

Fill in Missing Values and Sum Values with Pivot Tables - Instruction 2

The screenshot shows a web browser window displaying a DataCamp exercise. The exercise title is "Fill in missing values and sum values with pivot tables". The instructions state that the `.pivot_table()` method has several useful arguments, including `fill_value` and `margins`. The `fill_value` argument replaces missing values with a real value (known as imputation). The `margins` argument is a shortcut for when you pivoted by two variables, but also wanted to pivot by each of those variables separately: it gives the row and column totals of the pivot table contents. The exercise asks the user to print the mean `weekly_sales` by department and type, filling in any missing values with 0, and then print the mean `weekly_sales` by department and type, filling in any missing values with 0 and summing all rows and columns. The code editor shows the following code:

```
1 # Print the mean weekly_sales by department and type; fill missing values with 0s; sum all
  rows and cols
2 print(sales.pivot_table(values="weekly_sales", index="department", columns="type", _____))
```

The output shows a pivot table with 80 rows and 2 columns:

| | type | weekly_sales |
|-----|------------|--------------|
| 3 | 17169.063 | 36580.655 |
| 4 | 44285.399 | 51219.654 |
| 5 | 34821.011 | 63236.875 |
| ... | ... | ... |
| 95 | 123933.787 | 77882.102 |
| 96 | 21367.043 | 9528.538 |
| 97 | 28471.267 | 5828.873 |
| 98 | 12875.423 | 217.428 |
| 99 | 379.124 | 0.000 |

The `.pivot_table()` method has several useful arguments, including `fill_value` and `margins`.

`fill_value` replaces missing values with a real value (known as imputation). `margins` is a shortcut for when you pivoted by two variables, but also wanted to pivot by each of those variables separately: it gives the row and column totals of the pivot table contents.

In this exercise, you'll practice using these arguments to up your pivot table skills, which will help you crunch numbers more efficiently!

`sales` is available and `pandas` is imported as `pd`.

Final Answer - Instruction 2

```
# Print the mean weekly_sales by department and type; fill missing values
with 0s and sum all rows and columns
print(sales.pivot_table(values="weekly_sales", index="department",
columns="type", fill_value=0, margins=True))
```