

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 exercise instructions are as follows:

Fill in missing values and sum values with pivot tables

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*). What to replace missing values with is a topic big enough to have its own course (Dealing with Missing Data in Python), but the simplest thing to do is to substitute a dummy value.
- `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`.

Instructions 2/2 (50 XP)

- Print the mean `weekly_sales` by `department` and `type`, filling in any missing values with `0`.
- Print the mean `weekly_sales` by `department` and `type`, filling in any missing values with `0` and summing all rows and columns.

[Take Hint \(-15 XP\)](#)

The code editor on the right 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 IPython Shell shows the output of the code:

```
3      17169.083    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

[80 rows x 2 columns]
```

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))
```