

Summary Statistics Python Notebook for Mini Project 2

Step 1: Import relevant Python packages and define functions for enerating summary statistics and producing a visualization.

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt

def generate_summary_stats(file_name):
    """Using the csv file passed in as an argument, this function creates a
    dataframe from it, and then generates summary statistics (mean, median,
    mode, standard deviation, as well as percentiles) for each column of the
    using the pandas describe method.
    """
    df = pd.read_csv(file_name)
    return df.describe(), df.median(numeric_only=True)

def generate_viz(file_name):
    """This function generates a scatter plot visualization of hours studied
    from the Student Performance dataset."""
    df = pd.read_csv(file_name)
    plt.scatter(df["Hours_Studied"], df["Exam_Score"], color="Green")
    plt.xlabel("Hours Studied")
    plt.ylabel("Student Exam Scores")
    plt.title("Relationship Between Hours Studied and Student Exam Scores")
    plt.savefig("performance.png")
    plt.show()
```

Step 2: Read in the StudentPerformanceFactors.csv file into a pandas dataframe.

```
In [2]: student_df = pd.read_csv("StudentPerformanceFactors.csv")
student_df.head()
```

Out[2]:

| | Hours_Studied | Attendance | Parental_Involvement | Access_to_Resources | Extracurricular_Activities |
|---|---------------|------------|----------------------|---------------------|----------------------------|
| 0 | 23 | 84 | Low | High | |
| 1 | 19 | 64 | Low | Medium | |
| 2 | 24 | 98 | Medium | Medium | |
| 3 | 29 | 89 | Low | Medium | |
| 4 | 19 | 92 | Medium | Medium | |

Step 3: Review the summary statistics of the data set.

```
In [3]: summary = generate_summary_stats("StudentPerformanceFactors.csv")
describe_stats = summary[0]
medians = summary[1]
print("Descriptive Statistics: \n", describe_stats, "\n")
print("Medians: \n", medians)
```

Descriptive Statistics:

| | Hours_Studied | Attendance | Sleep_Hours | Previous_Scores \ |
|-------|---------------|-------------|-------------|-------------------|
| count | 6607.000000 | 6607.000000 | 6607.000000 | 6607.000000 |
| mean | 19.975329 | 79.977448 | 7.02906 | 75.070531 |
| std | 5.990594 | 11.547475 | 1.46812 | 14.399784 |
| min | 1.000000 | 60.000000 | 4.00000 | 50.000000 |
| 25% | 16.000000 | 70.000000 | 6.00000 | 63.000000 |
| 50% | 20.000000 | 80.000000 | 7.00000 | 75.000000 |
| 75% | 24.000000 | 90.000000 | 8.00000 | 88.000000 |
| max | 44.000000 | 100.000000 | 10.00000 | 100.000000 |

| | Tutoring_Sessions | Physical_Activity | Exam_Score |
|-------|-------------------|-------------------|-------------|
| count | 6607.000000 | 6607.000000 | 6607.000000 |
| mean | 1.493719 | 2.967610 | 67.235659 |
| std | 1.230570 | 1.031231 | 3.890456 |
| min | 0.000000 | 0.000000 | 55.000000 |
| 25% | 1.000000 | 2.000000 | 65.000000 |
| 50% | 1.000000 | 3.000000 | 67.000000 |
| 75% | 2.000000 | 4.000000 | 69.000000 |
| max | 8.000000 | 6.000000 | 101.000000 |

Medians:

| | |
|-------------------|------|
| Hours_Studied | 20.0 |
| Attendance | 80.0 |
| Sleep_Hours | 7.0 |
| Previous_Scores | 75.0 |
| Tutoring_Sessions | 1.0 |
| Physical_Activity | 3.0 |
| Exam_Score | 67.0 |

dtype: float64

Step 4: Generate a visualization of the data set's columns (in this case, a scatterplot of hours studied vs. exam performance.)

```
In [4]: generate_viz("StudentPerformanceFactors.csv")
```

