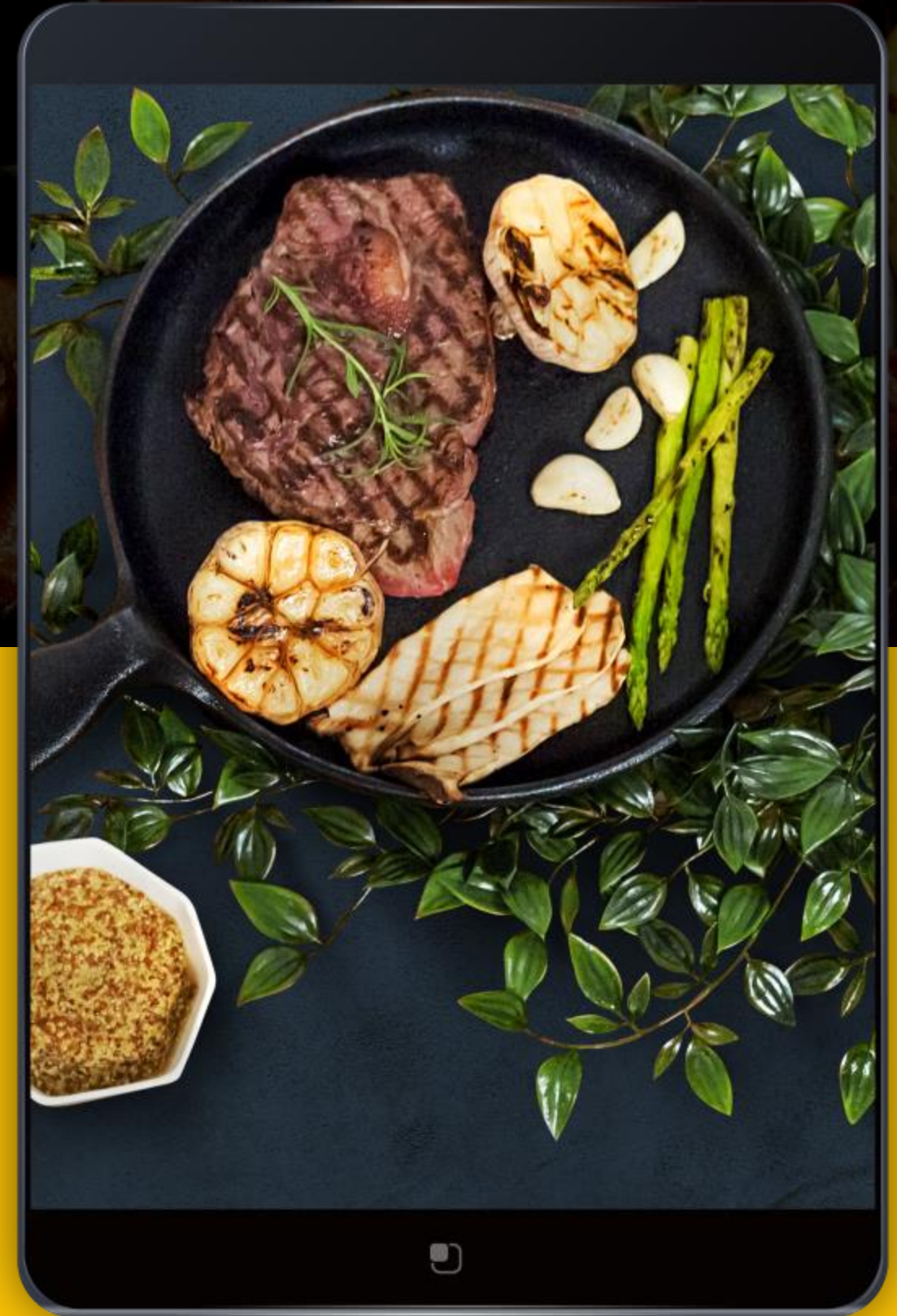


# 외식업 창업 신호등

- 강남구 상권을 중심으로 -







## TABLE OF CONTENTS

**01**

팀원소개

**02**

배경

**03**

목적 및 기대효과

**04**

프로젝트 일정

**05**

데이터 수집

**06**

데이터 처리 및 분석

**07**

결과 및 웹구현

**08**

Q&A

# 01. 팀원 소개

**이신\***

폐점, 생존률 데이터 전처리 및 시각화

**성현\***

Folium을 활용한 시각화

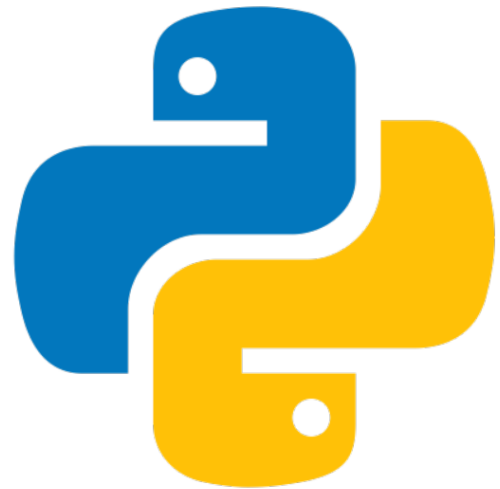
**황성\***

인구 데이터 수집 및 시각화

**황인\***

상권 관련 데이터 전처리 및 시각화

# 사용 언어 및 API



01

Python



02

Pandas

Folium



03

Folium

{JSON}

04

JSON

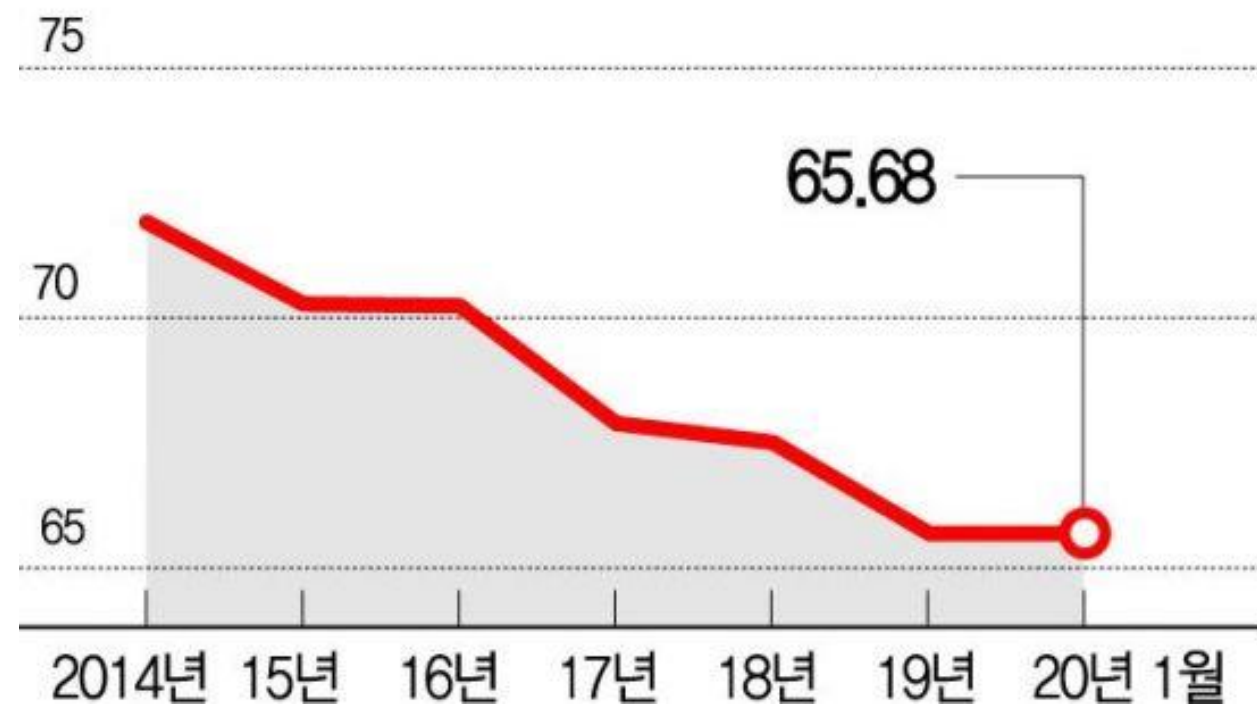


05

R

## 02.배경

외식업 경기지수 (자료: 한국외식산업연구원)



코로나 19 및 사회적 거리두기로 인한 외식업 침체의 장기화  
타업종 대비 높은 외식업의 높은 폐업률

음식점 10개 창업 때 8개 이상 폐업...폐업률, 타업종보다 높아  
2021.10.28 매일경제

손님 끊기고 빚만 늘어..."살려달라" 자영업자 절규 언제까지  
2021.12.25 연합뉴스

코로나19로 1년 새 서울 외식업 매출 20% 이상 감소  
2021.06.04 국토매일

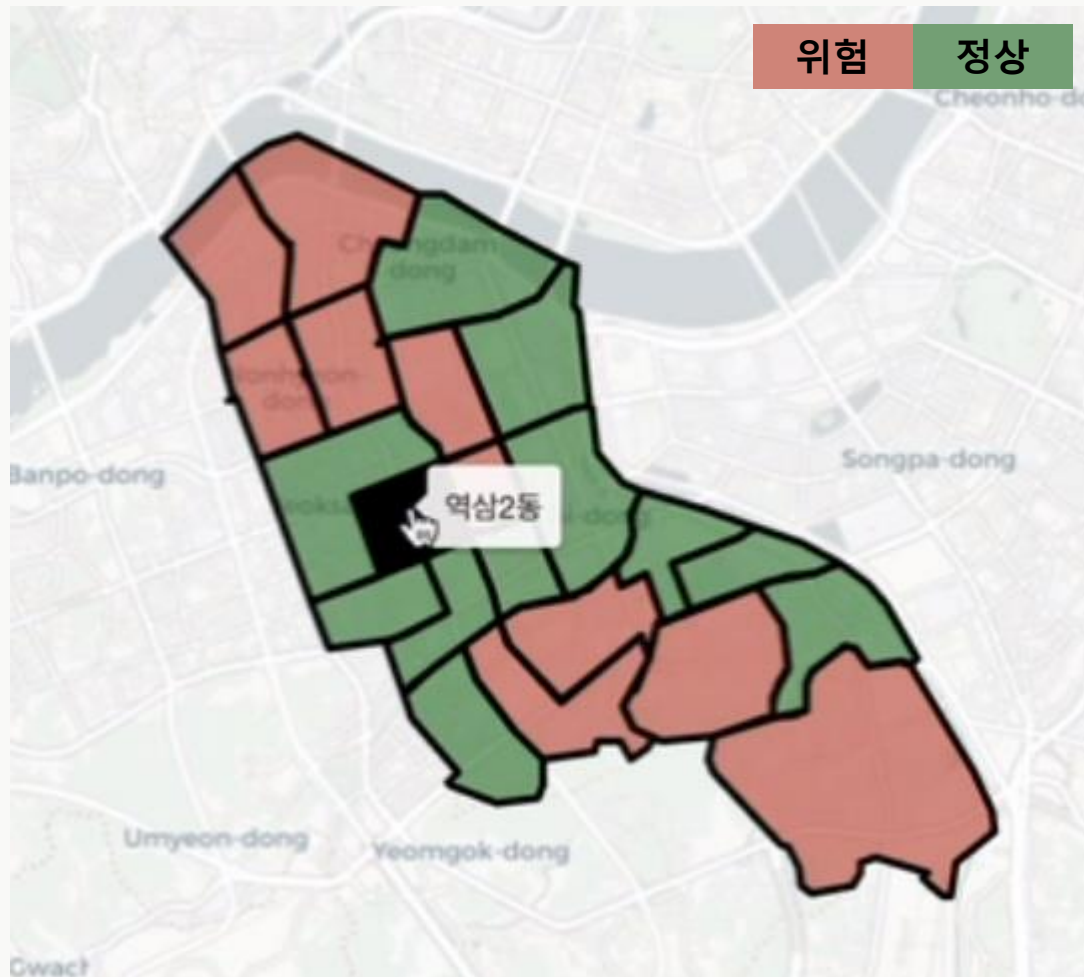


# 03. 목적 및 기대효과

상권별 창업 신호등을 제공



예비창업자가 해당상권의 위험도를 쉽게 파악할 수 있게 함



상권별 - 업종별 세부정보 제공

“ 동별 매출 및 집객시설, 운영개월수등  
매장운영 관련 정보제공 ”

“ 인구밀도, 상주인구, 직장인구등  
인구관련 정보제공 ”

“ 변수별, 폐점률 분석을 통한 추천업종  
정보제공 ”

# 04. 프로젝트 일정

| #       | 1주차 |  |  | 2주차 |  |  | 3주차 |  |  | 4주차 |  |  |
|---------|-----|--|--|-----|--|--|-----|--|--|-----|--|--|
| 주제선정    |     |  |  |     |  |  |     |  |  |     |  |  |
| 데이터 수집  |     |  |  |     |  |  |     |  |  |     |  |  |
| 데이터 전처리 |     |  |  |     |  |  |     |  |  |     |  |  |
| 데이터 시각화 |     |  |  |     |  |  |     |  |  |     |  |  |
| 데이터 분석  |     |  |  |     |  |  |     |  |  |     |  |  |
| PPT 제작  |     |  |  |     |  |  |     |  |  |     |  |  |
| 보고서 작성  |     |  |  |     |  |  |     |  |  |     |  |  |



## 05. 데이터 수집

# 1. 서울시 열린데이터 광장 - 서울시 우리마을가게 상권분석서비스 데이터

## - 상권, 인구 및 매출관련 data

| A    | B      | C     | D     | E       | F      | H        | I     | J    | K     | L    | M     | N    | O     | P     |
|------|--------|-------|-------|---------|--------|----------|-------|------|-------|------|-------|------|-------|-------|
| 기준_년 | 코기존_분기 | 상권_구분 | 상권_구분 | 상권_코드   | 상권_코드  | 서비스_업    | 서비스_업 | 점포_수 | 유사_업종 | 개업_율 | 개업_점포 | 폐업_률 | 폐업_점포 | 프랜차이즈 |
| 2019 | 1      | A     | 골목상권  | 1000901 | 논현로28ㄱ | CS100004 | 양식음식점 | 4    | 5     | 0    | 0     | 0    | 0     | 1     |
| 2019 | 1      | D     | 발달상권  | 1001111 | 서울 강남  | CS100001 | 한식음식점 | 41   | 48    | 2    | 1     | 2    | 1     | 7     |
| 2019 | 1      | U     | 관광특구  | 1001496 | 강남 마이  | CS100005 | 제과점   | 11   | 20    | 0    | 0     | 0    | 1     | 9     |
| 2019 | 1      | A     | 골목상권  | 1000921 | 삼성로64ㄱ | CS100010 | 커피-음료 | 3    | 3     | 0    | 0     | 0    | 0     | 0     |
| 2019 | 1      | D     | 발달상권  | 1001065 | 온마아파트  | CS100006 | 패스트푸드 | 1    | 3     | 0    | 0     | 0    | 0     | 2     |
| 2019 | 1      | D     | 발달상권  | 1001087 | 서울 강남  | CS100010 | 커피-음료 | 38   | 60    | 7    | 4     | 7    | 4     | 22    |
| 2019 | 1      | A     | 골목상권  | 1000897 | 논현로123 | CS100010 | 커피-음료 | 1    | 2     | 0    | 0     | 50   | 1     | 1     |
| 2019 | 1      | A     | 골목상권  | 1000900 | 논현로26ㄱ | CS100010 | 커피-음료 | 16   | 19    | 5    | 1     | 5    | 1     | 3     |
| 2019 | 1      | D     | 발달상권  | 1001098 | 포스코사거  | CS100002 | 중식음식점 | 8    | 8     | 13   | 1     | 0    | 0     | 0     |
| 2019 | 1      | A     | 골목상권  | 1000911 | 도산대로8  | CS100003 | 일식음식점 | 10   | 10    | 0    | 0     | 0    | 0     | 0     |
| 2019 | 1      | D     | 발달상권  | 1001111 | 서울 강남  | CS100003 | 일식음식점 | 12   | 16    | 0    | 19    | 3    | 4     | 4     |
| 2019 | 1      | U     | 관광특구  | 1001496 | 강남 마이  | CS100004 | 양식음식점 | 31   | 36    | 0    | 0     | 0    | 0     | 5     |
| 2019 | 1      | D     | 발달상권  | 1001099 | 포스코사거  | CS100010 | 커피-음료 | 11   | 18    | 6    | 1     | 0    | 0     | 0     |
| 2019 | 1      | A     | 골목상권  | 1000935 | 태해란로8  | CS100003 | 일식음식점 | 14   | 14    | 0    | 0     | 0    | 0     | 0     |
| 2019 | 1      | D     | 발달상권  | 1001131 | 청담역_2  | CS100001 | 한식음식점 | 18   | 21    | 0    | 0     | 0    | 0     | 3     |
| 2019 | 1      | D     | 발달상권  | 1001087 | 서울 강남  | CS100002 | 중식음식점 | 10   | 10    | 0    | 0     | 0    | 0     | 0     |
| 2019 | 1      | D     | 발달상권  | 1001062 | 서울 강남  | CS100010 | 커피-음료 | 30   | 42    | 5    | 2     | 5    | 2     | 12    |
| 2019 | 1      | A     | 골목상권  | 1000939 | 학동로77ㄱ | CS100010 | 커피-음료 | 3    | 3     | 33   | 1     | 33   | 1     | 0     |
| 2019 | 1      | D     | 발달상권  | 1001077 | 선릉역_1  | CS100004 | 양식음식점 | 10   | 11    | 9    | 1     | 9    | 1     | 1     |
| 2019 | 1      | A     | 골목상권  | 1000924 | 선릉로121 | CS100010 | 커피-음료 | 8    | 8     | 0    | 0     | 0    | 0     | 0     |
| 2019 | 1      | A     | 골목상권  | 1000916 | 봉은사로6  | CS100008 | 분식전문점 | 3    | 3     | 33   | 1     | 0    | 0     | 0     |
| 2019 | 1      | A     | 골목상권  | 1000911 | 도산대로8  | CS100010 | 커피-음료 | 3    | 3     | 33   | 1     | 67   | 2     | 0     |
| 2019 | 1      | D     | 발달상권  | 1001140 | 서울 강남  | CS100005 | 제과점   | 5    | 12    | 17   | 2     | 0    | 0     | 7     |
| 2019 | 1      | A     | 골목상권  | 1000935 | 태해란로8  | CS100009 | 호프-간이 | 7    | 11    | 0    | 0     | 0    | 0     | 4     |
| 2019 | 1      | A     | 골목상권  | 1000908 | 도산대로3  | CS100009 | 호프-간이 | 4    | 4     | 0    | 0     | 0    | 0     | 0     |
| 2019 | 1      | D     | 발달상권  | 1001035 | 매봉역    | CS100004 | 양식음식점 | 9    | 9     | 0    | 0     | 0    | 0     | 0     |
| 2019 | 1      | D     | 발달상권  | 1001069 | 온마아파트  | CS100004 | 양식음식점 | 0    | 0     | 0    | 0     | 0    | 0     | 0     |
| 2019 | 1      | A     | 골목상권  | 1000929 | 압구정로2  | CS100007 | 키친전문점 | 1    | 2     | 0    | 0     | 0    | 0     | 0     |

## 2. 서울시 행정동경계 (ver20220309) json file

```
{'type': 'FeatureCollection',
 'name': 'HangJeongDong_ver20191001',
 'crs': {'type': 'name',
 'properties': {'name': 'urn:ogc:def:crs:OGC:1.3:CRS84'}},
 'features': [{'type': 'Feature',
 'properties': {'OBJECTID': 358,
 'adm_nm': '신사동',
 'adm_cd': '1123051',
 'adm_cd2': '1168051000'},
 'geometry': {'type': 'MultiPolygon',
 'coordinates': [[[[[127.02807815323149, 37.519584003369],
 [127.02720592329885, 37.51926729481519],
 [127.02526080146446, 37.518556420544876],
 [127.02116872995951, 37.516819200932005],
 [127.01944104956044, 37.515987241640666],
 [127.01947830573368, 37.51634195419062],
 [127.0190251971729, 37.51823713836514],
 [127.01891540067083, 37.51857494289874],
 [127.01828346580481, 37.52051905145248],
```



## 06. 데이터 처리

### 1) 폐업률, 생존률 관련 데이터 가공

|      | 년도   | 폐업률      | 생존률       |
|------|------|----------|-----------|
| 동    |      |          |           |
| 강남구  | 2019 | 4.028546 | 95.971454 |
| 논현1동 | 2019 | 4.554852 | 95.445148 |
| 역삼2동 | 2019 | 3.418929 | 96.581071 |
| 삼성2동 | 2019 | 4.190119 | 95.809881 |
| 논현2동 | 2019 | 4.351290 | 95.648710 |

```
1 # 표준화
2 m = store19.mean(axis = 0)
3 s = store19.std(axis = 0)
4 store19 = (store19 - m) / s
5
6 m = store20.mean(axis = 0)
7 s = store20.std(axis = 0)
8 store20 = (store20 - m) / s
9
10 m = store21.mean(axis = 0)
11 s = store21.std(axis = 0)
12 store21 = (store21 - m) / s
```

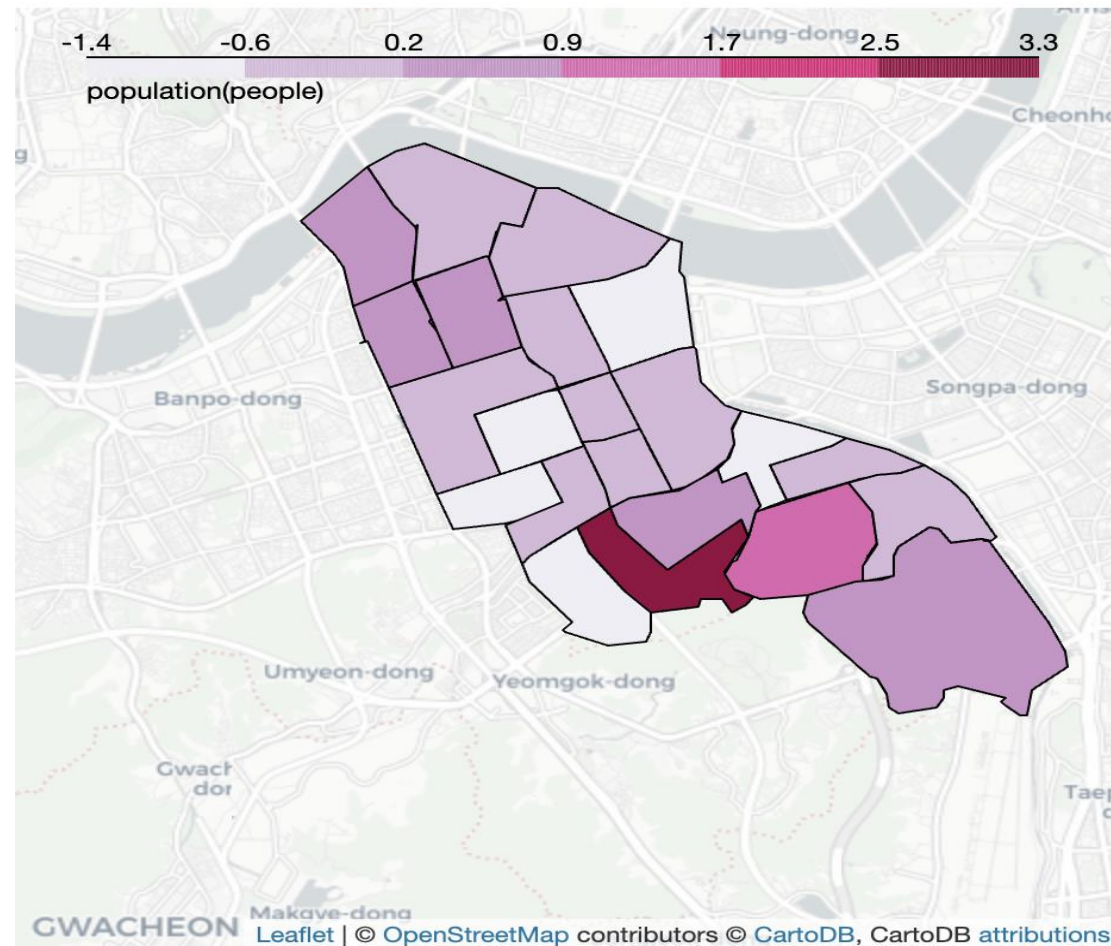
|      | 폐업률       | 생존률       |
|------|-----------|-----------|
| 동    |           |           |
| 논현1동 | 0.731487  | -0.731487 |
| 역삼2동 | 0.487933  | -0.487933 |
| 삼성2동 | 0.372851  | -0.372851 |
| 논현2동 | 0.202691  | -0.202691 |
| 대치2동 | 0.148884  | -0.148884 |
| 청담동  | 0.034301  | -0.034301 |
| 역삼1동 | -0.008101 | 0.008101  |
| 대치1동 | -0.090058 | 0.090058  |
| 압구정동 | -0.125243 | 0.125243  |
| 신사동  | -0.125599 | 0.125599  |
| 대치4동 | -0.213128 | 0.213128  |
| 도곡1동 | -0.455935 | 0.455935  |
| 도곡2동 | -0.648063 | 0.648063  |

기존 데이터에서 편차가 크지 않아, 년도별로 표준화 진행

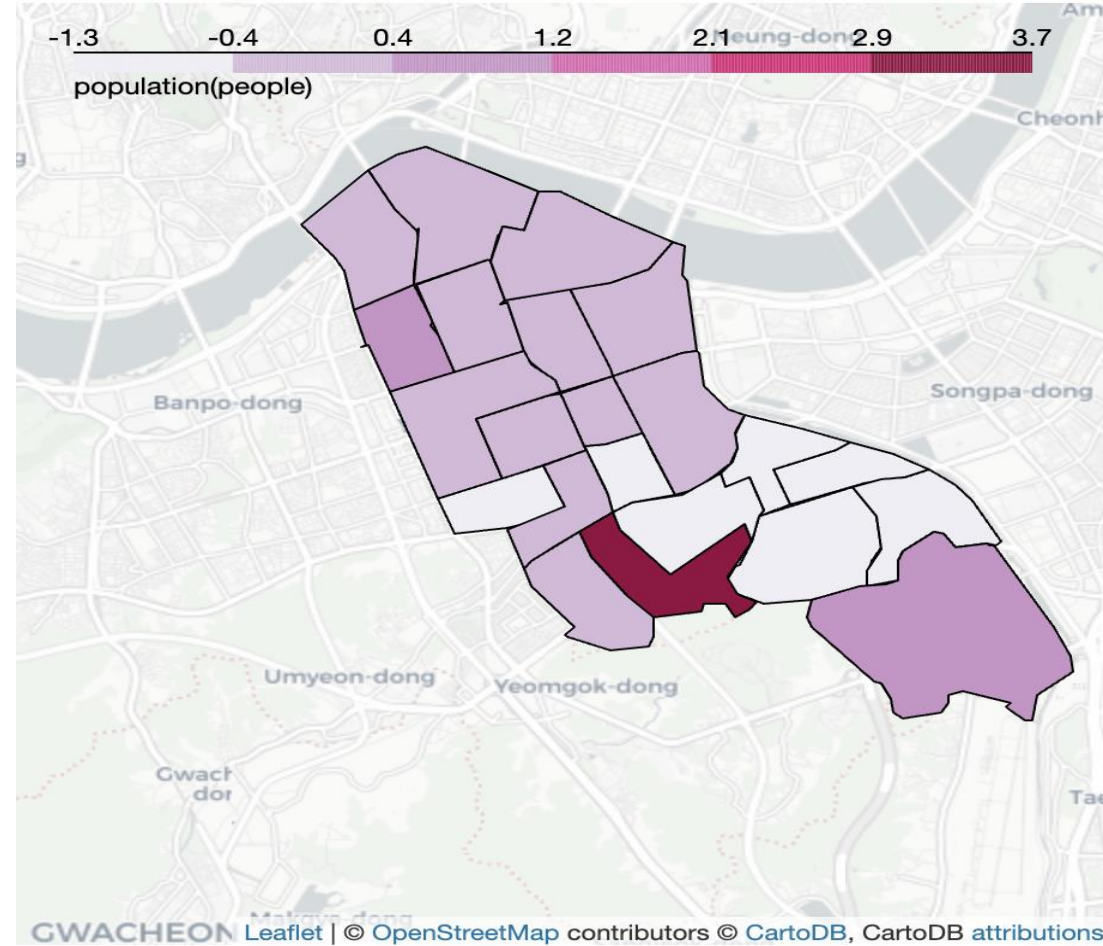
## 06. 데이터 처리

### 1) 폐업률, 생존률 관련 데이터 시각화

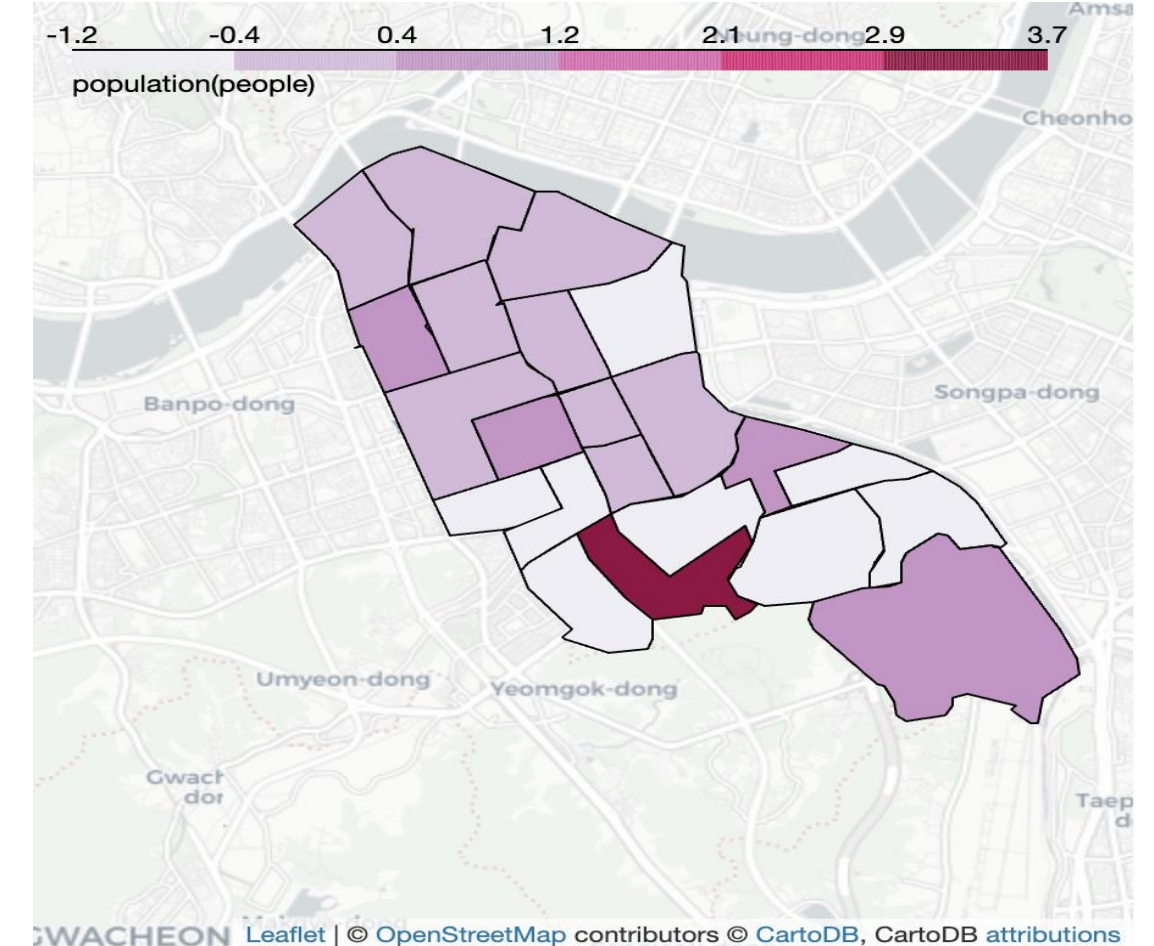
2019



2020



2021



- 밝을수록 폐업률 낮음.
- 강남남부지역(개포4동, 개포2동, 일원본동, 수서동, 일원 1동)에서 폐업률이 낮아지는 경향.



# 06. 데이터 처리

## 2) 상권관련 데이터 전처리 및 가공 - 집객시설 및 상권변화지표

### 상권 - 집객시설

#### size 와 구조

```
mydata3 <- read.csv("/Users/chan/Desktop/Dataproject/data/상권/집객시설.csv",fileEncoding = 'euc-kr')

dim(mydata3)

## [1] 38774    26

str(mydata3)

## 'data.frame':    38774 obs. of  26 variables:
## $ 기준_년_코드      : int   2021 2021 2021 2021 2021 2021 2021 2021 2021 2021 ...
## $ 기준_분기_코드    : int    3 3 3 3 3 3 3 3 3 3 ...
## $ 상권_구분_코드     : chr   "A" "A" "A" "A" ...
## $ 상권_구분_코드_명  : chr   "골목상권" "골목상권" "골목상권" "골목상권" ...
## $ 상권_코드         : int   1000289 1000835 1000839 1000583 1000110 1000111 1000116 1000234 1000245 1000455 ...
## $ 상권_코드_명      : chr   "인촌로17가길" "청림3길" "호암로24길" "남부순환로11길" ...
## $ 집객시설_수       : int    6 12 25 7 16 11 9 7 13 11 ...
## $ 관공서_수         : int   NA NA 1 1 NA NA NA 1 1 NA ...
## $ 은행_수           : int   NA NA 1 NA 2 NA NA NA NA NA ...
## $ 종합병원_수       : int   NA NA NA NA NA NA NA NA 1 NA ...
## $ 일반_병원_수      : int   NA NA NA NA NA NA NA NA NA NA ...
## $ 약국_수           : int    1 2 7 1 1 1 NA NA 5 2 ...
## $ 유치원_수         : int   NA 1 NA NA NA NA NA NA NA ...
## $ 초등학교_수       : int   NA NA NA NA 1 NA NA NA NA ...
## $ 중학교_수         : int   NA NA NA NA NA NA NA NA NA ...
## $ 고등학교_수       : int   NA NA NA NA NA NA NA NA NA ...
## $ 대학교_수         : int   NA NA NA NA NA NA NA NA NA ...
## $ 백화점_수         : int   NA NA NA NA NA NA NA NA NA ...
## $ 슈퍼마켓_수       : int   NA NA NA NA NA NA NA NA NA ...
## $ 극장_수           : int   NA NA NA NA NA NA NA NA NA ...
## $ 숙박_시설_수      : int   NA 2 NA NA NA NA NA NA NA ...
## $ 공항_수           : logi   NA NA NA NA NA NA ...
## $ 철도_역_수        : int   NA NA NA NA NA NA NA NA NA ...
## $ 버스_터미널_수    : int   NA NA NA NA NA NA NA NA NA ...
## $ 지하철_역_수     : int   NA NA NA NA 1 NA NA NA NA ...
## $ 버스_정거장_수    : int    2 2 3 1 3 7 2 2 1 NA ...
```

#### 결측치 확인 및 제거

```
for(i in 7:26 ){
  mydata3[,i] <- ifelse(is.na(mydata3[,i]),0,mydata3[,i])
} ## na 값이 들어있는 모든 열 0 으로 치환하는 반복문
table(is.na(mydata3))

##
##      FALSE
## 1008124

str(mydata3)

## 'data.frame':    38774 obs. of  26 variables:
## $ 기준_년_코드      : int   2021 2021 2021 2021 2021 2021 2021 2021 2021 2021 ...
## $ 기준_분기_코드    : int    3 3 3 3 3 3 3 3 3 3 ...
## $ 상권_구분_코드     : chr   "A" "A" "A" "A" ...
## $ 상권_구분_코드_명  : chr   "골목상권" "골목상권" "골목상권" "골목상권" ...
## $ 상권_코드         : int   1000289 1000835 1000839 1000583 1000110 1000111 1000116 1000234 1000245 1000455 ...
## $ 상권_코드_명      : chr   "인촌로17가길" "청림3길" "호암로24길" "남부순환로11길" ...
## $ 집객시설_수       : int    6 12 25 7 16 11 9 7 13 11 ...
## $ 관공서_수         : num    0 0 1 1 0 0 0 1 1 0 ...
## $ 은행_수           : num    0 0 1 0 2 0 0 0 0 0 ...
## $ 종합병원_수       : num    0 0 0 0 0 0 0 0 1 0 ...
## $ 일반_병원_수      : num    0 0 0 0 0 0 0 0 0 0 ...
## $ 약국_수           : num    1 2 7 1 1 1 0 0 5 2 ...
## $ 유치원_수         : num    0 1 0 0 0 0 0 0 0 0 ...
## $ 초등학교_수       : num    0 0 0 0 1 0 0 0 0 0 ...
## $ 중학교_수         : num    0 0 0 0 0 0 0 0 0 0 ...
## $ 고등학교_수       : num    0 0 0 0 0 0 0 0 0 0 ...
## $ 대학교_수         : num    0 0 0 0 0 0 0 0 0 0 ...
## $ 백화점_수         : num    0 0 0 0 0 0 0 0 0 0 ...
## $ 슈퍼마켓_수       : num    0 0 0 0 0 0 0 0 0 0 ...
## $ 극장_수           : num    0 0 0 0 0 0 0 0 0 0 ...
## $ 숙박_시설_수      : num    0 2 0 0 0 0 0 0 0 0 ...
## $ 공항_수           : num    0 0 0 0 0 0 0 0 0 0 ...
## $ 철도_역_수        : num    0 0 0 0 0 0 0 0 0 0 ...
## $ 버스_터미널_수    : num    0 0 0 0 0 0 0 0 0 0 ...
## $ 지하철_역_수     : num    0 0 0 0 1 0 0 0 0 0 ...
## $ 버스_정거장_수    : num    2 2 3 1 3 7 2 2 1 0 ...
```

# 06. 데이터 처리

## 2) 상권관련 데이터 전처리 및 가공 - 집객시설 및 상권변화지표

| 년도   | 행정동  | 운영자이 | 폐업자이 | 운영영업개월     | 폐업영업개월    | 서울운영영업     | 서울폐업영업    | 상권변화지표 | 주요시설    |
|------|------|------|------|------------|-----------|------------|-----------|--------|---------|
| 2019 | 개포2동 | H    | H    | 108.666667 | 53.833333 | 108.5      | 53.25     | 정체     | 개포동역    |
| 2020 | 개포2동 | H    | H    | 112.916667 | 55.333333 | 112        | 54        | 정체     | 개포동역    |
| 2021 | 개포2동 | H    | H    | 116.916667 | 56.166667 | 116.5      | 54.75     | 정체     | 개포동역    |
| 2019 | 개포4동 | H    | L    | 116.964286 | 49.785714 | 108.5      | 53.25     | 상권축소   | 데님호텔    |
| 2020 | 개포4동 | H    | L    | 120.464286 | 50.214286 | 112        | 54        | 상권축소   | 데님호텔    |
| 2021 | 개포4동 | H    | L    | 127.071429 | 51.25     | 116.5      | 54.75     | 상권축소   | 데님호텔    |
| 2019 | 논현1동 | L    | L    | 100.75     | 49.214286 | 108.5      | 53.25     | 다이나믹   | 엘리에나호텔  |
| 2020 | 논현1동 | L    | L    | 105.607143 | 49.857143 | 112        | 54        | 다이나믹   | 엘리에나호텔  |
| 2021 | 논현1동 | L    | L    | 110.910714 | 50.571429 | 116.5      | 54.75     | 다이나믹   | 엘리에나호텔  |
| 2019 | 논현2동 | L    | L    | 93.482143  | 49.267857 | 108.5      | 53.25     | 다이나믹   | 임페리얼펠리: |
| 2020 | 논현2동 | L    | L    | 96.392857  | 49.535714 | 112        | 54        | 다이나믹   | 임페리얼펠리: |
| 2021 | 논현2동 | L    | L    | 99.178571  | 50.375    | 116.5      | 54.75     | 다이나믹   | 임페리얼펠리: |
| 2019 | 대치1동 | L    | L    | 96.296296  | 49.537037 | 108.703704 | 53.296296 | 다이나믹   | 단국대학교사  |
| 2020 | 대치1동 | L    | L    | 101.944444 | 50.092593 | 112.296296 | 54        | 다이나믹   | 단국대학교사  |
| 2021 | 대치1동 | L    | L    | 108.203704 | 50.962963 | 116.703704 | 54.814814 | 다이나믹   | 단국대학교사  |
| 2019 | 대치2동 | L    | L    | 94.425926  | 48.629629 | 108.611111 | 53.277778 | 다이나믹   | 학여울역    |
| 2020 | 대치2동 | L    | L    | 99.092593  | 49.259259 | 112.166667 | 54        | 다이나믹   | 학여울역    |

```
## 컬럼 제거
df.drop(columns=["운영차이", "폐업차이"])
df=df.drop(columns=["행정동"])
## 수치 간소화
df1=df.round(2)
## 인덱스 제거
df1 =df1.set_index("년도")
### 동별 분리
df_개포2동 = df1.iloc[:3,:]
#...
df_청담동 = df1.iloc[51:54,:]
###행열치환
df_개포2동 = df_개포2동.transpose()
#...
df_청담동 = df_청담동.transpose()
```

df\_개포4동

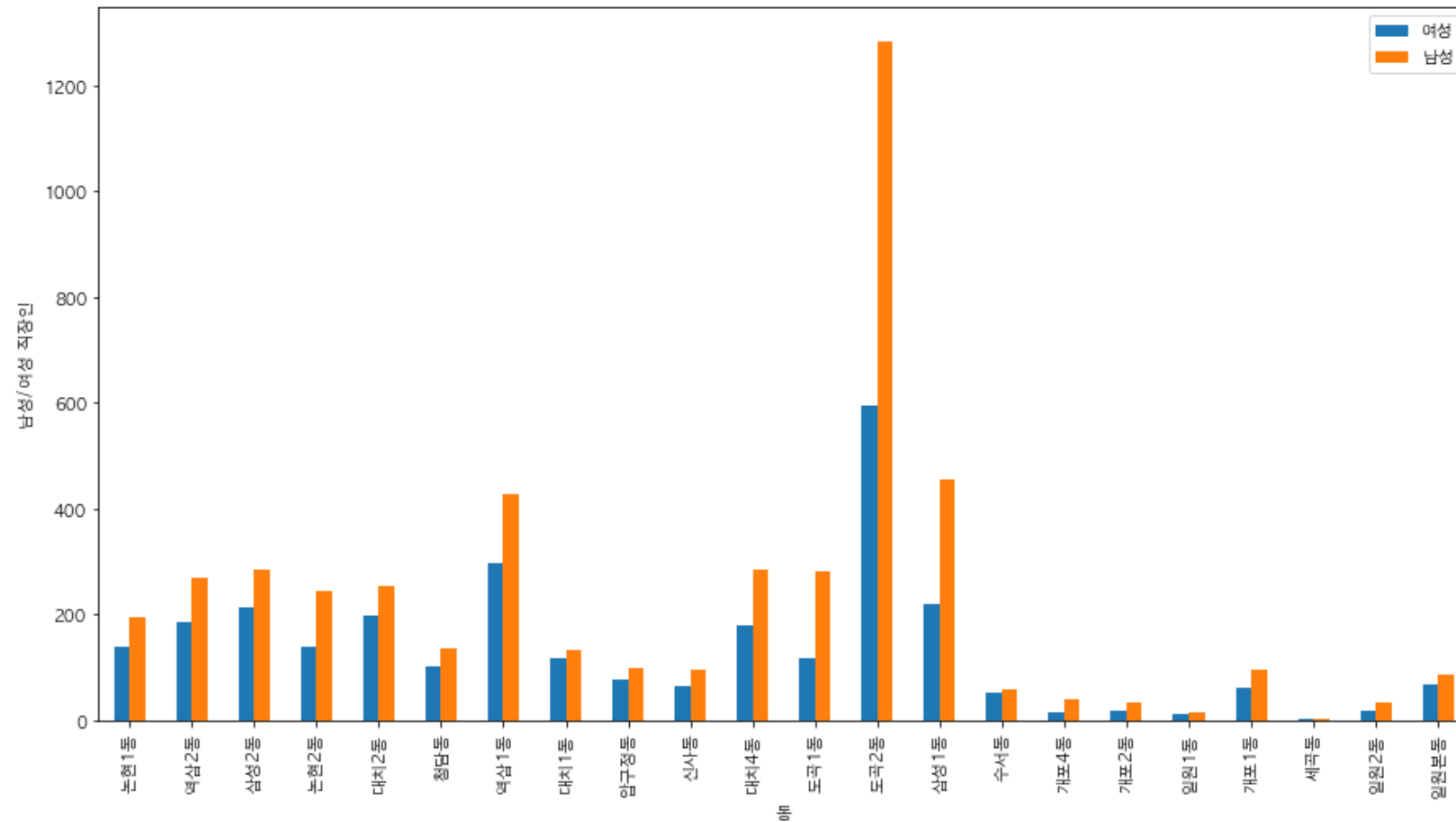
| 년도       | 2019   | 2020   | 2021   |
|----------|--------|--------|--------|
| 운영영업개월   | 116.96 | 120.46 | 127.07 |
| 폐업영업개월   | 49.79  | 50.21  | 51.25  |
| 서울운영영업개월 | 108.5  | 112.0  | 116.5  |
| 서울폐업영업개월 | 53.25  | 54.0   | 54.75  |
| 상권변화지표   | 상권축소   | 상권축소   | 상권축소   |
| 주요시설     | 데님호텔   | 데님호텔   | 데님호텔   |



# 06. 데이터 처리

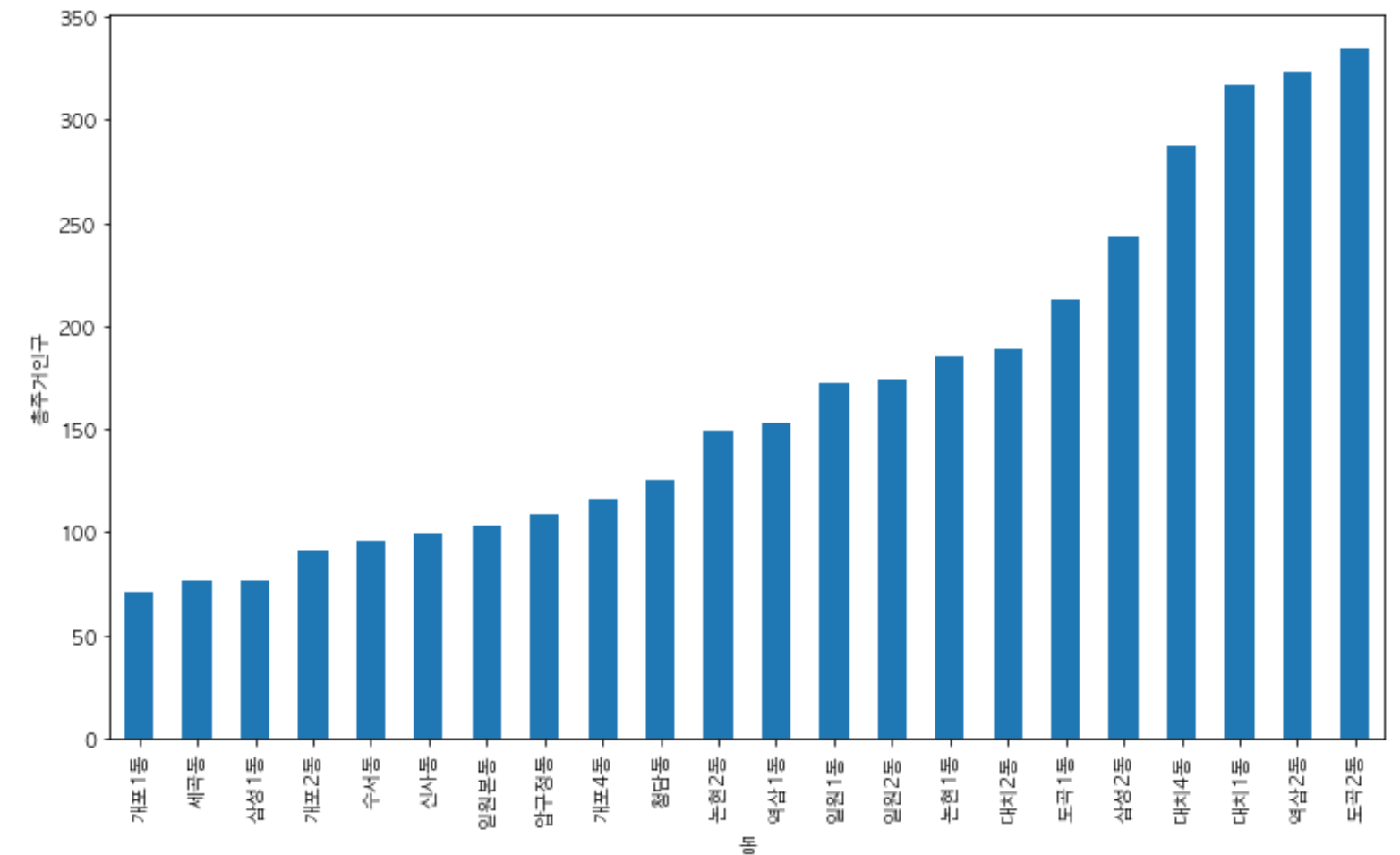
## 3) 상권관련 데이터 시각화 - 인구밀도

```
aa1.plot(y=['여성', '남성'],figsize=(15,8),kind="bar")  
plt.ylabel('남성/여성')
```



강남구 직장인 중 도곡2동이 여성과 남성이  
제일 높음

```
aa1["총주거인구"].sort_values().plot(kind="bar",figsize=(12,7))  
plt.ylabel('총주거인구')
```

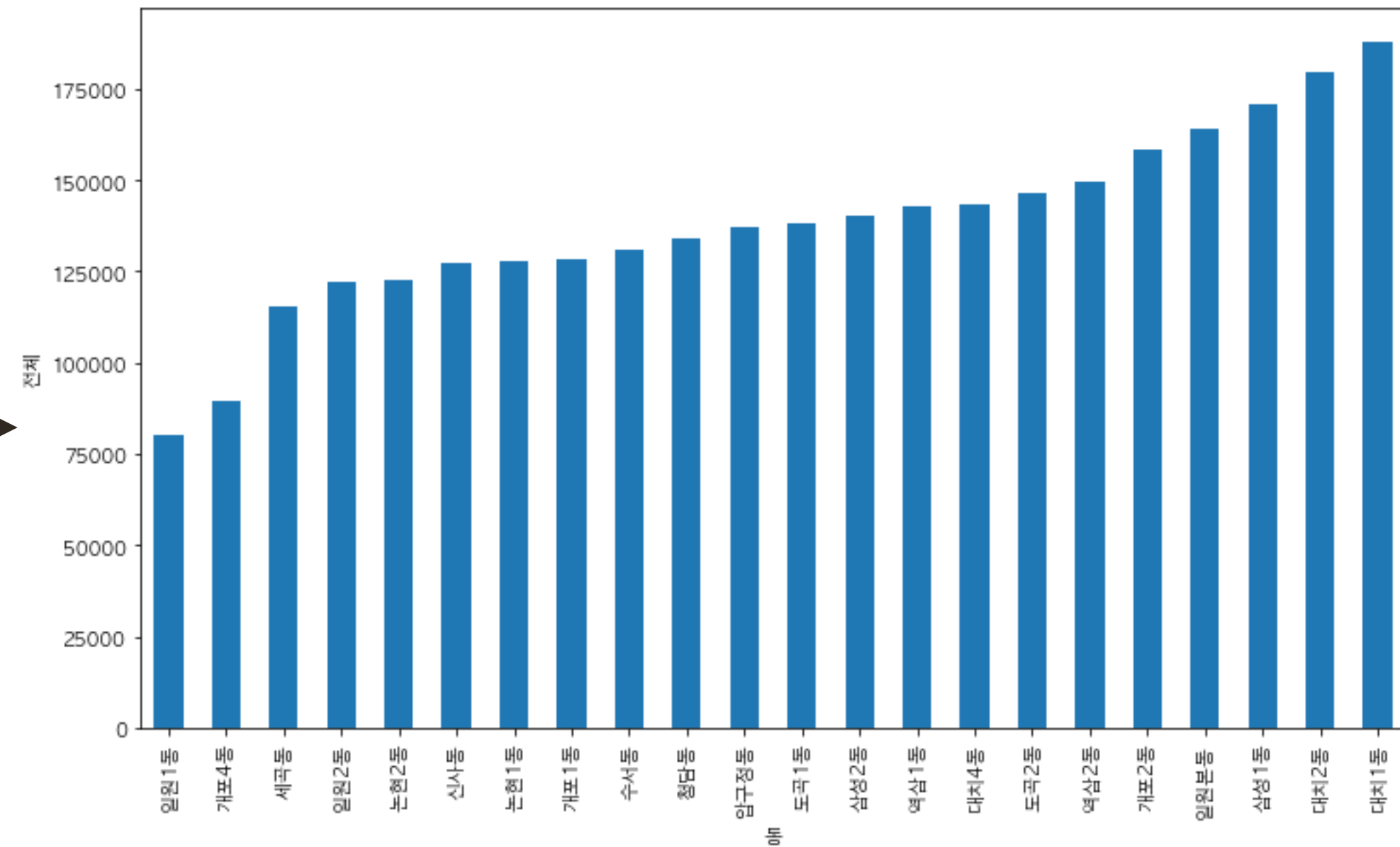


강남구 주거인구중 도곡2동이 제일 많음  
직장인 인구 또한 도곡2동이 제일 많음

# 06. 데이터 처리

## 4) 상권관련 데이터 시각화 - 임대료

```
aa1.set_index("동",inplace=True)  
aa1["전체"].sort_values().plot(kind="bar",figsize=(12,7))  
plt.ylabel('전체')
```





## 06. 데이터 처리

### 5) 상권관련 데이터 전처리 및 가공 - 추정매출

```
mydata = pd.read_csv("/Users/chan/Desktop/데이터프로젝/추정매출정리.csv")
mydata
```

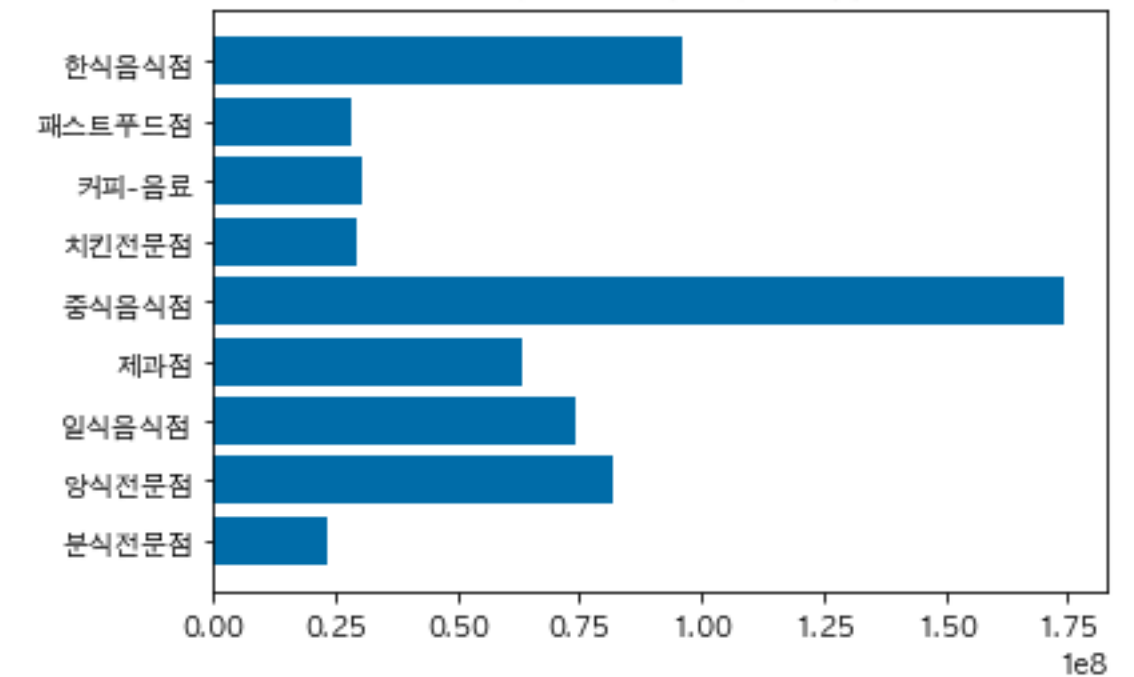
|    | 년    | 동    | 업종      | 2021년 총매출금액  | 2021년 총매출건수 | 총 건당 금액    | 주중매출비율평균  | 점포수   | 점포당 연간매출금액   |
|----|------|------|---------|--------------|-------------|------------|-----------|-------|--------------|
| 0  | 2021 | 강남구  | 분식전문점   | 2.810000e+11 | 19321963    | 8577597.0  | 80.727057 | 6502  | 4.316185e+07 |
| 1  | 2021 | 강남구  | 양식전문점   | 2.770000e+11 | 6476307     | 17818458.7 | 80.431372 | 4598  | 6.014403e+07 |
| 2  | 2021 | 강남구  | 일식음식점   | 2.840000e+11 | 7072943     | 19803042.7 | 83.419394 | 3406  | 8.338577e+07 |
| 3  | 2021 | 강남구  | 제과점     | 1.200000e+11 | 11181567    | 8355526.8  | 76.087872 | 1680  | 7.171699e+07 |
| 4  | 2021 | 강남구  | 중식음식점   | 1.780000e+11 | 5194091     | 11450621.8 | 79.489139 | 1727  | 1.030312e+08 |
| 5  | 2021 | 강남구  | 치킨전문점   | 4.813047e+10 | 1645323     | 8957467.5  | 78.011391 | 1255  | 3.835097e+07 |
| 6  | 2021 | 강남구  | 커피-음료   | 4.380000e+11 | 54336480    | 3964065.3  | 79.759047 | 9143  | 4.788285e+07 |
| 7  | 2021 | 강남구  | 패스트푸드점  | 7.256058e+10 | 6135494     | 6618989.8  | 75.289661 | 1693  | 4.285917e+07 |
| 8  | 2021 | 강남구  | 한식음식점   | 1.390000e+12 | 42389548    | 17564633.3 | 80.476969 | 18308 | 7.572943e+07 |
| 9  | 2021 | 강남구  | 호프-간이주점 | 1.950000e+11 | 4245294     | 21029687.0 | 82.052850 | 3019  | 6.463549e+07 |
| 10 | 2021 | 개포2동 | 분식전문점   | 9.048454e+09 | 941834      | 139241.3   | 71.250000 | 75    | 1.206461e+08 |

# 06. 데이터 처리

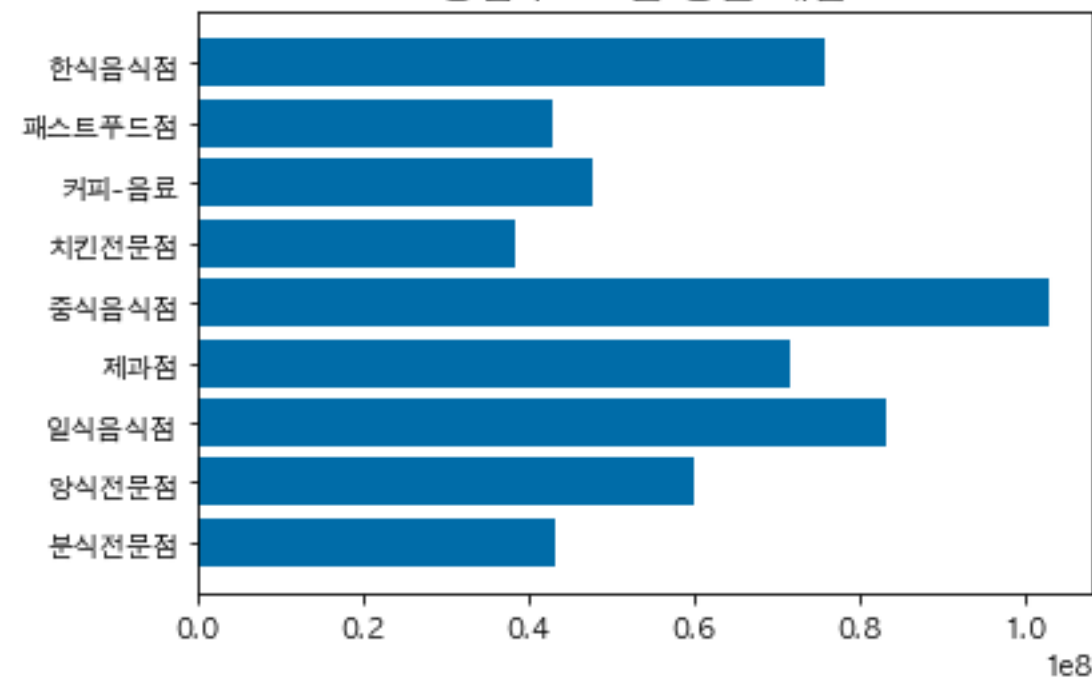
## 5) 상권관련 데이터 - 추정매출

```
plt.title('강남구 21년 평균 매출', fontsize=15)
plt.barh(mydata.iloc[0:9,2], mydata.iloc[0:9,8])
plt.title('개포2동 21년 평균 매출', fontsize=15)
plt.barh(mydata.iloc[10:19,2], mydata.iloc[10:19,8])
plt.title('개포4동 21년 평균 매출', fontsize=15)
plt.barh(mydata.iloc[20:29,2], mydata.iloc[20:29,8])
plt.title('대치4동 21년 평균 매출', fontsize=15)
plt.barh(mydata.iloc[70:79,2], mydata.iloc[70:79,8])
#...
plt.title('청담동 21년 평균 매출', fontsize=15)
plt.barh(mydata.iloc[179:188,2], mydata.iloc[179:188,8])
```

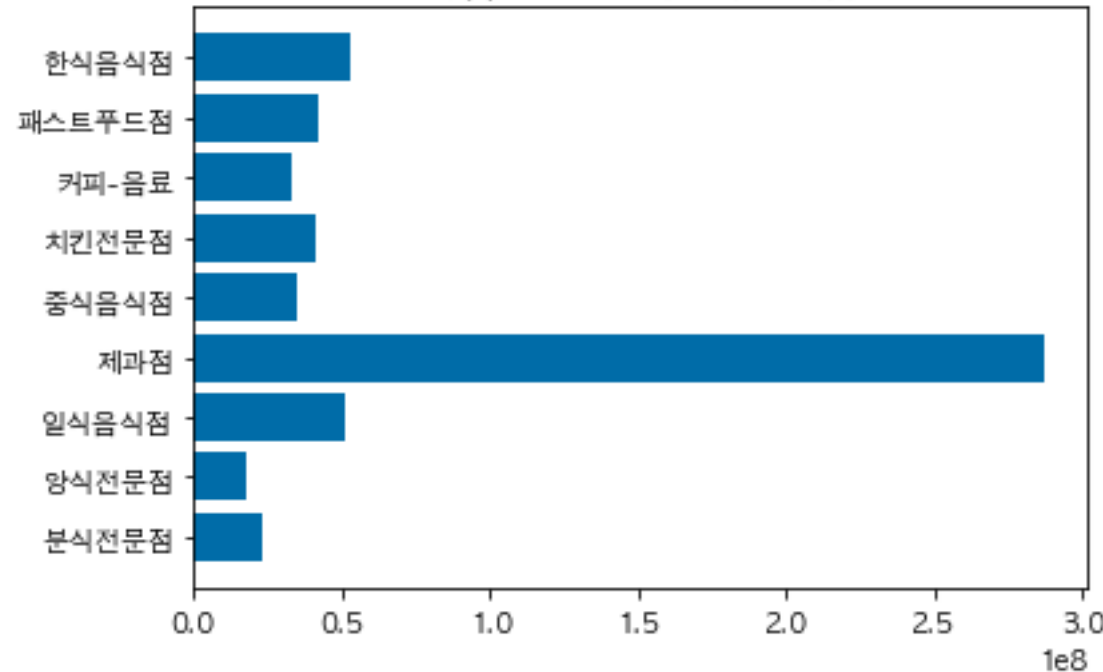
청담동 21년 평균 매출



강남구 21년 평균 매출



역삼2동 21년 평균 매출



강남구 전체 평균 매출과  
각 행정동의 업종별 평균 매출을  
그래프로 시각화

# 06. 데이터 처리

## 5) 상권관련 데이터 시각화 - 추정매출

```
# 강남구 & 개포2동
bar_width = 0.35

alpha = 0.5

index = np.arange(len(mydata.iloc[10:19,2]))

label = mydata.iloc[10:19,2]

p1 = plt.barh(index, mydata.iloc[10:19,8],

              bar_width,

              color='b',

              alpha=alpha,

              label='개포2동')

p2 = plt.barh(index+bar_width, mydata.iloc[0:9,8],

              bar_width,

              color='r',

              alpha=alpha,

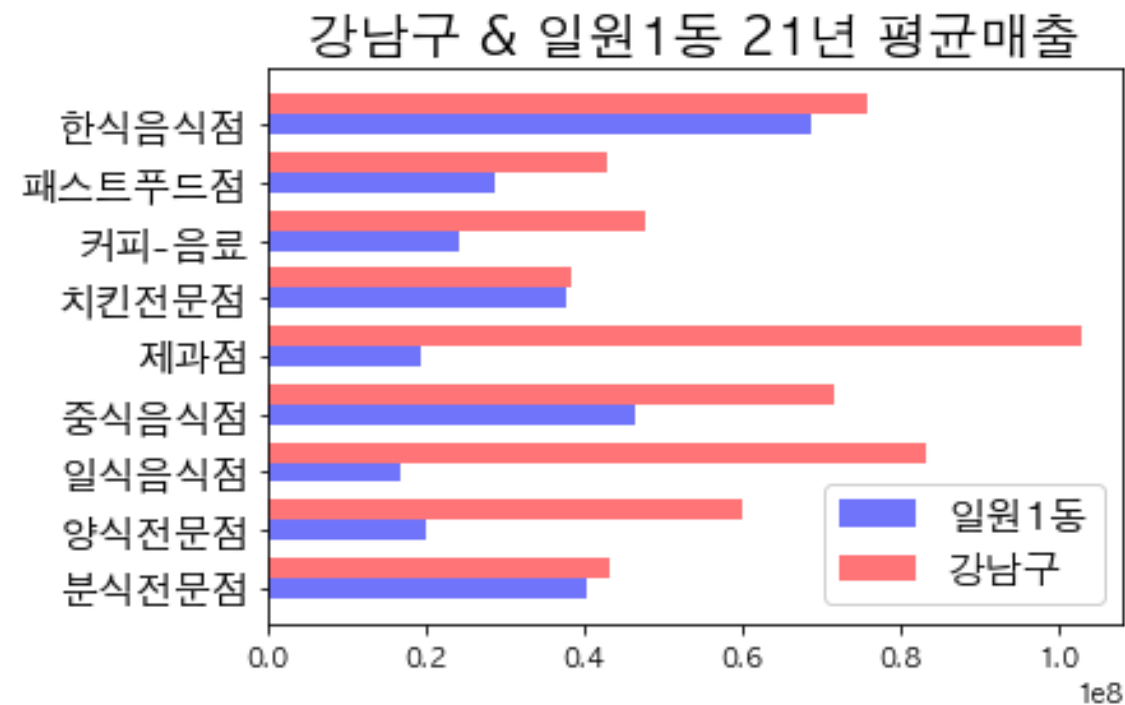
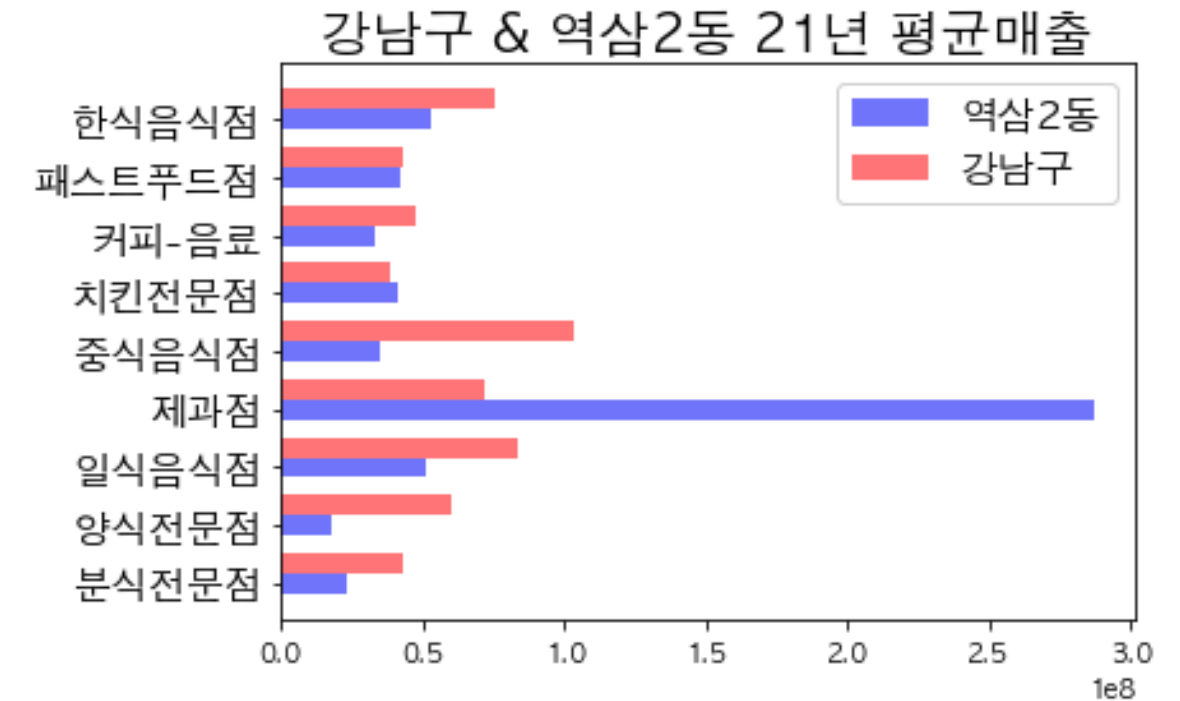
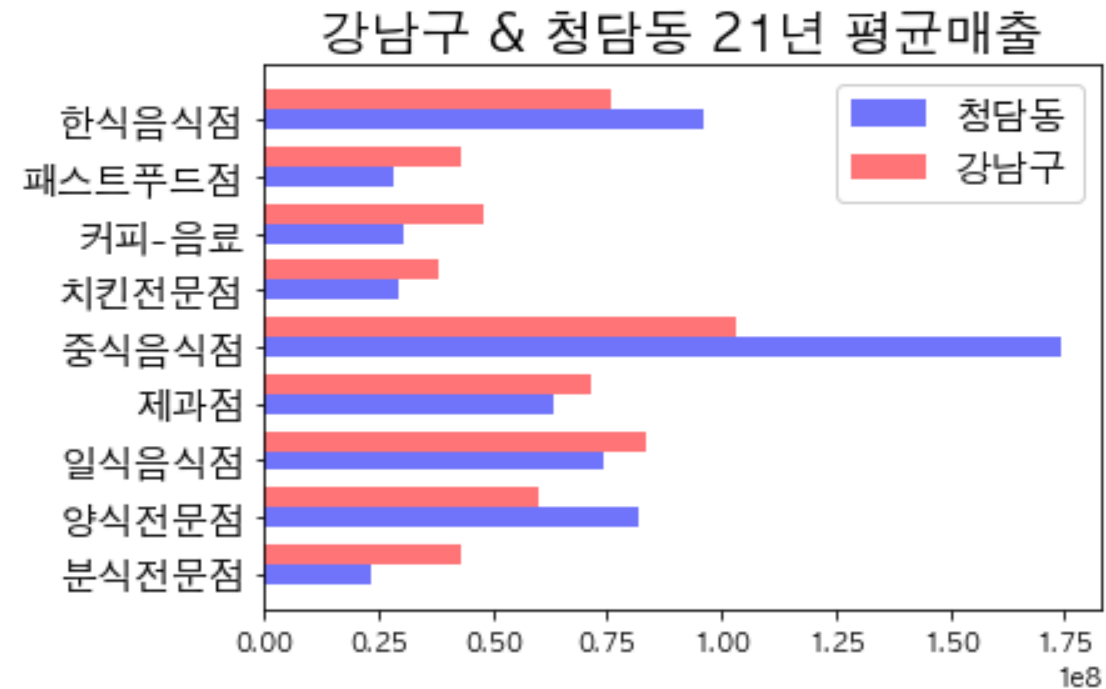
              label='강남구')

plt.title('강남구 & 개포2동 21년 평균매출', fontsize=20)

plt.yticks(index, label, fontsize=15)

plt.legend((p1[0], p2[0]), ('개포2동', '강남구'), fontsize=15)

plt.show()
```



강남구 전체 평균 매출과  
각 행정동의 업종별 평균  
매출을 비교 하는 그래프



# 06. 데이터 분석

## 1) 폐업률 및 주거인구밀도, 직장인구밀도 관계 분석

|    | X.U.FEFF.district | close        | survive      |
|----|-------------------|--------------|--------------|
| 1  | 논현1동              | 0.731486586  | -0.731486590 |
| 2  | 역삼2동              | 0.487932585  | -0.487932583 |
| 3  | 삼성2동              | 0.372851134  | -0.372851137 |
| 4  | 논현2동              | 0.202691018  | -0.202691021 |
| 5  | 대치2동              | 0.148883688  | -0.148883689 |
| 6  | 청담동               | 0.034300978  | -0.034300980 |
| 7  | 역삼1동              | -0.008101295 | 0.008101295  |
| 8  | 대치1동              | -0.090057737 | 0.090057739  |
| 9  | 압구정동              | -0.125242566 | 0.125242563  |
| 10 | 신사동               | -0.125599187 | 0.125599191  |
| 11 | 대치4동              | -0.213128148 | 0.213128150  |
| 12 | 도곡1동              | -0.455935310 | 0.455935311  |
| 13 | 도곡2동              | -0.648062840 | 0.648062838  |
| 14 | 삼성1동              | -0.652241569 | 0.652241570  |
| 15 | 수서동               | -0.703647458 | 0.703647461  |
| 16 | 개포4동              | -0.916130294 | 0.916130294  |

<close>

|    | X.U.FEFF.district | division | pop    | female | male   | X10to20 | X30to40 | X50to60 |
|----|-------------------|----------|--------|--------|--------|---------|---------|---------|
| 1  | 논현1동              | 총상촌인구    | 176016 | 90620  | 85396  | 52985   | 79081   | 43950   |
| 2  | 역삼2동              | 총상촌인구    | 235148 | 121423 | 113725 | 77360   | 96706   | 61082   |
| 3  | 삼성2동              | 총상촌인구    | 152254 | 76698  | 75556  | 45389   | 65910   | 40965   |
| 4  | 논현2동              | 총상촌인구    | 154175 | 77155  | 77020  | 43970   | 70733   | 39472   |
| 5  | 대치2동              | 총상촌인구    | 115805 | 59371  | 56434  | 41936   | 40495   | 33374   |
| 6  | 청담동               | 총상촌인구    | 83235  | 44961  | 38274  | 23593   | 34689   | 24953   |
| 7  | 역삼1동              | 총상촌인구    | 218969 | 113938 | 105031 | 69731   | 100172  | 49066   |
| 8  | 대치1동              | 총상촌인구    | 163061 | 86262  | 76799  | 72691   | 47779   | 42591   |
| 9  | 압구정동              | 총상촌인구    | 78344  | 42872  | 35472  | 24739   | 29773   | 23830   |
| 10 | 신사동               | 총상촌인구    | 89382  | 48781  | 40601  | 20099   | 35321   | 24962   |
| 11 | 대치4동              | 총상촌인구    | 255019 | 131654 | 123365 | 99618   | 93158   | 62243   |
| 12 | 도곡1동              | 총상촌인구    | 127084 | 68158  | 58926  | 39096   | 44630   | 43358   |
| 13 | 도곡2동              | 총상촌인구    | 140252 | 78675  | 61577  | 46258   | 47478   | 46516   |
| 14 | 삼성1동              | 총상촌인구    | 85746  | 42435  | 43311  | 23793   | 39682   | 22271   |
| 15 | 수서동               | 총상촌인구    | 42357  | 21556  | 20801  | 10124   | 14316   | 17917   |

<pop>

## 06. 데이터 분석

### 1) 폐업률 및 주거인구밀도, 직장인구밀도 관계 분석

```
#close의 개포1동 -> 개포1동 으로 변경
close[19,"X.U.FEFF.district"] = "개포1동"
close[19,"X.U.FEFF.district"]

#pop에서 주거인구의 밀도를 뽑아옴
pop_people = pop[pop["division"] == "총주거인구",]

pop_people <- pop_people[,c("X.U.FEFF.district", "pop")] #독립변수
close_survive <- close[, c("X.U.FEFF.district","close")] #종속변수
pop_people
close_survive

# 이 두개를 하나의 테이블로 합쳐줌
library(dplyr)
pop_close <- left_join(pop_people, close_survive, id = "X.U.FEFF.district")

#개포1동의 표본이 적어 데이터가 극단적으로 나타났기 때문에 개포1동의 행 제거
pop_close = pop_close[-19,]
View(pop_close)
```

데이터 전처리 진행

|    | X.U.FEFF.district | pop | close        |
|----|-------------------|-----|--------------|
| 1  | 논현1동              | 185 | 0.731486586  |
| 2  | 역삼2동              | 323 | 0.487932585  |
| 3  | 삼성2동              | 243 | 0.372851134  |
| 4  | 논현2동              | 149 | 0.202691018  |
| 5  | 대치2동              | 189 | 0.148883688  |
| 6  | 청담동               | 125 | 0.034300978  |
| 7  | 역삼1동              | 153 | -0.008101295 |
| 8  | 대치1동              | 317 | -0.090057737 |
| 9  | 압구정동              | 109 | -0.125242566 |
| 10 | 신사동               | 99  | -0.125599187 |
| 11 | 대치4동              | 287 | -0.213128148 |
| 12 | 도곡1동              | 213 | -0.455935310 |
| 13 | 도곡2동              | 334 | -0.648062840 |
| 14 | 삼성1동              | 76  | -0.652241569 |
| 15 | 수서동               | 96  | -0.703647458 |
| 16 | 개포4동              | 116 | -0.916130294 |
| 17 | 개포2동              | 91  | -1.047995032 |
| 18 | 일원1동              | 172 | -1.187604332 |

<pop\_close>

같은 방식으로 직장인구밀도 전처리진행

# 06. 데이터 분석

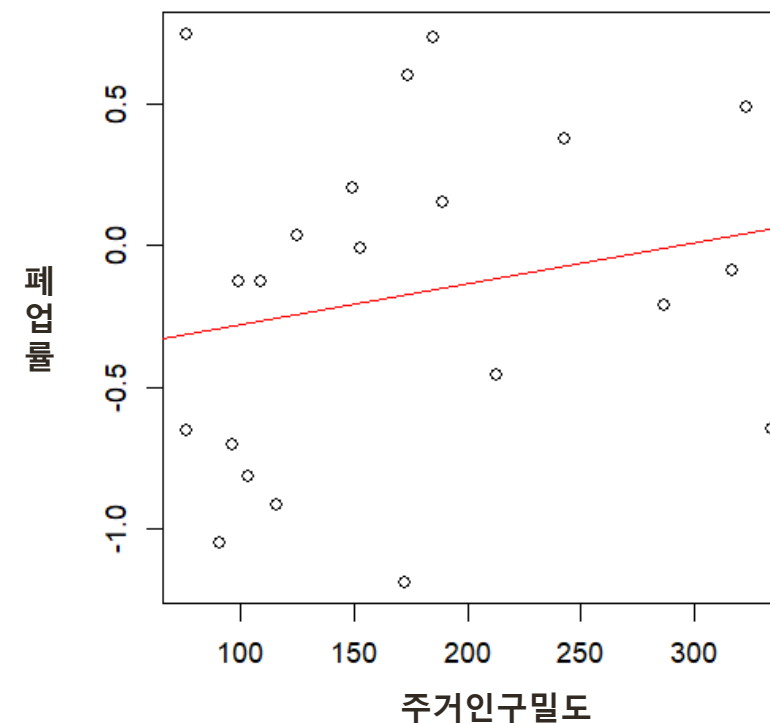
## 1) 폐업률 및 주거인구밀도, 직장인구밀도 관계 분석

```
#####상관관계 분석#####
#정규분포여부 확인
shapiro.test(pop_close$pop) #p-value : 0.02172
shapiro.test(pop_close$close) #p-value : 0.5233
#둘 다 정규분포

#상관관계 확인
cor(pop_close$pop, pop_close$close, method = "pearson") # 0.2058739
#####

#####시각화#####
library(UsingR)
plot(pop_close$pop, pop_close$close)
out = lm(pop_close$close ~ pop_close$pop, data = pop_close)
abline(out, col = "red")
#####
```

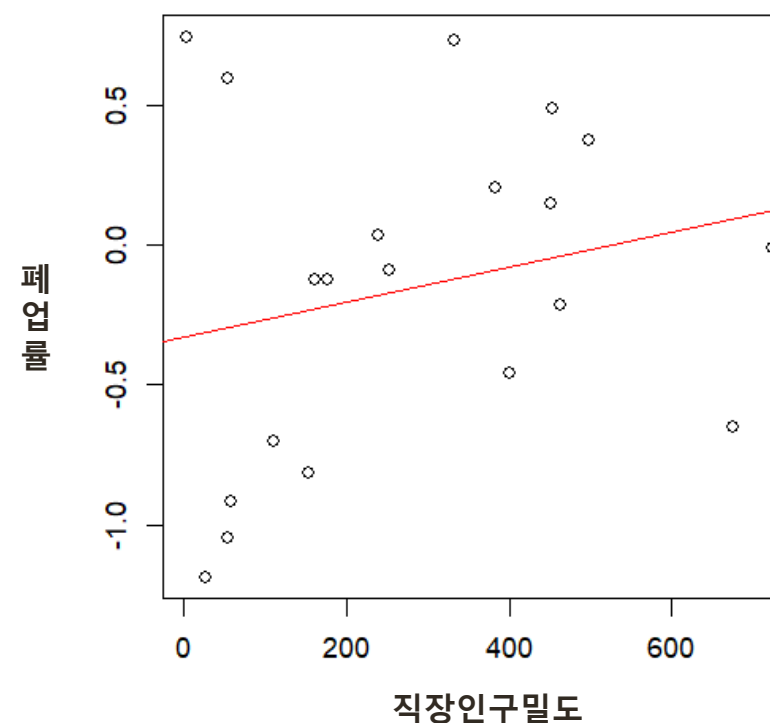
주거인구밀도와 폐업률간의 상관관계가 0.208



```
#####상관관계 분석#####
#정규분포여부 확인
shapiro.test(pop_close$pop) # p-value = 0.165
shapiro.test(pop_close$close) #p-value : 0.5233
#둘 다 정규분포

#상관관계 확인
cor(pop_close$pop, pop_close$close, method = "pearson") # 0.2292248
#####
```

직장인구밀도와 폐업률간의 상관관계가 0.229





# 06. 데이터 분석

## 2) 폐업률 및 임대료 관계 분석

### - 인구밀도와 같은 방식

```
#close의 개포1동 -> 개포1동 으로 변경
close[19,"X.U.FEFF.district"] = "개포1동"
close[19,"X.U.FEFF.district"]

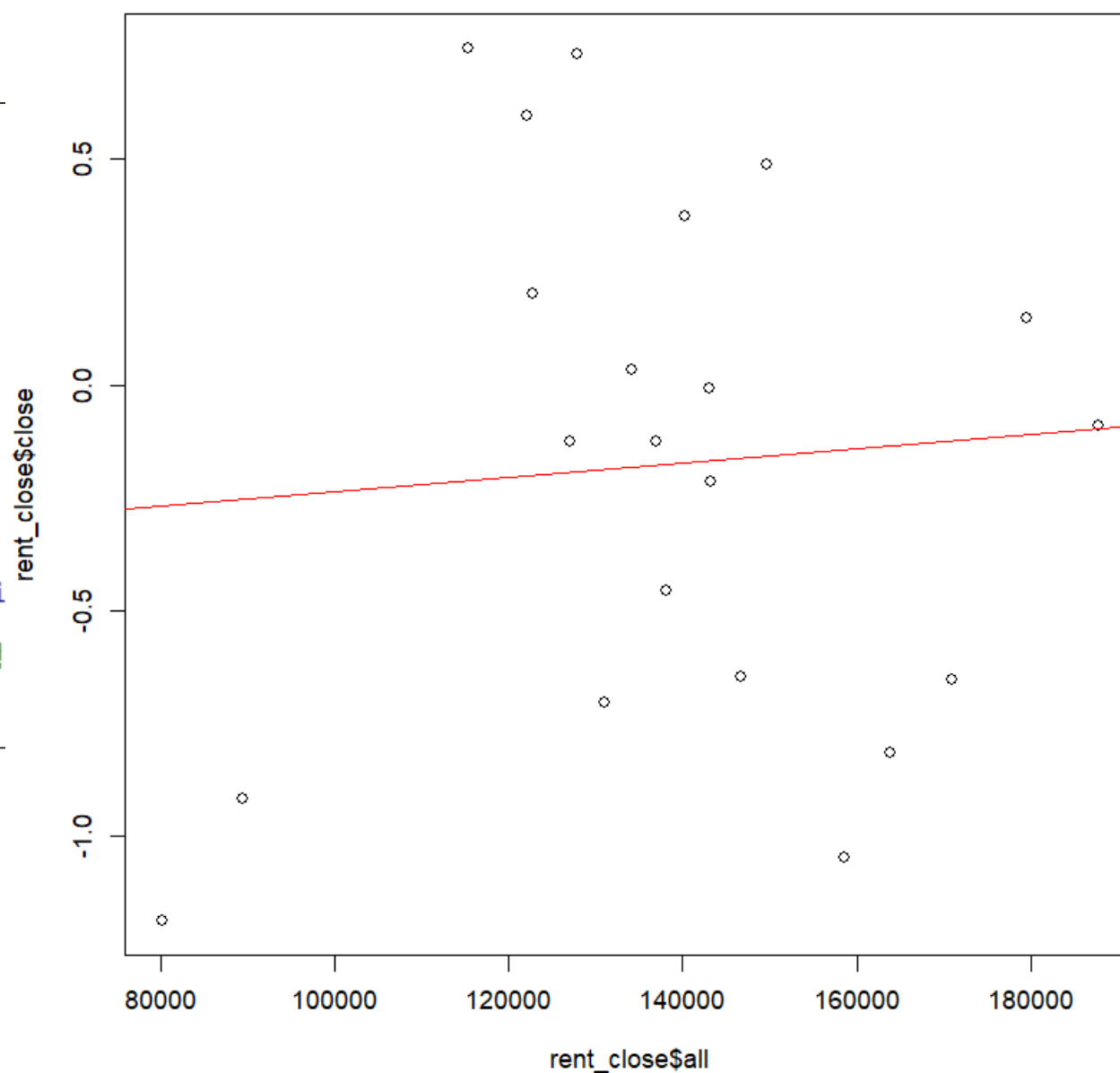
#pop에서 주거인구의 밀도를 뽑아옴
pop_people = pop[pop["division"] == "총주거인구",]

pop_people <- pop_people[,c("X.U.FEFF.district", "pop")] #독립변수
close_survive <- close[, c("X.U.FEFF.district","close")] #종속변수
pop_people
close_survive

# 이 두개를 하나의 테이블로 합쳐줌
library(dplyr)
pop_close <- left_join(pop_people, close_survive, id = "X.U.FEFF.district")

#개포1동의 표본이 적어 데이터가 극단적으로 나타났기 때문에 개포1동의 행
pop_close = pop_close[-19,]
View(pop_close)
```

데이터 전처리 진행

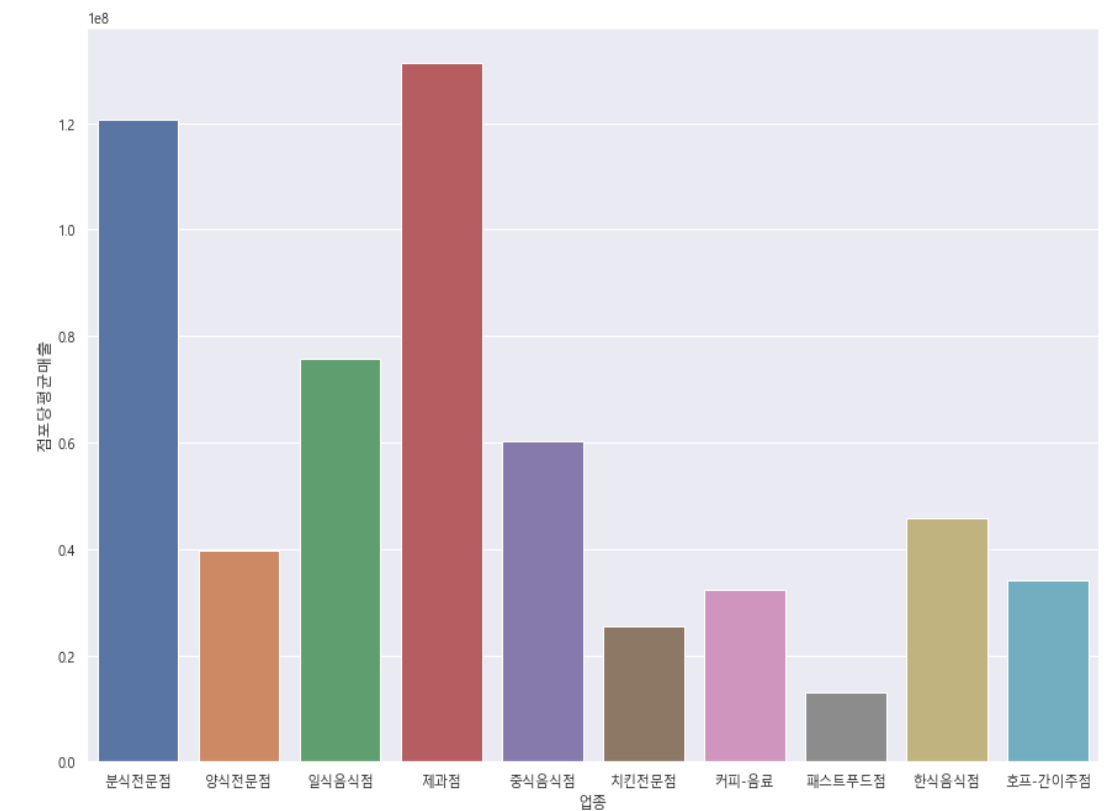
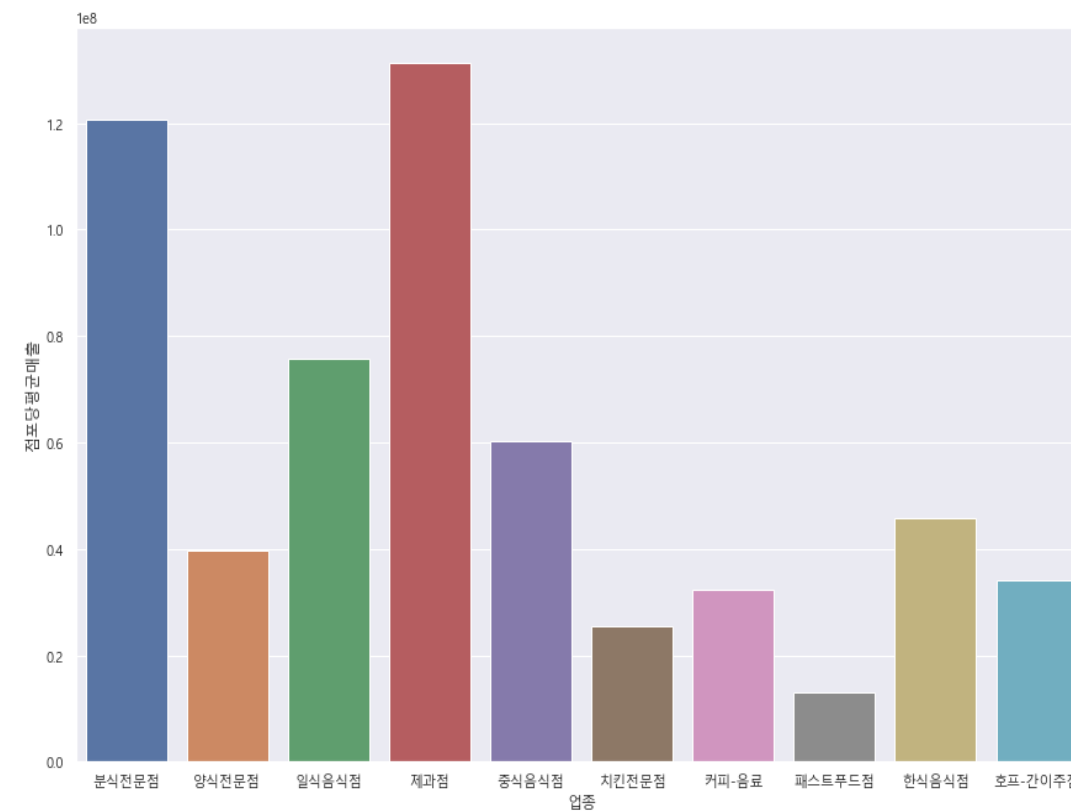
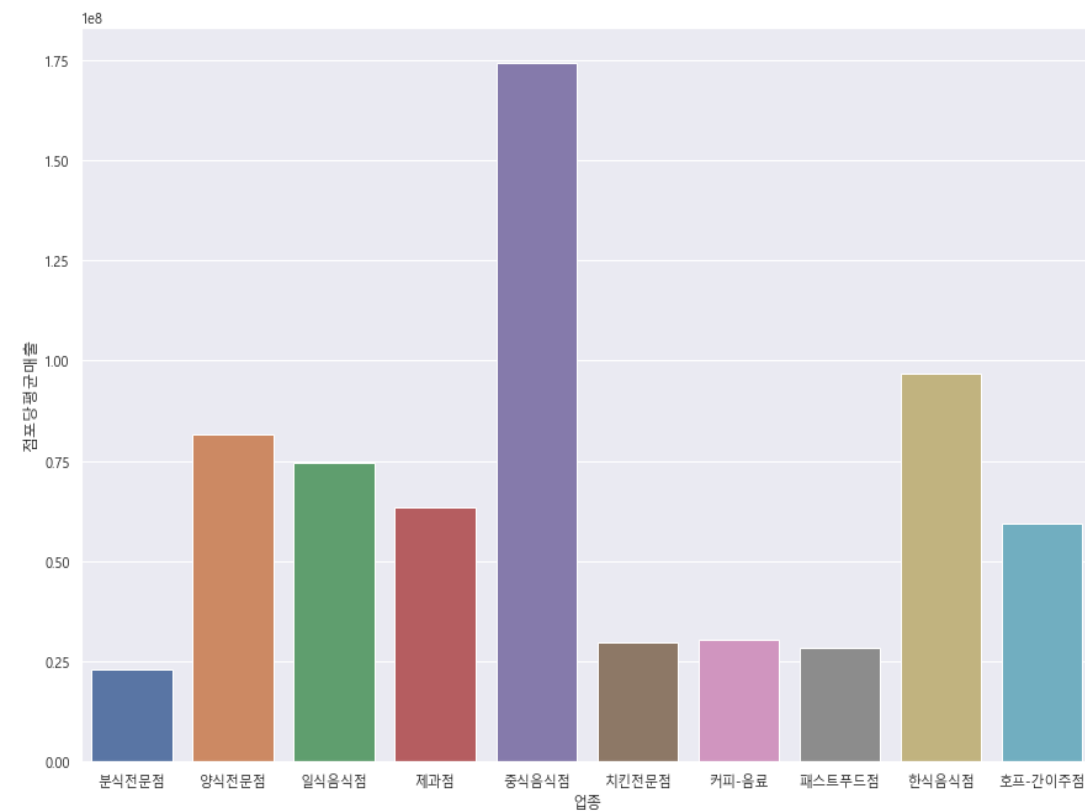


# 06. 데이터 분석

## 3) 업종별 평균 년매출에 따른 동별 추천 / 비추천 업종

```
fig, axes = plt.subplots(1,3, figsize = (50,10))
sns.barplot(x="업종", y = "점포당평균매출", data = income[income["동"] == "청담동"], ax = axes[0])
sns.barplot(x="업종", y = "점포당평균매출", data = income[income["동"] == "개포2동"], ax = axes[1])
sns.barplot(x="업종", y = "점포당평균매출", data = income[income["동"] == "개포2동"], ax = axes[2])
```

3개 정도의 동을 비교시, 동별-업종별 점포당 평균매출이 상이함을 알 수 있음



동별로 점포당 평균매출 상위2개 업종, 하위2개업종을 추천 / 비추천 업종으로 제공

## 06. 데이터 분석

### 3) 업종별 평균 년매출에 따른 동별 추천 / 비추천 업종

```
# 동별로 상위2개, 하위2개 뽑아내 dataframe으로 만들기
up_store = pd.DataFrame(columns=income.columns)
down_store = pd.DataFrame(columns=income.columns)
for dong in income["동"].drop_duplicates("동", keep="first")["동"]:
    up = income[income["동"] == dong].sort_values("점포당평균매출", ascending=False)[0:2] #상위 2개점의 매출
    down = income[income["동"] == dong].sort_values("점포당평균매출", ascending=False)[-2:] #하위 2개점의 매출
    up_store = pd.concat([up, up_store])
    down_store = pd.concat([down, down_store])
```

동별, 업종별 점포당 평균 매출금액 상위, 하위 2개매장씩 저장

```
dict = {}
for i in down_store["동"].drop_duplicates():
    list = []
    for s in down_store[down_store["동"] == str(i)]["업종"]:
        list.append(s)
    dict[i] = list
dict
```

저장한 dataframe에서, 지도에 세부정보 제공을 위해 동 - 업종 형태로 dict자료형으로 저장

```
{'청담동': ['패스트푸드점', '분식전문점'],
'일원1동': ['제과점', '일식음식점'],
'역삼2동': ['분식전문점', '양식전문점'],
'역삼1동': ['분식전문점', '패스트푸드점'],
'압구정동': ['치킨전문점', '패스트푸드점'],
'신사동': ['치킨전문점', '제과점'],
'수서동': ['호프-간이주점', '제과점'],
'삼성2동': ['치킨전문점', '패스트푸드점'],
'삼성1동': ['커피-음료', '치킨전문점'],
'도곡2동': ['치킨전문점', '패스트푸드점'],
'도곡1동': ['커피-음료', '패스트푸드점'],
'대치4동': ['패스트푸드점', '양식전문점'],
'대치2동': ['분식전문점', '양식전문점'],
'대치1동': ['호프-간이주점', '치킨전문점'],
'논현2동': ['양식전문점', '패스트푸드점'],
'논현1동': ['분식전문점', '양식전문점'],
'개포4동': ['분식전문점', '치킨전문점'],
'개포2동': ['치킨전문점', '패스트푸드점'],
'강남구': ['패스트푸드점', '치킨전문점']}
```



## 06. 데이터 분석

### 3) 업종별 평균 년매출에 따른 동별 추천 / 비추천 업종

- 동별 업종별 평균 매출의 순위

| 추천  | 개포2동    | 개포4동    | 논현1동    | ... | 역삼2동    | 일원1동    | 청담동     |
|-----|---------|---------|---------|-----|---------|---------|---------|
|     | 제과점     | 패스트푸드점  | 호프-간이주점 |     | 제과점     | 한식음식점   | 중식음식점   |
|     | 분식전문점   | 중식음식점   | 중식음식점   |     | 한식음식점   | 중식음식점   | 한식음식점   |
|     | 일식음식점   | 한식음식점   | 일식음식점   |     | 일식음식점   | 분식전문점   | 양식전문점   |
|     | 중식음식점   | 제과점     | 한식음식점   |     | 패스트푸드점  | 치킨전문점   | 일식음식점   |
|     | 한식음식점   | 일식음식점   | 제과점     |     | 치킨전문점   | 패스트푸드점  | 제과점     |
|     | 양식전문점   | 커피-음료   | 커피-음료   |     | 중식음식점   | 호프-간이주점 | 호프-간이주점 |
|     | 호프-간이주점 | 양식전문점   | 패스트푸드점  |     | 커피-음료   | 커피-음료   | 커피-음료   |
|     | 커피-음료   | 호프-간이주점 | 치킨전문점   |     | 호프-간이주점 | 양식전문점   | 치킨전문점   |
|     | 치킨전문점   | 분식전문점   | 분식전문점   |     | 분식전문점   | 제과점     | 패스트푸드점  |
| 비추천 | 패스트푸드점  | 치킨전문점   | 양식전문점   |     | 양식전문점   | 일식음식점   | 분식전문점   |

# 07. 결과 및 웹구현

## - 지도 스타일링

```
##### 지도
m = folium.Map(location=[37.498095, 127.057610], zoom_start=12,
               tiles='cartodbpositron')
choropleth = folium.GeoJson(data=geo_path, style_function=style_function,
                             m.add_child(choropleth))

for i in range(len(geo_str['features'])):
    geometry = geo_str['features'][i]['geometry']
    dong_name = geo_str['features'][i]['properties']['adm_nm']
    get_info(dong_name) #처음한번만
    popup = folium.Popup('<div style="overflow: scroll; width: 100%; height: 500px; padding: 10px;">'+
                        '<h3>'+dong_name+'</h3><br>'+
                        '<h4>&bull; '+dong_name+'의 추천 상위 매출 업종 1순위는 <u>'+str(high[dong_name][0])+'</u>, 2순위는 <u>'+str(high[dong_name][1])+'</u>,</h4><br>'+
                        '<h4>&bull; '+dong_name+'의 비추천 하위 매출 업종 1순위는 <u>'+str(low[dong_name][0])+'</u>, 2순위는 <u>'+str(low[dong_name][1])+'</u>,</h4><br>'+
                        '<br><CENTER><CENTER><br>'+
                        '<CENTER><CENTER><br>'+
                        ''+
                        '</div>',width=500, height=2000)

    tooltip = folium.Tooltip(dong_name)
    state_marker = folium.GeoJson(data=geo_str['features'][i]['geometry'],
                                  style_function=lambda x: style_1,
                                  highlight_function=highlight_style)

    state_marker.add_child(popup)
    state_marker.add_child(tooltip)
    m.add_child(state_marker)

m
m.save("figures/dong_map.html")
```

```
def highlight_style(feature):
    """
    style_function for when choropleth button
    is highlighted
    """
    return {'fillOpacity': 1,
            'weight': 1,
            'fillColor': '#000000',
            'color': '#000000'}
```

```
def style_function(feature):
    dong = feature['properties']['adm_nm']
    style = state_style(dong, year=2019, function=True)

    return style
```

```
def state_style(dong, year, function=False):

    state_results = seoulin
    yr = str(year)
    #Set state colour
    if state_results[yr][dong] >= 0:
        color = '#2aae2a' #green
    else:
        color = '#ff4040' #red

    #Set state style
    if function == False:
        # Format for style_dictionary
        state_style = {
            'opacity': 0.5,
            'color': color,
        }
    else:
        # Format for style_function
        state_style = {
            'fillOpacity': 0.5,
            'weight': 0.5,
            'fillColor': color,
            'color': '#000000'}

    return state_style
```

# 07. 결과

## - 막대그래프 제작

```
##### 지도
m = folium.Map(location=[37.498095, 127.057610], zoom_start=12,
               tiles='cartodbpositron')
choropleth = folium.GeoJson(data= geo_path, style_function=style_function, )
m.add_child(choropleth)

for i in range(len(geo_str['features'])):
    geometry = geo_str['features'][i]['geometry']
    dong_name = geo_str['features'][i]['properties']['adm_nm']
    get_info(dong_name) #처음한번만
    popup = folium.Popup('<div style="overflow: scroll; width: 100%; height: 500px; padding: 10px;">'+
        '<h3>'+dong_name+'</h3><br>'+
        '<h4>&bull; '+dong_name+'의 추천 상위 매출 업종 1순위는 <u>'+str(high[dong_name][0])+</u>, 2순위는 <u>'+str(high[dong_name][1])+</u>,<br>'+
        '<h4>&bull; '+dong_name+'의 비추천 하위 매출 업종 1순위는 <u>'+str(low[dong_name][0])+</u>, 2순위는 <u>'+str(low[dong_name][1])+</u>.<br>'+
        '<br><CENTER><CENTER><br>'+
        '<CENTER><CENTER><br>'+
        ''+
        '</div>',width=500, height=2000)

    tooltip = folium.Tooltip(dong_name)
    state_marker = folium.GeoJson(data=geo_str['features'][i]['geometry'],
                                  style_function=lambda x:style1,
                                  highlight_function = highlight_style)

    state_marker.add_child(popup)
    state_marker.add_child(tooltip)
    m.add_child(state_marker)

m
m.save("figures/dong_map.html")
```

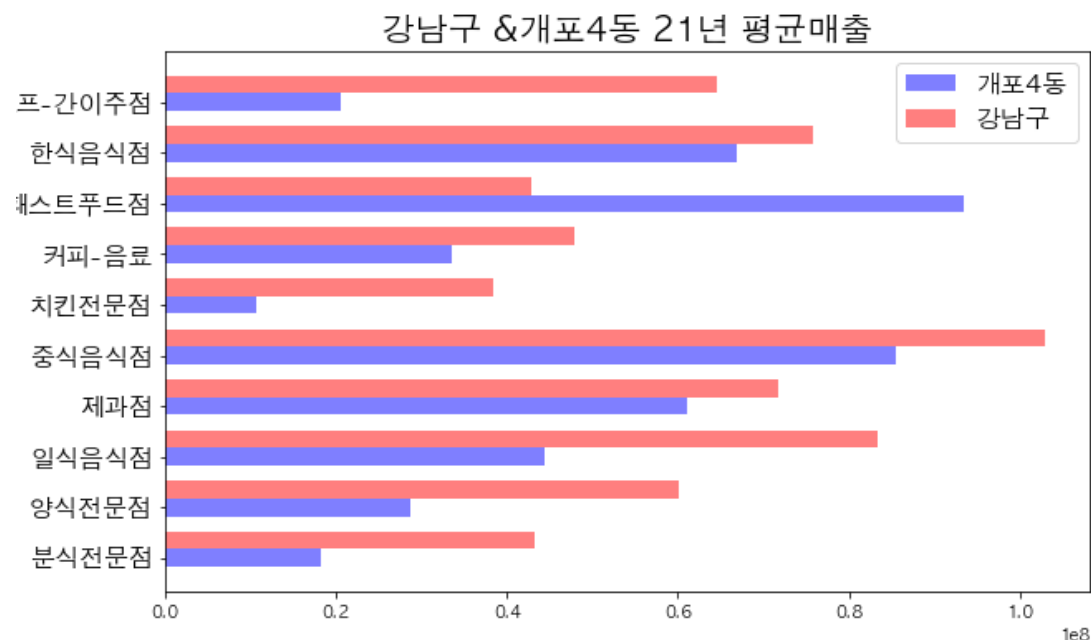
```
# 지도에 그래프 넣기
def get_info(dong_name):

    fig = plt.figure(figsize=(10,6))

    a = mydata[mydata['동']==dong_name]['업종']
    a = a.to_list()
    if a == []:
        plt.title(dong_name+' 준비중 입니다',fontsize=15)
        plt.barh(mydata[mydata['동']==dong_name]['업종'], mydata[mydata['동']==dong_name]['점포당 연간매출금액'])
        html = mpld3.fig_to_html(fig)
        iframe = folium.IFrame(html=html,width = 600, height = 300)
        plt.savefig('figures/'+dong_name+'.png')

    if a != []:
        bar_width = 0.35
        alpha = 0.5
        index = np.arange(len(mydata[mydata['동']==dong_name]['업종']))
        label = mydata[mydata['동']==dong_name]['업종']
        p1 = plt.barh(index, mydata[mydata['동']==dong_name]['점포당 연간매출금액'],
                     bar_width,
                     color='b',
                     alpha=alpha,
                     label=dong_name)
        p2 = plt.barh(index+bar_width, mydata[mydata['동']==dong_name]['점포당 연간매출금액'],
                     bar_width,
                     color='r',
                     alpha=alpha,
                     label='강남구')
        plt.title('강남구 &'+dong_name+' 21년 평균매출', fontsize=20)
        plt.yticks(index, label, fontsize=15)
        plt.legend((p1[0], p2[0]), (dong_name, '강남구'), fontsize=15)

        #plt.title(dong_name+' 21년 평균 매출',fontsize=15)
        #plt.barh(mydata[mydata['동']==dong_name]['업종'], mydata[mydata['동']==dong_name]['점포당 연간매출금액'])
        html = mpld3.fig_to_html(fig)
        iframe = folium.IFrame(html=html,width = 600, height = 300)
        plt.savefig('figures/'+dong_name+'.png')
    return iframe
```





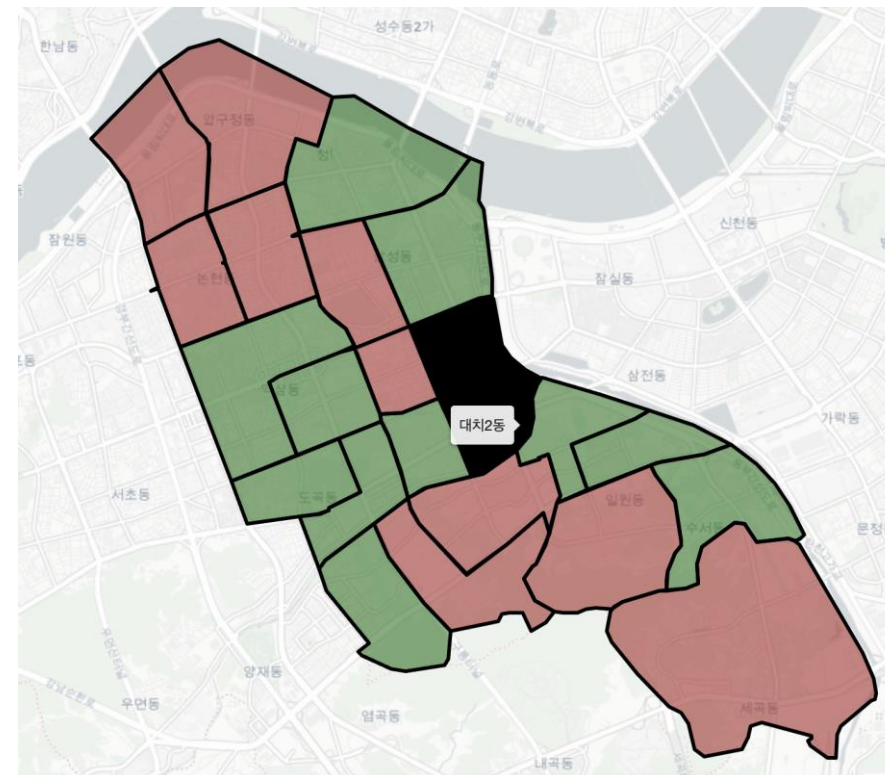
## 07. 결과

## - 창업신호등 시각화

```
##### 지도
m = folium.Map(location=[37.498095, 127.057610], zoom_start=12,
                tiles='cartodbpositron')
choropleth = folium.GeoJson(data= geo_path, style_function=style_function, )
m.add_child(choropleth)

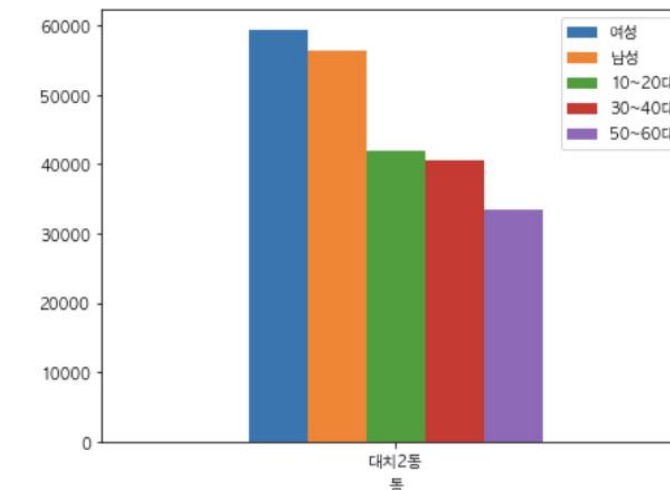
for i in range(len(geo_str['features'])):
    geometry = geo_str['features'][i]['geometry']
    dong_name = geo_str['features'][i]['properties']['adm_nm']
    get_info(dong_name) #처음한번만
    popup = folium.Popup('<div style="overflow: scroll; width: 100%; height: 500px; padding: 10px;">'+
        '<h3>'+dong_name+'</h3><br>'+
        '<h4>&bull; ' +dong_name+'의 추천 상위 매출 업종 1순위는 <u>'+str(high[dong_name][0])+'</u>, 2순위는 <u>'+
        '<h4>&bull; ' +dong_name+'의 비추천 하위 매출 업종 1순위는 <u>'+str(low[dong_name][0])+'</u>, 2순위는 <u>'+
        '<br><CENTER><br>'+
        '<CENTER><CENTER><br>'+
        ''+
        '<div>',width=500, height=2000)
    tooltip = folium.Tooltip(dong_name)
    state_marker = folium.GeoJson(data=geo_str['features'][i]['geometry'],
                                   style_function=lambda x:style1,
                                   highlight_function = highlight_style)
    state_marker.add_child(popup)
    state_marker.add_child(tooltip)
    m.add_child(state_marker)

m
m.save("figures/dong_map.html")
```



대치2동

- 대치2동의 추천 상위 매출 업종 1순위는 제과점, 2순위는 한식음식점입니다.(2021년 기준)
- 대치2동의 비추천 하위 매출 업종 1순위는 분식전문점, 2순위는 양식전문점입니다.(2021년 기준)



| 년도    | 2019  | 2020  | 2021   |
|-------|-------|-------|--------|
| 운영연개월 | 94.43 | 99.09 | 104.91 |





## 07. 결과

결과 (데모) → 웹

**Q&A**

**감사합니다**