


2주차 미니 프로젝트 (북촌한옥마을 CCTV 분석)

≡ 태그	2주차
≡ 종류	미니 프로젝트
📅 학습일	@2023년 1월 27일 → 2023년 1월 30일

 미니 프로젝트를 수행하고 보고서를 작성해주세요 😊

보고서 작성하기

▼ 라이브러리 설치 및 호출

```
import requests
import json
from urllib.parse import urlencode, unquote
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import json
import datetime as dt
import os
import time
import scipy.stats as stats
from scipy.stats import anderson
```

▼ API 호출 및 데이터 프레임 생성

```
api_key = '(------)'
file_type = 'json'
service_name = 'BukChonInOutPeopleInfo'
```

```
start = 1
end = 1000
d = pd.DataFrame()

for i in range(25):
    url = "http://openapi.seoul.go.kr:8088"
    url = '/'.join([url, api_key, file_type, service_name, str(start), str(end), ''])
    # print(url)
    response = requests.get(url)
    tmp = response.json()
    # print(tmp)
    data_tmp = pd.DataFrame(tmp['BukChonInOutPeopleInfo']['row'])
    d = pd.concat([d, data_tmp])
    start += 1000
    end += 1000
    # print(start, end)
d.reset_index(drop=True, inplace=True)
d
```

	DEVICEID	DEVICENAME	DESCRIPTION	STARTTIME	ENDTIME	INCOUNT	OUTCOUNT
0	2.0	계동길 69	계동교회 앞	2022-11-30 21:10:41	2022-11-30 21:20:41	9386.0	13749.0
1	1.0	울곡로3길 50	덕성여고 앞	2022-11-30 21:10:41	2022-11-30 21:20:41	7903.0	5279.0
2	2.0	계동길 69	계동교회 앞	2022-11-30 21:20:41	2022-11-30 21:30:41	9398.0	13776.0
3	1.0	울곡로3길 50	덕성여고 앞	2022-11-30 21:20:41	2022-11-30 21:30:41	7927.0	5312.0
4	2.0	계동길 69	계동교회 앞	2022-11-30 21:30:41	2022-11-30 21:40:41	9414.0	13800.0
...
24995	1.0	울곡로3길 50	덕성여고 앞	2023-01-15 16:40:41	2023-01-15 16:50:41	2022.0	1851.0
24996	2.0	계동길 69	계동교회 앞	2023-01-15 16:50:41	2023-01-15 17:00:41	2424.0	3085.0
24997	1.0	울곡로3길 50	덕성여고 앞	2023-01-15 16:50:41	2023-01-15 17:00:41	2051.0	1941.0
24998	2.0	계동길 69	계동교회 앞	2023-01-15 17:00:41	2023-01-15 17:10:41	2463.0	3121.0
24999	1.0	울곡로3길 50	덕성여고 앞	2023-01-15 17:00:41	2023-01-15 17:10:41	2090.0	2001.0

25000 rows x 7 columns

API 호출과 크롤링을 통해 총 2만 5천개의 데이터를 데이터 프레임에 저장

▼ 파생변수 추가 등 데이터 프레임 변환

```
d.dtypes
```

```
DEVICEID      float64
DEVICENAME     object
DESCRIPTION    object
STARTTIME      object
ENDTIME        object
INCOUNT      float64
OUTCOUNT      float64
dtype: object
```

```
d['ENDTIME'] = pd.to_datetime(d['ENDTIME'])
d.dtypes
```

```
DEVICEID      float64
DEVICENAME     object
DESCRIPTION    object
STARTTIME      object
ENDTIME        datetime64[ns]
INCOUNT      float64
OUTCOUNT      float64
dtype: object
```

‘ENDTIME’ 컬럼 타입을 연산에 용이하도록 datetime으로 변환

```
d['MONTH'] = d['ENDTIME'].dt.month
d
```

	DEVICEID	DEVICENAME	DESCRIPTION	STARTTIME	ENDTIME	INCOUNT	OUTCOUNT	MONTH
0	2.0	계동길 69	계동교회 앞	2022-11-30 21:10:41	2022-11-30 21:20:41	9386.0	13749.0	11
1	1.0	울곡로3길 50	덕성여고 앞	2022-11-30 21:10:41	2022-11-30 21:20:41	7903.0	5279.0	11
2	2.0	계동길 69	계동교회 앞	2022-11-30 21:20:41	2022-11-30 21:30:41	9398.0	13776.0	11
3	1.0	울곡로3길 50	덕성여고 앞	2022-11-30 21:20:41	2022-11-30 21:30:41	7927.0	5312.0	11
4	2.0	계동길 69	계동교회 앞	2022-11-30 21:30:41	2022-11-30 21:40:41	9414.0	13800.0	11
...
24995	1.0	울곡로3길 50	덕성여고 앞	2023-01-15 16:40:41	2023-01-15 16:50:41	2022.0	1851.0	1
24996	2.0	계동길 69	계동교회 앞	2023-01-15 16:50:41	2023-01-15 17:00:41	2424.0	3085.0	1
24997	1.0	울곡로3길 50	덕성여고 앞	2023-01-15 16:50:41	2023-01-15 17:00:41	2051.0	1941.0	1
24998	2.0	계동길 69	계동교회 앞	2023-01-15 17:00:41	2023-01-15 17:10:41	2463.0	3121.0	1
24999	1.0	울곡로3길 50	덕성여고 앞	2023-01-15 17:00:41	2023-01-15 17:10:41	2090.0	2001.0	1

25000 rows × 8 columns

```
d['DOW'] = d['ENDTIME'].dt.weekday ##요일을 추가 0:월요일 1:화요일 ... 6:일요일
d['DATE'] = d['ENDTIME'].dt.date
d
```

	DEVICEID	DEVICENAME	DESCRIPTION	STARTTIME	ENDTIME	INCOUNT	OUTCOUNT	MONTH	DOW	DATE
0	2.0	계동길 69	계동교회 앞	2022-11-30 21:10:41	2022-11-30 21:20:41	9386.0	13749.0	11	2	2022-11-30
1	1.0	울곡로3길 50	덕성여고 앞	2022-11-30 21:10:41	2022-11-30 21:20:41	7903.0	5279.0	11	2	2022-11-30
2	2.0	계동길 69	계동교회 앞	2022-11-30 21:20:41	2022-11-30 21:30:41	9398.0	13776.0	11	2	2022-11-30
3	1.0	울곡로3길 50	덕성여고 앞	2022-11-30 21:20:41	2022-11-30 21:30:41	7927.0	5312.0	11	2	2022-11-30
4	2.0	계동길 69	계동교회 앞	2022-11-30 21:30:41	2022-11-30 21:40:41	9414.0	13800.0	11	2	2022-11-30
...
24995	1.0	울곡로3길 50	덕성여고 앞	2023-01-15 16:40:41	2023-01-15 16:50:41	2022.0	1851.0	1	6	2023-01-15
24996	2.0	계동길 69	계동교회 앞	2023-01-15 16:50:41	2023-01-15 17:00:41	2424.0	3085.0	1	6	2023-01-15
24997	1.0	울곡로3길 50	덕성여고 앞	2023-01-15 16:50:41	2023-01-15 17:00:41	2051.0	1941.0	1	6	2023-01-15
24998	2.0	계동길 69	계동교회 앞	2023-01-15 17:00:41	2023-01-15 17:10:41	2463.0	3121.0	1	6	2023-01-15
24999	1.0	울곡로3길 50	덕성여고 앞	2023-01-15 17:00:41	2023-01-15 17:10:41	2090.0	2001.0	1	6	2023-01-15

25000 rows × 10 columns

추후 시각화를 위해 'MONTH' 및 'DOW', 'DATE' 컬럼 추가

```
d[d['DEVICEID']==4].groupby('MONTH').size()
```

```
MONTH
10      715
11     2858
dtype: int64
```

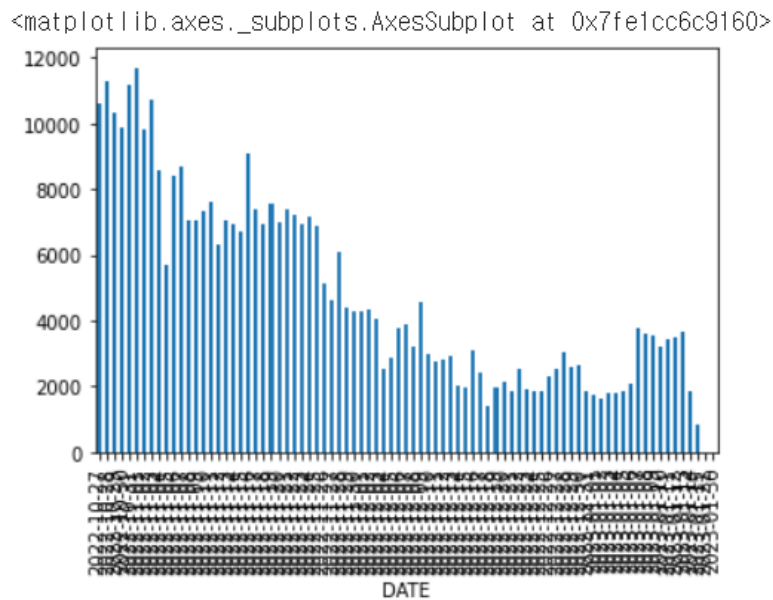
? 데이터에서 'DEVICEID'가 4인 데이터를 보면 10월 데이터가 적고 거의 11월에 모여있는 것을 확인할 수 있다. 원인을 찾기 위해 직접 사이트를 찾아본 결과 해당 위치의 CCTV는 10-11월 중으로 철거된 것으로 보인다. 따라서 적절한 분석을 위해 제거하고 진행할 것이다.

```
## DEVICEID가 4인 데이터 제거하는 코드
d = d[d['DEVICEID'] < 3]
len(d[d['DEVICEID'] == 4])
```

0

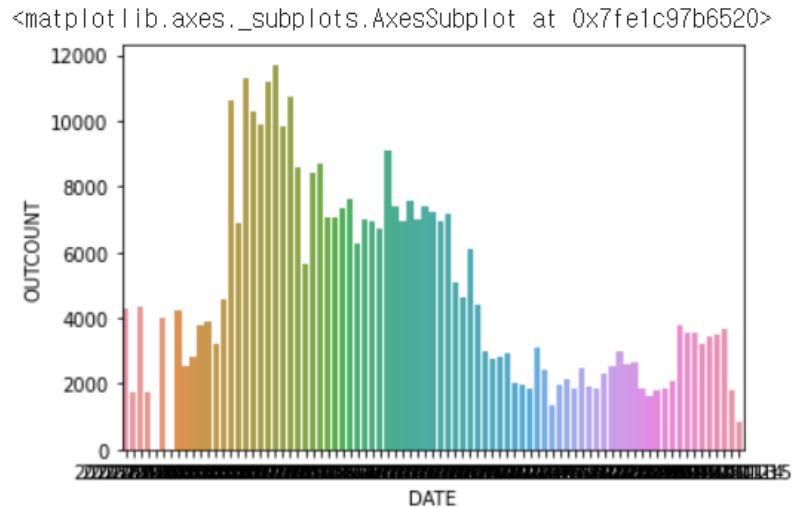
▼ 월별 및 요일별 유동인구 시각화

```
d.groupby('DATE')['OUTCOUNT'].mean().plot.bar()
```



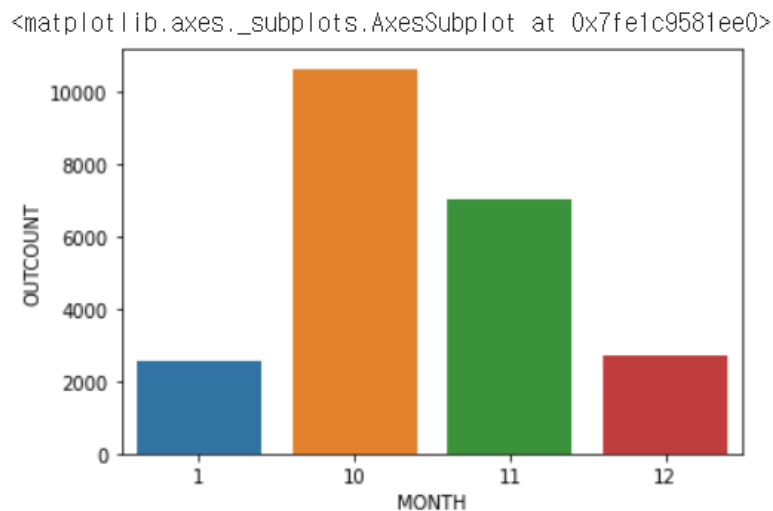
? 일자별로 데이터를 시각화 했을 때 어떤 경향성이 보이는 것 같아 월별로도 데이터를 groupby 하여 살펴볼 것이다. 날씨가 추워지면서 북촌 한옥마을 방문 유동인구에 변화가 있는 것으로 판단하였다.

```
sns.barplot(data=d, x='DATE', y='OUTCOUNT', ci=False)
```



➡ seaborn 라이브러리로 가장 최신 시간을 기점으로 과거로 향하도록 그래프를 그려주었다. 즉 10월에서 1월로 시간이 흐를수록 북촌한옥마을 유동인구가 감소하는 경향을 보인다.

```
sns.barplot(data=d, x='MONTH', y='OUTCOUNT', ci=False)
```



이는 월별 데이터의 막대 그래프를 통해서도 확인할 수 있다.

데이터가 월별로 유의한 차이가 있는지 확인하기 위해 분산 분석을 수행할 것이다. 분산 분석에 앞서 그 조건을 만족하는지 확인한다.

정규성 검정

```
tmp1 = d[["MONTH", "OUTCOUNT"]]
re = anderson(tmp1[tmp1["MONTH"]==10]["OUTCOUNT"])
re
```

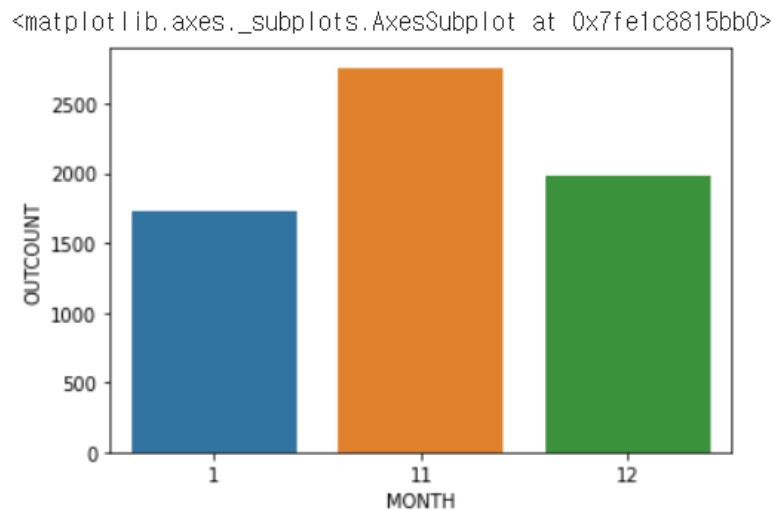
```
AndersonResult(statistic=32.25661201355115, critical_values=array([0.573, 0.652, 0.783, 0.913, 1.086]), significance_level=array([15. , 10. , 5. , 2.5, 1. ]))
```

정규성을 만족하지 못하므로 비모수적 방법인 Kruskal-Wallis 검정을 진행

```
oct = tmp1[tmp1["MONTH"]==10]["OUTCOUNT"]
nov = tmp1[tmp1["MONTH"]==11]["OUTCOUNT"]
dec = tmp1[tmp1["MONTH"]==12]["OUTCOUNT"]
jan = tmp1[tmp1["MONTH"]==1]["OUTCOUNT"]
stats.kruskal(oct,nov,dec,jan)
```

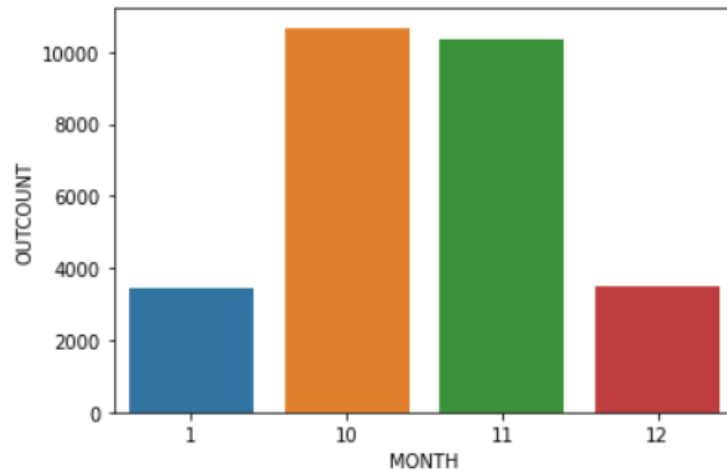
```
KruskalResult(statistic=1666.42981004179, pvalue=0.0)
```

```
sns.barplot(data=schoolzone, x='MONTH', y='OUTCOUNT',ci=False)
```



```
sns.barplot(data=churchzone, x='MONTH', y='OUTCOUNT',ci=False)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fe1c8789400>



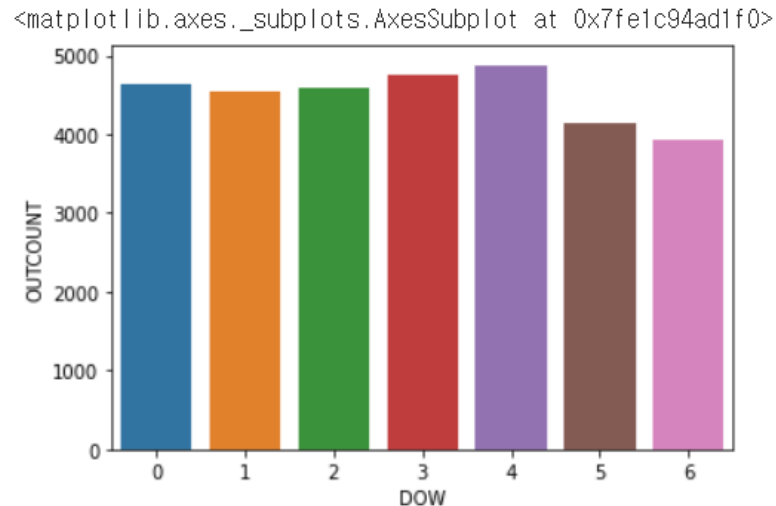
→ INCOUNT 시각화도 비슷한 결과를 보여 그래프는 생략하였다. 학교 앞과 교회 앞의 월별 유동인구를 시각화 하면, 학교 앞 월별 유동인구는 2000을 전후로 비슷하다. 반면, 12월에 들어서 교회 앞 유동인구가 1만에서 4천 이하로 급감한 것을 알 수 있다. 그러나 그럼에도 불구하고 교회 앞 유동인구가 훨씬 많다.

→ Kruskal-Wallis 검정의 유의수준이 0.0으로 매우 낮다. 즉, 월별 방문자수는 유의한 차이가 있다. 기온이 떨어짐에 따라 방문자 수가 감소하는 것으로 파악된다. 10월 말의 기온은 8-16도 정도로 아주 추운 날씨는 아니나, 11월 중순 이후로 0-7도로 떨어졌으며, 12월은 본격적으로 영하까지 온도가 내려갔다. 또한 12월 말이 되면 최고 기온이 대략 2도 정도를 평균으로 가지며 매우 추운 날씨임을 확인할 수 있었다. 마찬가지로 올해 1월의 경우 매우 추운 날들이 나타났다. 이처럼 기온이 하나의 영향을 주는 지표로 생각된다. 각 일자마다 기록된 기온 데이터를 확보한다면 이에 대한 분석을 진행하는 것도 좋을 것이다.

참고 자료

<https://ko.weatherspark.com/m/142033/1/%EB%8C%80%ED%95%9C%EB%AF%BC%EA%B5%AD-%EC%84%9C%EC%9A%B8%ED%8A%B9%EB%B3%84%EC%8B%9C%EC%97%90%EC%84%9C-1%EC%9B%94%EC%9D%98-%ED%8F%89%EA%B7%A0-%EB%82%A0%EC%94%A8#Figures-Temperature>

```
sns.barplot(data=d, x='DOW', y='OUTCOUNT',ci=False)
```



? 위 막대 그래프를 보면 주말과 주중에 차이가 있는 것으로 보인다. x축 인덱스 0부터 월요일 순이다. 유의한 차이가 있는지 확인하기 위해 분산 분석이 필요하다. 이를 위해 그 조건을 만족하는지 확인할 것이다.

정규성 검정

```
tmp2 = d[["DOW", "OUTCOUNT"]]
re = anderson(tmp2[tmp2["DOW"]!=0]["OUTCOUNT"])
re
```

```
AndersonResult(statistic=288.35539631939446, critical_values=array([0.575, 0.655, 0.786, 0.917, 1.091]), significance_level=array([15., 10., 5., 2.5, 1.]))
```

정규성을 만족하지 못하므로 비모수적 방법인 Kruskal-Wallis 검정을 진행

```
mon = tmp2[tmp2["DOW"]==0]["OUTCOUNT"]
tue = tmp2[tmp2["DOW"]==1]["OUTCOUNT"]
wed = tmp2[tmp2["DOW"]==2]["OUTCOUNT"]
thu = tmp2[tmp2["DOW"]==3]["OUTCOUNT"]
fri = tmp2[tmp2["DOW"]==4]["OUTCOUNT"]
sat = tmp2[tmp2["DOW"]==5]["OUTCOUNT"]
sun = tmp2[tmp2["DOW"]==6]["OUTCOUNT"]

stats.kruskal(mon, tue, wed, thu, fri, sat, sun)
```

```
KruskalResult(statistic=57.6963035906764, pvalue=1.320380051172481e-10)
```

➡ p값이 10의 -9승 수준으로 매우 낮다. 즉, **요일별로 방문자 수에 차이가 있다고 할 수 있다**. 어떠한 원인으로 이러한 결과가 나타났는지는 더 조사가 필요하다. **일요일의 방문자 수가 가장 적은 것은 일요일이 골목이 쉬는 날로 지정돼, 방문이 불가한 날이기 때문**으로 보인다. **그러나 이러한 원칙에도 불구하고 방문객들이 상당수 있는 것으로 추정된다**.

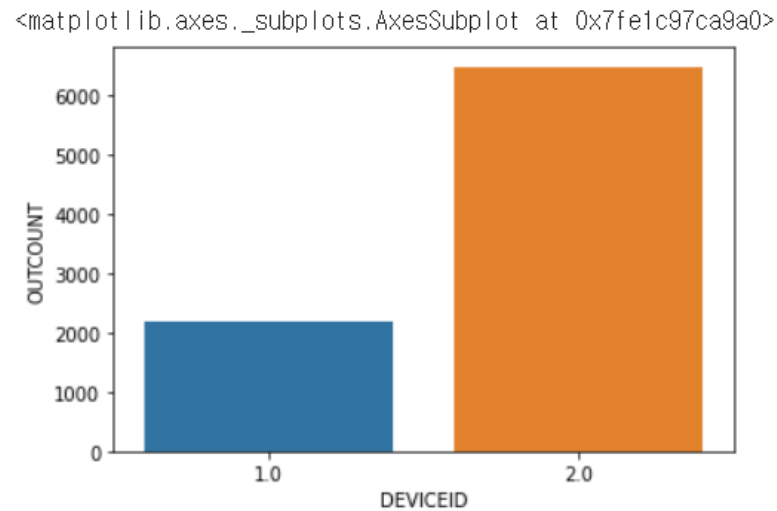
참고자료

<https://brunch.co.kr/@travel-bike/624>

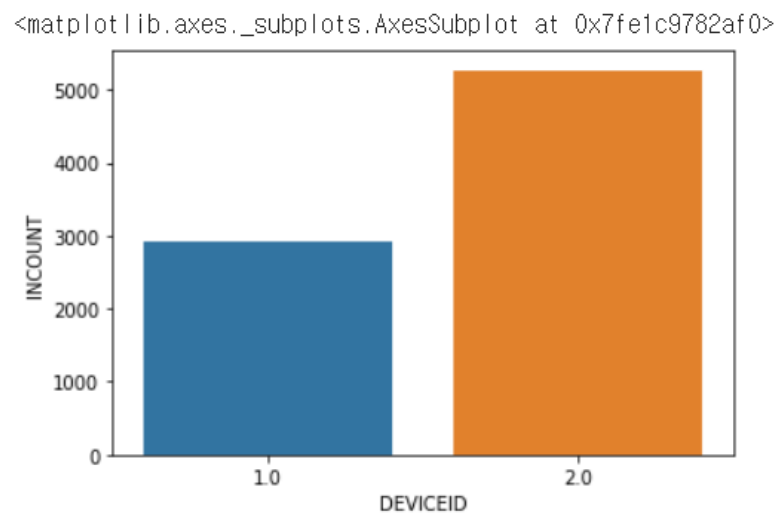
<https://www.metroseoul.co.kr/article/2018072400093>

▼ CCTV 디바이스별 유동인구 시각화

```
sns.barplot(data=d, x='DEVICEID', y='OUTCOUNT',ci=False)
```



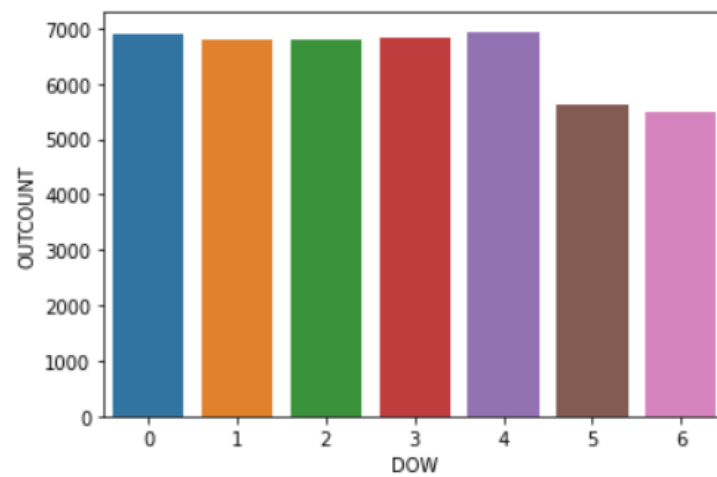
```
sns.barplot(data=d, x='DEVICEID', y='INCOUNT',ci=False)
```



위 막대 그래프를 보면 2번 디바이스의 위치에 1번 디바이스에 비해 더 많은 유동 인구가 있는 것으로 파악된다. 즉 **CCTV에 따라 유동인구 수의 차이가 큰 것으로 나타났다**. 2번 디바이스는 'DESCRIPTION' 열에 따라 계동교회 앞임을 알 수 있다.

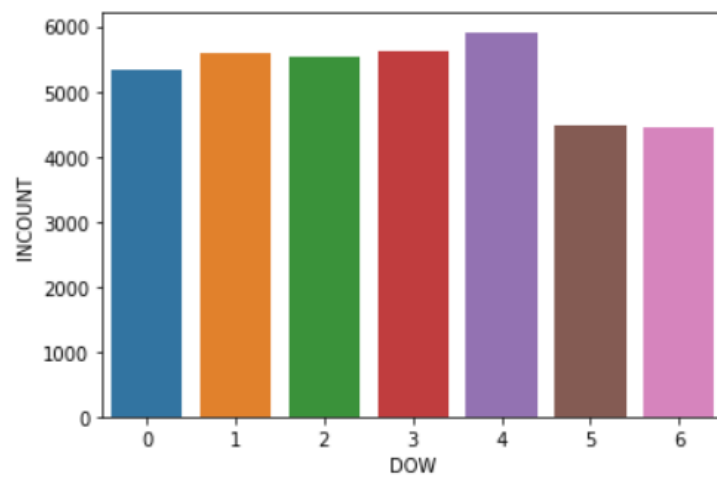
```
churchzone = d[d['DEVICEID'] == 2]
sns.barplot(data=churchzone, x='DOW', y='OUTCOUNT',ci=False)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fe1c89ac670>



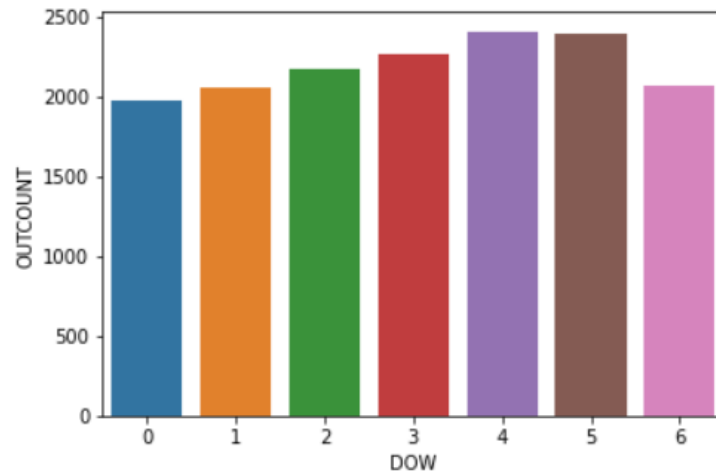
```
sns.barplot(data=churchzone, x='DOW', y='INCOUNT',ci=False)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fe1c897d9a0>



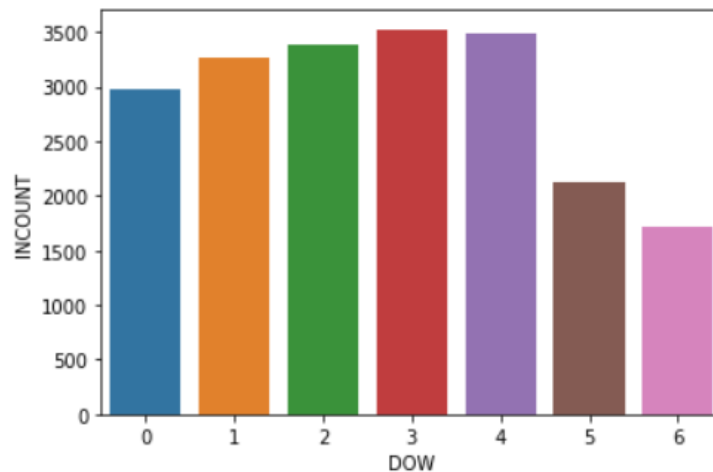
```
sns.barplot(data=schoolzone, x='DOW', y='OUTCOUNT',ci=False)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fe1c8d6b3d0>



```
sns.barplot(data=schoolzone, x='DOW', y='INCOUNT', ci=False)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fe1c8898b50>



주일임을 감안하여도 계동교회 앞 2번 CCTV의 유동인구는 주중보다 더 적은 것을 알 수 있다. 그러나 유입 인구와 유출 인구가 비슷한 반면, 덕성여고 앞 1번 CCTV의 주말 유출 인구는 많고 유입 인구는 훨씬 적다. 편의점 북촌한옥마을 지점의 주말에는 학생들보다 **관광객과 교회 방문객들에 집중하는 것이 좋을 것**이라 판단된다.

결론



MONTH, DOW, DEVICEID 변수가 머신러닝의 특성으로 적절할 것으로 보인다. 추가적으로 가능하다면 **기온** 등 기상 데이터를 추가하여 함께 고려하는 것이 필요하다. 대체적으로 교회 앞 2번 CCTV의 유동인구가 더 많다. 또한, CCTV별 유동인구는 보통 주중이 높으나, 특히 주말에는 교회 앞 유동인구가 훨씬 많으므로 **주말 한정 교회 방문객이나 관광객들을 대상으로 한 프로모션을 기획해도 좋을 것**이다.