To create a sample dataset of marks for various subjects using pandas in Python, you can follow these steps:

1. Import the pandas library:

- import pandas as pd

2. Create a dictionary containing the marks data for each subject:

```
marks_data = {
    'Maths': [80, 75, 90, 85, 95],
    'English': [70, 65, 80, 75, 85],
    'Science': [85, 80, 95, 90, 92],
    'History': [75, 70, 85, 80, 88]
}
```

3. Create a pandas DataFrame using the dictionary:

```
df = pd.DataFrame(marks_data)
```

- Here's the complete code:

```
import pandas as pd
marks_data = {
    'Maths': [80, 75, 90, 85, 95],
    'English': [70, 65, 80, 75, 85],
    'Science': [85, 80, 95, 90, 92],
    'History': [75, 70, 85, 80, 88]
}
df = pd.DataFrame(marks_data)
```

The resulting DataFrame `df` will have the subject names as columns and the marks data as rows. Each row represents a student, and each column represents a subject.

You can further manipulate and analyze the data using pandas functions and methods on the created DataFrame.

++

Selecting a single column and multiple columns from a DataFrame using the indexing operator in pandas:

```
import pandas as pd

data = {
    'Name': ['John', 'Emma', 'Tom', 'Emily'],
    'Age': [25, 28, 24, 26],
    'Subject': ['Maths', 'English', 'Science', 'History'],
    'Marks': [80, 90, 85, 70]
}

df = pd.DataFrame(data)

single_column = df['Subject']

print(single_column)

multiple_columns = df[['Name', 'Marks']]

print(multiple_columns)
```

In the above code, we create a sample DataFrame `df` with columns 'Name', 'Age', 'Subject', and 'Marks'.

To select a single column, we use the indexing operator `[]` and pass the column name as a string. For example, `df['Subject']` will return the 'Subject' column.

To select multiple columns, we use the indexing operator `[]` and pass a list of column names. For example, `df[['Name', 'Marks']]` will return a DataFrame with only the 'Name' and 'Marks' columns.

You can run this code and observe the output to see the selected columns from the DataFrame.