Matplotlib

Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy.

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

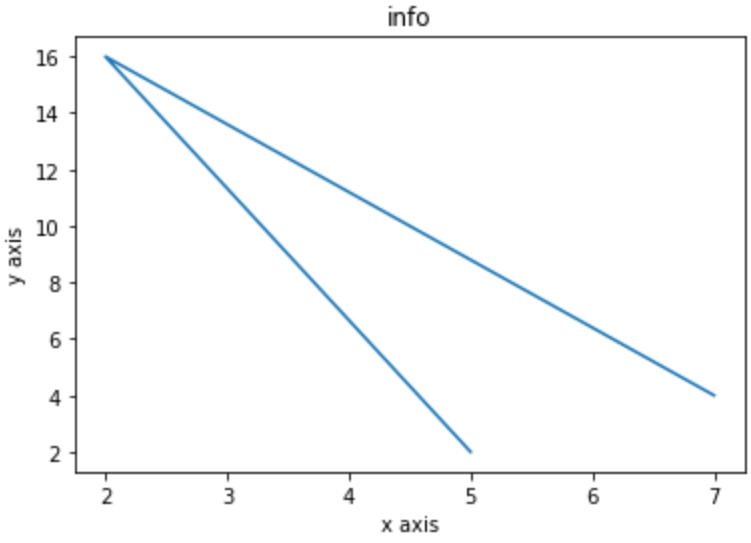
Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays

In [1]:
```
pip install matplotlib
```

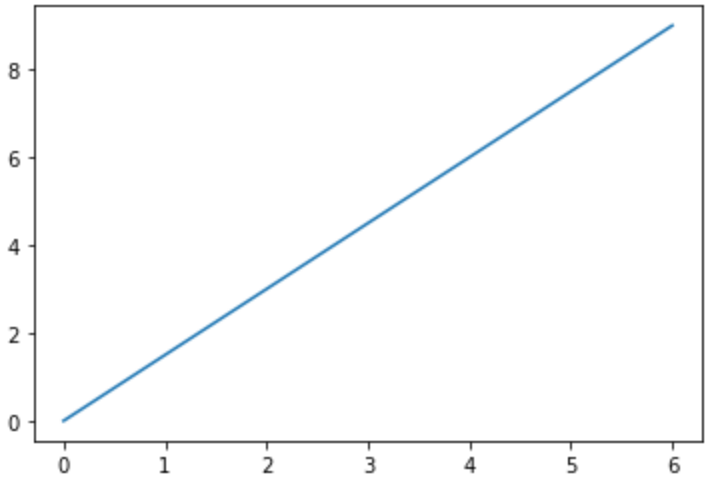
```
Requirement already satisfied: matplotlib in d:\archana\lib\site-packages (3.4.3)
Requirement already satisfied: cycler>=0.10 in d:\archana\lib\site-packages (from matplotl
ib) (0.10.0)
Requirement already satisfied: python-dateutil>=2.7 in d:\archana\lib\site-packages (from
matplotlib) (2.8.2)
Requirement already satisfied: pyparsing>=2.2.1 in d:\archana\lib\site-packages (from matp
lotlib) (3.0.4)
Requirement already satisfied: numpy>=1.16 in d:\archana\lib\site-packages (from matplotli
b) (1.20.3)
Requirement already satisfied: pillow>=6.2.0 in d:\archana\lib\site-packages (from matplot
lib) (8.4.0)
Requirement already satisfied: kiwisolver>=1.0.1 in d:\archana\lib\site-packages (from mat
plotlib) (1.3.1)
Requirement already satisfied: six in d:\archana\lib\site-packages (from cycler>=0.10->mat
plotlib) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [27]:
```
from matplotlib import pyplot as plt
import numpy as np
```

In [3]:
```
x=[5,2,7]
y=[2,16,4]
plt.plot(x,y)
plt.title('info')
plt.ylabel('y axis')
plt.xlabel('x axis')
plt.show()
```



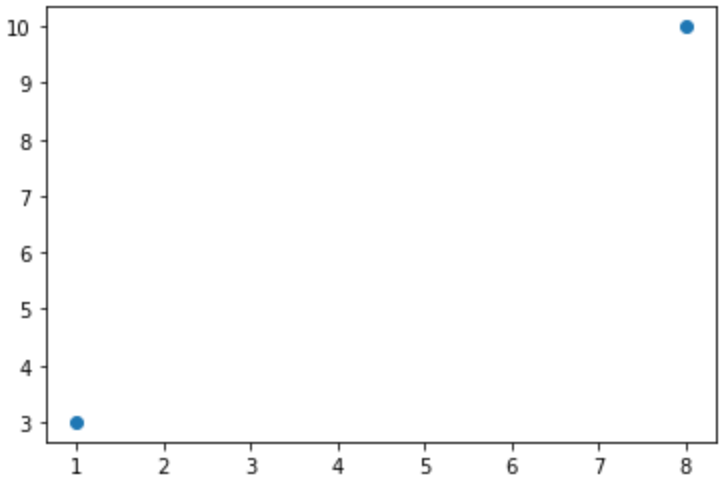
In [5]:
```
xpoints=np.array([0,6])
ypoints=np.array([0,9])
plt.plot(xpoints,ypoints)
plt.show()
```



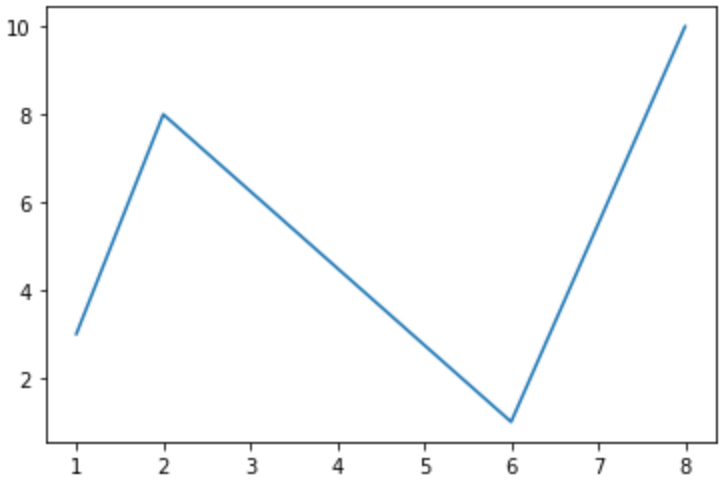
Plotting Without Line

To plot only the markers, you can use shortcut string notation parameter 'o', which means 'rings'.

In [6]:
```
xpoints=np.array([1,8])
ypoints=np.array([3,10])
plt.plot(xpoints,ypoints,'o')
plt.show()
```

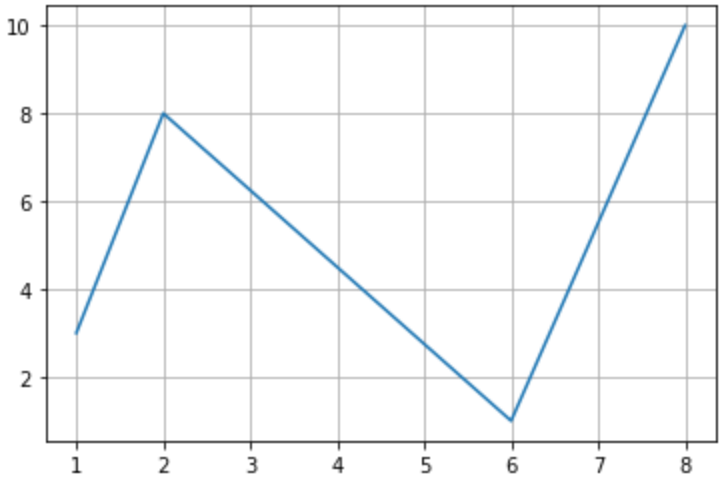


In [2]:
```
#multiple points
xpoints = np.array([1, 2, 6, 8])
ypoints = np.array([3, 8, 1, 10])
plt.plot(xpoints, ypoints)
plt.show()
```



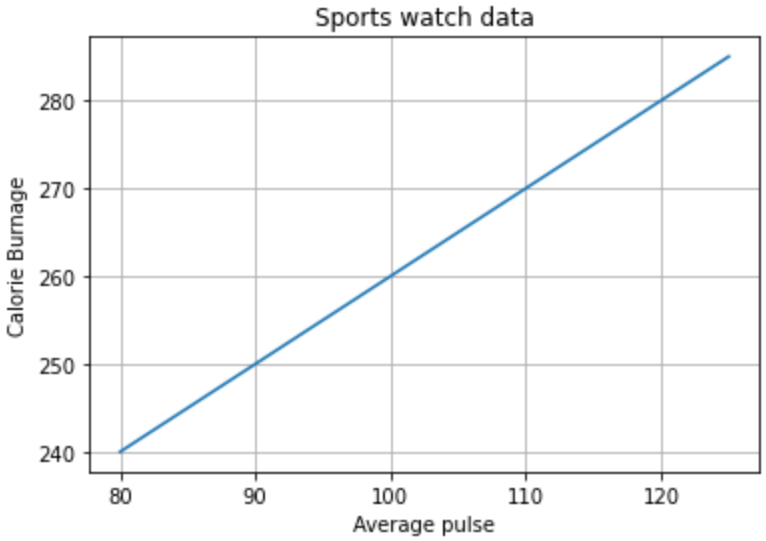
In [3]:

```
#With Pyplot, you can use the grid() function to add grid lines to the plot.
xpoints = np.array([1, 2, 6, 8])
ypoints = np.array([3, 8, 1, 10])
plt.plot(xpoints, ypoints)
plt.grid()
plt.show()
```



In [7]:

```
x=np.array([80,85,90,95,100,105,110,115,120,125])
y=np.array([240,245,250,255,260,265,270,275,280,285])
plt.title("Sports watch data")
plt.xlabel("Average pulse")
plt.ylabel("Calorie Burnage")
plt.plot(x,y)
plt.grid()
plt.show()
```



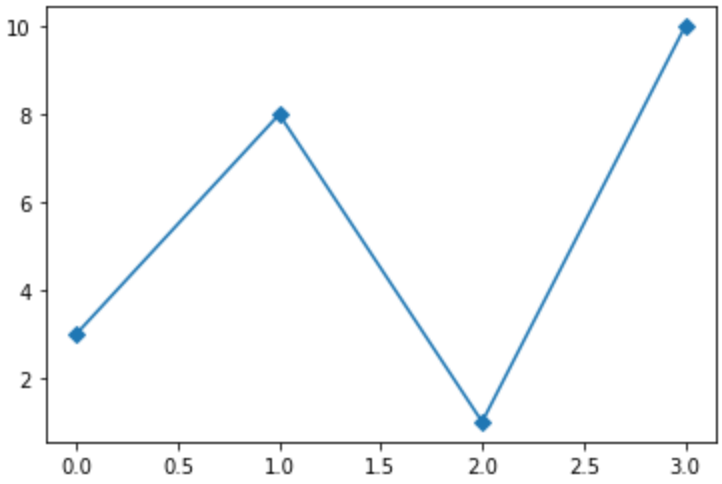
Markers

You can use the keyword argument marker to emphasize each point with a specified marker

o' Circle
'*' Star
'.' Point
',' Pixel
'x' X
'X' X (filled)
'+' Plus
'P' Plus (filled)
's' Square
'D' Diamond 'd' Diamond (thin)
'p' Pentagon
'H' Hexagon 'h' Hexagon 'v' Triangle Down
'^' Triangle Up '<' Triangle Left
'>' Triangle Right
'1' Tri Down
'2' Tri Up
'3' Tri Left
'4' Tri Right
'|' Vline
'_' Hline

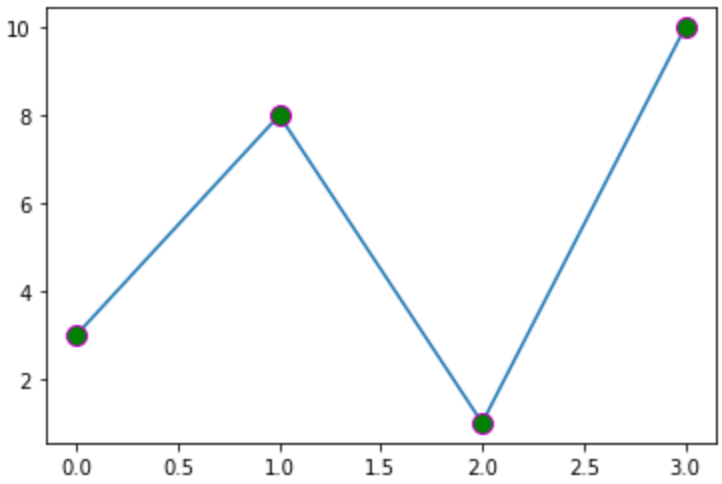
In [8]:

```
ypoints=np.array([3,8,1,10])
plt.plot(ypoints,marker='D')
plt.show()
```



In [9]:
```
#ms--marker size
#mec--marker edge color
#mfc--marker face color
plt.plot(ypoints,marker='o',ms=10,mec='m',mfc='g')
```

Out[9]:
```
[<matplotlib.lines.Line2D at 0x14ac29babb0>]
```



color syntax Description

'r' Red

'g' Green

'c' Cyan

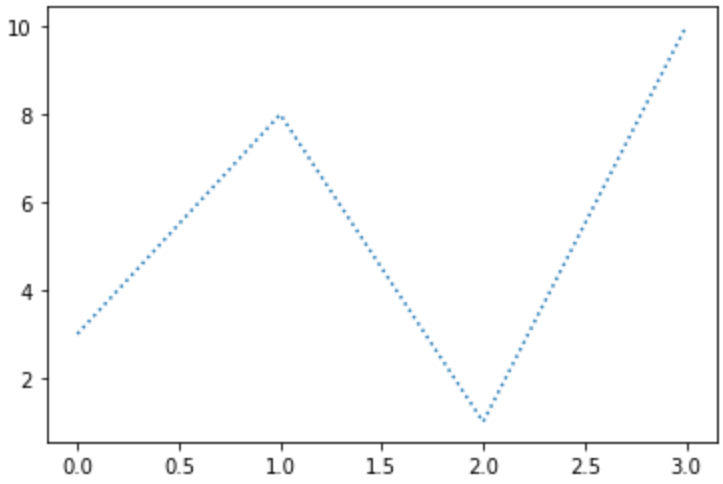
'y' Yellow

'k' Black

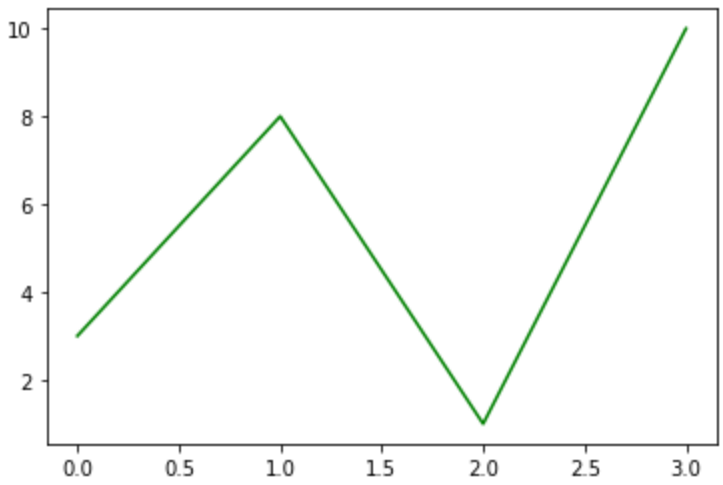
'w' White

line plot

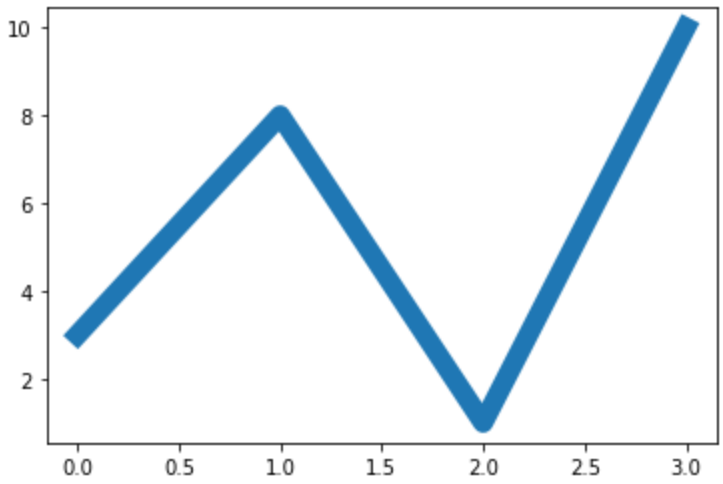
In [10]:
```
ypoints=np.array([3,8,1,10])
plt.plot(ypoints,linestyle='dotted')
plt.show()
```



In [11]:
```
plt.plot(ypoints,color='g')
plt.show()
```



In [12]:
```
plt.plot(ypoints,linewidth='10')
plt.show()
```

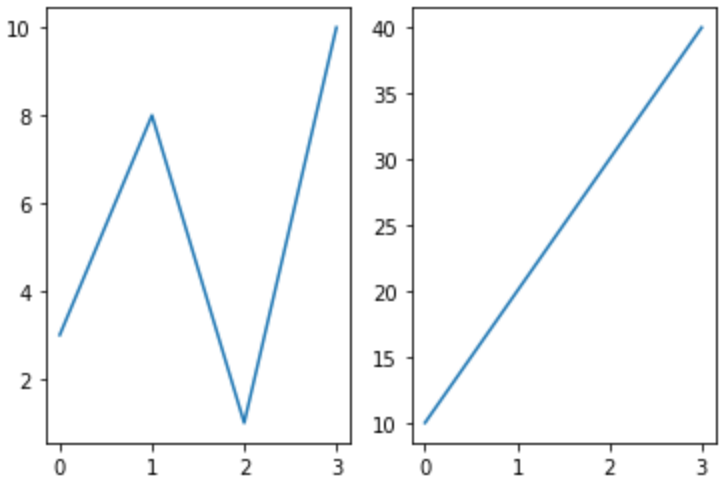


subplot

With the subplot() function you can draw multiple plots in one figure

In [13]:
```
#plot 1:
x = np.array([0, 1, 2, 3])
```

```
y = np.array([3, 8, 1, 10])
plt.subplot(1, 2, 1)
plt.plot(x,y)
#plot 2:
x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])
plt.subplot(1, 2, 2)
plt.plot(x,y)
plt.show()
```

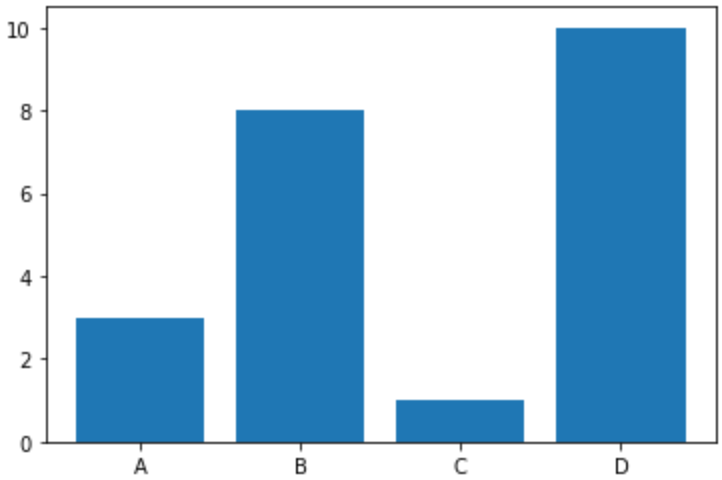


Barplot

With Pyplot, you can use the bar() function to draw bar graphs

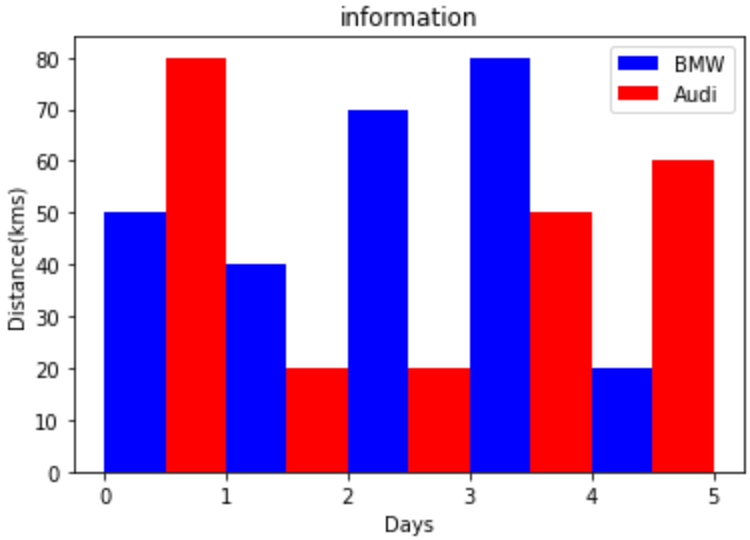
In [14]:

```
x = np.array(["A", "B", "C", "D"])
y = np.array([3, 8, 1, 10])
plt.bar(x,y)
plt.show()
```



In [7]:

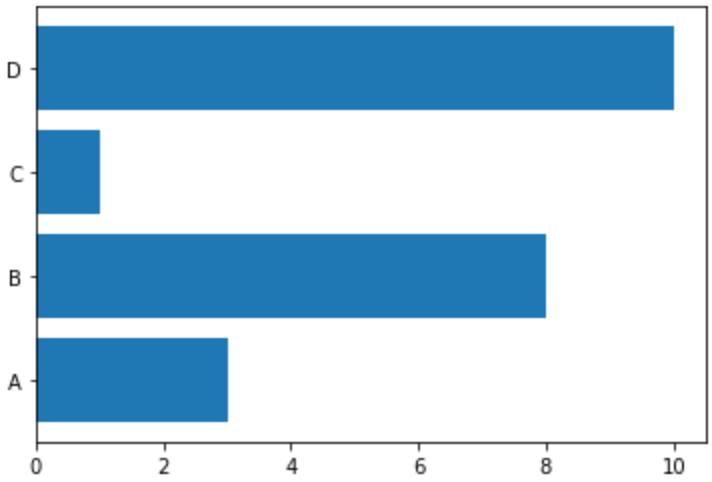
```
plt.bar([0.25,1.25,2.25,3.25,4.25],[50,40,70,80,20],label="BMW",color='b',width=0.5)
plt.bar([0.75,1.75,2.75,3.75,4.75],[80,20,20,50,60],label="Audi",color='r',width=0.5)
plt.legend()
plt.xlabel("Days")
plt.ylabel("Distance(kms)")
plt.title("information")
plt.show()
```



Horizontal Bars: If you want the bars to be displayed horizontally instead of vertically, use the barh() function

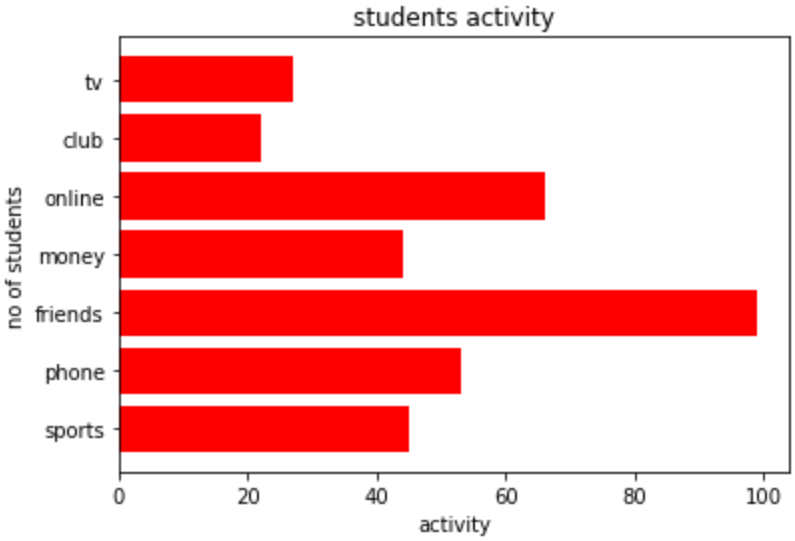
In [15]:

```
x = np.array(["A", "B", "C", "D"])
y = np.array([3, 8, 1, 10])
plt.barh(x, y)
plt.show()
```



In [3]:

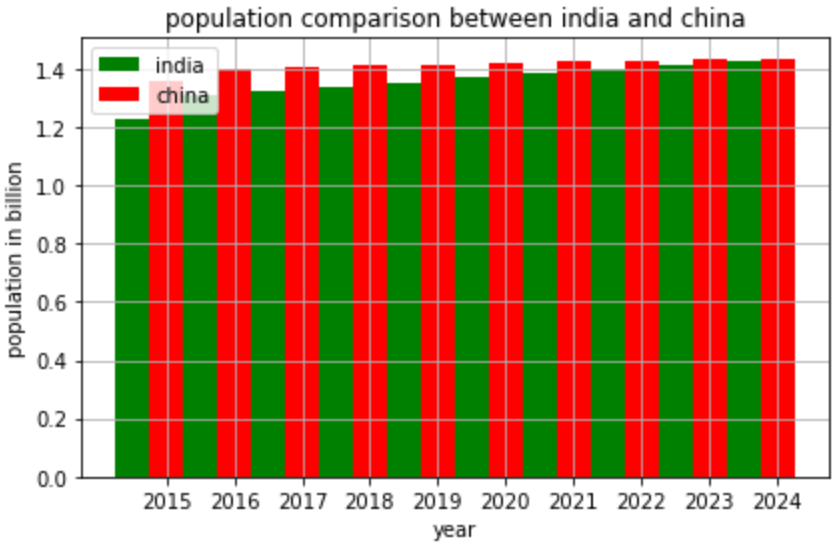
```
activities=['sports','phone','friends','money','online','club','tv']
frequency=[45,53,99,44,66,22,27]
plt.barh(activities,frequency,color='r')
plt.title("students activity")
plt.xlabel("activity")
plt.ylabel("no of students")
plt.show()
```



In [10]:

```
year=[2015,2016,2017,2018,2019,2020,2021,2022,2023,2024]
width=0.50
indices=np.arange(len(year))
print(indices)
population_china=[1.359,1.397,1.403,1.409,1.415,1.42,1.424,1.428,1.431,1.434]
population_india=[1.23,1.309,1.324,1.339,1.354,1.368,1.383,1.397,1.411,1.425]
plt.bar(indices-width,population_india,width=width,label="india",color="green")
plt.bar(indices,population_china,width=width,label="china",color="red")
plt.title("population comparison between india and china")
plt.xlabel('year')
plt.ylabel('population in billion')
plt.legend()
plt.xticks(ticks=indices,labels=year)
plt.tight_layout()
plt.grid(True)
plt.show()
```

```
[0 1 2 3 4 5 6 7 8 9]
```



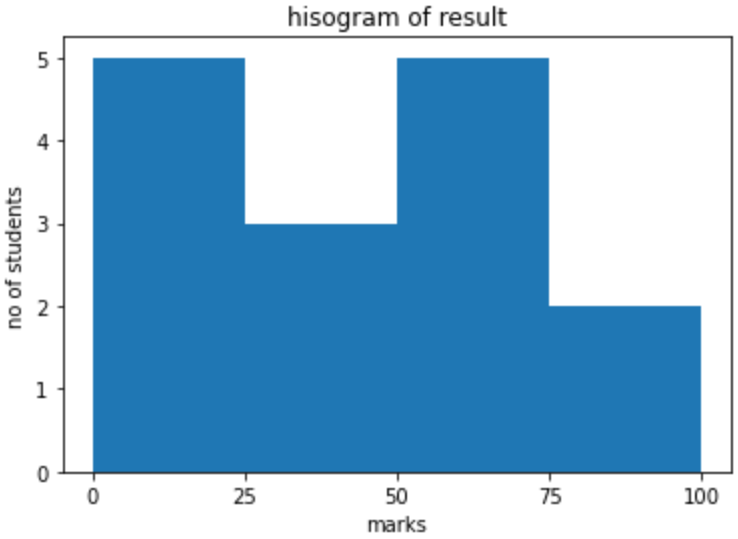
Histogram plot

A histogram is a graph showing frequency distributions.

It is a graph showing the number of observations within each given interval

The hist() function will use an array of numbers to create a histogram, the array is sent into the function as an argument

In [13]:
```
fig,ax=plt.subplots(1,1)
a=np.array([22,87,5,43,56,73,55,54,11,20,51,5,79,31,27])
ax.hist(a,bins=[0,25,50,75,100])
ax.set_title("hisogram of result")
ax.set_xticks([0,25,50,75,100])
ax.set_xlabel('marks')
ax.set_ylabel('no of students')
plt.show()
```



Pie plot

A pie chart, sometimes called a circle chart, is a way of summarizing a set of nominal data or displaying the different values of a given variable (e.g. percentage distribution). This type of chart is a circle divided into a series of segments.

In [14]:
```
y = np.array([35, 25, 25, 15])
plt.pie(y)
plt.show()
```

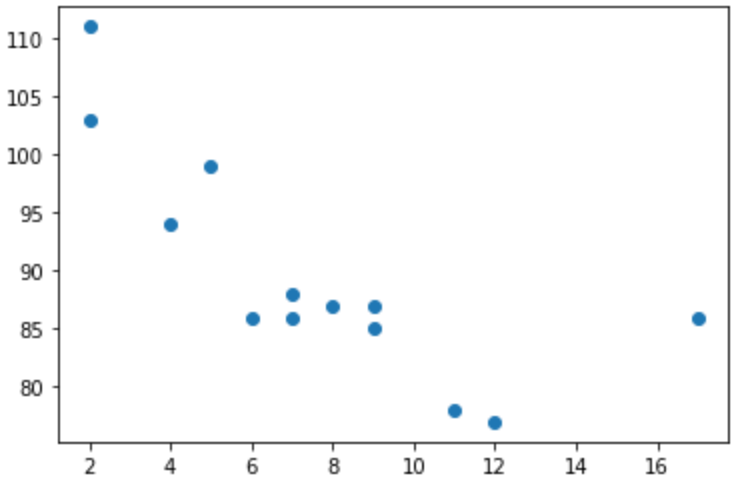
In [20]:
```
data=[1433783686,1366417754,329064917,27025568,216565318]
l= ['china','india','unitedstates','indonesia','pakistan']
mycolors = ['#ccffcc','orange','blue','green','#333300']
e=[0.1,0.0,0.0,0.0,0]
plt.pie(data, labels = l,explode=e,shadow=True,startangle=90,wedgeprops={'edgecolor':"blac
plt.title('population pie chart')
plt.show()
```

Scatter plot

The scatter() function plots one dot for each observation. It needs two arrays of the same length, one for the values of the x-axis, and one for values on the y-axis

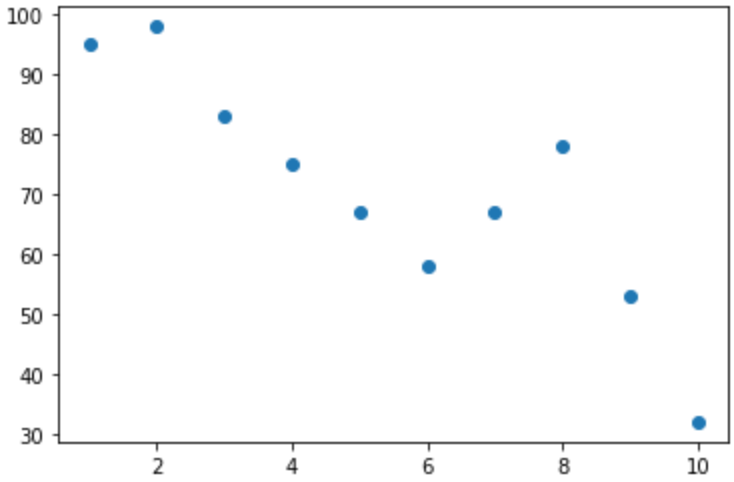
In [21]:

```
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])
plt.scatter(x, y)
plt.show()
```



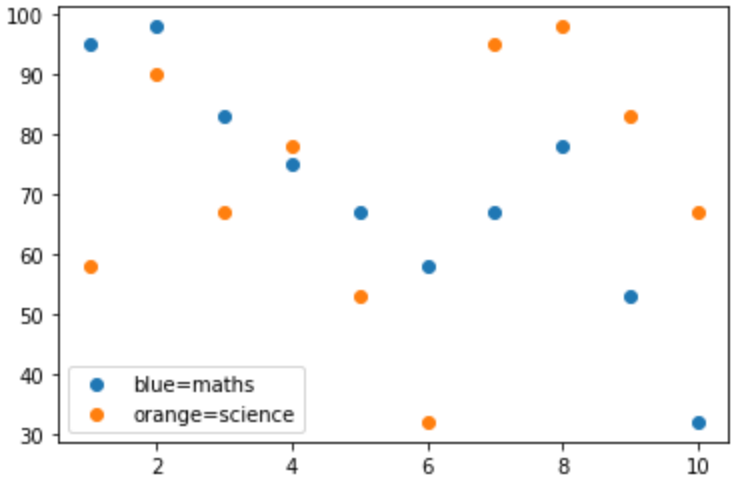
In [22]:

```
students_id=[1,2,3,4,5,6,7,8,9,10]
students_marks=[95,98,83,75,67,58,67,78,53,32]
plt.scatter(students_id,students_marks)
plt.show()
```



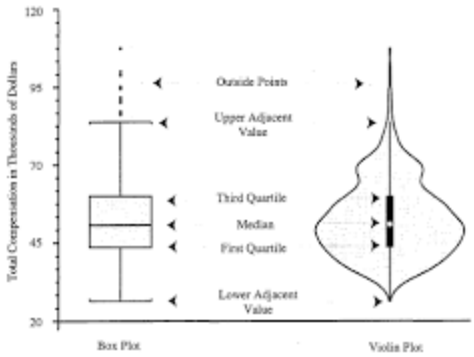
In [29]:

```
#maths marks
students_id=np.array([1,2,3,4,5,6,7,8,9,10])
students_marks=np.array([95,98,83,75,67,58,67,78,53,32])
plt.scatter(students_id,students_marks,label="blue=maths")
#science marks
students_id=np.array([1,2,3,4,5,6,7,8,9,10])
students_marks=np.array([58,90,67,78,53,32,95,98,83,67])
plt.scatter(students_id,students_marks,label="orange=science")
plt.legend()
plt.show()
```



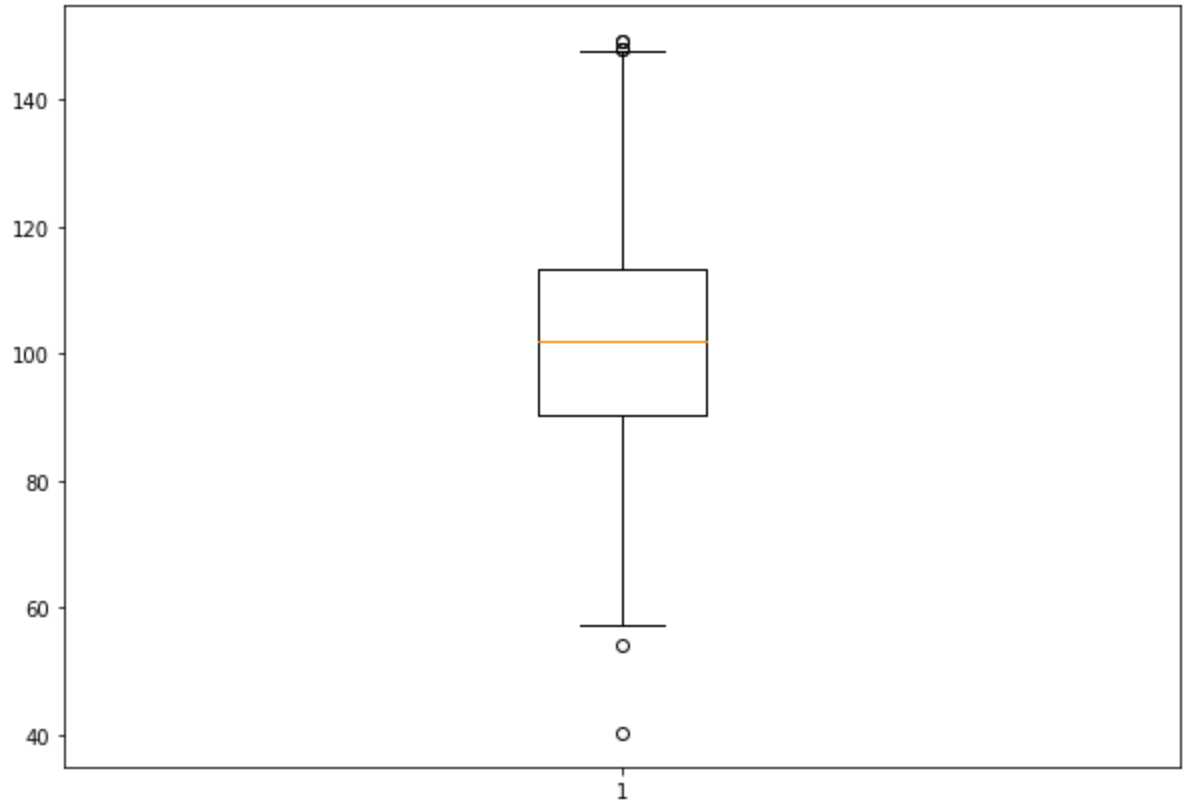
voilin plot

Violin plots are similar to box plots, except that they also show the probability density of the data at different values. These plots include a marker for the median of the data and a box indicating the interquartile range, as in the standard box plots



Box plot

A Box Plot is also known as Whisker plot is created to display the summary of the set of data values having properties like minimum, first quartile, median, third quartile and maximum. In the box plot, a box is created from the first quartile to the third quartile, a vertical line is also there which goes through the box at the median. Here x-axis denotes the data to be plotted while the y-axis shows the frequency distribution.

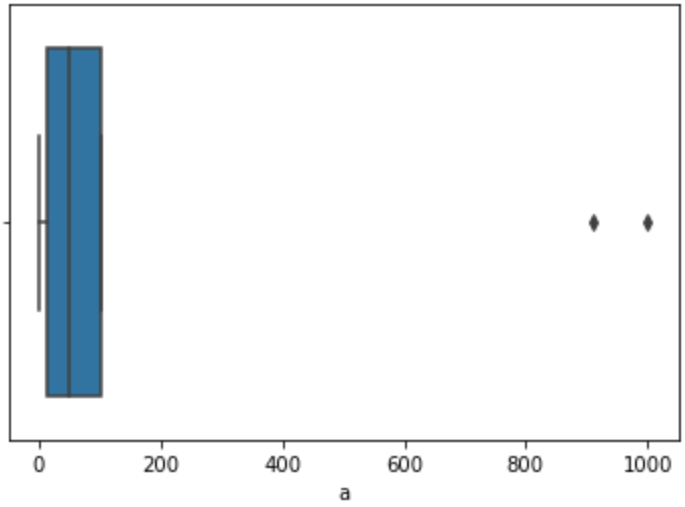


In [30]:

```
import seaborn as s
import pandas as pd
l=[0,1,50,60,50,14,909,1000,101]
df=pd.DataFrame(l,columns=['a'])
df
s.boxplot(df['a'])
```

```
D:\Archana\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following
variable as a keyword arg: x. From version 0.12, the only valid positional argument will b
e `data`, and passing other arguments without an explicit keyword will result in an error
or misinterpretation.
  warnings.warn(
<AxesSubplot:xlabel='a'>
```

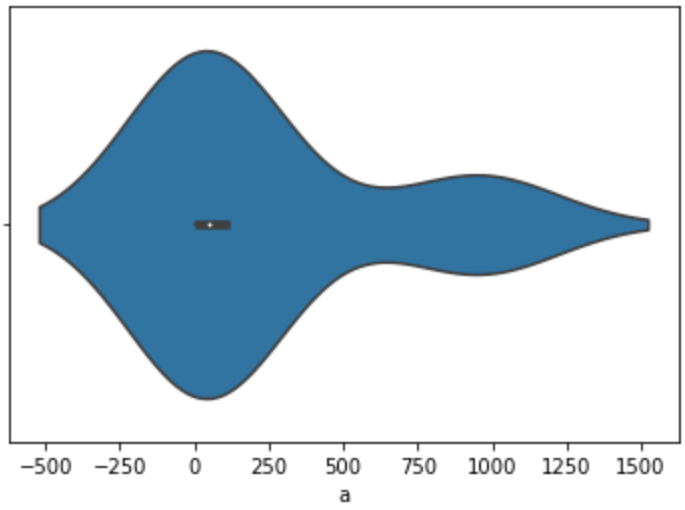
Out[30]:



In [32]:
```
s.violinplot(df['a'])
```

```
D:\Archana\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following
variable as a keyword arg: x. From version 0.12, the only valid positional argument will b
e `data`, and passing other arguments without an explicit keyword will result in an error
or misinterpretation.
  warnings.warn(
<AxesSubplot:xlabel='a'>
```

Out[32]:



In [1]:
```
import pandas as pd
import numpy as np
```

In [3]:
```
data=pd.read_csv("C:\\Users\\Dell\\Downloads\\archive (3)\\500 richest people 2021.csv",de
data
```

Out[3]:

	Rank	Name	Total Net Worth	$ Last Change	$ YTD Change	Country	Industry	Unnamed: 7	Unnamed: 8	Unnamed: 9	Unnamed 1
0	1.0	Jeff Bezos	$188B	+$1.68B	-$2.31B	United States	Technology	NaN	NaN	NaN	NaN
1	2.0	Elon Musk	$170B	-$2.89B	+$773M	United States	Technology	NaN	NaN	NaN	NaN
2	3.0	Bernard Arnault	$155B	+$892M	+$40.9B	France	Consumer	NaN	NaN	NaN	NaN

	Rank	Name	Total Net Worth	$ Last Change	$ YTD Change	Country	Industry	Unnamed: 7	Unnamed: 8	Unnamed: 9	Unnamed 1
3	4.0	Bill Gates	$144B	-$1.32B	+$12.2B	United States	Technology	NaN	NaN	NaN	NaN
4	5.0	Mark Zuckerberg	$114B	+$203M	+$10.9B	United States	Technology	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	.
498	500.0	Odd Reitan	$5.72B	-$19.9M	+$669M	Norway	Food & Beverage	NaN	NaN	NaN	NaN
499	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
500	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
501	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
502	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

503 rows × 11 columns

In [4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 503 entries, 0 to 502
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Rank             499 non-null    float64
 1   Name             499 non-null    object 
 2   Total Net Worth  499 non-null    object 
 3   $ Last Change    499 non-null    object 
 4   $ YTD Change     499 non-null    object 
 5   Country          499 non-null    object 
 6   Industry         499 non-null    object 
 7   Unnamed: 7       0 non-null      float64
 8   Unnamed: 8       0 non-null      float64
 9   Unnamed: 9       0 non-null      float64
 10  Unnamed: 10      0 non-null      float64
dtypes: float64(5), object(6)
memory usage: 43.4+ KB
```

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

Out[5]:

	Rank	Name	Total Net Worth	$ Last Change	$ YTD Change	Country	Industry	Unnamed: 7	Unnamed: 8	Unnamed: 9	Unnamed: 10
0	1.0	Jeff Bezos	$188B	+$1.68B	-$2.31B	United States	Technology	NaN	NaN	NaN	NaN
1	2.0	Elon Musk	$170B	-$2.89B	+$773M	United States	Technology	NaN	NaN	NaN	NaN
2	3.0	Bernard Arnault	$155B	+$892M	+$40.9B	France	Consumer	NaN	NaN	NaN	NaN
3	4.0	Bill Gates	$144B	-$1.32B	+$12.2B	United States	Technology	NaN	NaN	NaN	NaN

	Rank	Name	Total Net Worth	$ Last Change	$ YTD Change	Country	Industry	Unnamed: 7	Unnamed: 8	Unnamed: 9	Unnamed: 10
4	5.0	Mark Zuckerberg	$114B	+$203M	+$10.9B	United States	Technology	NaN	NaN	NaN	NaN

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

Out[6]:

	Rank	Name	Total Net Worth	$ Last Change	$ YTD Change	Country	Industry	Unnamed: 7	Unnamed: 8	Unnamed: 9	Unnamed: 10
498	500.0	Odd Reitan	$5.72B	-$19.9M	+$669M	Norway	Food & Beverage	NaN	NaN	NaN	NaN
499	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
500	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
501	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
502	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [8]:
```
s=data.drop(["Unnamed: 7","Unnamed: 8","Unnamed: 9","Unnamed: 10"],axis='columns',inplace=
```

In [9]:
```
data
```

Out[9]:

	Rank	Name	Total Net Worth	$ Last Change	$ YTD Change	Country	Industry
0	1.0	Jeff Bezos	$188B	+$1.68B	-$2.31B	United States	Technology
1	2.0	Elon Musk	$170B	-$2.89B	+$773M	United States	Technology
2	3.0	Bernard Arnault	$155B	+$892M	+$40.9B	France	Consumer
3	4.0	Bill Gates	$144B	-$1.32B	+$12.2B	United States	Technology
4	5.0	Mark Zuckerberg	$114B	+$203M	+$10.9B	United States	Technology
...	...	...	...	...	...	...	...
498	500.0	Odd Reitan	$5.72B	-$19.9M	+$669M	Norway	Food & Beverage
499	NaN	NaN	NaN	NaN	NaN	NaN	NaN
500	NaN	NaN	NaN	NaN	NaN	NaN	NaN
501	NaN	NaN	NaN	NaN	NaN	NaN	NaN
502	NaN	NaN	NaN	NaN	NaN	NaN	NaN

503 rows × 7 columns

In [10]:
```
data.head()
```

Out[10]:

	Rank	Name	Total Net Worth	$ Last Change	$ YTD Change	Country	Industry
0	1.0	Jeff Bezos	$188B	+$1.68B	-$2.31B	United States	Technology
1	2.0	Elon Musk	$170B	-$2.89B	+$773M	United States	Technology

	Rank	Name	Total Net Worth	$ Last Change	$ YTD Change	Country	Industry
2	3.0	Bernard Arnault	$155B	+$892M	+$40.9B	France	Consumer
3	4.0	Bill Gates	$144B	-$1.32B	+$12.2B	United States	Technology
4	5.0	Mark Zuckerberg	$114B	+$203M	+$10.9B	United States	Technology

In [11]:
```
data.isnull().sum()
```

Out[11]:
```
Rank                4
Name                4
Total Net Worth     4
$ Last Change       4
$ YTD Change        4
Country             4
Industry            4
dtype: int64
```

In [12]:
```
data.dropna(axis=0,inplace=True)
```

In [13]:
```
data.isnull().sum()
```

Out[13]:
```
Rank                0
Name                0
Total Net Worth     0
$ Last Change       0
$ YTD Change        0
Country             0
Industry            0
dtype: int64
```

In [14]:
```
data['Country'].values
```

Out[14]:
```
array(['United States', 'United States', 'France', 'United States',
       'United States', 'United States', 'United States', 'United States',
       'United States', 'United States', 'France', 'Spain ', 'India',
       'United States', 'United States', 'China', 'India',
       'United States', 'United States', 'United States', 'China',
       'United States', 'Mexico', 'France', 'United States',
       'United States', 'China', 'United States', 'United States',
       'China', 'China', 'France', 'France', 'Japan', 'United States',
       'Italy', 'United States', 'Germany', 'Japan', 'China', 'Russia',
       'Hong Kong', 'Hong Kong', 'China', 'India', 'China', 'Australia',
       'United States', 'United States', 'China', 'United Kingdom',
       'Russia', 'United Kingdom', 'Russia', 'Russia', 'Japan',
       'United States', 'Ireland', 'China', 'Italy', 'Germany',
       'United States', 'China', 'Chile', 'Brazil', 'Germany', 'China',
       'United States', 'Sweden', 'Australia', 'Singapore', 'China',
       'Hong Kong', 'United States', 'India', 'Hong Kong',
       'United States', 'United States', 'United States', 'United States',
       'United States', 'Singapore', 'United States', 'Germany', 'Mexico',
       'Russia', 'India', 'Malaysia', 'United States', 'Switzerland',
       'China', 'Hong Kong', 'United States', 'China', 'Hong Kong',
       'Russia', 'Hong Kong', 'Russia', 'United States', 'Russia',
       'China', 'Austria', 'China', 'United States', 'Hong Kong',
       'Netherlands', 'Russia', 'Russia', 'United States', 'Germany',
       'Germany', 'Germany', 'Saudi Arabia', 'Brazil', 'China', 'China',
       'China', 'Sweden', 'Nigeria', 'China', 'China', 'India',
       'Indonesia', 'Germany', 'China', 'United States', 'United States',
       'Thailand', 'United States', 'Mexico', 'United States',
```

'United States', 'India', 'Indonesia', 'China', 'Brazil',
'United States', 'United States', 'United States', 'India',
'United States', 'Germany', 'Germany', 'China', 'United States',
'France', 'China', 'China', 'Australia', 'Australia', 'Russia',
'China', 'China', 'Russia', 'United States', 'China',
'United States', 'Canada', 'United Kingdom', 'Colombia', 'China',
'United States', 'China', 'China', 'Russia', 'Hong Kong',
'Singapore', 'Canada', 'China', 'Korea', 'Italy', 'China', 'China',
'United States', 'United States', 'United States', 'China',
'Sweden', 'United States', 'United States', 'Brazil', 'Hong Kong',
'India', 'United States', 'India', 'China', 'United States',
'China', 'Ireland', 'Ireland', 'Australia', 'United States',
'United States', 'China', 'United States', 'United States',
'United States', 'United States', 'Israel', 'Monaco', 'Germany',
'United States', 'United States', 'Denmark', 'United States',
'New Zealand', 'United States', 'United States', 'United States',
'Russia', 'China', 'France', 'China', 'China', 'United States',
'Brazil', 'China', 'China', 'Viet Nam', 'Hong Kong', 'China',
'Korea', 'United States', 'India', 'Korea', 'Germany', 'Ukraine',
'United States', 'United States', 'Russia', 'Korea', 'Singapore',
'United States', 'United Kingdom', 'China', 'Singapore',
'United States', 'Australia', 'Austria', 'Germany', 'Germany',
'Sweden', 'United States', 'United States', 'United States',
'Switzerland', 'Cyprus', 'China', 'Italy', 'Thailand',
'United States', 'United States', 'Colombia', 'China',
'United States', 'Denmark', 'United States', 'India', 'Canada',
'Mexico', 'United States', 'United Kingdom', 'United States',
'China', 'Sweden', 'China', 'Germany', 'United States',
'United States', 'Russia', 'Singapore', 'United Kingdom', 'Russia',
'China', 'United States', 'Russia', 'United States', 'Switzerland',
'United States', 'Canada', 'United States', 'United States',
'Russia', 'United States', 'United Kingdom', 'South Africa',
'Italy', 'United Kingdom', 'China', 'Germany', 'Germany',
'United States', 'United States', 'Russia', 'China', 'Spain',
'Hong Kong', 'China', 'Germany', 'United States', 'United States',
'United States', 'Saudi Arabia', 'United States', 'United States',
'India', 'Mexico', 'Canda', 'Japan', 'China', 'United States',
'China', 'United States', 'Germany', 'Hong Kong', 'China',
'United States', 'United States', 'United States', 'Germany',
'China', 'China', 'Romania', 'Sweden', 'United States',
'Switzerland', 'France', 'France', 'France', 'United States',
'United States', 'China', 'Mexico', 'United States', 'China',
'United States', 'Switzerland', 'Germany', 'China', 'Australia',
'South Africa', 'Canada', 'Canada', 'Canada', 'United States',
'Russia', 'India', 'Hong Kong', 'Sweden', 'Sweden',
'United States', 'United States', 'United Kingdom', 'China',
'Germany', 'United States', 'China', 'United States', 'Germany',
'Korea', 'United States', 'China', 'Germany', 'Sweden',
'United Kingdom', 'Japan', 'Ireland', 'Singapore', 'Taiwan',
'China', 'Indonesia', 'Germany', 'Saudi Arabia', 'United States',
'United States', 'United States', 'Japan', 'China', 'Malaysia',
'United States', 'Russia', 'United States', 'Finland',
'United Kingdom', 'United States', 'Egypt', 'France',
'South Africa', 'Cayman Islands', 'Hong Kong', 'Israel',
'United States', 'China', 'Singapore', 'Israel', 'India',
'Denmark', 'United States', 'United States', 'United States',
'United States', 'Sweden', 'China', 'United States', 'Sweden',
'United States', 'France', 'United Kingdom', 'United States',
'Canada', 'Hong Kong', 'France', 'United States', 'United States',
'United States', 'United States', 'Denmark', 'Denmark', 'Denmark',
'China', 'France', 'United States', 'China', 'United Kingdom',
'Japan', 'India', 'India', 'United Kingdom', 'United Kingdom',
'China', 'United States', 'United States', 'Russia', 'Korea',
'France', 'United States', 'United States', 'Germany', 'Germany',
'China', 'United States', 'Australia', 'United States',
'Switzerland', 'Singapore', 'Russia', 'United States', 'Singapore',

```
       'Taiwan', 'United States', 'Canada', 'Canada', 'Germany', 'Norway',
       'Taiwan', 'United States', 'United States', 'Germany',
       'United States', 'China', 'Philippines', 'United States',
       'United States', 'Germany', 'Norway', 'Taiwan', 'United States',
       'United States', 'United States', 'Netherlands', 'Canada',
       'Russia', 'Switzerland', 'Switzerland', 'United Arab Emirates',
       'United States', 'China', 'United States', 'United Kingdom',
       'Italy', 'Hong Kong', 'Argentina', 'Georgia', 'Germany', 'Italy',
       'United States', 'Hong Kong', 'United States', 'China', 'Canada',
       'France', 'United Kingdom', 'Korea', 'United States', 'Canada',
       'Indonesia', 'United States', 'Kazakhstan', 'Norway'], dtype=object)
```

In [15]:

```
data=data.replace('Canda','Canada',regex=True)
data.head()
data.dtypes
```

Out[15]:

```
Rank                float64
Name                 object
Total Net Worth      object
$ Last Change        object
$ YTD Change         object
Country              object
Industry             object
dtype: object
```

In [16]:

```
data['Country'].values
```

Out[16]:

```
array(['United States', 'United States', 'France', 'United States',
       'United States', 'United States', 'United States', 'United States',
       'United States', 'United States', 'France', 'Spain ', 'India',
       'United States', 'United States', 'China', 'India',
       'United States', 'United States', 'United States', 'China',
       'United States', 'Mexico', 'France', 'United States',
       'United States', 'China', 'United States', 'United States',
       'China', 'China', 'France', 'France', 'Japan', 'United States',
       'Italy', 'United States', 'Germany', 'Japan', 'China', 'Russia',
       'Hong Kong', 'Hong Kong', 'China', 'India', 'China', 'Australia',
       'United States', 'United States', 'China', 'United Kingdom',
       'Russia', 'United Kingdom', 'Russia', 'Russia', 'Japan',
       'United States', 'Ireland', 'China', 'Italy', 'Germany',
       'United States', 'China', 'Chile', 'Brazil', 'Germany', 'China',
       'United States', 'Sweden', 'Australia', 'Singapore', 'China',
       'Hong Kong', 'United States', 'India', 'Hong Kong',
       'United States', 'United States', 'United States', 'United States',
       'United States', 'Singapore', 'United States', 'Germany', 'Mexico',
       'Russia', 'India', 'Malaysia', 'United States', 'Switzerland',
       'China', 'Hong Kong', 'United States', 'China', 'Hong Kong',
       'Russia', 'Hong Kong', 'Russia', 'United States', 'Russia',
       'China', 'Austria', 'China', 'United States', 'Hong Kong',
       'Netherlands', 'Russia', 'Russia', 'United States', 'Germany',
       'Germany', 'Germany', 'Saudi Arabia', 'Brazil', 'China', 'China',
       'China', 'Sweden', 'Nigeria', 'China', 'China', 'India',
       'Indonesia', 'Germany', 'China', 'United States', 'United States',
       'Thailand', 'United States', 'Mexico', 'United States',
       'United States', 'India', 'Indonesia', 'China', 'Brazil',
       'United States', 'United States', 'United States', 'India',
       'United States', 'Germany', 'Germany', 'China', 'United States',
       'France', 'China', 'China', 'Australia', 'Australia', 'Russia',
       'China', 'China', 'Russia', 'United States', 'China',
       'United States', 'Canada', 'United Kingdom', 'Colombia', 'China',
       'United States', 'China', 'China', 'Russia', 'Hong Kong',
       'Singapore', 'Canada', 'China', 'Korea', 'Italy', 'China', 'China',
       'United States', 'United States', 'United States', 'China',
       'Sweden', 'United States', 'United States', 'Brazil', 'Hong Kong',
```

'India', 'United States', 'India', 'China', 'United States',
'China', 'Ireland', 'Ireland', 'Australia', 'United States',
'United States', 'China', 'United States', 'United States',
'United States', 'United States', 'Israel', 'Monaco', 'Germany',
'United States', 'United States', 'Denmark', 'United States',
'New Zealand', 'United States', 'United States', 'United States',
'Russia', 'China', 'France', 'China', 'China', 'United States',
'Brazil', 'China', 'China', 'Viet Nam', 'Hong Kong', 'China',
'Korea', 'United States', 'India', 'Korea', 'Germany', 'Ukraine',
'United States', 'United States', 'Russia', 'Korea', 'Singapore',
'United States', 'United Kingdom', 'China', 'Singapore',
'United States', 'Australia', 'Austria', 'Germany', 'Germany',
'Sweden', 'United States', 'United States', 'United States',
'Switzerland', 'Cyprus', 'China', 'Italy', 'Thailand',
'United States', 'United States', 'Colombia', 'China',
'United States', 'Denmark', 'United States', 'India', 'Canada',
'Mexico', 'United States', 'United Kingdom', 'United States',
'China', 'Sweden', 'China', 'Germany', 'United States',
'United States', 'Russia', 'Singapore', 'United Kingdom', 'Russia',
'China', 'United States', 'Russia', 'United States', 'Switzerland',
'United States', 'Canada', 'United States', 'United States',
'Russia', 'United States', 'United Kingdom', 'South Africa',
'Italy', 'United Kingdom', 'China', 'Germany', 'Germany',
'United States', 'United States', 'Russia', 'China', 'Spain',
'Hong Kong', 'China', 'Germany', 'United States', 'United States',
'United States', 'Saudi Arabia', 'United States', 'United States',
'India', 'Mexico', 'Canada', 'Japan', 'China', 'United States',
'China', 'United States', 'Germany', 'Hong Kong', 'China',
'United States', 'United States', 'United States', 'Germany',
'China', 'China', 'Romania', 'Sweden', 'United States',
'Switzerland', 'France', 'France', 'France', 'United States',
'United States', 'China', 'Mexico', 'United States', 'China',
'United States', 'Switzerland', 'Germany', 'China', 'Australia',
'South Africa', 'Canada', 'Canada', 'Canada', 'United States',
'Russia', 'India', 'Hong Kong', 'Sweden', 'Sweden',
'United States', 'United States', 'United Kingdom', 'China',
'Germany', 'United States', 'China', 'United States', 'Germany',
'Korea', 'United States', 'China', 'Germany', 'Sweden',
'United Kingdom', 'Japan', 'Ireland', 'Singapore', 'Taiwan',
'China', 'Indonesia', 'Germany', 'Saudi Arabia', 'United States',
'United States', 'United States', 'Japan', 'China', 'Malaysia',
'United States', 'Russia', 'United States', 'Finland',
'United Kingdom', 'United States', 'Egypt', 'France',
'South Africa', 'Cayman Islands', 'Hong Kong', 'Israel',
'United States', 'China', 'Singapore', 'Israel', 'India',
'Denmark', 'United States', 'United States', 'United States',
'United States', 'Sweden', 'China', 'United States', 'Sweden',
'United States', 'France', 'United Kingdom', 'United States',
'Canada', 'Hong Kong', 'France', 'United States', 'United States',
'United States', 'United States', 'Denmark', 'Denmark', 'Denmark',
'China', 'France', 'United States', 'China', 'United Kingdom',
'Japan', 'India', 'India', 'United Kingdom', 'United Kingdom',
'China', 'United States', 'United States', 'Russia', 'Korea',
'France', 'United States', 'United States', 'Germany', 'Germany',
'China', 'United States', 'Australia', 'United States',
'Switzerland', 'Singapore', 'Russia', 'United States', 'Singapore',
'Taiwan', 'United States', 'Canada', 'Canada', 'Germany', 'Norway',
'Taiwan', 'United States', 'United States', 'Germany',
'United States', 'China', 'Philippines', 'United States',
'United States', 'Germany', 'Norway', 'Taiwan', 'United States',
'United States', 'United States', 'Netherlands', 'Canada',
'Russia', 'Switzerland', 'Switzerland', 'United Arab Emirates',
'United States', 'China', 'United States', 'United Kingdom',
'Italy', 'Hong Kong', 'Argentina', 'Georgia', 'Germany', 'Italy',
'United States', 'Hong Kong', 'United States', 'China', 'Canada',

```
'France', 'United Kingdom', 'Korea', 'United States', 'Canada',
'Indonesia', 'United States', 'Kazakhstan', 'Norway'], dtype=object)
```

In [17]:
```
data
```

Out[17]:

	Rank	Name	Total Net Worth	$ Last Change	$ YTD Change	Country	Industry
0	1.0	Jeff Bezos	$188B	+$1.68B	-$2.31B	United States	Technology
1	2.0	Elon Musk	$170B	-$2.89B	+$773M	United States	Technology
2	3.0	Bernard Arnault	$155B	+$892M	+$40.9B	France	Consumer
3	4.0	Bill Gates	$144B	-$1.32B	+$12.2B	United States	Technology
4	5.0	Mark Zuckerberg	$114B	+$203M	+$10.9B	United States	Technology
...	...	...	...	...	...	...	...
494	496.0	Lino Saputo	$5.75B	-$48.0M	+$772M	Canada	Food & Beverage
495	497.0	Prajogo Pangestu	$5.74B	-$74.7M	-$1.03B	Indonesia	Energy
496	498.0	Charles Dolan & Family	$5.74B	-$35.8M	+$212M	United States	Media & Telecom
497	499.0	Vladimir Kim	$5.72B	+$2.80M	+$792M	Kazakhstan	Commodities
498	500.0	Odd Reitan	$5.72B	-$19.9M	+$669M	Norway	Food & Beverage

499 rows × 7 columns

In [18]:
```
data=data.replace(['\$',""],"",regex=True)
data
```

Out[18]:

	Rank	Name	Total Net Worth	$ Last Change	$ YTD Change	Country	Industry
0	1.0	Jeff Bezos	188B	+1.68B	-2.31B	United States	Technology
1	2.0	Elon Musk	170B	-2.89B	+773M	United States	Technology
2	3.0	Bernard Arnault	155B	+892M	+40.9B	France	Consumer
3	4.0	Bill Gates	144B	-1.32B	+12.2B	United States	Technology
4	5.0	Mark Zuckerberg	114B	+203M	+10.9B	United States	Technology
...	...	...	...	...	...	...	...
494	496.0	Lino Saputo	5.75B	-48.0M	+772M	Canada	Food & Beverage
495	497.0	Prajogo Pangestu	5.74B	-74.7M	-1.03B	Indonesia	Energy
496	498.0	Charles Dolan & Family	5.74B	-35.8M	+212M	United States	Media & Telecom
497	499.0	Vladimir Kim	5.72B	+2.80M	+792M	Kazakhstan	Commodities
498	500.0	Odd Reitan	5.72B	-19.9M	+669M	Norway	Food & Beverage

499 rows × 7 columns

In [19]:
```
data["Total Net Worth"]=data["Total Net Worth"].replace("B","",regex=True)
```

In [20]:
```
data
```

Out[20]:

	Rank	Name	Total Net Worth	$ Last Change	$ YTD Change	Country	Industry
0	1.0	Jeff Bezos	188	+1.68B	-2.31B	United States	Technology
1	2.0	Elon Musk	170	-2.89B	+773M	United States	Technology
2	3.0	Bernard Arnault	155	+892M	+40.9B	France	Consumer
3	4.0	Bill Gates	144	-1.32B	+12.2B	United States	Technology
4	5.0	Mark Zuckerberg	114	+203M	+10.9B	United States	Technology
...	...	...	...	...	...	...	...
494	496.0	Lino Saputo	5.75	-48.0M	+772M	Canada	Food & Beverage
495	497.0	Prajogo Pangestu	5.74	-74.7M	-1.03B	Indonesia	Energy
496	498.0	Charles Dolan & Family	5.74	-35.8M	+212M	United States	Media & Telecom
497	499.0	Vladimir Kim	5.72	+2.80M	+792M	Kazakhstan	Commodities
498	500.0	Odd Reitan	5.72	-19.9M	+669M	Norway	Food & Beverage

499 rows × 7 columns

In [21]:
```
data["Total Net Worth"]=pd.to_numeric(data["Total Net Worth"],errors="coerce")
data["Total Net Worth"].dtype
```

Out[21]:
```
dtype('float64')
```

In [22]:
```
data.head()
```

Out[22]:

	Rank	Name	Total Net Worth	$ Last Change	$ YTD Change	Country	Industry
0	1.0	Jeff Bezos	188.0	+1.68B	-2.31B	United States	Technology
1	2.0	Elon Musk	170.0	-2.89B	+773M	United States	Technology
2	3.0	Bernard Arnault	155.0	+892M	+40.9B	France	Consumer
3	4.0	Bill Gates	144.0	-1.32B	+12.2B	United States	Technology
4	5.0	Mark Zuckerberg	114.0	+203M	+10.9B	United States	Technology

In [23]:
```
def value_to_float(x):
    if 'K' in x:
        if len(x)>1:
            return float(x.replace('K',''))*0.000001
    if 'M' in x:
        if len(x)>1:
            return float(x.replace('M',''))*1000
    if 'B' in x:
        if len(x)>1:
            return float(x.replace('B',''))*100000
data['$ Last Change']=data['$ Last Change'].apply(value_to_float)
```

In [24]:
```
data.head()
```

Out[24]:

	Rank	Name	Total Net Worth	$ Last Change	$ YTD Change	Country	Industry

	Rank	Name	Total Net Worth	$ Last Change	$ YTD Change	Country	Industry
0	1.0	Jeff Bezos	188.0	168000.0	-2.31B	United States	Technology
1	2.0	Elon Musk	170.0	-289000.0	+773M	United States	Technology
2	3.0	Bernard Arnault	155.0	892000.0	+40.9B	France	Consumer
3	4.0	Bill Gates	144.0	-132000.0	+12.2B	United States	Technology
4	5.0	Mark Zuckerberg	114.0	203000.0	+10.9B	United States	Technology

In [25]:
```
def value_to_float(x):
    if 'K' in x:
        if len(x)>1:
            return float(x.replace('K',''))*0.000001
    if 'M' in x:
        if len(x)>1:
            return float(x.replace('M',''))*1000
    if 'B' in x:
        if len(x)>1:
            return float(x.replace('B',''))*100000
data['$ YTD Change']=data['$ YTD Change'].apply(value_to_float)
```

In [26]:
```
data.head()
```

Out[26]:

	Rank	Name	Total Net Worth	$ Last Change	$ YTD Change	Country	Industry
0	1.0	Jeff Bezos	188.0	168000.0	-231000.0	United States	Technology
1	2.0	Elon Musk	170.0	-289000.0	773000.0	United States	Technology
2	3.0	Bernard Arnault	155.0	892000.0	4090000.0	France	Consumer
3	4.0	Bill Gates	144.0	-132000.0	1220000.0	United States	Technology
4	5.0	Mark Zuckerberg	114.0	203000.0	1090000.0	United States	Technology

In [27]:
```
data.dtypes
```

Out[27]:
```
Rank               float64
Name                object
Total Net Worth    float64
$ Last Change      float64
$ YTD Change       float64
Country             object
Industry            object
dtype: object
```

In [28]:
```
display(data[data['Total Net Worth']>=50])
```

	Rank	Name	Total Net Worth	$ Last Change	$ YTD Change	Country	Industry
0	1.0	Jeff Bezos	188.0	168000.0	-231000.0	United States	Technology
1	2.0	Elon Musk	170.0	-289000.0	773000.0	United States	Technology
2	3.0	Bernard Arnault	155.0	892000.0	4090000.0	France	Consumer
3	4.0	Bill Gates	144.0	-132000.0	1220000.0	United States	Technology
4	5.0	Mark Zuckerberg	114.0	203000.0	1090000.0	United States	Technology

	Rank	Name	Total Net Worth	$ Last Change	$ YTD Change	Country	Industry
5	6.0	Warren Buffett	108.0	-232000.0	2060000.0	United States	Diversified
6	7.0	Larry Page	104.0	-112000.0	2160000.0	United States	Technology
7	8.0	Sergey Brin	101.0	-106000.0	2080000.0	United States	Technology
8	9.0	Larry Ellison	90.6	-246000.0	1090000.0	United States	Technology
9	10.0	Steve Ballmer	89.1	-342000.0	871000.0	United States	Technology
10	11.0	Francoise Bettencourt Meyers	83.9	-133000.0	811000.0	France	Consumer
11	12.0	Amancio Ortega	79.3	-107000.0	1280000.0	Spain	Retail
12	13.0	Mukesh Ambani	74.1	402000.0	-261000.0	India	Energy
13	14.0	Charles Koch	64.3	-195000.0	742000.0	United States	Industrial
14	15.0	Julia Flesher Koch & Family	64.3	-195000.0	742000.0	United States	Industrial
15	16.0	Zhong Shanshan	63.8	-289000.0	-1440000.0	China	Diversified
16	17.0	Gautam Adani	62.8	210000.0	2900000.0	India	Industrial
17	18.0	Jim Walton	62.0	-508000.0	-479000.0	United States	Retail
18	19.0	Rob Walton	61.5	-496000.0	-111000.0	United States	Retail
19	20.0	Alice Walton	60.1	-489000.0	-225000.0	United States	Retail
20	21.0	Ma Huateng	58.4	-985000.0	194000.0	China	Technology
21	22.0	MacKenzie Scott	57.0	582000.0	-149000.0	United States	Technology
22	23.0	Carlos Slim	56.5	-348000.0	158000.0	Mexico	Diversified
23	24.0	Francois Pinault	53.7	-697000.0	593000.0	France	Consumer
24	25.0	Phil Knight & Family	53.3	123000.0	-622000.0	United States	Consumer
25	26.0	Michael Dell	50.4	-849000.0	1020000.0	United States	Technology

In [29]:
```
display(data[data.Country=='India'])
```

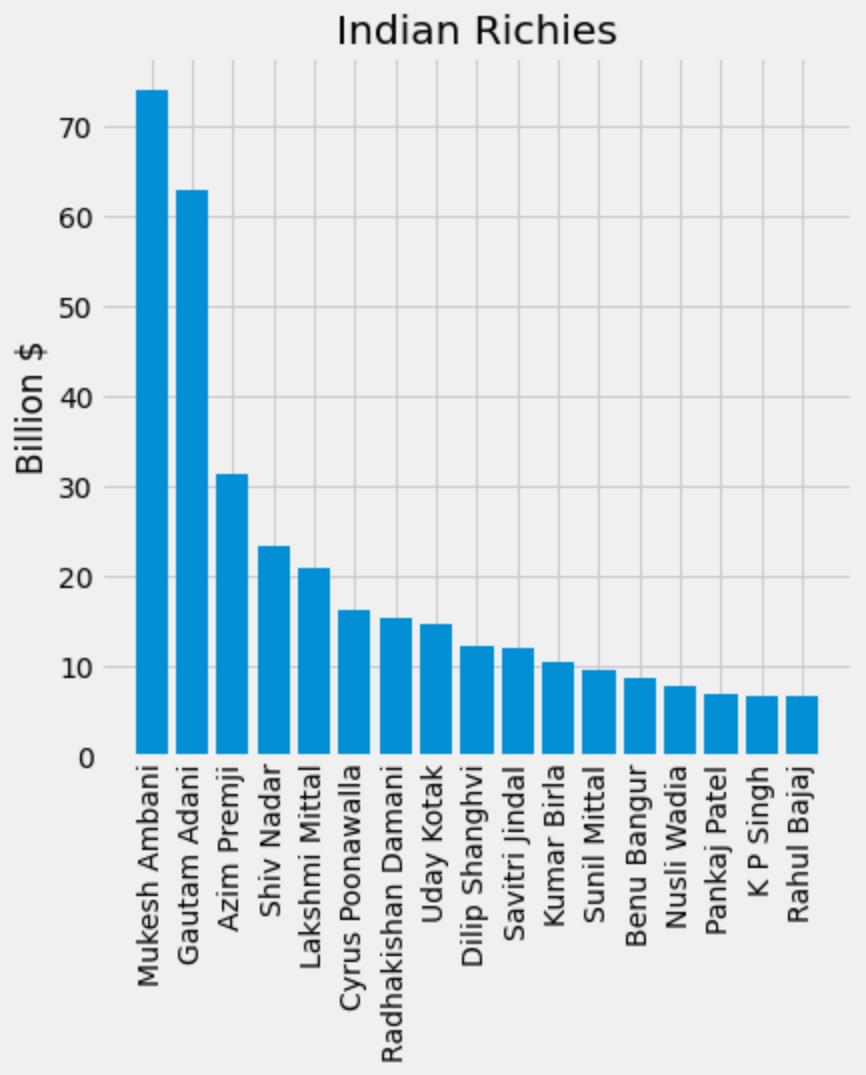
	Rank	Name	Total Net Worth	$ Last Change	$ YTD Change	Country	Industry
12	13.0	Mukesh Ambani	74.10	402000.0	-261000.0	India	Energy
16	17.0	Gautam Adani	62.80	210000.0	2900000.0	India	Industrial
44	45.0	Azim Premji	31.40	-347000.0	601000.0	India	Technology
74	75.0	Shiv Nadar	23.20	-102000.0	-904000.0	India	Technology
86	87.0	Lakshmi Mittal	20.90	-252000.0	448000.0	India	Retail
121	122.0	Cyrus Poonawalla	16.20	-45000.0	-25500.0	India	Health Care
132	133.0	Radhakishan Damani	15.40	8990.0	481000.0	India	Retail
139	140.0	Uday Kotak	14.60	-359000.0	-175000.0	India	Finance
182	183.0	Dilip Shanghvi	12.10	207000.0	189000.0	India	Health Care
184	185.0	Savitri Jindal	11.90	-266000.0	459000.0	India	Commodities
223	224.0	Kumar Birla	10.40	-62600.0	355000.0	India	Industrial
257	258.0	Sunil Mittal	9.59	-57800.0	696000.0	India	Media & Telecom

	Rank	Name	Total Net Worth	$ Last Change	$ YTD Change	Country	Industry
305	306.0	Benu Bangur	8.61	81000.0	105000.0	India	Commodities
346	347.0	Nusli Wadia	7.76	57700.0	-128000.0	India	Diversified
395	396.0	Pankaj Patel	6.86	75500.0	158000.0	India	Health Care
425	427.0	K P Singh	6.54	-6280.0	874000.0	India	Real-Estate
426	428.0	Rahul Bajaj	6.53	-21800.0	966000.0	India	Diversified

In [31]:
```
import matplotlib.pyplot as  plt
from matplotlib import style
from matplotlib import figure
style.use('fivethirtyeight')
```

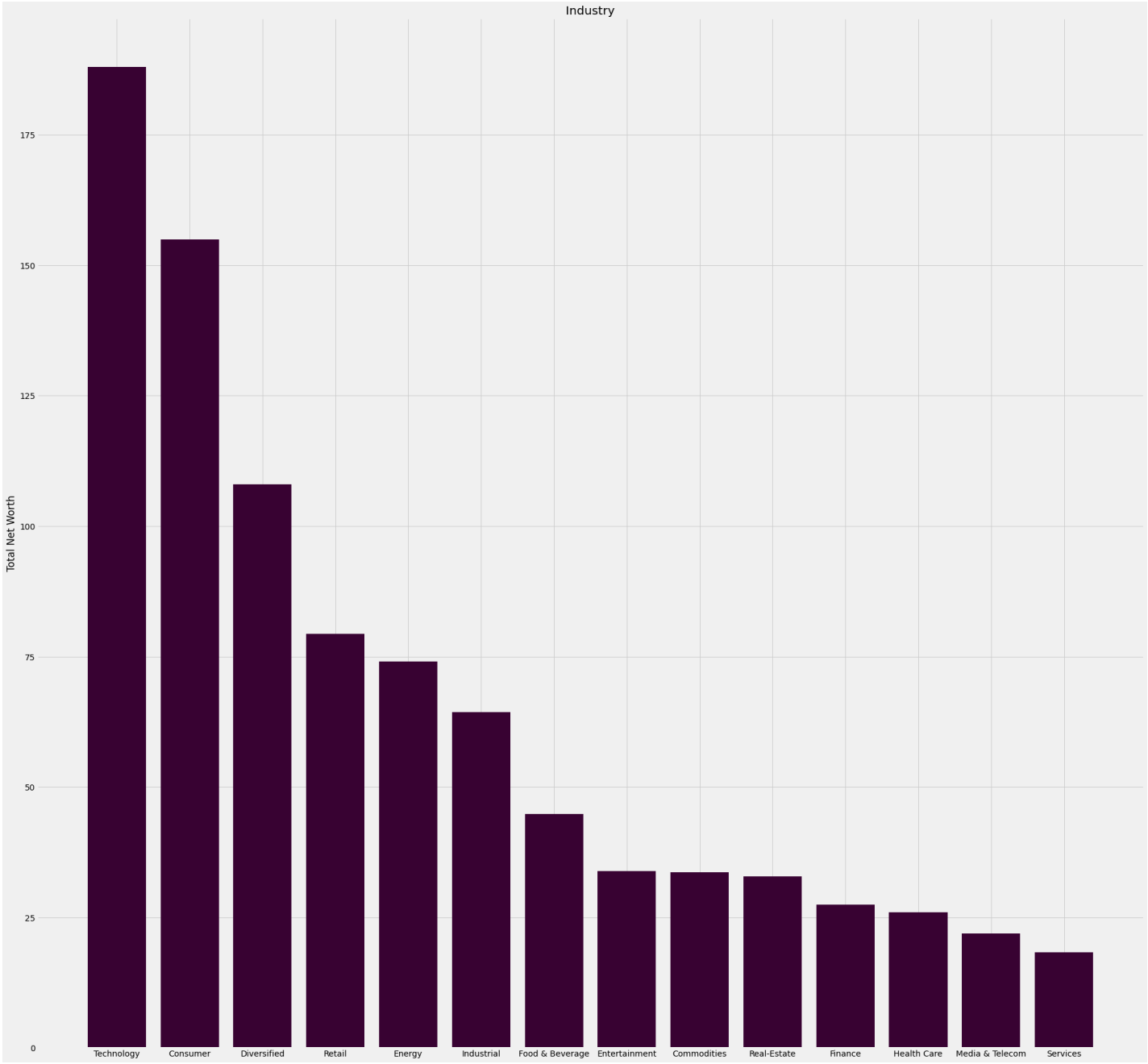
In [32]:
```
india=data[data.Country=='India']
```

In [33]:
```
plt.bar(india['Name'],india['Total Net Worth'])
plt.title('Indian Richies')
plt.ylabel('Billion $')
plt.gcf().set_size_inches(6,6)
plt.xticks(rotation=90)
plt.show()
```



In [34]:

```
plt.bar(data['Industry'],data['Total Net Worth'],color='#380232')
plt.title('Industry')
plt.ylabel('Total Net Worth')
plt.gcf().set_size_inches(30,30)
plt.show()
```



In [35]:

```
india['Industry']
```

Out[35]:

```
12              Energy
16          Industrial
44          Technology
74          Technology
86              Retail
121         Health Care
132             Retail
139             Finance
182         Health Care
184         Commodities
223          Industrial
257    Media & Telecom
305         Commodities
346         Diversified
```

```
395        Health Care
425        Real-Estate
426        Diversified
Name: Industry, dtype: object
```

In [36]:

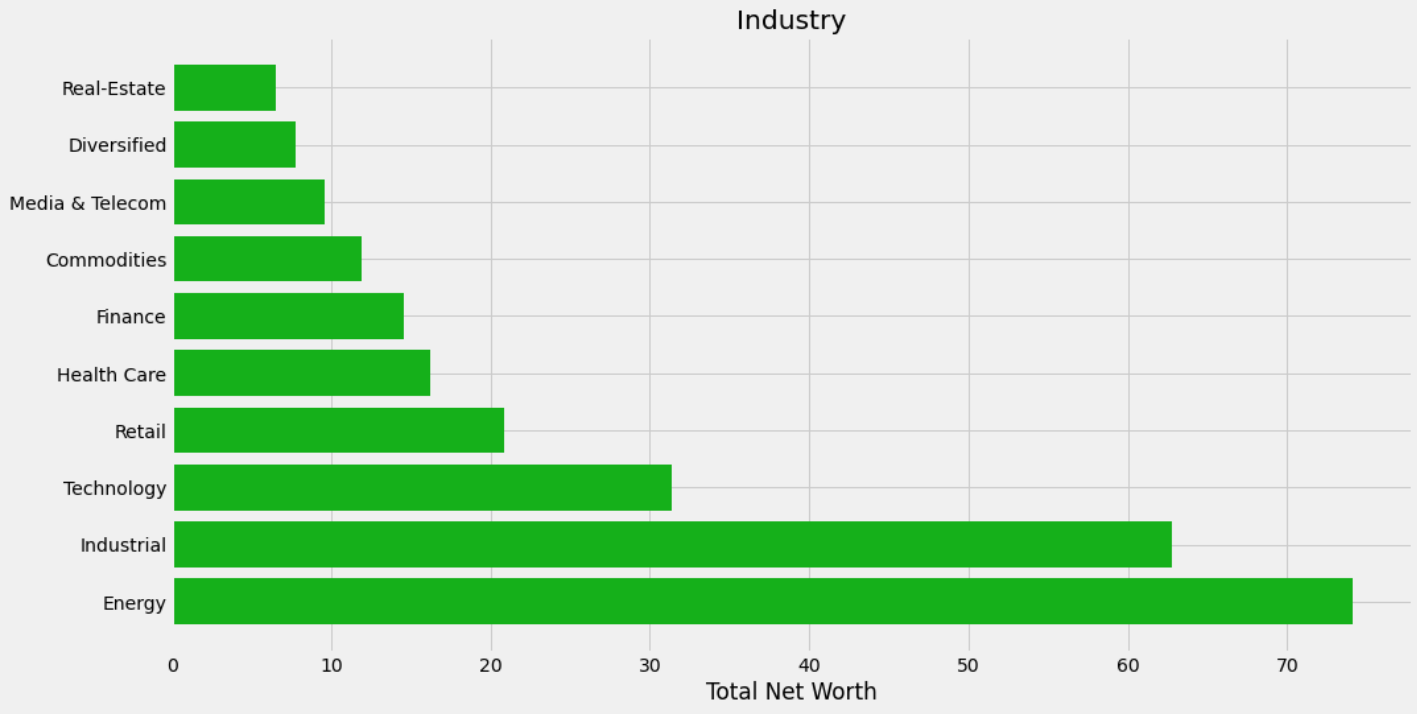
```
india
```

Out[36]:

	Rank	Name	Total Net Worth	$ Last Change	$ YTD Change	Country	Industry
12	13.0	Mukesh Ambani	74.10	402000.0	-261000.0	India	Energy
16	17.0	Gautam Adani	62.80	210000.0	2900000.0	India	Industrial
44	45.0	Azim Premji	31.40	-347000.0	601000.0	India	Technology
74	75.0	Shiv Nadar	23.20	-102000.0	-904000.0	India	Technology
86	87.0	Lakshmi Mittal	20.90	-252000.0	448000.0	India	Retail
121	122.0	Cyrus Poonawalla	16.20	-45000.0	-25500.0	India	Health Care
132	133.0	Radhakishan Damani	15.40	8990.0	481000.0	India	Retail
139	140.0	Uday Kotak	14.60	-359000.0	-175000.0	India	Finance
182	183.0	Dilip Shanghvi	12.10	207000.0	189000.0	India	Health Care
184	185.0	Savitri Jindal	11.90	-266000.0	459000.0	India	Commodities
223	224.0	Kumar Birla	10.40	-62600.0	355000.0	India	Industrial
257	258.0	Sunil Mittal	9.59	-57800.0	696000.0	India	Media & Telecom
305	306.0	Benu Bangur	8.61	81000.0	105000.0	India	Commodities
346	347.0	Nusli Wadia	7.76	57700.0	-128000.0	India	Diversified
395	396.0	Pankaj Patel	6.86	75500.0	158000.0	India	Health Care
425	427.0	K P Singh	6.54	-6280.0	874000.0	India	Real-Estate
426	428.0	Rahul Bajaj	6.53	-21800.0	966000.0	India	Diversified

In [38]:

```
plt.barh(india['Industry'],india['Total Net Worth'],color='#15B01A')
plt.xlabel("Total Net Worth")
plt.title('Industry')
plt.gcf().set_size_inches(15,8)
plt.show()
```



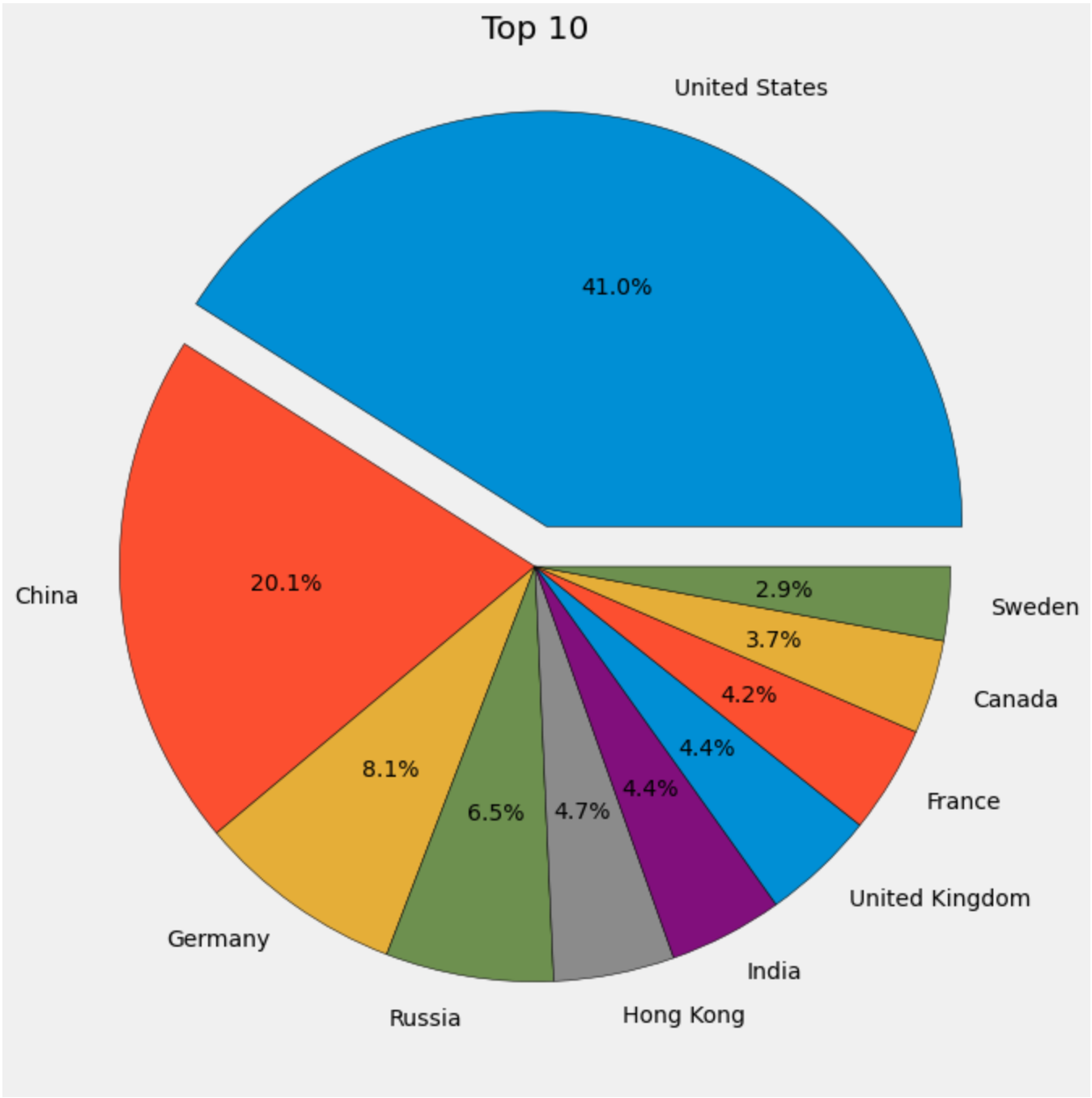
In [40]:
```
display(data["Country"].value_counts())
```

```
United States           157
China                    77
Germany                  31
Russia                   25
Hong Kong                18
India                    17
United Kingdom           17
France                   16
Canada                   14
Sweden                   11
Singapore                10
Australia                 8
Switzerland               8
Japan                     7
Italy                     7
Korea                     7
Mexico                    6
Denmark                   6
Brazil                    5
Taiwan                    4
Indonesia                 4
Ireland                   4
Norway                    3
South Africa              3
Israel                    3
Saudi Arabia              3
Netherlands               2
Austria                   2
Thailand                  2
Malaysia                  2
Colombia                  2
Egypt                     1
Cayman Islands            1
United Arab Emirates      1
Philippines               1
Spain                     1
Argentina                 1
Georgia                   1
Finland                   1
```

```
Cyprus                  1
Romania                 1
Spain                   1
Ukraine                 1
Viet Nam                1
New Zealand             1
Monaco                  1
Chile                   1
Nigeria                 1
Kazakhstan              1
Name: Country, dtype: int64
```

In [43]:

```
#percentage of richies from top 10 countries
country=data["Country"].value_counts().head(10).values
name=data["Country"].value_counts().head(10).index
plt.gcf().set_size_inches(20,11)
plt.pie(country,labels=name,autopct="%1.1f%%",wedgeprops={"edgecolor":"black"},explode=[0.
plt.title('Top 10')
plt.show()
```



In [46]:

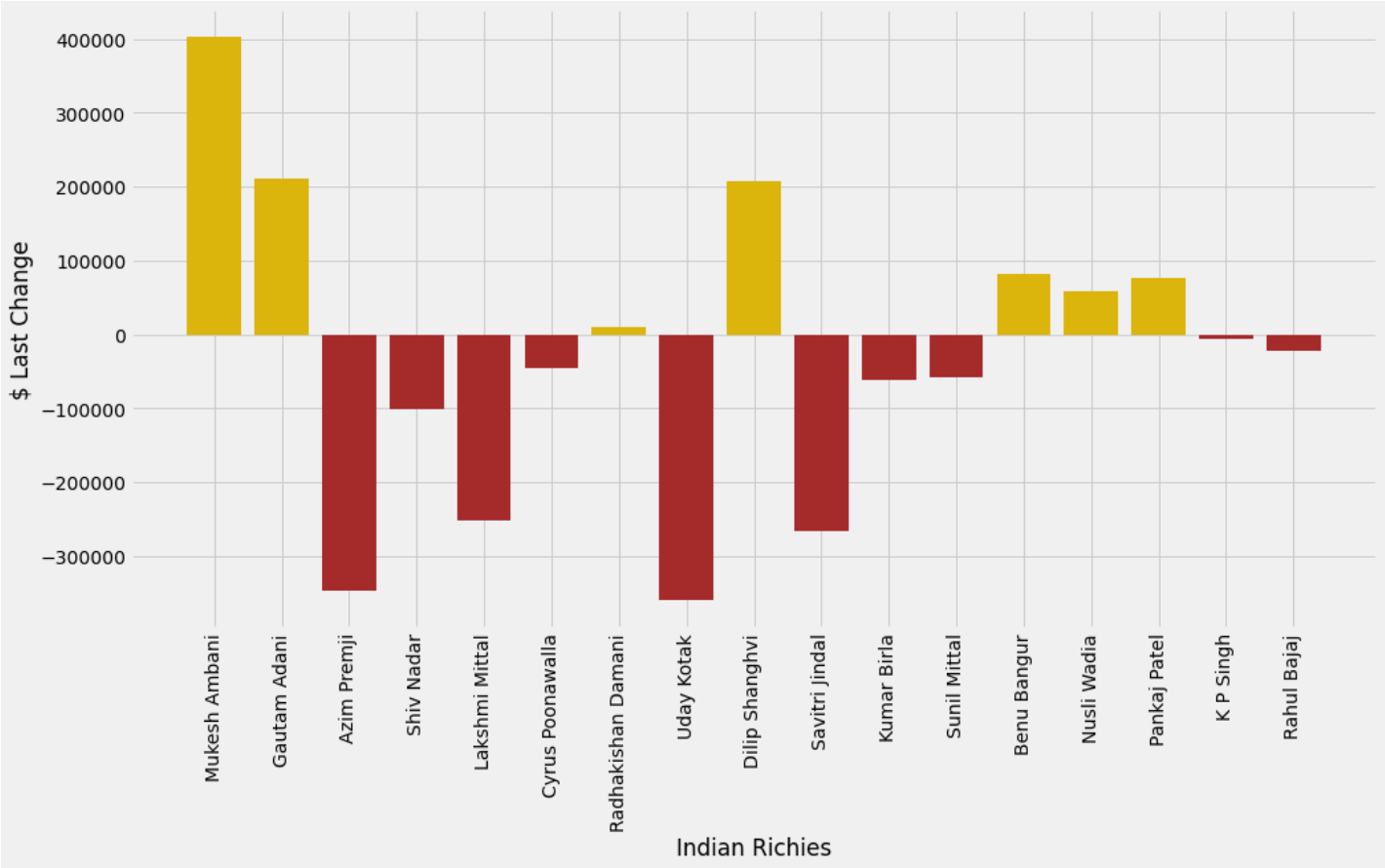
```
data[data.Country=='United States'].head(10)
```

Out[46]:

	Rank	Name	Total Net Worth	$ Last Change	$ YTD Change	Country	Industry
0	1.0	Jeff Bezos	188.0	168000.0	-231000.0	United States	Technology
1	2.0	Elon Musk	170.0	-289000.0	773000.0	United States	Technology
3	4.0	Bill Gates	144.0	-132000.0	1220000.0	United States	Technology
4	5.0	Mark Zuckerberg	114.0	203000.0	1090000.0	United States	Technology
5	6.0	Warren Buffett	108.0	-232000.0	2060000.0	United States	Diversified
6	7.0	Larry Page	104.0	-112000.0	2160000.0	United States	Technology
7	8.0	Sergey Brin	101.0	-106000.0	2080000.0	United States	Technology
8	9.0	Larry Ellison	90.6	-246000.0	1090000.0	United States	Technology
9	10.0	Steve Ballmer	89.1	-342000.0	871000.0	United States	Technology
13	14.0	Charles Koch	64.3	-195000.0	742000.0	United States	Industrial

In [47]:

```
#last change in india
plt.bar(india['Name'],india['$ Last Change'],color=(india['$ Last Change']>0.0).map({True:
plt.xlabel("Indian Richies")
plt.ylabel("$ Last Change")
plt.gcf().set_size_inches(15,8)
plt.xticks(rotation=90)
plt.show()
```



In [ ]: