

비즈니스 인텔리전스 보고서

2022.06.05

강원도 소방용수시설 최적 입지 선정

20173204 광명빈

20173218 김주형

20181203 김도훈

20198095 TA NGOC HA



1. 서론

1.1 상황분석 및 제안

[ISSUE] 동해안 덮친 최악 산불... 서울 면적 3분의 1 초토화

소방방재신문 2022.04.20. (<https://fpn119.co.kr/176510>)

지난 3월 강원 삼척과 경북 울진에서 발생한 산불로 인해 임야 2만 523헥타르가 소실되고 다수의 이재민이 발생했다.

겨울 가뭄과 강풍을 타고 순식간에 번진 경북 울진·강원 삼척 산불이 역대 최장기 산불 기록을 세우며 213시간 만에 주불을 진화했다. 강원 강릉과 영월 등 동해안 곳곳에서 발생한 산불까지 포함하면 서울 면적의 3분의 1 이상이 불타 두 번째 피해 규모다.



서울 면적 3분의 1 불타... 역대 두 번째 피해

울진에서 시작해 삼척까지 번진 산불은 213시간 만에 주불이 진화되면서 산불 역사상 최장 시간을 기록했다. 1986년 관련 통계를 집계한 이후 가장 길게 이어진 2000년도 동해안 산불(191시간)을 뛰어넘는다.

중앙재난안전대책본부(이하 중대본)가 집계한 이번 산불 피해 면적은 최초 2만 4940ha였다. 그러나 4월 6일을 기점으로 중대본은 2만523ha로 수정 발표했다. 지역별로는 울진·삼척이 1만6302ha, 강릉·동해가 4221ha였다.

피해 면적은 산불 통계를 집계한 1896년 이후 2000년 동해안 산불(2만3천 794ha)에 이어 두 번째로 큰 수준이다. 서울 면적(6만500ha)과 비교하면 33.9%, 축구장 면적(0.714ha)의 경우 2만8744개, 여의도 면적(290ha, 윤중로 제방 안쪽 면적)은 70.8배 정도다.

이에 따라 강원지역 및 소방청에서는 소방용수시설의 효율적 관리를 위해 운영위원회 개최, 빅데이터를 활용하여 소방용수시설을 보강하는 계획을 추진하는 등 소방용수시설 설치·유지·보강·효율적 관리에 힘을 쓰고 있다.

인제소방서, 2022년 소방용수시설 운영위원회 개최

정현희 기자 | 기사입력 2022/04/28 [11:30]

인제소방서(서장 소기웅)는 지난 27일 오전 본서 소회의실에서 소방용수시설 운영위원회를 개최했다고 밝혔다.

운영위원회는 소방용수시설의 효율적인 관리 방법을 위한 유관기관 협조체제를 강화하고 소방용수시설의 설치·유지와 관리의 효율적인 운영방안을 모색하기 위해 추진됐다.

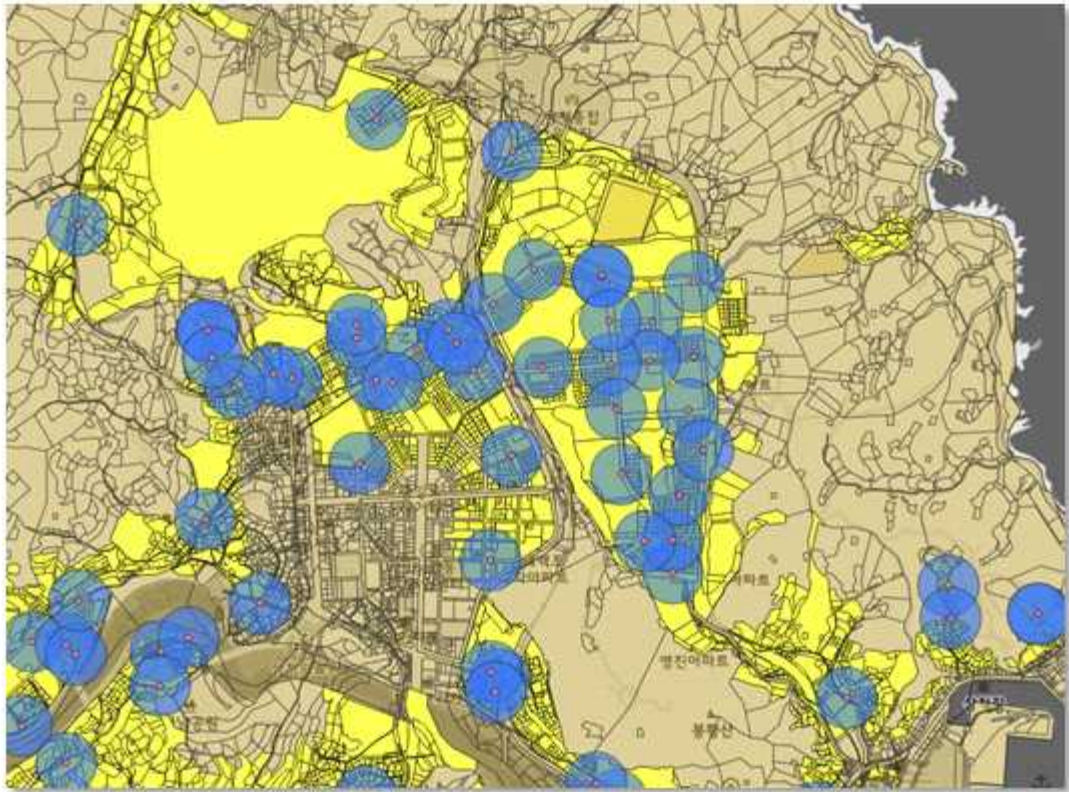
소방서 관계자는 “최근 동해안 산불 등에서 소화전과 비상소화장치함의 중요성이 입증됐고 인제군의 지리 여건상 평상시 소방용수시설의 관리가 매우 중요하다”며 “유관기관과의 협력을 통해 관내 소방용수시설 관리 등에 만전을 기하겠다”고 전했다.

“제2의 강원산불 막는다”...강원지역, 소방 용수 보강 추진

<https://www.edaily.co.kr/news/read?newsId=02945446625699384&mediaCod>

eNo=257&OutLnkChk=Y

소방청, 빅데이터 활용 소방용수 취약지역 분석



소방용수시설 및 상수관 설치 취약지역(삼척시) / 사진 = 소방청 제공.

<https://easylaw.go.kr/CSP/CnpClsMainBtr.laf?popMenu=ov&csmSeq=1574&ccfNo=3&cciNo=1&cnpClsNo=2>

1.2. 소방용수시설 설치기준

소방용수시설의 설치기준(제6조제2항관련)

1. 공통기준

가. 국토의계획및이용에관한법률 제36조제1항제1호의 규정에 의한 주거지역·상업지역 및 공업지역에 설치하는 경우 : 소방대상물과의 수평거리를 100미터 이하가 되도록 할 것

나. 가목 외의 지역에 설치하는 경우 : 소방대상물과의 수평거리를 140미터 이하가 되도록 할 것

2. 소방용수시설별 설치기준

가. 소화전의 설치기준 : 상수도와 연결하여 지하식 또는 지상식의 구조로 하고, 소방용호스와 연결하는 소화전의 연결금속구의 구경은 65밀리미터로 할 것

나. 급수탑의 설치기준 : 급수배관의 구경은 100밀리미터 이상으로 하고, 개폐밸브는 지상에서 1.5미터 이상 1.7미터 이하의 위치에 설치하도록 할 것

다. 저수조의 설치기준

(1) 지면으로부터의 낙차가 4.5미터 이하일 것

(2) 흡수부분의 수심이 0.5미터 이상일 것

(3) 소방펌프자동차가 쉽게 접근할 수 있도록 할 것

(4) 흡수에 지장이 없도록 토사 및 쓰레기 등을 제거할 수 있는 설비를 갖추어 줄 것

(5) 흡수관의 투입구가 사각형의 경우에는 한 변의 길이가 60센티미터 이상, 원형의 경우에는 지름이 60센티미터 이상일 것

(6) 저수조에 물을 공급하는 방법은 상수도에 연결하여 자동으로 급수되는 구조일 것

이와 같이 강원지역 및 소방청에서는 소방용수시설의 효율적 관리를 위해 소방용수시설 설치·유지·보강·효율적 관리에 관심을 가지고 개선하기 위한 노력을 기울였음을 확인할 수 있었습니다.

최근 발생한 대규모 산불 화재발생으로 인한 피해가 비효율적인 소방용수시설 배치와 직접적으로 연관되어 있다고 말하기는 어려울 수 있겠지만, 소방용수시설의 효율적 재배치를 통하여 초기 화재진압 여건 조성 및 화재 진압에 도움을 줄 수 있도록 하는 것이 우리조의 목표이다.

2. 데이터 수집 및 전처리

데이터	데이터 설명	특성	출처
강원도 소방본부 소방용수시설	소방용수시설 좌표	시군구명 위도 경도 배관 ...	공공데이터포털
구조 종별 신고 재난 데이터	구조종별 신고 재난 데이터 좌표	사고원인 위도 경도 현장도착시,분 ...	소방안전 빅데이터 플랫폼
화재 종별 신고 재난 데이터	화재 종별 신고 재난 데이터	사고원인 위도 경도 현장도착시,분 ...	소방안전 빅데이터 플랫폼
강원도 읍면동별 주민등록인구	강원도 읍면동별 주민등록인구	시군구역 행정인구 총인구 ...	공공데이터포털
강원도 건물통합정보	건물공간정보	—	국가공간정보포털

2.1. EDA

파이썬을 이용한 소방용수시설과 재난데이터 군집확인

```
import pandas as pd
import folium as g
from folium.plugins import MarkerCluster
```

Python

```
alarm = pd.read_csv('./data/강원도_강원도소방본부_소방용수시설.csv', encoding = 'cp949')
df = pd.read_csv('./data/구조_종별_신고_재난_데이터.csv', encoding='utf-8')
```

Python

필요 라이브러리 정의 후 데이터 로드

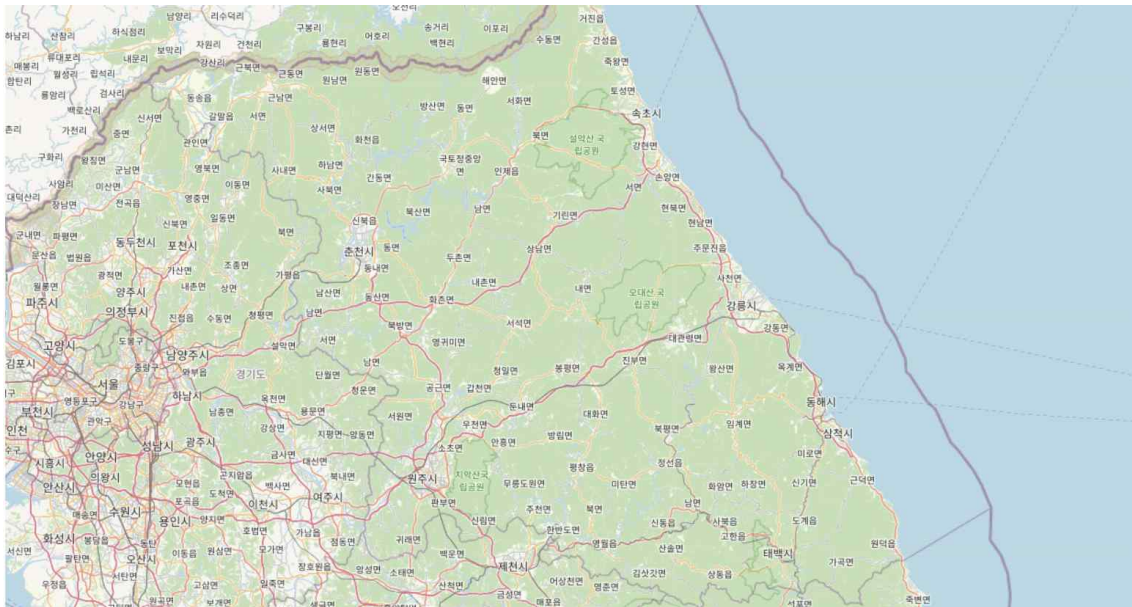
folium을 이용한 지도 시각화

```
m = g.Map(
    location=[37.8013, 128.3745],
    zoom_start=9
)

m.add_child(g.LatLngPopup()) # 지도에서 클릭하는 곳의 경도 위도 좌표를 표시

m
```

Python



강원도 전체가 보이게 중심 좌표와 zoom을 설정

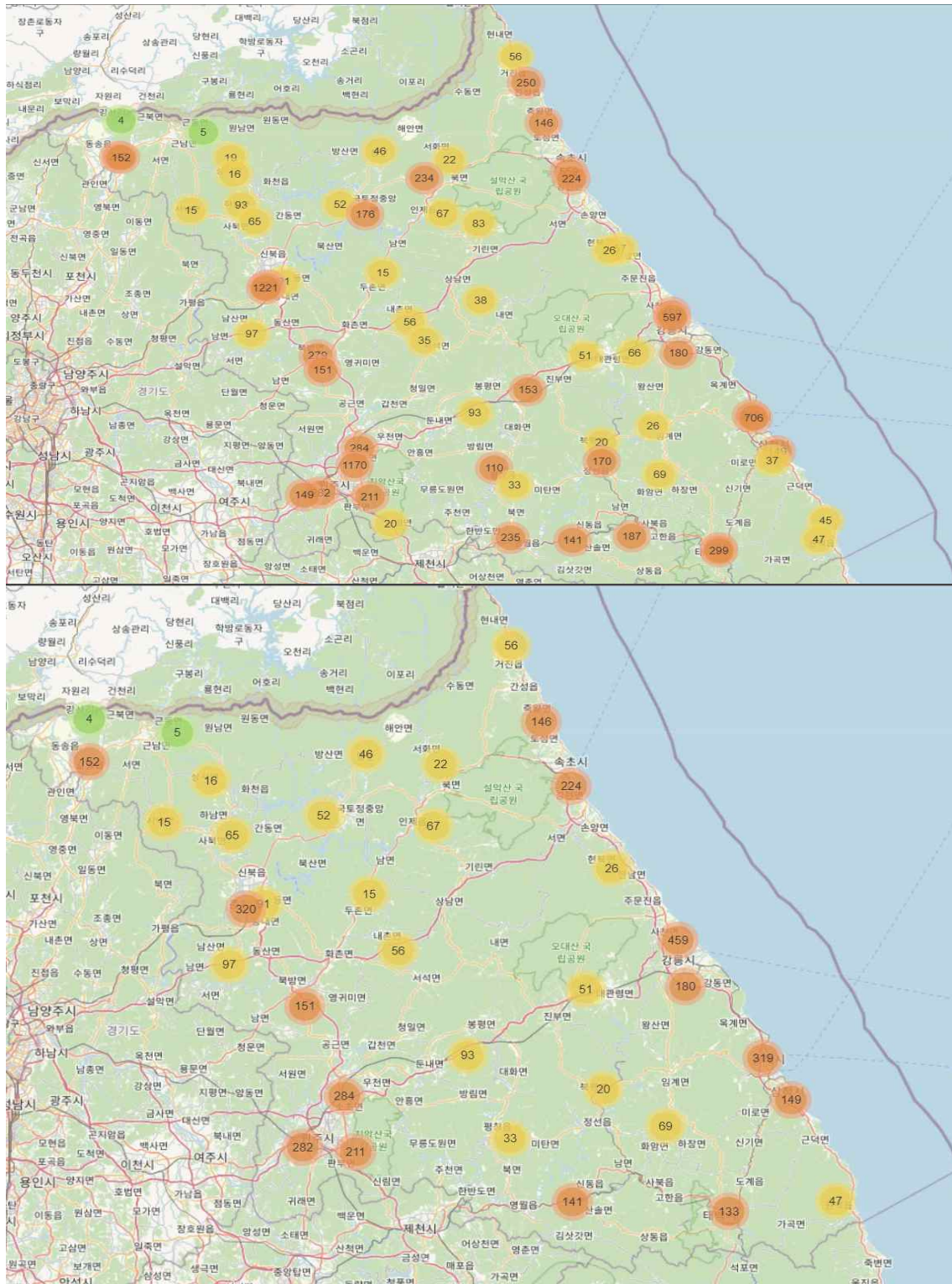
```
def plot(data):
    # 리스트를 이용해 여러 행의 데이터를 위,경도로 묶음
    locations = list(zip(data.LOC_INFO_X, data.LOC_INFO_Y))

    cluster = MarkerCluster(locations = locations)
    # 지도에 클러스터를 추가.
    cluster.add_to(m)
```

MarkerCluster 함수를 생성

위, 경도를 리스트를 이용하여 묶고 지도에 추가

소방용수시설, 화재발생위치 군집



소방용수시설과 화재발생위치를 대략적으로 파악 가능

2.2. 데이터 전처리

데이터 정의


```
# 필요 라이브러리 정의
import pandas as pd
```

전처리에 필요한 라이브러리 정의

```
# 데이터 정의
alarm = pd.read_csv('./data/강원도_강원도소방본부_소방용수시설.csv', encoding = 'cp949')
df = pd.read_csv('./data/구조_종별_신고_재난_데이터.csv', encoding='utf-8')

# 데이터 화재관련 값만 추출
df = df[(df['ACDNT_CAUSE'] == '기타화재') | (df['ACDNT_CAUSE'] == '화약류') | (df['ACDNT_CAUSE'] == '방화')
| (df['ACDNT_CAUSE'] == '폭발물') | (df['ACDNT_CAUSE'] == '화재확인')]

# 누락값 확인 및 누락값이 있는 열 삭제
df.isnull().sum()
df = df.dropna(axis = 1)
```

- 강원도 소방용수시설 데이터를 alarm으로 지정
- 강원도 구조 종별 신고, 재난 데이터를 df로 지정
- df에서 화재관련 값만 추출하여 df로 재지정
- df에서 누락값이 있는지 확인 후 제거

alarm.head()

	시설번호	시설유형 코드	시군구명	시군구코드	소재지번주소	위도	경도	안전센터명	보호대상 유무	사용가능 여부	설치연도	배관깊이	출수압력	배관지름	관할기관명	데이터기준일자
0	강촌 001호	1	춘천시	42110	강원도 춘천시 남산면 향촌리 145-1	37.786189	127.649237	강촌안전센터	N	Y	2007	0.7	7.0	100	춘천소방서	2021-11-19
1	강촌 002호	1	춘천시	42110	강원도 춘천시 남산면 향촌리 147-1	37.786725	127.647929	강촌안전센터	N	Y	2013	0.7	8.0	100	춘천소방서	2021-11-19
2	강촌 003호	1	춘천시	42110	강원도 춘천시 남산면 향촌리 126-9	37.785523	127.646686	강촌안전센터	Y	Y	2013	0.7	8.0	100	춘천소방서	2021-11-19
3	강촌 004호	1	춘천시	42110	강원도 춘천시 남산면 향촌리 219-15	37.784756	127.643488	강촌안전센터	Y	Y	2013	0.8	8.0	100	춘천소방서	2021-11-19
4	강촌 005호	1	춘천시	42110	강원도 춘천시 남산면 향촌리 238-2	37.781550	127.643992	강촌안전센터	Y	Y	2013	0.7	5.5	100	춘천소방서	2021-11-19

alarm데이터 확인

df.head()

	MSFRTN_RESC_REPRT_NO	ACDNT_CAUSE	PRCS_RESULT_SE_NM	DCLR_YMD	DCLR_TM	DCLR_YR	SEASON_SE_NM	QTR_SE	DCLR_MNTH	DCLR_DAY	...	SIDO_NM	SIGI
3	20204207506S00069	기타화재	기타	20201011	143700	2020	가을	4	10	11	...	강원도	
65	20214205202S00423	기타화재	오인(구조)	20210619	115100	2021	여름	2	6	19	...	강원도	
68	20214205202S00129	기타화재	오인(구조)	20210414	183000	2021	봄	2	4	14	...	강원도	
80	20204205103S00075	기타화재	안전조치	20201201	134800	2020	겨울	4	12	1	...	강원도	
101	20204205202S00074	기타화재	인명검색	20200212	92900	2020	겨울	1	2	12	...	강원도	

5 rows × 36 columns

df데이터 확인

두 개의 공통된 행을 만들어야 함

읍면동별 분석을 하기 위해서 소방용수시설의 읍면동 데이터 추가

```

# 소재지번호를 이용하여 읍면동 columns을 만들어야함
alarm.head()
# 소재지번호의 7번째 이후의 데이터를 만들어서 저장
alarm['EMD_NM'] = alarm['소재지번호'].str[7:]
alarm.head(5)

alarm['EMD_NM'] = alarm['EMD_NM'].str.strip() # 앞 공백을 제거
alarm['EMD_NM'].str.split(" ").head() # 공백을 기준으로 분할
dong = alarm['EMD_NM'].str.split(" ", expand=True) # expand = True 데이터 프레임 형태로 저장
dong

# 동의 첫번째 열을 EMD_NM로 지정
alarm['EMD_NM'] = dong[0]
alarm.head()

alarm['EMD_NM'].unique() # 일부 동의 띄어쓰기가 안되었어서 나누어 지지않음
alarm['EMD_NM'] = alarm['EMD_NM'].str.replace(pat=r'^\w|', repl=r'', regex=True) # 특수문자 제거
alarm['EMD_NM'].unique() # 숫자를 정규표현식을 이용하여 했지만 중앙로 n가 등 특수한 경우가 존재하여 제거 x 하나하나 지워야겠음

```

강원도 소방용수시설 데이터(alarm)에는 읍면동 데이터가 없음.

소재지번호주소
강원도 춘천시 남산면 창촌리 145-1
강원도 춘천시 남산면 창촌리 147-1
강원도 춘천시 남산면 창촌리 126-9
강원도 춘천시 남산면 창촌리 219-15
강원도 춘천시 남산면 창촌리 238-2
강원도 춘천시 남산면 창촌리 300-15
강원도 춘천시 남산면 창촌리 300-11

소재지번호의 7번째 이후의 단어를 'EMD_NM' 로 만들어서 저장

EMD_NM
남산면 창촌리 145-1
남산면 창촌리 147-1
남산면 창촌리 126-9
남산면 창촌리 219-15
남산면 창촌리 238-2

다음과 같은 열이 생성됨

공백을 기준으로 분할을 하고 dong으로 저장

	0	1	2	3
0	남산면	창촌리	145-1	None
1	남산면	창촌리	147-1	None
2	남산면	창촌리	126-9	None
3	남산면	창촌리	219-15	None
4	남산면	창촌리	238-2	None
...

dong의 0이 읍면동이므로 dong[0]을 EMD_NM으로 저장

일부 동이 띄어쓰기가 되어있지않아 동-주소형태로 표현이 되는 행 발견
 특수문자 제거후 정규표현식을 이용하여 숫자를 제거하려 했지만 교1동 같은 데이
 터가 존재하여 찾은 후 변경

```
# 전처리시 문제가 있던행을 변경
alarm['EMD_NM'] = alarm['EMD_NM'].str.replace('죽현동93317', '죽현동')
alarm['EMD_NM'] = alarm['EMD_NM'].str.replace('교동18991', '교동')
alarm['EMD_NM'] = alarm['EMD_NM'].str.replace('금학동73', '금학동')
alarm['EMD_NM'] = alarm['EMD_NM'].str.replace('성남동1016', '성남동')
alarm['EMD_NM'] = alarm['EMD_NM'].str.replace('성남동1114', '성남동')
alarm['EMD_NM'] = alarm['EMD_NM'].str.replace('성남동11310', '성남동')
alarm['EMD_NM'] = alarm['EMD_NM'].str.replace('발한동3099', '발한동')
alarm['EMD_NM'] = alarm['EMD_NM'].str.replace('이로동1021', '이로동')
alarm['EMD_NM'] = alarm['EMD_NM'].str.replace('이로동194', '이로동')
alarm['EMD_NM'] = alarm['EMD_NM'].str.replace('구미동4821', '구미동')
alarm['EMD_NM'] = alarm['EMD_NM'].str.replace('단구동1567', '단구동')
alarm['EMD_NM'] = alarm['EMD_NM'].str.replace('단구동15797', '단구동')
alarm['EMD_NM'] = alarm['EMD_NM'].str.replace('단구동15881', '단구동')
alarm['EMD_NM'] = alarm['EMD_NM'].str.replace('태장동1293', '태장동')
alarm['EMD_NM'] = alarm['EMD_NM'].str.replace('대포동957', '대포동')
alarm['EMD_NM'] = alarm['EMD_NM'].str.replace('포남동109721', '포남동')
alarm['EMD_NM'] = alarm['EMD_NM'].str.replace('발한동238', '발한동')
alarm['EMD_NM'] = alarm['EMD_NM'].str.replace('목호진동2447', '목호진동')
alarm['EMD_NM'] = alarm['EMD_NM'].str.replace('단구동15681', '단구동')
alarm['EMD_NM'] = alarm['EMD_NM'].str.replace('북평동128', '북평동')
```

Python

전처리시 문제가 되는 행을 변경

```
# 필요없는 데이터 columns 제거
del alarm['시설유형코드']
del alarm['시설번호']
del alarm['시군구코드']
del alarm['데이터기증일자']
```

Python

데이터에서 필요없다고 생각되는 열 제거

```
# 시군구명으로 해야함 => 중복되는 게 있음
alarm['Full_NM'] = alarm['시군구명'] + ' ' + alarm['EMD_NM']

df['Full_NM'] = df['SIGUNGU_NM'] + ' ' + df['EMD_NM']
```

Python

두 데이터에 ['Full_NM']이라고 새로운 열 생성

=> EMD_NM으로 병합할시, 중복되는 행이 있어서 불가능함 ex) 춘천시 교동, 강
 릉시 교동

2.3. 파생변수생성 및 전처리

```
# 개수를 데이터프레임으로 변환
water = pd.DataFrame(alarm['Full_NM'].value_counts())
fire = pd.DataFrame(df['Full_NM'].value_counts())

# 인덱스를 초기화
water = water.reset_index()
fire = fire.reset_index()

# columns 이름 변경
water.rename(columns={'Full_NM': '용수시설개수'}, inplace=True)
fire.rename(columns={'Full_NM': '화재발생횟수'}, inplace=True)

# 공통된 행 합치기(index 기준으로)
dat = pd.merge(water, fire, on = 'index')

dat.rename(columns={'index': 'Full_NM'}, inplace=True)
```

dat

✓ 0.8s

Python

각 읍면동별 용수시설의 개수와, 화재발생 횟수데이터를 groupby를 이용하여 생성
그 후 읍면동을 기준으로 공통된 행을 합쳐 하나의 데이터(dat)를 지정
결과

	Full_NM	용수시설개수	화재발생횟수
0	홍천군 홍천읍	183	63
1	횡성군 횡성읍	141	62
2	영월군 영월읍	137	54
3	원주시 지정면	127	60
4	태백시 황지동	112	49
...

읍면동별 용수시설개수와 화재발생횟수 데이터

```
# 위에서 결측값이 있는데 제거했더니 도착시간이 안나와서 새로 load
df = pd.read_csv('./data/구조 종별 신고, 재난 데이터.csv', encoding= 'utf-8')

# 데이터 화재관련 값만 추출

df = df[(df['ACDNT_CAUSE'] == '기타화재') | (df['ACDNT_CAUSE'] == '화약류') | (df['ACDNT_CAUSE'] == '방화')
        | (df['ACDNT_CAUSE'] == '폭발물') | (df['ACDNT_CAUSE'] == '화재확인')]

df['Full_NM'] = df['SIGUNGU_NM'] + ' ' + df['EMD_NM']

# 시간*60 + 분으로 해서 하나의 데이터로 만들었음
s_time = (df['DSP_HOUR']*60) + (df['DSP_MIN'])
a_time = (df['SPT_ARVL_HOUR']*60) + (df['SPT_ARVL_MIN'])

# 도착시간에 결측치가 있어서 0분으로 처리
a_time = a_time.fillna(s_time)

# 걸린시간
time = a_time - s_time

time = pd.DataFrame(time)
df['time'] = time
```

시간이라는 변수를 만들기 위해 구조 종별 신고, 재난 데이터에서 시간의 항목 추출

시간과 분의 단위가 달라서 하나의 단위인 ‘분’ 으로 통일

=> 시간*60 + 분

도착시간 - 출발시간 = 걸린시간

이라는 파생변수 생성후 time이라는 변수에 지정

```
# EMD_NM 별 화재도착시간 만들기
# 평균화재도착시간으로 하려했는데 23시출발 - 0시 도착이면 값이 이상해져서 중앙값으로 하였음.
time = pd.DataFrame(df.groupby(df['Full_NM'])['time'].median())
time = time.reset_index()
time
```

읍면동별 화재도착시간 생성

평균 화재도착시간으로 할시 23시 출발 0시도착의 경우 값의 문제가 생겨 중앙값

으로 대체

	Full_NM	time
0	강릉시 강동면	13.0
1	강릉시 강문동	5.0
2	강릉시 견소동	6.0
3	강릉시 교동	5.0
4	강릉시 구정면	8.0
...

시간 변수 데이터

```
dat = pd.merge(dat, time, on='Full_NM')
dat
```

✓ 0.7s Python

시간데이터와 기존 데이터를 읍면동기준으로 병합

```
# 풍속 (결측치가 너무많아서 일단 전에 행으로 대체)
df['TIME_UNIT_WS'] = df['TIME_UNIT_WS'].fillna(method='ffill')

# groupby를 이용하여 읍면동별 평균값
TIME_UNIT_WS = pd.DataFrame(df.groupby(df['Full_NM'])['TIME_UNIT_WS'].mean())
TIME_UNIT_WS = TIME_UNIT_WS.reset_index()
TIME_UNIT_WS

# 풍향 결측치를 이전행으로 대체(풍속과 같은이유)
df['TIME_UNIT_WD'] = df['TIME_UNIT_WD'].fillna(method='ffill')
TIME_UNIT_WD = pd.DataFrame(df.groupby(df['Full_NM'])['TIME_UNIT_WD'].mean())
TIME_UNIT_WD = TIME_UNIT_WD.reset_index()
TIME_UNIT_WD

dat = pd.merge(dat, TIME_UNIT_WS, on='Full_NM')
dat = pd.merge(dat, TIME_UNIT_WD, on='Full_NM')
dat
```

✓ 0.7s Python

풍속데이터 추가 읍면동별 평균값으로 groupby

```
fire = pd.read_csv('./data/화재 종별 신고,재난 데이터.csv', encoding='utf-8')
fire.head()

fire['Full_NM'] = fire['SIGUNGU_NM'] + ' ' + fire['EMD_NM']

# 화재발생위치로부터의 거리
dist = pd.DataFrame(fire.groupby(fire['Full_NM'])['SPT_FRSTT_DIST'].mean())
dist = dist.reset_index()
dist
dat = pd.merge(dat, dist, on='Full_NM')
dat
```

✓ 0.1s Python

화재 발생위치로 부터의 거리 파생변수 추가
읍면동별 소방서부터의 거리로 dist라는 파생변수로 지정
그 후, 기존의 데이터와 읍면동기준으로 병합

```

#배관깊이
depth = pd.DataFrame(alarm.groupby(alarm['Full_NM'])['배관깊이'].mean())
depth = depth.reset_index()
depth

#출수압력
pressure = pd.DataFrame(alarm.groupby(alarm['Full_NM'])['출수압력'].mean())
pressure = pressure.reset_index()
pressure

#배관지름
diameter = pd.DataFrame(alarm.groupby(alarm['Full_NM'])['배관지름'].mean())
diameter = diameter.reset_index()
diameter

# 데이터간 합
dat = pd.merge(dat, depth, on='Full_NM')
dat = pd.merge(dat, pressure, on='Full_NM')
dat = pd.merge(dat, diameter, on='Full_NM')
dat

dat['EMD_NM'] = dat['Full_NM'].str[4:]
dat['SGG_NM'] = dat['Full_NM'].str[:3]

```

✓ 0.7s Python

배관관련 파생변수 추가

```

dat['EMD_NM'] = dat['Full_NM'].str[4:]
dat['SGG_NM'] = dat['Full_NM'].str[:3]

```

✓ 0.6s

```

dat.head()

```

✓ 0.1s

	Full_NM	용수시설개수	화재발생횟수	time	TIME_UNIT_WS	TIME_UNIT_WD	SPT_FRSTT_DIST	배관깊이	출수압력	배관지름	EMD_NM	SGG_NM
0	홍천군 홍천읍	183	63	5.0	1.444444	152.857143	4.380952	0.997268	4.621311	100.0	홍천읍	홍천군
1	횡성군 횡성읍	141	62	5.0	2.290323	180.645161	4.800000	1.007092	5.184397	65.0	횡성읍	횡성군
2	영월군 영월읍	137	54	5.0	1.333333	143.518519	4.000000	0.936715	4.670803	100.0	영월읍	영월군
3	원주시 지정면	127	60	10.0	1.583333	177.333333	21.612903	0.675591	5.541732	100.0	지정면	원주시
4	태백시 황지동	112	49	4.0	1.387755	115.714286	2.219512	0.716161	4.650000	65.0	황지동	태백시

병합된 데이터에 읍면동과 시군구 추가 (나중에 필요함)

```

people = pd.read_csv('./data/강원도_읍면동별_주민등록인구.csv', encoding='cp949')
people.head()

```

✓ 0.1s Python

	시군	행정구역(동읍면)별	5세별	총인구수 (명)	남자인구수 (명)	여자인구수 (명)
0	춘천시	신북읍	계	7503	3763	3740
1	춘천시	동면	계	19415	9571	9844
2	춘천시	동산면	계	1451	781	670
3	춘천시	신동면	계	2604	1351	1253
4	춘천시	남면	계	1081	588	493

인구관련 파생변수 추가를 위하여 강원도 읍면동별 주민등록인구 데이터 로드 (데이터관련 문제가 있어서 엑셀로 시군항목추가하였음)

```

del people['5세별']
del people['남자인구수 (명)']
del people['여자인구수 (명)']

```

✓ 0.6s Python

데이터에서 필요없다고 생각되는 열 제거

```

people.rename(columns={'행정구역(동읍면)별':'EMD_NM'}, inplace=True)
people.rename(columns={'시군':'SGG_NM'}, inplace=True)

people['Full_NM'] = people['SGG_NM'] + ' ' + people['EMD_NM']
people.head()

```

✓ 0.1s Python

	SGG_NM	EMD_NM	5세별	총인구수 (명)	남자인구수 (명)	여자인구수 (명)	Full_NM
0	춘천시	신북읍	계	7503	3763	3740	춘천시 신북읍
1	춘천시	동면	계	19415	9571	9844	춘천시 동면
2	춘천시	동산면	계	1451	781	670	춘천시 동산면
3	춘천시	신통면	계	2604	1351	1253	춘천시 신통면
4	춘천시	남면	계	1081	588	493	춘천시 남면

통합을 위하여 열이를 변경

EMD_NM, SGG_NM항목 추가

두 개를 합친 Full_NM추가

```

del people['SGG_NM']
del people['EMD_NM']

dat = pd.merge(dat, people, on='Full_NM')
dat.head()

```

✓ 0.5s Python

```

dat = pd.merge(dat, people, on='Full_NM')
dat.head()

```

✓ 0.8s Python

	Full_NM	용수시설개수	화재발생횟수	time	TIME_UNIT_WS	TIME_UNIT_WD	SPT_FRSTT_DIST	배관길이	출수압력	배관지름	EMD_NM	SGG_NM	총인구수 (명)
0	홍천군 홍천읍	183	63	5.0	1.444444	152.857143	4.380952	0.997268	4.621311	100.0	홍천읍	홍천군	33754
1	횡성군 횡성읍	141	62	5.0	2.290323	180.645161	4.800000	1.007092	5.184397	65.0	횡성읍	횡성군	20976
2	영월군 영월읍	137	54	5.0	1.333333	143.518519	4.000000	0.936715	4.670803	100.0	영월읍	영월군	20883
3	원주시 지정면	127	60	10.0	1.583333	177.333333	21.612903	0.675591	5.541732	100.0	지정면	원주시	29115
4	태백시 황지동	112	49	4.0	1.387755	115.714286	2.219512	0.716161	4.650000	65.0	황지동	태백시	5970

dat과 Full_NM을 기준으로 병합

```

# 전처리된 데이터를 tidy_data로 저장
dat.to_csv('tidy_data.csv', encoding='cp949', index=False)

```

✓ 0.6s

전처리된 데이터를 tidy_data로 저장

2.4. 데이터 스케일링

```

# 전처리된 데이터를 tidy_data로 저장
dat.to_csv('tidy_data.csv', encoding='cp949', index=False)

```

✓ 0.6s

```

dat = pd.read_csv('tidy_data.csv', encoding='cp949')
dat.head()

```

✓ 0.7s Python

```

#데이터 스케일링을 위해 수치형, 범주형 데이터 분리
cat_dat = dat[['Full_NM', 'EMD_NM', 'SGG_NM']]
cat_dat.head()

```

✓ 0.4s Python

	Full_NM	EMD_NM	SGG_NM
0	홍천군 홍천읍	홍천읍	홍천군
1	횡성군 횡성읍	횡성읍	횡성군
2	영월군 영월읍	영월읍	영월군
3	원주시 지정면	지정면	원주시
4	태백시 황지동	황지동	태백시

```
#데이터 스케일링을 위해 수치형, 범주형 데이터 분리
num_dat = dat.drop(columns=['Full_NM', 'EMD_NM', 'SGG_NM'])
num_dat.head()
```

	용수시설개수	화재발생횟수	time	TIME_UNIT_WS	TIME_UNIT_WD	SPT_FRSTT_DIST	배관깊이	출수압력	배관지름	총인구수 (명)
0	183	63	5.0	1.444444	152.857143	4.380952	0.997268	4.621311	100.0	33754
1	141	62	5.0	2.290323	180.645161	4.800000	1.007092	5.184397	65.0	20976
2	137	54	5.0	1.333333	143.518519	4.000000	0.936715	4.670803	100.0	20883
3	127	60	10.0	1.583333	177.333333	21.612903	0.675591	5.541732	100.0	29115
4	112	49	4.0	1.387755	115.714286	2.219512	0.716161	4.650000	65.0	5970

데이터를 점수화 하기위하여 스케일링 사전 작업진행
 스케일링시 문자형데이터는 불가능하기 때문에 수치형, 범주형으로 분할

```
# 두개의 열을 새로운 계산을 위하여 따로 처리
data_count = dat[['용수시설개수', '화재발생횟수']]
data_count.head()
```

	용수시설개수	화재발생횟수
0	183	63
1	141	62
2	137	54
3	127	60
4	112	49

```
# 나눗셈을 하기위해 최소값-1 값을 추가
data_count.loc[len(data_count)] = [0,0]
```

‘화재예방지수’ 과생변수를 생성하기 위해선 화재대비 용수시설개수가 필요함
 => 화재 예방지수를 만들기 위해선 나눗셈을 해야하는데 MinMaxScaler를 사용하면 0인값이 존재 분모가 0이되는 행이 존재해 inf값이 나올 수 있으므로 최소값보다 작은 임의값을 추가하여 0이되는 것을 방지

2.4.1 MinMaxScaler

```
# MinMaxScaler
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
scaler.fit(num_dat)
df_norm = scaler.transform(num_dat)
df_norm = pd.DataFrame(df_norm, columns=num_dat.columns)
df_norm.head()
```

	용수시설개수	화재발생횟수	time	TIME_UNIT_WS	TIME_UNIT_WD	SPT_FRSTT_DIST	배관깊이	출수압력	배관지름	총인구수 (명)
0	1.000000	1.000000	0.142857	0.140351	0.394892	0.043193	0.138530	0.472174	1.0	0.695115
1	0.769231	0.983871	0.142857	0.407470	0.535015	0.049475	0.140482	0.619892	0.0	0.426308
2	0.747253	0.854839	0.142857	0.105263	0.347801	0.037481	0.126500	0.485158	1.0	0.424352
3	0.692308	0.951613	0.500000	0.184211	0.518315	0.301543	0.074621	0.713634	1.0	0.597526
4	0.609890	0.774194	0.071429	0.122449	0.207597	0.010787	0.082681	0.479700	0.0	0.110632

MinMaxScaler를 이용하여 데이터를 0~1사이값으로 변환
 데이터간의 크기차이가 존재해 점수의 차이가 심할 수 있으므로 스케일링후 점수화


```
# 용수시설개수와 화재발생횟수의 정규화. 마지막 행 제거
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
scaler.fit(data_count)
df_count = scaler.transform(data_count)
df_count = pd.DataFrame(df_count, columns=data_count.columns)
df_count = df_count.drop(143, axis=0)
df_count.head()
```

	용수시설개수	화재발생횟수
0	1.000000	1.000000
1	0.770492	0.984127
2	0.748634	0.857143
3	0.693989	0.952381
4	0.612022	0.777778

화재예방지수를 만들기위해 생성한 마지막행(임의의 값) 제거

```
df_norm['용수시설개수'] = df_count['용수시설개수']
df_norm['화재발생횟수'] = df_count['화재발생횟수']
```

따로 정규화 했던 용수시설개수와 화재발생횟수를 기존의 정규화한 것으로 변환

```
# 시간과 거리의 개념은 반대되는 개념이므로 -1 후에 음수로 변환
# 인구또한 적을수록 화재위험도가 낮으므로 -1 후에 음수로 변환

df['time'] = df['time'] - 1
df['SPT_FRSTT_DIST'] = df['SPT_FRSTT_DIST'] - 1
df['총인구수 (명)'] = df['총인구수 (명)'] - 1

df['time'] = -df['time']
df['SPT_FRSTT_DIST'] = -df['SPT_FRSTT_DIST']
df['총인구수 (명)'] = -df['총인구수 (명)']
```

시간과 거리, 인구는 값이 클수록 좋은 것이 아님

=> 그러므로 0~1사이 값에서 -1을 더한 후에 다시 음수를 곱하여 0이 가장 큰값으로 되게 변환

3. 분석

3.1. 파생변수 생성

```
df['화재예방지수'] = (df['용수시설개수'] / df['화재발생횟수'])

df.describe()
```

	용수시설개수	화재발생횟수	time	TIME_UNIT_WS	TIME_UNIT_WD	SPT_FRSTT_DIST	배관길이	출수압력	배관지름	총인구수 (명)	화재예방지수
count	143.000000	143.000000	1.430000e+02	143.000000	143.000000	143.000000	143.000000	143.000000	143.000000	143.000000	143.000000
mean	0.215102	0.355977	6.743257e-01	0.270859	0.509030	0.771018	0.162064	0.389376	0.806404	0.859308	0.675261
std	0.170640	0.222545	2.130029e-01	0.203772	0.188938	0.190974	0.109886	0.202940	0.364155	0.181084	0.501058
min	0.005464	0.015873	1.110223e-16	0.000000	0.000000	-0.000000	0.000000	0.000000	0.000000	-0.000000	0.031297
25%	0.101093	0.190476	5.714286e-01	0.121953	0.395252	0.671869	0.120112	0.240961	0.922908	0.839932	0.398770
50%	0.163934	0.301587	7.142857e-01	0.231579	0.506549	0.809352	0.140315	0.359706	1.000000	0.931462	0.602459
75%	0.278689	0.507937	8.571429e-01	0.363597	0.614604	0.929350	0.178808	0.488918	1.000000	0.961871	0.817623
max	1.000000	1.000000	1.000000e+00	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	4.810672

```
df['화재예방지수'] = (df['화재발생횟수'] + (df['time']) + (df['SPT_FRSTT_DIST']))
df.describe()
```

✓ 0.9s Python

	용수시설개수	화재발생횟수	time	TIME_UNIT_WS	TIME_UNIT_WD	SPT_FRSTT_DIST	배관깊이	출수압력	배관지름	총인구수 (명)	화재예방지수
count	143.000000	143.000000	1.430000e+02	143.000000	143.000000	143.000000	143.000000	143.000000	143.000000	143.000000	143.000000
mean	0.215102	0.355977	6.743257e-01	0.270859	0.509030	0.771018	0.162064	0.389376	0.806404	0.859308	2.120604
std	0.170640	0.222545	2.130029e-01	0.203772	0.188938	0.190974	0.109886	0.202940	0.364155	0.181084	0.598446
min	0.005464	0.015873	1.110223e-16	0.000000	0.000000	-0.000000	0.000000	0.000000	0.000000	-0.000000	0.766888
25%	0.101093	0.190476	5.714286e-01	0.121953	0.395252	0.671869	0.120112	0.240961	0.922908	0.839932	1.740146
50%	0.163934	0.301587	7.142857e-01	0.231579	0.506549	0.809352	0.140315	0.359706	1.000000	0.931462	2.069218
75%	0.278689	0.507937	8.571429e-01	0.363597	0.614604	0.929350	0.178808	0.488918	1.000000	0.961871	2.519086
max	1.000000	1.000000	1.000000e+00	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	5.723078

‘화재예방지수’ 파생변수 생성
화재예방지수 = (용수시설개수 / 화재발생횟수) + 시간 + 거리

```
df['용수시설지수'] = df['출수압력'] + df['배관지름']
```

✓ 0.6s Python

‘용수시설지수’ 파생변수 생성
용수시설지수 = 출수압력 + 배관지름

```
final_all = pd.concat([cat_dat, final_num], axis=1)
```

✓ 0.5s Python

```
final_all.head()
```

✓ 0.3s Python

	Full_NM	EMD_NM	SGG_NM	화재예방지수	용수시설지수	총인구수 (명)
0	홍천군 홍천읍	홍천읍	홍천군	2.813950	1.472174	0.304885
1	횡성군 횡성읍	횡성읍	횡성군	2.590587	0.619892	0.573692
2	영월군 영월읍	영월읍	영월군	2.693068	1.485158	0.575648
3	원주시 지정면	지정면	원주시	1.927146	1.713634	0.402474
4	태백시 황지동	황지동	태백시	2.704669	0.479700	0.889368

final_all이라는 변수에 수치형 변수와 범주형 변수를 합쳐 읍면동별 점수를 생성

```
final_num['Final Score'] = final_num['화재예방지수'] + final_num['용수시설지수'] + final_num['총인구수 (명)']
final_num
```

✓ 0.1s Python

	화재예방지수	용수시설지수	총인구수 (명)	Final Score
0	2.813950	1.472174	0.304885	4.591009
1	2.590587	0.619892	0.573692	3.784170
2	2.693068	1.485158	0.575648	4.753873
3	1.927146	1.713634	0.402474	4.043253
4	2.704669	0.479700	0.889368	4.073738
...

‘Final Score’ 라는 파생변수 생성
Final Score = 모든변수의 합

3.1.2. 최종 입지 결정

```
final_data = pd.concat([cat_dat, final_num], axis = 1)

del final_data['화재예방지수']
del final_data['용수시설지수']
del final_data['종민구수 (명)']
```

✓ 0.5s Python

```
final_data.sort_values(by='Final Score').head(5)
```

✓ 0.6s Python

	Full_NM	EMD_NM	SGG_NM	Final Score
142	삼척시 하장면	하장면	삼척시	2.305948
40	횡성군 둔내면	둔내면	횡성군	2.741306
93	정선군 남면	남면	정선군	2.877478
65	영월군 주천면	주천면	영월군	2.917247
138	강릉시 왕산면	왕산면	강릉시	3.112850

Final score를 제외한 수치형변수 제거

final score를 내림차순으로 정렬하여 최종적 입지선정

삼척군 하장면, 횡성군 둔내면, 정선군 남면, 영월군 주천면, 강릉시 왕산면

```
df_l = df[(df['Full_NM'] == '삼척시 하장면') | (df['Full_NM'] == '횡성군 둔내면') | (df['Full_NM'] == '정선군 남면')
| (df['Full_NM'] == '영월군 주천면') | (df['Full_NM'] == '강릉시 왕산면')]

qgis = df_l[['Full_NM', 'SIGUNGU_NM', 'EMD_NM', 'LOC_INFO_X', 'LOC_INFO_Y']]
qgis.dropna(axis=0)
# qgis.to_csv('qgisdata.csv', encoding='cp949')
```

✓ 0.1s Python

	Full_NM	SIGUNGU_NM	EMD_NM	LOC_INFO_X	LOC_INFO_Y
5	삼척시 하장면	삼척시	하장면	128.9242	37.3575
127	횡성군 둔내면	횡성군	둔내면	128.2436	37.5006
138	횡성군 둔내면	횡성군	둔내면	128.2086	37.5125
161	횡성군 둔내면	횡성군	둔내면	128.1865	37.4764
226	영월군 주천면	영월군	주천면	128.2699	37.2253
231	삼척시 하장면	삼척시	하장면	128.8698	37.3925

```
alarm1 = alarm[(alarm['Full_NM'] == '삼척시 하장면') | (alarm['Full_NM'] == '횡성군 둔내면') | (alarm['Full_NM'] == '정선군 남면')
| (alarm['Full_NM'] == '영월군 주천면') | (alarm['Full_NM'] == '강릉시 왕산면')]

qgis = alarm1[['Full_NM', 'EMD_NM', '위도', '경도']]
qgis.dropna(axis=0)
# qgis.to_csv('qgisdata_alarm.csv', encoding='cp949')
```

✓ 0.7s Python

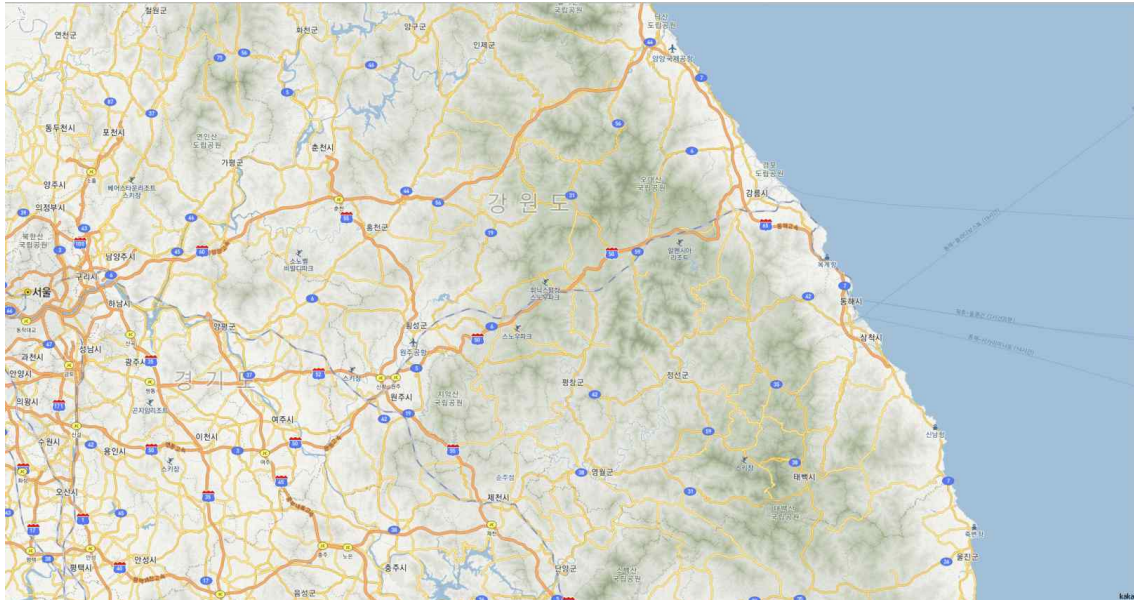
	Full_NM	EMD_NM	위도	경도
1433	횡성군 둔내면	둔내면	37.519203	128.201199
1434	횡성군 둔내면	둔내면	37.512914	128.214327
1435	횡성군 둔내면	둔내면	37.514725	128.216582
1436	횡성군 둔내면	둔내면	37.513427	128.217923
1437	횡성군 둔내면	둔내면	37.512626	128.216204
...

qgis에 최종입지분석을 넣기위해 입지선정 top5의 용수시설좌표와 화재발생위치 좌표를 생성

4. 시각화 및 결과 해석

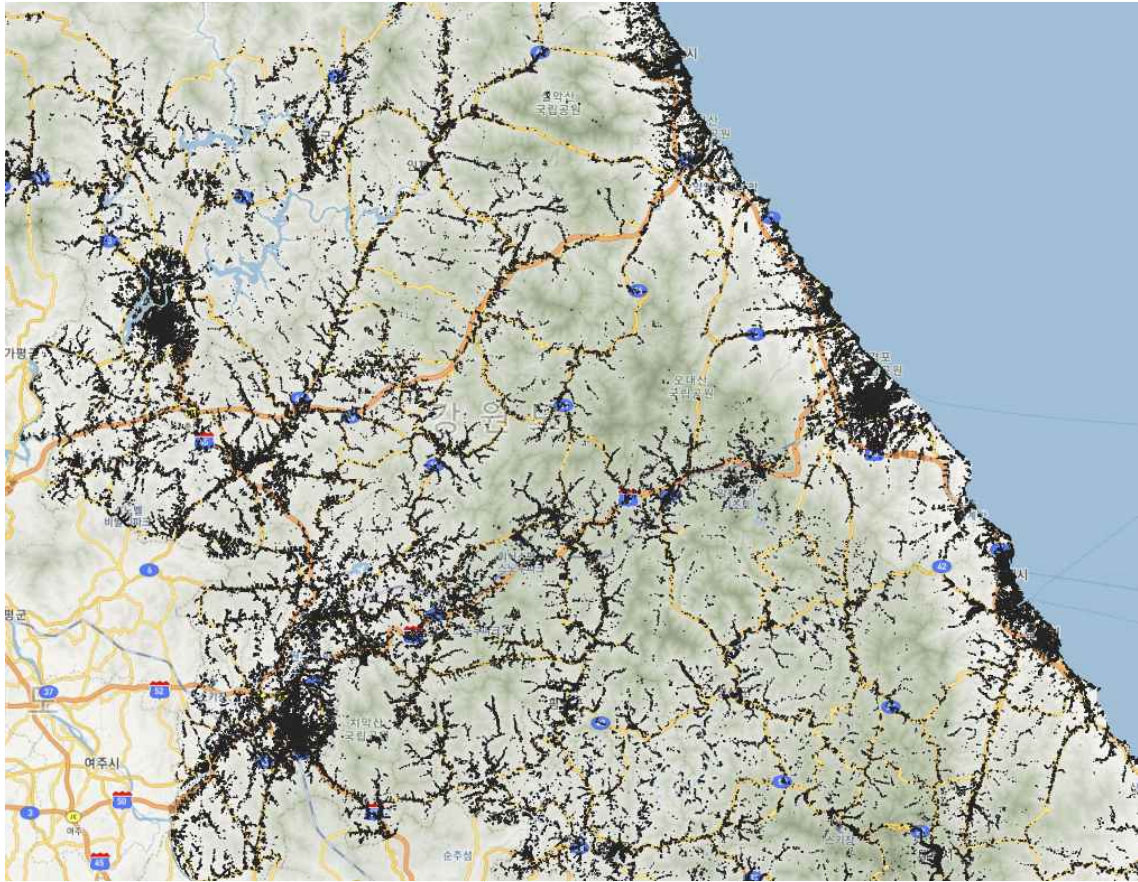
4.1. 시각화

QGIS를 이용한 시각화



QGIS의 Kakao maps를 Qgis에 로드

4.1.1. 노후건물 파악



국가공간정보포털에서 강원도 건물통합정보 데이터를 가져옴
=> 노후건물은 화재시 취약하기 때문에 노후건물데이터를 통해 입지선정

둔내면 — 필드 계산기

☐ 선택한 객체 0 개만 업데이트

☒ 새로운 필드 생성 ☐ 기존 필드를 갱신

☐ 가상 필드 생성

산출 필드 이름: YEAR

산출 필드 유형: 정수 (정수형)

산출 필드 길이: 10 정밀도: 3

표현식 함수 편집기

left("USEAPR_DAY", 4)

= + - / * ^ || () Wr

객체: 5300000000

미리보기: NULL

row_number

- 날짜와 시간
- 도형
- 래스터
- 레코드와 속.
- 맵
- 문자열
- 배열
- 변수
- 변환
- 색상
- 수학

이 레이어 상에서 정보를 편집하고 있지만 레이어가 현재 편집 모드가 아닙니다. '확인'을 클릭하면 자동적으로 편집 모드로 전환될 것입니다.

확인 취소 도움말

노후건물만 추출하기 위해서 ‘건물통합정보’에서 승인일자(8자리)에서 연도에 해당하는 앞 4자리를 가져와서 ‘YEAR’라는 변수로 지정

노후건물 선정기준

① 영 제2조제3항제1호에 따라 노후·불량건축물로 보는 기준은 다음 각 호와 같다.

가. 철근콘크리트·철골콘크리트·철골철근콘크리트 및 강구조인 공동주택: 별표 1에 따른 기간

나. 가목 이외의 공동주택: 20년

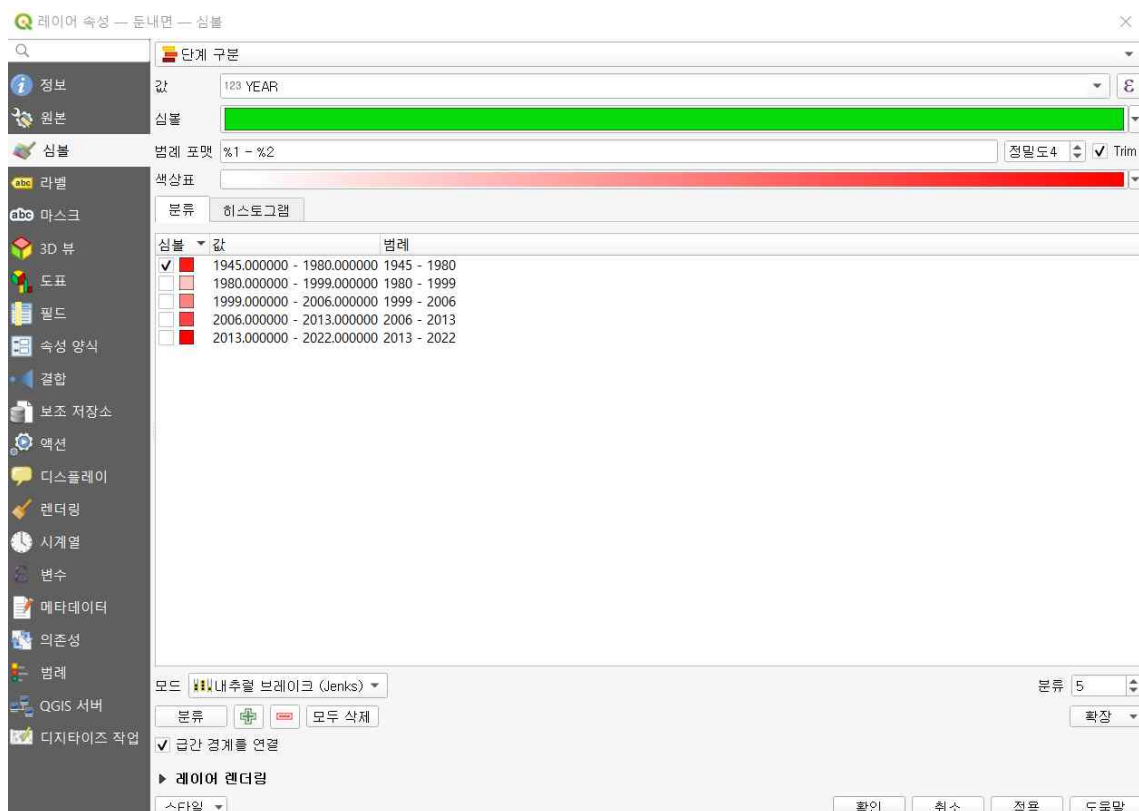
가. 철근콘크리트·철골콘크리트·철골철근콘크리트 및 강구조 건축물(「건축법 시행령」 별표 1 제1호에 따른 단독주택을 제외한다): 30년

② 영 제2조제2항제1호에 따른 노후·불량건축물은 건축대지로서 효용을 다할 수 없는 과소필지 안의 건축물로서 2009년 8월 11일 전에 건축된 건축물을 말한다.

9. "과소필지"란 토지면적이 90제곱미터 미만인 토지를 말한다.

노후건물은 가목 이외의 공통주택은 20년, 철근콘크리트 건축물은 30년으로 20~30년이 넘는 건물을 보통 노후건물이라고 지정
그러므로 30년을 기준으로 보고 ~ 1980년까지의 데이터를 노후 건축물이라고 판단하였음

단계구분도



~ 1980년까지의 데이터만 지정하여 설정

4.1.2. 위치설정

	Full_NM	EMD_NM	SGG_NM	Final Score
142	삼척시 하장면	하장면	삼척시	2.305948
40	횡성군 둔내면	둔내면	횡성군	2.741306
93	정선군 남면	남면	정선군	2.877478
65	영월군 주천면	주천면	영월군	2.917247
138	강릉시 왕산면	왕산면	강릉시	3.112850

Final Score를 통해 상위5개의 강원도 읍면동에 최적입지선정

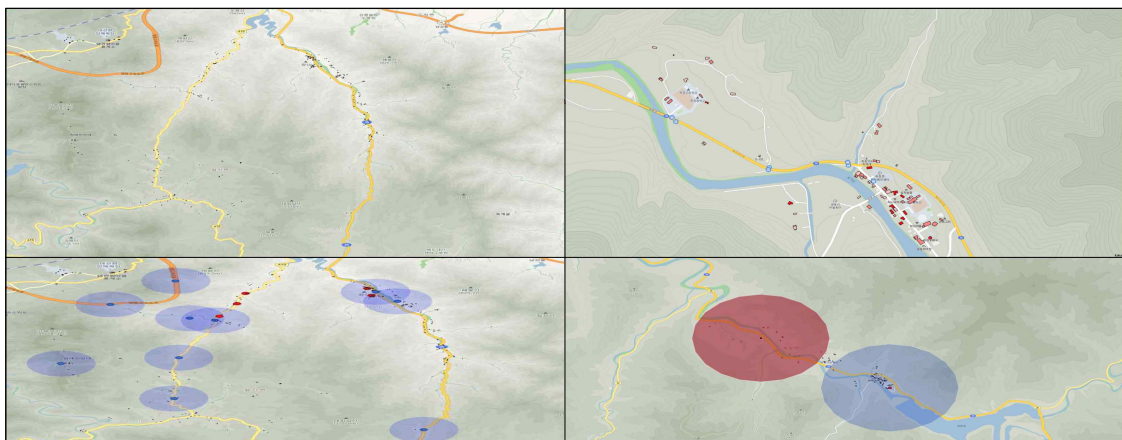
용수시설 설치기준

1. 공통기준

- 가. 국토의계획및이용에관한법률 제36조제1항제1호의 규정에 의한 주거지역·
상업지역 및 공업지역에 설치하는 경우 : 소방대상물과의 수평거리를 100
미터 이하가 되도록 할 것
- 나. 가목 외의 지역에 설치하는 경우 : 소방대상물과의 수평거리를 140미터
이하가 되도록 할 것

용수시설은 소방대상물과의 수평거리제한이 있음 이를 참고하여 기존 용수시설 버
퍼 생성

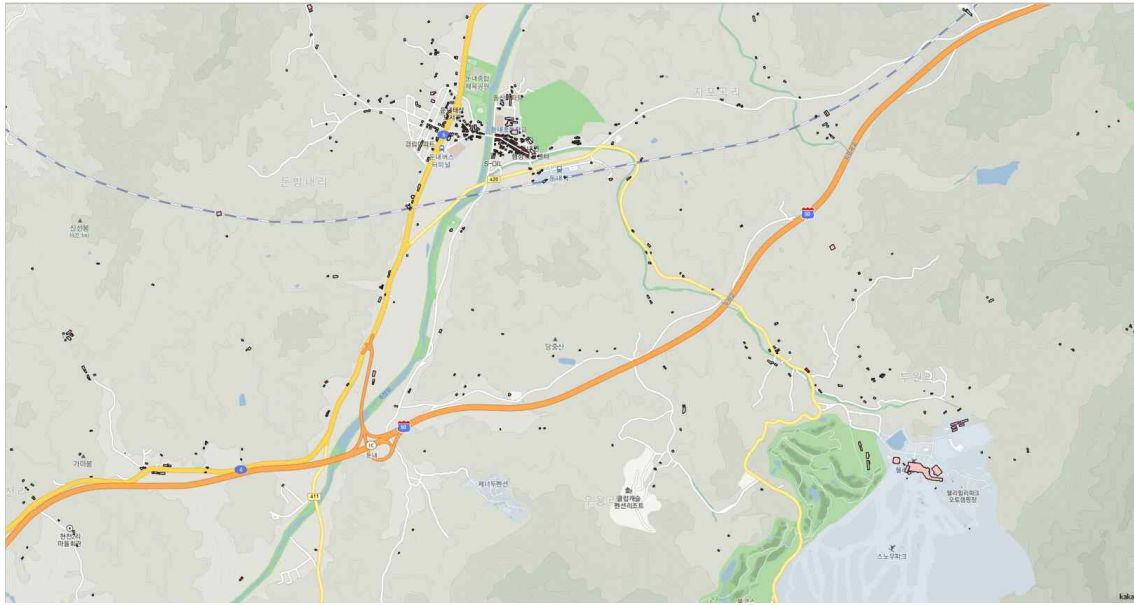
시각화 결과



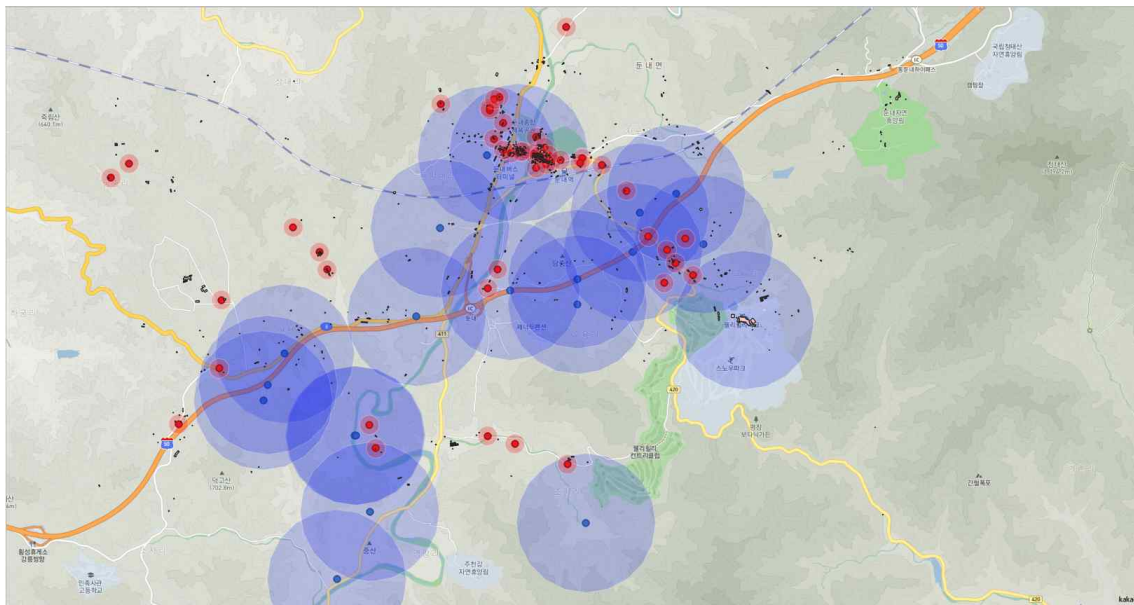


둔내면

둔내면 화재취약시설 위치



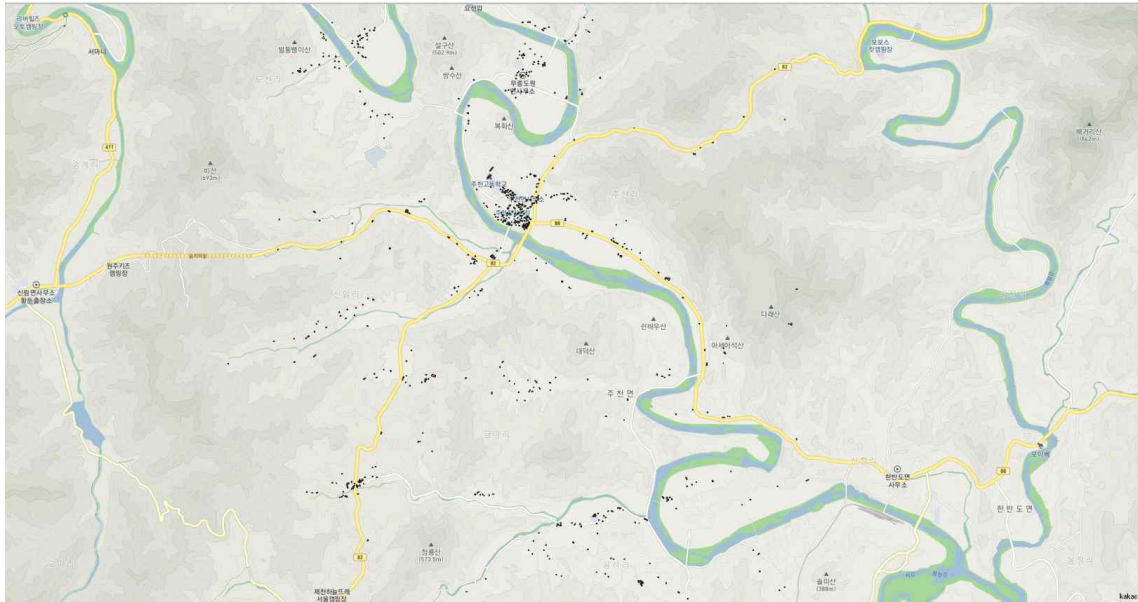
둔내면 화재발생위치와 용수시설



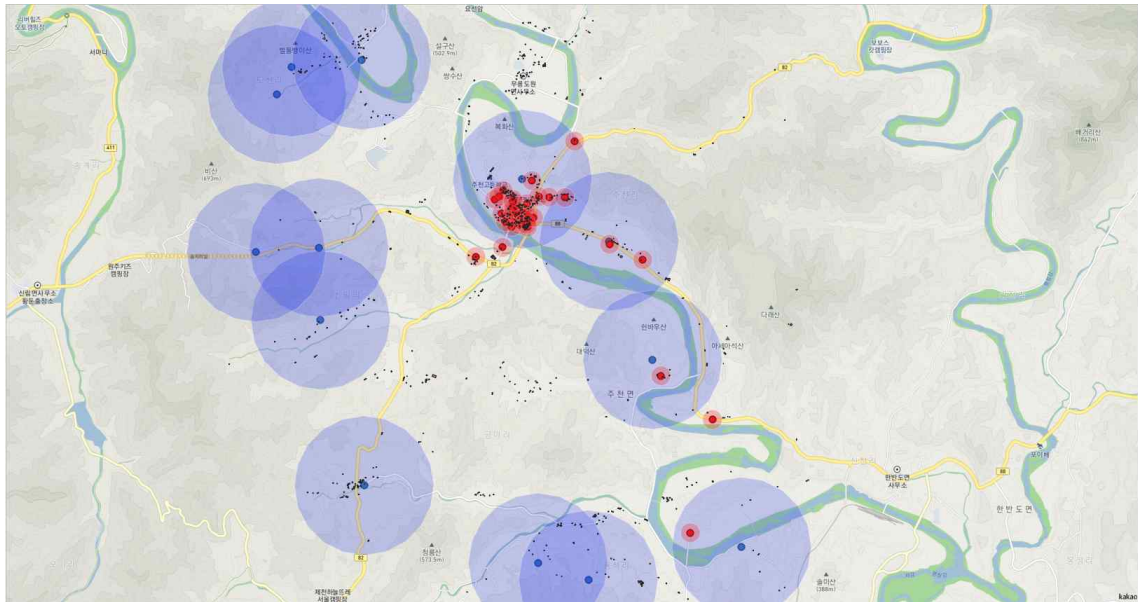
용수시설은 설치기준에 따라 최소값을 기준으로 버퍼를 생성하였음
화재반경과 용수시설반경안에 취약건물이 많은위치에 용수시설을 생성

주천면

주천면 화재취약시설 위치



주천면 화재발생위치와 용수시설



5. 결론 및 제언

5.1. 결론

소방용수시설의 최적 입지에 필요한 변수를 화재 발생 횟수, 용수시설, 풍속, 인구로 설정하고 ‘화재예방지수’, ‘용수시설’, ‘인구’ 세 개의 파생 변수를 생성하였다. 파생 변수 간의 합으로 ‘Final Score’를 생성하여 수치가 낮을수록 화재 피해가 높게 설정하였다.

5개의 입지 선정 결과 1. 삼척군 하장면, 2. 횡성군 둔내면, 3. 정선군 남면, 4. 영월군 주천면, 5. 강릉시 왕산면으로 나타났다.

설정된 읍면동의 화재 발생 위치와 용수시설 위치 버퍼를 생성하여 중복되지 않고, 노후 건물이 많은 곳에 용수시설을 설치하면 될 것이다.

5.2. 기대효과

[화재 예방]

소방용수시설은 화재가 발생했을 때 소방차에 물을 공급해 주는 용도로서 화재진압에 필수적인 시설로 부족할 시, 화재진압 시간이 지체될 수 있는데, 최적 입지를 선정함으로써 추가적인 설치로 인해 화재피해를 최소화할 수 있음.

[불필요한 인력 소모 방지]

더 많은 소방용수시설을 갖추므로, 인력이 부족한 소방센터에 인력의 효율적인 배분 가능

[전국적인 확산]

강원도에만 있는 데이터로 한 것이 아니고 전 지역 어디에서든 구할 수 있는 데이터로 분석을 진행했기 때문에, 전역적인 모델 구축 가능

5.3. 한계점

1. 화재 데이터 부족(1개년)으로 읍면동 단위 소방용수 시설이 적은 읍면동의 점수가 대체적으로 낮은 문제가 발생.
2. 데이터별로 기준이 달라, 병합 시 유실되는 데이터가 일부 있음.
3. 일부 데이터의 결측치가 있어 평균, 중앙값으로 일부 대체
4. Final Score의 당위성이 부족 (현실적으로 입지에 영향을 끼치는 모든 요인이 추출될 수 없으며 추출된다 하더라도 기술적으로 모두 고려할 수 없음.)
5. 노후 건물을 기준으로 용수시설의 입지를 선정하였는데, 화재취약시설은 노후 건물만 있는 것이 아님. 즉, 다양한 화재취약시설을 고려하지 못하였음
6. 추후, 소방용수시설 생성 시 해당 위치가 사유지라 만들지 못하는 문제가 있을 수 있음. (분석으로는 대략적인 위치만 파악 가능)

출처

- 자료조사

<https://fpn119.co.kr/176510> 소방방재신문 2022.04.20.

<https://easylaw.go.kr/CSP/CnpClsMainBtr.laf?popMenu=ov&csmSeq=1574&ccfNo=3&cciNo=1&cnpClsNo=2>

<https://www.edaily.co.kr/news/read?newsId=02945446625699384&mediaCodeNo=257&OutLnkChk=Y>

- 분석

- 공공데이터포털

<https://www.data.go.kr/>

- 국가공간정보포털

<http://www.nsdi.go.kr/lxportal/?menuno=2679>

- 소방안전 빅데이터 플랫폼

<https://www.bigdata-119.kr/goods/goodsList>