상권 분석 보고서

전처리 데이터 분석

restaurant_with_universities.csv 분석

- hdfs에서 파일을 읽으면 대학교 column에 두 개의 값이 들어있는 경우 "경희대학교, 한국외국어대학교" 형식으로 들어있어 미리 split했음
  ➔ trans_split_univ.csv

analysisData.csv 만들기

- trans_split_univ.csv와 university_info.csv를 hive를 사용해 join 및 변형
  ➔ analysisData.csv

analysisData 분석

	대학명	상권업종중분류명	상권업종소분류명	개수	밀도
0	가톨릭대학교 _제2캠퍼스	비알코올	카페	22	0.042807
1	가톨릭대학교 _제2캠퍼스	한식	백반/한정식	19	0.042807
2	가톨릭대학교 _제2캠퍼스	기타 간이	김밥/만두/분식	12	0.042807
3	가톨릭대학교 _제2캠퍼스	구내식당·뷔페	구내식당	11	0.042807
4	가톨릭대학교 _제2캠퍼스	기타 간이	빵/도넛	11	0.042807

➔ analysisData.csv

- 데이터 각 column 데이터 개수 확인

```
is_unique_0 = data['대학명'].nunique() == len(data['대학명'])
print(f"대학명 column이 unique한가? {is_unique_0}")
print(data['대학명'].nunique())
```

```
대학명 column이 unique한가? False
46
```

```
is_unique_1 = data['상권업종중분류명'].nunique() == len(data['상권업종중분류명'])
print(f"상권업종중분류명 column이 unique한가? {is_unique_1}")
print(data['상권업종중분류명'].nunique())
```

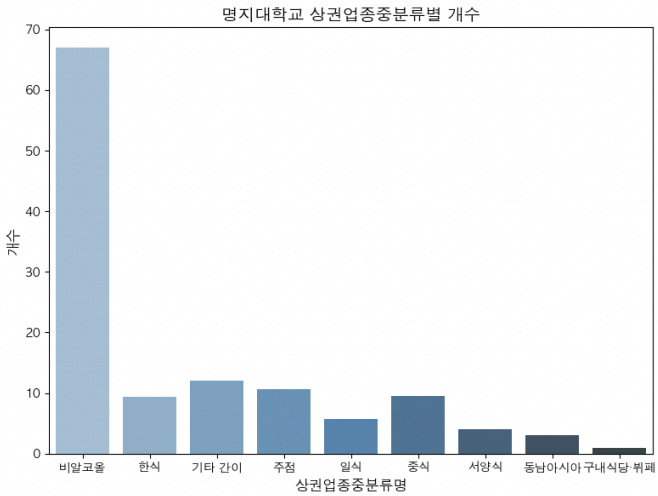
```
상권업종중분류명 column이 unique한가? False
10
```

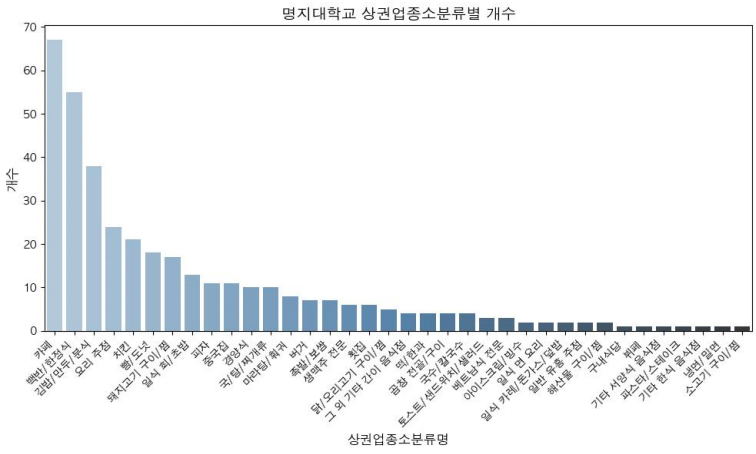
```
is_unique_2 = data['상권업종소분류명'].nunique() == len(data['상권업종소분류명'])
print(f"상권업종소분류명 column이 unique한가? {is_unique_2}")
print(data['상권업종소분류명'].nunique())
```

```
상권업종소분류명 column이 unique한가? False
43
```

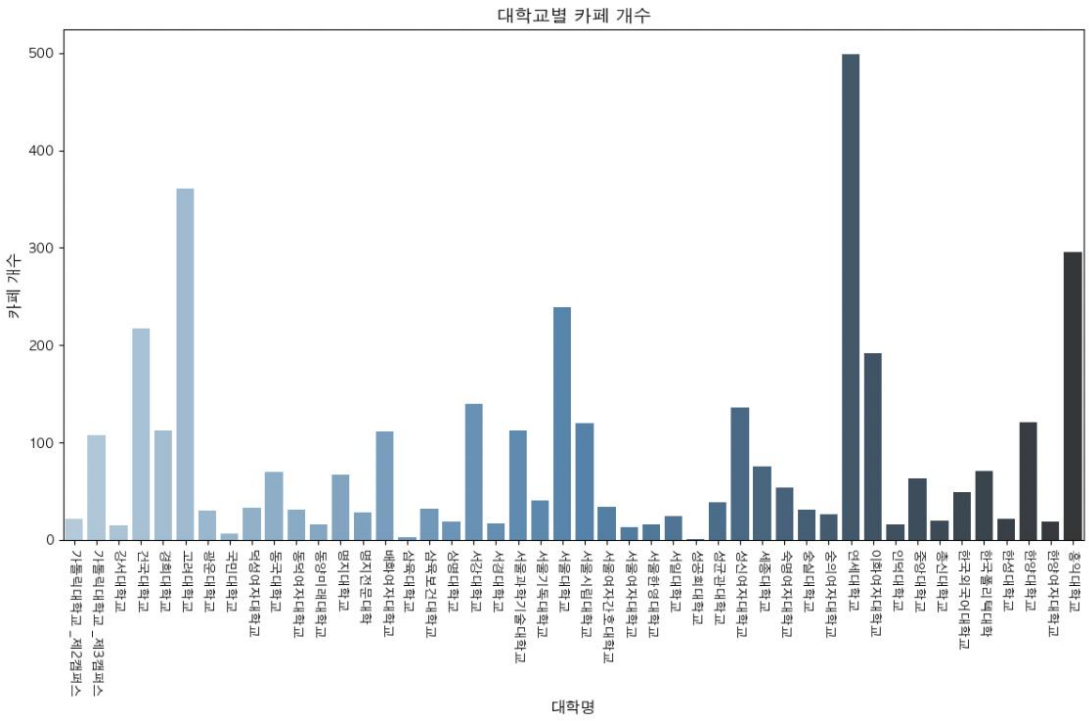
➔ '대학명': 46개, '상권업종중분류명': 10개, '상권업종소분류명': 43개

- 명지대 상권분포 plot

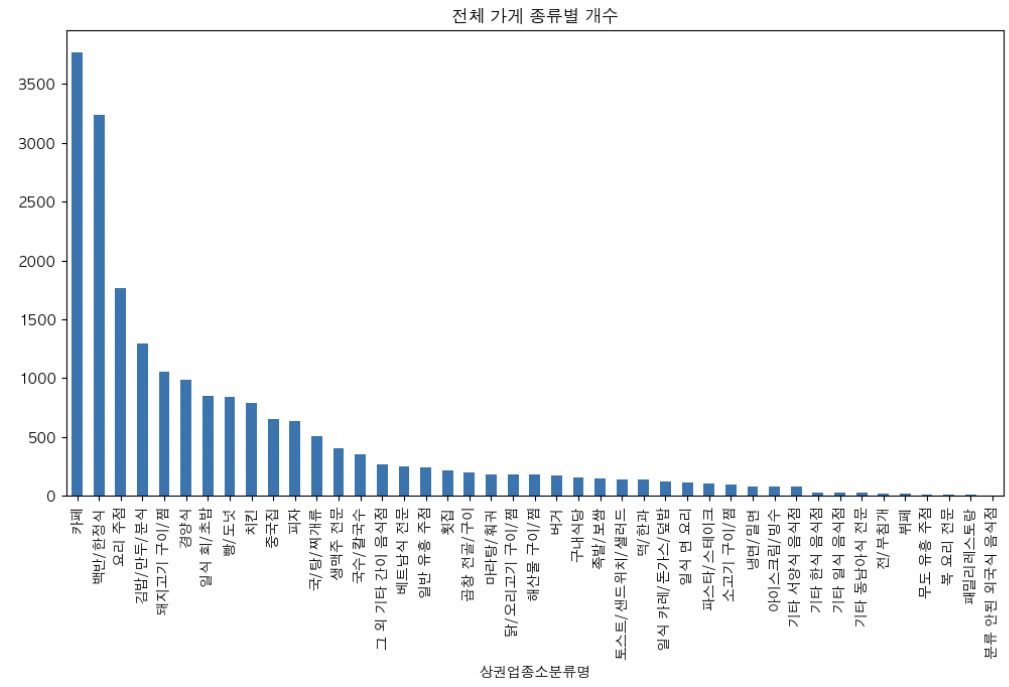




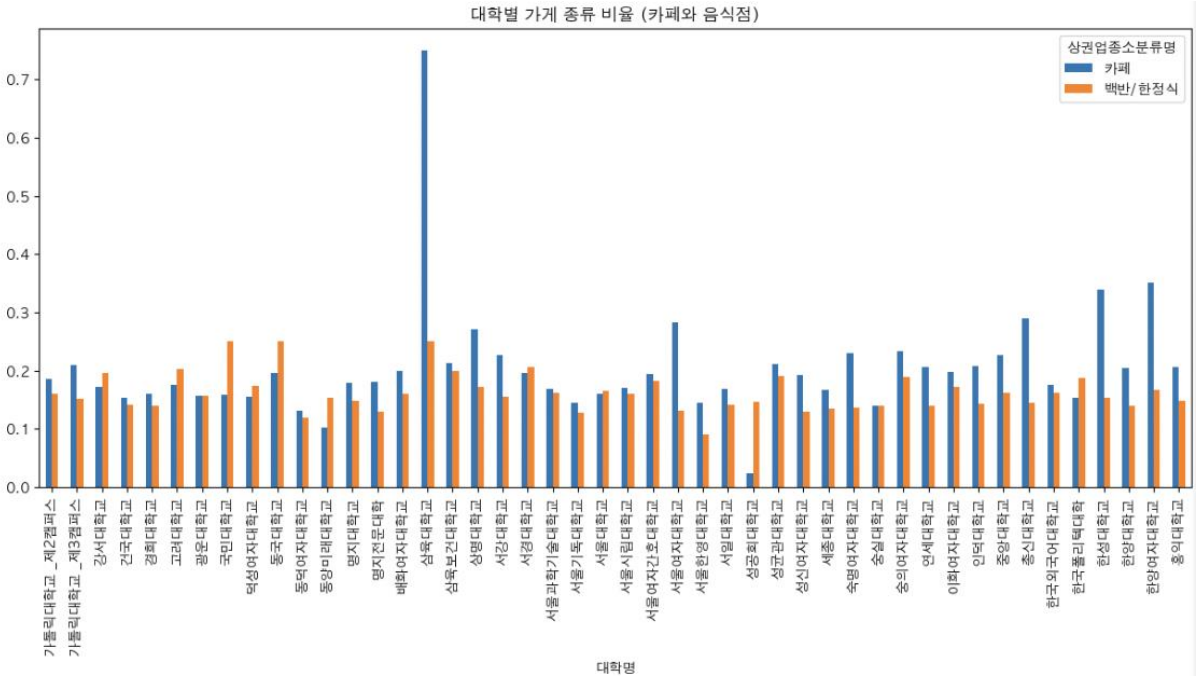
- 대학별 카페 개수



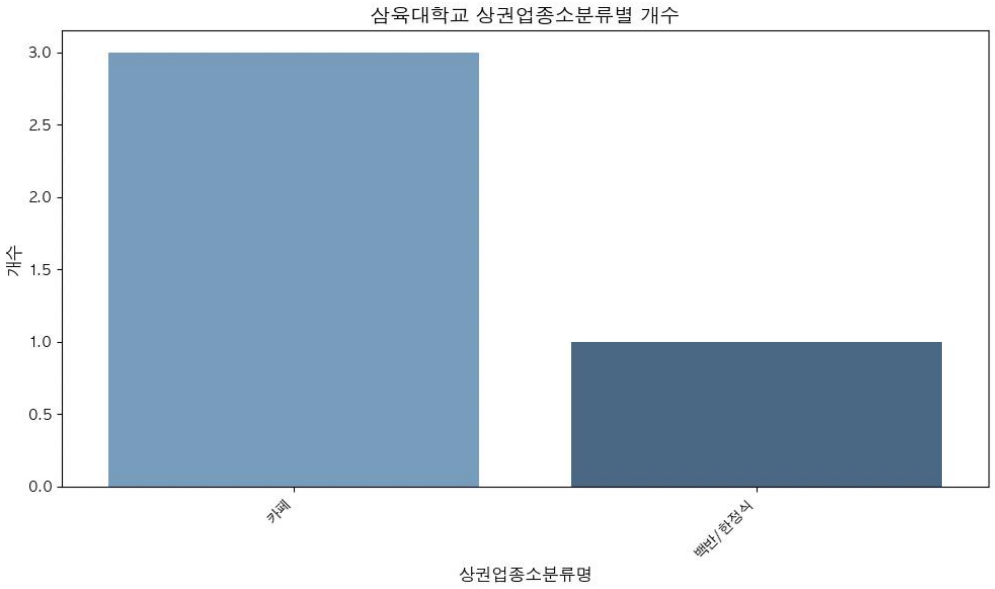
- 전체 가게 소분류명별 개수



- 개수가 가장 많은 카페와 백반/한정식 대학별 비율



- 삼육대의 비율이 이상하여 삼육대의 전체 소분류명 개수 plot 그려봄

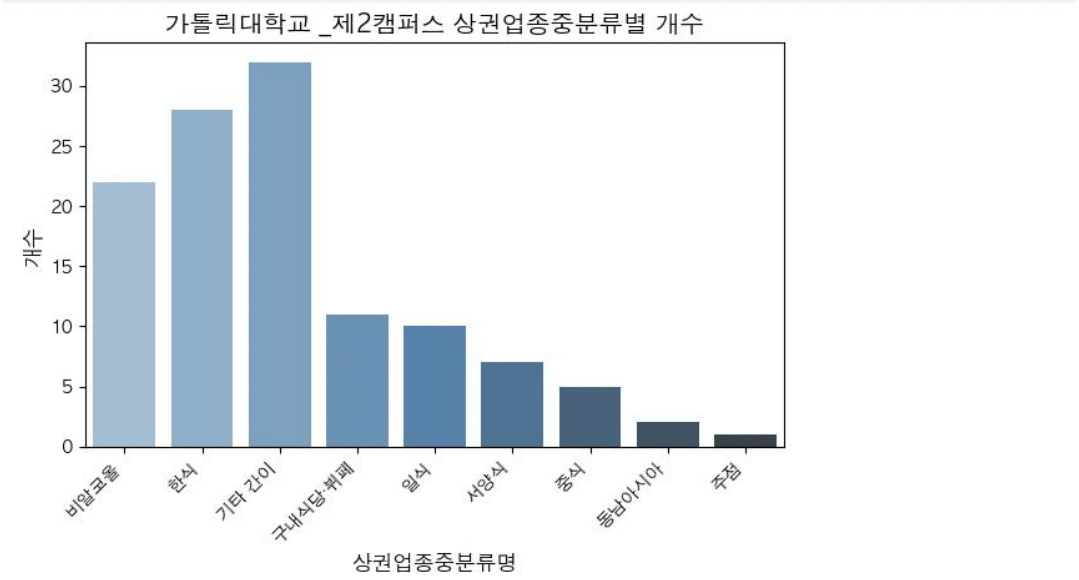


➔ 카페와 백반/한정식 밖에 없음

- 모든 대학의 중분류명별 개수 plot 그림

대학별 상권업종중분류 개수

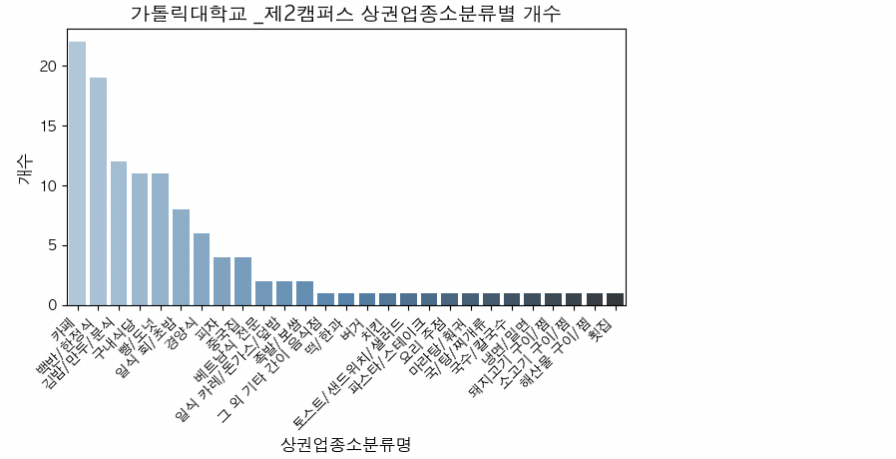
```
for i in data['대학명'].unique():
    filtered_data = data[data['대학명'] == i]
    sns.barplot(data=filtered_data, x='상권업종중분류명', y='개수', estimator=sum, palette='Blues_d',ci=None)
    plt.xlabel('상권업종중분류명', fontsize=12)
    plt.ylabel('개수', fontsize=12)
    plt.title(i+' 상권업종중분류별 개수', fontsize=14)
    plt.xticks(rotation=45, ha='right')
    plt.tight_layout()
    plt.show()
```



- 모든 대학의 소분류명별 개수 plot 그림

대학별 상권업종소분류명 당 개수 Plot

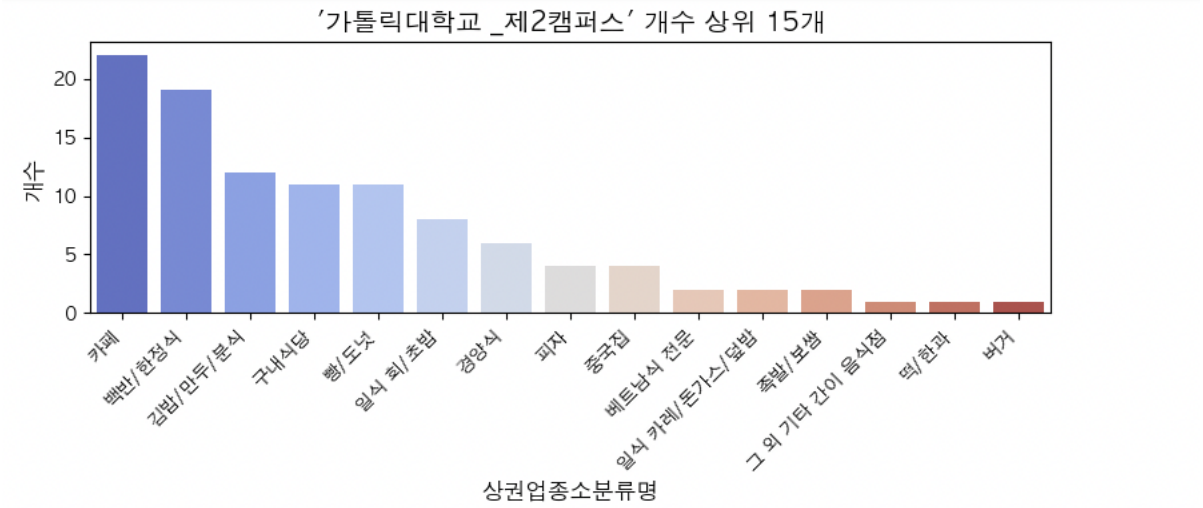
```
for i in data['대학명'].unique():
    filtered_data = data[data['대학명'] == i]
    sns.barplot(data=filtered_data, x='상권업종소분류명', y='개수', palette='Blues_d')
    plt.xlabel('상권업종소분류명', fontsize=12)
    plt.ylabel('개수', fontsize=12)
    plt.title(i+' 상권업종소분류별 개수', fontsize=14)
    plt.xticks(rotation=45, ha='right')
    plt.tight_layout()
    plt.show()
```



- 모든 대학의 상위 15개의 소분류명 개수 plot 그림

대학별 소분류명 개수 상위 15개

```
import matplotlib.pyplot as plt
import seaborn as sns

# 대학명을 하나씩 순회하며 처리
for university in data['대학명'].unique():
    # 특정 대학 데이터를 필터링
    filtered_data = data[data['대학명'] == university]

    # 상위 15개 선택
    top_15_filtered_data = filtered_data.nlargest(15, '개수')

    # 데이터가 존재하면 플롯 생성
    if not top_15_filtered_data.empty:
        plt.figure(figsize=(8, 4))
        sns.barplot(
            data=top_15_filtered_data,
            x='상권업종소분류명',
            y='개수',
            palette='coolwarm'
        )
        plt.title(f"'{university}' 개수 상위 15개", fontsize=14)
        plt.xlabel('상권업종소분류명', fontsize=12)
        plt.ylabel('개수', fontsize=12)
        plt.xticks(rotation=45, ha='right')
        plt.tight_layout()
        plt.show()
```



➔ 하위 15개는 보고서에서 제외.

- 대학별 소분류명의 개수가 많은 순으로 정렬

대학명	소분류명_1	소분류명_2	소분류명_3	소분류명_4	소분류명_5 \
가톨릭대학교 _제2캠퍼스	카페	백반/한정식	김밥/만두/분식	구내식당	빵/도넛
가톨릭대학교 _제3캠퍼스	카페	백반/한정식	요리 주점	김밥/만두/분식	돼지고기 구이/찜
강서대학교	백반/한정식	카페	요리 주점	치킨	김밥/만두/분식
건국대학교	카페	백반/한정식	요리 주점	돼지고기 구이/찜	김밥/만두/분식
경희대학교	카페	백반/한정식	요리 주점	김밥/만두/분식	중국집
고려대학교	백반/한정식	카페	요리 주점	돼지고기 구이/찜	김밥/만두/분식
광운대학교	카페	백반/한정식	김밥/만두/분식	치킨	중국집
국민대학교	백반/한정식	카페	요리 주점	일식 회/초밥	돼지고기 구이/찜
덕성여자대학교	백반/한정식	카페	요리 주점	돼지고기 구이/찜	김밥/만두/분식
동국대학교	백반/한정식	카페	요리 주점	경양식	일식 회/초밥
동덕여자대학교	카페	백반/한정식	김밥/만두/분식	빵/도넛	치킨
동양미래대학교	백반/한정식	카페	김밥/만두/분식	치킨	요리 주점
명지대학교	카페	백반/한정식	김밥/만두/분식	요리 주점	치킨
명지전문대학	카페	김밥/만두/분식	백반/한정식	빵/도넛	요리 주점
배화여자대학교	카페	백반/한정식	경양식	요리 주점	빵/도넛
삼육대학교	카페	백반/한정식	None	None	None
삼육보건대학교	카페	백반/한정식	김밥/만두/분식	돼지고기 구이/찜	요리 주점
상명대학교	카페	백반/한정식	김밥/만두/분식	요리 주점	빵/도넛
서강대학교	카페	백반/한정식	김밥/만두/분식	요리 주점	빵/도넛
서경대학교	백반/한정식	카페	김밥/만두/분식	치킨	구내식당
서울과학기술대학교	카페	백반/한정식	돼지고기 구이/찜	요리 주점	치킨
서울기독대학교	카페	백반/한정식	돼지고기 구이/찜	김밥/만두/분식	빵/도넛
서울대학교	백반/한정식	카페	요리 주점	김밥/만두/분식	돼지고기 구이/찜
서울시립대학교	카페	백반/한정식	김밥/만두/분식	요리 주점	치킨
서울여자간호대학교	카페	백반/한정식	김밥/만두/분식	돼지고기 구이/찜	치킨
서울여자대학교	카페	백반/한정식	김밥/만두/분식	빵/도넛	경양식
서울한영대학교	카페	요리 주점	치킨	백반/한정식	생맥주 전문
서일대학교	카페	백반/한정식	김밥/만두/분식	요리 주점	국/탕/찌개류
성공회대학교	치킨	백반/한정식	국수/칼국수	요리 주점	돼지고기 구이/찜
성균관대학교	카페	백반/한정식	요리 주점	김밥/만두/분식	치킨
성신여자대학교	카페	백반/한정식	요리 주점	돼지고기 구이/찜	김밥/만두/분식
세종대학교	카페	백반/한정식	요리 주점	김밥/만두/분식	치킨
숙명여자대학교	카페	백반/한정식	김밥/만두/분식	경양식	빵/도넛
숭실대학교	카페	백반/한정식	김밥/만두/분식	치킨	요리 주점
숭의여자대학교	카페	백반/한정식	경양식	돼지고기 구이/찜	요리 주점
연세대학교	카페	백반/한정식	요리 주점	경양식	일식 회/초밥
이화여자대학교	카페	백반/한정식	김밥/만두/분식	요리 주점	빵/도넛
인덕대학교	카페	백반/한정식	치킨	김밥/만두/분식	돼지고기 구이/찜
중앙대학교	카페	백반/한정식	김밥/만두/분식	요리 주점	일식 회/초밥
총신대학교	카페	백반/한정식	치킨	김밥/만두/분식	피자
한국외국어대학교	카페	백반/한정식	요리 주점	피자	경양식
한국폴리텍대학	백반/한정식	카페	요리 주점	김밥/만두/분식	돼지고기 구이/찜
한성대학교	카페	백반/한정식	김밥/만두/분식	빵/도넛	경양식
한양대학교	카페	백반/한정식	요리 주점	돼지고기 구이/찜	김밥/만두/분식
한양여자대학교	카페	백반/한정식	국/탕/찌개류	중국집	해산물 구이/찜
홍익대학교	카페	백반/한정식	요리 주점	일식 회/초밥	경양식

➔ 명지전문대를 제외한 모든 대학의 1등과 2등이 카페와 백반/한정식임

- 각 중분류명에 속하는 소분류명의 비율

각 상권업종중분류명에 속하는 상권업종소분류명의 비율

```
for i in data['상권업종중분류명'].unique():
    food_data = data[data['상권업종중분류명'] == i]

    food_count = food_data.groupby('상권업종소분류명')['개수'].sum()
    food_ratio = (food_count / food_count.sum()) * 100  # 퍼센트 비율

    food_ratio_df = pd.DataFrame({'개수': food_count, '비율(%)': food_ratio})

    food_ratio_df = food_ratio_df.sort_values(by='비율(%)', ascending=False)

    print(i)
    print(food_ratio_df, "\n\n")
```

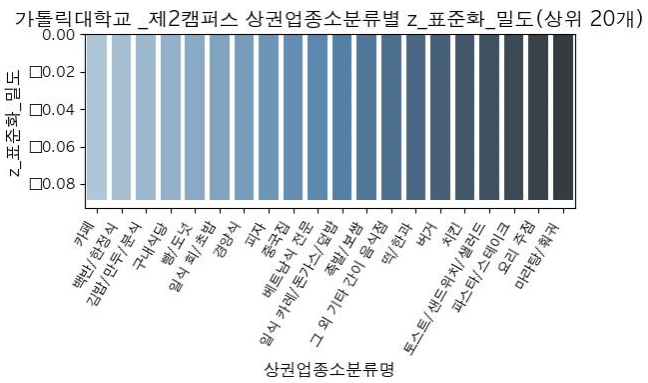
```
비알코올
            개수  비율(%)
상권업종소분류명
카페         3770  100.0


한식
                개수       비율(%)
상권업종소분류명
백반/한정식         3236  51.389551
돼지고기 구이/찜     1052  16.706368
국/탕/찌개류         506   8.035572
국수/칼국수          352   5.589963
횟집               210   3.334921
곱창 전골/구이       200   3.176116
닭/오리고기 구이/찜   182   2.890265
해산물 구이/찜       180   2.858504
족발/보쌈           147   2.334445
소고기 구이/찜        96   1.524535
냉면/밀면            81   1.286327
기타 한식 음식점      28   0.444656
전/부침개            21   0.333492
복 요리 전문          6   0.095283
```

재학생을 반경넓이로 나눠 정규화한 밀도를 이용해 분석

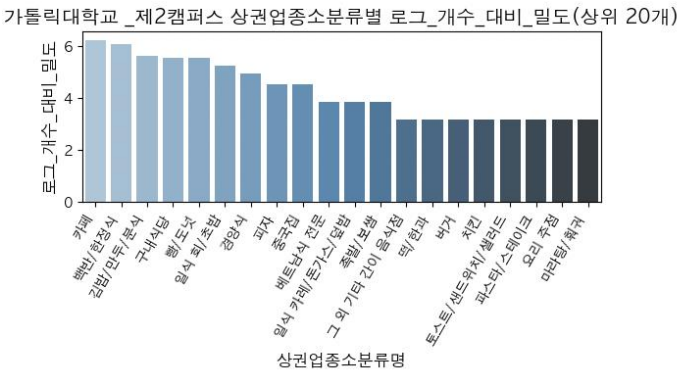
- 밀도 0 방지: 최소값 추가
- 개수 / 밀도 계산, 로그 변환, z-score 표준화 사용

	대학명	상권업종중분류명	상권업종소분류명	개수	밀도	개수_대비_밀도	로그_개수_대비_밀도	z_표준화_밀도
0	가톨릭대학교 _제2캠퍼스	비알코올	카페	22	0.042807	513.932391	6.244036	-0.08864
1	가톨릭대학교 _제2캠퍼스	한식	백반/한정식	19	0.042807	443.850701	6.097739	-0.08864
2	가톨릭대학교 _제2캠퍼스	기타 간이	김밥/만두/분식	12	0.042807	280.326759	5.639517	-0.08864
3	가톨릭대학교 _제2캠퍼스	구내식당·뷔페	구내식당	11	0.042807	256.966195	5.552829	-0.08864
4	가톨릭대학교 _제2캠퍼스	기타 간이	빵/도넛	11	0.042807	256.966195	5.552829	-0.08864
...	...	...	...	...	...	...	...	...
1362	홍익대학교	한식	족발/보쌈	3	0.496890	6.037554	1.951261	-0.08864
1363	홍익대학교	한식	기타 한식 음식점	2	0.496890	4.025036	1.614433	-0.08864
1364	홍익대학교	한식	냉면/밀면	2	0.496890	4.025036	1.614433	-0.08864
1365	홍익대학교	동남아시아	기타 동남아식 전문	1	0.496890	2.012518	1.102776	-0.08864
1366	홍익대학교	한식	전/부침개	1	0.496890	2.012518	1.102776	-0.08864

➔ Plot으로 시각화해보니 로그가 가장 적절해보임



Z-score

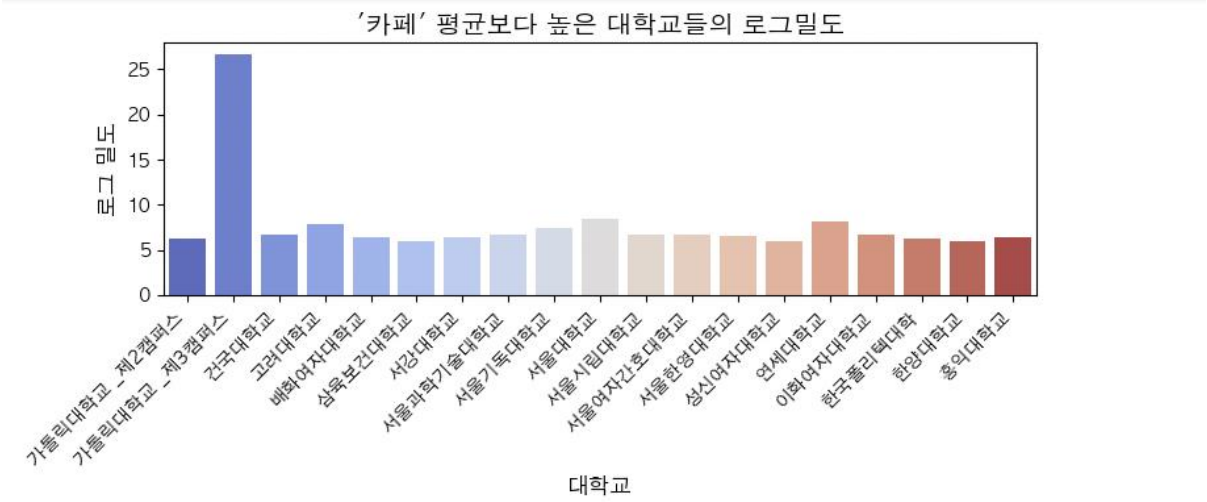


log

- 종소분류명별 평균을 구하고 평균보다 큰 대학교 plot

각 상권업종소분류명의 평균을 구하고 그 평균보다 큰 대학교

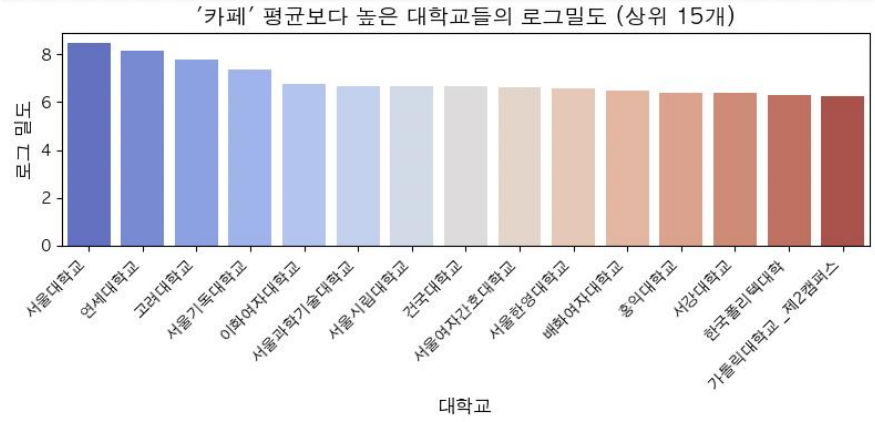
```
import matplotlib.pyplot as plt
import seaborn as sns

# 상권업종소분류명을 하나씩 순회하며 처리
for category in df['상권업종소분류명'].unique():
    # 특정 상권업종소분류명을 가진 데이터 필터링
    filtered_data = df[df['상권업종소분류명'] == category]

    # 로그 밀도의 평균 계산
    mean_log_density = filtered_data['로그_개수_대비_밀도'].mean()

    # 평균보다 큰 데이터 필터링
    above_mean = filtered_data[filtered_data['로그_개수_대비_밀도'] > mean_log_density]

    # 데이터가 존재하면 플롯 생성
    if not above_mean.empty:
        plt.figure(figsize=(8, 4))
        sns.barplot(
            data=above_mean,
            x='대학명',
            y='로그_개수_대비_밀도',
            palette='coolwarm'
        )
        plt.title(f"'{category}' 평균보다 높은 대학교들의 로그밀도", fontsize=14)
        plt.xlabel('대학교', fontsize=12)
        plt.ylabel('로그 밀도', fontsize=12)
        plt.xticks(rotation=45, ha='right')
        plt.tight_layout()
        plt.show()
```



➔ 가톨릭대학교 _제3캠퍼스를 제외하고 다시 그림.

가톨릭대학교 _제3캠퍼스때문에 분포를 파악하기 힘들어 제외했음

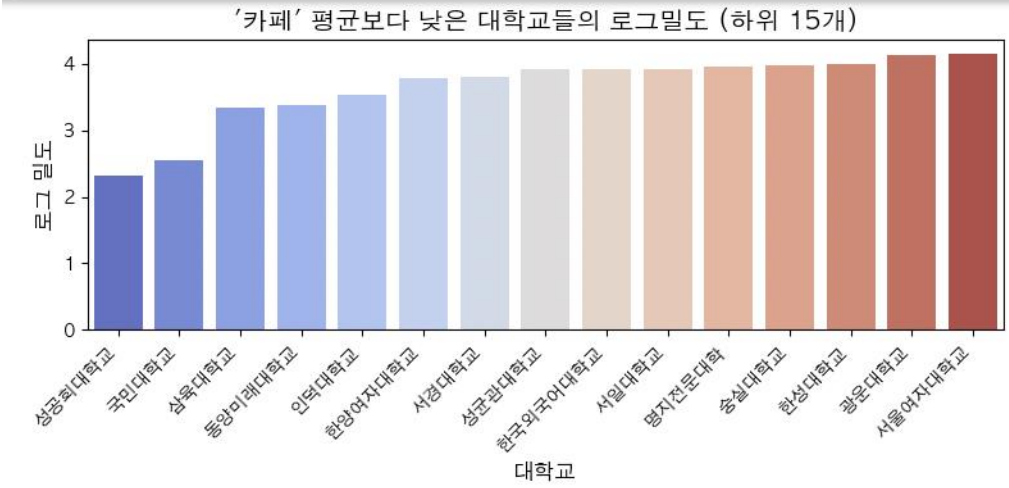
```python
import matplotlib.pyplot as plt
import seaborn as sns

# 상권업종소분류명을 하나씩 순회하며 처리
for category in df['상권업종소분류명'].unique():
    # 특정 상권업종소분류명을 가진 데이터 필터링
    filtered_data = df[df['상권업종소분류명'] == category]
    filtered_data = filtered_data[filtered_data['대학명'] != '가톨릭대학교 _제3캠퍼스']  # 특정 대학교 제외

    # 로그 밀도의 평균 계산
    mean_log_density = filtered_data['로그_개수_대비_밀도'].mean()

    # 평균보다 큰 데이터 필터링
    above_mean = filtered_data[filtered_data['로그_개수_대비_밀도'] > mean_log_density]

    # 상위 10개만 선택
    top_10_filtered_data = above_mean.nlargest(15, '로그_개수_대비_밀도')

    # 데이터가 존재하면 플롯 생성
    if not top_10_filtered_data.empty:
        plt.figure(figsize=(8, 4))
        sns.barplot(
            data=top_10_filtered_data,
            x='대학명',
            y='로그_개수_대비_밀도',
            palette='coolwarm'
        )
        plt.title(f"'{category}' 평균보다 높은 대학교들의 로그밀도 (상위 15개)", fontsize=14)
        plt.xlabel('대학교', fontsize=12)
        plt.ylabel('로그 밀도', fontsize=12)
        plt.xticks(rotation=45, ha='right')
        plt.tight_layout()
        plt.show()
```



- 종소분류명별 평균을 구하고 평균보다 작은 대학교 plot

평균보다 낮은 하위 15개 대학교

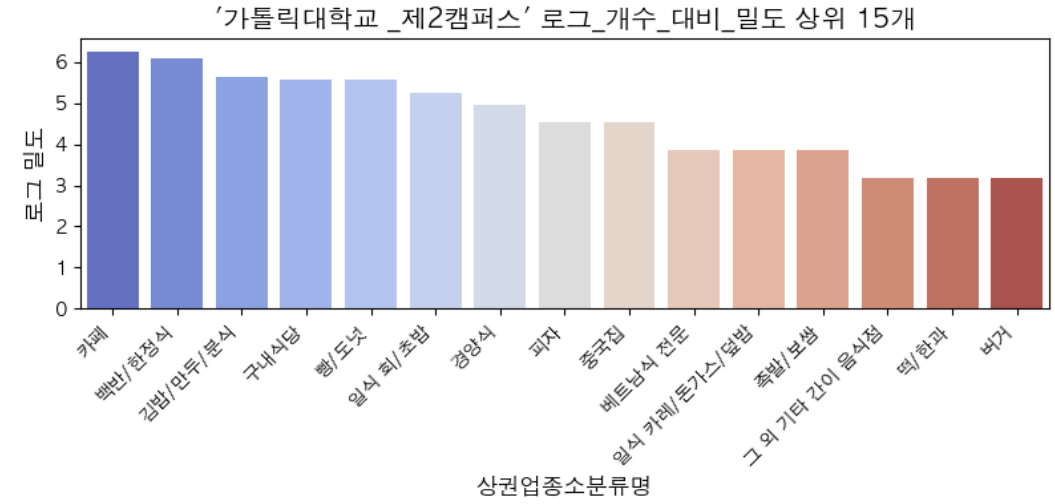
```
import matplotlib.pyplot as plt
import seaborn as sns

# 상권업종소분류명을 하나씩 순회하며 처리
for category in df['상권업종소분류명'].unique():
    # 특정 상권업종소분류명을 가진 데이터 필터링
    filtered_data = df[df['상권업종소분류명'] == category]
    filtered_data = filtered_data[filtered_data['대학명'] != '가톨릭대학교 _제3캠퍼스']  # 특정 대학교 제외

    # 로그 밀도의 평균 계산
    mean_log_density = filtered_data['로그_개수_대비_밀도'].mean()

    # 평균보다 낮은 데이터 필터링
    below_mean = filtered_data[filtered_data['로그_개수_대비_밀도'] <= mean_log_density]

    # 하위 15개만 선택
    bottom_15_filtered_data = below_mean.nsmallest(15, '로그_개수_대비_밀도')

    # 데이터가 존재하면 플롯 생성
    if not bottom_15_filtered_data.empty:
        plt.figure(figsize=(8, 4))
        sns.barplot(
            data=bottom_15_filtered_data,
            x='대학명',
            y='로그_개수_대비_밀도',
            palette='coolwarm'
        )
        plt.title(f"'{category}' 평균보다 낮은 대학교들의 로그밀도 (하위 15개)", fontsize=14)
        plt.xlabel('대학교', fontsize=12)
        plt.ylabel('로그 밀도', fontsize=12)
        plt.xticks(rotation=45, ha='right')
        plt.tight_layout()
        plt.show()
```



- 대학별 로그밀도 높은 상위 15개의 소분류명

대학별 로그밀도 높은 15개의 소분류명

```
import matplotlib.pyplot as plt
import seaborn as sns

# 대학명을 하나씩 순회하며 처리
for university in df['대학명'].unique():
    # 특정 대학 데이터를 필터링
    filtered_data = df[df['대학명'] == university]

    # 상위 15개 선택
    top_15_filtered_data = filtered_data.nlargest(15, '로그_개수_대비_밀도')

    # 데이터가 존재하면 플롯 생성
    if not top_15_filtered_data.empty:
        plt.figure(figsize=(8, 4))
        sns.barplot(
            data=top_15_filtered_data,
            x='상권업종소분류명',
            y='로그_개수_대비_밀도',
            palette='coolwarm'
        )
        plt.title(f"'{university}' 로그_개수_대비_밀도 상위 15개", fontsize=14)
        plt.xlabel('상권업종소분류명', fontsize=12)
        plt.ylabel('로그 밀도', fontsize=12)
        plt.xticks(rotation=45, ha='right')
        plt.tight_layout()
        plt.show()
```



- 대학별 로그밀도 낮은 하위 15개의 소분류명

대학별 로그밀도 하위 15개 소분류명

```
import matplotlib.pyplot as plt
import seaborn as sns

# 대학명을 하나씩 순회하며 처리
for university in df['대학명'].unique():
    # 특정 대학 데이터를 필터링
    filtered_data = df[df['대학명'] == university]

    # 상위 15개 선택
    bottom_15_filtered_data = filtered_data.nsmallest(15, '로그_개수_대비_밀도')

    # 데이터가 존재하면 플롯 생성
    if not bottom_15_filtered_data.empty:
        plt.figure(figsize=(8, 4))
        sns.barplot(
            data=bottom_15_filtered_data,
            x='상권업종소분류명',
            y='로그_개수_대비_밀도',
            palette='coolwarm'
        )
        plt.title(f"'{university}' 로그_개수_대비_밀도 하위 15개", fontsize=14)
        plt.xlabel('상권업종소분류명', fontsize=12)
        plt.ylabel('로그 밀도', fontsize=12)
        plt.xticks(rotation=45, ha='right')
        plt.tight_layout()
        plt.show()
```