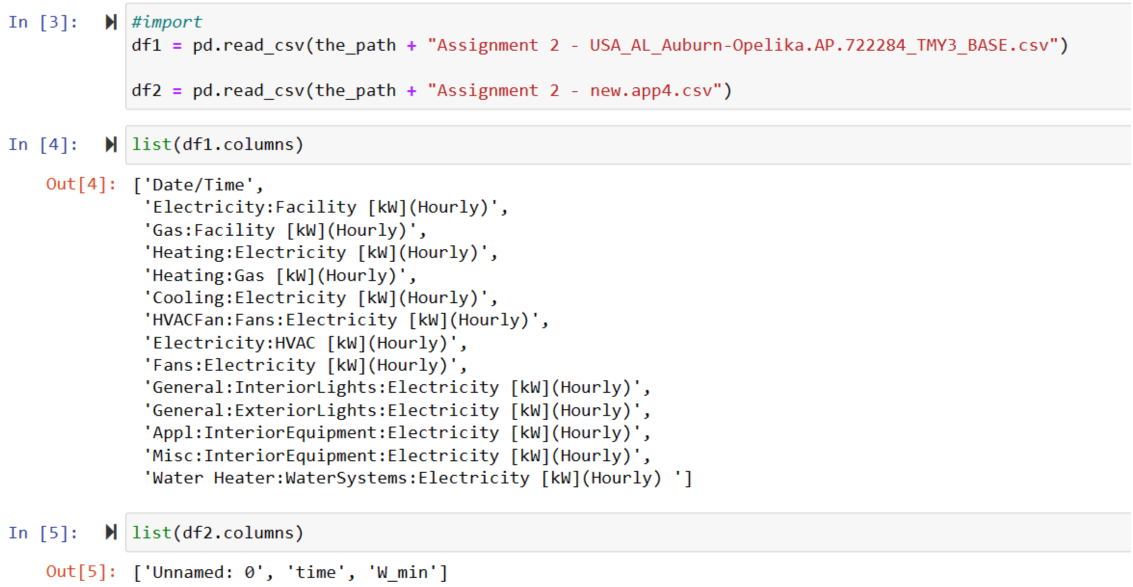
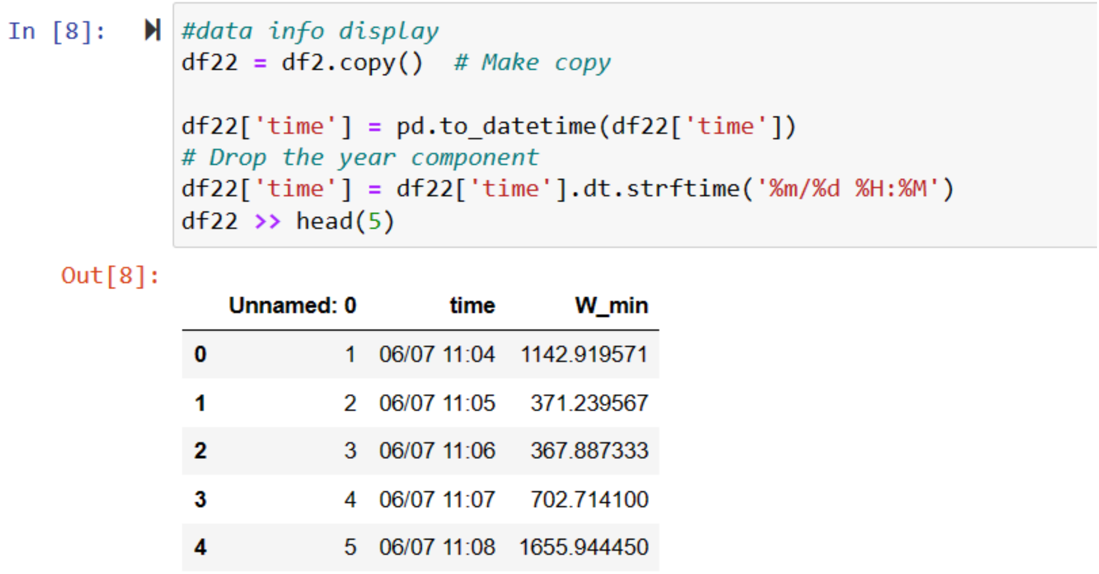
Shuxiang Sui

Check data:



```
In [3]:  #import
         df1 = pd.read_csv(the_path + "Assignment 2 - USA_AL_Auburn-Opelika.AP.722284_TMY3_BASE.csv")

         df2 = pd.read_csv(the_path + "Assignment 2 - new.app4.csv")

In [4]:  list(df1.columns)

Out[4]:  ['Date/Time',
          'Electricity:Facility [kW](Hourly)',
          'Gas:Facility [kW](Hourly)',
          'Heating:Electricity [kW](Hourly)',
          'Heating:Gas [kW](Hourly)',
          'Cooling:Electricity [kW](Hourly)',
          'HVACFan:Fans:Electricity [kW](Hourly)',
          'Electricity:HVAC [kW](Hourly)',
          'Fans:Electricity [kW](Hourly)',
          'General:InteriorLights:Electricity [kW](Hourly)',
          'General:ExteriorLights:Electricity [kW](Hourly)',
          'Appl:InteriorEquipment:Electricity [kW](Hourly)',
          'Misc:InteriorEquipment:Electricity [kW](Hourly)',
          'Water Heater:WaterSystems:Electricity [kW](Hourly) ']

In [5]:  list(df2.columns)

Out[5]:  ['Unnamed: 0', 'time', 'W_min']
```

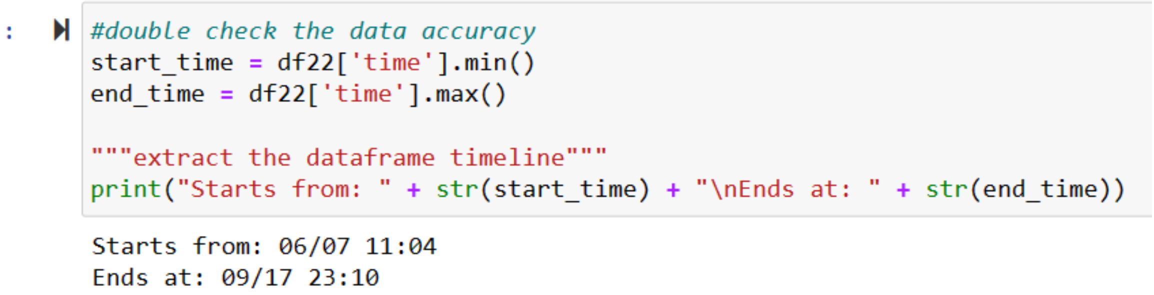
Transform 2 data source into same format:



```
In [8]:  #data info display
         df22 = df2.copy()   # Make copy

         df22['time'] = pd.to_datetime(df22['time'])
         # Drop the year component
         df22['time'] = df22['time'].dt.strftime('%m/%d %H:%M')
         df22 >> head(5)

Out[8]:
```

	Unnamed: 0	time	W_min
0	1	06/07 11:04	1142.919571
1	2	06/07 11:05	371.239567
2	3	06/07 11:06	367.887333
3	4	06/07 11:07	702.714100
4	5	06/07 11:08	1655.944450

Double check the data range:

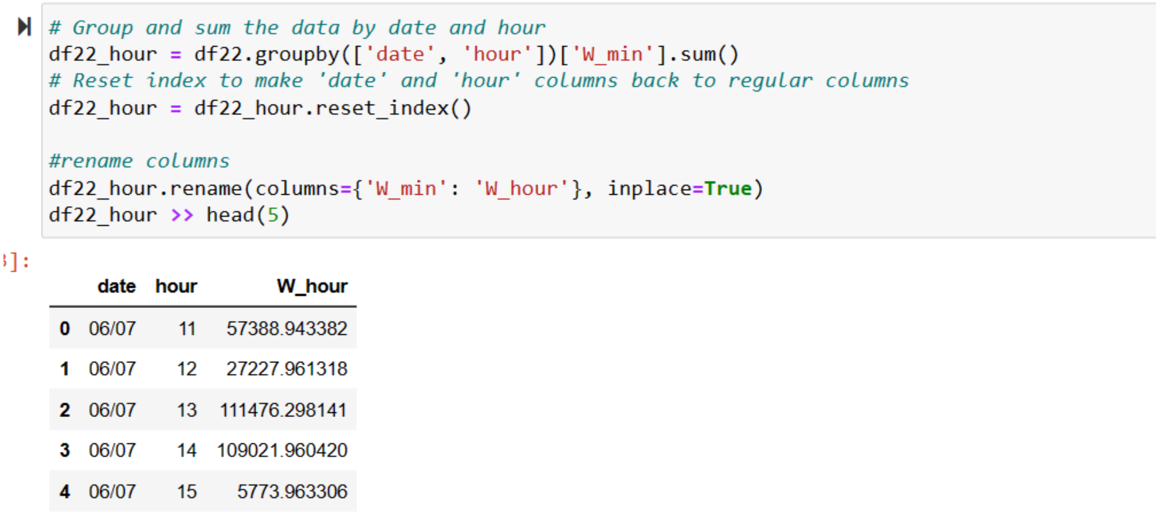


```
#double check the data accuracy
start_time = df22['time'].min()
end_time = df22['time'].max()

"""extract the dataframe timeline"""
print("Starts from: " + str(start_time) + "\nEnds at: " + str(end_time))
```

```
Starts from: 06/07 11:04
Ends at: 09/17 23:10
```

Due to the formatting issues, some data needs to be recoded and transformed into DATETIME, such as 24:00 to 00:00:

And I summarize the minute usage into hourly usage in W/hour:

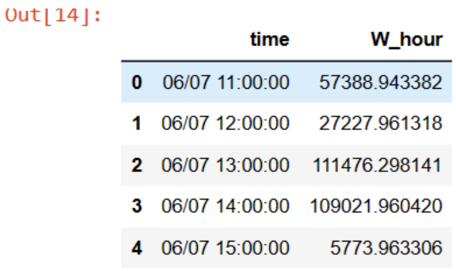


```
# Group and sum the data by date and hour
df22_hour = df22.groupby(['date', 'hour'])['W_min'].sum()
# Reset index to make 'date' and 'hour' columns back to regular columns
df22_hour = df22_hour.reset_index()

#rename columns
df22_hour.rename(columns={'W_min': 'W_hour'}, inplace=True)
df22_hour >> head(5)
```

]:

	date	hour	W_hour
0	06/07	11	57388.943382
1	06/07	12	27227.961318
2	06/07	13	111476.298141
3	06/07	14	109021.960420
4	06/07	15	5773.963306

While checking the data before merging, I found out that, there are about 65 hours of data are still missing, and this might result in unreasonable trend variation.

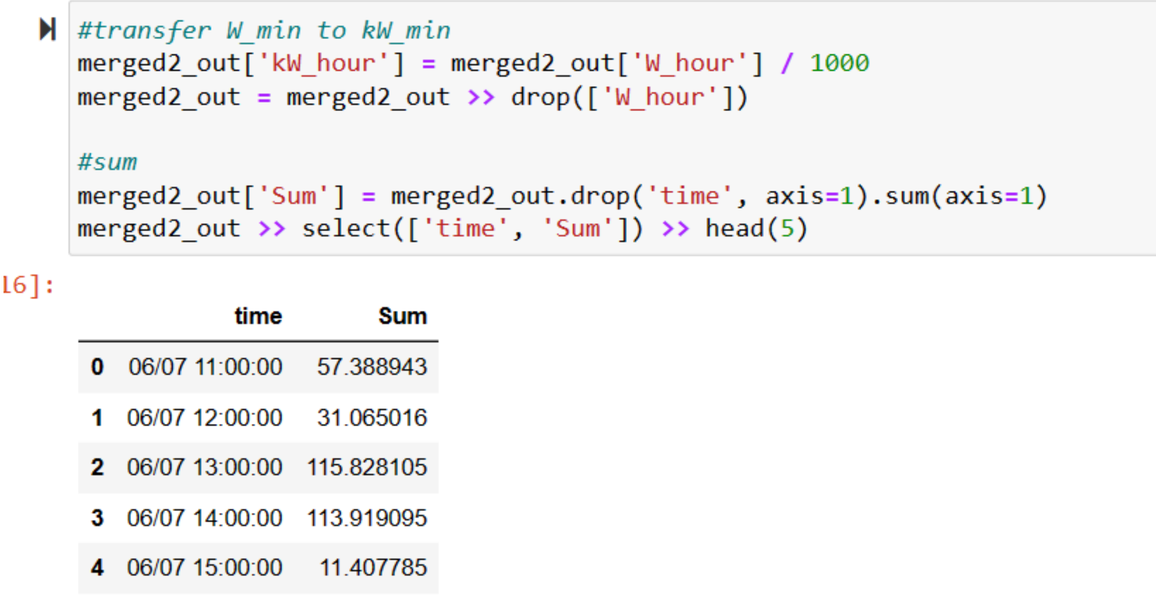


Out[14]:

	time	W_hour
0	06/07 11:00:00	57388.943382
1	06/07 12:00:00	27227.961318
2	06/07 13:00:00	111476.298141
3	06/07 14:00:00	109021.960420
4	06/07 15:00:00	5773.963306

double check with calculation, 10846 data / 60 = 180hours.
however dataset includes 245 hours, therefore not all of minute usages are recorded.

After checking the data missing, I merged the data and transfer the Watt into kilowatts and form the new df:



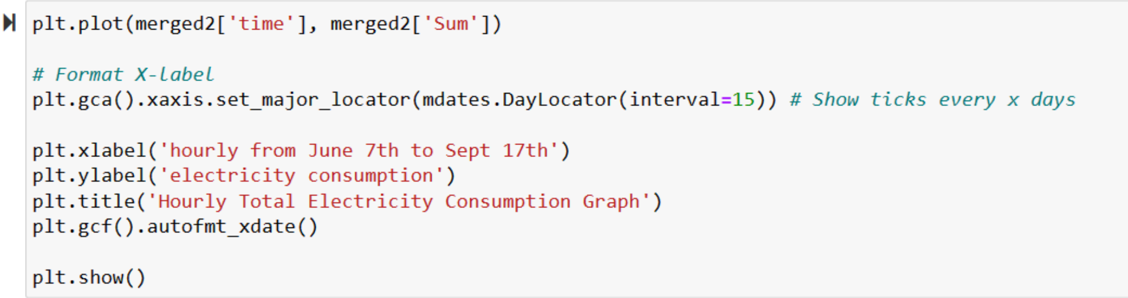
```
#transfer W_min to kW_min
merged2_out['kW_hour'] = merged2_out['W_hour'] / 1000
merged2_out = merged2_out >> drop(['W_hour'])

#sum
merged2_out['Sum'] = merged2_out.drop('time', axis=1).sum(axis=1)
merged2_out >> select(['time', 'Sum']) >> head(5)
```

16]:

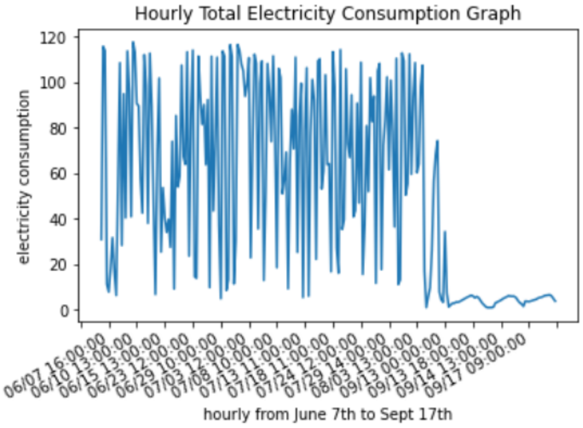
	time	Sum
0	06/07 11:00:00	57.388943
1	06/07 12:00:00	31.065016
2	06/07 13:00:00	115.828105
3	06/07 14:00:00	113.919095
4	06/07 15:00:00	11.407785

There are 2 merge methods, and I decided to keep it with inner merging since there will be too much missing data for many electricity consumption sources in the first csv file.

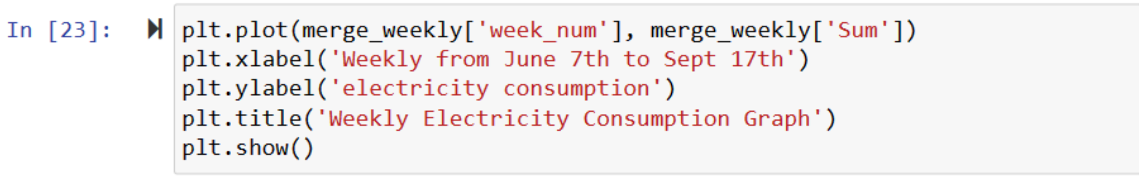
By merging, I am able to produce the Hourly Total Electricity Consumption graph:



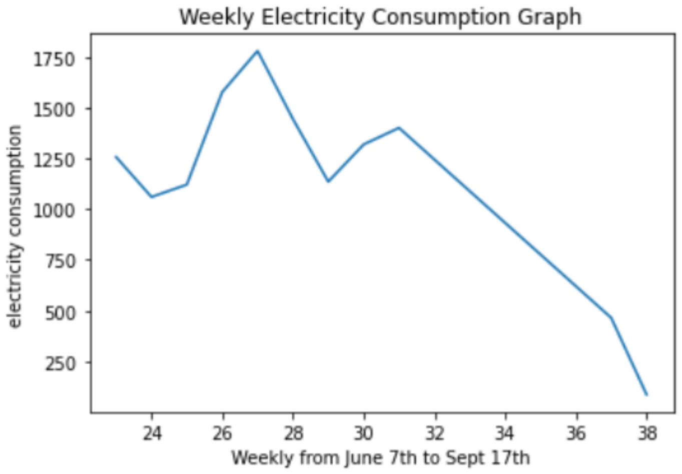
```
plt.plot(merged2['time'], merged2['Sum'])

# Format X-label
plt.gca().xaxis.set_major_locator(mdates.DayLocator(interval=15)) # Show ticks every x days

plt.xlabel('hourly from June 7th to Sept 17th')
plt.ylabel('electricity consumption')
plt.title('Hourly Total Electricity Consumption Graph')
plt.gcf().autofmt_xdate()

plt.show()
```



By merging the data of hourly consumption again, I got the daily data, and then I collected the weekly data and generated the Weekly Total Electricity Consumption Graph:

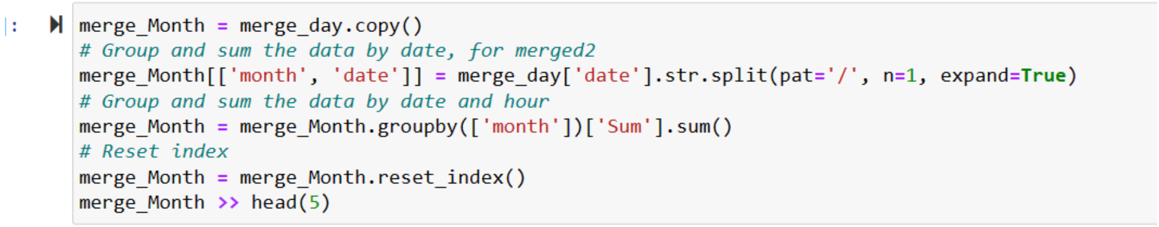


```
In [23]:  plt.plot(merge_weekly['week_num'], merge_weekly['Sum'])
           plt.xlabel('Weekly from June 7th to Sept 17th')
           plt.ylabel('electricity consumption')
           plt.title('Weekly Electricity Consumption Graph')
           plt.show()
```

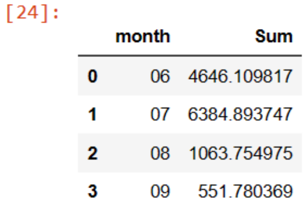


June 7th to Sept 17th, which is from 24th week to 38th week.

Then, I put the daily data into the month, collected the total consumption for each month from June to September:

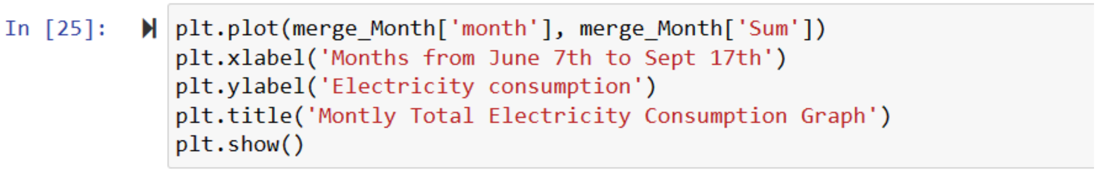


```
]:   merge_Month = merge_day.copy()
     # Group and sum the data by date, for merged2
     merge_Month[['month', 'date']] = merge_day['date'].str.split(pat='/', n=1, expand=True)
     # Group and sum the data by date and hour
     merge_Month = merge_Month.groupby(['month'])['Sum'].sum()
     # Reset index
     merge_Month = merge_Month.reset_index()
     merge_Month >> head(5)
```

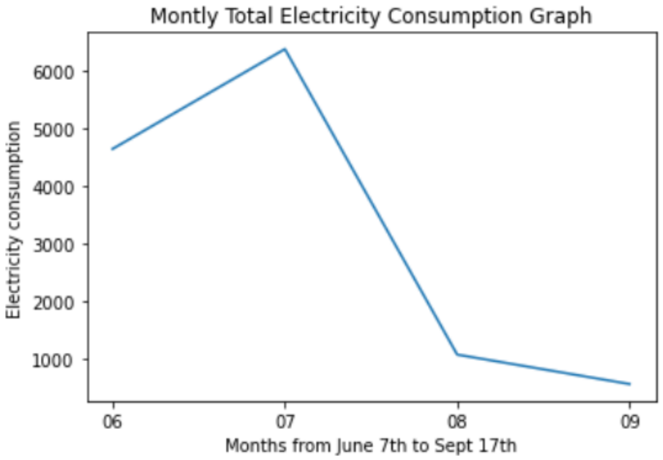


[24]:

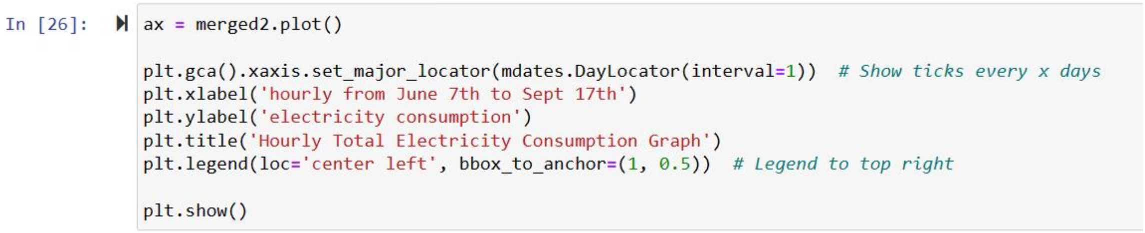
	month	Sum
0	06	4646.109817
1	07	6384.893747
2	08	1063.754975
3	09	551.780369



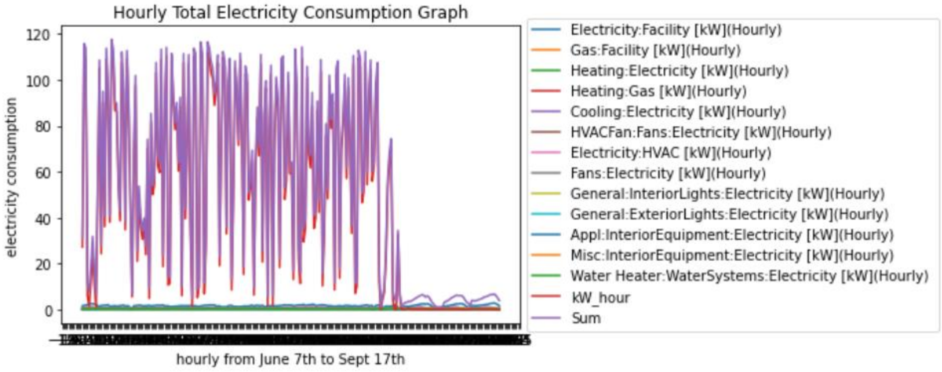
```
In [25]:  plt.plot(merge_Month['month'], merge_Month['Sum'])
          plt.xlabel('Months from June 7th to Sept 17th')
          plt.ylabel('Electricity consumption')
          plt.title('Montly Total Electricity Consumption Graph')
          plt.show()
```



The consumption decreased very strangely, and I decided to further develop the reason behind each electricity consumptions:



```
In [26]:  ax = merged2.plot()

          plt.gca().xaxis.set_major_locator(mdates.DayLocator(interval=1))  # Show ticks every x days
          plt.xlabel('hourly from June 7th to Sept 17th')
          plt.ylabel('electricity consumption')
          plt.title('Hourly Total Electricity Consumption Graph')
          plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))  # Legend to top right

          plt.show()
```



As we could observe from the map, the total consumption is highly correlated to the New Added appliance's electricity consumption, and it takes more than 80% of house consumption. And this might be the reason why the total consumption suddenly decrease in the last few days.