

## 2조 소프트

홍태기, 김준석, 전지영, 최원석, 권은경

### 목차

1. 소년범죄 종류별 범행동기 비율
2. 지역별 범죄 발생 현황
3. 검거기간별 상위 5개 범죄종류 비율
4. 대분류(강력, 폭력, 지능) 범죄 별 피해자 성별 & 연령대 비율
5. 서울시 구별 CCTV 현황 분석하기

### 1. 소년범죄 종류별 범행동기 비율

#### 가. 개요

요즘 촉법소년 문제가 사회적으로 대두되고 있다.

이를 통계적 시각으로 접근해 촉법소년들이 저지른 범죄종류를 기준으로 주로 그들이 저지른 범죄종류 5가지를 많은순으로 알아보았다.

```
In [32]: import csv
import matplotlib.pyplot as plt
from matplotlib import font_manager, rc
font_path = "c:/Windows/Fonts/NGULIM.TTF"
font = font_manager.FontProperties(fname=font_path).get_name()
rc('font', family=font)

file = open('data/소년범죄자범행동기.csv', 'r')
data = csv.reader(file)
header = next(data)
header = next(data)
header = next(data)
header = next(data)

data_list = []
for row in data:
    data_list.append(row)

year = {}
cri = []
top5 = []

rate = {}
```

```

rs_list = []
rs_list1 = []
rs_rate = []
rs_rate1 = []
rs = ['생활비마련', '유형비마련', '도박비마련', '허영사치심', '치부', '기타(이욕)', '사행성']

for row in data_list:
    if row[2] not in year:
        year[row[2]] = [0,0,0,0]
    for i in range(0, 4):
        for j in range(3, 22):
            year[row[2]][i] += int(row[j+19])

total = {}
for key, value in year.items():
    if key not in total:
        total[key] = [0]
    for i in range(0, 4):
        total[key][0] += value[i]

for key, value in sorted(total.items(), key=lambda x: x[1], reverse=True):
    cri.append(key)
    top5.append(value[0])
    for row in data_list:
        if row[2] == key:
            if row[2] not in rate:
                rate[row[2]] = {}
            for i in range(3, 22):
                if rs[i-3] not in rate[row[2]]:
                    rate[row[2]][rs[i-3]] = [0]
                for j in range(0, 4):
                    rate[row[2]][rs[i-3]][0] += int(row[i+19*j])

    if len(top5) >= 5:
        break

for key1, value1 in rate.items():
    for key, value in sorted(value1.items(), key=lambda x: x[1], reverse=True):
        if key != "미상" and key != "기타":
            rs_list.append(key)
            rs_rate.append(value[0])

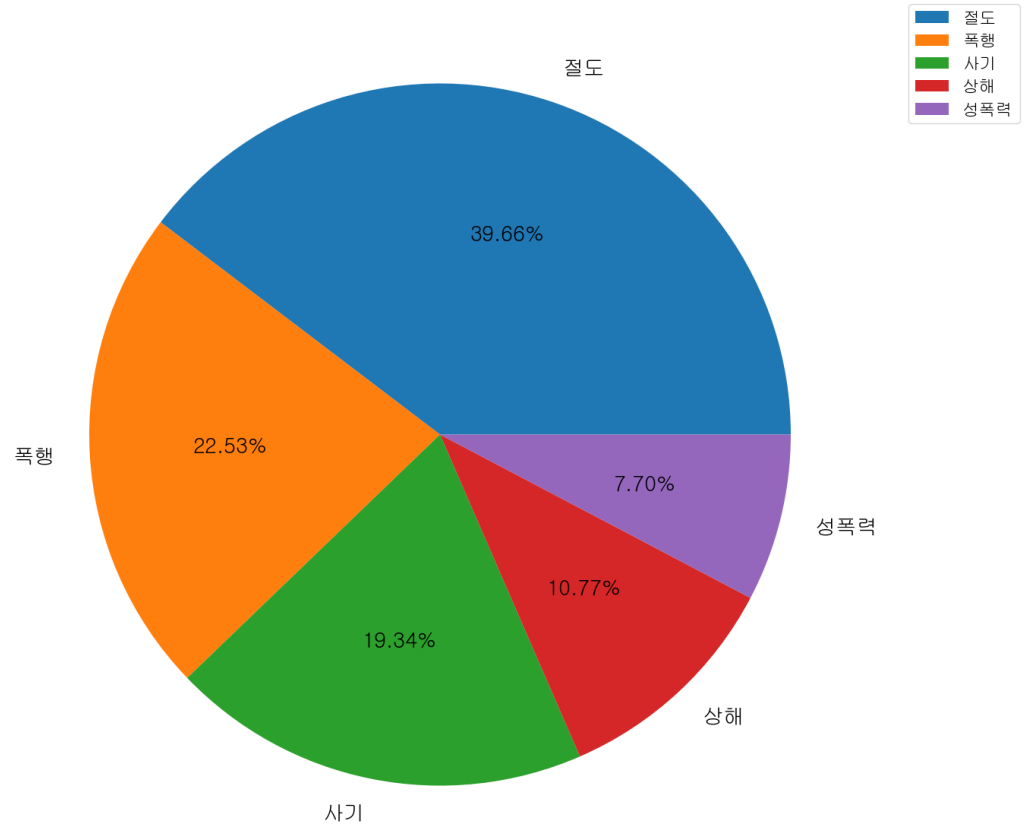
for i in range(5):
    temp = []
    temp1 = []
    for j in range(0, 8):
        temp.append(rs_rate[j+17*i])
        temp1.append(rs_list[j+17*i])
    rs_rate1.append(temp)
    rs_list1.append(temp1)

```

```

In [19]: plt.figure(figsize=(20,15))
plt.axis('equal')
plt.title('소년 범죄율이 높은 상위 5개 종류', fontsize=25)
plt.pie(top5, labels = cri, autopct='%.2f%%', textprops={'fontsize': 20})
plt.legend(loc=1, fontsize = 16)
plt.show()

```



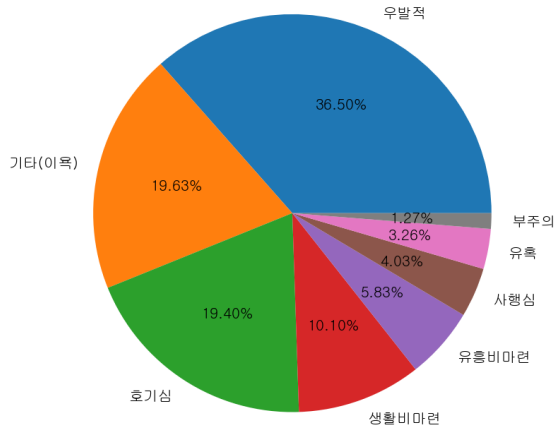
위 차트를 보면 그들의 범행은 주로 폭력과 관련되어있었으며,

많은 순으로 정렬해보았을때 절도, 폭행, 사기, 상해, 성폭력 순으로 범행을 저질렀다.

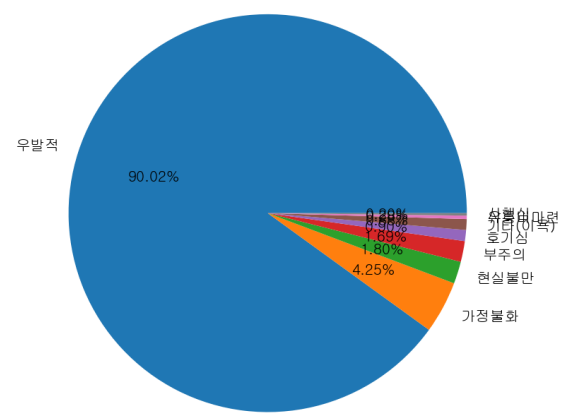
우리는 이 자료들을 기반으로 이들이 어떠한 이유로 범행을 저질렀는지 범행 동기를 조사해 보았다.

```
In [27]: cnt = 1
plt.figure(figsize=(20,30))
for i in range(5):
    plt.subplot(3,2,cnt)
    plt.axis('equal')
    plt.title(cri[i]+' 범죄 범행 동기 비율', fontsize=25)
    plt.pie(rs_rate1[i], labels = rs_list1[i], autopct='%.2f%%', textprops={'fontsize': 12})
    cnt += 1
plt.show()
```

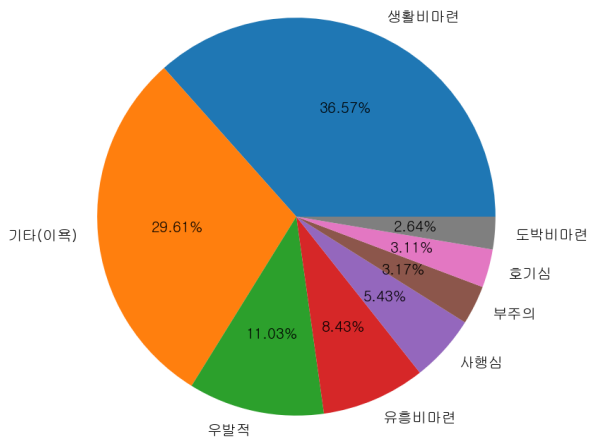
절도 범죄 범행동기 비율



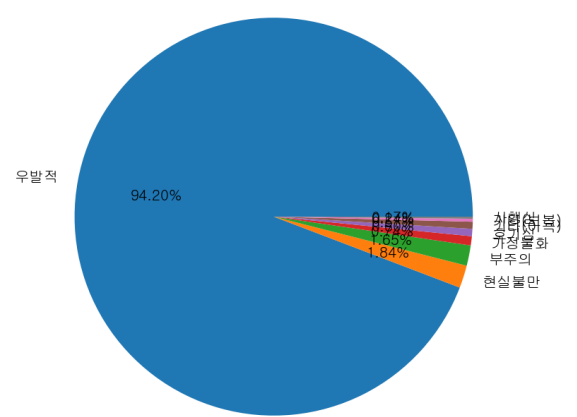
폭행 범죄 범행동기 비율



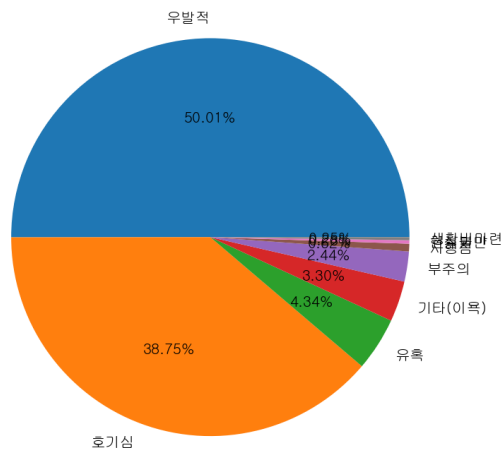
사기 범죄 범행동기 비율



상해 범죄 범행동기 비율



성폭력 범죄 범행동기 비율



## 나. 결론

측법소년들의 주된 범죄 범행동기는 평균 '약 56.35%'로 우발적인 범행동기를 가졌으며,

최대 '94.2%' 우발적 범행동기, 최저 '0.25%' 생활비마련의 이유로 범행을 저지른것으로 보인다.

이를 통해 우리는 측법소년들이 대체적으로 우발적으로 범행을 많이 저지르는것 같다.

앞으로 촉법소년과 문제가 생긴다면 맞서기보단 피하는게 좋을거 같다.\

## 다. 출처

'국가통계포털 - 소년범죄자 범행동기'

[https://kosis.kr/statHtml/statHtml.do?orgId=135&tblId=TX\\_13501\\_A150&conn\\_path=I2](https://kosis.kr/statHtml/statHtml.do?orgId=135&tblId=TX_13501_A150&conn_path=I2)

## 2. 지역별 범죄 발생 현황

### 가. 개요

최근 각종 범죄(살인, 강도, 강간, 절도, 폭력 등)의 발생은 장기적인 경기침체와 불황으로 꾸준히 증가하고 있습니다.

지금까지 분석된 다양한 범죄의 발생특성을 살펴보면, 그 발생지가 치안환경에 열악한 우범지역이라는 공통점이 있어,

범죄발생이 지역에 따라 얼마만큼 비율의 차이가 있는지 통계자료를 통해 우범지역 발생 이유를 유추해 볼수가 있었습니다.

```
In [2]: import csv
import matplotlib.pyplot as plt
from matplotlib import font_manager, rc

font_path = "C:/Windows/Fonts/NGULIM.TTF"
font = font_manager.FontProperties(fname=font_path).get_name()
rc('font', family=font)

file = open('data/crime_region_2019.csv', 'r')
data = csv.reader(file)
header = next(data)

data_list = []
for row in data:
    row[1:] = map(int, row[1:])
    data_list.append(row)

regions = ['서울', '부산', '대구', '인천', '광주', '대전', '울산', '세종', '경기', '

crime_dict = {}

for row in data_list:

    if row[0] not in crime_dict:
        crime_dict[row[0]] = {}
    for item in range(len(row)-1):
        if regions[item] not in crime_dict[row[0]]:
            crime_dict[row[0]][regions[item]] = row[item+1]

crime_sort = {}

for key, value in crime_dict.items():
    val = sorted(value.items(), key = lambda item:item[1], reverse = True)
```

```

if key not in crime_sort:
#     crime_sort[key] = val[:5]
    crime_sort[key] = val

```

## 나. 조사자료(통계 차트)

다음은 2019년도 전국을 대상으로 한 지역별 범죄 발생 빈도수를 조사한 자료를 기반으로 만든 차트입니다.

조사한 결과 전국 범죄의 연평균 발생수는 49631건이었으며, 평균 서울 '21.0%', 경기 '22.7%'로

주로 서울 경기 지역이 대부분 범죄 발생 비율 1, 2위를 나란히 했습니다.

하지만 그와는 상반되는 수치로 평균 '0.3%'로 제일 적은 범죄 발생 비율을 보인 세종시는 대부분의 범죄에서 적은 수치를 나타냈습니다.

```

In [3]: plt.rc('font', size=12)

for key, value in crime_sort.items():

    data = []
    labels = []

    for val in value:
        data.append(val[1])
        labels.append(val[0])

    plt.title(key + ' ' + '범죄 지역별 발생 비율', fontsize = 20)

    explodes = [0.1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.1]
    colors = ['red', 'firebrick', 'indianred', 'lightcoral', 'salmon', 'lightsalmon']
    wedgeprops = {'width': 0.75, 'edgecolor': 'w', 'linewidth': 1}

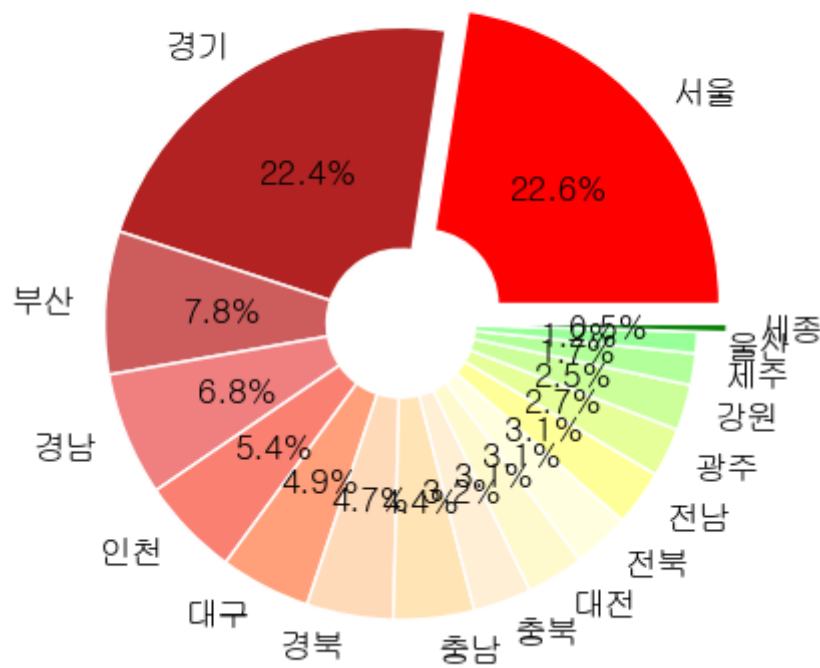
    plt.pie(data, labels = labels, autopct = '%.1f%%', explode = explodes, colors =

    plt.show()

    print('<{} 범죄>'.format(key))
    print('가장 많이 발생한 지역 : {}'.format(labels[0]))
    print('가장 적게 발생한 지역 : {}'.format(labels[16]))
    print()
    print()
    print()

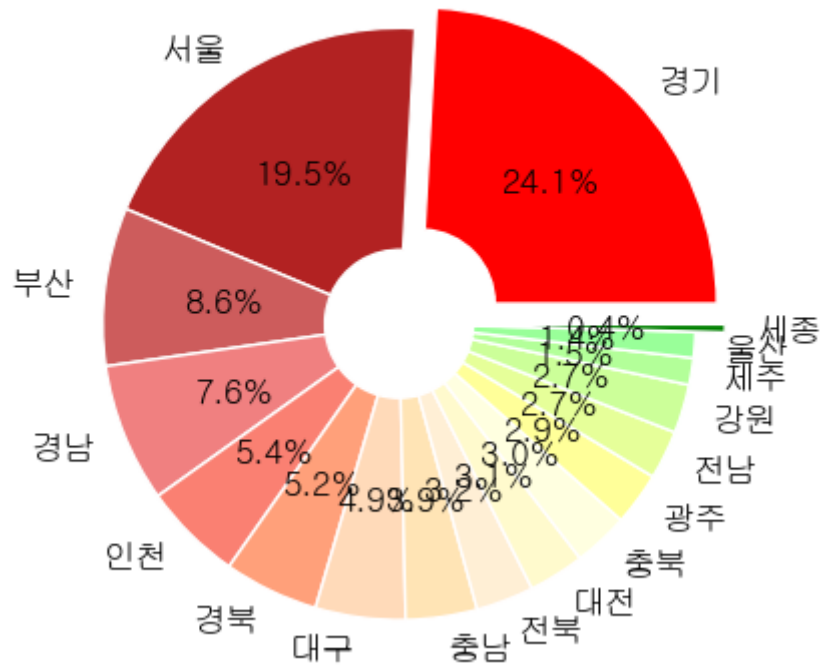
```

절도 범죄 지역별 발생 비율



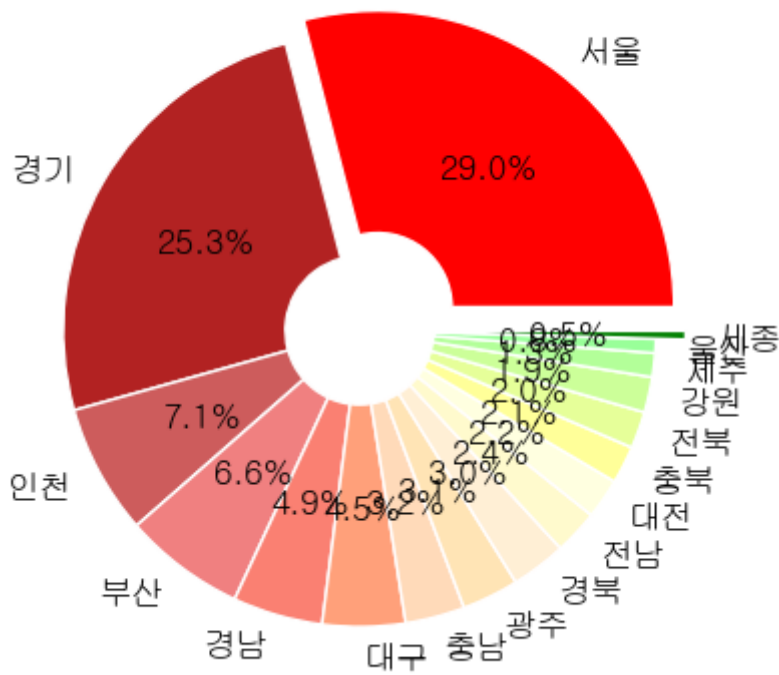
<절도 범죄>  
가장 많이 발생한 지역 : 서울  
가장 적게 발생한 지역 : 세종

사기 범죄 지역별 발생 비율



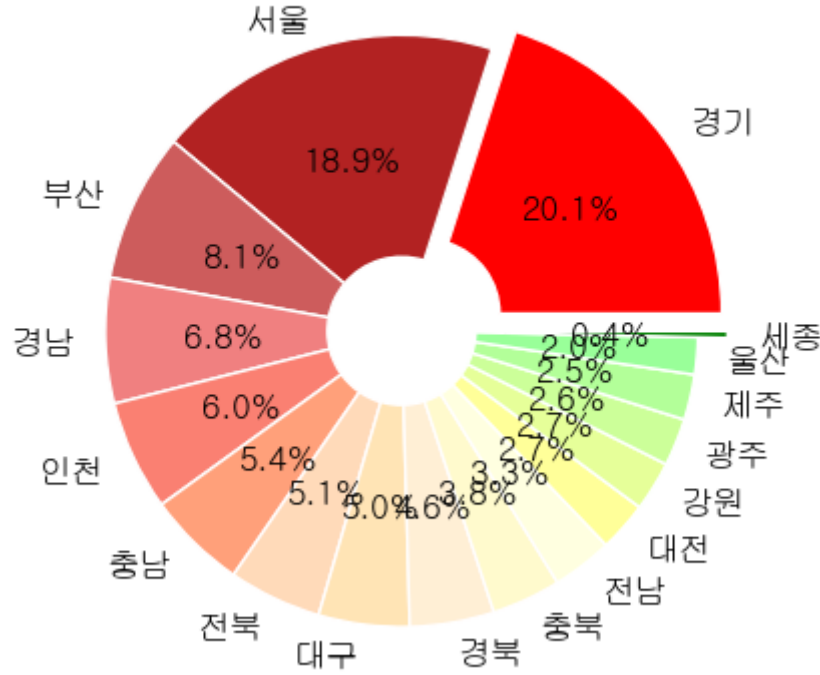
<사기 범죄>  
가장 많이 발생한 지역 : 경기  
가장 적게 발생한 지역 : 세종

횡령 범죄 지역별 발생 비율



<횡령 범죄>  
가장 많이 발생한 지역 : 서울  
가장 적게 발생한 지역 : 세종

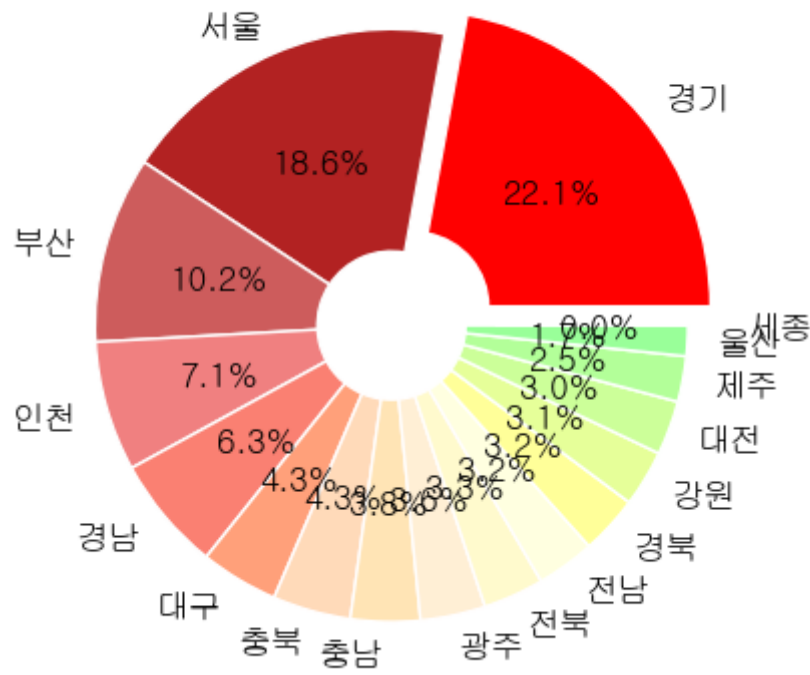
살인 범죄 지역별 발생 비율



<살인 범죄>  
가장 많이 발생한 지역 : 경기  
가장 적게 발생한 지역 : 세종

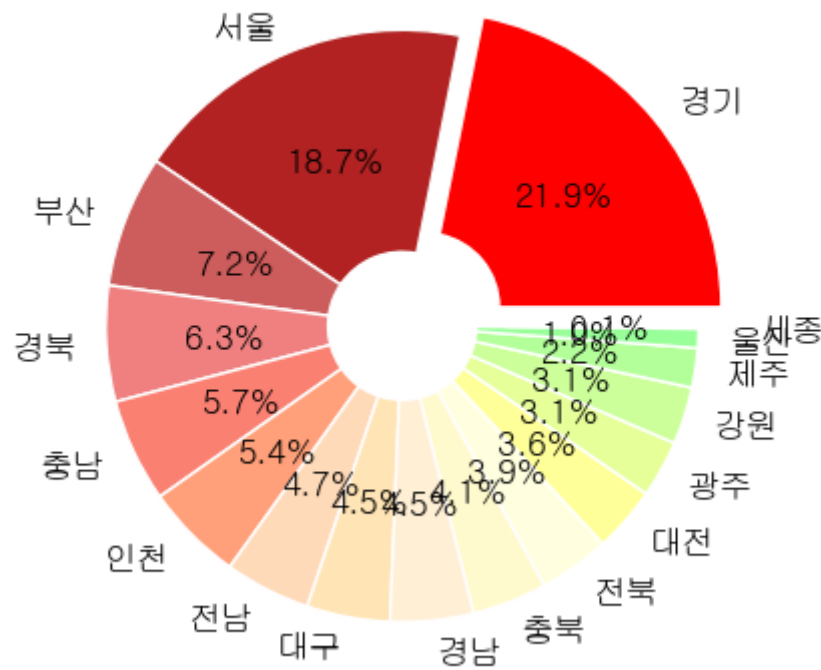


# 강도 범죄 지역별 발생 비율



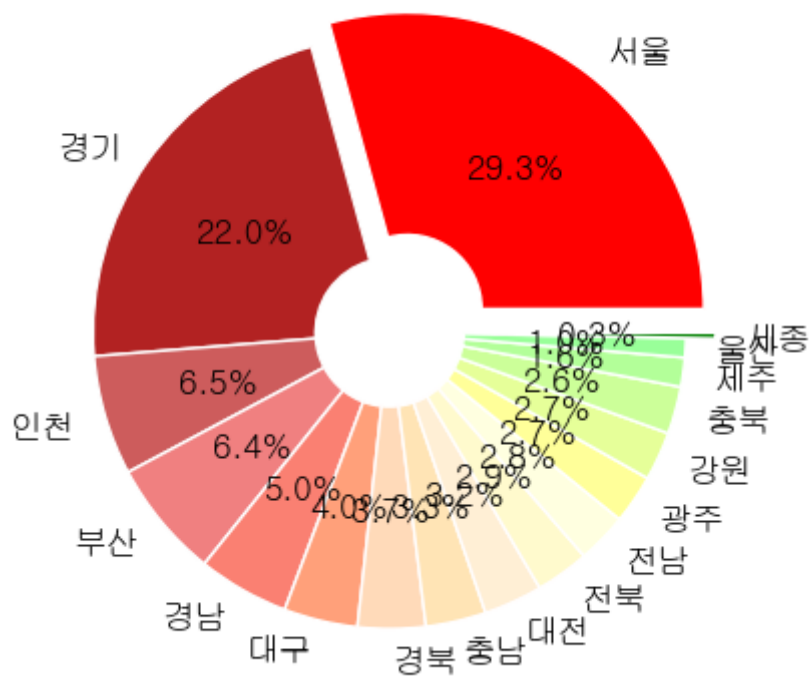
<강도 범죄>  
가장 많이 발생한 지역 : 경기  
가장 적게 발생한 지역 : 세종

# 방화 범죄 지역별 발생 비율



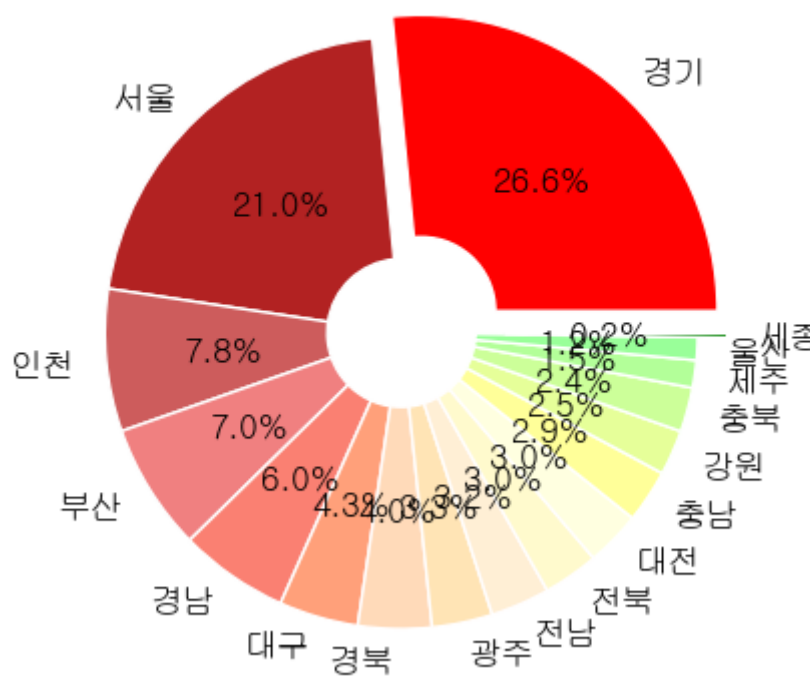
<방화 범죄>  
가장 많이 발생한 지역 : 경기  
가장 적게 발생한 지역 : 세종

성폭력 범죄 지역별 발생 비율



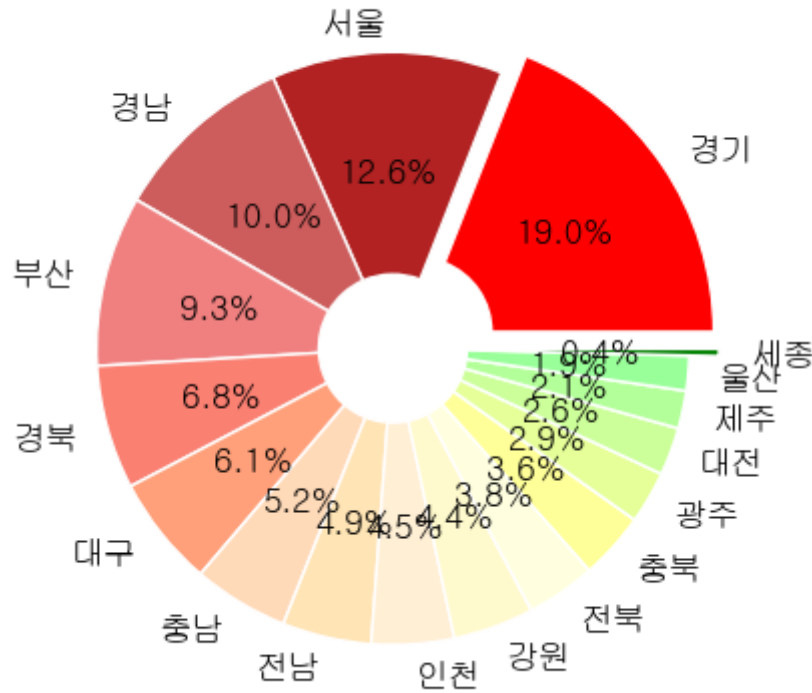
<성폭력 범죄>  
가장 많이 발생한 지역 : 서울  
가장 적게 발생한 지역 : 세종

폭행 범죄 지역별 발생 비율



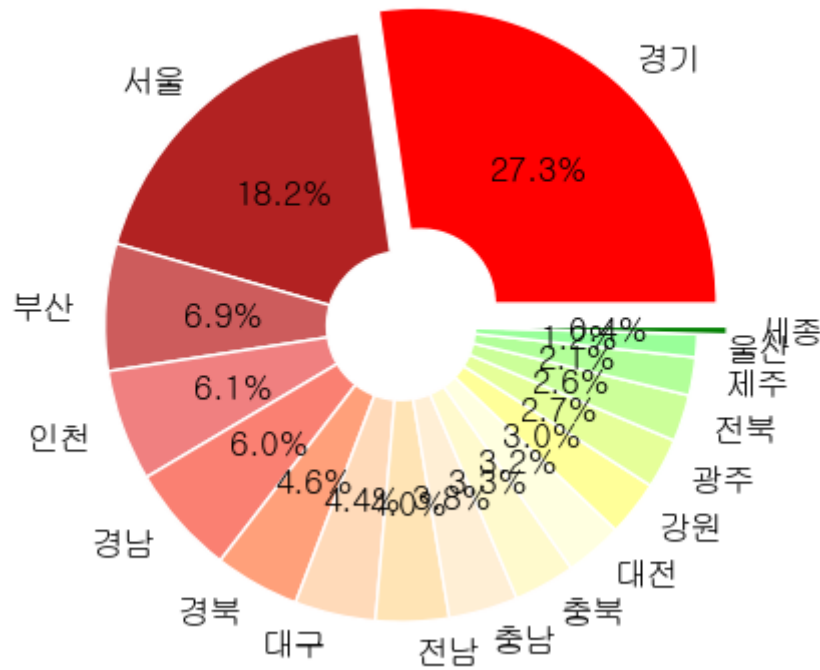
<폭행 범죄>  
가장 많이 발생한 지역 : 경기  
가장 적게 발생한 지역 : 세종

상해 범죄 지역별 발생 비율



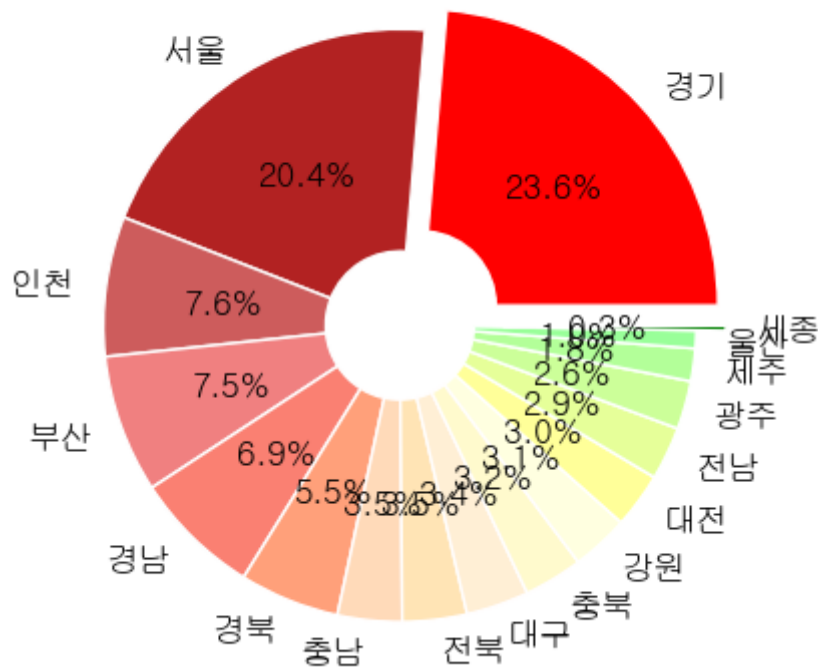
<상해 범죄>  
가장 많이 발생한 지역 : 경기  
가장 적게 발생한 지역 : 세종

협박 범죄 지역별 발생 비율



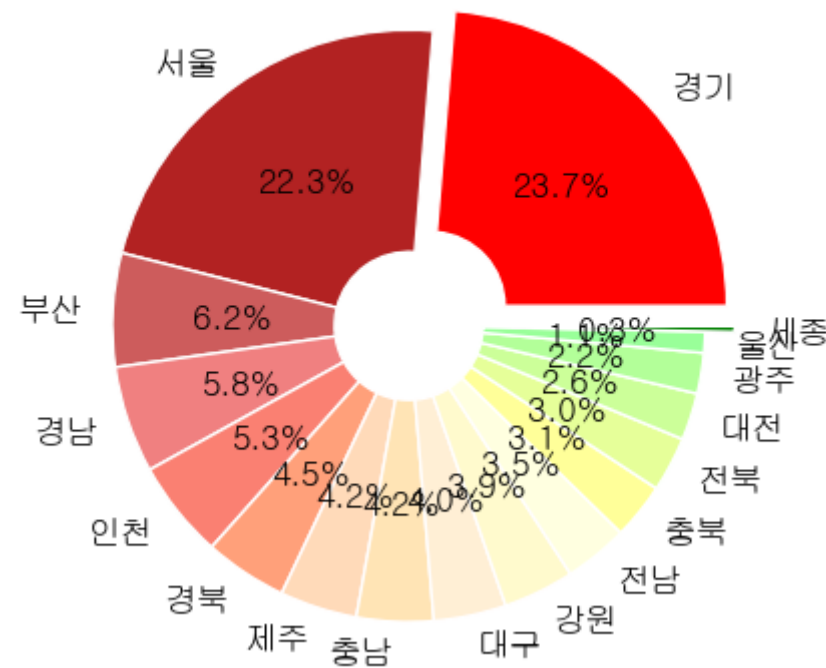
<협박 범죄>  
가장 많이 발생한 지역 : 경기  
가장 적게 발생한 지역 : 세종

# 공갈 범죄 지역별 발생 비율



<공갈 범죄>  
가장 많이 발생한 지역 : 경기  
가장 적게 발생한 지역 : 세종

# 주거침입 범죄 지역별 발생 비율



<주거침입 범죄>  
가장 많이 발생한 지역 : 경기  
가장 적게 발생한 지역 : 세종

## 다. 결론

위 자료를 토대로 우리는 대부분 범죄의 발생 비율이 높은 지역은 '22.7%'로 경기도와 '21%'로 서울 두 지역이 전국에서

가장 많은 비율을 차지하고 있음을 알 수가 있었습니다. 물론 지역별로 주로 범행이 일어나는 부분은 있겠지만,

조사한 자료를 기반으로 결론 지으면 인구가 몰려있는 만큼 범행이 일어날 확률도 높음을 알 수가 있었습니다.

이는 범죄의 발생비율만을 범죄 발생건수를 토대로 조사해 통계내었을 뿐이지, 위의 자료들이

발생비율이 높다하여 생활 수준의 높고 낮음을 평가할 수 있는 자료는 아니었습니다. 오히려 범죄 발생 비율이 높음을

자료를 통해 볼 수 있어, 이를 방지하기 위해 예산이나 인력의 추가 투입이 필요함을 알 수가 있었습니다.

## 라. 출처

'공공데이터포털 - 대검찰청 범죄 발생지 현황'

<https://www.data.go.kr/data/15085727/fileData.do>

## 3. 검거기간별 상위 5개 범죄종류 비율

### 가. 개요

우리사회는 급격한 사회전반의 트렌드가 급변하고 있으며 이는 새로운 범죄를 유발할 가능성이 높고 특히 불안감과 관련성이 높은 범죄를 관리해야 하는 중요성은 오히려 더욱 높아지고 있다.

따라서 발생한 범죄를 정확하게 조사하고 범인을 신속하게 검거하는 것이 강조된다.

이에 우리나라 38개 범죄에 대해 범죄발생부터 검거까지의 소요시간을 9개의 기간으로 분류하여 검거기간별 상위 5개 범죄 종류를 도출하였다.

```
In [5]: import csv
import numpy as np
import matplotlib.pyplot as plt
import matplotlib
matplotlib.rcParams['font.family'] = 'Malgun Gothic'
matplotlib.rcParams['axes.unicode_minus'] = False

file = open('data/검거기간.csv', 'r')
data = csv.reader(file)
sub = next(data)
header = next(data)

data_list = []
for row in data:
```

```

row[1:] = map(int, row[1:])
data_list.append(row)

cr=['살인기수', '살인미수등', '강도', '강간', '유사강간', '강제추행', '기타 강간 강제
'방화', '절도범죄', '상해', '폭행', '체포_감금', '협박', '약취_유인', '폭력행위등
'손괴', '직무유기', '직권남용', '증수뢰', '통화', '문서_인장', '유가증권인지', '
'배임', '성폭속범죄', '도박범죄', '특별경제범죄', '마약범죄', '보건범죄', '환경범죄
'교통범죄', '노동범죄', '안보범죄', '선거범죄', '병역범죄', '기타범죄']

crime_dict={}
for row in data_list:
    if row[0] not in crime_dict:
        crime_dict[row[0]]={}
        for item in range(len(row)-1):
            if cr[item] not in crime_dict[row[0]]:
                crime_dict[row[0]][cr[item]] = row[item+1]

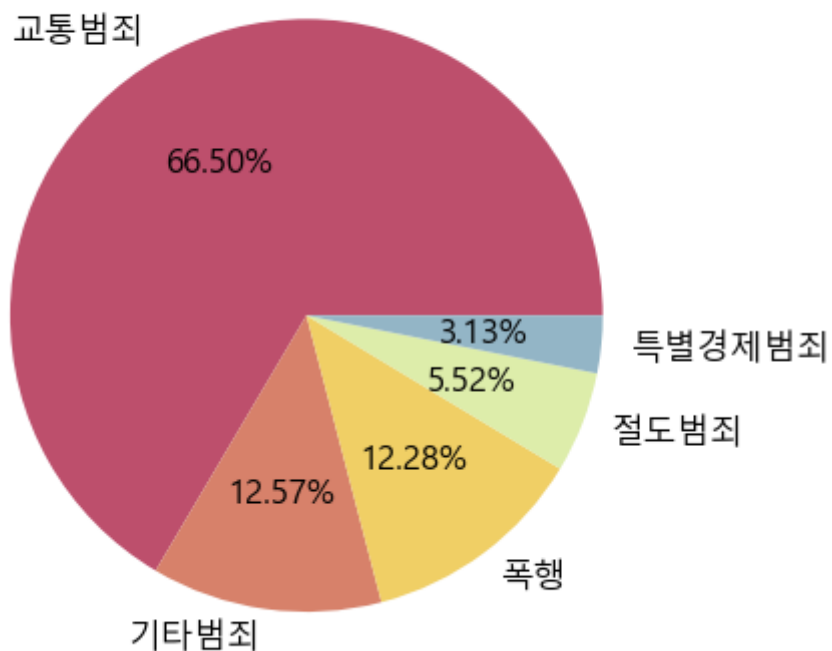
cr_sort={}

for key, value in crime_dict.items():
    va = sorted(value.items(), key = lambda item:item[1], reverse = True)
    if key not in cr_sort:
        cr_sort[key] = va[:5]

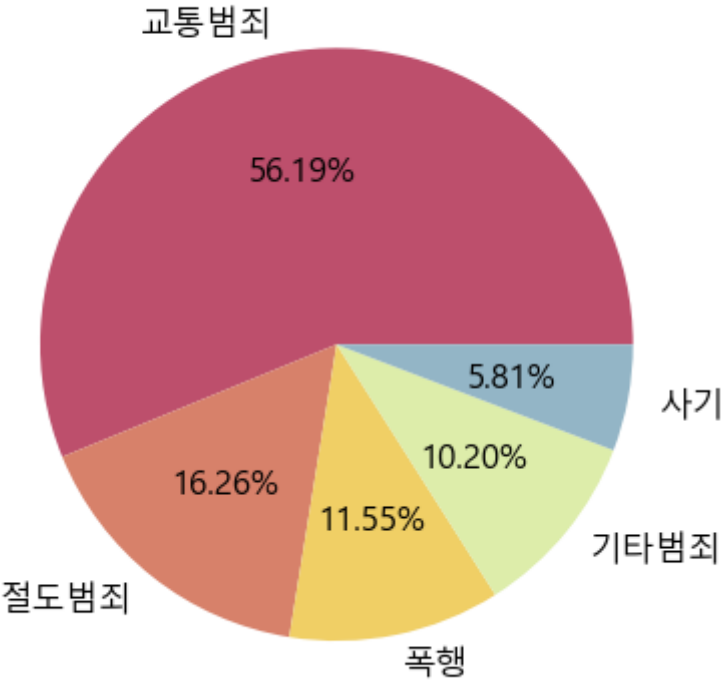
for key, value in cr_sort.items():
    bum=[]
    num=[]
    for val in value:
        bum.append(val[0])
        num.append(val[1])
my_colours = ["#bd4f6c", "#d7816a", "#f0cf65", "#ddedaa", "#93b5c6"]
plt.title('검거기간 ' + key + ' top5 범죄')
plt.pie(num, labels = bum, autopct='%0.2f%%', colors=my_colours)
plt.show()

```

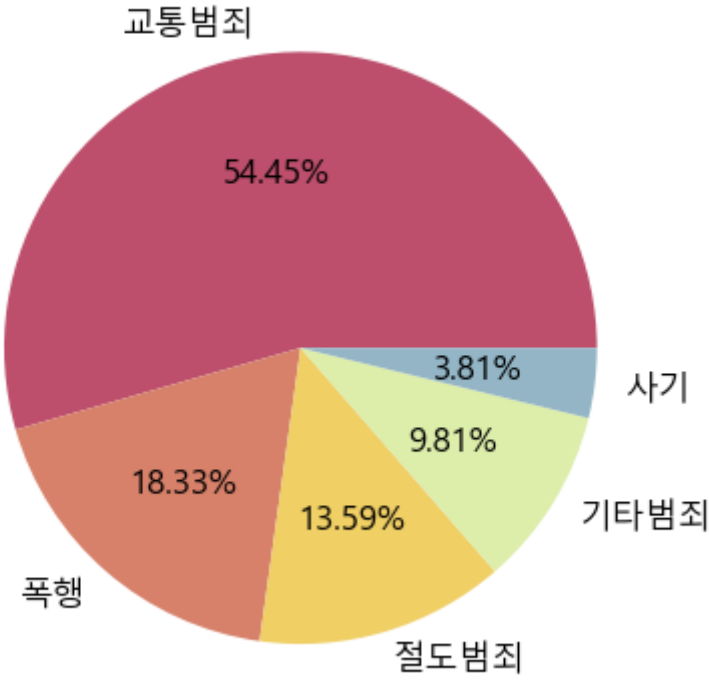
검거기간 1일 이내 top5 범죄



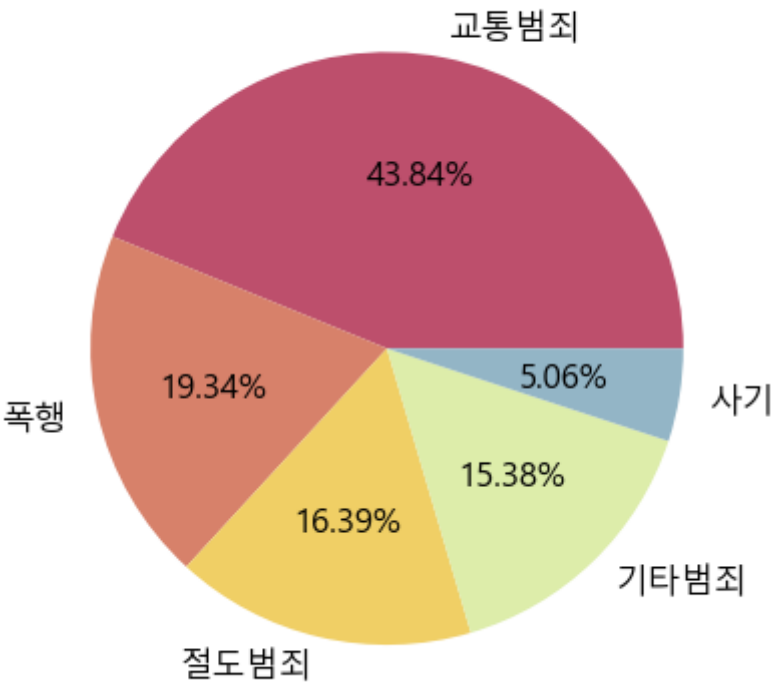
검거기간 2일 이내 top5 범죄



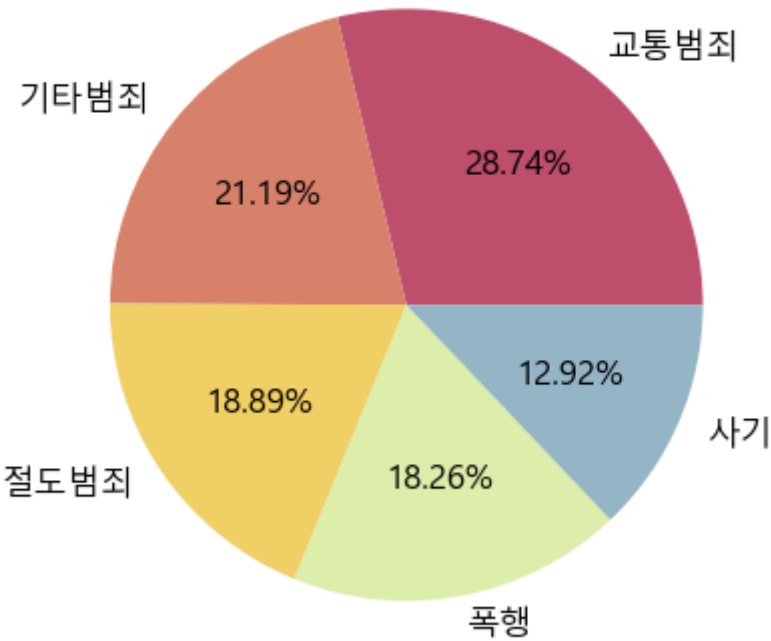
검거기간 3일 이내 top5 범죄



검거기간 10일이내 top5 범죄

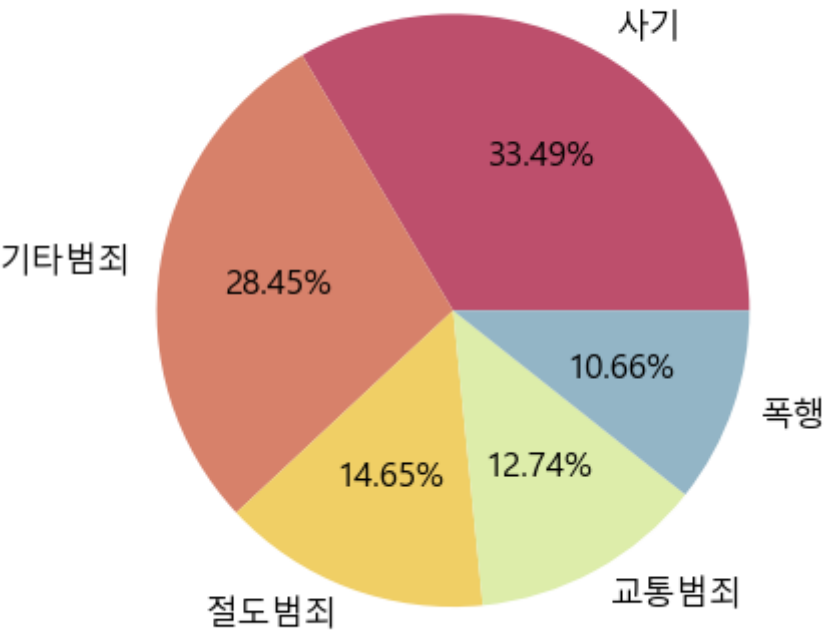


검거기간 1개월이내 top5 범죄

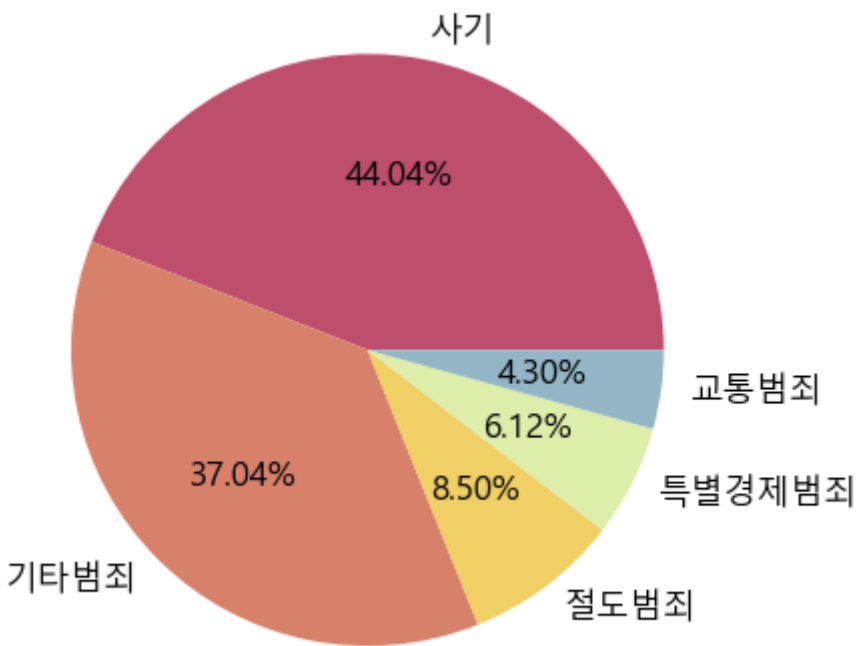




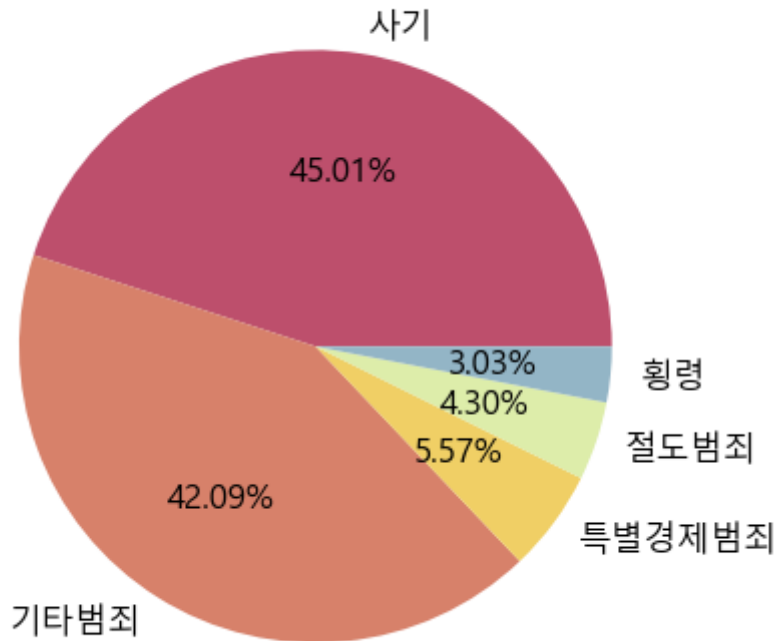
검거기간 3개월이내 top5 범죄



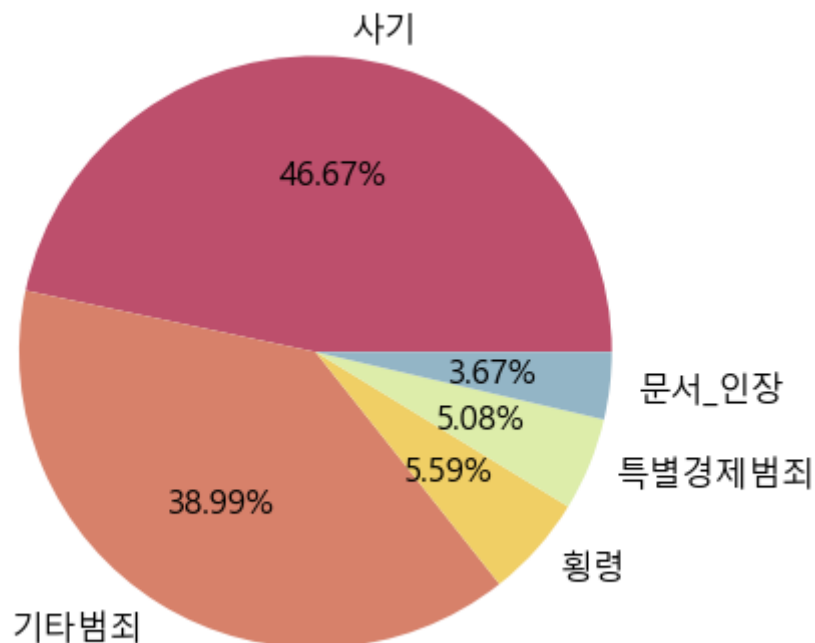
검거기간 6개월이내 top5 범죄



## 검거기간 1년 이내 top5 범죄



## 검거기간 1년 초과 top5 범죄



## 나. 결론

1일, 2일, 3일, 10일, 1개월 이내의 단기 검거 유형에서는 교통범죄가 높은 비율을 차지해 검거에 비교적 수월한 것으로 분석되었고

3개월, 6개월, 1년 이내, 1년 초과 장기 검거 유형은 사기 범죄가 가장 높아 검거에 어려움이 많은 것으로 분석되었다.

특히 교통범죄는 1일 이내 66.5%, 2일 이내 56.19%, 3일 이내 54.45%, 10일 이내 43.84%, 1개월 이내 28.74%, 3개월 이내 12.74%, 6개월 이내 4.3%로 검거 소요

일수가 타 범죄에 비해 상대적으로 짧았다.

또한 사기범죄는 단기 검거 유형에서는 낮은 비율을 차지하다가 3개월이내 33.49%, 6개월이내 44.04%, 1년이내 45.01%, 1년초과 46.67%를 차지하면서 검거에 상당히 많은 시간을 차지하는 걸 볼 수 있다.

분석한 자료를 토대로 해당 범죄가 검거기간에 따라 차지하는 비율이 높아진다는 것은 범죄 수사가 더욱 어려워진다는 것을 의미한다.

이는 범인검거를 담당하는 경찰서 내 인력투입과 운영과 밀접한 관련성이 있기 때문에 적정 인력규모나 인력재배치 등 관련 정책과 제도가 마련되어야 할 것이다.

## 다. 출처

'공공데이터포털 - 대검찰청 범죄 발생지 현황'

<https://www.data.go.kr/data/15085727/fileData.do>

## 4. 대분류(강력, 폭력, 지능) 범죄 별 피해자 성별 & 연령대 비율

### 가. 개요

범죄에는 여러 종류의 범죄가 있고 피해자가 있고, 여성과 남성의 범죄 피해 종류를 고려하여 범죄 피해자 지원방안을 마련하기 위한 기초 자료로 활용하고자 알아보았다.

```
In [7]: import csv
import matplotlib.pyplot as plt
from matplotlib import font_manager, rc

font_path = "C:/Windows/Fonts/NGULIM.TTF"
font = font_manager.FontProperties(fname=font_path).get_name()
rc('font', family=font)

file = open('data/2021_피해자_성별연령.csv', 'r')
victim = csv.reader(file)
victim_header = next(victim)

victim_list = []
for row in victim:
    row[2:] = map(int, row[2:])
    victim_list.append(row)

file.close()
```

## 나. 본론

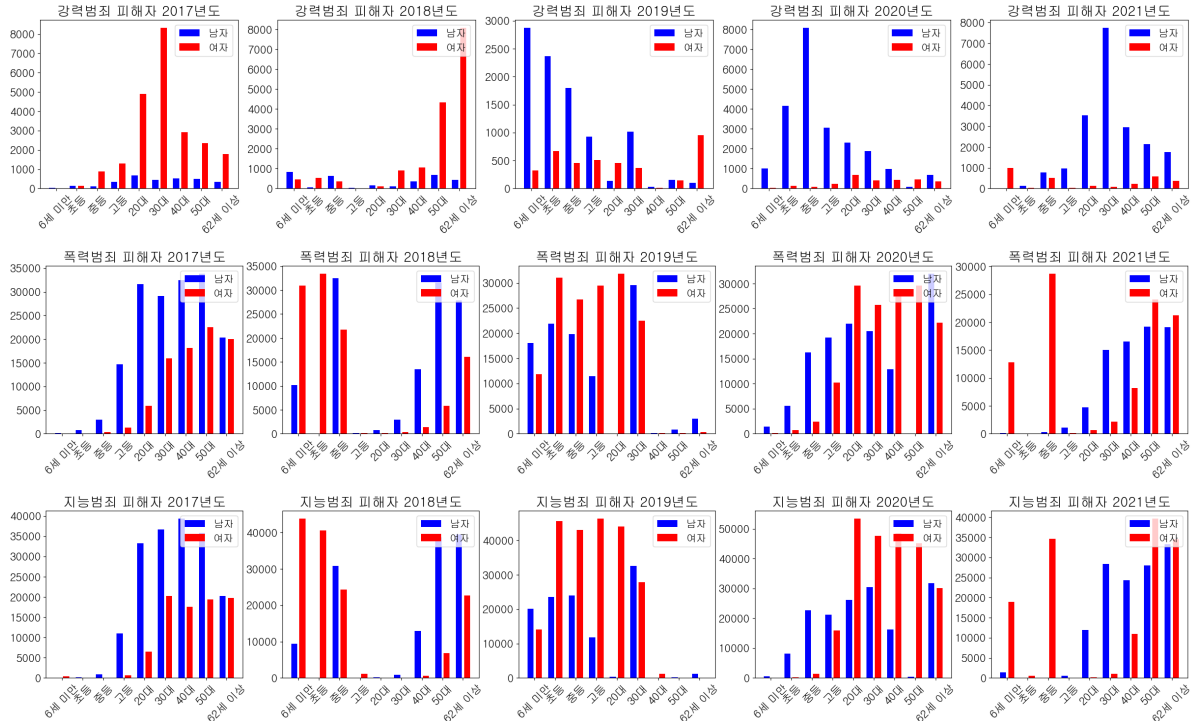
### 나-1. 추출 자료와 분류

대분류(강력, 폭력, 지능)범죄 별 8가지 종류의 피해자 데이터를 합산한 남 녀 두가지의 결과로 2017년 ~ 2021년도의 5년치 통계량을 도출했다.

합산한 데이터의 소분류로는 다음 표와 같다.



```
plt.subplot(3,5,cnt)
plt.title(key + ' 피해자 ' + str(k) + '년도', fontsize=20)
plt.bar(x2 - 0.4, m, width=w, color='blue')
plt.bar(age_title, f, width=w, color='red')
plt.xticks(x2-0.2, age_title, fontsize=16, rotation= 45)
plt.yticks(fontsize=16)
plt.legend(gender, fontsize=15, loc=1)
cnt += 1
plt.show()
```



## 다. 결론

범죄를 저지르는 사람들의 뇌는 전두엽의 한 부분인 안와전두피질의 활동이 결여돼 있다는 연구결과를 발표한 바 있다.

안와전두피질은 인지와 감정을 조절하는 자기조절중추로서, 욕구와 동기 그리고 도덕적 결정 등과 관련한 정보를 처리하기도 한다.

하지만 반대로 안와전두피질의 활동이 결여 된 사람이 훌륭한 교수가 된 케이스도 있다.(2)

따라서 뇌의 구조나 활동 여부의 중요성보다는 자라온 환경의 영향이 가장 크게 적용한다는 것이다.

만약 범죄가 줄어들길 원한다면 범죄자를 비판과 범죄의 동기에 대하여 시시비비를 가릴 것이 아니라 범죄의 이유를 재조명하고 범죄가 나오기 쉬운 사회적, 환경적 문제의 개선 필요성을 국민들이 느끼고 차근차근 개선해 나가야 할 문제라고 생각된다.

## 라. 출처

(1) 위키백과 『범죄』, wikipedia

(2) sciencetime 『살인자의 뇌 어떻게 다를까』, 이성규

## 5. 서울시 구별 CCTV 현황 분석하기

서울시 각 구별 CCTV수를 파악하고, 인구대비 CCTV 비율을 파악해서 순위 비교

인구대비 CCTV의 평균치를 확인하고 CCTV가 부족한 구를 확인

```
In [3]: import pandas as pd
```

```
CCTV_Seoul = pd.read_csv('D:/python/data/서울시 CCTV 현황.csv', encoding='euc-kr')
CCTV_Seoul.head()
```

```
Out[3]:
```

	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

```
In [4]: CCTV_Seoul.columns
```

```
Out[4]: Index(['기관명', '소계', '2013년도 이전', '2014년', '2015년', '2016년'], dtype='object')
```

```
In [5]: CCTV_Seoul.columns[0]
```

```
Out[5]: '기관명'
```

```
In [6]: CCTV_Seoul.rename(columns={CCTV_Seoul.columns[0] : '구별'}, inplace=True)
CCTV_Seoul.head()
```

```
Out[6]:
```

	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

```
In [7]: pop_Seoul = pd.read_excel('D:/python/data/서울시 인구현황.xls',
                                header = 2,
                                usecols = 'B, D, G, J, N')
pop_Seoul.head()
```

Out[7]:

	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

In [8]:

```
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()
```

Out[8]:

	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

In [9]:

```
CCTV_Seoul.head()
```

Out[9]:

	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

In [10]:

```
CCTV_Seoul.sort_values(by='소계', ascending=True).head(5)
```

Out[10]:

	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

CCTV의 전체 개수가 가장 작은 5개 구

: 도봉구 < 마포구 < 송파구 < 중랑구 < 중구

```
In [11]: CCTV_Seoul.sort_values(by='소계', ascending=False).head(5)
```

```
Out[11]:
```

	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

CCTV의 전체 개수가 가장 많은 5개 구

: 강남구 > 양천구 > 서초구 > 은평구 > 용산구

```
In [12]: CCTV_Seoul['최근증가율'] = (CCTV_Seoul['2016년'] + CCTV_Seoul['2015년'] + CCTV_Seoul
      / CCTV_Seoul['2013년도 이전'] * 100
CCTV_Seoul.sort_values(by='최근증가율', ascending=False).head(5)
```

```
Out[12]:
```

	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

2013 ~ 2016년 동안 증가율이 높은 순위:

종로구 > 도봉구 > 마포구 > 노원구 > 강동구

서울시 인구 데이터 파악하기

```
In [13]: pop_Seoul.head()
```

```
Out[13]:
```

	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

```
In [14]: pop_Seoul.drop([0], inplace=True)
pop_Seoul.head()
```



Out[14]:

	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

In [15]: pop\_Seoul['구별'].unique()

Out[15]: 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)

외국인/ 고령자 비율 구하기

In [16]: pop\_Seoul['외국인비율'] = pop\_Seoul['외국인'] / pop\_Seoul['인구수'] \* 100  
pop\_Seoul['고령자비율'] = pop\_Seoul['고령자'] / pop\_Seoul['인구수'] \* 100  
pop\_Seoul.head()

Out[16]:

	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

In [17]: pop\_Seoul.sort\_values(by='인구수', ascending=False).head(5)

Out[17]:

	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

인구수 : 송파구 > 강서구 > 강남구 > 노원구> 관악구

In [18]: pop\_Seoul.sort\_values(by='외국인', ascending=False).head(5)

Out[18]:

	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

외국인수 : 영등포구 > 구로구 > 금천구 > 관악구 > 용산구

```
In [19]: pop_Seoul.sort_values(by='외국인비율', ascending=False).head(5)
```

Out[19]:

	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

외국인 비율 : 영등포구 > 금천구 > 구로구 > 중구 > 용산구

```
In [20]: pop_Seoul.sort_values(by='고령자', ascending=False).head(5)
```

Out[20]:

	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

고령자 : 강서구 > 송파구 > 은평구 > 노원구 > 관악구

```
In [21]: pop_Seoul.sort_values(by='고령자비율', ascending=False).head(5)
```

Out[21]:

	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

고령자 비율 : 강북구 > 종로구 > 중구 > 용산구 > 서대문구

## CCTV 데이터와 인구 데이터 합치고 분석하기

In [22]:

CCTV\_Seoul.head()

Out[22]:

	구별	소계	2013년도 이전	2014년	2015년	2016년	최근증가율
0	강남구	2780	1292	430	584	932	150.619195
1	강동구	773	379	99	155	377	166.490765
2	강북구	748	369	120	138	204	125.203252
3	강서구	884	388	258	184	81	134.793814
4	관악구	1496	846	260	390	613	149.290780

In [23]:

pop\_Seoul.head()

Out[23]:

	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

In [24]:

data\_result = pd.merge(CCTV\_Seoul, pop\_Seoul, on='구별')  
data\_result.head()

Out[24]:

	구별	소계	2013 년도 이전	2014 년	2015 년	2016 년	최근증가율	인구수	한국인	외국인	고령자	외국인 비율
0	강남구	2780	1292	430	584	932	150.619195	570500.0	565550.0	4950.0	63167.0	0.867660
1	강동구	773	379	99	155	377	166.490765	453233.0	449019.0	4214.0	54622.0	0.929765
2	강북구	748	369	120	138	204	125.203252	330192.0	326686.0	3506.0	54813.0	1.061806
3	강서구	884	388	258	184	81	134.793814	603772.0	597248.0	6524.0	72548.0	1.080540
4	관악구	1496	846	260	390	613	149.290780	525515.0	507203.0	18312.0	68082.0	3.484582

In [25]:

del data\_result['2013년도 이전']  
del data\_result['2014년']  
del data\_result['2015년']  
del data\_result['2016년']  
data\_result.head()

Out[25]:

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

Out[26]:

	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

```
In [27]: data_result.rename(columns = {data_result.columns[0] : 'CCTV수'}, inplace=True)
data_result.head()
```

Out[27]:

	CCTV수	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

```
In [28]: data_result.describe()
```

Out[28]:

	CCTV수	최근증가율	인구수	한국인	외국인	고령자	외국인
<b>count</b>	25.000000	25.000000	25.000000	25.000000	25.000000	25.000000	25.000000
<b>mean</b>	1179.080000	119.030748	407904.16000	397078.720000	10825.440000	52858.320000	3.10564
<b>std</b>	556.728983	61.350043	130273.67526	131802.414218	8301.513236	14173.072031	2.48904
<b>min</b>	485.000000	34.671731	133240.00000	124312.000000	2017.000000	20764.000000	0.57854
<b>25%</b>	748.000000	64.973730	330192.00000	326686.000000	4445.000000	48161.000000	0.95841
<b>50%</b>	1015.000000	104.347826	412520.00000	400456.000000	8928.000000	54173.000000	2.52663
<b>75%</b>	1496.000000	150.619195	479978.00000	475949.000000	14747.000000	63167.000000	4.17244
<b>max</b>	2780.000000	248.922414	667483.00000	660584.000000	34913.000000	72548.000000	8.66359

## 외국인 및 고령자 비율

```
In [29]: import matplotlib.pyplot as plt
import seaborn as sns

#%%matplotlib inline

import platform

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

```
In [30]: target_col = ['외국인비율', '고령자비율']

data_result_sort = data_result.sort_values(by='고령자비율', ascending=False)

plt.figure(figsize = (10,10))
sns.heatmap(data_result_sort[target_col], annot=True, fmt='f',
            linewidths=.5, cmap='RdPu')
plt.title('외국인 및 고령자 비율(구별 고령자 비율 역순으로 정렬)')
plt.show()
```



```
In [31]: import numpy as np
np.corrcoef(data_result['고령자비율'], data_result['CCTV수'])
```

```
Out[31]: array([[ 1.          , -0.28078554],
                [-0.28078554,  1.          ]])
```

```
In [32]: np.corrcoef(data_result['외국인비율'], data_result['CCTV수'])
```

```
Out[32]: array([[ 1.          , -0.13607433],
                [-0.13607433,  1.          ]])
```

```
In [33]: np.corrcoef(data_result['인구수'], data_result['CCTV수'])
```

```
Out[33]: array([[ 1.          ,  0.30634228],
                [ 0.30634228,  1.          ]])
```

```
In [34]: data_result.sort_values(by='CCTV수', ascending=False).head(5)
```

Out[34]:

	CCTV수	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

In [35]:

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

Out[35]:

	CCTV수	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

인구수가 많은 구부터 보기

: 송파구 > 강서구 > 강남구 > 노원구 > 관악구

## CCTV와 인구현황 그래프로 분석하기

### CCTV 현황과 서울시 인구 현황 최종 데이터 확인

In [36]:

```
data_result.head()
```

Out[36]:

	CCTV수	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

In [37]:

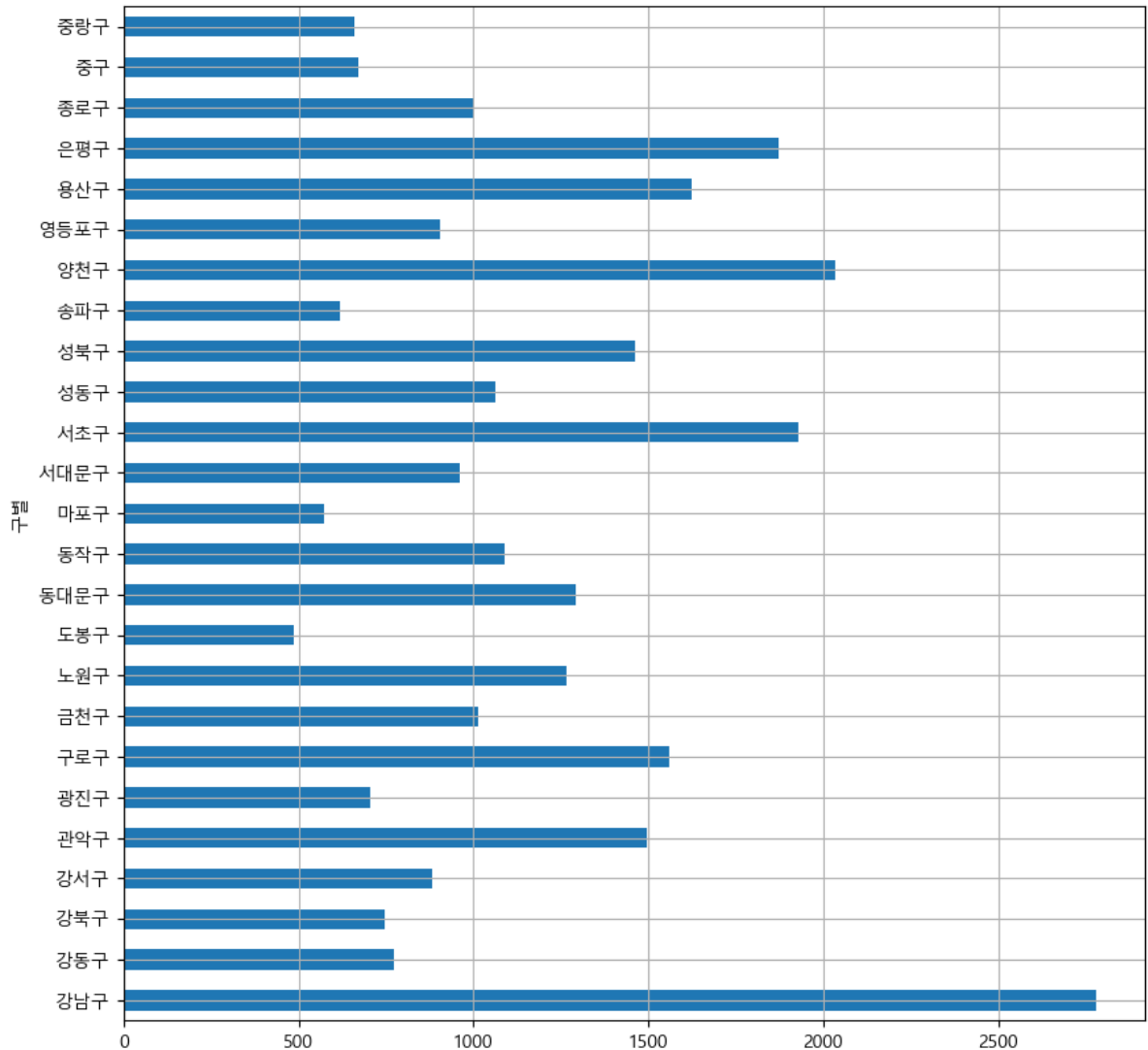
```
import matplotlib.pyplot as plt
import platform

from matplotlib import font_manager, rc
plt.rcParams['axes.unicode_minus'] = False

if platform.system() == 'Darwin':
    rc('font', family='AppleGothic')
```

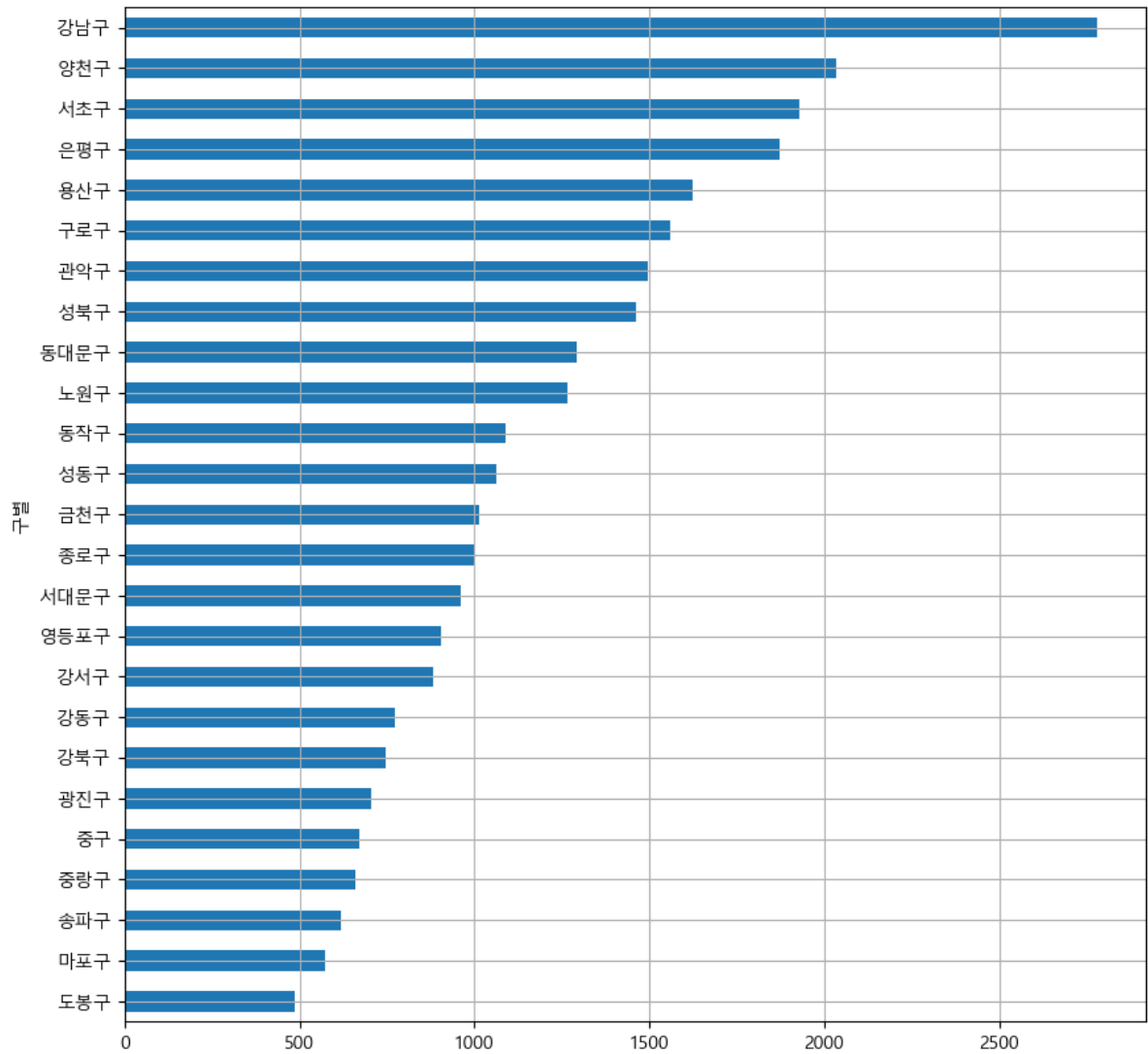
```
elif platform.system() == 'Windows':
    path = "c:/Windows/Fonts/malgun.ttf"
    font_name = font_manager.FontProperties(fname=path).get_name()
    rc('font', family=font_name)
else:
    print('Unknown system...')
```

```
In [38]: import matplotlib.pyplot as plt
plt.figure()
data_result['CCTV수'].plot(kind='barh', grid=True, figsize=(10,10))
plt.show()
```



```
In [39]: data_result['CCTV수'].sort_values().plot(kind='barh',
                                                    grid=True, figsize=(10,10))
plt.show()
```



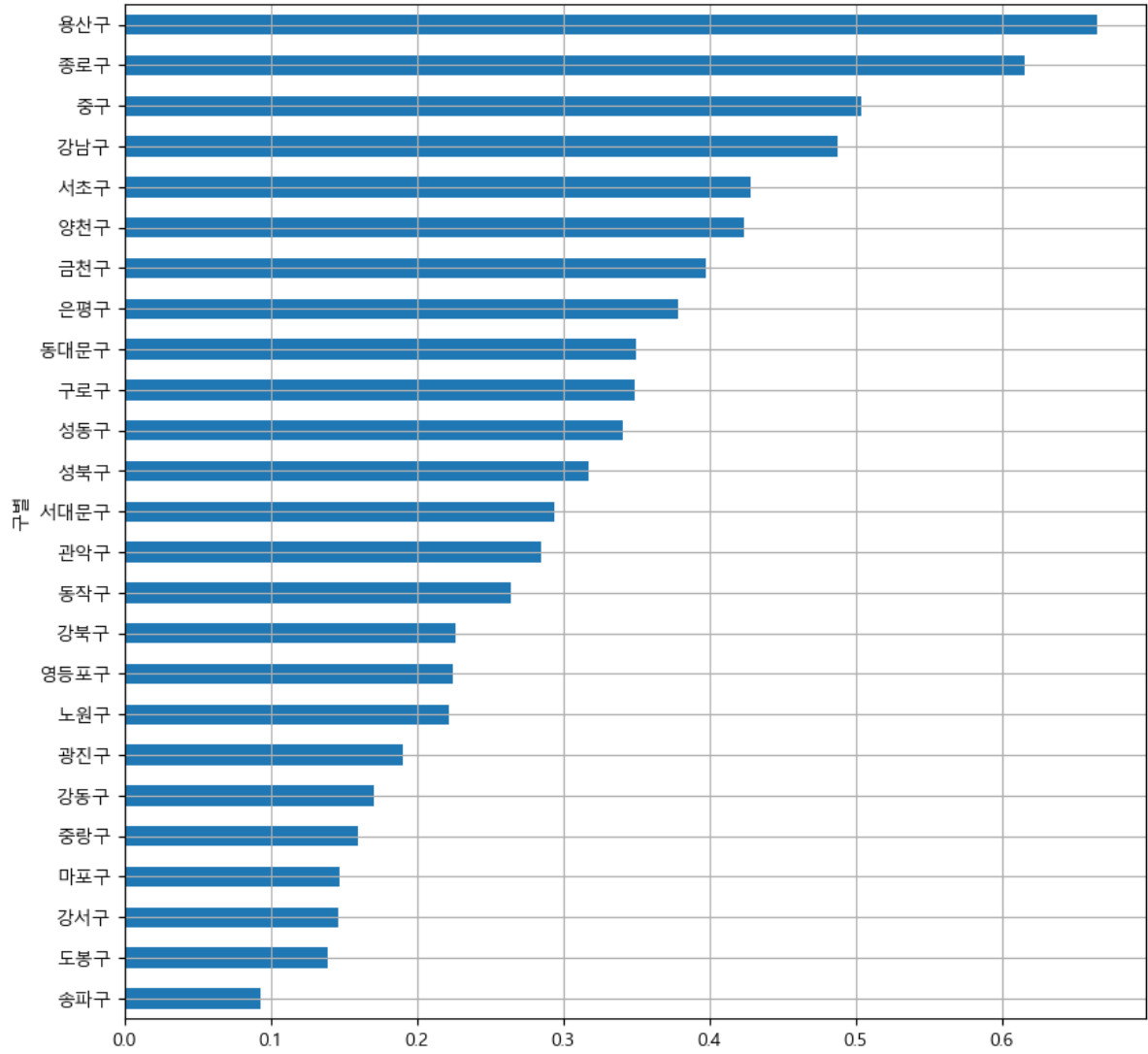


## 인구 대비 CCTV 비율

```
In [40]: data_result['CCTV비율'] = data_result['CCTV수'] / data_result['인구수'] * 100

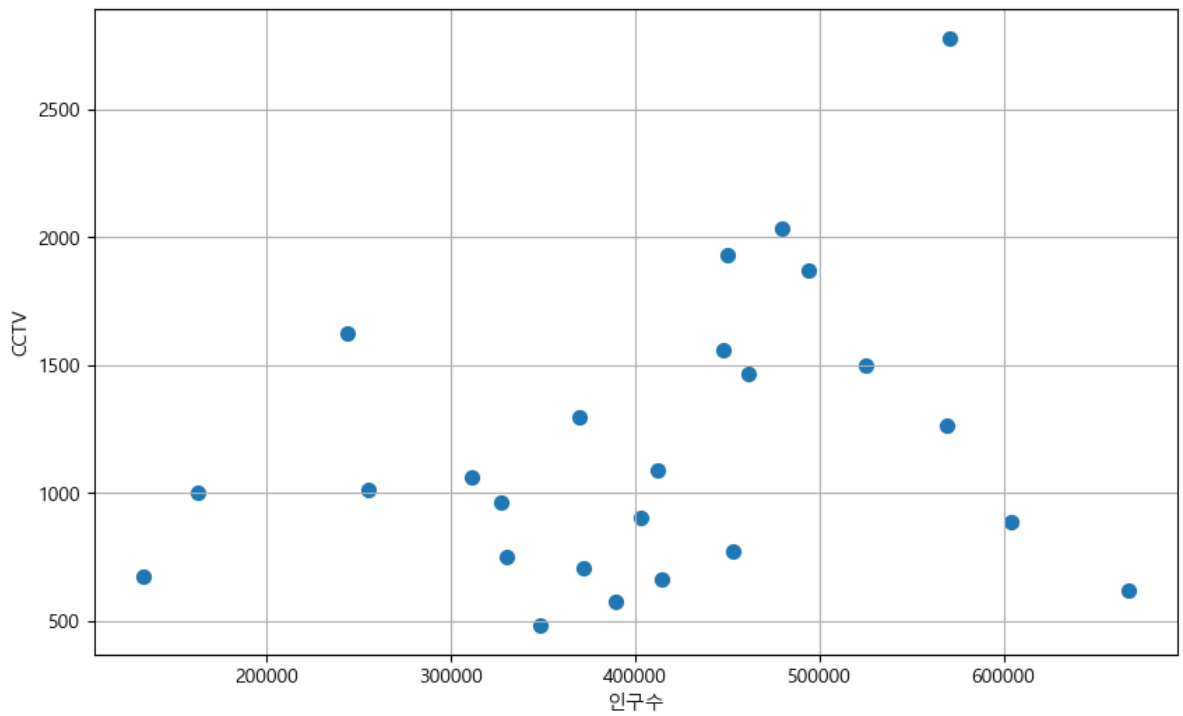
data_result['CCTV비율'].sort_values().plot(kind='barh',
                                             grid=True, figsize=(10,10))

plt.show()
```



### CCTV와 인구현황 그래프로 분석하기

```
In [41]: plt.figure(figsize=(10,6))
plt.scatter(data_result['인구수'], data_result['CCTV수'], s=50)
plt.xlabel('인구수')
plt.ylabel('CCTV')
plt.grid()
plt.show()
```

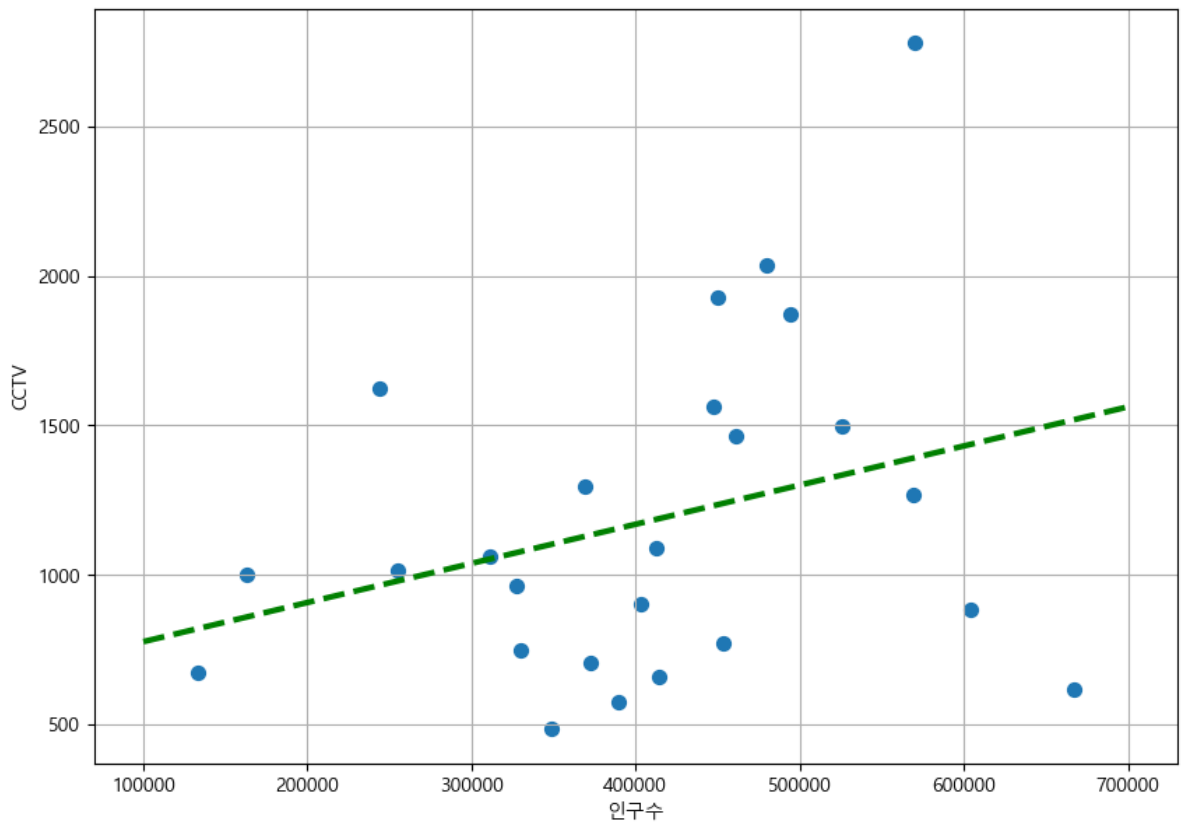


```
In [42]: fp1 = np.polyfit(data_result['인구수'], data_result['CCTV수'], 1)
         fp1
```

```
Out[42]: array([1.30916415e-03, 6.45066497e+02])
```

```
In [43]: f1 = np.poly1d(fp1)
         fx = np.linspace(100000, 700000, 100)

         plt.figure(figsize=(10,7))
         plt.scatter(data_result['인구수'], data_result['CCTV수'], s=50)
         plt.plot(fx, f1(fx), ls='dashed', lw=3, color='g')
         plt.xlabel('인구수')
         plt.ylabel('CCTV')
         plt.grid()
         plt.show()
```



위의 결과를 보면 CCTV와 인구수는 양의 상관관계를 가진다는 것을 알 수 있다

## 구별 인구수와 CCTV 수

```
In [44]: fp1 = np.polyfit(data_result['인구수'], data_result['CCTV수'], 1)

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

data_result['오차'] = np.abs(data_result['CCTV수'] - f1(data_result['인구수']))

df_sort = data_result.sort_values(by='오차', ascending=False)
df_sort.head()
```

Out[44]:

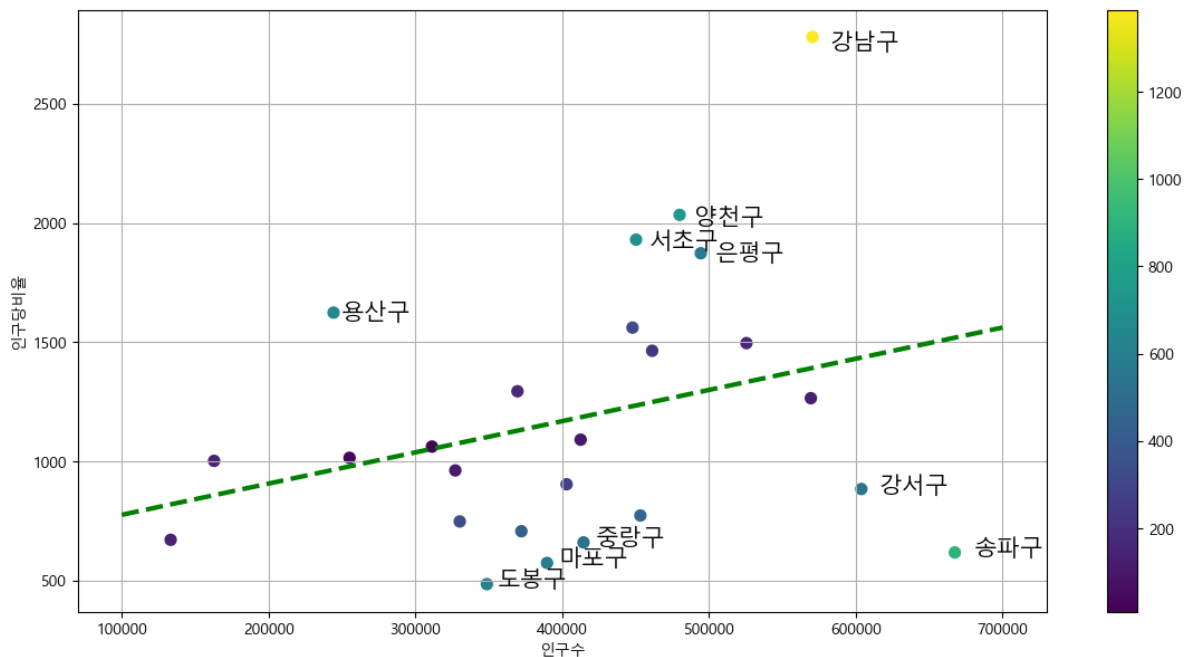
구별	CCTV 수	최근증가율	인구수	한국인	외국인	고령자	외국인 비율	고령자비율	CCTV비율	
강남구	2780	150.619195	570500.0	565550.0	4950.0	63167.0	0.867660	11.072217	0.487292	1388.05
송파구	618	104.347826	667483.0	660584.0	6899.0	72506.0	1.033584	10.862599	0.092587	900.97
양천구	2034	34.671731	479978.0	475949.0	4029.0	52975.0	0.839413	11.036964	0.423769	760.56
서초구	1930	63.371266	450310.0	445994.0	4316.0	51733.0	0.958451	11.488308	0.428594	695.40
용산구	1624	53.216374	244203.0	229456.0	14747.0	36231.0	6.038828	14.836427	0.665020	659.23

In [45]:

```
plt.figure(figsize=(14,7))
plt.scatter(data_result['인구수'], data_result['CCTV수'],
            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['CCTV수'][n]*0.98,
            df_sort.index[n], fontsize=15)

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



## 구별 CCTV 지도 시각화

서울시 중심의 위도와 경도 정보 입력하고 경계선 그리고 컬러맵은 CCTV소계로 지정

```
In [46]: import folium
import json
import warnings

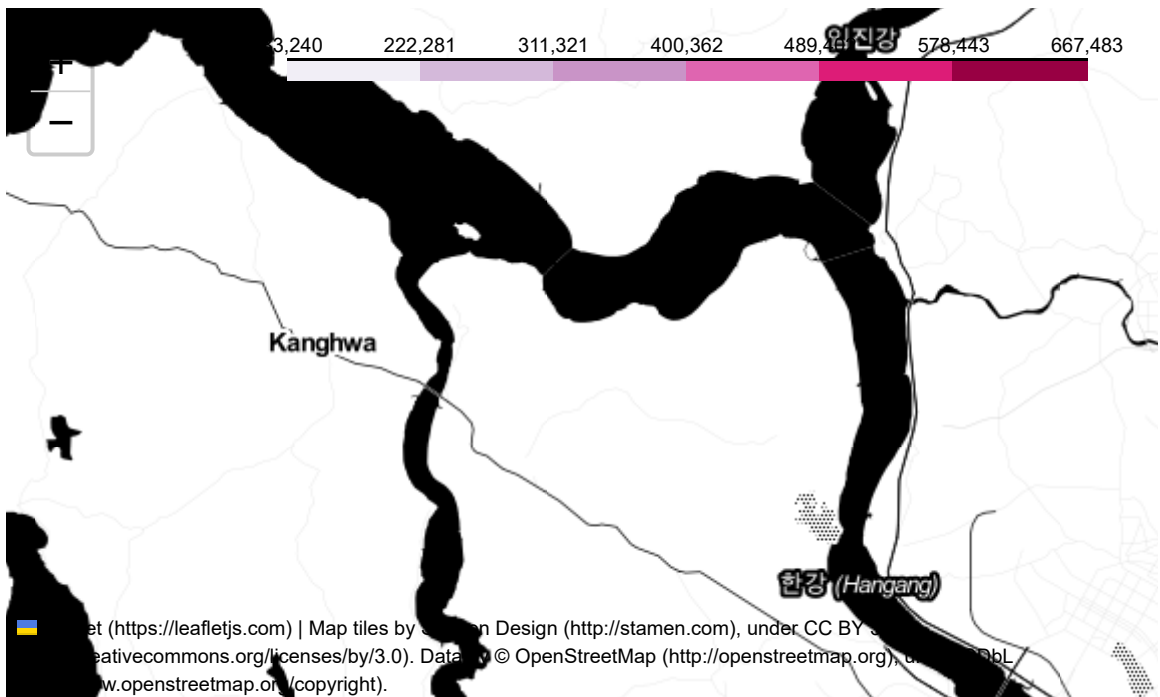
geo_path = 'D:/python/data/skorea_municipalities_geo_simple.json'
geo_str = json.load(open(geo_path, encoding='utf-8'))
warnings.filterwarnings(action='ignore')

map = folium.Map(location=[37.5502, 126.982], zoom_start=11,
                  tiles='Stamen Toner')

map.choropleth(geo_data = geo_str,
               data = data_result['CCTV수'],
               columns = [data_result.index, data_result['CCTV수']],
               fill_color = 'YlGnBu', #PuRd, YlGnBu
               key_on = 'feature.id')

map
```

Out[47]:



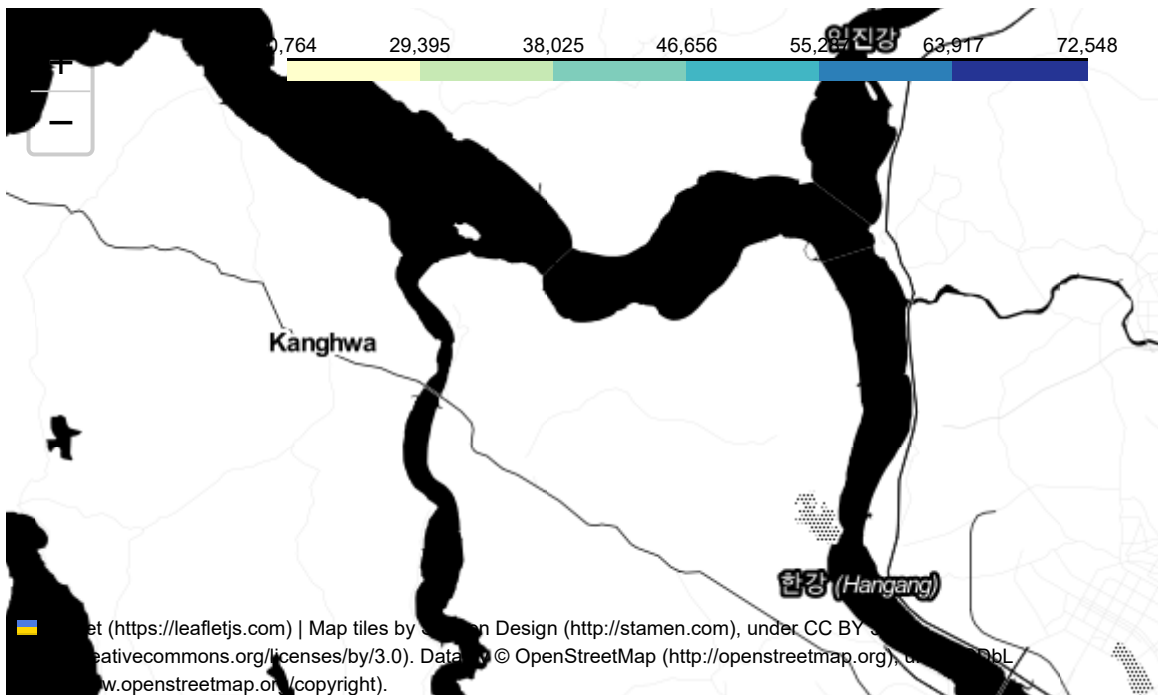
### 서울시 구별 고령자수로 컬러맵 지정

```
In [48]: map = folium.Map(location=[37.5502, 126.982], zoom_start=11,
                             tiles='Stamen Toner')

map.choropleth(geo_data = geo_str,
               data = data_result['고령자'],
               columns = [data_result.index, data_result['고령자']],
               fill_color = 'YlGnBu', #PuRd, YlGnBu
               key_on = 'feature.id')

map
```

Out[48]:



### 서울시 구별 CCTV 비율로 컬러맵 지정

```
In [49]: # 서울시 구별 CCTV 비율로 컬러맵 지정
map = folium.Map(location=[37.5502, 126.982], zoom_start=11,
                  tiles='Stamen Toner')
```

