DataScience

인구 데이터와 CCTV데이터를 통해 데이터 다루는법을 익히고 분석 하기

필요한 모듈 import

```
import pandas as pd
import numpy as np
```

```
# CCTV 데이터와 인구 데이터 합치고 분석하기
CCTV_Seoul = pd.read_csv('./data/01. CCTV_in_Seoul.csv', encoding='utf-8')
CCTV_Seoul.head()

CCTV_Seoul.columns

CCTV_Seoul.columns[0]
```

```
# 컬럼명 변경 : 기관명을 구별로 변경
CCTV_Seoul.rename(columns={CCTV_Seoul.columns[0] : '구별'}, inplace=True)
CCTV_Seoul.head()
```

```
# 인구 데이터 읽기 1
pop_Seoul = pd.read_excel('./data/01. population_in_Seoul.xls', encoding='u
pop_Seoul.head()

# 인구 데이터 읽기 2 - 필요한 데이터만 선별하여 읽기
pop_Seoul = pd.read_excel('./data/01. population_in_Seoul.xls',
                          header=2,    # 엑셀에서 3번째 행을 헤더로 해서 읽기
                          usecols='B, D, G, J, N',
                          encoding='utf-8')
pop_Seoul.head()
```

```
    자치구    계    계.1     계.2     65세이상고령자
0   합계 10197604.0  9926968.0   270636.0    1321458.0
1   종로구   162820.0    153589.0    9231.0  25425.0
2   중구 133240.0    124312.0    8928.0  20764.0
3   용산구   244203.0    229456.0    14747.0 36231.0
4   성동구   311244.0    303380.0    7864.0  39997.0
```

```
# 알기 쉬운 컬럼명으로 변경
pop_Seoul.rename(columns={pop_Seoul.columns[0]:'구별',
                          pop_Seoul.columns[1]:'인구수',
                          pop_Seoul.columns[2]:'한국인',
                          pop_Seoul.columns[3]:'외국인',
                          pop_Seoul.columns[4]:'고령자'}, inplace=True)
pop_Seoul.head()
```

```
구별 인구수    한국인    외국인    고령자
0   합계 10197604.0  9926968.0   270636.0    1321458.0
1   종로구   162820.0    153589.0    9231.0  25425.0
2   중구 133240.0    124312.0    8928.0  20764.0
3   용산구   244203.0    229456.0    14747.0 36231.0
4   성동구   311244.0    303380.0    7864.0  39997.0
```

```
# CCTV 데이터 파악하기
CCTV_Seoul.sort_values(by='소계', ascending=True).head(5)   # ascending=True

CCTV_Seoul.sort_values(by='소계', ascending=False).head(5)  # ascending=Fals
```

```
# 최근증가율 = (2016년+2015년+2014년)/2013년도 이전 * 100
CCTV_Seoul['최근증가율']=(CCTV_Seoul['2016년']+CCTV_Seoul['2015년']+\
                      CCTV_Seoul['2014년']) / CCTV_Seoul['2013년도 이전'] * 1


CCTV_Seoul.sort_values(by='최근증가율', ascending=False).head()
```

```
# 서울시 인구 데이터 파악하기
pop_Seoul.head()

# 첫번째 합계 행 삭제
pop_Seoul.drop([0], inplace=True)
pop_Seoul.head()

# '구별' 컬럼의 중복값 제거
pop_Seoul['구별'].unique()

# '구별' 컬럼의 NULL값 확인
pop_Seoul[pop_Seoul['구별'].isnull()]

# '구별' 컬럼의 NULL값 있는 행 제거
pop_Seoul.drop([26], inplace=True)
pop_Seoul.head()
```

> 데이터분석

1. 분석 데이터 수집
2. 수집된 데이터 형식 확인 및 local 전처리
3. 분석 prg에서 수집 데이터 읽기
4. 읽은 데이터 확인 및 2차 전처리

```
# 외국인 비율과 고령자 비율 추가
pop_Seoul['외국인비율'] = pop_Seoul['외국인'] / pop_Seoul['인구수'] * 100
pop_Seoul['고령자비율'] = pop_Seoul['고령자'] / pop_Seoul['인구수'] * 100
pop_Seoul.head()

# 각 컬럼 확인
pop_Seoul.sort_values(by='인구수', ascending=False).head(5)
pop_Seoul.sort_values(by='외국인', ascending=False).head(5)
pop_Seoul.sort_values(by='외국인비율', ascending=False).head(5)
pop_Seoul.sort_values(by='고령자', ascending=False).head(5)
pop_Seoul.sort_values(by='고령자비율', ascending=False).head(5)
```

CCTV 데이터와 인구 데이터 합치고 분석하기

```
# 두 개의 데이터 프레임을 합할 경우 동일 컬럼명은 하나('구별')로 통일 된다.
data_result = pd.merge(CCTV_Seoul, pop_Seoul, on='구별')
data_result.head()
```

```
del data_result['2013년도 이전']
del data_result['2014년']
del data_result['2015년']
del data_result['2016년']
data_result.head()
```

> CCTV에 대한 '소계' 컬럼을 제외한 나머지 CCTV 데이터 삭제

```
# 시각화 작업을 위한 구이름('구별')을 index화
data_result.set_index('구별', inplace=True)
data_result.head()
```

CCTV와 각 컬럼에 대한 상관관계 분석

```
# 상관관계 함수 : np.corrcoef()
np.corrcoef(data_result['고령자비율'],data_result['소계'])

np.corrcoef(data_result['외국인비율'],data_result['소계'])

np.corrcoef(data_result['인구수'],data_result['소계'])

data_result.sort_values(by='소계', ascending=False).head(5)

# CSV 파일로 저장
data_result.to_csv('./data/data_result.csv')
```

CCTV와 인구현황 그래프로 분석하기

```
import platform

# 폰트 설정(특히 한글부분)
import matplotlib.pyplot as plt
from matplotlib import font_manager, rc
plt.rcParams['axes.unicode_minus']=False

if platform.system() == 'Darwin':
    rc('font', family='AppleGothic')
elif platform.system() == 'Windows':
    path = "c:/Windows/Fonts/malgun.ttf"
    font_name = font_manager.FontProperties(fname=path).get_name()
    rc('font', family=font_name)
else:
    print('Unknown system.... sorry')
```

CCTV 비율을 구하고 그에 따른 시각화 작업

```
data_result['CCTV비율'] = data_result['소계'] / data_result['인구수'] * 100

data_result['CCTV비율'].sort_values().plot(kind = 'barh', grid=True, figsize
plt.show()
```

산점도

```
# 산점도 (인구수와 소계)
plt.figure(figsize=(6,6))
plt.scatter(data_result['인구수'], data_result['소계'], s=50)
plt.xlabel('인구수')
plt.ylabel('CCTV')
plt.grid()
plt.show()
```

> 인구수와 CCTV는 상관 계수가 양의 값이므로 산점도와 직선

직성 구하기 (Polyfit을 이용한 회귀선)
polyfit함수를 이용해서 예측 모델 z의 계수를 생성

```
fp1 = np.polyfit(data_result['인구수'], data_result['소계'], 1)
fp1
```

```
# 만들어진 예측 모델을 이용한 그래프 그리기
f1 = np.poly1d(fp1)     # y축 데이터
fx = np.linspace(100000, 700000, 100)   # x축 데이터

plt.figure(figsize=(10, 10))
plt.scatter(data_result['인구수'], data_result['소계'], s=50)
plt.plot(fx, f1(fx), ls='dashed', lw=3, color='g')
plt.xlabel('인구수')
plt.ylabel('CCTV')
plt.grid()
plt.show()
```