Announcements

Homework #1: Due 9/8

Agenda

Finish Lab 1 and start Lab 2 materials

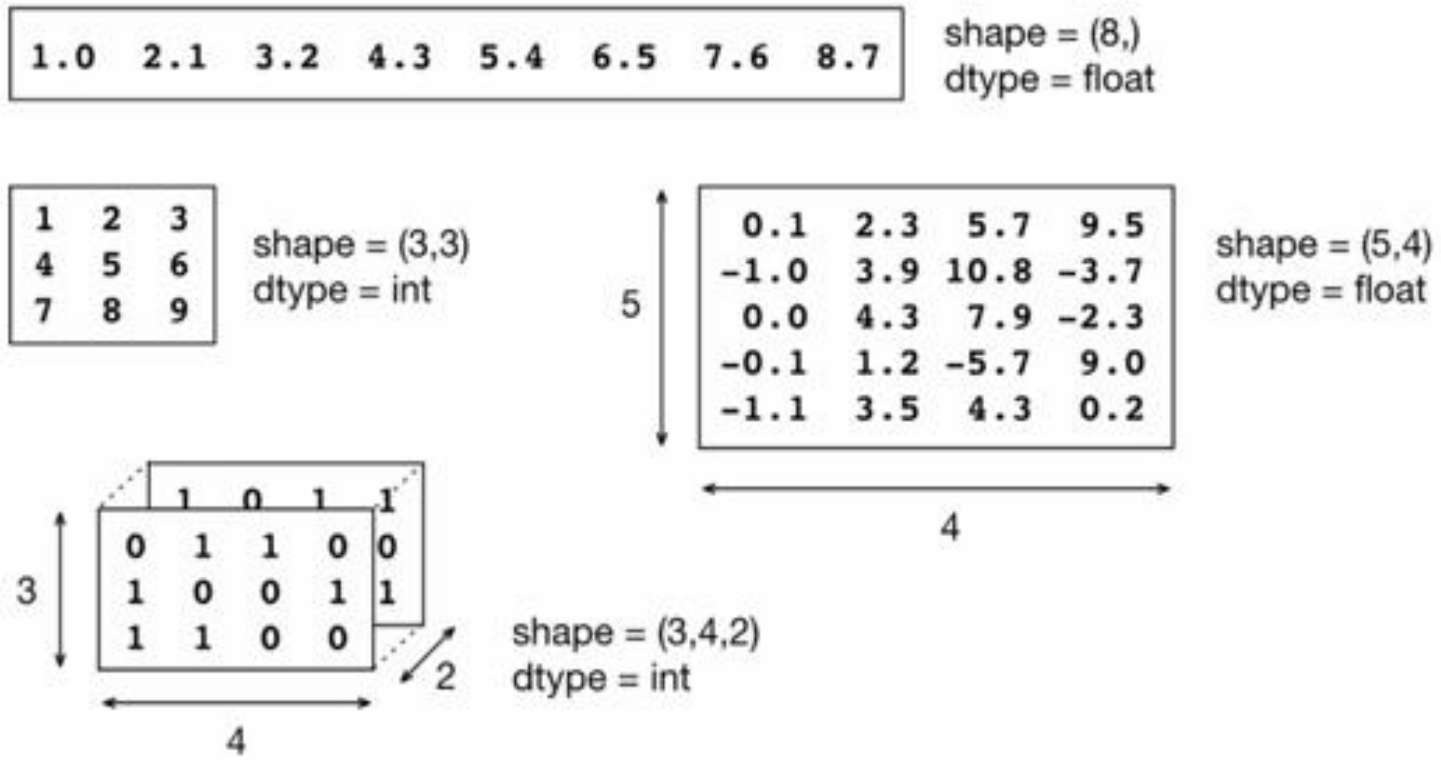
Numpy Arrays:
Similar in some ways to Python lists, but they're potentially multidimensional. They can be not just, as in a list, a single dimension of elements, but more than one. But they are homogeneous in type:

Python list, we have a series of elements in a particular order, and the types of those elements can be anything; they can be numbers or strings or other lists or other kinds of objects

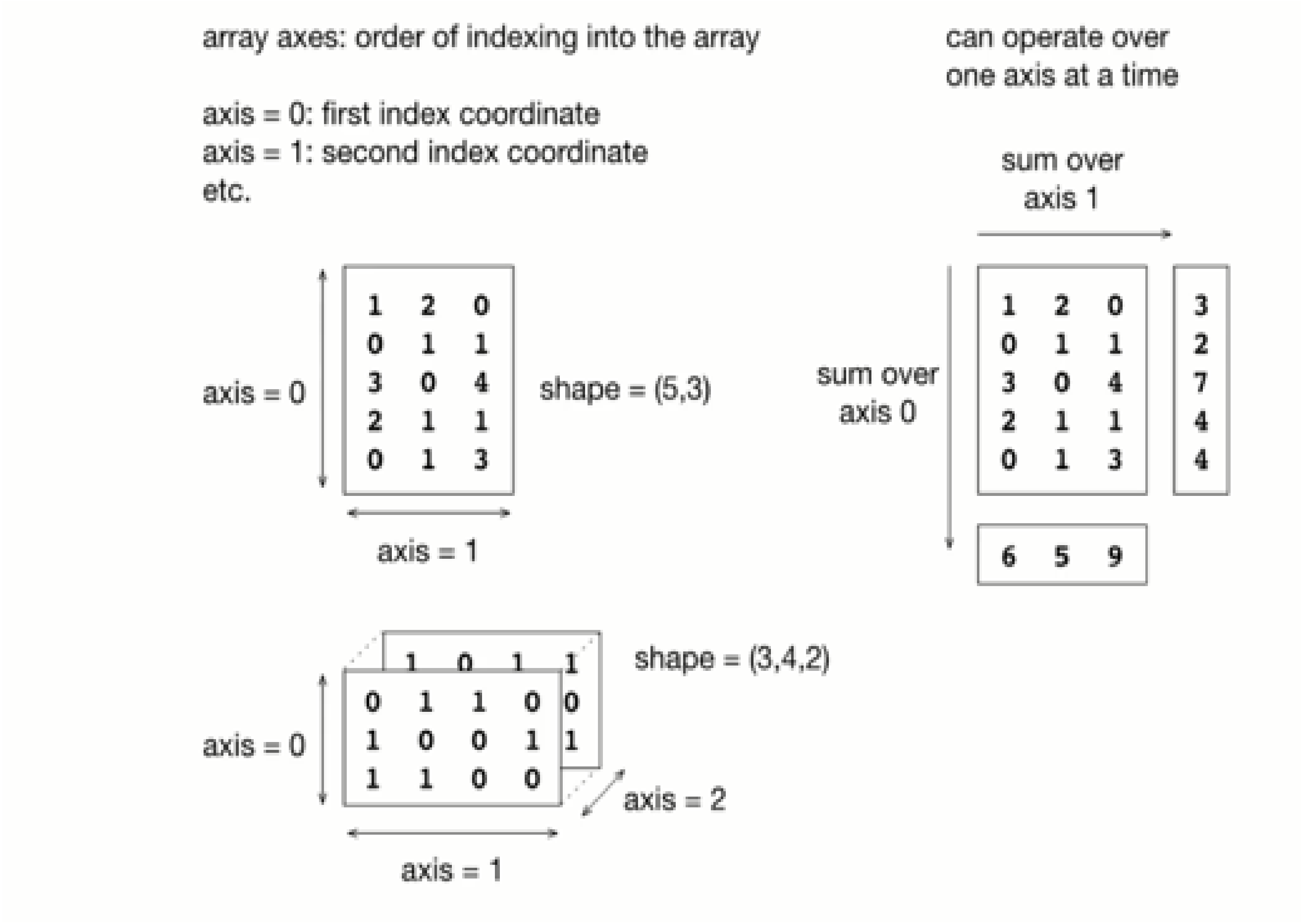
In NumPy arrays, we have a similar sort of positional ordering, but the types are fixed for a given array. So we might have a one-dimensional array of floating point numbers, or floats; might have a 2D array of integers, or ints; or a 2D array of floats;

array shape: tuple of integers describing the number of elements in each dimension

array dtype: datatype of array elements



Numpy Arrays



Definition and Usage

The `sort()` method sorts the list ascending by default.

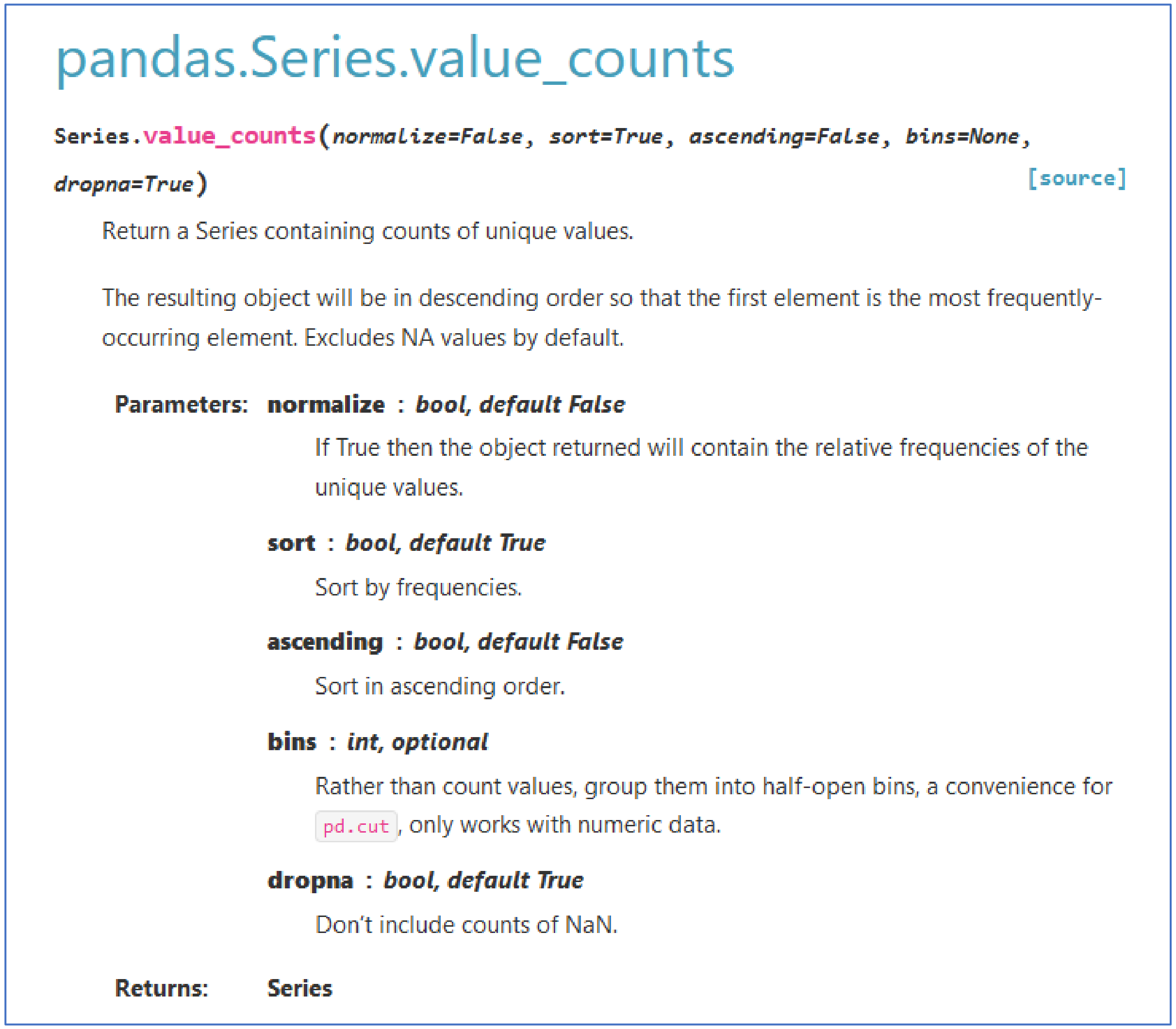
You can also make a function to decide the sorting criteria(s).

Syntax

```
list.sort(reverse=True|False, key=myFunc)
```

Parameter Values

Parameter	Description
reverse	Optional. reverse=True will sort the list descending. Default is reverse=False
key	Optional. A function to specify the sorting criteria(s)



pandas.Series.value_counts

`Series.value_counts(normalize=False, sort=True, ascending=False, bins=None, dropna=True)` [source]

Return a Series containing counts of unique values.

The resulting object will be in descending order so that the first element is the most frequently-occurring element. Excludes NA values by default.

Parameters: **normalize** : ***bool, default False***

If True then the object returned will contain the relative frequencies of the unique values.

sort : ***bool, default True***

Sort by frequencies.

ascending : ***bool, default False***

Sort in ascending order.

bins : ***int, optional***

Rather than count values, group them into half-open bins, a convenience for `pd.cut`, only works with numeric data.

dropna : ***bool, default True***

Don't include counts of NaN.

Returns: **Series**

https://pandas.pydata.org/docs/reference/api/pandas.Series.value_counts.html