**Exercise: Making a package and pip installing**

In this exercise, you will convert modularized code into a Python package.

This exercise requires three files, which are located on this page in the **Supporting materials** section.

* Gaussiandistribution.py
* Generaldistribution.py
* 3b\_answer\_python\_package.zip contains the solution to the exercise.

**Instructions**

Following the instructions from the previous video, convert the modularized code into a Python package.

On your local computer, you need to create a folder called 3a\_python\_package. Inside this folder, you need to create a few folders and files:

* A setup.py file, which is required in order to use pip install.
* A subfolder called distributions, which is the name of the Python package.
* Inside the distributions folder, you need:
  + The Gaussiandistribution.py file (provided).
  + The Generaldistribution.py file (provided).
  + The \_\_init\_\_.py file (you need to create this file).

Once everything is set up, in order to actually create the package, use your terminal window to navigate into the 3a\_python\_package folder.

Enter the following:

cd 3a\_python\_package

pip install

If everything is set up correctly, pip installs the distributions package into the workspace. You can then start the Python interpreter from the terminal by entering:

python

Then, within the Python interpreter, you can use the distributions package by entering the following:

from distributions import Gaussian

gaussian\_one = Gaussian(25, 2)

gaussian\_one.mean

gaussian\_one + gaussian\_one

In other words, you can import and use the Gaussian class because the distributions package is now officially installed as part of your Python installation.

If you get stuck, there's a solution provided in the **Supporting materials** section called 3b\_answer\_python\_package .

If you want to install the Python package locally on your computer, you might want to set up a virtual environment first. A virtual environment is a silo-ed Python installation apart from your main Python installation. That way you can easily delete the virtual environment without affecting your Python installation.

If you want to try using virtual environments in this workspace first, follow these instructions:

1. There is an issue with the Ubuntu operating system and Python3, in which the venv package isn't installed correctly. In the workspace, one way to fix this is by running this command in the workspace terminal: conda update python. For more information, see [venv doesn't create activate script python3](https://stackoverflow.com/questions/26215790/venv-doesnt-create-activate-script-python3). Then, enter y when prompted. It might take a few minutes for the workspace to update. If you are not using Anaconda on your local computer, you can skip this first step.
2. Enter the following command to create a virtual environment: python -m venv venv\_name where venv\_name is the name you want to give to your virtual environment. You'll see a new folder appear with the Python installation named venv\_name.
3. In the terminal, enter source venv\_name/bin/activate. You'll notice that the command line now shows (venv\_name)at the beginning of the line to indicate you are using the venv\_name virtual environment.
4. Enter pip install python\_package/. That should install your distributions Python package.
5. Try using the package in a program to see if everything works!

**Supporting Materials**

* [Generaldistribution](https://video.udacity-data.com/topher/2021/April/60785584_generaldistribution/generaldistribution.py)
* [Gaussiandistribution](https://video.udacity-data.com/topher/2021/April/60785598_gaussiandistribution/gaussiandistribution.py)
* [3b Answer Python Package](https://video.udacity-data.com/topher/2021/April/6078561d_3b-answer-python-package/3b-answer-python-package.zip)

;