

```
In [1]: import numpy as np
from numpy import array
from numpy import argmax
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OneHotEncoder
from matplotlib import cm
from matplotlib.colors import ListedColormap, LinearSegmentedColormap
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from PIL import Image
from sklearn import linear_model, metrics
```

```
In [2]: x=pd.read_csv(r"C:\Users\Admin\Downloads\2022_forbes_billionaires.csv")
```

```
In [3]: x
```

```
Out[3]:
```

	rank	name	networth	age	country	source	industry
0	1	Elon Musk	\$219 B	50	United States	Tesla, SpaceX	Automotive
1	2	Jeff Bezos	\$171 B	58	United States	Amazon	Technology
2	3	Bernard Arnault & family	\$158 B	73	France	LVMH	Fashion & Retail
3	4	Bill Gates	\$129 B	66	United States	Microsoft	Technology
4	5	Warren Buffett	\$118 B	91	United States	Berkshire Hathaway	Finance & Investments
...
2595	2578	Jorge Gallardo Ballart	\$1 B	80	Spain	pharmaceuticals	Healthcare
2596	2578	Nari Genomal	\$1 B	82	Philippines	apparel	Fashion & Retail
2597	2578	Ramesh Genomal	\$1 B	71	Philippines	apparel	Fashion & Retail
2598	2578	Sunder Genomal	\$1 B	68	Philippines	garments	Fashion & Retail
2599	2578	Horst-Otto Gerberding	\$1 B	69	Germany	flavors and fragrances	Food & Beverage

2600 rows × 7 columns

```
In [4]: x.shape
```

```
Out[4]: (2600, 7)
```

```
In [5]: x.size
```

```
Out[5]: 18200
```

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

Out[6]:

	rank	name	networth	age	country	source	industry
2595	2578	Jorge Gallardo Ballart	\$1 B	80	Spain	pharmaceuticals	Healthcare
2596	2578	Nari Genomal	\$1 B	82	Philippines	apparel	Fashion & Retail
2597	2578	Ramesh Genomal	\$1 B	71	Philippines	apparel	Fashion & Retail
2598	2578	Sunder Genomal	\$1 B	68	Philippines	garments	Fashion & Retail
2599	2578	Horst-Otto Gerberding	\$1 B	69	Germany	flavors and fragrances	Food & Beverage

In [7]: `x.info()` *#it will show all the information part in your data frame part*

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2600 entries, 0 to 2599
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  -
0   rank        2600 non-null   int64
1   name        2600 non-null   object
2   networth    2600 non-null   object
3   age         2600 non-null   int64
4   country     2600 non-null   object
5   source      2600 non-null   object
6   industry    2600 non-null   object
dtypes: int64(2), object(5)
memory usage: 142.3+ KB
```

In [8]: `x.head()` *#by using head command it will show starting five rows*

Out[8]:

	rank	name	networth	age	country	source	industry
0	1	Elon Musk	\$219 B	50	United States	Tesla, SpaceX	Automotive
1	2	Jeff Bezos	\$171 B	58	United States	Amazon	Technology
2	3	Bernard Arnault & family	\$158 B	73	France	LVMH	Fashion & Retail
3	4	Bill Gates	\$129 B	66	United States	Microsoft	Technology
4	5	Warren Buffett	\$118 B	91	United States	Berkshire Hathaway	Finance & Investments

In [9]: `x.describe()`

Out[9]:

	rank	age
count	2600.000000	2600.000000
mean	1269.570769	64.271923
std	728.146364	13.220607
min	1.000000	19.000000
25%	637.000000	55.000000
50%	1292.000000	64.000000
75%	1929.000000	74.000000
max	2578.000000	100.000000

In [10]: `x.isnull()`

Out[10]:

	rank	name	networth	age	country	source	industry
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
...
2595	False	False	False	False	False	False	False
2596	False	False	False	False	False	False	False
2597	False	False	False	False	False	False	False
2598	False	False	False	False	False	False	False
2599	False	False	False	False	False	False	False

2600 rows × 7 columns

In [11]: `x.dropna()`

Out[11]:

	rank	name	networth	age	country	source	industry
0	1	Elon Musk	\$219 B	50	United States	Tesla, SpaceX	Automotive
1	2	Jeff Bezos	\$171 B	58	United States	Amazon	Technology
2	3	Bernard Arnault & family	\$158 B	73	France	LVMH	Fashion & Retail
3	4	Bill Gates	\$129 B	66	United States	Microsoft	Technology
4	5	Warren Buffett	\$118 B	91	United States	Berkshire Hathaway	Finance & Investments
...
2595	2578	Jorge Gallardo Ballart	\$1 B	80	Spain	pharmaceuticals	Healthcare
2596	2578	Nari Genomal	\$1 B	82	Philippines	apparel	Fashion & Retail
2597	2578	Ramesh Genomal	\$1 B	71	Philippines	apparel	Fashion & Retail
2598	2578	Sunder Genomal	\$1 B	68	Philippines	garments	Fashion & Retail
2599	2578	Horst-Otto Gerberding	\$1 B	69	Germany	flavors and fragrances	Food & Beverage

2600 rows × 7 columns

In [12]: `x.isna()`

Out[12]:

	rank	name	networth	age	country	source	industry
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
...
2595	False	False	False	False	False	False	False
2596	False	False	False	False	False	False	False
2597	False	False	False	False	False	False	False
2598	False	False	False	False	False	False	False
2599	False	False	False	False	False	False	False

2600 rows × 7 columns

```
In [13]: x.isna().sum()
```

```
Out[13]: rank      0
name      0
networth  0
age       0
country   0
source    0
industry  0
dtype: int64
```

```
In [14]: y=x.head(10)
```

```
In [15]: y
```

```
Out[15]:
```

	rank	name	networth	age	country	source	industry
0	1	Elon Musk	\$219 B	50	United States	Tesla, SpaceX	Automotive
1	2	Jeff Bezos	\$171 B	58	United States	Amazon	Technology
2	3	Bernard Arnault & family	\$158 B	73	France	LVMH	Fashion & Retail
3	4	Bill Gates	\$129 B	66	United States	Microsoft	Technology
4	5	Warren Buffett	\$118 B	91	United States	Berkshire Hathaway	Finance & Investments
5	6	Larry Page	\$111 B	49	United States	Google	Technology
6	7	Sergey Brin	\$107 B	48	United States	Google	Technology
7	8	Larry Ellison	\$106 B	77	United States	software	Technology
8	9	Steve Ballmer	\$91.4 B	66	United States	Microsoft	Technology
9	10	Mukesh Ambani	\$90.7 B	64	India	diversified	Diversified

In [16]: `x.head`

Out[16]: <bound method NDFrame.head of

	rank	name	network	age	country	source	industry
0	1	Elon Musk	\$219 B	50	United States	Tesla, SpaceX	Automotive
1	2	Jeff Bezos	\$171 B	58	United States	Amazon	Technology
2	3	Bernard Arnault & family	\$158 B	73	France	LVMH	Fashion & Retail
3	4	Bill Gates	\$129 B	66	United States	Microsoft	Technology
4	5	Warren Buffett	\$118 B	91	United States	Berkshire Hathaway	Finance & Investments
...
2595	2578	Jorge Gallardo Ballart	\$1 B	80	Spain	pharmaceuticals	Healthcare
2596	2578	Nari Genomal	\$1 B	82	Philippines	apparel	Fashion & Retail
2597	2578	Ramesh Genomal	\$1 B	71	Philippines	apparel	Fashion & Retail
2598	2578	Sunder Genomal	\$1 B	68	Philippines	garments	Fashion & Retail
2599	2578	Horst-Otto Gerberding	\$1 B	69	Germany	flavors and fragrances	Food & Beverage

[2600 rows x 7 columns]>

In [17]: `pd.isnull(x)` *#it will show all false value*

Out[17]:

	rank	name	network	age	country	source	industry
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
...
2595	False	False	False	False	False	False	False
2596	False	False	False	False	False	False	False
2597	False	False	False	False	False	False	False
2598	False	False	False	False	False	False	False
2599	False	False	False	False	False	False	False

2600 rows × 7 columns

In [18]: `pd.isnull(x).sum()` *#it will check the null values #by using sum() method*

Out[18]:

rank	0
name	0
networth	0
age	0
country	0
source	0
industry	0
dtype:	int64

In [19]: `x.shape`

Out[19]: (2600, 7)

In [20]: `data_set=[['Elon Musk',50],['Jeff Bezos',58],['Bill Gates',66],['Warron Buffett',91]]`
`x_test=pd.DataFrame(data_set,columns=['Name', 'Age'])` *#creating the data frame*

In [21]: `x_test`

Out[21]:

	Name	Age
0	Elon Musk	50
1	Jeff Bezos	58
2	Bill Gates	66
3	Warron Buffett	91

In [22]: `x_test.dropna(inplace=True)`

In [23]: `x_test`

Out[23]:

	Name	Age
0	Elon Musk	50
1	Jeff Bezos	58
2	Bill Gates	66
3	Warron Buffett	91

In [24]: `x.columns` *#by using this command it will show the columns name*

Out[24]: Index(['rank', 'name', 'networth', 'age', 'country', 'source', 'industry'], dtype='object')

In [25]: `x.drop(['industry'],axis=1,inplace=True)`

In [26]:

x

Out[26]:

	rank	name	networth	age	country	source
0	1	Elon Musk	\$219 B	50	United States	Tesla, SpaceX
1	2	Jeff Bezos	\$171 B	58	United States	Amazon
2	3	Bernard Arnault & family	\$158 B	73	France	LVMH
3	4	Bill Gates	\$129 B	66	United States	Microsoft
4	5	Warren Buffett	\$118 B	91	United States	Berkshire Hathaway
...
2595	2578	Jorge Gallardo Ballart	\$1 B	80	Spain	pharmaceuticals
2596	2578	Nari Genomal	\$1 B	82	Philippines	apparel
2597	2578	Ramesh Genomal	\$1 B	71	Philippines	apparel
2598	2578	Sunder Genomal	\$1 B	68	Philippines	garments
2599	2578	Horst-Otto Gerberding	\$1 B	69	Germany	flavors and fragrances

2600 rows × 6 columns

In [27]: `x.rename(columns={'source':'Global Source'})` *#rename my columns name by using*

Out[27]:

	rank	name	networth	age	country	Global Source
0	1	Elon Musk	\$219 B	50	United States	Tesla, SpaceX
1	2	Jeff Bezos	\$171 B	58	United States	Amazon
2	3	Bernard Arnault & family	\$158 B	73	France	LVMH
3	4	Bill Gates	\$129 B	66	United States	Microsoft
4	5	Warren Buffett	\$118 B	91	United States	Berkshire Hathaway
...
2595	2578	Jorge Gallardo Ballart	\$1 B	80	Spain	pharmaceuticals
2596	2578	Nari Genomal	\$1 B	82	Philippines	apparel
2597	2578	Ramesh Genomal	\$1 B	71	Philippines	apparel
2598	2578	Sunder Genomal	\$1 B	68	Philippines	garments
2599	2578	Horst-Otto Gerberding	\$1 B	69	Germany	flavors and fragrances

2600 rows × 6 columns

In [28]: `x.describe()` *#describe syntax it will show the description in the data (i.e mean, std, min, max)*

Out[28]:

	rank	age
count	2600.000000	2600.000000
mean	1269.570769	64.271923
std	728.146364	13.220607
min	1.000000	19.000000
25%	637.000000	55.000000
50%	1292.000000	64.000000
75%	1929.000000	74.000000
max	2578.000000	100.000000

In [29]: `x[['name', 'networth', 'source']].describe()` *#it will show the column name which has the most unique values*

Out[29]:

	name	networth	source
count	2600	2600	2600
unique	2598	228	895
top	Wang Yanqing & family	\$1.5 B	real estate
freq	2	147	148

In [30]: `t=x.sample(10)`

In [31]: `t`

Out[31]:

	rank	name	networth	age	country	source
2419	2324	Lucio Tan	\$1.2 B	87	Philippines	diversified
1973	1929	Feng Yuxia	\$1.5 B	57	China	pharmaceuticals
861	851	Aerin Lauder	\$3.5 B	51	United States	cosmetics
1273	1238	Charles Munger	\$2.5 B	98	United States	Berkshire Hathaway
591	586	Leng Youbin	\$4.8 B	53	China	infant formula
1826	1818	Chen Shibin	\$1.6 B	55	China	quartz products
1283	1238	Suh Kyung-bae	\$2.5 B	59	South Korea	cosmetics
150	151	Ricardo Salinas Pliego & family	\$12.4 B	66	Mexico	retail, media
1349	1341	Chu Lam Yiu	\$2.3 B	52	Hong Kong	flavorings
286	284	Patrick Ryan	\$8 B	84	United States	insurance

```
In [32]: ab=sns.countplot(x='country',data=t)
for bars in ab.containers:
    ab.bar_label(bars)
```

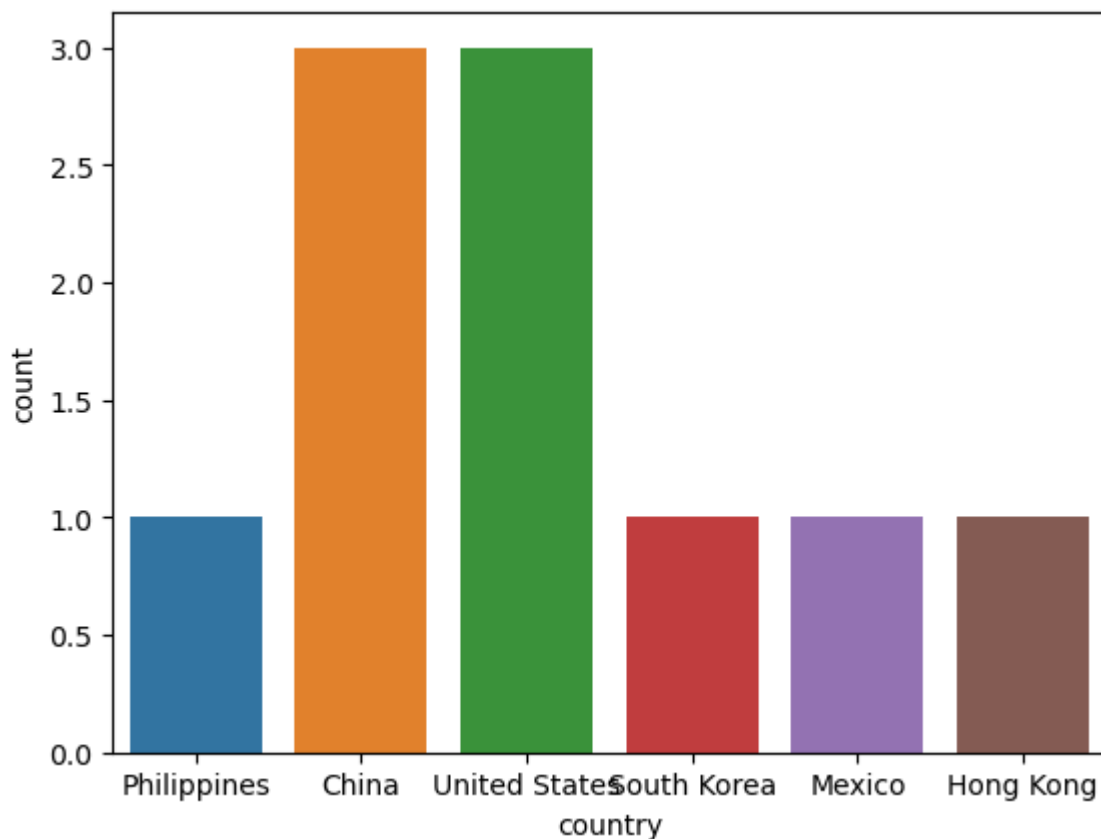
AttributeError

Traceback (most recent call last)

Cell In[32], line 2

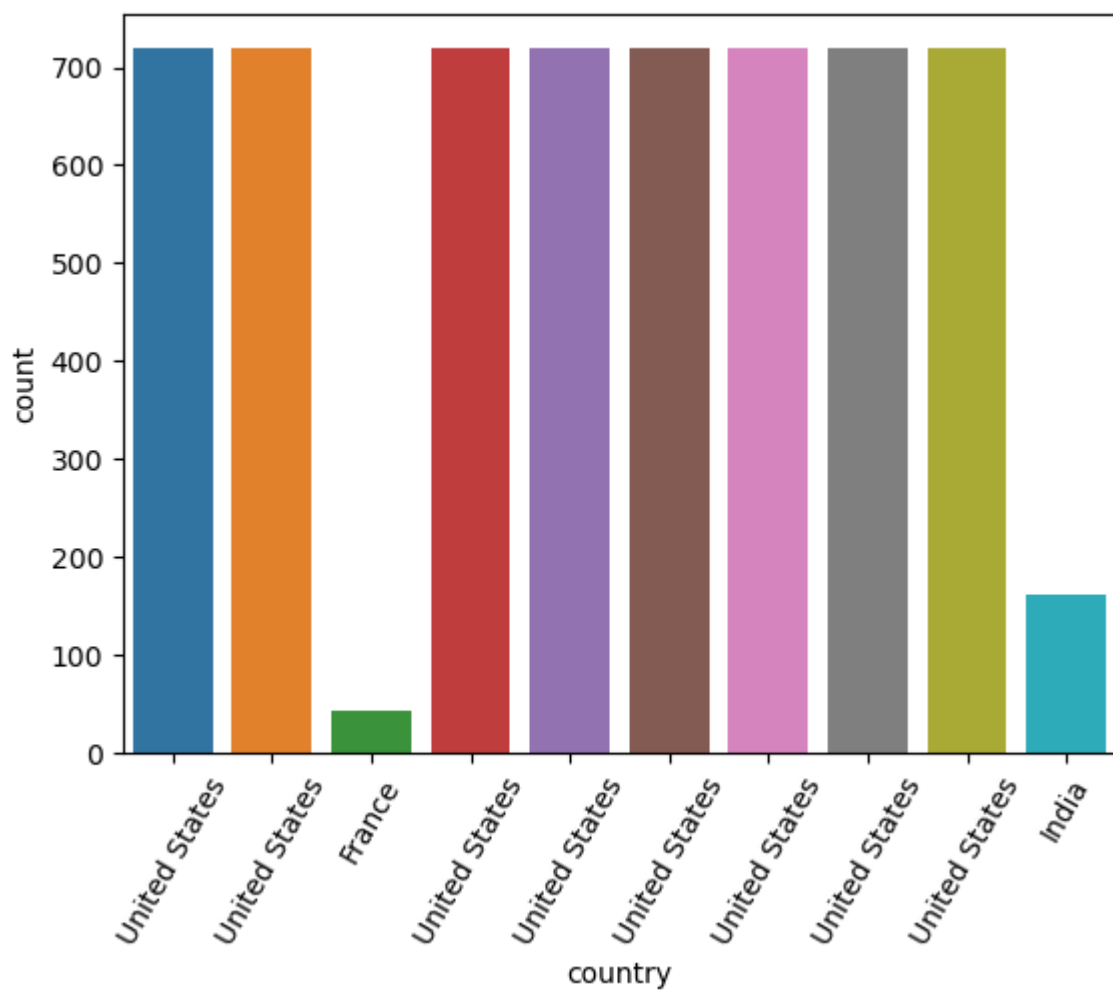
```
1 ab=sns.countplot(x='country',data=t)
----> 2 for bars in ab.containers:
3     ab.bar_label(bars)
```

AttributeError: 'Axes' object has no attribute 'containers'



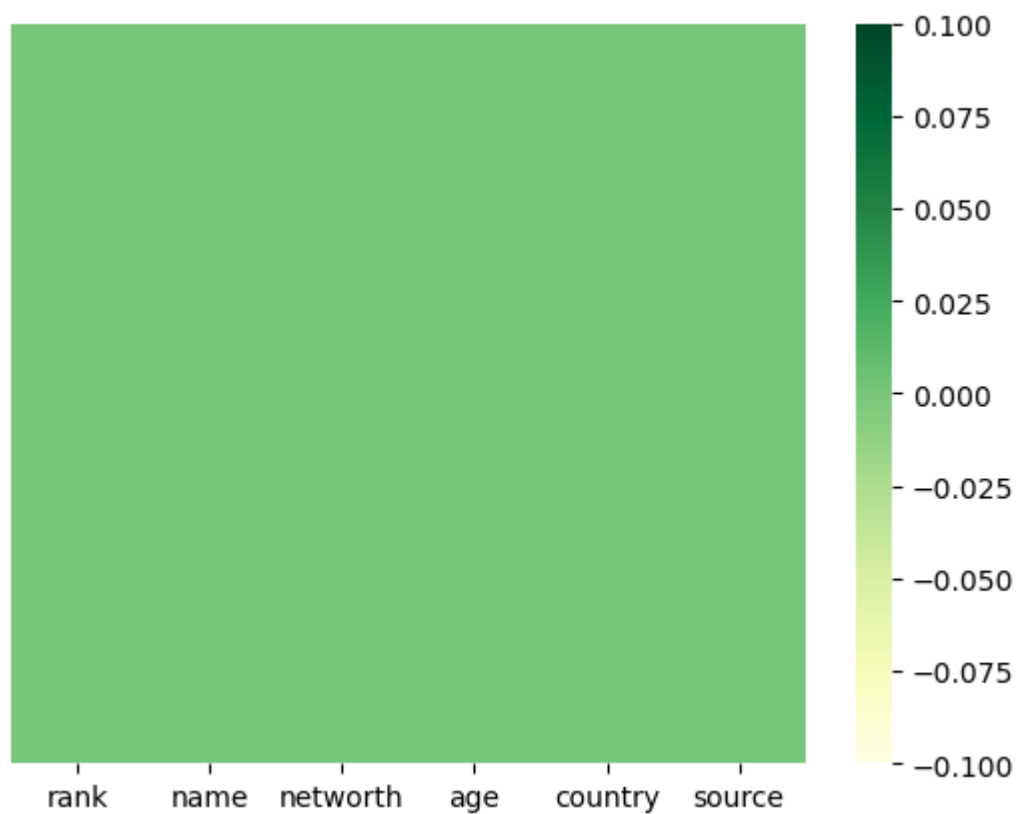
```
In [33]: ab=sns.countplot(x='country',data=x, order=x.country.iloc[:10])  
plt.xticks(rotation=60)
```

```
Out[33]: (array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),  
[Text(0, 0, 'United States'),  
Text(1, 0, 'United States'),  
Text(2, 0, 'France'),  
Text(3, 0, 'United States'),  
Text(4, 0, 'United States'),  
Text(5, 0, 'United States'),  
Text(6, 0, 'United States'),  
Text(7, 0, 'United States'),  
Text(8, 0, 'United States'),  
Text(9, 0, 'India')])
```



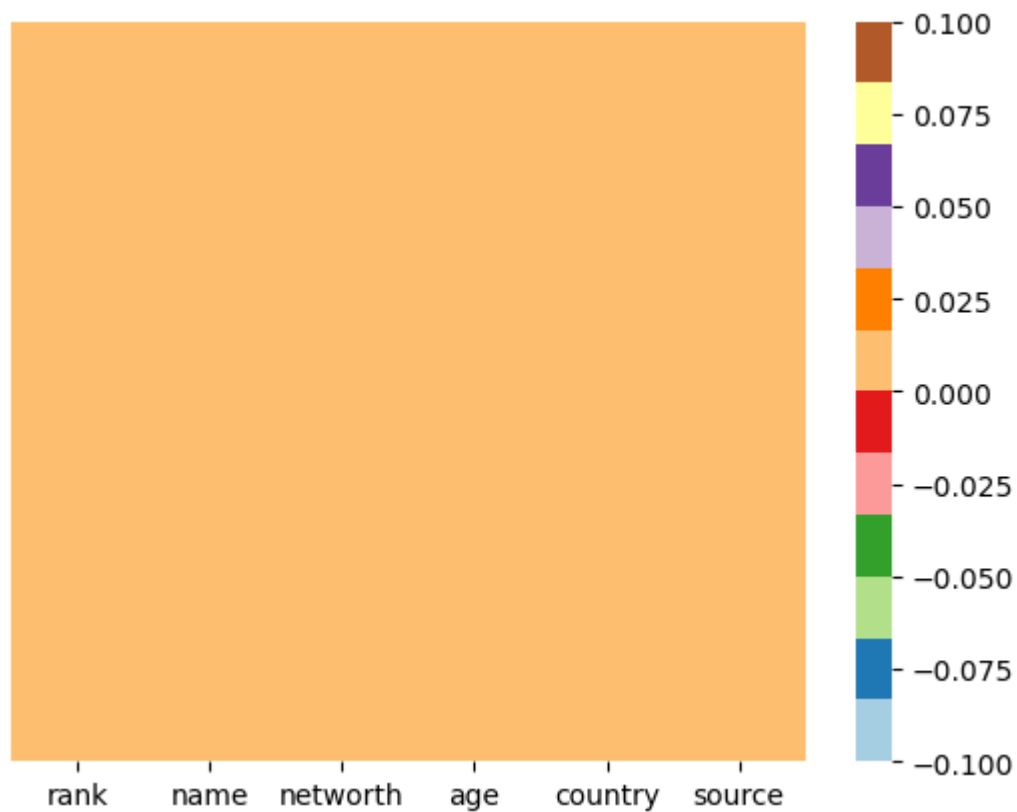
```
In [34]: sns.heatmap(x.isnull(),yticklabels=False,cmap="YlGn")
```

```
Out[34]: <Axes: >
```



```
In [35]: sns.heatmap(x.isnull(),yticklabels=False,cmap="Paired")
```

```
Out[35]: <Axes: >
```



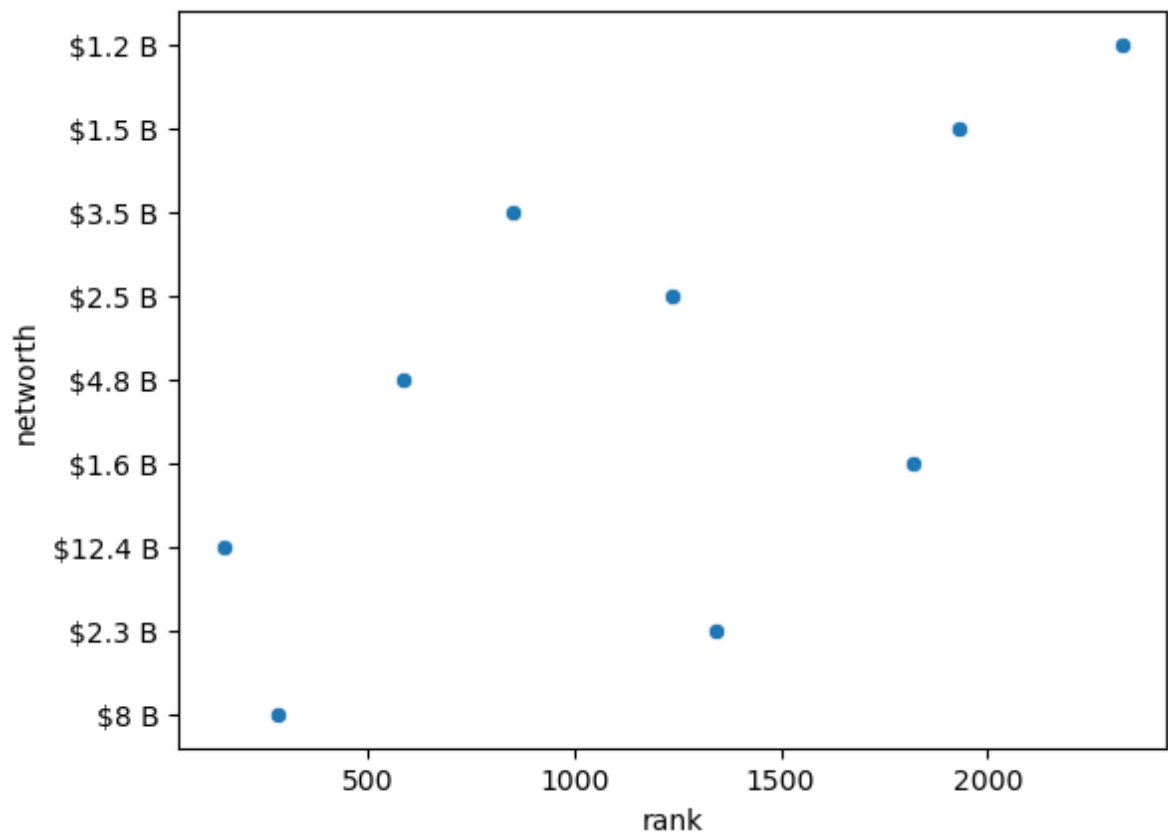
```
Out[36]: <Axes: xlabel='networth', ylabel='age'>
```



```
Out[37]: <Axes: xlabel='name', ylabel='source'>
```

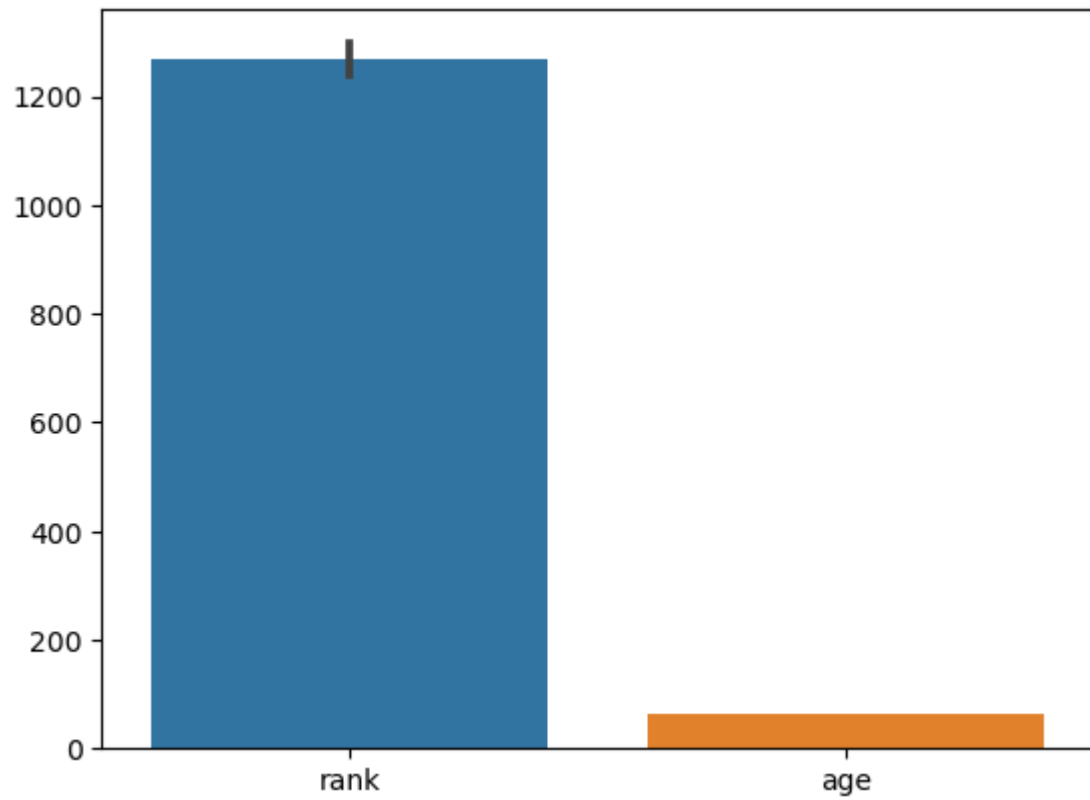


```
In [57]: sns.scatterplot(x='rank',y='networth',data=t);
```



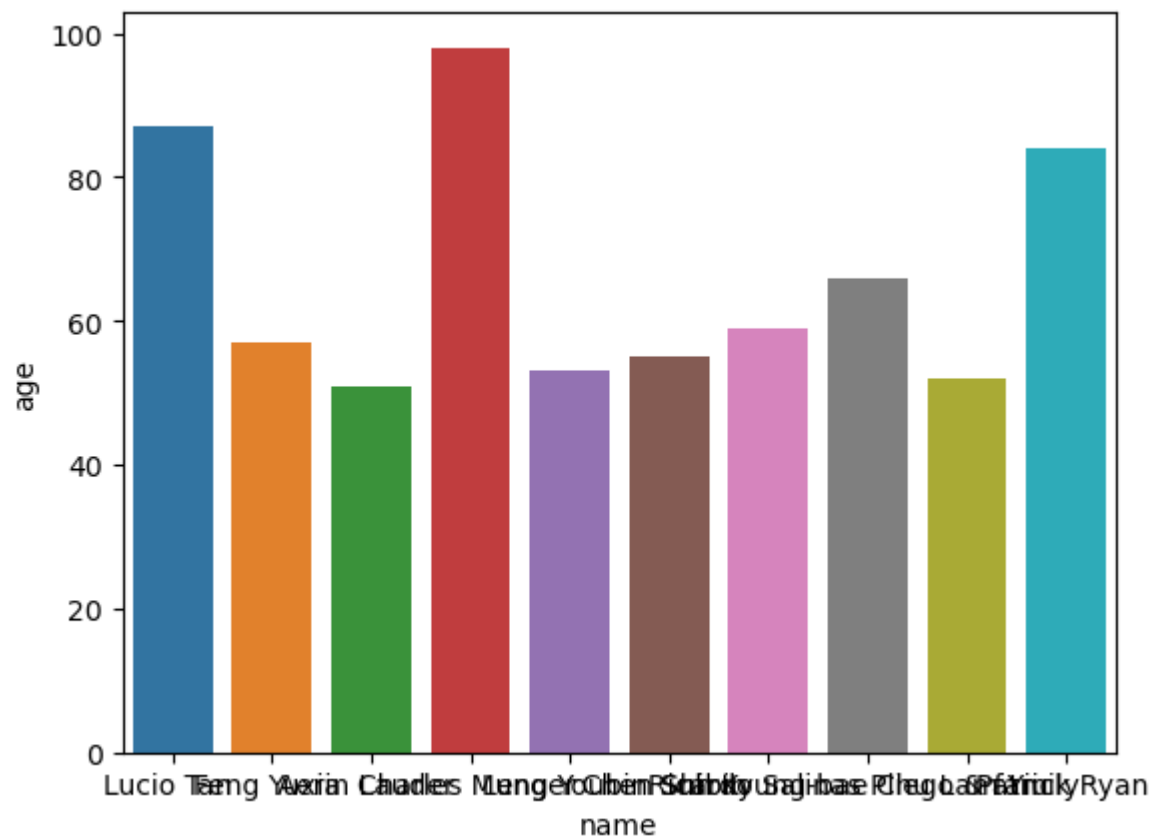
```
In [38]: sns.barplot(x)
```

```
Out[38]: <Axes: >
```



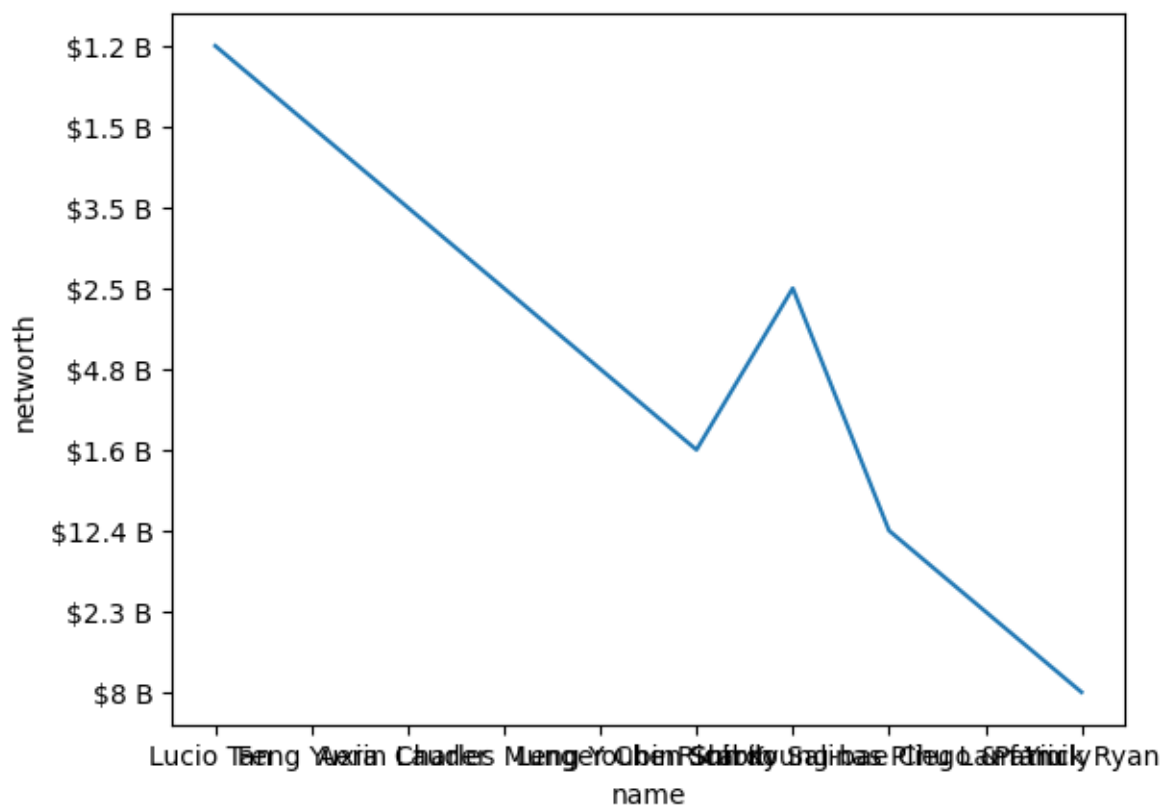

```
In [39]: sns.barplot(x="name",y="age",data=t)
```

```
Out[39]: <Axes: xlabel='name', ylabel='age'>
```



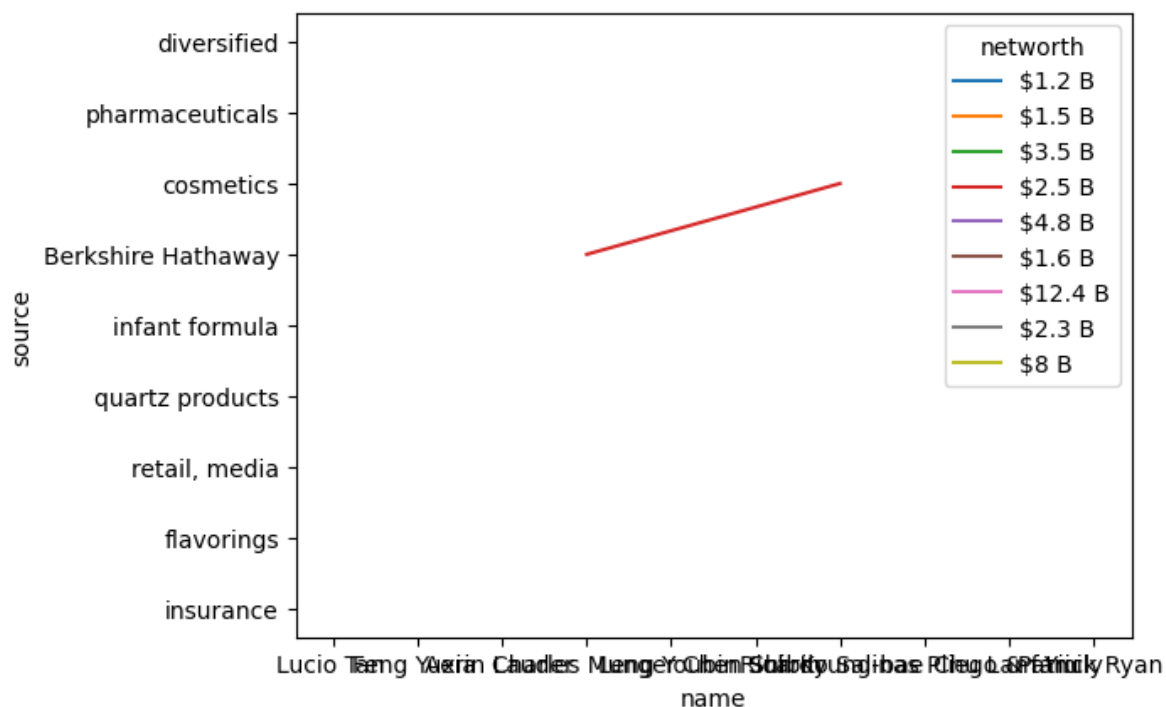
In [40]: `sns.lineplot(x=t.name,y=t.networth.iloc[:10])`

Out[40]: `<Axes: xlabel='name', ylabel='networth'>`



In [64]: `sns.lineplot(x=t.name,y=t.source,data=t,hue="networth")`

Out[64]: `<Axes: xlabel='name', ylabel='source'>`



```
In [68]: sns.regplot(x=t.name,y=t.source,data=t)
```

UFuncTypeError

Traceback (most recent call last)

Cell In[68], line 1

```
----> 1 sns.regplot(x=t.name,y=t.source,data=t)
```

File ~\anaconda3\Lib\site-packages\seaborn\regression.py:759, in regplot(data, x, y, x_estimator, x_bins, x_ci, scatter, fit_reg, ci, n_boot, units, see_d, order, logistic, lowess, robust, logx, x_partial, y_partial, truncate, dropna, x_jitter, y_jitter, label, color, marker, scatter_kws, line_kws, ax)

```
757 scatter_kws["marker"] = marker
```

```
758 line_kws = {} if line_kws is None else copy.copy(line_kws)
```

```
--> 759 plotter.plot(ax, scatter_kws, line_kws)
```

```
760 return ax
```

File ~\anaconda3\Lib\site-packages\seaborn\regression.py:368, in _RegressionPlotter.plot(self, ax, scatter_kws, line_kws)

```
365 self.scatterplot(ax, scatter_kws)
```

```
367 if self.fit_reg:
```

```
--> 368 self.lineplot(ax, line_kws)
```

```
370 # Label the axes
```

```
371 if hasattr(self.x, "name"):
```

File ~\anaconda3\Lib\site-packages\seaborn\regression.py:413, in _RegressionPlotter.lineplot(self, ax, kws)

```
411 """Draw the model."""
```

```
412 # Fit the regression model
```

```
--> 413 grid, yhat, err_bands = self.fit_regression(ax)
```

```
414 edges = grid[0], grid[-1]
```

```
416 # Get set default aesthetics
```

File ~\anaconda3\Lib\site-packages\seaborn\regression.py:199, in _RegressionPlotter.fit_regression(self, ax, x_range, grid)

```
197 else:
```

```
198 x_min, x_max = ax.get_xlim()
```

```
--> 199 grid = np.linspace(x_min, x_max, 100)
```

```
200 ci = self.ci
```

```
202 # Fit the regression
```

File ~\anaconda3\Lib\site-packages\numpy\core\function_base.py:129, in linspace(start, stop, num, endpoint, retstep, dtype, axis)

```
125 div = (num - 1) if endpoint else num
```

```
127 # Convert float/complex array scalars to float, gh-3504
```

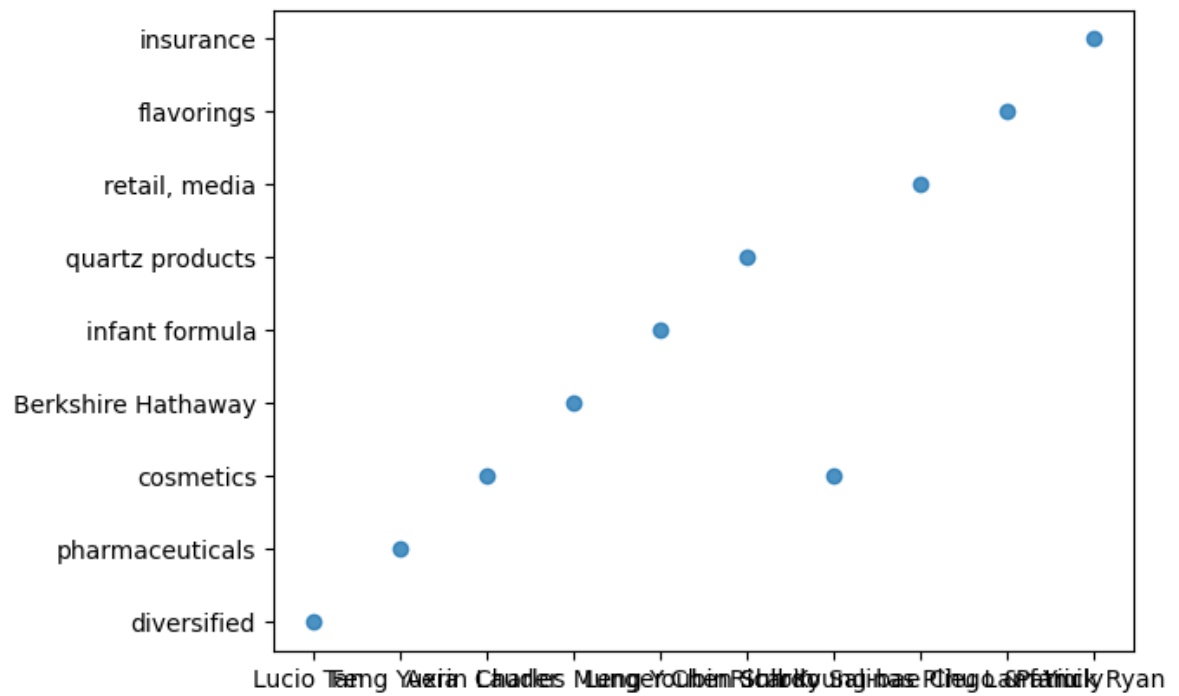
```
128 # and make sure one can use variables that have an __array_interface__  
_, gh-6634
```

```
--> 129 start = asanyarray(start) * 1.0
```

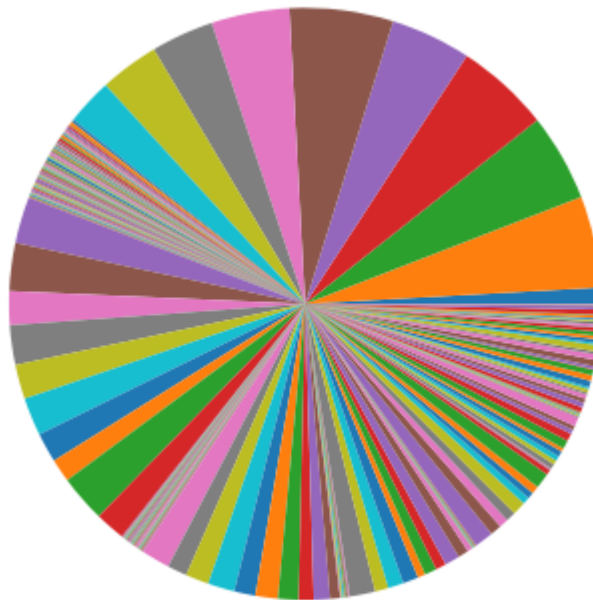
```
130 stop = asanyarray(stop) * 1.0
```

```
132 dt = result_type(start, stop, float(num))
```

UFuncTypeError: ufunc 'multiply' did not contain a loop with signature matching types (dtype('<U13'), dtype('float64')) -> None



```
In [41]: x=x.groupby('networth')['networth'].count()
plt.pie(x)
plt.show()
```



```
plt.pie(x, labels=x.index, autopct="%.2f%%")
plt.show()
```

