# Week 2

# Outcomes for this evening

By the end of today's class, you should be able to do the following:

- Describe examples of data structures like Scalars, Lists, Sets, Dictionaries
- Explain the difference between vectorized and elementwise operations
- Create and call functions in Python
- Load data into Pandas and create a scatter plot
- Decompose a complex function into multiple simpler functions

Lots of words, so we will need definitions

# I won't be able to teach you all of Python

# Resources for learning Python

- Online
  - Text – blue underscored text is hyperlinked in this presentation
    - https://nealcaren.github.io/python-tutorials/
  - Videos on Coursera, Udemy
- Books: see PDFs posted on the Syllabus
- Google/StackOverflow: You don't need to know everything to get started.
  Start…continue until stuck…google…unstuck…continue until stuck…Done
- Your instructor

https://brohrer.github.io/imposter_syndrome.html
Linked from https://www.kdnuggets.com/2017/09/data-science-imposter-syndrome.html

- Diversity of Software in Data Science
- Use of Python
- EDA using Jupyter + Pandas
- Visualization of data
- Tips from the Trenches
- Homework

# Survey of "standard" software for Data Science

*(not comprehensive)*

- Hive
- MapReduce
- Pig
- JavaScript
- Java
- Orange
- Rapid Miner
- SAS
  - Enterprise Miner
  - JMP
- Online services, ie from Azure, AWS, Google Compute
- Matlab
- Python
- R
- bash
- Tensor Flow
- Jupyter
- Julia
- Spark
- SQL
- Excel
  - PowerBI and other plugins
- Tableau

# Psychological consequences of diversity

- Intimidation (fear) due to knowing you don't know

Doctors: Googling stuff online does not make you a doctor.

Programmers:

# Psychological consequences of diversity

- Intimidation (fear) due to knowing you don't know

- When interacting with other people, lack of overlap causes issues
  - Communication
  - How you think about the problem
  - What the challenges are (cost, latency, scale)
  - How to collaborate

--> Each of these is a negotiation with other humans

Since 1991

# Simplify by focusing on just Python

Python 3 is 10+ years old!

Python 3.0 final was released on December 3rd, 2008

# Survey of "standard" Python packages for Data Science

*(not comprehensive)*

- Dask
- Scikit-learn
- Numpy
- Scipy
- Pandas
- Matplotlib
- Python Imaging Library (PIL)
- NLTK
- Beautifulsoup
- Statsmodels
- Seaborn
- PyTorch
- Keras
- Theano
- Gensim
- Plotly
- Bokeh
- PySpark
- Scrapy

*Optional challenge*: How many Python packages are there total?
Yes, this question is ill-defined. I'm happy to iterate with you after class

# Jargon alert!

- ***Script*** = a file with extension .py containing Python code.
- *Application*, aka *Analytic*, aka *program* = recipe of instructions
- ***Library*** = generic term for code that was designed with the aim of being usable by many applications.
- ***Module*** = a file with extension .py containing function definitions which can be referenced by other scripts. Use `import` to load these functions.
- ***Package*** = a collection of modules.

All packages are modules, but not all modules are packages.

Overloading terms is common.

If you are confused during the semester, raise your hand during class or see me after class or submit an anonymous question

# Python package management

Two package managers: conda and pip

- Conda is equivalent to pip+virtualenv in terms of capability.

*Difference*: conda is used for package management outside Python; pip is only for Python

| | conda | pip |
|---|---|---|
| install python package | ✅ | ✅ |
| create virtual environment | ✅, built-in | ❌, requires `virtualenv` or `venv` |
| package format | `.tar.bz2`, `.conda` | `.whl`, `.tar.gz` |
| manages | binaries | wheel or source |
| can require compilers | ❌ | ✅ |
| package types | any | Python-only |
| dependency checks | ✅ | ❌ |
| package sources | Anaconda repo and Anaconda cloud | PyPI |

# Virtues of a Data Scientist

Using libraries
enables laziness.

see http://blog.teamtreehouse.com/the-programmers-virtues

- ~~Building up from basics: the terminal~~
- ~~Diversity of Software in Data Science~~
- Use of Python
- EDA using Jupyter + Pandas
- Tips from the Trenches
- Homework

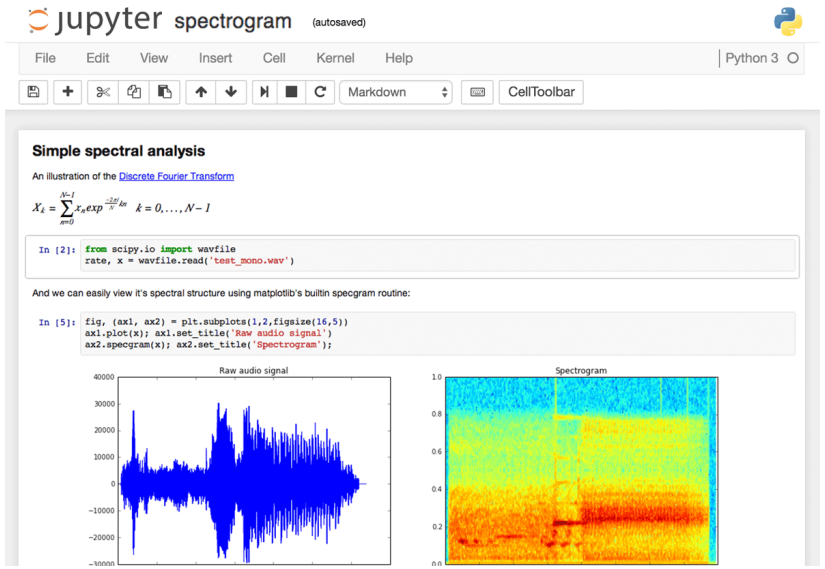
# How to interact with Python

- REPL: interactive command prompt

```
Bens-Air:~ benpayne$ python3
Python 3.5.1 (v3.5.1:37a07cee5969, Dec  5 2015, 21:1
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwi
Type "help", "copyright", "credits" or "license" for
>>>
```

- Scripts: files with .py extension
  - Text editors with syntax highlighting
  - IDEs like PyCharm with autocompletion

- Jupyter Notebook (via web browser)



# Python in your computer's terminal

- REPL = Read–eval–print loop

(There are improvements: https://bpython-interpreter.org/ )

*Demo*: show Python 3 REPL

# *Shown in Python3 Interpreter (REPL)*

- Returns expected results, ie 5, 5+4, x = 5+4, x == 9, x == 10, 'hello'

Lists:

- `mylist = [4, 5, 6, 'hello', 5, 2]`
- Accessing an element: `mylist[0]`
- `len(mylist)`
- `mylist[12]`
- No warning when over-writing: `mylist[0] = [3, 5, 9]`

# *Shown in Python3 Interpreter (REPL)*

- Dictionaries:
- mydict = {'four':'three'}
- mydict['four']
- mydict = {'k': 'v'}
- mydict['four']
- mydict[0]
- len(mydict)
- mydict = {'this key': 'a value', 'another key': 'another value'};
- mydict = {4:3, 5:4, 4:2}
- mydict = { 'a': 5, 'b':[3,5]}
- mydict = {[3, 4]:2, 4:3}
- mydict = { 'a':{4:2, 'b':9}, 6: 3}

# Programming exercise using the interpreter

1. Create a dictionary containing
   - when you ate today (the key)
   - what you ate today (the value)

2. Once completed, enter the dictionary in a Python3 REPL

3. Verify that you can select an entry by key
4. Measure the length of the dictionary

*Activity*: solo programming

# Example dictionary

```
>>> mymeals={'breakfast':'apple', 'lunch':['beans', 'salmon'],
  'dinner':['melon', 'salad', 'milk'}
  File "<stdin>", line 1
    mymeals={'breakfast':'apple', 'lunch':['beans', 'salmon'],
  'dinner':['melon', 'salad', 'milk'}
                                      ^
SyntaxError: invalid syntax
>>> mymeals={'breakfast':'apple', 'lunch':['beans', 'salmon'],
  'dinner':['melon', 'salad', 'milk']}

>>> mymeals['lunch']
['beans', 'salmon']
```

# Generic Computing Language Essentials

- Variable assignment
- Control statements (if, else)
- Loops (while, for)
- Sets, tuples
- Functions

# Python functions bundle code

```
def myfunc(input):
    # transform input to value
    return value
```

If a line of code is a sentence,
then a function is a paragraph

*Demo*: show Python3 REPL

# *Shown in Python3 Interpreter (REPL)*

```
a = 5
def my_func(input)
    a = 6
    print(a)
    return
my_func('hello')
print(a)
```

# tuples_arrays_functions.ipynb

# Idiosyncrasies of Python

- Spaces – see style guide https://www.python.org/dev/peps/pep-0008/

--> 4 spaces per indentation level

- no explicit type definitions in code (dynamic typing; types are resolved at runtime)
- Python indexes from 0. This isn't consistent across languages.

*Pro-tip*: use a linter like pylint and flake8

# Memorizing the nuances of a specific language

--> form pairs; each pair gets terms and definitions

*Activity*: match terms + definitions cards

Answers will be verified by consensus

When returning paper, please return term and definition in paired order:

(term1, def1, term2, def2, term3, def3, …)

Source:

https://www.brainscape.com/packs/python-1786943

- for more, see https://www.brainscape.com/subjects/python
- see also https://quizlet.com/subject/python/

# Quiz on Python data structures

Given a list,

```
my_list = ['a', 'b', 'c', 'd']
```

*Do not shout out the answer. We will vote.*

What would the command

```
len(my_list)
```

return?

# Quiz on Python data structures

Given a list,

```
my_list = ['a', 'b', 'c', 'd']
```

What would the command

```
len(my_list)
```

return?

Vote

3

4

NOT SURE

# Quiz on Python data structures

Given a list,

```
my_list = ['a', 'b', 'c', 'd']
```

*1* *2* *3* *4*

What would the command

```
len(my_list)
```

return? `len` = length of list

~~3~~

4

~~NOT SURE~~

# Quiz on Python data structures

Given a list,

```
my_list = ['a', 'b', 'c', 'd']
```

What would the command

```
my_list[1]
```

return?

a

b

NOT SURE

# Quiz on Python data structures

Given a list,

```
                0      1      2      3
my_list = ['a', 'b', 'c', 'd']
```

What would the command

```
my_list[1]
```

return?

Python indexes from 0

~~a~~

b

~~NOT SURE~~

# Essentials of Python for Exploratory Data Analysis (EDA)

*Covered so far*:

- Loading files, ie CSV, JSON, XML (see previous week)
- Use of the REPL
- Data structures like lists, dictionaries

*Pro-tip*: What variables exist in the interpreter's memory? `dir()`

- ~~Diversity of Software in Data Science~~
- ~~Use of Python~~
- EDA using Jupyter + Pandas
- Visualization of data
- Tips from the Trenches
- Homework

- Analyze video to detect 60 million faces

http://willcrichton.net/notes/rapid-prototyping-data-science-jupyter/ Jupyter for more than just learning and small scale

- How Netflix uses Jupyter notebooks

https://medium.com/netflix-techblog/notebook-innovation-591ee3221233

# Idiosyncrasies of Jupyter notebooks

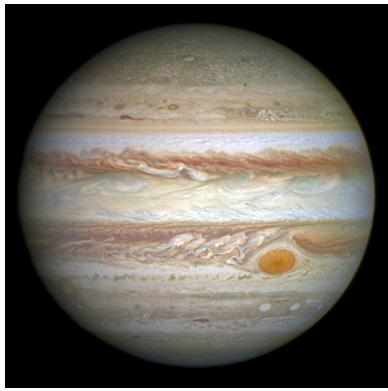
- Order of execution – designed to support non-linear exploration
- Has syntax highlighting
- Issues with Jupyter notebooks (comments)

- Why Jupyter is data scientists' computational notebook of choice (comments)

# Pandas

- A module for Python
- Widely used in Data Science
- Series
- DataFrames for tables
- Similar to data.frame in R
- Uses NumPy

- Many tutorials and documentation
- Fancy operations like
  - groupby
  - Map, apply



# Bowling data from a webpage

*Activity*: sketch your expectation
for time vs participation

## Data source:

https://www.bowl.com/Open_Championships/Open_Championships_Home/Past_Results_and_History/
Linked from https://www.bowl.com/records/

## Notebook:

Bowling_visualization*.ipynb

- ~~Diversity of Software in Data Science~~
- ~~Use of Python~~
- ~~EDA using Jupyter + Pandas~~
- Visualization of data
- Tips from the Trenches
- Homework

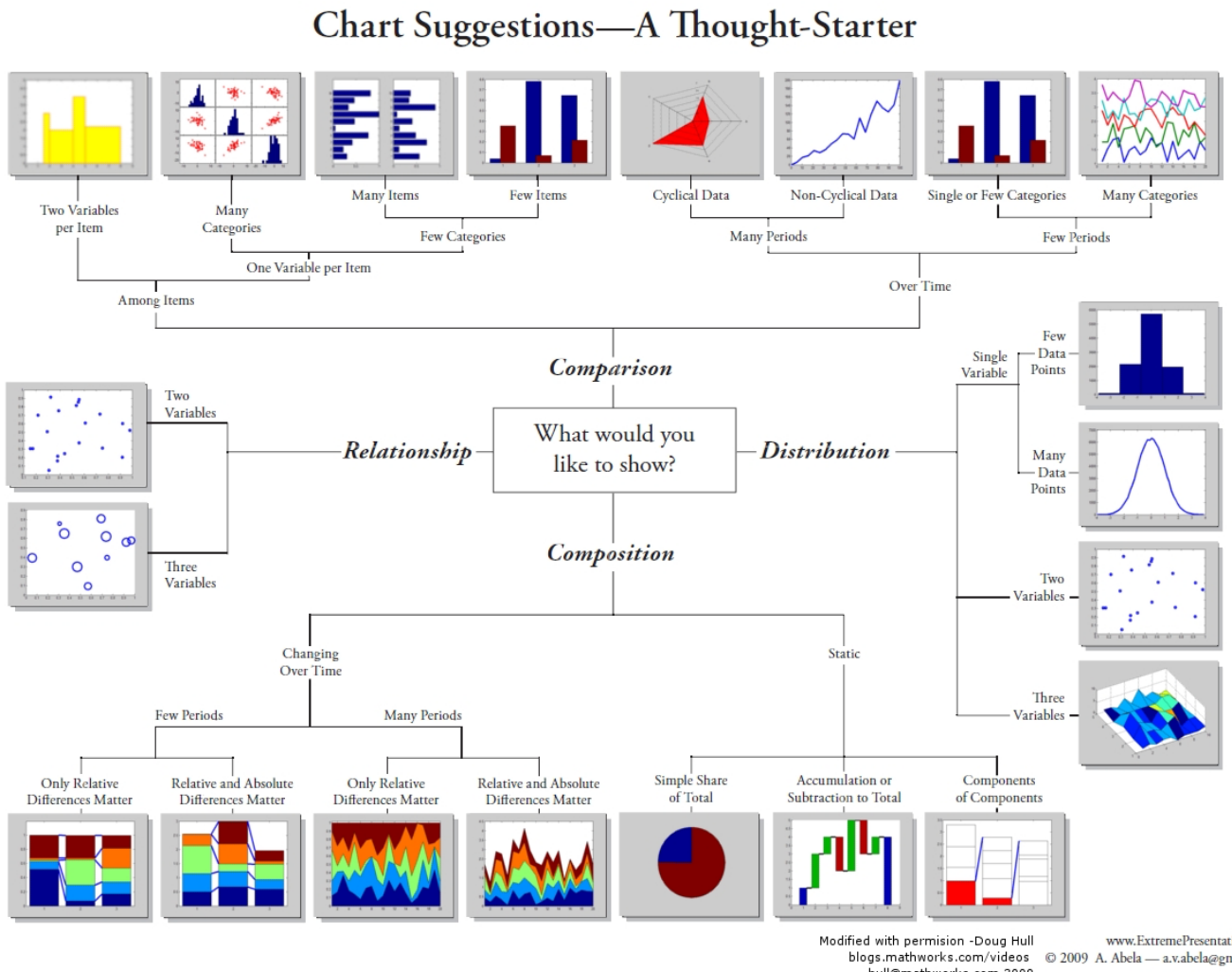
# Principles of Data Visualization

"Graphical excellence is that which gives to the viewer the greatest number of ideas in the shortest time with the least ink in the smallest space"

-- E. Tufte [citation]

# Options for Visualizations

- There are many options: https://datavizcatalogue.com/
- There are many ways to go wrong: http://viz.wtf/
- How to know when to use which?




https://blogs.mathworks.com/videos/2012/04/26/choosing-a-visualization-in-matlab/

# Data visualization may be useful, but it's old and no longer evolving

# Data visualization may be useful, but it's also no longer living

WRONG

# Visualization of data is advancing in multiple domains

Large-Scale data:

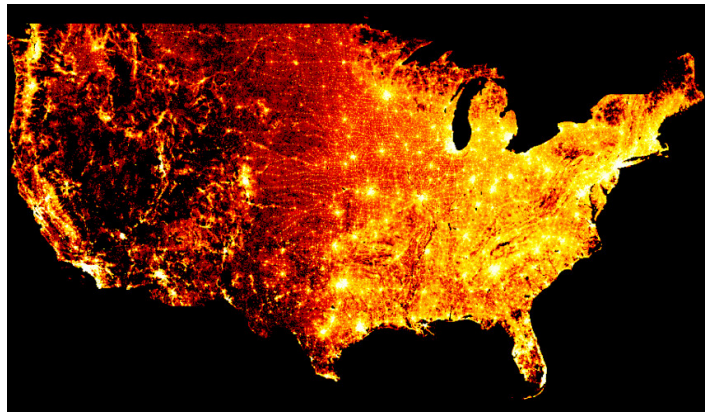
- http://datashader.org/ 300 million points of data (one per person in the USA) from the 2010 census

Grammar of graphics

- https://vega.github.io/vega/
- https://altair-viz.github.io/

Interactivity

- https://holoviews.org/



See https://www.youtube.com/watch?v=ms29ZPUKxbU

See https://www.youtube.com/watch?v=0jhUivliNSo

# Visualization of data is advancing in multiple domains

Large-Scale data:

- http://datashader.org/

Grammar of graphics

- https://vega.github.io/vega/
- https://altair-viz.github.io/

Interactivity enables EDA

- https://holoviews.org/

See https://www.youtube.com/watch?v=ms29ZPUKxbU

See https://www.youtube.com/watch?v=0jhUivliNSo

# Start with the basics

- Matplotlib - open source Python visualization similar to Matlab
  - To get started, find what you're interested in the gallery

- Seaborn - built on top of matplotlib and closely integrated with pandas data structures.
  - tutorial

- ~~Diversity of Software in Data Science~~
- ~~Use of Python~~
- ~~EDA using Jupyter + Pandas~~
- Tips from the Trenches
- Homework

# Not all CSVs are equivalent

There are best practices for what good tabular data looks like

- Each variable must have its own column.
- Each observation must have its own row.
- Each value must have i[illegible]

CUSTOMERS

| businessName | address | phone | order1 | order2 |
|---|---|---|---|---|
| Bob's Diner | 14 Rialto St.<br>Boston, MA 02119 | 617-447-0106<br>617-499-0976 | 4 doz. handbraided Guatemalan placemats | 8 basic curtains (floral) and curtain rods |
| Turpelo Cleaners | 205 South St.<br>Roxbury, MA 02334 | 617-547-0098 | 1 basic curtains (floral) and curtain rods | |
| … | … | … | … | … |

More observations on best practices

- https://cran.r-project.org/web/packages/tidyverse/vignettes/manifesto.html
- http://r4ds.had.co.nz/tidy-data.html

# Data cleaning as a showcase of skill

Example of data cleanup articles:

- http://www.developintelligence.com/blog/2017/08/data-cleaning-pandas-python/

These usually don't bother to capture the frustration of exploration.

# Real data is real dirty

http://usbcongress.http.internapcdn.net/usbcongress/bowl/recordsstats/pdfs/PTIndividualRecordsState.pdf

from

https://www.bowl.com/records/

# String manipulation

```
this_str='a long sentence is fun'
another_str=this_str + ' to write.'
type(another_str.split(' '))
str_as_list = another_str.split(' ')
```

Consider a string a list of characters, except **strings are immutable**. Changing a string does not modify the string. It creates a new one.

Strings are sliceable. Slicing a string gives you a new string from one point in the string, backwards or forwards, to another point, by given increments

source and this essay

# Buzzwords as indicators

- **The Cloud**
- **Machine Learning**
- **Artificial Intelligence**
- **Big Data**
- **Predictive Modeling**
- **Labeled Data**
- EDA
- ETL
- **Training models**
- **Deep Neural Network**
- **Moonshot**
- **Structured data**

*The claim*: these words are not used by normal people

# Question the speaker

- What do you mean by that phrase?
  - Is the definition shared by speaker and audience?
- What is an example of that?
  - What is the speaker's depth of experience?
- What is the speaker's expectation of the audience?
  - What depth is expected for audience?

# Pandas: Series and Dataframes

- Series = one-dimensional labeled array capable of holding data of any type (integer, string, float, python objects, etc.). The axis labels are collectively called index.
  - https://www.tutorialspoint.com/python_pandas/python_pandas_series.htm
  - Based on Numpy array

- Dataframe = Two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). Arithmetic operations align on both row and column labels. Can be thought of as a dict-like container for Series objects. The primary pandas data structure.
  - https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.html

# *Pro-tip*: When to write software and for how long

- Writing a script takes time and focus
- Flow state
  - Avoid interruptions and context switching

# When to persist,
# When to change course,
# When to seek help


Try attacking the challenge for 30 minutes
Then seek help or do something else for a while

https://en.wikipedia.org/wiki/Pomodoro_Technique

- ~~Building up from basics: the terminal~~
- ~~Diversity of Software in Data Science~~
- ~~Use of Python~~
- ~~EDA using Jupyter + Pandas~~
- ~~Tips from the Trenches~~
- Homework

# Watch https://www.youtube.com/watch?v=V0ucZ4ctof8

*Summary of the video*:

- Goal: Find the Mario Maker level with the most plays and the most star ratings.
- Problem: Collecting the data. Nintendo does not offer it in a giant spreadsheet. You can pull up a query that shows 100 or whatever at a time, but we need 10^6 rows to achieve the goal.
- At around 4:35 or so he breaks down the query, explains how someone wrote a script to scrape the website (a polite script too, which is smart - it didn't overwhelm the servers or cause the IP address to get blocked) and then briefly explains some of the other neat things you can find within the data set before diving in to address the goal.

# Homework for Week 2, Assignments 1 and 2

- *Assignment*:  Homework-Week2.ipynb
- *Assignment*: Download data in CSV format from one of the following sites: here or here. Load the data into a Pandas dataframe. Use Pandas to count the number of rows of data and number of columns. Your selection of a CSV should have at least 2 columns and at least 10 rows. Include a link to the data source in the Juptyer Notebook you submit.
- Name file Homework-Week2-pandas.ipynb

Where to submit??? Github.com

# Not enough homework?

Challenges abound:

- https://www.hackerrank.com/challenges/
- https://codesignal.com/interview-practice/
- https://www.codewars.com/

End of class

# Questions?

# Comments?