APS Workshop: **Introduction to Python**
San Francisco, CA, 24 May 2018

Overview

Christian C. Luhmann
Stony Brook University

Where to Find these Slides

github.com/cluhmann/python-psych-workshop

Who am I?

- B.S. in Computer Science
- Ph.D. in Psychology
- Stony Brook University
- Decision-making, learning, methods
- Computational modeling
- Using Python since ~2002

Who are You?

- Faculty/students?
- Who has used…
  - Matlab?
  - R?
  - Some other programming language (e.g., Java, C)?
  - SPSS?
  - Eprime?
  - SAS?

Goals

- Appreciation of the **ends**
  - benefits of Python
  - functionality provided by Python and its ecosystem
  - how to integrate these tools into your existing workflow
- Non-goals of this workshop: **means**
  - Ability to program Python without further consultation
  - Encyclopedic knowledge of packages, APIs, etc.
- Think of this as a open house
  - If you’d like buy, you still need to move all your stuff

What I will assume of you…

- Not much
- Not terrified of programming
- Use data in your research
- Looking for tools to conduct efficient, flexible, reproducible (maybe sharable) analyses
- Conduct laboratory experiments (maybe)

Why?

- Why Python?
- Matlab vs. R vs. whatever
  - why bother to learn another thing?
- Python…
  - is general-purpose
  - is free and open source
  - is eminently readable (i.e., readily learned)
  - has an extensive, well-integrated ecosystem of tools
  - and more!
- This workshop is, hopefully, a comprehensive answer

What is Python?

- Developed by Guido van Rossum in the early 1990s

- Python 2.0 was released October 16th, 2000

- Python 3.0 was released December 3rd, 2008

Python

- Free and open source
- Cross-platform
- Widely-used and well-supported
- Well-documented
- Multiple options for boosting performance
- **Highly readable**
- Substantial **standard library**
- Vibrant **third-party ecosystem**

Standard Library

```
>>> abs(-42)
42
```

```
>>> pow(2, 10)
1024
```

Standard Library

```
>>> min([1, 4, 12, 42])
1
```

```
>>> max([1, 4, 12, 42])
42
```

```
>>> len([1, 4, 12, 42])
4
```

```
>>> sum([1, 3, 5])
9
```

```
>>> sorted([2, 4, 6, 8, 1, 3, 5, 7, 9])
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

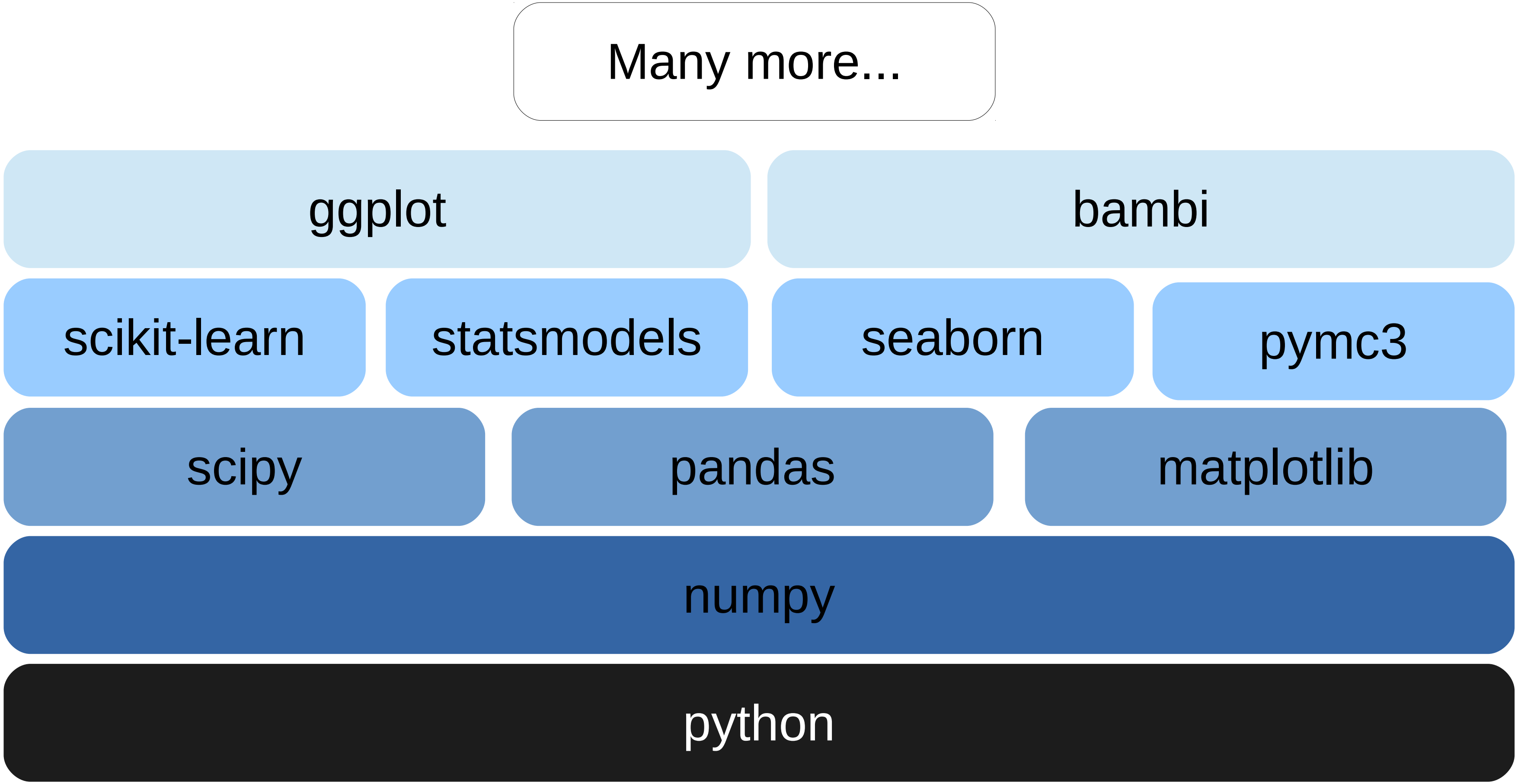
Standard Library

```
>>> print(‘Six times nine is ‘ + str(6*9))
Six times nine is 54
```

```
>>> file = open(‘myfile.txt’, ‘r’)
>>> contents = file.read()
>>> print(contents)
First line of my file.
Second line of my file.
Last line of my file.
```

```
>>> range(5)
[0, 1, 2, 3, 4]
```

Python’s Ecosystem


Many more...

ggplot

bambi

scikit-learn

statsmodels

seaborn

pymc3

scipy

pandas

matplotlib

numpy

python

Installing Python

- Anaconda
- Enthought’s Canopy
- WinPython (Windows only)
- Each of these projects provides:
  - Python
  - Packages
  - Package manager
  - Editor (IDE)
  - Other tools

Anaconda

www.anaconda.com/download

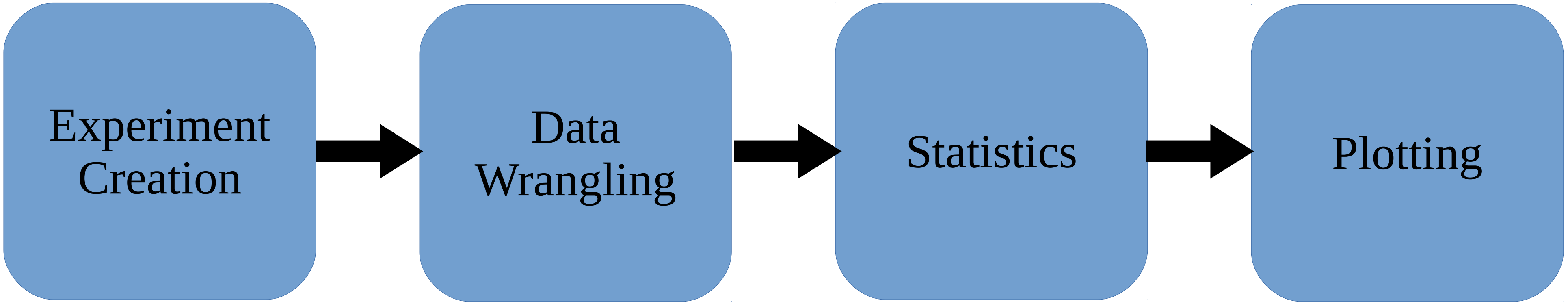
Installing Python

- Python 2.x or 3.x?

- Python 2.7's end-of-life initially 2015, but postponed to 2020
  - concern that much existing code could not easily be ported to Python 3

- Python 3.x is recommended

The Pipeline



Experiment Creation

Data Wrangling

Statistics

Plotting

Outline

1. Overview
2. Ways of using Python
3. Python basics
4. Data set overview
5. Data wrangling
6. Statistics
7. Plotting
8. Experiment creation

APS Workshop: **Introduction to Python**
San Francisco, CA, 24 May 2018

Using Python

Christian C. Luhmann
Stony Brook University

Python

- How do I **do** Python?

- Three ways we will cover today...

Console

Useful when...

- Playing with very brief pieces of code (i.e., one line at a time)
- No need to save code (or results) for later
- Trying things out
- Learning Python (exploring, trial & error)

Let’s try out the console

Scripts

Useful when…

- Executing large amounts of code
- Confident that your code works the way you want
- No need to “oversee” the code’s operation

Let’s see a script

Notebooks

Useful when…

- You want to play with code
- You want a record of what you do and results (mnemonic)
- You might want to share that record with others
- Teaching

Let’s try out a notebook

Notebooks

- Notebooks may be exported in a variety of formats, including…
  - PDF
  - LaTex
  - HTML
  - Python script
- Useful for supplying others (possibly non-programmers) with information about what you have done/found
  - Students/advisors
  - Collaborators
  - Readers
    - theatlantic.com/science/archive/2018/04/the-scientific-paper-is-obsolete/556676/

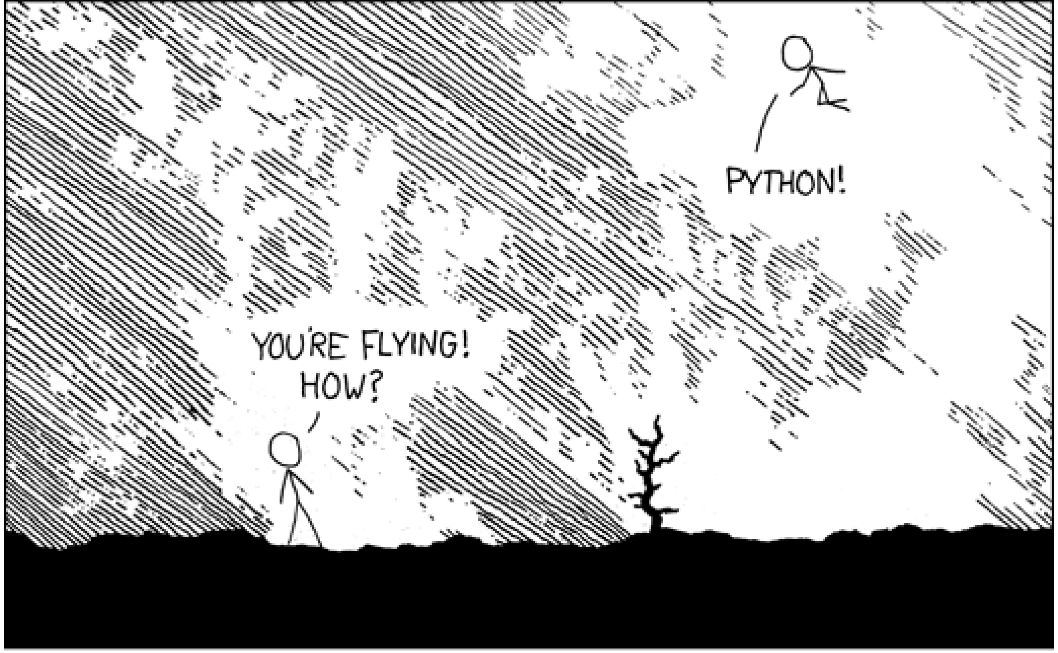
Outline

1. Overview
2. Ways of using Python
3. **Python basics**
4. Data set overview
5. Data wrangling
6. Statistics
7. Plotting
8. Experiment creation

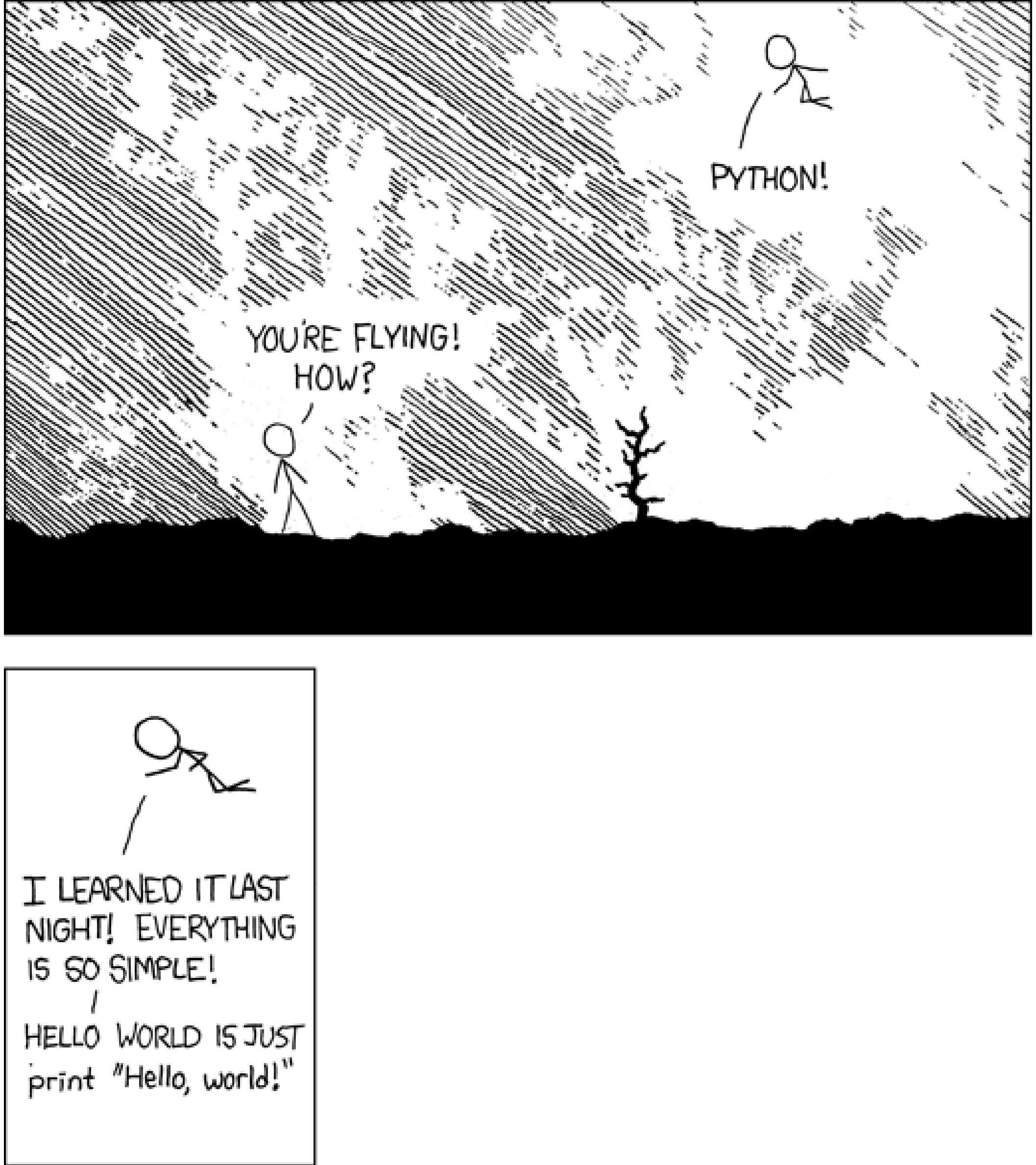
APS Workshop: **Introduction to Python**
San Francisco, CA, 24 May 2018

Python Basics

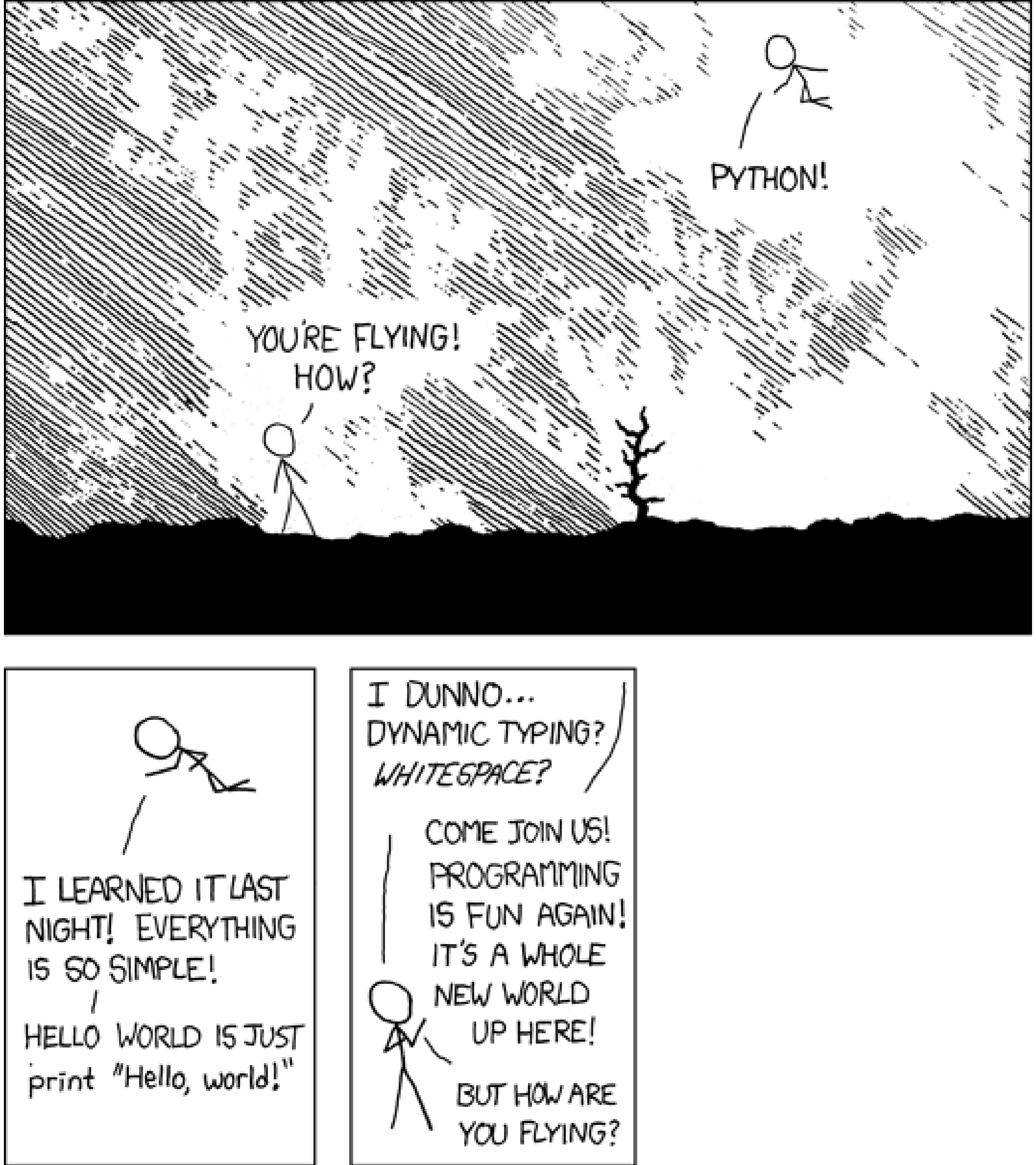
Christian C. Luhmann
Stony Brook University



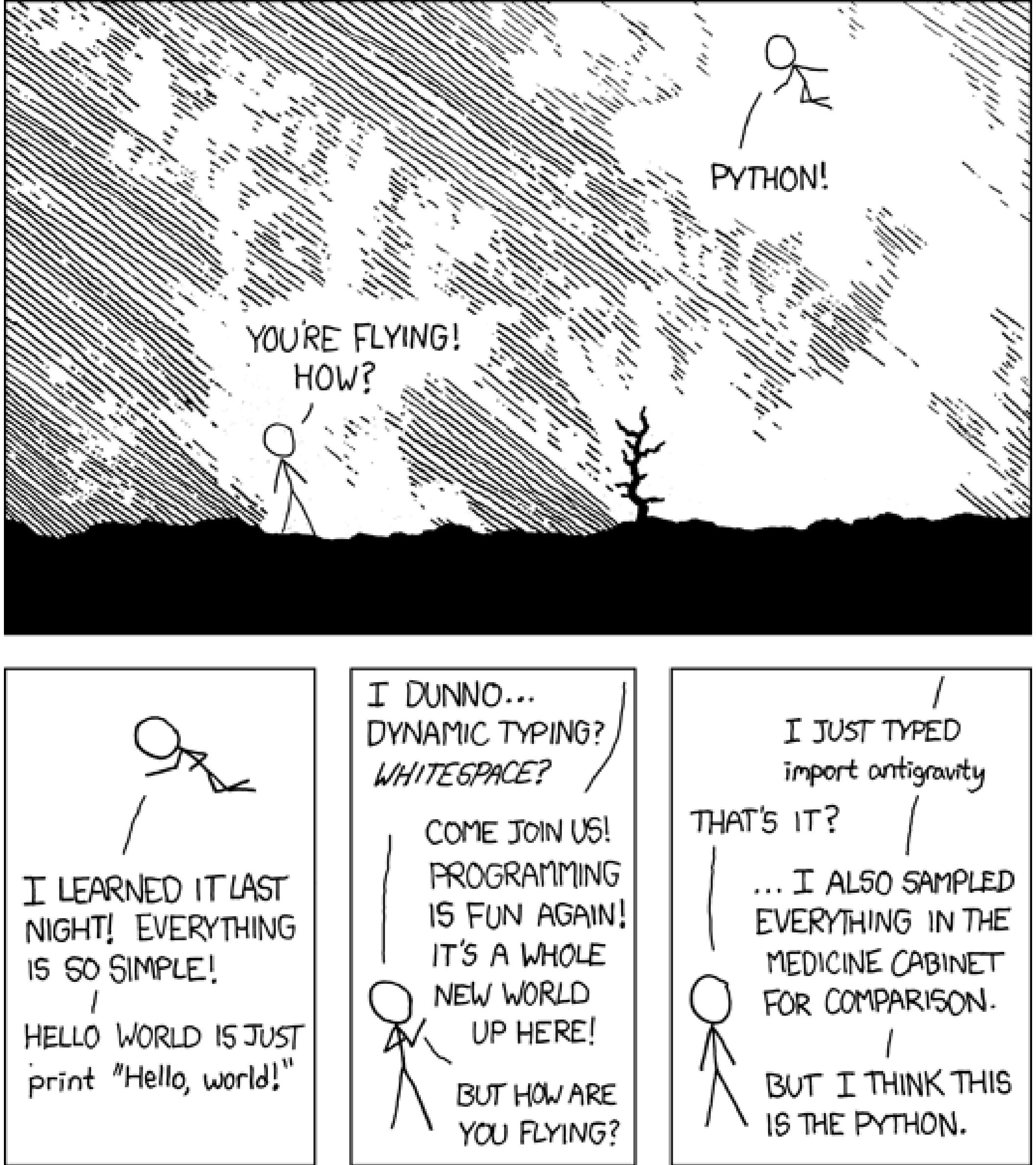
xkcd.com/353/


PYTHON!
YOU'RE FLYING! HOW?
I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST print "Hello, world!"

xkcd.com/353/


PYTHON!
YOU'RE FLYING! HOW?
I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST print "Hello, world!"
I DUNNO... DYNAMIC TYPING? WHITESPACE?
COME JOIN US! PROGRAMMING IS FUN AGAIN! IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?

xkcd.com/353/


YOU'RE FLYING!
HOW?
PYTHON!
I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST
print "Hello, world!"
I DUNNO... DYNAMIC TYPING? WHITESPACE?
COME JOIN US! PROGRAMMING IS FUN AGAIN! IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?
I JUST TYPED
import antigravity
THAT'S IT?
... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.
BUT I THINK THIS IS THE PYTHON.

xkcd.com/353/

```
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

Python

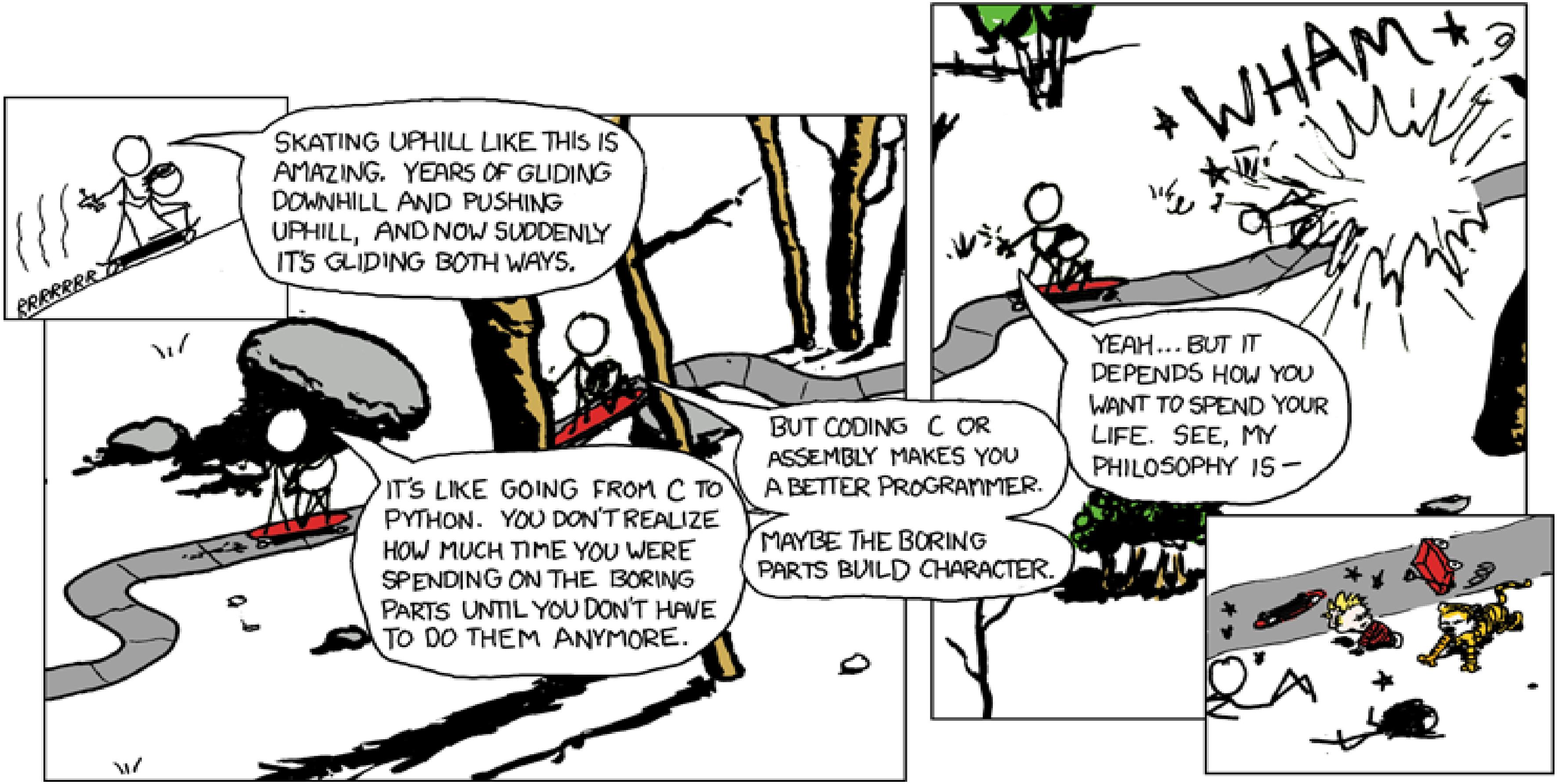
So let’s take a closer look at the language itself

Readability counts

- Every programming language represents a tradeoff between...
  - time it takes **to write** a program (your time)
  - time it takes **to run** a program (computer’s time)

- Python prioritizes **your** time

Python


RRRRRRR
SKATING UPHILL LIKE THIS IS AMAZING. YEARS OF GLIDING DOWNHILL AND PUSHING UPHILL, AND NOW SUDDENLY IT'S GLIDING BOTH WAYS.
IT'S LIKE GOING FROM C TO PYTHON. YOU DON'T REALIZE HOW MUCH TIME YOU WERE SPENDING ON THE BORING PARTS UNTIL YOU DON'T HAVE TO DO THEM ANYMORE.
BUT CODING C OR ASSEMBLY MAKES YOU A BETTER PROGRAMMER.
MAYBE THE BORING PARTS BUILD CHARACTER.
YEAH... BUT IT DEPENDS HOW YOU WANT TO SPEND YOUR LIFE. SEE, MY PHILOSOPHY IS —
WHAM

xkcd.com/409/

Python

For researchers, this means...

- students are less intimidated
- collaboration is easier
- improved transparency

Outline

1. Overview
2. Ways of using Python
3. Python basics
4. Data set overview
5. Data wrangling
6. Statistics
7. Plotting
8. Experiment creation

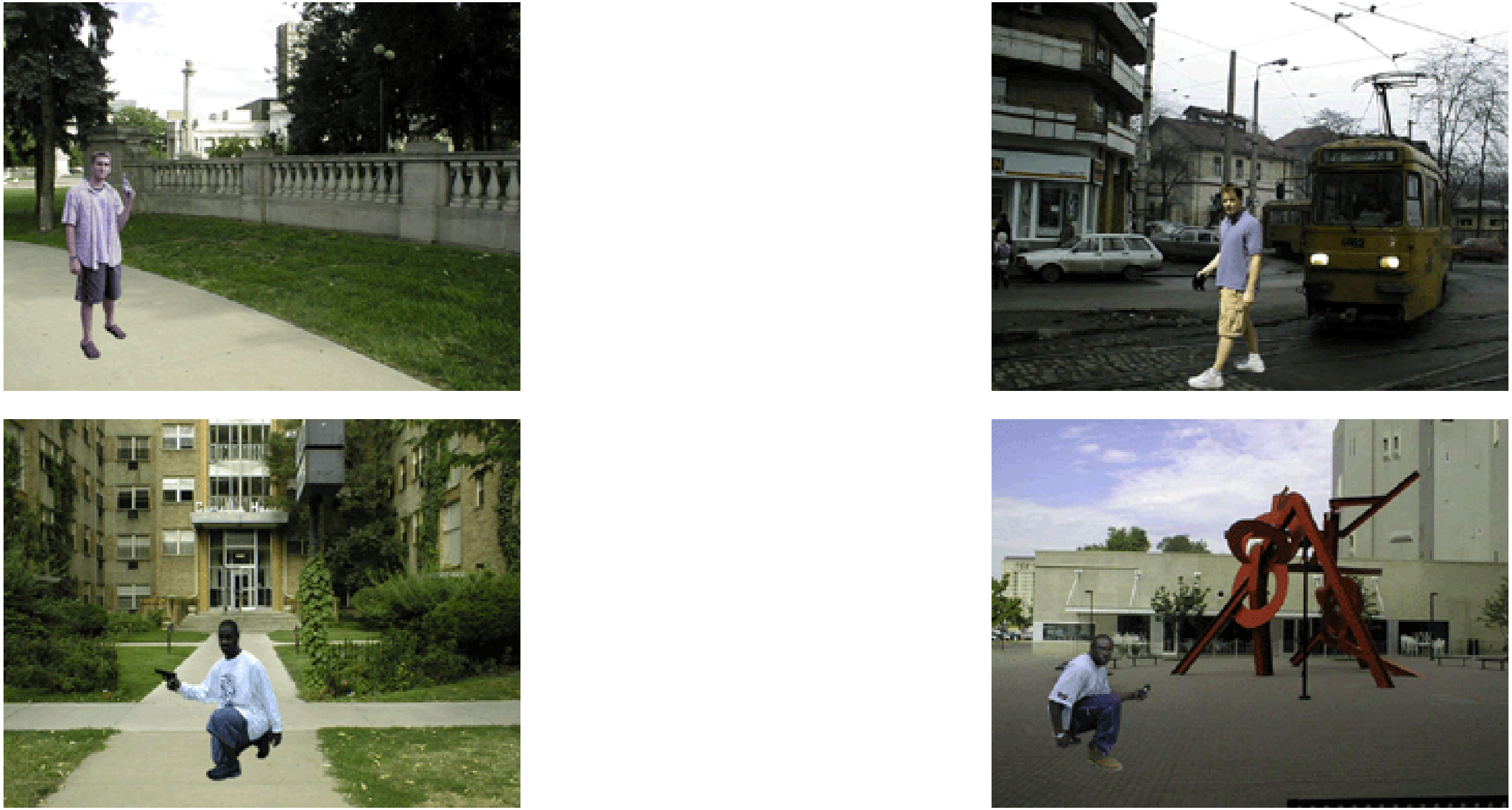
APS Workshop: **Introduction to Python**
San Francisco, CA, 24 May 2018

Our Dataset

Christian C. Luhmann
Stony Brook University

Dataset

- We will be illustrating various packages using a single data set
- Joshua Correll’s **police officer’s dilemma** task
  - psych.colorado.edu/~jclab/FPST.html
- We will build this task later on (twice!)



Task: to shoot or to not shoot

Design

		Race	
		White	Black
Object	Gun		
	Not gun		

Dataset

- We will present multiple images per “cell” of this design
- We will collect both responses (accuracy) and reaction times
- At the beginning of the experiment, we will ask for subjects’ ages

Outline

1. Overview
2. Ways of using Python
3. Python basics
4. Data set overview
5. Data wrangling
6. Statistics
7. Plotting
8. Experiment creation

APS Workshop: **Introduction to Python**
San Francisco, CA, 24 May 2018

Data Wrangling

Christian C. Luhmann
Stony Brook University

Wrangling

- Data wrangling, sometimes referred to as data munging, is the process of transforming and mapping data from one “raw” data form into another format with the intent of making it more appropriate and valuable for a variety of downstream purposes such as analytics.

- This may include further munging, data visualization, data aggregation, training a statistical model, as well as many other potential uses.

en.wikipedia.org/wiki/Data_wrangling

Relevant Packages

- numpy

- pandas

- matplotlib

Relevant Packages

- numpy
  - Matrix representation
  - Linear algebra
  - Fast

numpy

4.1	3.4	2.6
12.6	8.1	1.2
6.2	10.4	5.8

dtype = float

numpy

4	3	2
12	8	1
6	10	5

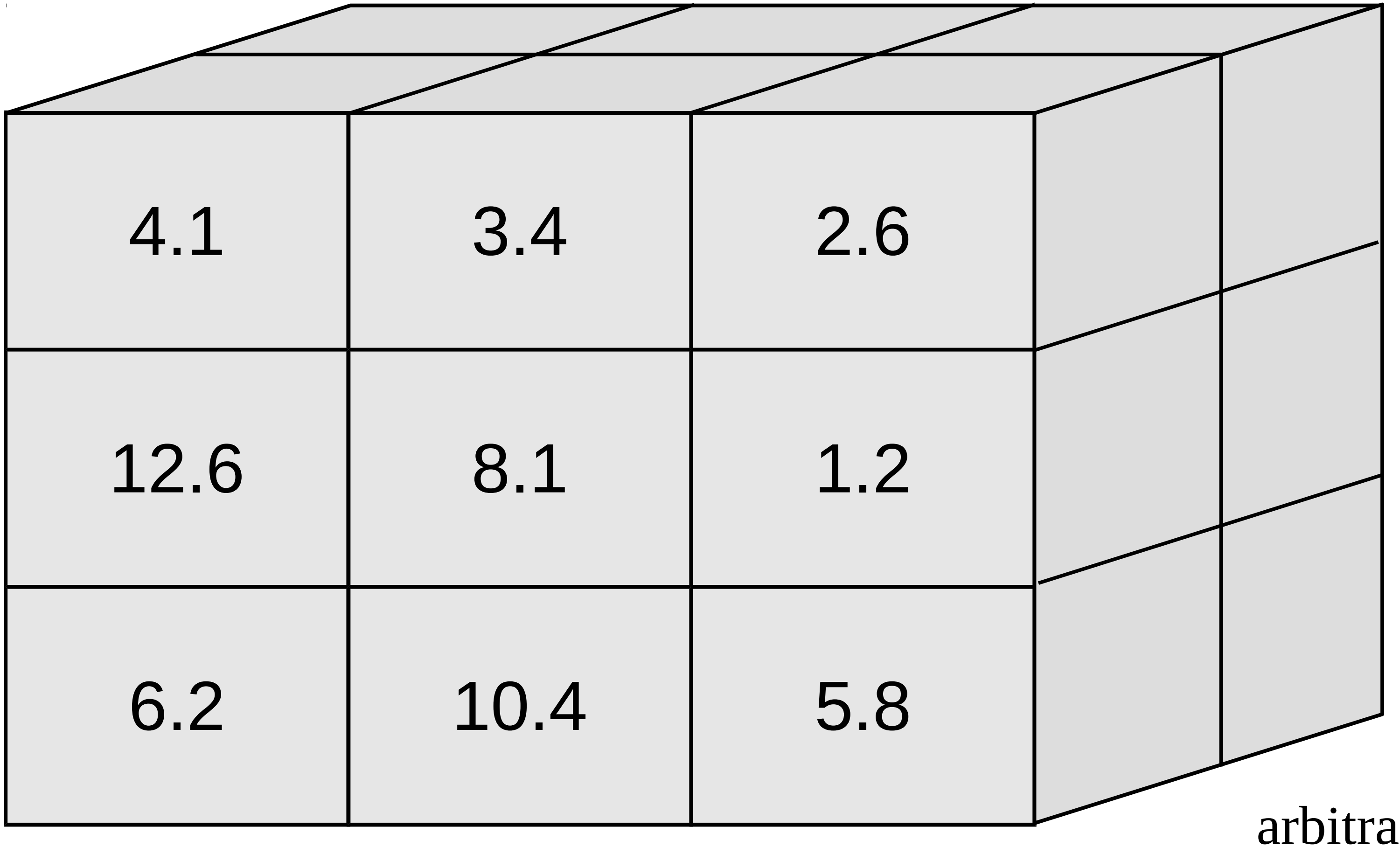
dtype = int32

numpy

4+9j	3+6j	2+4j
12+4j	8+9j	1+8j
6+8j	10+6j	5+2j

dtype = complex

numpy



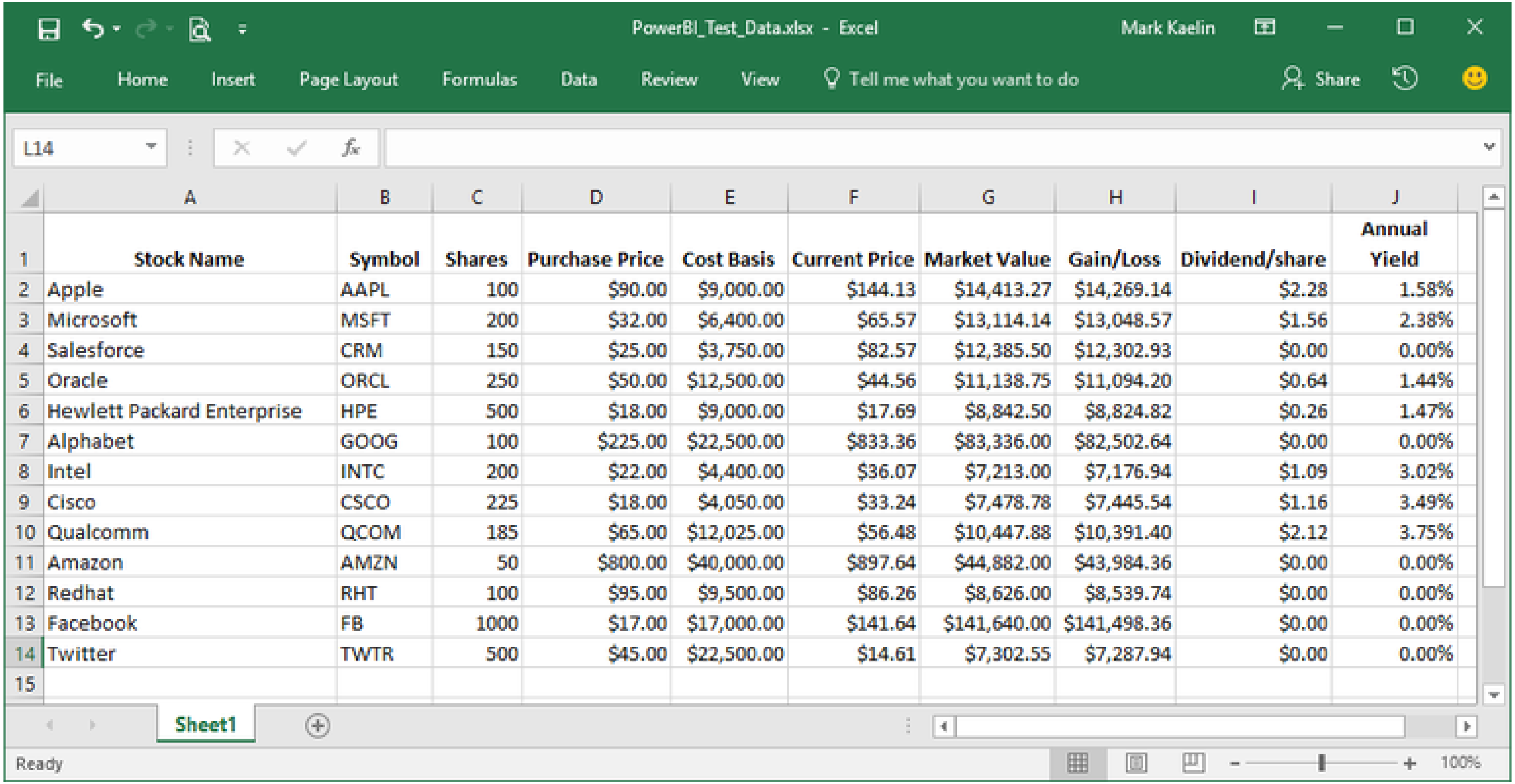
4.1	3.4	2.6
12.6	8.1	1.2
6.2	10.4	5.8

arbitrary # of dims

Relevant Packages

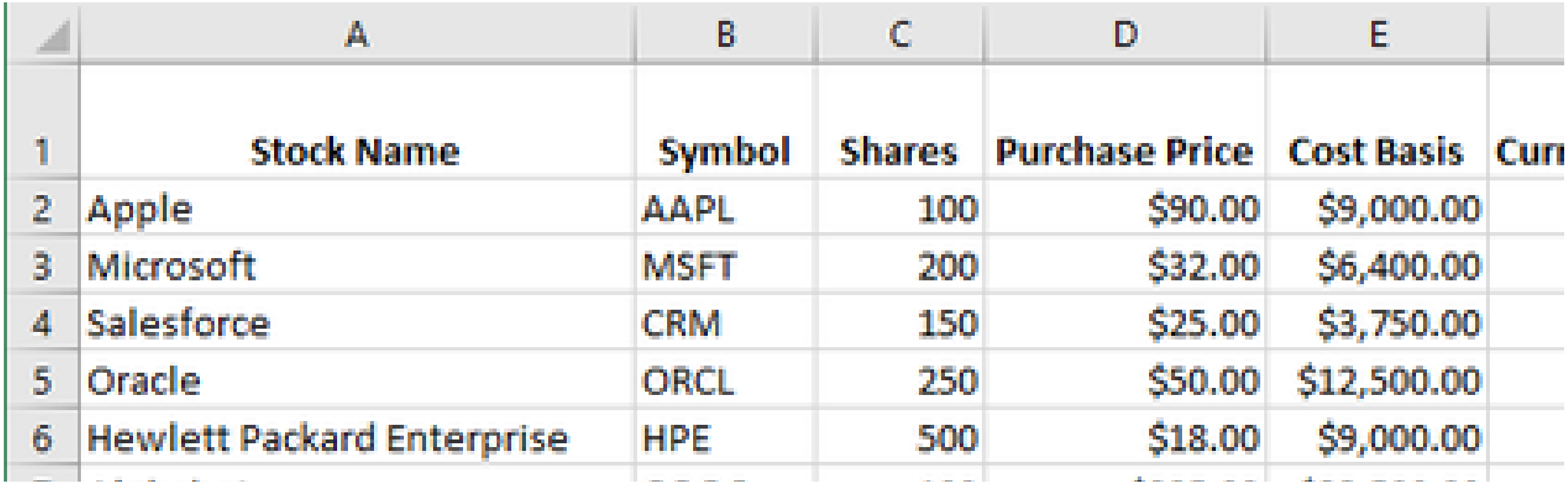
- numpy
  - Matrix representation
  - Linear algebra
  - Fast
- pandas
  - R-style dataframe
  - Best for a mixture of heterogenous data types (e.g., subject #, name, DOB)

Dataframe



	A	B	C	D	E	F	G	H	I	J
1	Stock Name	Symbol	Shares	Purchase Price	Cost Basis	Current Price	Market Value	Gain/Loss	Dividend/share	Annual Yield
2	Apple	AAPL	100	$90.00	$9,000.00	$144.13	$14,413.27	$14,269.14	$2.28	1.58%
3	Microsoft	MSFT	200	$32.00	$6,400.00	$65.57	$13,114.14	$13,048.57	$1.56	2.38%
4	Salesforce	CRM	150	$25.00	$3,750.00	$82.57	$12,385.50	$12,302.93	$0.00	0.00%
5	Oracle	ORCL	250	$50.00	$12,500.00	$44.56	$11,138.75	$11,094.20	$0.64	1.44%
6	Hewlett Packard Enterprise	HPE	500	$18.00	$9,000.00	$17.69	$8,842.50	$8,824.82	$0.26	1.47%
7	Alphabet	GOOG	100	$225.00	$22,500.00	$833.36	$83,336.00	$82,502.64	$0.00	0.00%
8	Intel	INTC	200	$22.00	$4,400.00	$36.07	$7,213.00	$7,176.94	$1.09	3.02%
9	Cisco	CSCO	225	$18.00	$4,050.00	$33.24	$7,478.78	$7,445.54	$1.16	3.49%
10	Qualcomm	QCOM	185	$65.00	$12,025.00	$56.48	$10,447.88	$10,391.40	$2.12	3.75%
11	Amazon	AMZN	50	$800.00	$40,000.00	$897.64	$44,882.00	$43,984.36	$0.00	0.00%
12	Redhat	RHT	100	$95.00	$9,500.00	$86.26	$8,626.00	$8,539.74	$0.00	0.00%
13	Facebook	FB	1000	$17.00	$17,000.00	$141.64	$141,640.00	$141,498.36	$0.00	0.00%
14	Twitter	TWTR	500	$45.00	$22,500.00	$14.61	$7,302.55	$7,287.94	$0.00	0.00%
15										

Dataframe



	A	B	C	D	E	
1	Stock Name	Symbol	Shares	Purchase Price	Cost Basis	Cur
2	Apple	AAPL	100	$90.00	$9,000.00	
3	Microsoft	MSFT	200	$32.00	$6,400.00	
4	Salesforce	CRM	150	$25.00	$3,750.00	
5	Oracle	ORCL	250	$50.00	$12,500.00	
6	Hewlett Packard Enterprise	HPE	500	$18.00	$9,000.00	

Relevant Packages

- numpy
  - Matrix representation
  - Linear algebra
  - Fast
- pandas
  - R-style dataframe
  - Best for a mixture of heterogenous data types (e.g., subject #, name, DOB)
  - Lots of slicing and dicing options
- matplotlib
  - Matlab-style plotting

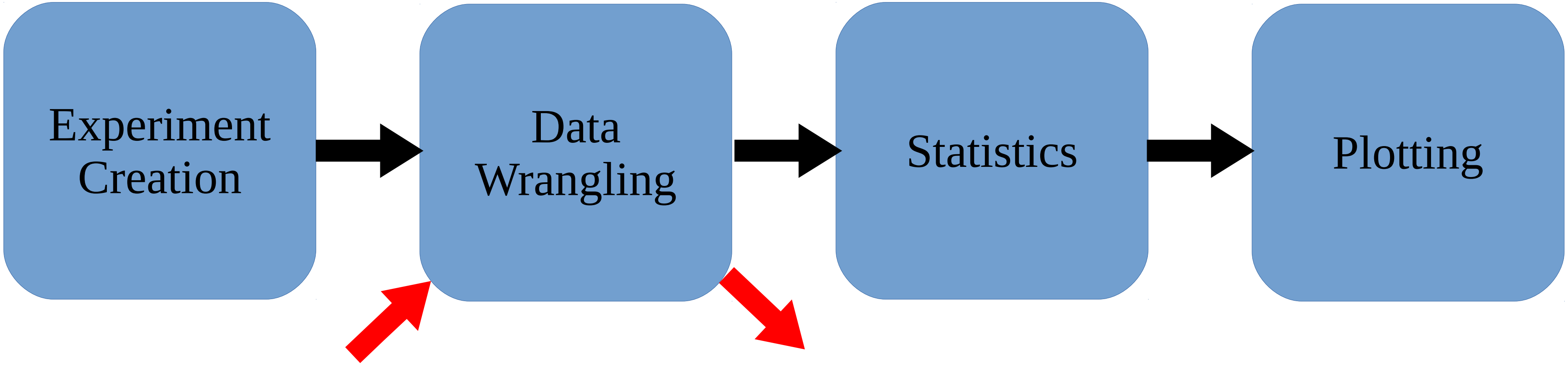
Wrangling

So let’s go wrangle some data

Pandas

- Pandas can read/write a variety of data formats...
  - CSV
  - JSON
  - HTML
  - Local clipboard
  - MS Excel
  - HDF5 Format
  - Feather Format
  - Parquet Format
  - Msgpack
  - Stata
  - SAS (read only)
  - Python Pickle Format
  - SQL
  - Google Big Query

The Pipeline



Experiment Creation

Data Wrangling

Statistics

Plotting

Take-homes

- Hopefully you have now…
-
- Seen data wrangling capabilities of Python and associated packages provide

Finish me

Outline

1. Overview
2. Ways of using Python
3. Python basics
4. Data set overview
5. Data wrangling
6. Statistics
7. Plotting
8. Experiment creation

APS Workshop: **Introduction to Python**
San Francisco, CA, 24 May 2018

Statistics

Christian C. Luhmann
Stony Brook University

Relevant Packages

- scipy
  - Distributions
  - Simple stats (e.g., *t*, χ^2, *z*, *r*, 1-way ANOVA)
- statsmodels
- pymc3
- bambi

Relevant Packages

- scipy
  - Distributions
  - Simple stats (e.g., *t*, χ^2, *z*, *r*, 1-way ANOVA)
- statsmodels
  - More complex stats (e.g., GLM, mixed linear models, survival analysis)
- pymc3
- bambi

Relevant Packages

- scipy
  - Distributions
  - Simple stats (e.g., *t*, χ^2, *z*, *r*, 1-way ANOVA)
- statsmodels
  - More complex stats (e.g., GLM, mixed models, survival analysis)
- pymc3
  - Full-featured Bayesian modeling (think Stan)
- bambi

Relevant Packages

- scipy
  - Distributions
  - Simple stats (e.g., *t*, χ^2, *z*, *r*, 1-way ANOVA)
- statsmodels
  - More complex stats (e.g., GLM, mixed models, survival analysis)
- pymc3
  - Full-featured Bayesian modeling (think Stan)
- bambi
  - Streamlined, Bayesian GLMs built on top of pymc3 (think brms?)

Statistics

Let’s go do some stats!

scikit-learn

- Machine learning
  - Supervised
    - Classification (e.g., GLM, LDA, SVM, random forests)
    - Regression (e.g., ridge, lasso)
  - Unsupervised
    - Clustering (k-means)
    - Dimension reduction (e.g., PCA)

- All the extras needed to fit, evaluate, and use these tools

Take-homes

- Hopefully you have now learned...

Finish me

Outline

1. Overview
2. Ways of using Python
3. Python basics
4. Data set overview
5. Data wrangling
6. Statistics
7. Plotting
8. Experiment creation

APS Workshop: **Introduction to Python**
San Francisco, CA, 24 May 2018

Plotting

Christian C. Luhmann
Stony Brook University

Relevant Packages

- pandas
- matplotlib
- seaborn
- bokeh

Plotting

Let’s go plot some stuff!

Take-homes

- Hopefully you have now learned...

Outline

1. Overview
2. Ways of using Python
3. Python basics
4. Data set overview
5. Data wrangling
6. Statistics
7. Plotting
8. Experiment creation

APS Workshop: **Introduction to Python**
San Francisco, CA, 24 May 2018

Experiment Creation

Christian C. Luhmann
Stony Brook University

Relevant Packages

- psychopy
- pyserial
- pyparallel
- pyopengl
- pyglet
- moviepy
- pillow

PsychoPy

- Originally created by Jon Pierce
- Initiated as a python replacement for Psychtoolbox
  - But it has grown in to much, much more
- It is recommended to install the Psychopy standalone alongside any data-centric python installation you might have (e.g., Anaconda)
- The “standalone” version of PsychoPy includes...
  - Python (2.x or, as of April 2018, 3.x)
  - PsychoPy
  - all the other packages required by PsychoPy

PsychoPy

- Coder
  - Coder is PsychoPy’s IDE (like Spyder)
  - Provides already-written demos and examples
- Builder
  - No programming required
  - Experiments are built using a graphical interface
  - (A couple) Demos/examples already built
- Can also use PsychoPy as a regular package within python (2.x or 3.x)

PsychoPy

- Running PsychoPy (using the shortcut created during installation) should get this…

PsychoPy

Finish me

Experiment Creation

Let’s go **build** some experiments!

Take-homes

- Hopefully you have now learned...

Finish me

Outline

1. Overview
2. Ways of using Python
3. Python basics
4. Data set overview
5. Data wrangling
6. Statistics
7. Plotting
8. Experiment creation