

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

The screenshot shows a web browser window with a DataCamp course page. The page title is 'Fill in missing values and sum values with pivot tables'. The left sidebar contains the exercise title and instructions. The main content area shows the exercise instructions and a code editor. The code editor contains the following Python 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 code editor also shows the output of the code, which is a pivot table with 80 rows and 2 columns. The output is displayed in a table format with the following data:

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

The interface also includes a 'Run Code' button and a 'Submit Answer' button. The bottom of the screen shows a Windows taskbar with various application icons and the system clock.

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))
```