STEP 1: Load and Inspect the Dataset

```
.import pandas as pd  # for data manipulationimport matplotlib.pyplot as plt  # for
plottingimport seaborn as sns  # advanced visualizations
```

Explanation:

`pandas` is used to load, clean, and explore tabular data

`matplotlib.pyplot` and `seaborn` are used for creating visualizations

Load the dataset:

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

Explanation:

`df`: A variable (short for **DataFrame**) used to hold the dataset

`pd.read_csv()`: Reads a CSV file and converts it into a DataFrame

`'StudentsPerformance.csv'`: File name (you need to have it in your working directory)

First look at the data

```
.df.head()
```

Explanation:

`df.head()`

shows the **first 5 rows** of the data

Useful to get an idea of the structure

Dataset Shape

```
.df.shape
```

Returns a tuple: (number of rows, number of columns)

Data Types and Missing Values

```
.df.info()
```

Tells us:

- Data types (e.g., object = text, int64 = numbers)
- Non-null values (helps spot missing values)

STEP 2: Data Cleaning

Check for missing values

```
.df.isnull().sum()
```

Explanation:

- `isnull()` returns True for missing values
- `sum()` counts how many missing values each column has

✓ Rename Columns for Simplicity

```
.df.rename(columns={
    'race/ethnicity': 'race',
    'parental level of education': 'parent_edu',
    'test preparation course': 'test_prep',
    'math score': 'math',
    'reading score': 'reading',
    'writing score': 'writing'
}, inplace=True)
```

Explanation:

- `rename()` is used to make column names shorter and easier to code
- `inplace=True` means the changes are applied directly to `df`

Summary Stats

```
.df.describe()
```

Explanation:

- `describe()` gives summary statistics (count, mean, std, min, max, etc.)
- Only applies to **numerical columns**

STEP 3: EDA (Exploratory Data Analysis)

Distribution of Numerical Columns

```
.plt.figure(figsize=(10,5))
sns.histplot(df[’math’], kde=True)
plt.title(’Math Score Distribution’)
```

Explanation:

`histplot()`: Draws a histogram (frequency of scores)

`kde=True`: Adds a curve to show distribution shape

`figsize`: Sets size of the figure

Count Plots for Categorical Columns

```
.sns.countplot(data=df, x=’gender’)
plt.title(’Gender Distribution’)
```

- explaination

Shows how many male vs. female students

Compare Math Score by Gender

```
.sns.boxplot(data=df, x=’gender’, y=’math’)
plt.title(’Math Score by Gender’)
```

- explanation

Boxplots show:

Median (middle line)

Quartiles

Outliers

Correlation Between Scores

```
.sns.heatmap(df[[’math’, ’reading’, ’writing’]].corr(), annot=True, cmap=’coolwarm’)
```

Explanation:

`corr()`: Calculates correlation between numerical columns

`heatmap()`: Visual representation of correlation

`annot=True`: Show numbers inside the squares

STEP 4: More Visual Insights

Effect of Test Preparation on Performance

```
.sns.boxplot(data=df, x=’test_prep’, y=’math’)
plt.title(’Effect of Test Prep on Math Score’)
```

explanation

Compares scores of students who completed test prep vs. not

Average Scores by Parental Education

```
.avg_scores = df.groupby(’parent_edu’)[[’math’, ’reading’,
’writing’]].mean().sort_values(by=’math’, ascending=False)
avg_scores.plot(kind=’bar’, figsize=(12,6))
plt.title(’Average Scores by Parental Education’)
plt.ylabel(’Average Score’)
plt.xticks(rotation=45)
```

Explanation:

`groupby()`: Groups data by parental education

`mean()`: Calculates average score per group

`plot(kind='bar')`: Plots a bar chart

✓ Summary of Tools and Their Purpose

Code	What It Does	Why It’s Used
pd.read_csv()	Load dataset	Bring data into Python
df.head()	Show first few rows	Preview data structure
df.describe()	Summary stats	Understand numerical data
sns.boxplot()	Boxplot	Visualize distribution and outliers
df.isnull().sum()	Check missing values	Needed for cleaning

Code	What It Does	Why It’ s Used
groupby().mean()	Group-wise mean	Compare categories
sns.heatmap()	Correlation plot	Check relationships
df.rename()	Rename columns	Simplify names for coding