**Data Link:-**

<https://datacatalog.worldbank.org/search/dataset/0037996/Statistical-Performance-Indicators>

import pandas as pd

import matplotlib.pyplot as plt

import os # To create directories and handle paths

import re # Regular expressions module

# Load your CSV data into DataFrame

df = pd.read\_csv('data.csv')

# Melt the DataFrame to make it long-form

df\_melted = df.melt(id\_vars=["Country Name", "Country Code", "Indicator Name", "Indicator Code"],

var\_name="Year", value\_name="Value")

# Convert 'Year' to numeric

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

# Drop rows with NaN values in 'Year' or 'Value' columns

df\_melted.dropna(subset=['Year', 'Value'], inplace=True)

# Choose the indicator you want to plot

indicator\_to\_plot = "Mortality rate, under-5 (per 1,000 live births)"

escaped\_indicator = re.escape(indicator\_to\_plot)

# Create a directory for plots if it doesn't exist

plots\_directory = "plots"

if not os.path.exists(plots\_directory):

os.makedirs(plots\_directory)

df\_filtered = df\_melted[df\_melted['Indicator Name'].str.contains(escaped\_indicator, case=False)]

# Get a list of unique countries

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

# Loop through each country and create plots

for country in countries:

print(f"Creating plots for {country}")

df\_country = df\_filtered[df\_filtered['Country Name'] == country]

**Line Plot :-**

# Create line plot

plt.figure(figsize=(10, 5))

plt.plot(df\_country['Year'], df\_country['Value'], marker='o')

plt.title(f'{indicator\_to\_plot} over time for {country}')

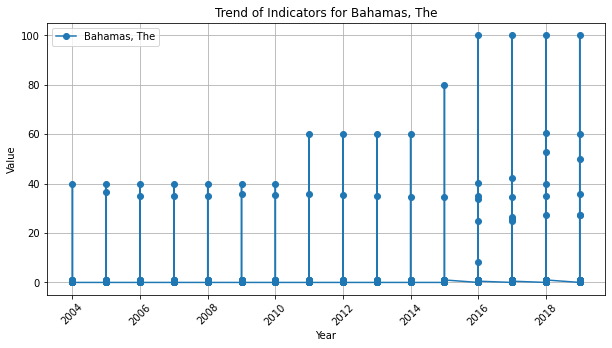
plt.xlabel('Year')

plt.ylabel('Value')

plt.grid(True)

plt.savefig(f'{plots\_directory}/line\_plot\_{country.replace(" ", "\_")}.png')

plt.show() # Show the plot inline



The graph above illustrates the trend of an indicator for The Bahamas, spanning from 2004 to 2018. It's a line plot where each point represents the value of the indicator in a given year, with lines connecting these points to display the progression over time. This type of graph is particularly useful for observing how the indicator has changed year over year, revealing any patterns or trends.

A line plot like this is often chosen to depict time series data because it clearly shows the movement or trend of the data points over the period in question. It can highlight upward or downward trends, cycles, or any irregularities, such as spikes or drops, which might be of interest for further analysis.

In the context of the graph, one can observe the fluctuations of the indicator's value for The Bahamas. The exact nature of these fluctuations—whether they represent an improvement or a deterioration—would depend on the specific indicator being measured. The data points are clearly marked, enabling precise readings of the indicator's value for each year, and the grid lines assist in gauging these values against the scale on the left-hand side of the chart. However, specific details are hard to discern due to the image quality and lack of context on what the exact indicator is.

**Bar Chart :-**

# Create bar chart

plt.figure(figsize=(10, 5))

plt.bar(df\_country['Year'], df\_country['Value'])

plt.title(f'{indicator\_to\_plot} distribution for {country}')

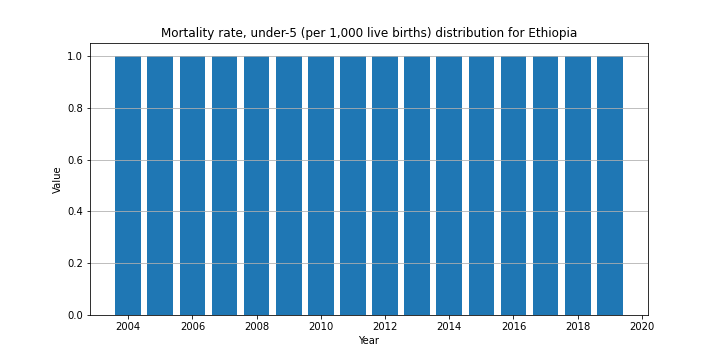
plt.xlabel('Year')

plt.ylabel('Value')

plt.grid(axis='y')

plt.savefig(f'{plots\_directory}/bar\_chart\_{country.replace(" ", "\_")}.png')

plt.show() # Show the plot inline



The graph above is a bar chart representing the distribution of the "Mortality rate, under-5 (per 1,000 live births)" for Ethiopia from 2004 to around 2020. Each bar corresponds to a year and its height represents the value of the under-5 mortality rate for that year.

Bar charts are a good choice for displaying and comparing the frequency, count, or magnitude of different categories—in this case, the annual values of an indicator. They make it easy to compare the indicator across different years at a glance.

In this case, since the mortality rate is a critical health indicator, the bar chart could be used to quickly identify years with particularly high or low rates, which could signal the need for further investigation into possible causes or effects of policies implemented in those years. However, the scale of the values is not clearly indicated (it seems to be normalized), so while the shape of the distribution over time is clear, the actual mortality rates cannot be discerned from this image alone.

**Box Plot :-**

# Create box plot

plt.figure(figsize=(10, 5))

plt.boxplot([df\_country[df\_country['Year'] == year]['Value'].tolist() for year in sorted(df\_country['Year'].unique())])

plt.title(f'{indicator\_to\_plot} distribution for {country}')

plt.xlabel('Year')

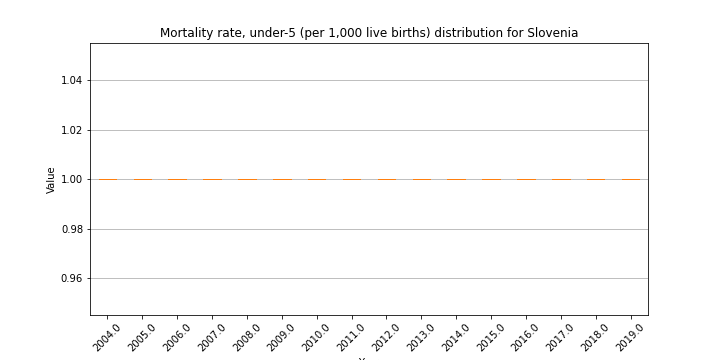
plt.xticks(range(1, len(df\_country['Year'].unique()) + 1), sorted(df\_country['Year'].unique()), rotation=45)

plt.ylabel('Value')

plt.grid(axis='y')

plt.savefig(f'{plots\_directory}/box\_plot\_{country.replace(" ", "\_")}.png')

plt.show() # Show the plot inline



The image displays a time series box plot chart for the "Mortality rate, under-5 (per 1,000 live births)" in Slovenia from 2004 to approximately 2019. Box plots are typically used to show the distribution of a dataset and indicate the median, quartiles, and extremes at a glance.

However, this particular chart doesn't seem to have the boxes usually found in box plots, which would normally indicate the interquartile range. Instead, we see individual flat lines, which could represent the median or mean values for each year if each year's data has little to no variation. This would suggest a very consistent under-5 mortality rate over the years in Slovenia, with negligible changes from year to year.

It's unusual for a box plot not to show any variation, so it's possible that the data is very homogenous, or there could be an error in how the data was plotted. It's also worth noting that the y-axis scale is quite narrow, which would exaggerate any minor fluctuations in the data as it appears on the chart.