Time Series Forecasting for Appliances Energy Prediction

Maleesha Hettiarachchi

4.18.2025

Table of Contents

Introduction

Energy is a vital element for sustainable development and economic growth. Efficient monitoring and forecasting of energy consumption allows better planning, optimization, and management of resources—especially in residential and commercial buildings.

This project focuses on predicting appliance energy consumption using a multivariate time-series dataset that records various environmental and temporal features at 10-minute intervals. The goal is to build a deep learning-based supervised model capable of learning patterns and dependencies from these inputs to forecast future energy usage.

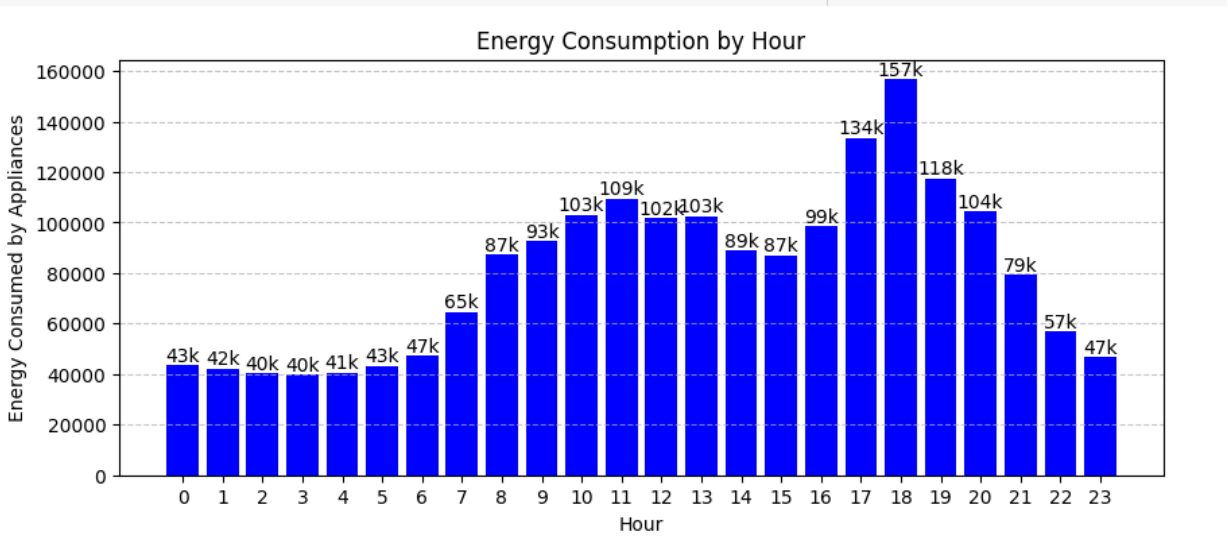
The implementation began with thorough data preprocessing and exploration. We confirmed the dataset contained no null values, and features such as temperature, humidity, and weather-related factors were analyzed through visualization techniques including line plots, box plots, and correlation heatmaps.

Time-based features like hour, day of week, and month were engineered to capture seasonal and daily patterns. Feature scaling using MinMaxScaler was applied to normalize values across different ranges. The dataset was then transformed into sequences suitable for time-series modeling.

Multiple forecasting models were implemented, including ARIMA, LSTM, Random Forest, and Gradient Boosting. Each model was trained, validated, and evaluated using metrics such as RMSE and MAE. Visualizations comparing actual and predicted values were used to interpret and assess model performance.

Data Insights

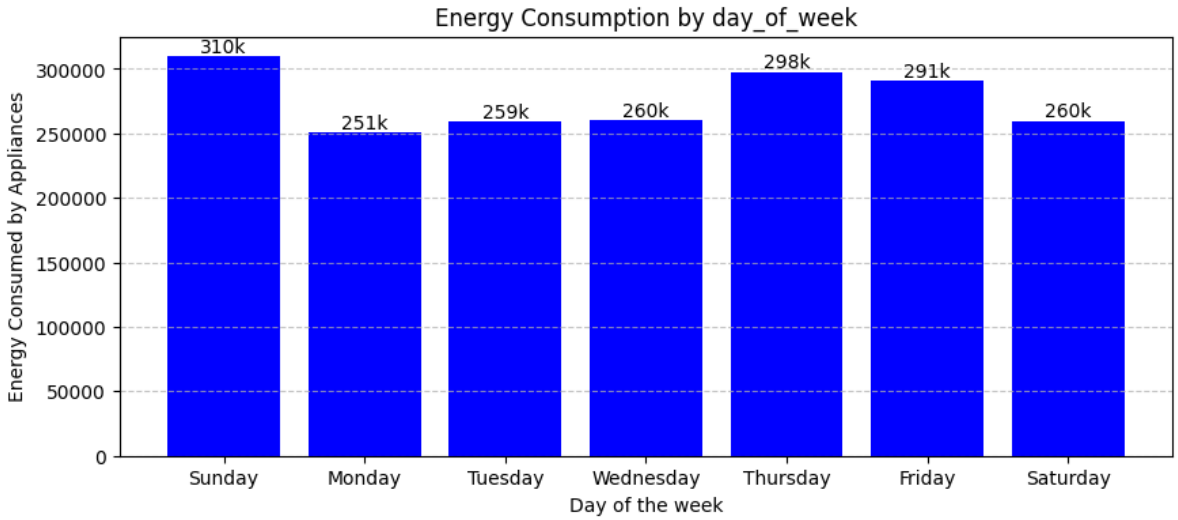
Hourly distribution of target variable



Observations:

- Highest electricity consumption occurs between 17:00 pm and 20:00 pm.
- Lowest electricity consumption is observed between 23:00 pm and 06:00 pm.

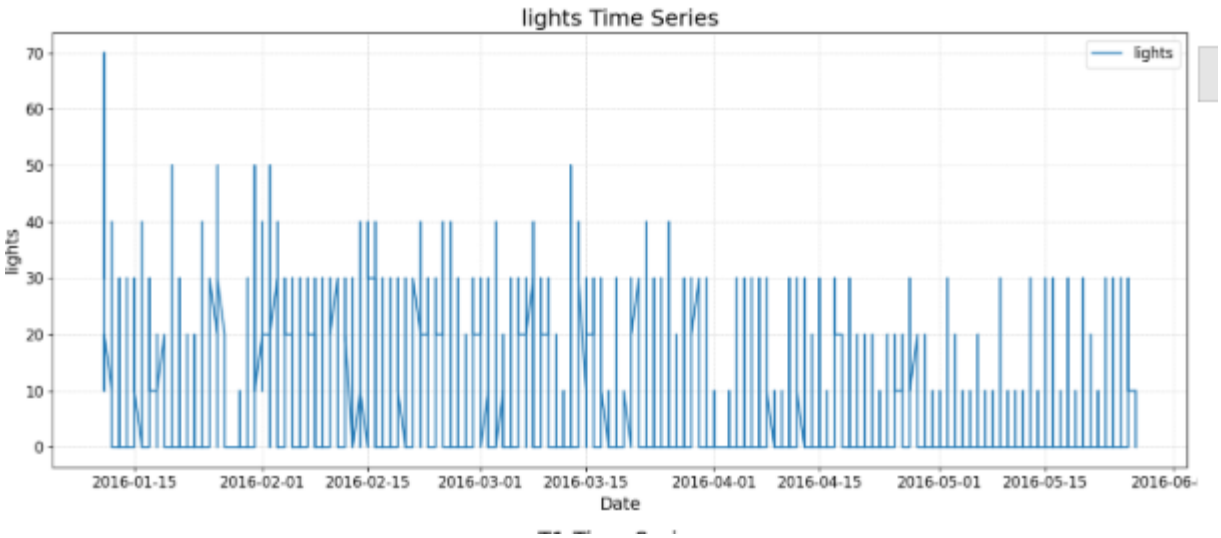
Weekday distribution of target variable

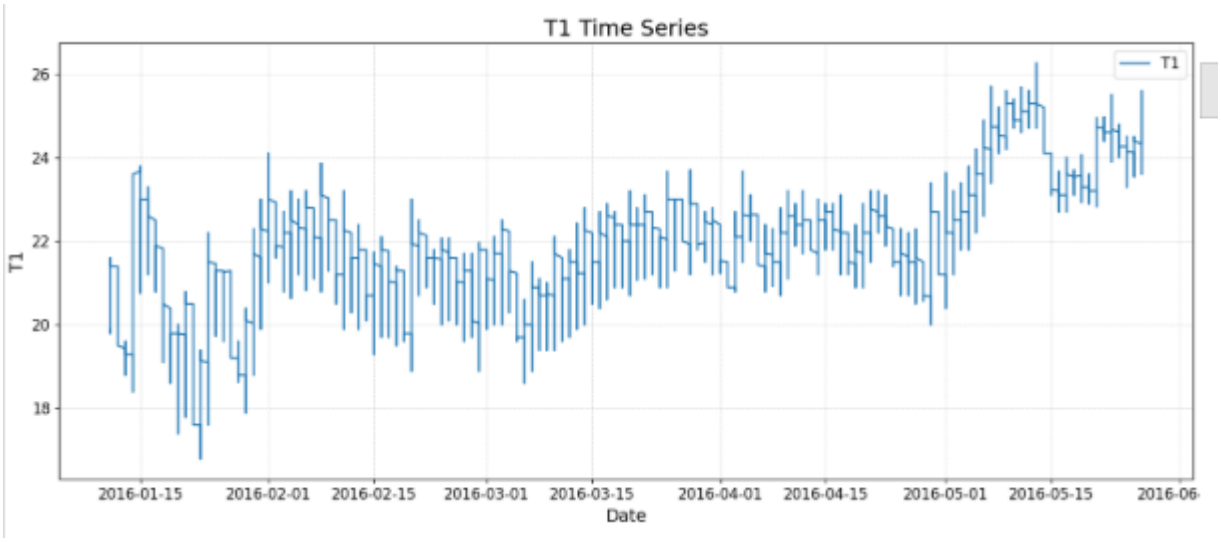


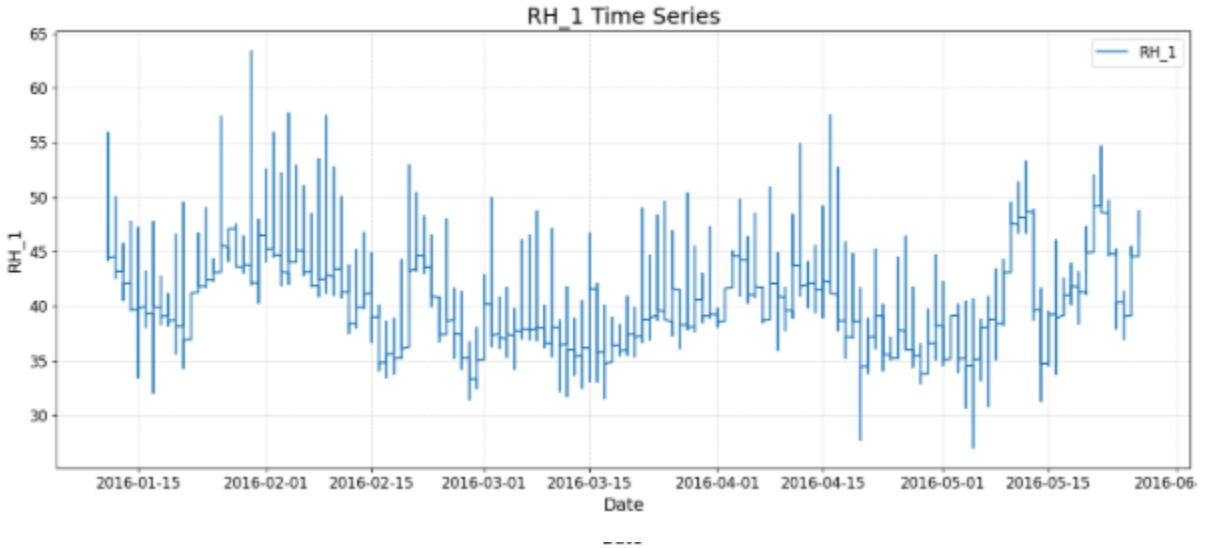
Observation:

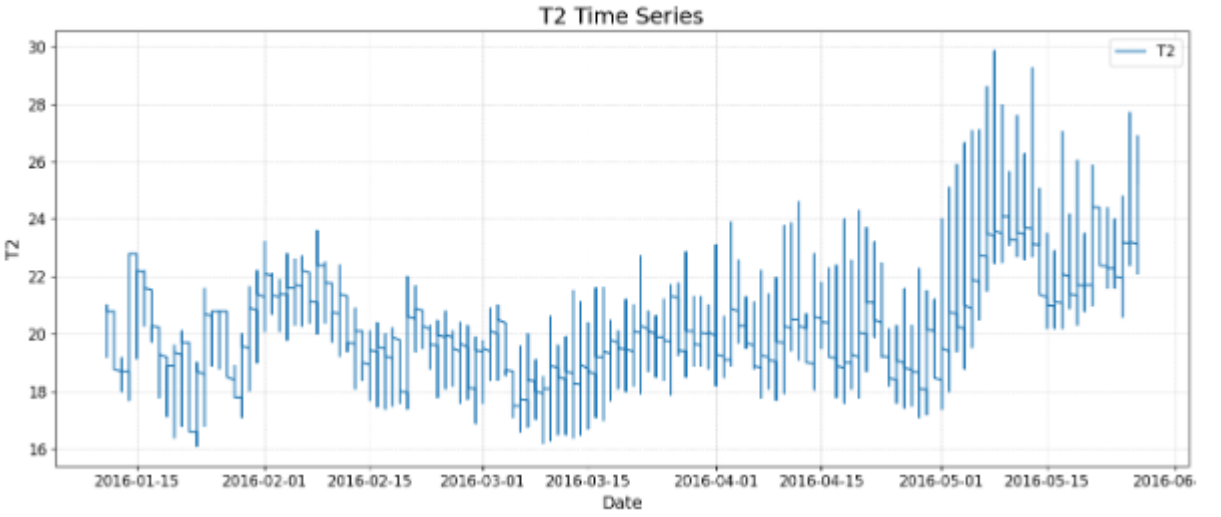
- electricity consumption on Sundays is significantly higher than on other days

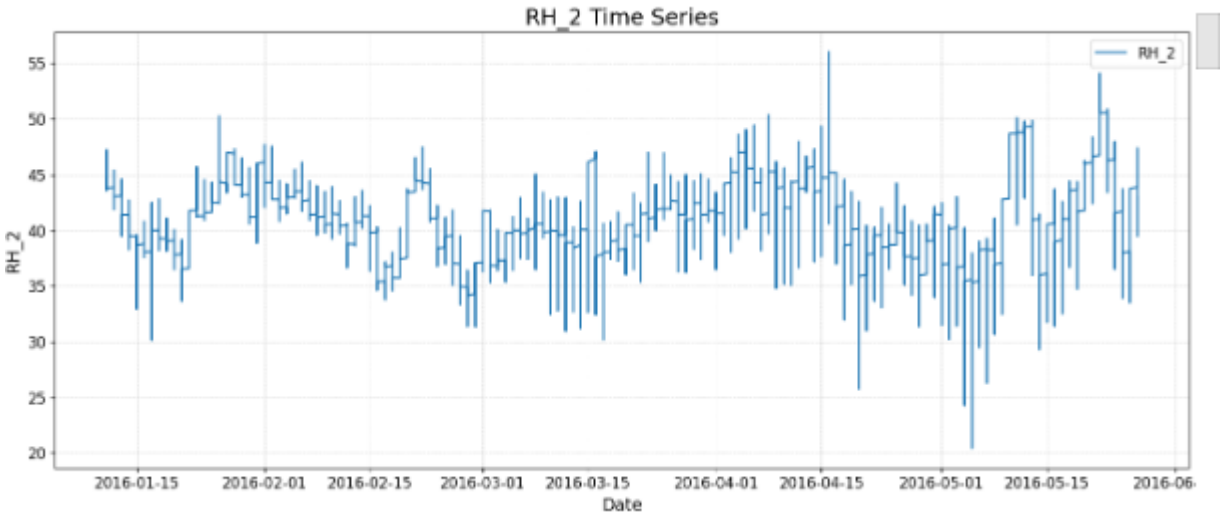
General view of the whole dataset

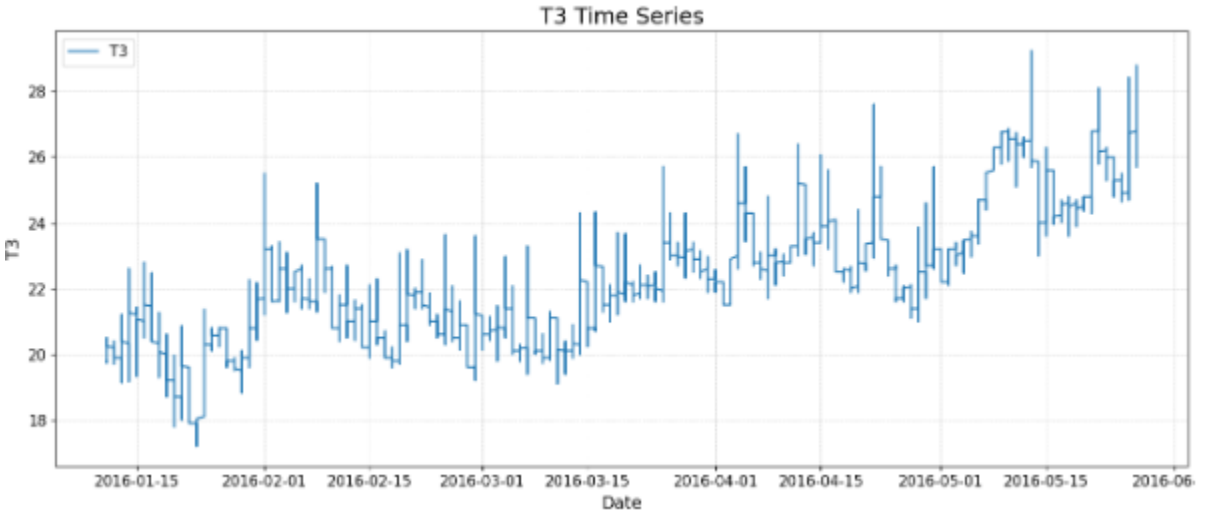


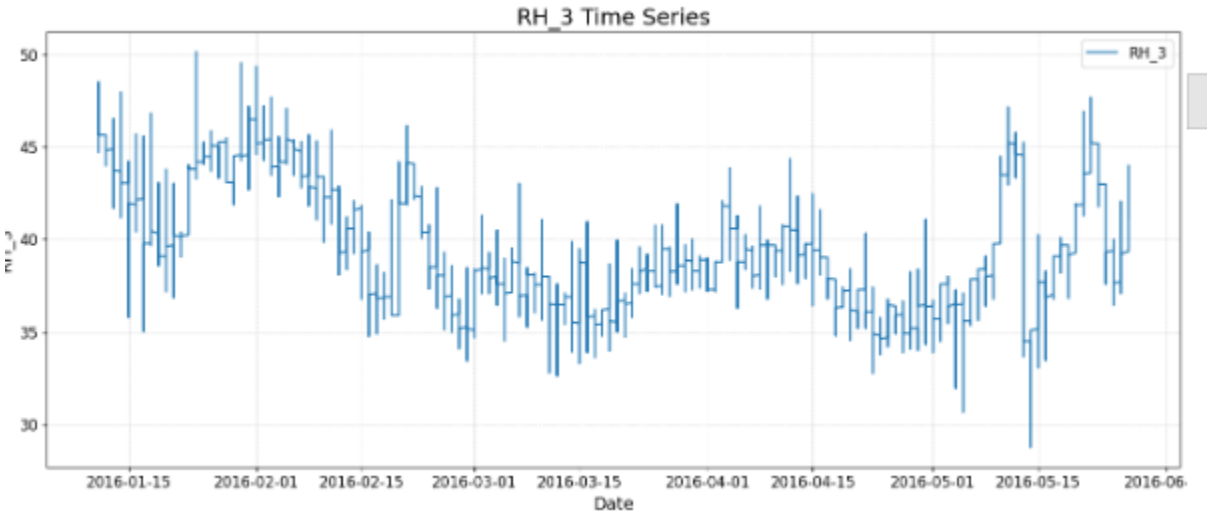

T1 Time Series
T1
26
24
22
20
18
2016-01-15
2016-02-01
2016-02-15
2016-03-01
2016-03-15
2016-04-01
2016-04-15
2016-05-01
2016-05-15
Date

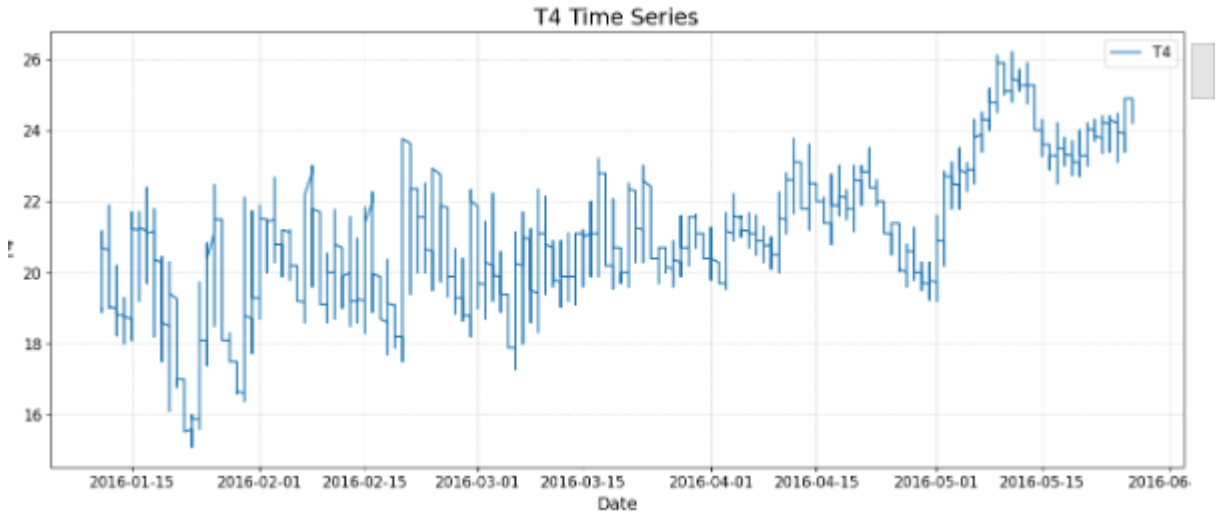

RH_1 Time Series
RH_1
65
60
55
50
45
40
35
30
2016-01-15
2016-02-01
2016-02-15
2016-03-01
2016-03-15
2016-04-01
2016-04-15
2016-05-01
2016-05-15
Date

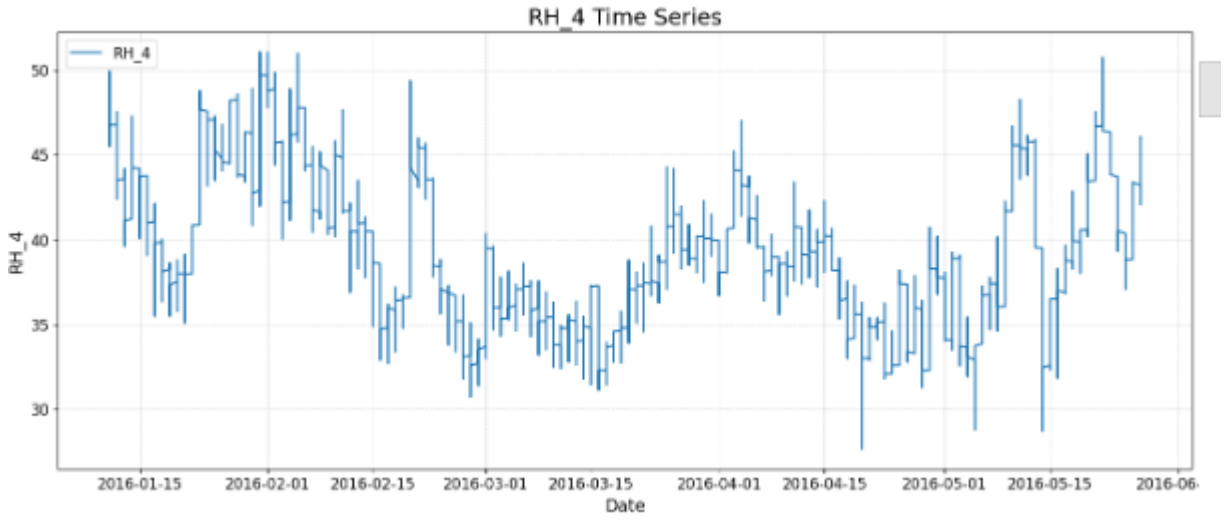

T2 Time Series
T2
30
28
26
24
22
20
18
16
2016-01-15
2016-02-01
2016-02-15
2016-03-01
2016-03-15
2016-04-01
2016-04-15
2016-05-01
2016-05-15
Date

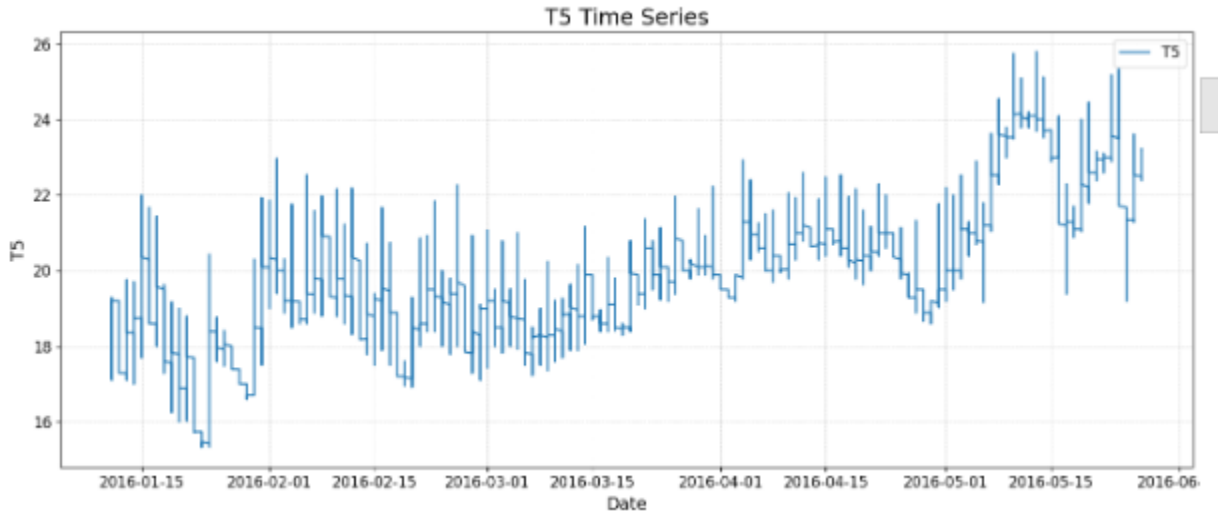

RH_2 Time Series
RH_2
RH_2
Date
2016-01-15
2016-02-01
2016-02-15
2016-03-01
2016-03-15
2016-04-01
2016-04-15
2016-05-01
2016-05-15

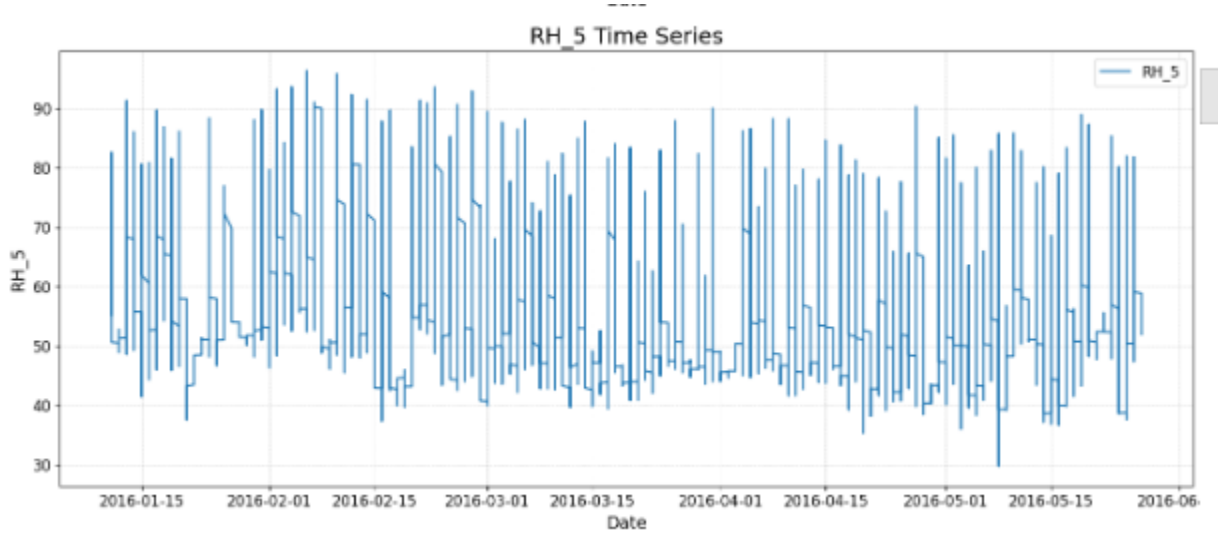

T3 Time Series
T3
T3
Date
2016-01-15
2016-02-01
2016-02-15
2016-03-01
2016-03-15
2016-04-01
2016-04-15
2016-05-01
2016-05-15

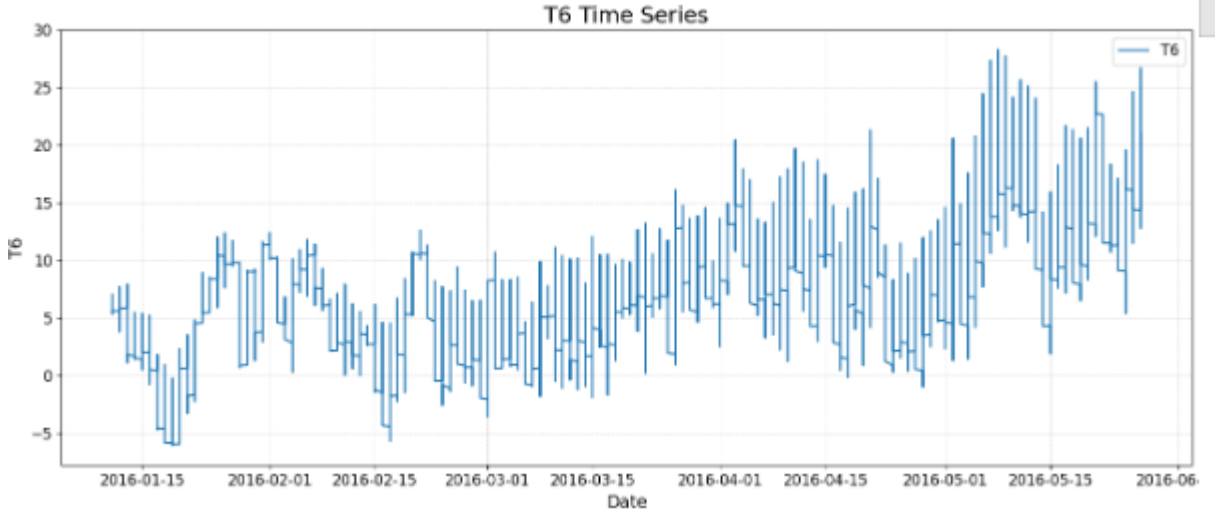

RH_3 Time Series
RH_3
Date
2016-01-15
2016-02-01
2016-02-15
2016-03-01
2016-03-15
2016-04-01
2016-04-15
2016-05-01
2016-05-15

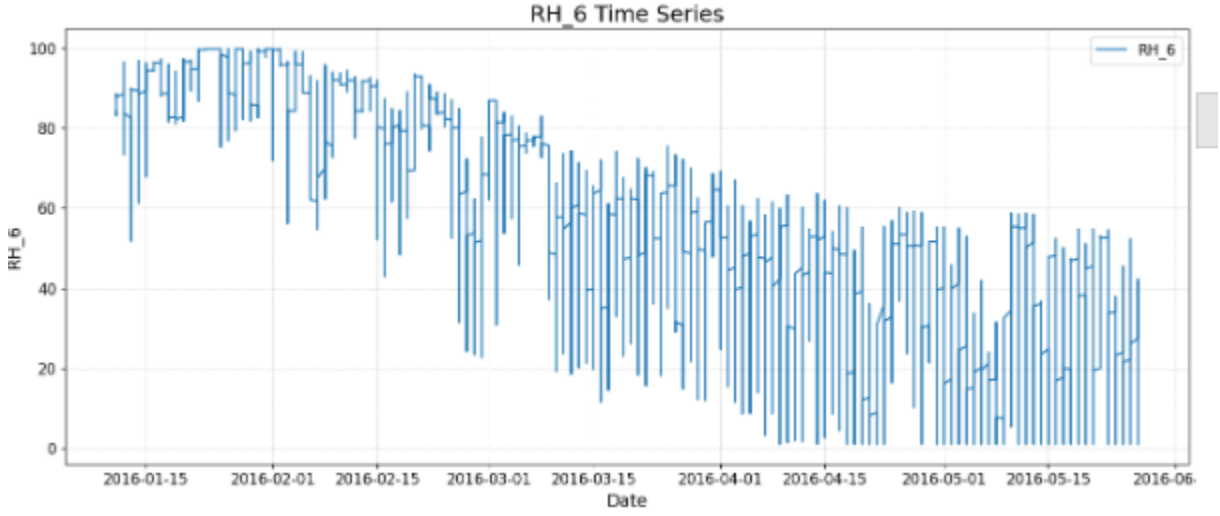

T4 Time Series
T4
26
24
22
20
18
16
2016-01-15
2016-02-01
2016-02-15
2016-03-01
2016-03-15
2016-04-01
2016-04-15
2016-05-01
2016-05-15
Date

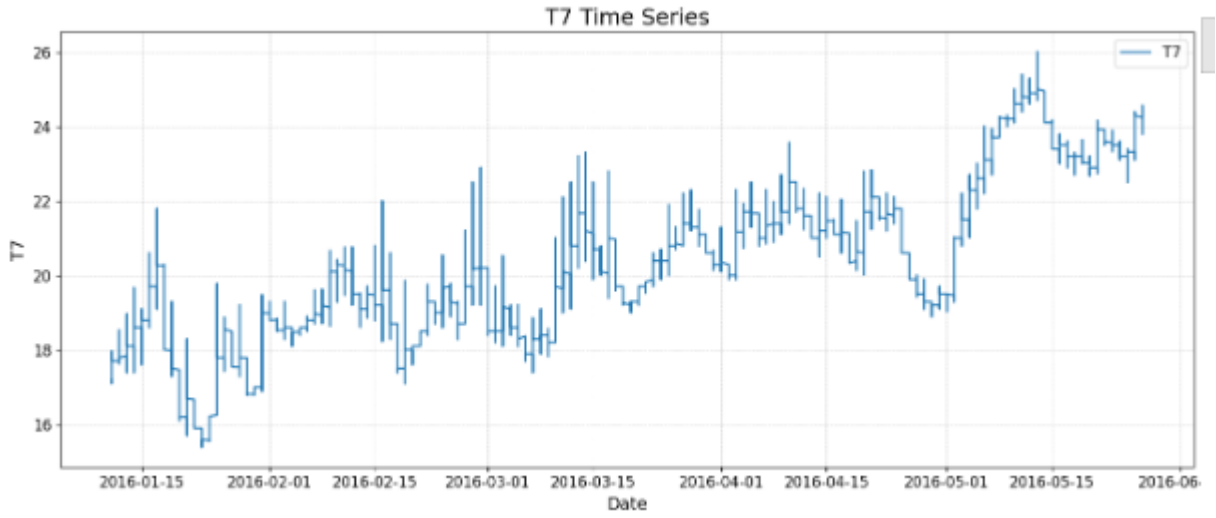

RH_4 Time Series
RH_4
50
45
40
35
30
RH_4
2016-01-15
2016-02-01
2016-02-15
2016-03-01
2016-03-15
2016-04-01
2016-04-15
2016-05-01
2016-05-15
Date

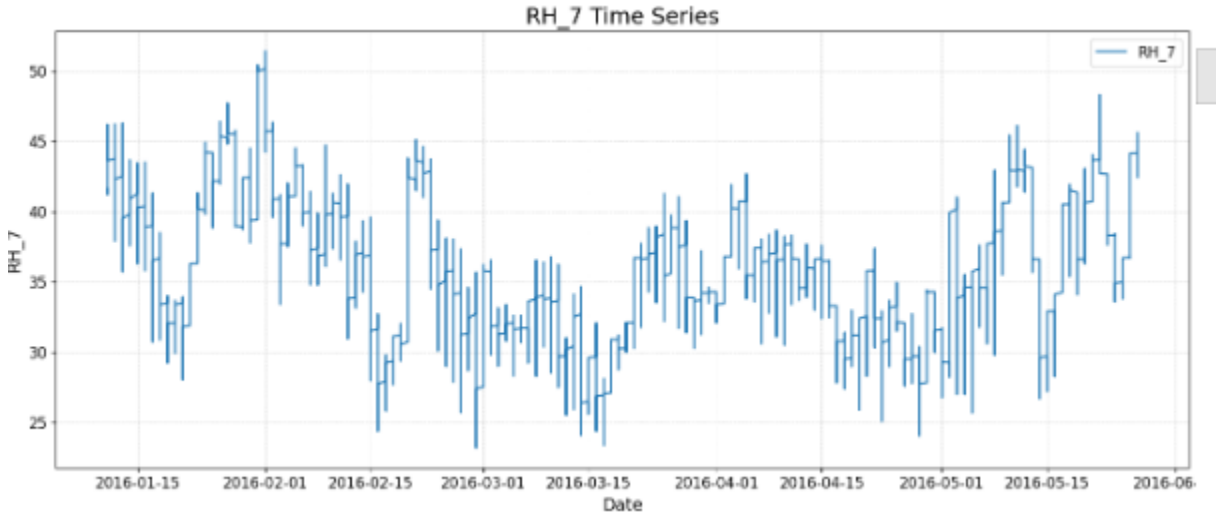

T5 Time Series
T5
26
24
22
20
18
16
T5
2016-01-15
2016-02-01
2016-02-15
2016-03-01
2016-03-15
2016-04-01
2016-04-15
2016-05-01
2016-05-15
Date

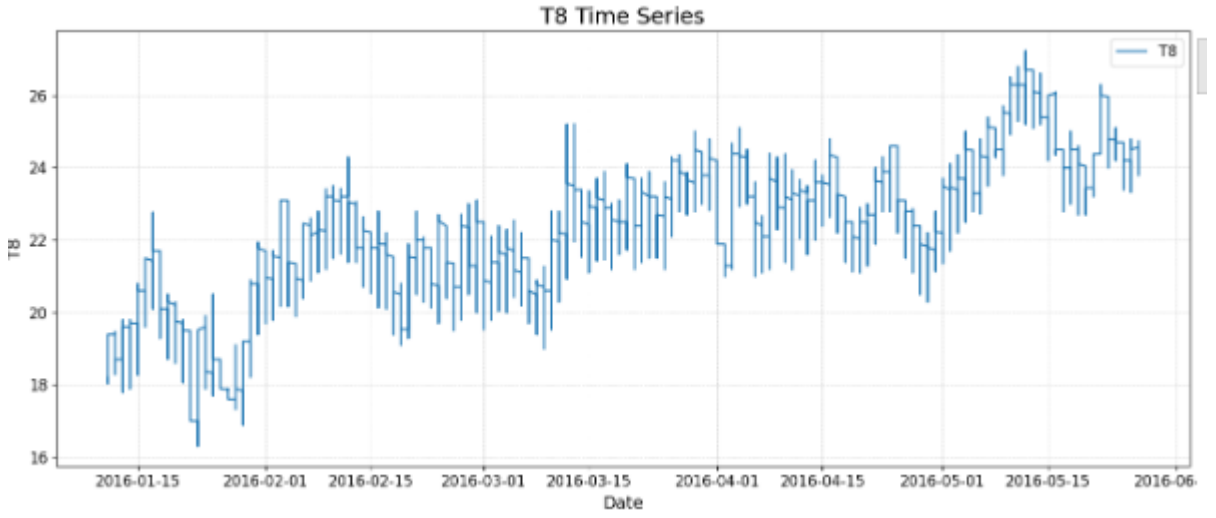

RH_5 Time Series
RH_5
RH_5
Date
2016-01-15
2016-02-01
2016-02-15
2016-03-01
2016-03-15
2016-04-01
2016-04-15
2016-05-01
2016-05-15

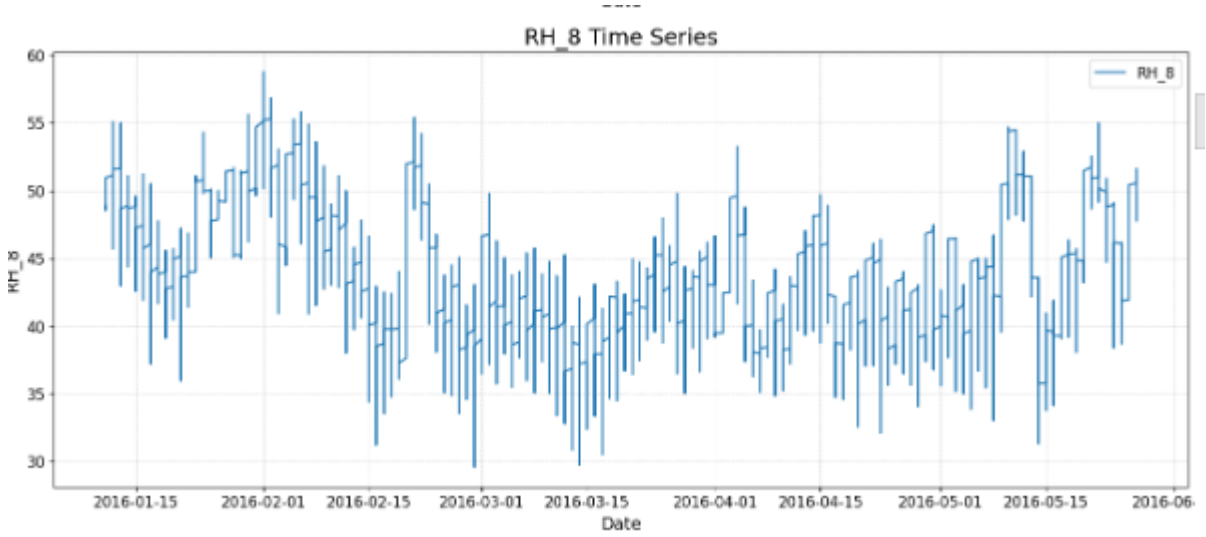

T6 Time Series
T6
T6
Date
2016-01-15
2016-02-01
2016-02-15
2016-03-01
2016-03-15
2016-04-01
2016-04-15
2016-05-01
2016-05-15

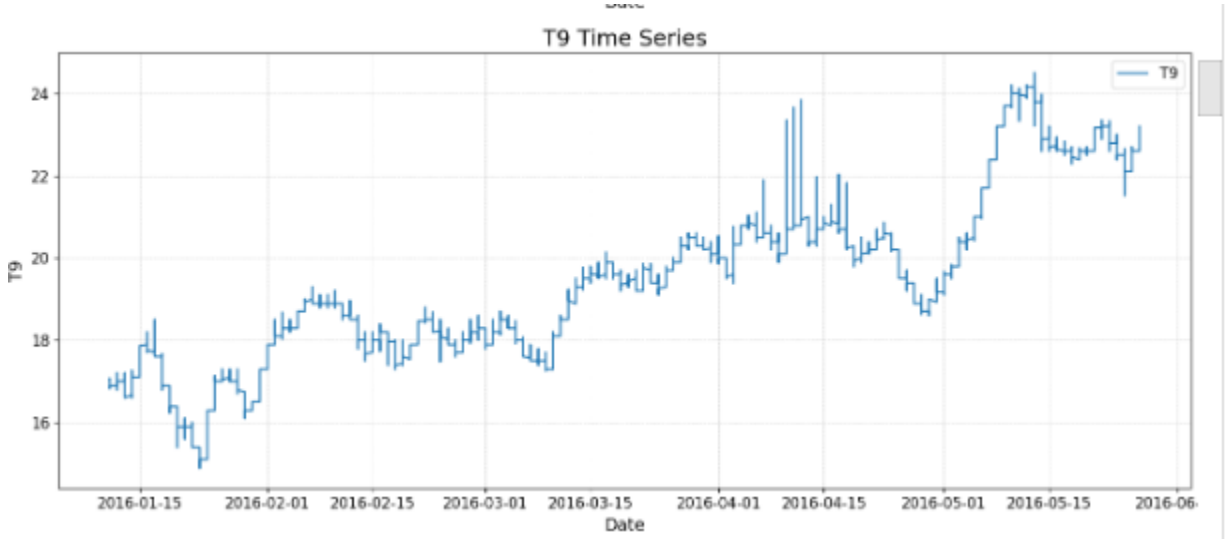

RH_6 Time Series
RH_6
RH_6
Date
2016-01-15
2016-02-01
2016-02-15
2016-03-01
2016-03-15
2016-04-01
2016-04-15
2016-05-01
2016-05-15

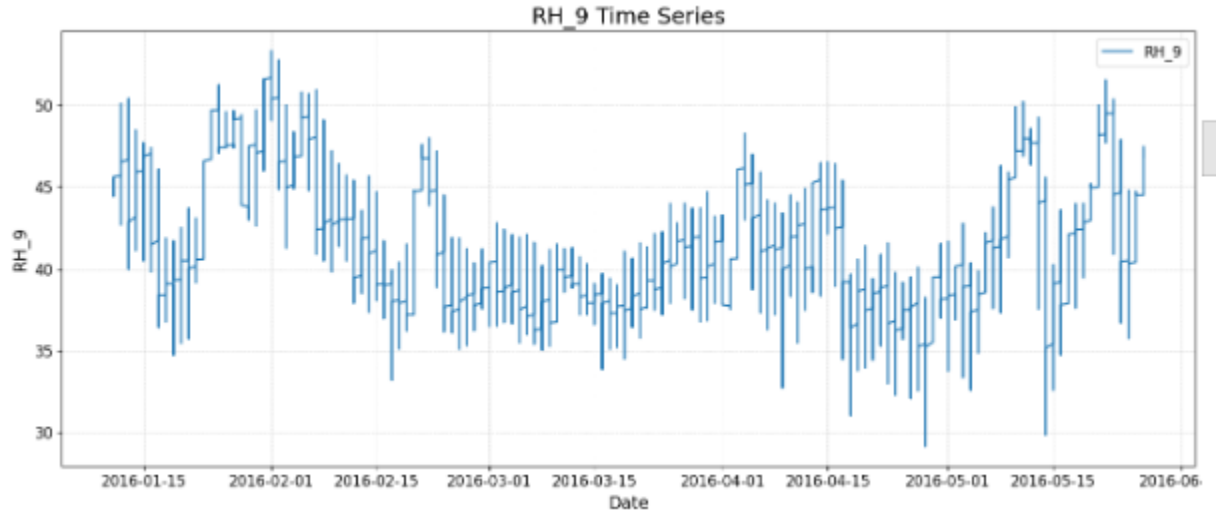

T7 Time Series
T7
26
24
22
20
18
16
2016-01-15
2016-02-01
2016-02-15
2016-03-01
2016-03-15
2016-04-01
2016-04-15
2016-05-01
2016-05-15
Date

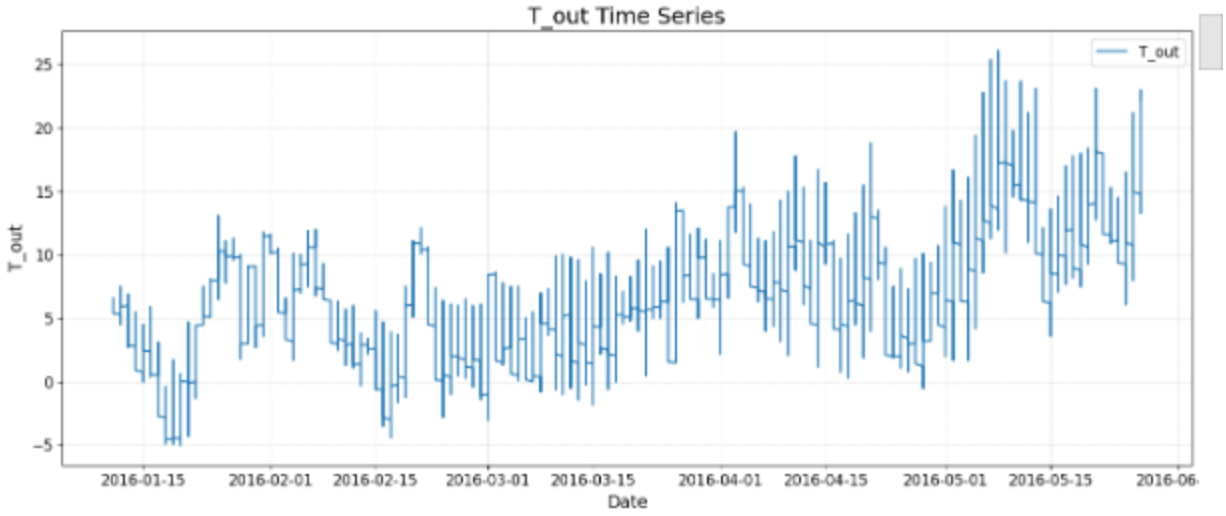

RH_7 Time Series
RH_7
50
45
40
35
30
25
2016-01-15
2016-02-01
2016-02-15
2016-03-01
2016-03-15
2016-04-01
2016-04-15
2016-05-01
2016-05-15
Date

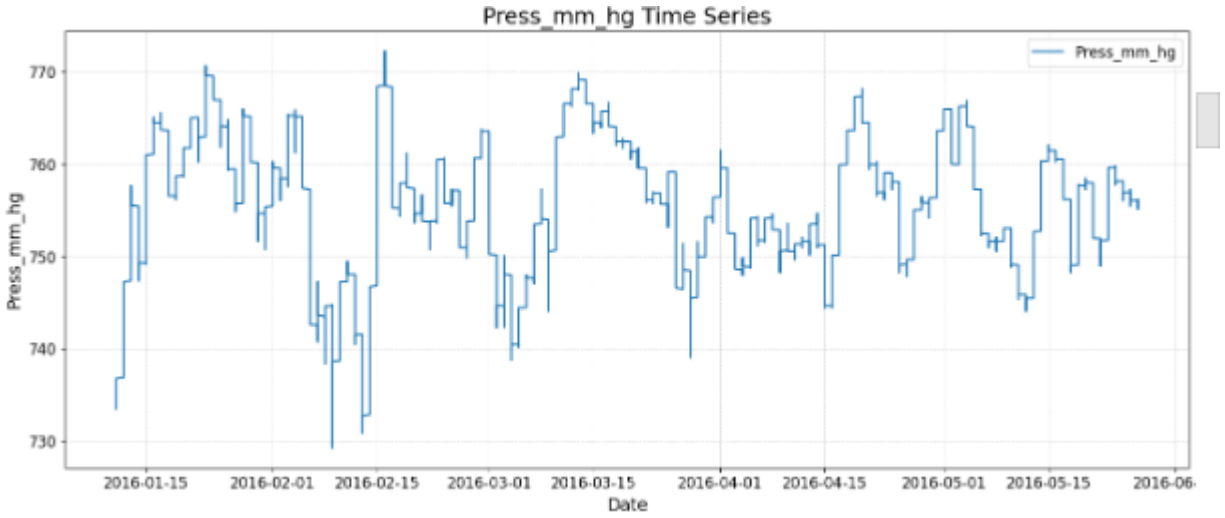

T8 Time Series
T8
26
24
22
20
18
16
2016-01-15
2016-02-01
2016-02-15
2016-03-01
2016-03-15
2016-04-01
2016-04-15
2016-05-01
2016-05-15
Date

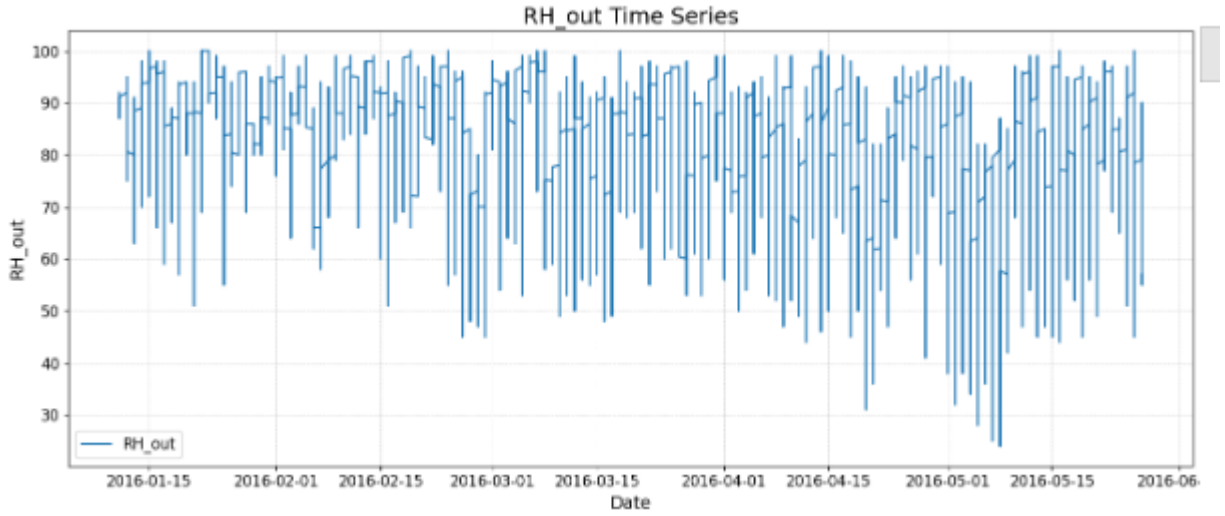

RH_8 Time Series
RH_8
RH_8
60
55
50
45
40
35
30
2016-01-15
2016-02-01
2016-02-15
2016-03-01
2016-03-15
2016-04-01
2016-04-15
2016-05-01
2016-05-15
2016-06
Date

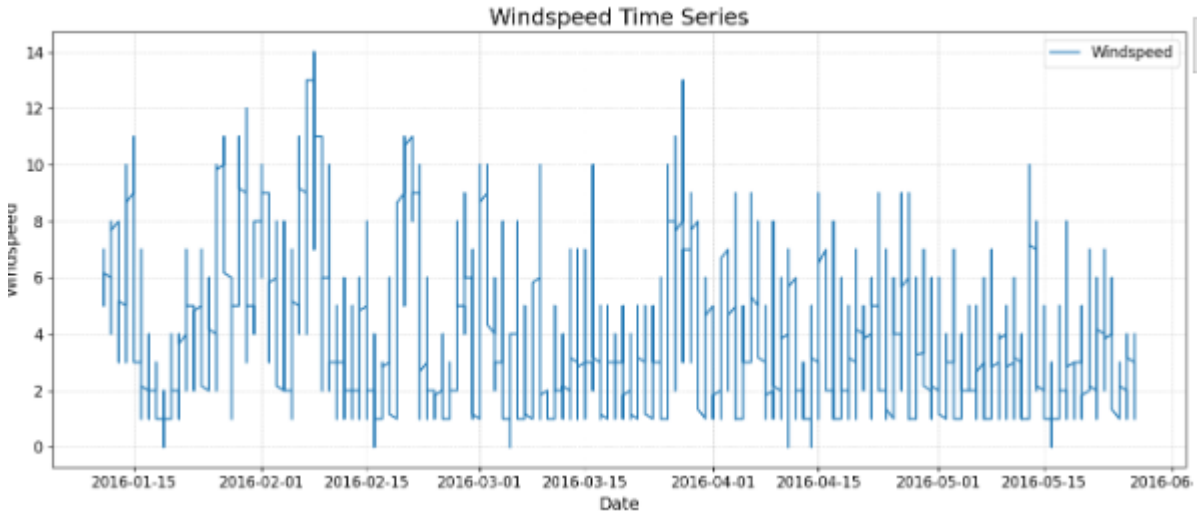

T9 Time Series
T9
T9
24
22
20
18
16
2016-01-15
2016-02-01
2016-02-15
2016-03-01
2016-03-15
2016-04-01
2016-04-15
2016-05-01
2016-05-15
2016-06
Date

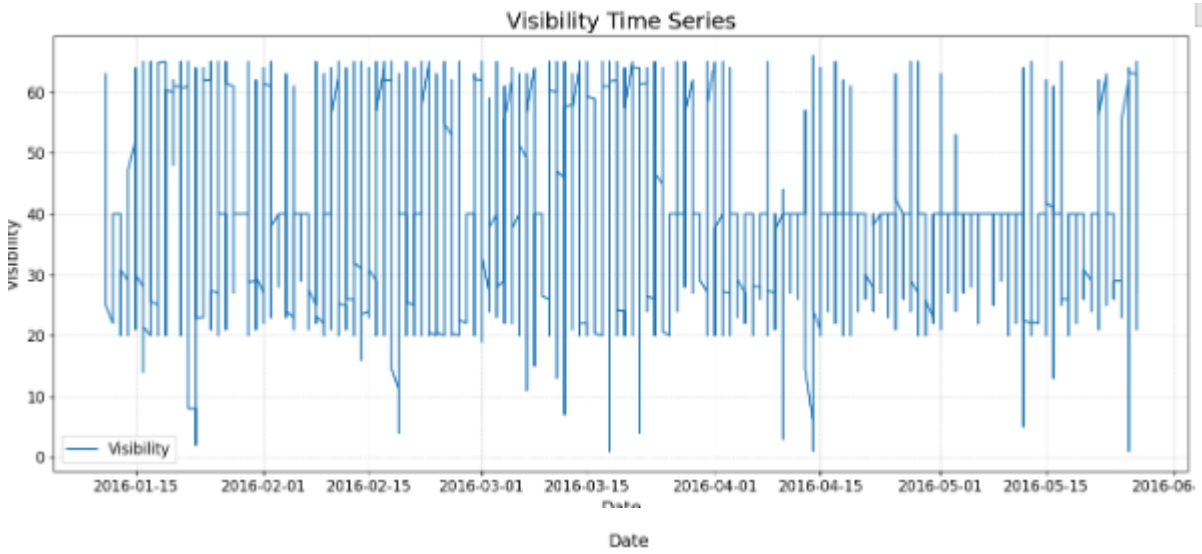

RH_9 Time Series
RH_9
RH_9
50
45
40
35
30
2016-01-15
2016-02-01
2016-02-15
2016-03-01
2016-03-15
2016-04-01
2016-04-15
2016-05-01
2016-05-15
2016-06
Date

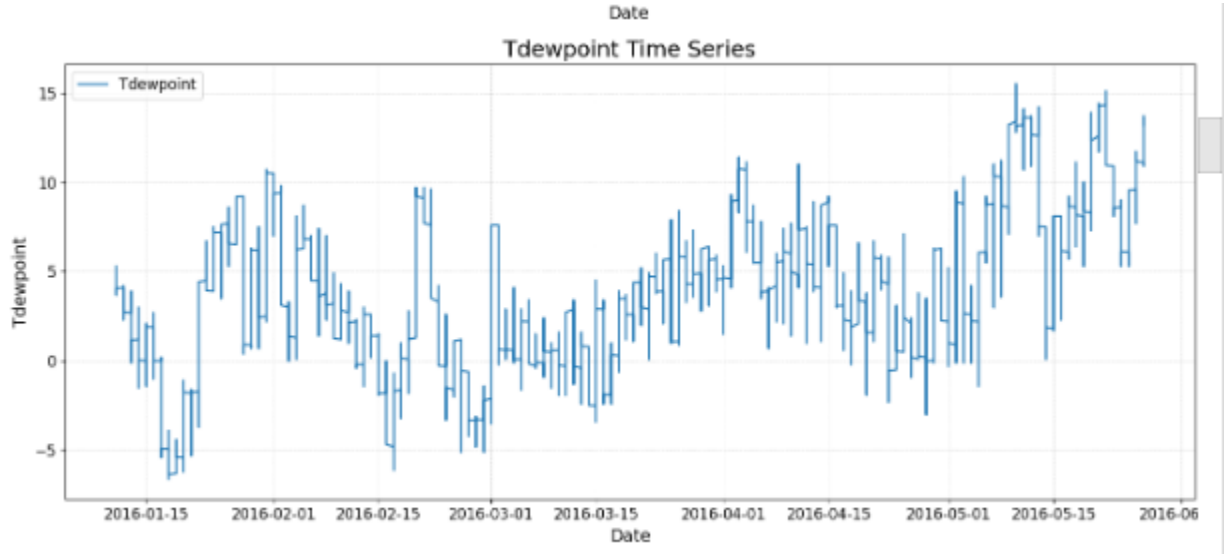

T_out Time Series
T_out
25
20
15
10
5
0
−5
2016-01-15
2016-02-01
2016-02-15
2016-03-01
2016-03-15
2016-04-01
2016-04-15
2016-05-01
2016-05-15
Date

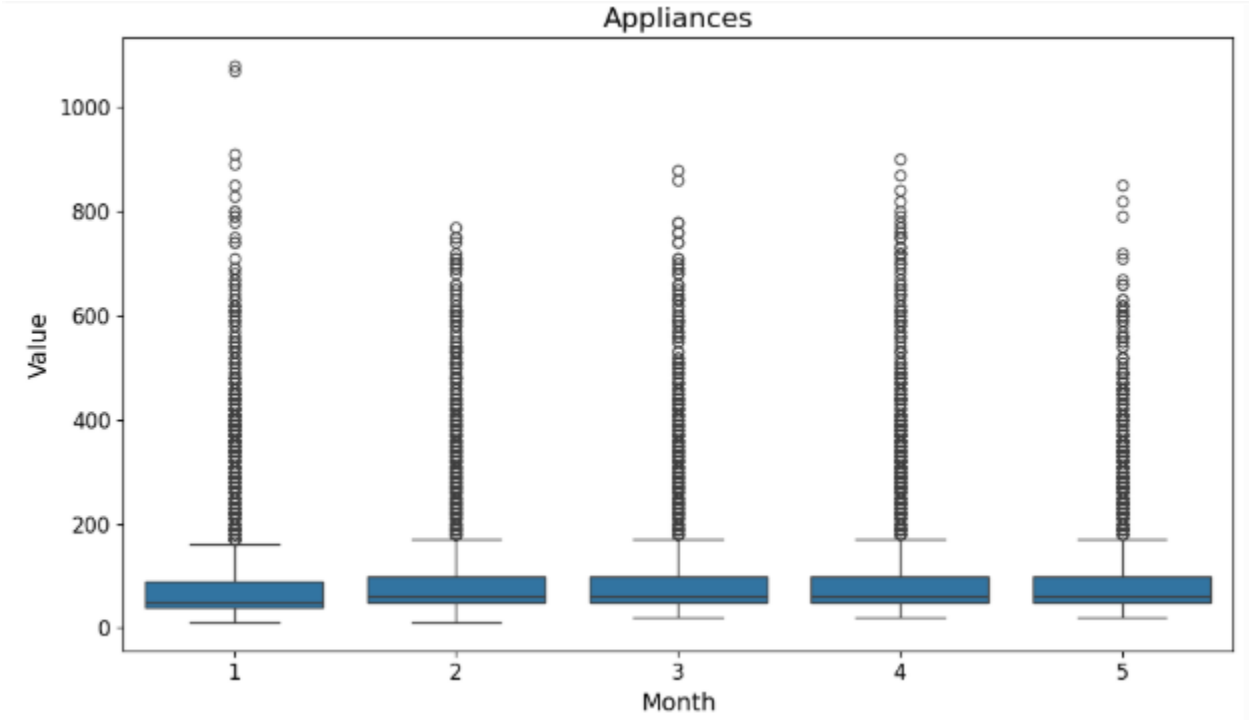

Press_mm_hg Time Series
Press_mm_hg
770
760
750
740
730
2016-01-15
2016-02-01
2016-02-15
2016-03-01
2016-03-15
2016-04-01
2016-04-15
2016-05-01
2016-05-15
Date

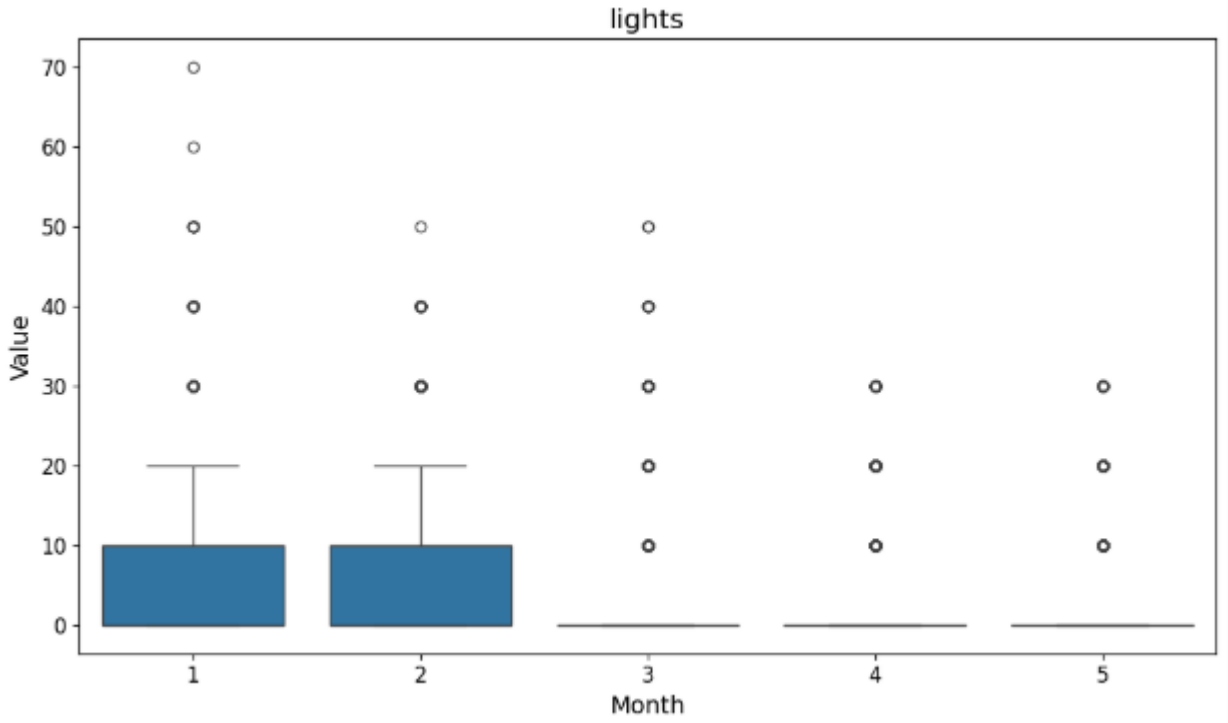

RH_out Time Series
RH_out
100
90
80
70
60
50
40
30
2016-01-15
2016-02-01
2016-02-15
2016-03-01
2016-03-15
2016-04-01
2016-04-15
2016-05-01
2016-05-15
Date

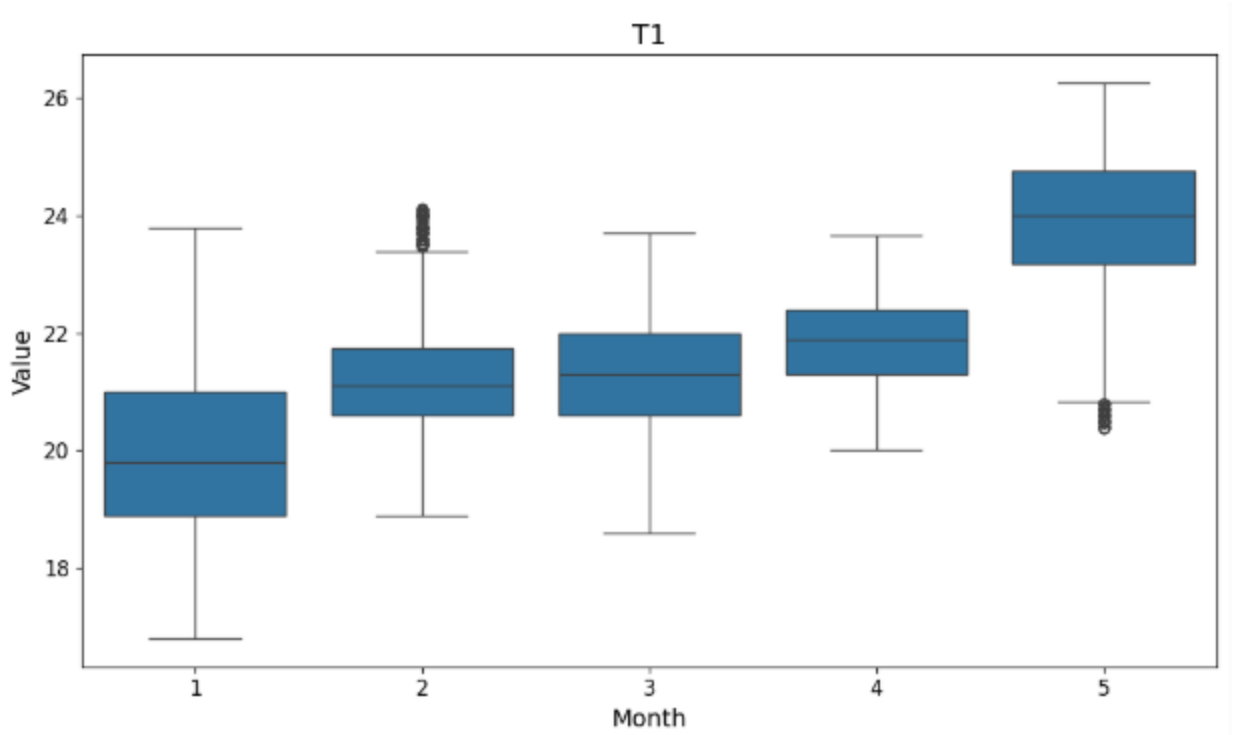

Windspeed Time Series
Windspeed
Windspeed
0
2
4
6
8
10
12
14
2016-01-15
2016-02-01
2016-02-15
2016-03-01
2016-03-15
2016-04-01
2016-04-15
2016-05-01
2016-05-15
Date

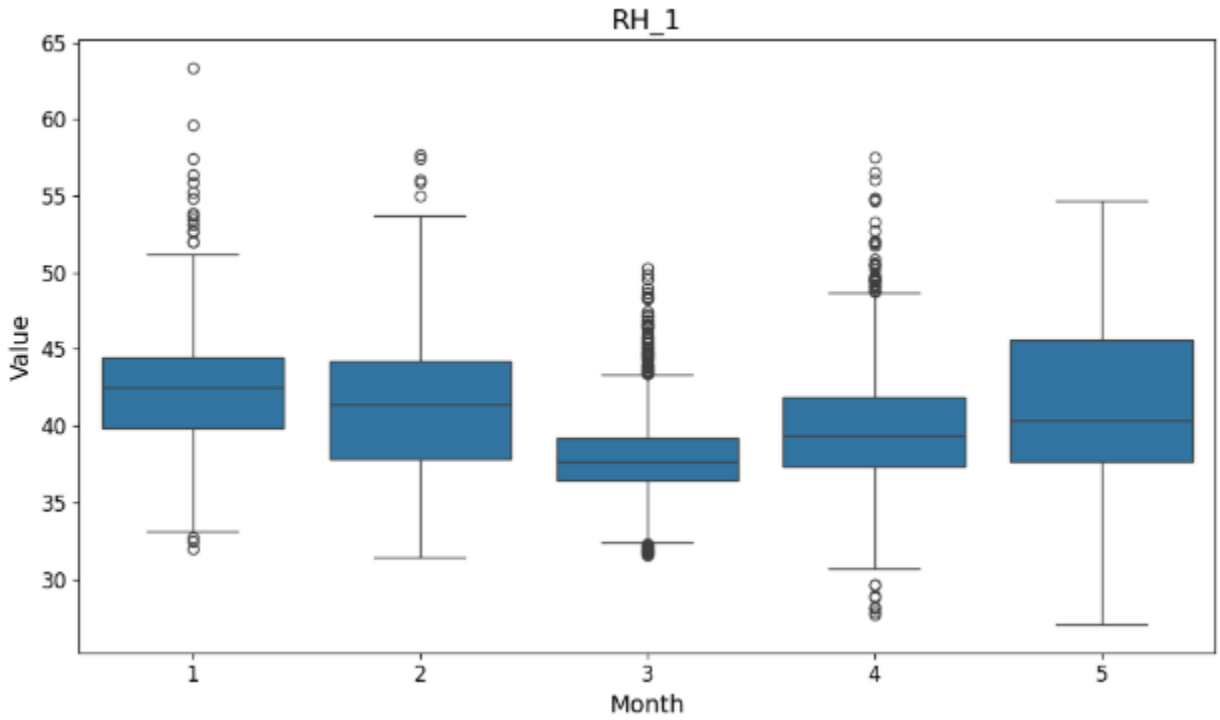

Visibility Time Series
Visibility
Visibility
0
10
20
30
40
50
60
2016-01-15
2016-02-01
2016-02-15
2016-03-01
2016-03-15
2016-04-01
2016-04-15
2016-05-01
2016-05-15
Date

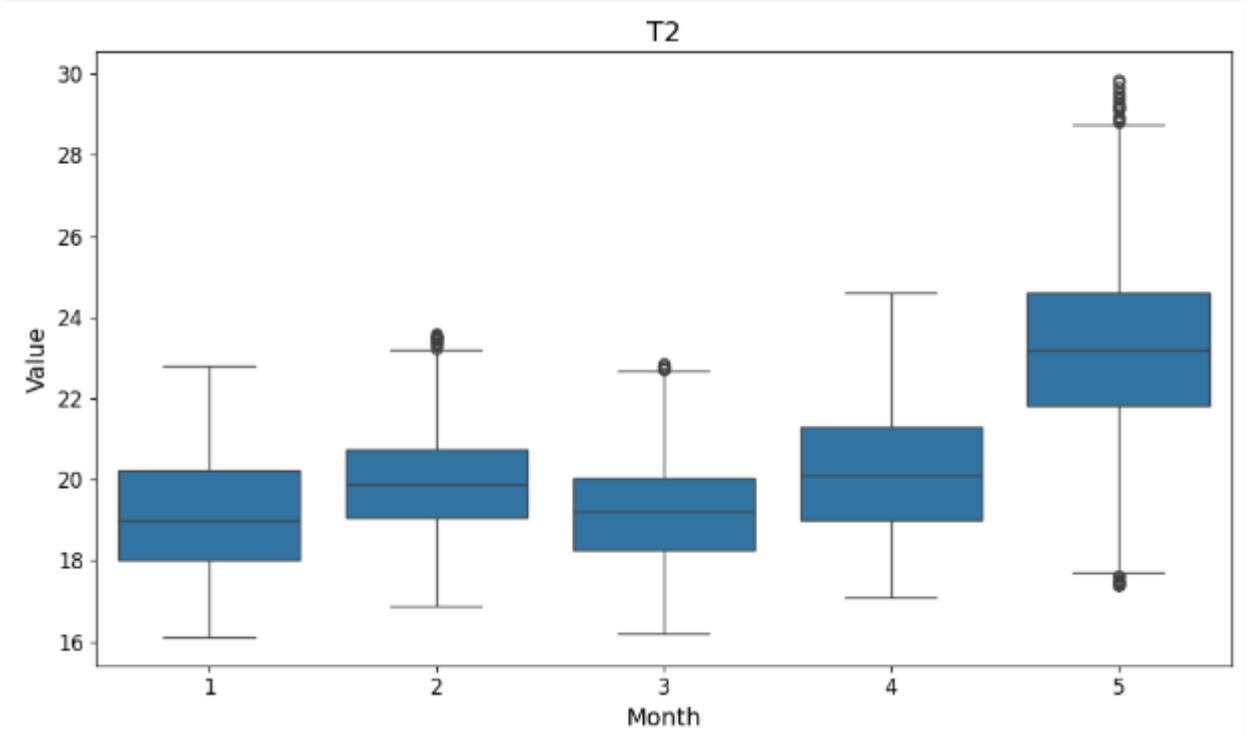

Tdewpoint Time Series
Tdewpoint
Tdewpoint
−5
0
5
10
15
2016-01-15
2016-02-01
2016-02-15
2016-03-01
2016-03-15
2016-04-01
2016-04-15
2016-05-01
2016-05-15
Date

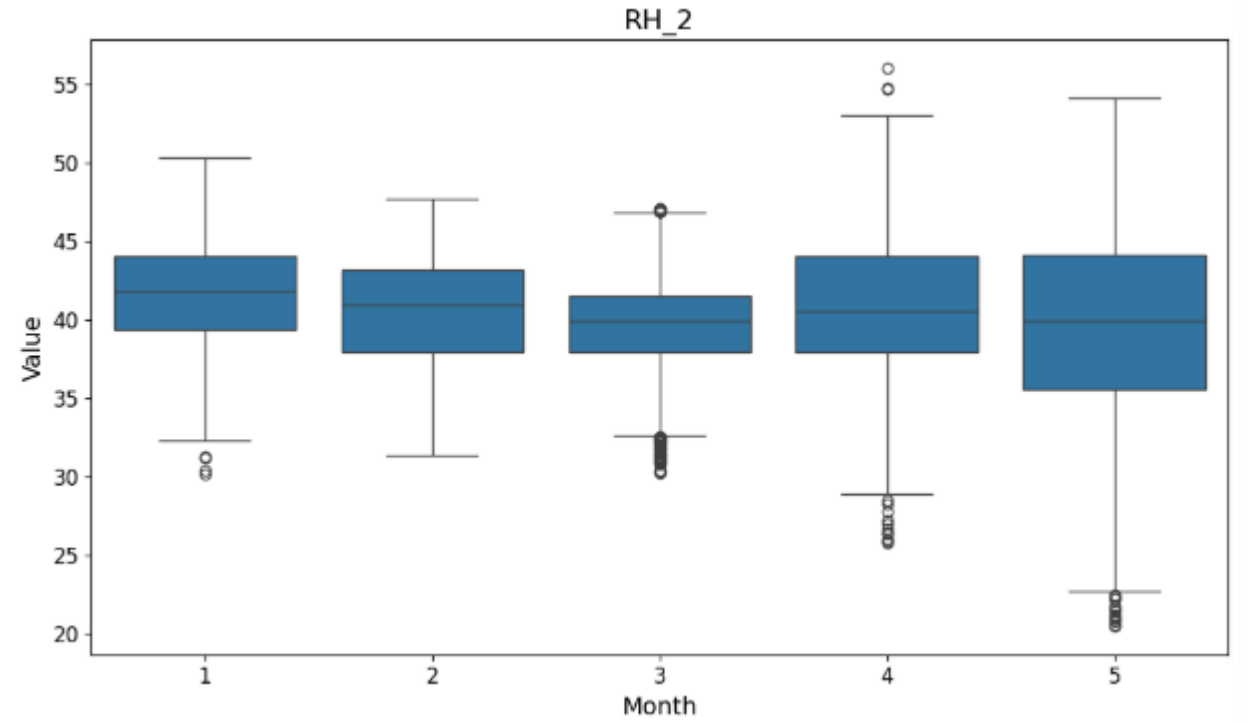

Appliances
Value
1000
800
600
400
200
0
1
2
3
4
5
Month

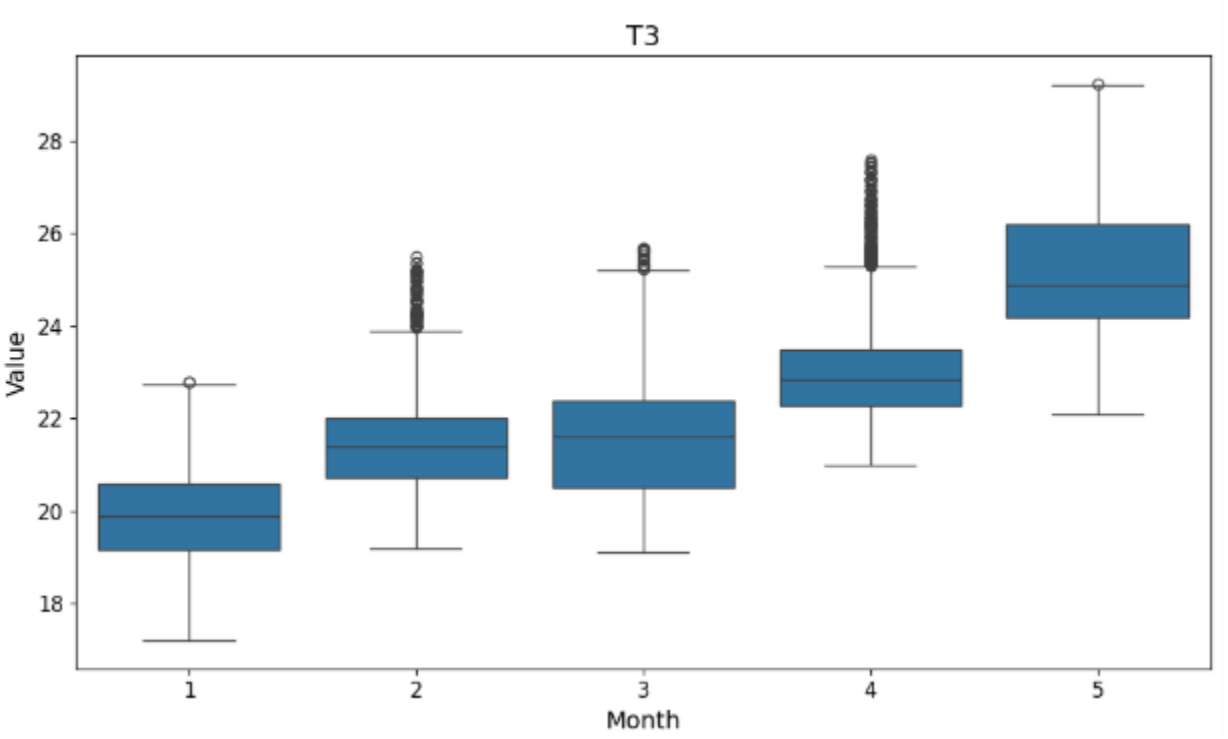

lights
Value
70
60
50
40
30
20
10
0
1
2
3
4
5
Month

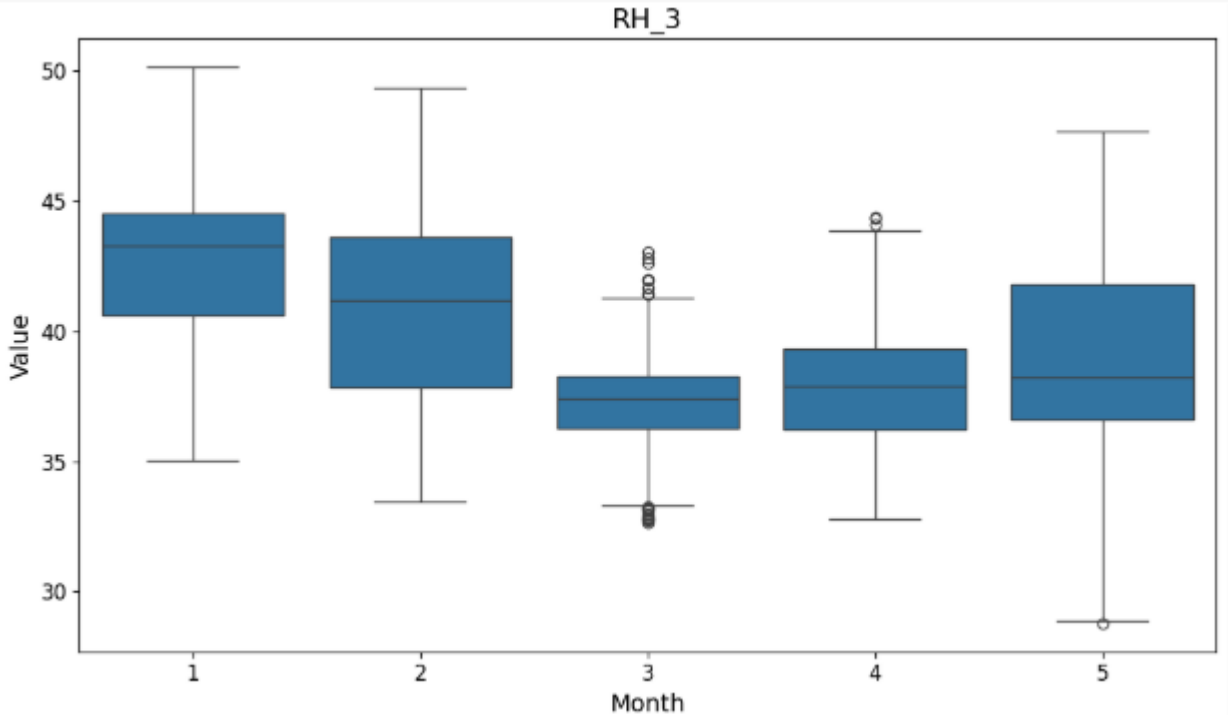

T1
Value
26
24
22
20
18
1
2
3
4
5
Month

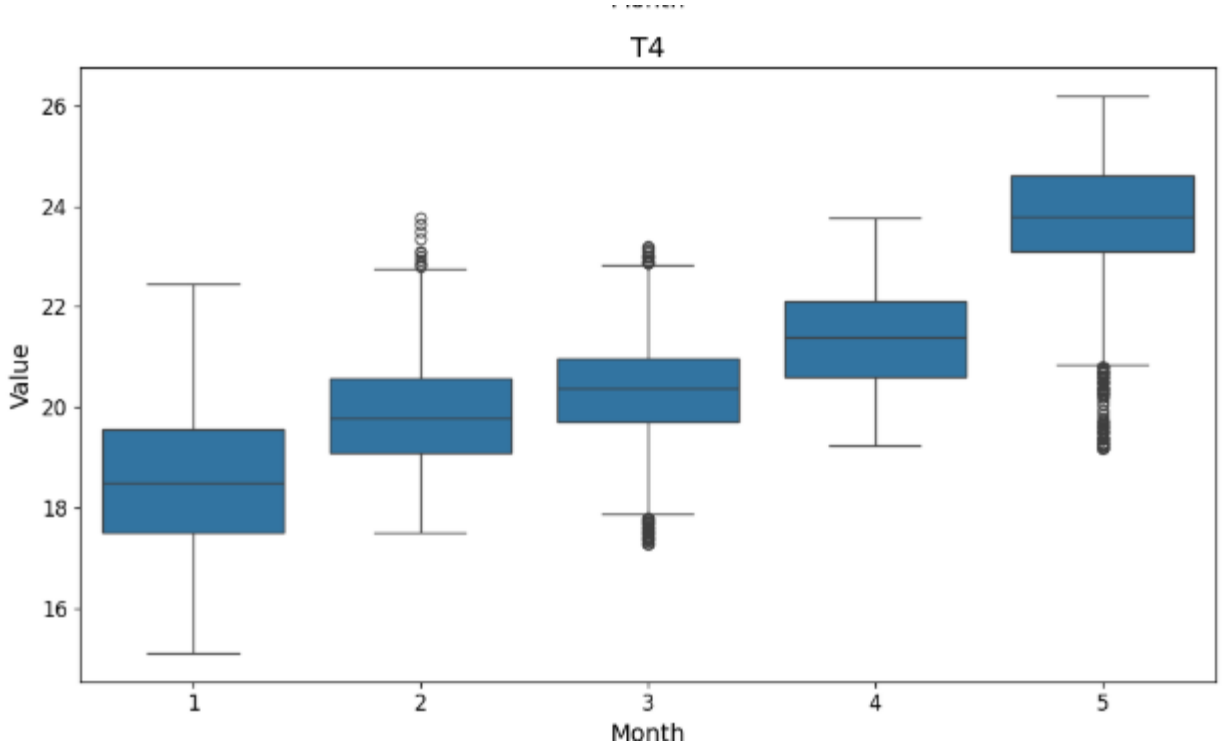

RH_1
Value
65
60
55
50
45
40
35
30
1
2
3
4
5
Month

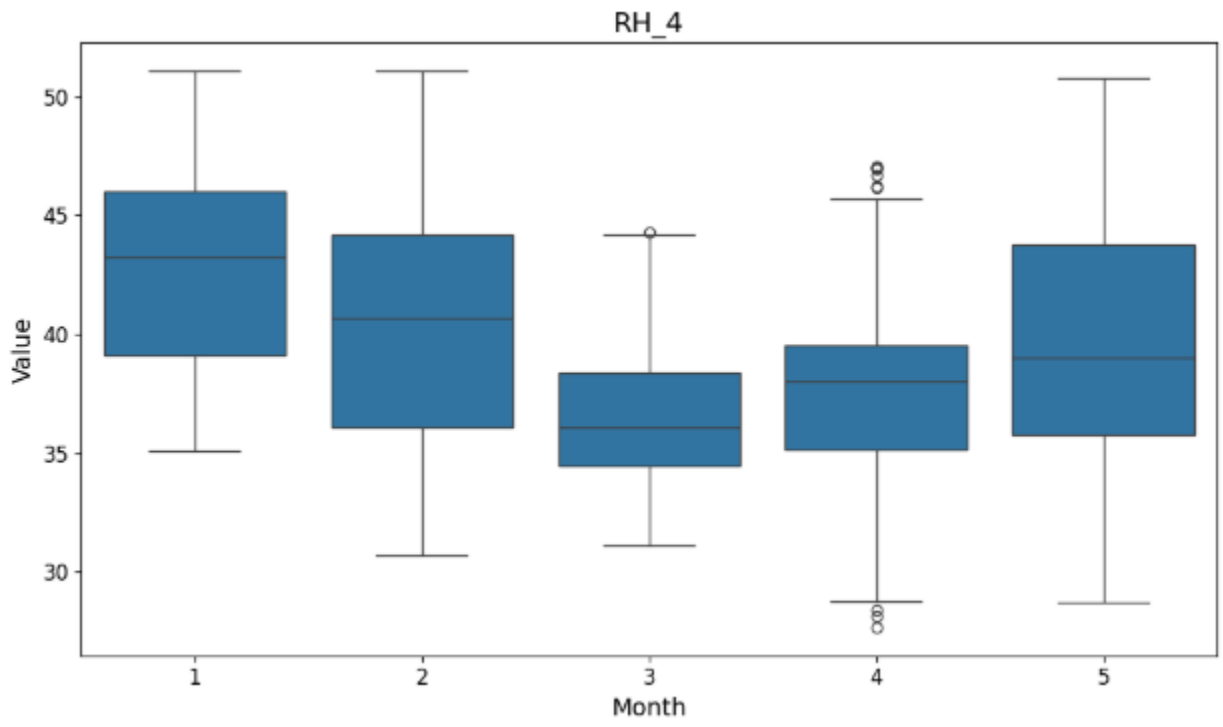

T2
Value
Month
30
28
26
24
22
20
18
16
1
2
3
4
5

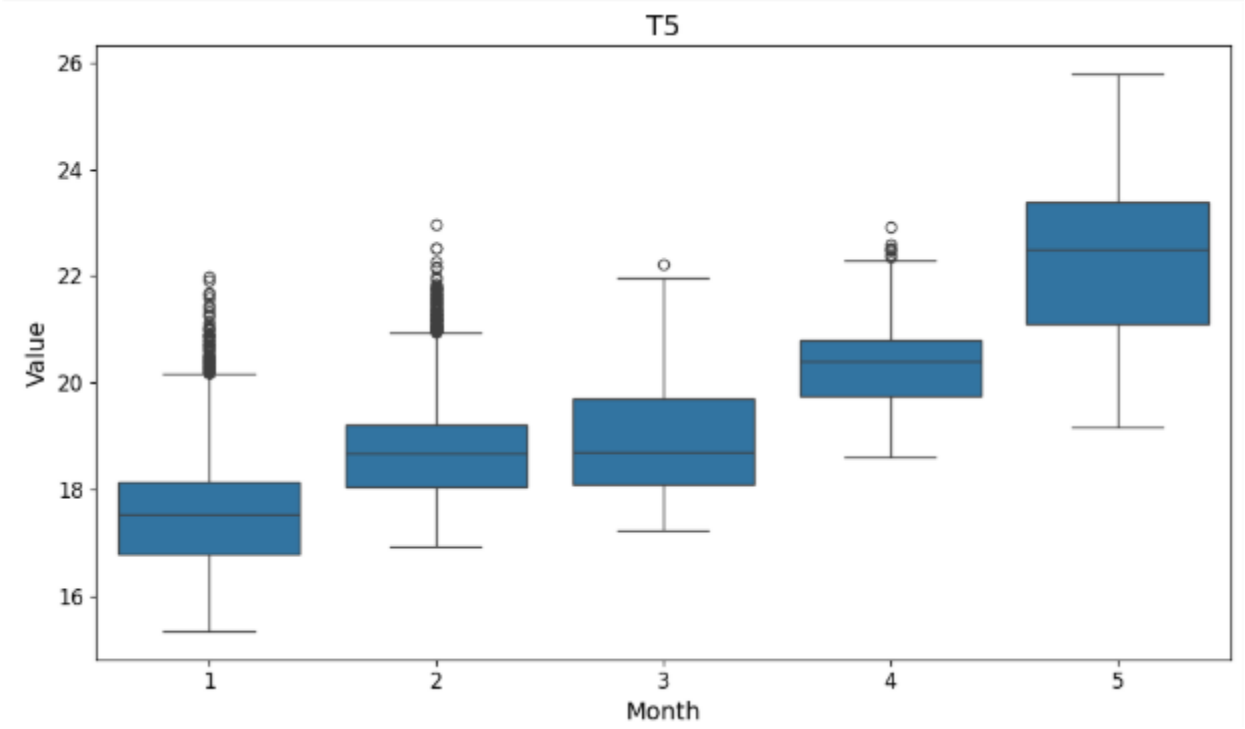

RH_2
Value
Month
55
50
45
40
35
30
25
20
1
2
3
4
5

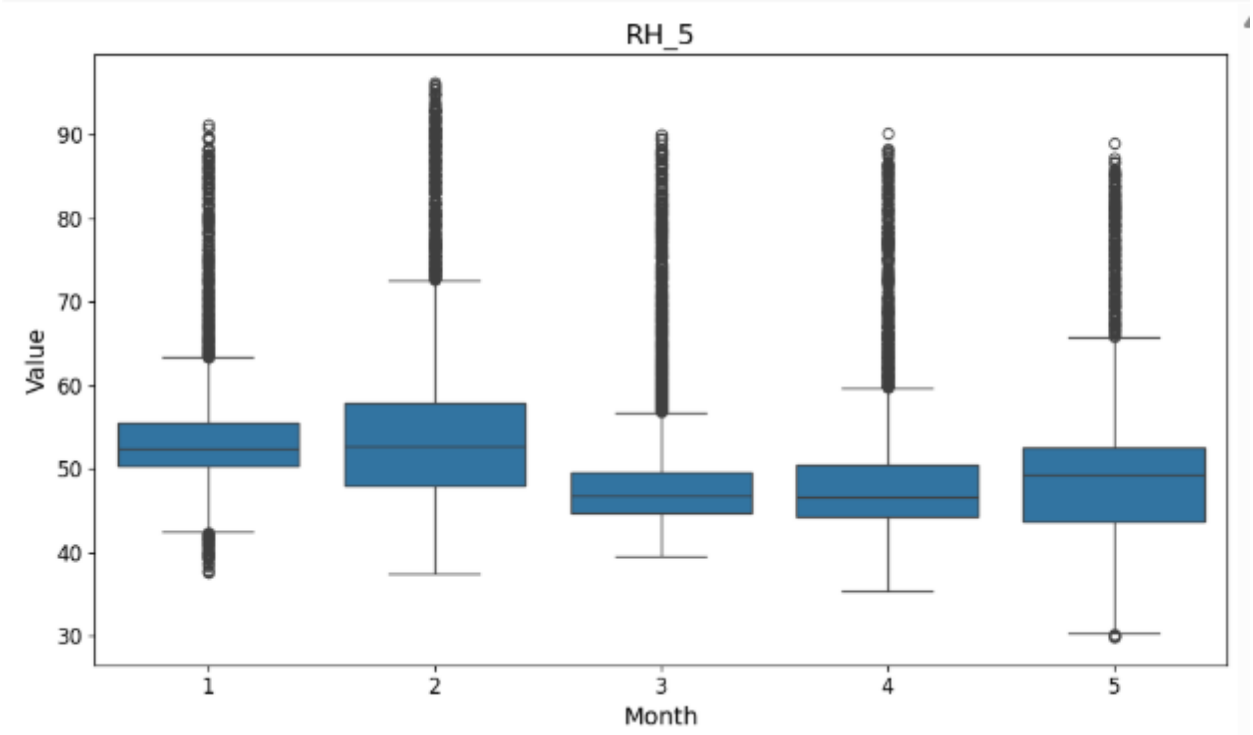

T3
Value
28
26
24
22
20
18
1
2
3
4
5
Month

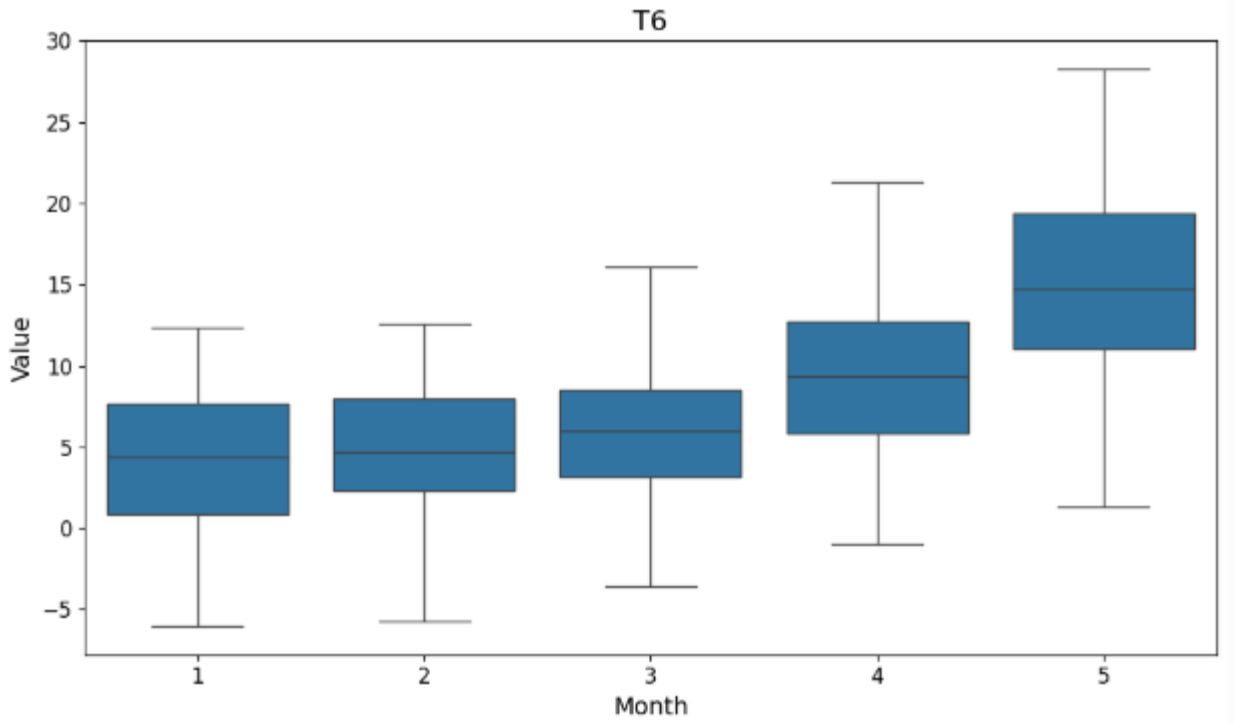

RH_3
Value
50
45
40
35
30
1
2
3
4
5
Month

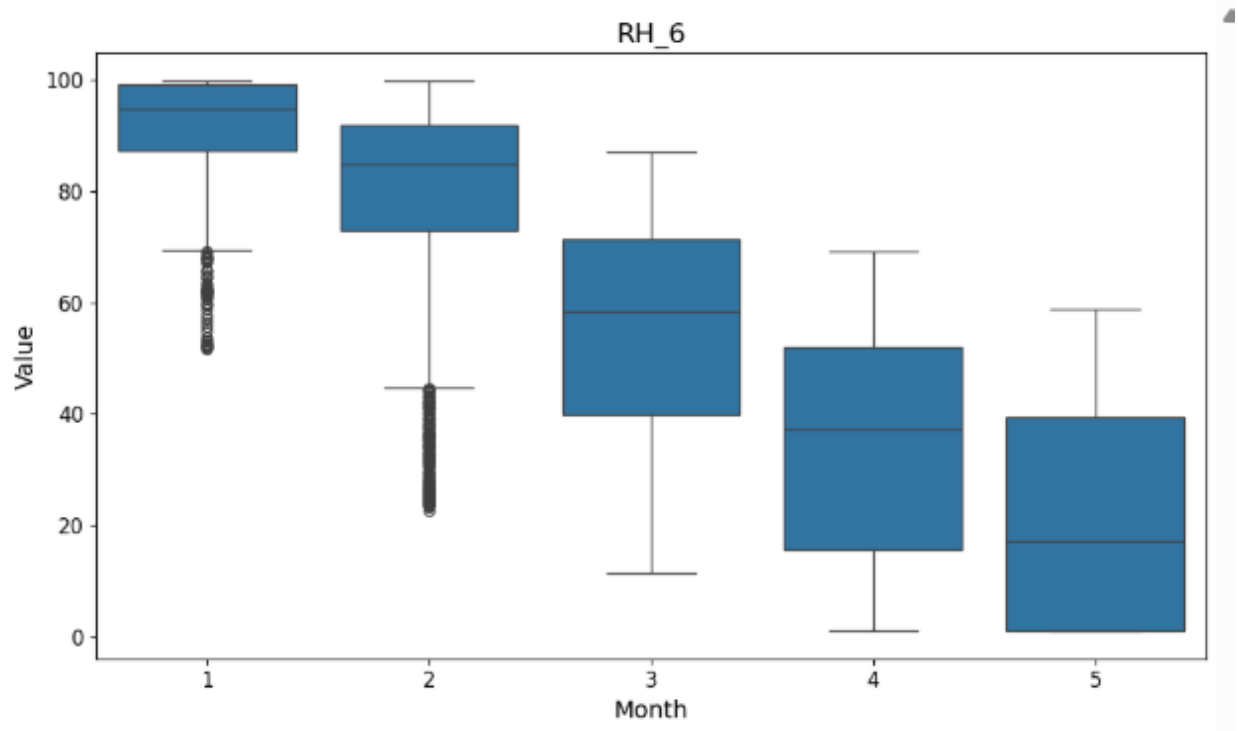

T4
Value
Month
26
24
22
20
18
16
1
2
3
4
5

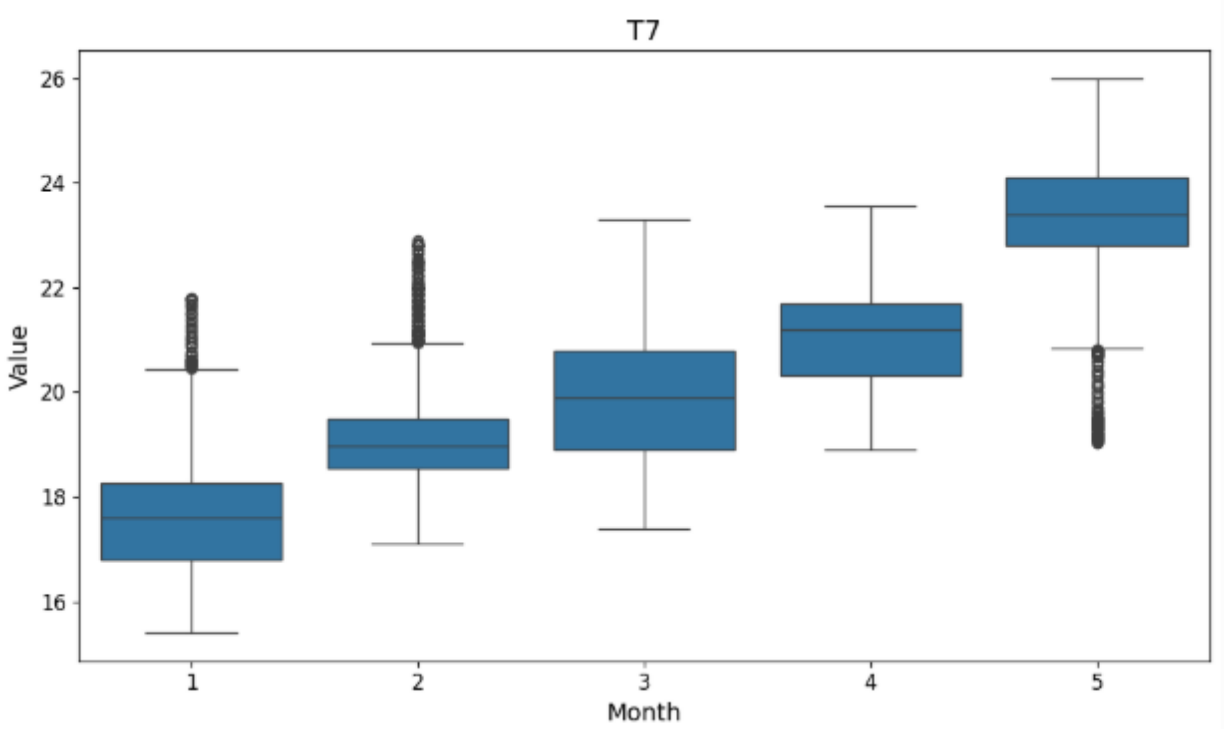

RH_4
Value
Month
50
45
40
35
30
1
2
3
4
5

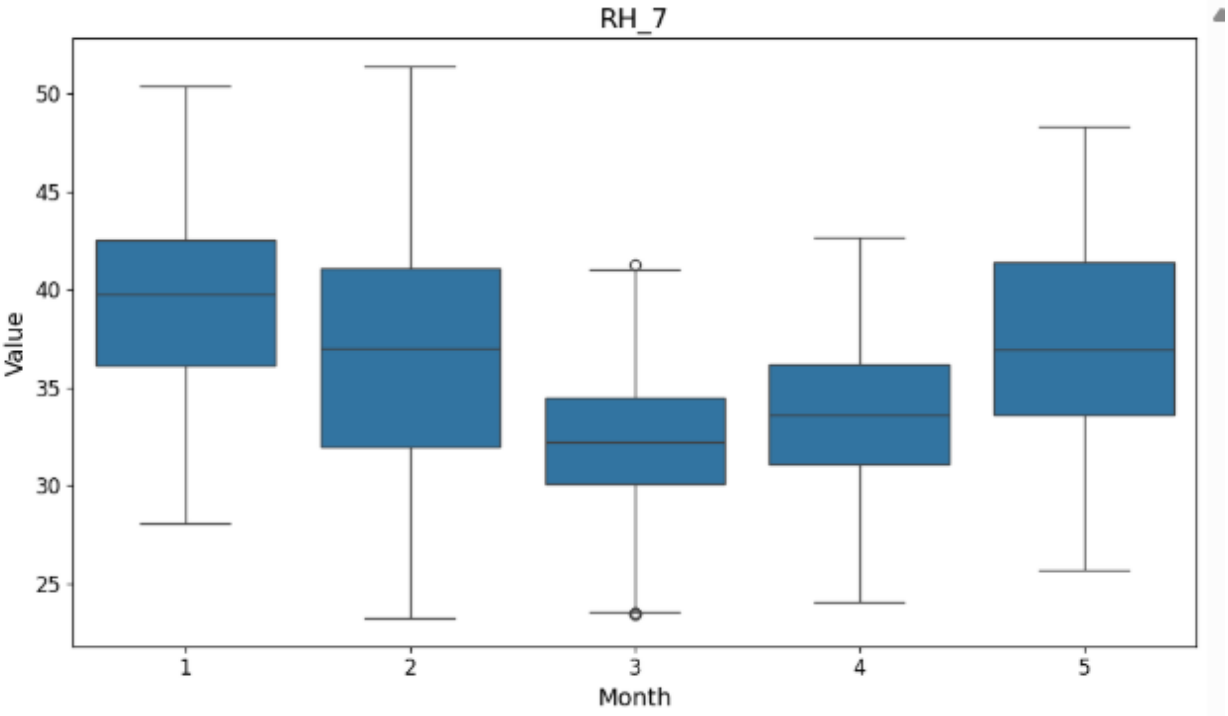

T5
Value
26
24
22
20
18
16
1
2
3
4
5
Month

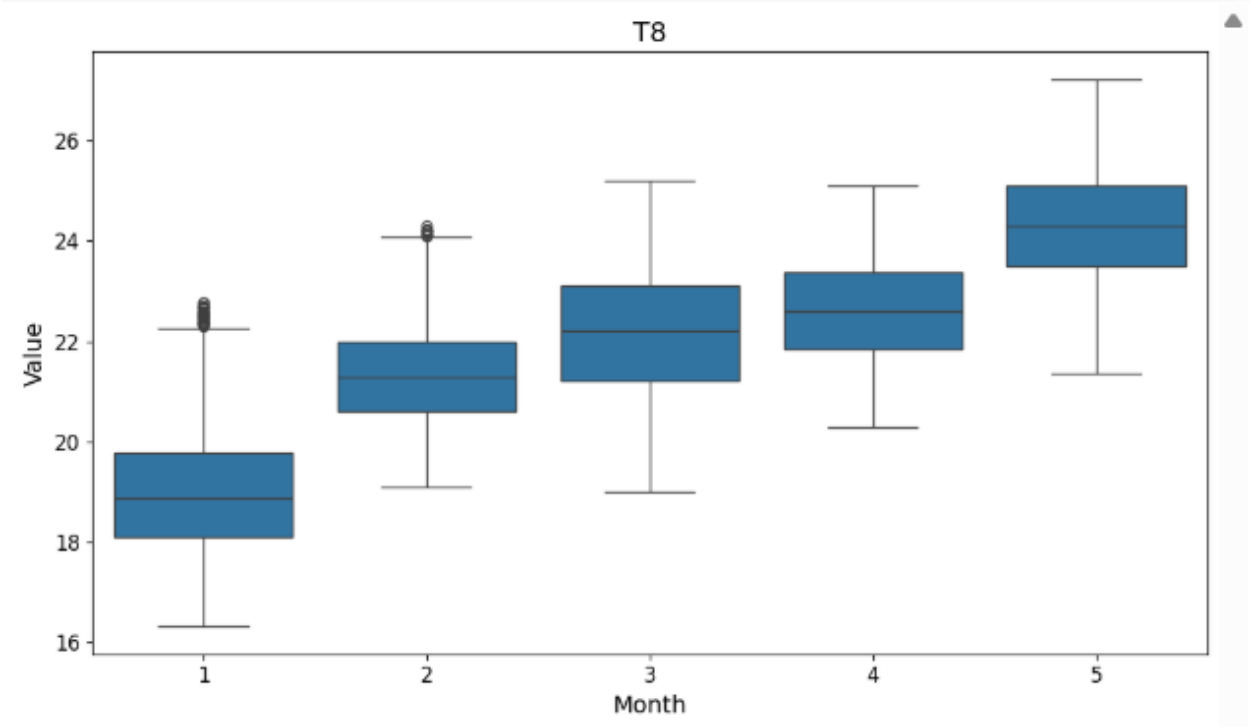

RH_5
Value
90
80
70
60
50
40
30
1
2
3
4
5
Month

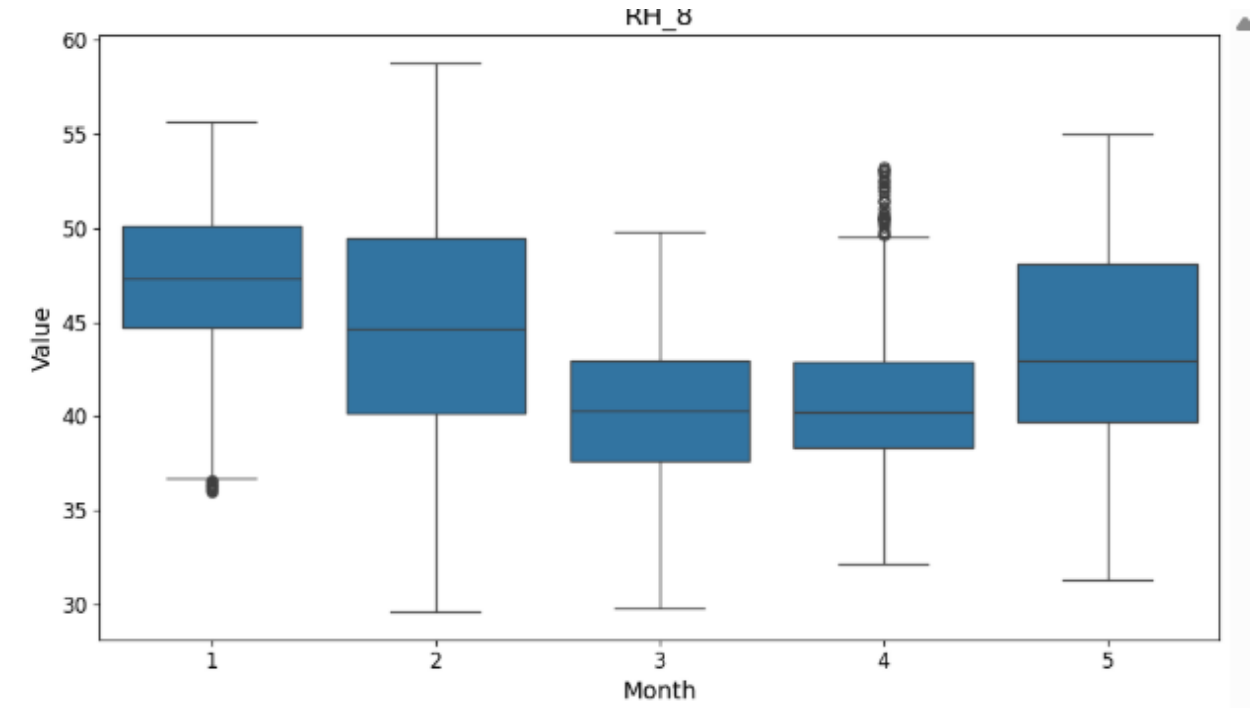

T6
Value
Month
30
25
20
15
10
5
0
−5
1
2
3
4
5

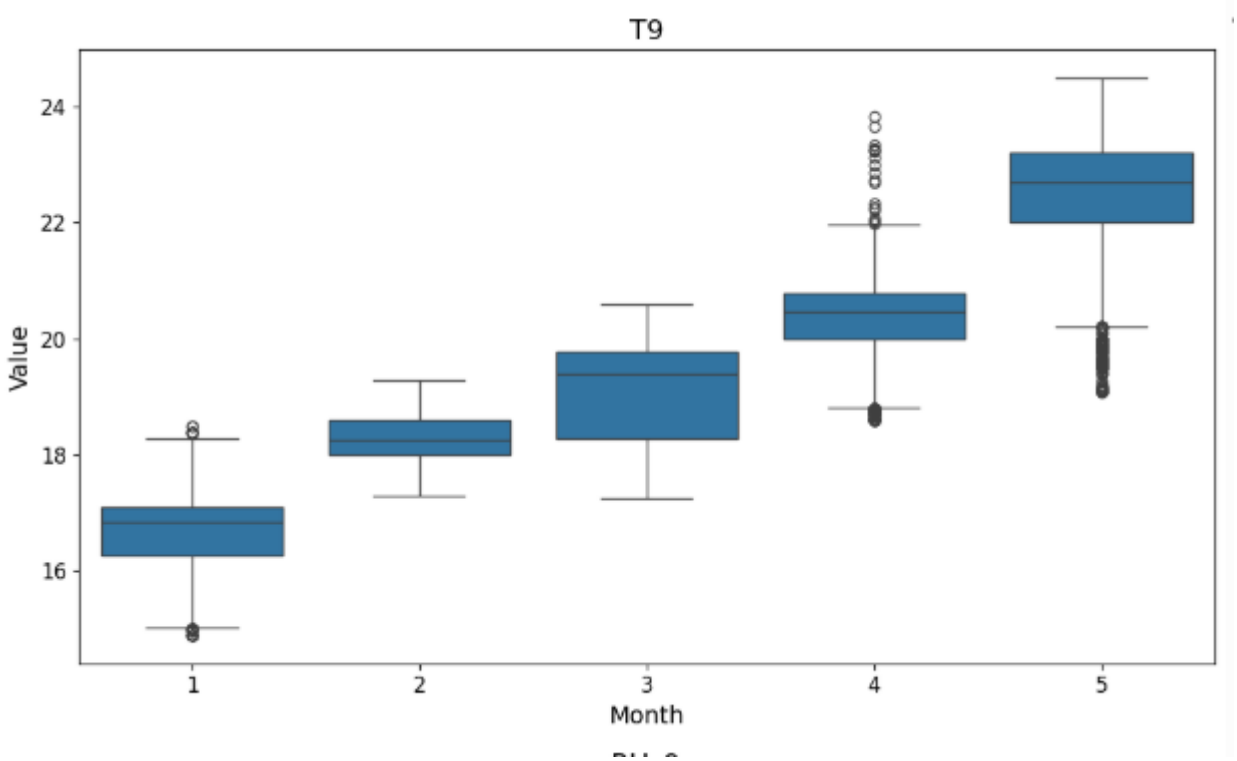

RH_6
Value
Month
100
80
60
40
20
0
1
2
3
4
5

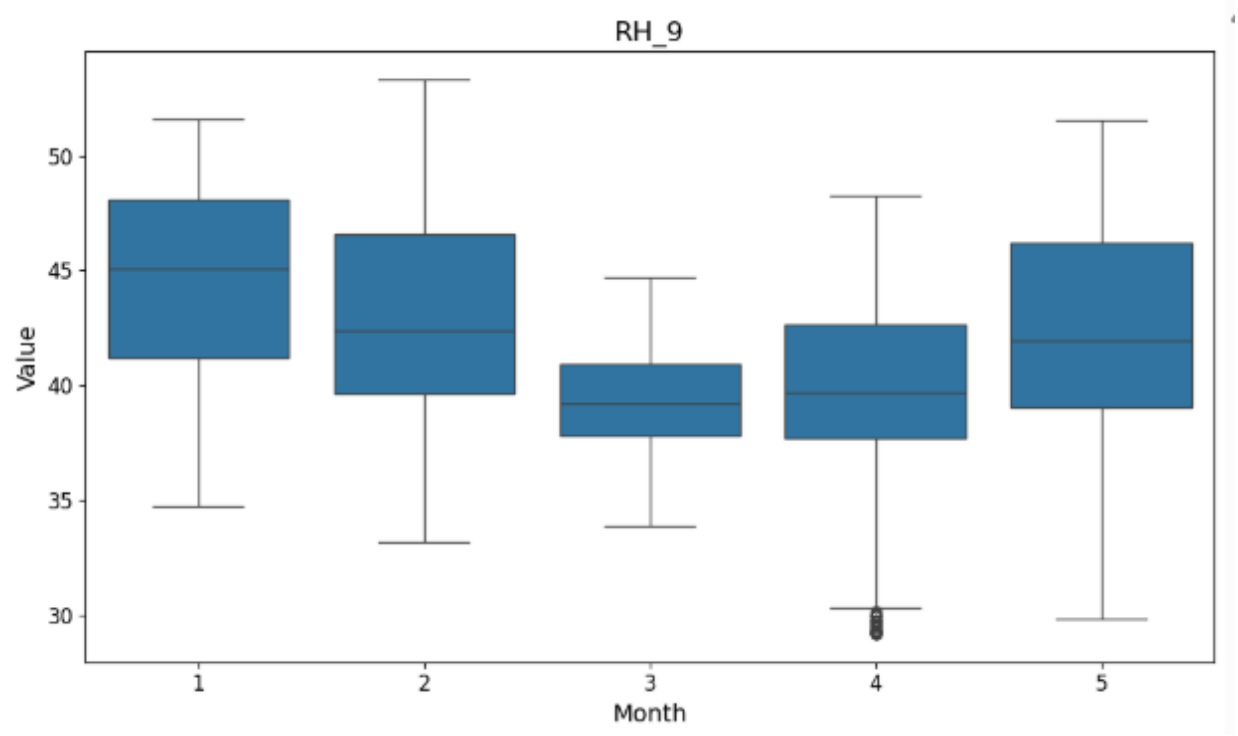

T7
Value
26
24
22
20
18
16
1
2
3
4
5
Month

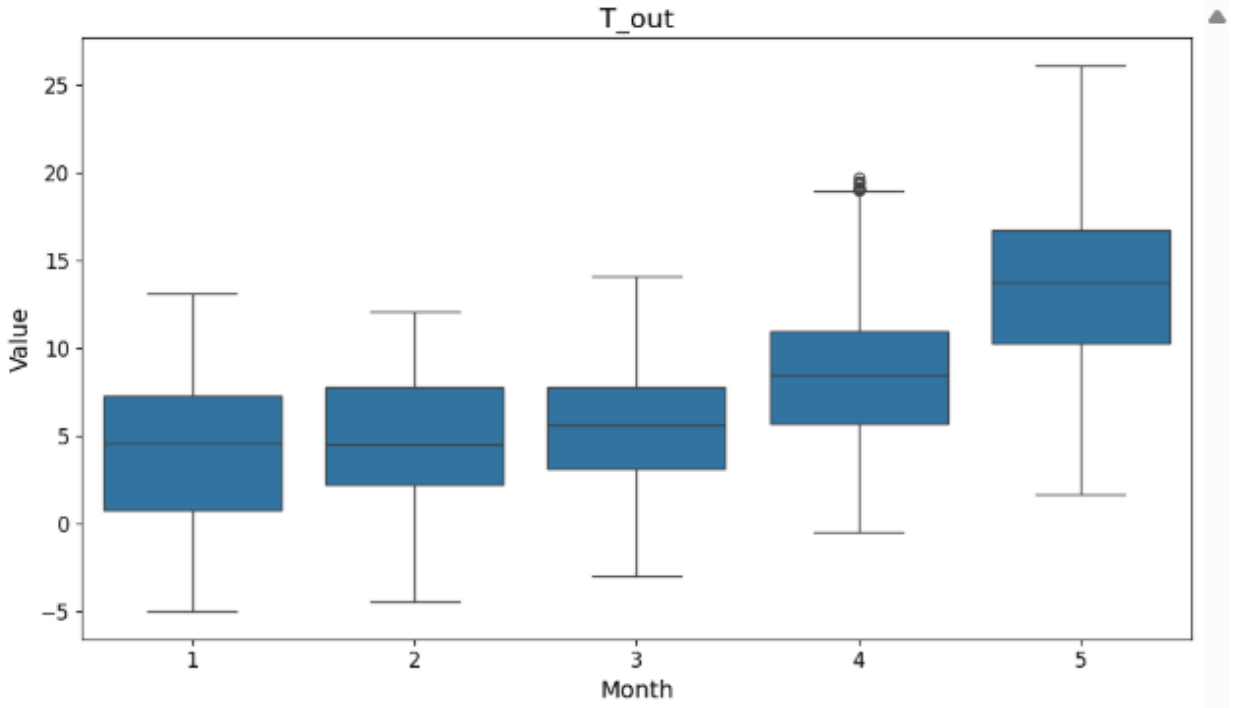

RH_7
Value
50
45
40
35
30
25
1
2
3
4
5
Month

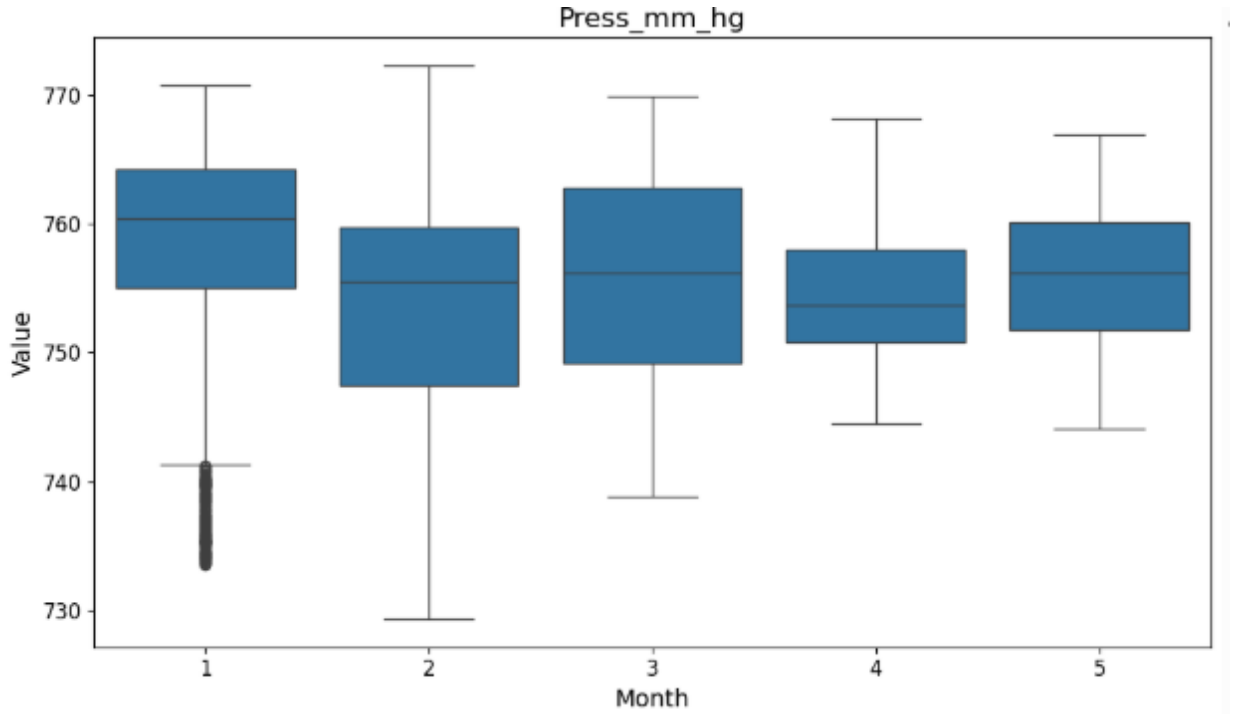

T8
Value
Month
16
18
20
22
24
26
1
2
3
4
5

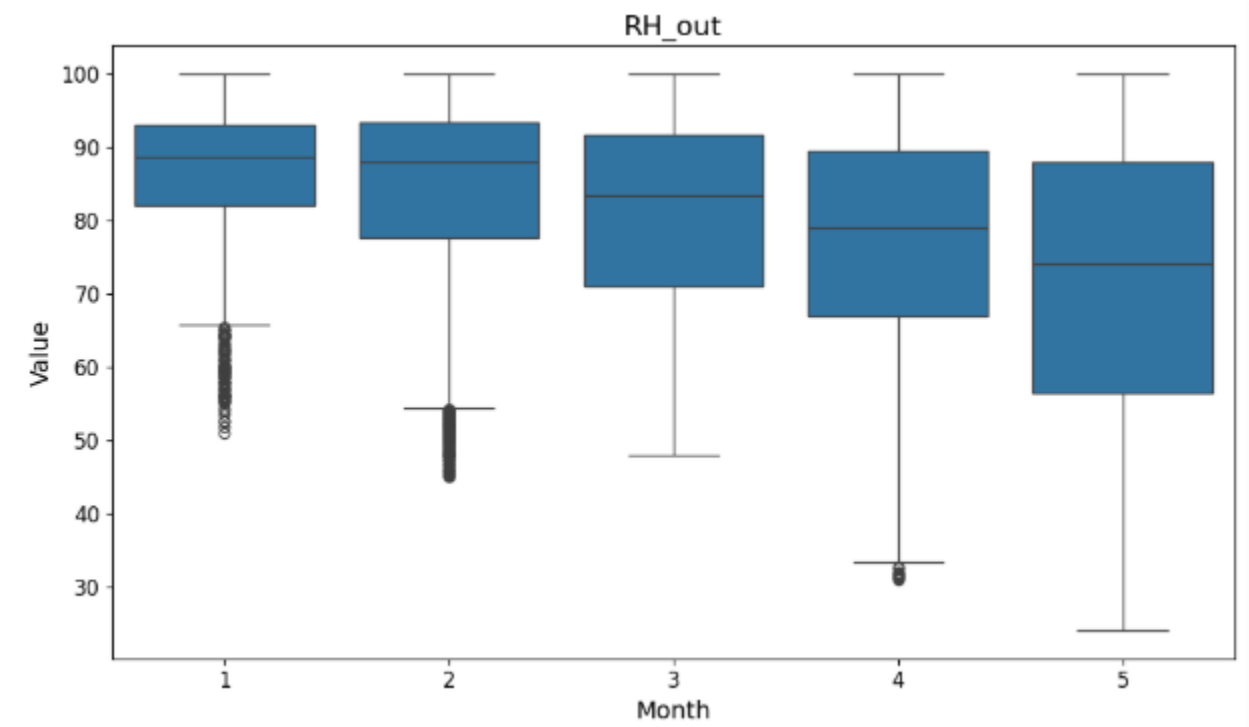

RH_8
Value
Month
30
35
40
45
50
55
60
1
2
3
4
5

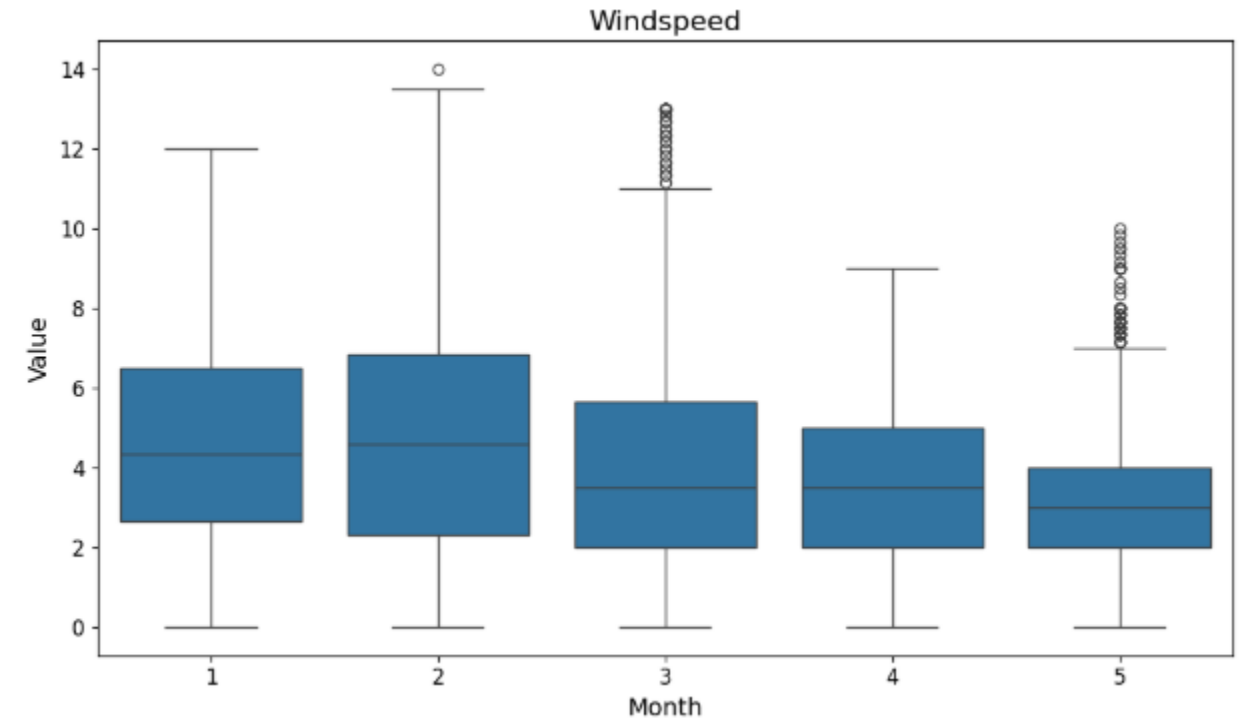

T9
Value
24
22
20
18
16
1
2
3
4
5
Month

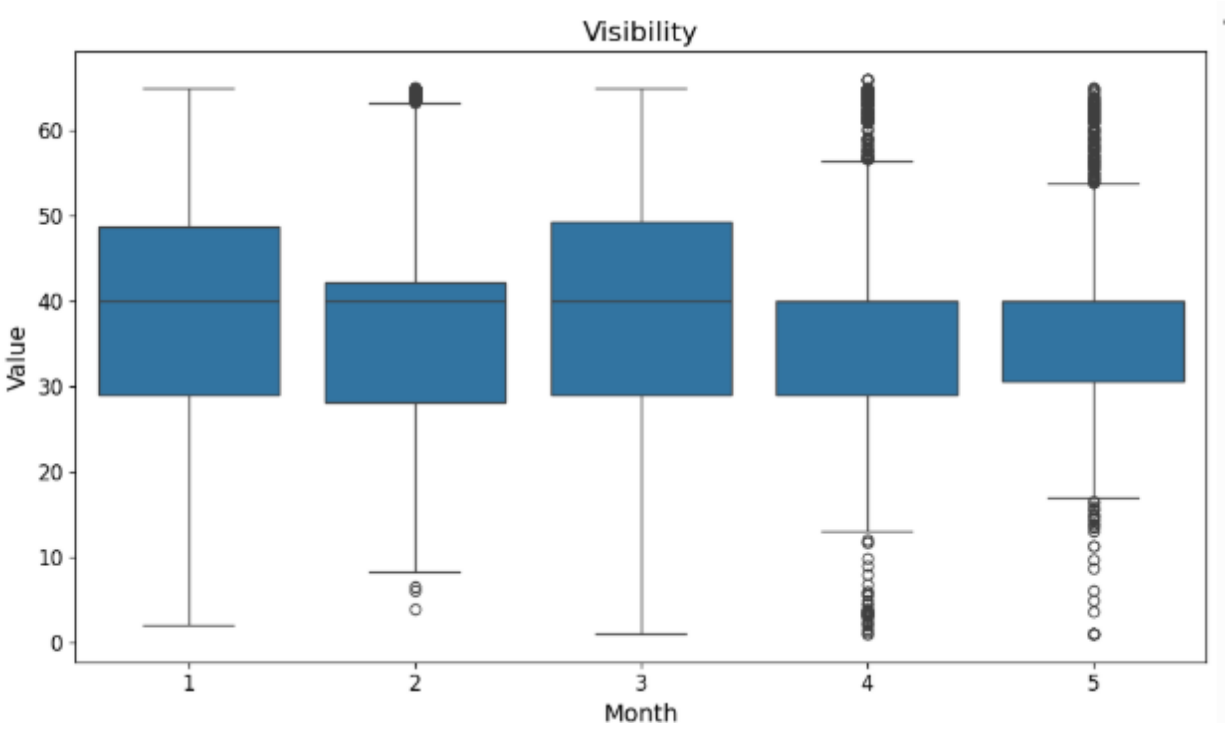

RH_9
Value
50
45
40
35
30
1
2
3
4
5
Month

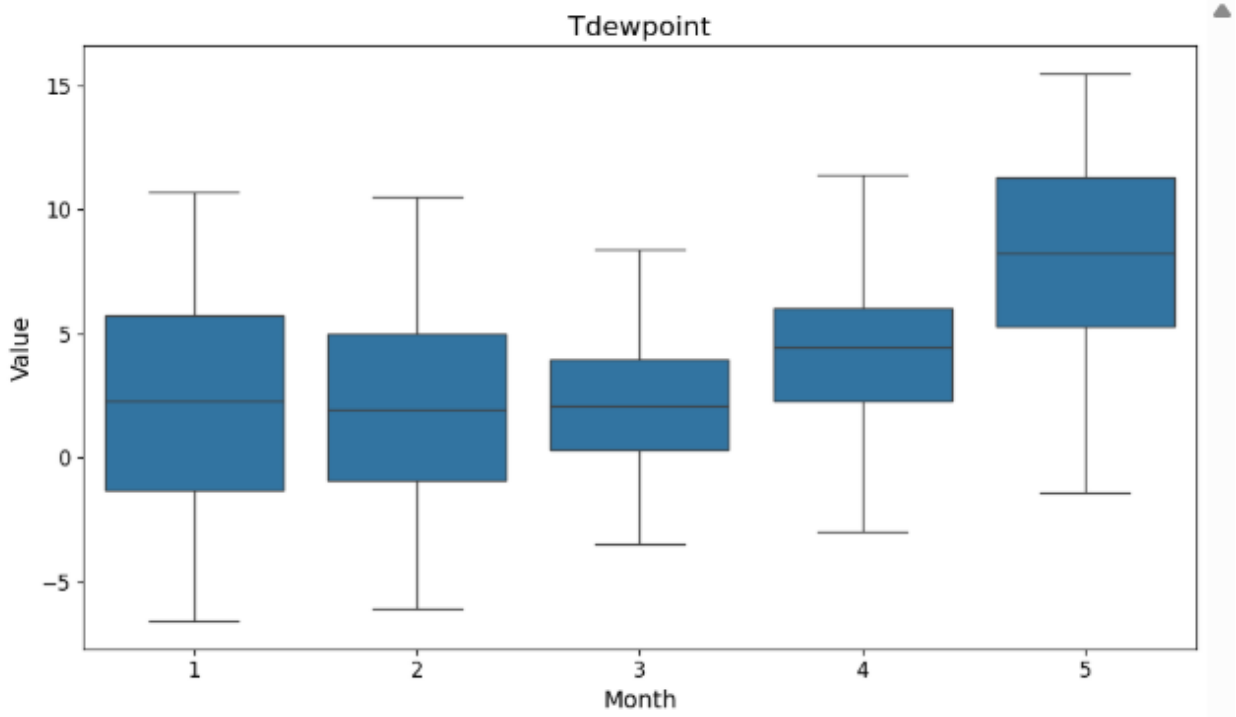

T_out
25
20
15
10
5
0
−5
Value
1
2
3
4
5
Month

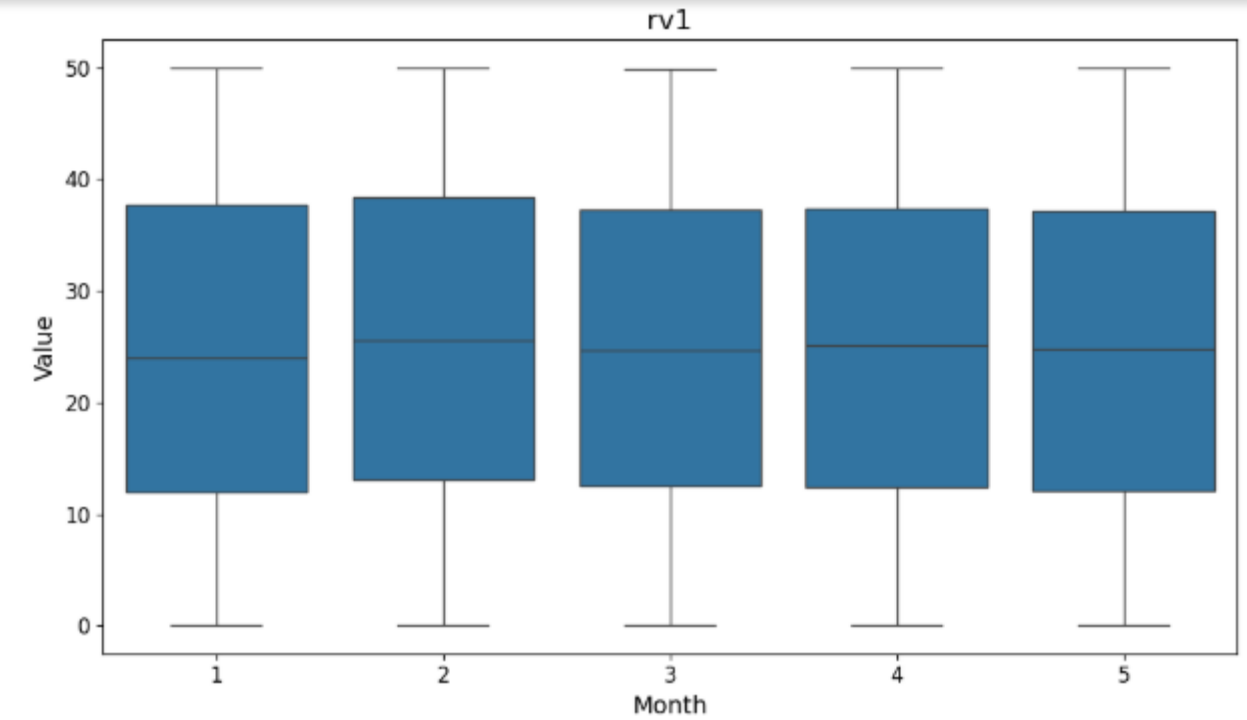

Press_mm_hg
770
760
750
740
730
Value
1
2
3
4
5
Month

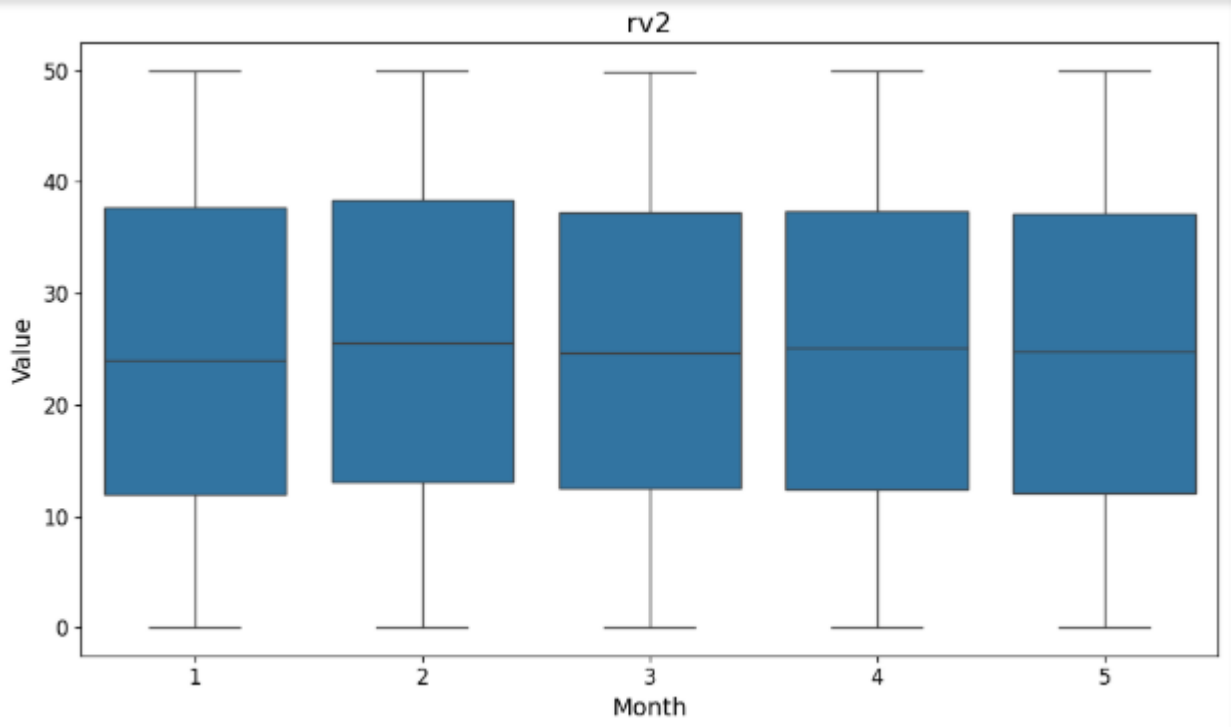

RH_out
Value
Month
100
90
80
70
60
50
40
30
1
2
3
4
5


Windspeed
Value
Month
14
12
10
8
6
4
2
0
1
2
3
4
5


Visibility
Value
Month
0
10
20
30
40
50
60
1
2
3
4
5

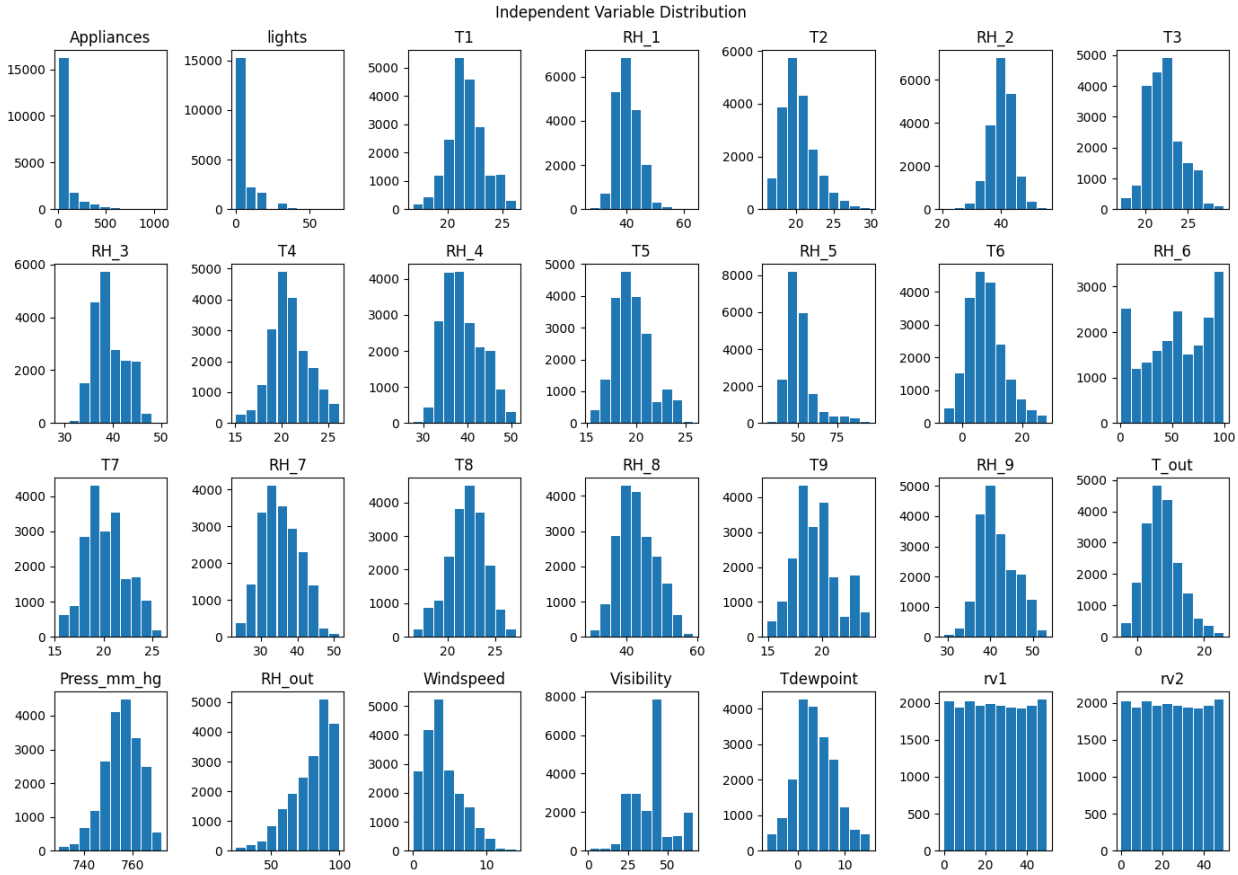

Tdewpoint
Value
Month
−5
0
5
10
15
1
2
3
4
5





Observations:

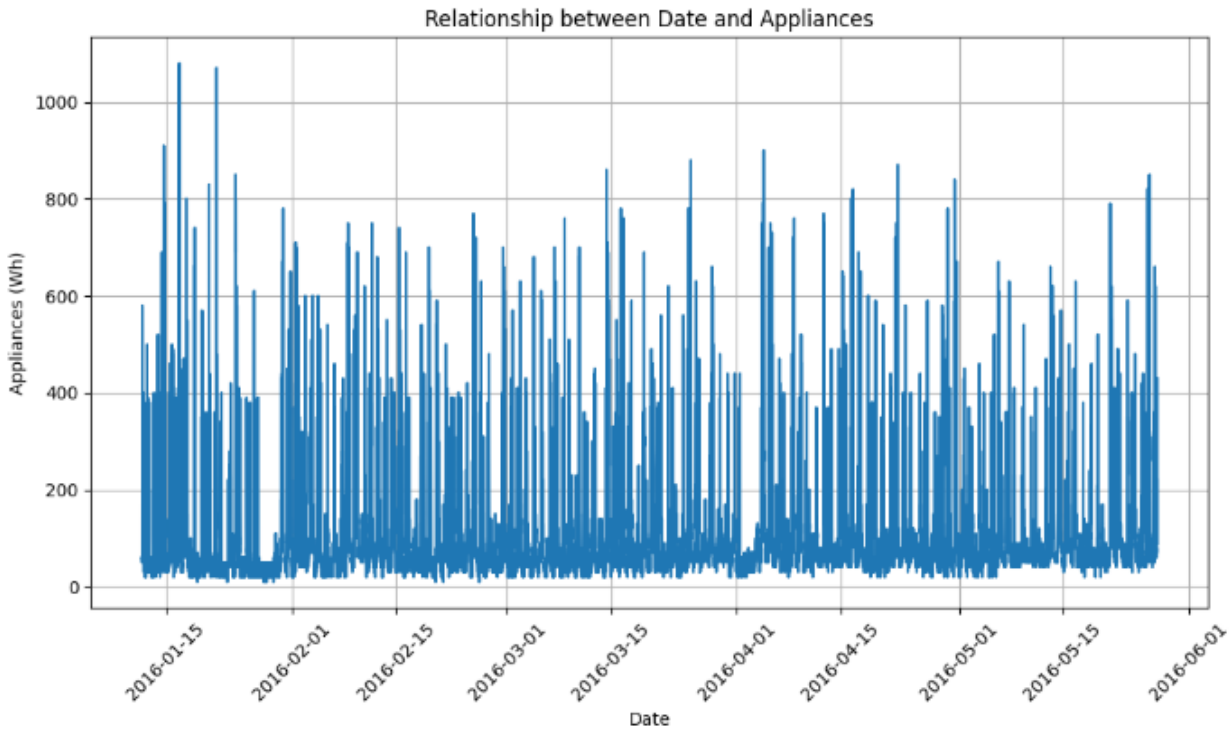
- For temperature (T), there is an overall increasing trend in the data.
- Other data show no significant features.



Observation:

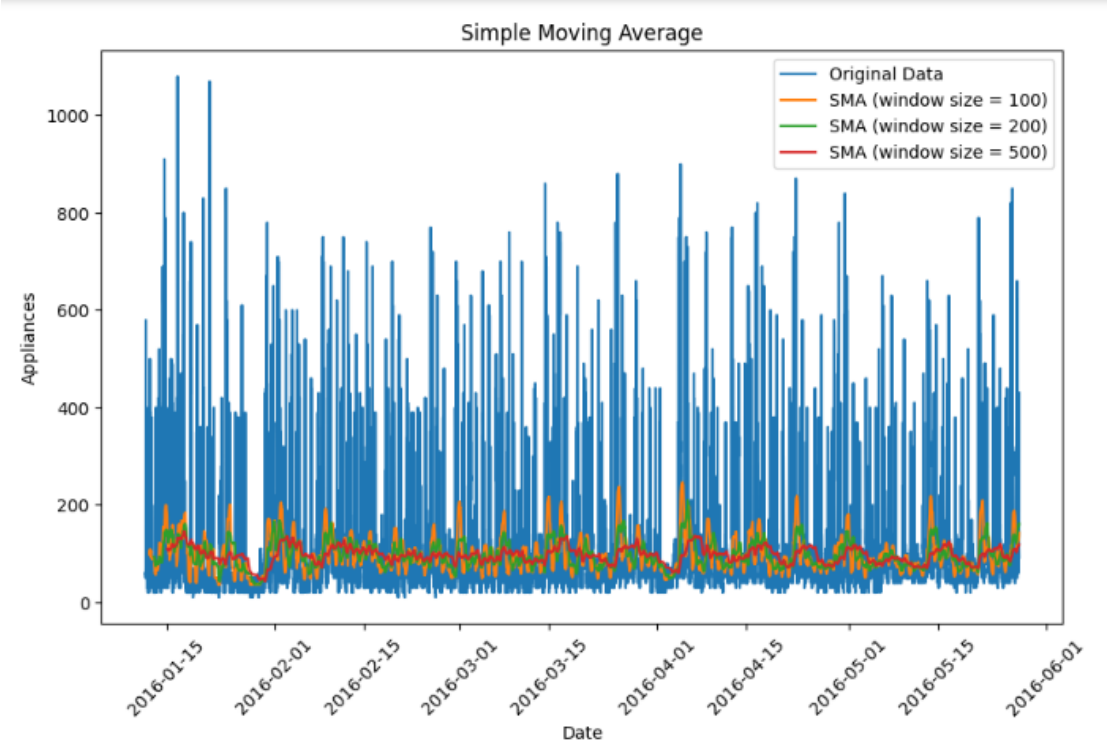
- Temperature and humidity exhibit a normal distribution, while other variables may be skewed or display multiple peaks.

A closer look at Appliances



Observation

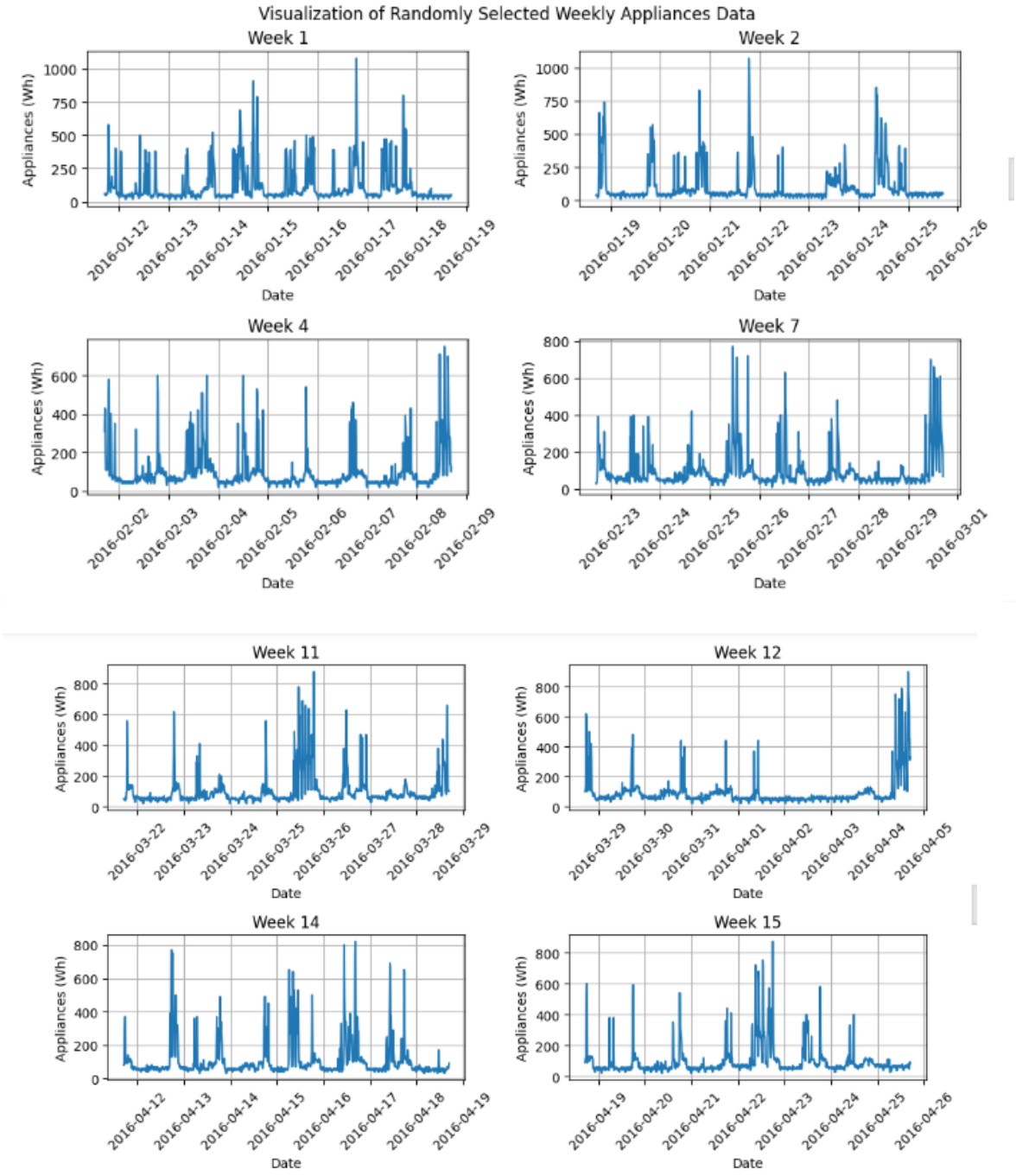
- From the plots, we can see that electricity usage (shown by the 'Appliances' variable) doesn't follow a clear or obvious pattern over time. But just because we can't easily see a pattern doesn’t mean there isn’t one. It means we need to look deeper and use data analysis techniques to find any hidden trends or relationships in the data.



Observation

- After using the Simple Moving Average (SMA) with window sizes of 100, 200, and 500 to smooth the data, the graphs didn't show any clear trends. This means that the 'Appliances' usage stayed mostly steady over time, without big changes or patterns.

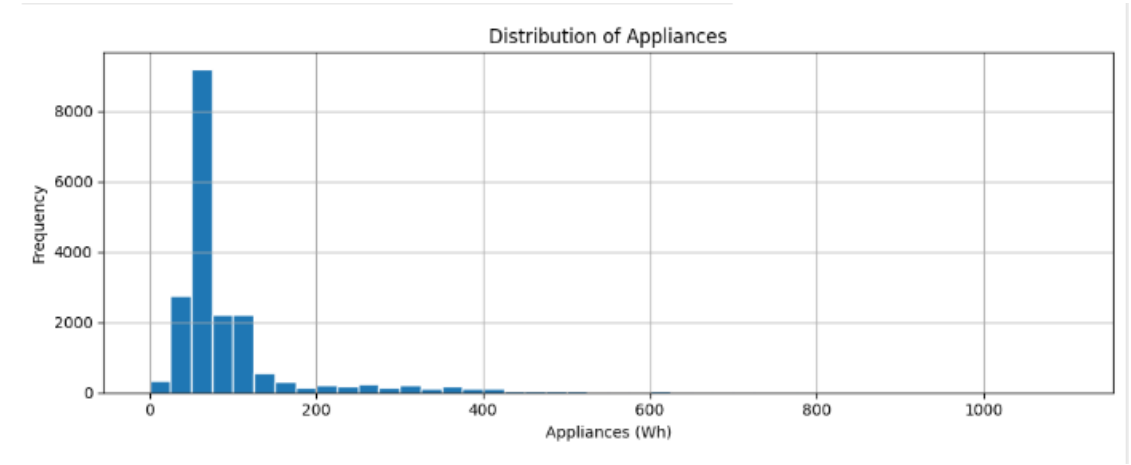
Visualization of Randomly Selected Weekly Data

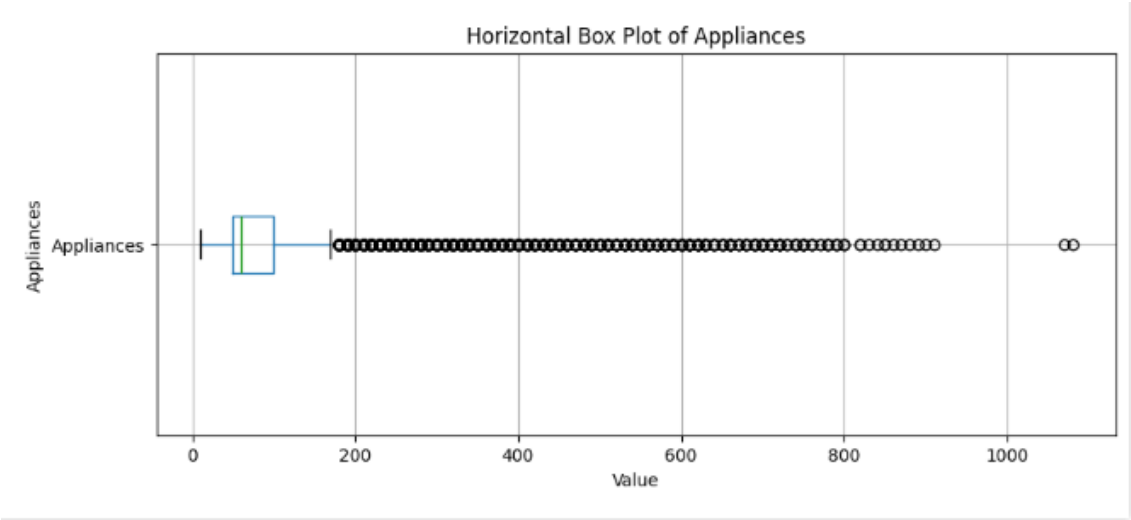


Observation

- When looking at a random week of data, we noticed a clear pattern showing regular ups and downs in energy use. Even though the week was picked at random, similar trends kept appearing, which means there are hidden habits or routines in how appliances are used. This matches what we saw earlier in the hourly analysis.

Plotting the histogram for the "Appliances" column

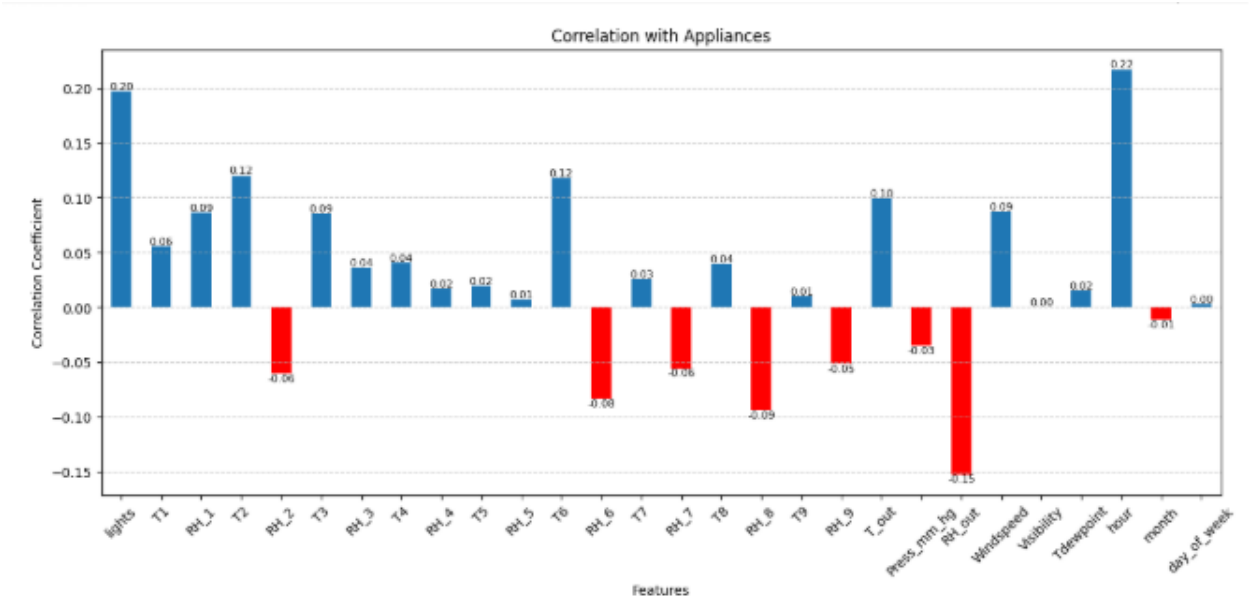




Observation

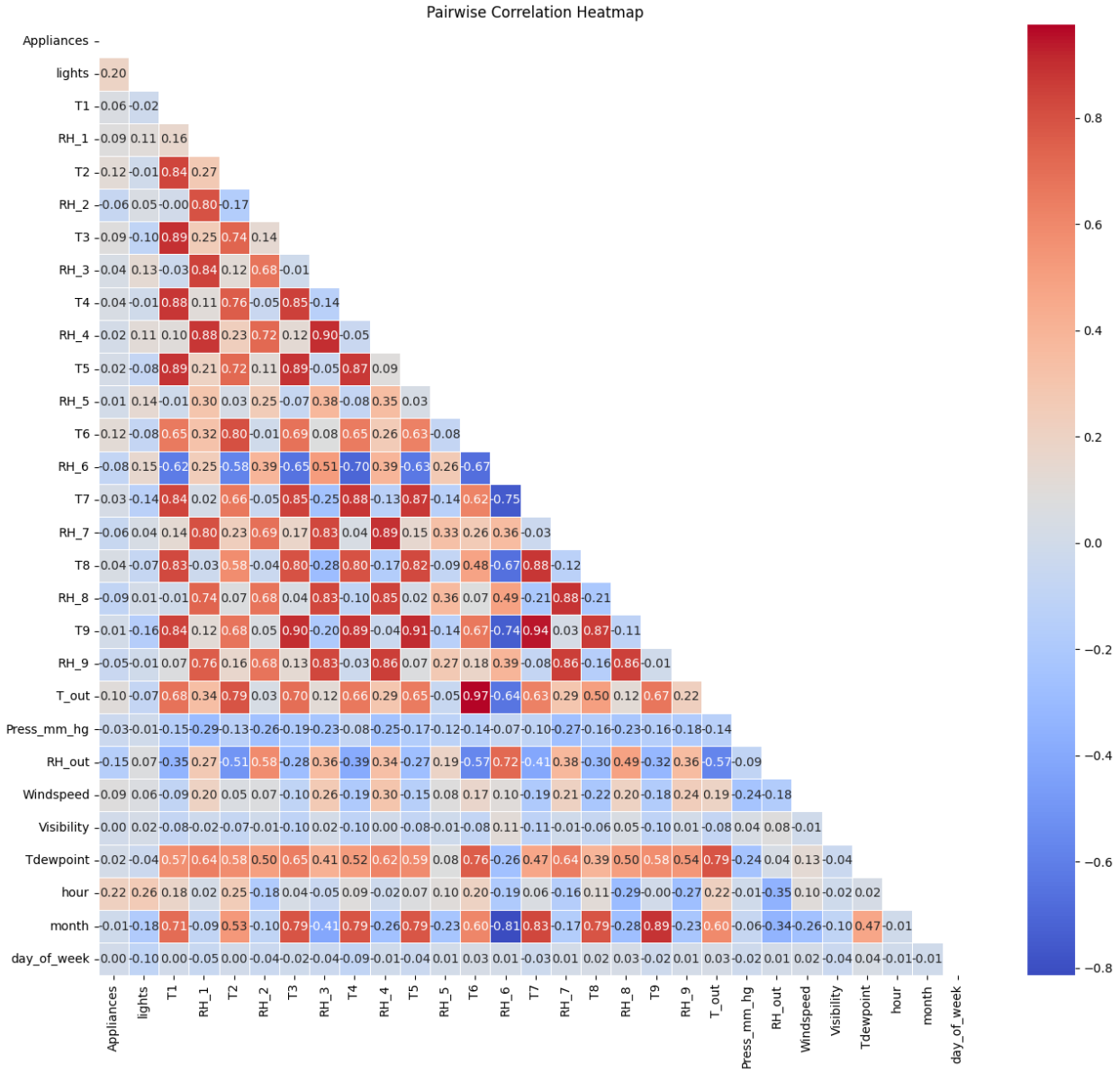
- The box plot has a long right-hand side tail, it indicates that it is positively skewed with some high energy use.

Correlation Analysis



Observation

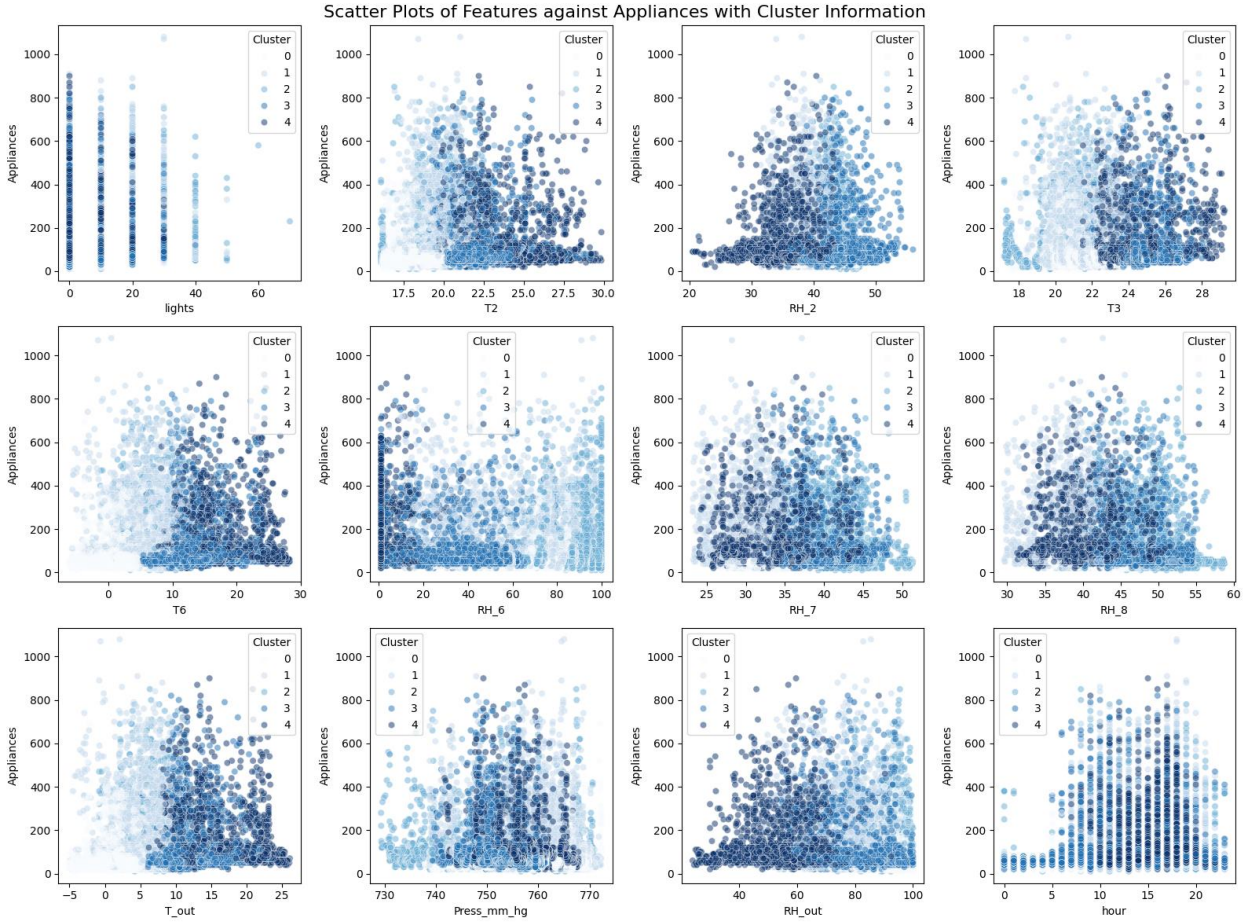
- The time of day shows the strongest link to appliance electricity use (correlation: 0.22), followed by the number of lights on (0.20). Air pressure has a weak negative correlation (-0.15), suggesting that appliance usage slightly decreases as air pressure rises.



Observation:

- This heatmap shows how different features are related to each other. We can see that temperatures are strongly linked with each other, and humidity levels also show some positive relationships, but not as strong.
- One interesting point is RH_6, which measures outdoor humidity. It has a strong negative relationship with temperature — meaning that when the temperature goes up, outdoor humidity tends to go down.
- On the other hand, our target feature, 'Appliances', doesn't show a strong relationship with any of the temperature or humidity features. So, these factors might not directly affect how much electricity appliances use.

Clustering Analysis



Observation

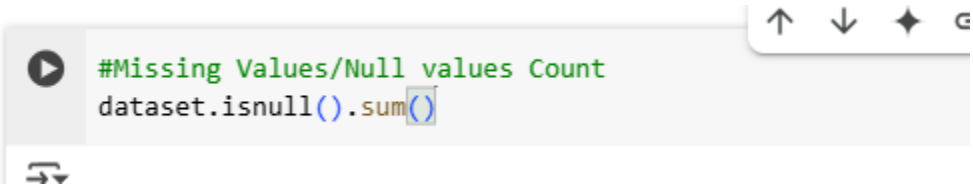
- The clusters in the data are clearly separated, especially in temperature-related plots, which show distinct groupings. Humidity-related plots show less clear separation. There's also a visible pattern related to lights. Overall, the data seems to be grouped well, making the classification look reliable.

Preprocessing

```
[ ]  #Checking for Duplicate Rows
     duplicate_count = dataset[dataset.duplicated()].shape
     duplicate_count
```

```
(0, 29)
```

- All rows are unique, and there are no duplicate rows in the dataset.

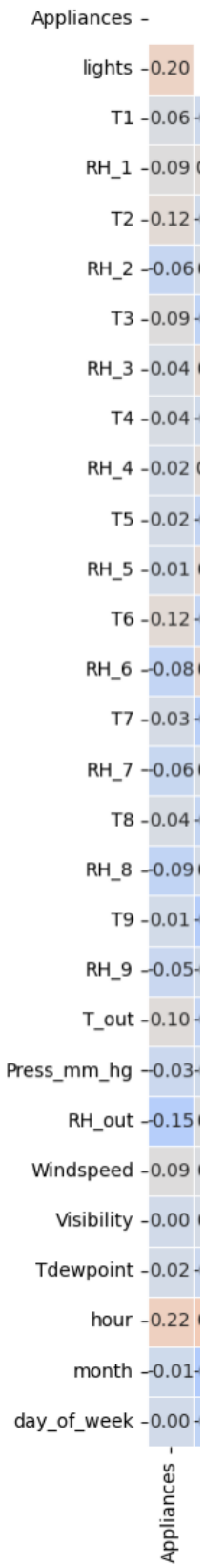


- There are no missing values in any of the columns.

```
# 4. Normalize the dataset using MinMaxScaler:
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_lstm_data = scaler.fit_transform(lstm_data_values)
```

Feature Engineering

The variables `Visibility` (-0.00) and `day_of_week` (-0.00) have near-zero correlation with the target (`Appliances`), indicating no meaningful linear relationship. Features with very low correlation add little predictive value and are typically excluded to simplify the model and avoid noise. RV1 and RV2 (likely these variables) were not selected for this reason.


Appliances
lights 0.20
T1 0.06
RH_1 0.09
T2 0.12
RH_2 0.06
T3 0.09
RH_3 0.04
T4 0.04
RH_4 0.02
T5 0.02
RH_5 0.01
T6 0.12
RH_6 0.08
T7 0.03
RH_7 0.06
T8 0.04
RH_8 0.09
T9 0.01
RH_9 0.05
T_out 0.10
Press_mm_hg 0.03
RH_out 0.15
Windspeed 0.09
Visibility 0.00
Tdewpoint 0.02
hour 0.22
month 0.01
day_of_week 0.00
Appliances

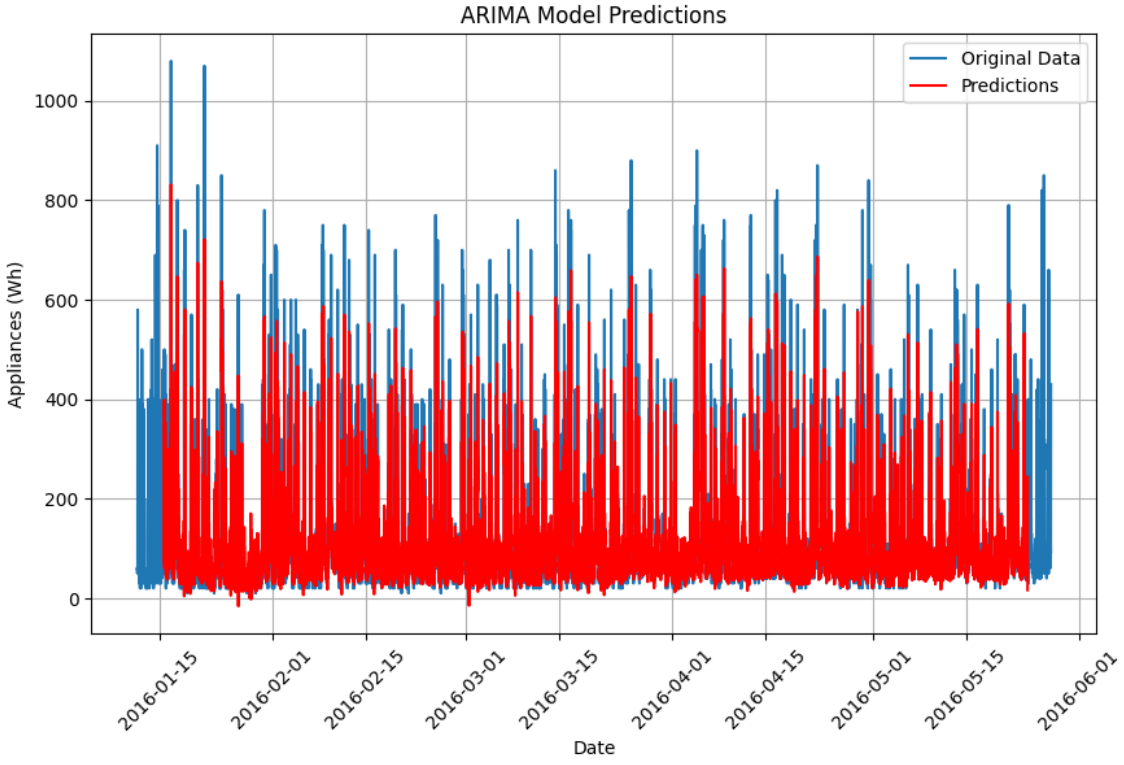
Model Design

The deep learning model used was an **LSTM-based sequential model** designed to capture temporal dependencies in energy usage data. The architecture included one LSTM layer with 64 units, followed by a Dropout layer to prevent overfitting, and a Dense output layer.

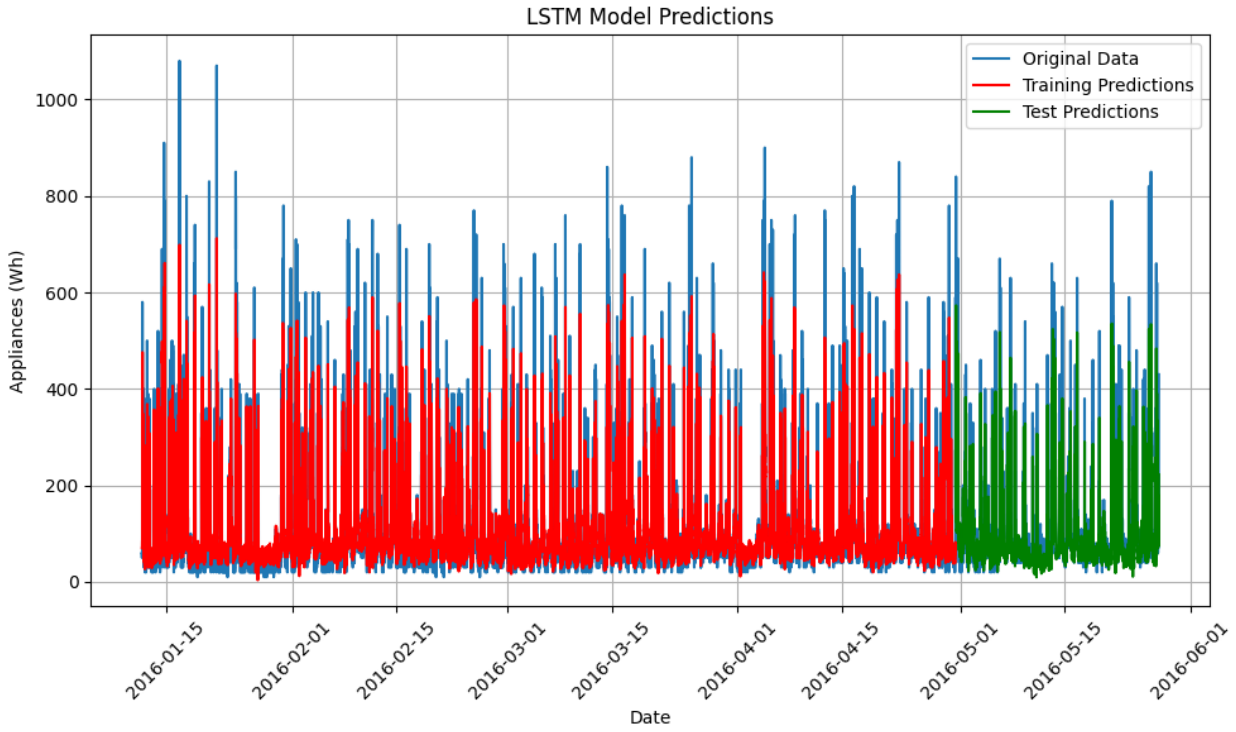
The **ReLU** activation function was used for non-linearity, and **Adam** optimizer was chosen for its efficiency in training deep networks. The model was compiled with **Mean Squared Error (MSE)** as the loss function, suitable for continuous value regression tasks like energy prediction.

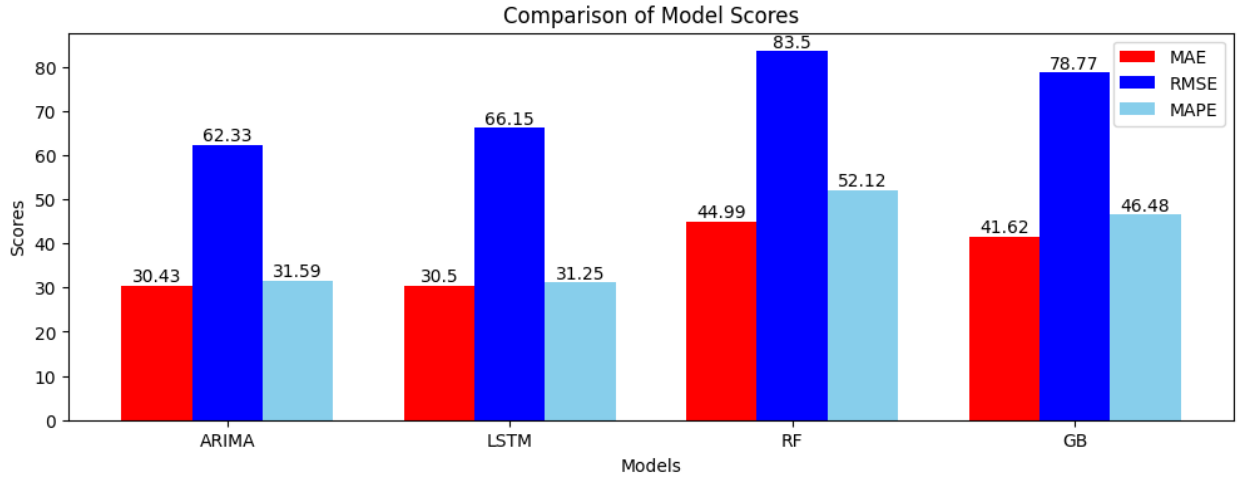
Results

ARIMA Model Predictions



LSTM Model Predictions





Model Optimization

Optimization Techniques

To improve the model, time-based **cyclical features** (sine/cosine transformations of hour and weekday) and **lag features** were engineered to capture seasonal and temporal patterns. Additionally, **hyperparameter tuning** was performed using **Keras Tuner** to optimize LSTM units, dropout rate, and learning rate.

After optimization, the LSTM model achieved:

- MAE: 27.98
- RMSE: 63.28
- MAPE: 23.71%
- R^2 Score: 0.52

These metrics show a notable improvement in prediction accuracy and generalization compared to initial untuned models.

Challenges and Solutions

This was my first time-series analysis project. I lacked a clear understanding of the process at first but overcame this by referring to online resources and revisiting concepts from previous prediction projects. I had to work with new models like ARIMA and LSTM, which I wasn't previously familiar with. While model performance was modest, this project laid a strong foundation, and with more time and exploration, I plan to further improve accuracy in future work.

Conclusion

Model scores of ARIMA and LSTM are very close, placing them in the top tier. Following closely are Gradient Boosting and Random Forest.

Model selection and parameter tuning are complex but crucial steps in time-series forecasting. Given the 10-minute interval data over 4+ months, both **ARIMA** and **LSTM** proved to be effective. ARIMA, with basic parameter tuning, achieved reasonable performance due to its ability to leverage time-series characteristics.

While traditional machine learning models like **Random Forest** and **Gradient Boosting** can be applied, they are generally more complex to implement in time-series contexts and yielded less optimal results in this case.

References

- Time Series Forecasting:

  Time Series Forecasting with LSTM Neural Networks (TensorFlow Tutorial)

- Feature Engineering for Time Series:

  Kaggle - Time Series Feature Engineering