PDF Assignment_Code Explanation

```
def get_pdf_probability(dataset, startRange, endRange):
```

Function Definition: The function get_pdf_probability takes three parameters:

- dataset: A dataset of numerical values (likely a list or array).
- startRange and endRange: These specify the range within which we calculate the probability density function (PDF).

```
from matplotlib import pyplot
from scipy.stats import norm
import seaborn as sns
```

Importing Libraries:

- pyplot from matplotlib is imported for plotting graphs.
- norm from scipy.stats provides tools for working with normal distributions.
- seaborn is imported as sns, a visualization library for statistical data.

```
ax = sns.distplot(dataset, kde=True, color='green', kde_kws={'color':'blue'})
```

Plotting the Distribution:

- sns.distplot is used to plot a histogram of the dataset.
- kde=True adds a kernel density estimate (KDE) line, which smooths the histogram into a continuous probability density curve.
- color='green' sets the histogram color to green, while kde_kws={'color':'blue'} makes the KDE line blue.

```
pyplot.axvline(startRange, color='Red')
pyplot.axvline(endRange, color='Red')
```

Marking the Range:

- axvline is used to draw vertical lines at startRange and endRange on the plot, in red. These lines visually indicate the range within which we'll calculate the probability density.

```
sample = dataset
```

Defining the Sample:

- sample is set to the input dataset for easier reference.

```
sample_mean = sample.mean()
sample_std = sample.std()
print('Mean = %.3f, Standard Deviation = %.3f' %(sample_mean, sample_std))
```

Calculating Mean and Standard Deviation:

- sample_mean and sample_std are the mean and standard deviation of the dataset, calculated using sample.mean() and sample.std().
- These values are printed with formatting to three decimal places.

```
dist = norm(sample_mean, sample_std)
```

Defining the Normal Distribution:

- dist is set to a normal distribution (norm) with the calculated mean and standard deviation of sample.

```
values = [value for value in range(startRange, endRange)]
```

Generating Values for Probability Calculation:

- A list comprehension creates a list of integers from startRange to endRange.
- These values represent the range for which the probability density will be calculated.

```
probabilities = [dist.pdf(value) for value in values]
```

Calculating Probability Density:

- dist.pdf(value) computes the probability density for each value in values.
- This creates a list probabilities containing the PDF values for each value in the specified range.

```
prob = sum(probabilities)
print("The area between range({},{}):{}".format(startRange, endRange, prob))
```

Summing the Probabilities:

- prob is the sum of all PDF values in probabilities, representing the approximate area (or total probability density) between startRange and endRange.
- The result is printed.

```
return prob
```

Returning the Probability:

- prob is returned as the output of the function, which represents the total PDF area between startRange and endRange.