

데이터처리프로그래밍(02) - 컴퓨터학과 20201001 이승연

1. 성북구 거주민의 나이대별 소비액 구하기

```
# 성북구 거주민의 정보만 남기고 지우기
index = bc_card[bc_card['CSTRM_CTY_RGNUM'] != '성북구'].index
seongbuk_bc_card = bc_card.drop(index)
#print(seongbuk_bc_card.tail(10))

# 인덱스 리셋
seongbuk_bc_card.reset_index(drop=True, inplace=True)
#print(seongbuk_bc_card.tail(10))

# 나이대별 소비액 구하기
seongbuk_age=pd.pivot_table(seongbuk_bc_card, index=['AGE_VAL'], aggfunc='sum')
seongbuk_age=seongbuk_age.drop(['ADM1_CTY_NO', 'CTY_RGN_NO', 'TP_GRP_NO', 'MAIN_BUZ_CODE', 'REG_YNM', 'CNT', 'CSTRM_CTY_RGN_NO', 'CSTRM_MEGA_CTY_NO', 'FLC', 'MEGA_CTY_NO', 'TP_BUZ_NO', 'SEX_CTGO_CD'], axis=1)
print(seongbuk_age)
```

AGE_VAL	AMT
20대	627141809
20세 미만	9734990
30대	1172119621
40대	1097412414
50대	75490916
60대 이상	536184728

<python-input-5-9718ec5aed5>11: FutureWarning: pivot_table dropped a column because it failed to aggregate. This behavior is deprecated and will raise in a future version of pandas. Select only the columns that can be aggregated.
seongbuk_age=pd.pivot_table(seongbuk_bc_card, index=['AGE_VAL'], aggfunc='sum')

2. 성북구 거주민의 남/여 소비액 구하기

```
# 성별을 인덱스로
seongbuksex=pd.pivot_table(seongbuk_bc_card, index=['SEX_CTGO_CD'], aggfunc='sum')
#print(seongbuksex)

seongbuksex=seongbuksex.drop(['ADM1_CTY_NO', 'CTY_RGN_NO', 'TP_GRP_NO', 'MAIN_BUZ_CODE', 'REG_YNM', 'CNT', 'CSTRM_CTY_RGN_NO', 'CSTRM_MEGA_CTY_NO', 'FLC', 'MEGA_CTY_NO', 'TP_BUZ_NO'], axis=1)
print(seongbuksex)
```

SEX_CTGO_CD	AMT
1	2014486610
2	2172966768

<python-input-6-60767c1e69b>2: FutureWarning: pivot_table dropped a column because it failed to aggregate. This behavior is deprecated and will raise in a future version of pandas. Select only the columns that can be aggregated.
seongbuksex=pd.pivot_table(seongbuk_bc_card, index=['SEX_CTGO_CD'], aggfunc='sum')

3. 성북구 거주민 중 여성의 편의점 카드 이용 건수 구하기

```
seongbuk_female = seongbuk_bc_card[seongbuk_bc_card['SEX_CTGO_CD'] == 2]
seongbuk_female_conv_cnt = seongbuk_female[seongbuk_female['TP_BUZ_NO'] == 4010]['CNT'].sum()

print(seongbuk_female_conv_cnt)
```

12867

4. 성북구 거주민 중 여성의 편의점 소비액 구하기

```
seongbuk_female_spending = seongbuk_bc_card[seongbuk_bc_card['SEX_CTGO_CD'] == 2]['AMT'].sum()

print(seongbuk_female_spending)
```

2172966768

5. 성북구의 서양음식, 일반한식 카드 이용 건수와 소비액 구하기

```
seongbuk_food_west=seongbuk_bc_card[seongbuk_bc_card['TP_BUZ_NM']=='서양음식']
seongbuk_food_kor=seongbuk_bc_card[seongbuk_bc_card['TP_BUZ_NM']=='일반한식']

# 서양음식, 일반한식 카드 이용 건수 구하기
count_west = seongbuk_food_west['CNT'].sum()
count_kor = seongbuk_food_kor['CNT'].sum()

# 서양음식, 일반한식 소비액 구하기
spending_west = seongbuk_food_west['AMT'].sum()
spending_kor = seongbuk_food_kor['AMT'].sum()

data = {
    '이용건수': [count_west, count_kor],
    '소비액': [spending_west, spending_kor]
}

df = pd.DataFrame(data, index=['서양음식', '일반한식'])
print(df)
```

	이용건수	소비액
서양음식	19936	175840421
일반한식	21750	568218638

6. 성북구의 일반음식, 음식료품 소비액 구하기

```
seongbuk_food=seongbuk_bc_card[seongbuk_bc_card['TP_GRP_NM']=='일반음식']
seongbuk_food_ingredient=seongbuk_bc_card[seongbuk_bc_card['TP_GRP_NM']=='음식료품']

spending_food = seongbuk_food['AMT'].sum()
spending_food_ingredient = seongbuk_food_ingredient['AMT'].sum()

print(spending_food)
print(spending_food_ingredient)
```

```
818332189
146177444
```

7. 강남구, 성북구 거주민의 음식 소비액 평균 구하기

```
# 강남구 거주민 정보 구하기
index2 = bc_card[bc_card['CSTMRTY_RGN_NM'] != '강남구'].index
gangnam_bc_card = bc_card.drop(index2)

gangnam_bc_card.reset_index(drop=True, inplace=True)

# 강남구 거주민 음식 소비액
gangnam_food = gangnam_bc_card[gangnam_bc_card['MAIN_BUZ_DESC'] == '음식']
gangnam_food_mean = gangnam_food['AMT'].mean()

# 성북구 거주민 음식 소비액
seongbuk_food = seongbuk_bc_card[seongbuk_bc_card['MAIN_BUZ_DESC'] == '음식']
seongbuk_food_mean = seongbuk_food['AMT'].mean()

print(gangnam_food_mean)
print(seongbuk_food_mean)
```

```
2927120.643678161
1323043.5721153845
```

8. 강남구의 나이대별 소비액과 카드 이용 건수 구하기

```
# 강남구 거주민의 정보
index3 = bc_card[bc_card['CTY_RGN_NM'] != '강남구'].index
gangnam_bc_card = bc_card.drop(index3)
#print(gangnam_bc_card.tail(10))

# 인덱스 리셋
gangnam_bc_card.reset_index(drop=True, inplace=True)
#print(gangnam_bc_card.tail(10))

# 나이대별 소비액 구하기
gangnam_age = pd.pivot_table(gangnam_bc_card, index=['AGE_VAL'], aggfunc='sum')
gangnam_age = gangnam_age.drop(['ADM_CTG_NO', 'CTY_RGN_NO', 'TP_GRP_NO', 'MAIN_BUZ_CODE', 'REG_YRMM', 'CSTMRTY_RGN_NO', 'CSTMRTY_MEGA_CTG_NO', 'FLC', 'MEGA_CTG_NO', 'TP_BUZ_NO', 'SEX_CTGO_CD'], axis=1)
print(gangnam_age)
```

	AMT	CNT
AGE_VAL		
20대	3830783543	226261
20세 미만	100503859	15703
30대	8765073786	302131
40대	1023498201	268518
50대	7556294764	172411
60대 이상	5008044339	101481

<ipython-input-54-13d0029b3b73>:11: FutureWarning: pivot_table dropped a column because it failed to aggregate. This behavior is deprecated and will raise in a future version of pandas. Select only the columns that can be aggregated.
gangnam_age = pd.pivot_table(gangnam_bc_card, index=['AGE_VAL'], aggfunc='sum')

9. 강남구 30대의 대분류별 소비액 구하기

```
# 강남구 30대
gangnam_30s = gangnam_bc_card[gangnam_bc_card['AGE_VAL'] == 30]

# 대분류별 소비액 구하기
gangnam_30s_spending = pd.pivot_table(gangnam_bc_card, index=['MAIN_BUZ_DESC'], aggfunc='sum')
gangnam_30s_spending = gangnam_30s_spending.drop(['ADM_CTG_NO', 'CTY_RGN_NO', 'TP_GRP_NO', 'MAIN_BUZ_CODE', 'REG_YRMM', 'CNT', 'CSTMRTY_RGN_NO', 'CSTMRTY_MEGA_CTG_NO', 'FLC', 'MEGA_CTG_NO', 'TP_BUZ_NO', 'SEX_CTGO_CD'], axis=1)
print(gangnam_30s_spending)
```

	AMT
MAIN_BUZ_DESC	
18E	1287378903
기타	970493596
내구재	165710300
문화	1163617519
생활	2862663775
쇼핑	1126745490
유흥	280034151
음식	3714848078

<ipython-input-56-2bc42dbbfac2>:5: FutureWarning: pivot_table dropped a column because it failed to aggregate. This behavior is deprecated and will raise in a future version of pandas. Select only the columns that can be aggregated.
gangnam_30s_spending = pd.pivot_table(gangnam_bc_card, index=['MAIN_BUZ_DESC'], aggfunc='sum')

10. 강남구의 쇼핑에 대한 카드 이용 건수, 소비액 구하기

```
gangnam_shopping=gangnam_bc_card[gangnam_bc_card['MAIN_BUZ_DESC']=='쇼핑']

gangnam_shopping_cnt = gangnam_shopping['CNT'].sum()
gangnam_shopping_amt = gangnam_shopping['AMT'].sum()

data = {
    '이용건수': [gangnam_shopping_cnt],
    '소비액': [gangnam_shopping_amt]
}

df = pd.DataFrame(data, index=['쇼핑'])
print(df)
```

	이용건수	소비액
쇼핑	36467	1126745490