

# World Development Indicators Analysis

Caleb Sharkey

2025-02-25

```
#!/pip install wbgapi

# Import necessary libraries
import pandas as pd
import wbgapi as wb

# Define the indicators to download
indicators = {
    'gdp_per_capita': 'NY.GDP.PCAP.CD',
    'gdp_growth_rate': 'NY.GDP.MKTP.KD.ZG',
    'inflation_rate': 'FP.CPI.TOTL.ZG',
    'unemployment_rate': 'SL.UEM.TOTL.ZS',
    'total_population': 'SP.POP.TOTL',
    'life_expectancy': 'SP.DYN.LE00.IN',
    'adult_literacy_rate': 'SE.ADT.LITR.ZS',
    'income_inequality': 'SI.POV.GINI',
    'health_expenditure_gdp_share': 'SH.XPD.CHEX.GD.ZS',
    'measles_immunisation_rate': 'SH.IMM.MEAS',
    'education_expenditure_gdp_share': 'SE.XPD.TOTL.GD.ZS',
    'primary_school_enrolment_rate': 'SE.PRM.ENRR',
    'exports_gdp_share': 'NE.EXP.GNFS.ZS'
}

# Get country codes for the "World" region
country_codes = wb.region.members('WLD')

# Download data for 2022
df = wb.data.DataFrame(indicators.values(), economy=country_codes, time=2022, skipBlanks=True)

# Remove 'economy' column if it exists
df = df.drop(columns=['economy'], errors='ignore')
```

```

# Rename columns to indicator names
df.rename(columns=lambda x: {v: k for k, v in indicators.items()}.get(x, x).lower(), inplace=True)

# Sort by country and reset index
df = df.sort_values('country').reset_index(drop=True)

# Save to CSV
df.to_csv("wdi.csv", index=False)

# Display dataset info
print(df.info())
print(df.head())

```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 217 entries, 0 to 216
```

```
Data columns (total 14 columns):
```

#	Column	Non-Null Count	Dtype
0	country	217 non-null	object
1	inflation_rate	173 non-null	float64
2	exports_gdp_share	179 non-null	float64
3	gdp_growth_rate	206 non-null	float64
4	gdp_per_capita	207 non-null	float64
5	adult_literacy_rate	54 non-null	float64
6	primary_school_enrolment_rate	156 non-null	float64
7	education_expenditure_gdp_share	137 non-null	float64
8	measles_immunisation_rate	193 non-null	float64
9	health_expenditure_gdp_share	20 non-null	float64
10	income_inequality	28 non-null	float64
11	unemployment_rate	186 non-null	float64
12	life_expectancy	209 non-null	float64
13	total_population	217 non-null	float64

```
dtypes: float64(13), object(1)
```

```
memory usage: 23.9+ KB
```

```
None
```

	country	inflation_rate	exports_gdp_share	gdp_growth_rate	\
0	Afghanistan	NaN	18.380042	-6.240172	
1	Albania	6.725203	37.197085	4.826688	
2	Algeria	9.265516	30.808979	3.600000	
3	American Samoa	NaN	46.957520	1.735016	
4	Andorra	NaN	NaN	9.564612	

	gdp_per_capita	adult_literacy_rate	primary_school_enrolment_rate	\
0	357.261153	NaN	NaN	
1	6846.426143	98.5	96.371231	
2	4961.552577	NaN	108.343933	
3	18017.458938	NaN	NaN	
4	42414.059009	NaN	90.147346	

	education_expenditure_gdp_share	measles_immunisation_rate	\
0	NaN	56.0	
1	2.744330	86.0	
2	4.749247	79.0	
3	NaN	NaN	
4	2.647290	98.0	

	health_expenditure_gdp_share	income_inequality	unemployment_rate	\
0	NaN	NaN	14.100	
1	NaN	NaN	10.137	
2	NaN	NaN	12.346	
3	NaN	NaN	NaN	
4	NaN	NaN	NaN	

	life_expectancy	total_population
0	62.879	40578842.0
1	76.833	2777689.0
2	77.129	45477389.0
3	NaN	48342.0
4	NaN	79705.0

## Exploratory Data Analysis (EDA)

Below, I perform EDA on three selected indicators from the dataset: **GDP per Capita**, **Life Expectancy**, and **Unemployment Rate**.

```
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
df = pd.read_csv("wdi.csv")
```

```

# Select three indicators for analysis
selected_columns = ["country", "gdp_per_capita", "life_expectancy", "unemployment_rate"]
df_subset = df[selected_columns]

# Display summary statistics
df_summary = df_subset.describe()
print(df_summary)

# Check for missing values
missing_values = df_subset.isnull().sum()
print("Missing Values:\n", missing_values)

# Plot distributions
fig, axes = plt.subplots(1, 3, figsize=(18, 5))

# GDP per capita distribution
sns.histplot(df_subset["gdp_per_capita"].dropna(), bins=30, ax=axes[0], kde=True)
axes[0].set_title("GDP per Capita Distribution")
axes[0].set_xlabel("GDP per Capita")

# Life Expectancy distribution
sns.histplot(df_subset["life_expectancy"].dropna(), bins=30, ax=axes[1], kde=True)
axes[1].set_title("Life Expectancy Distribution")
axes[1].set_xlabel("Life Expectancy (Years)")

# Unemployment Rate distribution
sns.histplot(df_subset["unemployment_rate"].dropna(), bins=30, ax=axes[2], kde=True)
axes[2].set_title("Unemployment Rate Distribution")
axes[2].set_xlabel("Unemployment Rate (%)")

plt.tight_layout()
plt.show()

```

	gdp_per_capita	life_expectancy	unemployment_rate
count	207.000000	209.000000	186.000000
mean	20520.336828	72.416519	7.227344
std	30640.741594	7.713322	5.844462
min	250.634225	52.997000	0.130000
25%	2599.752468	66.782000	3.478000
50%	7606.237525	73.514634	5.334000
75%	27542.145523	78.475000	9.261750
max	226052.001905	85.377000	35.359000

Missing Values:

```
country          0
gdp_per_capita    10
life_expectancy    8
unemployment_rate 31
dtype: int64
```

/opt/anaconda3/lib/python3.11/site-packages/seaborn/\_oldcore.py:1119: FutureWarning:

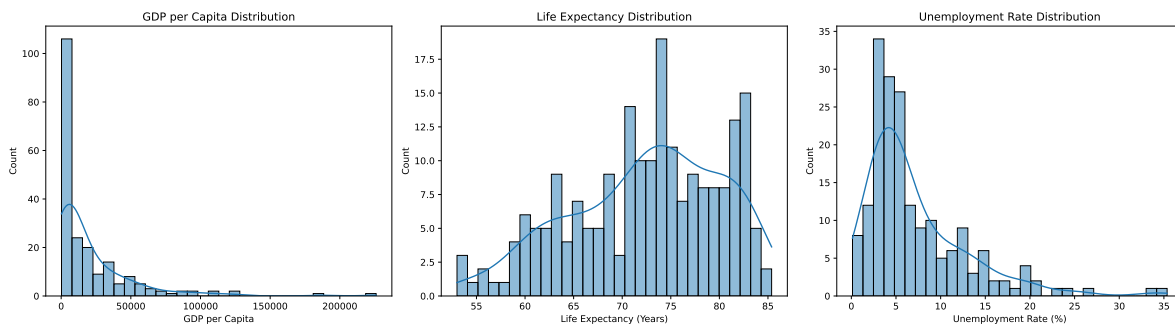
use\_inf\_as\_na option is deprecated and will be removed in a future version. Convert inf value

/opt/anaconda3/lib/python3.11/site-packages/seaborn/\_oldcore.py:1119: FutureWarning:

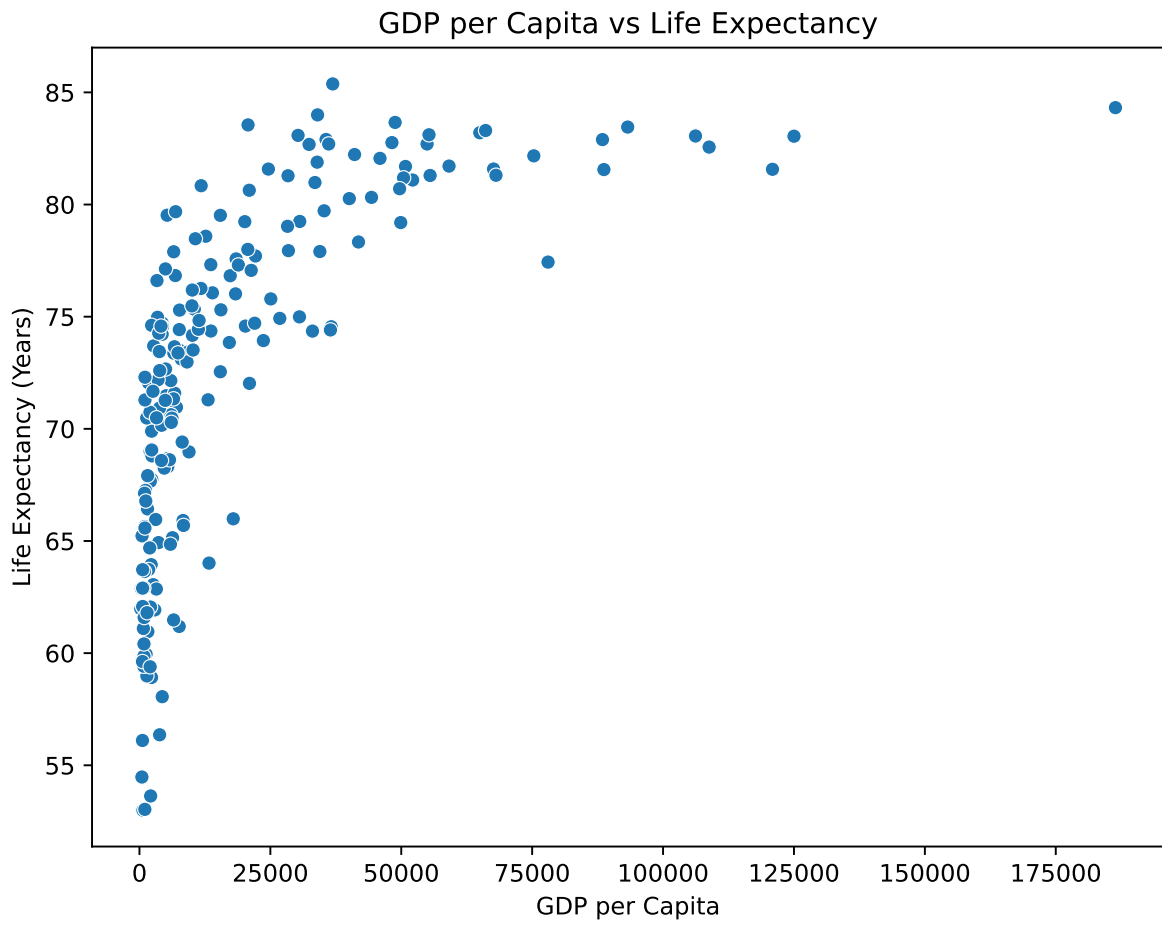
use\_inf\_as\_na option is deprecated and will be removed in a future version. Convert inf value

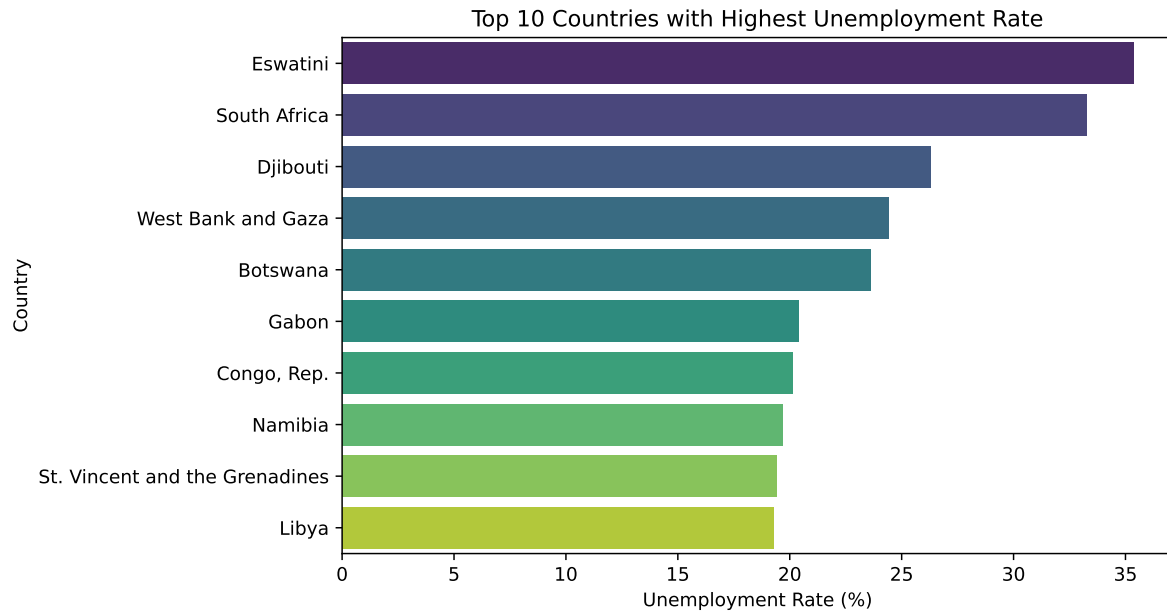
/opt/anaconda3/lib/python3.11/site-packages/seaborn/\_oldcore.py:1119: FutureWarning:

use\_inf\_as\_na option is deprecated and will be removed in a future version. Convert inf value



After running an EDA on this World Bank Indicator data, we have found that the median GDP per capita of countries represented here is just about \$7500. The distribution of per capita wealth by country is incredibly skewed with most countries having a GDP less than \$7000. We also found the median life expectancy is around 73 years old and the median unemployment rate is 5.3%.





	mean	median	min	max
gdp_per_capita	20520.336828	7606.237525	250.634225	226052.001905
life_expectancy	72.416519	73.514634	52.997000	85.377000
unemployment_rate	7.227344	5.334000	0.130000	35.359000