

smarcle data study 1조

...

-서울시 구별 CCTV 현황 분석-
강인영, 노지민, 정유찬, 유정수

목차

1. CCTV 파일 분석
2. 인구 파일 분석
3. 파일 병합 및 시각화

1. CCTV 파일 분석

...

➤ 서울시 CCTV 현황 파악하기

read_csv를 이용하여 csv파일을 가져온 후 데이터의 첫 5행까지 출력

```
CCTV_Seoul = pd.read_csv('../content/D1. CCTV_in_Seoul.csv', encoding='utf-8')  
CCTV_Seoul.head()
```

	기관명	소계	2013년도 이전	2014년	2015년	2016년
0	강남구	2780	1292	430	584	932
1	강동구	773	379	99	155	377
2	강북구	748	369	120	138	204
3	강서구	884	388	258	184	81
4	관악구	1496	846	260	390	613

➤ 이름 바꾸기

‘기관명’을 ‘구별’로 이름 바꾸기

```
In [5]: CCTV_Seoul.rename(columns={CCTV_Seoul.columns[0] : '구별'}, inplace=True)  
CCTV_Seoul.head()
```

```
Out[5]:
```

	구별	소계	2013년도 이전	2014년	2015년	2016년
0	강남구	2780	1292	430	584	932
1	강동구	773	379	99	155	377
2	강북구	748	369	120	138	204
3	강서구	884	388	258	184	81
4	관악구	1496	846	260	390	613

➤ 이름을 중심으로 정렬하기(1)

CCTV표를 '소계'로 오름차순 정렬 후 5행까지 출력

```
In [46]: CCTV_Seoul.sort_values(by='소계', ascending=True).head(5)
```

Out[46]:	구별	소계	2013년도 이전	2014년	2015년	2016년
9	도봉구	485	238	159	42	386
12	마포구	574	314	118	169	379
17	송파구	618	529	21	68	463
24	중랑구	660	509	121	177	109
23	중구	671	413	190	72	348

➤ 이름을 중심으로 정렬하기(2)

CCTV를 '소계'로 내림차순 정렬 후 5행까지 출력

```
In [47]: CCTV_Seoul.sort_values(by='소계', ascending=False).head(5)
```

```
Out[47]:
```

	구별	소계	2013년도 이전	2014년	2015년	2016년
0	강남구	2780	1292	430	584	932
18	양천구	2034	1843	142	30	467
14	서초구	1930	1406	157	336	398
21	은평구	1873	1138	224	278	468
20	용산구	1624	1368	218	112	398

➤ 열을 만들고 열을 중심으로 정렬하기

‘최근증가율’의 열을 만들어 내림차순 정렬 후 5행까지 출력

```
In [48]: CCTV_Seoul['최근증가율'] = (CCTV_Seoul['2016년'] + CCTV_Seoul['2015년'] + \
                                         CCTV_Seoul['2014년']) / CCTV_Seoul['2013년도 이전'] * 100
CCTV_Seoul.sort_values(by='최근증가율', ascending=False).head(5)
```

Out[48]:

	구별	소계	2013년도 이전	2014년	2015년	2016년	최근 증가율
22	종로구	1002	464	314	211	630	248.922414
9	도봉구	485	238	159	42	386	246.638655
12	마포구	574	314	118	169	379	212.101911
8	노원구	1265	542	57	451	516	188.929889
1	강동구	773	379	99	155	377	166.490765

2. 인구 파일 분석

...

➤ 서울시 인구 수 엑셀 파일 가져오기

- pd.read_excel을 이용하여 엑셀 파일 데이터 가져오기

(이때 utf-8을 사용하여 인코딩해야 함)

```
[6] pop_Seoul = pd.read_excel('/content/01. population_in_Seoul.xls')  
pop_Seoul.head()
```

	기간	자치구	세대	인구	인구.1	인구.2	인구.3	인구.4	인구.5	인구.6	인구.7	인구.8	세대당인구	65세이상고령자
0	기간	자치구	세대	합계	합계	합계	한국인	한국인	한국인	등록외국인	등록외국인	등록외국인	세대당인구	65세이상고령자
1	기간	자치구	세대	계	남자	여자	계	남자	여자	계	남자	여자	세대당인구	65세이상고령자
2	2017.1/4	합계	4202888	10197604	5000005	5197599	9926968	4871560	5055408	270636	128445	142191	2.36	1321458
3	2017.1/4	종로구	72654	162820	79675	83145	153589	75611	77978	9231	4064	5167	2.11	25425
4	2017.1/4	중구	59481	133240	65790	67450	124312	61656	62656	8928	4134	4794	2.09	20764

➤ 필요없는 데이터 지우기 1

- header을 사용하여 세 번째 행부터 읽기
- usecols를 사용하여 사용할 열만 지정하기 (parse_cols는 지원 안 함)

```
pop_Seoul = pd.read_excel('/content/01. population_in_Seoul.xls',  
                           header = 2,  
                           usecols = 'B,D,G,J,N')  
  
pop_Seoul.head()
```

	기간	자치구	세대	인구	인구.1	인구.2	인구.3	인구.4	인구.5	인구.6	인구.7	인구.8	세대당인구	65세이상고령자
0	기간	자치구	세대	합계	합계	합계	한국인	한국인	한국인	등록외국인	등록외국인	등록외국인	세대당인구	65세이상고령자
1	기간	자치구	세대	계	남자	여자	계	남자	여자	계	남자	여자	세대당인구	65세이상고령자
2	2017.1/4	합계	4202888	10197604	5000005	5197599	9926968	4871560	5055408	270636	128445	142191	2.36	1321458
3	2017.1/4	종로구	72654	162820	79675	83145	153589	75611	77978	9231	4064	5167	2.11	25425
4	2017.1/4	중구	59481	133240	65790	67450	124312	61656	62656	8928	4134	4794	2.09	20764

➤ 데이터 컬럼명 바꾸기

- rename을 사용하여 사용할 용도에 맞게 컬럼 이름 바꾸기

```
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

➤ 필요없는 데이터 지우기 2

- drop 명령을 사용하여 필요없는 전체 합계 행(0번째 행) 지우기

```
pop_Seoul.drop([0], inplace = True)  
pop_Seoul.head()
```

	구별	인구수	한국인	외국인	고령자
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
5	광진구	372164.0	357211.0	14953.0	42214.0

➤ Nan 데이터 지우기

- unique를 조사하여 Nan 데이터 값이 존재하는지 조사
- isnull() 을 사용하여 Nan 데이터 추출
- drop() 을 사용하여 Nan 데이터가 있는 행 지우기

```
pop_Seoul['구별'].unique()
```

```
array(['종로구', '중구', '용산구', '성동구', '광진구', '동대문구', '중랑구', '성북구', '강북구',  
      '도봉구', '노원구', '은평구', '서대문구', '마포구', '양천구', '강서구', '구로구', '금천구',  
      '영등포구', '동작구', '관악구', '서초구', '강남구', '송파구', '강동구', nan],  
      dtype=object)
```

```
pop_Seoul[pop_Seoul['구별'].isnull()]
```

	구별	인구수	한국인	외국인	고령자
--	----	-----	-----	-----	-----

26	NaN	NaN	NaN	NaN	NaN
----	-----	-----	-----	-----	-----

```
pop_Seoul.drop([26], inplace = True)  
pop_Seoul.head()
```

➤ 외국인 비율, 고령자 비율 계산

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

	구별	인구수	한국인	외국인	고령자	외국인비율	고령자비율
1	종로구	162820.0	153589.0	9231.0	25425.0	5.669451	15.615404
2	중구	133240.0	124312.0	8928.0	20764.0	6.700690	15.583909
3	용산구	244203.0	229456.0	14747.0	36231.0	6.038828	14.836427
4	성동구	311244.0	303380.0	7864.0	39997.0	2.526635	12.850689
5	광진구	372164.0	357211.0	14953.0	42214.0	4.017852	11.342849

➤ 외국인 비율, 고령자 비율 계산

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

	구별	인구수	한국인	외국인	고령자	외국인비율	고령자비율
1	종로구	162820.0	153589.0	9231.0	25425.0	5.669451	15.615404
2	중구	133240.0	124312.0	8928.0	20764.0	6.700690	15.583909
3	용산구	244203.0	229456.0	14747.0	36231.0	6.038828	14.836427
4	성동구	311244.0	303380.0	7864.0	39997.0	2.526635	12.850689
5	광진구	372164.0	357211.0	14953.0	42214.0	4.017852	11.342849

➤ 인구수, 외국인, 외국인 비율 기준을 내림차순 배열해보기

```
pop_Seoul.sort_values(by='인구수', ascending=False).head(5)
```

	구별	인구수	한국인	외국인	고령자	외국인비율	고령자비율
24	송파구	667483.0	660584.0	6899.0	72506.0	1.033584	10.862599
16	강서구	603772.0	597248.0	6524.0	72548.0	1.080540	12.015794
23	강남구	570500.0	565550.0	4950.0	63167.0	0.867660	11.072217
11	노원구	569384.0	565565.0	3819.0	71941.0	0.670725	12.634883
21	관악구	525515.0	507203.0	18312.0	68082.0	3.484582	12.955291

```
pop_Seoul.sort_values(by='외국인', ascending=False).head(5)
```

	구별	인구수	한국인	외국인	고령자	외국인비율	고령자비율
19	영등포구	402985.0	368072.0	34913.0	52413.0	8.663598	13.006191
17	구로구	447874.0	416487.0	31387.0	56833.0	7.007998	12.689506
18	금천구	255082.0	236353.0	18729.0	32970.0	7.342345	12.925255
21	관악구	525515.0	507203.0	18312.0	68082.0	3.484582	12.955291
6	동대문구	369496.0	354079.0	15417.0	54173.0	4.172440	14.661322

```
pop_Seoul.sort_values(by='외국인비율', ascending=False).head(5)
```

	구별	인구수	한국인	외국인	고령자	외국인비율	고령자비율
19	영등포구	402985.0	368072.0	34913.0	52413.0	8.663598	13.006191
18	금천구	255082.0	236353.0	18729.0	32970.0	7.342345	12.925255
17	구로구	447874.0	416487.0	31387.0	56833.0	7.007998	12.689506
2	중구	133240.0	124312.0	8928.0	20764.0	6.700690	15.583909
3	용산구	244203.0	229456.0	14747.0	36231.0	6.038828	14.836427

➤ 고령자, 고령자비율 기준을 내림차순 배열해보기

```
pop_Seoul.sort_values(by='고령자', ascending=False).head(5)
```

	구별	인구수	한국인	외국인	고령자	외국인비율	고령자비율
16	강서구	603772.0	597248.0	6524.0	72548.0	1.080540	12.015794
24	송파구	667483.0	660584.0	6899.0	72506.0	1.033584	10.862599
12	은평구	494388.0	489943.0	4445.0	72334.0	0.899091	14.631019
11	노원구	569384.0	565565.0	3819.0	71941.0	0.670725	12.634883
21	관악구	525515.0	507203.0	18312.0	68082.0	3.484582	12.955291

```
pop_Seoul.sort_values(by='고령자비율', ascending=False).head(5)
```

	구별	인구수	한국인	외국인	고령자	외국인비율	고령자비율
9	강북구	330192.0	326686.0	3506.0	54813.0	1.061806	16.600342
1	종로구	162820.0	153589.0	9231.0	25425.0	5.669451	15.615404
2	중구	133240.0	124312.0	8928.0	20764.0	6.700690	15.583909
3	용산구	244203.0	229456.0	14747.0	36231.0	6.038828	14.836427
13	서대문구	327163.0	314982.0	12181.0	48161.0	3.723221	14.720797

* 인구 대비 CCTV 현황 같은
내용을 확인하려면 두 데이터를
병합해야 함.

3. 파일 병합 및 시각화

...

➤ CCTV 현황 인구수 그래프 합치기

- 그래프 merge 하기
- 필요없는 부분 del 하기
- index(기준) 설정

➤ CCTV 현황 그래프로 분석하기 (시각화)

- matplotlib import , 폰트 설정 하기
- 1) 구별 CCTV barh 그래프
- 2) 인구수 별 CCTV 비율 barh 그래프
- 3) 인구수 별 CCTV 비율 dashed 그래프

➤ 표 merge 하기

```
In [87]: data_result = pd.merge(CCTV_Seoul, pop_Seoul, on='구별')  
data_result.head()
```

7]:

	구 별	소재지	2013 년도 이전	2014 년	2015 년	2016 년	최근증가율	인구수	한국인	외국인	고령자	외국인비 율	고령자비 율
0	강남구	2780	1292	430	584	932	150.619195	570500.0	565550.0	4950.0	63167.0	0.867660	11.072217
1	강동구	773	379	99	155	377	166.490765	453233.0	449019.0	4214.0	54622.0	0.929765	12.051638
2	강북구	748	369	120	138	204	125.203252	330192.0	326686.0	3506.0	54813.0	1.061806	16.600342
3	강서구	884	388	258	184	81	134.793814	603772.0	597248.0	6524.0	72548.0	1.080540	12.015794

➤ 표 del 하기, index 설정

```
In [88]: ▶ del data_result['2013년도 이전']  
del data_result['2014년']  
del data_result['2015년']  
del data_result['2016년']  
data_result.head()
```

Out [88]:

	구별	소계	최근증가율	인구수	한국인	외국인	고령자	외국인비율	고령자비율
0	강남구	2780	150.619195	570500.0	565550.0	4950.0	63167.0	0.867660	11.072217
1	강동구	773	166.490765	453233.0	449019.0	4214.0	54622.0	0.929765	12.051638
2	강북구	748	125.203252	330192.0	326686.0	3506.0	54813.0	1.061806	16.600342
3	강서구	884	134.793814	603772.0	597248.0	6524.0	72548.0	1.080540	12.015794
4	관악구	1496	149.290780	525515.0	507203.0	18312.0	68082.0	3.484582	12.955291

```
In [89]: ▶ data_result.set_index('구별', inplace=True)  
data_result.head()
```

➤ matplotlib import 하기

```
In [95]: ▶ import matplotlib.pyplot as plt  
%matplotlib inline
```

matplotlib.pyplot : 파이썬의 대표 시각화 도구,
matplotlib 중 그래프 그리는 모듈 .pyplot
%matplotlib inline : 결과 출력

➤ * 폰트 변경

matplotlib가 기본으로 한글을 지원하지 않기 때문에 폰트를 변경해야한다. 조건문으로 pc 운영체제 확인 후 폰트 변경

```
In [124]: ▶ import platform

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~~~~~')
```

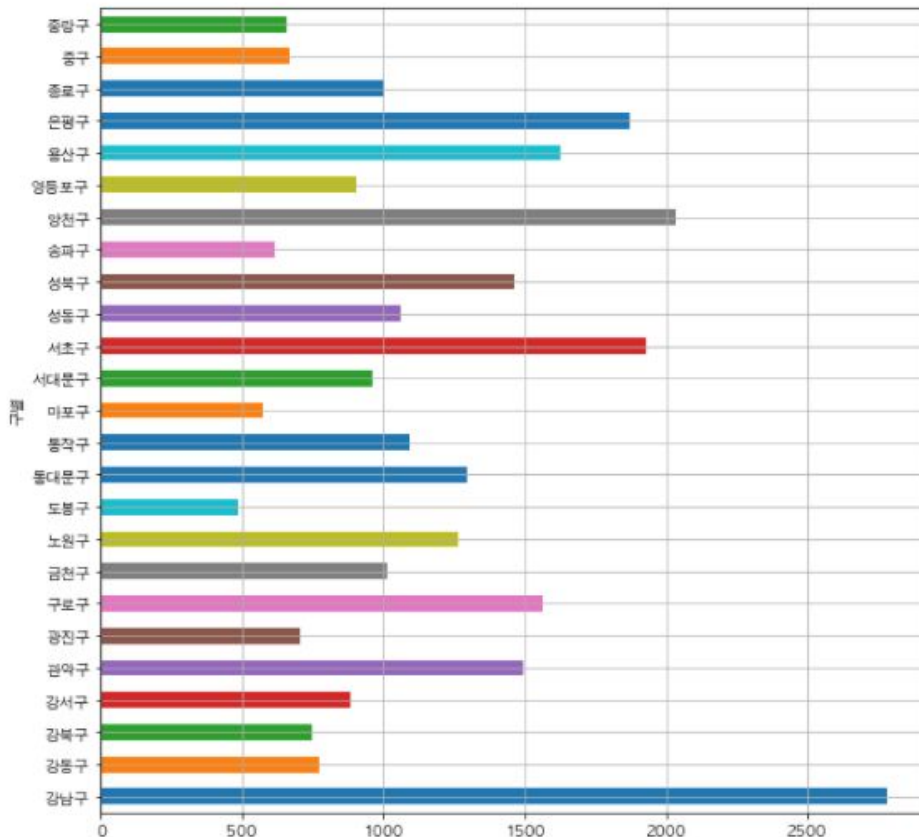

➤ 구별

CCTV barh

그래프

소계 값을 barh
형태로 grid
있게,
figsize는
가로:10,
세로:10 사이즈

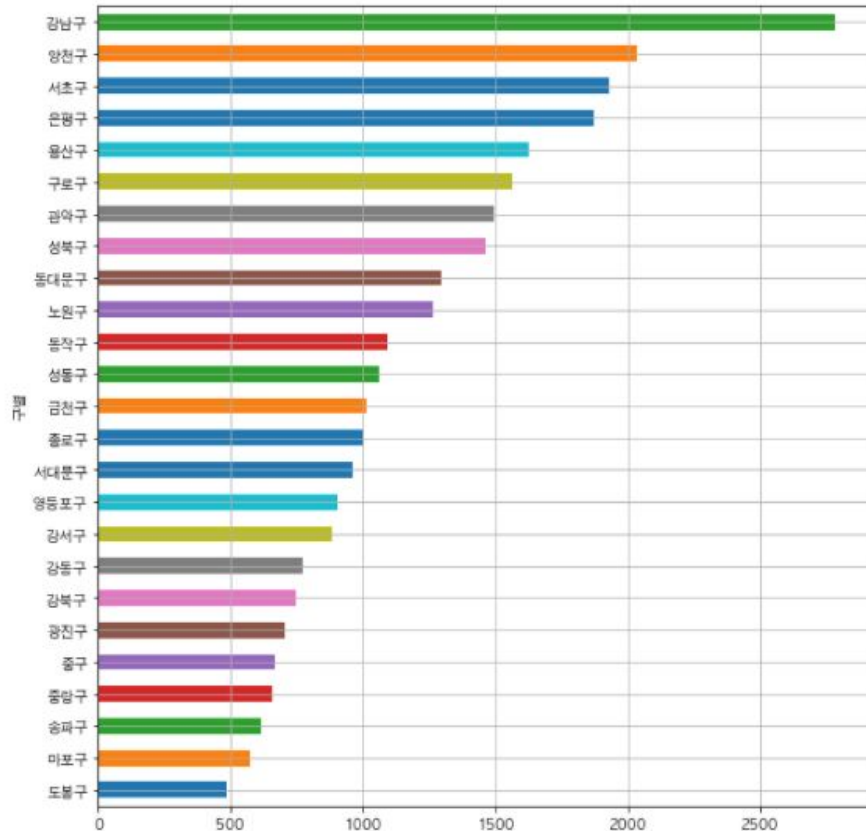
```
In [126]: plt.figure()  
data_result['소계'].plot(kind='barh', grid=True, figsize=(10,10))  
plt.show()
```



구별 CCTV barh 그래프 + 정렬

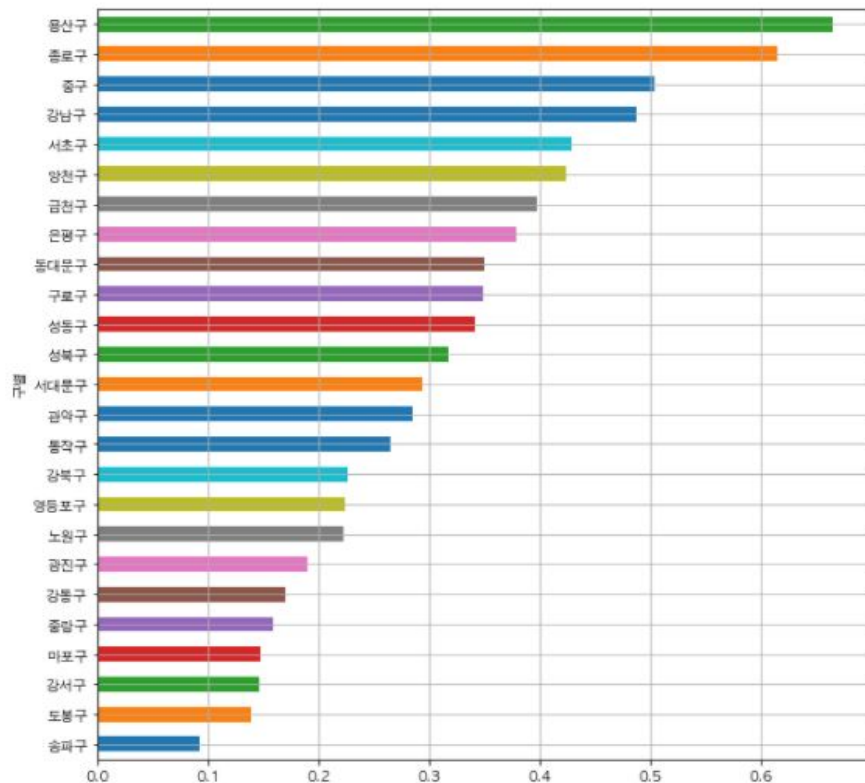
sort_values()
사용해서 정렬
-> 보기 편해짐

```
In [127]: data_result['소계'].sort_values() plot(kind='barh',  
grid=True, figsize=(10,10))  
plt.show()
```



➤ 인구 별 CCTV 비율 barh 그래프

```
In [128]: ▶ data_result['CCTV비율'] = data_result['소계'] / data_result['인구수'] * 100  
data_result['CCTV비율'].sort_values().plot(kind='barh',  
grid=True, figsize=(10,10))  
plt.show()
```



➤ 인구수 별 CCTV 비율 dashed 그래프

```
▶ plt.figure(figsize=(14,10))
plt.scatter(data_result['인구수'], data_result['소계'],
            c=data_result['오차'], s=50)
plt.plot(fx, f1(fx), ls='dashed', lw=3, color='g')

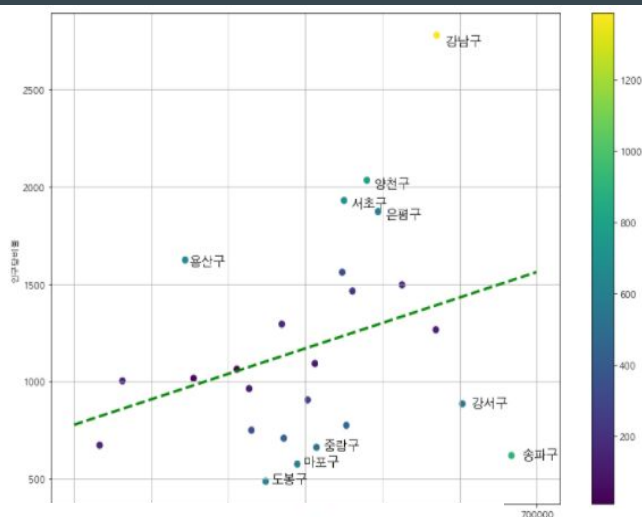
for n in range(10):
    plt.text(df_sort['인구수'][n]*1.02, df_sort['소계'][n]*0.98,
            df_sort.index[n], fontsize=15)

plt.xlabel('인구수')
plt.ylabel('인구당비율')
plt.colorbar()
plt.grid()
plt.show()
```

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

```
array([1.30916415e-03, 6.45066497e+02])
```

```
f1 = np.poly1d(fp1)
fx = np.linspace(100000, 700000, 100)
```



➤ 인구수 별
CCTV 비율
dashed 그래프

