

In [1]:

```
1 import pandas as pd
2 import numpy as np
3 import seaborn as sns
```

In [2]:

```
1 import matplotlib.pyplot as plt
2 plt.rc('font', family='Malgun Gothic')
3 plt.rc('axes', unicode_minus=False)
```

In [3]:

```
1 # 가져온 데이터파일을 판다스로 인식해 df(데이터 프레임 약칭)에 담는다.
2 # low_memory=False 다른 타입이 섞여 있어 써주어야 한다.
3 df = pd.read_csv("data/소상공인시장진흥공단_상가업소정보_의료기관_201909.csv", low_memory=False)
4
5 # df.shape 는 데이터 프레임의 행과 열의 수를 보여주고 (행, 열)로 표시된다.
6 df.shape
```

Out[3]:

(91335, 39)

In [4]:

```
1 # df.head() 는 데이터 프레임의 앞부분을 보여주고 기본값은 다섯 개만 보여주지만 괄호 안에 수를 넣
2 df.head(1)
```

Out[4]:

	상가업소 번호	상 호 명	지점 명	상 권 업 종 대 분 류 코 드	상 권 업 종 대 분 류 명	상 권 업 종 대 분 류 코 드	상 권 업 종 대 분 류 명	상 권 업 종 소 분 류 코 드	상 권 업 종 소 분 류 명	표준상 업분류 코드	...	건물관리번호	기 타 비 고
0	19956873	하 나 산 부 인 과	NaN	S	의 료	S01	병 원	S01B10	산 부 인 과	Q86201	...	4127310900110810000010857	상 권 업 종 소 분 류 명

1 rows × 39 columns

In [5]:

```
1 # df.tail() 은 데이터 프레임의 앞부분을 보여주고 기본값은 다섯 개만 보여주지만 괄호 안에 수를 넣
2 df.tail(1)
```

Out[5]:

상가업소 번호	상 호 명	지 점 명	상 권 업 종 대 분 류 코 드	상 권 업 종 대 분 류 명	상 권 업 종 중 분 류 코 드	상 권 업 종 중 분 류 명	상 권 업 종 소 분 류 코 드	상 권 업 종 소 분 류 명	표 준 산 업 분 류 코 드	...	건 물 관 리 번.
91334	16109073	천안김안과 천안역본점의원	NaN	S	의료	S01	병원	S01B13	안과의원	Q86201	... 44131107001026600170163'

1 rows × 39 columns

In [6]:

```

1 df.info() # df.info() 로 데이터 요약을 볼 수 있다.
2
3 # 첫번째는 컬럼명을 볼 수 있다
4 # 두번째는 각 컬럼의 수를 볼 수 있다. 다른 컬럼과 숫자 차이가 나면 결측치가 들어가 있다는 걸 알
5 # 세번째는 non-null 타입을 볼 수 있다.
6 # 네번째는 데이터의 타입을 볼 수 있다.
7 # 마지막줄에는 메모리 사용량이 적혀있다.

```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 91335 entries, 0 to 91334
```

```
Data columns (total 39 columns):
```

#	Column	Non-Null Count	Dtype
0	상가업소번호	91335 non-null	int64
1	상호명	91335 non-null	object
2	지점명	1346 non-null	object
3	상권업종대분류코드	91335 non-null	object
4	상권업종대분류명	91335 non-null	object
5	상권업종중분류코드	91335 non-null	object
6	상권업종중분류명	91335 non-null	object
7	상권업종소분류코드	91335 non-null	object
8	상권업종소분류명	91335 non-null	object
9	표준산업분류코드	86413 non-null	object
10	표준산업분류명	86413 non-null	object
11	시도코드	90956 non-null	float64
12	시도명	90956 non-null	object
13	시군구코드	90956 non-null	float64
14	시군구명	90956 non-null	object
15	행정동코드	91335 non-null	int64
16	행정동명	90956 non-null	object
17	법정동코드	91280 non-null	float64
18	법정동명	91280 non-null	object
19	지번코드	91335 non-null	int64
20	대지구분코드	91335 non-null	int64
21	대지구분명	91335 non-null	object
22	지번본번지	91335 non-null	int64
23	지번부번지	72079 non-null	float64
24	지번주소	91335 non-null	object
25	도로명코드	91335 non-null	int64
26	도로명	91335 non-null	object
27	건물본번지	91335 non-null	int64
28	건물부번지	10604 non-null	float64
29	건물관리번호	91335 non-null	object
30	건물명	46453 non-null	object
31	도로명주소	91335 non-null	object
32	구우편번호	91323 non-null	float64
33	신우편번호	91333 non-null	float64
34	동정보	7406 non-null	object
35	층정보	44044 non-null	object
36	호정보	15551 non-null	object
37	경도	91335 non-null	float64
38	위도	91335 non-null	float64

```
dtypes: float64(9), int64(7), object(23)
```

```
memory usage: 27.2+ MB
```

In [7]:

```
1 df.columns # df.columns() 로 컬럼명만 출력할 수 있다.
```

Out[7]:

```
Index(['상가업소번호', '상호명', '지점명', '상권업종대분류코드', '상권업종대분류명',  
      '상권업종중분류코드',  
      '상권업종중분류명', '상권업종소분류코드', '상권업종소분류명', '표준산업분류코  
드', '표준산업분류명', '시도코드',  
      '시도명', '시군구코드', '시군구명', '행정동코드', '행정동명', '법정동코드',  
'법정동명', '지번코드',  
      '대지구분코드', '대지구분명', '지번본번지', '지번부번지', '지번주소', '도로명  
코드', '도로명', '건물본번지',  
      '건물부번지', '건물관리번호', '건물명', '도로명주소', '구우편번호', '신우편번  
호', '동정보', '층정보',  
      '호정보', '경도', '위도'],  
      dtype='object')
```

In [8]:

```
1 df.dtypes # df.dtypes() 로 데이터 타입만 출력 할 수 있다.
```

Out[8]:

```
상가업소번호      int64
상호명            object
지점명            object
상권업종대분류코드  object
상권업종대분류명    object
상권업종중분류코드  object
상권업종중분류명    object
상권업종소분류코드  object
상권업종소분류명    object
표준산업분류코드    object
표준산업분류명      object
시도코드          float64
시도명            object
시군구코드         float64
시군구명          object
행정동코드        int64
행정동명          object
법정동코드        float64
법정동명          object
지번코드          int64
대지구분코드      int64
대지구분명        object
지번본번지        int64
지번부번지        float64
지번주소          object
도로명코드        int64
도로명            object
건물본번지        int64
건물부번지        float64
건물관리번호      object
건물명            object
도로명주소        object
구우편번호        float64
신우편번호        float64
동정보            object
층정보            object
호정보            object
경도              float64
위도              float64
dtype: object
```

In [9]:

```

1 null_count = df.isnull().sum() # df.isnull() 로 결측치를 볼 수 있다.
2
3 # null값은 True 로 표시 된다.
4 # df.isnull().sum() 을 하면 파이썬에서는 True==1 이기 때문에 sum으로 결측치를 셀 수 있다.
5
6 null_count

```

Out[9]:

```

상가업소번호      0
상호명            0
지점명          89989
상권업종대분류코드      0
상권업종대분류명      0
상권업종중분류코드      0
상권업종중분류명      0
상권업종소분류코드      0
상권업종소분류명      0
표준산업분류코드    4922
표준산업분류명      4922
시도코드          379
시도명            379
시군구코드        379
시군구명          379
행정동코드        0
행정동명          379
법정동코드        55
법정동명          55
지번코드          0
대지구분코드      0
대지구분명        0
지번본번지        0
지번부번지        19256
지번주소          0
도로명코드        0
도로명            0
건물본번지        0
건물부번지        80731
건물관리번호      0
건물명            44882
도로명주소        0
구우편번호        12
신우편번호        2
동정보            83929
층정보            47291
호정보            75784
경도              0
위도              0
dtype: int64

```



In [11]:

```
1 df_null_count = null_count.reset_index() # df.reset_index 를 하면 자료를 데이터 프레임의 형태로
2 df_null_count.head()
```

Out[11]:

	index	0
0	상가업소번호	0
1	상호명	0
2	지점명	89989
3	상권업종대분류코드	0
4	상권업종대분류명	0

In [12]:

```
1 #df.columns = [ " ", " " ] 에서 대괄호 안에 스트링을 넣으면 컬럼명을 지정할 수 있다.
2 df_null_count.columns = ["컬럼명", "결측치수"]
3 df_null_count.head()
```

Out[12]:

	컬럼명	결측치수
0	상가업소번호	0
1	상호명	0
2	지점명	89989
3	상권업종대분류코드	0
4	상권업종대분류명	0



In [13]:

```

1 # df.sort_values() 로 지정한 값을 정렬할 수 있다.
2 df_null_count_top = df_null_count.sort_values(by="결측치수", ascending=False).head(10)
3
4 # by= 뒤에 컬럼명을 입력하면 컬럼을 지정 할 수 있다
5 # ascending= 뒤에 True 를 입력하면 오름차순 False 를 입력하면 내림차순이 된다.
6
7 df_null_count_top

```

Out [13]:

	컬럼명	결측치수
2	지점명	89989
34	동정보	83929
28	건물부번지	80731
36	호정보	75784
35	층정보	47291
30	건물명	44882
23	지번부번지	19256
9	표준산업분류코드	4922
10	표준산업분류명	4922
11	시도코드	379

In [14]:

```

1 df["지점명"].head() # df[" "].head() 로 지정한 컬럼의 앞부분을 불러 올 수 있다.
2
3 # NaN 로 표시되는 것은 Not a Number 의 약자로 결측치를 의미한다.

```

Out [14]:

```

0    NaN
1    NaN
2    NaN
3    NaN
4    수지점
Name: 지점명, dtype: object

```

In [15]:

```

1 # df[" "].tolist() 컬럼 하나를 리스트로 만들어준다.
2 drop_columns = df_null_count_top["컬럼명"].tolist()
3 drop_columns

```

Out[15]:

```

['지점명',
 '동정보',
 '건물부번지',
 '호정보',
 '층정보',
 '건물명',
 '지번부번지',
 '표준산업분류코드',
 '표준산업분류명',
 '시도코드']

```

In [16]:

```
1 df[drop_columns].head()
```

Out[16]:

	지점 명	동정 보	건물부 번지	호정 보	층정 보	건물명	지번부 번지	표준산업분 류코드	표준산업분류명	시도 코드
0	NaN	NaN	NaN	NaN	NaN	산호한양아 파트	NaN	Q86201	일반 의원	41.0
1	NaN	NaN	NaN	NaN	4	NaN	14.0	Q86201	일반 의원	11.0
2	NaN	NaN	NaN	NaN	NaN	한라프라자	1.0	Q86201	일반 의원	41.0
3	NaN	NaN	NaN	NaN	5	NaN	1.0	NaN	NaN	26.0
4	수지 점	NaN	NaN	NaN	1	NaN	2.0	G47811	의약품 및 의료용품 소매업	41.0

In [17]:

```

1 print(df.shape)
2
3 # df.drop() 으로 원하는 행이나 열을 지울 수 있다.
4 df = df.drop(drop_columns, axis=1)
5
6 # 첫 부분에 지우고 싶은 위치를 선정해준다.
7 # axis= 뒤에 0을 넣으면 행을 삭제하고 1을 넣으면 열을 삭제한다.
8
9 print(df.shape)

```

(91335, 39)

(91335, 29)

In [18]:

1 df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 91335 entries, 0 to 91334
Data columns (total 29 columns):
#   Column                Non-Null Count  Dtype
---  -
0   상가업소번호          91335 non-null  int64
1   상호명               91335 non-null  object
2   상권업종대분류코드    91335 non-null  object
3   상권업종대분류명      91335 non-null  object
4   상권업종중분류코드    91335 non-null  object
5   상권업종중분류명      91335 non-null  object
6   상권업종소분류코드    91335 non-null  object
7   상권업종소분류명      91335 non-null  object
8   시도명               90956 non-null  object
9   시군구코드           90956 non-null  float64
10  시군구명             90956 non-null  object
11  행정동코드           91335 non-null  int64
12  행정동명             90956 non-null  object
13  법정동코드           91280 non-null  float64
14  법정동명             91280 non-null  object
15  지번코드             91335 non-null  int64
16  대지구분코드         91335 non-null  int64
17  대지구분명           91335 non-null  object
18  지번본번지           91335 non-null  int64
19  지번주소             91335 non-null  object
20  도로명코드           91335 non-null  int64
21  도로명               91335 non-null  object
22  건물본번지           91335 non-null  int64
23  건물관리번호         91335 non-null  object
24  도로명주소           91335 non-null  object
25  구우편번호           91323 non-null  float64
26  신우편번호           91333 non-null  float64
27  경도                 91335 non-null  float64
28  위도                 91335 non-null  float64
dtypes: float64(6), int64(7), object(16)
memory usage: 20.2+ MB
```

In [19]:

1 df["위도"].mean() # df[" "].mean() 지정한 컬럼의 평균값을 알 수 있다.

Out[19]:

36.62471119236673

In [20]:

1 df["위도"].median() # df[" "].median() 지정한 컬럼의 중간 값을 알 수 있다.

Out[20]:

37.2346523177033

In [21]:

```
1 df["위도"].max() # df[" "].max() 지정한 컬럼의 최대값을 알 수 있다.
```

Out[21]:

38.4996585705598

In [22]:

```
1 df["위도"].min() # df[" "].min() 지정한 컬럼의 최소값을 알 수 있다.
```

Out[22]:

33.2192896688307

In [23]:

```
1 df["위도"].count() # df[" "].count 로 지정한 컬럼의 개수를 셀 수 있다.
```

Out[23]:

91335

In [24]:

```
1 df["위도"].describe() # df[" "]. describe() 지정한 컬럼의 데이터 요약값을 볼 수 있다.  
2  
3 # 25%는 앞에서 1/4 되는 값이고, 1사분위수라 부릅니다.  
4 # 50%는 중앙값이고, 2사분위수라고도 합니다.  
5 # 75%는 앞에서 3/4 번째 되는 수고, 3사분위수라 부릅니다.  
6 # mean, min, max는 각각 평균, 최소값, 최대값입니다.
```

Out[24]:

```
count    91335.000000  
mean      36.624711  
std       1.041361  
min       33.219290  
25%       35.811830  
50%       37.234652  
75%       37.507463  
max       38.499659  
Name: 위도, dtype: float64
```

In [25]:

```
1 df[["위도", "경도"]].describe() # df[[" ", " "].describe() 요약값을 시리즈가 아닌 데이터 프레임
```

Out[25]:

	위도	경도
count	91335.000000	91335.000000
mean	36.624711	127.487524
std	1.041361	0.842877
min	33.219290	124.717632
25%	35.811830	126.914297
50%	37.234652	127.084550
75%	37.507463	128.108919
max	38.499659	130.909912

In [26]:

```
1 df.describe(include="object")
2
3 # 다음표 안에 number 가 들어가면 숫자로 된 데이터만 요약
4
5 # 다음표 안에 object 가 들어가면 문자열 데이터만 요약
6 ## count 는 데이터의 개수를 세준다.
7 ## unique 는 중복을 제거한 값을 보여준다.
8 ## top 은 가장 높은 빈도로 저장된 데이터, freq는 그 데이터의 빈도수를 표시한다.
```

Out[26]:

	상호명	상권 업종 대분류 코드	상권 업종 대분류명	상권 업종 중분류 코드	상권 업종 중분류명	상권 업종 소분류 코드	상권 업종 소분류명	시도명	시군구명	행정동명	법정동명	대구명
count	91335	91335	91335	91335	91335	91335	91335	90956	90956	90956	91280	91335
unique	56910	1	1	5	5	34	34	17	228	2791	2822	91335
top	리원	S	의료	S01	병원	S02A01	약국	경기도	서구	중앙동	중동	대구
freq	152	91335	91335	60774	60774	18964	18964	21374	3165	1856	874	9121

In [27]:

```
1 df["상권업종대분류명"].unique() #unique 중복을 제거한 값
```

Out[27]:

```
array(['의료'], dtype=object)
```

In [28]:

```
1 df["상권업종대분류명"].nunique() #nunique unique의 수
```

Out[28]:

1

In [29]:

```
1 df["상권업종중분류명"].unique()
```

Out[29]:

```
array(['병원', '약국/한약방', '수의업', '유사의료업', '의료관련서비스업'], dtype=object)
```

In [30]:

```
1 df["상권업종중분류명"].nunique()
```

Out[30]:

5

In [31]:

```
1 df["상권업종소분류명"].unique()
```

Out[31]:

```
array(['산부인과', '내과/외과', '신경외과', '기타병원', '약국', '동물병원', '한약방', '탕제원',
      '정형/성형외과', '소아과', '이비인후과의원', '노인/치매병원', '언어치료', '수의업-종합', '한의원',
      '치과의원', '침구원', '일반병원', '안과의원', '조산원', '한방병원', '종합병원', '유사의료업기타',
      '응급구조대', '혈액원', '치과병원', '척추교정치료', '피부과', '비뇨기과', '치과기공소', '산후조리원',
      '점골원', '수의업-기타', '제대혈'], dtype=object)
```

In [32]:

```
1 df["상권업종소분류명"].nunique()
```

Out[32]:

34

In [33]:

```
1 len(df["상권업종소분류명"].unique()) # len() 은 괄호안 데이터의 길이를 표시해준다.
```

Out[33]:

34

In [34]:

```
1 df["시도명"].head()
```

Out[34]:

```
0    경기도
1    서울특별시
2    경기도
3    부산광역시
4    경기도
Name: 시도명, dtype: object
```

In [35]:

```
1 city = df["시도명"].value_counts() #value_counts 카테고리 형태의 데이터 개수 세기
2 city
```

Out[35]:

```
경기도      21374
서울특별시   18943
부산광역시    6473
경상남도     4973
인천광역시    4722
대구광역시    4597
경상북도     4141
전라북도     3894
충청남도     3578
전라남도     3224
광주광역시    3214
대전광역시    3067
충청북도     2677
강원도       2634
울산광역시    1997
제주특별자치도 1095
세종특별자치시  353
Name: 시도명, dtype: int64
```

In [36]:

```
1 city_normalize = df["시도명"].value_counts(normalize=True) #normalize=True 비율 구하기
2 city_normalize
```

Out[36]:

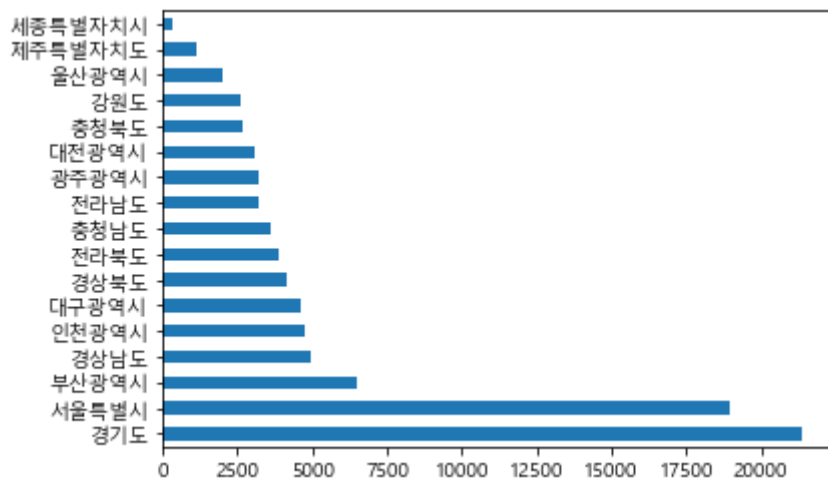
```
경기도      0.234993
서울특별시   0.208266
부산광역시   0.071166
경상남도     0.054675
인천광역시   0.051915
대구광역시   0.050541
경상북도     0.045528
전라북도     0.042812
충청남도     0.039338
전라남도     0.035446
광주광역시   0.035336
대전광역시   0.033720
충청북도     0.029432
강원도       0.028959
울산광역시   0.021956
제주특별자치도 0.012039
세종특별자치시 0.003881
Name: 시도명, dtype: float64
```

In [37]:

```
1 city.plot.barh()
```

Out[37]:

&lt;AxesSubplot:&gt;





In [38]:

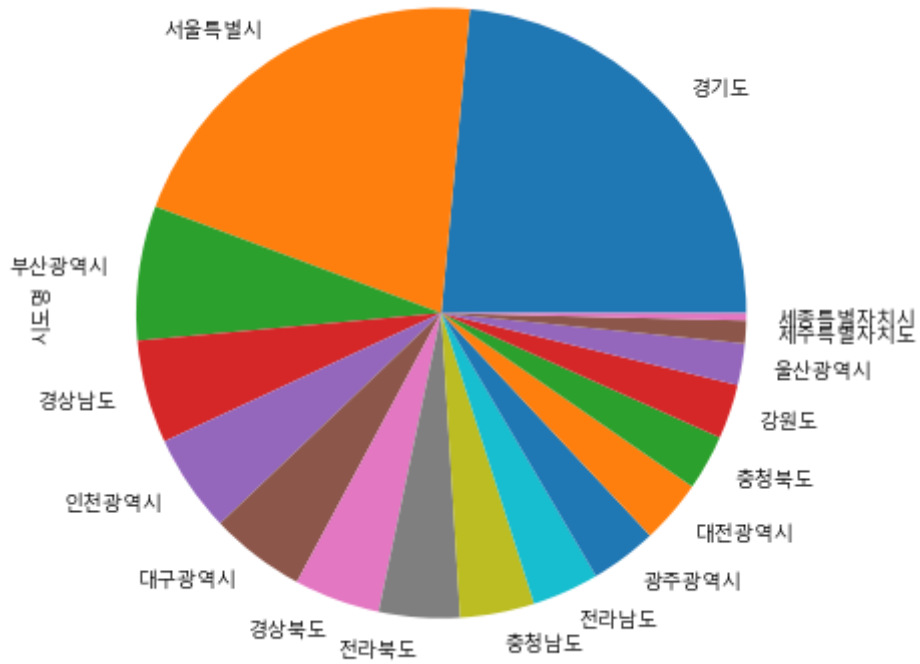
```

1 city_normalize.plot.pie(figsize=(7,7)) #df.plot.pie() 자료의 파이 그래프를 보여준다.
2
3 # 파이 그래프는 자료의 크기 차이를 확연히 보기 힘들기 때문에 그래프 해석에 오류가 생길 수 있어

```

Out [38]:

&lt;AxesSubplot:ylabel='시도명'&gt;

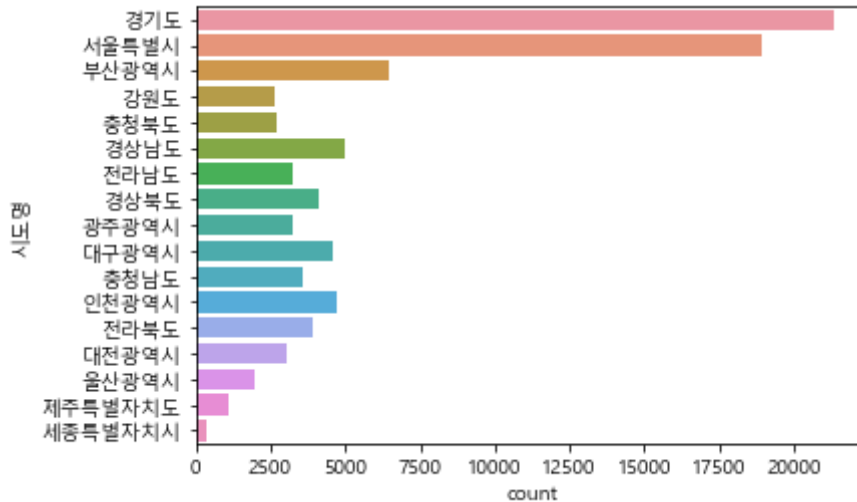


In [39]:

```

1 c = sns.countplot(data=df, y=" 시도명 ") # seaborn 으로 그래프를 그린다.
2
3 # seaborn 에서 색을 부여하기 때문에 미관상 보기 좋다.
4 # data=df 데이터를 df 로 지정한다.
5 # x = 을 하면 세로 막대 그래프가 나오고 y = 을 하면 가로 막대 그래프가 나온다.
6 # 큰 따옴표 안에 컬럼명을 적으면 그 컬럼의 그래프를 그린다.

```



In [40]:

```

1 df["상권업종대분류명"].value_counts()

```

Out[40]:

의료 91335

Name: 상권업종대분류명, dtype: int64

In [41]:

```
1 c = df["상권업종중분류명"].value_counts()
2 c
```

Out[41]:

```
병원          60774
약국/한약방    20923
수의업        5323
유사의료업     3774
의료관련서비스업 541
Name: 상권업종중분류명, dtype: int64
```

In [42]:

```
1 n = df["상권업종중분류명"].value_counts(normalize=True)
2 n
```

Out[42]:

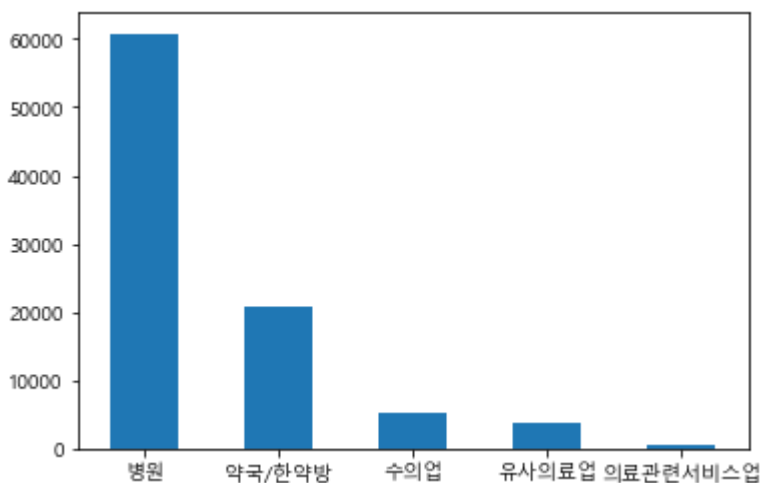
```
병원          0.665397
약국/한약방    0.229080
수의업        0.058280
유사의료업     0.041320
의료관련서비스업 0.005923
Name: 상권업종중분류명, dtype: float64
```

In [43]:

```
1 c.plot.bar(rot=0) #rot=0 글씨를 가로로 돌리기
```

Out[43]:

&lt;AxesSubplot:&gt;

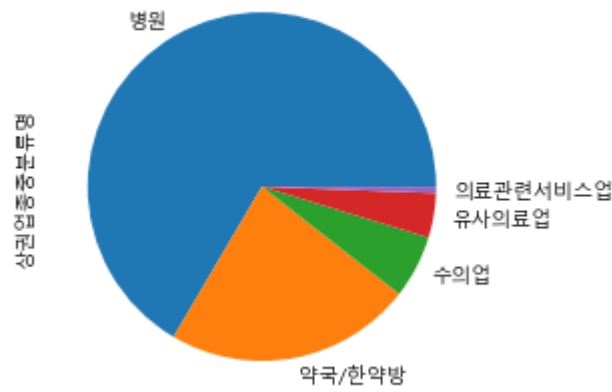


In [44]:

```
1 n.plot.pie()
```

Out[44]:

<AxesSubplot:ylabel='상권업종중분류명'>



In [45]:

```
1 c = df["상권업종소분류명"].value_counts()
2 c
```

Out[45]:

약국	18964
치과의원	13731
한의원	13211
내과/외과	11374
기타병원	4922
일반병원	3385
동물병원	3098
정형/성형외과	2562
소아과	2472
수의업-종합	2216
치과기공소	1724
이비인후과의원	1486
한약방	1442
피부과	1273
산부인과	1116
노인/치매병원	1055
안과의원	1042
비뇨기과	809
종합병원	762
치과병원	756
언어치료	664
유사의료업기타	629
탕제원	517
산후조리원	511
신경외과	421
한방병원	397
척추교정치료	338
침구원	154
혈액원	130
응급구조대	125
조산원	30
점골원	9
수의업-기타	9
재대혈	1

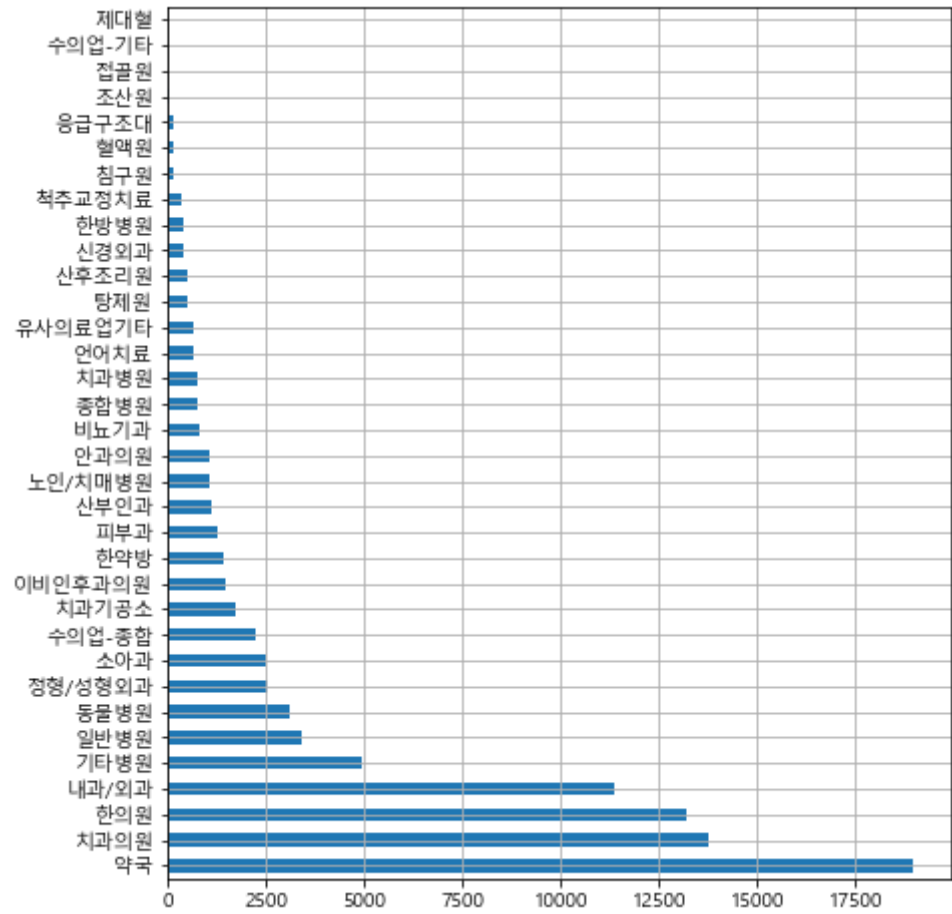
Name: 상권업종소분류명, dtype: int64

In [46]:

```
1 c.plot.barh(figsize=(7,8), grid=True) #grid=True 격자 표시, figsize=(x,y) 그래프 크기 변환 x 7
```

Out[46]:

<AxesSubplot:>



```
1 df_madical = df[df["상권업종중분류명"] == "약국/한약방"].copy() # == 불리언 값으로 원하는 데이터
2 df_madical.head(1)
```

건물본번지	도로명	도로명코드	지번주소	...	시군구코드	시도명	상권업종소분류명	상권업종소분류	상권업종중분류명	상권업종중분류코드	상권업종대분류명	상권업종대분류코드	상호명	상가업소번호
32 41	경기도용인시수지구문정로	414653205024	경기도용인시수지구덕풍천동712-2	...	41465.0	경기도	약국	S02A01	약국/한약방	S02	의료	S	더블유스토어수지점	20364049

```
1 m = df["상권업종대분류명"] == "의료"
2 df.loc[m, "상권업종중분류명"].value_counts() # .loc 행과 열 함께 가져오기
3
4 # df.loc[df["상권업종대분류명"] == "의료", "상권업종중분류명"].value_counts() 같은 결과의 코드
```

```

병원          60774
약국/한약방   20923
수의업        5323
유사의료업    3774
의료관련서비스업 541
Name: 상권업종중분류명, dtype: int64

```

```
1 df_medi = df[df["상권업종중분류명"] == "유사의료업"]
2 df_medi.shape
```

(3774, 29)

In [50]:

```
1 df["상호명"].value_counts().head(10)
```

Out[50]:

```
리원          152
온누리약국    149
경희한의원    141
우리약국      119
충양약국      111
전자담배      98
조은약국      95
건강약국      87
제일약국      79
사랑약국      73
Name: 상호명, dtype: int64
```

In [51]:

```
1 df_medi["상호명"].value_counts().head(10)
```

Out[51]:

```
리원          32
고려수지침    22
대한적십자사  17
헌혈의집      12
고려수지침학회  10
수치과기공소  10
제일치과기공소  9
미소치과기공소  8
아트치과기공소  8
이사랑치과기공소  8
Name: 상호명, dtype: int64
```



In [52]:

```
1 # 판다스 and연산자는 & 우선순위에 의해서 괄호 필요
2 df_seoul_drug = df[(df["상권업종소분류명"] == "약국") & (df["시도명"] == "서울특별시")]
3 print(df_seoul_drug.shape)
4 df_seoul_drug.head(1)
```

(3579, 29)

Out[52]:

	상가업소 번호	상 호 명	상 권 업 종 대 분 류 코 드	상 권 업 종 대 분 류 명	상 권 업 종 중 분 류 코 드	상 권 업 종 중 분 류 명	상 권 업 종 소 분 류 코 드	상 권 업 종 소 분 류 명	시 도 명	시 군 구 코 드	...	지 번 주 소	도로명코드	도로명	건 물 본 번 지	
33	20816709	이즈 타워 약	S	의료	S02	약국/ 한약방	S02A01	약국	서울특별시	11680.0	...	서울특별시강남구역삼동 821	116803122010		서울특별시강남구테헤란로	101

1 rows × 29 columns

In [53]:

```
1 c = df_seoul_drug["시군구명"].value_counts()
2 c.head()
```

Out [53]:

```
강남구      374
동대문구    261
광진구      212
서초구      191
송파구      188
Name: 시군구명, dtype: int64
```

In [54]:

```
1 n = df_seoul_drug["시군구명"].value_counts(normalize=True)
2 n.head()
```

Out [54]:

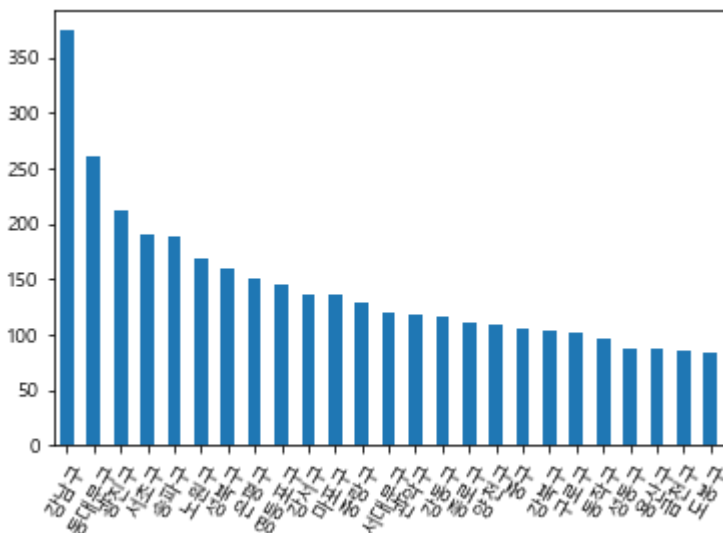
```
강남구      0.104498
동대문구     0.072925
광진구      0.059234
서초구      0.053367
송파구      0.052529
Name: 시군구명, dtype: float64
```

In [55]:

```
1 c.plot.bar(rot=60)
```

Out [55]:

&lt;AxesSubplot:&gt;



In [56]:

```
1 df_seoul_hospital = df[(df["상권업종소분류명"] == "종합병원") & (df["시도명"] == "서울특별시")]
2 df_seoul_hospital
```

Out[56]:

상가업소 번호	상호명	상권업종 대분류코드	상권업종 대분류명	상권업종 중분류코드	상권업종 중분류명	상권 소분류 코드	상권업 소분류명	시 도명	시군구 코드	...	지번 주소	도로명코드	도로명	건물 본번지		
305	25155642	대진 의료 재단	S	의료	S01	병원	S01B01	종합병원	서울특별시	11215.0	...	서울특별시 광진구 중곡1	112153104006	서울특별시 광진구	119	1121510

In [57]:

```
1 df_seoul_hospital["시군구명"].value_counts()
```

Out[57]:

```

강남구      15
영등포구     8
광진구       6
서초구       6
강동구       5
종구         5
송파구       5
강북구       4
도봉구       4
서대문구     4
양천구       4
성북구       3
강서구       2
중랑구       2
종로구       2
동대문구     2
구로구       2
노원구       2
금천구       2
성동구       2
관악구       2
동작구       1
마포구       1
용산구       1
은평구       1
Name: 시군구명, dtype: int64
```

In [58]:

```

1 # str.contains() 컬럼의 특정 단어만 검색 가능
2 df_seoul_hospital.loc[~df_seoul_hospital["상호명"].str.contains("종합병원"), "상호명"].unique()
3
4 # ~ 표시를 하면 원하는 값이 아닌 것만 가져올수 있다.

```

Out [58]:

```

array(['대진의료재단', '홍익병원별관', 'SNUH', '평화드림여의도성모병원의료기매장',
'한양', '백산의료재단친구병원',
'서울보훈병원', '서울성모병원장례식장꽃배달', '서울대학교병원', '알콜중독및정
신질환상담소',
'강남성모병원장례식장꽃배달', '제일병원', '이랜드클리닉', '사랑나눔의료재단',
'우울증센터', '성심의료재단',
'다나의료재단', '서울아산병원신관', '원자력병원장례식장', '국민의원', '고려대
학교구로병원', '학교법인일송학원',
'삼성의료원장례식장', '희명스포츠의학센터인공신장실', '연세대학교의과대학강남
세브란스', '국립정신병원',
'코아클리닉', '수서제일의원', '사랑의원', '한국전력공사부속한일병원', '신촌
연세병원', '창동제일의원',
'영동세브란스병원', '제일성심의원', '삼성의료재단강북삼성대', '서울시립보라매
병원', '서울이의원',
'서울대학교병원비상계획외래', '평화드림서울성모병원의료', '홍익병원', '사랑나
눔의료재단서', '독일의원',
'서울연합의원', '우신향병원', '동부제일병원', '아산재단금강병원', '명곡안연구
소', '아산재단서울중앙병원',
'메디힐특수여객', '삼성생명공익재단삼성서', '성광의료재단차병원', '한국건강관
리협회서울특',
'정해복지부설한신메디피아', '성베드로병원', '성애의료재단', '실로암의원', 'Y&
T성모마취과', '광진성모의원',
'서울현대의원', '이노신경과의원', '송정혜밀리의원', '서울중앙의원', '영남의료
재단', '인제대학교서울백병원',
'한국필의료재단', '세브란스의원', '가톨릭대학교성바오로병원장례식장', '서울연
세의원', '사랑의병원',
'성삼의료재단미즈메디병원', '씨엠충무병원', '성신의원', '원진재단부설녹색병
원', '송파제일의원',
'카톨릭성모의원', '한양성심의원', '관악성모의원', '강남센트럴병원', '우이한솔
의원', '우리들병원',
'서울성모병원어린이집', '건국대학교병원', '서울적십자병원', '북부성모의원',
'한림대학교부속한강성심병원장례식장',
'서울성모병원응급의료센터', '라마르의원', '가톨릭대학교여의도성모병원', '씨엠
병원'], dtype=object)

```

In [59]:

```
1 df_seoul_hospital[df_seoul_hospital["상호명"].str.contains("꽃배달")]
```

Out[59]:

상가업소 번호	상호명	상권업종대분류코드	상권업종대분류명	상권업종중분류코드	상권업종중분류명	상권업종소분류코드	상권업종소분류명	시도명	시군구 코드	...	지번 주소	도로명코드	도로명	건물본 번지
2803	20895655	S	서울성모병원장례식장꽃배달	의료	S01	병원	S01B01	종합병원	11650.0	...	서울특별시서초구반포동551	116504163330	서울특별시서초구사평대로28길	55
4644	22020310	S	강남성모병원장례식장꽃배달	의료	S01	병원	S01B01	종합병원	11650.0	...	서울특별시서초구반포동547-6	116504163242	서울특별시서초구반포대로39길	56

2 rows × 29 columns

In [60]:

```
1 df_seoul_hospital[df_seoul_hospital["상호명"].str.contains("의료기")]
```

Out[60]:

상가업소 번호	상호명	상권업종대분류코드	상권업종대분류명	상권업종중분류코드	상권업종중분류명	상권업종소분류코드	상권업종소분류명	시도명	시군구 코드	...	지번주소	도로명코드	도로명	건물본 번지
1917	평화드림여의도성모병원 의료기매장	S	의료	S01	병원	S01B01	종합병원	서울특별시	11560.0	...	서울특별시영등포구여의도동62	115603118001	서울특별시영등포구63로	10 1

1 rows × 29 columns

In [61]:

```
1 drop_row = df_seoul_hospital[df_seoul_hospital["상호명"].str.contains("꽃배달|의료기|장례식장|상")]
2
3 # index 원하는 데이터의 인덱스 번호만 뽑아로 수 있다.
4
5 drop_row = drop_row.tolist()
6 drop_row
```

Out[61]:

[1917, 2803, 4431, 4644, 7938, 10283, 47008, 60645, 70177]

In [62]:

```
1 # str.endswith 는 str.contains 와 비슷한 기능을 한다.
2 drop_row2 = df_seoul_hospital[df_seoul_hospital["상호명"].str.endswith("의원")].index
3 drop_row2 = drop_row2.tolist()
4 drop_row2
```

Out [62]:

```
[8479,
 12854,
 13715,
 14966,
 16091,
 18047,
 20200,
 20415,
 30706,
 32889,
 34459,
 34720,
 35696,
 37251,
 45120,
 49626,
 51575,
 55133,
 56320,
 56404,
 56688,
 57551,
 62113,
 76508]
```

In [63]:

```
1 drop_row = drop_row + drop_row2
2 len(drop_row)
```

Out [63]:

33

In [64]:

```
1 print(df_seoul_hospital.shape)
2 df_seoul_hospital = df_seoul_hospital.drop(drop_row, axis=0)
3 print(df_seoul_hospital.shape)
```

(91, 29)

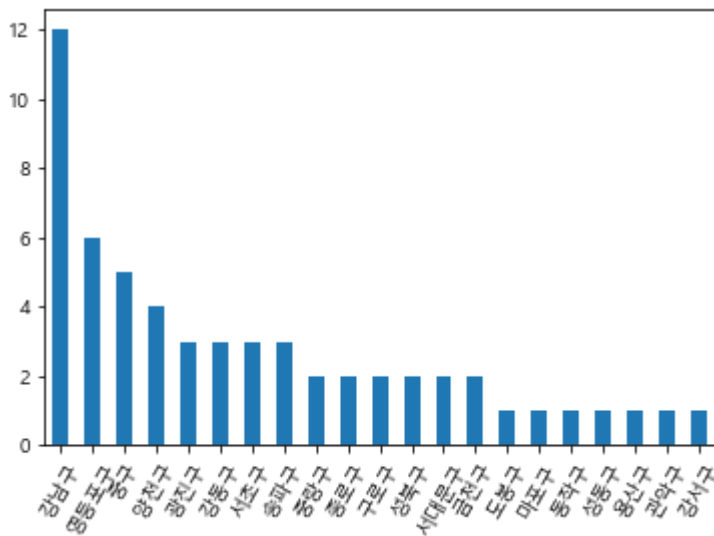
(58, 29)

In [65]:

```
1 df_seoul_hospital["시군구명"].value_counts().plot.bar(rot = 60)
```

Out[65]:

&lt;AxesSubplot:&gt;

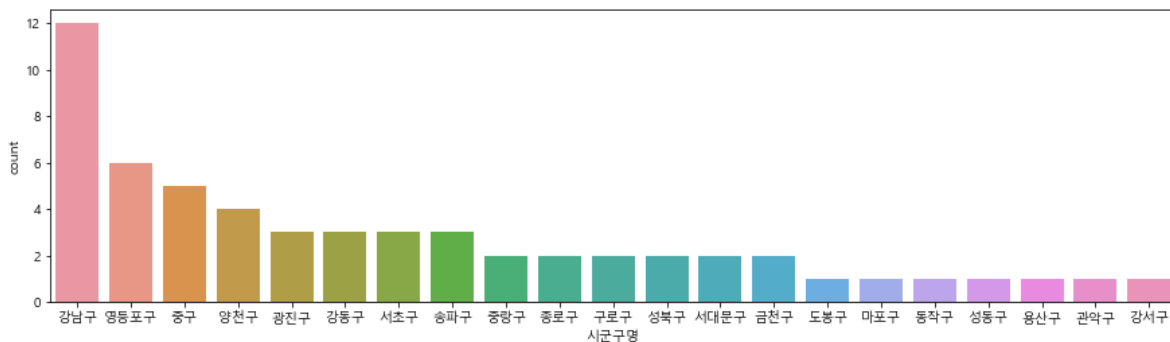


In [66]:

```
1 plt.figure(figsize=(15, 4)) # plt.figure(figsize=(15, 4)) seaborn 그래프 크기변형
2 sns.countplot(data=df_seoul_hospital, x="시군구명", order=df_seoul_hospital["시군구명"].value_c
3 # order=df[" "].value_counts().index 는 크기순 정렬
```

Out[66]:

&lt;AxesSubplot:xlabel='시군구명', ylabel='count'&gt;





In [67]:

```
1 df_seoul_hospital["상호명"].unique()
```

Out [67]:

```
array(['대진의료재단', '홍익병원별관', 'SNUH', '한양', '백산의료재단친구병원', '서울  
보훈병원',  
      '서울대학교병원', '제일병원', '이랜드클리닉', '사랑나눔의료재단', '우울증센  
터', '성심의료재단',  
      '다나의료재단', '서울아산병원신관', '고려대학교구로병원', '학교법인일송학원',  
      '희명스포츠의학센터인공신장실',  
      '연세대학교의과대학강남세브란스', '국립정신병원', '코아클리닉', '한국전력공사  
부속한일병원', '신촌연세병원',  
      '영동세브란스병원', '삼성의료재단강북삼성대', '서울시립보라매병원', '서울대학  
교병원비상계획외래',  
      '평화드림서울성모병원의료', '홍익병원', '사랑나눔의료재단서', '우신향병원',  
      '동부제일병원', '아산재단금강병원',  
      '명곡안연구소', '아산재단서울중앙병원', '메디힐특수어객', '삼성생명공익재단삼  
성서', '성광의료재단차병원',  
      '한국건강관리협회서울특', '정해복지부설한신메디피아', '성베드로병원', '성애의  
료재단', 'Y&T성모마취과',  
      '영남의료재단', '인제대학교서울백병원', '한국필의료재단', '사랑의병원', '성삼  
의료재단미즈메디병원',  
      '씨엠충무병원', '원진재단부설녹색병원', '강남센트럴병원', '우리들병원', '건국  
대학교병원', '서울적십자병원',  
      '서울성모병원응급의료센터', '가톨릭대학교여의도성모병원', '씨엠병원'], dtype=  
object)
```

In [68]:

```
1 df_seoul = df[df["시도명"] == "서울특별시"].copy()
2 df_seoul.shape
```

Out [68]:

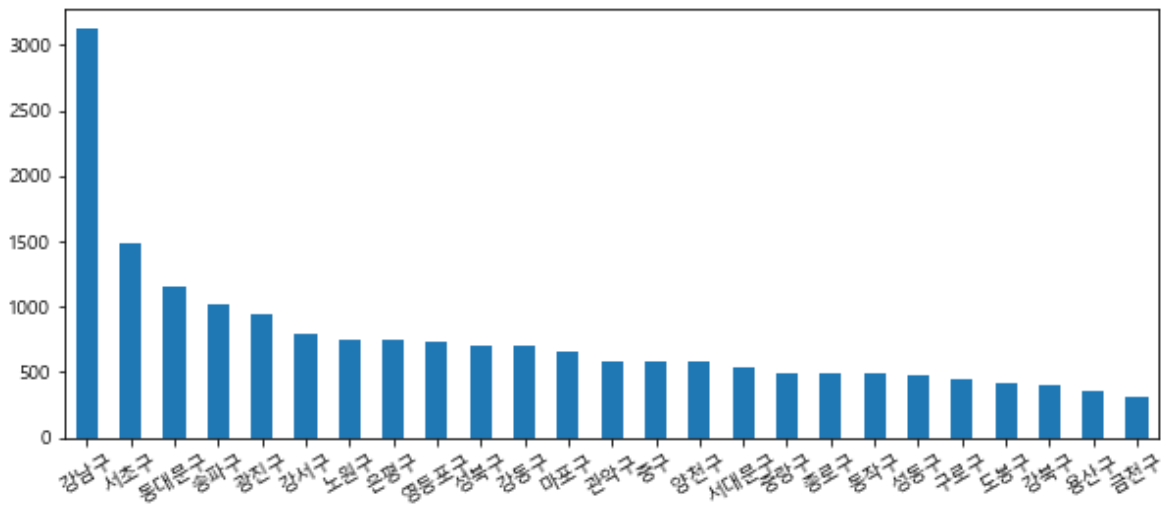
(18943, 29)

In [69]:

```
1 df_seoul["시군구명"].value_counts().plot.bar(figsize=(10,4), rot=30)
```

Out [69]:

&lt;AxesSubplot:&gt;

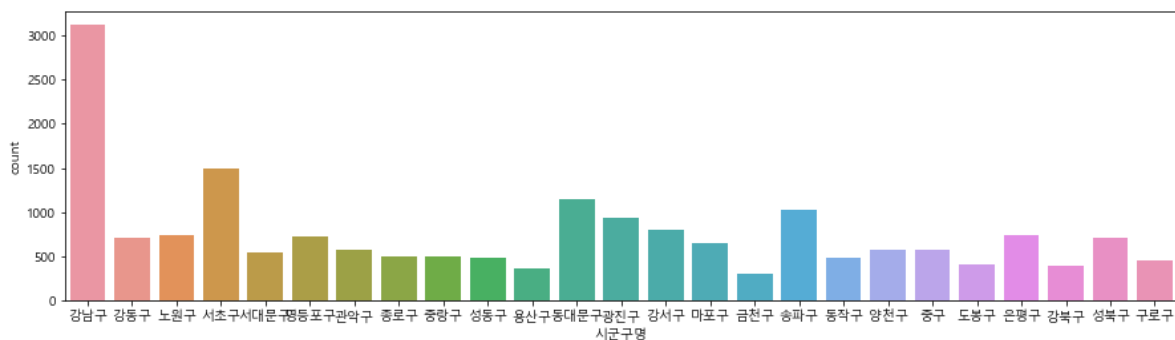


In [70]:

```
1 plt.figure(figsize=(15,4))
2 sns.countplot(data=df_seoul, x="시군구명")
```

Out [70]:

&lt;AxesSubplot:xlabel='시군구명', ylabel='count'&gt;

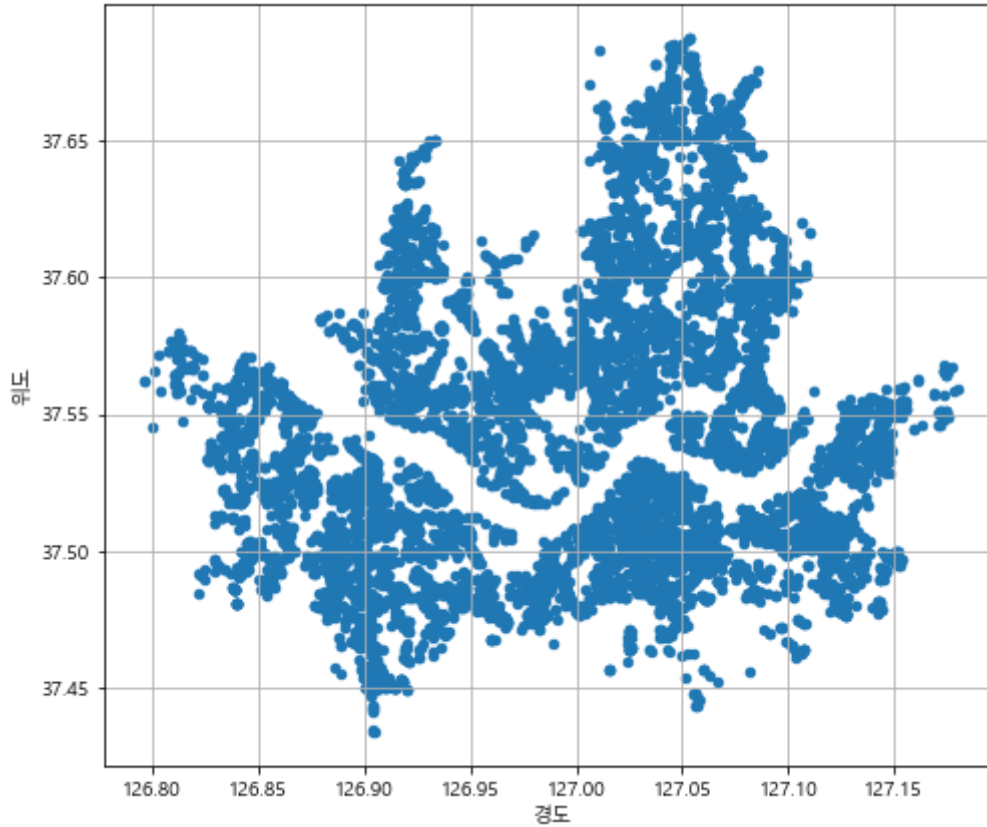


In [71]:

```
1 # plot.scatter(x=" ", y=" ") 는 위도와 경도를 시각화 할 수 있게 한다.  
2 # x=" ", y=" " 을 꼭 넣어주어야 한다.  
3 df_seoul[["경도", "위도", "시군구명"]].plot.scatter(x="경도", y="위도", figsize=(8,7), grid=True)
```

Out[71]:

&lt;AxesSubplot:xlabel='경도', ylabel='위도'&gt;

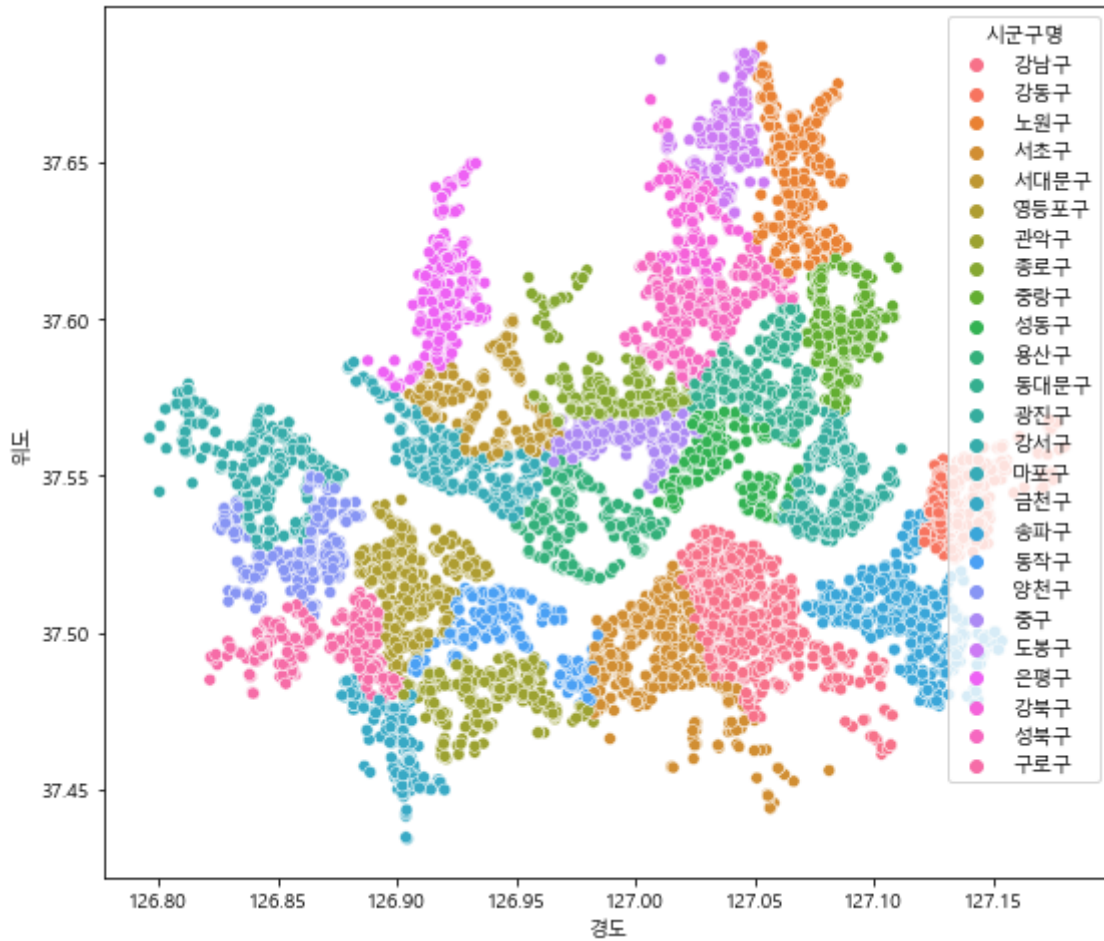


In [72]:

```
1 plt.figure(figsize=(9, 8))
2 sns.scatterplot(data=df_seoul, x="경도", y="위도", hue="시군구명") # hue=" " 은 지정한 컬럼을
```

Out[72]:

&lt;AxesSubplot:xlabel='경도', ylabel='위도'&gt;

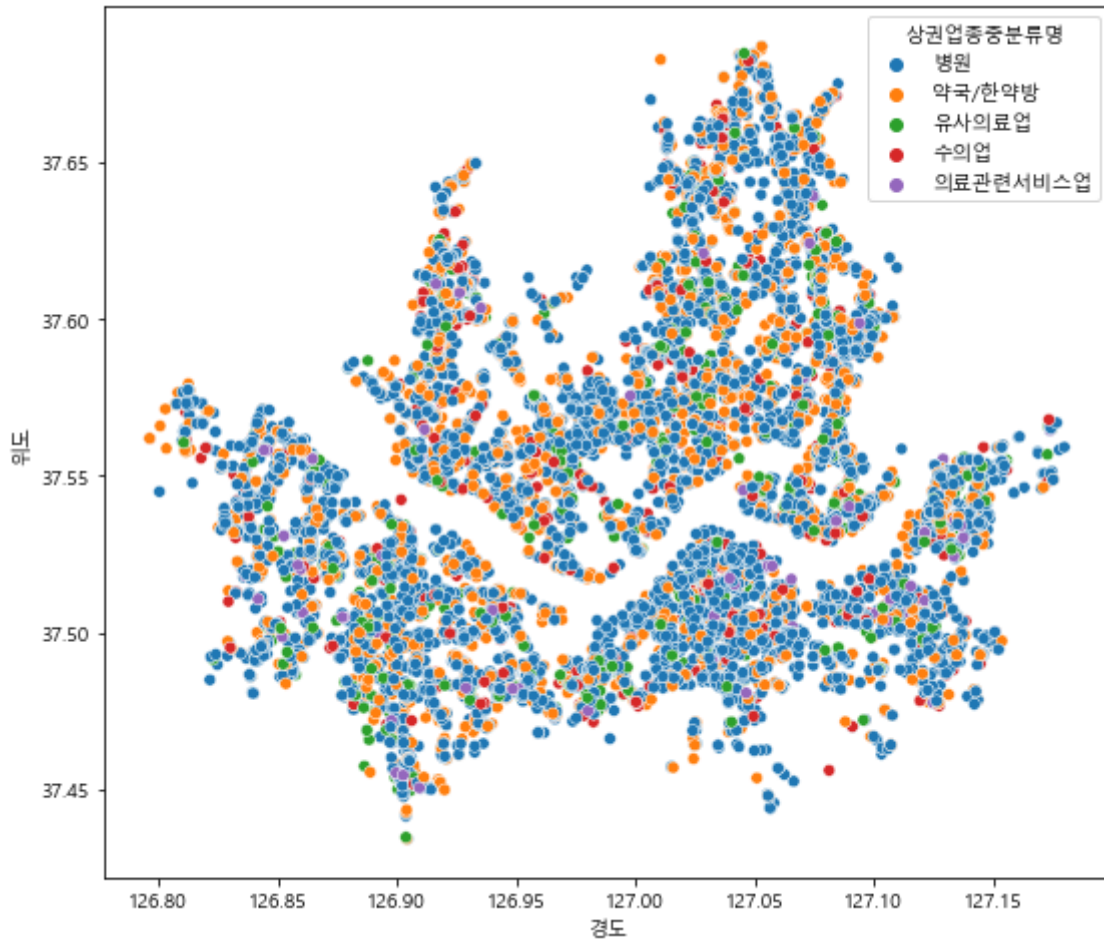


In [73]:

```
1 plt.figure(figsize=(9, 8))
2 sns.scatterplot(data=df_seoul, x="경도", y="위도", hue="상관업종중분류명")
```

Out[73]:

&lt;AxesSubplot: xlabel='경도', ylabel='위도'&gt;

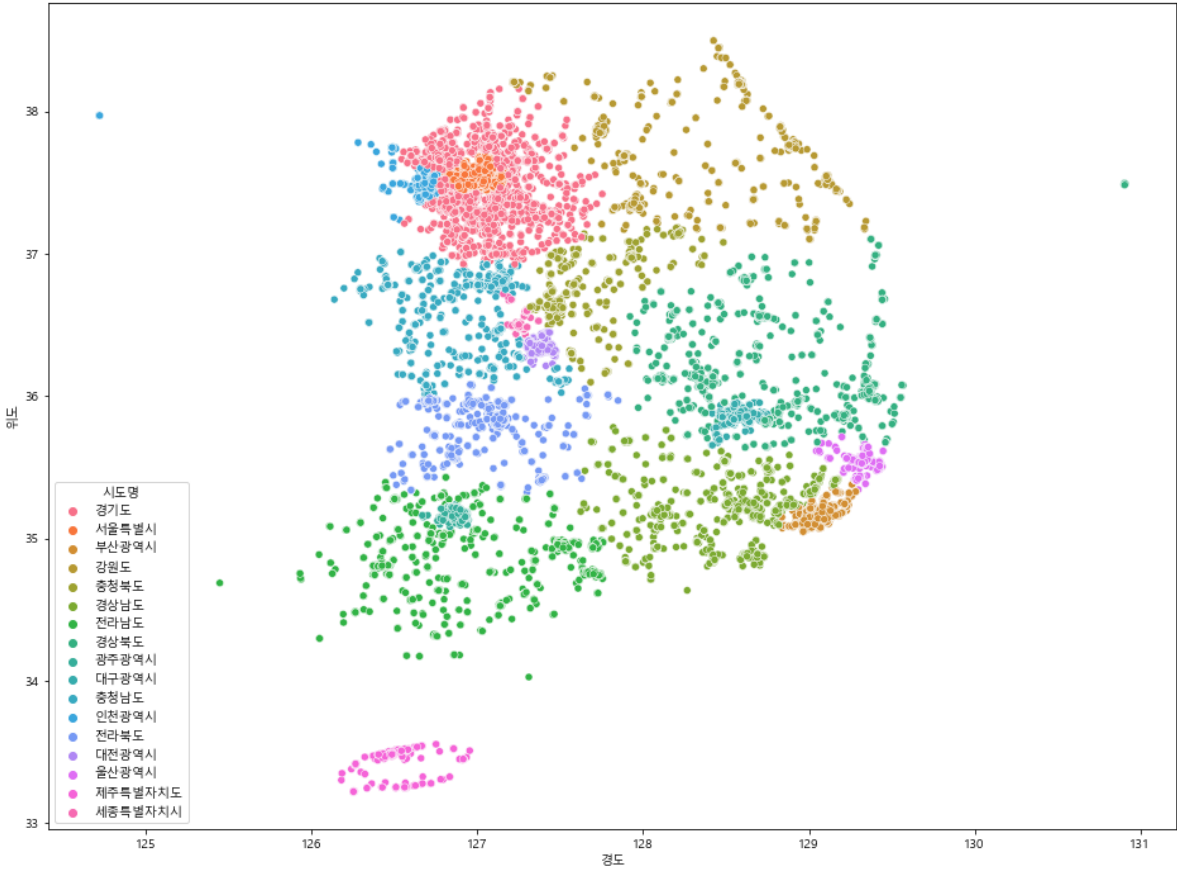


In [74]:

```
1 plt.figure(figsize=(16, 12))
2 sns.scatterplot(data=df, x="경도", y="위도", hue="시도명")
```

Out[74]:

<AxesSubplot:xlabel='경도', ylabel='위도'>



In [75]:

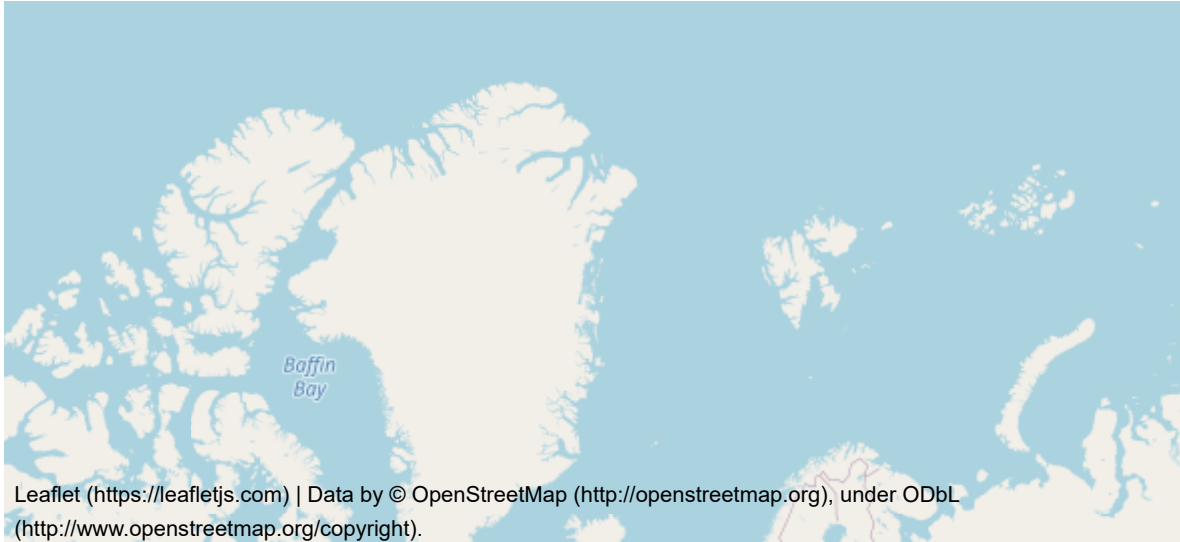
```
1 import folium # 지도 시각화 라이브러리
```

Out[75]:

Make this Notebook Trusted to load map: File -&gt; Trust Notebook

+

-



In [78]:

```
1 df_seoul_hospital["위도"].mean()  
2 df_seoul_hospital["경도"].mean()
```

Out[78]:

126.9963589356625

In [83]:

```
1 df_seoul_hospital.tail(1)
```

Out[83]:

	상가업소 번호	상호명	상권업종대분류코드	상권업종대분류명	상권업종중분류코드	상권업종중분류명	상권업종소분류코드	상권업종소분류명	시도명	시군구 코드	...	지번 주소	도로명코드	도로명	건 본
90788	16162338	성베드로병원	S	의료	S01	병원	S01B01	종합병원	서울특별시	11680.0	...	서울특별시 강남구 곡동 910-27	116802000003	서울특별시 강남구 남부순환로	264

1 rows × 29 columns



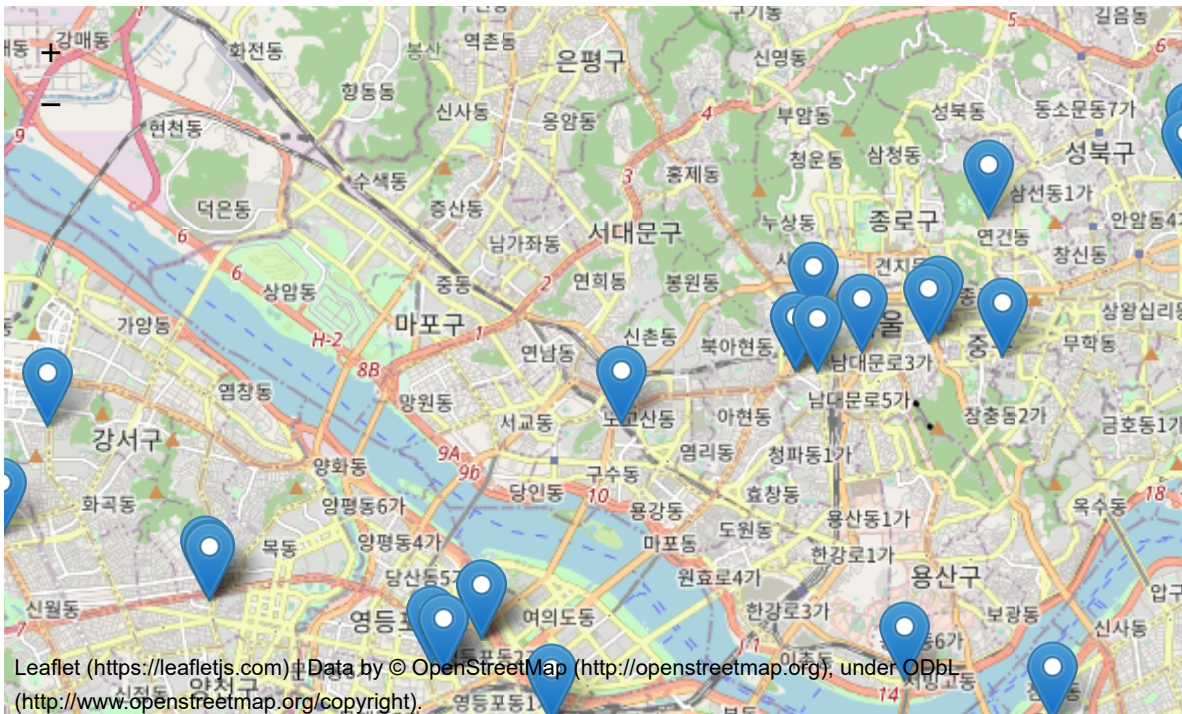
In [94]:

```

1 map = folium.Map(location=[df_seoul_hospital["위도"].mean(), df_seoul_hospital["경도"].mean()],
2                   zoom_start=12)
3
4 # folium.Map(location=[x, y]) x, y를 중심으로 지도를 보여준다.
5 # zoom_start= 지정한 값만큼 지도 크기를 키운다
6
7 for n in df_seoul_hospital.index:
8     name = df_seoul_hospital.loc[n, "상호명"]
9     address = df_seoul_hospital.loc[n, "도로명주소"]
10    popup = f"{name}-{address}"
11    location = df_seoul_hospital.loc[n, "위도"], df_seoul_hospital.loc[n, "경도"]
12    folium.Marker(
13        location = location,
14        popup = popup,
15    ).add_to(map)
16 map
17
18 # df에 있는 데이터들을 popup을 해준다.

```

Out[94]:



In [ ]:

1