**✅ Cell 1**

python

CopyEdit

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

* **Problem Step**: Import all required Python Libraries.
* **Definition**:
  + import pandas as pd: Imports **pandas** for working with structured data using DataFrames.
  + import seaborn as sns: Imports **seaborn**, a library based on matplotlib, for statistical data visualization.
  + import matplotlib.pyplot as plt: Imports **matplotlib**, used for creating static visualizations.
* **Why**: These libraries are essential for data wrangling (pandas) and creating both basic and advanced plots (matplotlib, seaborn) to interpret the dataset.

**✅ Cell 2**

python

CopyEdit

df = pd.read\_csv("StudentsPerformance.csv")

df.head()

* **Problem Step**: Load the dataset and view the first few rows.
* **Definition**:
  + pd.read\_csv(): Loads data from a CSV file into a pandas DataFrame.
  + df.head(): Displays the first 5 rows of the dataset.
* **Why**: To inspect the structure and contents of the dataset to understand what kind of preprocessing or analysis is needed.

**✅ Cell 3**

python

CopyEdit

df.info()

* **Problem Step**: Inspect the data types and null values in the dataset.
* **Definition**:
  + .info(): Displays column names, number of non-null values, and data types.
* **Why**: Helps verify data types and check for missing values that may need handling.

**✅ Cell 4**

python

CopyEdit

df.describe()

* **Problem Step**: Generate summary statistics for numeric columns.
* **Definition**:
  + .describe(): Provides metrics such as mean, standard deviation, min, max, and quartiles.
* **Why**: Gives an overview of the distribution and spread of numeric scores (e.g., in math, reading, writing).

**✅ Cell 5**

python

CopyEdit

df.isnull().sum()

* **Problem Step**: Check for missing values.
* **Definition**:
  + .isnull() identifies null values.
  + .sum() totals the number of missing values per column.
* **Why**: To confirm data completeness; missing data would require cleaning.

**✅ Cell 6**

python

CopyEdit

df.duplicated().sum()

* **Problem Step**: Identify duplicate rows in the dataset.
* **Definition**:
  + .duplicated() returns a Boolean Series indicating duplicate rows.
  + .sum() counts them.
* **Why**: Duplicate entries can distort analysis and must be addressed.

**✅ Cell 7**

python

CopyEdit

df['total score'] = df['math score'] + df['reading score'] + df['writing score']

df['average score'] = df['total score']/3

df.head()

* **Problem Step**: Create new columns for total and average scores.
* **Definition**:
  + Adds individual scores to compute a total score, then divides by 3 to get the average score.
* **Why**: Simplifies performance comparison across students using a single score metric.

**✅ Cell 8**

python

CopyEdit

sns.histplot(df['average score'], kde=True, bins=20)

plt.title("Distribution of Average Scores")

plt.xlabel("Average Score")

plt.ylabel("Frequency")

plt.show()

* **Problem Step**: Plot the distribution of average scores.
* **Definition**:
  + sns.histplot(): Plots a histogram with an optional KDE curve.
  + kde=True: Adds a density line showing score distribution.
  + bins=20: Sets number of histogram intervals.
* **Why**: Helps visualize how average scores are spread — normal, skewed, or multimodal.

**✅ Cell 9**

python

CopyEdit

sns.boxplot(x='gender', y='average score', data=df)

plt.title("Average Score Distribution by Gender")

plt.show()

* **Problem Step**: Visualize how average scores vary by gender.
* **Definition**:
  + sns.boxplot(): Plots a box-and-whisker diagram, showing median, quartiles, and outliers.
* **Why**: Highlights performance differences between genders using descriptive statistics.

**✅ Cell 10**

python

CopyEdit

sns.violinplot(x='race/ethnicity', y='average score', data=df)

plt.title("Average Score Distribution by Race/Ethnicity")

plt.xticks(rotation=45)

plt.show()

* **Problem Step**: Analyze score distribution across race/ethnicity.
* **Definition**:
  + sns.violinplot(): Combines boxplot with KDE to show both summary stats and distribution shape.
* **Why**: More informative than boxplots alone for understanding variance within each group.

**✅ Cell 11**

python

CopyEdit

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

sns.barplot(x='parental level of education', y='average score', data=df)

plt.title("Average Score by Parental Level of Education")

plt.xticks(rotation=45)

plt.show()

* **Problem Step**: Investigate how parental education level influences student scores.
* **Definition**:
  + sns.barplot(): Plots mean values for categories with confidence intervals.
  + plt.figure(figsize=(10,6)): Enlarges the plot for readability.
* **Why**: Visualizes trends linking parents’ education to student performance.

**✅ Cell 12**

python

CopyEdit

sns.heatmap(df.corr(), annot=True, cmap='coolwarm')

plt.title("Correlation Matrix of Scores")

plt.show()

* **Problem Step**: Visualize correlation between all numeric features.
* **Definition**:
  + df.corr(): Computes correlation matrix.
  + sns.heatmap(): Displays matrix as color-coded plot.
  + annot=True: Shows correlation values inside heatmap cells.
* **Why**: Quickly identifies strong positive or negative relationships among scores.