

Pivoting on One Variable - Instruction 3

The screenshot shows a web browser with multiple tabs. The active tab is 'campus.datacamp.com/courses/data-manipulation-with-pandas/aggregating-dataframes?ex=14'. The page content is from a DataCamp course titled 'Data Manipulation with pandas'. It features an 'Exercise' section for 'Pivoting on one variable'. The instructions are as follows:

- Get the mean `weekly_sales` by `type` using `.pivot_table()` and store as `mean_sales_by_type`.
- Get the mean and median (using NumPy functions) of `weekly_sales` by `type` using `.pivot_table()` and store as `mean_med_sales_by_type`.
- Get the mean of `weekly_sales` by `type` and `is_holiday` using `.pivot_table()` and store as `mean_sales_by_type_holiday`.

A 'Take Hint (-8 XP)' button is visible. To the right is a code editor with the following Python code:

```
1 # Pivot for mean weekly_sales by store type and holiday
2 mean_sales_by_type_holiday = sales.pivot_table(____)
3
4 # Print mean_sales_by_type_holiday
5 print(mean_sales_by_type_holiday)
```

Below the code editor is a terminal window showing the output of the script:

```
<script.py> output:
      mean      median
type weekly_sales weekly_sales
A      23674.667    11943.92
B      25696.678    13356.08
```

The terminal prompt is `In [1]:`.

Pivot tables are the standard way of aggregating data in spreadsheets.

In pandas, pivot tables are essentially another way of performing grouped calculations. That is, the `pivot_table()` method is an alternative to `.groupby()`.

In this exercise, you'll perform calculations using `.pivot_table()` to replicate the calculations you performed in the last lesson using `.groupby()`.

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

Final Answer - Instruction 3

```
# Pivot for mean weekly_sales by store type and holiday
mean_sales_by_type_holiday = sales.pivot_table(values="weekly_sales",
index="type", columns="is_holiday")
```

```
# Print mean_sales_by_type_holiday
print(mean_sales_by_type_holiday)
```