

# jupyter\_tour

February 16, 2018

## 1 Jupyter notebooks

### 1.1 an interactive environment for writing code and markdown

### 1.2 Where can you try it and follow along?

Visit [this page on jupyter's website](https://try.jupyter.org/) to play around without having to install anything.

<https://try.jupyter.org/>

If you like it, see the installation instructions at the end of the notebook!

I'll use the iris dataset for examples, as that is available in the online demo version.

How does it work?

- There are two types of cells, markdown and code
- We type explanations or instructions into the markdown cells (here).
- And we type R code into the code cells. (it supports lots of programming languages!)
- The code can then be run, and the markdown rendered.

```
In [2]: x = 3+4  
        x*2
```

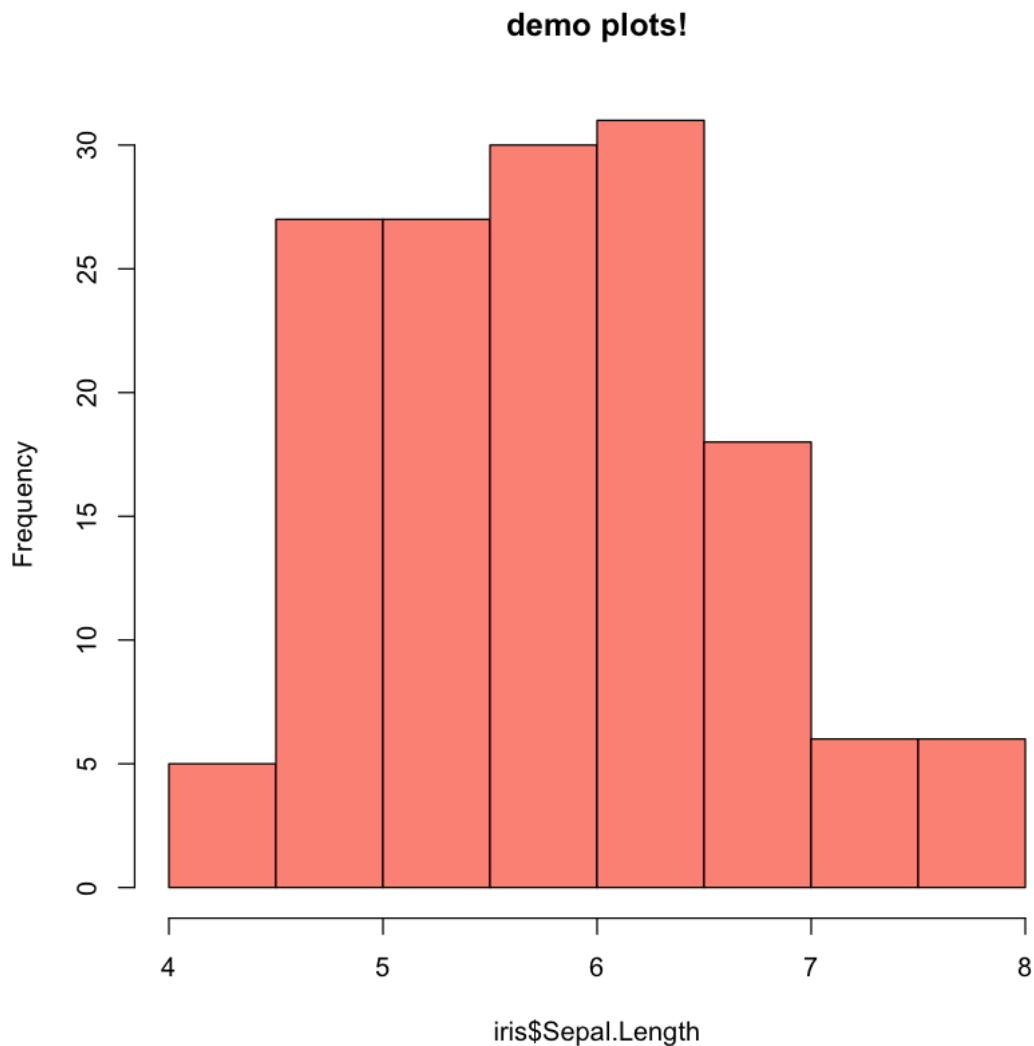
14

```
In [1]: head(iris)
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa

It pretty prints tables into an easily readable format.

```
In [4]: hist(iris$Sepal.Length, main='demo plots!', col='salmon')
```



```
In [7]: just_one_species = iris$Sepal.Length[iris$Species == 'setosa']
        just_one_species
```

```
1. 5.1 2. 4.9 3. 4.7 4. 4.6 5. 5. 6. 5.4 7. 4.6 8. 5. 9. 4.4 10. 4.9 11. 5.4 12. 4.8 13. 4.8 14. 4.3 15. 5.8 16. 5.7
17. 5.4 18. 5.1 19. 5.7 20. 5.1 21. 5.4 22. 5.1 23. 4.6 24. 5.1 25. 4.8 26. 5. 27. 5. 28. 5.2 29. 5.2 30. 4.7 31. 4.8
32. 5.4 33. 5.2 34. 5.5 35. 4.9 36. 5. 37. 5.5 38. 4.9 39. 4.4 40. 5.1 41. 5. 42. 4.5 43. 4.4 44. 5. 45. 5.1 46. 4.8
47. 5.1 48. 4.6 49. 5.3 50. 5
```

```
In [8]: t.test(just_one_species, mu=10, alternative="less")
```

One Sample t-test

```
data: just_one_species
t = -100.18, df = 49, p-value < 2.2e-16
alternative hypothesis: true mean is less than 10
95 percent confidence interval:
    -Inf 5.089575
sample estimates:
mean of x
    5.006
```

Plots pop up in line with the code, so it is easy to show which inputs relate to which outputs

### 1.3 Outputs are saved!

If I work in the notebook and save it, I can then send it to people and they can see the outputs. The output graphs and tables don't disappear when I close the notebook. This feature is great for sharing results.

### 1.4 Reproducibility

```
In [5]: x = 1738
```

Note the number in the cell next to the code. It shows you the order I ran the code, so you can see when I backtrack.

### 1.5 Sharing with others

I can share a notebook in several formats:

- I can send it to the as an .ipynb (jupyter) file. Or I can post these on github and it automatically renders them (meta)
- In the top left under the file header I can use Download as to save the notebook as a plain R file, a pdf, a latex file, or a markdown file

#### 1.5.1 You don't have as much control with the format/aesthetics as r markdown, but the files are easy to make!

#### 1.5.2 Note you can save to different formats from the command line as well

- I can use the following command to isolate just the R code (in terminal)

```
jupyter nbconvert jupyter_tour.ipynb --to script
```

- I can make a .pdf of the notebook with the following command (in terminal)

```
jupyter nbconvert jupyter_tour.ipynb --to pdf
```

### 1.5.3 I use jupyter notebooks to:

- share tutorials/code for some of the classes I TA (better for code than powerpoint slides)
  - I can run code in front of students, and send out a pdf copy for later reference
- write blog posts
- write tutorials (meta!)

## 1.6 Misc. Other features

- The help function pops up the message on the bottom of your screen!

In [6]: ?plot

- Cells can be moved around easily, so editing is simple
- shift + enter is the shortcut for running the selected cell
- you can use the arrow keys to move from inside one cell to another
- you can also use the run button at the top.
- to clear the output of all cells use the Kernel

## 1.7 Installation Instructions for jupyter notebooks with r support

So have I convinced you? If so here are the download instructions.

I've tried to come up with an optimal between simplicity and not forcing you to download way more things than you need. This is the bare bones jupyter setup. Process it to download a python package manager, then set up jupyter notebook and install the r version.

### 1.7.1 Step 1. Download miniconda for python 3.6

This is found here: <https://conda.io/miniconda.html> Get the version corresponding to your operating system.

### 1.7.2 Step 2. Install miniconda

Once you download it, follow the setup instructions. ##### WINDOWS IMPORTANT NOTE: during the installation select the 'add conda to your \$PATH option' or else you can't use it from command prompt.

### 1.7.3 Step 3. installing jupyter and jupyter's r support

1. MAC: open terminal (it is in your applications/utilities).  
WINDOWS: open command prompt (just search for it).
2. Run the following three commands in terminal/command prompt.

```
conda install jupyter
```

```
conda install -c r r
```

```
conda install -c r r-irkernel
```

#### 1.7.4 Step 4. running jupyter

If that all worked it should now be installed! To start the notebook do the following.

**Mac** In a terminal type the following:

```
jupyter notebook
```

**PC** In command prompt type the following:

```
python -m notebook
```

Now your web browser will open and you can open a new jupyter notebook! You can navigate around you computer and make a notebook. Note the working directory for a notebook is by default where it is saved.