190F Foundations of Data Science

Fall 2018

Lecture 10

Iteration

Announcements

Comparison

Comparison Operators

The result of a comparison expression is a `bool` value

```
x = 2                y = 3
```

Assignment statements

```
x > 1                x > y            y >= 3


x == y               x != 2           2 < x < 5
```

Comparison expressions

(Demo)

Combining Comparisons

Boolean operators can be applied to `bool` values

```
a = True        b = False
```

Evaluate to True

```
not b           a or b            a and not b
```

```
a and b         not (a or b)      b and b
```

Evaluate to False

(Demo)

Aggregating Comparisons

Summing an array or list of bool values will count the True values only.

```
1    + 0     + 1          == 2
True + False + True       == 2
sum([1   , 0    , 1   ))  == 2
sum([True, False, True))  == 2
```

(Demo)

Conditional Statements

These statements *control* the sequence of computations that are performed in a program

- The keyword **`if`** begins a control statement
- The purpose of **`if`** is to define functions that choose different behavior based on their arguments
- **`if`** statements use comparisons to choose between different possible behaviors.

(Demo)

Random Selection

Random Selection

`np.random.choice`

- Selects at random
- with replacement
- from an array
- a specified number of times

```
np.random.choice(some_array, sample_size)
```

(Demo)

Discussion Question

```
d = np.arange(6) + 1
```

What results from evaluating the following 2 expressions? Are they the same? Do they describe the same process?

```
np.random.choice(d, 1000) + np.random.choice(d, 1000)
```

```
2 * np.random.choice(d, 1000)
```

Control Statements

More Control Statements

These statements *control* the sequence of computations that are performed in a program

- The purpose of `for` is to perform a computation for every element in a list or array

(Demo)