# Code :

import pandas as pd

import numpy as np

from sklearn.linear\_model import LinearRegression

# Load the investment data

investment\_file = 'finvestment.xlsx'

df = pd.read\_excel(investment\_file)

# Convert columns starting with 'YR' to numerical type

for col in df.columns:

if col.startswith('YR'):

df[col] = pd.to\_numeric(df[col], errors='coerce')

# Initialize lists to store results

mean\_investments = []

median\_investments = []

growth\_rates = []

# Function to calculate growth rate using linear regression

def calculate\_growth\_rate(values, years):

values = np.array(values)

years = np.array(years)

# Check lengths

if len(years) != len(values):

print(f"Length of years: {len(years)}, Length of values: {len(values)}")

raise ValueError("Mismatch between years and values lengths.")

# Create a boolean mask to filter out NaNs

valid = ~np.isnan(values)

# Filter valid data

valid\_years = years[valid]

valid\_values = values[valid]

# Check if there are enough valid data points

if len(valid\_years) < 2:

return np.nan

model = LinearRegression()

model.fit(valid\_years.reshape(-1, 1), valid\_values)

growth\_rate = model.coef\_[0] \* 100 # Convert to percentage

return growth\_rate

# Group by country and calculate statistics

grouped = df.groupby('Country Name')

for country, group in grouped:

# Drop non-numeric columns

numeric\_data = group.drop(columns=['Series Name', 'Country Name'])

# Check that numeric\_data contains the correct number of columns

year\_columns = [col for col in numeric\_data.columns if col.startswith('YR')]

years = [int(col.replace('YR', '')) for col in year\_columns]

for idx, row in numeric\_data.iterrows():

values = row[year\_columns].values

print(f"Country: {country}, Years: {years}, Values: {values}")

# Calculate mean and median for each country

mean\_investments.append(numeric\_data[year\_columns].mean(axis=1).mean())

median\_investments.append(numeric\_data[year\_columns].median(axis=1).median())

# Calculate growth rate

growth\_rate = calculate\_growth\_rate(values, years)

growth\_rates.append(growth\_rate)

# Create a summary DataFrame

summary\_df = pd.DataFrame({

'Country Name': grouped.size().index,

'Mean Investment': mean\_investments,

'Median Investment': median\_investments,

'Growth Rate (%)': growth\_rates

})

# Save the summary DataFrame to an Excel file

summary\_df.to\_excel('investment\_analysis\_summary.xlsx', index=False)

print("Investment analysis completed and saved to 'investment\_analysis\_summary.xlsx'.")

print(summary\_df)

# output :

Country: africa eastern and southern, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Country: africa western and central, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Country: albania, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 2.846e+08 0.000e+00

0.000e+00 0.000e+00 1.248e+08 0.000e+00 0.000e+00]

Country: arab world, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Country: argentina, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [2.088e+09 1.298e+08 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00

0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00]

Country: bangladesh, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 1.7958e+08

0.0000e+00 9.0926e+08 8.6100e+08 0.0000e+00 3.7753e+08 0.0000e+00]

Country: brazil, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.000000e+00 1.616400e+09 3.374923e+10 1.752800e+09 8.961000e+08

5.496000e+08 1.041400e+09 2.763010e+09 2.821600e+08 6.527030e+09

1.041998e+10 0.000000e+00]

Country: cameroon, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 5.680e+08 0.000e+00

0.000e+00 3.073e+07 0.000e+00 5.250e+07 0.000e+00]

Country: caribbean small states, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Country: central europe and the baltics, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Country: china, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [1.730000e+08 1.558500e+09 5.009410e+09 2.913740e+09 3.469630e+09

1.258744e+10 2.596480e+10 2.251550e+10 2.788890e+09 6.504820e+09

2.843359e+10 0.000000e+00]

Country: colombia, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.00000e+00 1.04770e+09 4.24790e+09 4.49590e+09 5.99854e+09 3.68500e+08

1.86474e+09 2.64500e+09 1.00000e+09 5.86220e+08 2.07000e+09 0.00000e+00]

Country: costa rica, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.00e+00 1.61e+08 0.00e+00 6.63e+08 0.00e+00 0.00e+00 0.00e+00 0.00e+00

0.00e+00 0.00e+00 0.00e+00 0.00e+00]

Country: cote d'ivoire, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.0000e+00 1.4000e+08 0.0000e+00 0.0000e+00 0.0000e+00 4.7130e+08

0.0000e+00 0.0000e+00 0.0000e+00 4.6831e+08 1.7560e+08 0.0000e+00]

Country: early-demographic dividend, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00

1.417791e+10 0.000000e+00 1.689165e+10 0.000000e+00 0.000000e+00

0.000000e+00 0.000000e+00]

Country: east asia & pacific (excluding high income), Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [1.926900e+09 2.644600e+09 5.846510e+09 3.937640e+09 6.408030e+09

2.453999e+10 2.946223e+10 3.097380e+10 2.788890e+09 1.599241e+10

3.228289e+10 0.000000e+00]

Country: east asia & pacific (ida & ibrd countries), Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [1.926900e+09 2.644600e+09 5.846510e+09 3.937640e+09 6.408030e+09

2.453999e+10 2.946223e+10 3.097380e+10 2.788890e+09 1.599241e+10

3.228289e+10 0.000000e+00]

Country: ecuador, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.00e+00 0.00e+00 0.00e+00 0.00e+00 6.65e+08 0.00e+00 3.77e+08 1.00e+08

5.00e+08 2.60e+08 0.00e+00 0.00e+00]

Country: egypt, arab rep., Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.00000e+00 3.68300e+08 0.00000e+00 0.00000e+00 0.00000e+00 4.98000e+08

0.00000e+00 1.50000e+08 5.01885e+09 0.00000e+00 0.00000e+00 0.00000e+00]

Country: euro area, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Country: europe & central asia, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Country: europe & central asia (excluding high income), Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 3.53848e+09

0.00000e+00 0.00000e+00 0.00000e+00 9.76582e+09 0.00000e+00 0.00000e+00]

Country: europe & central asia (ida & ibrd countries), Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 3.58888e+09

0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00]

Country: european union, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Country: fragile and conflict affected situations, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Country: gabon, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 3.4970e+08 0.0000e+00

0.0000e+00 3.4000e+08 2.2203e+08 0.0000e+00 0.0000e+00 0.0000e+00]

Country: georgia, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [ 0. 0. 0. 0. 0. 0. 0.

93000000. 0. 0. 0. 0.]

Country: ghana, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.0e+00 1.0e+07 0.0e+00 0.0e+00 1.5e+09 5.5e+08 0.0e+00 0.0e+00 0.0e+00

0.0e+00 0.0e+00 0.0e+00]

Country: heavily indebted poor countries (hipc), Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Country: high income, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Country: honduras, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.00e+00 1.20e+08 2.25e+07 8.79e+07 0.00e+00 0.00e+00 0.00e+00 0.00e+00

2.09e+08 0.00e+00 0.00e+00 0.00e+00]

Country: ibrd only, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [8.620200e+09 7.634900e+09 5.311169e+10 6.560606e+10 1.892749e+10

3.356987e+10 4.922832e+10 4.259428e+10 1.356620e+10 4.056269e+10

5.549419e+10 0.000000e+00]

Country: india, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [1.90000e+06 9.64000e+07 2.93165e+09 2.04722e+09 2.62312e+09 3.48290e+09

6.60752e+09 6.46984e+09 1.86186e+09 5.82331e+09 9.27164e+09 0.00000e+00]

Country: indonesia, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [1.16000e+08 0.00000e+00 0.00000e+00 0.00000e+00 4.62000e+08 6.00000e+09

2.90343e+09 0.00000e+00 0.00000e+00 6.73729e+09 9.33000e+07 0.00000e+00]

Country: iran, islamic rep., Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.00e+00 0.00e+00 0.00e+00 0.00e+00 2.35e+08 0.00e+00 0.00e+00 0.00e+00

0.00e+00 0.00e+00 0.00e+00 0.00e+00]

Country: iraq, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.0e+00 0.0e+00 1.3e+08 0.0e+00 0.0e+00 1.2e+08 0.0e+00 0.0e+00 0.0e+00

2.0e+08 0.0e+00 0.0e+00]

Country: jamaica, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.00e+00 0.00e+00 0.00e+00 0.00e+00 4.52e+08 0.00e+00 0.00e+00 1.10e+08

0.00e+00 0.00e+00 0.00e+00 0.00e+00]

Country: japan, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Country: jordan, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [ 0. 0. 94000000. 0. 0. 0. 0.

0. 0. 0. 0. 0.]

Country: kazakhstan, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 7.50e+07 0.00e+00 0.00e+00

5.85e+08 3.00e+08 0.00e+00 0.00e+00]

Country: kenya, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 9.6890e+07

2.1392e+08 0.0000e+00 5.7597e+08 0.0000e+00 1.3080e+08 0.0000e+00]

Country: lao pdr, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.00e+00 1.50e+06 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 5.70e+09

0.00e+00 0.00e+00 9.15e+07 0.00e+00]

Country: late-demographic dividend, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.000000e+00 5.722700e+09 4.300654e+10 1.164764e+10 1.090017e+10

2.097154e+10 3.161211e+10 3.255189e+10 5.581760e+09 1.612467e+10

4.289407e+10 0.000000e+00]

Country: latin america & caribbean, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.000000e+00 4.488000e+09 3.944584e+10 1.478370e+10 8.453240e+09

0.000000e+00 4.087500e+09 0.000000e+00 2.357860e+09 0.000000e+00

0.000000e+00 0.000000e+00]

Country: latin america & caribbean (excluding high income), Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.000000e+00 4.488000e+09 3.944584e+10 1.478370e+10 8.453240e+09

2.340100e+09 4.087500e+09 8.496010e+09 2.357860e+09 8.158650e+09

1.352516e+10 0.000000e+00]

Country: latin america & the caribbean (ida & ibrd countries), Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.000000e+00 4.488000e+09 3.944584e+10 1.478370e+10 8.453240e+09

0.000000e+00 4.087500e+09 0.000000e+00 2.357860e+09 0.000000e+00

1.400116e+10 0.000000e+00]

Country: least developed countries: un classification, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Country: low & middle income, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00

3.722164e+10 0.000000e+00 5.328791e+10 0.000000e+00 4.275850e+10

0.000000e+00 0.000000e+00]

Country: low income, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Country: lower middle income, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00

0.00000e+00 1.75874e+10 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00]

Country: malaysia, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [7.0000e+05 1.0797e+09 0.0000e+00 0.0000e+00 0.0000e+00 1.5430e+09

0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00]

Country: mexico, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [4.6034e+09 7.9360e+08 6.2940e+08 7.8710e+08 4.2270e+08 1.0069e+09

5.8936e+08 2.6000e+08 1.3770e+08 6.5540e+08 2.2800e+08 0.0000e+00]

Country: middle income, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.000000e+00 8.088090e+09 0.000000e+00 0.000000e+00 2.042749e+10

3.650524e+10 0.000000e+00 5.268354e+10 0.000000e+00 4.152900e+10

5.697062e+10 0.000000e+00]

Country: peru, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.000e+00 0.000e+00 6.196e+08 6.997e+09 1.890e+07 4.151e+08 2.150e+08

0.000e+00 2.290e+08 0.000e+00 8.000e+08 0.000e+00]

Country: philippines, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.00000e+00 4.90000e+06 8.37100e+08 1.02390e+09 2.47190e+09 5.35500e+07

0.00000e+00 1.73325e+09 0.00000e+00 1.86000e+09 2.17000e+09 0.00000e+00]

Country: russian federation, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.00000e+00 1.09400e+08 0.00000e+00 1.82220e+09 7.94000e+07 2.03700e+09

1.62270e+09 3.35777e+09 9.25710e+08 1.09000e+08 0.00000e+00 0.00000e+00]

Country: south asia, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [1.90000e+06 9.64000e+07 2.93165e+09 2.39722e+09 2.62312e+09 3.78248e+09

6.60752e+09 7.61910e+09 2.72286e+09 6.47331e+09 9.64917e+09 0.00000e+00]

Country: south asia (ida & ibrd), Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [1.90000e+06 9.64000e+07 2.93165e+09 2.39722e+09 2.62312e+09 3.78248e+09

6.60752e+09 7.61910e+09 2.72286e+09 6.47331e+09 9.64917e+09 0.00000e+00]

Country: turkiye, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.00000e+00 0.00000e+00 4.74340e+09 4.31032e+10 7.79000e+08 9.91880e+08

6.92390e+09 2.37800e+08 0.00000e+00 8.22200e+09 3.00000e+07 0.00000e+00]

Country: turkmenistan, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Country: turks and caicos islands, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Country: tuvalu, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Country: uganda, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00 0.000e+00

0.000e+00 0.000e+00 2.295e+08 0.000e+00 0.000e+00]

Country: ukraine, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.0e+00 0.0e+00 0.0e+00 0.0e+00 0.0e+00 1.5e+08 0.0e+00 0.0e+00 1.5e+07

1.2e+08 0.0e+00 0.0e+00]

Country: united arab emirates, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Country: united kingdom, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Country: united states, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Country: upper middle income, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [8.618300e+09 6.998700e+09 4.924894e+10 6.253494e+10 1.359297e+10

2.933502e+10 4.262080e+10 3.509614e+10 6.670490e+09 3.124658e+10

4.208205e+10 0.000000e+00]

Country: uruguay, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00

0.00e+00 2.20e+08 4.76e+08 0.00e+00]

Country: uzbekistan, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Country: vanuatu, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Country: venezuela, rb, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Country: viet nam, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 4.50000e+06 0.00000e+00

0.00000e+00 1.02505e+09 0.00000e+00 8.90300e+08 1.49450e+09 0.00000e+00]

Country: virgin islands (u.s.), Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Country: west bank and gaza, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Country: yemen, rep., Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Country: zambia, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Country: zimbabwe, Years: [1990, 2000, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023], Values: [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Investment analysis completed and saved to 'investment\_analysis\_summary.xlsx'.

Country Name Mean Investment Median Investment \

0 africa eastern and southern 0.000000e+00 0.0

1 africa western and central 0.000000e+00 0.0

2 albania 3.411667e+07 0.0

3 arab world 0.000000e+00 0.0

4 argentina 1.848167e+08 0.0

.. ... ... ...

73 virgin islands (u.s.) 0.000000e+00 0.0

74 west bank and gaza 0.000000e+00 0.0

75 yemen, rep. 0.000000e+00 0.0

76 zambia 0.000000e+00 0.0

77 zimbabwe 0.000000e+00 0.0

Growth Rate (%)

0 0.000000e+00

1 0.000000e+00

2 1.413771e+08

3 0.000000e+00

4 -5.054808e+09

.. ...

73 0.000000e+00

74 0.000000e+00

75 0.000000e+00

76 0.000000e+00

77 0.000000e+00

# CODE:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

# Load the mortality data

mortality\_file = 'fmortality.xlsx'

df = pd.read\_excel(mortality\_file)

# Convert columns starting with 'YR' to numerical type

for col in df.columns:

if col.startswith('YR'):

df[col] = pd.to\_numeric(df[col], errors='coerce')

# Calculate average mortality rate for each country

df['Average Mortality Rate'] = df.filter(like='YR').mean(axis=1)

# Initialize lists to store results

countries = df['Country Name'].unique()

trends = []

# Analyze trends over time

for country in countries:

country\_data = df[df['Country Name'] == country]

years = [int(col.replace('YR', '')) for col in df.columns if col.startswith('YR')]

mortality\_rates = country\_data.filter(like='YR').values.flatten()

# Remove NaN values

valid\_indices = ~np.isnan(mortality\_rates)

valid\_years = np.array(years \* len(country\_data))[valid\_indices]

valid\_mortality\_rates = mortality\_rates[valid\_indices]

# Plot trend

plt.figure(figsize=(10, 6))

plt.plot(valid\_years, valid\_mortality\_rates, marker='o', linestyle='-', label=country)

plt.xlabel('Year')

plt.ylabel('Mortality Rate (per 100,000 population)')

plt.title(f'Mortality Rate Trends for {country}')

plt.legend()

plt.grid(True)

plt.savefig(f'{country}\_mortality\_trends.png')

plt.close()

# Print average mortality rates

print("Average Mortality Rates:")

print(df[['Country Name', 'Average Mortality Rate']])

# Save the results to an Excel file

df[['Country Name', 'Average Mortality Rate']].to\_excel('average\_mortality\_rates.xlsx', index=False)

print("Mortality rates analysis completed and saved to 'average\_mortality\_rates.xlsx'.")

OUTPUT:

|  |  |
| --- | --- |
| **Country Name** | **Average Mortality Rate** |
| albania | 7.466667 |
| argentina | 7.883333 |
| bangladesh | 8.725 |
| brazil | 11.31667 |
| cameroon | 17.05 |
| china | 10.775 |
| colombia | 10.2 |
| costa rica | 9.083333 |
| cote d'ivoire | 13.88333 |
| ecuador | 12.025 |
| egypt, arab rep. | 6.333333 |
| gabon | 13.46667 |
| georgia | 7.875 |
| ghana | 14.55833 |
| honduras | 9.358333 |
| india | 9.183333 |
| indonesia | 7.191667 |
| iran, islamic rep. | 12.96667 |
| iraq | 13.075 |
| jamaica | 7.775 |
| japan | 3.016667 |
| jordan | 12.60833 |
| kazakhstan | 9.166667 |
| kenya | 15.975 |
| lao pdr | 9.483333 |
| malaysia | 13.725 |
| mexico | 7.8 |
| peru | 8.191667 |
| philippines | 6.758333 |
| russian federation | 9.875 |
| turkiye | 5.2 |
| turkmenistan | 8.766667 |
| turks and caicos islands | 0 |
| tuvalu | 0 |
| uganda | 17.81667 |
| ukraine | 7.025 |
| united arab emirates | 8.608333 |
| united kingdom | 2.1 |
| united states | 7.508333 |
| uruguay | 8.875 |
| uzbekistan | 6.425 |
| vanuatu | 8.758333 |
| venezuela, rb | 20.61667 |
| viet nam | 15.70833 |
| virgin islands (u.s.) | 0 |
| west bank and gaza | 0.441667 |
| yemen, rep. | 15.4 |
| zambia | 12.64167 |
| zimbabwe | 22.30833 |
| africa eastern and southern | 17.02974 |
| africa western and central | 14.58651 |
| arab world | 11.64485 |
| caribbean small states | 8.53776 |
| central europe and the baltics | 5.738324 |
| early-demographic dividend | 9.554469 |
| east asia & pacific (excluding high income) | 10.74218 |
| east asia & pacific (ida & ibrd countries) | 10.70376 |
| euro area | 3.669332 |
| europe & central asia | 5.48832 |
| europe & central asia (excluding high income) | 7.830976 |
| europe & central asia (ida & ibrd countries) | 7.603479 |
| european union | 4.038773 |
| fragile and conflict affected situations | 14.6029 |
| heavily indebted poor countries (hipc) | 16.07363 |
| high income | 5.357123 |
| ibrd only | 10.08163 |
| late-demographic dividend | 11.0184 |
| latin america & caribbean | 10.47511 |
| latin america & caribbean (excluding high income) | 10.04997 |
| latin america & the caribbean (ida & ibrd countries) | 10.56898 |
| least developed countries: un classification | 14.45647 |
| low & middle income | 10.87422 |
| low income | 16.32064 |
| lower middle income | 10.21055 |
| middle income | 10.30615 |
| south asia | 8.972203 |
| south asia (ida & ibrd) | 8.972203 |
| upper middle income | 10.39281 |

CODE:

import pandas as pd

import numpy as np

# Load the SPI data

spi\_file = 'fSPI.xlsx'

df = pd.read\_excel(spi\_file)

# Ensure that all SPI columns are numeric

# Assuming all columns are SPI columns starting with 'YR'

for col in df.columns:

if col.startswith('YR'):

df[col] = pd.to\_numeric(df[col], errors='coerce')

# Drop non-numeric columns for SPI calculations

spi\_data = df.drop(columns=['Series Name', 'Country Name'])

# Calculate summary statistics

mean\_spi = spi\_data.mean(axis=1)

median\_spi = spi\_data.median(axis=1)

range\_spi = spi\_data.max(axis=1) - spi\_data.min(axis=1)

# Add summary statistics to DataFrame

df['Mean SPI'] = mean\_spi

df['Median SPI'] = median\_spi

df['Range SPI'] = range\_spi

# Print summary statistics

print("Summary Statistics for SPI Scores:")

print(f"Mean:\n{df['Mean SPI'].describe()}")

print(f"Median:\n{df['Median SPI'].describe()}")

print(f"Range:\n{df['Range SPI'].describe()}")

# Save the results to an Excel file

df[['Country Name', 'Mean SPI', 'Median SPI', 'Range SPI']].to\_excel('spi\_summary\_statistics.xlsx', index=False)

print("SPI performance analysis completed and saved to 'spi\_summary\_statistics.xlsx'.")

OUPUT:  
  
Summary Statistics for SPI Scores:

Mean:

count 78.000000

mean 18.301282

std 18.185870

min 0.000000

25% 0.000000

50% 19.791667

75% 30.833333

max 58.333333

Name: Mean SPI, dtype: float64

Median:

count 78.000000

mean 26.474359

std 28.078149

min 0.000000

25% 0.000000

50% 21.250000

75% 46.875000

max 100.000000

Name: Median SPI, dtype: float64

Range:

count 78.000000

mean 37.756410

std 35.443823

min 0.000000

25% 0.000000

50% 42.500000

75% 68.750000

max 100.000000

Name: Range SPI, dtype: float64

SPI performance analysis completed and saved to 'spi\_summary\_statistics.xlsx'.

CODE:  
  
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

# Load the SPI data

spi\_file = 'fSPI.xlsx'

df = pd.read\_excel(spi\_file)

# Convert columns starting with 'YR' to numerical type

for col in df.columns:

if col.startswith('YR'):

df[col] = pd.to\_numeric(df[col], errors='coerce')

# Extract year columns

year\_columns = [col for col in df.columns if col.startswith('YR')]

years = [int(col.replace('YR', '')) for col in year\_columns]

# Prepare to collect descriptive statistics

descriptive\_stats = []

# Plot infrastructure performance scores over time for each country

countries = df['Country Name'].unique()

plt.figure(figsize=(14, 10))

for country in countries:

country\_data = df[df['Country Name'] == country]

spi\_scores = country\_data[year\_columns].values.flatten()

# Remove NaN values

valid\_indices = ~np.isnan(spi\_scores)

valid\_years = np.array(years \* len(country\_data))[valid\_indices]

valid\_spi\_scores = spi\_scores[valid\_indices]

# Calculate descriptive statistics

mean\_spi = np.mean(valid\_spi\_scores)

median\_spi = np.median(valid\_spi\_scores)

std\_dev\_spi = np.std(valid\_spi\_scores)

min\_spi = np.min(valid\_spi\_scores)

max\_spi = np.max(valid\_spi\_scores)

descriptive\_stats.append({

'Country': country,

'Mean SPI Score': mean\_spi,

'Median SPI Score': median\_spi,

'Standard Deviation': std\_dev\_spi,

'Min SPI Score': min\_spi,

'Max SPI Score': max\_spi

})

# Plot

plt.plot(valid\_years, valid\_spi\_scores, marker='o', linestyle='-', label=country)

plt.xlabel('Year')

plt.ylabel('SPI Score')

plt.title('Infrastructure Performance Scores Over Time by Country')

plt.legend(loc='upper left', bbox\_to\_anchor=(1, 1))

plt.grid(True)

plt.tight\_layout()

# Save the plot

plt.savefig('infrastructure\_performance\_trends.png')

plt.show()

# Print descriptive statistics

stats\_df = pd.DataFrame(descriptive\_stats)

print("Descriptive Statistics for Infrastructure Performance Scores:")

print(stats\_df)

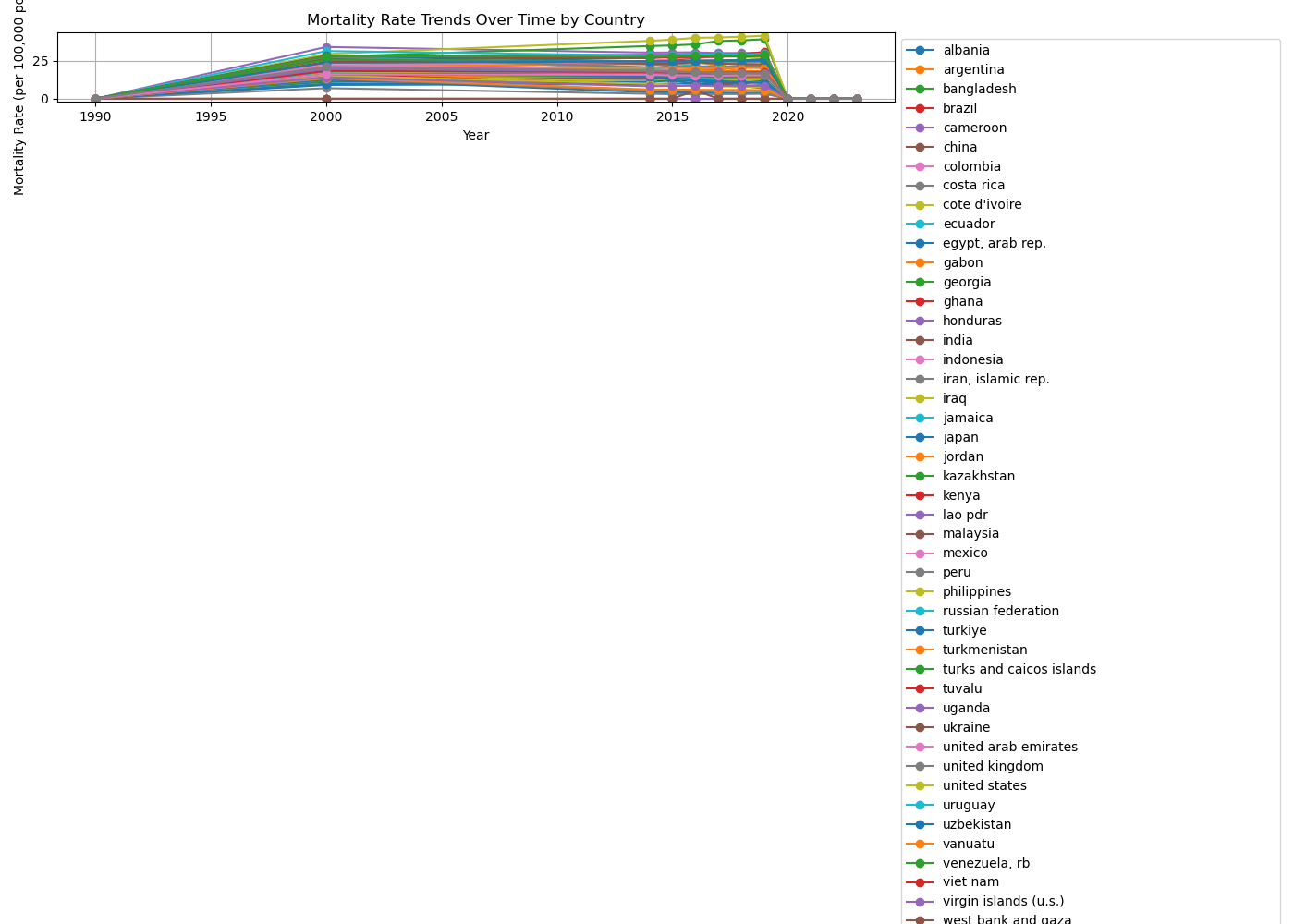
print("Infrastructure performance trends plotted and saved to 'infrastructure\_performance\_trends.png'.")  
  
  
output:

Descriptive Statistics and Trends for Mortality Rates

# Descriptive Statistics for Mortality Rates

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Country | Mean Mortality Rate | Median Mortality Rate | Standard Deviation | Min Mortality Rate | Max Mortality Rate |
| albania | 7.466666666666668 | 11.2 | 6.384660436459319 | 0.0 | 14.3 |
| argentina | 7.883333333333333 | 12.45 | 6.694753335427842 | 0.0 | 14.1 |
| bangladesh | 8.725 | 13.05 | 7.488449438969326 | 0.0 | 15.9 |
| brazil | 11.316666666666668 | 16.55 | 9.725382026201109 | 0.0 | 23.6 |
| cameroon | 17.05 | 28.15 | 14.424891680702492 | 0.0 | 30.2 |
| china | 10.775 | 17.5 | 9.158067936706592 | 0.0 | 21.4 |
| colombia | 10.200000000000001 | 15.850000000000001 | 8.6880185696548 | 0.0 | 20.0 |
| costa rica | 9.083333333333334 | 14.3 | 7.7389526997449005 | 0.0 | 17.8 |
| cote d'ivoire | 13.883333333333333 | 23.1 | 11.747257363127595 | 0.0 | 25.4 |
| ecuador | 12.025 | 16.8 | 10.451963531636851 | 0.0 | 24.0 |
| egypt, arab rep. | 6.333333333333333 | 9.95 | 5.391248051755322 | 0.0 | 12.3 |
| gabon | 13.466666666666669 | 22.0 | 11.401559347543454 | 0.0 | 23.9 |
| georgia | 7.875 | 11.8 | 6.750570963506223 | 0.0 | 15.8 |
| ghana | 14.558333333333332 | 24.2 | 12.31330299129992 | 0.0 | 25.8 |
| honduras | 9.358333333333333 | 15.7 | 7.9122750977340415 | 0.0 | 16.4 |
| india | 9.183333333333334 | 15.25 | 7.772369151180496 | 0.0 | 16.9 |
| indonesia | 7.191666666666666 | 11.45 | 6.124331029227238 | 0.0 | 14.5 |
| iran, islamic rep. | 12.966666666666667 | 20.65 | 11.060841840575346 | 0.0 | 26.8 |
| iraq | 13.075000000000001 | 20.4 | 11.289901461040305 | 0.0 | 27.3 |
| jamaica | 7.7749999999999995 | 12.0 | 6.622829833235941 | 0.0 | 15.1 |
| japan | 3.016666666666667 | 3.6500000000000004 | 3.4277867819078565 | 0.0 | 12.5 |
| jordan | 12.608333333333334 | 17.3 | 10.876692077812793 | 0.0 | 24.1 |
| kazakhstan | 9.166666666666666 | 13.25 | 7.971755696763976 | 0.0 | 19.3 |
| kenya | 15.975 | 26.5 | 13.510312419284265 | 0.0 | 28.3 |
| lao pdr | 9.483333333333334 | 14.5 | 8.073292733176194 | 0.0 | 17.9 |
| malaysia | 13.725 | 22.6 | 11.64947244871343 | 0.0 | 26.8 |
| mexico | 7.8 | 12.9 | 6.600631282940544 | 0.0 | 14.2 |
| peru | 8.191666666666666 | 13.6 | 6.945437151268609 | 0.0 | 15.8 |
| philippines | 6.758333333333333 | 10.350000000000001 | 5.749559161844981 | 0.0 | 12.5 |
| russian federation | 9.874999999999998 | 12.45 | 9.249245464721254 | 0.0 | 27.9 |
| turkiye | 5.2 | 7.550000000000001 | 4.461501989240843 | 0.0 | 9.9 |
| turkmenistan | 8.766666666666667 | 13.8 | 7.458030720117892 | 0.0 | 16.4 |
| turks and caicos islands | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| tuvalu | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| uganda | 17.816666666666666 | 29.549999999999997 | 15.09623757394169 | 0.0 | 33.9 |
| ukraine | 7.0249999999999995 | 10.1 | 6.068652376489088 | 0.0 | 13.7 |
| united arab emirates | 8.608333333333334 | 10.100000000000001 | 7.721232450558366 | 0.0 | 18.1 |
| united kingdom | 2.1 | 2.95 | 2.053046516764781 | 0.0 | 6.9 |
| united states | 7.508333333333333 | 11.7 | 6.418652290179163 | 0.0 | 15.7 |
| uruguay | 8.875 | 13.75 | 7.549185938452791 | 0.0 | 16.8 |
| uzbekistan | 6.425 | 9.95 | 5.4601778054076835 | 0.0 | 11.9 |
| vanuatu | 8.758333333333333 | 14.45 | 7.4098423659946295 | 0.0 | 15.5 |
| venezuela, rb | 20.61666666666667 | 30.95 | 17.642034715102703 | 0.0 | 39.0 |
| viet nam | 15.708333333333334 | 24.4 | 13.38459676477239 | 0.0 | 30.6 |
| virgin islands (u.s.) | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| west bank and gaza | 0.44166666666666665 | 0.0 | 1.4648426157403016 | 0.0 | 5.3 |
| yemen, rep. | 15.399999999999999 | 23.1 | 13.142298124757327 | 0.0 | 29.4 |
| zambia | 12.641666666666666 | 20.5 | 10.833932675728708 | 0.0 | 27.4 |
| zimbabwe | 22.308333333333334 | 33.6 | 19.07544050751006 | 0.0 | 41.2 |
| africa eastern and southern | 17.029744958956403 | 28.411961456499434 | 14.408755860514376 | 0.0 | 31.21171011881088 |
| africa western and central | 14.58651014380609 | 24.3964363841949 | 12.36788579444977 | 0.0 | 28.1901840782264 |
| arab world | 11.644849539785843 | 19.65705830552318 | 9.843084764795282 | 0.0 | 20.19808073920409 |
| caribbean small states | 8.537760274165137 | 14.072811204597695 | 7.24216005515367 | 0.0 | 16.56420313174923 |
| central europe and the baltics | 5.738323577025761 | 8.581077212938492 | 5.218503909440614 | 0.0 | 15.99063982971203 |
| early-demographic dividend | 9.554469017063866 | 16.177356637713082 | 8.076683035122032 | 0.0 | 16.84754551050895 |
| east asia & pacific (excluding high income) | 10.742178829272566 | 17.7236219624504 | 9.111431693678025 | 0.0 | 20.81189419886947 |
| east asia & pacific (ida & ibrd countries) | 10.703758432943834 | 17.64402629791669 | 9.08023633271862 | 0.0 | 20.78729531156606 |
| euro area | 3.669332266687846 | 4.95565330159863 | 3.77224057619437 | 0.0 | 13.1659929378945 |
| europe & central asia | 5.488320177107624 | 7.608019421783419 | 4.993036045345993 | 0.0 | 15.04903649791817 |
| europe & central asia (excluding high income) | 7.830976181976754 | 10.783152618746715 | 6.898431063222207 | 0.0 | 18.35907176916058 |
| europe & central asia (ida & ibrd countries) | 7.60347875342374 | 10.61375174061095 | 6.694206010454129 | 0.0 | 18.03971623849077 |
| european union | 4.038772612670976 | 5.653862161363103 | 3.9773109125759536 | 0.0 | 13.45938669563903 |
| fragile and conflict affected situations | 14.602904594368548 | 24.646550883395946 | 12.344562429431969 | 0.0 | 25.57200465918897 |
| heavily indebted poor countries (hipc) | 16.073633838546357 | 27.067231621539413 | 13.58813867781728 | 0.0 | 28.16358827157177 |
| high income | 5.357123371328434 | 8.229609187478943 | 4.776156303688117 | 0.0 | 14.05292661902378 |
| ibrd only | 10.081632117597609 | 16.549537219746647 | 8.552215273103371 | 0.0 | 19.45224713379908 |
| late-demographic dividend | 11.018401668281067 | 17.64740902140283 | 9.372236064577244 | 0.0 | 21.91962095318036 |
| latin america & caribbean | 10.47510628771119 | 17.363751437249796 | 8.862897394196569 | 0.0 | 19.00125070413312 |
| latin america & caribbean (excluding high income) | 10.04997192447762 | 16.445724285624983 | 8.508391958143836 | 0.0 | 18.43361681155497 |
| latin america & the caribbean (ida & ibrd countries) | 10.568982492386228 | 17.52774797456449 | 8.943206195635716 | 0.0 | 19.21196787697923 |
| least developed countries: un classification | 14.456471611961296 | 23.990492818786393 | 12.226852590483224 | 0.0 | 25.6267757099447 |
| low & middle income | 10.874215272975102 | 18.18630221226485 | 9.203741276312233 | 0.0 | 20.13195776902686 |
| low income | 16.320636437993205 | 27.574588143686714 | 13.79615589153627 | 0.0 | 28.65550040510488 |
| lower middle income | 10.210552191822513 | 17.251991251476802 | 8.632201091781846 | 0.0 | 18.14035944721485 |
| middle income | 10.306154822980766 | 17.06105503190594 | 8.732013678324062 | 0.0 | 19.49531544840296 |
| south asia | 8.972202982409682 | 15.085418593703665 | 7.586245148980242 | 0.0 | 15.93745614784808 |
| south asia (ida & ibrd) | 8.972202982409682 | 15.085418593703665 | 7.586245148980242 | 0.0 | 15.93745614784808 |
| upper middle income | 10.39280933426016 | 16.64301913764477 | 8.843372216385303 | 0.0 | 20.78703829487171 |

# Mortality Rate Trends Over Time by Country



# Countries with Increasing Mortality Rates

bangladesh, ecuador, venezuela, rb, west bank and gaza, yemen, rep., zimbabwe

# Countries with Decreasing Mortality Rates

albania, argentina, brazil, cameroon, china, colombia, costa rica, cote d'ivoire, egypt, arab rep., gabon, georgia, ghana, honduras, india, indonesia, iran, islamic rep., iraq, jamaica, japan, jordan, kazakhstan, kenya, lao pdr, malaysia, mexico, peru, philippines, russian federation, turkiye, turkmenistan, uganda, ukraine, united arab emirates, united kingdom, united states, uruguay, uzbekistan, vanuatu, viet nam, zambia, africa eastern and southern, africa western and central, arab world, caribbean small states, central europe and the baltics, early-demographic dividend, east asia & pacific (excluding high income), east asia & pacific (ida & ibrd countries), euro area, europe & central asia, europe & central asia (excluding high income), europe & central asia (ida & ibrd countries), european union, fragile and conflict affected situations, heavily indebted poor countries (hipc), high income, ibrd only, late-demographic dividend, latin america & caribbean, latin america & caribbean (excluding high income), latin america & the caribbean (ida & ibrd countries), least developed countries: un classification, low & middle income, low income, lower middle income, middle income, south asia, south asia (ida & ibrd), upper middle income

CODE:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

# Load the investment and mortality data

investment\_file = 'finvestment.xlsx'

mortality\_file = 'fmortality.xlsx'

df\_investment = pd.read\_excel(investment\_file)

df\_mortality = pd.read\_excel(mortality\_file)

# Clean column names

df\_investment.columns = df\_investment.columns.str.strip()

df\_mortality.columns = df\_mortality.columns.str.strip()

# Drop unnecessary columns (e.g., 'Unnamed' columns)

df\_investment = df\_investment.loc[:, ~df\_investment.columns.str.contains('^Unnamed')]

df\_mortality = df\_mortality.loc[:, ~df\_mortality.columns.str.contains('^Unnamed')]

# Extract year columns

year\_columns\_investment = [col for col in df\_investment.columns if col.startswith('YR')]

year\_columns\_mortality = [col for col in df\_mortality.columns if col.startswith('YR')]

# Create long format DataFrames

df\_investment\_long = df\_investment.melt(id\_vars=['Country Name'], value\_vars=year\_columns\_investment, var\_name='Year', value\_name='Investment')

df\_mortality\_long = df\_mortality.melt(id\_vars=['Country Name'], value\_vars=year\_columns\_mortality, var\_name='Year', value\_name='Mortality')

# Remove 'YR' prefix from 'Year' column and convert to numeric

df\_investment\_long['Year'] = df\_investment\_long['Year'].str.replace('YR', '').astype(int)

df\_mortality\_long['Year'] = df\_mortality\_long['Year'].str.replace('YR', '').astype(int)

# Merge the long format DataFrames on 'Country Name' and 'Year'

merged\_long = pd.merge(df\_investment\_long, df\_mortality\_long, on=['Country Name', 'Year'])

# Calculate correlation for each year

correlation\_per\_year = merged\_long.groupby('Year').apply(lambda x: x[['Investment', 'Mortality']].corr().iloc[0, 1])

# Plot the correlations over time

plt.figure(figsize=(12, 6))

plt.plot(correlation\_per\_year.index, correlation\_per\_year.values, marker='o', linestyle='-', color='b')

plt.xlabel('Year')

plt.ylabel('Correlation between Investment and Mortality')

plt.title('Correlation between Transport Investment and Road Traffic Mortality Rates Over Time')

plt.grid(True)

plt.tight\_layout()

# Save the plot

plt.savefig('investment\_vs\_mortality\_correlation.png')

plt.show()

# Display average correlation

average\_correlation = correlation\_per\_year.mean()

print("Correlation analysis completed and plot saved to 'investment\_vs\_mortality\_correlation.png'.")

print("Average Correlation:", average\_correlation)

# Descriptive statistics for correlations

stats = {

'Mean Correlation': average\_correlation,

'Median Correlation': correlation\_per\_year.median(),

'Standard Deviation': correlation\_per\_year.std(),

'Min Correlation': correlation\_per\_year.min(),

'Max Correlation': correlation\_per\_year.max()

}

print("\nDescriptive Statistics for Correlation:")

for key, value in stats.items():

print(f"{key}: {value:.4f}")  
  
OUTPUT:  
  
Correlation analysis completed and plot saved to 'investment\_vs\_mortality\_correlation.png'.

Average Correlation: 0.03388590125706035

Descriptive Statistics for Correlation:

Mean Correlation: 0.0339

Median Correlation: 0.0251

Standard Deviation: 0.0423

Min Correlation: -0.0297

Max Correlation: 0.0935

CODE:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from docx import Document

from docx.shared import Inches

# Load the data from the three files

investment\_file = 'finvestment.xlsx'

mortality\_file = 'fmortality.xlsx'

infrastructure\_file = 'fspi.xlsx'

df\_investment = pd.read\_excel(investment\_file)

df\_mortality = pd.read\_excel(mortality\_file)

df\_infrastructure = pd.read\_excel(infrastructure\_file)

# Clean column names

df\_investment.columns = df\_investment.columns.str.strip()

df\_mortality.columns = df\_mortality.columns.str.strip()

df\_infrastructure.columns = df\_infrastructure.columns.str.strip()

# Drop unnecessary columns (e.g., 'Unnamed' columns)

df\_investment = df\_investment.loc[:, ~df\_investment.columns.str.contains('^Unnamed')]

df\_mortality = df\_mortality.loc[:, ~df\_mortality.columns.str.contains('^Unnamed')]

df\_infrastructure = df\_infrastructure.loc[:, ~df\_infrastructure.columns.str.contains('^Unnamed')]

# Extract year columns

year\_columns\_investment = [col for col in df\_investment.columns if col.startswith('YR')]

year\_columns\_mortality = [col for col in df\_mortality.columns if col.startswith('YR')]

year\_columns\_infrastructure = [col for col in df\_infrastructure.columns if col.startswith('YR')]

# Create long format DataFrames

df\_investment\_long = df\_investment.melt(id\_vars=['Country Name'], value\_vars=year\_columns\_investment, var\_name='Year', value\_name='Investment')

df\_mortality\_long = df\_mortality.melt(id\_vars=['Country Name'], value\_vars=year\_columns\_mortality, var\_name='Year', value\_name='Mortality')

df\_infrastructure\_long = df\_infrastructure.melt(id\_vars=['Country Name'], value\_vars=year\_columns\_infrastructure, var\_name='Year', value\_name='Infrastructure')

# Remove 'YR' prefix from 'Year' column and convert to numeric

df\_investment\_long['Year'] = df\_investment\_long['Year'].str.replace('YR', '').astype(int)

df\_mortality\_long['Year'] = df\_mortality\_long['Year'].str.replace('YR', '').astype(int)

df\_infrastructure\_long['Year'] = df\_infrastructure\_long['Year'].str.replace('YR', '').astype(int)

# Descriptive statistics function

def get\_descriptive\_stats(df\_long, column\_name):

stats = df\_long.groupby('Country Name')[column\_name].agg(['mean', 'median', 'std', 'min', 'max'])

return stats

# Generate descriptive statistics for each measure

investment\_stats = get\_descriptive\_stats(df\_investment\_long, 'Investment')

mortality\_stats = get\_descriptive\_stats(df\_mortality\_long, 'Mortality')

infrastructure\_stats = get\_descriptive\_stats(df\_infrastructure\_long, 'Infrastructure')

# Create a Word document

doc = Document()

doc.add\_heading('Descriptive Statistics and Time Series Analysis', 0)

# Function to add descriptive statistics table to the document

def add\_stats\_table(doc, stats, title):

doc.add\_heading(title, level=1)

table = doc.add\_table(rows=1, cols=len(stats.columns) + 1)

hdr\_cells = table.rows[0].cells

hdr\_cells[0].text = 'Country'

for i, column in enumerate(stats.columns):

hdr\_cells[i + 1].text = column.capitalize()

for country, row in stats.iterrows():

row\_cells = table.add\_row().cells

row\_cells[0].text = country

for i, value in enumerate(row):

row\_cells[i + 1].text = f"{value:.2f}"

# Add descriptive statistics tables

add\_stats\_table(doc, investment\_stats, 'Descriptive Statistics for Investment')

add\_stats\_table(doc, mortality\_stats, 'Descriptive Statistics for Mortality')

add\_stats\_table(doc, infrastructure\_stats, 'Descriptive Statistics for Infrastructure')

# Plot Investment, Mortality, and Infrastructure over time

fig, ax = plt.subplots(3, 1, figsize=(14, 18), sharex=True)

# Plot Investment over time

for country in df\_investment\_long['Country Name'].unique():

country\_data = df\_investment\_long[df\_investment\_long['Country Name'] == country]

ax[0].plot(country\_data['Year'], country\_data['Investment'], label=country)

ax[0].set\_title('Investment Over Time')

ax[0].set\_ylabel('Investment')

ax[0].legend(loc='upper right', bbox\_to\_anchor=(1.15, 1.02), title='Country')

ax[0].grid(True)

# Plot Mortality over time

for country in df\_mortality\_long['Country Name'].unique():

country\_data = df\_mortality\_long[df\_mortality\_long['Country Name'] == country]

ax[1].plot(country\_data['Year'], country\_data['Mortality'], label=country)

ax[1].set\_title('Mortality Over Time')

ax[1].set\_ylabel('Mortality')

ax[1].legend(loc='upper right', bbox\_to\_anchor=(1.15, 1.02), title='Country')

ax[1].grid(True)

# Plot Infrastructure over time

for country in df\_infrastructure\_long['Country Name'].unique():

country\_data = df\_infrastructure\_long[df\_infrastructure\_long['Country Name'] == country]

ax[2].plot(country\_data['Year'], country\_data['Infrastructure'], label=country)

ax[2].set\_title('Infrastructure Over Time')

ax[2].set\_ylabel('Infrastructure')

ax[2].set\_xlabel('Year')

ax[2].legend(loc='upper right', bbox\_to\_anchor=(1.15, 1.02), title='Country')

ax[2].grid(True)

plt.tight\_layout()

plt.savefig('time\_series\_plots.png')

plt.show()

# Add the plot to the document

doc.add\_heading('Time Series Analysis', level=1)

doc.add\_picture('time\_series\_plots.png', width=Inches(6))

# Save the document

doc.save('investment\_mortality\_infrastructure\_analysis.docx')

print("Descriptive statistics and time series plots saved to 'investment\_mortality\_infrastructure\_analysis.docx'.")  
  
output:  
Descriptive Statistics and Time Series Analysis

# Descriptive Statistics for Investment

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Country | Mean | Median | Std | Min | Max |
| africa eastern and southern | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| africa western and central | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| albania | 34116666.67 | 0.00 | 86657547.89 | 0.00 | 284600000.00 |
| arab world | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| argentina | 184816666.67 | 0.00 | 600507787.90 | 0.00 | 2088000000.00 |
| bangladesh | 193947500.00 | 0.00 | 342645461.06 | 0.00 | 909260000.00 |
| brazil | 4966475833.33 | 1328900000.00 | 9581167643.93 | 0.00 | 33749230000.00 |
| cameroon | 54269166.67 | 0.00 | 162643747.19 | 0.00 | 568000000.00 |
| caribbean small states | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| central europe and the baltics | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| china | 9326610000.00 | 4239520000.00 | 10454991238.70 | 0.00 | 28433590000.00 |
| colombia | 2027041666.67 | 1456220000.00 | 1960259259.91 | 0.00 | 5998540000.00 |
| costa rica | 68666666.67 | 0.00 | 192804344.22 | 0.00 | 663000000.00 |
| cote d'ivoire | 104600833.33 | 0.00 | 181051799.24 | 0.00 | 471300000.00 |
| early-demographic dividend | 2589130000.00 | 0.00 | 6074520870.05 | 0.00 | 16891650000.00 |
| east asia & pacific (excluding high income) | 13066990833.33 | 6127270000.00 | 12748732057.97 | 0.00 | 32282890000.00 |
| east asia & pacific (ida & ibrd countries) | 13066990833.33 | 6127270000.00 | 12748732057.97 | 0.00 | 32282890000.00 |
| ecuador | 158500000.00 | 0.00 | 235581871.51 | 0.00 | 665000000.00 |
| egypt, arab rep. | 502929166.67 | 0.00 | 1432091644.23 | 0.00 | 5018850000.00 |
| euro area | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| europe & central asia | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| europe & central asia (excluding high income) | 1108691666.67 | 0.00 | 2909884634.42 | 0.00 | 9765820000.00 |
| europe & central asia (ida & ibrd countries) | 299073333.33 | 0.00 | 1036020417.04 | 0.00 | 3588880000.00 |
| european union | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| fragile and conflict affected situations | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| gabon | 75977500.00 | 0.00 | 140750083.10 | 0.00 | 349700000.00 |
| georgia | 7750000.00 | 0.00 | 26846787.52 | 0.00 | 93000000.00 |
| ghana | 171666666.67 | 0.00 | 447108555.73 | 0.00 | 1500000000.00 |
| heavily indebted poor countries (hipc) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| high income | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| honduras | 36616666.67 | 0.00 | 67571037.82 | 0.00 | 209000000.00 |
| ibrd only | 32409657500.00 | 37066280000.00 | 21918942477.63 | 0.00 | 65606060000.00 |
| india | 3434780000.00 | 2777385000.00 | 3001403999.73 | 0.00 | 9271640000.00 |
| indonesia | 1359335000.00 | 46650000.00 | 2484187950.22 | 0.00 | 6737290000.00 |
| iran, islamic rep. | 19583333.33 | 0.00 | 67838656.63 | 0.00 | 235000000.00 |
| iraq | 37500000.00 | 0.00 | 70340082.97 | 0.00 | 200000000.00 |
| jamaica | 46833333.33 | 0.00 | 131454681.67 | 0.00 | 452000000.00 |
| japan | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| jordan | 7833333.33 | 0.00 | 27135462.65 | 0.00 | 94000000.00 |
| kazakhstan | 80000000.00 | 0.00 | 181170437.08 | 0.00 | 585000000.00 |
| kenya | 84798333.33 | 0.00 | 169949305.64 | 0.00 | 575970000.00 |
| lao pdr | 482750000.00 | 0.00 | 1643217568.45 | 0.00 | 5700000000.00 |
| late-demographic dividend | 18417757500.00 | 13886155000.00 | 15655514002.14 | 0.00 | 43006540000.00 |
| latin america & caribbean | 6134678333.33 | 1178930000.00 | 11428780697.24 | 0.00 | 39445840000.00 |
| latin america & caribbean (excluding high income) | 8844671666.67 | 6323325000.00 | 10773755248.11 | 0.00 | 39445840000.00 |
| latin america & the caribbean (ida & ibrd countries) | 7301441666.67 | 3222680000.00 | 11460201989.05 | 0.00 | 39445840000.00 |
| least developed countries: un classification | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| low & middle income | 11105670833.33 | 0.00 | 20390082399.42 | 0.00 | 53287910000.00 |
| low income | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| lower middle income | 1465616666.67 | 0.00 | 5077045062.17 | 0.00 | 17587400000.00 |
| malaysia | 218616666.67 | 0.00 | 519884831.36 | 0.00 | 1543000000.00 |
| mexico | 842796666.67 | 609380000.00 | 1221811293.03 | 0.00 | 4603400000.00 |
| middle income | 18016998333.33 | 4044045000.00 | 22695314131.50 | 0.00 | 56970620000.00 |
| peru | 774550000.00 | 116950000.00 | 1978385750.37 | 0.00 | 6997000000.00 |
| philippines | 846216666.67 | 445325000.00 | 974304070.59 | 0.00 | 2471900000.00 |
| russian federation | 838598333.33 | 109200000.00 | 1121106727.99 | 0.00 | 3357770000.00 |
| south asia | 3742060833.33 | 2827255000.00 | 3185712566.59 | 0.00 | 9649170000.00 |
| south asia (ida & ibrd) | 3742060833.33 | 2827255000.00 | 3185712566.59 | 0.00 | 9649170000.00 |
| turkiye | 5419265000.00 | 508400000.00 | 12229451363.23 | 0.00 | 43103200000.00 |
| turkmenistan | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| turks and caicos islands | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| tuvalu | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| uganda | 19125000.00 | 0.00 | 66250943.39 | 0.00 | 229500000.00 |
| ukraine | 23750000.00 | 0.00 | 52532457.50 | 0.00 | 150000000.00 |
| united arab emirates | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| united kingdom | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| united states | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| upper middle income | 27337077500.00 | 30290800000.00 | 19947364901.65 | 0.00 | 62534940000.00 |
| uruguay | 58000000.00 | 0.00 | 146041090.11 | 0.00 | 476000000.00 |
| uzbekistan | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| vanuatu | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| venezuela, rb | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| viet nam | 284529166.67 | 0.00 | 531330228.79 | 0.00 | 1494500000.00 |
| virgin islands (u.s.) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| west bank and gaza | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| yemen, rep. | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| zambia | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| zimbabwe | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

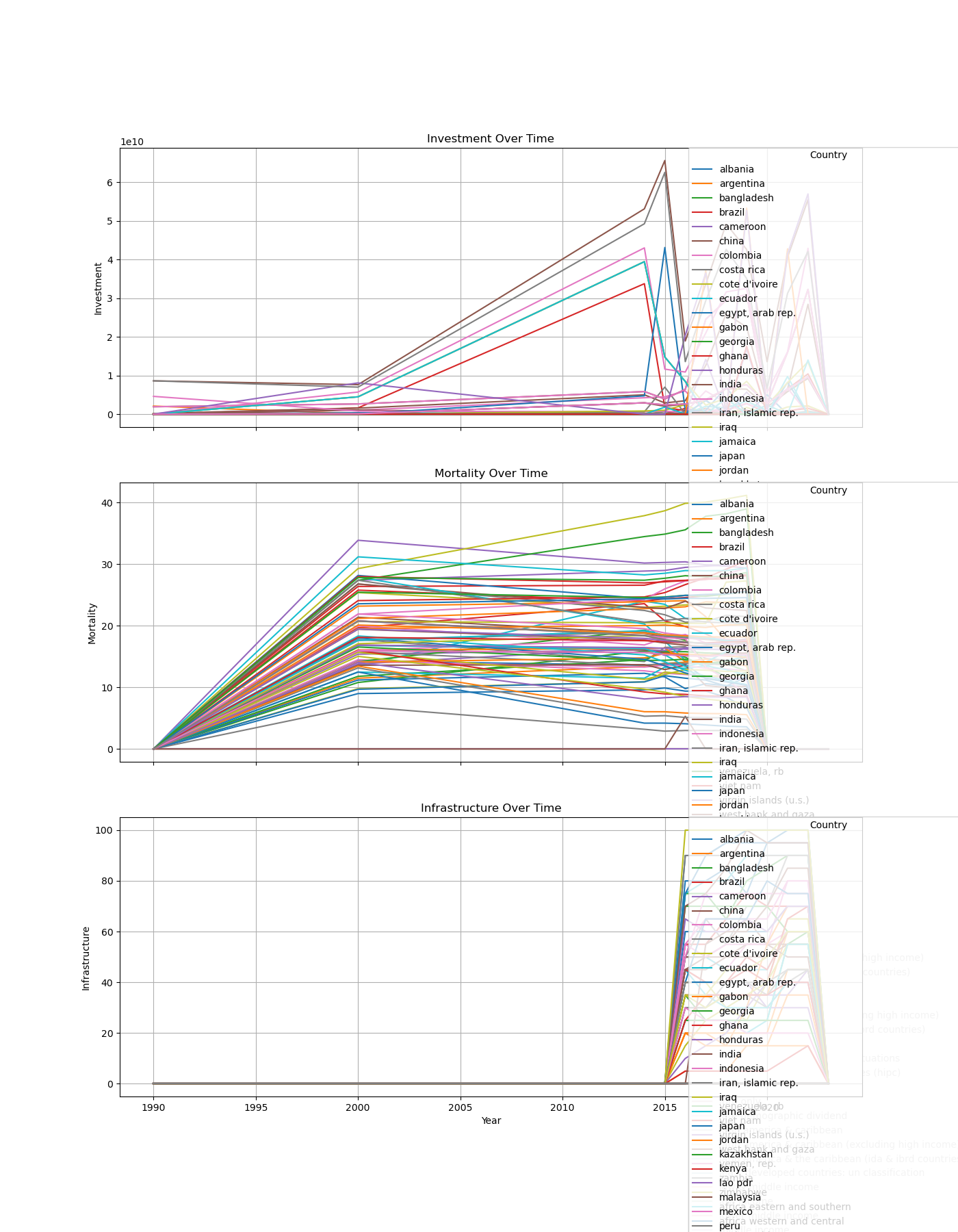
# Descriptive Statistics for Mortality

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Country | Mean | Median | Std | Min | Max |
| africa eastern and southern | 17.03 | 28.41 | 15.05 | 0.00 | 31.21 |
| africa western and central | 14.59 | 24.40 | 12.92 | 0.00 | 28.19 |
| albania | 7.47 | 11.20 | 6.67 | 0.00 | 14.30 |
| arab world | 11.64 | 19.66 | 10.28 | 0.00 | 20.20 |
| argentina | 7.88 | 12.45 | 6.99 | 0.00 | 14.10 |
| bangladesh | 8.72 | 13.05 | 7.82 | 0.00 | 15.90 |
| brazil | 11.32 | 16.55 | 10.16 | 0.00 | 23.60 |
| cameroon | 17.05 | 28.15 | 15.07 | 0.00 | 30.20 |
| caribbean small states | 8.54 | 14.07 | 7.56 | 0.00 | 16.56 |
| central europe and the baltics | 5.74 | 8.58 | 5.45 | 0.00 | 15.99 |
| china | 10.78 | 17.50 | 9.57 | 0.00 | 21.40 |
| colombia | 10.20 | 15.85 | 9.07 | 0.00 | 20.00 |
| costa rica | 9.08 | 14.30 | 8.08 | 0.00 | 17.80 |
| cote d'ivoire | 13.88 | 23.10 | 12.27 | 0.00 | 25.40 |
| early-demographic dividend | 9.55 | 16.18 | 8.44 | 0.00 | 16.85 |
| east asia & pacific (excluding high income) | 10.74 | 17.72 | 9.52 | 0.00 | 20.81 |
| east asia & pacific (ida & ibrd countries) | 10.70 | 17.64 | 9.48 | 0.00 | 20.79 |
| ecuador | 12.03 | 16.80 | 10.92 | 0.00 | 24.00 |
| egypt, arab rep. | 6.33 | 9.95 | 5.63 | 0.00 | 12.30 |
| euro area | 3.67 | 4.96 | 3.94 | 0.00 | 13.17 |
| europe & central asia | 5.49 | 7.61 | 5.22 | 0.00 | 15.05 |
| europe & central asia (excluding high income) | 7.83 | 10.78 | 7.21 | 0.00 | 18.36 |
| europe & central asia (ida & ibrd countries) | 7.60 | 10.61 | 6.99 | 0.00 | 18.04 |
| european union | 4.04 | 5.65 | 4.15 | 0.00 | 13.46 |
| fragile and conflict affected situations | 14.60 | 24.65 | 12.89 | 0.00 | 25.57 |
| gabon | 13.47 | 22.00 | 11.91 | 0.00 | 23.90 |
| georgia | 7.88 | 11.80 | 7.05 | 0.00 | 15.80 |
| ghana | 14.56 | 24.20 | 12.86 | 0.00 | 25.80 |
| heavily indebted poor countries (hipc) | 16.07 | 27.07 | 14.19 | 0.00 | 28.16 |
| high income | 5.36 | 8.23 | 4.99 | 0.00 | 14.05 |
| honduras | 9.36 | 15.70 | 8.26 | 0.00 | 16.40 |
| ibrd only | 10.08 | 16.55 | 8.93 | 0.00 | 19.45 |
| india | 9.18 | 15.25 | 8.12 | 0.00 | 16.90 |
| indonesia | 7.19 | 11.45 | 6.40 | 0.00 | 14.50 |
| iran, islamic rep. | 12.97 | 20.65 | 11.55 | 0.00 | 26.80 |
| iraq | 13.08 | 20.40 | 11.79 | 0.00 | 27.30 |
| jamaica | 7.77 | 12.00 | 6.92 | 0.00 | 15.10 |
| japan | 3.02 | 3.65 | 3.58 | 0.00 | 12.50 |
| jordan | 12.61 | 17.30 | 11.36 | 0.00 | 24.10 |
| kazakhstan | 9.17 | 13.25 | 8.33 | 0.00 | 19.30 |
| kenya | 15.97 | 26.50 | 14.11 | 0.00 | 28.30 |
| lao pdr | 9.48 | 14.50 | 8.43 | 0.00 | 17.90 |
| late-demographic dividend | 11.02 | 17.65 | 9.79 | 0.00 | 21.92 |
| latin america & caribbean | 10.48 | 17.36 | 9.26 | 0.00 | 19.00 |
| latin america & caribbean (excluding high income) | 10.05 | 16.45 | 8.89 | 0.00 | 18.43 |
| latin america & the caribbean (ida & ibrd countries) | 10.57 | 17.53 | 9.34 | 0.00 | 19.21 |
| least developed countries: un classification | 14.46 | 23.99 | 12.77 | 0.00 | 25.63 |
| low & middle income | 10.87 | 18.19 | 9.61 | 0.00 | 20.13 |
| low income | 16.32 | 27.57 | 14.41 | 0.00 | 28.66 |
| lower middle income | 10.21 | 17.25 | 9.02 | 0.00 | 18.14 |
| malaysia | 13.72 | 22.60 | 12.17 | 0.00 | 26.80 |
| mexico | 7.80 | 12.90 | 6.89 | 0.00 | 14.20 |
| middle income | 10.31 | 17.06 | 9.12 | 0.00 | 19.50 |
| peru | 8.19 | 13.60 | 7.25 | 0.00 | 15.80 |
| philippines | 6.76 | 10.35 | 6.01 | 0.00 | 12.50 |
| russian federation | 9.88 | 12.45 | 9.66 | 0.00 | 27.90 |
| south asia | 8.97 | 15.09 | 7.92 | 0.00 | 15.94 |
| south asia (ida & ibrd) | 8.97 | 15.09 | 7.92 | 0.00 | 15.94 |
| turkiye | 5.20 | 7.55 | 4.66 | 0.00 | 9.90 |
| turkmenistan | 8.77 | 13.80 | 7.79 | 0.00 | 16.40 |
| turks and caicos islands | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| tuvalu | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| uganda | 17.82 | 29.55 | 15.77 | 0.00 | 33.90 |
| ukraine | 7.02 | 10.10 | 6.34 | 0.00 | 13.70 |
| united arab emirates | 8.61 | 10.10 | 8.06 | 0.00 | 18.10 |
| united kingdom | 2.10 | 2.95 | 2.14 | 0.00 | 6.90 |
| united states | 7.51 | 11.70 | 6.70 | 0.00 | 15.70 |
| upper middle income | 10.39 | 16.64 | 9.24 | 0.00 | 20.79 |
| uruguay | 8.88 | 13.75 | 7.88 | 0.00 | 16.80 |
| uzbekistan | 6.42 | 9.95 | 5.70 | 0.00 | 11.90 |
| vanuatu | 8.76 | 14.45 | 7.74 | 0.00 | 15.50 |
| venezuela, rb | 20.62 | 30.95 | 18.43 | 0.00 | 39.00 |
| viet nam | 15.71 | 24.40 | 13.98 | 0.00 | 30.60 |
| virgin islands (u.s.) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| west bank and gaza | 0.44 | 0.00 | 1.53 | 0.00 | 5.30 |
| yemen, rep. | 15.40 | 23.10 | 13.73 | 0.00 | 29.40 |
| zambia | 12.64 | 20.50 | 11.32 | 0.00 | 27.40 |
| zimbabwe | 22.31 | 33.60 | 19.92 | 0.00 | 41.20 |

# Descriptive Statistics for Infrastructure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Country | Mean | Median | Std | Min | Max |
| africa eastern and southern | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| africa western and central | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| albania | 51.25 | 77.50 | 45.88 | 0.00 | 100.00 |
| arab world | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| argentina | 24.58 | 35.00 | 23.50 | 0.00 | 60.00 |
| bangladesh | 20.83 | 25.00 | 21.62 | 0.00 | 60.00 |
| brazil | 38.33 | 55.00 | 34.33 | 0.00 | 75.00 |
| cameroon | 21.25 | 35.00 | 18.96 | 0.00 | 45.00 |
| caribbean small states | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| central europe and the baltics | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| china | 30.83 | 47.50 | 27.62 | 0.00 | 65.00 |
| colombia | 34.58 | 47.50 | 32.37 | 0.00 | 80.00 |
| costa rica | 40.83 | 52.50 | 37.83 | 0.00 | 90.00 |
| cote d'ivoire | 22.08 | 20.00 | 24.44 | 0.00 | 65.00 |
| early-demographic dividend | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| east asia & pacific (excluding high income) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| east asia & pacific (ida & ibrd countries) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| ecuador | 29.58 | 45.00 | 26.58 | 0.00 | 60.00 |
| egypt, arab rep. | 36.67 | 60.00 | 32.57 | 0.00 | 70.00 |
| euro area | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| europe & central asia | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| europe & central asia (excluding high income) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| europe & central asia (ida & ibrd countries) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| european union | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| fragile and conflict affected situations | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| gabon | 15.83 | 20.00 | 16.63 | 0.00 | 45.00 |
| georgia | 46.67 | 70.00 | 41.74 | 0.00 | 90.00 |
| ghana | 28.33 | 40.00 | 25.79 | 0.00 | 60.00 |
| heavily indebted poor countries (hipc) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| high income | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| honduras | 20.83 | 30.00 | 18.81 | 0.00 | 40.00 |
| ibrd only | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| india | 30.83 | 50.00 | 27.29 | 0.00 | 55.00 |
| indonesia | 32.50 | 52.50 | 28.80 | 0.00 | 60.00 |
| iran, islamic rep. | 20.42 | 30.00 | 18.52 | 0.00 | 45.00 |
| iraq | 22.50 | 32.50 | 20.28 | 0.00 | 45.00 |
| jamaica | 20.83 | 30.00 | 18.93 | 0.00 | 45.00 |
| japan | 54.58 | 90.00 | 48.22 | 0.00 | 95.00 |
| jordan | 23.33 | 17.50 | 27.33 | 0.00 | 70.00 |
| kazakhstan | 39.17 | 60.00 | 34.76 | 0.00 | 70.00 |
| kenya | 28.75 | 40.00 | 27.06 | 0.00 | 70.00 |
| lao pdr | 16.25 | 12.50 | 17.21 | 0.00 | 40.00 |
| late-demographic dividend | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| latin america & caribbean | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| latin america & caribbean (excluding high income) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| latin america & the caribbean (ida & ibrd countries) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| least developed countries: un classification | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| low & middle income | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| low income | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| lower middle income | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| malaysia | 29.17 | 45.00 | 25.92 | 0.00 | 55.00 |
| mexico | 41.67 | 62.50 | 37.44 | 0.00 | 75.00 |
| middle income | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| peru | 22.08 | 30.00 | 20.05 | 0.00 | 45.00 |
| philippines | 23.75 | 32.50 | 22.27 | 0.00 | 55.00 |
| russian federation | 49.58 | 77.50 | 44.03 | 0.00 | 90.00 |
| south asia | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| south asia (ida & ibrd) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| turkiye | 55.00 | 82.50 | 49.04 | 0.00 | 100.00 |
| turkmenistan | 6.25 | 5.00 | 6.78 | 0.00 | 15.00 |
| turks and caicos islands | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| tuvalu | 4.17 | 5.00 | 4.69 | 0.00 | 15.00 |
| uganda | 37.50 | 60.00 | 33.27 | 0.00 | 70.00 |
| ukraine | 51.25 | 72.50 | 46.03 | 0.00 | 100.00 |
| united arab emirates | 39.58 | 60.00 | 35.58 | 0.00 | 80.00 |
| united kingdom | 52.50 | 90.00 | 46.34 | 0.00 | 90.00 |
| united states | 58.33 | 100.00 | 51.49 | 0.00 | 100.00 |
| upper middle income | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| uruguay | 19.17 | 22.50 | 20.32 | 0.00 | 55.00 |
| uzbekistan | 38.75 | 52.50 | 35.56 | 0.00 | 80.00 |
| vanuatu | 12.50 | 15.00 | 13.06 | 0.00 | 35.00 |
| venezuela, rb | 14.58 | 25.00 | 12.87 | 0.00 | 25.00 |
| viet nam | 20.42 | 30.00 | 18.40 | 0.00 | 40.00 |
| virgin islands (u.s.) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| west bank and gaza | 34.58 | 27.50 | 37.20 | 0.00 | 85.00 |
| yemen, rep. | 13.33 | 20.00 | 12.12 | 0.00 | 30.00 |
| zambia | 24.17 | 40.00 | 21.41 | 0.00 | 45.00 |
| zimbabwe | 27.50 | 35.00 | 25.45 | 0.00 | 60.00 |

# Time Series Analysis



CODE:

import pandas as pd

import matplotlib.pyplot as plt

# Load the data from the files

investment\_file = 'finvestment.xlsx'

mortality\_file = 'fmortality.xlsx'

infrastructure\_file = 'fspi.xlsx'

df\_investment = pd.read\_excel(investment\_file)

df\_mortality = pd.read\_excel(mortality\_file)

df\_infrastructure = pd.read\_excel(infrastructure\_file)

# Clean column names

df\_investment.columns = df\_investment.columns.str.strip()

df\_mortality.columns = df\_mortality.columns.str.strip()

df\_infrastructure.columns = df\_infrastructure.columns.str.strip()

# Drop unnecessary columns (e.g., 'Unnamed' columns)

df\_investment = df\_investment.loc[:, ~df\_investment.columns.str.contains('^Unnamed')]

df\_mortality = df\_mortality.loc[:, ~df\_mortality.columns.str.contains('^Unnamed')]

df\_infrastructure = df\_infrastructure.loc[:, ~df\_infrastructure.columns.str.contains('^Unnamed')]

# Extract year columns

year\_columns\_investment = [col for col in df\_investment.columns if col.startswith('YR')]

year\_columns\_mortality = [col for col in df\_mortality.columns if col.startswith('YR')]

year\_columns\_infrastructure = [col for col in df\_infrastructure.columns if col.startswith('YR')]

# Create long format DataFrames

df\_investment\_long = df\_investment.melt(id\_vars=['Country Name'], value\_vars=year\_columns\_investment, var\_name='Year', value\_name='Investment')

df\_mortality\_long = df\_mortality.melt(id\_vars=['Country Name'], value\_vars=year\_columns\_mortality, var\_name='Year', value\_name='Mortality')

df\_infrastructure\_long = df\_infrastructure.melt(id\_vars=['Country Name'], value\_vars=year\_columns\_infrastructure, var\_name='Year', value\_name='Infrastructure')

# Remove 'YR' prefix from 'Year' column and convert to numeric

df\_investment\_long['Year'] = df\_investment\_long['Year'].str.replace('YR', '').astype(int)

df\_mortality\_long['Year'] = df\_mortality\_long['Year'].str.replace('YR', '').astype(int)

df\_infrastructure\_long['Year'] = df\_infrastructure\_long['Year'].str.replace('YR', '').astype(int)

# Merge the long format DataFrames on 'Country Name' and 'Year'

merged\_investment\_mortality = pd.merge(df\_investment\_long, df\_mortality\_long, on=['Country Name', 'Year'])

merged\_infrastructure\_mortality = pd.merge(df\_infrastructure\_long, df\_mortality\_long, on=['Country Name', 'Year'])

# Average values per country

avg\_investment = merged\_investment\_mortality.groupby('Country Name')['Investment'].mean()

avg\_mortality\_investment = merged\_investment\_mortality.groupby('Country Name')['Mortality'].mean()

avg\_infrastructure = merged\_infrastructure\_mortality.groupby('Country Name')['Infrastructure'].mean()

avg\_mortality\_infrastructure = merged\_infrastructure\_mortality.groupby('Country Name')['Mortality'].mean()

# Combine average values into DataFrames for plotting

df\_investment\_mortality = pd.DataFrame({

'Average Investment': avg\_investment,

'Average Mortality': avg\_mortality\_investment

}).reset\_index()

df\_infrastructure\_mortality = pd.DataFrame({

'Average Infrastructure': avg\_infrastructure,

'Average Mortality': avg\_mortality\_infrastructure

}).reset\_index()

# Print description of the data used for plotting

print("Average Investment vs Mortality Rates Data:")

print(df\_investment\_mortality.describe())

print("\nAverage Infrastructure vs Mortality Rates Data:")

print(df\_infrastructure\_mortality.describe())

# Create scatter plot for Investment vs Mortality

plt.figure(figsize=(16, 8))

plt.subplot(1, 2, 1)

plt.scatter(df\_investment\_mortality['Average Investment'], df\_investment\_mortality['Average Mortality'], alpha=0.7)

plt.xlabel('Average Investment')

plt.ylabel('Average Mortality')

plt.title('Investment vs Mortality Rates')

plt.grid(True)

# Create scatter plot for Infrastructure vs Mortality

plt.subplot(1, 2, 2)

plt.scatter(df\_infrastructure\_mortality['Average Infrastructure'], df\_infrastructure\_mortality['Average Mortality'], alpha=0.7, color='orange')

plt.xlabel('Average Infrastructure')

plt.ylabel('Average Mortality')

plt.title('Infrastructure Scores vs Mortality Rates')

plt.grid(True)

# Save and show plots

plt.tight\_layout()

plt.savefig('scatter\_plots.png')

plt.show()

print("Scatter plots completed and saved to 'scatter\_plots.png'.")  
  
OUTPUT:  
  
Average Investment vs Mortality Rates Data:

Average Investment Average Mortality

count 7.800000e+01 78.000000

mean 2.582582e+09 9.883245

std 6.004069e+09 4.474323

min 0.000000e+00 0.000000

25% 0.000000e+00 7.646359

50% 6.333333e+07 9.518901

75% 1.296674e+09 12.633333

max 3.240966e+10 22.308333

Average Infrastructure vs Mortality Rates Data:

Average Infrastructure Average Mortality

count 78.000000 78.000000

mean 18.301282 9.883245

std 18.185870 4.474323

min 0.000000 0.000000

25% 0.000000 7.646359

50% 19.791667 9.518901

75% 30.833333 12.633333

max 58.333333 22.308333

CODE:  
  
import pandas as pd

import numpy as np

import statsmodels.api as sm

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import mean\_squared\_error, r2\_score

# Load the data from the files

investment\_file = 'finvestment.xlsx'

mortality\_file = 'fmortality.xlsx'

infrastructure\_file = 'fspi.xlsx'

df\_investment = pd.read\_excel(investment\_file)

df\_mortality = pd.read\_excel(mortality\_file)

df\_infrastructure = pd.read\_excel(infrastructure\_file)

# Clean column names

df\_investment.columns = df\_investment.columns.str.strip()

df\_mortality.columns = df\_mortality.columns.str.strip()

df\_infrastructure.columns = df\_infrastructure.columns.str.strip()

# Drop unnecessary columns (e.g., 'Unnamed' columns)

df\_investment = df\_investment.loc[:, ~df\_investment.columns.str.contains('^Unnamed')]

df\_mortality = df\_mortality.loc[:, ~df\_mortality.columns.str.contains('^Unnamed')]

df\_infrastructure = df\_infrastructure.loc[:, ~df\_infrastructure.columns.str.contains('^Unnamed')]

# Extract year columns

year\_columns\_investment = [col for col in df\_investment.columns if col.startswith('YR')]

year\_columns\_mortality = [col for col in df\_mortality.columns if col.startswith('YR')]

year\_columns\_infrastructure = [col for col in df\_infrastructure.columns if col.startswith('YR')]

# Create long format DataFrames

df\_investment\_long = df\_investment.melt(id\_vars=['Country Name'], value\_vars=year\_columns\_investment, var\_name='Year', value\_name='Investment')

df\_mortality\_long = df\_mortality.melt(id\_vars=['Country Name'], value\_vars=year\_columns\_mortality, var\_name='Year', value\_name='Mortality')

df\_infrastructure\_long = df\_infrastructure.melt(id\_vars=['Country Name'], value\_vars=year\_columns\_infrastructure, var\_name='Year', value\_name='Infrastructure')

# Remove 'YR' prefix from 'Year' column and convert to numeric

df\_investment\_long['Year'] = df\_investment\_long['Year'].str.replace('YR', '').astype(int)

df\_mortality\_long['Year'] = df\_mortality\_long['Year'].str.replace('YR', '').astype(int)

df\_infrastructure\_long['Year'] = df\_infrastructure\_long['Year'].str.replace('YR', '').astype(int)

# Merge the long format DataFrames on 'Country Name' and 'Year'

merged\_df = pd.merge(df\_investment\_long, df\_mortality\_long, on=['Country Name', 'Year'])

merged\_df = pd.merge(merged\_df, df\_infrastructure\_long, on=['Country Name', 'Year'])

# Pivot data to get average values per country

average\_df = merged\_df.groupby('Country Name').agg({

'Investment': 'mean',

'Mortality': 'mean',

'Infrastructure': 'mean'

}).reset\_index()

# Define features and target variable

X = average\_df[['Investment', 'Infrastructure']]

y = average\_df['Mortality']

# Add constant term for intercept

X = sm.add\_constant(X)

# Perform regression analysis using statsmodels

model = sm.OLS(y, X).fit()

print(model.summary())

# For scikit-learn

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Linear Regression Model

lin\_reg = LinearRegression()

lin\_reg.fit(X\_train, y\_train)

# Predictions

y\_pred = lin\_reg.predict(X\_test)

# Model Evaluation

print("Mean Squared Error:", mean\_squared\_error(y\_test, y\_pred))

print("R^2 Score:", r2\_score(y\_test, y\_pred))  
  
output:  
  
OLS Regression Results

==============================================================================

Dep. Variable: Mortality R-squared: 0.011

Model: OLS Adj. R-squared: -0.016

Method: Least Squares F-statistic: 0.3986

Date: Sun, 21 Jul 2024 Prob (F-statistic): 0.673

Time: 22:31:01 Log-Likelihood: -226.63

No. Observations: 78 AIC: 459.3

Df Residuals: 75 BIC: 466.3

Df Model: 2

Covariance Type: nonrobust

==================================================================================

coef std err t P>|t| [0.025 0.975]

----------------------------------------------------------------------------------

const 10.3446 0.826 12.524 0.000 8.699 11.990

Investment 9.573e-14 8.96e-11 0.001 0.999 -1.78e-10 1.79e-10

Infrastructure -0.0252 0.030 -0.852 0.397 -0.084 0.034

==============================================================================

Omnibus: 1.773 Durbin-Watson: 1.597

Prob(Omnibus): 0.412 Jarque-Bera (JB): 1.178

Skew: 0.074 Prob(JB): 0.555

Kurtosis: 3.584 Cond. No. 1.05e+10

==============================================================================

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The condition number is large, 1.05e+10. This might indicate that there are

strong multicollinearity or other numerical problems.

Mean Squared Error: 24.595701734432044

R^2 Score: -0.0027993415906517605

CODE:  
import pandas as pd

import numpy as np

from sklearn.decomposition import PCA

from sklearn.preprocessing import StandardScaler

import matplotlib.pyplot as plt

# Load the data from the files

investment\_file = 'finvestment.xlsx'

mortality\_file = 'fmortality.xlsx'

infrastructure\_file = 'fspi.xlsx'

df\_investment = pd.read\_excel(investment\_file)

df\_mortality = pd.read\_excel(mortality\_file)

df\_infrastructure = pd.read\_excel(infrastructure\_file)

# Clean column names

df\_investment.columns = df\_investment.columns.str.strip()

df\_mortality.columns = df\_mortality.columns.str.strip()

df\_infrastructure.columns = df\_infrastructure.columns.str.strip()

# Drop unnecessary columns (e.g., 'Unnamed' columns)

df\_investment = df\_investment.loc[:, ~df\_investment.columns.str.contains('^Unnamed')]

df\_mortality = df\_mortality.loc[:, ~df\_mortality.columns.str.contains('^Unnamed')]

df\_infrastructure = df\_infrastructure.loc[:, ~df\_infrastructure.columns.str.contains('^Unnamed')]

# Extract year columns

year\_columns\_investment = [col for col in df\_investment.columns if col.startswith('YR')]

year\_columns\_mortality = [col for col in df\_mortality.columns if col.startswith('YR')]

year\_columns\_infrastructure = [col for col in df\_infrastructure.columns if col.startswith('YR')]

# Create long format DataFrames

df\_investment\_long = df\_investment.melt(id\_vars=['Country Name'], value\_vars=year\_columns\_investment, var\_name='Year', value\_name='Investment')

df\_mortality\_long = df\_mortality.melt(id\_vars=['Country Name'], value\_vars=year\_columns\_mortality, var\_name='Year', value\_name='Mortality')

df\_infrastructure\_long = df\_infrastructure.melt(id\_vars=['Country Name'], value\_vars=year\_columns\_infrastructure, var\_name='Year', value\_name='Infrastructure')

# Remove 'YR' prefix from 'Year' column and convert to numeric

df\_investment\_long['Year'] = df\_investment\_long['Year'].str.replace('YR', '').astype(int)

df\_mortality\_long['Year'] = df\_mortality\_long['Year'].str.replace('YR', '').astype(int)

df\_infrastructure\_long['Year'] = df\_infrastructure\_long['Year'].str.replace('YR', '').astype(int)

# Merge the long format DataFrames on 'Country Name' and 'Year'

merged\_df = pd.merge(df\_investment\_long, df\_mortality\_long, on=['Country Name', 'Year'])

merged\_df = pd.merge(merged\_df, df\_infrastructure\_long, on=['Country Name', 'Year'])

# Pivot data to get average values per country

average\_df = merged\_df.groupby('Country Name').agg({

'Investment': 'mean',

'Mortality': 'mean',

'Infrastructure': 'mean'

}).reset\_index()

# Print statistics about the average data used for PCA

print("Average Data Used for PCA:")

print(average\_df.describe())

# Prepare data for PCA

X = average\_df[['Investment', 'Infrastructure']]

y = average\_df['Mortality']

# Standardize the data

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

# Apply PCA

pca = PCA(n\_components=2)

principal\_components = pca.fit\_transform(X\_scaled)

# Create DataFrame for PCA results

pca\_df = pd.DataFrame(data=principal\_components, columns=['Principal Component 1', 'Principal Component 2'])

pca\_df['Country Name'] = average\_df['Country Name']

pca\_df['Mortality'] = y

# Plot PCA results

plt.figure(figsize=(12, 6))

plt.scatter(pca\_df['Principal Component 1'], pca\_df['Principal Component 2'], c=pca\_df['Mortality'], cmap='viridis', s=100)

plt.colorbar(label='Mortality Rate')

plt.xlabel('Principal Component 1')

plt.ylabel('Principal Component 2')

plt.title('PCA of Transport Investment and Infrastructure Performance')

plt.grid(True)

plt.tight\_layout()

# Save the plot

plt.savefig('pca\_analysis.png')

plt.show()

# Print explained variance ratio

print("Explained Variance Ratio:", pca.explained\_variance\_ratio\_)

# Print the PCA components

print("\nPCA Components:")

print(pd.DataFrame(pca.components\_, columns=['Investment', 'Infrastructure'], index=['PC1', 'PC2']))  
  
OUTPUT:  
Average Data Used for PCA:

Investment Mortality Infrastructure

count 7.800000e+01 78.000000 78.000000

mean 2.582582e+09 9.883245 18.301282

std 6.004069e+09 4.474323 18.185870

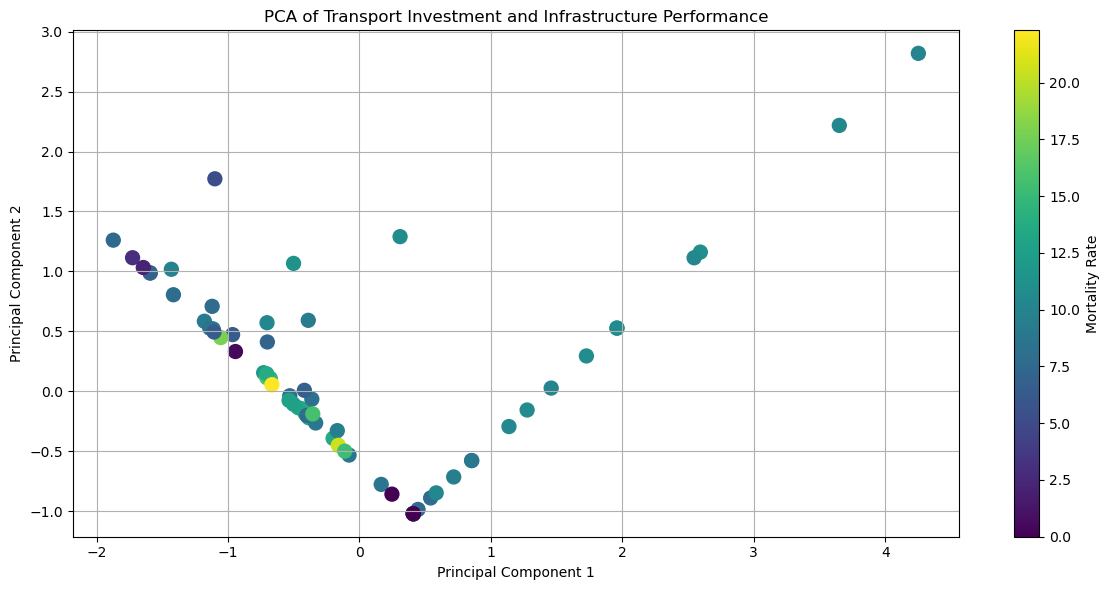
min 0.000000e+00 0.000000 0.000000

25% 0.000000e+00 7.646359 0.000000

50% 6.333333e+07 9.518901 19.791667

75% 1.296674e+09 12.633333 30.833333

max 3.240966e+10 22.308333 58.333333

Explained Variance Ratio: [0.64859045 0.35140955]

PCA Components:

Investment Infrastructure

PC1 0.707107 -0.707107

PC2 0.707107 0.707107