Jupyter Lab

Andrew Denner

August 2020 Central Iowa Linux Users Group

Video of talk at: https://youtu.be/3_r0LqzMAD8
Source: https://gitlab.com/denner1/cialug-august-2020-jupyter-lab

Welcome

We will get started shortly

Welcome to CIALUG

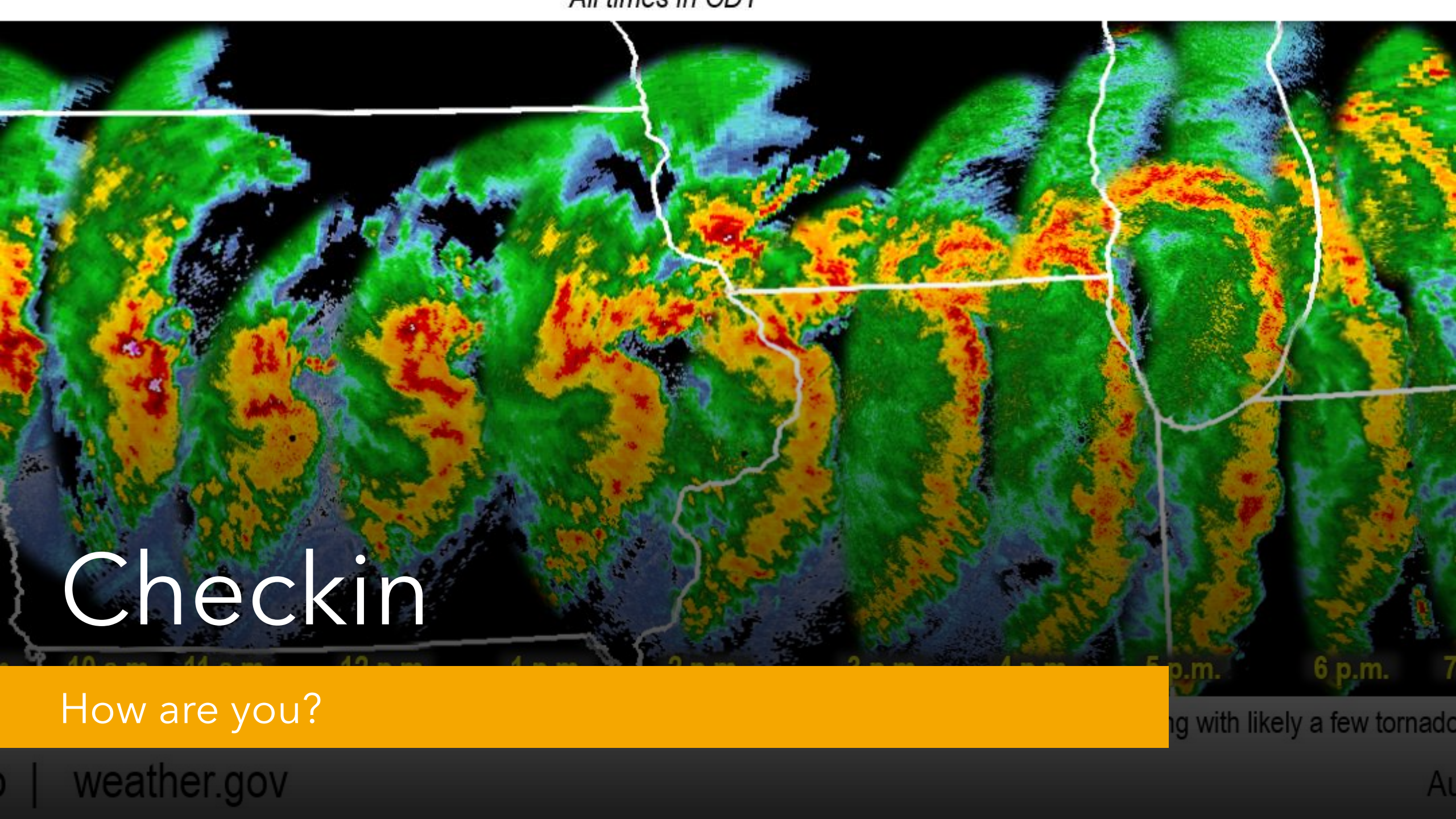
Third Wednesday of the month

http://cialug.org

Email List

IRC/Slack

Want to present?



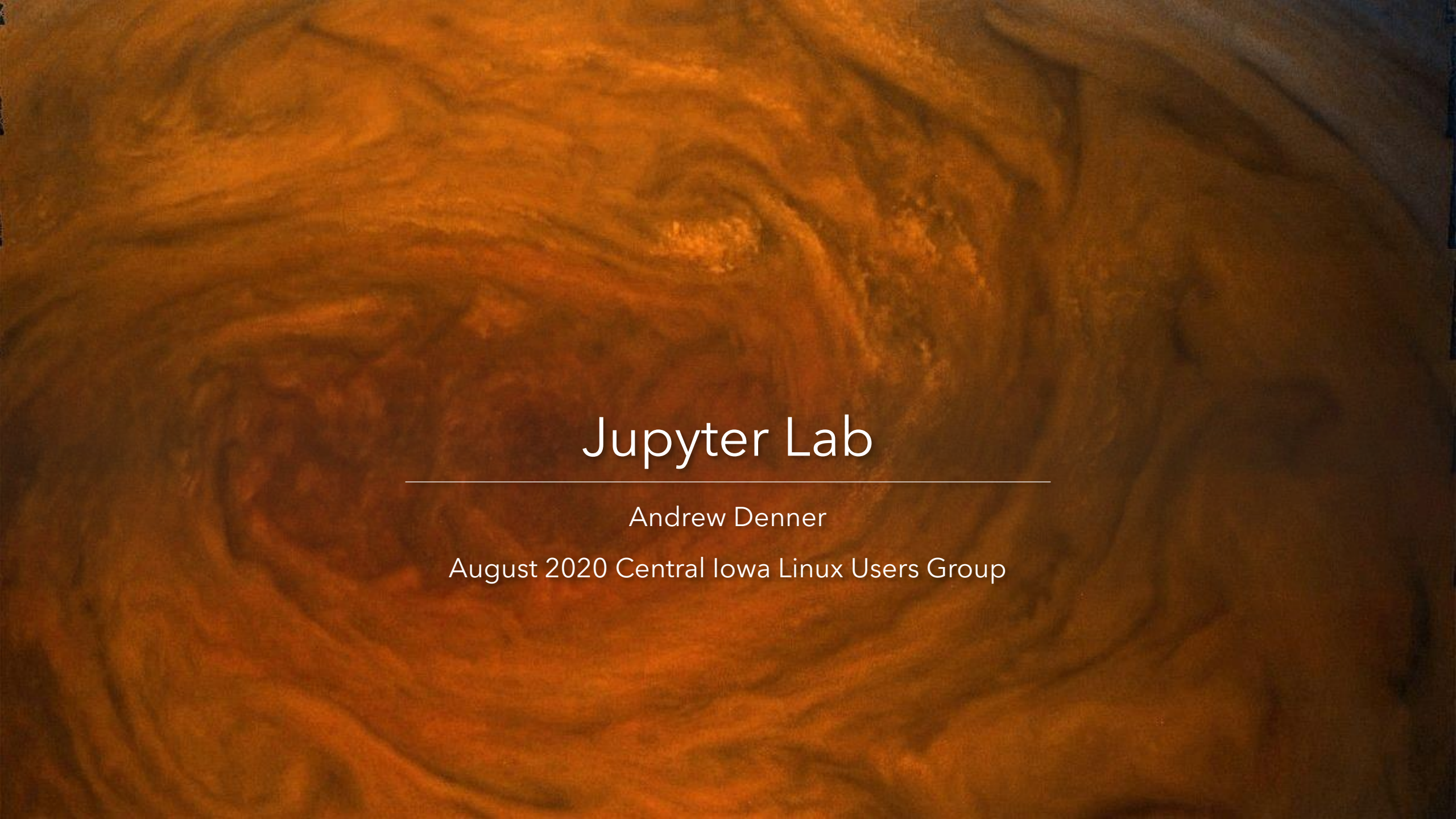
Checkin

How are you?

Linux News

Our Presenter

- Senior Software Developer
- Linux Tinkerer
- Twitter: @adenner
- Slides will be at: http://denner.co



Jupyter Lab

Andrew Denner

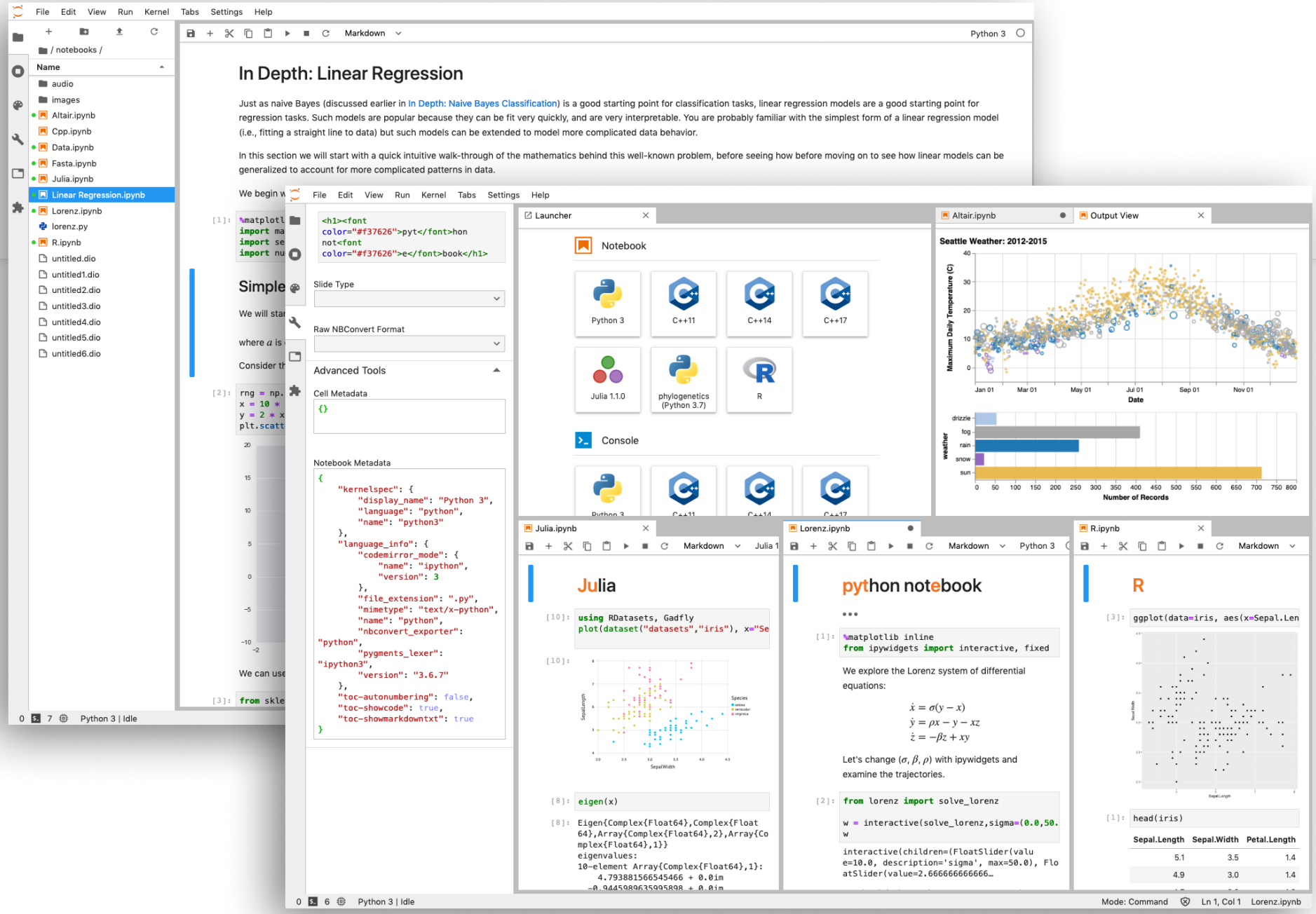
August 2020 Central Iowa Linux Users Group

A brief history

- Spun off from ipython in 2014 by Fernando Pérez
- Jupyter core languages
  - Julia
  - Python
  - R
  - But now rather agnostic (.net anyone?)

Notebooks TNG

- Released in 2018
- Same building blocks
- Cool new interface!


File Edit View Run Kernel Tabs Settings Help
/ notebooks /
Name
audio
images
Altair.ipynb
Cpp.ipynb
Data.ipynb
Fasta.ipynb
Julia.ipynb
Linear Regression.ipynb
Lorenz.ipynb
lorenz.py
R.ipynb
untitled.dio
untitled1.dio
untitled2.dio
untitled3.dio
untitled4.dio
untitled5.dio
untitled6.dio
Markdown
Python 3
In Depth: Linear Regression
Just as naive Bayes (discussed earlier in In Depth: Naive Bayes Classification) is a good starting point for classification tasks, linear regression models are a good starting point for regression tasks. Such models are popular because they can be fit very quickly, and are very interpretable. You are probably familiar with the simplest form of a linear regression model (i.e., fitting a straight line to data) but such models can be extended to model more complicated data behavior.
In this section we will start with a quick intuitive walk-through of the mathematics behind this well-known problem, before seeing how before moving on to see how linear models can be generalized to account for more complicated patterns in data.
Slide Type
Raw NBConvert Format
Advanced Tools
Cell Metadata
Notebook Metadata
Launcher
Notebook
Python 3
C++11
C++14
C++17
Julia 1.1.0
phylogenetics (Python 3.7)
R
Console
Altair.ipynb
Output View
Seattle Weather: 2012-2015
Maximum Daily Temperature (C)
Date
weather
Number of Records
drizzle
fog
rain
snow
sun
Julia.ipynb
Julia
Lorenz.ipynb
python notebook
We explore the Lorenz system of differential equations:
Let's change (σ, β, ρ) with ipywidgets and examine the trajectories.
R.ipynb
R
Sepal.Length
Sepal.Width
Petal.Length
Python 3 | Idle
Mode: Command
Ln 1, Col 1
Lorenz.ipynb

jupyter

Install About Us Community Documentation NBViewer JupyterHub Widgets Blog

Try Jupyter

You can try Jupyter out right now, without installing anything. Select an example below and you will get a temporary Jupyter server just for you, running on mybinder.org. If you like it, you can install Jupyter yourself.

Try Classic Notebook

A tutorial introducing basic features of Jupyter notebooks and the

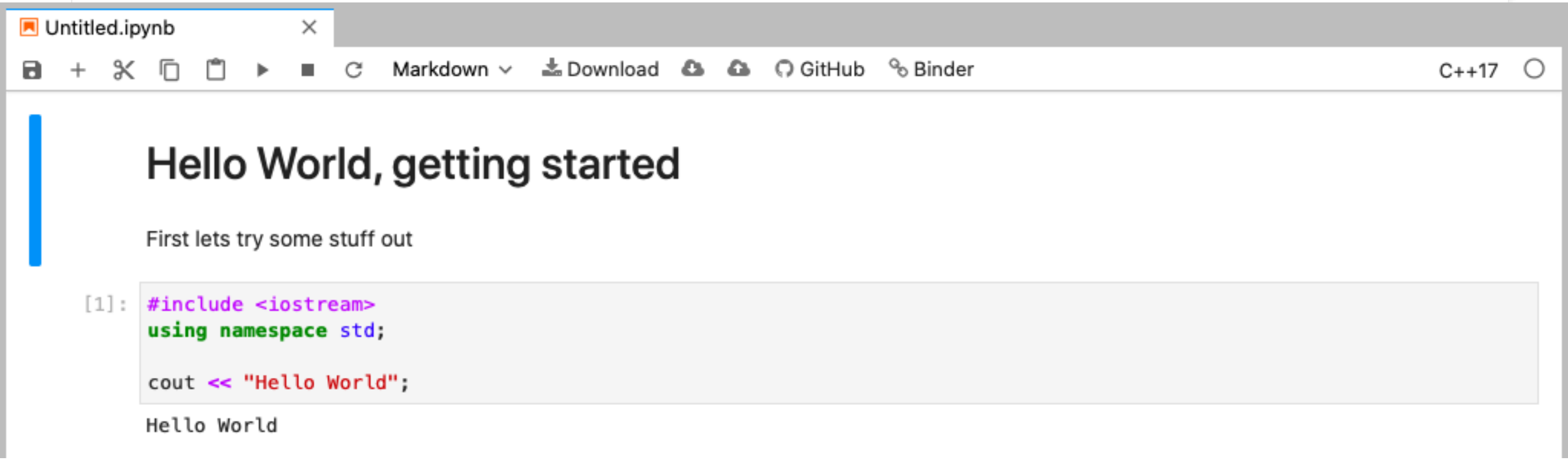
Try JupyterLab

JupyterLab is the new interface for Jupyter notebooks and is ready for

Try Jupyter with Julia

A basic example of using Jupyter with Julia.

https://mybinder.org/v2/gh/jupyterlab/jupyterlab-demo/try.jupyter.org?urlpath=lab


Untitled.ipynb
Markdown
Download
GitHub
Binder
C++17
Hello World, getting started
First lets try some stuff out
[1]:
#include <iostream>
using namespace std;
cout << "Hello World";
Hello World

Calculate Pi

[3]:
```
#include<iostream>
#include<iomanip>
#include<cmath>
#include<math.h>

double pi(int n)
{
    double sum = 0.0;
    int sign = 1;
    for (int i = 0; i < n; ++i)
    {
        sum += sign/(2.0*i+1.0);
        sign *= -1;
    }
    return 4.0*sum;
}
```

[7]:
```
cout << pi(1) << endl << pi(10) << endl << pi(100) << endl <<  pi(9999999) << endl;
```

```
4
3.04184
3.13159
3.14159
```

I want more!

First Some setup

- Follow along: https://gitlab.com/denner1/cialug-august-2020-jupyter-lab
- Initial setup on Ubuntu Fossa (20.04) Docker image
  - docker run -p 8080:8080 ubuntu:20.04
  - Or easy button to skip ahead
    ```
    docker run --rm -it -p 8080:8080 registry.gitlab.com/denner1/cialug-august-2020-jupyter-lab:latest
    ```
  - **Please note: this image is not optimized for size or security and is not production ready!**

Setup cont.

- First setup virtualenv and some prereqs
- apt-get install git r-base r-cran-devtools nodejs wget python3-pip build-essential libssl-dev libffi-dev python-dev python3-venv
- source venv/bin/activate
- Verify which env we are in with ‘which pip’ (also note hint in the prompt)

Why virtualenv

- Not really needed for docker
- Think of it as python in a bottle
- Other options are conda or just yell YOLO (or docker in our case)

Setup cont.

- pip install jupyterlab ipywidgets
- jupyter lab –ip=0.0.0.0 –port=8080 –allow root

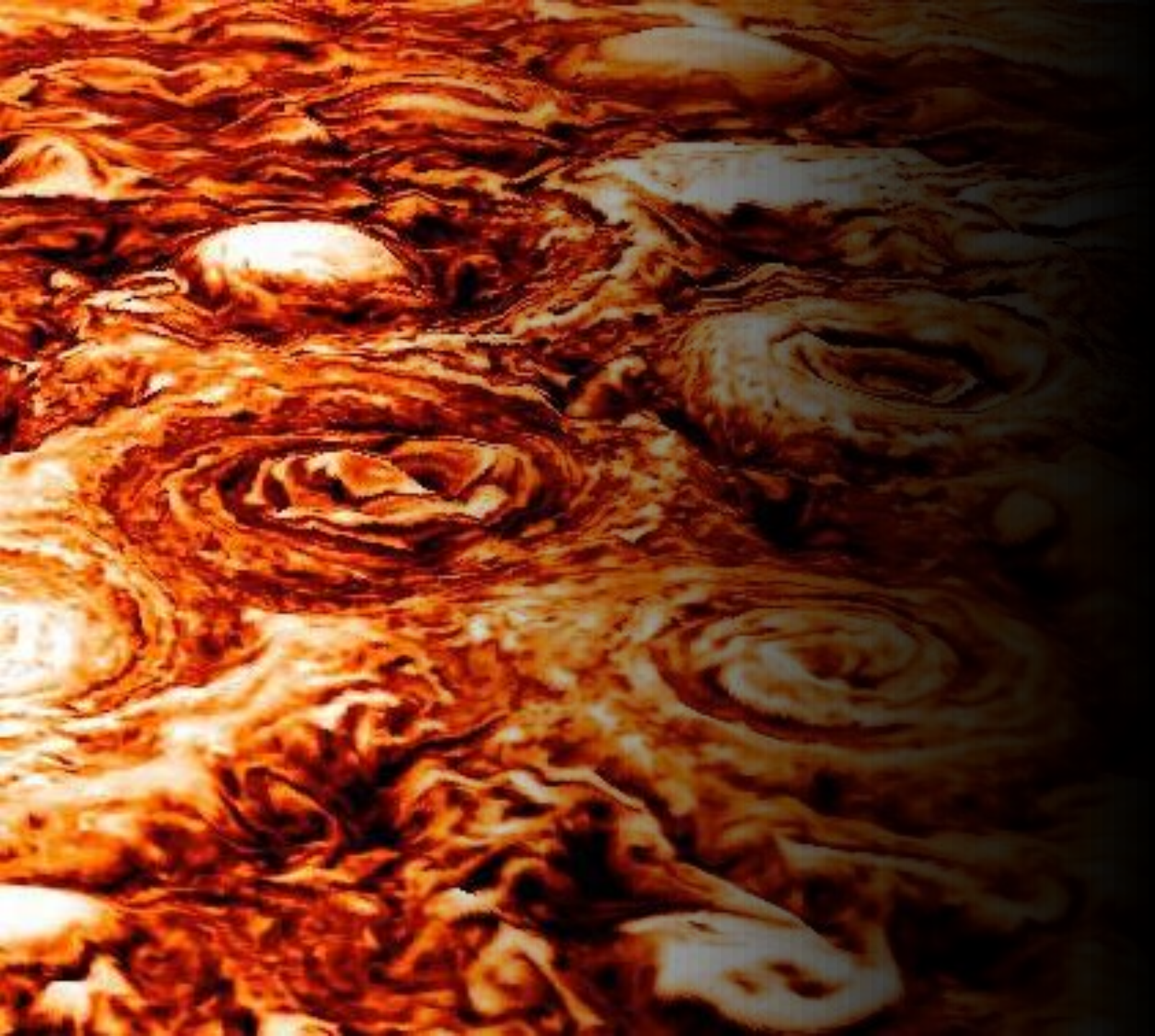
- Install R apt-get install r-base r-cran-devtools
- In R:
  - devtools::install_github("IRkernel/IRkernel")
  - IRkernel::installspec()
- GnuPlot
- apt install -y gnuplot
- pip install gnuplot_kernel
- python -m gnuplot_kernel install --user

Useful tips

- ! Will run in the shell eg.
  - !pip install packagename
- Tab completion is your friend
- help()
- quickref

- Lots of Kernels out there: https://github.com/jupyter/jupyter/wiki/Jupyter-kernels


Demo



Links

- https://plotly.com/python/ipython-notebook-tutorial/
- https://gist.github.com/frfahim/73c0fad6350332cef7a653bcd762f08d
- https://jupyter.org/try
- https://github.com/jupyter/jupyter/wiki/Jupyter-kernels