

- **Median:** Middle value of sorted data.
- **Skewness:** Asymmetry of the distribution.
- **Kurtosis:** Tailedness of the distribution.

#### 4. Visualization:

- Creates a histogram and overlays a normal distribution curve.

#### 5. Outputs:

- Displays the graph and prints the statistical results in the terminal.

---

## Expected Output

1. **Graph:** A histogram of scores with a green dotted line showing the normal distribution.

2. **Terminal Output:**

```
yaml
```

```
CSV loaded successfully.  
Mean: 85.13333333333334  
Median: 85.0  
Standard Deviation: 6.669332171296397  
Skewness: -0.02144405485824689  
Kurtosis: -1.3028744910160028
```

Let me know if you face any issues while running the code!



explain the code word by word



Here's a word-by-word explanation of the code provided:

---

## Step 1: Importing Libraries

```
python
```

```
from scipy.stats import skew, kurtosis
```

- ``from``: Specifies the source library.
- ``scipy.stats``: A module in the SciPy library providing statistical functions.
- ``skew``: A function to compute skewness.
- ``kurtosis``: A function to compute kurtosis.

```
python
```

```
import matplotlib.pyplot as plt
```

- `import`: Imports a library or module.
- `matplotlib.pyplot`: A submodule for creating plots and graphs.
- `as plt`: Assigns a shorthand name (`plt`) for convenience.

python

```
import scipy.stats as stats
```

- `scipy.stats`: Provides statistical functions (e.g., normal distributions).

python

```
import pandas as pd
```

- `pandas`: A library for working with structured data like CSV files.
- `as pd`: Shorthand name (`pd`) for pandas.

python

```
import numpy as np
```

- `numpy`: A library for numerical computations.
- `as np`: Shorthand name (`np`) for numpy.

---

## Step 2: Creating a Sample Dataset

python

```
data = {  
    "Student Name": ["A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N",  
    "O"],  
    "Math Score": [78, 85, 90, 72, 88, 95, 80, 83, 89, 92, 87, 79, 91, 94, 82]  
}
```

- `data`: A Python dictionary containing:
  - `"Student Name"`: Keys represent column names (e.g., names of students).
  - `["A", "B", ...]`: List of student names.
  - `"Math Score"`: Another key representing the column of scores.
  - `[78, 85, ...]`: List of math scores.

python

```
df = pd.DataFrame(data)
```

- `pd.DataFrame()`: Converts the dictionary into a pandas DataFrame, a tabular structure.

```
python
```

```
df.to_csv("score.csv", index=False)
```

- `to_csv()`: Saves the DataFrame to a CSV file named `score.csv`.
  - `index=False`: Excludes the DataFrame index from the CSV file.
- 

### Step 3: Reading the CSV File

```
python
```

```
df = pd.read_csv("score.csv")
```

- `pd.read_csv()`: Reads the `score.csv` file into a DataFrame.
- `df`: Stores the DataFrame.

```
python
```

```
print("CSV loaded successfully.")
```

- `print()`: Displays a success message to confirm the file loaded.
- 

### Step 4: Extracting Data

```
python
```

```
x = df['Math Score'].dropna().to_numpy()[ :15]
```

- `df['Math Score']`: Accesses the "Math Score" column from the DataFrame.
- `.dropna()`: Removes missing (NaN) values, if any.
- `.to_numpy()`: Converts the column to a NumPy array.
- `[ :15]`: Selects the first 15 elements from the array.

```
python
```

```
x.sort()
```

- `.sort()`: Sorts the scores in ascending order.
- 

### Step 5: Calculating Basic Statistics

python

```
mean = np.mean(x)
```

- `np.mean(x)`: Calculates the mean (average) of the array `x`.

python

```
sd = np.std(x, ddof=1)
```

- `np.std()`: Computes the standard deviation of the array `x`.
- `ddof=1`: Ensures the formula uses the unbiased sample standard deviation.

python

```
median = np.median(x)
```

- `np.median()`: Finds the middle value of the sorted array `x`.

## Step 6: Normal Distribution

python

```
fit = stats.norm.pdf(x, mean, sd)
```

- `stats.norm.pdf()`: Calculates the probability density function (PDF) for a normal distribution.
- `x`: Data points.
- `mean`: The average value of `x`.
- `sd`: The standard deviation of `x`.

## Step 7: Visualization

python

```
plt.hist(x, density=True, color="yellow", ec="white", bins=6, alpha=0.7, label="Histogram")
```

- `plt.hist()`: Creates a histogram.
- `density=True`: Normalizes the histogram to show probabilities instead of counts.
- `color="yellow"`: Sets the bar color.
- `ec="white"`: Adds a white edge around the bars.
- `bins=6`: Specifies the number of bars (bins) in the histogram.
- `alpha=0.7`: Makes the bars slightly transparent.
- `label="Histogram"`: Adds a label for the histogram.

python

```
plt.plot(x, fit, "go--", label="Normal Distribution")
```

- `plt.plot()`: Plots the normal distribution curve.
- `"go--"`: Green circles connected by dashed lines.
- `label="Normal Distribution"`: Adds a label for the legend.

python

```
plt.title("Math Scores Distribution")
plt.xlabel("Marks")
plt.ylabel("Density")
plt.legend()
plt.show()
```

- `plt.title()`: Adds a title to the plot.
- `plt.xlabel()`: Sets the x-axis label.
- `plt.ylabel()`: Sets the y-axis label.
- `plt.legend()`: Displays the legend.
- `plt.show()`: Renders the graph.

## Step 8: Skewness and Kurtosis

python

```
skewness_value = skew(x)
```

- `skew(x)`: Computes the skewness of the array `x`.
  - Positive = Right-skewed.
  - Negative = Left-skewed.

python

```
kurtosis_value = kurtosis(x, fisher=False)
```

- `kurtosis(x, fisher=False)`: Computes the kurtosis of the array `x`.
  - High kurtosis = Heavy tails (extreme values).
  - Low kurtosis = Light tails.

## Step 9: Display Results

python

```
print(f"Mean: {mean}")
```

```
print(f"Standard Deviation: {sd}")
print(f"Skewness: {skewness_value}")
print(f"Kurtosis: {kurtosis_value}")
```

- `print()`: Displays the calculated values:
  - **Mean**: Average of the scores.
  - **Median**: Middle value.
  - **Standard Deviation**: Spread of scores.
  - **Skewness**: Asymmetry of the distribution.
  - **Kurtosis**: Tailedness of the distribution.

---

This explanation covers every word and concept in the code. Let me know if you have further questions!