

## Creating a Probability Distribution - Step 3/4

The screenshot displays a learning interface for 'Introduction to Statistics In Python'. The top navigation bar includes 'Learn / Courses / Introduction to Statistics In Python', a 'Course Outline' button, and a 'Daily XP 1489' indicator. The main content area is divided into two panels. The left panel, titled 'Exercise', contains the text: 'Creating a probability distribution. A new restaurant opened a few months ago, and the restaurant's management wants to optimize its seating space based on the size of the groups that come most often. On one night, there are 10 groups of people waiting to be seated at the restaurant, but instead of being called in the order they arrived, they will be called randomly. In this exercise, you'll investigate the probability of groups of different sizes getting picked first. Data on each of the ten groups is contained in the `restaurant_groups` DataFrame. Remember that expected value can be calculated by multiplying each possible outcome with its corresponding probability and taking the sum. The `restaurant_groups` data is available. `pandas` is loaded as `pd`, `numpy` is loaded as `np`, and `matplotlib.pyplot` is loaded as `plt`.' Below this is a section 'Instructions 3/4' with a '25 XP' badge. It lists four tasks: 1. Create a histogram of the `group_size` column of `restaurant_groups`, setting bins to `[2, 3, 4, 5, 6]`. Remember to show the plot. 2. Count the number of each `group_size` in `restaurant_groups`, then divide by the number of rows in `restaurant_groups` to calculate the probability of randomly selecting a group of each size. Save as `size_dist`. Reset the index of `size_dist`. Rename the columns of `size_dist` to `group_size` and `prob`. 3. Calculate the expected value of the `size_dist`, which represents the expected group size, by multiplying the `group_size` by the `prob` and taking the sum. 4. Calculate the probability of randomly picking a group of 4 or more people by subsetting for groups of size 4 or more and summing the probabilities of selecting those groups. A 'Take Hint (-7 XP)' button is located between instructions 3 and 4. The right panel, titled 'script.py', shows a code editor with the following Python code: 

```
1 # Create probability distribution
2 size_dist = restaurant_groups['group_size'].value_counts() /
  restaurant_groups.shape[0]
3 # Reset index and rename columns
4 size_dist = size_dist.reset_index()
5 size_dist.columns = ['group_size', 'prob']
6
7 # Calculate expected value
8 expected_value = np.sum(____)
9 print(expected_value)
```

 Below the code editor are buttons for 'Run Code' and 'Submit Answer'. At the bottom, there is an 'IPython Shell' section with a prompt 'In [1]:'.

Figure: Screenshot showing the calculation of expected value for group sizes in the restaurant data.

### Question

In step 3/4, you are tasked to calculate the expected value of the `size_dist` dataset. This expected value represents the average group size based on the given probabilities of each group size.

**\*\*Instructions for Step 3/4:\*\***

1. Divide the frequency of each group size by the total number of rows in the dataset to get the probability distribution.
2. Reset the index of the resulting probability table and rename its columns

for clarity.

3. Use the ``np.sum()`` function to calculate the weighted average of group sizes based on their probabilities and print the result.

## Correct Code Solution

```
# Check if you have calculated the probability distribution
size_dist = restaurant_groups['group_size'].value_counts() /
restaurant_groups.shape[0]

# Ensure you reset the index and rename the columns
size_dist = size_dist.reset_index()
size_dist.columns = ['group_size', 'prob']

# Use this to calculate the expected value
expected_value = np.sum(size_dist['group_size'] * size_dist['prob'])
print(expected_value)
```

## Explanation

1. **Probability Distribution:** The ``value_counts()`` method is used to count each group size's occurrence, and dividing by the total number of rows provides the probability of each size.
2. **Formatting:** Resetting the index and renaming columns ensures the resulting DataFrame is structured for clarity and usability in further calculations.
3. **Expected Value:** Using the weighted average formula (sum of each group size multiplied by its probability), ``np.sum()`` computes the average group size considering the given probabilities.
4. **Verification:** The printed result offers a clear insight into the likely average group size based on historical data.