iX

Welcome to iX

Intro to Data Analysis & Visualization with Python

Getting the lay of the land...

Week 1

Agenda

- Class 1: WHY – Learn data analysis and visualization?
- Class 2: HOW – Python for data analysis
- Class 3: HOW – Getting set up
- Class 4: HUH? – Jupyter and Python basics
- Class 5: HELP - Handy resources
- Class 6: HOMEWORK - Our first Python package: NumPy

Outcomes

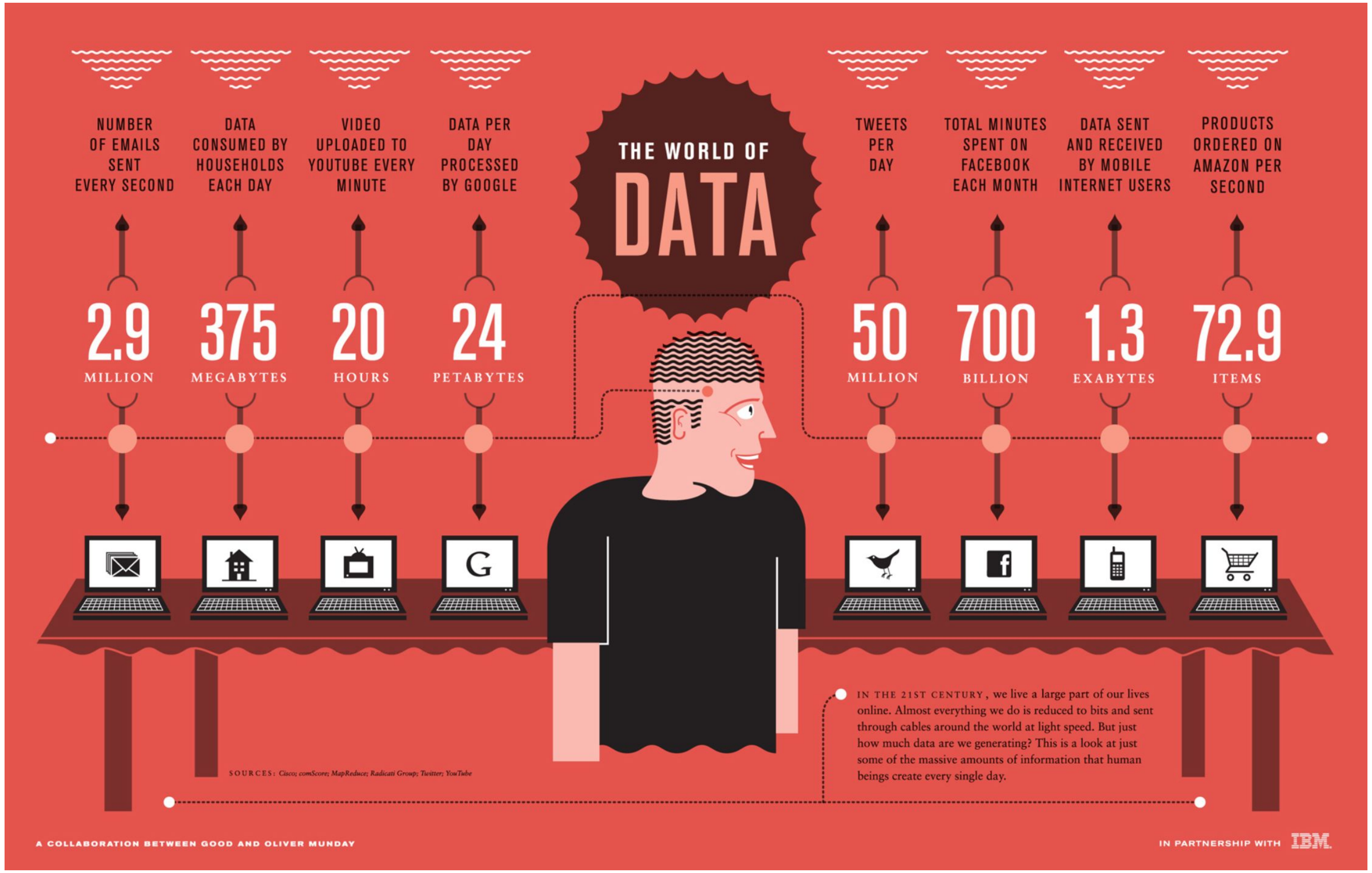
By the end of this week’s classes you should:

1. Get pumped about analyzing data in Python
2. Have Python and Jupyter notebooks installed
3. Know how to start and navigate around a Jupyter notebook
4. Become familiar with Python basics with NumPy

Why learn data analysis and visualization?

Class 1

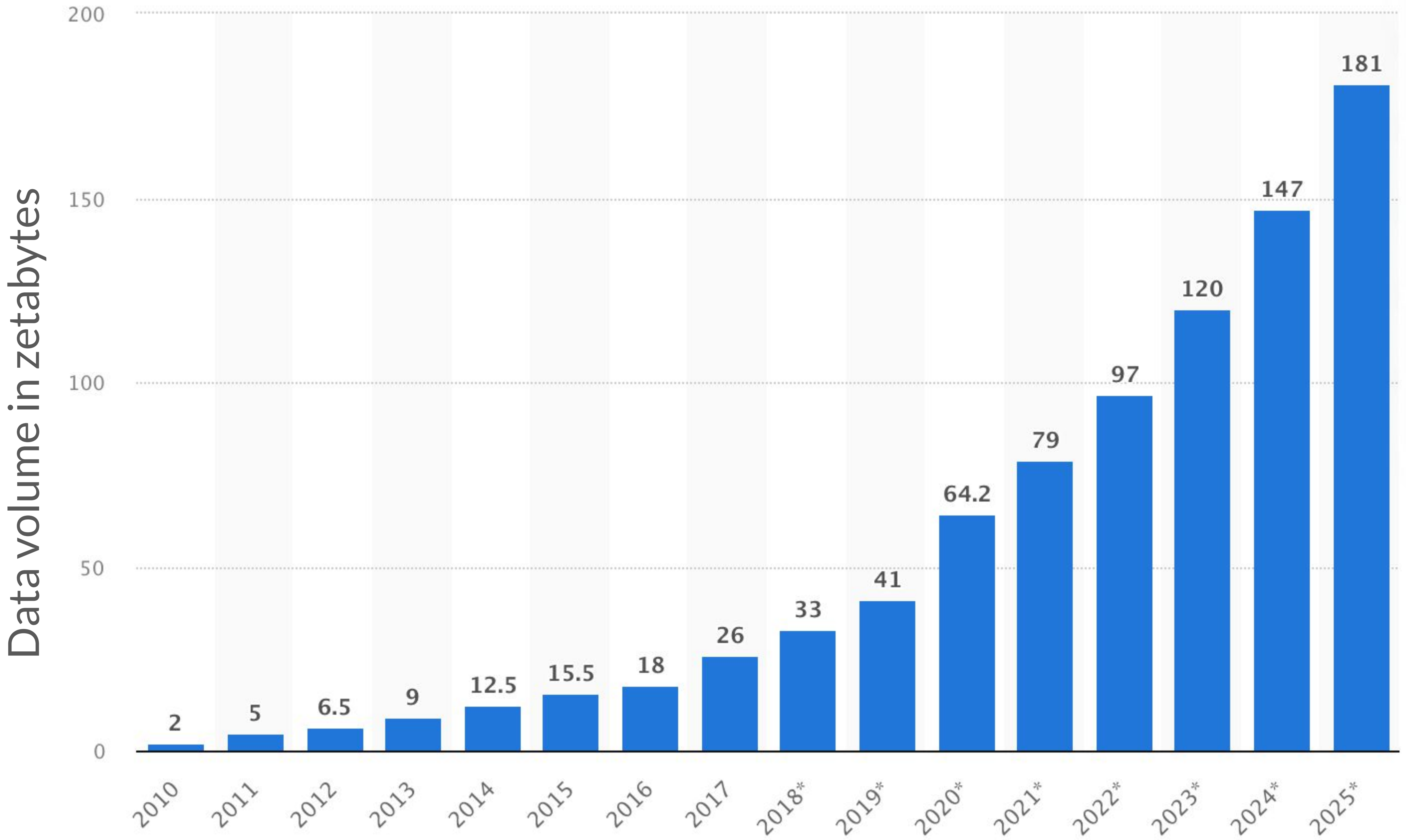
We live in a world of data


THE WORLD OF DATA
NUMBER OF EMAILS SENT EVERY SECOND
2.9 MILLION
DATA CONSUMED BY HOUSEHOLDS EACH DAY
375 MEGABYTES
VIDEO UPLOADED TO YOUTUBE EVERY MINUTE
20 HOURS
DATA PER DAY PROCESSED BY GOOGLE
24 PETABYTES
TWEETS PER DAY
50 MILLION
TOTAL MINUTES SPENT ON FACEBOOK EACH MONTH
700 BILLION
DATA SENT AND RECEIVED BY MOBILE INTERNET USERS
1.3 EXABYTES
PRODUCTS ORDERED ON AMAZON PER SECOND
72.9 ITEMS
G
A COLLABORATION BETWEEN GOOD AND OLIVER MUNDAY
IN PARTNERSHIP WITH IBM

IN THE 21ST CENTURY, we live a large part of our lives online. Almost everything we do is reduced to bits and sent through cables around the world at light speed. But just how much data are we generating? This is a look at just some of the massive amounts of information that human beings create every single day.

SOURCES: *Cisco; comScore; MapReduce; Radicati Group; Twitter; YouTube*

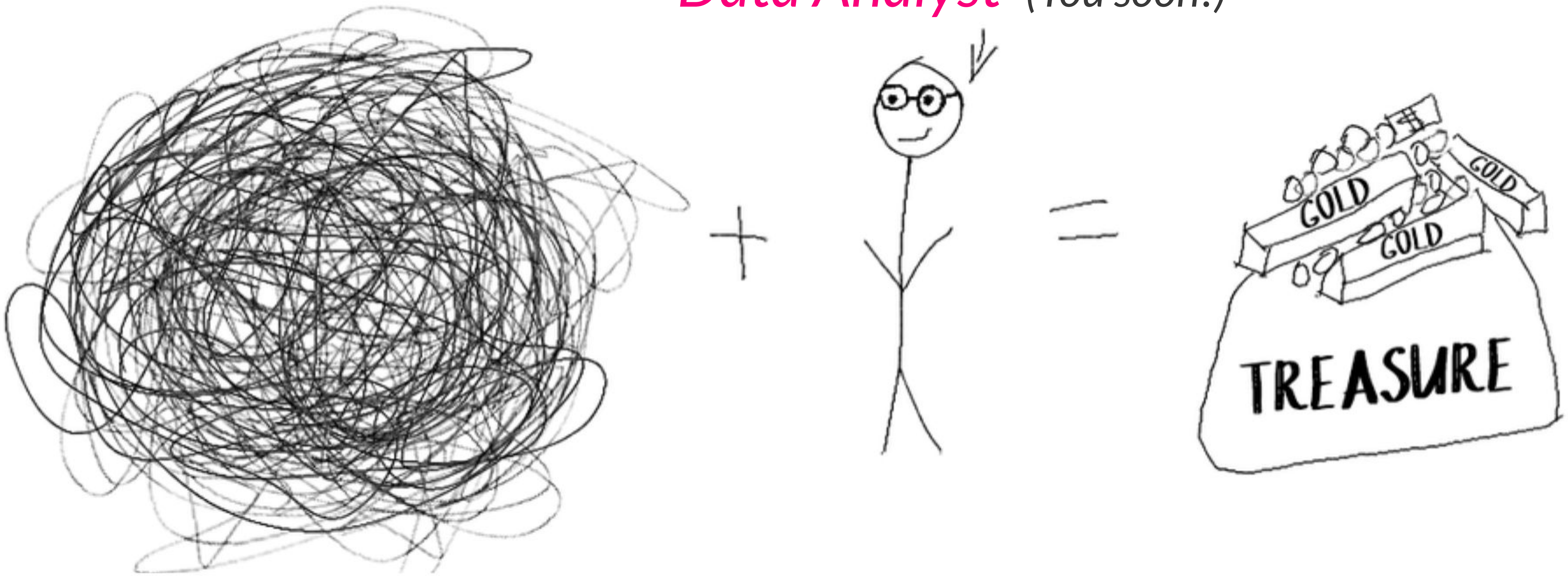
How much data?



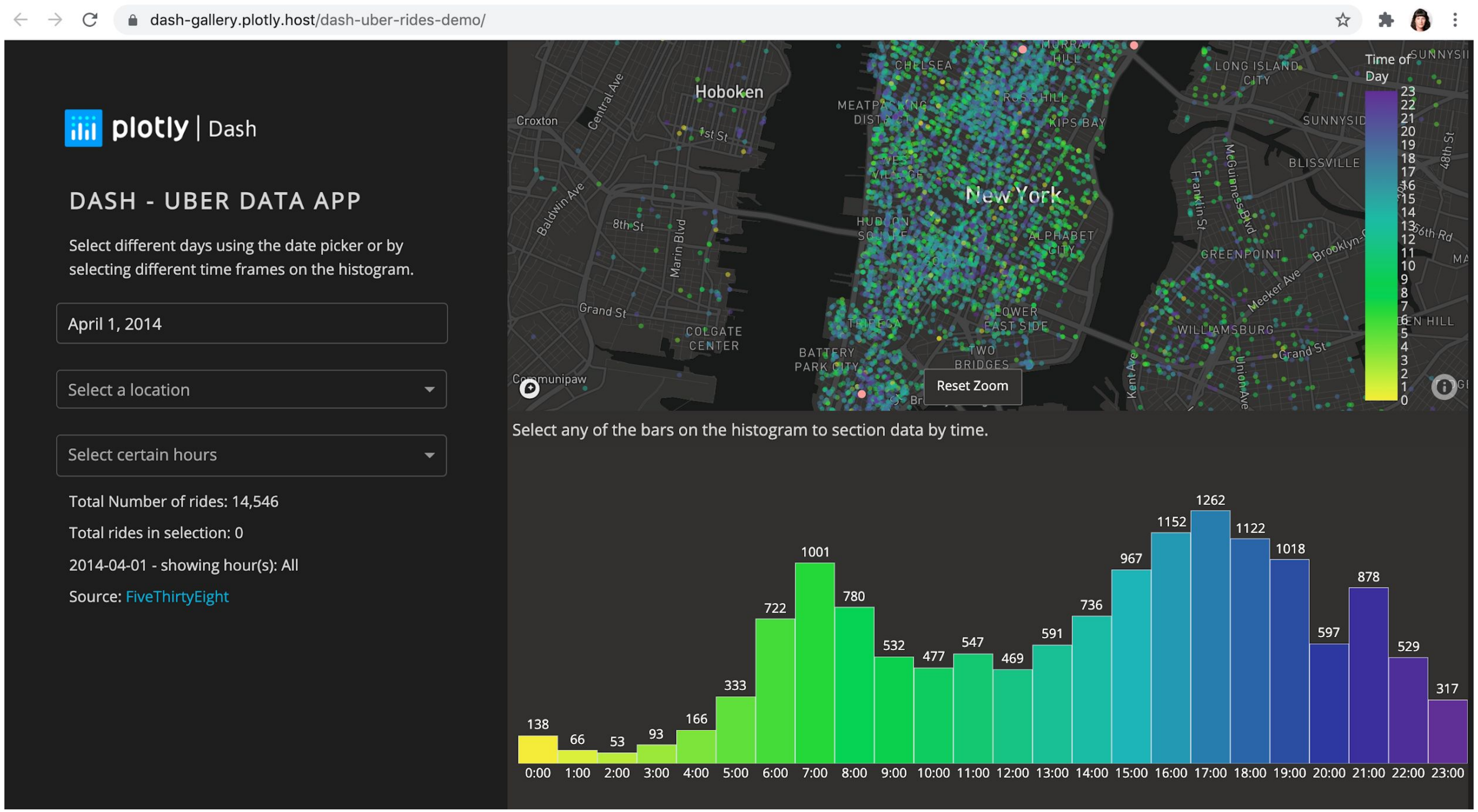
Zetabytes =
1 000 000 000 000 GB

Enter data analysis

Data Analyst *(You soon!)*



Examples of data analysis insights



DASH - UBER DATA APP

Select different days using the date picker or by selecting different time frames on the histogram.

April 1, 2014

Select a location

Select certain hours

Total Number of rides: 14,546

Total rides in selection: 0

2014-04-01 - showing hour(s): All

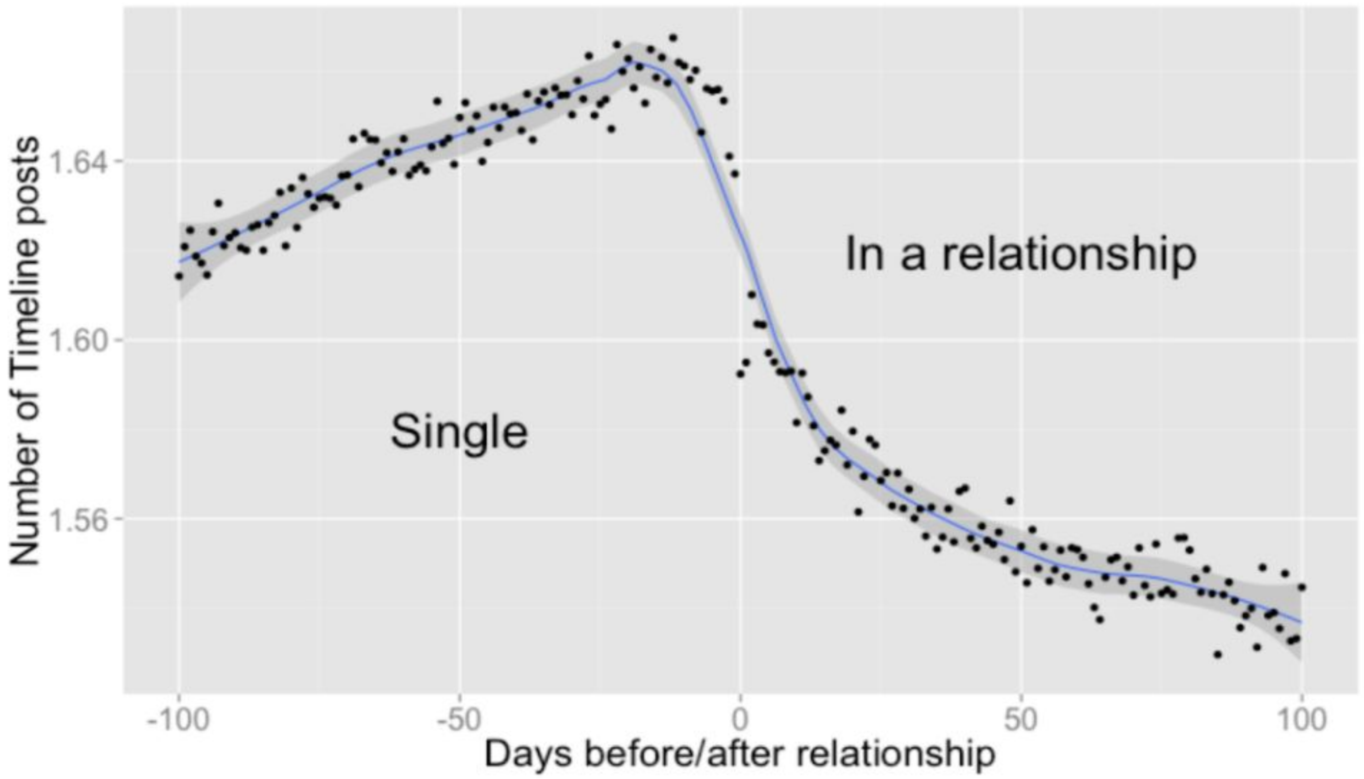
Source: FiveThirtyEight

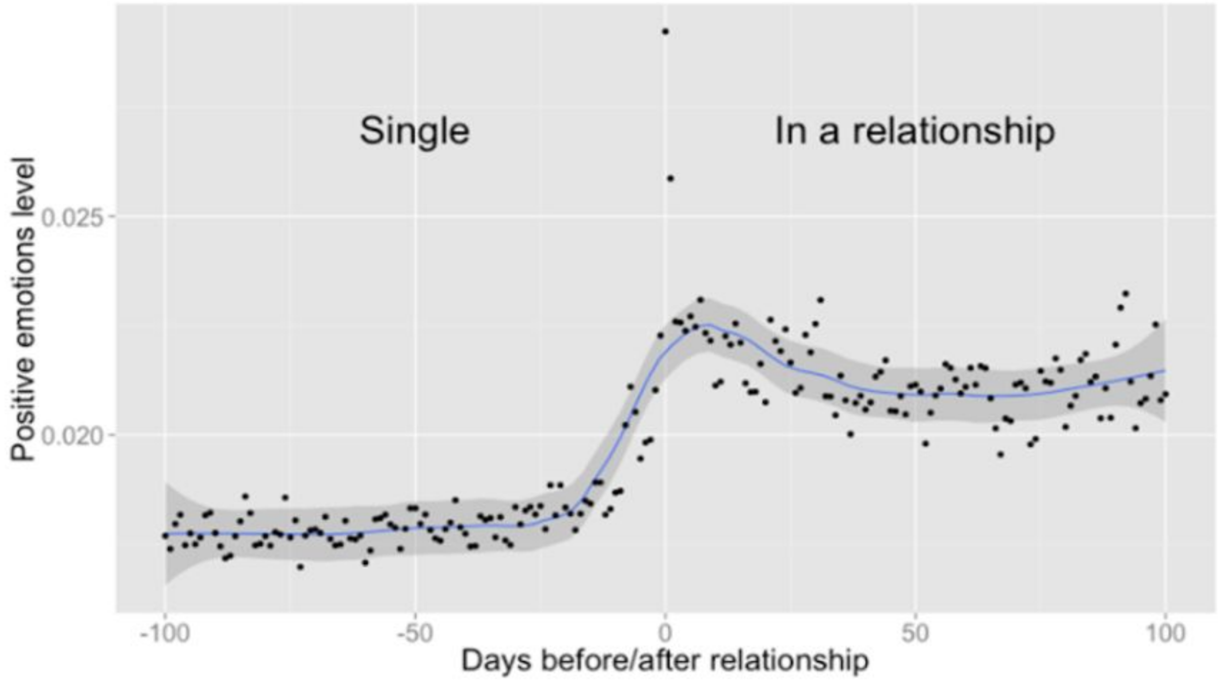
Select any of the bars on the histogram to section data by time.

Facebook's "Formation of love"

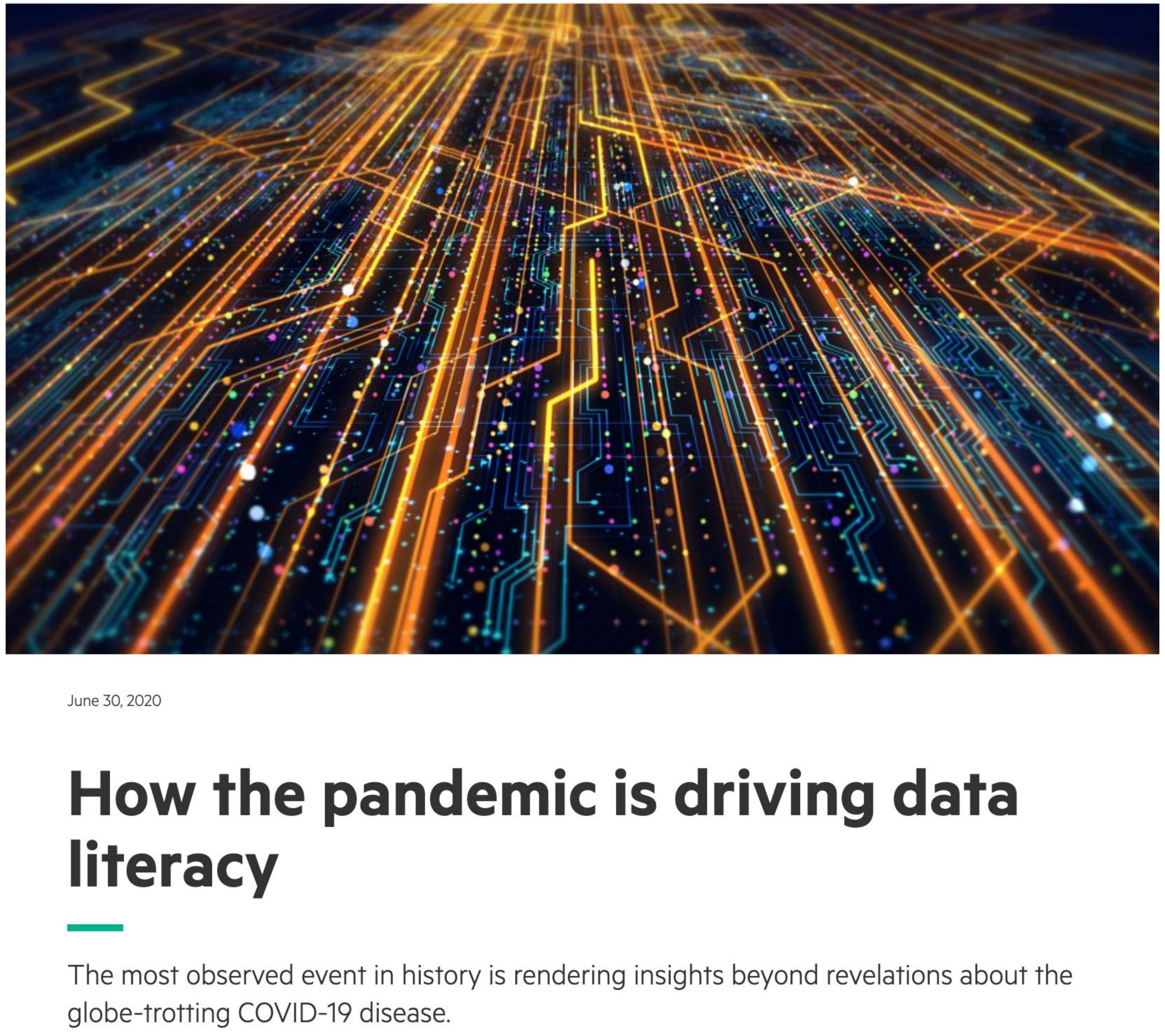
Facebook knows you're single – and it has nothing to do with your relationship status

- Facebook analyzes timeline post frequency and sentiment
- Able to recognize when a relationship is formed by:
  - number of posts decrease
  - positive emotions in posts increase





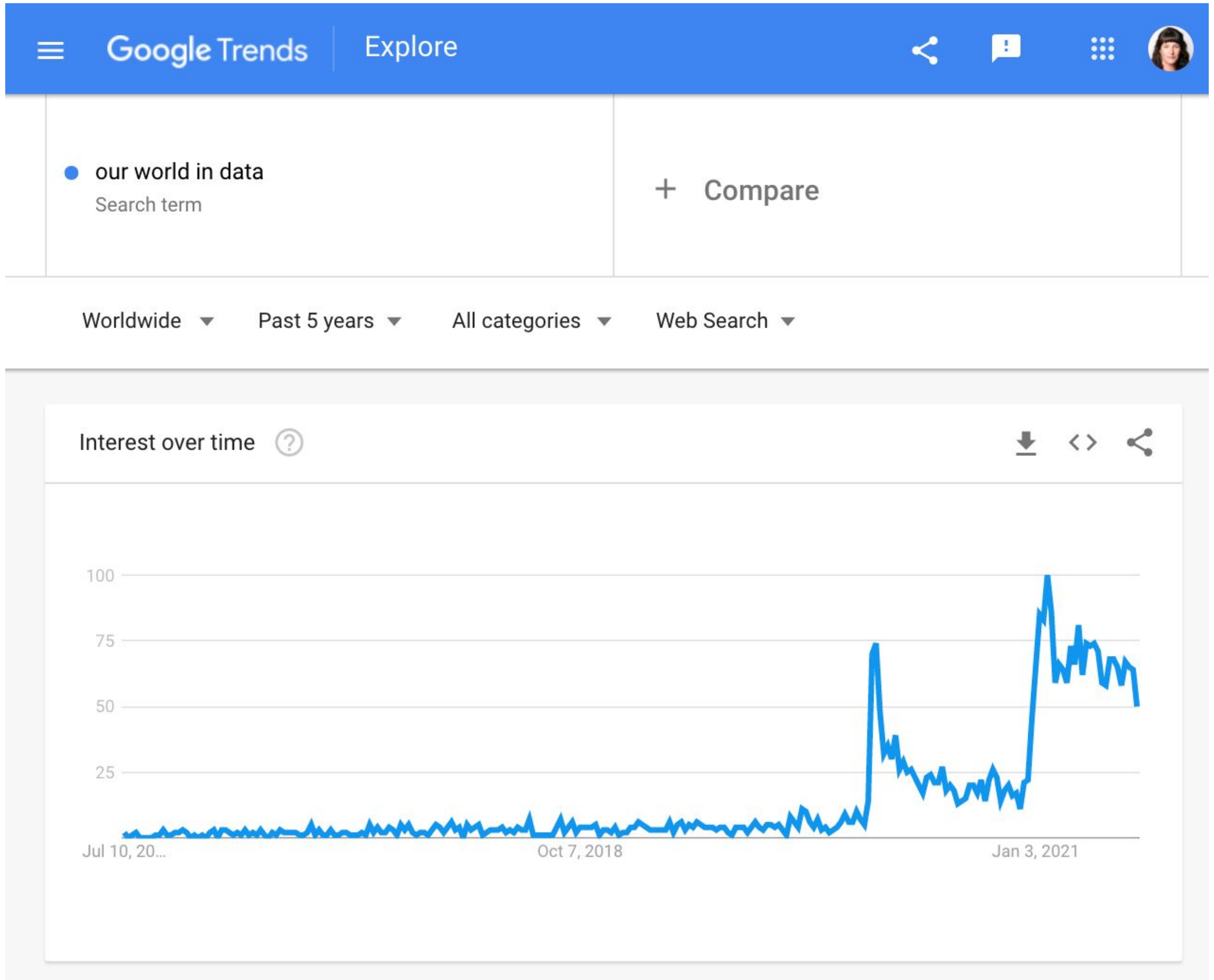
Examples of data analysis insights



June 30, 2020

How the pandemic is driving data literacy

The most observed event in history is rendering insights beyond revelations about the globe-trotting COVID-19 disease.

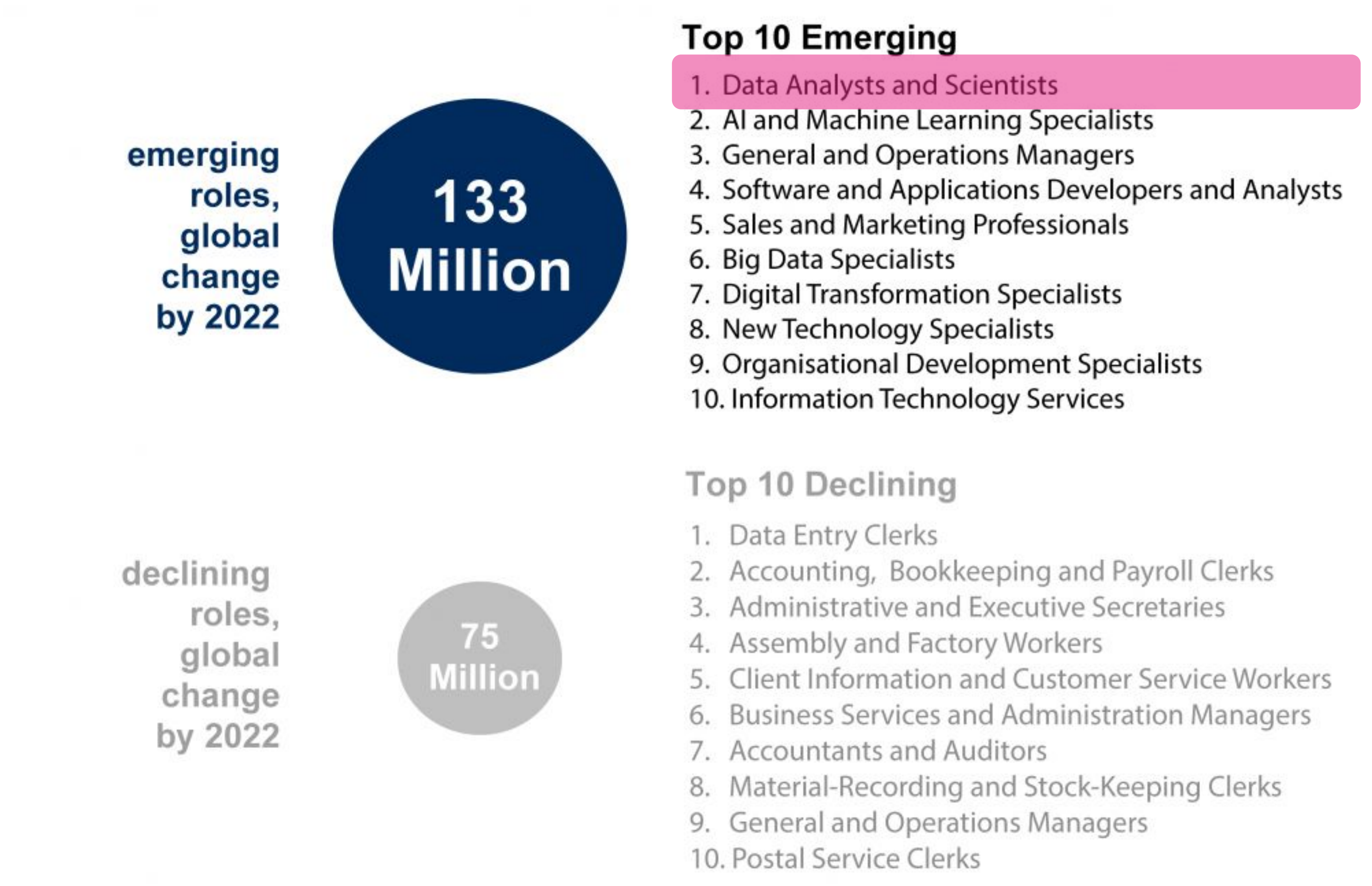


our world in data
Search term

Worldwide ▾ Past 5 years ▾ All categories ▾ Web Search ▾

Interest over time

The jobs landscape in 2022



emerging roles, global change by 2022

133 Million

Top 10 Emerging

1. Data Analysts and Scientists
2. AI and Machine Learning Specialists
3. General and Operations Managers
4. Software and Applications Developers and Analysts
5. Sales and Marketing Professionals
6. Big Data Specialists
7. Digital Transformation Specialists
8. New Technology Specialists
9. Organisational Development Specialists
10. Information Technology Services

declining roles, global change by 2022

75 Million

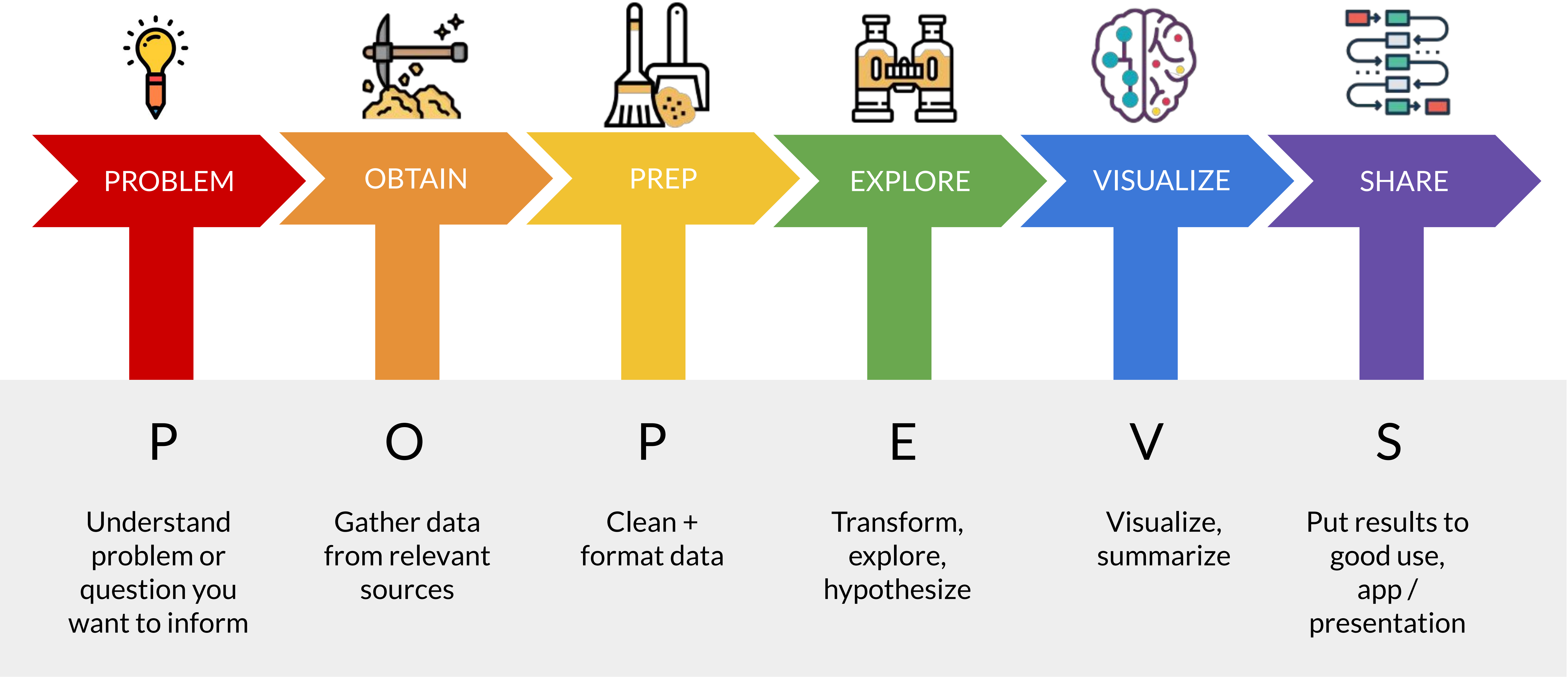
Top 10 Declining

1. Data Entry Clerks
2. Accounting, Bookkeeping and Payroll Clerks
3. Administrative and Executive Secretaries
4. Assembly and Factory Workers
5. Client Information and Customer Service Workers
6. Business Services and Administration Managers
7. Accountants and Auditors
8. Material-Recording and Stock-Keeping Clerks
9. General and Operations Managers
10. Postal Service Clerks

WORLD ECONOMIC FORUM

COMMITTED TO IMPROVING THE STATE OF THE WORLD

What we are going to learn over the next 8 weeks:



PROBLEM

P

Understand problem or question you want to inform

OBTAIN

O

Gather data from relevant sources

PREP

P

Clean + format data

EXPLORE

E

Transform, explore, hypothesize

VISUALIZE

V

Visualize, summarize

SHARE

S

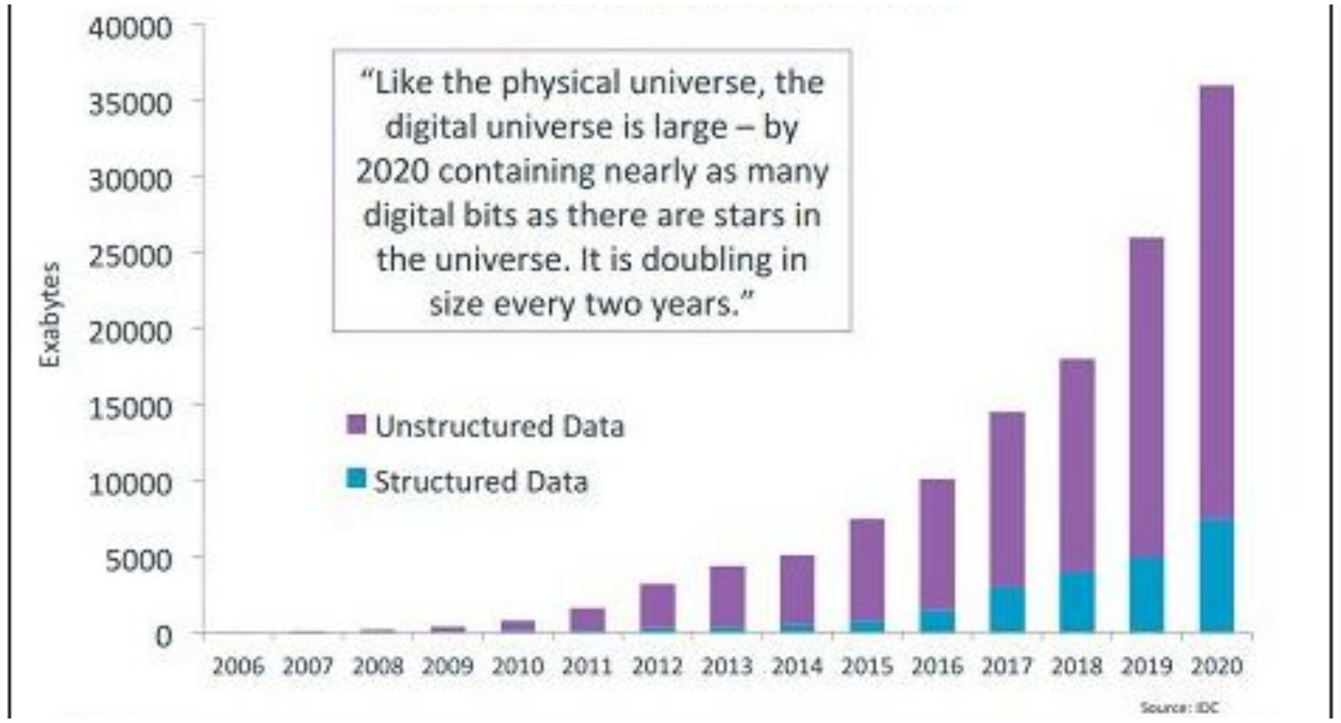
Put results to good use, app / presentation

HOW: Python for data analysis

Class 2

Why not just Excel?

- Excel is limited to 1,048,576 rows
- Generally manual non repeatable process
- Not easy to automate repeat analyses
- We are limited by what kind of data we can process in Excel (only structured)



"Like the physical universe, the digital universe is large – by 2020 containing nearly as many digital bits as there are stars in the universe. It is doubling in size every two years."

Source: IDC

PYTHON
VERSUS
EXCEL
COMPARING THE 2 ANALYTICS TOOLS
Simplicity
Automation
Scalability
Connectivity

Appendix 8, 9

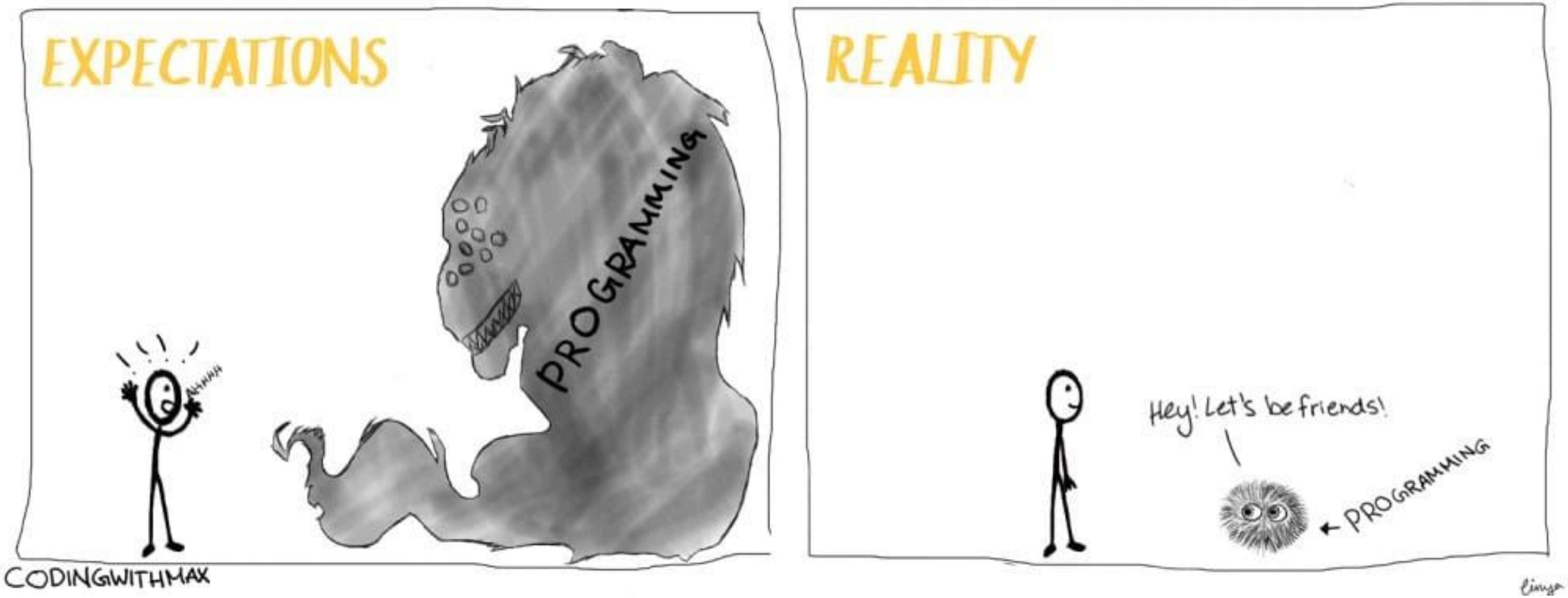
Why Python?

- Python is a free open source programming language
- Large online user community resources and additional guides and tutorials
- Ready-made tools and libraries to support data manipulation, visualization, dashboards and much, much more
- Easy to learn: it is not as hard as it seems (I promise)

Did you know?

Python's name isn't about a snake, but it is named by a popular British comedy group Monty Python (from the 1970s) because Guido (founder) is personally a huge fan of Monty Python's Flying Circus and due to his irreverent mood, he named the project 'Python'.

Expectations vs. reality

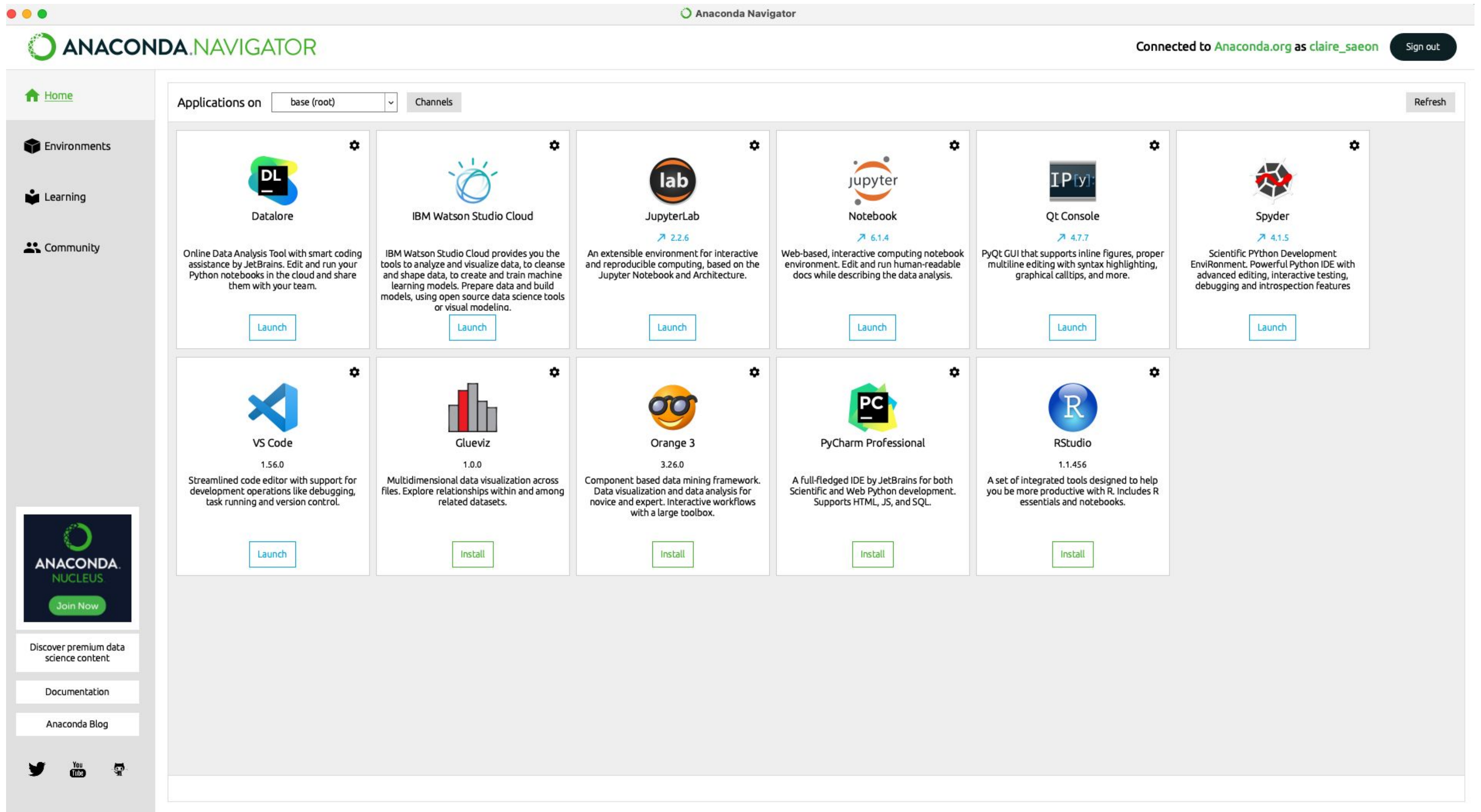


Getting set up: Anaconda & Jupyter

Class 3

Environment we will be using

- **Anaconda** is a distribution of packages built for data analysis
- It comes with **conda**, a package, and environment manager



Datalore
Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

IBM Watson Studio Cloud
IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

JupyterLab
2.2.6
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook
6.1.4
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Qt Console
4.7.7
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Spyder
4.1.5
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

VS Code
1.56.0
Streamlined code editor with support for development operations like debugging, task running and version control.

Glueviz
1.0.0
Multidimensional data visualization across files. Explore relationships within and among related datasets.

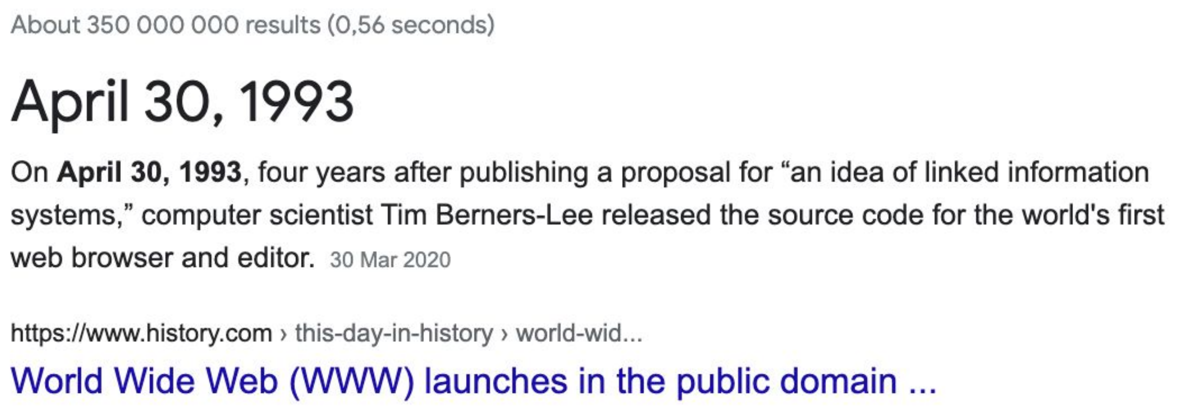
Orange 3
3.26.0
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

PyCharm Professional
A full-fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.

RStudio
1.1.456
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

What is a package?

- Essentially a package is an add on to the basic Python language (which was written in 1991)
- A package is a group of commands (aka functions) that allow you to do different kinds of analyses
- E.g., you will get visualization packages or packages that allow us to work with Twitter data or packages specific to NFL



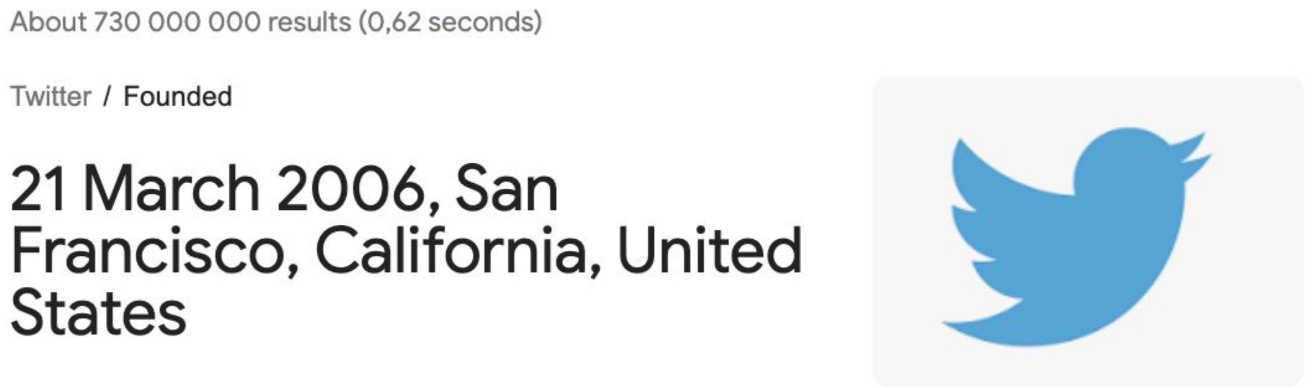
About 350 000 000 results (0,56 seconds)

April 30, 1993

On **April 30, 1993**, four years after publishing a proposal for “an idea of linked information systems,” computer scientist Tim Berners-Lee released the source code for the world's first web browser and editor. 30 Mar 2020

https://www.history.com › this-day-in-history › world-wid...

World Wide Web (WWW) launches in the public domain ...



About 730 000 000 results (0,62 seconds)

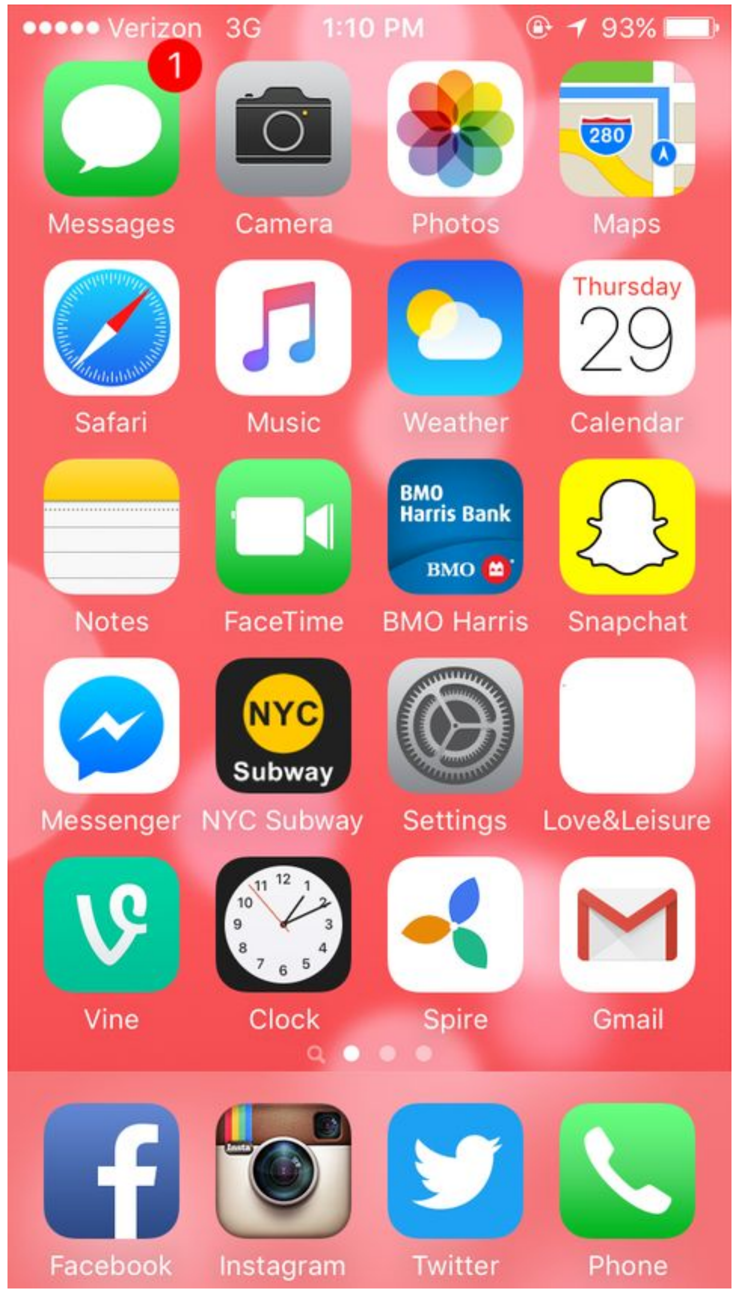
Twitter / Founded

21 March 2006, San Francisco, California, United States

What is a package?

- Essentially a package is an add on to the basic Python language (which was written in 1991)
- A package is a group of commands (aka functions) that allow you to do different kinds of analyses
- E.g., you will get visualization packages or packages that allow us to work with Twitter data or packages specific to NFL

- I find the easiest way to think about packages is like mobile apps

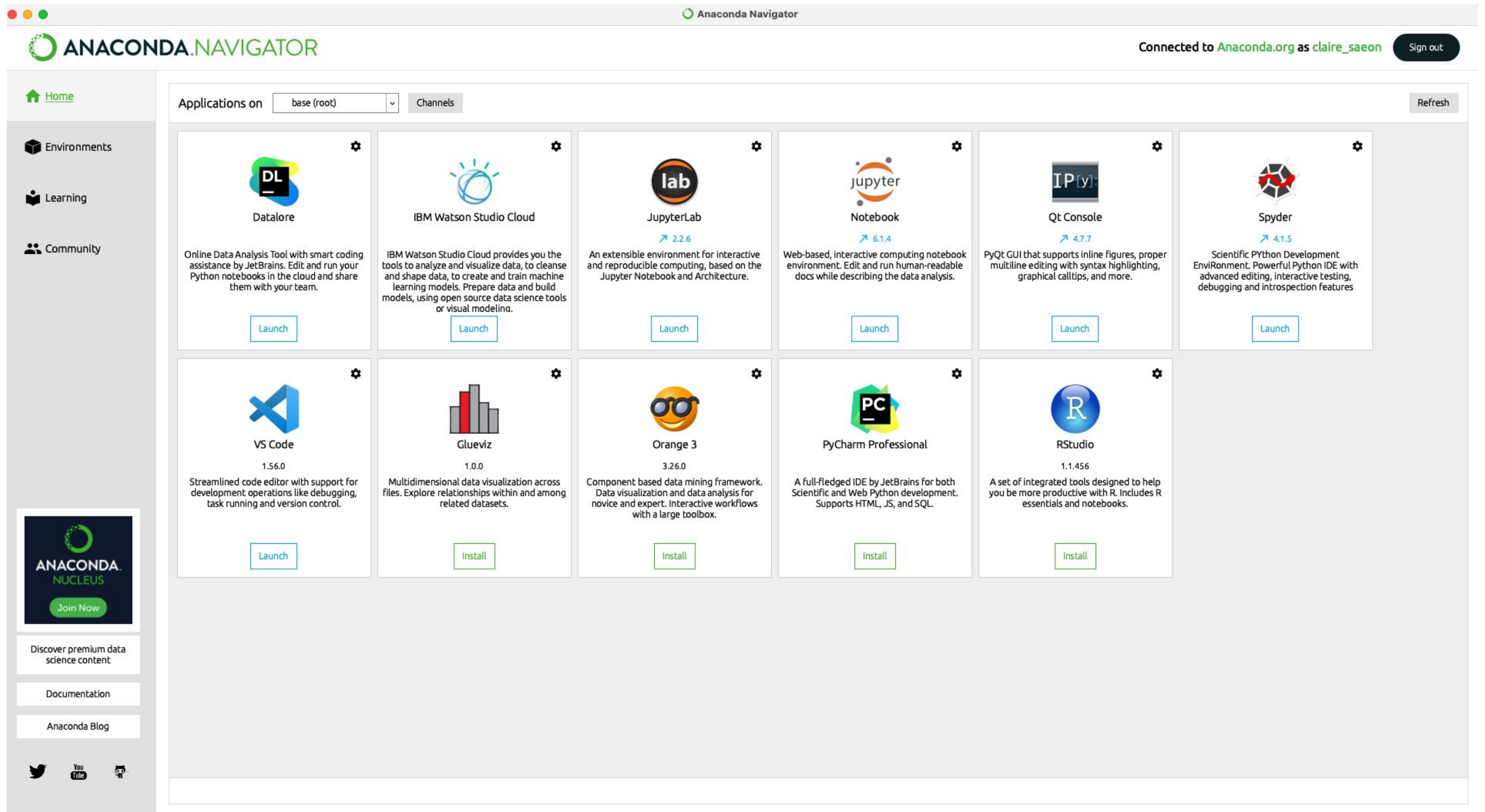


Environment we will be using

- **Anaconda** is a distribution of packages built for data analysis.
- It comes with **conda**, a package, and environment manager

- **Jupyter Notebooks** is a free, open-source, interactive web tool known as a computational notebook

- It combines code, output, text and multimedia resources in a single document



Connected to Anaconda.org as claire_saeon

Datalore: Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

IBM Watson Studio Cloud: IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

JupyterLab 2.2.6: An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook 6.1.4: Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Qt Console 4.7.7: PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Spyder 4.1.5: Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

VS Code 1.56.0: Streamlined code editor with support for development operations like debugging, task running and version control.

Glueviz 1.0.0: Multidimensional data visualization across files. Explore relationships within and among related datasets.

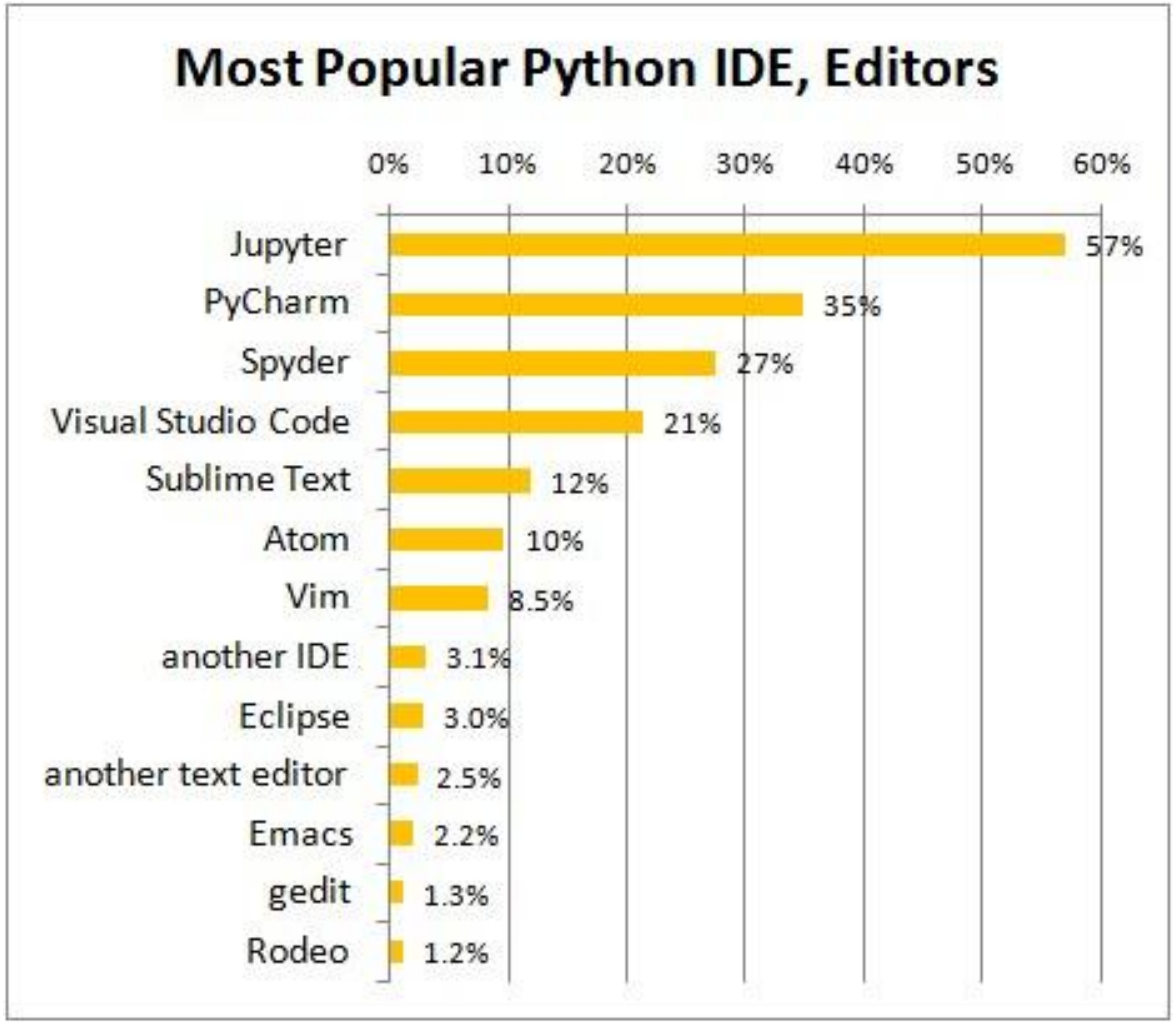
Orange 3 3.26.0: Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

PyCharm Professional: A full-fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.

RStudio 1.1.456: A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

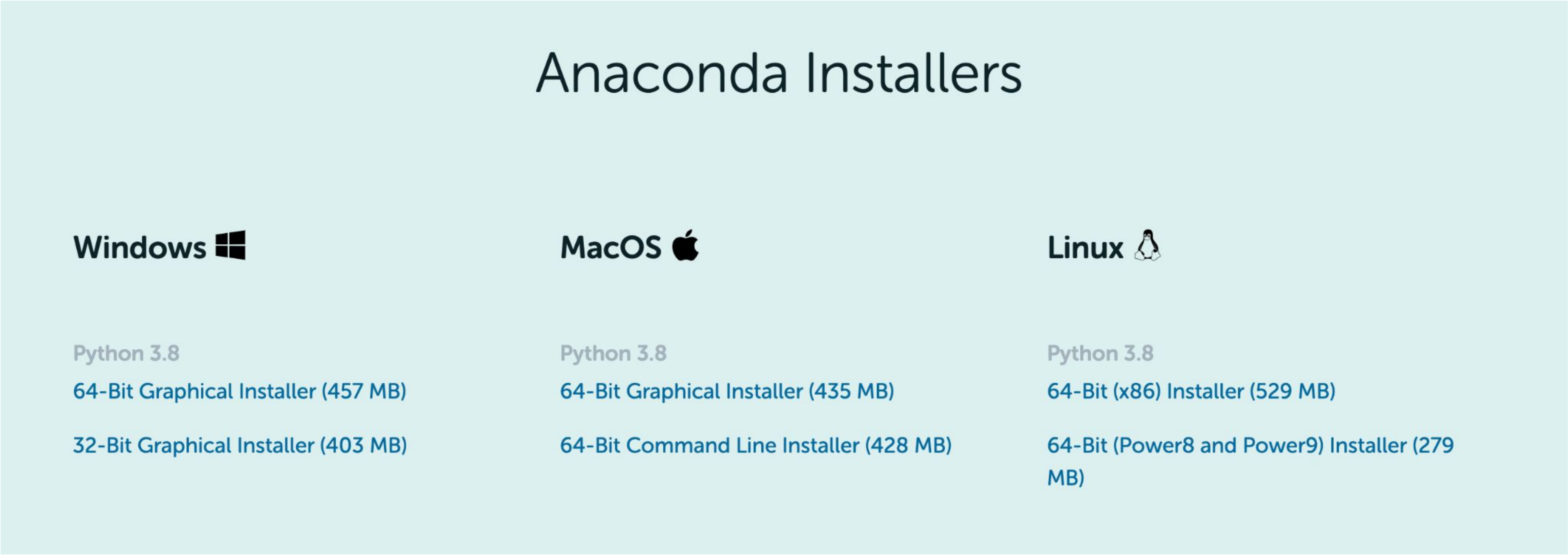
Discover premium data science content

Why Jupyter?


Most Popular Python IDE, Editors
0%
10%
20%
30%
40%
50%
60%
Jupyter
57%
PyCharm
35%
Spyder
27%
Visual Studio Code
21%
Sublime Text
12%
Atom
10%
Vim
8.5%
another IDE
3.1%
Eclipse
3.0%
another text editor
2.5%
Emacs
2.2%
gedit
1.3%
Rodeo
1.2%

Downloading and Installing

https://www.anaconda.com/products/individual



Anaconda Installers

Windows

Python 3.8

64-Bit Graphical Installer (457 MB)

32-Bit Graphical Installer (403 MB)

MacOS

Python 3.8

64-Bit Graphical Installer (435 MB)

64-Bit Command Line Installer (428 MB)

Linux

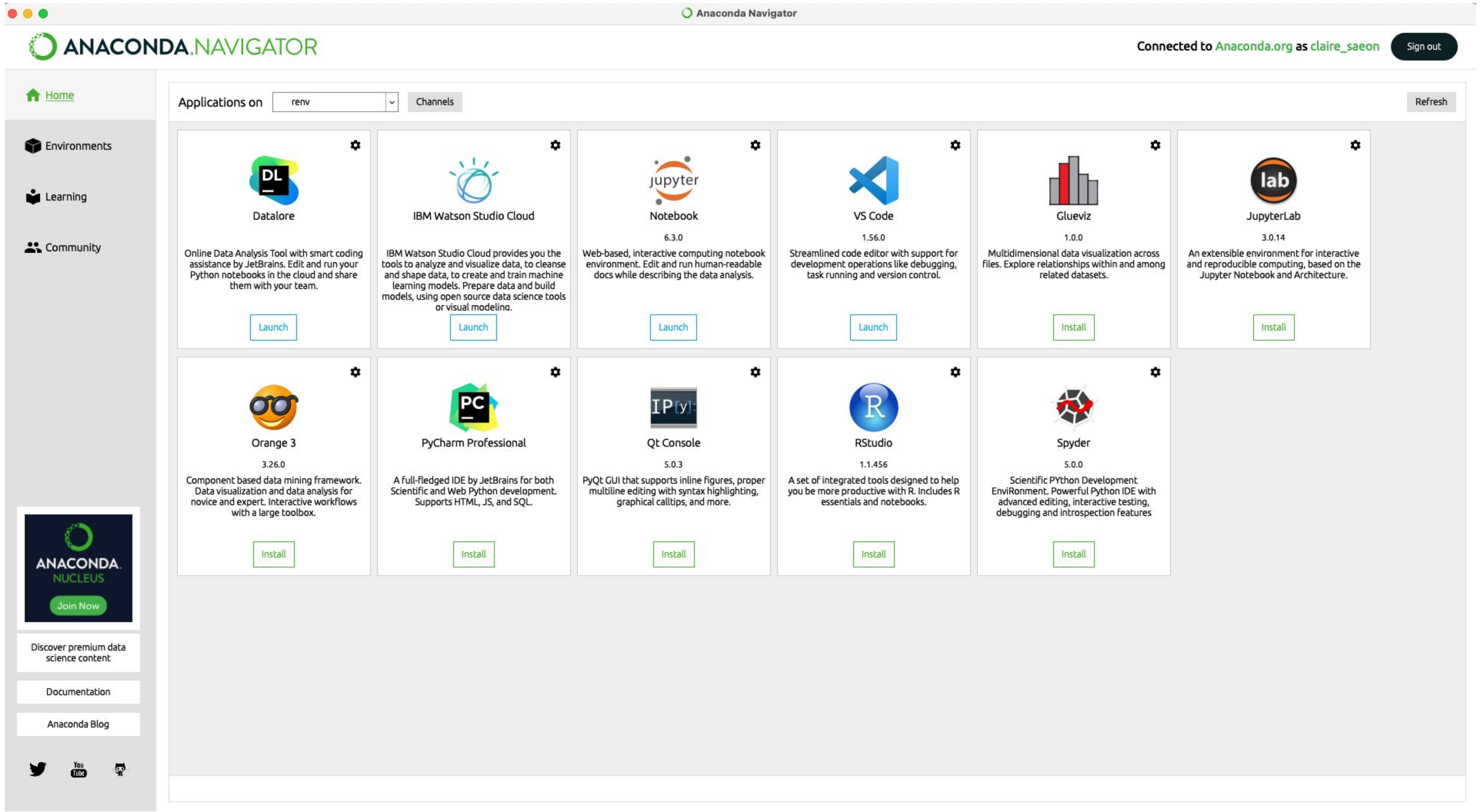
Python 3.8

64-Bit (x86) Installer (529 MB)

64-Bit (Power8 and Power9) Installer (279 MB)

Anaconda navigator home page

You should create an account to get started



Datalore

Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

IBM Watson Studio Cloud

IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

Notebook
6.3.0

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

VS Code
1.56.0

Streamlined code editor with support for development operations like debugging, task running and version control.

Glueviz
1.0.0

Multidimensional data visualization across files. Explore relationships within and among related datasets.

JupyterLab
3.0.14

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Orange 3
3.26.0

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

PyCharm Professional

A full-fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.

Qt Console
5.0.3

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

RStudio
1.1.456

A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Spyder
5.0.0

Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

HELP!!! – I can’t get Jupyter working

Sidebar

Google Colabatory

“Google Colab is a cloud based Jupyter Notebook that allows you to run and execute your Python code remotely and save your work into your personal Google Drive. It’s ace card is the availability of the free graphics processing units (GPU) to allow fast training.” - M. Parmar

- Colab allows you to write and execute Python in your browser, with
- Zero configuration required
- Free access to GPUs
- Easy sharing

+ colab

Google Colab Notebooks

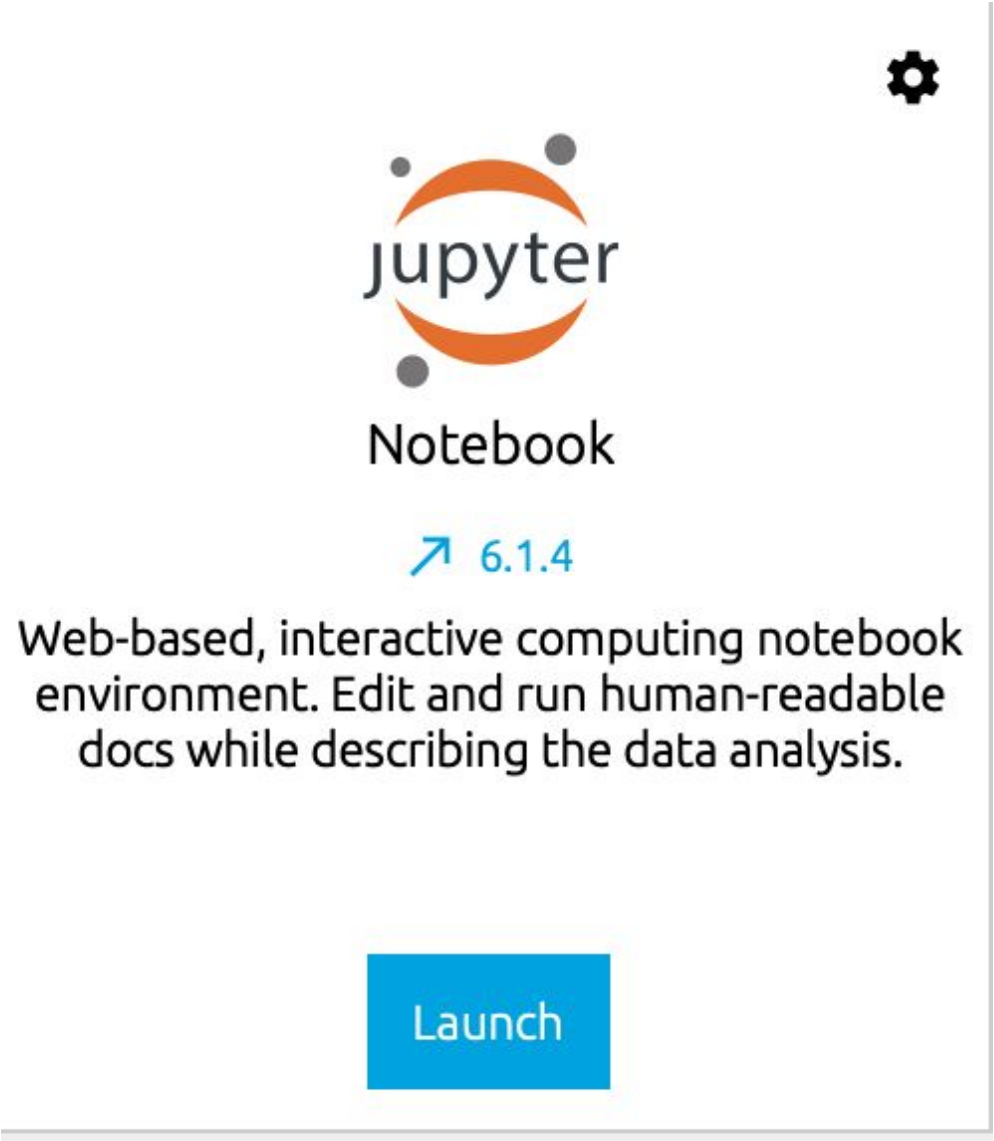
https://colab.research.google.com/

HUH? Jupyter & Python basics

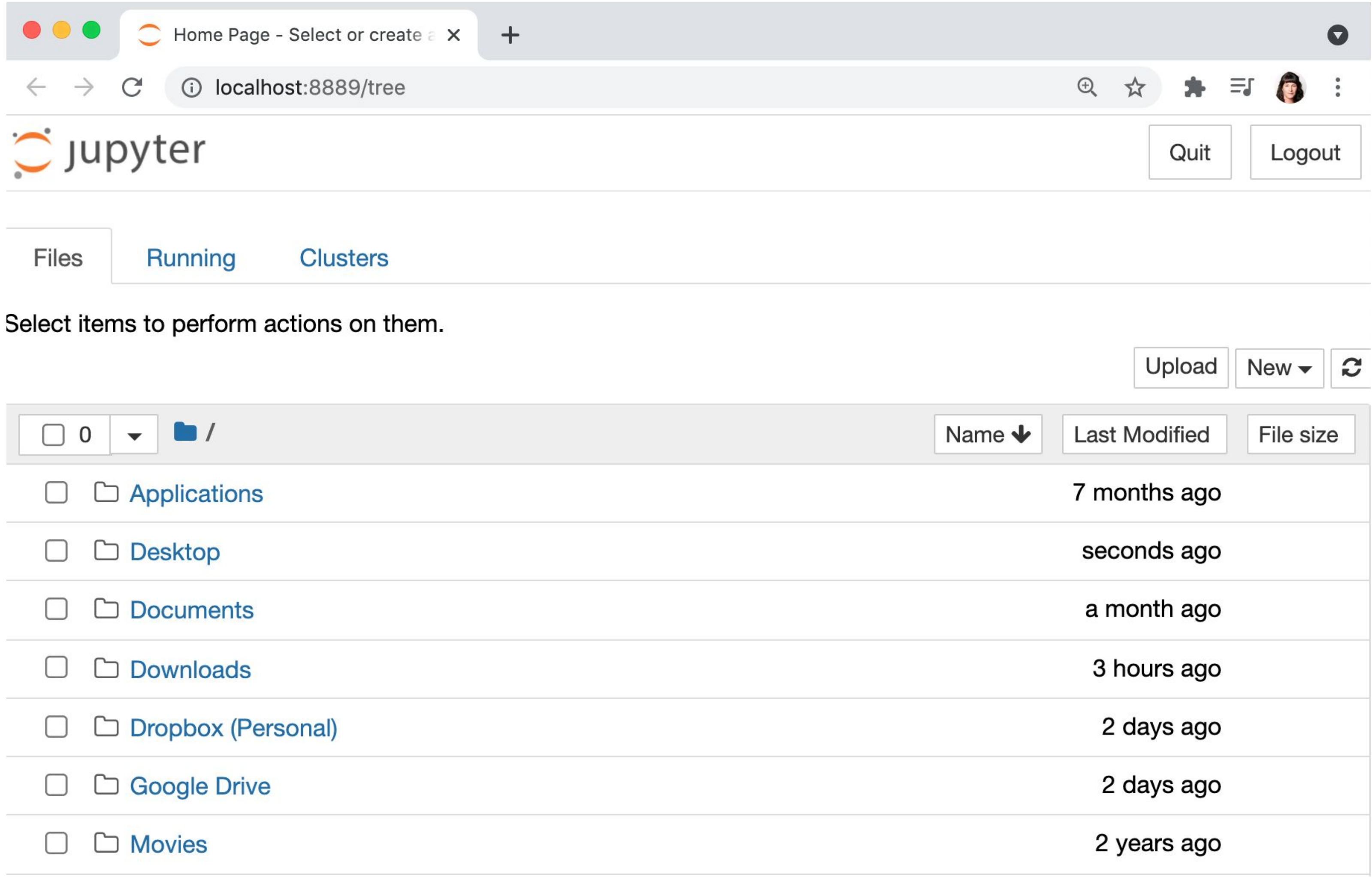
Class 4

Using Jupyter Notebook for coding

Launch Jupyter Notebook from Anaconda’s home page. This will open a window in your default web browser.



Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

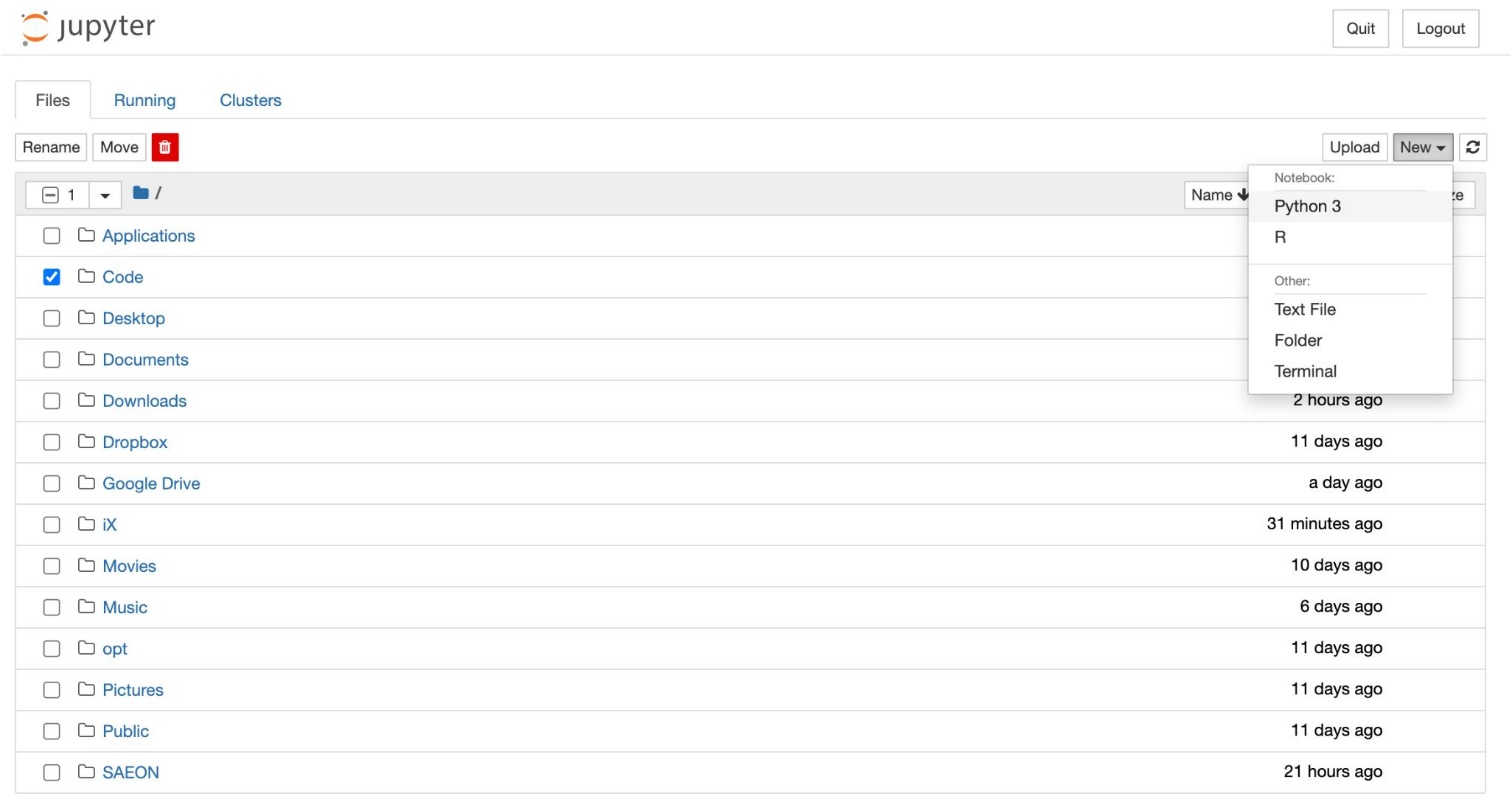


Select items to perform actions on them.

Name	Last Modified	File size
☐ Applications	7 months ago	
☐ Desktop	seconds ago	
☐ Documents	a month ago	
☐ Downloads	3 hours ago	
☐ Dropbox (Personal)	2 days ago	
☐ Google Drive	2 days ago	
☐ Movies	2 years ago	

Using Jupyter Notebook for coding

Click a folder name of your choosing (e.g. Code). Click New, and then choose Python 3 under Notebook or navigate to an existing Jupyter Notebook (.ipynb).



☐	Applications	
☑	Code	
☐	Desktop	
☐	Documents	
☐	Downloads	2 hours ago
☐	Dropbox	11 days ago
☐	Google Drive	a day ago
☐	iX	31 minutes ago
☐	Movies	10 days ago
☐	Music	6 days ago
☐	opt	11 days ago
☐	Pictures	11 days ago
☐	Public	11 days ago
☐	SAEON	21 hours ago

Let’s code!

HELP – and other handy resources

Class 5

Lifecycle of a question

Problem

Uh oh - your code isn’t working or you don’t understand a problem!

Check Notes

Is your problem related to a function or package we covered in class? Have we covered it already?

Google It

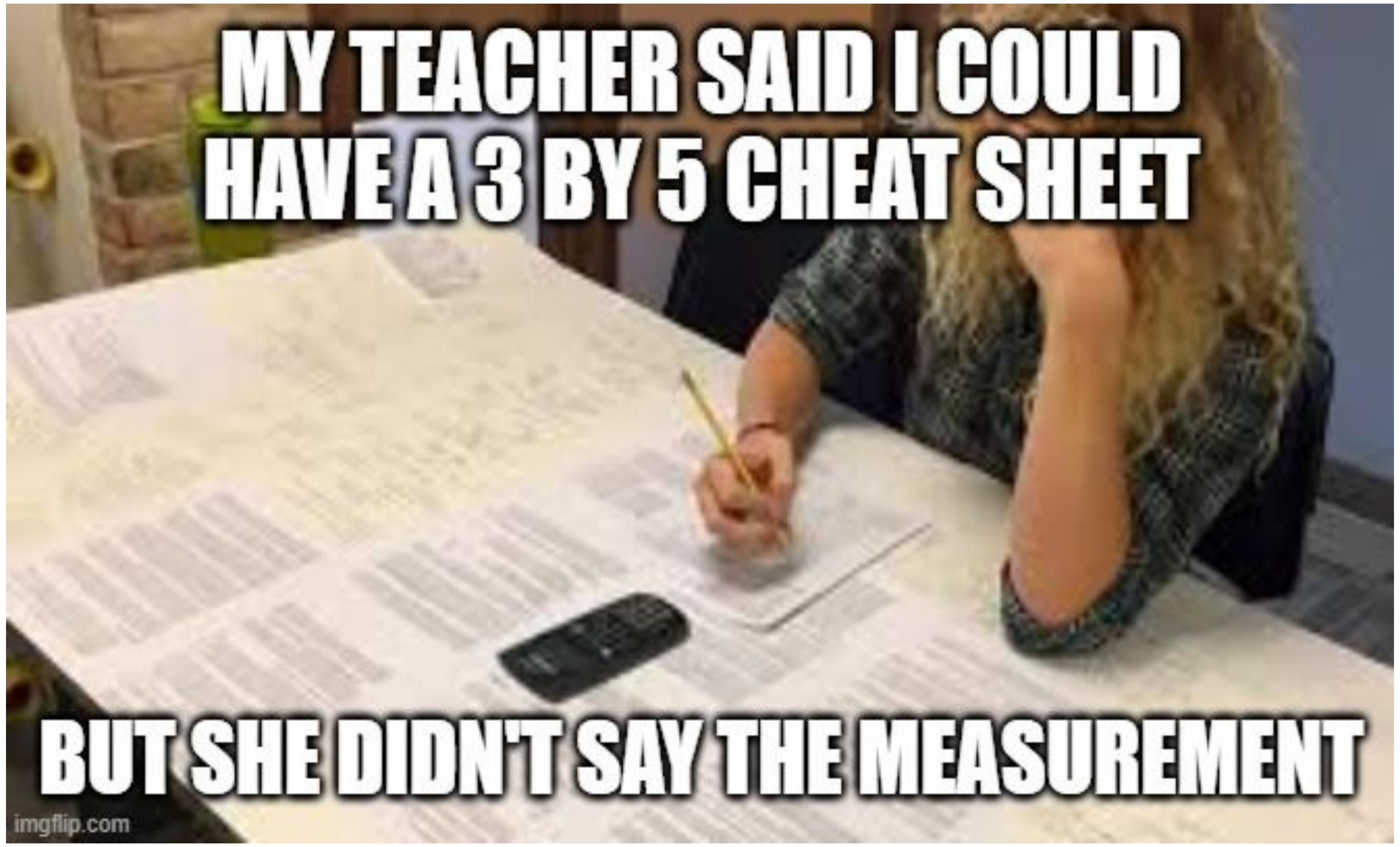
Try to find a nifty YouTube tutorial, or someone else who had the same problem. Stack Exchange is your friend!

Get Help

Use the **ix_stack_exchange** channel or **office hours** to ask for help. Make sure to include **screenshots of your code** and what you’ve already tried!

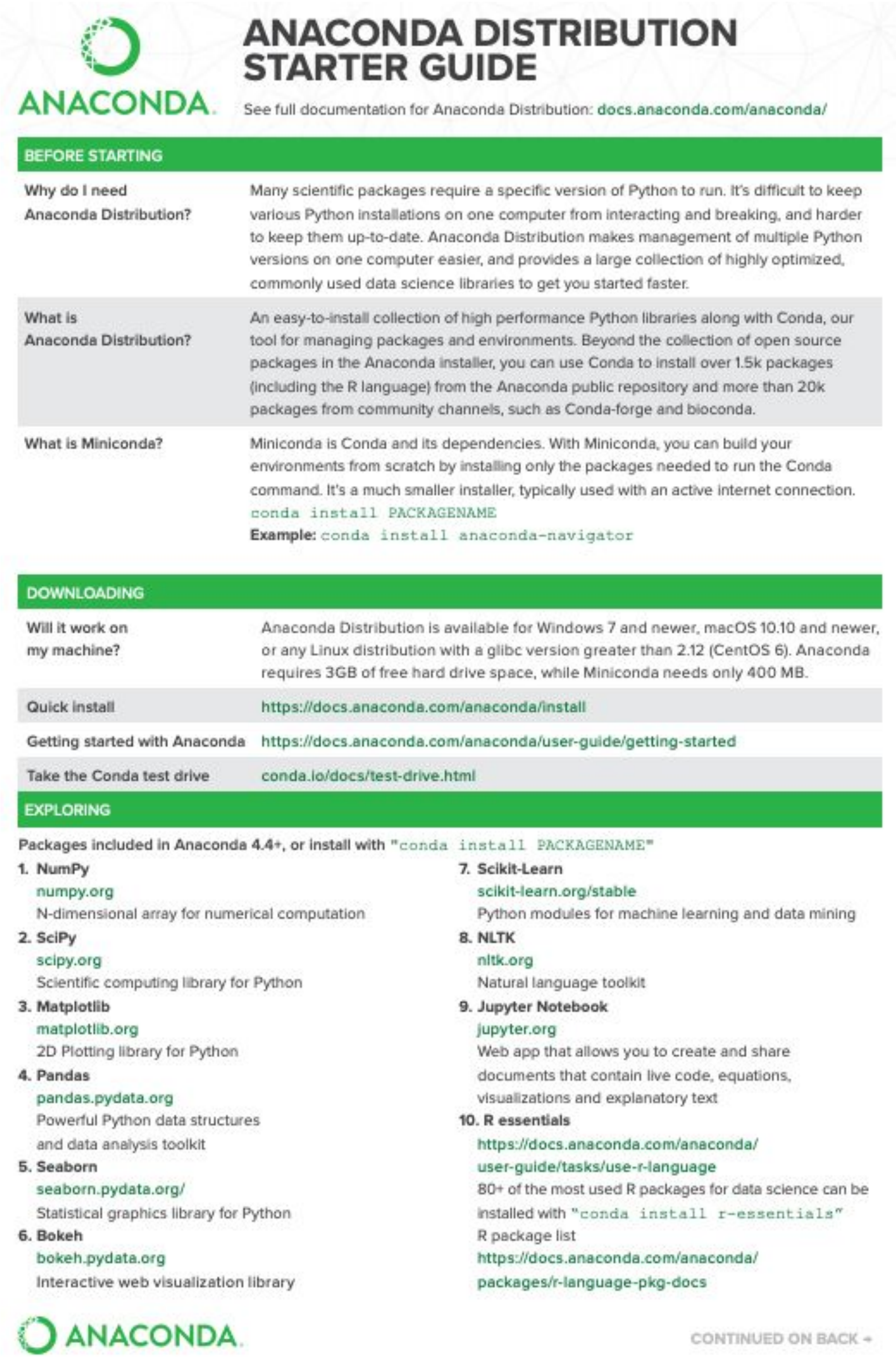
Cheat sheets

Downloadable resources from DataCamp


MY TEACHER SAID I COULD HAVE A 3 BY 5 CHEAT SHEET
BUT SHE DIDN'T SAY THE MEASUREMENT
imgflip.com

Anaconda cheat sheet

Link



ANACONDA DISTRIBUTION STARTER GUIDE

See full documentation for Anaconda Distribution: docs.anaconda.com/anaconda/

BEFORE STARTING	
Why do I need Anaconda Distribution?	Many scientific packages require a specific version of Python to run. It's difficult to keep various Python installations on one computer from interacting and breaking, and harder to keep them up-to-date. Anaconda Distribution makes management of multiple Python versions on one computer easier, and provides a large collection of highly optimized, commonly used data science libraries to get you started faster.
What is Anaconda Distribution?	An easy-to-install collection of high performance Python libraries along with Conda, our tool for managing packages and environments. Beyond the collection of open source packages in the Anaconda installer, you can use Conda to install over 1.5k packages (including the R language) from the Anaconda public repository and more than 20k packages from community channels, such as Conda-forge and bioconda.
What is Miniconda?	Miniconda is Conda and its dependencies. With Miniconda, you can build your environments from scratch by installing only the packages needed to run the Conda command. It's a much smaller installer, typically used with an active internet connection. `conda install PACKAGENAME` **Example:** `conda install anaconda-navigator`

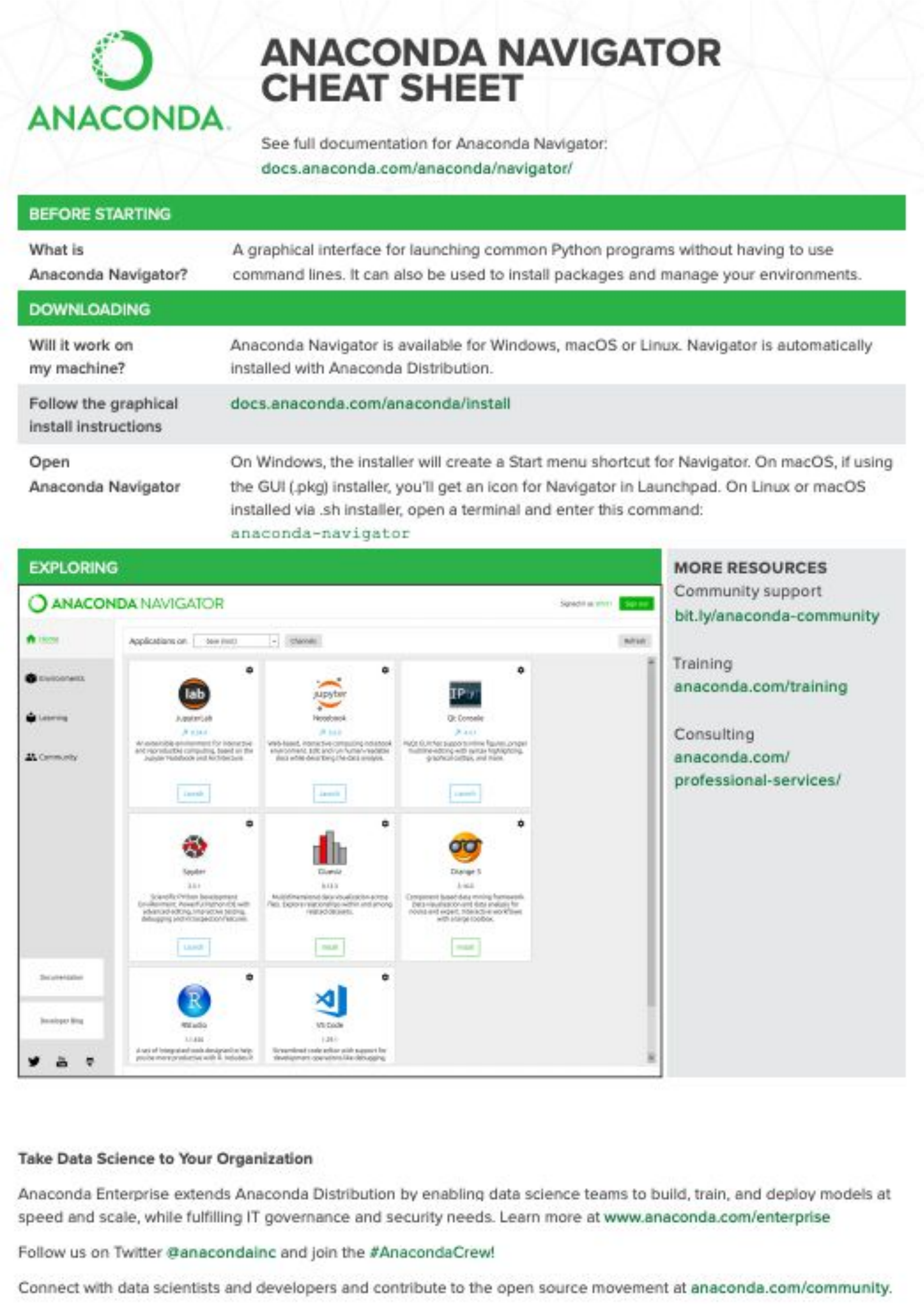
DOWNLOADING	
Will it work on my machine?	Anaconda Distribution is available for Windows 7 and newer, macOS 10.10 and newer, or any Linux distribution with a glibc version greater than 2.12 (CentOS 6). Anaconda requires 3GB of free hard drive space, while Miniconda needs only 400 MB.
Quick install	https://docs.anaconda.com/anaconda/install
Getting started with Anaconda	https://docs.anaconda.com/anaconda/user-guide/getting-started
Take the Conda test drive	conda.io/docs/test-drive.html

EXPLORING

Packages included in Anaconda 4.4+, or install with "`conda install PACKAGENAME`"

1. **NumPy**
   numpy.org
   N-dimensional array for numerical computation
2. **SciPy**
   scipy.org
   Scientific computing library for Python
3. **Matplotlib**
   matplotlib.org
   2D Plotting library for Python
4. **Pandas**
   pandas.pydata.org
   Powerful Python data structures and data analysis toolkit
5. **Seaborn**
   seaborn.pydata.org/
   Statistical graphics library for Python
6. **Bokeh**
   bokeh.pydata.org
   Interactive web visualization library
7. **Scikit-Learn**
   scikit-learn.org/stable
   Python modules for machine learning and data mining
8. **NLTK**
   nltk.org
   Natural language toolkit
9. **Jupyter Notebook**
   jupyter.org
   Web app that allows you to create and share documents that contain live code, equations, visualizations and explanatory text
10. **R essentials**
   https://docs.anaconda.com/anaconda/user-guide/tasks/use-r-language
   80+ of the most used R packages for data science can be installed with "`conda install r-essentials`"
   R package list
   https://docs.anaconda.com/anaconda/packages/r-language-pkg-docs

CONTINUED ON BACK →



ANACONDA NAVIGATOR CHEAT SHEET

See full documentation for Anaconda Navigator:
docs.anaconda.com/anaconda/navigator/

BEFORE STARTING	
What is Anaconda Navigator?	A graphical interface for launching common Python programs without having to use command lines. It can also be used to install packages and manage your environments.

DOWNLOADING	
Will it work on my machine?	Anaconda Navigator is available for Windows, macOS or Linux. Navigator is automatically installed with Anaconda Distribution.
Follow the graphical install instructions	docs.anaconda.com/anaconda/install
Open Anaconda Navigator	On Windows, the installer will create a Start menu shortcut for Navigator. On macOS, if using the GUI (.pkg) installer, you'll get an icon for Navigator in Launchpad. On Linux or macOS installed via .sh installer, open a terminal and enter this command: `anaconda-navigator`

EXPLORING

MORE RESOURCES

Community support
bit.ly/anaconda-community

Training
anaconda.com/training

Consulting
anaconda.com/professional-services/

Take Data Science to Your Organization

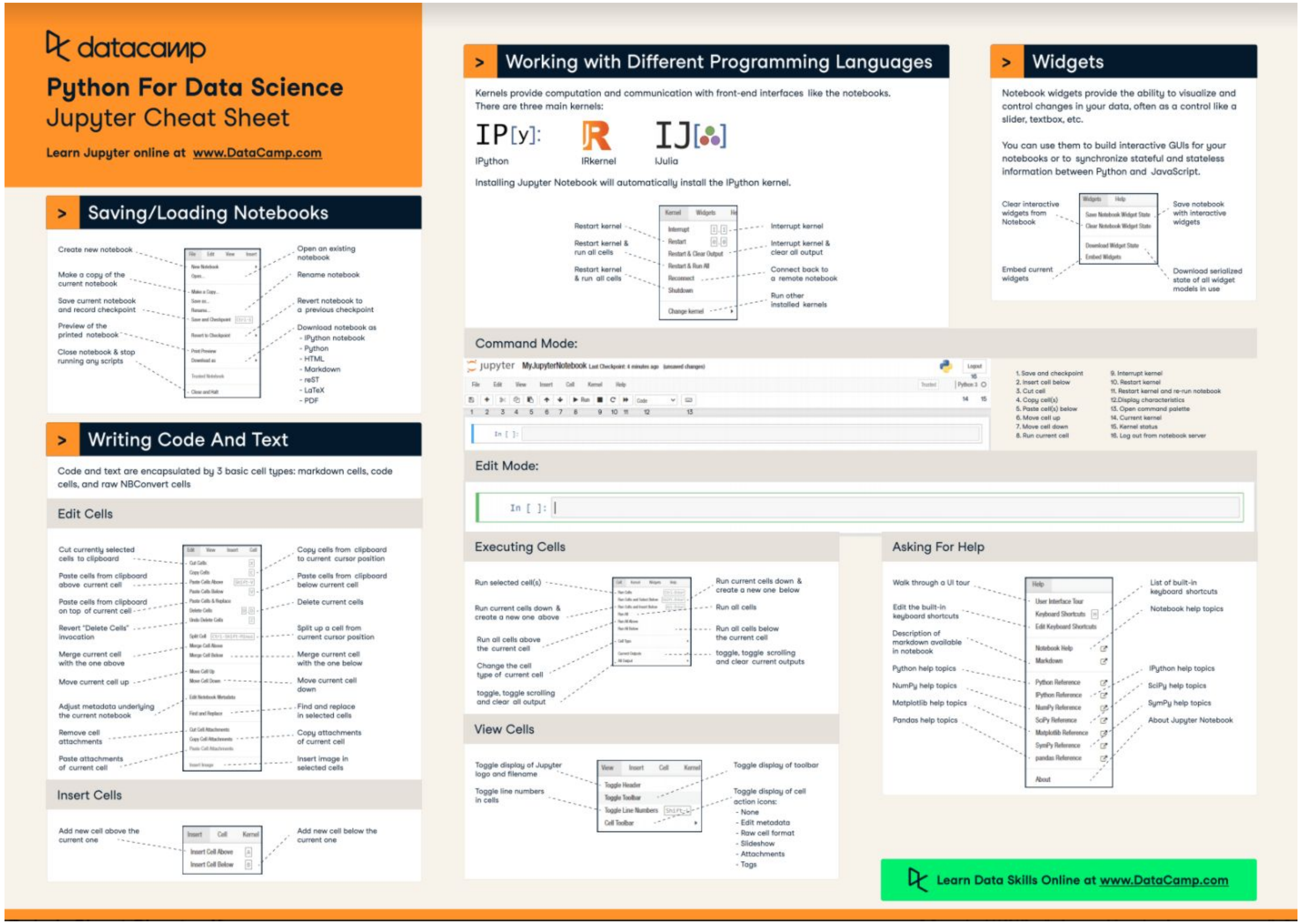
Anaconda Enterprise extends Anaconda Distribution by enabling data science teams to build, train, and deploy models at speed and scale, while fulfilling IT governance and security needs. Learn more at www.anaconda.com/enterprise

Follow us on Twitter @anacondainc and join the #AnacondaCrew!

Connect with data scientists and developers and contribute to the open source movement at anaconda.com/community.

Jupyter Notebooks cheat sheet

Link



datacamp

Python For Data Science

Jupyter Cheat Sheet

Learn Jupyter online at www.DataCamp.com

> Saving/Loading Notebooks

Create new notebook

Make a copy of the current notebook

Save current notebook and record checkpoint

Preview of the printed notebook

Close notebook & stop running any scripts

Open an existing notebook

Rename notebook

Revert notebook to a previous checkpoint

Download notebook as
- IPython notebook
- Python
- HTML
- Markdown
- reST
- LaTeX
- PDF

> Writing Code And Text

Code and text are encapsulated by 3 basic cell types: markdown cells, code cells, and raw NBConvert cells

Edit Cells

Cut currently selected cells to clipboard

Paste cells from clipboard above current cell

Paste cells from clipboard on top of current cell

Revert "Delete Cells" invocation

Merge current cell with the one above

Move current cell up

Adjust metadata underlying the current notebook

Remove cell attachments

Paste attachments of current cell

Copy cells from clipboard to current cursor position

Paste cells from clipboard below current cell

Delete current cells

Split up a cell from current cursor position

Merge current cell with the one below

Move current cell down

Find and replace in selected cells

Copy attachments of current cell

Insert image in selected cells

Insert Cells

Add new cell above the current one

Add new cell below the current one

> Working with Different Programming Languages

Kernels provide computation and communication with front-end interfaces like the notebooks. There are three main kernels:

IPython IRkernel IJulia

Installing Jupyter Notebook will automatically install the IPython kernel.

Restart kernel

Restart kernel & run all cells

Restart kernel & run all cells

Interrupt kernel

Interrupt kernel & clear all output

Connect back to a remote notebook

Run other installed kernels

Command Mode:

1. Save and checkpoint
2. Insert cell below
3. Cut cell
4. Copy cell(s)
5. Paste cell(s) below
6. Move cell up
7. Move cell down
8. Run current cell
9. Interrupt kernel
10. Restart kernel
11. Restart kernel and re-run notebook
12. Display characteristics
13. Open command palette
14. Current kernel
15. Kernel status
16. Log out from notebook server

Edit Mode:

```
In [ ]:
```

Executing Cells

Run selected cell(s)

Run current cells down & create a new one above

Run all cells above the current cell

Change the cell type of current cell

toggle, toggle scrolling and clear all output

Run current cells down & create a new one below

Run all cells

Run all cells below the current cell

toggle, toggle scrolling and clear current outputs

View Cells

Toggle display of Jupyter logo and filename

Toggle line numbers in cells

Toggle display of toolbar

Toggle display of cell action icons:
- None
- Edit metadata
- Raw cell format
- Slideshow
- Attachments
- Tags

> Widgets

Notebook widgets provide the ability to visualize and control changes in your data, often as a control like a slider, textbox, etc.

You can use them to build interactive GUIs for your notebooks or to synchronize stateful and stateless information between Python and JavaScript.

Clear interactive widgets from Notebook

Embed current widgets

Save notebook with interactive widgets

Download serialized state of all widget models in use

Asking For Help

Walk through a UI tour

Edit the built-in keyboard shortcuts

Description of markdown available in notebook

Python help topics

NumPy help topics

Matplotlib help topics

Pandas help topics

List of built-in keyboard shortcuts

Notebook help topics

IPython help topics

SciPy help topics

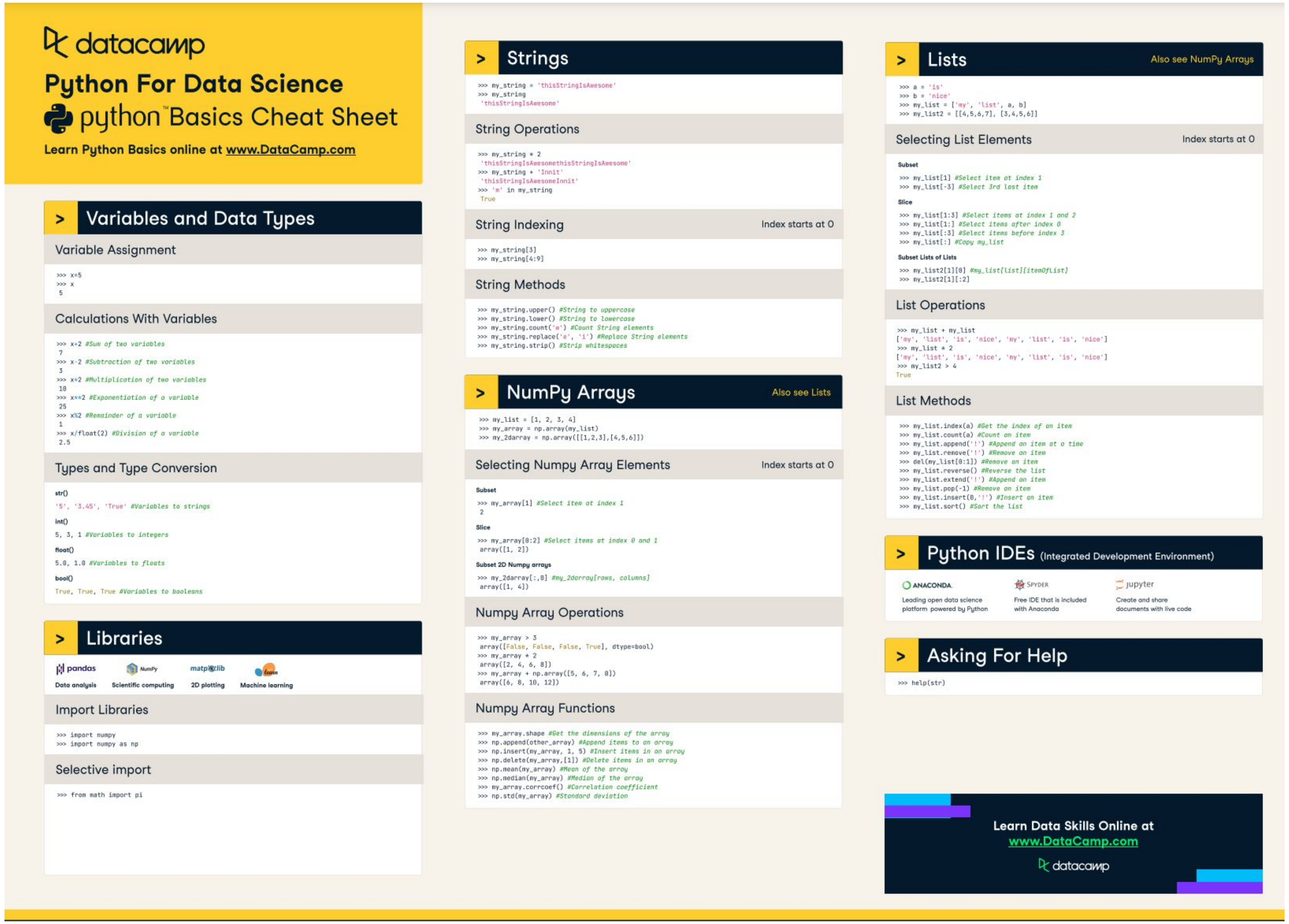
SymPy help topics

About Jupyter Notebook

Learn Data Skills Online at www.DataCamp.com

Python basics cheat sheet

Link



datacamp

Python For Data Science

python Basics Cheat Sheet

Learn Python Basics online at www.DataCamp.com

> Variables and Data Types

Variable Assignment

```
>>> x=5
>>> x
 5
```

Calculations With Variables

```
>>> x+2 #Sum of two variables
 7
>>> x-2 #Subtraction of two variables
 3
>>> x*2 #Multiplication of two variables
 10
>>> x**2 #Exponentiation of a variable
 25
>>> x%2 #Remainder of a variable
 1
>>> x/float(2) #Division of a variable
 2.5
```

Types and Type Conversion

str()

```
'5', '3.45', 'True' #Variables to strings
```

int()

```
5, 3, 1 #Variables to integers
```

float()

```
5.0, 1.0 #Variables to floats
```

bool()

```
True, True, True #Variables to booleans
```

> Libraries

pandas	NumPy	matplotlib	learn
Data analysis	Scientific computing	2D plotting	Machine learning

Import Libraries

```
>>> import numpy
>>> import numpy as np
```

Selective import

```
>>> from math import pi
```

> Strings

```
>>> my_string = 'thisStringIsAwesome'
>>> my_string
 'thisStringIsAwesome'
```

String Operations

```
>>> my_string * 2
 'thisStringIsAwesomethisStringIsAwesome'
>>> my_string + 'Innit'
 'thisStringIsAwesomeInnit'
>>> 'm' in my_string
 True
```

String Indexing

Index starts at 0

```
>>> my_string[3]
>>> my_string[4:9]
```

String Methods

```
>>> my_string.upper() #String to uppercase
>>> my_string.lower() #String to lowercase
>>> my_string.count('w') #Count String elements
>>> my_string.replace('e', 'i') #Replace String elements
>>> my_string.strip() #Strip whitespaces
```

> NumPy Arrays

Also see Lists

```
>>> my_list = [1, 2, 3, 4]
>>> my_array = np.array(my_list)
>>> my_2darray = np.array([[1,2,3],[4,5,6]])
```

Selecting Numpy Array Elements

Index starts at 0

Subset

```
>>> my_array[1] #Select item at index 1
 2
```

Slice

```
>>> my_array[0:2] #Select items at index 0 and 1
 array([1, 2])
```

Subset 2D Numpy arrays

```
>>> my_2darray[:,0] #my_2darray[rows, columns]
 array([1, 4])
```

Numpy Array Operations

```
>>> my_array > 3
 array([False, False, False, True], dtype=bool)
>>> my_array * 2
 array([2, 4, 6, 8])
>>> my_array + np.array([5, 6, 7, 8])
 array([6, 8, 10, 12])
```

Numpy Array Functions

```
>>> my_array.shape #Get the dimensions of the array
>>> np.append(other_array) #Append items to an array
>>> np.insert(my_array, 1, 5) #Insert items in an array
>>> np.delete(my_array,[1]) #Delete items in an array
>>> np.mean(my_array) #Mean of the array
>>> np.median(my_array) #Median of the array
>>> my_array.corrcoef() #Correlation coefficient
>>> np.std(my_array) #Standard deviation
```

> Lists

Also see NumPy Arrays

```
>>> a = 'is'
>>> b = 'nice'
>>> my_list = ['my', 'list', a, b]
>>> my_list2 = [[4,5,6,7], [3,4,5,6]]
```

Selecting List Elements

Index starts at 0

Subset

```
>>> my_list[1] #Select item at index 1
>>> my_list[-3] #Select 3rd last item
```

Slice

```
>>> my_list[1:3] #Select items at index 1 and 2
>>> my_list[1:] #Select items after index 0
>>> my_list[:3] #Select items before index 3
>>> my_list[:] #Copy my_list
```

Subset Lists of Lists

```
>>> my_list2[1][0] #my_list[list][itemOfList]
>>> my_list2[1][:2]
```

List Operations

```
>>> my_list + my_list
['my', 'list', 'is', 'nice', 'my', 'list', 'is', 'nice']
>>> my_list * 2
['my', 'list', 'is', 'nice', 'my', 'list', 'is', 'nice']
>>> my_list2 > 4
True
```

List Methods

```
>>> my_list.index(a) #Get the index of an item
>>> my_list.count(a) #Count an item
>>> my_list.append('!') #Append an item at a time
>>> my_list.remove('!') #Remove an item
>>> del(my_list[0:1]) #Remove an item
>>> my_list.reverse() #Reverse the list
>>> my_list.extend('!') #Append an item
>>> my_list.pop(-1) #Remove an item
>>> my_list.insert(0,'!') #Insert an item
>>> my_list.sort() #Sort the list
```

> Python IDEs (Integrated Development Environment)

ANACONDA	SPYDER	jupyter
Leading open data science platform powered by Python	Free IDE that is included with Anaconda	Create and share documents with live code

> Asking For Help

```
>>> help(str)
```

Learn Data Skills Online at
www.DataCamp.com
datacamp

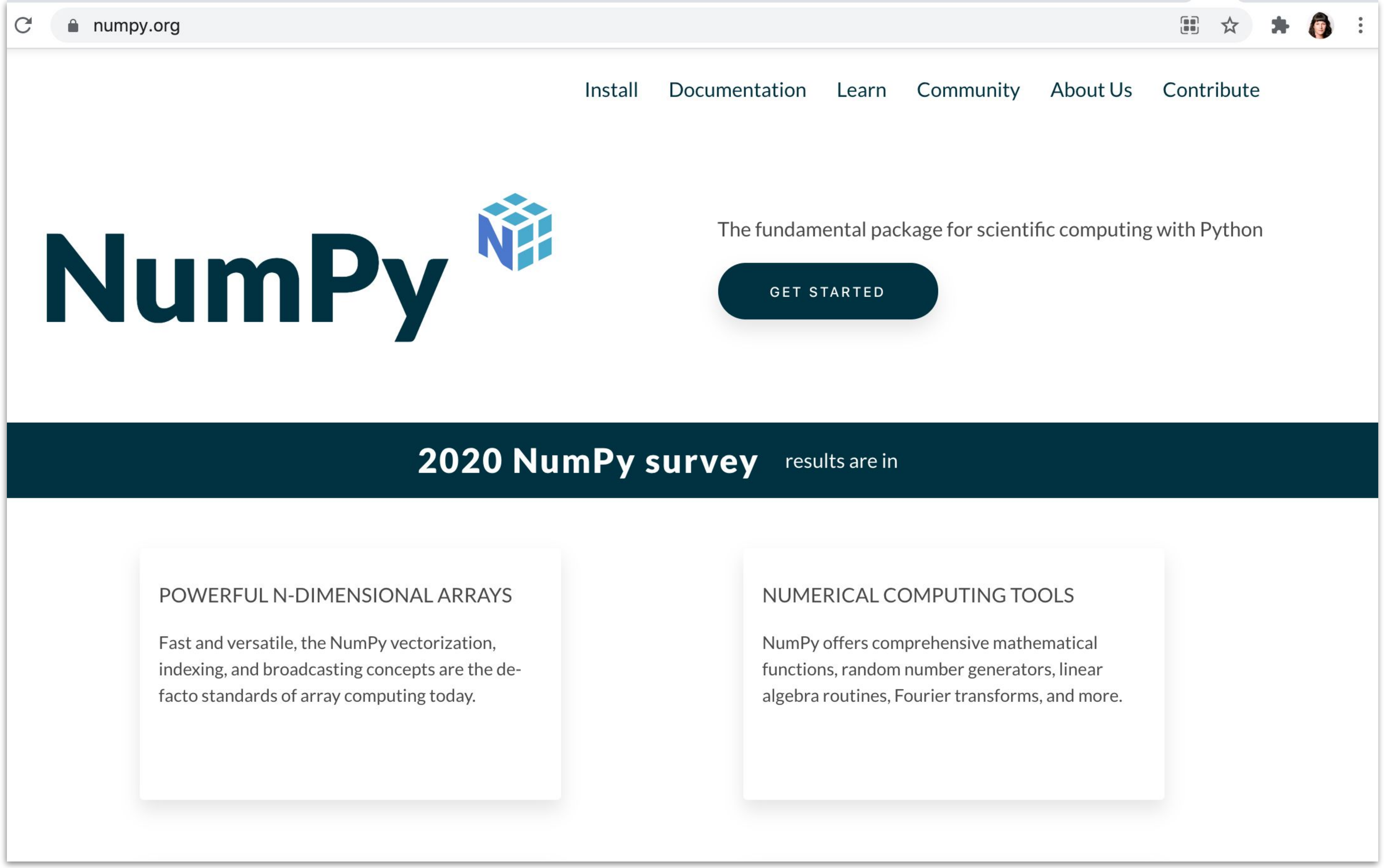
HOMEWORK – NumPy: Our first Python package

Class 6

NumPy

```
import numpy as np
```

- NumPy (short for Numerical Python) is “the fundamental package for scientific computing with Python”
- It is the base library upon which several other libraries which we will use were built
- https://numpy.org/



The fundamental package for scientific computing with Python

2020 NumPy survey results are in

POWERFUL N-DIMENSIONAL ARRAYS

Fast and versatile, the NumPy vectorization, indexing, and broadcasting concepts are the de-facto standards of array computing today.

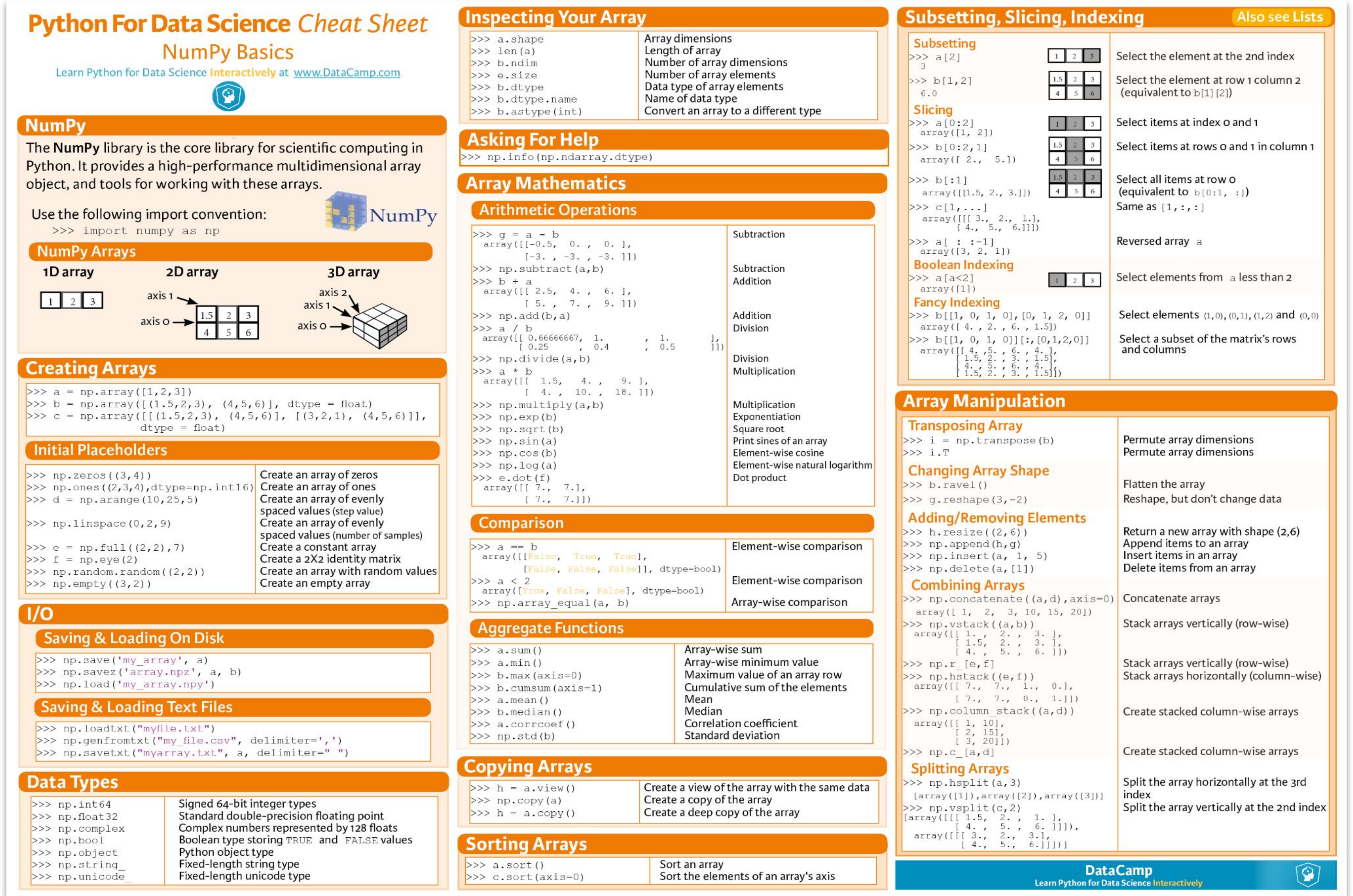
NUMERICAL COMPUTING TOOLS

NumPy offers comprehensive mathematical functions, random number generators, linear algebra routines, Fourier transforms, and more.

NumPy

- Watch this “chooturial” on NumPy
- Do NumPy Jupyter Notebook tutorial

time to learn how to ride a bike, son!
yay!
move to the left! the kinect isn't picking you up!
hold the handle bars better!
invisiblebread.com
more awesome pictures at THEMETAPICTURE.COM



Python For Data Science *Cheat Sheet*

NumPy Basics

Learn Python for Data Science Interactively at www.DataCamp.com

NumPy

The **NumPy** library is the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays.

Use the following import convention:

```
>>> import numpy as np
```

NumPy Arrays

Creating Arrays

```
>>> a = np.array([1,2,3])
>>> b = np.array([(1.5,2,3), (4,5,6)], dtype = float)
>>> c = np.array([[(1.5,2,3), (4,5,6)], [(3,2,1), (4,5,6)]],
                 dtype = float)
```

Initial Placeholders

`>>> np.zeros((3,4))`	Create an array of zeros
`>>> np.ones((2,3,4),dtype=np.int16)`	Create an array of ones
`>>> d = np.arange(10,25,5)`	Create an array of evenly spaced values (step value)
`>>> np.linspace(0,2,9)`	Create an array of evenly spaced values (number of samples)
`>>> e = np.full((2,2),7)`	Create a constant array
`>>> f = np.eye(2)`	Create a 2X2 identity matrix
`>>> np.random.random((2,2))`	Create an array with random values
`>>> np.empty((3,2))`	Create an empty array

I/O

Saving & Loading On Disk

```
>>> np.save('my_array', a)
>>> np.savez('array.npz', a, b)
>>> np.load('my_array.npy')
```

Saving & Loading Text Files

```
>>> np.loadtxt("myfile.txt")
>>> np.genfromtxt("my_file.csv", delimiter=',')
>>> np.savetxt("myarray.txt", a, delimiter=" ")
```

Data Types

`>>> np.int64`	Signed 64-bit integer types
`>>> np.float32`	Standard double-precision floating point
`>>> np.complex`	Complex numbers represented by 128 floats
`>>> np.bool`	Boolean type storing `TRUE` and `FALSE` values
`>>> np.object`	Python object type
`>>> np.string_`	Fixed-length string type
`>>> np.unicode_`	Fixed-length unicode type

Inspecting Your Array

`>>> a.shape`	Array dimensions
`>>> len(a)`	Length of array
`>>> b.ndim`	Number of array dimensions
`>>> e.size`	Number of array elements
`>>> b.dtype`	Data type of array elements
`>>> b.dtype.name`	Name of data type
`>>> b.astype(int)`	Convert an array to a different type

Asking For Help

```
>>> np.info(np.ndarray.dtype)
```

Array Mathematics

Arithmetic Operations

```
>>> g = a - b
  array([[-0.5,  0. ,  0. ],
         [-3. , -3. , -3. ]])
```
Subtraction

```
>>> np.subtract(a,b)
```
Subtraction

```
>>> b + a
  array([[ 2.5,  4. ,  6. ],
         [ 5. ,  7. ,  9. ]])
```
Addition

```
>>> np.add(b,a)
```
Addition

```
>>> a / b
  array([[ 0.66666667, 1.         , 1.         ],
         [ 0.25      , 0.4        , 0.5        ]])
```
Division

```
>>> np.divide(a,b)
```
Division

```
>>> a * b
  array([[  1.5,   4. ,   9. ],
         [  4. ,  10. ,  18. ]])
```
Multiplication

`>>> np.multiply(a,b)`	Multiplication
`>>> np.exp(b)`	Exponentiation
`>>> np.sqrt(b)`	Square root
`>>> np.sin(a)`	Print sines of an array
`>>> np.cos(b)`	Element-wise cosine
`>>> np.log(a)`	Element-wise natural logarithm

```
>>> e.dot(f)
  array([[ 7.,  7.],
         [ 7.,  7.]])
```
Dot product

Comparison

```
>>> a == b
  array([[False,  True,  True],
         [False, False, False]], dtype=bool)
```
Element-wise comparison

```
>>> a < 2
  array([True, False, False], dtype=bool)
```
Element-wise comparison

```
>>> np.array_equal(a, b)
```
Array-wise comparison

Aggregate Functions

`>>> a.sum()`	Array-wise sum
`>>> a.min()`	Array-wise minimum value
`>>> b.max(axis=0)`	Maximum value of an array row
`>>> b.cumsum(axis=1)`	Cumulative sum of the elements
`>>> a.mean()`	Mean
`>>> b.median()`	Median
`>>> a.corrcoef()`	Correlation coefficient
`>>> np.std(b)`	Standard deviation

Copying Arrays

`>>> h = a.view()`	Create a view of the array with the same data
`>>> np.copy(a)`	Create a copy of the array
`>>> h = a.copy()`	Create a deep copy of the array

Sorting Arrays

`>>> a.sort()`	Sort an array
`>>> c.sort(axis=0)`	Sort the elements of an array's axis

Subsetting, Slicing, Indexing

Also see Lists

Subsetting

```
>>> a[2]
 3
```
Select the element at the 2nd index

```
>>> b[1,2]
 6.0
```
Select the element at row 1 column 2 (equivalent to `b[1][2]`)

Slicing

```
>>> a[0:2]
 array([1, 2])
```
Select items at index 0 and 1

```
>>> b[0:2,1]
 array([ 2.,  5.])
```
Select items at rows 0 and 1 in column 1

```
>>> b[:1]
 array([[1.5, 2., 3.]])
```
Select all items at row 0 (equivalent to `b[0:1, :]`)

```
>>> c[1,...]
 array([[[ 3.,  2.,  1.],
         [ 4.,  5.,  6.]]])
```
Same as `[1,:,:]`

```
>>> a[ : :-1]
 array([3, 2, 1])
```
Reversed array `a`

Boolean Indexing

```
>>> a[a<2]
 array([1])
```
Select elements from `a` less than 2

Fancy Indexing

```
>>> b[[1, 0, 1, 0],[0, 1, 2, 0]]
 array([ 4. , 2. , 6. , 1.5])
```
Select elements `(1,0)`, `(0,1)`, `(1,2)` and `(0,0)`

```
>>> b[[1, 0, 1, 0]][:,[0,1,2,0]]
 array([[ 4. ,5. , 6. , 4. ],
        [ 1.5, 2. , 3. , 1.5],
        [ 4. , 5. , 6. , 4. ],
        [ 1.5, 2. , 3. , 1.5]])
```
Select a subset of the matrix’s rows and columns

Array Manipulation

Transposing Array

`>>> i = np.transpose(b)`	Permute array dimensions
`>>> i.T`	Permute array dimensions

Changing Array Shape

`>>> b.ravel()`	Flatten the array
`>>> g.reshape(3,-2)`	Reshape, but don’t change data

Adding/Removing Elements

`>>> h.resize((2,6))`	Return a new array with shape (2,6)
`>>> np.append(h,g)`	Append items to an array
`>>> np.insert(a, 1, 5)`	Insert items in an array
`>>> np.delete(a,[1])`	Delete items from an array

Combining Arrays

```
>>> np.concatenate((a,d),axis=0)
  array([ 1,  2,  3, 10, 15, 20])
```
Concatenate arrays

```
>>> np.vstack((a,b))
 array([[ 1. ,  2. ,  3. ],
        [ 1.5,  2. ,  3. ],
        [ 4. ,  5. ,  6. ]])
```
Stack arrays vertically (row-wise)

```
>>> np.r_[e,f]
```
Stack arrays vertically (row-wise)

```
>>> np.hstack((e,f))
 array([[ 7.,  7.,  1.,  0.],
        [ 7.,  7.,  0.,  1.]])
```
Stack arrays horizontally (column-wise)

```
>>> np.column_stack((a,d))
 array([[ 1, 10],
        [ 2, 15],
        [ 3, 20]])
```
Create stacked column-wise arrays

```
>>> np.c_[a,d]
```
Create stacked column-wise arrays

Splitting Arrays

```
>>> np.hsplit(a,3)
 [array([1]),array([2]),array([3])]
```
Split the array horizontally at the 3rd index

```
>>> np.vsplit(c,2)
[array([[[ 1.5,  2. ,  1. ],
         [ 4. ,  5. ,  6. ]]]),
 array([[[ 3.,  2.,  3.],
         [ 4.,  5.,  6.]]])]
```
Split the array vertically at the 2nd index

DataCamp
Learn Python for Data Science Interactively

[Cheat Sheet Link](#)

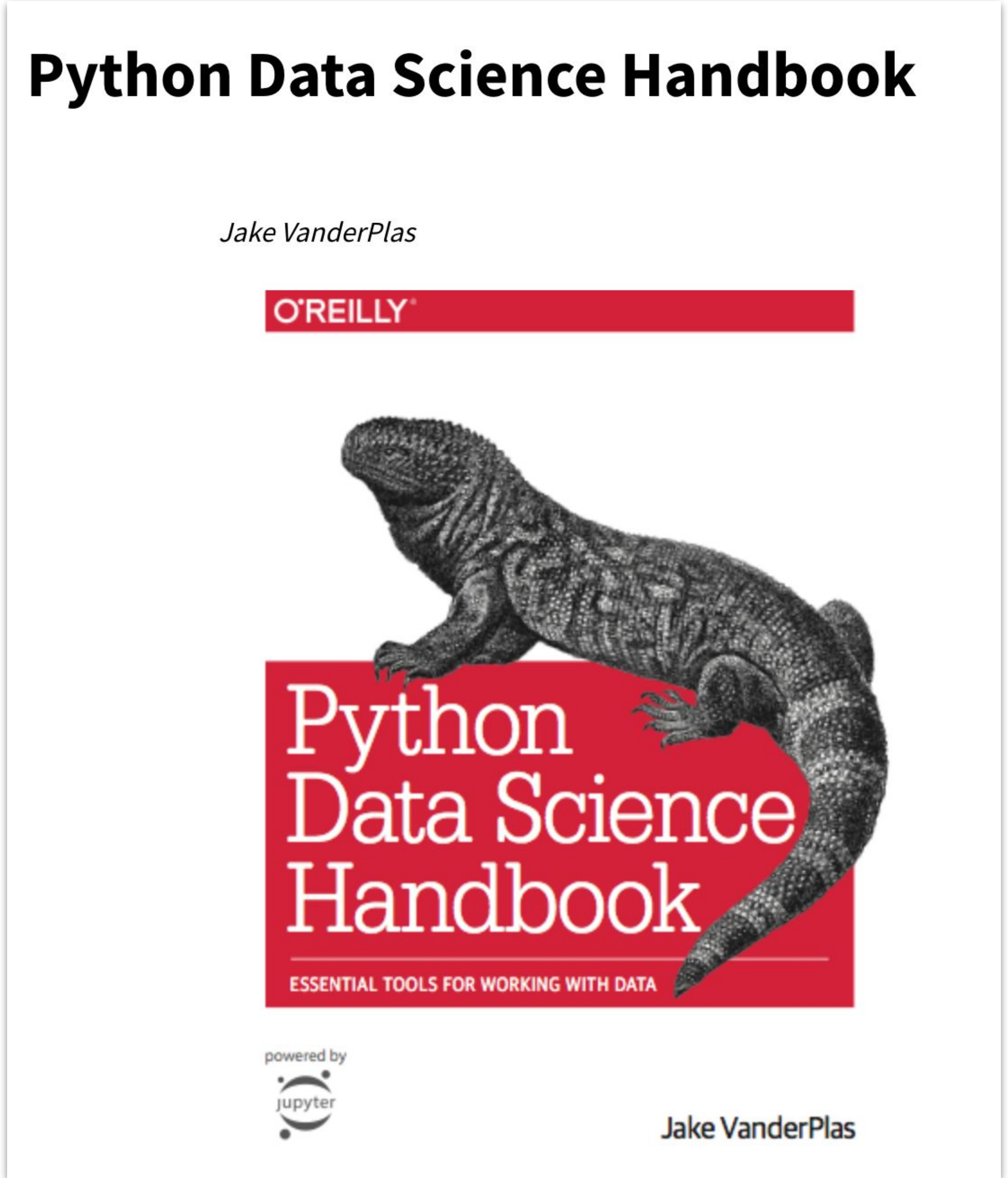
Appendix 17, 18

Bonus & optional notebooks

- If Python is **NOT** new for you and you want to push yourself a little bit more
- We offer extension “Bonus” & Optional notebooks which add a layer of knowledge or complexity
- These are NOT essential
- Just catering to a range of needs and levels.

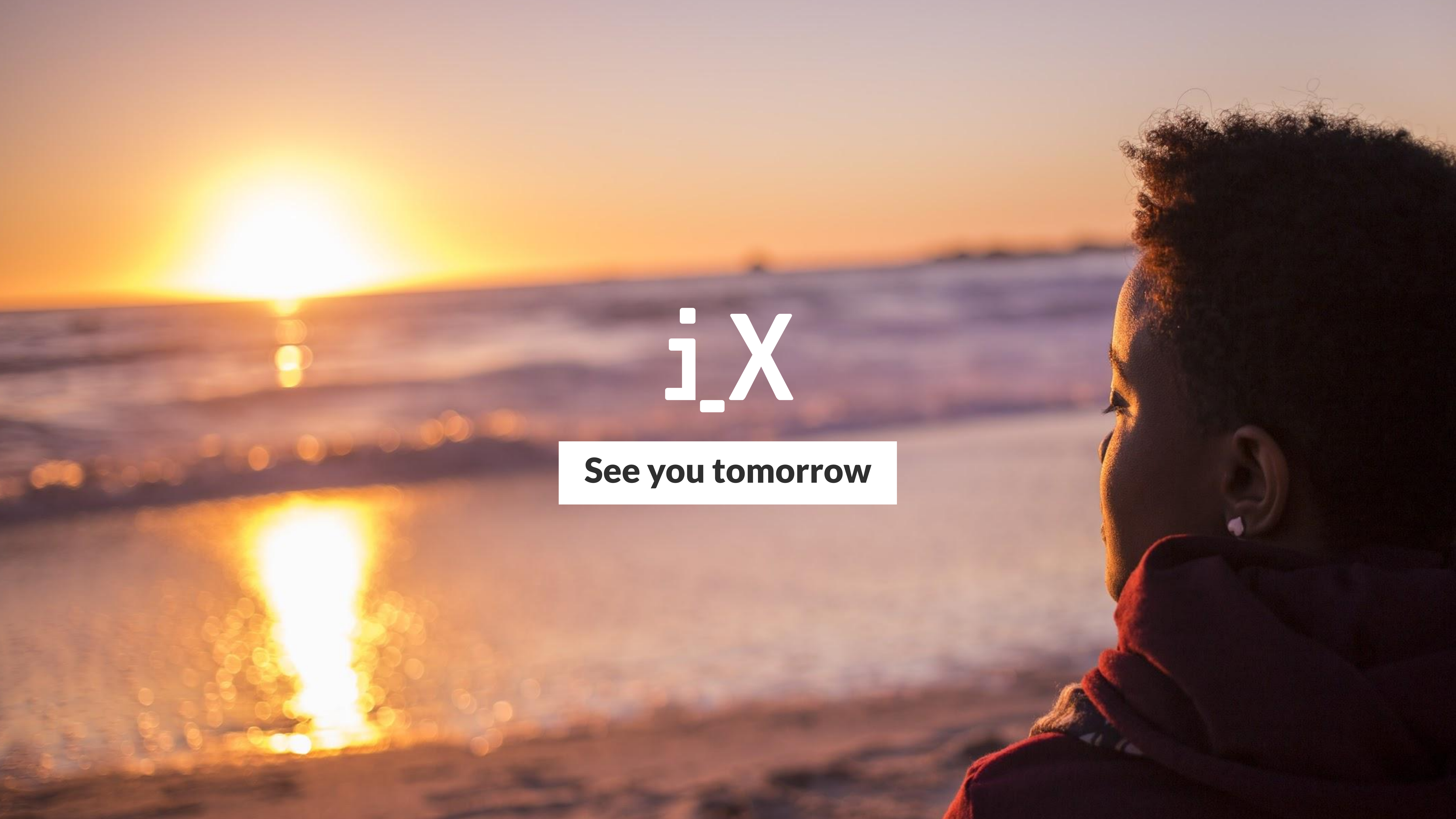
Some extra reading and help if you need …

1. What is Anaconda and Why should I bother about it? - https://medium.com/pankajmathur/what-is-anaconda-and-why-should-i-bother-about-it-4744915bf3e6
2. Python Basics - https://towardsdatascience.com/python-basics-for-data-science-6a6c987f2755
3. Jupyter Notebook Shortcuts - https://towardsdatascience.com/jypyter-notebook-shortcuts-bf0101a98330
4. Python Data Scient Handbook (Ch 1 – Intro to Python, Ch 2 – Intro to Numpy) - https://jakevdp.github.io/PythonDataScienceHandbook/


Python Data Science Handbook
Jake VanderPlas
O'REILLY
Python Data Science Handbook
ESSENTIAL TOOLS FOR WORKING WITH DATA
powered by jupyter
Jake VanderPlas

Appendix

1. https://www.good.is/infographics/the-world-of-data-we-re-creating-on-the-internet
2. https://www.statista.com/statistics/871513/worldwide-data-created/
3. https://codingwithmax.com/step-by-step-guide-getting-started-data-science/
4. https://dash-gallery.plotly.host/dash-uber-rides-demo/
5. https://m.facebook.com/notes/10158928005273415/
6. https://www.hpe.com/us/en/insights/articles/-how-the-pandemic-is-driving-data-literacy-2006.html
7. Future of Jobs Report 2018, World Economic Forum
8. https://twitter.com/timothy_hughes/status/619075227021090817
9. https://www.nobledesktop.com/learn/python/python-vs-excel
10. https://codingwithmax.com/
11. https://www.kdnuggets.com/2018/12/most-popular-python-ide-editor.html
12. https://miteshparmar1.medium.com/structure-your-code-better-in-google-colab-with-text-and-code-cells-b6fa73feec20
13. https://www.datacamp.com/community/data-science-cheatsheets?page=3&tag=python
14. https://docs.anaconda.com/_downloads/9ee215ff15fde24bf01791d719084950/Anaconda-Starter-Guide.pdf
15. http://datacamp-community-prod.s3.amazonaws.com/21fdc814-3f08-4aa9-90fa-247eedefd655
16. http://datacamp-community-prod.s3.amazonaws.com/0eff0330-e87d-4c34-88d5-73e80cb955f2
17. https://www.youtube.com/watch?v=xECXZ3tyONo
18. http://datacamp-community-prod.s3.amazonaws.com/0eff0330-e87d-4c34-88d5-73e80cb955f2



See you tomorrow