Project 5: Forbes Richest People Analysis

1. Load the file

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

dataset=pd.read_csv("/content/archive(3).zip")
```

2. Print first 5 rows of data

Double-click (or enter) to edit

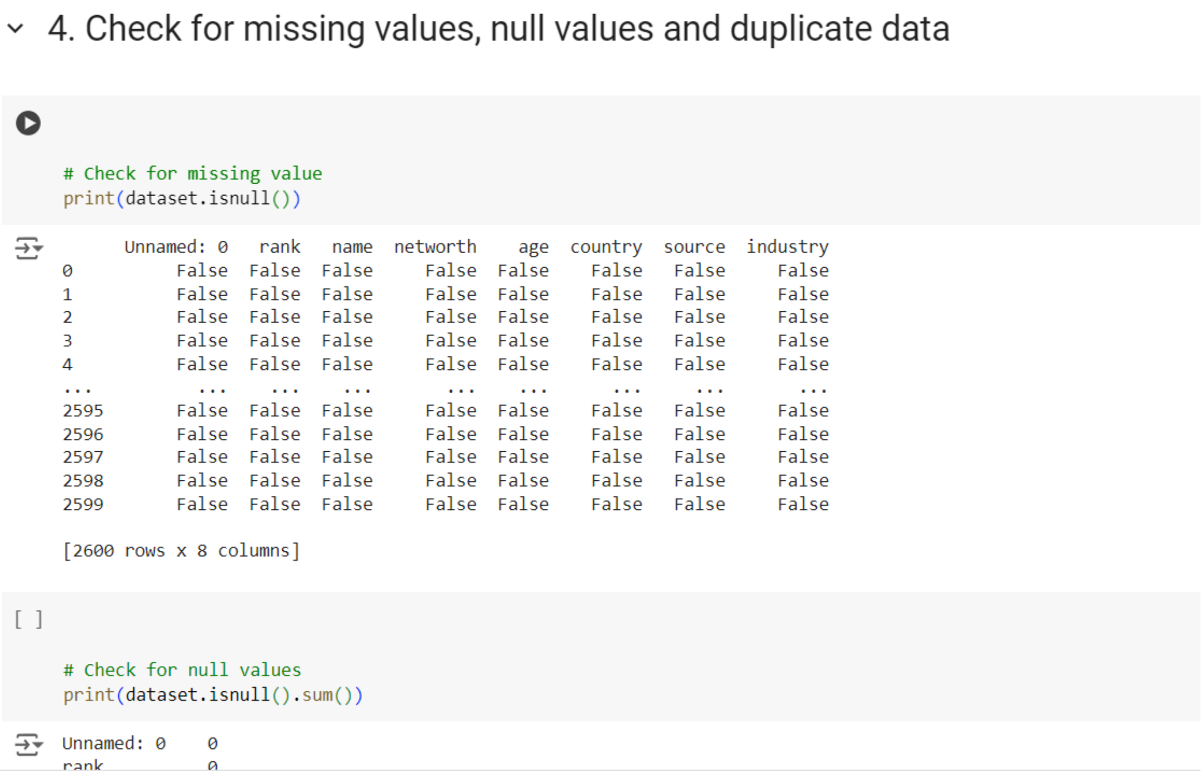
```
[ ]
dataset.head()
```

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

3. Print last 5 rows of data

```
dataset.tail()
```

```
      Unnamed: 0  rank                  name  networth  age      country                 source         industry
2595        2595  2578  Jorge Gallardo Ballart     $1 B   80        Spain        pharmaceuticals       Healthcare
2596        2596  2578          Nari Genomal      $1 B   82  Philippines                apparel  Fashion & Retail
2597        2597  2578        Ramesh Genomal      $1 B   71  Philippines                apparel  Fashion & Retail
2598        2598  2578        Sunder Genomal      $1 B   68  Philippines               garments  Fashion & Retail
2599        2599  2578  Horst-Otto Gerberding     $1 B   69      Germany  flavors and fragrances  Food & Beverage
```



4. Check for missing values, null values and duplicate data

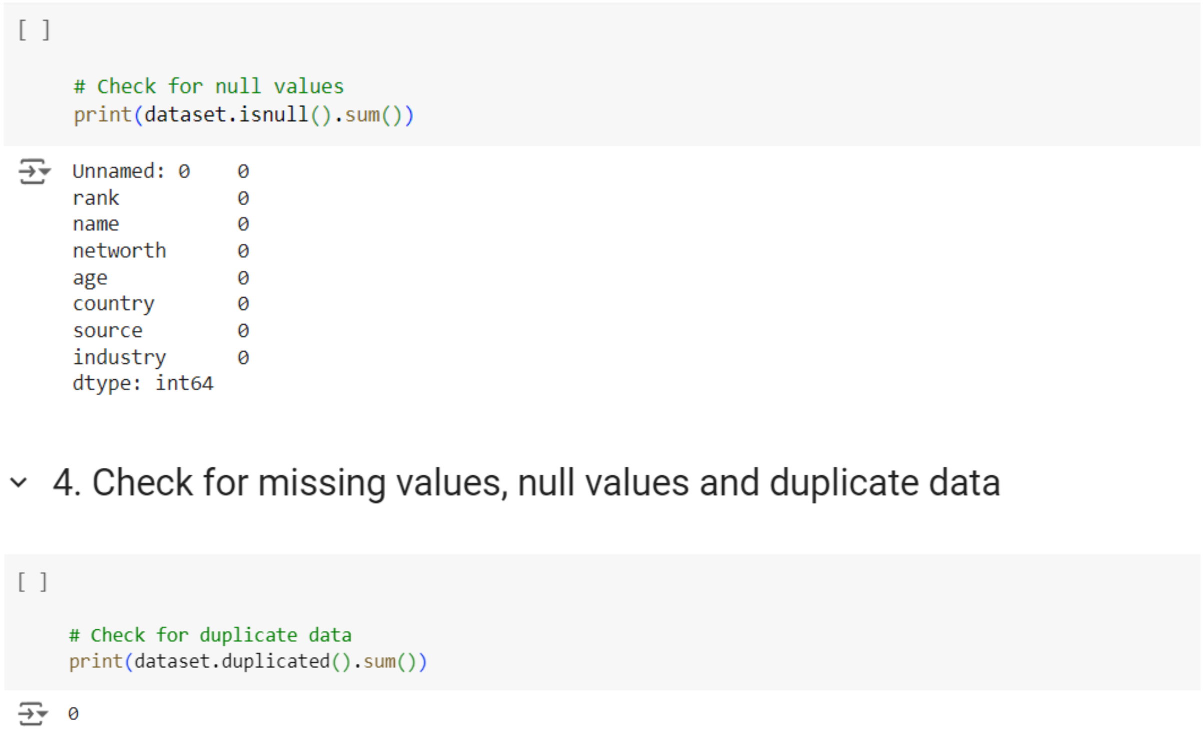
```
# Check for missing value
print(dataset.isnull())
```

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

[2600 rows x 8 columns]
```

[ ]

```
# Check for null values
print(dataset.isnull().sum())
```

```
Unnamed: 0    0
rank          0
```



[ ]

```
# Check for null values
print(dataset.isnull().sum())
```

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

4. Check for missing values, null values and duplicate data

[ ]

```
# Check for duplicate data
print(dataset.duplicated().sum())
```

```
0
```

5. Get some info about the data

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2600 entries, 0 to 2599
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   Unnamed: 0  2600 non-null   int64 
 1   rank        2600 non-null   int64 
 2   name        2600 non-null   object
 3   networth    2600 non-null   object
 4   age         2600 non-null   int64 
 5   country     2600 non-null   object
 6   source      2600 non-null   object
 7   industry    2600 non-null   object
dtypes: int64(3), object(5)
memory usage: 162.6+ KB
```

6. Get some description about the data

```
dataset.describe()
```

	Unnamed: 0	rank	age
count	2600.000000	2600.000000	2600.000000
mean	1299.500000	1269.570769	64.271923
std	750.699674	728.146364	13.220607
min	0.000000	1.000000	19.000000
25%	649.750000	637.000000	55.000000
50%	1299.500000	1292.000000	64.000000
75%	1949.250000	1929.000000	74.000000
max	2599.000000	2578.000000	100.000000

7. Get the shape of the data

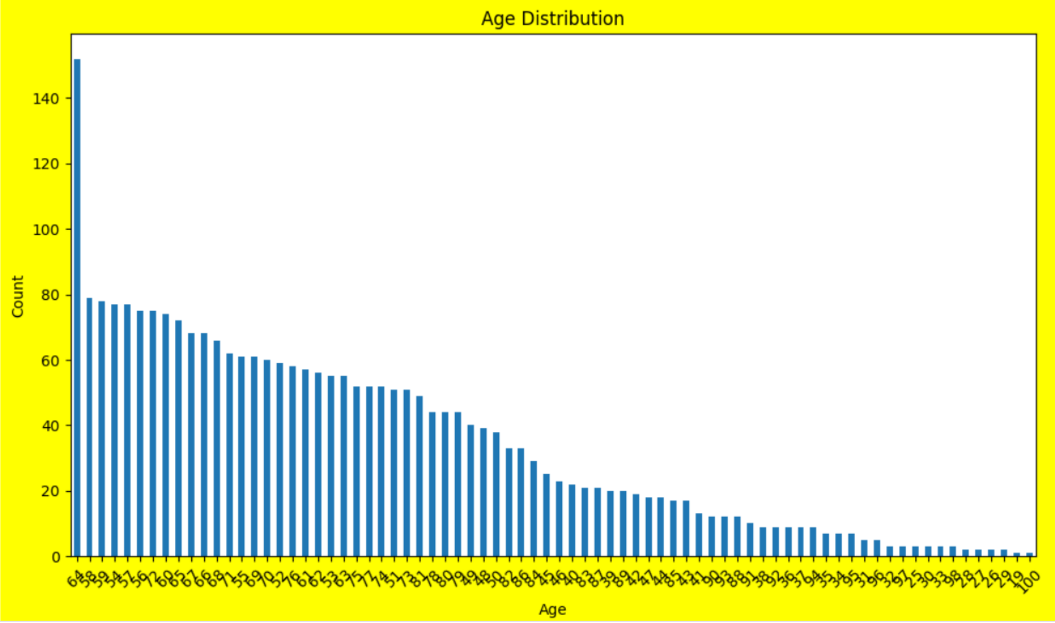
[3]

```
dataset.shape
```

```
(2600, 8)
```

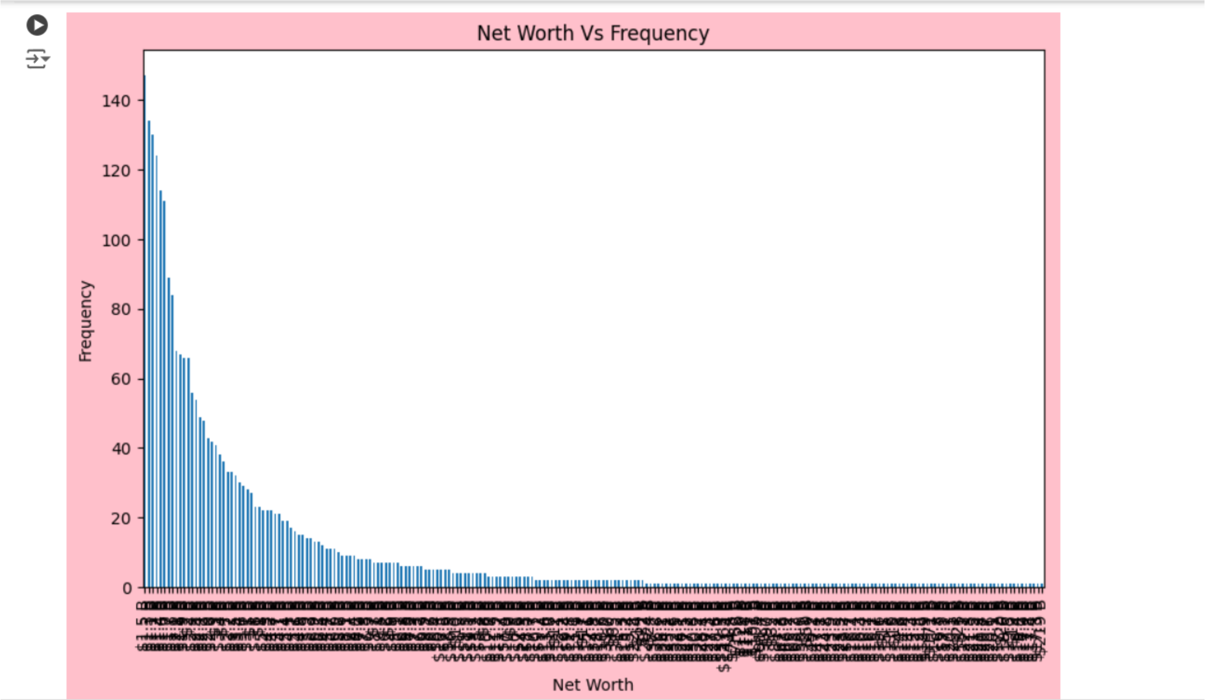
1. Show the Age disribution among the data using bar plot

```
age_distribution = dataset['age'].value_counts()
plt.figure(figsize=(10, 6), dpi=100, facecolor='yellow', edgecolor='black')
age_distribution.plot(kind='bar')
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

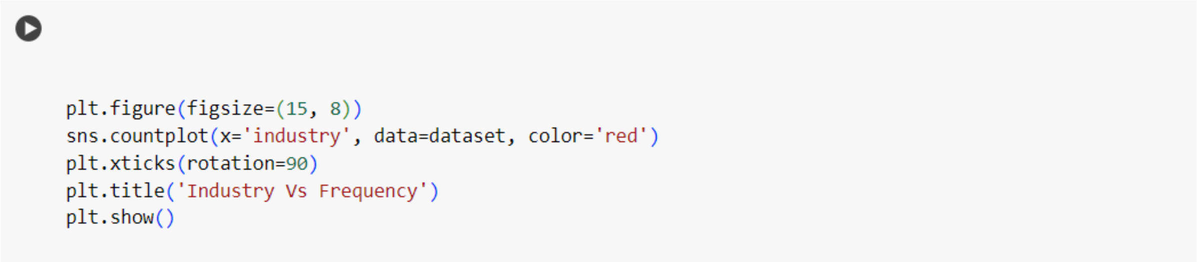


2. Show the Net Worth Vs Frequency using bar plot

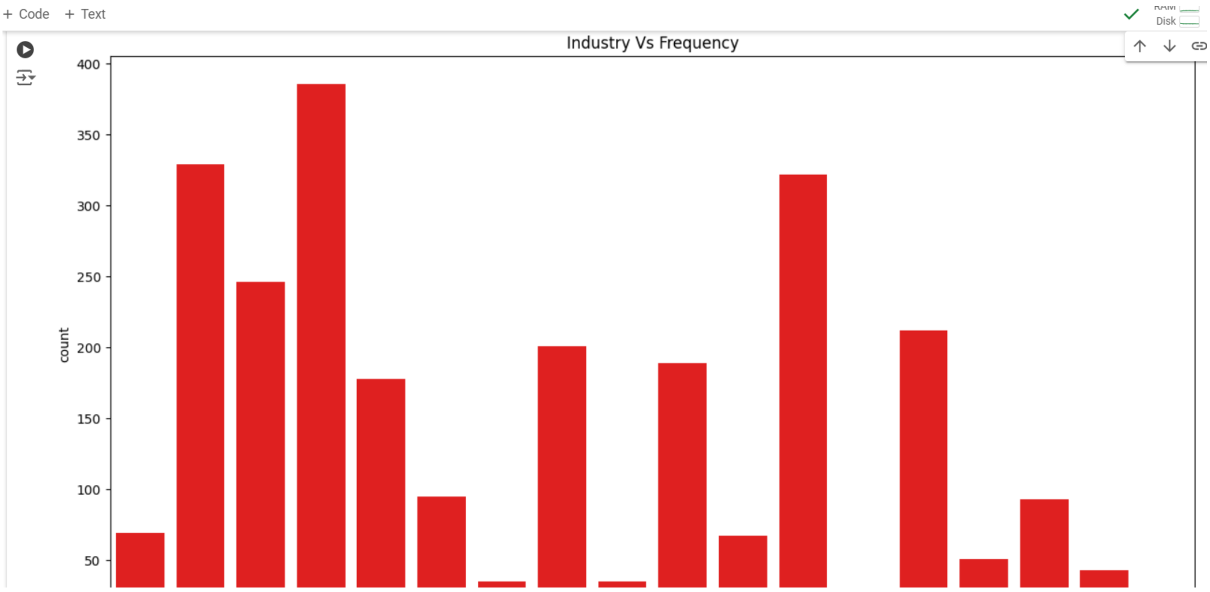
```
net_worth_frequency = dataset['networth'].value_counts()
plt.figure(figsize=(10, 6), dpi=100, facecolor='pink', edgecolor='black')
net_worth_frequency.plot(kind='bar')
plt.title('Net Worth Vs Frequency')
plt.xlabel('Net Worth')
plt.ylabel('Frequency')
plt.show()
```

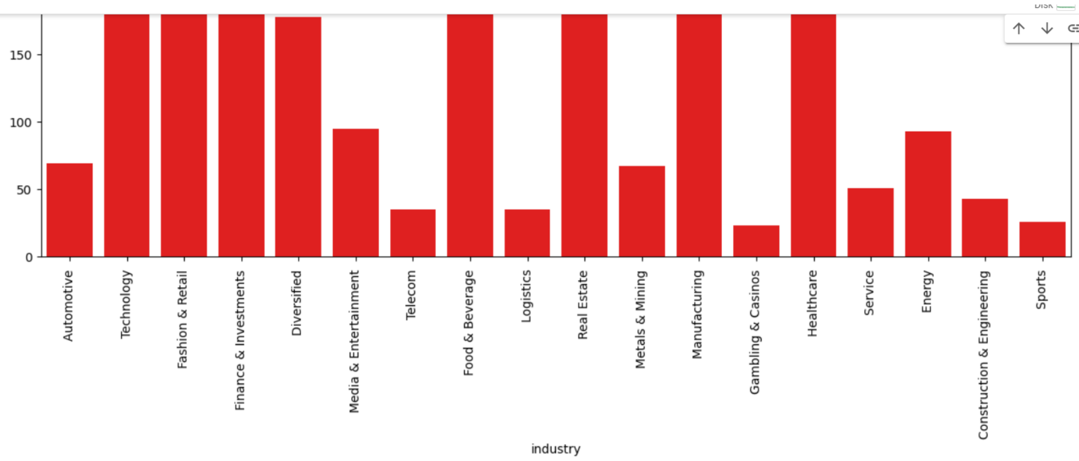


3. Show Industry Vs Frequency using bar plot



```
plt.figure(figsize=(15, 8))
sns.countplot(x='industry', data=dataset, color='red')
plt.xticks(rotation=90)
plt.title('Industry Vs Frequency')
plt.show()
```

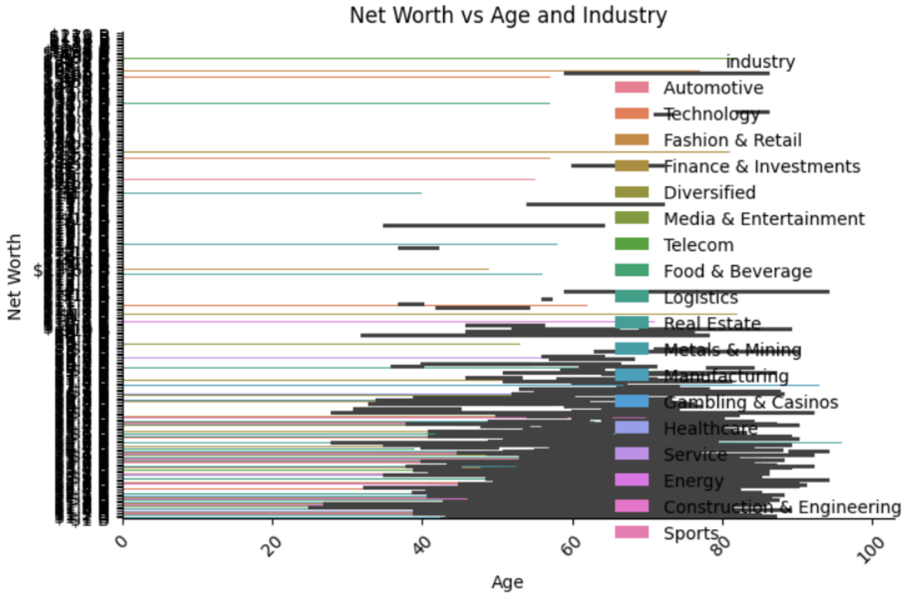




4. Show how does Net Worth Change with age and industry using cat plot

```
plt.figure(figsize=(10, 6))
sns.catplot(x='age', y='networth', hue='industry', data=dataset, kind='bar')

plt.title('Net Worth vs Age and Industry')
plt.xlabel('Age')
plt.ylabel('Net Worth')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
print(dataset.columns)
```

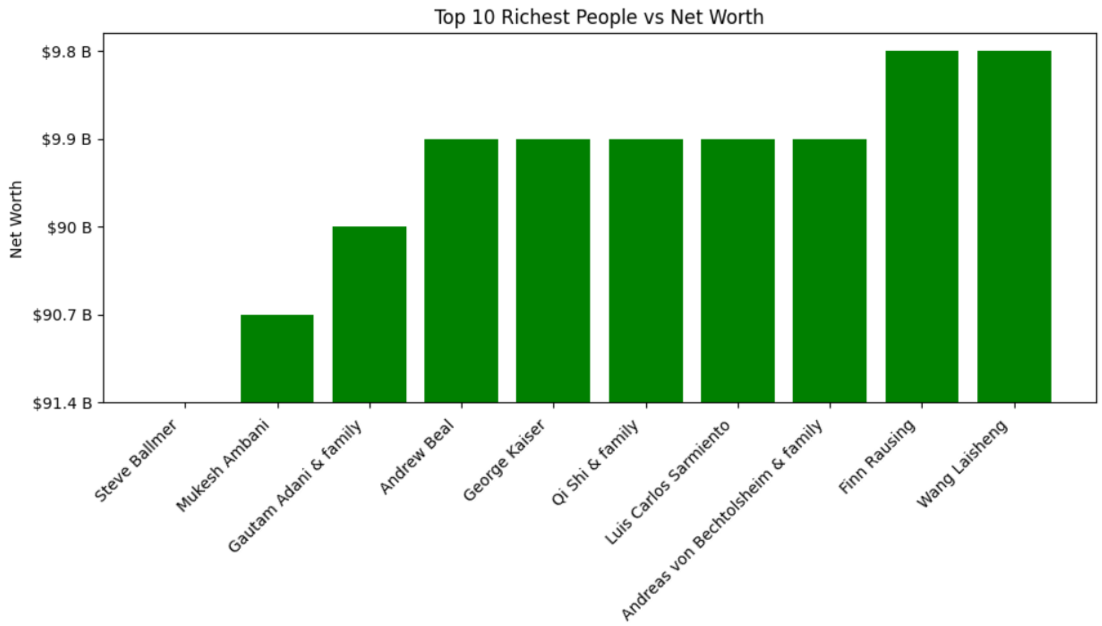
```
<Figure size 1000x600 with 0 Axes>
```



```
Index(['Unnamed: 0', 'rank', 'name', 'networth', 'age', 'country', 'source',
       'industry']
```

5. Show the top 10 richest people Vs net worth

```python
top_10_richest = dataset.sort_values(by='networth', ascending=False).head(10)


plt.figure(figsize=(10, 6))

plt.bar(top_10_richest['name'], top_10_richest['networth'], color='green')
plt.xlabel('Name')
plt.ylabel('Net Worth')
plt.title('Top 10 Richest People vs Net Worth')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

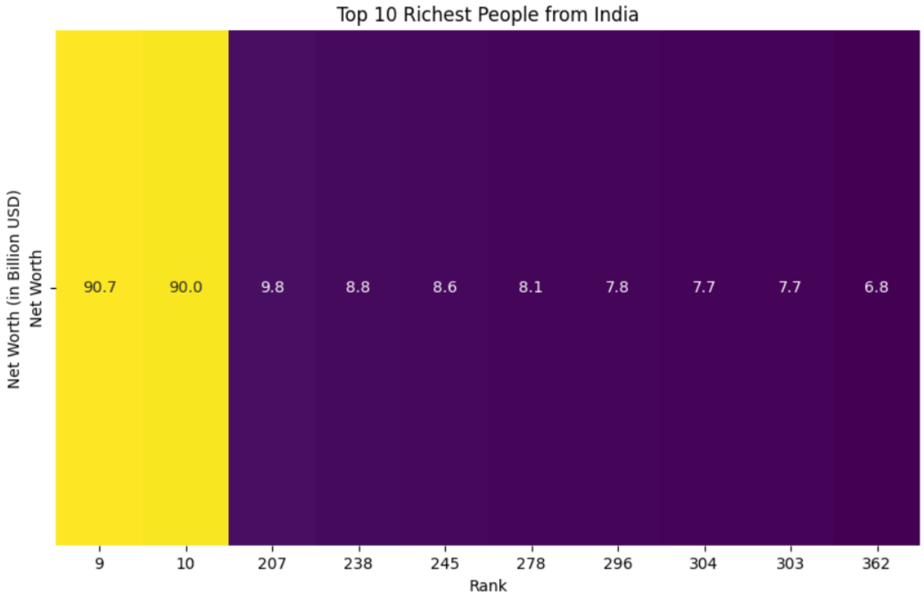


6. Show the richest people from India with the names using heatmap

```python
india_billionaires = dataset[dataset['country'] == 'India']


india_billionaires = india_billionaires.sort_values(by='networth', ascending=False)

top_india_billionaires = india_billionaires.head(10)
top_india_billionaires['Net Worth'] = top_india_billionaires['networth'].str.replace('$', '').str.replace(' B', '').astype(float)


plt.figure(figsize=(10, 6))
sns.heatmap(top_india_billionaires[['Net Worth']].transpose(), annot=True, fmt=".1f", cmap="viridis", cbar=False)
plt.title('Top 10 Richest People from India')
plt.xlabel('Rank')
plt.ylabel('Net Worth (in Billion USD)')
plt.show()
```

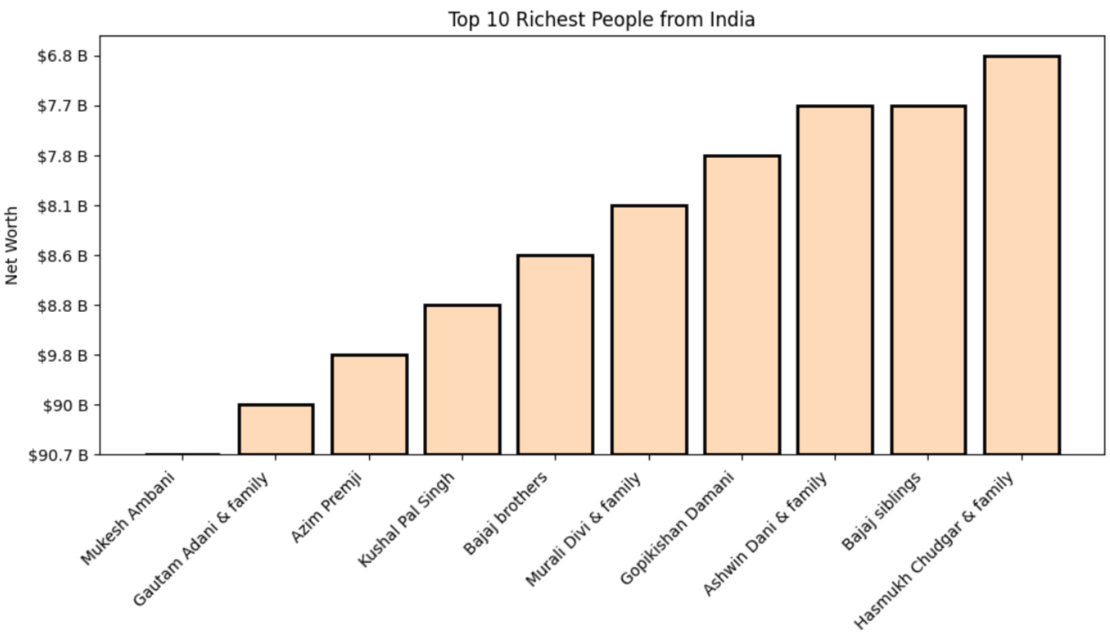
```
<ipython-input-45-a284049db388>:14: SettingWithCopyWarning: 
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a
  top_india_billionaires['Net Worth'] = top_india_billionaires['networth'].str.replace('$', '').str.replace(' B', '').astype(float)
```



6. Show the richest people from India with the names in any plot(Bar plot)

```
india_billionaires = dataset[dataset['country'] == 'India']


india_billionaires = india_billionaires.sort_values(by='networth', ascending=False)


top_india_billionaires = india_billionaires.head(10)


plt.figure(figsize=(10, 6))
plt.bar(top_india_billionaires['name'], top_india_billionaires['networth'], color='peachpuff')
plt.xlabel('Name')
plt.ylabel('Net Worth')
plt.title('Top 10 Richest People from India')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



7. Show the minimum age billionaire <=50 with name and industry

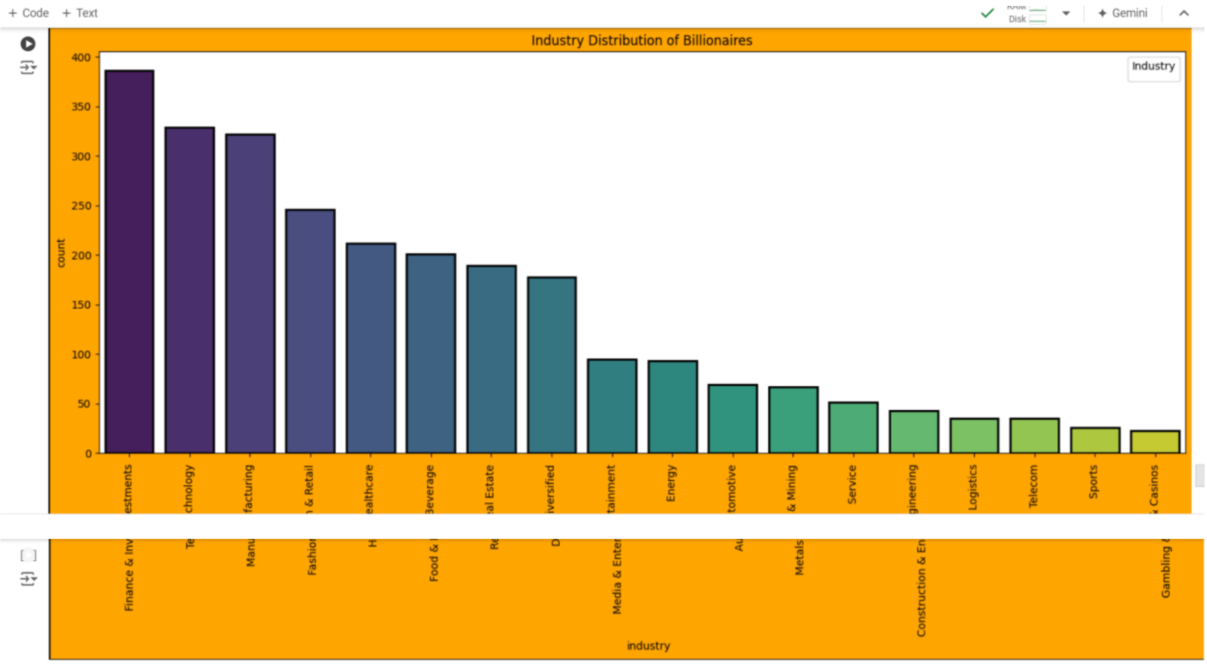
```
min_age_billionaires = dataset[dataset['age'] <= 50]
min_age_billionaires = min_age_billionaires.sort_values(by='age')
print(min_age_billionaires[['name', 'age', 'industry']].head(1))
```

```
                     name  age          industry
1311  Kevin David Lehmann   19  Fashion & Retail
```

8. Show in which industry billionaires are more

```
[ ]
```

```
plt.figure(figsize=(15, 8), dpi=100, facecolor='orange', edgecolor='black', linewidth=2)
sns.countplot(x='industry', data=dataset, color='violet', order=dataset['industry'].value_counts().index, palette='viridis',edgecolor='b
plt.xticks(rotation=90)
plt.title('Industry Distribution of Billionaires')
plt.legend(title='Industry', loc='upper right')
plt.tight_layout()
plt.show()
```


+ Code
+ Text
Disk
✦ Gemini
Industry Distribution of Billionaires
Industry
count
400
350
300
250
200
150
100
50
0
estments
chnology
facturing
n & Retail
ealthcare
Beverage
al Estate
iversified
tainment
Energy
tomotive
& Mining
Service
gineering
Logistics
Telecom
Sports
& Casinos
Finance & Inv
Te
Manu
Fashio
H
Food & I
Re
D
Media & Enter
Au
Metals
Construction & En
Gambling &
industry