

Fill in Missing Values and Sum Values with Pivot Tables

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 course navigation menu. The main content area shows the exercise instructions. The right sidebar contains a code editor with a Python script and a Jupyter Notebook shell.

Exercise

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 1/2 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 \(-15 XP\)](#)

```
script.py
1 # Print mean weekly_sales by department and type; fill missing values with 0
2 print(sales.pivot_table(_____))
```

[Run Code](#) [Submit Answer](#)

Jupyter Shell [Slides](#)

```
In [1]:
```

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 1

```
# Print mean weekly_sales by department and type; fill missing values with 0
print(sales.pivot_table(values="weekly_sales", index="department",
columns="type", fill_value=0))
```