

서울시 CCTV 현황 분석

<https://opengov.seoul.go.kr/data/2813904>

0. 한글폰트 설치 후 진행

1) 한글 폰트 설치

- 설치 후 '런타임 다시 시작'

```
!apt-get update
!apt-get install -y fonts-nanum
!fc-cache -fv
!rm ~/.cache/matplotlib -rf

Get:1 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu bionic InRelease [15.9 kB]
Hit:2 http://archive.ubuntu.com/ubuntu bionic InRelease
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 https://cloud.r-project.org/bin/linux/ubuntu bionic-cran40/ InRelease [3,626 B]
Hit:5 http://ppa.launchpad.net/cran/libgit2/ubuntu bionic InRelease
Get:6 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Hit:7 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu bionic InRelease
Hit:8 http://ppa.launchpad.net/graphics-drivers/ppa/ubuntu bionic InRelease
Get:9 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Ign:10 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64 InRelease
Ign:11 https://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64 InRelease
Hit:12 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64 Release
Hit:13 https://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64 Release
Get:14 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu bionic/main Sources [1,737 kB]
Get:15 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu bionic/main amd64 Packages [889 kB]
Get:16 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [2,395 kB]
Get:17 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [2,163 kB]
Get:18 http://archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [334 kB]
Get:21 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [1,929 kB]
Get:22 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [1,391 kB]
Fetched 11.1 MB in 3s (3,474 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  fonts-nanum
0 upgraded, 1 newly installed, 0 to remove and 29 not upgraded.
Need to get 9,604 kB of archives.
After this operation, 29.5 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic/universe amd64 fonts-nanum all 20170925-1 [9,604 kB]
Fetched 9,604 kB in 0s (44.3 MB/s)
Selecting previously unselected package fonts-nanum.
(Reading database ... 149414 files and directories currently installed.)
Preparing to unpack .../fonts-nanum_20170925-1_all.deb ...
Unpacking fonts-nanum (20170925-1) ...
Setting up fonts-nanum (20170925-1) ...
Processing triggers for fontconfig (2.12.6-0ubuntu2) ...
/usr/share/fonts: caching, new cache contents: 0 fonts, 1 dirs
/usr/share/fonts/truetype: caching, new cache contents: 0 fonts, 3 dirs
/usr/share/fonts/truetype/humor-sans: caching, new cache contents: 1 fonts, 0 dirs
/usr/share/fonts/truetype/liberation: caching, new cache contents: 16 fonts, 0 dirs
/usr/share/fonts/truetype/nanum: caching, new cache contents: 10 fonts, 0 dirs
/usr/local/share/fonts: caching, new cache contents: 0 fonts, 0 dirs
/root/.local/share/fonts: skipping, no such directory
/root/.fonts: skipping, no such directory
/var/cache/fontconfig: cleaning cache directory
/root/.cache/fontconfig: not cleaning non-existent cache directory
/root/.fontconfig: not cleaning non-existent cache directory
fc-cache: succeeded
```

2) 한글 폰트 사용

```
import matplotlib as mpl
import matplotlib.pyplot as plt

mpl.rcParams['axes.unicode_minus'] = False
```

```
plt.rc('font', family='NanumBarunGothic')
```

▼ I. Colab 파일 업로드

```
import warnings
warnings.filterwarnings('ignore')
```

▼ 1) 파일 업로드 확인

- seoulCCTV.csv
- seoulPopulation.xls

```
!!s -l
```

▼ II. 'seoulCCTV.csv' 파일 전처리

▼ 1) 'seoulCCTV.csv' 파일 읽어오기

```
import pandas as pd

SDF = pd.read_csv('seoulCCTV.csv', encoding = 'utf-8')

SDF.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

▼ 2) '기관명'을 '구별'로 변경

- 열(Column) 이름 확인

```
SDF.columns

Index(['기관명', '소계', '2013년도 이전', '2014년', '2015년', '2016년'], dtype='object')
```

- 첫번째 열 이름 변경('기관명' -> '구별')

```
SDF.rename(columns = {SDF.columns[0] : '구별'}, inplace = True)

SDF.head()
```

	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

3) CCTV 데이터 현황 파악

- '소계' 오름차순 정렬

```
SDF.sort_values(by = '소계', ascending = True).head(7)
```

	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
5	광진구	707	573	78	53	174
2	강북구	748	369	120	138	204

- '소계' 내림차순 정렬

```
SDF.sort_values(by = '소계', ascending = False).head(7)
```

	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
6	구로구	1561	1142	173	246	323
4	관악구	1496	846	260	390	613

4) '최근증가율' 열(Column) 추가

- 최근 3년 CCTV '최근증가율' 열 추가

```
SDF['최근증가율'] = ((SDF['2016년'] + SDF['2015년'] + SDF['2014년']) / SDF['2013년도 이전']) * 100
```

- '최근증가율' 열로 내림차순 정렬

```
SDF.sort_values(by = '최근증가율', ascending = False).head(7)
```

	구별	소계	2013년도 이전	2014년	2015년	2016년	최근증가율
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
19	영등포구	904	495	214	195	373	157.979798
0	강남구	2780	1292	430	584	932	150.619195

III. 'seoulPopulation.xls' 파일 전처리

1) 'seoulPopulation.xls' 파일 읽어오기

```
import pandas as pd
```

```
SDFP = pd.read_excel('seoulPopulation.xls',
                    header = 2,
                    usecols = 'B, D, G, J, N')
# encoding = 'UTF-8')
```

```
SDFP.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

2) 열(Column) 이름 변경

```
SDFP.rename(columns={SDFP.columns[0] : '구별',
                    SDFP.columns[1] : '인구수',
                    SDFP.columns[2] : '한국인',
                    SDFP.columns[3] : '외국인',
                    SDFP.columns[4] : '고령자'}, inplace=True)
```

```
SDFP.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

3) 인구 데이터 현황 파악

- '합계' 행(Row) 삭제

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

```
SDFP.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

- '구별' 열의 unique 정보 확인

```
SDFP['구별'].unique()
```

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

- NaN 정보 확인

SDFP[SDFP['구별'].isnull()]

	구별	인구수	한국인	외국인	고령자
26	NaN	NaN	NaN	NaN	NaN

SDFP.tail()

	구별	인구수	한국인	외국인	고령자
22	서초구	450310.0	445994.0	4316.0	51733.0
23	강남구	570500.0	565550.0	4950.0	63167.0
24	송파구	667483.0	660584.0	6899.0	72506.0
25	강동구	453233.0	449019.0	4214.0	54622.0
26	NaN	NaN	NaN	NaN	NaN

- NaN 행 삭제

SDFP.drop([26], inplace = True)

SDFP.tail()

	구별	인구수	한국인	외국인	고령자
21	관악구	525515.0	507203.0	18312.0	68082.0
22	서초구	450310.0	445994.0	4316.0	51733.0
23	강남구	570500.0	565550.0	4950.0	63167.0
24	송파구	667483.0	660584.0	6899.0	72506.0
25	강동구	453233.0	449019.0	4214.0	54622.0

- '외국인비율'과 '고령자비율' 열(Column)을 계산 후 추가

SDFP['외국인비율'] = SDFP['외국인'] / SDFP['인구수'] * 100
SDFP['고령자비율'] = SDFP['고령자'] / SDFP['인구수'] * 100

SDFP.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' 내림차순 정렬

SDFP.sort_values(by = '인구수', ascending = False).head(7)

	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
12	은평구	494388.0	489943.0	4445.0	72334.0	0.899091	14.631019
15	양천구	479978.0	475949.0	4029.0	52975.0	0.839413	11.036964

- '외국인' 내림차순 정렬

```
SDFP.sort_values(by = '외국인', ascending = False).head(7)
```

	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
5	광진구	372164.0	357211.0	14953.0	42214.0	4.017852	11.342849
3	용산구	244203.0	229456.0	14747.0	36231.0	6.038828	14.836427

- '외국인비율' 내림차순 정렬

```
SDFP.sort_values(by = '외국인비율', ascending = False).head(7)
```

	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
1	종로구	162820.0	153589.0	9231.0	25425.0	5.669451	15.615404
6	동대문구	369496.0	354079.0	15417.0	54173.0	4.172440	14.661322

- '고령자' 내림차순 정렬

```
SDFP.sort_values(by = '고령자', ascending = False).head(7)
```

	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
8	성북구	461260.0	449773.0	11487.0	64692.0	2.490353	14.025062
23	강남구	570500.0	565550.0	4950.0	63167.0	0.867660	11.072217

- '고령자비율' 내림차순 정렬

```
SDFP.sort_values(by = '고령자비율', ascending = False).head(7)
```

	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
10	도봉구	348646.0	346629.0	2017.0	51312.0	0.578524	14.717507
6	동대문구	369496.0	354079.0	15417.0	54173.0	4.172440	14.661322

▾ IV. CCTV 데이터와 인구 데이터 합치기

▾ 1) 두 데이터프레임에 공통으로 있는 '구별'로 merge

```
DF = pd.merge(SDF, SDFP, on = '구별')

DF.head()
```

	구별	소계	2013년도 이전	2014년	2015년	2016년	최근증가율	인구수	한국인	외국인	고령자	외국인비율	고령자비율
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
4	관악구	1496	846	260	390	613	149.290780	525515.0	507203.0	18312.0	68082.0	3.484582	12.955291

▾ 2) 불필요한 열(Column) 삭제

```
del DF['2013년도 이전']
del DF['2014년']
del DF['2015년']
del DF['2016년']

DF.head()
```

	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

▾ 3) '구별'을 index로 지정

```
DF.set_index('구별', inplace = True)

DF.head()
```

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

▾ 4) 상관계수(correlation coefficient)

- 범위 : -1 ~ 1(0이면 관계없음)
- '고령자비율' vs. '소계'

```
import numpy as np
```

```
print(np.corrcoef(DF['고령자비율'], DF['소계']))
```

```
[[ 1.          -0.28078554]
 [-0.28078554  1.          ]]
```

- '외국인비율' vs. '소계'

```
print(np.corrcoef(DF['외국인비율'], DF['소계']))
```

```
[[ 1.          -0.13607433]
 [-0.13607433  1.          ]]
```

- '인구수' vs. '소계'

```
print(np.corrcoef(DF['인구수'], DF['소계']))
```

```
[[1.          0.30634228]
 [0.30634228  1.          ]]
```

▼ 5) CCTV 개수('소계')와 '인구수'의 관계

- '소계'(CCTV 개수) 내림차순 정렬

```
DF.sort_values(by = '소계', ascending = False).head()
```

	소계	최근증가율	인구수	한국인	외국인	고령자	외국인비율	고령자비율
구별								
강남구	2780	150.619195	570500.0	565550.0	4950.0	63167.0	0.867660	11.072217
양천구	2034	34.671731	479978.0	475949.0	4029.0	52975.0	0.839413	11.036964
서초구	1930	63.371266	450310.0	445994.0	4316.0	51733.0	0.958451	11.488308
은평구	1873	85.237258	494388.0	489943.0	4445.0	72334.0	0.899091	14.631019
용산구	1624	53.216374	244203.0	229456.0	14747.0	36231.0	6.038828	14.836427

- '인구수' 내림차순 정렬

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

	소계	최근증가율	인구수	한국인	외국인	고령자	외국인비율	고령자비율
구별								
송파구	618	104.347826	667483.0	660584.0	6899.0	72506.0	1.033584	10.862599
강서구	884	134.793814	603772.0	597248.0	6524.0	72548.0	1.080540	12.015794
강남구	2780	150.619195	570500.0	565550.0	4950.0	63167.0	0.867660	11.072217
노원구	1265	188.929889	569384.0	565565.0	3819.0	71941.0	0.670725	12.634883
관악구	1496	149.290780	525515.0	507203.0	18312.0	68082.0	3.484582	12.955291

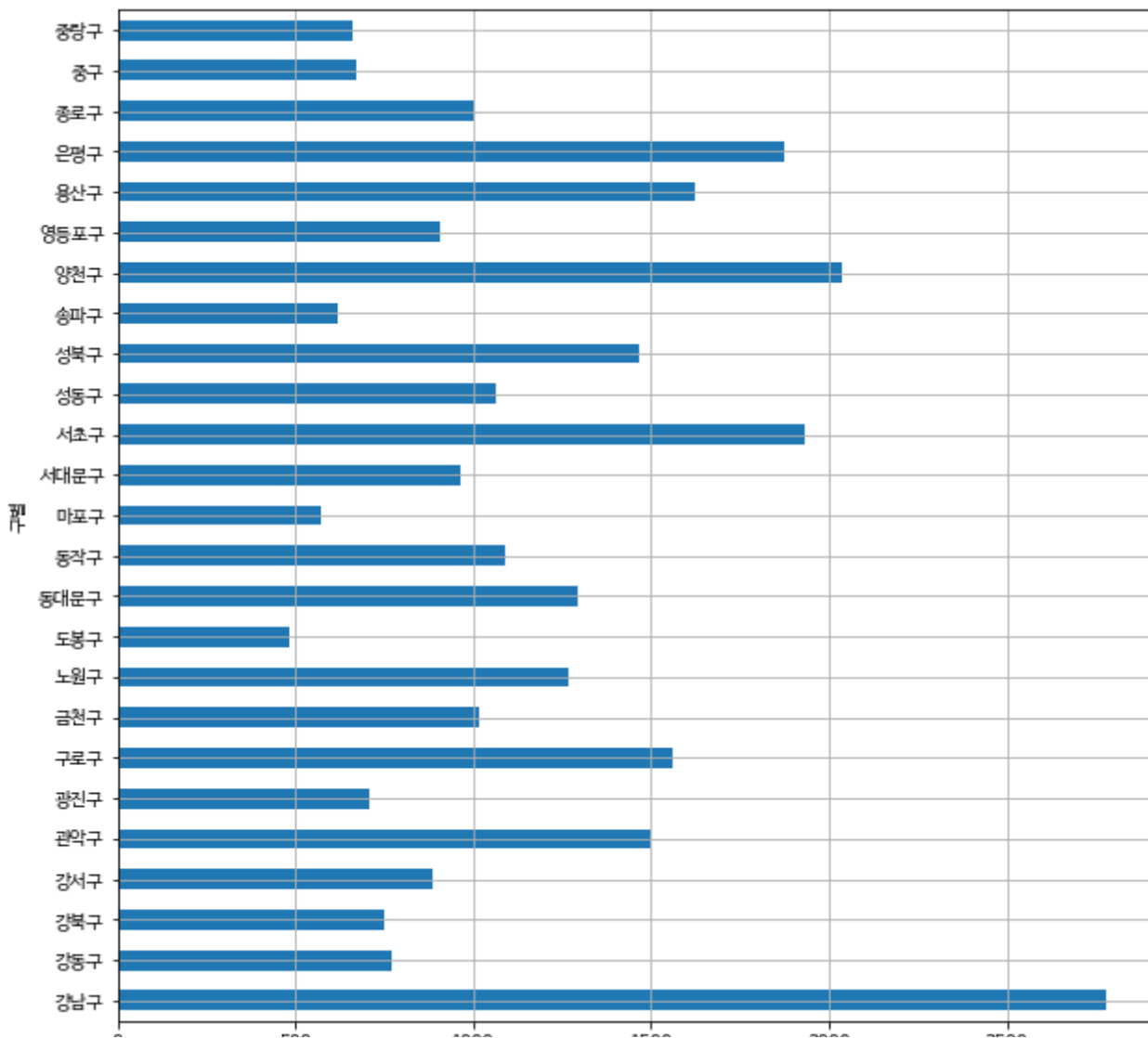
▼ V. Visualization

▼ 1) 막대(bar) 그래프 - pandas

- '소계'

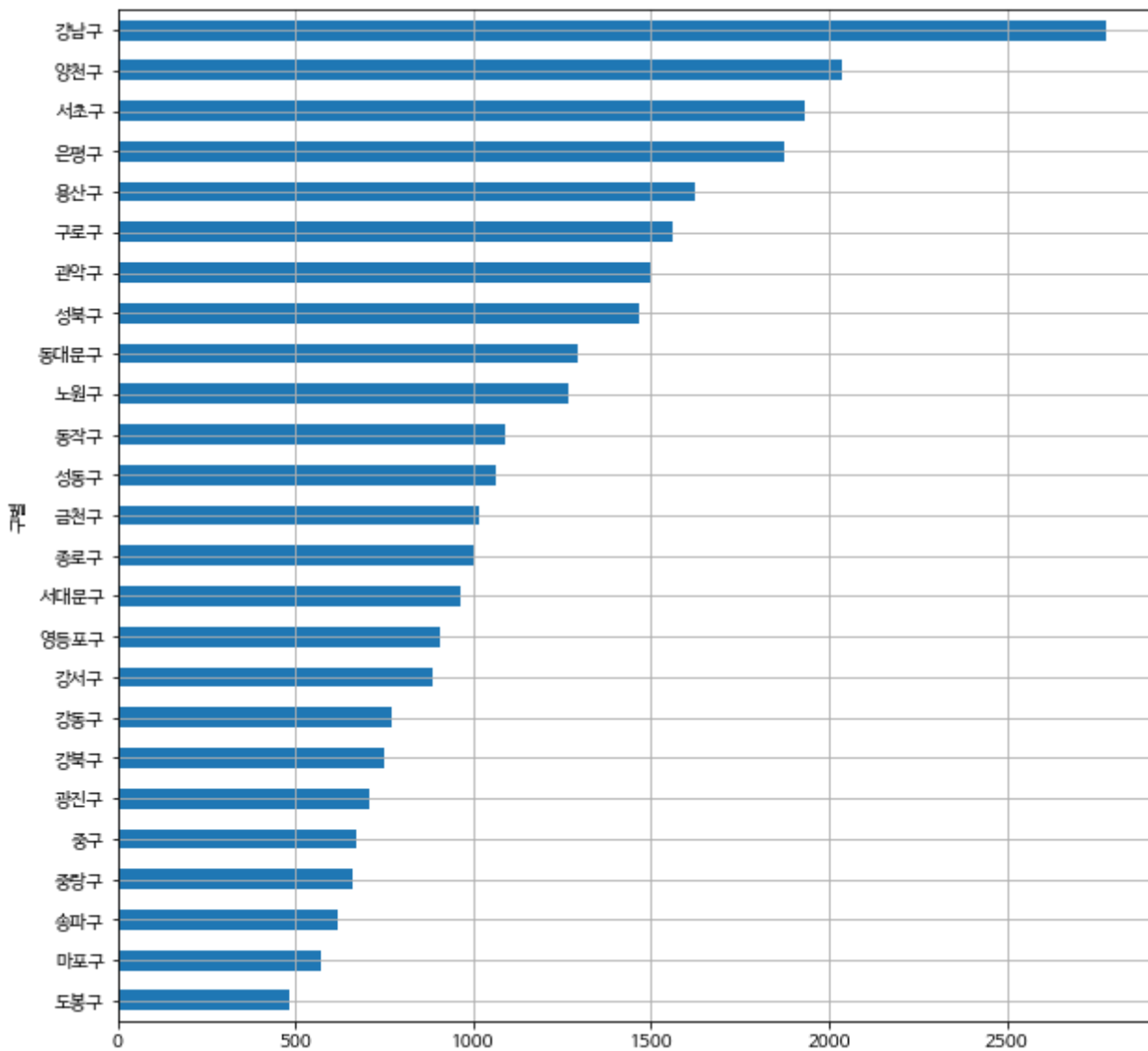
```
import matplotlib.pyplot as plt

DF['소계'].plot(kind = 'barh', grid = True, figsize = (10, 10))
plt.show()
```

- 정렬 - sort_value()

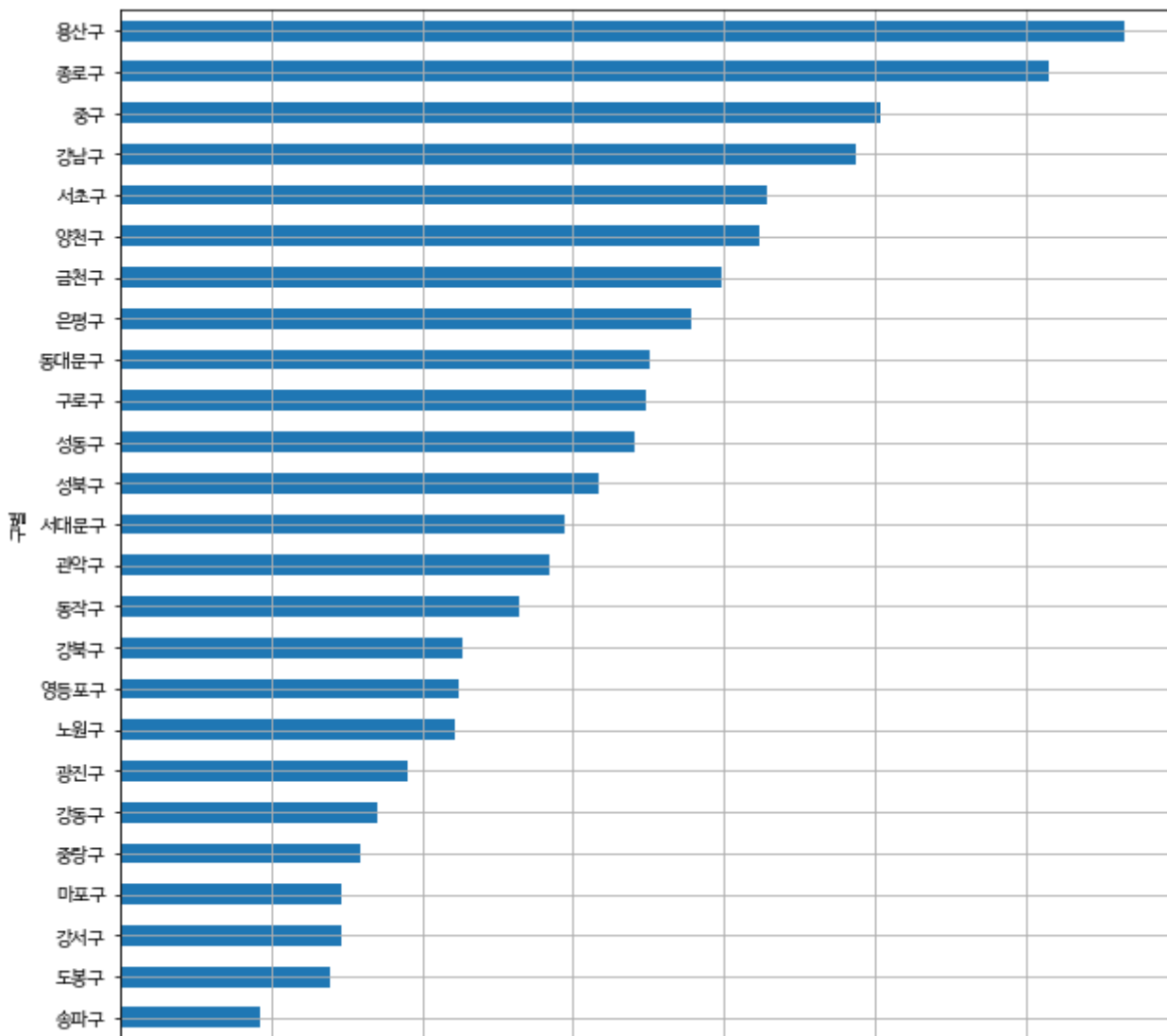
```
DF['소계'].sort_values().plot(kind = 'barh', grid = True, figsize = (10, 10))
plt.show()
```



- '인구수' 대비 CCTV비율 계산 후 정렬하여 시각화

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

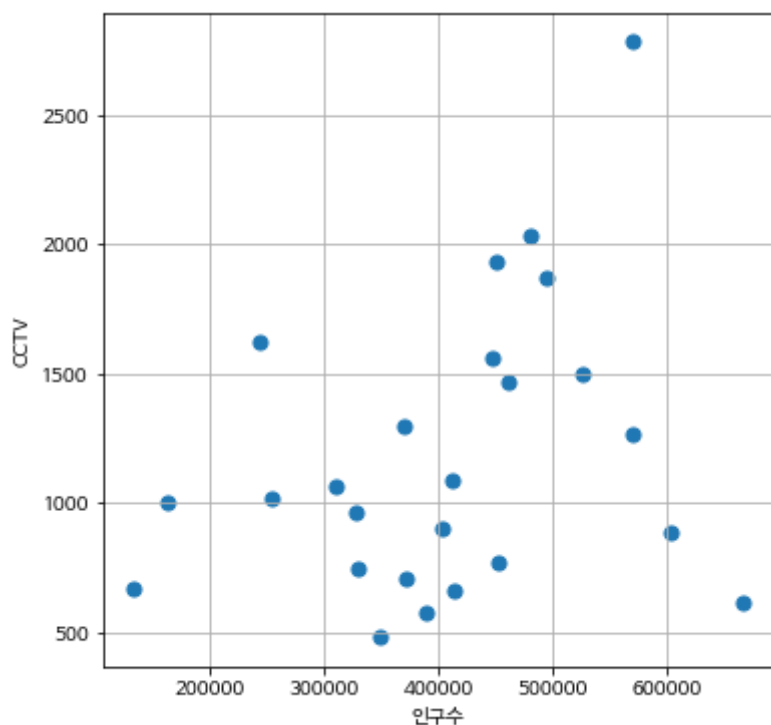
```
DF['CCTV비율'].sort_values().plot(kind = 'barh', grid = True, figsize = (10, 10))
plt.show()
```



▼ 2) 산점도(scatter) - matplotlib

- '소계' ~ '인구수'

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



▼ 3) 회귀계수 계산

- '소계' ~ '인구수'

```
import numpy as np
np.set_printoptions(suppress = True)

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

fp1

array([ 0.00130916, 645.06649658])
```

- x축, y축 데이터 계산

```
f1 = np.poly1d(fp1)
print(f1, 'Wn')

fx = np.linspace(100000, 700000, 100)
print(fx)

0.001309 x + 645.1

[100000.      106060.60606061 112121.21212121 118181.81818182
 124242.42424242 130303.03030303 136363.63636364 142424.24242424
 148484.84848485 154545.45454545 160606.06060606 166666.66666667
 172727.27272727 178787.87878788 184848.48484848 190909.09090909
 196969.6969697  203030.3030303  209090.90909091 215151.51515152
 221212.12121212 227272.72727273 233333.33333333 239393.93939394
 245454.54545455 251515.15151515 257575.75757576 263636.36363636
 269696.96969697 275757.57575758 281818.18181818 287878.78787879
 293939.39393939 300000.      306060.60606061 312121.21212121
 318181.81818182 324242.42424242 330303.03030303 336363.63636364
 342424.24242424 348484.84848485 354545.45454545 360606.06060606
 366666.66666667 372727.27272727 378787.87878788 384848.48484848
 390909.09090909 396969.6969697  403030.3030303  409090.90909091
 415151.51515152 421212.12121212 427272.72727273 433333.33333333
 439393.93939394 445454.54545455 451515.15151515 457575.75757576
 463636.36363636 469696.96969697 475757.57575758 481818.18181818
 487878.78787879 493939.39393939 500000.      506060.60606061
 512121.21212121 518181.81818182 524242.42424242 530303.03030303
 536363.63636364 542424.24242424 548484.84848485 554545.45454545
 560606.06060606 566666.66666667 572727.27272727 578787.87878788
 584848.48484848 590909.09090909 596969.6969697  603030.3030303
 609090.90909091 615151.51515152 621212.12121212 627272.72727273
 633333.33333333 639393.93939394 645454.54545455 651515.15151515
 657575.75757576 663636.36363636 669696.96969697 675757.57575758
 681818.18181818 687878.78787879 693939.39393939 700000.      ]
```

▾ 4) 회귀선 추가

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



5) '오차' 열(Column) 추가 후 시각화

- '오차' 열 계산 후 내림차순 정렬

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

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

DF['오차'] = np.abs(DF['소계'] - f1(DF['인구수']))

df_sort = DF.sort_values(by = '오차', ascending = False)

DF
```

구별	소계	최근증가율	인구수	한국인	외국인	고령자	외국인비율	고령자비율	CCTV비율	오차
강남구	2780	150.619195	570500.0	565550.0	4950.0	63167.0	0.867660	11.072217	0.487292	1388.055355
강동구	773	166.490765	453233.0	449019.0	4214.0	54622.0	0.929765	12.051638	0.170552	465.422892
강북구	748	125.203252	330192.0	326686.0	3506.0	54813.0	1.061806	16.600342	0.226535	329.342026
강서구	884	134.793814	603772.0	597248.0	6524.0	72548.0	1.080540	12.015794	0.146413	551.503155
관악구	1496	149.290780	525515.0	507203.0	18312.0	68082.0	3.484582	12.955291	0.284673	162.948104
광진구	707	53.228621	372164.0	357211.0	14953.0	42214.0	4.017852	11.342849	0.189970	425.290264
구로구	1561	64.973730	447874.0	416487.0	31387.0	56833.0	7.007998	12.689506	0.348536	329.592918
금천구	1015	100.000000	255082.0	236353.0	18729.0	32970.0	7.342345	12.925255	0.397911	35.989293
노원구	1265	188.929889	569384.0	565565.0	3819.0	71941.0	0.670725	12.634883	0.222170	125.483618
도봉구	485	246.638655	348646.0	346629.0	2017.0	51312.0	0.578524	14.717507	0.139110	616.501341
동대문구	1294	74.766355	369496.0	354079.0	15417.0	54173.0	4.172440	14.661322	0.350207	165.202586
동작구	1091	139.338235	412520.0	400456.0	12064.0	56013.0	2.924464	13.578251	0.264472	94.122892
마포구	574	212.101911	389649.0	378566.0	11083.0	48765.0	2.844355	12.515110	0.147312	581.180999
서대문구	962	48.578199	327163.0	314982.0	12181.0	48161.0	3.723221	14.720797	0.294043	111.376568
서초구	1930	63.371266	450310.0	445994.0	4316.0	51733.0	0.958451	11.488308	0.428594	695.403794
성동구	1062	81.780822	311244.0	303380.0	7864.0	39997.0	2.526635	12.850689	0.341211	9.464016
성북구	1464	63.627354	461260.0	449773.0	11487.0	64692.0	2.490353	14.025062	0.317391	215.068447
송파구	618	104.347826	667483.0	660584.0	6899.0	72506.0	1.033584	10.862599	0.092587	900.911312
양천구	2034	34.671731	479978.0	475949.0	4029.0	52975.0	0.839413	11.036964	0.423769	760.563512
영등포구	904	157.979798	402985.0	368072.0	34913.0	52413.0	8.663598	13.006191	0.224326	268.640012
용산구	1624	53.216374	244203.0	229456.0	14747.0	36231.0	6.038828	14.836427	0.665020	659.231690
은평구	1873	85.237258	494388.0	489943.0	4445.0	72334.0	0.899091	14.631019	0.378852	580.698457
종로구	1002	248.922414	162820.0	153589.0	9231.0	25425.0	5.669451	15.615404	0.615404	143.775396
중구	671	147.699758	133240.0	124312.0	8928.0	20764.0	6.700690	15.583909	0.503603	148.499528
중랑구	660	79.960707	414503.0	409882.0	4621.0	56774.0	1.114829	13.696885	0.159227	527.718965

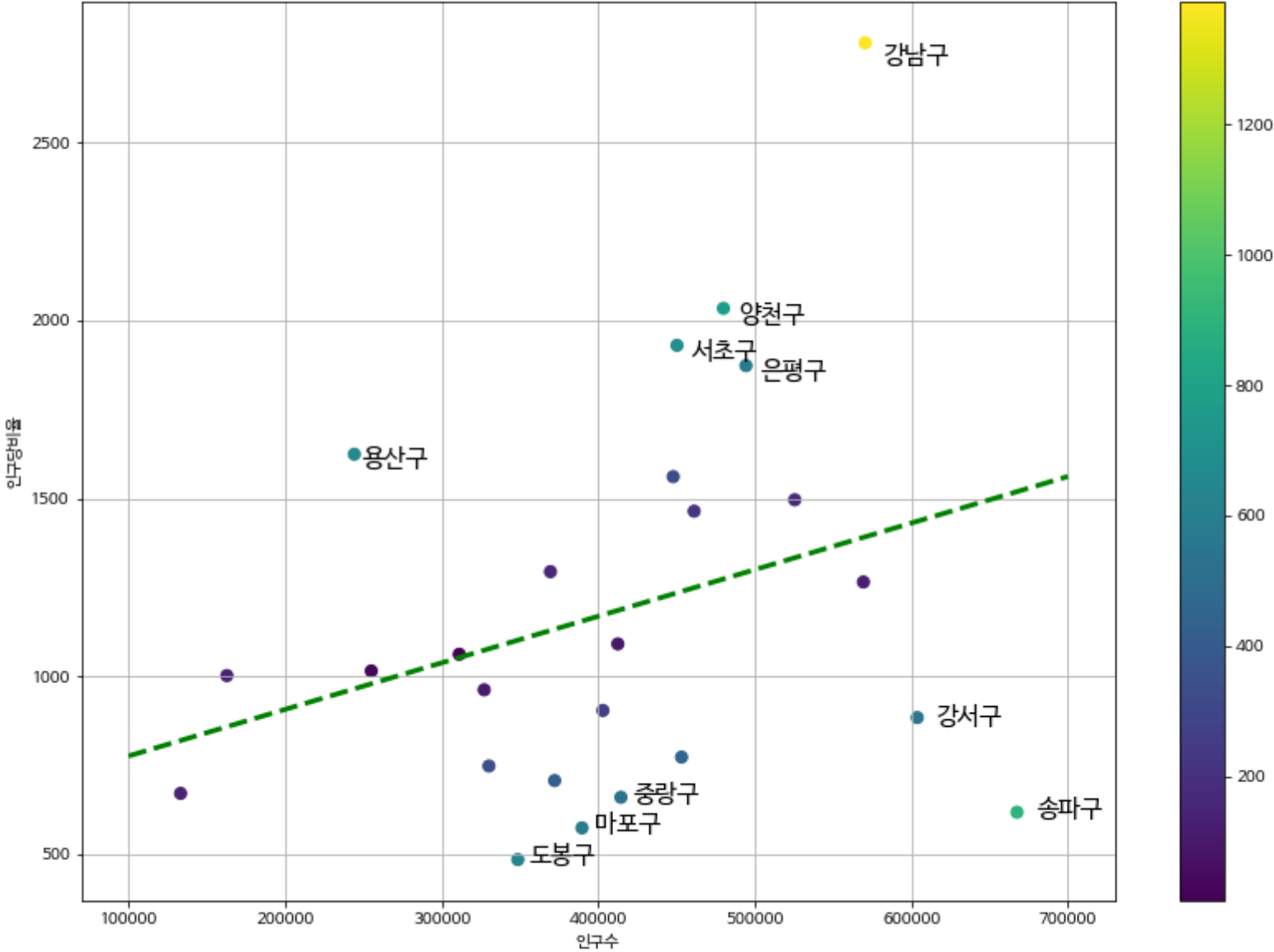
- 최종 시각화

```
plt.figure(figsize = (14, 10))
plt.scatter(DF['인구수'], DF['소계'], c = DF['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()
```



#

The End

#