

데이터 분석 포트폴리오 작성

(범죄 발생에 대한 대처와 해결방안)

- 목차

- 0. 데이터 분석을 위한 필요한 모듈 설치 및 준비하기
- 1. 제일 많이 일어나는 범죄(대분류)
- 2. 제일 많이 일어나는 범죄(중분류)
- 3. 범죄가 많이 일어나는 지역
- 4. 범죄가 많이 일어나는 장소
- 5-1. 범죄가 많이 일어나는 시간대
- 5-2. 범죄가 많이 일어나는 요일
- 6. 범죄 발생으로부터 인지하는데 까지 걸리는 기간
- 7. 전국 범죄 발생과 검거현황
- 8. 지역별 인구수에 따른 범죄 발생 율
- 9. 최종결과

0. 데이터 분석을 위한 필요한 모듈 설치 및 준비하기

<필요한 모듈 설치 및 준비하기>

```
# 날짜 모듈
import datetime

# 결측값 확인
import missingno as msno

# 그래프 그릴때 필요한 라이브러리
import seaborn as sns
import matplotlib.pyplot as plt

# 그래프 글자 출력 오류 해결
import matplotlib
import matplotlib.font_manager as fm

font_location = 'c:/windows/fonts/H2GTRE.TTF'
font_name = fm.FontProperties(fname=font_location).get_name()
matplotlib.rc('font', family = font_name)

import warnings as wa
# 오류 메시지 차단
wa.filterwarnings("ignore")

# pandas, numpy 모듈
import pandas as pd
import numpy as np

# 데이터 프레임 이미지로 변환 모듈
import dataframe_image as dfi
```

<파일 불러오기>

```
data = pd.read_csv('data/crime/경찰청_범죄 발생 지역별 통계_20151231.csv', encoding='cp949')
data

data1 = pd.read_csv('data/crime/경찰청_범죄 발생 장소별 통계_20221231.csv', encoding='cp949')
data1

data2 = pd.read_csv('data/crime/경찰청_범죄 발생 시간대 및 요일_20191231.csv', encoding='cp949')
data2

data3 = pd.read_csv('data/crime/경찰청_범죄 발생부터 인지까지의 기간_20211231.csv', encoding='cp949')
data3

data4 = pd.read_csv('data/crime/경찰청_전국 범죄 발생 및 검거 현황_20211231.csv', encoding='cp949')
data4

data5 = pd.read_csv('data/crime/202309_202309_주민등록인구및세대현황_월간.csv', encoding='cp949')
data5
```

1. 제일 많이 일어나는 범죄(대분류)

1. 데이터 분석

```
# 범죄대분류를 복사  
data['범죄분류'] = data['범죄대분류']  
data
```

```
# 범죄 대분류의 갯수  
data.groupby(['범죄대분류']).count()
```

```
# 범죄대분류를 기준으로 한 합계  
crime_class = data.groupby('범죄대분류').sum()  
crime_class
```

```
# 범죄대분류별 합계 계산  
# axis=1 은 같은 행별로 계산하는 옵션  
crime_class_sum = np.sum(crime_class.groupby('범죄대분류').sum(), axis=1)  
crime_class_sum
```

2. 데이터 정제

```
# 데이터프레임 형식으로 인덱스 재설정  
crime_class_g = pd.DataFrame(crime_class_sum).reset_index()  
  
# 컬럼명 변경  
# inplace = True: 기존 데이터프레임에 변경된 설정으로 덮어쓰겠다는 의미  
crime_class_g.rename(columns = {0: '총 합계'}, inplace = True)  
  
# 총 합계를 기준으로 내림차순 정렬  
# ascending=False: 내림차순, True는 오름차순  
crime_class_g.sort_values(by='총 합계', ascending=False, inplace = True)  
  
crime_class_g
```

3. 데이터 시각화

```
plt.figure(figsize=(20, 20))  
sns.barplot(x='범죄대분류', y='총 합계', data=crime_class_g)  
sns.despine(left=True) # 위, 오른쪽 축 숨기기  
# 한글깨짐 방지, 마이너스기호 깨짐 방지, 하얀배경에 선으로 보기편한 스타일  
sns.set_theme(font = 'Malgun Gothic', rc = {'axes.unicode_minus' : False}, style = 'whitegrid')  
plt.title('제일 많이 일어나는 범죄(대분류)', fontsize=20, fontfamily='Malgun Gothic', fontweight='bold', position=(0, 0))  
plt.xticks(rotation=30, fontsize=15)  
plt.savefig('data/crime/분석 그래프/1. 제일 많이 일어나는 범죄(대분류).png')  
plt.show()
```

1. 제일 많이 일어나는 범죄(대분류)

	범죄대분류	총 합계
1	교통범죄	596665
10	지능범죄	316121
12	폭력범죄	305947
9	절도범죄	245853
2	기타범죄	216197
11	특별경제범죄	86329
0	강력범죄	25334
13	풍속범죄	24491
5	병역범죄	18726
6	보건범죄	14602
4	마약범죄	6411
14	환경범죄	2955
3	노동범죄	1145
7	선거범죄	760
8	안보범죄	121

▶ 분석목표: 제일 많이 일어나는 범죄는 어떤 것들이 있는지 알아보기

▶ 활용한 데이터

- 경찰청_범죄 발생 지역별 통계_20151231.csv

▶ 표 생성 과정

- 범죄 대분류를 기준으로 그룹화 시킨 후 합계를 구하고 높은 순으로 재정렬 시킨다.

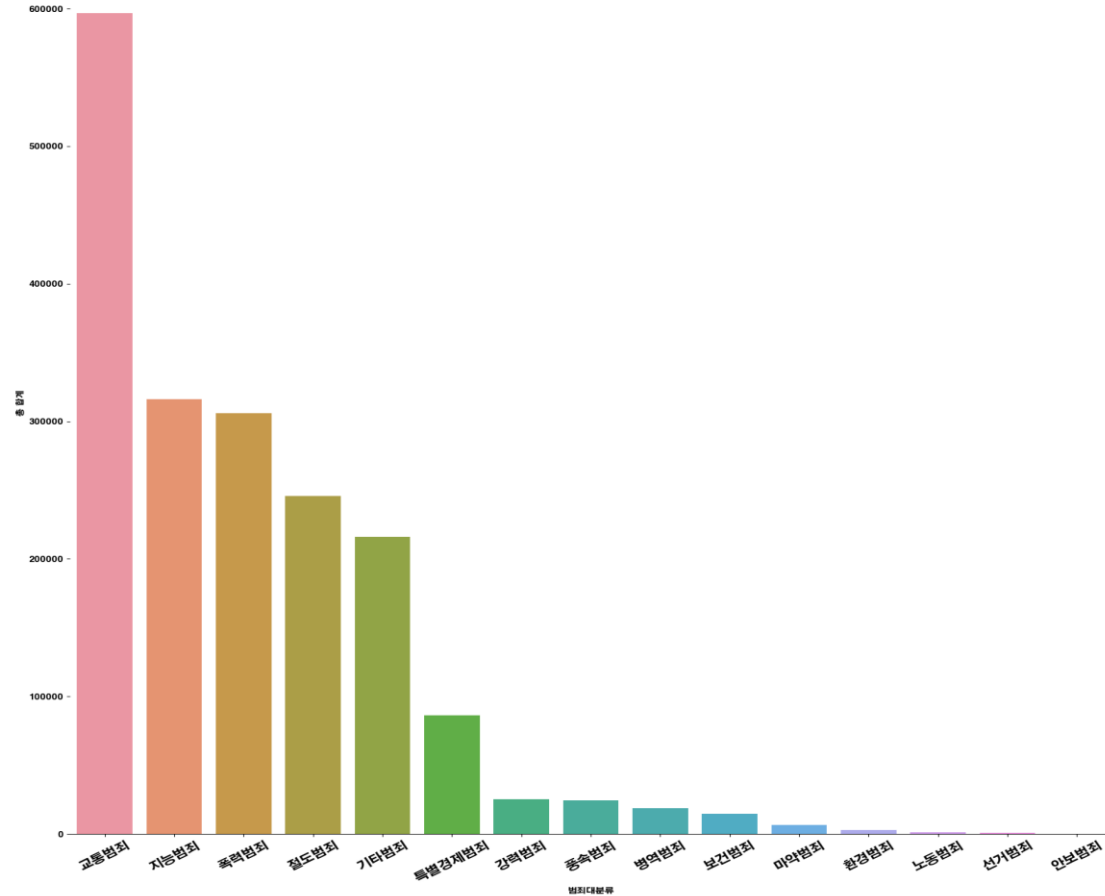
▶ 표 분석 결과

- 제일 많이 일어나는 범죄 1순위: 교통 범죄

- 2순위는 지능범죄, 3순위는 폭력범죄가 제일 많이 일어난다.

1. 제일 많이 일어나는 범죄(대분류)

제일 많이 일어나는 범죄(대분류)



그래프 분석 결과: 제일 많이 일어나는 범죄 1순위 는 교통 범죄이며 2순위는 지능범죄, 3순위는 폭력범죄가 제일 많이 일어나는 범죄 입니다.

2. 제일 많이 일어나는 범죄(중분류)

1. 데이터 분석

```
# 범죄중분류를 기준으로 한 합계
crime_class_s = data.groupby('범죄중분류').sum()
crime_class_s
```

```
# axis=1 은 같은 행별로 계산하는 옵션
crime_class_s_sum = np.sum(crime_class_s.groupby('범죄중분류').sum(), axis=1)
crime_class_s_sum
```

2. 데이터 정제

```
# 데이터프레임 형식으로 인덱스 재설정
# reset_index(): 기존 행 인덱스를 제거하고 인덱스를 데이터 열 추가
crime_class_s_g = pd.DataFrame(crime_class_s_sum).reset_index()

# 컬럼명 변경
# inplace = True: 기존 데이터프레임에 변경된 설정으로 덮어쓰겠다는 의미
crime_class_s_g.rename(columns = {0: '총 합계'}, inplace = True)

# 총 합계를 기준으로 내림차순 정렬
# ascending=False: 내림차순, True는 오름차순
crime_class_s_g.sort_values(by='총 합계', ascending=False, inplace = True)

crime_class_s_g
```

3. 데이터 시각화

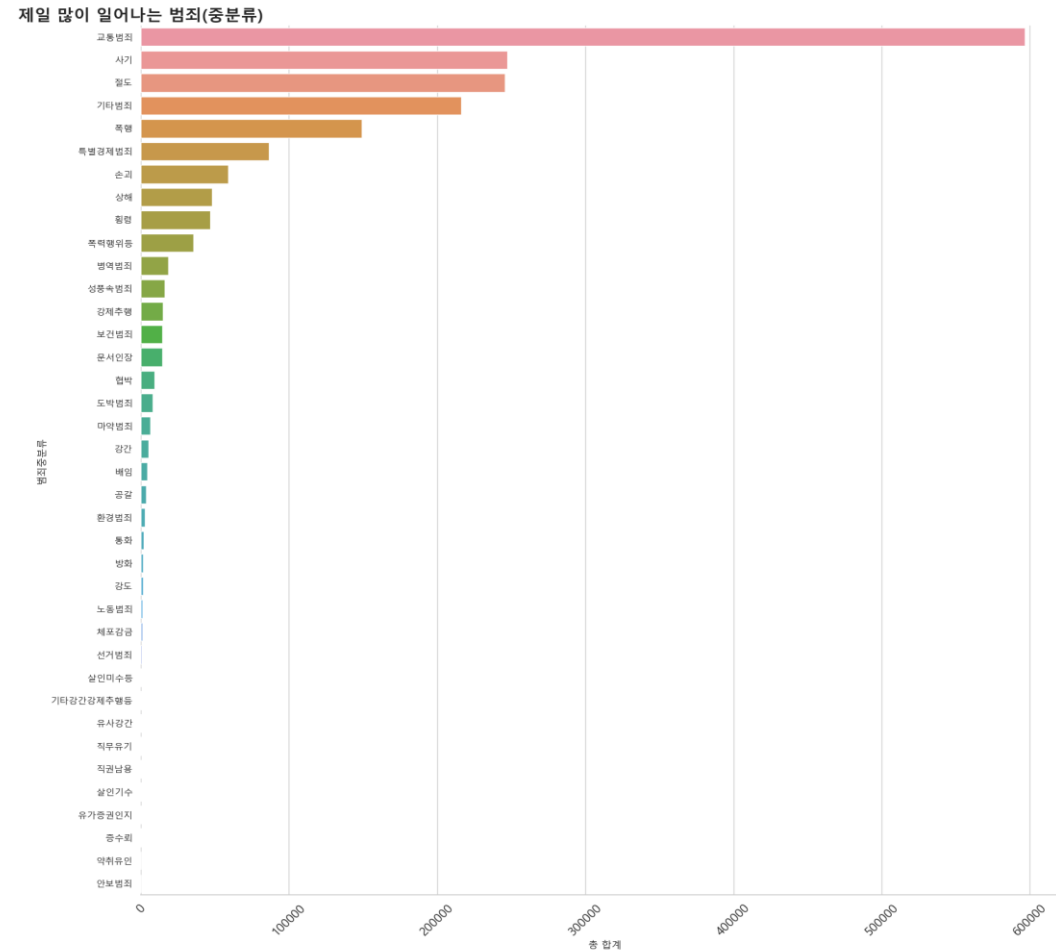
```
plt.figure(figsize=(20, 20))
sns.barplot(x='총 합계', y='범죄중분류', data=crime_class_s_g)
sns.despine(left=True) # 위, 오른쪽 축 숨기기
# 한글깨짐 방지, 마이너스기호 깨짐 방지, 하얀배경에 선으로 보기편한 스타일
sns.set_theme(font='Malgun Gothic', rc = {'axes.unicode_minus': False}, style='whitegrid')
plt.title('제일 많이 일어나는 범죄(중분류)', fontsize=20, fontfamily='Malgun Gothic', fontweight='bold', position=(0, 0))
plt.xticks(rotation=45, fontsize=15)
plt.savefig('data/crime/분석 그래프/2. 제일 많이 일어나는 범죄(중분류).png')
plt.show()
```

2. 제일 많이 일어나는 범죄(중분류)

	범죄중분류	총 합계
4	교통범죄	596665
15	사기	247293
26	절도	245853
6	기타범죄	216197
34	폭행	149278
32	특별경제범죄	86329
21	손괴	58977
18	상해	48005
37	횡령	46734
33	폭력행위등	35686
13	병역범죄	18726
20	성폭속범죄	16219
2	강제추행	15059
14	보건범죄	14602
10	문서인장	14546
35	협박	9187
8	도박범죄	8272
9	마약범죄	6411
0	강간	5151
12	배임	4289
3	공갈	3596
36	환경범죄	2955
31	통화	1822
11	방화	1673
1	강도	1446
7	노동범죄	1145
30	체포감금	1028
19	선거범죄	760
17	살인미수등	564
5	기타강간강제추행등	558
25	유사강간	518
29	직무유기	507
28	직권남용	401
16	살인기수	365
24	유가증권인지	282
27	중수뢰	247
23	약취유인	190
22	안보범죄	121

- ▶ 분석목표: 세부적으로 제일 많이 일어나는 범죄는 어떤 것들이 있는지 알아보기
- ▶ 활용한 데이터
 - 경찰청_범죄 발생 지역별 통계_20151231.csv
- ▶ 표 생성 과정
- ▶ - 범죄 중분류를 기준으로 그룹화 시킨 후 합계를 구하고 높은 순으로 재정렬 시킨다.
- ▶ 표 분석 결과
- ▶ - 제일 많이 일어나는 범죄 1순위: 교통 범죄
- ▶ - 2순위는 사기, 3순위는 절도가 제일 많이 일어난다.

2. 제일 많이 일어나는 범죄(중분류)



그래프 분석 결과: 세부적으로 많이 일어나는 범죄는 교통범죄가 1순위이며 다음으로는 사기와 절도, 기타 범죄, 폭행이 많이 일어나는 범죄입니다.

3. 범죄가 많이 일어나는 지역

1. 데이터 분석

```
# 범죄 대분류의 갯수확인하고 T로 행, 열 위치 교체  
data.groupby(['범죄대분류']).count().T
```

```
# 범죄대분류를 기준으로 행, 열 위치 교체와 지역별 합계  
crime_class_t = data.groupby('범죄대분류').sum().T  
crime_class_t
```

```
# 기존에 있는 범죄 대분류의 행의 값을 새로운 열로 생성  
crime_class_t = crime_class_t.reset_index()  
  
# 컬럼명 변경  
# inplace = True: 기존 데이터프레임에 변경된 설정으로 덮어쓰겠다는 의미  
crime_class_t.rename(columns = {'index' : '세부지역'}, inplace = True)
```

```
crime_class_t
```

```
# 지역 별로 구분 해보기  
def region(region_g):  
    if '서울' in region_g:  
        return '서울'  
    elif '부산' in region_g:  
        return '부산'  
    elif '대구' in region_g:  
        return '대구'  
    elif '인천' in region_g:  
        return '인천'  
    elif '광주' in region_g:  
        return '광주'  
    elif '대전' in region_g:  
        return '대전'  
    elif '울산' in region_g:  
        return '울산'  
    elif '세종' in region_g:  
        return '세종'  
    elif '경기' in region_g:  
        return '경기'  
    elif '강원' in region_g:  
        return '강원'  
    elif '충북' in region_g:  
        return '충북'  
    elif '충남' in region_g:  
        return '충남'  
    elif '전북' in region_g:  
        return '전북'  
    elif '전남' in region_g:  
        return '전남'  
    elif '경북' in region_g:  
        return '경북'  
    elif '경남' in region_g:  
        return '경남'  
    elif '제주' in region_g:  
        return '제주'  
    elif '기타도시' in region_g:  
        return '기타도시'  
    elif '도시미외' in region_g:  
        return '도시미외'  
  
crime_class_t['지역'] = crime_class_t['세부지역'].apply(region)  
crime_class_t
```

```
# 어느지역이 범죄가 제일 많이 발생하는지 알아보기
```

```
# axis=1 은 같은 행별로 계산하는 옵션
```

```
crime_class_t_sum = np.sum(crime_class_t.groupby('지역').sum(), axis=1)  
crime_class_t_sum
```

2. 데이터 정제

```
# 데이터프레임 형식으로 인덱스 재설정  
# reset_index(): 기존 행 인덱스를 제거하고 인덱스를 데이터 열 추가  
# 지역이라는 인덱스를 열로 변환  
crime_class_t_g = pd.DataFrame(crime_class_t_sum).reset_index()  
  
# 컬럼명 변경  
# inplace = True: 기존 데이터프레임에 변경된 설정으로 덮어쓰겠다는 의미  
crime_class_t_g.rename(columns = {0: '총 합계'}, inplace = True)  
  
# 총 합계를 기준으로 내림차순 정렬  
# ascending=False: 내림차순, True는 오름차순  
crime_class_t_g.sort_values(by='총 합계', ascending=False, inplace = True)  
  
crime_class_t_g
```

3. 데이터 시각화

```
plt.figure(figsize=(10, 10))  
sns.barplot(x='지역', y='총 합계', data=crime_class_t_g)  
sns.despine(left=True) # 위, 오른쪽 축 숨기기  
# 한글깨짐 방지, 마이너스기호 깨짐 방지, 하얀배경에 선으로 보기편한 스타일  
sns.set_theme(font = 'Malgun Gothic', rc = {'axes.unicode_minus' : False}, style = 'whitegrid')  
plt.title('범죄가 많이 일어나는 지역', fontsize=14, fontfamily='Malgun Gothic', fontweight='bold', position=(0, 0))  
plt.xticks(rotation=45, fontsize=10)  
plt.savefig('data/crime/분석 그래프/3. 범죄가 많이 일어나는 지역.png')  
plt.show()
```

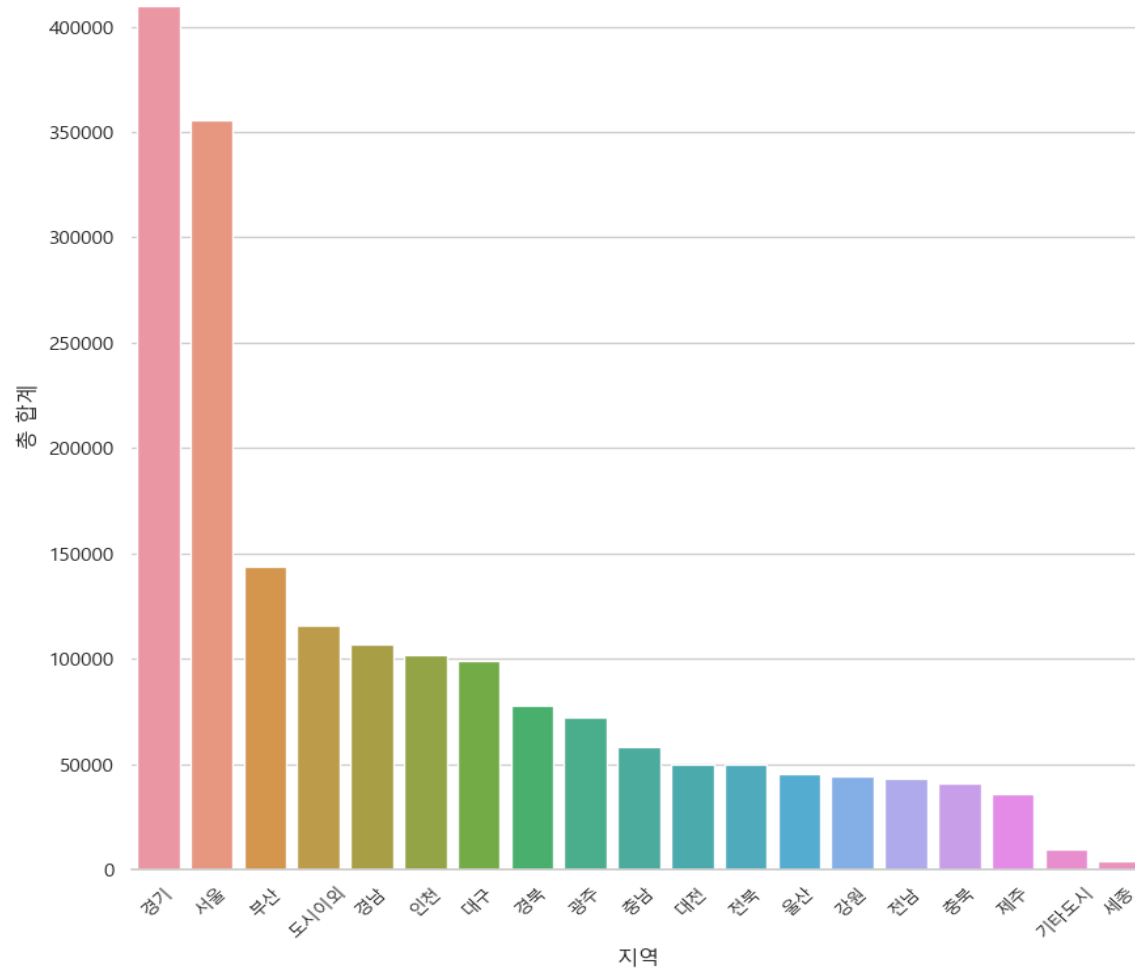
3. 범죄가 많이 일어나는 지역

	지역	총 합계
1	경기	409974
10	서울	355341
9	부산	143713
8	도시이외	115794
2	경남	107025
13	인천	101931
6	대구	98758
3	경북	77609
4	광주	71919
17	충남	58416
7	대전	49577
15	전북	49558
12	울산	45141
0	강원	44178
14	전남	43039
18	충북	40730
16	제주	35606
5	기타도시	9688
11	세종	3660

- ▶ 분석목표: 지역별로 제일 많이 일어나는 범죄는 어떤 것들이 있는지 알아보기
- ▶ 활용한 데이터
 - 경찰청_범죄 발생 지역별 통계_20151231.csv
- ▶ 표 생성 과정
- ▶ - 범죄 대분류를 기준으로 그룹화와 전치를 시킨 후 합계를 구하고 높은 순으로 재정렬 시킨다.
- ▶ 표 분석 결과
 - ▶ - 범죄가 제일 많이 일어나는 지역 1순위: 경기
 - ▶ - 2순위는 서울, 3순위는 부산이 제일 많이 일어난다.

3. 범위가 많이 일어나는 지역

범위가 많이 일어나는 지역



그래프 분석 결과: 범위가 많이 일어나는 지역은 경기와 서울이며 그 다음은 부산이 제일 많이 일어납니다.

4. 범죄가 많이 일어나는 장소

1. 데이터 분석

```
# 범죄대분류를 기준으로 행, 열 위치 교체와 지역별 합계  
crime_class_t1 = data1.groupby('범죄대분류').sum().T  
crime_class_t1
```

```
# 기존에 있는 범죄 대분류의 행의 값을 새로운 열로 생성  
crime_class_t1 = crime_class_t1.reset_index()  
  
# 컬럼명 변경  
# inplace = True: 기존 데이터프레임에 변경된 설정으로 덮어쓰겠다는 의미  
crime_class_t1.rename(columns = {'index' : '장소'}, inplace = True)  
  
crime_class_t1
```

```
# 어느장소에 범죄가 제일 많이 발생하는지 알아보기  
# axis=1 은 같은 행별로 계산하는 옵션  
crime_class_t1_sum = np.sum(crime_class_t1.groupby('장소').sum(), axis=1)  
crime_class_t1_sum.sort_values(ascending=False, inplace = True) # 높은순으로 정렬  
crime_class_t1_sum
```

2. 데이터 정제

```
# 데이터프레임 형식으로 인덱스 재설정  
# reset_index(): 기존 행 인덱스를 제거하고 인덱스를 데이터 열 추가  
# 장소라는 인덱스를 열로 변환  
crime_class_t1_g = pd.DataFrame(crime_class_t1_sum).reset_index()  
  
# 컬럼명 변경  
# inplace = True: 기존 데이터프레임에 변경된 설정으로 덮어쓰겠다는 의미  
crime_class_t1_g.rename(columns = {0: '총 합계'}, inplace = True)  
  
# 총 합계를 기준으로 내림차순 정렬  
# ascending=False: 내림차순, True는 오름차순  
crime_class_t1_g.sort_values(by='총 합계', ascending=False, inplace = True)  
  
crime_class_t1_g
```

3. 데이터 시각화

```
plt.figure(figsize=(15, 15))  
sns.barplot(x='총 합계', y='장소', data=crime_class_t1_g)  
sns.despine(left=True) # 위, 오른쪽 축 숨기기  
# 한글깨짐 방지, 마이너스기호 깨짐 방지, 하단배경에 선으로 보기편한 스타일  
sns.set_theme(font='Malgun Gothic', rc = {'axes.unicode_minus' : False}, style='whitegrid')  
plt.title('범죄가 많이 일어나는 장소', fontsize=14, fontfamily='Malgun Gothic', fontweight='bold', position=(0, 0))  
plt.xticks(rotation=0, fontsize=10)  
plt.savefig('data/crime/분석 그래프/4. 범죄가 많이 일어나는 장소.png')  
plt.show()
```

4. 범죄가 많이 일어나는 장소

	장소	총 합계
0	노상	501535
1	기타	393258
2	아파트_연립다세대	108301
3	유흥접객업소	79990
4	사무실	74327
5	단독주택	69742
6	상점	35593
7	주차장	27321
8	해상	21215
9	숙박업소_목욕탕	17055
10	편의점	16435
11	기타교통수단내	15778
12	슈퍼마켓	14738
13	시장_노점	11355
14	역_대합실	10367
15	고속도로	10233
16	산야	9127
17	의료기관	8782
18	유원지	6521
19	학교	6306
20	공장	6305
21	금융기관	5977
22	공사장_광산	5813
23	대형할인매장	4138
24	창고	3614
25	지하철	3589
26	피씨방	3585
27	공중화장실	3528
28	백화점	3047
29	종교기관	1828
30	은행장	1657
31	부대	1026
32	공지	218
33	구금장소	129

▶ 분석목표: 범죄가 제일 많이 일어나는 장소는 어떤 곳들이 있는지 알아보기

▶ 활용한 데이터

- 경찰청_범죄 발생 장소별 통계_20221231.csv

▶ 표 생성 과정

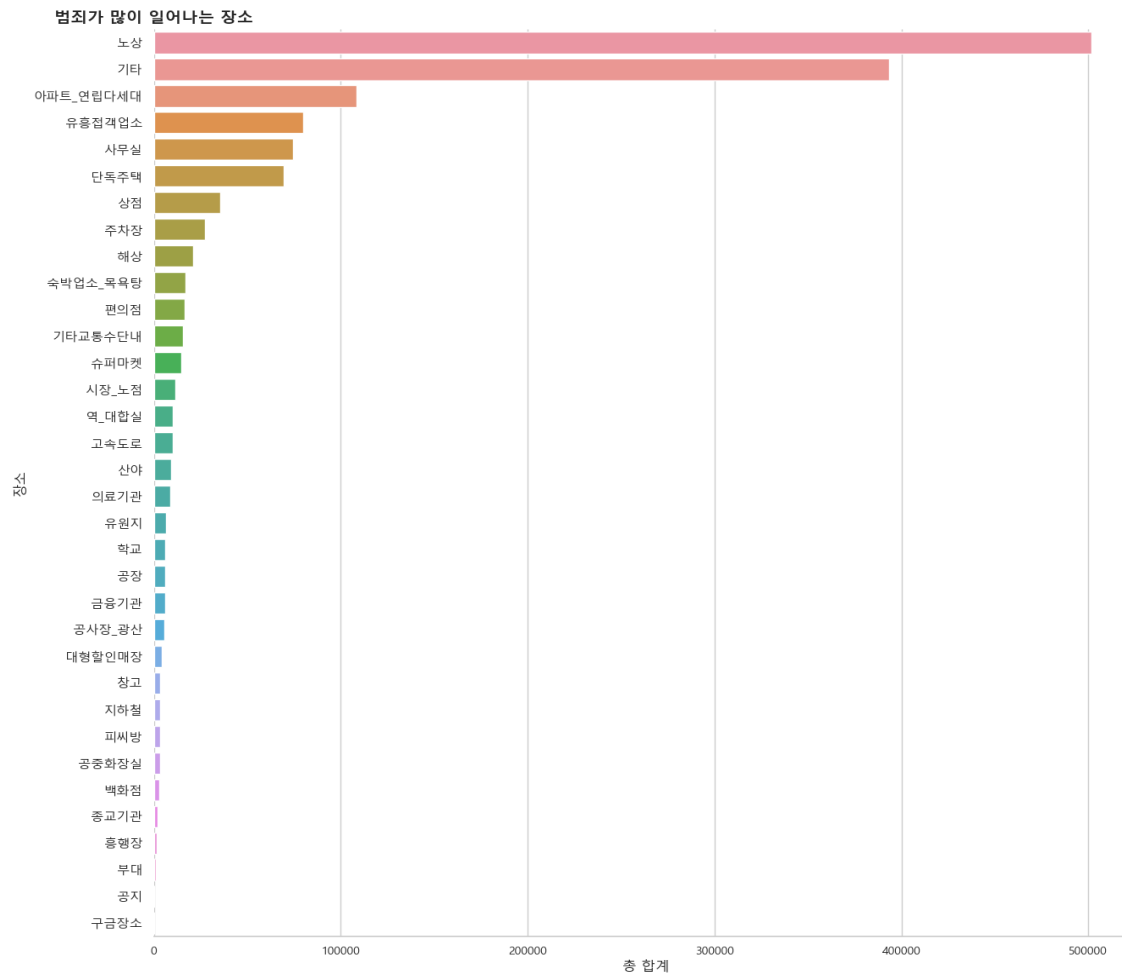
▶ - 범죄 대분류를 기준으로 그룹화와 전치를 시킨 후 합계를 구하고 높은 순으로 재정렬 시킨다.

▶ 표 분석 결과

▶ - 범죄가 제일 많이 일어나는 장소 1순위: 노상

▶ - 2순위는 기타, 3순위는 아파트에서 제일 많이 일어난다.

4. 범죄가 많이 일어나는 장소



그래프 분석 결과: 범죄가 많이 일어나는 장소는 노상(길거리)와 기타 등등 그 다음은 아파트와 유흥접객업소, 사무실, 단독주택에서 많이 일어납니다.

5. 범죄가 많이 일어나는 시간대와 요일

1. 데이터 분석(시간대, 요일)

```
# 범죄대분류를 기준으로 행, 열 위치 교체와 지역별 합계  
crime_class_t2 = data2.groupby('범죄대분류').sum().T  
crime_class_t2
```

```
# 기존에 있는 범죄 대분류의 행의 값을 새로운 열로 생성  
crime_class_t2 = crime_class_t2.reset_index()  
  
# 컬럼명 변경  
# inplace = True: 기존 데이터프레임에 변경된 설정으로 덮어쓰겠다는 의미  
crime_class_t2.rename(columns = {'index' : '시간대 및 요일'}, inplace = True)  
  
crime_class_t2
```

```
# 시간대만 남겨두기  
crime_class_t2_2 = crime_class_t2.drop(crime_class_t2.index[8:], axis=0)  
crime_class_t2_2
```

```
# 시간대 및 요일(시간대)에 범죄가 제일 많이 발생하는지 알아보기  
# axis=1 은 같은 행별로 계산하는 옵션  
crime_class_t2_sum = np.sum(crime_class_t2_2.groupby('시간대 및 요일').sum(), axis=1)  
crime_class_t2_sum.sort_values(ascending=False, inplace = True) # 높은순으로 정렬  
crime_class_t2_sum
```

```
# 요일만 남겨두기  
crime_class_t2_1 = crime_class_t2.drop(crime_class_t2.index[0:8], axis=0)  
crime_class_t2_1
```

```
# 시간대 및 요일(요일)에 범죄가 제일 많이 발생하는지 알아보기  
# axis=1 은 같은 행별로 계산하는 옵션  
crime_class_t2_sum = np.sum(crime_class_t2_1.groupby('시간대 및 요일').sum(), axis=1)  
crime_class_t2_sum.sort_values(ascending=False, inplace = True) # 높은순으로 정렬  
crime_class_t2_sum
```

3. 데이터 시각화 (시간대, 요일)

```
# explode: 부채꼴이 파이 차트의 중심에서 벗어나는 정도  
explode = [0.01, 0.02, 0.03, 0.04, 0.05,  
           0.06, 0.07, 0.08]  
  
# wedges: 테두리를 추가한다.  
wedges = ({'edgecolor': 'w', 'width': 0.7, 'linewidth': 4})  
  
# startangle: 파이 차트의 각도를 돌린다.  
plt.figure(figsize=(20, 20))  
plt.pie(crime_class_t2_g['총 합계'], labels=crime_class_t2_g['시간대'], autopct='%1.2f%%', wedges=wedges,  
        pctdistance=0.7, labeldistance=1.10, explode=explode, textprops={'size': 20, 'shadow': True, 'radius': 0.95},  
        startangle=90)  
  
plt.title('범죄가 많이 일어나는 시간대', fontsize=20, fontfamily='Malgun Gothic', fontweight='bold', position=(0, 0))  
plt.legend(fontsize=15, loc='best')  
plt.savefig('data/crime/분석 그래프/5.1. 범죄가 많이 일어나는 시간대(파이차트).png')  
plt.show()
```

```
# explode: 부채꼴이 파이 차트의 중심에서 벗어나는 정도  
explode = [0.01, 0.02, 0.03, 0.04, 0.05,  
           0.06, 0.07, 0.08]  
  
# wedges: 테두리를 추가한다.  
wedges = ({'edgecolor': 'w', 'width': 0.7, 'linewidth': 4})  
  
# startangle: 파이 차트의 각도를 돌린다.  
plt.figure(figsize=(20, 20))  
plt.pie(crime_class_t2_g_1['총 합계'], labels=crime_class_t2_g_1['요일'], autopct='%1.2f%%', wedges=wedges,  
        pctdistance=0.7, labeldistance=1.10, explode=explode, textprops={'size': 20, 'shadow': True, 'radius': 0.95},  
        startangle=90)  
  
plt.title('범죄가 많이 일어나는 요일', fontsize=20, fontfamily='Malgun Gothic', fontweight='bold', position=(0, 0))  
plt.legend(fontsize=20, loc='best')  
plt.savefig('data/crime/분석 그래프/5.2. 범죄가 많이 일어나는 요일(파이차트).png')  
plt.show()
```

2. 데이터 정제(시간대, 요일)

```
# 데이터프레임 형식으로 인덱스 재설정  
# reset_index(): 기존 행 인덱스를 제거하고 인덱스를 데이터 열 추가  
# 장소라는 인덱스를 열로 변환  
crime_class_t2_g_1 = pd.DataFrame(crime_class_t2_sum).reset_index()  
  
# 컬럼명 변경  
# inplace = True: 기존 데이터프레임에 변경된 설정으로 덮어쓰겠다는 의미  
crime_class_t2_g_1.rename(columns = {'시간대 및 요일' : '요일', 0: '총 합계'}, inplace = True)  
  
# 총 합계를 기준으로 내림차순 정렬  
# ascending=False: 내림차순, True는 오름차순  
crime_class_t2_g_1.sort_values(by='총 합계', ascending=False, inplace = True)  
  
crime_class_t2_g_1
```

```
# 데이터프레임 형식으로 인덱스 재설정  
# reset_index(): 기존 행 인덱스를 제거하고 인덱스를 데이터 열 추가  
# 장소라는 인덱스를 열로 변환  
crime_class_t2_g = pd.DataFrame(crime_class_t2_sum).reset_index()  
  
# 컬럼명 변경  
# inplace = True: 기존 데이터프레임에 변경된 설정으로 덮어쓰겠다는 의미  
crime_class_t2_g.rename(columns = {'시간대 및 요일' : '시간대', 0: '총 합계'}, inplace = True)  
  
# 총 합계를 기준으로 내림차순 정렬  
# ascending=False: 내림차순, True는 오름차순  
crime_class_t2_g.sort_values(by='총 합계', ascending=False, inplace = True)  
  
crime_class_t2_g
```

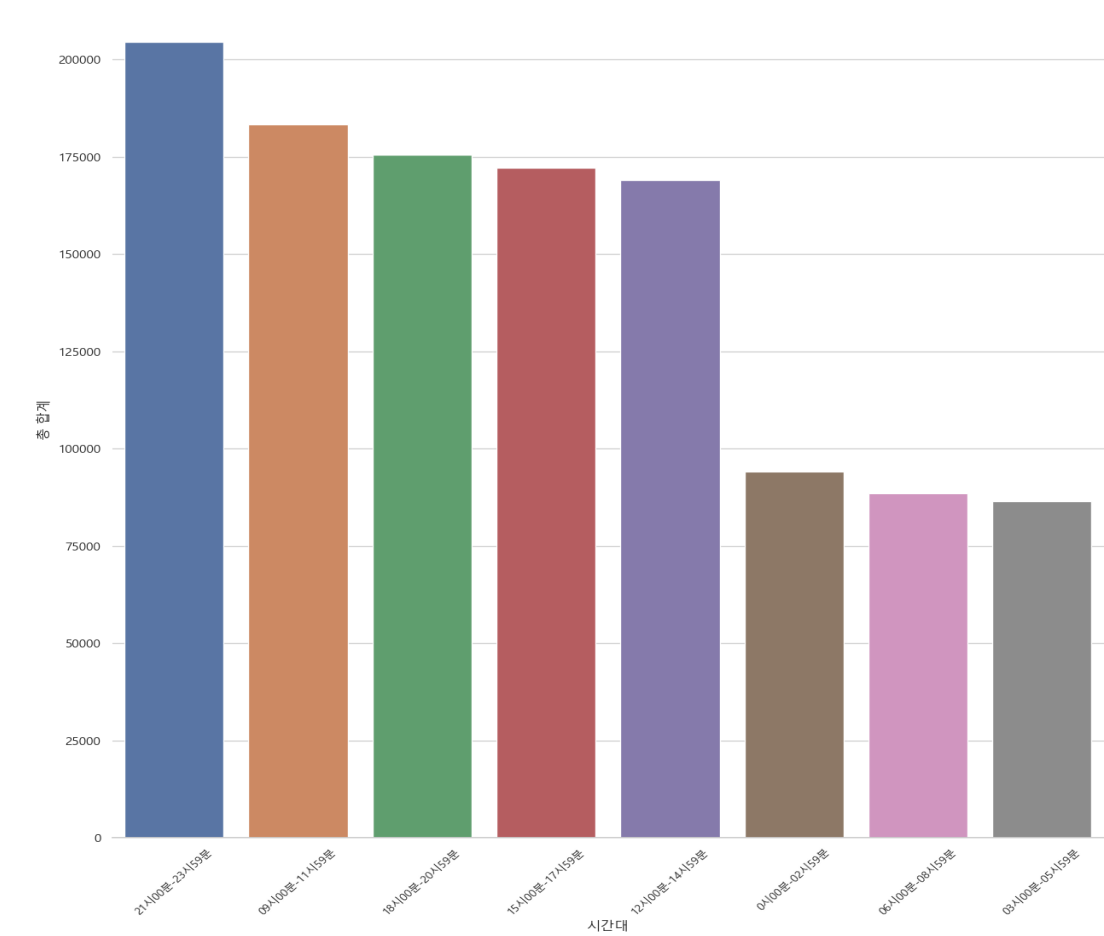

5-1. 범죄가 많이 일어나는 시간대

	시간대	총 합계
0	21시00분-23시59분	204508
1	09시00분-11시59분	183235
2	18시00분-20시59분	175503
3	15시00분-17시59분	172107
4	12시00분-14시59분	169090
5	0시00분-02시59분	94095
6	06시00분-08시59분	88488
7	03시00분-05시59분	86471

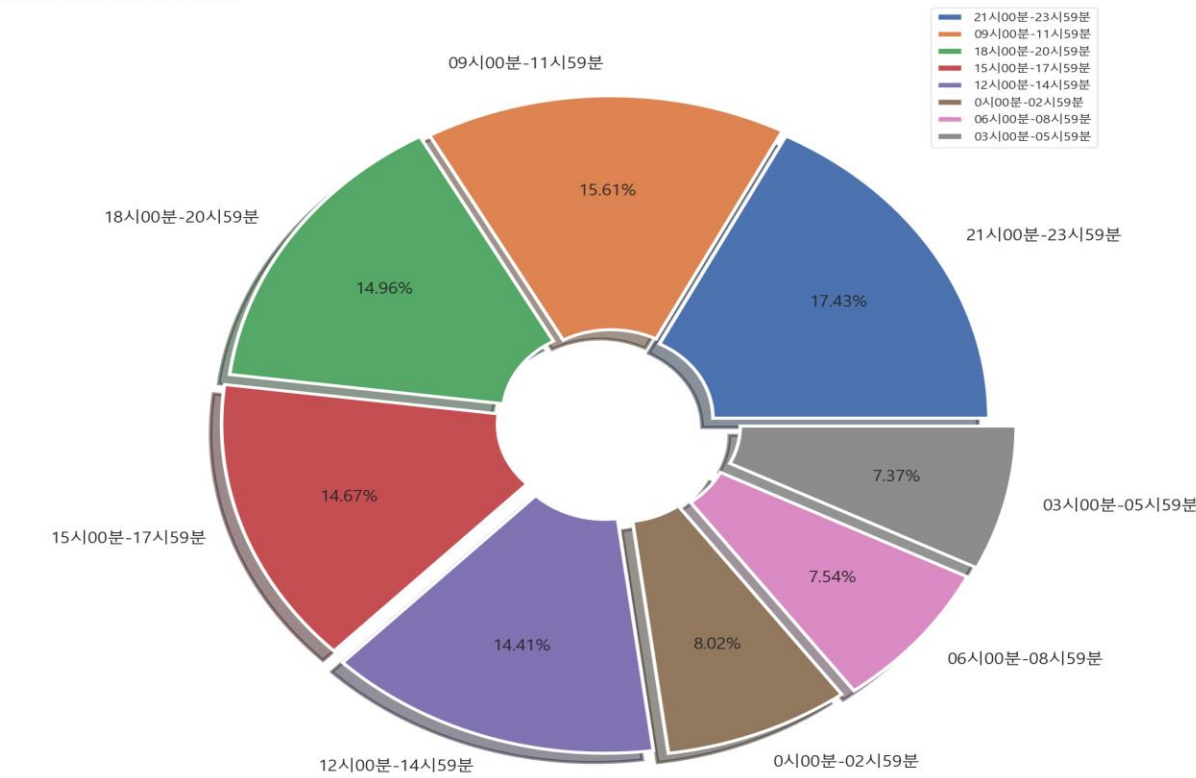
- ▶ 분석목표: 범죄가 제일 많이 일어나는 시간대는 어떤 시간대 인지 알아보기
- ▶ 활용한 데이터
 - 경찰청_범죄 발생 시간대 및 요일_20191231.csv
- ▶ 표 생성 과정
 - 범죄 대분류를 기준으로 그룹화와 전치를 시킨 후 합계를 구하고 필요 없는 행을 삭제 후 높은 순으로 재정렬 시킨다.
- ▶ 표 분석 결과
 - 범죄가 제일 많이 일어나는 장소 1순위: 저녁 9시 ~ 12시전
 - 2순위는 아침 9시 ~ 12시전, 3순위는 저녁6시 ~ 9시에 범죄가 제일 많이 일어난다.

5-1. 범죄가 많이 일어나는 시간대

범죄가 많이 일어나는 시간대



범죄가 많이 일어나는 시간대



그래프 분석결과: 범죄가 많이 일어나는 시간대는 저녁 9시부터 12까지 자주 일어납니다.

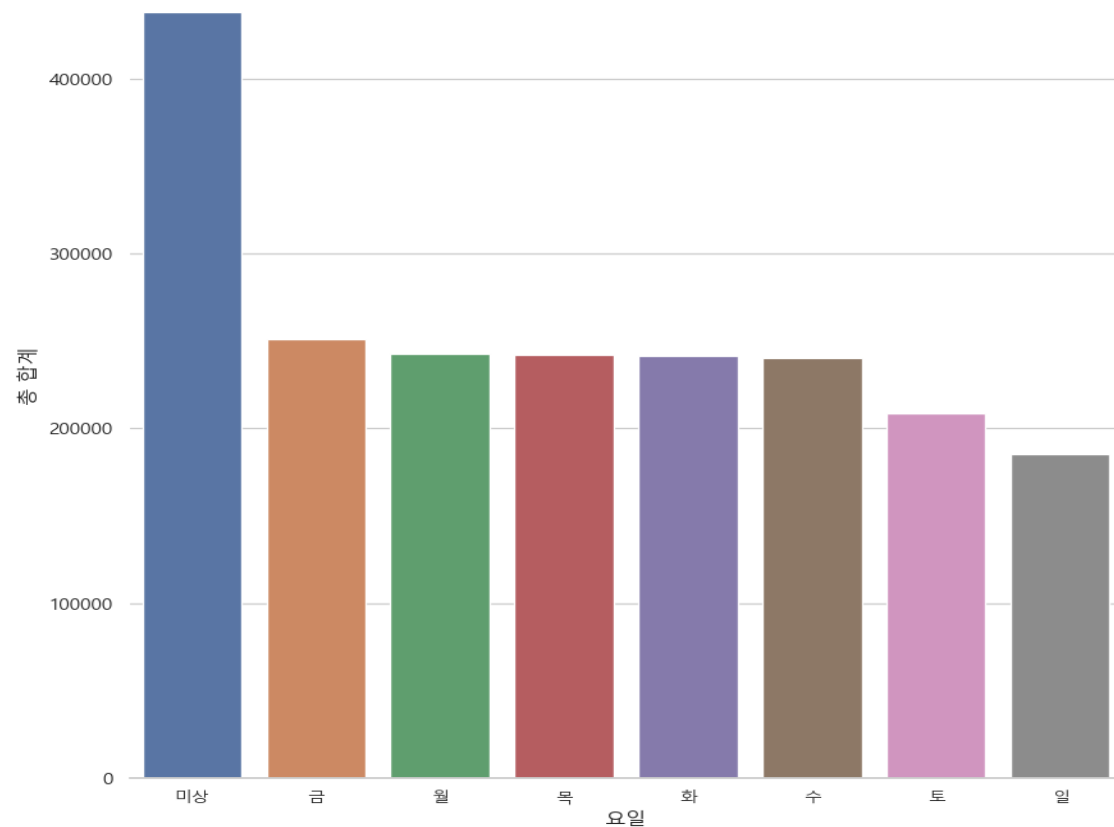
5-2. 범죄가 많이 일어나는 요일

	요일	총 합계
0	미상	438409
1	금	251352
2	월	242481
3	목	241974
4	화	241591
5	수	240590
6	토	208398
7	일	185520

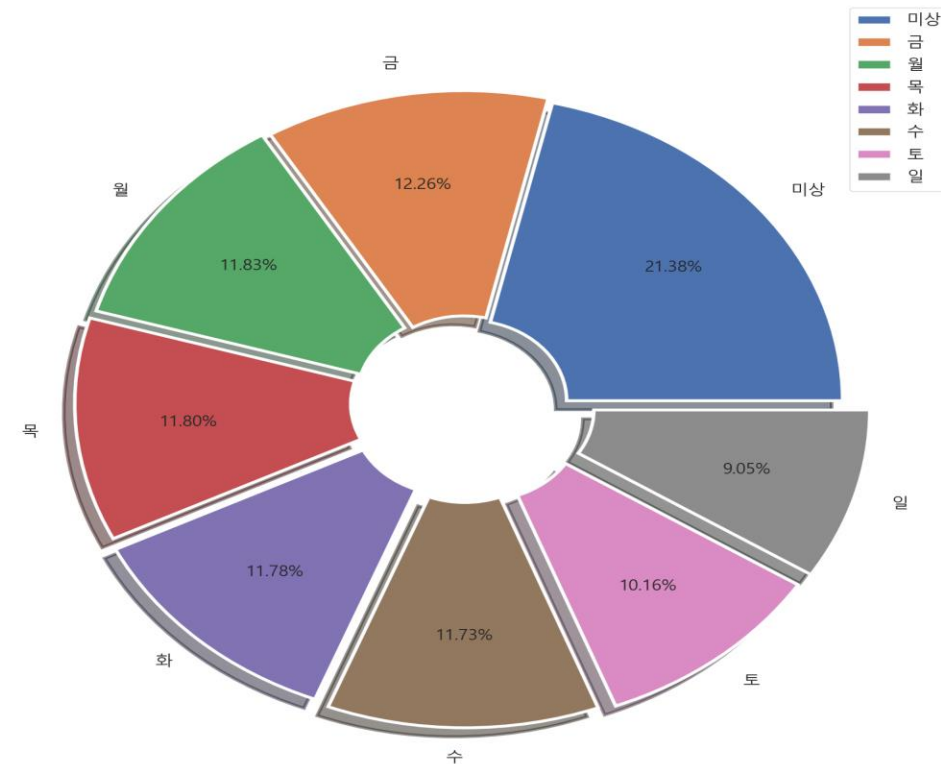
- ▶ 분석목표: 범죄가 제일 많이 일어나는 요일은 어떤 요일 인지 알아보기
- ▶ 활용한 데이터
 - 경찰청_범죄 발생 시간대 및 요일_20191231.csv
- ▶ 표 생성 과정
- ▶ - 범죄 대분류를 기준으로 그룹화와 전치를 시킨 후 합계를 구하고 필요 없는 행을 삭제 후 높은 순으로 재정렬 시킨다.
- ▶ 표 분석 결과
- ▶ - 범죄가 제일 많이 일어나는 장소 1순위:미상
- ▶ - 2순위는 금요일, 3순위는 월요일에 범죄가 제일 많이 일어난다.

5-2. 범죄가 많이 일어나는 요일

범죄가 많이 일어나는 요일



범죄가 많이 일어나는 요일



그래프 분석 결과: 범죄가 많이 일어나는 요일은 금요일이 제일 많이 일어납니다.

6. 범죄 발생으로부터 인지하는데 까지 걸리는 시간

1. 데이터 분석

```
# 범죄대분류를 기준으로 행, 열 위치 교체와 지역별 합계  
crime_class_t3 = data3.groupby('범죄대분류').sum().T  
crime_class_t3
```

```
# 기존에 있는 범죄 대분류의 행의 값을 새로운 열로 생성  
crime_class_t3 = crime_class_t3.reset_index()
```

```
# 컬럼명 변경  
# inplace = True: 기존 데이터프레임에 변경된 설정으로 덮어쓰겠다는 의미  
crime_class_t3.rename(columns = {'index' : '범죄인지기간'}, inplace = True)
```

```
crime_class_t3
```

```
# 범죄를인지하는기간중 어떤 기간이 제일높은지 알아보기  
# axis=1 은 같은 행별로 계산하는 옵션  
crime_class_t3_sum = np.sum(crime_class_t3.groupby('범죄인지기간').sum(), axis=1)  
crime_class_t3_sum.sort_values(ascending=False, inplace = True) # 높은순으로 정렬  
crime_class_t3_sum
```

2. 데이터 정제

```
# 데이터프레임 형식으로 인덱스 재설정  
# reset_index(): 기존 행 인덱스를 제거하고 인덱스를 데이터 열 추가  
# 장소라는 인덱스를 열로 변환  
crime_class_t3_g_1 = pd.DataFrame(crime_class_t3_sum).reset_index()
```

```
# 컬럼명 변경  
# inplace = True: 기존 데이터프레임에 변경된 설정으로 덮어쓰겠다는 의미  
crime_class_t3_g_1.rename(columns = {0: '총 합계'}, inplace = True)
```

```
# 총 합계를 기준으로 내림차순 정렬  
# ascending=False: 내림차순, True는 오름차순  
crime_class_t3_g_1.sort_values(by='총 합계', ascending=False, inplace = True)  
  
crime_class_t3_g_1
```

3. 데이터 시각화

```
# explode: 부채꼴이 파이 차트의 중심에서 벗어나는 정도  
explode = [0.01, 0.02, 0.03, 0.04, 0.05,  
           0.06, 0.07, 0.08, 0.09, 0.10,  
           0.11]  
  
# wedgeprops: 테두리를 추가한다.  
wedgeprops = ({'edgecolor' : 'w', 'width' : 0.7, "linewidth":4})  
  
# startangle: 파이 차트의 각도를 돌린다.  
plt.figure(figsize=(20, 20))  
plt.pie(crime_class_t3_g_1['총 합계'], labels=crime_class_t3_g_1['범죄인지기간'], autopct='%1.2f%%', wedgeprops=wedgeprops,  
        pctdistance=0.7, labeldistance=1.10, explode=explode, textprops={'size' : 20}, shadow=True, radius=0.95  
        , startangle=90  
        )  
plt.title('범죄 발생부터 인지까지 기간중 제일 높은 것', fontsize=15, fontfamily='Malgun Gothic', fontweight='bold', position=(0, 0))  
plt.legend(fontsize=15, loc='best')  
plt.savefig('data/crime/분석 그래프/6. 범죄 발생으로부터 인지하는데 까지 걸리는 시간(파이차트).png')  
plt.show()
```

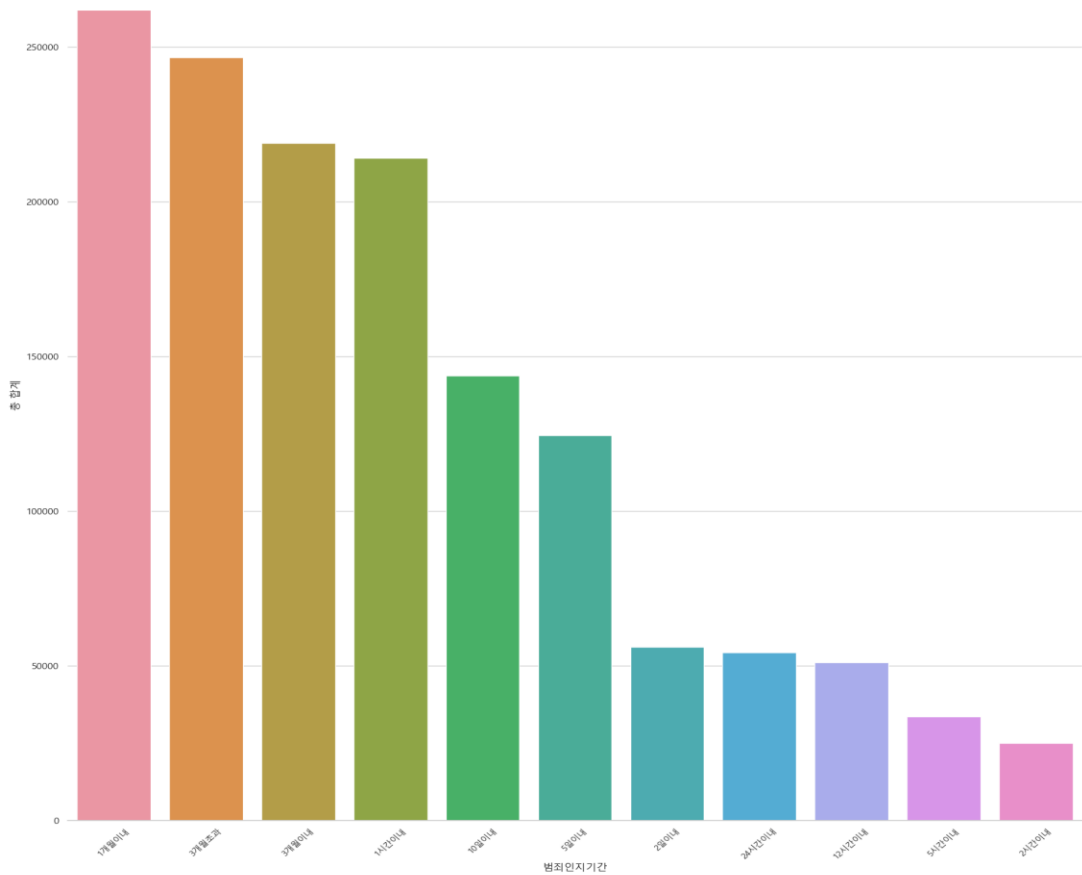
6. 범죄 발생으로부터 인지하는데 까지 걸리는 기간

	범죄인지기간	총 합계
0	1개월이내	261908
1	3개월초과	246699
2	3개월이내	218945
3	1시간이내	214147
4	10일이내	143663
5	5일이내	124467
6	2일이내	56068
7	24시간이내	54224
8	12시간이내	51006
9	5시간이내	33654
10	2시간이내	25045

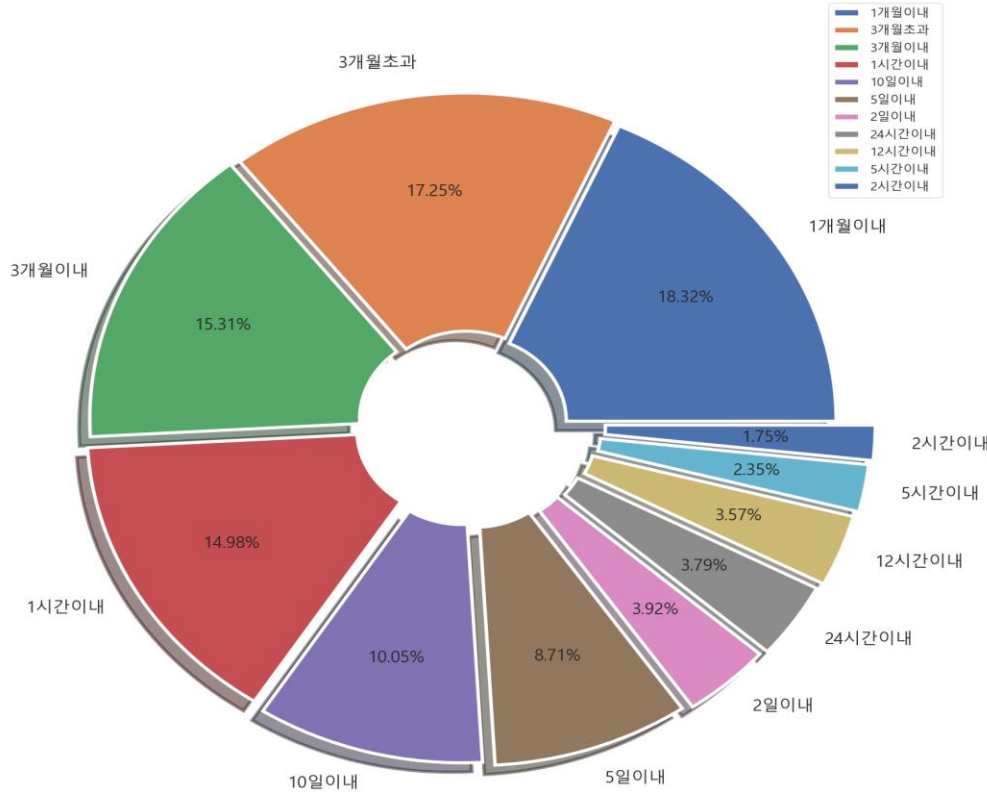
- ▶ 분석목표: 범죄발생후 인지까지 걸리는 기간이 제일 많은 기간 알아보기
- ▶ 활용한 데이터
 - 경찰청_범죄 발생부터 인지까지의 기간_20211231.CSV
- ▶ 표 생성 과정
- ▶ - 범죄 대분류를 기준으로 그룹화와 전처리를 시킨 후 합계를 구하고 높은 순으로 재정렬 시킨다.
- ▶ 표 분석 결과
 - ▶ - 제일 많은 기간 1순위: 1개월이내
 - ▶ - 2순위는 3개월초과, 3순위는 3개월이내가 제일 많이 일어난다.

6. 범죄 발생으로부터 인지하는데 까지 걸리는 기간

범죄 발생부터 인지까지 기간중 제일 높은 것



범죄 발생부터 인지까지 기간중 제일 높은 것



그래프 분석 결과: 범죄 발생부터 인지하는데 걸리는 기간은 1개월 이내와 3개월 초과가 제일 많은 기간으로 나타났습니다.

7. 전국 범죄 발생과 검거현황

1. 데이터 분석

```
# 범죄대분류를 기준으로 행, 열 위치 교체와 지역별 합계  
crime_class_t4 = data4.groupby('범죄대분류').sum().T  
crime_class_t4
```

```
# 기존에 있는 범죄 대분류의 행의 값을 새로운 열로 생성  
crime_class_t4 = crime_class_t4.reset_index()
```

```
# 컬럼명 변경  
# inplace = True: 기존 데이터프레임에 변경된 설정으로 덮어쓰겠다는 의미  
crime_class_t4.rename(columns = {'index' : '범죄 발생 및 검거'}, inplace = True)
```

```
crime_class_t4
```

```
# 범죄 발생 및 검거중 어떤 것이 제일높은지 알아보기  
# axis=1 은 같은 행별로 계산하는 옵션  
crime_class_t4_sum = np.sum(crime_class_t4.groupby('범죄 발생 및 검거').sum(), axis=1)  
crime_class_t4_sum.sort_values(ascending=False, inplace = True) # 높은순으로 정렬  
crime_class_t4_sum
```

2. 데이터 정제

```
# <데이터프레임 형식으로 인덱스 재설정>  
# reset_index(): 기존 행 인덱스를 제거하고 인덱스를 데이터 열 추가  
# 장소라는 인덱스를 열로 변환  
crime_class_t4_g_1 = pd.DataFrame(crime_class_t4_sum).reset_index()  
  
# 컬럼명 변경  
# inplace = True: 기존 데이터프레임에 변경된 설정으로 덮어쓰겠다는 의미  
crime_class_t4_g_1.rename(columns = {0: '총 합계'}, inplace = True)  
  
# 총 합계를 기준으로 내림차순 정렬  
# ascending=False: 내림차순, True는 오름차순  
crime_class_t4_g_1.sort_values(by='총 합계', ascending=False, inplace = True)  
  
crime_class_t4_g_1
```

3. 데이터 시각화

```
# pie는 labels이 있어야 출력된다.  
labels = ['발생', '검거', '검거인원(남)', '검거인원(여)', '불상', '법인체']  
  
# explode: 부채꼴이 파이 차트의 중심에서 벗어나는 정도  
explode = [0.01, 0.02, 0.03, 0.04, 0.8, 0.8]  
  
# wedgeprops: 테두리를 추가한다.  
wedgeprops = ({'edgecolor': 'w', 'width': 0.7, "linewidth":4})  
  
# startangle: 파이 차트의 각도를 돌린다.  
plt.figure(figsize=(20, 10))  
plt.pie(crime_class_t4_sum, labels=labels, autopct='%1.2f%%', wedgeprops=wedgeprops, pctdistance=0.8, labeldistance=1.15,  
        explode=explode, textprops={'size': 20}, shadow=True, radius=0.95  
        # startangle=50  
        )  
plt.title('전국 범죄 발생과 검거현황', fontsize=20, fontfamily='Malgun Gothic', fontweight='bold', position=(0, 0))  
plt.legend(fontsize=10, loc='best')  
plt.savefig('data/crime/분석 그래프/7. 전국 범죄 발생과 검거현황(파이차트).png')  
plt.show()
```


7. 전국 범죄 발생과 검거현황

	범죄 발생 및 검거	총 합계
0	발생	1429826
1	검거	1136665
2	검거인원(남)	1002380
3	검거인원(여)	278825
4	불상	79200
5	법인체	10317

- ▶ 분석목표: 전국 범죄 발생과 검거현황에 대해 알아보기

- ▶ 활용한 데이터

- 경찰청_전국 범죄 발생 및 검거 현황_20211231.csv

- ▶ 표 생성 과정

- ▶ - 범죄 대분류를 기준으로 그룹화와 전치를 시킨 후 합계를 구하고 높은 순으로 재정렬 시킨다.

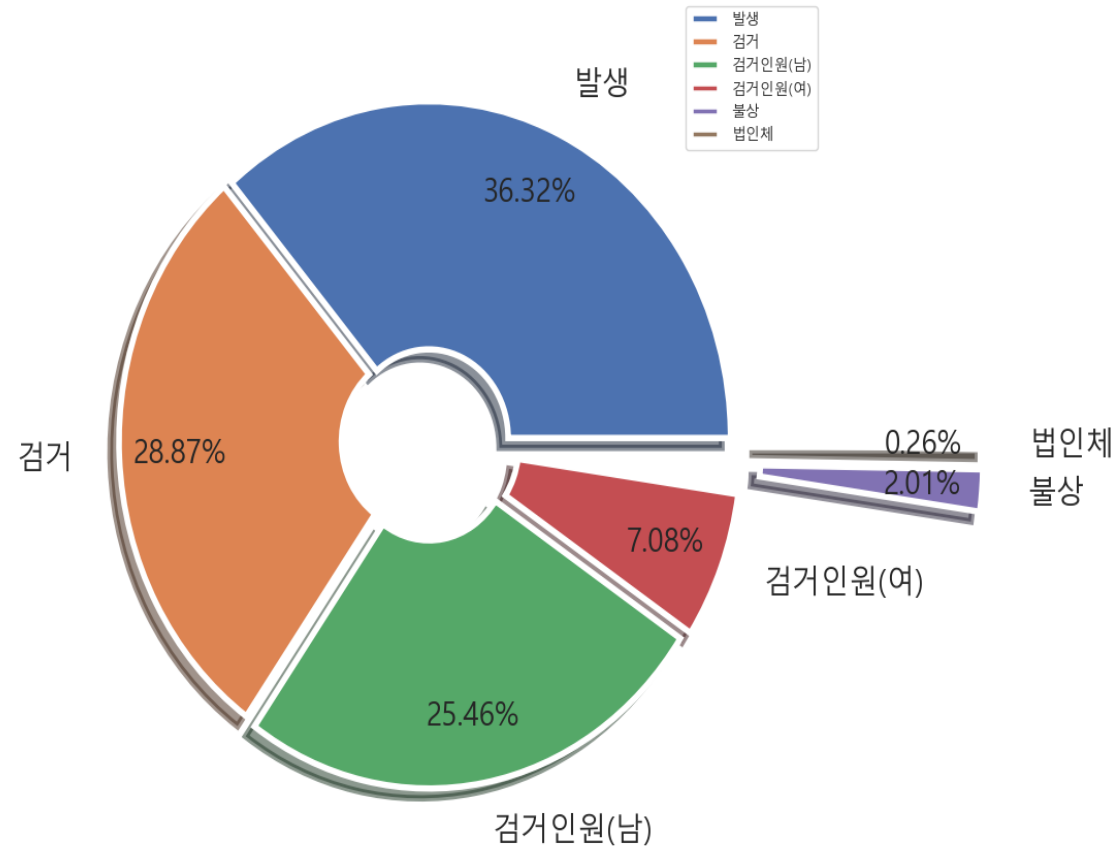
- ▶ 표 분석 결과

- ▶ - 전국 범죄 발생과 검거현황에 1순위: 범죄 발생

- ▶ - 2순위는 검거, 3순위는 검거인원(남)이 제일 많다.

7. 전국 범죄 발생과 검거현황

전국 범죄 발생과 검거현황



그래프 분석 결과: 전국적으로는 범죄 발생은 36%정도로 1순위이며 검거율은 28%, 검거되는 인원 중 남성 25%, 여성 7%정도로 남성이 범죄를 많이 일으킨다는 것을 알 수 있습니다.

8. 지역별 인구수에 따른 범죄 발생율

1. 데이터 분석

```
# 범죄 데이터프레임과 합칠 행 생성
```

```
def region(region_g):
```

```
    if '서울' in region_g:
```

```
        return '서울'
```

```
    elif '부산' in region_g:
```

```
        return '부산'
```

```
    elif '대구' in region_g:
```

```
        return '대구'
```

```
    elif '인천' in region_g:
```

```
        return '인천'
```

```
    elif '광주' in region_g:
```

```
        return '광주'
```

```
    elif '대전' in region_g:
```

```
        return '대전'
```

```
    elif '울산' in region_g:
```

```
        return '울산'
```

```
    elif '세종' in region_g:
```

```
        return '세종'
```

```
    elif '경기' in region_g:
```

```
        return '경기'
```

```
    elif '강원' in region_g:
```

```
        return '강원'
```

```
    elif '충청북도' in region_g:
```

```
        return '충북'
```

```
    elif '충청남도' in region_g:
```

```
        return '충남'
```

```
    elif '전라북도' in region_g:
```

```
        return '전북'
```

```
    elif '전라남도' in region_g:
```

```
        return '전남'
```

```
    elif '경상북도' in region_g:
```

```
        return '경북'
```

```
    elif '경상남도' in region_g:
```

```
        return '경남'
```

```
    elif '제주' in region_g:
```

```
        return '제주'
```

```
data5['지역'] = data5['행정구역'].apply(region)
data5
```

```
crime_class_t_gg = crime_class_t_g.rename(columns = {'총 합계' : '범죄 총 합계'})
crime_class_t_gg
```

```
# 전체인구와 범죄 데이터 합치기
```

```
national_crime = pd.merge(data5, crime_class_t_gg, how='outer')
national_crime
```

```
# 결측값(null) 제거하기
```

```
national_crime = national_crime.dropna(axis=0)
national_crime
```

```
# 범죄발생비율구하기
```

```
national_crime['범죄발생비율'] = national_crime['범죄 총 합계']/national_crime['2023년09월_총인구수']
national_crime
```

```
# 범죄율 데이터 프레임
```

```
national_crime1 = national_crime[['행정구역', '범죄발생비율']]
national_crime1
```

```
national_crime1 = national_crime1.rename(columns = {'행정구역' : '지역'})
national_crime1
```

2. 데이터 정제

```
national_crime1 = national_crime1.sort_values(by='범죄발생비율', ascending=False)
national_crime1
```

3. 데이터 시각화

```
# explode: 부채꼴이 파이 차트의 중심에서 벗어나는 정도
```

```
explode = [0.01, 0.02, 0.03, 0.04, 0.05,
           0.06, 0.07, 0.08, 0.09, 0.10,
           0.11, 0.12, 0.13, 0.14, 0.15,
           0.16, 0.17,]
```

```
# wedgeprops: 테두리를 추가한다.
```

```
wedgeprops = ({'edgecolor' : 'w', 'width' : 0.7, "linewidth":4})
```

```
# startangle: 파이 차트의 각도를 돌린다.
```

```
plt.figure(figsize=(20, 20))
```

```
plt.pie(national_crime1['범죄발생비율'], labels=national_crime1['지역'], autopct='%1.2f%%', wedgeprops=wedgeprops,
        pctdistance=0.7, labeldistance=1.10, explode=explode, textprops={'size' : 20}, shadow=True, radius=0.95
```

```
#, startangle=90
```

```
)
plt.title('지역별 인구수에 따른 범죄 발생율', fontsize=20, fontfamily='Malgun Gothic', fontweight='bold', position=(0, 0))
```

```
plt.legend(fontsize=10, loc='best')
```

```
plt.savefig('data/crime/분석 그래프/8. 지역별 인구수에 따른 범죄 발생율(파이차트).png')
```

```
plt.show()
```

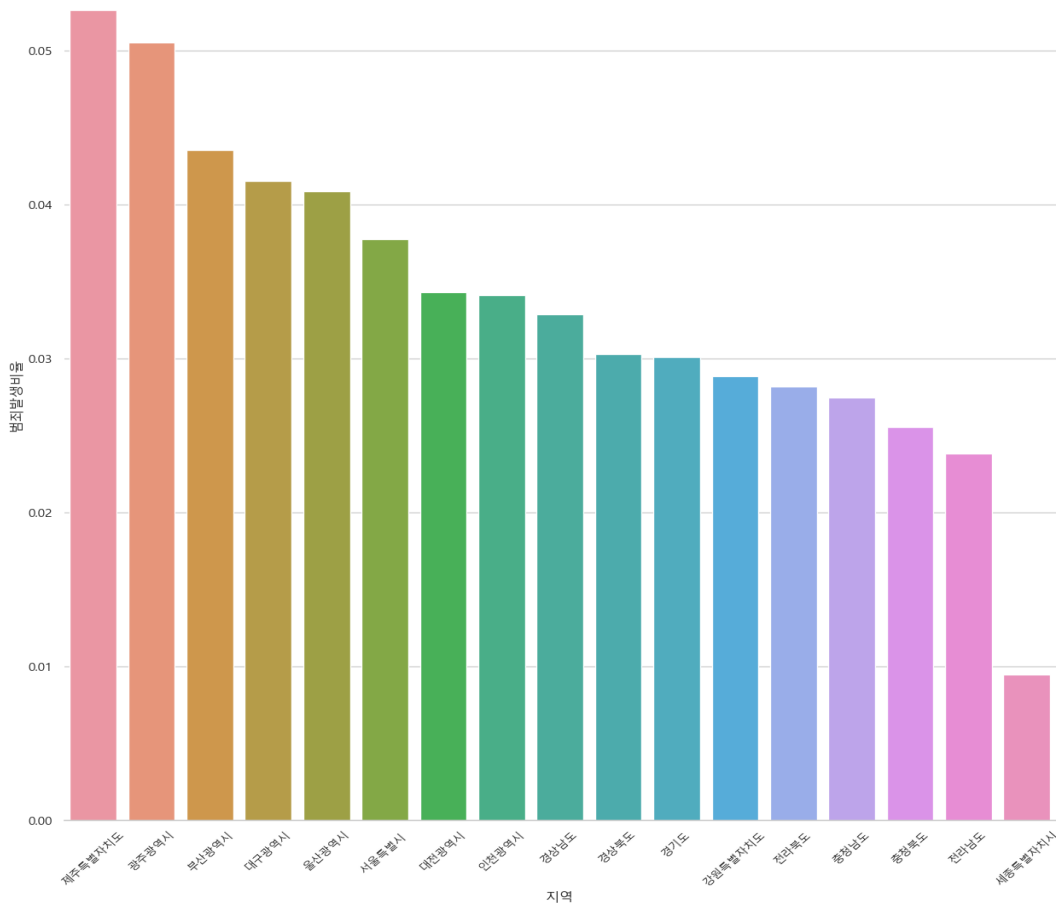
8. 지역별 인구수에 따른 범죄 발생율

	지역	범죄발생비율
17	제주특별자치도	0.052647
5	광주광역시	0.050540
2	부산광역시	0.043538
3	대구광역시	0.041533
7	울산광역시	0.040882
1	서울특별시	0.037772
6	대전광역시	0.034319
4	인천광역시	0.034114
16	경상남도	0.032860
15	경상북도	0.030317
9	경기도	0.030087
10	강원특별자치도	0.028861
13	전라북도	0.028179
12	충청남도	0.027456
11	충청북도	0.025547
14	전라남도	0.023814
8	세종특별자치시	0.009484

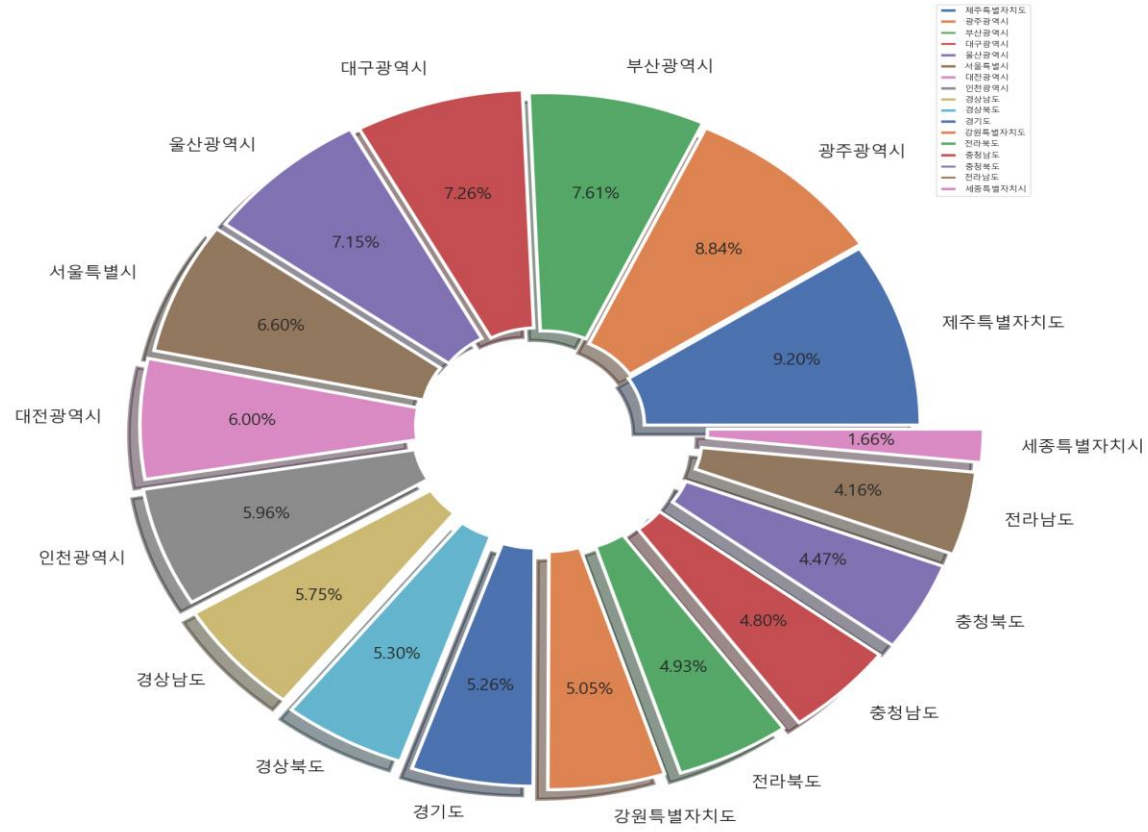
- ▶ 분석목표: 지역별 인구수에 따른 범죄 발생 율 에 대해 알아보기
- ▶ 활용한 데이터
 - 202309_202309_주민등록인구및세대현황_월간.csv
 - 정제된 데이터프레임(지역별 범죄 총 합계)
- ▶ 표 생성 과정
 - ▶ - 데이터프레임끼리 연결시킬 기준이 되는 컬럼을 생성시킨 후 두 데이터프레임을 합쳐서 비율계산식을 적용시킨 후 비율이 높은 순으로 적용 후 출력 시킨다.
- ▶ 표 분석 결과
 - ▶ - 전국 범죄 발생과 검거현황에 1순위: 범죄 발생
 - ▶ - 2순위는 검거, 3순위는 검거인원(남)이 제일 많다.

8. 지역별 인구수에 따른 범죄 발생율

지역별 인구수에 따른 범죄 발생율



지역별 인구수에 따른 범죄 발생율



그래프 분석 결과: 지역별 범죄가 자주 발생하는 곳 1순위는 제주이며 그 다음은 광주, 부산, 대구, 울산, 서울 순으로 서울이 6순위 입니다. 범죄가 많이 일어나는 서울보다 제주가 범죄 발생률이 높게 나왔습니다.

9. 최종 결과

- 자주 일어나는 범죄는 교통범죄
- 지역은 서울과 경기
- 장소는 노상(길거리)
- 시간대는 저녁 9시부터 12까지
- 요일은 금요일
- 범죄 발생시 인지할 수 있는 기간은 1개월 이내
- 범죄 검거 현황은 주로 남성비율이 높다.
- 범죄 발생율이 높은 곳은 서울이 아닌 제주가 높다.

<결론>

- 금요일 저녁 9시부터는 경찰관들이 순찰을 자주 다녀야 합니다. 특히 교통범죄가 많이 일어나기 때문에 교통순찰의 강화와 노상(길거리)위주로 순찰을 다녀야 합니다. 그리고 제주가 범죄 발생율이 높기 때문에 제주도 순찰을 강화해야 합니다.