Loading required packages

In [1]:
```
import pandas as pd # used to work with 2 dimesional datasets
import numpy as np # to perform mathematical functions
import matplotlib.pyplot as plt # visualizations
import seaborn as sns
```

Loading data

In [2]:
```
df = pd.read_csv('Population_Data.csv')
```

In [3]:
```
df.head()
```

Out[3]:

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	
0	Aruba	ABW	Population, total	SP.POP.TOTL	54608.0	55811.0	56682.0	57475.0	
1	Africa Eastern and Southern	AFE	Population, total	SP.POP.TOTL	130692579.0	134169237.0	137835590.0	141630546.0	14
2	Afghanistan	AFG	Population, total	SP.POP.TOTL	8622466.0	8790140.0	8969047.0	9157465.0	
3	Africa Western and Central	AFW	Population, total	SP.POP.TOTL	97256290.0	99314028.0	101445032.0	103667517.0	10
4	Angola	AGO	Population, total	SP.POP.TOTL	5357195.0	5441333.0	5521400.0	5599827.0	

5 rows × 66 columns

Data cleaning

In [4]:
```
df.loc[:, ~df.columns.isin(['Indicator Name','Indicator Code'])] # deleting the 2 column
```

Out[4]:

	Country Name	Country Code	1960	1961	1962	1963	1964	1965
0	Aruba	ABW	54608.0	55811.0	56682.0	57475.0	58178.0	58782.0
1	Africa Eastern and Southern	AFE	130692579.0	134169237.0	137835590.0	141630546.0	145605995.0	149742351.0
2	Afghanistan	AFG	8622466.0	8790140.0	8969047.0	9157465.0	9355514.0	9565147.0
3	Africa Western and Central	AFW	97256290.0	99314028.0	101445032.0	103667517.0	105959979.0	108336203.0
4	Angola	AGO	5357195.0	5441333.0	5521400.0	5599827.0	5673199.0	5736582.0
...	...	...	...	...	...	...	...	...
261	Kosovo	XKX	947000.0	966000.0	994000.0	1022000.0	1050000.0	1078000.0
262	Yemen, Rep.	YEM	5542459.0	5646668.0	5753386.0	5860197.0	5973803.0	6097298.0

263	South Africa	ZAF	16520441.0	16989464.0	17503133.0	18042215.0	18603097.0	19187194.0
264	Zambia	ZMB	3119430.0	3219451.0	3323427.0	3431381.0	3542764.0	3658024.0
265	Zimbabwe	ZWE	3806310.0	3925952.0	4049778.0	4177931.0	4310332.0	4447149.0

266 rows × 64 columns

In [5]:
```
df.isnull().sum() # checking for null values
```

Out[5]:
```
Country Name       0
Country Code       0
Indicator Name     0
Indicator Code     0
1960               2
                 ..
2017               1
2018               1
2019               1
2020               1
2021               1
Length: 66, dtype: int64
```

In [6]:
```
df[df.isna().any(axis=1)] # check if entire row is Nan
```

Out[6]:

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	...	2012	
110	Not classified	INX	Population, total	SP.POP.TOTL	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	
196	West Bank and Gaza	PSE	Population, total	SP.POP.TOTL	NaN	NaN	NaN	NaN	NaN	NaN	...	3979998.0	4

2 rows × 66 columns

In [7]:
```
df = df.drop(110) # delete the row which is completely NaN
```

In [8]:
```
df_countries = pd.read_csv('Country_Continent.csv') # Load countries dataset to clean th
```

In [9]:
```
df_countries
```

Out[9]:

	No	Country	Country Code	Region1	Region2	Continent	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed
0	1	Afghanistan	AFG	Southern Asia	NaN	Asia	NaN	NaN	NaN	Na
1	2	Åland Islands	ALA	Northern Europe	NaN	Europe	NaN	NaN	NaN	Na
2	3	Albania	ALB	Southern Europe	NaN	Europe	NaN	NaN	NaN	Na
3	4	Algeria	DZA	Northern Africa	NaN	Africa	NaN	NaN	NaN	Na
4	5	American Samoa	ASM	Polynesia	NaN	Oceania	NaN	NaN	NaN	Na
...	...	...	...	...	...	...	...	...	...	
244	245	Wallis and Futuna	WLF	Polynesia	NaN	Oceania	NaN	NaN	NaN	Na

		Islands								
245	246	Western Sahara	ESH	Northern Africa	NaN	Africa	NaN	NaN	NaN	Na
246	247	Yemen	YEM	Western Asia	NaN	Asia	NaN	NaN	NaN	Na
247	248	Zambia	ZMB	Eastern Africa	Sub-Saharan Africa	Africa	NaN	NaN	NaN	Na
248	249	Zimbabwe	ZWE	Eastern Africa	Sub-Saharan Africa	Africa	NaN	NaN	NaN	Na

249 rows × 19 columns

In [10]:
```
df_countries = df_countries.rename(columns={'Country':'Country Name'})
```

In [11]:
```
df_c = df_countries[['Country Name','Continent']] # choosing the required columns to mer
```

In [12]:
```
df_new =pd.merge(df, df_c, how='right') # performed outer join function to remove all th
```

In [13]:
```
df_new.head()
```

Out[13]:

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	19
0	Afghanistan	AFG	Population, total	SP.POP.TOTL	8622466.0	8790140.0	8969047.0	9157465.0	935551
1	Åland Islands	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
2	Albania	ALB	Population, total	SP.POP.TOTL	1608800.0	1659800.0	1711319.0	1762621.0	181413
3	Algeria	DZA	Population, total	SP.POP.TOTL	11394307.0	11598608.0	11778260.0	11969451.0	1217909
4	American Samoa	ASM	Population, total	SP.POP.TOTL	20085.0	20626.0	21272.0	21949.0	2265

5 rows × 67 columns

In [14]:
```
df_new.describe()
```

Out[14]:

	1960	1961	1962	1963	1964	1965	
count	1.840000e+02	1.840000e+02	1.840000e+02	1.840000e+02	1.840000e+02	1.840000e+02	1.840000
mean	1.401764e+07	1.418850e+07	1.443241e+07	1.474221e+07	1.505466e+07	1.537214e+07	1.570402
std	6.062126e+07	6.070186e+07	6.151329e+07	6.298252e+07	6.443114e+07	6.592405e+07	6.759963
min	2.646000e+03	2.888000e+03	3.171000e+03	3.481000e+03	3.811000e+03	4.161000e+03	4.531000
25%	2.681348e+05	2.765428e+05	2.855908e+05	2.952535e+05	3.049290e+05	3.128525e+05	3.117792
50%	2.364900e+06	2.409711e+06	2.466161e+06	2.518165e+06	2.569733e+06	2.611200e+06	2.652450
75%	7.651168e+06	7.864610e+06	8.026916e+06	8.136362e+06	8.235862e+06	8.340491e+06	8.568825
max	6.670700e+08	6.603300e+08	6.657700e+08	6.823350e+08	6.983550e+08	7.151850e+08	7.354000

8 rows × 62 columns

In [15]:
```
df_new.duplicated().sum() # check for duplicates
```

Out[15]:
```
0
```

In [16]:
```
df_new.nunique()
```

Out[16]:
```
Country Name      249
Country Code      184
Indicator Name      1
Indicator Code      1
1960              184
                 ...
2018              184
2019              184
2020              184
2021              184
Continent           7
Length: 67, dtype: int64
```

In [17]:
```
df_new.columns = list(map(str, df_new.columns)) #changing all the column values to strin
```

In [18]:
```
from plotly.subplots import make_subplots
import matplotlib.pyplot as plt
import plotly.express as px
from plotly.offline import iplot, init_notebook_mode
import plotly.offline as py
py.init_notebook_mode(connected=True)
```

1. EDA - Analyze the global population growth

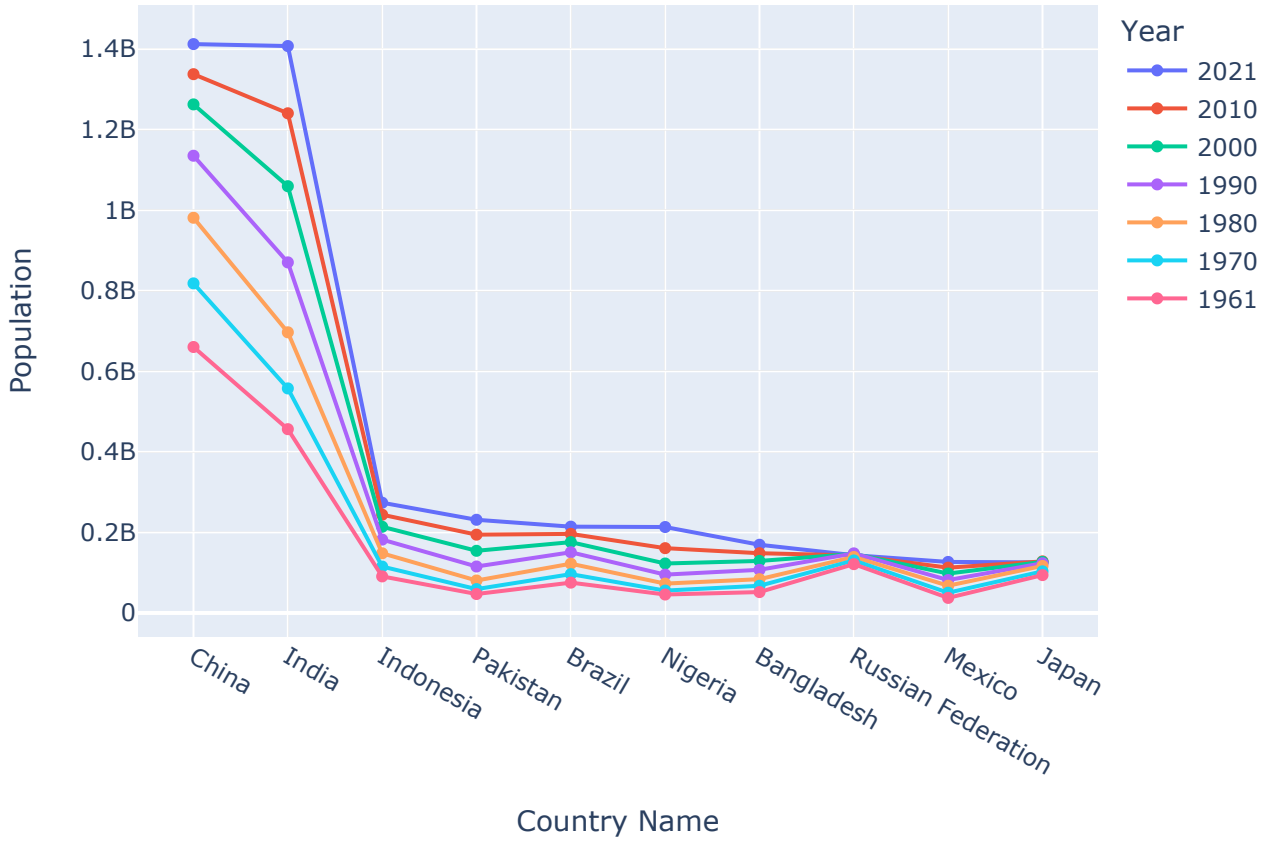
plot the graph for the top 10 countries with highest population

In [19]:
```
top10 = df_new.sort_values('2021', ascending=False).head(10)
top10 = top10.melt(id_vars=['Country Name'], value_vars=['2021', '2010', '2000', '1990',
top10.head(10)
```

Out[19]:

	Country Name	Year	Population
0	China	2021	1.412360e+09
1	India	2021	1.407564e+09
2	Indonesia	2021	2.737532e+08
3	Pakistan	2021	2.314021e+08
4	Brazil	2021	2.143262e+08
5	Nigeria	2021	2.134013e+08
6	Bangladesh	2021	1.693563e+08
7	Russian Federation	2021	1.434493e+08
8	Mexico	2021	1.267051e+08
9	Japan	2021	1.256816e+08

In [20]:
```
fig = px.line(top10, x='Country Name', y='Population', color='Year', markers=True, title
fig.show()
```



In [21]:
```
df_new = df_new.fillna(0)
```

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

Out[22]:

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	19
0	Afghanistan	AFG	Population, total	SP.POP.TOTL	8622466.0	8790140.0	8969047.0	9157465.0	935551
1	Åland Islands	0	0	0	0.0	0.0	0.0	0.0	
2	Albania	ALB	Population, total	SP.POP.TOTL	1608800.0	1659800.0	1711319.0	1762621.0	181413
3	Algeria	DZA	Population, total	SP.POP.TOTL	11394307.0	11598608.0	11778260.0	11969451.0	1217909
4	American Samoa	ASM	Population, total	SP.POP.TOTL	20085.0	20626.0	21272.0	21949.0	2265

5 rows × 67 columns

In [23]:
```
df_new.shape
```

Out[23]:
```
(249, 67)
```

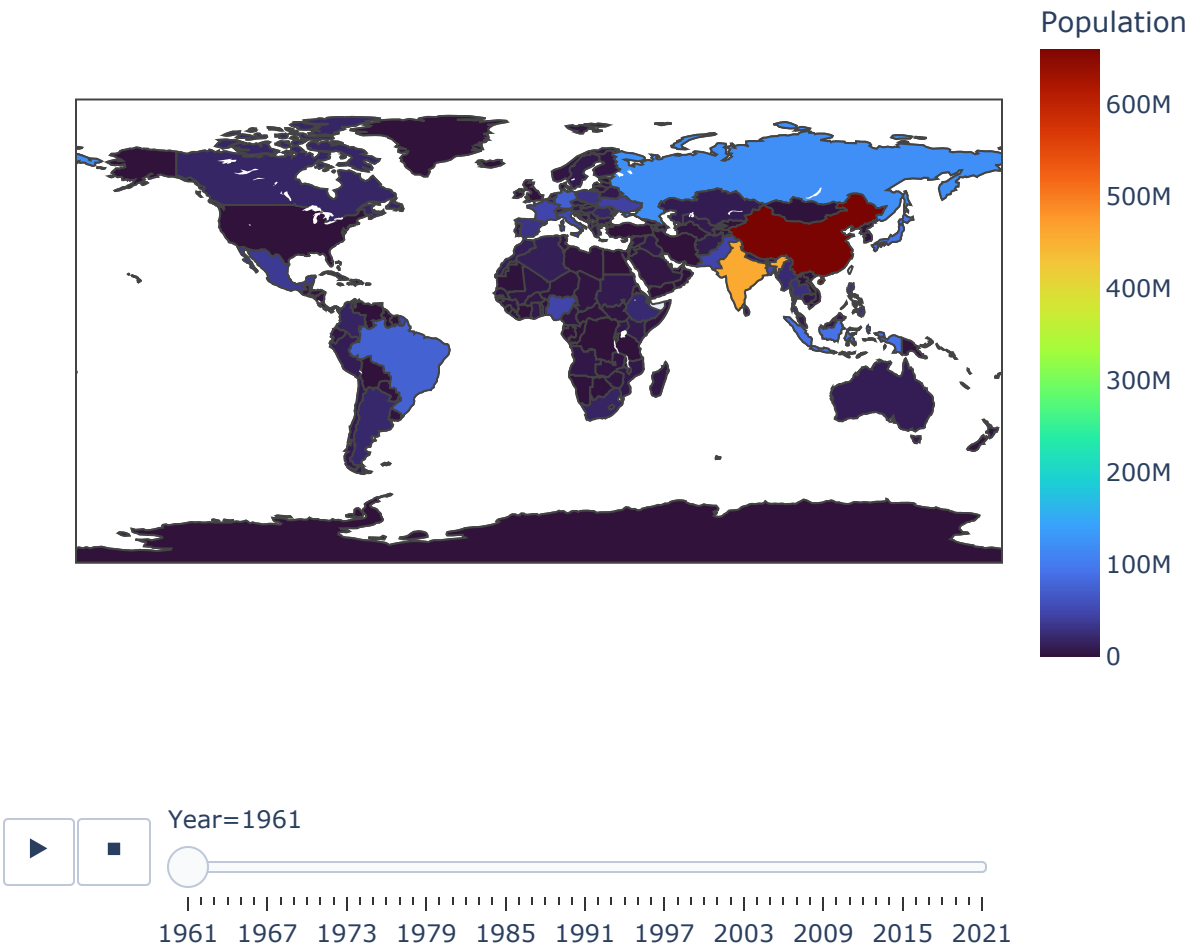
In [24]:
```
pop = df_new.melt(id_vars=['Country Name','Continent'], value_vars=df_new.iloc[:,5:67],
pop = pop.sort_values('Year')
pop.head(30)
```

Out[24]:

	Country Name	Continent	Year	Population
0	Afghanistan	Asia	1961	8790140.0
158	Nicaragua	North America	1961	1844630.0
159	Niger	Africa	1961	3602530.0
160	Nigeria	Africa	1961	45855507.0
161	Niue	Oceania	1961	0.0
162	Norfolk Island	Oceania	1961	0.0
163	North Macedonia	Europe	1961	1481112.0
164	Northern Mariana Islands	Oceania	1961	8965.0
165	Norway	Europe	1961	3609800.0
166	Oman	Asia	1961	546443.0
167	Pakistan	Asia	1961	47060915.0
168	Palau	Oceania	1961	9639.0
169	Panama	North America	1961	1160832.0
170	Papua New Guinea	Oceania	1961	2035672.0
171	Paraguay	South America	1961	1941208.0
172	Peru	South America	1961	10478096.0
173	Philippines	Asia	1961	29342411.0
174	Pitcairn	Oceania	1961	0.0
175	Poland	Europe	1961	29964000.0
176	Portugal	Europe	1961	8929316.0
177	Puerto Rico	North America	1961	2399722.0
178	Qatar	Asia	1961	40111.0
179	Republic of Korea	Asia	1961	0.0
180	Republic of Moldova	Europe	1961	0.0
181	Réunion	Africa	1961	0.0
182	Romania	Europe	1961	18555250.0
183	Russian Federation	Europe	1961	121236000.0
184	Rwanda	Africa	1961	3046654.0
157	New Zealand	Oceania	1961	2419700.0
185	Saint Barthélemy	North America	1961	0.0

In [25]:

```
fig = px.choropleth(pop,
                    locations = 'Country Name',
                    color="Population",
                    animation_frame="Year",
                    color_continuous_scale='Turbo',
                    locationmode='country names',
                    title='Total Population Over the World Animation (Year 1961-2021)',
                    height=600
                   )
fig.show()
```



In [26]:
```
continent = {'continent': [], 'sum': [], 'year': []}
years = list(map(str, list(range(1961,2022))))
print(years)
continents = ['Asia', 'Europe', 'Africa', 'Oceania', 'North America', 'South America']

k = 0

for i in years:
    for j in continents:
        continent['sum'].append(df_new[df_new['Continent'] == j][i].sum())
        continent['year'].append(years[k])

    continent['continent'] += continents

    k+=1
```

```
['1961', '1962', '1963', '1964', '1965', '1966', '1967', '1968', '1969', '1970', '1971',
'1972', '1973', '1974', '1975', '1976', '1977', '1978', '1979', '1980', '1981', '1982',
'1983', '1984', '1985', '1986', '1987', '1988', '1989', '1990', '1991', '1992', '1993',
'1994', '1995', '1996', '1997', '1998', '1999', '2000', '2001', '2002', '2003', '2004',
'2005', '2006', '2007', '2008', '2009', '2010', '2011', '2012', '2013', '2014', '2015',
'2016', '2017', '2018', '2019', '2020', '2021']
```

In [27]:
```
continents = pd.DataFrame(continent).sort_values('year')
continents.head(100)
```

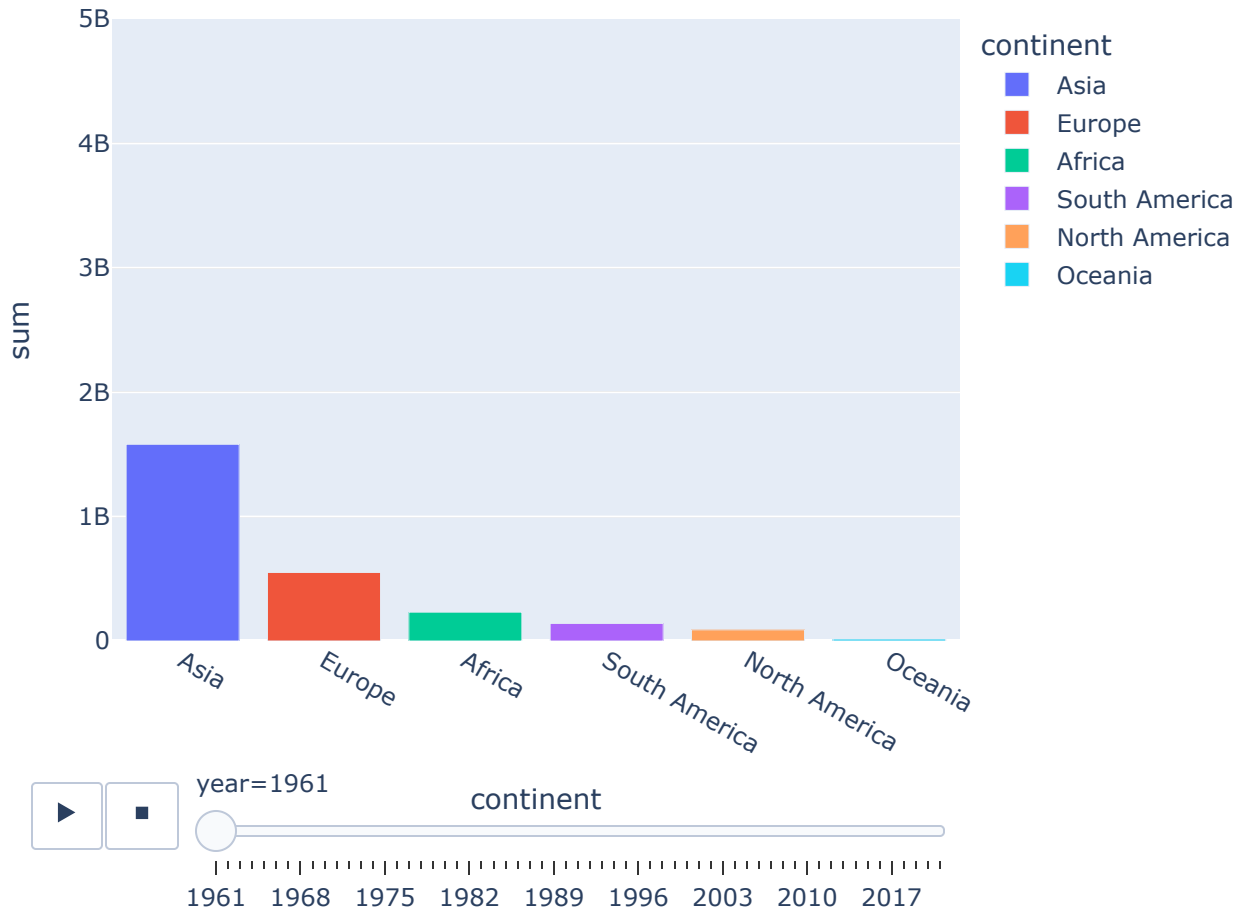
Out[27]:

	continent	sum	year

0	Asia	1.581046e+09	1961
1	Europe	5.510292e+08	1961
2	Africa	2.316143e+08	1961
3	Oceania	1.609791e+07	1961
4	North America	8.979225e+07	1961
...	...	...	...
90	Asia	2.229585e+09	1976
96	Asia	2.271625e+09	1977
97	Europe	6.186241e+08	1977
98	Africa	3.489870e+08	1977
99	Oceania	2.195606e+07	1977

100 rows × 3 columns

In [28]:
```
fig = px.bar(continents.sort_values(['year','sum'], ascending=[True, False]), x='contine
fig.update_yaxes(range=[0, 5000000000])
fig.show()
```



In [29]:
```
df_new.shape
```

Out[29]:
```
(249, 67)
```

In [30]:
```
df_t = df_new.T
```

In [31]:
```
df_t
```

Out[31]:

	0	1	2	3	4	5	6	7
Country Name	Afghanistan	Åland Islands	Albania	Algeria	American Samoa	Andorra	Angola	Anguilla
Country Code	AFG	0	ALB	DZA	ASM	AND	AGO	0
Indicator Name	Population, total	0	Population, total	Population, total	Population, total	Population, total	Population, total	0
Indicator Code	SP.POP.TOTL	0	SP.POP.TOTL	SP.POP.TOTL	SP.POP.TOTL	SP.POP.TOTL	SP.POP.TOTL	0
1960	8622466.0	0.0	1608800.0	11394307.0	20085.0	9443.0	5357195.0	0.0
...	...	...	...	...	...	...	...	...
2018	36686784.0	0.0	2866376.0	41927007.0	48424.0	75013.0	31273533.0	0.0
2019	37769499.0	0.0	2854191.0	42705368.0	47321.0	76343.0	32353588.0	0.0
2020	38972230.0	0.0	2837849.0	43451666.0	46189.0	77700.0	33428486.0	0.0
2021	40099462.0	0.0	2811666.0	44177969.0	45035.0	79034.0	34503774.0	0.0
Continent	Asia	Europe	Europe	Africa	Oceania	Europe	Africa	North America

67 rows × 249 columns

In [32]:
```
df_t.columns = df_t.iloc[0,:]
```

In [33]:
```
df_years=df_t.iloc[4:66,:]
```

In [34]:
```
df_years.columns = df_t.iloc[0,:]
```

In [35]:
```
df_years
```

Out[35]:

Country Name	Afghanistan	Åland Islands	Albania	Algeria	American Samoa	Andorra	Angola	Anguilla	Antarctic
1960	8622466.0	0.0	1608800.0	11394307.0	20085.0	9443.0	5357195.0	0.0	0
1961	8790140.0	0.0	1659800.0	11598608.0	20626.0	10216.0	5441333.0	0.0	0
1962	8969047.0	0.0	1711319.0	11778260.0	21272.0	11014.0	5521400.0	0.0	0
1963	9157465.0	0.0	1762621.0	11969451.0	21949.0	11839.0	5599827.0	0.0	0
1964	9355514.0	0.0	1814135.0	12179099.0	22656.0	12690.0	5673199.0	0.0	0
...	...	...	...	...	...	...	...	...	
2017	35643418.0	0.0	2873457.0	41136546.0	49463.0	73837.0	30208628.0	0.0	0
2018	36686784.0	0.0	2866376.0	41927007.0	48424.0	75013.0	31273533.0	0.0	0
2019	37769499.0	0.0	2854191.0	42705368.0	47321.0	76343.0	32353588.0	0.0	0
2020	38972230.0	0.0	2837849.0	43451666.0	46189.0	77700.0	33428486.0	0.0	0
2021	40099462.0	0.0	2811666.0	44177969.0	45035.0	79034.0	34503774.0	0.0	0

62 rows × 249 columns

Calculating the growth rate using the pct_change()

In [36]:
```
df_gr=df_years.pct_change()
```

In [37]:
```
df_gr.describe()
```

Out[37]:

Country Name	Afghanistan	Åland Islands	Albania	Algeria	American Samoa	Andorra	Angola	Anguilla	Antarctic
count	61.000000	0.0	61.000000	61.000000	61.000000	61.000000	61.000000	0.0	0
mean	0.026202	NaN	0.009304	0.022491	0.013528	0.035878	0.031044	NaN	Na
std	0.037623	NaN	0.015018	0.007546	0.020361	0.030169	0.008930	NaN	Na
min	-0.106629	NaN	-0.009341	0.013580	-0.024984	-0.031590	0.006984	NaN	Na
25%	0.020225	NaN	-0.006010	0.017237	-0.007283	0.015927	0.032791	NaN	Na
50%	0.025448	NaN	-0.000919	0.019988	0.019780	0.038640	0.034017	NaN	Na
75%	0.036412	NaN	0.023238	0.025254	0.030772	0.059841	0.036302	NaN	Na
max	0.161421	NaN	0.031701	0.049338	0.039447	0.084725	0.038303	NaN	Na

8 rows × 249 columns

In [38]:
```
df_gr=df_gr.T
```

In [39]:
```
df_gr
```

Out[39]:

Country Name	1960	1961	1962	1963	1964	1965	1966	1967	1968	
Afghanistan	NaN	0.019446	0.020353	0.021008	0.021627	0.022407	0.022791	0.023191	0.023751	0.0
Åland Islands	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Albania	NaN	0.031701	0.031039	0.029978	0.029226	0.027923	0.026696	0.026651	0.028833	0.0
Algeria	NaN	0.017930	0.015489	0.016233	0.017515	0.016599	0.018749	0.022494	0.022785	0.0
American Samoa	NaN	0.026936	0.031320	0.031826	0.032211	0.032442	0.031251	0.030097	0.030586	0.0
...	...	...	...	...	...	...	...	...	...	
Wallis and Futuna Islands	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Western Sahara	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Yemen	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Zambia	NaN	0.032064	0.032296	0.032483	0.032460	0.032534	0.032711	0.032721	0.032780	0.0
Zimbabwe	NaN	0.031433	0.031540	0.031644	0.031691	0.031742	0.031791	0.031854	0.032030	0.0

249 rows × 62 columns

In [40]:
```
df_gr['Continent']= df_new['Continent'].values
```

In [41]:
```
df_gr.reset_index(inplace=True)
df_gr = df_gr.rename(columns = {'index':'Country name'})
```

In [42]:
```
df_gr.fillna(0)
```

Out[42]:

	Country Name	1960	1961	1962	1963	1964	1965	1966	1967	1968
0	Afghanistan	0.0	0.019446	0.020353	0.021008	0.021627	0.022407	0.022791	0.023191	0.02375
1	Åland Islands	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2	Albania	0.0	0.031701	0.031039	0.029978	0.029226	0.027923	0.026696	0.026651	0.02883
3	Algeria	0.0	0.017930	0.015489	0.016233	0.017515	0.016599	0.018749	0.022494	0.02278
4	American Samoa	0.0	0.026936	0.031320	0.031826	0.032211	0.032442	0.031251	0.030097	0.03058
...	...	...	...	...	...	...	...	...	...	...
244	Wallis and Futuna Islands	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
245	Western Sahara	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
246	Yemen	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
247	Zambia	0.0	0.032064	0.032296	0.032483	0.032460	0.032534	0.032711	0.032721	0.03278
248	Zimbabwe	0.0	0.031433	0.031540	0.031644	0.031691	0.031742	0.031791	0.031854	0.03203

249 rows × 64 columns

In [43]:
```
continent = {'continent': [], 'average': [], 'year': []}
years = list(map(str, list(range(1961,2022))))
continents = ['Asia', 'Europe', 'Africa', 'Oceania', 'North America', 'South America']

k = 0

for i in years:
    for j in continents:
        continent['average'].append(df_gr[df_gr['Continent'] == j][i].mean())
        continent['year'].append(years[k])

    continent['continent'] += continents

    k+=1
```

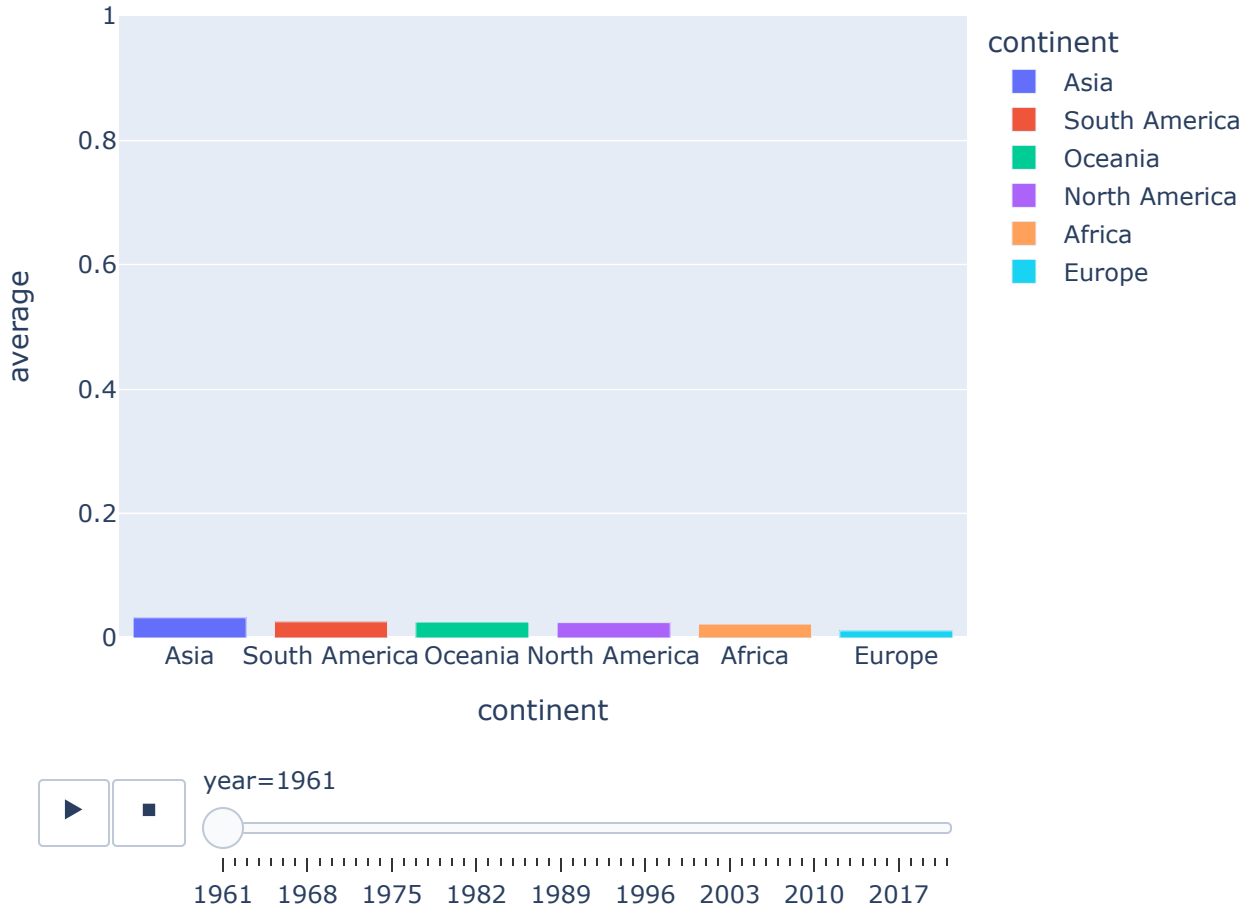
In [44]:
```
continents = pd.DataFrame(continent).sort_values('year')
continents.head()
```

Out[44]:

	continent	average	year
0	Asia	0.032542	1961
1	Europe	0.011525	1961
2	Africa	0.022411	1961

3	Oceania	0.025645	1961
4	North America	0.024730	1961

In [45]:
```
fig = px.bar(continents.sort_values(['year','average'], ascending=[True, False]), x='con
fig.update_yaxes(range=[0, 1])
fig.show()
```



In [46]:
```
pop_gr = df_gr.melt(id_vars=['Country Name','Continent'], value_vars=df_new.iloc[:,5:67]
pop_gr = pop_gr.sort_values('Year')
pop_gr.head(30)
```

Out[46]:

	Country Name	Continent	Year	Population_growth
0	Afghanistan	Asia	1961	0.019446
158	Nicaragua	North America	1961	0.030702
159	Niger	Africa	1961	0.030050
160	Nigeria	Africa	1961	0.020637
161	Niue	Oceania	1961	NaN
162	Norfolk Island	Oceania	1961	NaN
163	North Macedonia	Europe	1961	0.012818
164	Northern Mariana Islands	Oceania	1961	0.030223
165	Norway	Europe	1961	0.007975
166	Oman	Asia	1961	0.018432

167	Pakistan	Asia	1961	0.024082
168	Palau	Oceania	1961	0.020432
169	Panama	North America	1961	0.030030
170	Papua New Guinea	Oceania	1961	0.025183
171	Paraguay	South America	1961	0.024477
172	Peru	South America	1961	0.030071
173	Philippines	Asia	1961	0.030033
174	Pitcairn	Oceania	1961	NaN
175	Poland	Europe	1961	0.011018
176	Portugal	Europe	1961	0.008083
177	Puerto Rico	North America	1961	0.017694
178	Qatar	Asia	1961	0.102405
179	Republic of Korea	Asia	1961	NaN
180	Republic of Moldova	Europe	1961	NaN
181	Réunion	Africa	1961	NaN
182	Romania	Europe	1961	0.008059
183	Russian Federation	Europe	1961	0.011168
184	Rwanda	Africa	1961	0.027137
157	New Zealand	Oceania	1961	0.020196
185	Saint Barthélemy	North America	1961	NaN

In [47]:
```
df_total=pd.merge(pop, pop_gr, how='outer')
```

2. Group countries based on population, rate of population growth, and any other pertinent factors (clustering)

Clustering based on the population into low, moderate and high

In [48]:
```
df_total.loc[df_total.Population < 2000000000, 'Population_group'] = 'low'
df_total.loc[df_total.Population < 3000000000, 'Population_group'] = 'moderate'
df_total.loc[df_total.Population < 5000000000, 'Population_group'] = 'high'
```

In [49]:
```
df_total
```

Out[49]:

	Country Name	Continent	Year	Population	Population_growth	Population_group
0	Afghanistan	Asia	1961	8790140.0	0.019446	high
1	Nicaragua	North America	1961	1844630.0	0.030702	high
2	Niger	Africa	1961	3602530.0	0.030050	high
3	Nigeria	Africa	1961	45855507.0	0.020637	high
4	Niue	Oceania	1961	0.0	NaN	high
...	...	...	...	...	...	...
15184	Ghana	Africa	2021	32833031.0	0.020280	high

15185	Gibraltar	Europe	2021	32669.0	-0.001223	high
15186	Greece	Europe	2021	10641221.0	-0.005363	high
15187	Faroe Islands	Europe	2021	52889.0	0.009043	high
15188	Zimbabwe	Africa	2021	15993524.0	0.020668	high

15189 rows × 6 columns

Clustering based on the population growth rate into low, moderate and high

In [50]:
```
df_total.loc[df_total.Population_growth < 0.3, 'rate_pop_growth'] = 'low'
df_total.loc[df_total.Population_growth< 0.6, 'rate_pop_growth'] = 'moderate'
df_total.loc[df_total.Population_growth<=1, 'rate_pop_growth'] = 'high'
```

In [51]:
```
df_total.fillna(0)
```

Out[51]:

	Country Name	Continent	Year	Population	Population_growth	Population_group	rate_pop_growth
0	Afghanistan	Asia	1961	8790140.0	0.019446	high	high
1	Nicaragua	North America	1961	1844630.0	0.030702	high	high
2	Niger	Africa	1961	3602530.0	0.030050	high	high
3	Nigeria	Africa	1961	45855507.0	0.020637	high	high
4	Niue	Oceania	1961	0.0	0.000000	high	0
...	...	...	...	...	...	...	...
15184	Ghana	Africa	2021	32833031.0	0.020280	high	high
15185	Gibraltar	Europe	2021	32669.0	-0.001223	high	high
15186	Greece	Europe	2021	10641221.0	-0.005363	high	high
15187	Faroe Islands	Europe	2021	52889.0	0.009043	high	high
15188	Zimbabwe	Africa	2021	15993524.0	0.020668	high	high

15189 rows × 7 columns

3. redict the world population in the next 15, 30, and 50 years

Predicting the population after 15 years, 30 years and 50 years

In [52]:
```
def predicting_future_pop(present_pop, growth_rate, num_years):
    future_pop = present_pop + (present_pop*(1+growth_rate)**num_years)
    return future_pop
```

In [53]:
```
# Current world population

# growth rate after 15 years
#df_total['Population_after_15years']= df_total['Population'] + (df_total['Population']*
# df_total['Population_after_15years']= df_total['Population'] + df_total['Population_af
#df_total['Population_after_30years']= df_total['Population'] + (df_total['Population']*
```

```
df_total['Population_after_15years']= predicting_future_pop(df_total['Population'],df_to
df_total['Population_after_30years']= predicting_future_pop(df_total['Population'],df_to
df_total['Population_after_50years']= predicting_future_pop(df_total['Population'],df_to
```

In [54]:
```
df_total
```

Out[54]:

	Country Name	Continent	Year	Population	Population_growth	Population_group	rate_pop_growth
0	Afghanistan	Asia	1961	8790140.0	0.019446	high	high
1	Nicaragua	North America	1961	1844630.0	0.030702	high	high
2	Niger	Africa	1961	3602530.0	0.030050	high	high
3	Nigeria	Africa	1961	45855507.0	0.020637	high	high
4	Niue	Oceania	1961	0.0	NaN	high	NaN
...	...	...	...	...	...	...	...
15184	Ghana	Africa	2021	32833031.0	0.020280	high	high
15185	Gibraltar	Europe	2021	32669.0	-0.001223	high	high
15186	Greece	Europe	2021	10641221.0	-0.005363	high	high
15187	Faroe Islands	Europe	2021	52889.0	0.009043	high	high
15188	Zimbabwe	Africa	2021	15993524.0	0.020668	high	high

15189 rows × 10 columns

In [55]:
```
future_years =['2036','2051','2081']
df_total.rename(columns={'Population_after_15years':'2036','Population_after_30years':'2
```

In [56]:
```
df_total
```

Out[56]:

	Country Name	Continent	Year	Population	Population_growth	Population_group	rate_pop_growth
0	Afghanistan	Asia	1961	8790140.0	0.019446	high	high
1	Nicaragua	North America	1961	1844630.0	0.030702	high	high
2	Niger	Africa	1961	3602530.0	0.030050	high	high
3	Nigeria	Africa	1961	45855507.0	0.020637	high	high
4	Niue	Oceania	1961	0.0	NaN	high	NaN
...	...	...	...	...	...	...	...
15184	Ghana	Africa	2021	32833031.0	0.020280	high	high
15185	Gibraltar	Europe	2021	32669.0	-0.001223	high	high
15186	Greece	Europe	2021	10641221.0	-0.005363	high	high
15187	Faroe Islands	Europe	2021	52889.0	0.009043	high	high
15188	Zimbabwe	Africa	2021	15993524.0	0.020668	high	high

15189 rows × 10 columns

In [57]:
```
pop_pred = df_total.melt(id_vars=['Country Name','Continent'], value_vars=df_total.iloc[
pop_pred = pop_pred.sort_values('Year')
pop_pred.head(30)
```

Out[57]:

	Country Name	Continent	Year	Predicted population
0	Afghanistan	Asia	2036	2.052452e+07
10120	Albania	Europe	2036	5.718510e+06
10121	Algeria	Africa	2036	6.955424e+07
10122	American Samoa	Oceania	2036	1.180763e+05
10123	Andorra	Europe	2036	1.675896e+05
10124	Angola	Africa	2036	4.467267e+07
10125	Anguilla	North America	2036	NaN
10126	Antarctica	Antarctica	2036	NaN
10127	Antigua and Barbuda	North America	2036	1.721446e+05
10128	Argentina	South America	2036	8.167924e+07
10129	Armenia	Asia	2036	5.780465e+06
10130	Curaçao	North America	2036	NaN
10131	Aruba	North America	2036	2.089375e+05
10132	Austria	Europe	2036	1.655989e+07
10133	Azerbaijan	Asia	2036	1.722200e+07
10134	Bahamas	North America	2036	NaN
10135	Bahrain	Asia	2036	1.810494e+06
10136	Bangladesh	Asia	2036	3.067413e+08
10137	Barbados	North America	2036	5.417921e+05
10138	Belarus	Europe	2036	1.912180e+07
10139	Belgium	Europe	2036	2.111778e+07
10140	Belize	North America	2036	6.460685e+05
10141	Benin	Africa	2036	1.854388e+07
10142	Bermuda	North America	2036	1.359926e+05
10143	Bhutan	Asia	2036	1.506685e+06
10144	Bolivia (Plurinational State of)	South America	2036	NaN
10119	Åland Islands	Europe	2036	NaN
10118	Afghanistan	Asia	2036	4.169692e+07
10117	Latvia	Europe	2036	4.262732e+06
10116	Bonaire, Sint Eustatius and Saba	North America	2036	NaN

In [58]:
```
fig = px.choropleth(pop_pred,
              locations = 'Country Name',
              color="Predicted population",
              animation_frame="Year",
              color_continuous_scale='Turbo',
              locationmode='country names',
              title='Total Population Over the World Animation (After 15 years(2036), Af
```

```
                height=600
               )
fig.show()
```