Logic, Control Flow and Filtering

Tuesday, October 6, 2020 11:10 AM

Logic, Control Flow and Filtering

Datacamp

Comparison
Make sure you make comparison bw objects with the same types (i.e. int < int, str - str)

- Exceptions: floats-int & numpy array-int

For comparison between numpy arrays and int, it figures out you want to compare each item in the array with the integer and returns the corresponding boolean expression.

```
bmi
```

```
array([21.852, 20.975, 21.75 , 24.747, 21.441])
```

```
bmi > 23
```

```
array([False, False, False, True, False], dtype=bool)
```

- Numpy built an array with the same size filled with the number 23 and compares each array item

Array Comparison
The operational operators like < and >= worked with Numpy arrays out of the box.
Unfortunately, this is not true for the boolean operators and, or, and not.

- `logical_and()`
- `logical_or()`
- `logical_not()`

```
np.logical_and(bmi > 21, bmi < 22)
```

```
array([True, False, True, False, True], dtype=bool)
```

```
bmi[np.logical_and(bmi > 21, bmi < 22)]
```

```
array([21.852, 21.75, 21.441])
```

Conditional Statement
Control scripts can help dictate the flow of your code

```
if condition :
    expression
else :
    expression
```

`control.py`

```
z = 5
if z % 2 == 0 :    # False
    print("z is even")
else :
    print("z is odd")
```

```
z is odd
```

Dataframe
Put it all together and filter data out with a condition

```
         country    capital    area  population
BR        Brazil   Brasilia   8.516      200.40
RU        Russia     Moscow  17.100      143.50
IN         India  New Delhi   3.286     1252.00
CH         China    Beijing   9.597     1357.00
SA  South Africa   Pretoria   1.221       52.98
```

- Select countries with area over 8 million km2
- 3 steps
  - Select the area column
  - Do comparison on area column
  - Use result to select countries

- You can use a boolean series to subset a pandas dataframe: giving you a table of True/False with the same key pairings
  - is_huge stores the series of the True/False statements
  - Numpy might implement brics[is_huge] in a way that only takes True statements

Simplifying it would look like this:

```
         country    capital    area  population
BR        Brazil   Brasilia   8.516      200.40
RU        Russia     Moscow  17.100      143.50
IN         India  New Delhi   3.286     1252.00
CH         China    Beijing   9.597     1357.00
SA  South Africa   Pretoria   1.221       52.988
```

```
is_huge = brics["area"] > 8
brics[is_huge]
```

```
   country   capital    area  population
BR  Brazil  Brasilia   8.516       200.4
RU  Russia    Moscow  17.100       143.5
CH   China   Beijing   9.597      1357.0
```

```
brics[brics["area"] > 8]
```

```
   country   capital    area  population
BR  Brazil  Brasilia   8.516       200.4
RU  Russia    Moscow  17.100       143.5
CH   China   Beijing   9.597      1357.0
```

Using numpy's boolean operators (i.e. logical_and) to specify the filter even further

Jupyter Notebook

Filtering Dataframes by condition
# Import cars data
import pandas as pd
cars = pd.read_csv('cars.csv', index_col = 0)

# Extract drives_right column as Series: dr
dr = cars["drives_right"]
Extract the drives_right column as a Pandas Series and store it as dr.

# Use dr to subset cars: sel
sel = cars[dr]
Use dr, a boolean Series, to subset the cars DataFrame. Store the resulting selection in sel.

# Print sel
print(sel)
Print sel, and assert that drives_right is True for all observations.

```
# Import cars data
import pandas as pd
cars = pd.read_csv('cars.csv', index_col = 0)

# Import numpy, you'll need this
import numpy as np

# Create medium: observations with cars_per_cap between 100 and 500
cpc = cars['cars_per_cap']
between = np.logical_and(cpc > 100, cpc < 500)
medium = cars[between]

# Print medium
print(medium)
```

```
country    capital    area  population
BR         Brazil   Brasilia   8.516      200.40
RU         Russia     Moscow  17.100      143.50
IN          India  New Delhi   3.286     1252.60
CH          China    Beijing   9.597     1357.80
SA  South Africa   Pretoria   1.221       52.98
```

```
import numpy as np
np.logical_and(brics['area'] > 8, brics['area'] < 10)
```

```
BR     True
RU    False
IN    False
CH     True
SA    False
Name: area, dtype: bool
```

```
brics[np.logical_and(brics['area'] > 8, brics['area'] < 10)]
```

```
    country   capital   area  population
BR  Brazil  Brasilia  8.516       200.4
CH   China   Beijing  9.597      1357.8
```