**Code Functionality**

**1. Function Definition and Imports**

- def get_pdf_probability(dataset, startrange, endrange):
    - Defines a function that takes three arguments: dataset (the data), startrange, and endrange (the range of values for which to calculate the probability).
- from matplotlib import pyplot as plt
    - Imports the pyplot module from the matplotlib library for plotting, aliasing it as plt.
- from scipy.stats import norm
    - Imports the norm class from scipy.stats, which is used to model and work with the normal distribution.
- import seaborn as sns
    - Imports the seaborn library for statistical data visualization, aliasing it as sns.

**2. Data Visualization**

- ax=sns.distplot(dataset,kde=True,kde_kws={'color':'blue'},color='Green')
    - This line generates a distribution plot of the dataset using seaborn.
    - kde=True plots a Kernel Density Estimate (KDE) curve, which is a smoothed curve representing the data's distribution.
    - kde_kws={'color':'blue'} sets the color of the KDE curve to blue.
    - color='Green' sets the color of the histogram bars to green.
    - The returned axes object is stored in the ax variable.
- plt.axvline(startrange,color='Red')
    - This draws a vertical red line on the plot at the startrange value.
- plt.axvline(endrange,color='Red')
    - This draws another vertical red line on the plot at the endrange value.
    - These lines visually represent the range for which the probability is being calculated.

**3. Parameter Calculation**

- sample = dataset
    - Creates a copy or reference to the dataset.
- sample_mean=sample.mean()
    - Calculates the mean (average) of the dataset.
- sample_std=sample.std()

- Calculates the standard deviation of the dataset.

**4. Printing Parameters**

- print('Mean={:,.3f}, Standard Deviation={:,.3f}'.format(sample_mean, sample_std))
    - This is a corrected or alternative way to print the same information using str.format(), which correctly formats the numbers with three decimal places and a thousands separator (the ,).

**5. Distribution Modeling**

- dist=norm(sample_mean, sample_std)
    - This is a crucial step. It creates a norm (normal distribution) object from scipy.stats. This object is parameterized with the calculated sample_mean and sample_std. This line effectively models the dataset as a normal distribution.

**6. Probability Calculation**

- #sample probabilities for a range of outcomes
    - A comment indicating the next section's purpose.
- values=[value for value in range(startrange, endrange)]
    - This creates a list of integer values from startrange up to (but not including) endrange.
- probabilities=[dist.pdf(value) for value in values]
    - This is a list comprehension that calculates the Probability Density Function (PDF) value for each integer in the values list. The PDF gives the relative likelihood of a value occurring.
- prob=sum(probabilities)
    - This line sums the PDF values of the integer points within the specified range.

**7. Final Output and Return**

- print("The area between range({}:{}):{}".format(startrange,endrange,sum(probabilities)))
    - Prints the calculated sum of probabilities along with the start and end range.
- return prob
    - The function returns the calculated probability (the sum of the PDF values).