5-6 PM is expected to be 5092.59 MWh.

There is a limitation to be highlighted. As briefly mentioned in the section 3. Methodology, the dataset has a very clear annual seasonality, but it is not considered due to the significant amount of time auto.arima() function takes. However, as indicated in PACF plot, the past values from the previous years are not strongly correlated to the recent values and especially to the next 36 future values. In addition, the SARIMA(2,1,1)(0,1,0) [168] model shows that the residuals are white noise, normally distributed and independent, and I consider this shows that the model fits the dataset well. Thus, I decide to consider only the recent 1,000 values for the prediction.