**Filtering pandas DataFrames**

In previous videos, I gave some examples of how the NumPy array can be useful to do comparison operations and boolean operations on an element-wise basis. Let's now use this knowledge on the Pandas DataFrame.

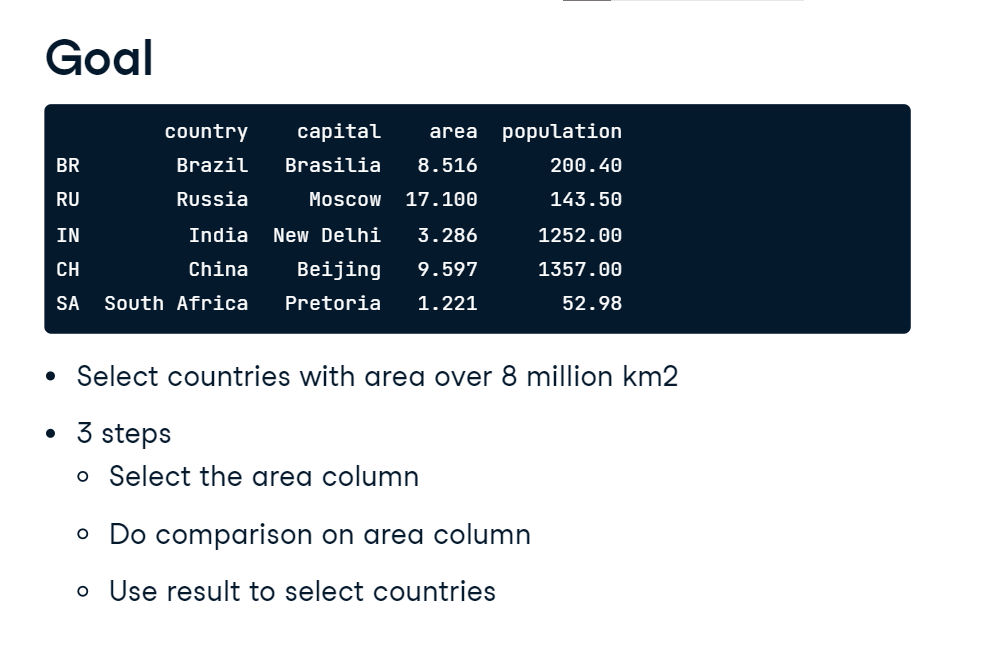
**brics**

00:16 - 00:23

For starters, let's import the BRICS dataset again from the CSV file; here it is.

**Goal**

Suppose you now want to keep the countries, so the observations in this case, for which the area is greater than 8 million square kilometers. There are three steps to this. First of all, we want to get the area column from brics. Next, we perform the comparison on this column and store its result. Finally, we should use this result to do the appropriate selection on the DataFrame.



**Step 1: Get column**

So the first step, getting the area column from brics. There are many different ways to do this. What's important here, is that we ideally get a Pandas Series, not a Pandas DataFrame. Let's do this with square brackets, like this. This loc alternative, and this iloc version, would also work perfectly fine.

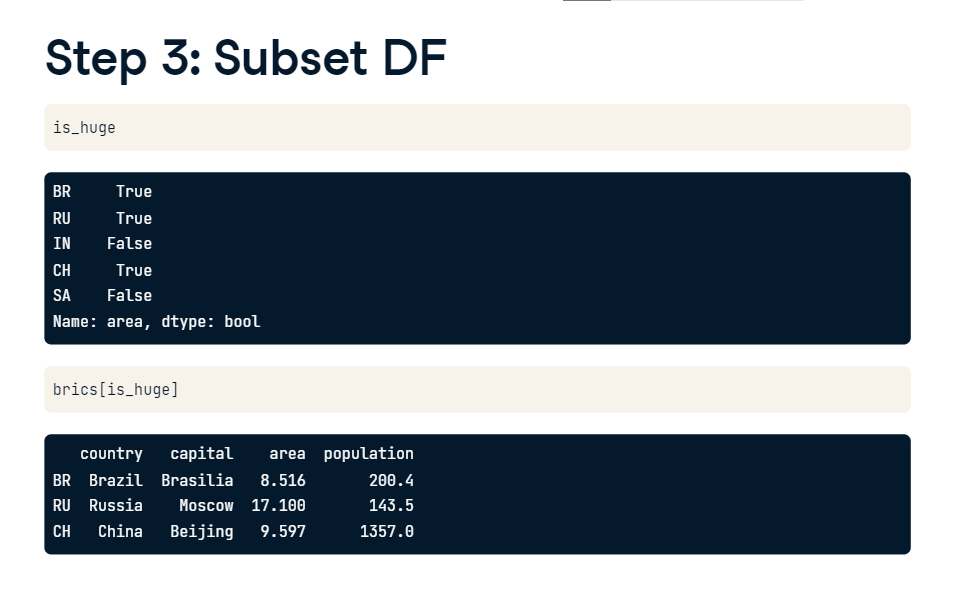


**Step 2: Compare**

Next, we actually perform the comparison. To see which rows have an area greater than 8, we simply append greater than 8 to the code from before, like this. Now we get a Series containing booleans. If you compare it to the actual area values, you can see that the areas with a value over 8 correspond to True, and the ones with a value under 8 correspond to False now. Let me store this Boolean Series as is\_huge.

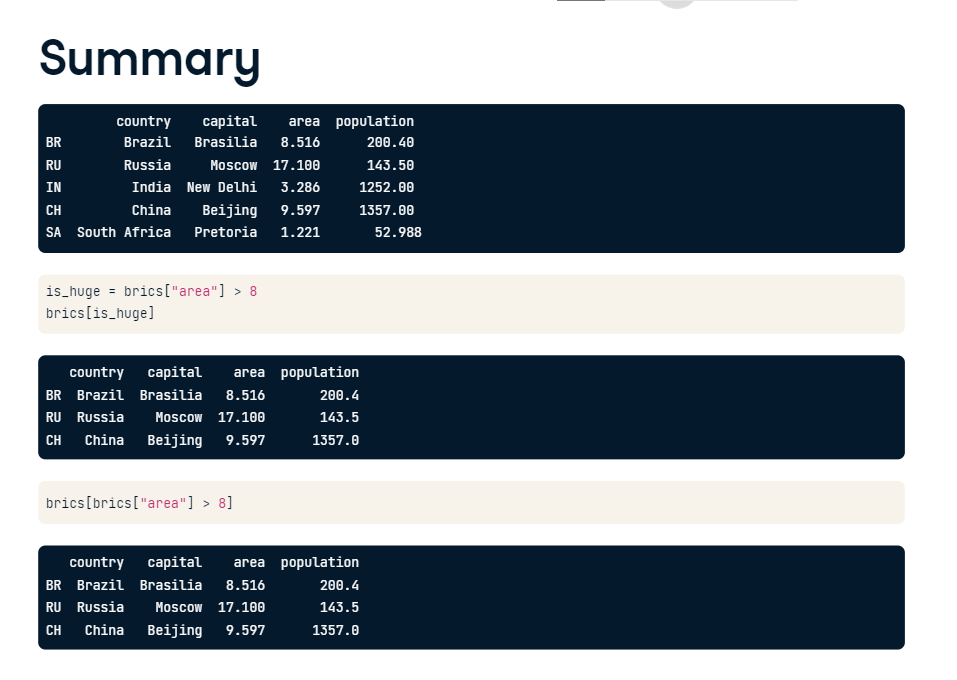


**3: Subset DF**

The final step is using this boolean Series to subset the Pandas DataFrame. This is something I haven't shown you yet. To do this, you put is\_huge inside square brackets. The result is exactly what we want: only the countries with an area greater than 8, namely Brazil, Russia and China. 

**Summary**

So let's summarize this: I selected the area column, performed a comparison on this column and the stored it as is\_huge so that I can use it to index the brics DataFrame. These different commands do the trick. However, we can also write this in a one-liner: simply put the code that defines is\_huge directly in the square brackets. Great!



**Boolean operators**

Now we haven't used boolean operators yet. Remember that we used this logical\_and function from the NumPy package to do an element wise boolean operation on NumPy arrays? Because Pandas is built on NumPy, you can also use that function here. Suppose you only want to keep the observations that have an area between 8 and 10 million square kilometers. After importing numpy as np, we can use the logical\_and() function to create a Boolean Series. The only thing left to do is placing this code inside square brackets to subset brics appropriately. This time, only Brazil and China are included. Russia has an area of 17 million square kilometers, which doesn't meet the conditions. I hope these examples have shown you how easy it is to filter DataFrames to get interesting results.

