파이썬 개별 프로젝트

사고 자료 시각화하기

2018. 01. 30

김지원, 권정아

Contents

분석 개요

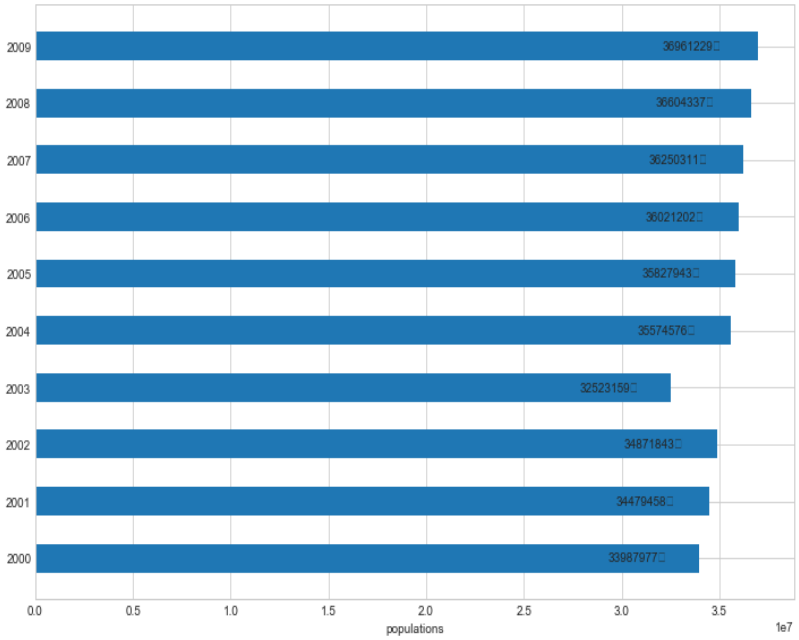
분석 개요

과제1 : 통계분석

- 미국의 연도별인구를 나타냄

과제2 : 워드클라우드

- 단톡방 데이터 분석



많은 양의 데이터를 한눈에 알아볼 수 있게 시각화

분석 개요

교통분야에 활용

- 교통사고 데이터분석
  - **도별 사고유형 빈도수**
  - **도별 사고원인**

분석방법

- Python을 통한 그래프 & 워드클라우드로 데이터를 시각화
  - **그래프 : 사고 유형별 빈도수를 나타내기 적합**
  - **워드클라우드 : 사고원인을 한눈에 보여줌**

분석 개요

분석 과정

- 데이터 수집 ➡ 데이터 정제 ➡ 데이터 분석 ➡ 분석 결과 검토

- 데이터 수집

  2014년도 경찰청 사고 정보 데이터 (**사고 년월, 법정동 코드, HIT 법정동 코드, 사고상황, 사고 유형 코드**)

  법정동 코드 데이터 (**법정동 코드, 법정동명폐지여부**)

  코드 데이터

- 데이터 정제

  사고 유형 코드 데이터 + 법정동 코드 데이터

  사고 상황 데이터 + 법정동 코드 데이터

  ➡ 〈법정동별 groupby〉
  사고 유형 코드 빈도수
  사고 상황

분석 개요

분석 과정

- 데이터 분석

1) 도별 사고 유형 표출

Matplotlib, Seaborn 사용

➡ 전체 지역 별 사고 유형 빈도수 / 도별 사고유형 누적 빈도수 / 도별 상위 6개 파이 그래프

2) 도별 사고 상황 분석

형태소 분석기 Twitter, WordCloud 사용

➡ 대한민국, 도별 사고 상황 워드 클라우드

구현 결과

1) 도별 사고 유형 표출
2) 도별 사고 상황 분석

도별 사고 유형 표출

- 각 사고 유형별 빈도수 그래프

```
# 사고 정보 데이터 불러오기
ac_data = pd.read_excel("2014 경찰청 사고 정보 데이터.xlsx")
ac_data.columns=['사고년월','법정동코드','HIT법정동코드','사고상황','code']
ac_data = ac_data.drop(0)
```

```
# 법정동 코드 정보 불러오기
code1 = pd.read_excel("법정동코드.xls")
code1.columns=['법정동코드','법정동','a','2','3','4','5']
del code1["a"]
code2=code1.dropna(how="any", axis=1)
```

```
# 사고 정보 데이터와 법정동 코드를 '법정동코드'로 합치기
result = pd.merge(ac_data, code1, on=['법정동코드'])
```

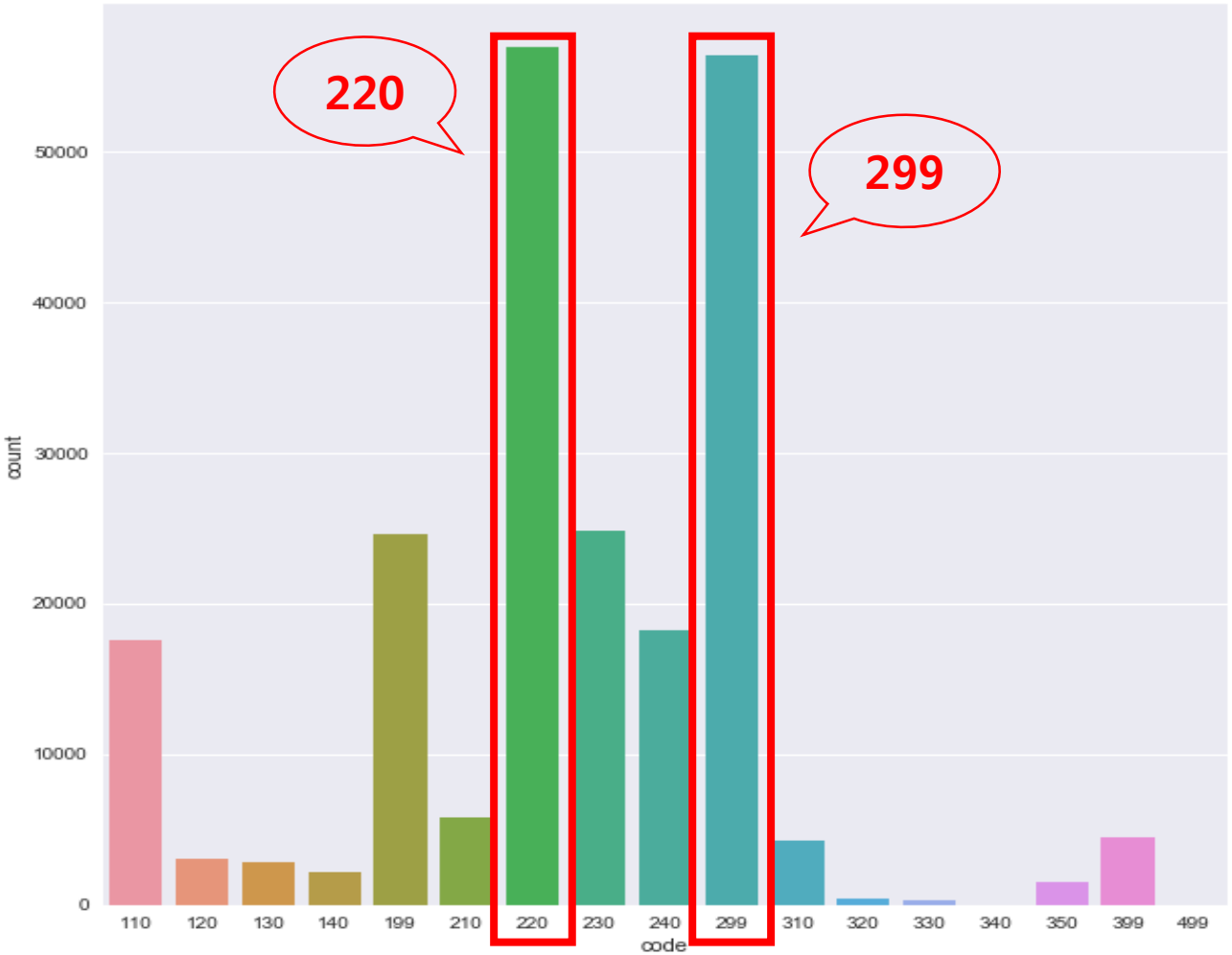
```
# font와 해상도 설정
font_location ="C:/Windows/Fonts/HMKMRHD.TTF"
font_name = font_manager.FontProperties(fname=font_location).get_name()
rc('font',family=font_name)
fig=plt.figure(dpi= 500)
```

```
# 격자 그리기
sns.set(style="darkgrid")
```

```
# 전체 지역에서 사고 유형별 빈도수 그래프 그리기
graph = sns.countplot(x="code", data = result)
plt.show()
```

도별 사고 유형 표출

- 220 – 측면직각충돌 & 299 – 차대차(기타)가 가장 많음



코드	사고 유형
110	횡단중
120	차도통행중
130	길가장자리구역통행중
140	보도통행중
199	차대사람 – 기타
210	정면충돌
220	측면직각충돌
230	추돌(진행중)
240	추돌(주정차중)
299	차대차-기타
310	공작물추돌
320	도로외이탈(추락)
330	도로외이탈(기타)
340	주차차량추돌
350	전도전복
399	차량단독-기타
410	차단기돌파
420	경보기무시
430	직진진행
499	철길건널목-기타

도별 사고 유형 표출

- 각 도별 사고 빈도수

```
# 각 도별로 사고유형코드 개수 세기
result2 = result1["code"].value_counts()
```

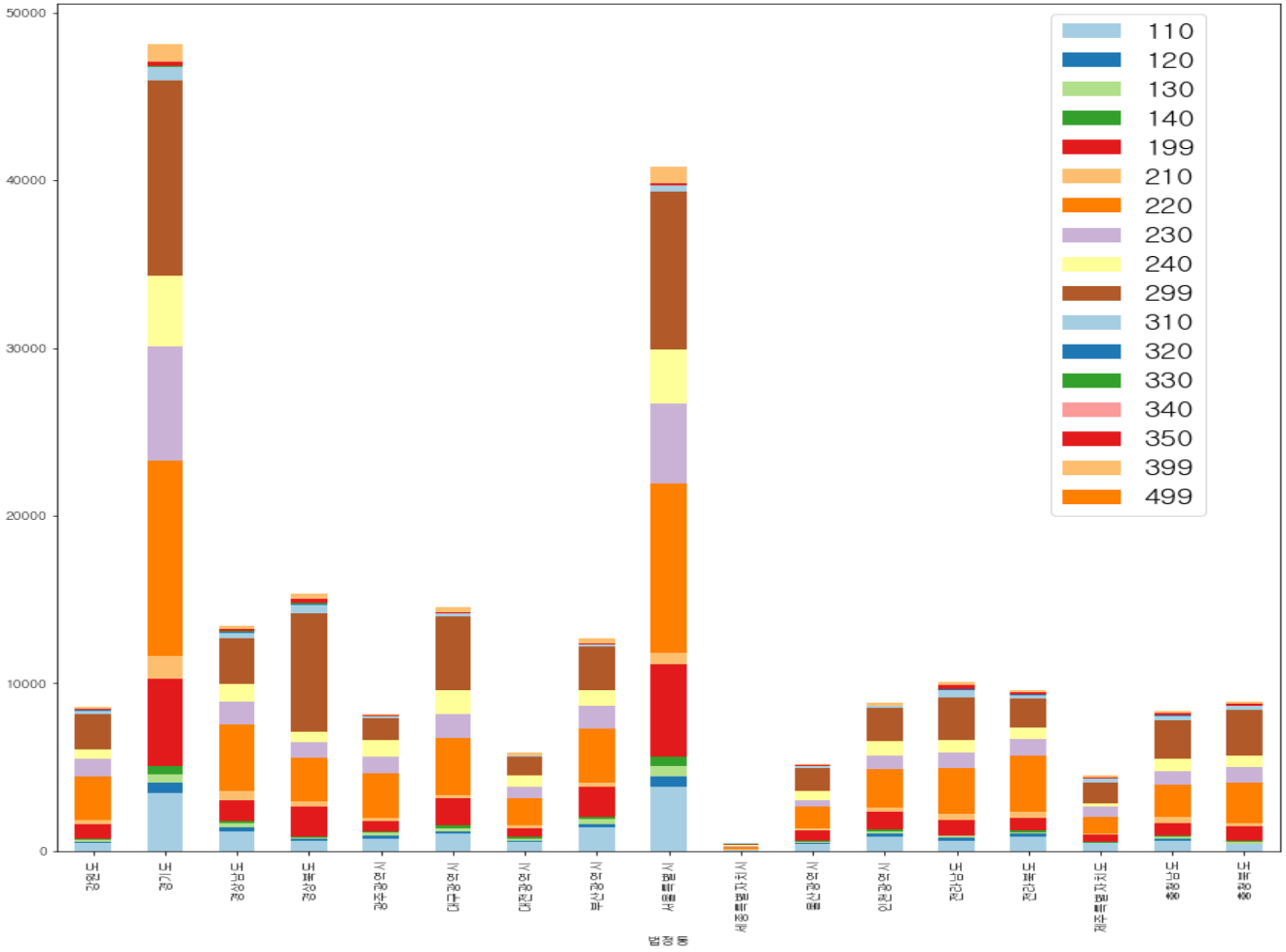
```
# result2 unstack
result4 = result2.unstack(level=0)
result4.head()
```

법정동	강원도	경기도	경상남도	경상북도	광주광역시	대구광역시	대전광역시	부산광역시	서울특별시	세종특별자치시	울산광역시	인천광역시	전라남도	전라북도	제주특별자치도	충청남도	충
code																	
110	475.0	3471.0	1169.0	613.0	750.0	1010.0	539.0	1406.0	3850.0	33.0	439.0	860.0	632.0	843.0	474.0	584.0	39
120	86.0	608.0	266.0	104.0	176.0	153.0	96.0	218.0	606.0	3.0	57.0	171.0	153.0	169.0	19.0	126.0	4
130	99.0	519.0	209.0	93.0	160.0	159.0	116.0	303.0	605.0	1.0	68.0	117.0	106.0	94.0	11.0	117.0	9
140	71.0	445.0	122.0	49.0	91.0	186.0	106.0	133.0	549.0	3.0	33.0	111.0	47.0	128.0	13.0	60.0	4
199	837.0	5217.0	1262.0	1772.0	586.0	1624.0	524.0	1740.0	5508.0	39.0	602.0	1094.0	925.0	747.0	489.0	775.0	91

```
# 누적 그래프 그리기
kk = result4.T.plot(kind='bar', stacked=True,
                    colormap=ListedColormap(sns.color_palette("Paired",20)),
                    figsize=(15,15))
kk.legend(loc=2, prop={'size': 20},bbox_to_anchor=(0.8, 1.0))

plt.show()
```

도별 사고 유형 표출

- 각 도별 사고 빈도수
- 경기도, 서울특별시가 사고가 가장 많음

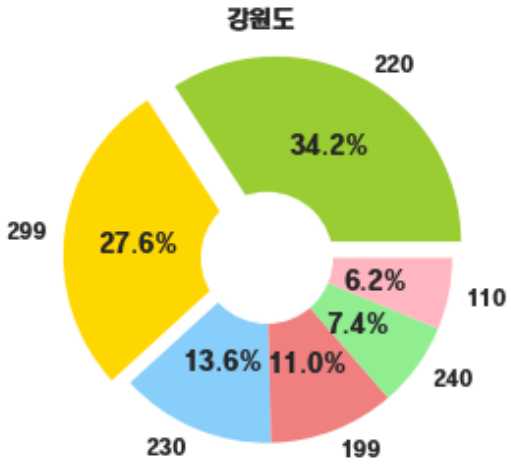


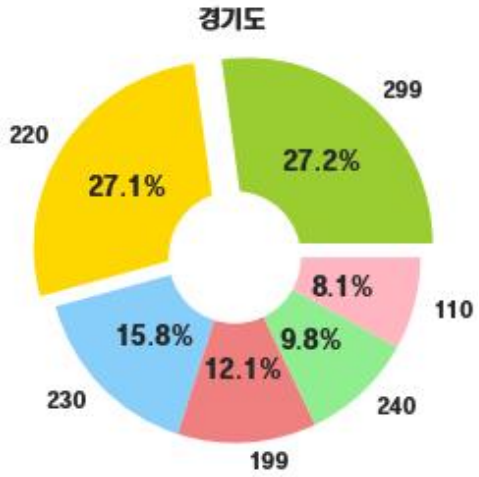
도별 사고 유형 표출

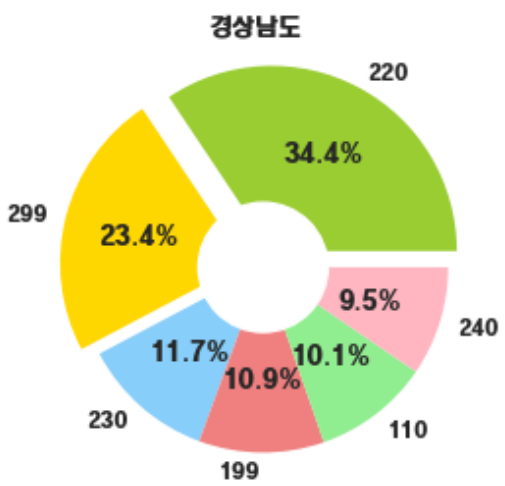
- 각 도별 상위 6개 사고유형 파이 그래프

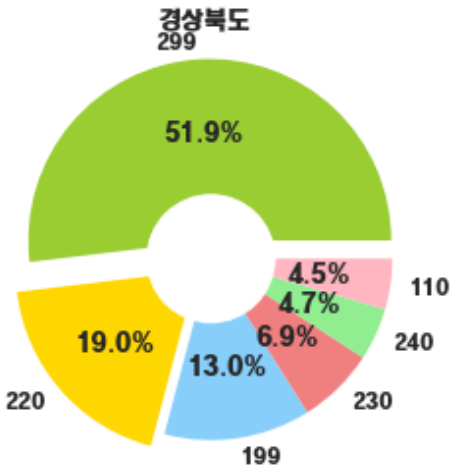
```
# 도별 상위 6개로 파이 그래프 그리기
for i in m_list:
    a=result2['%s' %i]
    b = pd.DataFrame(a)
    b.columns = ['빈도수']
    c = b[:6]
    c.reset_index()

    labels = c.index
    sizes = c.빈도수
    colors = ['yellowgreen','gold','lightskyblue', 'lightcoral','lightgreen','lightpink']

    sizes = c.빈도수

    explode = (0.1,0.1,0,0,0,0)


    plt.pie(sizes, labels = labels, autopct = '%1.1f%%', explode=explode, colors = colors)

    centre_circle = plt.Circle((0,0),0.35,color='white',fc='white',linewidth=1.25 )
    fig = plt.gcf()
    fig.gca().add_artist(centre_circle)

    plt.title('%s' %i)
    plt.axis('equal')
    plt.show()
```

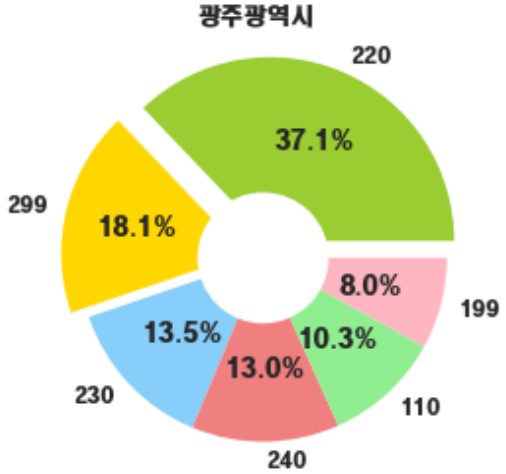
도별 사고 유형 표출

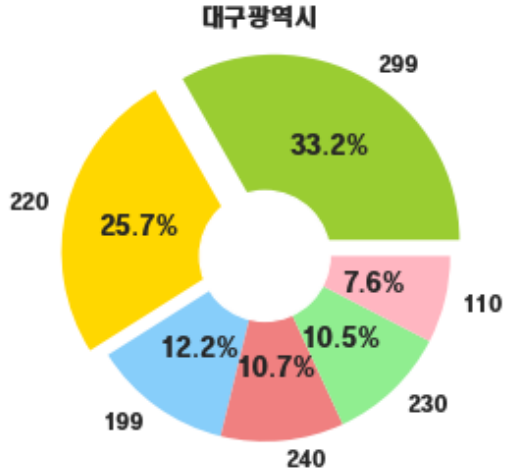

강원도
220
34.2%
299
27.6%
6.2%
110
7.4%
240
13.6%
11.0%
230
199


경기도
299
220
27.2%
27.1%
8.1%
110
15.8%
9.8%
12.1%
230
240
199

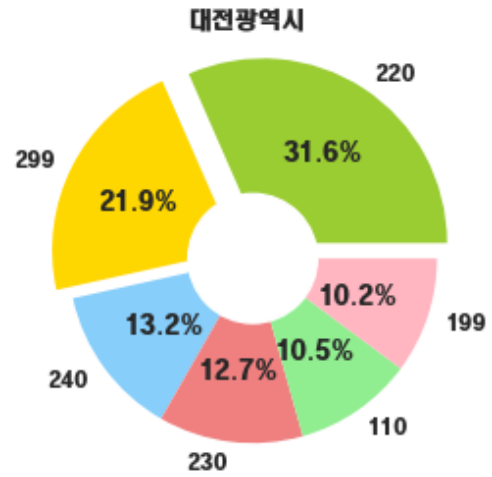

경상남도
220
34.4%
299
23.4%
9.5%
240
11.7%
10.1%
10.9%
230
110
199

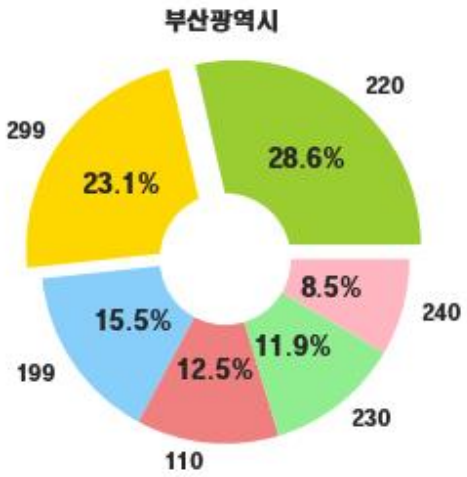

경상북도
299
51.9%
4.5%
110
4.7%
19.0%
240
6.9%
13.0%
220
230
199

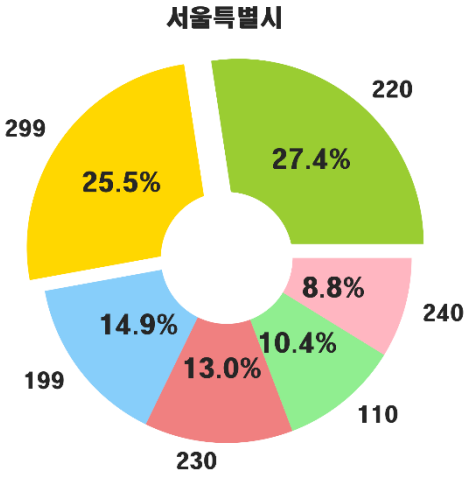

광주광역시
220
37.1%
299
18.1%
8.0%
199
13.5%
10.3%
13.0%
230
110
240

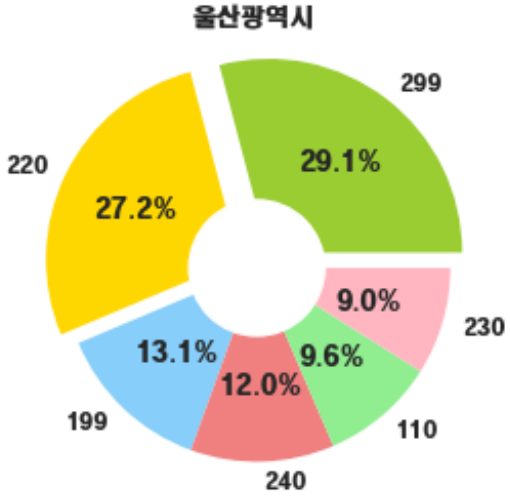

대구광역시
299
33.2%
220
25.7%
7.6%
110
12.2%
10.5%
10.7%
199
230
240

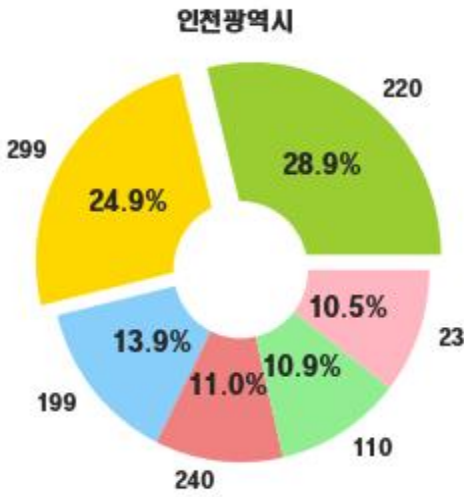
도별 사고 유형 표출

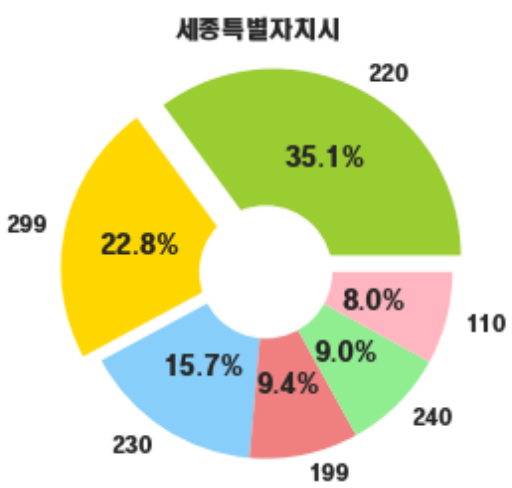

대전광역시
220
299
31.6%
21.9%
10.2%
199
13.2%
10.5%
240
12.7%
110
230


부산광역시
220
299
28.6%
23.1%
8.5%
240
15.5%
11.9%
199
12.5%
230
110

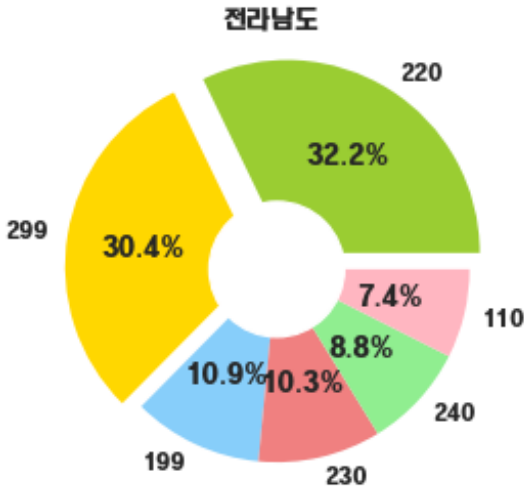

서울특별시
220
299
27.4%
25.5%
8.8%
240
14.9%
10.4%
199
13.0%
110
230

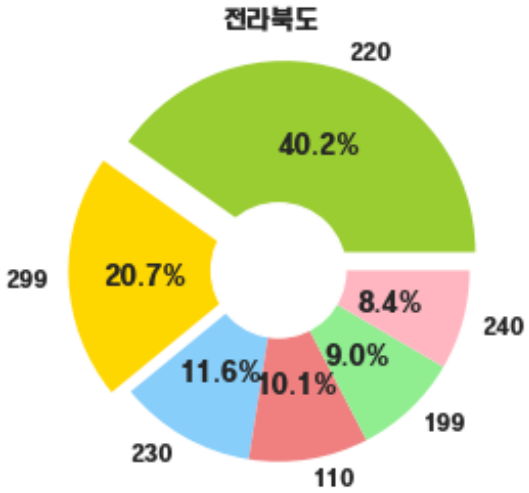

울산광역시
299
220
29.1%
27.2%
9.0%
230
13.1%
9.6%
199
12.0%
110
240

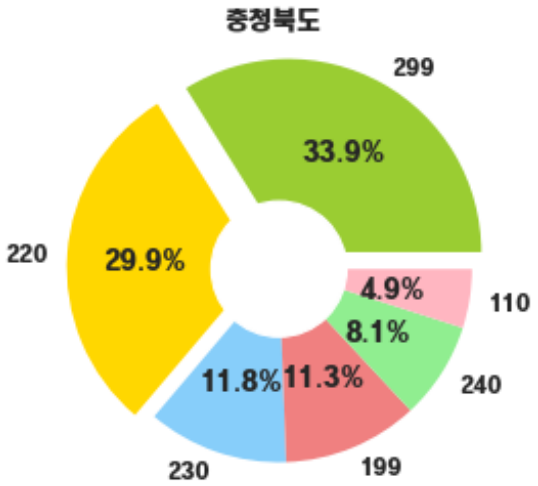

인천광역시
220
299
28.9%
24.9%
10.5%
23
13.9%
10.9%
199
11.0%
110
240

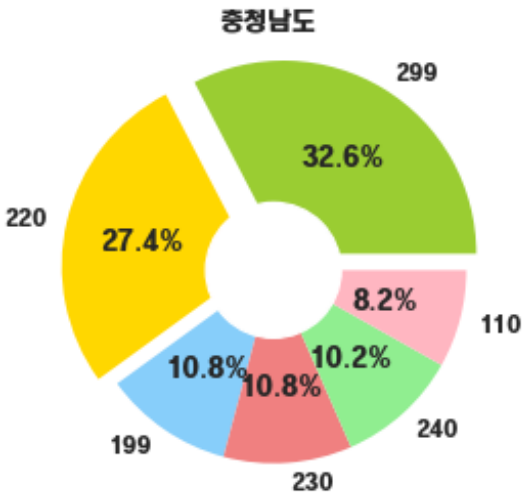

세종특별자치시
220
35.1%
299
22.8%
8.0%
110
15.7%
9.0%
9.4%
240
230
199

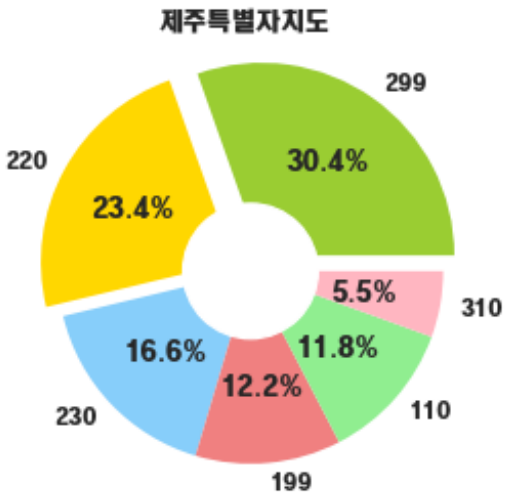
도별 사고 유형 표출


전라남도
220
32.2%
299
30.4%
7.4%
110
8.8%
10.9%
10.3%
240
199
230


전라북도
220
40.2%
299
20.7%
8.4%
240
11.6%
10.1%
9.0%
199
230
110


충청북도
299
33.9%
220
29.9%
4.9%
110
8.1%
11.8%
11.3%
240
230
199


충청남도
299
32.6%
220
27.4%
8.2%
110
10.8%
10.2%
10.8%
240
199
230


제주특별자치도
299
30.4%
220
23.4%
5.5%
310
16.6%
11.8%
12.2%
110
230
199

도별 사고 상황 분석

```
# color function 설정
def grey_color(word, font_size, position, orientation, random_state=None, **kwargs):
    return 'hsl(0, 0%%, %d%%)' % random.randint(50, 100)
```

```
# 폰트 설정
font_location = 'C:/Windows/Fonts/HMKMRHD.TTF'
font_name = font_manager.FontProperties(fname=font_location).get_name()
rc('font',family=font_name)
```

```
#for 문 설정 - 사고상황 데이터만 추출 후 워드클라우드 그리기
for i in m_list:

    a = result1[result1['법정동'] == '%s' % i]
    b = a['사고상황']
    c = ",".join(str(v) for v in b)
    d = nlp.nouns(c)
    e = ",".join(d)
    mask = np.array(Image.open("C:/python/프로젝트/마스크/%s.jpg" %i))
    wc = WordCloud(background_color="black", max_words=2000, mask=mask, stopwords = stopwords,font_path = font_location)
    wc2 = wc.generate(e)

    fig = plt.figure()
    title = ('%s' %i)
    fig.set_figwidth(14)
    fig.set_figheight(18)

    plt.imshow(wc.recolor(color_func=grey_color, random_state=3))
    plt.title(title, size=30)
    plt.axis('off')
    plt.show()
    plt.savefig("%s.png" %i)
```

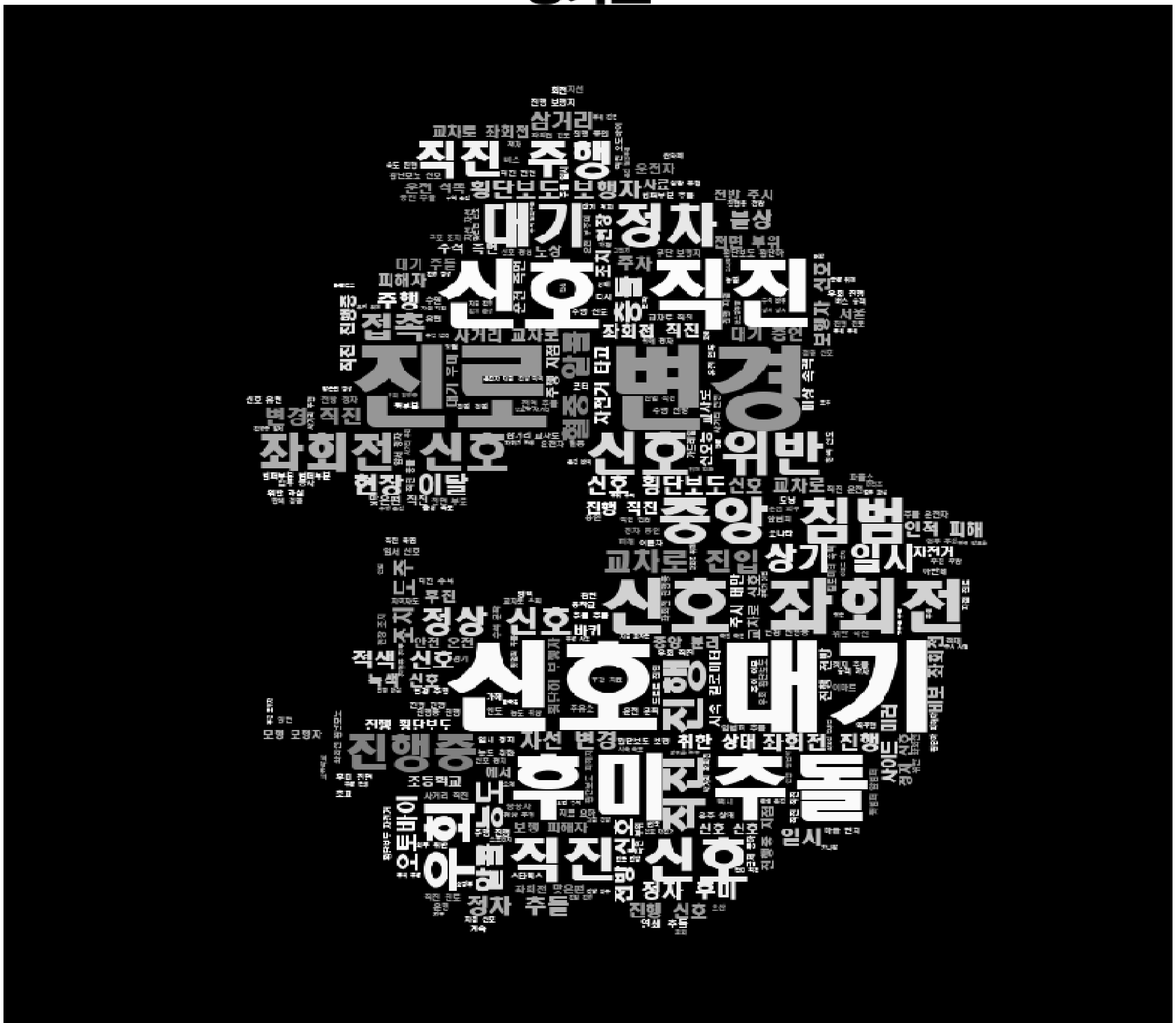
도별 사고 상황 분석

도별 사고 상황 분석

강원도

도별 사고 상황 분석

경기도



도별 사고 상황 분석

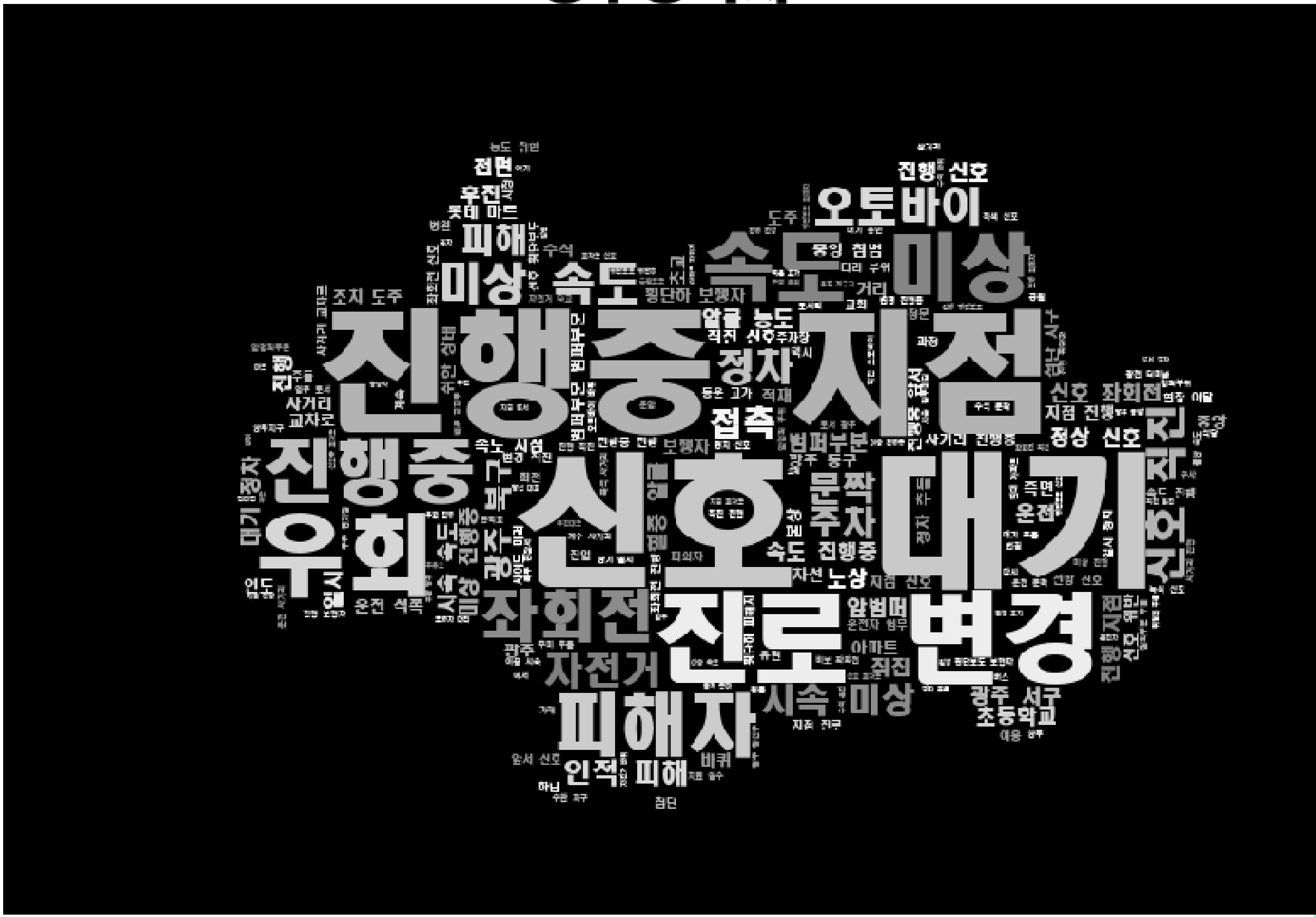
경상남도

도별 사고 상황 분석

경상북도

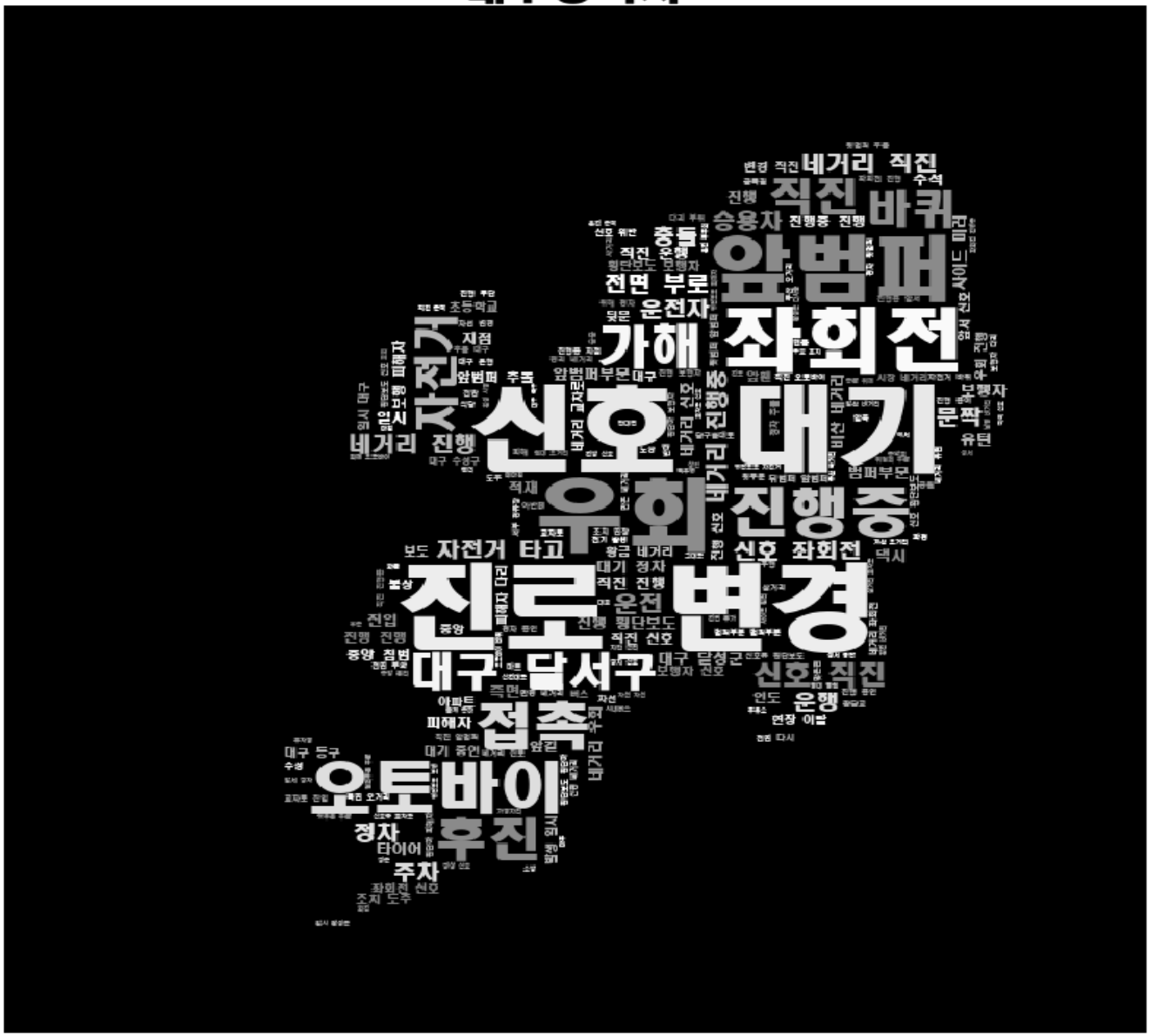
도별 사고 상황 분석

광주광역시



도별 사고 상황 분석

대구광역시



도별 사고 상황 분석

대전광역시

도별 사고 상황 분석

부산광역시

도별 사고 상황 분석

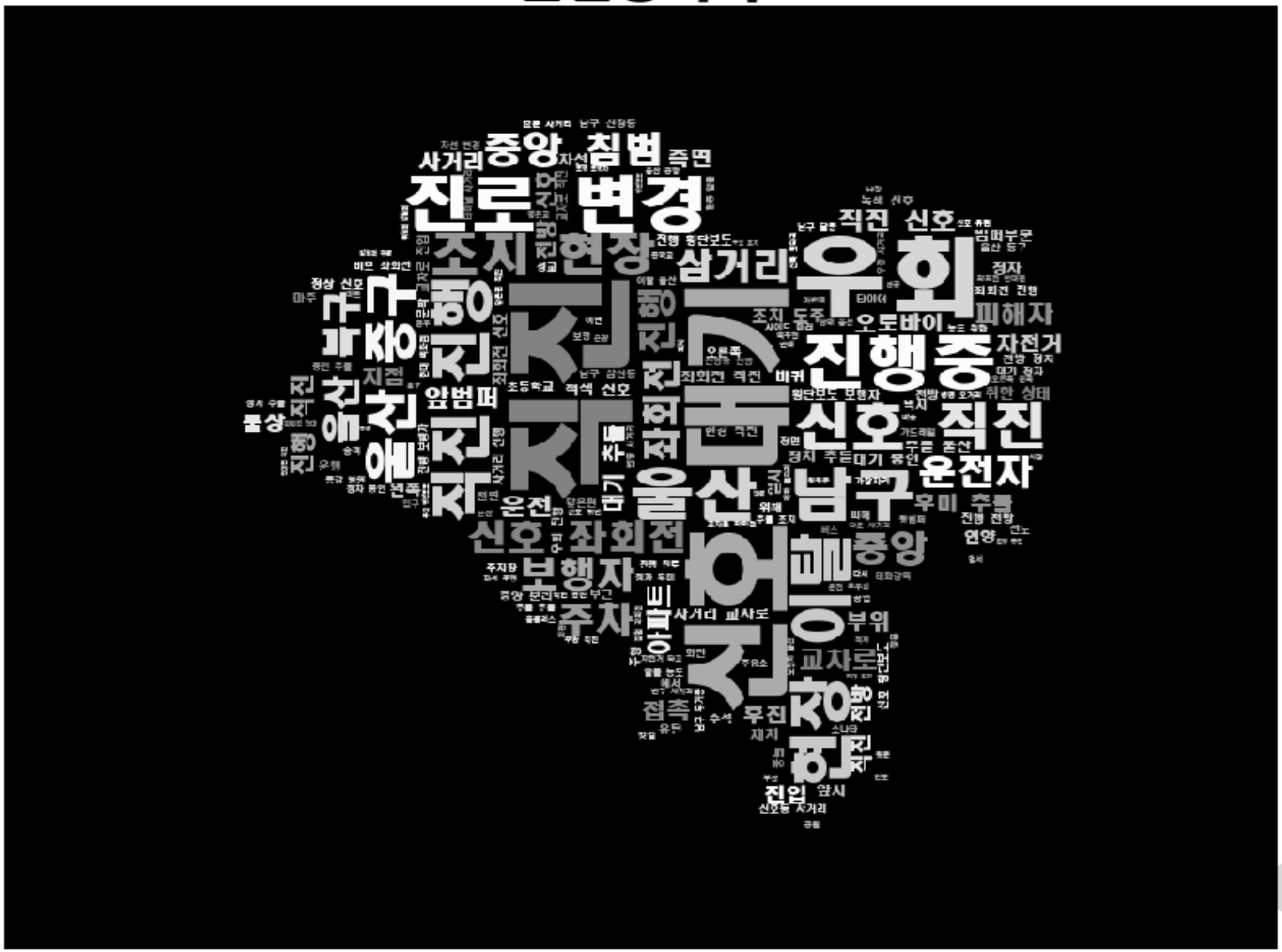
서울특별시

도별 사고 상황 분석

세종특별자치시

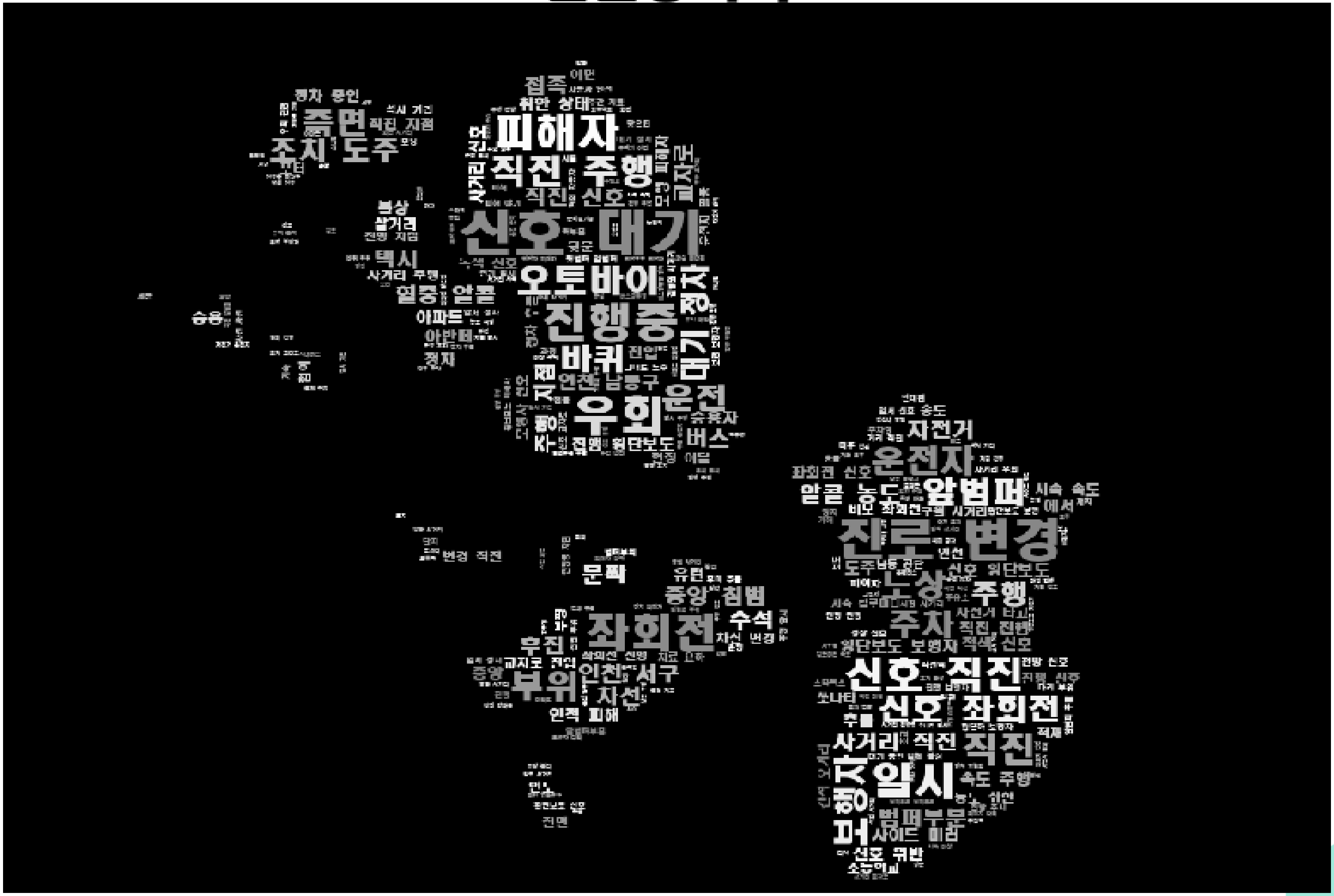
도별 사고 상황 분석

울산광역시



도별 사고 상황 분석

인천광역시



도별 사고 상황 분석

전라남도

도별 사고 상황 분석

전라북도

도별 사고 상황 분석

제주특별자치도

도별 사고 상황 분석

충청남도

도별 사고 상황 분석

충청북도

Ⅲ 결론

결론 – 도별 사고 유형 표출

결론

- 측면직각충돌과 차 대 차 사고가 가장 많음
- 경기도, 서울특별시 사고 많음
- 각 도별로 상위 6개 일정 (진행중 추돌 – 차대 사람 – 횡단 중 사고 – 주정차중사고)

프로젝트 수행 한계

- 도별 사고 빈도수 그래프에 면적 등의 요소 고려하지 않음
- 파이차트에서 상위 6개 값 이외 항목 무시
- 데이터 정제

결론 – 도별 사고 상황 분석

결론

- 신호대기, 진로 변경, 직진 중 등이 공통적으로 나타남

프로젝트 수행 한계

- STOPWORDS 설정 한계

Thank you