

Multiple Grouped Summaries

The screenshot shows a web browser window with multiple tabs. The active tab is a DataCamp course page titled 'Data Manipulation with pandas'. The page displays an exercise titled 'Multiple grouped summaries'. The exercise text explains that the `.agg()` method is useful for computing multiple statistics on grouped data. It mentions that NumPy, imported as `np`, has various summary statistics functions like `np.min`, `np.max`, `np.mean`, and `np.median`. The instructions list three tasks: 1. Import `numpy` as `np`. 2. Use `groupby()` and `.agg()` to calculate min, max, mean, and median for `weekly_sales` by store type, storing the result in `sales_stats`. 3. Use `groupby()` and `.agg()` to calculate min, max, mean, and median for `unemployment` and `fuel_price_usd_per_l` by store type, storing the result in `unemp_fuel_stats`. A 'Take Hint (-30 XP)' button is visible. On the right, a code editor shows a partially completed Python script. The bottom of the interface shows an 'IPython Shell' with a message: 'We closed your session due to inactivity. If the problem persists, please report an issue.' and a 'Restart Session' button. The browser's taskbar at the bottom shows the system clock as 14:43 on 25/11/2024.

Exercise

Multiple grouped summaries

Earlier in this chapter, you saw that the `.agg()` method is useful to compute multiple statistics on multiple variables. It also works with grouped data. NumPy, which is imported as `np`, has many different summary statistics functions, including: `np.min`, `np.max`, `np.mean`, and `np.median`.

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

Instructions

- Import `numpy` with the alias `np`.
- Get the min, max, mean, and median of `weekly_sales` for each store type using `.groupby()` and `.agg()`. Store this as `sales_stats`. Make sure to use `numpy` functions!
- Get the min, max, mean, and median of `unemployment` and `fuel_price_usd_per_l` for each store type. Store this as `unemp_fuel_stats`.

Take Hint (-30 XP)

```
1 # Import numpy with the alias np
2 ----
3
4 # For each store type, aggregate weekly_sales: get min, max, mean, and median
5 sales_stats = ----
6
7 # Print sales_stats
8 print(sales_stats)
9
10 # For each store type, aggregate unemployment and fuel_price_usd_per_l: get min, max, mean,
11 and median
12 unemp_fuel_stats = ----
13
14 # Print unemp_fuel_stats
15 print(unemp_fuel_stats)
```

Run Code Submit Answer

IPython Shell Slides

We closed your session due to inactivity.
If the problem persists, please report an issue.

Restart Session

Earlier in this chapter, you saw that the `.agg()` method is useful to compute multiple statistics on multiple variables. It also works with grouped data. NumPy, which is imported as `np`, has many different summary statistics functions, including: `np.min`, `np.max`, `np.mean`, and `np.median`.

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

Final Answer

```
# Import numpy with the alias np
import numpy as np
```

```
# For each store type, aggregate weekly_sales: get min, max, mean, and
median
sales_stats = sales.groupby("type")["weekly_sales"].agg([np.min, np.max,
np.mean, np.median])
```

```
# Print sales_stats
print(sales_stats)
```

```
# For each store type, aggregate unemployment and fuel_price_usd_per_l:
get min, max, mean, and median
```

```
unemp_fuel_stats = sales.groupby("type")[["unemployment",  
"fuel_price_usd_per_l"]].agg([np.min, np.max, np.mean, np.median])  
  
# Print unemp_fuel_stats  
print(unemp_fuel_stats)
```