**✅ Cell 1**

python

CopyEdit

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

* **Problem Step**: *Import necessary Python libraries.*
* **Definitions**:
  + pandas (pd): For working with tabular data (like Excel/CSV files).
  + matplotlib.pyplot (plt): Basic plotting tool for visualizations.
  + seaborn (sns): Statistical data visualization built on top of matplotlib.
* **Why Used**:  
  These are core libraries for loading, exploring, and plotting data (especially during exploratory data analysis).

**✅ Cell 2**

python

CopyEdit

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

df

* **Problem Step**: *Load the dataset into a DataFrame.*
* **Definition**:
  + pd.read\_csv(): Loads a CSV file into a structured DataFrame.
* **Why Used**:  
  To load the “StudentsPerformance” dataset and prepare it for analysis.

**✅ Cell 3**

python

CopyEdit

df.info()

* **Problem Step**: *Inspect structure and data types.*
* **Definition**:
  + .info(): Shows column names, non-null counts, and data types.
* **Why Used**:  
  To check if any data is missing and what types each column is (e.g., int, float, object).

**✅ Cell 4**

python

CopyEdit

df.describe()

* **Problem Step**: *Summarize numerical features.*
* **Definition**:
  + .describe(): Displays summary stats like mean, min, max, etc., for numeric columns.
* **Why Used**:  
  To understand the distribution of scores and detect possible outliers or skewness.

**✅ Cell 5**

python

CopyEdit

df.isnull().sum()

* **Problem Step**: *Check for missing values.*
* **Definitions**:
  + .isnull(): Flags null values with True.
  + .sum(): Counts the total number of missing values per column.
* **Why Used**:  
  To confirm data completeness before proceeding with modeling or visualization.

**✅ Cell 6**

python

CopyEdit

df.duplicated().sum()

* **Problem Step**: *Check for duplicate rows.*
* **Definitions**:
  + .duplicated(): Marks duplicate rows.
  + .sum(): Counts them.
* **Why Used**:  
  Duplicate rows can skew analysis and model results.

**✅ Cell 7**

python

CopyEdit

df = df.drop\_duplicates()

* **Problem Step**: *Remove duplicate entries.*
* **Definition**:
  + .drop\_duplicates(): Removes repeated rows.
* **Why Used**:  
  Cleans the dataset, ensuring each record is unique.

**✅ Cell 8**

python

CopyEdit

plt.figure(figsize=(15,8))

sns.heatmap(df.corr(),annot=True)

plt.show()

* **Problem Step**: *Check correlation between numerical variables.*
* **Definitions**:
  + df.corr(): Computes pairwise correlations.
  + sns.heatmap(): Visualizes correlation matrix as a heatmap.
  + annot=True: Shows values in each cell.
* **Why Used**:  
  To understand relationships between features like math, reading, and writing scores.

**✅ Cell 9**

python

CopyEdit

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

plt.title("Math Score Based on Gender")

plt.show()

* **Problem Step**: *Visualize gender-wise performance in math.*
* **Definitions**:
  + sns.boxplot(): Plots distributions showing median, quartiles, and outliers.
* **Why Used**:  
  Helps compare how males and females perform in math, and identify any outliers.

**✅ Cell 10**

python

CopyEdit

sns.boxplot(x='test preparation course', y='math score', data=df)

plt.title("Math Score Based on Test Preparation")

plt.show()

* **Problem Step**: *Analyze effect of test preparation on math scores.*
* **Why Used**:  
  To visually assess if completing a test prep course impacts performance.

**✅ Cell 11**

python

CopyEdit

sns.pairplot(df)

* **Problem Step**: *Visualize pairwise relationships among numeric features.*
* **Definition**:
  + sns.pairplot(): Creates scatter plots between all pairs of numerical columns.
* **Why Used**:  
  Useful to observe trends, relationships, or clusters between score variables.

**✅ Cell 12**

python

CopyEdit

sns.countplot(x='parental level of education', hue='gender', data=df)

plt.title("Parental Level of Education Based on Gender")

plt.xticks(rotation=45)

plt.show()

* **Problem Step**: *Explore how parental education varies across gender.*
* **Definitions**:
  + sns.countplot(): Counts occurrences of categorical values.
  + hue='gender': Splits bars by gender.
* **Why Used**:  
  Helps understand demographic distribution that may influence performance.

**✅ Cell 13**

python

CopyEdit

sns.countplot(x='lunch', hue='gender', data=df)

plt.title("Lunch Type Based on Gender")

plt.show()

* **Problem Step**: *Visualize lunch type distribution across gender.*
* **Why Used**:  
  Could offer insights on socioeconomic status and its impact on performance.

**✅ Cell 14**

python

CopyEdit

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

plt.title("Math Score Based on Race/Ethnicity")

plt.show()

* **Problem Step**: *Explore score distribution by ethnicity.*
* **Definition**:
  + sns.violinplot(): Combines boxplot and KDE to show full distribution.
* **Why Used**:  
  Gives a richer picture than a boxplot alone, including distribution shape.