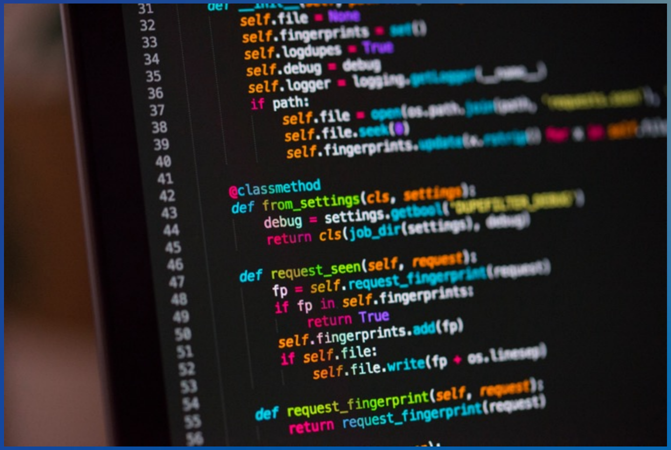
Data Science Upskilling Workshop

Session 2: Programming, Best Practices & Scientific Libraries

Oren Livne and Jiangang Hao
Educational Testing Service

NCME 2023 Training Workshop – 4/12/2023

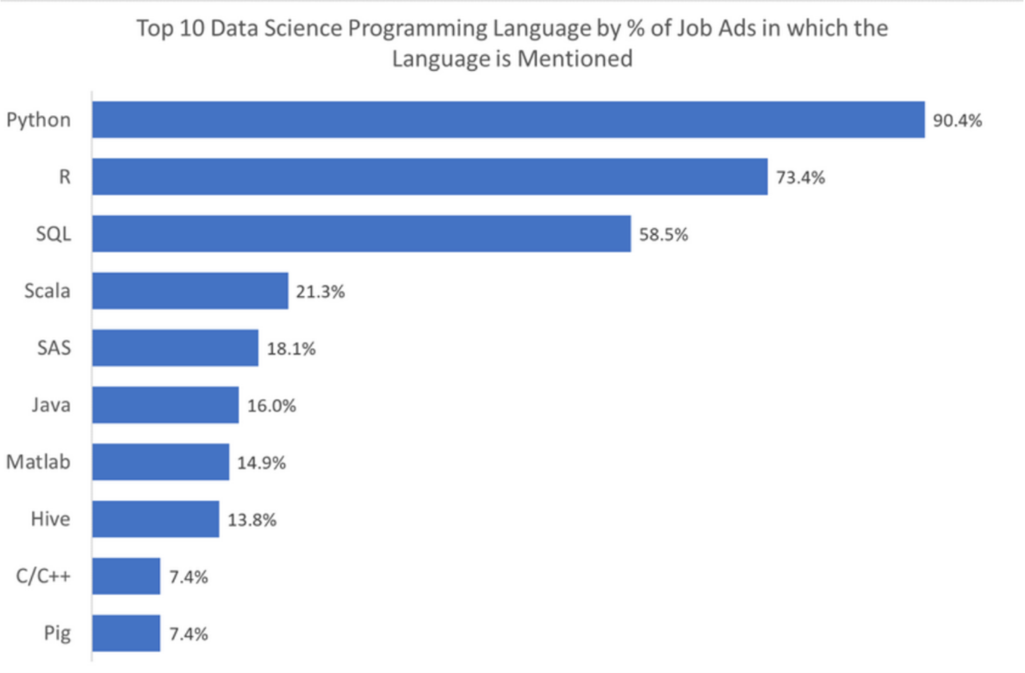
Overview of Python Programming



```
    def __init__(self,
        self.file = None
        self.fingerprints = set()
        self.logdupes = True
        self.debug = debug
        self.logger = logging.getLogger(__name__)
        if path:
            self.file = open(os.path.join(path, 'requests.seen'),
            self.file.seek(0)
            self.fingerprints.update(x.rstrip() for x in self.file

    @classmethod
    def from_settings(cls, settings):
        debug = settings.getbool('DUPEFILTER_DEBUG')
        return cls(job_dir(settings), debug)

    def request_seen(self, request):
        fp = self.request_fingerprint(request)
        if fp in self.fingerprints:
            return True
        self.fingerprints.add(fp)
        if self.file:
            self.file.write(fp + os.linesep)

    def request_fingerprint(self, request):
        return request_fingerprint(request)
```

Why Python for Data Science?

- Easy to learn
- Powerful
- Scalable
- Many libraries
  - Machine Learning
  - Deep Learning
  - Visualization
- Python Community



Overview of the Python Language: 1/2

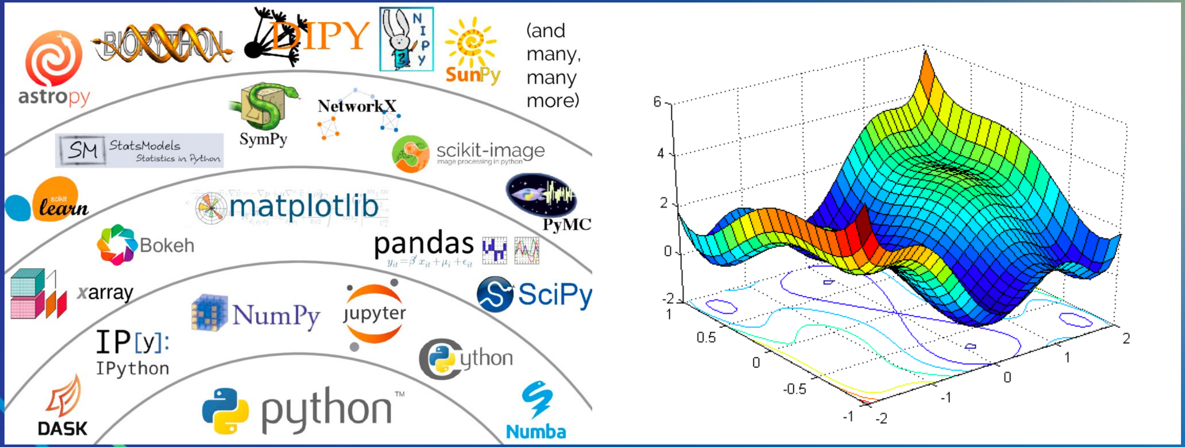
- Data Types
  - None = missing value
- Flow control
  - If-else
  - Loop
  - Function
- Data Structures
  - List []
  - Tuple ()
  - Set {}
  - Dictionary {}
- Printing & formatting
- Importing packages; getting help
- Python Style Guide
- List comprehension: [x**2 for x in range(10)]

Overview of the Python Language: 2/2

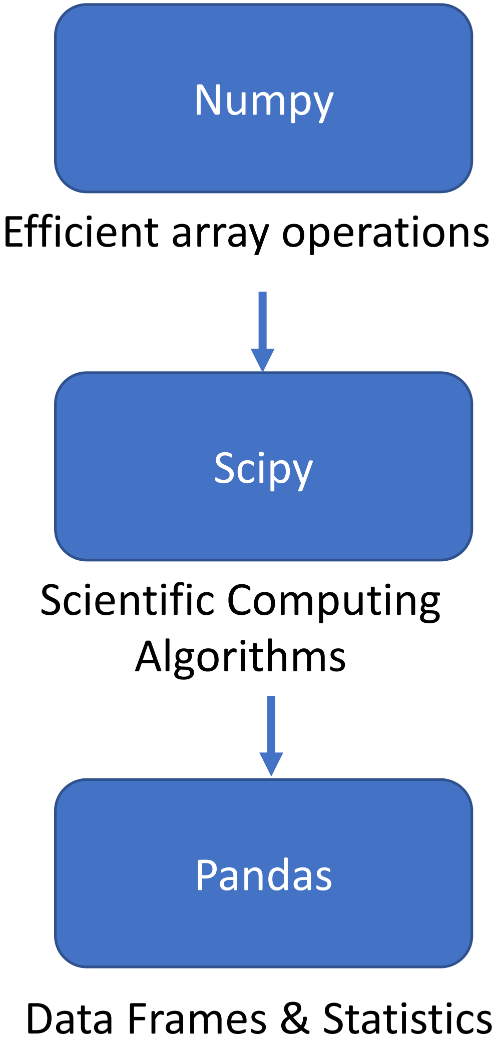
- Recap: data structures, generators
- Index ranges, negative index, str as list
- Filtering lists with list comprehension
- Functional programming: map, reduce, filter
- timeit
- Zip

- Further reading:
  - https://docs.python.org/3/tutorial/

Scientific Computing Packages



Basic Data Science Packages



Numpy

Efficient array operations

Scipy

Scientific Computing Algorithms

Pandas

Data Frames & Statistics

Package Highlights

- **Numpy**: Tutorial Notebook
- **Scipy**: https://scipy-lectures.org/intro/scipy.html
    - scipy.linalg - linear algebra (1.6.3)
    - scipy.curvefit – curve fitting (1.6.5)
    - scipy.stats - statistics and random numbers (1.6.6)
        - Statistical testing
- **Pandas** Tutorial Notebook
    - Series
    - DataFrame
    - Reading a CSV file

- Further reading:
    - https://numpy.org/doc/stable/user/quickstart.html
    - 10 minutes to pandas

Interfacing with other Languages

- Calling an R function from Python
  - Call R for computing -> numpy array
  - Call R -> pandas DataFrame

Session 2 - Exercises

1. Complete numpy tutorial assignments in the bottom of the numpy_tutorial.ipynb notebook under the Session 2 materials on the Teams site.

2. Calculate summary statistics: mean, std, 0%,10%,20%,..., percentiles (look for how to do that in the numpy documentation).

3. Run a t-test of two Gaussian distribution means using scipy (see https://scipy-lectures.org/intro/scipy.html, Sec. 1.6.6.3.