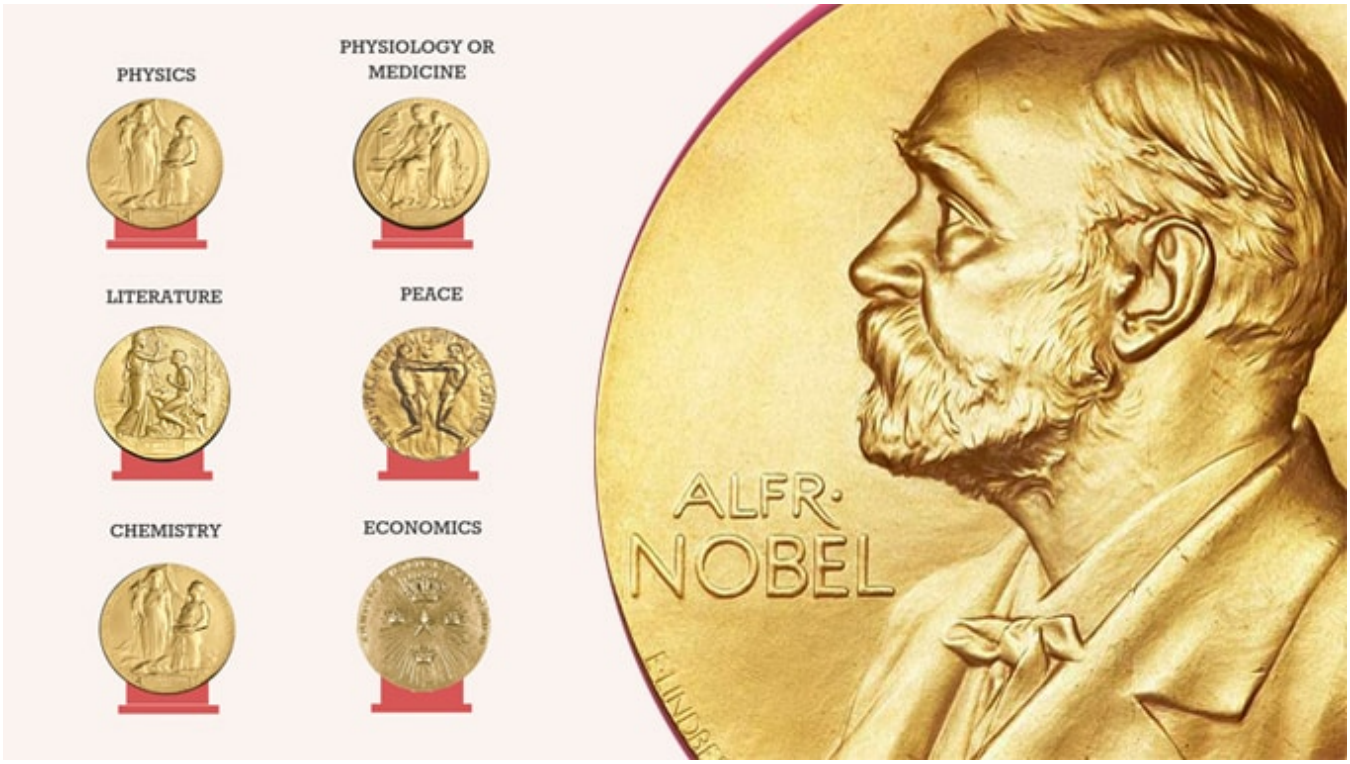

The first Nobel Prizes were awarded in 1901 and they have been awarded annually since then. There have been years in that time when the Nobel Prizes have not been awarded - mostly during World War I (1914-1918) and II (1939-1945). Between 1901 and 2022, the Nobel Prizes and the Prize in Economic Sciences were awarded to 989 Individuals / Organizations. This dataset includes a record for every Individual or Organization that was awarded the Nobel Prize since 1901. This dataset was acquired from the Nobel Prize API. The columns included in the dataset are self-explainatory.

EDA

In [1]:

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

In [2]:

```
df = pd.read_csv('nobel_latest.csv')

```

In [3]:

```
1 df.head()
```

Out[3]:

	Year	Laureate_Id	Firstname	Lastname	Category	Gender	Prize_Share	Motivation	B
0	1901	1	Wilhelm Conrad	Röntgen	physics	male	1	"in recognition of the extraordinary services ...	
1	1901	293	Emil	von Behring	medicine	male	1	"for his work on serum therapy especially its ...	
2	1901	462	Henry	Dunant	peace	male	2	"for his humanitarian efforts to help wounded ...	
3	1901	463	Frédéric	Passy	peace	male	2	"for his lifelong work for international peace...	
4	1901	569	Sully	Prudhomme	literature	male	1	"in special recognition of his poetic composit...	

In [4]:

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 989 entries, 0 to 988
Data columns (total 19 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
 0   Year                  989 non-null    int64 
 1   Laureate_Id           989 non-null    int64 
 2   Firstname             989 non-null    object
 3   Lastname              957 non-null    object
 4   Category              989 non-null    object
 5   Gender                989 non-null    object
 6   Prize_Share           989 non-null    int64 
 7   Motivation            989 non-null    object
 8   Birth_Date            989 non-null    object
 9   Birth_Country         958 non-null    object
 10  Birth_City            956 non-null    object
 11  Birth_Country_Code    958 non-null    object
 12  Death_Date            989 non-null    object
 13  Death_Country         646 non-null    object
 14  Death_City            640 non-null    object
 15  Death_Country_Code    646 non-null    object
 16  Organization_Name     727 non-null    object
 17  Organization_City     722 non-null    object
 18  Organization_Country  724 non-null    object
dtypes: int64(3), object(16)
memory usage: 146.9+ KB
```

In [5]:

```
1 df.shape
```

Out[5]:

```
(989, 19)
```

Deleting Columns which are not required

In [6]:

```
1 df1 = df.drop(['Death_Country','Death_City','Death_Country_Code'],axis=1)
```

In [7]:

```
df1.head(2)
```

Out[7]:

	Year	Laureate_Id	Firstname	Lastname	Category	Gender	Prize_Share	Motivation	Bir
0	1901	1	Wilhelm Conrad	Röntgen	physics	male	1	"in recognition of the extraordinary services ...	
1	1901	293	Emil	von Behring	medicine	male	1	"for his work on serum therapy especially its ...	

Handling Null Values

In [8]:

```
df1.isnull().sum()
```

Out[8]:

```
Year                    0
Laureate_Id             0
Firstname               0
Lastname               32
Category                0
Gender                  0
Prize_Share             0
Motivation              0
Birth_Date              0
Birth_Country          31
Birth_City             33
Birth_Country_Code     31
Death_Date              0
Organization_Name     262
Organization_City     267
Organization_Country  265
dtype: int64
```

In [9]:

```

df1['Lastname'].fillna(' ',inplace=True)
```

In [10]:

```
df1['Birth_Country'].fillna('Not Mentioned',inplace=True)
```

In [11]:

```
df1['Birth_City'].fillna('Not Mentioned',inplace=True)
```

In [12]:

```
df1['Birth_Country_Code'].fillna('Not Mentioned',inplace=True)
```

In [13]:

```
df1['Organization_Name'].fillna('Not Mentioned',inplace=True)
```

In [14]:

```
df1['Organization_City'].fillna('Not Mentioned',inplace=True)
```

In [15]:

```
df1['Organization_Country'].fillna('Not Mentioned',inplace=True)
```

In [16]:

```
df1.isnull().sum()
```

Out[16]:

```
Year                    0
Laureate_Id             0
Firstname               0
Lastname                0
Category                0
Gender                  0
Prize_Share             0
Motivation              0
Birth_Date              0
Birth_Country           0
Birth_City              0
Birth_Country_Code      0
Death_Date              0
Organization_Name       0
Organization_City       0
Organization_Country    0
dtype: int64
```

Concatenating Firstname and Lastname into Name column

In [17]:

```
df1['Name'] = df1['Firstname']+' '+ df1['Lastname']
```

since we no more requires two seprate columns for name so drop firstname and last name

In [18]:

```
df1 = df1.drop(['Firstname','Lastname'],axis=1)
```

Handling and checking other Columns

category column

In [22]:

```
df1['Category'].unique()
```

Out[22]:

```
array(['physics', 'medicine', 'peace', 'literature', 'chemistry',
       'economics'], dtype=object)
```

Gender Column

In [23]:

```
df1['Gender'].value_counts()
```

Out[23]:

```
male      898
female     61
org        30
Name: Gender, dtype: int64
```

Here 'org' stands for Organization

In [24]:

```
1 df1[df1['Gender']=='org']
```

Out[24]:

	Year	Laureate_Id	Category	Gender	Prize_Share	Motivation	Birth_Date	Birth_Country	Birth_Cit
22	1904	467	peace	org	1	"for its striving in public law to develop pea...	1873-00-00	Not Mentioned	Nc Mentione
58	1910	477	peace	org	1	"for acting as a link between the peace societ...	1891-00-00	Not Mentioned	Nc Mentione
87	1917	482	peace	org	1	"for the efforts to take care of wounded soldi...	1863-00-00	Not Mentioned	Nc Mentione

In [228]:

```
1 df1.head(3)
```

Out[228]:

	Year	Laureate_Id	Firstname	Lastname	Category	Gender	Prize_Share	Motivation	Bir
0	1901	1	Wilhelm Conrad	Röntgen	physics	male	1	"in recognition of the extraordinary services ...	
1	1901	293	Emil	von Behring	medicine	male	1	"for his work on serum therapy especially its ...	
2	1901	462	Henry	Dunant	peace	male	2	"for his humanitarian efforts to help wounded ...	

Prize_Share Column

In [25]:

```
1 df1['Prize_Share'].unique()
```

Out[25]:

```
array([1, 2, 4, 3], dtype=int64)
```

Birth_Date Column

In [26]:

```
1 df1['Birth_Date'].unique()
```

Out[26]:

```
array(['1845-03-27', '1854-03-15', '1828-05-08', '1822-05-20',
       '1839-03-16', '1852-08-30', '1852-10-09', '1817-11-30',
       '1843-05-21', '1833-02-19', '1865-05-25', '1857-05-13',
       '1853-07-18', '1859-05-15', '1828-03-18', '1852-12-15',
       '1867-11-07', '1860-12-15', '1832-12-08', '1859-02-19',
       '1832-04-19', '1842-11-12', '1873-00-00', '1852-10-02',
       '1849-09-14', '1830-09-08', '1835-10-31', '1846-05-05',
       '1862-06-07', '1843-06-09', '1843-12-11', '1835-07-27',
       '1858-10-27', '1856-12-18', '1852-05-01', '1852-09-28',
       '1843-07-07', '1845-06-18', '1860-05-20', '1865-12-30',
       '1833-09-20', '1852-12-19', '1871-08-30', '1846-01-05',
       '1837-04-21', '1845-05-15', '1854-03-14', '1844-10-27',
       '1845-08-16', '1874-04-25', '1841-08-25', '1858-11-20',
       '1853-09-02', '1850-06-06', '1829-07-26', '1852-11-22',
       '1830-03-15', '1891-00-00', '1837-11-23', '1853-09-16',
       '1847-03-27', '1864-11-11', '1838-04-28', '1864-01-13',
       '1862-08-29', '1862-06-05', '1862-11-15', '1873-06-28',
       '1854-11-05', '1871-05-06', '1869-11-30', '1845-02-15',
```

In [29]:

```
1 def handleyear(value):
2     date = value.split("-")
3     if len(date[0]) == 2:
4         return int(date[2])
5     elif len(date[0]) == 4:
6         return int(date[0])
```

In [30]:

```
1 df1['Birth_Date'] = df1['Birth_Date'].apply(handleyear)
```

In [31]:

```
df1['Birth_Date'].head()
```

Out[31]:

```
0    1845
1    1854
2    1828
3    1822
4    1839
Name: Birth_Date, dtype: int64
```

Death_Date

using same handleyear function to handle Death_Year

In [32]:

```
def handleyear(value):
    date = value.split("-")
    if len(date[0]) == 2:
        return int(date[2])
    elif len(date[0]) == 4:
        return int(date[0])
```

In [33]:

```
df1['Death_Date'] = df['Death_Date'].apply(handleyear)
```

In [34]:

```
df1['Death_Date']
```

Out[34]:

```
0      1923
1      1917
2      1910
3      1912
4      1907
       ...
984       0
985       0
986       0
987       0
988       0
Name: Death_Date, Length: 989, dtype: int64
```

Birth_Country Column

In [35]:

```
def handlingcountry(string):
    if "(" in string:
        start_index = string.index("(")
        end_index = string.index(")")
        extracted_text = string[start_index + 1 : end_index]
        extracted_text = extracted_text.strip()
        return extracted_text
    elif string =="nan":
        return "not specified"
    else :
        return string
df1["country"] = df1["Birth_Country"].apply(handlingcountry)
df1["country"].unique()
```

Out[35]:

```
array(['now Germany', 'now Poland', 'Switzerland', 'France',
       'the Netherlands', 'India', 'United Kingdom', 'Denmark', 'Norway',
       'Sweden', 'Spain', 'Not Mentioned', 'Scotland', 'Russia', 'Poland',
       'now Slovakia', 'now Czech Republic', 'Germany', 'now Italy',
       'USA', 'Italy', 'now India', 'New Zealand', 'now Ukraine',
       'Luxembourg', 'now Latvia', 'Belgium', 'now Russia', 'Austria',
       'Australia', 'Canada', 'now Slovenia', 'Ireland', 'now Indonesia',
       'now Austria', 'Argentina', 'now Hungary', 'now Croatia',
       'now Finland', 'Chile', 'Portugal', 'Japan', 'South Africa',
       'now France', 'Iceland', 'China', 'now Algeria', 'Brazil',
       'Guadeloupe Island', 'now Zimbabwe', 'Hungary',
       'now Bosnia and Herzegovina', 'now Azerbaijan', 'now Turkey',
       'Egypt', 'Guatemala', 'now Belarus', 'Vietnam', 'Romania',
       'Northern Ireland', 'now Lithuania', 'now Saint Lucia',
       'now North Macedonia', 'now Greece', 'now Pakistan', 'Venezuela',
       'Bulgaria', 'Colombia', 'Lithuania', 'Mexico', 'Madagascar',
       'Taiwan', 'Nigeria', 'now South Korea', 'Costa Rica', 'now China',
       'now Myanmar', 'Saint Lucia', 'now Israel', 'East Timor',
       'now Ghana', 'Trinidad and Tobago', 'Iran', 'Kenya',
       'now Bangladesh', 'Turkey', 'now Iran', 'Finland', 'Peru',
       'Cyprus', 'Yemen', 'Liberia', 'Morocco', 'Pakistan', 'Ukraine',
       'Iraq', 'now Democratic Republic of the Congo', 'Ethiopia',
       'Philippines', 'Lebanon'], dtype=object)
```

In [36]:

```
def handling_now(value):
    if "now" in value:
        return value.replace("now","")
    else :
        return  value
df1["country"] = df1["country"].apply(handling_now)
df1["country"].unique()
```

Out[36]:

```
array([' Germany', ' Poland', 'Switzerland', 'France', 'the Netherlands',
       'India', 'United Kingdom', 'Denmark', 'Norway', 'Sweden', 'Spain',
       'Not Mentioned', 'Scotland', 'Russia', 'Poland', ' Slovakia',
       ' Czech Republic', 'Germany', ' Italy', 'USA', 'Italy', ' India',
       'New Zealand', ' Ukraine', 'Luxembourg', ' Latvia', 'Belgium',
       ' Russia', 'Austria', 'Australia', 'Canada', ' Slovenia',
       'Ireland', ' Indonesia', ' Austria', 'Argentina', ' Hungary',
       ' Croatia', ' Finland', 'Chile', 'Portugal', 'Japan',
       'South Africa', ' France', 'Iceland', 'China', ' Algeria',
       'Brazil', 'Guadeloupe Island', ' Zimbabwe', 'Hungary',
       ' Bosnia and Herzegovina', ' Azerbaijan', ' Turkey', 'Egypt',
       'Guatemala', ' Belarus', 'Vietnam', 'Romania', 'Northern Ireland',
       ' Lithuania', ' Saint Lucia', ' North Macedonia', ' Greece',
       ' Pakistan', 'Venezuela', 'Bulgaria', 'Colombia', 'Lithuania',
       'Mexico', 'Madagascar', 'Taiwan', 'Nigeria', ' South Korea',
       'Costa Rica', ' China', ' Myanmar', 'Saint Lucia', ' Israel',
       'East Timor', ' Ghana', 'Trinidad and Tobago', 'Iran', 'Kenya',
       ' Bangladesh', 'Turkey', ' Iran', 'Finland', 'Peru', 'Cyprus',
       'Yemen', 'Liberia', 'Morocco', 'Pakistan', 'Ukraine', 'Iraq',
       ' Democratic Republic of the Congo', 'Ethiopia', 'Philippines',
       'Lebanon'], dtype=object)
```

In [37]:

```
def handling_space(value):
    if " " in value:
        return value.replace(" ","")
    else :
        return  value
df1["country"] = df1["country"].apply(handling_space)
df1["country"].unique()
```

Out[37]:

```
array(['Germany', 'Poland', 'Switzerland', 'France', 'theNetherlands',
       'India', 'UnitedKingdom', 'Denmark', 'Norway', 'Sweden', 'Spain',
       'NotMentioned', 'Scotland', 'Russia', 'Slovakia', 'CzechRepublic',
       'Italy', 'USA', 'NewZealand', 'Ukraine', 'Luxembourg', 'Latvia',
       'Belgium', 'Austria', 'Australia', 'Canada', 'Slovenia', 'Ireland',
       'Indonesia', 'Argentina', 'Hungary', 'Croatia', 'Finland', 'Chile',
       'Portugal', 'Japan', 'SouthAfrica', 'Iceland', 'China', 'Algeria',
       'Brazil', 'GuadeloupeIsland', 'Zimbabwe', 'BosniaandHerzegovina',
       'Azerbaijan', 'Turkey', 'Egypt', 'Guatemala', 'Belarus', 'Vietnam',
       'Romania', 'NorthernIreland', 'Lithuania', 'SaintLucia',
       'NorthMacedonia', 'Greece', 'Pakistan', 'Venezuela', 'Bulgaria',
       'Colombia', 'Mexico', 'Madagascar', 'Taiwan', 'Nigeria',
       'SouthKorea', 'CostaRica', 'Myanmar', 'Israel', 'EastTimor',
       'Ghana', 'TrinidadandTobago', 'Iran', 'Kenya', 'Bangladesh',
       'Peru', 'Cyprus', 'Yemen', 'Liberia', 'Morocco', 'Iraq',
       'DemocraticRepublicoftheCongo', 'Ethiopia', 'Philippines',
       'Lebanon'], dtype=object)
```

In [39]:

```
df1.drop(["Birth_Country"],axis = 1 , inplace = True)
```

Birth_country_code

In [41]:

```
df1["Birth_Country_Code"].fillna("Not specified",inplace = True)
```

Organization_Name Column

In [43]:

```
1 df1.isnull().sum()
```

Out[43]:

```
Year                   0
Laureate_Id            0
Category               0
Gender                 0
Prize_Share            0
Motivation             0
Birth_Date             0
Birth_City             0
Birth_Country_Code     0
Death_Date             0
Organization_Name      0
Organization_City      0
Organization_Country   0
Name                   0
country                0
dtype: int64
```

Creating New Age column to calculate at what age person won the award

In [44]:

```
1 df1['Win_year'] = df1['Year']-df1['Birth_Date']
```

Handling win_year column

In [45]:

```
1 df1['Win_year'].unique()
```

Out[45]:

```
array([ 56,  47,  73,  79,  62,  49,  50,  85,  59,  69,  37,  45,  44,
        75,  51,  36,  43,  71,  72,  31,  52,  55,  74,  70,  48,  54,
        63,  42,  64,  35,  68,  80,  57,  19,  39,  58,  41,  67,  60,
        38,  46,  53,  25,  40,  61,  77,  32,  86,  65,  66,  17,  81,
        78, 300,  30,   4,  33,  34, 100,  87,  76,  16,  84,  82,  83,
         5,  28,  88,  90,  89,   2,  10,  96,  97,  15], dtype=int64)
```

In [46]:

```
1 print(df1['Win_year'].max())
2 print(df1['Win_year'].min())
```

```
300
2
```

In [47]:

```
def handling_win_year(value):
    if value<17:
        return np.nan
    elif value>96:
        return np.nan
    else:
        return value
```

In [48]:

```
df1['Win_year'] = df1['Win_year'].apply(handling_win_year)
df1['Win_year'].unique()
```

Out[48]:

```
array([56., 47., 73., 79., 62., 49., 50., 85., 59., 69., 37., 45., 44.,
       75., 51., 36., 43., 71., 72., 31., 52., 55., 74., 70., 48., 54.,
       63., 42., 64., 35., 68., 80., 57., 19., 39., 58., 41., 67., 60.,
       38., 46., 53., 25., 40., 61., 77., 32., 86., 65., 66., 17., 81.,
       78., nan, 30., 33., 34., 87., 76., 84., 82., 83., 28., 88., 90.,
       89., 96.])
```

Laureate_type column

In [50]:

```
def handlinglaureate(gender):
    if gender == 'org':
        return 'organization'
    else:
        return 'Individual'


```

In [51]:

```
df1['Laureate_type'] = df1['Gender'].apply(handlinglaureate)
df1['Laureate_type'].unique()
```

Out[51]:

```
array(['Individual', 'organization'], dtype=object)
```

In [52]:

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 989 entries, 0 to 988
Data columns (total 17 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Year                  989 non-null    int64
 1   Laureate_Id           989 non-null    int64
 2   Category              989 non-null    object
 3   Gender                989 non-null    object
 4   Prize_Share           989 non-null    int64
 5   Motivation            989 non-null    object
 6   Birth_Date            989 non-null    int64
 7   Birth_City            989 non-null    object
 8   Birth_Country_Code    989 non-null    object
 9   Death_Date            989 non-null    int64
 10  Organization_Name     989 non-null    object
 11  Organization_City     989 non-null    object
 12  Organization_Country  989 non-null    object
 13  Name                  989 non-null    object
 14  country               989 non-null    object
 15  Win_year              978 non-null    float64
 16  Laureate_type         989 non-null    object
dtypes: float64(1), int64(5), object(11)
memory usage: 131.5+ KB
```

Re-arranging columns

In [53]:

```
columns = ['Year','Category','Laureate_Id','Laureate_type','Name','Gender','Prize_Sh
           'Win_year','Birth_City','Birth_Country_Code','Death_Date','Organization_N
           'Organization_City','Organization_Country','Motivation']
df1= df1[columns]
```

Converting Dataframe into CSV

In [54]:

```
df1.to_csv('Nobel_prize.csv')

```

In [ ]:

```

```