Nobel Prize Winners Analysis with Python

The Nobel Prize is an international award administered by the Nobel Foundation in Stockholm, Sweden, and is based on the fortune of Alfred Nobel, a Swedish inventor, engineer, and industrialist. In 1968, Sveriges Riksbank established The Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel, founder of the Nobel Prize.

Between 1901 and 2023, the Nobel Prizes and the Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel were awarded 621 times to 1,000 people and organisations. With some receiving the Nobel Prize more than once, this makes a total of 965 individuals and 27 organisations.

Import Library

In [2]:
```
import pandas as pd
```

In [3]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import seaborn as sns
```

```
C:\Users\Syed Arif\anaconda3\lib\site-packages\scipy\__init__.py:146: UserWar
ning: A NumPy version >=1.16.5 and <1.23.0 is required for this version of Sc
iPy (detected version 1.25.1
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

Uploading Csv fle

In [4]:
```
df = pd.read_csv(r"C:\Users\Syed Arif\Desktop\Nobel_Prize.csv")
```

Data Preprocessing

.head()

head is used show to the By default = 5 rows in the dataset

In [5]:
```
df.head()
```

Out[5]:

	year	category	motivation	prizeShare	laureateID	fullName	gender	born	bornCountr
0	2023	medicine	for their discoveries concerning nucleoside ba...	2	1024	Katalin Kariko	female	17-01-1955	Hungar
1	2023	economics	for having advanced our understanding of women...	1	1034	Claudia Goldin	female	1946-00-00	US/
2	2023	peace	for her fight against the oppression of women ...	1	1033	Narges Mohammadi	female	21-04-1972	Ira
3	2023	literature	for his innovative plays and prose which give ...	1	1032	Jon Fosse	male	29-09-1959	Norwa
4	2023	chemistry	for the discovery and synthesis of quantum dots	3	1031	Alexei Ekimov	male	1945-00-00	Russi

.tail()

tail is used to show rows by Descending order

In [6]:
```
df.tail()
```

Out[6]:

eShare	laureateID	fullName	gender	born	bornCountry	bornCity	died	diedCountry	die
2	463	Frederic Passy	male	1822-05-20	France	Paris	6/12/1912	France	
2	462	Henry Dunant	male	1828-05-08	Switzerland	Geneva	10/30/1910	Switzerland	H
1	293	Emil von Behring	male	1854-03-15	Poland	Lawice	3/31/1917	Germany	Ma
1	160	Jacobus H. van 't Hoff	male	1852-08-30	the Netherlands	Rotterdam	3/1/1911	Germany	
1	1	Wilhelm Conrad Rontgen	male	1845-03-27	Germany	Remscheid	2/10/1923	Germany	M

.shape

It show the total no of rows & Column in the dataset

In [7]:
```
df.shape
```

Out[7]:
```
(1000, 16)
```

.Columns

It show the no of each Column

In [8]:
```
df.columns
```

Out[8]:
```
Index(['year', 'category', 'motivation', 'prizeShare', 'laureateID',
       'fullName', 'gender', 'born', 'bornCountry', 'bornCity', 'died',
       'diedCountry', 'diedCity', 'organizationName', 'organizationCountry',
       'organizationCity'],
      dtype='object')
```

.dtypes

This Attribute show the data type of each column

In [9]:
```
df.dtypes
```

Out[9]:
```
year                   int64
category              object
motivation            object
prizeShare             int64
laureateID             int64
fullName              object
gender                object
born                  object
bornCountry           object
bornCity              object
died                  object
diedCountry           object
diedCity              object
organizationName      object
organizationCountry   object
organizationCity      object
dtype: object
```

.unique()

In a column, It show the unique value of specific column.

In [10]:
```
df["category"].unique()
```

Out[10]:
```
array(['medicine', 'economics', 'peace', 'literature', 'chemistry',
       'physics'], dtype=object)
```

.nuique()

It will show the total no of unque value from whole data frame

In [11]:
```
df.nunique()
```

Out[11]:
```
year                  120
category                6
motivation            690
prizeShare              4
laureateID            992
fullName              992
gender                  3
born                  968
bornCountry            88
bornCity              646
died                  653
diedCountry            48
diedCity              319
organizationName      322
organizationCountry    27
organizationCity      191
dtype: int64
```

.describe()

It show the Count, mean , median etc

In [12]:
```
df.describe()
```

Out[12]:

	year	prizeShare	laureateID
count	1000.000000	1000.000000	1000.000000
mean	1973.721000	2.027000	509.099000
std	34.523195	0.944014	298.130617
min	1901.000000	1.000000	1.000000
25%	1949.750000	1.000000	250.750000
50%	1979.000000	2.000000	500.500000
75%	2003.000000	3.000000	764.250000
max	2023.000000	4.000000	1034.000000

.value_counts

It Shows all the unique values with their count

In [13]:
```
df["category"].value_counts()
```

Out[13]:
```
medicine      227
physics       225
chemistry     194
peace         141
literature    120
economics      93
Name: category, dtype: int64
```

.isnull()

It shows the how many null values

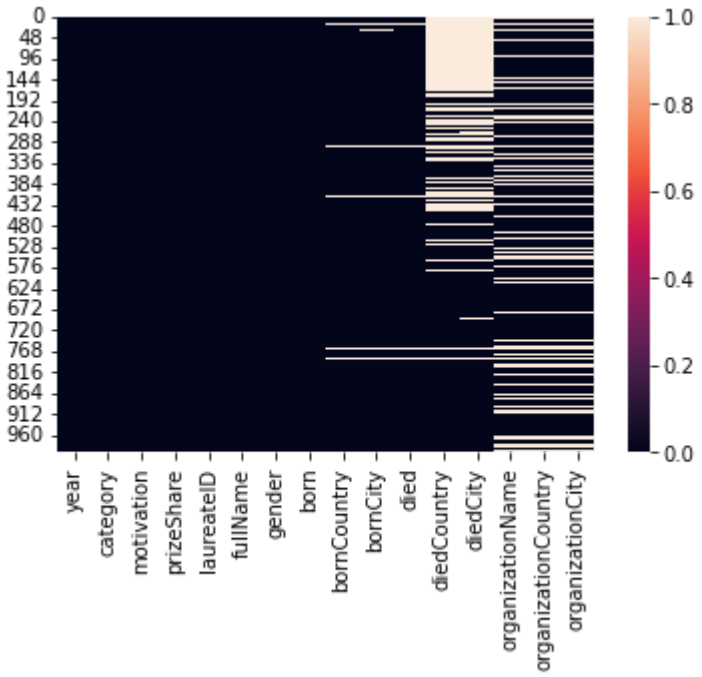
In [14]:
```
df.isnull()
```

Out[14]:

›n	prizeShare	laureateID	fullName	gender	born	bornCountry	bornCity	died	diedCountry	die
se	False	False	False	False	False	False	False	False	True	
se	False	False	False	False	False	False	False	False	True	
se	False	False	False	False	False	False	False	False	True	
se	False	False	False	False	False	False	False	False	True	
se	False	False	False	False	False	False	True	False	True	
...	...	...	...	...	...	...	...	...	...	
se	False	False	False	False	False	False	False	False	False	
se	False	False	False	False	False	False	False	False	False	
se	False	False	False	False	False	False	False	False	False	
se	False	False	False	False	False	False	False	False	False	
se	False	False	False	False	False	False	False	False	False	

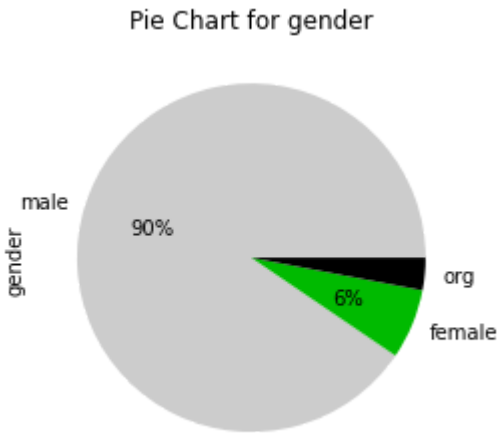
In [15]:
```
sns.heatmap(df.isnull())
```

Out[15]: <AxesSubplot:>

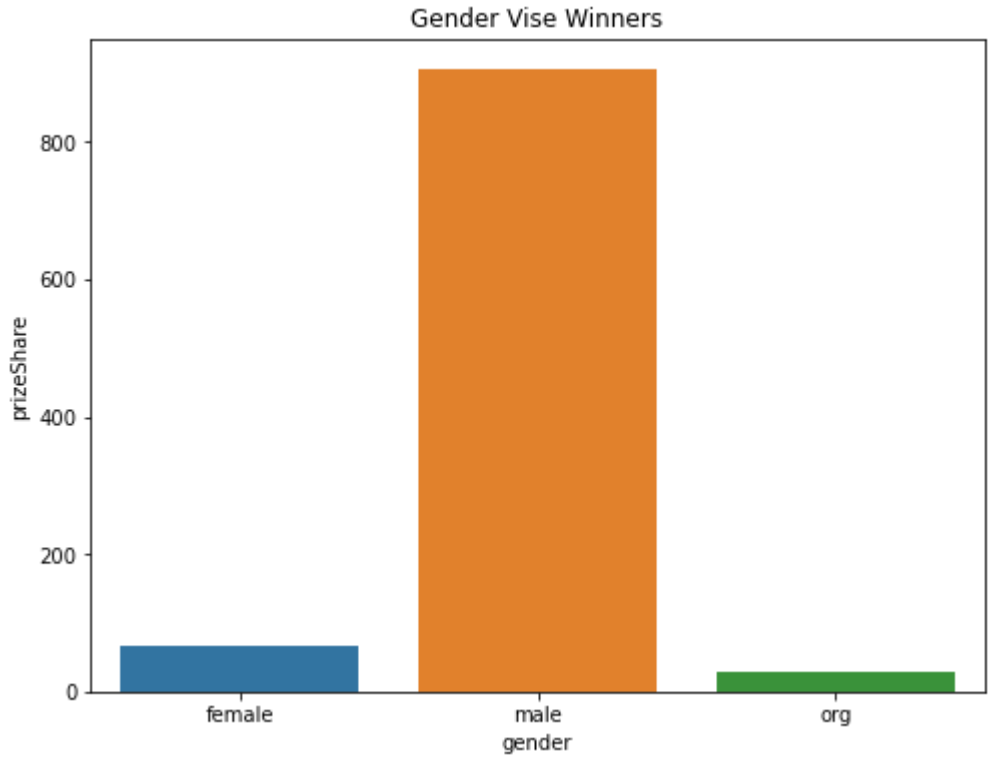


In [18]:
```
df['gender'].value_counts().plot(kind = 'pie' , title = 'Pie Chart for gender',
                                 autopct="%.0f%%", colormap='nipy_spectral_r')
```

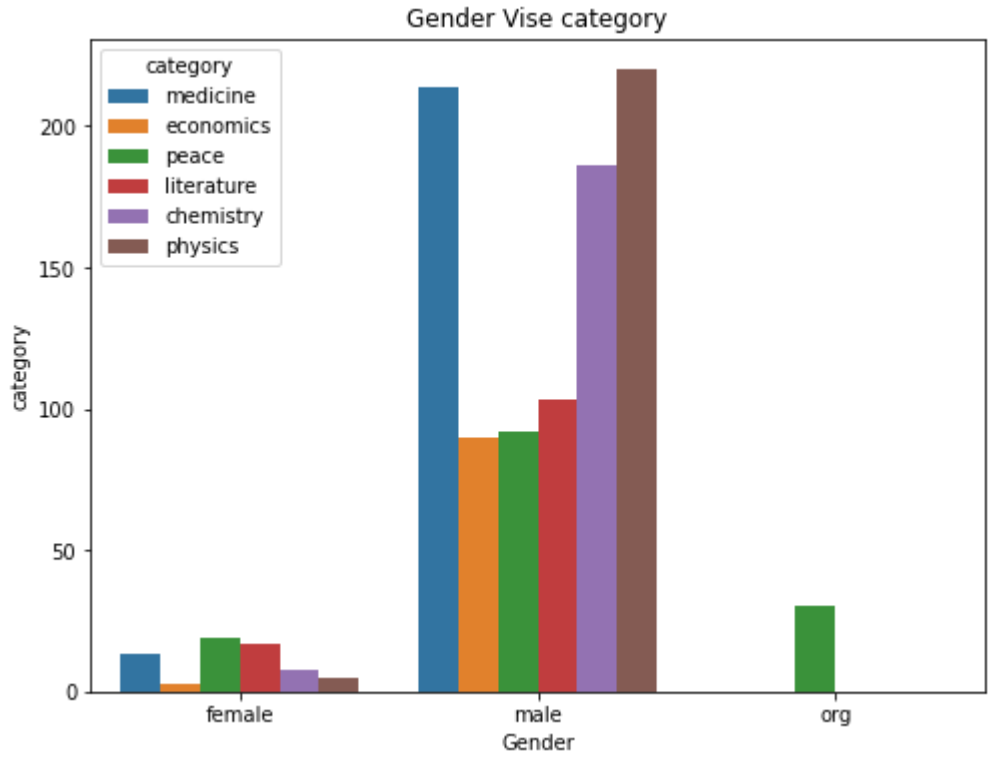
Out[18]: <AxesSubplot:title={'center':'Pie Chart for gender'}, ylabel='gender'>



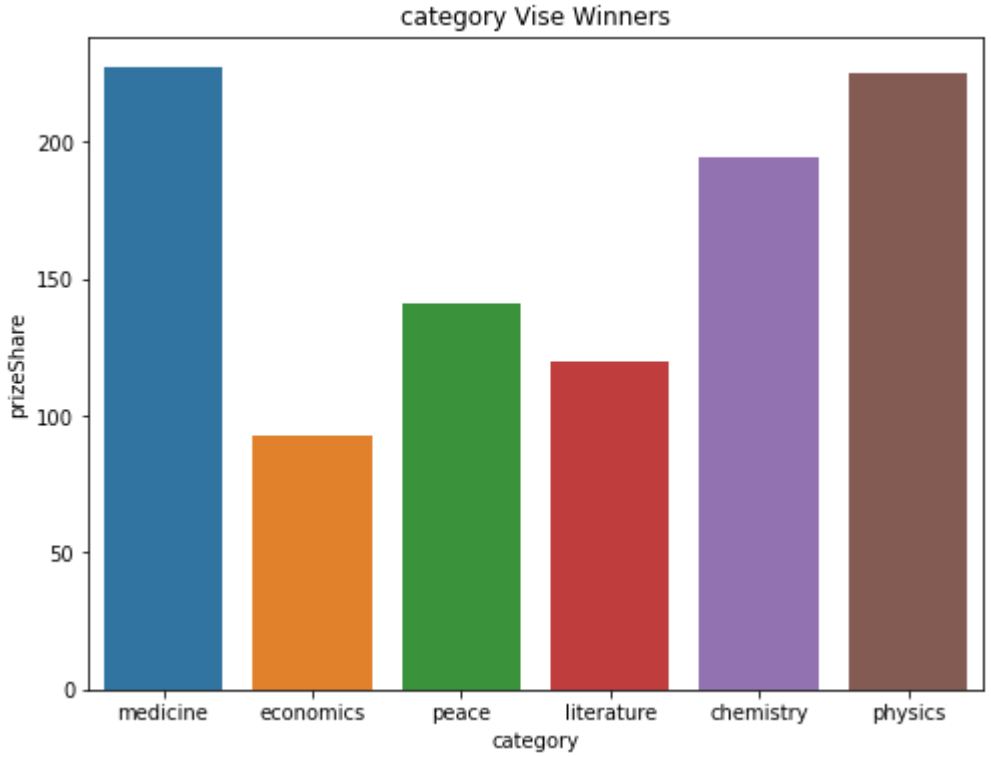
In [19]:
```
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='gender')
plt.xlabel('gender')
plt.ylabel('prizeShare')
plt.title('Gender Vise Winners')
plt.show()
```



In [20]:
```
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='gender', hue ="category")
plt.xlabel('Gender')
plt.ylabel('category')
plt.title('Gender Vise category')
plt.show()
```



In [23]:
```
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='category')
plt.xlabel('category')
plt.ylabel('prizeShare')
plt.title('category Vise Winners')
plt.show()
```



In [32]:
```
# Create a line plot to visualize the relationship between age and heart rate
plt.figure(figsize=(8, 6))
sns.lineplot(data=df, x='category', y='year')
plt.title('Year By Category Win')
plt.xlabel('category')
plt.ylabel('year')
plt.tight_layout()
plt.show()
```