

Subsetting Rows by Categorical Variables

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 displays a DataCamp exercise interface. On the left, the exercise title is "Subsetting rows by categorical variables". The instructions state: "Subsetting data based on a categorical variable often involves using the 'or' operator (|) to select rows from multiple categories. This can get tedious when you want all states in one of three different regions, for example. Instead, use the .isin() method, which will allow you to tackle this problem by writing one condition instead of three separate ones." Below the instructions, there is a code editor with the following code:

```
colors = ["brown", "black", "tan"]
condition = dogs["color"].isin(colors)
dogs[condition]
```

The instructions also mention: "homelessness is available and pandas is loaded as pd." and "Filter homelessness for cases where the USA census state is in the list of Mojave states, canu, assigning to mojave_homelessness. View the printed result." There is a "Take Hint (-30 XP)" button.

On the right, a code editor shows the solution code:

```
1 # The Mojave Desert states
2 canu = ["California", "Arizona", "Nevada", "Utah"]
3
4 # Filter for rows in the Mojave Desert states
5 mojave_homelessness = homelessness[homelessness["state"].isin(canu)]
6
7 # See the result
8 print(mojave_homelessness)
```

At the bottom of the screenshot, a message states: "We closed your session due to inactivity. If the problem persists, please report an issue. Restart Session".

Question

Filter `homelessness` for cases where the USA census `state` is in the list of Mojave states, `canu`, assigning the result to `mojave_homelessness`. View the printed result.

Answer

```
# The Mojave Desert states
canu = ["California", "Arizona", "Nevada", "Utah"]
```

```
# Filter for rows in the Mojave Desert states
mojave_homelessness = homelessness[homelessness["state"].isin(canu)]
```

```
# See the result
print(mojave_homelessness)
```

Code Explanation

Explanation of the code:

1. ``canu = ["California", "Arizona", "Nevada", "Utah"]``: Creates a list of Mojave states to be used for filtering.
2. ``homelessness["state"].isin(canu)``: Checks if each value in the ``state`` column is present in the ``canu`` list, creating a boolean mask.
3. ``homelessness[homelessness["state"].isin(canu)]``: Filters the ``homelessness`` DataFrame to include only rows where the mask is ``True``.
4. ``mojave_homelessness``: Stores the filtered DataFrame for further use.
5. ``print(mojave_homelessness)``: Prints the resulting DataFrame to verify the filter.