Data 601 @ UMBC
Class 2: Python

Data 601

September 6, 2018

From Week 1: Reading summarization

Find someone who was assigned the other article

- Tell the other person one idea presented last week
- Listen your partner's recollection from last week
- Describe what you learned from reading the article
- Ask what they learned reading their article

Activity: talk with a partner

Answer to anonymous question

- How will tests be structured?

Rubric is subject to change

aspect	deficient: 0 points	points for sufficient	description of sufficient
project proposal submitted on time	missed deadline	1	submitted prior to deadline
project proposal submitted using Blackboard	submitted via email or paper	1	submitted using Blackboard
project proposal explained plan for proposal	lacks detail	1	clear plan
project proposal includes alternative plan	no alternative provided	1	alternative plan provided
project proposal relevant to Data 601 objectives	complexity below objective or out of scope for 601	1	appropriate complexity and within scope
data gathered from	no source cited	1	explanation of where data came from

aspect	deficient: 0 points	points for sufficient	description of sufficient
project proposal submitted on time	missed deadline	1	submitted prior to deadline
project proposal submitted using Blackboard	submitted via email or paper	1	submitted using Blackboard
project proposal explained plan for proposal	lacks detail	1	clear plan
project proposal includes alternative plan	no alternative provided	1	alternative plan provided
project proposal relevant to Data 601 objectives	complexity below objective or out of scope for 601	1	appropriate complexity and within scope
data gathered from documented source	no source cited	1	explanation of where data came from
data quantity is appropriate	too little data (manual analysis is feasible) or excessive data based on compute resource	1	more than manual analysis, and fits on your compute device
data gathered in a manner compliant with source constraints	violates source's policy regarding access	1	explanation of why gathering is acceptable
data inconsistences are documented	no explanation of issues	1	problems in data are identified
data cleaned (using Python) if necessary	data has inconsistencies not corrected using Python	1	data has no inconsistencies resolvable in Python
documentation on what cleaning was performed	no explanation of cleaning process	1	justification for changes to data is provided
analysis of cleaned data for characteristics	no characterization of data is performed	1	relevant aspects of data set are enumerated
analysis includes at least one visualization	no visualizations present	1	at least one visualization used
visualization uses appropriate plot types	inappropriate plot type	1	valid use of histogram or scatter plot
visualization has labeled axes and caption	missing caption or axis label	1	descriptive captions and labels used
axis label for visualization includes units if appropriate	axis label lacks units	1	units, where appropriate, are present
visualization is intuitive to understand and not misleading	naive reading of plot yeilds incorrect conclusion	1	purpose of the visualization is clear to reader
code used for getting data, cleaning, analysis, and visualization is concise	dead code is present	1	only relevant code is provided
story of patterns observed	no explanation regarding correlations	1	correlations are explained
pattern predictions are made	no predictions made	1	reasonable and testable predictions are explained
test of predictions is made	predictions are left untested	1	test of predictions are provided
explain what you learned doing this project	no lessons learned are documented	1	knowledge gained is documented

Rubric is subject to change

Course schedule and outline (scope)

- Aug 30: Overview Data Science
- Sept 6: Python in Jupyter
- Sept 13: Math (stats)
- Sept 20: Regression
- Sept 27: Clustering
- Oct 4: Evaluation, cross-validation, overfitting
- Oct 11: *Substitute's choice*
- Oct 18: Getting data
- Oct 25: Automation
- Nov 1: Data cleanup
- Nov 8: Scaling up
- Nov 15: Property graphs
- Nov 22: No class (Thanksgiving)
- Nov 29: Elasticity, Cost/benefit
- Dec 6: Ethics and Legality
- Dec 13: Presentations

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
- Demonstrate use of piped functions in bash

Lots of words, so we will need definitions

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

Resources for learning Python

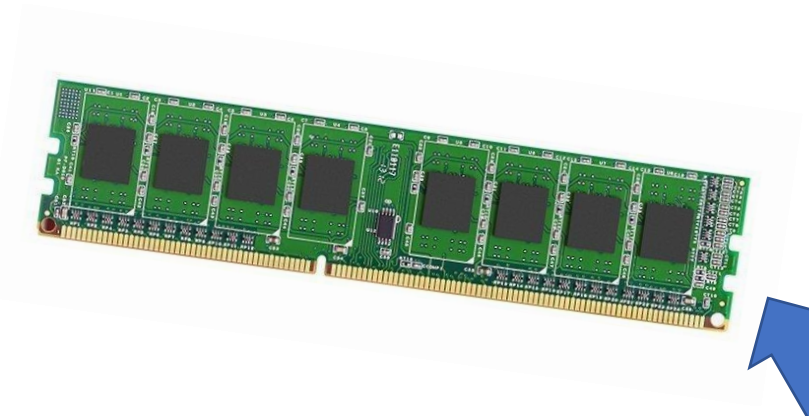
- Online
  - Text – blue underscored text is hyperlinked in this presentation
    - https://nealcaren.github.io/python-tutorials/; see also comments
  - Videos on Coursera
- Books: see PDFs posted on Blackboard
- Your instructor

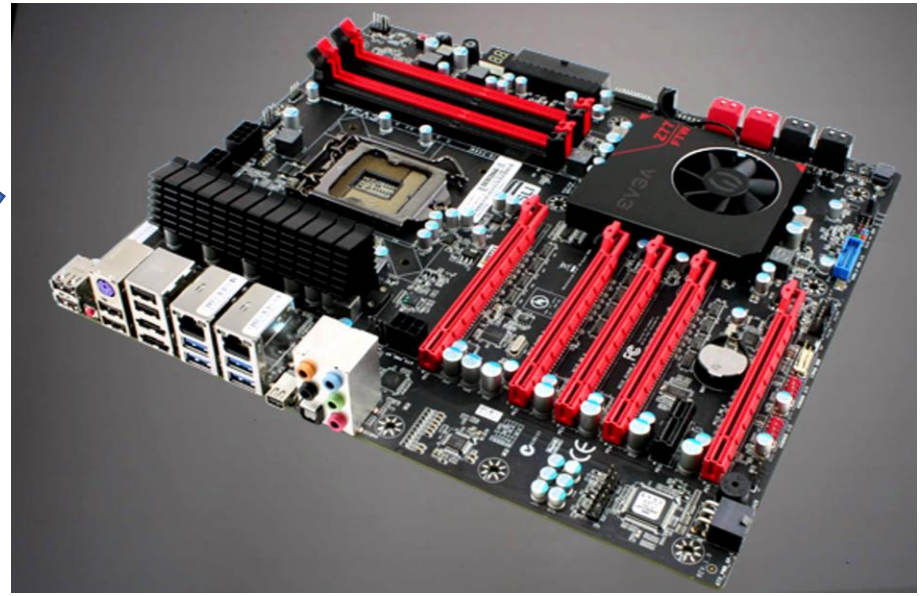
Caveat: I'm not a programmer.

https://brohrer.github.io/imposter_syndrome.html

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

Software runs on hardware





Hardware is fundamental to computing

- Hardware gets faster and cheaper due to market competition
- Constrained by Physics and by being tangible objects
- Rate of change bounded by money and logistics

- Software = recipe of instructions
- Instructions executed on hardware; hardware is the constraint
- Software evolution is fast

Minimum hardware for Data 601?

Any computer that supports Jupyter with Python 3 kernel

Quiz on Default environments

- What software for programming is available on a default installation of Windows?

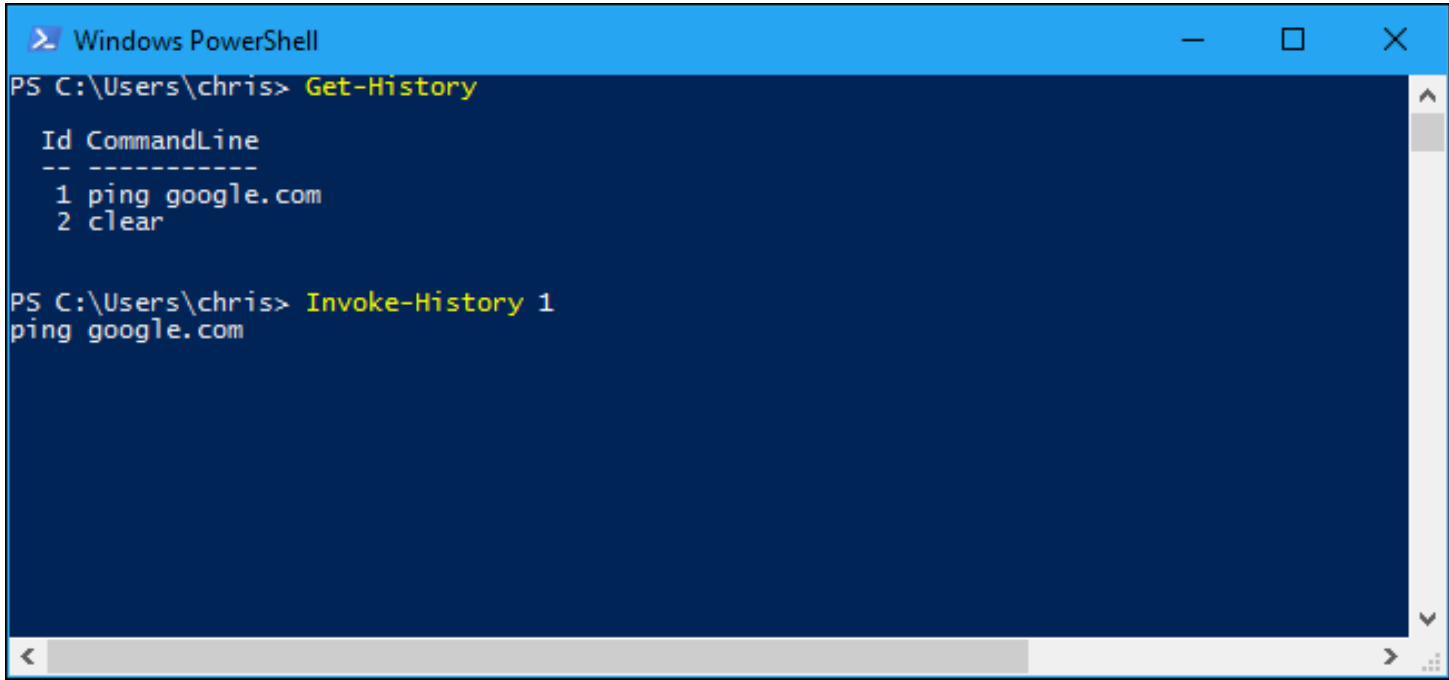
Programming tools on a default installation of Windows

- Command prompt
- Powershell
- VBS
- C# (ref1, ref2)

Using your computer's terminal

Interact with your computer via text and typing

- Linux and Mac OS X
- Windows command prompt
  - See also PowerShell



Demo: show both

Shown in bash

```
pwd
ls
echo "hello"
echo "hello, ben, another" | cut -d',' -f2
man cut
echo -e "1\n3\n2" | sort
cd umbc/fall2018/jupyter_notebooks/week1_data_formats
cat sample.csv
cat sample.csv | cut -d',' -f2 | sort
cat sample.csv | cut -d',' -f3 | sort
cat sample.csv | cut -d',' -f3 | sort | uniq
cat sample.csv | cut -d',' -f3 | sort | uniq -c
```

Shown in Powershell: cd, ls

Administrator: Windows PowerShell

```
PS C:\Users\[redacted]> pwd

Path
----
C:\Users\[redacted]


PS C:\Users\[redacted]> ls


    Directory: C:\Users\[redacted]


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----         5/11/2018   4:28 PM            .MCTranscodingSDK
d-r--          8/8/2018  12:10 PM            Contacts
d-r--         8/29/2018   3:08 PM            Desktop
d-r--         8/28/2018   7:09 PM            Documents
d-r--         8/10/2018   2:14 PM            Downloads
d-r--          8/8/2018  12:10 PM            Favorites
d-r--          8/8/2018  12:10 PM            Links
d-r--          8/8/2018  12:10 PM            Music
d-r--          8/8/2018  12:10 PM            Pictures
d-r--          8/8/2018  12:10 PM            Saved Games
d-r--          8/8/2018  12:10 PM            Searches
d-r--          8/8/2018  12:10 PM            Videos


PS C:\Users\[redacted]> _
```

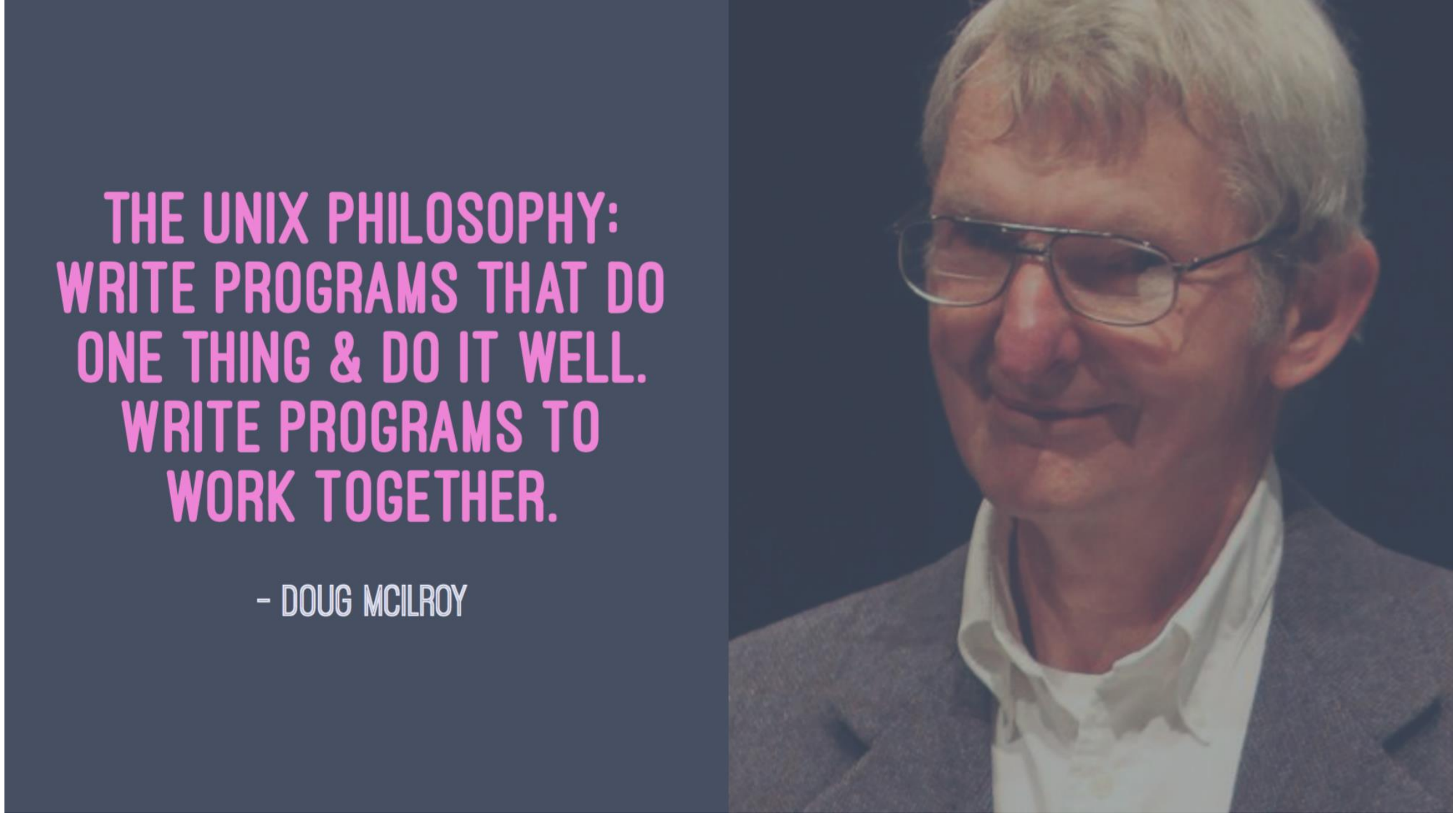
Shown in Powershell: CSV

Administrator: Windows PowerShell

```
PS C:\temp> ls


    Directory: C:\temp


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
-a---          9/2/2018   4:11 PM        125 sample.csv


PS C:\temp> Import-Csv .\sample.csv

Name                        year                        class                       grade
----                        ----                        -----                       -----
Ming                        2013                        Data 601                    B
Imgb                        2015                        Date 601                    B
Rimgw                       2012                        Data 601                    A
Wemf                        2014                        Data 602                    C


PS C:\temp>
```

Shown in Powershell: sort column

```
Administrator: Windows PowerShell
PS C:\temp> ls

    Directory: C:\temp

Mode                LastWriteTime     Length Name
----                -------------     ------ ----
-a---          9/2/2018   4:11 PM        125 sample.csv

PS C:\temp> Import-Csv .\sample.csv

Name                        year                        class                       grade
----                        ----                        -----                       -----
Ming                        2013                        Data 601                    B
Imgb                        2015                        Date 601                    B
Rimgw                       2012                        Data 601                    A
Wemf                        2014                        Data 602                    C

PS C:\temp> Import-Csv .\sample.csv | Sort-Object -Property year

Name                        year                        class                       grade
----                        ----                        -----                       -----
Rimgw                       2012                        Data 601                    A
Ming                        2013                        Data 601                    B
Wemf                        2014                        Data 602                    C
Imgb                        2015                        Date 601                    B

PS C:\temp>
```



Unix Philosophy

Running bash commands online

- https://www.tutorialspoint.com/execute_bash_online.php
- https://repl.it/
- If you have 64bit Windows 10, see Windows Subsystem for Linux

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
- 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

Psychological consequences of diversity

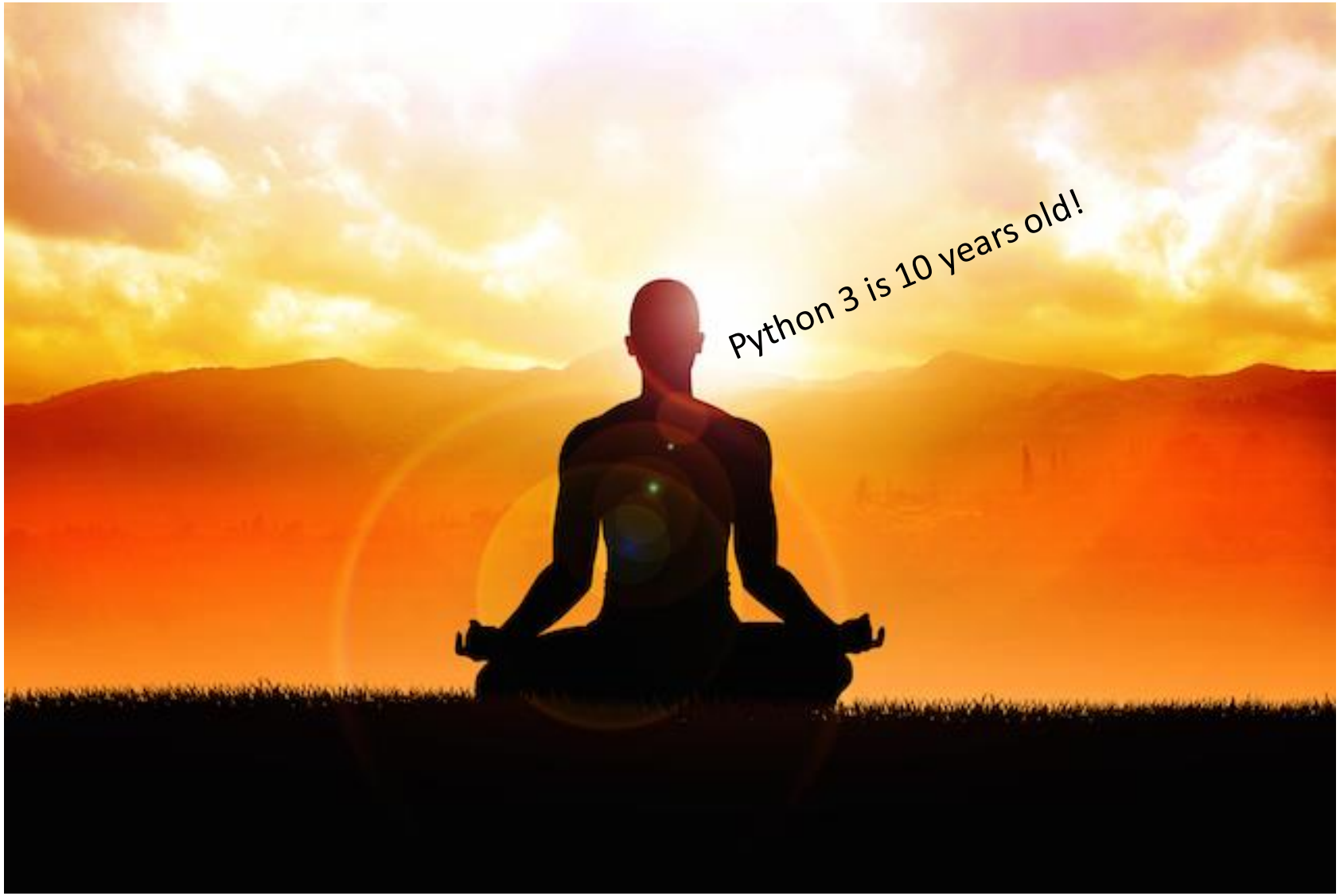
- Intimidation (fear) due to knowing you don't know

- 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

Simplify by focusing on just Python

Since 1991



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

- For this class I'll advocate for using the simplicity of conda
- I use pip

Virtues of a Data Scientist

Using libraries

enables laziness.

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

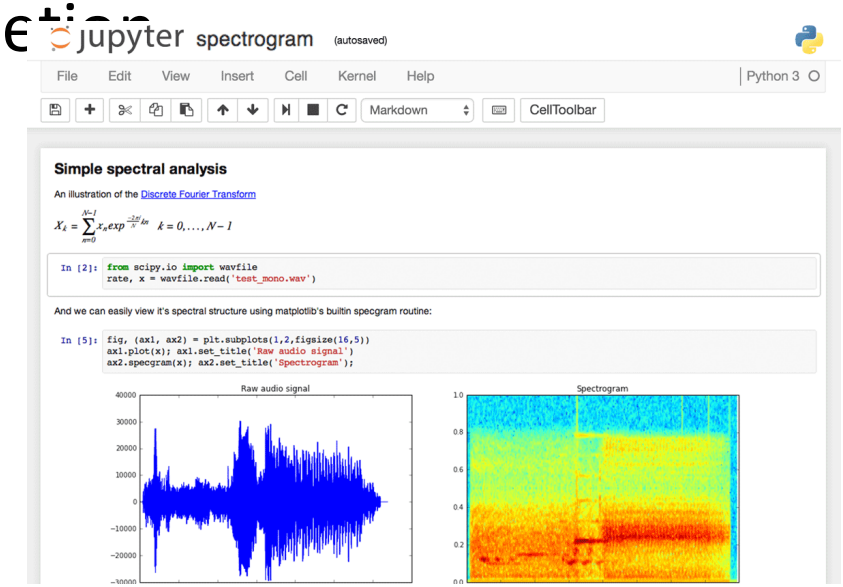
How to interact with Python

- REPL: interactive command prompt

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

- 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 Python3 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]; mylist[0]; len(mylist); mylist[12]

mylist[0] = [3, 5, 9]

- 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. On paper, write down 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

Reading:
Beginner: Data Wrangling with Python, chapter 2, pages 17 to 40
Learning Python – everything
Advanced: Python Cookbook

PDFs are posted to Blackboard

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)
```

Creating and editing .py scripts

- On Windows, Notepad is available; Mac has textEdit
- You will need to be able to view file extensions
- There are many text editors and IDEs

Pair programming exercise

- Find a partner who you have not yet collaborated with

Pair programming exercise

- Find a partner who you have not yet collaborated with

- Determine who is the less experienced programmer

- The less experienced person creates a .py file that prints "hello"
- Or, if both of you are not challenged by this,

create a .py file that prints all palindromic numbers with 7 digits (see this page)

- Run the script using the Python interpreter

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

Exercise: use terms + definitions as flashcards

Source:

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

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

Activity: form pairs; split the cards in half;
terms + definitions matching; partner checks once done
Answers will be verified as a class

Quiz on Python data structures

Given a list,

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

What would the command

```
len(my_list)
```

return?

Do not shout out the answer. We will vote.

Quiz on Python data structures

Given a list,

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

What would the command

```
len(my_list)
```

return?

3: vote with blue card

4: vote with yellow card

NOT SURE: orange card

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?

~~3: vote with blue card~~

4: vote with yellow card

~~NOT SURE: orange card~~

`Len` = length of list

Quiz on Python data structures

Given a list,

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

What would the command

```
my_list[1]
```

return?

a: vote with blue card

b: vote with yellow card

NOT SURE: orange card

Quiz on Python data structures

Given a list,

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

0 1 2 3

What would the command

```
my_list[1]
```

return?

~~a: vote with blue card~~

b: vote with yellow card

~~NOT SURE: orange card~~

Python indexes from 0

Essentials of Python for Exploratory Data Analysis (EDA)

Covered so far:

- Loading files, ie CSV, JSON, XML (see week 1)
- Use of the REPL
- Data structures like lists, dictionaries

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

Jupyter for more than just learning and small scale analysis

- Analyze video to detect 60 million faces

http://willcrichton.net/notes/rapid-prototyping-data-science-jupyter/

- 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

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

Data source:

https://www.bowl.com/Open_Championships/Open_Championships_Home/Past_Results_and_History/

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

Notebook:

https://github.com/umbcdata601/fall2018/tree/master/jupyter_notebooks/week2_python

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 its own cell.

CUSTOMERS

businessName	address	phone	order1	order2
Bob's Diner	14 Rialto St. Boston, MA 02119	617-447-0106 617-499-0976	4 doz. handbraided Guatemalan placemats	8 basic curtains (floral) and curtain rods
Turpelo Cleaners	205 South St. 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(' ')
```

In Python, **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

Timing execution

Manual

```
import timeit
start_time = timeit.default_timer()
# code you want to evaluate
elapsed = timeit.default_timer() - start_time
```

Cell magics

- %time
- %timeit

Jupyter Extension

- https://github.com/ipython-contrib/jupyter_contrib_nbextensions/tree/master/src/jupyter_contrib_nbextensions/nbextensions/execute_time

https://github.com/umbcdata601/fall2018/blob/master/jupyter_notebooks/week2_python/timing%20code%20execution.ipynb

Pro-tip: Offline Program Design

Getty images

- I use bad meetings as the time to write out programs on paper
- Assume clean data
- Know the expected input, desired output, and relevant transforms

Map and apply

- Map: execute a function on each element of a list or series
- Apply: execute a function on each element of an axis in Pandas
- http://www.bogotobogo.com/python/python_fncs_map_filter_reduce.php
- http://manishamde.github.io/blog/2013/03/07/pandas-and-python-top-10/

Advanced: Concurrency and parallelism

- http://dask.pydata.org/en/latest/
- https://docs.python.org/3/library/multiprocessing.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


AND NOW FOR
SOMETHING
COMPLETELY
DIFFERENT

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**

Ben's 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?

Scoring for the gradient of homework

1. Warm up: graded using 0 to 5 scale; maximum 6
2. Assignment: graded using 0 to 5 scale; maximum 6
3. Challenge: 0 or 1

Rubric for Jupyter notebook warmup and assignment

0. Nothing turned in
0. Not using Python 3
1. Code turned in but one or more cells do not compile or execution takes more than 5 minutes
2. Cells compile but order of execution for cells is not sequential
3. Cells compile and are sequential, but function does not return correct values when given valid input
4. Code compiles and returns correct values when given valid input; does not handle invalid input
5. Code compiles and returns correct values when given valid input; returns an indicator of problem when invalid input is provided

- +1 for use of descriptive variable names (ie expected content, type)
- +1 for use of comments
- +1 for no dead code or unused variables

Maximum score of 6. Any score above 6 is a 6. For example, 2+1=3; 3+2=5; 4+3=6; 5+2=6

Rubric for Challenge

0. Nothing turned in
1. Some code provided; written documentation of what attempt was made. Indicate what gave you difficulty. Include citations if applicable. Maximum of 1 page of text. Alternatively, working software.

For challenges, I'm not expecting everyone to solve the problem.

Homework for Week 2

- *Warm up*: Write a function that returns the count of letters and words in a string provided as input.

- *Assignment*: Write a function that takes a list as input and produces a list with each element shifted left by one index.

  For example, [3, 7, 4, 1] becomes [1, 3, 7, 4]

https://www.hackerrank.com/challenges/

- *Challenge*: "Write a function that prints the numbers from 1 to N, where N is an input. For multiples of three print “Fizz” instead of the number and for the multiples of five print “Buzz”. For numbers which are multiples of both three and five print “FizzBuzz”."

http://wiki.c2.com/?FizzBuzzTest
https://en.wikipedia.org/wiki/Fizz_buzz

Next week: code review

Reading Assignment

Beginner: Data Wrangling with Python, chapter 2, pages 17 to 40
Learning Python

Write a half page summary of the text

Turn via Blackboard

End of class

Questions?

Comments?

Bonus material

Python advanced: yield and generators

- https://jeffknupp.com/blog/2013/04/07/improve-your-python-yield-and-generators-explained/

Multiple (sequential) JSON blobs in a file

Use of yield

- https://stackoverflow.com/questions/20400818/python-trying-to-deserialize-multiple-json-objects-in-a-file-with-each-object-s