

빅데이터 기반경영 / 사업타당성분석

# 마포구 상권내 업종 시너지분석

---

COVID-19 위기상황 전후로의 업종 시너지 비교

3조 황희선 남유정 한주엽



# 목차

## 01 주제설명

---

- 프로젝트 목적
- 기존 연구들과의 차별점
- 시너지 정의

## 02 데이터 분석

---

- 데이터 전처리
- 데이터 분석 과정
- 데이터 분석 결과

## 03 결론

---

- 결론 도출
- 프로젝트 시사점

## 04 Reference

---

## 05 Q&A

---



### 목적

---

마포구 상권 내의 빈도가 높은  
업종의 시너지 분석

코로나와 같은 위기 상황에도  
기존의 시너지가 유지되는가에  
대하여 검증

코로나 이후, 또 다른 위기 상황에서도  
강건하게 버텨낼 수 있는  
업종별 시너지를 찾아  
소상공인들의 점포 입지 선정에  
도움을 주고자 함

# 01. 주제설명

## 기존 연구들과의 차별점

### 공통점

- 상권 내의 업종 간의 시너지를 분석
- 매출과 연계소비를 중점으로 분석

### 차별점

- 기존 시너지를 COVID-19라는 위기 상황을 기준 시점으로 잡고 나누어 검증
- 기존 시너지 검증을 위해 데이터에 대한 광범위하고 개괄적인 분석

# 01. 주제설명

시너지 정의

기존의 선행연구를 활용하여 시너지를 정의

01

## 매출과의 상관관계

업계 평균과 비교하여  
매출이 유의미하게  
상승한 경우

02

## 연계소비

업종 간의 연계소비가  
발생하는 경우

# 01. 주제설명

기존 시너지 활용

기존 시너지 업종 조합

한식 - 커피 전문점  
기타 유흥업소 - 노래방  
한식 - 기타 요식업  
미용서비스 - 미용실  
화장품 - 미용실  
미용실 - 안경점  
커피 전문점 - 미용실  
화장품-안경

## 02. 데이터 분석

## 분석 툴 및 데이터

### 분석 툴



EXCEL, PYTHON, QGIS

### 사용 데이터



빅데이터 캠퍼스 신한카드 내외국인 매출 데이터

소상공인시장진흥공단 상권 데이터

### 1. 패키지 설치

```
# 위-경도 좌표변환을 위한
from pyproj import CRS
from pyproj import Transformer
from pyproj import Proj, transform
# 파일 입출력, 디렉토리 관리
import io
import os
# 데이터 전처리 등 기본 핸들링 위한
import pandas as pd
import numpy as np
import math
from sklearn.preprocessing import LabelEncoder
```

```
# 반복문 진행률 시각화
import time
from tqdm import tqdm
# 데이터 시각화
import seaborn as sns
import matplotlib.pyplot as plt
# k-means 군집화 알고리즘
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_samples
```

데이터 분석을  
위한 패키지 설치



## 2. 데이터 청크 나누기

```
path_dir = 'E:/1. 내국인(블록)_일자별시간대별/' # 카드소비 데이터 폴더위치
file_list = os.listdir(path_dir) # 카드소비 폴더내의 파일들 이름
print(file_list[24:-10] # 19년1월~20년12월)

total_seoul_sell = pd.DataFrame()
total_seoul_sell_1 = pd.DataFrame()

# 19년 1,3월
for file_path in tqdm(file_list[24:26]): # 1~3월
    path = path_dir+file_path
    a = pd.read_csv(path, sep='|', encoding='utf-8', quotechar='"')
    total_seoul_sell_1 = pd.concat([total_seoul_sell_1, a], axis=0, ignore_index=True)
total_seoul_sell_1.shape
```

```
100%|██████████████████████████████████████████████████████████████████████████████| 3/3 [01:01<00:00, 20.79s/it]
(52847292, 8)
```

과부화를 피하기 위해 데이터 크기를 나눔

### 3. 위치 데이터 처리

```
def transfer_crs(block_total):  
    # ※ 본 데이터의 좌표계는 Korea 2000/Central Belt(EPsg:5181) 입니다.  
    # SHP상에 나와있는 EPSG:5181 X,Y 좌표를 WGS84 위경도 좌표계로 변환  
  
    proj_1 = Proj(init='epsg:5181') # 카드 매출 데이터의 shp 좌표 기준(매출데이터는 블록코드만 나와있으니 블록지역 경계도 써야함)  
    proj_2 = Proj(init='epsg:4326') # 변환코자 하는 위경도 좌표 기준  
    block_total = block_total.loc[:,['엑스좌표(XCNTS)', '와이좌표(YDNST)']] # 엑스,와이 좌표만 불러올 수 있도록(for 메모리 효율성)  
    DataFrame = block_total.copy() # 서울시 블록 지역 경계도 파일  
  
    df = pd.DataFrame()  
    x_list = []  
    y_list = []  
    for idx, row in DataFrame.iterrows():  
        x, y = row['엑스좌표(XCNTS)'], row['와이좌표(YDNST)']  
        x_, y_ = transform(proj_1, proj_2, x, y)  
        x_list.append(x_)  
        y_list.append(y_)  
    df['lon'] = x_list  
    df['lat'] = y_list  
  
    return df
```

CRS를 통해 카드소비블록의 좌표를 위도,경도로 변환

4. 데이터 통일작업 진행

```
store_upzong_mapo_cluster_inner = pd.merge(store_upzong_mapo_cluster, johap, left_on='상권업종중분류코드', right_on='신한코드', how='inner')
store_upzong_mapo_cluster_inner
```

	위도	경도	상권업종중분류명	상권업종중분류코드	상권업종소분류코드	상권업종소분류명	동정보	상호명	시군구명	건물명	cluster	신한업종	신한코드	매칭키
0	37.549	126.952	학원-보습교습입시	R01	R01A01	학원-입시	아현동	무비디자인	마포구	1111	189	학원	R01	SB047
1	37.5449	126.949	학원-보습교습입시	R01	R01A01	학원-입시	아현동	더바운스	마포구	131	218	학원	R01	SB047
2	37.5565	126.955	학원-보습교습입시	R01	R01A01	학원-입시	아현동	바른부동산	마포구	131	223	학원	R01	SB047
3	37.5556	126.956	학원-보습교습입시	R01	R01A01	학원-입시	아현동	사랑방	마포구	1477	285	학원	R01	SB047

	시도	시군구	행정구역명	행정동	법정동	행정구역분류 (통계청)	행정동코드	행정기관코드 (행정안전부)	행정동코드	법정동코드	관할지역	행정동	영문명칭	Unnamed: 10
0	서울특별시	서울특별시	서울특별시	서울특별시	서울특별시		11	1100000000	1100000000		NaN	Seoul		NaN
1	서울특별시	종로구	종로구	종로구	종로구		11010	1111000000	1111000000		NaN	Jongno-gu		NaN
2	서울특별시	종로구	청운효자동	청운효자동	청운동		1101072	1111051500	1111010100		NaN	Cheongunhyoja-dong		NaN
3	서울특별시	종로구	청운효자동	청운효자동	신교동		1101072	1111051500	1111010200		NaN	Cheongunhyoja-dong		NaN
4	서울특별시	종로구	청운효자동	청운효자동	궁정동		1101072	1111051500	1111010300		NaN	Cheongunhyoja-dong		NaN

```
# 마포동 동코드 구하기
mapodong_code = dong_code[dong_code['시군구']=='마포구']

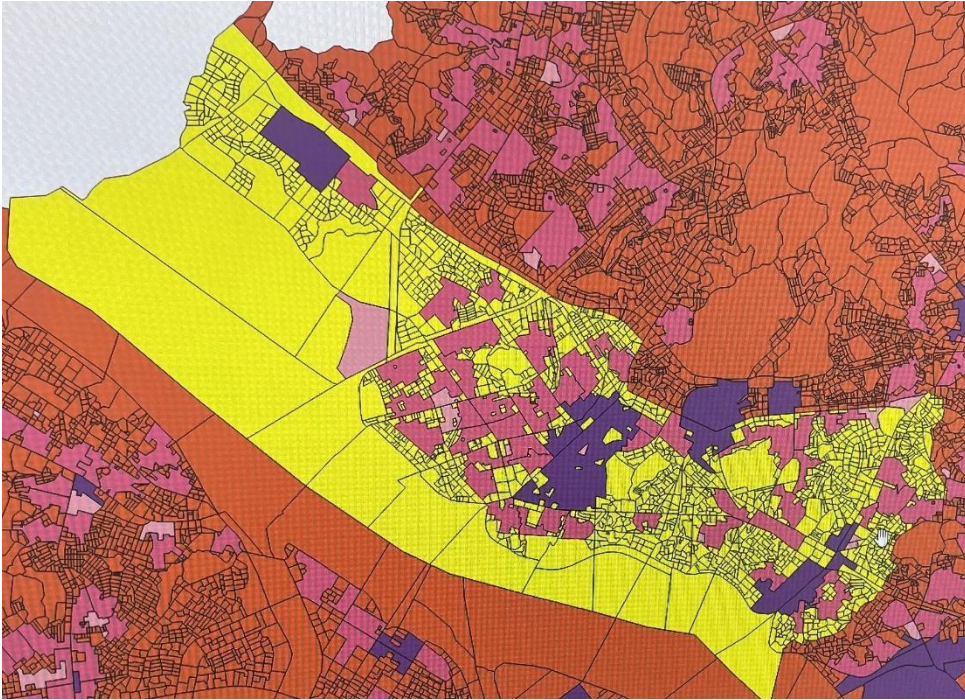
mapodong_code_df = pd.DataFrame(mapodong_code['행정동']).reset_index()
mapodong_code_df['동코드'] = np.int64(mapodong_code['행정기관코드(행정안전부 행정동코드)']/100) # 코드 숫자 통일
del mapodong_code_df['index']
mapodong_code_df.drop_duplicates(inplace=True)
mapodong_code_df.reset_index(inplace=True)
del mapodong_code_df['index']
mapodong_code_df = mapodong_code_df.iloc[1:] # 동 코드중 첫번째 행(마포구) 제거
mapodong_code_df
```

소상공인 진흥공단에서 제공하는 데이터와  
빅데이터캠퍼스에서 제공하는 데이터의  
업종분류, 행정동 분류가 다르기 때문에  
이를 통일시켜주는 작업을 진행

## 02. 데이터 분석

## 데이터 전처리

### 4\_참고. 빅데이터 캠퍼스 데이터 위치정보(블록) 시각화



- 빅데이터 캠퍼스에서 제공하는 데이터에 기반하여 마포구를 블록단위로 시각화

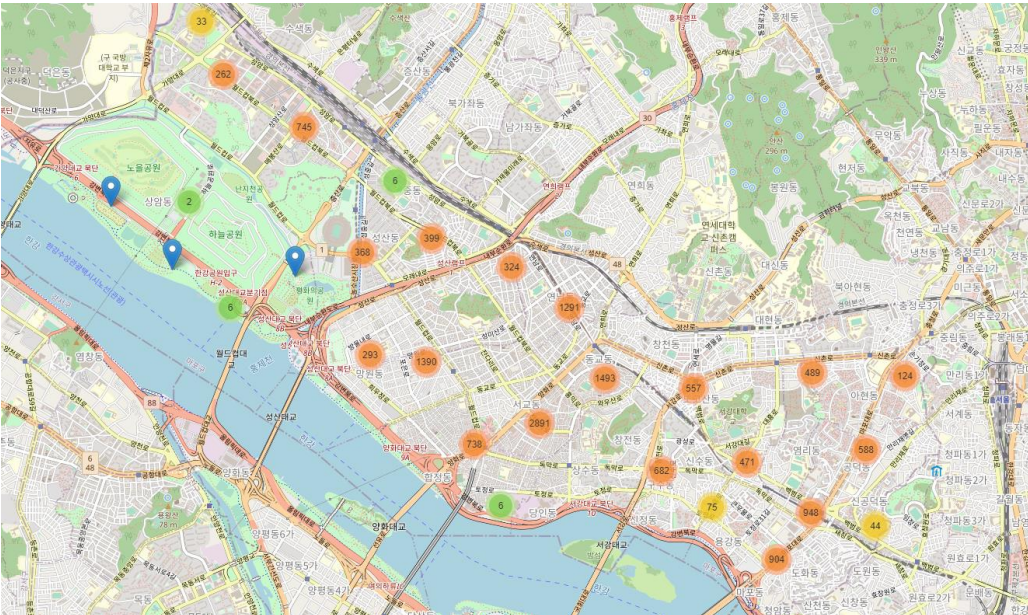


# 02. 데이터 분석

# 데이터 전처리

## 5. 마포구 상권내 상가업소 군집화\_상권데이터

20	1.7E+07	LAD	Q	음식	Q01	한식	Q01A01	한식/백반/56111	한식 음식?	11	서울특별시	11440	마포구	1.1E+09	서교동	1.1E+09	서교동	1.1E+18	1	대지	395	112	서울특별시	1.1E+1			
23	1.7E+07	온진이네	Q	음식	Q01	한식/백반/56111	한식 음식?	11	서울특별시	11440	마포구	1.1E+09	공덕동	1.1E+09	아현동	1.1E+18	1	대지	437	3	서울특별시	1.1E+1					
27	2.5E+07	헬빙	F	생활서비스	F01	이/미용/건	F01A03	비안/피부관리		11	서울특별시	11440	마포구	1.1E+09	용강동	1.1E+09	영리동	1.1E+18	1	대지	172	14	서울특별시	1.1E+1			
37	2.5E+07	노벨수학교습소	R	학원/교육	R01	학원/보습	R01A01	학원/입시	P85501	일반	교과		11	서울특별시	11440	마포구	1.1E+09	아현동	1.1E+09	공덕동	1.1E+18	1	대지	209	1	서울특별시	1.1E+1
52	1.6E+07	불국일음	D	소매	D01	음/식료품	D01A20	음료/가게	G47221	음료	소매		11	서울특별시	11440	마포구	1.1E+09	합정동	1.1E+09	합정동	1.1E+18	1	대지	393	18	서울특별시	1.1E+1
99	2.4E+07	대룡치킨	R	학원/교육	R10	도서관/독	R10A01	독서실	R90212	독서실	온		11	서울특별시	11440	마포구	1.1E+09	대흥동	1.1E+09	대흥동	1.1E+18	1	대지	161		서울특별시	1.1E+1
146	2.9E+07	한샘아현키아현키친프	D	소매	D03	종합소매점	D03A01	편의점	G47122	제인화 편의		11	서울특별시	11440	마포구	1.1E+09	공덕동	1.1E+09	아현동	1.1E+18	1	대지	692		서울특별시	1.1E+1	
167	1.6E+07	복음보청기 마포센터	D	소매	D20	의약/의료	D20A02	보청기판매	G47812	의료용 기		11	서울특별시	11440	마포구	1.1E+09	서교동	1.1E+09	서교동	1.1E+18	1	대지	441	2	서울특별시	1.1E+1	
190	2.1E+07	매마드커피	Q	음식	Q12	커피점/커피	Q12A01	커피전문점	S6220	비알콜 음		11	서울특별시	11440	마포구	1.1E+09	서교동	1.1E+09	서교동	1.1E+18	1	대지	327	15	서울특별시	1.1E+1	
217	2.5E+07	에이블씨연씨	D	소매	D16	화장품소매	D16A01	화장품판매	G47813	화장품 및		11	서울특별시	11440	마포구	1.1E+09	서강동	1.1E+09	창전동	1.1E+18	1	대지	145	15	서울특별시	1.1E+1	
265	1.6E+07	COUPGROUP	D	소매	D21	절물/산방	D21A06	절물/간식	G47511	절물 및 난		11	서울특별시	11440	마포구	1.1E+09	공덕동	1.1E+09	신공덕동	1.1E+18	1	대지	167		서울특별시	1.1E+1	
269	2.9E+07	복음보청기 마포센터	Q	음식	Q10	발식/퓨전	Q10A01	죽전문점	S6111	한식 음식?		11	서울특별시	11440	마포구	1.1E+09	아현동	1.1E+09	공덕동	1.1E+18	1	대지	467		서울특별시	1.1E+1	
318	2.2E+07	태화빌딩센터	N	관공/여가	N01	PC/오락/엔	N01A06	볼링장	R91134	볼링장 온		11	서울특별시	11440	마포구	1.1E+09	서교동	1.1E+09	서교동	1.1E+18	1	대지	343	2	서울특별시	1.1E+1	
330	2.8E+07	삼국육해공	Q	음식	Q01	한식	Q01A02	갈비/삼갈	S6111	한식 음식?		11	서울특별시	11440	마포구	1.1E+09	망원동	1.1E+09	망원동	1.1E+18	1	대지	411	42	서울특별시	1.1E+1	
337	2.9E+07	스타일류보전다리	D	소매	D05	의복의류	D05A01	일반의류	G47416	셔츠 및 기		11	서울특별시	11440	마포구	1.1E+09	서교동	1.1E+09	서교동	1.1E+18	1	대지	370	12	서울특별시	1.1E+1	
362	2.1E+07	별매식당	Q	음식	Q01	한식	Q01A01	한식/백반/56111	한식 음식?	11	서울특별시	11440	마포구	1.1E+09	대흥동	1.1E+09	노고산동	1.1E+18	1	대지	31	23	서울특별시	1.1E+1			
370	2.2E+07	레포산	D	소매	D14	운동/경기	D14A01	운동/경기	G47631	운동 및 경		11	서울특별시	11440	마포구	1.1E+09	서강동	1.1E+09	창전동	1.1E+18	1	대지	402	17	서울특별시	1.1E+1	
386	1.5E+07	1동공인중개사사무소	L	부동산	L01	부동산중개	L01A01	부동산중개	L68221	부동산 자		11	서울특별시	11440	마포구	1.1E+09	연남동	1.1E+09	연남동	1.1E+18	1	대지	487	422	서울특별시	1.1E+1	
408	2.4E+07	진천코퍼레이션	D	소매	D23	자동차/자	D23A06	자동차부품	G45219	기타 자동차		11	서울특별시	11440	마포구	1.1E+09	서강동	1.1E+09	창전동	1.1E+18	1	대지	13	3	서울특별시	1.1E+1	
410	2.9E+07	빈트리망충당원점	Q	음식	Q12	커피점/커피	Q12A01	커피전문점	S6220	비알콜 음		11	서울특별시	11440	마포구	1.1E+09	망원동	1.1E+09	망원동	1.1E+18	1	대지	418	22	서울특별시	1.1E+1	
440	2.2E+07	스코피	홍대입구점	F	생활서비스	F06	사진	F06A01	사진관	M73301	인물 사진		11	서울특별시	11440	마포구	1.1E+09	서교동	1.1E+09	동교동	1.1E+18	1	대지	168	8	서울특별시	1.1E+1
444	2.4E+07	한강종합보수	D	소매	D21	절물/산방	D21A02	보일러/냉	G47511	절물 및 난		11	서울특별시	11440	마포구	1.1E+09	도화동	1.1E+09	도화동	1.1E+18	1	대지	357		서울특별시	1.1E+1	
498	2.4E+07	CU	상암프라자	D	소매	D03	종합소매점	D03A01	편의점	G47122	제인화 편의		11	서울특별시	11440	마포구	1.1E+09	상암동	1.1E+09	상암동	1.1E+18	1	대지	1687		서울특별시	1.1E+1
518	2.2E+07	현대공인중개사사무소	L	부동산	L01	부동산중개	L01A01	부동산중개	L68221	부동산 자		11	서울특별시	11440	마포구	1.1E+09	용강동	1.1E+09	마포동	1.1E+18	1	대지	35	1	서울특별시	1.1E+1	
535	2.3E+07	수경사	F	생활서비스	F02	세탁/가사	F02A01	세탁소/빨	S96912	가정용 세		11	서울특별시	11440	마포구	1.1E+09	성산동	1.1E+09	성산동	1.1E+18	1	대지	252	30	서울특별시	1.1E+1	
582	2.4E+07	더바운스	D	소매	D05	의복의류	D05A01	일반의류	G47416	셔츠 및 기		11	서울특별시	11440	마포구	1.1E+09	서강동	1.1E+09	상수동	1.1E+18	1	대지	331	8	서울특별시	1.1E+1	
599	2.2E+07	폴리카를리카	D	소매	D16	화장품소매	D16A01	화장품판매	G47813	화장품 및		11	서울특별시	11440	마포구	1.1E+09	성산동	1.1E+09	성산동	1.1E+18	1	대지	515		서울특별시	1.1E+1	
616	2.5E+07	백산부동산	L	부동산	L01	부동산중개	L01A01	부동산중개	L68221	부동산 자		11	서울특별시	11440	마포구	1.1E+09	서교동	1.1E+09	노고산동	1.1E+18	1	대지	56	7	서울특별시	1.1E+1	
638	1.6E+07	남촌성고기구이	Q	음식	Q01	한식	Q01A02	갈비/삼갈	S6111	한식 음식?		11	서울특별시	11440	마포구	1.1E+09	대흥동	1.1E+09	대흥동	1.1E+18	1	대지	39		서울특별시	1.1E+1	
691	2.4E+07	맨즈	D	소매	D05	의복의류	D05A01	일반의류	G47416	셔츠 및 기		11	서울특별시	11440	마포구	1.1E+09	서교동	1.1E+09	서교동	1.1E+18	1	대지	335	20	서울특별시	1.1E+1	
707	1.6E+07	나라인테리어	D	소매	D07	가정/주방	D07A01	인테리어	G47599	그외 기타		11	서울특별시	11440	마포구	1.1E+09	합정동	1.1E+09	합정동	1.1E+18	1	대지	361	17	서울특별시	1.1E+1	
715	2.4E+07	GS리테일GS25월드점	D	소매	D03	종합소매점	D03A01	편의점	G47122	제인화 편의		11	서울특별시	11440	마포구	1.1E+09	성산동	1.1E+09	성산동	1.1E+18	1	대지	515		서울특별시	1.1E+1	
734	2.3E+07	새나라이음원	F	생활서비스	F01	이/미용/건	F01A04	남성미용	S96111	이음원		11	서울특별시	11440	마포구	1.1E+09	망원동	1.1E+09	망원동	1.1E+18	1	대지	394	94	서울특별시	1.1E+1	
774	1.5E+07	경남주자	F	생활서비스	F14	자동차/이	F14A07	주차장운영	H52915	주차장 온		11	서울특별시	11440	마포구	1.1E+09	서교동	1.1E+09	서교동	1.1E+18	1	대지	353	2	서울특별시	1.1E+1	
803	2.5E+07	이치노사자문	F	생활서비스	F13	가사서비스	F13A01	사업경영	M71531	경영컨설팅		11	서울특별시	11440	마포구	1.1E+09	도화동	1.1E+09	마포동	1.1E+18	1	대지	136	1	서울특별시	1.1E+1	
834	2.3E+07	루소소	D	소매	D05	의복의류	D05A06	맞춤양장	G47413	남녀용 장		11	서울특별시	11440	마포구	1.1E+09	아현동	1.1E+09	아현동	1.1E+18	1	대지	347	31	서울특별시	1.1E+1	
885	1.5E+07	제일재면소상암점	R	음식	Q04	분식	Q04A01	라면/분식	S6194	분식 및 김		11	서울특별시	11440	마포구	1.1E+09	성산동	1.1E+09	성산동	1.1E+18	1	대지	515		서울특별시	1.1E+1	
896	2E+07	콤비수제버거	Q	음식	Q07	패스트푸드	Q07A04	패스트푸드	S6199	그외 기타		11	서울특별시	11440	마포구	1.1E+09	영리동	1.1E+09	영리동	1.1E+18	1	대지	9	56	서울특별시	1.1E+1	
929	2.3E+07	우리사	F	생활서비스	F02	세탁/가사	F02A01	세탁소/빨	S96912	가정용 세		11	서울특별시	11440	마포구	1.1E+09	상암동	1.1E+09	상암동	1.1E+18	1	대지	33	4	서울특별시	1.1E+1	
945	2E+07	커피럽	Q	음식	Q12	커피점/커피	Q12A01	커피전문점	S6220	비알콜 음		11	서울특별시	11440	마포구	1.1E+09	서교동	1.1E+09	서교동	1.1E+18	1	대지	327	19	서울특별시	1.1E+1	



- 소상공인 진흥공단의 상가(상권) 정보 데이터를 활용하여 마포구내의 상가들의 위치를 파악

- K-means 클러스터링 사용

### 6. 카드 매출 데이터를 군집화 된 상권에 할당 \_개별상가에 대한 매출데이터를 업종별 상가별로 소속

```
# 신한카드 소비 블록 각각의 데이터가
# 서울시 마포구 상권 k-means의 군집중 어느 군집(상권)에 해당될지
# 예측 및 할당하는 작업
def gubyeol_kmeans_mapo(sell_df, size):
    result_df = pd.DataFrame()
    mapogu_sell = pd.DataFrame()
    for i in tqdm(total_stores_new['시군구명'].unique()):
        if i == '마포구':
            gu_store = total_stores_new[total_stores_new['시군구명']==i].reset_index()
            gu_store = gu_store.loc[:, ['상호명', '상권업종중분류코드', '상권업종중분류명', '상권업종소분류명', '시군구명', '건물명', '위도', '경도']]
            gu_store['건물명'] = le.fit_transform(gu_store['건물명']) #라벨인코더 적용

            # k-means clustering 실행
            points = gu_store.loc[:, ['위도', '경도']]
            center_len = int(np.ceil(len(points)/size)) # 대략 사용자가 지정한 size당 하나꼴의 군집(상권)이 잡히도록 center_len을 설정.
            kmeans = KMeans(init='k-means++', random_state=11, n_clusters=center_len)
            kmeans.fit(points)

            wido_gyundo = sell_df[['위도', '경도']]
            mapogu_sell = kmeans.predict(wido_gyundo)
            # 결과 확인
            gu_cluster_market = gu_store.copy()
            gu_cluster_market['cluster'] = kmeans.labels_
            result_df = pd.concat([result_df, gu_cluster_market], axis=0)
            print(gu_cluster_market.head(3))
    return result_df, mapogu_sell

mapo_df = pd.DataFrame()
mapogu_cluster = pd.DataFrame()
mapo_df, mapogu_cluster = gubyeol_kmeans_mapo(sell_block_mapo_new, 50)
print(gubyeol_df.iloc[[0, 100, 5000, 20000, -1], :])
```

- 신한카드 매출데이터의 블록 정보는 상권에 대한 정보를 반영하지 않음
- 두 데이터를 통합하기 위해 상권에 대한 정보가 포함된 소상공인 진흥공단 데이터에 맞추는 과정
- 신한카드 매출데이터를 행정동별로 분리, 상권정보를 행정동별로 분리
- 신한카드 매출데이터를 군집화 된 상권에 할당함으로써 두 데이터를 통합

### 7. 분기별 매출 변화율 데이터 분석을 위한 클래스 정의

```
class 분기변화율_스케일조정:
    def __init__(self, 분기, data):
        from sklearn.preprocessing import MinMaxScaler
        self.scaler = MinMaxScaler()
        self.df = data
        self.분기 = 분기
        self.temp = self.df['{}분기_변화'.format(self.분기)]
        if __name__ == "__main__":
            print("자동으로 해당분기 업종별 변화율의 MinMaxScaler를 적용해줍니다")
            print(__name__)
        else:
            print("임포트되어 사용됨")
            print("자동으로 해당분기 업종별 변화율의 MinMaxScaler를 적용해줍니다")
            print(__name__)
    def rescale(self):
        self.temp = self.temp.replace(np.inf, np.nan)
        self.temp.dropna(axis=0, inplace=True)
        minmax = pd.DataFrame()
        # 업종별 변화추세에 따른 minmax scaler 적용
        for i in self.df['업종'].unique():
            my_df = self.df[self.df['업종'] == i]
            my_df['{}분기_변화'.format(self.분기)] = self.scaler.fit_transform(my_df['{}분기_변화'.format(self.분기)].values.reshape(-1, 1))
            minmax = pd.concat([minmax, my_df])
        return minmax
    def info(self):
        print("데이터 업종 개수", len(self.df['업종'].unique()))
        print("데이터 상권 군집수", len(self.df['상권군집번호'].unique())) # 1분기_변화%가 무한으로 나오는걸 지워서 군집도 다소 낮아감
        print("데이터 전체수", len(self.df))
```

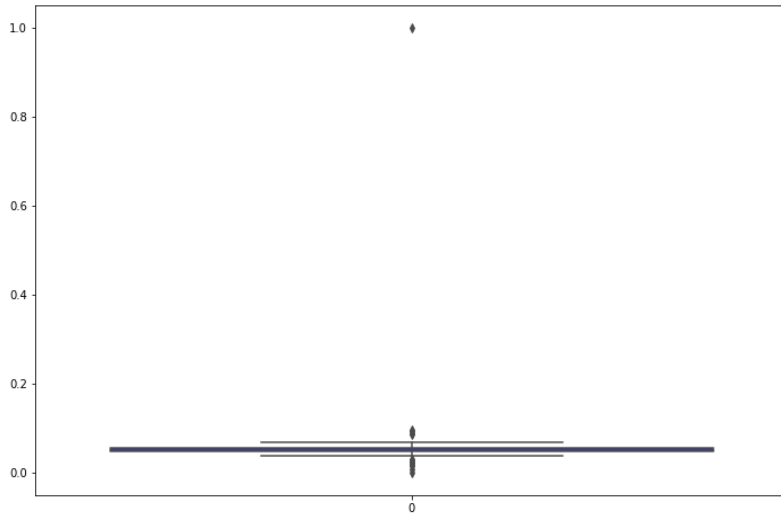
```
# 자식 클래스
class 분기변화율_재조정(분기변화율_스케일조정):
    def __init__(self, 분기, data):
        super(분기변화율_재조정, self).__init__(분기, data) # 부모 클래스 생성자 호출
        print('분기변화율_재조정 생성자 호출')
    def outliers_iqr(self, data):
        q1, q3 = np.percentile(data, [25, 75])
        iqr = q3 - q1
        lower_bound = q1 - (iqr * 7.0) # 보통 정하는 1.5 기준보다 더 크게 벗어난 이상치들을 제거(너무 데이터가 적어질것을 우려)
        upper_bound = q3 + (iqr * 7.0)
        return np.where((data > upper_bound) | (data < lower_bound))
    def del_outlier(self):
        temp = self.rescale().iloc[:, [1, 2, 3]]
        result = pd.DataFrame()
        outlier_df = pd.DataFrame()
        for i in temp.업종.unique():
            if temp.groupby('업종').count()['4분기_변화%'].loc[i] == 1:
                idx = temp[temp['업종'] == i].index
                outlier_df = pd.concat([outlier_df, temp.loc[idx]], axis=0)
                result = temp.drop(idx)
            # 아래 문장은 이상하게 오류가 나므로 임시 패스
            # try: # 위의 과정으로 인해 이미 없어진 업종이 아니라면 아래작업 수행(수치적 이상치)
            #     out_idx = self.outliers_iqr(temp[temp['업종'] == i]['4분기_변화%'])[0]
            #     outlier_df = pd.concat([outlier_df, temp.loc[out_idx]], axis=0)
            #     result = temp.drop(out_idx)
            # except: # 이미 없어진 업종이라면 pass
            #     pass
        return result, outlier_df
```

업종별 매출 변화율 데이터에 MinMaxScaler를 적용

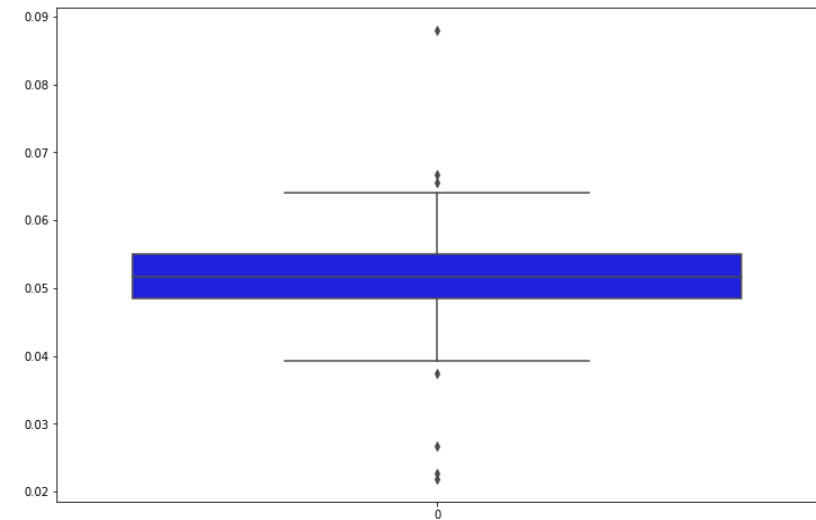
이상치 제거를 위한 자식 class 생성

### 8. 이상치 제거

```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(12,8))
sns.boxplot(data = ang[ang['업종']=='SB016']['4분기_변화%'],color='blue')
plt.show()
```



```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(12,8))
sns.boxplot(data = ang[ang['업종']=='SB016'].iloc[:157]['4분기_변화%'],color='blue')
plt.show()
```



코드 실행을 통해, 이상치 제거 후 데이터 시각화



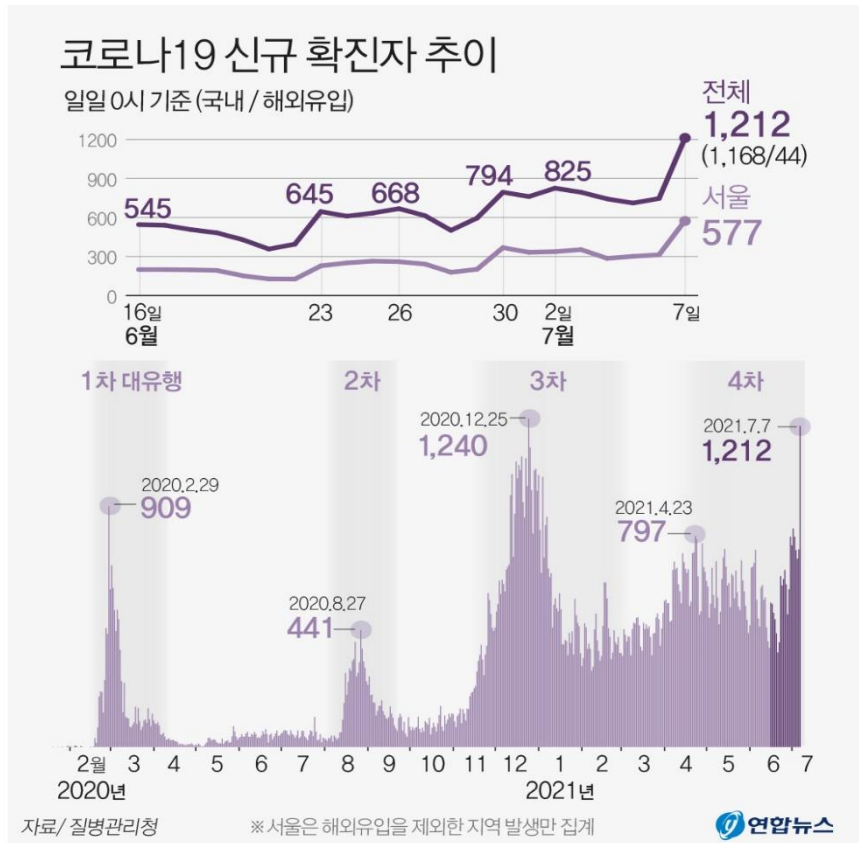
## 02. 데이터 분석

개괄설명

기존 시너지 업종에 대한 코로나 전후 시너지 검증\_ 시점 선정

- 코로나 전후가 뚜렷하게 드러날 수 있는 대유행 시점을 기준으로 분석
  - 계절적인 요인을 제거하기 위해 분기별 비교 실행
  - 19년도 1,3월 매출 변화율 VS 20년도 1,3월 매출 변화율
  - 19년도 7,9월 매출 변화율 VS 20년도 7,9월 매출 변화율을
  - 19년도 10,12월 매출 변화율 VS 20년도 10,12월 매출변화율
- | 20년도 변화율 - 19년도 변화율 | 이 클수록 COVID-19의 영향을

크게 받았다고 판단



## 02. 데이터 분석

## 기존 시너지 업종 조합

마포구 상권 내 빈도가 높은 업종을 기준으로  
선행연구에서 시너지가 검증된 업종 조합 선정

### 마포구 상권 내 빈도가 높은 업종

한식	학원
기타 요식업	노래방
커피전문점	유흥주점
미용실	기타 유흥업소
미용서비스	화장품
편의점	안경

### 시너지가 검증된 업종 조합

한식 - 커피 전문점  
한식 - 기타 요식업  
미용서비스 - 미용실  
화장품 - 미용실  
미용실 - 안경점  
커피 전문점 - 미용실  
화장품-안경

## 02. 데이터 분석

## 업종 조합 CASE

코로나 전과 후를 비교하여 코로나에 상관없이  
시너지 효과를 보이는 업종끼리 조합

### CASE 1

#### 연계소비

업종 간의 연계소비가  
발생하는 조합

### CASE 2

#### 밀집도

유사업종 간의 밀집도를  
기반으로 한 조합

### CASE 3

#### 입지특성

입지특성이 비슷한  
업종끼리 조합

## 02. 데이터 분석

데이터 분석\_매출 데이터 분석

### 업종별 평균 매출 변화율 시기별 비교

```
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
from mlxtend.preprocessing import TransactionEncoder

중간결과_19년3분기변화 = b5.replace(np.inf, np.nan).dropna(axis=0)
중간결과_19년3분기변화 = 분기변화율_스케일조정(3, 중간결과_19년3분기변화).rescale()

분기변화율_19년3분기, outliers = 분기변화율_재조정(3, 중간결과_19년3분기변화).del_outlier()
tot_list = []
for i in 분기변화율_19년3분기.groupby('상권군집번호')['업종'].unique():
    tot_list.append(list(i))
print(len(tot_list))
te = TransactionEncoder()
te_result = te.fit(tot_list).transform(tot_list)

# 상권군집번호 라는 중요한 index정보를 잃어버리지 않도록
hallo = pd.DataFrame(te_result, columns = te.columns_)
hallo.index = 분기변화율_19년3분기.groupby('상권군집번호')['업종'].unique().index
```

먼저 시기를 나눈 뒤 분석단위(상권군집)별로,  
업종별 평균 매출액 변화율을 앞선 클래스를 활용하여 계산

### 매출 변화율 상위 30% 셀 추출

```
for iter,i in enumerate(분기변화율_19년3분기.groupby('상권군집번호')['업종'].unique()):
    idx = 분기변화율_19년3분기.groupby('상권군집번호')['업종'].unique().index[iter]
    for j in i:
        temp = hallo[hallo.index==idx].T
        temp = list(temp[temp[idx]==True].index)
        globals()['군집{0}_{1}의근처업종'.format(idx,j)] = temp # 본인 업종 포함하고, 근처업종

top_19년3분기_업종별_상위30 = 분기변화율_19년3분기.groupby('업종').apply(top,ratio=0.3, column='3분기_변화%',opt=False)
top_19년3분기_업종별_하위30 = 분기변화율_19년3분기.groupby('업종').apply(top,ratio=0.3, column='3분기_변화%',opt=True)

for 업종 in top_19년3분기_업종별_상위30.업종.unique():
    globals()['{}_list'.format(업종)] = []
    globals()['te_{}'.format(업종)] = pd.DataFrame()
    for 상권 in top_19년3분기_업종별_상위30.loc[업종].상권군집번호.unique():
        try:
            globals()['{}_list'.format(업종)].append(globals()['군집{0}_{1}의근처업종'.format(상권,업종)])
            globals()['te_{}'.format(업종)] = te.fit(globals()['{}_list'.format(업종)]).transform(globals()['{}_list'.format(업종)])
            globals()['te_{}'.format(업종)] = pd.DataFrame(globals()['te_{}'.format(업종)], columns=te.columns_)
        except:
            pass
```

자동으로 해당분기 업종별 변화%의 MinMaxScaler를 적용해줍니다

\_\_main\_\_

자동으로 해당분기 업종별 변화%의 MinMaxScaler를 적용해줍니다

\_\_main\_\_

분기변화율\_재조정 생성자 호출

298

매출 변화율의 MinMax 결과값이

높은 상위 30%의 (업종,클러스터) 셀 추출

## 02. 데이터 분석

데이터 분석\_매출 데이터 분석

### 매출 변화율 상위 30% 셀 추출

```
비교업종 = ['SB001', 'SB006', 'SB008', 'SB039', 'SB016', 'SB047']

te = TransactionEncoder()
myzip_w193 = zipfile.ZipFile('sample193_1.zip', 'w')
for i in 비교업종:
    te_df = pd.DataFrame(globals()['te_{}'.format(i)])
    itemset = apriori(te_df, min_support=0.10, use_colnames=True)
    상위_15업종조합 = itemset.sort_values(by='support', ascending=False).iloc[0:15]
    globals()['업종조합_19년3분기_{}매출_상위15개'.format(i)] = 상위_15업종조합
    상위_15업종조합.to_excel('업종조합_19년3분기_{}매출_상위15개.xlsx'.format(i))
    myzip_w193.write('업종조합_19년3분기_{}매출_상위15개.xlsx'.format(i))
myzip_w193.close()
zipfile.is_zipfile('sample193_1.zip')
```

True

```
비교업종2 = ['SB009', 'SB011', 'SB010', 'SB040', 'SB041', 'SB024']

te = TransactionEncoder()
myzip_w193 = zipfile.ZipFile('sample193_2.zip', 'w')
for i in 비교업종2:
    te_df = pd.DataFrame(globals()['te_{}'.format(i)])
    itemset = apriori(te_df, min_support=0.10, use_colnames=True)
    상위_15업종조합 = itemset.sort_values(by='support', ascending=False).iloc[0:15]
    globals()['업종조합_19년3분기_{}매출_상위15개'.format(i)] = 상위_15업종조합
    상위_15업종조합.to_excel('업종조합_19년3분기_{}매출_상위15개.xlsx'.format(i))
    myzip_w193.write('업종조합_19년3분기_{}매출_상위15개.xlsx'.format(i))
myzip_w193.close()
zipfile.is_zipfile('sample193_2.zip')
```

True



















정규화된 매출 변화율이 높은 상위 30%의 업종 조합 생성 및

연관규칙 지지도 기반 상위 15개 시기별 파일 추출

## 02. 데이터 분석

데이터 분석\_매출 기반 시너지 추출

### 매출 변화율을 기반으로 연관성 분석

 업종조합_SB001매출_상위15개	2021-12-12 오후 3:07	Microsoft Excel ...	6KB
 업종조합_SB002매출_상위15개	2021-12-12 오후 3:07	Microsoft Excel ...	6KB
 업종조합_SB003매출_상위15개	2021-12-12 오후 3:07	Microsoft Excel ...	6KB
 업종조합_SB004매출_상위15개	2021-12-12 오후 3:07	Microsoft Excel ...	6KB
 업종조합_SB005매출_상위15개	2021-12-12 오후 3:07	Microsoft Excel ...	6KB
 업종조합_SB006매출_상위15개	2021-12-12 오후 3:07	Microsoft Excel ...	6KB
 업종조합_SB007매출_상위15개	2021-12-12 오후 3:07	Microsoft Excel ...	6KB
 업종조합_SB008매출_상위15개	2021-12-12 오후 3:07	Microsoft Excel ...	6KB
 업종조합_SB009매출_상위15개	2021-12-12 오후 3:08	Microsoft Excel ...	6KB
 업종조합_SB010매출_상위15개	2021-12-12 오후 3:08	Microsoft Excel ...	6KB
 업종조합_SB011매출_상위15개	2021-12-12 오후 3:11	Microsoft Excel ...	6KB
 업종조합_SB013매출_상위15개	2021-12-12 오후 3:17	Microsoft Excel ...	6KB
 업종조합_SB014매출_상위15개	2021-12-12 오후 3:17	Microsoft Excel ...	6KB
 업종조합_SB015매출_상위15개	2021-12-12 오후 3:17	Microsoft Excel ...	6KB
 업종조합_SB016매출_상위15개	2021-12-12 오후 3:17	Microsoft Excel ...	6KB
 업종조합_SB017매출_상위15개	2021-12-12 오후 3:17	Microsoft Excel ...	6KB
 업종조합_SB018매출_상위15개	2021-12-12 오후 3:17	Microsoft Excel ...	6KB
 업종조합_SB019매출_상위15개	2021-12-12 오후 3:17	Microsoft Excel ...	6KB

추출한 셀을 클러스터 차원에서 Apriori를 통하여

규칙성을 보이는 조합을 추출

Apriori로 나온 조합을 매출 기반 시너지를 가지고 있는 조합이라고 가정

지지도(Support)를 기반으로 규칙성을 검증

# 02. 데이터 분석

데이터 분석\_연계소비 기반 시너지 추출

매출 기준 시너지 조합을 연계소비 기준으로 재검증

```
시간대별_이용순위['군집'] = 시간대별_이용순위[0].apply(lambda x: x.split(',')[0])
시간대별_이용순위['시간대'] = 시간대별_이용순위[0].apply(lambda x: x.split(',')[1])
시간대별_이용순위['업종'] = 시간대별_이용순위[0].apply(lambda x: x.split(',')[2])
```

```
시간대별_이용순위['군집'] = 시간대별_이용순위.군집.apply(lambda x: x.replace('(', ''))
시간대별_이용순위['업종'] = 시간대별_이용순위.업종.apply(lambda x: x.replace(' ', '').replace('"', ''))
```

```
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib as mpl # 기본 설정 만지는 용도
import matplotlib.font_manager as fm # 폰트 관련 용도
%matplotlib inline
```

```
fm._rebuild()
mpl.rcParams['axes.unicode_minus'] = False
plt.rc('font', family='NanumBarunGothic')
```

```
한식_소비주요시간 = 시간대별_이용순위[시간대별_이용순위.업종=='SB001']['시간대'].iloc[0:int(len(시간대별_이용순위[시간대별_이용순위.업종=='SB001'])/2)]
plt.bar(한식_소비주요시간.value_counts().sort_index().index, 한식_소비주요시간.value_counts().sort_index().values)
plt.title('SB001(한식) 하이라이트 소비시간 분포', fontsize=20)
plt.xlabel('Time', fontsize=18)
plt.ylabel('Frequency', fontsize=18)
plt.figure(figsize=(18,12))
plt.savefig('SB001(한식)하이라이트소비.png')
plt.show()
```

매출 기준 시너지 조합들의 최고 매출액 시간대가 비슷하다면

연계소비가 발생했다고 판단

매출 기준 시너지와 연계소비 시너지 둘 다 고려하여 최종적인

시너지 조합을 추출



## 02. 데이터 분석

## 데이터 분석 결과\_단순분석 결과 제시

### 기존 시너지 검증

#### < 한식 - 커피 전문점 >

- 한식 입장:

커피전문점과 시너지 효과가

크지 않음

- 커피전문점 입장:

한식과 시너지 효과가 큼

	A	B	C	D	E	F
1		support	itemsets			
2	0	1 frozenset({'SB001'})				
3	47	0.880952 frozenset({'SB001', 'SB039'})				
4	20	0.880952 frozenset({'SB039'})				
5	7	0.785714 frozenset({'SB008'})				
6	34	0.785714 frozenset({'SB008', 'SB001'})				
7	268	0.702381 frozenset({'SB008', 'SB001', 'SB039'})				
8	129	0.702381 frozenset({'SB008', 'SB039'})				
9	37	0.630952 frozenset({'SB016', 'SB001'})				
10	27	0.630952 frozenset({'SB047'})				
11	54	0.630952 frozenset({'SB047', 'SB001'})				
12	10	0.630952 frozenset({'SB016'})				
13	185	0.547619 frozenset({'SB047', 'SB039'})				
14	324	0.547619 frozenset({'SB047', 'SB001', 'SB039'})				
15	284	0.535714 frozenset({'SB016', 'SB001', 'SB039'})				
16	145	0.535714 frozenset({'SB016', 'SB039'})				
17						

한식 (SB001) 입장 시너지 업종

	A	B	C	D	E	F
1		support	itemsets			
2	5	1 frozenset({'SB006'})				
3	0	0.935484 frozenset({'SB001'})				
4	32	0.935484 frozenset({'SB001', 'SB006'})				
5	7	0.870968 frozenset({'SB008'})				
6	113	0.870968 frozenset({'SB008', 'SB006'})				
7	307	0.83871 frozenset({'SB008', 'SB001', 'SB006'})				
8	34	0.83871 frozenset({'SB008', 'SB001'})				
9	19	0.806452 frozenset({'SB039'})				
10	125	0.806452 frozenset({'SB039', 'SB006'})				
11	46	0.774194 frozenset({'SB001', 'SB039'})				
12	319	0.774194 frozenset({'SB001', 'SB006', 'SB039'})				
13	26	0.741935 frozenset({'SB047'})				
14	132	0.741935 frozenset({'SB047', 'SB006'})				
15	157	0.709677 frozenset({'SB008', 'SB047'})				
16	326	0.709677 frozenset({'SB047', 'SB001', 'SB006'})				
17						

커피전문점(SB006) 입장 시너지 업종

# 02. 데이터 분석

## 기존 시너지 검증

### < 미용실 - 커피 전문점 >

- 미용실 입장 :

커피전문점과 시너지 효과가  
크지 않음

- 커피전문점 입장 :

미용실과 시너지 효과가  
크지 않음

	A	B	C	D	E	F
1		support	itemsets			
2	0	1	frozenset({'SB001'})			
3	18	1	frozenset({'SB039'})			
4	44	1	frozenset({'SB001', 'SB039'})			
5	33	0.830986	frozenset({'SB008', 'SB001'})			
6	7	0.830986	frozenset({'SB008'})			
7	141	0.830986	frozenset({'SB008', 'SB039'})			
8	297	0.830986	frozenset({'SB008', 'SB001', 'SB039'})			
9	355	0.71831	frozenset({'SB047', 'SB001', 'SB039'})			
10	51	0.71831	frozenset({'SB047', 'SB001'})			
11	199	0.71831	frozenset({'SB047', 'SB039'})			
12	25	0.71831	frozenset({'SB047'})			
13	9	0.605634	frozenset({'SB016'})			
14	35	0.605634	frozenset({'SB016', 'SB001'})			
15	158	0.605634	frozenset({'SB016', 'SB039'})			
16	314	0.605634	frozenset({'SB016', 'SB001', 'SB039'})			
17						

미용실 (SB039) 입장 시너지 업종

	A	B	C	D	E	F
1		support	itemsets			
2	5	1	frozenset({'SB006'})			
3	0	0.935484	frozenset({'SB001'})			
4	32	0.935484	frozenset({'SB001', 'SB006'})			
5	7	0.870968	frozenset({'SB008'})			
6	113	0.870968	frozenset({'SB008', 'SB006'})			
7	307	0.83871	frozenset({'SB008', 'SB001', 'SB006'})			
8	34	0.83871	frozenset({'SB008', 'SB001'})			
9	19	0.806452	frozenset({'SB039'})			
10	125	0.806452	frozenset({'SB039', 'SB006'})			
11	46	0.774194	frozenset({'SB001', 'SB039'})			
12	319	0.774194	frozenset({'SB001', 'SB006', 'SB039'})			
13	26	0.741935	frozenset({'SB047'})			
14	132	0.741935	frozenset({'SB047', 'SB006'})			
15	157	0.709677	frozenset({'SB008', 'SB047'})			
16	326	0.709677	frozenset({'SB047', 'SB001', 'SB006'})			
17						

커피전문점(SB006) 입장 시너지 업종

## 02. 데이터 분석

데이터 분석 결과\_단순분석 결과 제시

### 기존 시너지 검증

#### 〈 기타 요식업 - 한식 〉

· 기타 요식업 입장:

한식과 시너지 효과가 큼

- 한식 입장:

기타 요식업과 시너지 효과가 큼

#### 〈 화장품 - 미용실 〉

- 화장품 입장:

미용실과 시너지 효과가 큼

- 미용실 입장:

화장품과 시너지 효과가 크지 않음

## 02. 데이터 분석

데이터 분석 결과\_단순분석 결과 제시

### 기존 시너지 검증

#### 〈 미용실 - 안경점 〉

- 미용실 입장:  
안경점과 시너지 효과가  
크지 않음
- 안경점 입장:  
미용실과 시너지 효과가 큼

#### 〈 미용 서비스 - 미용실 〉

- 미용 서비스 입장:  
미용실과 시너지 효과가 큼
- 미용실 입장:  
미용 서비스와 시너지 효과가  
크지 않음

#### 〈 화장품 - 안경점 〉

- 화장품 입장:  
미용실과 시너지 효과가 큼
- 미용실 입장:  
화장품과 시너지 효과가 크지 않음

## 02. 데이터 분석

데이터 분석 결과\_단순분석 결과 제시

### 새롭게 나타나거나 사라진 시너지

#### 〈 커피 전문점 - 편의점 〉

- 기타요식업 입장 :

한식과 시너지 효과가 큼

- 한식 입장 :

기타 요식업과 시너지 효과가 큼

#### 〈 학원 - 편의점 〉

- 화장품 입장 :

미용실과 시너지 효과가 큼

- 미용실 입장 :

화장품과 시너지 효과가 크지 않음

# 03. 결론

결론 도출\_데이터 분석 결과 해석

## CASE 1: 기존 연구와 다르게 시너지가 나지 않은 조합

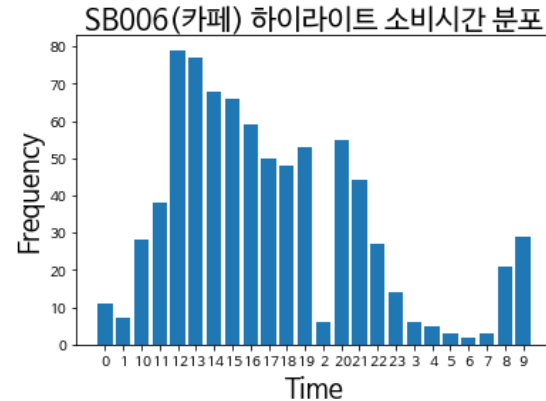
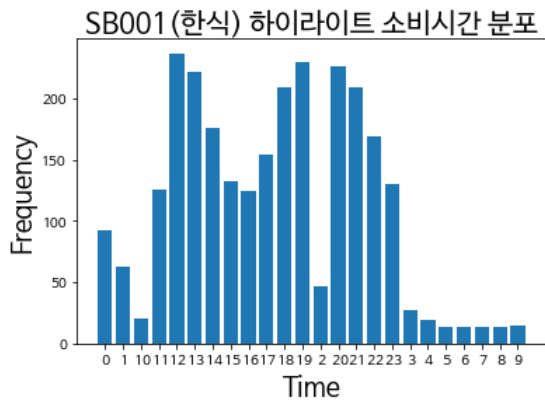
〈미용실 - 커피 전문점〉

〈화장품 - 안경〉

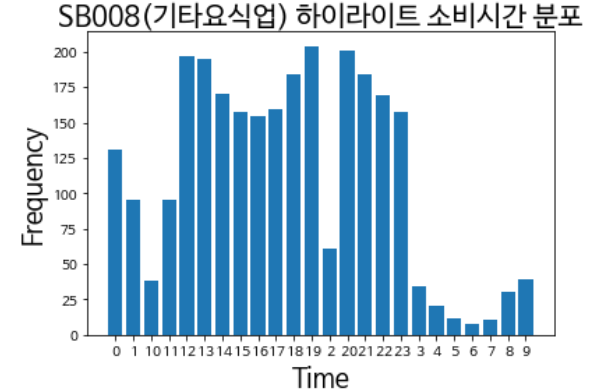
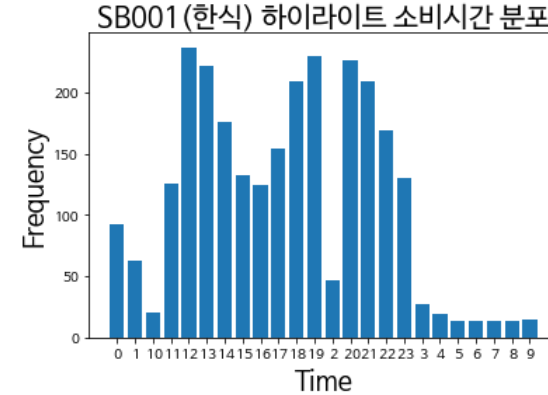
- 입지 특성 상 시너지가 있을 것이라 고려 되었지만,  
실제 시너지 효과가 크지 않았음

### CASE 2 : 위기상황에서도 강건한 업종조합

#### 〈한식 - 커피 전문점〉



#### 〈한식 - 기타 요식업〉



- 매출 기반 분석 + 매출 시간대 분석
- 하이라이트 소비시간이 식사시간에 겹치므로 연계소비가 발생한다고 판단
- 두 자료 모두 유의미하게 시너지가 있음을 나타냄

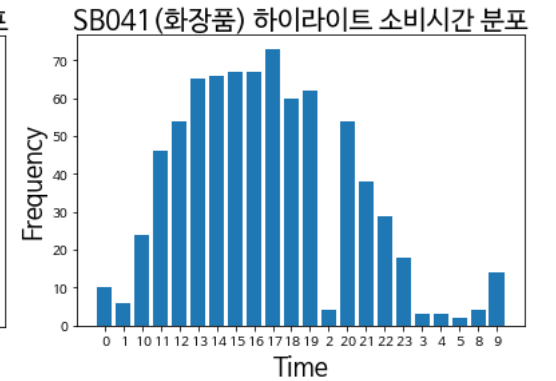
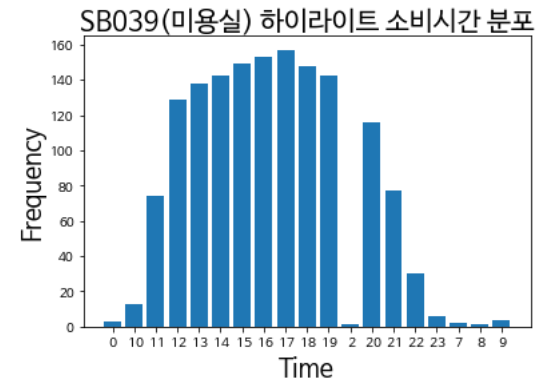
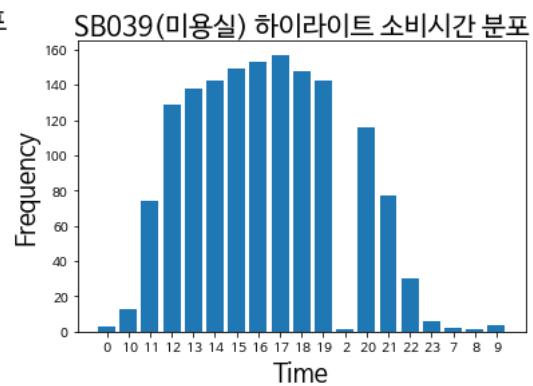
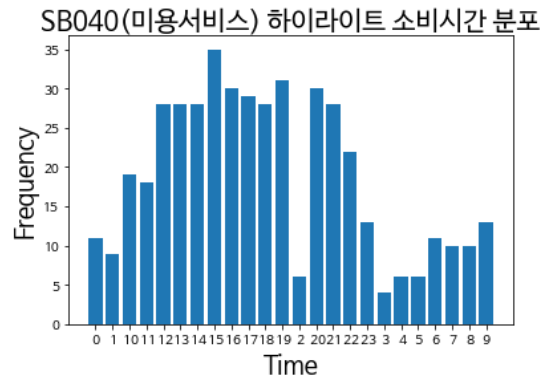
# 03. 결론

## 결론 도출\_데이터 분석 결과 해석

### CASE 2 : 위기상황에서도 강건한 업종조합

#### 〈 미용서비스 - 미용실 〉

#### 〈 미용실 - 화장품 〉



- 매출 기반 분석 + 매출 시간대 분석

- 하이라이트 소비시간이 낮시간대에 균일하게 겹치므로 연계소비가 발생한다고 판단

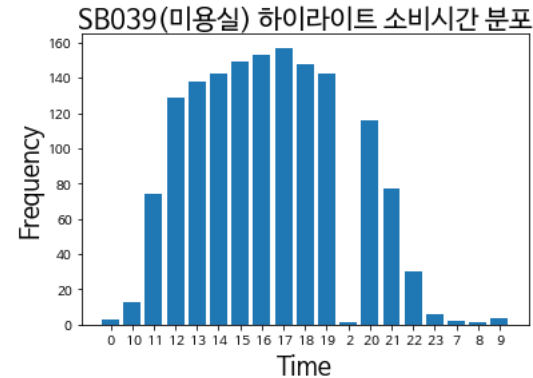
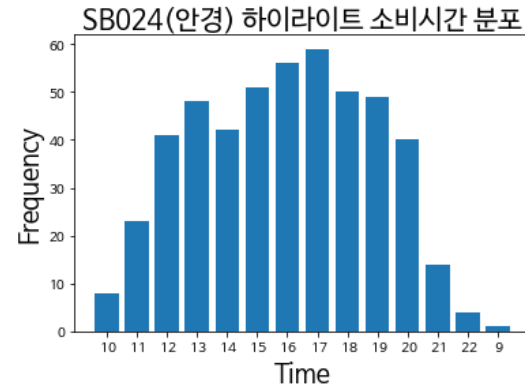
- 두 자료 모두 유의미하게 시너지가 있음을 나타냄



# 03. 결론

## CASE 2 : 위기상황에서도 강건한 업종조합

### 〈 안경점 - 미용실 〉



- 매출 기반 분석 + 매출 시간대 분석
- 하이라이트 소비시간이 낮시간대에 균일하게 겹치므로 연계소비가 발생한다고 판단
  - 두 자료 모두 유의미하게 시너지가 있음을 나타냄

### CASE 3 : 위기상황에 영향을 받은 업종 조합

#### 〈 편의점 - 커피 전문점 〉

##### 커피전문점 입장 시너지 업종 리스트

- |                                |   |  |
|--------------------------------|---|--|
| - 19년도 1분기: 한식 기타요식, 미용서비스, 학원 |   | - 20년도 1분기: 한식, <b>편의점</b> , 미용서비스, 기타요식, 학원 |
| - 19년도 3분기: 한식 기타요식, 미용서비스, 학원 | → | - 20년도 3분기: 한식 미용서비스, 기타요식, 학원               |
| - 19년도 4분기: 한식 미용서비스, 기타요식, 일식 |   | - 20년도 4분기: 한식 기타요식, <b>편의점</b> , 미용서비스      |

- 19년도 (코로나 이전)에는 편의점-커피전문점의 시너지 효과가 크지 않았음.
- 20년도(코로나 이후)에 편의점-커피전문점의 시너지 효과가 크게 나타남

### CASE 3 : 위기상황에 영향을 받은 업종 조합

#### 〈 편의점 - 학원 〉

##### 학원 입장 시너지 업종 리스트

- |                                       |   |                           |
|---------------------------------------|---|---------------------------|
| - 19년도 1분기: 한식, 기타요식, 미용실             |   | - 20년도 1분기: 한식, 기타요식, 미용실 |
| - 19년도 3분기: 한식, 기타요식, 미용실, <u>편의점</u> | → | - 20년도 3분기: 한식, 기타요식, 미용실 |
| - 19년도 4분기: 한식, 기타요식, 미용실, <u>편의점</u> |   | - 20년도 4분기: 한식, 기타요식, 미용실 |
- 
- 19년도 (코로나 이전)에는 편의점-학원의 시너지 효과가 크게 나타남.
  - 20년도 (코로나 이후)에 편의점-학원의 시너지 효과가 감소함

### 시사점 제시 및 발전방향

현재 프로젝트 결과 CASE들에 해당하는 업종 조합을 참고하여 추후 소상공인들이  
점포 입지 선정을 진행 할 때 도움을 줄 수 있을 것이다.

이번 프로젝트는 마포구와 소수 업종들 만을 분석하였지만, 분석의 범위를 넓혀서  
진행한다면 위기상황에 반응하는 업종조합을 더욱 다양하게 추출해 낼 수 있을 것이며,  
프로젝트의 활용도 또한 높아질 것이다.

### 프로젝트 한계점

연계소비에 대한 분석의 타당성이 부족하다고 생각함,  
매출 시간대의 유사성만으로 연계소비를 증명하기에는 근거가 부족

신한카드 매출 데이터만을 이용하였기 때문에,  
현금매출에 민감한 업종들의 매출을 정확하게 반영하지 못함.

시간이 부족하여 더 많은 지역과 다양한 업종에 대한 분석을 진행할 수 없었음.

감사합니다.





# Q&A