

연도별 / 지역별 외국인 비율과 범죄율의 상관관계

3-C 202344073 강찬휘

[깃허브](#)



시작하기에 앞서

날이 춥더라고요, 감기조심하세요.

코드 전문은 깃허브 페이지의 [project.ipynb](#)파일에서 확인할 수 있습니다.

엑셀/pdf 이미지를 클릭하시면, 해당 파일의 깃허브 주소로 바로 이동하실 수 있습니다.

엑셀 파일에서 가져온 표의 경우 모든 데이터가 들어가있지 않을 수 있습니다. 데이터의 확인을 원하신다면 깃허브 페이지에서 확인 부탁드립니다.

애니메이션이 포함된 표를 html형태로 생성하였으나, ppt/pdf에 직접 삽입이 불가능한 것으로 보입니다. 같이 첨부 드린 파일을 확인해주세요.

목차

- 데이터 취합 및 전처리 과정
 - '내국인 수' 데이터의 전처리 과정
 - '외국인 수' 데이터의 전처리 과정
 - '범죄 수' 데이터의 전처리 과정
 - ★Wide Format과 Long Format★
 - 연도별, 지역별 외국인 비율 얻기
 - 연도별, 지역별 범죄율 얻기
- 시각화 과정
 - 대표 4개 지역의 외국인 비율과 범죄 수의 복합 그래프
 - Plotly를 이용한 전국 범죄율 Choropleth 애니메이션
- 결론

데이터 취합 및 전처리 과정

- '내국인 수' 데이터의 전처리 과정
- '외국인 수' 데이터의 전처리 과정
- '범죄 수' 데이터의 전처리 과정
- ★Wide Format과 Long Format★
- 연도별, 지역별 외국인 비율 얻기
- 연도별, 지역별 범죄율 얻기



‘내국인 수’ 데이터의 전처리 과정

1. 외부에서 초기 데이터 가져오기



[전처리 전 내국인 수.xlsx](#)

※ 매년 말일자 통계 현황		2015년					2016년					2017년					2018년					2019년					2020년										
행정기관코드	행정기관	총인구수	세대수	세대당 인구	남자 인구수	여자 인구수	남여 비율	총인구수	세대수	세대당 인구	남자 인구수	여자 인구수	남여 비율	총인구수	세대수	세대당 인구	남자 인구수	여자 인구수	남여 비율	총인구수	세대수	세대당 인구	남자 인구수	여자 인구수	남여 비율	총인구수	세대수	세대당 인구	남자 인구수	여자 인구수	남여 비율						
1000000000	전국	51,529,338	21,011,152	2.45	25,758,186	25,771,152	1.00	51,696,216	21,294,009	2.43	25,827,594	25,868,622	1.00	51,778,544	21,632,851	2.39	25,855,919	25,922,625	1.00	51,826,059	22,042,947	2.35	25,866,129	25,959,930	1.00	51,849,861	22,481,466	2.31	25,864,816	25,985,045	1.00	51,829,023	23,093,108	2.24	25,841,029	25,987,994	0.99
1100000000	서울특별시	10,022,181	4,189,948	2.39	4,930,943	5,091,238	0.97	9,930,616	4,189,839	2.37	4,876,789	5,053,827	0.96	9,857,426	4,220,082	2.34	4,830,206	5,027,220	0.96	9,765,623	4,263,868	2.29	4,773,899	4,991,724	0.96	9,729,107	4,327,605	2.25	4,744,059	4,985,048	0.95	9,668,465	4,417,954	2.19	4,701,723	4,966,742	0.95
2600000000	부산광역시	3,513,777	1,437,818	2.44	1,735,570	1,778,207	0.98	3,498,529	1,451,270	2.41	1,725,607	1,772,922	0.97	3,470,653	1,467,555	2.36	1,709,508	1,761,145	0.97	3,441,453	1,480,468	2.32	1,692,421	1,749,032	0.97	3,413,841	1,497,908	2.28	1,675,417	1,738,424	0.96	3,391,946	1,530,431	2.22	1,661,000	1,730,946	0.96
2700000000	대구광역시	2,487,829	982,360	2.53	1,237,291	1,250,538	0.99	2,484,557	994,220	2.50	1,234,169	1,250,388	0.99	2,475,231	1,006,753	2.46	1,227,814	1,247,417	0.98	2,461,769	1,021,266	2.41	1,219,342	1,242,427	0.98	2,438,031	1,031,251	2.36	1,205,286	1,232,745	0.98	2,418,346	1,056,627	2.29	1,193,109	1,225,237	0.97
2800000000	인천광역시	2,925,815	1,154,004	2.54	1,469,869	1,455,946	1.01	2,943,069	1,171,399	2.51	1,477,671	1,465,398	1.01	2,948,542	1,188,917	2.48	1,479,597	1,468,945	1.01	2,954,642	1,213,201	2.44	1,481,844	1,472,798	1.01	2,957,026	1,238,641	2.39	1,482,249	1,474,777	1.01	2,942,828	1,267,956	2.32	1,473,943	1,468,885	1.00
2900000000	광주광역시	1,472,199	580,427	2.54	729,662	742,537	0.98	1,469,214	586,464	2.51	728,137	741,077	0.98	1,463,770	592,818	2.47	725,295	738,475	0.98	1,459,336	603,107	2.42	722,581	736,755	0.98	1,456,468	616,485	2.36	720,740	735,728	0.98	1,450,062	633,582	2.29	717,348	732,714	0.98
3000000000	대전광역시	1,518,775	597,008	2.54	759,778	758,997	1.00	1,514,370	606,137	2.50	757,289	757,081	1.00	1,502,227	614,639	2.44	750,969	751,258	1.00	1,489,936	624,965	2.38	744,338	745,598	1.00	1,474,870	635,343	2.32	736,607	738,263	1.00	1,463,882	652,783	2.24	730,699	733,183	1.00
3100000000	울산광역시	1,173,534	451,688	2.60	604,889	568,645	1.06	1,172,304	455,352	2.57	603,797	568,507	1.06	1,165,132	458,547	2.54	599,480	565,652	1.06	1,155,623	461,756	2.50	593,819	561,804	1.06	1,148,019	468,659	2.45	589,712	558,307	1.06	1,136,017	476,893	2.38	583,708	552,309	1.06
3600000000	세종특별자치시	210,884	81,806	2.58	105,752	105,132	1.01	243,048	94,343	2.58	121,505	121,543	1.00	280,100	109,490	2.56	139,734	140,366	1.00	314,126	123,762	2.54	156,831	157,295	1.00	340,575	135,408	2.52	169,845	170,730	0.99	355,831	144,275	2.47	177,568	178,263	1.00
4100000000	경기도	12,522,606	4,885,012	2.56	6,299,812	6,222,794	1.01	12,716,780	5,003,406	2.54	6,395,453	6,321,327	1.01	12,873,895	5,131,379	2.51	6,475,328	6,398,572	1.01	13,077,153	5,306,214	2.46	6,577,501	6,499,652	1.01	13,239,666	5,468,920	2.42	6,659,995	6,579,671	1.01	13,427,014	5,676,401	2.37	6,754,469	6,672,545	1.01
4200000000	강원도	1,549,507	685,005	2.26	781,434	768,073	1.02	1,550,086	692,254	2.24	781,146	769,660	1.01	1,550,142	700,265	2.21	780,635	769,507	1.01	1,543,052	707,245	2.18	776,456	766,596	1.01	1,541,502	719,524	2.14	775,386	766,116	1.01	1,542,840	736,301	2.10	776,505	766,335	1.01
4300000000	충청북도	1,583,952	669,503	2.37	798,896	785,056	1.02	1,591,625	680,960	2.34	803,240	788,385	1.02	1,594,432	692,140	2.30	805,091	789,341	1.02	1,599,252	707,220	2.26	808,971	790,281	1.02	1,600,007	722,123	2.22	810,384	789,623	1.03	1,600,837	745,644	2.15	811,591	789,246	1.03
4400000000	충청남도	2,077,649	885,968	2.35	1,054,439	1,023,210	1.03	2,096,727	902,294	2.32	1,064,765	1,031,962	1.03	2,116,770	923,499	2.29	1,076,270	1,040,500	1.03	2,126,282	943,611	2.25	1,082,009	1,044,273	1.04	2,123,709	959,255	2.21	1,081,938	1,041,771	1.04	2,121,029	983,153	2.16	1,082,634	1,038,395	1.04
4500000000	전라북도	1,869,711	782,986	2.39	930,255	939,456	0.99	1,864,791	790,084	2.36	927,505	937,286	0.99	1,854,607	797,916	2.32	922,307	932,300	0.99	1,836,832	806,235	2.28	913,297	923,535	0.99	1,818,917	816,191	2.23	904,110	914,807	0.99	1,804,104	836,660	2.16	896,874	907,230	0.99
4600000000	전라남도	1,908,996	833,901	2.29	953,881	955,115	1.00	1,903,914	842,688	2.26	951,889	952,025	1.00	1,896,424	851,376	2.23	948,290	948,134	1.00	1,882,970	860,303	2.19	943,000	939,970	1.00	1,868,745	872,628	2.14	937,674	931,071	1.01	1,851,549	893,152	2.07	930,615	920,934	1.01
4700000000	경상북도	2,702,826	1,169,648	2.31	1,357,306	1,345,520	1.01	2,700,398	1,180,643	2.29	1,356,868	1,343,530	1.01	2,691,706	1,193,261	2.26	1,353,164	1,338,542	1.01	2,676,831	1,207,351	2.22	1,346,689	1,330,142	1.01	2,665,836	1,227,548	2.17	1,342,037	1,323,799	1.01	2,639,422	1,254,662	2.10	1,329,211	1,310,211	1.01
4800000000	경상남도	3,364,702	1,367,142	2.46	1,694,981	1,669,721	1.02	3,373,871	1,385,684	2.43	1,699,219	1,674,652	1.01	3,380,404	1,406,011	2.40	1,701,413	1,678,991	1.01	3,373,988	1,425,271	2.37	1,697,412	1,676,576	1.01	3,362,553	1,450,822	2.32	1,692,032	1,670,521	1.01	3,340,216	1,484,601	2.25	1,681,423	1,658,793	1.01
5000000000	제주특별자치도	624,395	256,928	2.43	313,428	310,967	1.01	641,597	266,972	2.40	322,545	319,052	1.01	657,083	278,203	2.36	330,823	326,260	1.01	667,191	287,104	2.32	335,719	331,472	1.01	670,989	293,155	2.29	337,345	333,644	1.01	674,635	302,033	2.23	338,609	336,026	1.01

‘내국인 수’ 데이터의 전처리 과정

2. 필요 없는 요소 제거하기



[내국인 수_v1.xlsx](#)

	2015년	2016년	2017년	2018년	2019년	2020년
전국	51529338	51696216	51778544	51826059	51849861	51829023
서울특별시	10022181	9930616	9857426	9765623	9729107	9668465
부산광역시	3513777	3498529	3470653	3441453	3413841	3391946
대구광역시	2487829	2484557	2475231	2461769	2438031	2418346
인천광역시	2925815	2943069	2948542	2954642	2957026	2942828
광주광역시	1472199	1469214	1463770	1459336	1456468	1450062
대전광역시	1518775	1514370	1502227	1489936	1474870	1463882
울산광역시	1173534	1172304	1165132	1155623	1148019	1136017
경기도	12522606	12716780	12873895	13077153	13239666	13427014
강원도	1549507	1550806	1550142	1543052	1541502	1542840
충청북도	1583952	1591625	1594432	1599252	1600007	1600837
충청남도	2077649	2096727	2116770	2126282	2123709	2121029
전라북도	1869711	1864791	1854607	1836832	1818917	1804104
전라남도	1908996	1903914	1896424	1882970	1868745	1851549
경상북도	2702826	2700398	2691706	2676831	2665836	2639422
경상남도	3364702	3373871	3380404	3373988	3362553	3340216
제주특별자치도	624395	641597	657083	667191	670989	674635

내국인 수의 df 정렬

쓸모 없는 행, 열 1차 제거

```
korean.drop([0, 1, 2], axis=0, inplace=True)
```

```
korean.drop([0], axis=1, inplace=True)
```

행, 열 인덱스 재설정

```
korean = korean.reset_index(drop=True)
```

```
korean.columns = range(korean.shape[1])
```

반복문을 이용해 연도별 인구수만 남기고 나머지 열 제거

```
for i in range(1, len(korean.columns)):
```

```
    if i % 6 == 1:
```

```
        continue
```

```
    korean.drop([i], axis=1, inplace=True)
```

행 인덱스 재설정

```
korean.index = korean[0].str.strip() # 공백 제거 후 행 인덱스로 설정
```

```
korean.drop([0], axis=1, inplace=True)
```

```
korean.columns = [str(i)+'년' for i in range(2015, 2021)]
```

세종시는 데이터가 부족해서 삭제

```
korean.drop('세종특별자치시', axis=0, inplace=True)
```

,이 포함된 문자열을 ,을 제거한 후 int형으로 형 변환

```
for i in range(len(korean)):
```

```
    for j in range(len(korean.columns)):
```

```
        v = korean.iloc[i, j]
```

```
        if pd.isna(v):
```

```
            continue
```

```
        v = str(v).replace(',', '', ' ').strip()
```

```
        if v == '':
```

```
            korean.iloc[i, j] = 0
```

```
        else:
```

```
            korean.iloc[i, j] = int(v)
```

```
korean
```

‘외국인 수’ 데이터의 전처리 과정

1. 외부에서 초기 데이터 가져오기



[전처리 전 외국인 수.xlsx](#)

행정 구역별	연령별	2015			2016			2017			2018			2019			2020		
		전체	여자	남자	전체	여자	남자	전체	여자	남자	전체	여자	남자	전체	여자	남자	전체	여자	남자
전국	계	1,363,712	575,049	788,663	1,413,758	598,291	815,467	1,479,247	633,584	845,663	1,651,561	705,920	945,641	1,778,918	761,510	1,017,408	1,695,643	753,024	942,619
	0~4세	23,273	11,388	11,885	25,775	12,632	13,143	27,022	13,324	13,698	32,275	15,834	16,441	39,818	19,457	20,361	37,164	18,244	18,920
	5~9세	14,901	7,218	7,683	17,566	8,414	9,152	15,152	7,351	7,801	18,792	9,263	9,529	25,432	12,560	12,872	25,679	12,680	12,999
	10~14세	9,432	4,564	4,868	10,534	5,094	5,440	8,816	4,313	4,503	11,860	5,876	5,984	14,979	7,394	7,585	15,368	7,529	7,839
	15~19세	23,534	12,573	10,961	27,311	14,738	12,573	30,563	16,421	14,142	37,692	19,526	18,166	41,728	22,085	19,643	27,559	15,040	12,519
	20~24세	145,172	66,253	78,919	152,146	70,177	81,969	161,704	75,697	86,007	194,511	87,597	106,914	209,915	94,542	115,373	171,488	79,989	91,499
	25~29세	237,392	90,391	147,001	239,214	90,469	148,745	243,718	93,125	150,593	260,938	98,488	162,450	269,082	101,626	167,456	242,981	97,154	145,827
	30~34세	200,576	68,685	131,891	213,718	75,274	138,444	231,716	84,152	147,564	258,920	95,967	162,953	275,001	103,445	171,556	268,499	105,916	162,583
	35~39세	143,273	48,195	95,078	150,460	51,586	98,874	166,309	59,001	107,308	191,494	70,116	121,378	208,488	77,957	130,531	211,257	82,721	128,536
	40~44세	122,906	46,565	76,341	120,621	45,509	75,112	120,666	46,570	74,096	134,146	52,363	81,783	148,113	58,257	89,856	148,367	60,255	88,112
	45~49세	121,216	55,273	65,943	123,306	55,312	67,994	128,530	57,592	70,938	136,565	60,221	76,344	142,085	61,204	80,881	130,370	57,939	72,431
	50~54세	117,471	58,603	58,868	123,372	60,646	62,726	121,787	59,707	62,080	124,949	60,452	64,497	130,662	61,429	69,233	126,371	61,336	65,035
	55~59세	93,754	47,318	46,436	89,458	45,183	44,275	90,876	46,310	44,566	107,372	54,433	52,939	117,773	58,683	59,090	118,725	60,566	58,159
	60~64세	62,516	31,095	31,421	68,827	34,272	34,555	76,527	38,245	38,282	81,013	40,847	40,166	84,414	42,619	41,795	90,080	46,647	43,433
	65~69세	26,148	13,482	12,666	29,011	14,992	14,019	33,057	17,184	15,873	36,694	19,168	17,526	42,017	22,046	19,971	48,414	25,760	22,654
	70~74세	9,809	5,561	4,248	9,929	5,722	4,207	10,764	6,277	4,487	12,022	7,075	4,947	15,163	8,640	6,523	18,137	10,508	7,629
	75세이상	12,339	7,885	4,454	12,510	8,271	4,239	12,040	8,315	3,725	12,318	8,694	3,624	14,248	9,566	4,682	15,184	10,740	4,444

Data 중 일부 발췌

‘외국인 수’ 데이터의 전처리 과정

2. 필요 없는 요소 제거하기



외국인 수_v1.xlsx

	2015년	2016년	2017년	2018년	2019년	2020년
전국	1363712	1413758	1479247	1651561	1778918	1695643
서울특별시	337116	335167	343927	374425	390177	366454
부산광역시	44070	46168	48808	55506	60502	54914
대구광역시	29282	30492	31365	35280	40092	37398
인천광역시	67850	71873	79170	88489	100174	99212
광주광역시	21592	23701	26052	29833	33286	30573
대전광역시	19080	19687	20534	22858	25147	22928
울산광역시	29860	28792	26601	27162	28604	26895
경기도	452632	472699	497089	558197	594795	583462
강원도	18306	19364	20996	24149	27008	24654
충청북도	40758	44901	48237	54835	59783	57094
충청남도	71082	77533	84752	95838	104018	99166
전라북도	29930	31549	33764	39131	43960	42366
전라남도	34611	36630	37212	42520	47778	45236
경상북도	57565	60706	63978	74071	81840	74547
경상남도	90361	92185	91072	97494	105908	96669
제주특별자치도	15146	17476	20400	25287	28017	26831

```
# 외국인 수의 df 정렬

# 쓸모 없는 행, 열 1차 제거
foreigner.drop([0, 1], axis=0, inplace=True)
foreigner.reset_index(drop=True, inplace=True)

# 17개 단위로 반복되는 행 중 첫 번째 행(각 연별 합계)만 남기고 나머지 행 제거
for i in range(foreigner.shape[0]):
    if i % 17 != 0:
        foreigner.drop([i], axis=0, inplace=True)
    continue

# 행 인덱스 재설정
foreigner.reset_index(drop=True, inplace=True)
foreigner.drop([1, 2, 3, 11], axis=0, inplace=True)
foreigner.reset_index(drop=True, inplace=True)

# 열 인덱스 재설정
foreigner.drop([1], axis=1, inplace=True)
foreigner.index = foreigner[0].str.strip() # 공백 제거 후 행 인덱스로 설정
foreigner.drop([0], axis=1, inplace=True)
foreigner.columns = range(foreigner.shape[1])

# 반복문을 이용해 남녀 합만 남기고 나머지 열 제거
for i in range(foreigner.shape[1]):
    if i % 3 == 0:
        continue
    foreigner.drop([i], axis=1, inplace=True)

# 행 인덱스 재설정
foreigner.columns = range(foreigner.shape[1])
foreigner.columns = [str(i)+'년' for i in range(2015, 2021)]
# ,이 포함된 문자열을 ,을 제거한 후 int형으로 형 변환
for i in range(len(foreigner)):
    for j in range(len(foreigner.columns)):
        v = foreigner.iloc[i, j]
        if pd.isna(v):
            continue
        v = str(v).replace(',', '').strip()
        if v == '':
            foreigner.iloc[i, j] = 0
        else:
            foreigner.iloc[i, j] = int(v)

foreigner
```


‘범죄 수’ 데이터의 전처리 과정

1. 외부의 데이터 확인

〈표 - 4〉 지역별 전체범죄 발생건수 및 구성비 추이(2016년~2020년)

지역	2016년		2017년		2018년		2019년		2020년	
	발생건수	구성비	발생건수	구성비	발생건수	구성비	발생건수	구성비	발생건수	구성비
서울	343,104	18.9	320,193	19.3	308,997	20.0	309,269	19.7	296,178	19.2
부산	129,211	7.1	116,449	7.0	110,628	7.2	113,138	7.2	113,652	7.4
대구	86,150	4.7	75,016	4.5	72,341	4.7	75,797	4.8	72,373	4.7
인천	100,387	5.5	91,385	5.5	86,391	5.6	90,608	5.8	88,143	5.7
광주	54,271	3.0	48,213	2.9	43,569	2.8	44,274	2.8	43,517	2.8
대전	49,038	2.7	46,988	2.8	45,527	3.0	45,167	2.9	44,623	2.9
울산	39,799	2.2	34,912	2.1	32,764	2.1	32,780	2.1	31,381	2.0
세종	-	-	-	-	-	-	6,111	0.4	6,276	0.4
경기	476,348	26.2	417,066	25.1	388,938	25.2	394,276	25.1	394,226	25.6
강원	58,814	3.2	53,171	3.2	45,233	2.9	46,096	2.9	44,571	2.9
충북	56,179	3.1	50,387	3.0	48,315	3.1	48,085	3.1	47,864	3.1
충남	72,072	4.0	67,276	4.0	68,577	4.4	62,666	4.0	61,612	4.0
전북	53,375	2.9	48,979	2.9	47,944	3.1	48,252	3.1	47,446	3.1
전남	62,794	3.5	53,892	3.2	50,993	3.3	52,698	3.4	51,519	3.3
경북	88,059	4.8	80,061	4.8	74,356	4.8	77,449	4.9	73,762	4.8
경남	114,410	6.3	100,775	6.1	90,267	5.9	96,212	6.1	97,998	6.4
제주	35,003	1.9	31,368	1.9	27,427	1.8	26,090	1.7	27,129	1.7
계	1,819,014	100.0	1,636,131	100.0	1,542,267	100.0	1,568,968	100.0	1,542,270	100.0

단위: 건, %



[15-19년도간 범죄.pdf](#)



[16-20년도간 범죄.pdf](#)

‘범죄 수’ 데이터의 전처리 과정

2. 하드 코딩으로 data 확보



[범죄 수.xlsx](#)

	2015년	2016년	2017년	2018년	2019년	2020년
전국	1363712	1413758	1479247	1651561	1778918	1695643
서울특별시	337116	335167	343927	374425	390177	366454
부산광역시	44070	46168	48808	55506	60502	54914
대구광역시	29282	30492	31365	35280	40092	37398
인천광역시	67850	71873	79170	88489	100174	99212
광주광역시	21592	23701	26052	29833	33286	30573
대전광역시	19080	19687	20534	22858	25147	22928
울산광역시	29860	28792	26601	27162	28604	26895
경기도	452632	472699	497089	558197	594795	583462
강원도	18306	19364	20996	24149	27008	24654
충청북도	40758	44901	48237	54835	59783	57094
충청남도	71082	77533	84752	95838	104018	99166
전라북도	29930	31549	33764	39131	43960	42366
전라남도	34611	36630	37212	42520	47778	45236
경상북도	57565	60706	63978	74071	81840	74547
경상남도	90361	92185	91072	97494	105908	96669
제주특별자치도	15146	17476	20400	25287	28017	26831

```
# 연도별, 지역별 범죄 수 데이터프레임 생성
# 경찰청에서 배포한 pdf 파일을 보고 하드 코딩
value = {
    '2015년': [1834836, 356575, 140700, 98897, 99915, 61234, 49593, 44569,
    '2016년': [1819014, 343104, 129211, 86150, 100387, 54271, 49038, 39799,
    '2017년': [1636131, 320193, 116449, 75016, 91385, 48213, 46988, 34912,
    '2018년': [1542267, 308997, 110628, 72341, 86391, 43569, 45527, 32764,
    '2019년': [1568968, 309269, 113138, 75797, 90608, 44274, 45167, 32780,
    '2020년': [1542270, 296178, 113652, 72373, 88143, 43517, 44623, 31381,

}

row = [
    '전국', '서울특별시', '부산광역시', '대구광역시', '인천광역시', '광주광역시', '대전광역시',
    '경기도', '강원도', '충청북도', '충청남도', '전라북도', '전라남도', '경상북도', '경상남도',
]

column = ['2015년', '2016년', '2017년', '2018년', '2019년', '2020년']
criminal = pd.DataFrame(value, index=row, columns=column)

criminal
```

★Wide Format과 Long Format★

Wide Format

Team	Points	Assists	Rebounds
A	88	12	22
B	91	17	28
C	99	24	30
D	94	28	31

Long Format

Team	Variable	Value
A	Points	88
A	Assists	12
A	Rebounds	22
B	Points	91
B	Assists	17
B	Rebounds	28
C	Points	99
C	Assists	24
C	Rebounds	30
D	Points	94
D	Assists	28
D	Rebounds	31

Wide Format의 특징

- 요약 통계를 제공할 때 유리하다.
- 구조가 직관적이고 이해하기 쉽다.

Long Format의 특징

- 시각화 라이브러리에 적합하다.
- 사람이 보기 불편하다.

연도별, 지역별 외국인 비율 얻기

1. 내국인 data와 외국인 data로 외국인 비율을 갖는 df생성

```
# 연도별, 지역별 외국인 비율
value = {
    '2015년': [i for i in (foreigner['2015년'] / (korean['2015년'] + foreigner['2015년']) * 100)],
    '2016년': [i for i in (foreigner['2016년'] / (korean['2016년'] + foreigner['2016년']) * 100)],
    '2017년': [i for i in (foreigner['2017년'] / (korean['2017년'] + foreigner['2017년']) * 100)],
    '2018년': [i for i in (foreigner['2018년'] / (korean['2018년'] + foreigner['2018년']) * 100)],
    '2019년': [i for i in (foreigner['2019년'] / (korean['2019년'] + foreigner['2019년']) * 100)],
    '2020년': [i for i in (foreigner['2020년'] / (korean['2020년'] + foreigner['2020년']) * 100)]
}

row = [
    '전국', '서울특별시', '부산광역시', '대구광역시', '인천광역시', '광주광역시', '대전광역시', '울산광역시',
    '경기도', '강원도', '충청북도', '충청남도', '전라북도', '전라남도', '경상북도', '경상남도', '제주특별자치도'
]

column = ['2015년', '2016년', '2017년', '2018년', '2019년', '2020년']

foreignerPercent = pd.DataFrame(value, index=row, columns=column)
```

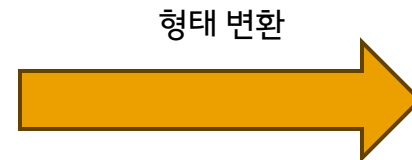
연도별, 지역별 외국인 비율 얻기

2. Long Format으로 형태 변환하기

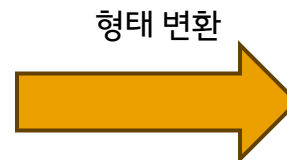
```
# 1. 기존 행 인덱스를 'region' 열로 변환
# 2. melt로 long-format 변환
fp_long = foreignerPercent.reset_index().rename(columns={'index':'region'})
fp_long = fp_long.melt(id_vars='region', var_name='year', value_name='foreigner_pct')

# 연도 숫자 추출(정렬용)
fp_long['year_num'] = fp_long['year'].astype(str).str.extract(r'(\d{4})').astype(int)
```

	2015년	2016년	2017년	2018년	2019년	2020년
전국	2.578244	2.661944	2.777522	3.088322	3.317096	3.167966
서울특별시	3.254236	3.264895	3.371386	3.692537	3.855777	3.651788
부산광역시	1.23867	1.302453	1.386803	1.587265	1.741394	1.59316
대구광역시	1.163318	1.212382	1.251299	1.412868	1.617837	1.522879
인천광역시	2.266453	2.383893	2.614846	2.907827	3.276658	3.261364
광주광역시	1.44545	1.587565	1.748665	2.003332	2.234329	2.064857
대전광역시	1.240689	1.283329	1.348472	1.510979	1.676448	1.542093
울산광역시	2.481315	2.397144	2.232127	2.296444	2.431025	2.312729
경기도	3.488429	3.58391	3.71767	4.093749	4.299372	4.16447
강원도	1.167614	1.233242	1.336356	1.5409	1.721889	1.572829
충청북도	2.508632	2.743678	2.936502	3.315122	3.601841	3.44369
충청남도	3.308092	3.565949	3.8497	4.312908	4.669244	4.466545
전라북도	1.575561	1.663678	1.787996	2.085915	2.359791	2.294432
전라남도	1.780761	1.887615	1.924457	2.208269	2.492952	2.384878
경상북도	2.085393	2.198613	2.321674	2.692608	2.978517	2.746789
경상남도	2.615321	2.659651	2.623437	2.808426	3.053458	2.812692
제주특별자치도	2.368261	2.651603	3.011146	3.651668	4.00812	3.824989



외국인 비율
Wide.xlsx



외국인 비율
Long.xlsx

	region	year	foreigner_pct	year_num
0	전국	2015년	2.578244	2015
1	서울특별시	2015년	3.254236	2015
2	부산광역시	2015년	1.23867	2015
3	대구광역시	2015년	1.163318	2015
4	인천광역시	2015년	2.266453	2015
5	광주광역시	2015년	1.44545	2015
6	대전광역시	2015년	1.240689	2015
7	울산광역시	2015년	2.481315	2015
8	경기도	2015년	3.488429	2015
9	강원도	2015년	1.167614	2015
10	충청북도	2015년	2.508632	2015
11	충청남도	2015년	3.308092	2015
12	전라북도	2015년	1.575561	2015
13	전라남도	2015년	1.780761	2015
14	경상북도	2015년	2.085393	2015
15	경상남도	2015년	2.615321	2015
16	제주특별자치도	2015년	2.368261	2015
17	전국	2016년	2.661944	2016
18	서울특별시	2016년	3.264895	2016
19	부산광역시	2016년	1.302453	2016
20	대구광역시	2016년	1.212382	2016

이하 생략

연도별, 지역별 범죄율 얻기

1. 범죄 수 data와 내국인, 외국인 data로 범죄율을 갖는 df생성

```
row = [
    '전국', '서울특별시', '부산광역시', '대구광역시', '인천광역시', '광주광역시', '대전광역시', '울산광역시',
    '경기도', '강원도', '충청북도', '충청남도', '전라북도', '전라남도', '경상북도', '경상남도', '제주특별자치도'
]

column = ['2015년', '2016년', '2017년', '2018년', '2019년', '2020년']

# 연도별, 지역별 범죄율
value = {
    '2015년': [i for i in (criminal['2015년'] / (korean['2015년'] + foreigner['2015년'])) * 100],
    '2016년': [i for i in (criminal['2016년'] / (korean['2016년'] + foreigner['2016년'])) * 100],
    '2017년': [i for i in (criminal['2017년'] / (korean['2017년'] + foreigner['2017년'])) * 100],
    '2018년': [i for i in (criminal['2018년'] / (korean['2018년'] + foreigner['2018년'])) * 100],
    '2019년': [i for i in (criminal['2019년'] / (korean['2019년'] + foreigner['2019년'])) * 100],
    '2020년': [i for i in (criminal['2020년'] / (korean['2020년'] + foreigner['2020년'])) * 100]
}

criminalPercent = pd.DataFrame(value, index=row, columns=column)
```


연도별, 지역별 범죄율 얻기

2. Long Format으로 형태 변환하기

```
# 1. 기존 행 인덱스를 'region' 열로 변환
# 2. melt로 long-format 변환
cp_long = criminalPercent.reset_index().rename(columns={'index':'region'})
cp_long = cp_long.melt(id_vars='region', var_name='year', value_name='criminal_percent')

# 연도 숫자 추출(정렬용)
cp_long['year_num'] = cp_long['year'].astype(str).str.extract(r'(\d{4})').astype(int)
```

	2015년	2016년	2017년	2018년	2019년	2020년
전국	3.468955	3.424995	3.072097	2.883948	2.925608	2.881419
서울특별시	3.442077	3.34221	3.138731	3.047293	3.056234	2.951474
부산광역시	3.954639	3.645192	3.308717	3.163549	3.256385	3.297262
대구광역시	3.928988	3.425381	2.992744	2.89706	3.058646	2.947091
인천광역시	3.337548	3.329649	3.018286	2.838885	2.963758	2.897496
광주광역시	4.099235	3.635237	3.236158	2.925726	2.9719	2.939077
대전광역시	3.224816	3.196622	3.085711	3.009465	3.011099	3.001258
울산광역시	3.703608	3.313557	2.929515	2.770072	2.785939	2.698484
경기도	3.320093	3.611576	3.119187	2.852424	2.849956	2.813794
강원도	3.631045	3.745709	3.384235	2.886228	2.93884	2.843456
충청북도	3.30539	3.43282	3.067386	2.920947	2.897053	2.886972
충청남도	3.416389	3.314783	3.055886	3.086107	2.813002	2.775072
전라북도	3.01857	2.814632	2.593717	2.555701	2.590187	2.569552
전라남도	3.428831	3.235897	2.787081	2.648313	2.749667	2.716122
경상북도	3.328369	3.189268	2.905304	2.702968	2.818709	2.717865
경상남도	3.439677	3.30087	2.902944	2.600244	2.77391	2.851361
제주특별자치도	5.288011	5.310944	4.630079	3.960703	3.732443	3.867472



범죄율
Wide.xlsx

형태 변환



범죄율
Long.xlsx

형태 변환



	region	year	criminal_percent	year_num
0	전국	2015년	3.468954806	2015
1	서울특별시	2015년	3.442077199	2015
2	부산광역시	2015년	3.954638859	2015
3	대구광역시	2015년	3.928988432	2015
4	인천광역시	2015년	3.337547788	2015
5	광주광역시	2015년	4.099234766	2015
6	대전광역시	2015년	3.224816384	2015
7	울산광역시	2015년	3.703608295	2015
8	경기도	2015년	3.320093242	2015
9	강원도	2015년	3.631045284	2015
10	충청북도	2015년	3.305389885	2015
11	충청남도	2015년	3.416388557	2015
12	전라북도	2015년	3.018570351	2015
13	전라남도	2015년	3.428831034	2015
14	경상북도	2015년	3.328369061	2015
15	경상남도	2015년	3.439676787	2015
16	제주특별자치도	2015년	5.288011246	2015
17	전국	2016년	3.424995087	2016
18	서울특별시	2016년	3.342209747	2016
19	부산광역시	2016년	3.645191676	2016
20	대구광역시	2016년	3.425380579	2016

이하 생략

시각화 과정

- 연도별 지역별 외국인 비율
 - 전국의 외국인 비율
 - 인구 규모 기준 외국인 비율
- 연도별 지역별 범죄 수
 - 전국의 범죄 수
 - 인구 규모 기준 범죄 수
- 대표 4개 지역의 외국인 비율과 범죄 수의 복합 그래프
- Plotly를 이용한 전국 범죄율 Choropleth 애니메이션



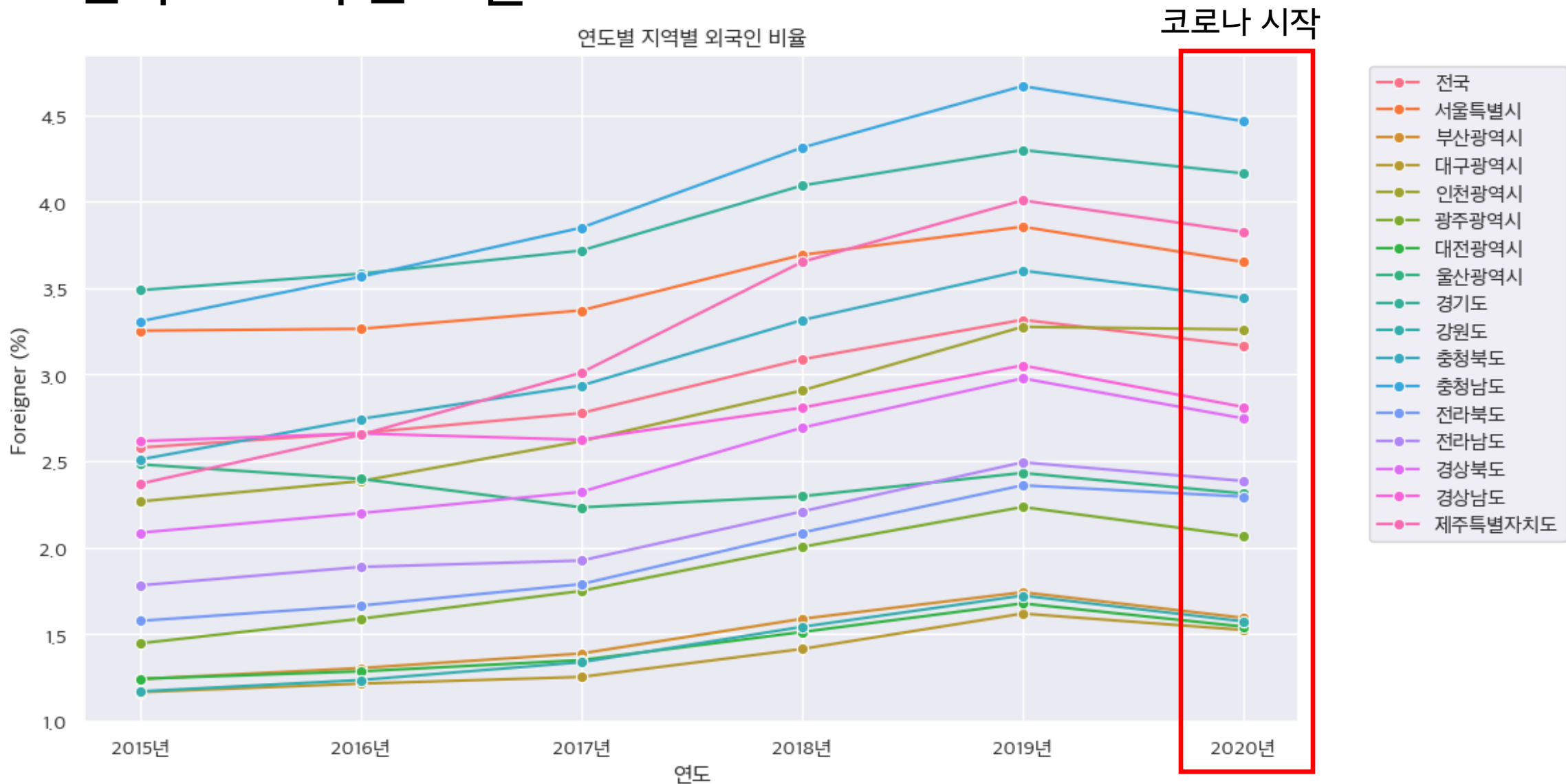
연도별, 지역별 외국인 비율

1. 전국의 외국인 비율

```
# Plot: x=year_num, y=foreigner_pct, hue=region
plt.figure(figsize=(12,6))
sns.lineplot(data=fp_long, x='year_num', y='foreigner_pct', hue='region', marker='o')
plt.xticks(fp_long['year_num'].sort_values().unique(), [str(y)+'년' for y in fp_long['year_num']])
plt.xlabel('연도')
plt.ylabel('Foreigner (%)')
plt.title('연도별 지역별 외국인 비율')
plt.legend(bbox_to_anchor=(1.05,1), loc='upper left')
plt.tight_layout()
plt.show()
```

연도별, 지역별 외국인 비율

1. 전국의 외국인 비율



연도별, 지역별 외국인 비율

2. 인구 규모 기준 외국인 비율

```
# Plot: x=year_num, y=foreigner_pct, hue=region
```

```
fig = plt.figure(figsize=(20,10))
```

```
ax1 = fig.add_subplot(2,2,1)
```

```
ax2 = fig.add_subplot(2,2,2)
```

```
ax3 = fig.add_subplot(2,2,3)
```

```
ax4 = fig.add_subplot(2,2,4)
```

```
# 첫 번째 그래프
```

```
regions_to_plot = ['서울특별시', '경기도', '부산광역시', '인천광역시']
```

```
cm_long1 = cm_long[cm_long['region'].isin(regions_to_plot)]
```

```
sns.lineplot(data=cm_long1, x='year_num', y='foreigner_pct', hue='region', marker='o', ax=ax1)
```

```
ax1.set_title('대도시군의 외국인 비율 변화')
```

```
ax1.set_xlabel('연도')
```

```
ax1.set_ylabel('외국인 비율 (%)')
```

```
ax1.legend(loc='upper left')
```

```
# 두 번째 그래프
```

```
regions_to_plot = ['대구광역시', '광주광역시', '대전광역시', '울산광역시']
```

```
cm_long2 = cm_long[cm_long['region'].isin(regions_to_plot)]
```

```
sns.lineplot(data=cm_long2, x='year_num', y='foreigner_pct', hue='region', marker='o', ax=ax2)
```

```
ax2.set_title('중위권 도시군의 외국인 비율 변화')
```

```
ax2.set_xlabel('연도')
```

```
ax2.set_ylabel('외국인 비율 (%)')
```

```
ax2.legend(loc='upper left')
```

```
# 세 번째 그래프
```

```
regions_to_plot = ['강원도', '충청북도', '충청남도']
```

```
cm_long3 = cm_long[cm_long['region'].isin(regions_to_plot)]
```

```
sns.lineplot(data=cm_long3, x='year_num', y='foreigner_pct', hue='region', marker='o', ax=ax3)
```

```
ax3.set_title('저인구 도시군의 외국인 비율 변화1')
```

```
ax3.set_xlabel('연도')
```

```
ax3.set_ylabel('외국인 비율 (%)')
```

```
ax3.legend(loc='upper left')
```

```
# 네 번째 그래프
```

```
regions_to_plot = ['전라북도', '전라남도', '경상북도', '경상남도', '제주특별자치도']
```

```
cm_long4 = cm_long[cm_long['region'].isin(regions_to_plot)]
```

```
sns.lineplot(data=cm_long4, x='year_num', y='foreigner_pct', hue='region', marker='o', ax=ax4)
```

```
ax4.set_title('저인구 도시군의 외국인 비율 변화2')
```

```
ax4.set_xlabel('연도')
```

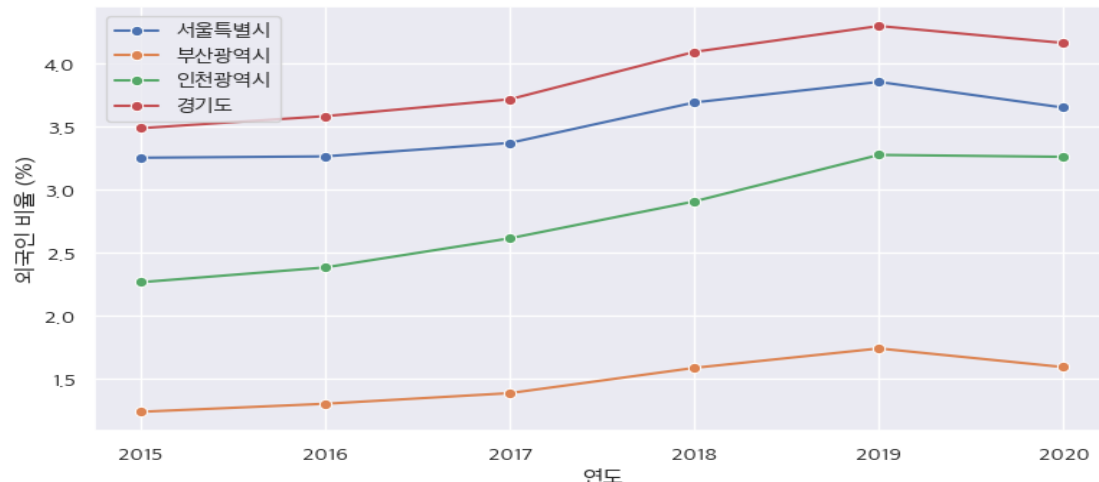
```
ax4.set_ylabel('외국인 비율 (%)')
```

```
ax4.legend(loc='upper left')
```

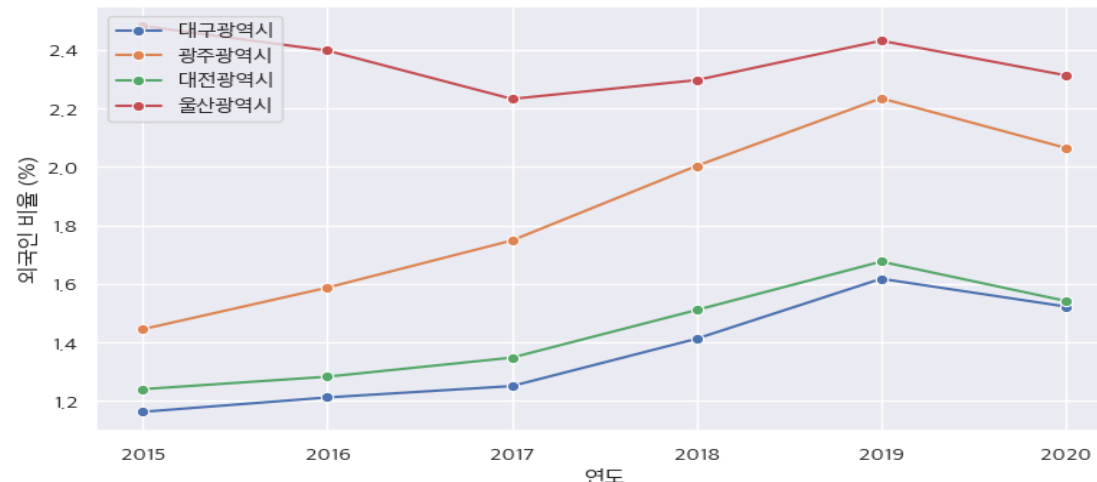
연도별, 지역별 외국인 비율

2. 인구 규모 기준 외국인 비율

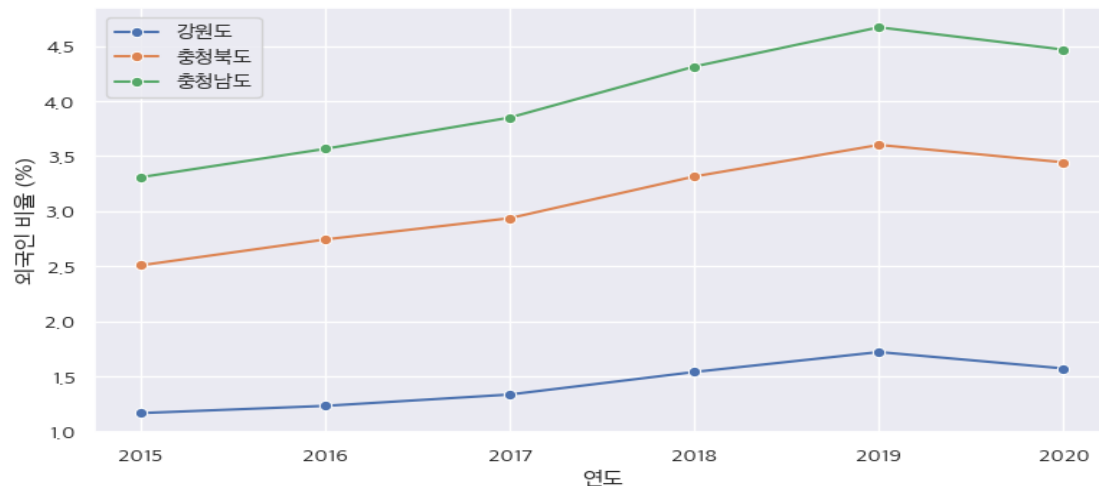
대도시군의 외국인 비율 변화



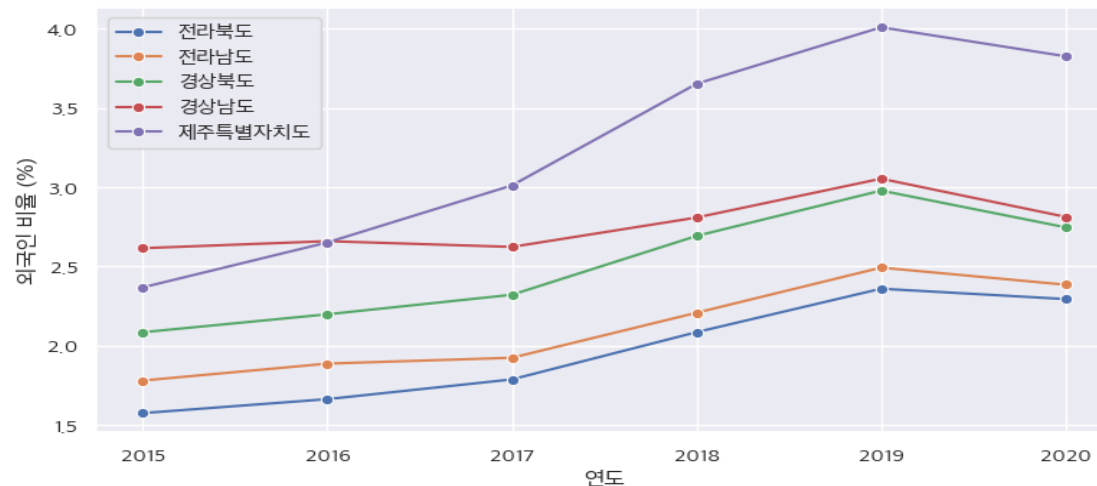
중위권 도시군의 외국인 비율 변화



저인구 도시군의 외국인 비율 변화1



저인구 도시군의 외국인 비율 변화2



연도별, 지역별 범죄 수

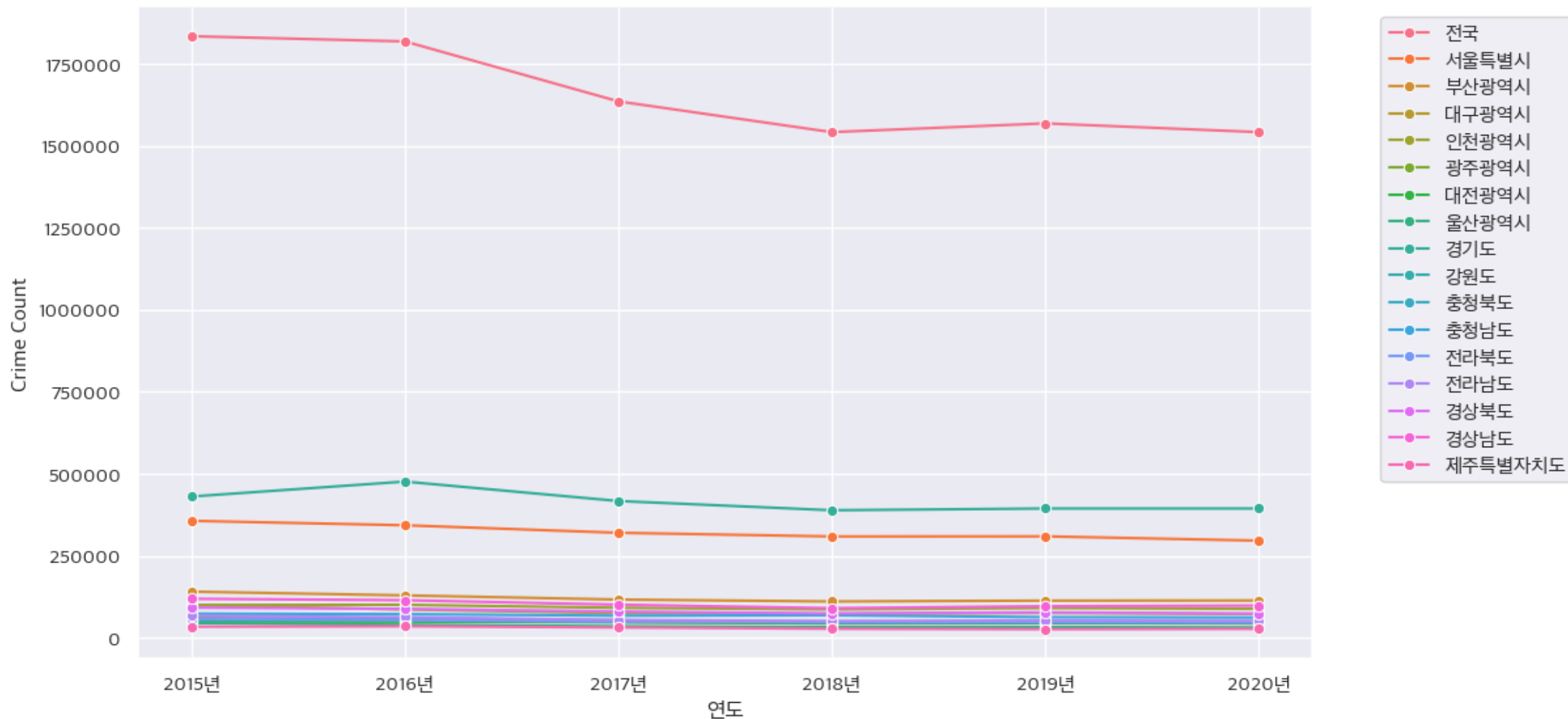
1. 전국의 범죄 수

```
# Plot: x=year_num, y=crime_count, hue=region
plt.figure(figsize=(12,6))
sns.lineplot(data=cm_long, x='year_num', y='crime_count', hue='region', marker='o')
plt.xticks(cm_long['year_num'].sort_values().unique(), [str(y)+'년' for y in cm_long['year_num']])
plt.xlabel('연도')
plt.ylabel('Crime Count')
plt.title('연도별 지역별 범죄 수')
plt.legend(bbox_to_anchor=(1.05,1), loc='upper left')
plt.tight_layout()
plt.ticklabel_format(style='plain', axis='y')
```

연도별, 지역별 범죄 수

1. 전국의 범죄 수

연도별 지역별 범죄 수



연도별, 지역별 범죄 수

2. 인구 규모 기준 범죄 수

```
# Plot: x=year_num, y=crime_count, hue=region
fig = plt.figure(figsize=(20,10))
ax1 = fig.add_subplot(2,2,1)
ax2 = fig.add_subplot(2,2,2)
ax3 = fig.add_subplot(2,2,3)
ax4 = fig.add_subplot(2,2,4)

# 첫 번째 그래프
regions_to_plot = ['서울특별시', '경기도', '부산광역시', '인천광역시']
cm_long1 = cm_long[cm_long['region'].isin(regions_to_plot)]
sns.lineplot(data=cm_long1, x='year_num', y='crime_count', hue='region', marker='o', ax=ax1)
ax1.set_title('대도시군의 범죄 수 변화')
ax1.set_xlabel('연도')
ax1.set_ylabel('범죄 수')
ax1.legend(loc='upper left')

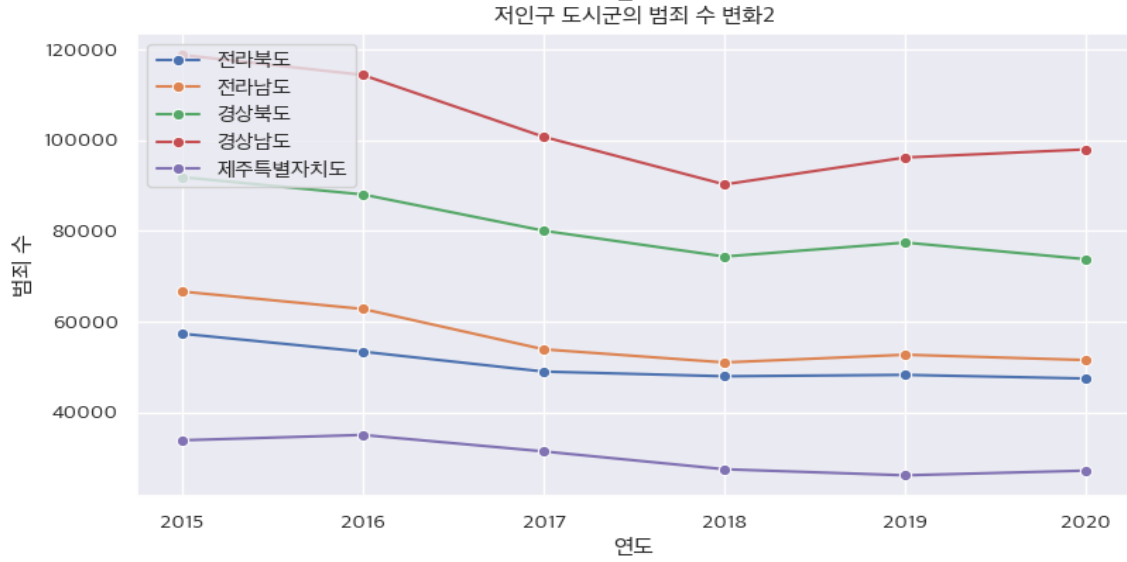
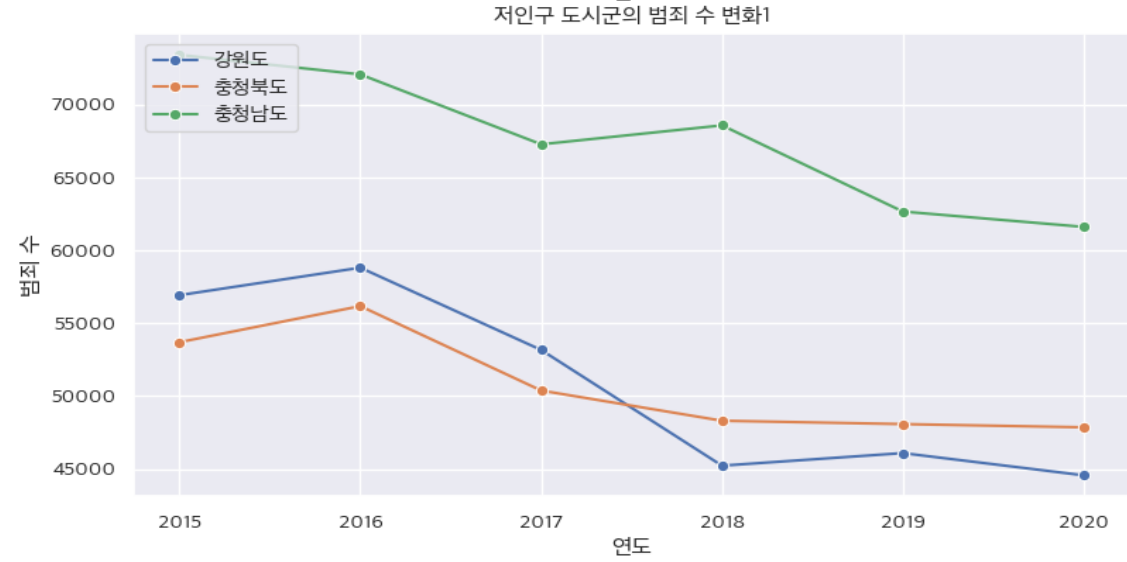
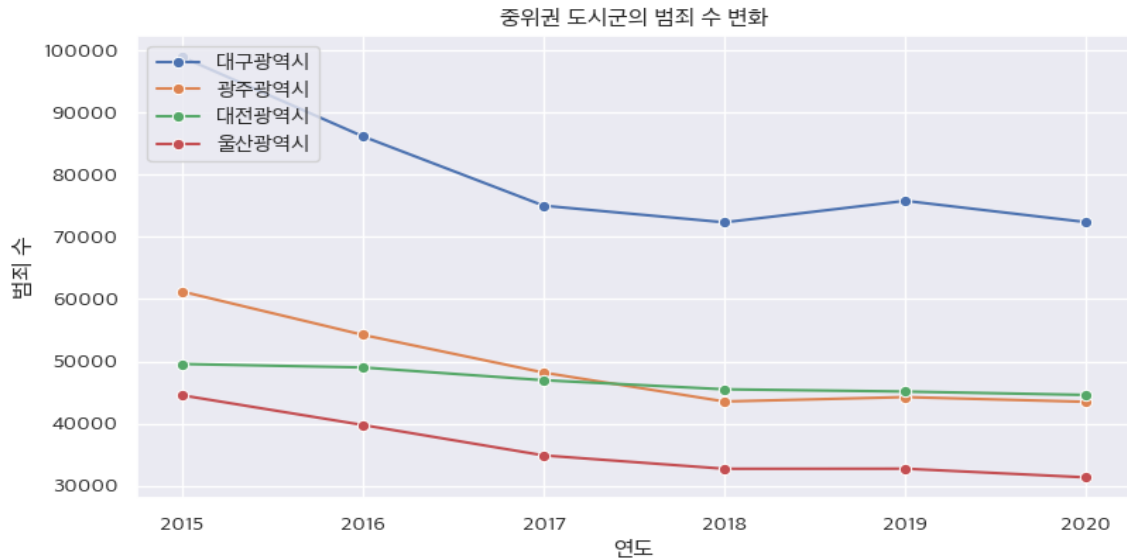
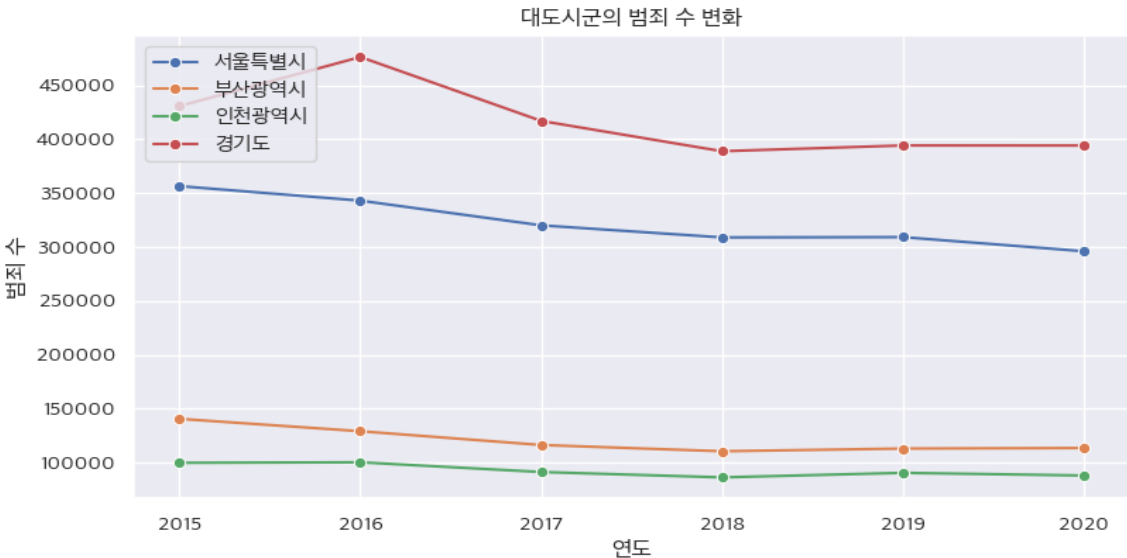
# 두 번째 그래프
regions_to_plot = ['대구광역시', '광주광역시', '대전광역시', '울산광역시']
cm_long2 = cm_long[cm_long['region'].isin(regions_to_plot)]
sns.lineplot(data=cm_long2, x='year_num', y='crime_count', hue='region', marker='o', ax=ax2)
ax2.set_title('중위권 도시군의 범죄 수 변화')
ax2.set_xlabel('연도')
ax2.set_ylabel('범죄 수')
ax2.legend(loc='upper left')

# 세 번째 그래프
regions_to_plot = ['강원도', '충청북도', '충청남도']
cm_long3 = cm_long[cm_long['region'].isin(regions_to_plot)]
sns.lineplot(data=cm_long3, x='year_num', y='crime_count', hue='region', marker='o', ax=ax3)
ax3.set_title('저인구 도시군의 범죄 수 변화1')
ax3.set_xlabel('연도')
ax3.set_ylabel('범죄 수')
ax3.legend(loc='upper left')

# 네 번째 그래프
regions_to_plot = ['전라북도', '전라남도', '경상북도', '경상남도', '제주특별자치도']
cm_long4 = cm_long[cm_long['region'].isin(regions_to_plot)]
sns.lineplot(data=cm_long4, x='year_num', y='crime_count', hue='region', marker='o', ax=ax4)
ax4.set_title('저인구 도시군의 범죄 수 변화2')
ax4.set_xlabel('연도')
ax4.set_ylabel('범죄 수')
ax4.legend(loc='upper left')
```

연도별, 지역별 범죄 수

2. 인구 규모 기준 범죄 수



대표 4개 지역의 외국인 비율과 범죄 수의 복합 그래프

Q. 왜 모든 시/도를 대상으로 분석하지 않았는가?

A. 상기의 외국인 비율과 범죄 수의 연도별 추이를 보면 그 형태가 대동소이하여 대표 4개의 지역으로 좁혔다.

Q. 4개의 지역의 선정 기준은 어떻게 되는가?

A. 데이터 변동성이 안정적이며, 신뢰 있는 데이터를 얻기 위해서는 규모가 중요하다. 때문에, 상위 인구수 4개의 시/도를 기준으로 하였다.

```
# --- subplot 만들기 ---
fig, axes = plt.subplots(2, 2, figsize=(20, 12))
axes = axes.flatten()

for i, region in enumerate(regions):
    ax = axes[i]

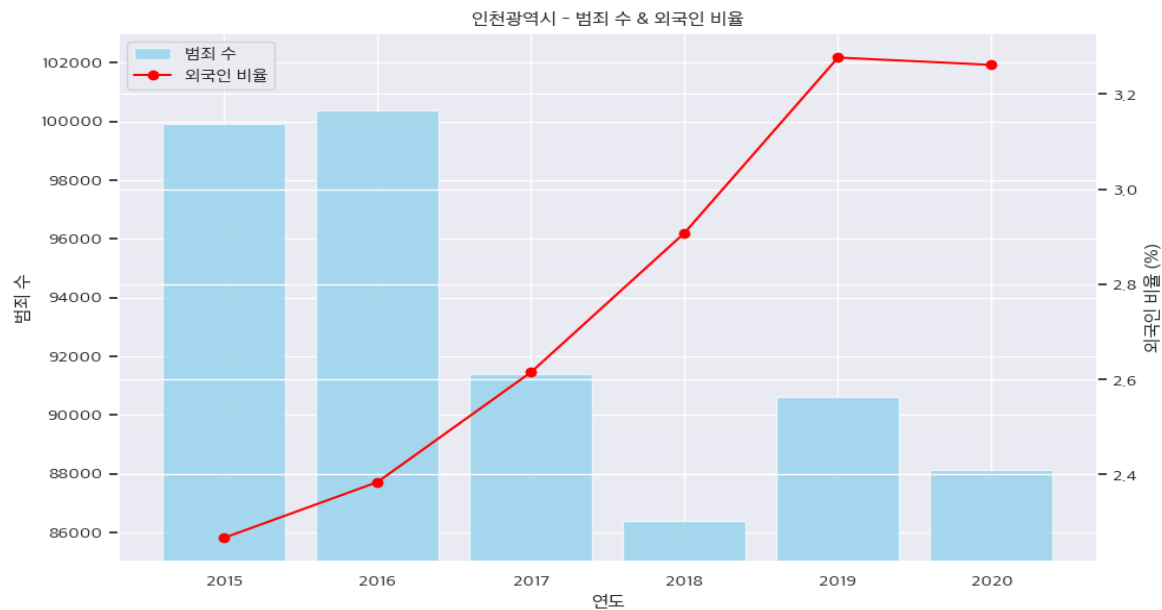
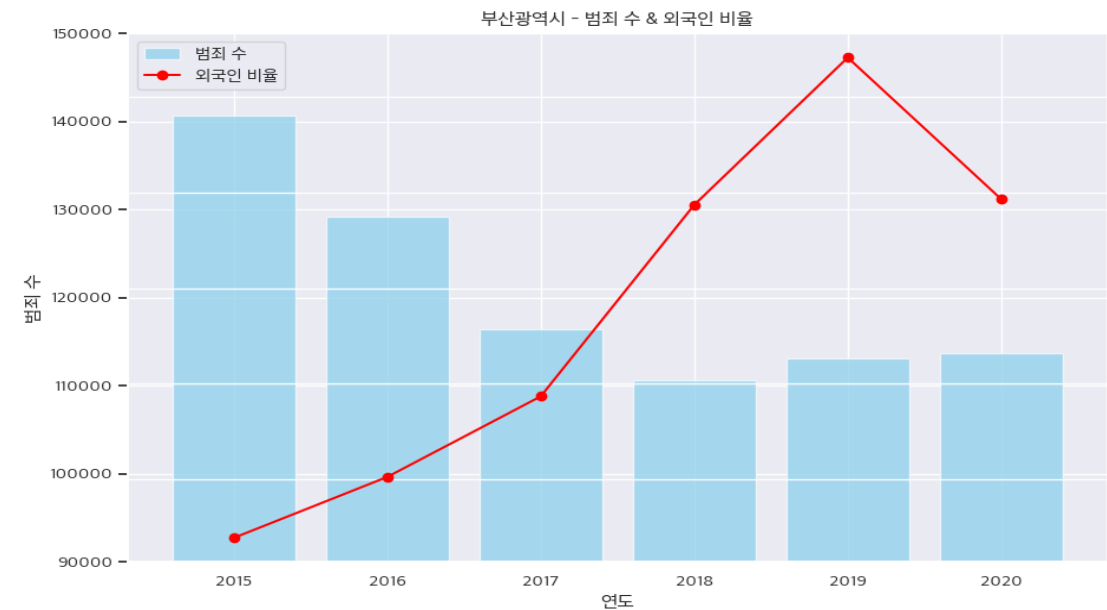
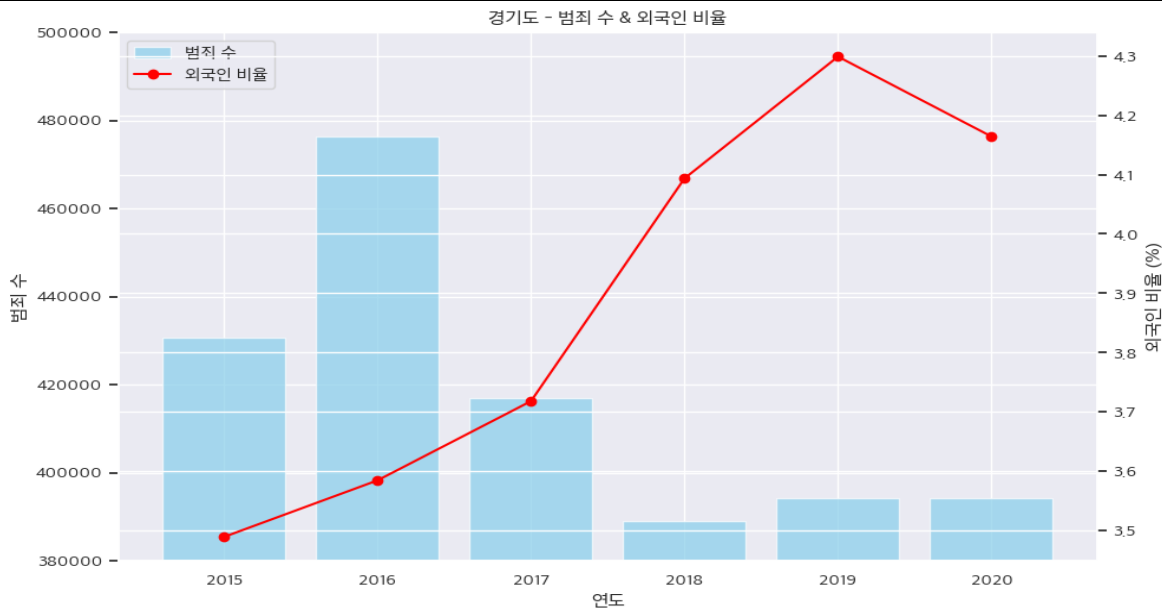
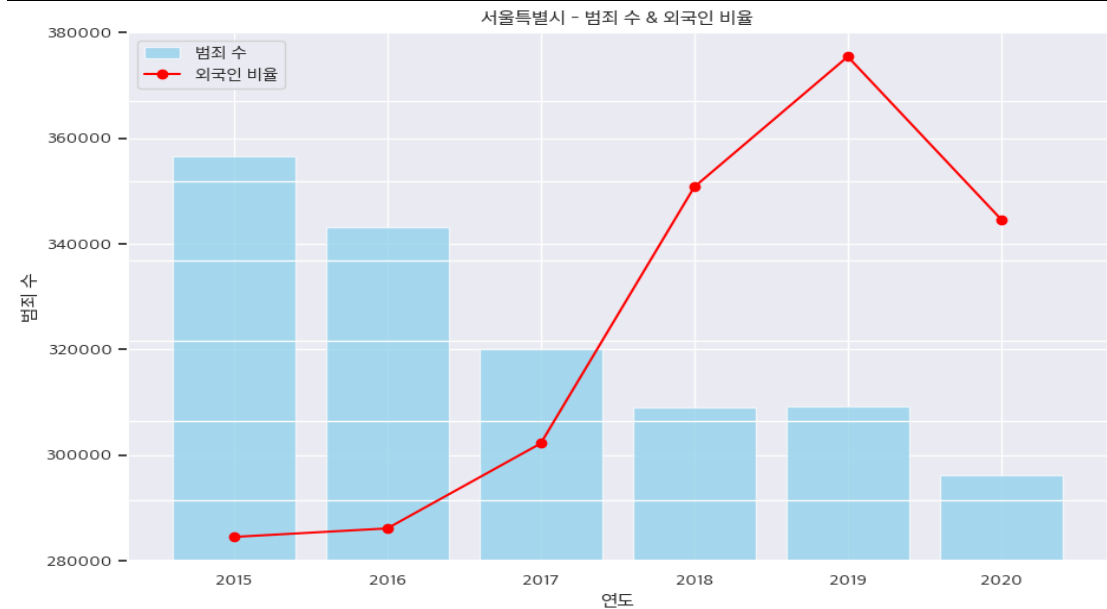
    # 지역별 데이터 추출
    c = crime_long[crime_long['region'] == region]
    f = fp_long[fp_long['region'] == region]

    # 막대그래프(범죄 수)
    ax.bar(c['year_num'], c['crime'], color='skyblue', alpha=0.7, label='범죄 수')
    ax.set_xlabel('연도')
    ax.set_ylabel('범죄 수')
    ax.set_title(f'{region} - 범죄 수 & 외국인 비율')
    if i==0: # 각 그래프마다 y축 범위 지정
        ax.set_ylim(280000, 380000)
    elif i==1:
        ax.set_ylim(380000, 500000)
    elif i==2:
        ax.set_ylim(90000, 150000)
    elif i==3:
        ax.set_ylim(85000, 101000)

    # 선그래프(외국인 비율) - 보조축
    ax2 = ax.twinx()
    ax2.plot(f['year_num'], f['pct'], marker='o', color='red', label='외국인 비율')
    ax2.set_ylabel('외국인 비율 (%)')

    # 범례 합치기
    handles1, labels1 = ax.get_legend_handles_labels()
    handles2, labels2 = ax2.get_legend_handles_labels()
    ax.legend(handles1 + handles2, labels1 + labels2, loc='upper left')
```

대표 4개 지역의 외국인 비율과 범죄 수의 복합 그래프



Plotly를 이용한 전국 범죄율 Choropleth 애니메이션

```
# 인구 대비 범죄 수(범죄율)로 지도 만들고, 호버했을 때 외국인 비율이 보이게

cp_df = cp_long.query("region != '전국']").copy()
fp_df = fp_long.query("region != '전국']").copy()
# 범죄율과 외국인 비율을 merge
df = pd.merge(cp_df, fp_df[['region', 'year', 'pct']], on=['region', 'year'])
df.rename(columns={'pct': 'foreign_ratio'}, inplace=True)

# korea_geojson = 'data/skorea-provinces-2018-geo.json'
# 시/도로 구분된 Geo-json file을 open
with open("data/skorea-provinces-2018-geo.json", "r", encoding="utf-8") as f:
    korea_geojson = json.load(f)

fig = px.choropleth(
    df,
    geojson=korea_geojson,      # 시/도 경계 GeoJSON
    locations='region',         # 데이터와 GeoJSON 매칭용 컬럼
    featureidkey='properties.name',
    color='criminal_percent',    # 색으로 표시
    hover_data=['foreign_ratio'],
    hover_name='region',
    animation_frame='year',
    range_color=[2, 4]         # 직접 범위 지정
)

fig.update_geos(fitbounds="locations", visible=False)
```

Plotly를 이용한 전국 범죄율 Choropleth 애니메이션

같이 첨부 드린 'korea_crime_animation.html' 파일을 확인해주세요.
감사합니다. 😊

결론



결론

서울특별시 - 범죄 수 & 외국인 비율

관련 없습니다.

복합 그래프를 그리기도 전에, 이미 관련이 없다는 것을 알았습니다.
대부분의 지역이 외국인 비율은 상승하는데 반해, 범죄 수는 하향하고 있었으니까요.

그럼에도 몇몇 분들은 반중 감정을 비롯한, 외국인들에 대한 감정은 마냥 긍정적이지만은 않습니다. 재미없는 말을 덧붙이자면, 모두가 노력해야 하겠죠

프로젝트를 진행하면서 아쉬웠던 점은 아래와 같습니다.

1. 화재로 인해 최신 데이터를 얻지 못했던 것
 - 1-1. 최신 데이터를 얻었어도 코로나로 인해 적합한 데이터일지는 미지수라는 점
2. 연도별 애니메이션을 ppt/pdf에 직접 삽입하고 싶었는데, 방법을 못 찾은 점

감사합니다.

