

https://rldnjs3258.github.io/Traffic_accidents_in_Korea/index.html

도로교통공단 교통사고 데이터

데이터 시각화 HW

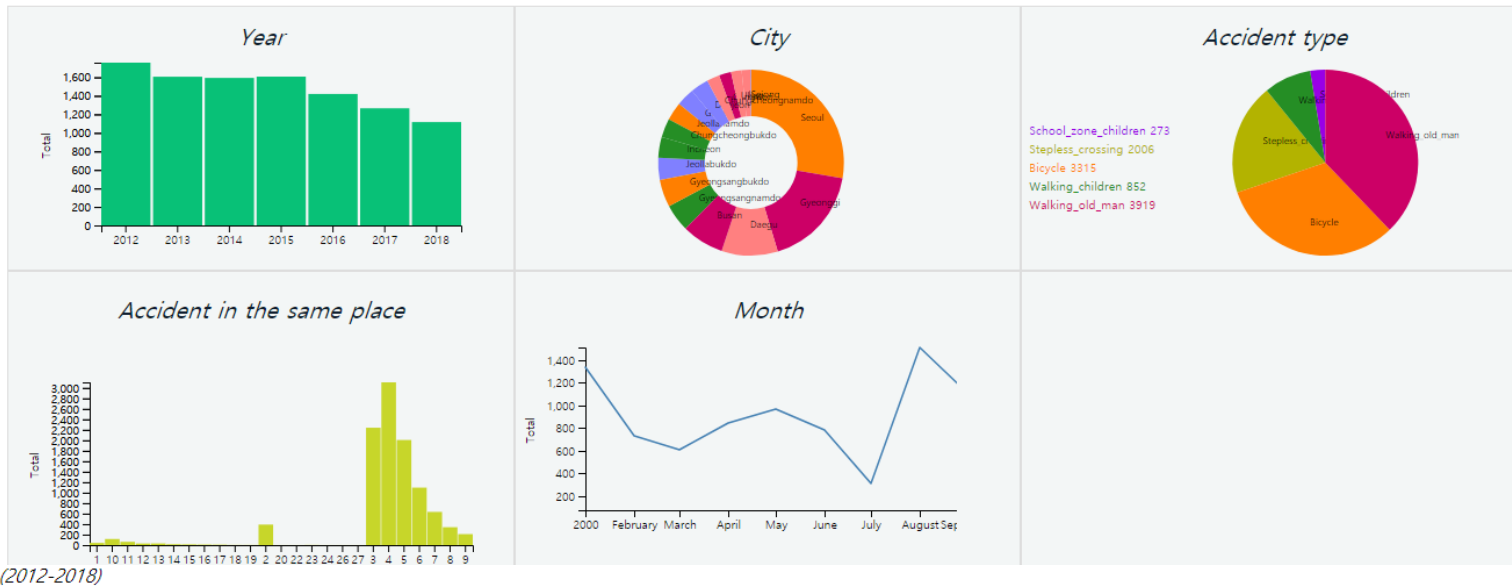




웹 페이지

Traffic Accident Data in Korea

Data Visualization (D3.js) All ▾



Hypothesis :

1. As the number of cars registered annually increases, traffic accidents will gradually increase. (False)
2. The more populated the city, the more traffic accidents will occur. (Half)
3. Traffic accidents will occur most often while walking. (True)

Result :

1. Traffic accidents gradually decreased by year. Efforts are needed to maintain current trends.
2. In traffic accidents, Seoul ranked first, Gyeonggi second, and Daegu third. The population rankings were Gyeonggi first, Seoul second, and Busan third. The two showed similar trends but not in direct proportion.

웹 페이지 웹 페이지

웹 페이지 웹 페이지 설명

■ URL

[https://rldnjs3258.github.io/Traffic accidents in Korea/index.html](https://rldnjs3258.github.io/Traffic%20accidents%20in%20Korea/index.html)

■ 제목

Traffic Accident Data in Korea

■ 시각화

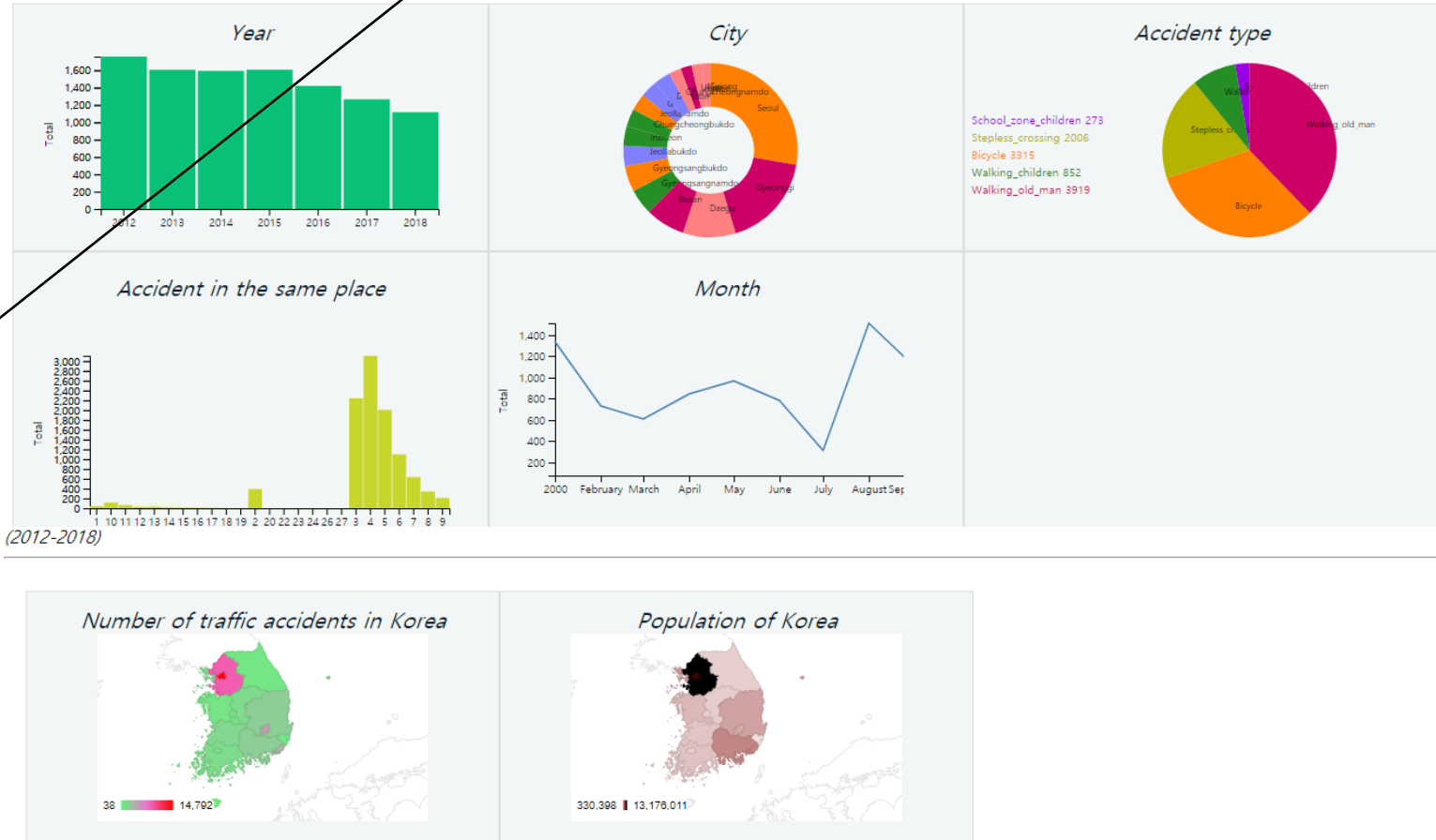
- 기본 시각화 5개가 링크됨
- 맵 시각화 2개

■ 가설 및 결론

Hypothesis & Result

Traffic Accident Data in Korea

Data Visualization (D3.js) All ▾



Hypothesis :

1. As the number of cars registered annually increases, traffic accidents will gradually increase. (False)
2. The more populated the city, the more traffic accidents will occur. (Half)
3. Traffic accidents will occur most often while walking. (True)

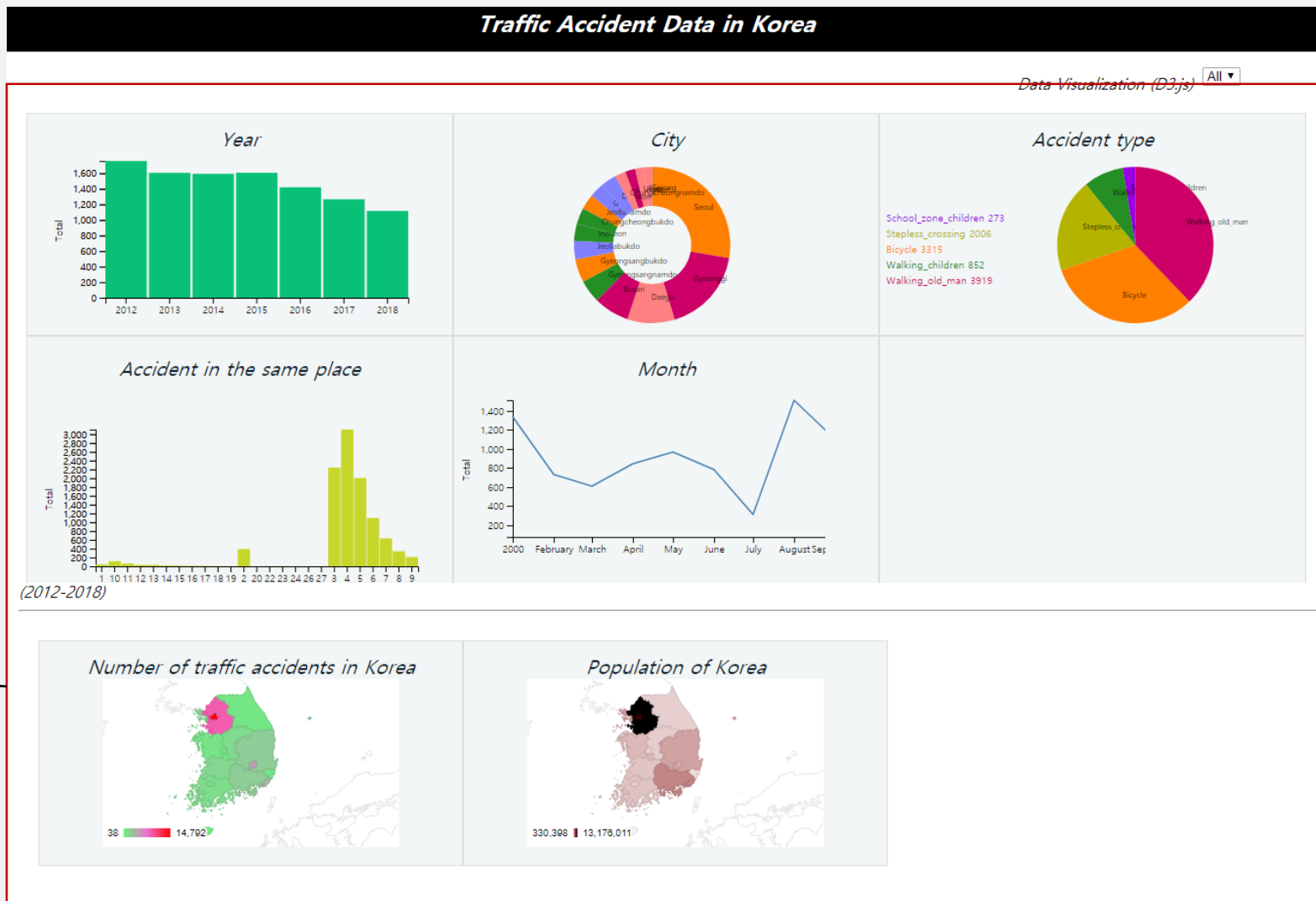
Result :

1. Traffic accidents gradually decreased by year. Efforts are needed to maintain current trends.
2. In traffic accidents, Seoul ranked first, Gyeonggi second, and Daegu third. The population rankings were Gyeonggi first, Seoul second, and Busan third. The two showed similar trends but not in direct proportion.

웹 페이지 시각화 설명

■ 시각화

- Year : 연도별 교통사고 발생건수
- City : 행정 구역별 교통사고 발생건수
- Accident type : 교통사고 타입
- Accident in the same place : 같은 장소에서
교통사고가 발생한 빈도수
- Month : 월별 교통사고 발생건수
- Number of traffic accidents in Korea : 행정
구역별 교통사고 발생건수 지도화
- Population of Korea : 행정 구역별 인구수



Hypothesis :

1. As the number of cars registered annually increases, traffic accidents will gradually increase. (False)
2. The more populated the city, the more traffic accidents will occur. (Half)
3. Traffic accidents will occur most often while walking. (True)

Result :

1. Traffic accidents gradually decreased by year. Efforts are needed to maintain current trends.
2. In traffic accidents, Seoul ranked first, Gyeonggi second, and Daegu third. The population rankings were Gyeonggi first, Seoul second, and Busan third. The two showed similar trends but not in direct proportion.

2

데이터

데이터 데이터 설명

■ 데이터

도로교통공단 교통사고다발지역 데이터

■ 데이터 설명

- 등록일 : 2019-10-10

- URL : <https://www.data.go.kr/dataset/15003493/fileData.do>

- 설명 : 스쿨존어린이, 보행어린이, 보행노인, 무단횡단, 자전거 교통사고 등 주제별로 선정한 2012년 ~ 2018년의 연간 교통 사고 다발지역 데이터

데이터 데이터 보기

사고지역구	사고년도	사고유형구	위치코드	시도시군구명	사고지역	발생건수	사상자수	사망자수	중상자수	경상자수	부상자수	위도	경도	사고다발지역	데이터기준일자
2013060	2012	스쿨존어린이	1111000	서울특별시	서울특별시	2	2	0	1	1		37.58841291	126.9996071	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	1129000	서울특별시	서울특별시	2	2	0	0	1		37.59550472	127.0357332	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	1129000	서울특별시	서울특별시	2	2	0	2	0		37.59757144	127.0145636	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	1129000	서울특별시	서울특별시	3	3	0	2	1		37.61042892	127.0595956	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	1132000	서울특별시	서울특별시	2	2	0	0	2		37.64812892	127.0244298	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	1135000	서울특별시	서울특별시	2	2	1	1	0		37.66443684	127.056828	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	1153000	서울특별시	서울특별시	2	3	0	1	0		37.48993952	126.8569301	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	1153000	서울특별시	서울특별시	2	2	0	1	1		37.50392059	126.8458306	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	1171000	서울특별시	서울특별시	2	2	0	0	2		37.50979654	127.122101	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	2629000	부산광역시	부산광역시	2	2	0	1	1		35.1225485	129.1096602	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	2629000	부산광역시	부산광역시	2	2	0	1	1		35.12877156	129.1136624	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	2635000	부산광역시	부산광역시	2	2	0	2	0		35.19955555	129.1191075	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	2641000	부산광역시	부산광역시	3	3	0	1	1		35.26277537	129.0881971	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	2729000	대구광역시	대구광역시	2	2	0	0	2		35.85302603	128.5715066	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	2917000	광주광역시	광주광역시	2	2	0	1	1		35.20643137	126.8728694	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	2920000	광주광역시	광주광역시	2	2	0	1	1		35.17766246	126.8149078	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	3023000	대전광역시	대전광역시	2	2	0	1	1		36.3559508	127.4300776	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	4111700	경기도 수원	경기도 수원	2	2	0	1	1		37.26266404	127.0446475	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	4115000	경기도 의정부	경기도 의정부	2	2	0	1	1		37.72334512	127.0534968	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	4119500	경기도 부천	경기도 부천	2	2	0	1	1		37.490115	126.7601216	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	4122000	경기도 평택	경기도 평택	2	2	0	1	1		37.05039551	127.0573566	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	4127300	경기도 안산	경기도 안산	2	2	0	1	1		37.3314114	126.8180732	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	4127300	경기도 안산	경기도 안산	3	3	0	2	1		37.3453435	126.8164611	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	4136000	경기도 남양주	경기도 남양주	2	2	0	0	2		37.60065299	127.1568879	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	4217000	강원도 동해	강원도 동해	2	2	0	1	1		37.48569878	129.1051372	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	4623000	전라남도 광안	전라남도 광안	2	2	0	0	2		34.9370074	127.6914809	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	4719000	경상북도 구미	경상북도 구미	2	2	0	0	2		36.09896837	128.3601479	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	4790000	경상북도 예천	경상북도 예천	3	3	0	0	3		36.51129446	128.3005673	type:Polygc	2019-10-10
2013060	2012	스쿨존어린이	4812900	경상남도 창원	경상남도 창원	2	4	0	0	4		35.10347408	128.8057126	type:Polygc	2019-10-10

데이터 데이터 컬럼

사고지역관리번호	사고년도	사고유형구분	위치코드	시도시군구명	사고지역위치가발생건수	사상자수	사망자수	중상자수	경상자수	부상자수	위도	경도	사고다발지역데이터기준일자
2013060	2012	스쿨존어린이	11110001	서울특별시 서울특별시	2	2	0	1	1	0	37.58841291	126.9996071	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	11290001	서울특별시 서울특별시	2	2	0	0	1	1	37.59550472	127.0357332	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	11290002	서울특별시 서울특별시	2	2	0	2	0	0	37.59757144	127.0145636	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	11290003	서울특별시 서울특별시	3	3	0	2	1	0	37.61042892	127.0595956	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	11320001	서울특별시 서울특별시	2	2	0	0	2	0	37.64812892	127.0244298	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	11350001	서울특별시 서울특별시	2	2	1	1	0	0	37.66443684	127.0568228	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	11530001	서울특별시 서울특별시	2	3	0	1	0	2	37.48939952	126.8569301	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	11530002	서울특별시 서울특별시	2	2	0	1	1	0	37.50392059	126.8458306	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	11710001	서울특별시 서울특별시	2	2	0	0	2	0	37.50979654	127.122101	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	26290001	부산광역시 부산광역시	2	2	0	1	1	0	35.1225485	129.1096602	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	26290002	부산광역시 부산광역시	2	2	0	1	1	0	35.12877156	129.1136624	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	26350001	부산광역시 부산광역시	2	2	0	2	0	0	35.19955555	129.1191075	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	26410001	부산광역시 부산광역시	3	3	0	1	1	1	35.26277537	129.0881971	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	27290001	대구광역시 대구광역시	2	2	0	0	2	0	35.85302603	128.5715066	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	29170001	광주광역시 광주광역시	2	2	0	1	1	0	35.20643137	126.8728594	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	29300001	광주광역시 광주광역시	2	2	0	1	1	0	35.17765246	126.8149079	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	30230001	대전광역시 대전광역시	2	2	0	1	1	0	36.3559598	127.4300776	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	41117001	경기도 수원 경기도 수원	2	2	0	1	1	0	37.26266404	127.0446475	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	41150001	경기도 의정부 경기도 의정부	2	2	0	1	1	0	37.72334512	127.0534968	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	41195001	경기도 부천 경기도 부천	2	2	0	1	1	0	37.490115	126.7601216	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	41220001	경기도 평택 경기도 평택	2	2	0	1	1	0	37.05039951	127.0573566	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	41273001	경기도 안산 경기도 안산	2	2	0	1	1	0	37.3314114	126.8180732	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	41273002	경기도 안산 경기도 안산	3	3	0	2	1	0	37.3453435	126.8164611	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	41360001	경기도 남양 경기도 남양	2	2	0	0	2	0	37.60065299	127.1568879	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	42170001	강원도 동해 강원도 동해	2	2	0	1	1	0	37.48569878	128.1051372	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	46230001	전라남도 함안 전라남도 함안	2	2	0	0	2	0	34.9370074	127.6914809	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	47190001	경상북도 구 경상북도 구	2	2	0	0	2	0	36.09896837	128.3601479	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	47900001	경상북도 예 경상북도 예	3	3	0	0	3	0	36.51129446	128.3008673	(type:Polygon, 2019-10-10)
2013060	2012	스쿨존어린이	48129001	경상남도 함안 경상남도 함안	2	4	0	0	4	0	35.10347406	128.8057126	(type:Polygon, 2019-10-10)

- 데이터 컬럼
- 사고지역관리번호
- 사고년도
- 사고유형구분
- 위치구분
- 시도시군구명
- 사고지역위치
- 발생건수
- 사상자수
- 사망자수
- 중상자수
- 경상자수
- 부상자수
- 위도
- 경도
- 사고다발지역폴리곤정보
- 데이터기준일자

데이터 데이터 전처리 1

R

```
# <데이터 전처리>
# - 도로교통공단_교통사고다발지역_20191010.csv의 데이터를 전처리 한다.
# - 데이터 중 '사고년도, 사고유형구분, 시도시군구명, 사고지역위치명, 발생건수, 사상자수, 사망자수, 중상자수, 경상자수, 부상자수, 위도,
# - 데이터의 열 이름을 변경한다.
# - 데이터를 사고년도 기준으로 오름정렬 한다.

# 1. 데이터 로드
rawdata <- read.csv(file.choose()) #데이터 로드
summary(rawdata) # summary
head(rawdata) # head

# 2. 데이터 정제 1 : 열 선택
# - 데이터 중 사용할 열을 선택하여 정제한다.
data <- data.frame(rawdata$사고년도, rawdata$사고유형구분, rawdata$시도시군구명, rawdata$사고지역위치명, rawdata$발생건수, rawdata$사상
summary(data)
head(data)
str(data)

# 3. 데이터 정제 2 : 열 이름 변경
colnames(data) = c("사고년도", "사고유형구분", "시도시군구명", "사고지역위치명", "발생건수", "사상자수", "사망자수", "중상자수", "경상자
head(data)

# 4. 데이터 정제 3 : 오름정렬
# - 데이터 중 '사고년도' 기준으로 오름정렬 한다.
data <- data[c(order(data$사고년도)),]
head(data)

# 5. 저장
write.csv(data, "C:/Users/JIHYE/Desktop/데이터 시각화 과제/Data/data.csv")
```

데이터 데이터 전처리 2

R

```
# - 시도시군구명 가장 앞 단어를 추출하여 '시도' 열 만들기.

# 1. 데이터 로드
rawdata <- read.csv(file.choose()) #데이터 로드
summary(rawdata) # summary
head(rawdata) # head

# 2. 데이터 전처리 1 : '시도' 열 만들기
rawdata$시도시군구명 <- as.character(rawdata$시도시군구명)
시도_list <- character()
for (i in 1:10365){
  a <- strsplit(rawdata$시도시군구명, split = " ")[[i]][1]
  시도_list <- append(시도_list, a)
}

# 3. 데이터 전처리 2 : '시도' 열 정제하기
# - '시도' 열 중, 같은 값이지만 다른 라벨이 붙은 데이터들을 같은 라벨로 만든다.
unique(시도_list)
시도_list <- gsub("서울특별시", "서울", 시도_list)
시도_list <- gsub("충북", "충청북도", 시도_list)
시도_list <- gsub("충남", "충청남도", 시도_list)
시도_list <- gsub("대구광역시", "대구", 시도_list)
시도_list <- gsub("부산광역시", "부산", 시도_list)
시도_list <- gsub("제주특별자치도", "제주", 시도_list)
시도_list <- gsub("강원도", "강원", 시도_list)
시도_list <- gsub("전남", "전라남도", 시도_list)
시도_list <- gsub("전북", "전라북도", 시도_list)
시도_list <- gsub("경기도", "경기", 시도_list)
시도_list <- gsub("대전광역시", "대전", 시도_list)
시도_list <- gsub("울산광역시", "울산", 시도_list)
시도_list <- gsub("경북", "경상북도", 시도_list)
시도_list <- gsub("경남", "경상남도", 시도_list)
시도_list <- gsub("광주광역시", "광주", 시도_list)
시도_list <- gsub("인천광역시", "인천", 시도_list)

# 4. 데이터 전처리 2 : 열 선택
data <- data.frame(rawdata$사고년도, rawdata$사고유형구분, rawdata$시도시군구명, rawdata$사고지역위치명, rawdata$발생건수, rawdata$사상자수, rawdata$사망자수)
head(data)

# 5. 데이터 전처리 3 : 열 이름 정제
colnames(data) = c("사고년도", "사고유형구분", "시도시군구명", "사고지역위치명", "발생건수", "사상자수", "사망자수", "중상자수")

# 6. 저장
write.csv(data, "C:/Users/JIHYE/Desktop/데이터시각화HW - 도로교통공단 교통사고 데이터/2. 데이터/Raw data/data2.csv")
```

데이터 데이터 전처리 3

■ R

```
# - data2.csv 파일을 이용하여 전처리한다.  
# - 사고유형구분별 발생건수 구하기  
# - 사고년도별 발생건수 구하기  
# - 시도별 발생건수 구하기
```

```
# 1. 데이터 로드  
rawdata <- read.csv(file.choose()) #데이터 로드  
summary(rawdata) # summary  
head(rawdata) # head
```

```
sum_stepless_crossing  
sum_walking_oldman  
sum_walking_children  
sum_schoolzone_children  
sum_bicycle
```

```
accident_data <- data.frame(sum_stepless_crossing,  
                             sum_walking_oldman,  
                             sum_walking_children,  
                             sum_schoolzone_children,  
                             sum_bicycle) # 데이터프레임 만들기
```

```
# 3. 데이터 전처리 2 : 사고년도별 발생건수 구하기  
sum_2012 = 0  
sum_2013 = 0  
sum_2014 = 0  
sum_2015 = 0  
sum_2016 = 0  
sum_2017 = 0  
sum_2018 = 0
```

```
unique(rawdata$사고년도)  
for (i in c(1:length(rawdata$사고년도))){  
  if (rawdata$사고년도[i] == 2012){  
    sum_2012 = sum_2012 + rawdata$발생건수[i]  
  } else if (rawdata$사고년도[i] == 2013){  
    sum_2013 = sum_2013 + rawdata$발생건수[i]  
  } else if (rawdata$사고년도[i] == 2014){  
    sum_2014 = sum_2014 + rawdata$발생건수[i]  
  } else if (rawdata$사고년도[i] == 2015){  
    sum_2015 = sum_2015 + rawdata$발생건수[i]  
  } else if (rawdata$사고년도[i] == 2016){  
    sum_2016 = sum_2016 + rawdata$발생건수[i]  
  }  
}
```

데이터 데이터 전처리 3

R

```
sum_2016 = sum_2016 + rawdata$발생건수[i]
} else if (rawdata$사고년도[i] == 2017){
sum_2017 = sum_2017 + rawdata$발생건수[i]
} else if (rawdata$사고년도[i] == 2018){
sum_2018 = sum_2018 + rawdata$발생건수[i]
}
}

sum_2012
sum_2013
sum_2014
sum_2015
sum_2016
sum_2017
sum_2018

year_data <- data.frame(sum_2012,
                        sum_2013,
                        sum_2014,
                        sum_2015,
                        sum_2016,
                        sum_2017,
                        sum_2018) # 데이터프레임 만들기

# 4. 데이터 전처리 3 : 시도별 발생건수 구하기
sum_Seoul = 0
sum_Busan = 0
sum_Daegu = 0
sum_Gwangju = 0
sum_Gyeongsangbuk_do = 0
sum_Gyeongsangnam_do = 0
sum_Jeju = 0
sum_Incheon = 0
sum_Ulsan = 0
sum_Chungcheongbuk_do = 0
sum_Chungcheongnam_do = 0
sum_Jeollabuk_do = 0
sum_Sejong = 0

unique(rawdata$시도)
for (i in c(1:length(rawdata$시도))){
  if (rawdata$시도[i] == '서울'){
    sum_Seoul = sum_Seoul + rawdata$발생건수[i]
  } else if (rawdata$시도[i] == '부산'){
    sum_Busan = sum_Busan + rawdata$발생건수[i]
  } else if (rawdata$시도[i] == '대구'){
    sum_Daegu = sum_Daegu + rawdata$발생건수[i]
  } else if (rawdata$시도[i] == '광주'){
    sum_Gwangju = sum_Gwangju + rawdata$발생건수[i]
  } else if (rawdata$시도[i] == '대전'){
    sum_Daejeon = sum_Daejeon + rawdata$발생건수[i]
  } else if (rawdata$시도[i] == '경기'){
    sum_Gyeonggi = sum_Gyeonggi + rawdata$발생건수[i]
  } else if (rawdata$시도[i] == '강원'){
    sum_Gangwon = sum_Gangwon + rawdata$발생건수[i]
  } else if (rawdata$시도[i] == '전라남도'){
    sum_Jeollanam_do = sum_Jeollanam_do + rawdata$발생건수[i]
  } else if (rawdata$시도[i] == '경상북도'){
    sum_Gyeongsangbuk_do = sum_Gyeongsangbuk_do + rawdata$발생건수[i]
  } else if (rawdata$시도[i] == '경상남도'){
    sum_Gyeongsangnam_do = sum_Gyeongsangnam_do + rawdata$발생건수[i]
  } else if (rawdata$시도[i] == '제주'){
    sum_Jeju = sum_Jeju + rawdata$발생건수[i]
  } else if (rawdata$시도[i] == '인천'){
    sum_Incheon = sum_Incheon + rawdata$발생건수[i]
  } else if (rawdata$시도[i] == '울산'){
    sum_Ulsan = sum_Ulsan + rawdata$발생건수[i]
  } else if (rawdata$시도[i] == '충청북도'){
    sum_Chungcheongbuk_do = sum_Chungcheongbuk_do + rawdata$발생건수[i]
  } else if (rawdata$시도[i] == '충청남도'){
    sum_Chungcheongnam_do = sum_Chungcheongnam_do + rawdata$발생건수[i]
  } else if (rawdata$시도[i] == '전라북도'){
    sum_Jeollabuk_do = sum_Jeollabuk_do + rawdata$발생건수[i]
  } else if (rawdata$시도[i] == '세종'){
    sum_Sejong = sum_Sejong + rawdata$발생건수[i]
  }
}
```

데이터 데이터 전처리 3

R

```
} else if (rawdata$시도[i] == '인천'){
  sum_Incheon = sum_Incheon + rawdata$발생건수[i]
} else if (rawdata$시도[i] == '울산'){
  sum_Ulsan = sum_Ulsan + rawdata$발생건수[i]
} else if (rawdata$시도[i] == '충청북도'){
  sum_Chungcheongbuk_do = sum_Chungcheongbuk_do + rawdata$발생건수[i]
} else if (rawdata$시도[i] == '충청남도'){
  sum_Chungcheongnam_do = sum_Chungcheongnam_do + rawdata$발생건수[i]
} else if (rawdata$시도[i] == '전라북도'){
  sum_Jeollabuk_do = sum_Jeollabuk_do + rawdata$발생건수[i]
} else if (rawdata$시도[i] == '세종특별자치시'){
  sum_Sejong = sum_Sejong + rawdata$발생건수[i]
}
}

city_data <- data.frame(sum_Seoul,
                        sum_Busan,
                        sum_Daegu,
                        sum_Gwangju,
                        sum_Daejeon,
                        sum_Gyeonggi,
                        sum_Gangwon,
                        sum_Jeollanam_do,
                        sum_Gyeongsangbuk_do,
                        sum_Gyeongsangnam_do,
                        sum_Jeju,
                        sum_Incheon,
                        sum_Ulsan,
                        sum_Chungcheongbuk_do)

# 5. 데이터 전처리 정확한지 확인하기
# - 세 값 모두 49242로 같다. 데이터 전처리가 정확함을 알 수 있다.
sum_2012 + sum_2013 + sum_2014 + sum_2015 + sum_2016 + sum_2017 + sum_2018
sum_stepless_crossing + sum_walking_oldman + sum_walking_children + sum_schoolzone_children + sum_bicycle
sum_Seoul + sum_Busan + sum_Daegu + sum_Gwangju + sum_Daejeon + sum_Gyeonggi + sum_Gangwon + sum_Jeollanam_do + sum_Gyeongsangbuk_do + sum_Jeju + sum_Incheon + sum_Ulsan + sum_Chungcheongbuk_do

# 6. 저장
write.csv(accident_data, "C:/Users/JIHYE/Desktop/데이터시각화HW - 도로교통공단 교통사고 데이터/2. 데이터/Final data/accident.csv")
write.csv(year_data, "C:/Users/JIHYE/Desktop/데이터시각화HW - 도로교통공단 교통사고 데이터/2. 데이터/Final data/year.csv")
write.csv(city_data, "C:/Users/JIHYE/Desktop/데이터시각화HW - 도로교통공단 교통사고 데이터/2. 데이터/Final data/city.csv")
```

데이터 데이터 전처리 4

R

```
# - data2.csv 파일을 전처리한다.  
# - 사고유형구분별 사상자수, 사망자수, 중상자수, 경상자수, 부상자수를 구한다.
```

```
# 1. 데이터 로드  
rawdata <- read.csv(file.choose()) #데이터 로드  
summary(rawdata) # summary  
head(rawdata) # head
```

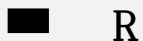
```
# 2. 데이터 전처리 1 : 사고유형구분별 사상자수 구하기  
# 변수 초기화  
all_stepless_crossing = 0  
all_walking_oldman = 0  
all_walking_children = 0  
all_schoolzone_children = 0  
all_bicycle = 0
```

```
unique(rawdata$사고유형구분)  
# 사고유형별 사상자수를 더해서 각각 구한다.  
for (i in c(1:length(rawdata$사고유형구분))){  
  if (rawdata$사고유형구분[i] == '무단횡단'){  
    all_stepless_crossing = all_stepless_crossing + rawdata$사상자수[i]  
  } else if (rawdata$사고유형구분[i] == '보행노인'){  
    all_walking_oldman = all_walking_oldman + rawdata$사상자수[i]  
  } else if (rawdata$사고유형구분[i] == '보행어린이'){  
    all_walking_children = all_walking_children + rawdata$사상자수[i]  
  } else if (rawdata$사고유형구분[i] == '스쿨존어린이'){  
    all_schoolzone_children = all_schoolzone_children + rawdata$사상자수[i]  
  } else if (rawdata$사고유형구분[i] == '자전거'){  
    all_bicycle = all_bicycle + rawdata$사상자수[i]  
  }  
}  
all_data <- data.frame(all_stepless_crossing,  
                      all_walking_oldman,  
                      all_walking_children,  
                      all_schoolzone_children,  
                      all_bicycle) # 데이터프레임 만들기
```

```
# 3. 데이터 전처리 2 : 사고유형구분별 사망자수 구하기  
# 변수 초기화  
dead_stepless_crossing = 0  
dead_walking_oldman = 0  
dead_walking_children = 0  
dead_schoolzone_children = 0  
dead_bicycle = 0
```

```
unique(rawdata$사고유형구분)  
# 사고유형별 사망자수를 더해서 각각 구한다.  
for (i in c(1:length(rawdata$사고유형구분))){  
  if (rawdata$사고유형구분[i] == '무단횡단'){  
    dead_stepless_crossing = dead_stepless_crossing + rawdata$사망자수[i]  
  } else if (rawdata$사고유형구분[i] == '보행노인'){  
    dead_walking_oldman = dead_walking_oldman + rawdata$사망자수[i]  
  } else if (rawdata$사고유형구분[i] == '보행어린이'){  
    dead_walking_children = dead_walking_children + rawdata$사망자수[i]  
  } else if (rawdata$사고유형구분[i] == '스쿨존어린이'){  
    dead_schoolzone_children = dead_schoolzone_children + rawdata$사망자수[i]  
  } else if (rawdata$사고유형구분[i] == '자전거'){  
    dead_bicycle = dead_bicycle + rawdata$사망자수[i]  
  }  
}  
dead_data <- data.frame(dead_stepless_crossing,  
                        dead_walking_oldman,  
                        dead_walking_children,
```

데이터 데이터 전처리 4



```
        dead_schoolzone_children,
        dead_bicycle) # 데이터프레임 만들기

# 4. 데이터 전처리 3 : 사고유형구분별 증상자수 구하기
# 변수 초기화
serious_stepless_crossing = 0
serious_walking_oldman = 0
serious_walking_children = 0
serious_schoolzone_children = 0
serious_bicycle = 0

unique(rawdata$사고유형구분)
# 사고유형별 증상자수를 더해서 각각 구한다.
for (i in c(1:length(rawdata$사고유형구분))){
  if (rawdata$사고유형구분[i] == '무단횡단'){
    serious_stepless_crossing = serious_stepless_crossing + rawdata$증상자수[i]
  } else if (rawdata$사고유형구분[i] == '보행노인'){
    serious_walking_oldman = serious_walking_oldman + rawdata$증상자수[i]
  } else if (rawdata$사고유형구분[i] == '보행어린이'){
    serious_walking_children = serious_walking_children + rawdata$증상자수[i]
  } else if (rawdata$사고유형구분[i] == '스쿨존어린이'){
    serious_schoolzone_children = serious_schoolzone_children + rawdata$증상자수[i]
  } else if (rawdata$사고유형구분[i] == '자전거'){
    serious_bicycle = serious_bicycle + rawdata$증상자수[i]
  }
}

serious_data <- data.frame(serious_stepless_crossing,
                           serious_walking_oldman,
                           serious_walking_children,
                           serious_schoolzone_children,
                           serious_bicycle) # 데이터프레임 만들기

# 5. 데이터 전처리 4 : 사고유형구분별 경상자수 구하기
# 변수 초기화
light_stepless_crossing = 0
light_walking_oldman = 0
light_walking_children = 0
light_schoolzone_children = 0
light_bicycle = 0

unique(rawdata$사고유형구분)
# 사고유형별 경상자수를 더해서 각각 구한다.
for (i in c(1:length(rawdata$사고유형구분))){
  if (rawdata$사고유형구분[i] == '무단횡단'){
    light_stepless_crossing = light_stepless_crossing + rawdata$경상자수[i]
  } else if (rawdata$사고유형구분[i] == '보행노인'){
    light_walking_oldman = light_walking_oldman + rawdata$경상자수[i]
  } else if (rawdata$사고유형구분[i] == '보행어린이'){
    light_walking_children = light_walking_children + rawdata$경상자수[i]
  } else if (rawdata$사고유형구분[i] == '스쿨존어린이'){
    light_schoolzone_children = light_schoolzone_children + rawdata$경상자수[i]
  } else if (rawdata$사고유형구분[i] == '자전거'){
    light_bicycle = light_bicycle + rawdata$경상자수[i]
  }
}

light_data <- data.frame(
  light_stepless_crossing,
  light_walking_oldman,
  light_walking_children,
  light_schoolzone_children,
  light_bicycle) # 데이터프레임 만들기
```


데이터 데이터 전처리 4

R

```
# 6. 데이터 전처리 5 : 사고유형구분별 부상자수 구하기
# 변수 초기화
injured_stepless_crossing = 0
injured_walking_oldman = 0
injured_walking_children = 0
injured_schoolzone_children = 0
injured_bicycle = 0

unique(rawdata$사고유형구분)
# 사고유형별 부상자수를 더해서 각각 구한다.
for (i in c(1:length(rawdata$사고유형구분))){
  if (rawdata$사고유형구분[i] == '무단횡단'){
    injured_stepless_crossing = injured_stepless_crossing + rawdata$부상자수[i]
  } else if (rawdata$사고유형구분[i] == '보행노인'){
    injured_walking_oldman = injured_walking_oldman + rawdata$부상자수[i]
  } else if (rawdata$사고유형구분[i] == '보행어린이'){
    injured_walking_children = injured_walking_children + rawdata$부상자수[i]
  } else if (rawdata$사고유형구분[i] == '스쿨존어린이'){
    injured_schoolzone_children = injured_schoolzone_children + rawdata$부상자수[i]
  } else if (rawdata$사고유형구분[i] == '자전거'){
    injured_bicycle = injured_bicycle + rawdata$부상자수[i]
  }
}

injured_data <- data.frame(injured_stepless_crossing,
                           injured_walking_oldman,
                           injured_walking_children,
                           injured_schoolzone_children,
                           injured_bicycle) # 데이터프레임 만들기

# 7. 사고유형구분별 사상자수, 사망자수, 중상자수, 경상자수, 부상자수 잘 구했나 확인
a = all_stepless_crossing + all_walking_oldman + all_walking_children + all_schoolzone_children + all_bicycle
b = dead_stepless_crossing + dead_walking_oldman + dead_walking_children + dead_schoolzone_children + dead_bicycle
c = serious_stepless_crossing + serious_walking_oldman + serious_walking_children + serious_schoolzone_children + serious_bicycle
d = light_stepless_crossing + light_walking_oldman + light_walking_children + light_schoolzone_children + light_bicycle
e = injured_stepless_crossing + injured_walking_oldman + injured_walking_children + injured_schoolzone_children + injured_bicycle

a == sum(rawdata$사상자수)
b == sum(rawdata$사망자수)
c == sum(rawdata$중상자수)
d == sum(rawdata$경상자수)
e == sum(rawdata$부상자수)

# 8. 최종 데이터
data <- data.frame(all_data,
                   dead_data,
                   serious_data,
                   light_data,
                   injured_data)

# 9. 저장
write.csv(data, "C:/Users/JIHYE/Desktop/데이터시각화HW - 도로교통공단 교통사고 데이터/2. 데이터_good/Final data/all_dead_ser
```

■ 데이터 전처리

- 데이터 중 '사고년도, 사고유형구분, 시도시군구명, 사고지역위치명, 발생건수, 사상자수, 사망자수, 중상자수, 경상자수, 부상자수, 위도, 경도' 데이터를 선택했다.
- 데이터의 열 이름을 정제했다.
- 데이터를 사고년도 기준으로 오름정렬 했다.
- 시도시군구명의 가장 앞 단어를 추출하여 ' 시도 ' 열을 만들었다.
- 사고유형구분별 교통사고 발생건수를 구했다.
- 사고년도별 교통사고 발생건수를 구했다.
- 시도별 교통사고 발생건수를 구했다.
- 사고유형구분별 사상자수, 사망자수, 중상자수, 경상자수, 부상자수를 구했다.



문제제기 및 가설 설정

문제제기 및 가설설정 문제제기



- 2019년 기준 한국에서는 교통사고로 인해 사흘에 한 명 꼴로 사망한다.
- 도로교통공단의 교통사고 데이터를 확인하여 교통사고 현황을 확인할 수 있어야 한다.
- 도로교통공단의 교통사고 데이터를 확인하여 교통사고를 줄일 수 있는 방법을 모색해야 한다.

문제제기 및 가설설정 가설

연 도	'07	'08	'09	'10	'11	'12	'13	'14	'15	'16	'17	'18
대수(만 대)	1,643	1,679	1,733	1,794	1,844	1,887	1,940	2,012	2,099	2,180	2,253	2,320
증가(천 대)	533	366	531	616	496	433	530	717	872	813	725	674
증가율 (%)	3.4	2.2	3.2	3.6	2.8	2.3	2.8	3.7	4.3	3.9	3.3	3.0



- 1. 연도별로 자동차 등록 대수가 늘어남에 따라, 교통사고는 점차 늘어날 것이다.
- 2. 인구가 많은 도시일 수록 교통사고가 많이 일어날 것이다.
- 3. 보행중 교통사고가 가장 많을 것이다.

4

결론

결론 가설검증



1. 연도별로 자동차 등록 대수가 늘어남에 따라, 교통사고는 점차 늘어날 것이다. (X)
2. 인구가 많은 도시일 수록 교통사고가 많이 일어날 것이다. (△)
3. 보행중 교통사고가 가장 많을 것이다. (O)

결론

1. 연도별로 교통사고는 점차 감소하는 추이를 보였다. 2015년 잠깐 늘어나는 듯 했으나 그 이후 다시 줄어들었다. 그 원인으로서는 음주운전 처벌 강화, 도심 제한속도 인하, 교통사고 사망자 30% 줄이기 업무 협약 등의 노력을 생각해 볼 수 있다. 현재의 교통사고가 감소하는 추세를 유지하기 위한 노력이 필요하다.
2. 교통사고는 서울이 1위, 경기도가 2위, 대구가 3위였고, 인구 순위는 경기도가 1위, 서울이 2위, 부산이 3위였다. 둘은 비슷한 추이를 보였지만 정비례 하지는 않았다.
3. 교통사고는 보행노인이 3919로 가장 많았고, 자전거가 두 번째 였다. 보행자 중 보행노인의 교통사고를 줄이기 위한 정책 및 논의가 필요 하다.

감사합니다

