



PYTHON II: INTRODUCTION TO DATA ANALYSIS WITH PYTHON

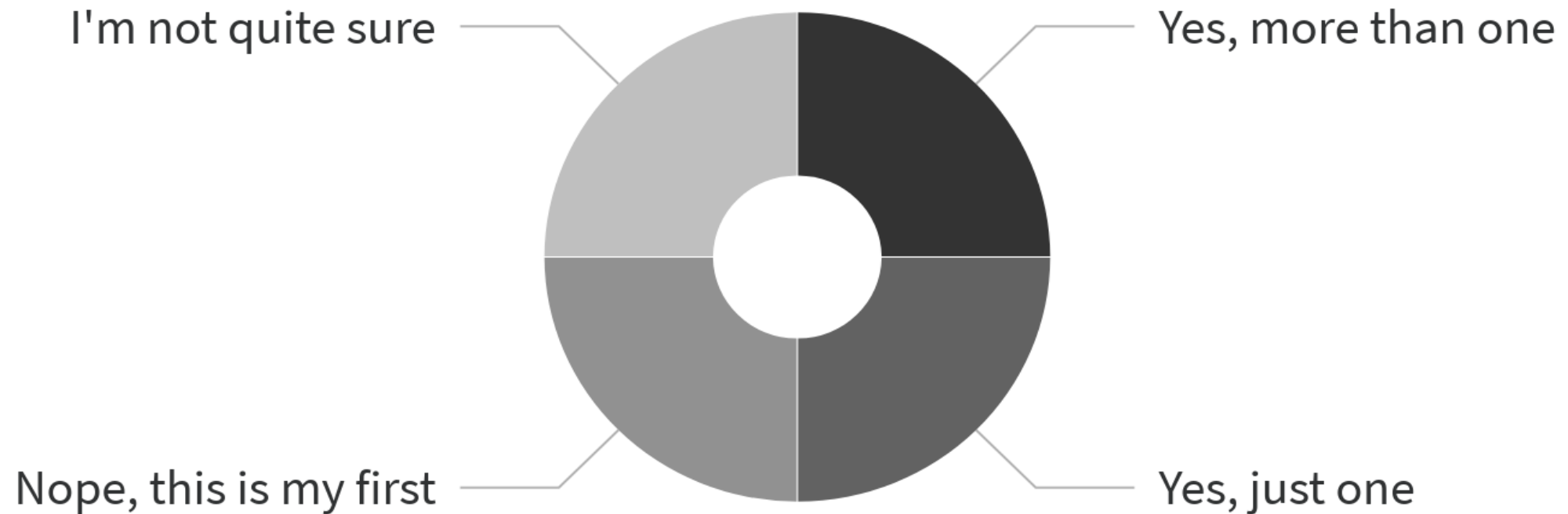
Dartmouth College | Research Computing

OVERVIEW

- What is Python?
- Why Python for data analysis?
- Development Environments
- Hands-on: Basic Data Structures in Python, Looping
- Defining a function in Python
- Importing a dataset in to a Python data structure, using modules
- Python scripts and parameters
- Questions, Resources & Links

Have you attended a Research Computing / ITC training event or workshop in the past?

Yes, more than one **A** Yes, just one **B** Nope, this is my first **C** I'm not quite sure **D**



RC.DARTMOUTH.EDU

Software

Hardware

Consulting

Training



[Request an Account](#) [Contact Us](#) [Request Help](#)

[High Performance Computing](#) [Services](#) [Help](#) [Training](#) [Partnership](#) [News](#) [About Us](#)

Our Services



Research Computing offers a wide range of services from research software support to code development and grant support. Most of our services are available at no-cost to members of the Dartmouth research community including faculty, post-docs, graduate, and undergraduate students.

Recent Posts

- [Analysing 1.4 billion rows with python](#)
- [Tech Discussion: 3D Roman Basilica](#)
- [Python IDEs and Code Editors \(Guide\)](#)
- [Slides from our recent workshop, Using R for Basic Spatial Analysis](#)
- [Slides from Programming with R, Part 1](#)

Keep in Touch

- [E-mail us](#)
- [Contact us](#)
- [Mailing list sign-up](#)

Our Services Include:

Software Support

We provide technical support and installation services for research software applications. Note: We offer support for *research* software, for help with non-research applications such as Microsoft Office please contact Dartmouth Computing.

[View a list of the Software Applications we support](#)

Data Support

We provide data management services including data storage, sharing, and access via web interfaces and support for off-site collaborators.

Code Development

We develop, debug, and optimize code to effectively use resources and improve code performance.

Parallel Programming

WHAT IS PYTHON?

- Python is an open-source programming language
- It is relatively easy to learn
- It is a powerful tool with many modules (libraries) that can be imported in to extend its functionality
- Python can be used to automate tasks and process large amounts of data
- Python can be used on Mac's, PC's, Linux, as well as in a high-performance computing environment (Polaris, Andes, Discovery machines here at Dartmouth)

WHY PYTHON FOR DATA ANALYSIS?

- Python can be used to import datasets quickly
- Python's importable libraries make it an attractive language for data analysis
 - NumPy
 - SciPy
 - Statsmodels
 - Pandas
 - Matplotlib
 - Natural Language Toolkit (NLTK)
- Python can import and export common data formats such as CSV files

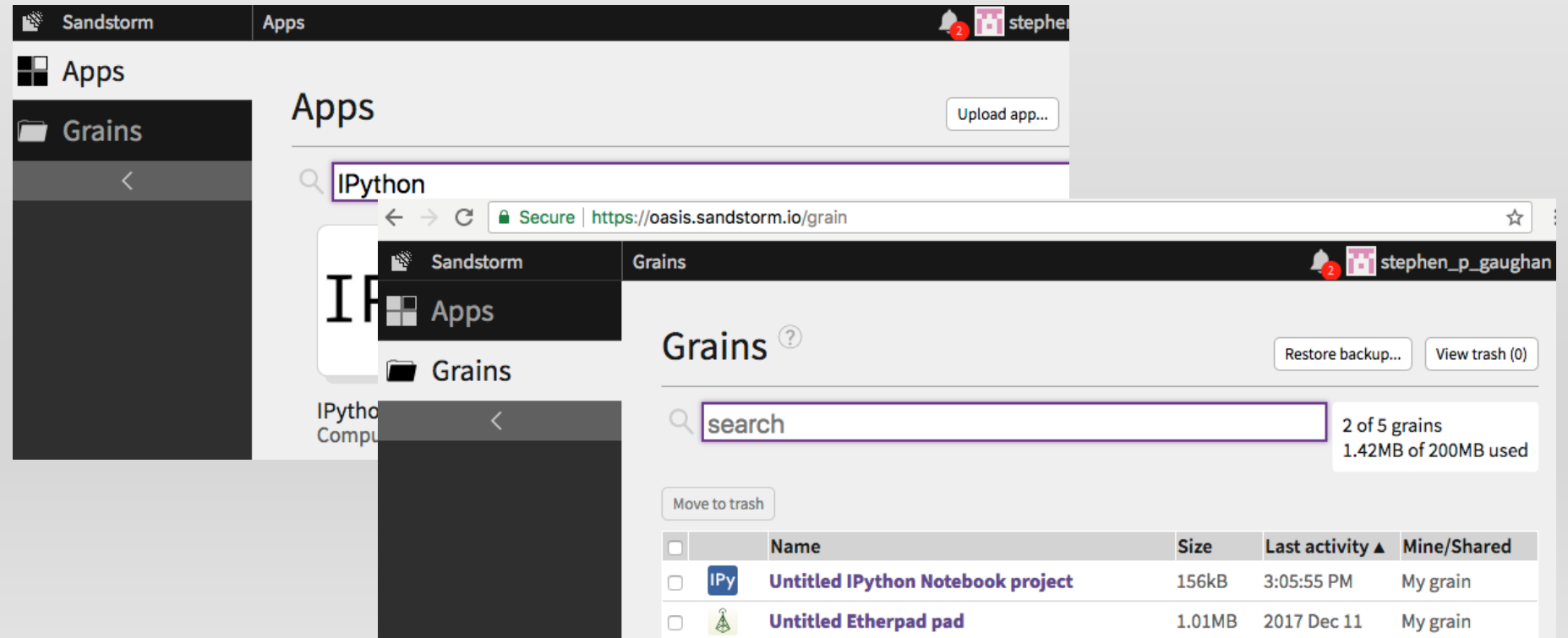
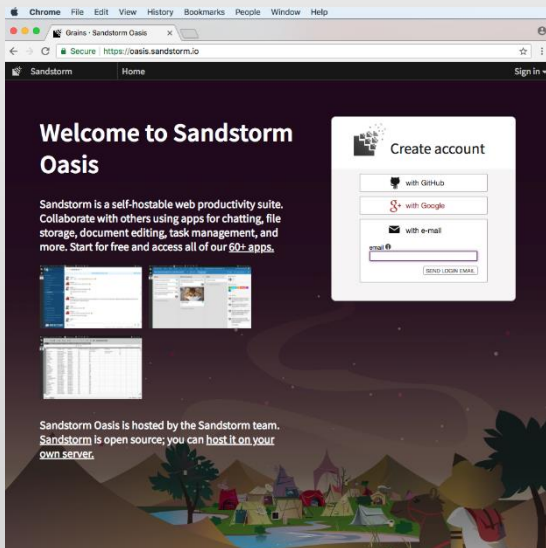
Reference: *Python for Data Analytics*, Wes McKinney, 2012, O'Reilly Publishing

DEVELOPMENT ENVIRONMENTS(I)

- Python can be run in a variety of environments with various tools
 - From the command line (most Mac's have Python installed by default)
 - From a windows terminal
 - From a Linux terminal
 - Using an Integrated Development Environment such as Eclipse or PyCharm IDE
 - Using a web-hosted “sandbox” environment

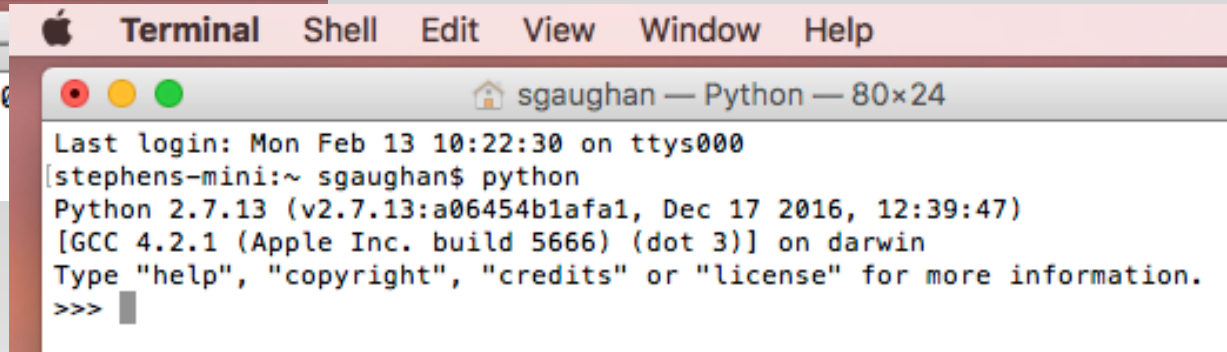
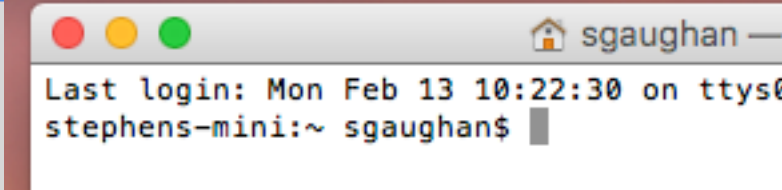
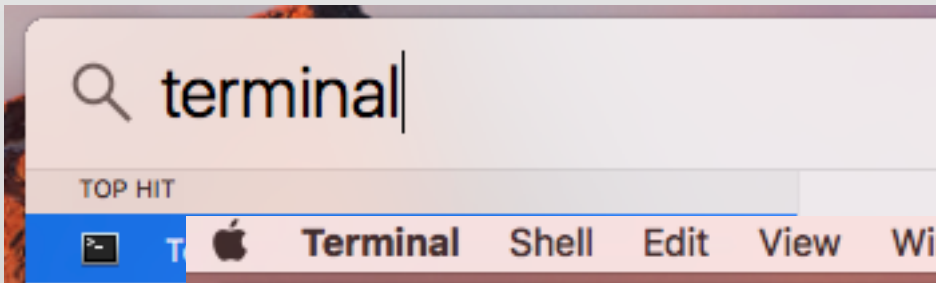
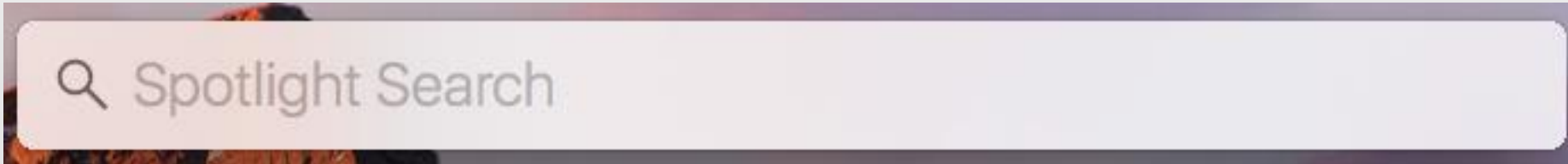
DEVELOPMENT ENVIRONMENTS (II)

- Browser-based sandbox



DEVELOPMENT ENVIRONMENTS (III)

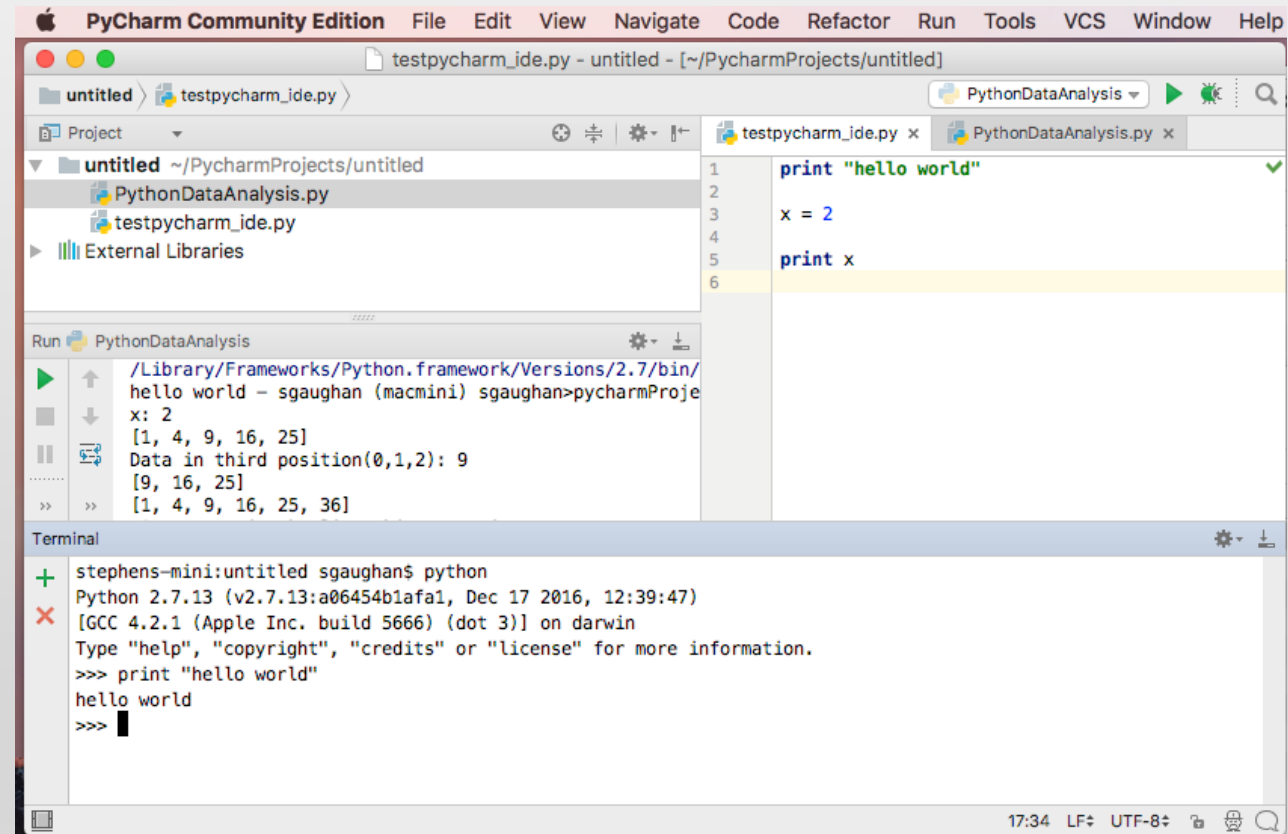
- Mac Terminal



DEVELOPMENT ENVIRONMENTS (IV)

Entering Python code:

Command line or Optional IDE



Python Integrated Development Environment

PYTHON SOFTWARE FOUNDATION AND MATERIALS FOR THIS TUTORIAL

- Materials download: www.dartgo.org/pyii
- Material reference and basis, Python Software Foundation at Python.org: <https://docs.python.org/3/tutorial/>
- Note about Python 2.x and Python 3.x:
 - There are a variety of differences between the versions.
 - Some include:
 - Print “hi world” in 2.x is now print(“hi world”) in 3.x
 - Division with integers can now yield a floating point number
 - In 2.x, $11/2=5$, whereas in 3.x, $11/2=5.5$
 - More at <https://wiki.python.org/moin/Python2orPython3>

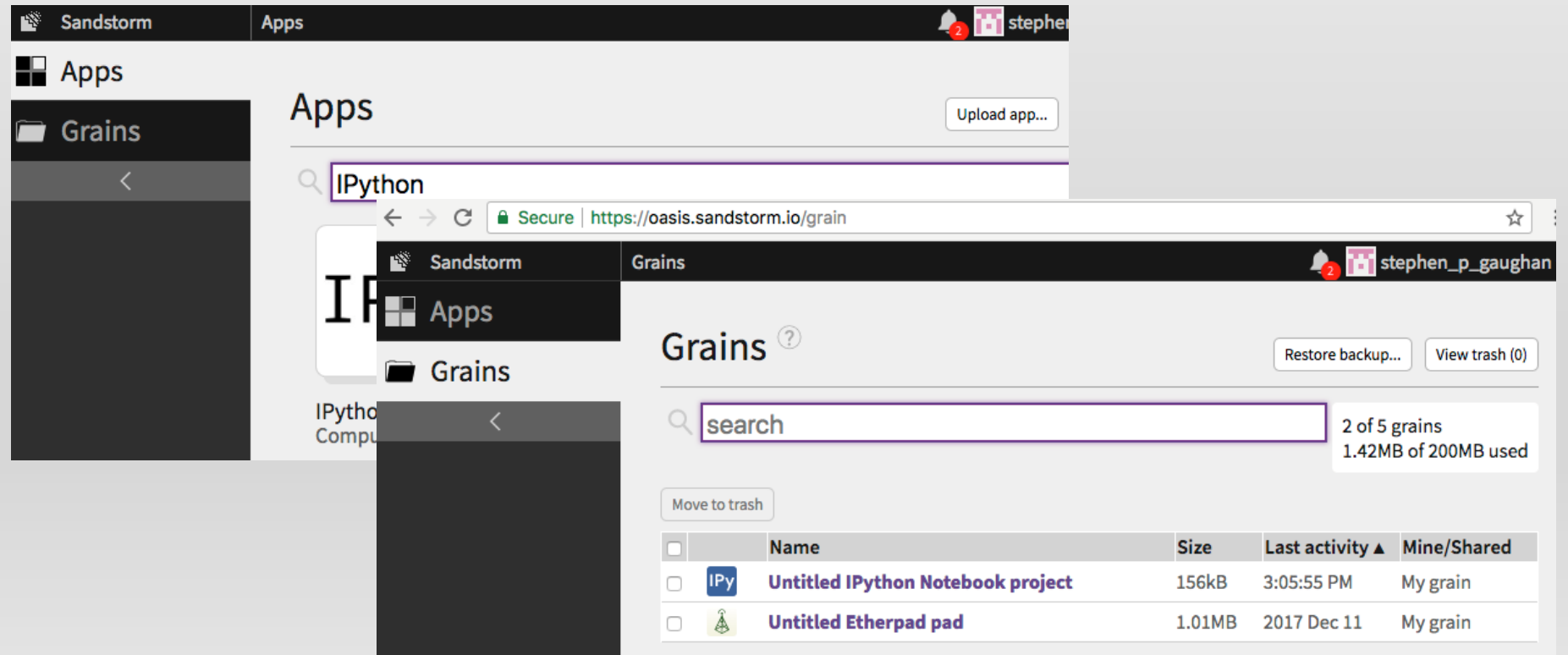
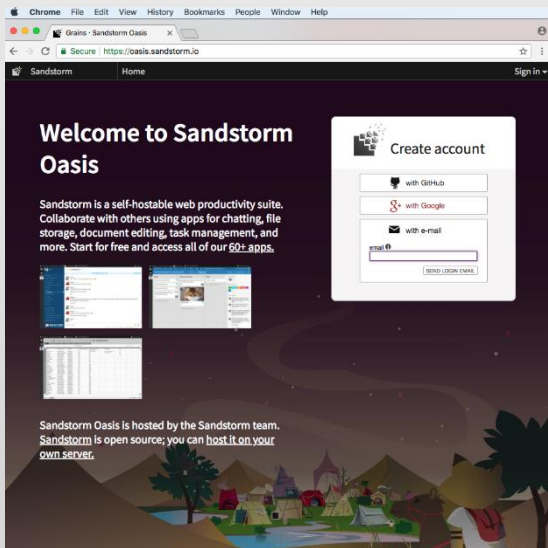
HANDS ON PRACTICE:

GETTING STARTED

- Preliminary Steps
 - Download data from Dartgo link (www.dartgo.org/pyii)
 - Get the dataset to either:
 - A familiar location on your desktop (e.g.g desktop/python-novice-inflammation/data)
 - Or uploaded in to the sandstorm sandbox web environment
- Opening Python
 - Open your browser to <https://oasis.sandstorm.io/> (Create an account or sign in with existing account)
 - Or, open a terminal on your Mac or PC

HANDS ON PRACTICE: GETTING STARTED

- Open a web browser
- Navigate to oasis.sandstorm.io

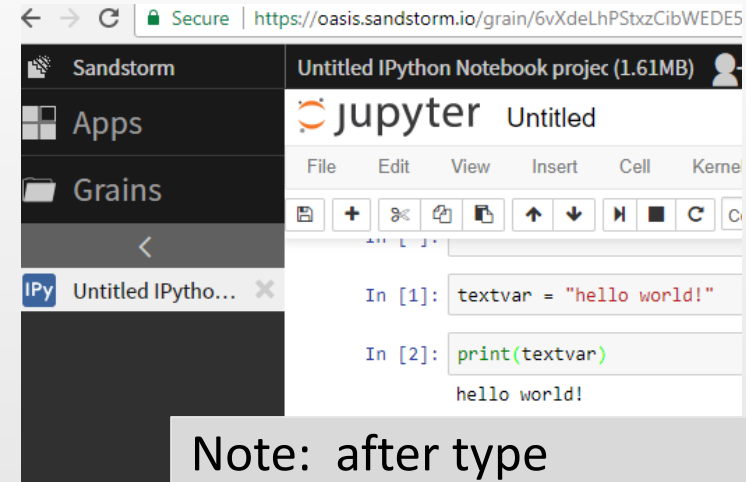


HANDS ON: DIVING IN

Using a Python interpreter or IDE:

```
In [1]: textvar = "hello world!"
```

```
In [2]: print(textvar)
```



Note: after type
A line, click Alt+Enter
To run the line and go to next line

```
# this a comment  
#Using a Python sandbox, interpreter or IDE:
```

```
textvar = 'hello world!'  
print(textvar)
```

```
# This creates our first variable. It is a string or text variable.  
#Next, we'll define a variable that contains a numerical value:
```

```
numbervar = 5  
print(numbervar)
```

Materials reference: <https://docs.python.org/3/tutorial/>

BASIC DATA STRUCTURES IN PYTHON: LISTS

Create a list

```
In [7]: listofsquares = [1,4,9,16,25]
```

```
In [12]: print(listofsquares)
```

A list in Python a basic sequence type

```
squares = [1, 4, 9, 16, 25]
```

```
print(squares[2])
```

Basic list functions: retrieve a value, append, insert

```
print(squares[1])
```

```
squares.append(35) # add a value to end of list
```

```
print(squares)
```

```
squares[5] = 36 # ... and then fix our error, 6*6=36!
```

```
print(squares)
```

BASIC DATA STRUCTURES IN PYTHON: LISTS WITH CONDITIONALS

This is where the sandbox environment, or an IDE, becomes very useful

```
# a basic conditional structure
```

```
if 0 == 0:  
    print("true")
```

```
# used with a list element  
if squares[1] == (2*2):  
    print('correct!')  
else:  
    print('wrong!')
```

```
squares[:] = [] # clear out the list
```

```
In [11]: if listofsquares[1] == (2*2):  
         print('correct!')  
         else:  
             print('wrong!')
```


LOOPING OVER A BASIC DATA STRUCTURE

```
#Loop over a data structure
```

```
berries = ['raspberry', 'blueberry', 'strawberry']
```

```
#Loop over a data structure
```

```
berries = ['raspberry', 'blueberry', 'strawberry']
```

```
for i in berries:
```

```
    print("Today's pies: " + i)
```

```
# sort the structure and then loop over it
```

```
for i in sorted(berries):
```

```
    print("Today's pies(alphabetical): " + i)
```

BASIC DATA STRUCTURES: TUPLES AND SETS

A “Tuple” is a type of *sequence* that can contain a variety of data types

```
# Create a tuple
```

```
mytuple = ('Bill', 'Jackson', 'id', 5)  
Print(mytuple)
```

```
# Use indexing to access a tuple element. Note: tuple elements  
start counting at 0, not 1
```

```
mytuple[3]
```

BASIC DATA STRUCTURES: DICTIONARIES

```
# Create a Dictionary or look-up table
# The leading elements are known as “keys” and the
  trailing elements are known as “values”
lookuptable = {'Dave': 4076, 'Jen': 4327, 'Joanne':
  4211}
lookuptable['Dave']
# show the keys
lookuptable.keys()
lookuptable.values()
# check to see if an element exists
'Jen' in lookuptable
# output: true
```

BASIC DATA STRUCTURES: DICTIONARIES

Create a Dictionary or look-up table

Use the key for error-checking to see if a value exists

leading elements are known as “keys” and the trailing # check to see if an element exists

```
if 'Jen' in lookuptable:  
    print("Jen's extension is: " + str(lookuptable['Jen']))  
else:  
    print("No telephone number listed")
```

How is the speed of the workshop?

Too Fast

Too Slow

About
the right
pace

DATA STRUCTURES: LOOPING

```
# Loop over a dictionary data structure  
# print the whole dictionary  
for i,j in lookuptable.iteritems():  
    print i,j
```

WHILE LOOPS AND LOOP COUNTERS

- Use a “while” loop to generate a Fibonacci series

```
a, b = 0, 1
i = 0
fibonacci = '1'
while i < 7:
    print(b)
    fibonacci = fibonacci + ', ' + str(b)
    a=b
    b=a+b
    i=i+1 # increment the loop counter
print(fibonacci)
```

IMPORTING AND USING MODULES

Modules greatly extend the power and functionality of Python, much like libraries in R, JavaScript and other languages

```
import sys
```

```
# check the version of Python that is installed
```

```
sys.version
```

```
'3.4.2 (default, Oct 8 2014, 10:45:20) \n[GCC 4.9.1]' in this  
sandbox!
```

```
# check the working directory
```

```
import os
```

```
os.getcwd()
```

```
'/var/home' - this is less applicable in the sandbox - on  
laptop or a linux server it is essential to know the working  
directory
```


IMPORTING AND USING MODULES

```
# multiply some consecutive numbers  
1*2*3*4*5*6*7  
5040
```

```
# save time and labor by using modules effectively  
import math  
math.factorial(7)
```

MODULES

```
# Modules  
from math import pi  
print(pi)  
round(pi)  
round(pi, 5)
```

DEFINING A FUNCTION IN PYTHON

Functions save time by storing repeatable processes

Defining a function is easy:

use the 'def' function in Python

```
def xsquared( x ):
    # find the square of x
    x2 = x * x;
    # the 'return' statement returns the function
value
    return x2
```

```
# call the function
```

```
y = xsquared(5)
print str(y)
```

```
# Output: 25
```

WITH AND FOR COMMANDS

We'll use the `WITH` and `FOR` commands to help us read in and loop over the rows in a CSV file; here's some pseudo-code of what we'd like to do:

`WITH open (file.extension) as fileobject:`

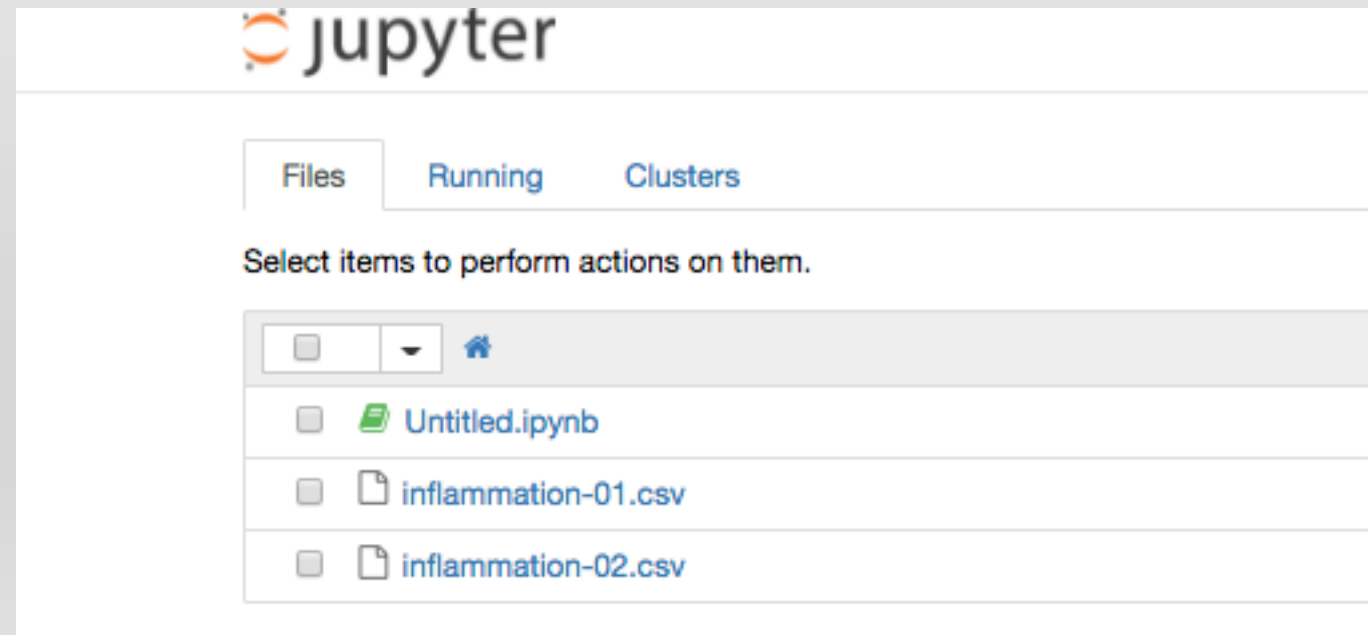
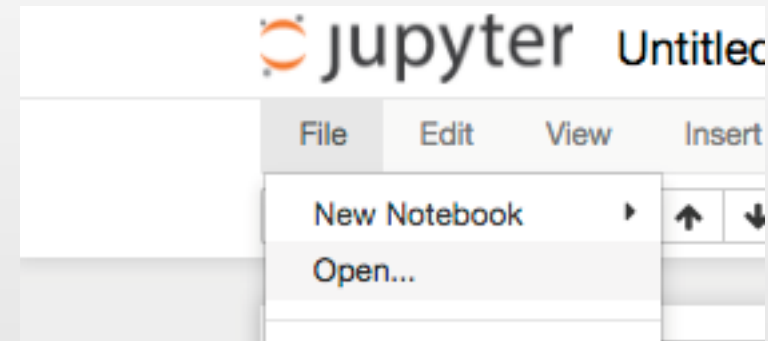
 {get data in file}

`FOR rows in file:`

 {do something with data elements in the rows}

UPLOAD DATA

- To upload data in to the hosted python instance, click the “jupyter” title to go back to upload screen
- Use the “Files” tab to upload
- Upload > Browse
- The hosted environment supports the upload of reasonably-sized csv files



DATA ANALYSIS – INFLAMMATION DATASET

- Next, let's examine a dataset of patients (rows) and forty days of inflammation values (columns)

```
import os
os.listdir()

f = open('inflammation-01.csv')
filecontent = f.read()

print(filecontent)
```

```
# load with numpy
import numpy

numpy.loadtxt(fname='inflammation-01.csv',
delimiter=',') # load csv

# load in to a variable

data = numpy.loadtxt(fname='inflammation-
01.csv', delimiter=',') # load csv to variable

print(data)
print(type(data))
print(data.dtype)
print(data.shape)
```

DATA ANALYSIS – INFLAMMATION DATASET

- View data elements with matrix addressing

```
print('first value in data:', data [0,0])
```

```
print(data[30,20])
```

```
maxval = numpy.max(data)
```

```
print('maximum inflammation: ', maxval)
```

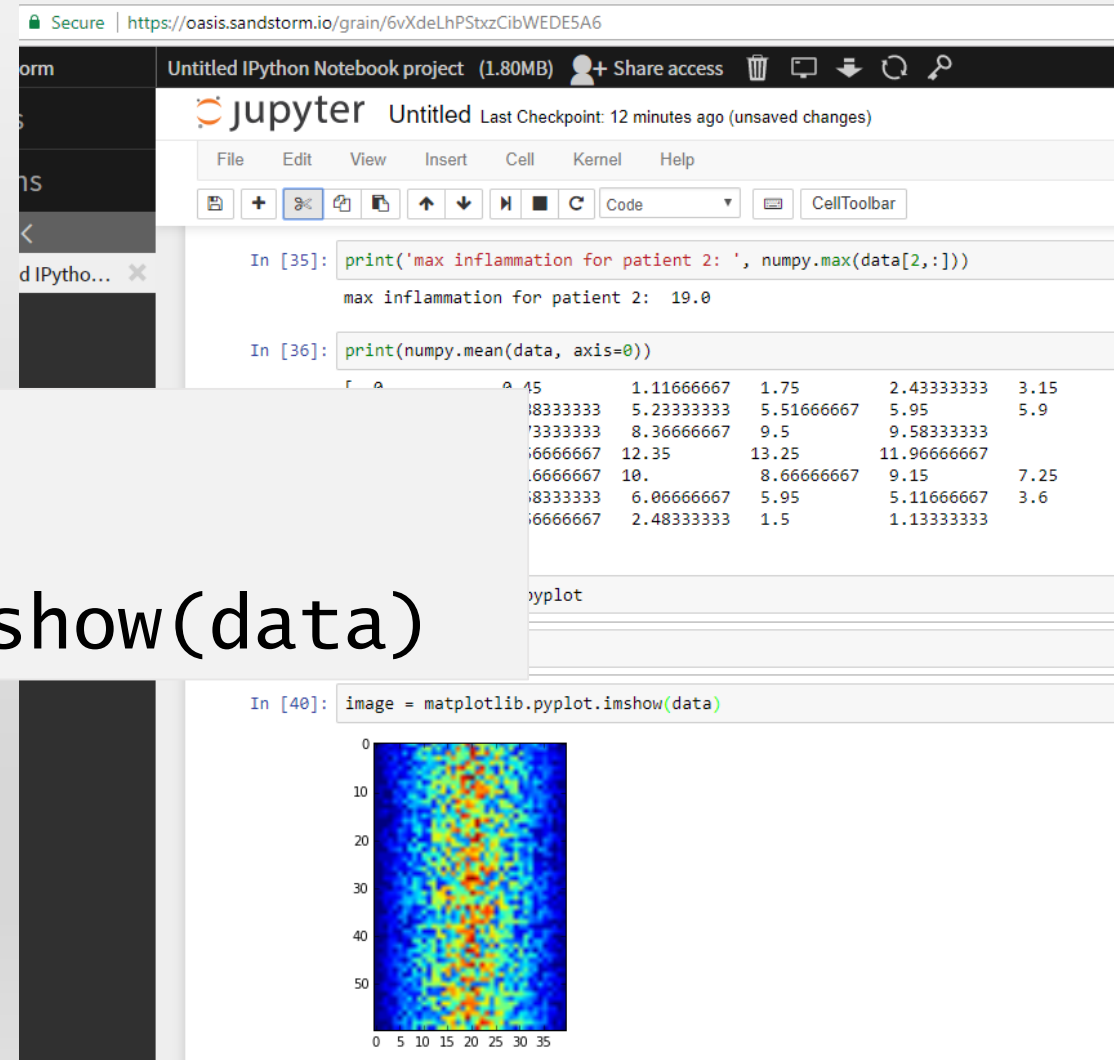
```
stdval = numpy.std(data)
```

```
print('standard deviation: ', stdval)
```

DATA ANALYSIS – INFLAMMATION DATASET

- Next, let's examine a dataset of patients (rows) and forty days of inflammation values

```
import matplotlib.pyplot  
%matplotlib inline  
image = matplotlib.pyplot.imshow(data)
```



The screenshot shows a Jupyter Notebook interface with the following content:

```
In [35]: print('max inflammation for patient 2: ', numpy.max(data[2,:]))  
max inflammation for patient 2: 19.0
```

```
In [36]: print(numpy.mean(data, axis=0))  
0.45      1.11666667  1.75      2.43333333  3.15  
8.3333333  5.23333333  5.51666667  5.95      5.9  
7.3333333  8.36666667  9.5       9.58333333  
6.6666667  12.35      13.25     11.96666667  
6.6666667  10.        8.66666667  9.15      7.25  
8.3333333  6.06666667  5.95      5.11666667  3.6  
6.6666667  2.48333333  1.5       1.13333333
```

```
In [40]: image = matplotlib.pyplot.imshow(data)
```

The heatmap visualization shows a vertical grid of inflammation values for 56 patients (rows) over 35 days (columns). The color scale ranges from blue (low inflammation) to red (high inflammation). The y-axis is labeled from 0 to 50, and the x-axis is labeled from 0 to 35.

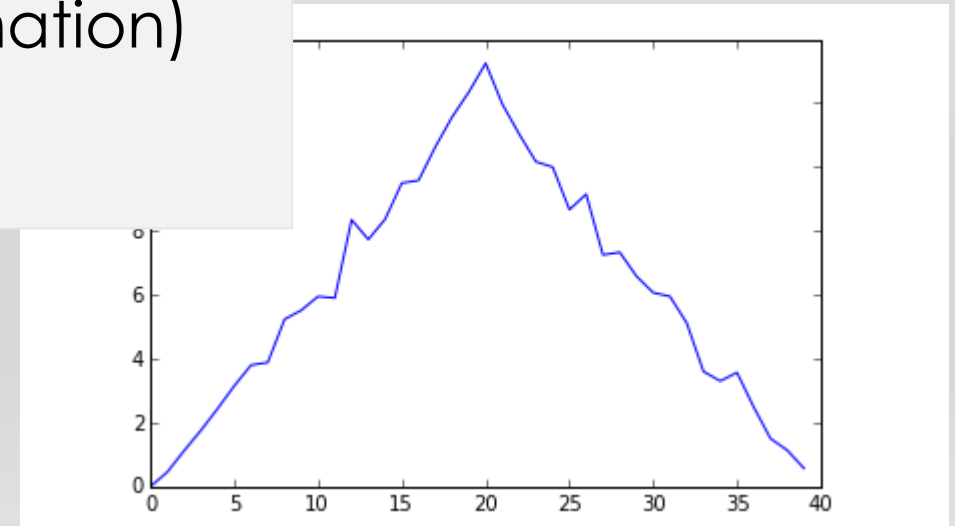
DATA ANALYSIS – INFLAMMATION DATASET

- Next, let's examine a dataset of patients (rows) and forty days of inflammation values

```
ave_inflammation = numpy.mean(data, axis=0)
```

```
ave_plot = matplotlib.pyplot.plot(ave_inflammation)
```

```
matplotlib.pyplot.show()
```



SCRIPTS AND PARAMETERS

- Use an IDE or friendly text-editor

```
#!/usr/bin/python
#-----
# my first script!

import sys
print('My first script!')
print('Number of arguments:', len(sys.argv), 'arguments.')
print('Argument List:', str(sys.argv))
#-----
```

READING MULTIPLE FILES

- Programming for speed, reusability
- Data analysis over many files

```
strfiles = ['inflammation-01.csv', 'inflammation-02.csv']  
for f in strfiles:  
    print(f)  
    #data = numpy.loadtxt(fname=f, delimiter=',')  
    #print('mean ', f, numpy.mean(data, axis=0))
```

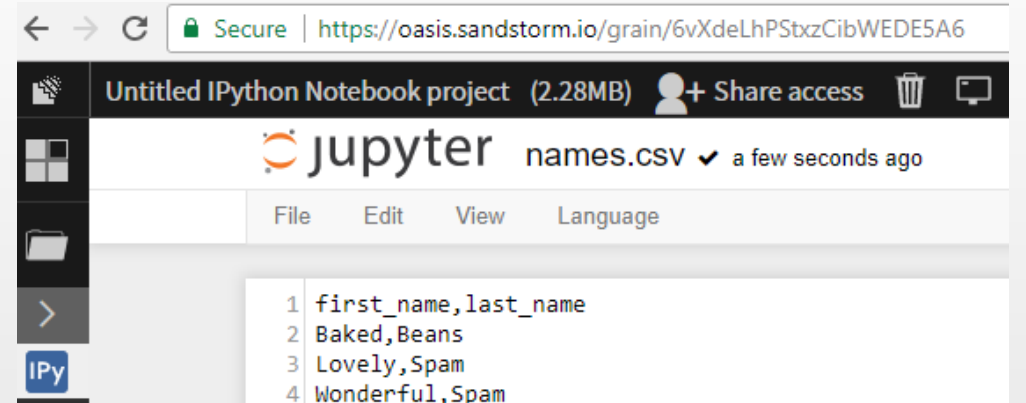
Got lots of files?
This is where RC systems like Polaris or
Discovery can be very useful



WRITE TO CSV!

```
import csv

with open('names.csv', 'w', newline='') as csvfile:
    fieldnames = ['first_name', 'last_name']
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
    writer.writeheader()
    writer.writerow({'first_name': 'Baked', 'last_name': 'Beans'})
    writer.writerow({'first_name': 'Lovely', 'last_name': 'Spam'})
    writer.writerow({'first_name': 'Wonderful', 'last_name': 'Spam'})
```



CSV HEADER ROW AND FIRST DATA ROW

- Read first rows:

```
with open('inflammation-01.csv') as f:  
    reader2=csv.reader(f)  
    row1 = next(reader2) # gets the first line  
    row2 = next(reader2)  
    print ("CSV column headers:" + str(row1))  
    print ("CSV first line: " + str(row2))
```

SCRIPTS AND PARAMETERS

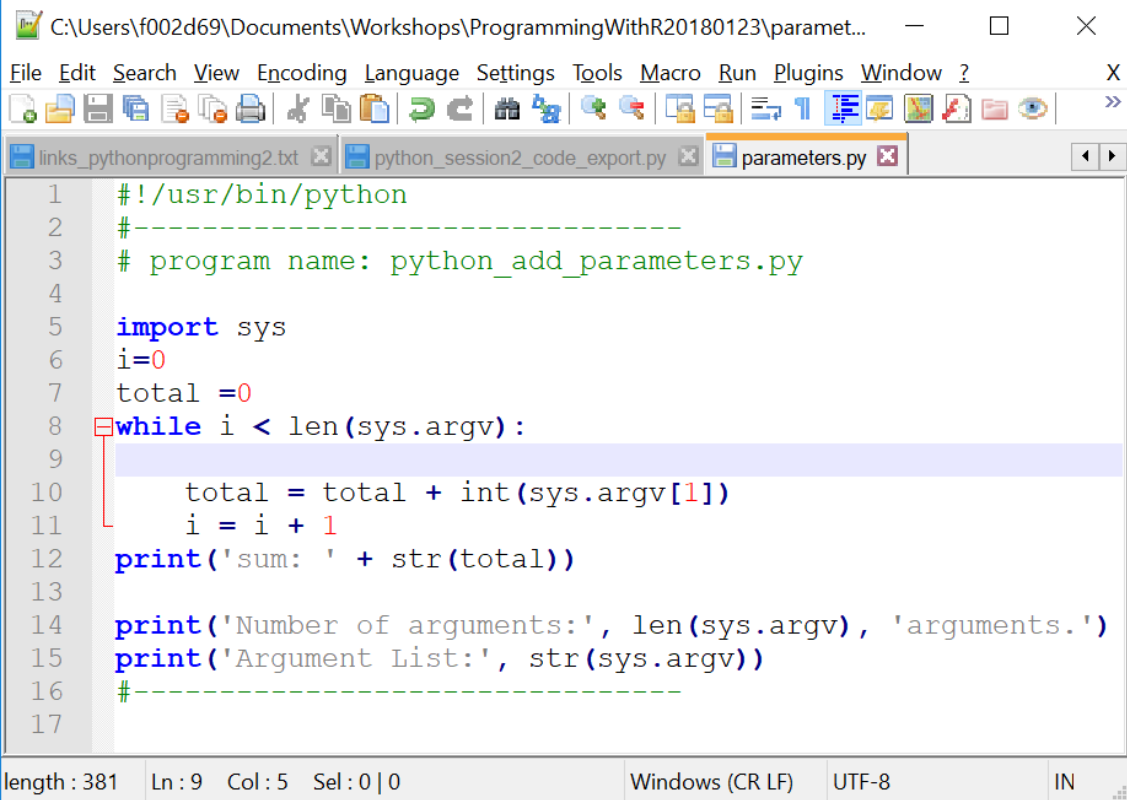
- Use an IDE or friendly text-editor

```
#!/usr/bin/python
#-----
# program name: python_add_parameters.py

import sys
i=0
total =0
while i < len(sys.argv):

    total = total + int(sys.argv[1])
    i = i + 1
print('sum: ' + str(total))

print('Number of arguments:',
len(sys.argv), 'arguments.')
print('Argument List:', str(sys.argv))
#-----
```



The screenshot shows a code editor window with the following content:

```
C:\Users\f002d69\Documents\Workshops\ProgrammingWithR20180123\paramet...
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
iinks_pythonprogramming2.txt python_session2_code_export.py parameters.py
1  #!/usr/bin/python
2  #-----
3  # program name: python_add_parameters.py
4
5  import sys
6  i=0
7  total =0
8  while i < len(sys.argv):
9
10     total = total + int(sys.argv[1])
11     i = i + 1
12     print('sum: ' + str(total))
13
14     print('Number of arguments:', len(sys.argv), 'arguments.')
15     print('Argument List:', str(sys.argv))
16     #-----
17
length : 381  Ln : 9  Col : 5  Sel : 0 | 0  Windows (CR LF)  UTF-8  IN
```

CSV LIBRARY

- Csv library built-in to Python

```
import csv
```

```
with open('inflammation-01.csv') as f:
```

```
    reader2=csv.reader(f)
```

```
    row1=next(reader2)
```

```
    print(str(row1))
```

- Output: ['0', '0', '1', '3', '1', '2', '4', '7', '8', '3', '3', '3'....

IMPORTING A DATASET INTO PYTHON: USING THE OS AND CSV MODULES

Find out where you are in the directory structure, import the operating system library (OS)

Reference: <https://docs.python.org/2/library/csv.html> section 13.1

```
import os
cwd = os.getcwd()
print "Working Directory is: " + cwd
os.chdir('c:\\temp')
os.getcwd()
```

Import the CSV file into a reader function

Download the CSV and copy it to the working directory

Note: the CSV module's reader and writer objects read and write *sequences*

```
with open('HawaiiEmergencyShelters.csv') as csvfile:
```

```
    reader = csv.DictReader(csvfile)
```

```
    for row in reader:
```

```
        print(row['NAME'], row['ADDRESS'])
```


STATISTICS FROM CSV COLUMNS

Loop through column, find average

```
with open('HawaiiEmergencyShelters.csv') as csvfile:
    reader = csv.DictReader(csvfile)
    x_sum = 0
    x_length = 0
    for row in reader:
        try:
            x = row['NUMCOTS']
            x_sum += int(x)
            x_length += 1
        except ValueError:
            print("Error converting: {0:s}".format(x))
    x_average = x_sum / x_length
    print ('Average: ')
    print(x_average)
```

| | A | B | C | D |
|----|----------|------------------------------|---------------------------|---------|
| 1 | OBJECTID | NAME | ADDRESS | NUMCOTS |
| 2 | 1 | Hilo High School | 556 Waianuenue Avenue | 80 |
| 3 | 2 | Holualoa Elementary School | 76-5957 Mamalahoa High | 80 |
| 4 | 3 | Honaunau Elementary School | 83-5360 Mamalahoa High | 80 |
| 5 | 4 | Hookena Elementary School | 86-4355 Mamalahoa High | 80 |
| 6 | 5 | Kau High and Pahala Element | 96-3150 Pikake Street | 80 |
| 7 | 6 | Kaumana Elementary School | 1710 Kaumana Drive | 80 |
| 8 | 7 | Kohala Elementary School | 54-3609 Akoni Pule Highw | 80 |
| 9 | 8 | Waiakeawaena Elementary Sc | 2420 Kilauea Ave | 100 |
| 10 | 9 | Hilo Intermediate School | 587 Waianuenue Avenue | 100 |
| 11 | 10 | Keaau Middle School | 16-565 Keaau Pahoa Road | 100 |
| 12 | 11 | Pahoa High and Intermediate | 15-3038 Pahoa Village Ro | 100 |
| 13 | 12 | Waiakea Elementary School | 180 West Puainako Street | 20 |
| 14 | 13 | Kealakehe Intermediate Scho | 74-5062 Onipaa Street | 20 |
| 15 | 14 | Kealakehe Elementary School | 74-5118 Kealakaa Street | 20 |
| 16 | 15 | Waimea Elementary and Inter | 67-1225 Mamalahoa High | 20 |
| 17 | 16 | Konawaena High School | 81-1043 Konawaena Scho | 20 |
| 18 | 17 | Kohala High School | 54-3611 Akoni Pule Highw | 20 |
| 19 | 18 | Honokaa High and Intermedia | 45-527 Pakalana Street | 60 |
| 20 | 19 | Ernest Bowen de Silva Elemer | 278 Ainako Avenue | 60 |
| 21 | 20 | Kahakai Elementary School | 76-147 Royal Poinciana D | 60 |
| 22 | 21 | Mountain View Elementary Sc | 18-1235 Volcano Highway | 60 |
| 23 | 22 | Waiakea High School | 155 West Kawili Street | 60 |
| 24 | 23 | Keonepoko Elementary Scho | 15-890 Kahakai Boulevarc | 50 |
| 25 | 24 | Kealakehe High School | 74-5000 Puohuluhuli Stree | 50 |
| 26 | 25 | Keaau High School | 16-725 Keaau-Pahoa Road | 50 |
| 27 | 26 | Pahoa Elementary School | 15-3030 Kuuhome Street | 50 |
| 28 | 27 | Waimea District Court | 67-5175 Kamamalu Stree | 50 |

NUMERICAL FUNCTIONS

```
# Float and Int
```

```
x = 3.453
```

```
xint = int(x)
```

```
yfloat = float(2)
```

```
Xround = round(x)
```

INSTALLING NUMPY FOR PYTHON 2.7

“Numpy” is a helper module in Python for numerical processing

To get the NUMPY installer

Mac -

<https://sourceforge.net/projects/numpy/files/NumPy/1.8.0/numpy-1.8.0-py2.7-python.org-macosx10.6.dmg/download>

Pc - <https://sourceforge.net/projects/numpy/files/NumPy/1.8.0/>

Click on the dmg file. You may need to change Mac security preference (Sys Pref > Security >) to allow the DMG installer to run

STATISTICAL OPERATIONS

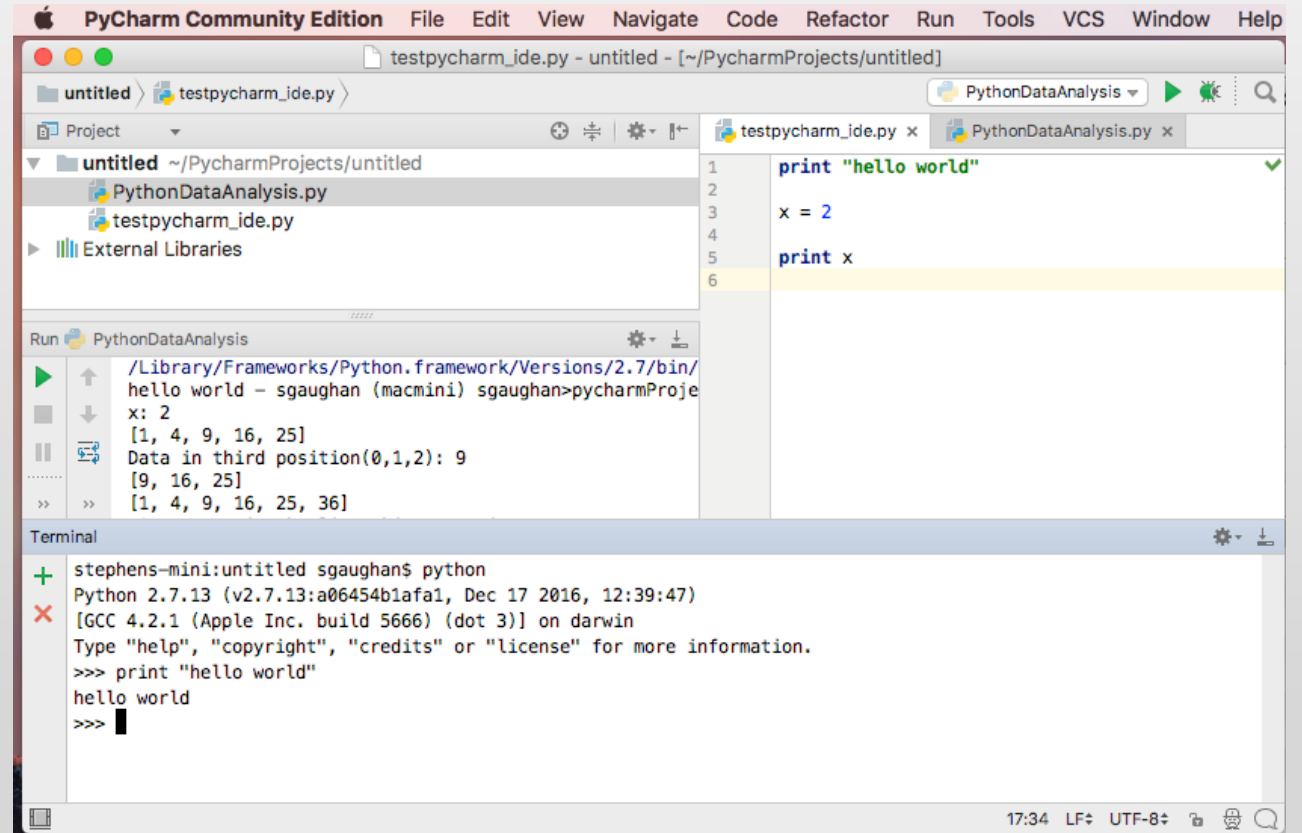
NUMPY FOR PYTHON 2.7

Reference: <https://docs.scipy.org/doc/numpy/reference/routines.statistics.html>

```
Numpy .median  
      .average  
      .std  
      .var  
      .corrcoef (Pearson product-moment correlation)  
      .correlate  
      .cov (estimate of covariance matrix)  
      .histogram  
      .amin  
      .amax  
      .percentile
```

SAVING PYTHON SCRIPTS

- Python files can be written in a simple text editor, or using an IDE editor.
- The file extension is .py



The screenshot displays the PyCharm Community Edition interface. The main editor window shows a Python script named `testpycharm_ide.py` with the following code:

```
1 print "hello world"
2
3 x = 2
4
5 print x
6
```

The Run console shows the output of the script:

```
hello world - sgaughan (macmini) sgaughan>pycharmProje
x: 2
[1, 4, 9, 16, 25]
Data in third position(0,1,2): 9
[9, 16, 25]
[1, 4, 9, 16, 25, 36]
```

The Terminal window shows the command prompt output:

```
stephens-mini:untitled sgaughan$ python
Python 2.7.13 (v2.7.13:a06454b1afa1, Dec 17 2016, 12:39:47)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print "hello world"
hello world
>>>
```

A MODULE FOR BASIC STATISTICAL ANALYSIS: USING THE NUMPY LIBRARY

```
# importing the library  
# running basic functions
```

```
>>> import numpy  
>>> numpy.mean(3, 6, 9)  
6.0  
>>> numpy.std([2, 4, 6, 8])  
2.2360679774997898
```

```
# Reference: https://docs.scipy.org/doc/numpy/reference/generated/numpy.mean.html and  
https://docs.scipy.org/doc/numpy/reference/routines.statistics.html
```

THE OS MODULE: SOME USEFUL OS COMMANDS

- More OS library tasks:
 - [os.path.realpath\(path\)](#) canonical path
 - [os.path.dirname\(path\)](#) directory
 - [os.getcwd\(\)](#) get working directory (as string)
 - [os.chdir\(path\)](#) change the working directory

PYTHON ON DARTMOUTH RESEARCH COMPUTING MACHINES

- Research Computing shared Linux resources include Polaris and Andes, as well as the high-performance computing platform Discovery.
- These machines have several versions of Python installed, and commonly-used modules. Additional modules can be installed upon request
- Polaris currently has Python 2.6.6 as the default, and Numpy and Scipy libraries are installed.
- Andes currently has Python 2.7.5 as the default, with Numpy, Scipy and the Pandas modules installed. Pandas is another commonly used data analysis library.

PYTHON SOFTWARE FOUNDATION AND MATERIALS FOR THIS TUTORIAL

- Materials download: www.dartgo.org/workshopsg and download [IntroDataAnalysisPython](#)
- Material reference and basis, Python Software Foundation at Python.org: <https://docs.python.org/2/tutorial/>

RESOURCES & LINKS

- Research Computing
 - Research.computing@dartmouth.edu
 - <http://rc.dartmouth.edu>
- Python Foundation
- Online tutorials
- Web forums
 - Stack overflow:
<http://stackoverflow.com/questions/tagged/python>

LEARNING MORE...

- Python Tutorials
 - Python 2.7.13 <https://docs.python.org/2/tutorial/>
 - Python 3.6 <https://docs.python.org/3.6/tutorial/>
- Numpy, Scipy tutorials
 - <https://docs.scipy.org/doc/numpy-dev/user/quickstart.html>
 - <http://cs231n.github.io/python-numpy-tutorial/>
- Python CSV library tutorial
 - <https://docs.python.org/2/library/csv.html>
- Lynda, Youtube Online tutorials
 - Lynda, log in with Dartmouth credentials:
www.lynda.com/portal/dartmouth
 - Search for Python Programming, Numpy, Scipy

QUESTIONS?



Workshop feedback

When survey is active, respond at PollEv.com/dartrc

0 surveys done

 0 surveys underway

Start the presentation to see live content. Still no live content? Install the app or get help at PollEv.com/app