

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

```
df=pd.read_csv('/content/suciderate.csv')
```

```
df
```



	State	Year	Type_code	Type	Gender	Age_group	Total
0	A & N Islands	2001	Causes	Illness (Aids/STD)	Female	0-14	0.0
1	A & N Islands	2001	Causes	Bankruptcy or Sudden change in Economic	Female	0-14	0.0
2	A & N Islands	2001	Causes	Cancellation/Non-Settlement of Marriage	Female	0-14	0.0
3	A & N Islands	2001	Causes	Physical Abuse (Rape/Incest Etc.)	Female	0-14	0.0
4	A & N Islands	2001	Causes	Dowry Dispute	Female	0-14	0.0
...
210589	Tripura	2001	Causes	Love Affairs	Female	45-59	3.0
210590	Tripura	2001	Causes	Ideological Causes/Hero Worshipping	Female	45-59	0.0
210591	Tripura	2001	Causes	Insanity/Mental Illness	Female	45-59	4.0

```
df.columns
```



```
Index(['State', 'Year', 'Type_code', 'Type', 'Gender', 'Age_group', 'Total'],
      dtype='object')
```

```
df.info()
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210594 entries, 0 to 210593
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  -
0   State       210594 non-null object
1   Year        210594 non-null int64
2   Type_code   210594 non-null object
```

```

3   Type      210594 non-null object
4   Gender    210593 non-null object
5   Age_group 210593 non-null object
6   Total     210593 non-null float64
dtypes: float64(1), int64(1), object(5)
memory usage: 11.2+ MB

```

```
df.isnull().sum()
```

```

⇒ State      0
   Year      0
   Type_code  0
   Type      0
   Gender     1
   Age_group  1
   Total      1
dtype: int64

```

```
df.shape
```

```
⇒ (210594, 7)
```

```
df['Total'].describe()
```

```

⇒ count      210593.000000
   mean         56.624375
   std          838.374640
   min           0.000000
   25%           0.000000
   50%           0.000000
   75%           5.000000
   max          63343.000000
   Name: Total, dtype: float64

```

```
df['State'].unique()
```

```

⇒ array(['A & N Islands', 'Andhra Pradesh', 'Arunachal Pradesh', 'Assam',
        'Bihar', 'Chandigarh', 'Chhattisgarh', 'D & N Haveli',
        'Daman & Diu', 'Delhi (Ut)', 'Goa', 'Gujarat', 'Haryana',
        'Himachal Pradesh', 'Jammu & Kashmir', 'Jharkhand', 'Karnataka',
        'Kerala', 'Lakshadweep', 'Madhya Pradesh', 'Maharashtra',
        'Manipur', 'Meghalaya', 'Mizoram', 'Nagaland', 'Odisha',
        'Puducherry', 'Punjab', 'Rajasthan', 'Sikkim', 'Tamil Nadu',
        'Total (All India)', 'Total (States)', 'Total (Uts)', 'Tripura'],
        dtype=object)

```

```
df['Year'].unique()
```

```

⇒ array([2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011,
        2012])

```

```
df['Type_code'].unique()
```

```
⇒ array(['Causes', 'Education_Status', 'Means_adopted',  
        'Professional_Profile', 'Social_Status'], dtype=object)
```

```
df['Type'].unique()
```

```
⇒ array(['Illness (Aids/STD)', 'Bankruptcy or Sudden change in Economic',  
        'Cancellation/Non-Settlement of Marriage',  
        'Physical Abuse (Rape/Incest Etc.)', 'Dowry Dispute',  
        'Family Problems', 'Ideological Causes/Hero Worshipping',  
        'Other Prolonged Illness', 'Property Dispute',  
        'Fall in Social Reputation', 'Illegitimate Pregnancy',  
        'Failure in Examination', 'Insanity/Mental Illness',  
        'Love Affairs', 'Professional/Career Problem', 'Divorce',  
        'Drug Abuse/Addiction', 'Not having Children(Barrenness/Impotency',  
        'Causes Not known', 'Unemployment',  
        'Other Causes (Please Specity)', 'Poverty', 'Death of Dear Person',  
        'Cancer', 'Suspected/Illicit Relation', 'Paralysis', 'Diploma',  
        'No Education', 'Post Graduate and Above', 'Middle', 'Graduate',  
        'Hr. Secondary/Intermediate/Pre-Universit', 'Primary',  
        'Matriculate/Secondary', 'By Consuming Insecticides', 'By Hanging',  
        'By Jumping from (Other sites)', 'By touching electric wires',  
        'By Machine', 'By Fire/Self Immolation',  
        'By Jumping off Moving Vehicles/Trains',  
        'By Other means (please specify)', 'By Self Infliction of injury',  
        'By Over Alcoholism', 'By Consuming Other Poison',  
        'By coming under running vehicles/trains',  
        'By Overdose of sleeping pills', 'By Jumping from (Building)',  
        'By Drowning', 'By Fire-Arms', 'Retired Person', 'Unemployed',  
        'Public Sector Undertaking', 'Service (Private)', 'House Wife',  
        'Self-employed (Business activity)', 'Professional Activity',  
        'Student', 'Others (Please Specify)',  
        'Farming/Agriculture Activity', 'Service (Government)', 'Married',  
        'Seperated', 'Widowed/Widower', 'Divorcee', 'Never Married',  
        'Bankruptcy or Sudden change in Economic Status',  
        'Not having Children (Barrenness/Impotency', 'By Other means',  
        'Dowry Dispu'], dtype=object)
```

```
df['Gender'].unique()
```

```
⇒ array(['Female', 'Male', nan], dtype=object)
```

```
df['Age_group'].unique()
```

```
⇒ array(['0-14', '15-29', '30-44', '45-59', '60+', '0-100+', nan],  
        dtype=object)
```

```
df = df.drop(df[(df.State == 'Total (Uts)') |(df.State == 'Total (All India)') |  
              (df.State == 'Total (States)')].index)
```

```
len(df[df['Total'] == 0])
```

122921

```
df = df.drop(df[df.Total == 0].index)
```

```
df
```



	State	Year	Type_code	Type	Gender	Age_group	Total
13	A & N Islands	2001	Causes	Love Affairs	Female	0-14	1.0
20	A & N Islands	2001	Causes	Other Causes (Please Specity)	Female	0-14	1.0
32	A & N Islands	2001	Causes	Other Prolonged Illness	Male	0-14	1.0
47	A & N Islands	2001	Causes	Failure in Examination	Male	0-14	1.0
54	A & N Islands	2001	Causes	Other Prolonged Illness	Female	15-29	8.0
...
210587	Tripura	2001	Causes	Poverty	Female	45-59	3.0
210589	Tripura	2001	Causes	Love Affairs	Female	45-59	3.0
210591	Tripura	2001	Causes	Insanity/Mental Illness	Female	45-59	4.0



Next steps:

[Generate code with df](#)



[View recommended plots](#)

```
statewise_total_suicide = df.groupby("State")["Total"].sum()
max = statewise_total_suicide.idxmax()
```

```
print("The state which has most number of suicides:", max)
```



The state which has most number of suicides: Maharashtra

```
yearwise_total_suicide = df.groupby("Year")["Total"].sum()
max = yearwise_total_suicide.idxmax()
```

```
print('Year with the most number of suicides:', max)
```



Year with the most number of suicides: 2012

```
sns.set_style('darkgrid')
matplotlib.rcParams['font.size'] = 12
matplotlib.rcParams['figure.figsize'] = (9, 5)
matplotlib.rcParams['figure.facecolor'] = '#00000000'

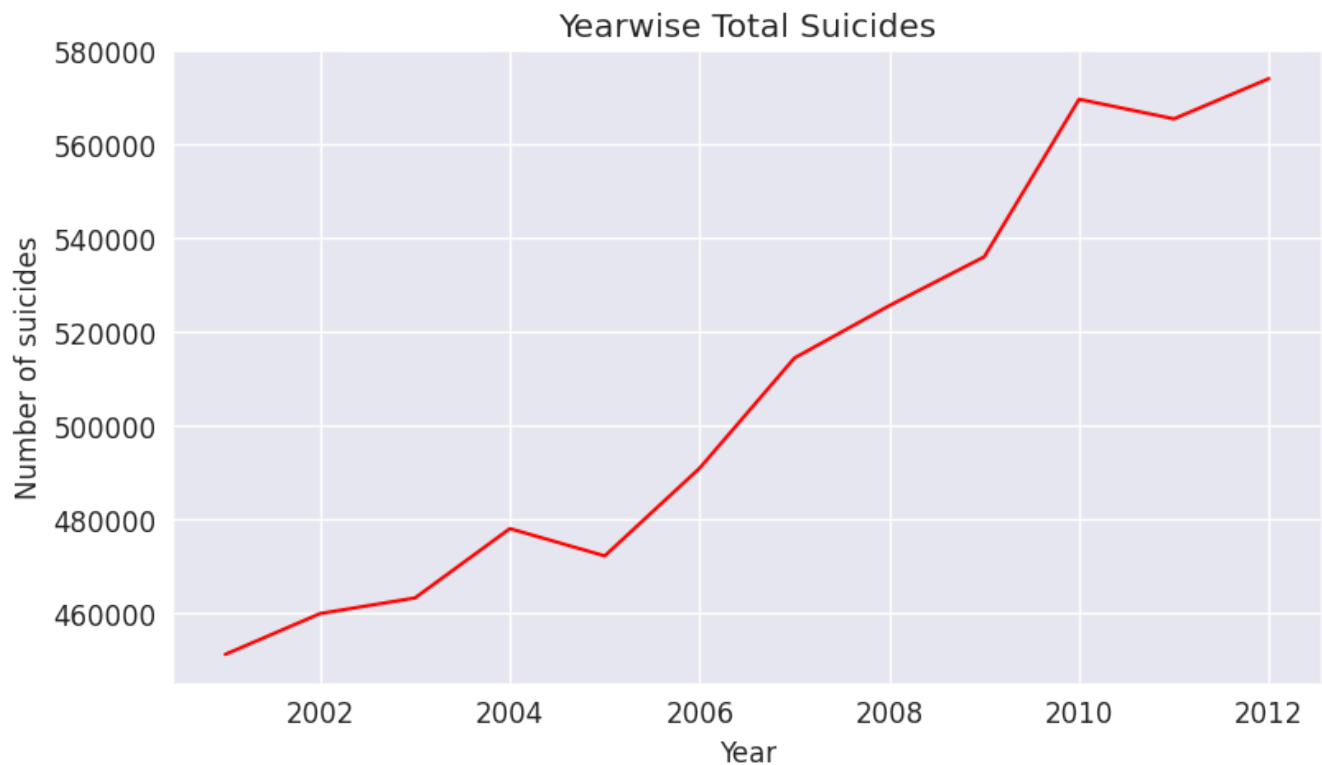
yearwise_total_suicide = df.groupby("Year")["Total"].sum()
print('Total number of suicides in each year: ', yearwise_total_suicide)

#plotting this in a bar graph
yearwise_total_suicide.plot(kind="line", figsize = (9,5), color = 'r')
plt.title('Yearwise Total Suicides')
plt.xlabel('Year')
plt.ylabel('Number of suicides');
```

↔ Total number of suicides in each year: Year

2001	451297.0
2002	459999.0
2003	463330.0
2004	478094.0
2005	472285.0
2006	490968.0
2007	514446.0
2008	525606.0
2009	536000.0
2010	569574.0
2011	565404.0
2012	573970.0

Name: Total, dtype: float64



```
statewise_total_suicide = df.groupby("State")["Total"].sum()
print('Total number of suicides in each State : ', statewise_total_suicide)
```

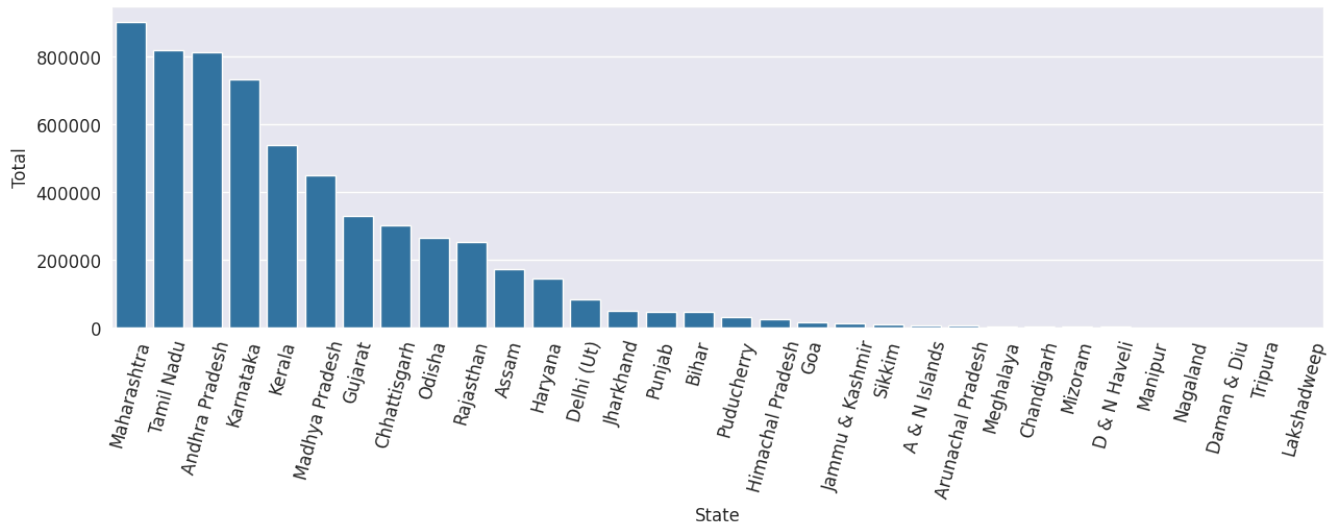
```
#plotting the above data
filter_state = pd.DataFrame(statewise_total_suicide.reset_index())
sort_state = filter_state.sort_values('Total',ascending = False)
f, ax = plt.subplots(figsize=(15, 4))
sns.barplot(y = 'Total', x = 'State',data = sort_state, edgecolor = 'w')
plt.xticks(rotation=75);
plt.show()
```



Total number of suicides in each State : State

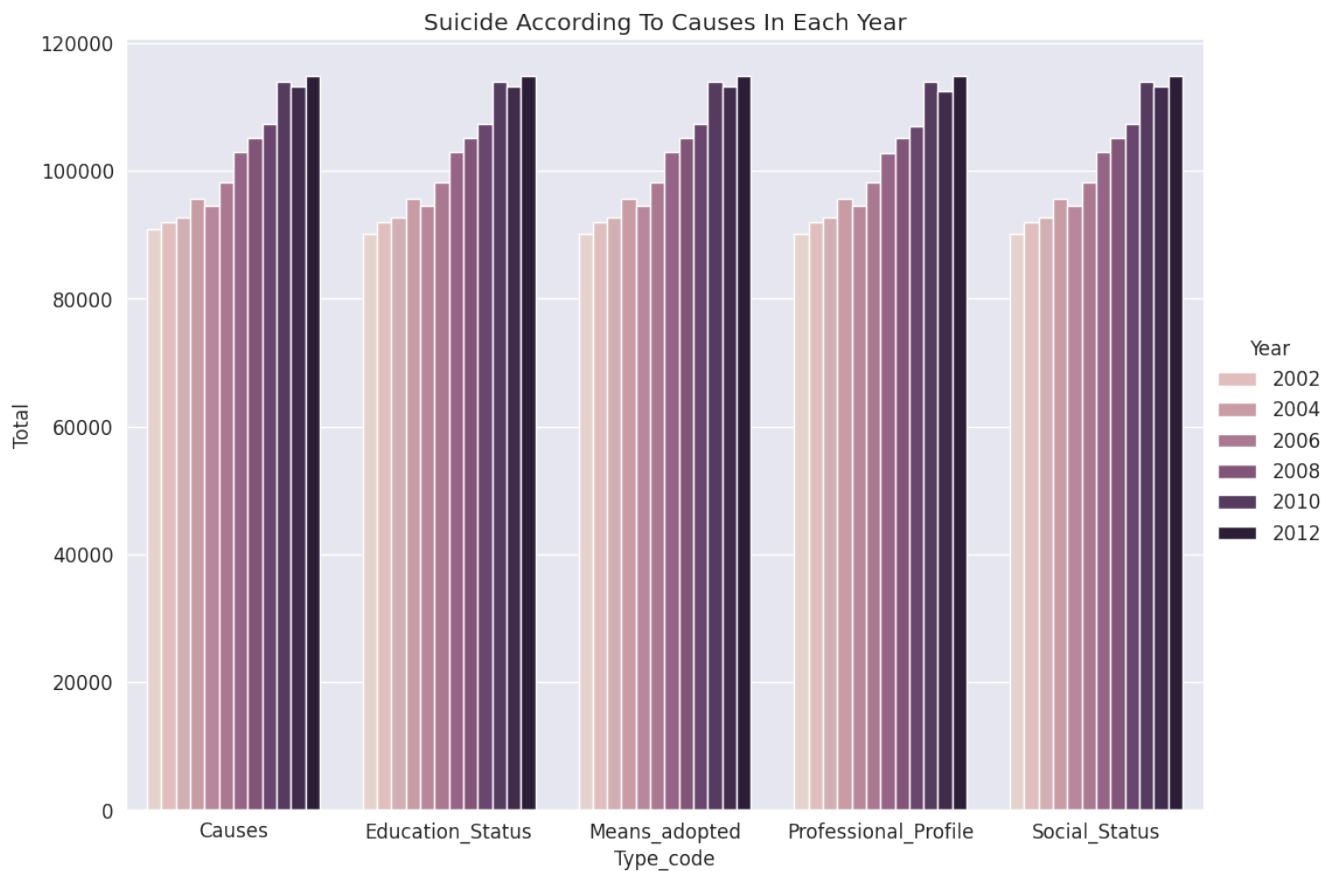
A & N Islands	8109.0
Andhra Pradesh	814059.0
Arunachal Pradesh	6633.0
Assam	172276.0
Bihar	46214.0
Chandigarh	5164.0
Chhattisgarh	302354.0
D & N Haveli	3430.0
Daman & Diu	1391.0
Delhi (Ut)	84272.0
Goa	17363.0
Gujarat	330858.0
Haryana	147176.0
Himachal Pradesh	26562.0
Jammu & Kashmir	14821.0
Jharkhand	49720.0
Karnataka	734825.0
Kerala	538946.0
Lakshadweep	50.0
Madhya Pradesh	451535.0
Maharashtra	901945.0
Manipur	2102.0
Meghalaya	5415.0
Mizoram	4154.0
Nagaland	1728.0
Odisha	267234.0
Puducherry	32144.0
Punjab	46350.0
Rajasthan	255134.0
Sikkim	9606.0
Tamil Nadu	818691.0
Tripura	712.0

Name: Total, dtype: float64



```
#Filtering the data according to 'type_code','Total' and 'Year'
filter_type_code = pd.DataFrame(df.groupby(["Type_code","Year"])["Total"].sum()).reset_index()

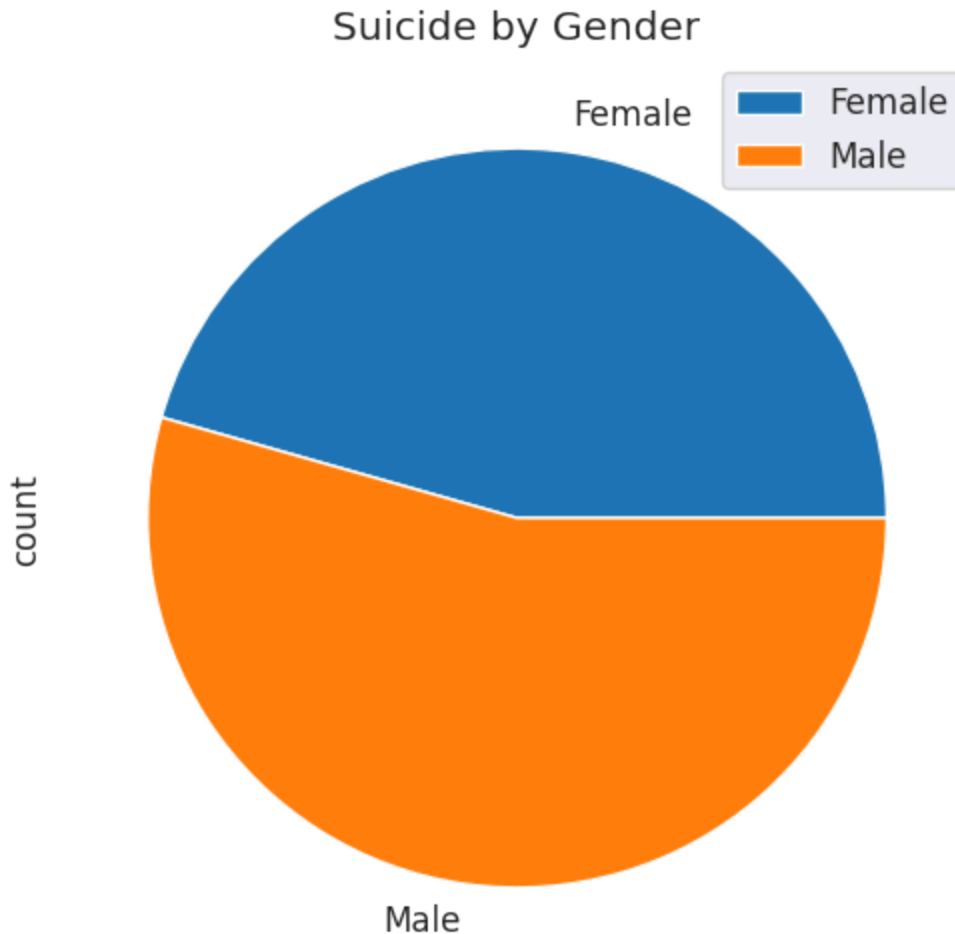
#Plotting the above data
sns.catplot(x="Type_code", y="Total",hue="Year", kind="bar", data=filter_type_code,height=7
plt.title('Suicide According To Causes In Each Year');
```




```
counts = df['Gender'].value_counts().sort_index()
print(counts)

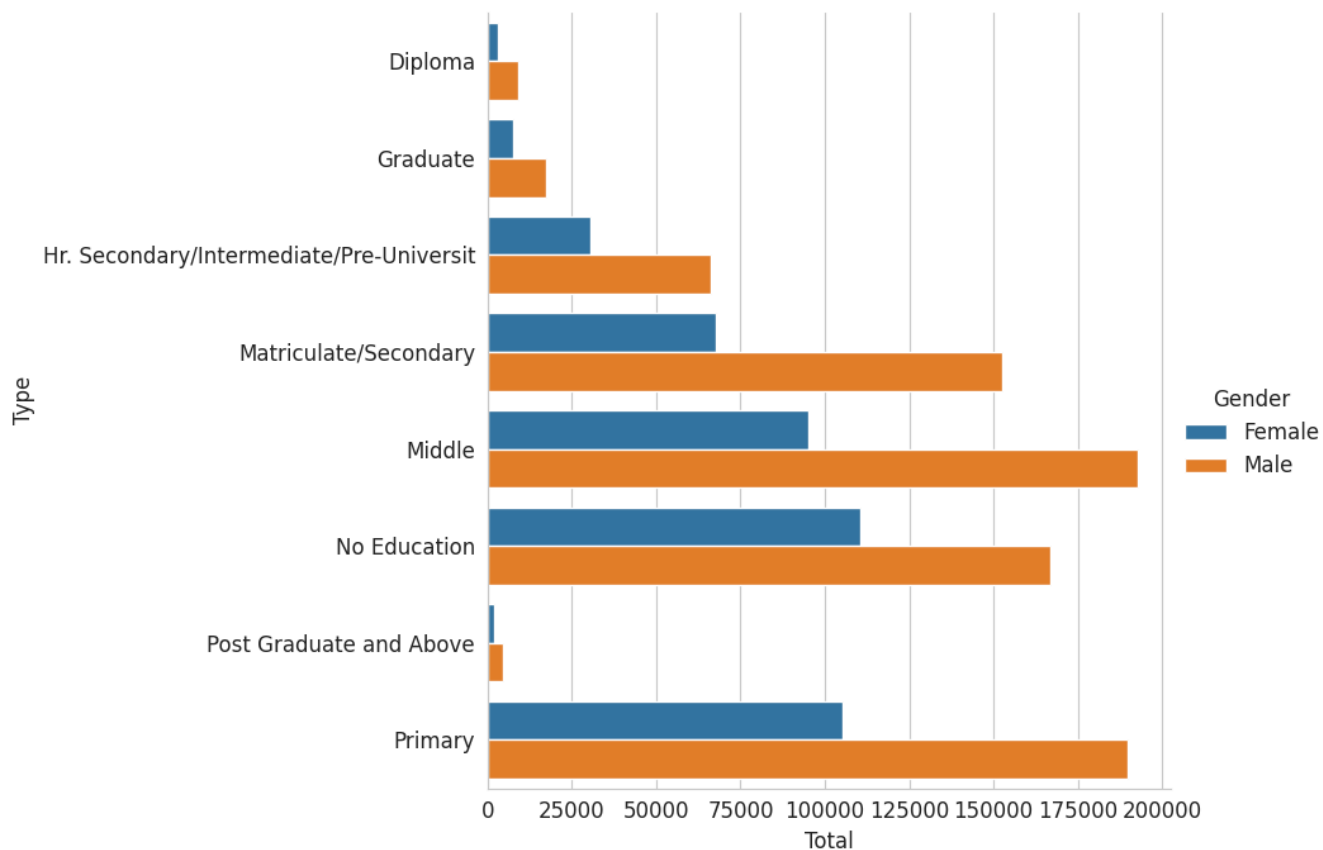
# Plot a pie chart
counts.plot(kind='pie', title='Suicide by Gender',figsize=(6,6))
plt.legend()
plt.show()
```

```
Gender
Female    39562
Male      47174
Name: count, dtype: int64
```



```
education_status = pd.DataFrame(df[df["Type_code"]=="Education_Status"].groupby(["Type","Ge
education_status
```

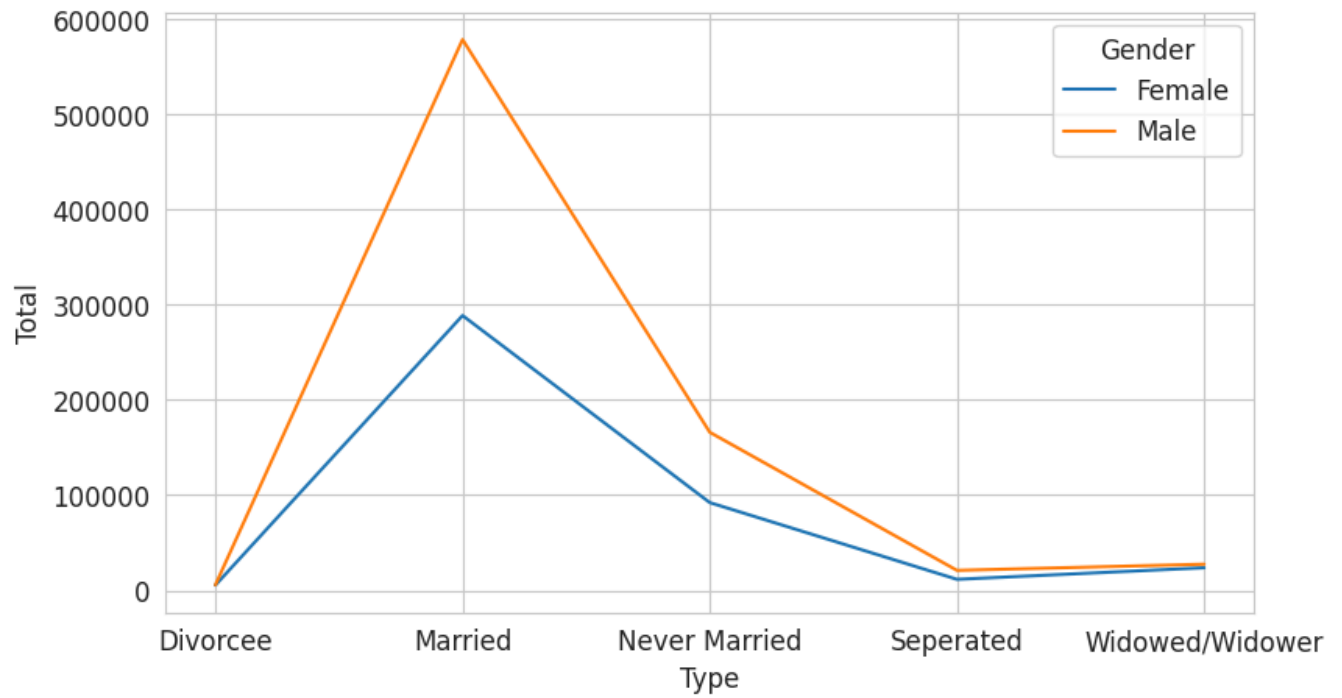
```
#plotting the above data
g = sns.catplot(y="Type", x="Total",hue="Gender", kind="bar", data=education_status,height=
sns.set_style('whitegrid');
```



```
filter_social_status = pd.DataFrame(df[df["Type_code"]=="Social_Status"].groupby(["Type", "G
```

```
#plotting the above data
```

```
sns.lineplot(x="Type", y="Total", hue="Gender", data=filter_social_status);
```



```
statewise_Age_df = df.drop(df[df.Age_group == '0-100+'].index)
```

```
counts = statewise_Age_df['Age_group'].value_counts().sort_index()
print(counts)
```

```
counts.plot(kind='bar', title='Age Distribution',figsize=(7,5))
plt.xticks(rotation=0)
plt.show()
```



Age_group

0-14 8142

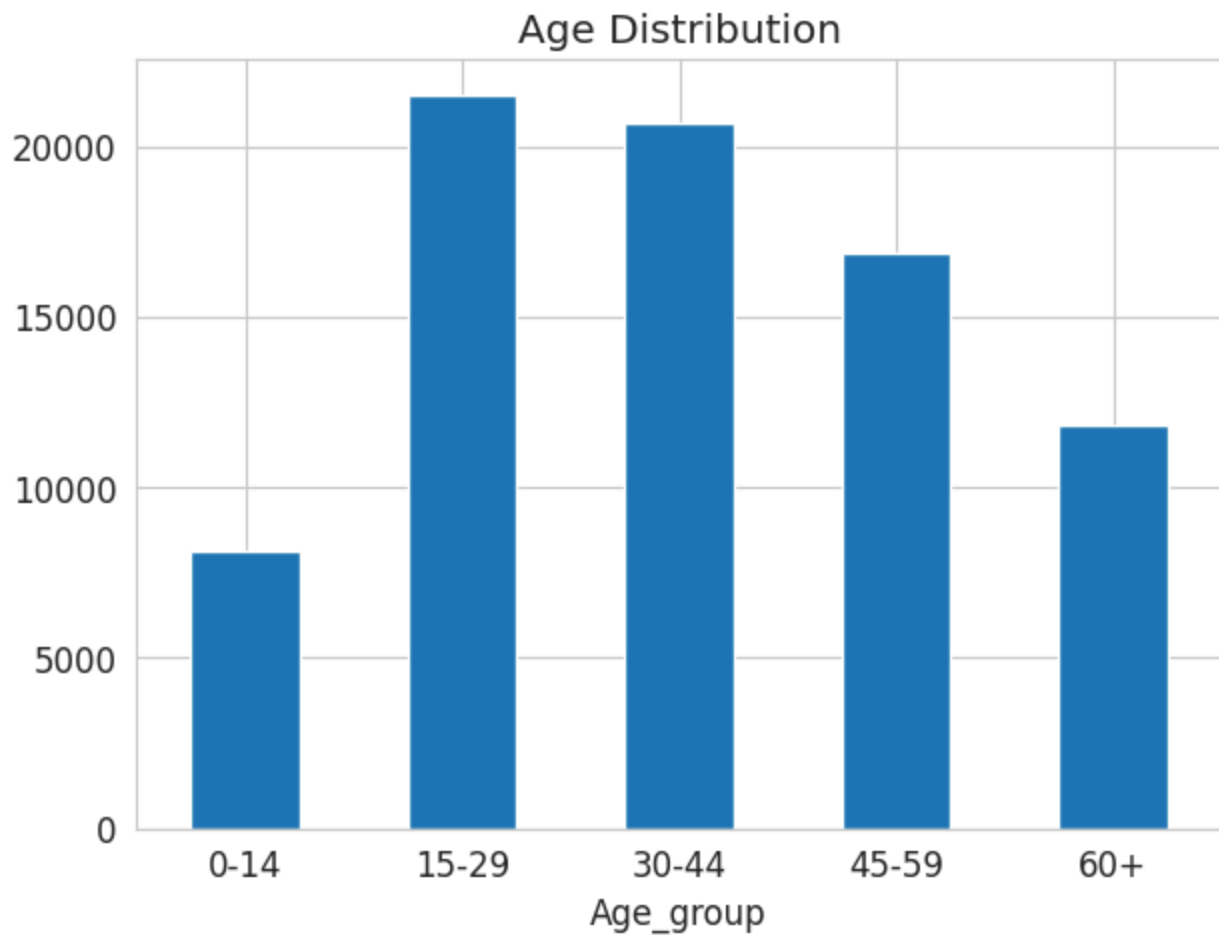
15-29 21494

30-44 20659

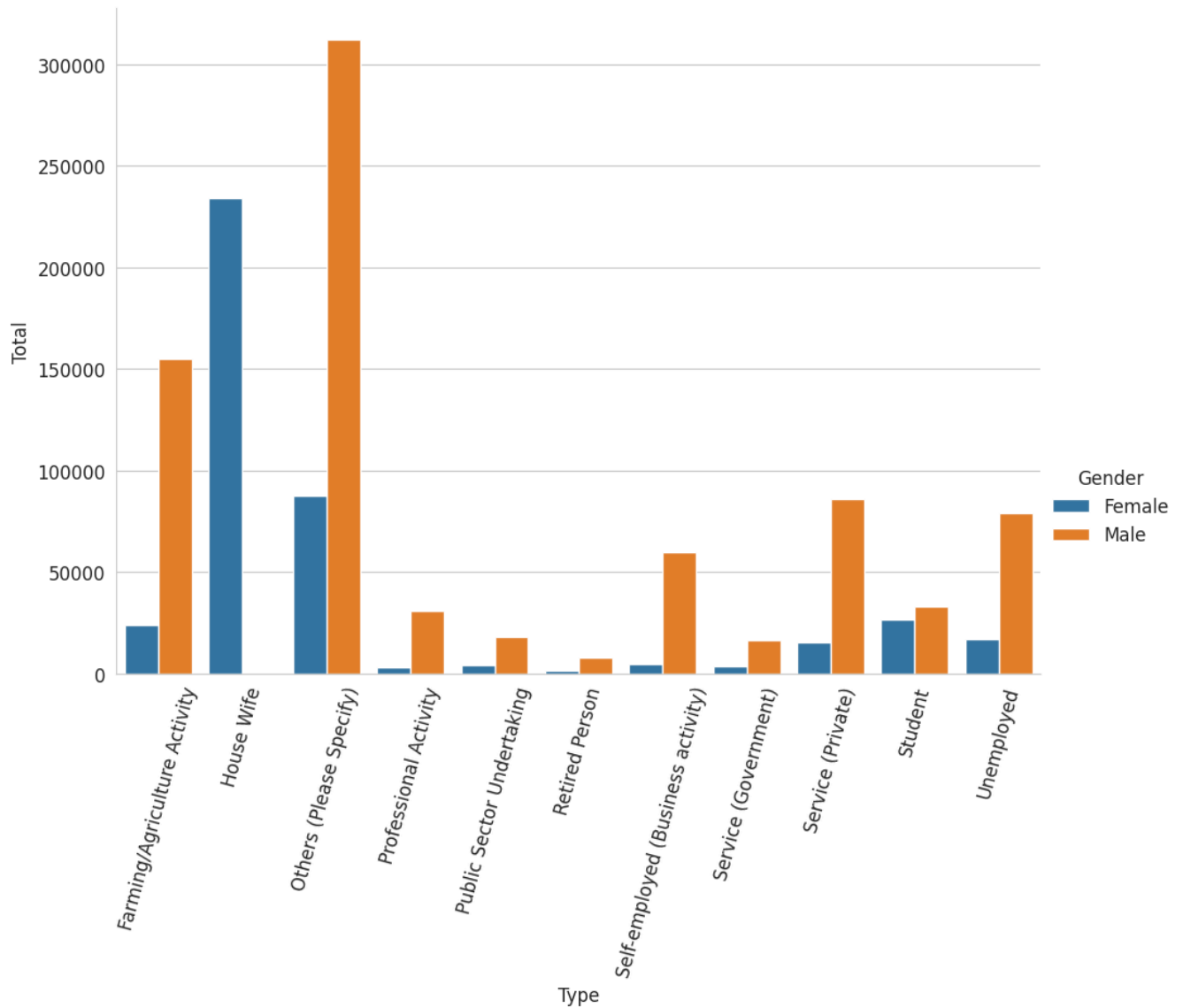
45-59 16842

60+ 11803

Name: count, dtype: int64



```
filter_social_status = pd.DataFrame(df[df["Type_code"]=="Professional_Profile"].groupby(["T  
g = sns.catplot(x="Type", y="Total",hue="Gender", kind="bar", data=filter_social_status,hei  
g.set_xticklabels(rotation=75);
```



```
grouped_year = df.groupby(["Year", "Gender"])["Total"].sum()
grouped_year = pd.DataFrame(grouped_year).reset_index()
# grouped_year
sns.lmplot(x="Year", y="Total", hue="Gender", data=grouped_year, height=7, aspect=11.7/8.27)
```

