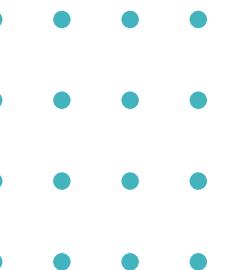




NIGAR AGAHYeva

Analysis of Suicides In India

2001-2012



Get Started





INFOGRAPHIC

About Our Data

In these dataset we have suicides' age causes, place, year, education degree, gender, and age group professional and social statuses, means adopted.



Causes, professional profile, social status, and means adopted were given in the same column as a “code type”.



Additionally there types given in the next column which was called type.

Means_adopted	By Over Alcoholism
Means_adopted	By Consuming Other Poison
Means_adopted	By Jumping from (Other sites)
Professional_Profile	Professional Activity
Professional_Profile	Service (Private)
Professional_Profile	House Wife
Social_Status	Married
Social_Status	Widowed/Widower
Social_Status	Divorcee
Social_Status	Separated
Causes	Drug Abuse/Addiction
Causes	Professional/Career Problem
Causes	Cancer

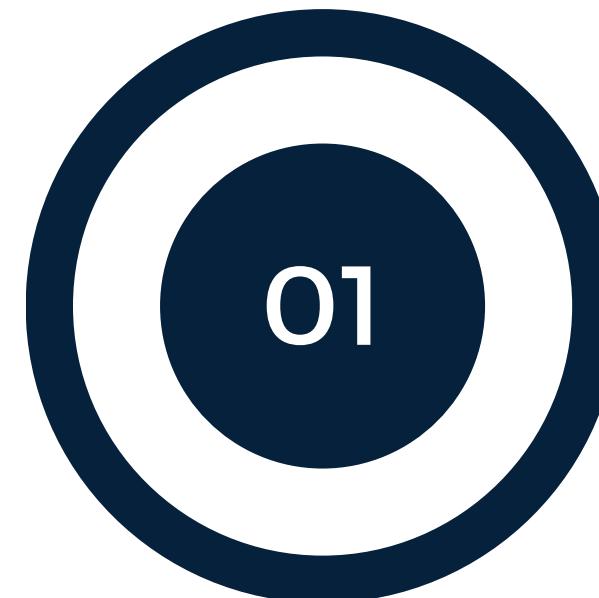
Next Page





INFOGRAPHIC

Firstly, we start with Python in order to make data cleaner and do some analysis. Here is steps:



Data collection



Quick EDA



Analysis and visualization Findings From The Analysis



Next Page 



Data Collection

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

```
data=pd.read_csv('Suicides in India 2001-2012.csv')
```

Quick EDA



INFOGRAPHIC

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 237519 entries, 0 to 237518
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype  
---  --          -----          --    
0   State        237519 non-null   object 
1   Year         237519 non-null   int64  
2   Type_code    237519 non-null   object 
3   Type         237519 non-null   object 
4   Gender       237519 non-null   object 
5   Age_group    237519 non-null   object 
6   Total         237519 non-null   int64  
dtypes: int64(2), object(5)
memory usage: 12.7+ MB
```

```
data.duplicated().sum()
```

```
0
```

```
data.size
```

```
1662633
```

```
data.shape
```

```
(237519, 7)
```

```
data.isnull().sum()
```

```
State      0
Year       0
Type_code  0
Type       0
Gender     0
Age_group  0
Total      0
dtype: int64
```

```
▶ data['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',
       'Uttar Pradesh', 'Uttarakhand', 'West Bengal'], dtype=object)
```

We can drop 'Total (All India)', 'Total (States)', 'Total (Uts)' columns

```
▶ # Define a list of values to remove
```

```
values_to_remove = ['Total (All India)', 'Total (States)', 'Total (Uts)']
```

```
# Filter rows where 'Date' column is not in the list of values to remove
data = data[~data['State'].isin(values_to_remove)]
```

```
[ ] data['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',
       'Tripura', 'Uttar Pradesh', 'Uttarakhand', 'West Bengal'],
       dtype=object)
```

Next Page



Quick EDA

```
data['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',
```

We can see that some values are repeated like 'Bankruptcy or Sudden change in Economic', 'Bankruptcy or Sudden change in Economic Status', and 'Unemployment', 'Unemployed' we need to rename them under the same name

```
[ ] data['Type'] = data['Type'].replace(['Bankruptcy or Sudden change in Economic'], 'Bankruptcy or Sudden change in Economic Status')
[ ] data['Type'] = data['Type'].replace(['Unemployed'], 'Unemployment')
```

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

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

```
▶ data['Age_group'].unique()

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

We are going to make two DataFrames one contains 0-100 age group used to analysis total suicide count and another contains the without 0-100 age group to remove ambiguity in dataset while calculating age related analysis.

```
[ ] data_ab = data[data['Age_group'] != '0-100+']
```

Quick EDA

We can remove 'Causes Not known', 'Other Causes (Please Speciy)', 'By Other means (please specify)', 'By Other means' rows in order to make our analys clearer

```
[ ] # Define a list of values to remove
values_to_remove = ['Causes Not known', 'Other Causes (Please Speciy)', 'By Other means (please specify)', 'By Other means', 'Others (Please Specify)']

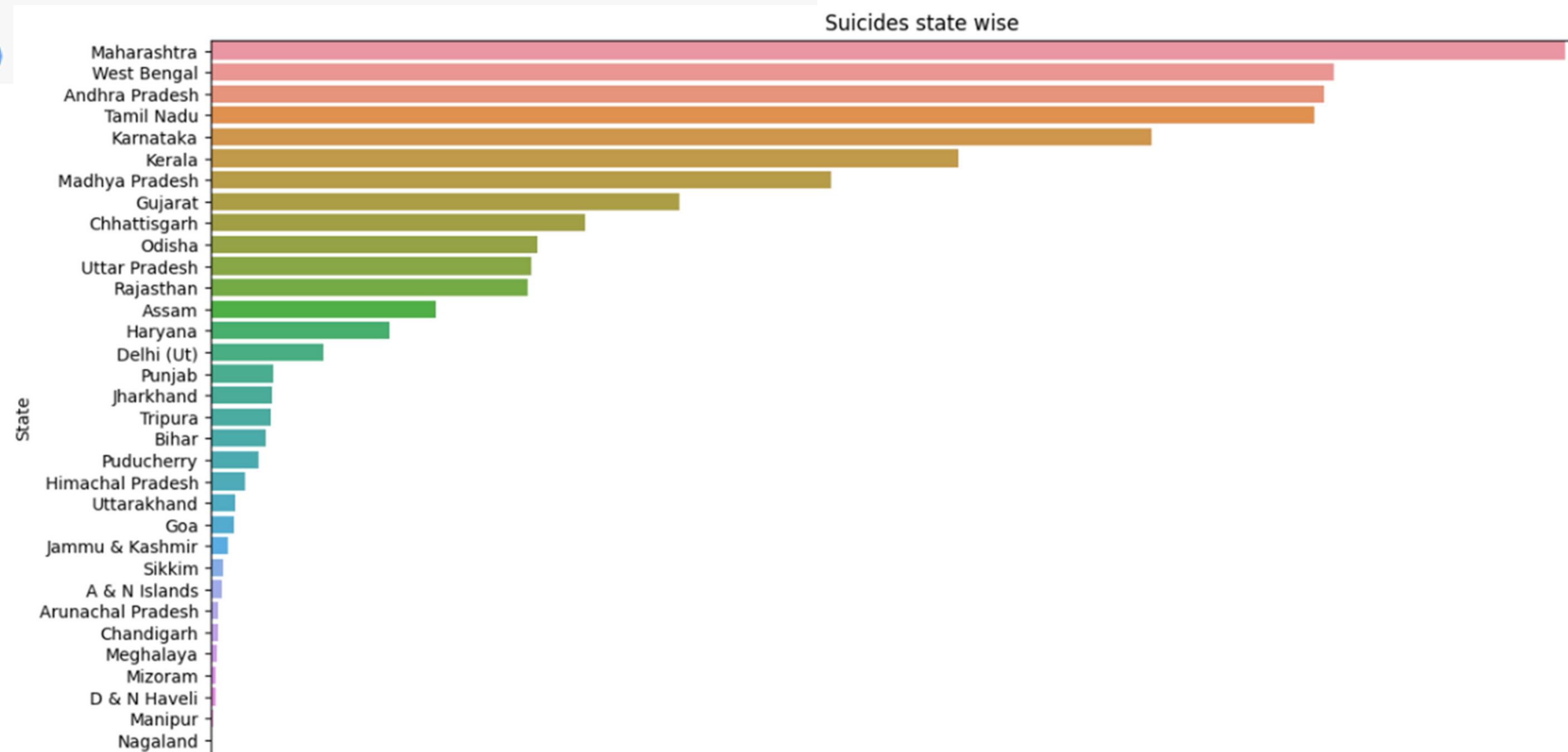
# Filter rows where 'Date' column is not in the list of values to remove
data = data[~data['Type'].isin(values_to_remove)]
```

```
data['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',
       'Unemployment', '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 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', 'Farming/Agriculture Activity', 'Service (Government)',
       'Married', 'Seperated', 'Widowed/Widower', 'Divorcee',
       'Never Married', 'Bankruptcy or Sudden change in Economic Status',
```

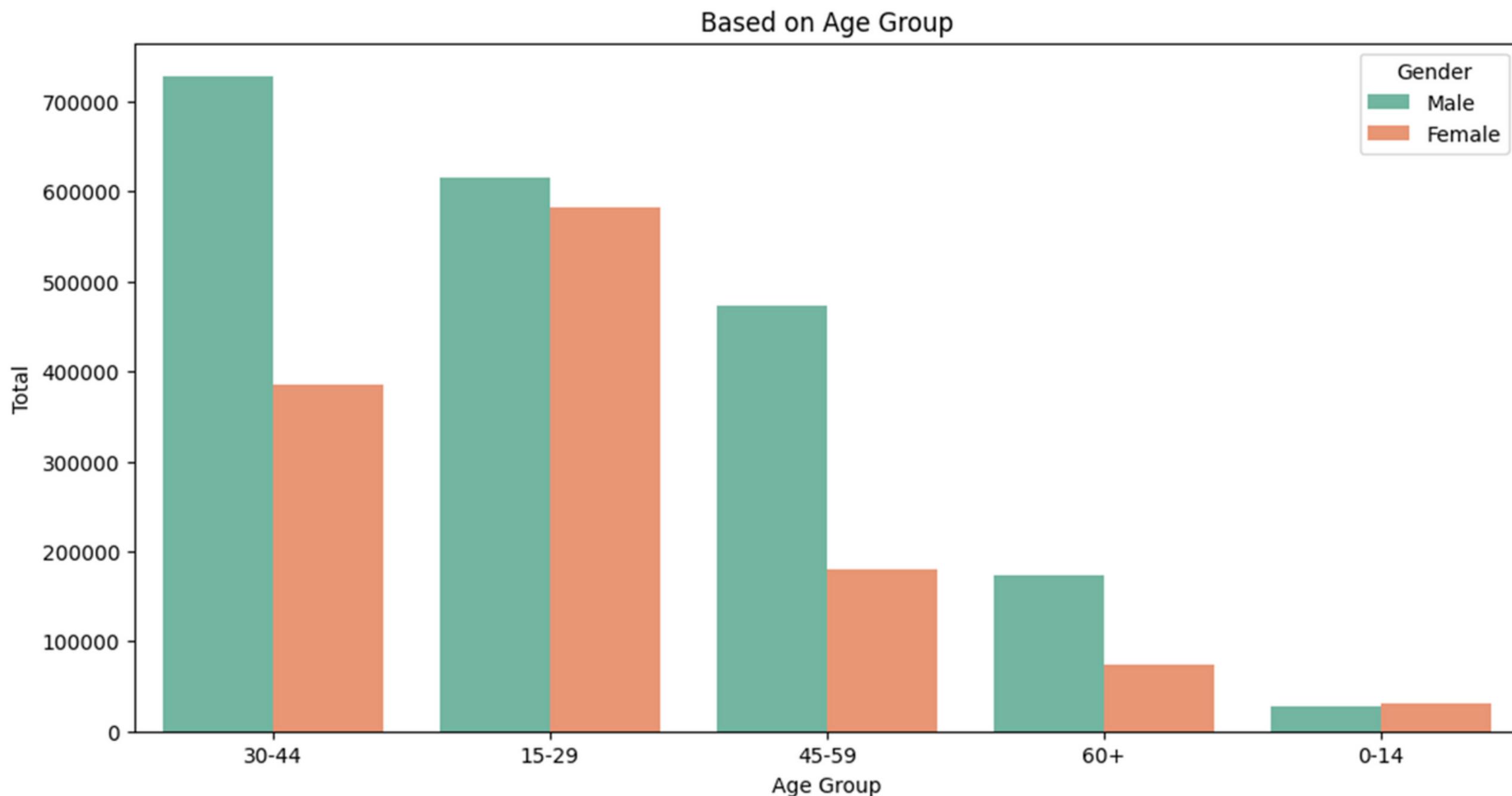
Analysis and visualization

```
# Grouping data based on State counting the total suicides  
data_st = data.groupby(by=[ 'State']).sum().Total.sort_values(ascending = False)  
  
# Bargraph for State and Total count  
plt.figure(figsize=(15,8))  
sns.barplot(x=data_st,y=data_st.index)  
plt.xlabel('Total suicides')  
plt.title('Suicides state wise')
```



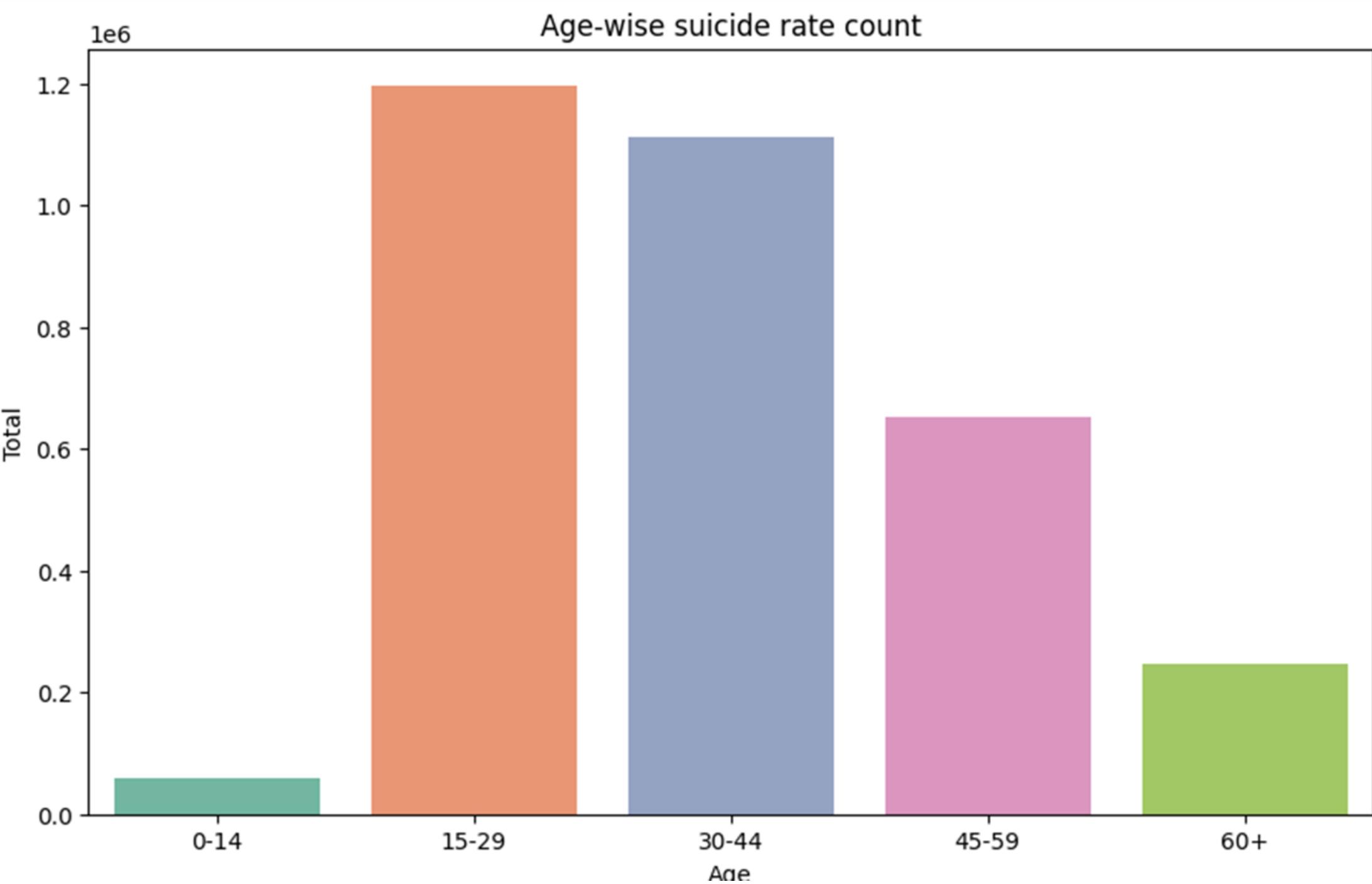
Analysis and visualization

```
df_ab = data_ab[['Age_group','Gender','Total']]  
  
# Grouping Data Based on the Age_group and Gender and Count total Suicides  
edSort = df_ab.groupby(['Age_group','Gender'],as_index=False).sum().sort_values('Total',ascending=False)  
  
#BarGraph for Age_group, Gender and Total count  
plt.figure(figsize=(12,6))  
sns.barplot(x='Age_group',y='Total',hue='Gender',data=edSort, palette=sns.color_palette('Set2'))  
plt.xlabel('Age Group')  
plt.title('Based on Age Group')
```



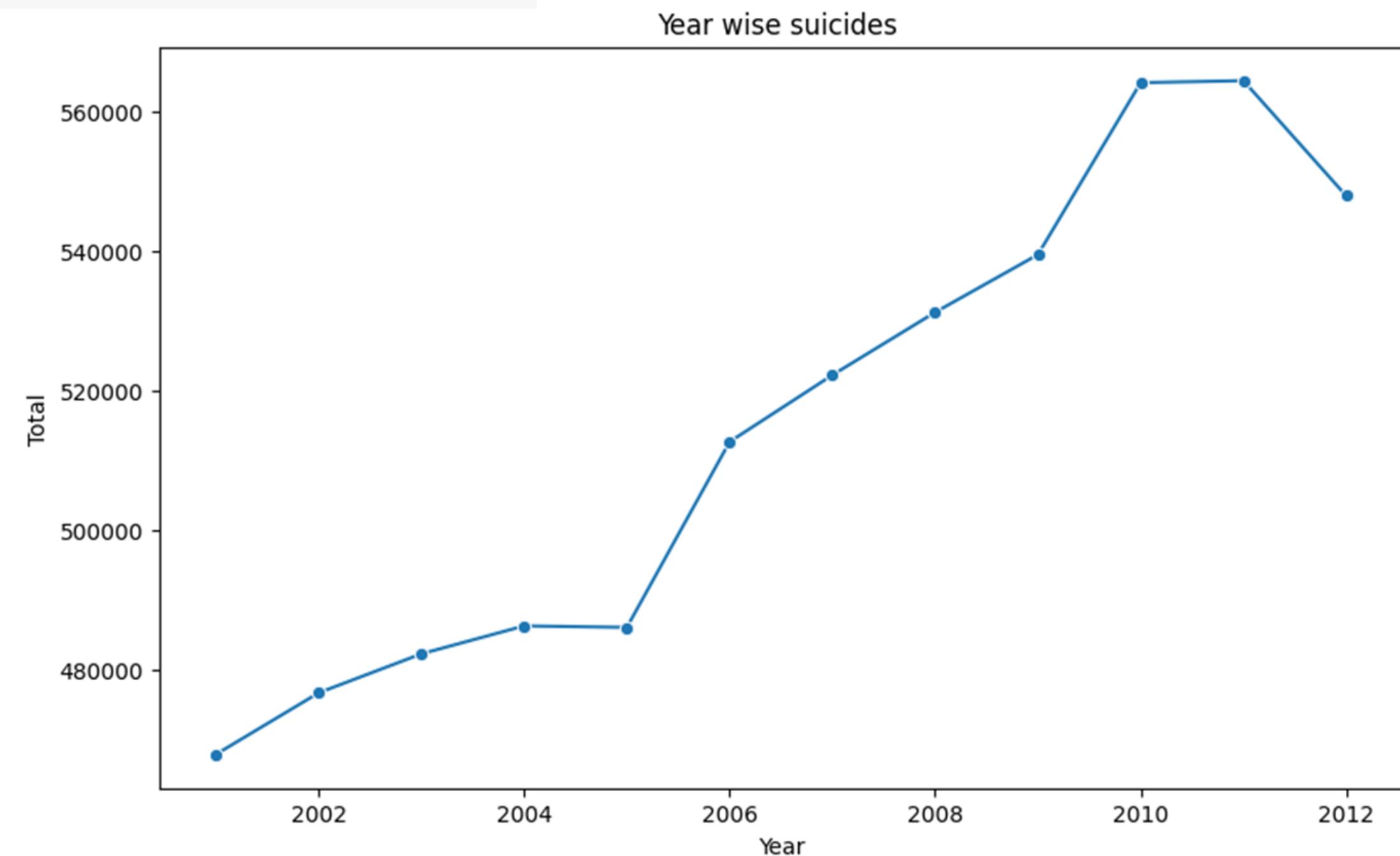
Analysis and visualization

```
# Grouping Data Based on Age_group  
df_age = df_ab.groupby(by = ['Age_group'])  
  
# BarGraph  
plt.figure(figsize=(10,6))  
sns.barplot(x=df_age.sum().Total.index,y=df_age.sum().Total.values, palette=sns.color_palette('Set2'))  
plt.xlabel('Age')  
plt.ylabel('Total')  
plt.title('Age-wise suicide rate count')
```



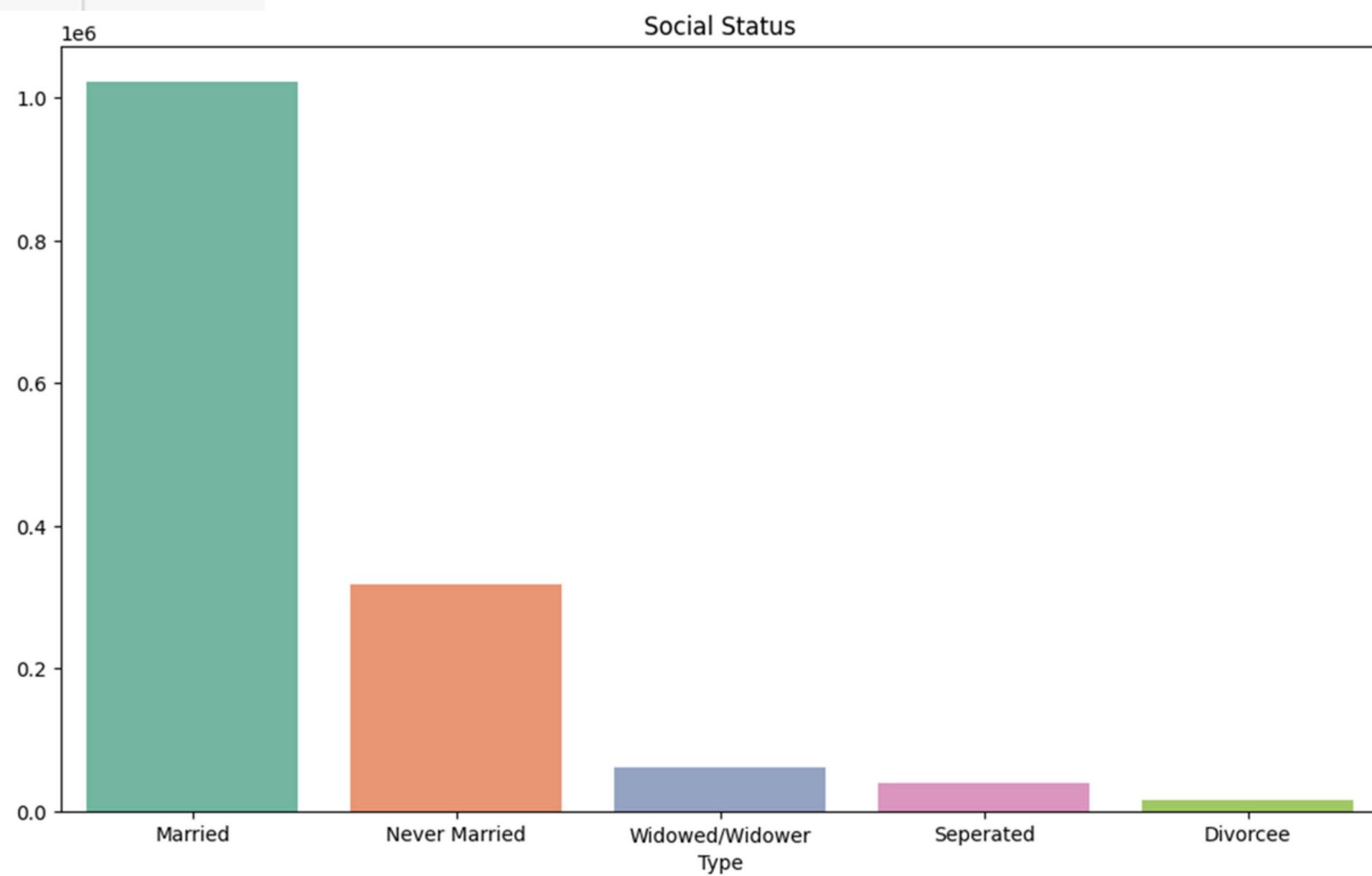
Analysis and visualization

```
# Grouping Data Based on Year  
df_year = data.groupby(by = ['Year'])  
  
# Graph  
plt.figure(figsize = (10,6))  
sns.lineplot(x=df_year.sum().index, y = df_year.sum().Total.values,marker='o')  
plt.ylabel('Total')  
plt.title('Year wise suicides ')
```



Analysis and visualization

```
# Filter data by Social_Status  
df_social_st = data[data['Type_code']=='Social_Status']  
  
# Grouping Data based on Social Status  
df_socgrp = df_social_st.groupby(by=[ 'Type'])  
df_sgtotal = df_socgrp.sum().Total.sort_values(ascending = False)  
  
# Bargraph based on Social Status  
plt.figure(figsize=(12,7))  
sns.barplot(x=df_sgtotal.index, y=df_sgtotal.values, palette=sns.color_palette('Set2'))  
plt.title('Social Status')
```

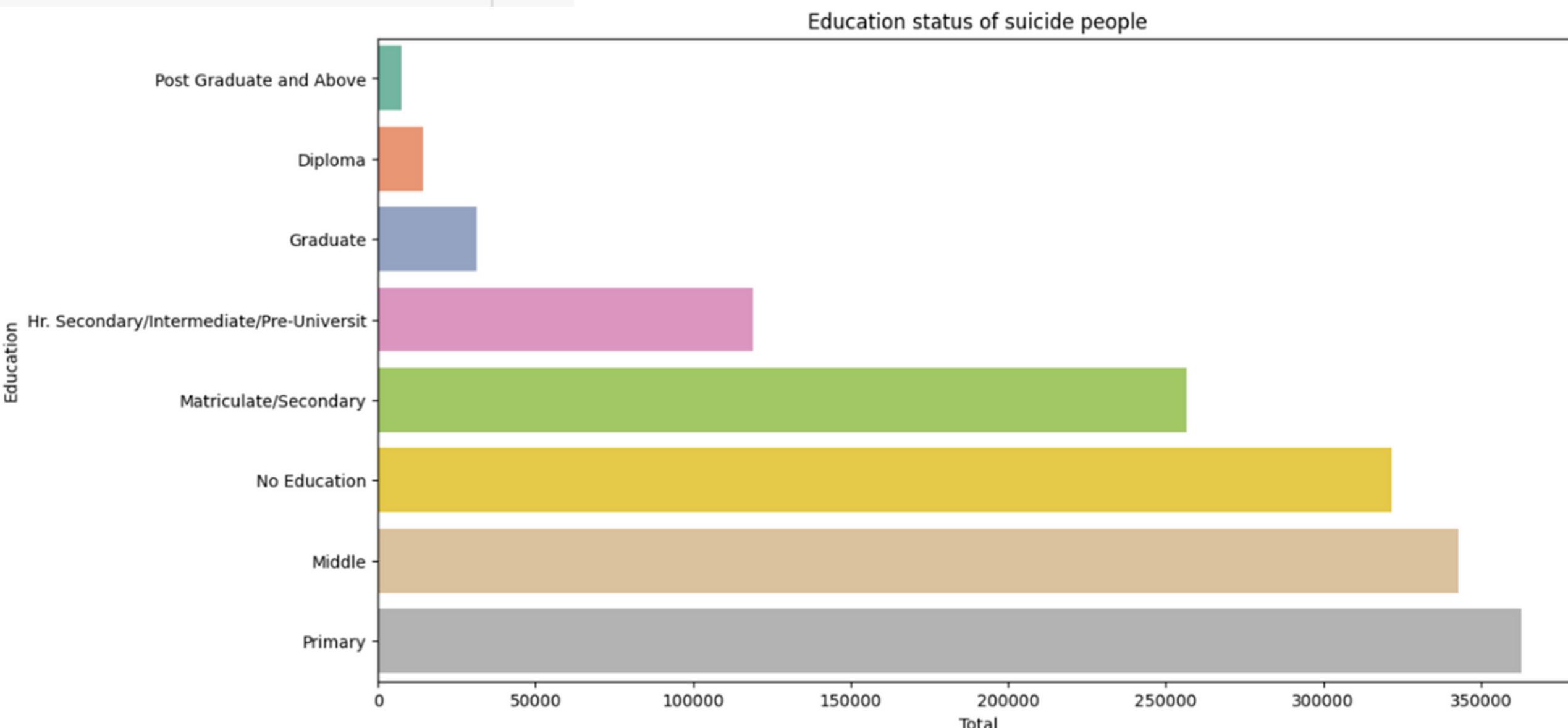


Analysis and visualization

```
# Filter data based on type code
df_edu_st = data[data['Type_code']=='Education_Status']

# Grouping Data
df_edugrp = df_edu_st.groupby(by = ['Type']).sum().Total.sort_values(ascending=True)

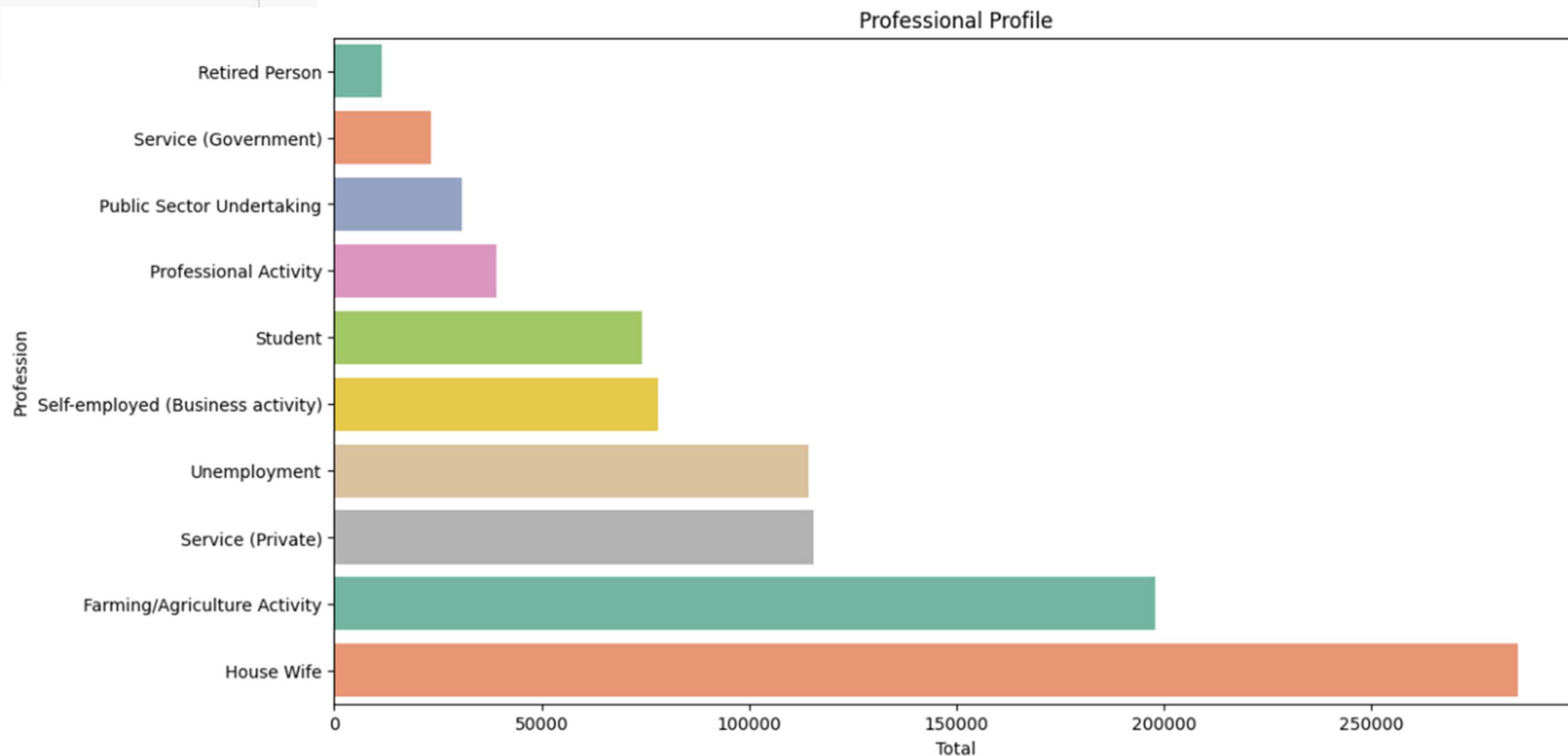
# Bargraph based on Education Status and total suicides
plt.figure(figsize=(13,7))
sns.barplot(y=df_edugrp.index,x=df_edugrp.values, palette=sns.color_palette('Set2'))
plt.ylabel('Education')
plt.title('Education status of suicide people')
plt.xlabel('Total')
```



Analysis and visualization

```
# Filter the data based on type code  
df_profile = data[data['Type_code']=='Professional_Profile']  
df_profile= df_profile[df_profile['Type']!='Others (Please Specify)']
```

```
# Grouping Data of different professional profile  
df_progrp = df_profile.groupby(by = ['Type']).sum().Total.sort_values(ascending=True)  
  
# Bargraph for professional profile and total count  
plt.figure(figsize=(13,7))  
sns.barplot(y=df_progrp.index,x=df_progrp.values, palette=sns.color_palette('Set2'))  
plt.title('Professional Profile')  
plt.xlabel('Total')  
plt.ylabel('Profession')
```

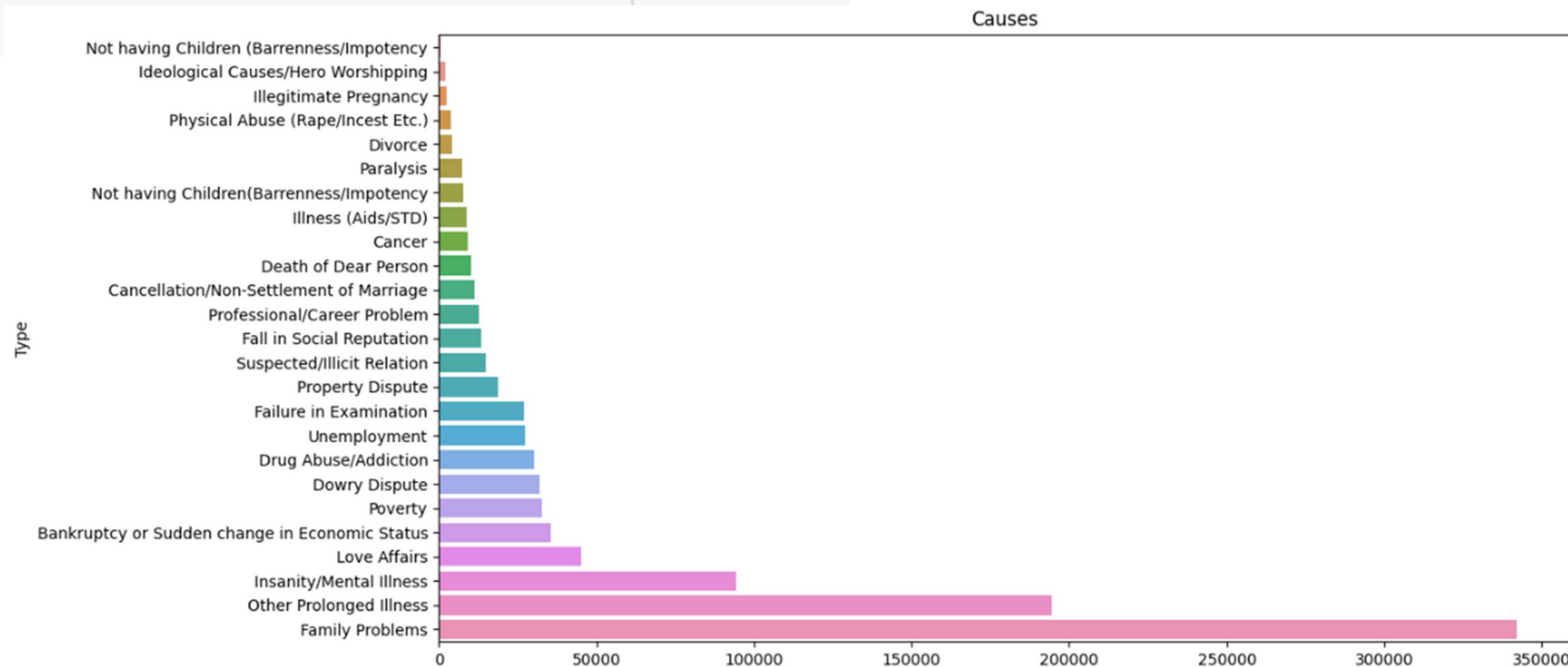


Analysis and visualization

```
# Filter data based on causes
filtered = np.where((data['Type_code'] == 'Causes'))

# Group the data
df_casgrp = data.iloc[filtered].groupby(by = ['Type']).sum().Total.sort_values(ascending=True)

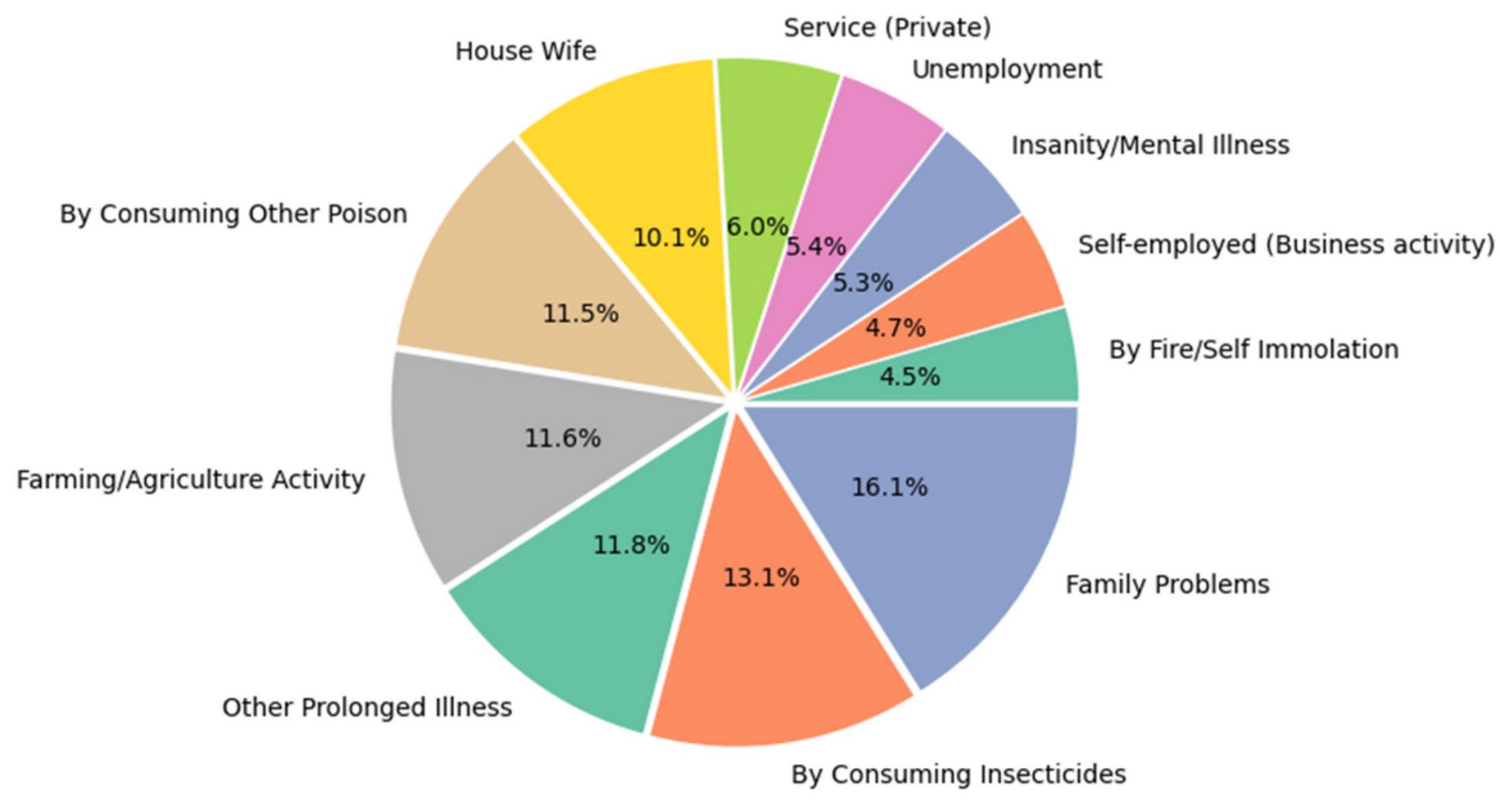
# graph for different causes and total count
plt.figure(figsize=(13,7))
sns.barplot(y=df_casgrp.index,x=df_casgrp.values)
plt.title('Causes')
```



Analysis and visualization

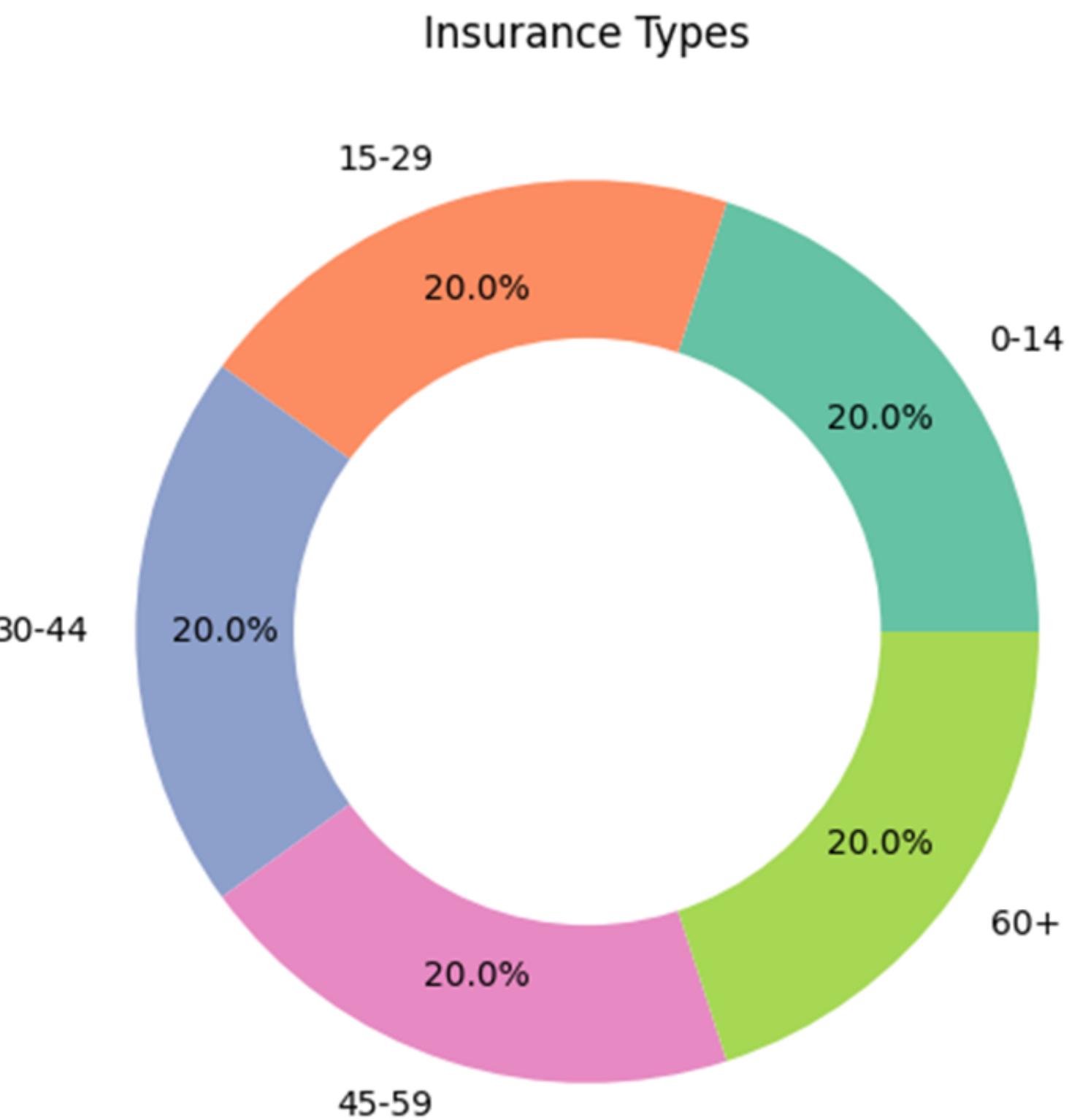
```
# filter Data based on age 45-59
filtered = np.where(data['Age_group'] == '45-59')
df_45 = data.iloc[filtered]

# removing unnecessary information
filtered = np.where(df_45['Type'] != 'Others (Please Specify)')
df_45 = df_45.iloc[filtered]
df_45grp = df_45.groupby(by = ['Type']).sum().Total.sort_values(ascending=False)
# pie chart of age group 45-59 and their reasons
pie, ax = plt.subplots(figsize=[10,6])
df = df_45grp.values[11:-1]
keys = df_45grp.index[11:-1]
palette_color = sns.color_palette('dark')
plt.pie(x=df, autopct="%1f%%", explode=[0.03]*11, labels=keys, pctdistance=0.5, colors=sns.color_palette('Set2'))
plt.xlabel('45-59 age group their profession and reasons')
```



Analysis and visualization

```
plt.figure(figsize=(8,6))
count_of_each_age = df_ab.groupby('Age_group').size()
plt.pie(count_of_each_age, labels=['0-14', '15-29', '30-44', '45-59', '60+'], autopct='%1.1f%%', colors=sns.color_palette('Set2'),
# move the percentage inside the arcs
pctdistance=0.80)
### Add a hole in the pie
hole = plt.Circle((0, 0), 0.65, facecolor='white')
plt.gcf().gca().add_artist(hole)
plt.title('Insurance Types')
```



Findings From The Analysis

Maharashtra have the high suicide count.

Suicide in male are more than female in every age group.

Married committed more suicide.

From professional profile, house wife commit more suicide.

Agriculture sector have the high number of suicides in professional profile after house wife.

From the Analysis, higher educational status have low suicide count.

Family problems is the common and major causes for the high suicide count in different age groups.

Power BI

We do similar steps in Power BI

The screenshot shows three separate filter panes from a Power BI report titled "Suicides in India 2001-2012".

- Left Filter Pane (State):** Contains options for sorting (Sort Ascending, Sort Descending), clearing filters (Clear Filter), and removing empty items (Remove Empty). It also includes a "Text Filters" section with a search bar and a list of states and totals:
 - Meghalaya
 - Mizoram
 - Nagaland
 - Odisha
 - Puducherry
 - Punjab
 - Rajasthan
 - Sikkim
 - Tamil Nadu
 - Total (All India)
 - Total (States)
 - Total (Uts)
 - Tripura
 - Uttar Pradesh
 - Uttarakhand
 - West Bengal
- Middle Filter Pane (Cause Type):** Contains options for sorting (Sort Ascending, Sort Descending), clearing filters (Clear Filter), and removing empty items (Remove Empty). It also includes a "Text Filters" section with a search bar and a list of causes:
 - By Jumping from Buildings
 - By Jumping off Moving Vehicles/Trains
 - By Machine
 - By Other means
 - By Other means (please specify)
 - By Over Alcoholism
 - By Overdose of sleeping pills
 - By Self Infliction of injury
 - By touching electric wires
 - Cancellation/Non-Settlement of Marriage
 - Cancer
 - Causes Not known
 - Death of Dear Person
 - Diploma
 - Divorce
 - Divorcee
 - Domestic Dispute
- Right Filter Pane (Reason):** Contains options for sorting (Sort Ascending, Sort Descending), clearing filters (Clear Filter), and removing empty items (Remove Empty). It also includes a "Text Filters" section with a search bar and a list of reasons:
 - No Education
 - Not having Children (Barrenness/Impotency)
 - Not having Children(Barrenness/Impotency)
 - Other Causes (Please Specify)
 - Other Prolonged Illness
 - Others (Please Specify)
 - Paralysis
 - Physical Abuse (Rape/Incest Etc.)
 - Post Graduate and Above
 - Poverty
 - Primary
 - Professional Activity
 - Professional/Career Problem
 - Property Dispute
 - Public Sector Undertaking
 - Retired Person
 - Self-employed (Business activity)

Next Page >

Power BI

Replace Values

Replace one value with another in the selected columns.

Value To Find

Replace With

► Advanced options

Replace Values

Replace one value with another in the selected columns.

Value To Find

Replace With

► Advanced options

Replace Values

Replace one value with another in the selected columns.

Value To Find

Replace With

► Advanced options

OK

Cancel

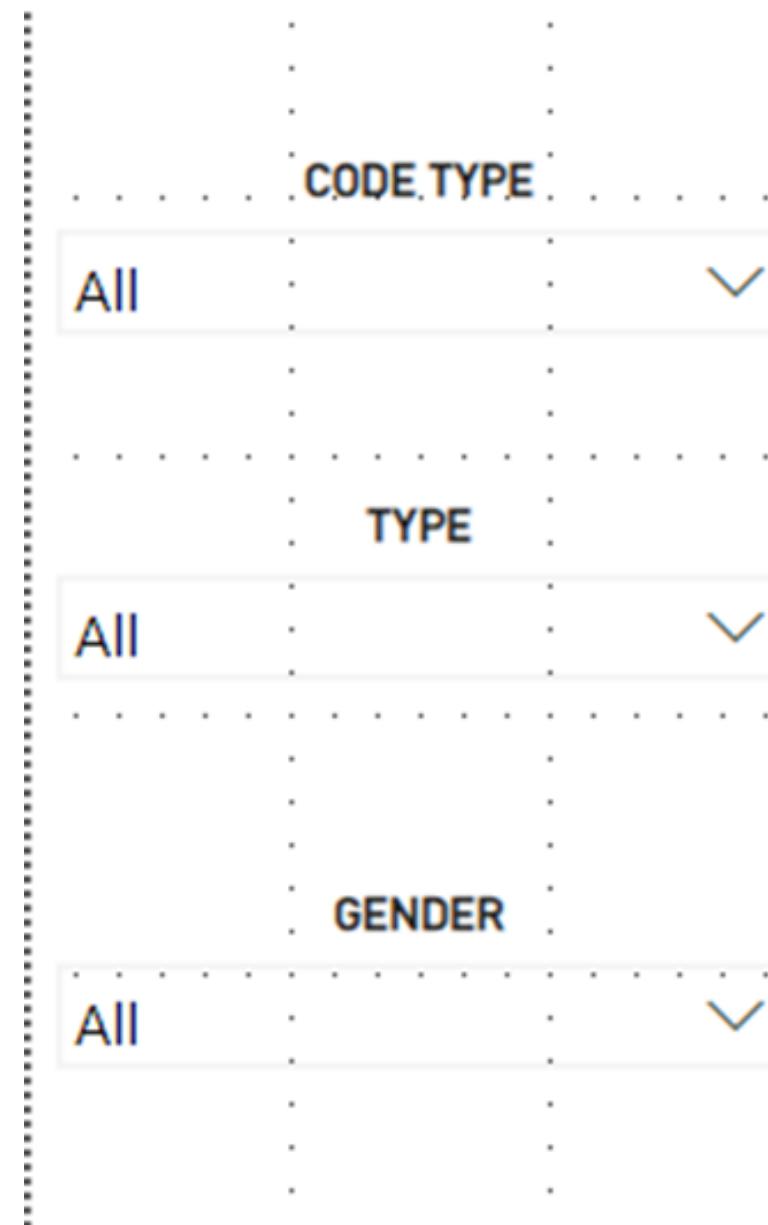
OK

Cancel

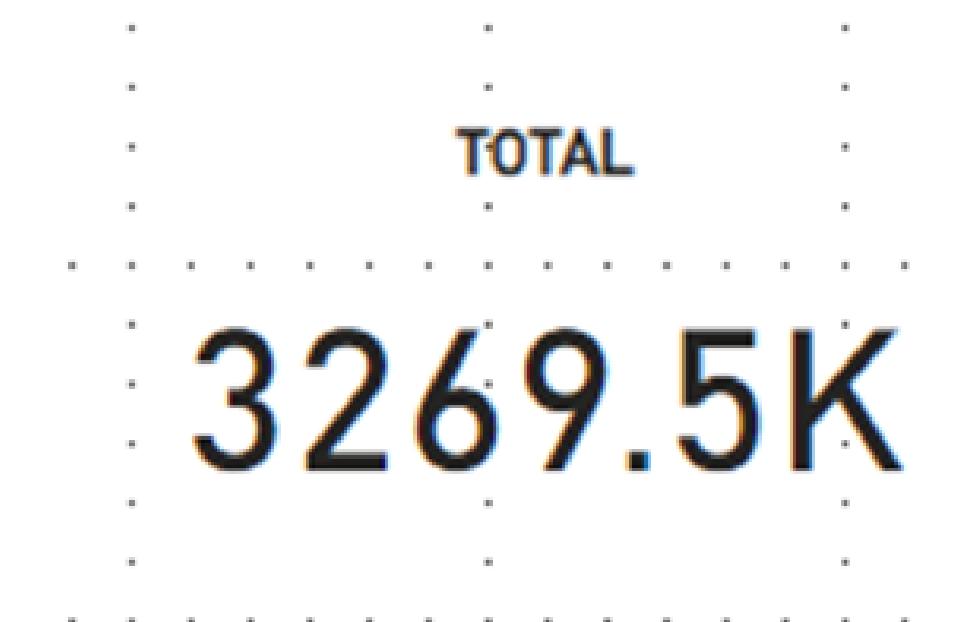
Power BI

Let's start doing visualizations

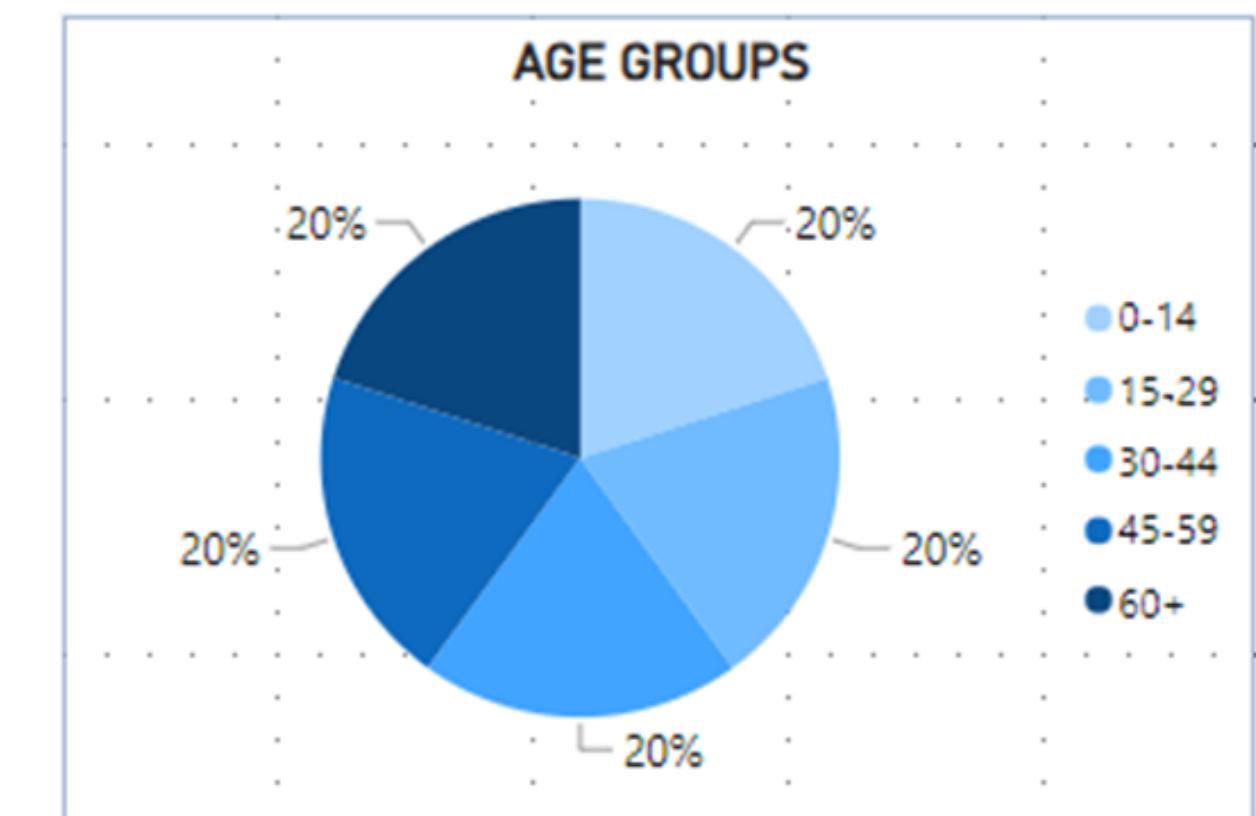
First of all we add **slicers** to select code type, type, and gender



Then put in card to show total number of people

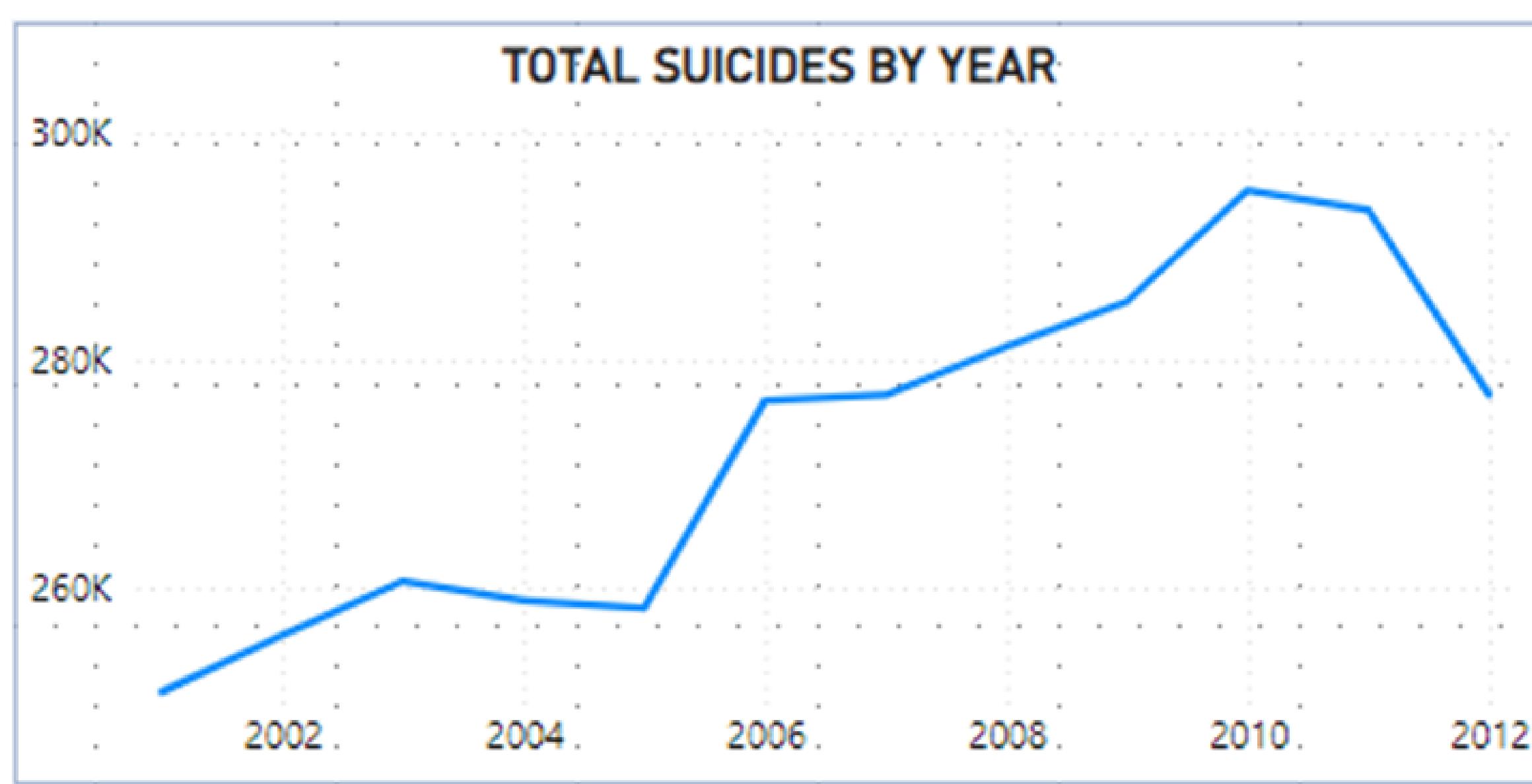


After that, we use pie chart to show each age group part and we can observe that selected number of people are the same for each age group



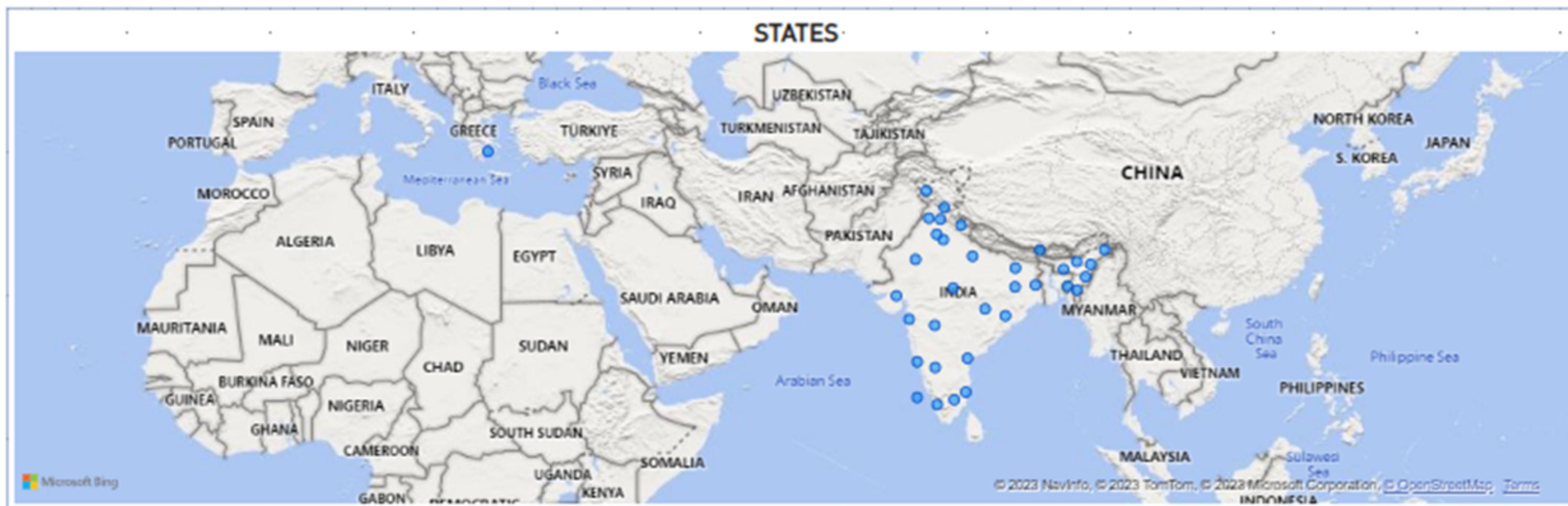
Power BI

We utilize line chart to demonstrate total suicides in each year



Power BI

In the last step we use map to show states



Final view after theme editing

