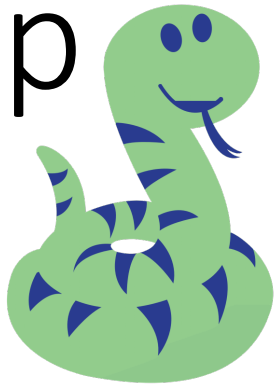
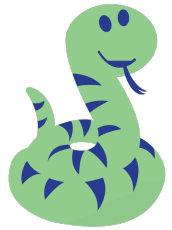
Python Workshop

Crockett Lab, Fall 2020

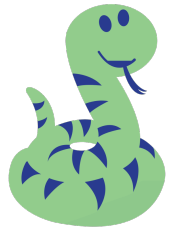


Workshop Topics



- Why code, why Python?
  - Advantages and disadvantages
- What *is* Python?
- Getting up and running with Python
  - Downloading
  - Installing
- Interactive coding, scripting, and notebooks
  - Anaconda
    - Downloading
    - Installing
- The basics
  - Variables, data types, operators, expressions, and statements
  - Functions and methods
    - Abstraction, encapsulation, and object-orientated programming
  - Conditionals and Flow Control
  - Libraries
- Working with data
  - Reading data files
    - Coronavirus dataset
  - Data structures
    - Tuples, lists, and dictionaries
    - Accessing data
  - Working with Pandas
    - Statistical operations
    - Data Visualization

Why Code



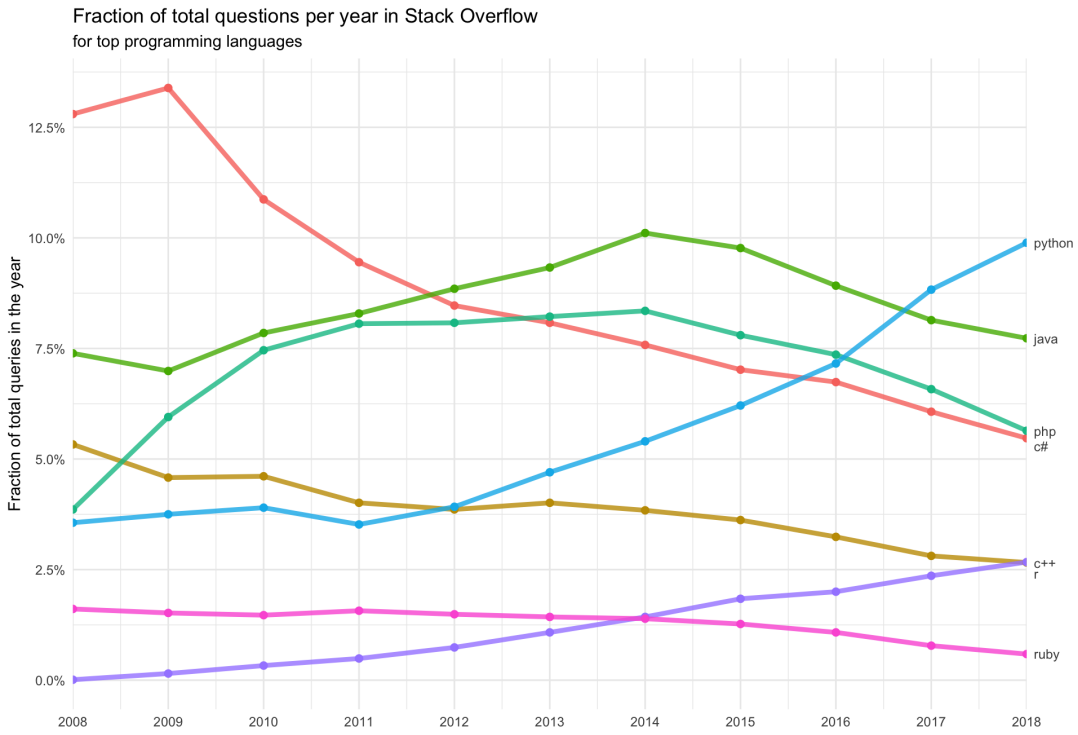
- Why should behavioral scientists code?
  - Why not use consumer products, like SPSS or Jamovi?
    - Open Science
      - Replication and Reproducibility
    - Interdisciplinarity
    - Pedagogy and statistical training
    - Flexibility and novelty
    - Computational methods
    - Transferable skills and ‘market Value’?

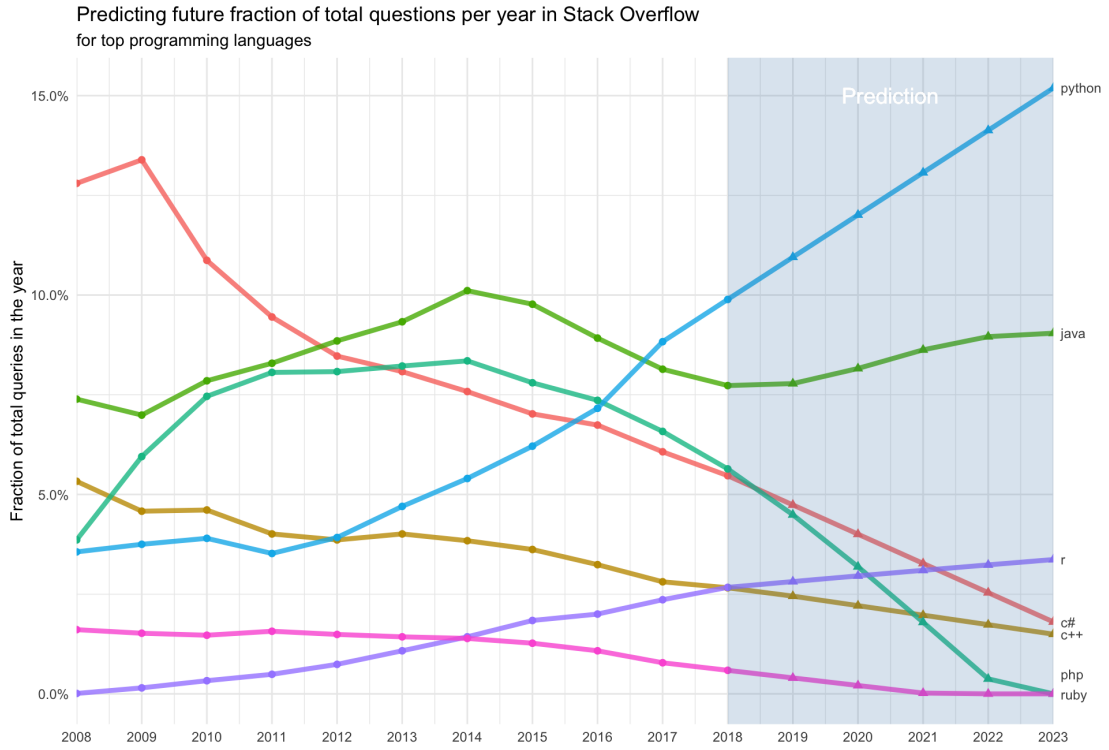
Why code in Python

Worldwide, Jan 2020 compared to a year ago:

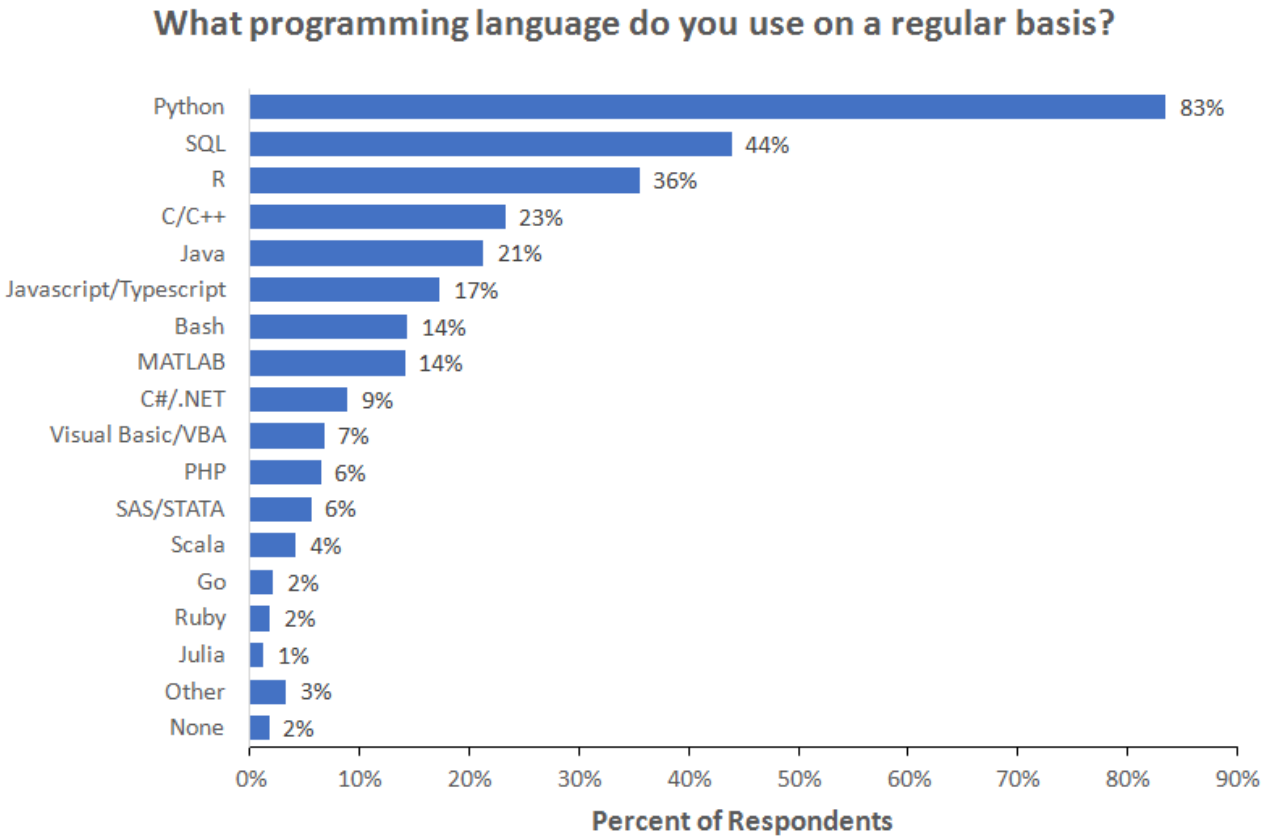
Rank	Change	Language	Share	Trend
1		Python	29.72 %	+4.3 %
2		Java	19.03 %	-1.9 %
3		Javascript	8.2 %	+0.1 %
4		C#	7.28 %	-0.2 %
5		PHP	6.09 %	-1.1 %
6		C/C++	5.91 %	-0.3 %
7		R	3.72 %	-0.2 %
8		Objective-C	2.47 %	-0.6 %
9		Swift	2.36 %	-0.2 %
10		Matlab	1.79 %	-0.2 %

Why code in Python





Why code in Python



Note: Data are from the 2018 Kaggle Machine Learning and Data Science Survey.

Why code in Python

Advantages

- Data science, AI, and machine learning packages
- Quicker execution
- Easy to learn
- Powerful
- Progression

Disadvantages

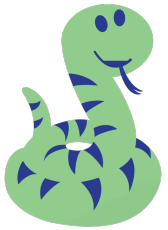
- Fewer libraries for statistics used in the social sciences
  - Beginning to change
- R is probably more common in psychology

What is Python?

- Developed in the 1980s by Guido van Rossum
- Now in it’s third iteration, Python 3
- High-level programming language
  - Built on C
- Functional
- Procedural
- Object-orientated
- Playful
- Easy to understand
- Error handling

Guido van Rossum

Getting Python Running



Python 3.8

https://www.python.org/downloads/

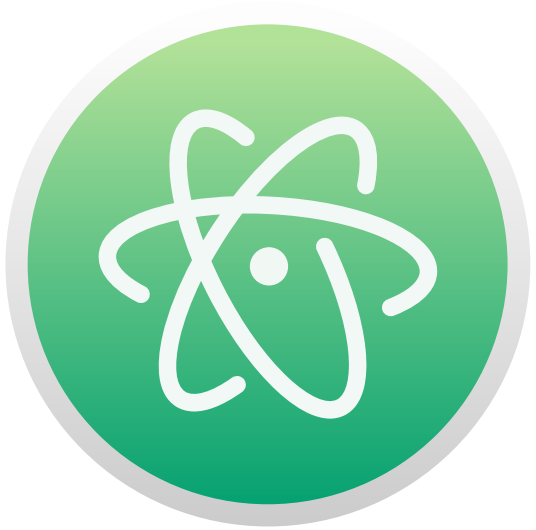
Opening Python

- IDLE and interactive programming

- IDLE, scripting, and procedural programming

Opening Python

Compilers

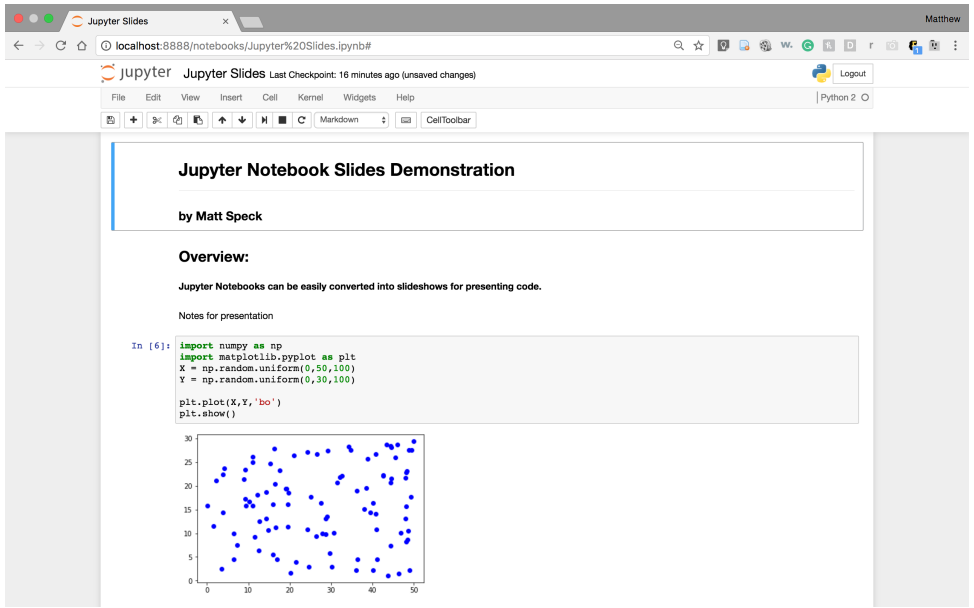


README.md | .travis.yml | python-tool... | python-tool... | tools.py | Settings

```
49              sys.path.insert(0, path)
50
51          script = jedi.api.Script(
52              source=request['source'],
53              line=request['line'] + 1,
54              column=request['col'],
55              path=request.get('path', ''),
56          )
57
58          if request['type'] == 'usages':
59              self._write_response(self._serialize('usages', script.usages()))
60          elif request['type'] == 'gotoDef':
61              self._write_response(self._serialize('gotoDef', script.goto_definitions()))
62
63      def _write_response(self, response):
64          sys.stdout.write(response + '\n')
65          sys.stdout.flush()
66
67      def watch(self):
68          while True:
69              try:
70                  data = self._input.readline()
```

Opening Python

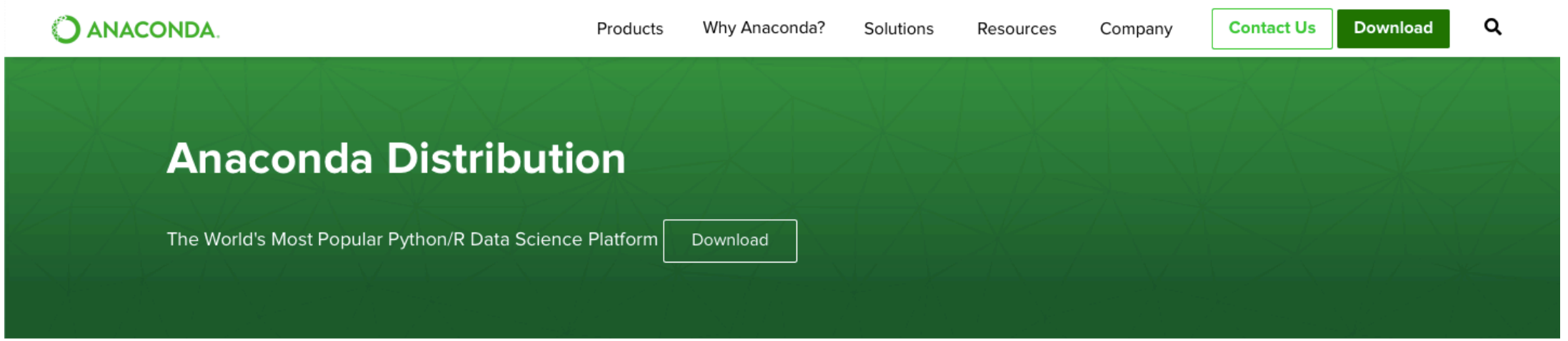
Notebooks



Getting Jupyter Notebooks Running

Anaconda 2019.10

https://www.anaconda.com/distribution/



Workshop Files

GitHub Repository

https://github.com/KillianMcL/PythonSocSci