Datasets and Boolean Indexing: Takeaways

Syntax

- Loading a CSV file:

```
data = np.genfromtxt('filename.csv', delimiter=',')
```

COMPARISON OPERATORS

- The comparison operations in NumPy perform the comparison element by element.

```
x < y
x <= y
x == y
x != y
x > y
x >= y
```

- Broadcasting with comparison operators:

```
leq_3 = x <= 3
```

LOGICAL OPERATORS

- With NumPy we use **&** for **and**, **|** for **or** and **~** for **not**:

```
(x >= 2) & (x <= 3)
(x < 2) | (x > 3)
~(x == 0)
```

BOOLEAN MASKS

- Boolean masks:

```
x[x > 3]
x[(x >= 2) & (x <= 3)]
```

- Column or row selection:

```
row_mask = np.array([False, True, True])
x[row_mask,:]
```

Concepts

- Even though we can read CSVs with it, NumPy is not suited to handle non-numerical data.
- We can read column names using NumPy, but then we lose the benefits of working with ndarrays.

- Using masks, we can select parts of the data that verify certain conditions. Boolean masks can also be used to select a subset of rows or columns. These can be combined with array slicing. When combining boolean operators with comparison operators, we need to use parenthesis to surround the conditions.

Resources

- Reading a CSV file with NumPy
- Indexing and selecting data