Name:Sanika A Jadhav.

TrackCode:DS

Task1:Create a bar chart to visualize the distribution of a categorical or continous variable

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
%cd /content/drive/MyDrive/datasets_for_coding/

```

```
/content/drive/MyDrive/datasets_for_coding
```

```
import pandas as pd
import seaborn as sns
import numpy as np
```

```
df = pd.read_csv('Summer-Olympic-medals-1976-to-2008.csv',sep=',',  encoding='latin-1')
df.head()
```

	City	Year	Sport	Discipline	Event	Athlete	Gender	Country_Code	Country	Event_gender	Medal
0	Montreal	1976.0	Aquatics	Diving	3m springboard	KÖHLER, Christa	Women	GDR	East Germany	W	Silver
1	Montreal	1976.0	Aquatics	Diving	3m springboard	KOSENKOV, Aleksandr	Men	URS	Soviet Union	M	Bronze
2	Montreal	1976.0	Aquatics	Diving	3m springboard	BOGGS, Philip George	Men	USA	United States	M	Gold
3	Montreal	1976.0	Aquatics	Diving	3m springboard	CAGNOTTO, Giorgio Franco	Men	ITA	Italy	M	Silver
4	Montreal	1976.0	Aquatics	Diving	10m platform	WILSON, Deborah Keplar	Women	USA	United States	W	Bronze

```
df.shape
```

```
(15433, 11)
```

```
df.isnull().sum()
```

```
City            117
Year            117
Sport           117
Discipline      117
Event           117
Athlete         117
Gender          117
Country_Code    117
Country         117
Event_gender    117
Medal           117
dtype: int64
```

```
df.dropna(inplace=True)
```

```
%matplotlib inline
from matplotlib import pyplot as plt
```

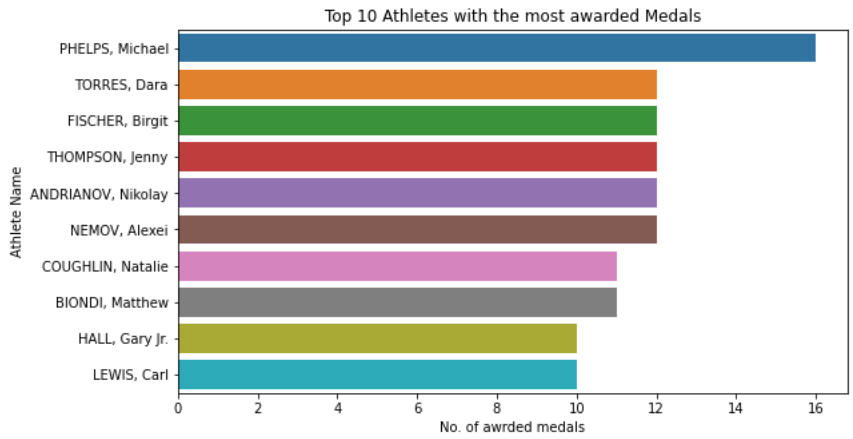
```
plt.figure(figsize=(10, 5))
sns.countplot(df['Year'])
plt.title('Total Athletes contribution in summer olympics over time')
plt.xlabel('Years')
plt.ylabel('No. of Athlete')

```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning:
```

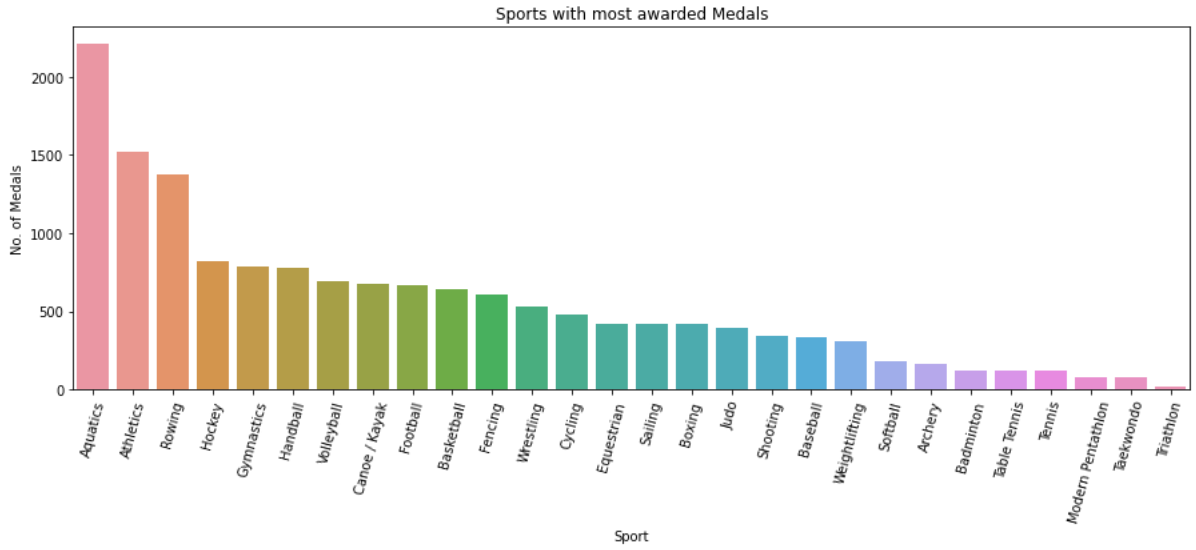
```
Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be  data , and passing other
```

```
athlete_order = df['Athlete'].value_counts().head(10).index
plt.figure(figsize=(9, 5))
sns.countplot(data=df, y='Athlete', order=athlete_order)
plt.title('Top 10 Athletes with the most awarded Medals')
plt.xlabel('No. of awrded medals')
plt.ylabel('Athlete Name');
```




```
plt.figure(figsize=(15, 5))
highest_sport = df['Sport'].value_counts().index
sns.countplot(data=df, x='Sport', order=highest_sport)
plt.xticks(rotation=75)
plt.title('Sports with most awarded Medals')
plt.xlabel('Sport')
plt.ylabel('No. of Medals')
```

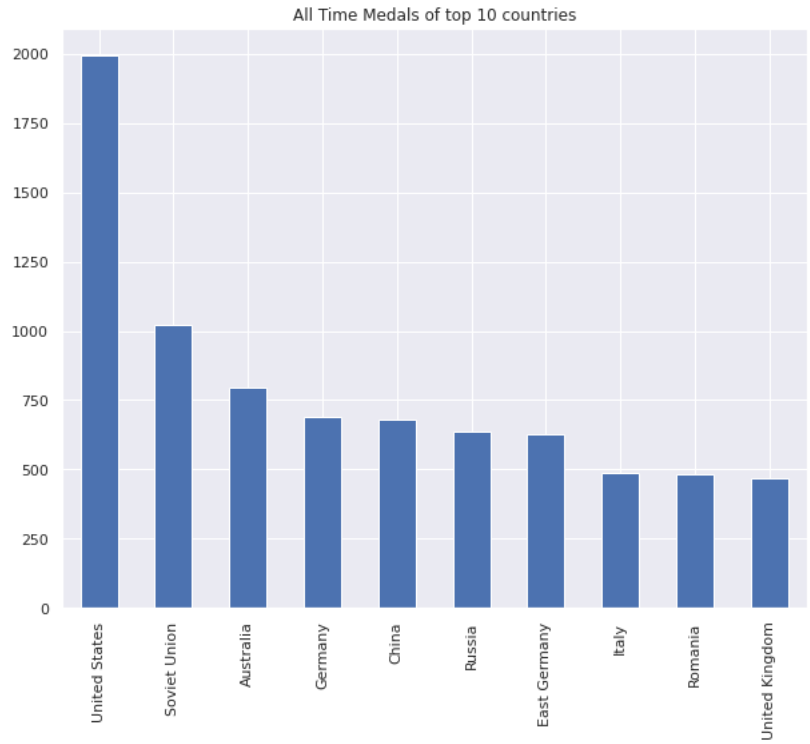
```
Text(0, 0.5, 'No. of Medals')
```




```
Start coding or generate with AI.
```

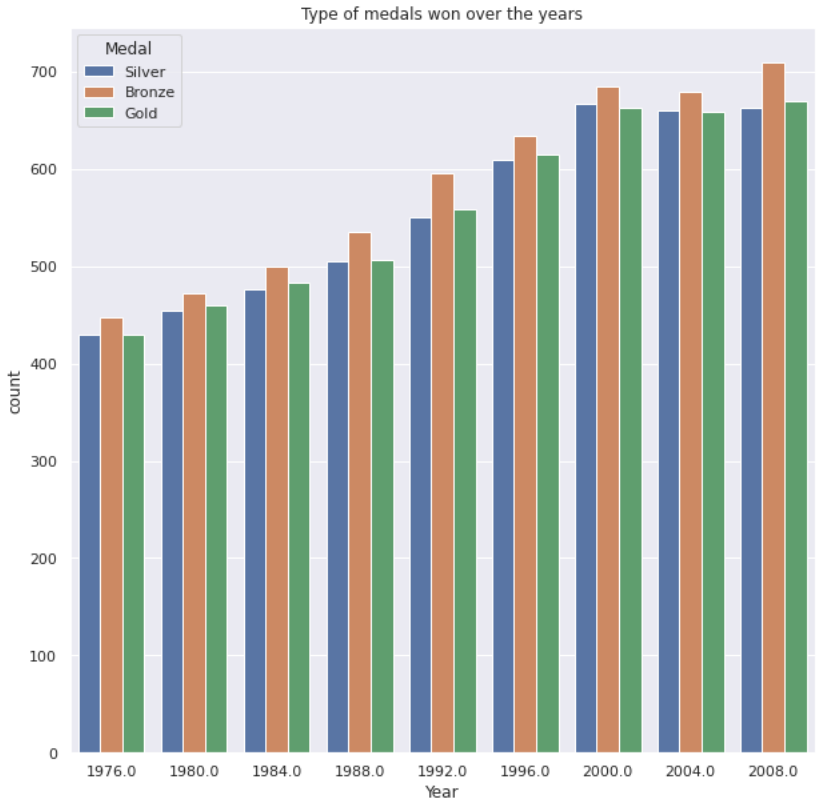
```
top_10 = df['Country'].value_counts()[:10]
top_10.plot(kind='bar',figsize=(10,8))
plt.title('All Time Medals of top 10 countries')
```

```
Text(0.5, 1.0, 'All Time Medals of top 10 countries')
```



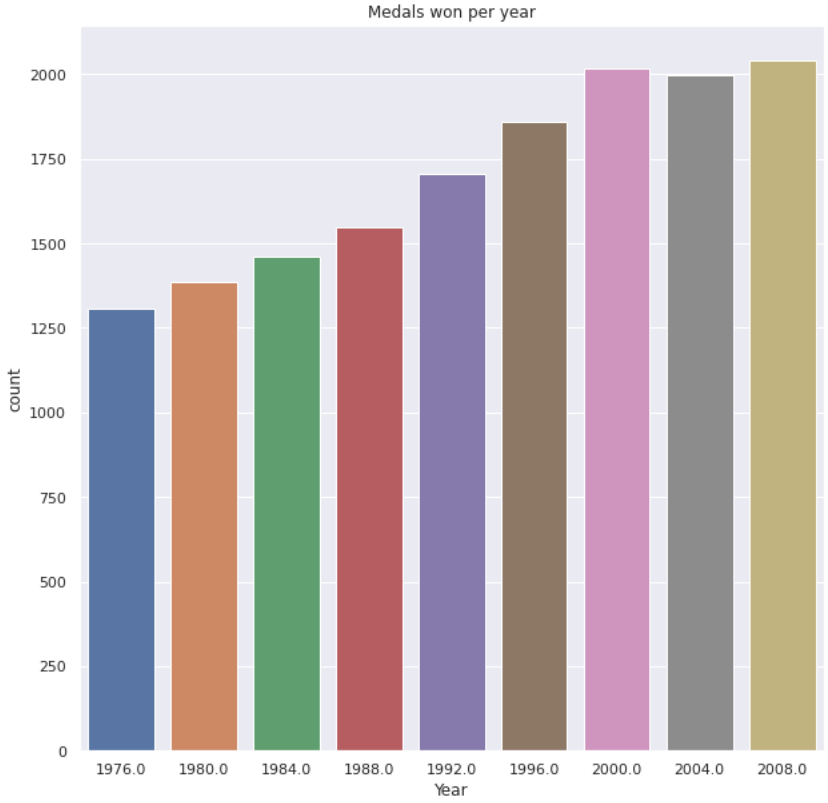

```
1 sns.countplot(x='Year',hue='Medal',data=df)
2 sns.set(rc={'figure.figsize':(10,10)})
3 plt.title("Type of medals won over the years")
```

```
Text(0.5, 1.0, 'Type of medals won over the years')
```



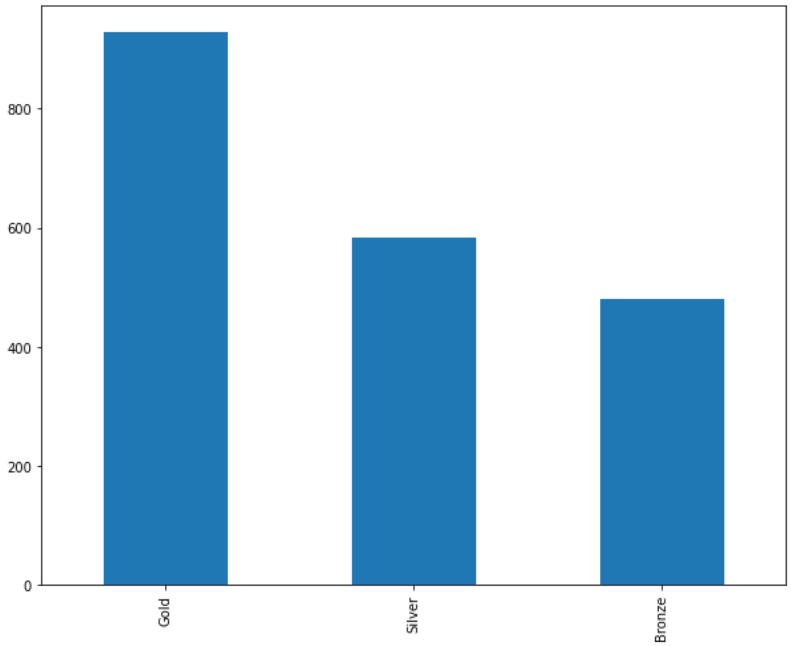

```
1 sns.countplot(x='Year',data=df)
2 sns.set(rc={'figure.figsize':(10,10)}).plot(kind='bar',figsize=(10,8))
3
4 plt.title("Medals won per year")
```

```
Text(0.5, 1.0, 'Medals won per year')
```



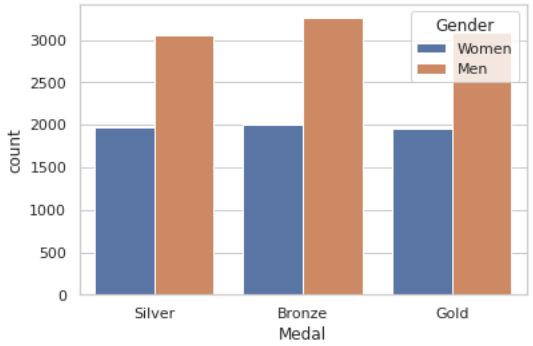

```
1 indpie = df[df['Country']=='United States']['Medal'].value_counts()
2 indpie.plot(kind='bar',figsize=(10,8))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7faacbf0c3d0>
```



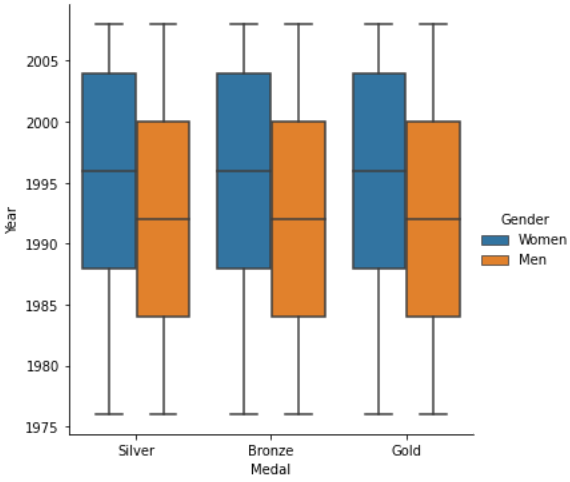

```
1  sns.countplot(x="Medal", hue="Gender", data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f49829be490>
```



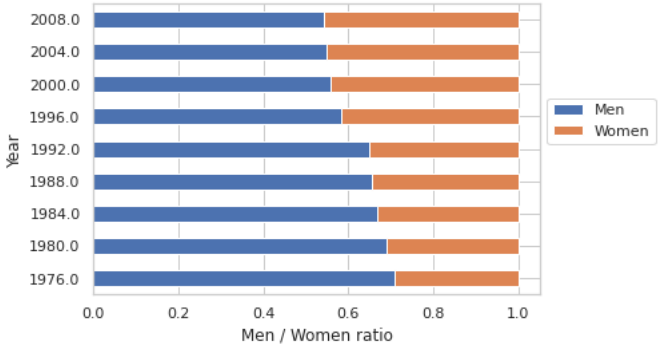

```
1 sns.catplot(x="Medal", y="Year", hue="Gender",kind="box", data=df)
```

```
<seaborn.axisgrid.FacetGrid at 0x7faac0798550>
```



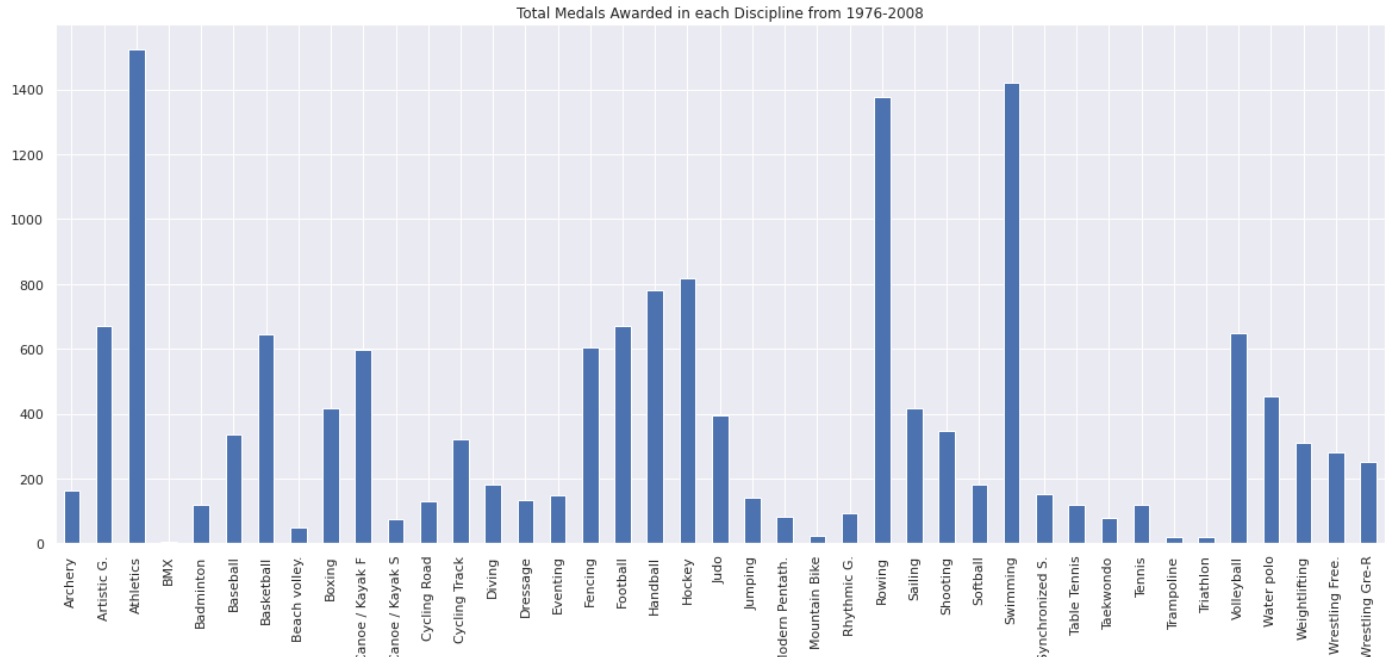

```
1 gender_group = df.groupby(['Year', 'Gender']).size().unstack()
2 gender_group.apply(lambda x:x/x.sum(), axis=1).plot(kind='barh', stacked=True, legend=False)
3 plt.legend(['Men', 'Women'], bbox_to_anchor=(1.0, 0.7))
4 plt.xlabel('Men / Women ratio')
```

```
Text(0.5, 0, 'Men / Women ratio')
```



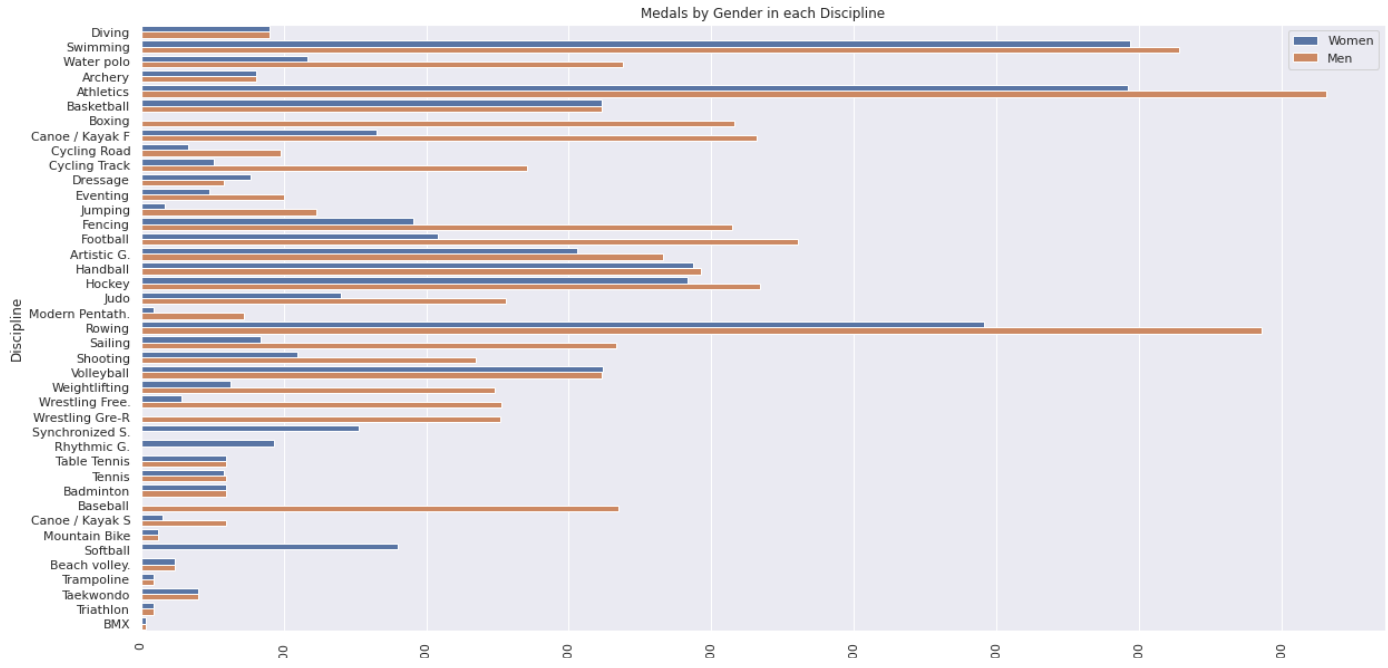

```
1 p = df.groupby('Discipline').agg('count')
2 p.plot(y='Medal',kind='bar',legend=False,title='Total Medals Awarded in each Discipline from 1976-2008',figsize=(20,8))
3
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f497ed8d350>
```



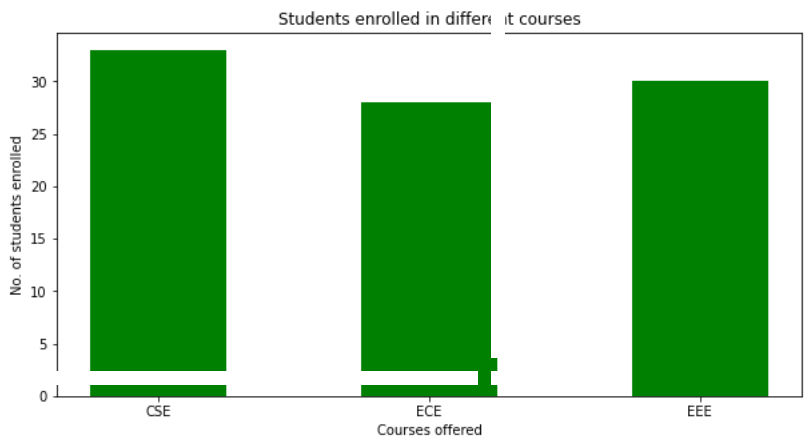

```
1 sns.countplot(y='Discipline',hue='Gender',data=df)
2 sns.set(rc={'figure.figsize':(10,10)})
3 plt.xticks(rotation=90)
4 plt.title('Medals by Gender in each Discipline')
5 plt.legend(loc=1)  # 1 is code for 'upper right'
```

```
<matplotlib.legend.Legend at 0x7f49665a4190>
```

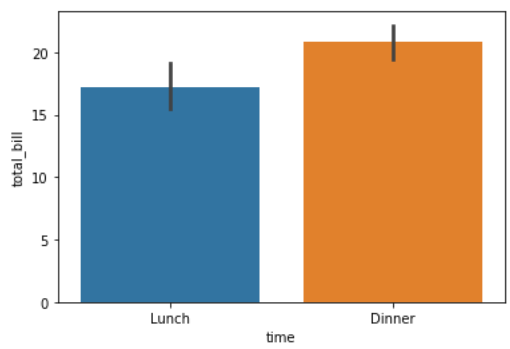



```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4
5 # Dataset generation
6 data_dict = {'CSE':33, 'ECE':28, 'EEE':30}
7 courses = list(data_dict.keys())
8 values = list(data_dict.values())
9
10 fig = plt.figure(figsize = (10, 5))
11
12 #  Bar plot
13 plt.bar(courses, values, color ='green',
14         width = 0.5)
15
```

```
plt.xlabel("Courses offered")
plt.ylabel("No. of students enrolled")
plt.title("Students enrolled in different courses"
plt.show()
```




```
import seaborn as sns
import matplotlib.pyplot as plt
df = sns.load_dataset('tips')
sns.barplot(x = 'time',
            y = 'total_bill',
            data = df)
plt.show()
```

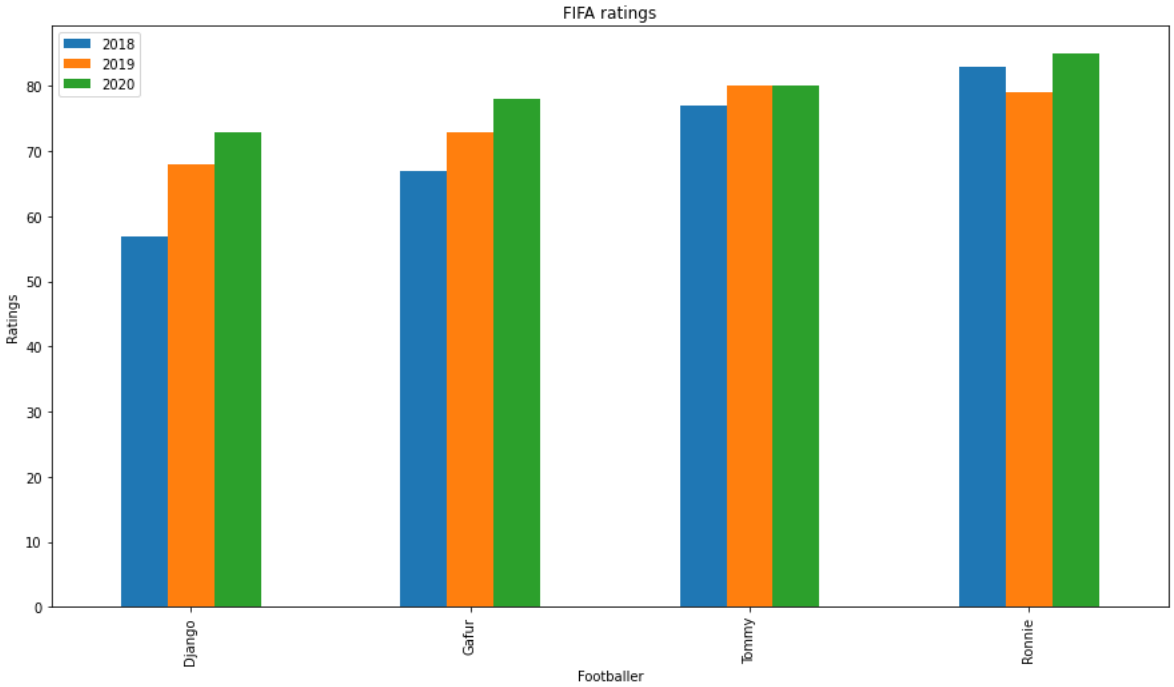



```
import plotly.express as px
data_canada = px.data.gapminder().query("country == 'Canada'")
fig = px.bar(data_canada, x='year', y='pop')
fig.show()
```

```
import pandas as pd
plotdata = pd.DataFrame({
    "2018":[57,67,77,83],
    "2019":[68,73,80,79],
    "2020":[73,78,80,85]},
    index=["Django", "Gafur", "Tommy", "Ronnie"])
plotdata.plot(kind="bar",figsize=(15, 8))
plt.title("FIFA ratings")
plt.xlabel("Footballer")
plt.ylabel("Ratings")

```

```
Text(0, 0.5, 'Ratings')
```




```
import pandas as pd
plotdata = pd.DataFrame({
    "2018":[57,67,77,83],
    "2019":[68,73,80,79],
    "2020":[73,78,80,85]},
    index=["Django", "Gafur", "Tommy", "Ronnie"])
plotdata.plot(kind='bar', stacked=True,figsize=(15, 8))
plt.title("FIFA ratings")
plt.xlabel("Footballer")
plt.ylabel("Ratings")
```

```
Text(0, 0.5, 'Ratings')
```