World Development Indicators Analysis (2022)

Joyce Chen

2025-02-26

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see https://quarto.org.

```
# Install the necessary libraries
# pip install pandas
# !pip install wbgapi

# Import the libraries
import pandas as pd
import wbgapi as wb
```

Task 2

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

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

# Delete the 'economy' column
df = df.drop(columns=['economy'], errors='ignore')

# Create a reversed dictionary mapping indicator codes to names
# Rename the columns and convert all names to lowercase
df.rename(columns=lambda x: {v: k for k, v in indicators.items()}.get(x, x).lower(), inplace=

# Sort 'country' in ascending order
df = df.sort_values('country', ascending=True)

# Reset the index after sorting
df = df.reset_index(drop=True)

# Display the number of rows and columns
print(df.shape)

# Display the first few rows of the data
print(df.head(3))

# Save the data to a CSV file
df.to_csv('wdi.csv', index=False)
```

```
(217, 14)
       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

   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
```

```
   education_expenditure_gdp_share  measles_immunisation_rate  \
0                              NaN                       56.0
1                         2.744330                       86.0
2                         4.749247                       79.0

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

   life_expectancy  total_population
0           62.879        40578842.0
1           76.833         2777689.0
2           77.129        45477389.0
```

Task 3

```
# Select relevant indicators
selected_indicators = ["gdp_per_capita", "inflation_rate", "unemployment_rate"]
df[selected_indicators].describe()

# Check for missing values
df[selected_indicators].isnull().sum()
```

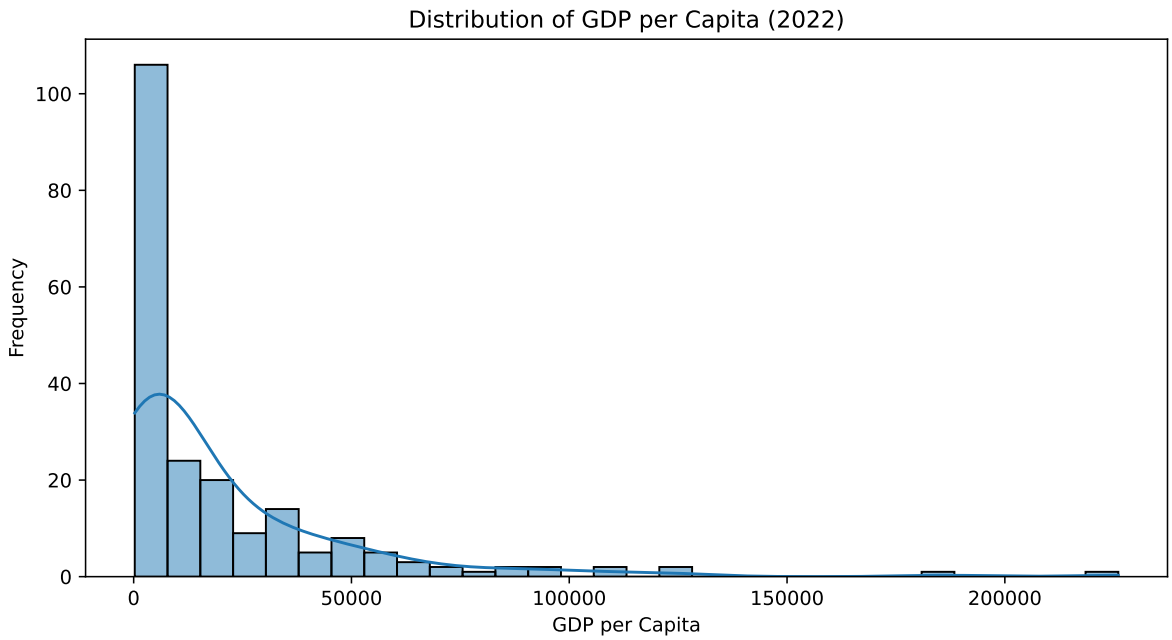
```
gdp_per_capita       10
inflation_rate       44
unemployment_rate    31
dtype: int64
```

Task 4

GDP per Capita Distribution (Histogram)

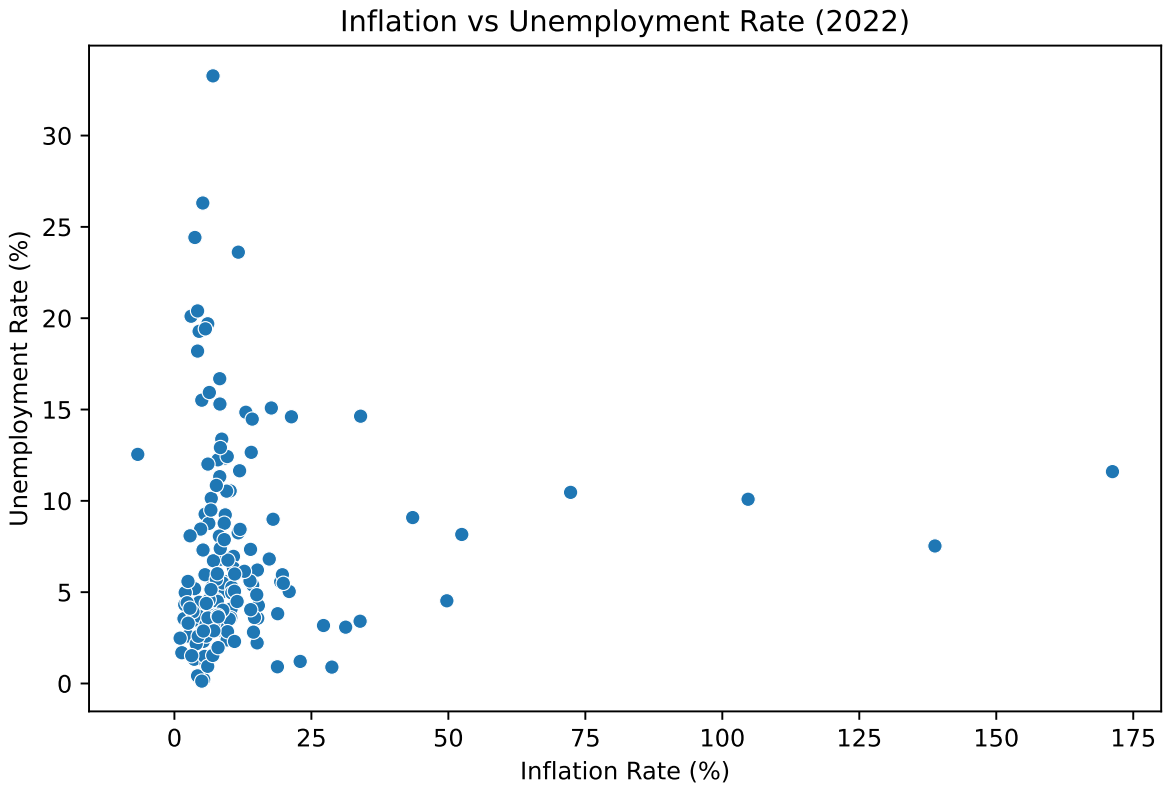
```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(10, 5))
sns.histplot(df["gdp_per_capita"].dropna(), bins=30, kde=True)
```

```
plt.xlabel("GDP per Capita")
plt.ylabel("Frequency")
plt.title("Distribution of GDP per Capita (2022)")
plt.show()
```



Scatter Plot: Inflation Rate vs Unemployment Rate

```
plt.figure(figsize=(8, 5))
sns.scatterplot(x=df["inflation_rate"], y=df["unemployment_rate"])
plt.xlabel("Inflation Rate (%)")
plt.ylabel("Unemployment Rate (%)")
plt.title("Inflation vs Unemployment Rate (2022)")
plt.show()
```



Task 5

```
summary_table = df[selected_indicators].describe().T
summary_table
```

	count	mean	std	min	25%	50%	75%
gdp_per_capita	207.0	20520.336828	30640.741594	250.634225	2599.752468	7606.237525	27542
inflation_rate	173.0	12.404067	19.467053	-6.687321	5.518129	7.930929	11.665
unemployment_rate	186.0	7.227344	5.844462	0.130000	3.478000	5.334000	9.2617