In [1]:

```
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import pandas as pd
import csv
```

In [2]:

```
# set the file to a variable, here we use 'a'
a = pd.read_csv(r"C:\Users\mdmoh\Desktop\master.csv")
```

In [3]:

```
# look at the chart on the jupyter environment
a.head(5)
```

Out[3]:

	country	year	sex	age	suicides_no	population	suicides/100k pop	country-year	HDI for year	gdp_for_year ($)	gdp_per_capita ($)	generation
0	Albania	1987	male	15-24 years	21	312900	6.71	Albania1987	NaN	2,156,624,900	796	Generation X
1	Albania	1987	male	35-54 years	16	308000	5.19	Albania1987	NaN	2,156,624,900	796	Silent
2	Albania	1987	female	15-24 years	14	289700	4.83	Albania1987	NaN	2,156,624,900	796	Generation X
3	Albania	1987	male	75+ years	1	21800	4.59	Albania1987	NaN	2,156,624,900	796	G.I. Generation
4	Albania	1987	male	25-34 years	9	274300	3.28	Albania1987	NaN	2,156,624,900	796	Boomers

In [4]:

```
#look at the summary of the data
a.describe()
```

Out[4]:

	year	suicides_no	population	suicides/100k pop	HDI for year	gdp_per_capita ($)
count	27820.000000	27820.000000	2.782000e+04	27820.000000	8364.000000	27820.000000
mean	2001.258375	242.574407	1.844794e+06	12.816097	0.776601	16866.464414
std	8.469055	902.047917	3.911779e+06	18.961511	0.093367	18887.576472
min	1985.000000	0.000000	2.780000e+02	0.000000	0.483000	251.000000
25%	1995.000000	3.000000	9.749850e+04	0.920000	0.713000	3447.000000
50%	2002.000000	25.000000	4.301500e+05	5.990000	0.779000	9372.000000
75%	2008.000000	131.000000	1.486143e+06	16.620000	0.855000	24874.000000
max	2016.000000	22338.000000	4.380521e+07	224.970000	0.944000	126352.000000

In [5]:

```
a.info()
''''Here we can determine the type of data in each column. This is important for when doing statistical analysis for
example we can not analyze columns with whoes data type is 'object'/'''
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27820 entries, 0 to 27819
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   country             27820 non-null  object 
 1   year                27820 non-null  int64  
 2   sex                 27820 non-null  object 
 3   age                 27820 non-null  object 
 4   suicides_no         27820 non-null  int64  
 5   population          27820 non-null  int64  
 6   suicides/100k pop   27820 non-null  float64
 7   country-year        27820 non-null  object 
 8   HDI for year        8364 non-null   float64
 9    gdp_for_year ($)   27820 non-null  object 
 10  gdp_per_capita ($)  27820 non-null  int64  
 11  generation          27820 non-null  object 
dtypes: float64(2), int64(4), object(6)
memory usage: 2.5+ MB
```

Out[5]:

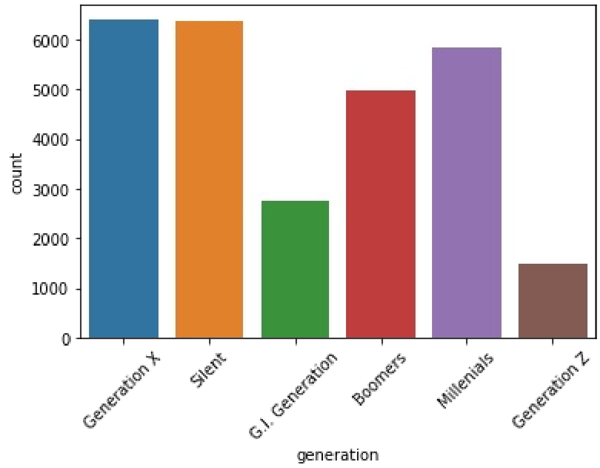
```
"''Here we can determine the type of data in each column. This is important for when doing statistic
al analysis for \nexample we can not analyze columns with whoes data type is 'object'/"
```

In [39]:

```
gen_plot = sns.countplot('generation',data=adata)
gen_plot.set_xticklabels(gen_plot.get_xticklabels(), rotation=45)
```

Out[39]:

```
[Text(0, 0, 'Generation X'),
 Text(0, 0, 'Silent'),
 Text(0, 0, 'G.I. Generation'),
 Text(0, 0, 'Boomers'),
 Text(0, 0, 'Millenials'),
 Text(0, 0, 'Generation Z')]
```



In [ ]:

```

```

In [6]:

```
''''Here we drop the column 'HDI for year' as lot of this data is missing. Refrence the 'count' row above. Every other
column has a count of 27820.000000, while 'HDI for year' has a count of 8364.000000. We then make a new database with out
the HDI for year colummn'''

adata = a.drop(columns ='HDI for year')
```

In [7]:

```
#look at the unique values per column. Notice this does not include HDI for year
adata.nunique()
```

Out[7]:

```
country                 101
year                     32
sex                       2
age                       6
suicides_no            2084
population            25564
suicides/100k pop      5298
country-year           2321
 gdp_for_year ($)      2321
gdp_per_capita ($)     2233
generation                6
dtype: int64
```

In [8]:

```
'''Here we look at the unique countries which makes up this data base and the occurance of these unique countries
.
Delete the '#' from in fromt of the print fuction to see 'frequency' '''

(unique, counts) = np.unique(adata['country'], return_counts=True)
frequency = np.asarray((unique, counts)).T
#print(frequency)
```

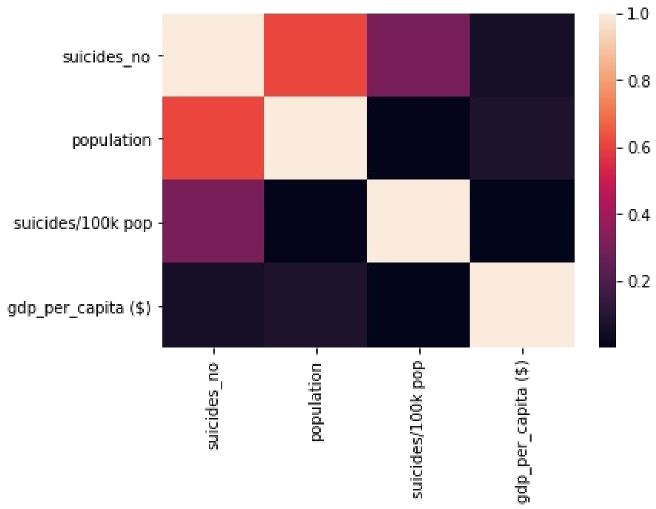
In [9]:

```
print(adata[['suicides_no','population','suicides/100k pop','gdp_per_capita ($)']].corr())
sns.heatmap((adata[['suicides_no','population','suicides/100k pop','gdp_per_capita ($)']].corr()))
#The only correlation that seems to be significient is that the higher the population the more suicides that coun
try has. However, thats given so it is not really helpful. ''''
```

```
                    suicides_no  population  suicides/100k pop  \
suicides_no            1.000000    0.616162           0.306604   
population             0.616162    1.000000           0.008285   
suicides/100k pop      0.306604    0.008285           1.000000   
gdp_per_capita ($)     0.061330    0.081510           0.001785   

                    gdp_per_capita ($)  
suicides_no                   0.061330  
population                    0.081510  
suicides/100k pop             0.001785  
gdp_per_capita ($)            1.000000  
```

Out[9]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x16fc860f688>
```

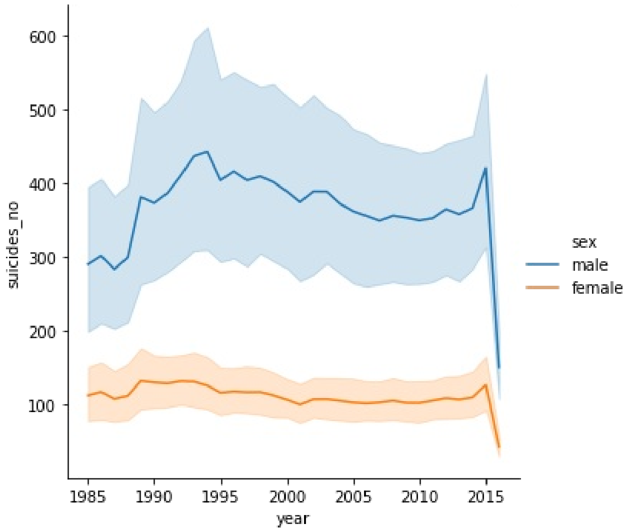


In [10]:

```
sns.relplot(x='year',y='suicides_no', hue='sex', kind = 'line', data = adata )
#here we have a line plot of the mean suicides number for all the countries by year
```

Out[10]:

```
<seaborn.axisgrid.FacetGrid at 0x16fc6700a48>
```

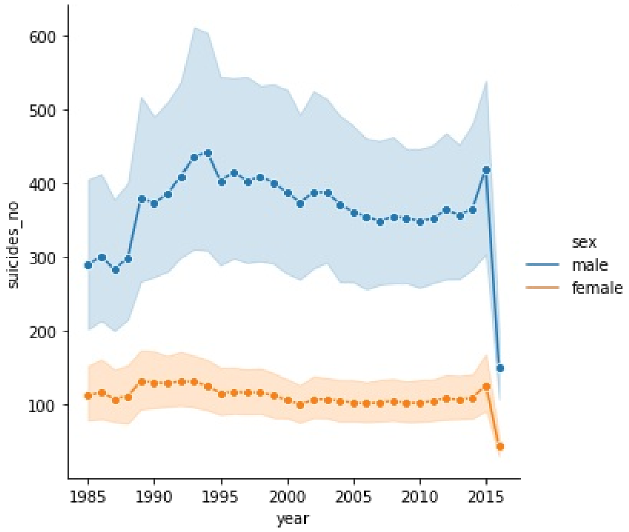


In [11]:

```
'''line plots can be tricky because its often diffuclt to see the actual markers. Looking at the graph above for example
one may come to the false conclusion that suicide rates have gradually declined from 2014 - 2015, however, looking at the
markers below we can see this is not true. There a single data point which is skewing the data, thus, this is an outlier.
We can take out the outliers by considering the data in an appropriate range: multiple x interquartile range (IQR)'''

sns.relplot(x='year',y='suicides_no', hue='sex', kind = 'line', marker='o', data = adata )
```

Out[11]:

```
<seaborn.axisgrid.FacetGrid at 0x16fca44fec8>
```



In [12]:

```
import scipy.stats

def find_remove_outlier_iqr(data_sample):
    q1 = np.percentile(data_sample, 25)
    q3 = np.percentile(data_sample, 75)

    iqr = q3 - q1

    cutoff = iqr * 1.5

    lower, upper = q1-cutoff, q3+cutoff

    outliers =[]
    outliers_removed = []
    for x in data_sample:
        if x < lower or x > upper:
            outliers.append(x)
        if x > lower and x < upper:
            outliers_removed.append(x)
    return outliers_removed
```

In [13]:

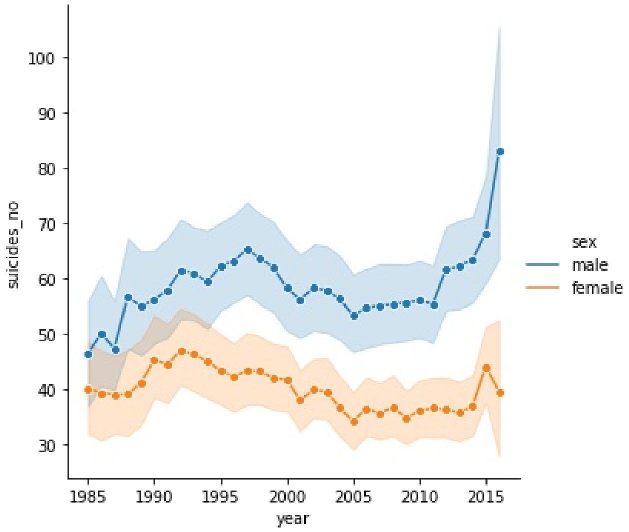
```
outliers_removed = find_remove_outlier_iqr(adata["suicides_no"])
out_df = adata[adata["suicides_no"].isin(outliers_removed)]
```

In [14]:

```
sns.relplot(x='year',y='suicides_no', hue='sex', kind = 'line', marker='o', data = out_df)
#After gettign rid of the outliers we can see how our mean suicide number for each year changes
```

Out[14]:

<seaborn.axisgrid.FacetGrid at 0x16fca500588>



In [15]:

```
#Lets look at which country in the list has the higest suicide rates
a=adata.sort_values(by='suicides_no', ascending=False)
a.head(10)
```

Out[15]:

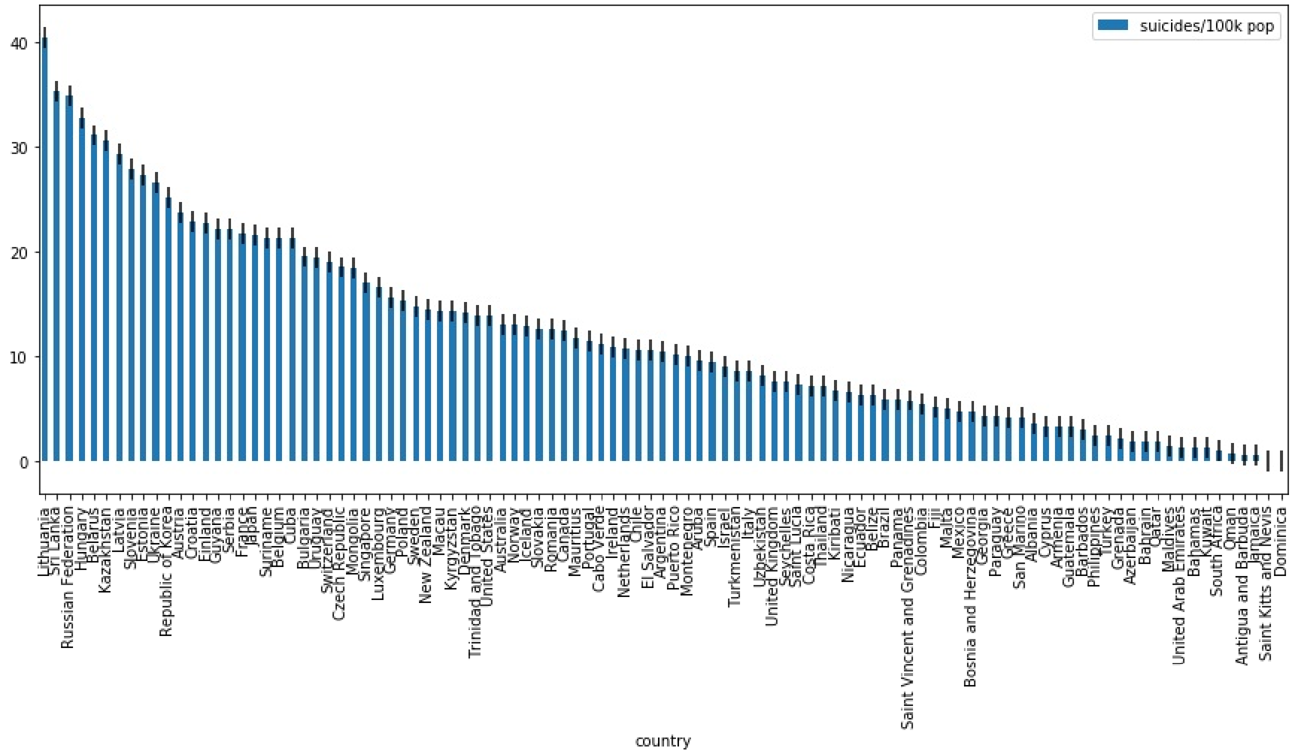
	country	year	sex	age	suicides_no	population	suicides/100k pop	country-year	gdp_for_year ($)	gdp_per_capita ($)	generation
20996	Russian Federation	1994	male	35-54 years	22338	19044200	117.30	Russian Federation1994	395,077,301,248	2853	Boomers
21008	Russian Federation	1995	male	35-54 years	21706	19249600	112.76	Russian Federation1995	395,531,066,563	2844	Boomers
21080	Russian Federation	2001	male	35-54 years	21262	21476420	99.00	Russian Federation2001	306,602,673,980	2229	Boomers
21068	Russian Federation	2000	male	35-54 years	21063	21378098	98.53	Russian Federation2000	259,708,496,267	1879	Boomers
21057	Russian Federation	1999	male	35-54 years	20705	21016400	98.52	Russian Federation1999	195,905,767,669	1412	Boomers
21020	Russian Federation	1996	male	35-54 years	20562	19507100	105.41	Russian Federation1996	391,719,993,757	2813	Boomers
20984	Russian Federation	1993	male	35-54 years	20256	18908000	107.13	Russian Federation1993	435,083,713,851	3160	Boomers
21092	Russian Federation	2002	male	35-54 years	20119	21320535	94.36	Russian Federation2002	345,110,438,692	2527	Boomers
21033	Russian Federation	1997	male	35-54 years	18973	19913400	95.28	Russian Federation1997	404,926,534,140	2907	Boomers
21105	Russian Federation	2003	male	35-54 years	18681	21007346	88.93	Russian Federation2003	430,347,770,732	3141	Boomers

In [16]:

```
countrydata = adata.groupby('country').mean()[['suicides/100k pop']]
countrydata_sorted = countrydata.sort_values(by='suicides/100k pop', ascending=False)
countrydata_sorted.plot.bar(y='suicides/100k pop',yerr = True, figsize =(15,6))

#sns.scatterplot(x='suicides_no', y='gdp_per_capita ($)', hue= 'country', size='population', legend =None, alpha
= .5, data = adata)
```

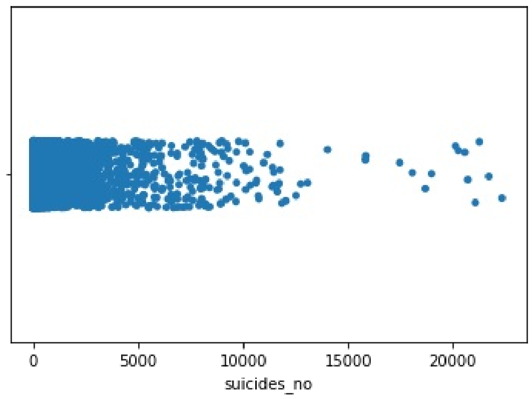
Out[16]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x16fca593cc8>
```



In [19]:

```
sns.stripplot(x='suicides_no', data=adata)
plt.show()
```



In [17]:

```
# do the stacked bar graph here
```

In [ ]:

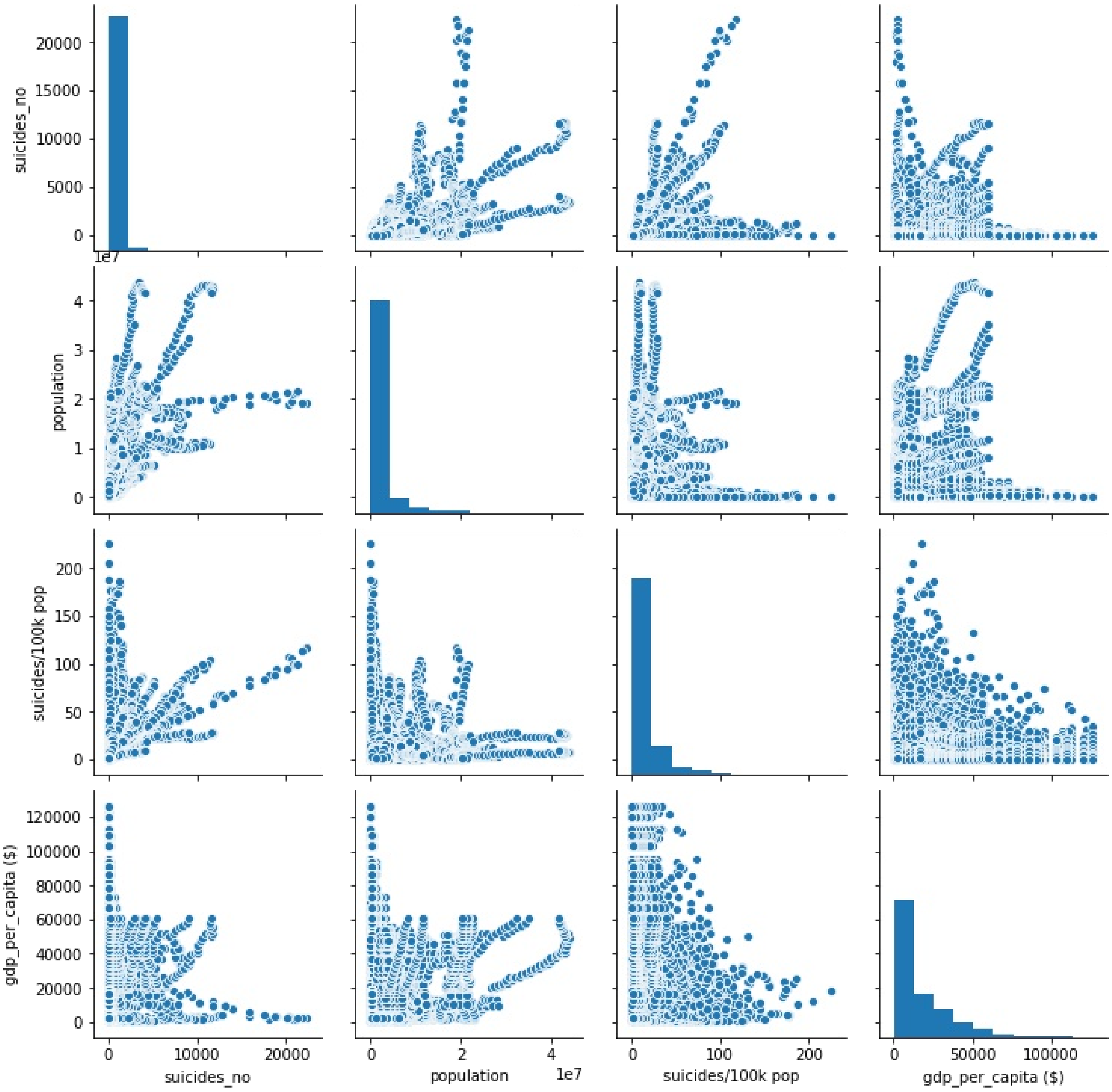
```

```

In [18]:

```
#We can look at the correlation of each column to one another via scatter plots with pairplot
sns.pairplot(adata[['suicides_no','population','suicides/100k pop','gdp_per_capita ($)']], diag_kind='hist')
```

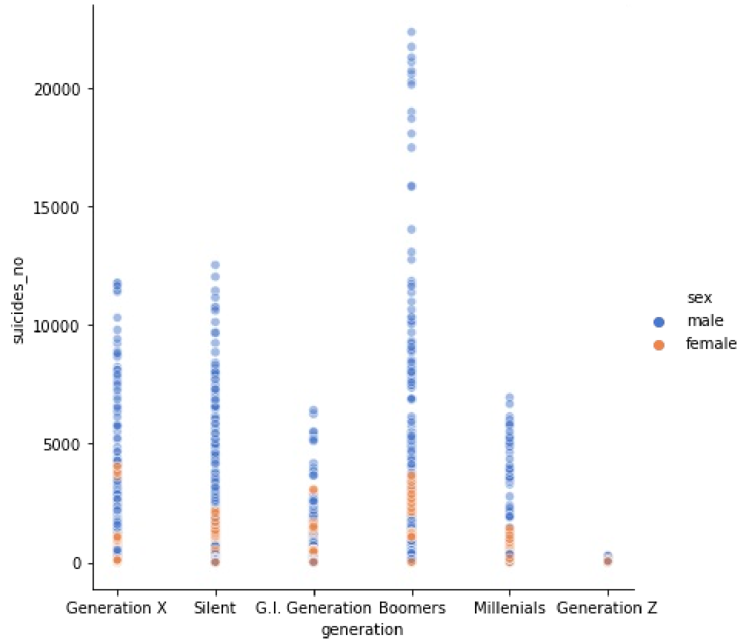
Out[18]:

```
<seaborn.axisgrid.PairGrid at 0x16fca125dc8>
```



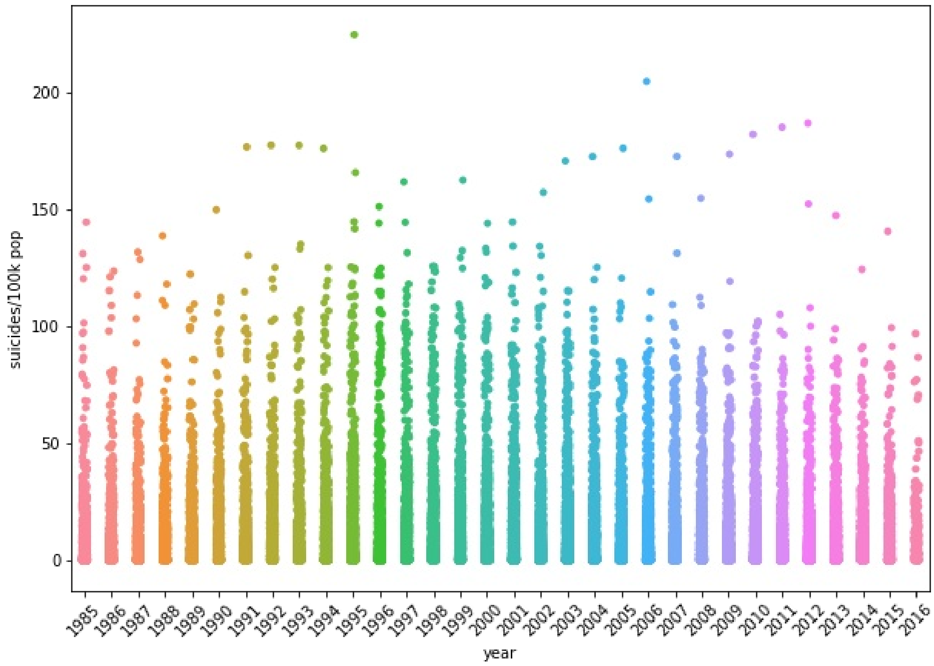
In [20]:

```
sns.relplot(x="generation",y="suicides_no",hue="sex",
            sizes=(40, 400), alpha=.5, palette="muted",
            height=6, data=adata)
plt.show()
```



In [30]:

```
plt.figure(figsize=(10,7))
sns.stripplot(x="year",y='suicides/100k pop',data=adata)
plt.xticks(rotation=45)
plt.show()
```



In [ ]: