Introduction to NumPy: Takeaways

Syntax

SELECTING ROWS, COLUMNS, AND ITEMS FROM AN NDARRAY

- Convert a list of lists into a ndarray:

```
import numpy as np

f = open("nyc_taxis.csv", "r")

taxi_list = list(csv.reader(f))

taxi = np.array(converted_taxi_list)
```

- Selecting a row from an ndarray:

```
second_row = taxi[1]
```

- Selecting multiple rows from an ndarray:

```
all_but_first_row = taxi[1:]
```

- Selecting a specific item from an ndarray:

```
fifth_row_second_column = taxi[4,1]
```

SLICING VALUES FROM AN NDARRAY

- Selecting a single column:

```
second_column = taxi[:,1]
```

- Selecting multiple columns:

```
second_third_columns = taxi[:,1:3]

cols = [1,3,5]

second_fourth_sixth_columns = taxi[:, cols]
```

- Selecting a 2D slice:

```
twod_slice = taxi[1:4, :3]
```

VECTOR MATH

- `vector_a + vector_b` - Addition
- `vector_a - vector_b` - Subtraction
- `vector_a * vector_b` - Multiplication (this is unrelated to the vector multiplication used in linear algebra).
- `vector_a / vector_b` - Division
- `vector_a % vector_b` - Modulus (find the remainder when `vector_a` is divided by `vector_b`)
- `vector_a ** vector_b` - Exponent (raise `vector_a` to the power of `vector_b`)
- `vector_a // vector_b` - Floor Division (divide `vector_a` by `vector_b`, rounding down to the nearest integer)

CALCULATING STATISTICS FOR 1D NDARRAYS

- `ndarray.min()` to calculate the minimum value
- `ndarray.max()` to calculate the maximum value
- `ndarray.mean()` to calculate the mean average value
- `ndarray.sum()` to calculate the sum of the values

CALCULATING STATISTICS FOR 2D NDARRAYS

- Max value for an entire 2D Ndarray:

```
taxi.max()
```

- Max value for each row in a 2D Ndarray (returns a 1D Ndarray):

```
taxi.max(axis=1)
```

- Max value for each column in a 2D Ndarray (returns a 1D Ndarray):

```
taxi.max(axis=0)
```

ADDING ROWS AND COLUMNS TO NDARRAYS

- Joining a sequence of arrays:

```
np.concatenate([a1, a2], axis=0)
```

- Expanding the shape of an array:

```
np.expand_dims([1, 2], axis=0)
```

SORTING

- Sorting a 1D Ndarray:

```
np.argsort(taxi[0])
```

- Sorting a 2D NDarray by a specific column:

```
sorted_order = np.argsort(taxi[:,15])

taxi_sorted = taxi[sorted_order]
```

Concepts

- Python is considered a high-level language because we don't have to manually allocate memory or specify how the CPU performs certain operations. A low-level language like C gives us this control and lets us improve specific code performance, but a tradeoff in programmer productivity is made. The NumPy library lets us write code in Python but take advantage of the performance that C offers. One way NumPy makes our code run quickly is **vectorization**, which takes advantage of **Single Instruction Multiple Data (SIMD)** to process data more quickly.
- A list in NumPy is called a 1D Ndarray and a list of lists is called a 2D Ndarray. NumPy ndarrays use indices along both rows and columns and is the primary way we select and slice values.

Resources

- Arithmetic functions from the NumPy documentation.
- NumPy ndarray documentation