**get\_pdf\_probability() Explained**

This function get\_pdf\_probability() is designed to:

1. **Visualize the distribution** of a dataset
2. **Distribution and PDF (Probability Density Function):**
3. **Calculate the approximate total PDF (probability density function) area** between two values in the dataset
4. **Visualization**

ax = sns.distplot(dataset, kde=True, kde\_kws={'color':'blue'}, color='Green')

Plots a histogram and a **KDE (kernel density estimate)** to visualize the data distribution.

Histogram bars = green

KDE (smooth curve) = blue

pyplot.axvline(startrange, color='Red')

pyplot.axvline(endrange, color='Red')

Adds red vertical lines to the plot at startrange and endrange.

1. **Distribution and PDF (Probability Density Function):**

sample = dataset

sample\_mean = sample.mean()

sample\_std = sample.std()

print('Mean=%.3f, Standard Deviation=%.3f' % (sample\_mean, sample\_std))

Calculates and prints the **mean** and **standard deviation** of the dataset.

dist = norm(sample\_mean, sample\_std)

Creates a **normal distribution object** using the dataset’s mean and std deviation.

1. **PDF Area Calculation:**

values = [value for value in range(startrange, endrange)]

Generates a list of integers from startrange to endrange - 1.

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

Computes the **PDF value at each point**. (Note: this is **not a probability**, but a density.)

prob = sum(probabilities)

print("The area between range({},{}):{}".format(startrange,endrange, sum(probabilities)))

Adds the densities (not true probability area under the curve). Returns the **sum of PDF values**, which approximates the area but not exactly unless scaled properly.