Bernoulli  
  
1)This Python function, named generate\_bernoulli, generates random variables following a Bernoulli distribution.

* p: The probability of success in each Bernoulli trial. This is a parameter you pass to the function.
* size: The number of random variables to generate. This is also a parameter you pass to the function.

The function uses NumPy's np.random.binomial function, which simulates a binomial distribution with n=1 (because it's a Bernoulli distribution). The function returns an array (random\_variables) containing the generated random variables.  
  
2) The function bernoulli\_pmf(x, p) calculates the probability mass function (PMF) of a Bernoulli distribution for a given value x and probability of success p.

the formula p^x \* (1-p)^(1-x) is the probability mass function of a Bernoulli distribution. It gives the probability of observing a particular outcome x (0 or 1) in a single Bernoulli trial with probability of success p. The term p^x corresponds to the probability of success if x is 1, and (1-p)^(1-x) corresponds to the probability of failure if x is 0.  
  
3) The bernoulli\_cdf(x, p) function calculates the cumulative distribution function (CDF) of a Bernoulli distribution for a given value x and probability of success p. The CDF represents the probability that the random variable takes a value less than or equal to a given point.

* If x is less than 0, the CDF is 0 because the random variable in a Bernoulli distribution cannot take negative values.
* If 0 <= x < 1, the CDF is equal to 1 - p. This corresponds to the probability that the random variable is less than or equal to 0, which is the probability of failure (1) minus the probability of success (p).
* If x is 1 or greater, the CDF is 1 because the random variable in a Bernoulli distribution can only take values of 0 or 1, and the cumulative probability is 1 when x is 1 or greater.

4) The function bernoulli\_variance(p) calculates the variance of a Bernoulli distribution, where p is the probability of success in a Bernoulli trial. The variance is a measure of the spread or dispersion of a probability distribution.

The formula for the variance of a Bernoulli distribution is given by:

Var(X)=p⋅(1−p)Var(X)=p⋅(1−p)

where:

* Var(X)Var(X) is the variance of the Bernoulli distribution.
* pp is the probability of success in a single Bernoulli trial.

The expression p⋅(1−p)p⋅(1−p) represents the product of the probability of success and the probability of failure. It quantifies how much the distribution of the Bernoulli random variable deviates from its mean, which is pp for a Bernoulli distribution.

5) The function bernoulli\_expectation(p) calculates the expected value (or mean) of a Bernoulli distribution, where p is the probability of success in a Bernoulli trial. The expected value represents the average or central tendency of a probability distribution.

For a Bernoulli distribution, the expected value is equal to the probability of success. Mathematically, it can be expressed as:

E(X)=pE(X)=p

where:

* E(X)E(X) is the expected value of the Bernoulli distribution.
* pp is the probability of success in a single Bernoulli trial.

5)

The plot\_pmf(p) function is designed to plot the probability mass function (PMF) of a Bernoulli distribution for a given probability of success p. The PMF represents the probabilities of different discrete outcomes of a random variable.

Here's a breakdown of the function:

* **Line 2:** It calculates the PMF values for each possible outcome of the Bernoulli distribution using the bernoulli\_pmf(x, p) function, where x is one of the unique values (presumably 0 and 1).
* **Line 3-4:** It creates a bar plot using Matplotlib's plt.bar() function. The x-axis represents the unique values of the random variable (0 and 1), and the height of the bars represents the corresponding PMF values. The color='blue' argument sets the color of the bars, and align='center' centers the bars on the x-axis. The alpha=0.7 argument controls the transparency of the bars, and width=0.01 sets the width of the bars. The label='PMF' is used for the legend.
* **Line 5:** It adds scatter points on top of the bars using Matplotlib's plt.scatter() function. This can be useful to visually emphasize the specific data points.
* **Line 6-7:** It sets the x-axis ticks to be at the unique values (0 and 1) and labels the x-axis as 'X (Random variable)' and the y-axis as 'PMF'.
* **Line 8:** It adds a title to the plot indicating that it represents the Probability Mass Function for a Bernoulli distribution with the specified probability of success p.
* **Line 9:** It adds a legend to the plot.
* **Line 10:** It displays the plot using plt.show().

6) The plot\_cdf(p) function is designed to plot the cumulative distribution function (CDF) of a Bernoulli distribution for a given probability of success p. The CDF represents the probability that the random variable takes a value less than or equal to a given point.

Here's an explanation of the function:

* **Line 2:** It calculates the CDF values for each possible outcome of the Bernoulli distribution using the bernoulli\_cdf(x, p) function, where x is one of the unique values (presumably 0 and 1).
* **Line 3-4:** It creates a step plot using Matplotlib's plt.step() function. The x-axis represents the unique values of the random variable (0 and 1), and the y-axis represents the corresponding CDF values. The marker='o' argument adds circular markers at the data points, and color='green' sets the color of the step plot. The where='post' argument ensures that the step function increases at the right endpoint of each step. The label='CDF' is used for the legend.
* **Line 5:** It adds a title to the plot indicating that it represents the Cumulative Distribution Function for a Bernoulli distribution with the specified probability of success p.
* **Line 6-7:** It sets the x-axis ticks to be at the unique values (0 and 1) and labels the x-axis as 'X (Random variable)' and the y-axis as 'CDF'.
* **Line 8:** It adds a legend to the plot.
* **Line 9:** It displays the plot using plt.show().