**Title:** Multiple categorical variables

**Description**: Use the `crosstab` function to create contingency tables with multiple categorical variables.

**Task 5 Explanation**

**Report: Contingency Table Analysis Using crosstab**

**Objective**

The purpose of this report is to demonstrate how to create and interpret a contingency table using Python's pandas.crosstab function. Contingency tables summarize relationships between categorical variables and provide insights into how different categories interact.

**Overview of Contingency Tables**

A contingency table is a statistical tool used to analyze the frequency distribution of categorical variables. It is widely applied in data science for exploring relationships between variables, identifying trends, and performing preliminary data analysis.

Key features of a contingency table:

* **Rows and Columns**: Represent unique categories of variables.
* **Frequencies**: Show counts of observations for each combination of categories.
* **Row and Column Margins**: Provide totals for rows and columns, allowing for easier interpretation of distributions.

The pandas.crosstab function in Python offers a flexible and efficient way to generate contingency tables with options for multiple variables, aggregation, and marginal totals.

**Dataset**

The analysis uses a small dataset containing three categorical variables:

* **Gender**: Male or Female.
* **Department**: HR, IT, or Finance.
* **Shift**: Day or Night.

The dataset is structured as follows:

| **Gender** | **Department** | **Shift** |
| --- | --- | --- |
| Male | HR | Day |
| Female | IT | Night |
| Female | Finance | Day |
| Male | IT | Night |
| Male | HR | Day |
| Female | Finance | Night |

**Methodology**

The following steps were taken to create the contingency table:

1. **Import Libraries**: Import the pandas library to handle data manipulation and analysis.
2. **Prepare Data**: Create a pandas DataFrame from the dataset.
3. **Apply crosstab**:
   * Row categories: Combinations of Gender and Department.
   * Column categories: Shift (Day or Night).
4. **Include Totals**: Add row and column totals using the margins=True parameter.

**Results**

The generated contingency table summarizes the relationship between Gender, Department, and Shift as follows:

| **Shift** | **Day** | **Night** | **All** |
| --- | --- | --- | --- |
| Female Finance | 1 | 1 | 2 |
| Female IT | 0 | 1 | 1 |
| Female HR | 0 | 0 | 0 |
| Male Finance | 0 | 0 | 0 |
| Male IT | 0 | 1 | 1 |
| Male HR | 2 | 0 | 2 |
| **All** | 3 | 3 | 6 |

**Interpretation**

1. **Category Combinations**:
   * The rows represent combinations of Gender and Department.
   * The columns represent the Shift categories (Day and Night).
2. **Insights**:
   * Female employees in the Finance department worked both Day and Night shifts (1 occurrence each).
   * Male employees in the HR department only worked Day shifts (2 occurrences).
   * The total number of shifts recorded is 6, evenly split between Day and Night shifts.
3. **Row and Column Totals**:
   * The **row totals** indicate the total number of shifts worked by each combination of Gender and Department.
   * The **column totals** show the total number of Day and Night shifts across all categories.

**Conclusion**

The contingency table provides a clear and concise summary of how Gender, Department, and Shift interact in the dataset. This analysis can be extended further by adding more variables, using aggregation functions, or visualizing the data through charts.

For more complex datasets or larger-scale analyses, this approach can be applied to uncover trends and relationships in categorical data efficiently.

Top of Form

Bottom of Form