End-to-End Data Science Workflows in Jupyter Notebooks

About Your Instructor

- Jamie Whitacre
- Worked for Project Jupyter at the UC Berkeley Institute for Data Science
- Currently teaching data analysis and data science

Poll – Which Operating System Are You Using Today?

- MacOS
- Windows
- Linux

Poll – What is Your Experience Level with Jupyter?

1. **Complete Newbie**: this is my first time installing and seeing Jupyter Notebooks
2. **Some Experience**: I've used Jupyter Notebooks created by others but have not created my own Notebook
3. **Advanced Beginner**: I've created a few of my own data analyses
4. **Intermediate**: I've been using Jupyter Notebooks regularly
5. **Advanced User**: I use Jupyter Notebooks every day for complex analyses and/or set up multi-user Jupyter environments

Materials for Today

1. Anaconda distribution of Python. This is a software package includes the Jupyter Notebook and other tools we'll use today. If you haven't already, please copy & install the version suitable to your OS (Windows, Mac, Linux). https://www.anaconda.com/download/
2. A public dataset from Data.gov.
3. A sample Jupyter Notebook including source code to guide you through today's session.

Sample Notebook on GitHub

```
git clone \
    https://github.com/JamiesHQ/DS_Workflows_Jupyter5_20180621
```

Which Python Version Do I Install?

Important

This documentation covers IPython versions 6.0 and higher. Beginning with version 6.0, IPython stopped supporting compatibility with Python versions lower than 3.3 including all versions of Python 2.7.

If you are looking for an IPython version compatible with Python 2.7, please use the IPython 5.x LTS release and refer to its documentation (LTS is the long term support release).

Differences between Mac & Windows

- When you install Anaconda on Windows you may need to run “install Jupyter” in the command line.
- Typing “jupyter notebook” should just work on a Mac
- Mac file paths use forward slashes, Windows uses backslashes

Getting Started

-1-
Install Jupyter

-2-
Learn what it is & how to use it

-3-
Complete an end-to-end data analysis workflow using the Jupyter Notebook

Jupyter is great for learning and doing Python!
Basic knowledge of Python is assumed for today's lesson

Accessing Azure Notebooks

https://notebooks.azure.com/

nteract

https://nteract.io/desktop

open notebooks natively on Mac, PC, Linux

Stay Organized

- Create a folder for this webinar that you can easily find
- Save your notebook(s), data files, and any resources to your webinar folder

Project Jupyter & the Jupyter Ecosystem

Jupyter Notebook

jupyter Jupyter_Metrics_CleanExploreData Last Checkpoint: 3 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Markdown CellToolbar

Repository Metrics

Import libraries

```
In [ ]: %matplotlib inline
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

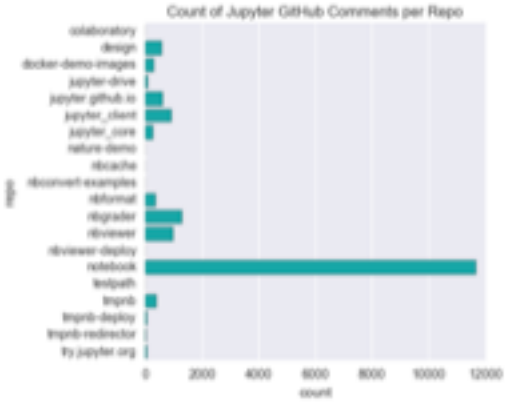
Import data

```
In [ ]: df = pd.read_csv('issue_comments_jupyter_copy.csv')
df['org'] = df['org'].astype('str')
df['repo'] = df['repo'].astype('str')
df['comments'] = df['comments'].astype('str')
df['user'] = df['user'].astype('str')
```

```
In [6]: import seaborn as sns

plt.figure(figsize=(5,5))

sns.countplot(y='repo', data=df, color ='c').set_title('Count of Jupyter GitHub Comments per Repo')
plt.show()
```



Pearson

A Tool For Interactive Computing

- A conversation between the human and the computer.
- Assemble ideas using the computer as playground, as “data microscope.”
- A way to assemble the “building blocks” of scientific computing and data science.

Jupyter in the Wild

- Jupyter is open source—which means anyone can download it and use it for free.
- Digital journalism, scientific research, commercial products, classrooms, coding boot camps, online learning, conferences, data engineering pipelines, high performance computing (HPC) . . .
- Physics, astronomy, biology, economics and finance, social sciences, geo sciences, digital humanities . . .
- Commercial big data platforms: Microsoft, IBM, Google, Bloomberg, more . . .

The Jupyter Ecosystem

- Jupyter Notebook
- JupyterHub
- JupyterLab
- Widgets
- Binder
- IPython

Jupyter's Roots are in IPython

- IPython created in 2001 by Fernando Perez
- IPython notebooks created in 2011
- The Jupyter Notebook — named in 2014 — is the evolution of the IPython Notebook into a language neutral, language agnostic environment

Project Jupyter & the Jupyter Team

- Project Jupyter is this *ecosystem* of tools built by a global team of scientists, researchers, and developers over 15 years
- The core team is actually fairly small- about 18-30 core, full-time contributors at any one time.
- The project has had a long history in the scientific computing community and has benefited from contributions from over 500 volunteer scientists and developers in academia, industry and beyond.
- The Jupyter Code of Conduct sets expectations for the community to enable a diverse group of users and contributors to participate.

Sustaining Jupyter

- Academic grants
- Corporate grants
- Institutional partners

NumFOCUS

- 501(c)(3) nonprofit that supports and promotes open source scientific computing platforms and projects.
- Provides operational infrastructure for other well-known open source projects including pandas, matplotlib, scikit-learn, and many others.
- Organizes global PyData conferences

If you'd like to support NumFOCUS or open source projects like Jupyter, see the NumFOCUS and Jupyter websites.

Finding Resources on jupyter.org

Finding Resources on jupyter.org

- Installation
- About Jupyter
- Community
- Documentation
- Hosted notebooks: nbviewer / GitHub /Binder
- Blog

Learn By Example

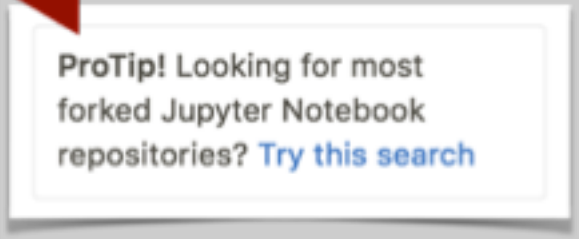
- Gallery of Interesting Jupyter Notebooks

Learn By Example

- Gallery of Interesting Jupyter Notebooks
- Notebooks on GitHub

Over 1 Million Public Notebooks on GitHub

CHECK OUT “TRENDING REPOSITORIES”



https://github.com/trending/jupyter-notebook?since=weekly

Additional Resources

- Project Jupyter GitHub repositories
- Many conference videos online
  - JupyterCon
  - SciPy
  - EuroSciPy
  - PyData
  - PyCon
  - Plotcon
  - JuliaCon
  - Strata

Contributing to the Jupyter Ecosystem

- Contributing to the existing code base via GitHub
  - github.com/jupyter
  - github.com/jupyterlab
  - github.com/jupyterhub
  - github.com/jupyter-widgets
  - github.com/ipython

Contributing to the Jupyter Ecosystem

- Contributing to the existing code base via GitHub
- Adding new core functionality through extensions and enhancement proposals

Jupyter Notebook

Jupyter Notebook

- Is the evolution of the *IPython Notebook* into a language agnostic computing environment supporting over 100 programming languages including Python, R, and Julia.

Jupyter Notebook

- Is the evolution of the *IPython Notebook* into a language agnostic computing environment supporting over 100 programming languages including Python, R, and Julia.
- A digital “document” that gives you a way to centrally record your thoughts, code, and visualizations.

Jupyter Notebook

- Is the evolution of the *IPython Notebook* into a language agnostic computing environment supporting over 100 programming languages including Python, R, and Julia.
- A digital “document” that gives you a way to centrally record your thoughts, code, and visualizations.
- A tool to explore, communicate, and tell stories about your data in a systematic, reproducible way.

Jupyter Notebook

- Is the evolution of the *IPython Notebook* into a language agnostic computing environment supporting over 100 programming languages including Python, R, and Julia.
- A digital “document” that gives you a way to centrally record your thoughts, code, and visualizations.
- A tool to explore, communicate, and tell stories about your data in a systematic, reproducible way.
- Provides access to many Python libraries

Jupyter Notebook - JSON

- The underlying framework of the Jupyter Notebook is JSON (JavaScript Object Notation).
- JSON provides the ability to store diverse metadata in the notebook.

JSON underlying a Jupyter Notebook


```
{
 "cells": [
  {
   "cell_type": "markdown",
   "metadata": {
    "deletable": true,
    "editable": true
   },
   "source": [
    "# Jupyter Comment Clusters\n",
    "### (K-Means Clustering)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 2,
   "metadata": {
    "collapsed": true,
    "deletable": true,
    "editable": true
   },
   "outputs": [],
   "source": [
    "import numpy as np\n",
    "import pandas as pd\n",
    "from sklearn.feature_extraction import text\n",
    "from sklearn.feature_extraction.text import CountVectorizer"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 3,
   "metadata": {
    "collapsed": true,
    "deletable": true,
    "editable": true
   },
   "outputs": [],
   "source": [
    "df = pd.read_csv('issue_comments_jupyter_copy.csv')\n",
    "df['org'] = df['org'].astype('str')\n",
    "df['number'] = df['number'].astype('str')\n",
    "df['repo'] = df['repo'].astype('str')\n",
    "df['comments'] = df['comments'].astype('str')\n",
    "df['user'] = df['user'].astype('str')"
   ]
  },
```

Jupyter Notebook - JSON

- The underlying framework of the Jupyter Notebook is JSON (JavaScript Object Notation).
- JSON provides the ability to store diverse metadata in the notebook.
- Useful when exporting the notebook for sharing.

Let’s Dive In

Navigation and Basic Commands

Differences Between Mac & Windows

- When you install Anaconda on Windows you may need to run “install Jupyter” in the command line.
- Typing “jupyter notebook” should just work on a Mac
- Mac file paths use forward slashes, Windows uses backslashes

Which Python Version Do I Install?

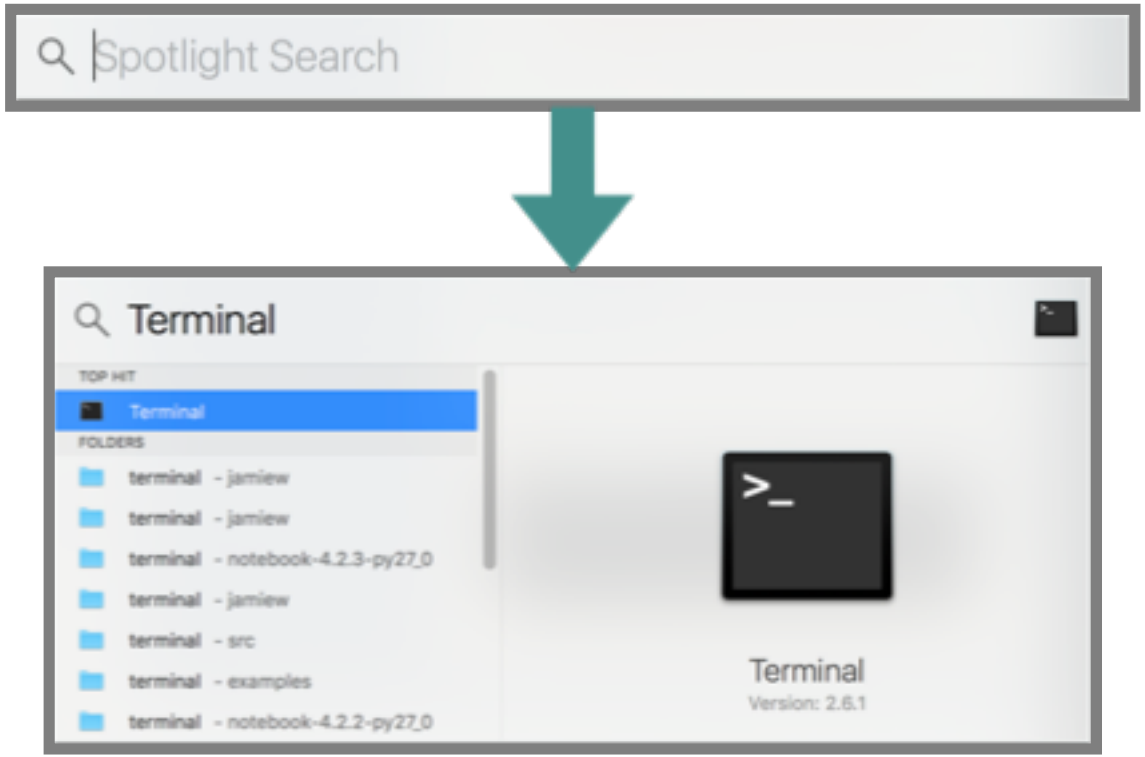
Important

This documentation covers IPython versions 6.0 and higher. Beginning with version 6.0, IPython stopped supporting compatibility with Python versions lower than 3.3 including all versions of Python 2.7.

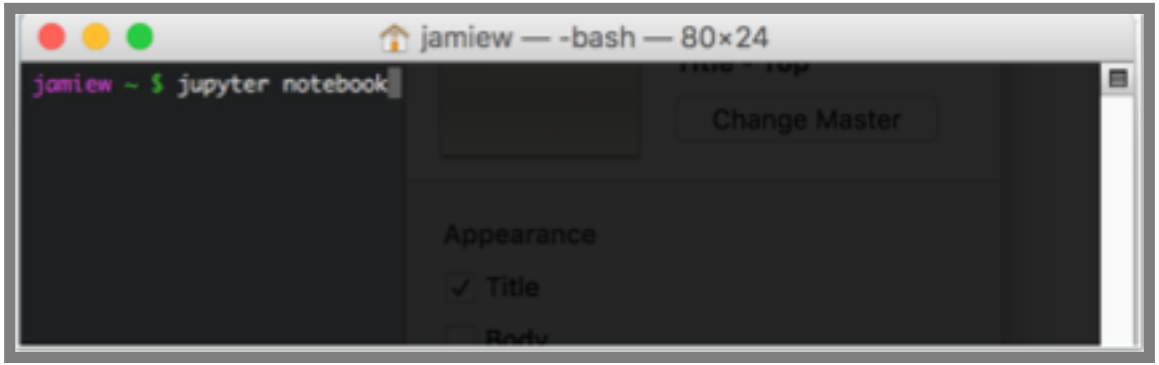
If you are looking for an IPython version compatible with Python 2.7, please use the IPython 5.x LTS release and refer to its documentation (LTS is the long term support release).

- Jupyter Installation - Success?

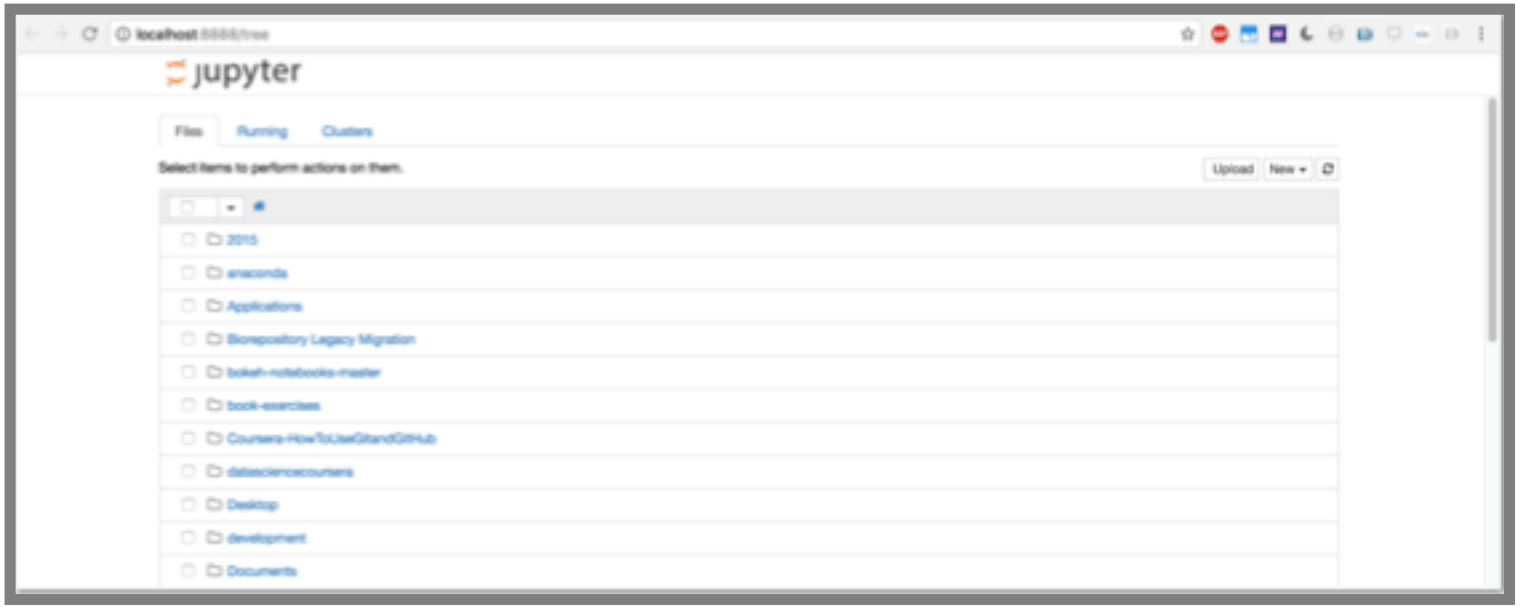
Open Your Terminal via Spotlight Search



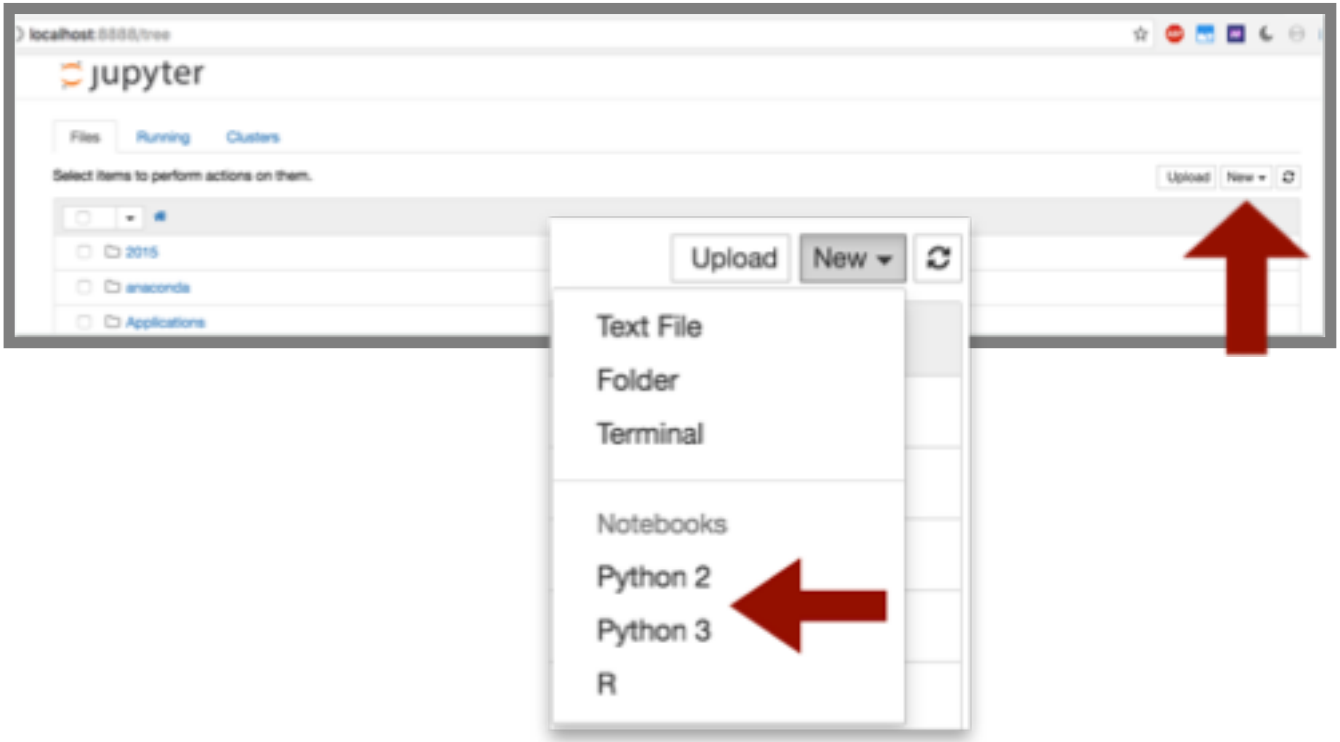
Launch Jupyter From the Command Line



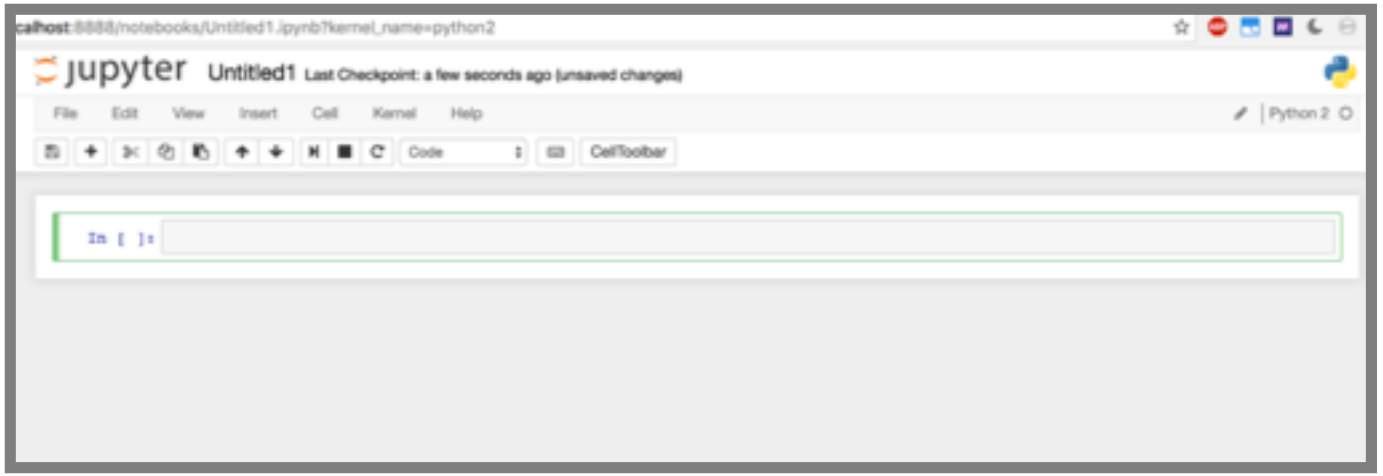
Jupyter File Browser



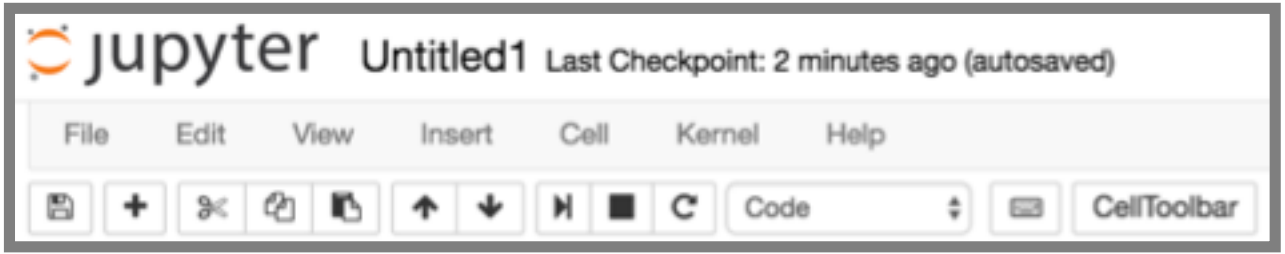
Create a New Notebook



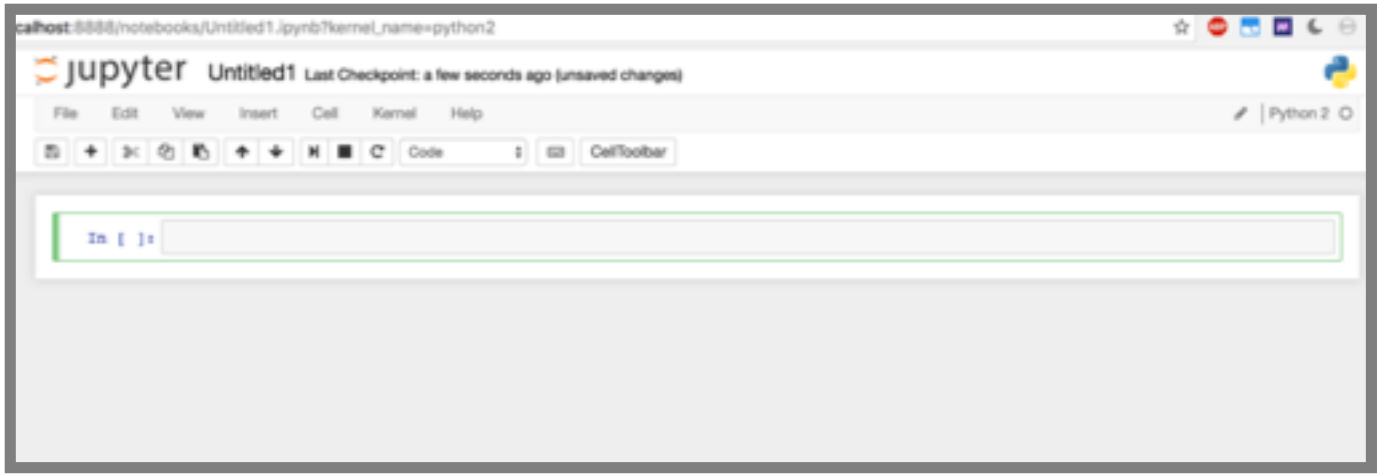
The Jupyter Notebook Document



Navigating the Jupyter Notebook



The Jupyter Notebook Document



Warm Up

```
#Define a variable
x = 15
print(x)
```

Break

Jupyter Notebook

Quantitative and Visual Exploratory Data Analysis (EDA) in Python

The Data Science Workflow

1. Pose a question or problem
2. Acquire data
3. Explore the data by writing & running code
4. Prepare and clean the data
5. Complete final analysis & visualize results
6. Write up analysis for publication (blogs, journals, etc.)
7. Share what you've done
8. Reproduce what other people have done

A Note About Notebook Hygiene

- For better or for worse, the larger/longer your notebook analysis, the easier it is to get disoriented.
- Leave breadcrumbs. Use markdown headers to break up the sections of your analysis.
- Be kind to future you (or to your colleagues) who will open the notebook later and either wonder what you were thinking or be grateful it's easy to follow your analysis.

Saving Your Notebook in GitHub

- Use GitHub (or another file or code repository) to store your Notebook. That way you don't lose all your hard work!

Jupyter Notebook Exploratory Data Analysis

Importing Libraries

Importing Libraries

import ____ as _____

CONVENTIONS

import matplotlib.pyplot as plt

%matplotlib inline

import pandas as pd

import seaborn as sns

from sklearn import _____

Import Libraries Into Your Notebook

```
%matplotlib inline

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Jupyter Notebook Exploratory Data Analysis

Connecting to Data Sets Using Pandas

Reading in Data

```
df = pd.read_csv('Chronic_Disease_Indicators_CDI_US.csv', dtype=object)

%time
```

Did it work?

Did it work?

```
df.head()
```

Did it work?

```
In [5]: df.head()
```

Out[5]:

	YearStart	YearEnd	LocationAbbr	LocationDesc	DataSource	Topic	Question	Response	DataValueUnit	DataValueType	...	LocationID	Topi
0	2015	2015	AK	Alaska	YRBSS	Alcohol	Alcohol use among youth	NaN	%	Crude Prevalence	...	2	ALC
1	2015	2015	AL	Alabama	YRBSS	Alcohol	Alcohol use among youth	NaN	%	Crude Prevalence	...	1	ALC
2	2015	2015	AR	Arkansas	YRBSS	Alcohol	Alcohol use among youth	NaN	%	Crude Prevalence	...	5	ALC
3	2015	2015	AZ	Arizona	YRBSS	Alcohol	Alcohol use among youth	NaN	%	Crude Prevalence	...	4	ALC
4	2015	2015	CA	California	YRBSS	Alcohol	Alcohol use among youth	NaN	%	Crude Prevalence	...	6	ALC

Manipulating Data Using Pandas

Demo in Jupyter Notebook

Jupyter Notebook Exploratory Data Analysis

Making Charts

Popular Data Visualization Packages

- Matplotlib
- Seaborn
- Bokeh
- Altair
- Plotly

Making Charts with Matplotlib

http://matplotlib.org/users/screenshots.html#simple-plot

- Simple plot
- Histograms
- Bar Charts
- Scatter plots

Always remember to use '%matplotlib inline' to display the plot in your notebook

Making Charts with Seaborn

Seaborn

Making Charts with Bokeh

Bokeh

Making Charts with Altair

Altair

Making Charts with Plotly

Plotly

Break

Return in 10 minutes

Break

Welcome Back!

Kernels

Kernels

- Kernels enable language-specific notebooks
- A kernel is a program that runs and introspects your code.
- You can install over 100 other language kernels – including the R kernel – by following the latest online documentation for your kernel of interest:
  github.com/jupyter/jupyter/wiki/Jupyter-kernels

Kernels in Jupyter

- An IPython kernel extension also enables %%R and %%Julia cell magics in a Python Notebook
- The Jupyter frontend interface communicates with your kernel.
- Sometimes you will encounter a dead kernel –
  - save your notebook when this occurs
  - either restart the kernel, refresh your browser, restart your notebook or do all three in succession.
- Sometimes you'll want to interrupt your kernel, use the menu bar option to interrupt
- The kernel shuts down when the Jupyter application is closed, typing 'Ctrl-C' in your terminal will also shut down the kernel

Converting & Sharing Notebooks

nbconvert

nbconvert

Converts your Jupyter Notebook into another format:

- web-display: presentation slides, html
- publishable documents: (PDF)
- plain-text: markdown
- executable scripts: e.g., *.py

nbconvert Requires *pandoc*

To check if pandoc is installed:

pandoc - -version

If needed, install it by :

sudo apt-get install pandoc

or

brew install pandoc

Convert Your Notebook to HTML

```
> jupyter nbconvert --to html CDC_Chronic_Disease_Indicators_CDI_Analysis_US.ipynb
```

This command creates an HTML output file named

CDC_Chronic_Disease_Indicators_CDI_Analysis_US.html

Check your project folder for the output file

Convert Your Notebook to *.py File

```
> jupyter nbconvert --to script CDC_Chronic_Disease_Indicators_CDI_Analysis_US.ipynb
```

This command creates a executable output file named

CDC_Chronic_Disease_Indicators_CDI_Analysis_US.py

Check your project folder for the output file

Converting & Sharing Notebooks

RISE: Present Your Analysis

RISE

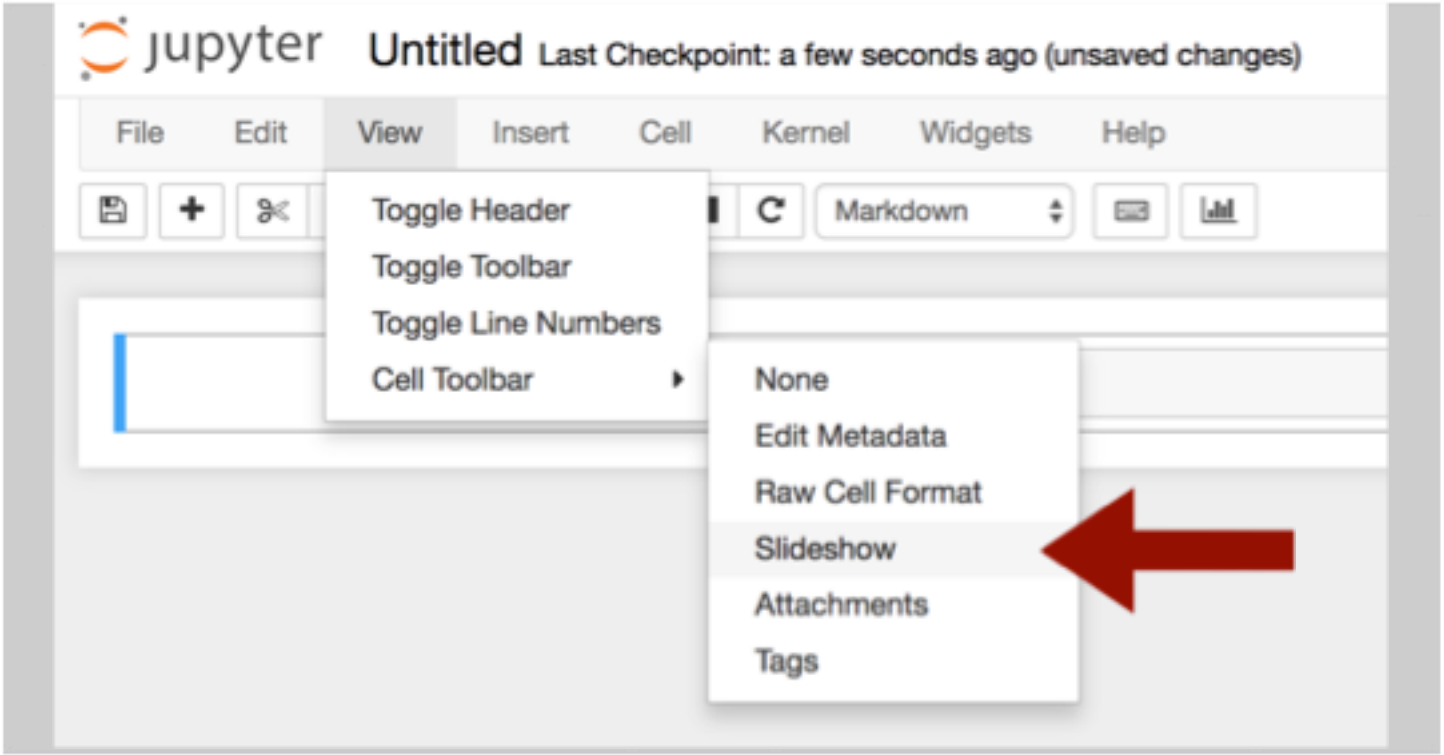
- ‘Reveal.js-Jupyter/IPython Slideshow Extension’
- Renders your notebook as an executable slideshow

Install RISE

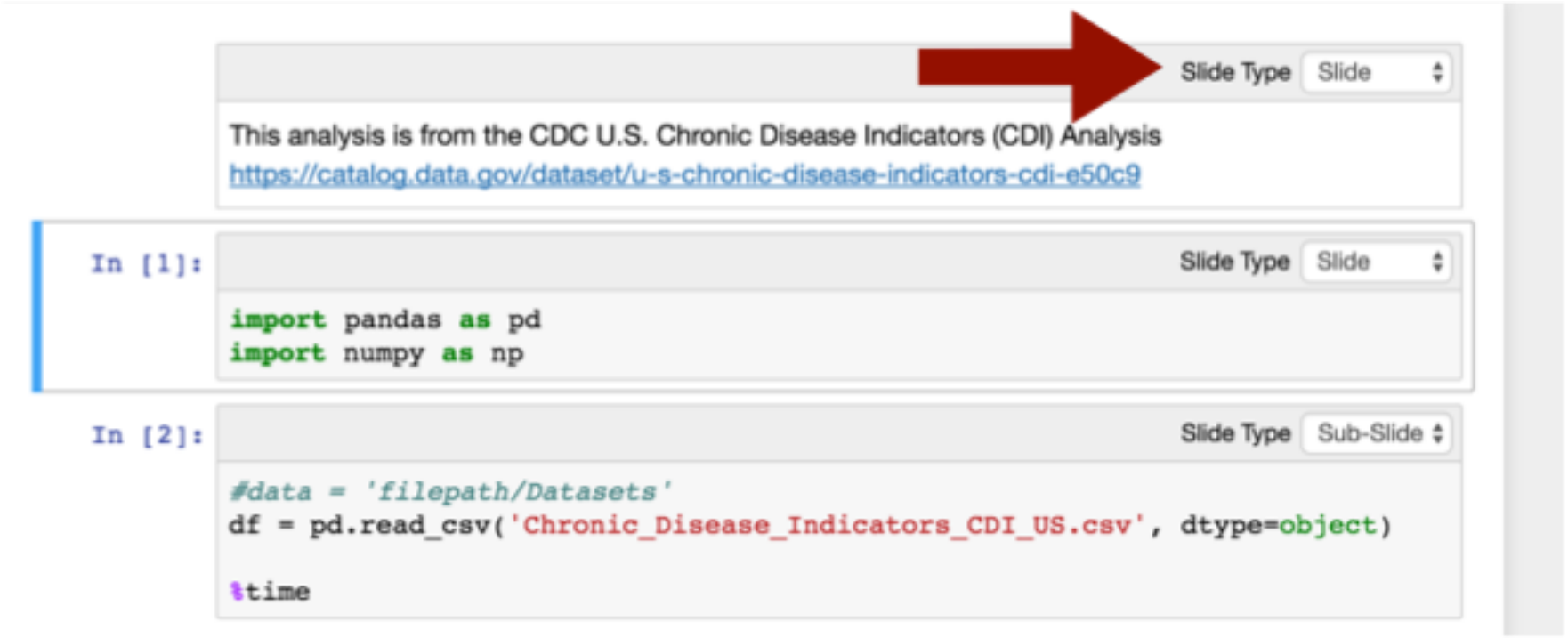
```
> conda install -c damianavila82 rise
```

Open Your Notebook

Select Slideshow from View Menu



Set Slide Type and Slide Order



Click ’Enter Slideshow’ Icon

U.S. CDC Chronic Disease Indicators (CDI) Analysis

Learn more about RISE

https://github.com/damianavila/RISE

Converting & Sharing Notebooks

Sharing Notebooks

Working with *.ipynb files

- *.ipynb files are just like any other document file and can be shared via e-mail or external drives
- Best practice is to use a version control system like Git to keep track of changes

GitHub

- Standard Git repository hosting service that provides version control
- Accessible to collaborators via the GitHub url or through a private url if your company is running an internal instance
- Renders static notebooks natively – viewers cannot interact with analysis
- Collaborators can fork and/or download your files and submit changes via pull requests

Azure notebooks

- Provide free online access to a notebook environment
- Requires an account & sign-in
- https://notebooks.azure.com/

nbviewer

- The web application behind The Jupyter Notebook Viewer hosted by Rackspace.
- Can be used outside of or within an organization to share notebooks through a centralized server
- Use it here: nbviewer.jupyter.org
- Learn more: github.com/jupyter/nbviewer

nteract

- Desktop application for Jupyter notebooks
- Automatically opens *.ipynb files (without the Terminal)
- Auto-updates
- Actively maintained by a great team
- Packages from nteract are also used in Hydrogen and JupyterLab
- The nteract team wants you to use your favorite tool in every analysis
- web version of the nteract interface is currently under development
- More at nteract.io

nbdime

- Tools for diffing and merging notebooks
- Documentation: nbdime.readthedocs.io

Break

Return in 10 minutes

Other Projects in the Jupyter Ecosystem

Other Projects in the Jupyter Ecosystem

JupyterHub

JupyterHub

- Designed to handle multiple users from a single server
- Each user receives their own Jupyter instance that is independent of all other users
- Used extensively in education, research, and on data science teams
- JupyterHub requires Python 3.4 or higher.
- Running JupyterHub from Windows is not supported

JupyterHub Deployments

- UC Berkeley Foundations of Data Science Course
- UC San Diego Supercomputing Center
- George Washington University
- University of Minnesota
- CERN
- Compute Canada
- Quantopian, 100,000+ users compete and share algorithms

JupyterHub on Cloud Service Providers

- Amazon Web Services (AWS)
- Google Cloud Platform
- Microsoft Azure
- Rackspace
- Red Hat
- Everware

Zero to JupyterHub provides instructions for Kubernetes model deployment across multiple cloud services at z2jh.jupyter.org.

System Requirements

- Linux/Unix-based system
- Python 3.5 or greater
- A domain name
- nodejs/npm
- TLS certificate and key for HTTPS communication
- Jupyter Notebook 4.0 or greater

JupyterHub: Key Elements

- A proxy (configurable-http-proxy)
- An Authenticator
- A Spawner
- The Hub
- Jupyter Notebook

IMPORTANT!

Deploy JupyterHub securely with SSL encryption. Follow the instructions in the JupyterHub Security Settings documentation.

Learn More About JupyterHub

- jupyterhub.readthedocs.io/
- zero-to-jupyterhub.readthedocs.io
- github.com/jupyterhub/jupyterhub-tutorial

Other Projects in the Jupyter Ecosystem

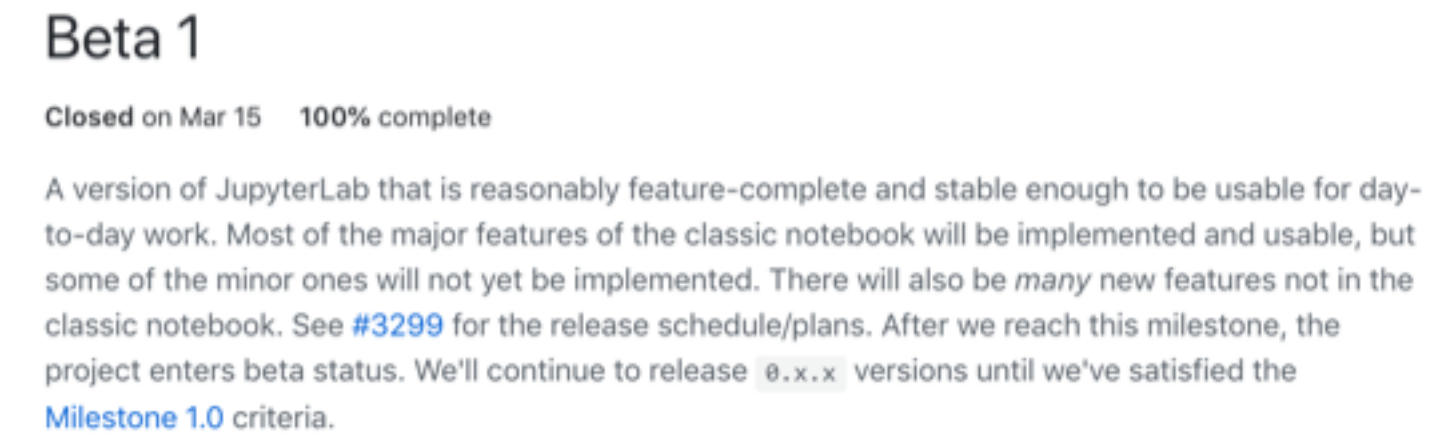
JupyterLab

JupyterLab

- Interactive development environment for working with notebooks, code, and data.
- Integrates all the Jupyter tools into a comprehensive work space.
- Includes new features in addition to features found in the classic notebook.
- JupyterLab Beta 1 released in early 2018, a first in a series of beta releases leading up to the full 1.0 release.

JupyterLab (Beta)

- Evolution of the Jupyter Notebook's interactive computing environment



Beta 1

Closed on Mar 15 **100%** complete

A version of JupyterLab that is reasonably feature-complete and stable enough to be usable for day-to-day work. Most of the major features of the classic notebook will be implemented and usable, but some of the minor ones will not yet be implemented. There will also be *many* new features not in the classic notebook. See #3299 for the release schedule/plans. After we reach this milestone, the project enters beta status. We'll continue to release `0.x.x` versions until we've satisfied the Milestone 1.0 criteria.

Install JupyterLab

Installation

With conda:

```
conda install -c conda-forge jupyterlab
```

With pip:

```
pip install jupyterlab
```

Explore JupyterLab

Launch JupyterLab by typing the command:

```
jupyter lab
```

JupyterLab can also be launched by entering the notebook server's URL (http://localhost:8888) in the browser.

JupyterLab Demo

https://github.com/jupyterlab/jupyterlab/milestones

Other Projects in the Jupyter Ecosystem

Real Time Collaboration

In Alpha: Real Time Collaboration

- Notebooks integrated with Google Drive
- Uses Google Drive API* to enable multiple users to edit the same notebook at one time
- Integrated chat window for additional collaboration support

** Google has deprecated their Realtime API. Existing applications will work until December 2018, new applications will not be able to use the API.*

Other Projects in the Jupyter Ecosystem

Widgets

Jupyter Widgets

- Quickly growing part of the project
- Advanced tools that make your notebook more interactive
- Include
  - Maps
  - 3-D visualizations
  - Beaker integration
  - Sliders
  - Progress bars
  - Text input
- Documentation
  - jupyter.org/widgets.html
  - ipywidgets.readthedocs.io/en/stable/user_guide.html

Other Projects in the Jupyter Ecosystem

Binder & BinderHub

Binder & BinderHub

- Allow you to build, share, and re-create custom computing environments for Jupyter & JupyterHub
- Learn more and follow their development:
  - mybinder.org
  - binderhub.readthedocs.io

Wrap-up

What We Learned Today

- How to install Jupyter
- The history of Jupyter
- How to conduct and end-to-end data analysis using the Jupyter Notebook
- How to share a data analysis
- Jupyter projects available in 2018

Additional Resources

- Learn By Example: Gallery of Interesting Jupyter Notebooks
- Notebooks on GitHub
- Conference videos online:
  - JupyterCon
  - SciPy
  - EuroSciPy
  - PyData
  - PyCon
  - Plotcon
  - JuliaCon
  - Strata

Thanks!

Jamie Whitacre
@datajamie

Intructor’s Resources

- Trademark usage policy