Institute of language, communication and the brain

ILCB

Data in Cognitive Science: EEG TP

Setting Up Python IDE

For this practical session, you will need to have a Python IDE installed on your computer that, ideally allows you to work with the Jupyter Notebook. If you encounter problems with Jupyter, you can simply use a Python IDE such as Spyder, PyCharm etc., in which case you will use the python scripts (ending in *.py).

You will also require the following packages:

- **MNE-Python**: https://mne.tools/stable/index.html

  pip install mne
  conda install --channel=conda-forge mne-base

- **NumPy:**

  conda install numpy
  pip install numpy

- **Matplotlib:**

  pip install matplotlib
  conda install matplotlib

- **PyEDFlib:**

  Pip install pyEDFlib
  Conda install -c conda-forge pyedflib

Please ensure that everything is installed before the practical session.

All installation instructions are provided here. You can test your installation with the Jupyter notebook provided, "DataEEG-TP-test.ipynb" or the python script "DataEEG-TP-test.py" .

It is very important that you can visualize and interact with the figures, so do test your Jupyter or Python installation.

If you have any problems with the installation, you can contact me at:
deirdre.bolger@uni-amu.fr

This document contains instructions on installing and setting up the following:

1. Installing Anaconda Python Distribution and Jupyter Notebook App

2. PyCharm IDE
3. **DataSpell**.

Installing Anaconda Python Distribution and Jupyter Notebook App

Anaconda Python Distribution with at Python version 3.6 or higher. You can download the version suitable for you operating system (MacOs, Windows) here

The Jupyter Notebook App is generally provided in the Anaconda distribution; the main Anaconda navigator window should resemble figure 1. In figure 1, the Jupyter Notebook is installed and you only need to launch. If for Jupyter Notebook, instead of "Launch" you see "Install", click on the "Install" button to…install it.

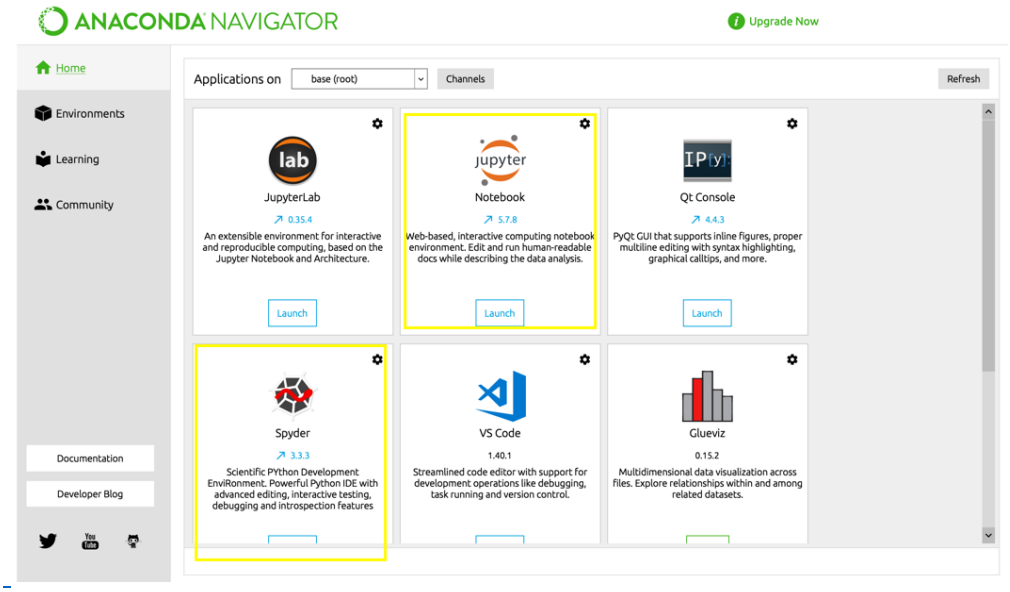
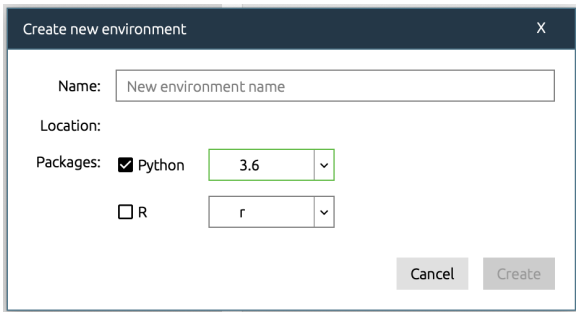


Figure 1: Main Anaconda Navigator window.

It could be a good idea to create a new "Environment" for this practical session. You can do this via the "Environments" window by clicking on the "Create" icon at the bottom of the page (Figure 2). The following window will open:



In « **Name** » insert whatever you would like to call the new environment (e.g. **DataCogSci_EEG**). Select Python version **3.6 or 3.9**. The click on "**Create**"

Now you are ready to install mne-Python and its dependencies in the Environment that you have created. Already, you can check which packages are installed in your new Environment

via the “Environments” window (Figure 3). In figure 3 you can see that the environment, “new_mne” has “numpy” installed, among others.
MNE-Python needs “numpy”, “matplotlib” and “scipy” to run, so ensure that these packages are installed. It may also be a good idea to have “pandas” also.

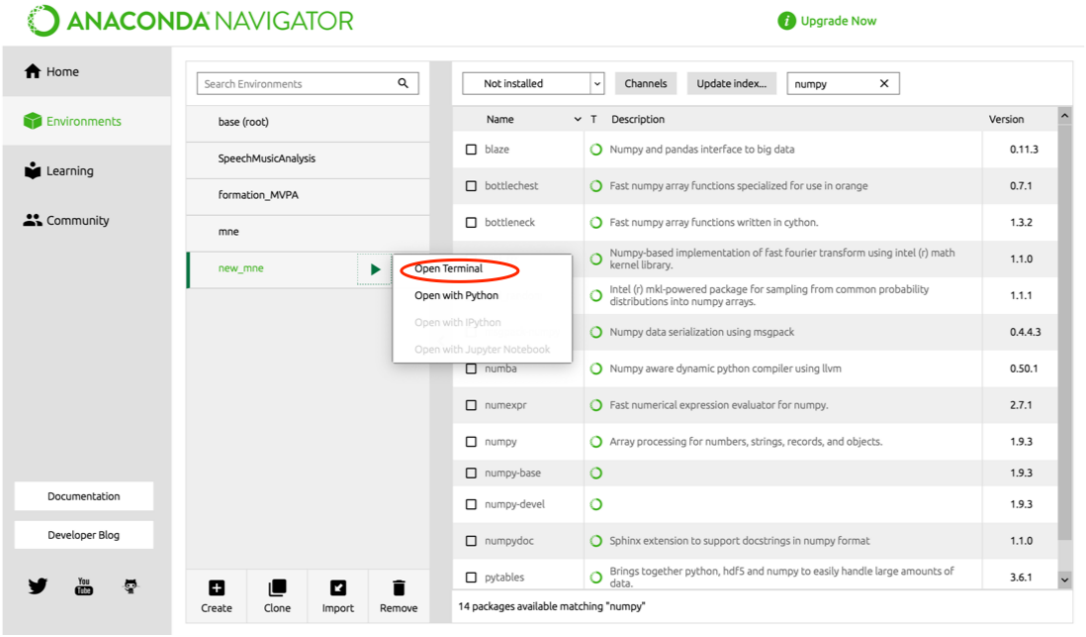


Figure 3: New Jupyter Notebook environment.

You can use open a terminal in your new environment (Figure 4) to install MNE-Python.

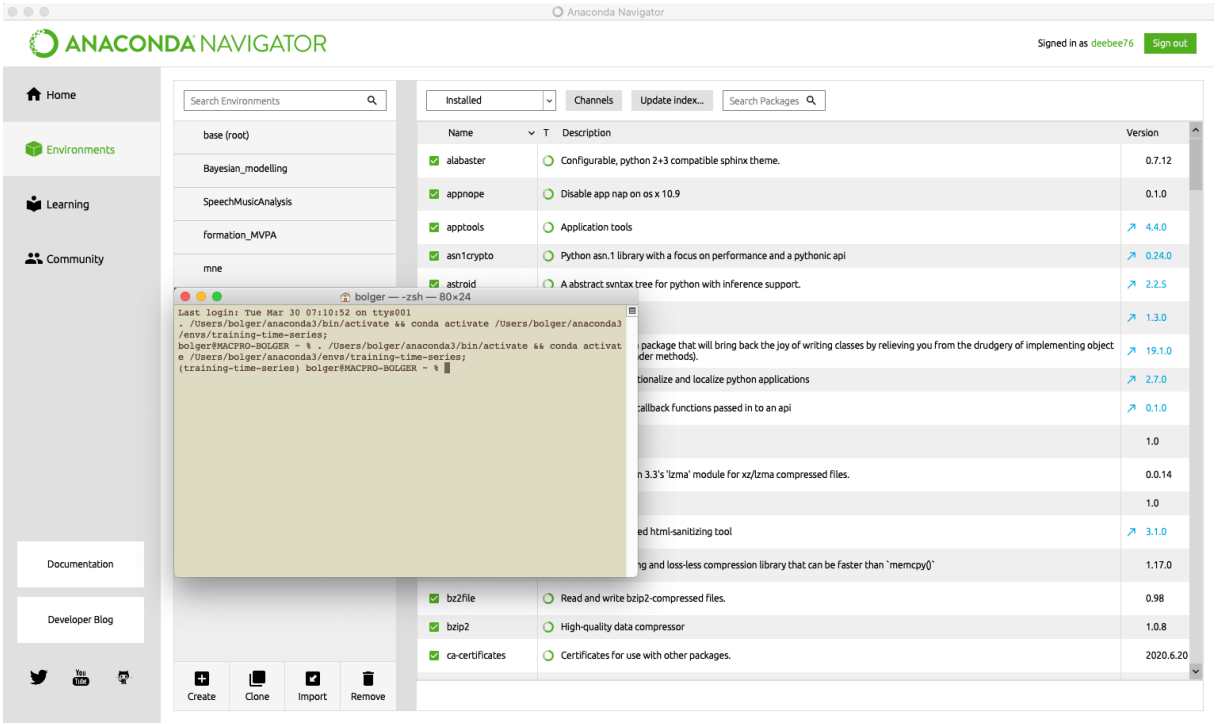


Figure 4: New Jupyter Notebook environment with terminal open (MacOs).

Institute of language, communication and the brain

Installing MNE Python and Dependencies

Conda is an open source package management system and environment management system that runs on Windows, macOS and Linux. Conda quickly installs, runs and updates packages and their dependencies.
To install the MNE-Python using conda, insert the following code in the Anaconda terminal (for MacOs) or in the "Anaconda Prompt" in the Start men (for Windows):

```
conda install -c conda-forge mne
```

If you are missing dependencies, then:

```
conda install -c conda-forge numpy
conda install -c conda-forge pandas
conda install -c conda-forge scipy
conda install -c conda-forge matplotlib
conda install -c conda-forge seaborn
```

Another option is detailed here involves creating a dedicated MNE environment directly in the command line in Anaconda:

```
$ curl -O https://raw.githubusercontent.com/mne-tools/mne-
python/master/environment.ym
$ conda env create -f environment.yml
$ source activate mne
```

The following is a very page explaining how to run the Jupyter Notebook App. with Anaconda:
https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/execute.html

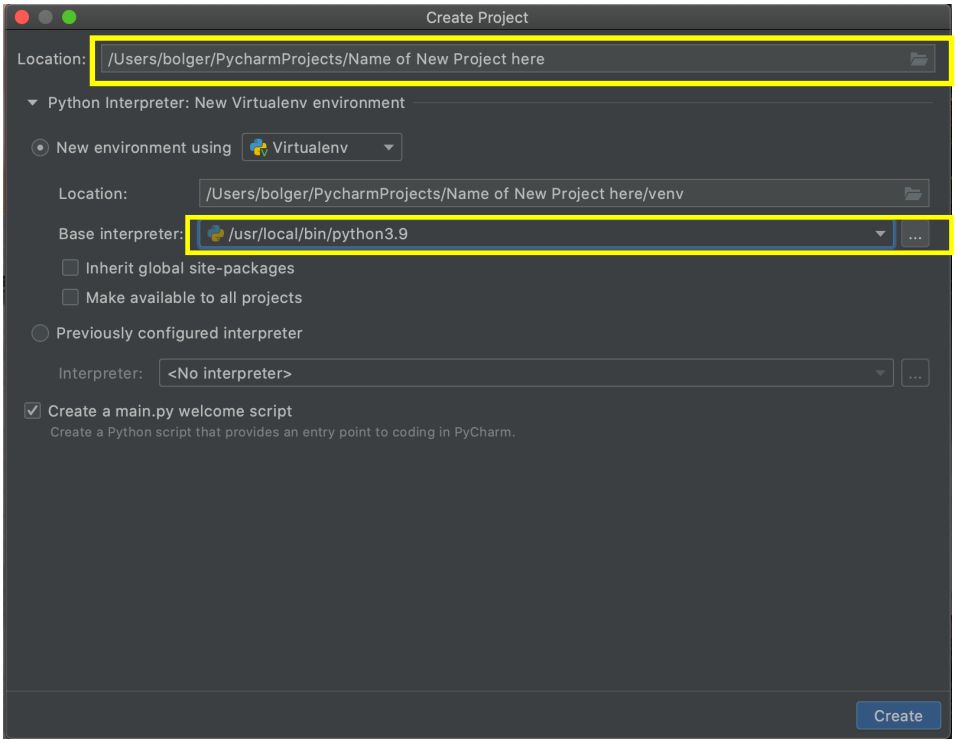
PyCharm and DataSpell

PyCharm IDE

Another option is to install **PyCharm**, a very good Python IDE. However, Jupyter notebook does not function in PyCharm.
You can download PyCharm from the following link:
https://www.jetbrains.com/pycharm/

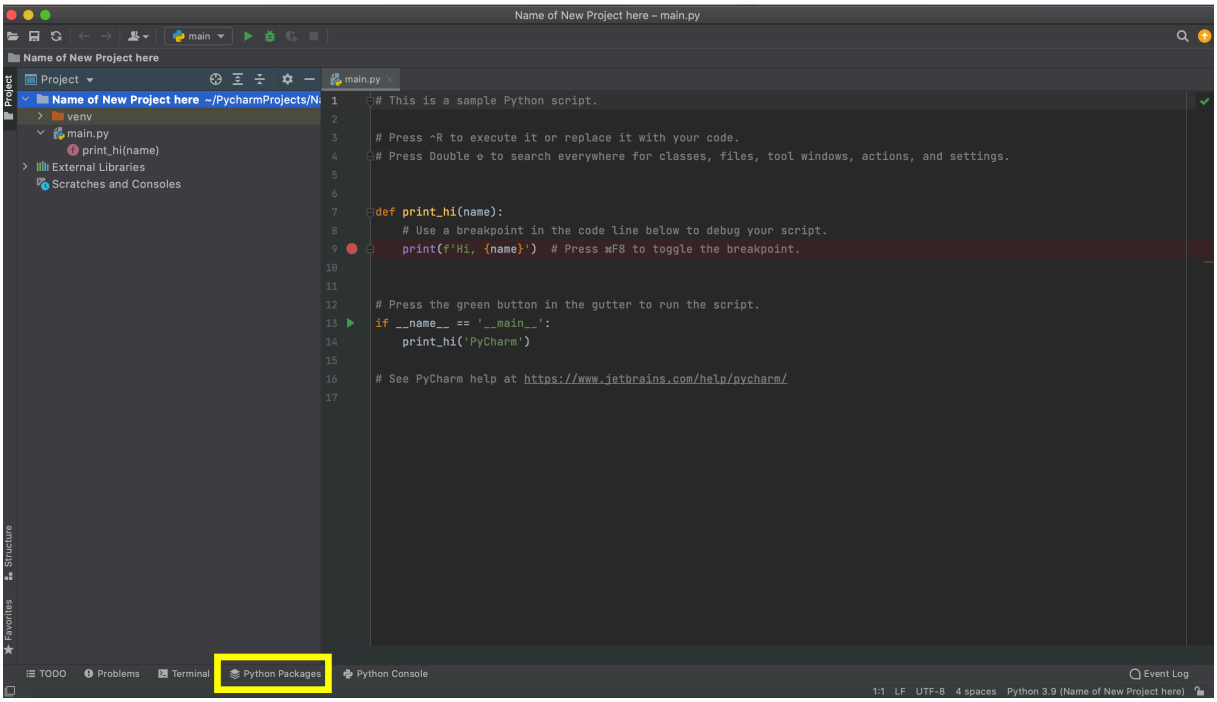
In PyCharm you can set up a new "Project". To do this: >>File>>New Project
This opens the following window (figure below).
In this window you can define the "Name of your current project" in **Location**. It will show you where this project will be saved.
In the **Base Interpreter**, you define the version of Python that your project will use. Try to use Python 3 and above, or Python 3.6 to 3.9 ideally.

Then click on “create” to create your project.

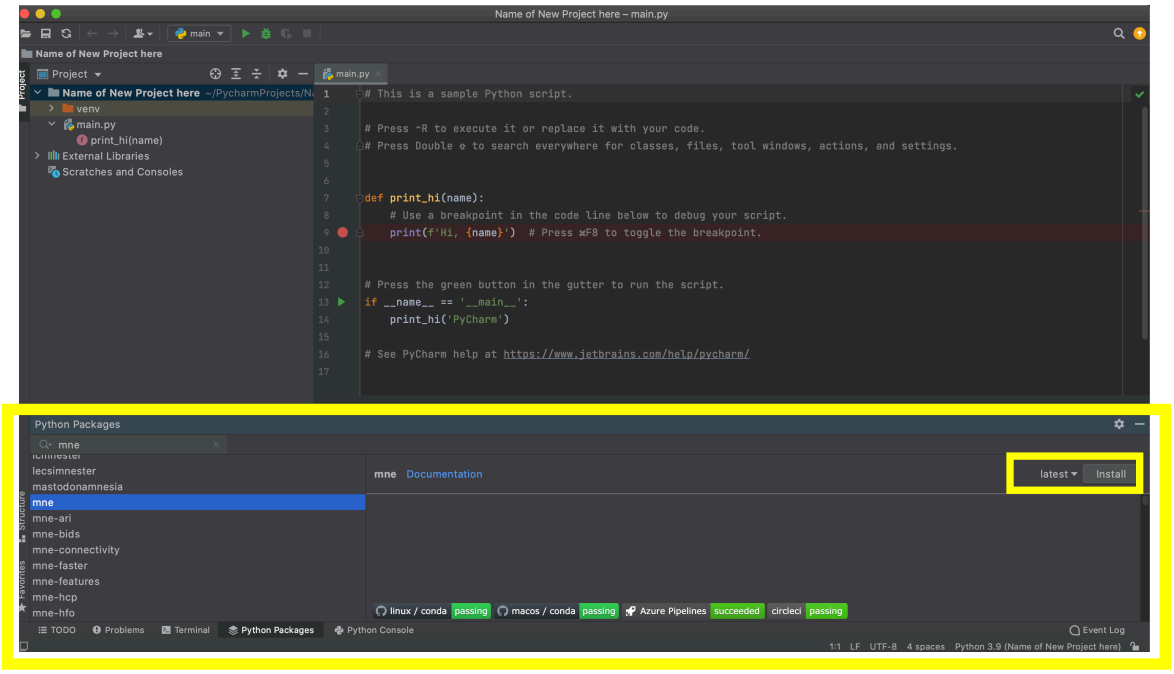


Once you have noted where this PyCharm project is located, the easiest thing to do then is to copy the folder with the scripts and datasets required for this TP into the project folder. They will automatically appear in the main PyCharm project window.

Installing Python packages is very easy in PyCharm (or at least it should be).
At the bottom of your main PyCharm project window, you should see a menu title called “Python Packages” (see figure below).



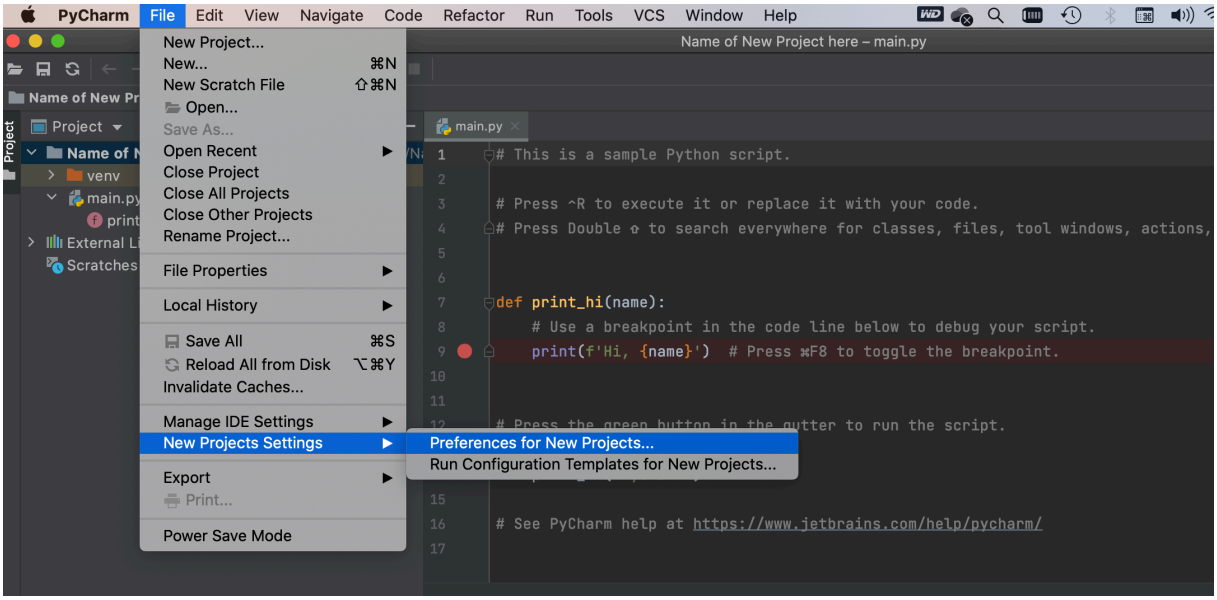
Click on the “Python Packages” title to reveal this window:



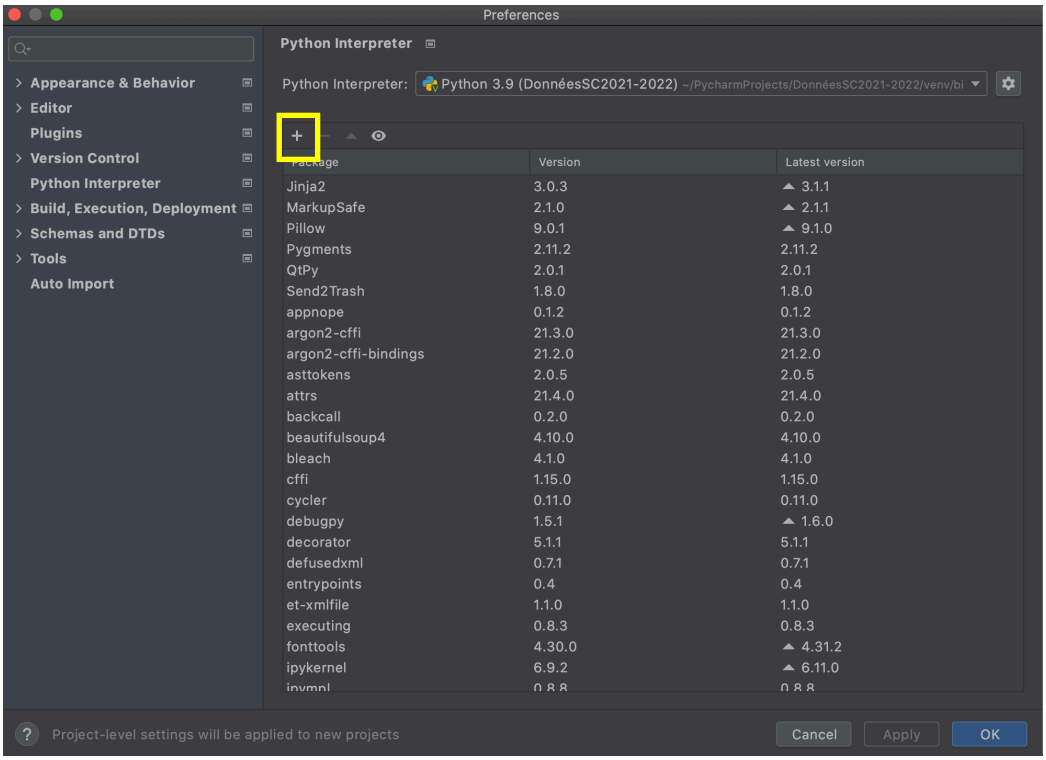
In the space provided, search for a package; in the above example we are searching for “mne”. The package is installed into the current project by clicking on “Install” to the right of the window.

If you cannot find the package that you are looking for, you can try the following:

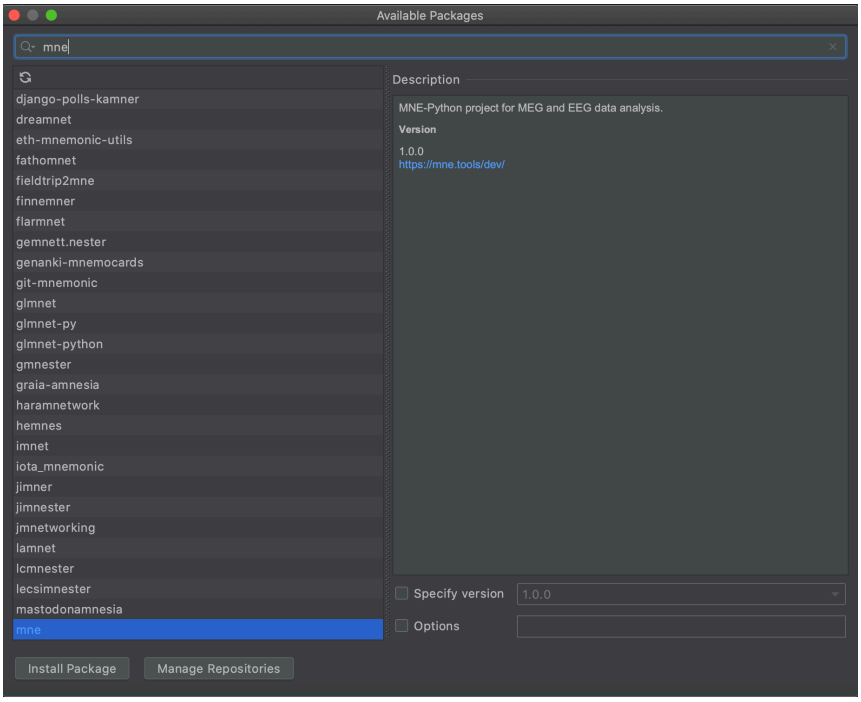
>>File>>New Project Settings>>Preferences for New Projects



The click on the "+" sign to add a new package to the list of packages.



Once you've pressed "Ok" the following window will open, allowing you to search for the package that you want. In the example below we are searching for MNE package. You can specify the version that you want, it is best to select the most recent version. The press "Install Package". Now you should be able to find the package in the "Python Packages" menu.



Jupyter dans PyCharm

Malheureusement, la version gratuite de PyCharm ne vous donne pas l'option de créer des fichiers en format Jupyter (*.ipynb) dans le projet PyCharm. Mais, vous avez la possibilité d'installer soit « Jupyter notebook », « JupyterLab » à partir des « Python Packages » ; j'ai installé « jupyterLab ».

Une fois JupyterLab ou Jupyter notebook installé, vous pouvez ouvrir un environnement dans votre browser en tapant « jupyter lab » dans le terminal de PyCharm. Normalement, dans l'environnement Jupyter, vous devriez voir tous les scripts qui sont dans votre projet PyCharm…en tout cas, c'est le cas pour moi avec JupyterLab.

…just realised that I added the above part in French…

DataSpell

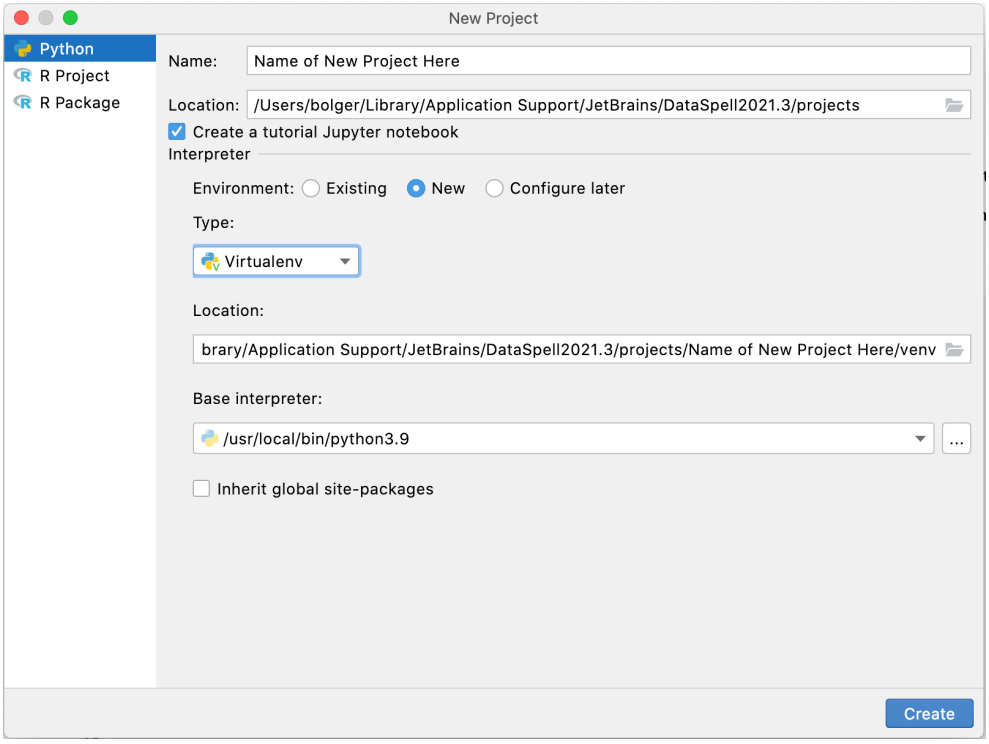
DataSpell facilitates working with Jupyter Notebook and is an excellent tool for working with Jupyter notebooks. It works just like PyCharm.

In **DataSpell** you can work with both python scripts and Jupyter Notebook.
However, DataSpell is not free but you could avail of the 30-day free trial, which works very well.

Download DataSpell from: https://www.jetbrains.com/fr-fr/dataspell/

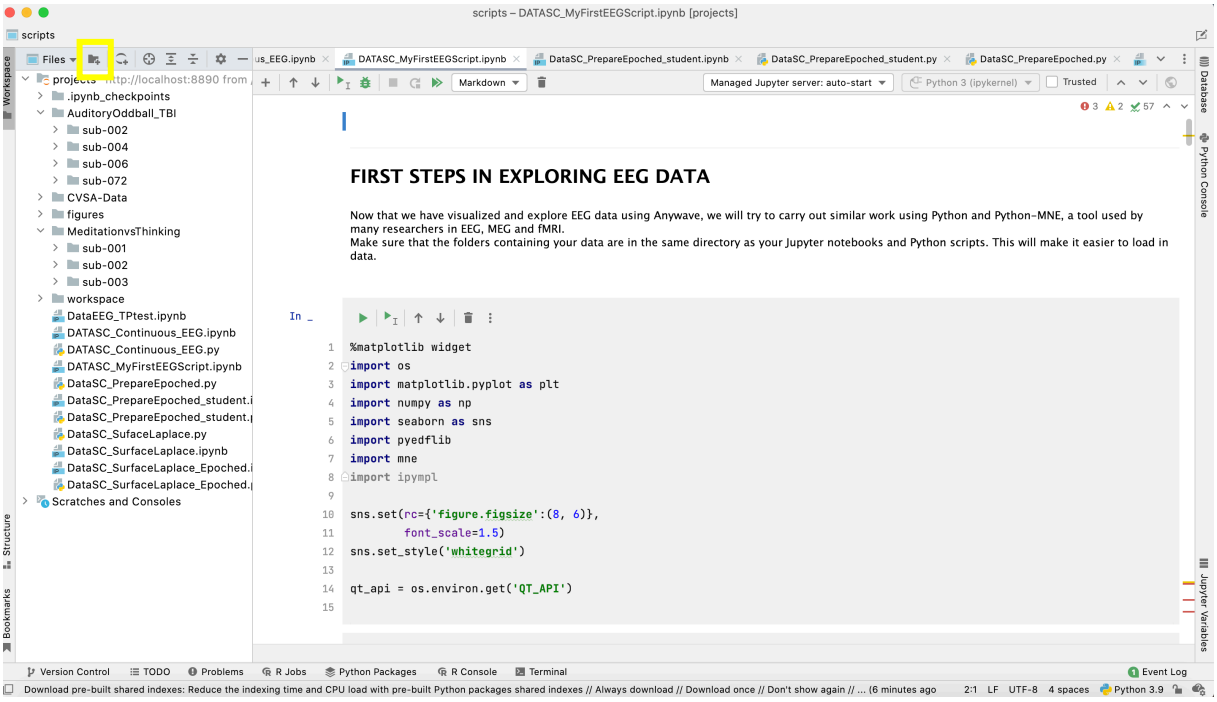
In DataSpell, rather than importing scripts and data into your DataSpell project, you can attach or link your current DataSpell project to a folder on your computer. The contents of this folder will automatically appear in your current DataSpell Project.

You can create a new project in DataSpell in the same way as you do in PyCharm.
>>File>>New Project
This opens this window in which you specify the name of your new project. You can see where your project is located (in **Location**) and you can define the version of Python that your project will use in **Base Interpreter**.

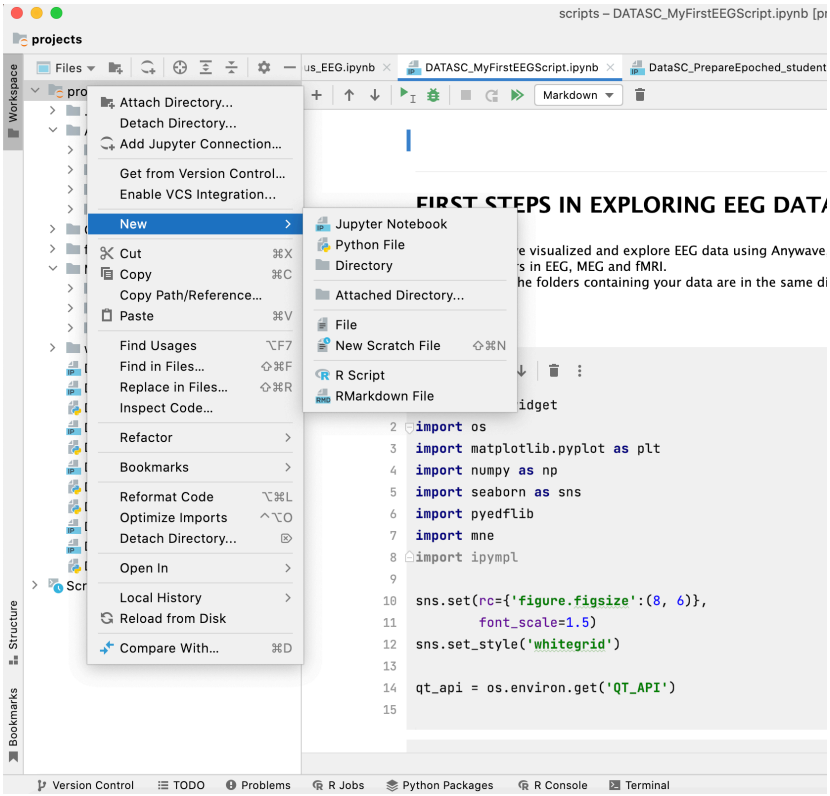


To attach a directory to your project: >>File>> Attach Directory (or icon indicated in the figure below). This opens a search window where you can navigate to the directory that you want to attach to your project. You could, for example, attach the "**DataSC_EEG_Project**" directory that you downloaded for this TP.

Institute of language, communication and the brain



You can create a new Jupyter notebook by right clicking on project folder and selecting “New Jupyter Notebook”.



To create a link to Jupyter Notebook: >> File >> Add Jupyter Connection
This will allow you prepare your Jupyter notebook in DataSpell and deploy it online.


ILCB
Institute of
language, communication
and the brain