Name Muhammad Junaid Saif

Student Id 22030494

Assignment 2 Statistics and Trends Report

Git Hub link <https://github.com/ms22akc/Assignment2_Statistics>

**Exploring the Impacts of Climate Change Using World Bank Data**

The purpose of this analysis was to investigate the interrelations between climate change and various factors in ten selected countries (Countries names are mentioned in table 1 with their abbreviations) from different continents. The factors examined were Total greenhouse gas emissions (kt of CO2 equivalent), Population growth (annual %), Forest area (sq. km), CO2 emissions (metric tons per capita), and Arable land (% of land area).

The analysis identified certain correlations among the factors, and the underlying causes behind these correlations were investigated.

Chart, bar chart

Description automatically generated

# Figure 1: Greenhouse Gas Emissions by Countries (2000-2005)

Figure 1 illustrates greenhouse gas emissions by different countries, using data from 2000 to 2005 and grouped into six-year increments. The Arab World stands out with the highest levels of greenhouse gas emissions, displaying a significant upward trend over the years, which became more pronounced in 2005. On the other hand, Figure 2 provides a more detailed view of the downward trend in CO2 emissions in the Arab World. These findings suggest that although greenhouse gas emissions remain high, there have been some successful efforts to reduce CO2 emissions in the region. Nonetheless, targeted, and sustained actions are still necessary to combat the growing climate crisis.

Chart, bar chart, histogram

Description automatically generated

*Figure 2: CO2 Emissions by Countries (2000-2005)*

The upward trend in greenhouse gas emissions is not unique to the Arab World, as depicted in both Figure 1 and 2. Notably, Angola and Argentina also exhibited similar trends, which is noteworthy on a global scale.

Table 1 displays the annual percentage population growth of various countries between 2000 and 2005. The data reveals that the United Arab Emirates had the highest population growth rate with a mean value of 5.58% in 2000 and 6.96% in 2005. Additionally, several other countries also showed a positive population growth rate, such as Angola and Arab World. On the other hand, Albania and Armenia had negative population growth rates, indicating a decrease in population during the specified time. Overall, the table provides an overview of the annual percentage population growth for a selection of countries between 2000 and 2005.

*Table 1: Annual Percentage Population Growth by Country (2000-2005)*

|  |  |  |
| --- | --- | --- |
| ***Countries*** | ***2000*** | ***2005*** |
| Afghanistan (AFG) | 1.443803 | 3.576508 |
| Africa Eastern and Southern (AFE) | 2.583579 | 2.666242 |
| Africa Western and Central AFW() | 2.7496 | 2.841211 |
| Albania (ALB) | -0.63736 | -0.51179 |
| Andorra (AND) | 0.67096 | 3.691435 |
| Angola (ANG) | 3.244121 | 3.557659 |
| Arab World (ARB) | 2.285934 | 2.333327 |
| Argentina (ARG) | 1.133277 | 1.033476 |
| Armenia (ARM) | -1.17679 | -0.60524 |
| United Arab Emirates (ARE) | 5.580387 | 6.955727 |

Chart, line chart

Description automatically generated

Figure 3, depict the population growth trends of Afghanistan and United Arab Emirates over time, a line graph can be used, with the x-axis showing the years from 2000 to 2005 and the y-axis representing the population size or growth rate. The population growth rate for Afghanistan increased sharply in 2003 and then fluctuated before experiencing a downward trend from 2005 onwards. On the other hand, the population growth rate of the United Arab Emirates showed a gradual decline from 2000 until 2004. However, from 2004 to 2005, the population started to increase steadily, indicating a change in the trend of population growth.

Chart, treemap chart

Description automatically generated

Figure 4 shows a heatmap of the selected indicators for the Arab World from 2000 to 2005. The colours on the heatmap represent the correlation between the indicators, with yellow to orange indicating moderate positive and negative correlation, and orange to dark red indicating strong positive correlation. The intensity of the colour indicates the strength of the correlation, with stronger colours indicating higher correlation.

Chart

Description automatically generated with medium confidence

This histogram represents the distribution of values for the selected indicators in Argentina for the year 2001. The x-axis represents the range of values, divided into 10 equally spaced bins, and the y-axis represents the frequency of occurrence of values in each bin. The histogram shows that most of the values fall within the first and last two bins, with very few values in the middle bins, indicating a skewed distribution.

<https://data.worldbank.org/topic/climate-change>

**Code:**

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

def read\_data(file):

# Read the data from the file into a pandas dataframe

df = pd.read\_csv(file, header=1, skiprows=range(3))

# Add an extra index column to the dataframe

df.insert(0, 'Index', range(1, len(df)+1))

# Write the modified dataframe to a new CSV file

df.to\_csv('C:/Users/samre/Downloads/New folder (2)/New folder/climate data.csv'', index=False)

# Transpose the dataframe so that years are columns and countries are rows

df\_transposed = df.transpose()

# Assign the correct column names

df\_transposed.columns = df\_transposed.iloc[0]

df\_transposed = df\_transposed.iloc[1:]

# Drop any rows with missing data and drop the 'Indicator Code' column

df\_transposed.dropna(inplace=True)

if 'Indicator Code' in df\_transposed.columns:

df\_transposed.drop('Indicator Code', axis=1, inplace=True)

# Separate the dataframes into one with years as columns and one with countries as columns

df\_by\_years = df\_transposed.copy()

df\_by\_years.columns = pd.to\_numeric(df\_by\_years.columns, errors='coerce')

df\_by\_years.dropna(axis=1, inplace=True)

df\_by\_countries = df\_transposed.transpose()

return df\_by\_years, df\_by\_countries

# Calculate summary

def calculate\_summary(df\_by\_countries, countries, indicators, year\_cols):

# Use loc to select the rows that meet the conditions

selected\_cols = ['Country Name', 'Indicator Name'] + year\_cols

selected\_rows = df\_by\_countries.loc[(df\_by\_countries['Country Name'].isin(countries)) &

(df\_by\_countries['Indicator Name'].isin(indicators)), selected\_cols]

selected\_rows.fillna(selected\_rows.mean(), inplace=True)

# Calculate summary statistics for the selected countries and indicators

summary\_stats = selected\_rows.groupby(['Country Name', 'Indicator Name'])[year\_cols].agg(['mean', 'median', 'std'])

return summary\_stats

# Bar chart Total greenhouse gas emissions (kt of CO2 equivalent)

def plot\_bar\_chart(df, countries, indicator, year\_cols):

# Use loc to select the rows that meet the conditions

selected\_cols = ['Country Name', 'Indicator Name'] + year\_cols

selected\_rows = df.loc[(df['Country Name'].isin(countries)) &

(df['Indicator Name'] == indicator), selected\_cols]

# Fill in missing values with 0

selected\_numeric\_cols = year\_cols

selected\_rows[selected\_numeric\_cols] = selected\_rows[selected\_numeric\_cols].fillna(0)

# Transpose the selected rows to make the years the columns and the countries the rows

selected\_transposed = selected\_rows.set\_index(['Country Name', 'Indicator Name']).transpose()

# Replace the index name with 'Year'

selected\_transposed.index.name = 'Year'

# Reset the index to turn the year columns into a regular column

selected\_transposed = selected\_transposed.reset\_index()

# Melt the data to create separate columns for each country

melted\_data = pd.melt(selected\_transposed, id\_vars='Year', var\_name='Country Name', value\_name=indicator)

# Replace long country names with abbreviations

melted\_data['Country Name'] = melted\_data['Country Name'].map(country\_abbr).fillna(melted\_data['Country Name'])

# Use seaborn to create a bar plot

plt.figure(figsize=(5, 5))

sns.barplot(x='Country Name', y=indicator, hue='Year', data=melted\_data)

plt.title(indicator, fontsize='14')

plt.xlabel('Country', fontsize='14')

plt.legend(title='Year', loc='center left', bbox\_to\_anchor=(1, 0.5), fontsize='14')

plt.show()

return selected\_rows

# Bar chart CO2 emissions (metric tons per capita)

def plot\_bar\_chart1(df, countries, indicator, year\_cols):

# Use loc to select the rows that meet the conditions

selected\_cols = ['Country Name', 'Indicator Name'] + year\_cols

selected\_rows = df.loc[(df['Country Name'].isin(countries)) &

(df['Indicator Name'] == indicator), selected\_cols]

# Fill in missing values with 0

selected\_numeric\_cols = year\_cols

selected\_rows[selected\_numeric\_cols] = selected\_rows[selected\_numeric\_cols].fillna(0)

# Transpose the selected rows to make the years the columns and the countries the rows

selected\_transposed = selected\_rows.set\_index(['Country Name', 'Indicator Name']).transpose()

# Replace the index name with 'Year'

selected\_transposed.index.name = 'Year'

# Reset the index to turn the year columns into a regular column

selected\_transposed = selected\_transposed.reset\_index()

# Melt the data to create separate columns for each country

melted\_data = pd.melt(selected\_transposed, id\_vars='Year', var\_name='Country Name', value\_name=indicator)

# Replace long country names with abbreviations

melted\_data['Country Name'] = melted\_data['Country Name'].map(country\_abbr).fillna(melted\_data['Country Name'])

# Use seaborn to create a bar plot

plt.figure(figsize=(5, 5))

sns.barplot(x='Country Name', y=indicator, hue='Year', data=melted\_data)

plt.title(indicator, fontsize='14')

plt.xlabel('Country', fontsize='14')

plt.ylabel(indicator, fontsize='14')

plt.legend(title='Year', loc='center left', bbox\_to\_anchor=(1, 0.5), fontsize='14')

plt.show()

return selected\_rows

# Correlation Heatmap

def plot\_heatmap(df\_by\_countries, country, indicators, year\_cols):

# Select data for a specific country

selected\_cols = ['Indicator Name'] + year\_cols

selected\_rows = df\_by\_countries.loc[(df\_by\_countries['Country Name'] == country) &

(df\_by\_countries['Indicator Name'].isin(indicators)), selected\_cols]

# Pivot the data to create a correlation matrix

corr = selected\_rows.pivot\_table(index=None, columns='Indicator Name', values=year\_cols).corr()

# Plot a heatmap of the correlation matrix

plt.figure(figsize=(5, 5))

sns.heatmap(corr, cmap='YlOrRd', annot=True, vmin=-1, vmax=1)

plt.xlabel('Indicator Name', fontsize=14)

plt.ylabel('Indicator Name', fontsize=14)

plt.xticks(fontsize=14)

plt.yticks(fontsize=14)

plt.title(f'Correlation between selected indicators for {country} for year 2000 t0 2005', fontsize='14')

plt.show()

# Time seriesplot

def plot\_time\_series(df\_by\_countries, countries, indicator, year\_cols, legend\_size=14):

# Use loc to select the rows that meet the conditions

selected\_cols = ['Country Name', 'Indicator Name'] + year\_cols

selected\_rows = df\_by\_countries.loc[(df\_by\_countries['Country Name'].isin(countries)) &

(df\_by\_countries['Indicator Name'] == indicator), selected\_cols]

# Plot line charts to visualize trends over time for each country

plt.figure(figsize=(5, 5))

plt.title(f'Trends over time for {indicator} in selected countries')

for country in countries:

abbr = country\_abbr.get(country, country) # Use the abbreviation if available, otherwise use the country name

data = selected\_rows.loc[selected\_rows['Country Name'] == country, year\_cols].values[0]

plt.plot(year\_cols, data, label=abbr)

plt.xlabel('Year')

plt.legend(loc='center left', bbox\_to\_anchor=(1, 0.5), fontsize=14)

plt.show()

# Call the read\_data function

df\_by\_years, df\_by\_countries = read\_data('C:/Users/samre/Downloads/New folder (2)/New folder/climate data.csv'')

# Select the columns with the specified countries, indicators, and years

countries = ['Africa Eastern and Southern', 'Afghanistan', 'Africa Western and Central', 'Angola', 'Albania', 'Andorra', 'Arab World', 'United Arab Emirates', 'Argentina', 'Armenia']

indicators = ['Total greenhouse gas emissions (kt of CO2 equivalent)', 'Population growth (annual %)', 'Forest area (sq. km)', 'CO2 emissions (metric tons per capita)', 'Arable land (% of land area)']

year\_cols = ['2000', '2001', '2002', '2003', '2004', '2005']

# Define a dictionary of country name abbreviations

country\_abbr = {'Africa Eastern and Southern': 'AFE', 'Afghanistan':'AFG', 'Africa Western and Central': 'AFW',

'Angola': 'ANG', 'Albania': 'ALB', 'Andorra': 'AND', 'Arab World': 'ARB',

'United Arab Emirates': 'ARE', 'Argentina': 'ARG', 'Armenia': 'ARM'}

# Call the bar\_chart function

# Call the function for Total greenhouse gas emissions and store the selected rows in a variable

selected\_data = plot\_bar\_chart(df\_by\_countries, countries, 'Total greenhouse gas emissions (kt of CO2 equivalent)', year\_cols)

# Call the function for CO2 emissions (metric tons per capita) and store the selected rows in a variable

selected\_data = plot\_bar\_chart1(df\_by\_countries, countries, 'CO2 emissions (metric tons per capita)', year\_cols)

# Use the selected\_data variable to access the countries and year\_cols variables

print(selected\_data['Country Name'].unique()) # Output: ['Africa Eastern and Southern' 'Afghanistan' 'Africa Western and Central' 'Albania' 'Algeria' 'Andorra' 'Angola' 'Arab World' 'United Arab Emirates' 'Argentina' 'Armenia']

print(selected\_data.columns[2:]) # Output: Index(['2000', '2001', '2002', '2003', '2004', '2005'], dtype='object')

# Call the plot\_heatmap function

plot\_heatmap(df\_by\_countries, 'Arab World', ['Total greenhouse gas emissions (kt of CO2 equivalent)', 'Population growth (annual %)', 'Forest area (sq. km)', 'CO2 emissions (metric tons per capita)', 'Arable land (% of land area)'], ['2000', '2001', '2002', '2003', '2004', '2005'])

# Call the plot\_time\_series function

plot\_time\_series(df\_by\_countries, countries, 'Population growth (annual %)', year\_cols)

# Calculate summary statistics for the selected countries and indicators

summary\_stats = calculate\_summary(df\_by\_countries, countries, indicators, year\_cols)

print(summary\_stats)

# Save the summary file

summary\_stats.to\_csv('D:/New folder/summary\_stats.csv')

# Create histrogram

def plot\_histogram(df\_by\_countries, country, indicators, year\_col):

# Select data for a specific country and year

selected\_cols = ['Indicator Name'] + year\_col

selected\_rows = df\_by\_countries.loc[(df\_by\_countries['Country Name'] == country) &

(df\_by\_countries['Indicator Name'].isin(indicators)), selected\_cols]

# Melt the data to create a long format dataframe

df\_melt = selected\_rows.melt(id\_vars=['Indicator Name'], value\_vars=year\_col)

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

# Plot a histogram of the data

plt.figure(figsize=(8, 4))

sns.histplot(data=df\_melt, x='value', bins=10)

plt.title(f'Histogram of {country} for year {year\_col[0]}')

plt.xticks(fontsize=14)

plt.yticks(fontsize=14)

plt.xlabel('Values', fontsize=14)

plt.ylabel('Frequency', fontsize=14)

plt.show()

# For selected data

country = 'Argentina'

year\_col = ['2001']

indicators = ['Total greenhouse gas emissions (kt of CO2 equivalent)', 'Population growth (annual %)', 'Forest area (sq. km)', 'CO2 emissions (metric tons per capita)', 'Arable land (% of land area)']

plot\_histogram(df\_by\_countries, country, indicators, year Col)