강남 3구는 안전한가 - 2

범죄 데이터 구별로 정리하기

```
import pandas as pd
import numpy as np

from crime_in_seoul import station_lng, station_lat
crime_anal_raw = pd.read_csv('./data/02. crime_in_Seoul_include_gu_name.csv
                             encoding='utf-8')

crime_anal_raw.head()
```

pandas의 pivot_table

```
pd.pivot_table(df,                  피벗 테이블을 만들기 위한 기본 데이터
               index = [],          pivot_table의 index를 설정(multi index도 가
               columns = [],        원하는 columns을 설정
               values = [],         columns에 해당하는 값
               aggfunc = [],        분석을 위한 파라미터
                                    예) np.sum, np.mean 사용
               fill_value = 0,      Nan 값을 채우기.
               margins = True)      모든 데이터의 결과를 아래에 붙일 것인지 설정

--> 원하는 컬럼을 인덱스화 시켜서 나머지 데이터들을 재정렬시켜주는 것이 피벗 테이블의
```

```
# pivot_table을 이용
# 저장한 데이터를 관서별에서 구별로..
crime_anal = pd.pivot_table(crime_anal_raw, index='구별', aggfunc=np.sum)
crime_anal.head()

print(crime_anal)
print(type(crime_anal))
```

각 번죄별 검거율을 계산하고,
검거 건수는 검거율로 대체한 후, 검거 건수는 삭제

```
crime_anal['강간검거율'] = crime_anal['강간 검거']/crime_anal['강간 발생']*100
crime_anal['강도검거율'] = crime_anal['강도 검거']/crime_anal['강도 발생']*100
crime_anal['살인검거율'] = crime_anal['살인 검거']/crime_anal['살인 발생']*100
crime_anal['절도검거율'] = crime_anal['절도 검거']/crime_anal['절도 발생']*100
crime_anal['폭력검거율'] = crime_anal['폭력 검거']/crime_anal['폭력 발생']*100

del crime_anal['강간 검거']
del crime_anal['강도 검거']
del crime_anal['살인 검거']
del crime_anal['절도 검거']
del crime_anal['폭력 검거']

print(crime_anal.head())
```

100이 넘는 숫자들은 100으로 처리

```
con_list = ['강간검거율', '강도검거율', '살인검거율', '절도검거율', '폭력검거율']

for column in con_list:
    crime_anal.loc[crime_anal[column] > 100, column] = 100

print(crime_anal.head())

# 컬럼 뒤에 발생이라는 단어 삭제 : rename()를 사용
crime_anal.rename(columns = {'강간 발생' : '강간',
                             '강도 발생' : '강도',
                             '살인 발생' : '살인',
                             '절도 발생' : '절도',
                             '폭력 발생' : '폭력'}, inplace=True)
print(crime_anal.head())
```

데이터 표현을 위해 전처리

강도와 살인은 두자리수,
절도와 폭력은 네 자리수로 구성되어 있어
각각을 비슷한 범위에 놓고 비교하는 것이 편리하기 때문에
각 컬럼별로 정규화(NOMALIZE) 작업

각 항목의 최대값을 1로 두면,
추후 범죄 발생 건수를 종합적으로 비교할 때 편리

간간, 강도, 살인, 절도, 폭력에 대하여
각 컬럼별로 정규화

파이썬의 머신러닝에 관한 모듈 중
scikit learn에 있는 전처리(preprocessing) 도구에는
최소, 최대값을 이용하여 정규화시키는 함수가 존재 : MinMaxScaler()

```
from sklearn import preprocessing

col =['강간', '강도', '살인', '절도', '폭력']

x = crime_anal[col].values
min_max_scaler = preprocessing.MinMaxScaler()

x_scaled = min_max_scaler.fit_transform(x.astype(float))

crime_anal_norm = pd.DataFrame(x_scaled, columns = col, index = crime_anal.
```

정규화된 데이터 프레임에 검거율 추가

```
col2 = ['강간검거율', '강도검거율', '살인검거율', '절도검거율', '폭력검거율']

crime_anal_norm[col2] = crime_anal[col2]

print(crime_anal_norm.head())
```

CCTV_result.csv에서 구별 인구수와 CCTV 개수만 추가

```
result_CCTV = pd.read_csv('./data/01. CCTV_result.csv', encoding='utf-8', i

crime_anal_norm[['인구수', 'CCTV']] = result_CCTV[['인구수', '소계']]

print("인구수와 CCRV 개수 => ", crime_anal_norm.head())
```

발생 건수의 합을 '범죄'라는 컬럼으로 합하여 추가

```
col = ['강간', '강도', '살인', '절도', '폭력']

```

```
crime_anal_norm['범죄'] = np.sum(crime_anal_norm[col], axis=1)

print("범죄라는 컬럼으로 합 => ", crime_anal_norm.head())
```

검거율도 통합하여 추가

```
col = ['강간검거율', '강도검거율', '살인검거율', '절도검거율', '폭력검거율']

crime_anal_norm['검거'] = np.sum(crime_anal_norm[col], axis=1)

print("검거율도 통합 => ", crime_anal_norm.head())
```

시각화 하기

```
import matplotlib.pyplot as plt

import seaborn as sns
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')


print(crime_anal_norm.head())
```

pairplot() 상관관계 : “강도” “살인” “폭력”

```
sns.pairplot(crime_anal_norm, vars=["강도", "살인", "폭력"], kind='reg', size
plt.show()
```

> 강도와 폭력, 살인과 폭력, 강도와 살인 모두 양의 상관관계를 보임

pairplot() 상관관계 : “인구수”, “CCTV”, “살인”, “강도”

```
sns.pairplot(crime_anal_norm, x_vars=["인구수", "CCTV"], y_vars=["살인", "강도
             kind='reg', size=3)

plt.show()
```

> 전체적인 상관계수는 CCTV와 살인의 관계가 낮을지 몰라도
> CCTV가 없을때 살인사건 많은 구간 있음.
> 즉, CCTV수를 기준으로 좌측면에 살인과 강도의 높은 수를 갖는 데이터가 보임.

pairplot() 상관관계: “인구수”, “CCTV”, “살인검거율”, “폭력검거율”

```
sns.pairplot(crime_anal_norm, x_vars=["인구수", "CCTV"], y_vars=["살인검거율"
             kind='reg', size=3)

plt.show()
```

> 살인 및 폭력 검거율과 CCTV의 관계가 음의 상관계수도 보여줌
> 인구수와 살인 및 폭력 검거율도 음의 상관관계를 보임

pairplot() 상관관계: “인구수”, “CCTV”, “절도검거율”, “강도검거율”

```
sns.pairplot(crime_anal_norm, x_vars=["인구수", "CCTV"], y_vars=["절도검거율"
             kind='reg', size=3)

plt.show()
```

> 상관 없다

검거율의 합계인 검거 항목 최고 값을 100으로 한정한 후, 그 값으로 정렬

```
tmp_max = crime_anal_norm['검거'].max()

crime_anal_norm['검거'] = crime_anal_norm['검거'] / tmp_max * 100

crime_anal_norm_sort = crime_anal_norm.sort_values(by='검거', ascending=Fals

crime_anal_norm_sort.head()
```

heatmap으로 시각화

```
target_col = ['강간검거율', '강도검거율', '살인검거율', '절도검거율', '폭력검거율'

crime_anal_norm_sort = crime_anal_norm.sort_values(by='검거', ascending=Fals

plt.figure(figsize=(10,10))

sns.heatmap(crime_anal_norm_sort[target_col],
            annot=True, fmt='f',
            linewidths=.5,  # linewidths는 칸 간격 의미
            cmap='RdPu')

plt.title('범죄 검거 비율 (정규화된 검거의 합으로 정렬)')
plt.show()
```

발생 건수 정렬하여 heatmap으로 시각화

```
target_col = ['강간검거율', '강도검거율', '살인검거율', '절도검거율', '폭력검거율'

crime_anal_norm['범죄'] = crime_anal_norm['범죄'] / 5

crime_anal_norm_sort = crime_anal_norm.sort_values(by='범죄', ascending=Fals

plt.figure(figsize=(10,10))

sns.heatmap(crime_anal_norm_sort[target_col],
            annot=True, fmt='f',
            linewidths=.5,  # linewidths는 칸 간격 의미
            cmap='RdPu')

plt.title('범죄 비율 (정규화된 발생 건수로 정렬)')
plt.show()
```