

Adding a Second Factor to the Analysis (Updated)

The screenshot shows a web browser window with a Jupyter Notebook interface. The browser's address bar shows the URL: `campus.datacamp.com/courses/analyzing-police-activity-with-pandas/exploring-the-relationship-between-gender-and-policing?ex=10`. The notebook has a sidebar on the left with a 'Task Description' section. The main area is divided into two panes: a 'Code' pane on the left and a 'Python Shell' pane on the right. The 'Code' pane contains the following code:

```
1 # Calculate the search rate for each combination of gender and violation
2 print(ri.groupby(['violation', 'driver_gender']).search_conducted.mean())
```

The 'Python Shell' pane shows the output of the code: `violation driver_gender search_conducted`. The notebook interface includes a 'Run Code' button and a 'Submit Answer' button. The bottom of the browser window shows a Windows taskbar with various icons and the system clock.

Task Description

1. Use a `groupby()` to calculate the search rate for each combination of violation and gender.
2. Analyze whether males and females are searched at about the same rate for each violation type.

Code Solution

```
# Reverse the ordering to group by violation before gender
violation_gender_search_rate = ri.groupby(['violation',
'driver_gender']).search_conducted.mean()
print(violation_gender_search_rate)
```

Code Explanation

1. The line `'violation_gender_search_rate = ri.groupby(['violation', 'driver_gender']).search_conducted.mean()'` groups the DataFrame by both 'violation' and 'driver_gender' columns, then calculates the mean of 'search_conducted' for each group. By grouping first by violation, the results are organized to allow for comparison of search rates within each violation type across genders.
2. The line `'print(violation_gender_search_rate)'` outputs the calculated search rates to enable analysis of any patterns or differences between genders for each violation.