데이터 분석

- 사드 배치의 영향으로 중국인 관광객이 얼마나 줄었을까?
  - 다수 개의 파일 합하는 방법
  - 원하는 컬럼 추가
  - 반복 작업을 위한 함수 작성
  - 시계열 그래프, 히트맵 그래프

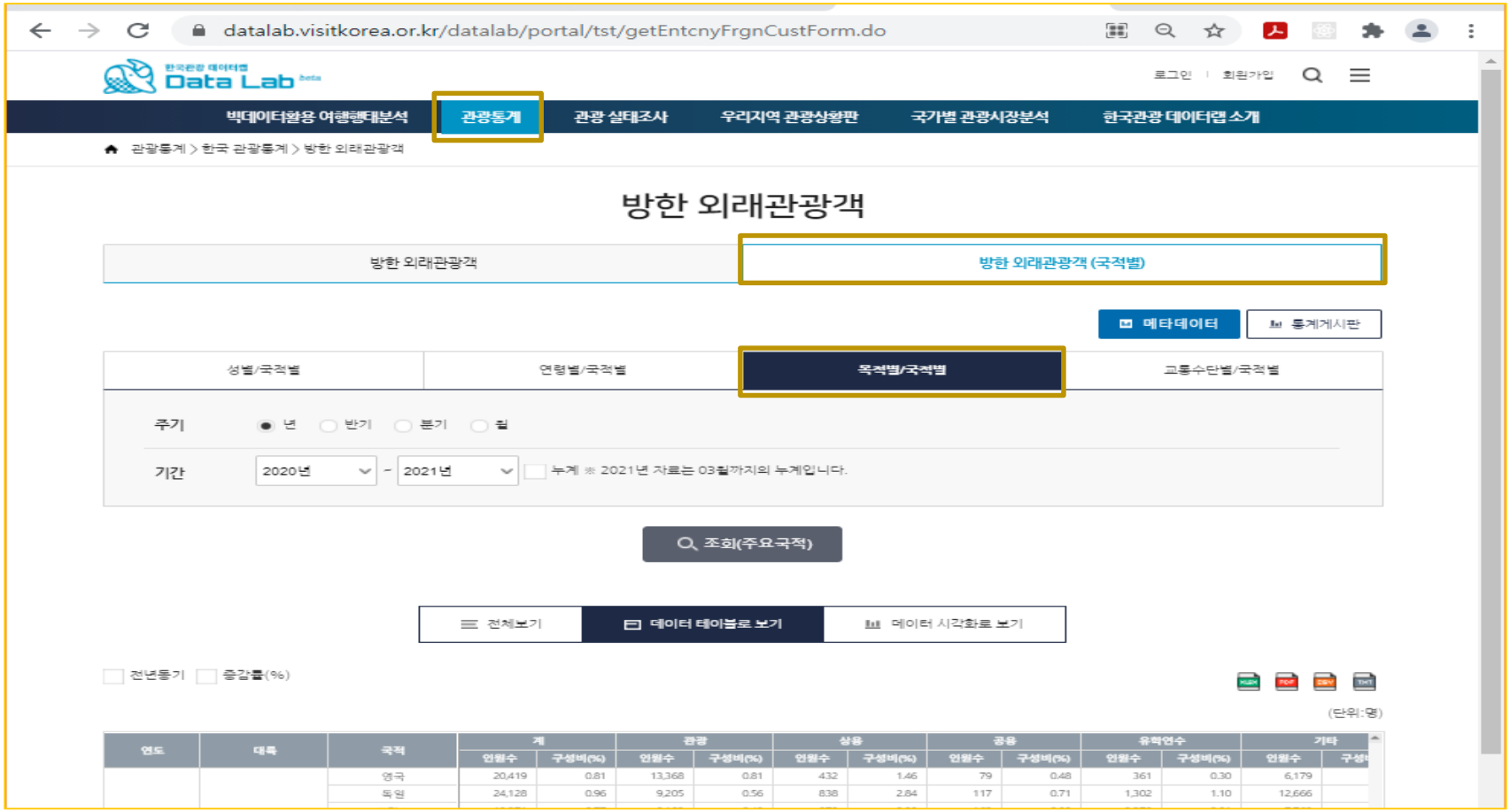
2021년 4월 29일

사드 배치의 영향으로 중국인 관광객이 얼마나 줄었을까?

- 월별 외국인 관광객 통계에 대한 데이터 수집
- 파이썬으로 전처리
- 전처리 결과를 “선 그래프(line plot) : 시계열 그래프”로 시각화
- 목표
  - 국적별로 외국인 관광객의 숫자에는 어떤 계절적인 패턴이 있는지 확인
  - 외국인 관광객들의 방문이 증가(또는 감소)한 원인이 되는 이벤트가 무엇인지 확인
- 사용 파이썬 라이브러리
  - pandas, matplotlib, seaborn
- 엑셀로도 가능하지만 단순반복작업을 계속 -> 시간소모적
  - kto_201001.xlsx ~kto_202005.xlsx의 125개 파일
  - 파이썬을 통해 반복 작업을 손쉽게 처리

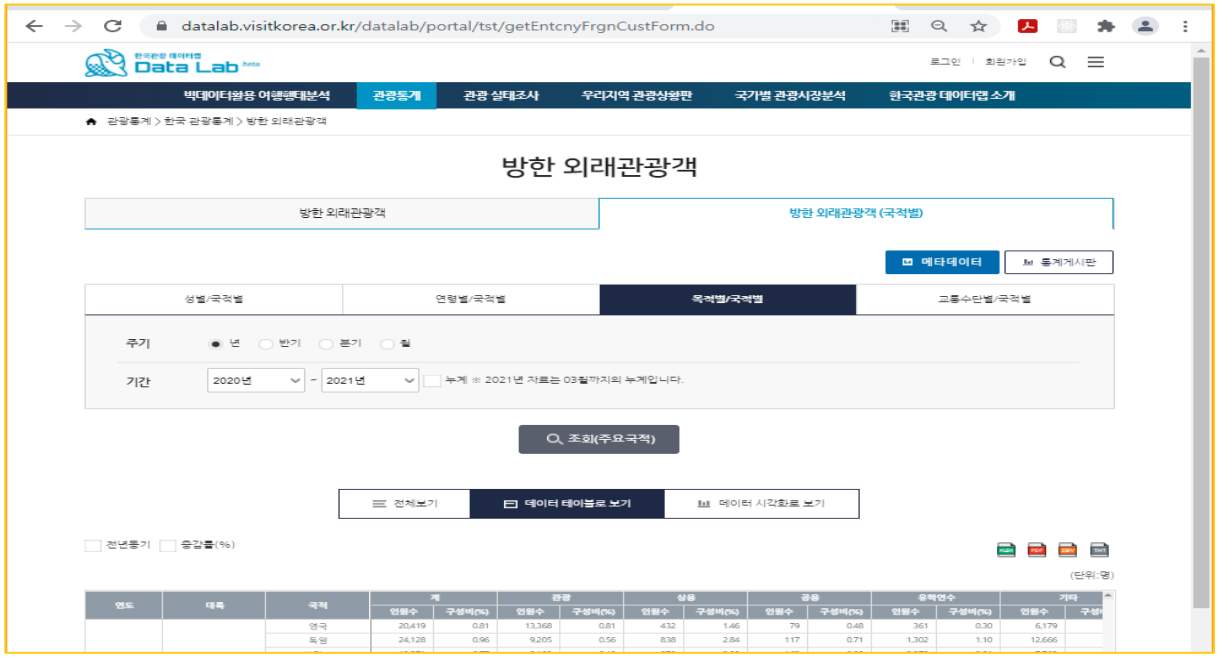
외국인 출입국 통계 데이터 구하기(1)

- 한국관광공사에서 수집
- 관광 정책 및 마케팅 전략 수립의 기초 자료로 활용할 목적으로 방한 외래관광객과 국민 해외관광객의 통계 자료를 2003년 1월부터 매월 발표
- 성별, 목적별, 연령별, 교통수단별, 국적별로 집계
- https://datalab.visitkorea.or.kr/datalab/portal/tst/getEntcnyFrgnCustForm.do



외국인 출입국 통계 데이터 구하기(2)

- [관광통계]-[방한 외래관광객(국적별)]-[목적별/국적별]-[월]
- 목적 분류 : 관광, 상용, 공용, 유학/연수, 기타
- 이 중 관광 목적으로 입국한 외국인 데이터 활용해 실습
- 2010년 1월 ~ 2020년 5월 데이터 : tour_files 폴더에 저장 후 배포
- 엑셀 데이터의 형태 파악
  - 월 별 관리 데이터는 대부분 형태(포맷) 동일 -> 관광 데이터는 포맷이 동일하지만 안 그런 데이터도 존재하므로 확인 필요
  - 그러므로, 한 파일의 형태를 파악해 두면 나머지도 동일한 형태로 처리



데이터 불러오기 및 전처리(1)

파이썬에서 관광객 엑셀 파일을 불러올 때 고려 사항

- 첫 번째 행(row)에는 해당 데이터가 어느 시점의 어떤 종류 데이터인지 서술 => 데이터 분석에 활용하지 않을 항목(속성)이므로 불러올 때 제외
- 두 번째 행에는 총 10개의 열(컬럼, column)이 순서대로 나열됨 => 실습에는 A부터 G까지 7개의 열(국적, 관광, 상용, 공용, 유학/연수, 기타, 계)을 활용
- 3~69 행에는 대륙, 국가에 따른 관광객 수 데이터로 구성됨 => 실제 분석에 활용할 데이터
- 70~73 행에는 계, 전년동기, 성장률, 구성비 등 데이터의 요약 정보로 구성됨 => 데이터 분석에 활용하지 않을 항목(행)이므로 불러올 때 제외

	A	B	C	D	E	F	G	H	I	J
1	2018년 02월 외래객 입국-목적별/국적별									
2	국적	관광	상용	공용	유학/연수	기타	계	전년동기	성장률(%)	구성비(%)
3	아시아주	658,805	9,302	1,362	49,322	126,437	845,228	1,084,723	-22.1	80.9
4	일본	159,831	1,966	144	1,898	4,402	168,241	185,032	-9.1	16.1
5	대만	86,02								
6	홍콩	49,17								
7	마카오	4,76								
8	태국	37,39								
9	말레이시아	19,51								
10	필리핀	14,54								
11	인도네시아	10,18								
12	싱가포르	7,22								

63	아프리카주	1,990	442	50	519	1,097	4,098	4,340	-5.6	0.4
64	남아프리카공화국	487	10	1	8	589	1,095	1,244	-12	0.1
65	아프리카 기타	1,503	432	49	511	508	3,003	3,096	-3	0.3
66	기타대륙	28	7	0	2	16	53	57	-7	0
67	국적미상	28	7	0	2	16	53	57	-7	0
68	교포소계	0	0	0	0	16,867	16,867	17,084	-1.3	1.6
69	교포	0	0	0	0	16,867	16,867	17,084	-1.3	1.6
70	계	790,477	11,323	3,669	54,618	185,328	1,045,415			
71	전년동기	952,948	13,790	3,003	59,952	222,387	1,252,080			
72	성장률(%)	-17	-17.9	22.2	-8.9	-16.7	-16.5			
73	구성비(%)	75.6	1.1	0.4	5.2	17.7	100			

데이터 불러오기 및 전처리(2)

- **엑셀 데이터 불러오기 : pandas의 read_excel() 함수 사용**

```
kto_201901 = pd.read_excel('./tour_files/kto_201901.xlsx',
                           header=1,
                           usecols='A:G',
                           skipfooter=4)
kto_201901.head()
```

	국적	관광	상용	공용	유학/연수	기타	계
0	아시아주	765082	10837	1423	14087	125521	916950
1	일본	198805	2233	127	785	4576	206526
2	대만	86393	74	22	180	1285	87954
3	홍콩	34653	59	2	90	1092	35896
4	마카오	2506	2	0	17	45	2570

```
kto_201901.tail()
```

	국적	관광	상용	공용	유학/연수	기타	계
62	아프리카 기타	768	718	90	206	908	2690
63	기타대륙	33	4	0	1	16	54
64	국적미상	33	4	0	1	16	54
65	교포소계	0	0	0	0	15526	15526
66	교포	0	0	0	0	15526	15526

- header=1 : 두 번째 줄에 변수명(타이틀)이 있는 것 의미
- usecols='A:G‘ : A~G 컬럼까지의 데이터 가져오기
- skipfooter=4 : 맨 아래 4줄 생략하고 불러오기
- kto_201901.head() : 윗 줄 가져오기
- kto_201901.tail() : 아랫줄 가져오기
- 데이터를 불러온 다음 꼭 head(), tail() 확인 => 원하는 형태로 불러왔는지 확인 필요, 원하는 형태로 불러오지 않은 경우 분석에 문제 생길 가능성이 커짐
- 전처리 후 2010년 1월 ~ 각각 월별 데이터를 불러와 하나의 테이블로 통합

데이터 불러오기 및 전처리(3)

데이터 전처리(Data Preprocessing)

- **데이터 가공(Data Manipulation), 데이터 클렌징(Data Cleaning), 데이터 핸들링(Data Handling) 등으로도 불림**
- **전처리에 포함되는 과정**
  - 데이터 변수별로 값에 이상이 없는지 확인
  - 결측치 처리
  - 이상치 처리
  - 변수 정규화
  - 파생 변수 생성
- **데이터 전처리에는 분석하려는 데이터에 대한 이해가 선행되어야 함 => 데이터 분석 기준을 정하기가 수월해 짐**
- **데이터 도메인에 대한 지식 필요**

데이터 불러오기 및 전처리(4)

데이터 탐색

- **컬럼별 특징 살펴보기**
- **kto_201901.info()**

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 67 entries, 0 to 66
Data columns (total 7 columns):
국적          67 non-null object
관광          67 non-null int64
상용          67 non-null int64
공용          67 non-null int64
유학/연수       67 non-null int64
기타          67 non-null int64
계           67 non-null int64
dtypes: int64(6), object(1)
memory usage: 3.8+ KB
```

- 데이터는 pandas 데이터프레임 클래스로 구성
- 0~66의 인덱스를 가진 데이터는 총 67개 행으로 구성
- 데이터 열은 7개(아래 나온 국적 ~계)
- 67 non-null : 7개의 변수 모두 빈 칸이 없음
- dtypes: 6개의 정수형 변수(국적 외), 1개의 문자형 변수(국적)로 구성됨을 알려줌

데이터 불러오기 및 전처리(5)

데이터 탐색

- 정수형 변수 특징 살펴보기
- kto_201901.describe()

	관광	상용	공용	유학/연수	기타	계
count	67.00000	67.000000	67.000000	67.000000	67.000000	67.000000
mean	26396.80597	408.208955	132.507463	477.462687	5564.208955	32979.194030
std	102954.04969	1416.040302	474.406339	2009.484800	17209.438418	122821.369969
min	0.00000	0.000000	0.000000	0.000000	16.000000	54.000000
25%	505.00000	14.500000	2.500000	17.500000	260.000000	927.000000
50%	1304.00000	45.000000	14.000000	43.000000	912.000000	2695.000000
75%	8365.00000	176.500000	38.000000	182.000000	2824.500000	14905.500000
max	765082.00000	10837.000000	2657.000000	14087.000000	125521.000000	916950.000000

- 평균적으로 가장 많은 외국인의 입국은 관광의 목적으로 온 경우 확인
- 다음으로 기타, 유학/연수, 상용, 공용 순을 확인
- 관광, 상용, 유학/연수, 공용의 최솟값이 0인 것 확인

데이터 불러오기 및 전처리(6)

데이터 탐색

- 각 컬럼에서 0인 데이터 필터링

```
condition = (kto_201901['관광'] == 0) ##
            | (kto_201901['상용'] == 0) ##
            | (kto_201901['공용'] == 0) ##
            | (kto_201901['유학/연수'] == 0)
kto_201901[condition]
```

- 교포소계와 교포의 4개 컬럼만 모두 0
- 이는 교포는 통계 집계시 기타 목적으로 분류되어 있기 때문

	국적	관광	상용	공용	유학/연수	기타	계
4	마카오	2506	2	0	17	45	2570
20	이스라엘	727	12	0	9	57	805
22	우즈베키스탄	1958	561	0	407	2828	5754
38	스위스	613	18	0	19	97	747
45	그리스	481	17	4	0	273	775
46	포르투갈	416	14	0	13	121	564
51	크로아티아	226	12	0	3	250	491
54	폴란드	713	10	0	27	574	1324
59	대양주 기타	555	3	4	0	52	614
63	기타대륙	33	4	0	1	16	54
64	국적미상	33	4	0	1	16	54
65	교포소계	0	0	0	0	15526	15526
66	교포	0	0	0	0	15526	15526

데이터 불러오기 및 전처리(7)

데이터프레임에 기준년월 추가

- 2010년 1월 부터 2020년 5월 까지의 데이터를 활용할 예정이기 때문에 각 데이터 마다 기준년월 정보 필요
- kto_201901['기준년월'] = '2019-01'
- kto_201901.head()

	국적	관광	상용	공용	유학/연수	기타	계	기준년월
0	아시아주	765082	10837	1423	14087	125521	916950	2019-01
1	일본	198805	2233	127	785	4576	206526	2019-01
2	대만	86393	74	22	180	1285	87954	2019-01
3	홍콩	34653	59	2	90	1092	35896	2019-01
4	마카오	2506	2	0	17	45	2570	2019-01

데이터 불러오기 및 전처리(8)

국적 데이터만 남기기

- 국적 컬럼 항목 확인
- kto_201901['국적'].unique()

```
array(['아시아주', '일본', '대만', '홍콩', '마카오', '태국', '말레이시아', '필리핀', '인도네시아',
       '싱가포르', '미얀마', '베트남', '인도', '스리랑카', '파키스탄', '방글라데시', '캄보디아', '몽골',
       '중국', '이란', '이스라엘', '터키', '우즈베키스탄', '카자흐스탄', 'GCC', '아시아 기타', '미주',
       '미국', '캐나다', '멕시코', '브라질', '미주 기타', '구주', '영국', '독일', '프랑스',
       '네덜란드', '스웨덴', '스위스', '이탈리아', '덴마크', '노르웨이', '벨기에', '오스트리아', '스페인',
       '그리스', '포르투갈', '핀란드', '아일랜드', '우크라이나', '러시아', '크로아티아', '루마니아',
       '불가리아', '폴란드', '구주 기타', '대양주', '오스트레일리아', '뉴질랜드', '대양주 기타',
       '아프리카주', '남아프리카공화국', '아프리카 기타', '기타대륙', '국적미상', '교포소계', '교포'],
      dtype=object)
```

- unique() : 컬럼 내 중복을 제거한 값(원소)들을 보여주는 함수
- 국적 컬럼에는 대륙과 국가가 혼용되어 있는 상태
- 하나의 컬럼에는 하나의 특징을 가진 값들이 들어있어야 분석에 활용 용이 => 국적 컬럼에 국가만 남기는 작업 필요

데이터 불러오기 및 전처리(9)

대륙 목록 만들기

- 앞 장의 국적 리스트에서 빨간 박스가 대륙 레벨임(7개)
- 국가 레벨을 제외한 7개의 대륙 레벨 데이터 값을 continents_list로 만들기
  - continents_list = ['아시아주', '미주', '구주', '대양주', '아프리카주', '기타대륙', '교포소계']
- kto_201901의 국적 컬럼에서 continents_list에 포함되지 않는 국가명만 선택

```
condition = (kto_201901.국적.isin(continents_list) == False)
kto_201901_country = kto_201901[condition]
kto_201901_country['국적'].unique()
```

```
1  kto_201901.국적.isin(continents_list)

0      True
1     False
2     False
3     False
4     False
      ...
62    False
63     True
64    False
65     True
66    False
Name: 국적, Length
```

	국적	condition
0	아시아주	False
1	일본	True
2	대만	True
3	홍콩	True
4	마카오	True
...	...	...
62	아프리카 기타	True
63	기타대륙	False
64	국적미상	True
65	교포소계	False
66	교포	True

```
array(['일본', '대만', '홍콩', '마카오', '태국', '말레이시아', '필리핀', '인도네시아', '싱가포르',
       '미얀마', '베트남', '인도', '스리랑카', '파키스탄', '방글라데시', '캄보디아', '몽골', '중국',
       '이란', '이스라엘', '터키', '우즈베키스탄', '카자흐스탄', 'GCC', '아시아 기타', '미국',
       '캐나다', '멕시코', '브라질', '미주 기타', '영국', '독일', '프랑스', '네덜란드', '스웨덴',
       '스위스', '이탈리아', '덴마크', '노르웨이', '벨기에', '오스트리아', '스페인', '그리스', '포르투갈',
       '핀란드', '아일랜드', '우크라이나', '러시아', '크로아티아', '루마니아', '불가리아', '폴란드',
       '구주 기타', '오스트레일리아', '뉴질랜드', '대양주 기타', '남아프리카공화국', '아프리카 기타',
       '국적미상', '교포'], dtype=object)
```

데이터 불러오기 및 전처리(10)

인덱스 재설정

	국적	관광	상용	공용	유학/연수	기타	계	기준년월
1	일본	198805	2233	127	785	4576	206526	2019-01
2	대만	86393	74	22	180	1285	87954	2019-01
3	홍콩	34653	59	2	90	1092	35896	2019-01
4	마카오	2506	2	0	17	45	2570	2019-01
5	태국	34004	37	199	96	6998	41334	2019-01

- **kto_201901_country.head()**
  - pandas에서 데이터프레임의 인덱스 값은 0부터 시작
  - kto_201901_country 는 인덱스가 1부터 시작
  - kto_201901_country 데이터가 kto_201901에서 필터링한 결과이기 때문 => 원본의 인덱스가 그대로 따라옴 => kto_201901_country의 인덱스 값은 대륙에 해당하는 인덱스가 누락된 상태
  - 인덱스 초기화 필요
- **kto_201901_country_newindex = kto_201901_country.reset_index(drop=True)**
  - reset_index() : 인덱스 값을 0부터 순차적으로 초기화
  - drop=True : 생략인 경우 기존 인덱스 값이 새로운 컬럼으로 생성됨
- **kto_201901_country_newindex.head()**

	국적	관광	상용	공용	유학/연수	기타	계	기준년월
0	일본	198805	2233	127	785	4576	206526	2019-01
1	대만	86393	74	22	180	1285	87954	2019-01
2	홍콩	34653	59	2	90	1092	35896	2019-01
3	마카오	2506	2	0	17	45	2570	2019-01
4	태국	34004	37	199	96	6998	41334	2019-01

데이터 불러오기 및 전처리(11)

대륙 컬럼 생성

- **국적 컬럼에 국가명만 남았으니 각 국가들이 어느 대륙에 속하는지에 대한 정보도 가지고 있으면 분석에 도움**
- **대륙 컬럼을 만들기 위해 컬럼의 내용을 어떻게 구성할지 고려**
  - 국가별로 매칭되는 대륙이 무엇인지 확인 => 엑셀 파일에서 파악
  - 아시아주 25개국(일본부터 아시아 기타)
  - 미주 5개국(미국부터 미주 기타)
  - 유럽(구주) 23개국(영국부터 구주 기타)
  - 오세아니아(대양주) 3개국(오스트레일리아, 뉴질랜드, 대양주 기타)
  - 아프리카주 2개국(남아프리카공화국, 아프리카주 기타)
  - 기타대륙 1개(국적미상)
  - 교포소계 1개(교포)
- **대륙 컬럼에 넣을 continents 리스트 생성**
  - continents = ['아시아']*25 + ['아메리카']*5 + ['유럽']*23 + ['오세아니아']*3 ₩
  + ['아프리카']*2 + ['기타대륙'] + ['교포']
  - print(continents)

```
['아시아', '아시아', '아시아', '아시아', '아시아', '아시아', '아시아', '아시아', '아시아', '아시아', '아시아', '아시아', '아시아', '아시아', '아시아', '아시아', '아시아', '아시아', '아시아', '아시아', '아시아', '아시아', '아시아', '아메리카', '아메리카', '아메리카', '아메리카', '아메리카', '유럽', '유럽', '유럽', '유럽', '유럽', '유럽', '유럽', '유럽', '유럽', '유럽', '유럽', '유럽', '유럽', '유럽', '유럽', '유럽', '유럽', '유럽', '유럽', '유럽', '유럽', '오세아니아', '오세아니아', '오세아니아', '아프리카', '아프리카', '기타대륙', '교포']
```

데이터 불러오기 및 전처리(12)

- continents를 kto_201901_country_newindex 데이터의 대륙 컬럼으로 추가
- kto_201901_country_newindex['대륙'] = continents
- kto_201901_country_newindex.head()
- kto_201901_country_newindex.tail()

	국적	관광	상용	공용	유학/연수	기타	계	기준년월	대륙
0	일본	198805	2233	127	785	4576	206526	2019-01	아시아
1	대만	86393	74	22	180	1285	87954	2019-01	아시아
2	홍콩	34653	59	2	90	1092	35896	2019-01	아시아
3	마카오	2506	2	0	17	45	2570	2019-01	아시아
4	태국	34004	37	19					

	국적	관광	상용	공용	유학/연수	기타	계	기준년월	대륙
55	대양주 기타	555	3	4	0	52	614	2019-01	오세아니아
56	남아프리카공화국	368	9	1	6	616	1000	2019-01	아프리카
57	아프리카 기타	768	718	90	206	908	2690	2019-01	아프리카
58	국적미상	33	4	0	1	16	54	2019-01	기타대륙
59	교포	0	0	0	0	15526	15526	2019-01	교포

데이터 불러오기 및 전처리(13)

국적별 관광객 비율 살펴보기

- 방문하는 외국인 중에서 관광 목적으로 입국하는 비율을 국가별로 비교
- 국적별로 관광객 수를 전체 입국객 수로 나눈 '관광객비율(%)' 컬럼 생성
- 관광객 비율 = 관광객수/전체입국객수 * 100 => 소수점 1자리까지
- kto_201901_country_newindex['관광객비율(%)'] = ₩
  round(kto_201901_country_newindex['관광'] /₩
  kto_201901_country_newindex['계'] * 100, 1)
- kto_201901_country_newindex.head()

	국적	관광	상용	공용	유학/연수	기타	계	기준년월	대륙	관광객비율(%)
0	일본	198805	2233	127	785	4576	206526	2019-01	아시아	96.3
1	대만	86393	74	22	180	1285	87954	2019-01	아시아	98.2
2	홍콩	34653	59	2	90	1092	35896	2019-01	아시아	96.5
3	마카오	2506	2	0	17	45	2570	2019-01	아시아	97.5
4	태국	34004	37	199	96	6998	41334	2019-01	아시아	82.3

데이터 불러오기 및 전처리(14)

관광객 비율 높은/낮은 국가 검색(5개국)

- 관광객비율(%) 컬럼으로 내림차순 정렬
- sort_values(by='관광객비율(%)', ascending=False) : 데이터 정렬 함수
- 파라미터 ascending=False ：내림차순 정렬에 사용
- 파라미터 ascending=True ：오름차순 정렬에 사용
- kto_201901_country_newindex.sort_values(by='관광객비율(%)', ascending=False).head()
- kto_201901_country_newindex.sort_values(by='관광객비율(%)', ascending=True).head()

	국적	관광	상용	공용	유학/연수	기타	계	기준년월	대륙	관광객비율(%)
1	대만	86393	74	22	180	1285	87954	2019-01	아시아	98.2
3	마카오	2506	2	0	17	2	2527	2019-01	아시아	99.2
2	홍콩	34653	59	2	90	1	34805	2019-01	아시아	99.6
0	일본	198805	2233	127	785	1	201951	2019-01	아시아	98.4
55	대양주 기타	555	3	4	0	0	562	2019-01	대양주	98.8

	국적	관광	상용	공용	유학/연수	기타	계	기준년월	대륙	관광객비율(%)
59	교포	0	0	0	0	15526	15526	2019-01	교포	0.0
14	방글라데시	149	126	27	97	848	1247	2019-01	아시아	11.9
12	스리랑카	157	54	5	28	1043	1287	2019-01	아시아	12.2
13	파키스탄	238	178	10	193	413	1032	2019-01	아시아	23.1
15	캄보디아	635	39	55	51	1915	2695	2019-01	아시아	23.6

데이터 불러오기 및 전처리(15)

대륙별 관광객 비율의 평균

- 피벗 테이블로 확인
- kto_201901_country_newindex.pivot_table(values = '관광객비율(%)',
  index = '대륙',
  aggfunc = 'mean')

대륙	관광객비율(%)
교포	0.000000
기타대륙	61.100000
아메리카	68.200000
아시아	59.624000
아프리카	32.700000
오세아니아	84.833333
유럽	63.826087

- 평균 관광객 비율이 높은 대륙 : 오세아니아
- 반대로 아프리카는 32.7%만 관광 목적으로 방문
- 거리가 가까운 아시아가 아메리카나 유럽보다 낮은 이유는 앞에서 살펴보듯이 관광객 비율 최하 5개국이 아시아에 속한 이유 때문 => 관광 목적으로 방문하는 비율이 낮은 나라에 대한 이유는 다른 요인을 추가 분석한 후 결론을 도출해야 함

데이터 불러오기 및 전처리(16)

중국 관광객 비율

- 중국 국적만 필터링
- condition = (kto_201901_country_newindex.국적 == '중국')
- kto_201901_country_newindex[condition]

	국적	관광	상용	공용	유학/연수	기타	계	기준년월	대륙	관광객비율(%)
17	중국	320113	2993	138	8793	60777	392814	2019-01	아시아	81.5

중국인 5명 중 4명은 관광객

```
1  kto_201901_country_newindex.국적 == '중국'
```

```
0     False
1     False
2     False
3     False
4     False
5     False
6     False
7     False
8     False
9     False
10    False
11    False
12    False
13    False
14    False
15    False
16    False
17     True
18    False
19    False
20    False
21    False
```

```
1  kto_201901_country_newindex
```

	국적	관광	상용	공용	유학/연수	기타	계	기준년월	대륙	관광객비율(%)
0	일본	198805	2233	127	785	4576	206526	2019-01	아시아	96.3
1	대만	86393	74	22	180	1285	87954	2019-01	아시아	98.2
2	홍콩	34653	59	2	90	1092	35896	2019-01	아시아	96.5
3	마카오	2506	2	0	17	45	2570	2019-01	아시아	97.5
4	태국	34004	37	199	96	6998	41334	2019-01	아시아	82.3
5	말레이시아	19043	95	7	99	2821	22065	2019-01	아시아	86.3
6	필리핀	14279	211	161	184	15638	30473	2019-01	아시아	46.9
7	인도네시아	14183	136	38	187	4298	18842	2019-01	아시아	75.3
8	싱가포르	8372	94	8	48	1333	9855	2019-01	아시아	85.0
9	미얀마	1304	10	31	67	3877	5289	2019-01	아시아	24.7
10	베트남	10739	763	110	1667	6904	20183	2019-01	아시아	53.2
11	인도	2318	2656	46	177	3474	8671	2019-01	아시아	26.7
12	스리랑카	157	54	5	28	1043	1287	2019-01	아시아	12.2
13	파키스탄	238	178	10	193	413	1032	2019-01	아시아	23.1
14	방글라데시	149	126	27	97	848	1247	2019-01	아시아	11.9
15	캄보디아	635	39	55	51	1915	2695	2019-01	아시아	23.6
16	몽골	8358	77	304	484	562	9785	2019-01	아시아	85.4
17	중국	320113	2993	138	8793	60777	392814	2019-01	아시아	81.5
18	이란	60	45	10	23	46	184	2019-01	아시아	32.6
19	이스라엘	727	12	0	9	57	805	2019-01	아시아	90.3
20	터키	792	13	32	36	912	1785	2019-01	아시아	44.4

데이터 불러오기 및 전처리(17)

기준년월별로 전체 외국인 관광객 대비 국적별 관광객 비율 확인

- 외국인 관광객의 국적별 비율
- 관광 목적으로 방문하는 전체 외국인들 대비 국적별 관광객 비율
- 2019년 1월 전체 외국인 관광객 숫자 검색
  - tourist_sum = sum(kto_201901_country_newindex['관광'])
  - tourist_sum => 결과 : 884293
- 국적별관광객비율 = 국적별관광객수/tourist_sum*100
  - kto_201901_country_newindex['전체비율(%)'] = ₩
    round(kto_201901_country_newindex['관광'] / tourist_sum * 100, 1)
  - kto_201901_country_newindex.head()

	국적	관광	상용	공용	유학/연수	기타	계	기준년월	대륙	관광객비율(%)	전체비율(%)
0	일본	198805	2233	127	785	4576	206526	2019-01	아시아	96.3	22.5
1	대만	86393	74	22	180	1285	87954	2019-01	아시아	98.2	9.8
2	홍콩	34653	59	2	90	1092	35896	2019-01	아시아	96.5	3.9
3	마카오	2506	2	0	17	45	2570	2019-01	아시아	97.5	0.3
4	태국	34004	37	199	96	6998	41334	2019-01	아시아	82.3	3.8

데이터 불러오기 및 전처리(17)

전체비율(%) 컬럼 상위 5개국과 비율 검색

- kto_201901_country_newindex.sort_values('전체비율(%)', ₩ ascending=False).head()
- 중국인 관광객이 36.2% : 가장 높은 비율
- 일본, 대만, 미국, 홍콩 순
- 상위 5개국이 77.3% 차지

	국적	관광	상용	공용	유학/연수	기타	계	기준년월	대륙	관광객비율(%)	전체비율(%)
17	중국	320113	2993	138	8793	60777	392814	2019-01	아시아	81.5	36.2
0	일본	198805	2233	127	785	4576	206526	2019-01	아시아	96.3	22.5
1	대만	86393	74	22	180	1285	87954	2019-01	아시아	98.2	9.8
25	미국	42989	418	2578	229	16523	62737	2019-01	아메리카	68.5	4.9
2	홍콩	34653	59	2	90	1092	35896	2019-01	아시아	96.5	3.9

데이터 전처리 소결론

- 데이터를 분석하기 적합한 형태로 만드는 과정
- 데이터에 대한 이해도를 높이는 과정(미시적 관점으로 살펴봄)
- 위 두 과정이 결과를 해석하는데 가장 필요한 과정

데이터 전처리 과정을 함수로 만들기

- 2019년 1월 데이터에 대한 처리만 진행했지만 125개 파일에 대해 같은 작업 진행 필요 => 2010년 1월~2020년 5월 데이터를 하나로 생성
- 작업 단위별로 함수 만든 후 반복문으로 처리

```
def create_kto_data(yy, mm):
    #1. 불러올 Excel 파일 경로를 지정해주기
    file_path = './tour_files/kto_{}{}.xlsx'.format(yy, mm)

    # 2. Excel 파일 불러오기
    df = pd.read_excel(file_path, header=1, skipfooter=4, usecols='A:G')

    # 3. "기준년월" 컬럼 추가하기
    df['기준년월'] = '{}-{}'.format(yy, mm)

    # 4. "국적" 컬럼에서 대륙 제거하고 국가만 남기기
    ignore_list = ['아시아주', '미주', '구주', '대양주', '아프리카주', '기타대륙', '교포소계']    # 제거할 대륙명 선정하기
    condition = (df['국적'].isin(ignore_list) == False)    # 대륙 미포함 조건
    df_country = df[condition].reset_index(drop=True)

    # 5. "대륙" 컬럼 추가하기
    continents = ['아시아']*25 + ['아메리카']*5 + ['유럽']*23 + ['대양주']*3 + ['아프리카']*2 + ['기타대륙'] + ['교포']
    df_country['대륙'] = continents

    # 6. 국가별 "관광객비율(%)" 컬럼 추가하기
    df_country['관광객비율(%)'] = round(df_country.관광 / df_country.계 * 100, 1)

    # 7. "전체비율(%)" 컬럼 추가하기
    tourist_sum = sum(df_country['관광'])
    df_country['전체비율(%)'] = round(df_country['관광'] / tourist_sum * 100, 1)

    # 8. 결과 출력하기
    return(df_country)
```

함수 작동 확인

- 테스트 데이터 : 2018년 12월
- kto_test = create_kto_data(2018, 12)
- kto_test.head()

	국적	관광	상용	공용	유학/연수	기타	계	기준년월	대륙	관광객비율(%)	전체비율(%)
0	일본	252461	1698	161	608	3593	258521	2018-12	아시아	97.7	22.7
1	대만	85697	71	22	266	1252	87308	2018-12	아시아	98.2	7.7
2	홍콩	58355	41	3	208	939	59546	2018-12	아시아	98.0	5.2
3	마카오	6766	0	1	20	36	6823	2018-12	아시아	99.2	0.6
4	태국	47242	42	302	58	6382	54026	2018-12	아시아	87.4	4.2

	A	B	C	D	E	F	G	H	I	J
1	2018년 12월 외래객 입국-목적별/국적별									
2	국적	관광	상용	공용	유학/연수	기타	계	전년동기	성장률(%)	구성비(%)
3	아시아주	978,485	10,685	2,198	17,595	116,380	1,125,343	941,752	19.5	85
4	일본	252,461	1,698	161	608	3,593	258,521	193,705	33.5	19.5
5	대만	85,697	71	22	266	1,252	87,308	75,738	15.3	6.6
6	홍콩	58,355	41	3	208	939	59,546	58,761	1.3	4.5
7	마카오	6,766	0	1	20	36	6,823	8,017	-14.9	0.5
8	태국	47,242	42	302	58	6,382	54,026	60,147	-10.2	4.1
9	말레이시아	52,698	89	3	114	2,761	55,665	45,333	22.8	4.2
10	필리핀	28,222	140	127	54	14,107	42,650	37,019	15.2	3.2
11	인도네시아	21,276	185	209	141	4,278	26,089	22,012	18.5	2

반복문을 통해 다수의 엑셀 데이터를 불러와서 합치기

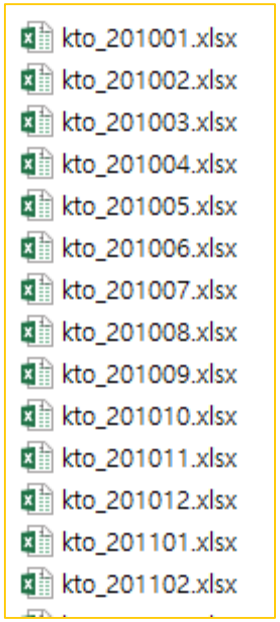
- **파일 형태 확인**
- **파일 이름 특징**
  - kto_yyyymm.xlsx
  - yyyy : 2010~2020까지 순차적으로 들어감
  - 하나의 yyyy에 대해 mm은 1~12의 값이 순차적으로 들어감
  - mm은 “01”, “02”, ……, “12”와 같이 두 자리의 문자로 구성
  - 년도에 대한 반복문과 그 안에서 월에 대한 반복문 필요
  - 예 : 이중 반복문으로 기준년월 출력

```
for yy in range(2010, 2021):
    for mm in range(1, 13):
        yymm = '{}{}'.format(yy, mm)
        print(yymm)
```

  - 월 mm을 두 자리 문자열로 표현하는 것이 반영되지 않음
  - zfill()을 사용 : ()안의 숫자만큼 0을 채워 자릿수를 맞춤
  - 주의점은 zfill() 앞의 값은 문자열이어야 함

```
1  mm=1
2  print(mm)
3  print(str(mm).zfill(2))
```

```
1
01
```




```
20101
20102
20103
20104
20105
20106
20107
20108
20109
201010
201011
201012
20111
20112
```

- **6자리로 정렬하여 기준년월 출력**

```
for yy in range(2010, 2021):
    for mm in range(1, 13):
        mm_str = str(mm).zfill(2)
        yymm = '{}{}'.format(yy, mm_str)
        print(yymm)
```

```
201001
201002
201003
201004
201005
201006
201007
201008
201009
201010
201011
201012
201101
201102
```

- **125개 데이터 파일 합치기**
  - 각 파일을 합해 새 파일 만들기 위한 작업 필요
    - df = pd.DataFrame()
  - 엑셀 파일 불러와서 합하기

```
for yy in range(2010, 2021):
    for mm in range(1, 13):
        temp = create_kto_data(str(yy), str(mm).zfill(2))
        df = df.append(temp, ignore_index=True)
```

	국적	관광	상용	공용	유학/연수	기타	계	기준년월	대륙	관광객비율(%)	전체비율(%)
0	일본	202825	1750	89	549	3971	209184	2010-01	아시아	97.0	50.6
1	대만	35788	41	17	37	516	36399	2010-01	아시아	98.3	8.9
2	홍콩	13874	55	0	21	595	14545	2010-01	아시아	95.4	3.5
3	마카오	554	0	0	0	0	554	2010-01	아시아	100.0	0.1
4	태국	13374	39	13	53	4335	17814	2010-01	아시아	75.1	3.3

	국적	관광	상용	공용	유학/연수	기타	계	기준년월	대륙	관광객비율(%)	전체비율(%)
7495	대양주 기타	1	0	1	0	3	5	2020-05	대양주	20.0	0.0
7496	남아프리카공화국	1	0	3	0	25	29	2020-05	아프리카	3.4	0.0
7497	아프리카 기타	3	1	3	6	118	131	2020-05	아프리카	2.3	0.0
7498	국적미상	1	0	0	0	3	4	2020-05	기타대륙	25.0	0.0
7499	교포	0	0	0	0	790	790	2020-05	교포	0.0	0.0

  - 2020년은 5월까지 밖에 없으므로 2020년 6월 처리시 에러 발생
  - 우리의 목적은 2020년 5월까지 합하는 것이므로 에러 무시 => 꼭 필요하면 수정해도 좋음
  - 결과를 head()와 tail()로 확인

- df 데이터를 info()로 확인
  - df.info()
  - 125개월*60개국적 => 7500개 행(row) 생성
- 통합데이터 엑셀로 저장
  - df.to_excel('./tour_files/kto_total.xlsx', index = False)

국적별 필터링된 데이터를 엑셀 파일로 저장

- 통합 데이터가 있으면 이후 분석 과정에 필요한 작업을 적절히 수행 가능
- 경우에 따라서는 특성에 맞는 개별 데이터로 저장할 필요성 있음
- kto_total.xlsx에 저장한 df에서 국적별로 필터링 => 60개의 국적별 엑셀 파일로 저장
- 중국인 관광객 예제

```
1 condition = (df['국적'] == '중국')
2 df_filter = df[condition]
3 df_filter.head()
```

	국적	관광	상용	공용	유학/연수	기타	계	기준년월	대륙	관광객비율(%)	전체비율(%)
17	중국	40425	11930	55	2751	36091	91252	2010-01	아시아	44.3	10.1
77	중국	60590	7907	68	29546	42460	140571	2010-02	아시아	43.1	13.6
137	중국	50330	13549	174	14924	62480	141457	2010-03	아시아	35.6	9.2
197	중국	84252	13306	212	2199	47711	147680	2010-04	아시아	57.1	15.5
257	중국	89056	12325	360	2931	49394	154066	2010-05	아시아	57.8	17.0

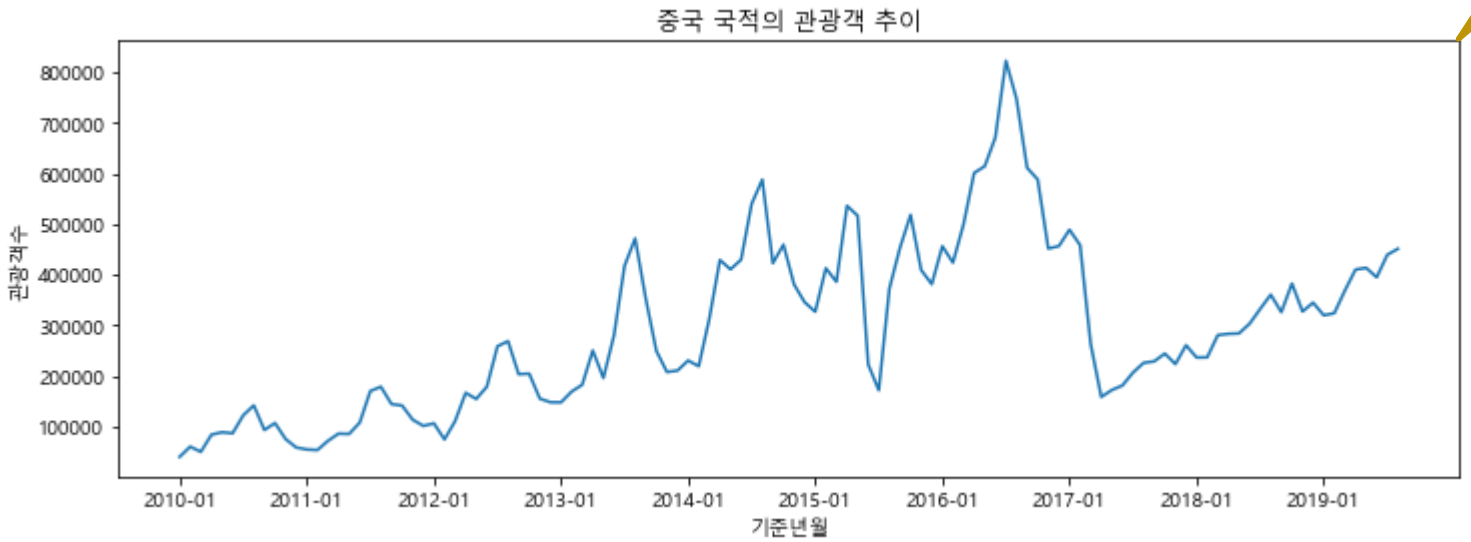
데이터 시각화

시각화의 중요성 : 간단하고 효과적으로 메시지 전달 가능

- 시계열 그래프와 히트맵 그래프로 시각화하여 결과 확인

	국적	관광	상용	공용	유학/연수	기타	계	기준년월	대륙	관광객비율(%)	전체비율(%)
17	중국	40425	11930	55	2751	36091	91252	2010-01	아시아	44.3	10.1
77	중국	60590	7907	68	29546	42460	140571	2010-02	아시아	43.1	13.6
137	중국	50330	13549	174	14924	62480	141457	2010-03	아시아	35.6	9.2
197	중국	84252	13306	212	2199	47711	147680	2010-04	아시아	57.1	15.5
257	중국	89056	12325	360	2931	49394	154066	2010-05	아시아	57.8	17.0

전체적인 추세나 계절적인 패턴을 직관적으로 확인



시계열 그래프 그리기

- 월별, 국적별 관광객 데이터의 특징을 시각화
- 시각화 대상 데이터 불러오기

```
df = pd.read_excel('tour_files/kto_total.xlsx')
df.head()
```

	국적	관광	상용	공용	유학/연수	기타	계	기준년월	대륙	관광객비율(%)	전체비율(%)
0	일본	202825	1750	89	549	3971	209184	2010-01	아시아	97.0	50.6
1	대만	35788	41	17	37	516	36399	2010-01	아시아	98.3	8.9
2	홍콩	13874	55	0	21	595	14545	2010-01	아시아	95.4	3.5
3	마카오	554	0	0	0	0	554	2010-01	아시아	100.0	0.1
4	태국	13374	39	13	53	4335	17814	2010-01	아시아	75.1	3.3

- 환경 설정 : 사용 중인 각 운영체제에 따라 한글을 지원하는 글꼴로 변경하는 코드

```
from matplotlib import font_manager, rc
import platform

if platform.system() == 'Windows':
    path = 'c:/Windows/Fonts/malgun.ttf'
    font_name = font_manager.FontProperties(fname=path).get_name()
    rc('font', family=font_name)
elif platform.system() == 'Darwin':
    rc('font', family='AppleGothic')
else:
    print('Check your OS system')
```

- **시각화 라이브러리 불러오기**

  import matplotlib.pyplot as plt

  - seaborn 라이브러리도 matplotlib를 토대로 확장한 라이브러리임

- **월별 중국인 관광객 그래프 그리기**

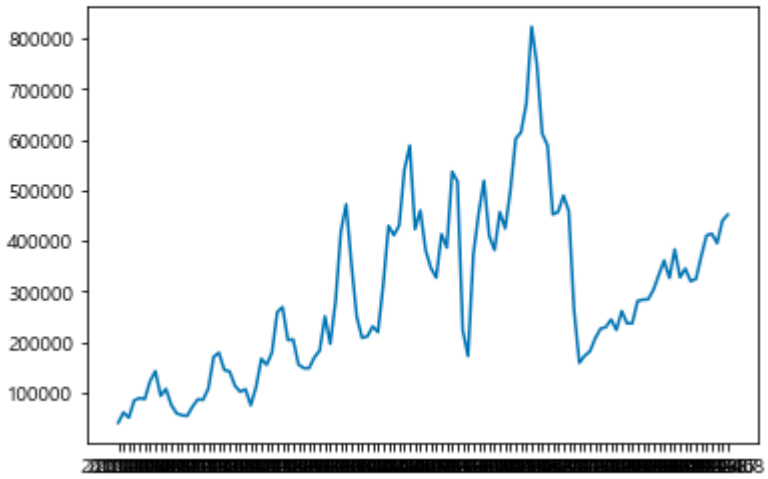
  - 전체 관광객 데이터 중 중국 국적 관광객수 추출

```
1 condition = (df['국적'] == '중국')
2 df_filter = df[condition]
3 df_filter.head()
```

	국적	관광	상용	공용	유학/연수	기타	계	기준년월	대륙	관광객비율(%)	전체비율(%)
17	중국	40425	11930	55	2751	36091	91252	2010-01	아시아	44.3	10.1
77	중국	60590	7907	68	29546	42460	140571	2010-02	아시아	43.1	13.6
137	중국	50330	13549	174	14924	62480	141457	2010-03	아시아	35.6	9.2
197	중국	84252	13306	212	2199	47711	147680	2010-04	아시아		
257	중국	89056	12325	360	2931	49394	154066	2010-05	아시아		

- 시계열 그래프 그리기1

```
1 plt.plot(df_filter['기준년월'], df_filter['관광'])
2 plt.show()
```



- plt.plot()
  - 첫번째 인수 : 그래프 x 축에 지정할 컬럼
  - 두번째 인수 : 그래프 y 축에 지정할 컬럼
  - 기준년월이 x 축, 관광이 y 축에 들어간 결과 그래프

▪ 축 값의 기준 변경을 통한 보기 좋은 그래프 작성

- 그래프 크기 조절

  plt.figure(figsize = (12, 4)) : 그래프 크기를 가로 12인치, 세로 4인치의 그래프 생성

- 그래프 내용 설정

  plt.plot(df_filter['기준년월'], df_filter['관광']) : x축에는 기준년월, y축에는 관광인 변수 출력

- 그래프 타이틀, X축, Y축 이름 지정

  plt.title('중국 국적의 관광객 추이') : 그래프 제목 입력

  plt.xlabel('기준년월') : x축 이름

  plt.ylabel('관광객수') : y축 이름

- X 축 눈금값 설정

  plt.xticks(['2010-01', '2011-01', '2012-01', '2013-01', '2014-01', '2015-01', '2016-01', '2017-01', '2018-01', '2019-01', '2020-01']) : 그래프에 표시할 x 축 눈금 값 지정

- 그래프 표현

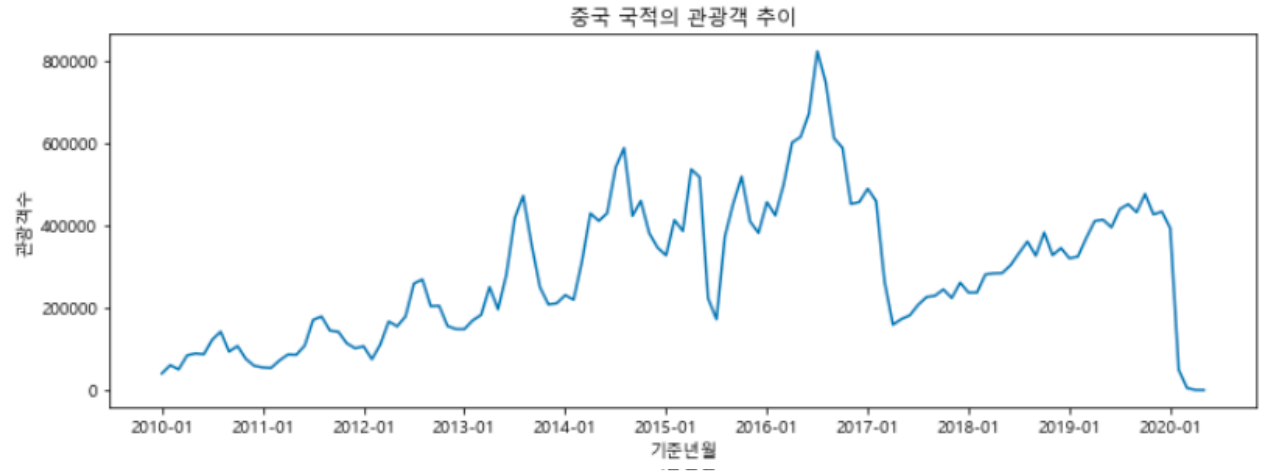
  plt.show()

상위 5개 국가 별로 시계열 그래프 작성

- **중국, 일본, 대만, 미국, 홍콩**
- **앞에서 그린 중국 관광객 그래프 참고해 그래프 생성**
  - 국적 조건과 타이틀 부분은 각각 변경 필요
  - 다른 부분은 동일
  - 반복문 사용
  - 외국인 관광객 중 상위 5개 국가 리스트 만들기

    cntry_list = ['중국', '일본', '대만', '미국', '홍콩']
  - 반복문으로 여러 그래프 그리기

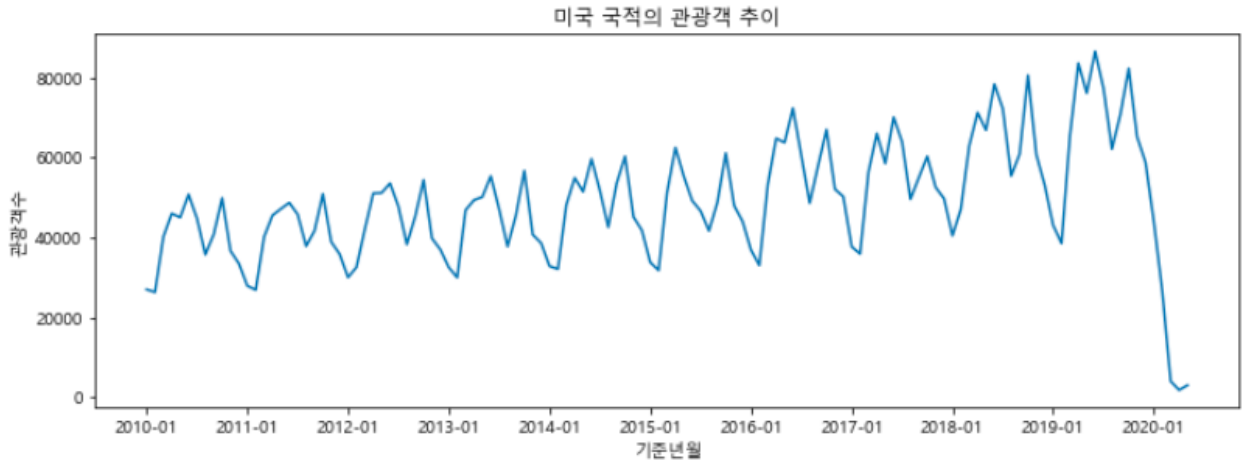
```
for cntry in cntry_list:
    # 국적 관광객만 추출하기
    condition = (df['국적'] == cntry)
    df_filter = df[condition]

    # 그래프 그리기
    ## 그래프 크기 조절
    plt.figure(figsize = (12, 4))

    ## 그래프 내용 설정
    plt.plot(df_filter['기준년월'], df_filter['관광'])

    ## 그래프 타이틀, X축, Y축 이름 달기
    plt.title('{} 국적의 관광객 추이'.format(cntry))
    plt.xlabel('기준년월')
    plt.ylabel('관광객수')

    ## x 축 눈금 값 설정
    plt.xticks(['2010-01', '2011-01', '2012-01', '2013-01',#
                '2014-01', '2015-01', '2016-01', '2017-01',#
                '2018-01', '2019-01', '2020-01'])

    ## 그래프 표현하기
    plt.show()
```


중국 국적의 관광객 추이
관광객수
800000
600000
400000
200000
0
2010-01
2011-01
2012-01
2013-01
2014-01
2015-01
2016-01
2017-01
2018-01
2019-01
2020-01
기준년월


일본 국적의 관광객 추이
관광객수
350000
300000
250000
200000
150000
100000
50000
0
2010-01
2011-01
2012-01
2013-01
2014-01
2015-01
2016-01
2017-01
2018-01
2019-01
2020-01
기준년월


홍콩 국적의 관광객 추이
관광객수
80000
70000
60000
50000
40000
30000
20000
10000
0
2010-01
2011-01
2012-01
2013-01
2014-01
2015-01
2016-01
2017-01
2018-01
2019-01
2020-01
기준년월


대만 국적의 관광객 추이
관광객수
120000
100000
80000
60000
40000
20000
0
2010-01
2011-01
2012-01
2013-01
2014-01
2015-01
2016-01
2017-01
2018-01
2019-01
2020-01
기준년월


미국 국적의 관광객 추이
관광객수
80000
60000
40000
20000
0
2010-01
2011-01
2012-01
2013-01
2014-01
2015-01
2016-01
2017-01
2018-01
2019-01
2020-01
기준년월

히트맵 그래프 그리기

- 매트릭스(matrix) 형태에 포함된 각 값을 컬러로 표현하는 데이터 시각화 방법
- 전체 데이터를 한 눈에 파악할 수 있다는 장점
- X축, Y축, 그래프 내용에 어떤 변수들이 들어가야 할 지 고민해야 함
- 관광객 실습 히트맵 그래프
  - X축 : 월, Y축 : 년도, 그래프 내용 : 관광객 숫자
  - 데이터 확인

```
1  df.head()
```

	국적	관광	상용	공용	유학/연수	기타	계	기준년월	대륙	관광객비율(%)	전체비율(%)
0	일본	202825	1750	89	549	3971	209184	2010-01	아시아	97.0	50.6
1	대만	35788	41	17	37	516	36399	2010-01	아시아	98.3	8.9
2	홍콩	13874	55	0	21	595	14545	2010-01	아시아	95.4	3.5
3	마카오	554	0	0	0	0	554	2010-01	아시아	100.0	0.1
4	태국	13374	39	13	53	4335	17814	2010-01	아시아	75.1	3.3

년도와 월을 구분된 변수로 만들기

▪ str.slice() 함수를 이용한 년도, 월 컬럼 만들기

```
df['년도'] = df['기준년월'].str.slice(0,4)
df['월'] = df['기준년월'].str.slice(5, 7)
df.head()
```

	국적	관광	상용	공용	유학/연수	기타	계	기준년월	대륙	관광객비율(%)	전체비율(%)	년도	월
0	일본	202825	1750	89	549	3971	209184	2010-01	아시아	97.0	50.6	2010	01
1	대만	35788	41	17	37	516	36399	2010-01	아시아	98.3	8.9	2010	01
2	홍콩	13874	55	0	21	595	14545	2010-01	아시아	95.4	3.5	2010	01
3	마카오	554	0	0	0	0	554	2010-01	아시아	100.0	0.1	2010	01
4	태국	13374	39	13	53	4335	17814	2010-01	아시아	75.1	3.3	2010	01

▪ 매트릭스 형태 만들기 :pivot_table() 함수 이용

- 중국 국적 데이터만 추출

```
condition = (df['국적'] == '중국')
df_filter = df[condition]
df_filter.head()
```

	국적	관광	상용	공용	유학/연수	기타	계	기준년월	대륙	관광객비율(%)	전체비율(%)	년도	월
17	중국	40425	11930	55	2751	36091	91252	2010-01	아시아	44.3	10.1	2010	01
77	중국	60590	7907	68	29546	42460	140571	2010-02	아시아	43.1	13.6	2010	02
137	중국	50330	13549	174	14924	62480	141457	2010-03	아시아	35.6	9.2	2010	03
197	중국	84252	13306	212	2199	47711	147680	2010-04	아시아	57.1	15.5	2010	04
257	중국	89056	12325	360	2931	49394	154066	2010-05	아시아	57.8	17.0	2010	05

df_filter 데이터를 매트릭스 형태로 변환

- pivot_table()를 사용해 히트맵 그래프로 표현하고자 하는 형태 만들기
- index : y축에 표현될 년도
- columns : x축에 표현될 월
- value : 표 내용을 의미하는 관광객수

```
df_pivot = df_filter.pivot_table(values = '관광'
                                 , index = '년도'
                                 , columns = '월')

df_pivot
```

월	01	02	03	04	05	06	07	08	09	10	11	12
년도												
2010	40425.0	60590.0	50330.0	84252.0	89056.0	87080.0	122432.0	142180.0	93545.0	107237.0	75686.0	58987.0
2011	55070.0	53863.0	72003.0	86397.0	85668.0	108060.0	170524.0	178937.0	144704.0	141824.0	113856.0	101605.0
2012	106606.0	74895.0	110965.0	166843.0	154841.0	179074.0	258907.0	268988.0	203857.0	204866.0	155503.0	148320.0
2013	148118.0	169395.0	182850.0	250549.0	196306.0	280319.0	417991.0	472005.0	353359.0	249850.0	208175.0	210950.0
2014	230706.0	219533.0	313400.0	429419.0	410971.0	429991.0	540683.0	588181.0	423133.0	459708.0	381118.0	345957.0
2015	327225.0	413096.0	386386.0	536428.0	517154.0	223101.0	172075.0	372990.0	453670.0	518651.0	409635.0	381722.0
2016	456636.0	424232.0	500018.0	601460.0	614636.0	671493.0	823016.0	747818.0	611538.0	588561.0	452082.0	456882.0
2017	489256.0	458952.0	263788.0	158784.0	172527.0	181507.0	207099.0	226153.0	229172.0	244541.0	223743.0	260983.0
2018	236825.0	237075.0	281020.0	283533.0	284317.0	303405.0	332657.0	360982.0	326438.0	382922.0	327664.0	345135.0
2019	320113.0	324291.0	369165.0	410542.0	413949.0	395196.0	439699.0	451570.0	432018.0	476460.0	426849.0	433577.0
2020	393336.0	49520.0	5040.0	522.0	179.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN

히트맵 그래프

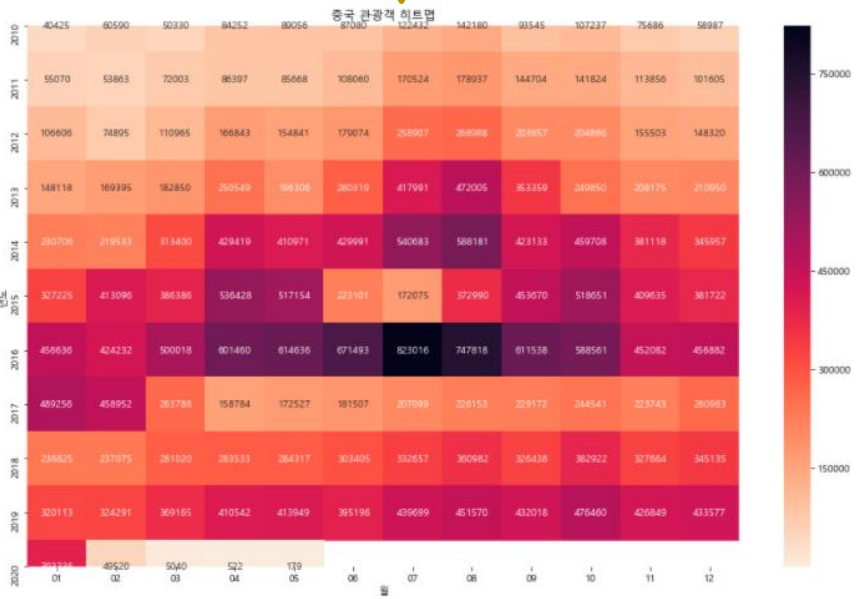
- 파이썬의 기본 시각화 라이브러리인 matplotlib 에서 지원하지 않고
- seaborn 라이브러리를 통해 생성
- 라이브러리 seaborn은 matplotlib에 종속되므로 다음과 같이 import

  import matplotlib.pyplot as plt

  import seaborn as sns
- 히트맵 그리기 위한 설정

```
## 그래프 크기 설정
plt.figure(figsize = (16, 10))

## 히트맵 그래프 그리기
sns.heatmap(df_pivot, annot = True, fmt = '.0f', cmap = 'rocket_r')

## 그래프 타이틀 달기
plt.title('중국 관광객 히트맵')

## 그래프 표현
plt.show()
```

관광객 수가 많을수록
색이 진해짐



- sns.heatmap() : 히트맵 그래프 그리는 함수
  - df_pivot : 히트맵 그래프로 나타낼 데이터 지정
  - annot = True : 히트맵 그래프에 각 칸에 실제 값 표시
  - fmt = '.0f' : 숫자 형태를 소수점이 없는 실수형으로 표현
  - cmap = 'rocket_r' : 그래프의 색깔 조합 지정

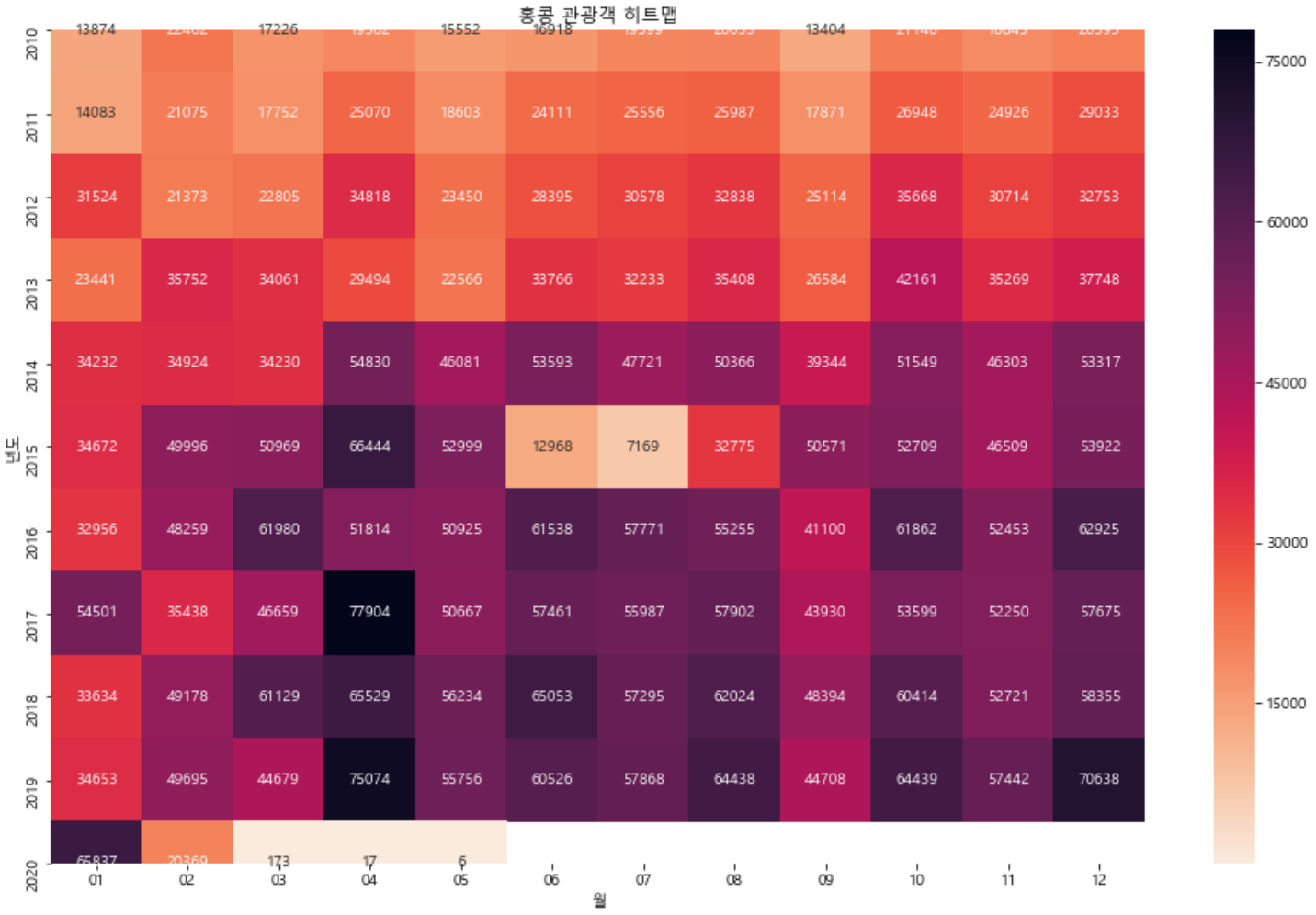
https://matplotlib.org/tutorials/colors/colormaps.html

외국인 관광객 방문 상위 5개국에 대한 히트맵 작성

- 시계열 그래프에서 사용한 상위 5개국 리스트(cntry_list) 재활용

```
for cntry in cntry_list:
    condition = (df['국적'] == cntry)
    df_filter = df[condition]

    df_pivot = df_filter.pivot_table(values = '관광'
                                     , index = '년도'
                                     , columns = '월')

    # 그래프 크기 설정
    plt.figure(figsize = (16, 10))

    # 히트맵 그래프 그리기
    sns.heatmap(df_pivot, annot = True, fmt = '.0f', cmap = 'rocket_r')

    # 그래프 타이틀 달기
    plt.title('{} 관광객 히트맵'.format(cntry))

    # 그래프 표현
    plt.show()
```

외국인 관광객 방문 상위 5개국에 대한 히트맵 표현



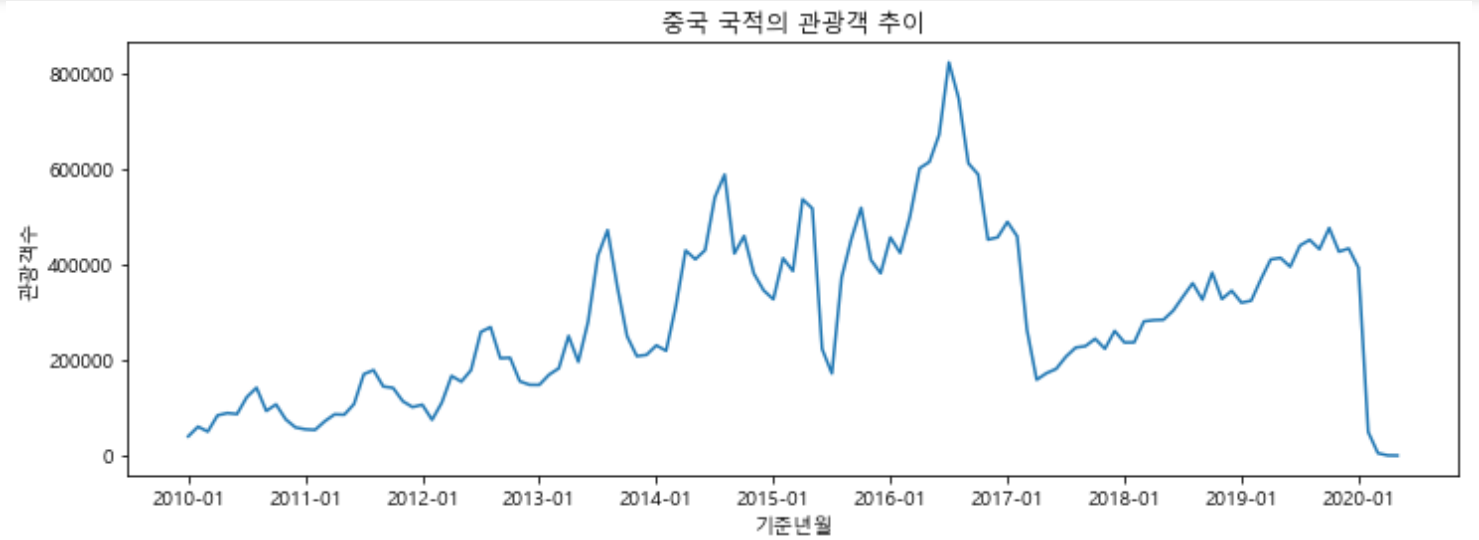
시각화 해석

중국인 관광객 시각화 결과 확인

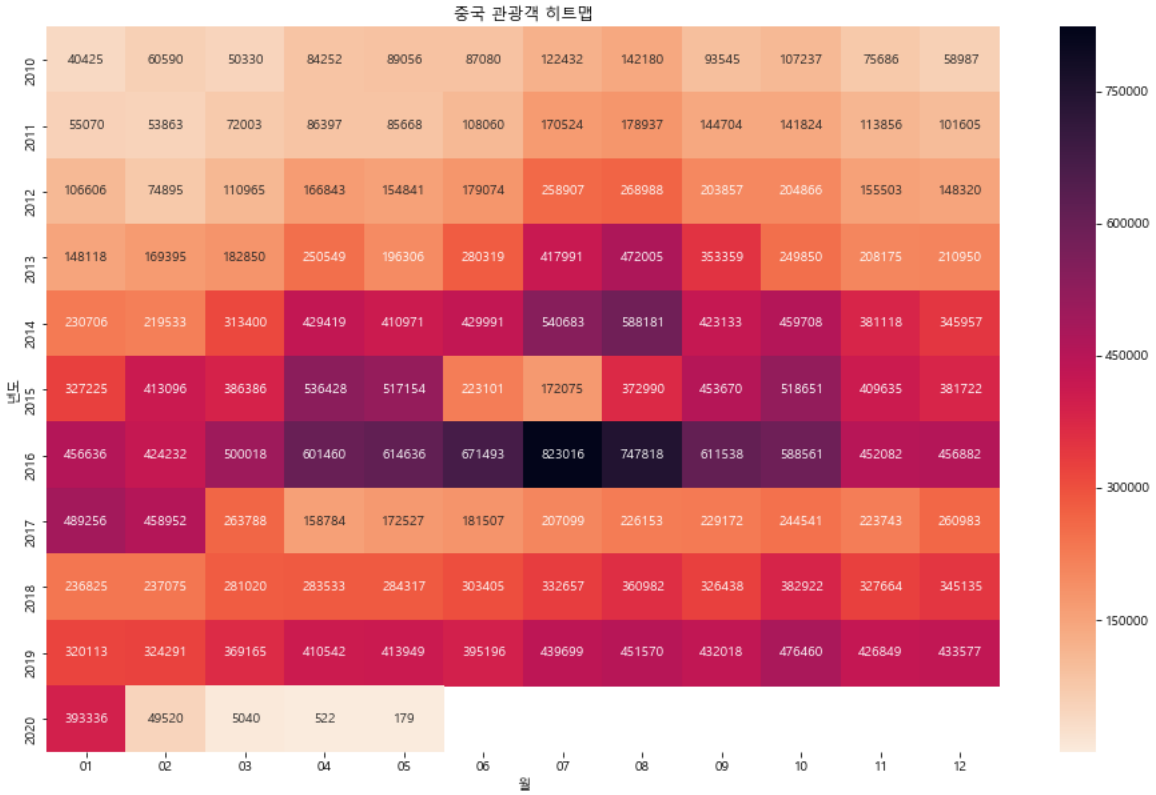
- 계절적인 패턴 존재(seasonality)?
- 추세(trend)가 어떤지?
- 큰 변화(event)가 있는지?
- 변화의 원인이 무엇인지?
- 시계열 그래프부터 확인

시계열 그래프 특징

- (Trend) 2010년 ~ 2016년까지 관광객 수가 꾸준히 증가하는 추세
- (Event) 2017년 초에 관광객 수가 큰 폭으로 감소
- (Trend) 2017년 중순부터 관광객 수가 완만히 증가하는 추세
- (Seasonality) 매년 여름에 관광객 수가 최대값을 가짐
- (Event) 단, 2015년 여름에는 관광객 수가 큰 폭으로 감소
- (Event) 2020년 초에 관광객 수가 0에 가까워질 만큼 급격히 감소



히트맵 그래프 확인

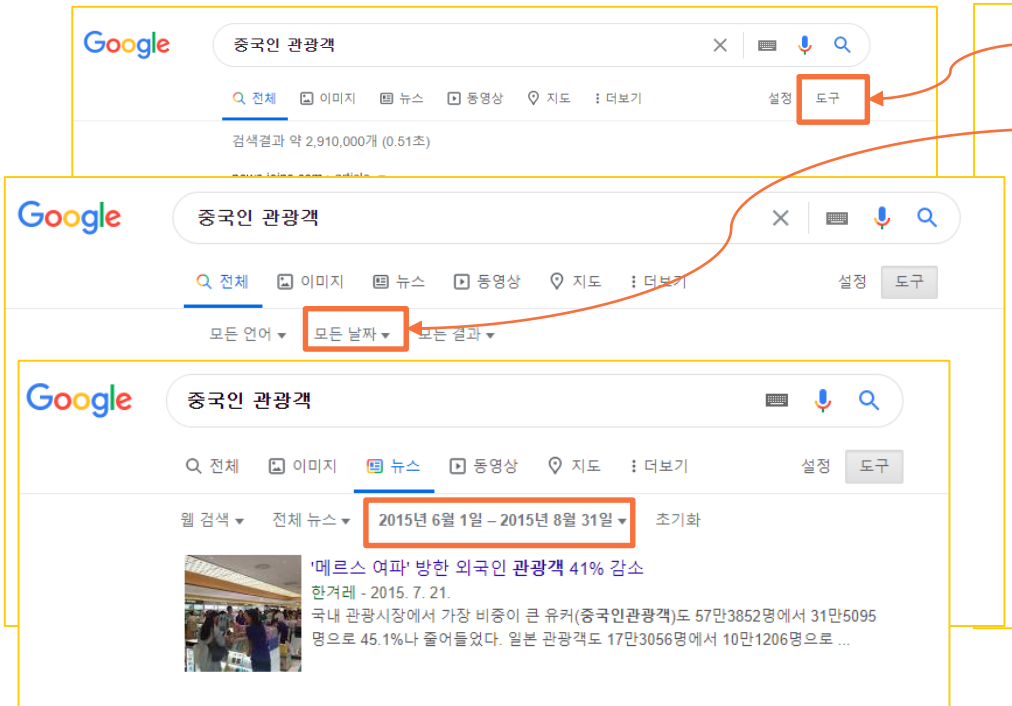


히트맵 그래프 특징

- (Trend) 위에서 아래로 오면서 색깔이 진해지는 것으로 보아 2010년 1월 부터 2017년 3월까지 관광객 수가 꾸준히 증가하는 추세
- (Seasonality) 각 년도를 기준으로 봤을 때 7~8월이 대체로 진한 색깔로서 가장 관광객이 많이 방문하는 패턴을 보이며 , 그 다음으로는 4월, 10월이 높은 패턴이 나타남
- (Event) 2015년 6~8월에 관광객 수가 적음
- (Event) 2017년 3~6월까지 관광객 수가 매우 적음
- (Trend) 2017년 7월부터 2020년 1월까지 관광객 수가 점차 많아짐
- (Event) 2020년 2월부터 관광객 수가 급격히 줄어듬

그래프를 보고 알 수 있는 사실 정리

- **계절적 특징**
  - 중국인 관광객은 여름>봄,가을>겨울 순으로 많이 방문
- **트렌드**
  - 분석을 통해 중국인 관광객의 숫자가 계속해서 증가
- **이벤트**
  - 2015년 여름(6월~8월), 2017년 3월, 2020년 2월에 관광객이 급감한 이유 분석 필요
  - 이벤트 확인을 위해 세 기간 동안 중국인 관광객 키워드로 뉴스 검색



- **검색 결과 페이지에서 도구 선택**
- **모든 날짜 선택**
- **기간설정 : 6/1/2015와 8/31/2015입력 후 실행 => 메르스 여파**
- **기간설정 : 3/1/2017과 3/31/2017 입력 후 실행 => 사드 여파**
- **기간설정 : 2/1/2020과 5/31/2020 입력 후 실행 => COVID-19 여파**
- **메르스는 일시적 감소지만 사드는 2017년 3월부터 아직도 지속되고 있음 => 점차 증가하기는 중 코로나 여파로 급감**