(용인시) 청년 창업 지원 대책

군집분석을 통한
청년 창업 입지 추천

COMPAS

CONTENTS

분석 목적

용인시 상권 현황 및 환경 요인들과의 상관관계 파악 및 지역 유형별 특징을 반영한 창업 지원 정책 제안

분석 프로세스

데이터 전처리
결측치 처리
격자 통합

⇨

EDA
사업체 입지 현황
청년 사업체 인사이트 도출

⇨

데이터 분석
핫스팟 분석
로지스틱 회귀 분석
EM알고리즘 기반 군집분석

⇨ 결론 및 제언 ⇨ 기대효과

분석 결과

용인시 상권 분할을 통해 최적의 입지를 추천하고 상권 확장 및 청년 창업 정책 활용방안 제시

01 공공 데이터 수집

제시된 데이터 외에도 상권에 영향을 미치는 교통 접근성, 교육, 의료 서비스, 문화 관련 등
시각화 및 분석에 활용될 데이터를 수집

교통접근성	교육	의료 서비스	문화
전철, 지하철 역	초등학교	대학병원	문화
버스정류장 위치	중학교	의원	랜드마크
	고등학교	한의원	주요관광지
	대학교	병원	

02 외부데이터 크롤링

공공 데이터의 경우, 위치에 대한 정보가 도로명 주소로 존재하여 Google Maps Geocoding API를 통해
위경도 변수를 추가하여 주소 데이터에서 위치 데이터로 변형

교통접근성	교육	의료 서비스	문화
전철, 지하철 역 버스정류장 위치	초등학교 중학교 고등학교 대학교	대학병원 의원 한의원 병원	문화 랜드마크 주요관광지

주소 데이터를 위도와 경도로 표현되는 위치 데이터로 변화

03 결측 데이터 보완

주어진 데이터 중, 1.용인시_상권_정보.csv 파일에 관해 행정동명 결측치 데이터 125개가 존재.
따라서 위경도의 정보를 통한 Google Map 크롤링으로 결측 행정동명을 파악하고 이 값을 보완.

```
df[df['행정동명'].isnull()==True]
```

	대분류 코드	중분류 코드	소분류 코드	표준산업분류 코드	표준산업분류명	행정동코드	행정동명	도로명주소	경도	위도
110	Q	Q01	Q01A99	I56111	한식 음식점업	4146131000	NaN	경기도 용인시 처인구 모현면 문현로 145	127.225779	37.343952
1149	Q	Q12	Q12A01	I56220	비알콜 음료점업	4146358500	NaN	경기도 용인시 기흥구 동백중앙로 358-8	127.160004	37.282058
2612	F	F03	F03A08	S95391	신발, 의복 및 기타 가정용 직물 제품 수리업	4146358500	NaN	경기도 용인시 기흥구 어정로 139	127.144714	37.275884
4765	F	F01	F01A01	S96112	두발미용업	4146131000	NaN	경기도 용인시 처인구 모현면 백옥대로 2332번길 15	127.248809	37.332080
5202	Q	Q12	Q12A01	I56220	비알콜 음료점업	4146131000	NaN	경기도 용인시 처인구 모현면 백옥대로 2366번길 10-25	127.250334	37.334162
...	...	...	...	...	...	...	...	...	...	...
39821	Q	Q06	Q06A01	I56114	서양식 음식점업	4146351500	NaN	경기도 용인시 기흥구 흥덕2로87번길 18	127.071578	37.275901
39825	R	R05	R05A02	P85620	예술 학원	4146351500	NaN	경기도 용인시 기흥구 흥덕중앙로105번길 41	127.076612	37.281013
39859	D	D14	D14A01	G47631	운동 및 경기용품 소매업	4146351500	NaN	경기도 용인시 기흥구 중부대로56번길 10	127.075146	37.266269
39875	Q	Q10	Q10A05	I56111	한식 음식점업	4146358500	NaN	경기도 용인시 기흥구 동백죽전대로 444	127.151591	37.277988
39902	L	L01	L01A01	L68221	부동산 자문 및 중개업	4146351500	NaN	경기도 용인시 기흥구 덕영대로 1981-1	127.098493	37.261763

125 rows × 10 columns

→

```
df['행정동명'].isnull().sum()
```

```
0
```

04 격자 통합

기존의 데이터들은 250*250 격자 형태도 있고, 100*100 격자도 존재함.
격자에 따른 공간의 차이를 없애기 위해 500*500격자로 통합하는 과정을 진행.

100*100 격자

총인구 데이터

고령인구 데이터

생산가능 인구 데이터

유소년 인구 데이터

건물연면적

250*250 격자

소상공인 매출정보 데이터

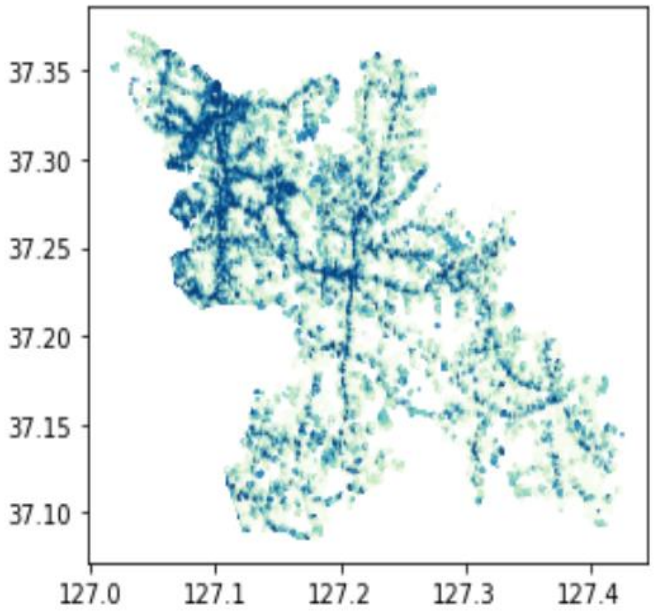
→

	gid500	all_cnt	old_cnt	adult_cnt	young_cnt	건물연면적	ws_cnt	found_age_1	found_age_2	found_age_3	.
0	다바66a99a	0.0	0.0	0.0	0.0	0.00	0.0	0.0	0.0	0.0	.
1	다바66a99b	0.0	0.0	0.0	0.0	0.00	0.0	0.0	0.0	0.0	.
2	다바66b99a	88.0	13.0	56.0	0.0	1781.97	0.0	0.0	0.0	0.0	.

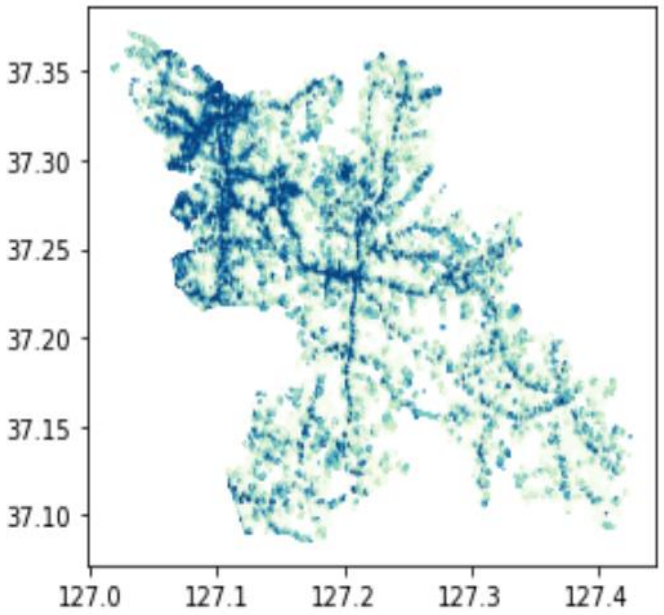
500*500 격자를 기준으로 기존 데이터 통합

01 유동인구 시각화

07~09시까지 시간대를 출근시간대로 정의하고, 17시~19시 까지를 퇴근시간대로 정의해
출근시간대의 유동인구와 퇴근시간대의 유동인구를 비교



출근시간대 유동인구



퇴근시간대 유동인구

→

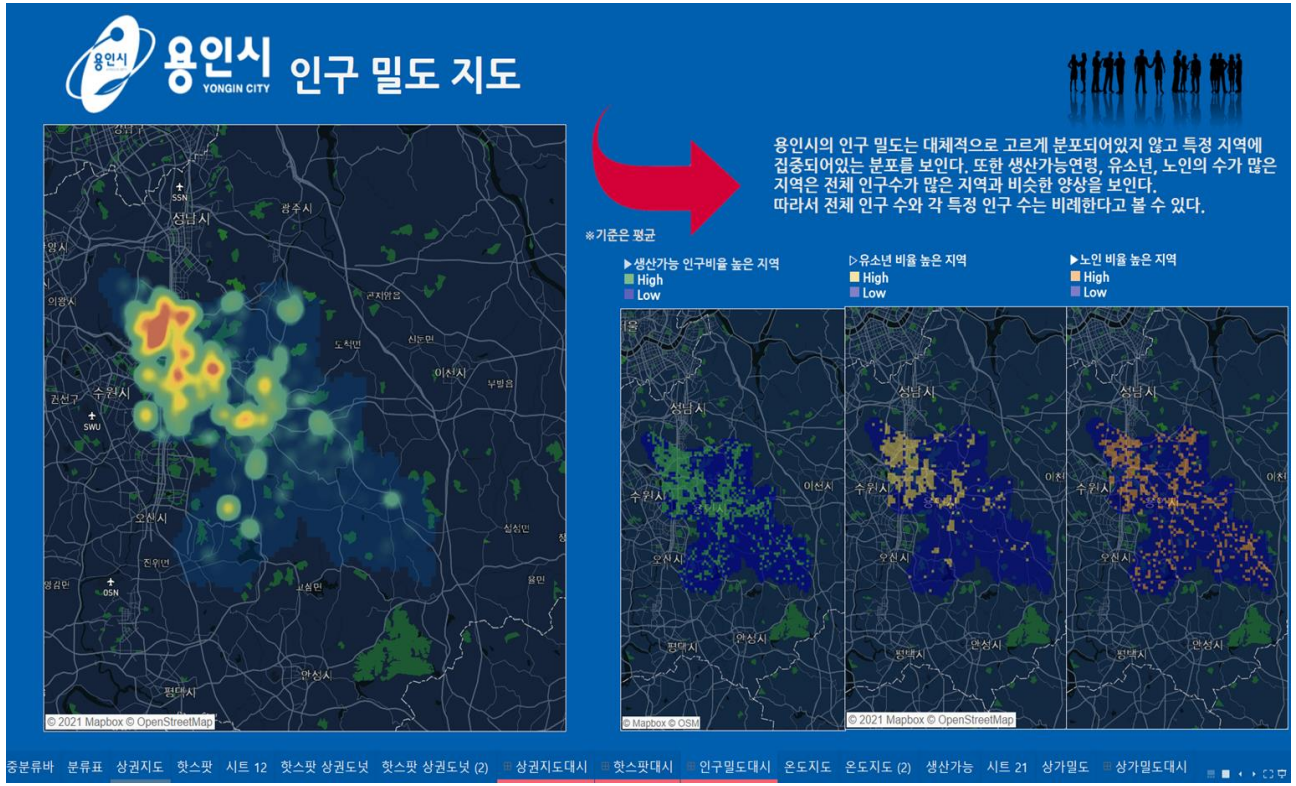
유동인구는 시간대에 따라
큰 변화를 보이지 않았고,

수지구 및 기흥구 북부에
몰려 있어 시간보다는 공간에 많
은 영향을 받는다는 것을 시각화
를 통해 확인..

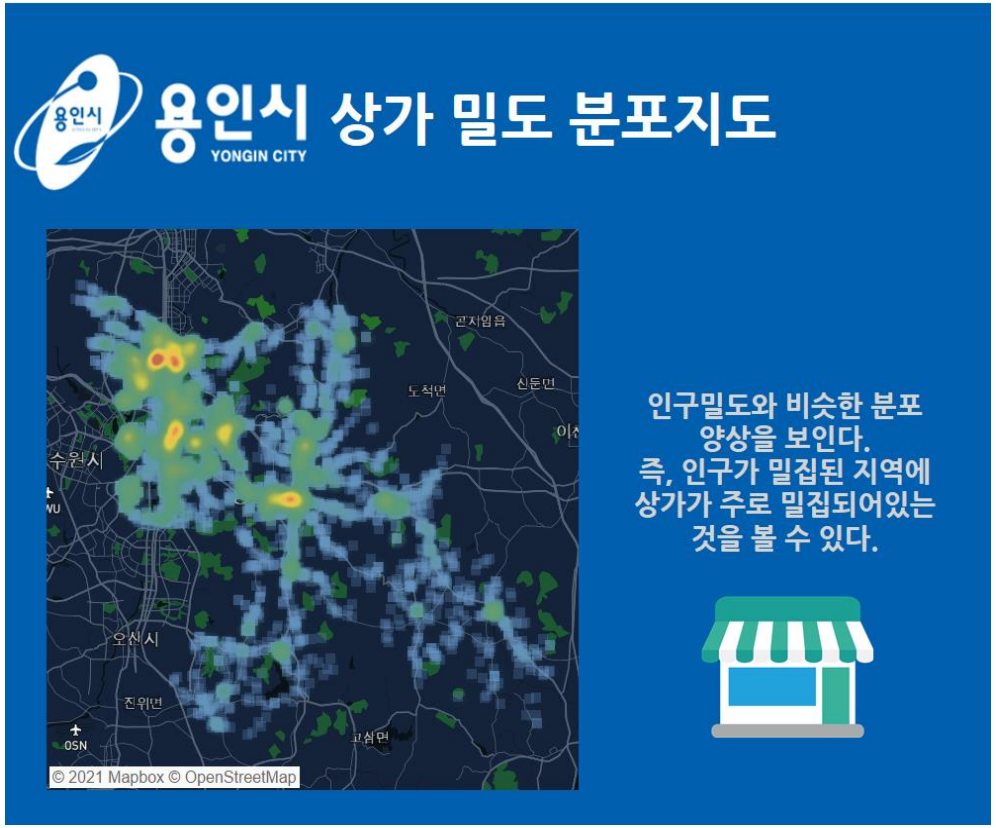
02 인구 밀도 및 상점 개수 히트맵

태블로를 활용하여 인구, 상점 개수 밀도 지수 시각화했을 때, 인구와 상점의 밀도 분포는 비슷한 양상을 보입니다.

용인시 인구 밀도 지도
: 전체&생산가능&유소년&노인별 인구 분포 시각화

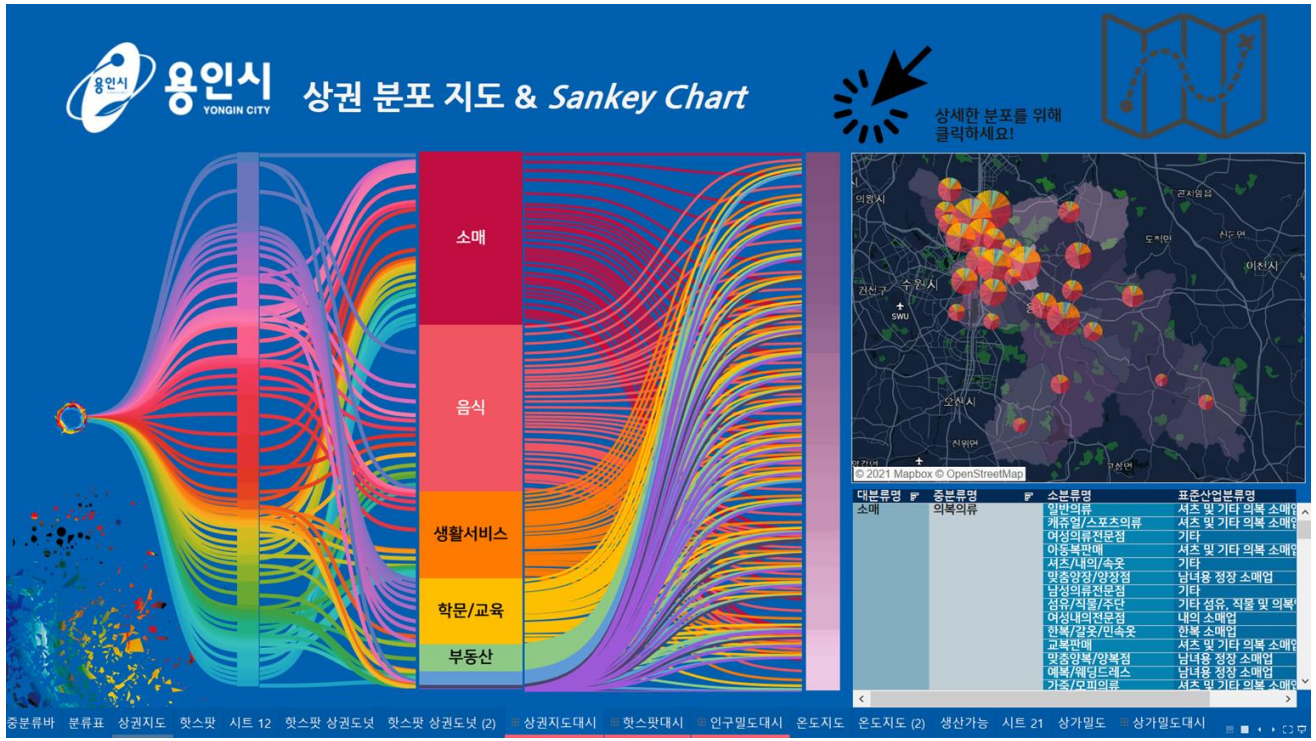


상가 밀도 분포지도
: 인구밀도분포와 비슷한 양상

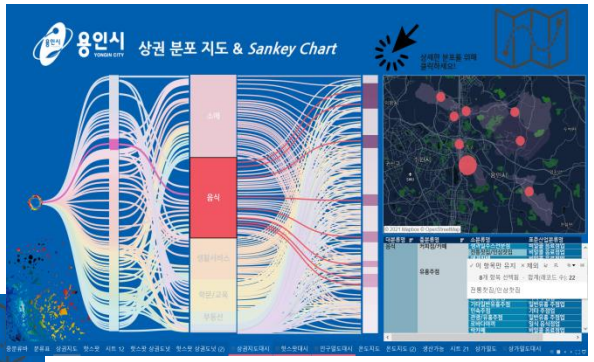


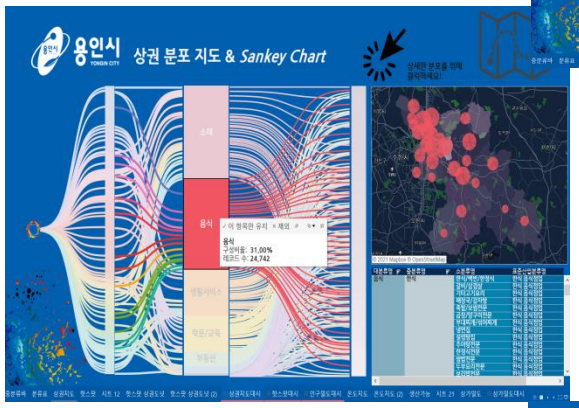
03 생키 차트를 활용한 상권 분포 지도

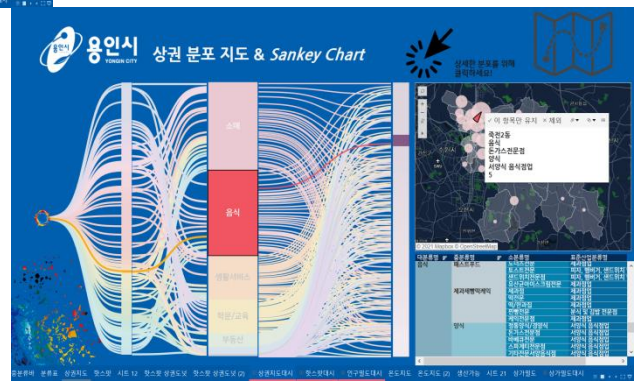
용인시 상가의 대분류별, 중분류별, 소분류별, 표준분류명별 분포를 태블로를 활용해 생키차트로 시각화 및 실행



원하는 파트를 클릭하면
해당 업종의 분포를
각 생키차트, 분류표, 지도에서
확인할 수 있다.

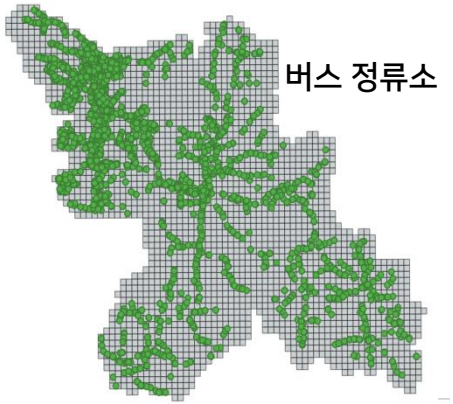


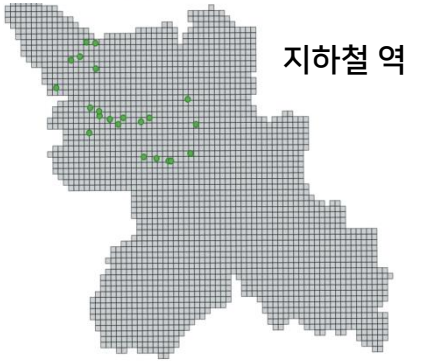




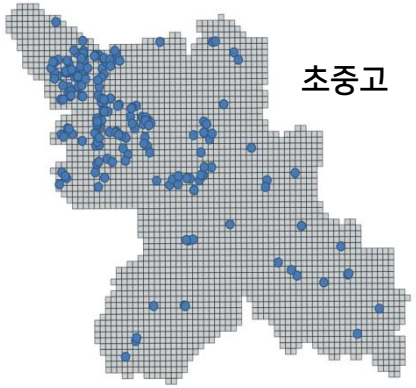
04 환경요인(독립변수) 분포

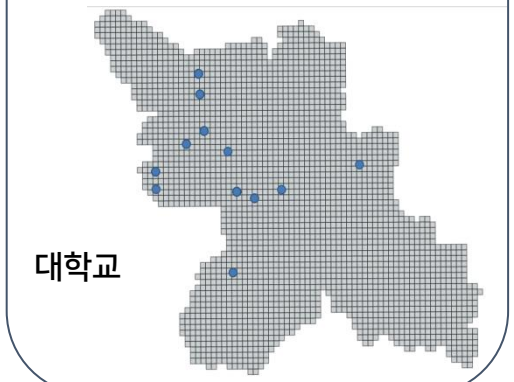
교통관련 변수



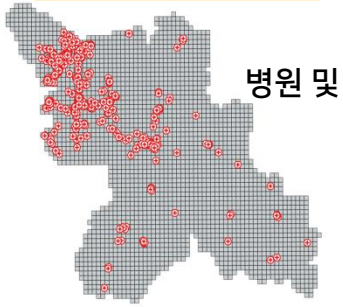


교육관련 변수



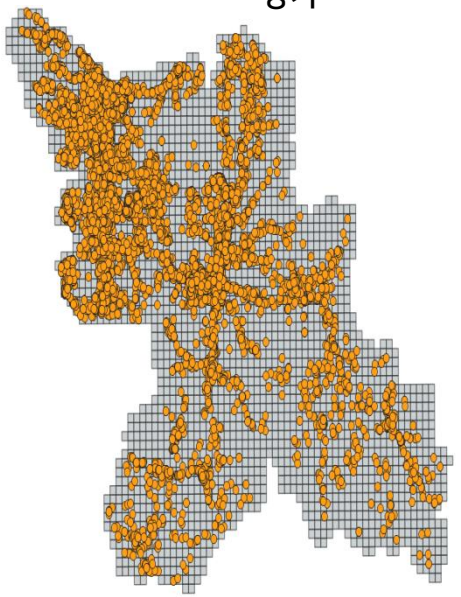


의료 관련 변수



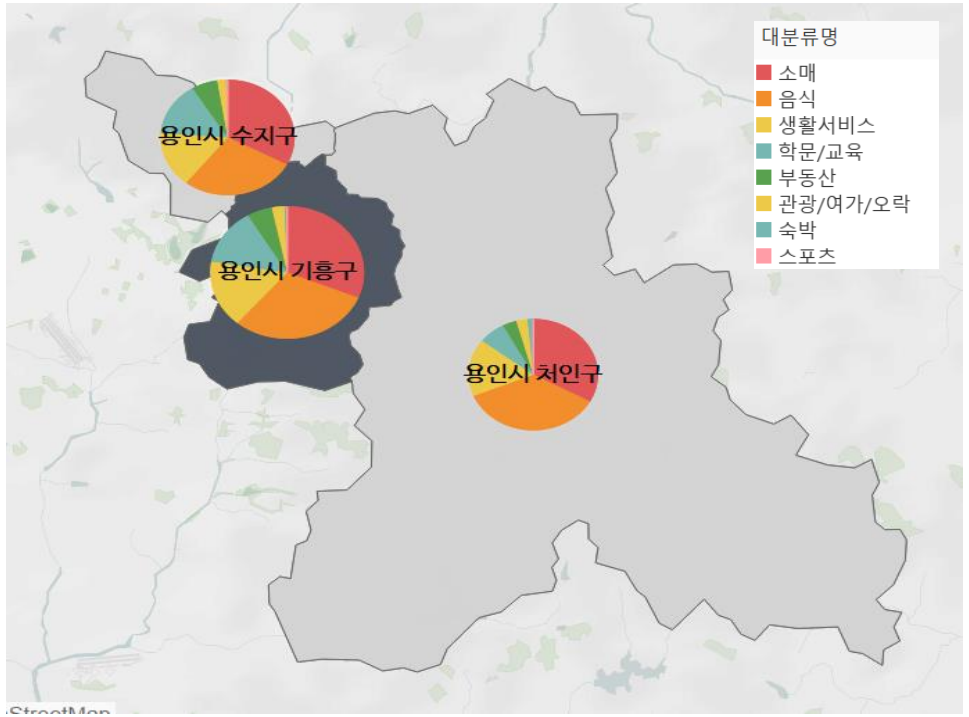
문화 관련 변수

사업체 변수

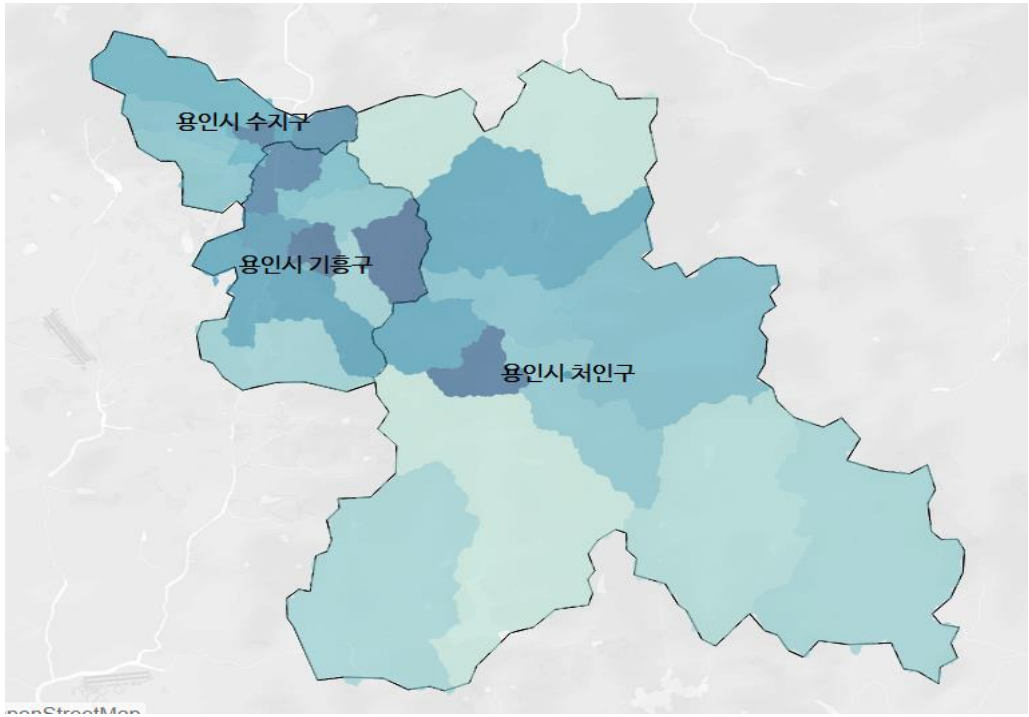


05 행정동별 시각화

구별 행정동별 상가의 수를 시각화.
구별로는 기흥구가 상가가 가장 많으며
행정동별로는 기흥구의 구갈동, 동백동와 처인구의 중앙동에 많은 현황을 확인함.



구별 사업체 시각화
** 파이차트로 대분류별 업종 분포 표현*

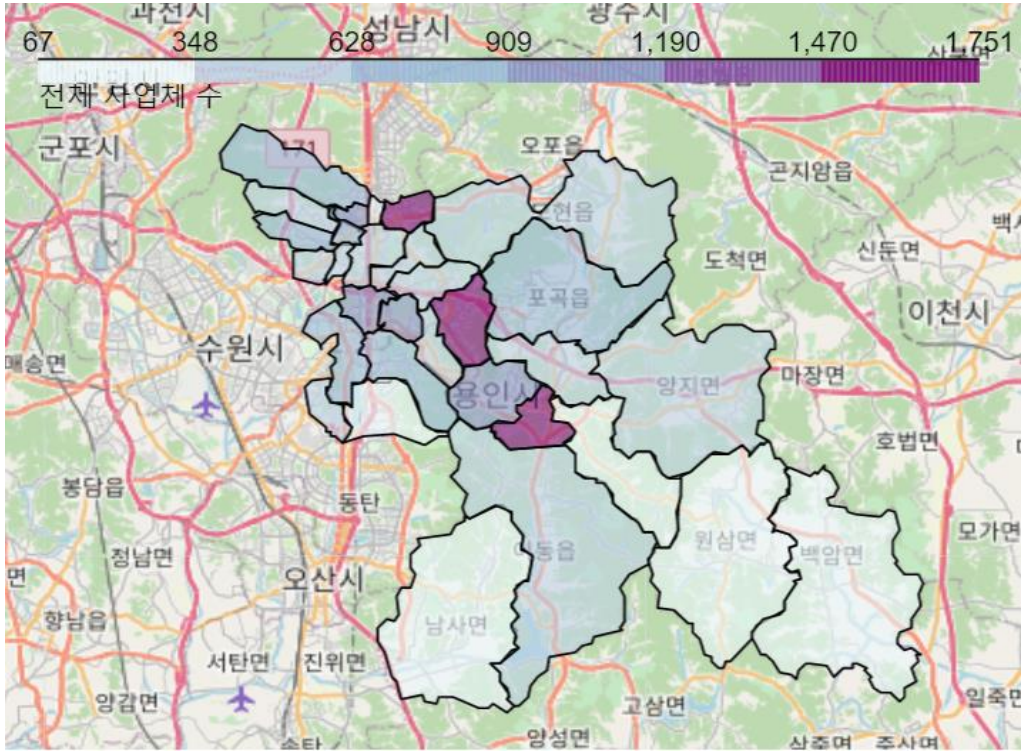


행정동별 상가 시각화

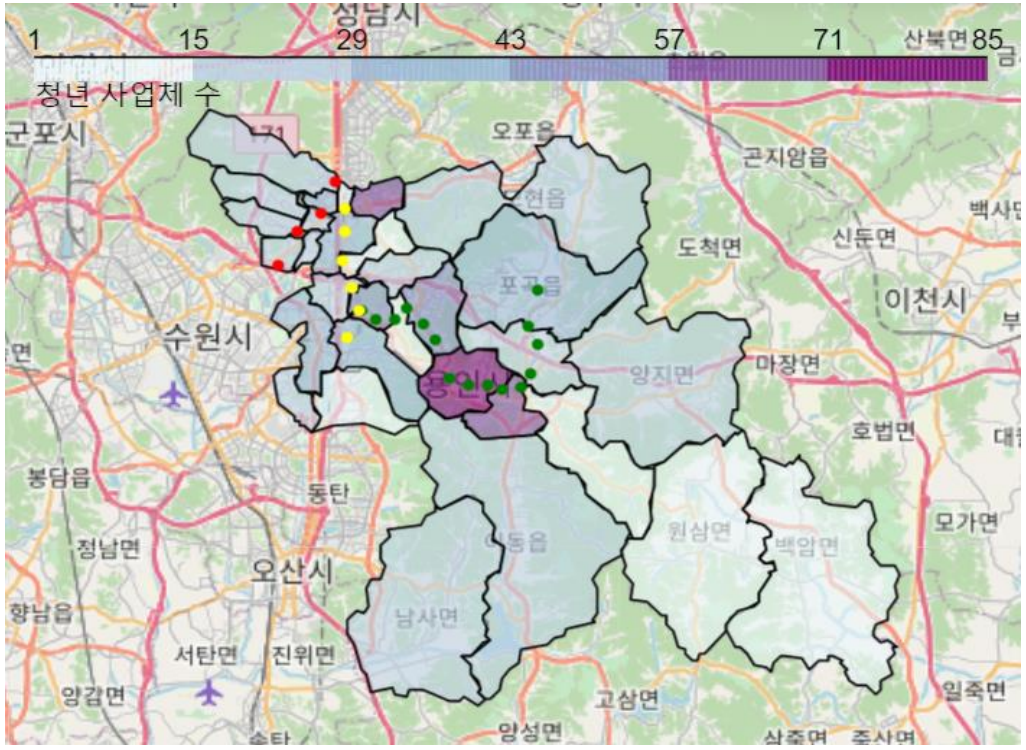
06 청년 사업체 시각화

Folium을 활용해 전체 사업체 및 청년사업체를 행정동별로 시각화
역삼동의 경우, 전체 사업체 수 대비 청년 사업체수가 두드러지게 높게 나타남.

**청년사업체 : 대표 연령이 10대, 20대인 사업체 수의 합*



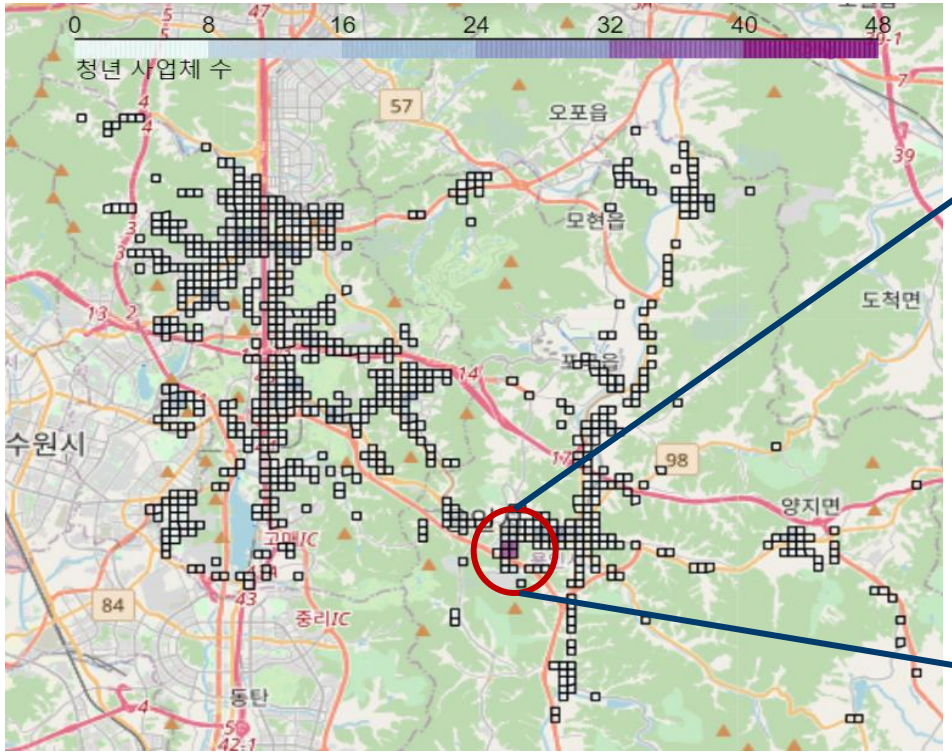
행정동별 전체 사업체 수



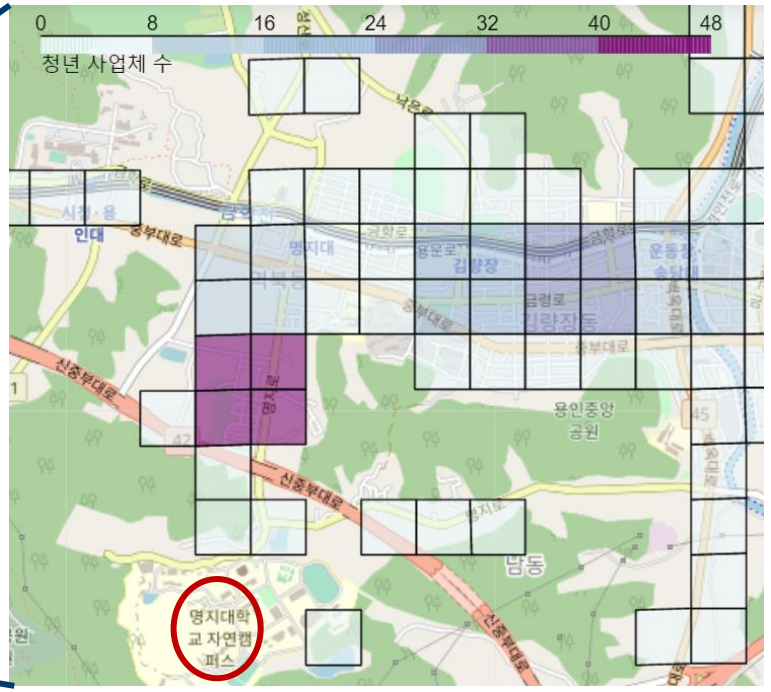
행정동별 청년 사업체 수

06 청년사업체 시각화

행정동별이 아닌 격자별로 청년사업체수의 분포를 파악해,
청년사업체가 많은 격자의 특징을 살펴봄. 그 결과 청년사업체가 가장 많은 구간은 대부분은 대학교 근처에 위치함을 발견하여 청년 사업체와 대학교의 관계를 확인할 수 있었음.



격자별 청년 사업체수



가장 값이 높은 부분: 명지대학교 근처에 위치

07 청년사업체 EDA

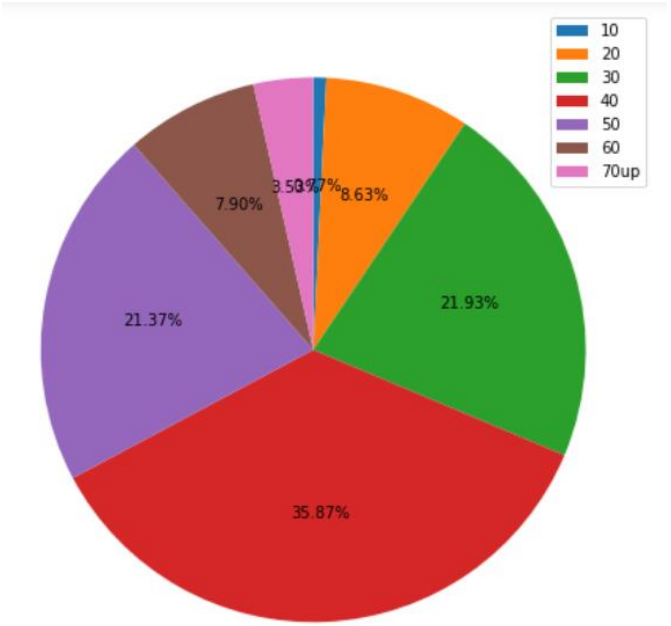
청년사업체 수 합과 청년사업체 비율 평균 순 행정동
둘 다 상위 10개에 속하는 행정동인 죽전1동, 역삼동, 포곡읍, 상갈동 4개의 동을 청년사업체가 많은 행정동으로 지정하여 EDA를 진행

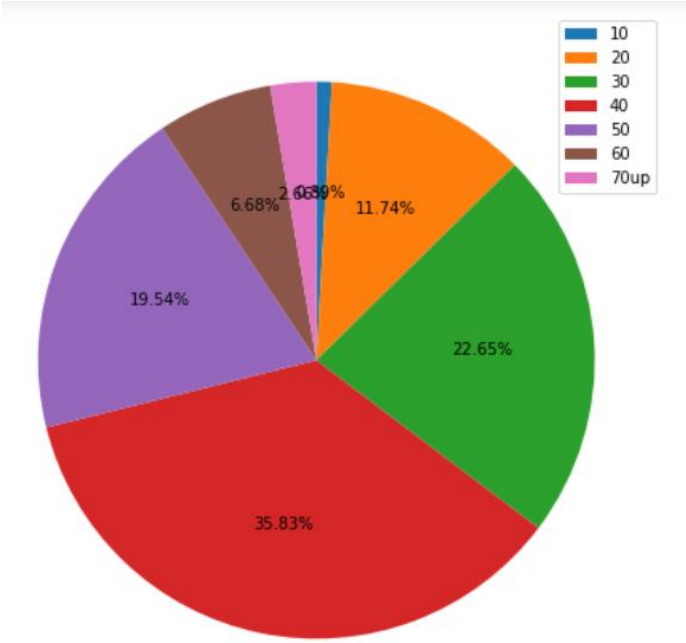
```
list(set(sum_up).intersection(mean_up))
```

```
['죽전1동', '역삼동', '포곡읍', '상갈동']
```

	ADM_DR_NM	sum
0	역삼동	85.0
1	중앙동	69.0
2	죽전1동	61.0
3	동백동	55.0
4	구갈동	49.0
5	포곡읍	38.0
6	영덕동	37.0
7	풍덕천1동	34.0
8	상갈동	32.0
9	보정동	32.0
10	신갈동	27.0

	ADM_DR_NM	mean
0	역삼동	0.066635
1	죽전2동	0.059267
2	양지면	0.058421
3	남사면	0.056718
4	서농동	0.053140
5	유림동	0.049799
6	상갈동	0.043689
7	성복동	0.039978
8	상현2동	0.038957
9	죽전1동	0.037369
10	포곡읍	0.037073





청년사업체 많은 행정동이 20대 카드 사용자 비율이 11.74%로, 전체 행정동에 비해 높음

07 청년사업체 EDA

청년사업체 많은 행정동의 업종 비율을 전체 행정동과 비교.
제조업, 교육 서비스업, 보건업 및 사회복지 서비스업, 예술 스포츠 및 여가관련 서비스업 등
대부분 서비스업과 관련된 업종이 많음.

01 격자별 핫스팟 분석

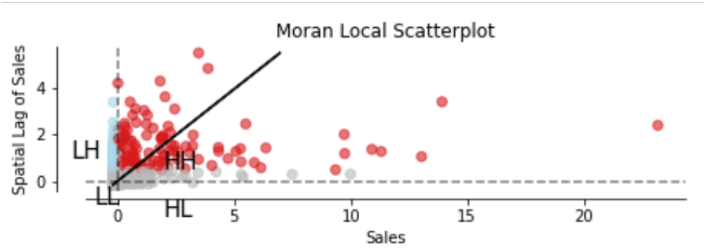
핫스팟 분석(Hot Spot Analysis) 이론

1. Getis-Ord General G : 전역적 군집 패턴 파악
   - 귀무가설 H_0 : 분석 대상 공간 값의 공간적 군집 경향이 없다.
   - Z 검정통계량 : $G = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{i,j}\, x_i\, x_j}{\sum_{i=1}^{n}\sum_{j=1}^{n} x_i\, x_j}$, $\forall\, i \neq j$

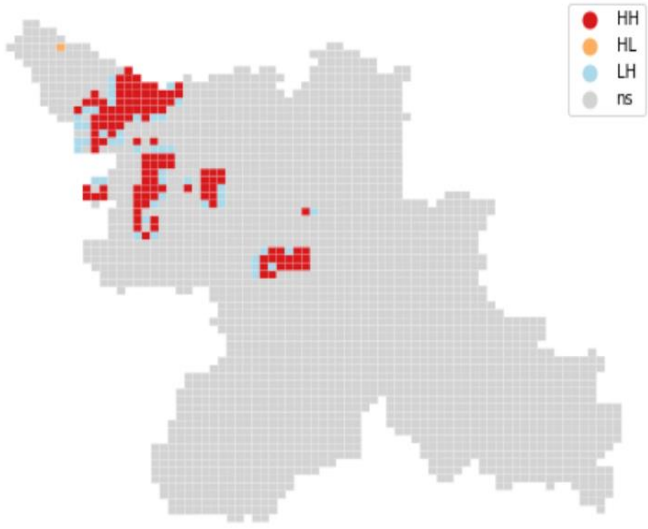
2. Getis-Ord Gi* : 국지적 군집 패턴 파악
   - 귀무가설 H_0 : 인접 지역들과의 개별적 군집경향(핫스팟)이 없다.
   - Z 검정통계량 : $G_i^*(d) = \frac{\sum_{j=1}^{n} w_{i,j}\, x_j - \bar{X}\sum_{j=1}^{n} w_{i,j}}{SD\sqrt{\frac{[n\sum_{j=1}^{n} w^2_{i,j} - (\sum_{j=1}^{n} w_{i,j})^2]}{n-1}}}$, all j

- i, j : 분석의 공간 단위 (grid)
- x_i, x_j : i 또는 j 단위의 데이터
- $w_{i,j}$: i 또는 j 지역간 공간 가중치 (Spatial Weight)
- n: 분석 공간 단위(grid)의 수

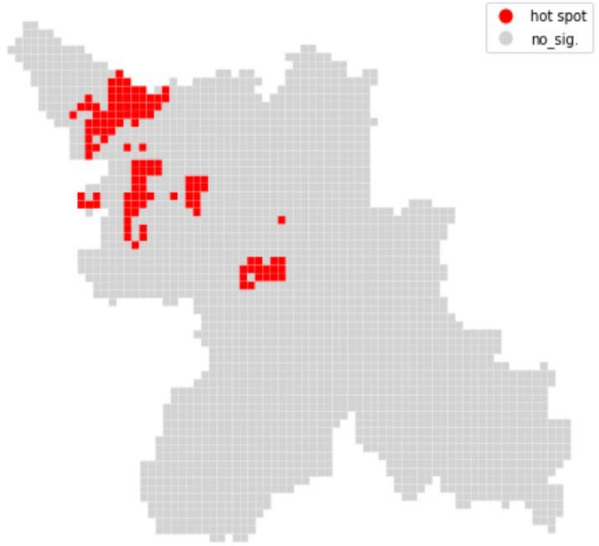
1,2번에 대해 전역적, 국지적 군집 패턴을 파악하기 위해 각 검정통계량으로 귀무가설 검정



1. Moran 지수를 사용해 전역적 군집 패턴 파악



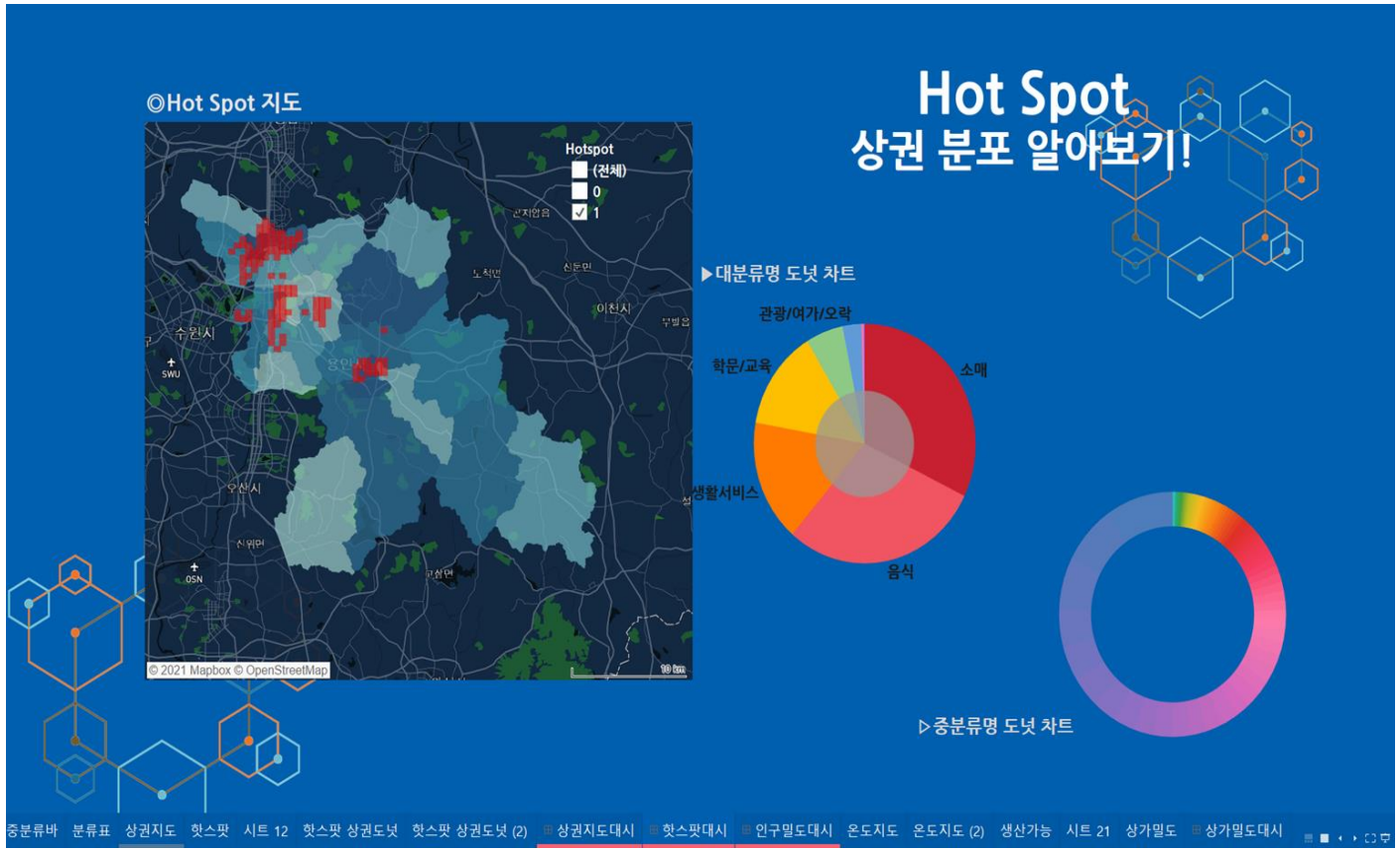
2. 주변값과 중심값이 모두 높은 군집인 HH를 핫스팟 후보군집으로 선택



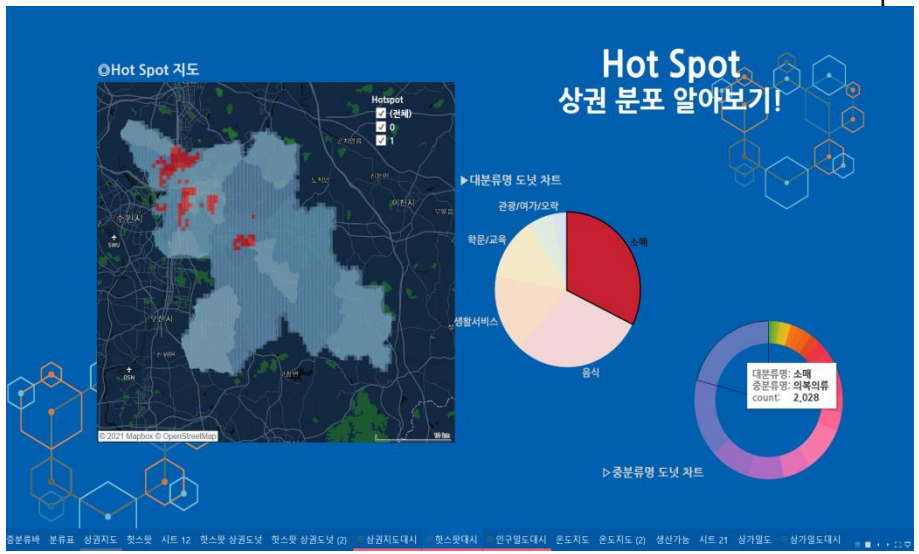
3. 선택된 후보군집을 국지적 패턴파악을 시행해 P-value에 따라 유의미한 핫스팟 군집 추출

02 로지스틱 회귀 분석을 이용한 핫스팟 분석

Hot Spot 상권분포 : 앞에 Getis를 활용해 나온 Hot spot 상권을 태블로로 시각화
전체 업종 대분류 정보를 도넛 차트로 시각화함.



대분류명 도넛차트를 클릭하면
그 대분류에 해당하는 중분류 업종의
비율과 이름을 볼 수 있음.



02 로지스틱 회귀 분석을 이용한 핫스팟 분석

목적) 종속변수의 두 범주 중 어떤 범주에 속할 것인지 분류예측을 할 수 있다. 또한 이런 결과가 나오게 된 이유, 각 독립변수의 영향 정도를 파악할 수 있음.

회귀분석 사용할 데이터:fin_df

```
In [18]: fin_df.columns
Out[18]: Index(['gid500', 'all_cnt', 'old_cnt', 'adult_cnt', 'young_cnt', '건물연면적',
       'ws_cnt', 'found_age_1', 'found_age_2', 'found_age_3', 'found_age_4',
       'found_age_5', 'found_age_6', 'runout_cnt', 'smbiz_yn_cnt',
       'rpr_per_gender_m', 'rpr_per_gender_f', 'rpr_per_age_bin_10',
       'rpr_per_age_bin_20', 'rpr_per_age_bin_30to50',
       'rpr_per_age_bin_50over', 'sme_loan_cnt', 'sme_loan_y_1',
       'sme_loan_y_2', 'sme_loan_y_3', 'sme_loan_y_4', 'sme_loan_y_5',
       'sme_loan_y_6', 'sales_est_amt_201703', 'sales_est_amt_201706',
       'sales_est_amt_201709', 'sales_est_amt_201712', 'sales_est_amt_201803',
       'sales_est_amt_201806', 'sales_est_amt_201809', 'sales_est_amt_201812',
       'sales_est_amt_201903', 'sales_est_amt_201906', 'sales_est_amt_201909',
       'sales_est_amt_201912', 'sales_est_amt_202003', 'sales_est_amt_202006',
       'sales_est_amt_202009', 'age10_ratio', 'age20_ratio', 'age30_ratio',
       'age40_ratio', 'age50_ratio', 'age60_ratio', 'age70_ratio',
       'cnt_market', 'cnt_busstop', 'cnt_subway', 'cnt_clinic', 'cnt_hospital',
       'cnt_element_s', 'cnt_middles_s', 'cnt_high_s', 'cnt_university',
       'cnt_landmark', 'geometry'],
      dtype='object')
```

독립변수 선정 :

1. 지하철 역, 버스 정류장, 도로폭 -> 교통 접근성 : 'cnt_busstop', 'cnt_subway' -->합해서 하나의 변수로 만들어준다: traffic
2. 초중고, 대학교 -> 교육 'cnt_element_s', 'cnt_middles_s', 'cnt_high_s', 'cnt_university' -->합:schools
3. 병원 -> 의료 :'cnt_clinic', 'cnt_hospital' -->합:medical
4. 랜드마크 -> 문화 :'cnt_landmark' --> landmark
5. 인구 밀도 -> all_cnt old_cnt adult_cnt young_cnt --> all_cnt만 사용(태블로 인구지도로 근거) : population
6. 사업체 수:'ws_cnt' 'cnt_market'-->합: store (아래 상관계수 근거)

각 변수들을 연관성있는 변수끼리 묶어
새로운 독립변수들로 만들어준 후 분석을 시행

```
x=exist_hot[['traffic', 'schools', 'medical', 'landmark',
       'population', 'store']]
y=exist_hot['hotspot']
```

```
exist_hot['hotspot'].value_counts()
```

```
0.0    1096
1.0     117
Name: hotspot, dtype: int64
```

```
from imblearn.over_sampling import SMOTE
os = SMOTE(random_state=0)
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)
columns = X_train.columns
os_data_X,os_data_y=os.fit_sample(X_train, y_train)
os_data_X = pd.DataFrame(data=os_data_X,columns=columns )
os_data_y= pd.DataFrame(data=os_data_y,columns=['hotspot'])
# we can Check the numbers of our data
print("length of oversampled data is ",len(os_data_X))
print("Number of no hotspot in oversampled data",len(os_data_y[os_data_y['hotspot']==0]))
print("Number of hotspot",len(os_data_y[os_data_y['hotspot']==1]))
print("Proportion of no hotspot data in oversampled data is ",len(os_data_y[os_data_y['hotspot']==0])/len(os_data_X))
print("Proportion of hotspot data in oversampled data is ",len(os_data_y[os_data_y['hotspot']==1])/len(os_data_X))
```

```
length of oversampled data is  1746
Number of no hotspot in oversampled data 873
Number of hotspot 873
Proportion of no hotspot data in oversampled data is  0.5
Proportion of hotspot data in oversampled data is  0.5
```

독립변수들과 핫스팟변수(hotspot이면 1, 아니면 0)를 사용해 로지스틱 회귀분석을 시행하는데
앞서, SMOTE 알고리즘 (Synthetic Minority Oversampling Technique)을 사용해 오버샘플링을 적용

02 로지스틱 회귀 분석을 이용한 핫스팟 분석

RFE (recursive feature elimination)

모든 변수를 우선 다 포함시키고 시작하며, 반복해서 학습을 진행하면서 중요도가 낮은 변수를 하나씩 제거하는 방식

```
from sklearn.feature_selection import RFE
from sklearn.linear_model import LogisticRegression
log_reg = LogisticRegression()
rfe = RFE(log_reg, 6)
rfe = rfe.fit(os_data_X, os_data_y.values.ravel())
print(rfe.support_)
print(rfe.ranking_)
```

```
[ True  True  True  True  True  True]
[1 1 1 1 1 1]
```

```
import statsmodels.api as sm
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary2())##traffic, medical, landmark, population, store 변수가 유의하다.
```

```
Optimization terminated successfully.
         Current function value: 0.460469
         Iterations 8
                          Results: Logit
==================================================================
Model:              Logit            Pseudo R-squared: 0.336
Dependent Variable: hotspot          AIC:              1619.9576
Date:               2021-01-24 03:35 BIC:              1652.7480
No. Observations:   1746             Log-Likelihood:   -803.98
Df Model:           5                LL-Null:          -1210.2
Df Residuals:       1740             LLR p-value:      2.2716e-173
Converged:          1.0000           Scale:            1.0000
No. Iterations:     8.0000
------------------------------------------------------------------
                Coef.   Std.Err.     z     P>|z|    [0.025   0.975]
------------------------------------------------------------------
traffic        -0.2333   0.0290  -8.0328  0.0000  -0.2902  -0.1764
schools        -0.1601   0.1119  -1.4307  0.1525  -0.3794   0.0592
medical        -0.2359   0.0399  -5.9085  0.0000  -0.3142  -0.1577
landmark       -3.1747   0.8652  -3.6694  0.0002  -4.8705  -1.4790
population      0.0003   0.0000   6.4879  0.0000   0.0002   0.0004
store           0.0149   0.0017   8.8905  0.0000   0.0116   0.0182
==================================================================
```

유의한 변수들로 로지스틱회귀분석 재실시

```
cols=['traffic', 'medical', 'landmark','population', 'store']
X=os_data_X[cols]
y=os_data_y['hotspot']
```

```
import statsmodels.api as sm
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary2())
```

```
Optimization terminated successfully.
         Current function value: 0.461109
         Iterations 8
                          Results: Logit
==================================================================
Model:              Logit            Pseudo R-squared: 0.335
Dependent Variable: hotspot          AIC:              1620.1935
Date:               2021-01-28 06:46 BIC:              1647.5189
No. Observations:   1746             Log-Likelihood:   -805.10
Df Model:           4                LL-Null:          -1210.2
Df Residuals:       1741             LLR p-value:      4.5644e-174
Converged:          1.0000           Scale:            1.0000
No. Iterations:     8.0000
------------------------------------------------------------------
                Coef.   Std.Err.     z     P>|z|    [0.025   0.975]
------------------------------------------------------------------
traffic        -0.2409   0.0286  -8.4301  0.0000  -0.2969  -0.1849
medical        -0.2376   0.0395  -6.0137  0.0000  -0.3151  -0.1602
landmark       -3.1939   0.8691  -3.6750  0.0002  -4.8972  -1.4905
population      0.0003   0.0000   6.4071  0.0000   0.0002   0.0004
store           0.0150   0.0017   8.9856  0.0000   0.0117   0.0183
==================================================================
```

02 로지스틱 회귀 분석을 이용한 핫스팟 분석

```
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
```

```
LogisticRegression()
```

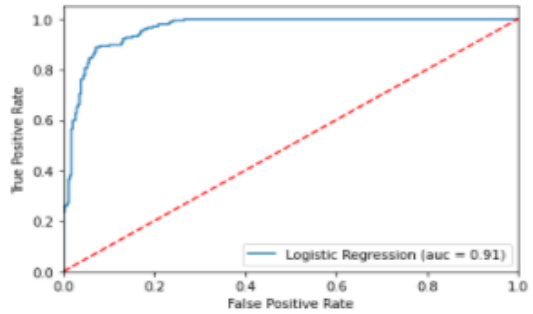
```
y_pred = logreg.predict(X_test)
print('test set에 대한 분류기 정확도: {:.2f}'.format(logreg.score(X_test, y_test)))
```

```
test set에 대한 분류기 정확도: 0.91
```

```
#confusion matrix
from sklearn.metrics import confusion_matrix
confusion_matrix = confusion_matrix(y_test, y_pred)
print(confusion_matrix)
```

```
[[254  22]
 [ 27 221]]
```

```
##ROC Curve
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
logit_roc_auc = roc_auc_score(y_test, logreg.predict(X_test))
fpr, tpr, thresholds = roc_curve(y_test, logreg.predict_proba(X_test)[:,1])
plt.figure()
plt.plot(fpr, tpr, label='Logistic Regression (auc = %0.2f)' % logit_roc_auc)
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.legend()
plt.savefig('Log_ROC')
plt.show()
```




```
np.exp(result.params)#오즈비
```

```
traffic       0.785905
medical       0.788496
landmark      0.041013
population    1.000306
store         1.015110
dtype: float64
```

```
1/np.exp(result.params)#오즈비 역수
```

```
traffic       1.272419
medical       1.268238
landmark     24.382778
population    0.999694
store         0.985115
dtype: float64
```

오즈비는 1을 기준으로 종속변수 y에 미치는 영향을 파악할 수 있으며,
1에서 멀리 떨어질수록 강한 관계를 의미.
위의 변수 중에서는 landmark의 오즈비는 0.04(역수는 24.38)로 1에서 가장 멀리 떨어져있음.
따라서 **핫스팟에는 landmark가 가장 큰 영향을 주는 강한 관계**임을 알 수 있음.
그 외에도 **traffic, medical이 영향을 주지만 그 정도가 크지 않음을 해석할 수 있고, population과 store은 1에 매우 가까워 별다른 영향을 주지 않았음**을 알 수 있음.

02 로지스틱 회귀 분석을 이용한 핫스팟 분석

앞에서 유의한 결과를 얻었지만

축약변수 안의 세부적인 변수들에 대한 영향도 알아보고자
전체 변수를 사용한 로지스틱 회귀분석(핫스팟)을 시행.

```
import statsmodels.api as sm
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary2())
```

```
Warning: Maximum number of iterations has been exceeded.
         Current function value: 0.332720
         Iterations: 35
```

```
/opt/app-root/lib/python3.6/site-packages/statsmodels/base/model.py:568: Converger
verge. Check mle_retvals
  ConvergenceWarning)
```

```
                          Results: Logit
==========================================================================
Model:                 Logit               Pseudo R-squared:    0.520
Dependent Variable:    hotspot             AIC:                1193.8599
Date:                  2021-01-28 07:02    BIC:                1281.3012
No. Observations:      1746                Log-Likelihood:     -580.93
Df Model:              15                  LL-Null:            -1210.2
Df Residuals:          1730                LLR p-value:        4.1808e-259
Converged:             0.0000              Scale:              1.0000
No. Iterations:        35.0000
--------------------------------------------------------------------------
                 Coef.    Std.Err.      z      P>|z|     [0.025     0.975]
--------------------------------------------------------------------------
all_cnt         -0.1283     0.0092  -13.9296  0.0000    -0.1463    -0.1102
old_cnt          0.1290     0.0092   13.9670  0.0000     0.1109     0.1471
adult_cnt        0.1299     0.0093   13.9933  0.0000     0.1117     0.1481
young_cnt        0.1234     0.0090   13.7137  0.0000     0.1058     0.1410
건물연면적           -0.0000     0.0000   -3.4325  0.0006    -0.0000    -0.0000
ws_cnt           0.0197     0.0074    2.6643  0.0077     0.0052     0.0341
cnt_market       0.0193     0.0048    4.0256  0.0001     0.0099     0.0287
cnt_busstop     -0.0401     0.0393   -1.0197  0.3079    -0.1172     0.0370
cnt_subway      -1.7634     0.4070   -4.3328  0.0000    -2.5612    -0.9657
cnt_clinic      -0.2579     0.0491   -5.2557  0.0000    -0.3541    -0.1617
cnt_hospital    -0.4956     0.3667   -1.3515  0.1765    -1.2143     0.2231
cnt_element_s   -0.8599     0.2976   -2.8897  0.0039    -1.4432    -0.2767
cnt_middles_s   -0.8250     0.4549   -1.8138  0.0697    -1.7166     0.0665
cnt_high_s       0.6474     0.5571    1.1620  0.2453    -0.4446     1.7394
cnt_university -19.9596 18765.8842   -0.0011  0.9992 -36800.4167 36760.4975
cnt_landmark    -3.1957     1.0221   -3.1265  0.0018    -5.1991    -1.1924
==========================================================================
```

유의한 변수들로 로지스틱회귀분석 재실시

```
cols=['all_cnt', 'old_cnt', 'adult_cnt', 'young_cnt', '건물연면적',
       'cnt_market', 'cnt_subway', 'cnt_clinic',
       'cnt_element_s','cnt_landmark']
X=os_data_X[cols]
y=os_data_y['hotspot']
```

```
import statsmodels.api as sm
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary2())
```

```
Optimization terminated successfully.
         Current function value: 0.340121
         Iterations 9
                          Results: Logit
==========================================================================
Model:                 Logit               Pseudo R-squared:    0.509
Dependent Variable:    hotspot             AIC:                1207.7027
Date:                  2021-01-28 07:04    BIC:                1262.3535
No. Observations:      1746                Log-Likelihood:     -593.85
Df Model:              9                   LL-Null:            -1210.2
Df Residuals:          1736                LLR p-value:        1.0216e-259
Converged:             1.0000              Scale:              1.0000
No. Iterations:        9.0000
--------------------------------------------------------------------------
                 Coef.    Std.Err.      z      P>|z|     [0.025     0.975]
--------------------------------------------------------------------------
all_cnt         -0.1293     0.0086  -15.0576  0.0000    -0.1462    -0.1125
old_cnt          0.1299     0.0086   15.1216  0.0000     0.1130     0.1467
adult_cnt        0.1309     0.0087   15.0927  0.0000     0.1139     0.1479
young_cnt        0.1253     0.0084   14.9051  0.0000     0.1088     0.1418
건물연면적           -0.0000     0.0000   -4.6151  0.0000    -0.0000    -0.0000
cnt_market       0.0281     0.0030    9.4491  0.0000     0.0223     0.0340
cnt_subway      -1.5714     0.3772   -4.1660  0.0000    -2.3107    -0.8321
cnt_clinic      -0.1606     0.0397   -4.0416  0.0001    -0.2385    -0.0827
cnt_element_s   -1.0759     0.2858   -3.7650  0.0002    -1.6361    -0.5158
cnt_landmark    -3.4347     1.0330   -3.3250  0.0009    -5.4593    -1.4100
==========================================================================
```

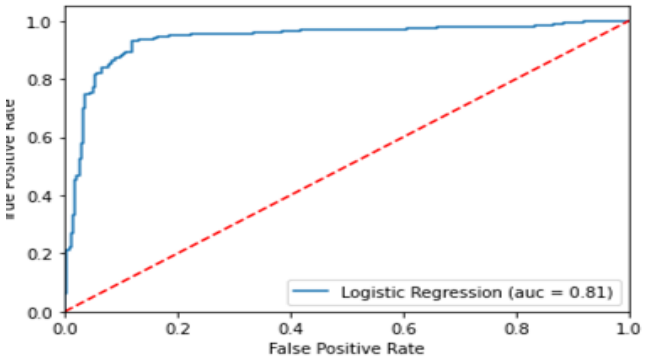
02 로지스틱 회귀 분석을 이용한 핫스팟 분석

```
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
```

```
_pred = logreg.predict(X_test)
rint('test set에 대한 분류기 정확도: {:.2f}'.format(logreg.score(X_test, y_test)))
```

```
est set에 대한 분류기 정확도: 0.80
```

```
#ROC Curve
rom sklearn.metrics import roc_auc_score
rom sklearn.metrics import roc_curve
ogit_roc_auc = roc_auc_score(y_test, logreg.predict(X_test))
pr, tpr, thresholds = roc_curve(y_test, logreg.predict_proba(X_test)[:,1])
lt.figure()
lt.plot(fpr, tpr, label='Logistic Regression (auc = %0.2f)' % logit_roc_auc)
lt.plot([0, 1], [0, 1],'r--')
lt.xlim([0.0, 1.0])
lt.ylim([0.0, 1.05])
lt.xlabel('False Positive Rate')
lt.ylabel('True Positive Rate')
lt.legend()
lt.savefig('Log_ROC')
lt.show()
```



`np.exp(result.params)#오즈비`

all_cnt	0.878676
old_cnt	1.138669
adult_cnt	1.139813
young_cnt	1.133514
건물연면적	0.999987
cnt_market	1.028548
cnt_subway	0.207747
cnt_clinic	0.851616
cnt_element_s	0.340975
cnt_landmark	0.032236

dtype: float64

`1/np.exp(result.params)#오즈비 역수`

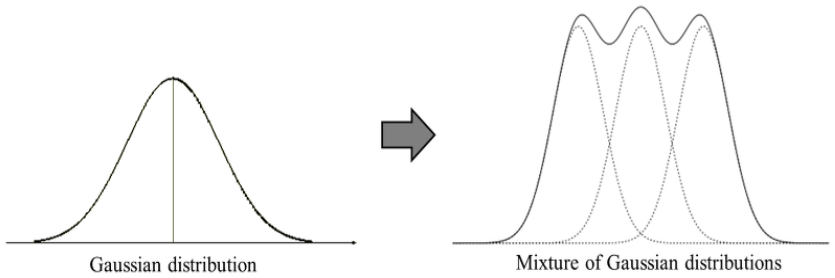
all_cnt	1.138076
old_cnt	0.878219
adult_cnt	0.877337
young_cnt	0.882212
건물연면적	1.000013
cnt_market	0.972244
cnt_subway	4.813556
cnt_clinic	1.174238
cnt_element_s	2.932763
cnt_landmark	31.021250

dtype: float64

오즈비를 통해 y(Hot spot)에 미치는 영향을 보니,
cnt_landmark가 1과 가장 멀리 떨어져있고 0.03으로 아주 강한 영향을 주는 관계임을 알 수 있음.
또한 **cnt_subway, cnt_element_s도 큰 영향을 주며, 나머지 변수들은 1에 매우 가까워 별다른 영향을 크게 주지 않았음**을 확인.

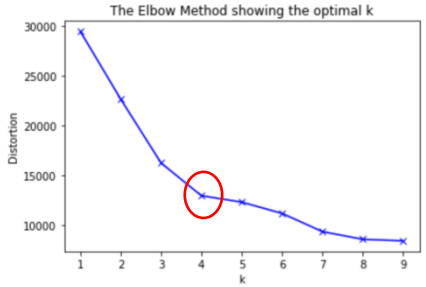
03 매출 변수 제외 변수들 사용 격자별 군집분석

Gaussian Mixture Model을 활용한 EM 알고리즘



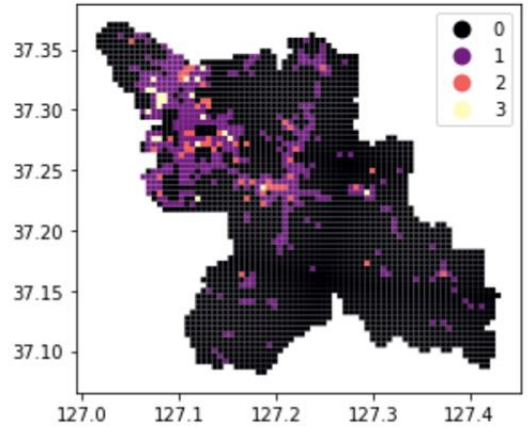
데이터가 k개의 정규분포로 이루어졌다고 가정하고,
군집별 조건부 확률을 최대화로 하는 군집에 할당해
군집분석을 시행.
이 알고리즘비례하도록 클로스터를 지정해
정확도를 향상시키는 은 여러개의 클러스터에 대해,
사후확률에 장점이 존재

군집 개수 설정



군집의 수를 증가시킴에 따라
SSE값이 급격하게 감소하는
elbow point로 군집 개수설정

군집분석 결과 시각화

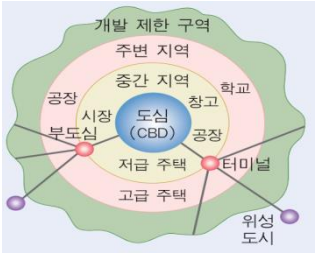


3번 군집 앞선 hot-spot 분석과 겹치는 부분이 많음.
-> 도시 내부 구조 중, 도심역할을 하는 군집으로 파악

2번 군집 3번 군집인 도심지역 주변에 위치한 경우가 많음.
- > 부도심 역할의 군집

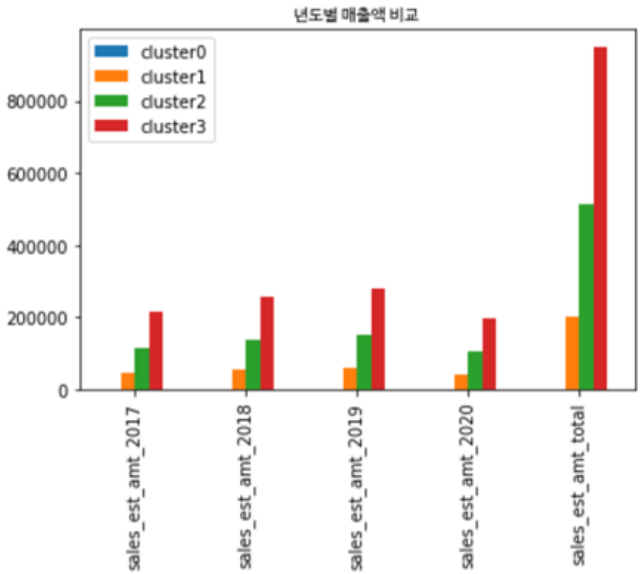
1번 군집 중간지역

0번 군집 주변지역

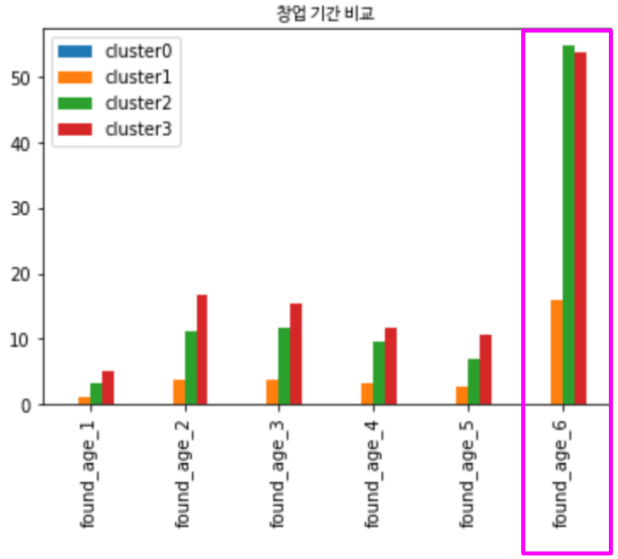


도시 내부 구조

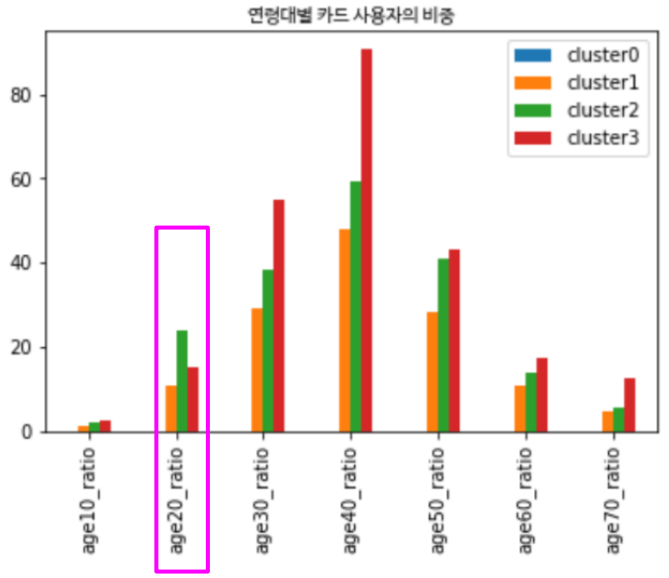
04 군집별 세부 특징



도심지역의 3번 군집은
17~20년도까지 조사된
모든 기간에 대해 매출액이
가장 높았고, 도심지역으로서
hot-spot의 결과와 일치

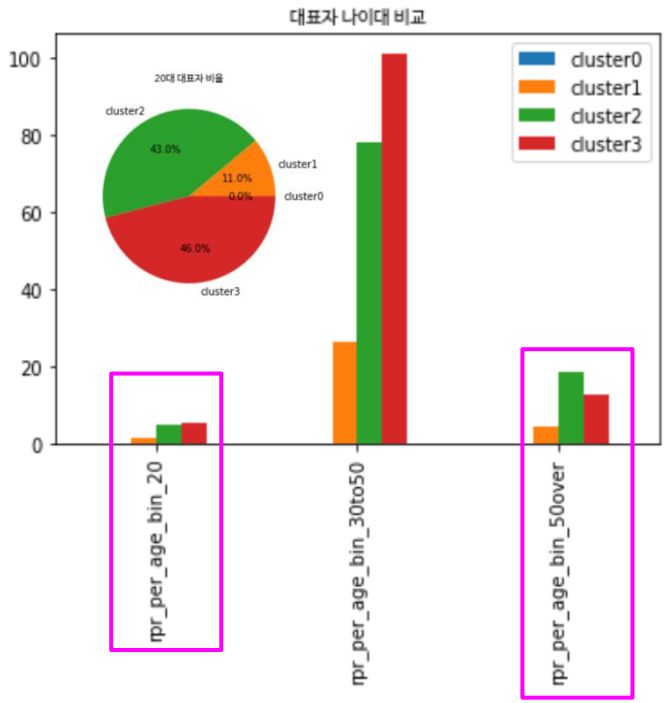


창업 후 3년~ 5년 기간의
'죽음의 계곡'을 넘긴
5년 이상의 사업체가
가장 많았던 군집은 2번 군집

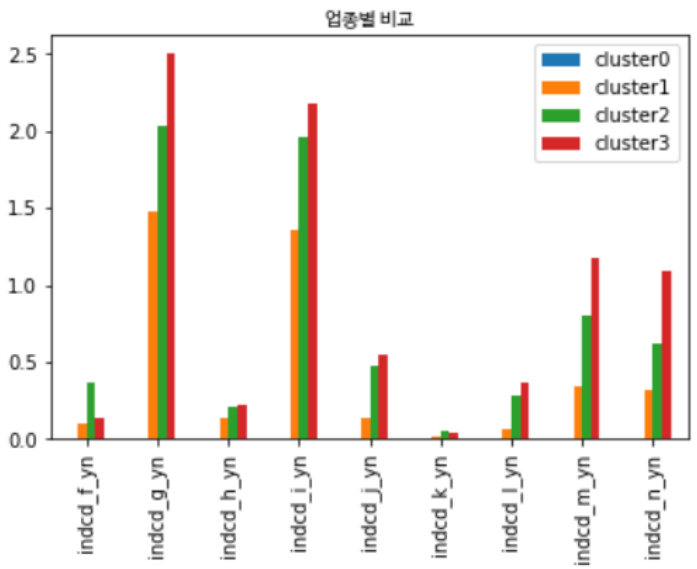


20대 카드 사용자 비중은
2번 군집에서 가장 높음

04 군집별 세부 특징

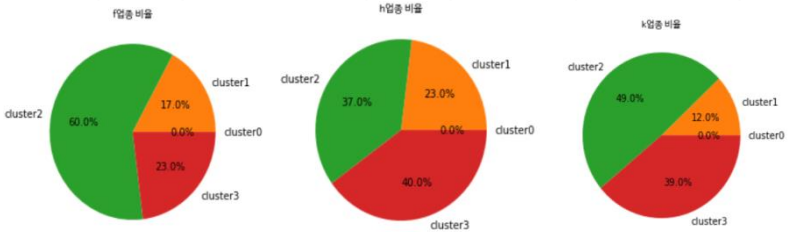


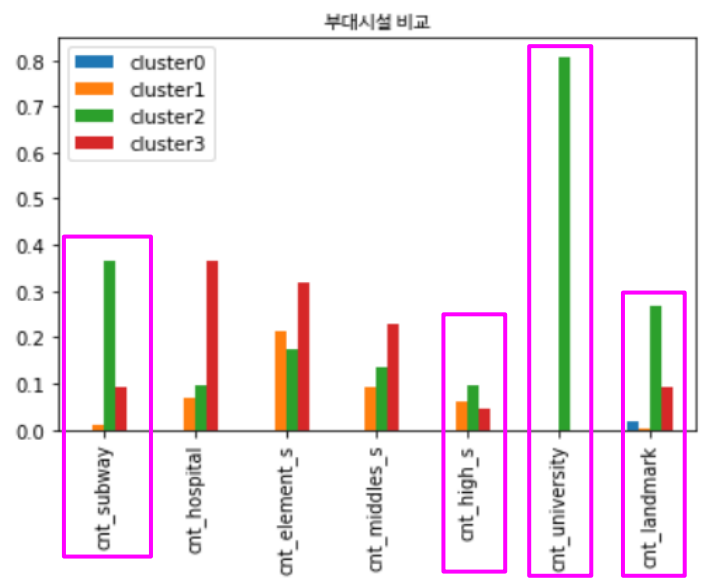
50대 이상의 대표자의 사업체수 변수는 2번 군집에
많이 존재하고, 20대 대표자는
다른 연령에서는 군집 별 차이가 큰 편에 비해
20대 대표자의 경우 2번 군집과 3번 군집에 큰 차이가 나지 않음



2번 군집에서 많이 관찰된 업종

f(건설업) h(운수업) k(금융및보험업)



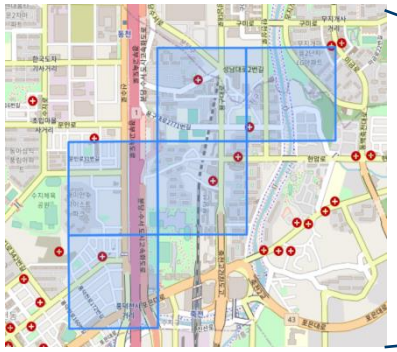


지하철역 수, 고등학교 수, 랜드마크 수가
2번 군집이 가장 많았으며,
특히 대학교는 2번군집에
매우 많이 몰려있는 것을 확인.

04 군집별 세부 특징

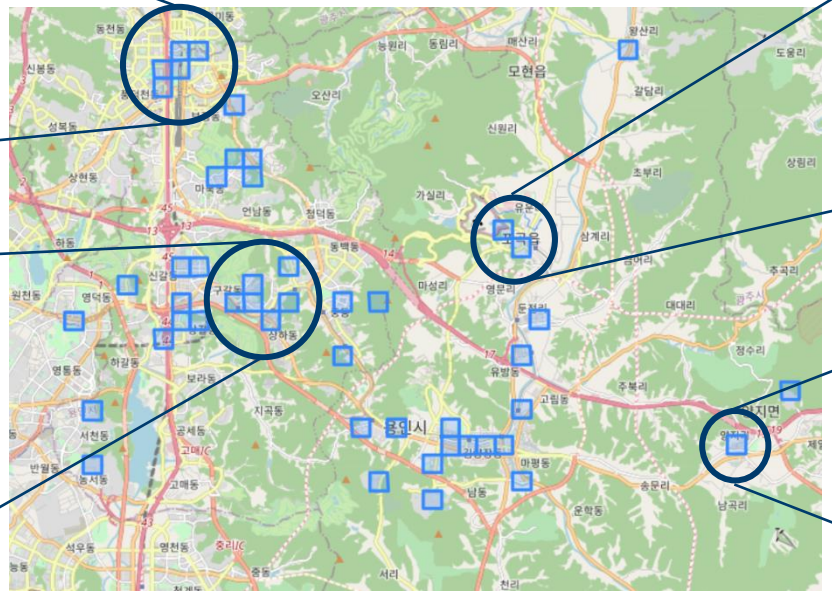
분석 결과

3번 군집은 대부분의 변수가 높은 값을 가졌지만, 2번 군집의 경우
20대 카드 사용자, 20대 대표자, 지하철 역 개수, 고등학교와 대학교 수 , 랜드마크 수 등
청년과 관련된 변수에 대해 높은 값을 가지는 것을 알 수 있음.
따라서, 2번 군집은 청년 창업에 특화되어있는 군집으로 볼 수 있음



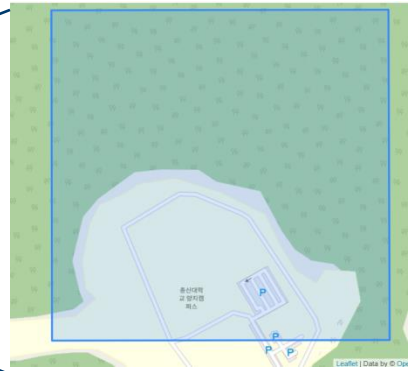
아파트 밀집 및 대로 주변

기업체 주변



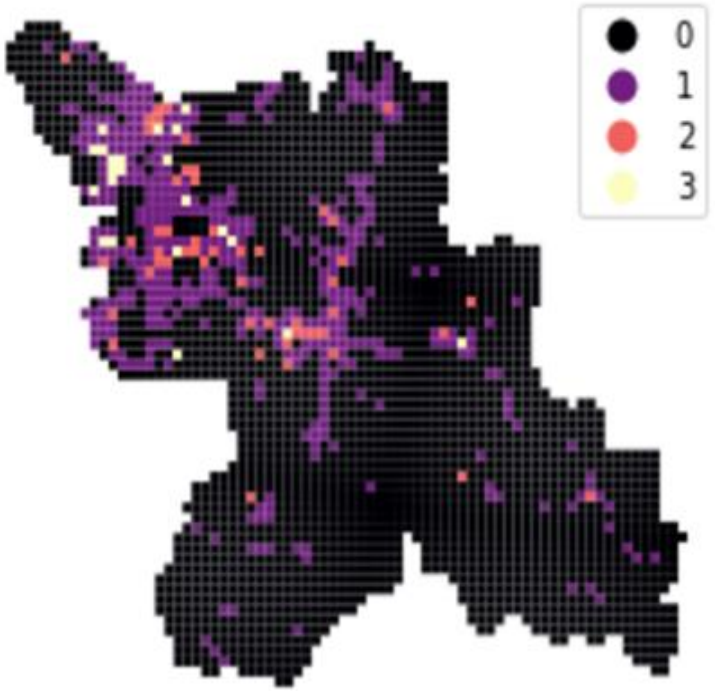
2번 군집의 특징

랜드마크 주변



대학교 인근

01 결론 및 제언



- 3번 군집은 대부분의 변수가 높은 값을 갖는 것으로 보아, 용인시 활동에서 거점이 되는 중심가인 신도심이라고 볼 수 있음.
- 신도심의 경우, 임대료가 비싼 가격을 형성하는 것으로 해석.
- 다른 세대에 비해 상대적으로 초기자본이 적은 청년 창업자에게 신도심은 좋은 입지임에도 불구하고 창업 지역으로 적절하지 않음.
- 따라서 2번 군집에 청년 사업체가 형성되는 경향을 보이며, **청년 창업에 특화되어 있는 2번 군집을 청년 창업을 위한 입지로 추천** 가능.
- 정부는 **해당 2번 군집에 해당하는 지역에 청년 창업 지원을 통해 청년 창업을 활성화**시킬 수 있음.

01 결론 및 제언

인사이트 : 청년몰

**청년몰 : 소상공인시장진흥공단에서 전통시장의 빈 점포와 같은 유휴공간에 20개 이상의 청년상인 점포를 입점시킨 공간*

전통시장에 청년 창업자들이 새로운 트렌드를 적용하며,
전통시장의 활기를 불러 일으켰다.
하지만 청년몰은 청년 창업을 위한 것이 아닌,
전통시장이라는 입지적 악조건을 시작하는 만큼
성공사례보다 실패사례가 훨씬 많다.

연합뉴스(2020.01)

- 청년몰의 인사이트를 얻어, 전통시장 활성화가 아닌 청년 상권 확대를 위한 '청년몰'을 2번 군집에 형성하도록 지원할 필요가 있음.
- 특히, 2번 군집은 현재 청년 창업에 적합한 위치이며, 청년 사업체가 다수 존재하므로, 사업체 간의 연결을 통해 시너지 효과로 하나의 새로운 상권 , 랜드마크를 형성해야 함.
- 청년몰 형성 후, 자연스러운 인구 유입으로 주변 상권의 발달과 함께 상권의 확장으로 새로운 주요 상권의 생성이 목표.

02 기대효과

청년 창업의 특성을 분석해
청년창업의 최적화된 입지 추천
으로 휴업 및 폐업의 수를
줄이고, 지속적으로
성장할 수 있는 창업 분위기를
조성할 수 있음

추천된 입지는 신도심 근처에
형성되어있어 원도심과 신도심
을 연결하는 수단으로
활용될 수 있기 때문에
신도심의 경제효과 확산을
기대할 수 있음

주요 상권의 확대를 통해
청년몰 부근의 상권의 발달도
기대할 수 있음

03 참고 문헌 및 출처

참고문헌

입지추천보고서	NICEBIZMAP
업종추천보고서	NICEBIZMAP
업종별 창업 및 폐업의 지리적 특성 분석	이금숙,박소현 저
시계열 군집분석과 로지스틱 회귀분석을 이용한 골목상권 성장요인 연구	강현모, 이상경 저
서울시 오피스 건물의 입지패턴과 공간적 군집에 미치는 요인 연구	이재수 저
상점 밀도와 업종 다양성을 이용한 서울시 골목상권의 동태적 변화 모니터링 연구	김현철, 안영수
상권 업종별 분포 및 매출 영향요인 분석	정대석, 김형보 저

사용 도구

QGIS

tableau

외부 데이터

데이터명	출처
경기버스 지역별 이용객수	경기도교통정보센터
용인시내 관광지 데이터(2020)	관광지식정보시스템 (위경도 변환 - 구글)
경기도 용인시 전철/지하철 역별 이용객수(2019)	경기도교통정보센터
경기도 용인시 버스정류소 현황(2018)	경기데이터드림
용인시 초등학교, 중학교, 고등학교 현황(2019)	경기디에터드림
전국 고등교육기관(대학)(2020)	교육통계서비스
병원, 의원 인허가 데이터	LOCALDATA 지방행정인허가데이터
도로 폭 정보	용인시 도로공사

감사합니다!