

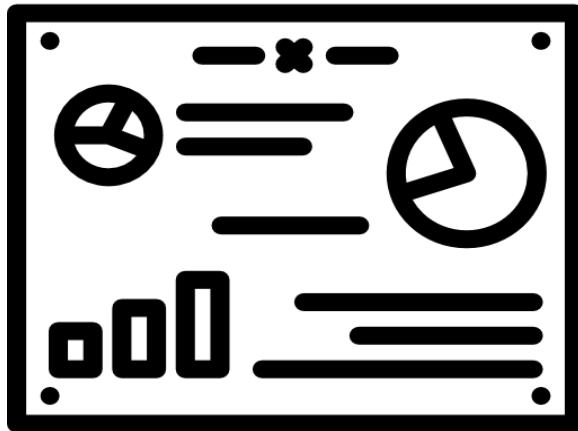
# **Big Data Project Final Presentation**

**11조**

**강유민, 김근하, 김대원**

# Topic

진료 내역 데이터를 이용한 **질병 유형** 및  
**발병 경향 분석**



# Goal

**2013년 ~ 2015년(3년)**

진료 내역 데이터를 활용하여  
상위 **5개** 항목의 **질병의 유형**과  
**발병 경향**을 분석 하고 이를 통  
해 의료 산업의 **마케팅 전략**에  
도움이 될 만한 정보를 제공

구분	의과_보건기 관	구분	의과_보건기 관
0	일반의	23	가정의학과
1	내과	24	응급의학과
2	신경과	25	산업의학과
3	정신과	26	예방의학과
4	외과	50	구강악안면외 과
5	정형외과	51	치과보철과
6	신경외과	52	치과교정과
7	흉부외과	53	소아치과
8	성형외과	54	치주과
9	마취통증의학 과	55	치과보존과
10	산부인과	56	구강내과
11	소아청소년과	57	구강악안면방 사선과
12	안과	58	구강병리과
13	이비인후과	59	예방치과
14	피부과	80	한방내과
15	비뇨기과	81	한방부인과
16	영상의학과	82	한방소아과
17	방사선 종양학과	83	한방안과, 이비인
18	병리과	84	한방신경정신 과
19	진단검사의학 과	85	침구과
20	결핵과	86	한방재활의학 과
21	재활의학과	87	사상체질과
22	핵의학과	88	한방응급
ZZ	결측	-	정상 또는 해당사항 없음

# Data1. 진료 내역 정보 데이터

- 국민 건강 보험 공단에서 제공한 데이터
- 연도별 데이터 제공 (2013년 ~ 2015년)
- 국민건강보험가입자 중 해당 년도에  
요양(병/의원)기관으로 부터의 진료내역이  
1건 이상 있는 가입자 **100만 명을 무작위로**  
선별하여 표본 형성
- 진료 내역 데이터 셋의 크기를 축소하기 위해  
구간 분포 비율을 유지한 채로 **정제 과정 거침**

# Data1. 진료 내역 정보 데이터

19 columns

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
	기준년도	가입자 일련번호	진료내역일	성별코드	연령대코드	시도코드	요양개시일	서식코드	진료과목코드	주상병코드	부상병코드	요양일수	입내원일수	심결가산율	심결요양급	심결본인부	심결보험지	총처방일수	데이터 기준일자	
2	2005	800942	1	1	11	27	20051230	3	1 E14	K769		1	1	15	42770	12830	29940	15	20151220	
3	2005	816566	2	2	5	27	20051217	3	5 S801	J060		1	1	15	16460	4930	11530	2	20151220	
4	2005	816566	3	2	5	27	20051219	3	5 S335	S801		7	7	15	89480	21000	68480	6	20151220	
5	2005	991692	4	2	8	27	20051214	3	14 L239	B352		1	1	15	10740	3000	7740	5	20151220	
6	2005	385646	5	1	11	11	20051216	3	1 I10			1	1	15	9050	3000	6050	15	20151220	
7	2005	795595	6	2	2	27	20051202	3	14 L239	J209		1	1	15	10740	3000	7740	2	20151220	
8	2005	88924	7	2	15	27	20051203	3	1 J450	J060		3	3	15	30630	4500	26130	4	20151220	
9	2005	806394	8	2	12	27	20051221	3	5 M545	J069		2	2	15	21310	6000	15310	4	20151220	
0	2005	800941	9	2	10	27	20051206	3	5 M170	K31		1	1	15	11460	3000	8460	2	20151220	
1	2005	789654	10	2	15	27	20051215	3	14 L239	J060		1	1	15	12380	1500	10880	2	20151220	
2	2005	504830	11	2	11	11	20051217	3	4 T242			2	2	15	43390	13010	30380	3	20151220	
3	2005	995208	12	2	3	11	20051231	3	11 J209	D212		1	1	15	59910	17970	41940	3	20151220	
4	2005	23710	13	1	10	11	20051205	3	1 I63	E039		2	2	15	16730	6000	10730	60	20151220	
5	2005	657672	14	2	6	11	20051203	3	1 J209	B351		2	2	15	20810	6000	14810	14	20151220	
6	2005	615056	15	1	7	26	20051202	3	7 K760	B351		2	2	15	15360	6000	9360	38	20151220	
7	2005	65367	16	1	11	26	20051202	3	1 J869	K30		2	2	15	15360	6000	9360	30	20151220	
8	2005	245120	17	1	7	26	20051203	3	7 M791	J209		2	2	15	21820	6000	15820	4	20151220	
9	2005	739185	18	2	5	26	20051205	3	7 K52	R51		2	2	15	18450	6000	12450	4	20151220	
0	2005	426197	19	2	13	26	20051206	3	7 J209			3	3	15	30400	9000	21400	8	20151220	
1	2005	73412	20	2	3	26	20051207	3	7 J209			1	1	15	12170	3000	9170	2	20151220	
2	2005	739143	21	1	10	26	20051210	3	7 J209			1	1	15	12170	3000	9170	2	20151220	
3	2005	74855	22	1	14	48	20051212	3	7 I508	M255		1	1	15	12020	1500	10520	2	20151220	

## Basic information

- 진료내역 번호, 성별, 연령, 시·도

## Treatment information

- 요양 개시일, 진료과목, 주상병코드, 부상병코드, 요양일수, 입 내원 일수

... ..

More than 10 million rows

# Data2. 시·도별 인구수 데이터

- 통계청에서 제공한 데이터
- 연도별 데이터 제공 (2013년 ~ 2015년)
- 시·도별 인구수 제공
- 데이터 활용을 위해 필요한 형태로 새롭게 가공

# Data2. 시·도별 인구수 데이터

	A	B	C	D	E	F	G	H	I	
1	통계표명:	지역별 인구 및 인구밀도								
2	단위:	천명, 명/㎢								
3		2012		2013		2014		2015		
4		인구	인구밀도	인구	인구밀도	인구	인구밀도	인구	인구밀도	
5	계	50,200	501	50,429	503	50,747	506	51,015	509	
6	서울	10,036	16,583	9,990	16,507	9,975	16,482	9,941	16,425	
7	부산	3,462	4,498	3,456	4,489	3,452	4,485	3,452	4,484	
8	대구	2,480	2,807	2,476	2,802	2,475	2,801	2,469	2,794	
9	인천	2,794	2,684	2,830	2,718	2,862	2,732	2,883	2,748	
10	광주	1,504	3,000	1,504	3,000	1,505	3,002			
11	대전	1,540	2,852	1,545	2,860	1,553	2,879	A	B	
12	울산	1,125	1,061	1,137	1,073	1,151	1,085	city	number	
13	세종	102	220	118	255	132	285	11	29906000	
14	경기	11,974	1,177	12,126	1,192	12,282	1,207	26	10360000	
15	강원	1,504	90	1,506	89	1,510	90	27	7420000	
16	충북	1,553	210	1,565	211	1,578	213	28	8575000	
17	충남	2,043	249	2,062	251	2,088	254	29	4515000	
18	전북	1,817	225	1,821	226	1,829	227	30	4640000	
19	전남	1,782	145	1,784	145	1,792	146	31	3452000	
20	경북	2,656	140	2,661	140	2,671	140	31	3452000	
21	경남	3,265	310	3,278	311	3,307	314	41	36831000	
22	제주	561	303	570	308	583	315	42	4533000	
23	수도권	24,805	2,099	24,946	2,111	25,119	2,124	43	4732000	
24	출처:	통계청 「장래인구추계 시도편 : 2015-2045」, 국토교통부 「지적통계」							44	6253000
25	주석:	* 수도권 : 서울, 인천, 경기							45	5485000
26		* 통계청 「장래인구추계 시도편 : 2015-2045」의 시도별 인구와 국토교통부 「지적통계」의 시도별 국토면적을 곱							46	5373000
27		* 시도별 장래인구추계는 2017년에 작성된 자료임							47	8010000
								48	9915000	
								49	1752000	

- 2013년 ~ 2015년 3년 시도별 인구 합산!

- 2013년 ~ 2015년 3년 시도별 인구 합산!

# Data3. 지역별 진료과목 데이터

- 대한 병원 협회에서 제공한 데이터
- 지역별 진료과목 현황 제공
- 데이터 활용을 위해 필요한 형태로 새롭게 가공

- 시도, 내과, 소아 청소년과,
- 이비인후과, 정형외과, 안과

1	city	Internal m	pediatrics	otolaryng	orthopedi	ophthalmology
2	11	175	104	54	141	46
3	26	132	77	36	98	16
4	27	99	58	21	66	12
5	28	48	29	12	36	6
6	29	46	30	10	29	7
7	30	50	17	12	32	8
8	31	41	23	12	23	4
9	41	240	134	67	168	3
10	42	47	23	19	37	11
11	43	46	21	8	29	7
12	44	62	27	16	39	8
13	45	74	27	15	51	8
14	46	82	33	22	55	13
15	47	100	33	26	55	16
16	48	134	76	50	89	14
17	49	8	6	4	7	4



# Analysis : EDA – Step 1

5개 주요 진료 과목 선정



지역 별 분석

연령 & 성별 분석

월별 분석

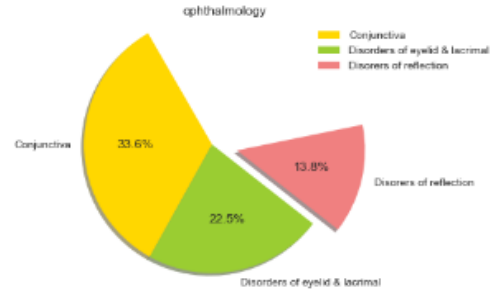
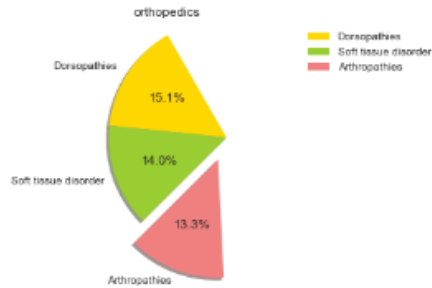
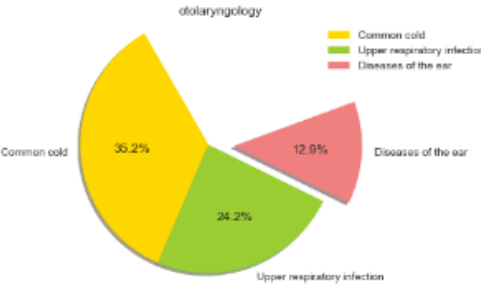
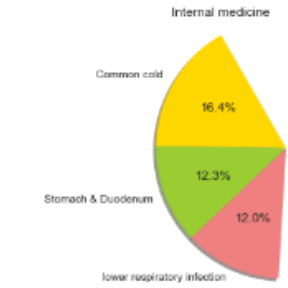
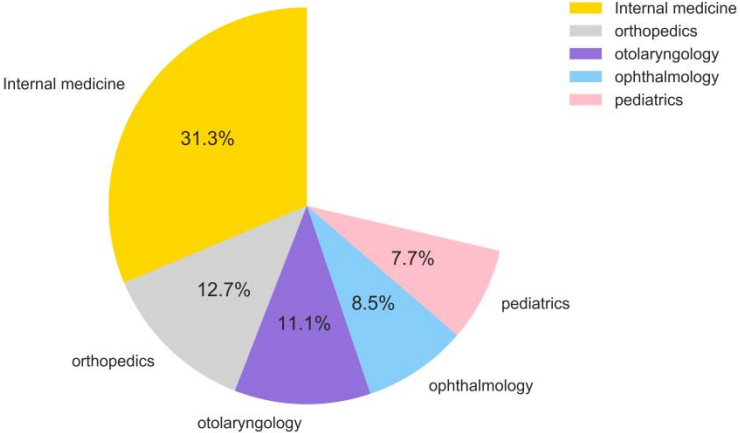


<진료 과목 별 환자 분포 그래프>

# 지역 별 분석

```
In [94]: fig = plt.figure(figsize=(20,10))
# 내과
data_su_sick_1=data_su[data_su['treat_kind']==1]
data_su_sick_1=data_su_sick_1.groupby('main_sick_s').count()
data_su_sick_1['ratio']=data_su_sick_1['city']/data_su_sick_1['city'].sum()
data_su_sick_1=data_su_sick_1.sort_values('ratio',ascending=False)
data_su_sick_1=data_su_sick_1.rename(index={'J0':'Common cold','K2':'Stomach & Duodenum','J2':'lower respiratory infection'})
data_su_sick_1=data_su_sick_1.iloc[:3]
explode = (0, 0, 0)
colors = ['gold', 'yellowgreen', 'lightcoral']
ax1 = plt.subplot2grid((2,2),(0,0))
plt.pie(data_su_sick_1.ratio, explode=explode, shadow=True, colors=colors, autopct='%1.1f%%', startangle=120, labels=data_su_sick_1.index)
plt.title('Internal medicine')
plt.axis('equal')
plt.legend()
# 정형외과
data_su_sick_2=data_su[data_su['treat_kind']==5]
data_su_sick_2=data_su_sick_2.groupby('main_sick_s').count()
data_su_sick_2['ratio']=data_su_sick_2['city']/data_su_sick_2['city'].sum()
data_su_sick_2=data_su_sick_2.sort_values('ratio',ascending=False)
data_su_sick_2=data_su_sick_2.rename(index={'M5':'Dorsopathies','M7':'Soft tissue disorders'})
data_su_sick_2=data_su_sick_2.iloc[:3]
explode = (0, 0, 0.2)
ax1 = plt.subplot2grid((2,2),(0,1))
plt.pie(data_su_sick_2.ratio, explode=explode, shadow=True, colors=colors, autopct='%1.1f%%', startangle=120, labels=data_su_sick_2.index)
plt.title('orthopedics')
plt.axis('equal')
plt.legend()
# 이비인후과
data_su_sick_3=data_su[data_su['treat_kind']==13]
data_su_sick_3=data_su_sick_3.groupby('main_sick_s').count()
data_su_sick_3['ratio']=data_su_sick_3['city']/data_su_sick_3['city'].sum()
data_su_sick_3=data_su_sick_3.sort_values('ratio',ascending=False)
data_su_sick_3=data_su_sick_3.rename(index={'J0':'Common cold','J3':'Upper respiratory infection'})
data_su_sick_3=data_su_sick_3.iloc[:3]
explode = (0, 0, 0.2)
ax1 = plt.subplot2grid((2,2),(1,0))
plt.pie(data_su_sick_3.ratio, explode=explode, shadow=True, colors=colors, autopct='%1.1f%%', startangle=120, labels=data_su_sick_3.index)
plt.title('otolaryngology')
plt.axis('equal')
plt.legend()
# 안과
data_su_sick_4=data_su[data_su['treat_kind']==14]
data_su_sick_4=data_su_sick_4.groupby('main_sick_s').count()
data_su_sick_4['ratio']=data_su_sick_4['city']/data_su_sick_4['city'].sum()
data_su_sick_4=data_su_sick_4.sort_values('ratio',ascending=False)
data_su_sick_4=data_su_sick_4.rename(index={'H10':'Conjunctiva','H11':'Disorders of eyelid & lacrimal','H12':'Disorders of refraction'})
data_su_sick_4=data_su_sick_4.iloc[:3]
explode = (0, 0, 0.2)
ax1 = plt.subplot2grid((2,2),(1,1))
plt.pie(data_su_sick_4.ratio, explode=explode, shadow=True, colors=colors, autopct='%1.1f%%', startangle=120, labels=data_su_sick_4.index)
plt.title('ophthalmology')
plt.axis('equal')
plt.legend()
```

the number of visitor in seoul



# 연령 & 성별 분석

## Treat kind 12 안과

```
In [33]: twe=JR_2010_2015[JR_2010_2015['treat_kind']==12]
```

```
In [32]: t_w=twe[twe['sex']==2]
t_m=twe[twe['sex']==1]
t_w_t=t_w[['age','treat_kind']]
t_m_t=t_m[['age','treat_kind']]
t_w_t=pd.crosstab(t_w_t.age, t_w_t.treat_kind)
t_m_t=pd.crosstab(t_m_t.age, t_m_t.treat_kind)
t_m_t=t_m_t.rename(columns={12:'man'})
t_w_t=t_w_t.rename(columns={12:'woman'})
t_wm=pd.merge(t_w_t, t_m_t, left_index=True, right_index=True)
t_wm.columns.name='sex'
t_wm=t_wm.reset_index()
```

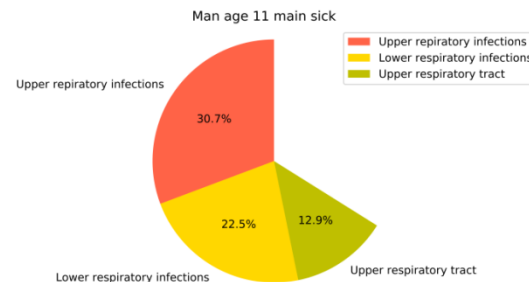
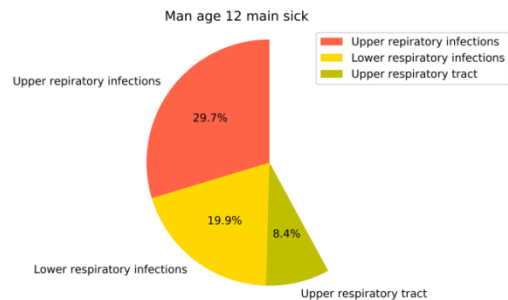
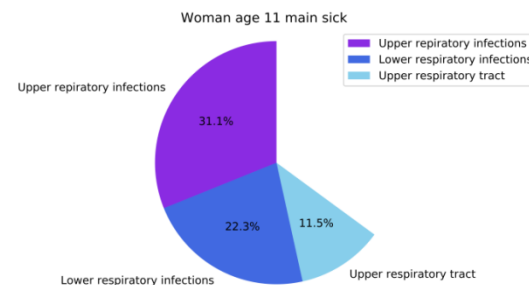
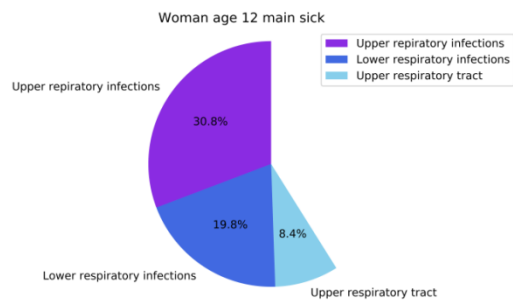
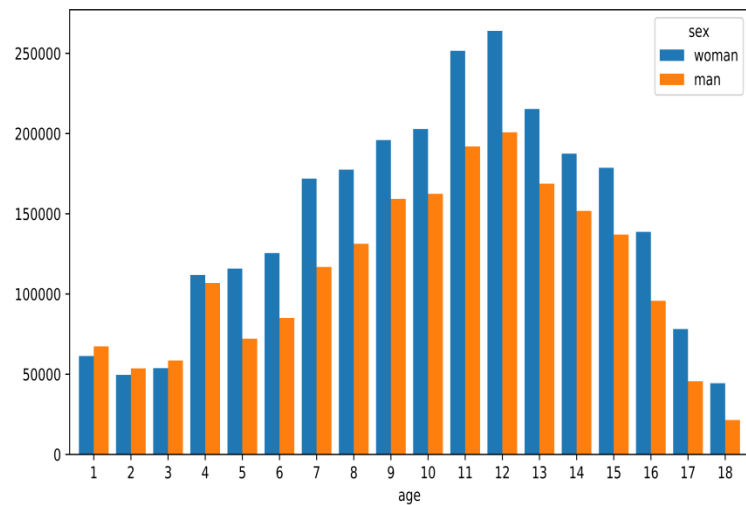
```
In [34]: #t_wm.plot.bar(x='age',y=['woman','man'],rot=0,width=0.8,figsize=(10,
plt.savefig('안과기능검사10.png',dpi=1000)
```

## main sick by woman age 12,11

```
In [ ]: # woman age 12 main sick 3%
t_w_m=t_w[['age','main_sick_s']]
t_m_m=t_m[['age','main_sick_s']]
main12w=t_w_m[t_w_m['age']==12].groupby('main_sick_s').count()
main12w=main12w.sort_values('age',ascending=False)
main12_1=main12w[:3]
main12_1=main12_1.reset_index()
main12_1['ratio']=main12_1['age']/54650
# woman age 11 main sick 3
main12w2=t_w_m[t_w_m['age']==11].groupby('main_sick_s').count()
main12w2=main12w2.sort_values('age',ascending=False)
main12_2=main12w2[:3]
main12_2=main12_2.reset_index()
main12_2['ratio']=main12_2.loc[:,['age']]/50500
main12_1
```

## main sick by man age 2,12

```
In [ ]: # man age 12 main sick 3%
```

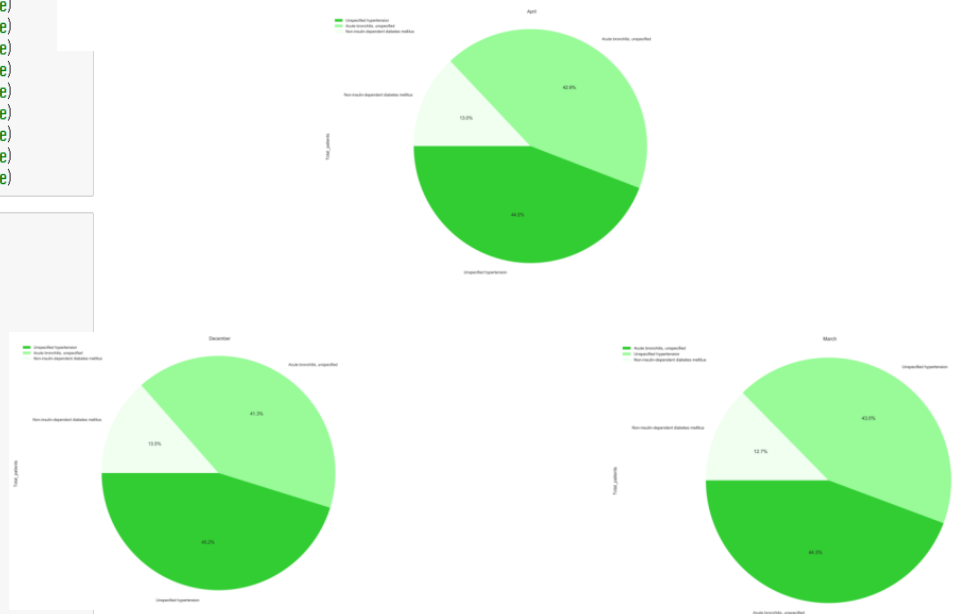
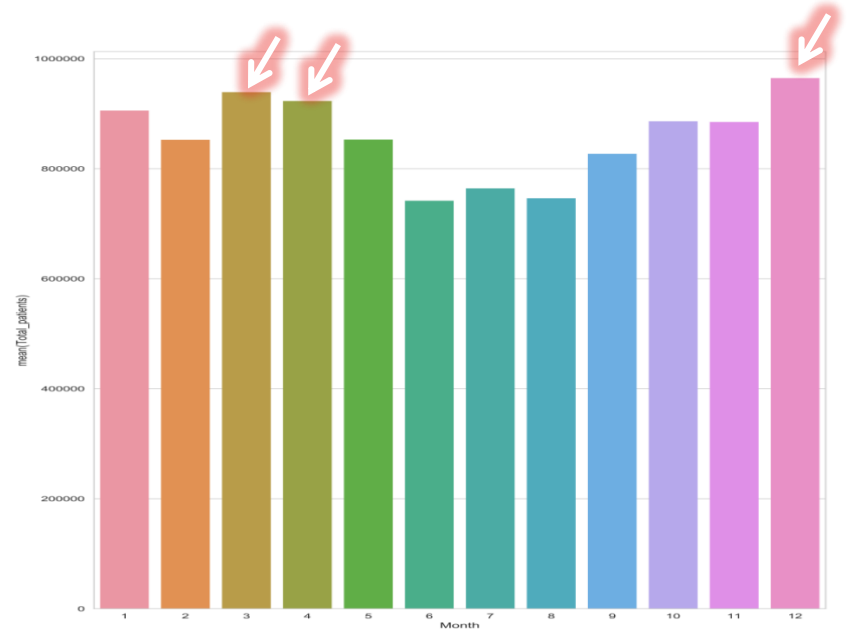


# 월별 분석

## First Month Find 3 Large main\_sick

```
In [12]: a1 = pd.value_counts(ms_2010_2015_code1_st[ms_2010_2015_code1_st['main_sick'].str.match('A')]['main_sick'], values, sort = True)
b1 = pd.value_counts(ms_2010_2015_code1_st[ms_2010_2015_code1_st['main_sick'].str.match('B')]['main_sick'], values, sort = True)
c1 = pd.value_counts(ms_2010_2015_code1_st[ms_2010_2015_code1_st['main_sick'].str.match('C')]['main_sick'], values, sort = True)
d1 = pd.value_counts(ms_2010_2015_code1_st[ms_2010_2015_code1_st['main_sick'].str.match('D')]['main_sick'], values, sort = True)
e1 = pd.value_counts(ms_2010_2015_code1_st[ms_2010_2015_code1_st['main_sick'].str.match('E')]['main_sick'], values, sort = True)
f1 = pd.value_counts(ms_2010_2015_code1_st[ms_2010_2015_code1_st['main_sick'].str.match('F')]['main_sick'], values, sort = True)
g1 = pd.value_counts(ms_2010_2015_code1_st[ms_2010_2015_code1_st['main_sick'].str.match('G')]['main_sick'], values, sort = True)
h1 = pd.value_counts(ms_2010_2015_code1_st[ms_2010_2015_code1_st['main_sick'].str.match('H')]['main_sick'], values, sort = True)
i1 = pd.value_counts(ms_2010_2015_code1_st[ms_2010_2015_code1_st['main_sick'].str.match('I')]['main_sick'], values, sort = True)
j1 = pd.value_counts(ms_2010_2015_code1_st[ms_2010_2015_code1_st['main_sick'].str.match('J')]['main_sick'], values, sort = True)
k1 = pd.value_counts(ms_2010_2015_code1_st[ms_2010_2015_code1_st['main_sick'].str.match('K')]['main_sick'], values, sort = True)
l1 = pd.value_counts(ms_2010_2015_code1_st[ms_2010_2015_code1_st['main_sick'].str.match('L')]['main_sick'], values, sort = True)
m1 = pd.value_counts(ms_2010_2015_code1_st[ms_2010_2015_code1_st['main_sick'].str.match('M')]['main_sick'], values, sort = True)
n1 = pd.value_counts(ms_2010_2015_code1_st[ms_2010_2015_code1_st['main_sick'].str.match('N')]['main_sick'], values, sort = True)
o1 = pd.value_counts(ms_2010_2015_code1_st[ms_2010_2015_code1_st['main_sick'].str.match('O')]['main_sick'], values, sort = True)
p1 = pd.value_counts(ms_2010_2015_code1_st[ms_2010_2015_code1_st['main_sick'].str.match('P')]['main_sick'], values, sort = True)
q1 = pd.value_counts(ms_2010_2015_code1_st[ms_2010_2015_code1_st['main_sick'].str.match('Q')]['main_sick'], values, sort = True)
r1 = pd.value_counts(ms_2010_2015_code1_st[ms_2010_2015_code1_st['main_sick'].str.match('R')]['main_sick'], values, sort = True)
s1 = pd.value_counts(ms_2010_2015_code1_st[ms_2010_2015_code1_st['main_sick'].str.match('S')]['main_sick'], values, sort = True)
t1 = pd.value_counts(ms_2010_2015_code1_st[ms_2010_2015_code1_st['main_sick'].str.match('T')]['main_sick'], values, sort = True)
u1 = pd.value_counts(ms_2010_2015_code1_st[ms_2010_2015_code1_st['main_sick'].str.match('U')]['main_sick'], values, sort = True)
v1 = pd.value_counts(ms_2010_2015_code1_st[ms_2010_2015_code1_st['main_sick'].str.match('V')]['main_sick'], values, sort = True)
w1 = pd.value_counts(ms_2010_2015_code1_st[ms_2010_2015_code1_st['main_sick'].str.match('W')]['main_sick'], values, sort = True)
x1 = pd.value_counts(ms_2010_2015_code1_st[ms_2010_2015_code1_st['main_sick'].str.match('X')]['main_sick'], values, sort = True)
y1 = pd.value_counts(ms_2010_2015_code1_st[ms_2010_2015_code1_st['main_sick'].str.match('Y')]['main_sick'], values, sort = True)
z1 = pd.value_counts(ms_2010_2015_code1_st[ms_2010_2015_code1_st['main_sick'].str.match('Z')]['main_sick'], values, sort = True)
```

```
In [13]: code1_list = []
a1 = pd.DataFrame(a1)
a1 = a1.rename(columns = {0: 'counts'})
a1_max = a1.iloc[a1['counts'].values.argmax()]
code1_list.append(a1_max)
b1 = pd.DataFrame(b1)
b1 = b1.rename(columns = {0: 'counts'})
code1_list.append(b1.iloc[b1['counts'].values.argmax()])
c1 = pd.DataFrame(c1)
c1 = c1.rename(columns = {0: 'counts'})
code1_list.append(c1.iloc[c1['counts'].values.argmax()])
d1 = pd.DataFrame(d1)
d1 = d1.rename(columns = {0: 'counts'})
code1_list.append(d1.iloc[d1['counts'].values.argmax()])
e1 = pd.DataFrame(e1)
e1 = e1.rename(columns = {0: 'counts'})
code1_list.append(e1.iloc[e1['counts'].values.argmax()])
f1 = pd.DataFrame(f1)
f1 = f1.rename(columns = {0: 'counts'})
```



# Analysis : EDA – Step 2

	number	ratio
main_sick_s		
J0	3256078	9.606094
J2	2820127	8.319949
I1	2090834	6.168386
M5	1890460	5.577242
J3	1508061	4.449087
M1	1296151	3.823910
M7	1117589	3.297115
K2	1087235	3.207565
L2	899726	2.654375
E1	888924	2.622507
M4	826313	2.437792
H6	724304	2.136844
S5	427922	1.262457
B3	406011	1.197815
A0	399608	1.178925
K5	377183	1.112767

감기 및 호흡기 질환

관절, 척추 및 연조직 질환

식도, 위 및 십이지장 질환

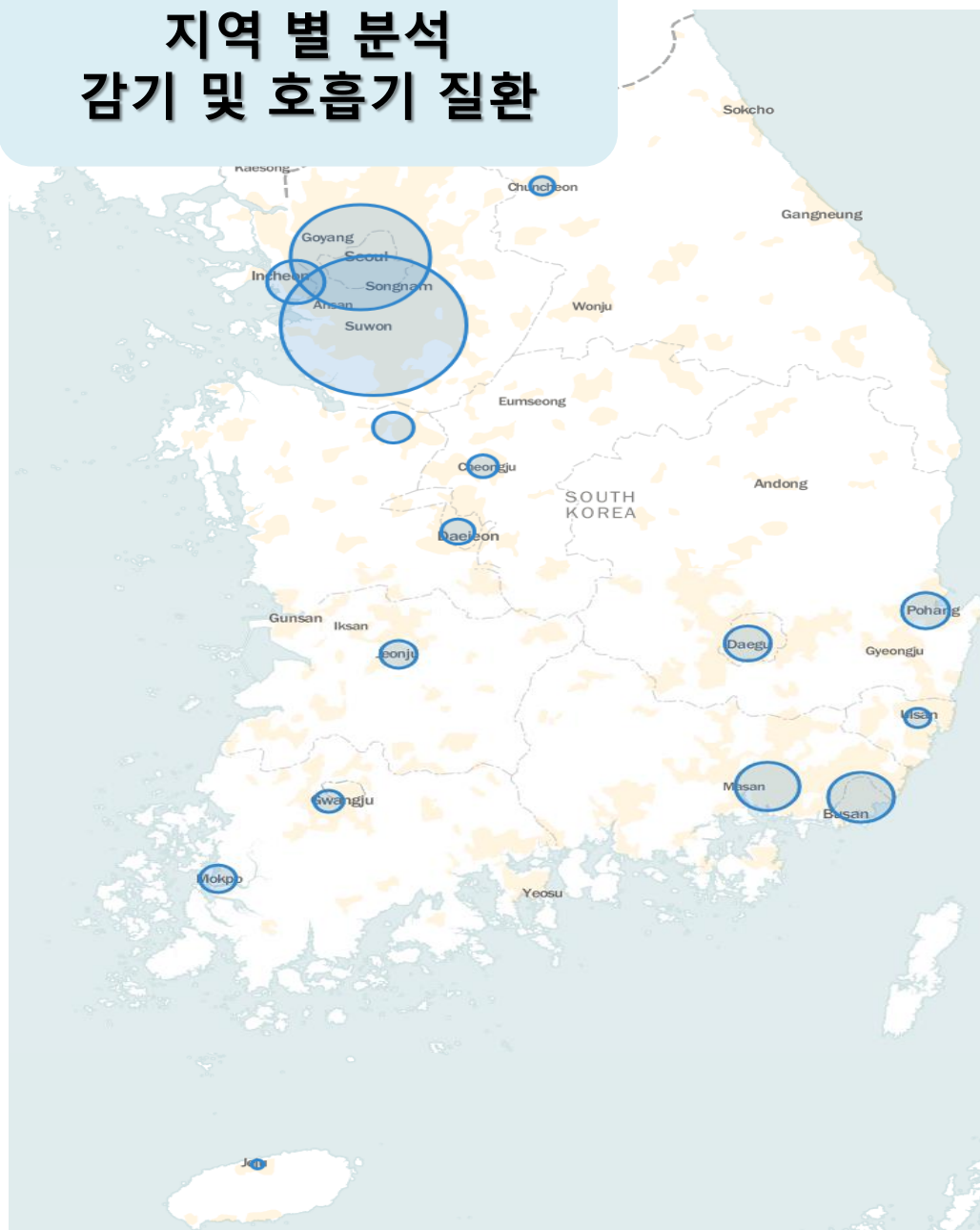
주요 질병  
(STEP 1 기준)

지역 별 분석

월별 분석

연령 & 성별 분석

## 지역 별 분석 감기 및 호흡기 질환



city	number
Geonggi	1951700
Seoul	1441390
Busan	501095
Geongnam	487949
Incheon	436450
Geongbuk	360681
Daegu	350615
Chungnam	308911
Jeonbuk	276578
Jeonnam	275215
Daejeon	252140
Chungbuk	228575
Gwangju	220330
Ulsan	192124
Gangwon	185537
Jeju	92684

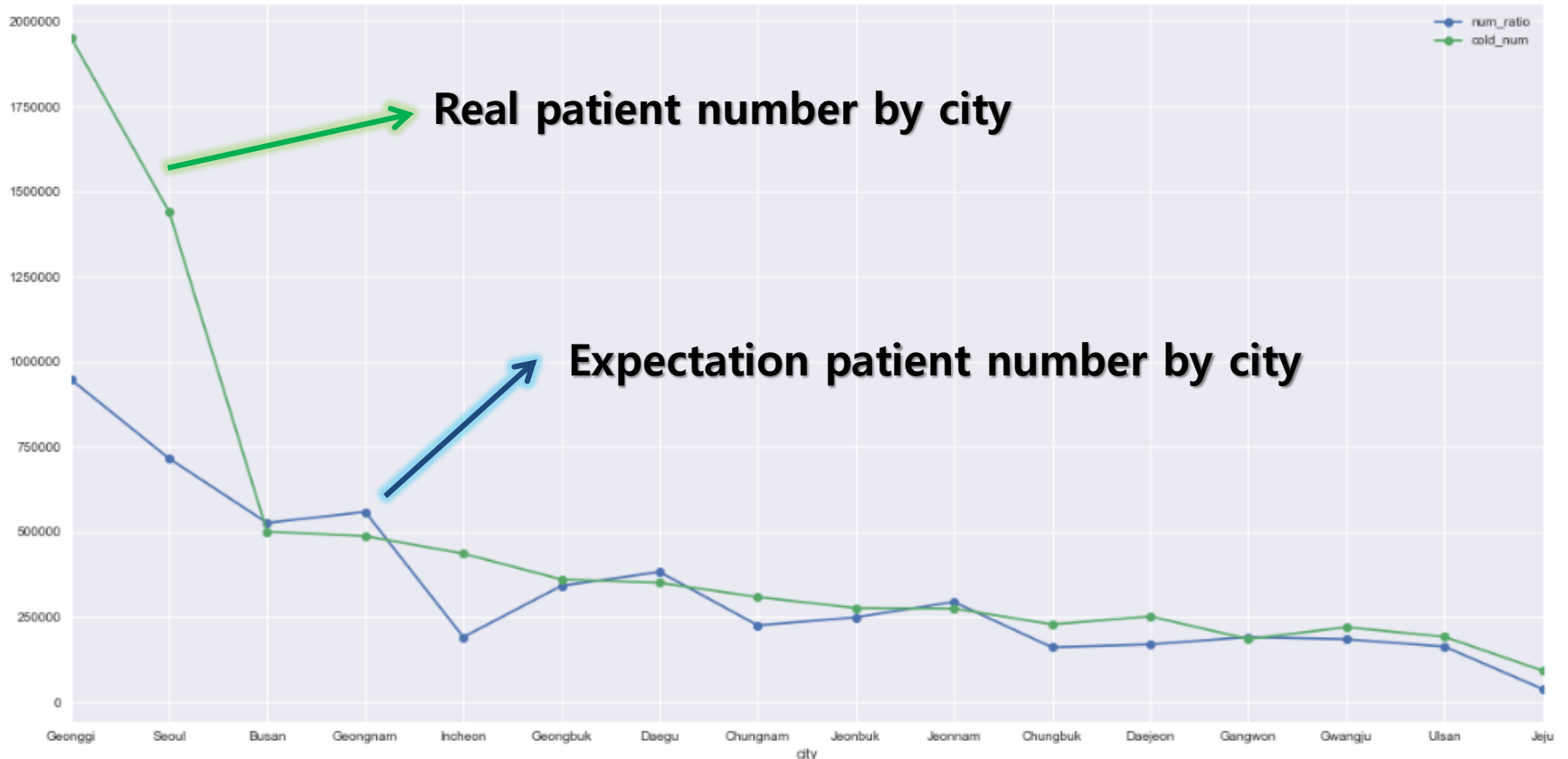
## 지역 별 분석 감기 및 호흡기 질환

$$\text{Expectation patient number} = 3 * 2880 * (N_i * R_i + N_p * R_p + N_o * R_o)$$

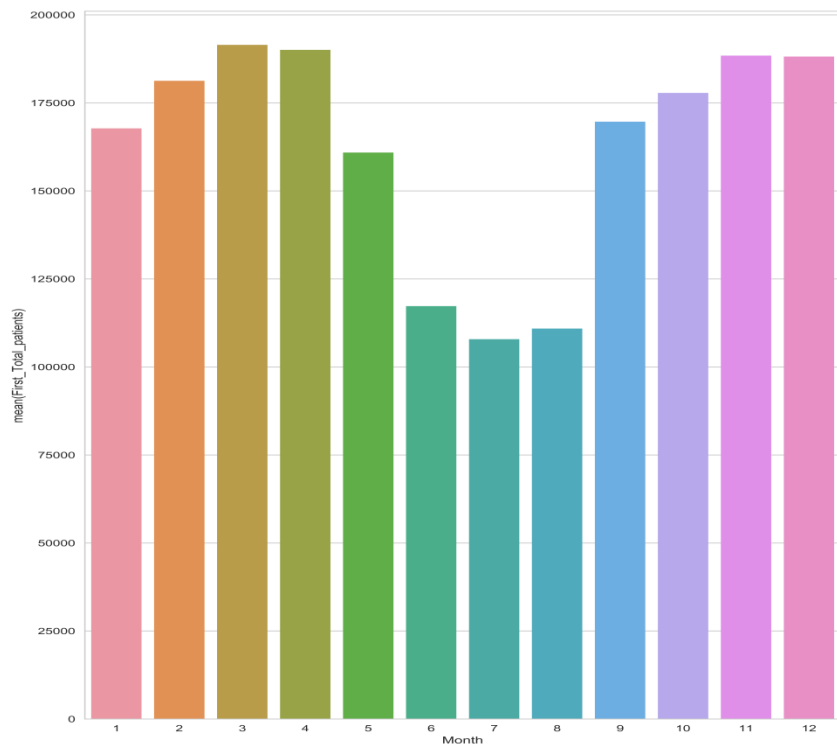
3 year

Patient number per doctor in 1 year  
(2 patient) \* (6 hour) \* (5 days) \* (48 weeks)

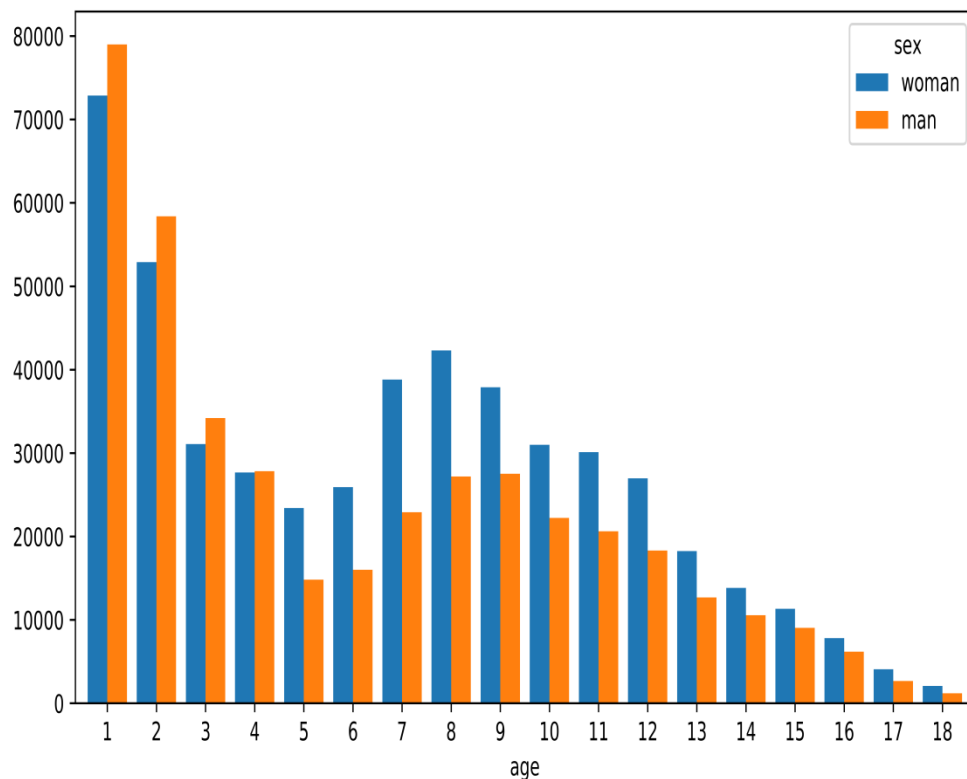
Real cold patient number & Expectaion cold patient number



# 월 별 & 연령 및 성별 분석 감기 및 호흡기 질환



월 별에 따른  
감기 및 호흡기 환자 수  
[경기 지역]



연령 및 성별에 따른  
감기 및 호흡기 환자 수  
[경기 지역]



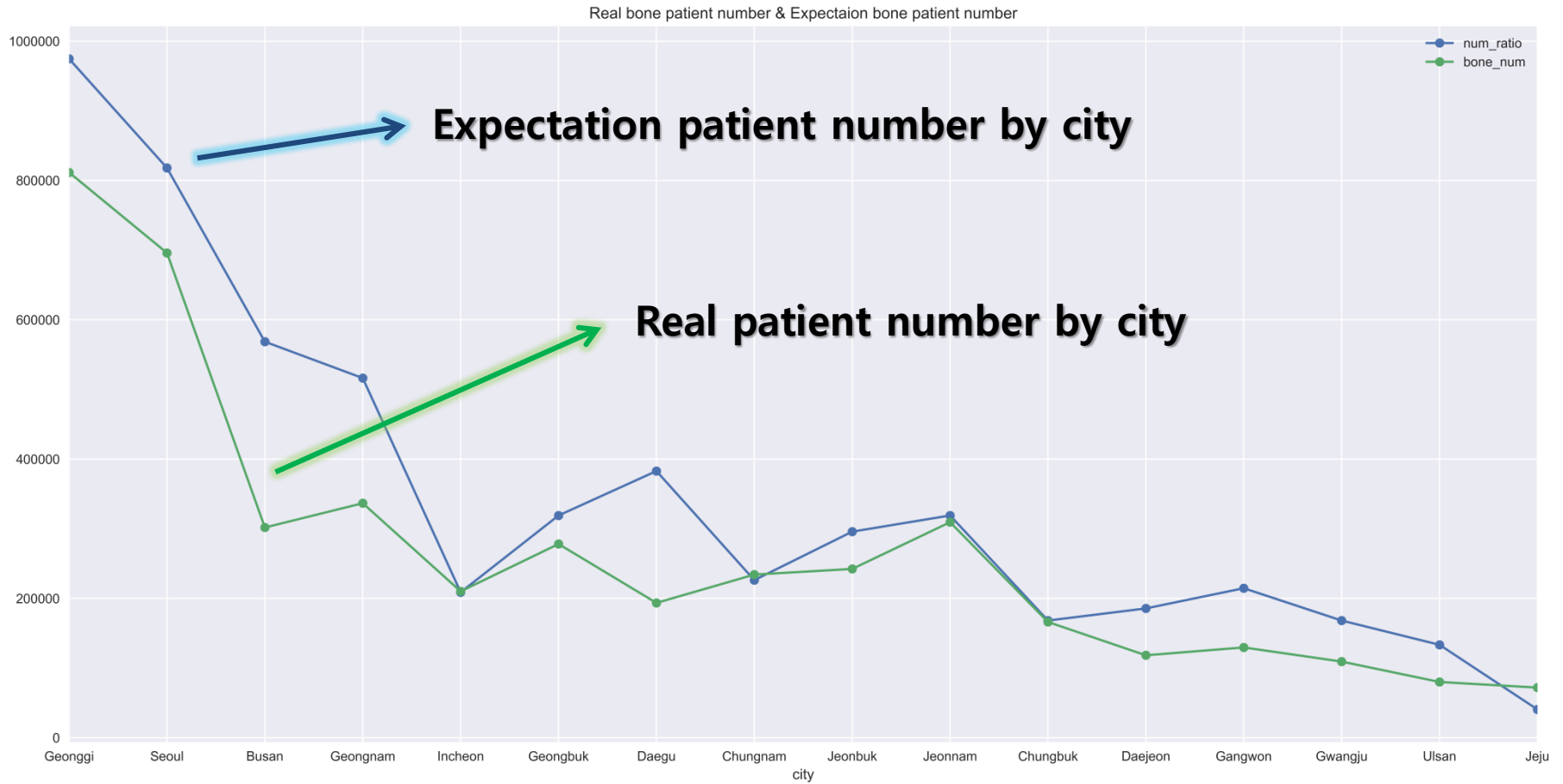
## 지역 별 분석 관절, 척추 및 연조직 질환



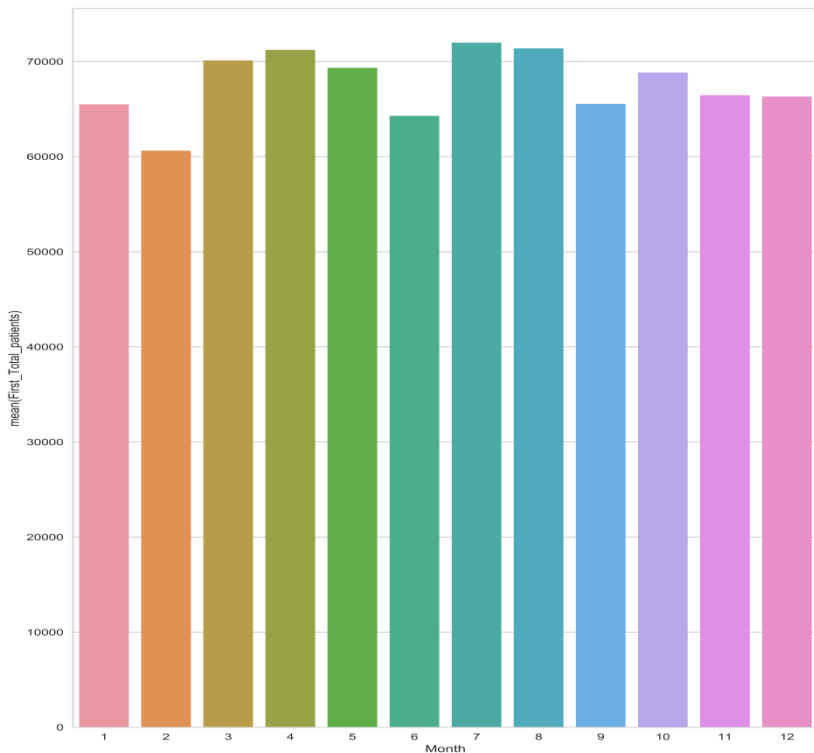
city	number
Geonggi	811525
Seoul	695991
Geongnam	336663
Jeonnam	309454
Busan	301820
Geongbuk	278134
Jeonbuk	242348
Chungnam	234223
Incheon	210143
Daegu	193588
Chungbuk	166157
Gangwon	129633
Daejeon	118360
Gwangju	109343
Ulsan	80067
Jeju	71977

## 지역 별 분석 관절, 척추 및 연조직 질환

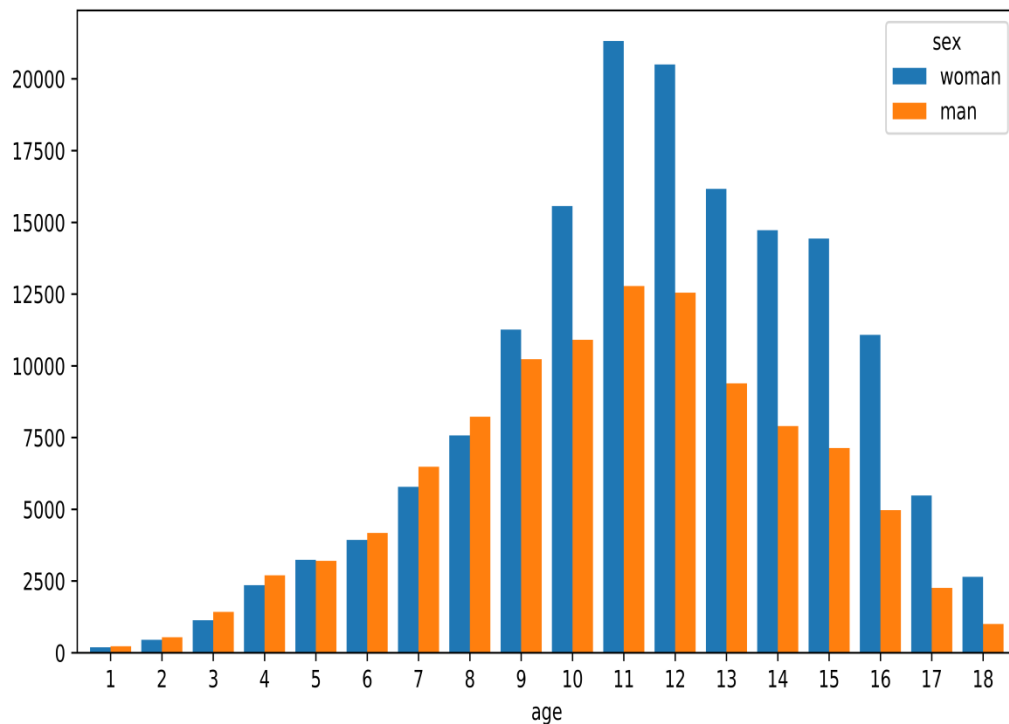
$$\text{Expectation patient number} = 3 * 2880 * (N_o * R_o)$$



## 월 별 & 연령 및 성별 분석 관절, 척추 및 연조직 질환



월 별에 따른  
관절, 척추 및 연조직 환자 수  
[경기 지역]



연령 및 성별에 따른  
관절, 척추 및 연조직 환자 수  
[경기 지역]

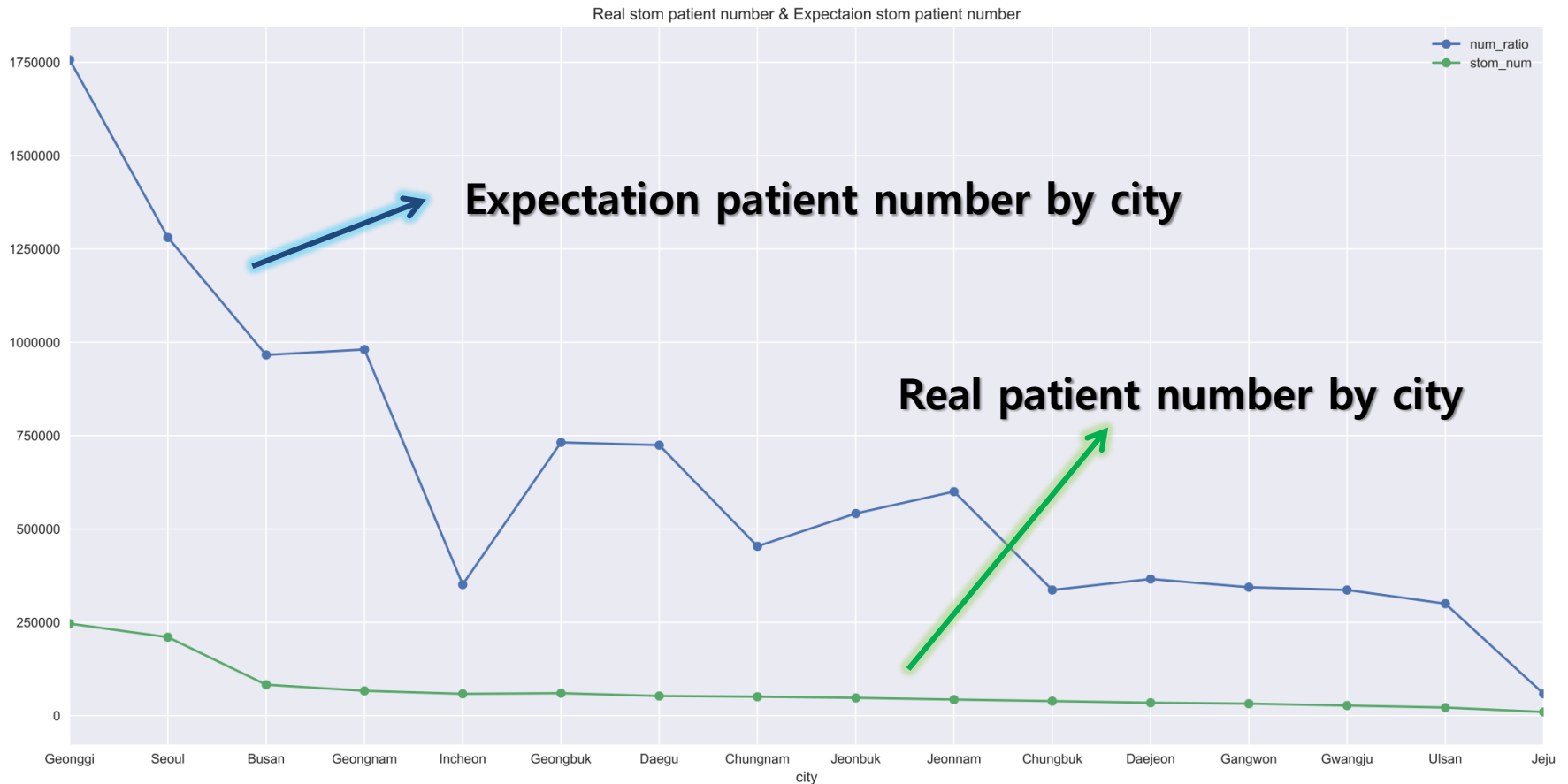
## 지역 별 분석 식도, 위 및 십이지장 질환



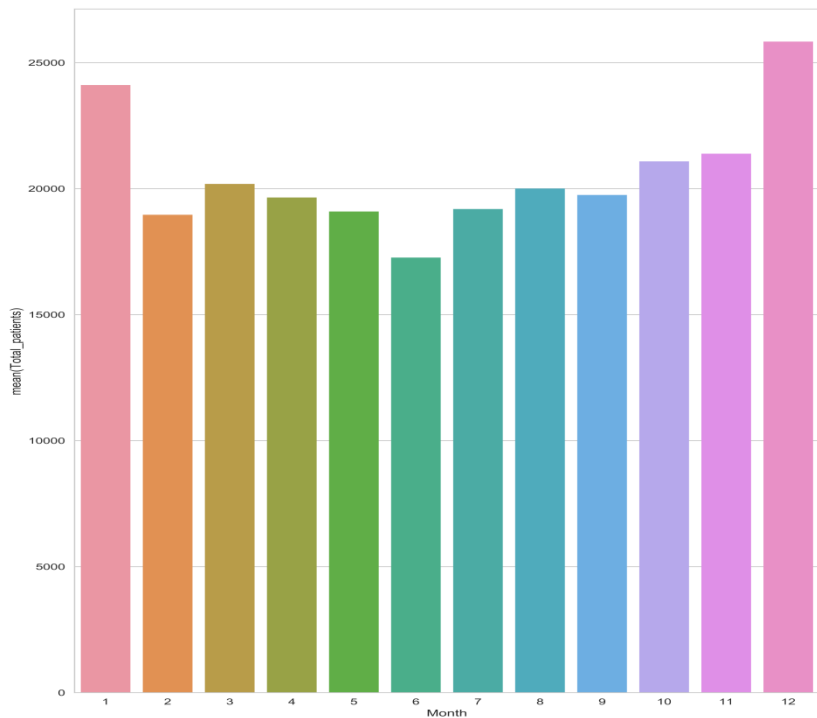
city	number
Geonggi	246536
Seoul	210314
Busan	82897
Geongnam	66491
Geongbuk	60179
Incheon	58449
Daegu	52605
Chungnam	50693
Jeonbuk	47649
Jeonnam	43136
Chungbuk	38955
Daejeon	34632
Gangwon	32161
Gwangju	27148
Ulsan	21724
Jeju	9992

## 지역 별 분석 식도, 위 및 십이지장 질환

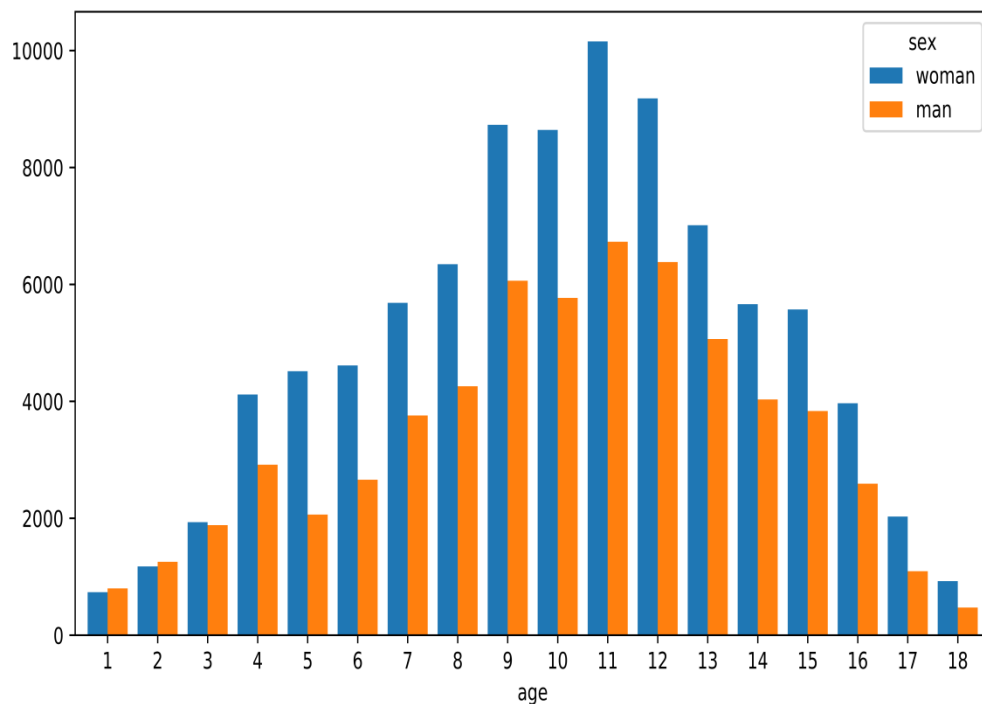
$$\text{Expectation patient number} = 3 * 2880 * (N_i * R_i)$$



## 월 별 & 연령 및 성별 분석 식도, 위 및 십이지장 질환



월 별에 따른  
식도, 위 및 십이지장 환자 수  
[경기 지역]



연령 및 성별에 따른  
식도, 위 및 십이지장 환자 수  
[경기 지역]

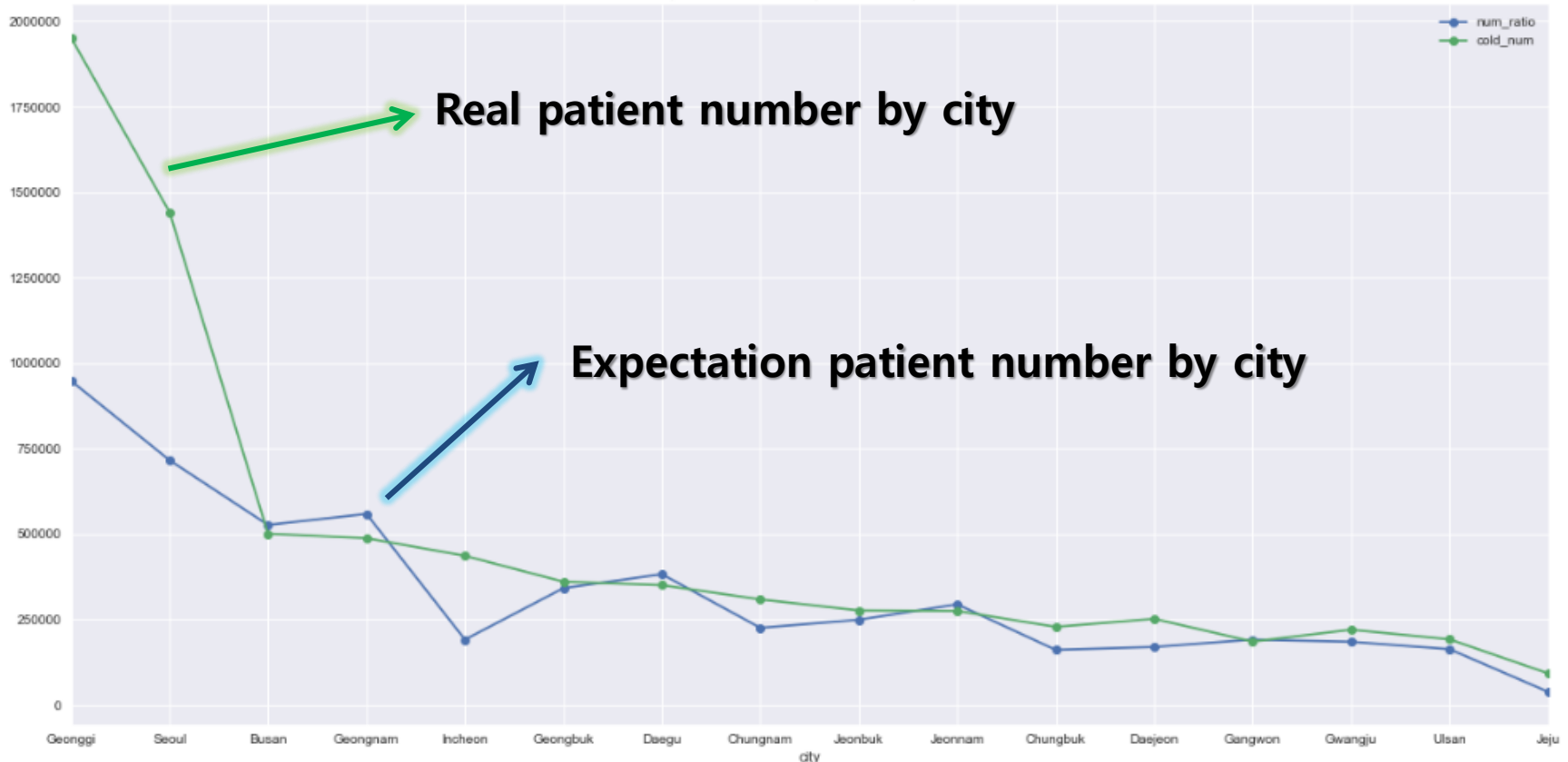
# Result

Real patient number > Expectation patient number  
⇒ Need to **increase** hospital

Real patient number < Expectation patient number  
⇒ Exist more than **enough** hospital

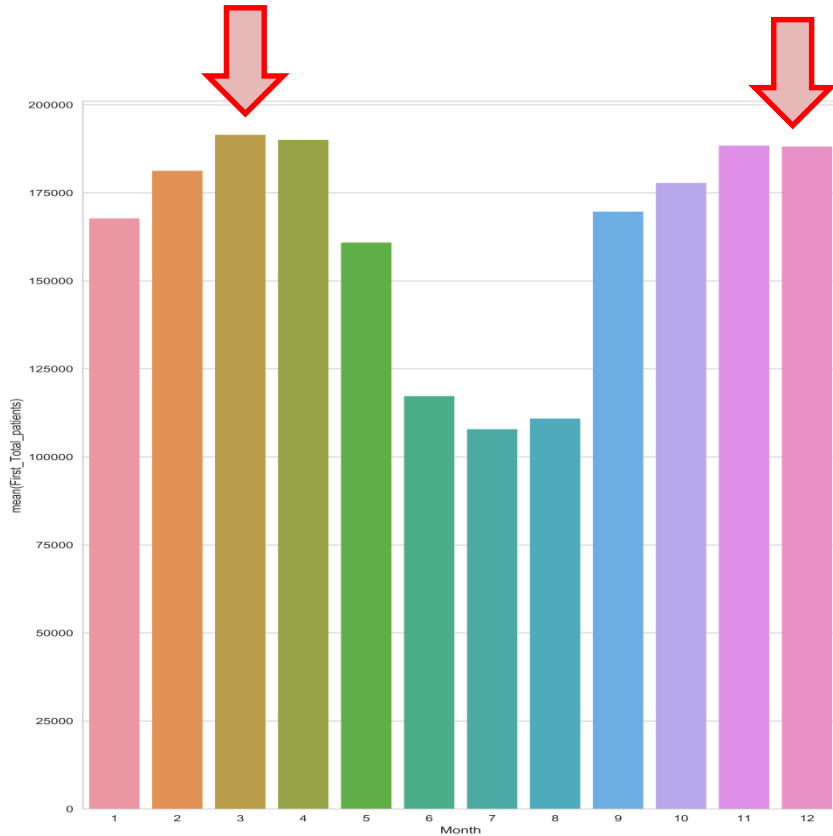
## 지역 별 분석 감기 및 호흡기 질환

Real cold patient number & Expectaion cold patient number

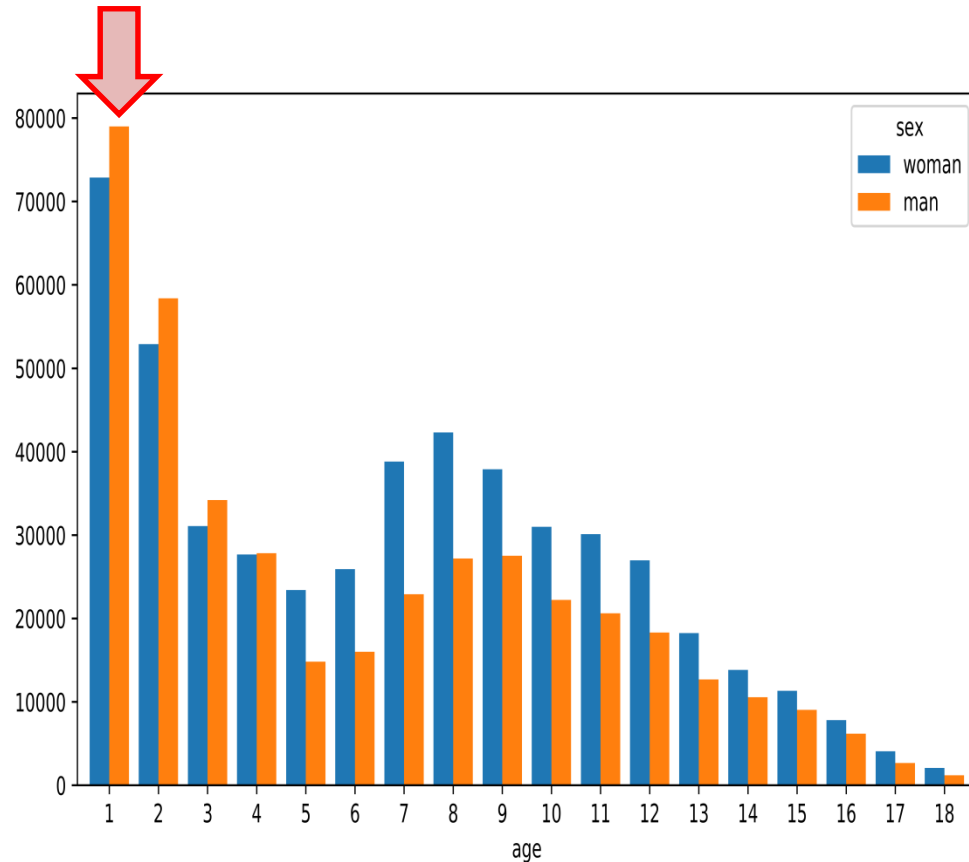


# Result

3월과 12월에 병원을 찾는 0~9세의 영유아들을 위해  
병원이 그에 맞는 서비스를 제공하면 Good



월 별에 따른  
감기 및 호흡기 환자 수  
[경기 지역]



연령 및 성별에 따른  
감기 및 호흡기 환자 수  
[경기 지역]



**Q&A**