

Setting the Index

The screenshot shows a web browser window with multiple tabs. The active tab is titled 'Setting the Index' and shows a DataCamp exercise page. The page has a left sidebar with the exercise title and instructions, and a main area with a code editor and a Python shell. The instructions on the left state: 'The first step that you'll take in this chapter is to set the stop_datetime column as the DataFrame's index. By replacing the default index with a DateTimeIndex, you'll make it easier to analyze the dataset by date and time, which will come in handy later in the course!'. The code editor on the right contains the following Python code:

```
1 # Set 'stop_datetime' as the index
2 ri.set_index('stop_datetime', inplace=True)
3
4 # Examine the index
5 print(ri.index)
6
7 # Examine the columns
8 print(ri.columns)
```

Below the code editor is a Python shell with the output: `In [1]:`. The bottom of the browser window shows a Windows taskbar with various icons and the system clock displaying 14:01 on 08/12/2024.

Task Description

1. Set 'stop_datetime' as the DataFrame index.
2. Examine the index to verify that it is a DateTimeIndex.
3. Examine the DataFrame columns to confirm that 'stop_datetime' is no longer one of the columns.

Code Solution

```
# Set 'stop_datetime' as the index
ri.set_index('stop_datetime', inplace=True)
```

```
# Examine the index
print(ri.index)
```

```
# Examine the columns
print(ri.columns)
```

Code Explanation

1. The line 'ri.set_index('stop_datetime', inplace=True)' sets the 'stop_datetime' column as the index of the DataFrame. This replaces the default integer index, enabling easier analysis by date and time.
2. The line 'print(ri.index)' prints the current index of the DataFrame, confirming that it is now a DateTimeIndex.

3. The line `print(ri.columns)` prints the list of columns in the DataFrame, verifying that `'stop_datetime'` is no longer included in the list of columns since it has been moved to the index.