Steam_Heating_Dataset_Analysis

September 4, 2021

[1]:
```
import pandas as pd
```

[2]:
```
data = pd.read_csv('C:/Users/alaga/Desktop/Elutions/Building_all_data.csv')
```

[3]:
```
data.head()
```

[3]:
```
      timestamp  Steam Demand (klbs/hr)  AHU1 (kW)  AHU2 (kW)  AHU3 (kW)  \
0  10/1/17 0:00                 12.3486    1.01875    6.63750    6.08750
1  10/1/17 0:05                 12.3486    1.00625    6.70625    6.21875
2  10/1/17 0:10                  9.5198    1.01250    6.71250    6.21875
3  10/1/17 0:15                  5.2766    1.01875    6.71875    6.21875
4  10/1/17 0:20                  5.2766    1.00625    6.76250    6.21875

   AHU4 (kW)  Temperature (F)
0   4.883333             57.9
1   4.950000              NaN
2   4.945833              NaN
3   4.929167              NaN
4   4.929167              NaN
```

[4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35424 entries, 0 to 35423
Data columns (total 7 columns):
timestamp                 35424 non-null object
Steam Demand (klbs/hr)    32292 non-null float64
AHU1 (kW)                 34360 non-null float64
AHU2 (kW)                 34362 non-null float64
AHU3 (kW)                 34363 non-null float64
AHU4 (kW)                 34365 non-null float64
Temperature (F)           1441 non-null float64
dtypes: float64(6), object(1)
memory usage: 1.9+ MB
```

[5]:
```
def perc_missing(df):
    '''prints out columns with missing values with its %'''
    for col in df.columns:
        pct = df[col].isna().mean() * 100
        if (pct != 0):
            print('{} => {}%'.format(col, round(pct, 2)))

perc_missing(data)
```

```
Steam Demand (klbs/hr) => 8.84%
AHU1 (kW) => 3.0%
AHU2 (kW) => 3.0%
AHU3 (kW) => 3.0%
AHU4 (kW) => 2.99%
Temperature (F) => 95.93%
```

[6]:

```
data.describe()
```

[6]:

```
       Steam Demand (klbs/hr)     AHU1 (kW)     AHU2 (kW)     AHU3 (kW)  \
count            32292.000000  34360.000000  34362.000000  34363.000000
mean               904.219129      6.764803      5.180266      5.122441
std                635.462415      4.071454      3.391569      3.022818
min                  0.000000      0.918750      0.818750      0.937500
25%                578.199892      2.556250      1.987500      2.143750
50%                846.396670      6.231250      5.587500      5.731250
75%               1275.868400      9.937500      7.193750      7.306250
max               2872.947800     22.162500     30.100000     20.643749

          AHU4 (kW)  Temperature (F)
count  34365.000000      1441.000000
mean       5.054626        47.057529
std        2.278577        15.474195
min        0.829167        12.900000
25%        3.020833        36.000000
50%        6.062500        48.000000
75%        6.695833        59.000000
max       22.016666        84.900000
```

[7]:

```
data['timestamp'] = pd.to_datetime(data['timestamp'],␣
 ↪infer_datetime_format=True)
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35424 entries, 0 to 35423
Data columns (total 7 columns):
timestamp                 35424 non-null datetime64[ns]
Steam Demand (klbs/hr)    32292 non-null float64
AHU1 (kW)                 34360 non-null float64
AHU2 (kW)                 34362 non-null float64
AHU3 (kW)                 34363 non-null float64
AHU4 (kW)                 34365 non-null float64
```

```
Temperature (F)           1441 non-null float64
dtypes: datetime64[ns](1), float64(6)
memory usage: 1.9 MB
```

[8]:

```
new_df = data
new_df.head()
```

[8]:

```
            timestamp  Steam Demand (klbs/hr)  AHU1 (kW)  AHU2 (kW)  \
0 2017-10-01 00:00:00                 12.3486    1.01875    6.63750
1 2017-10-01 00:05:00                 12.3486    1.00625    6.70625
2 2017-10-01 00:10:00                  9.5198    1.01250    6.71250
3 2017-10-01 00:15:00                  5.2766    1.01875    6.71875
4 2017-10-01 00:20:00                  5.2766    1.00625    6.76250

   AHU3 (kW)  AHU4 (kW)  Temperature (F)
0    6.08750   4.883333             57.9
1    6.21875   4.950000              NaN
2    6.21875   4.945833              NaN
3    6.21875   4.929167              NaN
4    6.21875   4.929167              NaN
```

[9]:

```
new_df.drop('Temperature (F)', axis=1, inplace=True)
new_df.head()
```

[9]:

```
            timestamp  Steam Demand (klbs/hr)  AHU1 (kW)  AHU2 (kW)  \
0 2017-10-01 00:00:00                 12.3486    1.01875    6.63750
1 2017-10-01 00:05:00                 12.3486    1.00625    6.70625
2 2017-10-01 00:10:00                  9.5198    1.01250    6.71250
3 2017-10-01 00:15:00                  5.2766    1.01875    6.71875
4 2017-10-01 00:20:00                  5.2766    1.00625    6.76250

   AHU3 (kW)  AHU4 (kW)
0    6.08750   4.883333
1    6.21875   4.950000
2    6.21875   4.945833
3    6.21875   4.929167
4    6.21875   4.929167
```

[10]:

```
new_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35424 entries, 0 to 35423
Data columns (total 6 columns):
timestamp                 35424 non-null datetime64[ns]
Steam Demand (klbs/hr)    32292 non-null float64
AHU1 (kW)                 34360 non-null float64
AHU2 (kW)                 34362 non-null float64
AHU3 (kW)                 34363 non-null float64
AHU4 (kW)                 34365 non-null float64
```

```
dtypes: datetime64[ns](1), float64(5)
memory usage: 1.6 MB
```

[11]:

```
new_df.bfill(inplace=True)
new_df.isnull().sum()
```

[11]:

```
timestamp                0
Steam Demand (klbs/hr)   0
AHU1 (kW)                0
AHU2 (kW)                0
AHU3 (kW)                0
AHU4 (kW)                0
dtype: int64
```

[12]:

```
new_df = new_df.set_index('timestamp')
new_df.head()
```

[12]:

```
                     Steam Demand (klbs/hr)  AHU1 (kW)  AHU2 (kW)  AHU3 (kW)  \
timestamp
2017-10-01 00:00:00                 12.3486    1.01875    6.63750    6.08750
2017-10-01 00:05:00                 12.3486    1.00625    6.70625    6.21875
2017-10-01 00:10:00                  9.5198    1.01250    6.71250    6.21875
2017-10-01 00:15:00                  5.2766    1.01875    6.71875    6.21875
2017-10-01 00:20:00                  5.2766    1.00625    6.76250    6.21875

                     AHU4 (kW)
timestamp
2017-10-01 00:00:00   4.883333
2017-10-01 00:05:00   4.950000
2017-10-01 00:10:00   4.945833
2017-10-01 00:15:00   4.929167
2017-10-01 00:20:00   4.929167
```

[13]:

```
new_df['Month'] =  new_df.index.month
new_df['Weekday Name'] =  new_df.index.weekday_name
new_df['Hour'] =  new_df.index.hour
new_df.sample(5, random_state=0)
```

[13]:

```
                     Steam Demand (klbs/hr)  AHU1 (kW)  AHU2 (kW)  AHU3 (kW)  \
timestamp
2018-01-11 13:00:00               728.16785  16.143749   20.18125   15.02500
2018-01-10 21:35:00              1552.89230  16.468750   14.86250   10.06875
2017-12-10 22:10:00              1438.13530   5.950000    3.89375    3.65625
2017-11-25 01:15:00              1072.43970   2.775000    1.95000    2.12500
2018-01-10 13:30:00              1284.59640  18.137501   13.13125    7.40625

                     AHU4 (kW)  Month Weekday Name  Hour
timestamp
2018-01-11 13:00:00   9.141666      1     Thursday    13
2018-01-10 21:35:00   8.862500      1    Wednesday    21
```

```
2017-12-10 22:10:00   5.600000     12       Sunday    22
2017-11-25 01:15:00   2.887500     11     Saturday     1
2018-01-10 13:30:00   9.875000      1    Wednesday    13
```

[14]:
```python
new_df['Day'] =  new_df.index.day
new_df.sample(5, random_state=0)
```

[14]:
```
                     Steam Demand (klbs/hr)  AHU1 (kW)  AHU2 (kW)  AHU3 (kW)  \
timestamp
2018-01-11 13:00:00               728.16785  16.143749   20.18125   15.02500
2018-01-10 21:35:00              1552.89230  16.468750   14.86250   10.06875
2017-12-10 22:10:00              1438.13530   5.950000    3.89375    3.65625
2017-11-25 01:15:00              1072.43970   2.775000    1.95000    2.12500
2018-01-10 13:30:00              1284.59640  18.137501   13.13125    7.40625

                     AHU4 (kW)  Month Weekday Name  Hour  Day
timestamp
2018-01-11 13:00:00   9.141666      1     Thursday    13   11
2018-01-10 21:35:00   8.862500      1    Wednesday    21   10
2017-12-10 22:10:00   5.600000     12       Sunday    22   10
2017-11-25 01:15:00   2.887500     11     Saturday     1   25
2018-01-10 13:30:00   9.875000      1    Wednesday    13   10
```

[15]:
```python
DF = new_df
DF.head()
```

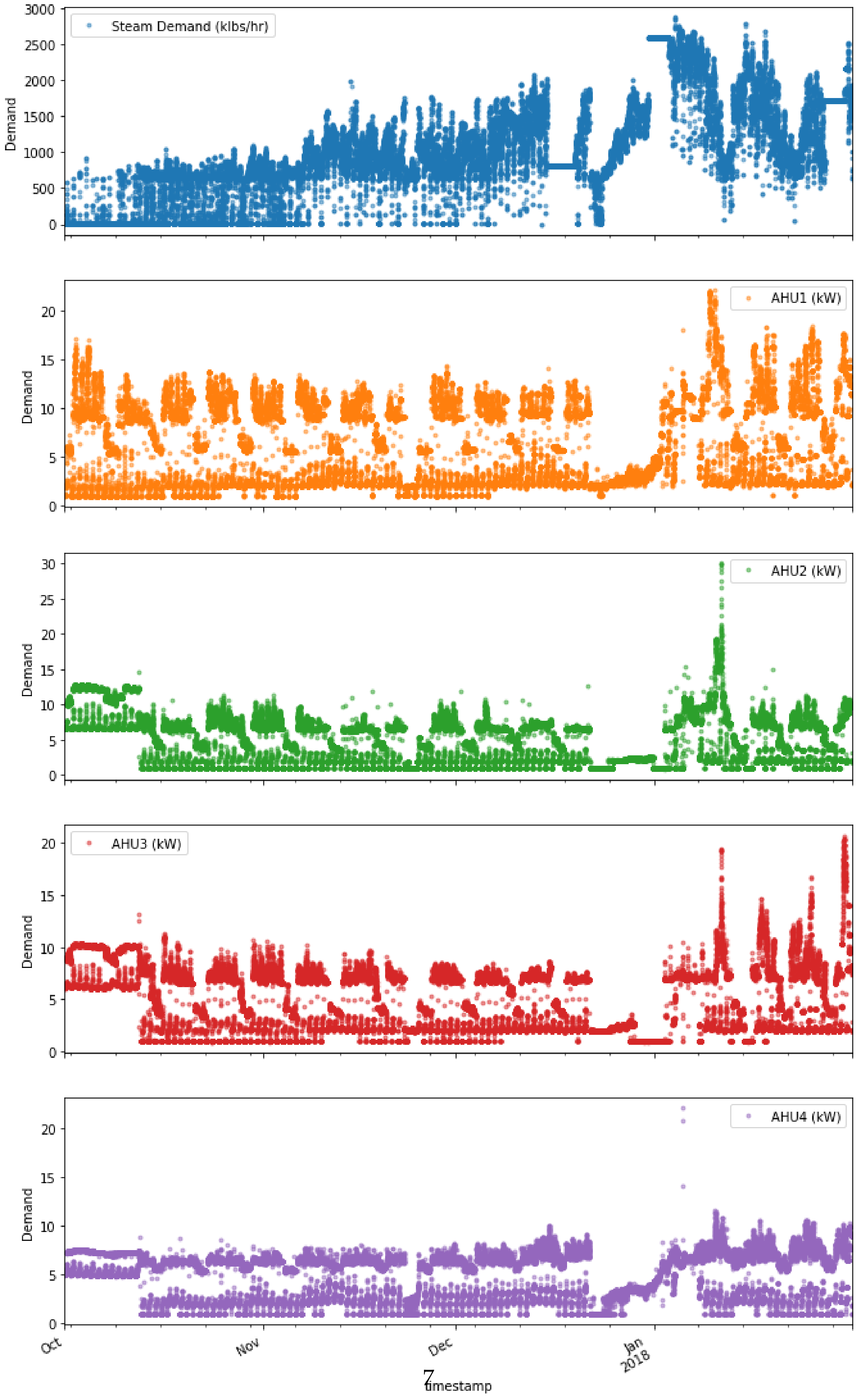
[15]:
```
                     Steam Demand (klbs/hr)  AHU1 (kW)  AHU2 (kW)  AHU3 (kW)  \
timestamp
2017-10-01 00:00:00                 12.3486    1.01875    6.63750    6.08750
2017-10-01 00:05:00                 12.3486    1.00625    6.70625    6.21875
2017-10-01 00:10:00                  9.5198    1.01250    6.71250    6.21875
2017-10-01 00:15:00                  5.2766    1.01875    6.71875    6.21875
2017-10-01 00:20:00                  5.2766    1.00625    6.76250    6.21875

                     AHU4 (kW)  Month Weekday Name  Hour  Day
timestamp
2017-10-01 00:00:00   4.883333     10       Sunday     0    1
2017-10-01 00:05:00   4.950000     10       Sunday     0    1
2017-10-01 00:10:00   4.945833     10       Sunday     0    1
2017-10-01 00:15:00   4.929167     10       Sunday     0    1
2017-10-01 00:20:00   4.929167     10       Sunday     0    1
```

[16]:
```python
import matplotlib.pyplot as plt
```

[17]:
```python
import seaborn as sns
```

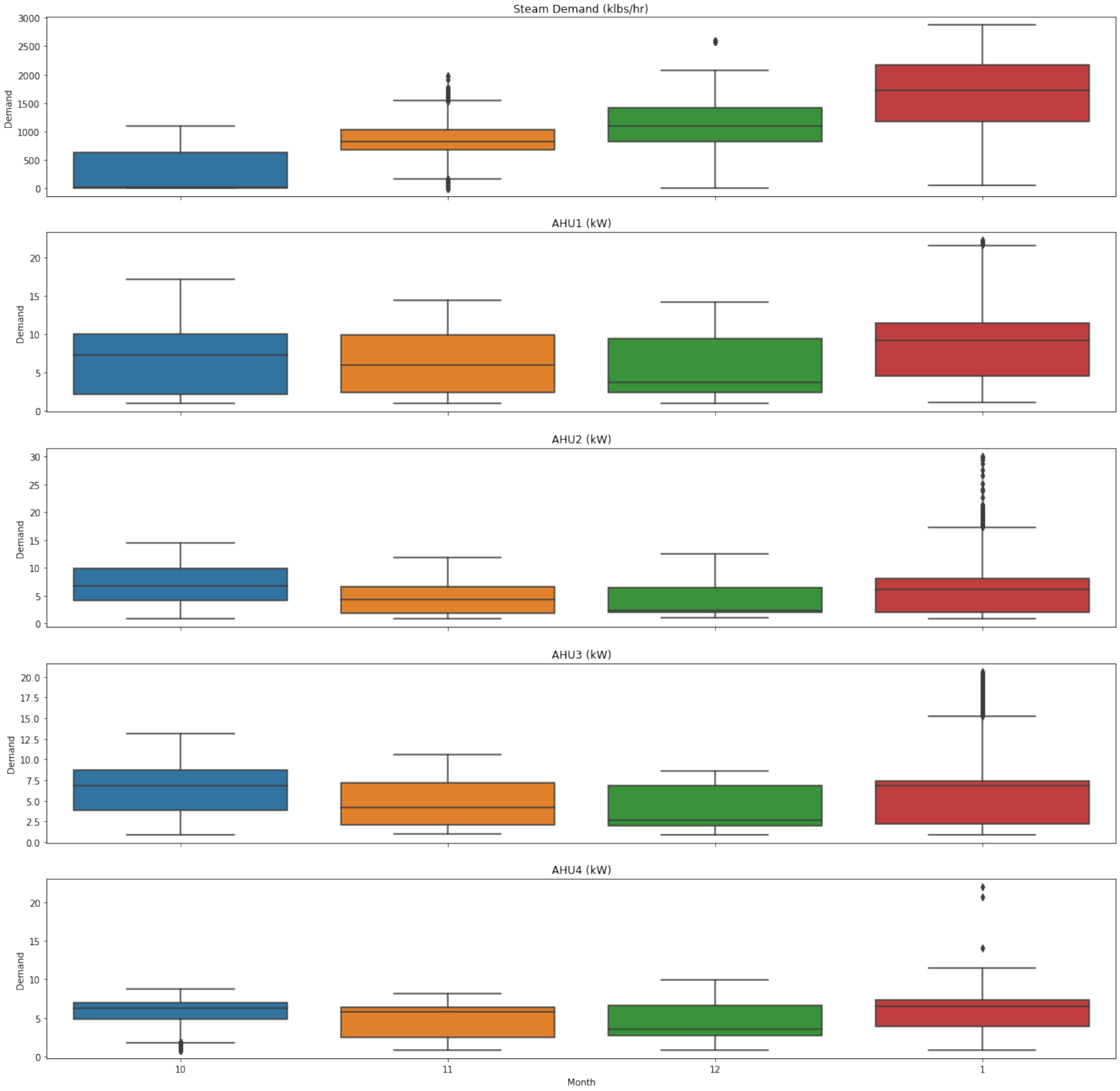
[18]:
```python
cols_plot = ['Steam Demand (klbs/hr)', 'AHU1 (kW)', 'AHU2 (kW)', 'AHU3 (kW)',␣
 ↪'AHU4 (kW)']
```

```
axes = DF[cols_plot].plot(marker='.', alpha=0.5, linestyle='None', figsize=(11,␣
 ↪21), subplots=True)

for ax in axes:
    ax.set_ylabel('Demand')
```


Steam Demand (klbs/hr)
AHU1 (kW)
AHU2 (kW)
AHU3 (kW)
AHU4 (kW)
Demand
3000
2500
2000
1500
1000
500
0
20
15
10
5
30
25
Oct
Nov
Dec
Jan
2018
timestamp

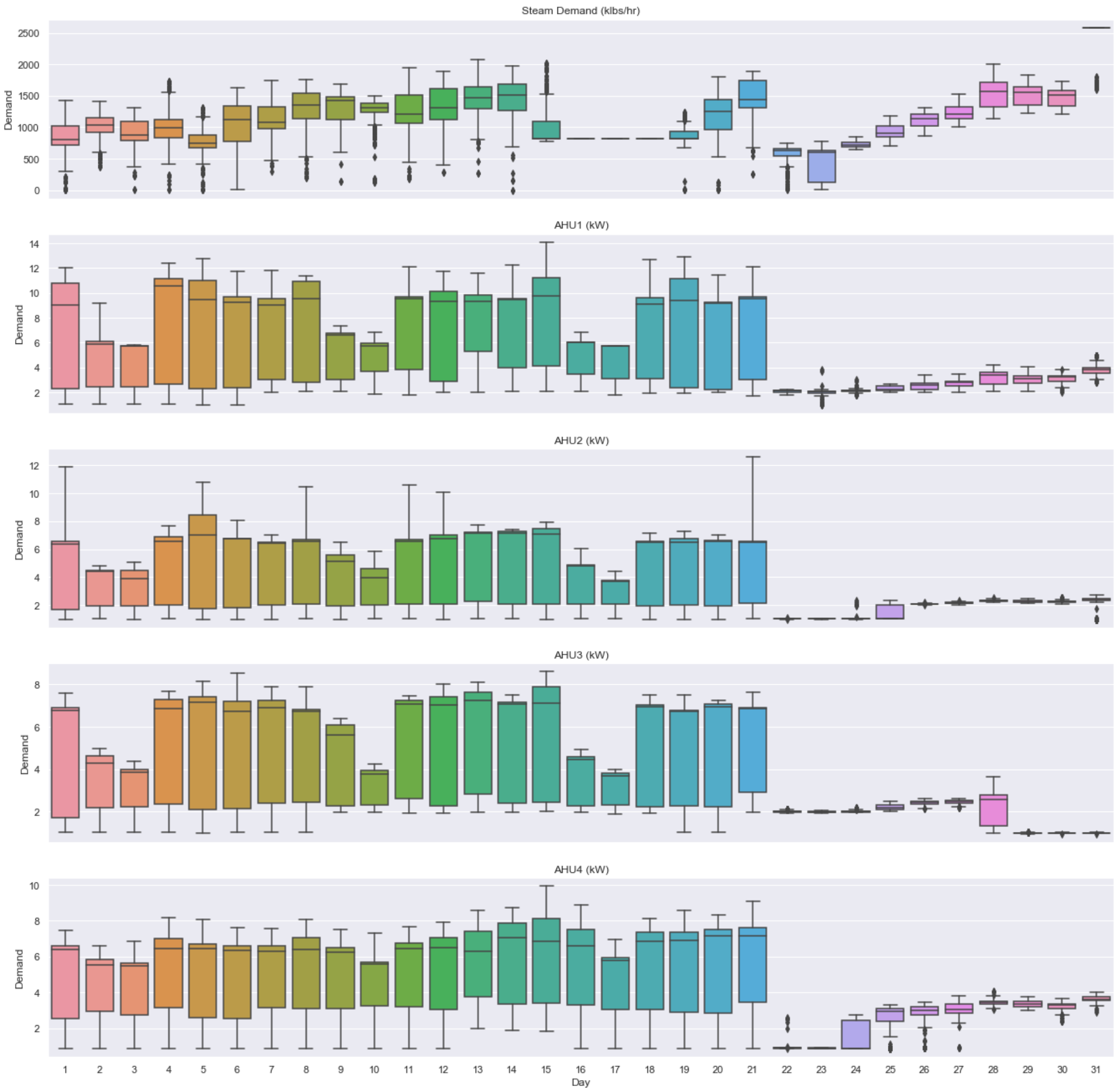
[39]:

```
fig, axes = plt.subplots(5, 1, figsize=(21, 21), sharex=True)

for name, ax in zip(['Steam Demand (klbs/hr)', 'AHU1 (kW)', 'AHU2 (kW)', 'AHU3␣
 ↪(kW)', 'AHU4 (kW)'], axes):
    sns.boxplot(data=DF, x='Month', y=name, ax=ax, order=[10,11,12,1])
    ax.set_ylabel('Demand')
    ax.set_title(name)
# Remove the automatic x-axis label from all but the bottom subplot
    if ax != axes[-1]:
        ax.set_xlabel('')
```



[113]:

```
fig, axes = plt.subplots(5, 1, figsize=(21, 21), sharex=True)

for name, ax in zip(['Steam Demand (klbs/hr)', 'AHU1 (kW)', 'AHU2 (kW)', 'AHU3␣
 ↪(kW)', 'AHU4 (kW)'], axes):
    sns.boxplot(data=DF[DF['Month']==12], x='Day', y=name, ax=ax)
    ax.set_ylabel('Demand')
    ax.set_title(name)
# Remove the automatic x-axis label from all but the bottom subplot
    if ax != axes[-1]:
        ax.set_xlabel('')
```



[119]:

```
data_columns = ['Steam Demand (klbs/hr)', 'AHU1 (kW)', 'AHU2 (kW)', 'AHU3␣
 ↪(kW)', 'AHU4 (kW)']
```

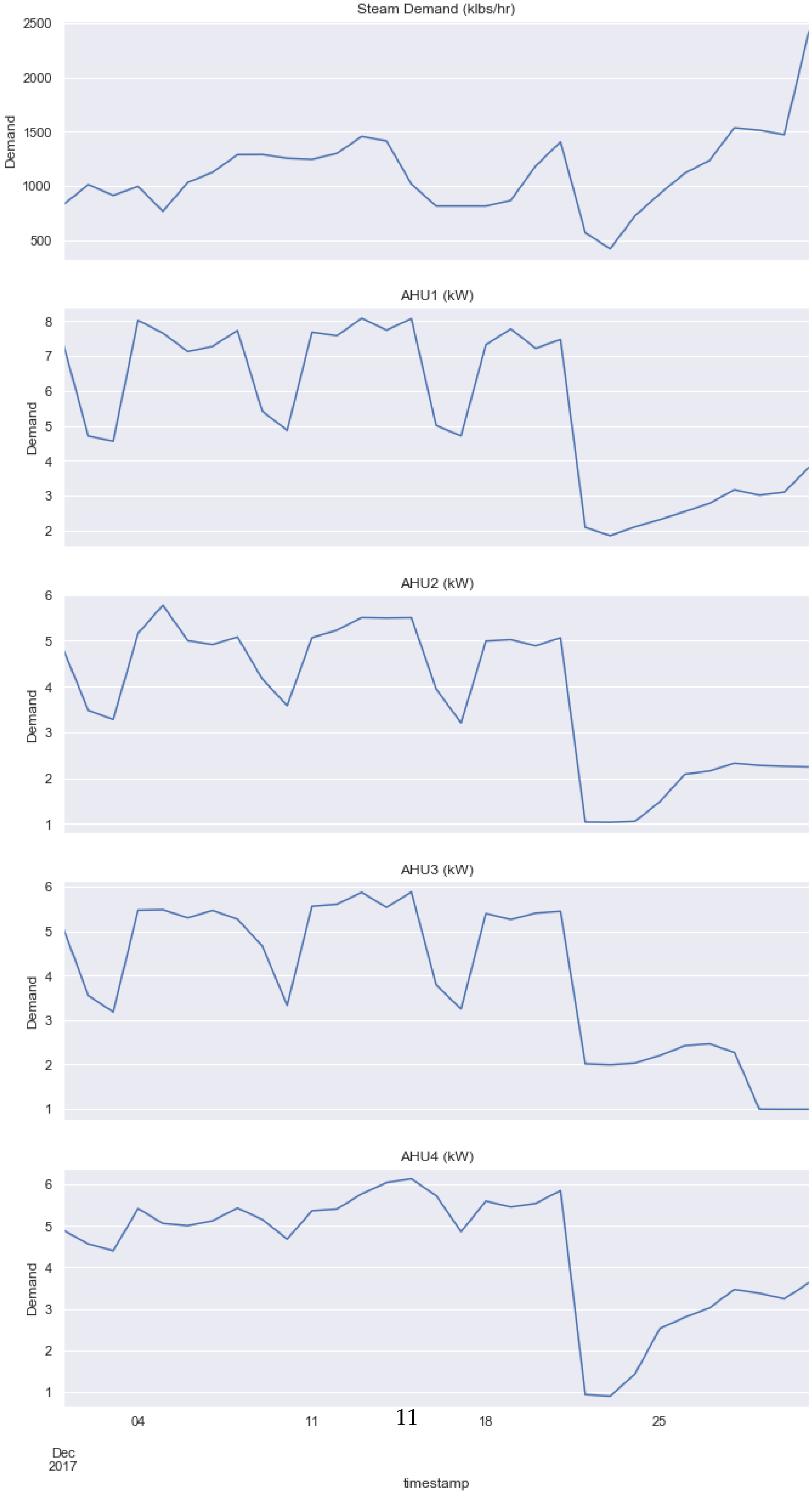
```
# Resample to Daily frequency, aggregating with mean
DF_daily = DF[data_columns].resample('D').mean()
DF_daily.head(3)
```

[119]:

```
            Steam Demand (klbs/hr)  AHU1 (kW)  AHU2 (kW)  AHU3 (kW)  AHU4 (kW)
timestamp
2017-10-01               31.147404   4.364887   9.192339   8.175065   6.570660
2017-10-02               27.138783   8.410135  10.661133   8.949805   6.716160
2017-10-03               43.165955   8.016840  10.690213   8.919336   6.677112
```

[122]:

```
fig, axes = plt.subplots(5, 1, figsize=(11, 21), sharex=True)

for name, ax in zip(['Steam Demand (klbs/hr)', 'AHU1 (kW)', 'AHU2 (kW)', 'AHU3␣
 ↪(kW)', 'AHU4 (kW)'], axes):
    DF_daily.loc['2017-12', name].plot(ax=ax)
    ax.set_ylabel('Demand')
    ax.set_title(name)
# Remove the automatic x-axis label from all but the bottom subplot
    if ax != axes[-1]:
        ax.set_xlabel('')
```


Steam Demand (klbs/hr)
Demand
2500
2000
1500
1000
500
AHU1 (kW)
Demand
8
7
6
5
4
3
2
AHU2 (kW)
Demand
6
5
4
3
2
1
AHU3 (kW)
Demand
6
5
4
3
2
1
AHU4 (kW)
Demand
6
5
4
3
2
1
04
11
18
25
Dec
2017
timestamp

[123]:

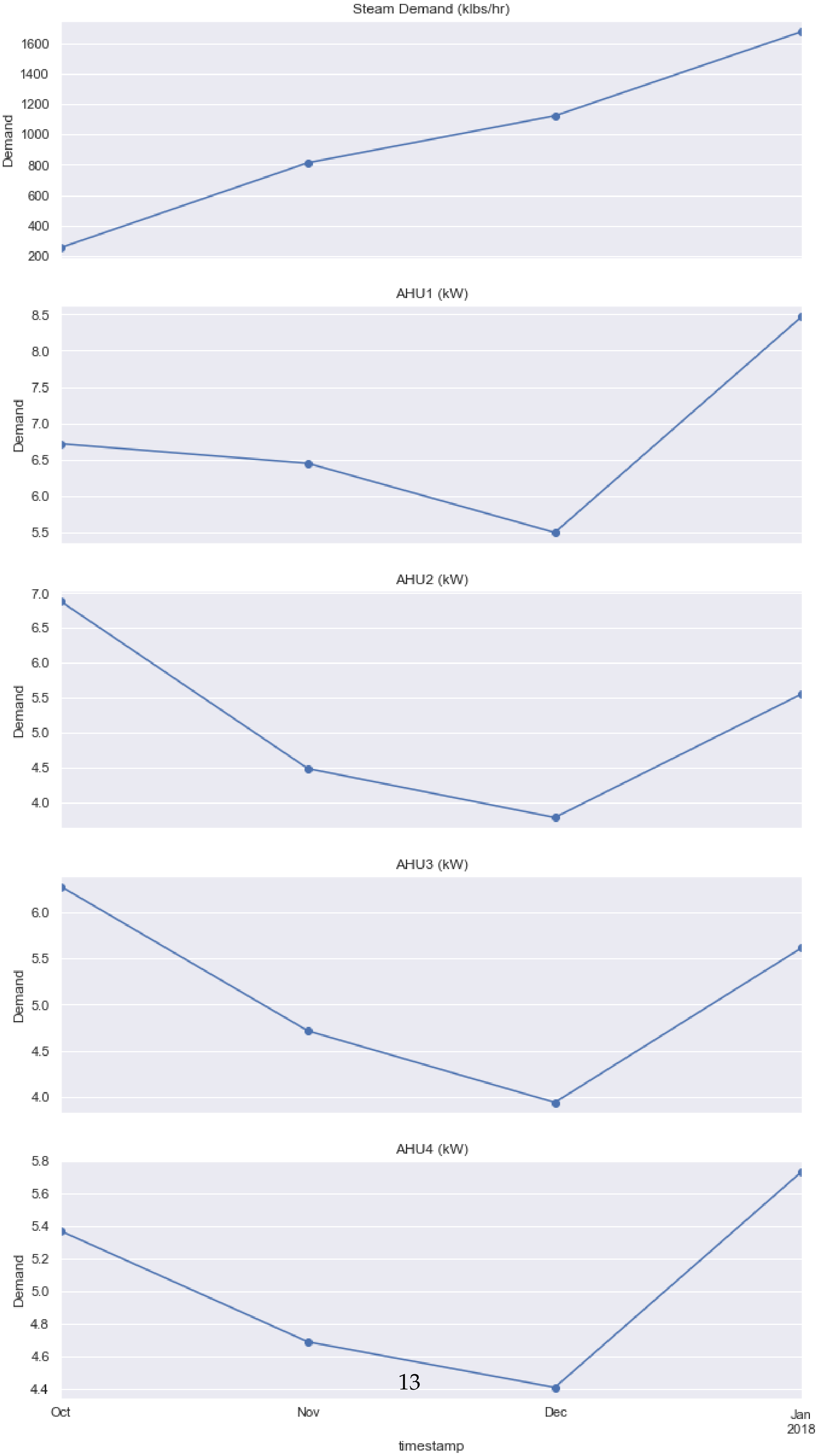
```
data_columns = ['Steam Demand (klbs/hr)', 'AHU1 (kW)', 'AHU2 (kW)', 'AHU3␣
 ↪(kW)', 'AHU4 (kW)']

# Resample to monthly frequency, aggregating with mean
DF_monthly = DF[data_columns].resample('M').mean()
DF_monthly.head(3)
```

[123]:

```
            Steam Demand (klbs/hr)  AHU1 (kW)  AHU2 (kW)  AHU3 (kW)  AHU4 (kW)
timestamp
2017-10-31              256.037630   6.715296   6.878556   6.281271   5.370450
2017-11-30              813.910385   6.444917   4.480529   4.715876   4.691602
2017-12-31             1122.608762   5.489457   3.778628   3.939259   4.412102
```

[141]:

```
fig, axes = plt.subplots(5, 1, figsize=(11, 21), sharex=True)

for name, ax in zip(['Steam Demand (klbs/hr)', 'AHU1 (kW)', 'AHU2 (kW)', 'AHU3␣
 ↪(kW)', 'AHU4 (kW)'], axes):
    DF_monthly.loc[:, name].plot(ax=ax, marker='o')
    ax.set_ylabel('Demand')
    ax.set_title(name)
# Remove the automatic x-axis label from all but the bottom subplot
    if ax != axes[-1]:
        ax.set_xlabel('')
```


Steam Demand (klbs/hr)
Demand
1600
1400
1200
1000
800
600
400
200
AHU1 (kW)
Demand
8.5
8.0
7.5
7.0
6.5
6.0
5.5
AHU2 (kW)
Demand
7.0
6.5
6.0
5.5
5.0
4.5
4.0
AHU3 (kW)
Demand
6.0
5.5
5.0
4.5
4.0
AHU4 (kW)
Demand
5.8
5.6
5.4
5.2
5.0
4.8
4.6
4.4
Oct
Nov
Dec
Jan
2018
timestamp

[ ]:

```
```