UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES

3250 Foundations of Data Science

Module 3: NumPy

Course Plan

Module Titles
Module 1 – Introduction to Data Science
Module 2 – Introduction to Python
Current Focus: Module 3 - NumPy
Module 4 – Pandas
Module 5 – Data Collection and Cleaning
Module 6 – Descriptive Statistics and Visualization
Module 7 – Workshop (No Content)
Module 8 – Time Series
Module 9 – Introduction to Regression and Classification
Module 10 – Databases and SQL
Module 11 – Data Privacy and Security
Module 12 – Term Project Presentations (no content)

Learning Outcomes for this Module

- Further build your Python skills
- Download and share work with GitHub
- Use the NumPy data analysis library to work with large arrays
- Use kNN to classify observations

Topics for this Module

- **3.1** Working with Python
- **3.2** Git and GitHub
- **3.3** NumPy
- **3.4** kNN: k Nearest Neighbors
- **3.5** Resources and Homework

UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES

Module 3 – Section 1

Working with Python

Imports

- Use them to bring in packages you need

- Syntax
  - `import pandas`
  - `import pandas as pd`
  - `from pandas import DataFrame`

Review: For Loops & Ranges

- `for i in range(0, 5):`
- `for c in “this string”:`

Reading & Writing Files

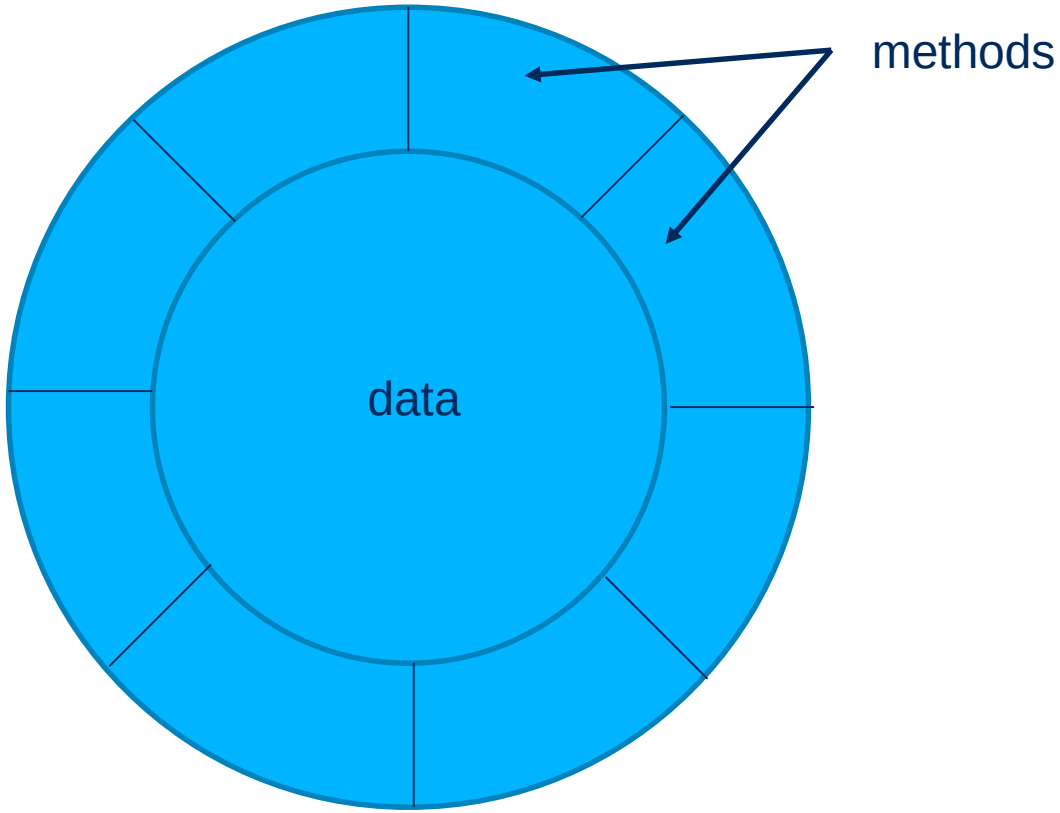
```
f = open('myfile.txt', 'r')
for line in f:
    print(line)
f.close()

fout = open('mynewfile.txt', 'w')
for i in range(2, 17):
    fout.write(str(i)+ '\n')
fout.close()
```

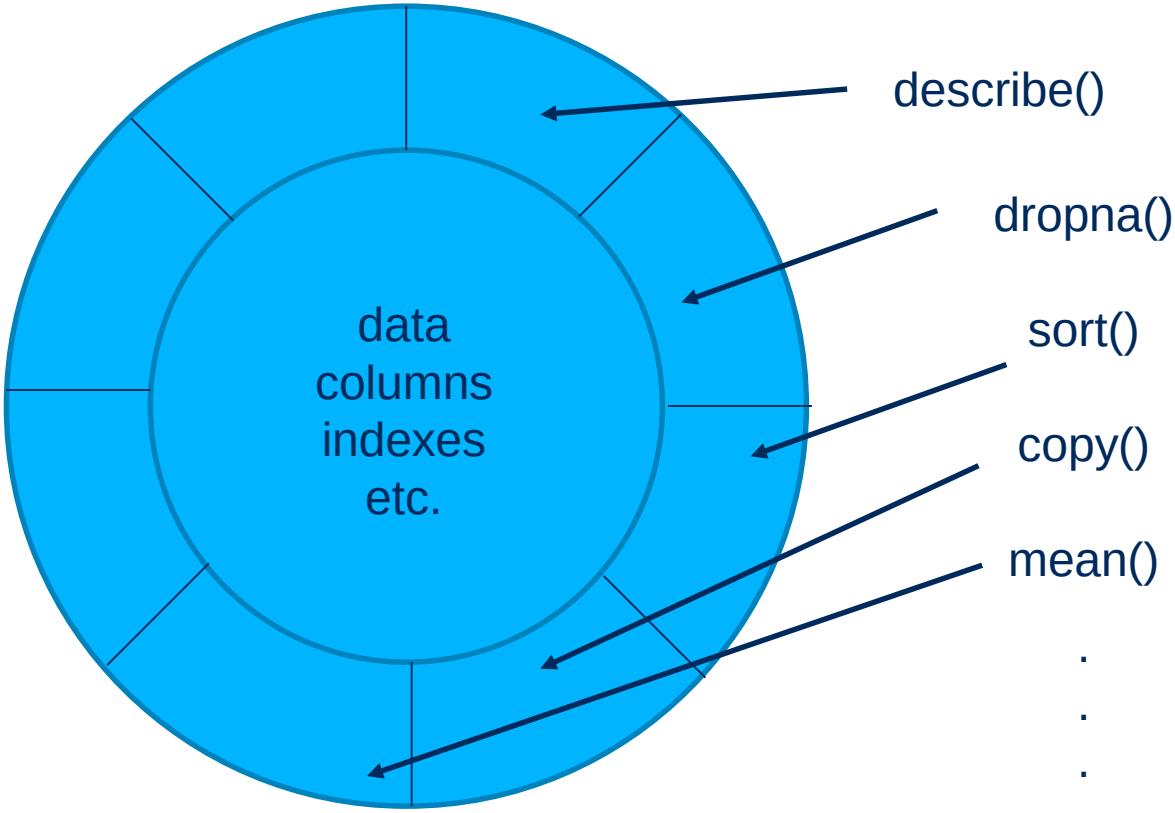
Software Objects

- Characteristics of Python objects:
  - Type
  - Value
  - Identity

- Object Attributes:
  - Data attributes
  - Methods



An Example Software Object: DataFrame



Getting Help in Jupyter

- `?` or `help()` will show some documentation for an object
- `tab-shift` will show methods or parameters

UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES

Module 3 – Section 2

Git and GitHub

Git

- Distributed Version Control System
  - Enables collaborative, distributed development
  - Keeps track of all changes to the code base, who has made the change and what has been changed in the update
  - Allows to revert back to the previous version

- Uses Code Repository
  - Stores the code base
  - Retains a complete copy of the entire project throughout its lifetime
  - Manages the revisions and history of a project

History of Git

- Git was created in 2005 by Linus Torvalds to support the development of the Linux® kernel.
- Linux community needed a version control system that would:
  - Facilitate distributed development
  - Scale to handle thousands of developers
  - Perform quickly and efficiently
  - Maintain integrity and trust
  - Enforce Accountability
  - Support immutability
  - Atomic Transactions
  - Support and encourage branched development
  - Complete repositories
  - Clean Internal Design
  - Be free, as in Freedom

GitHub

- GitHub is a web-hosted Git repository
- Introduces the model of social coding, enables collaborative distributed development
- Allows developers to:
  - Create a repository
  - Work remotely, online and offline
  - Collaborate on independent streams of history
  - Merge the code built by multiple developers
  - Pull, Fork, Clone and Watch

How to Start with GitHub

- Go to GitHub and Sign Up for the free account

- On Demand Training

- Other videos:
  - Git and GitHub for Beginners: GitHub basics, and how to use GitHub Desktop
  - Git & GitHub Crash Course For Beginners

UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES

Module 3 – Section 3

NumPy

Download & Launch NumPy Notebook

- To start the download, go to github.com/wesm/pydata-book
- Click the green “Clone or download” button
- Click the zip option
- Copy the downloaded file “pydata-book-2nd-edition” to your folder for this course and unzip it
- Delete the zip
- Start jupyter notebook, navigate to pydata-book-2nd-edition and launch ch04.ipynb

NumPy: Numerical Python

- Fast, space-efficient n-dimensional array
- Standard math operations on entire arrays without loops
- Linear algebra and random number generation
- Tools for integrating to fast C, C++ and Fortran libraries

NumPy Array

- Adds arrays of more than one dimension to Python
- The entire array must be of a single type, typically numbers
- Has a `shape` e.g. (2, 4, 3) and `size` e.g. 24
- Supports
  - Views
  - Indexing
  - Slicing

Creating an Array

```
import numpy as np
a = np.array([1,2,3])
b = np.array([[1,2,3], [3,4,5]])
```

Indexing

- Dimensions are called axes
- An ndarray can have one index per axis
- Indexes on arrays are integers starting at 0

- Example:

```
a = np.array([[1,2,3],[4,5,6]])
a[0, 2]
a[1, 1]
```

Views

- Most operations on ndarrays do not make copies of the data
  - Simple assignments
  - Passing ndarrays as arguments to a function
  - `.view()` which creates a “window” on part of an ndarray
  - Slicing returns a view

- `.copy()` will make a deep copy

Slicing

- Slicing an array returns a view of it

- Examples:

```
a = np.random.rand(10,10)
a[0:2]
a[0:4, 2:6]
a[:, 2:6]
```

Module 3 – Section 4

k Nearest Neighbors

Intro to kNN

- Let's look at the Fisher iris data set
- And a k-Nearest Neighbors implementation in Python

UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES

Module 3 – Section 5

Resources and Homework

Resources

- NumPy indexing reference
- Jupyter cheat sheet

Next Class

- Pandas
- Assignment 1 is due

Follow us on social

Join the conversation with us online:

facebook.com/uoftscs

@uoftscs

linkedin.com/company/university-of-toronto-school-of-continuing-studies

@uoftscs

Any questions?

UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES

Thank You

Thank you for choosing the University of Toronto School of Continuing Studies