

Question No. 2

주어진 3개의 파일들은 한 공장의 전력 사용량에 대한 데이터로써, 각각 날씨와 온도, 용도별 전력량계, 전력 총 사용량을 담고있다.

해당 데이터를 종합적으로 이용하여 다음 문제를 풀이하시오.

데이터 파일 설명:

1. E15Q21_usage.csv
 - 900초마다 기록된 900초 단위 전력 총 사용량
 - 1번 컬럼: Datetime (UnixTimestamp)
 - 2번 컬럼: Usage
2. E15Q22_weather.csv
 - 일자별 평균 기온
 - 1번 컬럼: Date (YYYY-MM-DD)
 - 2번 컬럼: Daily Average Temperature
3. E15Q23_usage_history.tsv
 - 1분에 2번씩 기록된 각 용도별 전력 누적사용량
 - 1번 컬럼: Time (HH:MM)
 - 2번 컬럼: Weather Class (A/B/C/D)
 - 3-7번 컬럼: 각 용도(A/B/C/D/E)별 전력 누적 사용량

① 첫번째 제공 파일의 총사용량 컬럼을 용도별로 분류하고, 연월과 사용 목적별로 전력의 하루 평균 사용량을 구하여 도표를 도출하시오.

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
usage = pd.read_csv(r'C:\Users\정재훈\Desktop\WADP 준비\WDataAnaIPrac-master\WDataAnaIPrac-master\WE15Wc
weather = pd.read_csv(r'C:\Users\정재훈\Desktop\WADP 준비\WDataAnaIPrac-master\WDataAnaIPrac-master\WE15
usage_history = pd.read_table(r'C:\Users\정재훈\Desktop\WADP 준비\WDataAnaIPrac-master\WDataAnaIPrac-ma
```

C:\Wanaconda\lib\site-packages\Wipykernel_launcher.py:3: FutureWarning: read_table is deprecated, use read_csv instead, passing sep='Wt'.

This is separate from the ipykernel package so we can avoid doing imports until

In [3]:

```
usage.head()
```

Out[3]:

	timestamp	amount
0	1504224000	329.257482
1	1504224900	326.940143
2	1504225800	329.305272
3	1504226700	313.669972
4	1504227600	271.574547

In [4]:

```
usage_history.head()
```

Out[4]:

	time	wclass	A	B	C	D	E
0	00:00	C	2.000014	3.878765	3.366386	2.580289	0.796893
1	00:00	C	3.433473	9.688855	4.190782	2.580289	0.796893
2	00:01	C	3.553828	9.688855	8.330437	3.852316	0.796893
3	00:01	B	6.226130	11.278011	9.277804	7.297126	5.110191
4	00:02	B	6.501121	14.323436	11.711736	8.526549	11.444439

In [6]:

```
usage.shape
```

Out[6]:

(6720, 2)

In [7]:

```
usage_history.shape
```

Out[7]:

(201600, 7)

In [8]:

```
201600/6720
```

Out[8]:

30.0

In [11]:

```
temp = usage_history[['A', 'B', 'C', 'D', 'E']] - usage_history[['A', 'B', 'C', 'D', 'E']].shift(30)
```

In [17]:

```
temp = temp.dropna()
```

In [50]:

```
pd.DataFrame(usage_history[['A', 'B', 'C', 'D', 'E']].iloc[29]).T
```

Out[50]:

	A	B	C	D	E
29	38.197242	89.010833	62.74631	64.966934	74.336163

In [53]:

```
merged = pd.concat([pd.DataFrame(usage_history[['A', 'B', 'C', 'D', 'E']].iloc[29]).T, temp.iloc[range(29, 6720)]]
```

In [54]:

```
merged['sum'] = merged['A'] + merged['B'] + merged['C'] + merged['D'] + merged['E']
```

In [69]:

```
merged.shape, usage.shape
```

Out[69]:

```
((6720, 6), (6720, 2))
```

In [73]:

```
merged = merged.reset_index(drop = True)
```

In [94]:

```
data = pd.concat([usage, merged], axis = 1)
```

In [104]:

```
time = pd.DataFrame(pd.to_datetime(data.timestamp, unit = 's'))
```

In [105]:

```
time.columns = ['time']
```

In [107]:

```
data1 = pd.concat([data, time], axis = 1)
```

In [109]:

```
data1['hour'] = data1.time.dt.hour
```

In [110]:

```
data1['month'] = data1.time.dt.month
```

In [113]:

```
grouped = data1[['A', 'B', 'C', 'D', 'E', 'month']].groupby('month').agg('mean')
```

In [115]:

```
grouped.index = ['202001', '202002', '202003']
```

In [116]:

```
grouped
```

Out[116]:

	A	B	C	D	E
202001	46.038974	79.030711	75.954619	70.711437	84.956577
202002	45.189092	80.516384	74.983270	70.539364	85.179792
202003	44.255227	78.000228	74.050265	70.665597	81.625251

YYYYMM	A	B	C	D	E
202001	—	—	—	—	—
202002	—	—	—	—	—
202003	—	—	—	—	—

② 요일별 평균 전력사용량을 도출하시오. 또한 가로축을 요일, 세로축을 평균사용량으로 하여 요일별 평균 사용량을 시각화하여 제출하시오.

In [123]:

```
data1['weekday'] = data1.time.dt.weekday_name
```

In [126]:

```
data1.head()
```

Out[126]:

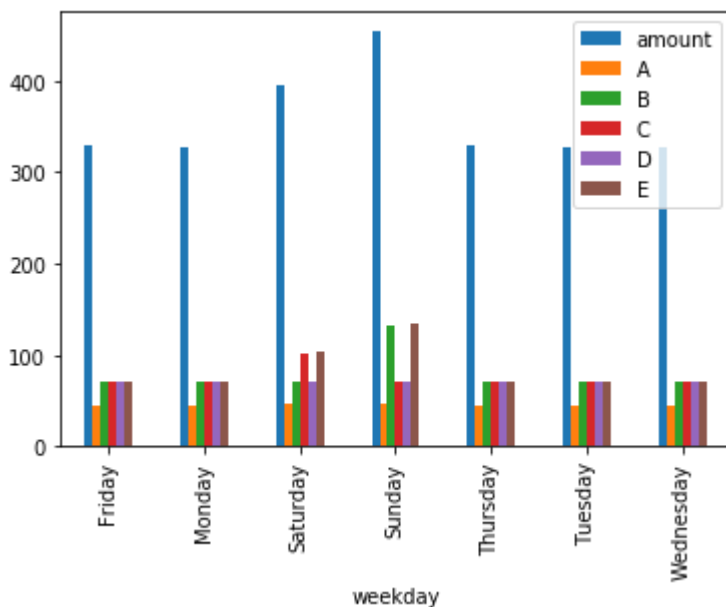
	timestamp	amount	A	B	C	D	E	sum
0	1504224000	329.257482	38.197242	89.010833	62.746310	64.966934	74.336163	329.257482
1	1504224900	326.940143	40.546283	73.279510	78.688553	69.288731	65.137066	326.940143
2	1504225800	329.305272	50.061684	74.490450	66.380516	76.157617	62.215005	329.305272
3	1504226700	313.669972	45.556235	64.579094	72.612875	59.656396	71.265372	313.669972
4	1504227600	271.574547	45.141068	78.651412	40.968460	53.955836	52.857772	271.574547

In [128]:

```
data1[['amount', 'A', 'B', 'C', 'D', 'E', 'weekday']].groupby('weekday').agg('mean').plot.bar()
```

Out[128]:

<matplotlib.axes._subplots.AxesSubplot at 0x2f2449d4c50>



In [135]:

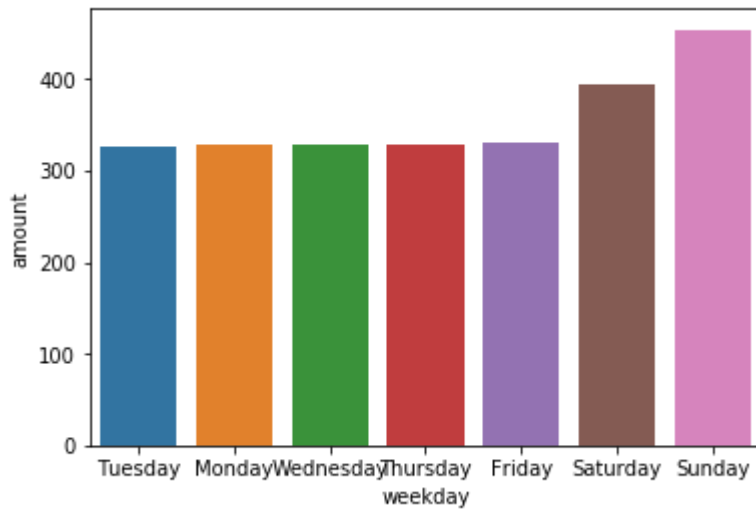
```
grouped2 = data1[['amount', 'weekday']].groupby('weekday').agg('mean').sort_values(by = 'amount')
```

In [138]:

```
sns.barplot(grouped2.index, grouped2.amount)
```

Out[138]:

<matplotlib.axes._subplots.AxesSubplot at 0x2f246b50208>



③ 요일별 총 전력 사용량의 평균값의 차이를 분석하여, 가장 큰 차이를 보이는 요일은 어떤 요일인지 제시하시오.

In [141]:

```
data1[['amount', 'weekday']]
```

Out[141]:

	amount	weekday
0	329.257482	Friday
1	326.940143	Friday
2	329.305272	Friday
3	313.669972	Friday
4	271.574547	Friday
5	323.703523	Friday
6	316.650322	Friday
7	320.071847	Friday
8	348.990117	Friday
9	346.091139	Friday
10	356.633475	Friday
11	303.130775	Friday
12	280.879298	Friday
13	358.973558	Friday
14	330.625229	Friday
15	374.283517	Friday
16	333.324335	Friday
17	332.911312	Friday
18	335.310167	Friday
19	366.873043	Friday
20	345.010102	Friday
21	325.518012	Friday
22	301.939024	Friday
23	350.420266	Friday
24	341.798243	Friday
25	361.390448	Friday
26	335.632616	Friday
27	338.152373	Friday
28	371.708929	Friday
29	318.868516	Friday
...
6690	324.625189	Thursday
6691	325.642946	Thursday
6692	373.167109	Thursday

	amount	weekday
6693	295.432027	Thursday
6694	330.141848	Thursday
6695	322.746100	Thursday
6696	344.601523	Thursday
6697	315.051308	Thursday
6698	337.723637	Thursday
6699	336.786661	Thursday
6700	326.134411	Thursday
6701	295.057151	Thursday
6702	320.297116	Thursday
6703	319.412167	Thursday
6704	292.849463	Thursday
6705	322.101496	Thursday
6706	322.945772	Thursday
6707	329.309477	Thursday
6708	309.049143	Thursday
6709	358.278524	Thursday
6710	355.494148	Thursday
6711	335.252356	Thursday
6712	326.127765	Thursday
6713	330.795465	Thursday
6714	322.284219	Thursday
6715	309.533838	Thursday
6716	323.631351	Thursday
6717	307.982679	Thursday
6718	346.212419	Thursday
6719	301.703415	Thursday

6720 rows × 2 columns

In [146]:

```
data1[data1.weekday == 'Friday']['amount']
```

Out[146]:

```
0      329.257482
1      326.940143
2      329.305272
3      313.669972
4      271.574547
5      323.703523
6      316.650322
7      320.071847
8      348.990117
9      346.091139
10     356.633475
11     303.130775
12     280.879298
13     358.973558
14     330.625229
15     374.283517
16     333.324335
17     332.911312
18     335.310167
19     366.873043
20     345.010102
21     325.518012
22     301.939024
23     350.420266
24     341.798243
25     361.390448
26     335.632616
27     338.152373
28     371.708929
29     318.868516
...
6114   311.607191
6115   326.450769
6116   339.496706
6117   347.126476
6118   369.341652
6119   311.899689
6120   352.720466
6121   310.731922
6122   326.447369
6123   310.988530
6124   322.644622
6125   352.463936
6126   335.072336
6127   340.226993
6128   329.926365
6129   294.553427
6130   316.480793
6131   313.385056
6132   313.485049
6133   346.098878
6134   324.214706
6135   369.373704
6136   304.812758
6137   291.123239
```

```
6138    323.241511
6139    346.762443
6140    307.962384
6141    326.314349
6142    298.401877
6143    371.411545
Name: amount, Length: 960, dtype: float64
```

In [152]:

```
from scipy.stats import f_oneway
fstat, pval = f_oneway(data1[data1.weekday == 'Friday']['amount'], data1[data1.weekday == 'Saturday']
                        data1[data1.weekday == 'Sunday']['amount'], data1[data1.weekday == 'Monday']
                        data1[data1.weekday == 'Tuesday']['amount'], data1[data1.weekday == 'Wednesda
                        data1[data1.weekday == 'Thursday']['amount'])
```

In [153]:

```
print(pval)
```

0.0

④ 각 날짜별 평균 기온과 용도별 전력사용량의 관계를 분석하여, 기온과 가장 밀접한 관계를 지닌 사용 용도의 종류를 제시하시오.

In [162]:

```
data1.head()
```

Out [162]:

	timestamp	amount	A	B	C	D	E	sum
0	1504224000	329.257482	38.197242	89.010833	62.746310	64.966934	74.336163	329.257482
1	1504224900	326.940143	40.546283	73.279510	78.688553	69.288731	65.137066	326.940143
2	1504225800	329.305272	50.061684	74.490450	66.380516	76.157617	62.215005	329.305272
3	1504226700	313.669972	45.556235	64.579094	72.612875	59.656396	71.265372	313.669972
4	1504227600	271.574547	45.141068	78.651412	40.968460	53.955836	52.857772	271.574547

In [163]:

```
weather.head()
```

Out[163]:

	dt	avg_temp
0	2017-09-01	26.747546
1	2017-09-02	26.316174
2	2017-09-03	20.353118
3	2017-09-04	14.575156
4	2017-09-05	23.138287

In [166]:

```
data1['dt'] = data1.time.dt.strftime("%Y-%m-%d")
```

In [169]:

```
weather_merged = pd.merge(data1, weather, how = 'left')
```

In [172]:

```
sns.heatmap(weather_merged[['amount', 'A', 'B', 'C', 'D', 'E', 'avg_temp']].corr(), annot = True, cmap =
```

Out[172]:

<matplotlib.axes._subplots.AxesSubplot at 0x2f24311fc18>

