

Subsetting rows with .query() - Step 4

The screenshot shows a Jupyter Notebook titled "Subsetting rows with .query()". The left sidebar contains instructions for the exercise, which involves merging GDP and population data for Australia and Sweden, calculating GDP per capita, pivoting the data by date and country, and then using the .query() method to filter for dates on or after 1991-01-01. The main area of the notebook contains the corresponding Python code. The bottom output area displays the result of the .query() operation as a table with columns 'country', 'date', 'gdp', and 'pop'.

```
1 # Merge gdp and pop on date and country with fill
2 gdp_pop = pd.merge_ordered(gdp, pop, on=['country', 'date'], fill_method='ffill')
3
4 # Add a column named gdp_per_capita to gdp_pop that divides the gdp by pop
5 gdp_pop['gdp_per_capita'] = gdp_pop['gdp'] / gdp_pop['pop']
6
7 # Pivot data so gdp_per_capita, where index is date and columns is country
8 gdp_pivot = gdp_pop.pivot_table('gdp_per_capita', 'date', 'country')
9
10 # Select dates equal to or greater than 1991-01-01
11 recent_gdp_pop = gdp_pivot.query('date >= "1991-01-01"')
12
13 # Plot recent_gdp_pop
14 recent_gdp_pop.plot(rot=90)
15 plt.show()
```

country	Australia	Sweden
date		
1980-01-01	0.009	0.009
1980-04-01	0.009	0.009
1980-07-01	0.009	0.009
1980-10-01	0.009	0.009
1981-01-01	0.009	0.009
1981-04-01	0.009	0.009
1981-07-01	0.009	0.009
1981-10-01	0.009	0.009
1982-01-01	0.009	0.009
1982-04-01	0.009	0.009
1982-07-01	0.009	0.009
1982-10-01	0.009	0.009
1983-01-01	0.009	0.009
1983-04-01	0.009	0.009
1983-07-01	0.009	0.009
1983-10-01	0.011	0.009

Question:

Use .query() to select rows from gdp_pivot where date is greater than or equal to '1991-01-01'. Save as recent_gdp_pop and plot.

Answer:

```
# Merge gdp and pop on date and country with fill
gdp_pop = pd.merge_ordered(
    gdp, pop,
    on=['country', 'date'],
    fill_method='ffill'
)
```

```
# Add a column named gdp_per_capita to gdp_pop that divides the gdp by
pop
gdp_pop['gdp_per_capita'] = gdp_pop['gdp'] / gdp_pop['pop']
```

```
# Pivot table of gdp_per_capita, where index is date and columns is country
gdp_pivot = gdp_pop.pivot_table(
    values='gdp_per_capita',
    index='date',
    columns='country'
)
```

```
# Use .query() to select rows where date >= '1991-01-01'  
recent_gdp_pop = gdp_pivot.query('date >= "1991-01-01"')
```

```
# Plot recent_gdp_pop  
recent_gdp_pop.plot(  
    rot=90  
)  
plt.show()
```