



Lesson 1: Introduction to Data Science with Python



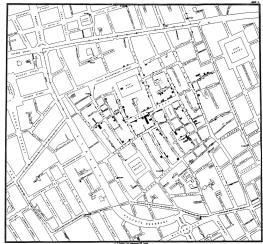
Getting the Materials

- **Resources:** <http://hopelessoptimism.com/data-science-fundamentals>
- **Forum:** <https://gitter.im/hopelessoptimism/data-science-fundamentals>

Getting the Materials

- **Code:** <https://github.com/hopelessoptimism/data-science-fundamentals>
- **Data:**
 - Inside Airbnb: <http://insideairbnb.com/get-the-data.html>
 - Foursquare API: <https://developer.foursquare.com>

A Brief Historical Diversion



Broad Street
Cholera Outbreak
(1854)

John Tukey
(1962)



Data Science
Coined
(1974)

First Conference
(1996)



Building Data Science Teams
(2008)



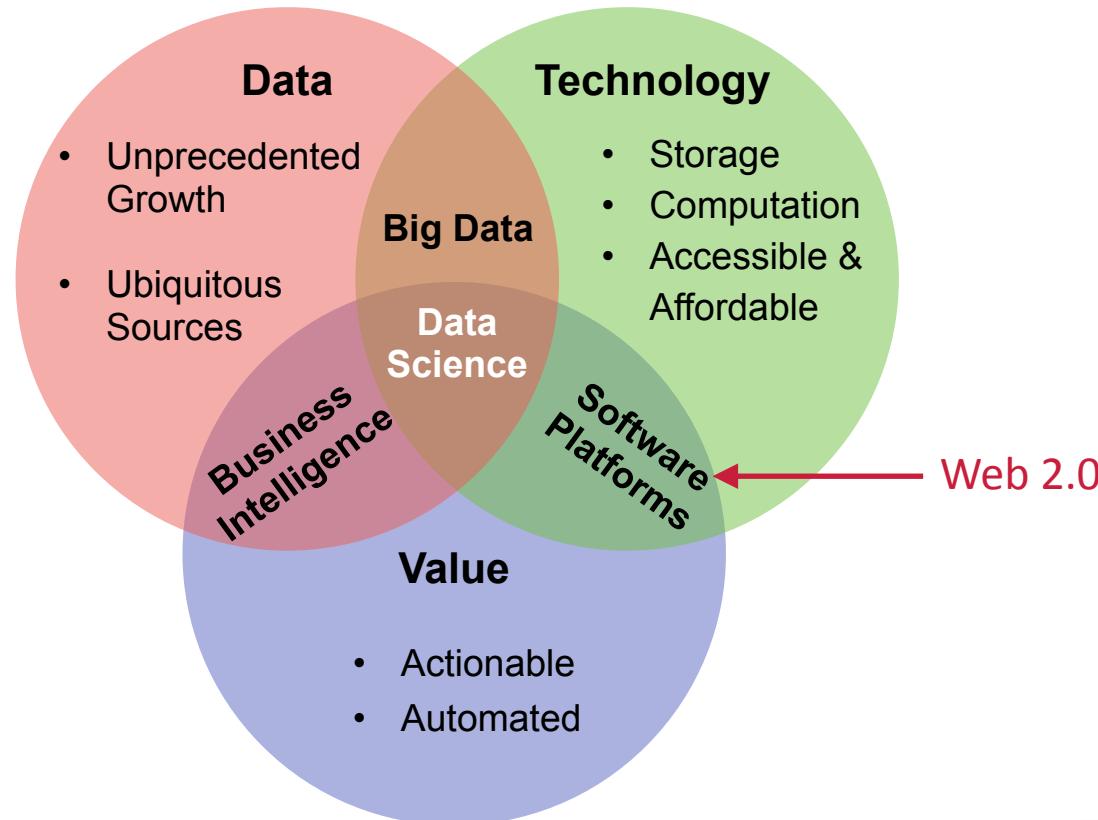
A Brief Historical Diversion

foundation -> application -> definition -> discussion -> profession



source: https://commons.wikimedia.org/wiki/File:The_Tamaroa_in_the_Storm_by_Terrence_Maley_DVIDS1082872.jpg

Perfect Storm



Data

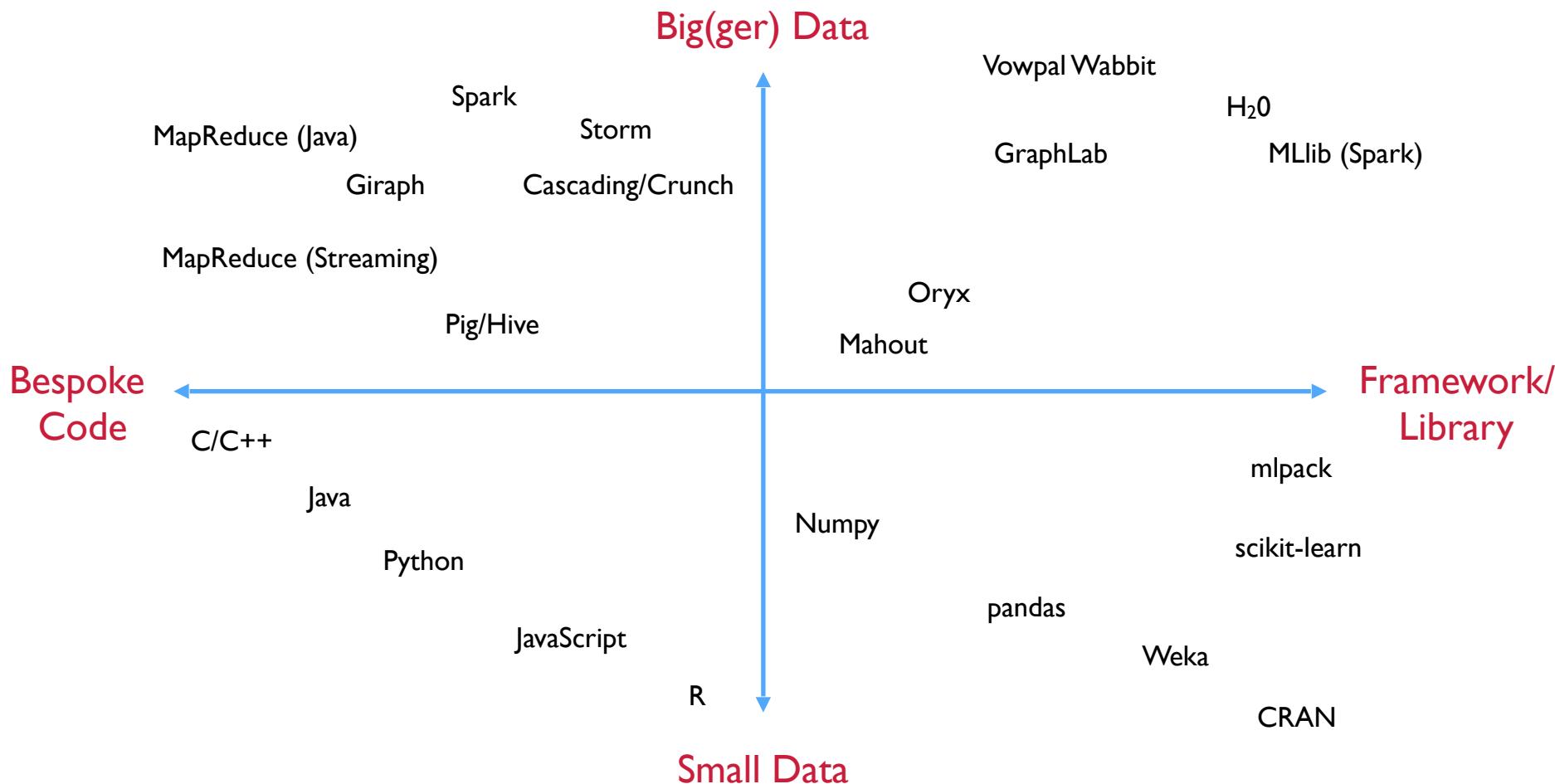
Sensor Web Mobile Personal

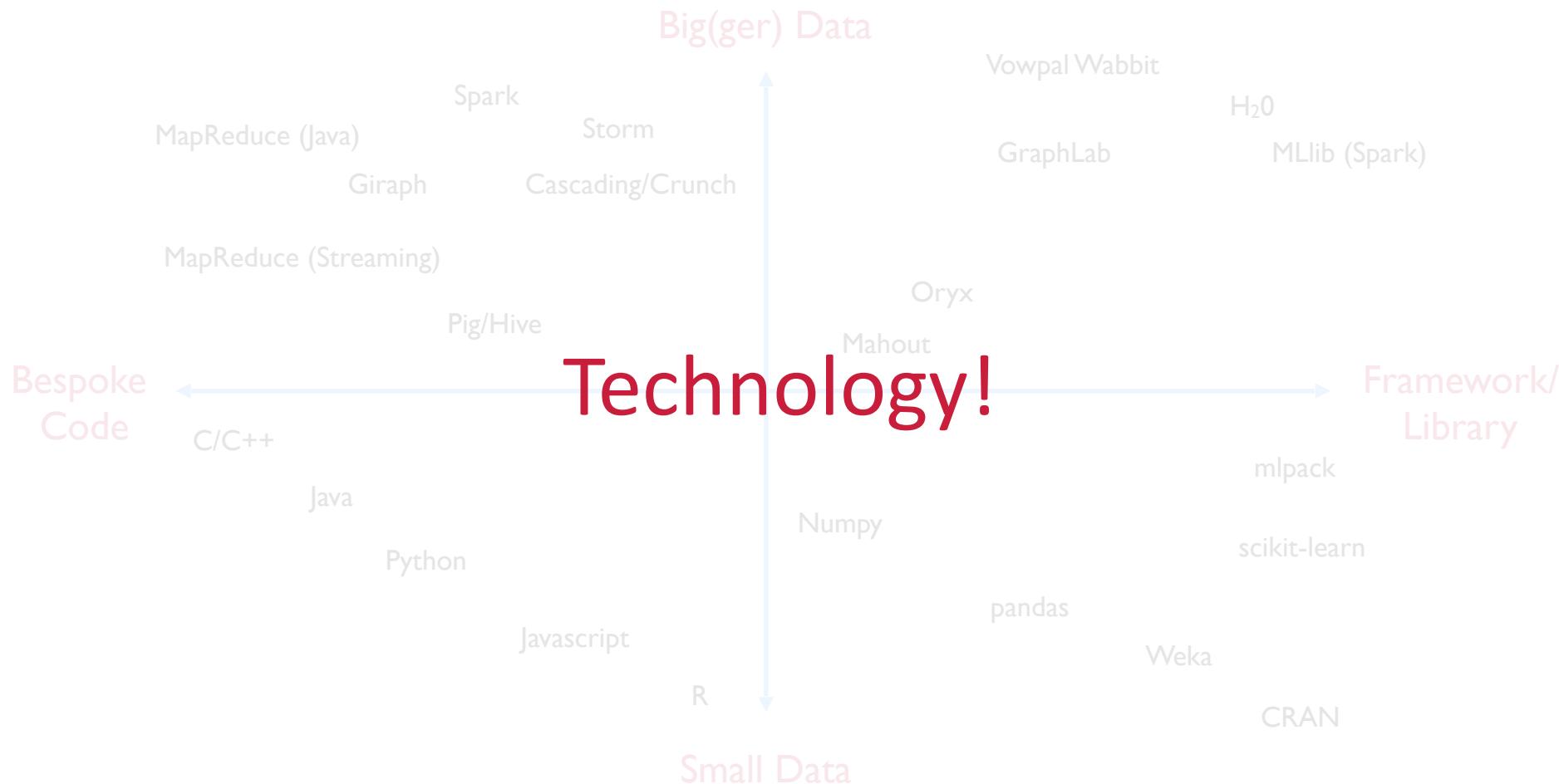
- | | | | |
|--------------|--------------------|------------|----------------|
| • RFID | • Application Logs | • Location | • Fitbit |
| • GPS | • Transactions | • Photos | • 23andMe |
| • Industrial | • Advertising | • Messages | • MyFitnessPal |
| • IoT | • Social Media | • Metadata | • Smart Meter |

Data

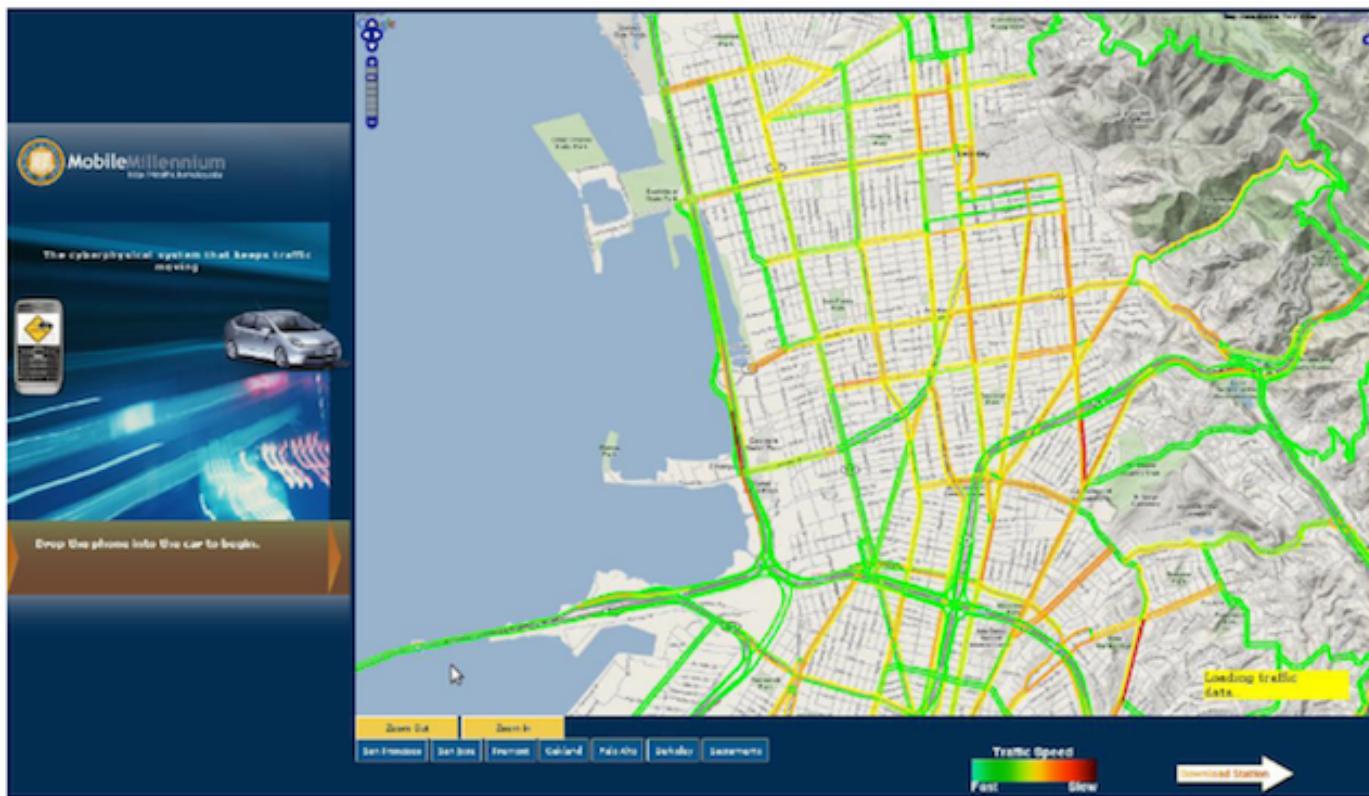
- Facebook collects 600TB of data every day (2014).
- YouTube processes 300 hours of video every minute (2014).
- 500 million tweets per day (2013).
- 8 billion check-ins on Foursquare.

Live Dashboard: <http://www.internetlivestats.com/>





Value: Mobile Millenium



source: <http://arstechnica.com/gadgets/2011/02/calling-all-cars-measuring-traffic-using-cell-phone-data/1/>

Value

Finance

- Fraud detection, HFT

Consumer

- Customer risk analysis, product recommendations

Advertising

- Targeting, CTR prediction

Telecommunications

- IT Security, intrusion detection, packet analysis

Healthcare

- Drug interactions, diagnostic assistance

3 Applications of Data Science

<i>Application</i>	<i>Benefit</i>	<i>Type</i>	<i>Example</i>
Analytics	<i>Metrics which provide business value</i>	<i>Inferential</i>	<i>CLV, A/B Testing, Churn</i>

Data Analytics



Build an Audience

Hi Jonathan,

You haven't visited Zipfian Academy in a while and there are some quick things you can do to build interest in your Page:

- ✓ Post a new update or photo.
- ✓ Make sure all the parts of your Page are completed, including profile picture, cover photo, and about section.
- ✓ See our [Page posting tips](#) to increase engagement on your Page.

[Create a Post](#)

Thanks,
The Facebook Team

3 Applications of Data Science

<i>Application</i>	<i>Benefit</i>	<i>Type</i>	<i>Example</i>
Analytics	<i>Metrics which provide business value</i>	<i>Inferential</i>	<i>CLV, A/B Testing, Churn</i>
Products	<i>Data (and ML) used to enhance existing application</i>	<i>Predictive</i>	<i>PYMK, Pandora, Google Photos</i>



Recent

2015

2014

2013

2012

2011

2010

2009

2008

2007

2006

Born

The diagram illustrates the biomechanics of neck posture. It shows five human figures from the side, each with a highlighted spine and head. The first figure is labeled '0 degrees' with '10-12lbs' of pressure on the discs. The second figure is labeled '15 degrees' with '27lbs'. The third figure is labeled '30 degrees' with '40lbs'. The fourth figure is labeled '45 degrees' with '49lbs'. The fifth figure is labeled '60 degrees' with '60lbs'. This visualizes how increasing neck flexion leads to exponentially greater pressure on the cervical discs.

Alex Fitzpatrick

Friends

Following

Message

Timeline

About

Friends 9 Mutual

Photos

More

- Listened to You & I by Local Natives on Monday on Spotify
- Became friends with Seamus Kirst and 1 other person
- 9 mutual friends including Tanner Curtis and Denver Nicks
- Deputy Tech Editor at TIME
Past: Mashable and City of Rochester, Communications Bureau

...

FRIENDS - 803 (9 Mutual)

Ashley Ross

Vaughn Wallace

Alexandra Sifferlin

Post Photo / Video

Write something...

Alex Fitzpatrick
20 hrs ·

All this talk of measles has...
Gone viral

Like · Comment · Share

Craig Kanalley, Rubina Madan Fillion and 29 others like this.

View 12 more comments

Robert Potter Alex Fitzpatrick when you have your first kid I want to be there when you look into its eyes and say 'aw, you are such a little dick'
16 hrs · Like ·

Alex Fitzpatrick Robert I'm not planning on getting pregnant until at least after the wedding
16 hrs · Like ·

Ryan Teague Beckwith I'm actually wondering if this "babies are dicks" thing isn't just a spectacular misunderstanding of the birds and the bees talk.
16 hrs · Like ·

Sponsored

Data Products

Consume Data

Create Data

GIFGIF

Search | Results | Data | About

Which better expresses **excitement**?

Change Questio



Achievements

Import | Export | Delete

Your votes: 44



Analysis: Excitement

Global votes: 2,530,975

Be



Wor.



3 Applications of Data Science

<i>Application</i>	<i>Benefit</i>	<i>Type</i>	<i>Example</i>
Analytics	<i>Metrics which provide business value</i>	<i>Inferential</i>	<i>CLV, A/B Testing, Churn</i>
Products	<i>Data (and ML) used to enhance existing application</i>	<i>Predictive</i>	<i>PYMK, Pandora, Google Photos</i>
Tools	<i>Interface to make data more accessible</i>	<i>Exploratory</i>	<i>Data USA</i>

PROFILES
LOCATIONS
INDUSTRIES
OCCUPATIONS
EDUCATION
STORIES
MAPS
DATA
ABOUT

DATAUSA

THE MOST COMPREHENSIVE
VISUALIZATION OF U.S. PUBLIC DATA

 Computer Science

All

Search

CREATED IN COLLABORATION

Deloitte.



Datawheel

STORY



FEATURED

MANUFACTURING CONNECTIONS

PROFILE

ECONOMY

STATE

WAGE BY RACE & ETHNICITY
IN WEST VIRGINIA

Wage Distribution

OPTIONS

WAGE GINI

0.312

Computer & information systems managers had a wage GINI coefficient of 0.312 in 2014, which is lower than the national average of 0.485. In other words, wages are more evenly distributed for Computer & information systems managers than the overall labor force.

Dataset: 2014 ACS PUMS 1-year Estimate

Source: Census Bureau



Wage by Location

OPTIONS

THE STATES WHERE COMPUTER & INFORMATION SYSTEMS MANAGERS ARE PAID THE HIGHEST ARE

1. New Jersey

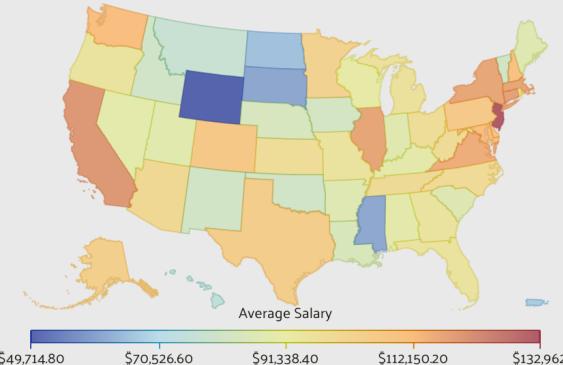
2. California

3. Connecticut

This map shows the states in the United States shaded by wage for Computer & information systems managers.

Dataset: 2014 ACS PUMS 5-year Estimate

Source: Census Bureau



**Surround us every day often
without realizing**

Recommendation

**Netflix, Pandora, Amazon,
Facebook Newsfeed**

Spam Filtering, Search Results, Web Advertising

Environment

Python

C`Python`

Anaconda

Enthought

J`ython`

Editor

Sublime Text

Notepad++

PyCharm/Spyder

Emacs/vim/nano

Terminal

Terminal.app

Command Prompt

iTerm

Cygwin

DOWNLOAD ANACONDA NOW!

Jump to: [Windows](#) | [OS X](#) | [Linux](#)

Get Superpowers with Anaconda

Anaconda is the leading open data science platform powered by Python. The open source version of Anaconda is a high performance distribution of Python and R and includes over 100 of the most popular [Python](#), R and Scala packages for data science. Additionally, you'll have access to over 720 packages that can easily be installed with conda, our renowned package, dependency and environment manager, that is included in Anaconda. Anaconda is BSD licensed which gives you permission to use Anaconda commercially and for redistribution. See [the packages included with Anaconda](#) and [the Anaconda changelog](#).

Which version should I download and install?

Because Anaconda includes installers for Python 2.7 and 3.5, either is fine. Using either version, you can use Python 3.4 with the conda command. You can create a 3.5 environment with the conda command if you've downloaded 2.7 — and vice versa.

If you don't have time or disk space for the entire distribution, try [Miniconda](#), which contains only conda and Python. Then install just the individual packages you want through the conda command.



The banner features the Anaconda logo (green stylized 'A') and the word 'ANACONDA' in green. Below it says 'now available for cloudera'. A blue button at the bottom right says 'GET NOW'.



The banner has a white header with the text 'VIEW OUR WEBINARS' in blue. Below it is a section with the text 'See upcoming and previous webinars from our experts.' and a blue 'VIEW NOW' button.

Anaconda for OS X

PYTHON 2.7	PYTHON 3.5
MAC OS X 64-BIT GRAPHICAL INSTALLER <small>339M (OS X 10.7 or higher)</small>	MAC OS X 64-BIT GRAPHICAL INSTALLER <small>342M (OS X 10.7 or higher)</small>
Mac OS X 64-bit Command-Line installer <small>290M (OS X 10.7 or higher)</small>	Mac OS X 64-bit Command-Line installer <small>293M (OS X 10.7 or higher)</small>

OS X Anaconda Installation

Choose either the graphical installer or the command line installer for OS X.

Graphical Installer:

1. Download the graphical installer.
2. Double-click the downloaded .pkg file and follow the instructions.

Command Line Installer:

1. Download the command line installer.
2. In your terminal window, type one of the below and follow the instructions:

Python 2.7:

```
bash Anaconda2-4.0.0-MacOSX-x86_64.sh
```

Python 3.5:

```
bash Anaconda3-4.0.0-MacOSX-x86_64.sh
```



Anaconda for OS X



OS X Anaconda Installation

Choose either the graphical installer or the command line installer for OS X.

Graphical Installer:

1. Download the graphical installer.
2. Double-click the downloaded .pkg file and follow the instructions.

Command Line Installer:

1. Download the command line installer.
2. In your terminal window, type one of the below and follow the instructions:

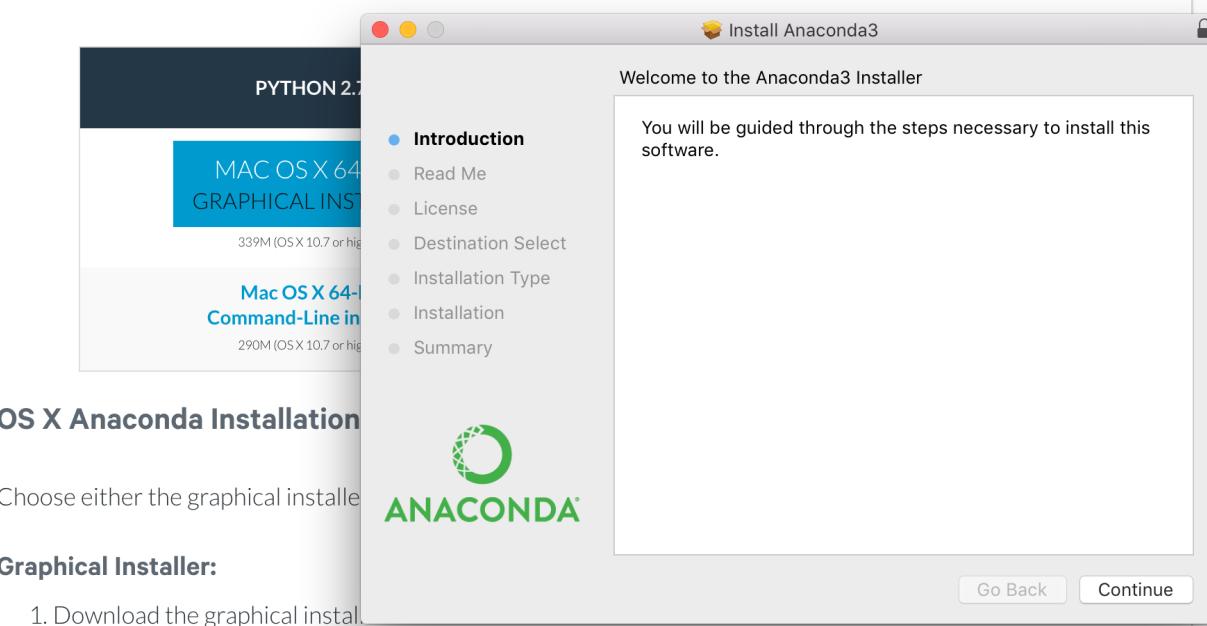
Python 2.7:

```
bash Anaconda2-4.0.0-MacOSX-x86_64.sh
```

Python 3.5:

```
bash Anaconda3-4.0.0-MacOSX-x86_64.sh
```

Anaconda for OS X



OS X Anaconda Installation

Choose either the graphical installer or the command-line installer.

Graphical Installer:

1. Download the graphical installer.
2. Double-click the downloaded .pkg file and follow the instructions.

Command Line Installer:

1. Download the command line installer.
2. In your terminal window, type one of the below and follow the instructions:

Python 2.7:

```
bash Anaconda2-4.0.0-MacOSX-x86_64.sh
```

Python 3.5:

```
bash Anaconda3-4.0.0-MacOSX-x86_64.sh
```

Python version: 3.5

Number of supported packages: 395

Name	Version	Summary / License	In Installer
abstract-rendering  Linux Mac	0.5.1	Rendering as a binning process / 3-clause BSD	
alabaster 	0.7.7	configurable sidebar-enabled Sphinx theme / BSD	✓
anaconda-build 	0.14.1	Anaconda build client library / proprietary - Continuum Analytics, Inc.	
anaconda-client 	1.4.0	anaconda.org command line client library / BSD	✓
anaconda-navigator 	1.1.0	Anaconda Navigator / proprietary - Continuum Analytics, Inc.	✓
ansi2html 	1.1.0	Convert text with ANSI color codes to HTML. / GPLv3+	
appnope  Mac	0.1.0	Disable App Nap on OS X 10.9 / BSD	✓
appscript  Mac	1.0.1	Control AppleScriptable applications from Python / Public-Domain	✓
argcomplete 	1.0.0	Bash tab completion for argparse. Tab complete all the things! / Apache	✓
astroid 	1.4.4	abstract syntax tree for Python with inference support. / LGPL	
astropy 	1.1.2	Community-developed python astronomy tools / BSD	✓
attrs 	15.2.0	attributes without boilerplate / MIT	
azure 	1.0.2	easy to access the Microsoft Azure components / Apache License 2.0	
babel 	2.2.0	Internationalization utilities / BSD	✓
basemap  Linux Mac	1.0.7	Plot data on map projections with matplotlib / PSF	
bcolz  Linux Mac	0.12.1	columnar and compressed data containers. / BSD	
bcrypt 	2.0.0	modern password hashing for your software and your servers / Apache License, Version 2.0	

Included Packages

Get started

Using conda

Building packages

Help & reference

Get involved

Installing Packages

Conda

Conda is an open source package management system and environment management system for installing multiple versions of software packages and their dependencies and switching easily between them. It works on Linux, OS X and Windows, and was created for Python programs but can package and distribute any software.

Conda is included in Anaconda and Miniconda. Conda is also included in the Continuum [subscriptions](#) of Anaconda, which provide on-site enterprise package and environment management for Python, R, Node.js, Java, and other application stacks. Conda is also available on pypi, although that approach may not be as up-to-date.

- Miniconda is a small “bootstrap” version that includes only conda and conda-build, and installs Python. Over 720 scientific packages and their dependencies can be installed individually from the Continuum repository with the “conda install” command.
- Anaconda includes conda, conda-build, Python, and over 150 automatically installed scientific packages and their dependencies. As with Miniconda, over 250 additional scientific packages can be installed individually with the “conda install” command.
- pip install conda uses the released version on pypi. This version allows you to create new conda environments using any python installation, and a new version of Python will then be installed into those environments. These environments are still considered “Anaconda installations.”

The `conda` command is the primary interface for managing [Anaconda](#) installations. It can query and search the Anaconda package index and current Anaconda installation, create new conda environments, and install and update packages into existing conda environments.

Python 2 versus Python 3

Python 2

- Terminal... EOL date set for **2020**
- Legacy code and libraries

Python 3

- Much better Unicode Support
- `asyncio`
- Backwards **incompatible**

Python 2 versus Python 3 (syntax)

<i>Python 2</i>	<i>Python 3</i>
<code>print "Hello"</code>	<code>print("Hello")</code>
<code>1/4 == 0</code>	<code>1/4 == .25 ; 1//4 == 0</code>
<code>xrange()</code>	<code>range()</code>
<code>file("path_to_data.csv")</code>	<code>open("path_to_data.csv")</code>
<code>raw_input()</code>	<code>input()</code>

Python 2 versus Python 3

Python 2

- Terminal... EOL
- Legacy code and libraries

When in doubt...

Python 3

- Much better Unicode Support
- `asyncio`
- Backwards **incompatible**

Python 2 versus Python 3

Python 2

- Terminal... EOL
- Legacy code and libraries

Python 3

- Much better Unicode Support
- `asyncio`
- Backwards **incompatible**

When in doubt... Python 3

Python Quick Start

- [Official Tutorial](#)
- [Learn X in Y minutes \(Python cheatsheet\)](#)
- [Python Style Guide \(PEP 8\)](#)
- [Crash Into Python or Python for Programmers with 3 Hours](#) (Python 2)
- Scoping
 - [pass-by-reference or pass-by-value](#)
 - [LEGB Rule](#)
- [Common Gotchas](#)

Data structures: Lists

```
my_var = 4

# Initialize a new List. Can contain heterogeneous types
my_list = [1, "jonathan", [1,2,3], my_var]

# Indexing
my_list[1] #=> 'jonathan'

# Slicing (end point exclusive)
my_list[:2] #=> [1, 'jonathan']

# Negative Indexing (everything but last element)
my_list[:-1] #=> [1, 'jonathan', [1,2,3]]
```

Data structures: Lists

```
# In-place Append
my_list.append(10)
print(my_list) #=> [1, 'jonathan', [1, 2, 3], 4, 10]

# Item Assignment
my_list[2] = "dinu"
print(my_list) #=> [1, 'jonathan', 'dinu', 4, 10]

# Concatenate two lists
my_list + [11, 12, 13] #=> [1, 'jonathan', 'dinu', 4, 10, 11, 12, 13]

# Concatenate and update
my_list += [11, 12, 13]
print(my_list) #=> [1, 'jonathan', 'dinu', 4, 10, 11, 12, 13]
```

Data structures: Lists

```
num_list = [3, 2, 7, 1, 5]

# Sorting
sorted(num_list) #=> [1, 2, 3, 5, 7]

# In-place Sort
num_list.sort() #=> None
print(num_list) #=> [1, 2, 3, 5, 7]
```

Data structures: Tuples

```
tup = ("hello", 2, 3)

# Indexing like lists
tup[0] => 'hello'

# Tuples are Immutable however
tup[1] = 5 => TypeError: 'tuple' object does not support item assignment
```

Data structures: Dictionaries

```
employees = {'Jon': 42, 'Anne': 53213}

# Set new key
employees['Elisa'] = 7821

# Check Membership
'Jon' in employees #=> True

# Remove a key
del employees['Jon']
'Jon' in employees #=> False

employees.items() #=> dict_items([('Anne', 53213), ('Elisa', 7821)])

tup = ('Jonathan', 'Dinu')
employees[tup] = 21235

print(dictionary) #=> {'Anne': 53213, 'Elisa': 7821, ('Jonathan', 'Dinu'): 21235}
```

Data structures: Sets

```
words = set(['the', 'red', 'panda', 'jumped', 'over', 'the', 'red', 'barn'])

# No intrinsic order
print(words) #=> {'panda', 'over', 'red', 'the', 'barn', 'jumped'}
```



```
# Great for membership checks
'fox' in words #=> False
```



```
sentence = set(['the', 'red', 'fox', 'jumped', 'over', 'the', 'brown', 'fence'])
```

Data structures: Set Logic

```
# Difference
sentence - words #=> {'brown', 'fence', 'fox'}
words - sentence #=> {'barn', 'panda'}
```



```
# Union
words | sentence #=> {'barn', 'brown', 'fence', 'fox', 'jumped', 'over',
'panda', 'red', 'the'}
```



```
# Intersection
words & sentence #=> {'jumped', 'over', 'red', 'the'}
```

Control Flow: Conditionals

```
if my_var == 42:  
    print('The Universe makes sense')  
elif 42 == 42:  
    print("Numbers work")  
else:  
    print('Numbers are broken ^\_\(^/_^')  
#=> 'Numbers work'  
  
# Can use Ternary operator too!  
'Numbers work' if 42 == 42 else 'Numbers are broken ^\_\(^/_^'  
#=> 'Numbers work'
```

Control Flow: Loops/Iteration

```
for item in my_list[4:]:
    print(item)
=> 10
=> 11
=> 12
=> 13
```

```
for idx in range(3):
    print(my_list[idx])
=> 1
=> 'jonathan'
=> 'dinu'
```

- **Official Tutorial:** <https://docs.python.org/3/tutorial/>
- **Cheat Sheet:** <https://learnxinyminutes.com/docs/python3/>
- **Python Time Complexity:** <https://wiki.python.org/moin/TimeComplexity>

Python in a Slide

```
def wordbuzz(input_arr):
    for word in input_arr:
        counts = {}
        for letter in word:
            if letter in counts:
                counts[letter] += 1
            else:
                counts[letter] = 1
        if max(counts.values()) > 1:
            print(word)
```

Function Definition → `def wordbuzz(input_arr):`

Loops → `for word in input_arr:`
Loops → `for letter in word:`

Conditional → `if letter in counts:`
Conditional → `else:`

Built-in Function → `max(counts.values())`

Function Parameters (or arguments) → `input_arr`

Dictionary → `counts = {}`

In place add → `counts[letter] += 1`

Print statement (notice the parentheses) → `print(word)`



LEARN PYTHON

THE HARD WAY

3RD
EDITION

Welcome to the 3rd Edition of Learn Python the Hard Way. You can visit the companion site to the book at <http://learnpythonthehardway.org/> where you can purchase digital downloads and paper versions of the book. The free HTML version of the book is available at <http://learnpythonthehardway.org/book/>.

Table Of Contents

- Preface
- Introduction: The Hard Way Is Easier
- Exercise 0: The Setup
- Exercise 1: A Good First Program
- Exercise 2: Comments And Pound Characters
- Exercise 3: Numbers And Math
- Exercise 4: Variables And Names
- Exercise 5: More Variables And Printing
- Exercise 6: Strings And Text
- Exercise 7: More Printing
- Exercise 8: Printing, Printing
- Exercise 9: Printing, Printing, Printing
- Exercise 10: What Was That?
- Exercise 11: Asking Questions
- Exercise 12: Prompting People
- Exercise 13: Parameters, Unpacking, Variables
- Exercise 14: Prompting And Passing
- Exercise 15: Reading Files

1
2
3
4
MAIN



PLAY VIDEO



PREVIOUS



NEXT



HELP



Follow

Learn to Code

- [Learn Python the Hard Way](#) (online book)
- [Think Python](#) (online book)
- [Dive into Python](#) (online book)
- [Google's Python Class](#) (videos + exercises)
- [Software Carpentry](#) (videos + exercises + in-person)
- [Introduction to Computer Science and Programming Using Python](#) (MOOC)
- [Structure and Interpretation of Computer Programs](#) (MOOC)
- [Codecademy Python](#) (interactive)
- [Learning Python](#) (paid book)
- [HGTB: Learning Python](#) (compilation)
- [Full Stack Python Resources](#) (compilation)

- [Think Python](#) (online book)
 - [Dive into Python](#) (online book)
 - [Learn Python the Hard Way](#) (online book)
 - [Google's Python Class](#) (videos + exercises)
 - [Software Carpentry](#) (videos + exercises + in-person)
 - [Introduction to Programming in Python](#) (MOOC)
 - [Structure and Interpretation of Computer Programs](#) (MOOC)
 - [Codecademy Python](#) (interactive)
 - [Learning Python](#) (paid book)
 - [HGTB: Learning Python](#) (compilation)
 - [Full Stack Python Resources](#) (compilation)
- No Shortage of Resources!!**

<https://pinboard.in/u:hopelessoptimism/t:learnpython/>

[Python](#)[PSF](#)[Docs](#)[PyPI](#)[Jobs](#)[Community](#)

Search

GO

Socialize

Sign In

[About](#)[Downloads](#)[Documentation](#)[Community](#)[Success Stories](#)[News](#)[Events](#)

```
Python 3.5.1 (default, Dec 18 2015, 00:00:00)
[GCC 4.8.4] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

Online console from [PythonAnywhere](#)

Python is a programming language that lets you work quickly
and integrate systems more effectively. [»» Learn More](#)

Get Started

Whether you're new to
programming or an experienced

Download

Python source code and installers
are available for download for all

Docs

Documentation for Python's
standard library, along with tutorials

Jobs

Looking for work or have a Python
related position that you're trying to

[Python](#)[PSF](#)[Docs](#)[PyPI](#)[Jobs](#)[Community](#)

Search

GO

Socialize

Sign In

[About](#)[Downloads](#)[Documentation](#)[Community](#)[Success Stories](#)[News](#)[Events](#)

```
Python 3.5.1 (default, Dec 18 2015, 00:00:00)
[GCC 4.8.4] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

Online console from [PythonAnywhere](#)

Python is a programming language that lets you work quickly
and integrate systems more effectively. [»» Learn More](#)

Get Started

Whether you're new to
programming or an experienced

Download

Python source code and installers
are available for download for all

Docs

Documentation for Python's
standard library, along with tutorials

Jobs

Looking for work or have a Python
related position that you're trying to

Coming from Other Languages

- General Purpose Language
- Multi-paradigm (functional and object oriented)
- Interpreted
- Dynamic Typing
- Automatic Memory Management
- Pass by Value
- Open Source
- Comprehensive standard library
- C extensions

Why Python (for Data Science)?

- Easy to read (executable pseudocode!)
- Easy to write (and learn)
- Open Source
- Community
 - Resources and Documentation
 - Events and Mentorship
 - Interoperable ecosystem of libraries

Why Python (for Data Science)?

- Easy to read (executable pseudocode!)
- Easy to write (and learn)
- Open Source
- Community
 - Resources and Documentation
 - Events and Mentorship
 - Interoperable ecosystem of libraries

Ecosystem

pandas

matplotlib

statsmodels

scikit-learn

Flask &
Django

Python Interpreter

CPython
(C implementation)

Databases
(SQL, NoSQL, etc.)

Blaze & pyspark
(parallel/distributed)

Theano (GPU)

Ecosystem

pandas

matplotlib

statsmodels

scikit-learn

Flask &
Django

It's the Glue!

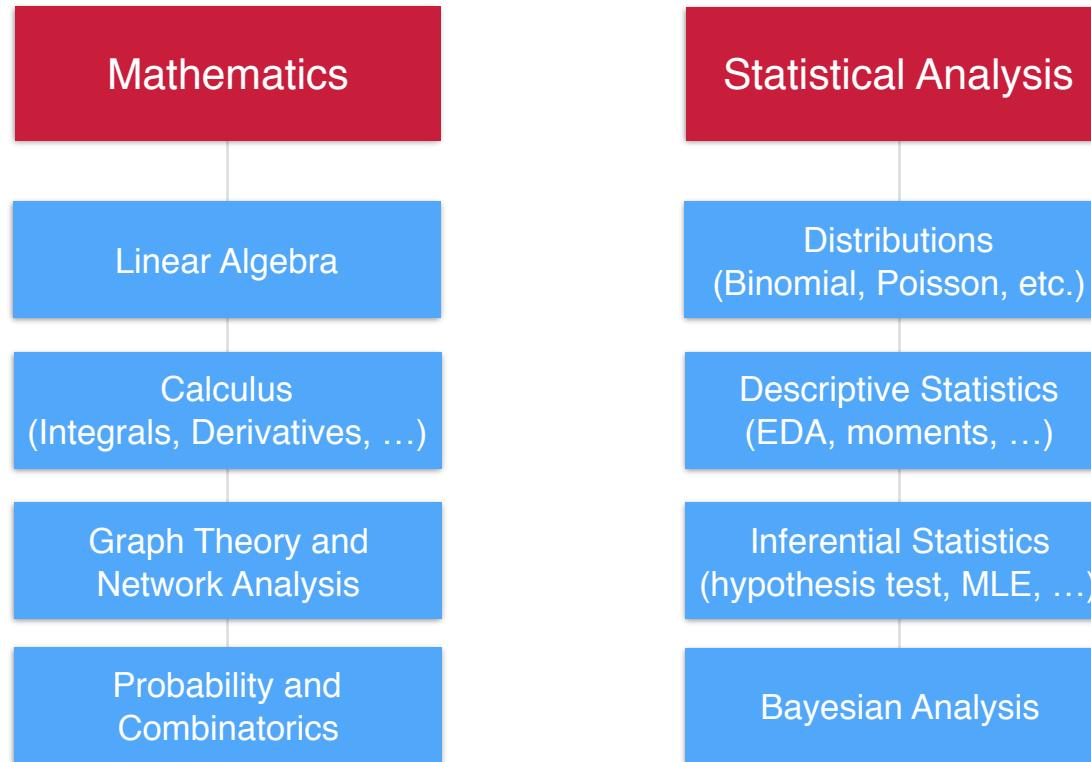
CPython
(C extensions)

Databases
(SQL, NoSQL, etc.)

Blaze & pyspark
(parallel/distributed)

Theano (GPU)

Theory



Techniques

Machine Learning

Supervised
(KNN, Logistic Regression, ...)

Unsupervised
(K-means, PCA/SVD, ...)

Validation, Model Comparison

NLP/ Information Retrieval

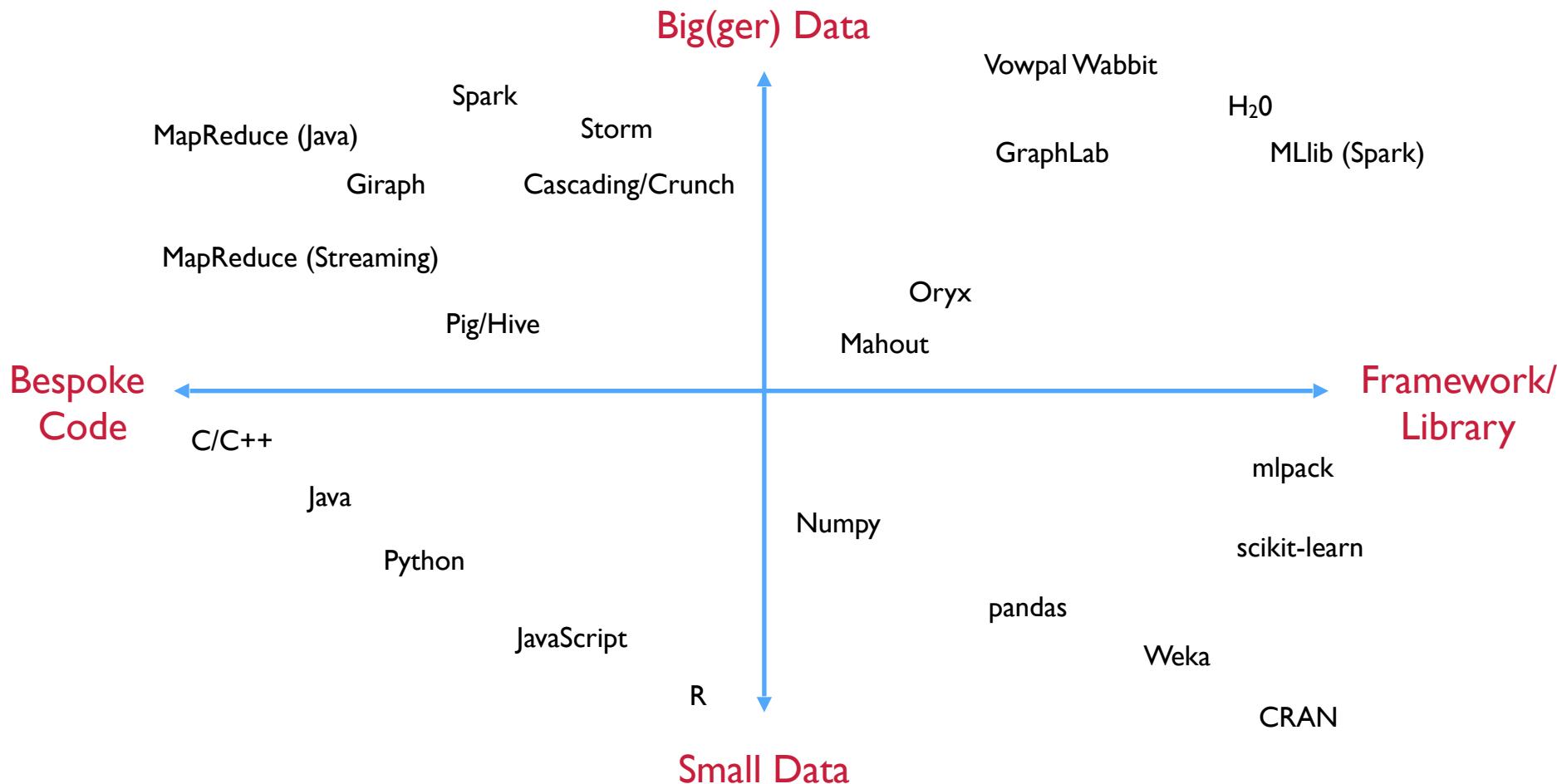
Software Engineering

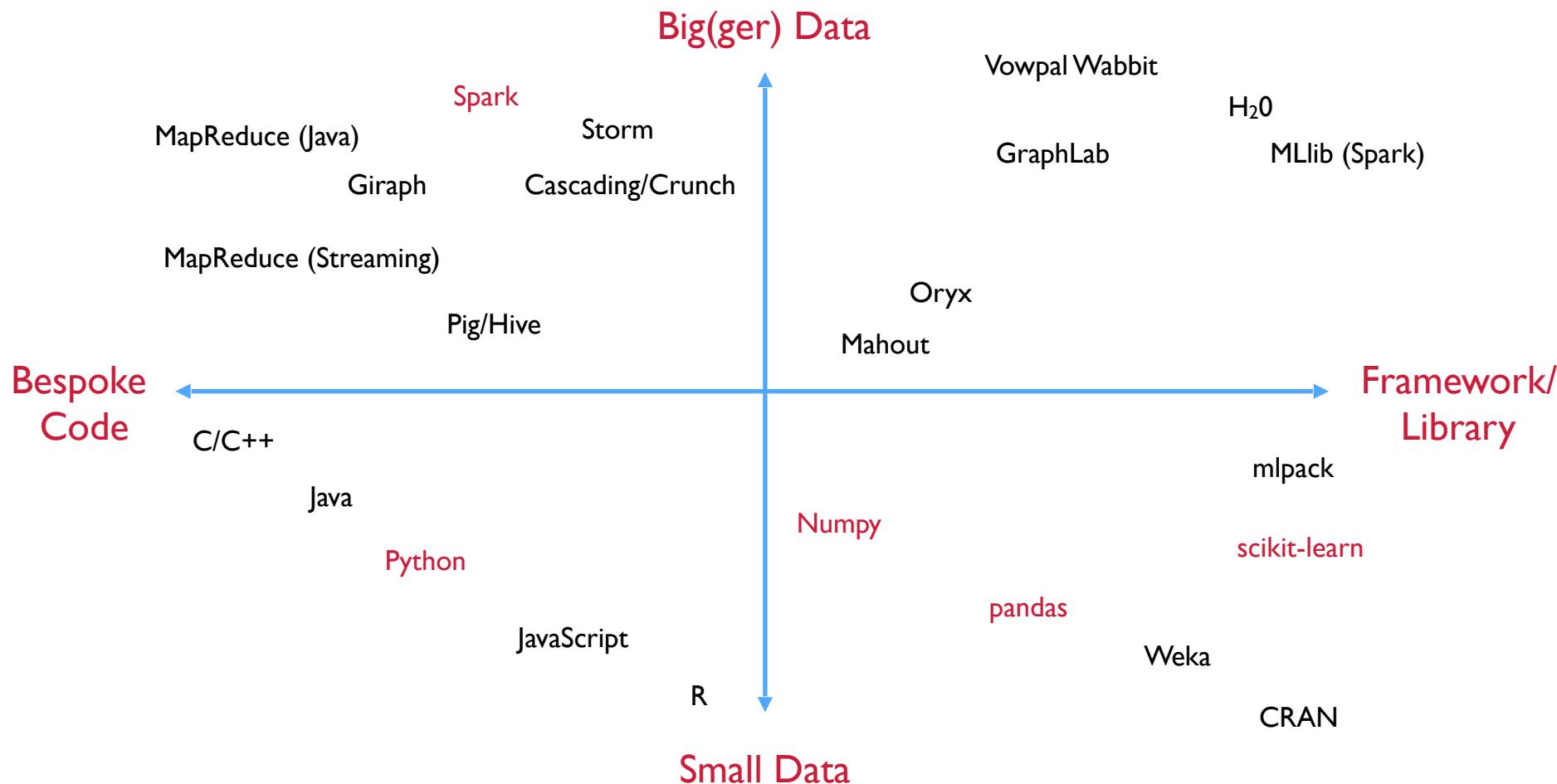
Algorithms and Data Structures

Databases and SQL

Distributed Computing

Data Visualization and Data Products





Review

- Data science has a long (and rich) history
- The applications and potential are seemingly limitless
- Python is well suited for data science due to its general purpose nature, rich ecosystem of libraries, and its strong community
- Python is easy (and fun) to learn and program