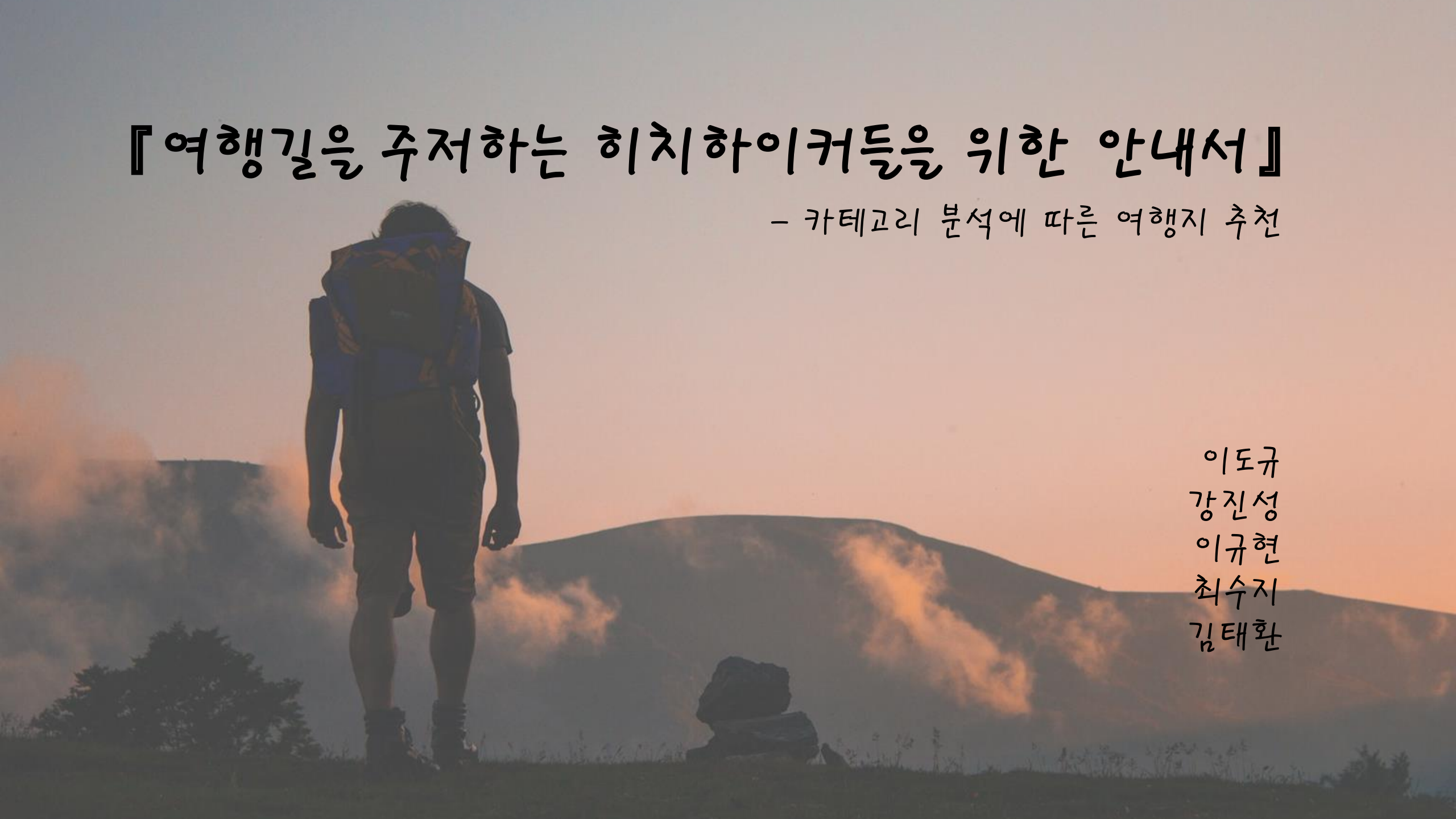


『여행길을 주저하는 히치하이커들을 위한 안내서』

- 카테고리 분석에 따른 여행지 추천

이도규
강진성
이규현
최수지
김태환



『개요』

여행 추천

여행 추천

해외여행 추천

신혼여행 추천

일본 여행 추천

국내여행 추천

가족여행 추천

베트남 여행 추천

겨울 여행 추천

도쿄 여행 추천

태교여행 추천

관심사를 반영한 컨텍스트 자동완성

자동완성 끄기 · 도움말 · 신고

- 일단 사람이 많은 곳 가면 본전 아닌가?
- 나는 한적한 곳이 좋은데?
- 여행은 쇼핑이 최고
- 경험이 여행의 핵심
- 더욱 땀 시원~한 바다지!
- 무슨 소리? 운동도 되는 등산이 최고지
- 겨울엔 따뜻한 실내에서 즐기는게 좋지
- 밖에 눈이 왔는데, 그게 무슨 소리냐?

『개요』

■ 사람마다 선호하는 카테고리가 다르다

Ex) 인도어 · 아웃도어, 바다 선호 · 산 선호, 화창한 날 · 비 내리는 날

■ 남녀의 취향 차이

Ex) 낚시, 등산, 운동, 쇼핑, 핫플레이스 관광

■ 구성원에 따른 관광, 여행지 선정

Ex) 어르신들 계시는데 등산? 남자는 나 혼자인데 낚시?



『 계획 』

- 분석 데이터를 대분류, 중분류, 소분류로 세분화
- 검색에 쓰일 카테고리, 결과에 쓰일 카테고리
- 어떤 카테고리로 나뉘야 사용자에게 만족할만한 결과를 내놓을까?
- 뭔가 독창성을 부여 할만한 방법은?
- 그밖에 등등



『 계획 』

▣ 프로젝트 타임라인

	목요일	금요일	토요일	일요일	월요일	화요일	수요일	목요일
계획과 수집								
데이터 전처리								
데이터 도출								
최종 정리								

한국관광 데이터랩

아름가이드북

[로그인](#) |
 [회원가입](#)

지역별 분석	국가별 분석	관광산업 분석	빅데이터	관광통계/실태조사	데이터랩 소개
------------------------	------------------------	-------------------------	----------------------	---------------------------	-------------------------

여행 떠나요!

Data & Tourism Vol.23

글로벌 소셜 데이터로 알아보는
주요 방한 20개국에서 인기 있는 K-O

주요 분석 메뉴

- 지역별 관광현황
- 문화관광축제 분석
- 중심-연관 관광지 지도

*로그인을 하시면 즐겨찾는 데이터를 보실 수 있습니다.

관광소비

방문객

소셜미디어

숙박체류

유입유출

공지사항 | 데이터 업데이트 주기

한국관광 데이터랩 신규 서비스 오픈 안내(지역별 관광/비관광 소...)

지역별 관광현황 소셜 데이터 변경 안내 (키워드 빠지)

한국관광 데이터랩 신규 서비스 오픈 안내(캠핑관광 현황, 걷기여...)

한국관광 데이터랩 '2023 외래객 수 맞추기 이벤트' 개최 알림

2023 한국관광 데이터랩 모의고사 및 활용 능력고사 전회차 참여...

방문객_방문객 네비분포.csv

방문객_방문자 거주지분포.csv

방문객_방문자 성별분포.csv

방문객_방문자 수추이.csv

소셜미디어_등 반유행 키워드.csv

유입유출_유입 유출.csv

인기관광지_내 국인관심관광지.csv

인기관광지_인 기관광검색.csv

인기관광지_인 기관광맛집검색.csv

인기관광지_인 기관광입장객.csv

『사전 분석』

【현재 가지고 나올 수 있는 여행지 추천 카테고리】

■ 동반별
-가족, 자녀
-친구
-연인,배우자
-싱글

■ 목적
-문화관광
-자연관광
-역사관광
-음식
-쇼핑
-체험
-레저스포츠

■ 장기체류 가치
무박
1박
2박

■ 인기 관광지
■ 연령별 *
■ 관심지점 비교(분석) *

+ 연간 기온, 강수량

* 전체에서 나눠서 점수로 만들면 좋겠는데.....

■ 소비자는 입력을 한다. 어떻게 할거야?

- 가족

[Result_동반유형키워드.csv 참조]의 분류> 관련키워드 내림차순

- 성별 분포

[Result_방문자성별분포.csv]를 참고하자

→ 데이트코스면 여성 쪽을 참고해야지? 남자는 뭘 할 수 있는데 너가

→ 낚시를 가고 싶어. 그럼 여성쪽도 방문이 많은 쪽을 골라야겠지? 안그러면 후일 장담못함.

- 목적

[Result_목적지검색.csv]

→ 만약 음식이 목적이면 [Result_인기관광맛집검색.csv]

→ 관광 들어가면 [Result_인기관광검색.csv]

- 차를 가지고 다니는 사람이 쓴다. + 체류기간이 길어도 상관없다

[Result_방문객네비분포.csv]

- 검색 관련

[Result_여행유형트렌드키워드.csv]의 분류 > 관련키워드 내림차순

[Result_여행유형트렌드언급량.csv]으로 많은 순으로

- 체류기간

[Result_지역별.csv] 참고 → 만약에 검색에 숙박이 들어간다 하면 참고내용 들어가게함

- 연평균 기온, 강수량

날씨 안 좋았다간 넌 아내한테 죽어 임마.

하물며 낚시 갔는데 안좋아? 수고하십쇼.

『분석』

■ input을 해보자

ex) 봄에 꽃 보러 가고 싶어요~ 추천해주세요~

→ 테마별에서 봄을 뽑아서 점수

→ 화원이나 공원이 들어가면 좋은 점수

▶ 맵에 마커 표시

→ 동반별로 카테고리 출력

→ 목적별로 카테고리 출력 + 음식이었다면 인기 맛집 추천하면 좋지 않을까?

→ 1박 이상 체류할 가치가 있는가? (통계 내 보자)

+ 만약 연인들끼리 갈 경우에는 최신 트렌드(2023)을 반영할 필요가 있지 않을까?

→ [가족과 봄에 갈만한 곳] 이라고 검색

결과를 어떻게 내줘야 할까?

→ 핵심 키워드는 '가족', '봄'

→ [Result_동반유형키워드.csv 참조]를 보면, 지역당 가족에 따른 언급 건 수가 있다.

→ [Result_동반유형키워드.csv 참조]에 따라 [Result_목적지검색.csv]를 참조하여 해당 지역에서 높은 목적지 유형을

+ 외지인이 높을수록 좋겠지?(Result_관심지점비교.csv 참고)

→ 위의 결과에 하단의 내용을 추가한다.

1. 마커 - 어딘지는지 알아야 될거 아니야

2. 각종 지수

- 체류: [Result_지역별.csv],[Result_연별 체류 방문자.csv]을 참고. 숙박까지 할 가치가 있는가 없는가를 확인 가능

- [Result_방문자성별분포.csv]로 남성 우세, 여성 우세

- 기온: 나들이하기 좋은 날씨인지 아닌지 알 수 있어용

【정리】

■ 검색에 사용할 파일

[Result_동반유형키워드.csv]

[Result_목적지검색.csv]

[Result_관심지점비교.csv]

[Result_방문자수추이.csv]

[Result_방문객네비분포.csv]

[Result_인기관광맛집검색.csv]

[Result_인기관광검색.csv]

[Result_인기관광입장객.csv]

[Result_여행유형트렌드언급량.csv]

[Result_여행유형트렌드키워드.csv]

■ 결과(마커 표시)에 사용할 파일

[Result_지역별.csv]

[Result_연별 체류 방문자.csv]

[Result_방문객네비분포.csv]

[Result_방문자성별분포.csv]

+ 날씨

■ 만들어야 될 코드(예상)

- 입력받을 코드(input)

- 입력받은 값에서 다음의 값을 포함하고 있나?(contain 쓰나?)

- 시각화(matplotlib, folium)

→ 방문자 성별 그래프

→ 연별 체류 기간

→ 날씨

『전처리』

```
import os
import re

import pandas as pd

# 파일 전처리 코드
# 카테고리에 따라 Work/ 설정
csv_root_path = 'work/인기관광지'
csv_folder_path = ''
csv_folder_list = os.listdir(csv_root_path)

data_dict = dict()

# CSV 파일에서 파일 경로 가져오기
for csv_folder_name in csv_folder_list:
    folder_path = os.path.join(csv_root_path, csv_folder_name)

    folder_year = csv_folder_name.split('-')[0][-6:-2]
    folder_region = csv_folder_name.split('-')[0][:-6]

    file_list = os.listdir(folder_path)

    # 경로 딕셔너리에 리스트로 넣어주기
    temp_data_list = []
    for file in file_list:
        temp_name = file.split('_')[1].split('.')[0]

        temp_data_list.append(os.path.join(folder_path, file))

    temp_key = folder_year + folder_region

    data_dict[temp_key] = temp_data_list

    temp_data_list
```

```
data_frame_dict = {}
# 데이터 딕셔너리 넣어주기
for key, values in data_dict.items():
    csv_list = {}

    for i, item in enumerate(values):

        make_key = ''
        if len(item.split('_')[1:]) == 1:
            make_key = item.split('_')[1:][0]
            make_key = make_key.split('.')[0]
        elif len(item.split('_')[1:]) == 2:
            make_key = item.split('_')[1:][0] + item.split('_')[1:][1]
            make_key = make_key.split('.')[0]
        elif len(item.split('_')[1:]) == 3:
            make_key = item.split('_')[1:][0] + item.split('_')[1:][2]
            make_key = make_key.split('.')[0]
        else:
            make_key = item.split('_')[1:][0]
            make_key = make_key.split('.')[0]

        if key == '2020광주광역시':
            print(i, " : ", make_key.strip())
            csv_list[make_key.strip()] = pd.read_csv(item, encoding='euc-kr')

    data_frame_dict[key] = csv_list

# 저장된 형태
#data_frame_dict['2020강원특별자치도']['유입 목적지 유형 분포']
```

『전처리』

```
import re

work_list = ['영어권', '일본어권', '중간체권', '중번체권']

year_filter = {}

for work_item in work_list:

    for key, values in data_frame_dict.items():
        work_file_name = key + '_' + work_item + '.csv'
        try:
            year_filter[key] = pd.read_csv(work_file_name, encoding='euckr')
        except:
            continue

        # 파일 년도 / 지역 나누기
        numbers = re.findall('\d+', key)
        message = re.findall('\D+', key)

        # 컬럼 추가하기
        year_filter[key].insert(len(year_filter[key].columns), '파일년도', numbers[0])
        year_filter[key].insert(len(year_filter[key].columns), '파일지역', message[0])

# 저장된 데이터 전부 합치기
result_data = pd.concat(year_filter)

year_filter.clear()

result_data = result_data.reset_index()
result_data = result_data.drop(columns=['level_0', 'level_1'])
# 결과를 도출
result_data.to_csv('Result_{}.csv'.format(work_item), encoding='euckr')
```

```
# 사용 방법
# 인기 관광지로 수정후 세팅
for key, values in data_frame_dict.items():

    df01 = values[5]
    df02 = values[6]
    df03 = values[7]

    df_merge = pd.merge(df01, df02, on=['관광지명', '주소', '분류'], how='outer')
    df_merge = pd.merge(df_merge, df03, on=['관광지명', '주소', '분류'], how='outer')
    df_merge.columns = ['순위_외지인', '관광지명', '주소', '분류', '외지인 검색 수', '순위_전체', '전체 검색수', '순위_내지인', '현지인 검색수']
    df_merge = df_merge[['순위_외지인', '순위_내지인', '순위_전체', '관광지명', '주소', '분류', '외지인 검색 수', '현지인 검색수', '전체 검색수']]

    df_merge.to_csv(key + '_' + '인기관광지검색순위.csv', encoding='euckr')
```

```
for key, values in data_frame_dict.items():

    # 관광소비 내국인, 관광소비_외국인 합치기
    # 파일 불러오기

    df0 = values[2].copy()
    df1 = values[3].copy()

    # 공통 그룹 만들어 정렬해주기
    df_temp_00 = df0.sort_values(['업종대분류명', '업종중분류명'], ascending=True)
    df_temp_01 = df1.sort_values(['업종대분류명', '업종중분류명'], ascending=True)
    # 불필요한 부분 지워주기
    df_temp_00.drop(['업종대분류명', '업종중분류명'], axis=1, inplace=True)
    # 파일 합치기
    df2 = pd.concat([df_temp_01, df_temp_00], axis=1)
    # 컬럼 명 바꾸기
    df2.columns = ['업종대분류명', '업종중분류명', '외국인 소비액', '내국인 소비액']
    # 컬럼 순서 바꾸기
    df3 = df2[['업종대분류명', '업종중분류명', '내국인 소비액', '외국인 소비액']]
    # 파일 저장하기
    df3.to_csv((key + '_관광소비.csv'), encoding='euckr')
```

『전처리』

```
for key, values in data_frame_dict.items():
    # 관광소비 주이(내국인, 외국인 합치기)
    # 파일 불러오기
    df = values[0].copy()
    df1 = values[1].copy()
    # 전체값 row 삭제해주기
    df_temp_00 = df.drop([0,1,2,3,4,5,6,7,8,9,10,11],axis=0)
    df_temp_01 = df1.drop([0,1,2,3,4,5,6,7,8,9,10,11],axis=0)
    # 공통 그룹 만들어 정렬해주기
    df_temp_02 = df_temp_00.sort_values(['기준연월', '업종대분류명'], ascending=True)
    df_temp_03 = df_temp_01.sort_values(['기준연월', '업종대분류명'], ascending=True)
    # 불필요한 부분(= 공통컬럼) 지워주기
    df_temp_04 = df_temp_03.drop(['기준연월', '업종대분류명'],axis=1)
    # 파일 합치기
    df2 = pd.concat([df_temp_02,df_temp_04],axis=1)
    # 컬럼 명 바꾸기
    df2.columns = ['기준연월', '업종대분류명', '내국인 소비액', '외국인 소비액']
    # 파일 저장하기
    df2.to_csv(key + '_관광소비주이.csv',encoding='euckr')
```

```
for key, values in data_frame_dict.items():
    df01 = values[8]
    df02 = values[9]
    df03 = values[10]

    df01.columns = ['순위', '광역지자체명', '기초지자체명', '관광지명', '기준연월', '입장객수', '내국인연합산']
    df02.columns = ['순위', '광역지자체명', '기초지자체명', '관광지명', '기준연월', '입장객수', '외국인연합산']
    df03.columns = ['순위', '광역지자체명', '기초지자체명', '관광지명', '기준연월', '입장객수', '전체연합산']

    df01['순위'].astype(int)
    df02['순위'].astype(int)
    df03['순위'].astype(int)

    df_merge = pd.merge(df01,df02,on=['광역지자체명', '기초지자체명', '관광지명', '기준연월'],how='outer')
    df_merge['순위_y'] = df_merge['순위_y'].fillna(0)
    df_merge['순위_y'] = df_merge['순위_y'].astype(int)
    df_merge = pd.merge(df_merge,df03,on=['광역지자체명', '기초지자체명', '관광지명', '기준연월'],how='outer')
    df_merge.columns = ['순위_내국인', '광역지자체명', '기초지자체명', '관광지명', '기준연월', '입장객수_내국인', '내국인연합산', '순위_외국인', '입장객수_외국인', '외국인연합산']
    df_merge = df_merge[['광역지자체명', '기초지자체명', '관광지명', '기준연월', '순위_내국인', '입장객수_내국인', '내국인연합산', '순위_외국인', '입장객수_외국인', '외국인연합산']]
    df_merge
```

숙박 체류시간 영역

```
#
for key, values in data_frame_dict.items():

    data1 = values[0]
    data2 = values[1]
    data3 = values[2]
    data4 = values[3]
    data5 = values[4]
    data6 = values[5]
    data7 = values[6]
    data8 = values[7]

    data_01 = data1.set_index('지역명')
    data_temp_6 = data6.rename(columns = {'숙박방문자 비율' : '숙박방문자 비율 주이'})

    # 2,4,5
    data_merge = pd.merge(data2,data4,on='기준연월',how='outer')
    data_merge = pd.merge(data_merge,data5,on='기준연월',how='outer')
    data_merge

    # 6,7,8
    data_merge01 = pd.merge(data6,data7,on='기준연월',how='outer')
    data_merge01 = pd.merge(data_merge01,data8,on=['기준연월', '지역명'],how='outer')
    data_merge01

    # 1 3
    data03 = data3.rename(columns = {'광역지자체명':'지역명'})
    data_merge02 = pd.merge(data1,data03,how='outer')

    #data_merge01 # 연별 체류 방문자
    #data_merge02 # 지역별 방문자

    data_merge01.to_csv("{}_연별 체류 방문자.csv".format(key),encoding='euckr')
    data_merge02.to_csv("{}_지역별.csv".format(key),encoding='euckr')
```

『전처리』

유입 유출

```
#
for key, values in data_frame_dict.items():

    # 유입 목적지 유형 분포
    df = values['유입 목적지 유형 분포']
    # 유출 목적지 유형 분포
    df1 = values['유출 목적지 유형 분포']
    # 유입 방문 목적지
    df2 = values['주 유입 방문 목적지']
    # 유출 방문 목적지
    df3 = values['주 유출 방문 목적지']

    df.columns = ['유형 중분류명', '유형 소분류명', '소분류 검색건수']
    df1.columns = ['유형 중분류명', '유형 소분류명', '소분류 검색건수']

    df00 = pd.merge(df, df1, on = ['유형 중분류명', '유형 소분류명'], how = 'outer')
    df01 = pd.merge(df2, df3, on = ['업종중분류명', '방문자수', '순위'], how = 'outer')

    df00.rename(columns={'소분류 검색건수_x': '유입 소분류 검색건수', '소분류 검색건수_y': '유출 소분류 검색건수', '유형 중분류명': '업종중분류명'}, inplace = True)
    df123 = pd.merge(df00, df01, on='업종중분류명', how='outer')
    df123.to_csv("{}_유입유출.csv".format(key), encoding='euckr')
```

```
# 관광소비 내국인, 관광소비 외국인 합치기
# 파일 불러오기
df = pd.read_csv(r'D:\SourceTree\DataProject\501\CSV\work\관광소비\강원특별자치도202101_202112데이터업다운로드\20231216183912_관광소비_내국인.csv', encoding='cp949')
df1 = pd.read_csv(r'D:\SourceTree\DataProject\501\CSV\work\관광소비\강원특별자치도202101_202112데이터업다운로드\20231216183912_관광소비_외국인.csv', encoding='cp949')
# 공통 그룹 만들어 정렬해주기
df.sort_values(['업종대분류명', '업종중분류명'], ascending=True, inplace=True)
df1.sort_values(['업종대분류명', '업종중분류명'], ascending=True, inplace=True)
# 불필요한 부분 지워주기
df.drop(['업종대분류명', '업종중분류명'], axis=1, inplace=True)
# 파일 합치기
df2 = pd.concat([df1, df], axis=1)
# 컬럼 명 바꾸기
df2.columns = ['업종대분류명', '업종중분류명', '외국인 소비액', '내국인 소비액']
# 컬럼 순서 바꾸기
df3 = df2[['업종대분류명', '업종중분류명', '내국인 소비액', '외국인 소비액']]
# 파일 저장하기
df3.to_csv(r'temp.csv', encoding='cp949')
df3
```

```
# 관광소비 주이(내국인, 외국인 합치기)
# 파일 불러오기
df = pd.read_csv(r'C:\Users\GJ\Downloads\관광소비\20231216183912_관광소비 주이_내국인.csv', encoding='cp949')
df1 = pd.read_csv(r'C:\Users\GJ\Downloads\관광소비\20231216183912_관광소비 주이_외국인.csv', encoding='cp949')
# 전체값 row 삭제해주기
df.drop([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11], axis=0, inplace=True)
df1.drop([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11], axis=0, inplace=True)
# 공통 그룹 만들어 정렬해주기
df.sort_values(['기준연월', '업종대분류명'], ascending=True, inplace=True)
df1.sort_values(['기준연월', '업종대분류명'], ascending=True, inplace=True)
# 불필요한 부분 (= 공통컬럼) 지워주기
df1.drop(['기준연월', '업종대분류명'], axis=1, inplace=True)
# 파일 합치기
df2 = pd.concat([df, df1], axis=1)
# 컬럼 명 바꾸기
df2.columns = ['기준연월', '업종대분류명', '내국인 소비액', '외국인 소비액']
# 파일 저장하기
df2.to_csv(r'C:\Users\rkgml\Desktop\2_충남_숙박_채류시설\new\1.csv', encoding='cp949')
```

```
for key, values in data_frame_dict.items():
```

```
    try:
        data1 = values['인기관광지전체']#pd.read_csv(r'인기관광지_전체.csv', encoding='euckr')
        data2 = values['인기관광지외지인']#pd.read_csv(r'인기관광지_외지인.csv', encoding='euckr')
        data3 = values['인기관광지현지인']#pd.read_csv(r'인기관광지_현지인.csv', encoding='euckr')
    except:
        print("인기관광 : " + key)
```

```
data4 = None
data5 = None
data6 = None
```

```
    try:
        data7 = values['지역전체']#pd.read_csv(r'지역_맛집_전체.csv', encoding='euckr')
        data8 = values['지역외지인']#pd.read_csv(r'지역_맛집_외지인.csv', encoding='euckr')
        data9 = values['지역맛집현지인']#pd.read_csv(r'지역맛집_현지인.csv', encoding='euckr')
    except:
        print("맛집 : " + key)
```

```
data01 = data1.rename(columns={'순위' : '전체 순위'})
data01 = data01.rename(columns={'Unnamed: 4' : '전체 검색수'})
```

```
data01 = data01[['관광지명', '주소', '분류', '전체 순위', '전체 검색수']]
```

```
data02 = data2.rename(columns={'순위' : '외지인 순위'})
data03 = data3.rename(columns={'순위' : '현지인 순위'})
```

```
data_merge01 = pd.merge(data01, data03, on=['관광지명', '주소', '분류'])
data_merge01 = pd.merge(data_merge01, data02, on=['관광지명', '주소', '분류'])
```

```
data07 = data7.rename(columns={'순위' : '전체 순위'})
data08 = data8.rename(columns={'Unnamed: 4' : '전체 검색수'})
```

```
data07 = data07[['업소명', '주소', '분류', '전체 순위']]
```

```
data08 = data8.rename(columns={'순위' : '외지인 순위'})
data09 = data9.rename(columns={'순위' : '현지인 순위'})
```

```
data_merge03 = pd.merge(data07, data09, on=['업소명', '주소', '분류'])
data_merge03 = pd.merge(data_merge03, data08, on=['업소명', '주소', '분류'])
```

```
data_merge01.to_csv("{}_인기관광검색.csv".format(key), encoding='euckr')
data_merge03.to_csv("{}_인기관광맛집검색.csv".format(key), encoding='euckr')
```

『전처리』

```
data1 = pd.read_csv(r'D:\SourceTree\DataProject\S01\CSV\work\유사지역\강원특별자치도202001-202012데이터랩다운로드\20231216132607_강원특별자치도 네비게이션 검색 유형.csv', encoding = 'cp949')
data2 = pd.read_csv(r'D:\SourceTree\DataProject\S01\CSV\work\유사지역\강원특별자치도202001-202012데이터랩다운로드\20231216132607_강원특별자치도 방문자 구성.csv', encoding = 'cp949')
data3 = pd.read_csv(r'D:\SourceTree\DataProject\S01\CSV\work\유사지역\강원특별자치도202001-202012데이터랩다운로드\20231216132607_경상남도 방문자 구성.csv', encoding = 'cp949')
data4 = pd.read_csv(r'D:\SourceTree\DataProject\S01\CSV\work\유사지역\강원특별자치도202001-202012데이터랩다운로드\20231216132607_경상북도 방문자 구성.csv', encoding = 'cp949')
data5 = pd.read_csv(r'D:\SourceTree\DataProject\S01\CSV\work\유사지역\강원특별자치도202001-202012데이터랩다운로드\20231216132607_전라남도 네비게이션 검색 유형.csv', encoding = 'cp949')
data6 = pd.read_csv(r'D:\SourceTree\DataProject\S01\CSV\work\유사지역\강원특별자치도202001-202012데이터랩다운로드\20231216132607_제주특별자치도 네비게이션 검색 유형.csv', encoding = 'cp949')
data7 = pd.read_csv(r'D:\SourceTree\DataProject\S01\CSV\work\유사지역\강원특별자치도202001-202012데이터랩다운로드\20231216132607_지역관광진단(유사지역).csv', encoding = 'cp949')
data8 = pd.read_csv(r'D:\SourceTree\DataProject\S01\CSV\work\유사지역\강원특별자치도202001-202012데이터랩다운로드\20231216132607_충청남도 네비게이션 검색 유형.csv', encoding = 'cp949')
data9 = pd.read_csv(r'D:\SourceTree\DataProject\S01\CSV\work\유사지역\강원특별자치도202001-202012데이터랩다운로드\20231216132607_충청북도 방문자 구성.csv', encoding = 'cp949')
```

```
data1_1 = data1.groupby(['지역명', '업종중분류명']).sum()
```

```
data2_1 = data2.groupby(['지역명', '성/연령']).sum()
```

```
data3_1 = data3.groupby(['지역명', '성/연령']).sum()
```

```
data4_1 = data4.groupby(['지역명', '성/연령']).sum()
```

```
# 전라남도 네비게이션 검색 유형
```

```
data5_1 = data5.groupby(['지역명', '업종중분류명']).sum()
```

```
data6_1 = data6.groupby(['지역명', '업종중분류명']).sum()
```

```
data7_1 = data7.groupby(['지역명', '지표 코드']).sum()
```

```
data8_1 = data8.groupby(['지역명', '업종중분류명']).sum()
```

```
data9_1 = data9.groupby(['지역명', '성/연령']).sum()
```

```
data1_5_6_8 = pd.concat([data1_1, data5_1, data6_1, data8_1], axis = 0)
```

```
data1_5_6_8
```

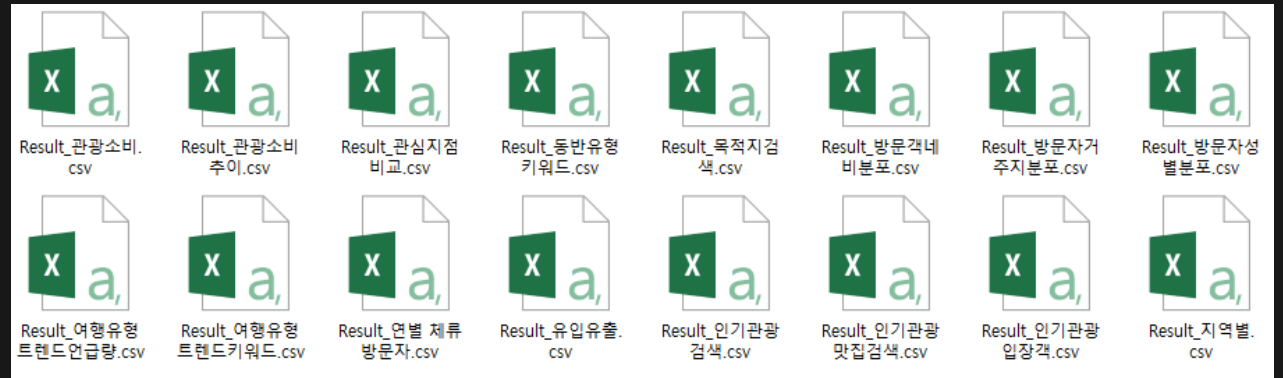
```
data2_3_4_9 = pd.concat([data2_1, data3_1, data4_1, data9_1], axis = 0)
```

```
data2_3_4_9
```

```
data1_5_6_8.to_csv('지역별 네비게이션 검색 유형_합본.csv')
```

```
data2_3_4_9.to_csv('지역별 방문자 구성_합본.csv')
```

```
data7_1.to_csv('지역관광진단(유사지역)_합본.csv')
```



『분석 & 도출』

```
< import pandas as pd
● import numpy as np

import matplotlib as mat
import folium as fl
```

✓ 1.2s

```
data1 = pd.read_csv(r'C:\Users\GJ\Desktop\전처리\카테고리\분류')
data2 = pd.read_csv(r'C:\Users\GJ\Desktop\전처리\카테고리\여행')
data3 = pd.read_csv(r'C:\Users\GJ\Desktop\전처리\카테고리\인기')
data4 = pd.read_csv(r'C:\Users\GJ\Desktop\전처리\카테고리\지역')
data5 = pd.read_csv(r'C:\Users\GJ\Desktop\전처리\카테고리\체류')
data6 = pd.read_csv(r'C:\Users\GJ\Desktop\전처리\카테고리\키워드')
```

```
import folium
from folium import Marker
```

```
inputData = '맛집투어'
result = ''
for item in data1['분류'].values:
    confirm = data1['키워드'][data1['분류'] == item].apply(lambda x : inputData in x)
    if confirm.bool() == True:
        result = item
        break
maxTemp = data6[(data6['유형 중분류명'] == result)][['중분류 검색건 수']].max()
```

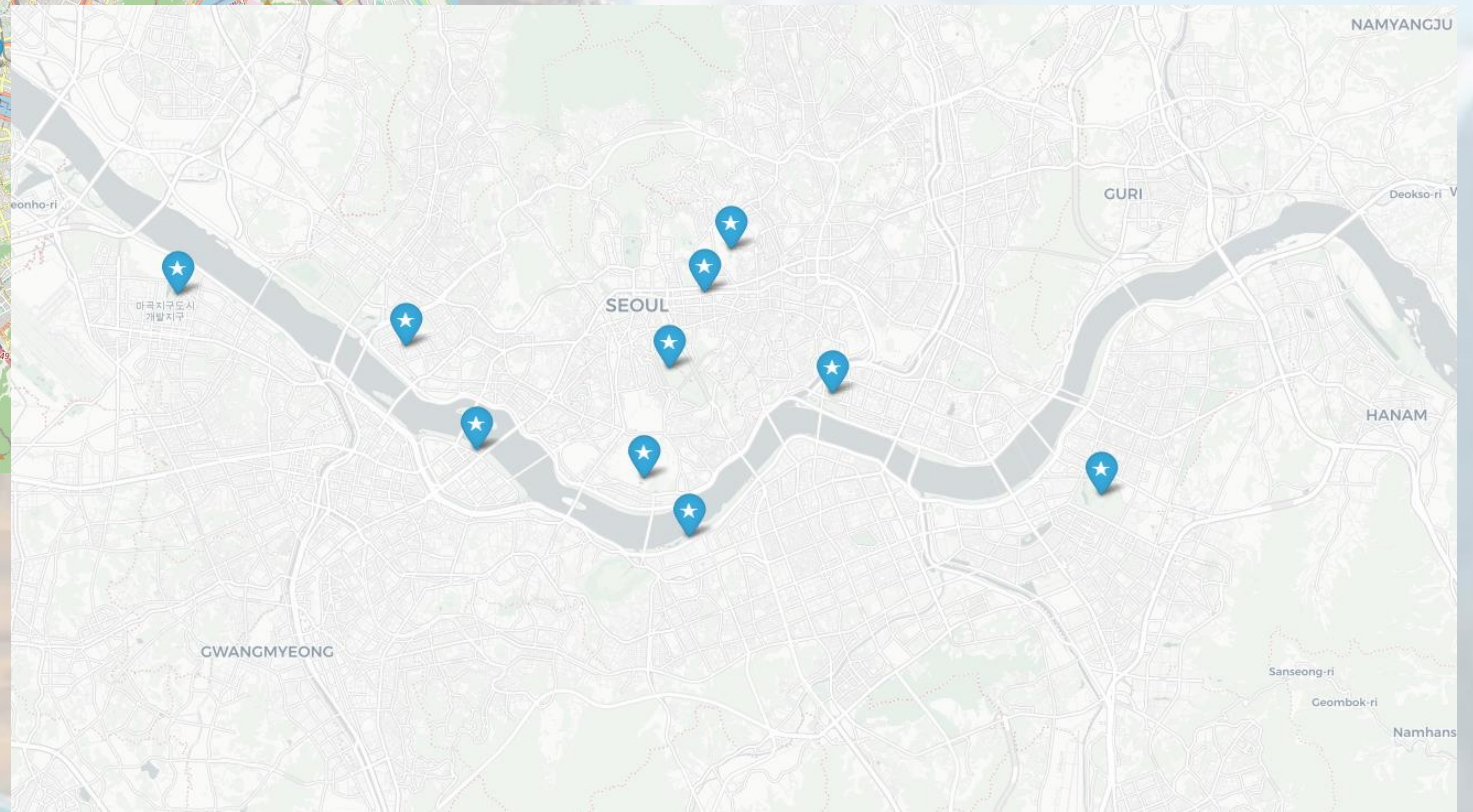
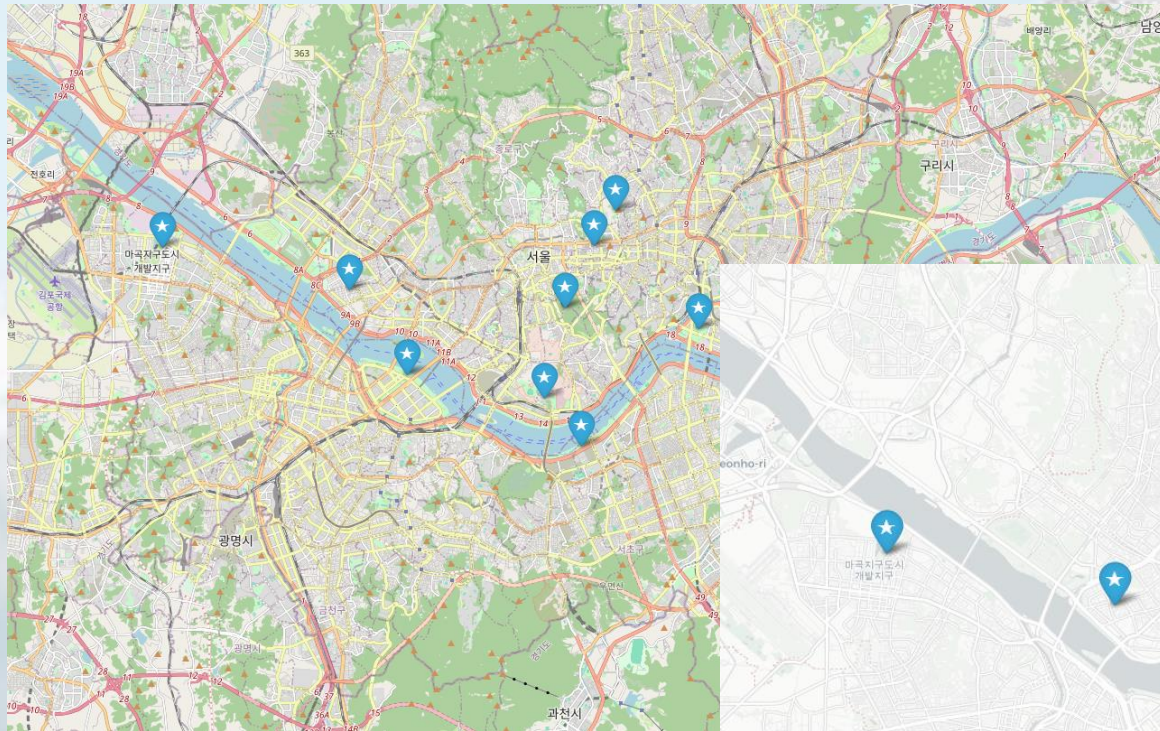
```
tempData = data6[(data6['유형 중분류명'] == result)]
result_data = tempData[tempData['중분류 검색건 수'].max() == tempData['중분류 검색건 수']][['파일지역']].iloc[0]
```

```
temp_data = data2[(data2['도'] == '서울특별시') & (data2['목적'] == result)]
center = [37.541, 126.986]
tour_map = folium.Map(location=center, zoom_start=10)
```

```
# 타일 스타일
# tiles = 'CartoDB positron'
# fl.TileLayer(tiles=tiles).add_to(tour_map)
```

```
for i in range(len(temp_data)):
    name = temp_data.iloc[i]['관광지명']
    x_pos = temp_data.iloc[i]['위도']
    y_pos = temp_data.iloc[i]['경도']
    Marker(location = [x_pos, y_pos],
            popup=name,
            icon=folium.Icon(color='blue',icon='star')
    ).add_to(tour_map)
tour_map
```

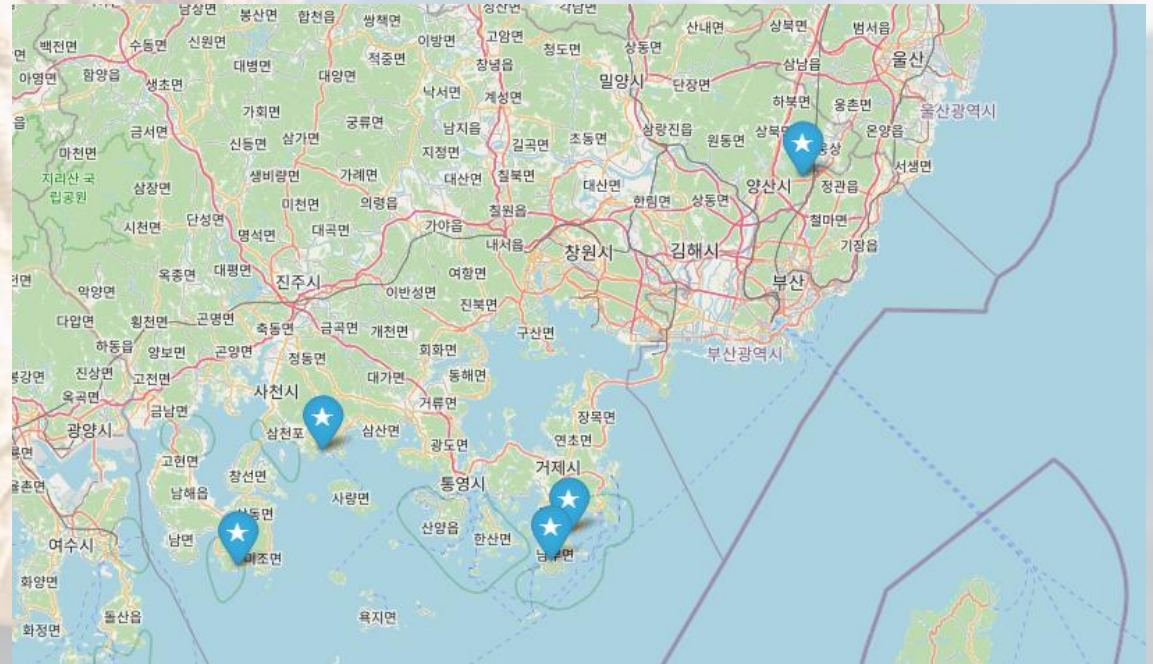
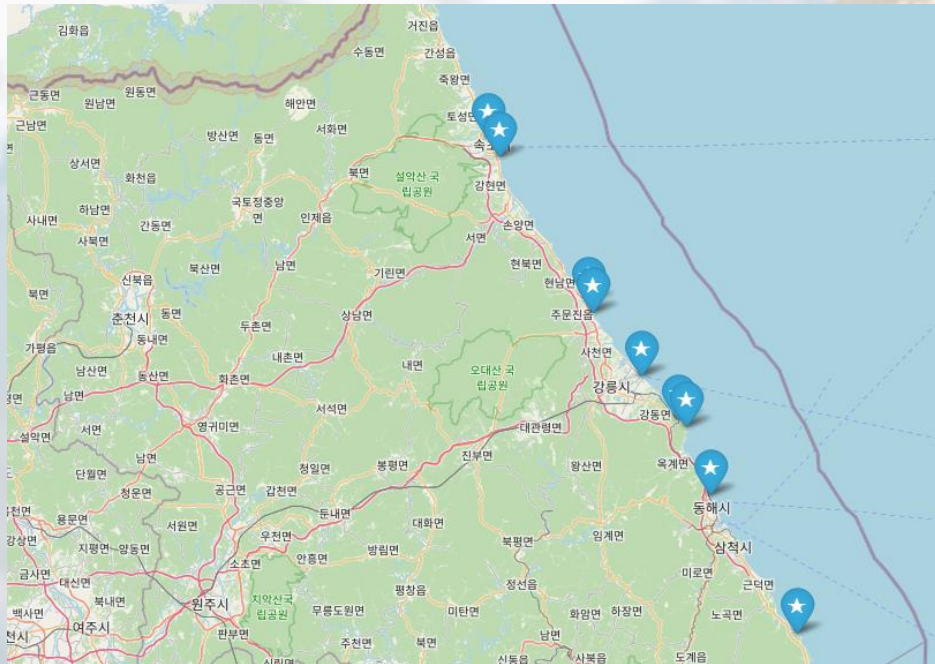
『분석 & 도출』



『분석 & 도출』

Unnamed: 0.1	Unnamed: 0	관령지명	경도	위도	도	주소	목적	KTO 대	KTO 중	KTO 소
1	1	연옥해변	128.9473	37.7731	강원특별자치도	강원 강릉시 창해로14번길 20-1	레저스포츠	자연	자연관광지	해수욕장
2	2	주문진항	128.8299	37.8920	강원특별자치도	강원 강릉시 해안로 1758-14	레저스포츠	자연	자연관광지	항구/포구
5	5	장호항	129.3174	37.2880	강원특별자치도	강원 삼척시 장호항길 103-0	레저스포츠	자연	자연관광지	항구/포구
6	6	목호항	129.1124	37.5498	강원특별자치도	강원 동해시	레저스포츠	자연	자연관광지	항구/포구
7	7	외옹치해변	128.6090	38.1832	강원특별자치도	강원 속초시 해오름로 85-3	레저스포츠	자연	자연관광지	해수욕장
8	8	정동삼곡배다부처길	129.0536	37.6766	강원특별자치도	강원 강릉시	레저스포츠	레크	육상 레포츠	트레킹
10	10	주문진해변	128.8192	37.9110	강원특별자치도	강원 강릉시 주문북로 0-0	레저스포츠	자연	자연관광지	해수욕장
13	13	영랑호	128.5812	38.2184	강원특별자치도	강원 속초시	레저스포츠	자연	자연관광지	호수
15	15	정동진해변	129.0345	37.6908	강원특별자치도	강원 강릉시 정동역길 17-0	레저스포츠	자연	자연관광지	해수욕장

	Unnamed: 0.1	Unnamed: 0	관광지명	경도	위도	도	주소	목적	KTO 대	KTO 중	KTO 소
41	41	41	학동흑진주등돌 해변	128.6414	34.7747	경상남도	경남 거제시 학동6길 18-1	레저스포츠	자연	자연관광지	해수욕장
42	42	42	상주은모래비치	127.9880	34.7207	경상남도	경남 남해군 상주로 17-0	레저스포츠	자연	자연관광지	해수욕장
46	46	46	삼족암군립공원	128.1542	34.9086	경상남도	경남 고성군	레저스포츠	자연	자연관광지	군립공원
51	51	51	법기수원지	129.1044	35.3512	경상남도	경남 양산시 법기로 198-13	레저스포츠	자연	자연관광지	자연휴양림
54	54	54	저구항	128.6069	34.7307	경상남도	경남 거제시 저구해안길 59-0	레저스포츠	자연	자연관광지	항구/포구



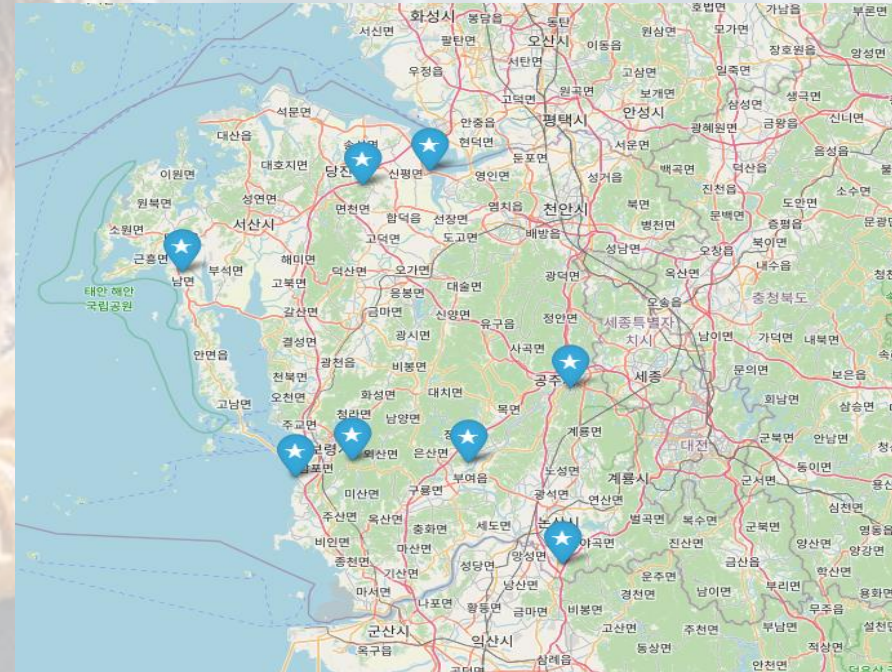
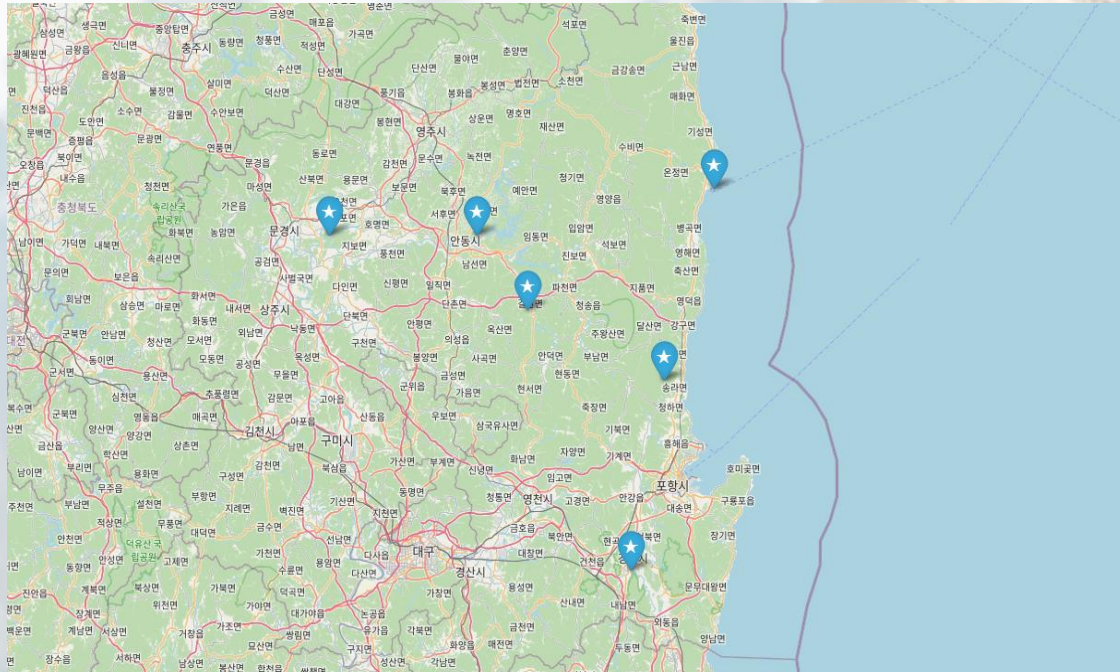
『분석 & 도출』

Unnamed: 0.1	Unnamed: 0	관광지명	경도	위도	도	주소	목적	KTO대	KTO중	KTO소
66	66	월영교	128.7610	36.5766	경상북도	경북 안동시	기타관광	인문(문화/예술/역사)	건축/조형물	다리/대교
72	72	등기산스카이워크	129.4648	36.6802	경상북도	경북 울진군	기타관광	인문(문화/예술/역사)	건축/조형물	유명건물
73	73	월정교	129.2181	35.8292	경상북도	경북 경주시	기타관광	인문(문화/예술/역사)	역사관광지	유적지/사적지
79	79	만휴정	128.9098	36.4104	경상북도	경북 안동시 목계하리길 42-0	기타관광	인문(문화/예술/역사)	역사관광지	유적지/사적지
81	81	보경사	129.3179	36.2522	경상북도	경북 포항시 북구 보경로 523-0	기타관광	인문(문화/예술/역사)	역사관광지	사찰
83	83	회룡포	128.3209	36.5764	경상북도	경북 예천군	기타관광	자연	자연관광지	자연생태관광지

도 : 충청남도 # 목적 : 쇼핑

0.0s

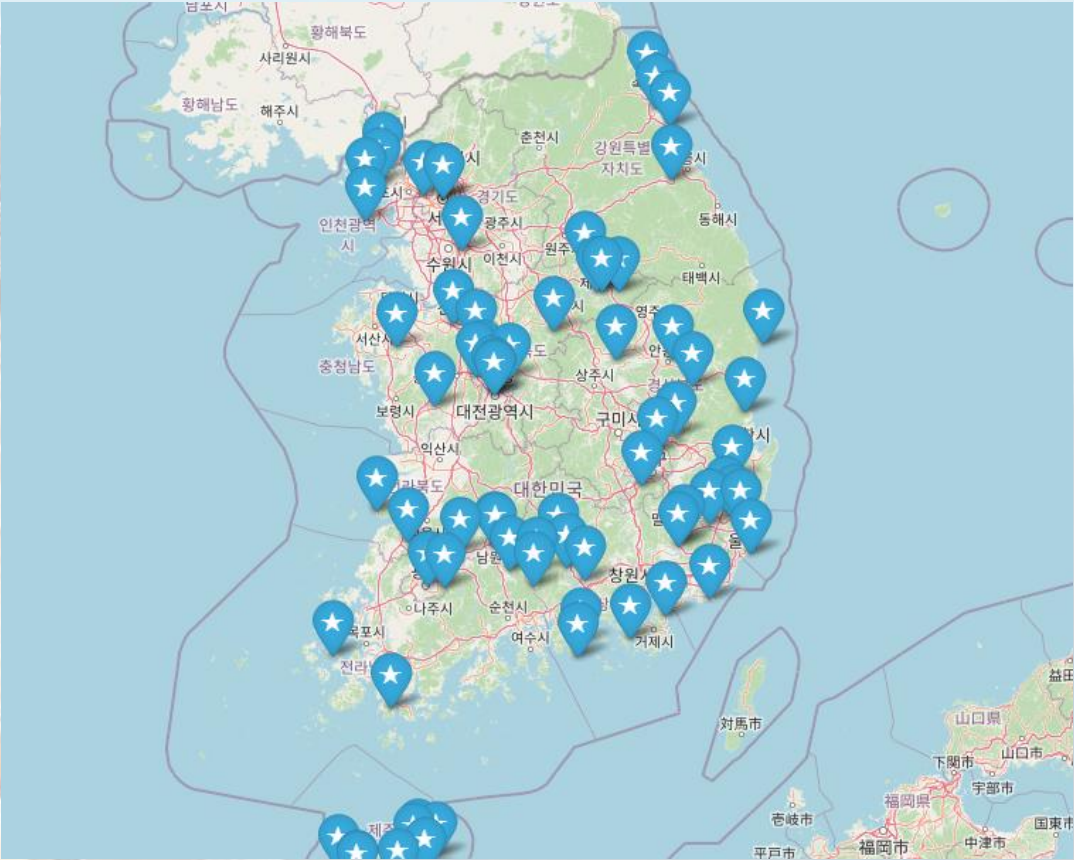
Unnamed: 0.1	Unnamed: 0	관광지명	경도	위도	도	주소	목적	KTO대	KTO중	KTO소
356	356	백제문화단지	126.9066	36.3066	충청남도	충남 부여군 백제문화단지	문화/예술/역사	인문(문화/예술/역사)	유아관광지	관광단지
360	360	상화원	126.5380	36.2770	충청남도	충남 보령시 남포방조제로 408-52	문화/예술/역사	인문(문화/예술/역사)	유아관광지	공원
361	361	아미미술관	126.6803	36.8617	충청남도	충남 당진시 남부로 753-4	문화/예술/역사	인문(문화/예술/역사)	문화시설	미술관/화랑
364	364	개화예술공원	126.6574	36.3117	충청남도	충남 보령시 성주산로 673-47	문화/예술/역사	인문(문화/예술/역사)	유아관광지	공원
367	367	공주산성시장	127.1236	36.4575	충청남도	충남 공주시 용당길 20-0	문화/예술/역사	인문(문화/예술/역사)	쇼핑	쇼핑
368	368	청산수목원	126.2971	36.6872	충청남도	충남 태안군 연꽃길 70-0	자연	자연	자연관광지	수목원
373	373	삼교호놀이동산	126.8223	36.8901	충청남도	충남 당진시 삼교천3길 15-0	문화/예술/역사	인문(문화/예술/역사)	유아관광지	테마공원
374	374	선사인랜드	127.1066	36.1024	충청남도	충남 논산시 봉황로 102-0	문화/예술/역사	인문(문화/예술/역사)	산업관광지	기타



『분석 & 도출』

Unnamed: 0.1	Unnamed: 0	관광지명	경도	위도	도	주소	목적	KTO대	KTO중	KTO소	
0	0	0	낙산사	128.6274	38.1239	강원특별 자치도	강원 양양군 낙산사로 100-0	기타 관광	인문(문화/예 술/역사)	역사관광 지	사찰
4	4	4	대관령양떼 목장	128.7527	37.6883	강원특별 자치도	강원 평창군 대관령마루길 483-32	기타 관광	인문(문화/예 술/역사)	체험관광 지	농.산.어촌 체험
14	14	14	하조대	128.7352	38.0212	강원특별 자치도	강원 양양군 조준길 99-0	기타 관광	자연	자연관광 지	해안절경
16	16	16	청간정	128.5584	38.2643	강원특별 자치도	강원 고성군 동해대로 5110-0	기타 관광	인문(문화/예 술/역사)	역사관광 지	유적지/사 적지
30	30	30	행주산성	126.8281	37.5961	경기도	경기 고양시 덕양구 행주 로15번길 0-0	기타 관광	인문(문화/예 술/역사)	역사관광 지	성
...
378	378	378	청풍호반케 이블카	128.1666	37.0013	충청북도	충북 제천시 문화재길 166-0	기타 관광	인문(문화/예 술/역사)	산업관광 지	기타
381	381	381	청남대	127.4931	36.4633	충청북도	충북 청주시 상당구 청남 대길 646-0	기타 관광	인문(문화/예 술/역사)	건축/조 형물	유명건물
382	382	382	산막이옛길	127.8361	36.7559	충청북도	충북 괴산군	기타 관광	레포츠	육상 레 포츠	트레킹
386	386	386	배론성지	128.0824	37.1607	충청북도	충북 제천시 배론성지길 296-0	기타 관광	인문(문화/예 술/역사)	역사관광 지	종교성지
387	387	387	정방사	128.2074	36.9990	충청북도	충북 제천시 옥순봉로12 길 165-0	기타 관광	인문(문화/예 술/역사)	역사관광 지	사찰
71 rows × 16 columns											

71 rows × 16 columns



『데이터 기대감』

데이터를 이용함으로써 계속해서 누적되는 데이터

- 보다 더 양질의 카테고리 세분화 가능
- 결과의 정확성 향상
- 새로운 경로의 카테고리 및 이용자 UI 확장 가능성
- 자체 피드백 가능
- 데이터 조사 비용 절감
- 상업적으로 이용 가능

『개선점 & 아쉬운 점』

■ 데이터 관련

- 카테고리 구분 및 잉여 데이터 수집에 시간 할애
- 목표의 구체화 필요를 체감
- 기술 향상의 필요성
- 카테고리 세분화 부족
- 기발한 아이디어

■ 프로젝트 관련

- 프로젝트 견적 측정 실패
- 시간 활용 측면

『참조』

한국관광공사 데이터랩

<https://datalab.visitkorea.or.kr/datalab/portal/main/getMainForm.do>

기상자료개방포털

<https://data.kma.go.kr/cmmn/main.do>

공공데이터포털

<https://www.data.go.kr/>

Thank you!