고깃집
매출 대박을 위하여

목차

주제선정 이유

- 이유는 조원 중 지인이 고깃집을 운영
- -> 점포를 확장할 계획 중

- 실생활에 도움이 되는 주제를 하고싶었음
- -> 좋은 기회라고 생각

- 현재 백석동에서 운영 중
- 다음엔 고양시의 어느 행정동에서 가장 성공확률이 높을지 분석

선형회귀분석을 선택한 이유는?

선형회귀분석은 쌍으로 관찰된 연속형 변수들 사이의 관계에 있어서 한 변수를 원인으로 하고 다른 변수들을 결과로 하는 분석

→ 종속 변수와 독립 변수 간의 선형 관계를 기반으로 하여 새로운 데이터에 대한 값을 예측할 때 사용

→ 성공의 기준(매출)을 종속 변수로 잡고, 다른 요인들을 분석해서 어떤 변수가 가장 매출에 좋은 관련이 많은 지 파악

→ 선형회귀분석은 다시 독립변수의 개수에 따라 단순 선형과 다중선형으로 구분
독립변수가 여러 개이므로 중회귀분석

→ 이를 순위를 매겨 가장 매출이 높을 것을 예상되는 행정동을 찾음

데이터 전처리 과정

```python
import pandas as pd

file_path = '경기도_상권분석통합.csv'
df = pd.read_csv(file_path, encoding='cp949')

# 필요한 열만 선택
selected_columns = ['행정동명', '전체 매출액', '시장성', '성장성', '개폐업안정성', '경쟁도', '유동인구 시장성', '거주인구 시장성']
df_selected = df[selected_columns]

# 전처리된 데이터 출력
print(df_selected)
```

데이터 전처리 과정

	행정동명	전체 매출액	시장성	성장성	개폐업안정성	경쟁도	유동인구 시장성	거주인구 시장성
0	고봉동	115077351	8.0	44.4	25	100.0	12.7	7.7
1	고양동	134326969	16.3	36.8	25	84.6	7.3	6.2
2	관산동	156011971	4.6	41.1	10	99.6	4.9	7.6
3	능곡동	98072296	11.1	65.6	5	55.2	7.3	3.2
4	대덕동	46576482	6.6	34.2	5	50.5	6.7	0.5
5	대화동	197868915	11.4	39.3	35	54.6	8.2	12.1
6	마두1동	145487635	7.0	42.5	10	35.0	7.2	8.0
7	마두2동	93890214	10.2	48.9	10	15.9	3.9	5.4
8	백석1동	159985614	7.6	38.7	15	22.0	10.7	10.1
9	백석2동	109468868	10.1	38.4	15	33.2	8.1	7.0
10	삼송동	140570103	4.7	31.4	20	30.8	4.7	5.3
11	성사1동	113363017	6.6	41.0	15	23.1	4.0	4.5
12	성사2동	52266758	5.7	52.5	10	8.0	2.0	2.3
13	송산동	311840931	11.4	39.0	10	86.2	26.6	17.3
14	송포동	121204682	10.0	42.0	5	81.3	23.5	7.2
15	식사동	267302913	10.7	33.7	30	47.3	11.3	12.5
16	원신동	133880914	11.2	36.5	25	85.3	7.3	4.9
17	일산1동	142482056	5.6	26.4	15	29.0	4.8	9.1
18	일산2동	95166780	5.7	38.2	15	23.7	6.8	6.2
19	일산3동	188994021	8.3	39.7	5	25.6	6.6	11.8
20	장항1동	70731084	6.2	36.7	20	97.7	7.3	3.8
21	장항2동	179920881	10.1	34.7	45	36.6	15.2	9.1
22	정발산동	130660348	8.3	29.1	20	28.8	5.7	7.8
23	주교동	49994174	3.9	44.5	15	20.8	3.0	3.0
24	주엽1동	164680623	9.0	33.8	15	21.1	4.5	9.4
25	주엽2동	160353779	13.0	37.8	5	21.6	4.9	9.7
26	중산동	265906387	9.8	36.2	5	33.9	8.2	15.6
27	창릉동	125887042	18.5	34.3	5	78.7	7.6	4.9
28	탄현동	261622229	11.6	42.7	10	42.2	14.3	17.2
29	풍산동	207587399	18.7	40.8	15	54.4	9.7	12.9
30	행신1동	123452259	3.1	30.1	15	15.2	1.3	4.5
31	행신2동	178848478	9.1	41.6	20	18.4	4.7	6.6
32	행신3동	278410666	9.2	36.5	20	11.3	4.2	9.3
33	행주동	84255328	10.0	53.5	15	66.3	7.3	3.9
34	화전동	156515009	5.9	22.2	10	53.3	3.4	5.8

중회귀분석

```python
import pandas as pd
import numpy as np
import statsmodels.api as sm

# 전체 지표를 독립변수로 두기

# 데이터 로드
df = pd.read_csv('경기도_상권분석통합.csv', encoding='cp949')

# 종속 변수 선택
y = df['전체 매출액']

# 독립 변수 선택
X = df[['시장성', '성장성', '개폐업안정성', '경쟁도', '유동인구 시장성', '거주인구 시장성']]

# 데이터 타입이 숫자형인 열만 선택
X = X.select_dtypes(include=[np.number])

# 회귀 모델 적합
model = sm.OLS(y.astype(float), X.astype(float)).fit()

# 회귀 결과 출력
print(model.summary())
```

성공기준 = 매출

종속변수 = 전체 매출액
전체 매출액 =
배달앱 전체 이용금액 + 외식업종카드 결제 금액

독립변수 =
시장성, 성장성, 개폐업안정성, 경쟁도, 유동인구 시장성, 거주인구 시장성

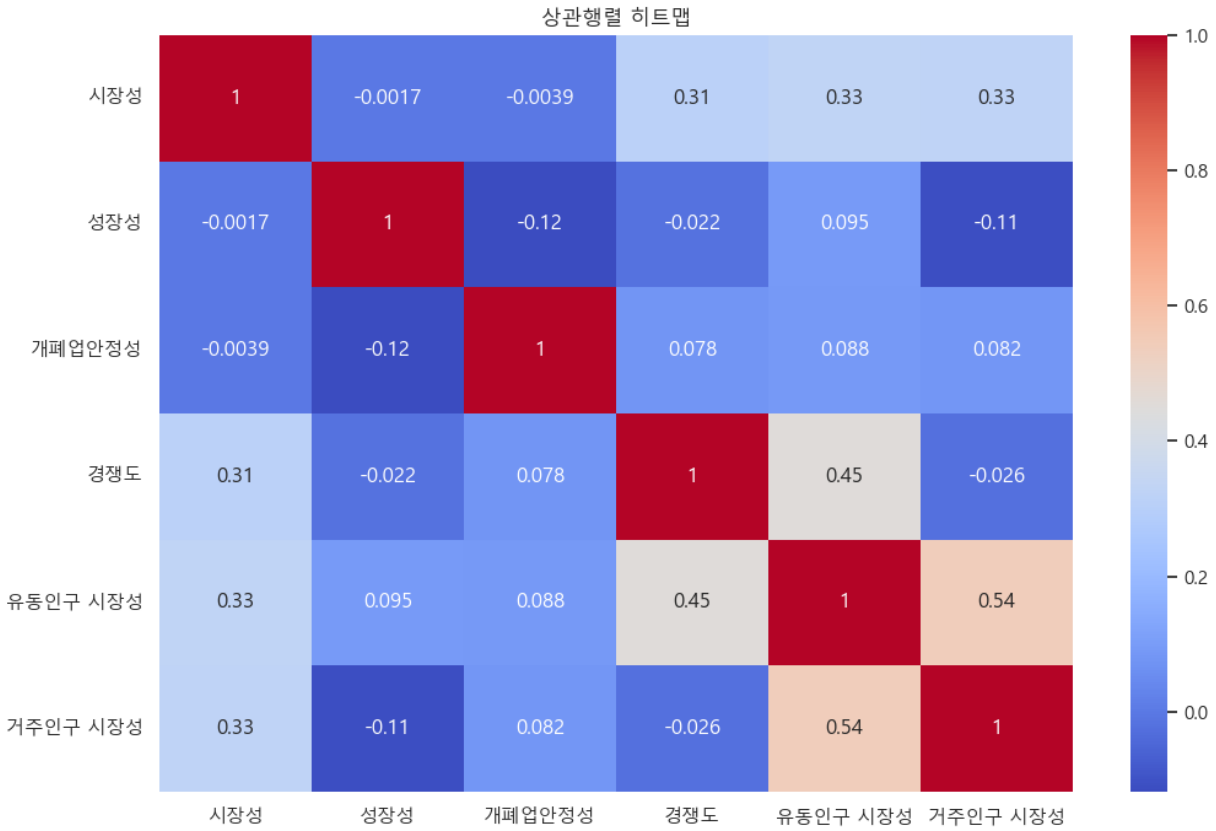
중회귀분석

```
C:\Python\python.exe C:\Pycharm\Chap02\final1.py
                            OLS Regression Results
==============================================================================
Dep. Variable:                 전체 매출   R-squared (uncentered):                   0.955
Model:                            OLS   Adj. R-squared (uncentered):              0.947
Method:                 Least Squares   F-statistic:                              116.8
Date:                Tue, 05 Dec 2023   Prob (F-statistic):                    8.84e-21
Time:                        12:14:59   Log-Likelihood:                         -734.07
No. Observations:                  39   AIC:                                      1480.
Df Residuals:                      33   BIC:                                      1490.
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
시장성         4.154e+06   1.92e+06      2.158      0.038    2.38e+05    8.07e+06
성장성        -7.134e+04   4.46e+05     -0.160      0.874    -9.8e+05    8.37e+05
개폐업안정성      6.115e+05   6.96e+05      0.879      0.386   -8.04e+05    2.03e+06
경쟁도        -1.804e+04   2.77e+05     -0.065      0.948   -5.81e+05    5.45e+05
유동인구 시장성   -1.164e+06   1.73e+06     -0.671      0.507   -4.69e+06    2.37e+06
거주인구 시장성     1.54e+07   2.05e+06      7.512      0.000    1.12e+07    1.96e+07
==============================================================================
Omnibus:                        5.052   Durbin-Watson:                   1.150
Prob(Omnibus):                  0.080   Jarque-Bera (JB):                4.045
Skew:                           0.778   Prob(JB):                        0.132
Kurtosis:                       3.258   Cond. No.                         26.9
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Process finished with exit code 0
```

P-value값을 보면 성장성,경쟁도,유동인구 시장성은 0.05 초과

->각 변수들의 상관관계를 분석

상관관계분석




```python
# 상관 행렬 확인
correlation_matrix = X.corr()

# Heatmap 시각화
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm")
plt.title("상관행렬 히트맵")
plt.show()

# 회귀 모델 적합
model = sm.OLS(y.astype(float), X.astype(float)).fit()
```

상관계수가 대부분 낮음(0.5미만)
하지만 거주인구시장성&유동인구 시장성 지표가 0.5를 넘음

→ 이번에는 변수선택법(전진단계별 선택법) 사용

변수선택(Python)

```python
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt

df = pd.read_csv('경기도_상권분석통합.csv', encoding='cp949')

print(df.head(5))

##전진 단계별 선택별
variables = df.columns[10:16].tolist()
y = df['전체 매출액']
selected_variables = []
sl_enter = 0.05
sl_remove = 0.05

sv_per_step = []
adjusted_r_squared = []
steps = []
step = 0
while len(variables) > 0:
    remainder = list(set(variables) - set(selected_variables))
    pval = pd.Series(index=remainder)

    for col in remainder:
        X = df[selected_variables + [col]]
        X = sm.add_constant(X)
        model = sm.OLS(y, X).fit()
        pval[col] = model.pvalues[col]
```

```python
    min_pval = pval.min()
    if min_pval < sl_enter:
        selected_variables.append(pval.idxmin())

        while len(selected_variables) > 0:
            selected_X = df[selected_variables]
            selected_X = sm.add_constant(selected_X)
            selected_pval = sm.OLS(y, selected_X).fit().pvalues[1:]
            max_pval = selected_pval.max()
            if max_pval >= sl_remove:
                remove_variable = selected_pval.idxmax()
                selected_variables.remove(remove_variable)
            else:
                break

        step += 1
        steps.append(step)
        adj_r_squared = sm.OLS(y, sm.add_constant(df[selected_variables])).fit().rsquared_adj
        adjusted_r_squared.append(adj_r_squared)
        sv_per_step.append(selected_variables.copy())
    else:
        break

print(selected_variables)
```

변수선택(Python)

```
C:\Python\python.exe C:\Pycharm\Chap02\final2.py
      기준년월       행정동코드 광역시도명     시군구명  ... 개폐업안정성    경쟁도  유동인구 시장성  거주인구 시장성
0  202309  4128560000    경기도  고양시일산동구  ...      25  100.0       12.7        7.7
1  202309  4128159000    경기도   고양시덕양구  ...      25   84.6        7.3        6.2
2  202309  4128160000    경기도   고양시덕양구  ...      10   99.6        4.9        7.6
3  202309  4128161000    경기도   고양시덕양구  ...       5   55.2        7.3        3.2
4  202309  4128167000    경기도   고양시덕양구  ...       5   50.5        6.7        0.5

[5 rows x 16 columns]
['거주인구 시장성', '성장성']

Process finished with exit code 0
```

결과값을 살펴보면 최종적으로 선택된 변수는 '거주인구 시장성'과 '성장성'

이 두 변수는 전체 매출액에 대한 가장 유의미한 영향을 미치는 것으로 판단

변수선택(Python)

```python
import pandas as pd
import numpy as np
import statsmodels.api as sm

# 전체 지표를 독립변수로 두기

# 데이터 로드
df = pd.read_csv('경기도_상권분석통합.csv', encoding='cp949')



# 종속 변수 선택
y = df['전체 매출액']

# 독립 변수 선택
X = df[['성장성', '거주인구 시장성']]

# 데이터 타입이 숫자형인 열만 선택
X = X.select_dtypes(include=[np.number])

# 회귀 모델 적합
model = sm.OLS(y.astype(float), X.astype(float)).fit()

# 회귀 결과 출력
print(model.summary())
```

성장성, 거주인구 시장성을
독립 변수로 선택하여
중회귀분석을 실행

변수선택(Python)

```
C:\Python\python.exe C:\Pycharm\Chap02\final3.py
                            OLS Regression Results
==============================================================================
Dep. Variable:                전체 매출액   R-squared (uncentered):                   0.946
Model:                            OLS   Adj. R-squared (uncentered):              0.943
Method:                 Least Squares   F-statistic:                              323.0
Date:                Tue, 05 Dec 2023   Prob (F-statistic):                    3.76e-24
Time:                        20:01:51   Log-Likelihood:                         -737.70
No. Observations:                  39   AIC:                                      1479.
Df Residuals:                      37   BIC:                                      1483.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
성장성         6.142e+05   3.31e+05      1.856      0.071   -5.63e+04    1.28e+06
거주인구 시장성    1.681e+07   1.49e+06     11.278      0.000    1.38e+07    1.98e+07
==============================================================================
Omnibus:                        9.030   Durbin-Watson:                   0.933
Prob(Omnibus):                  0.011   Jarque-Bera (JB):                8.060
Skew:                           1.058   Prob(JB):                       0.0178
Kurtosis:                       3.692   Cond. No.                         9.19
==============================================================================
```

다른 값은 정상 BUT
더빈-왓슨비가 2에 근접하지 않음

➔ 파이썬에서는 OLS결과를 보면서 수작업으로 변수를 조정해야 하지만 R에서 step() 함수를 활용하여 변수선택 한 경우의 수를 한번에 확인 가능

```
Start:  AIC=1363.26
전체매출액 ~ 시장성 + 성장성 + 개폐업안정성 +
    경쟁도 + 유동인구시장성 + 거주인구시장성

                Df  Sum of Sq        RSS    AIC
- 개폐업안정성     1 1.8256e+10 4.1314e+16 1361.3
- 유동인구시장성   1 5.1463e+12 4.1319e+16 1361.3
- 경쟁도           1 9.4937e+14 4.2263e+16 1362.2
<none>                         4.1314e+16 1363.3
- 시장성           1 3.5427e+15 4.4857e+16 1364.5
- 성장성           1 7.1585e+15 4.8473e+16 1367.5
- 거주인구시장성   1 5.5086e+16 9.6400e+16 1394.3

Step:  AIC=1361.26
전체매출액 ~ 시장성 + 성장성 + 경쟁도 + 유동인구시장성 +
    거주인구시장성

                Df  Sum of Sq        RSS    AIC
- 유동인구시장성   1 5.1309e+12 4.1319e+16 1359.3
- 경쟁도           1 9.5312e+14 4.2267e+16 1360.2
<none>                         4.1314e+16 1361.3
- 시장성           1 3.5559e+15 4.4870e+16 1362.5
+ 개폐업안정성     1 1.8256e+10 4.1314e+16 1363.3
- 성장성           1 7.2462e+15 4.8560e+16 1365.6
- 거주인구시장성   1 5.5208e+16 9.6522e+16 1392.3

Step:  AIC=1359.27
전체매출액 ~ 시장성 + 성장성 + 경쟁도 + 거주인구시장성

                Df  Sum of Sq        RSS    AIC
- 경쟁도           1 1.2546e+15 4.2574e+16 1358.4
<none>                         4.1319e+16 1359.3
- 시장성           1 3.5547e+15 4.4874e+16 1360.5
+ 유동인구시장성   1 5.1309e+12 4.1314e+16 1361.3
+ 개폐업안정성     1 2.8180e+09 4.1319e+16 1361.3
- 성장성           1 7.5868e+15 4.8906e+16 1363.8
- 거주인구시장성   1 9.0058e+16 1.3138e+17 1402.4
```

```
> #데이터 바인딩
> data <-read.csv("C:/Users/abise/인공지능/상권분석통합_경기도_202309.csv",header = T, fileEncoding = "CP949")
>
> #데이터 열 선택
> selected_data <- data[, 9:15]
>
> #데이터 확인
> head(selected_data)
  전체매출액 시장성 성장성 개폐업안정성 경쟁도 유동인구시장성 거주인구시장성
1  115077351    8.0   44.4          25  100.0          12.7           7.7
2  134326969   16.3   36.8          25   84.6           7.3           6.2
3  156011971    4.6   41.1          10   99.6           4.9           7.6
4   98072296   11.1   65.6           5   55.2           7.3           3.2
5   46576482    6.6   34.2           5   50.5           6.7           0.5
6  197868915   11.4   39.3          35   54.6           8.2          12.1
>
> # 전체 모델 생성
> full_model <- lm(전체매출액 ~ 시장성 + 성장성 + 개폐업안정성 + 경쟁도 + 유동인구시장성 + 거주인구시장성, data = selected_data)
>
> # 단계적 회귀 적합
> step_model <- step(full_model, direction = "both")
```

```
Step:  AIC=1358.43
전체매출액 ~ 시장성 + 성장성 + 거주인구시장성

                Df  Sum of Sq        RSS    AIC
<none>                         4.2574e+16 1358.4
- 시장성           1 2.5579e+15 4.5132e+16 1358.7
+ 경쟁도           1 1.2546e+15 4.1319e+16 1359.3
+ 유동인구시장성   1 3.0658e+14 4.2267e+16 1360.2
+ 개폐업안정성     1 1.1256e+13 4.2563e+16 1360.4
- 성장성           1 7.3555e+15 4.9929e+16 1362.7
- 거주인구시장성   1 9.5236e+16 1.3781e+17 1402.2
```

R프로그램에서 AIC값이 가장 작게 나온 결과를 보니
시장성, 성장성, 거주인구시장성 으로 변수선택을 하는 것이 가장 이상적

변수선택(R)

```python
import pandas as pd
import numpy as np
import statsmodels.api as sm

# 전체 지표를 독립변수로 두기

# 데이터 로드
df = pd.read_csv('경기도_상권분석통합.csv', encoding='cp949')


# 종속 변수 선택
y = df['전체 매출액']

# 독립 변수 선택
X = df[['시장성', '성장성', '거주인구 시장성']]

# 데이터 타입이 숫자형인 열만 선택
X = X.select_dtypes(include=[np.number])

# 회귀 모델 적합
model = sm.OLS(y.astype(float), X.astype(float)).fit()

# 회귀 결과 출력
print(model.summary())
```

따라서 독립변수에
'**시장성, 성장성, 거주인구 시장성** '
추가해서 회귀분석 진행

변수선택(R)

```
C:\Python\python.exe C:\Pycharm\Chap02\final3.py
                            OLS Regression Results
==============================================================================================
Dep. Variable:                 전체 매출액   R-squared (uncentered):                   0.953
Model:                            OLS   Adj. R-squared (uncentered):              0.949
Method:                 Least Squares   F-statistic:                               243.9
Date:                Tue, 05 Dec 2023   Prob (F-statistic):                     5.75e-24
Time:                        20:08:38   Log-Likelihood:                         -734.88
No. Observations:                  39   AIC:                                      1476.
Df Residuals:                      36   BIC:                                      1481.
Df Model:                           3
Covariance Type:            nonrobust
================================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
시장성         4.038e+06   1.71e+06      2.363      0.024    5.73e+05     7.5e+06
성장성         1.395e+04   4.02e+05      0.035      0.973   -8.02e+05     8.3e+05
거주인구 시장성    1.497e+07   1.61e+06      9.319      0.000    1.17e+07    1.82e+07
==============================================================================
Omnibus:                        5.842   Durbin-Watson:                   1.174
Prob(Omnibus):                  0.054   Jarque-Bera (JB):                4.882
Skew:                           0.857   Prob(JB):                       0.0871
Kurtosis:                       3.260   Cond. No.                         13.4
==============================================================================
```

그 결과 이전 지표보다 안정적인 값 출력

행정동명	평균지표
능곡동	26.633333
풍산동	24.133333
탄현동	23.833333
송산동	22.566667
행주동	22.466667
마두2동	21.500000
대화동	20.933333
중산동	20.533333
성사2동	20.166667
주엽2동	20.166667
고봉동	20.033333
일산3동	19.933333
고양동	19.766667
송포동	19.733333
창릉동	19.233333
마두1동	19.166667
행신2동	19.100000
식사동	18.966667
백석1동	18.800000
백석2동	18.500000
행신3동	18.333333
장항2동	17.966667
관산동	17.766667
원신동	17.533333
주엽1동	17.400000
성사1동	17.366667
주교동	17.133333
화정2동	17.100000
화정1동	17.000000
일산2동	16.700000
흥도동	16.600000
장항1동	15.566667
정발산동	15.066667
삼송동	13.800000

```python
import pandas as pd

# 데이터프레임 로드
df = pd.read_csv('경기도_상권분석통합.csv', encoding='cp949')

# 시장성, 성장성, 거주인구 시장성을 더한 새로운 열 생성
df['평균지표'] = (df['시장성'] + df['성장성'] + df['거주인구 시장성']) / 3

# 행정동명을 기준으로 그룹화하고 각 그룹에 대해 '평균지표'의 평균 계산
result_df = df.groupby('행정동명')['평균지표'].mean().reset_index()
result_df_sorted = result_df.sort_values(by='평균지표', ascending=False)

print(result_df_sorted)
```

따라서 중회귀 분석을 통해
매출액과 가장 상관관계가 높다고 예상되는
독립변수는?
시장성, 성장성, 거주인구 시장성

시장성, 성장성, 거주인구 시장성 수치를 더하여
평균을 내어 내림차순으로 정렬하여 순위를 매기면
왼쪽과 같은 결과 출력

결론

지점 확장 시 매출액이 가장 높을 것 같은 상위 5개의 동

1순위: 능곡동

2순위: 풍산동

3순위: 탄현동

4순위: 송산동

5순위: 행주동