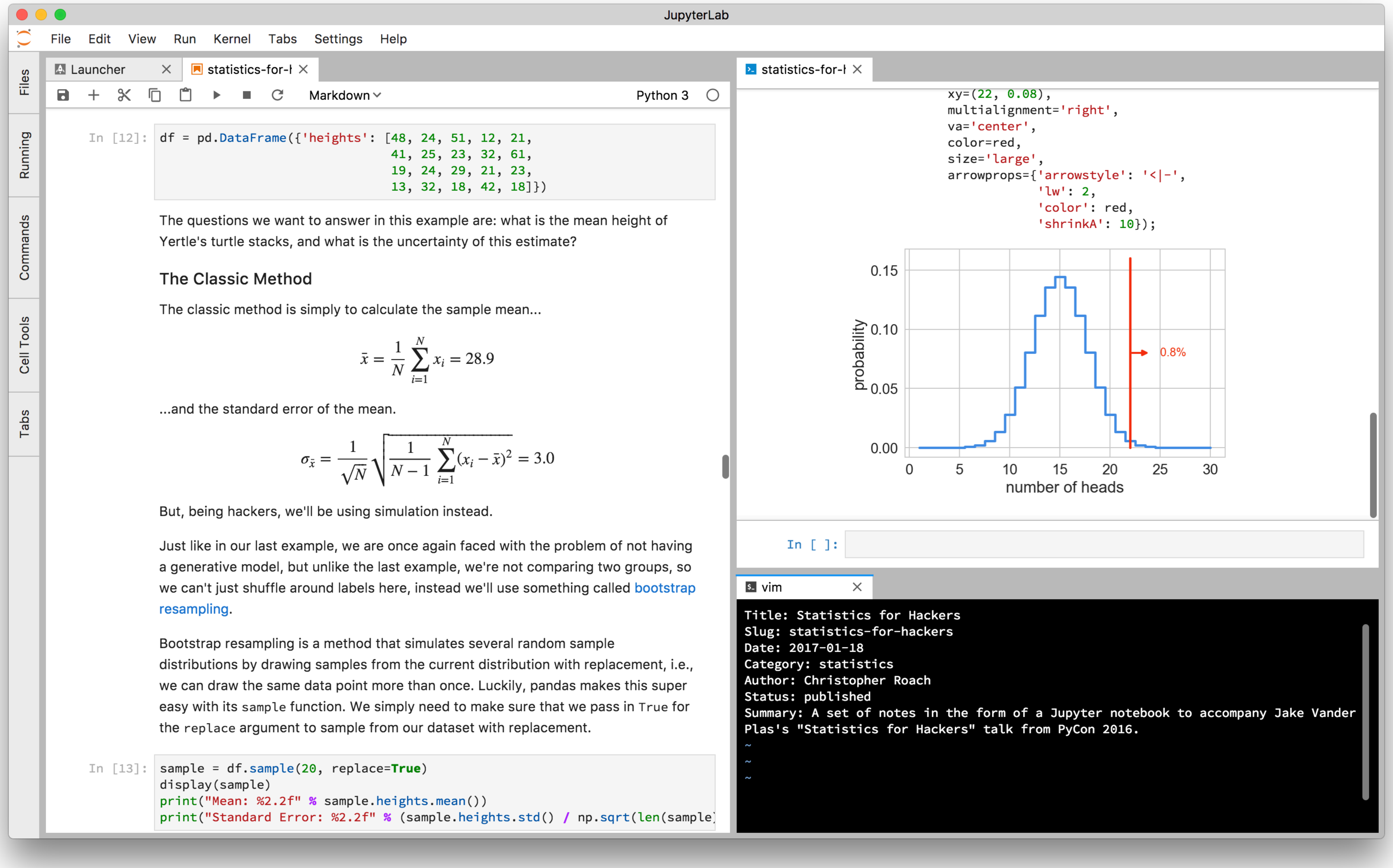
# Introduction to

# jupyter lab

Anh LP Nguyen
Date: 30 Aug 2018



```
In [12]: df = pd.DataFrame({'heights': [48, 24, 51, 12, 21,
                                        41, 25, 23, 32, 61,
                                        19, 24, 29, 21, 23,
                                        13, 32, 18, 42, 18]})
```

The questions we want to answer in this example are: what is the mean height of Yertle's turtle stacks, and what is the uncertainty of this estimate?

### The Classic Method

The classic method is simply to calculate the sample mean...

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i = 28.9$$

...and the standard error of the mean.

$$\sigma_{\bar{x}} = \frac{1}{\sqrt{N}}\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2} = 3.0$$

But, being hackers, we'll be using simulation instead.

Just like in our last example, we are once again faced with the problem of not having a generative model, but unlike the last example, we're not comparing two groups, so we can't just shuffle around labels here, instead we'll use something called bootstrap resampling.

Bootstrap resampling is a method that simulates several random sample distributions by drawing samples from the current distribution with replacement, i.e., we can draw the same data point more than once. Luckily, pandas makes this super easy with its `sample` function. We simply need to make sure that we pass in `True` for the `replace` argument to sample from our dataset with replacement.

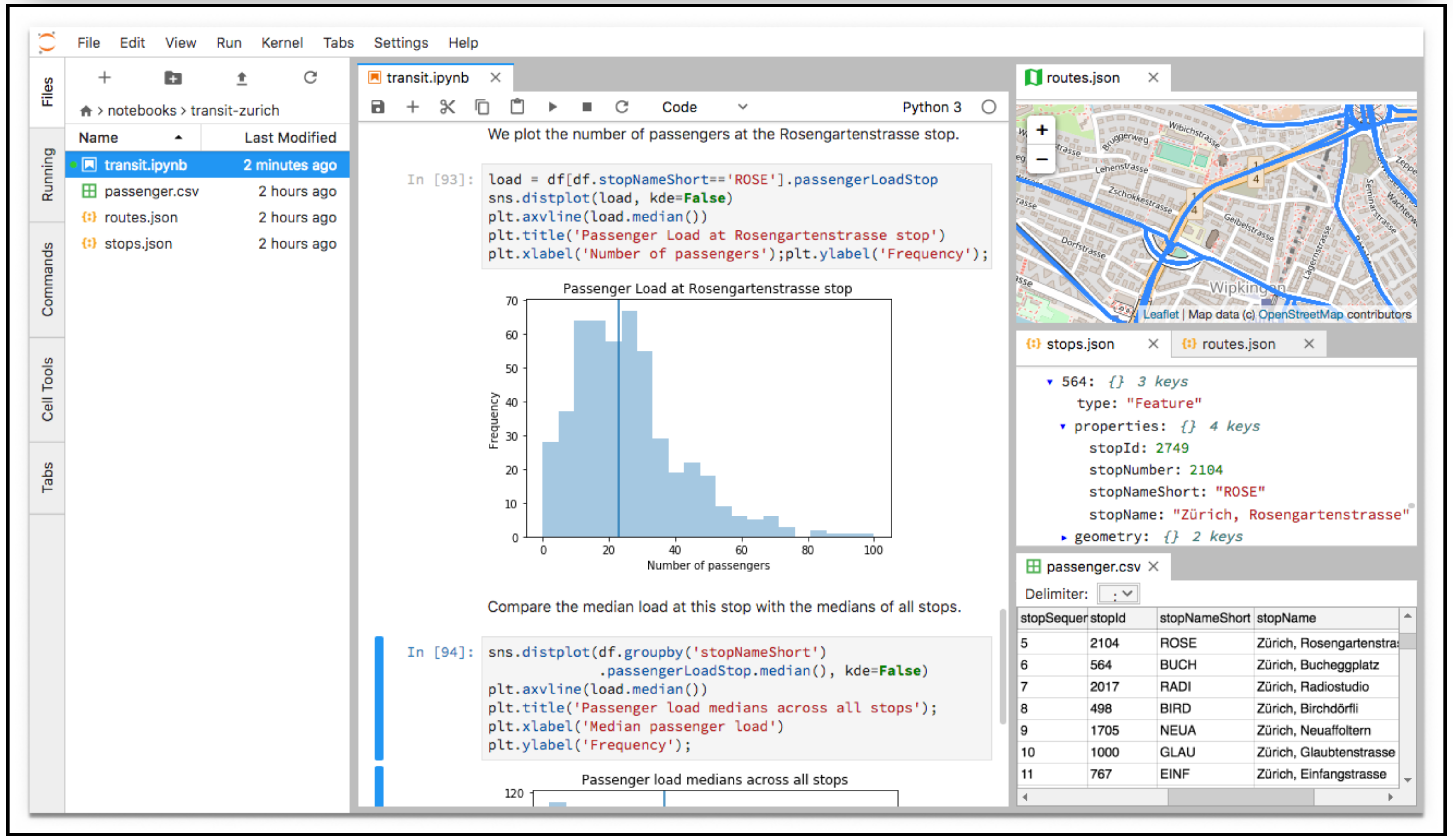
```
In [13]: sample = df.sample(20, replace=True)
         display(sample)
         print("Mean: %2.2f" % sample.heights.mean())
         print("Standard Error: %2.2f" % (sample.heights.std() / np.sqrt(len(sample)
```

```
             xy=(22, 0.08),
             multialignment='right',
             va='center',
             color=red,
             size='large',
             arrowprops={'arrowstyle': '<|-',
                         'lw': 2,
                         'color': red,
                         'shrinkA': 10});
```

```
In [ ]:
```

```
Title: Statistics for Hackers
Slug: statistics-for-hackers
Date: 2017-01-18
Category: statistics
Author: Christopher Roach
Status: published
Summary: A set of notes in the form of a Jupyter notebook to accompany Jake Vander
Plas's "Statistics for Hackers" talk from PyCon 2016.
~
~
~
```

**versus**


File Edit View Run Kernel Tabs Settings Help
Files
Running
Commands
Cell Tools
Tabs
transit.ipynb
Code
Python 3
Passenger Load at Rosengartenstrasse stop
Frequency
Number of passengers
Passenger load medians across all stops
routes.json
Wipkingen
Leaflet | Map data (c) OpenStreetMap contributors
stops.json
routes.json
passenger.csv

notebooks › transit-zurich

| Name | Last Modified |
|---|---|
| transit.ipynb | 2 minutes ago |
| passenger.csv | 2 hours ago |
| routes.json | 2 hours ago |
| stops.json | 2 hours ago |

We plot the number of passengers at the Rosengartenstrasse stop.

```
In [93]: load = df[df.stopNameShort=='ROSE'].passengerLoadStop
         sns.distplot(load, kde=False)
         plt.axvline(load.median())
         plt.title('Passenger Load at Rosengartenstrasse stop')
         plt.xlabel('Number of passengers');plt.ylabel('Frequency');
```

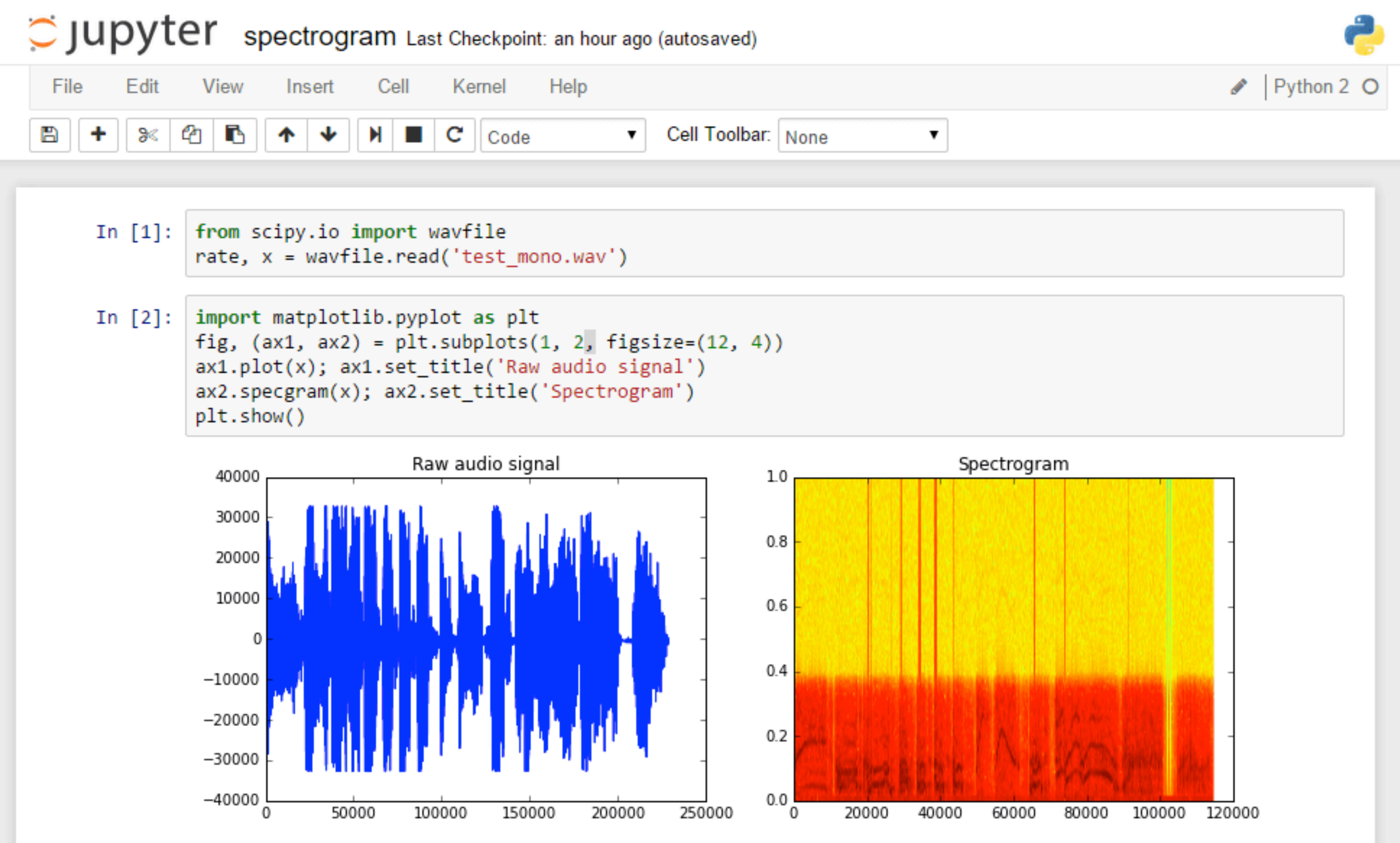
Compare the median load at this stop with the medians of all stops.

```
In [94]: sns.distplot(df.groupby('stopNameShort')
                      .passengerLoadStop.median(), kde=False)
         plt.axvline(load.median())
         plt.title('Passenger load medians across all stops');
         plt.xlabel('Median passenger load')
         plt.ylabel('Frequency');
```

```
▾ 564: {} 3 keys
    type: "Feature"
  ▾ properties: {} 4 keys
      stopId: 2749
      stopNumber: 2104
      stopNameShort: "ROSE"
      stopName: "Zürich, Rosengartenstrasse"
  ▸ geometry: {} 2 keys
```

Delimiter: ;

| stopSequer | stopId | stopNameShort | stopName |
|---|---|---|---|
| 5 | 2104 | ROSE | Zürich, Rosengartenstra: |
| 6 | 564 | BUCH | Zürich, Bucheggplatz |
| 7 | 2017 | RADI | Zürich, Radiostudio |
| 8 | 498 | BIRD | Zürich, Birchdörfli |
| 9 | 1705 | NEUA | Zürich, Neuaffoltern |
| 10 | 1000 | GLAU | Zürich, Glaubtenstrasse |
| 11 | 767 | EINF | Zürich, Einfangstrasse |



spectrogram Last Checkpoint: an hour ago (autosaved)

Cell Toolbar: None

```
In [1]: from scipy.io import wavfile
        rate, x = wavfile.read('test_mono.wav')
```

```
In [2]: import matplotlib.pyplot as plt
        fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
        ax1.plot(x); ax1.set_title('Raw audio signal')
        ax2.specgram(x); ax2.set_title('Spectrogram')
        plt.show()
```

# *Show time ...*

Jupyter Lab demo on Binder

# JupyterLab Interface

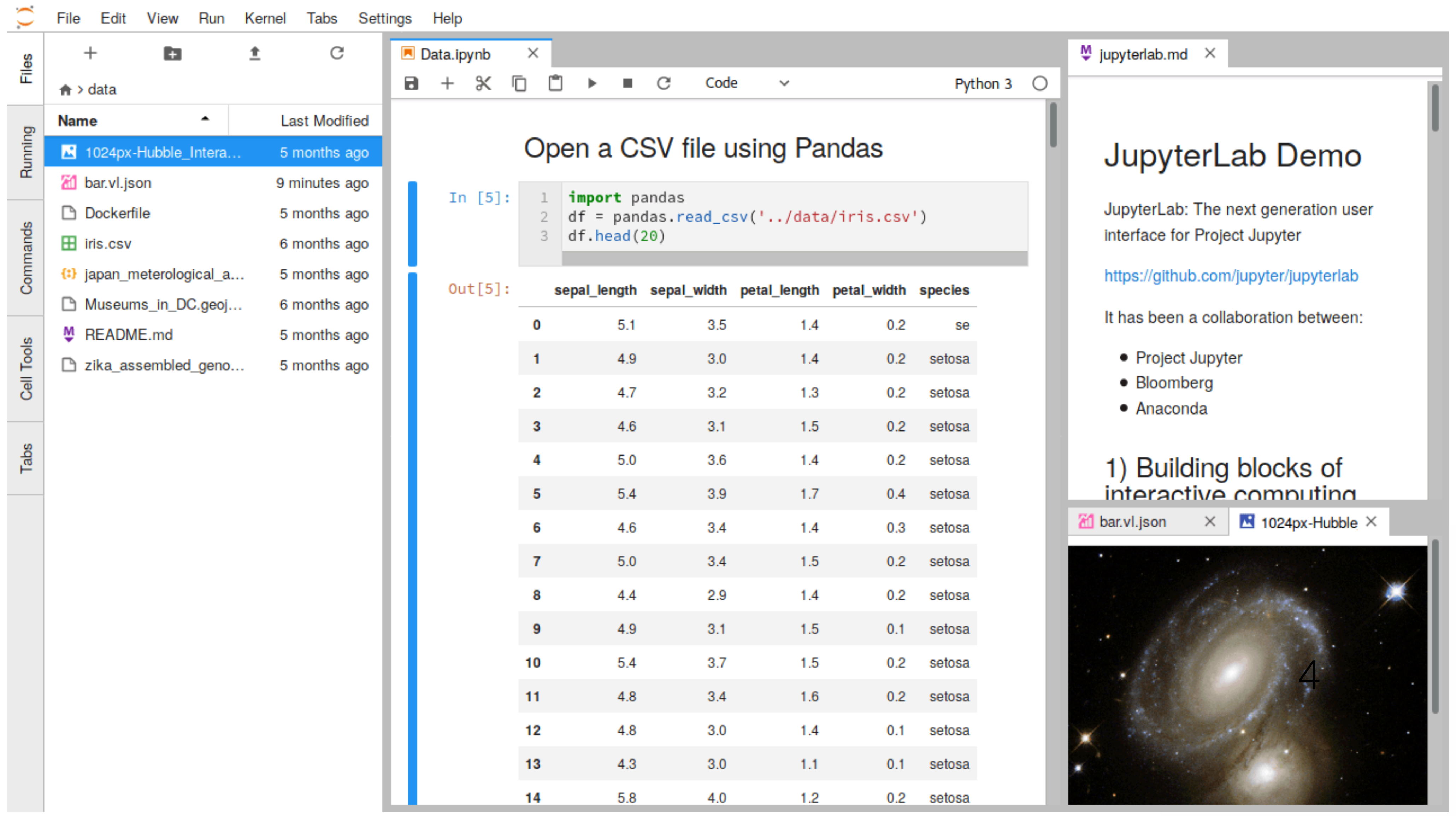
## The menu bar:

- *File* - working with files and directories
- *Edit* - working with editing documents and other activities
- *View* - adjusting the appearance of JupyterLab
- *Run* - Running code in different activities, eg. in notebooks or code consoles
- *Kernel* - Managing kernels
- *Tabs* - A list of the open documents and activities in the dock panel
- *Settings* - Common settings and an advanced setting editors
- *Help* - Help links to JupyterLab and kernel

To run code in the cell: SHIFT + ENTER

## The left sidebar:

- a file browser
- a list of running kernels and terminals
- a command palette
- a notebook cell tools inspector
- a tabs list



| Name | Last Modified |
|---|---|
| 1024px-Hubble_Intera… | 5 months ago |
| bar.vl.json | 9 minutes ago |
| Dockerfile | 5 months ago |
| iris.csv | 6 months ago |
| japan_meterological_a… | 5 months ago |
| Museums_in_DC.geoj… | 6 months ago |
| README.md | 5 months ago |
| zika_assembled_geno… | 5 months ago |

### Open a CSV file using Pandas

```
In [5]:
import pandas
df = pandas.read_csv('../data/iris.csv')
df.head(20)
```

Out[5]:

|  | sepal_length | sepal_width | petal_length | petal_width | species |
|---|---|---|---|---|---|
| 0 | 5.1 | 3.5 | 1.4 | 0.2 | se |
| 1 | 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 2 | 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 3 | 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 4 | 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| 5 | 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| 6 | 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| 7 | 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| 8 | 4.4 | 2.9 | 1.4 | 0.2 | setosa |
| 9 | 4.9 | 3.1 | 1.5 | 0.1 | setosa |
| 10 | 5.4 | 3.7 | 1.5 | 0.2 | setosa |
| 11 | 4.8 | 3.4 | 1.6 | 0.2 | setosa |
| 12 | 4.8 | 3.0 | 1.4 | 0.1 | setosa |
| 13 | 4.3 | 3.0 | 1.1 | 0.1 | setosa |
| 14 | 5.8 | 4.0 | 1.2 | 0.2 | setosa |

### JupyterLab Demo

JupyterLab: The next generation user interface for Project Jupyter

https://github.com/jupyter/jupyterlab

It has been a collaboration between:

- Project Jupyter
- Bloomberg
- Anaconda

### 1) Building blocks of interactive computing

# Supported File & Output Formats

- Markdown – `.md`
- Images – `.bmp, .gif, .jpeg, .jpg, .png, .svg`
- Delimiter-separated values – `.csv`
- JSON – `.json`
- HTML – `.html`
- LaTex – `.tex`
- PDF – `.pdf`
- Vega/Vega-Lite – `.vg, .vg,jason, .vl, .vl.jason`
- Virtual DOM – `.vdom, .json`

# Markdown

- A markup language
- Used in text cells in the Notebook
- A text file with extension `.md`
- Using Latex syntax to edit equations

Markdown Cheatsheet

# Markdown Cheatsheet

## List

```
1. Item 1
2. Item 2
3. Item 3
   * Item 3a
   * Item 3b

* or - or +
```

## Header

```
# Header 1
## Header 2
### Header 3
#### Header 4
##### Header 5

Header 1
========

Header 2
--------
```

## Emphasis

```
*This text is italic*
_This is also italic_

**This text is bold**
__This is also bold__

*You **can** combine them*

~~Strike through~~
```

## Task List

```
- [x] This is a complete item
- [ ] This is an incomplete item
- [x] List syntax
```

## Image

```
![Jupyter Logo](/images/logo.png)
```

## Links

```
[GitHub](http://github.com)
```

## Syntax Highlighting

~~~
```python
s = "Python Syntax Highlighting"
print s
```
~~~

## Blockquotes

```
Winston Churchill said:

> If you're going through hell,
> keep going
```

## Table

```
First Header | Second Header
------------ | -------------
Content 1    | Content 2
Content 3    | Content 4
Content 5    | Content 6
```

## Backslash Escapes

```
\*literal asterisks\*
\ ` * _ {} [] () # + - . ! >
```

## Footnote

```
Here's a sentence with a
footnote.[^1]

[^1]: This is the footnote
```

# Images

| Short cut | Doing |
|---|---|
| + / − | zooming the image |
| [ / ] | rotating the image |
| H / V | flipping the image horizontally and vertically |
| I | inverting the colors |
| 0 | resetting the image |

# Installation of Jupyter Lab

**Requirement:** Jupyter Notebook (installation) version 4.3 or later

```
jupyter notebook --version
```

**Using Jupyter Notebook Version 5.3 or earlier**

```
jupyter serverextension enable --py jupyterlab --sys-prefix
```

**Using conda**

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

**Using pip**

```
pip install jupyterlab
```

**Supported Browser**

The latest version of

◉ Firefox

◉ Chrome

◉ Safari

**Running the Lab**

```
jupyter lab
```

# Installation of Jupyter Notebook

**Requirement:** Python 3.3 or 2.7

**Using Anaconda**

1. Download Anaconda's latest Python 3 version

2. Following the instructions on the download page

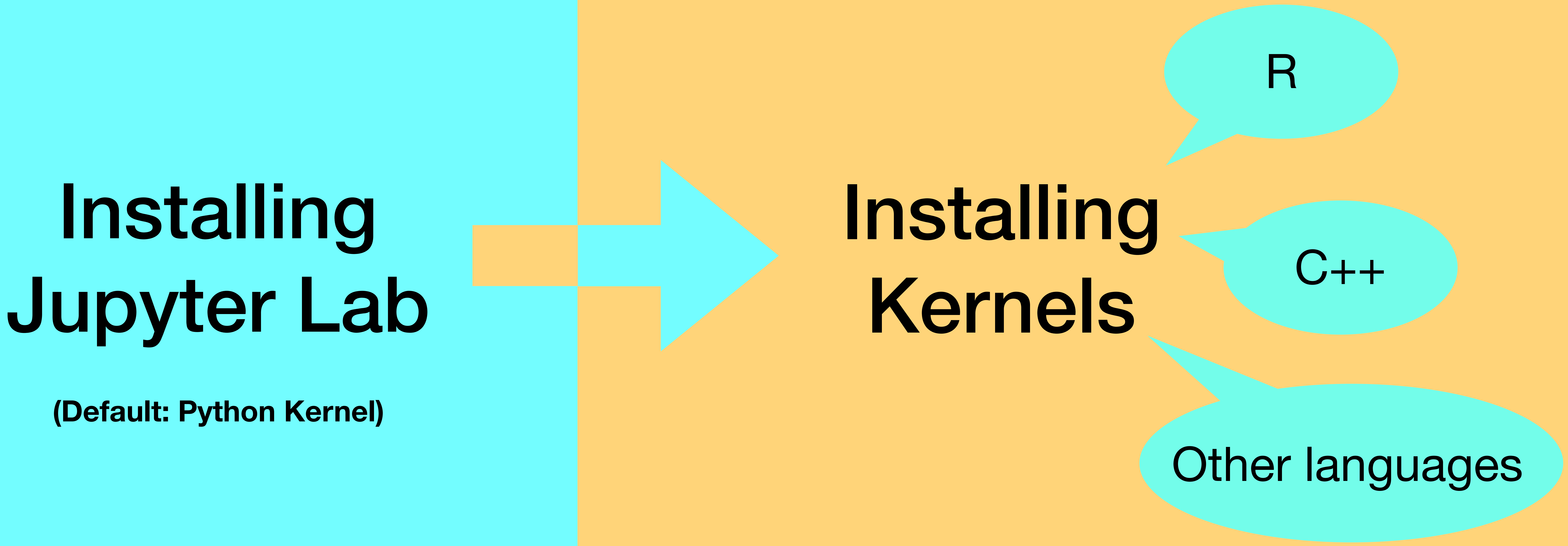
**Using pip**

```
pip3 install --upgrade pip

pip3 install jupyter
```

**Running notebook**

```
jupyter notebook
```



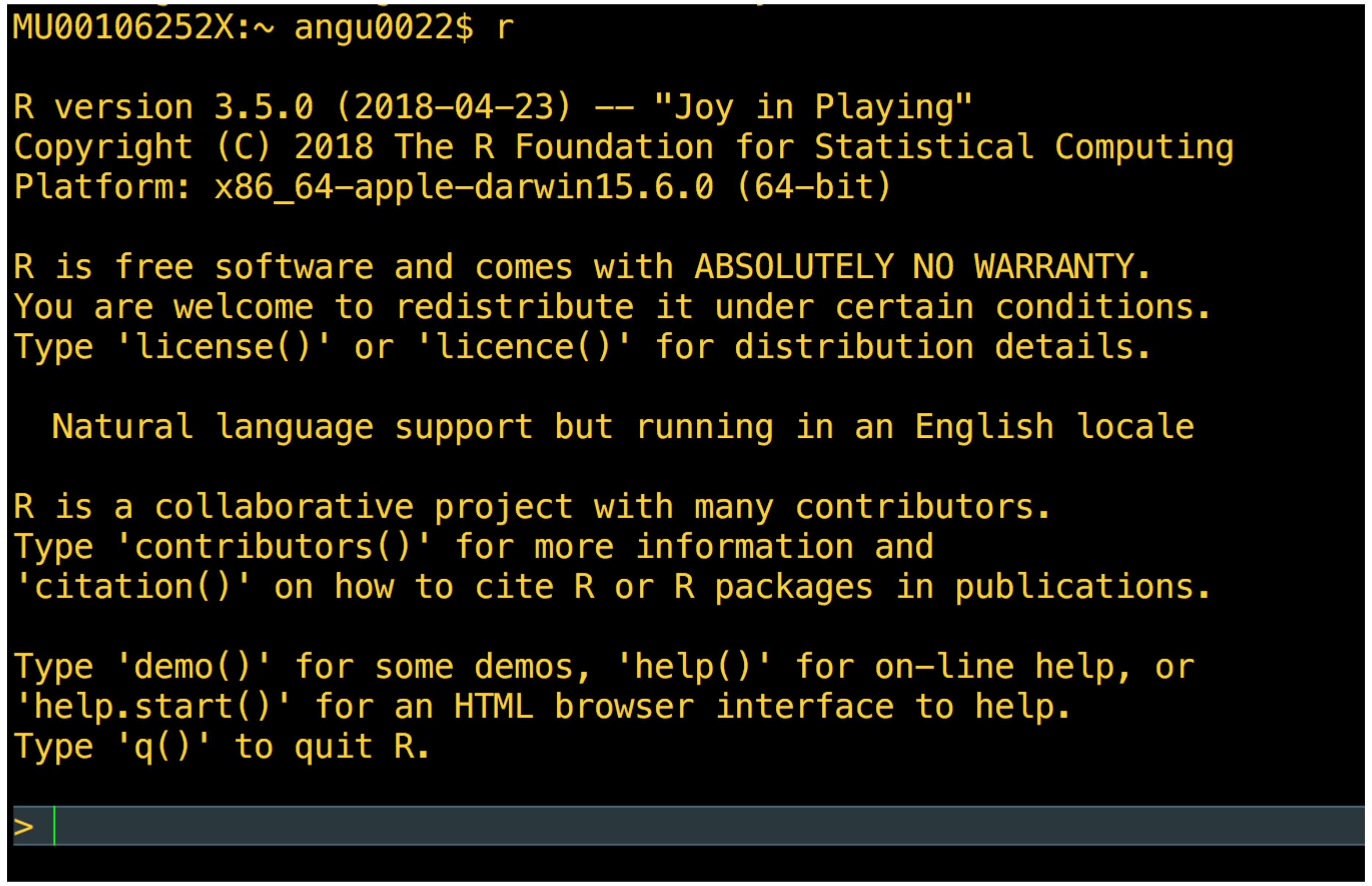
# Installing Jupyter Lab

(Default: Python Kernel)

# Installing Kernels

# Installation of R kernel

## Starting R console



```
MU00106252X:~ angu0022$ r

R version 3.5.0 (2018-04-23) -- "Joy in Playing"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 
```

```
install.packages(c('repr', 'IRdisplay', 'evaluate', 'crayon', 'pbdZMQ', 'devtools', 'uuid', 'digest'))
```

```
devtools::install_github('IRkernel/IRkernel')
```

```
IRkernel::installspec()
```

# Installation of C++ kernel

Install with Miniconda

```
conda create -n xeus python=3.6 jupyterlab -c conda-forge
```

```
source activate xeus
```

```
conda install xeus-cling -c QuantStack -c conda-forge
```

Everytime you want to use C++ kernel, run this

```
source activate xeus
```

before calling your lab

```
jupyter lab
```

https://xeus-cling.readthedocs.io/en/latest/