Muhammad Suleman 2023 10Alytics Alumni Hackathon Entry

May 14, 2023

1 From Numbers to Knowledge: A Journey Through Poverty, GDP, and Life Expectancy in Africa

Muhammad Suleman, (10Alytics Alumni)

2 Exploratory Data ANalysis

[2]:
```
# Import necessary libraries

# Data analysis libraries
import pandas as pd
import numpy as np

# data visualization libraries
import matplotlib.pyplot as plt
import seaborn as sns

# data preprocessing
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler



import warnings
warnings.filterwarnings("ignore")
```

2.1 Data Cleaning

[3]:
```
# read in the poverty dataset
df1 = pd.read_excel(r"C:\Users\PC\Desktop\hack\Poverty line data.xlsx")

# read in the GDP dataset
df2 = pd.read_excel(r"C:\Users\PC\Desktop\hack\gdp py.xlsx")

# read in the life expectancy dataset
df3 = pd.read_excel(r"C:\Users\PC\Desktop\hack\life-expectancy.xlsx")

# read in the country code dataset
```

```
df4 = pd.read_excel(r"C:\Users\PC\Desktop\hack\Country Code Data.xlsx")
```

[4]:
```
df1.head()
```

[4]:

	Country Code	Year	Poverty Line	Number of People
0	DZA	1988	$40	156887
1	DZA	1988	$30-$40	156293
2	DZA	1988	$20-$30	505149
3	DZA	1988	$10-$20	3346519
4	DZA	1988	$6.85-$10	4721041

[5]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2151 entries, 0 to 2150
Data columns (total 4 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   Country Code      2151 non-null   object
 1   Year              2151 non-null   int64 
 2   Poverty Line      2151 non-null   object
 3   Number of People  2151 non-null   int64 
dtypes: int64(2), object(2)
memory usage: 67.3+ KB
```

[6]:
```
df1.describe()
```

[6]:

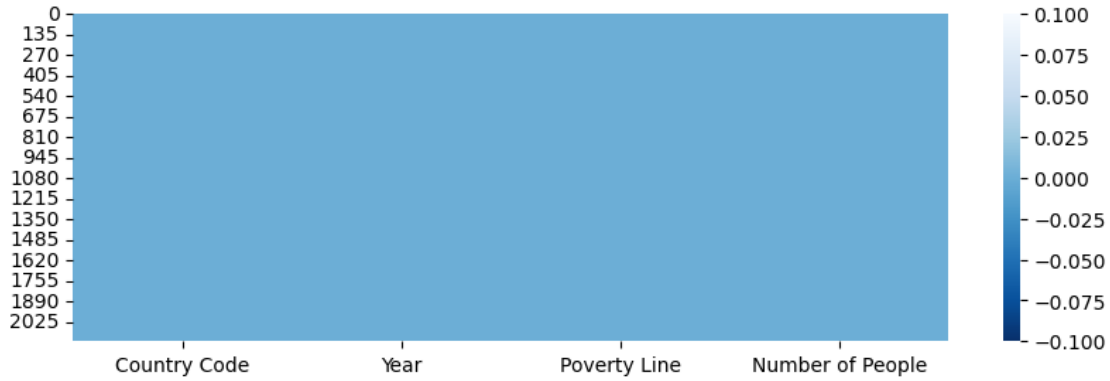
	Year	Number of People
count	2151.000000	2.151000e+03
mean	2004.648536	2.446894e+06
std	9.477668	6.058364e+06
min	1980.000000	0.000000e+00
25%	1997.000000	4.224350e+04
50%	2005.000000	3.173360e+05
75%	2013.000000	2.314042e+06
max	2019.000000	6.387384e+07

[7]:
```
# Check for missing values in the dataframe
missing_values_count = df1.isnull().sum()
print(missing_values_count)


# Visualize missing values using a heatmap
plt.figure(figsize = (10,3))
sns.heatmap(df1.isnull(),cbar = True, cmap = "Blues_r")
```

```
Country Code        0
Year                0
Poverty Line        0
```

```
Number of People    0
dtype: int64
```

[7]: <AxesSubplot:>



[8]:
```
df2.head()
```

[8]:

	Country Code	Year	GDP
0	AGO	1980	5.930503e+09
1	AGO	1981	5.550483e+09
2	AGO	1982	5.550483e+09
3	AGO	1983	5.784342e+09
4	AGO	1984	6.131475e+09

[9]:
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2853 entries, 0 to 2852
Data columns (total 3 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Country Code  2853 non-null   object
 1   Year          2853 non-null   int64
 2   GDP           2853 non-null   float64
dtypes: float64(1), int64(1), object(1)
memory usage: 67.0+ KB
```

[10]:
```
df2.describe()
```

[10]:

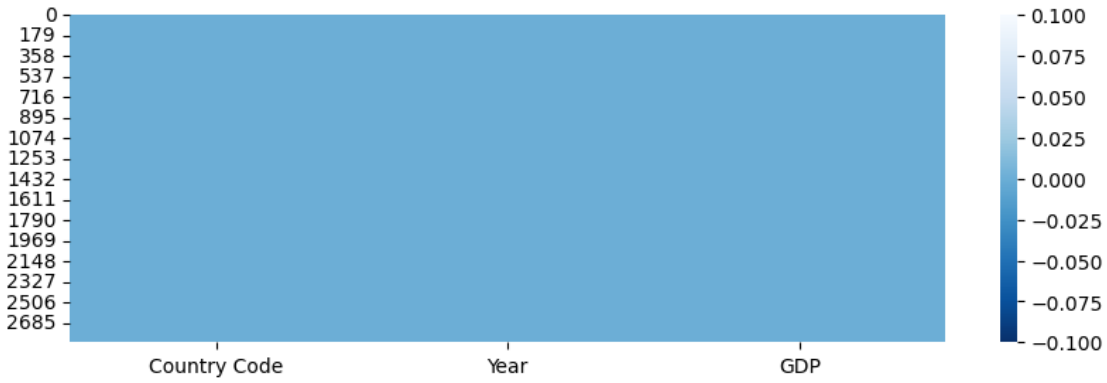
	Year	GDP
count	2853.000000	2.853000e+03
mean	1992.913775	1.827951e+10
std	17.411535	5.178105e+10
min	1960.000000	9.122751e+06

```
25%     1979.000000  9.326486e+08
50%     1994.000000  3.259345e+09
75%     2008.000000  1.184019e+10
max     2021.000000  5.741838e+11
```

[11]:
```
# Check for missing values in the dataframe
missing_values_count = df2.isnull().sum()
print(missing_values_count)


# Visualize missing values using a heatmap
plt.figure(figsize = (10,3))
sns.heatmap(df2.isnull(),cbar = True, cmap = "Blues_r")
```

```
Country Code    0
Year            0
GDP             0
dtype: int64
```

[11]: <AxesSubplot:>



[12]:
```
df3.head()
```

[12]:

	Country Code	Year	Life expectancy at birth (historical)
0	DZA	1923	28.82
1	DZA	1933	31.22
2	DZA	1943	33.72
3	DZA	1950	42.40
4	DZA	1951	42.50

[13]:
```
df3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3937 entries, 0 to 3936
```

```
Data columns (total 3 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   Country Code                            3937 non-null   object
 1   Year                                    3937 non-null   int64
 2   Life expectancy at birth (historical)  3937 non-null   float64
dtypes: float64(1), int64(1), object(1)
memory usage: 92.4+ KB
```

[14]:
```
df3.describe()
```

[14]:

	Year	Life expectancy at birth (historical)
count	3937.000000	3937.000000
mean	1984.886208	51.123172
std	21.384480	10.307799
min	1921.000000	12.400000
25%	1967.000000	43.400000
50%	1985.000000	50.700000
75%	2003.000000	58.700000
max	2021.000000	76.600000

[15]:
```
df4.head()
```

[15]:

	Country Code	Country Name	IncomeGroup	Region
0	AGO	Angola	Lower middle income	Middle Africa
1	BDI	Burundi	Low income	Eastern Africa
2	BEN	Benin	Lower middle income	Western Africa
3	BFA	Burkina Faso	Low income	Western Africa
4	BWA	Botswana	Upper middle income	Southern Africa

[16]:
```
df4.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54 entries, 0 to 53
Data columns (total 4 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Country Code  54 non-null     object
 1   Country Name  54 non-null     object
 2   IncomeGroup   54 non-null     object
 3   Region        54 non-null     object
dtypes: object(4)
memory usage: 1.8+ KB
```

[17]:
```
df4.describe()
```

[17]:

	Country Code	Country Name	IncomeGroup	Region
count	54	54	54	54

```
unique          54          54           4                5
top            AGO      Angola  Low income  Eastern Africa
freq             1           1          24               18
```

[18]:
```
# Check for missing values in the dataframe
missing_values_count = df4.isnull().sum()
print(missing_values_count)


# Visualize missing values using a heatmap
plt.figure(figsize = (10,3))
sns.heatmap(df4.isnull(),cbar = True, cmap = "Blues_r")
```

```
Country Code    0
Country Name    0
IncomeGroup     0
Region          0
dtype: int64
```

[18]: <AxesSubplot:>

[19]:
```
# Create data model

merged_df1 = pd.merge(df1, df4, on='Country Code', how='left')
merged_df2 = pd.merge(df2, df4, on='Country Code', how='left')
merged_df3 = pd.merge(df3, df4, on='Country Code', how='left')
```

[20]:
```
merged_df1.head()
```

[20]:
```
  Country Code  Year Poverty Line  Number of People Country Name  \
0          DZA  1988          $40            156887      Algeria
1          DZA  1988      $30-$40            156293      Algeria
2          DZA  1988      $20-$30            505149      Algeria
```

```
3          DZA  1988      $10-$20           3346519      Algeria
4          DZA  1988    $6.85-$10           4721041      Algeria

           IncomeGroup           Region
0  Lower middle income  Northern Africa
1  Lower middle income  Northern Africa
2  Lower middle income  Northern Africa
3  Lower middle income  Northern Africa
4  Lower middle income  Northern Africa
```

[23]:
```
# Group merged_df1 by 'Country Name' and select data between 1990 and 2019
grouped_df1 = merged_df1.loc[(merged_df1['Year'] >= 1990) & (merged_df1['Year']␣
↪<= 2019)]\
                .groupby(['Country Name'])[['Number of People']].sum()

# Print the first 5 rows of the grouped_df
print(grouped_df1.head())
```

```
              Number of People
Country Name
Algeria                65419226
Angola                 68900900
Benin                  39042382
Botswana                7177958
Burkina Faso           85918523
```

[24]:
```
merged_df2.head()
```

[24]:
```
  Country Code  Year           GDP Country Name          IncomeGroup  \
0          AGO  1980  5.930503e+09       Angola  Lower middle income
1          AGO  1981  5.550483e+09       Angola  Lower middle income
2          AGO  1982  5.550483e+09       Angola  Lower middle income
3          AGO  1983  5.784342e+09       Angola  Lower middle income
4          AGO  1984  6.131475e+09       Angola  Lower middle income

          Region
0  Middle Africa
1  Middle Africa
2  Middle Africa
3  Middle Africa
4  Middle Africa
```

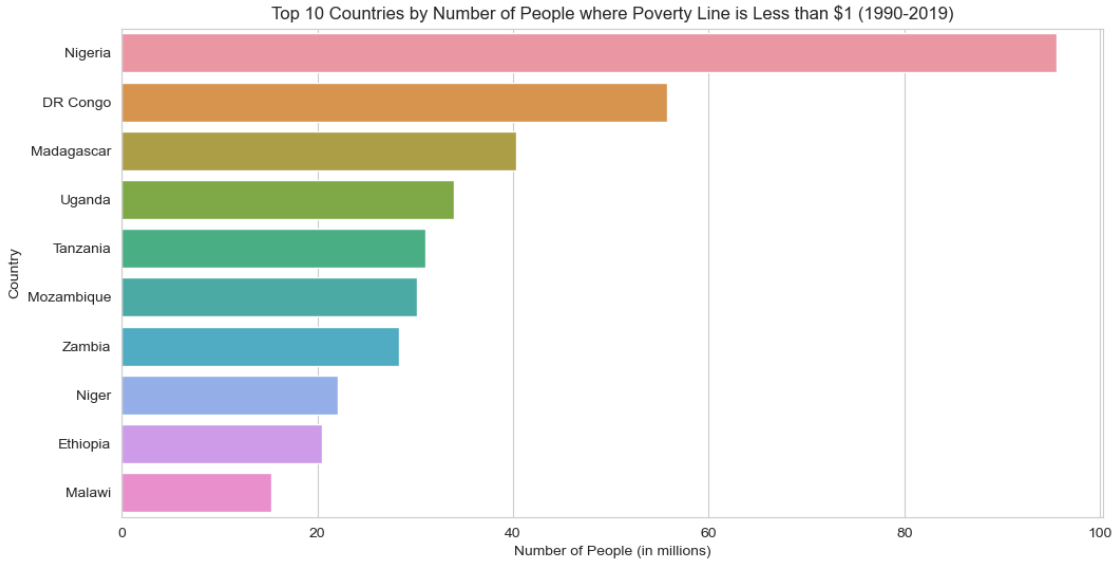
[25]:
```
merged_df3.head()
```

[25]:
```
  Country Code  Year  Life expectancy at birth (historical) Country Name  \
0          DZA  1923                                 28.82      Algeria
1          DZA  1933                                 31.22      Algeria
2          DZA  1943                                 33.72      Algeria
```

```
3          DZA  1950                                    42.40      Algeria
4          DZA  1951                                    42.50      Algeria

           IncomeGroup           Region
0  Lower middle income  Northern Africa
1  Lower middle income  Northern Africa
2  Lower middle income  Northern Africa
3  Lower middle income  Northern Africa
4  Lower middle income  Northern Africa
```

2.2 Poverty rate exploration

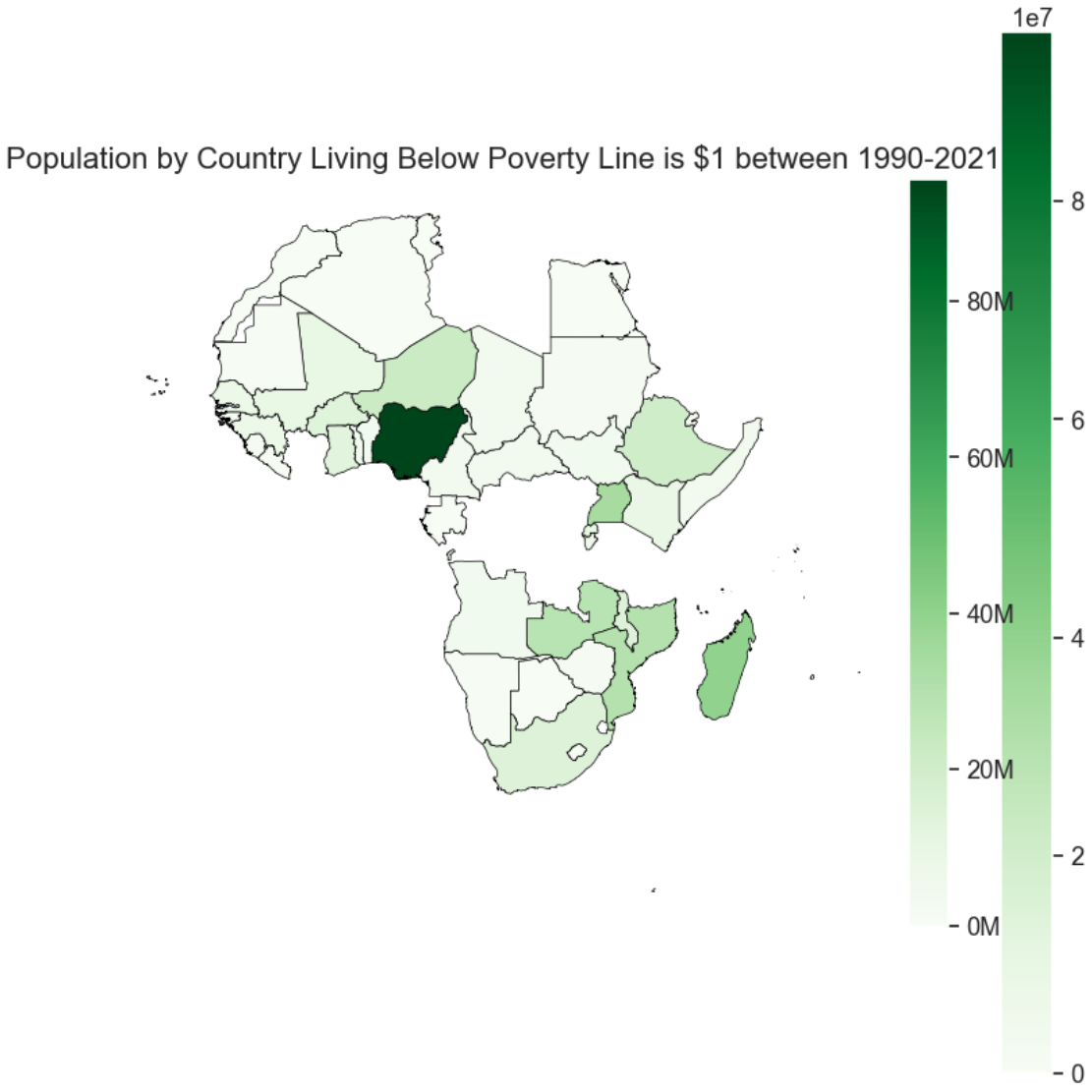
[141]:
```
# Subset the data to include only poverty line "Less than $1" and year range␣
 ↪1990-2019
df_sub = merged_df1[(merged_df1['Poverty Line'] == '$1') & (merged_df1['Year'].
 ↪between(1990, 2019))]

# Group the data by country and sum the number of people
df_sub = df_sub.groupby('Country Name')['Number of People'].sum().reset_index()

# Sort the data by number of people and select the top 10 countries
df_sub = df_sub.sort_values('Number of People', ascending=False).head(10)

# Convert number of people to millions for better readability
df_sub['Number of People'] = df_sub['Number of People'] / 1000000

# Set the color palette
sns.set_palette(['#FFC300', '#FF5733', '#C70039', '#900C3F'])

# Create the barplot
plt.figure(figsize=(12,6))
ax = sns.barplot(y='Country Name', x='Number of People', data=df_sub)

# Set the axis labels and title
ax.set_ylabel('Country')
ax.set_xlabel('Number of People (in millions)')
ax.set_title('Top 10 Countries by Number of People where Poverty Line is Less␣
 ↪than $1 (1990-2019)')

plt.show()
```



- We notice from the chart above, Nigeria has the highest number of people living below poverty line within the year 1990 and 2019.

[110]:
```
import geopandas as gpd
import matplotlib.pyplot as plt
import matplotlib.ticker as mtick
from matplotlib.colors import Normalize
from mpl_toolkits.axes_grid1 import make_axes_locatable

# Load the shapefile of African countries
africa = gpd.read_file(r"C:
 ↪\Users\PC\Downloads\ne_10m_admin_0_countries\ne_10m_admin_0_countries.shp")
africa = africa.rename(columns={'SOVEREIGNT': 'Country Name'})

# Filter the data to include rows where Poverty Line is $1 and Year is between␣
 ↪1990 and 2021
df_filtered = merged_df1[(merged_df1['Poverty Line'] == '$1') &␣
 ↪(merged_df1['Year'].between(1990, 2021))]

# Aggregate the data to get the total Number of People by Country Name
df_grouped = df_filtered.groupby('Country Name').agg({'Number of People':␣
 ↪'sum'}).reset_index()

# Merge the shapefile with the data
merged_map = africa.merge(df_grouped, on='Country Name', how='left')

# Set the figure size and title
fig, ax = plt.subplots(figsize=(10,10))
```

```
ax.set_title('Population by Country Living Below Poverty Line is $1 between␣
 ↪1990-2021', fontsize=16)

# Plot the map
cmap = 'Greens'
merged_map.plot(column='Number of People', cmap=cmap, linewidth=0.5,␣
 ↪edgecolor='black', legend=True, ax=ax)

# Remove the axis
ax.axis('off')

# Set up the colorbar
vmin, vmax = merged_map['Number of People'].min(), merged_map['Number of␣
 ↪People'].max()
norm = Normalize(vmin=vmin, vmax=vmax)
sm = plt.cm.ScalarMappable(cmap=cmap, norm=norm)
sm.set_array([])
divider = make_axes_locatable(ax)
cax = divider.append_axes('right', size='5%', pad=0.1)
cb = plt.colorbar(sm, cax=cax)
cb.ax.yaxis.set_major_formatter(mtick.FuncFormatter(lambda x, pos: f'{x/1000000:
 ↪.0f}M'))

# Show the plot
plt.show()
```



- More than 80 million people are living under the poverty line in Nigeria, almost twice as much as in Madagascar
- Apart from Nigeria, all other countries in Africa according to the data have around 40M people and below under the poverty Line

It makes me inquisitive about the proportion of population under the poverty line in each country.

[97]:
```
# Filter the data to include only the Poverty Line of $1
df_povline_1 = merged_df1[merged_df1['Poverty Line'] == '$1']

# Calculate the total number of people living below the poverty line of $1 for␣
↪each country
```

```
total_povline_1 = df_povline_1.groupby('Country Name')['Number of People'].
 ↪sum().reset_index()

# Calculate the total number of people for each country
total_pop = merged_df1.groupby('Country Name')['Number of People'].sum().
 ↪reset_index()

# Merge the two dataframes on the Country Name column
df_merged = total_povline_1.merge(total_pop, on='Country Name')

# Calculate the proportion of people living below the poverty line of $1 for␣
 ↪each country
df_merged['Proportion of People Living Below Poverty Line of $1'] =␣
 ↪df_merged['Number of People_x'] / df_merged['Number of People_y']

# Display the resulting dataframe
df_merged.head()
```

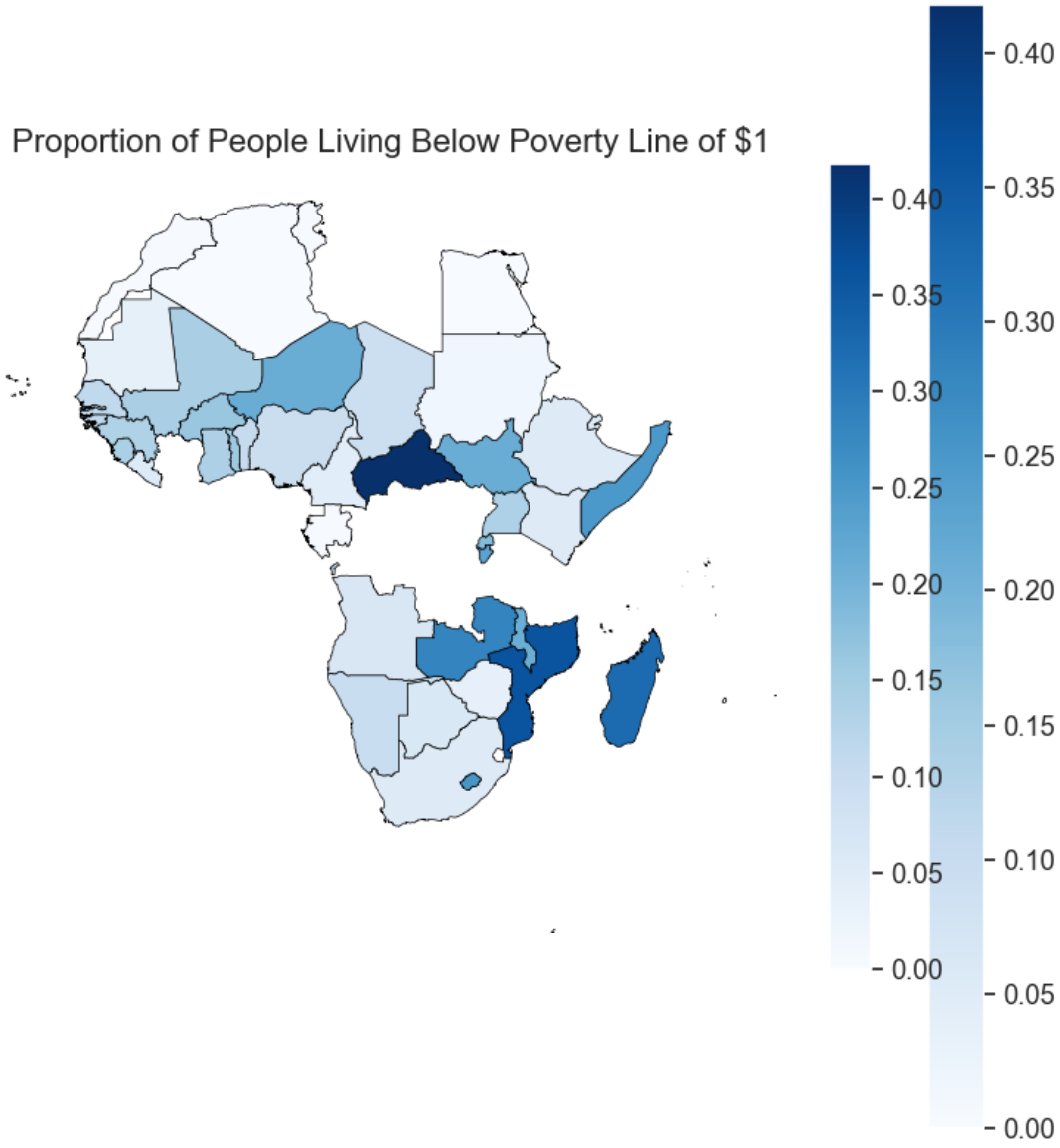
[97]:

	Country Name	Number of People_x	Number of People_y	\
0	Algeria	100557	89862698	
1	Angola	4431964	68900900	
2	Benin	4073749	39042382	
3	Botswana	516521	8247543	
4	Burkina Faso	13893856	85918523	

	Proportion of People Living Below Poverty Line of $1
0	0.001119
1	0.064324
2	0.104342
3	0.062627
4	0.161710

[99]:
```
import geopandas as gpd
import plotly.express as px

# Load the shapefile of African countries
africa = gpd.read_file(r"C:
 ↪\Users\PC\Downloads\ne_10m_admin_0_countries\ne_10m_admin_0_countries.shp")
africa = africa.rename(columns={'SOVEREIGNT': 'Country Name'})


# Merge the shapefile with the data
merged_map = africa.merge(df_merged, on='Country Name', how='left')

# Set the figure size and title
fig, ax = plt.subplots(figsize=(10,10))
```

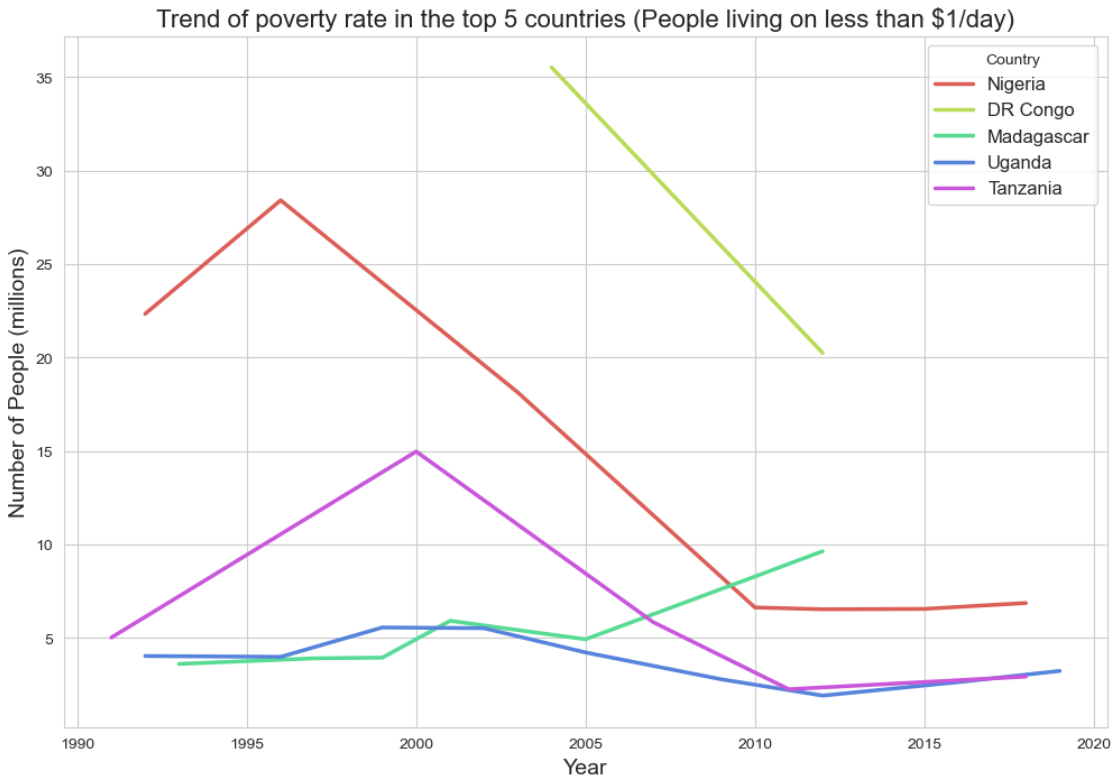
```
ax.set_title('Proportion of People Living Below Poverty Line of $1',␣
 ↪fontsize=16)

# Plot the map
cmap = 'Blues'
merged_map.plot(column='Proportion of People Living Below Poverty Line of $1',␣
 ↪cmap=cmap, linewidth=0.5, edgecolor='black', legend=True, ax=ax)

# Remove the axis
ax.axis('off')

# Set up the colorbar
vmin, vmax = merged_map['Proportion of People Living Below Poverty Line of $1'].
 ↪min(), merged_map['Proportion of People Living Below Poverty Line of $1'].
 ↪max()
norm = Normalize(vmin=vmin, vmax=vmax)
sm = plt.cm.ScalarMappable(cmap=cmap, norm=norm)
sm.set_array([])
divider = make_axes_locatable(ax)
cax = divider.append_axes('right', size='5%', pad=0.1)
cb = plt.colorbar(sm, cax=cax)


# Show the plot
plt.show()
```



- Looking at the data from a proportion point of view, we can clearly see that the high population in Nigeria accounts for the 80M+ people living below the poverty line.
- Central African Republic has the highest proportion of people living below the poverty line.
- Madagascar and Mozambique follow central African Republic closely in terms of proportion of Population Under the poverty line.

[152]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Filter for poverty line less than $1 and years between 1990 and 2019
merged_df1_filtered = merged_df1[(merged_df1['Poverty Line'] == '$1') &␣
 ↪(merged_df1['Year'].between(1990, 2019))]
```

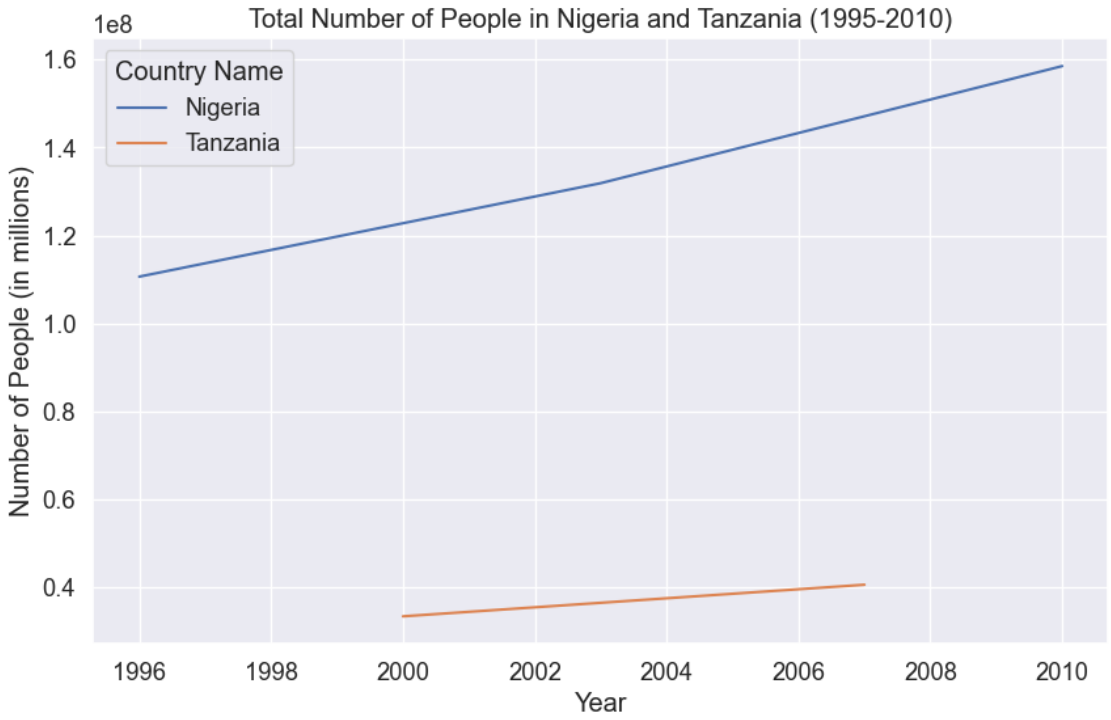
```
# Get the top 10 countries by number of people
top10 = merged_df1_filtered.groupby('Country Name')['Number of People'].sum().
 ↪nlargest(5).index.tolist()

# Create a pivot table for the top 10 countries
merged_df1_filtered_top10 = merged_df1_filtered[merged_df1_filtered['Country␣
 ↪Name'].isin(top10)]
merged_df1_filtered_top10_pivot = merged_df1_filtered_top10.pivot(index='Year',␣
 ↪columns='Country Name', values='Number of People')

# Set the color scheme
colors = sns.color_palette("hls", len(top10))

# Create the plot
fig, ax = plt.subplots(figsize=(12,8))
for i, country in enumerate(top10):
    sns.lineplot(data=merged_df1_filtered_top10_pivot[country]/1e6,␣
 ↪color=colors[i], ax=ax, linewidth=2.5, label=country)

# Set the axis labels and legend
ax.set_xlabel('Year', fontsize=14)
ax.set_ylabel('Number of People (millions)', fontsize=14)
ax.set_title('Trend of poverty rate in the top 5 countries (People living on␣
 ↪less than $1/day)', fontsize=16)
ax.legend(title='Country', fontsize=12)

# Show the plot
plt.show()
```



- Trend lines for Madagascar and DR Congo were truncated at around year 2012 due to incompleteness of the data provided.
- Nigeria and Tanzania experience a decline in the late 1990s and early 2000s with Tanzania's declined starting at exactly year 2000

[111]:
```
import matplotlib.pyplot as plt

# Select the countries and the years
countries = ['Nigeria', 'Tanzania']
years = list(range(1995, 2011))

# Filter the data to include only rows for the selected countries and years
df_filtered = merged_df1[(merged_df1['Country Name'].isin(countries)) &␣
 ↪(merged_df1['Year'].isin(years))]

# Group the data by country and year, and sum the number of people
df_grouped = df_filtered.groupby(['Country Name', 'Year']).sum().reset_index()

# Create a line plot of total number of people by year and country
fig, ax = plt.subplots(figsize=(10, 6))
sns.lineplot(x='Year', y='Number of People', hue='Country Name',␣
 ↪data=df_grouped, ax=ax)
```

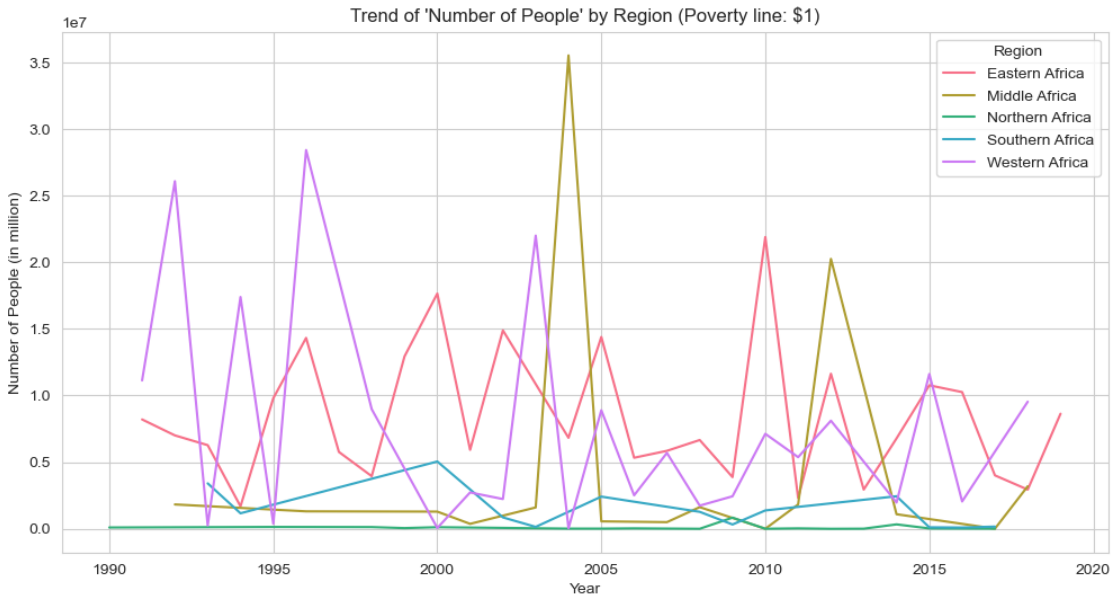
```
ax.set_title('Total Number of People in Nigeria and Tanzania (1995-2010)')
ax.set_xlabel('Year')
ax.set_ylabel('Number of People (in millions)')
plt.show()
```



- We can see that Nigeria and Tanzania (2000 - 2007) has done a great job in reducing the number of people below the poverty line given that they both experienced a rise in total population in the past few decades.
- According to https://www.bing.com/ck/a?!&&p=60d3e58c1154bf32JmltdHM9MTY4NDAyMjQwMCZpZ3V
877b-6885-3de0-14e2866669e6&psq=how+did+nigeria+alleaviate+poverty+between+1995+to+2010+usin-
and https://www.bing.com/ck/a?!&&p=b455beeedb9f9d42JmltdHM9MTY4NDAyMjQwMCZpZ3VpZD0wZ
877b-6885-3de0-14e2866669e6&psq=how+did+nigeria+alleaviate+poverty+between+1995+to+2010&u=a1
, Nigeria implemented several poverty alleviation policies within these years including NEEDS(National Economic Empowerment and Development Strategy), NAPEP(National Poverty Alleviation Program), etc…

[153]:
```
# Filter the data for poverty line is $1 and years between 1990 and 2021
filtered_data = merged_df1[(merged_df1['Poverty Line'] == '$1') &␣
↪(merged_df1['Year'] >= 1990) & (merged_df1['Year'] <= 2021)]

# Group the data by Region and Year
grouped_data = filtered_data.groupby(['Region', 'Year']).sum().reset_index()

# Visualize the trend of 'Number of People' by Region
```

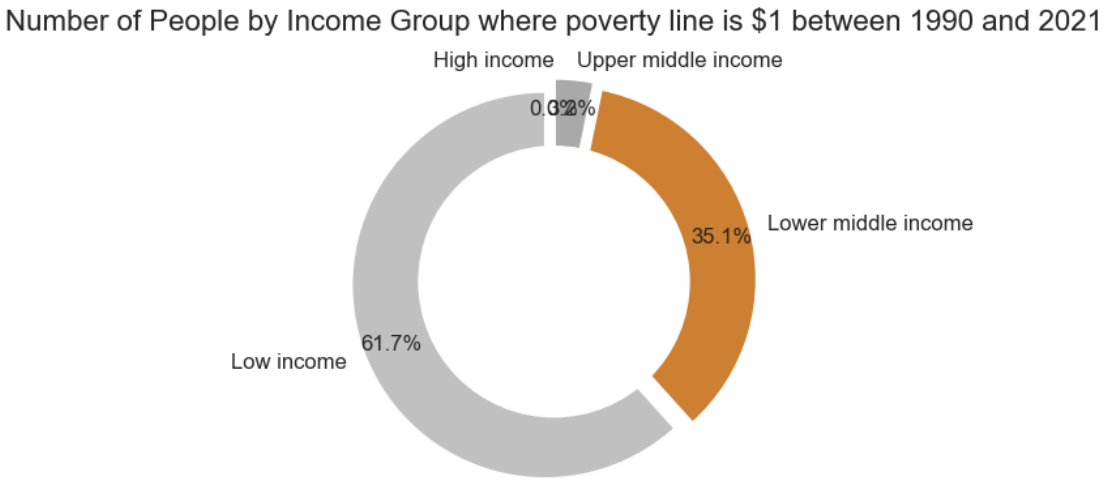
```
plt.figure(figsize=(12,6))
sns.lineplot(data=grouped_data, x='Year', y='Number of People', hue='Region')
plt.title("Trend of 'Number of People' by Region (Poverty line: $1)")
plt.xlabel('Year')
plt.ylabel('Number of People (in million)')
plt.show()
```



- Northern and Southern Africa show a low population below poverty line as well as intermittent declines along the years.
- There is a spike in poverty line in middle Africa in the year 2004

[156]:
```
# Filter for rows where poverty line is '$1' and year is between 1990 and 2021
df_filtered = merged_df1[(merged_df1['Poverty Line'] == '$1') &␣
 ↪(merged_df1['Year'] >= 1990) & (merged_df1['Year'] <= 2021)]

# Aggregate the data by IncomeGroup and sum the Number of People
df_grouped = df_filtered.groupby('IncomeGroup')['Number of People'].sum().
 ↪reset_index()

# Create a list of colors to use in the chart
colors = ['#FFD700', '#C0C0C0', '#CD7F32', '#A9A9A9']

# Create the doughnut chart
plt.pie(df_grouped['Number of People'], labels=df_grouped['IncomeGroup'],␣
 ↪colors=colors, autopct='%1.1f%%', startangle=90, pctdistance=0.85,␣
 ↪explode=[0.05, 0.05, 0.05, 0.05], textprops={'fontsize': 12})
```

```
# Add a circle to create a doughnut chart
centre_circle = plt.Circle((0,0),0.70,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)

# Add a title
plt.title('Number of People by Income Group where poverty line is $1 between␣
 ↪1990 and 2021', fontsize=16)

# Show the chart
plt.show()
```



- Most of the population across board between 1990 and 2021 fall in the Low income group.
- From the data Seychelles (the Outlier) formed the High income group, forming 0% of the data between 1990 to 2021 while the Upper middle group constitute 3.2%

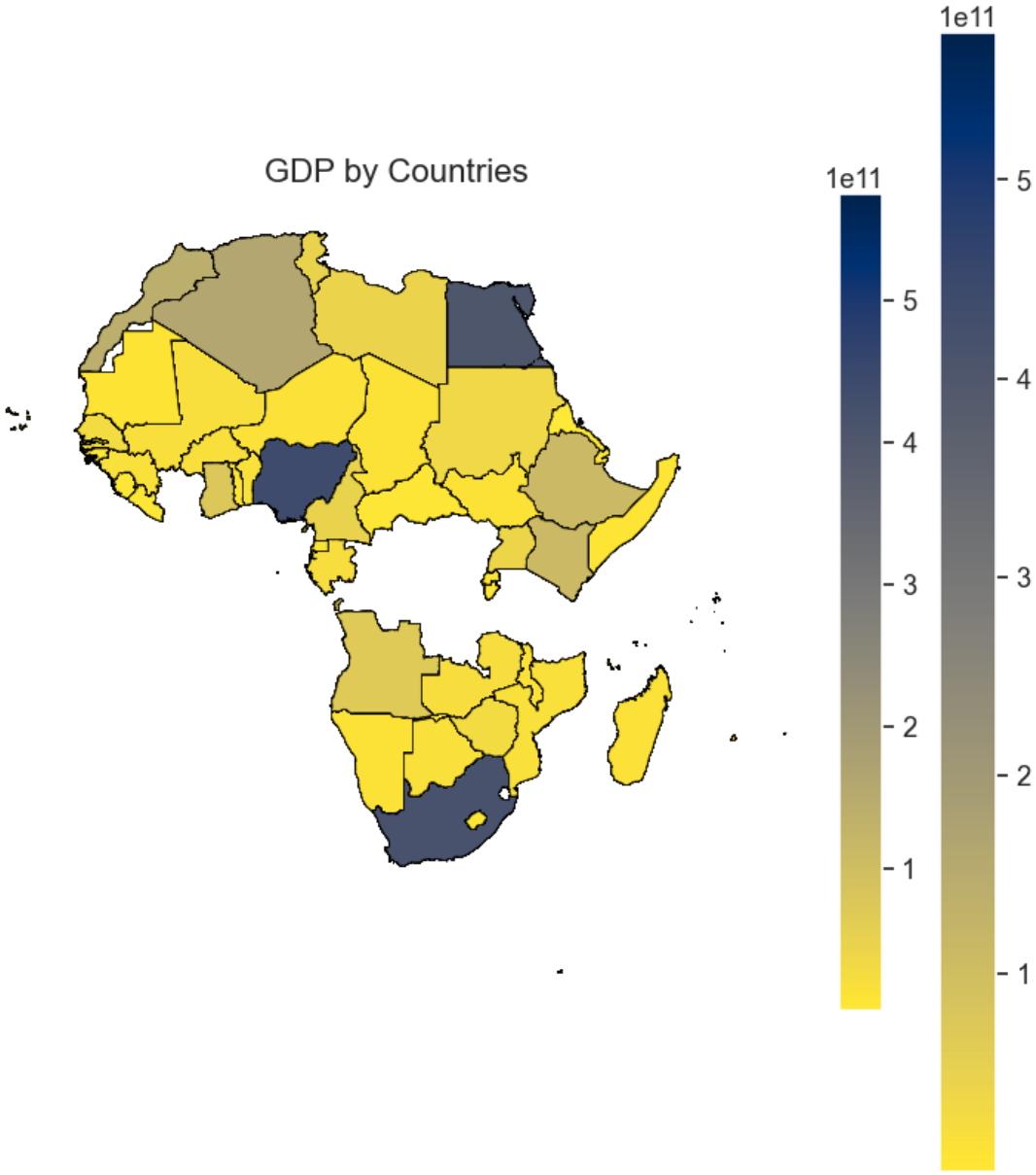
2.3 GDP exploration

[116]:
```
merged_df2.head()
```

[116]:
```
  Country Code  Year           GDP Country Name          IncomeGroup  \
0          AGO  1980  5.930503e+09       Angola  Lower middle income
1          AGO  1981  5.550483e+09       Angola  Lower middle income
2          AGO  1982  5.550483e+09       Angola  Lower middle income
3          AGO  1983  5.784342e+09       Angola  Lower middle income
4          AGO  1984  6.131475e+09       Angola  Lower middle income

          Region
0  Middle Africa
```

```
1  Middle Africa
2  Middle Africa
3  Middle Africa
4  Middle Africa
```

[115]:
```
# Load the shapefile of African countries
africa = gpd.read_file(r"C:
 ↪\Users\PC\Downloads\ne_10m_admin_0_countries\ne_10m_admin_0_countries.shp")
africa = africa.rename(columns={'SOVEREIGNT': 'Country Name'})


# Merge the shapefile with the data
merged_map = africa.merge(merged_df2, on='Country Name', how='left')

# Set the figure size and title
fig, ax = plt.subplots(figsize=(10,10))
ax.set_title('GDP by Countries', fontsize=16)

# Plot the map
cmap = 'cividis_r'
merged_map.plot(column='GDP', cmap=cmap, linewidth=0.5, edgecolor='black',␣
 ↪legend=True, ax=ax)

# Remove the axis
ax.axis('off')

# Set up the colorbar
vmin, vmax = merged_map['GDP'].min(), merged_map['GDP'].max()
norm = Normalize(vmin=vmin, vmax=vmax)
sm = plt.cm.ScalarMappable(cmap=cmap, norm=norm)
sm.set_array([])
divider = make_axes_locatable(ax)
cax = divider.append_axes('right', size='5%', pad=0.1)
cb = plt.colorbar(sm, cax=cax)


# Show the plot
plt.show()
```

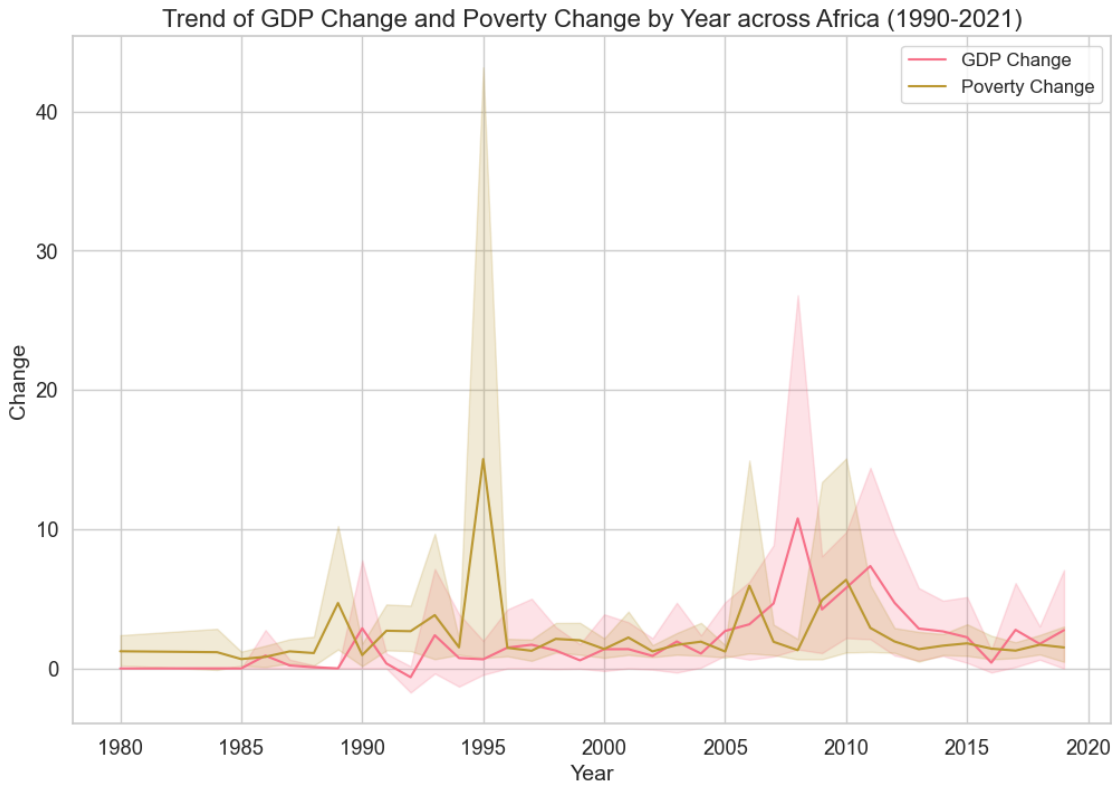


- Across the data provided, Nigeria, Egypt and South Africa have high GDPs compared to others and most of their neigbouring countries.

[119]:
```
#COMPUTE POVERTY RATE CHANGE AND GDP CHANGE OVER TIME

# select columns of interest
df1_select = merged_df1[['Country Name', 'Year', 'Number of People']]
df2_select = merged_df2[['Country Name', 'Year', 'GDP']]

# merge selected columns based on 'Country Code' and 'Year'
```

```
merged_df = pd.merge(df1_select, df2_select, on=['Country Name', 'Year'],␣
 ↪how='inner')

# calculate the change in GDP and poverty rate
merged_df['GDP Change'] = merged_df.groupby('Country Name')['GDP'].pct_change()
merged_df['Poverty Change'] = merged_df.groupby('Country Name')['Number of␣
 ↪People'].pct_change()

# create a new dataframe with the change in GDP and poverty rate
df_change = merged_df[['Country Name', 'Year', 'GDP Change', 'Poverty Change']].
 ↪dropna()
```

[121]:
```
# Scale GDP change to values between 0 to 1000
df_change['GDP_Change_Scaled'] = df_change['GDP Change'] * 50

# Create a line plot with trend lines
sns.set_style("whitegrid")
sns.set_palette("husl")
fig, ax = plt.subplots(figsize=(12,8))
sns.lineplot(x='Year', y='GDP_Change_Scaled', data=df_change, label='GDP␣
 ↪Change', ax=ax)
sns.lineplot(x='Year', y='Poverty Change', data=df_change, label='Poverty␣
 ↪Change', ax=ax)
ax.set_title('Trend of GDP Change and Poverty Change by Year across Africa␣
 ↪(1990-2021)', fontsize=16)
ax.set_xlabel('Year', fontsize=14)
ax.set_ylabel('Change', fontsize=14)
ax.legend(fontsize=12)
plt.show()
```



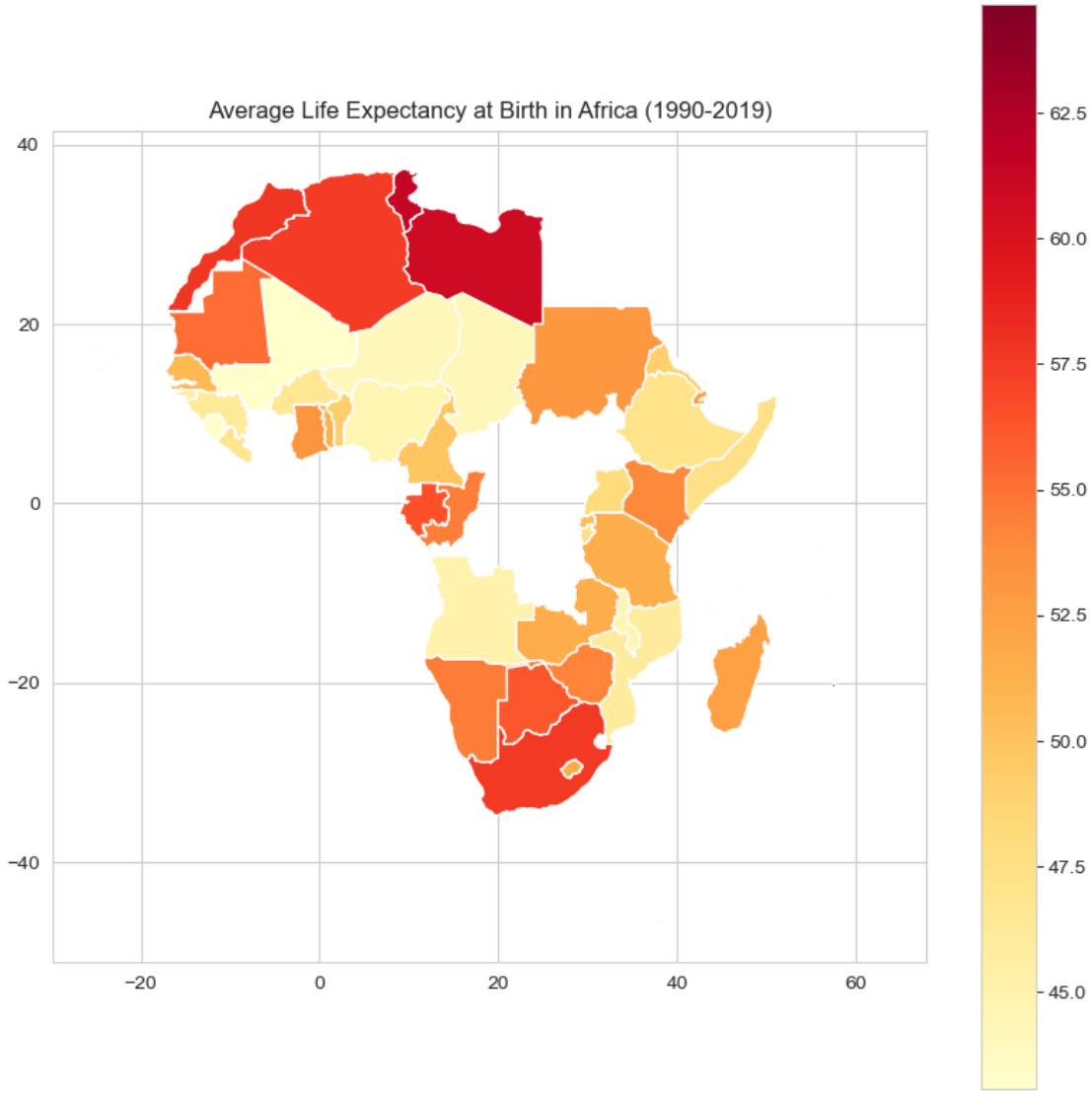
- Investigating GDP chaange alongside poverty change across the years, There is a seemingly a significant relationship between GDP and Poverty.

2.4 Life expectancy exploration

[169]:
```
merged_df3.head()
```

[169]:
```
  Country Code  Year  Life expectancy at birth (historical) Country Name  \
0          DZA  1923                                 28.82      Algeria
1          DZA  1933                                 31.22      Algeria
2          DZA  1943                                 33.72      Algeria
3          DZA  1950                                 42.40      Algeria
4          DZA  1951                                 42.50      Algeria

           IncomeGroup           Region
0  Lower middle income  Northern Africa
1  Lower middle income  Northern Africa
2  Lower middle income  Northern Africa
3  Lower middle income  Northern Africa
4  Lower middle income  Northern Africa
```

[201]:
```
import geopandas as gpd
import pandas as pd
import matplotlib.pyplot as plt

# Load shapefile with African countries polygons
africa = gpd.read_file(r"C:
 ↪\Users\PC\Downloads\ne_10m_admin_0_countries\ne_10m_admin_0_countries.shp")

# Filter data to include only African countries
african_countries = ['Algeria', 'Angola', 'Benin', 'Botswana', 'Burkina Faso',␣
 ↪'Burundi', 'Cabo Verde',
                     'Cameroon', 'Central African Republic', 'Chad', 'Comoros',␣
 ↪'Congo', "Côte d'Ivoire",
                     'Djibouti', 'DR Congo' 'Egypt', 'Equatorial Guinea',␣
 ↪'Eritrea', 'Eswatini', 'Ethiopia',
                     'Gabon', 'Gambia, The', 'Ghana', 'Guinea',␣
 ↪'Guinea-Bissau', 'Kenya', 'Lesotho', 'Liberia',
                     'Libya', 'Madagascar', 'Malawi', 'Mali', 'Mauritania',␣
 ↪'Mauritius', 'Morocco', 'Mozambique',
                     'Namibia', 'Niger', 'Nigeria', 'Rwanda', 'Sao Tome and␣
 ↪Principe', 'Senegal', 'Seychelles',
                     'Sierra Leone', 'Somalia', 'South Africa', 'South Sudan',␣
 ↪'Sudan', 'Tanzania', 'Togo',
                     'Tunisia', 'Uganda', 'Zambia', 'Zimbabwe']
merged_df3 = merged_df3[merged_df3['Country Name'].isin(african_countries)]

# Compute average life expectancy by country
life_expectancy_avg = merged_df3.groupby(['Country Name'])['Life expectancy at␣
 ↪birth (historical)'].mean().reset_index()

# Merge life expectancy data with African shapefile
africa_life_expectancy = pd.merge(africa, life_expectancy_avg, left_on='NAME',␣
 ↪right_on='Country Name')

# Plot map
fig, ax = plt.subplots(figsize=(10, 10))
africa_life_expectancy.plot(column='Life expectancy at birth (historical)',␣
 ↪cmap='YlOrRd', legend=True, ax=ax)
ax.set_title('Average Life Expectancy at Birth in Africa (1990-2019)')
plt.show()
```



- The heat map above shows a relatively high average life expectancy in both Northern Africa and Southern Africa. Libya being the highest

[172]:
```
print(africa.columns)
```

```
Index(['ADMIN', 'ISO_A3', 'ISO_A2', 'geometry'], dtype='object')
```

[180]:
```
merged_df5 = pd.merge(merged_df1, merged_df3, on='Country Name')
merged_df5.head()
```

[180]:

	Country Code_x	Year_x	Poverty Line	Number of People	Country Name
0	DZA	1988	$40	156887	Algeria
1	DZA	1988	$40	156887	Algeria
2	DZA	1988	$40	156887	Algeria

```
3            DZA    1988          $40            156887      Algeria
4            DZA    1988          $40            156887      Algeria

         IncomeGroup_x         Region_x  Country_Num Country Code_y  Year_y  \
0  Lower middle income  Northern Africa            0            DZA    1923
1  Lower middle income  Northern Africa            0            DZA    1933
2  Lower middle income  Northern Africa            0            DZA    1943
3  Lower middle income  Northern Africa            0            DZA    1950
4  Lower middle income  Northern Africa            0            DZA    1951

   Life expectancy at birth (historical)        IncomeGroup_y         Region_y
0                                  28.82  Lower middle income  Northern Africa
1                                  31.22  Lower middle income  Northern Africa
2                                  33.72  Lower middle income  Northern Africa
3                                  42.40  Lower middle income  Northern Africa
4                                  42.50  Lower middle income  Northern Africa
```

[183]:
```
df_change = df_change.merge(merged_df3[['Country Name', 'Year', 'Life␣
 ↪expectancy at birth (historical)']], on=['Country Name', 'Year'])
```

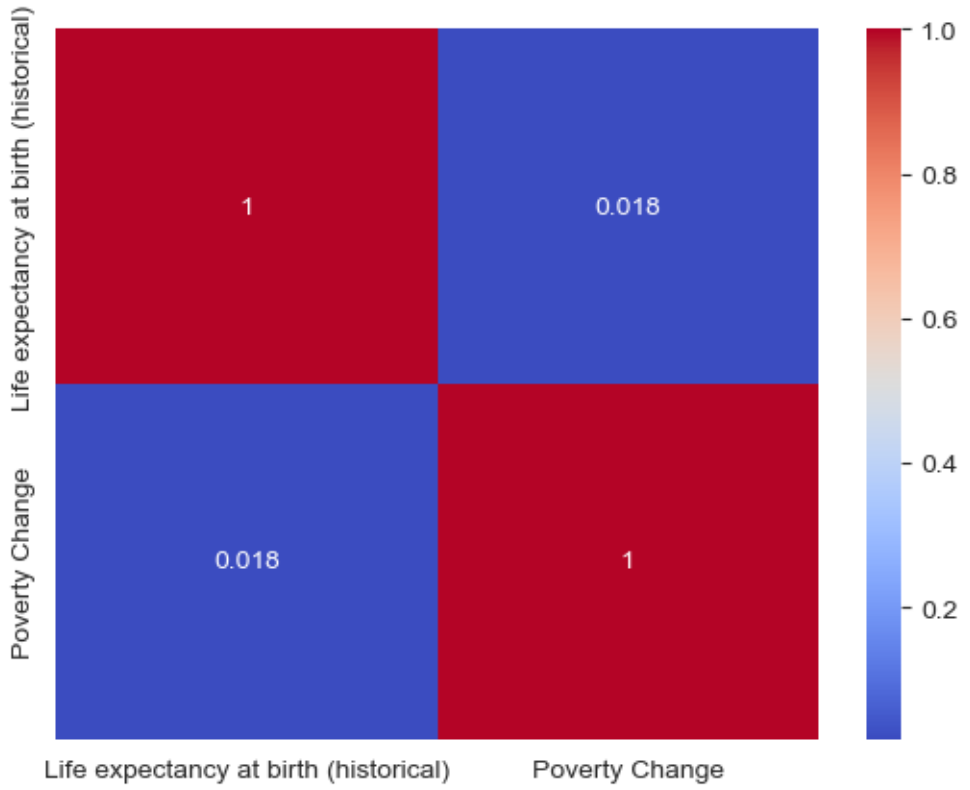
[184]:
```
df_change.head()
```

[184]:
```
  Country Name  Year  GDP Change  Poverty Change  GDP_Change_Scaled  \
0      Algeria  1988         0.0       -0.003786                0.0
1      Algeria  1988         0.0        2.232064                0.0
2      Algeria  1988         0.0        5.624816                0.0
3      Algeria  1988         0.0        0.410732                0.0
4      Algeria  1988         0.0        0.990966                0.0

   Scaled_GDP_Change  Life expectancy at birth (historical)
0                0.0                                  67.0
1                0.0                                  67.0
2                0.0                                  67.0
3                0.0                                  67.0
4                0.0                                  67.0
```

[186]:
```
import seaborn as sns

# select relevant columns from df_change
df_corr = df_change[['Life expectancy at birth (historical)', 'Poverty Change']]

# filter data for years between 1990 and 2021
df_corr = df_corr[df_change['Year'].between(1990, 2021)]

# compute correlation matrix
corr_matrix = df_corr.corr()
```

```
# plot heatmap
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
```

[186]: <AxesSubplot:>



- This heatmap shows a weak correlation between Life expetancy and poverty change over time.

3 How are higher Income regions successful?

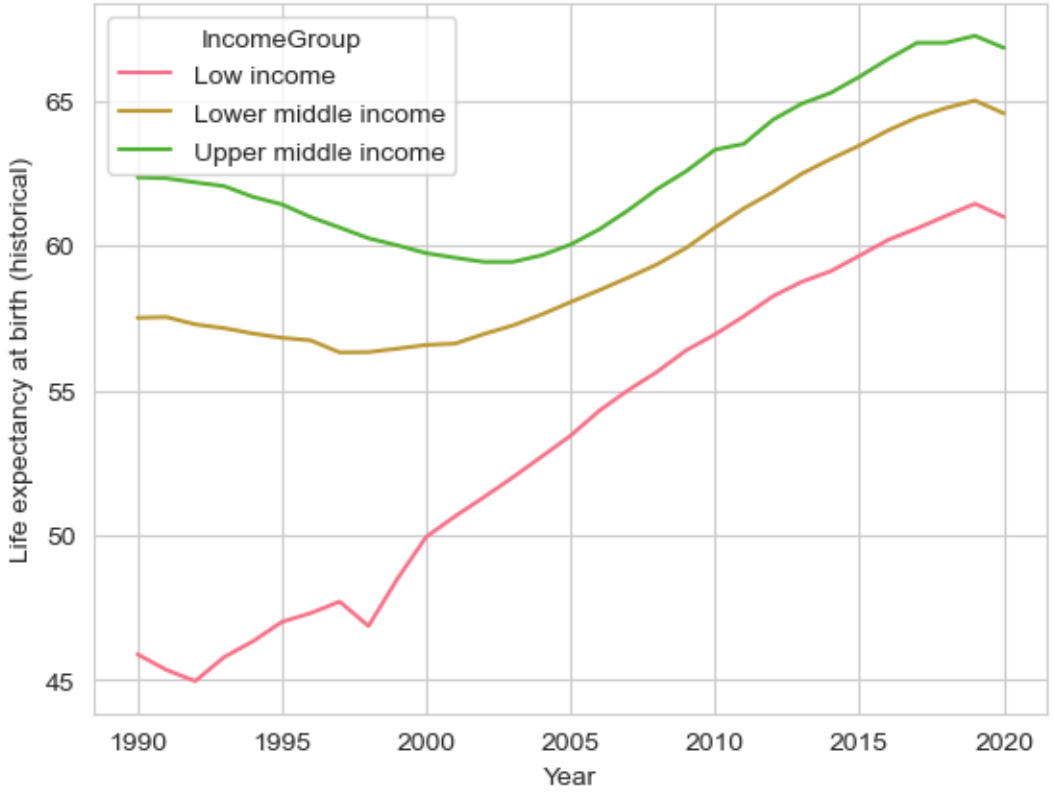
[187]:
```
merged_df3.head()
```

[187]:
```
  Country Code  Year  Life expectancy at birth (historical) Country Name  \
0          DZA  1923                                 28.82      Algeria
1          DZA  1933                                 31.22      Algeria
2          DZA  1943                                 33.72      Algeria
3          DZA  1950                                 42.40      Algeria
4          DZA  1951                                 42.50      Algeria

           IncomeGroup           Region
0  Lower middle income  Northern Africa
1  Lower middle income  Northern Africa
```

```
2  Lower middle income  Northern Africa
3  Lower middle income  Northern Africa
4  Lower middle income  Northern Africa
```

[191]:
```
import seaborn as sns

# filter data for selected income groups and years
selected_years = range(1990, 2021)
selected_groups = ['Low income', 'Lower middle income', 'Upper middle income']
filtered_df = merged_df3.loc[(merged_df3['Year'].isin(selected_years)) &␣
 ↪(merged_df3['IncomeGroup'].isin(selected_groups))]

# group by year and income group and calculate the mean life expectancy
grouped_df = filtered_df.groupby(['Year', 'IncomeGroup'])['Life expectancy at␣
 ↪birth (historical)'].mean().reset_index()

# plot the trend line using seaborn
sns.lineplot(data=grouped_df, x='Year', y='Life expectancy at birth␣
 ↪(historical)', hue='IncomeGroup')
```

[191]: <AxesSubplot:xlabel='Year', ylabel='Life expectancy at birth (historical)'>

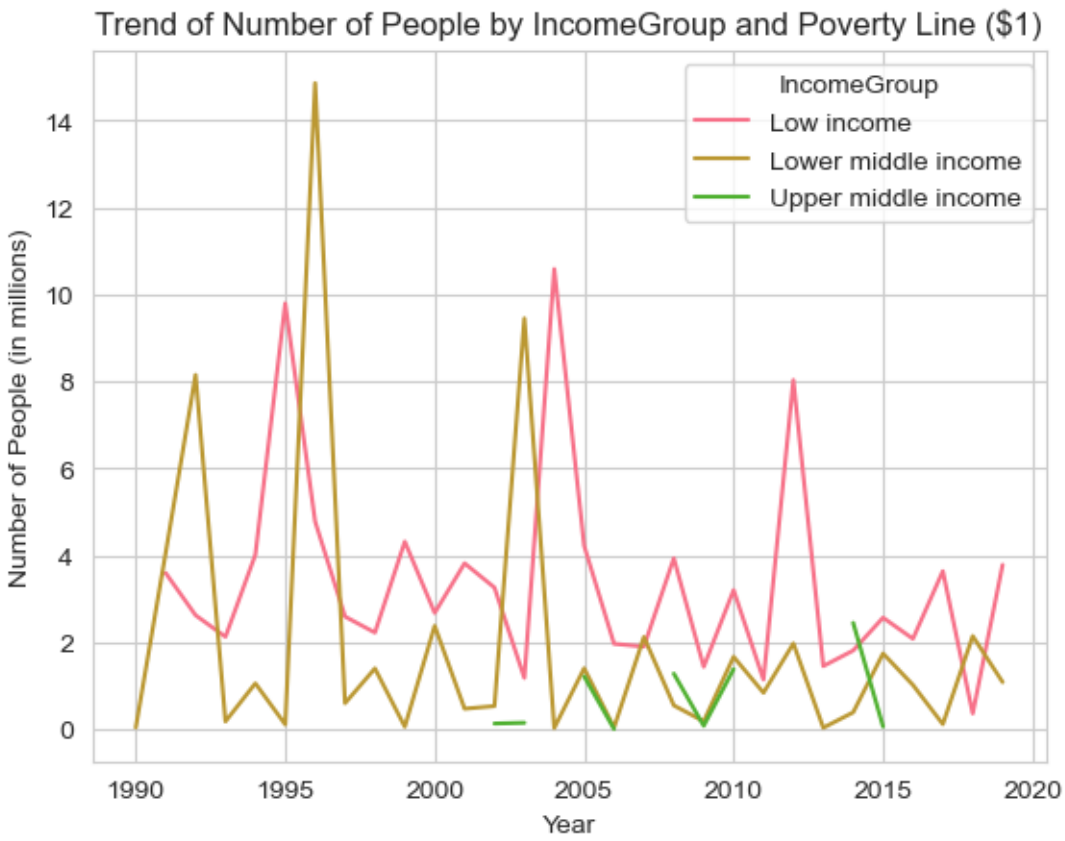


- before the year 2000, lower income group experienced an increase in life expectancy after which the group follow similar trend to the Lower middle and Upper middle. The Higher income group was excluded due to Outlier detection. seychelles.

[192]:
```
merged_df1.head()
```

[192]:
```
  Country Code  Year Poverty Line  Number of People Country Name  \
0          DZA  1988          $40            156887      Algeria
1          DZA  1988      $30-$40            156293      Algeria
2          DZA  1988      $20-$30            505149      Algeria
3          DZA  1988      $10-$20           3346519      Algeria
4          DZA  1988    $6.85-$10           4721041      Algeria

            IncomeGroup           Region  Country_Num
0  Lower middle income  Northern Africa            0
1  Lower middle income  Northern Africa            0
2  Lower middle income  Northern Africa            0
3  Lower middle income  Northern Africa            0
4  Lower middle income  Northern Africa            0
```

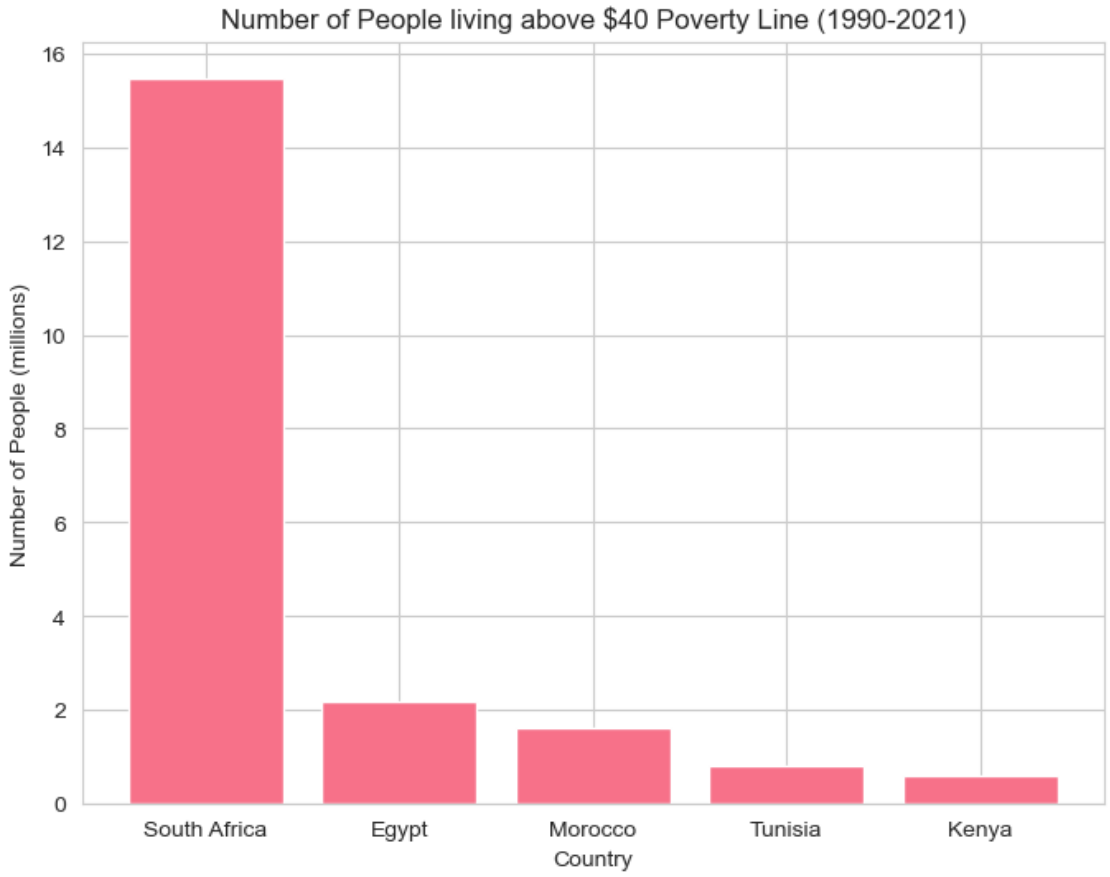
[198]:
```
# Filter the data to only include rows where Poverty Line is $1 and Year is␣
 ↪greater than or equal to 1990
df_filtered = merged_df1[(merged_df1['Poverty Line'] == '$1') &␣
 ↪(merged_df1['Year'] >= 1990)]

# Group the data by IncomeGroup and Year to calculate the average Number of␣
 ↪People in millions
df_grouped = df_filtered.groupby(['IncomeGroup', 'Year']).agg({'Number of␣
 ↪People': 'mean'}).reset_index()
df_grouped['Number of People'] = df_grouped['Number of People'] / 1000000

# Filter the data to only include rows where IncomeGroup is Low income, Lower␣
 ↪middle income, or Upper middle income
df_filtered2 = df_grouped[df_grouped['IncomeGroup'].isin(['Low income', 'Lower␣
 ↪middle income', 'Upper middle income'])]

# Pivot the data to have IncomeGroup as columns and Year as index
df_pivot = df_filtered2.pivot(index='Year', columns='IncomeGroup',␣
 ↪values='Number of People')

# Plot the line chart
df_pivot.plot(kind='line')
plt.xlabel('Year')
plt.ylabel('Number of People (in millions)')
plt.title('Trend of Number of People by IncomeGroup and Poverty Line ($1)')
plt.show()
```



- Looking at the trend of population below the poverty line by income group, we notice that the lower middle income group had a significant spike in population below poverty line in the late 90s.

[199]:
```
import matplotlib.pyplot as plt

# Filter for poverty line = $40 and years 1990 to 2021
df_filtered = merged_df1[(merged_df1['Poverty Line'] == '$40') &␣
 ↪(merged_df1['Year'] >= 1990) & (merged_df1['Year'] <= 2021)]

# Group by country and sum the Number of People
grouped_df = df_filtered.groupby(['Country Name']).sum().sort_values('Number of␣
 ↪People', ascending=False).head(5)

# Convert Number of People to millions
grouped_df['Number of People (millions)'] = grouped_df['Number of People'] /␣
 ↪1000000

# Plot bar chart
plt.figure(figsize=(8, 6))
```

```
plt.bar(grouped_df.index, grouped_df['Number of People (millions)'])
plt.title('Number of People living above $40 Poverty Line (1990-2021)')
plt.xlabel('Country')
plt.ylabel('Number of People (millions)')
plt.show()
```



- South Africa has by far the highest Number of people above poverty Line

3.0.1 Deeper Dive Into South AFrica & Recommendations

[202]:
```
# Filter the data to only include rows where Poverty Line is $40, Country Name␣
 ↪is South Africa, and Year is between 1990 and 2021
df_filtered = merged_df1[(merged_df1['Poverty Line'] == '$40') &␣
 ↪(merged_df1['Country Name'] == 'South Africa') & (merged_df1['Year'] >=␣
 ↪1990)]

# Convert Number of People to millions
df_filtered['Number of People'] = df_filtered['Number of People'] / 1000000
```

```
# Group the data by IncomeGroup and Year to calculate the average Number of␣
 ↪People
df_grouped = df_filtered.groupby(['IncomeGroup', 'Year']).agg({'Number of␣
 ↪People': 'mean'}).reset_index()

# Filter the data to only include rows where IncomeGroup is Low income, Lower␣
 ↪middle income, or Upper middle income
df_filtered2 = df_grouped[df_grouped['IncomeGroup'].isin(['Low income', 'Lower␣
 ↪middle income', 'Upper middle income'])]

# Pivot the data to have IncomeGroup as columns and Year as index
df_pivot = df_filtered2.pivot(index='Year', columns='IncomeGroup',␣
 ↪values='Number of People')

# Plot the trend line chart
df_pivot.plot(kind='line')
plt.xlabel('Year')
plt.ylabel('Number of People (in millions)')
plt.title('Trend of Number of People by IncomeGroup and Poverty Line ($40) in␣
 ↪South Africa')
plt.show()
```

According to https://www.numbeo.com/cost-of-living/compare_cities.jsp?country1=South+Africa&country2=Ni

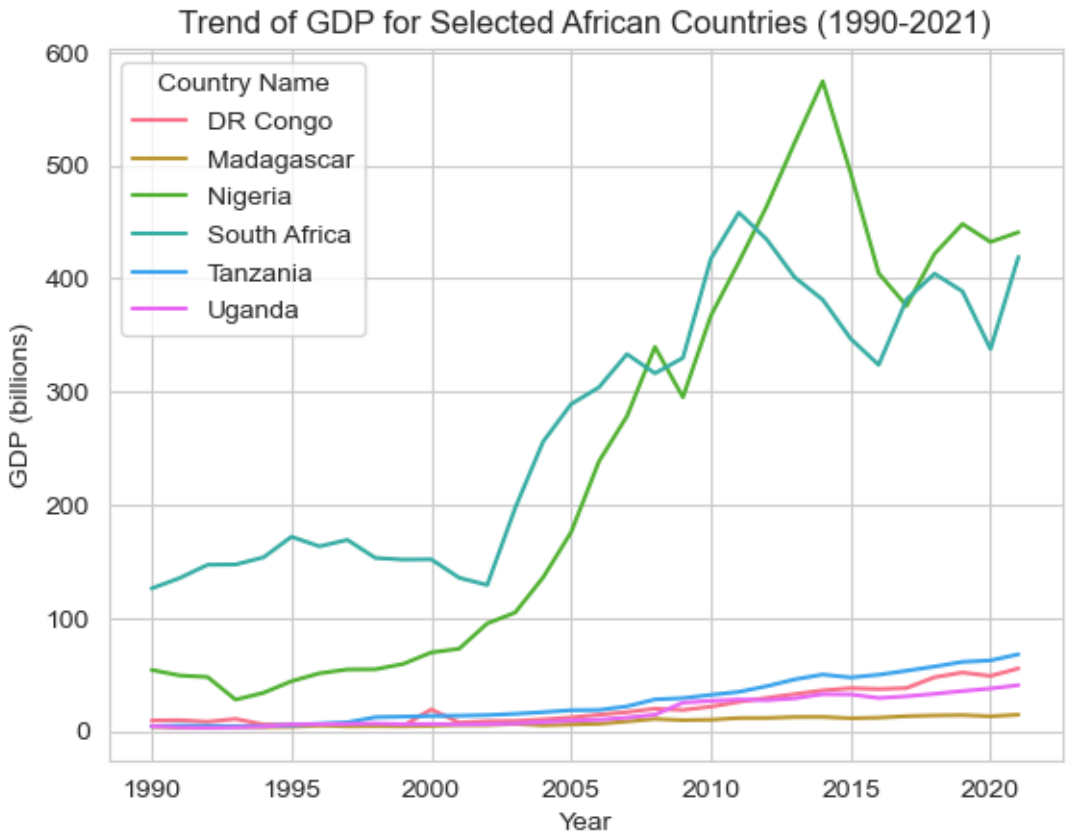
- You would need around 53,555.7R (1,282,410.1₦) in Lagos to maintain the same standard of life that you can have with 45,000.0R in Cape Town (assuming you rent in both cities). This calculation uses our Cost of Living Plus Rent Index to compare cost of living. This assumes net earnings (after income tax). You can change the amount in this calculation.

Indices Difference Info - Consumer Prices in Lagos are 16.2% higher than in Cape Town (without rent) - Consumer Prices Including Rent in Lagos are 19.0% higher than in Cape Town - Rent Prices in Lagos are 24.6% higher than in Cape Town - Restaurant Prices in Lagos are 3.9% higher than in Cape Town - Groceries Prices in Lagos are 43.7% higher than in Cape Town - Local Purchasing Power in Lagos is 85.8% lower than in Cape Town

Recommendation - From the trend line above South Africa benefits from the low cost of living - Countries like Nigeria may want to look into strict price regulation and economic diversification.

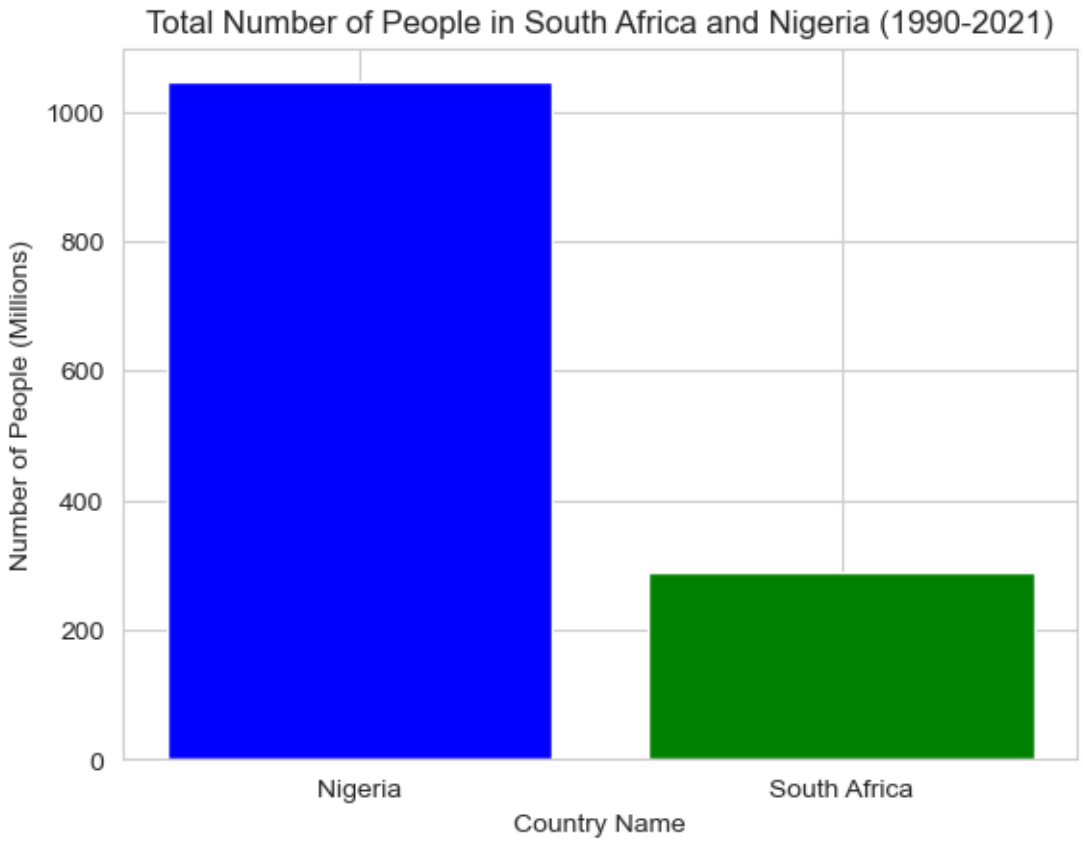
3.0.2 GDP comparison between South Africa and the low performing Countries

[203]:
```
# Filter the data to only include rows for the given countries and years␣
 ↪between 1990 and 2021
df_filtered = merged_df2[(merged_df2['Country Name'].isin(['South Africa',␣
 ↪'Nigeria', 'DR Congo', 'Madagascar', 'Uganda', 'Tanzania'])) &␣
 ↪(merged_df2['Year'] >= 1990) & (merged_df2['Year'] <= 2021)]

# Pivot the data to have Country Name as columns and Year as index
df_pivot = df_filtered.pivot(index='Year', columns='Country Name', values='GDP')

# Convert GDP to billions
df_pivot = df_pivot / 1000000000

# Plot the line chart
df_pivot.plot(kind='line')
plt.xlabel('Year')
plt.ylabel('GDP (billions)')
plt.title('Trend of GDP for Selected African Countries (1990-2021)')
plt.show()
```



- Investigating the GDP trend of South AFrica and the Top 5 countries with the highest number of population living below poverty line, The GDP trends of Nigeria and South AFrica follow similar path. prompting an investigation into their total population.

[207]:
```
# Filter the data to only include rows where Country Name is South Africa or␣
 ↪Nigeria and Year is between 1990 and 2021
df_filtered = merged_df1[(merged_df1['Country Name'].isin(['South Africa',␣
 ↪'Nigeria'])) & (merged_df1['Year'] >= 1990)]

# Group the data by Country Name to calculate the total Number of People
df_grouped = df_filtered.groupby(['Country Name']).agg({'Number of People':␣
 ↪'sum'}).reset_index()

# Plot the bar chart
plt.bar(df_grouped['Country Name'], df_grouped['Number of People']/1000000,␣
 ↪color=['blue', 'green'])
plt.xlabel('Country Name')
plt.ylabel('Number of People (Millions)')
plt.title('Total Number of People in South Africa and Nigeria (1990-2021)')
plt.show()
```



- The population difference between South Africa and Nigeria is highly significant during the last three decades.

Recommendation: - Countries Like Nigeria may want to promote family planning as sources show that south africa practice it a lot(Department of health, south africa. WHO and United Nations population fund)

3.0.3 TO PREDICT THE GDP FOR BOTH COUNTRIES USING THEIR POPULATION DATA

[209]:
```
# Merge the two dataframes on 'Country Name' and 'Year' columns
merge_south_naija = pd.merge(merged_df2, merged_df1, on=['Country Name',␣
 ↪'Year'])

# Select only the rows for 'South Africa' and 'Nigeria'
merge_south_naija = merge_south_naija[(merge_south_naija['Country Name'] ==␣
 ↪'South Africa') | (merge_south_naija['Country Name'] == 'Nigeria')]

# Select only the rows for years 1990 to 2019
merge_south_naija = merge_south_naija[(merge_south_naija['Year'] >= 1990) &␣
 ↪(merge_south_naija['Year'] <= 2019)]
```

[211]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

# Filter the data to only include rows where Country Name is South Africa or␣
 ↪Nigeria
df_filtered = merge_south_naija[(merge_south_naija['Country Name'].isin(['South␣
 ↪Africa', 'Nigeria'])) & (merge_south_naija['Year'] >= 1990)]

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(df_filtered['Number of␣
 ↪People'], df_filtered['GDP'], test_size=0.2, random_state=42)

# Train a linear regression model on the training set
model = LinearRegression()
model.fit(X_train.values.reshape(-1, 1), y_train)

# Predict GDP for the test set
y_pred = model.predict(X_test.values.reshape(-1, 1))

# Evaluate the model using R-squared
r2 = model.score(X_test.values.reshape(-1, 1), y_test)
print('R-squared:', r2)
```

```
R-squared: 0.03909623366683257
```

[122]:
```
# Merge merged_df1 and merged_df2
merged_df12 = pd.merge(merged_df1, merged_df2, on='Country Name', how='inner')

# Merge merged_df12 and merged_df3
merged_df123 = pd.merge(merged_df12, merged_df3, on='Country Name', how='inner')
```

[ ]:
```

```

- An R-squared value of 0.039 means that only 3.9% of the variability in the GDP can be explained by the variability in the Number of People. This indicates a weak or low correlation between these two variables. Therefore, it may not be reliable to use the Number of People alone to predict the GDP of South Africa and Nigeria.

[126]:
```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score


# Filter the data for Poverty Line of $1
df = merged_df123[merged_df123['Poverty Line']=='$1']

# Define the features and target variables
X = df[['GDP', 'Life expectancy at birth (historical)']]
```

```
y = df['Number of People']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

# Initialize and fit the model
model = LinearRegression()
model.fit(X_train, y_train)

# Predict the target variable using the test data
y_pred = model.predict(X_test)

# Evaluate the model using R^2 score
r2 = r2_score(y_test, y_pred)

print('R^2 score:', r2)
```

```
R^2 score: 0.05422107089158257
```

- The r squared shows that GDP and Life Expectancy as far as the data goes, only explain 5.4% of the population living under the poverty Line.

[130]:
```
from sklearn.linear_model import LinearRegression
import numpy as np

# Create a new DataFrame with the relevant columns
df = merged_df123[['Number of People', 'GDP', 'Life expectancy at birth␣
 ↪(historical)', 'IncomeGroup', 'Poverty Line']]

# Filter the data to only include rows where Poverty Line is $1
df = df[df['Poverty Line'] == '$1']
# Filter the data to only include rows where Poverty Line is $1
df = df[df['Poverty Line'] == '$1']

# Convert the Poverty Line column to float
df['Poverty Line'] = df['Poverty Line'].str.replace('$', '').astype(float)

# Convert IncomeGroup to a one-hot encoded categorical variable
df = pd.concat([df, pd.get_dummies(df['IncomeGroup'])], axis=1)
df = df.drop(columns=['IncomeGroup'])

# Split the data into training and test sets
train_set = df.sample(frac=0.8, random_state=1)
test_set = df.drop(train_set.index)

# Train the model on the training set
```

```
model = LinearRegression().fit(train_set.drop(columns=['Number of People']),␣
 ↪train_set['Number of People'])

# Evaluate the model on the test set
y_true = test_set['Number of People']
y_pred = model.predict(test_set.drop(columns=['Number of People']))
r_squared = model.score(test_set.drop(columns=['Number of People']),␣
 ↪test_set['Number of People'])

print(f"R^2: {r_squared}")
```

```
R^2: 0.09497072524440431
```

- Adding IncomeGroup increased the model performance to around 10%.

3.1 Conclusion

- Countries Like Nigeria may want to promote family planning as sources including the data show that south africa has steadily maintained population growth over the decades while growing their GDP(Department of health, south africa. WHO and United Nations population fund)
- More data cutting across Africa is needed to help predict with higher accuracy how the popualtion trend of peope living under the poverty line will behave considering other factors in order to profer more impactful recommendations to mitigate poverty.
- The incomplete data also affected the result of this analysis as some countries had data truncated leaving on a few years for our analysis. Complete data of countries like DR Congo especially will have provided us with relevant information concerning poverty rate decline as far as we saw in the data.

Thank you for your time.

I hope you found this analysis Insightful

[ ]: