

To run Jupyter Notebooks with accuracy and area estimation equations on PC, follow these steps*:

**the whole process took me about 15 minutes from start to finish, but might be slower for your depending on your PC*

Please contact Alexandra (Sasha) Tyukavina at atyukav@umd.edu if you have any questions

- 1) **Install the latest version of Anaconda** from the official website:

<https://www.anaconda.com/docs/getting-started/anaconda/install#windows-installation>

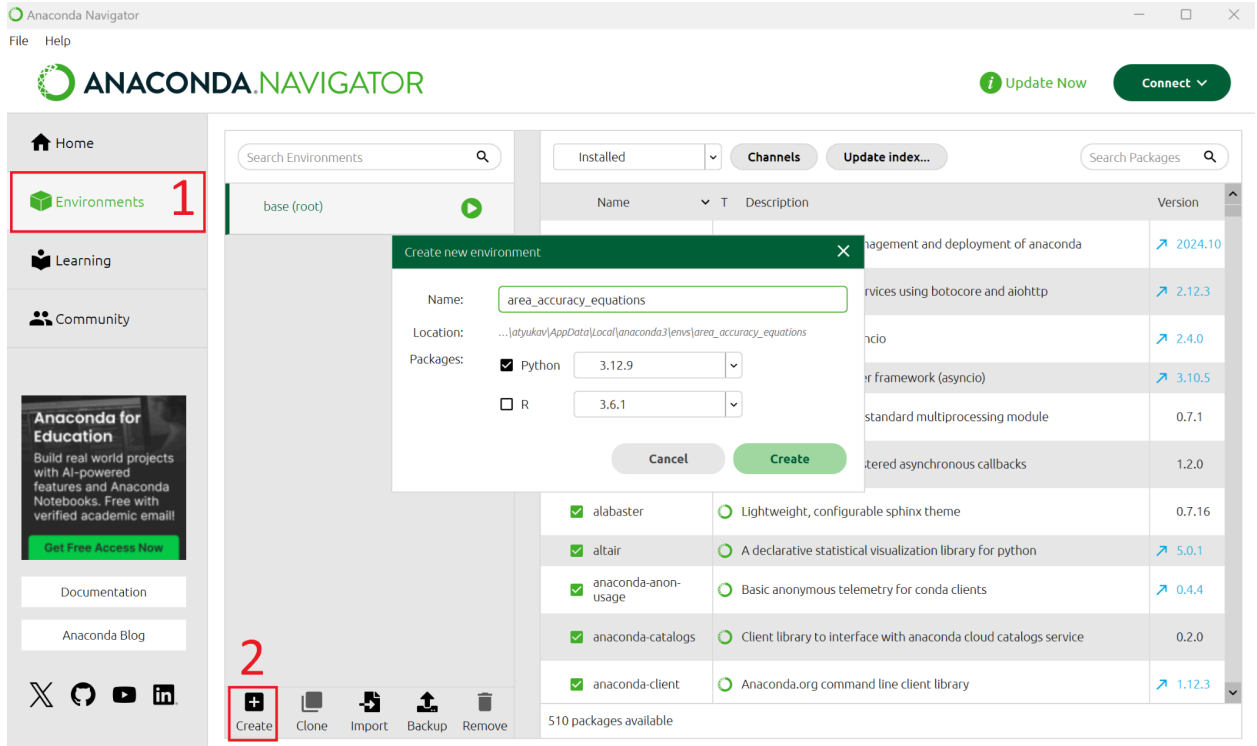
During the installation, I recommend not adding Anaconda to the PATH variable, because it can interfere with other software. Instead, use Anaconda by opening the Anaconda Navigator or Anaconda Prompt from the Start menu.

The option of registering Anaconda as your default Python may be left checked, unless you are planning to install multiple versions of Anaconda or Python.

Do not uncheck “Create shortcuts (supported packages only)” - this option creates Start menu shortcuts.

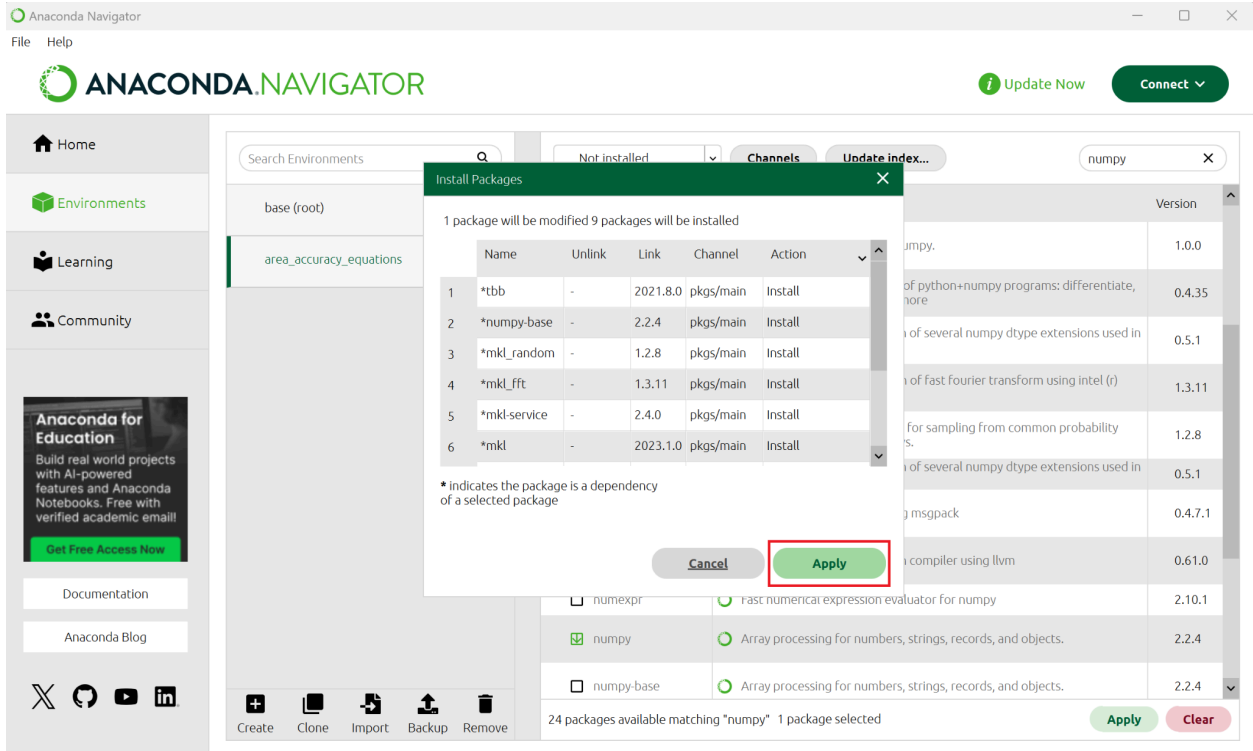
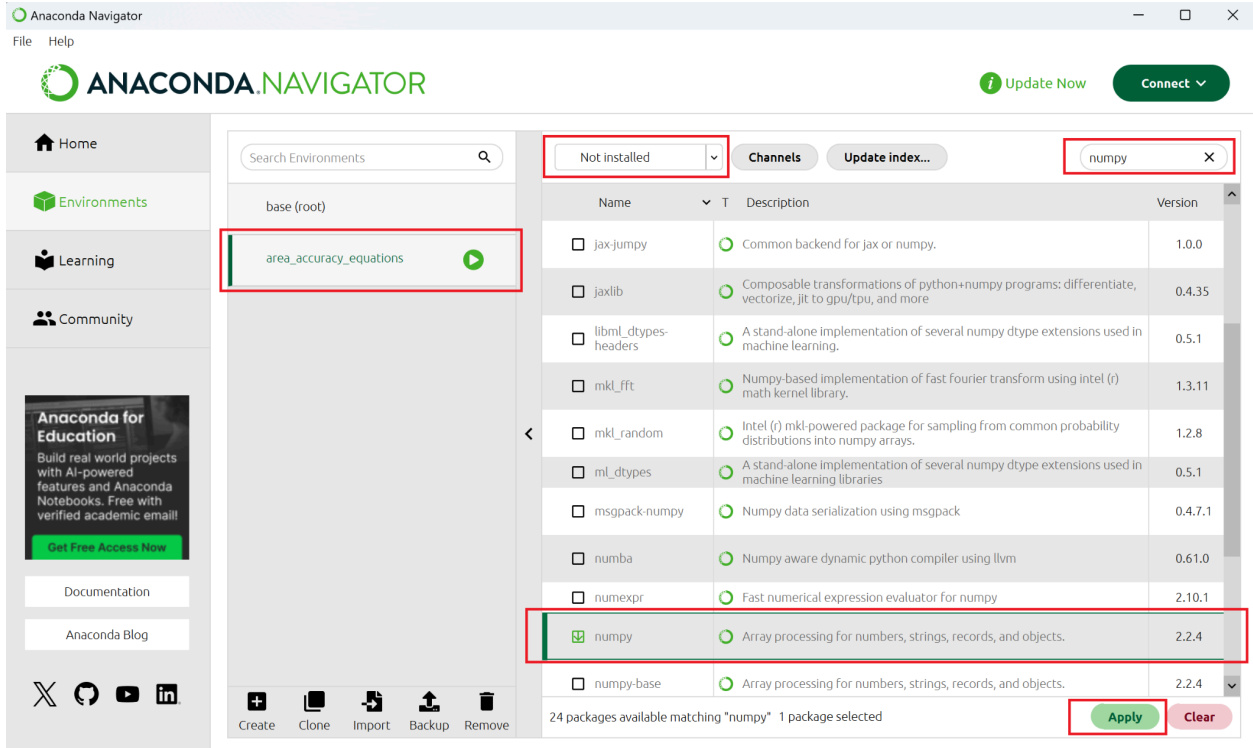
- 2) **Open Anaconda Navigator** from the Start menu. This is a GUI for interacting with Anaconda. You can do all the same things described below (creating environments, installing packages, opening Jupyter notebooks) from a command line utility Anaconda Prompt instead if working in a command prompt is more familiar to you. In this guide we will focus on the GUI steps.
- 3) **Create the environment** you will be working in (Environments -> Create). You can also work in the base environment, but packages or versions of the packages required for different projects might have conflicting prerequisites. So, it’s a good practice to create a separate environment for each type of project. Say, you can name your environment “area_accuracy_equations”.

You can choose the latest version of Python for this environment, since the code uses basic functions of numpy and pandas that shouldn’t change between versions.



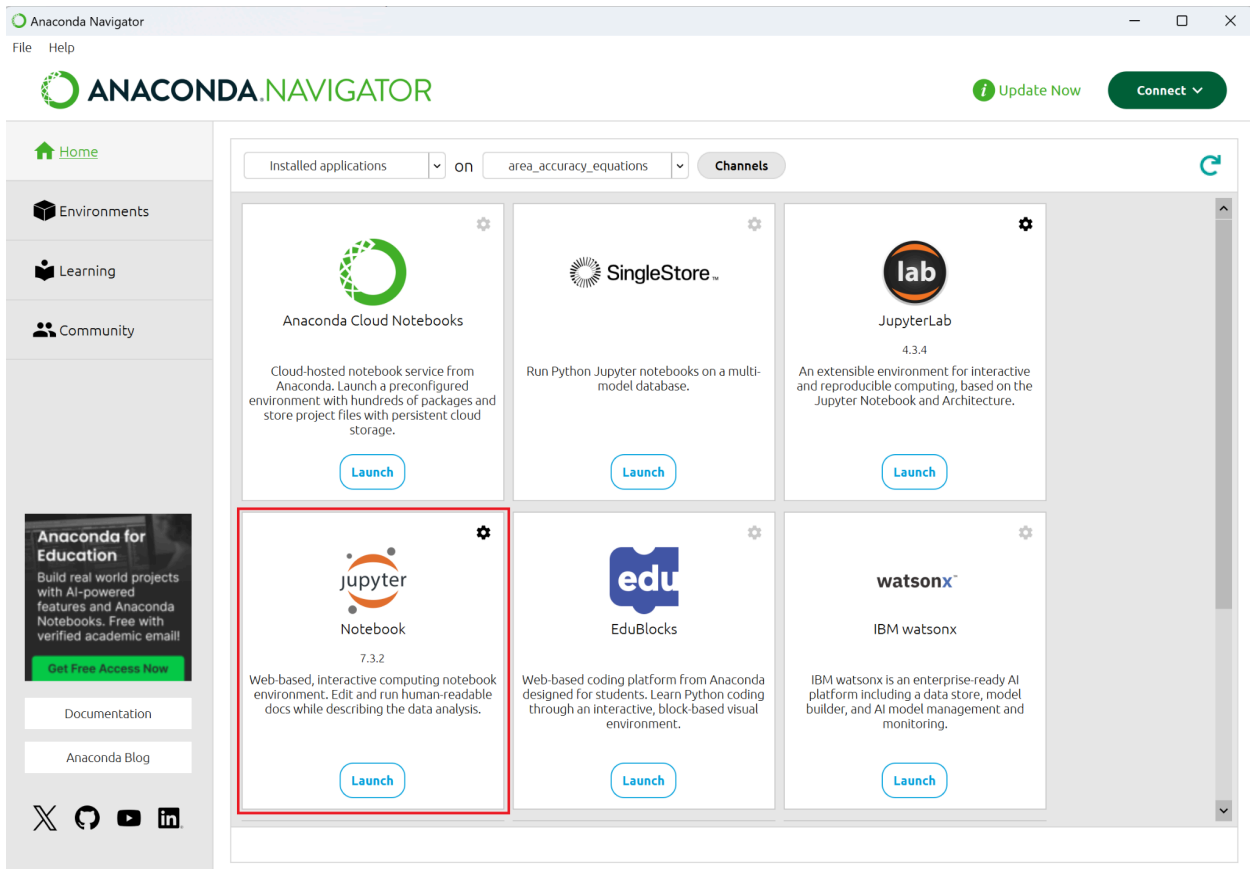
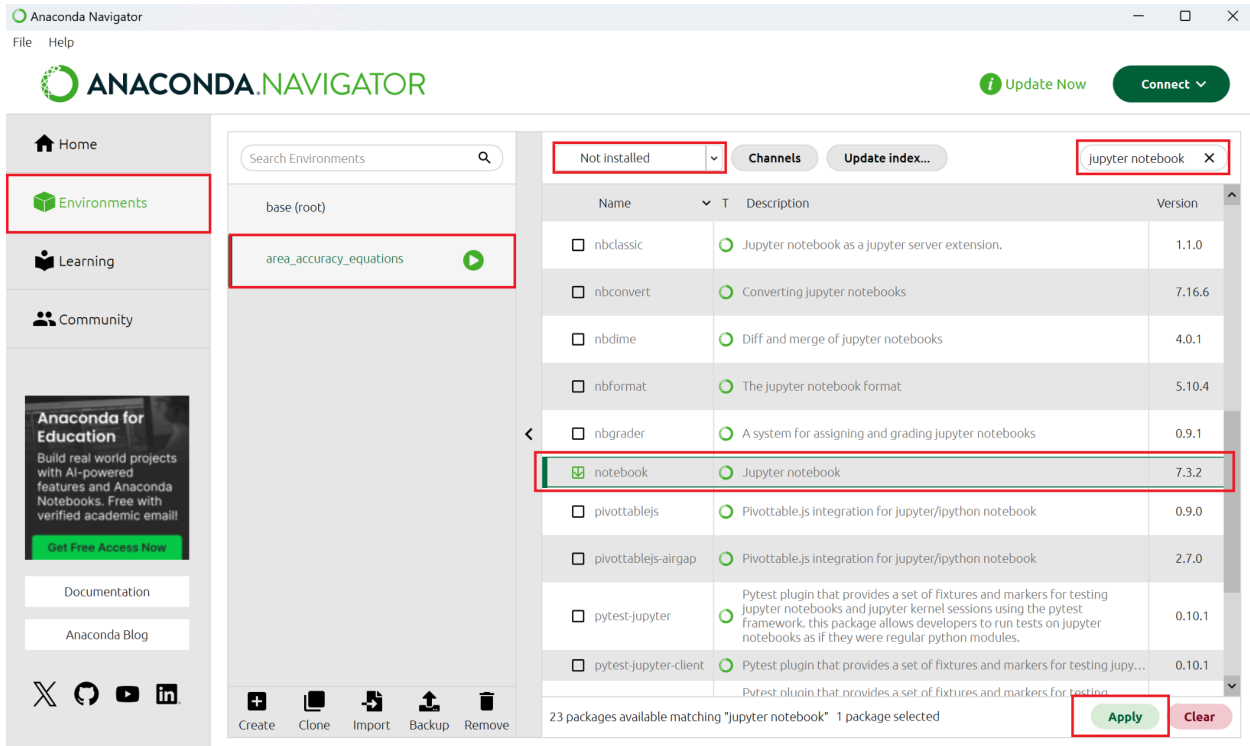
- 4) **Check if the required packages are already installed** in your newly created environment. The packages required for the current project are: **pandas** and **numpy**. If these packages are not installed by default, **you need to install them**. I had to install both (see the screenshots below with the process for installing numpy; the same sequence of steps needs to be repeated for pandas).

You will need to agree to installing or modifying any other packages that your required packages depend on.



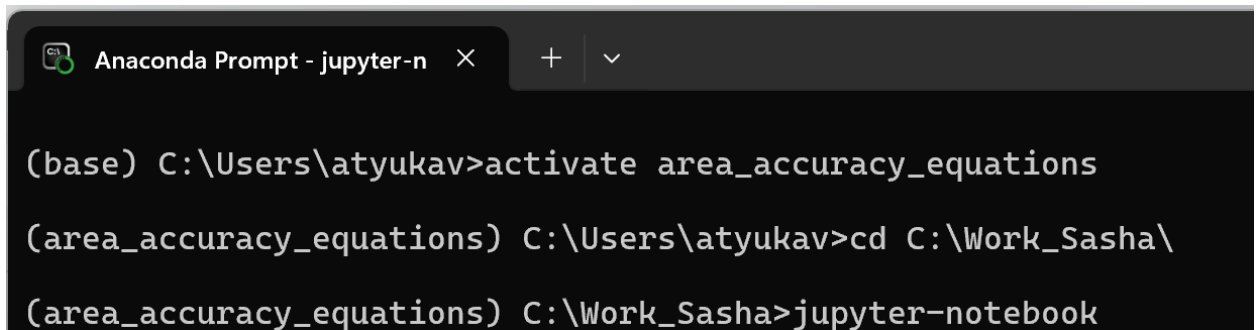
5) **Install Jupyter Notebook** (application for working with notebooks) the same way you installed required packages (see screenshot below). After installing it you

should be able to see a Jupyter Notebook application on the Home tab of Anaconda Navigator.



- 6) **Launch Jupyter Notebook application.** By default it opens in your user folder, e.g., C:\Users\username. You can create a project subfolder within that user folder, this is where you will download the Jupyter notebook(s) from Github.

I couldn't find an easy way to modify the default start folder in GUI, but you can navigate to any folder from the Anaconda prompt (Start menu -> Anaconda prompt) and launch jupyter-notebook from that folder. You will also need to activate the environment you want to work in before navigating to the desired folder (see the screenshot below).



```
Anaconda Prompt - jupyter-n  ×  +  ▾  
  
(base) C:\Users\atyukav>activate area_accuracy_equations  
  
(area_accuracy_equations) C:\Users\atyukav>cd C:\Work_Sasha\  
  
(area_accuracy_equations) C:\Work_Sasha>jupyter-notebook
```

This will open the jupyter notebook in your default browser window in a C:\Work_Sasha folder (different from a default) in an “area_accuracy_equations” environment that we have created through the Anaconda Navigator GUI.

- 7) **Download the Jupyter notebook(s)** you need from the Github page and save it to your project folder (“Download raw file” icon, see screenshot below). You can also download the sample data into the same folder.

The screenshot shows a GitHub repository for 'Global_sampling' by user 'sashatyu'. The file 'A.1.1 Equal probability sampling.ipynb' is selected. The notebook content is as follows:

```
In [1]: import pandas as pd
import numpy as np
```

Area and accuracy estimation for equal probability sampling, from Tyukavina et al. (2025)
"Practical global sampling methods for estimating area and map accuracy of land cover and change" <https://doi.org/10.1016/j.rse.2025.114714>, Appendix A.1.1

Example of input data

```
In [2]: #Read strata info table with columns:
#"Stratum": stratum ID, 1 - nstrata; for non-stratified sampling nstrata == 1 and thus only stratum ID = 1 should be
#"Area": stratum area in km2 or any other area units, needs to be consistend with pixel size area units in data tabl
#"Count": total number of units (pixels, polygons) in each stratum
strata = pd.read_csv('A.1.1.Strata_info.txt', sep = '\t')
```

```
In [3]: strata.head().style.hide(axis="index")
```

```
Out[3]:
```

Stratum	Area	Count
---------	------	-------

8) **Open the downloaded notebook** from within the launched Jupyter Notebook application. Now you can run code cells one by one (see the screenshot below). You have to run the 1st cell of each notebook (importing packages) before running any other cells. And you need to run the cells defining functions before you can use those functions elsewhere in the code. Other than that, you can run individual cells in any order.

Here is a general tutorial on using Jupyter notebooks:
<https://www.dataquest.io/blog/jupyter-notebook-tutorial/>



```
[1]: import pandas as pd
import numpy as np
```

Area and accuracy estimation for equal probability sampling, from Tyukavina et al. (2025) "Practical global sampling methods for estimating area

Example of input data

```
[2]: #Read strata info table with columns:
# "Stratum": stratum ID, 1 - nstrata; for non-stratified sampling nstrata == 1 and thus only stratum ID = 1 should be used
# "Area": stratum area in km2 or any other area units, needs to be consistend with pixel size area units in data table;
# "Count": total number of units (pixels, polygons) in each stratum
strata = pd.read_csv('A.1.1.Strata_info.txt', sep = '\t')
```

```
[3]: strata.head().style.hide(axis="index")
```

Stratum	Area	Count
1	595255.012800	788889906
2	332992.902600	781490883
3	946369.335100	1280971777
4	486272.535600	859622001
5	669855.474600	909836775

```
[4]: #Read sample interpretation table with columns:
# "Stratum": stratum ID, 1 - nstrata; for non-stratified sampling nstrata == 1 and thus only stratum ID = 1 should be used
# "Pixarea": area of each sample unit (in units that are desired for area reporting);
# "Map" (for accuracy assessment only): proportion of target class from the map (0-1) for each sample unit;
# "Reference": proportion of target class from reference sample classification for each sample unit;
# Allowed values are from -1 to 1 for area estimation, and from 0 to 1 for map accuracy assessment.
# (optional) "RefType": type Labels, if the are of target class needs to be estimated separately for multiple sub-types;
# (optional) "Correct": proportion of sample unit (0-1), which is correctly mapped.
# This column is necessary for maps with more than two classes.
```