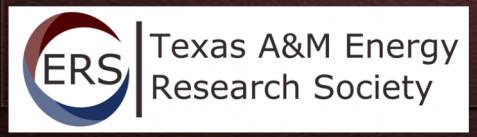
# HPRC Data Workshop

## Texas A&M Conference on Energy

**Zhenhua He**

12/03/2021




High Performance
Research Computing
DIVISION OF RESEARCH

Original slides created by Dr. Jian Tao

# Data Labs

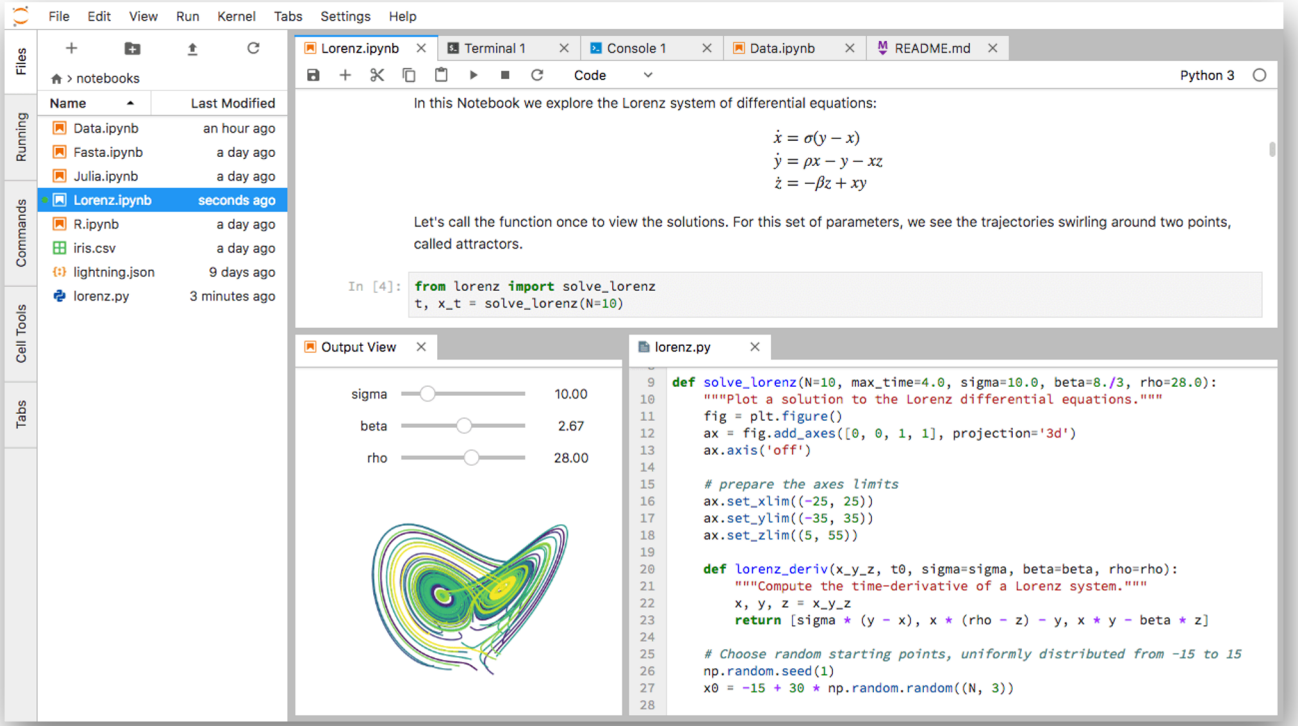
## Lab I. JupyterLab

We will load a module with required libraries and run JupyterLab on the HPRC Terra Portal.

## Lab II. Data Exploration

We will go through some examples with two popular Python libraries: Pandas and Matplotlib for data exploration.

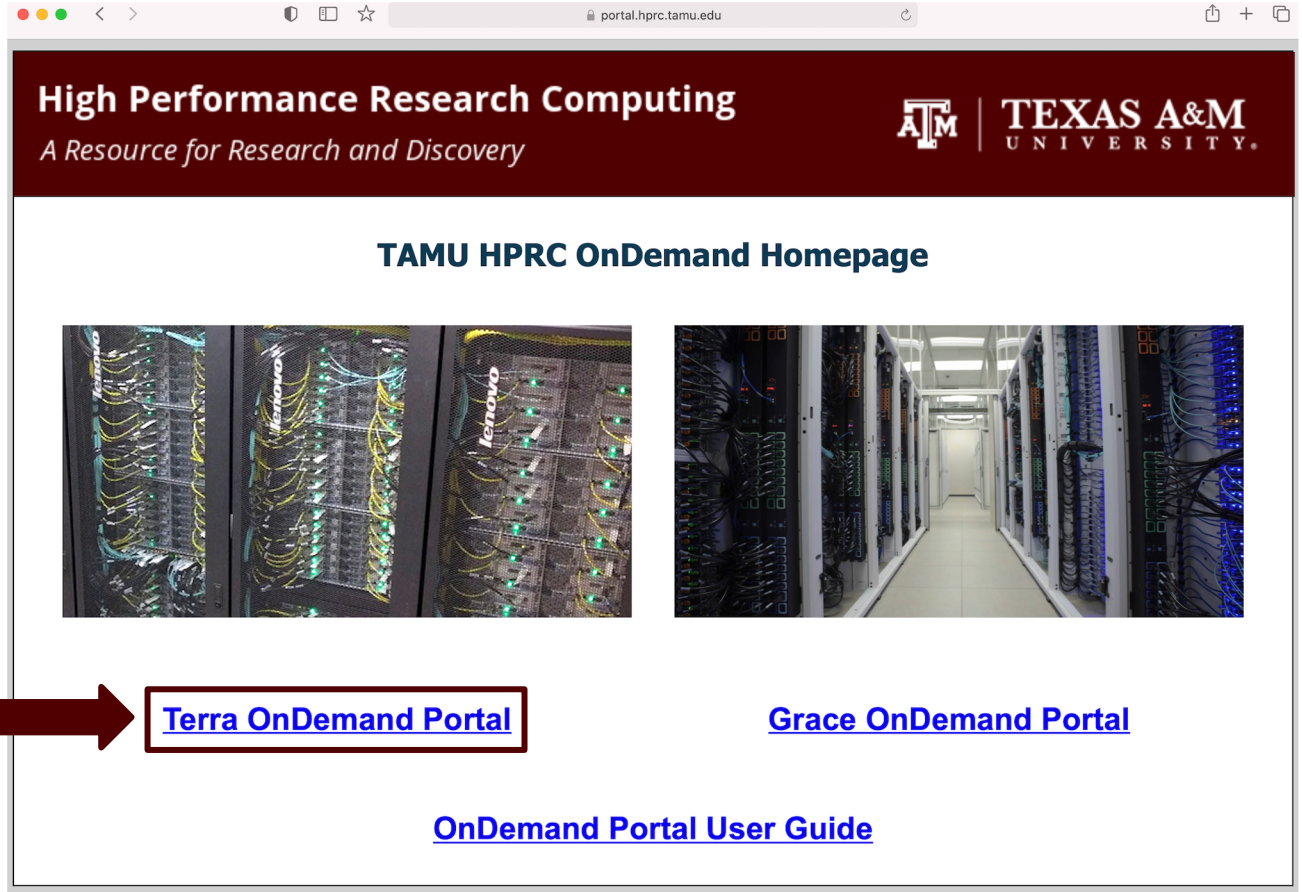
# Lab I. JupyterLab



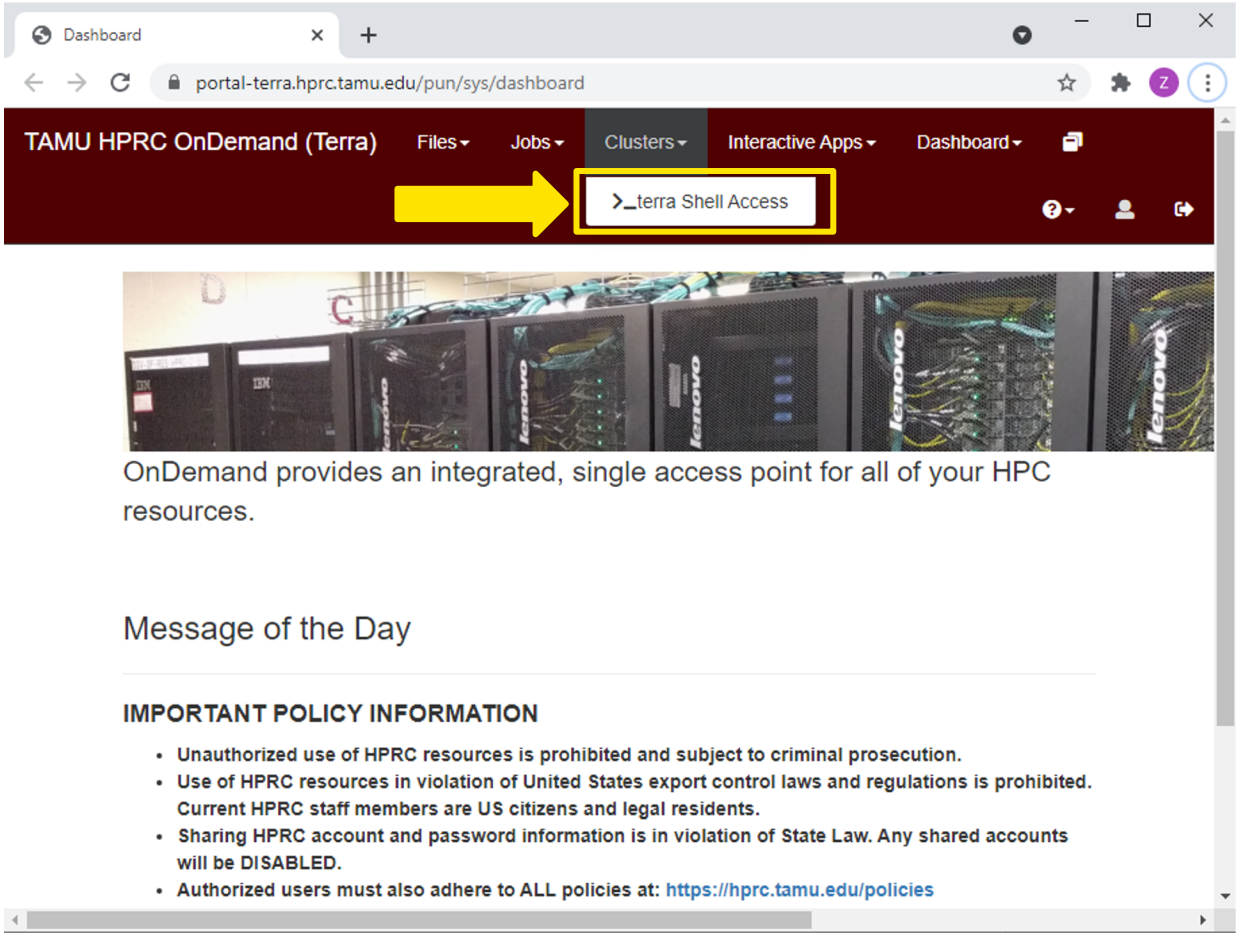
# L1 - Resources

- Texas A&M High Performance Research Computing (HPRC)
- Terra Quick Start Guide
- HPRC Portal
- HPRC YouTube Channel
- Jupyter Project
- help@hprc.tamu.edu

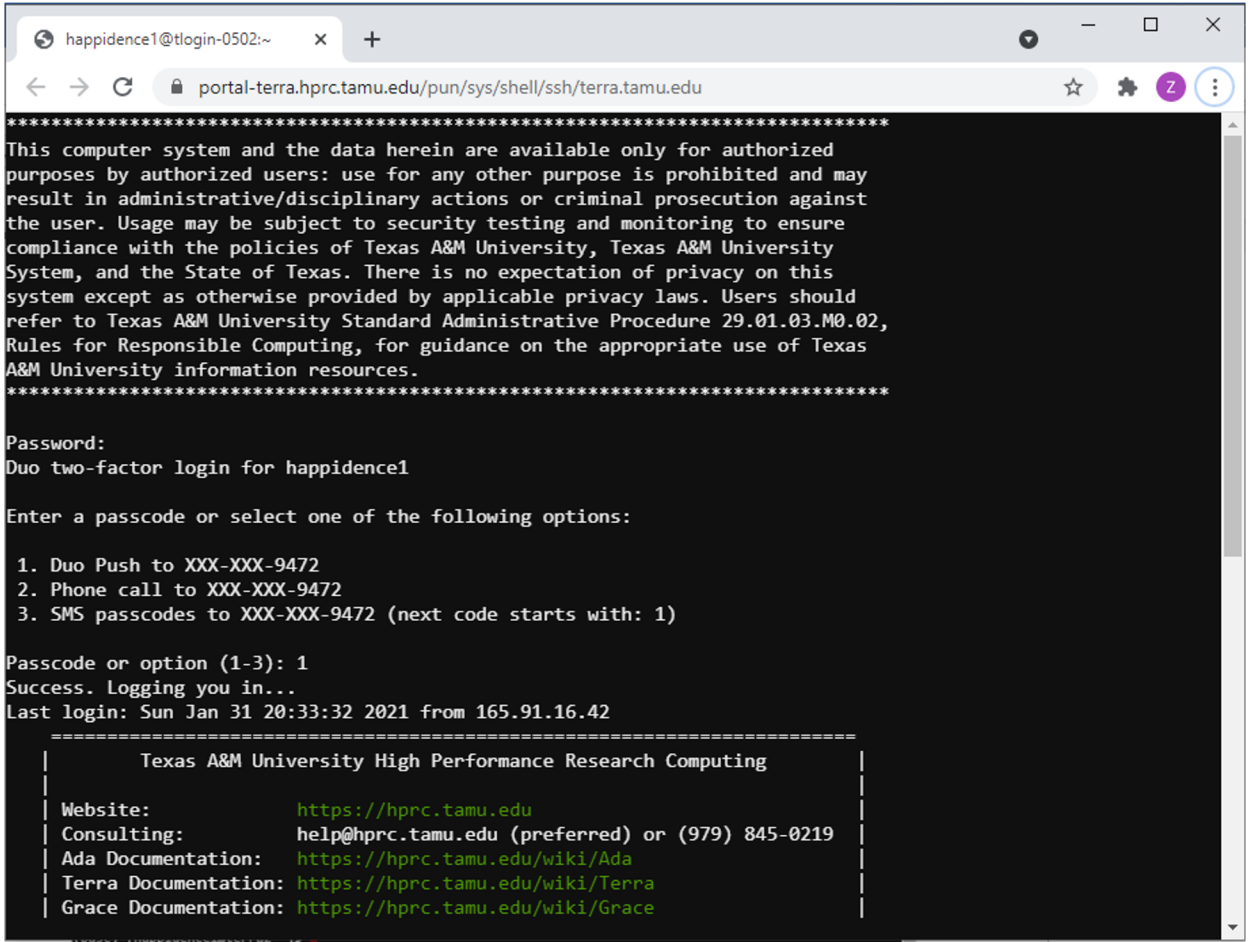
# Login HPRC Portal



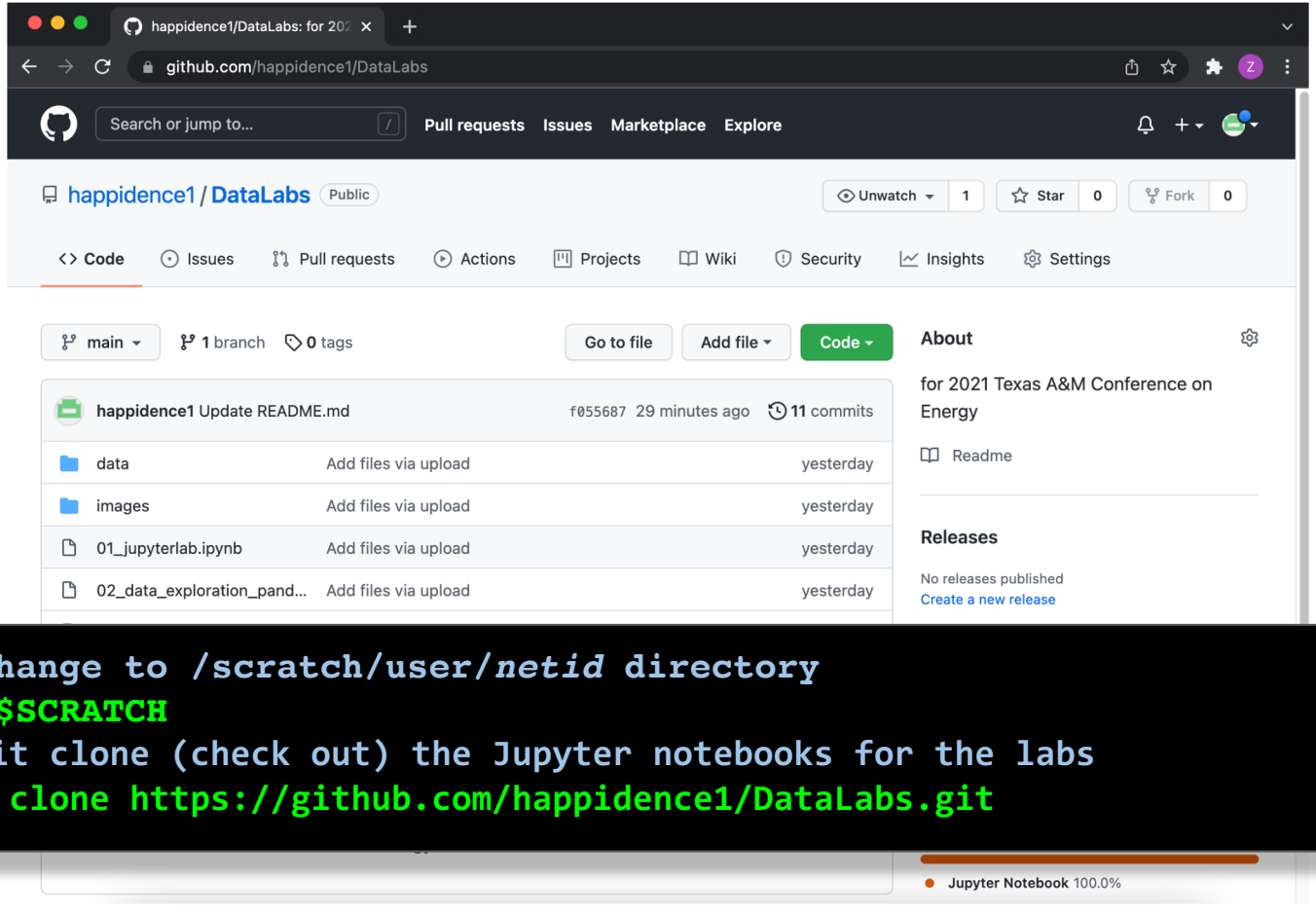
# Shell Access - I



# Shell Access - II

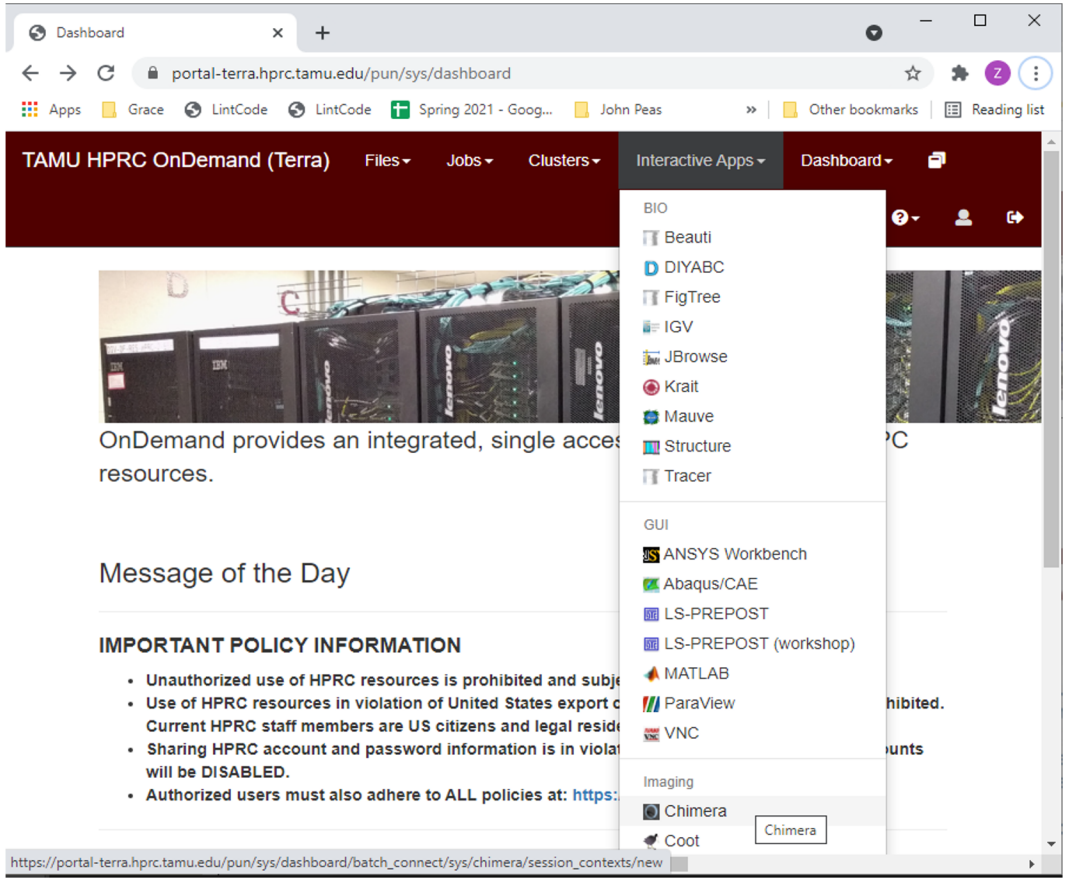


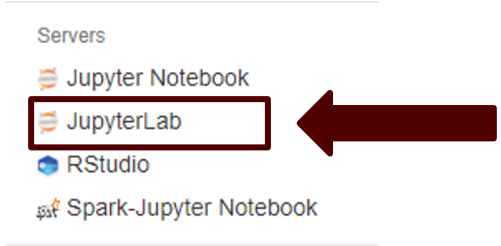
# Check out Exercises



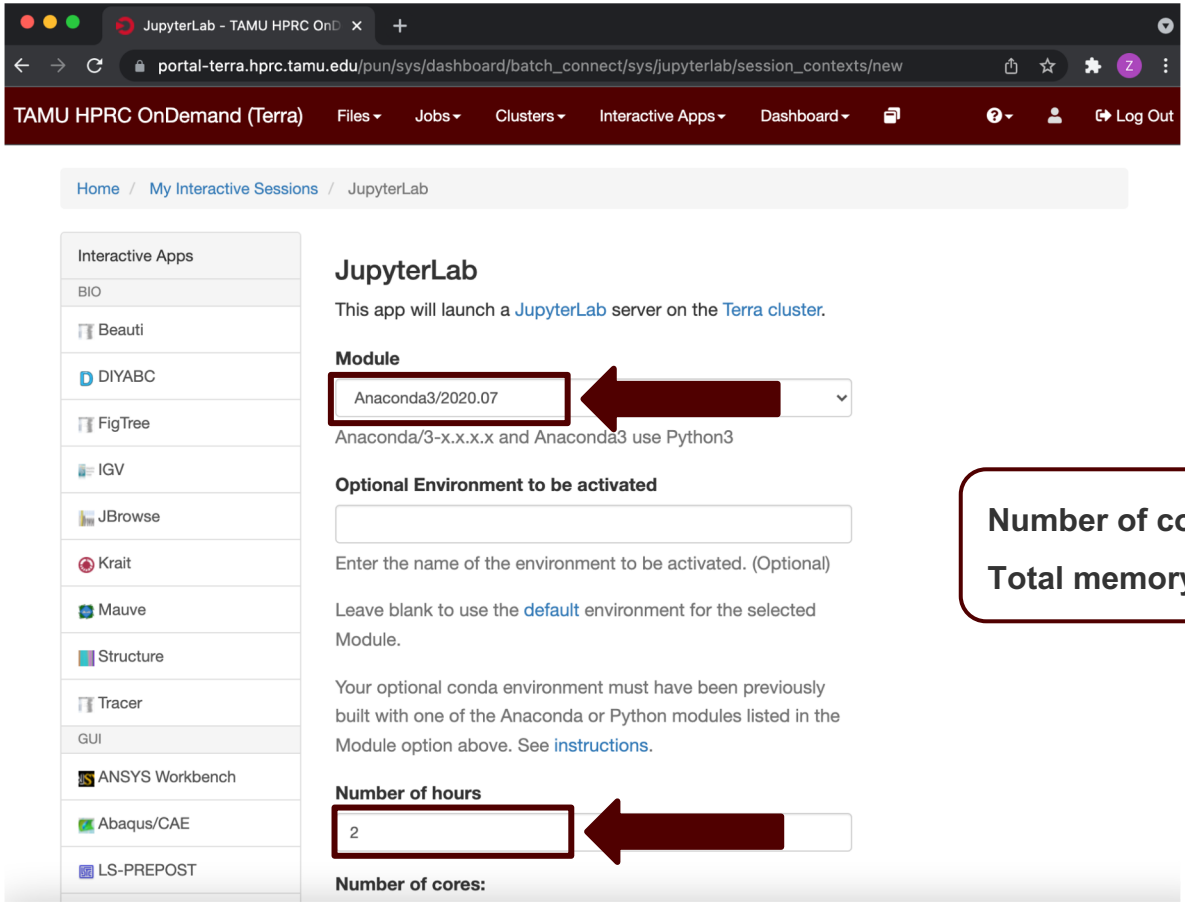

```
# Change to /scratch/user/netid directory
cd $SCRATCH
# git clone (check out) the Jupyter notebooks for the labs
git clone https://github.com/happidence1/DataLabs.git
```

# Go to JupyterLab Page



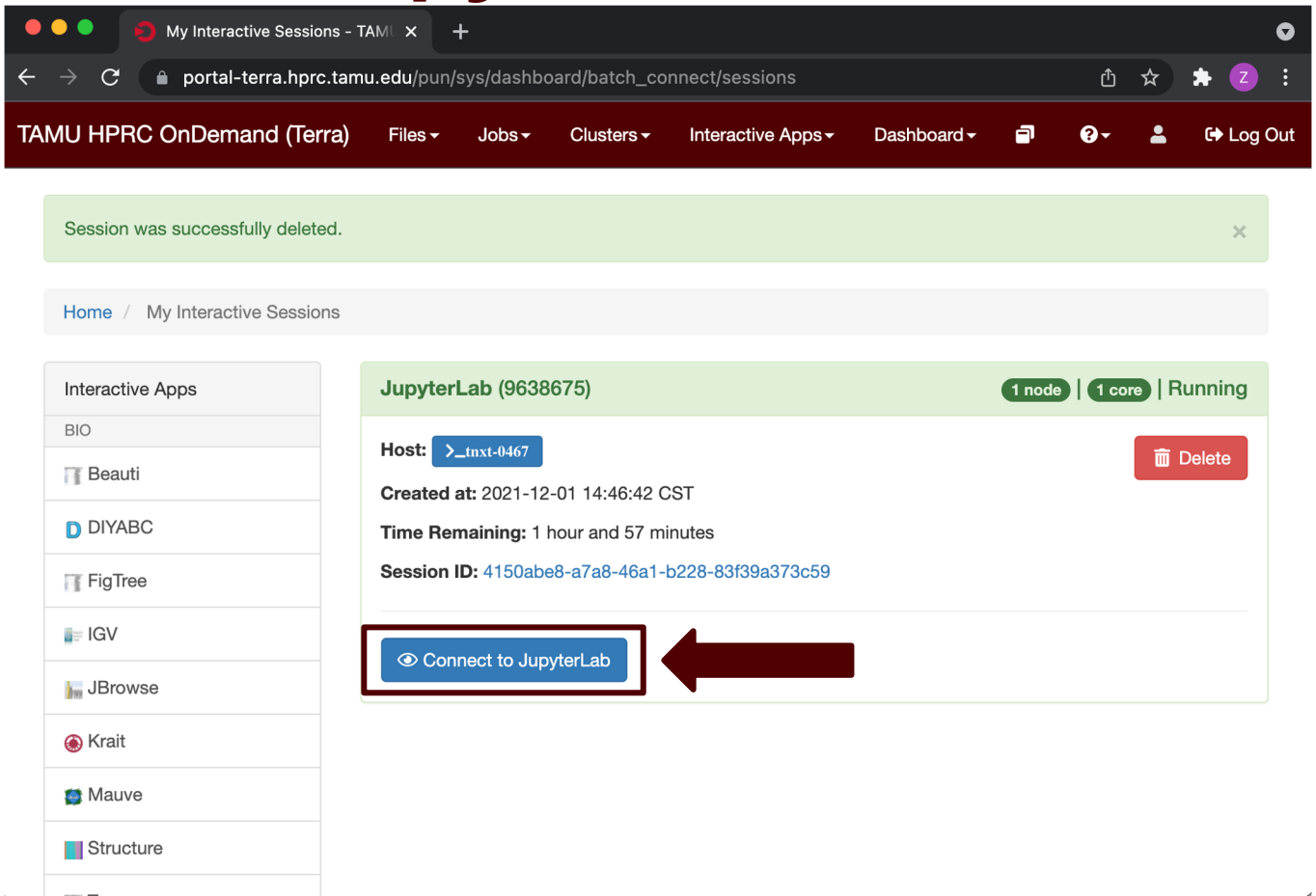
# Load module



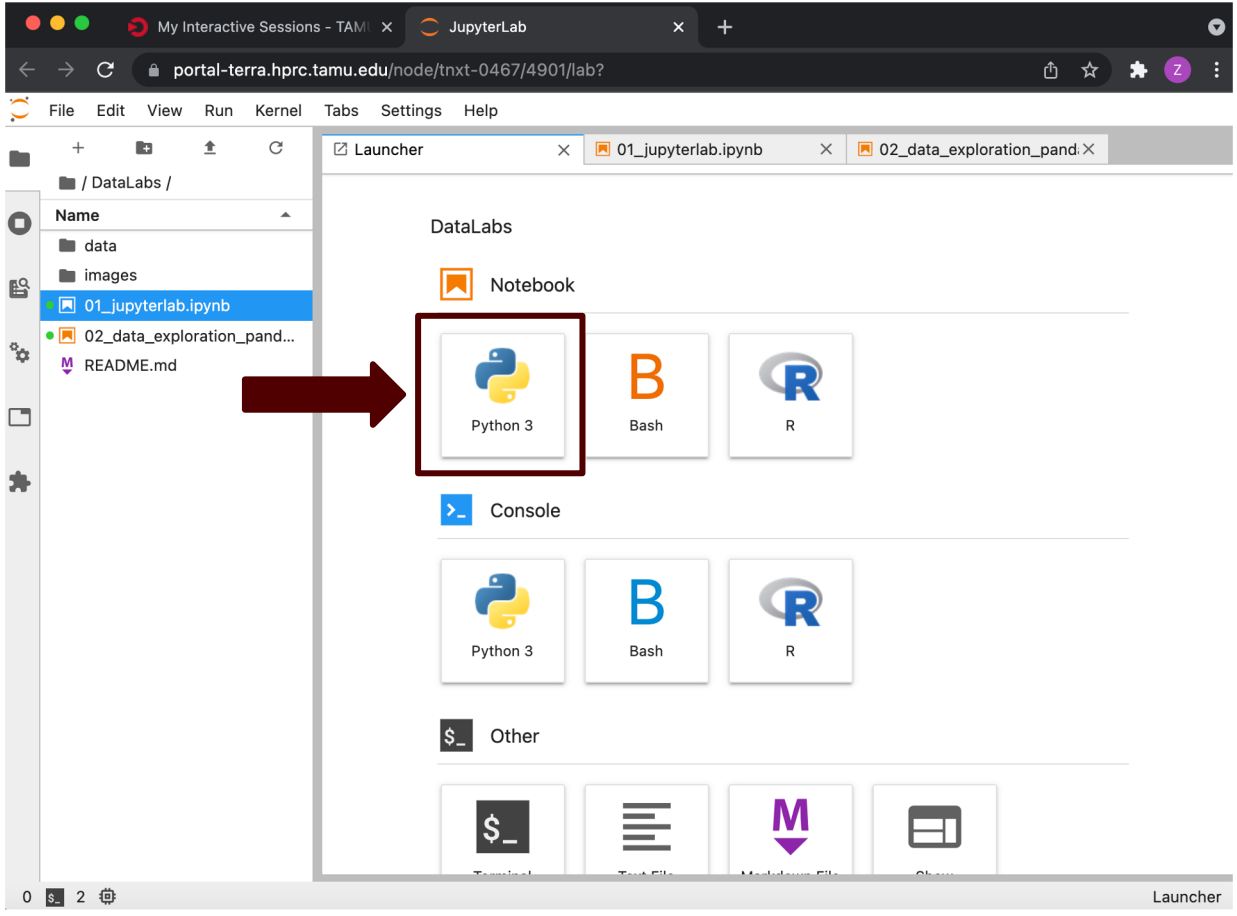

**Number of cores: 2**

**Total memory (GB): 4**

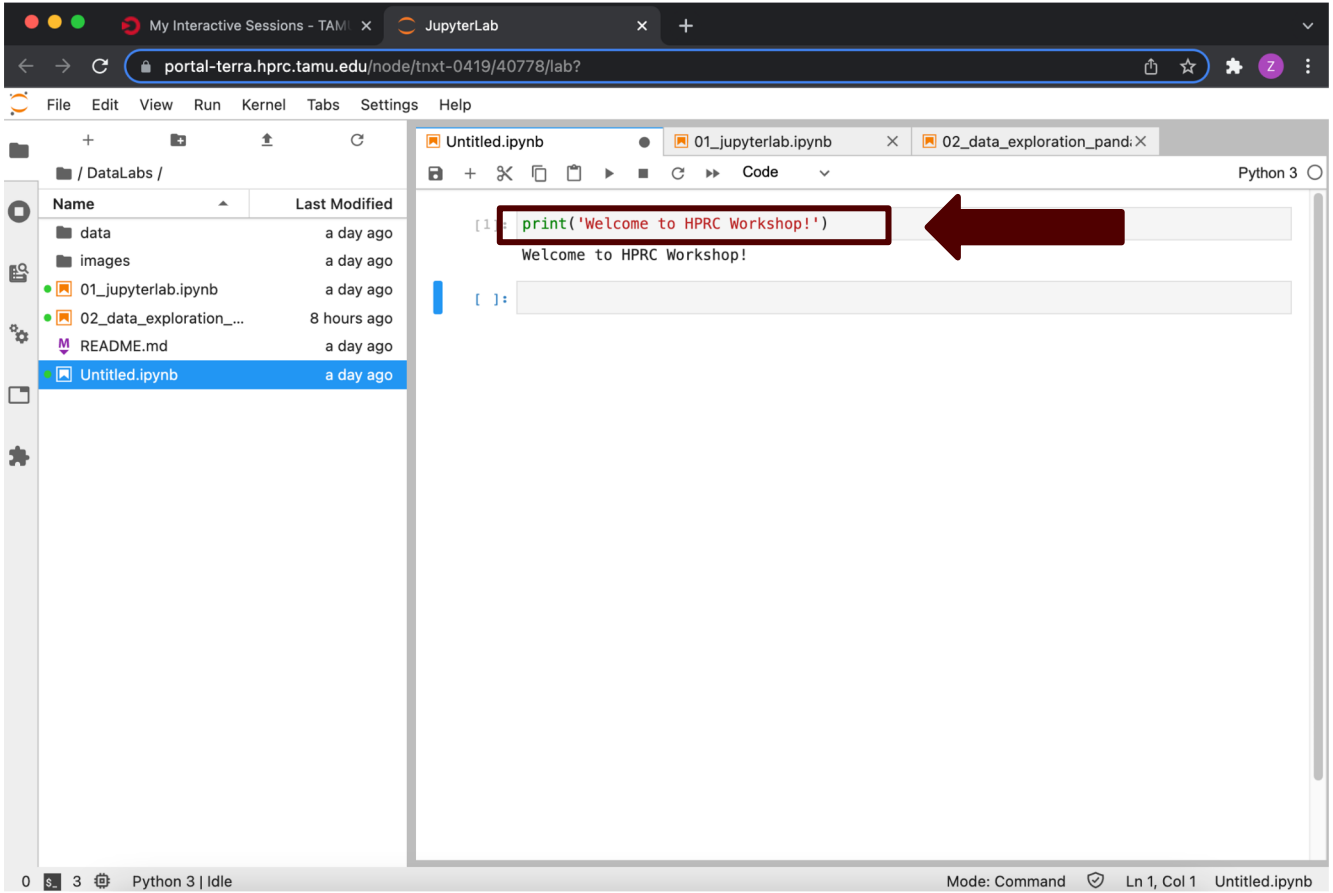
# Connect to JupyterLab



# Create a Jupyter Notebook



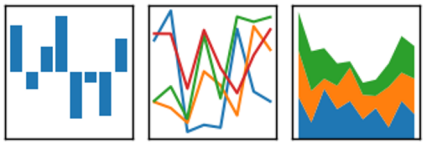
# Test JupyterLab



# Lab II. Data Exploration

pandas

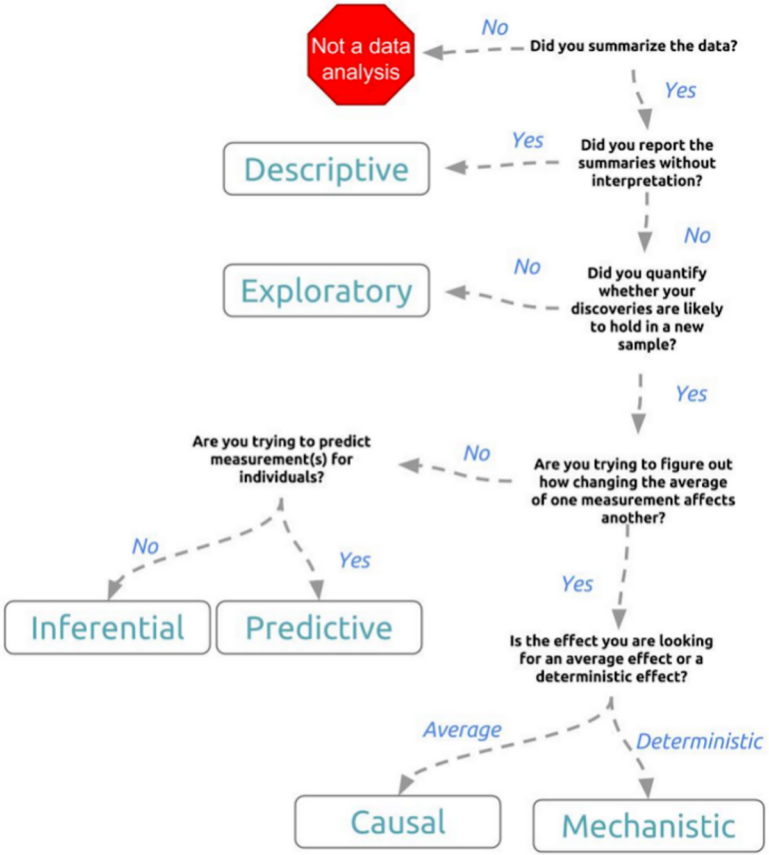
$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



matplotlib

# Types of Data Science Problems

- **Descriptive** (summaries, e.g., census)
- **Exploratory** (search for unknowns, e.g., four-planet solar system)
- **Inferential** (find correlations, e.g., many social studies)
- **Predictive** (make predictions, e.g., Face ID, Echo, Siri)
- **Causal** (explore causation, e.g., smoking versus lung cancer)
- **Mechanistic** (determine governing principles, e.g., experimental science)




Credit: Jeff Leek - The Elements of Data Analytic Style

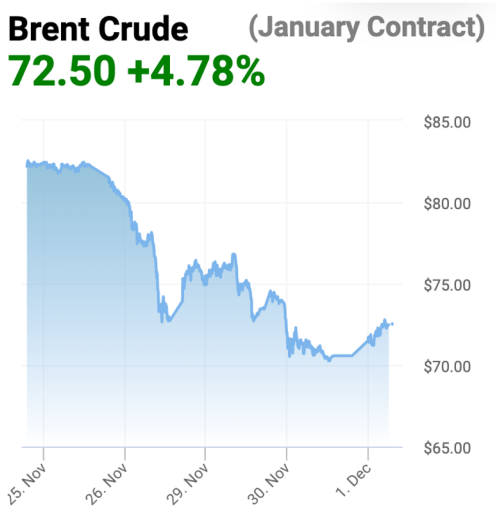
# Data Structures

**Pandas** has two data structures that are descriptive and optimized for data with different dimensions.

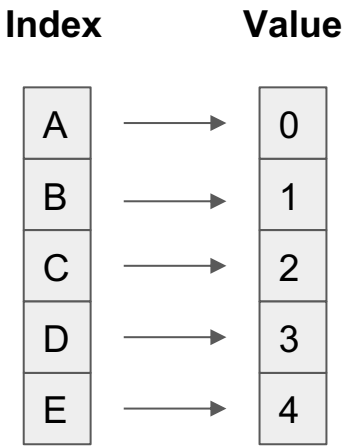
- **Series:** 1D labeled array
- **DataFrame:** General 2D labeled, size-mutable tabular structure with potentially heterogeneously-typed columns

# Series in pandas

- One-dimensional labeled array
- Capable of holding any data type (integers, strings, floating point numbers, etc.)
- Example: time-series oil price data




(source: oilprice.com)



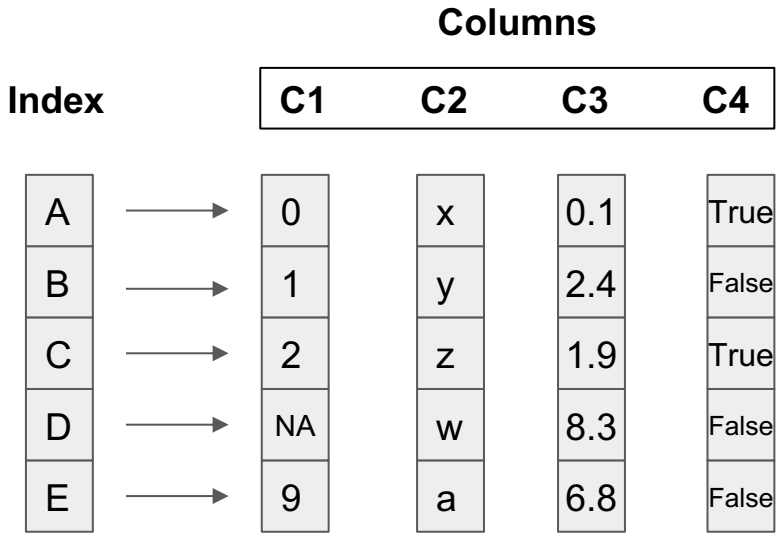
# DataFrame in pandas

- Primary Pandas data structure
- A dict-like container for Series objects
- Two-dimensional size-mutable
- Heterogeneous tabular data structure

**Top producing oil fields in the United States, 2019* being updated**

| Rank | Field | State | Discovery Year | Million Bbl/Day |
|---|---|---|---|---|
| 1 | Permian | Texas/New Mexico | 1920 | 4.2 |
| 2 | Eagle Ford Shale | Texas | 2008 | 1.34 |
| 3 | Bakken | North Dakota/Montana | 1951 | 1.33 |
| 4 | Prudhoe Bay Oil Field | Alaska | 1967 | .791 |
| 5 | Wattenberg Gas Field | Colorado | 1970 | .473 |
| 6 | Shenzi | Federal Gulf of Mexico | 2002 | .353 |
| 7 | Kuparuk River oil field | Alaska | 1969 | .295 |
| 8 | Midway-Sunset Oil Field | California | 1901 | .288 |
| 9 | Atlantis Oil Field | Federal Gulf of Mexico | 1998 | .273 |
| 10 | Sugarkane | Texas | 2009 | .258 |

(source: wikipedia)



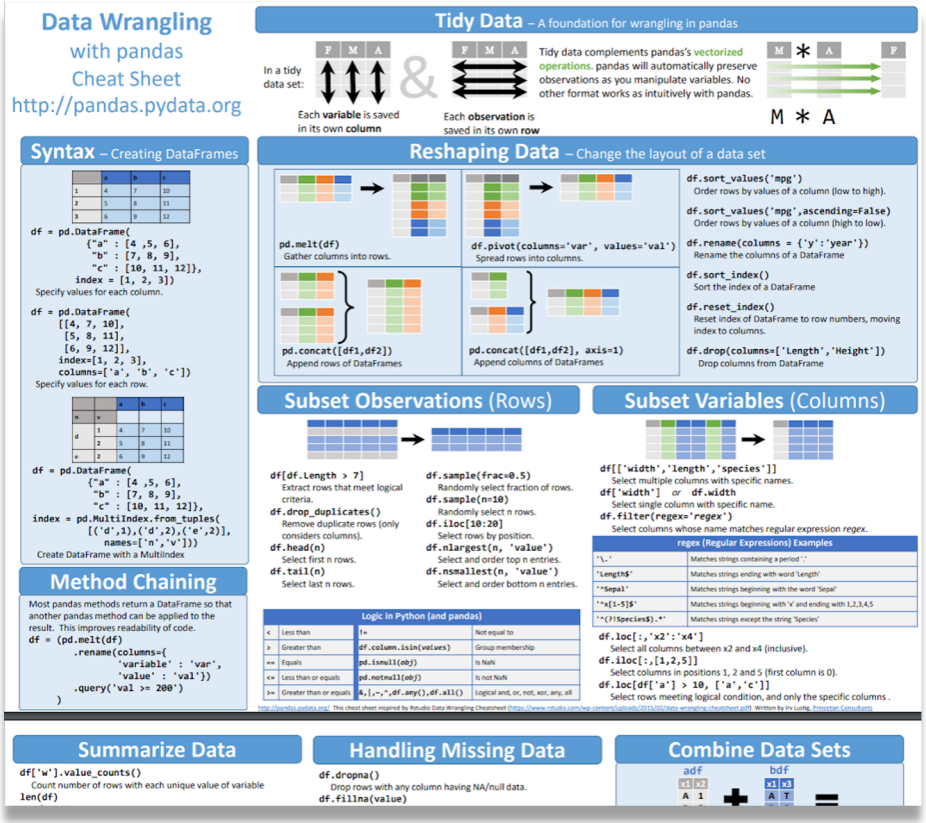
# Pandas Learning Objectives

**After this lesson, you will know how to**:

- Create a DataFrame
- Drop Entries
- Index, Select, and Filter data
- Sort data
- Input and Output

**JupyterLab Exercises**

# Pandas Cheat Sheet




https://pandas.pydata.org/Pandas_Cheat_Sheet.pdf

# Key Plotting Concepts in Matplotlib

- **Matplotlib: Figure**

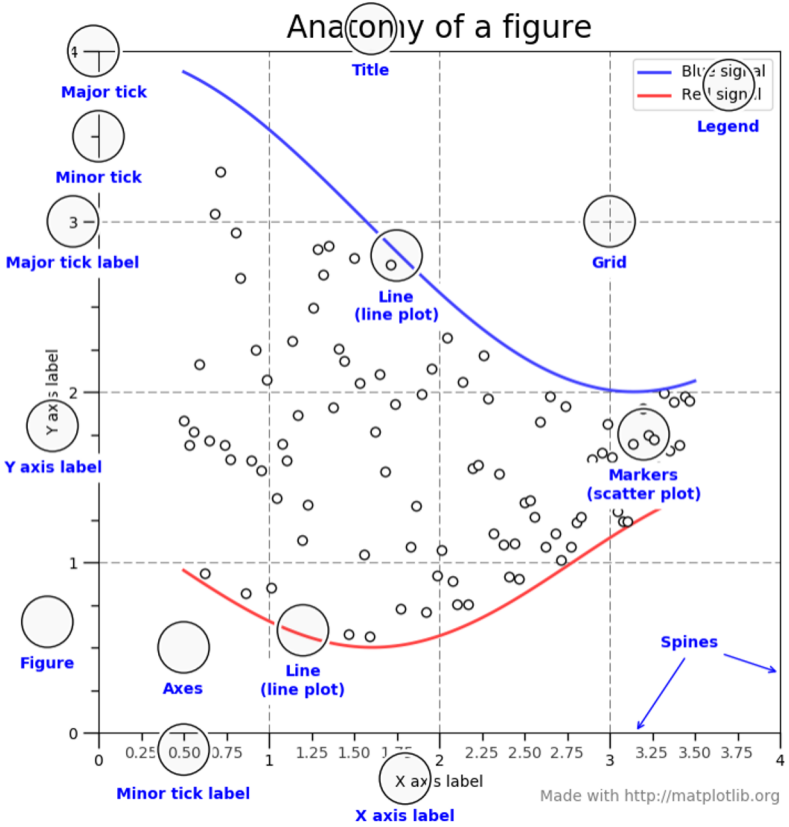
  Figure is the object that keeps the whole image output. Adjustable parameters include:

  1. Image size (set_size_inches())
  2. Whether to use tight_layout (set_tight_layout())

- **Matplotlib: Axes**

  Axes object represents the pair of axis that contain a single plot (x-axis and y-axis). The Axes object also has more adjustable parameters:

  1. The plot frame (set_frame_on() or set_frame_off())
  2. X-axis and Y-axis limits (set_xlim() and set_ylim())
  3. X-axis and Y-axis Labels (set_xlabel() and set_ylabel())
  4. The plot title (set_title())




(Credit: matplotlib.org)

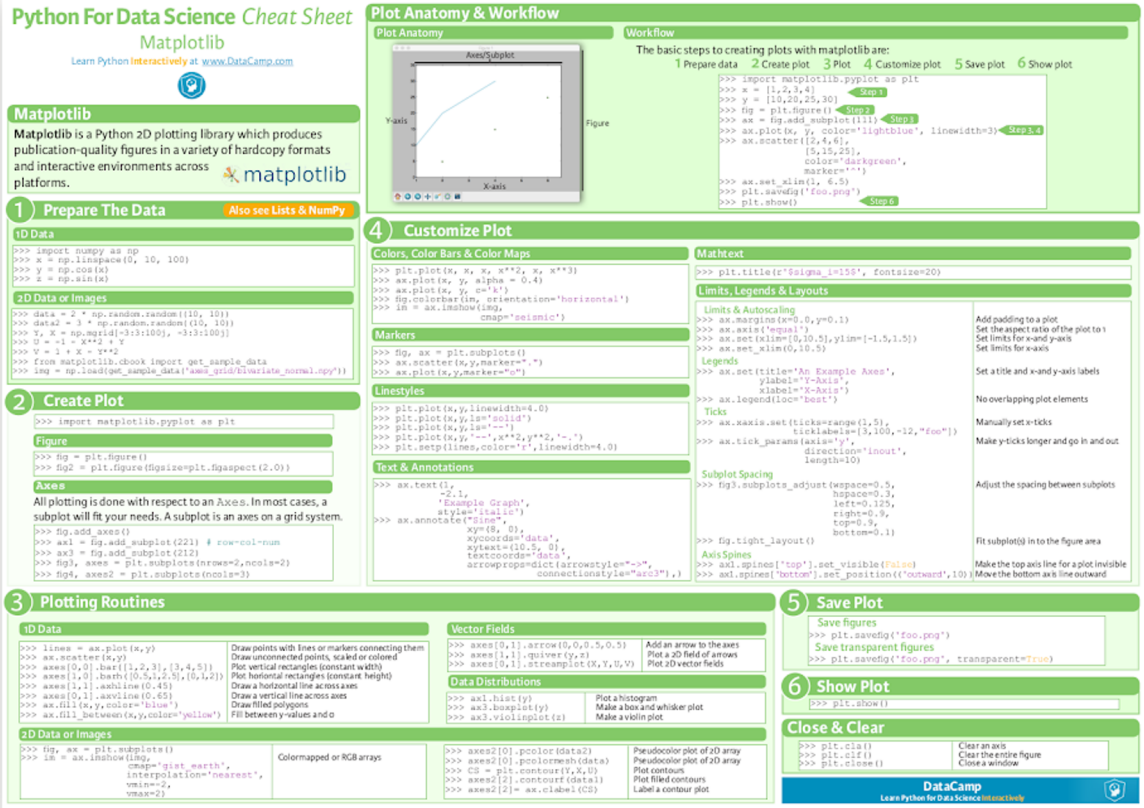
# Matplotlib Learning Objectives

After this lesson, you will know how to:

- Scatter plot and Line plot
- Subplots
- Color map
- Contour figures
- 3D figures
  - Surface plots
  - Wire-frame plot
  - Contour plots with projections

**JupyterLab Exercises**

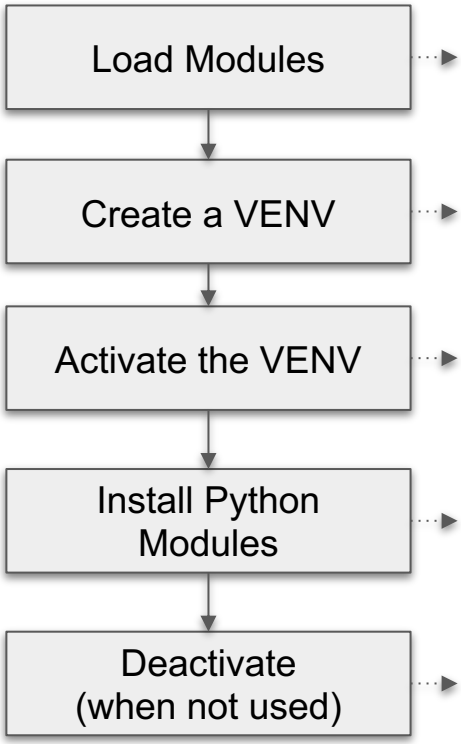
# Matplotlib Cheat Sheet




https://s3.amazonaws.com/assets.datacamp.com/blog_assets/Python_Matplotlib_Cheat_Sheet.pdf

# Additional Materials

# Python Virtual Environment (VENV)




```
# Change to /scratch/user/netid directory
cd $SCRATCH

# clean up and load Anaconda
module purge
module load Anaconda/3-5.0.0.1

# create a Python virtual environment
conda create -n mylab

# activate the virtual environment
source activate mylab

# install required package to be used in the portal
conda install -c conda-forge jupyterlab=1.2.2
conda install pandas matplotlib
conda install scikit-learn
conda install tensorflow

# deactivate the virtual environment
# source deactivate
```

# Common Anaconda Commands

```
# Conda virtual environment
conda info                          # show Conda installation
conda create -n VENV                # create a virtual environment
conda create -n VENV python=3.4 # create a venv with a py version
conda env list                      # list installed venv

# Conda package management
conda list                          # list all installed packages
conda search  PACKAGENAME           # search a Conda package
conda install PACKAGENAME           # install a Conda package
conda update  PACKAGENAME           # update a Conda package
conda remove  PACKAGENAME           # remove a Conda package
```