Pip, conda and environments

DATA ANALYTICS BOOTCAMP | IRONHACK

WHAT IS A PYTHON LIBRARY?

- In python, **libraries are containers** with lots **of useful new variables types and functions/methods** developed by third parties. This allow us to save time avoiding the need of coding them.

- A good analogy can be to think of libraries like bookshelves and the functions as books. If we need a book, we need to go to the corresponding bookshelf and then pick the book or books we want.

HOW WE CAN INSTALL A PYTHON LIBRARY?

- There are basically two ways to install them (if they are not already installed):
  - **Conda**
    - Pros: easy to use
    - Cons: not all the libraries are available and migth be outdated
  - **Pip**
    - Pros: easy to use, more libraries and most recent version
    - Cons: None

- However, please **do not mix libraries conda and pip libraries together** as this will break your system.

WHAT IS THE DIFFERENCE BETWEEN CONDA AND PIP?

- **Conda**:
  - Package/library manager
  - It has support for many different programming languages and libraries
  - No frequent updates
  - **Allows** to create **environments**
  - It has it's own set of repositories called "channels"

- **Pip**:
  - Package/library manager
  - It only has support for **Python** libraries
  - Frequent updates (latests versions)
  - **Doesn't allow** to create **environments**
  - It has it's own repository called Pypi

- A given library can be installed with either conda or pip, however **with pip you will always install the latest version of a given library**.
- **IS NOT RECOMMENDED** to mix libraries from conda and/or pip together in the system.

ENVIRONMENTS:

- They are isolated spaces in your computer that allows you to avoid mixing libraries of different projects.

- This allows you have different versions of the same library in your system separated by a different environment.

- Each Data Analytics project must have:
    - Project folder
    - Environment
        - Libraries
- Once an environment has been created, it comes ONLY minimal libraries to run Python.

Conda environments

TOOLS TO CREATE ENVIRONMENTS: CONDA

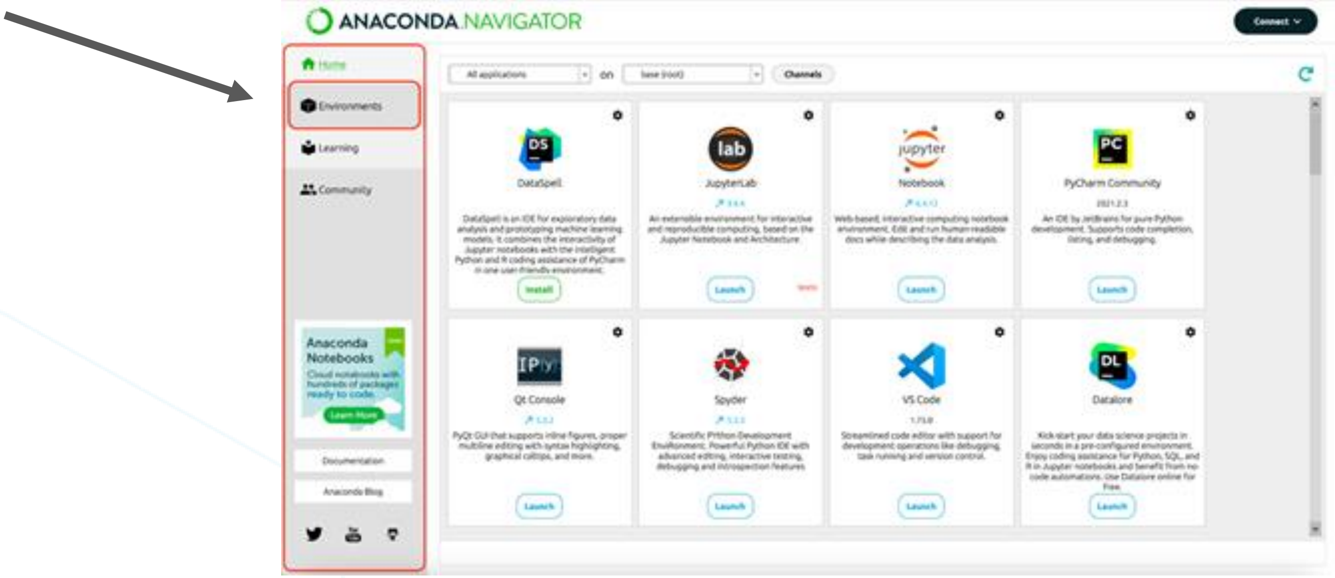
- This is a tool that can be used to install libraries AND create environments.

- In conda, there are two ways to create environments:

  - Using Anaconda Navigator
  - Using the Terminal

CREATING AN ENVIRONMENT WITH ANACONDA NAVIGATOR

- Open Anaconda Navigator and go to the left panel and select “Environments”



DataSpell is an IDE for exploratory data analysis and prototyping machine learning models. It combines the interactivity of Jupyter notebooks with the intelligent Python and R coding assistance of PyCharm in one user-friendly environment.

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

An IDE by JetBrains for pure Python development. Supports code completion, listing, and debugging.

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

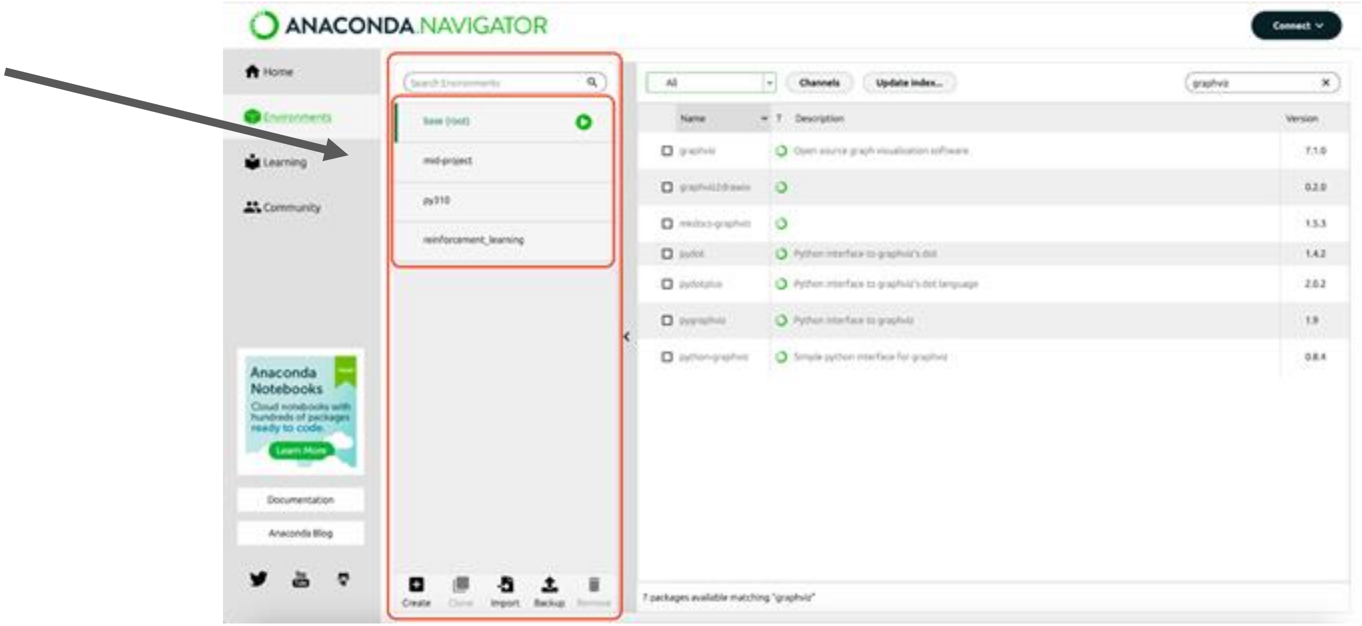
Scientific Python Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Streamlined code editor with support for development operations like debugging, task running and version control.

Kick-start your data science projects in seconds in a pre-configured environment. Enjoy coding assistance for Python, SQL, and R in Jupyter notebooks and benefit from no-code automations. Use Datalore online for Free.

CREATING AN ENVIRONMENT WITH ANACONDA NAVIGATOR

- In the middle panel of the next screen you will see a list of all the environments present in your system



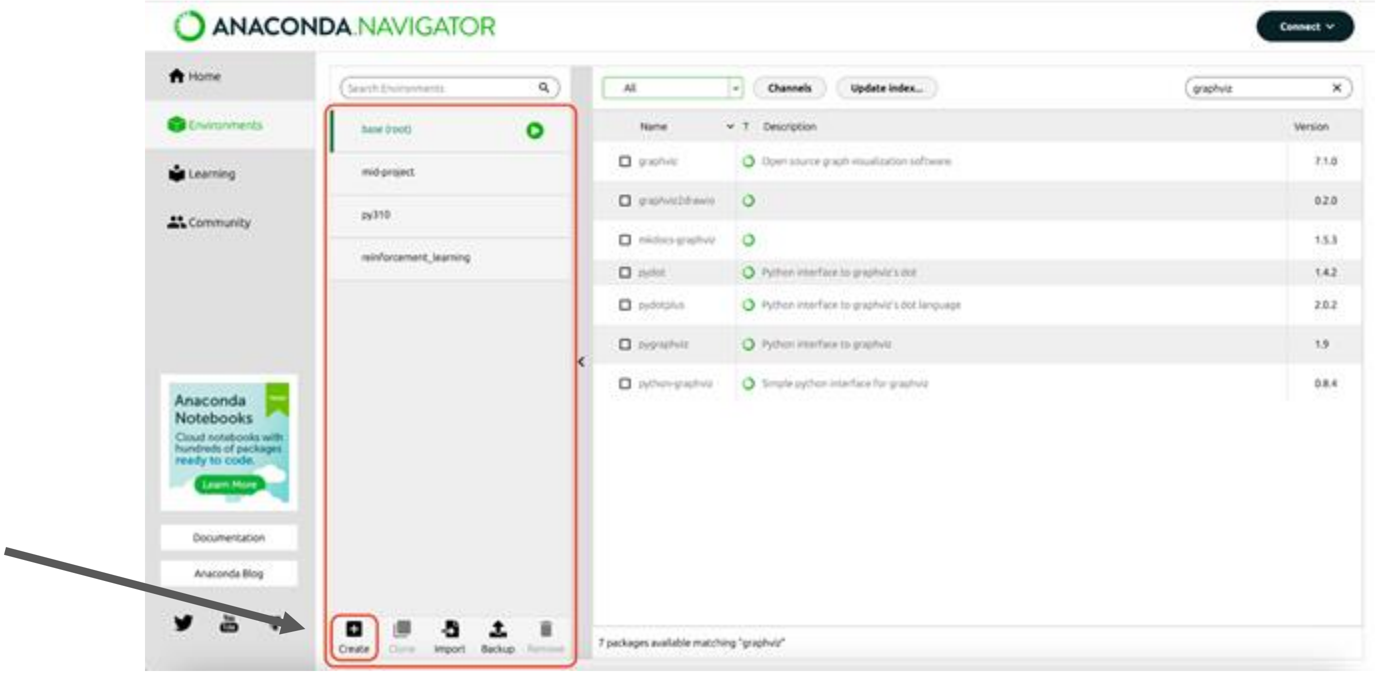
base (root)
mrl-project
py310
reinforcement_learning

Name	T	Description	Version
□ graphviz	○	Open source graph visualization software	7.1.0
□ graphviz2drawio	○		0.2.0
□ mkdocs-graphviz	○		1.5.3
□ pydot	○	Python interface to graphviz's dot	1.4.2
□ pydotplus	○	Python interface to graphviz's dot language	2.0.2
□ pygraphviz	○	Python interface to graphviz	1.9
□ python-graphviz	○	Simple python interface for graphviz	0.8.4

7 packages available matching "graphviz"

CREATING AN ENVIRONMENT WITH ANACONDA NAVIGATOR

- At the bottom of the middle panel select the “Create” icon

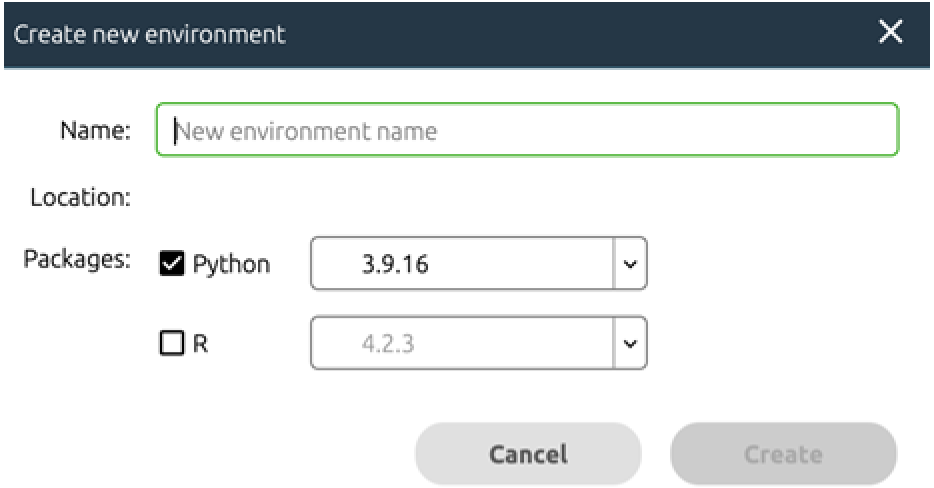


	Name	T	Description	Version
□	graphviz	○	Open source graph visualization software	7.1.0
□	graphviz2drawio	○		0.2.0
□	mkdocs-graphviz	○		1.5.3
□	pydot	○	Python interface to graphviz's dot	1.4.2
□	pydotplus	○	Python interface to graphviz's dot language	2.0.2
□	pygraphviz	○	Python interface to graphviz	1.9
□	python-graphviz	○	Simple python interface for graphviz	0.8.4

7 packages available matching "graphviz"

CREATING AN ENVIRONMENT WITH ANACONDA NAVIGATOR

- Afterwards, a popup window will appear asking you to provide a new name for environment. Input a name for it
- Mark the checkbox for Python, leave the default version and click on “Create”



Create new environment

Name: New environment name

Location:

Packages: ☑ Python 3.9.16

□ R 4.2.3

CREATING A CONDA ENVIRONMENT WITH YOUR TERMINAL

- To do this using the terminal, please:
    - Windows users:
        - go to the search bar
        - Search for “Anaconda prompt”
        - Open it
    - Linux/MacOS users:
        - Open the “Terminal” or “iTerm2” application

CREATING A CONDA ENVIRONMENT WITH YOUR TERMINAL

- In the terminal type the following command:

```
conda create --prefix ./environment_name python=3.9 jupyter ipykernel
```

replace “environment_name” by the name that you want for your environment.

- Keep in mind that the new environment will not have any of the libraries you’re used to (Numpy, Pandas, Matplotlib, and Seaborn)

ACTIVATING A CONDA ENVIRONMENT WITH ANACONDA NAVIGATOR

- Open the “Anaconda navigator” (if it’s not yet open)

- On the left panel, select the “Environments” option.

- Simply click on the environment that you want to activate (it will take a while)

- Keep in mind that:
  - Only one environment can be selected at a time
  - Each time that a Terminal or Anaconda Prompt is open, it will be opened in the default environment “base”

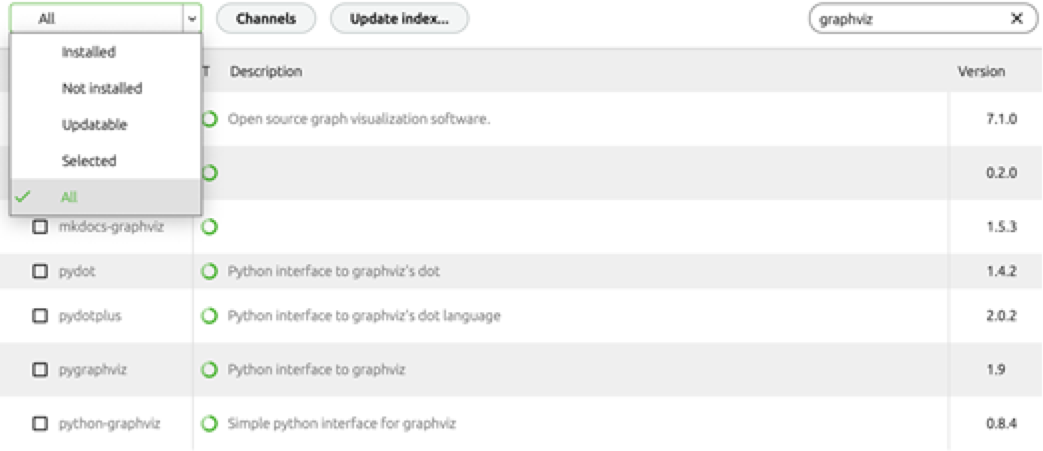
ACTIVATING/DEACTIVATING A CONDA ENVIRONMENT WITH THE TERMINAL

- This process is different for Windows and Linux/MacOS users:
  - Windows users:
    - Open Anaconda prompt and:
      - Move to the project folder where you created the environment
      - To activate the environment type: `conda activate ./environment_name`
      - To deactivate the environment type: `conda deactivate`

  - Linux/MacOS users:
    - Open the Terminal or iTerm2
      - Move to the project folder where you created the environment
      - To activate the environment type: `conda activate ./environment_name`
      - To deactivate the environment type: `conda deactivate`

INSTALLING CONDA LIBRARIES USING ANACONDA NAVIGATOR

- This process is painfully slow and buggy. Therefore, it's not recommended.
- Open Anaconda navigator, in the left panel, select the "Environments" option.
- Next, select/activate the environment in which you want to install the new library.
- Once the environment is active, go to the right panel, and in the dropdown menu select the "All" option as shown below.



	T	Description	Version
	○	Open source graph visualization software.	7.1.0
	○		0.2.0
□ mkdocs-graphviz	○		1.5.3
□ pydot	○	Python interface to graphviz's dot	1.4.2
□ pydotplus	○	Python interface to graphviz's dot language	2.0.2
□ pygraphviz	○	Python interface to graphviz	1.9
□ python-graphviz	○	Simple python interface for graphviz	0.8.4

INSTALLING CONDA LIBRARIES USING ANACONDA NAVIGATOR

- In the top right corner, go to the search bar and type the name of the library that you want to install.

- If the library that you’re looking for, is available in the conda repository, it will appear.

- Then mark the checkbox and go to the bottom right corner, and click on apply.

- This process will take a lot of time because conda needs to know wich other libraries are needed before installing the one you want to install. Next, conda will download them and intall them one by one.

INSTALLING CONDA LIBRARIES USING THE TERMINAL

- This process is a bit faster than using the Anaconda navigator but not too much.
- The process depends on your operative system.
  - Windows users:
    - Open the Anaconda Prompt
    - Make sure to activate the environment in which you want to install the libraries
    - Type:
      - `conda install library_name`
      - `conda install library_name=version`
  - Linux/MacOS users:
    - Open the Terminal/iTerm2
    - Make sure to activate the environment in which you want to install the libraries
    - Type:
      - `conda install library_name`
      - `conda install library_name=version`

Venv environments

CREATING AN ENVIRONMENT WITH VENV:

- “venv” is another tool to create environments.
- Environments created with “venv” can’t be seen by Anaconda navigator/conda
- Anaconda Navigator/conda environments can’t be seen by “venv”.
- This tool can only be used with the Terminal/iTerm2 or Anaconda prompt.

CREATING AN ENVIRONMENT WITH VENV:

- To create an environment with this tool:
    - Windows users:
        - Open Anaconda Powershell prompt
        - Move to the project folder
        - Type:
            - python -m venv ./environment_name
    - Linux/MaxOs users:
        - Open Terminal or iTerm2
        - Mode to the project folder
        - Type:
            - python -m venv ./environment_name

Please replace “environment_name” by the name you want to give to your environment.

ACTIVATING/DEACTIVATING A VENV ENVIRONMENT:

- To activate/deactivate a venv environment:
  - Windows users:
    - Open Anaconda prompt
    - Move to the project folder
    - To activate the environment type: **environment_name\Scripts\activate**
    - To deactivate the environment type: **deactivate**
  - Linux/MaxOs users:
    - Open Terminal or iTerm2
    - Move to the project folder
    - To activate the environment type: **source environment_name/bin/activate**
    - To deactivate the environment type: **deactivate**

INSTALLING PIP LIBRARIES:

- This process is way faster than using the conda/Anaconda navigator

- The process depends on your operative system.

  - Windows users:
    - Open the Anaconda Powershell Prompt
    - Make sure to activate the environment in which you want to install the libraries
    - Type:

      - pip install library_name

      - pip install library_name==version
  - Linux/MacOS users:
    - Open the Terminal/iTerm2
    - Make sure to activate the environment in which you want to install the libraries
    - Type:

      - pip install library_name

      - pip install library_name==version

LAUNCHING JUPYTER ON A VENV ENVIROMENT:

- In conda/Anaconda, is not possible to launch Jupyter notebooks from a “venv” environment. You will have to use the terminal.
- Before that, make sure to:
  - Windows users:
    - Open Anaconda Prompt
    - Navigate to the project folder
    - Activate the “venv” environment
  - Linux/MacOS users:
    - Open Terminal/iTerm2
    - Navigate to the propject folder
    - Activate the “venv” environment
- Next type the following lines (to make sure that you have pip, jupyter and ipykernel):
  - `python -m pip install --upgrade pip`
  - `python -m pip install  pip-tools  jupyter  ipykernel`

LAUNCHING JUPYTER ON A VENV ENVIROMENT:

- Next we need to activate the “ipykernel” tool that we installed before in the environment

- This tool will allows us to select in which environment we want to launch the juypter notebook.

- Type in the Terminal/iTerm2/Anaconda Prompt:
  - ○ python -m ipykernel install --user --name=environment_name

  Replace “environment_name” by the name of the “**venv**” in which you want to install this tool.

- Type in the Terminal/iTerm2/Anaconda Prompt::
  - ○ jupyter notebook
- This will launch a new tab in your browser with jupyter.

- However, you will not be able to use the terminal in which you launched jupyter until you close jupyter.

SUMMARY

- There are two tools to create environments and installing libraries:
  - conda(terminal)/Anaconda navigator(graphical user interface)
  - venv + pip (terminal)
- Each environment tool can’t see the environments created with the other
- If you use **conda/Anaconda**: install libraries with **conda**
- If you use **venv**: install libraries with **pip**
- **Never mix conda and pip libraries in the same environment**
- **Pip is way faster than conda/anaconda and less buggy but requires you to use the terminal.**