Data filtering

Tuesday, May 7, 2019 2:02 PM

Application: data filtering

- It's commonly desirable to remove outliers from data.
- By another mechanism, determine what reasonable values can be.
- Then filter a list, by removing outliers.

Example 1: reject values outside the range of 5 to 15.

```
low = 5
high = 15
weights = [2, 6, -19, 2, 34, 12, 6, 12]
   acceptable = []
   for w in weights:
   if  w >= low and w <= high:
      acceptable.append(w)
   pprint(acceptable)
```

This prints

```
[6, 12, 6, 12]
```

Example 2: complex data filtering

We want a list of exam scores. We know the acceptable scores are greater than or equal to zero, and less than or equal to 100, and they are not None. Filter the scores to eliminate unacceptable values and print the rest.

```
   scores = [None, 100, 90, -40, 20, 60,
              80, None, 25, -70]
   acceptable = []
```

```
    for s in scores:
        if s is not None and s >= 0 and s
        <=100:
            acceptable.append(s)
    pprint(acceptable)
```

This prints:

```
    [100, 90, 20, 60, 80, 25]
```

Very subtle

In the "if" statement above, order is important.

If the first statement "`s is not None`" is false, then the other statements will fail to execute. One can't compare None to anything. This stops execution.

Fortunately, we know that for x and y to be True, both must be true.

So "`and`" stops execution if x is False!

This is called "*short-circuit 'and' logic.*"

In like manner,

for "`x or y`" to be true, either must be true. If x is True, we don't need to compute y.

This is called "*short-circuit 'or' logic*"

Advanced: list comprehensions

There is a simple, if non-intuitive, syntax for doing the above, called a *list comprehension*.

Example 1 above:

```
    acceptable = [x for x in weights
```

```
                    if x >= 5 and x <= 15]
```

Example 2 above:

```
   acceptable = [x for x in weights
                 if x is not None and
                 x >= 0 and x <= 100]
```

These do exactly the same things as the above code.

Some programmers love them.

Others hate them.

What do you think?