# Suicide. Why?

Some battles in life are meant to be fought alone. However much you might want, you are unable to use cheat codes like reaching out to family and friends. There has always been a point in everyone's life when they thought nobody around them really understood them, in spite of several assurances from loved ones alluding otherwise. We feel englufed in loneliness, self-pity and helplessness, while being surrounded by 'our' people. A dark well from where you nobody but you can help yourself out from. Now, although several people argue that such experiences are a necessity in life, to understand it better and to add to your variety of experiences, to be comfortable in your own skin and to face adversity bravely; but you know you've reached a new low when the zest of life leaves you altogether and scenarios of a world without yourself in it often occupy your mind. While some take professional help, some reach out to their dear ones and some power through, many give in to their proclivities.

Why?

While we might never be able to answer the question in its true philosophical sense, we can endeavour to empathise with the victims, and try to understand their convictions, motivations and compulsions.

In doing so, we build a more sensitive society which could prove to be a net for this drastic step and identify vulnerable groups around us, a burning need of today. More so in a country such as India, which riddled with its subsistential social and economic problems, added to the stigma surrounding mental health, rarely gives us the opportunity of attempting to understand or even recognising this as a real problem.

If we try to analyse past data, and interact more freely regarding this, we will ultimately strengthen both the spirit and emotional quotient of us as a people, and perhaps help the law enforcement and formulation agencies to educate people better.

**Now, how do we study suicides in India?**

## Data

We will here try to get an analytical peek at Indian Suicide data for the years of 2001 to 2012, as provided publicly by the National Crime Records Bureau (NCRB) of India and dabble with a little correlation according to the 2011 Indian Census Data.

In [5]:
```
# Importing necessary libraries
!pip3 install geopandas --quiet

import pandas as pd
import geopandas as gpd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import plotly.express as px

import warnings
warnings.filterwarnings("ignore")
```

In [6]:
```
# Reading file path
filepath='./data/Suicides in India/Suicides in India 2001-2012.csv'
# Setting the 1st column as index column and reading dataset from filepath
data=pd.read_csv(filepath)
```

In [7]:
```
print(data.shape)
display(data)
```

```
(237519, 7)
```

| | State | Year | Type_code | Type | Gender | Age_group | Total |
|---|---|---|---|---|---|---|---|
| **0** | A & N Islands | 2001 | Causes | Illness (Aids/STD) | Female | 0-14 | 0 |
| **1** | A & N Islands | 2001 | Causes | Bankruptcy or Sudden change in Economic | Female | 0-14 | 0 |
| **2** | A & N Islands | 2001 | Causes | Cancellation/Non-Settlement of Marriage | Female | 0-14 | 0 |
| **3** | A & N Islands | 2001 | Causes | Physical Abuse (Rape/Incest Etc.) | Female | 0-14 | 0 |
| **4** | A & N Islands | 2001 | Causes | Dowry Dispute | Female | 0-14 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **237514** | West Bengal | 2012 | Social_Status | Seperated | Male | 0-100+ | 149 |
| **237515** | West Bengal | 2012 | Social_Status | Widowed/Widower | Male | 0-100+ | 233 |
| **237516** | West Bengal | 2012 | Social_Status | Married | Male | 0-100+ | 5451 |
| **237517** | West Bengal | 2012 | Social_Status | Divorcee | Male | 0-100+ | 189 |
| **237518** | West Bengal | 2012 | Social_Status | Never Married | Male | 0-100+ | 2658 |

237519 rows × 7 columns

The following is a quick glance at the dataset provided in csv form

In [8]:
```
# getting a peek at our dataset
data.sample(10)
```

Out[8]:

| | State | Year | Type_code | Type | Gender | Age_group | Total |
|---|---|---|---|---|---|---|---|
| **163691** | Nagaland | 2003 | Means_adopted | By Hanging | Male | 60+ | 0 |
| **202283** | Sikkim | 2012 | Causes | Unemployment | Female | 45-59 | 0 |
| **147863** | Manipur | 2011 | Means_adopted | By Overdose of sleeping pills | Male | 0-14 | 0 |
| **99601** | Jammu & Kashmir | 2009 | Professional_Profile | Self-employed (Business activity) | Male | 30-44 | 5 |
| **60130** | Daman & Diu | 2011 | Professional_Profile | Others (Please Specify) | Male | 60+ | 0 |
| **41620** | Chhattisgarh | 2002 | Professional_Profile | Others (Please Specify) | Male | 60+ | 38 |
| **116724** | Kerala | 2004 | Causes | Illegitimate Pregnancy | Male | 15-29 | 0 |
| **24602** | Assam | 2008 | Professional_Profile | Farming/Agriculture Activity | Female | 0-14 | 0 |
| **167237** | Nagaland | 2010 | Causes | Property Dispute | Female | 15-29 | 0 |
| **140023** | Maharashtra | 2009 | Means_adopted | By Over Alcoholism | Male | 0-14 | 0 |

In [9]:

```
# DATA CLEANING

# We remove categories such as all india, states and uts, as it leads to duplicacy in our results
data=data[(data['State']!='Total (All India)') & (data['State']!='Total (States)') & (data['State']!='Total (Uts)')]
```

In [10]:

```
# Data cleaning

# We select a type code as a dataframe to judge parameters such as gender and state, because keeping other type codes
causes=data[data['Type_code']=='Causes']
```

First thing that comes to mind. Which groups are more susceptible to committing suicides? Let's have a look

In [11]:

```
# Checking for null values
data.isnull().sum()
```

Out[11]:

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

# Where?

A look at the raw numbers of suicides in various states, across years

In [12]:

```
state=causes.groupby('State')['Total'].sum().to_frame()
```

In [13]:

```
state.sort_values(by='Total', ascending=True, inplace=True)
state.style.background_gradient(subset='Total', cmap='Reds')
```

Out[13]:

| State | Total |
|---|---|
| Lakshadweep | 10 |
| Daman & Diu | 279 |
| Nagaland | 347 |
| Manipur | 421 |
| D & N Haveli | 686 |
| Mizoram | 834 |
| Chandigarh | 1034 |
| Meghalaya | 1086 |
| Arunachal Pradesh | 1328 |
| A & N Islands | 1623 |
| Sikkim | 1924 |
| Jammu & Kashmir | 2968 |
| Goa | 3475 |
| Uttarakhand | 3702 |
| Himachal Pradesh | 5319 |
| Puducherry | 6429 |
| Tripura | 9194 |
| Bihar | 9245 |
| Punjab | 9270 |
| Jharkhand | 9950 |
| Delhi (Ut) | 16857 |
| Haryana | 29437 |
| Assam | 34469 |
| Uttar Pradesh | 46680 |
| Rajasthan | 51027 |
| Odisha | 53448 |
| Chhattisgarh | 60495 |
| Gujarat | 66177 |
| Madhya Pradesh | 90307 |
| Kerala | 107936 |
| Karnataka | 146965 |
| West Bengal | 161030 |
| Andhra Pradesh | 162820 |
| Tamil Nadu | 163813 |
| Maharashtra | 180389 |

In [15]:

```
# Reading geopandas shape file of
shp_gdf = gpd.read_file('./data/India GIS Data/India States/Indian_states.shp')
shp_gdf.set_index('st_nm', inplace = True)

# changing a few names according to the original data file, so that our data is able to be mapped
shp_gdf.rename(index={'Andaman & Nicobar Island':'A & N Islands', 'Arunanchal Pradesh': 'Arunachal Pradesh', 'Dadara &
```

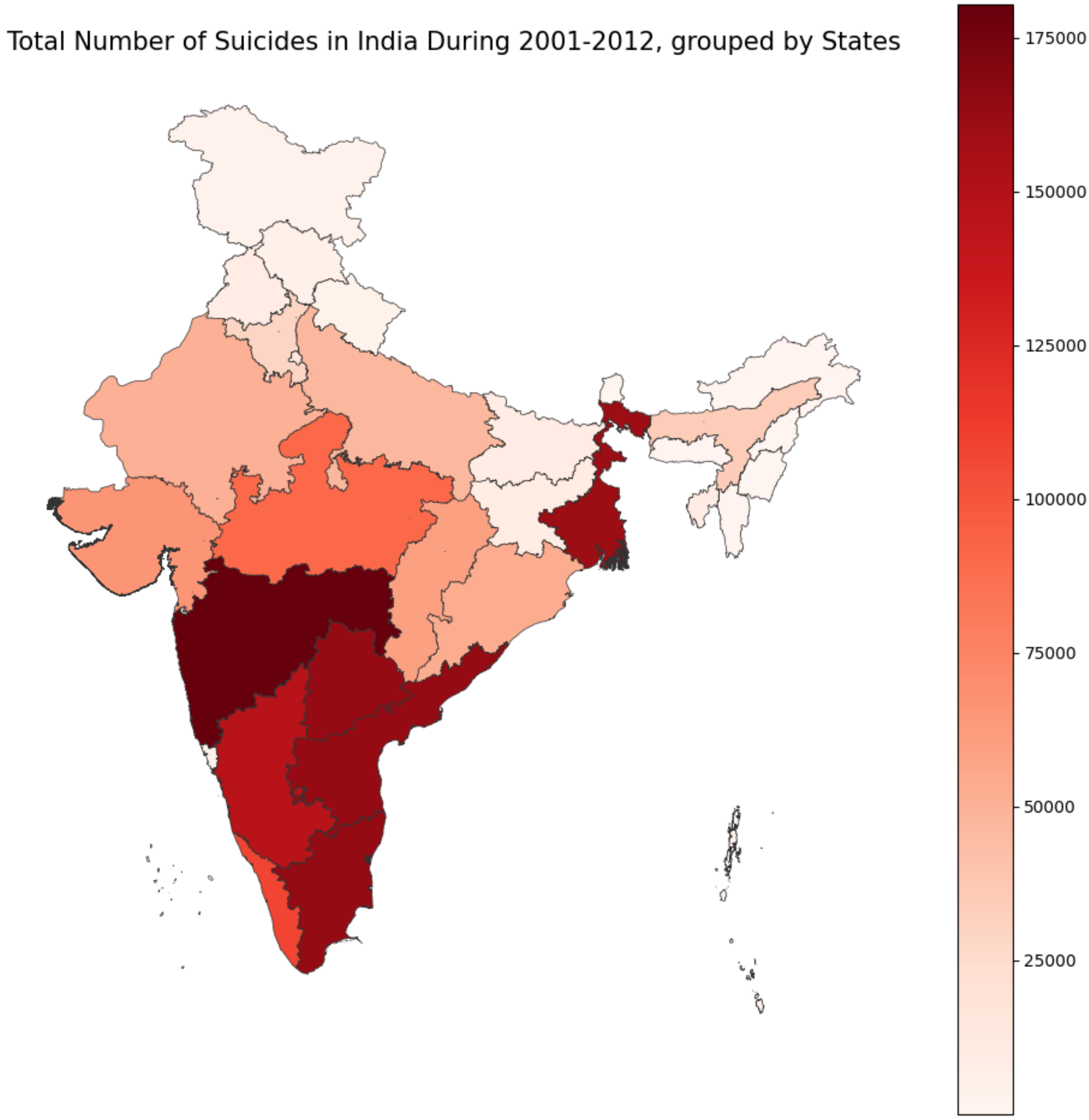
In [16]:

```
# joining both the tables
merged = shp_gdf.join(state.iloc[:, 0])

# as telangana was formed from andhra pradesh after 2012- the end of this data collection, we will input the same valu
merged.at['Telangana', 'Total']=merged.at['Andhra Pradesh', 'Total']
```

In [19]:

```
fig, ax = plt.subplots(1, figsize=(12, 12))
ax.axis('off')
ax.set_title('Total Number of Suicides in India During 2001-2012, grouped by States',
             fontdict={'fontsize': '15', 'fontweight' : '5'})
fig = merged.plot(column='Total', cmap='Reds', linewidth=0.5, ax=ax, edgecolor='0.2',legend=True)
print(fig)
```

```
AxesSubplot(0.125,0.148351;0.62x0.693299)
```




In [20]:
```
# collecting information on different types of 'type_code' and their numbers
data['Type_code'].value_counts()
```

Out[20]:
```
Causes                  109200
Means_adopted            67200
Professional_Profile     49263
Education_Status          6720
Social_Status             4200
Name: Type_code, dtype: int64
```

In [21]:
```
# creating different datasets for different type codes
means=data[data['Type_code']=='Means_adopted']
professional_profile=data[data['Type_code']=='Professional_Profile']
edu_status=data[data['Type_code']=='Education_Status']
social_status=data[data['Type_code']=='Social_Status']
```

In [22]:
```
# Data cleaning

# Replacing vague and long headings and grouping similar or repetitive ones together
causes['Type'].replace(to_replace=['Causes Not known', 'Other Causes (Please Specity)', 'Other Prolonged Illness', 'B
```

## Why?

We get a broad look at reasons behind suicides. This might not feel very intuitive, so we will later group them according to other factors such as age and gender

In [23]:
```
why=causes.groupby('Type')['Total'].sum().sort_values(ascending=False).to_frame()
why.style.background_gradient(subset='Total', cmap='Reds')
```

Out[23]:

| Type | Total |
|---|---|
| Unknown | 453119 |
| Family Problems | 341952 |
| Prolonged Illness | 194565 |
| Insanity/Mental Illness | 94229 |
| Love Affairs | 45039 |
| Bankruptcy or Sudden change in Economic Status | 35410 |
| Poverty | 32684 |
| Dowry Dispute | 31970 |
| Drug Abuse/Addiction | 30046 |
| Unemployment | 27365 |
| Failure in Examination | 27005 |
| Property Dispute | 18652 |
| Suspected/Illicit Relation | 14911 |
| Fall in Social Reputation | 13464 |
| Professional/Career Problem | 12554 |
| Cancellation/Non-Settlement of Marriage | 11296 |
| Death of Dear Person | 10321 |
| Cancer | 9058 |
| Illness (Aids/STD) | 8723 |
| Not having Children (Barrenness/Impotency | 8588 |
| Paralysis | 7286 |
| Divorce | 4133 |
| Physical Abuse (Rape/Incest Etc.) | 3992 |
| Illegitimate Pregnancy | 2494 |
| Ideological Causes/Hero Worshipping | 2118 |

In [24]:

```
# Data cleaning

# replacing vague and long headings and grouping similar or repetitive ones together
to_replace=['By Hanging', 'By Consuming Insecticides', 'By Other means (please specify)', 'By Fire/Self Immolation',
value=['Hanging', 'Consuming Insecticides', 'Undescribed Means', 'Self-Immolation', 'Drowning', 'Letting vehicles run
means['Type'].replace(to_replace=to_replace, value=value, inplace=True)
```

# How?

In [25]:

```
how=means.groupby('Type')['Total'].sum().sort_values(ascending=False).to_frame()
how.style.background_gradient(subset='Total', cmap='Reds')
```

Out[25]:

| Type | Total |
|---|---|
| Hanging | 460955 |
| Consuming Insecticides | 275501 |
| Consuming Poison | 231178 |
| Undescribed Means | 144370 |
| Self-Immolation | 128006 |
| Drowning | 96711 |
| Letting vehicles run over | 45299 |
| Jumping off | 24114 |
| Over Alcoholism | 15973 |
| Electrocution | 10816 |
| Sleeping pills overdose | 9960 |
| Fire arms | 6294 |
| Injury self infliction | 5093 |
| Machines | 1661 |

In [26]:

```
# Data cleaning

# Replacing vague and long headings and grouping similar or repetitive ones together
to_replace=['Others (Please Specify)', 'Farming/Agriculture Activity', 'Service (Private)', 'Self-employed (Business
```

```
value=['Unspecified', 'Farming/Agriculture', 'Private Service', 'Self-employed', 'Unspecified', 'Government Service',
professional_profile['Type'].replace(to_replace=to_replace, value=value, inplace=True)
```

# Who?

### Professional Profile

In [27]:
```
who=professional_profile.groupby('Type')['Total'].sum().sort_values(ascending=False).to_frame()
who.style.background_gradient(subset='Total', cmap='Reds')
```

Out[27]:

| Type | Total |
|---|---|
| Unspecified | 508351 |
| House Wife | 285243 |
| Farming/Agriculture | 197923 |
| Private Service | 115472 |
| Unemployed | 114374 |
| Self-employed | 78112 |
| Student | 74323 |
| Public Sector Undertaking | 30786 |
| Government Service | 23325 |
| Retired | 11334 |

### Educational Profile

In [28]:
```
who_ed=edu_status.groupby('Type')['Total'].sum().sort_values(ascending=False).to_frame()
who_ed.style.background_gradient(subset='Total', cmap='Reds')
```

Out[28]:

| Type | Total |
|---|---|
| Primary | 362827 |
| Middle | 342971 |
| No Education | 321757 |
| Matriculate/Secondary | 256566 |
| Hr. Secondary/Intermediate/Pre-Universit | 118908 |
| Graduate | 31274 |
| Diploma | 14153 |
| Post Graduate and Above | 7475 |

### Social Profile

In [29]:
```
who_so=social_status.groupby('Type')['Total'].sum().sort_values(ascending=False).to_frame()
who_so.style.background_gradient(subset='Total', cmap='Reds')
```

Out[29]:

| Type | Total |
|---|---|
| Married | 1021774 |
| Never Married | 318301 |
| Widowed/Widower | 62113 |
| Seperated | 38471 |
| Divorcee | 15272 |

### Age

In [30]:
```
#age group wise suicides
age=causes.groupby('Age_group')['Total'].sum().to_frame()
age.style.background_gradient(subset='Total', cmap='Reds')
```

Out[30]:

| Age_group | Total |
|---|---|
| 0-14 | 32685 |
| 15-29 | 509776 |
| 30-44 | 488713 |
| 45-59 | 294333 |
| 60+ | 115467 |

**Gender**

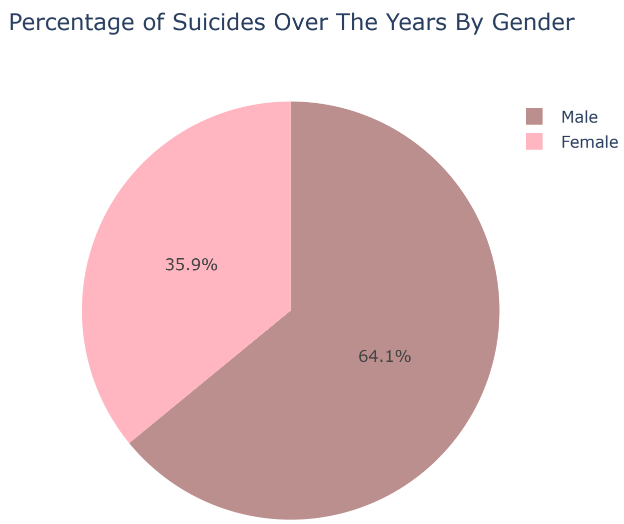
The raw number of suicides by men greatly surpasses the number of suicides by women. Among other factors, this is also due to men overnumbering women population-wise too

In [31]:
```
gender=causes.groupby('Gender')['Total'].sum().to_frame()
gender
```

Out[31]:

| Gender | Total |
|---|---|
| Female | 517736 |
| Male | 923238 |

In [32]:
```
colours=['rosybrown', 'lightpink']
fig = px.pie(gender, values=gender['Total'], names=gender.index, title='Percentage of Suicides Over The Years By Gend
fig.show()
```




# When?

In [33]:
```
year=causes.groupby('Year')['Total'].sum().to_frame()
year.style.background_gradient(subset='Total', cmap='OrRd')
```

Out[33]:

| Year | Total |
|---|---|
| 2001 | 108506 |
| 2002 | 110417 |
| 2003 | 110851 |
| 2004 | 113697 |
| 2005 | 113914 |
| 2006 | 118112 |
| 2007 | 122637 |
| 2008 | 125017 |
| 2009 | 127151 |
| 2010 | 134599 |
| 2011 | 135585 |
| 2012 | 120488 |

In [34]:
```
women=causes[causes['Gender']=='Female']
women_year=women.groupby('Year')['Total'].sum().to_frame()
women_year.rename(columns={'Total': 'Female'}, inplace=True)
```

In [35]:
```
men=causes[causes['Gender']=='Male']
men_year=men.groupby('Year')['Total'].sum().to_frame()
men_year.rename(columns={'Total': 'Male'}, inplace=True)
```

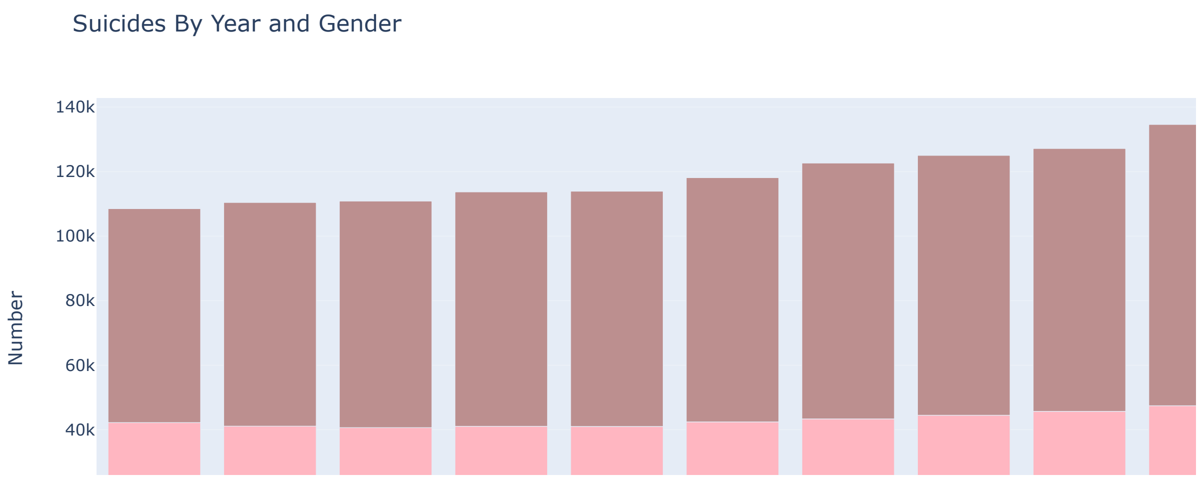
## Male and Female Suicides by Year

In [36]:
```
gender_year=men_year.join(women_year)
gender_year.style.background_gradient(cmap='OrRd', axis=0)
```

Out[36]:

| Year | Male | Female |
|---|---|---|
| 2001 | 66314 | 42192 |
| 2002 | 69332 | 41085 |
| 2003 | 70221 | 40630 |
| 2004 | 72651 | 41046 |
| 2005 | 72916 | 40998 |
| 2006 | 75702 | 42410 |
| 2007 | 79295 | 43342 |
| 2008 | 80544 | 44473 |
| 2009 | 81471 | 45680 |
| 2010 | 87180 | 47419 |
| 2011 | 87839 | 47746 |
| 2012 | 79773 | 40715 |

In [37]:
```
fig = px.bar(gender_year, x=year.index, y=['Female', 'Male'], title='Suicides By Year and Gender', labels={'value':'N
fig.show()
```




## Male and Female Suicides By Year and State

In [38]:
```
state_2011 = (causes.groupby(["State","Gender", 'Year'])
        .agg(value=pd.NamedAgg(column="Total",aggfunc="sum"))['value']
        .unstack( fill_value=0)
        .rename_axis(None, axis=1)
        .unstack())
state_2011['Total']=state_2011.sum(axis=1, numeric_only=True)
state_2011.style.background_gradient(cmap='Reds')
```

Out[38]:

| | 2001 | | 2002 | | 2003 | | 2004 | | 2005 | | 2006 | | 2007 | | 2008 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Gender** | **Female** | **Male** | **Female** | **Male** | **Female** | **Male** | **Female** | **Male** | **Female** | **Male** | **Female** | **Male** | **Female** | **Male** | **Female** | **Male** | **Fema** |
| **State** | | | | | | | | | | | | | | | | | |
| **A & N Islands** | 50 | 79 | 53 | 91 | 43 | 70 | 41 | 81 | 53 | 86 | 37 | 96 | 56 | 100 | 45 | 98 | 4 |
| **Andhra Pradesh** | 4143 | 6379 | 4122 | 7571 | 3896 | 7513 | 4444 | 9082 | 4577 | 8865 | 4413 | 8863 | 4933 | 9949 | 4633 | 9721 | 46 |
| **Arunachal Pradesh** | 35 | 76 | 43 | 71 | 27 | 54 | 30 | 49 | 21 | 49 | 33 | 96 | 31 | 98 | 33 | 77 | 3 |
| **Assam** | 811 | 1836 | 808 | 1702 | 920 | 1676 | 894 | 1945 | 895 | 1951 | 915 | 2116 | 946 | 2116 | 1019 | 1970 | 108 |
| **Bihar** | 296 | 307 | 294 | 426 | 248 | 351 | 139 | 212 | 216 | 327 | 319 | 299 | 431 | 534 | 406 | 609 | 53 |
| **Chandigarh** | 31 | 39 | 34 | 53 | 35 | 68 | 23 | 52 | 26 | 63 | 24 | 56 | 22 | 60 | 25 | 58 | |
| **Chhattisgarh** | 1385 | 2640 | 1321 | 2629 | 1327 | 2592 | 1509 | 2986 | 1576 | 3305 | 1566 | 3060 | 1583 | 3256 | 1697 | 3248 | 192 |
| **D & N Haveli** | 15 | 35 | 18 | 32 | 22 | 30 | 19 | 20 | 28 | 41 | 19 | 23 | 32 | 44 | 28 | 32 | 2 |
| **Daman & Diu** | 4 | 10 | 2 | 15 | 6 | 18 | 8 | 5 | 7 | 25 | 8 | 14 | 9 | 6 | 5 | 14 | |
| **Delhi (Ut)** | 509 | 730 | 410 | 643 | 412 | 741 | 457 | 799 | 468 | 777 | 532 | 960 | 515 | 966 | 508 | 795 | 54 |
| **Goa** | 89 | 167 | 83 | 226 | 87 | 213 | 102 | 212 | 85 | 197 | 92 | 183 | 92 | 178 | 104 | 183 | 9 |
| **Gujarat** | 2142 | 2649 | 1938 | 2706 | 1916 | 2650 | 1926 | 2850 | 1863 | 2902 | 1987 | 3048 | 2205 | 3375 | 2430 | 3735 | 248 |
| **Haryana** | 643 | 1364 | 630 | 1570 | 614 | 1613 | 628 | 1454 | 553 | 1493 | 550 | 1766 | 581 | 1852 | 682 | 1974 | 68 |
| **Himachal Pradesh** | 130 | 177 | 126 | 208 | 146 | 240 | 135 | 236 | 145 | 214 | 156 | 301 | 144 | 258 | 213 | 417 | 19 |
| **Jammu & Kashmir** | 62 | 91 | 88 | 96 | 75 | 63 | 61 | 51 | 98 | 196 | 110 | 152 | 81 | 153 | 133 | 177 | 15 |
| **Jharkhand** | 109 | 141 | 119 | 153 | 119 | 153 | 184 | 233 | 285 | 523 | 335 | 521 | 409 | 880 | 309 | 602 | 36 |
| **Karnataka** | 4010 | 7871 | 4190 | 8080 | 3986 | 8375 | 3919 | 8018 | 3850 | 7707 | 4164 | 8048 | 3912 | 8392 | 4006 | 8216 | 398 |
| **Kerala** | 2785 | 6787 | 2645 | 7165 | 2503 | 6935 | 2455 | 6598 | 2414 | 6830 | 2443 | 6583 | 2374 | 6588 | 2439 | 6130 | 243 |
| **Lakshadweep** | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 2 | 1 | 0 | 0 | |
| **Madhya Pradesh** | 3324 | 3536 | 3320 | 3579 | 3224 | 3538 | 3161 | 3634 | 2597 | 2851 | 2932 | 3503 | 2859 | 3470 | 3264 | 4365 | 40 |
| **Maharashtra** | 5280 | 9338 | 5082 | 9447 | 4950 | 9810 | 4826 | 9903 | 4823 | 9603 | 4984 | 10510 | 4764 | 10420 | 4485 | 9889 | 45 |
| **Manipur** | 17 | 24 | 9 | 30 | 5 | 21 | 9 | 32 | 4 | 23 | 12 | 24 | 11 | 28 | 11 | 23 | |
| **Meghalaya** | 18 | 69 | 18 | 49 | 9 | 32 | 16 | 39 | 21 | 50 | 23 | 69 | 27 | 60 | 28 | 57 | 4 |
| **Mizoram** | 8 | 46 | 10 | 56 | 11 | 41 | 7 | 53 | 8 | 47 | 10 | 60 | 6 | 22 | 6 | 35 | |
| **Nagaland** | 15 | 25 | 8 | 19 | 5 | 17 | 6 | 25 | 6 | 21 | 9 | 19 | 7 | 17 | 10 | 32 | |
| **Odisha** | 1963 | 2089 | 1757 | 2631 | 1680 | 2740 | 1671 | 2544 | 1654 | 2554 | 1692 | 2373 | 1820 | 2488 | 1749 | 3155 | 179 |
| **Puducherry** | 206 | 323 | 157 | 410 | 194 | 388 | 201 | 338 | 198 | 340 | 177 | 349 | 198 | 319 | 173 | 334 | 19 |
| **Punjab** | 252 | 396 | 110 | 397 | 125 | 506 | 129 | 516 | 106 | 482 | 162 | 610 | 215 | 632 | 226 | 643 | 2 |
| **Rajasthan** | 1236 | 1959 | 1156 | 2092 | 1272 | 2389 | 1239 | 2486 | 1320 | 2858 | 1353 | 2910 | 1283 | 3154 | 1564 | 3602 | 155 |
| **Sikkim** | 32 | 62 | 24 | 54 | 37 | 68 | 42 | 56 | 47 | 62 | 47 | 98 | 38 | 84 | 71 | 216 | 1 |
| **Tamil Nadu** | 4162 | 7128 | 4261 | 6983 | 4718 | 7154 | 4893 | 7946 | 4569 | 7507 | 4872 | 7509 | 5124 | 8687 | 5382 | 9043 | 54 |
| **Tripura** | 391 | 463 | 286 | 493 | 356 | 488 | 337 | 433 | 314 | 401 | 337 | 428 | 286 | 419 | 310 | 442 | 2 |
| **Uttar Pradesh** | 1715 | 1801 | 1924 | 2326 | 1676 | 1987 | 1683 | 1954 | 1677 | 1772 | 1338 | 1761 | 1873 | 2054 | 1909 | 2179 | 19 |
| **Uttarakhand** | 129 | 182 | 156 | 205 | 129 | 262 | 89 | 148 | 93 | 180 | 153 | 173 | 118 | 130 | 87 | 104 | 14 |
| **West Bengal** | 6195 | 7495 | 5883 | 7124 | 5856 | 7424 | 5763 | 7661 | 6401 | 8614 | 6605 | 9120 | 6355 | 8505 | 6483 | 8369 | 615 |

In [39]:

```
to_replace=['Causes Not known', 'Other Causes (Please Specity)', 'Other Prolonged Illness', 'Bankruptcy or Sudden cha
value=['Unknown', 'Unknown', 'Prolonged Illness', 'Bankruptcy or Sudden change in Economic Status', 'Not having Child
data['Type'].replace(to_replace=to_replace, value=value, inplace=True)
```

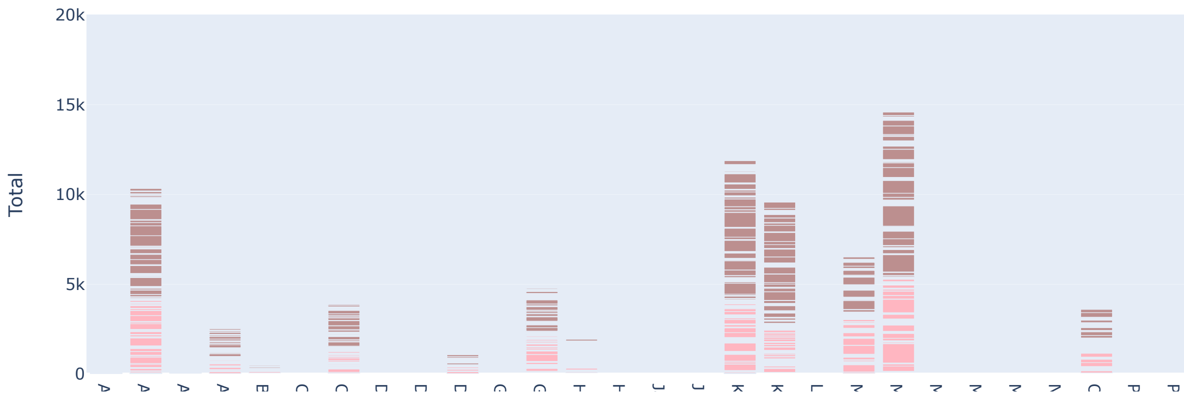
## Male and Female Suicides by Year, State and Causes

In this graph, you can use the slider to view gender wise suicides in each state by year. Also, the stacks here represent the causes, which can be viewed upon hovering them

In [40]:

```
fig = px.bar(causes, x="State", y='Total', color="Gender",
  animation_frame="Year",animation_group='Total',range_y=[0,20000], color_discrete_sequence=['lightpink', 'rosybrown'
fig.show()
```

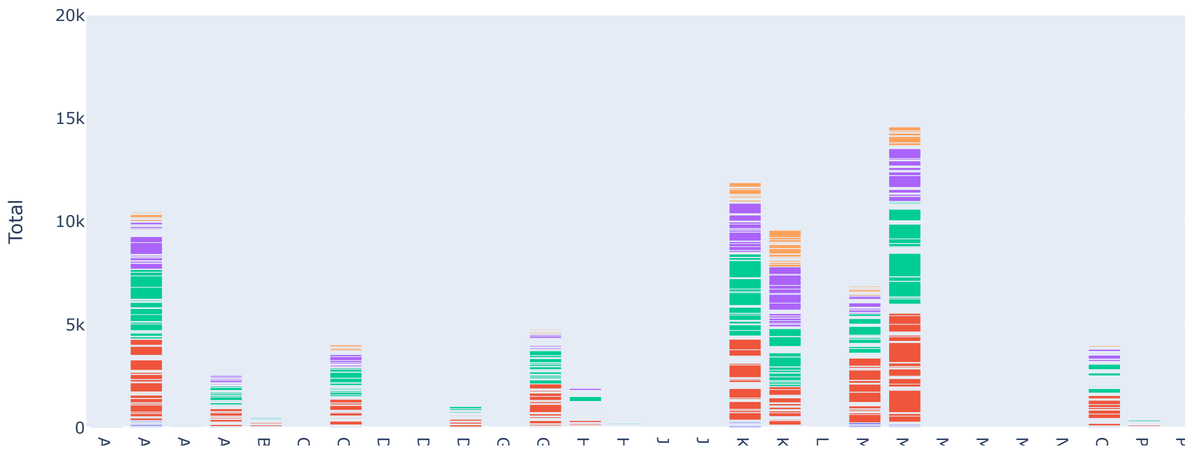
Male and Female Suicides by Year, State amd Causes




# Age group wise suicides according to year, state and causes

In this graph, you can use the slider to view age group wise suicides in each state by year. Also, the stacks here represent the causes, which can be viewed upon hovering them

In [41]:
```
fig = px.bar(causes, x="State", y='Total', color="Age_group",
  animation_frame="Year",animation_group='Total',range_y=[0,20000], hover_name='Type')
fig.show()
```




In [42]:
```
state_age_gen= (causes.groupby(["State","Gender", 'Age_group'])
        .agg(value=pd.NamedAgg(column="Total",aggfunc="sum"))['value']
        .unstack( fill_value=0)
        .rename_axis(None, axis=1)
        .unstack(level=1).unstack().unstack())
state_age_gen.style.background_gradient(cmap='Reds', axis=1)
```

Out[42]:

| | State | A & N Islands | Andhra Pradesh | Arunachal Pradesh | Assam | Bihar | Chandigarh | Chhattisgarh | D & N Haveli | Daman & Diu | Delhi (Ut) | Goa | Gujarat | Haryana | Himachal Pradesh |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Gender | | | | | | | | | | | | | | |
| 0-14 | Female | 31 | 1878 | 26 | 310 | 283 | 9 | 917 | 2 | 5 | 235 | 38 | 486 | 274 | 53 |
| | Male | 22 | 1387 | 23 | 424 | 289 | 7 | 1123 | 4 | 6 | 188 | 23 | 385 | 441 | 54 |
| 15-29 | Female | 298 | 23640 | 220 | 5260 | 1811 | 241 | 9095 | 155 | 62 | 3825 | 523 | 12257 | 3727 | 934 |
| | Male | 332 | 31481 | 352 | 8124 | 1909 | 290 | 12953 | 148 | 69 | 5036 | 697 | 13924 | 7866 | 1140 |
| 30-44 | Female | 133 | 16884 | 126 | 3713 | 1439 | 96 | 5823 | 62 | 21 | 1612 | 294 | 8213 | 2576 | 576 |
| | Male | 331 | 39886 | 398 | 9034 | 1859 | 253 | 14131 | 121 | 67 | 3776 | 960 | 14754 | 8090 | 1189 |
| 45-59 | Female | 57 | 8325 | 16 | 1502 | 382 | 21 | 2799 | 42 | 4 | 364 | 133 | 3810 | 958 | 256 |
| | Male | 254 | 26592 | 144 | 5099 | 859 | 90 | 9108 | 79 | 38 | 1382 | 502 | 8058 | 3990 | 719 |
| 60+ | Female | 31 | 3669 | 3 | 246 | 138 | 11 | 1105 | 19 | 3 | 102 | 117 | 1738 | 288 | 112 |
| | Male | 134 | 9078 | 20 | 757 | 276 | 16 | 3441 | 54 | 4 | 337 | 188 | 2552 | 1227 | 286 |

## Suicides as a function of Population

Mapping Suicides in 2011 as a function of respective states' poplulation according to 2011 census

In [44]:
```
#reading the census file
census=pd.read_csv('./data/India Census 2011/india-districts-census-2011.csv')
```

In [45]:
```
#grouping census according to population
pop=census.groupby('State name')['Population'].sum().to_frame()
```

In [46]:
```
#grouping census according to male population
male=census.groupby('State name')['Male'].sum().to_frame()
```

In [47]:
```
#grouping census according to female population
female=census.groupby('State name')['Female'].sum().to_frame()
```

In [48]:
```
#joining total population table with male population table
pop=pop.join(male)
```

In [49]:
```
#joining total population table with female population table
pop=pop.join(female)
```

2011 state wise population

In [50]:
```
pop.style.background_gradient(cmap='Greens')
```

Out[50]:

| State name | Population | Male | Female |
|---|---|---|---|
| ANDAMAN AND NICOBAR ISLANDS | 380581 | 202871 | 177710 |
| ANDHRA PRADESH | 84580777 | 42442146 | 42138631 |
| ARUNACHAL PRADESH | 1383727 | 713912 | 669815 |
| ASSAM | 31205576 | 15939443 | 15266133 |
| BIHAR | 104099452 | 54278157 | 49821295 |
| CHANDIGARH | 1055450 | 580663 | 474787 |
| CHHATTISGARH | 25545198 | 12832895 | 12712303 |
| DADRA AND NAGAR HAVELI | 343709 | 193760 | 149949 |
| DAMAN AND DIU | 243247 | 150301 | 92946 |
| GOA | 1458545 | 739140 | 719405 |
| GUJARAT | 60439692 | 31491260 | 28948432 |
| HARYANA | 25351462 | 13494734 | 11856728 |
| HIMACHAL PRADESH | 6864602 | 3481873 | 3382729 |
| JAMMU AND KASHMIR | 12541302 | 6640662 | 5900640 |
| JHARKHAND | 32988134 | 16930315 | 16057819 |
| KARNATAKA | 61095297 | 30966657 | 30128640 |
| KERALA | 33406061 | 16027412 | 17378649 |
| LAKSHADWEEP | 64473 | 33123 | 31350 |
| MADHYA PRADESH | 72626809 | 37612306 | 35014503 |
| MAHARASHTRA | 112374333 | 58243056 | 54131277 |
| MANIPUR | 2855794 | 1438586 | 1417208 |
| MEGHALAYA | 2966889 | 1491832 | 1475057 |
| MIZORAM | 1097206 | 555339 | 541867 |
| NAGALAND | 1978502 | 1024649 | 953853 |
| NCT OF DELHI | 16787941 | 8987326 | 7800615 |
| ORISSA | 41974218 | 21212136 | 20762082 |
| PONDICHERRY | 1247953 | 612511 | 635442 |
| PUNJAB | 27743338 | 14639465 | 13103873 |
| RAJASTHAN | 68548437 | 35550997 | 32997440 |
| SIKKIM | 610577 | 323070 | 287507 |
| TAMIL NADU | 72147030 | 36137975 | 36009055 |
| TRIPURA | 3673917 | 1874376 | 1799541 |
| UTTAR PRADESH | 199812341 | 104480510 | 95331831 |
| UTTARAKHAND | 10086292 | 5137773 | 4948519 |
| WEST BENGAL | 91276115 | 46809027 | 44467088 |

In [51]:

```
#as the names are all capitals, we will change the format to match that of our suicides tables
for i in pop.index[:]:
    pop.rename(index={i:i.title()}, inplace=True)
#renaming the remaining discrepencies
pop.rename(index={'Andaman And Nicobar Islands':'A & N Islands', 'Dadra And Nagar Haveli':'D & N Haveli', 'Daman And I
pop.index.name='State'
```

In [52]:

```
#grouping our data according to the year 2011, gender and then summing total suicides
state_1_2011 = (causes[causes['Year']==2011].groupby(["State","Gender"])
        .agg(value=pd.NamedAgg(column="Total",aggfunc="sum"))['value']
        .unstack("Gender",fill_value=0)
        .rename_axis(None, axis=1))
state_1_2011['Total Suicides']=state_1_2011.sum(axis=1, numeric_only=True)
state_1_2011.rename(columns={'Female':'Female Suicides', 'Male':'Male Suicides'}, inplace=True)
```

**A table represnting the number of suicides and total population, both gender wise and then per lakh suicides gender wise. It also gives us the average numbers**

In [53]:

```
suicide_1_2011=pop.join(state_1_2011)
suicide_1_2011['per lakh']=(suicide_1_2011['Total Suicides'] / suicide_1_2011['Population'])*100000
suicide_1_2011['male per lakh']=(suicide_1_2011['Male Suicides'] / suicide_1_2011['Male'])*100000
suicide_1_2011['female per lakh']=(suicide_1_2011['Female Suicides'] / suicide_1_2011['Female'])*100000
suicide_1_2011.loc['AVERAGE'] = suicide_1_2011.mean()
suicide_1_2011.sort_values(by='per lakh', ascending=True, inplace=True)
suicide_1_2011.style.background_gradient(cmap='OrRd')
```

Out[53]:

| State | Population | Male | Female | Female Suicides | Male Suicides | Total Suicides | per lakh | male per lakh | p |
|---|---|---|---|---|---|---|---|---|---|
| Lakshadweep | 64473.000000 | 33123.000000 | 31350.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0. |
| Bihar | 104099452.000000 | 54278157.000000 | 49821295.000000 | 349.000000 | 446.000000 | 795.000000 | 0.763693 | 0.821693 | 0. |
| Manipur | 2855794.000000 | 1438586.000000 | 1417208.000000 | 15.000000 | 18.000000 | 33.000000 | 1.155546 | 1.251229 | 1. |
| Nagaland | 1978502.000000 | 1024649.000000 | 953853.000000 | 8.000000 | 25.000000 | 33.000000 | 1.667929 | 2.439860 | 0. |
| Jammu & Kashmir | 12541302.000000 | 6640662.000000 | 5900640.000000 | 143.000000 | 144.000000 | 287.000000 | 2.288439 | 2.168459 | 2. |
| Uttar Pradesh | 199812341.000000 | 104480510.000000 | 95331831.000000 | 2130.000000 | 2713.000000 | 4843.000000 | 2.423774 | 2.596657 | 2. |
| Uttarakhand | 10086292.000000 | 5137773.000000 | 4948519.000000 | 125.000000 | 192.000000 | 317.000000 | 3.142879 | 3.737028 | 2. |
| Punjab | 27743338.000000 | 14639465.000000 | 13103873.000000 | 211.000000 | 755.000000 | 966.000000 | 3.481917 | 5.157292 | 1 |
| Jharkhand | 32988134.000000 | 16930315.000000 | 16057819.000000 | 387.000000 | 825.000000 | 1212.000000 | 3.674048 | 4.872916 | 2 |
| Meghalaya | 2966889.000000 | 1491832.000000 | 1475057.000000 | 42.000000 | 111.000000 | 153.000000 | 5.156917 | 7.440516 | 2. |
| Rajasthan | 68548437.000000 | 35550997.000000 | 32997440.000000 | 1332.000000 | 3016.000000 | 4348.000000 | 6.342960 | 8.483588 | 4. |
| Himachal Pradesh | 6864602.000000 | 3481873.000000 | 3382729.000000 | 174.000000 | 269.000000 | 443.000000 | 6.453397 | 7.725727 | 5 |
| Mizoram | 1097206.000000 | 555339.000000 | 541867.000000 | 13.000000 | 77.000000 | 90.000000 | 8.202653 | 13.865405 | 2 |
| Assam | 31205576.000000 | 15939443.000000 | 15266133.000000 | 900.000000 | 1826.000000 | 2726.000000 | 8.735618 | 11.455858 | 5. |
| Arunachal Pradesh | 1383727.000000 | 713912.000000 | 669815.000000 | 35.000000 | 99.000000 | 134.000000 | 9.683991 | 13.867255 | 5. |
| Chandigarh | 1055450.000000 | 580663.000000 | 474787.000000 | 48.000000 | 57.000000 | 105.000000 | 9.948363 | 9.816365 | 10 |
| Delhi (Ut) | 16787941.000000 | 8987326.000000 | 7800615.000000 | 548.000000 | 1168.000000 | 1716.000000 | 10.221623 | 12.996079 | 7. |
| Gujarat | 60439692.000000 | 31491260.000000 | 28948432.000000 | 2470.000000 | 3912.000000 | 6382.000000 | 10.559286 | 12.422494 | 8. |
| Odisha | 41974218.000000 | 21212136.000000 | 20762082.000000 | 2181.000000 | 3060.000000 | 5241.000000 | 12.486236 | 14.425704 | 10. |
| Madhya Pradesh | 72626809.000000 | 37612306.000000 | 35014503.000000 | 4019.000000 | 5240.000000 | 9259.000000 | 12.748736 | 13.931611 | 11 |
| Haryana | 25351462.000000 | 13494734.000000 | 11856728.000000 | 781.000000 | 2464.000000 | 3245.000000 | 12.800051 | 18.258974 | 6. |
| AVERAGE | 34595856.485714 | 17807721.657143 | 16788134.828571 | 1364.171429 | 2509.685714 | 3873.857143 | 13.089025 | 16.804277 | 9. |
| Daman & Diu | 243247.000000 | 150301.000000 | 92946.000000 | 15.000000 | 18.000000 | 33.000000 | 13.566457 | 11.975968 | 16. |
| Maharashtra | 112374333.000000 | 58243056.000000 | 54131277.000000 | 5060.000000 | 10887.000000 | 15947.000000 | 14.190963 | 18.692357 | 9. |
| Andhra Pradesh | 84580777.000000 | 42442146.000000 | 42138631.000000 | 4957.000000 | 10120.000000 | 15077.000000 | 17.825563 | 23.844223 | 11. |
| West Bengal | 91276115.000000 | 46809027.000000 | 44467088.000000 | 6868.000000 | 9624.000000 | 16492.000000 | 18.068254 | 20.560137 | 15 |
| D & N Haveli | 343709.000000 | 193760.000000 | 149949.000000 | 22.000000 | 41.000000 | 63.000000 | 18.329459 | 21.160198 | 14 |
| Tripura | 3673917.000000 | 1874376.000000 | 1799541.000000 | 256.000000 | 447.000000 | 703.000000 | 19.134891 | 23.847937 | 14. |
| Goa | 1458545.000000 | 739140.000000 | 719405.000000 | 98.000000 | 195.000000 | 293.000000 | 20.088513 | 26.382012 | 13. |
| Karnataka | 61095297.000000 | 30966657.000000 | 30128640.000000 | 4150.000000 | 8472.000000 | 12622.000000 | 20.659528 | 27.358458 | 13. |
| Tamil Nadu | 72147030.000000 | 36137975.000000 | 36009055.000000 | 5681.000000 | 10282.000000 | 15963.000000 | 22.125651 | 28.452065 | 15. |
| Kerala | 33406061.000000 | 16027412.000000 | 17378649.000000 | 2219.000000 | 6212.000000 | 8431.000000 | 25.237935 | 38.758597 | 12. |
| Chhattisgarh | 25545198.000000 | 12832895.000000 | 12712303.000000 | 2229.000000 | 4527.000000 | 6756.000000 | 26.447241 | 35.276530 | 17. |
| Sikkim | 610577.000000 | 323070.000000 | 287507.000000 | 77.000000 | 107.000000 | 184.000000 | 30.135429 | 33.119757 | 26 |
| A & N Islands | 380581.000000 | 202871.000000 | 177710.000000 | 42.000000 | 94.000000 | 136.000000 | 35.734837 | 46.334863 | 23. |
| Puducherry | 1247953.000000 | 612511.000000 | 635442.000000 | 161.000000 | 396.000000 | 557.000000 | 44.633091 | 64.651900 | 25. |

In [54]:

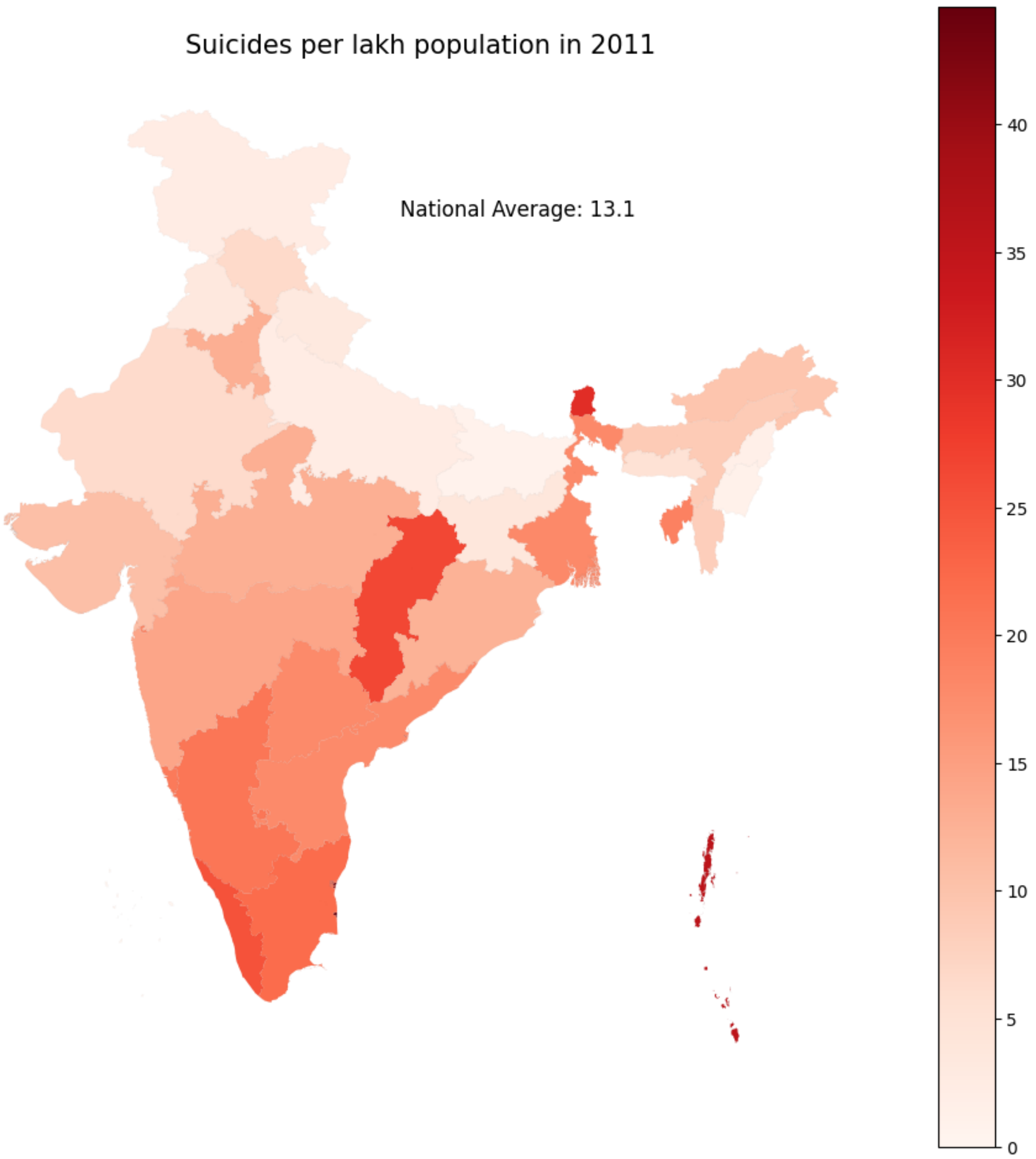
```
merged2 = shp_gdf.join(suicide_1_2011)
merged2
#as telangana was formed from andhra pradesh after 2012- the end of this data collection, we will input the same valu
merged2.at['Telangana', 'per lakh']=merged2.at['Andhra Pradesh', 'per lakh']
merged2.at['Telangana', 'male per lakh']=merged2.at['Andhra Pradesh', 'male per lakh']
merged2.at['Telangana', 'female per lakh']=merged2.at['Andhra Pradesh', 'female per lakh']
```

**Suicide density: suicides per lakh people**

In [55]:

```
fig, ax = plt.subplots(1, figsize=(12, 12))
ax.axis('off')
ax.set_title('Suicides per lakh population in 2011',
             fontdict={'fontsize': '15', 'fontweight' : '5'})
plt.suptitle('National Average: 13.1', x=.5, y=.75)
fig = merged2.plot(column='per lakh', cmap='Reds', linewidth=0.01, ax=ax, edgecolor='0.01',legend=True)
print(fig)
```

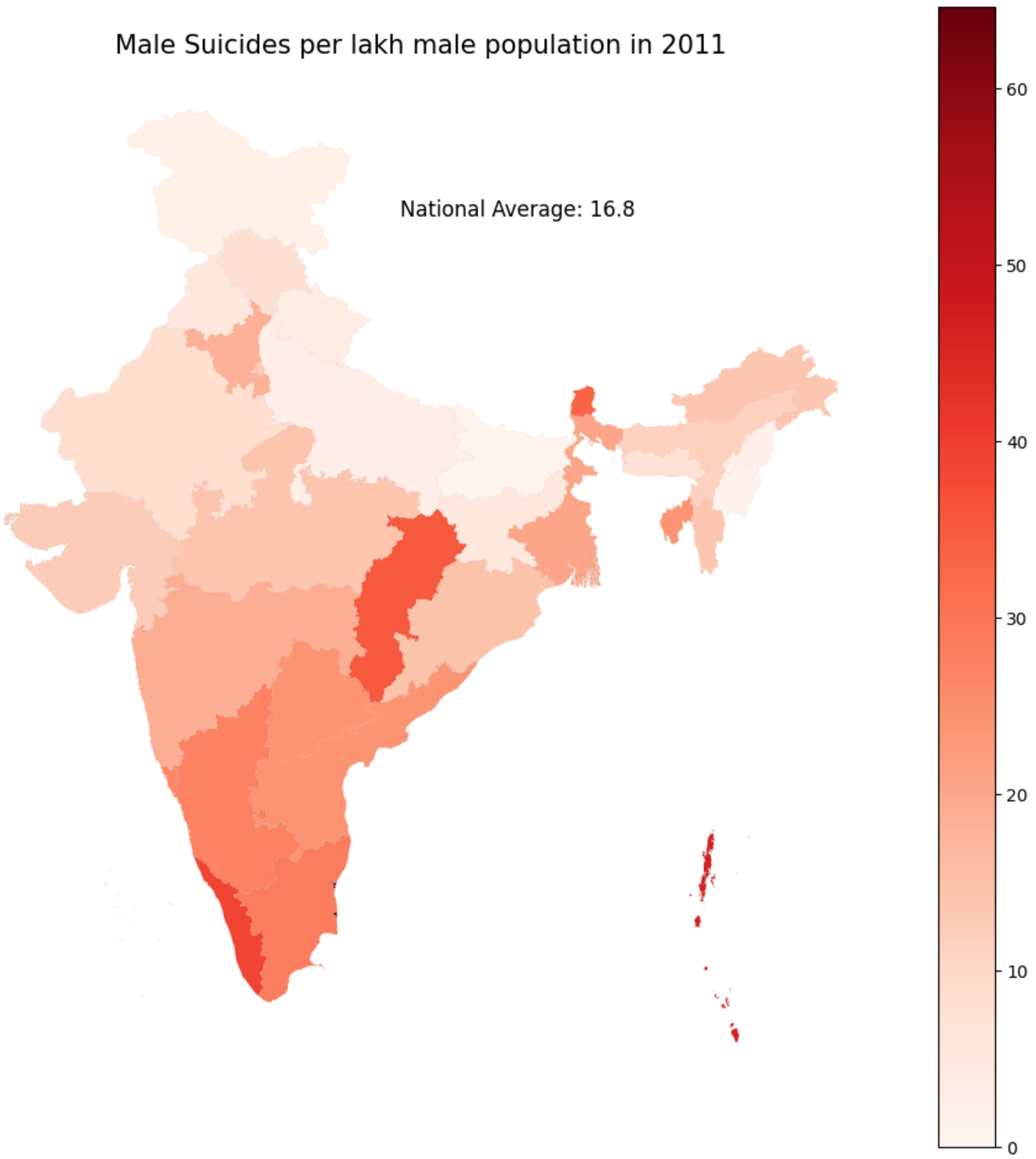
```
AxesSubplot(0.125,0.148351;0.62x0.693299)
```




**Male Suicide density: suicides per lakh males**

```
In [56]: fig, ax = plt.subplots(1, figsize=(12, 12))
         ax.axis('off')
         ax.set_title('Male Suicides per lakh male population in 2011',
                      fontdict={'fontsize': '15', 'fontweight' : '5'})
         plt.suptitle('National Average: 16.8', x=.5, y=.75)
         fig = merged2.plot(column='male per lakh', cmap='Reds', linewidth=0.01, ax=ax, edgecolor='0.01',legend=True)
         print(fig)
```

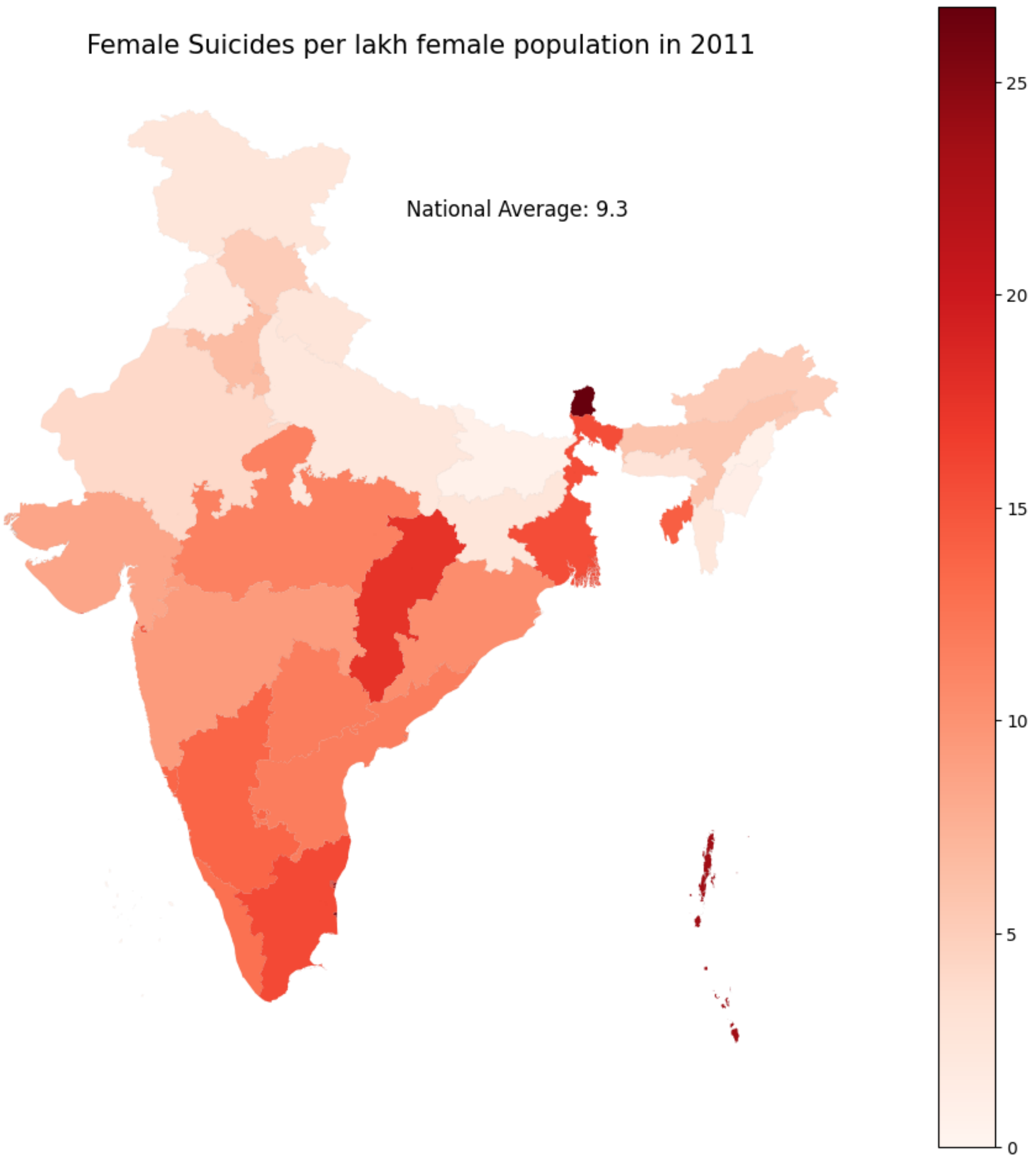
```
AxesSubplot(0.125,0.148351;0.62x0.693299)
```




**Female Suicide density: suicides per lakh women**

In [57]:
```
fig, ax = plt.subplots(1, figsize=(12, 12))
ax.axis('off')
ax.set_title('Female Suicides per lakh female population in 2011',
             fontdict={'fontsize': '15', 'fontweight' : '5'})
plt.suptitle('National Average: 9.3', x=.5, y=.75)
fig = merged2.plot(column='female per lakh', cmap='Reds', linewidth=0.01, ax=ax, edgecolor='0.01',legend=True)
print(fig)
```

```
AxesSubplot(0.125,0.148351;0.62x0.693299)
```




In [58]:
```
age1=data.groupby('Age_group')['Total'].sum().to_frame()
age1.loc["Total"] = age1.sum()
```

In [59]:
```
age=causes.groupby('Age_group')['Total'].sum().to_frame()
age.loc["Total"] = age.sum()
age.style.background_gradient(subset='Total', cmap='OrRd')
```

Out[59]:

| Age_group | Total |
|---|---|
| 0-14 | 32685 |
| 15-29 | 509776 |
| 30-44 | 488713 |
| 45-59 | 294333 |
| 60+ | 115467 |
| Total | 1440974 |

In [60]:
```
data=data[data['Total']>0]
```

In [61]:
```
to_14=data[data['Age_group']=='0-14']
to_29=data[data['Age_group']=='15-29']
to_44=data[data['Age_group']=='30-44']
to_59=data[data['Age_group']=='45-59']
above_60=data[data['Age_group']=='60+']
allage=data[data['Age_group']=='0-100+']
```

In [62]:
```
to_29['Type_code'].unique()
```

Out[62]: array(['Causes', 'Means_adopted', 'Professional_Profile'], dtype=object)

In [63]:
```
to_59['Type_code'].unique()
```

Out[63]: array(['Causes', 'Means_adopted', 'Professional_Profile'], dtype=object)

In [64]:
```
to_44['Type_code'].unique()
```

Out[64]: array(['Causes', 'Means_adopted', 'Professional_Profile'], dtype=object)

In [65]:
```
above_60['Type_code'].unique()
```

Out[65]: array(['Causes', 'Means_adopted', 'Professional_Profile'], dtype=object)

In [66]:
```
allage['Type_code'].unique()
```

Out[66]: array(['Education_Status', 'Social_Status'], dtype=object)

In [67]:
```
to_14[to_14['Type']=='Retired']
```

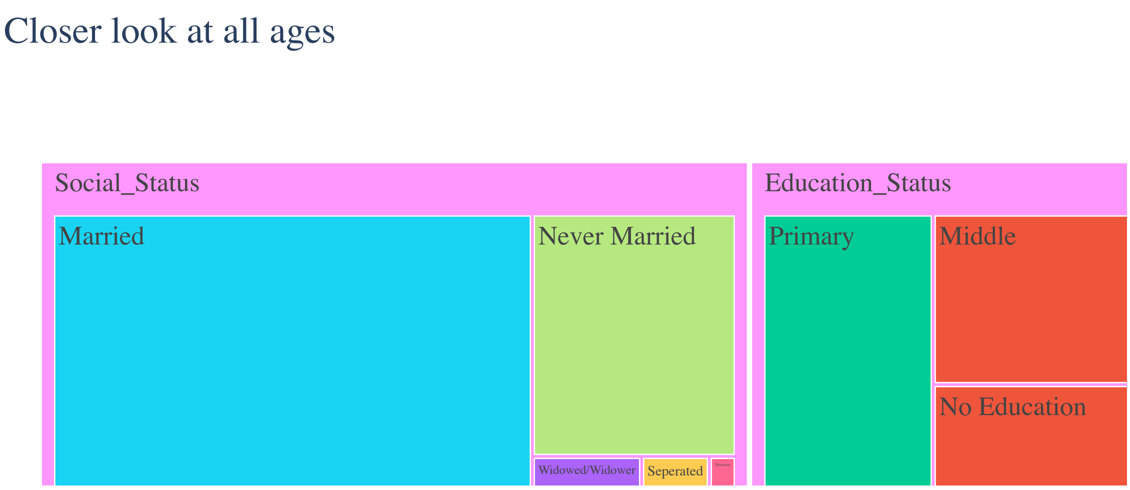
Out[67]:

| | State | Year | Type_code | Type | Gender | Age_group | Total |
|---|---|---|---|---|---|---|---|
| **80907** | Gujarat | 2012 | Professional_Profile | Retired | Female | 0-14 | 3 |
| **84319** | Haryana | 2006 | Professional_Profile | Retired | Male | 0-14 | 1 |
| **132233** | Madhya Pradesh | 2007 | Professional_Profile | Retired | Female | 0-14 | 2 |
| **175528** | Odisha | 2012 | Professional_Profile | Retired | Female | 0-14 | 2 |
| **228939** | Uttarakhand | 2009 | Professional_Profile | Retired | Male | 0-14 | 1 |

# Social and Educational Status of Victims

Since this part of the data is not clasified age-wise, we'll have an overall look. You can hover on these maps to get the numbers

In [68]:
```
fig = px.treemap(allage, path=['Type_code', 'Type'],
                 values='Total',
                 color='Type', title='Closer look at all ages')
fig.update_layout(
    font=dict(
        family="Times",
        size=20
    )
)

fig.show()
```




While the correlation of suicide victims with their social or educational status might be a risky one to make before we know the representation of the same in general populace, one is compelled to notice that the attainment of education makes one less vulnerable and perhaps more rational while taking such a drastic step.

# Age wise visualisation of causes, means adopted and professional profiles of victims

**Ages 0-14**

In [69]:

```
fig = px.treemap(to_14, path=['Type_code', 'Type'],
                 values='Total',
                 color='Type', height=1200, width=1200, title='Closer look at age group of 0-14')
fig.update_layout(
    font=dict(
        family="Times",
        size=20
    )
)


fig.show()
```

Closer look at age group of 0-14

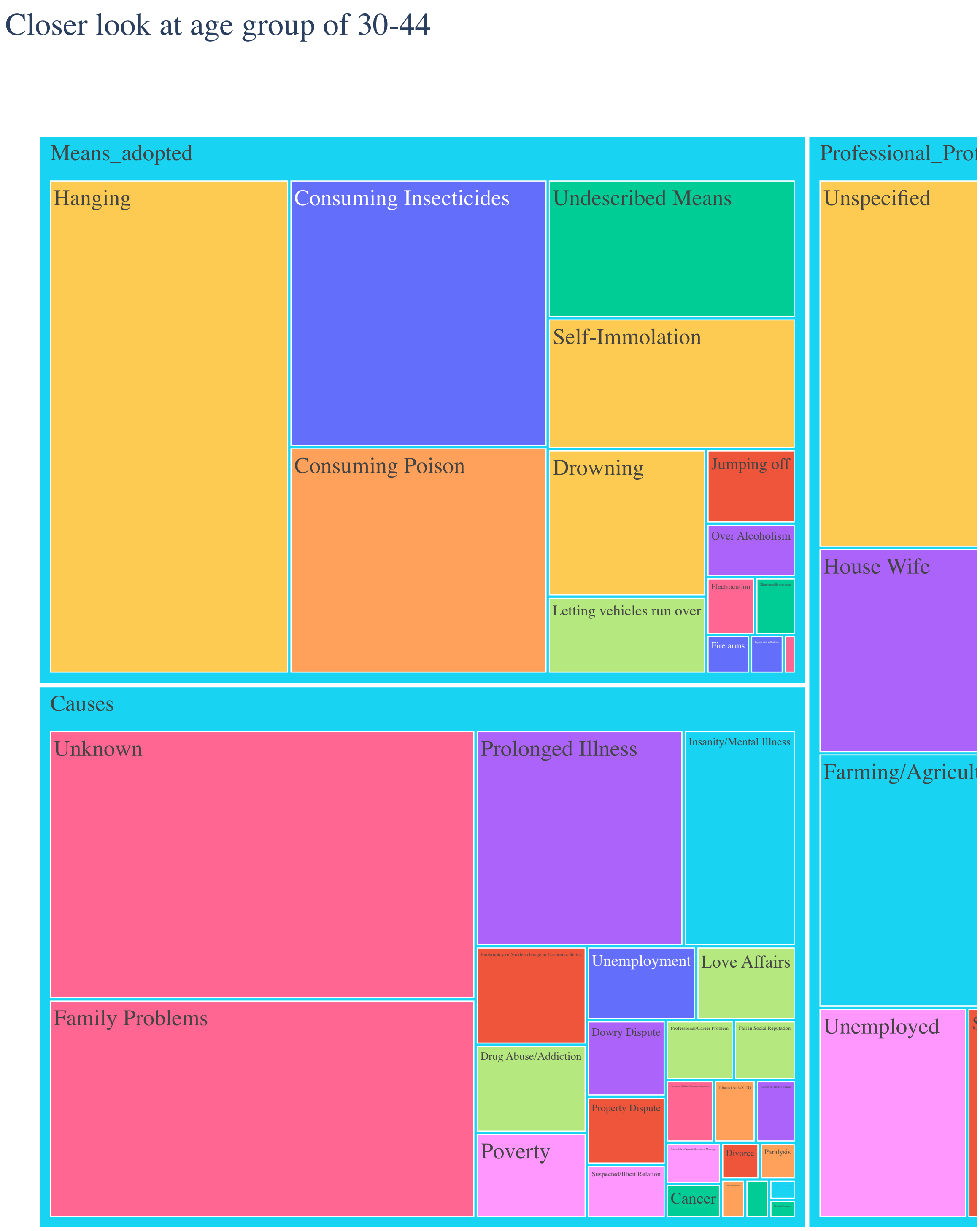

**Ages 15-29**

```
In [70]: fig = px.treemap(to_29, path=['Type_code', 'Type'],
                          values='Total',
                          color='Type', height=1200, width=1200, title='Closer look at age group of 15-29')
         fig.update_layout(
             font=dict(
                 family="Times",
                 size=20
             )
         )


         fig.show()
```

Closer look at age group of 15-29


**Ages 30-44**

In [71]:
```
fig = px.treemap(to_44, path=['Type_code', 'Type'],
                 values='Total',
                 color='Type', height=1200, width=1200, title='Closer look at age group of 30-44')
fig.update_layout(
    font=dict(
        family="Times",
        size=20
    )
)


fig.show()
```




**Ages 45-59**

In [72]:

```
fig = px.treemap(to_59, path=['Type_code', 'Type'],
                 values='Total',
                 color='Type', height=1200, width=1200, title='Closer look at age group of 45-59')
fig.update_layout(
    font=dict(
        family="Times",
        size=20
    )
)


fig.show()
```

Closer look at age group of 45-59


#### Ages 60+

In [73]:
```
fig = px.treemap(above_60, path=['Type_code', 'Type'],
                 values='Total',
                 color='Type', height=1200, width=1200, title='Closer look at age group of 60+')
fig.update_layout(
    font=dict(
        family="Times",
        size=20
    )
)


fig.show()
```

Closer look at age group of 60+