

# 2023년 서울특별시 음주운전 사고의 데이터 분석

Data Visualization Report

Data Science



분석카세 1호

32214712 최현석(조장), 32212293 신명섭, 32214548 최석민





# Contents



1. 문제 상황 및 주제 소개

2. 가설 설정

3. 데이터 수집 및 전처리

4. 데이터 시각화 및 분석

5. 분석 결과 및 해결방안



## 문제 상황 및 주제 소개

# 문제 상황 및 주제 소개

## 2023년 서울특별시 음주운전 사고의 데이터분석

음주운전 사고는 주요 교통사고 원인중 하나로, 법적 처벌과 단속 강화에도 근절되지 않는 반복적 사회 문제이다. 사고 시간대, 장소, 연령대 등 다양한 요인을 구체적으로 분석하지 못하면 효과적인 대책 마련이 어렵다.

그러므로 음주운전 적발 및 사고 데이터에서의 단속, 시간대, 성별, 위치, 연령 등 다양한 변수가 포함된 데이터로 데이터 분석과 시각화를 하면 숨겨진 패턴을 발견하고 사고를 예방하는 데 도움을 줄 수 있다.





# 가설설정

# 가설 설정



1. 평일보다 주말에 음주운전 단속 적발 건수가 많을 것이다.



2. 젊은 남성이 음주운전 단속에서 가장 많이 적발될 것이다.



3. 겨울철(연말) 밤 시간대에 음주운전 단속 적발 건수가 가장 많을 것이다.



4. 음주운전 단속 적발 건수가 많을수록 사고 발생 건수와 사상자 수가 높아질 것이다.



5. 서울시 유흥가에서 음주운전 단속 적발과 사고 발생 건수가 가장 많을 것이다.





## 데이터 수집

# 데이터 수집 과정

2023년 서울시 음주운전 단속 적발은

공공 데이터 포털(<https://www.data.go.kr>)에서 제공하는 데이터를 활용.

수집기관: 대한민국 경찰청

해당 데이터에는 음주운전 적발의 운전자의 정보(성별, 나이), 적발횟수, 알코올농도, 측정거부, 측정일시, 관할경찰서의 정보가 들어있음.

데이터 1

DATA GO.KR 데이터포털 국가데이터랩 데이터요청 데이터활용 정보공유 위원회 이용안내

로그인 회원가입 사이트맵 ENGLISH

데이터 상세

경찰청\_음주운전

음주운전 적발기록을 통한 음주대행 자료 활용  
개공 자료 : 성별, 적발횟수, 나이, 알코올농도, 측정일시, 관할경찰서

파일데이터 | 오픈API | 추천데이터

공공데이터활용지침에서는 공공데이터포털에 개방되는 3단계 이상의 오픈 포맷 파일데이터를 오픈 API(OpenAPI) 기반의 JSON/XML로 자동변환하여 제공합니다.  
오픈 API를 활용하기 위해서는 공공데이터포털 회원 가입 및 활용신청이 필요하며, 활용 관련 문의는 공공데이터활용지침센터로 연락주시기 바랍니다.  
데이터 자체에 대한 문의는 아래 제공기관의 관리부서 전화번호로 연락주시기 바랍니다.  
파일데이터는 로그인 없이 다운로드를 통해 이용하실 수 있습니다.

CSV | 경찰청\_음주운전

경찰청\_음주운전\_2023.csv [C:]

다운로드

Q9

	A	B	C	D	E	F	G	H	I	J
1	연번	성별	적발횟수	나이	알코올농도	측정거부	측정일시	관할경찰서		
2	2	1 남자	1	31	0.136 N		#####	수원남부경찰서		
3	3	2 남자	3	56	0.065 N		#####	일산동부경찰서		
4	4	3 남자	1	34	0.094 N		#####	수원서부경찰서		
5	5	4 남자	1	35	0.191 N		#####	분당경찰서		
6	6	5 남자	1	28	0.043 N		#####	천안서북경찰서		
7	7	6 남자	2	47	0 Y		#####	구미경찰서		
8	8	7 남자	4	60	0.067 N		#####	고양경찰서		
9	9	8 남자	2	52	0.179 N		#####	평택경찰서		
10	10	9 남자	2	47	0.178 N		#####	아산경찰서		
11	11	10 남자	3	54	0.132 N		#####	영월경찰서		
12	12	11 남자	1	46	0.139 N		#####	청주흥덕경찰서		
13	13	12 남자	1	27	0.039 N		#####	부산남부경찰서		
14	14	13 남자	1	24	0.086 N		#####	의정부경찰서		
15	15	14 여자	1	46	0.127 N		#####	부산해운대경찰서		
16	16	15 여자	2	59	0.059 N		#####	구미경찰서		
17	17	16 남자	5	50	0.075 N		#####	서울성북경찰서		
18	18	17 남자	1	29	0.111 N		#####	부천원미경찰서		
19	19	18 남자	1	46	0.072 N		#####	광주광산경찰서		
20	20	19 남자	1	40	0.101 N		#####	이천경찰서		
21	21	20 남자	1	24	0.066 N		#####	대구중부경찰서		
22	22	21 남자	1	46	0.097 N		#####	부안경찰서		
23	23	22 남자	5	41	0.094 N		#####	광주북부경찰서		
24	24	23 남자	2	25	0.093 N		#####	군산경찰서		
25	25	24 여자	1	34	0.179 N		#####	담양경찰서		
26	26	25 남자	1	33	0.113 N		#####	속초경찰서		
27	27	26 남자	2	36	0.156 N		#####	시흥경찰서		

경찰청\_음주운전\_2023.csv




# 데이터 수집 과정

2023년 서울시 음주운전 교통사고 현황은  
서울 열린 데이터 광장(<https://data.seoul.go.kr>) 에서 제공  
하는 데이터를 활용.

수집기관: 한국도로교통공단

해당 데이터에는 서울시 자치구별 발생건수(건)와 사망자수  
(명), 부상자수(명)의 정보가 들어있음.



The screenshot shows the '서울시 음주운전사고 현황 통계' (Seoul Drunk Driving Accident Status Statistics) page. It includes a table with metadata such as '공개일자' (2014.10.21), '최신수정일자' (2022.07.01), and '데이터명' (음주운전사고 현황). A download button is visible on the right.

Below the screenshot is an Excel spreadsheet titled '음주운전\_2023\_교통사고\_현황.xlsx'. The spreadsheet contains the following data:

자치구별(1)	자치구별(2)	2023 발생건수 (건)	사망자수 (명)	부상자수 (명)
서울시	소계	1,938	8	3,113
	종로구	52	-	73
	중구	37	-	55
	용산구	64	2	114
	성동구	50	-	102
	광진구	57	-	82
	동대문구	56	1	74
	중랑구	57	-	98
	성북구	37	-	50
	강북구	36	-	61
	도봉구	44	-	72
	노원구	79	1	133
	은평구	56	1	80
	서대문구	34	-	41
	마포구	104	-	168
	양천구	65	-	103
	강서구	89	-	158

음주운전\_2023\_교통사고\_현황.xlsx

데이터 2

# 데이터 수집 과정

2019~2023년 음주운전 사고 다발지정보는  
교통사고정보 개방시스템 Open API

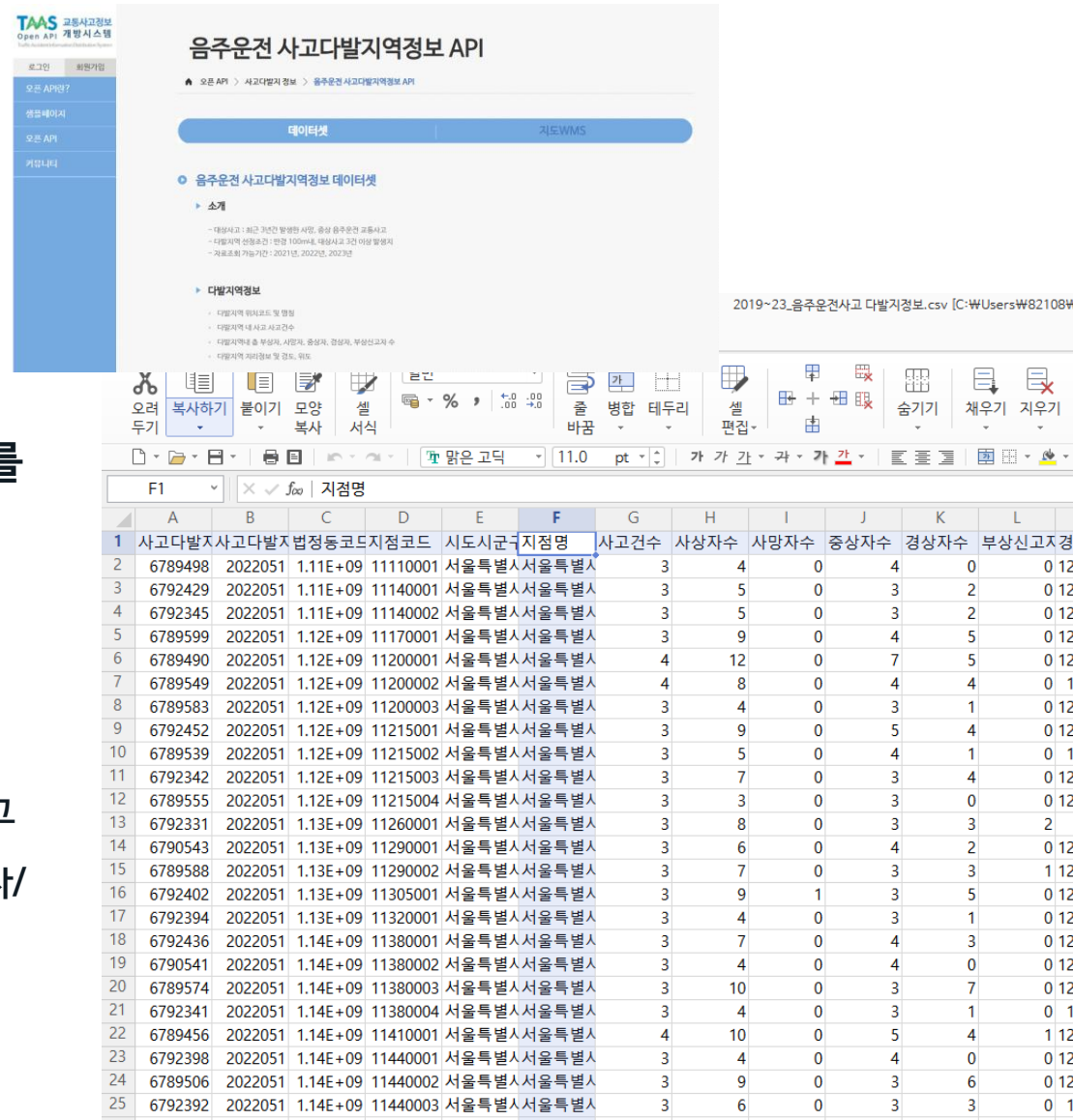
(<https://opendata.koroad.or.kr>) 에서 제공하는 데이터를  
활용.

수집 기관: **한국도로교통공단**

해당 데이터는 사고 다발지의 식별 번호, 법정동 코드, 지점 코드 및 사고  
발생 지점의 시군구명, 지점명, 사고 건수, 사상자/사망자/중상자/경상자/  
부상신고자의 수, 위도 및 경도, 다발지역폴리곤의 정보가 들어있음.

데이터 3

다발지역 선정조건: 반경 100m내, 사고 3건 이상 발생지



	A	B	C	D	E	F	G	H	I	J	K	L
1	사고다발지	사고다발지	법정동코드	지점코드	시도시군구	지점명	사고건수	사상자수	사망자수	중상자수	경상자수	부상신고자
2	6789498	2022051	1.11E+09	11110001	서울특별시	서울특별시	3	4	0	4	0	0 12
3	6792429	2022051	1.11E+09	11140001	서울특별시	서울특별시	3	5	0	3	2	0 12
4	6792345	2022051	1.11E+09	11140002	서울특별시	서울특별시	3	5	0	3	2	0 12
5	6789599	2022051	1.12E+09	11170001	서울특별시	서울특별시	3	9	0	4	5	0 12
6	6789490	2022051	1.12E+09	11200001	서울특별시	서울특별시	4	12	0	7	5	0 12
7	6789549	2022051	1.12E+09	11200002	서울특별시	서울특별시	4	8	0	4	4	0 1
8	6789583	2022051	1.12E+09	11200003	서울특별시	서울특별시	3	4	0	3	1	0 12
9	6792452	2022051	1.12E+09	11215001	서울특별시	서울특별시	3	9	0	5	4	0 12
10	6789539	2022051	1.12E+09	11215002	서울특별시	서울특별시	3	5	0	4	1	0 1
11	6792342	2022051	1.12E+09	11215003	서울특별시	서울특별시	3	7	0	3	4	0 12
12	6789555	2022051	1.12E+09	11215004	서울특별시	서울특별시	3	3	0	3	0	0 12
13	6792331	2022051	1.13E+09	11260001	서울특별시	서울특별시	3	8	0	3	3	2
14	6790543	2022051	1.13E+09	11290001	서울특별시	서울특별시	3	6	0	4	2	0 12
15	6789588	2022051	1.13E+09	11290002	서울특별시	서울특별시	3	7	0	3	3	1 12
16	6792402	2022051	1.13E+09	11305001	서울특별시	서울특별시	3	9	1	3	5	0 12
17	6792394	2022051	1.13E+09	11320001	서울특별시	서울특별시	3	4	0	3	1	0 12
18	6792436	2022051	1.14E+09	11380001	서울특별시	서울특별시	3	7	0	4	3	0 12
19	6790541	2022051	1.14E+09	11380002	서울특별시	서울특별시	3	4	0	4	0	0 12
20	6789574	2022051	1.14E+09	11380003	서울특별시	서울특별시	3	10	0	3	7	0 12
21	6792341	2022051	1.14E+09	11380004	서울특별시	서울특별시	3	4	0	3	1	0 1
22	6789456	2022051	1.14E+09	11410001	서울특별시	서울특별시	4	10	0	5	4	1 12
23	6792398	2022051	1.14E+09	11440001	서울특별시	서울특별시	3	4	0	4	0	0 12
24	6789506	2022051	1.14E+09	11440002	서울특별시	서울특별시	3	9	0	3	6	0 12
25	6792392	2022051	1.14E+09	11440003	서울특별시	서울특별시	3	6	0	3	3	0 1

2019~23\_음주운전사고 다발지정보.csv

# 데이터 수집 과정

관할 경찰서의 위치와 경도는 오픈스트리트맵  
(<https://www.openstreetmap.org>)에서 서울시에 있는 경찰서의 위치를 검색.

위도 및 경도 좌표 추출: 각 경찰서의 위치 정보를 지도에서 확인한 후, 제공되는 위도(latitude)와 경도(longitude)를 수집.

딕셔너리 작성: 각 자치구별 경찰서 이름과 해당 위도 및 경도를 딕셔너리 형태로 정리.

데이터 4

Nominatim
Search
Reverse
Search By ID
About & Help

W602150678
Show

OSM type+id (N123, n123, W123, w123, R123, r123), Place id (1234) or URL (<https://openstreetmap.org/way/123>)

Data from API request (debug output)
Data last updated: 1 minute ago (Details)

## 서울강남경찰서 [link to this page](#)

Name	서울강남경찰서 (name) Seoul Gangnam Police Station (name:en) ソウル江南警察署 (name:ja) 서울강남경찰서 (name:ko)
Type	amenity:police
Last Updated	2024-11-09T02:58:38.557252+00:00
Search Rank	30
Address Rank	30 (house / building)
Importance	0.3772346674060423
Coverage	Polygon
Centre Point (lat,lon)	37.5094404,127.06707157813565
OSM	<a href="#">way 602150678</a>
Place Id	205272409 (on this server)
Wikipedia Calculated	<a href="#">ja:ソウル江南警察署</a>
Computed Postcode	06175 (how?)



# 데이터 전처리

# 데이터 전처리 과정

## 데이터 1

### 1. 데이터 로드 및 구조와 정보 확인

CSV 파일(경찰청\_음주운전\_2023.csv)을 pandas로 불러온 뒤, 데이터 구조 및 정보 확인.

### 2. 날짜 및 시간 처리 후 열 생성

[측정일시]를 datetime 형식으로 변환하여 월, 일, 시, 분, 요일명을 추출하여 새로운 열로 저장.

### 3. 나이/성별의 결측치 처리

[나이]과 [성별]의 '불명' 값을 결측치로 변환 후, 행 삭제. [나이]를 정수형으로 변환.

### 4. 연령대/count 열 생성

[연령대] 열 생성: 나이를 기준으로 10대, 20대 등 연령대 범주를 구분하는 열을 추가.

[count] 열 생성: 모든 행에 대해 단속 건수를 나타내는 [count] 열을 추가하여 값을 1로 설정.

### 5. 서울시 데이터 필터링

[관할경찰서] 열에서 '서울'이 포함된 행만 필터링하여 서울시 데이터만 추출 후 인덱스 정리

## 최종 데이터 – 2023년 서울시 음주운전 단속 적발 데이터 완성.

#### 1.1 데이터1: 2023년 전국 경찰청별 음주운전 적발 데이터

- 전국에서 서울특별시만 추출

```
## 1. 데이터 로드 및 구조와 정보 확인
# 2023년 전국 경찰청별 음주운전 적발 데이터
dd_caught = pd.read_csv('./DATA/경찰청_음주운전_2023.csv', encoding='euc-kr', index_col='연번')
dd_caught.head()
```

```
dd_caught.info() # 데이터정보
```

```
dd_caught.shape # 전국 125231개 행, 7개 열
```

```
## 2. 날짜 및 시간 처리 후 열 생성성
# 월, 일, 시간, 분 추출하여 열 추가
dd_caught['측정일시'] = pd.to_datetime(dd_caught['측정일시'])
dd_caught['월'] = dd_caught['측정일시'].dt.month
dd_caught['일'] = dd_caught['측정일시'].dt.day
dd_caught['시간'] = dd_caught['측정일시'].dt.hour
dd_caught['분'] = dd_caught['측정일시'].dt.minute
dd_caught.head()
```

```
# 요일을 추출하여 새로운 열 추가하고 요일명으로 값 변경
dd_caught['요일'] = dd_caught['측정일시'].dt.dayofweek
day_names = {0: '월', 1: '화', 2: '수', 3: '목', 4: '금', 5: '토', 6: '일'}
dd_caught.replace({'요일': day_names}, inplace=True)
dd_caught.head()
```

```
## 3. 나이/성별의 결측치 처리
# 나이가 '불명'인 행 삭제, object형에서 int형 변환
dd_caught['나이'].replace('불명', np.nan, inplace=True)
dd_caught.dropna(subset=['나이'], axis=0, inplace=True)
dd_caught['나이'] = dd_caught['나이'].astype('int')
dd_caught['나이'].dtypes
```

```
## 4. 연령대/count 열 생성성
# 연령대 열 생성
bins = [0, 19, 29, 39, 49, 59, 69, 100] # 나이 범위
labels = ['10대', '20대', '30대', '40대', '50대', '60대', '70대 이상'] # 범주 레이블
dd_caught['연령대'] = pd.cut(dd_caught['나이'], bins=bins, labels=labels, include_lowest=True)
dd_caught.head()
```

[+ 코드](#) [+ Markdown](#)

```
# 성별이 '불명'인 행 삭제
dd_caught['성별'].replace('불명', np.nan, inplace=True)
dd_caught_seoul = dd_caught.dropna(subset=['성별'], axis=0)
```

```
dd_caught['count'] = 1 # 카운트 열 추가
dd_caught.head()
```

```
## 5. 서울시 데이터 필터링
# 서울시 데이터 추출
dd_caught_seoul = dd_caught[dd_caught['관할경찰서'].str.contains('서울')]
dd_caught_seoul.reset_index(drop=True, inplace=True)
dd_caught_seoul
```

```
## 서울시 적발횟수, 나이, 알콜농도 기준으로 정보 확인
dd_caught_seoul.loc[:, dd_caught_seoul.columns.isin(['적발횟수', '나이', '알콜농도'])].describe().round(2)
```

# 데이터 전처리 과정

## 데이터 2

### 1. 데이터 로드 및 구조와 정보 확인

엑셀 파일(음주운전\_2023\_교통사고\_현황.xlsx)을 pandas로 불러온 뒤, 데이터 구조 및 정보 확인.

### 2. 불필요한 행/열 제거

모든 값이 서울시로 동일한 [자치구별(1)] 열 삭제, 전체 합계 데이터인 [소계] 행 삭제

### 3. 행/열 인덱스명 변경

[자치구별(2)] 열을 '서울자치구'로 이름 변경. [서울자치구] 열을 행 인덱스 변경

열 이름을 간소화하기 위해 '발생건수', '사망자', '부상자'로 변경.

### 4. 사망자의 결측치 및 데이터 형식 처리

[사망자] 열에서 값이 '-'로 표기된 항목을 결측치로 변환 후, 0으로 채움. 데이터 형식을 정수형으로 변환.

### 5. 사상자 열 추가

[사상자] 열 추가: [사망자]열과 [부상자]열을 더하여 총 사상자 수 계산.

#### 1.2 데이터2: 2023년 서울시 음주운전 교통사고 데이터

```
## 1. 데이터 로드 및 구조와 정보 확인
# 2023년 서울시 음주운전 교통사고 데이터
dd_accidents = pd.read_excel('./DATA/음주운전_2023_교통사고_현황.xlsx', header=1)
dd_accidents.head()
```

```
dd_accidents.info() # 데이터 기본 정보
```

```
dd_accidents.shape # 데이터 26개 행, 5개 열
```

```
## 2. 불필요한 행/열 삭제
# 자치구별(1)의 값이 모두 동일하므로 삭제
dd_accidents.drop('자치구별(1)', axis=1, inplace=True)
dd_accidents.head()
```

```
# 소계 - 0행 삭제
dd_accidents.drop(0, axis=0, inplace=True)
dd_accidents.head()
```

```
## 3. 행/열 인덱스명 변경
# 자치구별(2) 열 이름 변경
dd_accidents.rename({'자치구별(2)': '서울자치구'}, axis=1, inplace=True)
dd_accidents.head()
```

```
# 행 인덱스 변경
dd_accidents.set_index('서울자치구', inplace=True)
dd_accidents.head()
```

+ 코드

```
# 열 인덱스 이름 변경 (변수 이름 쉽게 코딩 하기위해)
dd_accidents.columns = ['발생건수', '사망자', '부상자']
dd_accidents.head()
```

```
## 4. 사망자의 결측치 및 데이터 형식 처리
# 사망자 열의 '-'를 0으로 변경, object형을 int형으로 변환
dd_accidents['사망자'].replace('-', np.nan, inplace=True)
dd_accidents.fillna(0, inplace=True)
dd_accidents['사망자'] = dd_accidents['사망자'].astype('int')
dd_accidents.head()
```

```
## 5. 사상자 열 추가
dd_accidents['사상자'] = dd_accidents['사망자'] + dd_accidents['부상자']
dd_accidents
```



# 데이터 전처리 과정

데이터 2.5

< 데이터 2 에 딕셔너리와 데이터1을 이용하여 '자치구별 적발 수' 열 추가 >

## 1. 경찰서별 자치구 딕셔너리

경찰서별 자치구를 구하여 경찰서명과 자치구를 딕셔너리 형태로 저장

## 2. 데이터프레임 생성

정리한 딕셔너리를 pandas 데이터프레임으로 변환 후 전치 후 정리

## 3. [데이터1]에 관할경찰서의 자치구 열 추가

관할 경찰서를 기준으로 [데이터1]에 left 병합

## 4. [데이터2]에 [데이터1]의 자치구별 적발 수 열 추가

음주운전 사고 현황 데이터[데이터2]에 [데이터1]의 자치구별 적발 수 열 추가

최종 데이터 – 2023년 서울시 자치구별 교통사고 현황 데이터 완성.

### 1.2.5 데이터1과 데이터2의 병합

- 2023년 서울시 음주운전 교통사고 데이터 + 2023년 전국 경찰청별 음주운전 적발 데이터

```
# 경찰서별 자치구 딕셔너리
police_stations = {
    '서울강남경찰서': '강남구',
    '서울강동경찰서': '강동구',
    '서울강북경찰서': '강북구',
    '서울강서경찰서': '강서구',
    '서울관악경찰서': '관악구',
    '서울광진경찰서': '광진구',
    '서울구로경찰서': '구로구',
    '서울금천경찰서': '금천구',
    '서울남대문경찰서': '종로',
    '서울노원경찰서': '노원구',
    '서울도봉경찰서': '도봉구',
    '서울동대문경찰서': '동대문구',
    '서울동작경찰서': '동작구',
    '서울마포경찰서': '마포구',
    '서울방배경찰서': '서초구',
    '서울서대문경찰서': '서대문구',
    '서울서부경찰서': '은평구',
    '서울서초경찰서': '서초구',
    '서울성동경찰서': '성동구',
    '서울성북경찰서': '성북구',
    '서울송파경찰서': '송파구',
    '서울양천경찰서': '양천구',
    '서울영등포경찰서': '영등포구',
    '서울용산경찰서': '용산구',
    '서울은평경찰서': '은평구',
    '서울종로경찰서': '종로',
    '서울중랑경찰서': '성북구',
    '서울중랑경찰서': '중랑구',
    '서울중부경찰서': '종로',
    '서울혜화경찰서': '종로'
}

# police 데이터프레임 생성
police = pd.DataFrame([police_stations]).transpose() # 전치
police = police.reset_index() # 인덱스 재설정
police.rename({'index': '관할경찰서', 0: '자치구'}, axis=1, inplace=True) # 열 이름 변경
police.head()

# 관할경찰서를 기준으로 left 병합 - 데이터1에 관할경찰서 자치구 열 추가
dd_caught_seoul = pd.merge(dd_caught_seoul, police, on='관할경찰서', how='left')
dd_caught_seoul.head()

# 교통사고 현황 데이터[데이터2]에 자치구별 적발 수 열[데이터1의 열] 추가
dd_accidents['적발수'] = dd_caught_seoul['자치구'].value_counts()
dd_accidents.head()

## 서울시 음주운전 사고 자치구별 발생건수, 사망자, 부상자, 사상자, 적발수 기초통계정보 확인
dd_accidents.describe().round(2)
```

# 데이터 전처리 과정

## 데이터 3

### 1. 데이터 로드 및 구조와 정보 확인

CSV 파일(2019~23\_음주운전사고 다발지정보.csv)을 pandas로 불러옴.  
데이터 구조 및 정보 확인.

### 2. 2023년 서울시 데이터 필터링

' 사고다발지id ' 값이 ' 2024 ', ' 지점명 ' 이 ' 서울 ' 로 시작하는 데이터를 필터링 후,  
인덱스 정리

### 3. 자치구 열 추가

[시도시군구명] 열에서 정규식을 사용해 자치구 이름만을 추출하여 새로운 열로 추가.

### 4. 필요한 컬럼만 선택

분석에 필요한 열([자치구],[경도],[위도],[지점명])만 선택하여 데이터 정리.

#### 1.3 데이터3: 2019~2023년 음주운전사고 다발지 정보 데이터

- 2023년 서울시 데이터만 추출

```
## 1. 데이터 로드 및 구조와 정보 확인
# 2019~23년 음주운전사고 다발지 정보 데이터
dd_zone = pd.read_csv('./DATA/2019~23_음주운전사고 다발지정보.csv', encoding='euc-kr')
dd_zone.head(3)
```

```
dd_zone.info() # 데이터 정보
```

```
dd_zone.shape # 총 916행 15열
```

```
## 2. 2023년 서울시 데이터 추출
ddz_seoul23 = dd_zone[dd_zone['사고다발지id'].astype(str).str.startswith('2024') & dd_zone['지점명'].str.startswith('서울')]
ddz_seoul23.reset_index(drop=True, inplace=True) # 정수형 인덱스로 재설정
ddz_seoul23.head(2)
```

```
## 3. 시도시군구명에서 추출한 자치구 열 추가
import re
ddz_seoul23['자치구'] = ddz_seoul23['시도시군구명'].apply(lambda x: re.sub(r"^(서울|경기|강원|충청|전라|경남|제주)\d+", '', x).strip())
ddz_seoul23.head(3)
```

```
## 4. 필요한 컬럼 추출
ddz_s23 = ddz_seoul23.loc[:, ['자치구', '경도', '위도', '지점명']]
ddz_s23
```

**최종 데이터 - 2023년 서울시 음주운전 사고 다발지의 자치구별 경도 및 위도 데이터 완성.**

# 데이터 전처리 과정

## 데이터 4

### 1. 데이터 수집 후 딕셔너리로 저장

오픈스트리트맵(OpenStreetMap)을 통해 서울시 주요 경찰서의 위도(latitude)와 경도(longitude) 데이터를 수집하여 딕셔너리 형태로 각 경찰서의 이름과 좌표 정보를 저장.

### 2. 데이터프레임 생성

정리한 딕셔너리를 pandas 데이터프레임으로 변환 후 전치

### 3. 관할 경찰서별 음주운전 적발 건수 열 추가

기존의 서울시 음주운전 단속 데이터에서 관할 경찰서별 음주운전 단속 횟수를 새로운 열로 추가

### 최종 데이터프레임

- 위도: 각 경찰서의 위도 좌표.
- 경도: 각 경찰서의 경도 좌표.
- 적발수: 각 경찰서 관할 지역에서 적발된 음주운전 단속 건수.

#### 1.4 데이터4. 경찰서 위도와 경도 데이터프레임

- 오픈스트리트맵으로 경찰서를 검색하여 위도 및 경도를 찾아 딕셔너리로 작성

```
## 1. 데이터 수집 후 딕셔너리로 저장
# 경찰서 위도와 경도 데이터프레임
police_stations_coordinates = {
    '서울강남경찰서': ('위도': 37.5094404, '경도': 127.06707157813565),
    '서울강동경찰서': ('위도': 37.5298016, '경도': 127.1243425),
    '서울강북경찰서': ('위도': 37.5286553, '경도': 127.12703758845649),
    '서울관악경찰서': ('위도': 37.5623742, '경도': 126.8083173),
    '서울관악경찰서': ('위도': 37.4745006, '경도': 126.95122219979444),
    '서울관악경찰서': ('위도': 37.5428572, '경도': 127.0838484520238),
    '서울구로경찰서': ('위도': 37.4949102, '경도': 126.88661471653955),
    '서울금천경찰서': ('위도': 37.45714415, '경도': 126.89673476372995),
    '서울남대문경찰서': ('위도': 37.554711350000005, '경도': 126.9735285900517),
    '서울노원경찰서': ('위도': 37.642401199999995, '경도': 127.07154054235139),
    '서울도봉경찰서': ('위도': 37.653649200000004, '경도': 127.05234500098712),
    '서울동대문경찰서': ('위도': 37.58513455, '경도': 127.04577800967854),
    '서울동작경찰서': ('위도': 37.51311465, '경도': 126.94279715828962),
    '서울마포경찰서': ('위도': 37.5508192, '경도': 126.9537886026813),
    '서울방배경찰서': ('위도': 37.48137625, '경도': 126.98299880975085),
    '서울서대문경찰서': ('위도': 37.5648953, '경도': 126.96674193431708),
    '서울서부경찰서': ('위도': 37.6020381, '경도': 126.9212474),
    '서울성동경찰서': ('위도': 37.495436, '경도': 127.00523076699471),
    '서울성동경찰서': ('위도': 37.561716, '경도': 127.036215),
    '서울성동경찰서': ('위도': 37.58971995, '경도': 127.01625459517814),
    '서울송파경찰서': ('위도': 37.50203315, '경도': 127.1270483060253),
    '서울수서경찰서': ('위도': 37.493566650000005, '경도': 127.07728521741271),
    '서울양천경찰서': ('위도': 37.51672765, '경도': 126.86589013441235),
    '서울영등포경찰서': ('위도': 37.52597925, '경도': 126.90110740245895),
    '서울용산경찰서': ('위도': 37.5411522, '경도': 126.9677416),
    '서울은평경찰서': ('위도': 37.628402750000001, '경도': 126.92871950943336),
    '서울종로경찰서': ('위도': 37.5755299, '경도': 126.98489591236623),
    '서울종로경찰서': ('위도': 37.6020575, '경도': 127.032176),
    '서울중랑경찰서': ('위도': 37.6185277, '경도': 127.10464524687671),
    '서울중랑경찰서': ('위도': 37.56358765, '경도': 126.98957762499465),
    '서울혜화경찰서': ('위도': 37.5728349, '경도': 126.9975849)
}

## 2. 경찰서 위도경도 데이터프레임 생성
police_coordinates = pd.DataFrame(police_stations_coordinates).transpose()

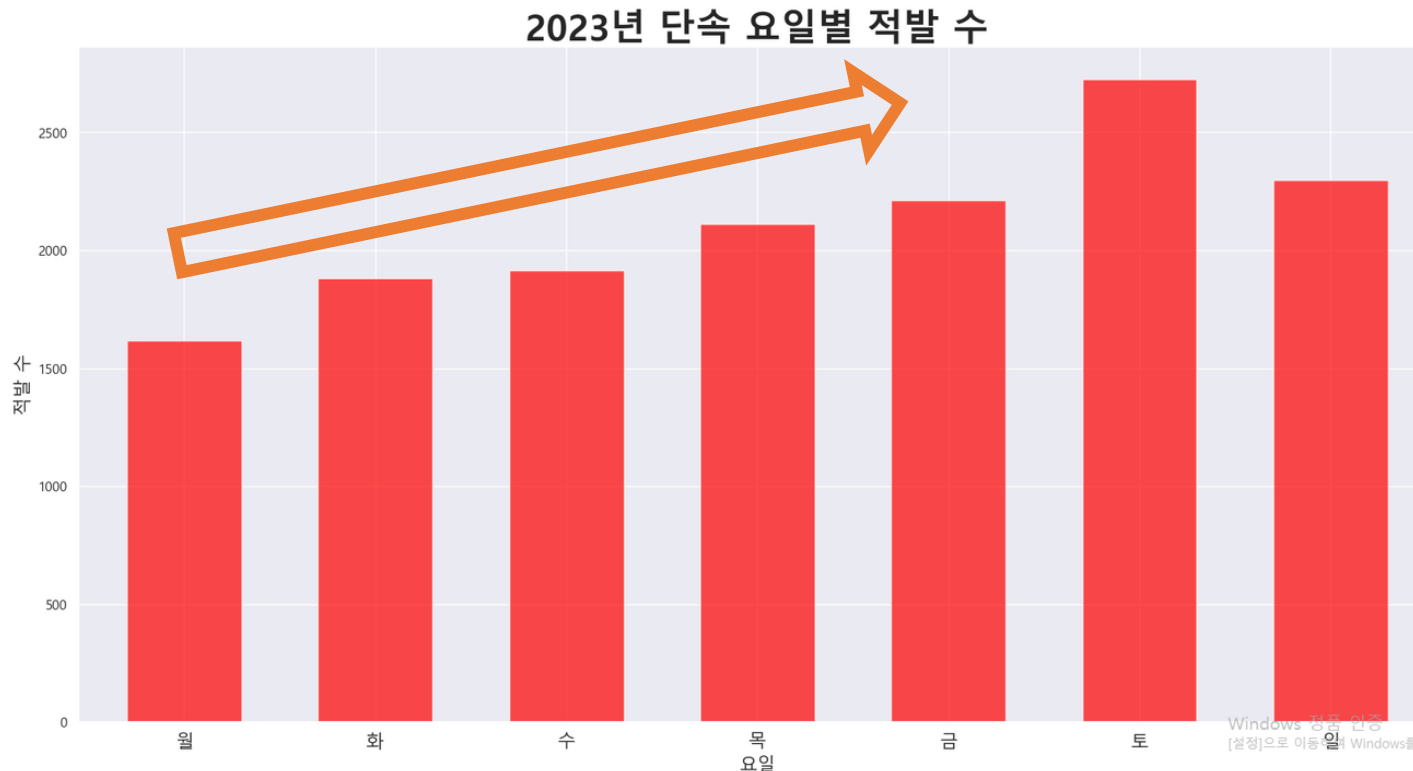
## 3. 경찰서 위도경도 데이터에 경찰서별 적발수 열 추가
police_coordinates['적발수'] = dd_caught_seoul['관할경찰서'].value_counts()
police_coordinates
```



# 데이터 시각화 및 분석

## 데이터 시각화

## (1) 음주운전 단속의 요일별 적발 - 막대그래프



금요일과 주말에  
단속 적발 수가 상대적으로  
높고, 토요일이 적발 건수가  
가장 많다.

## 2.1. 음주운전 단속의 요일별 막대그래프

- 가설1. 음주운전 단속은 평일보다 주말에 많이 적발될 것이다.
- 데이터1 사용

```
day_count = dd_caught_seoul['요일'].value_counts() # 요일별 카운트 수
day_count = day_count.reindex(['월', '화', '수', '목', '금', '토', '일']) # 월화수목금토일 순으로 재정렬
day_count
```

```
# matplotlib 한글 폰트 오류 문제 해결
from matplotlib import font_manager, rc
font_path = './read_write/maibun.ttf' # 폰트파일 위치
font_name = font_manager.FontProperties(fname=font_path).get_name()
rc('font', family=font_name)

# 요일別に 대한 막대그래프
day_count.plot(kind='bar', figsize=(20,10), width=0.6, color='red', alpha=0.7)
plt.title('2023년 단속 요일별 적발 수', size=30, weight='bold')
plt.xlabel('요일', size=15)
plt.ylabel('적발 수', size=15)
plt.xticks(rotation=0, fontsize=15)
```

## 데이터 시각화

## (2) 성별과 연령대별 단속 적발 - 막대그래프

전반적으로 남성의 적발 건수가 여성보다 높은 패턴을 보이며, 30대 남성이 음주운전 적발 건수가 가장 많다. 30대 이후의 적발 건수는 감소하는 경향이 있다.

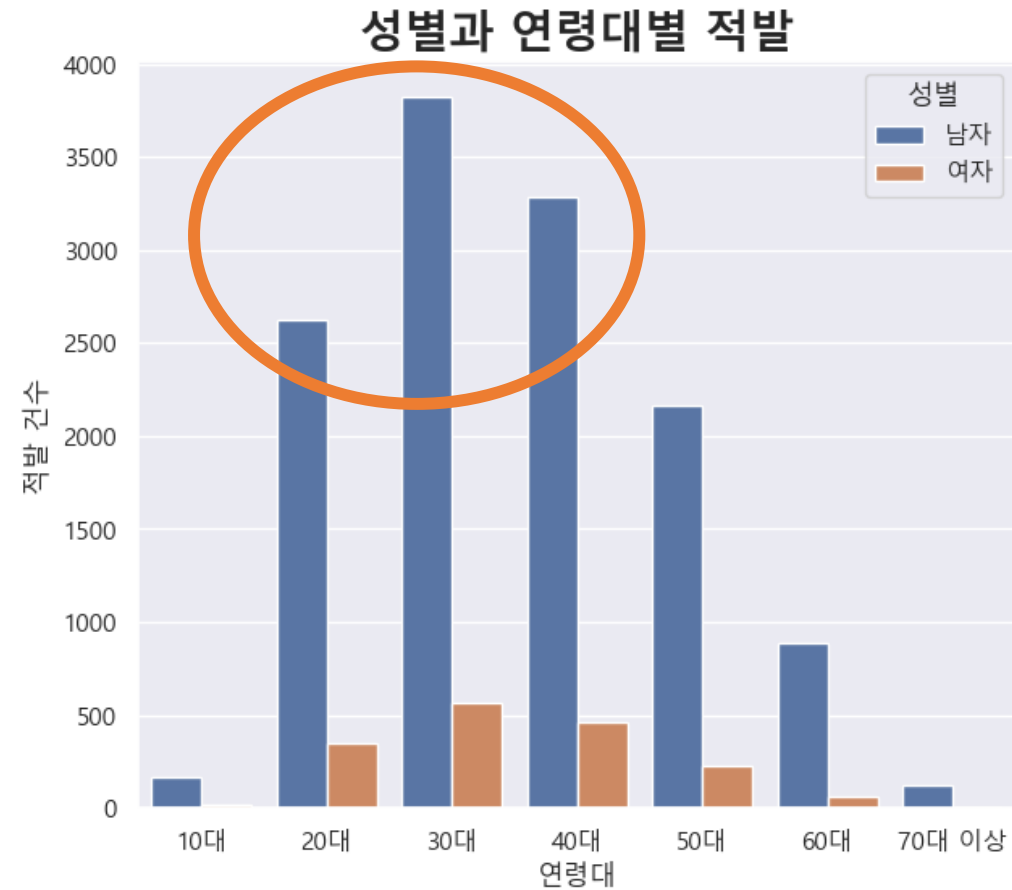
## 2.2 성별과 연령대별 단속 적발 막대그래프

- 가설2. 20-30대 남성이 단속이 많이 적발될 것이다.
- 데이터1 사용

```
sns.set_theme(rc={'figure.figsize':(7,6)})
plt.rcParams['font.family'] = 'Malgun Gothic' # 한글처리

sns.barplot(data=dd_caught_seoul, x='연령대', y='count', estimator='count', hue='성별', errorbar=None)
plt.title('성별과 연령대별 적발', size=20, weight='bold')
plt.ylabel('적발 건수')
```

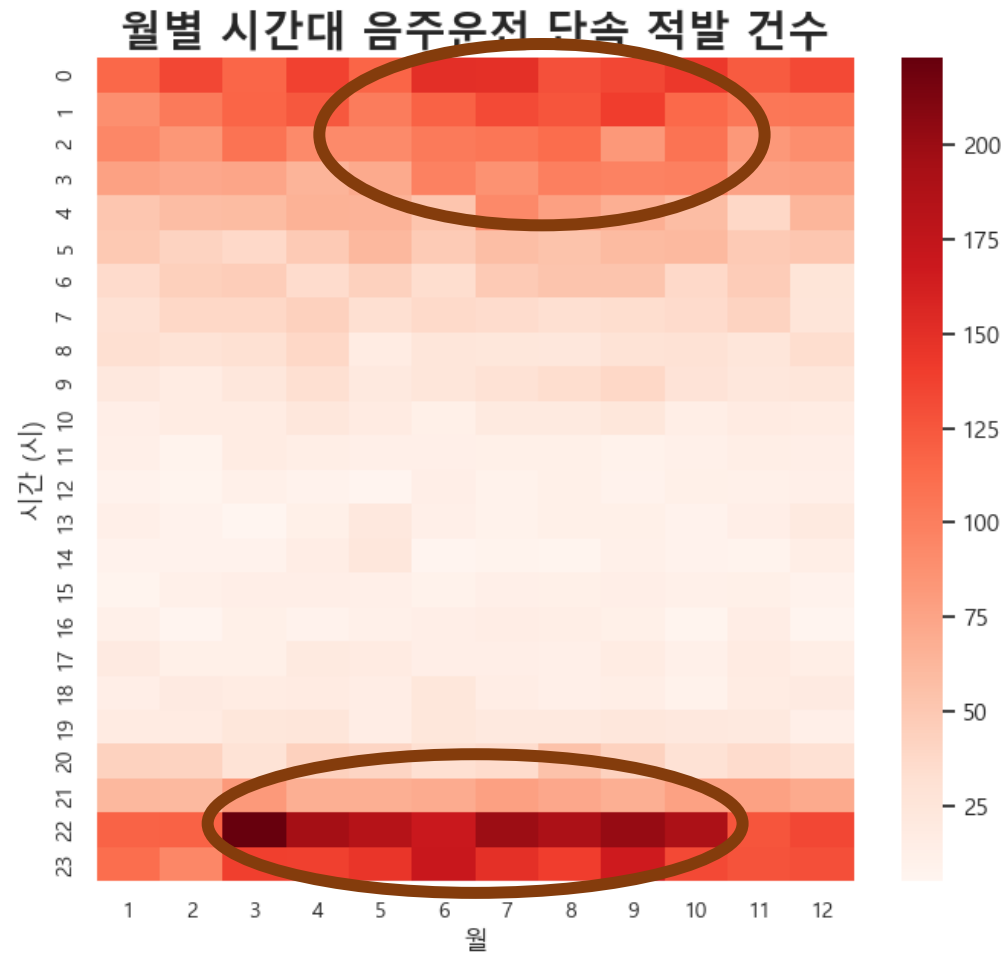
Text(0, 0.5, '적발 건수')





## 데이터 시각화

## (3) 월별 시간대 음주운전 단속 적발 - 히트맵



음주운전 단속이 주로  
여름철(6~9월)에 집중되며, 특  
히 밤 10시에서 새벽 2시 사  
이에 적발 건수가 가장 많은 경향  
을 나타낸다.

## 2.3 월별 시간대 음주운전 단속 적발 히트맵

- 가설3. 연말과 연초 밤에 단속 적발이 많을 것이다.
- 데이터1 사용

```
# 월별 시간대 피켓데이터를
dd_pt = pd.pivot_table(dd_caught_seoul, values='성별', index='시간', columns='월', aggfunc='count', fill_value=0)
dd_pt
```

```
# 월별 시간대 음주운전 단속 적발 건수 - 히트맵
sns.set_theme(rc={'figure.figsize':(8,7)})
plt.rcParams['font.family'] = 'Malgun Gothic' # 한글처리

sns.heatmap(dd_pt, cmap='Reds', cbar=True)

plt.title("월별 시간대 음주운전 단속 적발 건수", size=20, weight='bold')
plt.xlabel("월", size=12)
plt.ylabel("시간 (시)", size=12)
plt.xticks(fontsize=10)
plt.yticks(fontsize=10)
```

## 데이터 시각화

## (4) 자치구별 적발 수와 음주운전 발생, 사상자수의 관계 - 산점도

자치구별 적발 건수가 많을수록 음주운전 발생 건수와 사상자 수가 증가하는 경향이 있다.

## 2.4 자치구별 적발 수와 음주운전 발생, 사상자수의 산점도

- 가설4: 적발이 많을수록 음주운전 발생건수와 사상자수가 많을 것이다.
- 데이터2 사용

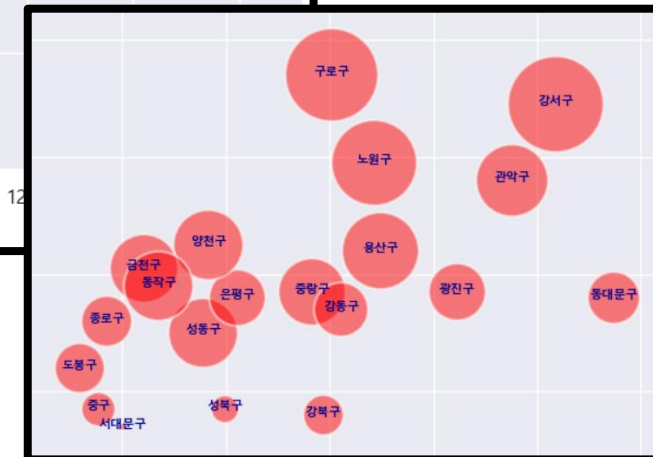
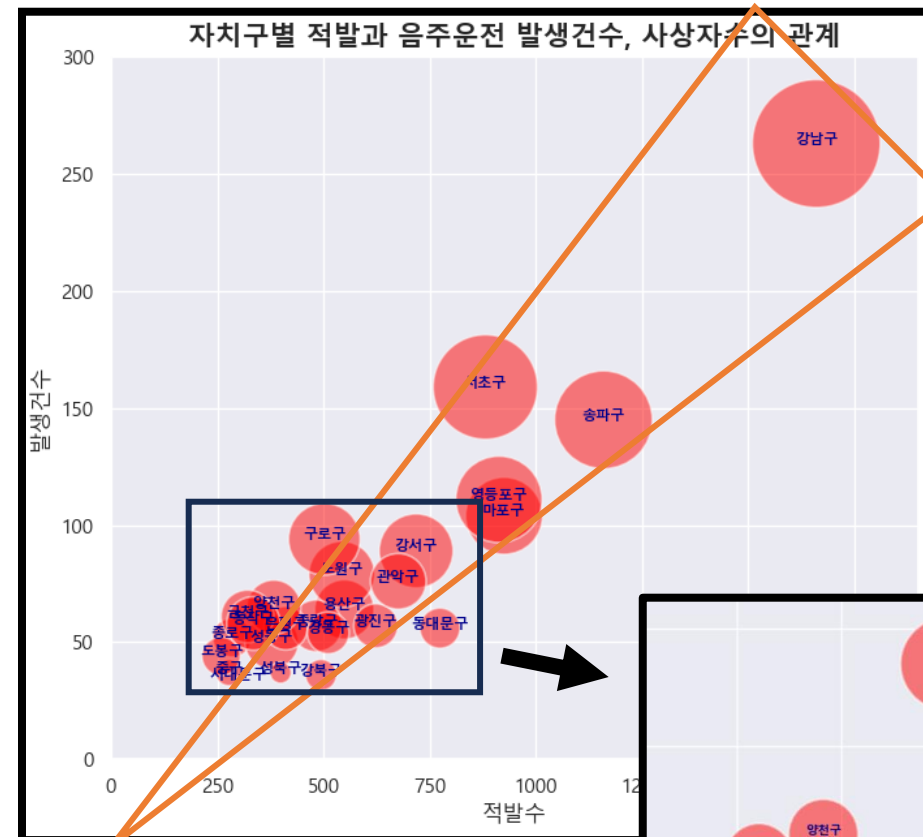
```
## 4.1 모든 자치구 시각화
sns.set_theme(rc={'figure.figsize':(8,7)})
plt.rcParams['font.family'] = 'Malgun Gothic' # 한글처리

sns.scatterplot(data=dd_accidents, x='적발수', y='발생건수', size='사상자', sizes=(10,5000), alpha=0.5, legend=False, color='red')

plt.title('자치구별 적발과 음주운전 발생건수, 사상자수의 관계', size=15, weight='bold')
plt.xlim(0,1000)
plt.ylim(0,300)

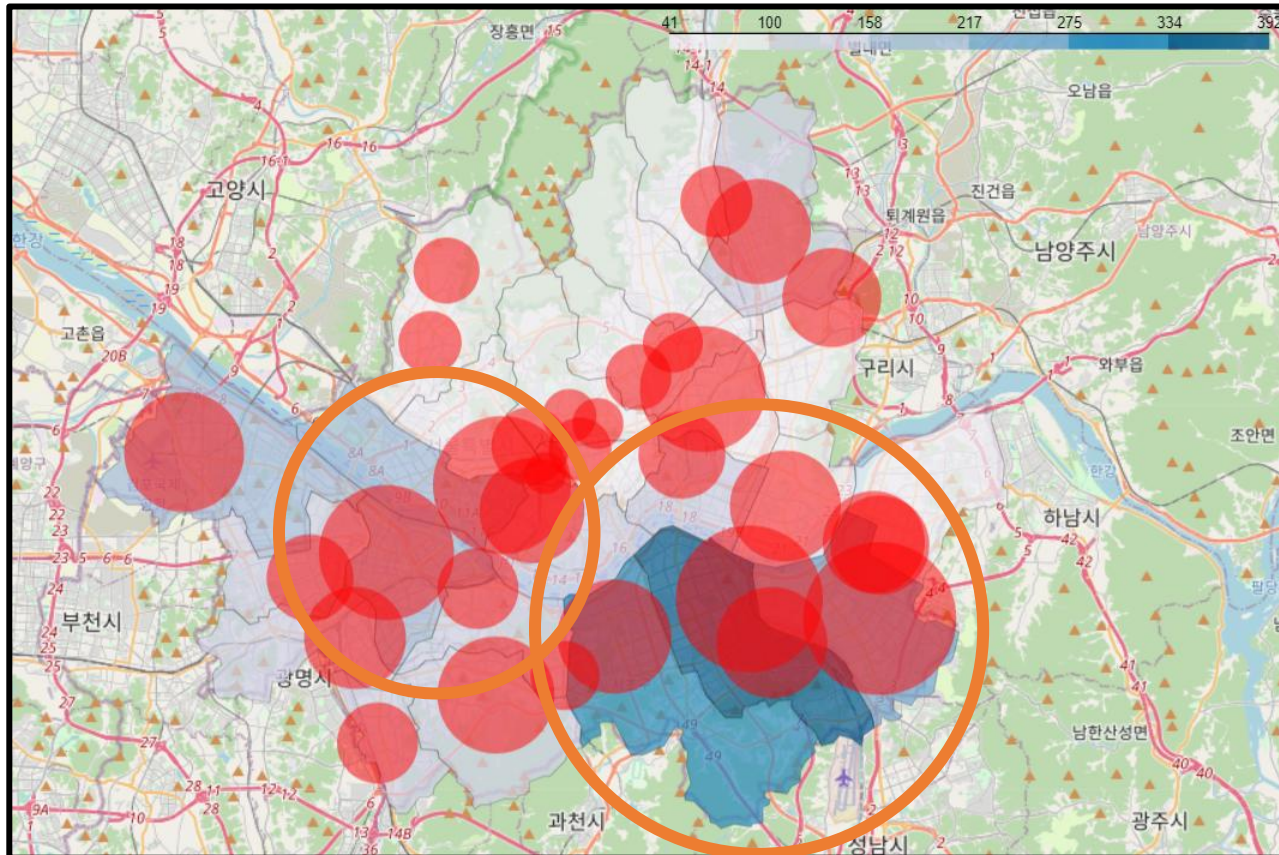
# 배열 위에 자치구명 표시
for i in range(0, dd_accidents.shape[0]):
    plt.text(x=dd_accidents.적발수[i], y=dd_accidents.발생건수[i], s=dd_accidents.index[i],
            horizontalalignment='center', size=8, color='darkblue', weight='bold')
```

- 사상자 수가 많을 수록 ● 크기가 커짐.



## (5) 음주운전 적발과 사고 지도화 데이터

## 1. 서울시 자치구별 음주운전 사상자와 적발수 지도시각화



대부분 강북보단 강남에 많이 분포되어 있다. 특히 강남구, 서초구, 송파구와 영등포구, 마포구 쪽이 적발수와 사상자가 많다.

- 가설5. 서울시 유흥가에 음수운전 적발과 사고가 많을 것이다.
- 데이터2, 3, 4 사용

```
import folium

# GeoJSON 파일을 URL로 불러오기
geo_json = 'https://raw.githubusercontent.com/southkorea/seoul-maps/master/kostat/2013/json/seoul_municipalities_geo_simple.json'
```

```
# 5.1 서울시 지역구별 음주운전 사상자와 적발수 지도 시각화
m1 = folium.Map(
    location = [37.566345, 126.977893], zoom_start = 11
)

folium.Choropleth(
    geo_data = geo_json,
    name = 'choropleth',
    data = dd_accidents,
    columns = [dd_accidents.index, '사상자'],
    key_on = 'feature.properties.name',
    fill_color = 'PuBu',
    fill_opacity=0.7,
    line_opacity = 0.2,
).add_to(m1)

for i in range(len(police_coordinates)):
    folium.CircleMarker(location=[police_coordinates.위도.iloc[i], police_coordinates.경도.iloc[i]],
        radius=(police_coordinates.적발수.iloc[i]**0.5)*1.5, # 원의 반지름
        color='red', # 원의 색
        stroke=False, # 윤곽선 없음
        fill=True, # 원의 내부 색
        fill_opacity='50%', # 원의 내부 색 투명도
    ).add_to(m1)

m1
```

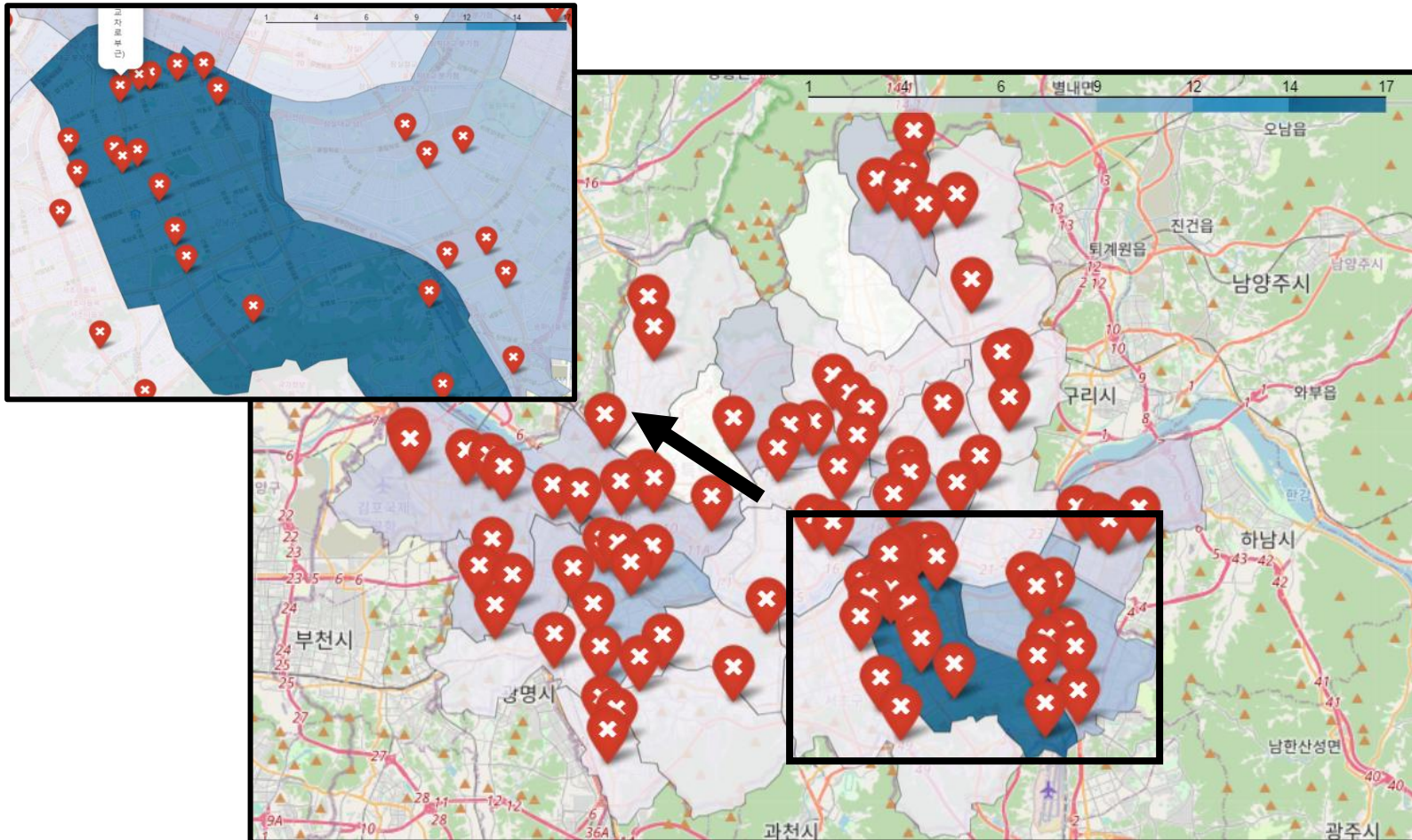
- 적발 수가 많을 수록  크기가 커짐.



## 데이터 시각화

## (5) 음주운전 적발과 사고 지도화 데이터

## 2. 서울시 자치구별 사상자가 높은 다발지역의 수와 해당 위치 마커 표시



```
# 6.2 서울시 자치구별 사상자가 높은 다발지역의 수와 해당 위치 마커 표시
m2 = folium.Map(
    location = [37.566345, 126.977893], zoom_start = 11
)

d = ddz_s23['자치구'].value_counts()

# 사상자가 높은 다발지역 - 자치구별 색칠
folium.Choropleth(
    geo_data = geo_json,
    name = 'choropleth',
    data = d,
    columns = [d.index, d.values],
    key_on = 'feature.properties.name',
    fill_color = 'PuBu',
    fill_opacity=0.8,
    line_opacity = 0.3,
    nan_fill_color='white' # 값이 NaN일 경우 흰색으로 표시
).add_to(m2)

# 음주운전 다발지역 위치를 지도에 마커 추가
for i in range(len(ddz_s23)):
    folium.Marker(location=[ddz_s23.loc[i, '위도'],
                           ddz_s23.loc[i, '경도']],
                  popup = ddz_s23.loc[i, '지점명'],
                  icon=folium.Icon(color='red', icon='glyphicon-remove')).add_to(m2)

m2
```

유형가가 많이 있는 강남구  
주위에 사고가 가장 많이  
발생하며, 특히 음주운전 사고가  
교차로에 많이 일어난다.



## 분석 결과 및 해결방안

# 분석 결과



1. 평일보다 주말에 음주운전 단속 적발 건수가 많을 것이다.



금요일과 주말에는 외출과 음주가 증가하므로  
음주운전 적발 건수가 평일보다 많다.

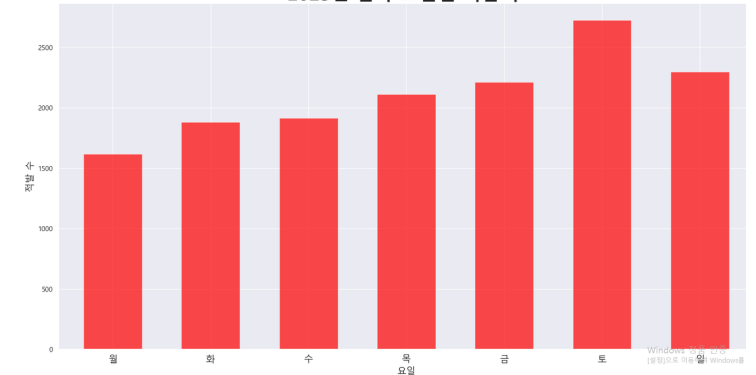


2. 젊은 남성이 음주운전 단속에서 가장 많이 적발될 것이다.

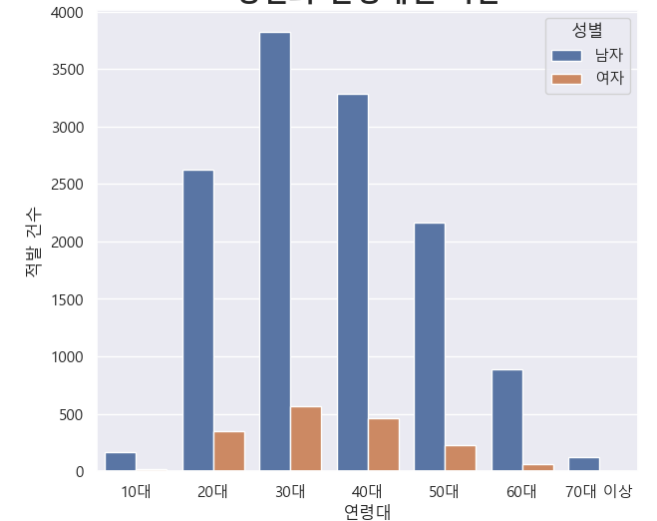


20~40대 남성이 외출과 음주가 빈번하고 음주운전 경향이  
상대적으로 높아, 단속에서 가장 많이 적발될 가능성이 높다.

2023년 단속 요일별 적발 수



성별과 연령대별 적발





# 분석 결과



3. 겨울철(연말)에 밤 시간대에 음주운전 단속 적발 건수가 가장 많을 것이다.



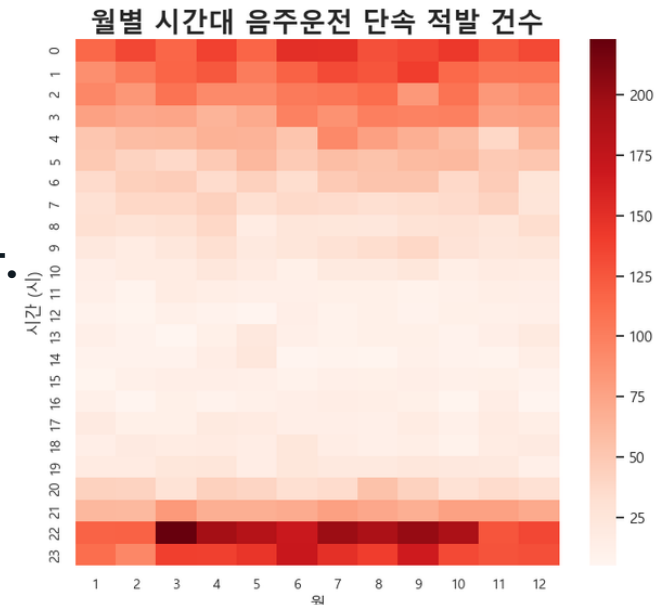
겨울철 밤보다는 여름철에 해가 늦게 지고 휴가철이 집중되는 시기이고, 밤과 새벽 시간대에 음주 후 운전이 많이 발생하기 때문에 단속 적발 건수가 많다.



4. 음주운전 단속 적발 건수가 많을수록 사고 발생 건수와 사상자 수가 높아질 것이다.



음주운전은 사고를 일으킬 위험이 높기 때문에 적발 건수가 많을수록 사고 발생 건수와 그에 따른 사상자 수도 비례적으로 늘어난다.



# 분석 결과

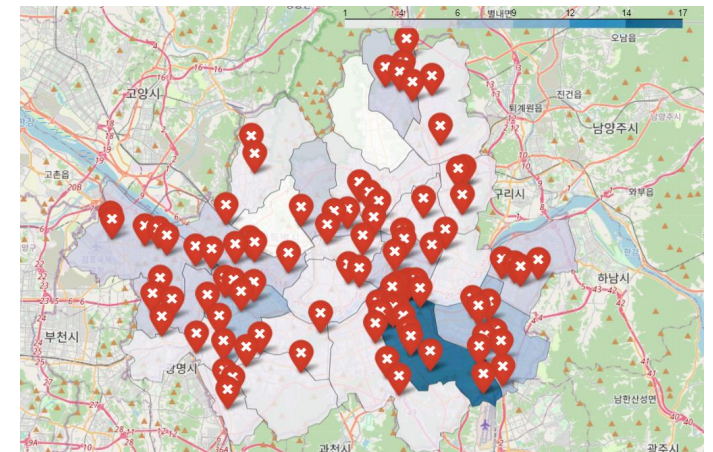
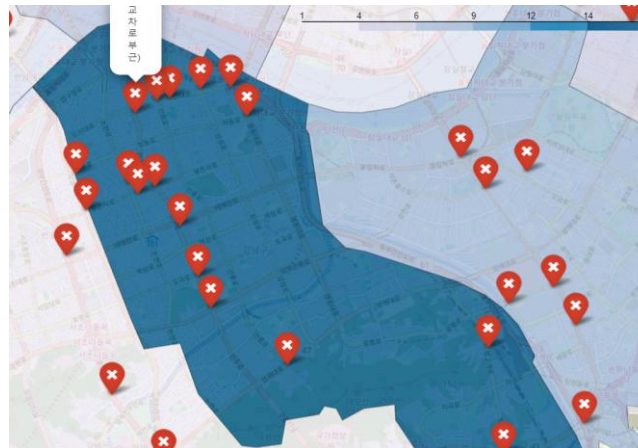
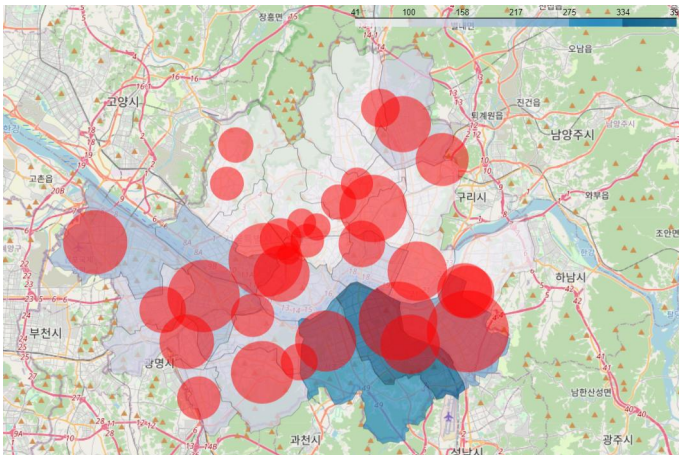


5. 서울시 유흥가에서 음주운전 단속과 사고 발생 건수가 가장 많을 것이다.



서울시 유흥가(강남구, 송파구, 마포구 등)에서는 술을 마신 사람들이 많아 밤늦은 시간대에 차량 이용이 증가하여 음주운전 단속과 사고 발생 건수가 높다.

또한 음주운전은 시야를 좁게 만들고, 판단력과 반응속도가 크게 저하되어 횡단보도, 신호 등이 있는 복잡한 구조의 교차로에서 음주운전 사고가 많이 일어난다.



# 해결 방안



- 음주운전 단속이 사고 발생과 연관성이 있으므로, 사고 예방을 위해 음주 측정 및 순찰 강화
- 특히, 여름철과 금요일, 주말 그리고 밤10시~새벽2시 사이에 경찰의 음주운전 단속을 강화
- 음주운전이 많이 적발되는 20~40대 남성을 중심으로 음주운전 예방 교육과 캠페인을 실시
- 유흥가 주변과 교차로 지점에서 음주 측정대를 추가 배치
- 대중교통 및 대리운전 서비스를 촉진하는 정책을 마련

# 감사합니다.

Thank you ☺



분석카세 1호

32214712 최현석(조장), 32212293 신명섭, 32214548 최석민

