

Software Setup Session



Advanced Methods in Applied Statistics, vol. 2021



UNIVERSITY OF
COPENHAGEN

Tania Kozynets
tetiana.kozynets@nbi.ku.dk
February 4, 2021

Hello, world & outline

> Welcome to AMAS-2021!

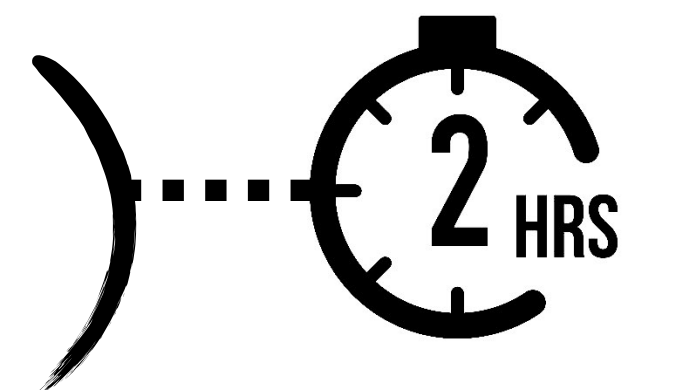
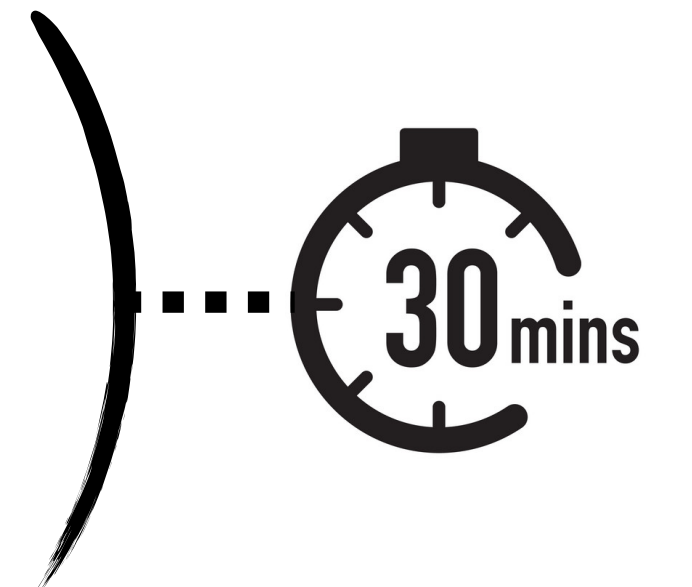
> Today we will cover:

(1) what software would be best to use for the course;

(2) how to install that software;

(3) specific packages that I'd recommend to have ready;

(4) any issues you might have when installing these – I encourage everyone to try now and tell me what's not working.



The platforms and the languages

- > Most generally, feel free to use any OS/programming language that works for you;
- > However, most “example code” from previous years is in Python, and I’m more likely to help you **today** if you’re on MacOS/Linux;
- > Going forward, conceptual statistics questions can still be helped with even if you run Matlab on Windows :)
- > To install/launch programs, using Terminal (MacOS/Linux) or Command Prompt (Windows) will be the way, except when there’s a dedicated installer program to download.

your TA



- > **Platform:**
MacOS Catalina 10.15.5;
MacOS Sierra 10.12.6;
access to Linux machines.
- > **Software:**
Python 3.8.5, 3.8.3, 3.7.6,
2.7.16...
- > **Anything else?**

I do most things from
the terminal & Jupyter;

I’ve never used R and
have some *very* rusty
knowledge of Matlab.

Briefly about Python

> For those of us choosing to program in Python, make sure it's Python 3.5+ ;

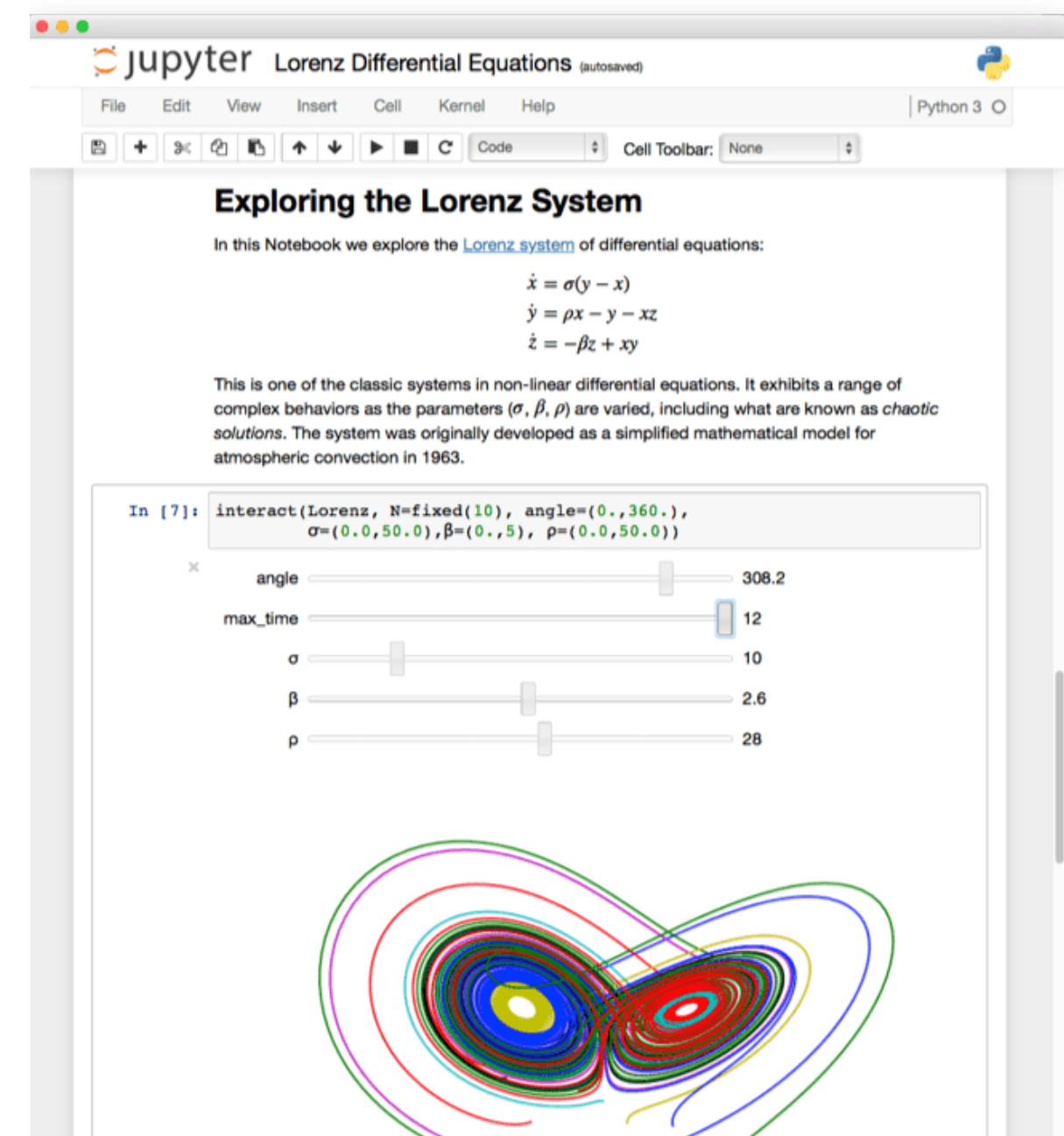
```
python --version
```

in your Terminal/Command Prompt window

> In principle, any Python environment should work for code development, however...

> Using a [Jupyter Notebook](#) would be the easiest.

In previous years, TAs provided their example solutions to exercises as Jupyter Notebooks, and so will I.



Briefly about virtual environments for Python

“A virtual environment is a tool that helps to keep dependencies required by different projects separate”

- > I'd recommend to treat this course as a separate project and create a new Python environment;
- > In that environment, you can install the Jupyter Notebook and all the packages we will need;
- > This way, we will make sure that we will not mess with any other versions of the same packages you have for different projects, and that those other versions will not mess with us.

This is optional but will likely make our lives easier.

Skip to slide 9 if you're happy with your environment.

Creating your own conda environment*

Step 0. [Install Anaconda](#).

Step 1. Create a new environment:

```
conda create --name AMAS
```

name of your choice



Step 2. Activate your environment:

```
conda activate AMAS
```

Step 3. Install ipykernel package:

```
conda install -c anaconda ipykernel
```

Step 4. Install an interactive python kernel in your environment:

```
python -m ipykernel install --user --name=AMAS
```

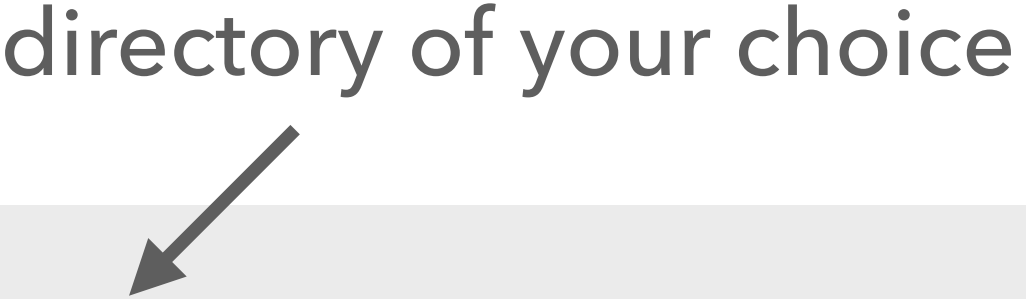
* – [reference](#)

Alternative no-conda route (virtualenv)*

Step 0. [Install virtualenv](#).

Step 1. Create a new environment:

```
virtualenv $HOME/AMAS
```



Step 2. Activate your environment:

```
source $HOME/AMAS/bin/activate  
(Windows: $HOME/AMAS/bin/activate)
```

Step 3. Install ipykernel package:

```
python -m pip install ipykernel
```

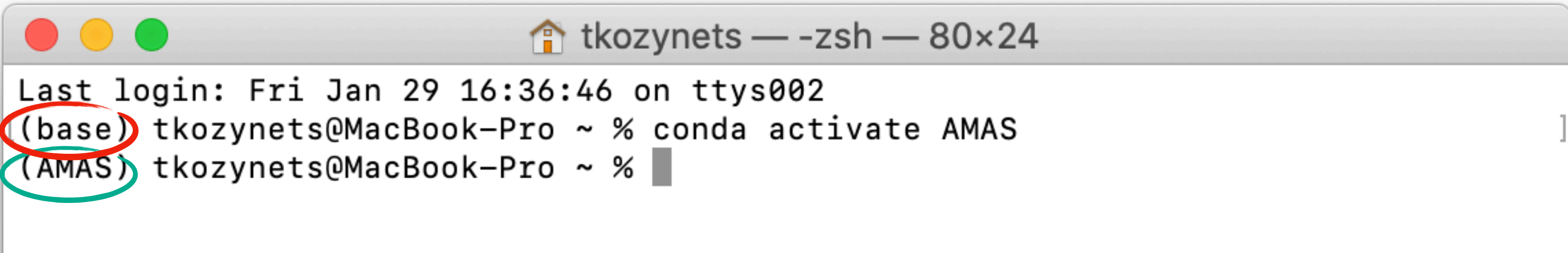
Step 4. Install an interactive python kernel in your environment:

```
python -m ipykernel install --user --name AMAS
```

* – [reference](#)

Your environment = your kingdom

> Remember to activate your preferred environment* prior to installing packages in it:



A terminal window titled "tkozynets — -zsh — 80x24" showing the process of activating a conda environment. The prompt changes from "(base)" to "(AMAS)". A red arrow points to the "(base)" prompt with the label "not activated". A green arrow points to the "(AMAS)" prompt with the label "activated".

```
Last login: Fri Jan 29 16:36:46 on ttys002
(base) tkozynets@MacBook-Pro ~ % conda activate AMAS
(AMAS) tkozynets@MacBook-Pro ~ %
```

optional; making sure that we're using pip that belongs to this conda environment

> Now we can install packages in the AMAS env. via `python -m pip install package_name`.

> You can also install some packages via `conda install package_name`. Always [check](#) for a specific package.

* – if your preferred environment is *base*, that's fine too

Installing Jupyter Notebook*

> After activating your preferred environment (if applicable), run either

```
conda install -c conda-forge notebook
```

OR

```
pip install notebook
```

> After a successful installation, you should be able to type `jupyter notebook` in your terminal and have a browser window pop up.

> You will see the structure of your current directory and will be able to create notebooks there.

What a successful installation should look like

create a new notebook

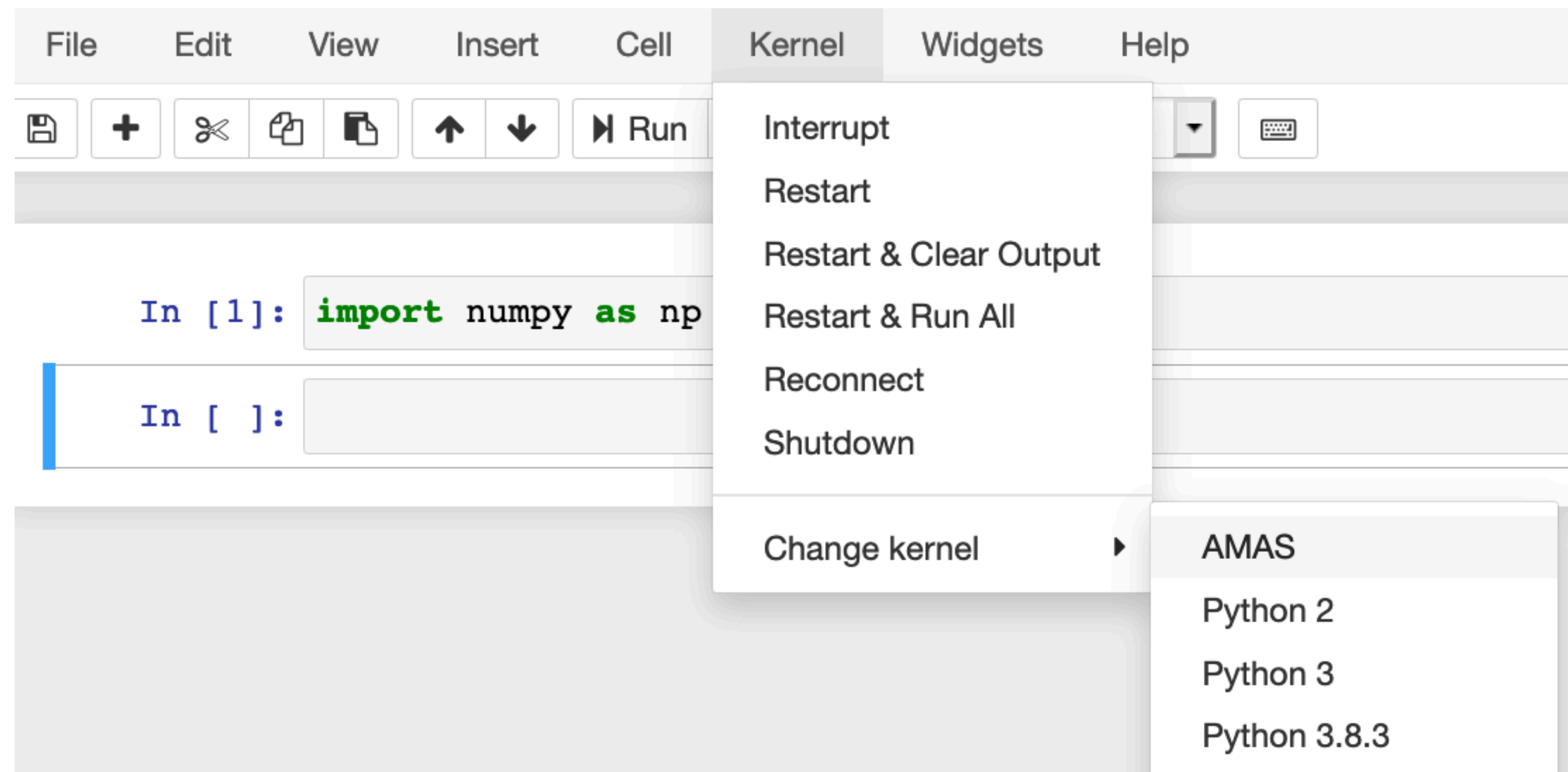
choose your preferred environment

The screenshot shows a file manager interface with a breadcrumb path: / Documents / Teaching / AAMS2021 / exercises / class1. The file list contains: .., class1_exercise1.ipynb, and FranksNumbers.txt. A 'New' dropdown menu is open, showing options under 'Notebook:' (AMAS, Python 2, Python 3, Python 3.8.3, Python 3.8.3, fenics2018) and 'Other:' (Text File, Folder, Terminal). Red arrows point from the text 'create a new notebook' to the 'New' button and from 'choose your preferred environment' to the 'AMAS' notebook option. A grey arrow points from the text 'here, you can create a simple python script (.py) if needed' to the 'Text File' option.

here, you can create a simple python script (.py) if needed

Choosing a kernel inside Jupyter

> If you accidentally (or intentionally) opened a different environment, you can always switch back to AMAS by clicking on Kernel → Change kernel → AMAS.



Python packages we will need

package name	what we will need it for
numpy	all math, sampling of simple (uniform, normal,...) distributions
scipy	sampling of custom ($x^4 + \ln(\sin(x))$) distributions, statistical tests, confidence intervals, minimizers and fitting routines, splines, KDE
matplotlib	all plotting
healpy	handling pixelated data on a sphere (autocorrelation lecture)
scikit-learn	boosted decision trees (BDTs), principal component analysis (PCA), other simple machine learning algorithms
nestle/pymultinest	nested sampling
PyMC/emcee	Markov Chain Monte Carlo (MCMC) samping, <i>but we'll code our own</i>

Last words before we take off to install things

- > Despite the dry information you just received, the course is actually very fun and useful;
- > This will be the only major “software help” session, but feel free to contact us later about conceptual matters.

Good luck!