

## Calculations with .groupby()

The screenshot shows a web browser window with the URL `campus.datacamp.com/courses/data-manipulation-with-pandas/aggregating-dataframes?ex=11`. The page is titled "Learn / Courses / Data Manipulation with pandas" and features a "Course Outline" button. The main content area is an exercise titled "Calculations with .groupby()". The instructions state: "The .groupby() method makes life much easier. In this exercise, you'll perform the same calculations as last time, except you'll use the .groupby() method. You'll also perform calculations on data grouped by two variables to see if sales differ by store type depending on if it's a holiday week or not." Below the instructions, there is a "Take Hint (-15 XP)" button. The code editor shows the following Python code:

```
1 # From previous step
2 sales_by_type = sales.groupby("type")["weekly_sales"].sum()
3
4 # Group by type and is_holiday; calc total weekly sales
5 sales_by_type_is_holiday = ____
6 print(sales_by_type_is_holiday)
```

The output of the script is displayed in the IPython Shell:

```
<script.py> output:
type
A    0.91
B    0.89
Name: weekly_sales, dtype: float64

In [1]:
```

The .groupby() method makes life much easier. In this exercise, you'll perform the same calculations as last time, except you'll use the .groupby() method. You'll also perform calculations on data grouped by two variables to see if sales differ by store type depending on if it's a holiday week or not.

sales is available and pandas is loaded as pd.

## Final Answer

```
# From previous step
sales_by_type = sales.groupby("type")["weekly_sales"].sum()

# Group by type and is_holiday; calc total weekly sales
sales_by_type_is_holiday = sales.groupby(["type", "is_holiday"])
["weekly_sales"].sum()
print(sales_by_type_is_holiday)
```