

용인시 청년창업정책  
제안

# Y-스포츠 창업

용인시 청년창업의 미래

팀  
배찬우가부릅니다



# 목차

## 01. 문제인식 및 시각화

- 분석 목적 및 데이터 선정 (p1~2)
- 시각화(1) (p3~10)
- 시각화(2) (p11~20)
- 결론 (p21)

## 02. 데이터 분석

- 분석 방법 및 전처리 (p22~24)
- 사용 함수 정리 (p25~29)
- 분석 결과 및 해석 (p30~35)

## 03. 결론 및 정책 제안

- 결론 및 정책 제안 (p36~37)

# 분석 목적 및 데이터 선정

**분석 목적 : 용인시의 창업 실태를 데이터 시각화와 분석을 통해 파악하고,  
현 상황을 정의해 창업에 효과적인 정책 제시**

**1. 사용한 데이터(한국교통연구원, KCB(코리아크레딧뷰))**

- 1.용인시\_상권\_정보.csv
- 2.용인시\_상권\_업종코드.csv
- 7.용인시\_유동인구.csv
- 13.용인시\_행정경계(읍면동).geojson
- 16.용인시\_용도지역.geojson
- 17.용인시\_소상공인\_매출정보.csv

## 2. 사용한 외부데이터

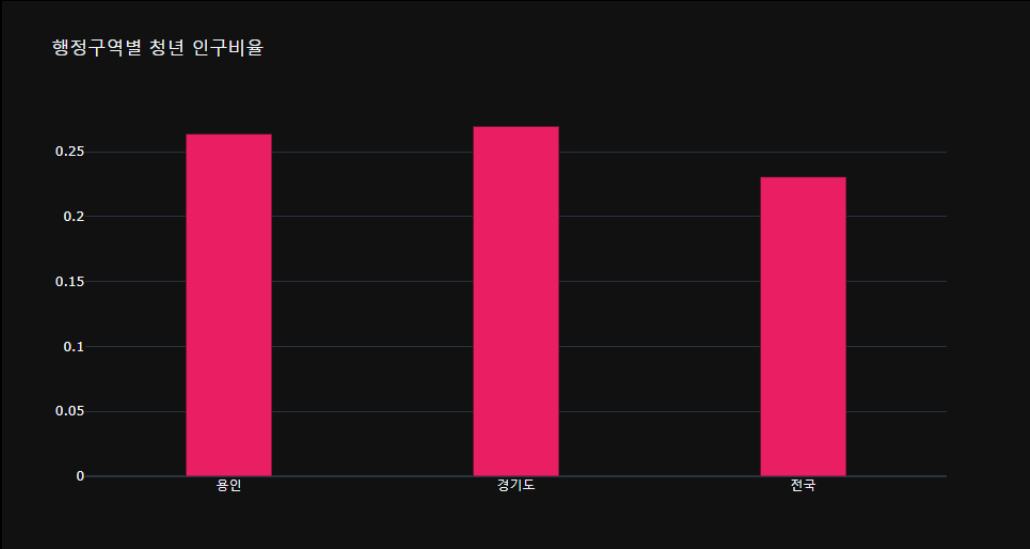
- TL\_SCCO\_SIG.shp : 전국 행정구역(시·군·구)경계를 포함하는 SHAPE 파일  
출처 : GIS DEVELOPER-공간정보시스템/딥러닝 기반 기술 연구소(2021년 1월 기준)
- 경기도만인구비율.csv : 경기도 시·군·구별 인구수와 청년인구의 비율을 포함한 파일  
출처 : 행정안전부 주민등록 인구통계(2020년 12월 기준)
- 청년비율.csv : 전국 시군구별 인구수와 청년인구의 비율을 포함한 파일  
출처 : 행정안전부 주민등록 인구통계(2020년 12월 기준)
- 성별지역별사업등록자.csv : 경기도 시·군·구별 남성과 여성의 사업체 비율을 포함한 파일  
출처 : 창업진흥원(2019년 창업기업실태조사 통계자료)
- 전공별 창업\_male.csv & 전공별창업\_female : 경기도 남성, 여성 창업자의 전공 비율을 포함한 파일  
출처 : 창업진흥원(2019년 창업기업실태조사 통계자료)
- 업종별\_남녀비율.csv : 경기도 업종별 창업자의 남녀 비율을 포함한 파일  
출처 : 창업진흥원(2019년 창업기업실태조사 통계자료)
- 나이\_성별.csv : 경기도 연령별 창업자의 남녀 비율을 포함한 파일  
출처 : 창업진흥원 (2019년 창업기업실태조사 통계자료)

# 시각화(1) 청년 창업 정보 시각화

## 1.1 용인시, 경기도, 전국 청년인구 비율 시각화

- 경기도 각 시·군·구별 인구수와 인구비율을 얻기 위해 '경기도만인구비율.csv'파일과 '청년비율.csv'을 불러옴
- '경기도만인구비율.csv'파일에서 경기도의 청년인구 비율과 용인의 청년인구 비율, '청년비율.csv'파일에서 전국의 청년인구 비율을 얻어냄

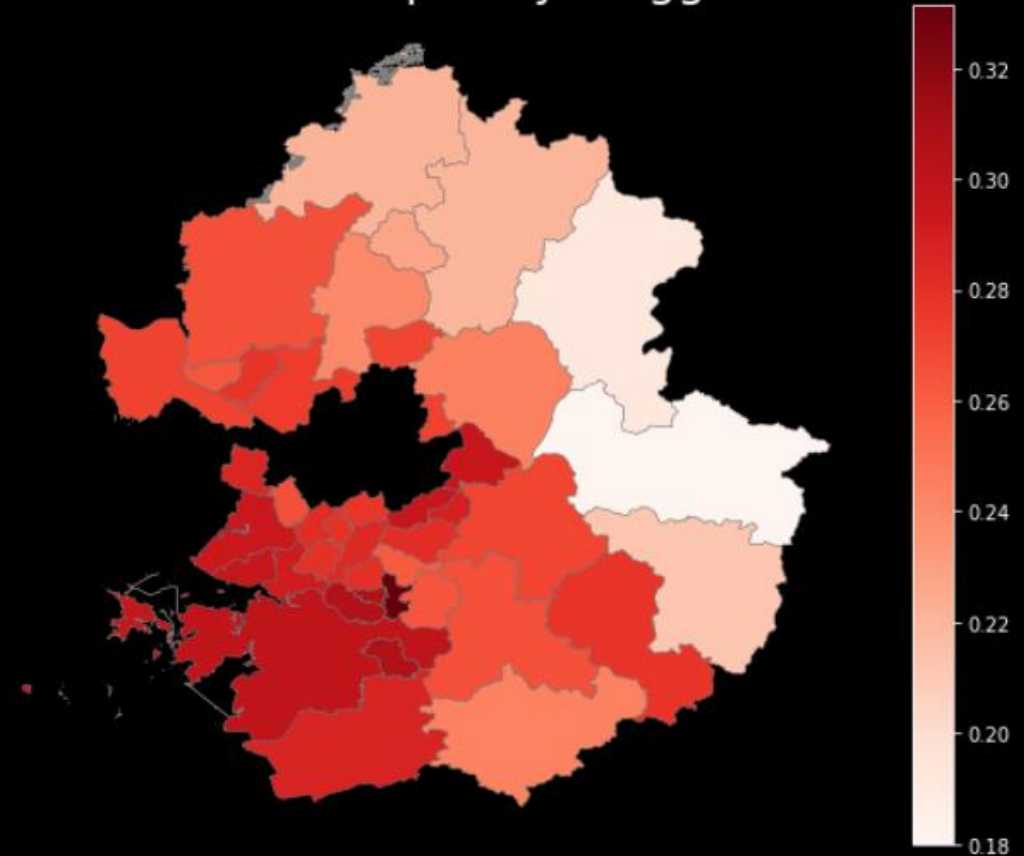
지역 구분	청년인구 비율
용인	0.26
경기도	0.27
전국	0.23



## 1.2 경기도 시·군·구별 청년 인구 비율 시각화

- 전국 시·군·구별 polygon 데이터를 얻기 위해 'TL\_SCCO\_SIG.shp'파일을 불러와 지도에 표시
- 경기도 시·군·구별 인구수와 인구 비율 데이터를 얻기 위해 '경기도만인구비율.csv'파일을 불러옴
- 행정구역 코드를 의미하는 'SIG\_CD'칼럼을 키 값으로 하여 불러온 두 파일을 병합하였다.
- 경기도 시·군·구별 청년인구 비율을 지도에 표시

Youth Ratio Map in Gyeonggi-do

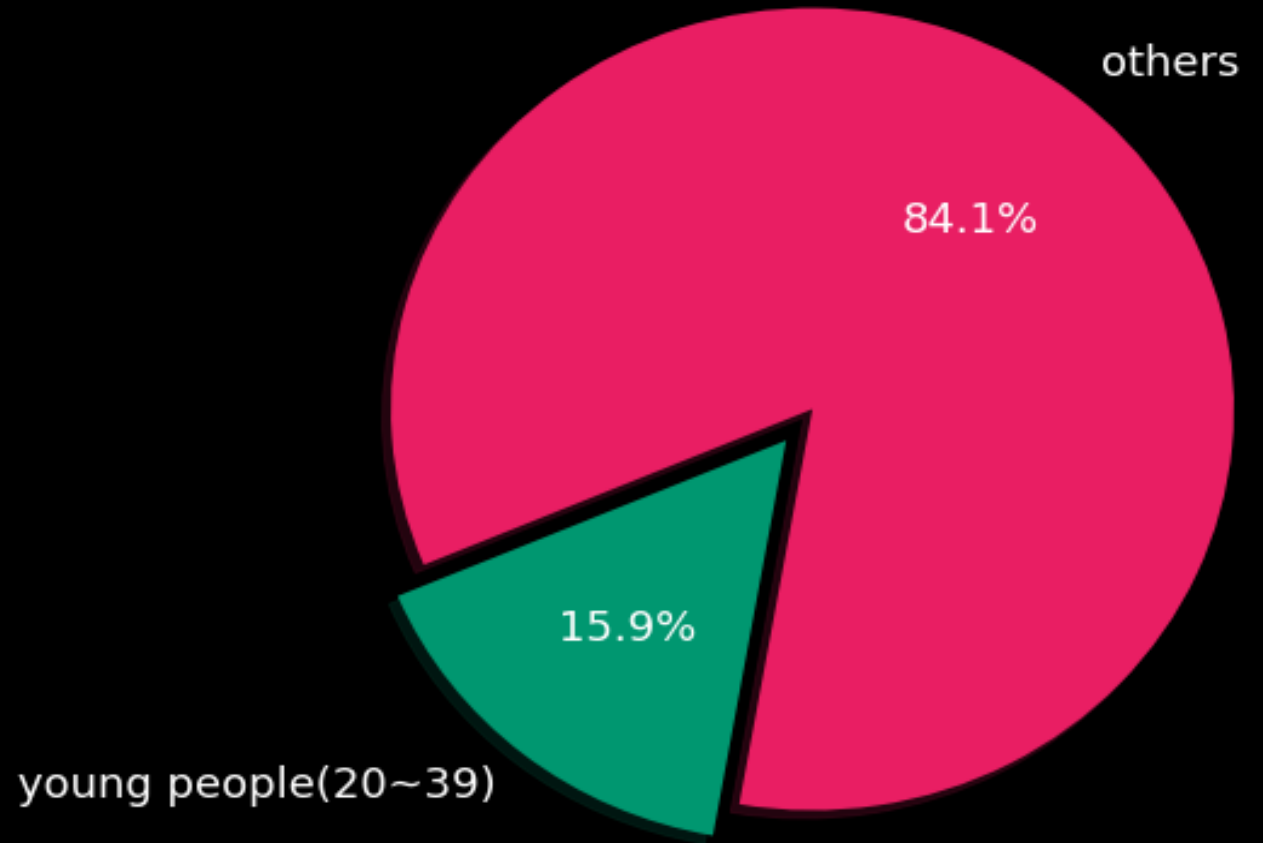




### 1.3 경기도 전체 청년창업자 비율 시각화

- 경기도 각 시의 사업체 대표자 연령이 담겨져 있는 '연령별사업체수.csv'파일을 불러옴
- 총 합을 나타내는 'Total'행을 가져옴
- 청년사업체(대표자가 39세 이하)와 그 외를 나누어서 더함
- 파이 그래프로 시각화
- **경기도의 전체 창업자 중에서 청년이 차지하는 비율이 15.9% 수준이며, 이는 인구비율 27%에 미치지 못하는 수준**

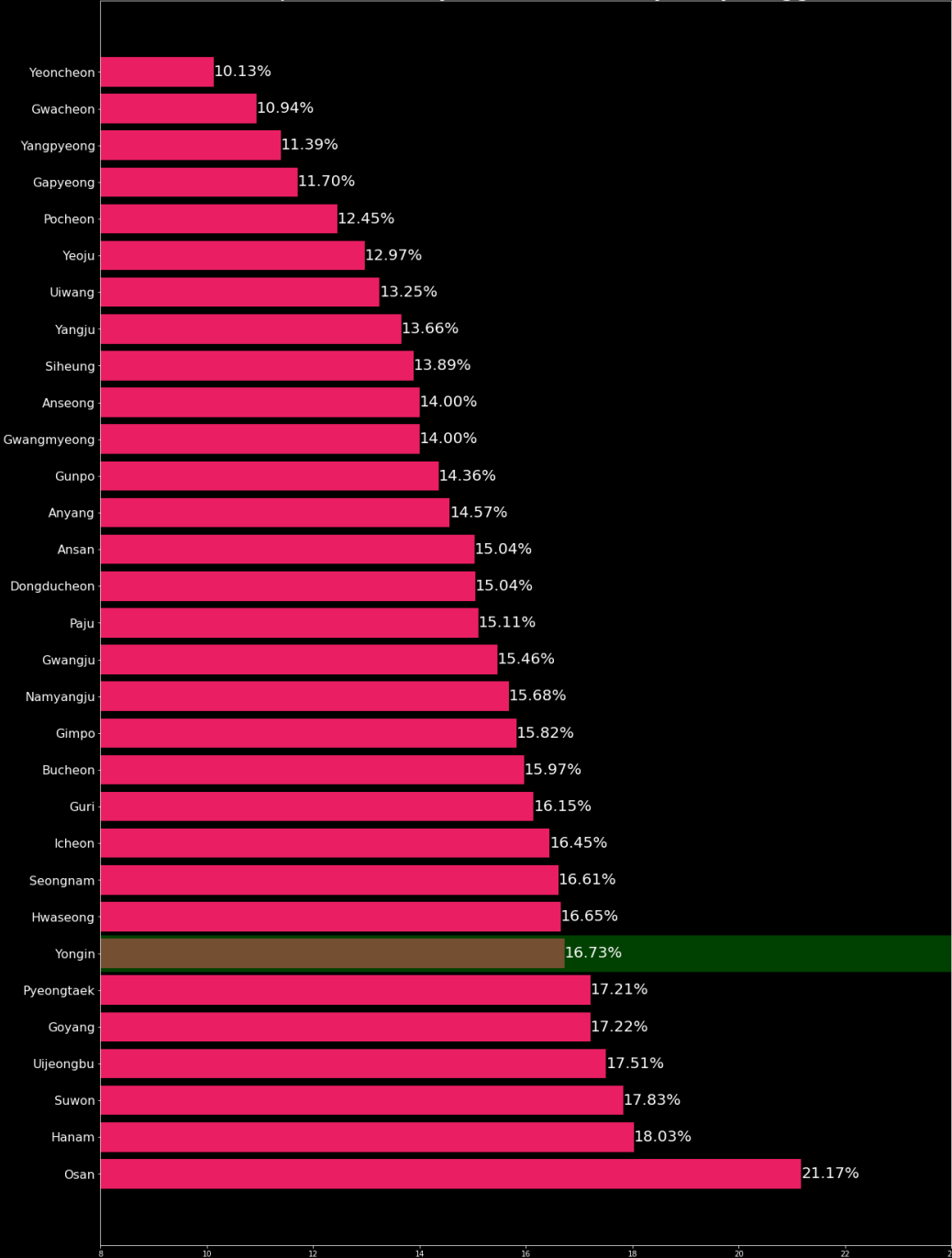
young people Percentage of the total



## 1.4 경기도 각 시·군·구별 청년사업체 비율 시각화

- 경기도 각 행정구역의 사업체 대표자 연령이 담겨져 있는 연령별사업체수.csv파일에서 Total행을 제외
- 퍼센트 비율은 행정구역마다 모든 사업체 중 청년사업체의 비율을 나타냄
- 막대그래프로 표현
- 용인시는 경기도에서 청년 창업자의 비중이 타 시에 비해 높은 수준

Start-up rate of the youth for each city in Gyeonggi-do



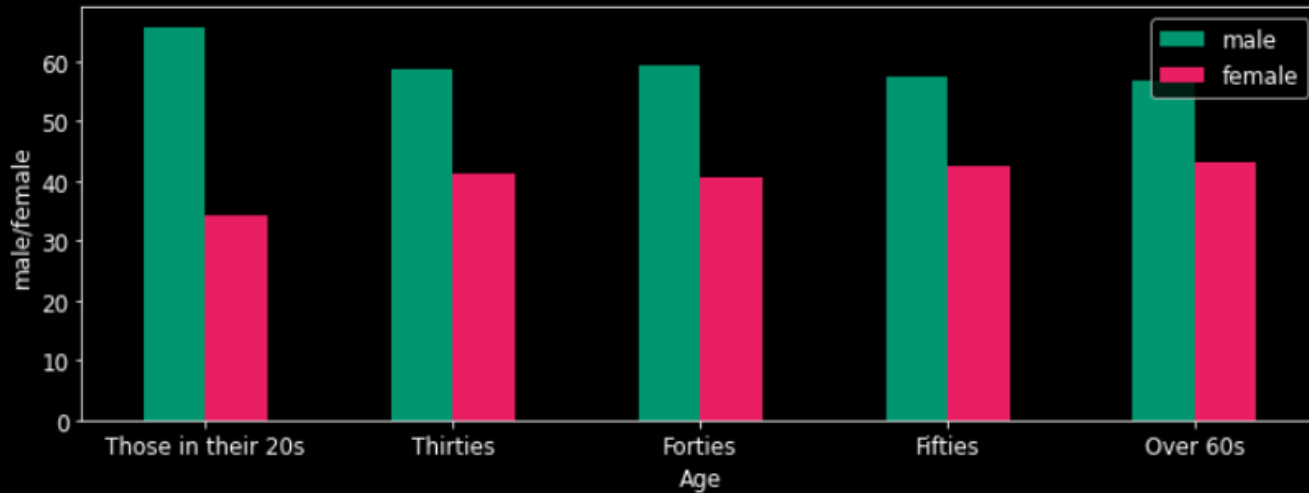
### 1.5 경기도 사업체 업종별 남녀 비율 시각화

- 전국 사업체 대표의 남녀 비율을 담고 있는 '업종별\_남녀비율.csv' 파일을 불러옴
- 구분별이 업종인 칼럼만 선택
- 남성과 여성의 비율을 히트맵으로 시각화
- **요식업, 기타 서비스를 제외한 대부분의 업종에서 남성의 비율이 높음을 확인할 수 있음**



## 1.6 대표자의 나이대별 성별 비율 시각화

- 경기도 사업체 대표의 남녀 비율을 담고 있는 '업종별\_남녀비율.csv' 파일을 불러옴
- '구분별'칼럼이 창업자 연령인 칼럼만 선택
- 창업자 연령별 남성과 여성의 비율을 막대그래프로 시각화



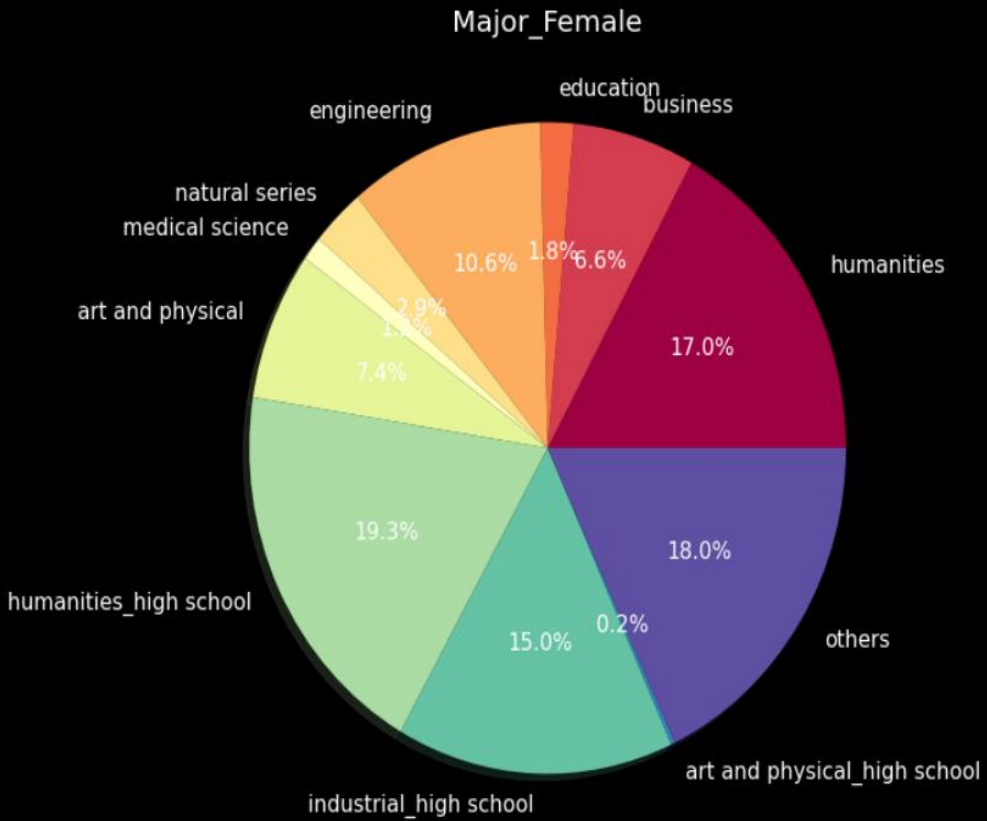
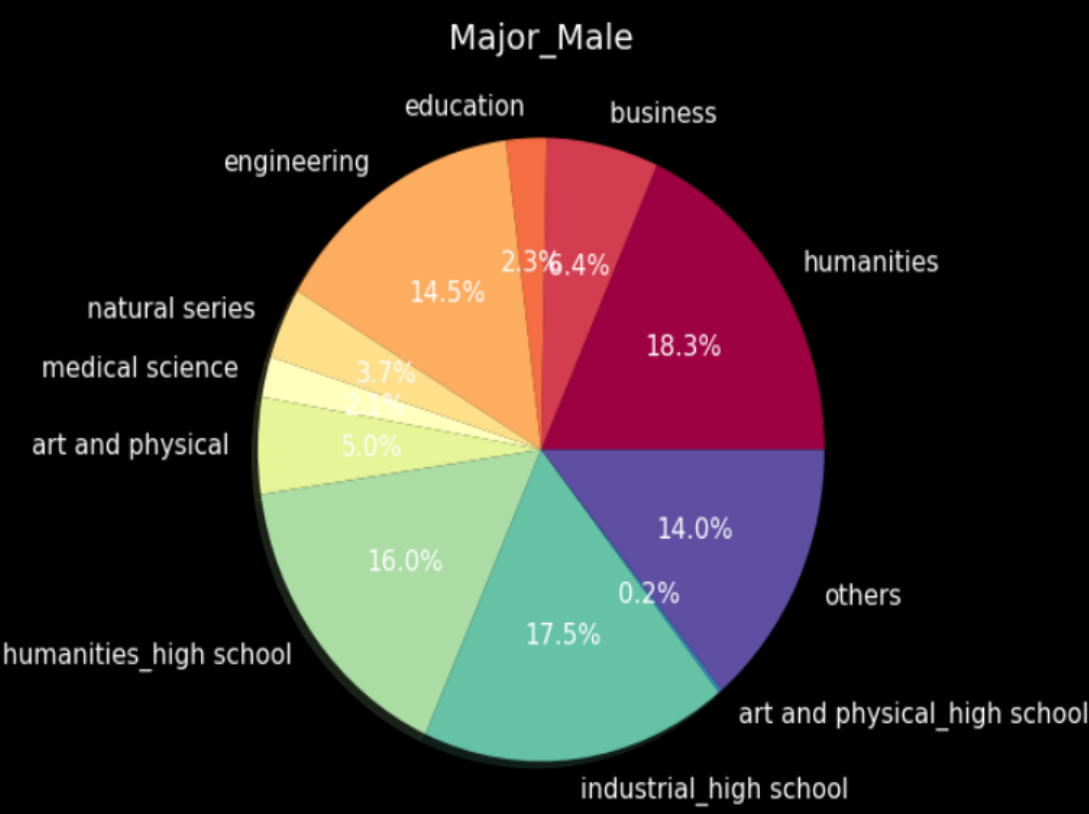
- 모든 연령대와 다양한 분야에서 남성이 많이 창업함을 알 수 있음

## 1.7 남녀 창업자별 전공 비율 시각화

- 남성 창업자의 전공 비율을 담고 있는 '전공별창업\_male.csv' 파일과 여성 창업자의 전공 비율을 담고 있는 '전공별창업\_female.csv' 을 불러옴
- 각 데이터를 파이차트로 시각화
- 각각의 칼럼이 의미하는 전공은 다음과 같음

- humanities: 인문,사회 계열
- business: 상경제열
- education: 교육계열
- engineering: 공학계열
- natural series: 자연계열
- medical science: 의학계열
- art and physical: 예체능계열
- humanities\_high school: 인문계(고졸)
- industrial\_high school: 실업계(고졸)
- art and physical\_high school: 예체능(고졸)
- others: 나머지 계열 + 중졸 이하

1.7 남녀 창업자별 전공 비율 시각화

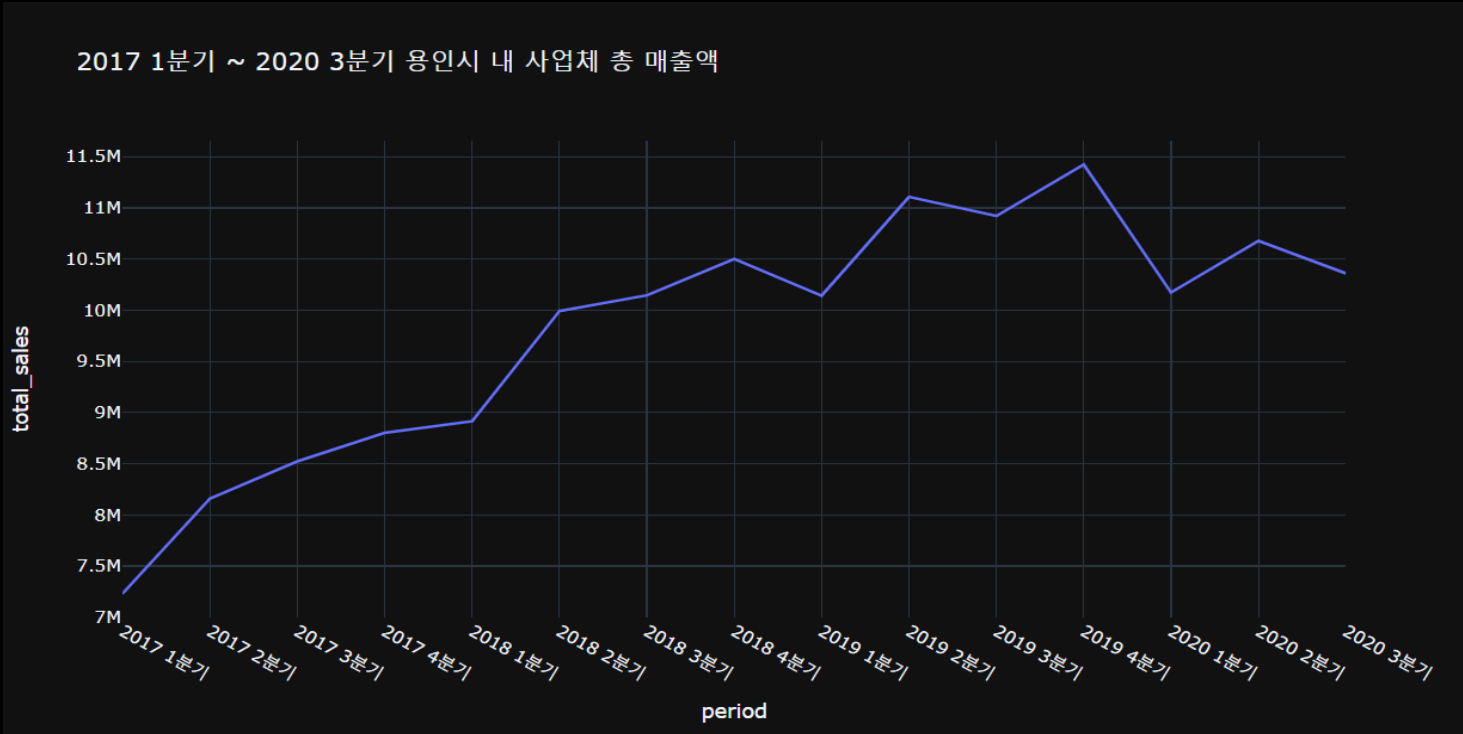


- 두 성별 모두 비슷한 경향을 보였으나 공학(engineering)에서 남성이 14.5%, 여성이 10.6%로 차이를 보임
- 예체능 계열 졸업자의 창업이 잘 이루어지지 않고 있음

## 시각화(2) 용인시 상권 정보 시각화

## 2.1 2017 1분기 ~ 2020 3분기 용인시 내 사업체 총 매출액 시각화

- 2017 1분기 ~ 2020 3분기 용인시 내 사업체 매출 정보를 담고 있는 '17.용인시\_소상공인\_매출정보.csv' 파일을 불러옴
- 매출정보를 담고있는 칼럼만 추출하여 용인시 내 사업체 총 매출액을 구함
- 꺾은선 그래프로 시각화

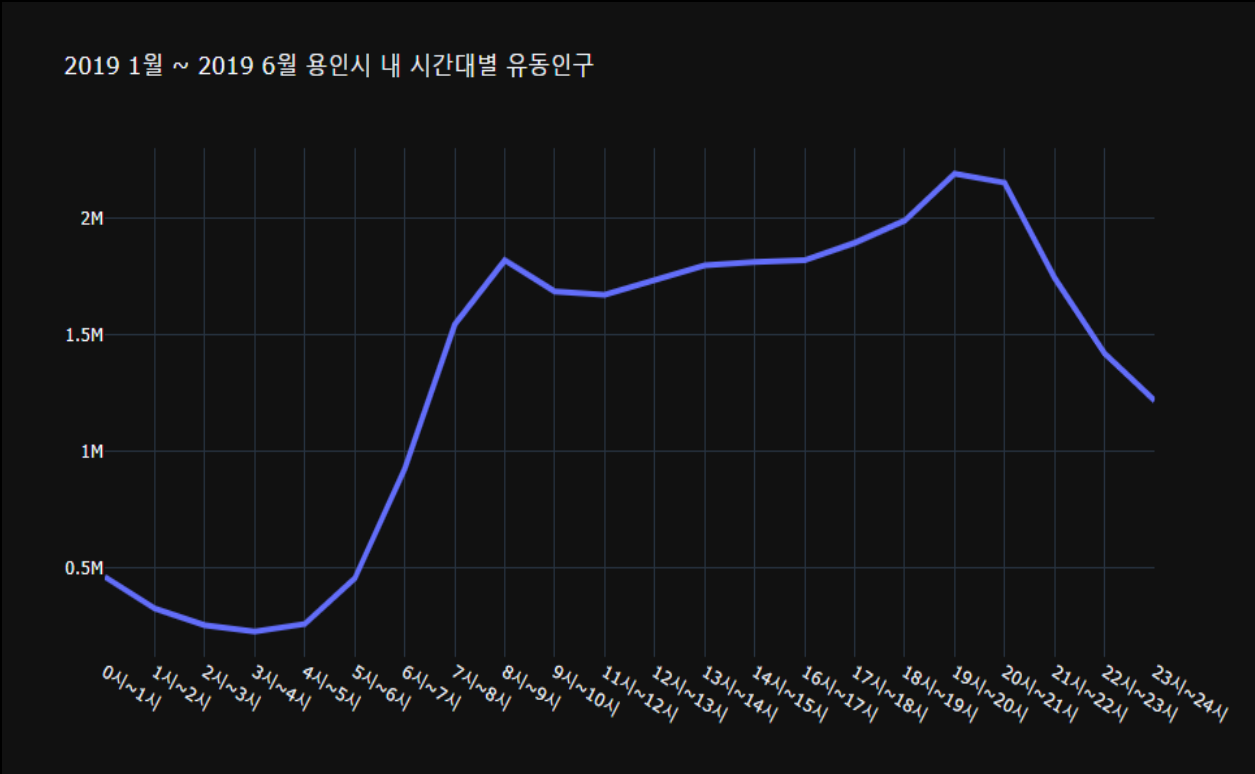


- 용인시의 전체 매출은 17년 1분기부터 성장세를 보였으나, 코로나19에 의해 19년 4분기부터 하락한 것으로 보인다.



## 2.2 유동인구 정보 시각화(1)

- 용인시 유동인구에 대한 정보를 얻기 위해 '7.용인시\_유동인구.csv'파일을 불러옴
- 시간대별로 유동인구의 총 합을 구함
- 위에서 얻은 시간대별 유동인구의 총 합을 꺾은선그래프로 시각화

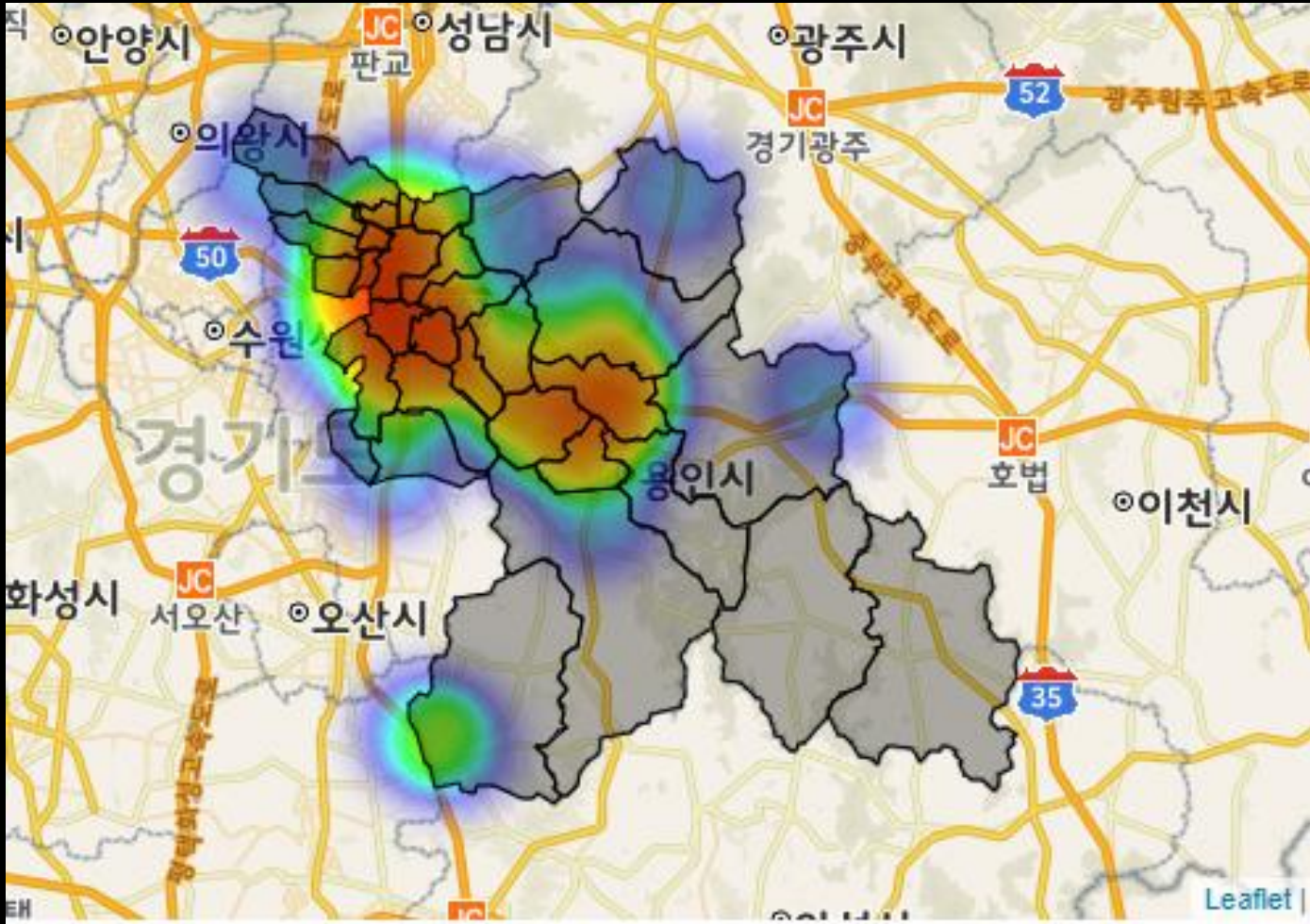


- 유동인구를 시간대별로 확인하고자 하였으며 출근시간, 퇴근시간에 그래프의 변화율이 가장 높은 것을 확인할 수 있다.
- 자정 ~ 7시 까지는 용인시 내 유동인구가 비교적 적기 때문에 시각화 부분에서는 이 부분을 다루지 않기로 결정

## 2.3 유동인구 정보 시각화(2)

- 용인시 유동인구에 대한 정보를 가지고 있는 '7.용인시\_유동인구.csv'파일에서 측정장소가 동일한 row(위도, 경도가 같은 row)를 하나로 합쳐 평균치를 구함
- 위도&경도 데이터를 바탕으로 취합(groupby)을 진행하기 위해 'lat'칼럼과 'lon'칼럼을 문자열로 바꿔 하나로 합침. 위도경도의 데이터를 담고 있는 합쳐진 칼럼을 바탕으로 취합을 진행
- 용인시 읍·면·동 경계에 대한 정보를 담고 있는 '13.용인시\_행정경계(읍면동).geojson' 파일을 불러와 지도에 읍·면·동 경계를 표시
- 유동인구가 80이상인 곳만 추출하여 지도에 히트맵으로 시각화

## 2.3 유동인구 정보 시각화(2)



- 기흥구와 수지구에 유동인구가 많은 것을 확인할 수 있다.
- 유동인구가 많은 곳에 활성화될 수 있는 업종이라면 수지구와 기흥구에 창업하는 것이 도움이 될 것이다.

## 2.4 용인시 업종별 사업체 개수 시각화

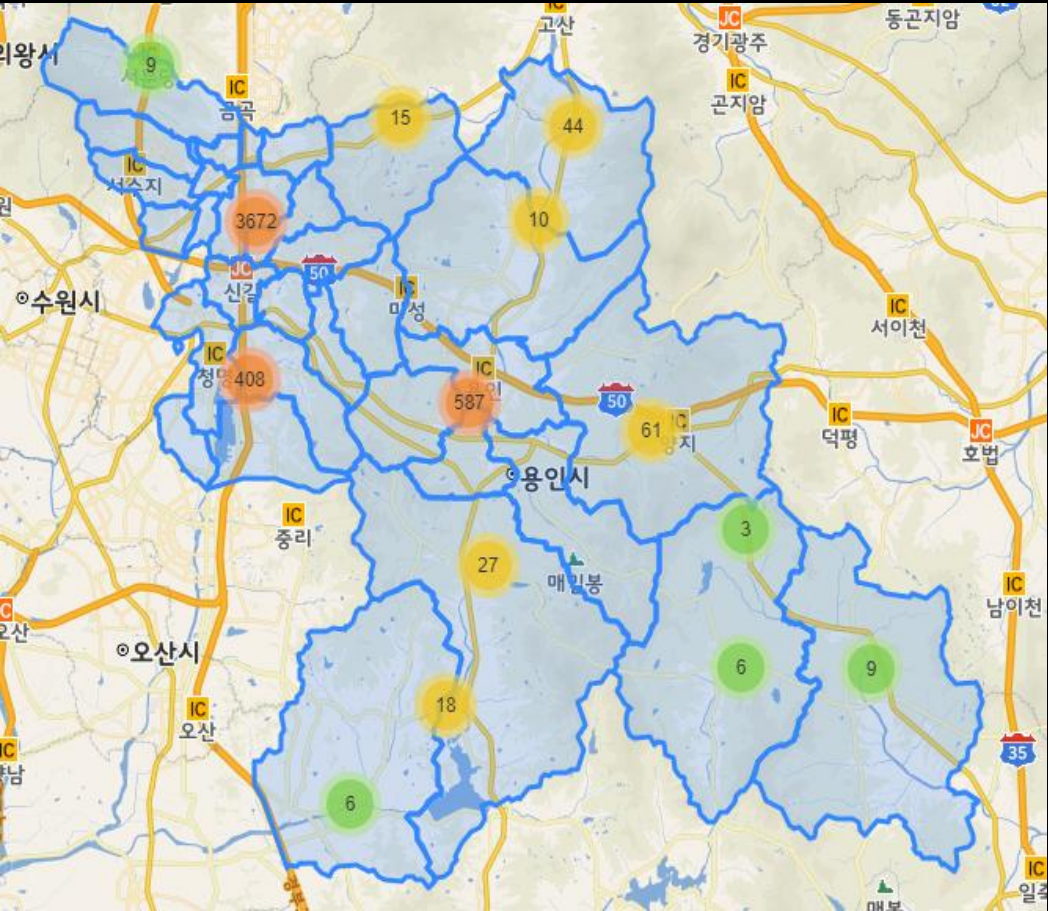
- 용인시 읍·면·동 경계에 대한 정보를 담고 있는 '13.용인시\_행정경계(읍면동).geojson' 파일을 불러와 지도에 읍·면·동 경계를 표시
- 용인시 "1.용인시\_상권\_정보.csv" 데이터에 있는 모든 사업체들을 대분류코드를 기준으로 업종별로 나눠서 개수를 지도에 표시
- "1.용인시\_상권\_정보.csv" 데이터에 존재하는 대분류코드는 다음과 같음('2.용인시\_상권\_업종코드.csv' 참조)

D: 소매업  
F: 생활서비스  
L: 부동산  
N: 관광/여가/오락  
O: 숙박  
P: 스포츠  
Q: 음식  
R: 학문/교육



## 2.4 용인시 업종별 사업체 개수 시각화 설명

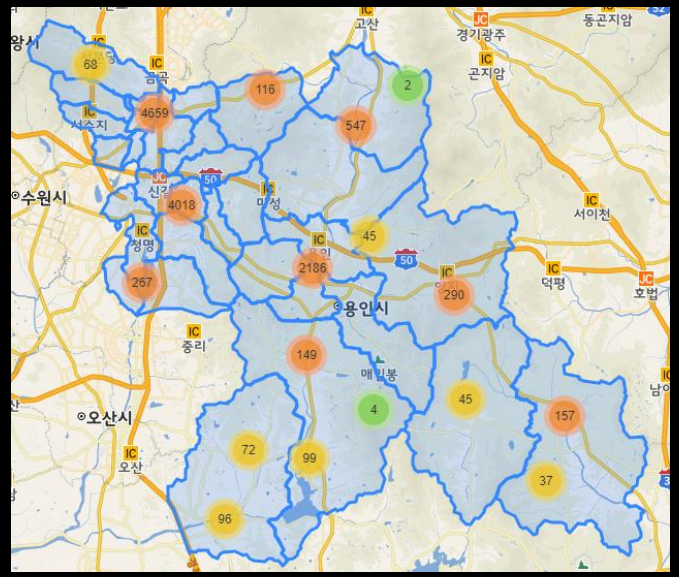
R : 학문/교육



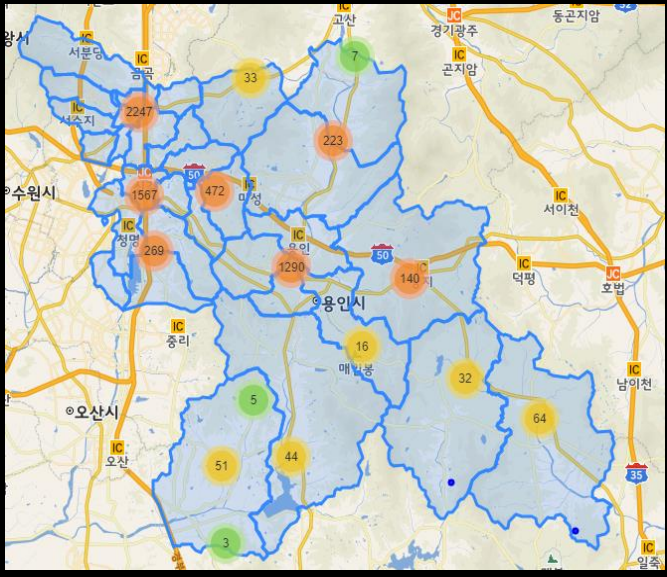
- "1.용인시\_상권\_정보.csv" 에 존재하는 상권들 중 대분류코드 하나를 정해 해당 상권들만 지도에 표시
- 대분류 코드별로 지도를 그림
- 지도의 숫자는 그 지역에 해당 대분류코드의 업종의 수를 의미함
- 왼쪽 그래프의 경우 학문/교육 업종을 가지고 있는 상권들을 용인시 지도에 표현
- 유동인구가 많았던 기흥구와 수지구를 중심으로 여러 업종이 발달한 것을 알 수 있다.

# 2.4 용인시 업종별 사업체 개수

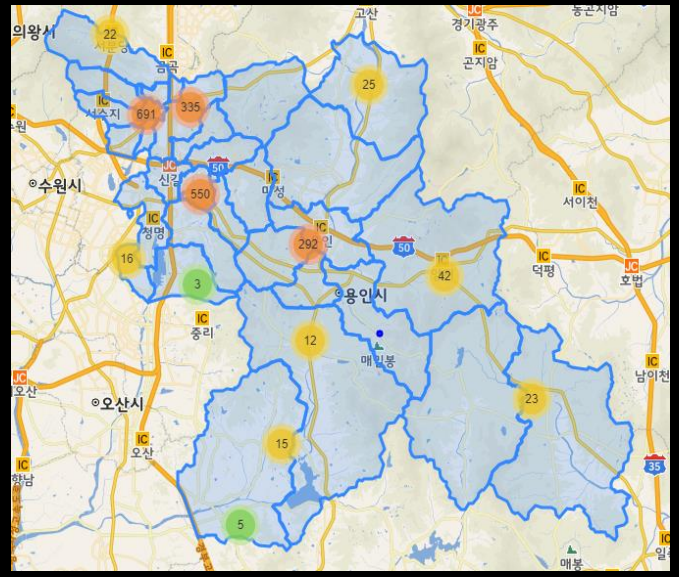
D : 소매업



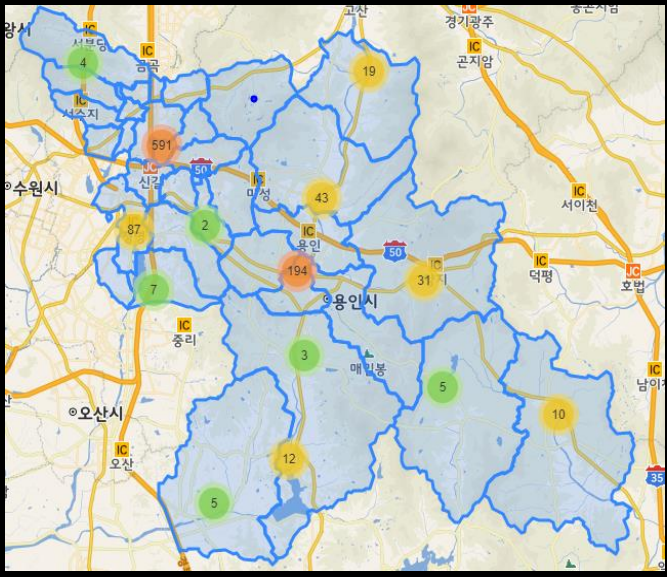
F : 생활서비스



L : 부동산

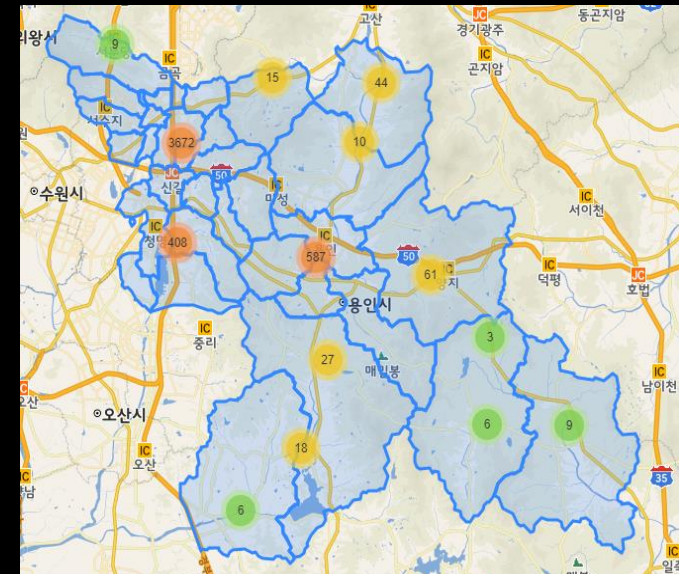
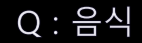
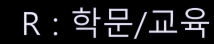


N : 관광/여가/오락





○ : 숙박업

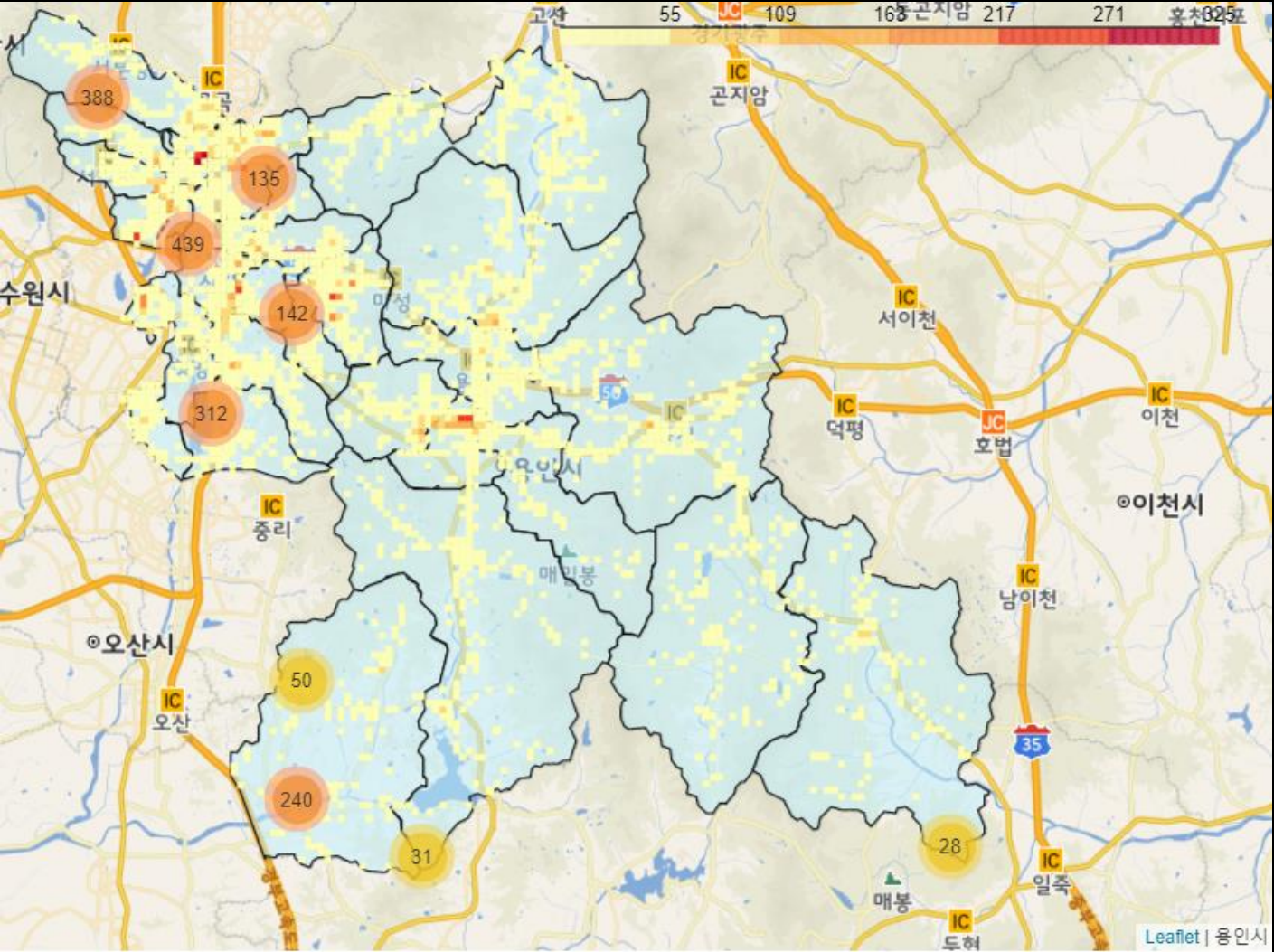


## 2.5 용인시 사업체 위치와 매출액이 높은 사업체 수 시각화

- 용인시 읍·면·동 경계에 대한 정보를 담고 있는 '13.용인시\_행정경계(읍면동).geojson' 파일을 불러와 지도에 읍·면·동 경계를 표시하여 용인시 지도를 표현
- 용인시 '17.용인시\_소상공인\_매출정보.csv' 를 사용하여 전체 사업장 수 정보를 가지고 있는 'ws\_cnt' 컬럼을 위의 지도의 표시
- 용인시 '16.용인시\_소상공인\_매출정보.geojson'에 있는 매출 데이터 'sales\_est\_amt\_201703'부터 'sales\_est\_amt\_202009'를 모두 합친 후 값이 1000이상인 격자의 수를 지도에 표현
- 결과적으로 용인시 내의 사업체를 가지고 있는 격자를 모두 표현하고 그 위에 매출액이 높은 사업체의 수를 표현
- 지도의 숫자는 매출액이 높은 사업체의 수를 표현
- 격자 내의 사업체 수에 따라 색상을 다르게 지도에 표현



2.5 용인시 사업체 위치와 매출액이 높은 사업체 수 시각화



- 용인시의 상권은 기흥구, 수지구  
를 중심으로 발달했음을 알 수 있  
다.
- 수지구와 기흥구는 용인에서 가장  
개발이 많이 진행된 곳으로 많은  
주거지역, 학군이 형성되어 상권  
역시 발달한 것으로 보인다.

## 시각화 (1) – 청년창업 실태 분석

- 현재 청년 창업인구는 전체 창업인구의 16%정도의 수준이다. 경기도 인구에서 청년(20~39)은 전체의 25%에 달하는 반면 창업인구비율은 적은 수준이다.
- 경기도의 창업형태로 보아 성별과 전공에 불균형이 존재하는 것을 알 수 있다.
- 따라서 청년 창업의 다양한 지표에 불균형이 존재한다는 사실을 인지하여야 하고, 이러한 불균형들을 정책적으로 해결할 수 있어야 한다.

## 시각화 (2) – 용인시 상권분석

- 용인시 매출정보를 통해 코로나19의 여파로 19년 4분기부터 매출이 급락했음을 알 수 있다.
- 기흥구, 수지구를 중심으로 많은 유동인구와 상권이 발달해 있으므로 이러한 정보들은 청년들이 창업하는데 좋은 정보로 활용될 수 있을 것이다.

# 분석 방법 및 전처리

## 분석 대상, 목적 및 모델링

- 대상 : 17.용인시\_소상공인\_매출정보.csv
- Gid로 구별된 격자 내의 정보를 담고 있는 데이터로 66개의 열과 2257개의 행의 매출 정보 및 격자 내 상권의 정보에 대해 담고 있어 회귀분석, 상관분석, 주성분분석 등에 활용이 가능하고, 다양한 인사이트의 도출이 가능하다.
- 분석방법은 '회귀분석', '상관분석' 등을 사용하였으며 회귀분석을 통해 선정한 종속변수들에 영향을 미친 독립변수를 가려내어 제안에 도움을 주고자 하였고, 보다 나은 수준의 상관분석을 위해 필요하다고 판단하였다.
- 종속변수별 회귀분석을 통해 가려낸 독립변수 간의 상관관계를 파악해 제안을 보다 효과적으로 완성하고자 하였다.
- 선정한 종속변수(mean\_sales, found\_age\_6, sme\_loan\_cnt, runout\_cnt)와 독립변수 간의 관계를 통해 각 종속변수에 영향을 미친 변수를 파악하려 하였으나 mean\_sales 분석 외에는 인사이트를 도출하기 어려웠다.

## 전처리

- 1) indcd\_a\_yn~ indcd\_u\_yn 열(격자내 특정 상권 존재 여부) 중 '2.용인시\_상권\_업종코드.csv' 파일에서 정보를 가지고 있지 않은 열은 삭제하였으며, NaN과 Y로 나뉘어진 열 내의 데이터를 0과 1로 바꾸어 분석에 사용 가능하도록 전처리하였다.
- 2) 각 격자 내의 분기별 매출액 정보를 담고 있는 sales\_est\_amt\_201703~sales\_est\_amt\_202009의 데이터를 mean\_sales라는 새로운 칼럼을 만들고 각 분기별 매출을 격자별로 평균내었다.
- 3) 사용할 각 종속변수의 분포도를 확인해 정규분포 형태를 띄지 않는 경우에는 로그변환을 통해 첨도를 줄이는 등 정규분포에 가깝게 변환시키고자 하였으며, 각 독립변수는 파이썬 패키지 중 minmaxscaler를 통해 0~1 사이의 값을 가지도록 스케일링하여 사용하였다.
- 4) 독립변수는 종속변수와 gid칼럼을 제외한 64개를 사용하였으며 모든 독립변수를 사용하는 것은 회귀식의 완성도를 떨어뜨리게 되므로 단계적 회귀분석을 위한 함수를 제작하였다.
- 5) train\_set과 test\_set으로 분리하여 분석하였으며, 비율은 (7:3)이다.

## 단계적 회귀분석(Forward, Backward, Stepwise)

변수가 많을 때 모든 조합을 계산하지 않고 합리적으로 가장 좋은 조합을 찾는다.

- 1) Forward(전진선택법): 변수 하나부터 변수 개수를 추가해가며 성능지표를 비교해가는 방법
- 2) Backward(후진소거법): 전체 변수부터 변수 개수를 제거해가며 성능지표를 비교해가는 방법
- 3) Stepwise(단계적선택법):
  - 처음엔 변수 하나부터 시작해서 변수 개수를 늘려가지만, 하나씩 변수를 추가하다 보면 중요하지 않는 변수가 들어올 수도 있음
  - 이 문제를 해결하기 위해 각 단계에서 변수의 중요성을 체크하고 중요하지 않는 변수를 제거
  - 제거 후 다시 다른 변수를 추가하는 식으로 나아가면서 성능지표를 비교하는 방법

# 사용 함수 정리



## Forward(전진선택법):

```
#전진선택법(step=1)
def foward(X, y, predictors):
    # 데이터 변수들이 미리정의된 predictors에 있는지 없는지 확인 및 분류
    remaining_predictors = [p for p in X.columns.difference(['const']) if p not in predictors]
    tic = time.time()
    results = []
    for p in remaining_predictors:
        results.append(processSubset(X=X, y=y, feature_set=predictors+[p]+['const']))
    #데이터프레임으로 변환
    models = pd.DataFrame(results)

    #AIC가 가장 낮은 것을 선택
    best_model = models.loc[models['AIC'].argmin()] #index
    toc = time.time()
    print("Processed ", models.shape[0], "models on", len(predictors)+1, "predictors in", (toc-tic))
    print('Selected predictors:', best_model['model'].model.exog_names, ' AIC:',best_model[0] )
    return best_model
```



## Forward(전진선택법):

### #전진선택법 모델

```
def forward_model(X,y):
    Fmodels = pd.DataFrame(columns=["AIC","model"])
    tic = time.time()
    #미리 정의된 데이터 변수
    predictors = []
    # 변수 1~10개 : 0~9 -> 1~10
    for i in range(1, len(X.columns.difference(['const']))+1):
        Forward_result = foward(X=X, y=y, predictors=predictors)
        if i > 1:
            if Forward_result['AIC'] > Fmodel_before:
                break
            Fmodels.loc[i] = Forward_result
            predictors = Fmodels.loc[i]["model"].model.exog_names
            Fmodel_before = Fmodels.loc[i]["AIC"]
            predictors = [ k for k in predictors if k != 'const']
    toc = time.time()
    print("Total elapsed time:", (toc - tic), "seconds.")

    return(Fmodels['model'][len(Fmodels['model'])])
```

## Backward(후진소거법):

#후진소거법(step=1)

```
def backward(X,y,predictors):
```

```
    tic = time.time()
```

```
    results = []
```

```
    # 데이터 변수들이 미리정의된 predictors 조합 확인
```

```
    for combo in itertools.combinations(predictors, len(predictors) - 1):
```

```
        results.append(processSubset(X=X, y=y,feature_set=list(combo)+['const']))
```

```
    models = pd.DataFrame(results)
```

```
    # 가장 낮은 AIC를 가진 모델을 선택
```

```
    best_model = models.loc[models['AIC'].argmin()]
```

```
    toc = time.time()
```

```
    print("Processed ", models.shape[0], "models on", len(predictors) - 1, "predictors in", (toc - tic))
```

```
    print('Selected predictors:', best_model['model'].model.exog_names, ' AIC:',best_model[0] )
```

```
    return best_model
```

## Backward(후진소거법):

### # 후진 소거법 모델

```
def backward_model(X, y):
    Bmodels = pd.DataFrame(columns=["AIC", "model"], index = range(1,len(X.columns)))
    tic = time.time()
    predictors = X.columns.difference(['const'])
    Bmodel_before = processSubset(X,y,predictors)['AIC']
    while(len(predictors)>1):
        Backward_result = backward(X=train_x, y=train_y, predictors = predictors)
        if Backward_result['AIC'] > Bmodel_before:
            break
        Bmodels.loc[len(predictors)-1] = Backward_result
        predictors = Bmodels.loc[len(predictors)-1]["model"].model.exog_names
        Bmodel_before = Backward_result['AIC']
        predictors = [k for k in predictors if k != 'const']

    toc = time.time()
    print("Total elapsed time:", (toc - tic), "seconds.")
    return (Bmodels['model'].dropna().iloc[0])
```

## Stepwise(단계적선택법):

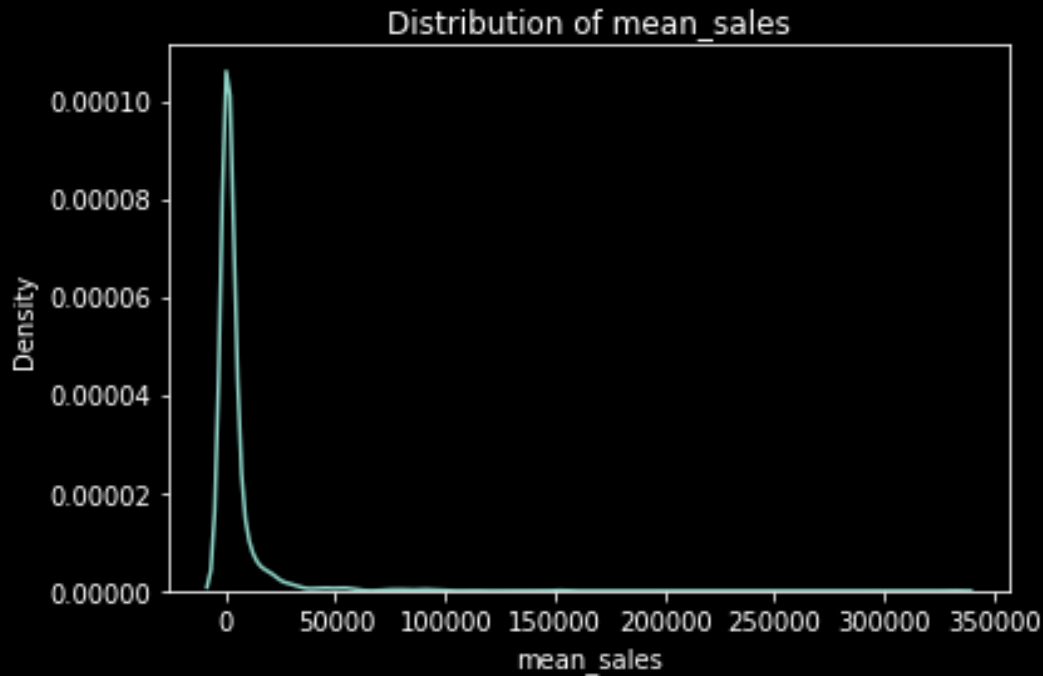
```
def Stepwise_model(X,y):
    Stepmodels = pd.DataFrame(columns=["AIC","model"])
    tic = time.time()
    predictors = []
    Smodel_before = processSubset(X,y,predictors+['const'])['AIC']
    #변수 1~10개 : 0~9 -> 1~10
    for i in range(1, len(X.columns.difference(['const']))+1):
        Forward_result = foward(X=X, y=y, predictors=predictors) # constant added
        print('foward')
        Stepmodels.loc[i] = Forward_result
        predictors = Stepmodels.loc[i]["model"].model.exog_names
        predictors = [k for k in predictors if k != 'const']
        Backward_result = backward(X=X, y=y, predictors=predictors)
        if Backward_result['AIC'] < Forward_result['AIC']:
            Stepmodels.loc[i] = Backward_result
            predictors = Stepmodels.loc[i]["model"].model.exog_names
            Smodel_before = Stepmodels.loc[i]["AIC"]
            predictors = [k for k in predictors if k != 'const']
            print('backward')
        if Stepmodels.loc[i]['AIC'] > Smodel_before:
            break
        else:
            Smodel_before = Stepmodels.loc[i]["AIC"]
    toc = time.time()
    print("Total elapsed time:", (toc-tic), "seconds.")
    return(Stepmodels['model'][len(Stepmodels['model'])])
```

1. 전진선택법에서 사용한 forward 함수로 변수를 하나씩 추가하며 시작
2. 후진소거법에서 사용한 backward 함수로 불필요한 변수를 제거
3. Forward와 backward가 반복되며, 더 이상 추가하거나 제거할 것이 없으면 종료

**가장 효율적인 방법으로 판단**

# 분석 결과 및 해석

## 데이터 분석(1) – 회귀분석; 종속변수 : mean\_sales



- 1) 종속변수로 사용할 mean\_sales의 경우 초기 분포의 형태가 정규분포와 거리가 멀게 나왔으므로 로그변환을 통해 종속변수의 분포를 확인하였다.
- 2) 독립변수는 MinMaxScaler()를 통해 스케일링하였다.

데이터 분석(1) – 회귀분석; 종속변수 : mean\_sales

3) 단계적 회귀분석 과정 중 빠른 속도와 효율을 가지고 있는 Stepwise 방법으로 회귀분석한 결과와 다른 방법과의 성능 비교 결과는 아래와 같다.

OLS Regression Results						
Dep. Variable:	mean_sales	R-squared:	0.378			
Model:	OLS	Adj. R-squared:	0.374			
Method:	Least Squares	F-statistic:	95.45			
Date:	Thu, 28 Jan 2021	Prob (F-statistic):	4.25e-154			
Time:	13:24:06	Log-Likelihood:	729.07			
No. Observations:	1579	AIC:	-1436.			
Df Residuals:	1568	BIC:	-1377.			
Df Model:	10					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
ws_cnt	3.6207	0.659	5.491	0.000	2.327	4.914
indcd_p_yn	0.1149	0.011	10.360	0.000	0.093	0.137
indcd_r_yn	0.0938	0.012	7.638	0.000	0.070	0.118
indcd_q_yn	0.0882	0.015	6.037	0.000	0.060	0.117
smbiz_yn_cnt	-1.5965	0.313	-5.096	0.000	-2.211	-0.982
rpr_per_age_bin_30to50	-1.4710	0.510	-2.883	0.004	-2.472	-0.470
age10_ratio	0.2743	0.105	2.618	0.009	0.069	0.480
found_age_2	-0.3738	0.150	-2.493	0.013	-0.668	-0.080
sme_loan_y_3	0.2004	0.112	1.781	0.075	-0.020	0.421
sme_loan_y_1	0.1322	0.086	1.536	0.125	-0.037	0.301
indcd_o_yn	0	0	nan	nan	0	0
const	0.4201	0.005	86.399	0.000	0.411	0.430
Omnibus:	33.943	Durbin-Watson:	2.031			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	35.839			
Skew:	-0.368	Prob(JB):	1.65e-08			
Kurtosis:	2.959	Cond. No.	1.33e+20			

4) 그러나 stepwise 방법으로도 의미가 중복되는 변수를 제거하지 못하는 등의 한계는 존재하게 된다.  
따라서 추출된 독립변수 중에서 의미를 가지는 변수를 추출하여 새로 회귀분석하는 과정이 필요하다.

	ALL	FORWARD	BACKWARD	STEPWISE
MSE	0.0238149	0.023641	0.023605	0.023641
RMSE	0.154321	0.153756	0.153639	0.153756
MAE	0.124646	0.124353	0.124016	0.124353

데이터 분석(1) – 회귀분석; 종속변수 : mean\_sales

OLS Regression Results						
Dep. Variable:	mean_sales	R-squared (uncentered):	0.449			
Model:	OLS	Adj. R-squared (uncentered):	0.446			
Method:	Least Squares	F-statistic:	142.4			
Date:	Thu, 28 Jan 2021	Prob (F-statistic):	2.53e-196			
Time:	13:24:06	Log-Likelihood:	-732.11			
No. Observations:	1579	AIC:	1482.			
Df Residuals:	1570	BIC:	1531.			
Df Model:	9					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
indcd_p_yn	0.4073	0.026	15.429	0.000	0.356	0.459
indcd_r_yn	0.2439	0.033	7.496	0.000	0.180	0.308
indcd_q_yn	0.1722	0.037	4.602	0.000	0.099	0.246
sme_loan_y_6	1.1843	0.324	3.658	0.000	0.549	1.819
indcd_f_yn	0.1041	0.059	1.763	0.078	-0.012	0.220
smbiz_yn_cnt	-2.1537	0.466	-4.626	0.000	-3.067	-1.241
rpr_per_age_bin_20	0.4271	0.256	1.665	0.096	-0.076	0.930
sme_loan_y_3	0.7103	0.296	2.399	0.017	0.130	1.291
rpr_per_age_bin_50over	2.9788	0.474	6.281	0.000	2.049	3.909
Omnibus:	288.504	Durbin-Watson:	0.777			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	724.898			
Skew:	-0.984	Prob(JB):	3.89e-158			
Kurtosis:	5.673	Cond. No.	40.0			

5) 변수를 다시 걸러내어 회귀분석한 결과는 다음과 같다.

6) 이 결과를 통해 격자별 매출에 영향을 미친 독립변수를 조사할 수 있었다.

(업종 p : 스포츠, 업종 r : 학문/ 교육, 업종 q : 음식, 업종 f : 생활서비스)의 경우 격자별 매출을 상승시킨 변수로 파악되고, 의외로 격자별 소상공인 수는 격자의 매출과 음의 상관관계를 보여 소상공인의 매출이 저조한 상황임을 알 수 있다.

또한 rpr\_per\_age\_bin\_(20,50)(카드 이용자 중 20대, 50대의 비율) 이 높은 구역일수록 매출이 높다는 해석을 할 수 있다.



## 분석 결과

- 격자 별 대출액의 평균값을 종속변수로 하여 나온 최종 분석결과는 크게 두가지로 나눌 수 있었다.

### 1. 카드 사용자 비율

20대와 50대 이상 카드사용자 비율이 대출액 변수와 양의 상관관계를 가짐

청년들이 창업을 할 시 20대와 50대이상을 주 고객 타겟층으로 설정 권장

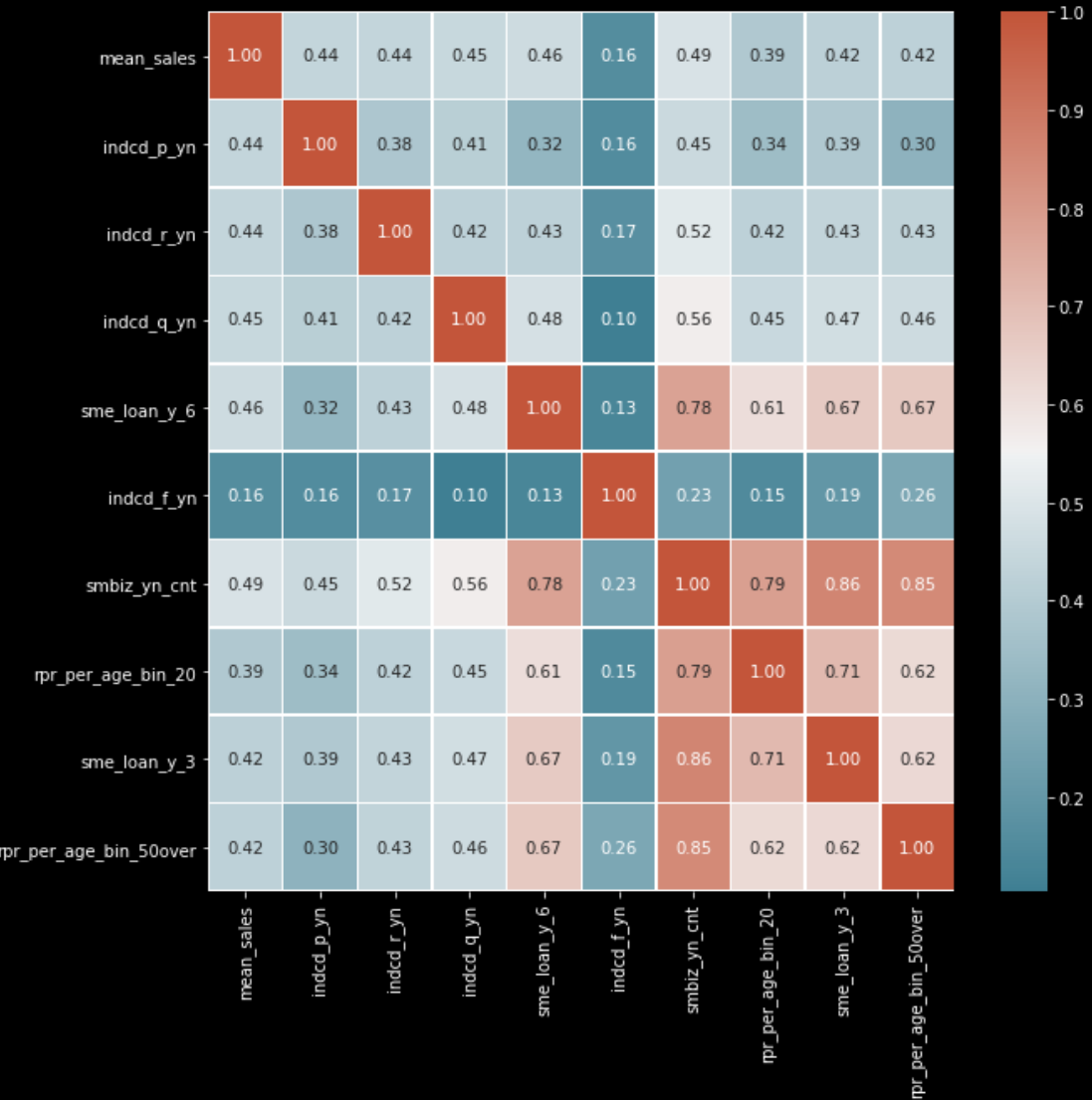
### 2. 대출액

(1억 초과 ~ 2억 이하), (10억 초과) 변수가 대출액 변수와 양의 상관관계

대출액 부분은 용인시의 지원 예산과 관련된 문제임으로 주로 다루지 않음

데이터 분석(2) – 상관분석; 변수

('mean\_sales','indcd\_p\_yn','indcd\_r\_yn','indcd\_q\_yn','sme\_loan\_y\_6','indcd\_f\_yn',  
'smbiz\_yn\_cnt', 'rpr\_per\_age\_bin\_20', 'sme\_loan\_y\_3', 'rpr\_per\_age\_bin\_50over')

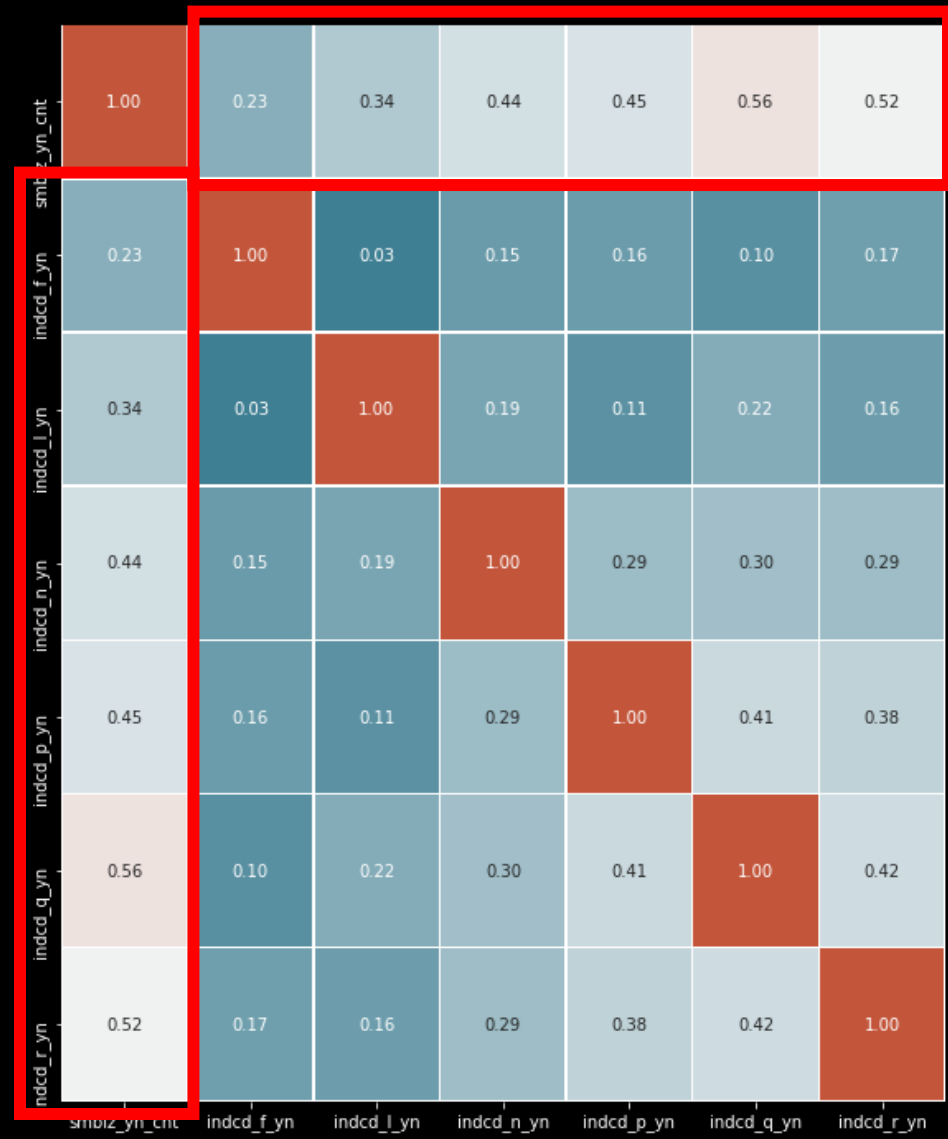


7) 최종적으로 사용한 독립변수의 상관분석을 통해 종속 변수(격자별 매출 평균)과 음의 상관관계를 가지는 격자 내 소상공인 수(simbiz\_yn\_cnt)와 다른 독립변수간의 상관관계를 알아보고자 하였다.

8) 격자 내 소상공인 수는 격자 내 카드사용 인구(20,50 대)와 강한 상관관계를 가지며 업종 q,r,p(음식점,학문/ 교육, 스포츠)과 약한 상관관계를 보여 소상공인들의 주 창업 업종이 아닌 것으로 보인다.

따라서 소상공인과 격자 내 업종 존재 유무 데이터간의 상관관계를 통해 소상공인의 주 창업 업종과 이유를 유추해보고자 하였다.

데이터 분석(2) – 상관분석; 변수('smbiz\_yn\_cnt', 'indcd\_f\_yn','indcd\_l\_yn', 'indcd\_n\_yn', 'indcd\_p\_yn','indcd\_q\_yn','indcd\_r\_yn')



9) 격자내 소상공인 수와 업종 (f,l,n,p,q,r) 간의 상관관계를 통해 소상공인의 창업 경향을 알아본 결과

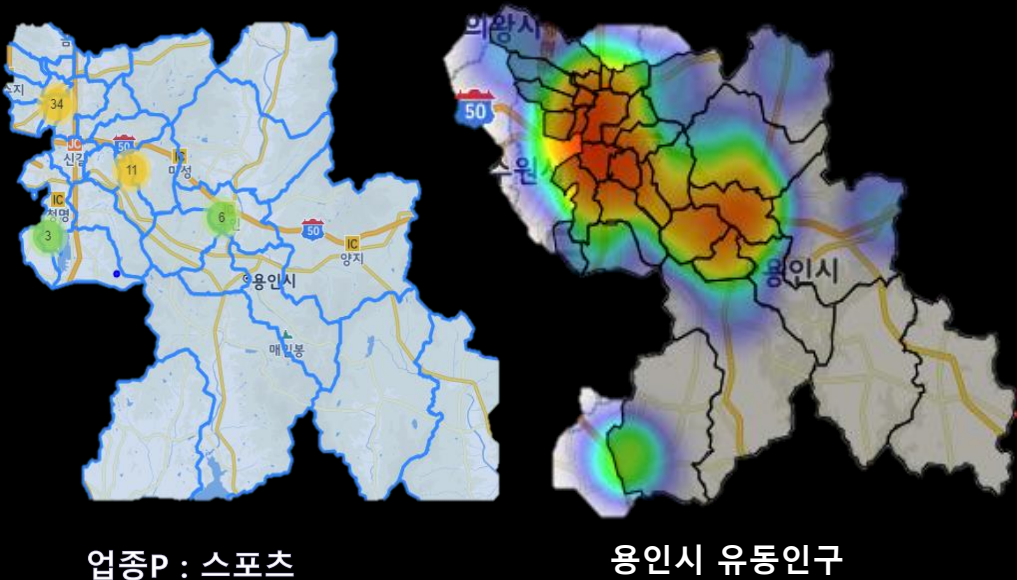
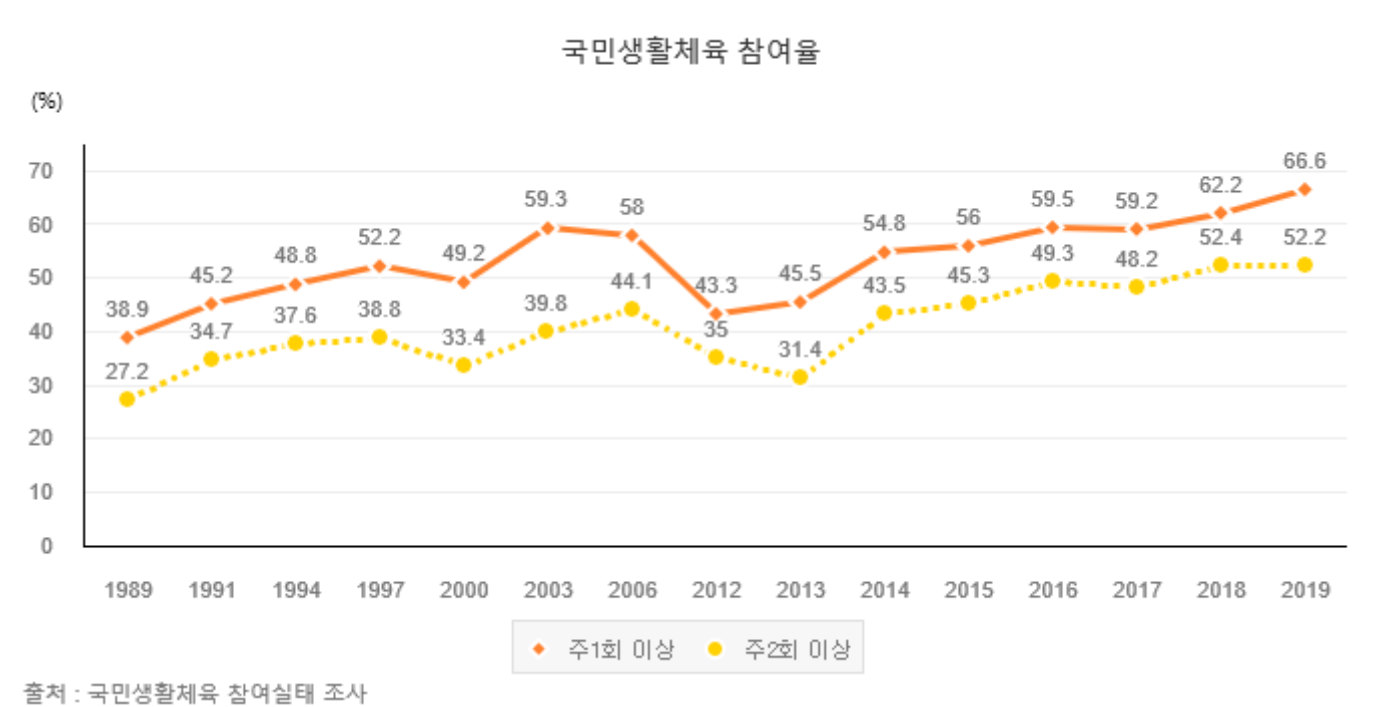
Q : 음식점, R :학문/교육 이 0.5 이상의 상관관계로 타 업종에 비해 많이 창업하는 경향이 있는 것으로 보인다.

이전 회귀분석에서 알 수 있었던 것처럼 P,R,Q,F 업종이 용인시 내의 매출과 양의 상관관계를 가짐에도 Q,R 업종에 많이 창업하고 있는 경향이 있어 P(스포츠), F(생활서비스) 업종에의 창업을 격려하는 등의 활동이 필요하다.

특히 P(스포츠)는 회귀식에서도 높은 계수값(0.4073)을 가져 주목할 필요가 있고

# 결론 및 정책 제언

## 결론 및 정책 제안 (Y-스포츠 창업 정책)



해가 갈수록 국민생활체육 참여율은 증가하고 있다. 그러나 늘어나는 참여율에 비해 유동인구수가 많은 용인시의 기흥구, 수지구를 중심으로 체육센터가 타 업종에 비해 부족한 수준이다.

‘창업진흥원 2019년 창업기업실태조사 통계자료’에 따르면 전국 개인사업체중 스포츠 업종 비율이 0.93%인데 이에 비해 용인은 0.13%수준이라는 점 역시 주목할 만하다.

## 결론 및 정책 제안 (Y-스포츠 창업 정책)

용인시가 경기도내 미래의 스포츠 산업 창업자에게 기회의 도시가 될 수 있으므로 다음의 정책을 제시하게 되었다.

### Y-스포츠 창업 정책

#### 1) Y - 스포츠 창업 공모전

- 스포츠 시설, 스포츠서비스 분야의 아이디어 공모전을 통해 좋은 아이디어를 가진 예비 창업자를 발굴하여 창업 자금을 지원한다.

#### 2) 대학연계 스포츠창업 교육

- 현재 이공계열 위주인 대학연계 창업교육에 스포츠 창업 교육을 추가한다.  
(용인시 관.학 창업지원센터에서 지원하는 170개의 보육실수 중 74.8%가 이공계열 교육, 그 외 디자인교육으로 이루어져 있다.)

(출처 : [https://www.yongin.go.kr/home/www/www\\_05/www\\_05\\_01/www\\_05\\_01\\_12/www\\_05\\_01\\_12\\_01.jsp](https://www.yongin.go.kr/home/www/www_05/www_05_01/www_05_01_12/www_05_01_12_01.jsp) )

#### 3) 용인시 플랫폼시티 도시개발사업 스포츠 산업단지 확보

- 2028년까지 준공될 예정인 용인시 플랫폼시티에 스포츠사업 창업자들이 상생할 수 있는 다양한 스포츠 산업단지 조성

**감사합니다.**