ED5340 - Data Science: Theory and Practise

L7 - Set container

Ramanathan Muthuganapathy (https://ed.iitm.ac.in/~raman)
Course web page: https://ed.iitm.ac.in/~raman/datascience.html
Moodle page: Available at https://courses.iitm.ac.in/

Set representation

- Set is a collection of dissimilar data types using curly brackets { }
  - set1 = {10, 20, 300, 400, 50}; set2 = {‘rat’, ‘cat’, ‘bat’, ‘lion’, ‘tiger’, ‘crocodile’}
  - set3 = {10.5, 22, ‘Antelope’, ‘rabbit’, 456789, 1, 1, 2, 2, 4, 89.9}
  - set4 = {100} # No need of comma after a single element.
  - set= set( ) #What is this?

Set basics and accessing

- Very similar representation to that of a ‘mathematical’ set.
- Set elements have to be unique i.e. cannot be repeated.
- It is an unordered collection - cannot be indexed and sliced.
- An entire set or each element can be printed.
- A Set is an `iterable’ i.e. you can iterate over its elements.
- Given a list, tuple or sting, they can be converted using set( ) function.

Demo using L7_set_ex_access.py

Set operations

- Still, sets are ‘mutable’ (you can add elements to the existing set).
- Immutable sets are possible using frozenset( )
- Two sets cannot be concatenated.
- searching (containment) and sorting - in, not in
- conversion / other functions - len, max, min, sum, sorted

Set methods - Member functions in the set

- Given a set S, you can apply the following member functions using the object (syntax - S.func( )).
  - add - at the end
  - remove - the element
  - discard - what is the difference between remove and discard?
  - clear - clears all the elements

CW: Define a set S and try the member functions in the previous slide

Set methods - Member functions in the list

- Given two sets S and T, you can apply the following member functions.
  - update - adds elements of one list to the other (E.g. S.update(T))
  - issuperset
  - issubset
  - isdisjoint
  - comparison of two sets using relational operators.

Set Varieties

- Set of Sets is not allowed, in general, can be done using frozenset
- Set embedding is not allowed
- Set unpacking (using * operator)

Demo using L7_set_varieties.py

Set Comprehension

same as list comprehension except use curly brackets

The syntax goes like this:

```
setA = {expression for var in sequence [optional for and/or if]}
```

The above is a replacement for the following

```
setA = set( ) #empty set
for var in sequence:
    setA.add(var)
```

CW: Create a set consisting of squares of integers from 1 to 9 using set comprehension

Mathematical set operations

Union, intersection and different

- A | B (union of two sets A and B)
- A & B (intersection of two sets A and B)
- A - B (Difference, Elements in A but not in B)
- B - A (Difference, Elements in B but not in A)
- A ^ B (Symmetric difference - Union of the two differences)of two sets A and B)

Updating set operations

similar to arithmetic expression a += b

- A |= B (update A with A | B)
- A &= B
- A -= B
- B -= A
- A ^= B

Demo using L7_set_math_op.py

HW: Define two sets of random numbers using set comprehension and try out the ‘updating set operations’