Summary Report for Assignment 2

Henghui Qi

Firstly, load both data files and check the datetime form of each file. The “USA_AL_Auburn-Opelika.AP.722284_TMY3_BASE” file, read as df1, has the datetime with form of ' 01/01 01:00:00' and the “new.app4” file, read as df2, has the datetime with form of '6/7/2013 11:04'.

```
df1 = pd.read_csv('C:/Users/QI/Desktop/2022summer/Intern/Guzman Energy/May2022_homework-main/data/Assignment 2 - USA_AL_Auburn-Opelika.AP.7222
df2 = pd.read_csv('C:/Users/QI/Desktop/2022summer/Intern/Guzman Energy/May2022_homework-main/data/Assignment 2 - new.app4.csv')
```

```
df1['Date/Time'].unique()
```

```
array([' 01/01  01:00:00', ' 01/01  02:00:00', ' 01/01  03:00:00', ...,
       ' 12/31  22:00:00', ' 12/31  23:00:00', ' 12/31  24:00:00'],
      dtype=object)
```

```
df2['time'].unique()
```

```
array(['6/7/2013 11:04', '6/7/2013 11:05', '6/7/2013 11:06', ...,
       '9/17/2013 23:08', '9/17/2013 23:09', '9/17/2013 23:10'],
      dtype=object)
```

Make a copy of source data (df1 to df11 and df2 to df22) to process further without changing the original data frames.

```
df11 = df1.copy()
df22 = df2.copy()
```

In order to find the limitation of df1, format ‘Date/Time’ of df11 to “datetime” form (replaced “24:00:00” to “00:00:00” before) and add “Month”, “Day”, and “Hour” columns. The limitation of df1 is January 1st at 1am and December 31st at 12am.

```
df11['Time'] = df11['Date/Time'].str.replace('24:00:00', '00:00:00')
df11['Time'] = pd.to_datetime(df11['Time'], format = ' %m/%d  %H:%M:%S')
df11['Month'] = df11['Time'].dt.month
df11['Day'] = df11['Time'].dt.day
df11['Hour'] = df11['Time'].dt.hour
print(df11[['Time', 'Month', 'Day', 'Hour']].iloc[[0, df11.shape[0]-1], :])#The limitation of df1
```

```
                 Time  Month  Day  Hour
0    1900-01-01 01:00:00      1    1     1
8759 1900-12-31 00:00:00     12   31     0
```

Also, format ‘Time’ of df22 to “datetime” form and add “Month”, “Day”, and “Hour” columns. The limitation of df2 is 2013 June 7th at 11:04:00 and 2013 September 17th at 23:10:00. Therefore, the overlap period is between June 7th at 11am to September 17th at 11pm.

```
df22['Time'] = pd.to_datetime(df22['time'])
df22['Month'] = df22['Time'].dt.month
df22['Day'] = df22['Time'].dt.day
df22['Hour'] = df22['Time'].dt.hour
print(df22[['Time', 'Month', 'Day', 'Hour']].iloc[[0, df22.shape[0]-1], :])#The limitation of df2
Year = df22['Time'].dt.year[0]
```

```
                  Time  Month  Day  Hour
0     2013-06-07 11:04:00      6    7    11
10845 2013-09-17 23:10:00      9   17    23
```

Merge df22 into hourly data and show the first ten rows of the new data frame.

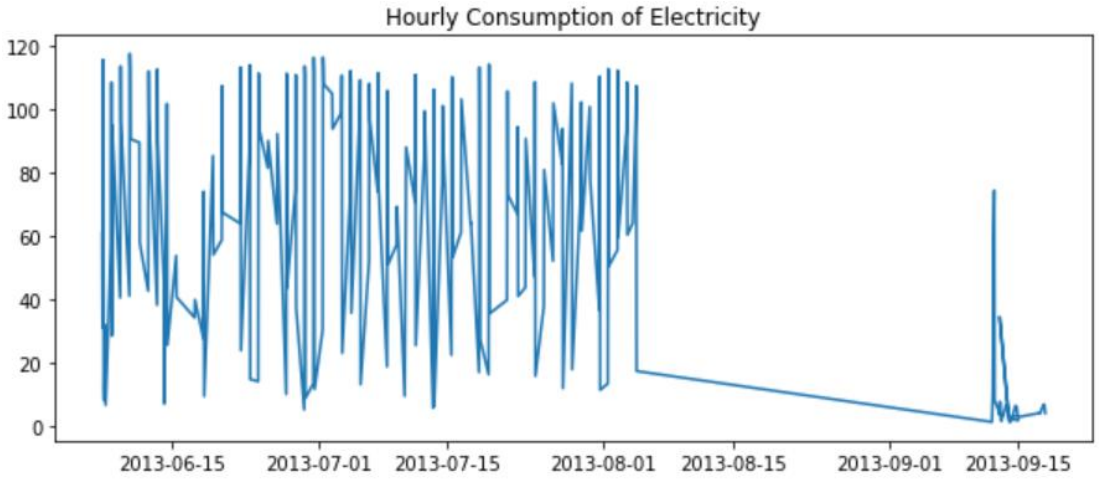
```
df2_sum = (df22 >>
           select(X.W_min, X.Month, X.Day, X.Hour) >>
           group_by(X.Month, X.Day, X.Hour)>>
           summarize(electricity = X.W_min.sum()/1000))
print(df2_sum)
```

```
    Hour  Day  Month  electricity
0     11    7      6    57.388943
1     12    7      6    27.227961
2     13    7      6   111.476298
3     14    7      6   109.021960
4     15    7      6     5.773963
5     16    7      6     1.619193
6     17    7      6    12.081027
7     18    7      6    25.478736
8     19    7      6    11.360621
9     20    7      6     1.054541
10    10    8      6    54.240433
```

Then, join two hourly data files on “Month”, “Day”, and “Hour”. Drop redundant column “Date/Time” (column “Time” is also included). Calculate the total hourly consumption of electricity by summing up columns of different kind of consumption.

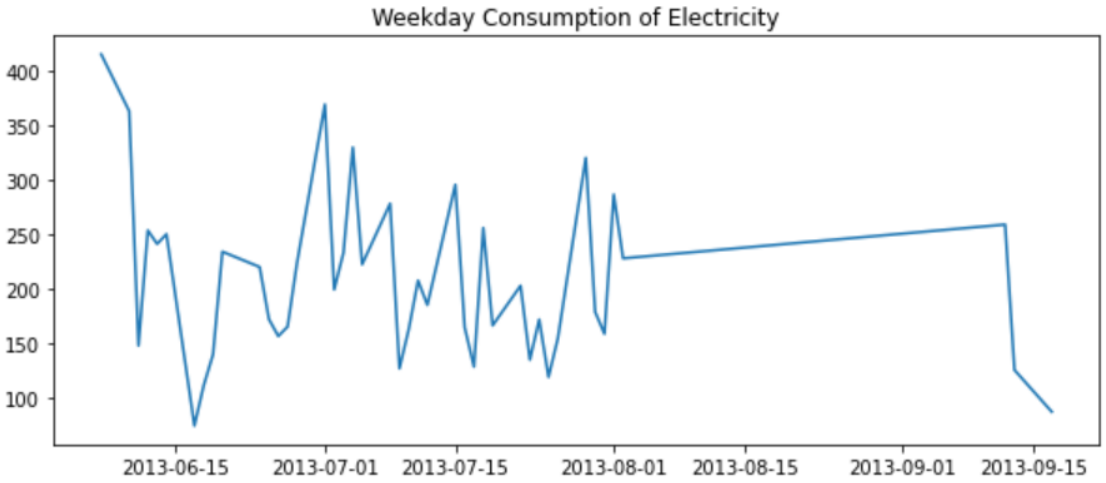
```
df = pd.merge(df11, df2_sum, on = ['Month', 'Day', 'Hour'])
df.drop(['Date/Time'],axis = 1, inplace = True)
df['total'] = np.sum(df.drop(['Time', 'Month', 'Day', 'Hour'], axis = 1), axis = 1)
for i in range(df.shape[0]):
    df.loc[i,'Time'] = df.loc[i,'Time'].replace(year = Year)
```

Plot hourly consumption of electricity. We can tell from the plot below that the total consumption of electricity from the beginning of August to the middle of September is abnormally low and steady. There could be problems collecting the data or some influential accidents happened at that time.



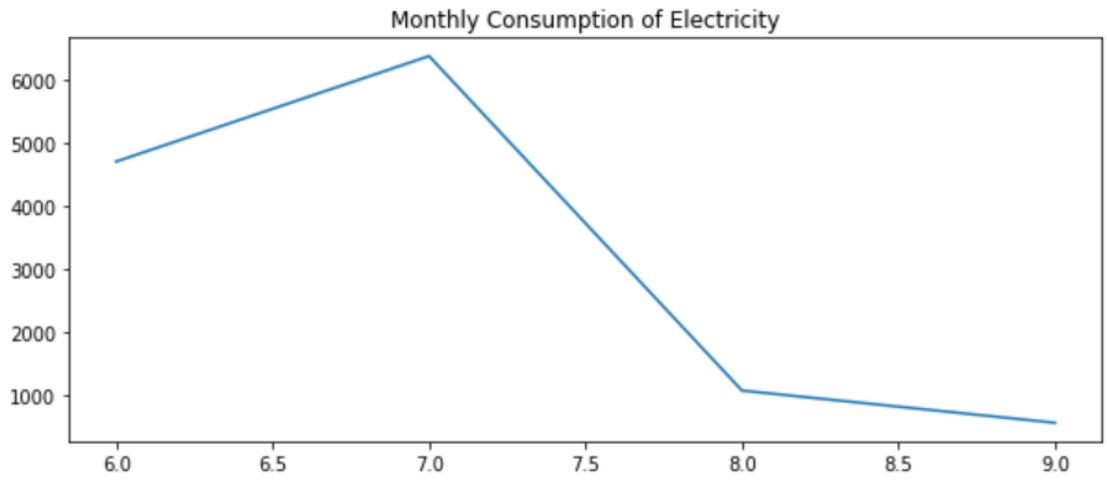
To plot daily (weekdays) consumption of electricity, get weekday of each day in the overlap period and filter them by weekday. Sum the total consumption of hours in one day and plot it. We can tell from the plot that besides the abnormal situation mentioned above, an outlier around 2013 June 16th is detected. The plot also indicates that a peak is predicable when it comes to the end of a month.

```
df_weekday = (df>>
              select(X.Time, X.Month, X.Day, X.Hour, X.total))
df_weekday['weekday'] = df_weekday['Time'].dt.weekday
df_weekday = (df_weekday>>
              filter_by(X.weekday < 5)>>
              group_by(X.Month, X.Day)>>
              summarize(weekday_total = X.total.sum()))
df_weekday['Date'] = 0
for i in range(df_weekday.shape[0]):
    df_weekday.loc[i, 'Date'] = datetime.datetime(Year, df_weekday.loc[i, 'Month'], df_weekday.loc[i, 'Day'])
fig, ax = plt.subplots(1, 1, figsize = (10, 4))
ax.plot(df_weekday['Date'], df_weekday['weekday_total'])
ax.set_title('Weekday Consumption of Electricity')
```



Sum the total consumption of days in each month and plot monthly consumption of electricity. July has the largest consumption according to the plot, which is in consistent with the law of nature. However, that June is not included wholely in the calculation period, and the data of August may be inaccurate, makes the explanation of the plot less reasonable.

```
df_month = (df >>
            select(X.Month, X.Day, X.Hour, X.total) >>
            group_by(X.Month)>>
            summarize(month_total = X.total.sum()))
fig, ax = plt.subplots(1, 1, figsize = (10, 4))
ax.plot(df_month['Month'], df_month['month_total'])
ax.set_title('Monthly Consumption of Electricity')
```


Monthly Consumption of Electricity
6000
5000
4000
3000
2000
1000
6.0
6.5
7.0
7.5
8.0
8.5
9.0