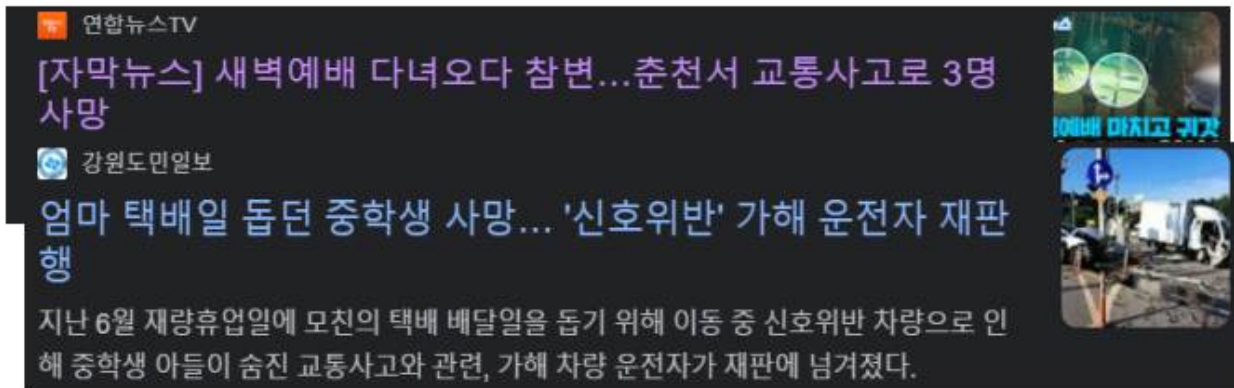
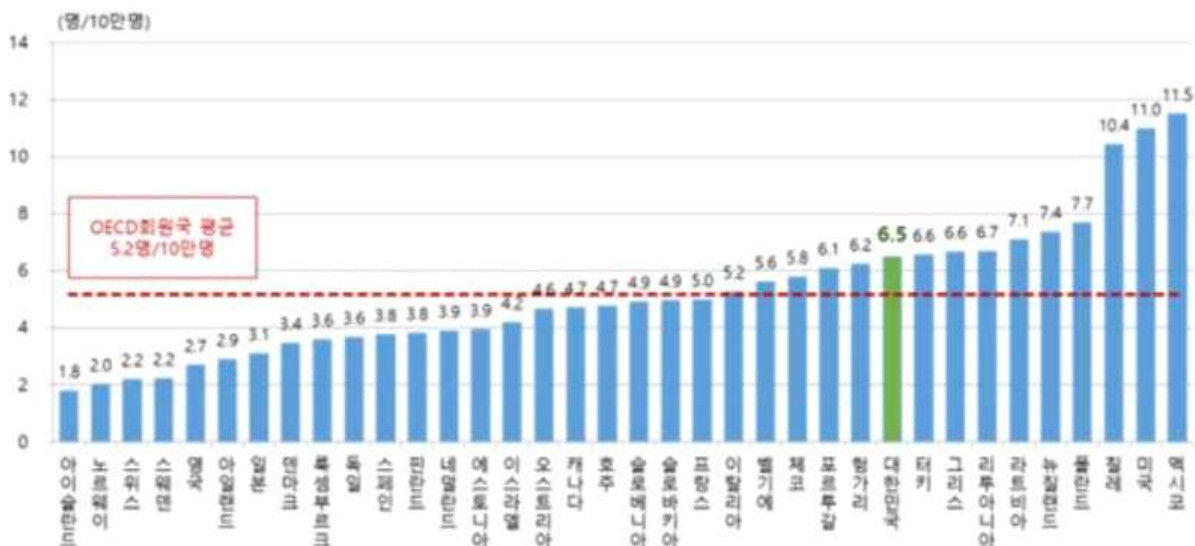


## 1.주제 선정 배경



최근 포털사이트의 메인 뉴스 기사에서는 “새벽예배 다녀오다 참변…춘천시 교통사고로 3명 사망”, “엄마 택배일 돕던 중학생 사망” 등 안타까운 교통사고 사례들이 자주 보도되고 있고, 사회적인 관심을 불러일으키고 있다. 도로교통공단이 국가별 교통사고 현황을 분석한 'OECD 회원국 교통사고 비교보고서 (2021년판)'에 따르면 인구 10만명당 교통사고 사망자 27위, 자동차 1만 대 당 사망자 31위, 특히 교통사고 사망자 중 보행자 비율이 38.9%로 OECD 회원국 가운데 최하위를 기록했다. 이러한 상황에서 최근 몇 년간 시행된 다양한 교통안전 관련 법률들과 더불어 안전속도 5030, 민식이법 등이 시행되었고 교통안전에 대한 개선과 사회적 대응이 필요한 시점에 있다. 교통사고 관련 데이터는 수집·분석을 통해 교통사고 위험을 예측하고 그에 맞는 대책을 세울 수 있기 때문에 교통사고를 예방하는 데 중요한 요소이다. 이러한 맥락에서 대한민국 교통사고 현황을 조사해 보기 위한 데이터를 분석하고자 한다.



- OECD 회원국 인구 10만명 당 교통사고 사망자 비교(2019년 기준) / 표 = 도로교통공단 제공


## 2. 데이터 개요

따라서 본 보고서에는 교통사고정보개방시스템\_사망교통사고정보 데이터를 근거로, 대한민국 교통사고 현황을 살펴보고자 한다.

데이터 소개
교통사고 일시 부터 30일 이내 사망한 경우를 사망교통사고라 정의한다.
데이터 대상
전국 약 42,602건의 교통사고를 대상
데이터 기간
2012년 1월 1일 ~ 2022년 12월 31일
데이터 항목
기본항목(발생년, 발생년월일시, 주야, 요일 등)을 포함하여 인명 피해 통계, 장소 및 사고 유형, 법규 위반 및 도로 형태

- 데이터 출처 : 교통사고정보개방시스템

<https://opendata.koroad.or.kr/api/selectDeathDataSet.do>




로그인 회원가입

오픈 API란?

샘플데이터

오픈 API

커뮤니티



# Open API

Application Program Interface

공공 데이터 개방 추진 사업의 일환으로  
사고다발지정보(13종), 교통안전정보(3종),  
사망사고정보(1종), 교통사고통계(1종)를 개방합니다.

교통사고정보 오픈 API를 검색해 보세요!

검색

사고다발지정보 API

교통안전정보 API

사망사고정보 API

교통사고통계 API

KoROAD 도로교통공단

Copyright (c) 2018 KoROAD. All rights reserved.

opendata.or.kr/api/main.do

(우)26466 강원도 원주시 혁신로 2 도로교통공단 | 문의 : 033-749-5264  
교통사고정보시스템(TAAS)에서 제공된 자료를 활용 시 출처 및 자료링크를 명시해 주시기 바랍니다.

### 3.데이터 가공

#### 1) '사망교통사고' csv 파일 읽어와 df 로 저장

```
df = pd.read_csv('사망교통사고정보.csv', encoding = 'cp949')
df
```

	발생 년	발생년월일 시	주 야	요일	사 망 자 수	사 상 자 수	중 상 자 수	경 상 자 수	부 상 신 고 자 수	발 생 지 시 도	...	사 고 유 형	법 규 위 반	도 로 형 태 _ 대 분 류	도 로 형 태	당 사 자 종 별_1 당_대 분 류	당 사 자 종 별_2 당_대 분 류	발 생 위 치 X_UTMK	발 생 위 치 Y_UTMK	경 도	위 도
0	2012	2012010101	아 간	일	1	1	0	0	0	서울	...	차 도 통 행 중	안 전 운 전 의 무 불 이 행	단 일 로	기 타 단 일 로	승 용 차	보 행 자	949860	1957179	126.931890	37.612680
1	2012	2012010101	아 간	일	1	6	5	0	0	전북	...	정 면 충 돌	중 앙 선 침 범	단 일 로	기 타 단 일 로	승 용 차	승 용 차	946537	1737695	126.909523	35.633956

#### 2) 발생년월일시를 datetime형식으로 변환

```
# 발생년월일시 컬럼을 datetime 형식으로 변환
df['발생년월일시'] = pd.to_datetime(df['발생년월일시'], format = "%Y%m%d%H")
df
```

	발생 년	발생년월 일시	주 야	요일	사 망 자 수	사 상 자 수	중 상 자 수	경 상 자 수	부 상 신 고 자 수	발 생 지 시 도	...	사 고 유 형	법 규 위 반	도 로 형 태 _ 대 분 류	도 로 형 태	당 사 자 종 별_1 당_대 분 류	당 사 자 종 별_2 당_대 분 류	발 생 위 치 X_UTMK	발 생 위 치 Y_UTMK	경 도	위 도
0	2012	2012-01-01 01:00:00	아 간	일	1	1	0	0	0	서울	...	차 도 통 행 중	안 전 운 전 의 무 불 이 행	단 일 로	기 타 단 일 로	승 용 차	보 행 자	949860	1957179	126.931890	37.612680

#### 3) 필요없는 컬럼 삭제

```
# 컬럼 삭제
df = df.drop(['발생지시군구', '요일', '주야', '사고유형_중분류', '사고유형', '도로형태', '당사자종별_1당_대분류', '당사자종별_2당_대분류', '발생위치X_UTMK', '발생위치Y_UTMK'], axis=1)
df.head()
```

	발생 년	발생년월일시	사 망 자 수	사 상 자 수	중 상 자 수	경 상 자 수	부 상 신 고 자 수	발 생 지 시 도	사 고 유 형_대 분 류	법 규 위 반	도 로 형 태_대 분 류	경 도	위 도
0	2012	2012-01-01 01:00:00	1	1	0	0	0	서울	차대사람	안전운전 의무 불이행	단일로	126.931890	37.612680
1	2012	2012-01-01 01:00:00	1	6	5	0	0	전북	차대차	중앙선 침범	단일로	126.909523	35.633956
2	2012	2012-01-01 08:00:00	1	1	0	0	0	충남	차량단독	안전운전 의무 불이행	단일로	126.830281	36.491268
3	2012	2012-01-01 10:00:00	2	2	0	0	0	경남	차대차	과속	교차로	128.155984	35.733503
4	2012	2012-01-01 03:00:00	1	1	0	0	0	경북	차량단독	안전운전 의무 불이행	단일로	128.284180	36.506769

4) 컬럼 이름 변경

```
# 컬럼 이름 변경
df.rename(columns = {'사고유형_대분류':'사고유형','도로형태_대분류':'도로형태'}, inplace = True)
df.head()
```

	발생년	발생년월일시	사망자수	사상자수	중상자수	경상자수	부상신고자수	발생지시도	사고유형	법규위반	도로형태	경도	위도
0	2012	2012-01-01 01:00:00	1	1	0	0	0	서울	차대사람	안전운전 의무 불이행	단일로	126.931890	37.612680
1	2012	2012-01-01 01:00:00	1	6	5	0	0	전북	차대차	중앙선 침범	단일로	126.909523	35.633956
2	2012	2012-01-01 08:00:00	1	1	0	0	0	충남	차량단독	안전운전 의무 불이행	단일로	126.830281	36.491268
3	2012	2012-01-01 10:00:00	2	2	0	0	0	경남	차대차	과속	교차로	128.155984	35.733503
4	2012	2012-01-01 03:00:00	1	1	0	0	0	경북	차량단독	안전운전 의무 불이행	단일로	128.284180	36.506769

5) 결측 데이터 확인

```
# 결측 데이터 확인
df.isnull().sum()
```

```
발생년월일시    0
발생년          0
발생월          0
발생시간        0
발생지시도      0
사망자수        0
사상자수        0
중상자수        0
경상자수        0
부상신고자수    0
사고유형        0
법규위반        0
도로형태        0
경도            0
위도            0
심각도          0
dtype: int64
```

6) 발생일, 발생시간 컬럼 생성

```
# 발생일, 발생시간 컬럼 생성
df['발생월'] = df['발생년월일시'].dt.month
df['발생시간'] = df['발생년월일시'].dt.hour
df
```

	발생년	발생년월일시	주 야	요일	사망 자수	사상 자수	중상 자수	경상 자수	부상신고 자수	발생지 시도	사고 유형	법규위반	도로 형태	경도	위도	발생 월	발생 시간
0	2012	2012-01-01 01:00:00	야	일	1	1	0	0	0	서울	차대 사람	안전운전 의무 불이행	단일로	126.931890	37.612680	1	1

7) 컬럼 순서 재정렬

```
# 컬럼 순서 재정렬
df = df[['발생년월일시','발생년','발생월','발생시간','발생지시도','사망자수','사상자수','중상자수','경상자수','부상신고자수','사고유형','법규위반','도로형태','경도','위도']]
df.head()
```

	발생년월일시	발생년	발생월	발생시간	발생지시도	사망자수	사상자수	중상자수	경상자수	부상신고자수	사고유형	법규위반	도로형태	경도	위도
0	2012-01-01 01:00:00	2012	1	1	서울	1	1	0	0	0	차대사람	안전운전 의무 불이행	단일로	126.931890	37.612680
1	2012-01-01 01:00:00	2012	1	1	전북	1	6	5	0	0	차대차	중앙선 침범	단일로	126.909523	35.633956

## 8) 대물피해환산 점수 컬럼 생성

대물피해환산법이란 비중 '1'로 주어진 물피사고를 기준으로 사망, 중상, 경상, 부상 사고에 상대적인 비중을 두어 가중치를 곱해 합산한 심각도지수로 교통사고의 심각도를 평가하는 방법으로 가중치로는 EPDO 계산 시 일반적으로 사용하는 사망(K) 12, 중상(A) 10, 경상(B) 3, 부상(C) 2, 물피(P) 1로 선정하였고 사고심각도에 대한 기준과 계산식은 다음과 같다.

사망: 교통사고로 인하여 30일 이내에 사망한 경우

중상: 교통사고로 인하여 3주 이상의 치료를 요하는 부상을 입은 경우

경상: 교통사고로 인하여 5일 이상 2주 미만의 치료를 요하는 부상을 입은 경우

부상: 교통사고로 인하여 5일 미만의 치료를 요하는 부상을 입은 경우

물피: 교통사고로 인하여 차량 또는 물건에 손괴를 입은 경우, 본 분석에 사용한 데이터에는 집계되지 않았다.

$$EPDO = (k) \times 12 + (A) \times 10 + (B) \times 3 + (C) \times 2 + (P) \times 1$$

```
df = df.assign(심각도 = df['사망자수'] * 12 + df['중상자수'] * 10 + df['경상자수'] * 3 + df['부상신고자수'] * 2 )
df.head()
```

	발생년월일시	발생 년	발생 월	발생시 간	발생지 시도	사망자 수	사상자 수	중상자 수	경상자 수	부상신고 자수	사고유 형	법규위반	도로형 태	경도	위도	심각 도
0	2012-01-01 01:00:00	2012	1	1	서울	1	1	0	0	0	차대사 람	안전운전 의무 불이행	단일로	126.931890	37.612680	12
1	2012-01-01 01:00:00	2012	1	1	전북	1	6	5	0	0	차대차	중앙선 침범	단일로	126.909523	35.633956	62

## 9) 심각도 분석을 위해 새로운 데이터 프레임 new\_df 생성

```
new_df = df.groupby('발생년').agg({'사망자수': 'sum', '사상자수': 'sum', '심각도': 'sum'})
new_df = new_df.T
new_df
```

발생년	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022
사망자수	5392	5092	4762	4621	4292	4185	3781	3349	3081	2916	2735
사상자수	8665	8068	7222	7282	6493	6390	5696	5121	4642	4241	3889
심각도	86573	81188	73245	73158	65279	63322	56665	50459	45771	42730	39240

10) new\_df 데이터 프레임에 사고건수 컬럼을 추가하기 위해 발생년도별 사고건수 데이터프레임 생성  
df 데이터프레임의 '발생년'열에서 각 고유값의 발생 횟수, 즉 사상자가 발생한 사고건수를 계산하고 그 결과를 df\_counts에 저장하는 과정이다.

```
df_counts = df.groupby('발생년').size().reset_index(name='사고건수')
df_counts
```

	발생년	사고건수
0	2012	5165
1	2013	4876
2	2014	4583
3	2015	4447
4	2016	4119



## 11) '발생년' 컬럼을 기준으로 두 데이터프레임 new\_df 와 df\_counts 병합

```
merged_df = pd.merge(new_df, df_counts, on='발생년')
merged_df
```

	발생년	사망자수	사상자수	심각도	사고건수
0	2012	5392	8665	86573	5165
1	2013	5092	8068	81188	4876
2	2014	4762	7222	73245	4583
3	2015	4621	7282	73158	4447
4	2016	4292	6493	65279	4119
5	2017	4185	6390	63322	4065
6	2018	3781	5696	56665	3657
7	2019	3349	5121	50459	3233
8	2020	3081	4642	45771	2983
9	2021	2916	4241	42730	2816
10	2022	2735	3889	39240	2658

## 11) 지수별 증감률 확인을 위한 컬럼 추가

```
merged_df['사망증감'] = merged_df['사망자수'].diff()
merged_df['사상증감'] = merged_df['사상자수'].diff()
merged_df['건수증감'] = merged_df['사고건수'].diff()
merged_df['심각도증감'] = merged_df['심각도'].diff()
merged_df['사망증감율'] = merged_df['사망증감'] / merged_df['사망자수'].shift() * 100
merged_df['사상증감율'] = merged_df['사상증감'] / merged_df['사상자수'].shift() * 100
merged_df['건수증감율'] = merged_df['건수증감'] / merged_df['사고건수'].shift() * 100
merged_df['심각도증감율'] = merged_df['심각도증감'] / merged_df['심각도'].shift() * 100
merged_df = merged_df[['발생년', '사망자수', '사상자수', '사고건수', '심각도', '사망증감', '사상증감',
                        '건수증감', '심각도증감', '사망증감율', '사상증감율', '건수증감율', '심각도증감율']]
merged_df
```

	발생년	사망자수	사상자수	사고건수	심각도	사망증감	사상증감	건수증감	심각도증감	사망증감율	사상증감율	건수증감율	심각도증감율
0	2012	5392	8665	5165	86573	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	2013	5092	8068	4876	81188	-300.0	-597.0	-289.0	-5385.0	-5.563798	-6.889786	-5.595353	-6.220184
2	2014	4762	7222	4583	73245	-330.0	-846.0	-293.0	-7943.0	-6.480754	-10.485870	-6.009024	-9.783466

## 12) 결측치 처리

```
merged_df.fillna(0, inplace=True)
merged_df
```

	발생년	사망자수	사상자수	사고건수	심각도	사망증감	사상증감	건수증감	심각도증감	사망증감율	사상증감율	건수증감율	심각도증감율
0	2012	5392	8665	5165	86573	0.0	0.0	0.0	0.0	0.000000	0.000000	0.000000	0.000000
1	2013	5092	8068	4876	81188	-300.0	-597.0	-289.0	-5385.0	-5.563798	-6.889786	-5.595353	-6.220184

### 3.시각화 / 분석

#### 1. 연도별 사망사고건수

```
# 연도별 사망사고건수 확인
df_year = df['발생년'].value_counts()
df_year.head()
```

```
2012    5165
2013    4876
2014    4583
2015    4447
2016    4119
Name: 발생년, dtype: int64
```

##### 1-1. 발생년도별 사망사고건수 시각화

```
# 발생년도별 사망사고건수 데이터 시각화 (막대그래프)
df['발생년'].value_counts().plot(kind = 'bar', rot = 0)
plt.title('발생년도별 사망사고건수 시각화')
plt.xlabel("발생년")
plt.ylabel("사망사고건수")
plt.show()
```



- 교통사고 건수는 2012년부터 2022년까지 계속 감소하는 추세를 보인다.

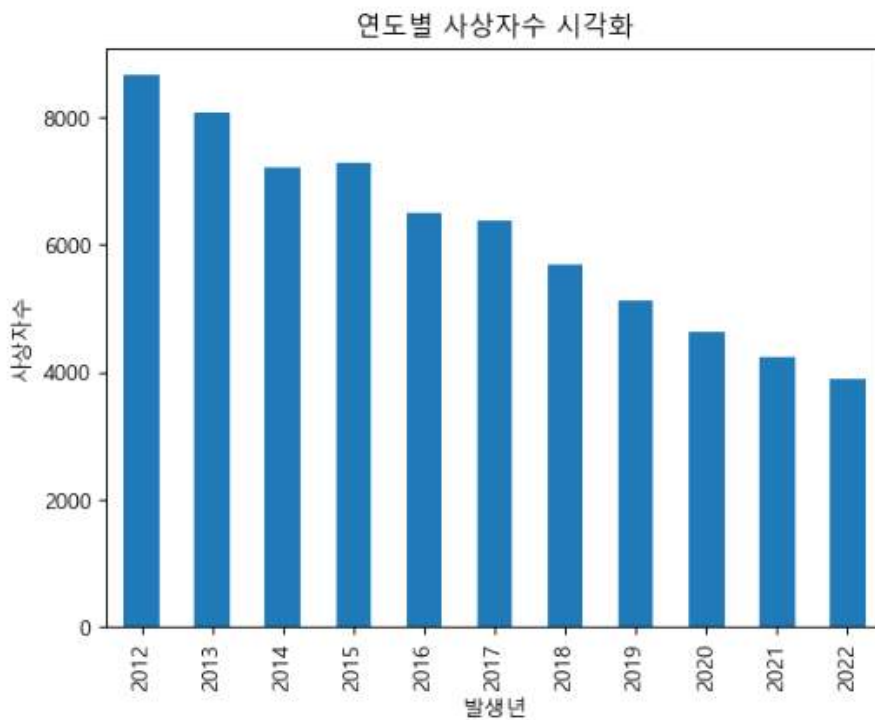
## 2. 연도별 사상자수

```
# 연도별 사상자수
years = df.groupby('발생년')['사상자수'].sum()
years
```

```
발생년
2012    8665
2013    8068
2014    7222
2015    7282
2016    6493
2017    6390
2018    5696
2019    5121
2020    4642
2021    4241
2022    3889
Name: 사상자수, dtype: int64
```

### 2-1. 발생년도별 사상자수 시각화

```
# 연도별 사상자수 시각화
df.groupby('발생년')['사상자수'].sum().plot(kind = 'bar')
plt.title('연도별 사상자수 시각화')
plt.xlabel("발생년")
plt.ylabel("사상자수")
plt.show()
```

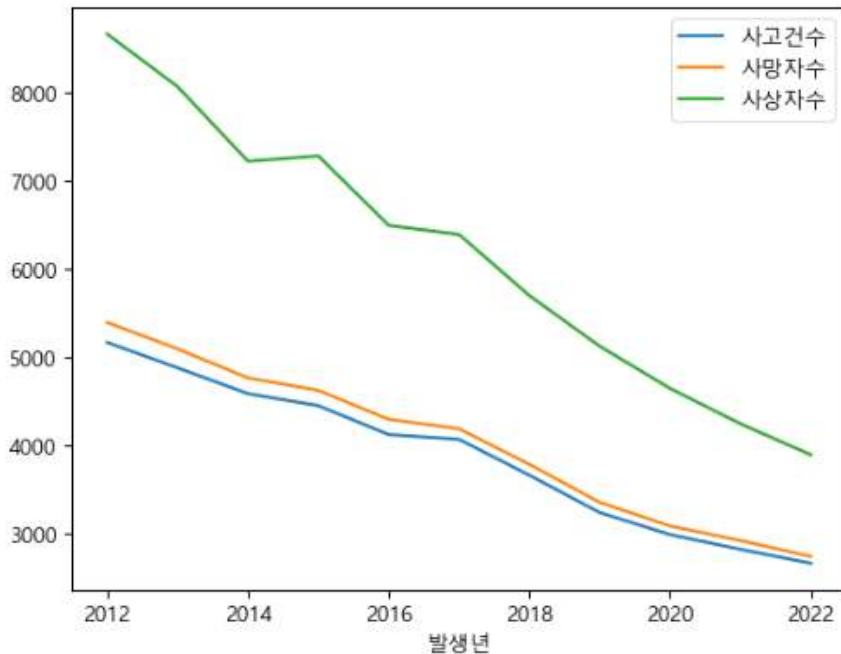


- 사망자 사고 건수와 마찬가지로 2012년부터 2022년까지 감소하는 추세를 보인다.



### 3. 사고건수, 사망자수, 사상자수 비교

```
# 연도별 사고건수, 사망자수, 사상자수를 선 그래프로 시각화
df['발생년'].value_counts().plot(label = '사고건수')
df.groupby('발생년')['사망자수'].sum().plot(label = '사망자수')
df.groupby('발생년')['사상자수'].sum().plot(label = '사상자수')
plt.legend()
plt.show()
```



- 사고 건수가 줄어들에 따라 사망자 수 사상자 수가 줄어드는 것을 확인할 수 있다.
- 사상자 수가 상대적으로 크게 나타나는 이유는 한 건의 교통사고에서 사상자가 많이 발생할 수 있기 때문이라고 추측된다.

#### 4. 연도별 심각도 변화 시각화

```
plt.bar(merged_df['발생년'], merged_df['심각도'], width=0.5, color='g', label='심각도지수')
plt.plot(merged_df['발생년'], merged_df['심각도'], color='r')

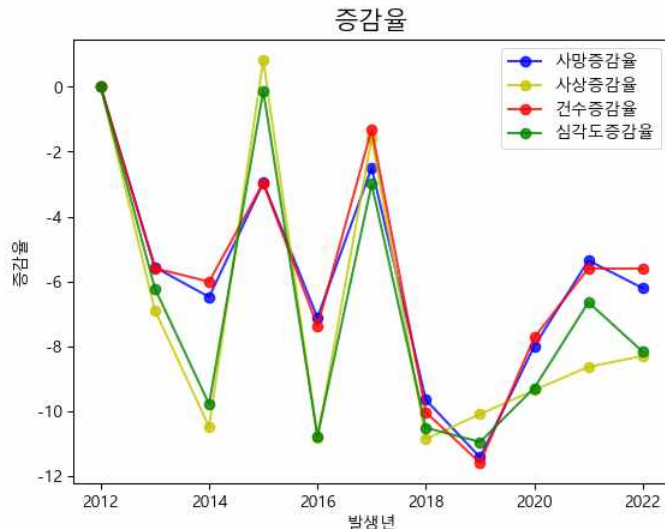
plt.title('년도별 심각도 변화', fontsize=15)
plt.ylabel('심각도', fontsize=10)
plt.xlabel('발생년', fontsize=10)
plt.legend()
plt.xticks(merged_df['발생년'])
plt.show()
```



- 사망사고건수와 사상자수 그래프와 마찬가지로 심각도 또한 감소하는 추세를 보이고 있다.

## 5. 지수별 증감을 시각화

```
In [22]: plt.plot(merged_df['발생년'], merged_df['사망증감율'], color='b', alpha=0.8, marker='o', label='사망증감율')
plt.plot(merged_df['발생년'], merged_df['사상증감율'], color='y', alpha=0.8, marker='o', label='사상증감율')
plt.plot(merged_df['발생년'], merged_df['건수증감율'], color='r', alpha=0.8, marker='o', label='건수증감율')
plt.plot(merged_df['발생년'], merged_df['심각도증감율'], color='g', alpha=0.8, marker='o', label='심각도증감율')
plt.title('증감율', fontsize=15)
plt.ylabel('증감율', fontsize=10)
plt.xlabel('발생년', fontsize=10)
plt.legend()
plt.show()
```



사망자수 증감율: 사망자수의 증감율은 이전 연도 대비 현재 연도의 사망자수의 증감을 나타냄

사상자수 증감율: 이전 연도 대비 현재 연도의 사상자수의 증감을 나타냄

사고건수 증감율: 사상자 발생 사고건수의 증감을 나타냄

심각도 증감율: 이전 연도 대비 현재 연도의 심각도의 증감을 나타냄

정부는交通安全 종합대책을 수립, '안전속도 5030'을 중점 추진과제로 2022년 까지 교통사고 사망자수를 2016년 4,292명의 절반 수준인 2,000명 대로 감축하겠다는 목표를 수립했다. 교통사고 사망자는 2012년 5,392명에서 2022년 2,735명으로 꾸준히 감소하였다. '안전속도 5030'이 도로교통법 시행규칙에 개정된 2019년 3,349명 2020년 3,081명, 2019년 2,916명, 2022년 2,735명을 기록하였으며 2018-2019년에 432명으로 사망자수가 가장 많이 감소하였다.

사망자발생 사고 건수는 2012년 5,165건에 최다발생 하였고 2013년 4,876건으로 그 뒤를 이었다. 마찬가지로 '안전속도 5030'이 도로교통법 시행규칙에 개정된 2019년 3,233건 2020년 2,983건, 2019년 2,816건, 2022년 2,658건을 기록하였으며 특히, 2018-2019년에 432건 감소로 사망사고 건수가 가장 많이 감소하였다.

심각도 역시 모든 연도에서 감소하고 있음을 확인 할 수 있고 2018-2019년 11%로 큰 폭으로 감소하였다

종합적으로 모든 항목에서 2012년부터 2022년 까지 감소하였고 특히 2018년에서 2019년까지의 큰 감소세를 보이고있는데, 이는 '안전속도 5030'이 교통사고 감축에 효과를 보이는 것으로 판단된다. 하지만 본 분석에 사용한 사고 데이터는 물피사고를 집계하지 않았으며 단순히 명목적인 시계열 자료만을 사용하여 정확한 결과를 위해 세부적인 사고자료의 분류와 이를 활용한 분석이 필요할 것으로 시사된다.

## 6. 법규위반과 사고유형에 따른 심각도 분석을 위한 피벗테이블 생성

```
# 법규위반과 사고유형에 따른 심각도 분석을 위한 피벗테이블 생성
pv1 = pd.pivot_table(df, values='심각도', index='법규위반', columns='사고유형')
pv1
```

	사고유형	차대사람	차대차	차량단독	철길건널목
법규위반					
	과속	13.403433	22.966355	27.874126	NaN
교차로 통행방법 위반		12.689655	16.766871	12.000000	NaN
	기타	14.152542	17.603908	17.179487	17.6
보행자 보호의무 위반		12.804023	16.769231	NaN	NaN
	신호위반	13.032825	19.143485	16.315789	NaN
안전거리 미확보		13.966667	18.501445	13.464286	NaN
안전운전 의무 불이행		12.619649	18.358778	14.850278	23.0
	중앙선 침범	14.467742	23.206920	16.018868	NaN

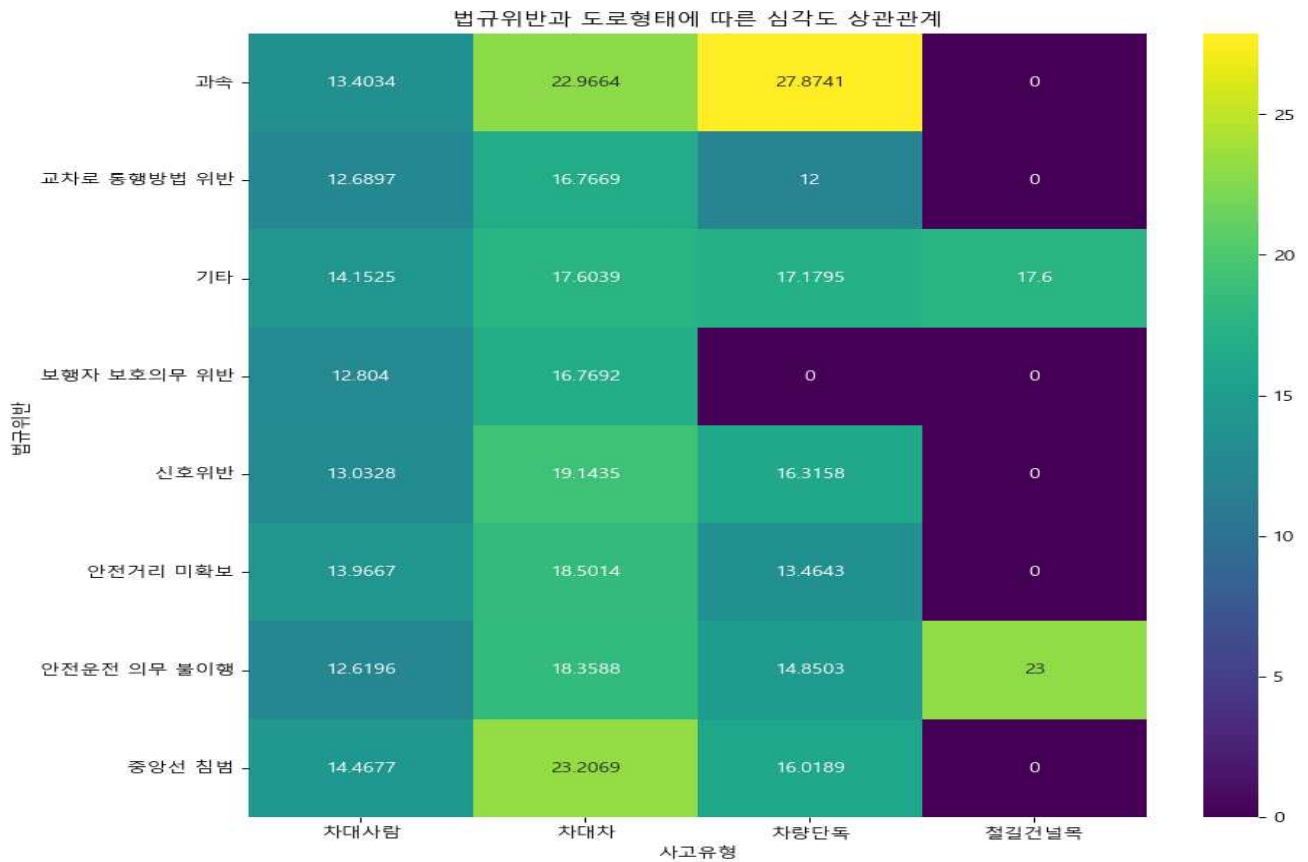
### 6-1 결측치 처리

```
# 결측치 처리
pv1 = pv1.fillna(0)
pv1
```

	사고유형	차대사람	차대차	차량단독	철길건널목
법규위반					
	과속	13.403433	22.966355	27.874126	0.0
교차로 통행방법 위반		12.689655	16.766871	12.000000	0.0
	기타	14.152542	17.603908	17.179487	17.6
보행자 보호의무 위반		12.804023	16.769231	0.000000	0.0
	신호위반	13.032825	19.143485	16.315789	0.0
안전거리 미확보		13.966667	18.501445	13.464286	0.0
안전운전 의무 불이행		12.619649	18.358778	14.850278	23.0
	중앙선 침범	14.467742	23.206920	16.018868	0.0

### 6-2 법규위반과 도로형태에 따른 심각도 상관관계 히트맵 그래프 시각화

```
# 법규위반과 도로형태에 따른 심각도 상관관계 히트맵 그래프
plt.figure(figsize=(10, 10))
sns.heatmap(pv1, annot=True, cmap='viridis', fmt='g')
plt.title('법규위반과 도로형태에 따른 심각도 상관관계')
plt.show()
```



-과속: "차대사람", "차대차", "차량단독"에서 모두 높은 심각도를 보이고있다. 특히 "차량단독"에서 가장 높은 심각도를 나타낸다. 과속이 교통사고에서 높은 심각도를 초래할 가능성이 있음을 시사

-교차로 통행방법 위반: "차대사람"과 "차대차"에서 높은 심각도를 나타낸다. 교차로 통행방법 위반은 교차로에서의 교통사고에서 상대적으로 높은 심각도를 가질 수 있음을 시사

-안전의무 불이행: "차대사람"과 "차량단독"에서 높은 심각도를 보이며, 특히 "절길건널목"에서 가장 높은 심각도를 나타낸다. 안전운전 의무 불이행은 높은 심각도를 초래할 수 있음을 시사

-신호위반: 모든 사고유형에서 상대적으로 높은 심각도를 나타낸다.

-안전거리 미확보: "차대사람"과 "차대차"에서 높은 심각도를 나타낸다.

-보행자 보호의무 위반: "보행자 보호의무 위반"은 "차대사람"과 "차대차"에서 높은 심각도를 나타낸다.

-중앙선 침범: "중앙선 침범"은 "차대사람"과 "차대차"에서 높은 심각도를 보이며, 도로 안전 규칙을 지키지 않는 행위의 위험성을 시사

## 7. 법규위반과 도로형태에 따른 심각도 분석을 위한 피벗테이블 생성

```
# 법규위반과 도로형태에 따른 심각도 분석
```

```
pv2 = pd.pivot_table(df, values='심각도', index='법규위반', columns='도로형태')
pv2
```

	도로형태	교차로	기타	기타/불명	단일로	불명	주차장	철길건널목
법규위반								
과속	17.209302	16.666667	16.200000	18.224835	NaN	NaN	NaN	NaN
교차로 통행방법 위반	16.701807	12.000000	NaN	12.315789	NaN	NaN	NaN	NaN
기타	16.898190	14.044444	14.190476	16.433962	12.000000	NaN	NaN	17.6
보행자 보호의무 위반	12.817544	12.000000	12.000000	12.956772	NaN	NaN	NaN	NaN
신호위반	18.167185	16.090909	31.250000	13.705882	12.000000	NaN	NaN	NaN
안전거리 미확보	15.881481	22.214286	17.222222	18.562500	NaN	NaN	NaN	NaN
안전운전 의무 불이행	14.167545	14.116239	14.455115	15.252569	13.428571	14.333333	23.0	23.0
중앙선 침범	20.604706	23.132075	22.684211	22.876964	19.250000	NaN	NaN	NaN

### 7-1 결측치 처리

```
# 결측치 처리
```

```
pv2 = pv2.fillna(0)
pv2
```

	도로형태	교차로	기타	기타/불명	단일로	불명	주차장	철길건널목
법규위반								
과속	17.209302	16.666667	16.200000	18.224835	0.000000	0.000000	0.0	0.0
교차로 통행방법 위반	16.701807	12.000000	0.000000	12.315789	0.000000	0.000000	0.0	0.0
기타	16.898190	14.044444	14.190476	16.433962	12.000000	0.000000	17.6	17.6
보행자 보호의무 위반	12.817544	12.000000	12.000000	12.956772	0.000000	0.000000	0.0	0.0
신호위반	18.167185	16.090909	31.250000	13.705882	12.000000	0.000000	0.0	0.0
안전거리 미확보	15.881481	22.214286	17.222222	18.562500	0.000000	0.000000	0.0	0.0
안전운전 의무 불이행	14.167545	14.116239	14.455115	15.252569	13.428571	14.333333	23.0	23.0
중앙선 침범	20.604706	23.132075	22.684211	22.876964	19.250000	0.000000	0.0	0.0

### 7-2 법규위반과 도로형태에 따른 심각도 상관관계 히트맵 그래프 시각화

```
plt.figure(figsize=(10, 6))
sns.heatmap(pv2, annot=True, cmap='coolwarm', fmt='g')
plt.title('법규위반과 도로형태에 따른 심각도')
plt.show()
```





-과속: "단일로"에서 과속에 의한 교통사고의 심각도가 가장 높았다. 이는 주로 고속도로나 국도 등의 단일 도로에서의 과속이 교통사고의 심각성을 증가시킨다는 것을 시사하며, "단일로" 뿐만 아니라 다른 도로에서도 높은 심각도 수치를 나타내고 있다.

-보행자 보호의무 위반: "보행자 보호의무 위반"은 모든 도로에서 비슷한 수치를 나타낸다. 이는 도로 형태에 상관없이 발생할 수 있는 교통사고의 일반적인 원인임을 알 수 있다.

-신호위반: "기타/불명"에서 신호위반에 따른 교통사고의 심각도가 가장 높았지만, "단일로"와 "교차로"에서도 높은 수치를 보이며, 신호위반이 심각한 교통사고를 초래하는 것으로 해석할 수 있다.

-안전거리 미확보: "기타/불명", "단일로"에서 안전거리 미확보에 의한 교통사고의 심각도가 높다. 고속도로 등에서의 차간 거리 유지에 대한 중요성 강조를 시사

-안전운전 의무 불이행: "안전운전 의무 불이"행은 철길건널목을 제외한 모든 도로에서 비슷한 수치를 나타낸다. 이는 도로에 상관없이 발생할 수 있는 교통사고의 일반적인 원인임을 알 수 있다. 하지만 철길건널목에서 눈에 띄게 높은 수치를 보이고 있는데, 보통 인적이 드문 지역의 철길 건널목을 지나는 운전자는 그 건널목은 늘 열차가 다니지 않던 곳이고 인지하고 있기 때문에 이러한 부주의함에서 사고가 발생한다고 추측할 수 있다.

- 중앙선 침범: 모든 도로에서 중앙선 침범에 의한 교통사고의 심각도가 매우 높다. 모든 도로에서 중앙선 침범이 교통사고의 가장 큰 원인 중 하나로 작용하는 것으로 보인다.

8. 사고유형과 도로형태에 따른 심각도 분석을 위한 피벗테이블 생성

```
pv3 = pd.pivot_table(df, values='심각도', index='사고유형', columns='도로형태')
pv3
```

도로형태	교차로	기타	기타/불명	단일로	불명	주차장	철길건널목
사고유형							
차대사람	12.654630	13.048872	12.687500	12.832902	12.0	14.105263	NaN
차대차	17.996769	18.616667	20.559006	20.264811	15.9	13.500000	NaN
차량단독	14.914626	13.436441	13.822581	15.227641	14.0	15.333333	NaN
철길건널목	NaN	NaN	NaN	NaN	NaN	NaN	18.5

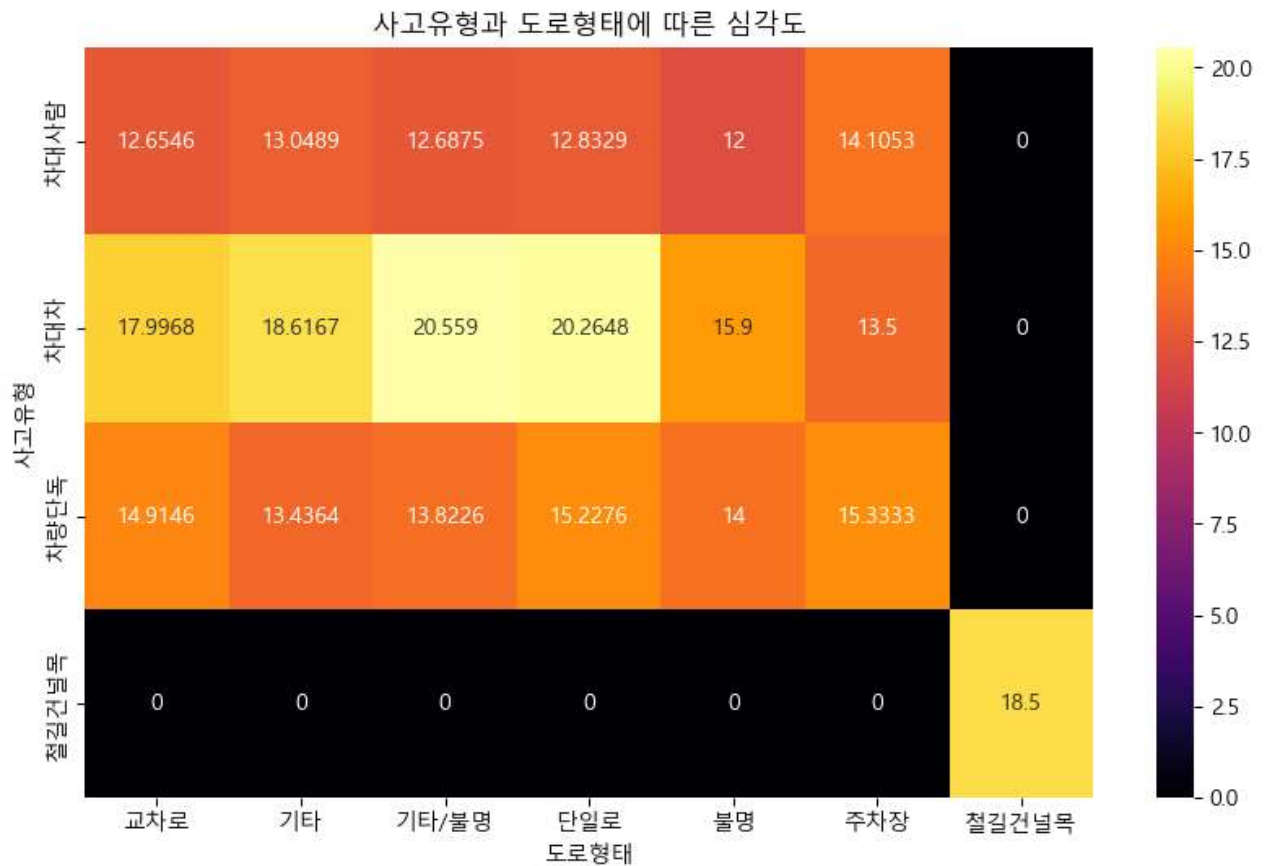
8-1 결측치 처리

```
# 결측치 처리
pv3 = pv3.fillna(0)
pv3
```

도로형태	교차로	기타	기타/불명	단일로	불명	주차장	철길건널목
사고유형							
차대사람	12.654630	13.048872	12.687500	12.832902	12.0	14.105263	0.0
차대차	17.996769	18.616667	20.559006	20.264811	15.9	13.500000	0.0
차량단독	14.914626	13.436441	13.822581	15.227641	14.0	15.333333	0.0
철길건널목	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	18.5

8-2 사고유형과 도로형태에 따른 심각도 히트맵 그래프 시각화

```
plt.figure(figsize=(10, 6))
sns.heatmap(pv3, annot=True, cmap='inferno', fmt='g')
plt.title('사고유형과 도로형태에 따른 심각도')
plt.show()
```



차대사람: "주차장"에서의 차대사람 사고의 심각도가 높는데, 주차장에서의 차량 속도가 낮음에도 불구하고, 운전자들이 주의를 기울이지 않아 발생한 것으로 추측된다. 이 외에 차대사람 사고유형은 모든 도로에서 비슷한 수치를 나타낸다.

차대차: 단일로와 기타/불명 도로형태에서 차대차 사고의 심각도가 높게 나타났다. 특히 단일로에서의 사고가 두드러지며, 이는 차량이 서로 마주치는 구간에서의 충돌이 더 위험하다는 것을 시사한다.

차량단독: 차량 단독사고란 차량이나 보행자와 충돌하는 게 아니라 가드레일, 전봇대 등 도로 시설물과 부딪히거나 차량이 뒤집혀 발생한 사고를 의미한다. 주로 주차장과 단일로에서 높은 심각도를 보이고 있으며, 차량 단독 사고의 원인은 주로 과속, 도로 이탈, 운전자의 주의 부족 등이 될 수 있다

## 9. 발생년도에 따른 지역별 사상자수

-데이터프레임 df를 '발생지시도', '발생년', '발생월' 열을 기준으로 그룹화

```
# 발생지시도별 발생년, 월별 사상자 발생 교통사고
year_month_casuality = df.groupby(['발생지시도', '발생년', '발생월'])['사상자수'].sum().reset_index()
year_month_casuality
```

	발생지시도	발생년	발생월	사상자수
0	강원	2012	1	20
1	강원	2012	2	121

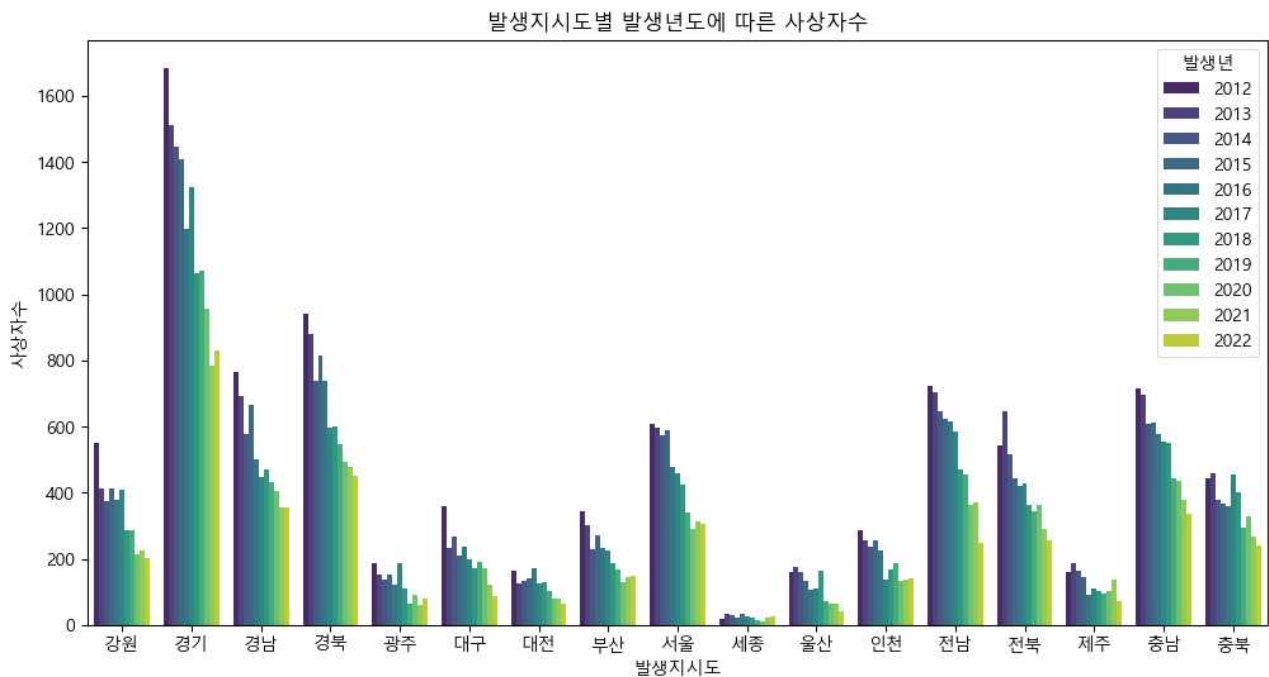
### 9-1 발생년도에 따른 지역별 사상자수의 합 데이터프레임 생성

```
# 발생년도에 따른 지역별 사상자수의 합 데이터 프레임 생성
year_casuality = year_month_casuality.groupby(['발생년', '발생지시도'])['사상자수'].sum().reset_index()
year_casuality
```

	발생년	발생지시도	사상자수
0	2012	강원	551
1	2012	경기	1684

### 9-2 발생년도에 따른 지역별 사상자수 시각화

```
plt.figure(figsize=(12, 6))
sns.barplot(x='발생지시도', y='사상자수', hue='발생년', data=year_casuality, palette='viridis')
plt.title('발생지시도별 발생년도에 따른 사상자수')
plt.show()
```



-대부분의 지역에서 2022년으로 갈수록 사상자수가 줄어드는 것을 확인할 수 있다.

## 10. 발생월에 따른 지역별 사상자수

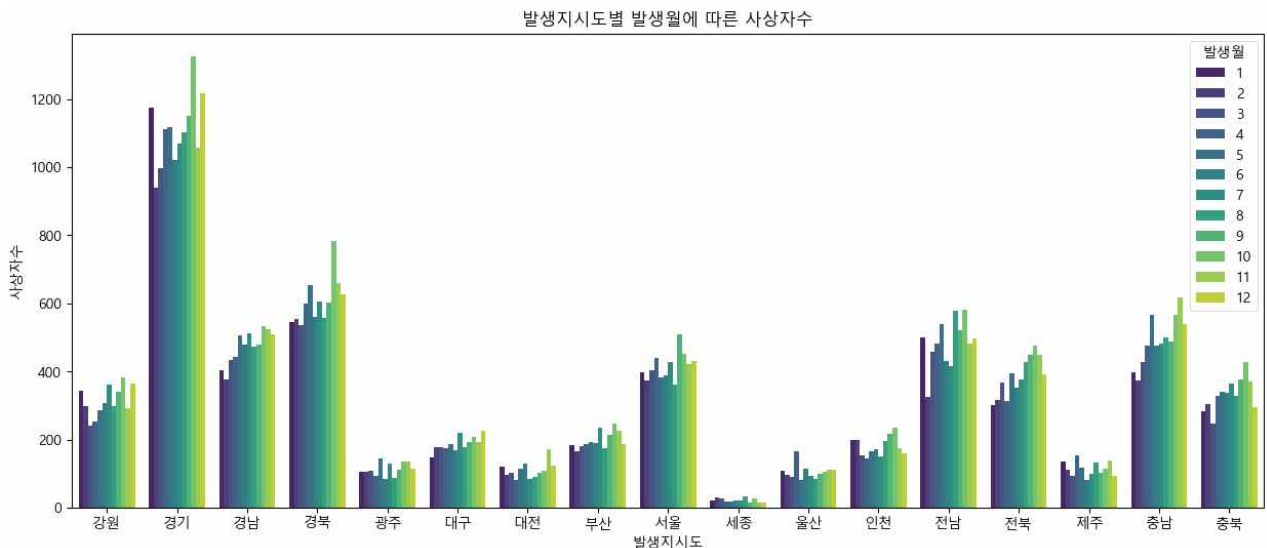
-발생월에 따른 지역별 사상자수의 합 데이터프레임 생성

```
# 발생지시도별 월별 사상자수
month_casualty = year_month_casualty.groupby(['발생월', '발생지시도'])['사상자수'].sum().reset_index()
month_casualty
```

발생월	발생지시도	사상자수
0	1	강원 342
1	1	경기 1173
2	1	경남 405
3	1	경북 546
4	1	광주 104
...	...	...

### 10-1. 발생월에 따른 지역별 사상자수 시각화

```
plt.figure(figsize=(15, 6))
sns.barplot(x='발생지시도', y='사상자수', hue='발생월', data=month_casualty, palette='viridis')
plt.title('발생지시도별 발생월에 따른 사상자수')
plt.show()
```



대부분의 지역에서 10~11월에 교통사고 사망자수가 많다는 것을 확인 할 수 있다. 대표적인 가을 행락철로 이동인구가 늘어나는 것이 원인이다. 운전자들에게 교통사고 위험성에 대한 경각심을 상기시키고, 과속, 졸음운전 등 교통사고 유발 위험 운전예 대한 특별 단속과 예방 홍보를 강화할 필요가 있다

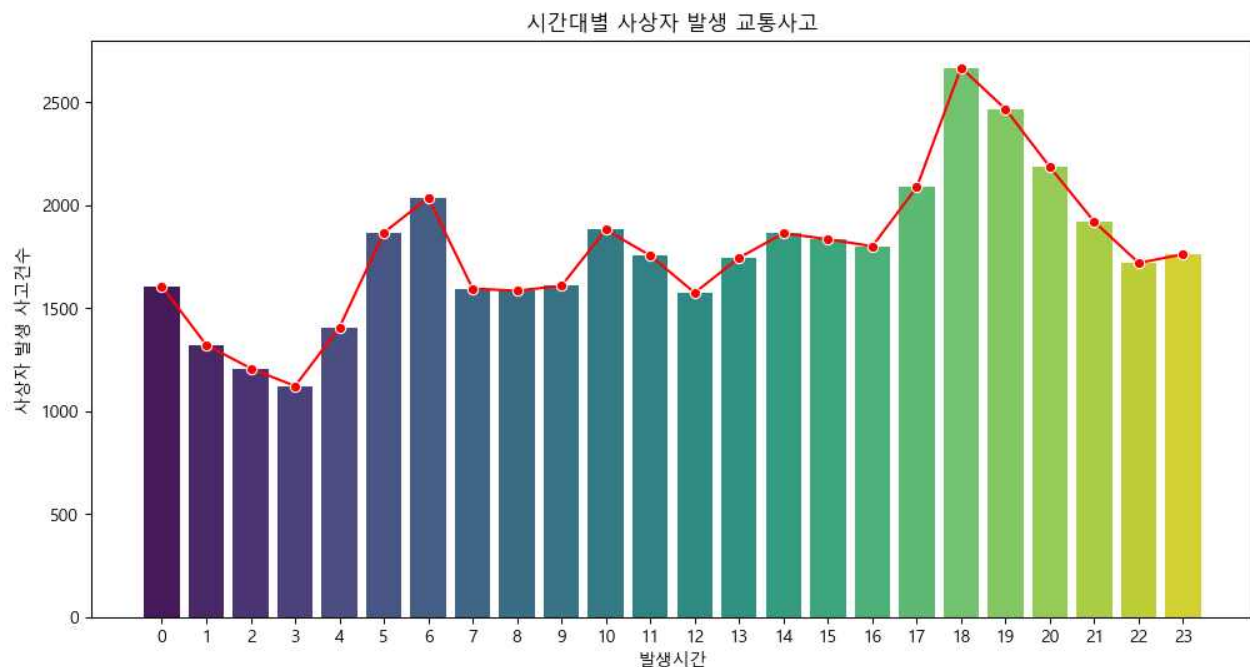
## 11. 시간대별 사상자 발생 교통사고 건수

```
hour_casualties = df.groupby('발생시간')['사상자수'].count().reset_index() # count로 사상자수의 합이 아닌 사상자수 발생 건수
hour_casualties
```

발생시간	사상자수
0	0 1606
1	1 1321
2	2 1207
3	3 1121
4	4 1406

### 11-1 시간대별 사상자 발생 교통사고 건수 시각화

```
plt.figure(figsize=(12, 6))
sns.barplot(x='발생시간', y='사상자수', data=hour_casualties, palette='viridis')
sns.lineplot(x='발생시간', y='사상자수', data=hour_casualties, color='red', marker='o')
plt.title('시간대별 사상자 발생 교통사고')
plt.xlabel('발생시간')
plt.ylabel('사상자 발생 사고건수')
plt.show()
```



교통량이 많은 출근시간대와 퇴근시간대에 주로 발생하는 것을 확인 할 수 있다.



## 12. 서울지역의 데이터를 활용한 교통사고 데이터 분석

# 서울 데이터만 추출

```
seoul = df[df['발생지시도'] == '서울']
seoul
```

	발생년월일시	발생 년	발생 월	발생시 간	발생지 시도	사망자 수	사상자 수	중상자 수	경상자 수	부상신고 자수	사고유 형	법규위반	도로형 태	경도	위도	심각 도
0	2012-01-01 01:00:00	2012	1	1	서울	1	1	0	0	0	차대사 람	안전운전 의무 불이행	단일로	126.931890	37.612680	12
9	2012-01-01 02:00:00	2012	1	2	서울	1	4	0	3	0	차대차	신호위반	교차로	126.993477	37.568273	21
16	2012-01-01 02:00:00	2012	1	2	서울	1	1	0	0	0	차대사 람	안전운전 의무 불이행	교차로	127.034833	37.483840	12
29	2012-01-01 18:00:00	2012	1	18	서울	1	1	0	0	0	차대사 람	안전운전 의무 불이행	단일로	126.903486	37.520151	12
37	2012-01-03 07:00:00	2012	1	7	서울	1	2	0	1	0	차대차	안전거리 미확 보	단일로	126.947969	37.481835	15
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

### 12-1 날씨 데이터 불러오기

# 날씨 데이터 불러오기

```
weather_seoul = pd.read_csv('서울_날씨.csv', encoding = 'cp949')
weather_seoul
```

	날짜	평균기온(°C)
0	2012-01-01	-3.0
1	2012-01-02	-4.8
2	2012-01-03	-4.5
3	2012-01-04	-7.4
4	2012-01-05	-5.7
...	...	...

### 12-2 결측치 제거 및 컬럼 수정

```
weather_seoul['날짜'] = weather_seoul['날짜'].str.strip() # 탭 문자 제거
weather_seoul = weather_seoul.dropna(subset=['날짜', '평균기온(°C)']) # 결측치 제거
weather_seoul.rename(columns = {'평균기온(°C)': '평균기온'}, inplace = True) # 컬럼 이름 변경
weather_seoul
```

	날짜	평균기온
0	2012-01-01	-3.0
1	2012-01-02	-4.8
2	2012-01-03	-4.5
3	2012-01-04	-7.4
4	2012-01-05	-5.7
...	...	...

### 12-3 날짜의 데이터타입을 datetime 으로 변경

# 날짜 컬럼 데이터타입을 datetime 으로 변경

```
weather_seoul['날짜'] = pd.to_datetime(weather_seoul['날짜'])
weather_seoul.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4018 entries, 0 to 4017
Data columns (total 2 columns):
#   Column  Non-Null Count  Dtype
---  ---
0   날짜    4018 non-null      datetime64[ns]
1   평균기온 4018 non-null      float64
dtypes: datetime64[ns](1), float64(1)
memory usage: 94.2 KB
```

## 12-4 데이터 결합

```
df_combined = pd.merge(seoul, weather_seoul, left_on=seoul['발생년월일시'].dt.date, right_on=weather_seoul['날짜'].dt.date, how='left')
df_combined
```

	key_0	발생년월일시	발생 년	발생 월	발생 시간	발생지 시도	사망 자수	사상 자수	중상 자수	경상 자수	부상신 고자수	사고 유형	법규위반	도로 형태	경도	위도	심각 도	날짜	평균 기온
0	2012-01-01	2012-01-01 01:00:00	2012	1	1	서울	1	1	0	0	0	차대 사람	안전운전 의무 불이행	단일 로	126.931890	37.612680	12	2012-01-01	-3.0

## 12-5 불필요한 컬럼 제거

```
df_combined = df_combined.drop(columns=['key_0', '날짜'])
df_combined
```

	발생년월일시	발생 년	발생 월	발생 시간	발생지 시도	사망 자수	사상 자수	중상 자수	경상 자수	부상신 고자수	사고 유형	법규위반	도로 형태	경도	위도	심각 도	평균 기온
0	2012-01-01 01:00:00	2012	1	1	서울	1	1	0	0	0	차대 사람	안전운전 의무 불이행	단일 로	126.931890	37.612680	12	-3.0
1	2012-01-01 02:00:00	2012	1	2	서울	1	4	0	3	0	차대 자	신호위반	교차 로	126.993477	37.568273	21	-3.0

## 12-6 필요한 열만 선택

```
# 필요한 열만 선택
df_selected = df_combined[['발생년월일시', '평균기온', '사상자수']]
df_selected
```

	발생년월일시	평균기온	사상자수
0	2012-01-01 01:00:00	-3.0	1
1	2012-01-01 02:00:00	-3.0	4
2	2012-01-01 02:00:00	-3.0	1

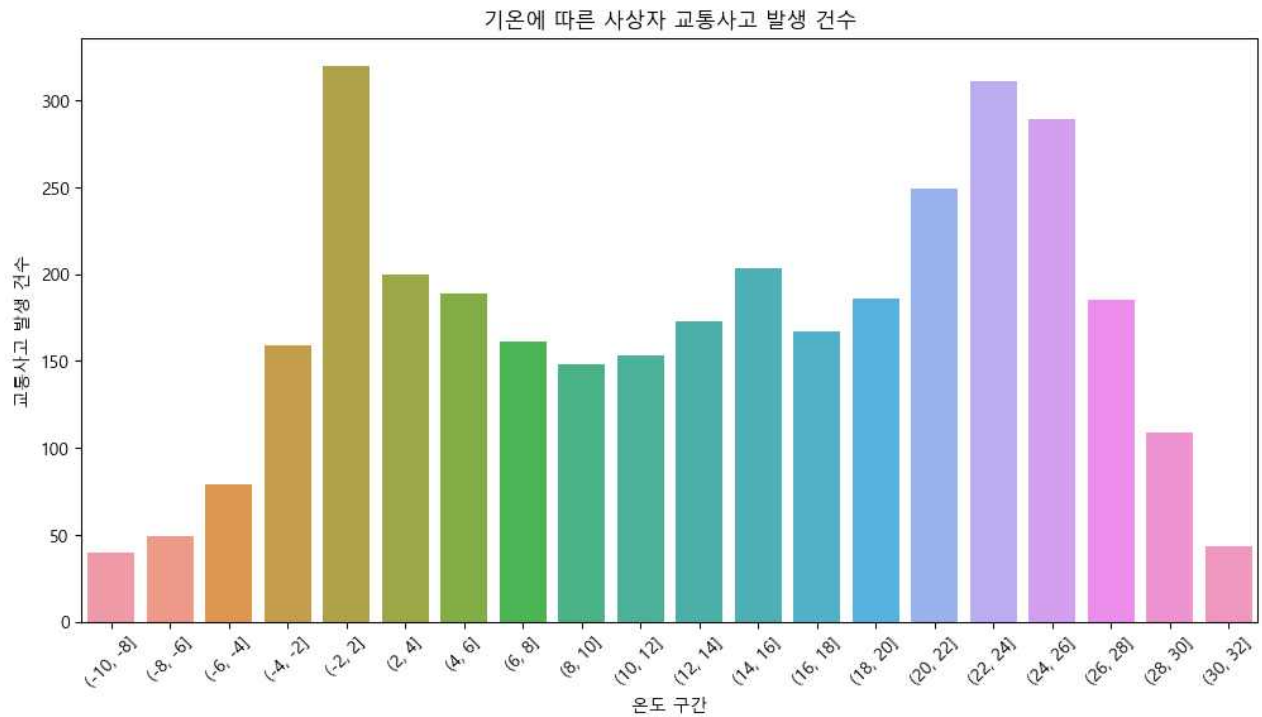
## 12-7 온도에 따른 사상자 발생 사고건수 그룹화

```
# 온도 구간 설정
온도구간 = [-10, -8, -6, -4, -2, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32]
# 온도 구간에 따른 교통사고 발생 건수
df_selected['온도구간'] = pd.cut(df_selected['평균기온'], bins=온도구간)
df_grouped = df_selected.groupby('온도구간')['사상자수'].count().reset_index() # count() 로 사상자수가 아닌 사상자 발생 사고건수
df_grouped
```

온도구간	사상자수
0 (-10, -8]	40
1 (-8, -6]	49
2 (-6, -4]	79
3 (-4, -2]	159
4 (-2, 2]	320
5 (2, 4]	200

## 12-8 기온에 따른 사상자 교통사고 발생 건수 시각화

```
# 시각화
plt.figure(figsize=(12, 6))
sns.barplot(x='온도구간', y='사상자수', data=df_grouped)
plt.title('기온에 따른 사상자 교통사고 발생 건수')
plt.xlabel('온도 구간')
plt.ylabel('교통사고 발생 건수')
plt.xticks(rotation=45)
plt.show()
```



낮은 온도와 높은 온도에서 교통사고가 각각 많이 발생함을 확인할 수 있다. 도로교통공단에 따르면 계절별 발생한 교통사고 현황을 분석한 결과, 사고의 심각성을 말해주는 치사율(사고 100건당 사망자 수)의 경우에는 블랙아이스로 인해 겨울철이 다른 계절에 비해 높은 것으로 나타났다.

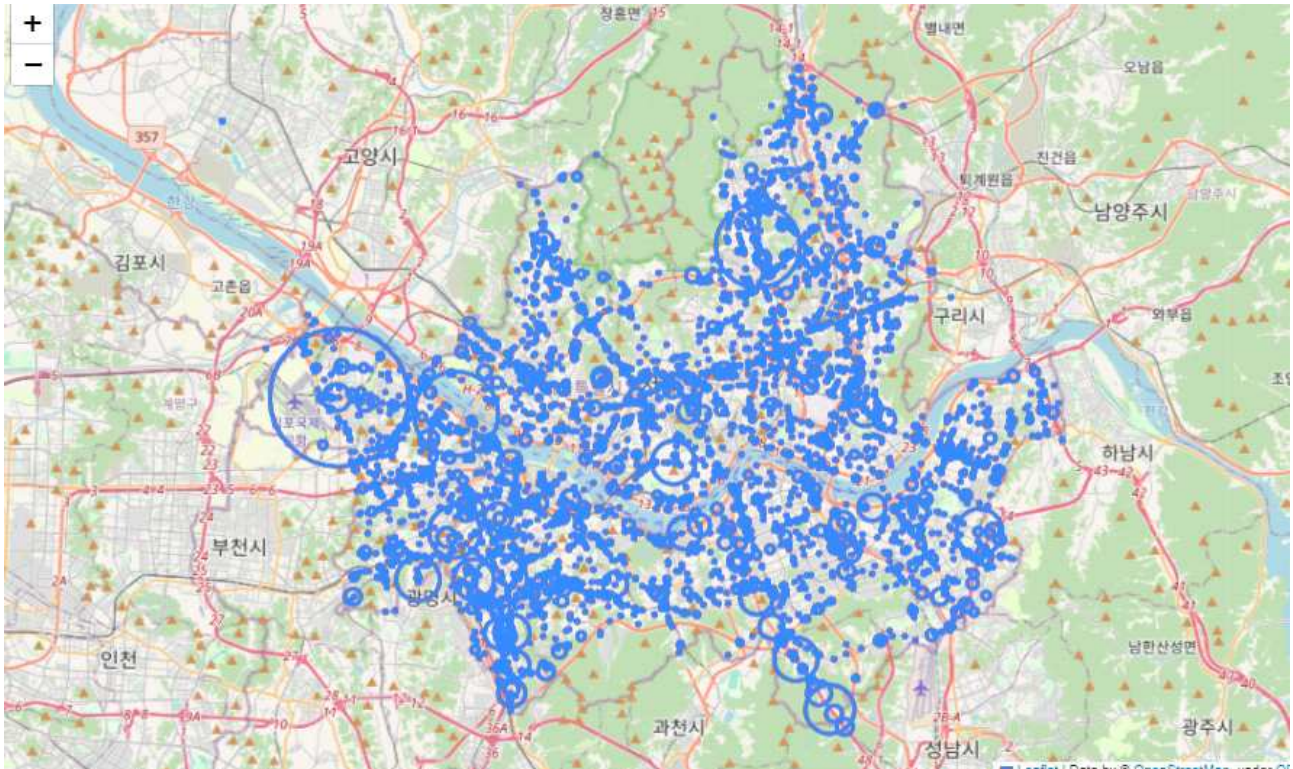
### 13. 지도 라이브러리를 활용한 시각화

folium 라이브러리를 활용하여 서울을 중심으로 지도 생성

```
!pip install folium
import folium
maps = folium.Map(location = (37.5655866, 126.977942), zoom_start = 11)
maps
```

#### 13-1 사상자수가 많이 발생한 구역을 지도에 표시

```
# 원 형태의 마커를 생성하고, 해당 마커의 반지름은 '사상자수' 값으로 설정
for lat, lng, volume in df_combined[['위도', '경도', '사상자수']].values:
    folium.CircleMarker([lat, lng], radius = volume).add_to(maps)
maps
```



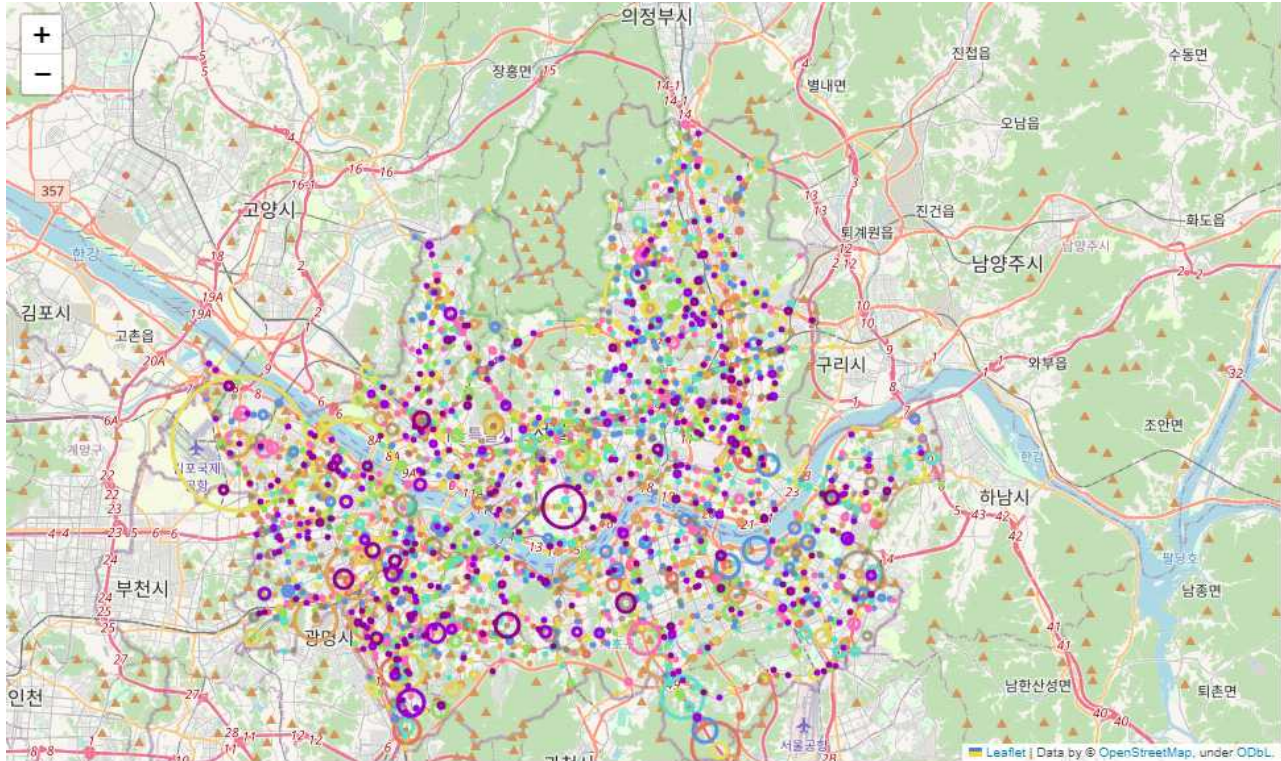


### 13-2 발생년도별 색상 다르게 표시

```
# 발생년도별 색상 다르게
maps_years = folium.Map(location = (37.5655866, 126.977942), zoom_start = 11)
color = ['#db745a', '#db8c5a', '#dbca5a', '#dfdb5a', '#8cdbc5a', '#5adbca3', '#5a8edb', '#875a', '#ff69b4', '#9400d3', '#8b008b']

for cnt, year in enumerate(df_combined['발생년'].unique()):
    marker = df_combined[df_combined['발생년'] == year]

    for lat, lng, volume in marker[['위도', '경도', '사상자수']].values:
        folium.CircleMarker([lat, lng], radius = volume, color = color[cnt]).add_to(maps_years)
maps_years
```



주로 대로와 고속도로에서 사상자가 많이 발생하는 것을 확인 할 수 있으며, 특히 경부고속도로에서 사상자가 많이 발생했다.

## 5. 결론

본 보고서는 2012년부터 2022년까지의 ‘사망자교통사고정보’ 데이터를 이용하여 교통사고의 특성과 원인을 다각도로 분석하고, 교통사고 예방과 감소를 위한 방안을 고찰 해보자는 목적으로 작성했다. 교통사고는 우리 사회의 중요한 문제이며 많은 사회적 비용과 인적 피해를 초래하고 있다. 따라서 교통사고를 줄이기 위한 노력은 우리나라 국민 모두의 협력이 필요하다.

결과적으로 교통사고의 수와 심각도는 전반적으로 감소하는 추세를 보이고 있으며, 이는 정부의 교통안전 정책과 법규 개정 등이 교통사고 감축에 기여한 것으로 판단된다. 특히 ‘안전속도 5030’이 시행된 2019년 이후에는 교통사고의 감소 폭이 더욱 커진 것을 확인할 수 있었다.

과속, 신호위반, 안전거리 미확보, 안전운전 의무 불이행, 보행자 보호의무 위반, 중앙선 침범 등의 법규위반은 교통사고의 심각도를 높이는 주요 요인이며, 차대사람, 차대차, 차량단독 등의 사고유형은 도로형태에 따라 심각도가 달라진다. 또한 날씨와 계절, 시간대와 지역 등의 환경적 요인도 교통사고의 발생과 심각도에 영향을 미친다.

‘사망자교통사고정보’ 데이터를 분석해보면서 교통사고의 심각성과 그 원인에 대해 이해할 수 있었다. 앞으로 우리는 이러한 분석을 바탕으로 교통안전 대책을 세우고 실행함으로써 교통사고를 더욱 줄일 수 있을 것이라고 생각한다. 특히, 과속, 교차로 통행방법 위반, 안전운전 의무 불이행 등의 법규위반에 대한 교육과 단속을 강화하고, 교통량이 많은 시간대와 계절, 그리고 특정 도로에서의 사고 예방에 초점을 맞추는 것이 중요하다고 생각된다.

하지만 분석을 통해 교통사고의 특성과 원인을 다각도로 조명하고자 하였으나, 분석에 사용한 데이터에는 한계가 있었다. 물피사고를 포함하지 않았으며, 명목적인 시계열 자료만을 활용하였다. 또한 서울지역의 데이터만을 활용하여 날씨에 따른 교통사고 분석을 수행했기 때문에 보다 정확하고 신뢰할 수 있는 분석을 위해서는 더 많은 데이터와 다양한 분석 방법이 필요하다고 느꼈다.



## 6. 참고문헌

- 한국도로공사 도로연구원, 고속도로 안전진단 기법 고도화 및 가이드라인 개발
- 연합뉴스, "가을 행락철 교통사고 더 주의해야"...하루평균 9.3명 사망  
<https://www.yna.co.kr/view/AKR20231017040100062>
- 도철웅 외. 『교통안전공학』. 청문각, 2019.