

Dropping Columns

The screenshot shows a web browser window with the URL `campus.datacamp.com/courses/analyzing-police-activity-with-pandas/preparing-the-data-for-analysis/ex-3`. The page is titled "Exercise" and "Dropping columns". The instructions on the left state: "Often, a DataFrame will contain columns that are not useful to your analysis. Such columns should be dropped from the DataFrame, to make it easier for you to focus on the remaining columns. In this exercise, you'll drop the 'county_name' column because it only contains missing values, and you'll drop the 'state' column because all of the traffic stops took place in one state (Illinois). Thus, these columns can be dropped because they contain no useful information. The number of missing values in each column has been printed for you."

The instructions list three steps:

- Examine the DataFrame's `shape` to find out the number of rows and columns.
- Drop both the 'county_name' and 'state' columns by passing the column names to the `.drop()` method as a list of strings.
- Examine the `shape` again to verify that there are now two fewer columns.

A "Take Hint (-50 XP)" button is visible below the instructions.

The code editor on the right shows the following Python script:

```
1 # Examine the shape of the DataFrame
2 print(ri.shape)
3
4 # Drop the 'county_name' and 'state' columns
5 ri.drop(['county_name', 'state'], axis='columns', inplace=True)
6
7 # Examine the shape of the DataFrame (again)
8 print(ri.shape)
```

The output of the script is displayed below the code editor:

```
state      0
stop_date  0
stop_time  0
county_name 91741
arrest_gender  5202
arrest_race  5202
citation_row  5202
citation     5202
search_conducted  0
search_type  81624
stop_outcome  5202
is_arrested  5202
stop_duration  5202
stop_related_stop  0
district      0
https://data
```

Task Description

1. Examine the DataFrame's shape to find out the number of rows and columns.
2. Drop both the 'county_name' and 'state' columns by passing the column names to the `.drop()` method as a list of strings.
3. Examine the shape again to verify that there are now two fewer columns.

Code Solution

```
# Examine the shape of the DataFrame
print(ri.shape)
```

```
# Drop the 'county_name' and 'state' columns
ri.drop(['county_name', 'state'], axis='columns', inplace=True)
```

```
# Examine the shape of the DataFrame (again)
print(ri.shape)
```

Code Explanation

1. The line `'print(ri.shape)'` prints the shape of the DataFrame. This outputs the number of rows and columns as a tuple (rows, columns).
2. The line `'ri.drop(['county_name', 'state'], axis='columns', inplace=True)'` drops the specified columns from the DataFrame. The 'axis' parameter is set

to 'columns' to indicate column-wise operation, and 'inplace=True' makes the change directly to the DataFrame.

3. The line 'print(ri.shape)' is used again to confirm that the DataFrame now has two fewer columns, indicating the columns were successfully removed.