Forecasting of office temperature

In this project, we would be investigating the temperature of an office space and extrapolate the temperature of the office over the next 50 hours. This will be useful in regulating office space air conditioner usage and ultimately becoming energy efficient. Also, no one likes a *freezing* morning or a *sweaty* afternoon!

Data

This dataset contains datapoints on temperature of every minute over a week period. Below is sample df.head( ).

Date	Temperature
2015-02-11 14:48:00	21.7600
2015-02-11 14:49:00	21.7900
2015-02-11 14:50:00	21.7675
2015-02-11 14:51:00	21.7675
2015-02-11 14:51:59	21.7900

Firstly, we resample our data to be able to display the trend and seasonality while minimizing the noise in the data.

We would resample over **days**, **hourly** and **every 30 mins**.

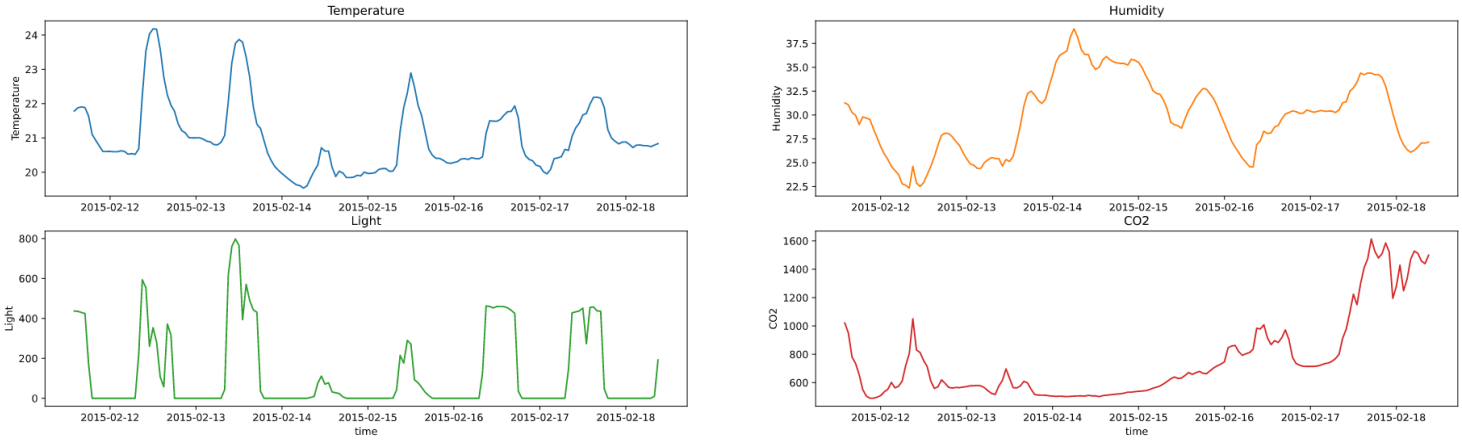
```
df_day = df.resample('D').mean()
df_hourly = df.resample('H').mean()
df_thirty_mins = df.resample('30T').mean()
```

Resampling of time series data

Resample daily

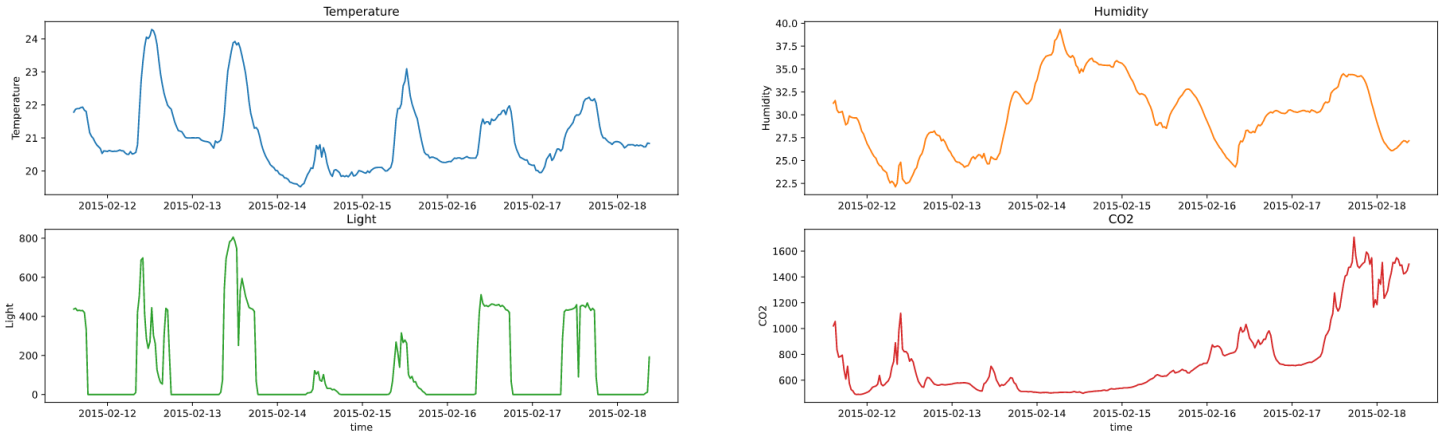
Data is not representative of seasonality within each day; thus, we would use a different resample metrics.

Resample hourly



Hourly resample is able to display the seasonality and also minimize most of the noise in the data.

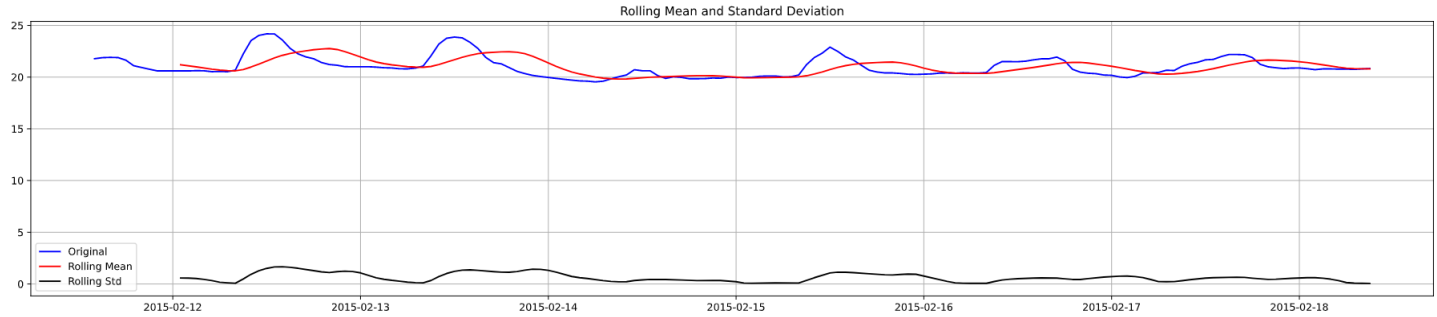
Resample every 30 minutes



30 minutes resample is able to display the seasonality and however we observe slightly more noise in the data.

In this prediction, we would be using an hourly resample data.

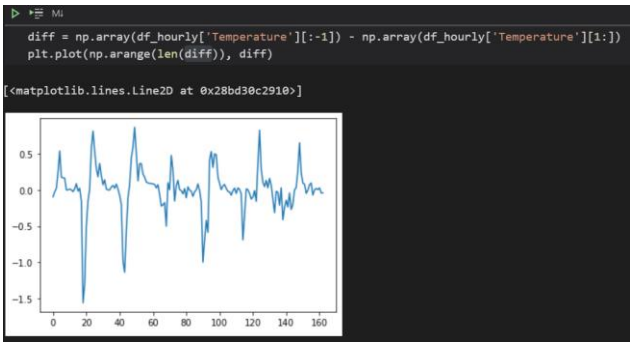
Observing rolling mean and std



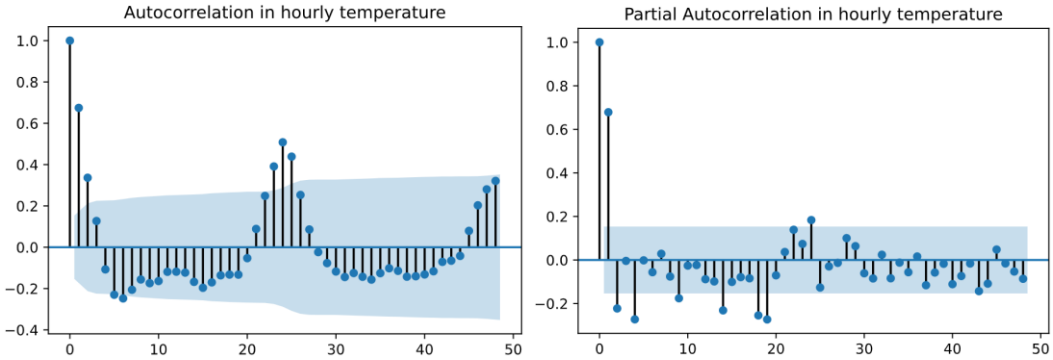
We can see some fluctuating variance in the data across the day and seasonality across the day. Let us do some transformation to make sure the timeseries is stationary.

Transformation of timeseries

We use the difference between the original temperature time and 1 lagged temperature. We achieve the below time series. Using the Augmented Dickey-Fuller test, the series is proven to be stationary with p-value = 3.76e-9.



Plotting acf and pacf

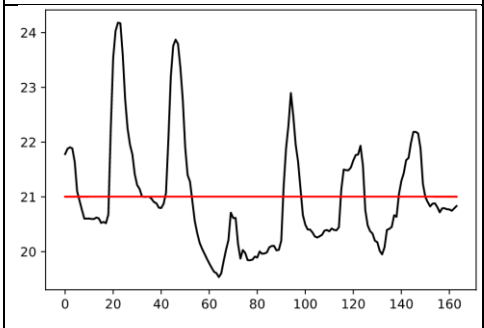
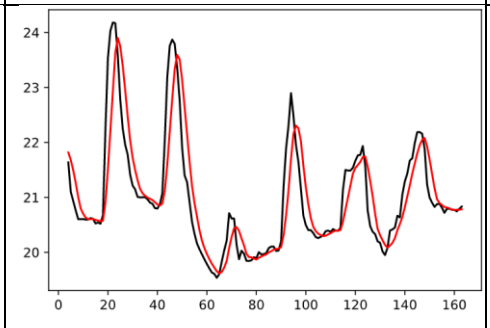
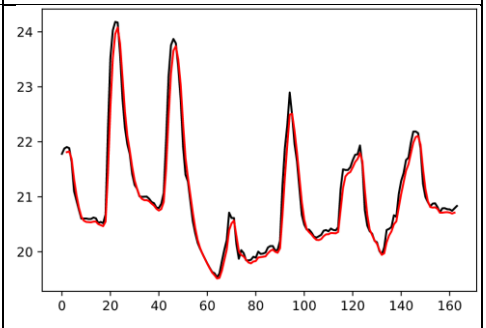


Based on the acf plot, we observe 3 statistically significant points, thus we can experiment with ar(3).

In the pacf plot, we observe 2 statistically significant points, thus we can experiment with ma(2).

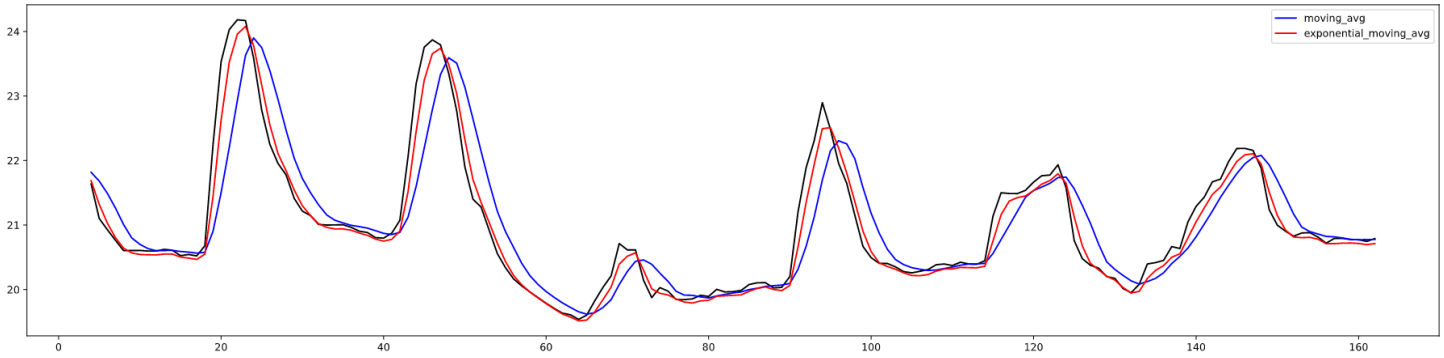
Smoothing

Before we go ahead and test our models, this segment is to showcase the fit of 3 types of smoothing.

Simple smoothing	Moving Average Smoothing	Exponential Smoothing
		
MSE: 167.63	MSE: 46.25	MSE: 6.42

* Red line is prediction

Combined model



Forecasting

The data was split into train and test data to estimate the effectiveness of the model in prediction of this data.

The initial 114 time period temperature is part of the train dataset, while the latter 50 time period temperature is part of the test dataset.

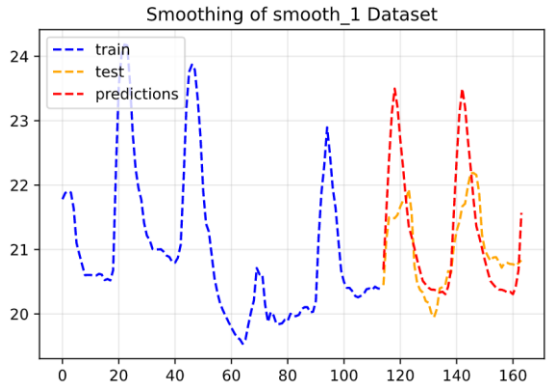
We would be using 4 different model to test the effectiveness – **Exponential Smoothing, ARMA, ARIMA** and **SARIMAX**.

```
test_size = 50
train = df_hourly_arr[:-test_size]
test = df_hourly_arr[-test_size:]
print('Train shape:',train.shape,'\n')
print('Test shape:',test.shape)
```

```
Train shape: (114,)

Test shape: (50,)
```

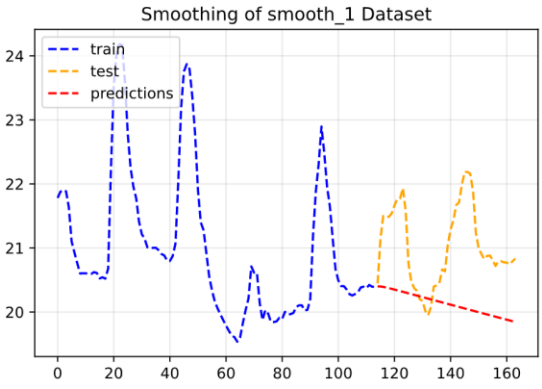
Exponential smoothing



MSE: 27.37

We observe that the rapid increase in temperature exceeds the actual temperature. We can note that this is due to past data tend to favor sharper increase in temperatures but overall captures the trend very closely.

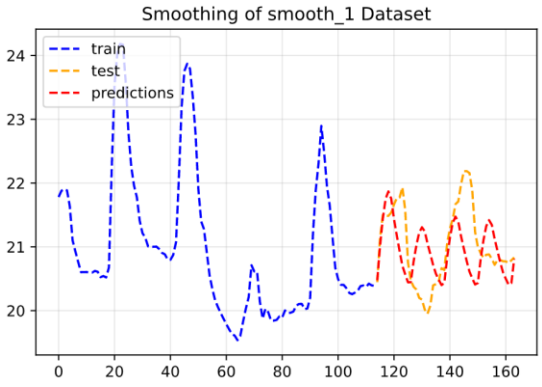
ARMA



MSE: 59.66

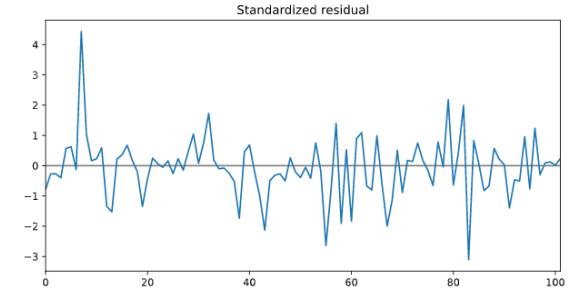
We use AR(3) and MA(2) as suggested above, however the predictions are really awful with a MSE of 59.66 and a relatively straight-line prediction despite the clear seasonality seen. This is likely since our dataset has a bigger number of datapoints and ARMA seem to only consider the nearest few datapoints. Hence ARMA seems to be better suited for short term predictions instead (i.e: 3 time periods ahead).

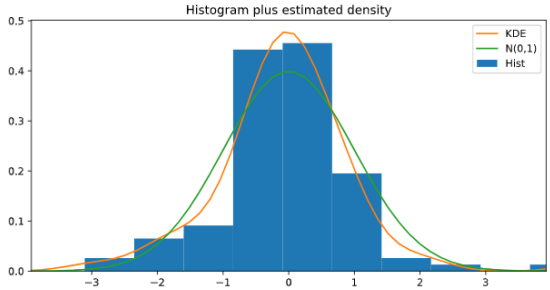
Auto ARIMA

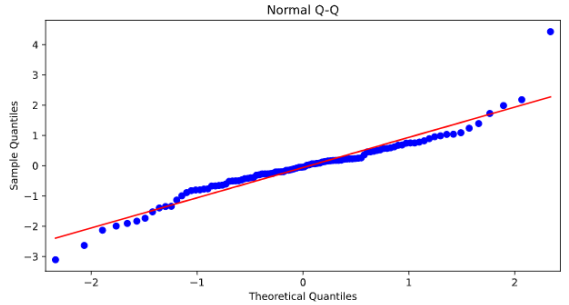


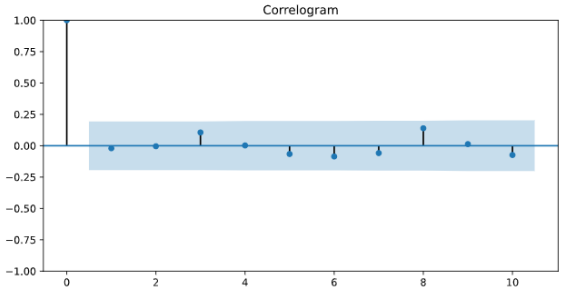
MSE: 21.61, Best model: ARIMA(3,0,1)(1,1,1)[12]

The auto ARIMA models seems to be able to capture seasonality much better than the ARMA model, however the time period for each seasonality cycle can be better optimized – current auto ARIMA model seasonality cycle is almost double the actual.



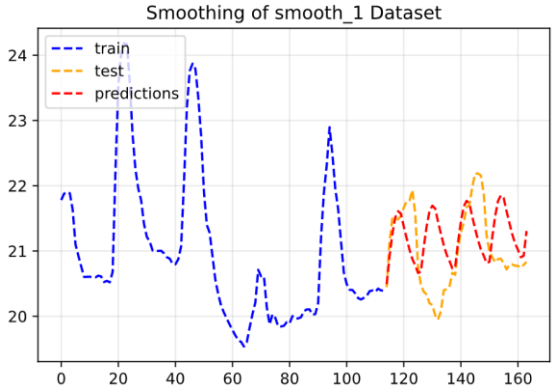






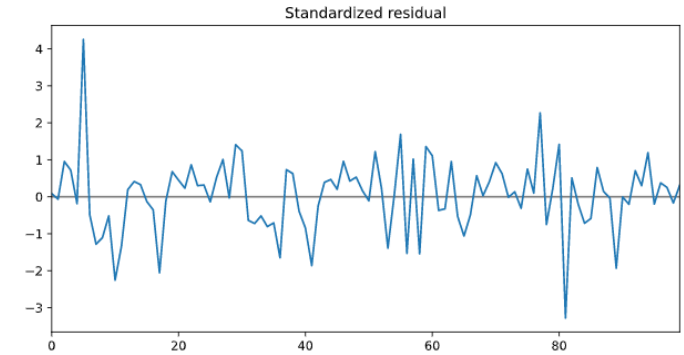
The residuals are random, which is a good sign that the trend and seasonality is able to captured well in the model. QQ plot is decent, however slight deviation of predictions nearing the tail ends – extreme values of temperature are not as consistently captured. The residual vs fitted plot has a clear normal distribution, indicating normally distributed residual across.

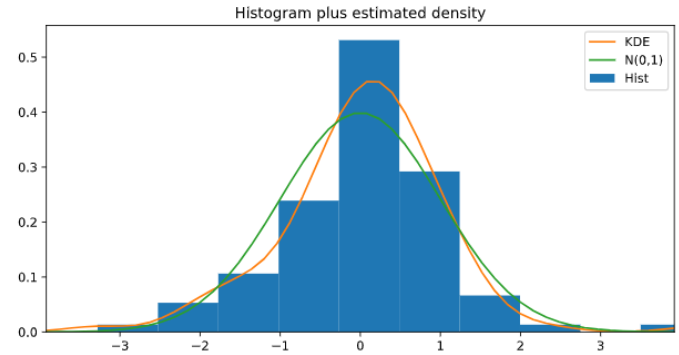
Auto SARIMAX

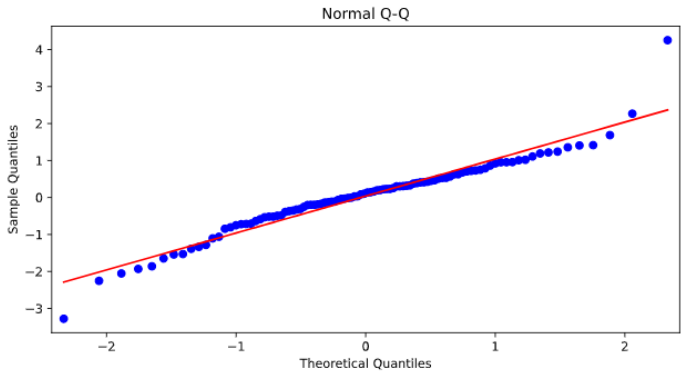


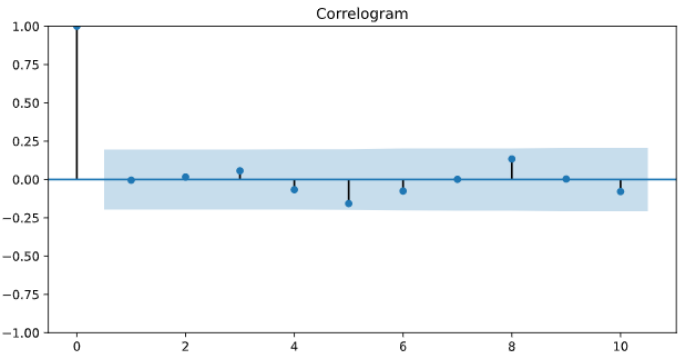
MSE: 25.67

Similarly, auto SARIMAX is able to predict decently, with similar results to ARIMA. However, with a worse MSE value.









Overall, very similar to auto ARIMA.

Best model

Based on MSE value, we would have preference for auto ARIMA due to the lowest MSE of 21.61. However, I would have a greater preference for exponential smoothing since exponential smoothing model is able to recognize the trend and seasonality involved, once optimized can better predict with a lower MSE than auto ARIMA which generally fails to recognize the trend.

Improvements

If possible, I would definitely want to try out the LSTM model with a better computer. However, with my limited hardware, I would try to optimize the exponential smoothing model to be able to recognize the magnitude of the seasonality using hyper parameter tuning with GridSearchCV.

Also, I would want to incorporate a regression model to predict temperature using other features such as **humidity, light intensity, occupancy levels** and **various other common variables** in a regular office space. This would give more flexibility to the model and account for the varying changes across the day (e.g: rainy day, company half day, Company-wide gathering).

Key finding

In this project, we explored 4 different models – namely Exponential smoothing, ARMA, Auto ARIMA and Auto SARIMAX.

Despite the simplicity of **exponential smoothing** as compared to other models, it is best able to capture trends. In the case of an office space temperature, this would be of more value as we would want our air condition to work harder when it is hotter and stop when it is colder!

ARMA has been observed to struggle with extrapolation into many periods into the future. Most likely that it is not optimized, but it is a poor model in this project as it too nearsighted.

Auto ARIMA and **auto SARIMAX** seems to capture the same trend with a slight difference and have amazing MSE values. However, it seems to capture the seasonality frequency wrongly and thus not as impactful. If implemented into our air conditioner regulating system, we would expect fluctuating office temperature across a short period of time.

Ultimately, these models can ensure we have a *cozy* time in the office with a *stationary* temperature!

Happy new year and I wish you clean data through the year 😊