# Data Collection and Data visualization

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

In [2]:
```
import pandas as pd
from matplotlib import pyplot as plt
```

In [5]:
```
my_dict = {'name': ['a', 'b', 'c', 'd', 'e', 'f', 'g'],
           'age' : [20, 27, 35, 55, 18, 21, 35],
           'designation' : ['VP', 'CEO', 'CFO', 'VP', 'VP', 'CEO', 'MD']}

df = pd.DataFrame(my_dict)
```

In [6]:
```
df
```

Out[6]:

| | name | age | designation |
|---|---|---|---|
| **0** | a | 20 | VP |
| **1** | b | 27 | CEO |
| **2** | c | 35 | CFO |
| **3** | d | 55 | VP |
| **4** | e | 18 | VP |
| **5** | f | 21 | CEO |
| **6** | g | 35 | MD |

In [10]:
```
df.to_csv("output_files/Output 1.csv")
```

In [11]:
```
df_csv = pd.read_csv('output_files/Output 1.csv')
df_csv
```

Out[11]:

| | Unnamed: 0 | name | age | designation |
|---|---|---|---|---|
| **0** | 0 | a | 20 | VP |
| **1** | 1 | b | 27 | CEO |
| **2** | 2 | c | 35 | CFO |
| **3** | 3 | d | 55 | VP |
| **4** | 4 | e | 18 | VP |
| **5** | 5 | f | 21 | CEO |
| **6** | 6 | g | 35 | MD |

# Loading Population Data

In [13]:
```
data = pd.read_csv(r'C:\Users\acer\Desktop\Sem 1\data science\mini project\dataSet\countries.csv')
```

In [14]:
```
data.head()
```

Out[14]:

| | country | year | population |
|---|---|---|---|
| **0** | Afghanistan | 1952 | 8425333 |
| **1** | Afghanistan | 1957 | 9240934 |
| **2** | Afghanistan | 1962 | 10267083 |
| **3** | Afghanistan | 1967 | 11537966 |
| **4** | Afghanistan | 1972 | 13079460 |

In [15]:
```
```

```
data.tail()
```

Out[15]:

| | country | year | population |
|---|---|---|---|
| **1699** | Zimbabwe | 1987 | 9216418 |
| **1700** | Zimbabwe | 1992 | 10704340 |
| **1701** | Zimbabwe | 1997 | 11404948 |
| **1702** | Zimbabwe | 2002 | 11926563 |
| **1703** | Zimbabwe | 2007 | 12311143 |

# Compare the population growth in the US and China

Selecting US Data

In [16]:
```
data[data.country == 'United States']
```

Out[16]:

| | country | year | population |
|---|---|---|---|
| **1608** | United States | 1952 | 157553000 |
| **1609** | United States | 1957 | 171984000 |
| **1610** | United States | 1962 | 186538000 |
| **1611** | United States | 1967 | 198712000 |
| **1612** | United States | 1972 | 209896000 |
| **1613** | United States | 1977 | 220239000 |
| **1614** | United States | 1982 | 232187835 |
| **1615** | United States | 1987 | 242803533 |
| **1616** | United States | 1992 | 256894189 |
| **1617** | United States | 1997 | 272911760 |
| **1618** | United States | 2002 | 287675526 |
| **1619** | United States | 2007 | 301139947 |

In [17]:
```
us = data[data.country == 'United States']
```

Selecting China's data

In [18]:
```
china = data[data.country == 'China']
```

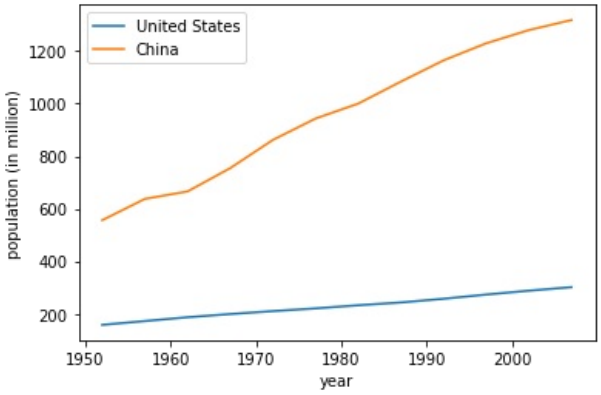
In [19]:
```
china
```

Out[19]:

| | country | year | population |
|---|---|---|---|
| **288** | China | 1952 | 556263527 |
| **289** | China | 1957 | 637408000 |
| **290** | China | 1962 | 665770000 |
| **291** | China | 1967 | 754550000 |
| **292** | China | 1972 | 862030000 |
| **293** | China | 1977 | 943455000 |
| **294** | China | 1982 | 1000281000 |
| **295** | China | 1987 | 1084035000 |
| **296** | China | 1992 | 1164970000 |
| **297** | China | 1997 | 1230075000 |
| **298** | China | 2002 | 1280400000 |
| **299** | China | 2007 | 1318683096 |

In [20]:
```
plt.plot(us.year, us.population / 10**6)
```

```
plt.plot(china.year, china.population / 10**6)
plt.legend(['United States', 'China'])
plt.xlabel('year')
plt.ylabel('population (in million)')
plt.show()
```

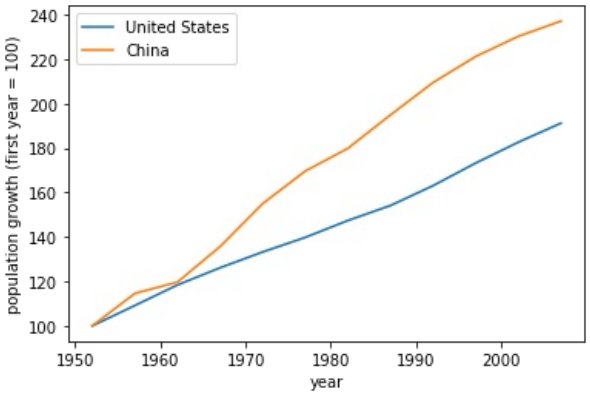



In [21]:
```
us.population
```

Out[21]:
```
1608    157553000
1609    171984000
1610    186538000
1611    198712000
1612    209896000
1613    220239000
1614    232187835
1615    242803533
1616    256894189
1617    272911760
1618    287675526
1619    301139947
Name: population, dtype: int64
```

# Show population in percentage

In [22]:
```
plt.plot(us.year, us.population / us.population.iloc[0] * 100)
plt.plot(china.year, china.population / china.population.iloc[0] * 100)
plt.legend(['United States', 'China'])
plt.xlabel('year')
plt.ylabel('population growth (first year = 100)')
plt.show()
```




In [23]:
```
data
```

Out[23]:

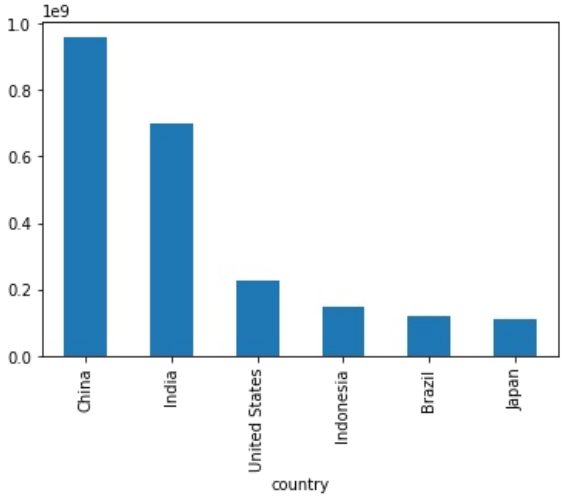
| | country | year | population |
|---|---|---|---|
| **0** | Afghanistan | 1952 | 8425333 |
| **1** | Afghanistan | 1957 | 9240934 |
| **2** | Afghanistan | 1962 | 10267083 |
| **3** | Afghanistan | 1967 | 11537966 |

| | | | |
|---|---|---|---|
| **4** | Afghanistan | 1972 | 13079460 |
| **...** | ... | ... | ... |
| **1699** | Zimbabwe | 1987 | 9216418 |
| **1700** | Zimbabwe | 1992 | 10704340 |
| **1701** | Zimbabwe | 1997 | 11404948 |
| **1702** | Zimbabwe | 2002 | 11926563 |
| **1703** | Zimbabwe | 2007 | 12311143 |

1704 rows × 3 columns

In [32]:
```
data.groupby("country").population.mean().sort_values(ascending=False)[:6].plot.bar()
```

Out[32]: <AxesSubplot:xlabel='country'>




In [ ]:
```

```