

# Exploratory Data Analysis on Homicides in US

## Importing Necessary Packages

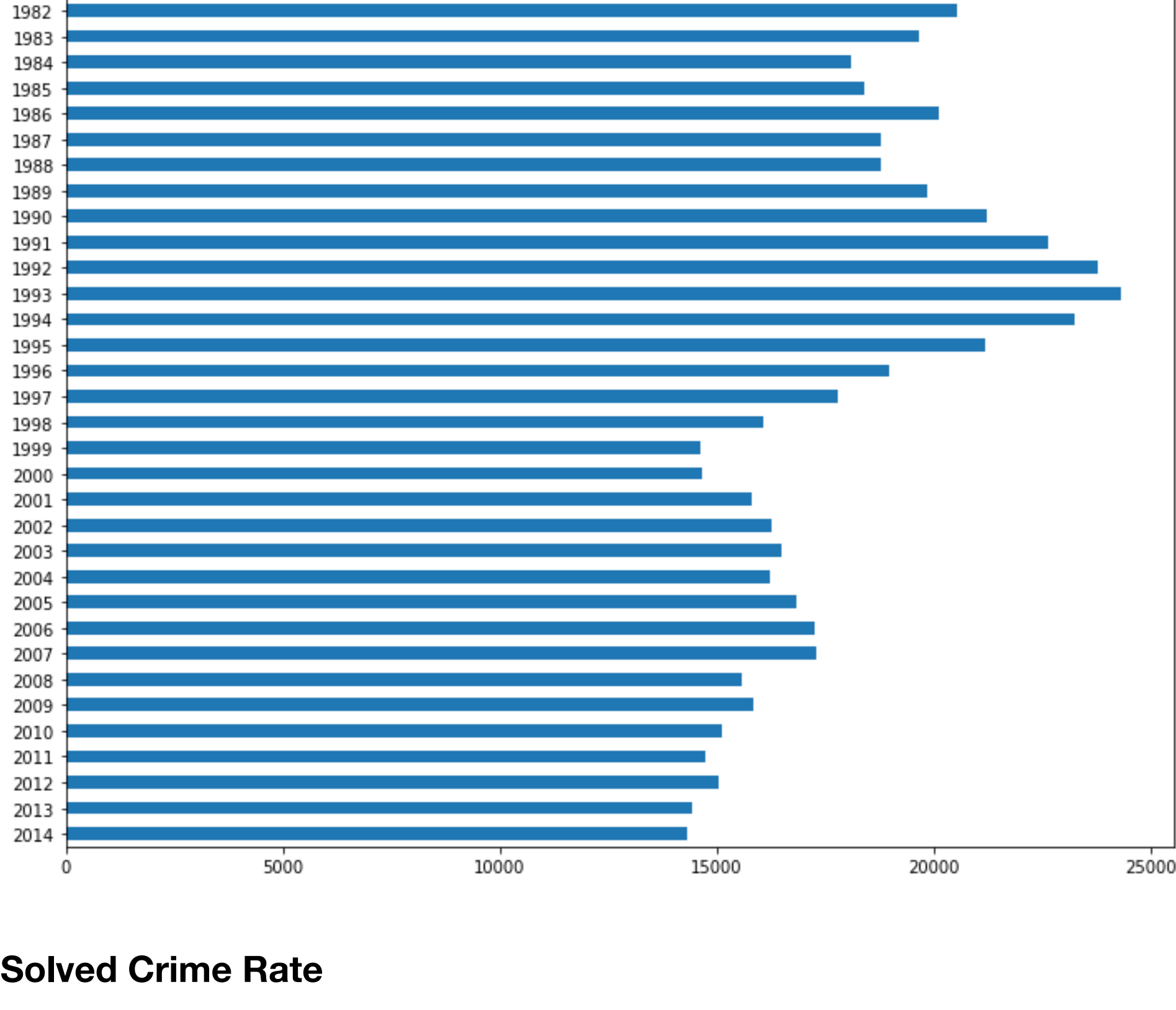
```
In [1]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

## Reading the DataSet

```
In [2]: data=pd.read_csv(r"/Users/krishnakanth/Downloads/database.csv", na_values=['NA'], dtype='unicode')
```

## Homicides in US between 1980 and 2014 by Years

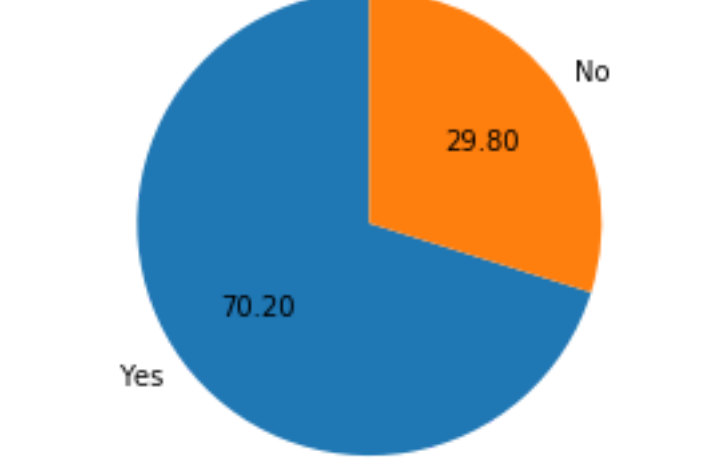
```
In [3]: years = pd.DataFrame(data, columns = ['Year'])
count_years = years.stack().value_counts()
homicides = count_years.sort_index(axis=0, ascending=False)
#plot the total of homicides
homicides.plot(kind='barh', fontsize=10,
               width=0.5,  figsize=(12, 10),
               title='Homicides in US between 1980 and 2014');
```



## Solved Crime Rate

```
In [4]: ## Rate of crimes solved
solved = pd.DataFrame(data, columns = ['Crime Solved'])
resolution = solved.stack().value_counts()
ax = resolution.plot(kind = 'pie',
                    title = 'Crimes solved between 1980 & 2014 (in %)',
                    startangle = 90,
                    autopct='%.2f')
ax.set_ylabel('')
```

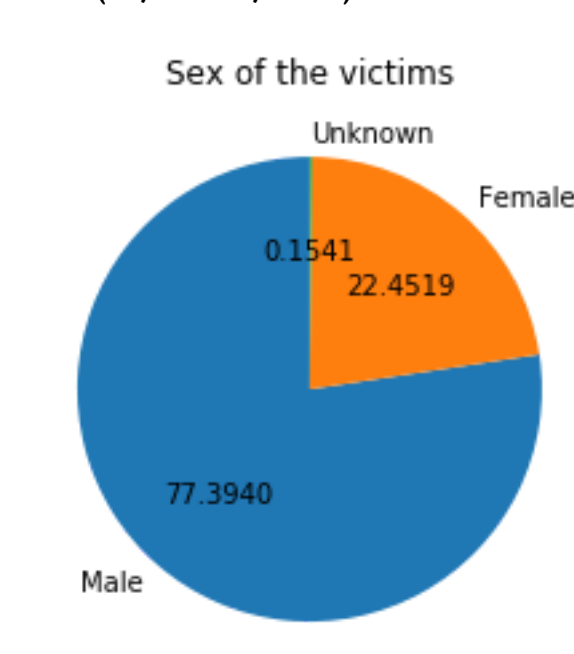
```
Out[4]: Text(0, 0.5, '')
```



## Gender of Victims

```
In [5]: #Gender of victims
sex = pd.DataFrame(data, columns = ['Victim Sex'])
count_sex = sex.stack().value_counts()
ax = count_sex.plot(kind = 'pie',
                   title = 'Sex of the victims',
                   startangle = 90,
                   autopct='%.4f')
ax.set_ylabel('')
```

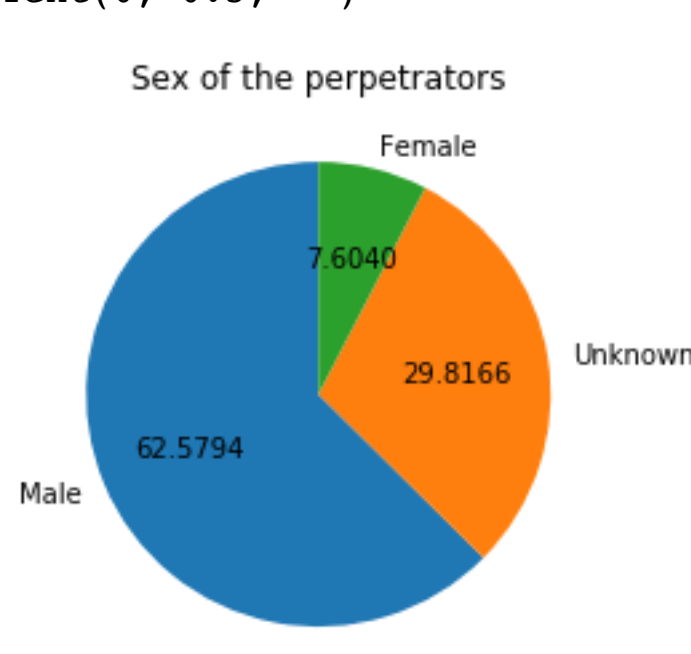
```
Out[5]: Text(0, 0.5, '')
```



## Gender of Perpetrator

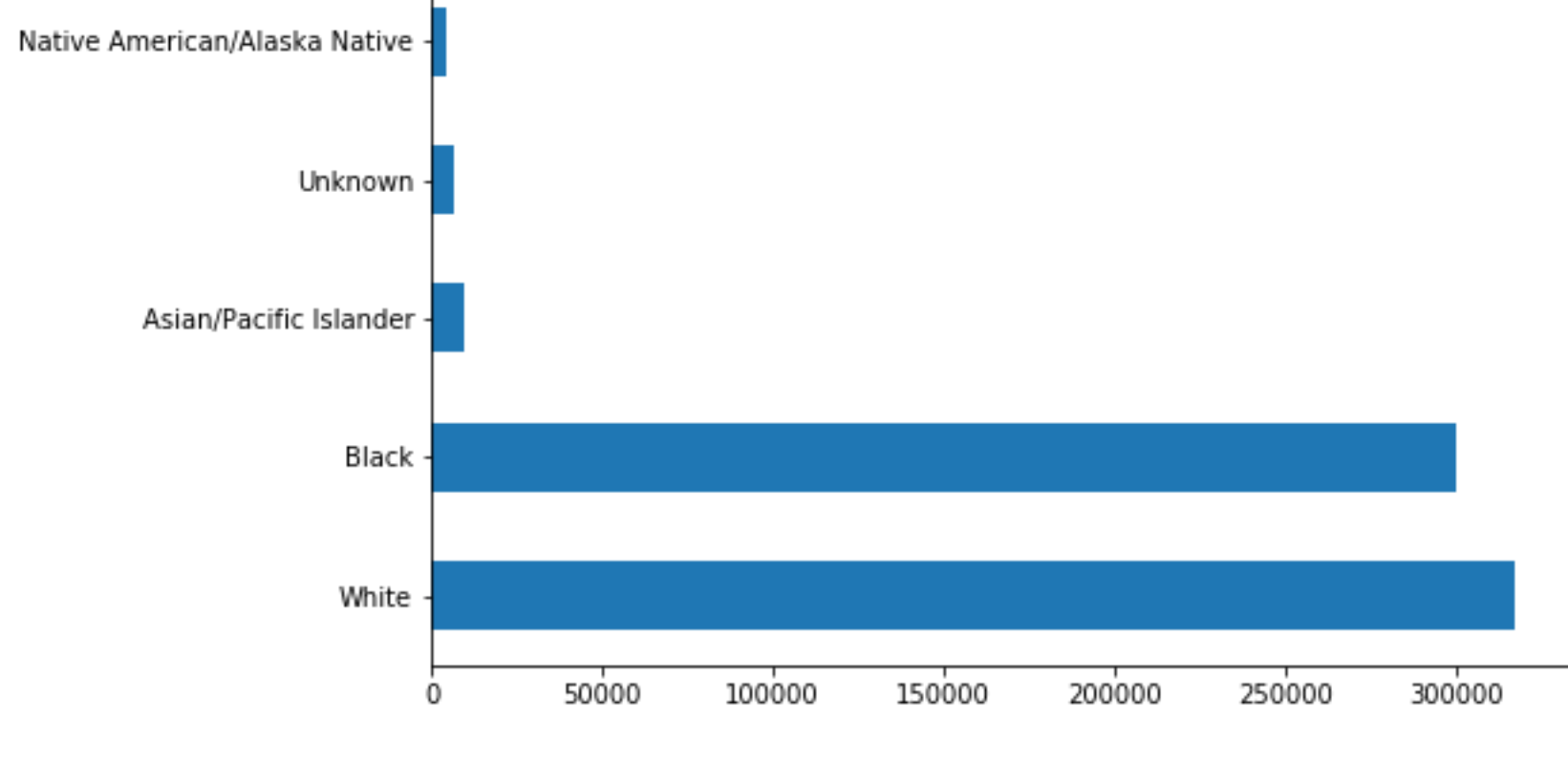
```
In [11]: # gender of the perpetrators
perpetrator_sex = pd.DataFrame(data, columns = ['Perpetrator Sex'])
count_perpetrator_sex = perpetrator_sex.stack().value_counts()
ax = count_perpetrator_sex.plot(kind = 'pie',
                              title = 'Sex of the perpetrators',
                              startangle = 90,
                              autopct='%.4f')
ax.set_ylabel('')
```

```
Out[11]: Text(0, 0.5, '')
```



## Race of Victims

```
In [7]: #Race of Victims
race = pd.DataFrame(data, columns = ['Victim Race'])
count_race = race.stack().value_counts()
count_race.plot(kind='barh', fontsize=10,
               width=0.5,  figsize=(8, 5),
               title='Comparison between Races');
```



## Comparison between victims under 21 and above 21

```
In [8]: ## Comparison between young and adult
data['Victim Age'] = data['Victim Age'].astype("int")
mask = (data['Victim Age'] < 21)
young_victims = pd.DataFrame(data.loc[mask], columns = ['Year'])
count_years = young_victims.stack().value_counts()
homicides_young = count_years.sort_index(axis=0, ascending=False)
mask2 = (data['Victim Age'] > 21)
adult_victims = pd.DataFrame(data.loc[mask2], columns = ['Year'])
count_years = adult_victims.stack().value_counts()
homicides_adult = count_years.sort_index(axis=0, ascending=False)
homicides_adult.to_frame()
homicides_young.to_frame()
homicides = pd.DataFrame({'Adult': homicides_adult, 'Young':homicides_young})
homicides.sort_index(inplace=True)
pos = list(range(len(homicides['Adult'])))
width = 0.3

# Plotting the bars
fig, ax = plt.subplots(figsize=(25,15))

# in position pos,
plt.bar(pos,
        #using homicides['Adult'] data,
        homicides['Adult'],
        # of width
        width,
        # with alpha 0.5
        alpha=0.5,
        # with color
        color='red',
        # with label the first value in year
        label=homicides.index[0])

# Create a bar with young data,
# in position pos + some width buffer,
plt.bar([p + width for p in pos],
        #using homicides['Young'] data,
        homicides['Young'],
        # of width
        width,
        # with alpha 0.5
        alpha=0.5,
        # with color
        color='blue',
        # with label the second value in year
        label=homicides.index[1])

# Set the y axis label
ax.set_ylabel('Adult / Young')

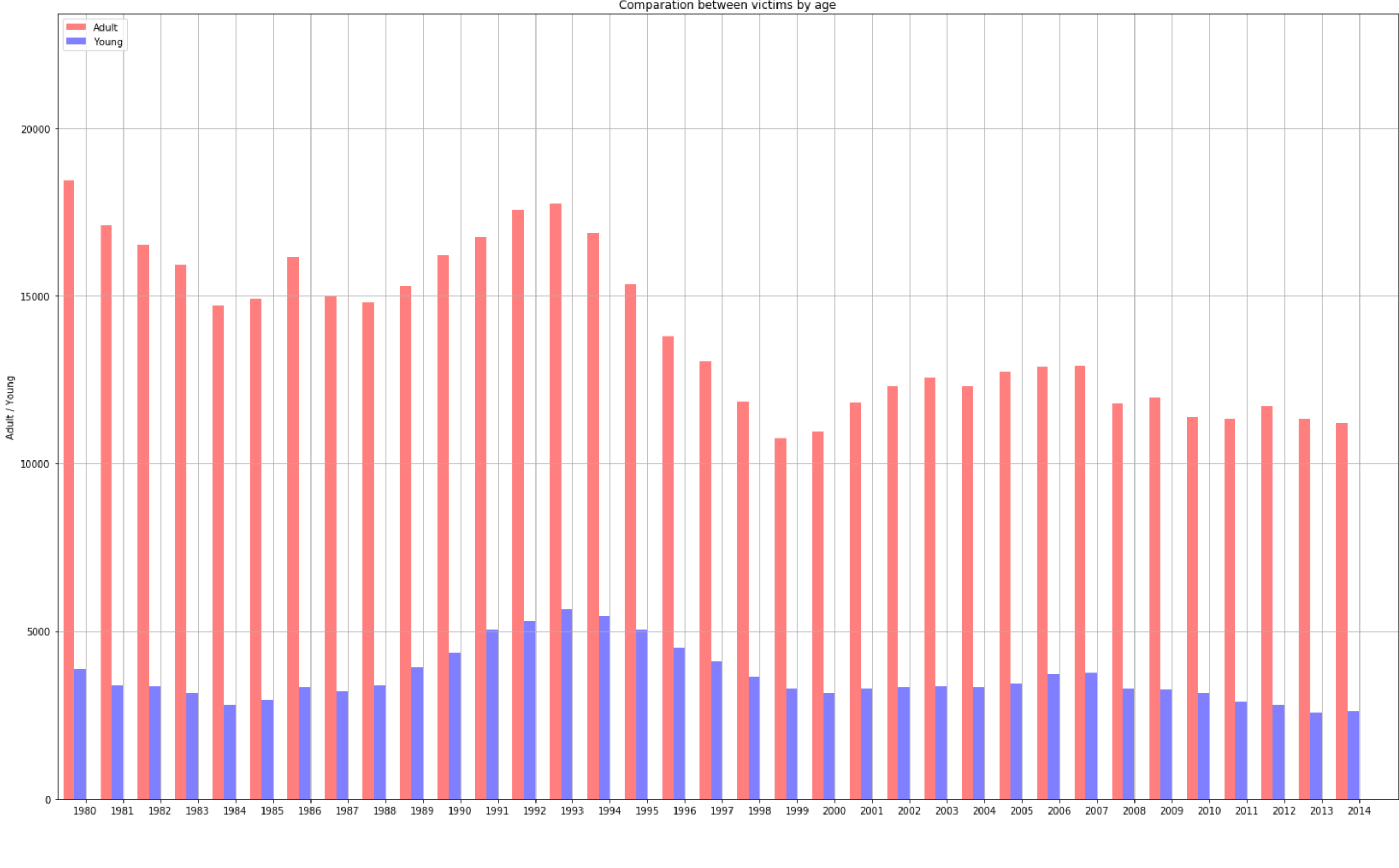
# Set the chart's title
ax.set_title('Comparison between victims by age')

# Set the position of the x ticks
ax.set_xticks([p + 1.5 * width for p in pos])

# Set the labels for the x ticks
ax.set_xticklabels(homicides.index)

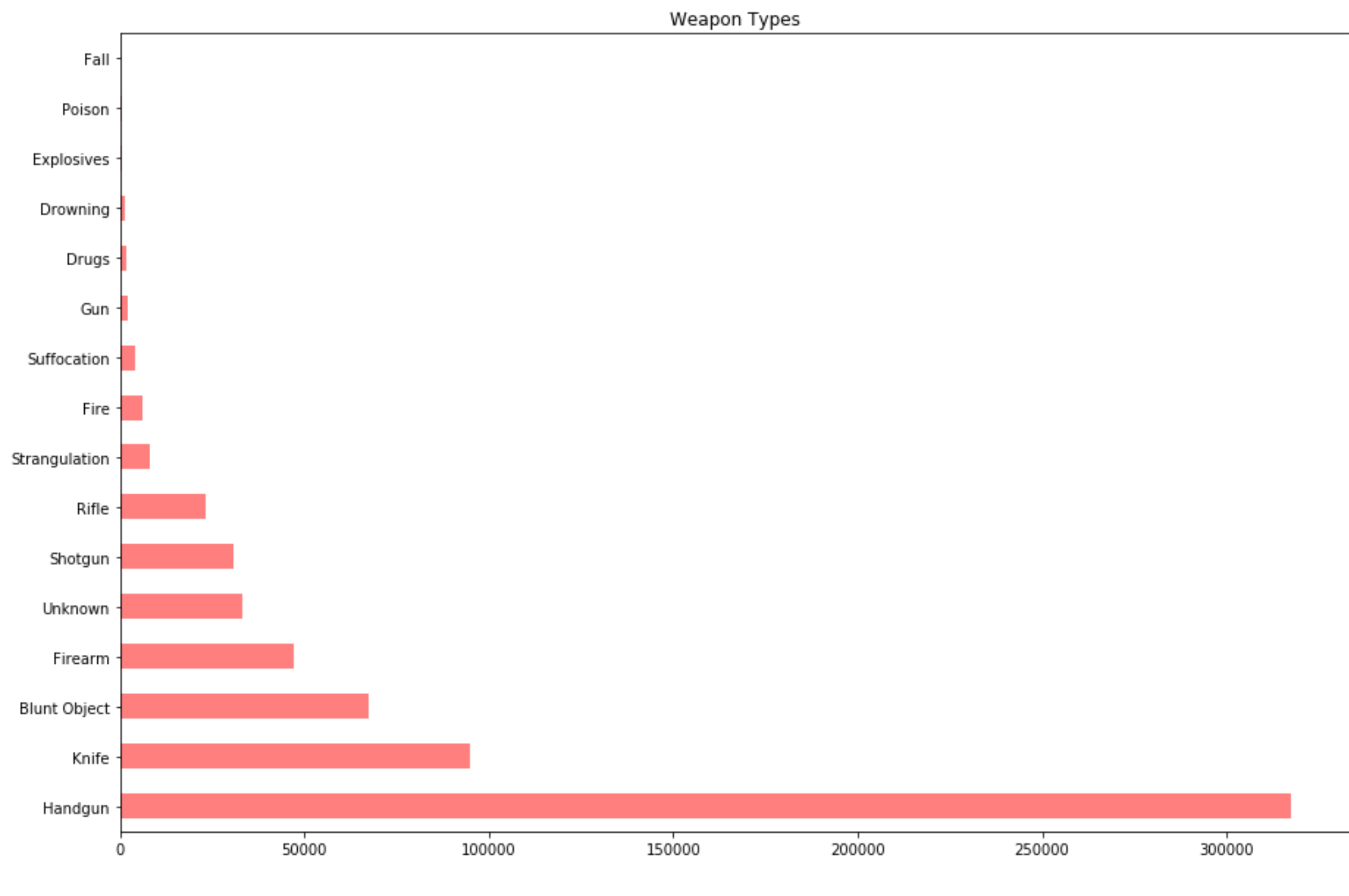
# Setting the x-axis and y-axis limits
plt.xlim(min(pos)-width, max(pos)+width*5)
plt.ylim([0, max(homicides['Adult'] + homicides['Young'])])

# Adding the legend and showing the plot
plt.legend(['Adult', 'Young'], loc='upper left')
plt.grid()
plt.show()
```



## Weapon types

```
In [30]: ## Rate of crime types
weapon_types = pd.DataFrame(data, columns = ['Weapon'])
weapon_count = weapon_types.stack().value_counts()
ax = weapon_count.plot(kind='barh',
                      color='red',alpha=0.5,fontsize=10,
                      width=0.5,  figsize=(15, 10), title='Weapon Types')
```



## Homicides in US by State

```
In [39]: #Crimes by State
state = pd.DataFrame(data, columns = ['State'])
count_states = state.stack().value_counts()
states = count_states.sort_index(axis=0, ascending=False)
#plot the total of homicides
print(states.plot(kind='barh', fontsize=10,
               width=0.5,  figsize=(15, 15),
               title='Homicides in US by State between 1980 and 2014'))
```

AxesSubplot(0.125,0.125;0.775x0.755)

