Population Data Analysis

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:
```
df = pd.read_csv("C:\\Users\\paree\\Downloads\\world_population_mock_data.csv")
```

In [4]:
```
df.head()
```

Out[4]:

	Country	Year	Population	Male Population	Female Population	Growth Rate (%)	Gender Ratio (M/F)	Literacy Rate (%)	Popu
0	India	2000	1050000000	540000000	510000000	1.7	106	65.4	
1	India	2010	1230000000	640000000	590000000	1.5	108	72.0	
2	India	2020	1390000000	710000000	680000000	1.3	104	74.4	
3	USA	2000	282000000	138000000	144000000	1.1	96	79.0	
4	USA	2010	309000000	150000000	159000000	0.9	94	85.0	

Exploring and cleaning the data

In [6]:
```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18 entries, 0 to 17
Data columns (total 9 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Country               18 non-null     object
 1   Year                  18 non-null     int64
 2   Population            18 non-null     int64
 3   Male Population       18 non-null     int64
 4   Female Population     18 non-null     int64
 5   Growth Rate (%)       18 non-null     float64
 6   Gender Ratio (M/F)    18 non-null     int64
 7   Literacy Rate (%)     18 non-null     float64
 8   Urban Population (%)  18 non-null     float64
dtypes: float64(3), int64(5), object(1)
memory usage: 1.4+ KB
None
```

In [7]:
```
df.describe()
```

Out[7]:

	Year	Population	Male Population	Female Population	Growth Rate (%)	Gender Ratio (M/F)
count	18.000000	1.800000e+01	1.800000e+01	1.800000e+01	18.000000	18.000000
mean	2010.000000	5.615556e+08	2.852222e+08	2.763333e+08	1.183333	99.833333
std	8.401681	5.306641e+08	2.739275e+08	2.567975e+08	0.728617	5.238433
min	2000.000000	1.230000e+08	6.200000e+07	6.100000e+07	0.200000	92.000000
25%	2000.000000	1.632500e+08	8.225000e+07	8.100000e+07	0.725000	96.000000
50%	2010.000000	2.475000e+08	1.215000e+08	1.265000e+08	1.050000	99.000000
75%	2020.000000	1.185000e+09	6.150000e+08	5.700000e+08	1.450000	104.000000
max	2020.000000	1.410000e+09	7.200000e+08	6.900000e+08	2.600000	108.000000

In [8]:
```
df.isnull().sum()
```

Out[8]:
```
Country                 0
Year                    0
Population              0
Male Population         0
Female Population       0
Growth Rate (%)         0
Gender Ratio (M/F)      0
Literacy Rate (%)       0
Urban Population (%)    0
dtype: int64
```

In [9]:
```
df.duplicated().sum()
```

Out[9]:
```
0
```

In [10]:
```
df.fillna(0,inplace= True)
```

Analysing India's population data

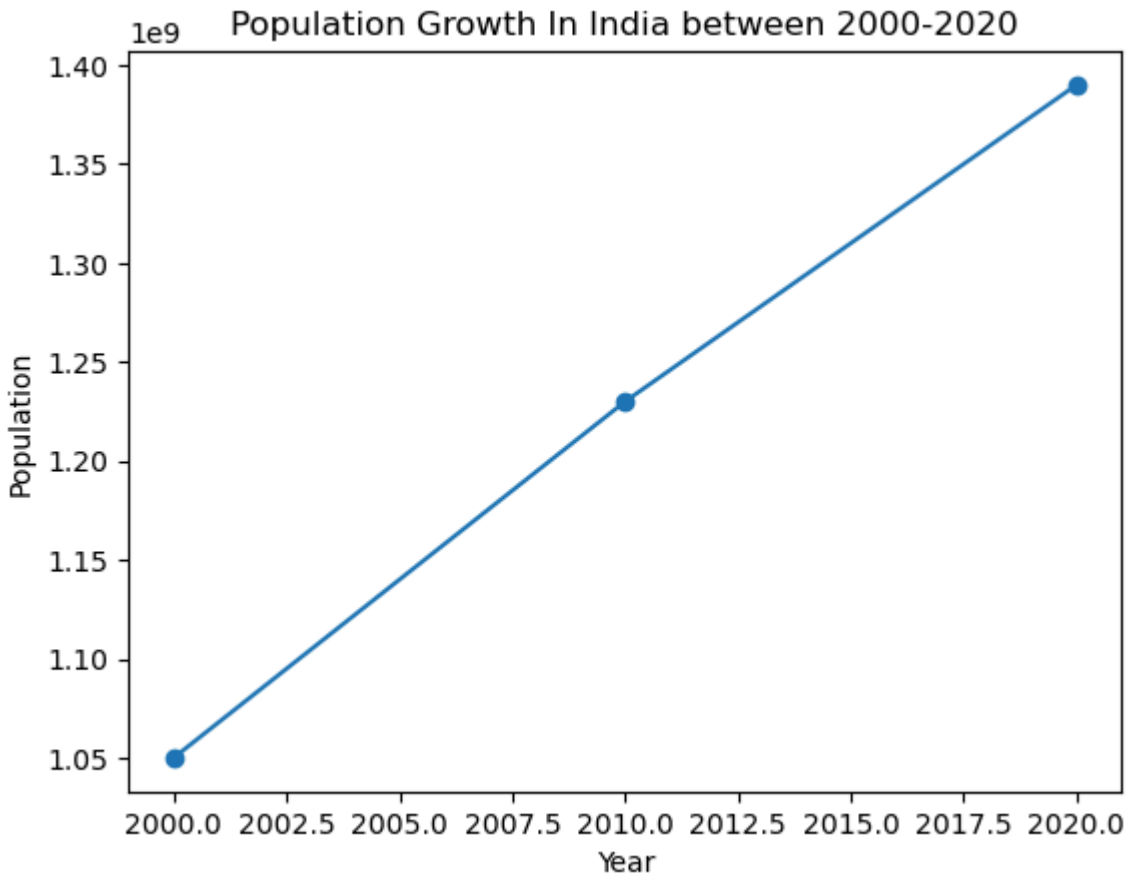
In [12]:
```
india_data = df[df['Country'] == 'India']
india_data.head()
```

Out[12]:

	Country	Year	Population	Male Population	Female Population	Growth Rate (%)	Gender Ratio (M/F)	Literacy Rate (%)	Popu
0	India	2000	1050000000	540000000	510000000	1.7	106	65.4	
1	India	2010	1230000000	640000000	590000000	1.5	108	72.0	
2	India	2020	1390000000	710000000	680000000	1.3	104	74.4	

In [13]:
```
plt.plot(india_data['Year'], india_data['Population'], marker='o' )
plt.title("Population Growth In India between 2000-2020")
```

```
plt.xlabel("Year")
plt.ylabel("Population")
plt.show()
```



In [14]:
```
#country_population = df.groupby('Country') #grouping the dataframe by country c
#country_population['Population'].sum() #calculating population for all the year
# country_population.head(10)
"""if following the above one then getting 6 bar charts while plotting
because of grouping the country column, any country may have data of 6 years in
so it leads to generate 6 graph while plotting total country population"""
country_population = df.groupby('Country').sum()
```

In [15]:
```
country_population.head(2)
```

Out[15]:

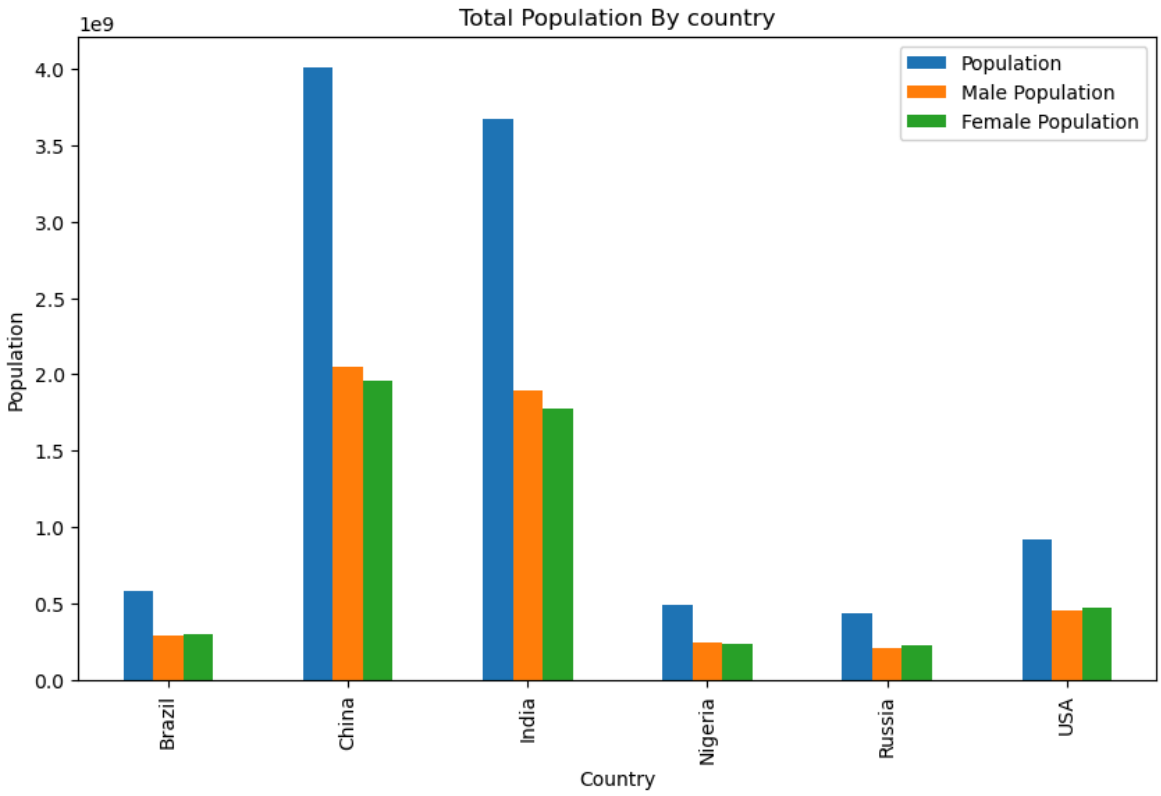
	Year	Population	Male Population	Female Population	Growth Rate (%)	Gender Ratio (M/F)	Literacy Rate (%)	U Popula
Country								
Brazil	6030	585000000	286000000	299000000	3.1	287	269.0	2
China	6030	4010000000	2050000000	1960000000	2.6	314	253.0	1

In [16]:
```
avg_population = df.groupby('Country')['Population'].mean() # calculating avg po
avg_population.head()
```

Out[16]:
```
Country
Brazil     1.950000e+08
China      1.336667e+09
India      1.223333e+09
Nigeria    1.626667e+08
Russia     1.443333e+08
Name: Population, dtype: float64
```

In [17]:
```
country_population = df.groupby('Country')[['Population', 'Male Population', 'Fe
```

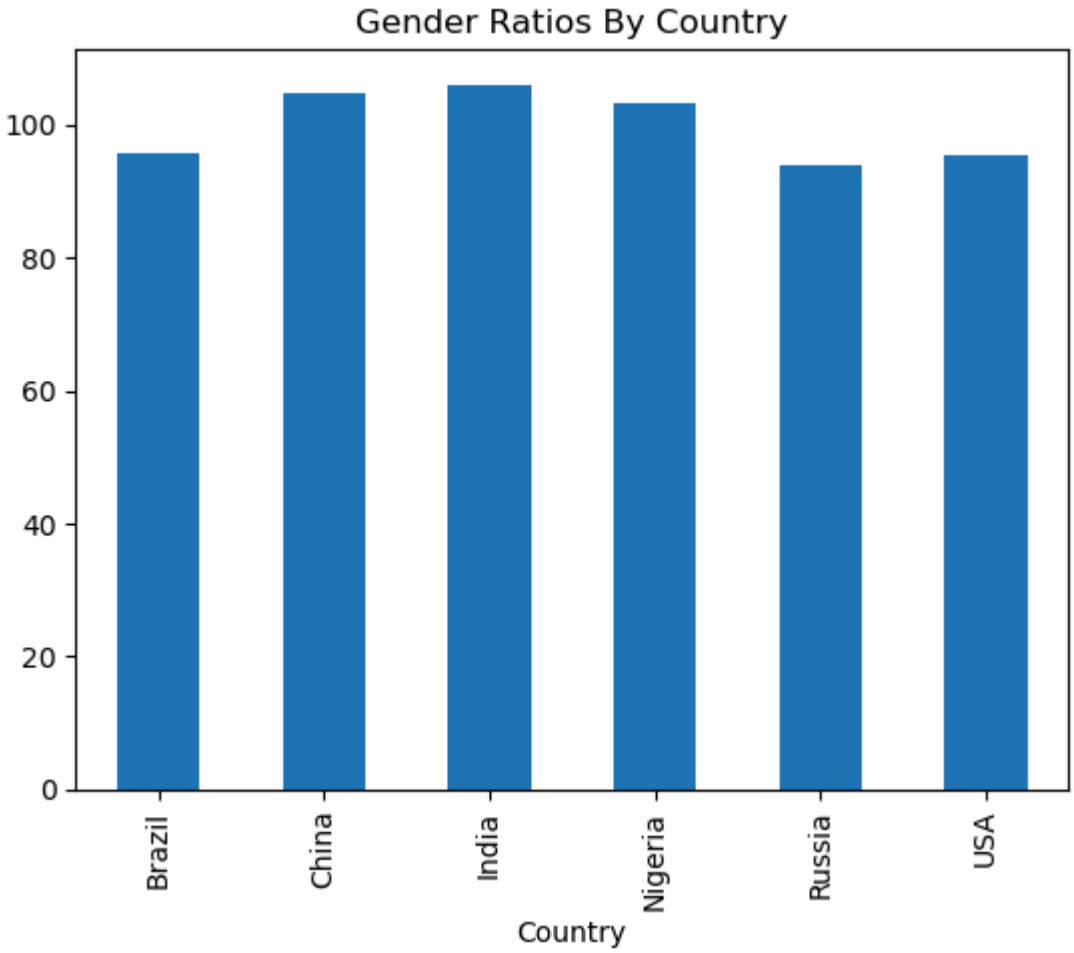
In [18]:
```
country_population.plot(kind='bar',figsize=(10,6))  # using figsize for better v
plt.title('Total Population By country')
plt.xlabel('Country')
plt.ylabel('Population')
plt.show()
```



If we are plotting directly using pandas, so we have to be careful that pandas assumes that our data is unique or grouped by, so sometimes it plot similar values on graph. So we should group the data. . If we take 'avg/mean' while grouping than will get nearly equal to the actual value but if we use 'sum' then we will get slighly higher values because it adds all the value while grouping

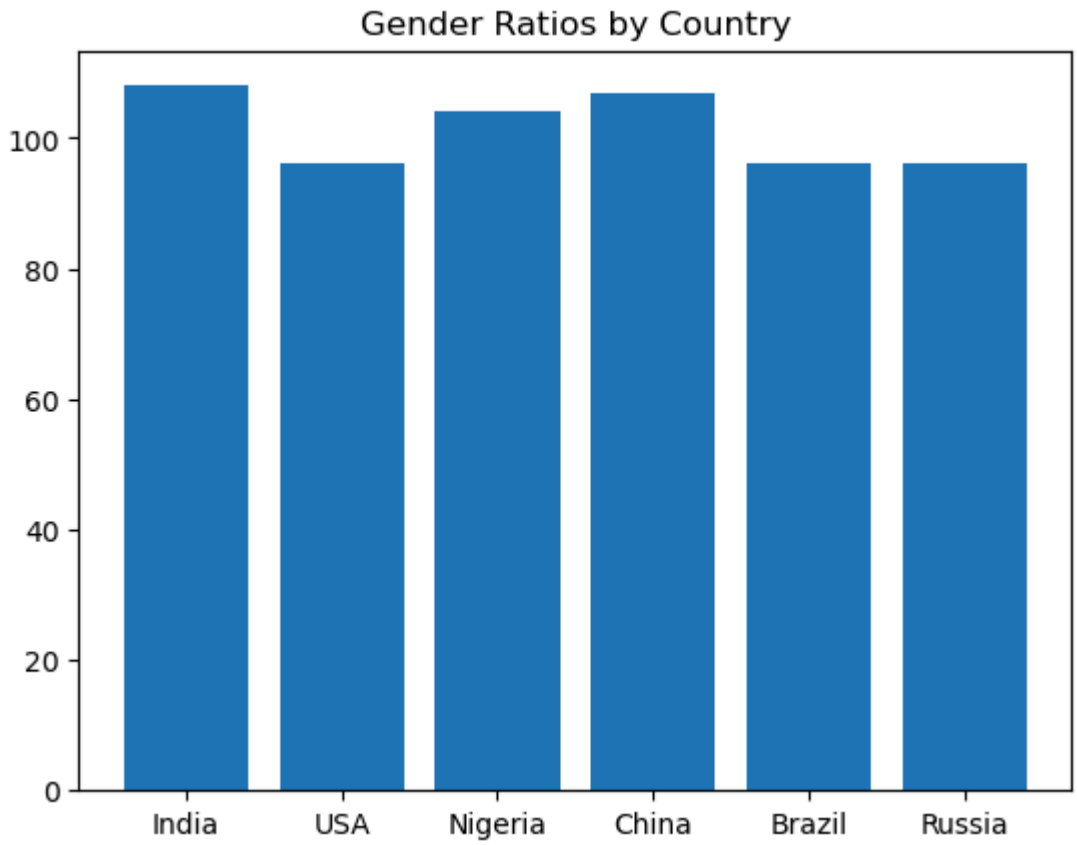
In [20]:
```
# Grouping by 'Country' and calculate the mean of 'Gender Ratio (M/F)' for each
df_grouped = df.groupby('Country')['Gender Ratio (M/F)'].mean() #grouping and ta
df_grouped.plot(x='Country', y='Gender Ratio (M/F)', kind='bar',title='Gender Ra
```

Out[20]:
```
<Axes: title={'center': 'Gender Ratios By Country'}, xlabel='Country'>
```



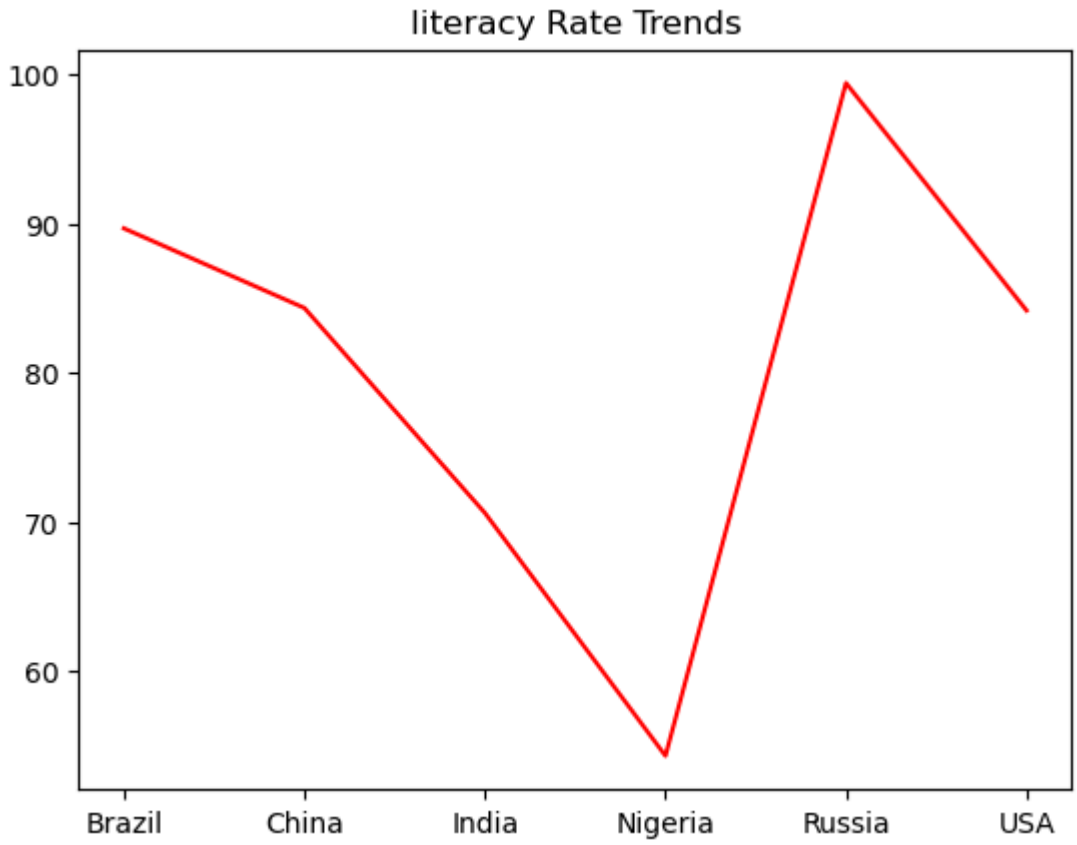
Below we are plotting using matplotlb without grouping the data and getting correct graph because matplotlib not consider all the values of x as categorical data/grouped and remove duplicated or repeated data automatically. But we should avoid using it directly without grouping to ensure accurate results.

In [22]:
```
plt.bar(df['Country'],df['Gender Ratio (M/F)'])
plt.title("Gender Ratios by Country")
plt.show()
```



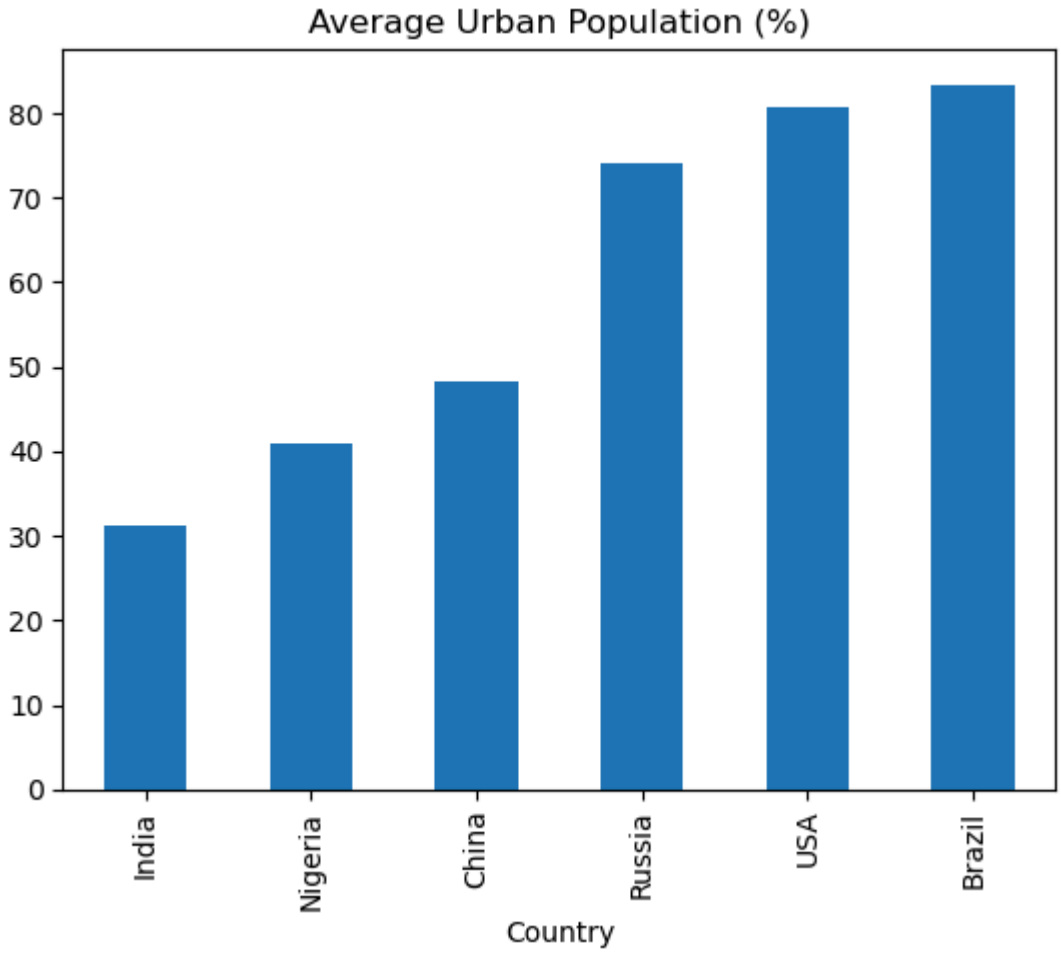
In [23]:
```
literacy_rate = df.groupby('Country')['Literacy Rate (%)'].mean()
literacy_rate=literacy_rate.reset_index()
plt.plot(literacy_rate['Country'],literacy_rate['Literacy Rate (%)'],color='red'
plt.title("literacy Rate Trends")
```

Out[23]: Text(0.5, 1.0, 'literacy Rate Trends')



PLOTTING AVERAGE URBAN POPULATION USING PANDAS

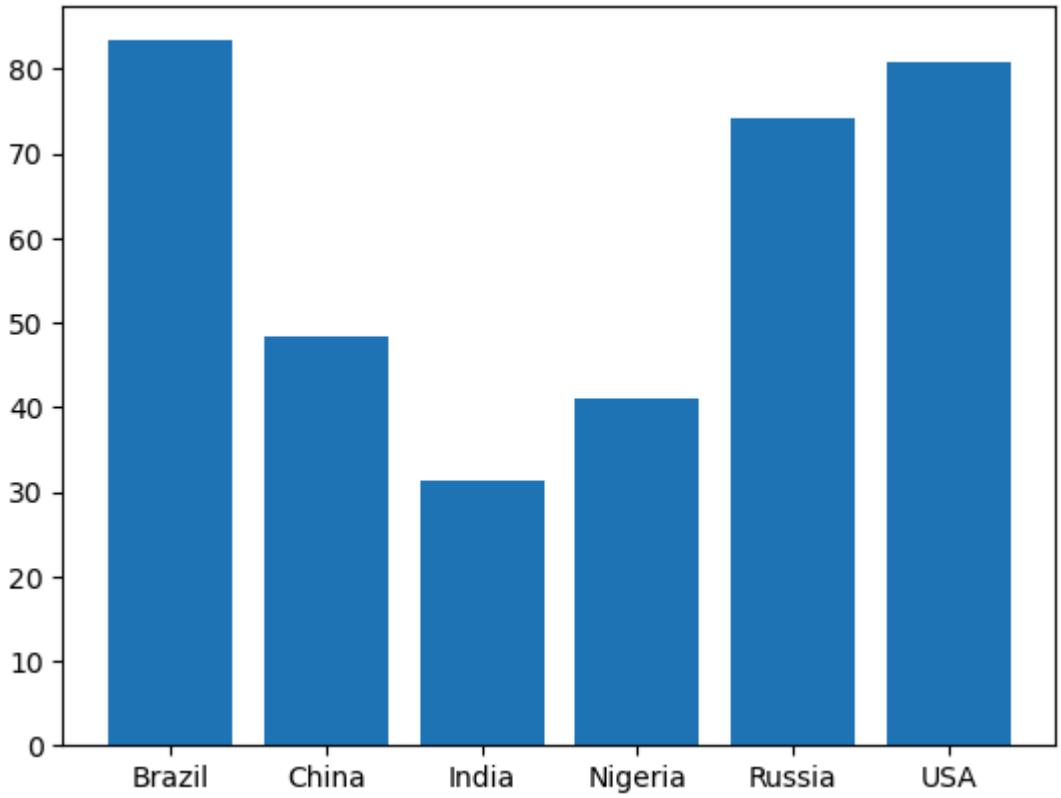
In [25]:
```
urban_population = df.groupby('Country')['Urban Population (%)'].mean()
#urban_population.plot(kind='bar',title='Average Urban Population (%)') # gettin
urban_population.sort_values().plot(kind='bar',title='Average Urban Population (
plt.show()
```



PLOTTING AVERAGE URBAN POPULATION USING MATPLOTLIB

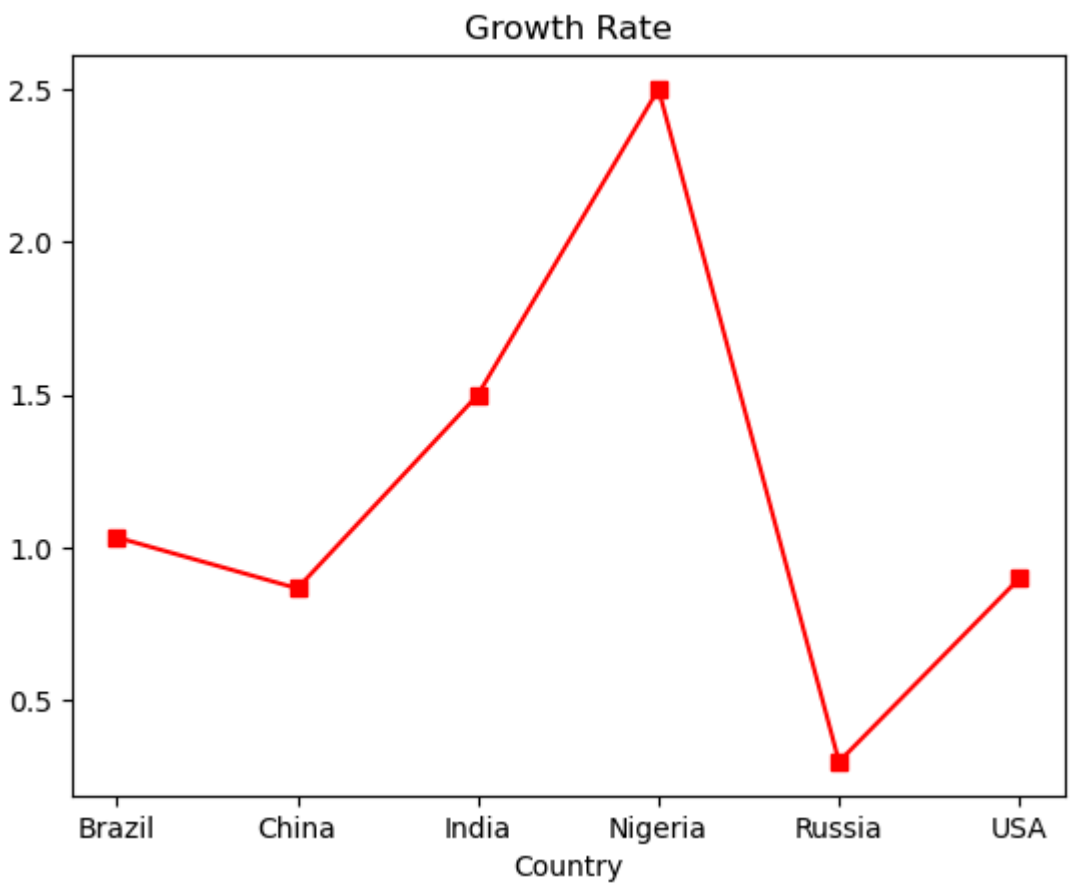
In [27]:
```
urban_population = df.groupby('Country')['Urban Population (%)'].mean()
urban_population= urban_population.reset_index()
plt.bar(urban_population['Country'],urban_population['Urban Population (%)'])
```

Out[27]:
```
<BarContainer object of 6 artists>
```



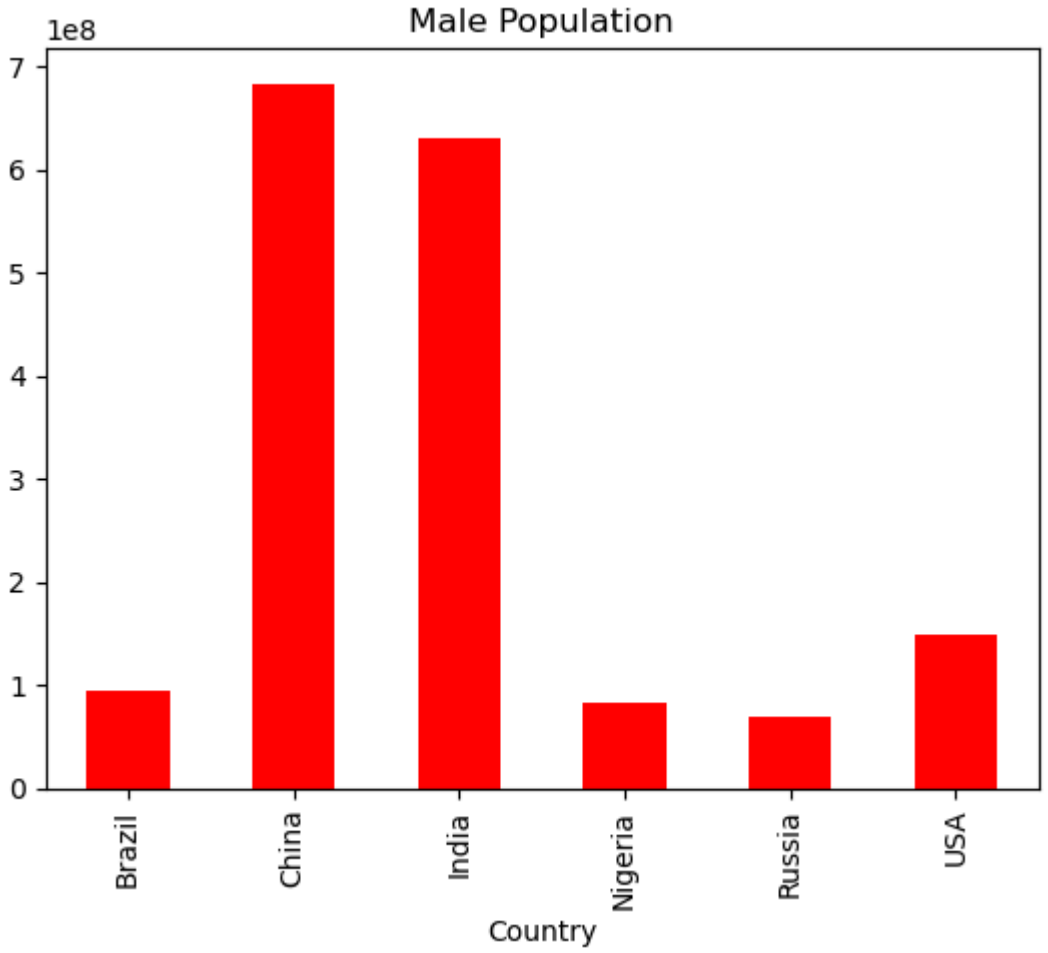
In [28]:
```
growth_rate=df.groupby('Country')['Growth Rate (%)'].mean()
#growth_rate.head()
growth_rate.plot(kind='line',title='Growth Rate',marker='s',color='red')
```

Out[28]: <Axes: title={'center': 'Growth Rate'}, xlabel='Country'>



In [29]:
```
male_population=df.groupby('Country')['Male Population'].mean()
male_population.plot(kind='bar',title='Male Population',color='red')
```

Out[29]:
```
<Axes: title={'center': 'Male Population'}, xlabel='Country'>
```



Male and Female Proportion

In [31]:
```
df['Male Proportion (%)'] = (df['Male Population'] / df['Population']) * 100
df['Female Proportion (%)'] = (df['Female Population'] / df['Population']) * 100
df.head(5)
```

Out[31]:

	Country	Year	Population	Male Population	Female Population	Growth Rate (%)	Gender Ratio (M/F)	Literacy Rate (%)	Popu
0	India	2000	1050000000	540000000	510000000	1.7	106	65.4	
1	India	2010	1230000000	640000000	590000000	1.5	108	72.0	
2	India	2020	1390000000	710000000	680000000	1.3	104	74.4	
3	USA	2000	282000000	138000000	144000000	1.1	96	79.0	
4	USA	2010	309000000	150000000	159000000	0.9	94	85.0	

Male proportion of India

In [33]:
```
india_data = df[df['Country'] == 'India']
india_data.head()
india_data.plot(x='Year',y='Male Proportion (%)',kind='bar',title='India Populat
```

Out[33]:
```
<Axes: title={'center': 'India Population'}, xlabel='Year'>
```