Data Science with Python

MSIS 2802 / IDIS 3802

Class hours: Sa 8:30AM - 11:15AM // 12:00PM - 2:45PM

Lucas Hall 309

About me

- **Denis Vrdoljak**
- Education
  - Master of Information and Data Science, UC Berkeley.
    - Master of International Affairs, Texas A&M
- Industry Experience
  - **Cisco**, Data Scientist
  - **Berkeley Data Science Group**, managing Director
  - **SanDisk** Engineering Project Manager, Data Scientist
- Academic Experience
  - Data Science Lecturer at UC Berkeley: 2016-2017
  - Data Science & Python Lecturer at SCU: since 2018.
- Hobbies
  - Dones, UAV's
  - Sailing, Photography, Guitar

MSIS 2802 – Data Science with Python

- Denis Vrdoljak (instructor)
  - Office hours: Wed. 6:00 to 7:00 PM (Lucas Hall) or after class in classroom.
  - Email: dvrdoljak@scu.edu
  - Role: teaching, grading

- Robinson Lu
  - Robinson Lu: blu3@scu.edu
  - Office hours: by appointment
  - Role: help with discussion board, support on software and coding

Outline

- Welcome Survey
  - https://forms.gle/167pLRWecKxKpDG79
- Download and install Anaconda3-5.2.0(**Python 3.6 Version**)
  - Download Anaconda Distribution: https://repo.anaconda.com/archive/
  - Use **Anaconda 3-5.2.0 version (with Python 3.6)** ONLY:
  - Direct link for Mac: Anaconda3-5.2.0-MacOSX-x86_64.pkg
  - Direct link for Windows: Anaconda3-5.2.0-Windows-x86_64.exe
    - Note: Latest Anaconda 2018.12 with Python 3.7 will not work for some packages we will install later
- Introduction to this course
- Python review

Do it now

Student Intro

- Self introduction
  - Your name
  - A bit about your background
  - Your major/minor
  - Why you’re taking this class
  - What do you think you’ll be able to help a classmate with (in the scope of this course)

Data Science

- “An interdisciplinary field about scientific processes and systems to extract knowledge or insights from data in various forms, either structured or unstructured” (from Wikipedia)
- “Is a process of analyzing large size data points to get answers on questions related to that data set”

- Interdisciplinary:
  - Statistics
  - Math
  - Operations Research
  - Databases
  - Machine learning
  - Artificial intelligence
  - …
- Data:
  - Structured (relational database, social network)
  - Unstructured (text, video, audio)

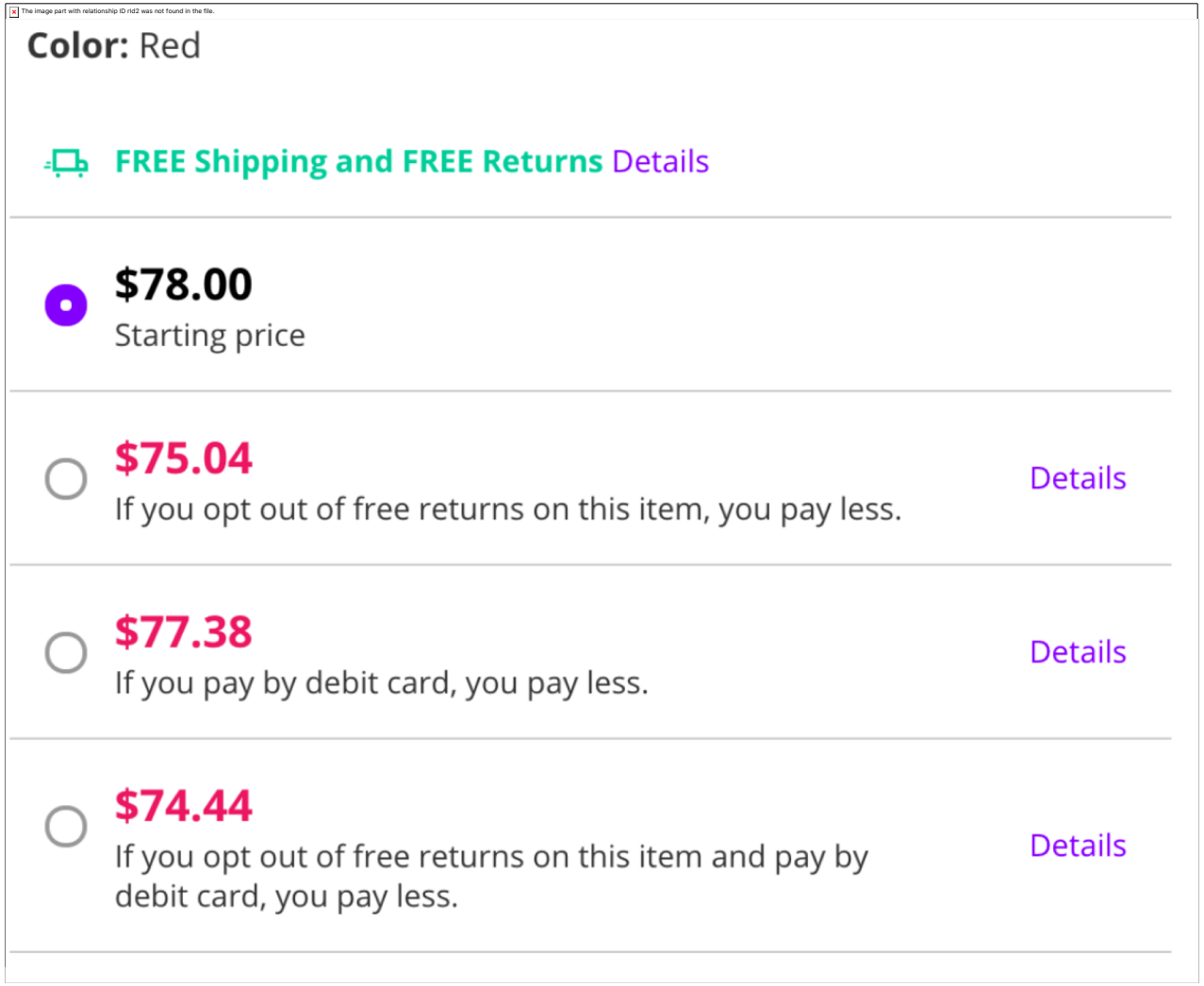
Two flavors of Data Science

We will focus on this

- **Data Mining**: Extract knowledge for decision making through ad-hoc analysis. Examples:
  - Find interesting patterns in the “welcome survey” results

- **Analytics**: Automatically use predictive techniques for optimal decision making. Examples:
  - Amazon and Netflix recommends us items we might like
  - Google and Facebook show us ads that we might be interested in
  - Return policy on jet.com

Jet.com pricing policy



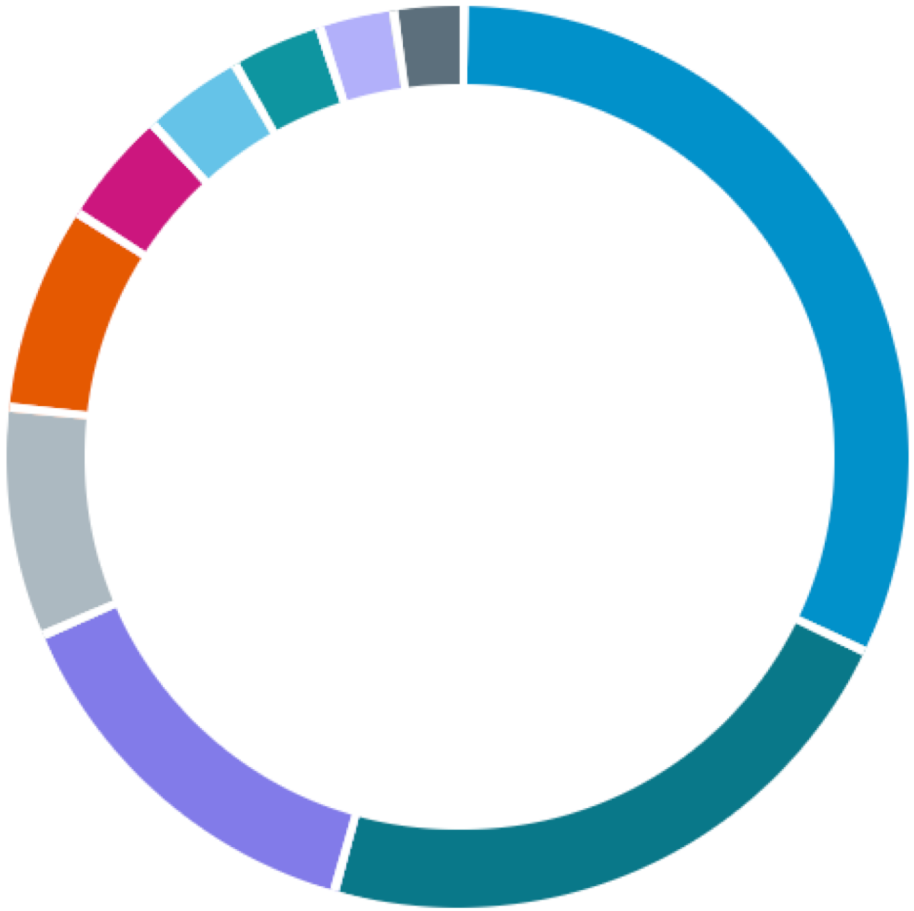
Jet.com Jet.com's Remorse Fee : $5.99 and 5%

Shortage of data scientists

- A McKinsey report finds that *”by 2018, the United States alone could face a shortage of 140,000 to 190,000 people with deep analytical skills as well as 1.5 million managers and analysts with the know-how to use the analysis of big data to make effective decisions.*”

Data Science related jobs

- Data Architect
- Data Scientist
- Data-Visualization Expert
- Data Engineer
- Data Translator
  - By 2026, the McKinsey Global Institute estimates that demand for translators in the United States alone may reach two to four million.



Breakdown of top 10

Top industries

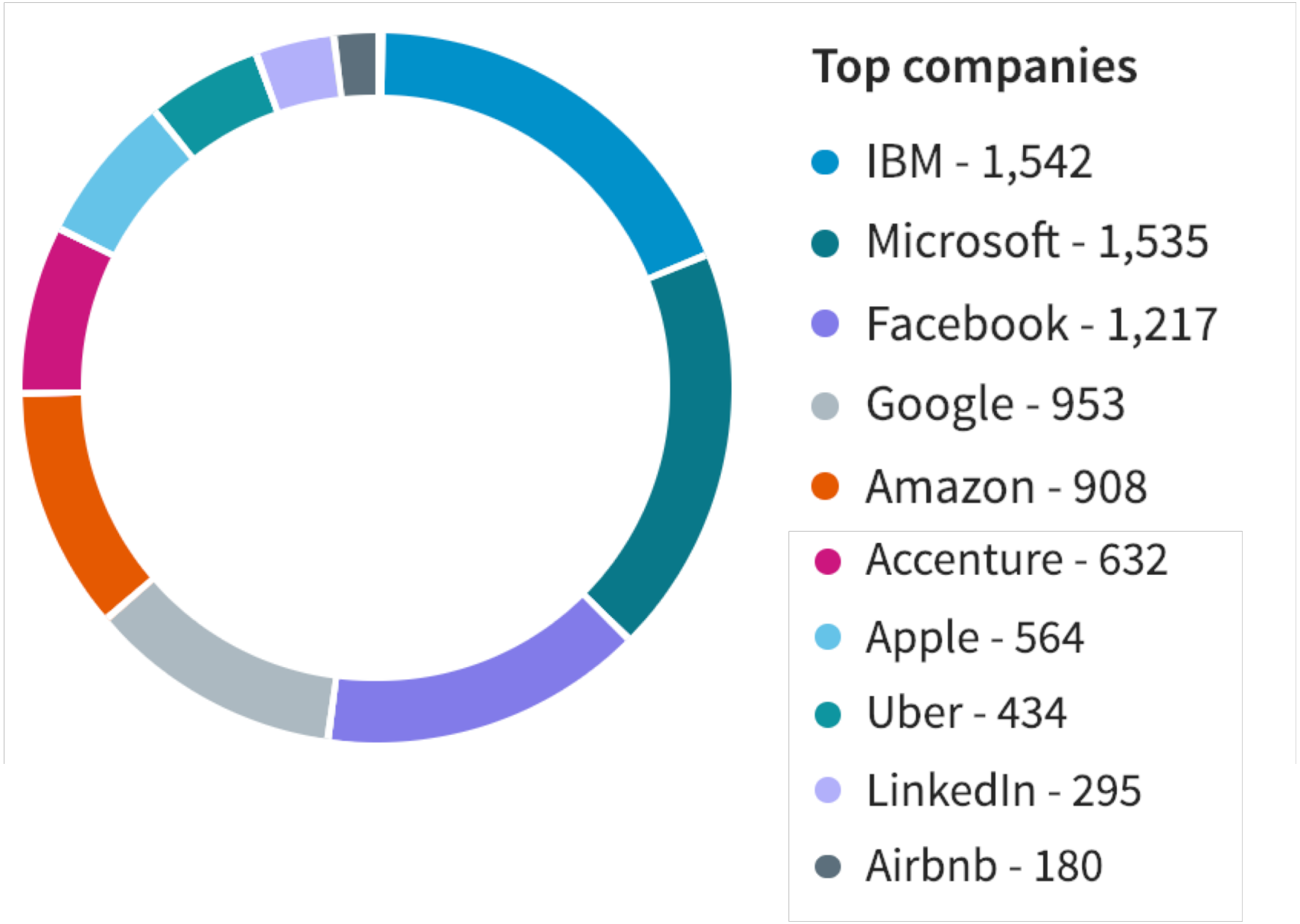
- Information Technology & Services - 31,374
- Computer Software - 21,955
- Research - 13,915
- Internet - 8,091
- Financial Services - 7,218
- Management Consulting - 4,044
- Higher Education - 3,538
- Banking - 3,214
- Marketing & Advertising - 2,633
- Insurance - 2,461

Insights about Data Scientist members on LinkedIn

Top locations

- United States - 53,487
- India - 11,810
- San Francisco Bay Area - 11,625
- United Kingdom - 8,907
- France - 8,703
- Greater New York City Area - 7,323
- Paris Area, France - 5,285
- Canada - 4,858
- Bengaluru Area, India - 4,087
- London, United Kingdom - 3,733

Insights about Data Scientist members on LinkedIn



Insights about Data Scientist members on LinkedIn

This course

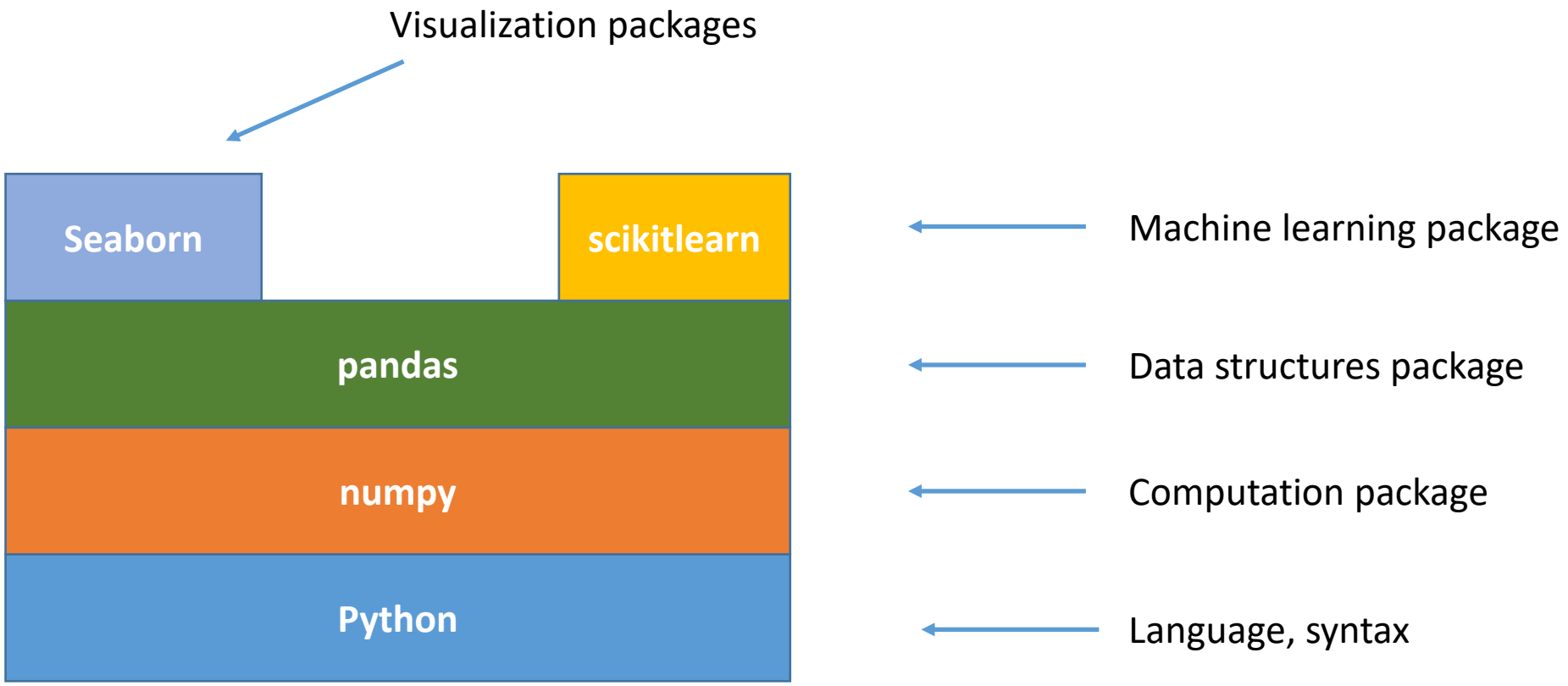
- **Course objectives**
  - To study some fundamental concepts of Data Science
  - To learn the Python packages for Data Science

- **Parts of the course**
  - Part 1: Becoming a data janitor (manipulation, cleaning, aggregating, visualization)
  - Part 2: Becoming a data scientist (some machine learning techniques)

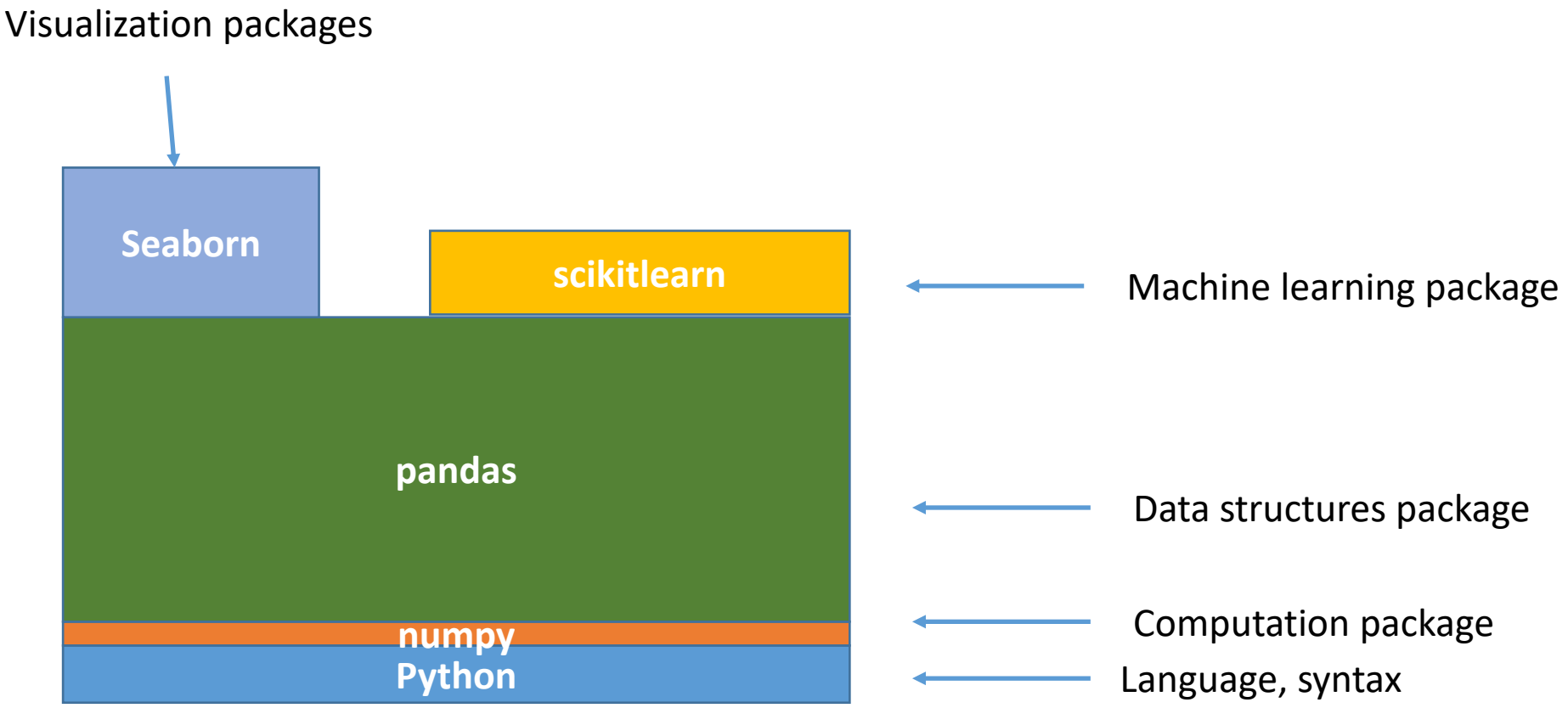
- **If you pass the course, add the following items to your résumé**
  - Python: numpy, pandas, seaborn, scikit-learn
  - Data Science: data wrangling, visualization, classification, regression, clustering
  - Experience with real-world data

- **What we won’t cover**
  - In-depth Python(as a language)
  - Big data
  - Theory of Statistics
  - Theory of Machine Learning

This course


Visualization packages
Seaborn
scikitlearn
pandas
numpy
Python
Machine learning package
Data structures package
Computation package
Language, syntax

This course*



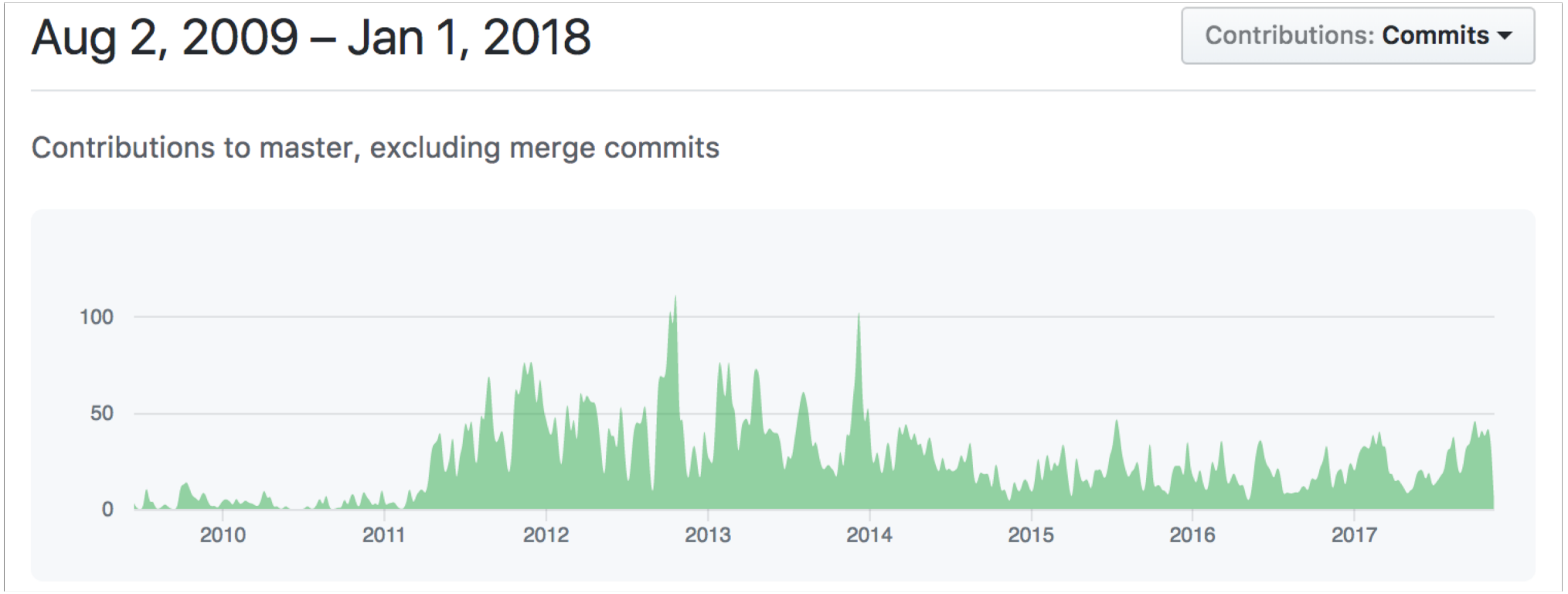
* The height of the boxes is scaled according to the expected amount of time spent

What is the "*pandas*"?

- *pandas* is the a Python module that make Data Science easy and effective.
- *pandas* is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.
- The *'pandas*' name itself is derived from ***panel data,*** an econometrics term for multidimensional structured datasets. Also a play on the phrase '*Python data analysis*'

Pandas History

In 2008, *pandas* development began at AQR Capital Management. By the end of 2009 it had been open sourced in github.



How this course will feel like

- Lots of coding (not a typical language learning coding)
- Hands-on
- Practice in and out of class

How the course looks like – pandas

In [18]:
```
sdata = {'Ohio': 35000, 'Texas': 71000, 'Oregon': 16000, 'Utah': 5000}
obj3 = Series(sdata)
obj3
```

Out[18]:
```
Ohio      35000
Oregon    16000
Texas     71000
Utah       5000
dtype: int64
```

Let us now crate another Series, where the indices are California, Ohio, Oregon, and Texas, but the values are for Ohio, Texas, Oregon, and Utah

In [19]:
```
states = ['California', 'Ohio', 'Oregon', 'Texas']
obj4 = Series(sdata, index=states)
obj4 # the value for Utah is not going to be in the Series, whereas the value of California is NaN
```

Out[19]:
```
California        NaN
Ohio          35000.0
Oregon        16000.0
Texas         71000.0
dtype: float64
```

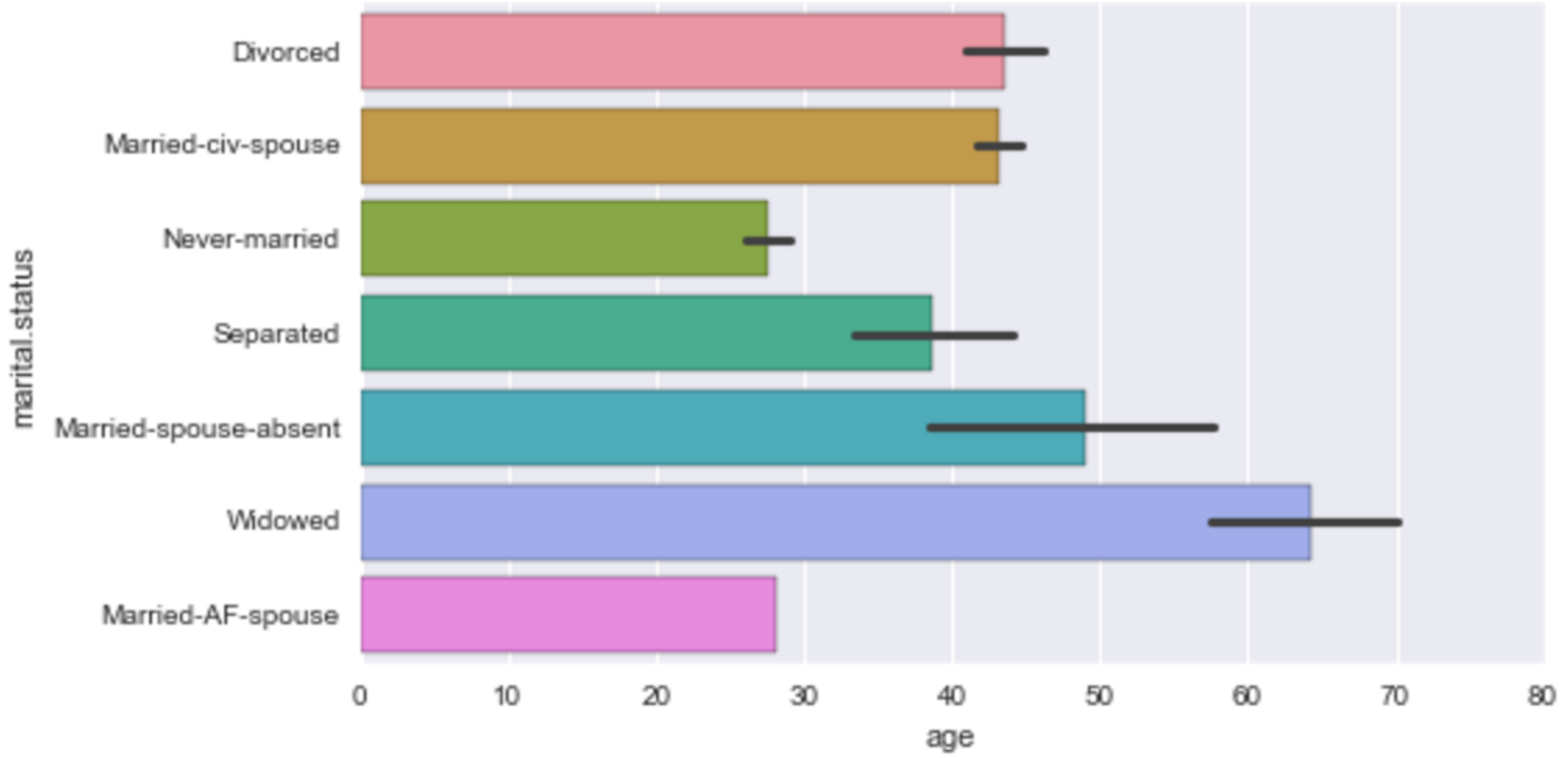
In [20]:
```
obj4.isnull()
```

Out[20]:
```
California     True
```

How the course looks like – seaborn

```
sns.factorplot(x='age',y='marital.status', data=df, kind='bar', aspect = 2)
```

```
<seaborn.axisgrid.FacetGrid at 0x205e9208>
```



How the course looks like – scikit-learn

In [21]:
```
# Import SciKit Learn Log Reg
from sklearn.cross_validation import train_test_split
from sklearn import metrics

# Split the data into Trainging and Testing sets
X_train, X_test, Y_train, Y_test = train_test_split(X, df.bin, test_size=0.4,random_state=3)
```

Support Vector Machines (SVMs)

In [50]:
```
import sklearn.svm as sv
```

In [51]:
```
svm = sv.SVC(probability=True) # probability = True enables the availability of the predict_proba below
```

In [52]:
```
svm.fit(X_train,Y_train)
```

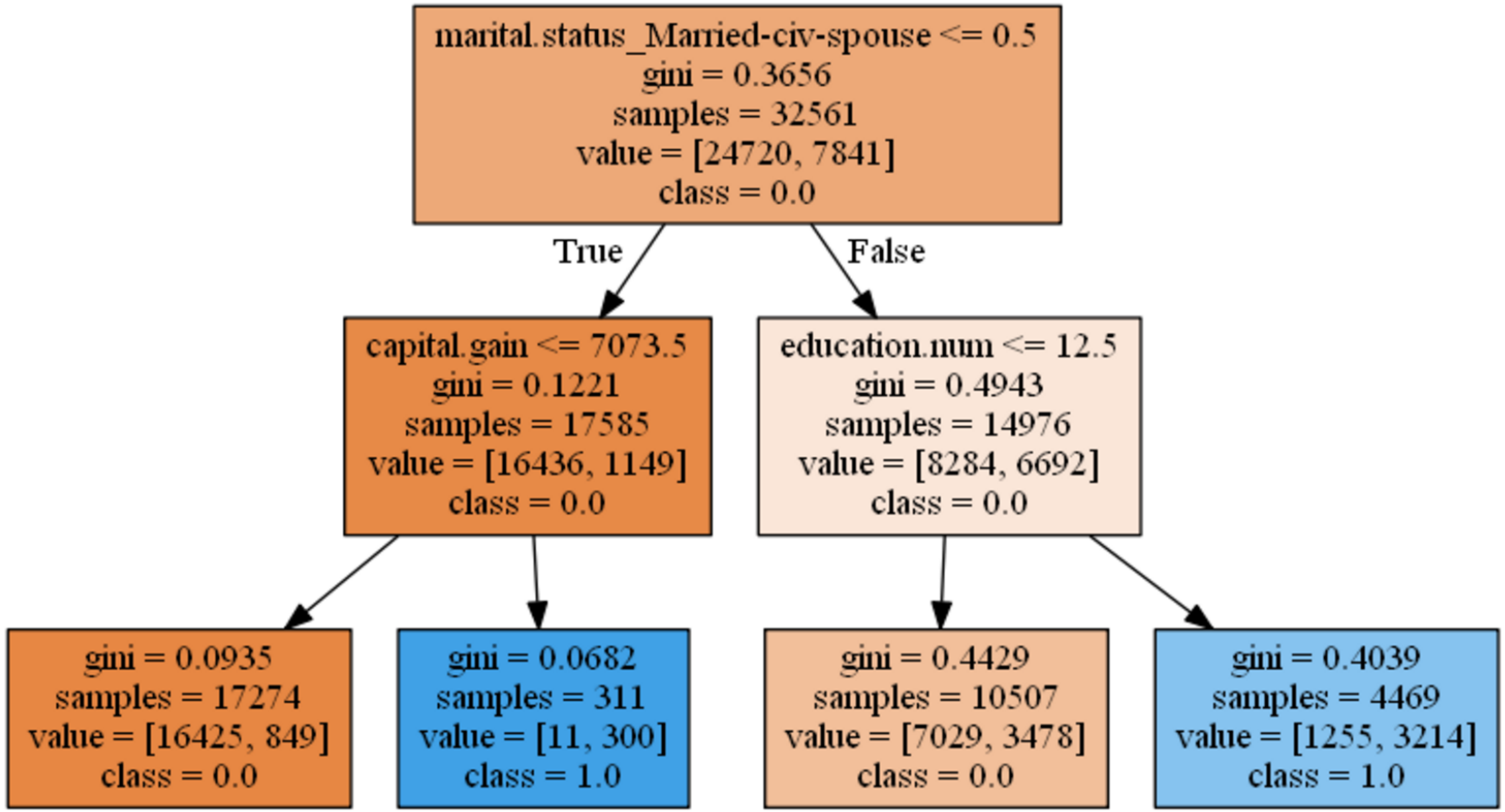
Out[52]:
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape=None, degree=3, gamma='auto', kernel='rbf',
  max_iter=-1, probability=True, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
```

Let's compute the confusion matrix on the test set

In [63]:
```
predY_svm = svm.predict(X_test)
predYProba_svm = svm.predict_proba(X_test)[:,1]
```

In [64]:
```
conf = metrics.confusion_matrix(Y_test, predY_svm, labels=[0,1])
```

How the course looks like – scikit learn



Take a look at syllabus

- Grading
- Quizzes
- Midterm/Final
- Laptop
- Course recordings
- Academic Integrity

Evaluation

- Online Python Tutorial 10%
  - Due in 3 weeks on 1/28
  - Start early!
- Homework 20%
  - Individual assignments, total 5 or 6 HW assignments
- Quizzes 15%
  - total 6 quizzes. In-class, time: TBD
- Midterm Exam 20%
  - On week 6 (2/13) during class time
- Group Project and presentation 15%
  - Groups of 2 people
  - The last two lectures will be dedicated to project presentations
  - Presentations are mandatory for everyone
- Final Exam 20%
  - On final week(3/18 – 3/22) during class time

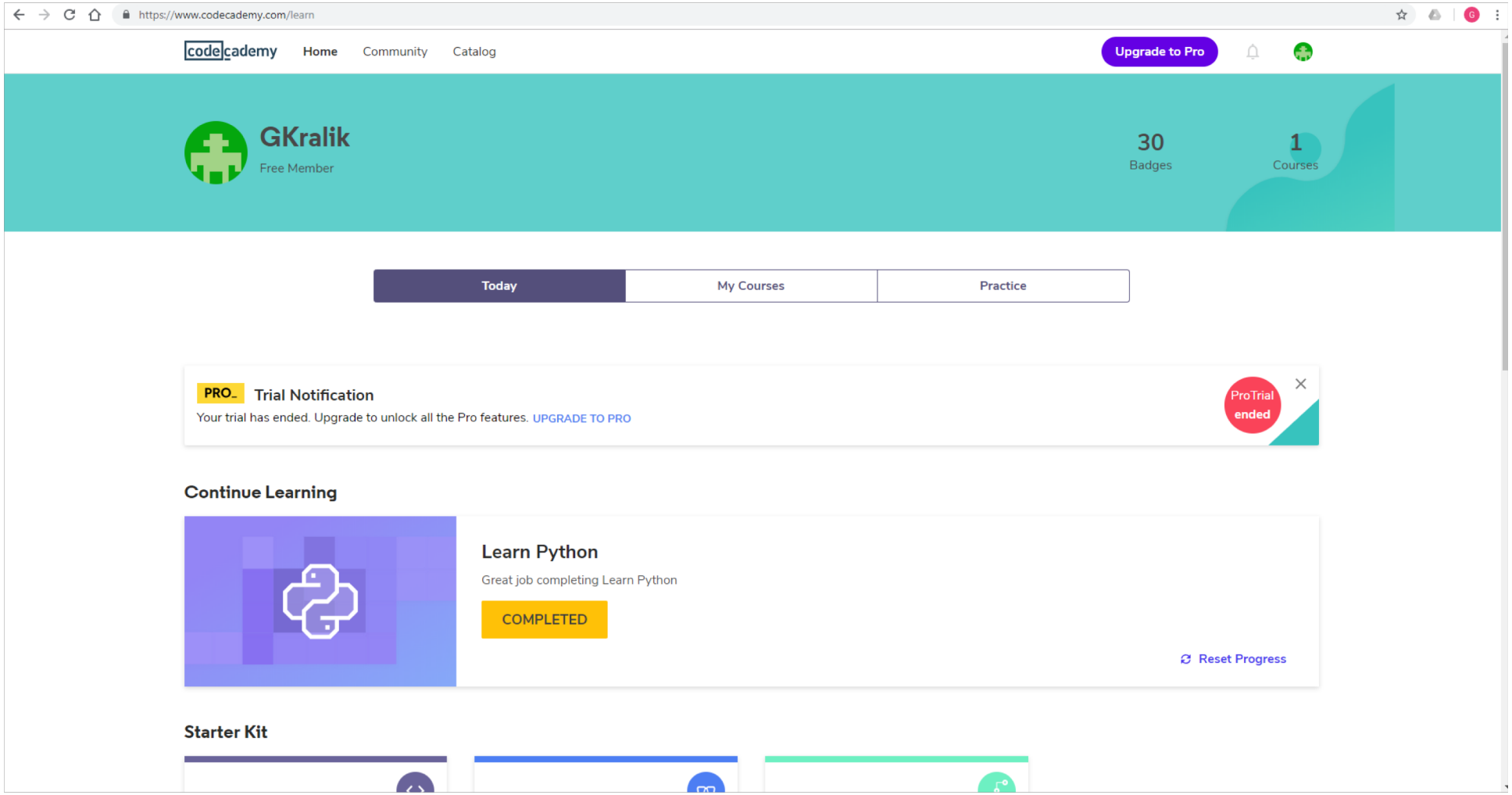
Submission for Completing Codecademy Assignment

This assignment will represent your completion of the following Python online tutorial:

https://www.codecademy.com/learn/python

The tutorial is free. No need to purchase the ‘pro’ version. Submit the final screenshot of course completion which include your user name.

Submit screenshot something like below :



Course policies

- Use the discussion board for the following issues:
  - Questions on homework and class material
    - **Please use discussion board to ask question.**
    - I only give help publicly on the discussion board.
    - In this way, I can give the same help to everyone
  - Feel free to post your code as long as it does not reveal the solution
- Email me for the following issues:
  - Questions on your grades
  - Personal situations
- I will assume that everyone reads the discussion board
- I will grade what you submit

Some more policies

- No make-up in-class Quizzes/Midterm/Final
- No late homework accepted
- If you think that I made a mistake grading your hws/tests, you'll have one week from the day I post the grades to send me an appeal via email. I will look at the appeals and correct the grades if needed. Those grades will be final.

Install Anaconda Distribution

- Download Anaconda Distribution: https://repo.anaconda.com/archive/
- Use **Anaconda 3-5.2.0 version(with Python 3.6)** ONLY:
- Direct link for Mac：Anaconda3-5.2.0-MacOSX-x86_64.pkg
- Direct link for Windows: Anaconda3-5.2.0-Windows-x86_64.exe

Note: Latest Anaconda 2018.12 with Python 3.7 will not work for some packages we will install later.

- You can also download the Anaconda Cheat Sheet for a quick guide to using Anaconda.
- Here is the Anaconda FAQ

What is Anaconda Distribution ?

- Anaconda Distribution is an open source, easy-to-install high performance Python and R distribution, with the conda package and environment manager and collection of 1,000+ open source packages with free community support.
- Anaconda Distribution contains **conda** and **Anaconda Navigator**

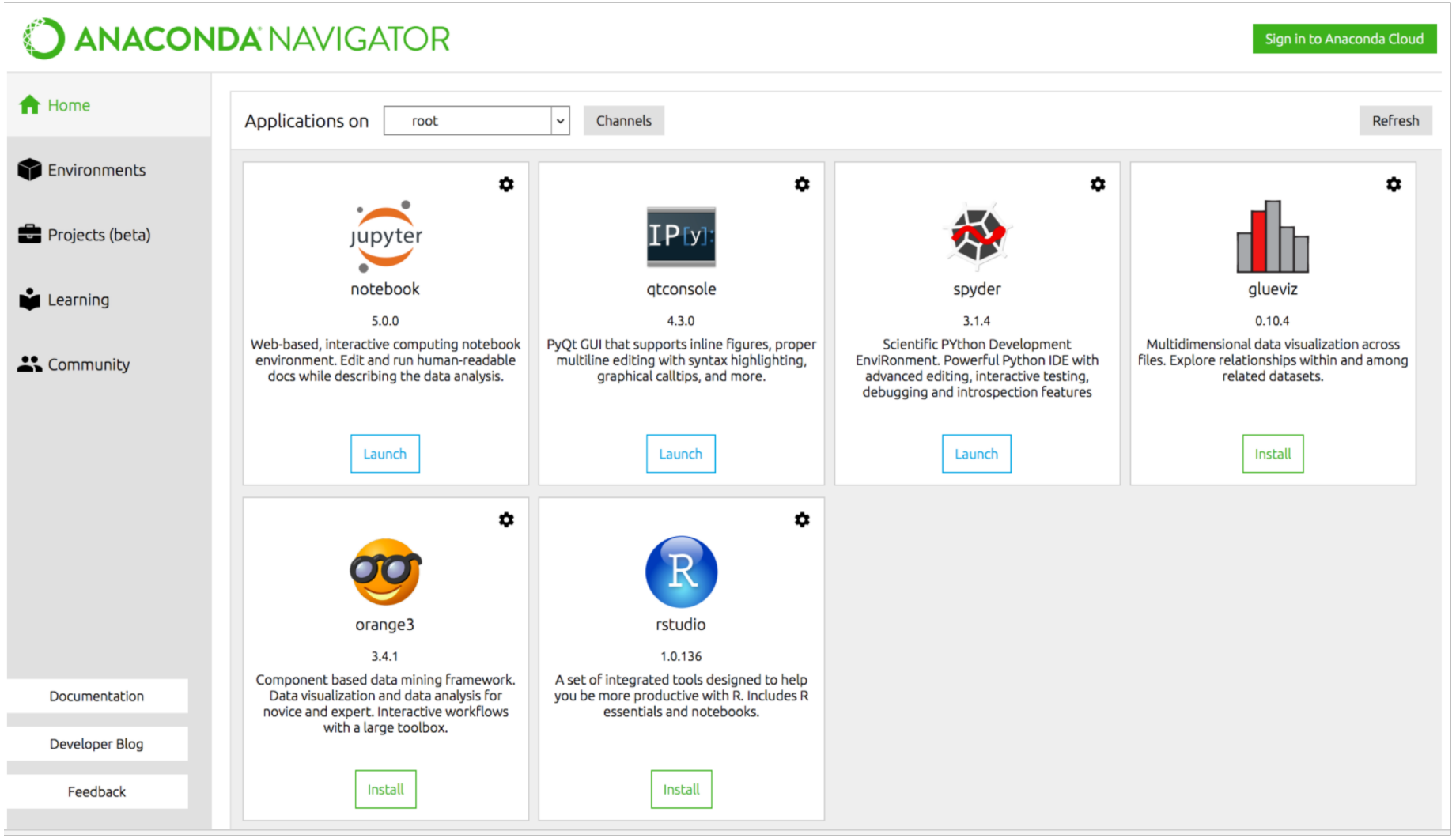
What is *Conda*?

- *Conda* is a package manager and environment management system that installs, runs, and updates packages and their dependencies.
- *Conda* allows users to easily
  - install different versions of binary software packages and any required libraries appropriate for their computing platform
  - switch between package versions
  - download and install updates from a software repository.

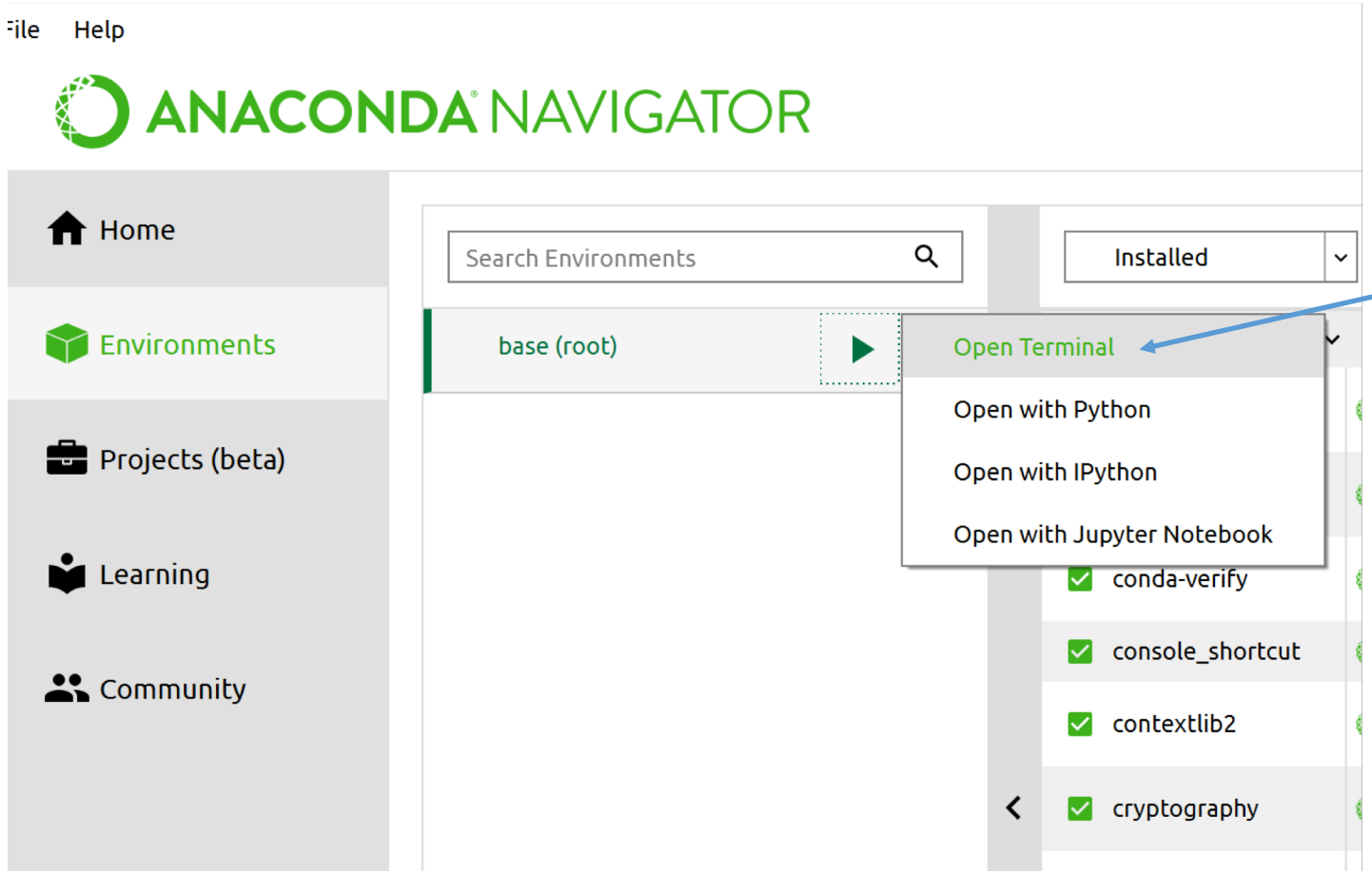
What is **Anaconda Navigator**?

- Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® distribution
- Allows you to launch applications and easily manage conda packages, environments and channels without using command-line commands.
- Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository. It is available for Windows, macOS and Linux.

What is Anaconda Navigator? (cont)

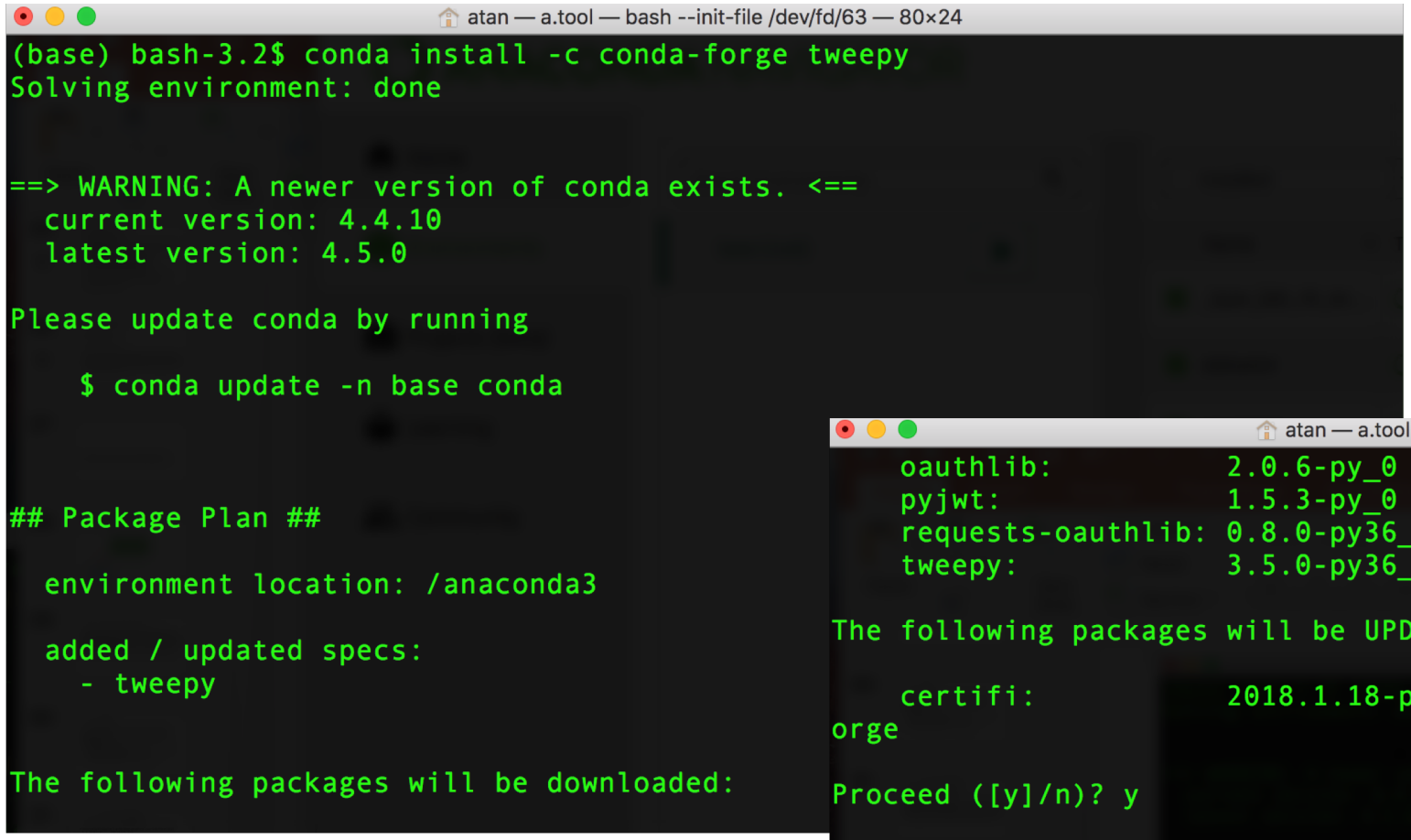


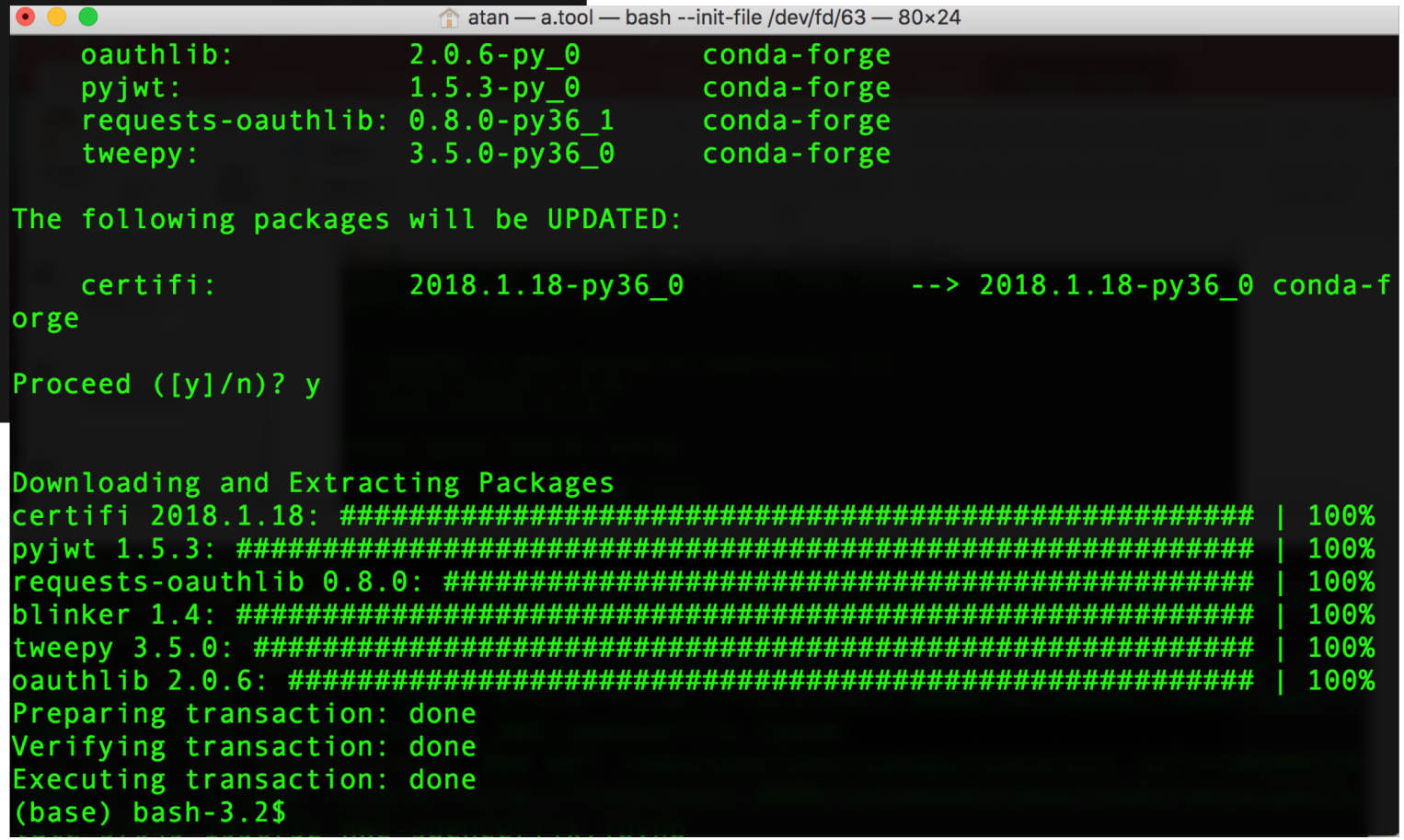
Conda – install your first package, Twitter package ***tweepy***



1, Open a terminal

2, in terminal type > ***conda install -c conda-forge tweepy***


atan — a.tool — bash --init-file /dev/fd/63 — 80×24
(base) bash-3.2$ conda install -c conda-forge tweepy
Solving environment: done
==> WARNING: A newer version of conda exists. <==
current version: 4.4.10
latest version: 4.5.0
Please update conda by running
$ conda update -n base conda
Package Plan
environment location: /anaconda3
added / updated specs:
- tweepy
The following packages will be downloaded:

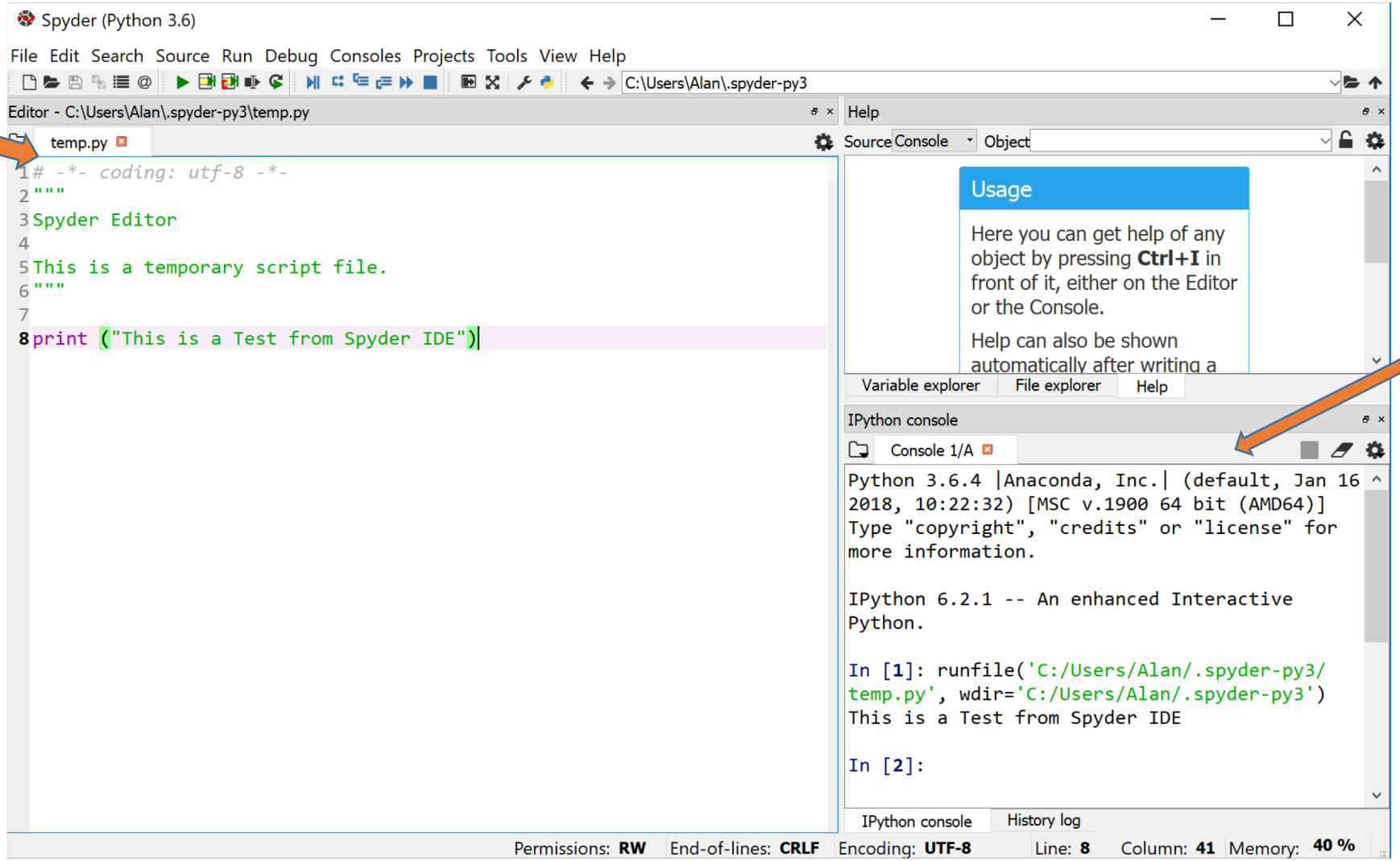

atan — a.tool — bash --init-file /dev/fd/63 — 80×24
oauthlib: 2.0.6-py_0 conda-forge
pyjwt: 1.5.3-py_0 conda-forge
requests-oauthlib: 0.8.0-py36_1 conda-forge
tweepy: 3.5.0-py36_0 conda-forge
The following packages will be UPDATED:
certifi: 2018.1.18-py36_0 --> 2018.1.18-py36_0 conda-forge
Proceed ([y]/n)? y
Downloading and Extracting Packages
certifi 2018.1.18: ########## | 100%
pyjwt 1.5.3: ########## | 100%
requests-oauthlib 0.8.0: ########## | 100%
blinker 1.4: ########## | 100%
tweepy 3.5.0: ########## | 100%
oauthlib 2.0.6: ########## | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
(base) bash-3.2$

Programming environment used in this course

- After installing Anaconda, we have access to:
  - **Spyder**: Scientific Python Development Environment
    - Powerful Python IDE to practice
  - **Jupyter notebook**: a web-based interactive computing environment
    - Excellent to package your projects
    - For the HWs/Tests/Projects you will be asked to submit a Jupyter file

Spyder (Python 3.6)

Editor window



Console Window

Jupyter

jupyter Module 8 seaborn - complete Last Checkpoint: 02/25/2018 (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

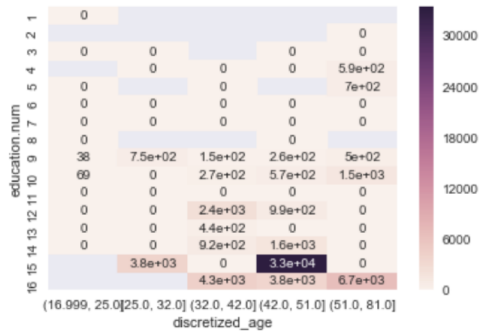
Run Code

Find average capital gain for age and education.num

In [30]:
```
gr = df2.groupby(['education.num','discretized_age'])['capital.gain'].mean()
```

In [31]:
```
sns.heatmap(gr.unstack(),annot=True)
```

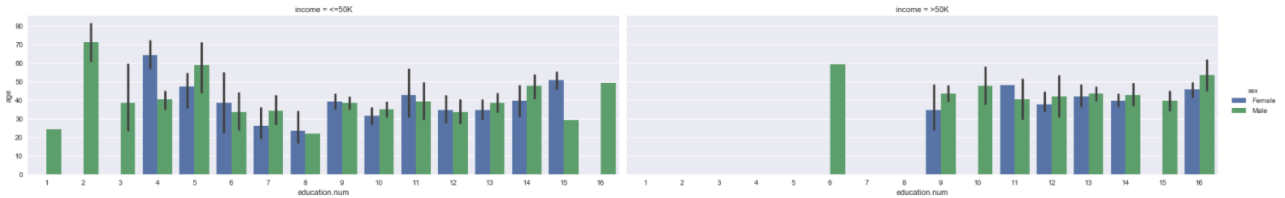
Out[31]: <matplotlib.axes._subplots.AxesSubplot at 0x11d07d7d0>



Four variables: two numeric and two categorical

In [32]:
```
sns.factorplot(x='education.num',y='age',hue='sex',col='income', data=df,kind='bar',aspect=3)
```

Out[32]: <seaborn.axisgrid.FacetGrid at 0x11d09d790>



Four variables: one numeric and three categorical

In [33]:
```
sns.factorplot(x='marital.status',y='age',hue='sex',col='income', data=df,kind='bar',aspect=3)
```

The Jupyter Notebook

Open Jupyter notebook

The Jupyter Notebook

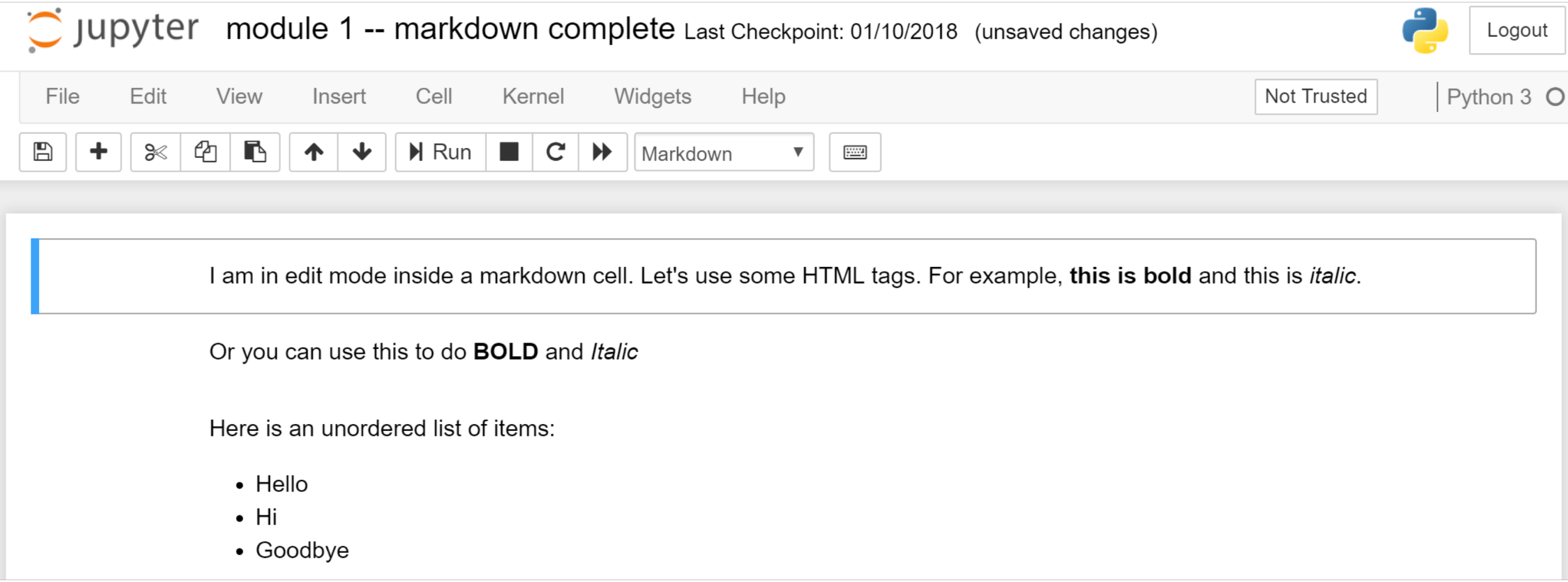
- Jupyter notebook, formerly known as the IPython notebook, is a flexible tool that helps you create readable analyses, as you can keep code, images, comments(formatted (html) text), formula and plots together.
- The name Jupyter is an indirect acronym of the three core languages it was designed for: **JU**lia, **PYT**hon, and **R** and is inspired by the planet Jupiter.

The Jupyter Notebook – cont.

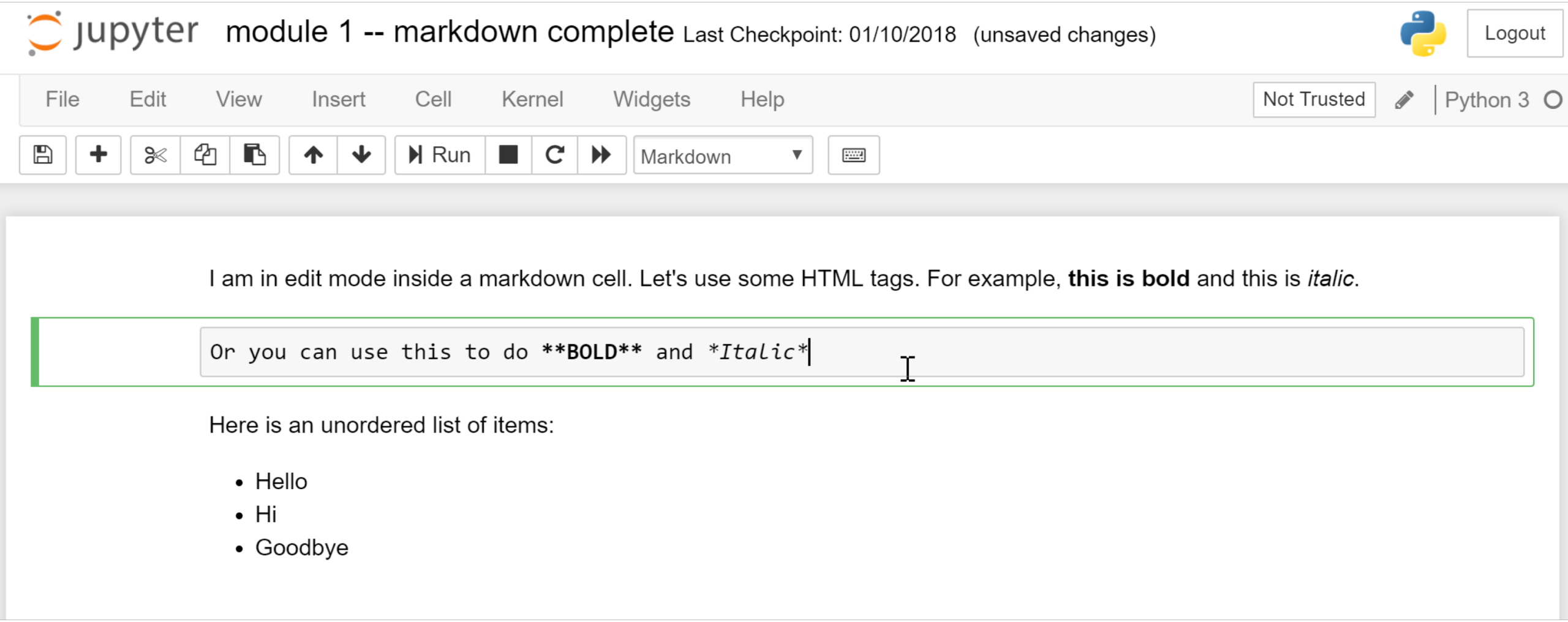
- Each cell can contain code or text (called **markdown**)
- Each cell can be executed with SHIFT+ENTER or CTRL+ENTER
- Executing a “code cell” prints the result
- Executing a “markdown cell” formats it and displays it

- Two modes:
  - Command mode
  - Edit mode
  - Esc will take you into command mode
  - Enter will take you into edit mode on current cell

Command mode



Edit mode


jupyter module 1 -- markdown complete Last Checkpoint: 01/10/2018 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown
I am in edit mode inside a markdown cell. Let's use some HTML tags. For example, this is bold and this is italic.
Or you can use this to do **BOLD** and *Italic*
Here is an unordered list of items:
Hello
Hi
Goodbye

Command mode

Press ESC to switch to command mode

Shortcut	What it does
Cmd Shift P	Command palette
a	Insert a cell above
b	Insert a cell below
Enter	Switches to edit mode inside the current cell
Esc	Switch to command mode
m	Changes the cell content to markdown
y	Changes the cell content to code
dd	Deletes the current cell
x,c,v	Cut, copy, paste a cell
z	Undo last cell deletion
Shift Enter	Run cell, select below
Ctrl Enter	Run cell

Command mode – cont.

Press ESC to switch to command mode

Shortcut	What it does
Ctrl Shift -	Split the current cell into two from where your cursor is
Shift Down/Up	selects the next cell in a down/upwards direction.
Shift M	merge multiple cells.

Edit mode

Press Enter to switch to Edit mode

Tags	What it does
	Bold
__String__ or **String**	Bold
	Italic
String or *String*	Italic
	Ordered list
Start with 1. follow by a space	Ordered list
	Unordered list
Start with – (or *) follow by two spaces	Unordered list
	List item
#	Header of level 1 (main header)
##	Header of level 2
…	…

```
I am in edit mode and I am editing a markdown cell. So, I can use HTML tags to format the text. For example, this is <b>bold</b>
and this is <i>italic</i>. Now, I can press SHIFT+ENTER to visualize the formatted cell
```

```
I can make an unordered list with a few items:
<ul>
<li>Hello</li>
<li>Hi</li>
<li>Goodbye</li>
</ul>
```

```
Or an ordered list of items:
<ol>
<li>Hello</li>
<li>Hi</li>
<li>Goodbye</li>
</ol>
```

```
To make a header, use hashtags:
```

Level 1

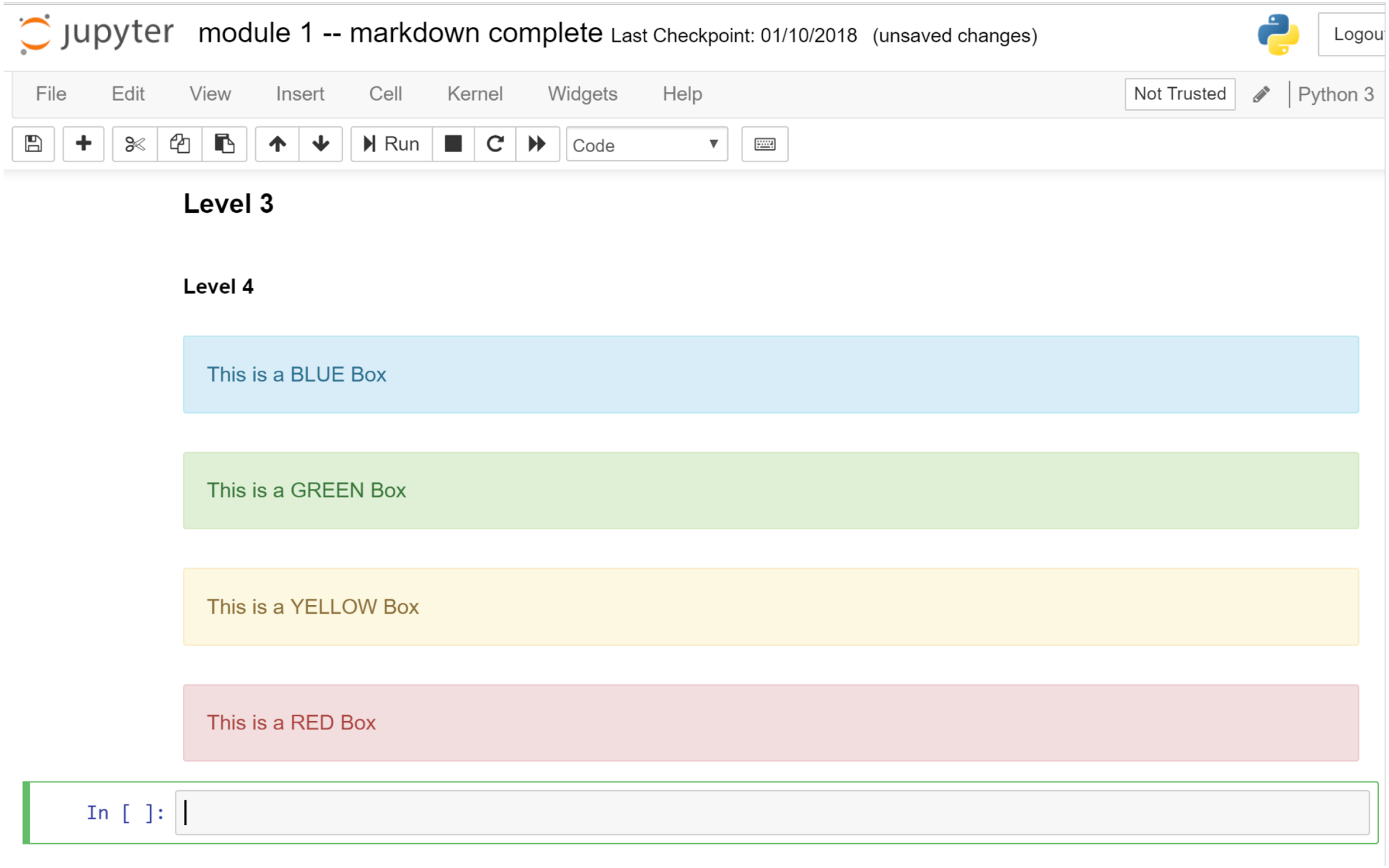
Level 2

Level 3

Level 4

Colored note boxes

- Blue boxes (Tips)
  - string
- Yellow boxes (Examples)
  - …
- Green boxes
  - …
- Red boxes
  - …


jupyter module 1 -- markdown complete Last Checkpoint: 01/10/2018 (unsaved changes)
Logou
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
Level 3
Level 4
This is a BLUE Box
This is a GREEN Box
This is a YELLOW Box
This is a RED Box
In [ ]:

Cheat sheet

- https://www.cheatography.com/weidadeyue/cheat-sheets/jupyter-notebook/
- https://medium.com/ibm-data-science-experience/markdown-for-jupyter-notebooks-cheatsheet-386c05aeebed

Open your first Jupyter file

- Open – 'module 1 -- markdown template.ipynb’

Tech Note: How to change the default working directory of Jupyter Notebook

- Go to Jupypter config directory *'/Users/YourUsername'* (if not sure, type '*jupyter --config-dir')*
- Change directory to *.jupyter* folder (pay attention with the 'dot' before the name) (*example: /Users/ttan/.jupyter*)
- Create(or edit) a file '*jupyter_notebook_config.py*'
- Insert a line

  c.NotebookApp.notebook_dir = '/Your/Path'

  (example: c.NotebookApp.notebook_dir = '/Users/ttan/MSIS2802/'

- Or change this line to

  #c.NotebookApp.notebook_dir = u''

  change it to:

  c.NotebookApp.notebook_dir = '/Your/Path'

Tech Note: How to disable *autosave* in Jupyter

- Method 1:
  - In code cell :
    - %autosave 0
- Method 2:
  - Go to user home directory/.jupyter/custom directory
    - Example: /Users/atan/.jupyter/custom
  - Create *custom.js* file
  - Custom.js file content

```
$([IPython.events]).on("notebook_loaded.Notebook", function () {
  IPython.notebook.set_autosave_interval(0);
});
```

  - Restart Jupyter
  - Use File/Save and Checkpoint to save status

Tech Note: IPython Magic Commands

- *%autosave* is a IPython Magic Commands
- Being based on the IPython kernel, Jupyter has access to all the Magics from the IPython kernel
- This will list all magic commands
  - *%lsmagic*
- Examples*:*
  - *%env*: Set Environment Variables
  - *%run*: Execute python code
  - *%%time* will give you information about a single run of the code in your cell.
  - *%%timeit* uses the Python timeit module which runs a statement 100,000 times (by default) and then provides the mean of the fastest three times.
    (one ”%” is line magic. Two “%%” is cell magic)

Tech Note: How to see the value of multiple statements at once

- Method 1:
  - In code cell :

    from IPython.core.interactiveshell import InteractiveShell
    InteractiveShell.ast_node_interactivity = "all"

- Method 2:
  - Go to user home directory, create a file ~/.ipython/profile_default/ipython_config.py with the lines below:

    c = get_config()

    # Run all nodes interactively
    c.InteractiveShell.ast_node_interactivity = "all"

  - Restart Jupyter

Tech Note: Few Best Practices in Jupyter

- After open an original file, make a copy and work on the copy
  - File -> Make a Copy
- Save your work
  - File -> Save and Checkpoint
- Rename your Jupyter file if needed
  - File -> Rename
- Close a Jupyter file
  - File -> Close and Halt
- Sometime need to start the Jupyter Notebook file
  - Kernel -> Restart and Run All (automatic run all cells)
  - In cmd mode, press “00” (Restart the current kernel, need manually run each cell)

Python Review

Open Jupyter notebook

In a nutshell

- High-level programming language
- Interpreted
- Dynamically typed
- Philosophy: code should be minimal

Open
module 1 -- Python review template.ipynb