Statistical Analysis of Olympic Medal Counts

Submitted By:

Gopi Charan

Roll Number: MA23BTECH11015

Varshith

Roll Number: MA23BTECH11014

Prajith

Roll Number: MA23BTECH11019

Course:

Applied Statistics

Table of Contents

List of Figures **II**

List of Tables **III**

1 Introduction **1**

2 Dataset and Preprocessing **1**

2.1 Data Sources 1

2.2 Snapshot of the Dataset 2

3 Descriptive Statistics **2**

3.1 Central Tendency Measures 2

3.2 Dispersion Measures 3

4 Visualizations **4**

4.1 Bar Diagram: Medals per country 4

4.2 Ogive: Cumulative Medal Distribution 5

4.3 Histogram: Medal Frequency Distribution 7

4.4 Pie Chart: Percentage Contribution of Each Country 8

4.5 Box Plot: Identifying Outliers and Spread 10

5 Application of Central Limit Theorem (CLT) **11**

5.1 Python Implementation 12

5.2 Density Plot of Sample Means 13

5.3 Explanation of CLT 13

6 Relationship Between Other Factors and Medal Counts **14**

6.1 Correlation Analysis 14

6.2 Hypothesis (Our Prediction) 15

6.3 Actual Correlation Values 15

6.4 Scatter Plots 16

6.5 Insights 19

List of Figures

1 Bar Diagram of Medals per Country 5
2 Ogive: Cumulative Medal Distribution 7
3 Histogram of Medal Frequency Distribution 8
4 Pie Chart of Percentage Contribution to Total Medals 10
5 Box Plot of Medal Counts 11
6 Density plot showing the distribution of sample means. 14
7 Scatter Plot for Medals vs GDP 16
8 Scatter Plot for Medals vs Population 17
9 Scatter Plot for Medals vs HDI 17

List of Tables

1 First five rows of the Olympic Medal Dataset 2
2 Comparison of Expected vs. Actual Correlations 16

1 Introduction

The Olympic Games have a rich history, showcasing athletic excellence across various sports. A crucial aspect of analyzing the performance of different countries is the number of medals won in various Olympic events. This dataset contains information on Olympic medal counts by country in Paris 2024 olympics.

The primary objective of this study is to analyze the Olympic medal count data using descriptive statistics, visualizations, and the Central Limit Theorem (CLT). The analysis is structured as follows:

- Compute and interpret descriptive statistics, including measures of central tendency (mean, median) and dispersion (variance, standard deviation, interquartile range).
- Visualize the dataset using bar diagrams, histograms, ogives, pie charts, and box plots to gain insights into the distribution and trends.
- Apply the Central Limit Theorem (CLT) by taking random samples of size 30 or more, computing sample means, and analyzing the resulting probability distribution.

Through this statistical exploration, we aim to uncover patterns in Olympic performance and demonstrate key statistical concepts in action.

2 Dataset and Preprocessing

2.1 Data Sources

The dataset used in this analysis comprises Olympic medal counts, along with supplementary data such as world population, Human Development Index (HDI), and Gross Domestic Product (GDP) of each country. The data has been sourced from the following:

- Olympic Medal Counts: Official olympics website
- World Population: Worldometer
- Human Development Index: HDI of all countries
- GDP Data: GDP by country

2.2 Snapshot of the Dataset

Below is a preview of the first few rows of the dataset:

```
import pandas as pd

df = pd.read_csv("olympic_data.csv")
print(df.head())
```

Country	Gold	Silver	Bronze	Total Medals	Population	GDP	HDI
United States	40	44	42	126	345.3	27720.709	0.927
China	40	27	24	91	1419.32	17794.78	0.788
Japan	20	12	13	45	123.75	4204.494	0.92
Australia	18	19	16	53	26.71	1728.057	0.946
France	16	26	22	64	66.55	3051.832	0.91

Table 1: First five rows of the Olympic Medal Dataset

3 Descriptive Statistics

In this section, we analyze the central tendency and dispersion of Olympic medal counts using descriptive statistics.

3.1 Central Tendency Measures

The central tendency of the dataset is analyzed using the mean, median, and mode.

- **Mean** (μ): Represents the average number of medals won by a country.
- **Median**: The middle value, which helps in understanding the distribution skewness.
- **Mode**: The most frequently occurring number of medals.

```
import pandas as pd
import numpy as np

df = pd.read_csv("olympics_data.csv")
mean_medals = df["Total Medals"].mean()
median_medals = df["Total Medals"].median()
```

```
mode_medals = df["Total Medals"].mode()[0]

print("Mean:", mean_medals)
print("Median:", median_medals)
print("Mode:", mode_medals)
```

Output:

```
Mean: 11.6179
Median: 5.0
Mode: 1
```

Here are some inferences based on above stats:

Mean (11.6179) is much higher than the Median (5.0): This suggests that the data is right-skewed (positively skewed), meaning some higher values are pulling the mean upwards.

Median (5.0) is the middle value: Although the maximum value is 126, half of the data points are below 5. This suggest there are less countries which got more medals. The fact that the median is much lower than the mean confirms the presence of high outliers.

Mode (1) is the most frequently occurring value: This suggests that 1 appears more often than any other number in the dataset. As the mode is lower than the median and mean, it indicates a large concentration of small values.

3.2 Dispersion Measures

We analyze variance, standard deviation, range, and interquartile range (IQR) to understand the spread of the data.

```
variance = df["Total Medals"].var()
std_dev = df["Total Medals"].std()
range_medals = df["Total Medals"].max() - df["Total Medals"].min()

# Calcalute quartiles
quartile_25 = df["Total Medals"].quantile(0.25)
quartile_50 = df["Total Medals"].quantile(0.50)
quartile_75 = df["Total Medals"].quantile(0.75)

iqr = quartile_75 - quartile_25
print("Variance:", variance)
```

```
print("Standard Deviation:", std_dev)
print("Range:", range_medals)
print("Quartile 25:", quartile_25)
print("Quartile 50:", quartile_50)
print("Quartile 75:", quartile_75)
print("Interquartile Range:", iqr)
```

Output:

```
Variance: 395.125
Standard Deviation: 19.877
Range: 125
Quartile 25: 2.0
Quartile 50: 5.0
Quartile 75: 9.0
IQR: 7
```

Inferences:

High variance indicates that data points are widely spread out. IQR = 7 indicates that middle half values of data is concentrated in 7 units only. High range + large standard deviation + small IQR → Suggests that a few very large values (outliers) are inflating the mean and variance.

4 Visualizations

4.1 Bar Diagram: Medals per country

The bar diagram provides a visual comparison of the total number of medals won by each country, effectively highlighting the most successful Olympic nations. The code snippet below demonstrates how to create this plot using Python's 'matplotlib' and 'seaborn' libraries.

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

# Load dataset
df = pd.read_csv("olympics_data.csv")

# Set figure size
```

```
plt.figure(figsize=(12, 8))

# Create bar plot
sns.barplot(x=df["Country"], y=df["Total Medals"])

# Rotate x-axis labels for better readability
plt.xticks(rotation=90)

# Adjust layout for better fit
plt.tight_layout()

plt.show()
```

Country names are listed on the x-axis. The United States, the United Kingdom, Germany, and France are among the top medal-winning countries, consistently dominating the medal tables. This hierarchy reflects their superior performance metrics. The figure highlights a clear hierarchy among countries, with a few nations standing out as medal leaders due to their consistent high performance.

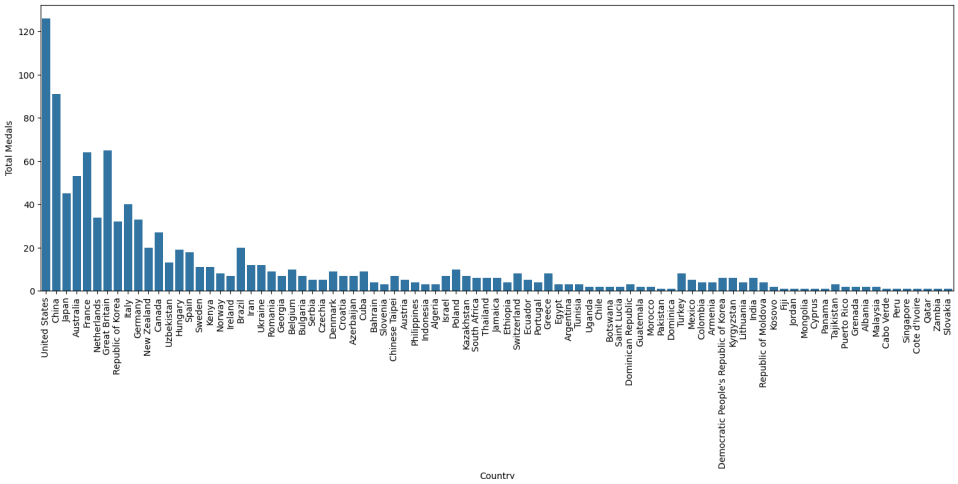

Figure 1: Bar Diagram of Medals per Country

4.2 Ogive: Cumulative Medal Distribution

The ogive plot illustrates the cumulative frequency distribution of the total medal counts between countries, highlighting how the medals accumulate over countries. The code below demonstrates how to create an ogive using Python.

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Load dataset
df = pd.read_csv("olympics_data.csv")

# Compute cumulative frequency
df["Cumulative"] = df["Total Medals"].cumsum()

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

# Plot cumulative distribution (Ogive)
plt.plot(df["Country"], df["Cumulative"], marker="o", linestyle="-", color="red")

# Set labels and title
plt.xlabel("Country")
plt.ylabel("Cumulative Medals")
plt.title("Ogive: Cumulative Medal Distribution")

# Rotate x-axis labels for better readability
plt.xticks(rotation=90)
plt.tight_layout()

plt.show()
```

The ogive curve's steep slope at the beginning indicates that a few countries, like the United States and China, contribute significantly to the cumulative medal count. This highlights the concentration of Olympic success among dominant nations. The figure illustrates how quickly the cumulative total grows as these leading countries are considered, demonstrating a nonlinear relationship between country position and cumulative medals.

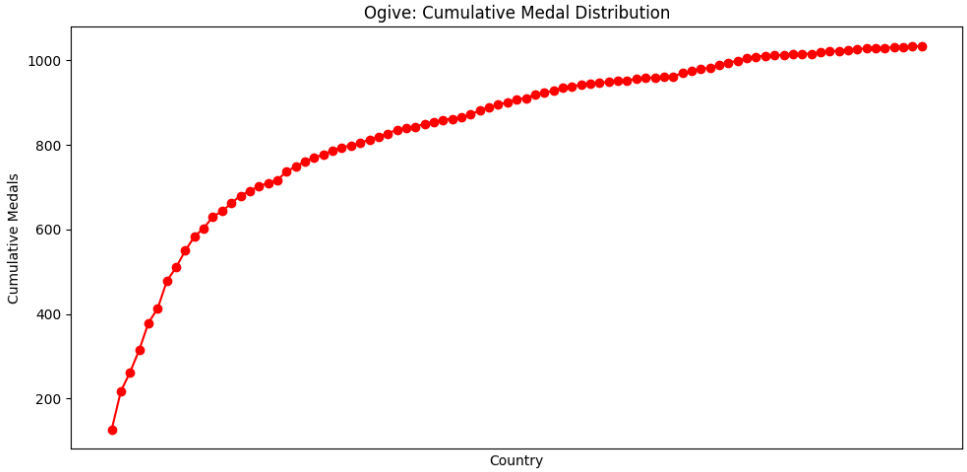

Figure 2: Ogive: Cumulative Medal Distribution

4.3 Histogram: Medal Frequency Distribution

The histogram shows the medal distribution pattern. The code snippet below shows how to create a histogram using Python.

```
import matplotlib.pyplot as plt
import pandas as pd

# Load dataset
df = pd.read_csv("olympics_data.csv")

# Define histogram parameters
min_value = 0
max_value = 126
class_length = 10
bins = list(range(min_value, max_value + class_length, class_length))

# Set figure size
plt.figure(figsize=(10, 6))

# Plot histogram
plt.hist(df["Total Medals"], bins=bins, edgecolor="black", alpha=0.7, color="blue")

# Set labels and title
```

```
plt.xlabel("Total Medals")
plt.ylabel("Frequency")
plt.title("Histogram of Total Medals")

# Set x-axis ticks for better readability
plt.xticks(bins)
plt.show()
```

Most countries have few medals, while a few, like the United States and China, accumulate many, creating a long tail on the right side. This skewness highlights the disparity in Olympic success between nations. The figure reveals a right-skewed distribution, indicating that most countries fall into the lower end of the medal count spectrum, while a few countries significantly outperform the rest.

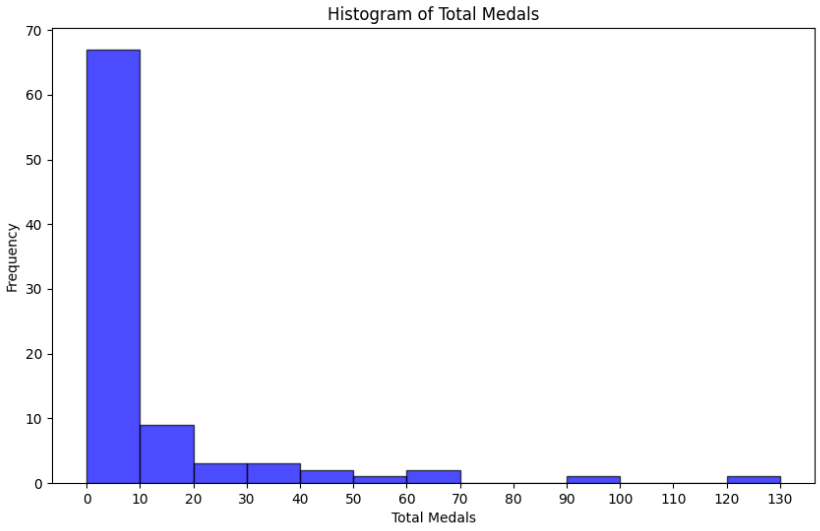

Figure 3: Histogram of Medal Frequency Distribution

4.4 Pie Chart: Percentage Contribution of Each Country

The pie chart illustrates the proportional contribution of each country to the total number of medals. The code below demonstrates how to create a pie chart using Python.

```
import matplotlib.pyplot as plt
import pandas as pd
```

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

# Sort by total medals in descending order
df = df.sort_values(by="Total Medals", ascending=False)

# Keep top 15 countries, group others into "Others"
top_n = 15
df_top = df[:top_n].copy()
df_others = pd.DataFrame({"Country": ["Others"],
"Total Medals": [df[top_n:]["Total Medals"].sum()]})
df_final = pd.concat([df_top, df_others])

# Pie chart
plt.figure(figsize=(10, 6))
plt.pie(df_final["Total Medals"], labels=df_final["Country"],
autopct="%1.1f%%", startangle=140, colors=plt.cm.Paired.colors,
wedgeprops={"edgecolor": "black"})

# Title
plt.title("Percentage Contribution of Top 15 Countries to Total Medals")

# Show plot
plt.show()
```

The pie chart shows that the top 15 countries, including the United States, China, and France, dominate the medal tally. The United States typically has the largest sector, reflecting its dominance. The figure highlights how a small group of countries, particularly the United States and China, contribute significantly to the overall medal count, illustrating a Pareto-like distribution where a few countries account for a large proportion of the total medals.

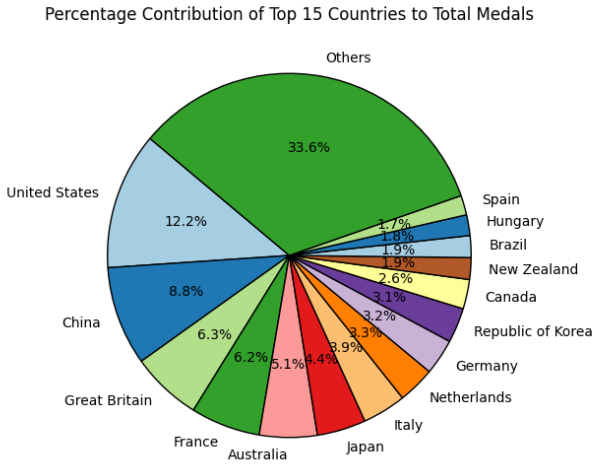

Figure 4: Pie Chart of Percentage Contribution to Total Medals

4.5 Box Plot: Identifying Outliers and Spread

The box plot visualizes the distribution of medal counts among countries, highlighting outliers and variability. The code below shows how to create a box plot using Python.

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

df = pd.read_csv('olympics_data.csv')

plt.figure(figsize=(8,5))
sns.boxplot(data=df["Total Medals"])
plt.title("Box Plot of Olympic Medal Counts")
plt.show()
```

The host nation Japan, and Countries such as the United States and China are identified as outliers due to their exceptionally high medal counts compared to the median. The interquartile range (IQR) provides additional insight into the variability of medal counts, with longer whiskers indicating more variability. The figure illustrates how these outliers significantly exceed the median,

highlighting their exceptional performance and contributing to the overall variance in medal counts.

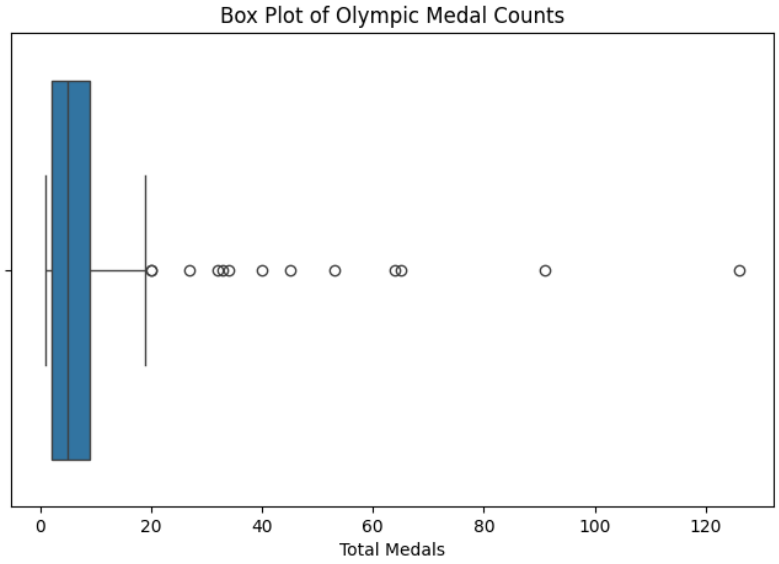

Figure 5: Box Plot of Medal Counts

5 Application of Central Limit Theorem (CLT)

The Central Limit Theorem (CLT) states that the distribution of the sample mean approaches a normal distribution as the sample size increases, regardless of the original population distribution. In this section, we demonstrate CLT by:

- Taking a random sample of size 30 or more from the dataset.
- Computing sample means multiple times (e.g., 1000 times).
- Plotting the density curve of the sample means.
- Observing how the distribution approximates a normal distribution.

Let $X_1, X_2, \ldots, X_n$ be a sequence of independent and identically distributed (i.i.d.) random variables, each having mean μ and variance σ^2. Then, the distribution of the sample mean:

$$X = X_1 + X_2 + \cdots + X_n$$

follows an approximately normal distribution as the sample size n increases. Specifically, CLT states that:

$$X \sim N\left(n\mu, n\sigma^2\right)$$

Now let $\bar{X}$ denote the random variable for sample mean.

$$\bar{X} = \frac{X_1 + X_2 + \cdots + X_n}{n}$$

Using properties of mean,variance and CLT,

$$\bar{X} \sim N\left(\mu, \frac{\sigma^2}{n}\right)$$

We shall verify this result using code.

5.1 Python Implementation

Below is the Python code used to illustrate CLT:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load dataset
df = pd.read_csv("olympics_data.csv")

# Define sample size and number of simulations
sample_size = 2000
num_simulations = 3000

# Store sample means
sample_means = []

# Perform simulations
for _ in range(num_simulations):
    sample = np.random.choice(df["Total Medals"], size=sample_size, replace=True)
    sample_means.append(np.mean(sample))

# Convert to Pandas Series
```

```
sample_means = pd.Series(sample_means)

# Plot density of sample means
plt.figure(figsize=(10, 6))
plt.hist(sample_means, bins=30, density=True, edgecolor='black', alpha=0.7)

# Overlay normal distribution curve
mu, sigma = 11.6179 , 19.877/np.sqrt(sample_size)
x = np.linspace(min(sample_means), max(sample_means), 100)
plt.plot(x, (1 / (sigma * np.sqrt(2 * np.pi))) * np.exp(-0.5 * ((x - mu) / sigma) **

# Labels and title
plt.xlabel("Sample Mean of Total Medals")
plt.ylabel("Density")
plt.title("CLT Approximation: Sample Means Distribution")
plt.legend()

# Show plot
plt.show()
```

5.2 Density Plot of Sample Means

5.3 Explanation of CLT

As observed in Figure 6, the histogram of sample means closely follows a normal distribution. This confirms the Central Limit Theorem, which states that:

- Even if the original data distribution is skewed or non-normal, the sample mean distribution will be approximately normal.
- The approximation improves as the sample size increases.
- The standard deviation of the sample mean decreases as the sample size increases, following the formula:

$$\sigma_{\bar{X}} = \frac{\sigma}{\sqrt{n}}$$

where σ is the population standard deviation, and n is the sample size.

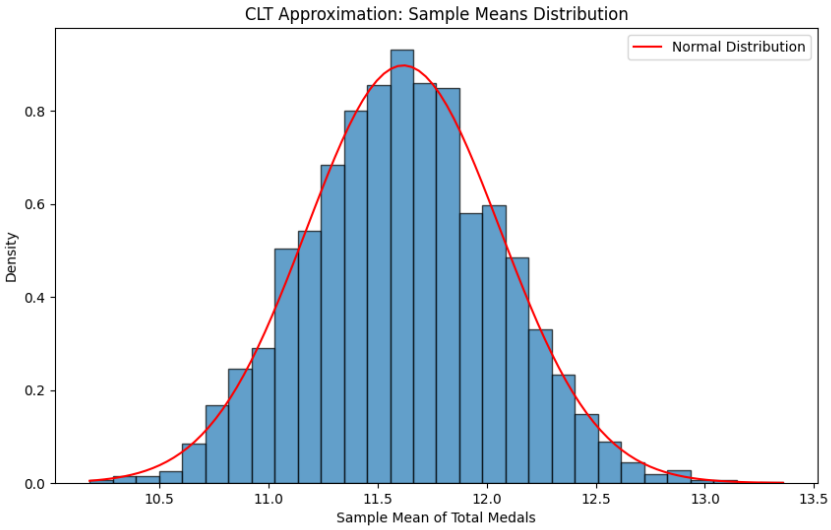

Figure 6: Density plot showing the distribution of sample means.

6 Relationship Between Other Factors and Medal Counts

In this section, we explore how different factors such as Gross Domestic Product (GDP), Population, and Human Development Index (HDI) are related to the number of Olympic medals won by each country. Understanding these relationships helps identify potential economic and social factors that contribute to a country's Olympic success.

6.1 Correlation Analysis

To measure the strength and direction of the relationships between medal counts and various factors, we compute the Pearson correlation coefficient:

$$r = \frac{\sum(X - \bar{X})(Y - \bar{Y})}{\sqrt{\sum(X - \bar{X})^2 \sum(Y - \bar{Y})^2}}$$

where X represents the independent variable (e.g., GDP, population, HDI) and Y represents the number of medals.
Here is the python implementation to find correlation:

```
import pandas as pd

# Load the data
```

```
df = pd.read_csv('olympics_data.csv')

# Find correlation between the 'Total Medals' and 'Population'
correlation_1 = df['Total Medals'].corr(df['Population'])

# Find correlation between the 'Total Medals' and 'GDP'
correlation_2 = df['Total Medals'].corr(df['GDP'])

# Find correlation between the 'Total Medals' and 'HDI'
correlation_3 = df['Total Medals'].corr(df['HDI'])

print("correlation_1:", correlation_1)
print("correlation_2:", correlation_2)
print("correlation_3:", correlation_3)
```

Output:

```
correlation_1: 0.391665
correlation_2: 0.859685
correlation_3: 0.349601
```

6.2 Hypothesis (Our Prediction)

Before computing correlations, we can predict the relationships based on intuition and historical trends:

- **GDP & Medals:** We expect a **strong positive correlation** ($r \approx 0.7 - 0.9$). Countries with higher GDPs typically have better sports infrastructure, training facilities, and athlete support.
- **Population & Medals:** We expect a **moderate correlation** ($r \approx 0.3 - 0.5$). A larger population increases the talent pool, but not all populous countries excel in the Olympics.
- **HDI & Medals:** We expect a **moderate to strong correlation** ($r \approx 0.5 - 0.8$). Countries with high HDI (better education, healthcare, and standard of living) are more likely to invest in sports development.

6.3 Actual Correlation Values

After computing correlations from the dataset, we obtained the following results:

Predictor	Expected Correlation	Actual Correlation
GDP & Medals	$0.7 - 0.9$	**0.8597** (Matches expectation)
Population & Medals	$0.3 - 0.5$	**0.3917** (Matches expectation)
HDI & Medals	$0.5 - 0.8$	**0.3496** (Weaker than expected)

Table 2: Comparison of Expected vs. Actual Correlations

6.4 Scatter Plots

To visually analyze these relationships, we plot scatter diagrams for each factor against the total number of medals won.

- **GDP vs. Medals:**

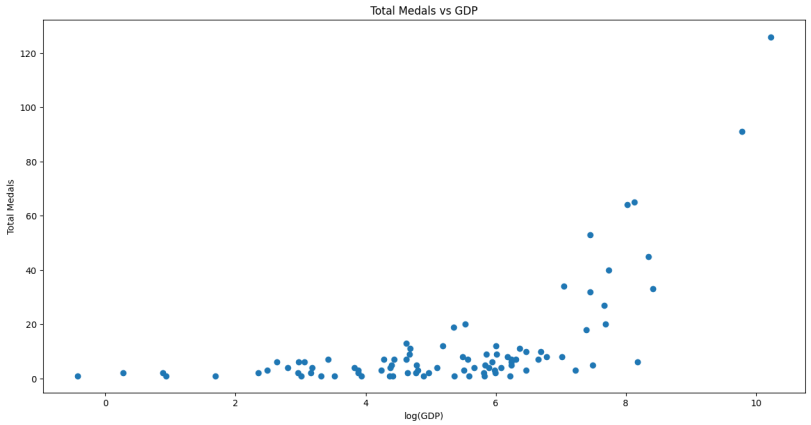

Figure 7: Scatter Plot for Medals vs GDP

  - The scatter plot suggests a positive correlation between GDP and total medal count. As GDP increases, the total number of medals tends to rise, which aligns with expectations.
  - The relationship appears to be non-linear; lower GDP countries have fewer medals, but beyond a certain GDP threshold, the medal count increases at a faster rate.

- **Population vs. Medals:**

  - The scatter plot suggests that there is a weak positive correlation between population size and total medals won. Countries with

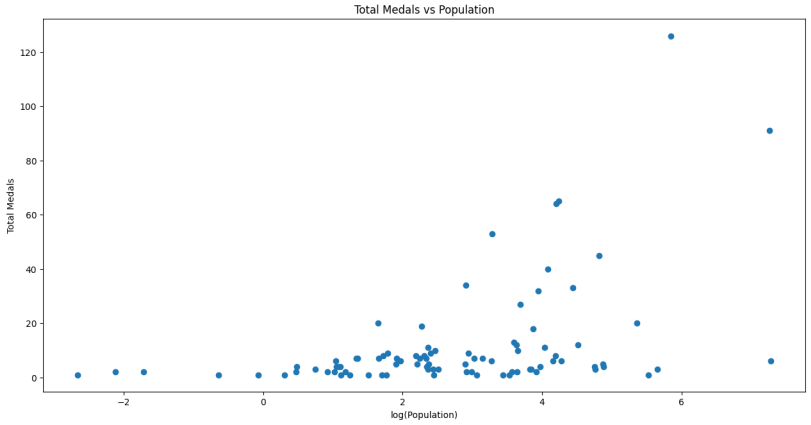

Figure 8: Scatter Plot for Medals vs Population

larger populations tend to win more medals, but this relationship is not very strong.

– Countries with moderate populations have similar medal counts to those with very large populations.This suggests that beyond a certain population size, having more people does not necessarily result in significantly more medals.

- **HDI vs. Medals:**

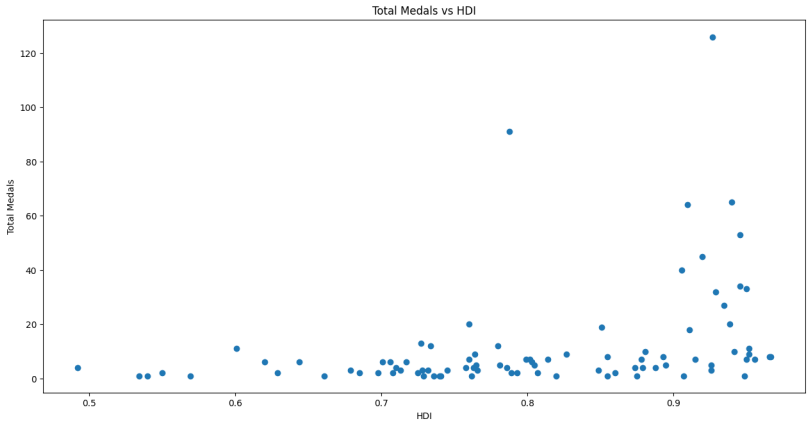

Figure 9: Scatter Plot for Medals vs HDI

– The scatter plot indicates a moderate positive correlation between

Human Development Index (HDI) and the total number of medals won.

- Countries with HDI above 0.85 dominate the medal count. Some countries with moderate HDI values (0.7-0.8) have won a notable number of medals, possibly due to strong sports culture or government investments.

Python Implementation:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the data
df = pd.read_csv('olympics_data.csv')

# Create a scatter plot for 'Total Medals' vs 'GDP'
# Scale GDP for better visualization
df['GDP'] = np.log(df['GDP'])
plt.scatter( df['GDP'],df['Total Medals'])
plt.ylabel('Total Medals')
plt.xlabel('log(GDP)')
plt.title('Total Medals vs GDP')
plt.show()

# Create a scatter plot for 'Total Medals' vs 'Population'
# Scale Population for better visualization
df['Population'] = np.log(df['Population'])
plt.scatter(df['Population'],df['Total Medals'])
plt.ylabel('Total Medals')
plt.xlabel('log(Population)')
plt.title('Total Medals vs Population')
plt.show()

# Scatter plot for medals vs HDI
plt.scatter(df['HDI'], df['Total Medals'])
plt.xlabel('HDI')
plt.ylabel('Total Medals')
plt.title('Total Medals vs HDI')
```

```
plt.show()
```

6.5 Insights

- **GDP is the strongest predictor** of Olympic success, confirming that economic power significantly impacts sports performance.
- **HDI has a weaker correlation than expected**, suggesting that economic investment in sports may matter more than general human development.
- **Population has a moderate correlation**, reinforcing the idea that wealth, not just size, is a key driver of Olympic performance.

The results suggest that economic power, population, and human development all play significant roles in a country's Olympic performance. However, outliers exist, where smaller nations with specialized training programs outperform expectations.