Super_Market_Data_Analysis

September 4, 2021

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

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

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

[3]:
```
              Unnamed: 0  Billing_Electric_Meter_Avg_Daily KW  \
328  25/11/2017 00:00:00                           173.229167
329  26/11/2017 00:00:00                           137.512500
330  27/11/2017 00:00:00                           173.987500
331  28/11/2017 00:00:00                           169.845833
332  29/11/2017 00:00:00                           170.795833

     Lighting_Meters_Avg (daily KW)  Air_Temperature_Daily_Avg  \
328                       24.998377                   2.416667
329                       13.781720                   4.114583
330                       24.969721                   7.218750
331                       24.922763                   3.833333
332                       24.937905                   3.159722

     HVAC_Meters_Avg_Daily_KW  Refrigeration_Electric_Avg_Daily KW
328                 17.923738                            -0.006355
329                 11.682986                            -0.006380
330                 18.282193                            -0.006469
331                 17.841279                            -0.006380
332                 17.633929                            -0.006437
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 333 entries, 0 to 332
Data columns (total 6 columns):
Unnamed: 0                             333 non-null object
Billing_Electric_Meter_Avg_Daily KW    333 non-null float64
Lighting_Meters_Avg (daily KW)         329 non-null float64
Air_Temperature_Daily_Avg              333 non-null float64
HVAC_Meters_Avg_Daily_KW               295 non-null float64
Refrigeration_Electric_Avg_Daily KW    291 non-null float64
```

```
dtypes: float64(5), object(1)
memory usage: 15.7+ KB
```

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

perc_missing(data)
```

```
Unnamed: 0 => 0.0%
Billing_Electric_Meter_Avg_Daily KW => 0.0%
Lighting_Meters_Avg (daily KW) => 1.2%
Air_Temperature_Daily_Avg => 0.0%
HVAC_Meters_Avg_Daily_KW => 11.41%
Refrigeration_Electric_Avg_Daily KW => 12.61%
```

[6]:
```python
data.describe()
```

[6]:
```
       Billing_Electric_Meter_Avg_Daily KW  Lighting_Meters_Avg (daily KW)  \
count                           333.000000                      329.000000   
mean                            204.037290                       31.091078   
std                              21.394318                        6.450581   
min                             137.512500                       13.287447   
25%                             188.808333                       25.021719   
50%                             206.495833                       33.474672   
75%                             216.958333                       33.892039   
max                             253.483333                       49.766792   

       Air_Temperature_Daily_Avg  HVAC_Meters_Avg_Daily_KW  \
count                 333.000000                295.000000   
mean                   10.934676                 19.867964   
std                     4.809461                  5.770472   
min                    -1.083333                  2.845333   
25%                     7.988542                 17.295369   
50%                    12.000000                 18.722812   
75%                    14.458333                 20.657079   
max                    24.036458                 38.605892   

       Refrigeration_Electric_Avg_Daily KW  
count                           291.000000  
mean                             41.253033  
std                              25.513854  
min                              -0.011786  
25%                              30.591439  
50%                              49.241892  
75%                              56.659410  
```

```
max                             107.081502
```

[7]:

```
data['Date'] = pd.to_datetime(data['Unnamed: 0'], dayfirst=True)
data.head()
```

[7]:

```
      Unnamed: 0  Billing_Electric_Meter_Avg_Daily KW  \
0  1/1/2017 0:00                           167.687500
1  2/1/2017 0:00                           189.033333
2  3/1/2017 0:00                           216.670833
3  4/1/2017 0:00                           238.391667
4  5/1/2017 0:00                           239.583333

   Lighting_Meters_Avg (daily KW)  Air_Temperature_Daily_Avg  \
0                             NaN                   4.960417
1                             NaN                   1.034375
2                             NaN                   0.797917
3                       49.766792                   4.000000
4                       38.924178                  -0.466667

   HVAC_Meters_Avg_Daily_KW  Refrigeration_Electric_Avg_Daily KW       Date
0                       NaN                                  NaN 2017-01-01
1                       NaN                                  NaN 2017-01-02
2                       NaN                                  NaN 2017-01-03
3                 22.968541                            50.672500 2017-01-04
4                 38.605892                            46.008329 2017-01-05
```

[8]:

```
data.drop('Unnamed: 0', axis=1, inplace=True)
data = data.set_index('Date')
data.head()
```

[8]:

```
            Billing_Electric_Meter_Avg_Daily KW  \
Date
2017-01-01                           167.687500
2017-01-02                           189.033333
2017-01-03                           216.670833
2017-01-04                           238.391667
2017-01-05                           239.583333

            Lighting_Meters_Avg (daily KW)  Air_Temperature_Daily_Avg  \
Date
2017-01-01                             NaN                   4.960417
2017-01-02                             NaN                   1.034375
2017-01-03                             NaN                   0.797917
2017-01-04                       49.766792                   4.000000
2017-01-05                       38.924178                  -0.466667

            HVAC_Meters_Avg_Daily_KW  Refrigeration_Electric_Avg_Daily KW
Date
2017-01-01                       NaN                                  NaN
```

```
2017-01-02                    NaN                                   NaN
2017-01-03                    NaN                                   NaN
2017-01-04              22.968541                             50.672500
2017-01-05              38.605892                             46.008329
```

[9]:

```
# imputing with mean
Lighting_Meters_Avg_mean_value = round(data['Lighting_Meters_Avg (daily KW)'].
 ↪mean(), 2)
HVAC_Meters_Avg_Daily_KW_mean_value = round(data['HVAC_Meters_Avg_Daily_KW'].
 ↪mean(), 2)
Refrigeration_Electric_Avg_Daily_KW_mean_value =␣
 ↪round(data['Refrigeration_Electric_Avg_Daily KW'].mean(), 2)

data['Lighting_Meters_Avg (daily KW)'].fillna(Lighting_Meters_Avg_mean_value,␣
 ↪inplace=True)
data['HVAC_Meters_Avg_Daily_KW'].fillna(HVAC_Meters_Avg_Daily_KW_mean_value,␣
 ↪inplace=True)
data['Refrigeration_Electric_Avg_Daily KW'].
 ↪fillna(Refrigeration_Electric_Avg_Daily_KW_mean_value, inplace=True)
```

[10]:

```
data.isnull().sum()
```

[10]:

```
Billing_Electric_Meter_Avg_Daily KW    0
Lighting_Meters_Avg (daily KW)         0
Air_Temperature_Daily_Avg              0
HVAC_Meters_Avg_Daily_KW               0
Refrigeration_Electric_Avg_Daily KW    0
dtype: int64
```

[11]:

```
import matplotlib.pyplot as plt
import seaborn as sns
```

[12]:

```
data['Month'] =  data.index.month
```

[13]:

```
data['Day'] =  data.index.day
```

[14]:

```
DF = data
DF.head()
```

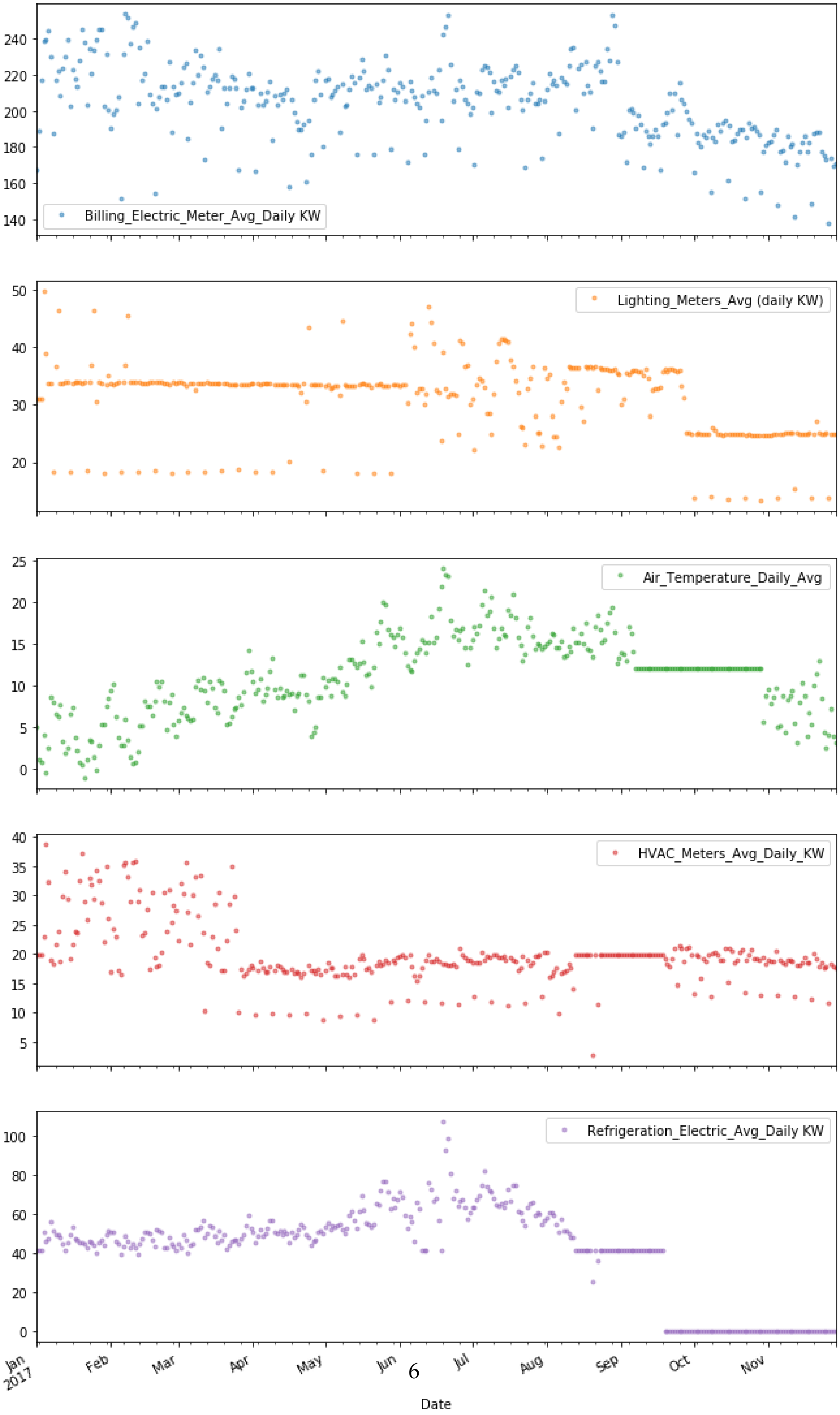
[14]:

```
            Billing_Electric_Meter_Avg_Daily KW  \
Date
2017-01-01                           167.687500
2017-01-02                           189.033333
2017-01-03                           216.670833
2017-01-04                           238.391667
2017-01-05                           239.583333

            Lighting_Meters_Avg (daily KW)  Air_Temperature_Daily_Avg  \
Date
2017-01-01                       31.090000                   4.960417
```

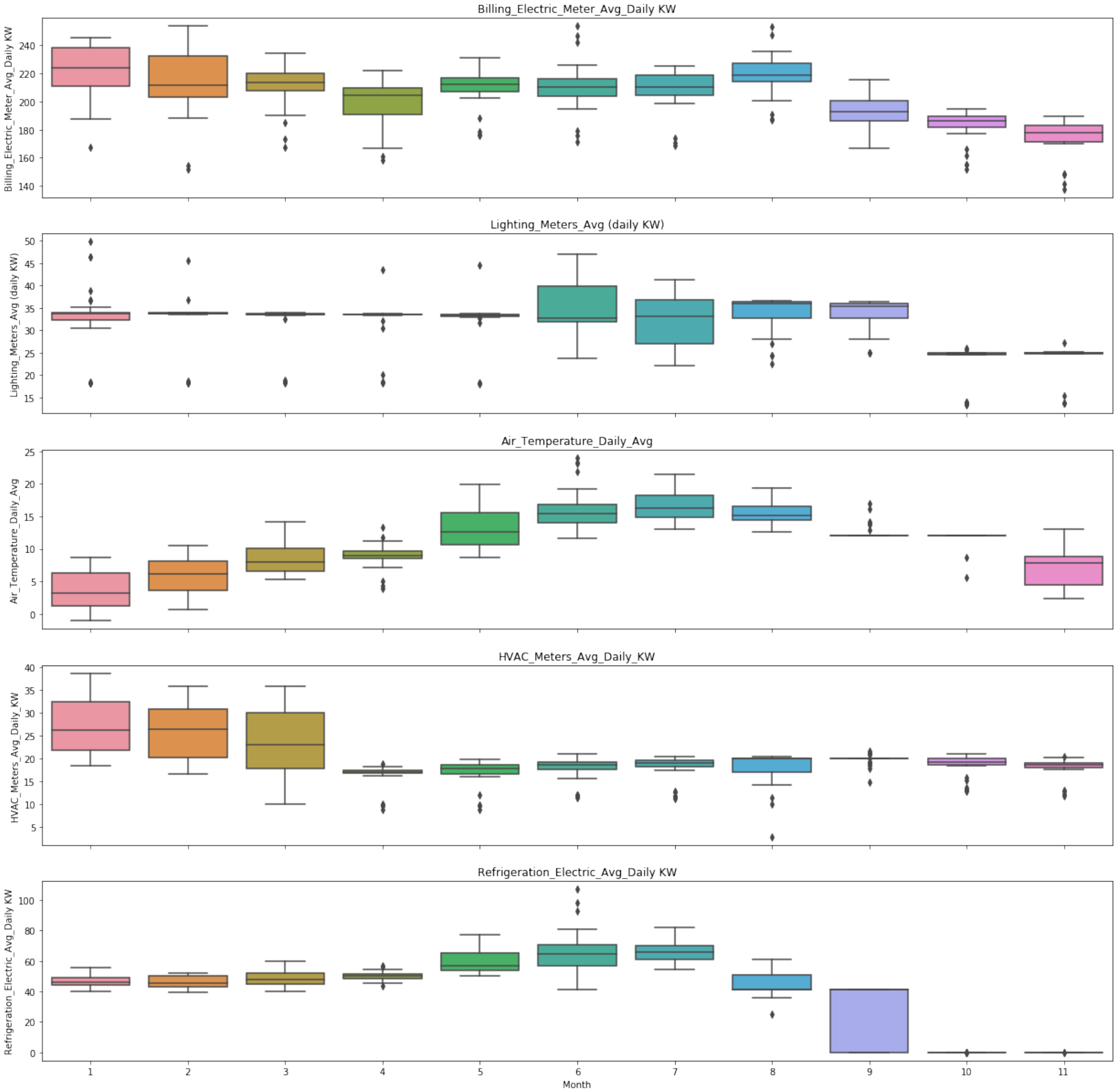
```
2017-01-02                      31.090000                   1.034375
2017-01-03                      31.090000                   0.797917
2017-01-04                      49.766792                   4.000000
2017-01-05                      38.924178                  -0.466667

            HVAC_Meters_Avg_Daily_KW  Refrigeration_Electric_Avg_Daily KW  \
Date
2017-01-01                 19.870000                            41.250000
2017-01-02                 19.870000                            41.250000
2017-01-03                 19.870000                            41.250000
2017-01-04                 22.968541                            50.672500
2017-01-05                 38.605892                            46.008329

            Month  Day
Date
2017-01-01      1    1
2017-01-02      1    2
2017-01-03      1    3
2017-01-04      1    4
2017-01-05      1    5
```

[15]:

```python
cols_plot = ['Billing_Electric_Meter_Avg_Daily KW', 'Lighting_Meters_Avg (daily␣
 ↪KW)', 'Air_Temperature_Daily_Avg',
             'HVAC_Meters_Avg_Daily_KW', 'Refrigeration_Electric_Avg_Daily KW']

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

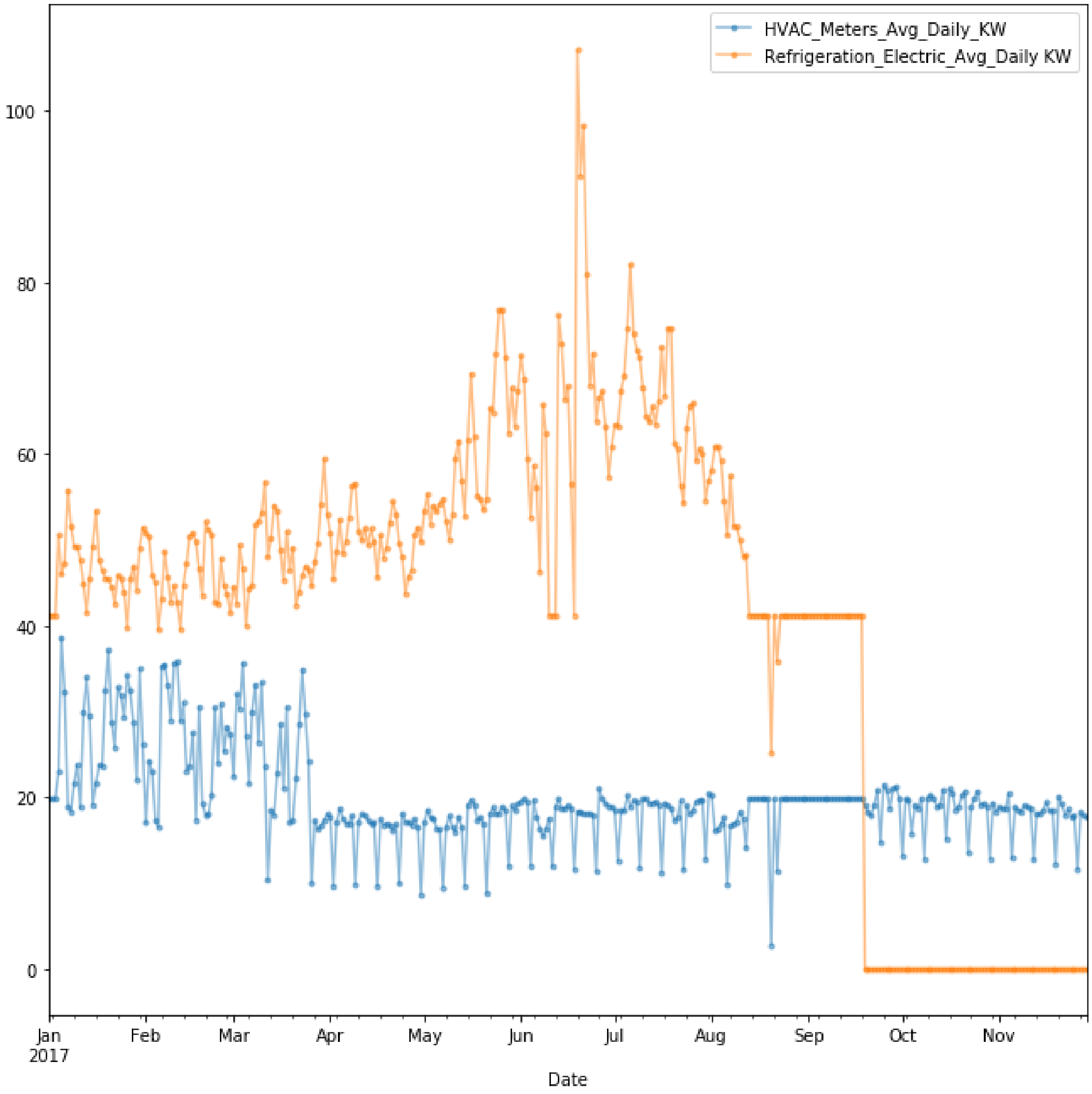

Billing_Electric_Meter_Avg_Daily KW
240
220
200
180
160
140
Lighting_Meters_Avg (daily KW)
50
40
30
20
Air_Temperature_Daily_Avg
25
20
15
10
5
0
HVAC_Meters_Avg_Daily_KW
40
35
30
25
20
15
10
5
Refrigeration_Electric_Avg_Daily KW
100
80
60
40
20
0
Jan
2017
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Date

```
[16]: cols = ['Billing_Electric_Meter_Avg_Daily KW', 'Lighting_Meters_Avg (daily␣
       ↪KW)', 'Air_Temperature_Daily_Avg',
                    'HVAC_Meters_Avg_Daily_KW', 'Refrigeration_Electric_Avg_Daily KW']

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

      for name, ax in zip(cols, axes):
          sns.boxplot(data=DF, x='Month', y=name, ax=ax)
          ax.set_title(name)
      # Remove the automatic x-axis label from all but the bottom subplot
          if ax != axes[-1]:
              ax.set_xlabel('')
```



[17]:
```
DF[['HVAC_Meters_Avg_Daily_KW', 'Refrigeration_Electric_Avg_Daily KW']].
 ↪plot(marker='.', alpha=0.5, linestyle='-', figsize=(11, 11))
```

[17]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x188edf84240>
```



[ ]: