

화재 발생시 피해 예측 분석

FPS

[Fire Protection Supporters]

목차

I. 분석 배경 및 목적

II. 데이터 분석

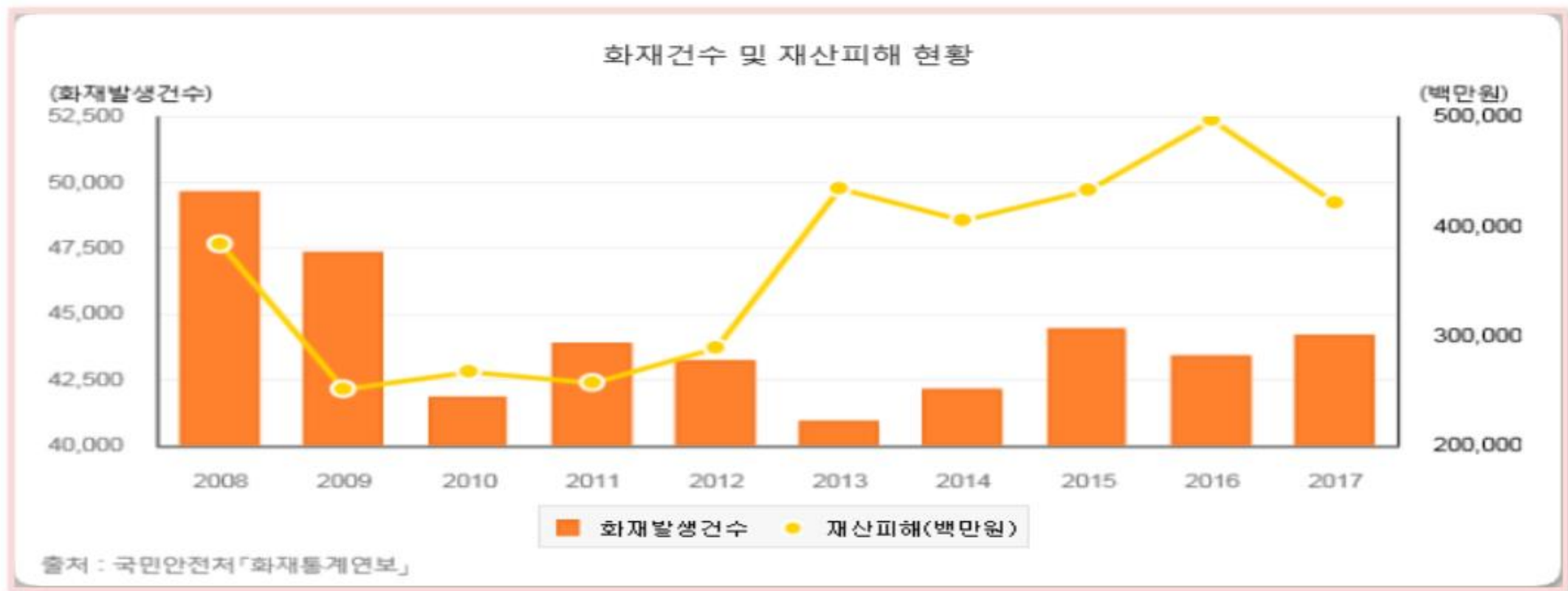
III. 분석 결과

IV. 참고자료



우리나라 화재 현황

I. 분석 배경 및 목적



“화재 발생 건수에 비하여 재산피해가 증가 하는 추세이다.”

주제 선정 배경

I. 분석 배경 및 목적

KT 아현 지사



✓ 재산피해: 80억



“서비스 기관 건물일 경우 피해액이 더욱 상승한다.”

화재 피해 현황

“재산 피해의 큰 비중
을 차지하는 화재는
소수의 대형화재”

화재 피해 요소

“화재를 분석하여
대형 화재에 관련된
요소를 찾음”



화재 규모 파악을 통해 대형화재 예방

A firefighter in full protective gear, including a helmet and a self-contained breathing apparatus (SCBA), is shown in profile, looking down. The background is a large fire with bright orange and yellow flames. A red horizontal bar is positioned above the firefighter's head, and another red horizontal bar is positioned below the firefighter's head.

2. 데이터 분석

II. 데이터 분석

출처 : 통계청 데이터 : <http://kosis.kr>

CSV

소방청_화재발생 정보_2017년

CSV

소방청_사건 관망구 화재발생 현황_2...

다운로드

상세정보

로그인

소방청_화재발생 정보_2017년

업데이트 주기	수시	사건등록예정일	2018-10-26
비밀부과유무	무로	비밀부과기준 및 단위	없음
다운로드 횟수	0		
등록일	2018-10-23	수정일	2018-10-26
비밀화법명	이동화법명 제한 없음		
제공형태	전자기출하제 제공 가능		
URL	https://www.data.go.kr/faces/tableView/fileDownload.do?atchField=FILE_000000001486799&fileDetail=Y		
설명	시도별 화재발생 상세주소, 화재원인, 인명피해, 재산피해		
관리부서명	정보통신망안전실	관리부서 연락처	044-205-7264

미리보기

파일 데이터의 일부 내용을 제공하고 있으며, 전체 내용이 열려진 경우 해당 파일을 다운로드 받으시기 바랍니다.

연번	사건	부상	인명피해(총인구)	화재발생년월일	시도	사건구	유형	대
1	0	0	0	2017-01-01 0:00	서울특별시	강남구	불연	
2	0	0	0	2017-01-01 0:09	경상북도	문경시	불연	
3	0	0	0	2017-01-01 0:14	경상남도	거제시	불연	
4	0	0	0	2017-01-01 0:57	경기도	안산시 남부구	불연	
5	0	0	0	2017-01-01 0:59	전라북도	익산시	인명피해 1기	
6	0	0	0	2017-01-01 1:11	경기도	하남시	불연	

출처 : 공공데이터 포털 :
<https://www.data.go.kr/dataset/15002228/fileData.do>



“공공데이터와 통계청에서 자료 수집”

자료접수일 : 2019-01-22 / 수록기간 : 월 2017.01 ~ 2018.12 / 자료문헌처 : 044-201-3861

입출정보 + 항목[4/4] 시도[12/12] 시군구[1/241] 책명[1/1/5] 시열[1/96]

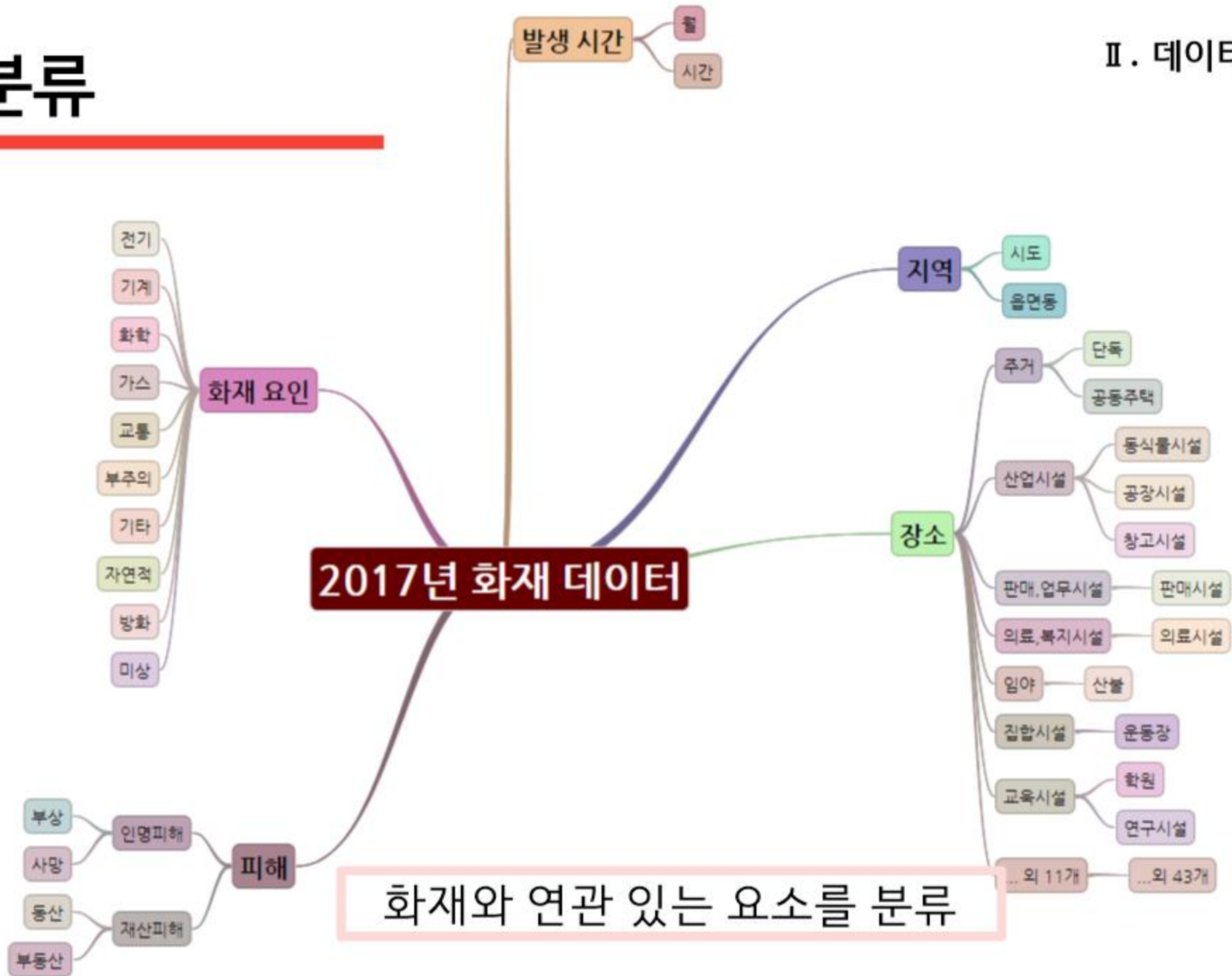
(단위 : 개)

시도	시군구	책명	2018. 12			
			권종	자료종	발행종	개
서울	계					
부산	계					
대구	계					
인천	계					
광주	계					
대전	계					
울산	계					
경기	계					
세종	계					
강원	계					
충북	계					
충남	계					
전북	계					
전남	계					
경북	계					
경남	계					
제주	계					
	합계		829	278,904	171,154	451,887

출처 : 통계청 데이터 : <http://kosis.kr>[illegible]

데이터 분류

II. 데이터 분석



결측치 처리

II. 데이터 분석

지역별, 장소 분류별 피해를 결측치에 대입



```
In [31]: file[file["부동산"]==0]
```

Out[31]:

연번	사망	부상	인명 피해 (명)	화재 발생 년월일	월	시간	월가 중지	시도	시군구	...	재산 피해 소계2	부동산	동산	부동산 피해 소계	1억 이상 가중치	1억 이상 가중치2	1억 이상 가중치3	장소 대분류	장소 중분류	장소 소분류
----	----	----	-----------------	-----------------	---	----	----------	----	-----	-----	-----------------	-----	----	-----------------	-----------------	------------------	------------------	-----------	-----------	-----------

0 rows x 23 columns

```
In [9]: len(file[file["부동산"]==0])
```

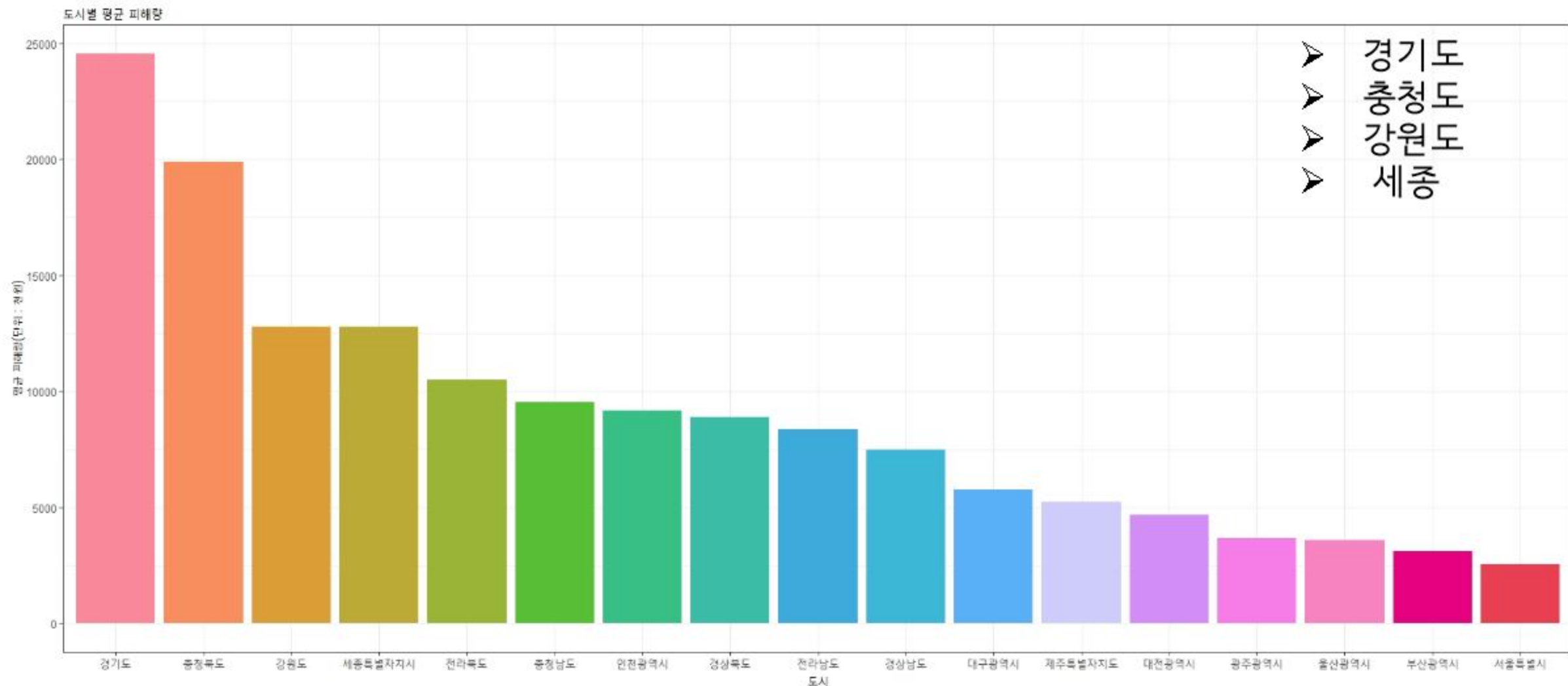
Out[9]: 4244



부동산 결측치 4244개 확인

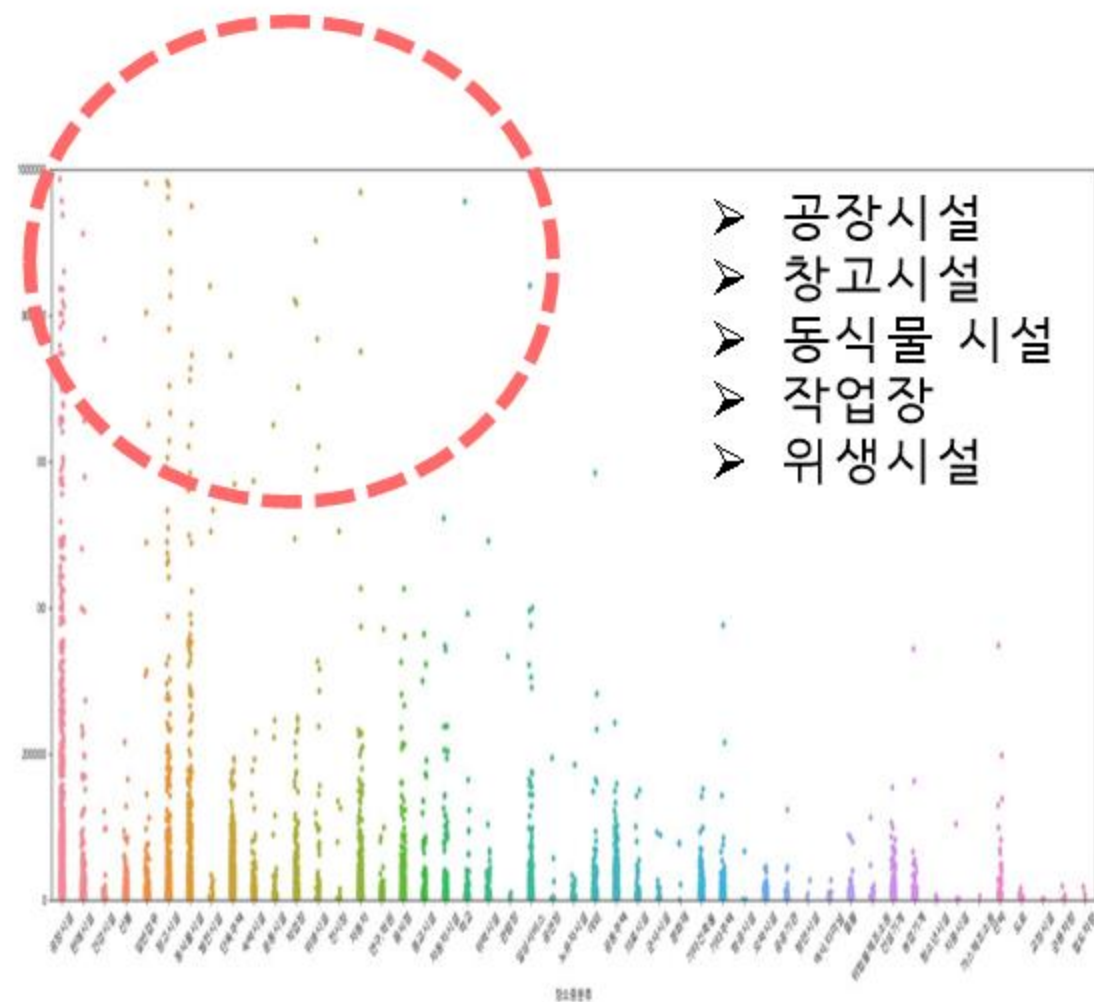
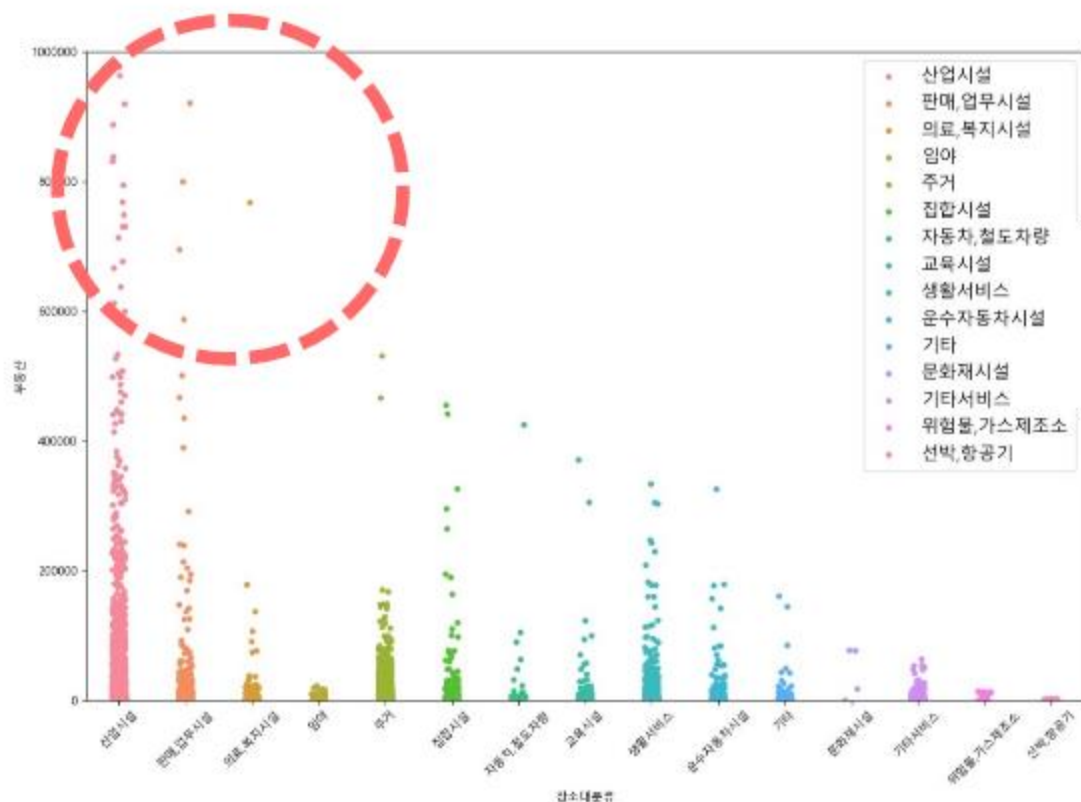
지역별 평균 피해량

Ⅱ. 데이터 분석



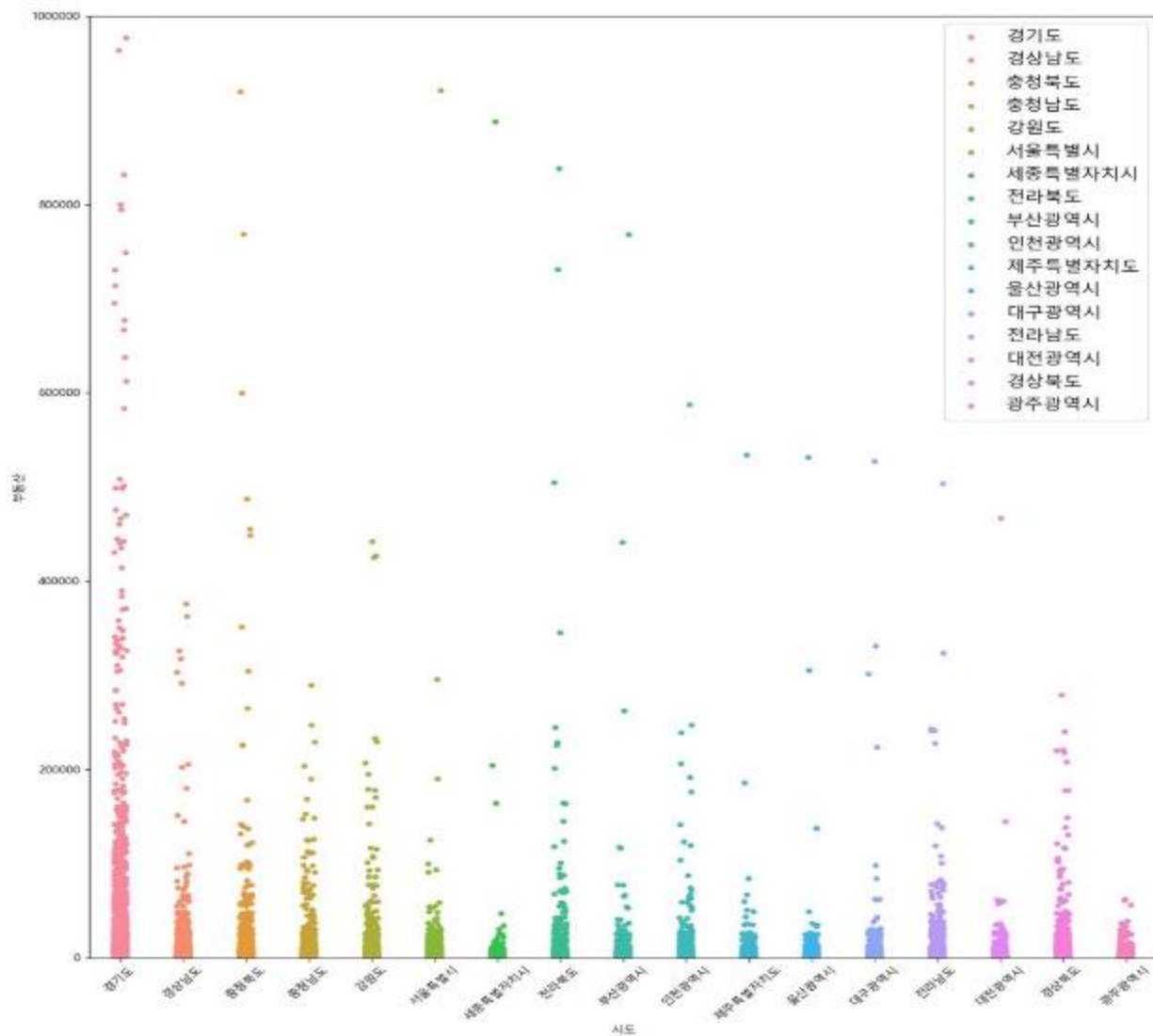
장소별 피해량

Ⅱ. 데이터 분석



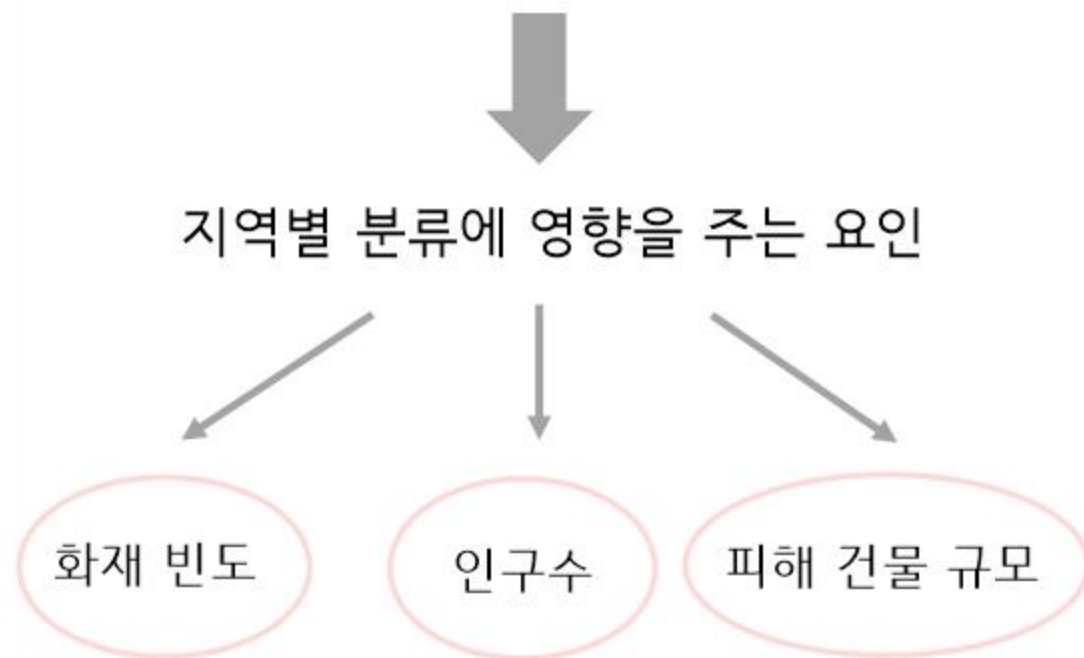
장소 대분류에서 산업, 판매, 업무 시설의 재산피해가 가장 크다

데이터 추출



II. 데이터 분석

왜 경기도 피해가 큰가?



피해가 규모가 큰 공장, 창고 시설들이 경기도
에 모여 있음을 파악 -> 특성 적용

인터뷰

Ⅱ. 데이터 분석

A1. 1월 화재는 부주의가 많다. 주로 아파트화재인데 부주의로 난방 관련 부주의가 많다.



A2. 화재 위험도에 따라서 관리를 다르게 하고있다.

A3. 지하실에서불이 나면 접근과 소화가 어렵다.

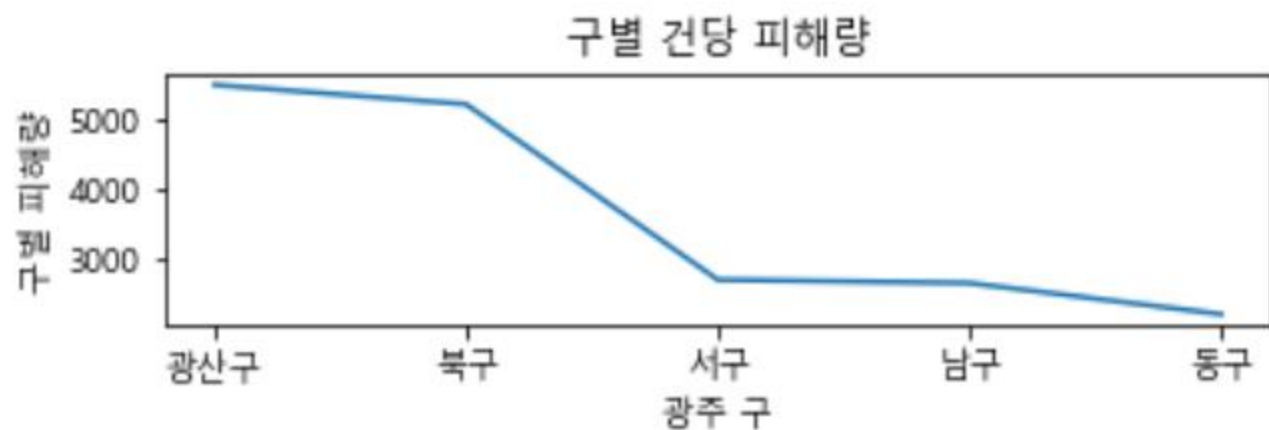
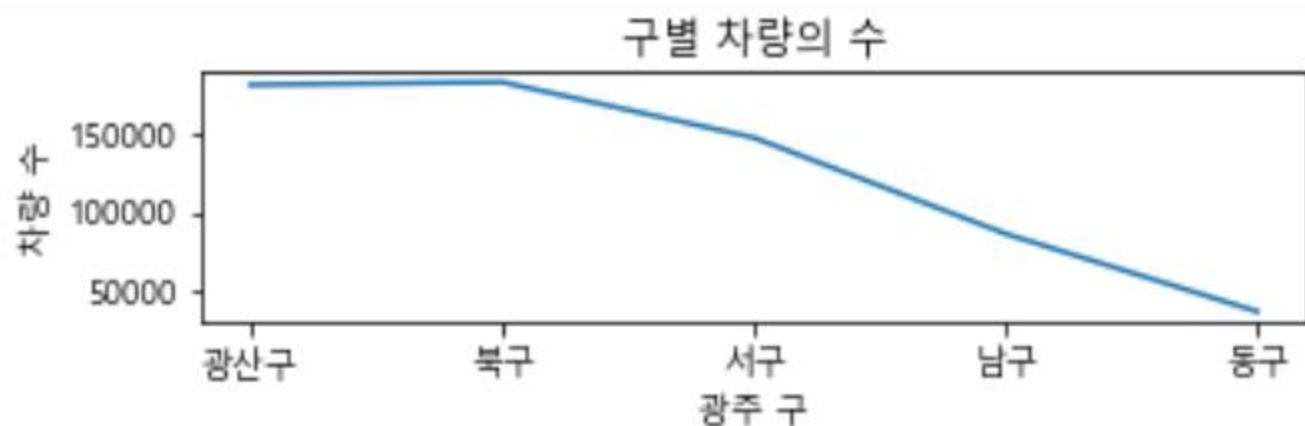
Q1. 화재발생에 영향을 크게 미치는 것은?

Q2. 화재 위험도가 도움이 되는지 ?

Q3. 화재 진압시 어려운 곳은?

데이터 상관관계 파악

Ⅱ. 데이터 분석



광주지역에 소급 적용



구별 차량 수에 따른
건당 피해량 비교



3. 분석 결과

분석 결과(Logistic Regression)

```
In [745]: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, stratify=y)
print("x_train : {}, x_test : {}, y_train : {}, y_test : {}".format(x_train.shape, x_test.shape, y_train.shape, y_test.shape))
model = LogisticRegression()
model.fit(x_train, y_train)
# 인명피해, 화재발생율, 장소분류(52개로분류), 사고요인(43개로분류)
print("train socre : {}".format(model.score(x_train, y_train)))
print("test socre : {}".format(model.score(x_test, y_test)))
#print("컬럼들의 중요도 : {}".format(model.feature_importances_.sort()))

print(metrics.classification_report(y_test, model.predict(x_test))) # 정밀도, 재현율, f1 스코어, 서포트를 알려준다.
```

```
x_train : (35342, 96), x_test : (8836, 96), y_train : (35342,), y_test : (8836,)
train socre : 0.802218323807368
test socre : 0.8020597555454957
```

	precision	recall	f1-score	support
1	1.00	0.02	0.03	65
2	0.80	0.62	0.70	3230
3	0.80	0.92	0.86	5541
avg / total	0.80	0.80	0.79	8836



분석 결과

Train score : 0.80
Test score : 0.80

분석 결과(RFC)

III. 분석 결과

```
In [740]: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, stratify=y)
print("x_train : {}, x_test : {}, y_train : {}, y_test : {}".format(x_train.shape, x_test.shape, y_train.shape, y_test.shape))
model = RandomForestClassifier()
model.fit(x_train, y_train)
# 인명피해, 화재발생월, 장소분류(52개로분류), 사고요인(43개로분류)
print("train socre : {}".format(model.score(x_train, y_train)))
print("test socre : {}".format(model.score(x_test, y_test)))
#print("컬럼들의 중요도 : {}".format(model.feature_importances_.sort()))

print(metrics.classification_report(y_test, model.predict(x_test))) # 정밀도, 재현율, f1 스코어, 서포트를 알려준다.
```

```
x_train : (35342, 96), x_test : (8836, 96), y_train : (35342,), y_test : (8836,)
train socre : 0.8533472921736178
test socre : 0.8036441828881847
```

	precision	recall	f1-score	support
1	0.25	0.03	0.05	65
2	0.76	0.70	0.73	3230
3	0.83	0.88	0.85	5541
avg / total	0.80	0.80	0.80	8836



분석 결과

Train score : 0.85
Test score : 0.80

분석 결과(XGB)

Ⅲ. 분석 결과

```
In [59]: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, stratify=y)
print("x_train : {}, x_test : {}, y_train : {}, y_test : {}".format(x_train.shape, x_test.shape, y_train.shape, y_test.shape))
model = XGBClassifier()
model.fit(x_train, y_train)
# 인명피해, 화재발생율, 장소분류(52개로분류), 사고요인(43개로분류)
print("train socre : {}".format(model.score(x_train, y_train)))
print("test socre : {}".format(model.score(x_test, y_test)))
#print("특성들의 중요도 : {}".format(model.feature_importances_.sort()))

print(metrics.classification_report(y_test, model.predict(x_test))) # 정밀도, 재현율, f1 스코어, 서포트를 알려준다.

x_train : (35342, 97), x_test : (8836, 97), y_train : (35342,), y_test : (8836,)
```

```
train socre : 0.8165072718012563
```

```
C:\Users\chosun\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:151: DeprecationWarning: The truth value of an empty array is ambiguous. Returning False, but in future this will result in an error.
not empty.
if diff:
```

```
test socre : 0.8066998641919421
```

```
C:\Users\chosun\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:151: DeprecationWarning: The truth value of an empty array is ambiguous. Returning False, but in future this will result in an error.
not empty.
if diff:
```

	precision	recall	f1-score	support
1	1.00	0.25	0.40	65
2	0.80	0.64	0.71	3230
3	0.81	0.91	0.86	5541
avg / total	0.81	0.81	0.80	8836

분석 결과

Train score : 0.81

Test score : 0.80

등급별 정밀도, 재현율, f1 스코어, 서포트
가 높은 점수를 보임

> 여러 모델 사용 결과 XGB가 좋은 결과를 보여 주었다

분석 결과(모델 비교)

Ⅲ. 분석 결과

	Logistic Regression	Random ForestClassifire	XGBClassifire
정밀도	0.87	0.61	0.87
재현율	0.52	0.54	0.6
F1 score	0.53	0.54	0.67
1등급 예측 support	65	65	65

분석 결과

Ⅲ. 분석 결과

교차검증

```
In [608]: result = cross_val_score  
print("모델의 정확도 :")
```

```
C:\Users\chosun\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:151:  
DeprecationWarning: The truth value of an empty array is ambiguous. Returning F  
alse, but in future this will result in an error. Use `array.size > 0` to check  
that an array is not empty.  
if diff:  
C:\Users\chosun\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:151:  
DeprecationWarning: The truth value of an empty array is ambiguous. Returning F  
alse, but in future this will result in an error. Use `array.size > 0` to check  
that an array is not empty.  
if diff:  
C:\Users\chosun\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:151:  
DeprecationWarning: The truth value of an empty array is ambiguous. Returning F  
alse, but in future this will result in an error. Use `array.size > 0` to check  
that an array is not empty.  
if diff:  
C:\Users\chosun\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:151:  
DeprecationWarning: The truth value of an empty array is ambiguous. Returning F  
alse, but in future this will result in an error. Use `array.size > 0` to check  
that an array is not empty.  
if diff:
```

```
모델의 정확도 : 0.800134376592481
```

DeprecationWarning: The truth value of an empty array is ambiguous. Returning False, but in future this will result in an error. Use `array.size > 0` to check that an array is not empty.

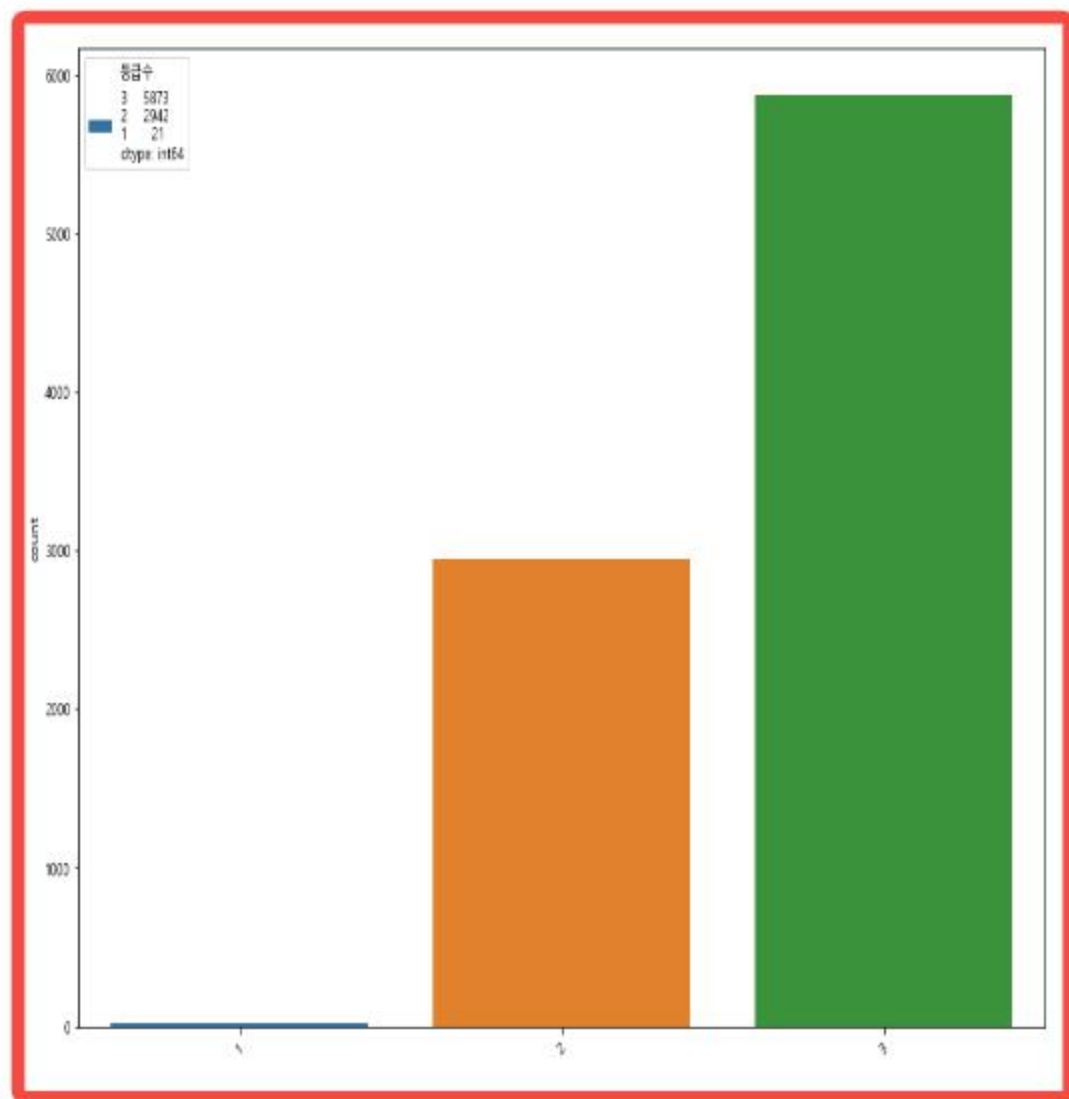
if diff:

모델의 정확도 : 0.800134376592481

> 교차 검증을 통하여 모델을 일반화 시켰다

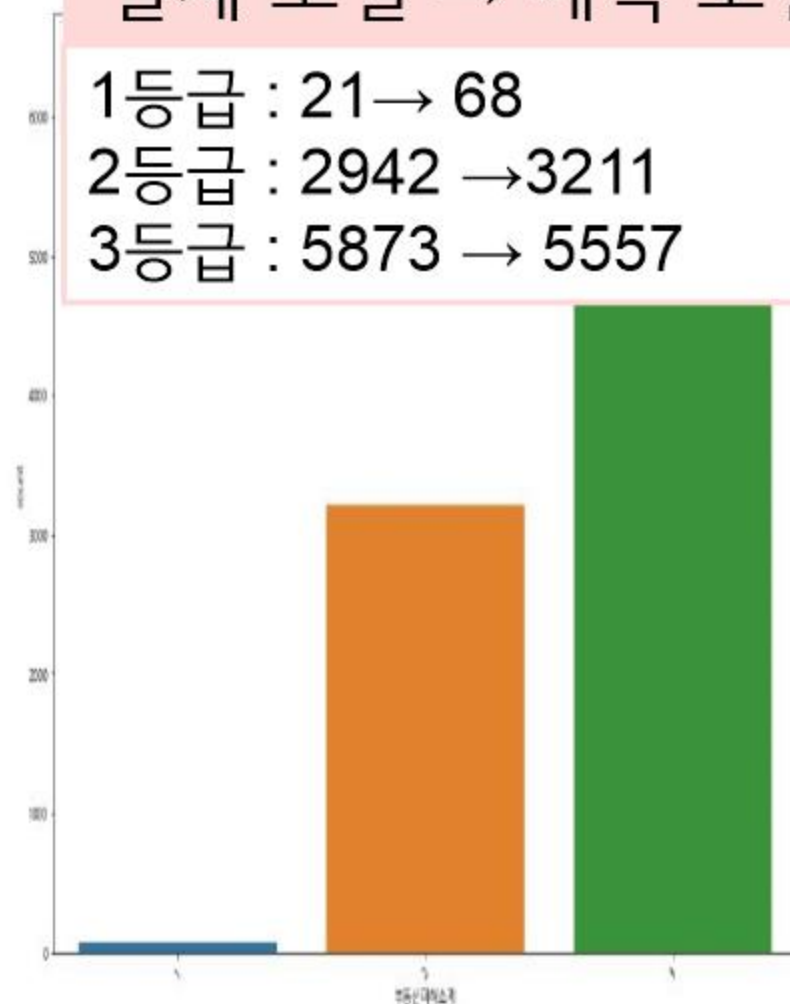
데이터 분석(XGB)

Ⅲ. 분석 결과



실제 모델 → 예측 모델

1등급 : 21 → 68
2등급 : 2942 → 3211
3등급 : 5873 → 5557



데이터 분석 시각화

Ⅲ. 분석 결과



- > 각각 등급표를 바탕으로 지역별 위험 관리 예측
- > 추가적인 지역 데이터를 세분화하여 관리 할 수 있다.

위험 등급별 시각화

Ⅲ. 분석 결과



등급별 지역별 화재 화재 건수

취약지역 확대 시각화

Ⅲ. 분석 결과



등급별 지역별 화재 화재 건수

개선할 점

Ⅲ. 분석 결과



✓ 취약지역 순찰 강화



<이미지제공=서울시청>

✓ 소방차 전용 구역 확대
✓ 진입로 경고문 부착



✓ 화재 예산 측정에 도움



✓ 예방 중심 정책



✓ 자발적 화재 예방

한계점 및 아쉬운점

Ⅲ. 분석 결과

데이터 분석

- ✓ 위험 등급 수를 자세하게 나누지 못한점 (A,B,C등급)

데이터 수집

- ✓ 데이터 테이블이 완벽하지 않은 점

전문 지식

- ✓ 데이터 분석 기준 설정에 있어서 발생하는 오차

시각화

- ✓ 시각화에 있어서 기존 화재 위험표와 상대적 차이

참고자료

IV. 참고 자료

데이터	형식	출처	기준연도	비고
소방청_화재발생정보_2017년	CSV	공공데이터 포털	2017	2017년 화재 정보
시간별 화재요인	CSV	통계청(크롤링)	2017년	2017년 화재 요인
2017년 광주 자동차 현황	CSV	통계청	2017년	2017년 광주 자동차 등록 현황

분석tool



-파이썬
자료 전처리
LogisticRegression,
RFC,XGB



-R studio
자료 시각화

참고 자료

-논문

건축물의 화재 취약성에 따른 등급화 방안
연구, 김한용, 2011년, 인재대학교

-인터뷰

북구 우산119 안전센터

-화재 보험협회 간행물

반복되는 대형화재의 시사점과 위험관리 개선방안

화재 피해 예측 시스템

Q&A





경청해주셔서
감사합니다!