

Lab meeting 16/04/2020

---

# CREATING JUPYTER NOTEBOOKS IN VIRTUAL ENVIRONMENTS

Bernat Bramon Mora

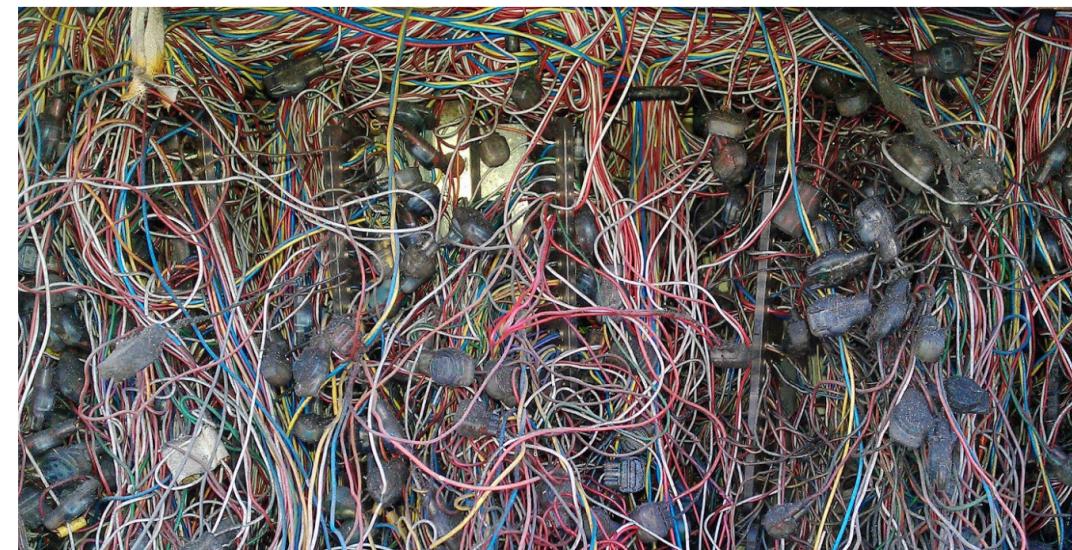
ETH Zürich  
Plant Ecology group

# 5 reasons why jupyter notebooks suck



Alexander Mueller [Follow](#)

Mar 24, 2018 · 3 min read ★



## Reproducing Watts and Strogatz ideas

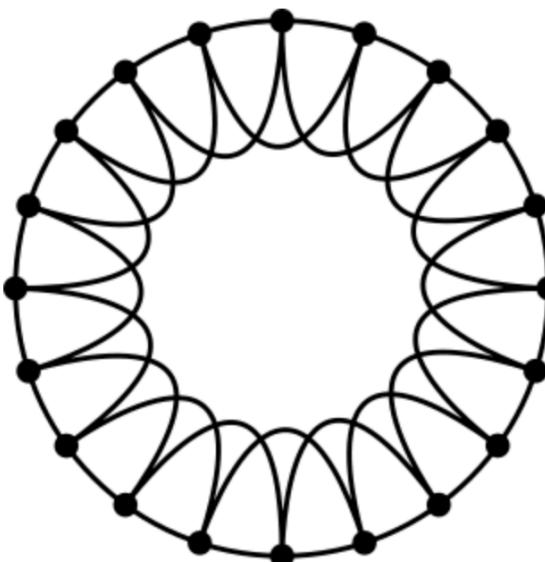
We've all said at some point "it's such a small world!" when facing an unexpected connection—a friend or a relative in common—with someone you just met. It just feels like an incredible coincidence! In their paper, Watts and Strogatz try to quantify that.

The authors show that some networked systems showcase what they call "small-worlds" features. These involve a short path length and high clustering coefficient. In plain terms, what this means is that some systems have their components very well connected to their close neighbors while not too far from distant ones. If you got completely lost with the jargon of the paper, [Veritasium and Neil deGrasse Tyson](#) have a nice video to illustrate the core idea.

Here, we will reproduce the main result from Watts and Strogatz. We will build a clustered network that we will progressively randomize to generate a gradient from "structured" to "randomized". Then, we will use network properties to show that there is a region in this gradient that shows "small-world" properties.

### Ring Lattice

Let's start by building the following network that the authors call "Ring Lattice" in the paper:



In this network, everyone is connected to their nearest neighbors. Can you imagine how this would look like in the form of an adjacency matrix? In this case, the network has  $N=20$  nodes, and  $K=4$  edges or links per node.

## Basic Shell Commands

1. acroread - Read or print a PDF file.

2. cat - Send a file to the screen in one go. Useful for piping to other programs

```
cat file1          # list file1 to screen
cat file1 file2 file3 > outfile # add files together into outfile
cat *.txt > outfile      # add all .txt files together
cat file1 file2 | grep fred # pipe files
```

3. cc - Compile a C program

```
cc test1.c          # compile test1.c to a.out
cc -O2 -o test2.prog test2.c # compile test2.c to test2.prog
```

4. cd - Change current directory

```
cd                  # go to home directory
cd ~/papers         # go to /home/user/papers
cd ~fred            # go to /home/fred
cd dir              # go to directory (relative)
cd /dir1/dir2/dir3... # go to directory (absolute)
cd -                # go to last directory you were in
```

5. cp - Copy file(s)

```
cp file1 file2          # copy file1 to file2
cp file1 directory       # copy file1 into directory
cp file1 file2 file3 ... directory # copy files into directory
cp -R dir1 dir2/ # copy dir1 into dir2 including subdirectories
cp -pR dir1 dir2/ # copy directory, preserving permissions
```

6. date - Shows current date

```
> date
Sat Aug 31 17:18:53 BST 2002
```

• • •

## Basic Shell Commands

1. acroread - Read or print a PDF file.

2. cat - Send a file to the screen in one go. Useful for piping to other programs

```
cat file1          # list file1 to screen
cat file1 file2 file3 > outfile # add files together into outfile
cat *.txt > outfile      # add all .txt files together
cat file1 file2 | grep fred # pipe files
```

3. cc - Compile a C program

```
cc test1.c          # compile test1.c to a.out
cc -O2 -o test2.prog test2.c # compile test2.c to test2.prog
```

4. cd - Change current directory

```
cd                  # go to home directory
cd ~/papers         # go to /home/user/papers
cd ~fred            # go to /home/fred
cd dir              # go to directory (relative)
cd /dir1/dir2/dir3... # go to directory (absolute)
cd -                # go to last directory you were in
```

5. cp - Copy file(s)

```
cp file1 file2          # copy file1 to file2
cp file1 directory       # copy file1 into directory
cp file1 file2 file3 ... directory # copy files into directory
cp -R dir1 dir2/ # copy dir1 into dir2 including subdirectories
cp -pR dir1 dir2/ # copy directory, preserving permissions
```

6. date - Shows current date

```
> date
Sat Aug 31 17:18:53 BST 2002
```

• • •



The basics: installing anaconda, jupyter and r-essentials

# The basics: installing anaconda, jupyter and r-essentials

## Anaconda Individual Edition

The World's Most Popular Python/R Data Science Platform

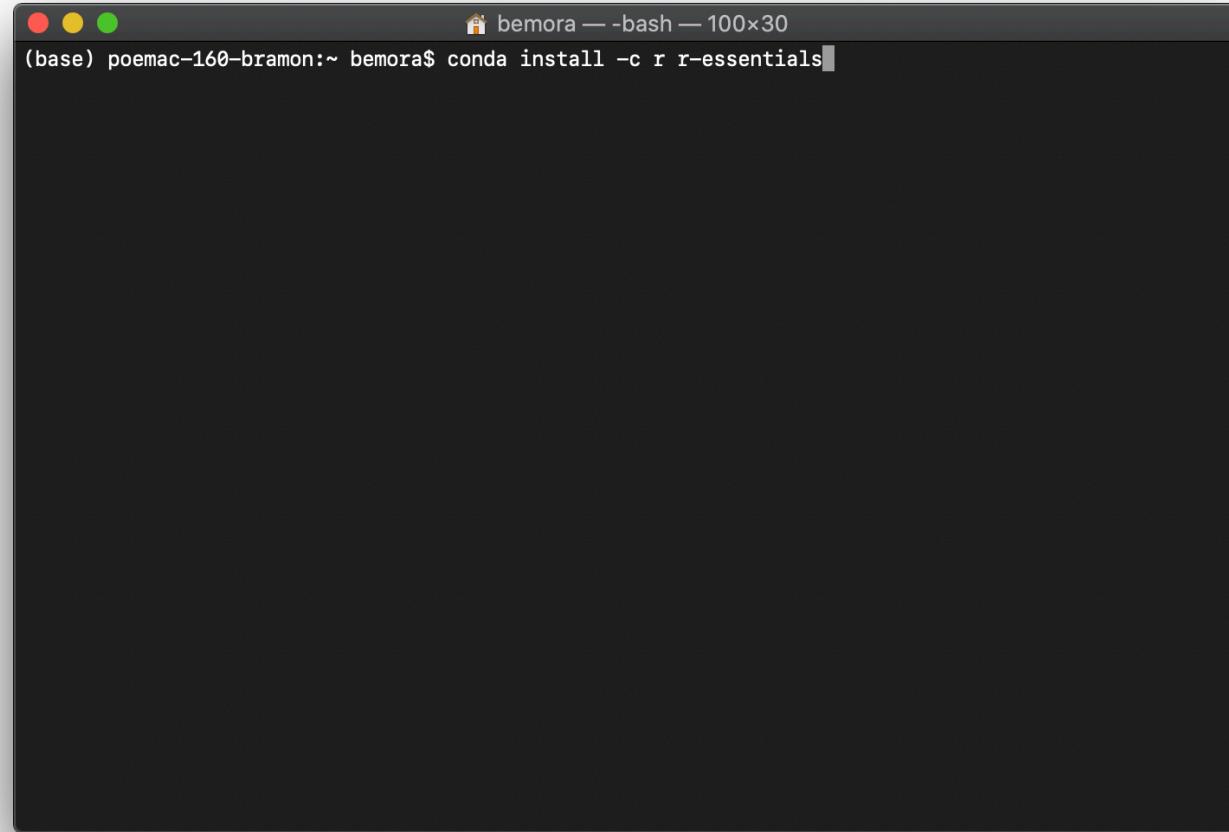
[Download](#)

The open-source [Anaconda Individual Edition](#) (formerly Anaconda Distribution) is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 19 million users worldwide, it is the industry standard for developing, testing, and training on a single machine, enabling *individual data scientists* to:

- Quickly download 7,500+ Python/R data science packages
- Manage libraries, dependencies, and environments with [Conda](#)
- Develop and train machine learning and deep learning models with [scikit-learn](#), [TensorFlow](#), and [Theano](#)
- Analyze data with scalability and performance with [Dask](#), [NumPy](#), [pandas](#), and [Numba](#)
- Visualize results with [Matplotlib](#), [Bokeh](#), [Datashader](#), and [Holoviews](#)



# The basics: installing anaconda, jupyter and r-essentials

A screenshot of a dark gray terminal window titled "bemora — bash — 100x30". The window has three colored circular icons in the top-left corner. The title bar shows the path "(base) poemac-160-bramon:~" followed by the command "bemora\$ conda install -c r r-essentials". The main body of the terminal is completely blank, indicating that the command has not yet been executed.

```
(base) poemac-160-bramon:~ bemora$ conda install -c r r-essentials
```

Type “jupyter notebook” in a terminal to start a notebook

# Create a Github repository for your lecture

The image shows a screenshot of the GitHub homepage. At the top, there is a navigation bar with links for "Why GitHub?", "Team", "Enterprise", "Explore", "Marketplace", and "Pricing". On the right side of the navigation bar are "Search GitHub", "Sign in", and "Sign up" buttons. The main content area features a large, bold title "Built for developers" and a subtitle explaining GitHub's purpose: "GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 40 million developers." To the right of this text is a sign-up form. The form consists of three input fields: "Username", "Email", and "Password". Below these fields is a note: "Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)". At the bottom of the form is a large green "Sign up for GitHub" button. Below the button, a small note states: "By clicking 'Sign up for GitHub', you agree to our [Terms of Service](#) and [Privacy Statement](#). We'll occasionally send you account related emails."

Why GitHub? ▾ Team Enterprise Explore ▾ Marketplace Pricing ▾

Search GitHub / Sign in Sign up

# Built for developers

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 40 million developers.

Username

Email

Password

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [Terms of Service](#) and [Privacy Statement](#). We'll occasionally send you account related emails.

# Create a Github repository for your lecture



A photograph of a person in a black wetsuit sailing a small sailboat on a body of water. The sail is light blue and has some text on it. The person is lying on their stomach on the board.

Set status

**Bernat Bramon Mora**  
bernibra

Edit profile

A physicist at heart, a biologist in the books, and data-scientist on LinkedIn.

ETH Zurich

Overview    **Repositories 29**    Projects 0    Packages 0    Stars 2    Followers 7    Following 6

Find a repository...    Type: All ▾    Language: All ▾    **New**

**jupyter-lab-meeting**  
This is a short tutorial on how to use and create jupyter notebooks  
Updated 3 minutes ago

**Quantitative-Approaches-practical**  
A notebook to reproduce Watts and Strogatz (1998) and apply to new ecological data  
Jupyter Notebook    Updated 5 hours ago

**extract-climate**  
This is just an example on how to extract climatic variables for a set of sites  
R    Updated 9 hours ago

**New**

Click on “New” to create a repository

Star

Star

Star

# Create a Github repository for your lecture

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

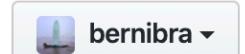
[Import a repository.](#)

### Repository template

Start your repository with a template repository's contents.

No template ▾

Owner



bernibra

Repository name \*

jupyter-lab-meeting



Name of the repository

Great repository names are short and memorable. Need inspiration? How about [bug-free-octo-fortnight](#)?

Description (optional)



Public

Anyone can see this repository. You choose who can commit.



Make it public if you want other people to access it



Private

You choose who can see and commit to this repository.

# Create a Github repository for your lecture

The screenshot shows a GitHub repository page for `bernibra/jupyter-lab-meeting`. The repository has 1 commit, 1 branch, 0 packages, 0 releases, and 1 contributor. The 'Code' tab is selected. A tooltip for the 'Clone with HTTPS' link is shown, with a red arrow pointing to it and a red box containing the text "copy the address of your repository".

No description, website, or topics provided.

Manage topics

1 commit 1 branch 0 packages 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

bernibra Create README.md

README.md Create README.md

README.md

jupyter-lab-meeting

Clone with HTTPS Use SSH  
Use Git or checkout with SVN using the web URL.  
<https://github.com/bernibra/jupyter-lab-meeting>

Open in Desktop Download ZIP

copy the address of your repository

# Create a Github repository for your lecture

```
[(base) poemac-160-bramon:~ bemora$ cd Desktop/  
(base) poemac-160-bramon:Desktop bemora$ git clone https://github.com/bernibra/jupyter-lab-meeting.git]
```

Use the shell commands to go to your preferred directory and clone the repository

Let's just try to create one...

# Using Binder to put your notebooks in virtual environments



Turn a Git repo into a collection of interactive notebooks

Have a repository full of Jupyter notebooks? With Binder, open those notebooks in an executable environment, making your code immediately reproducible by anyone, anywhere.

Build and launch a repository

GitHub repository name or URL

GitHub ▾ GitHub repository name or URL

Git branch, tag, or commit Path to a notebook file (optional)

Git branch, tag, or commit Path to a notebook file (optional) File ▾ launch

Copy the URL below and share your Binder with others:

Fill in the fields to see a URL for sharing your Binder.

# Using Binder to put your notebooks in virtual environments

## Specifying an R environment with a runtime.txt file

Jupyter+R: [launch binder](#)

RStudio: [launch binder](#)

RShiny: [launch binder](#)

Binder supports using R and RStudio, with libraries pinned to a specific snapshot on [MRAN](#).

You need to have a `runtime.txt` file that is formatted like:

```
r-<YYYY>-<MM>-<DD>
```

where YYYY-MM-DD is a snapshot at MRAN that will be used for installing libraries. In this line, you can request a [specific version of R](#). To do this list the version between the 'r' and the year, as in `r-3.6-2019-09-24`. Right now the default version of R is 3.6.

You also need a Python notebook file such as [this one](#).

You can also have an `install.R` file that will be executed during build, and can be used to install libraries.

Both [RStudio](#) and [IRKernel](#) are installed by default, so you can use either the Jupyter notebook interface or the RStudio interface.

This repository also contains an example of a Shiny app.

Last, note that if your Binder URL points to a folder, as in

<http://mybinder.org/v2/gh/binder-examples/r/master?urlpath=shiny/bus-dashboard/>,