

Sorting Rows in a DataFrame

This document includes the question, the solution, and a breakdown of the code provided in the screenshot.

Uploaded Screenshot

Below is the screenshot of the task:

The screenshot shows a web browser window displaying a DataCamp exercise page. The page title is 'Sorting rows'. The instructions state: 'Finding interesting bits of data in a DataFrame is often easier if you change the order of the rows. You can sort the rows by passing a column name to .sort_values(). In cases where rows have the same value (this is common if you sort on a categorical variable), you may wish to break the ties by sorting on another column. You can sort on multiple columns in this way by passing a list of column names.' A table shows the syntax for sorting: one column (df.sort_values("breed")) and multiple columns (df.sort_values(["breed", "weight_kg"])). The exercise instructions are numbered 1 to 3. A code editor on the right shows the following code:

```
1 # Sort homelessness by individuals
2 homelessness_ind = ____
3
4 # Print the top few rows
5 print(____)
```

Question

Sort the homelessness DataFrame by the number of individuals, from smallest to largest, and save this as `homelessness_ind`. Print the head of the sorted DataFrame.

Answer

```
# Sort homelessness by individuals
homelessness_ind = homelessness.sort_values("individuals")
```

```
# Print the top few rows
print(homelessness_ind.head())
```

Code Explanation

```
# Explanation of the code:
```

1. ``homelessness.sort_values("individuals")``: Sorts the ``homelessness`` DataFrame by the ``individuals`` column in ascending order by default.
2. ``homelessness_ind``: Stores the sorted DataFrame for further use.
3. ``print(homelessness_ind.head())``: Displays the first five rows of the sorted DataFrame to verify the sorting.