**Hands-on Lab : Download & install Anaconda**

Time efforts: **15** minutes

**Objectives of exercise**

* Download & install Anaconda
* Create Anaconda Environment for R and Python
* Install and run Jupyter Notebook

**Overview of Anaconda**

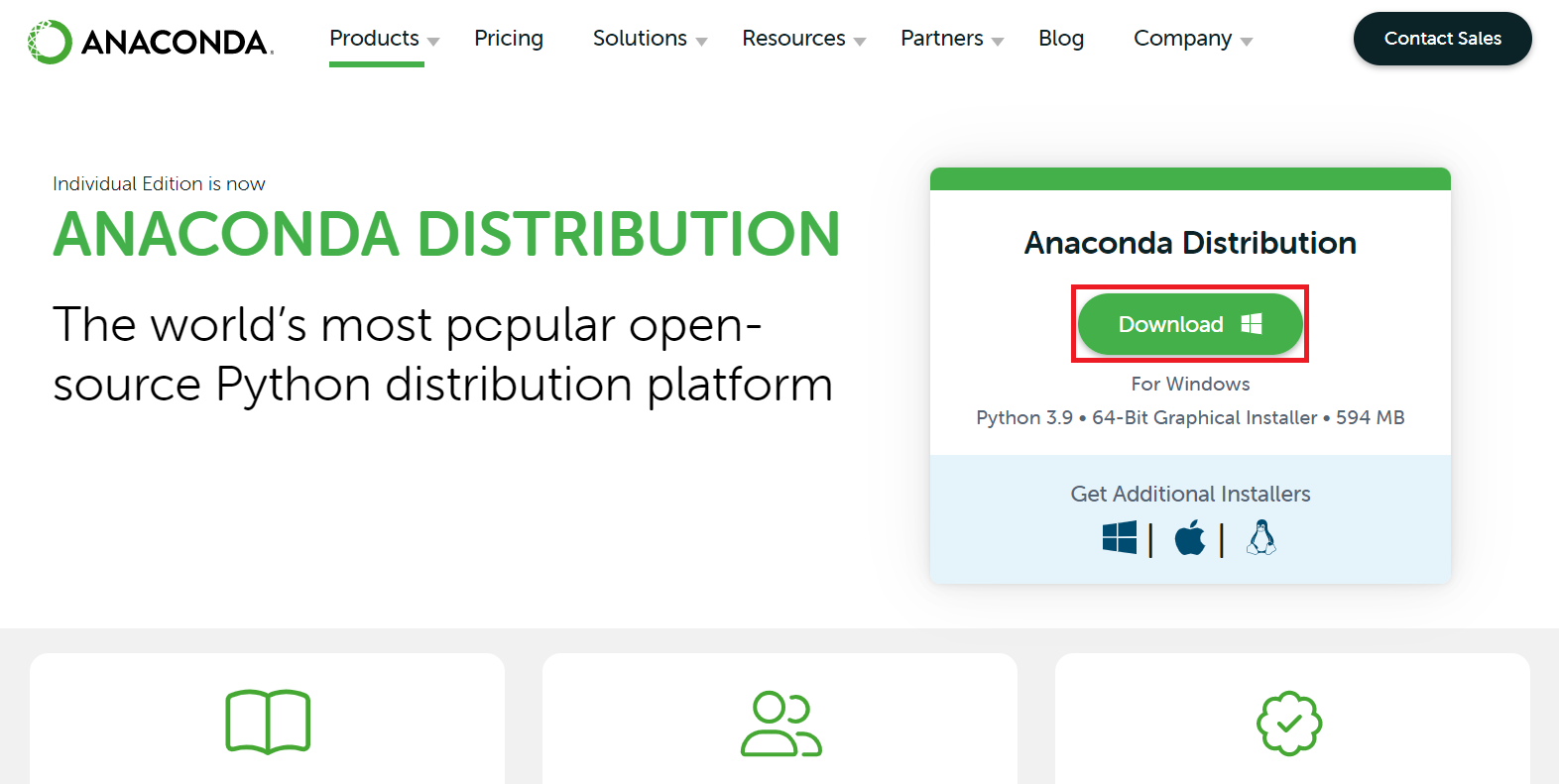
There are several cloud-based data science tools that can make team collaboration more accessible. At times it's useful to work directly on your desktop.

Anaconda Distribution is an open-source distribution of Python and R languages. It comes with a repository of a large number of packages for data science and machine learning, with the most popular and commonly used ones pre-installed. It includes Anaconda Navigator, a graphical interface (GUI) that contains several tools, and IDEs such as Jupyter Notebooks and R Studio. It has binaries for major platforms, including Windows, Linux, and macOS. This lab includes instructions for downloading and installing Anaconda on Windows.

**Exercise 1: Download & Install Anaconda Distribution**

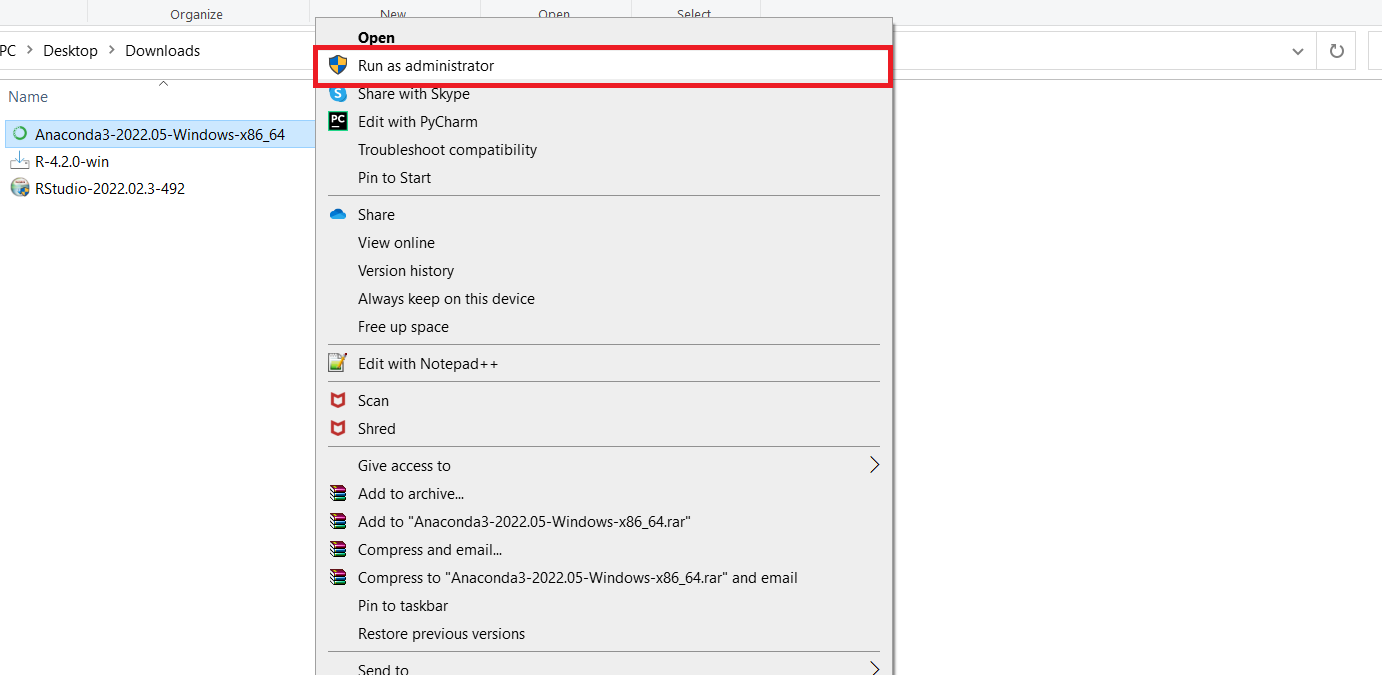
**Step 1**: Use the below link to download the Anaconda distribution:

**Link for Download Anaconda Distribution:** <https://www.anaconda.com/products/distribution>



***Note****: Depending on your****Operating system****, it would show the download link specific to your OS. Click the****Download****button to download it to your local machine.*

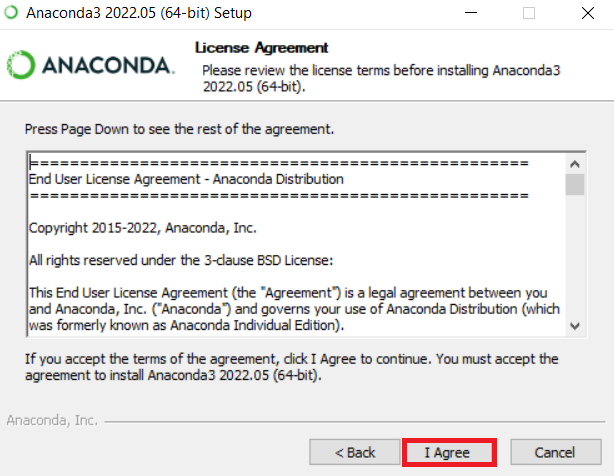
**Step 2**: Once the download completes, right-click the downloaded file and run it as **Administrator**.



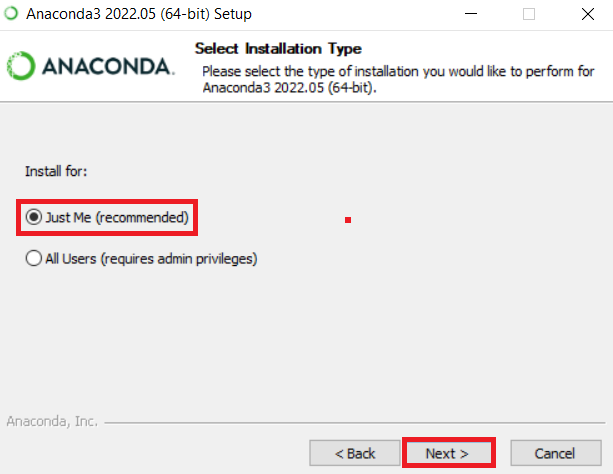
**Step 3**: At the beginning of the welcome window, you need to click **Next** to confirm the installation.



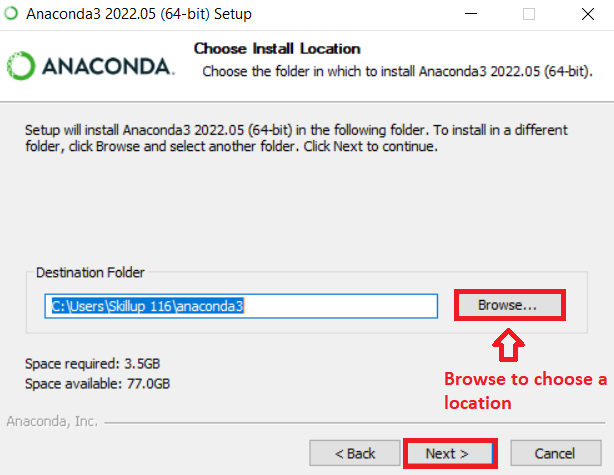
**Step 4**: Agree to the license.



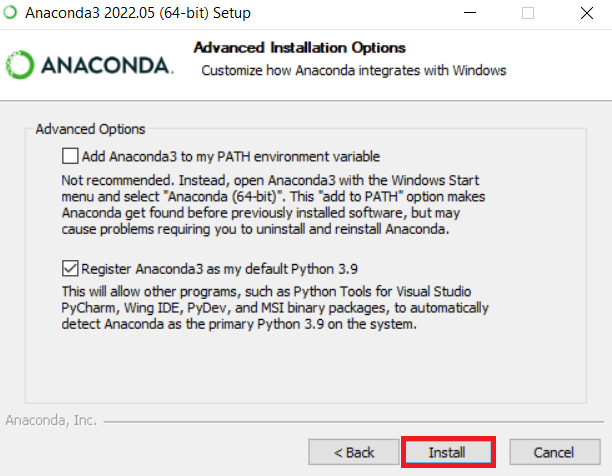
**Step 5**: In the installation window, select **Just me**, and click **Next**.



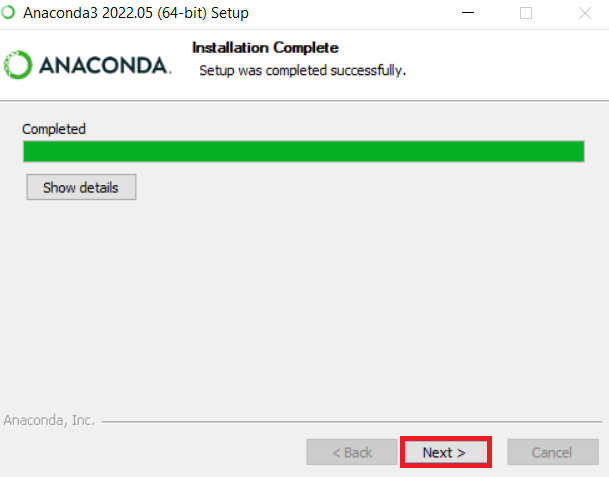
**Step 6**: Select the folder where you would like to **Install Anaconda**, or retain the **Default** installation location and click **Next**.



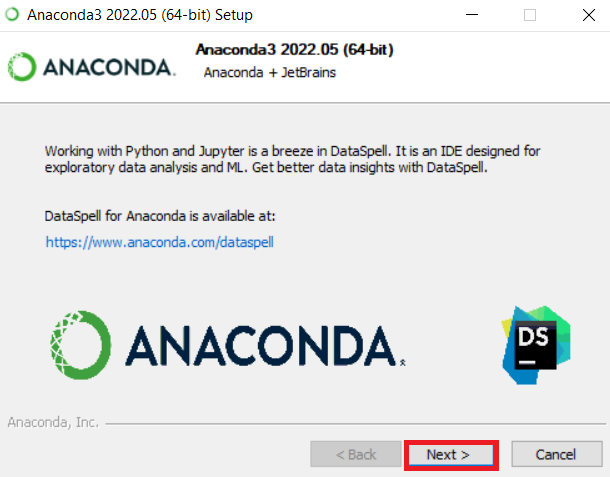
**Step 7**: In the **Advanced Installation Options** window, select **Register Anaconda3 as the default Python 3.9** option, and click **Install**.



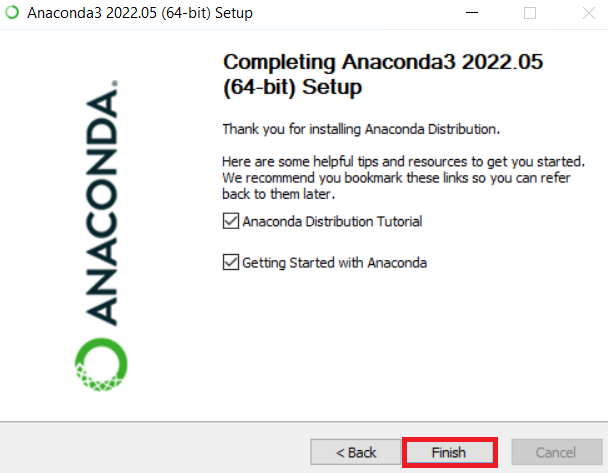
**Step 8**: You need to wait for the installation to complete. Once installation completes, click **Next**.



**Step 9**: Click **Next**.



**Step 10**: Click **Finish** to complete the installation of the Anaconda distribution.

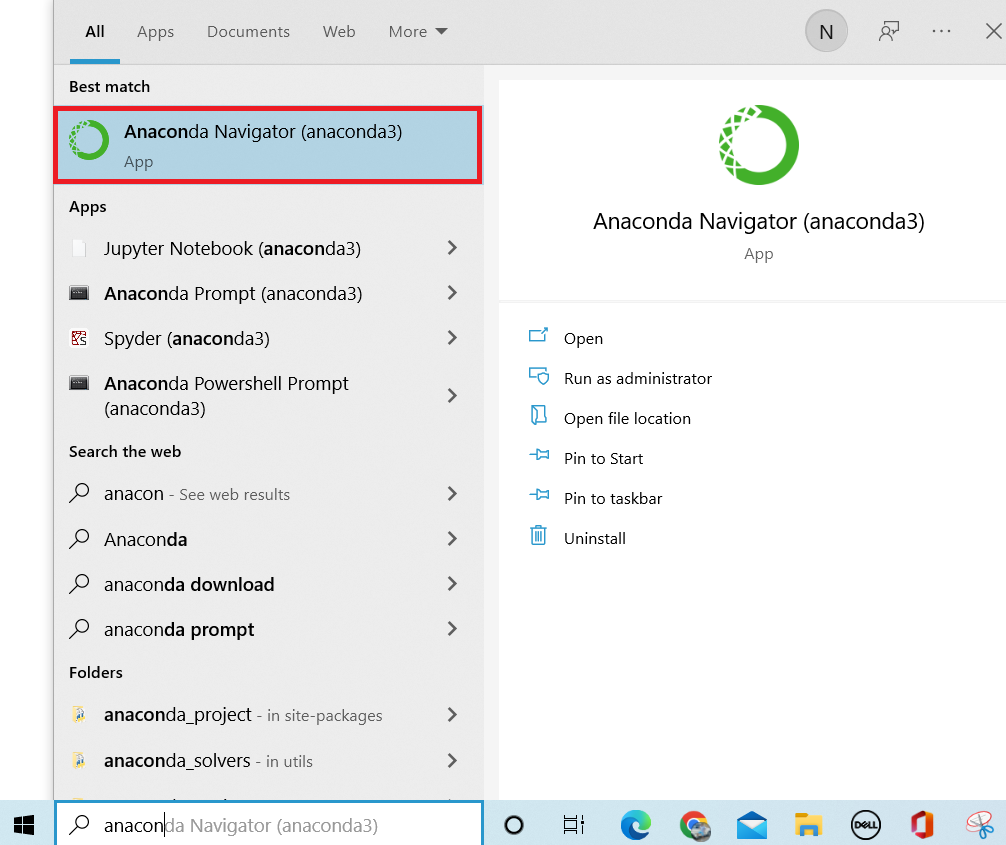


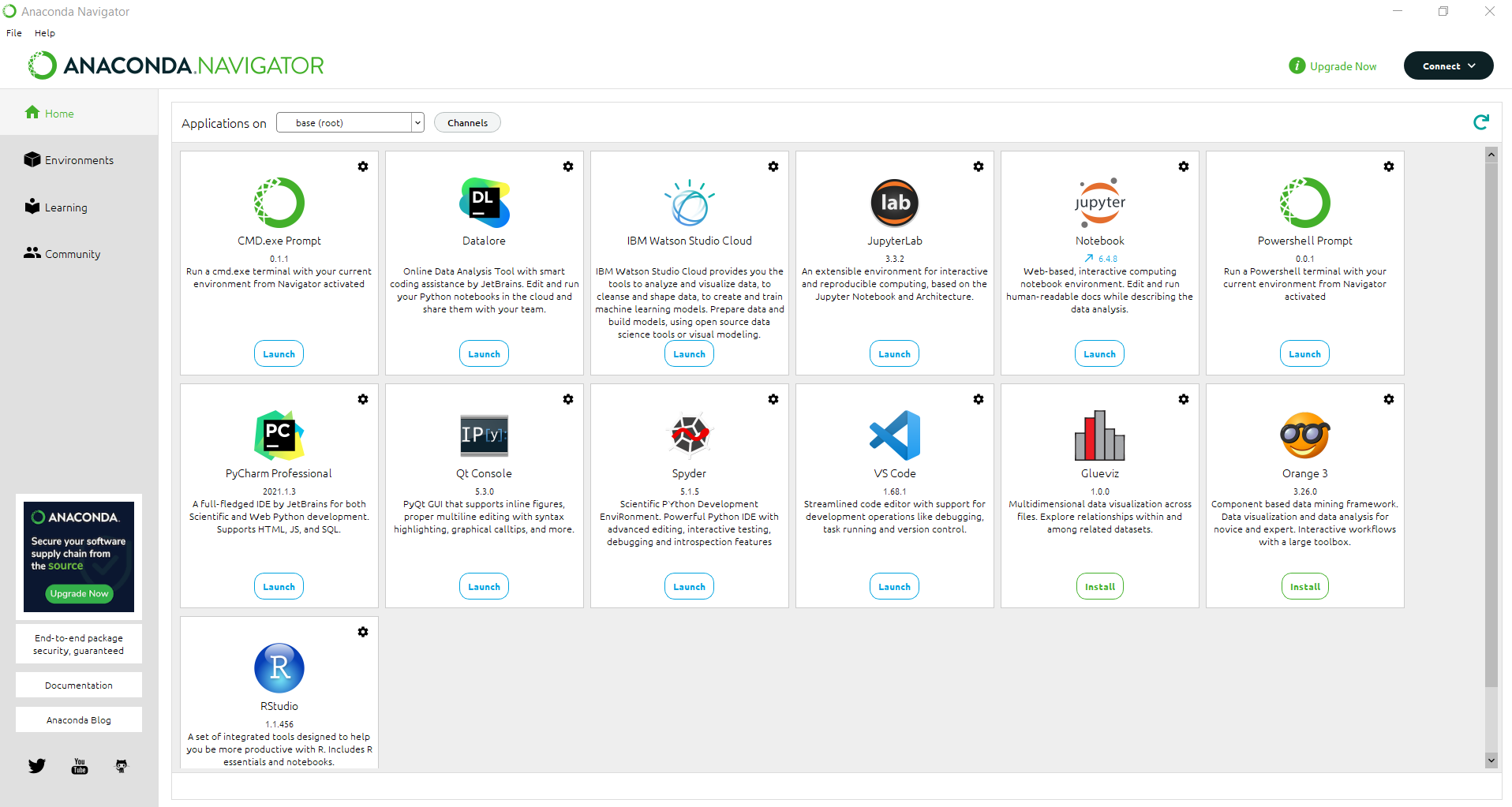
**Exercise 2: Create Anaconda Environment**

<q>Anaconda environment is a directory containing a specific collection of conda packages you have installed. For example, you may have one environment with NumPy 1.7 and its dependencies and another environment with NumPy 1.6 for legacy testing.</q>

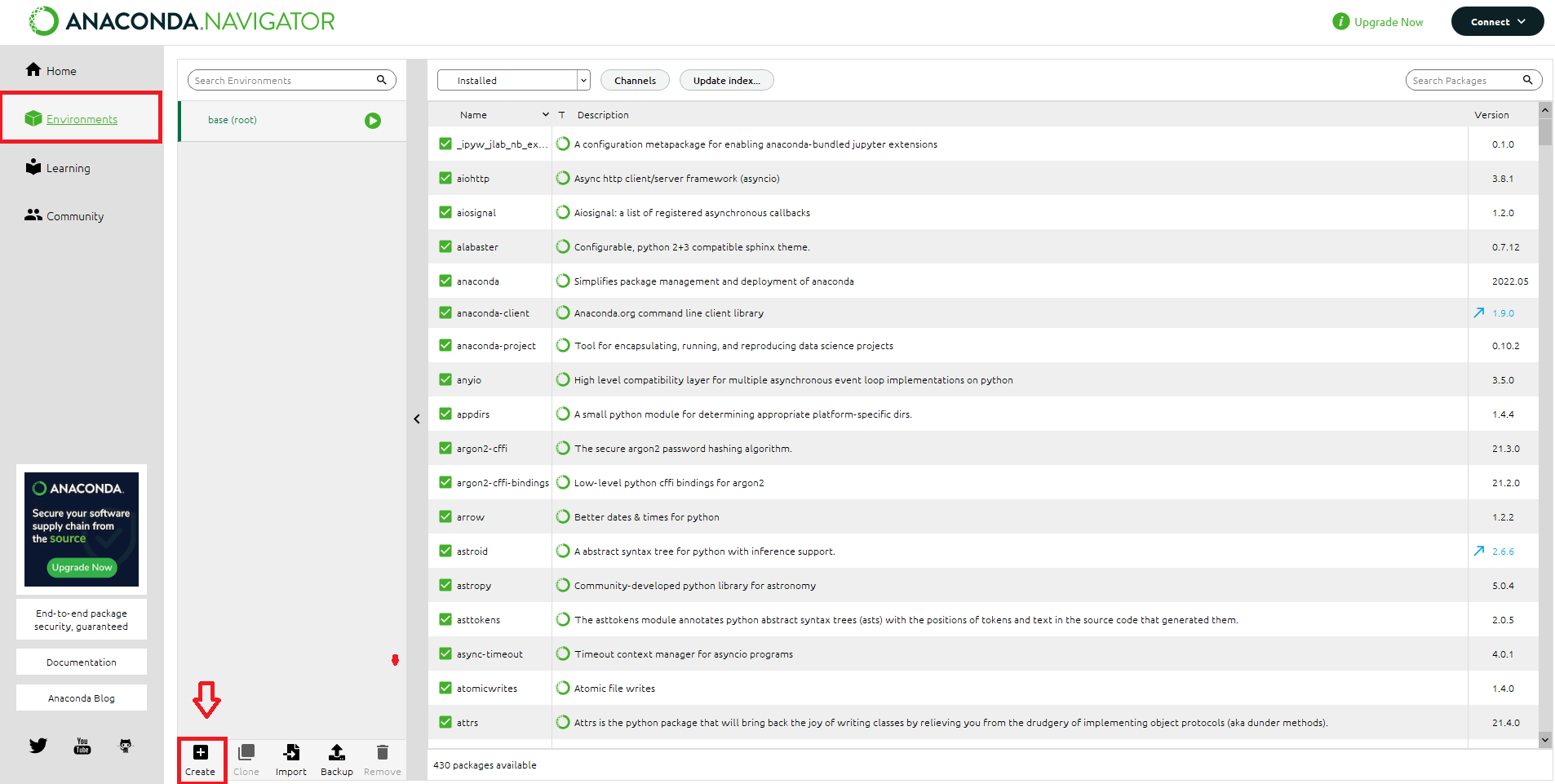
**Ref**: <https://conda.io/projects/conda/en/latest/user-guide/concepts/environments.html>

**Step 1**: Open the **Anaconda Navigator** from the Windows Start menu.



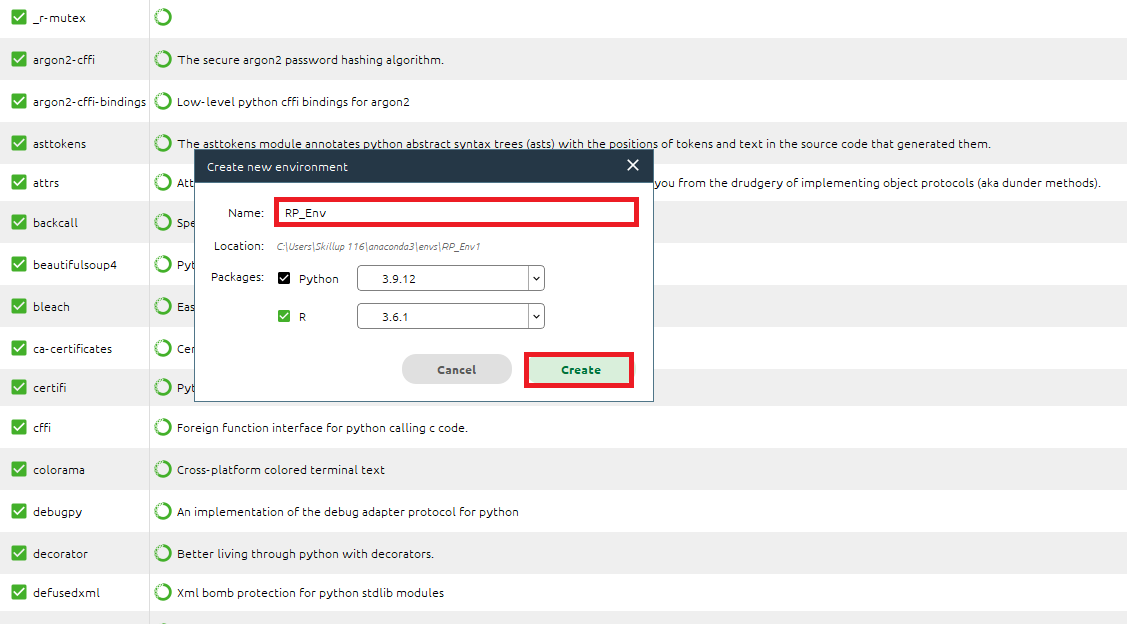


**Step 2**: Create an environment using Anaconda Navigator. Go to the **Environments** tab and click **Create** (at the bottom menu as highlighted below) to create an icon on the Anaconda environment.

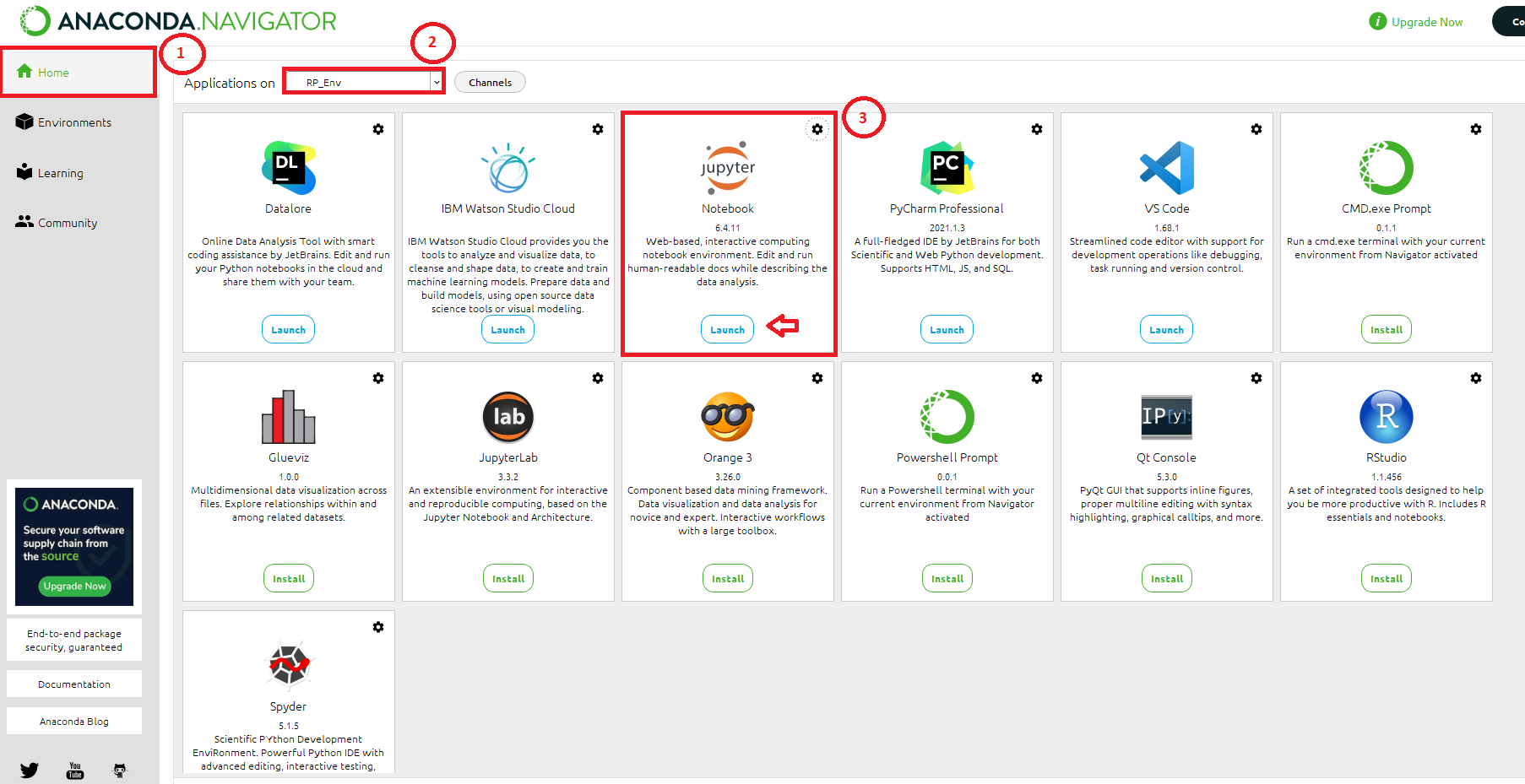


***Note****: It is always helpful to create a separate environment because different projects require different packages.*

**Step 3**: **Give a name** for your environment, select the suitable version and language and click **Create**.



**Step 4**: Once you create an Anaconda environment, go back to the **Home Page** and **Launch Jupyter** and create a **Python Notebook** (make sure to select the right environment).



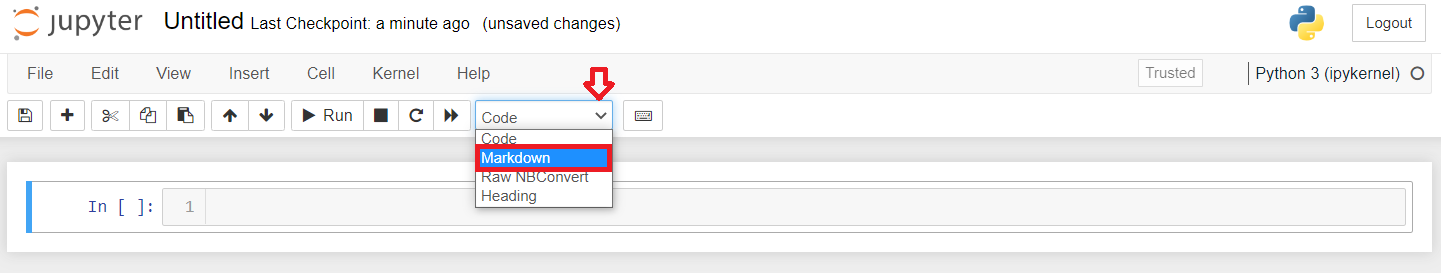
**Step 5**: This opens **Jupyter Notebook** in the default browser, and now you can select the **kernel** and create a **Notebook**.



**Exercise 3: Create and execute Python Jupyter Notebook**

**1. Create markdown cells and add text**

In your notebook, **click any code cell**, and in the drop-down menu, change the cell type from Code to Markdown. You will notice that you cannot create Markdown cells without first creating and converting them from Code to Markdown.



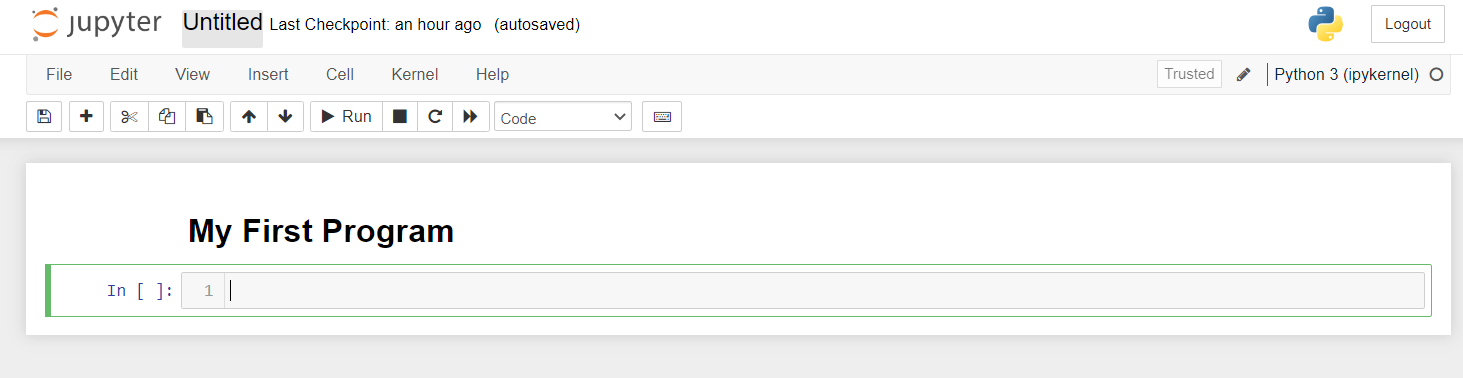
In the Markdown cell, write some text like **My First Program**.

To render the Markdown text, make sure the cell is selected (by clicking within it), and press **Play** in the menu or **Shift+Enter**.

1. 1
2. # My First Program

Copied!

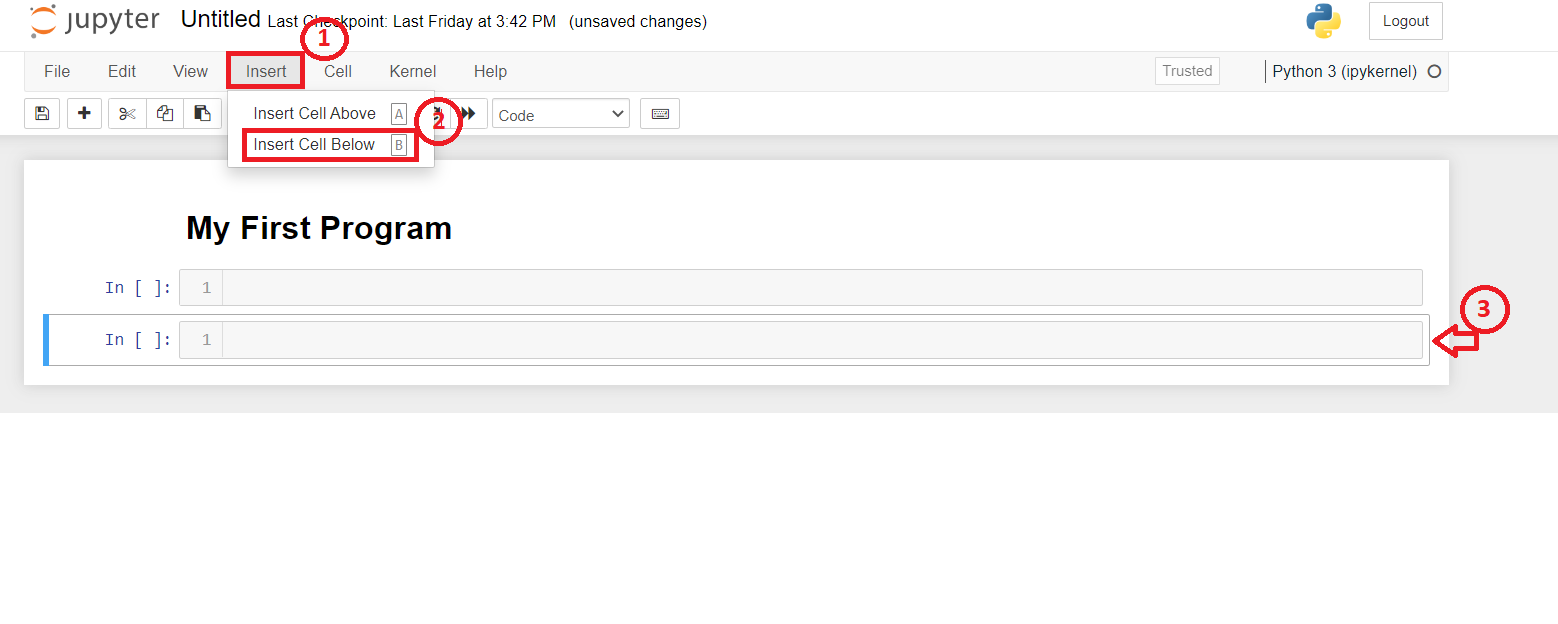
Your Markdown cell should now be rendered!

Output

***Note:****To edit your Markdown cell, double-click anywhere within the cell. Note you can use the keyboard shortcut: [m] - Convert Cell to Markdown.*

**2. Create new cells.**

* In your Jupyter Notebook, click any of the existing cells to select the cell.
* Click **Insert Cell Above** or **Insert Cell Below** to insert the cell from the Insert menu.

Output

***Note:****You can use the keyboard shortcuts: [a] - Insert a Cell Above; [b] - Insert a Cell Below.*

**3. Write and execute code.**

* In your new empty notebook, click within the gray code cell and write some code, like.

1. 1
2. 1+1

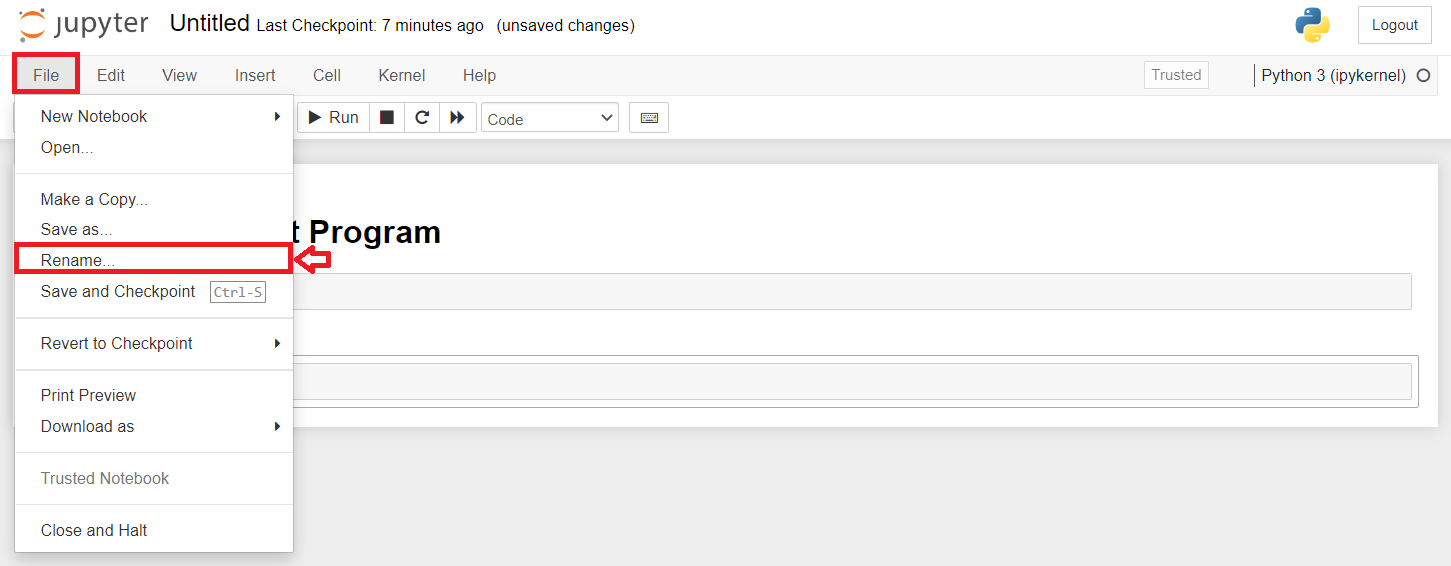
Copied!

* Execute the code by clicking the **Play** button in the menu above the notebook or pressing **Shift+Enter** on your notebook.
* You should see the output 2.

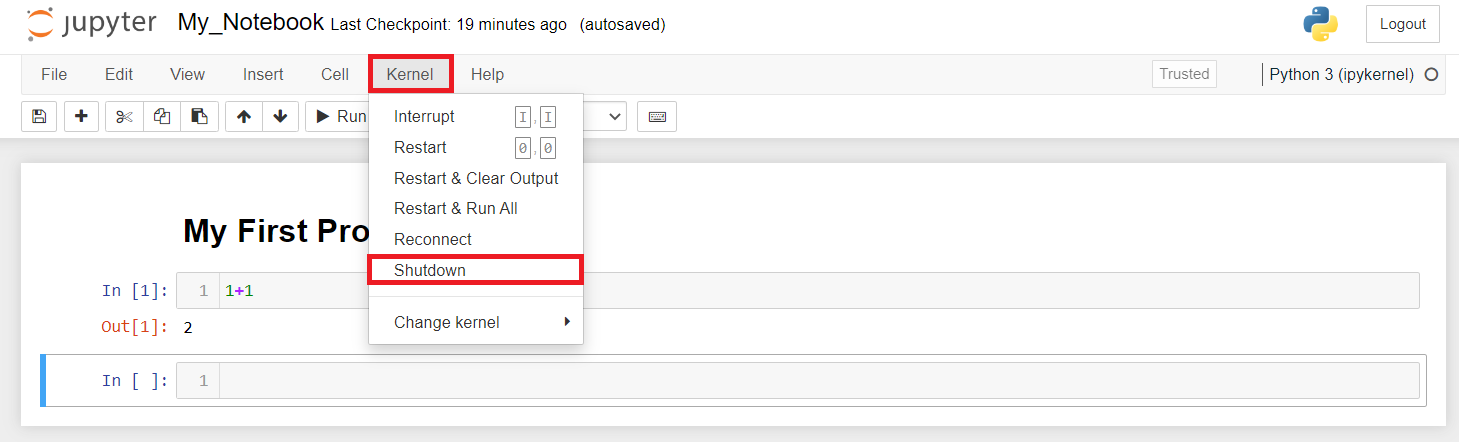
Output

**4. Rename, Shutdown kernel, and Save your Notebook**

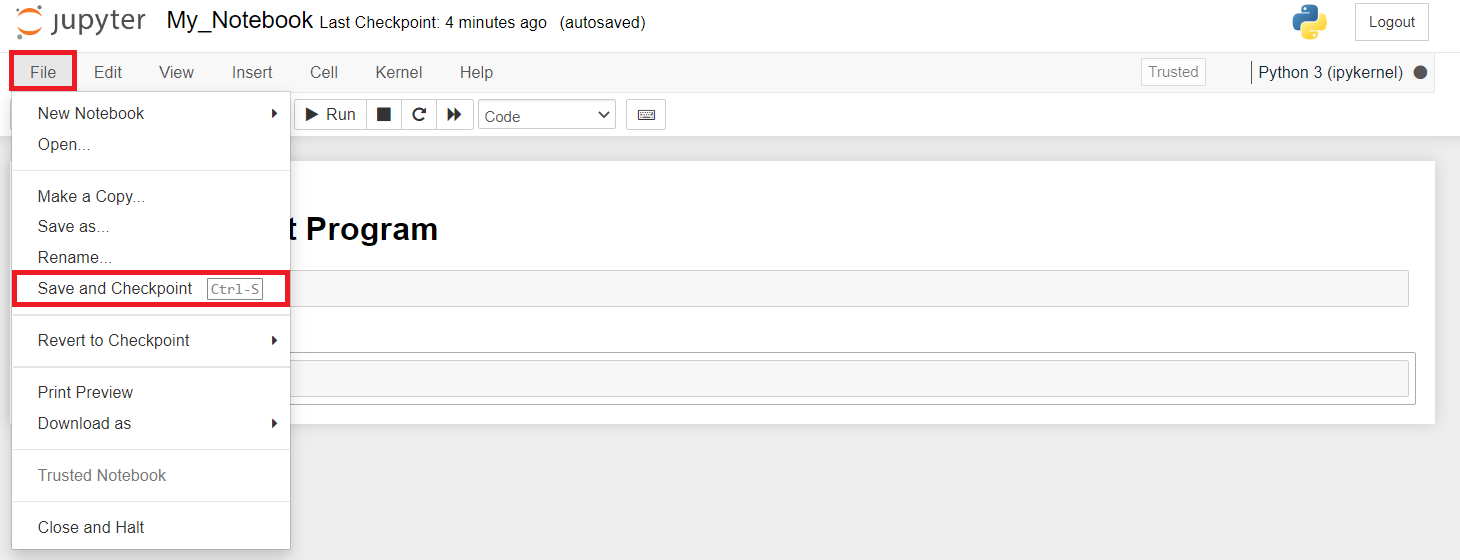
**Step 1**: Click **Rename** from the **File** menu to rename your notebook like ***My\_Notebook.ipynb***.



**Step 2:** To shut down the kernel, click **Shutdown** from the **Kernel** menu.

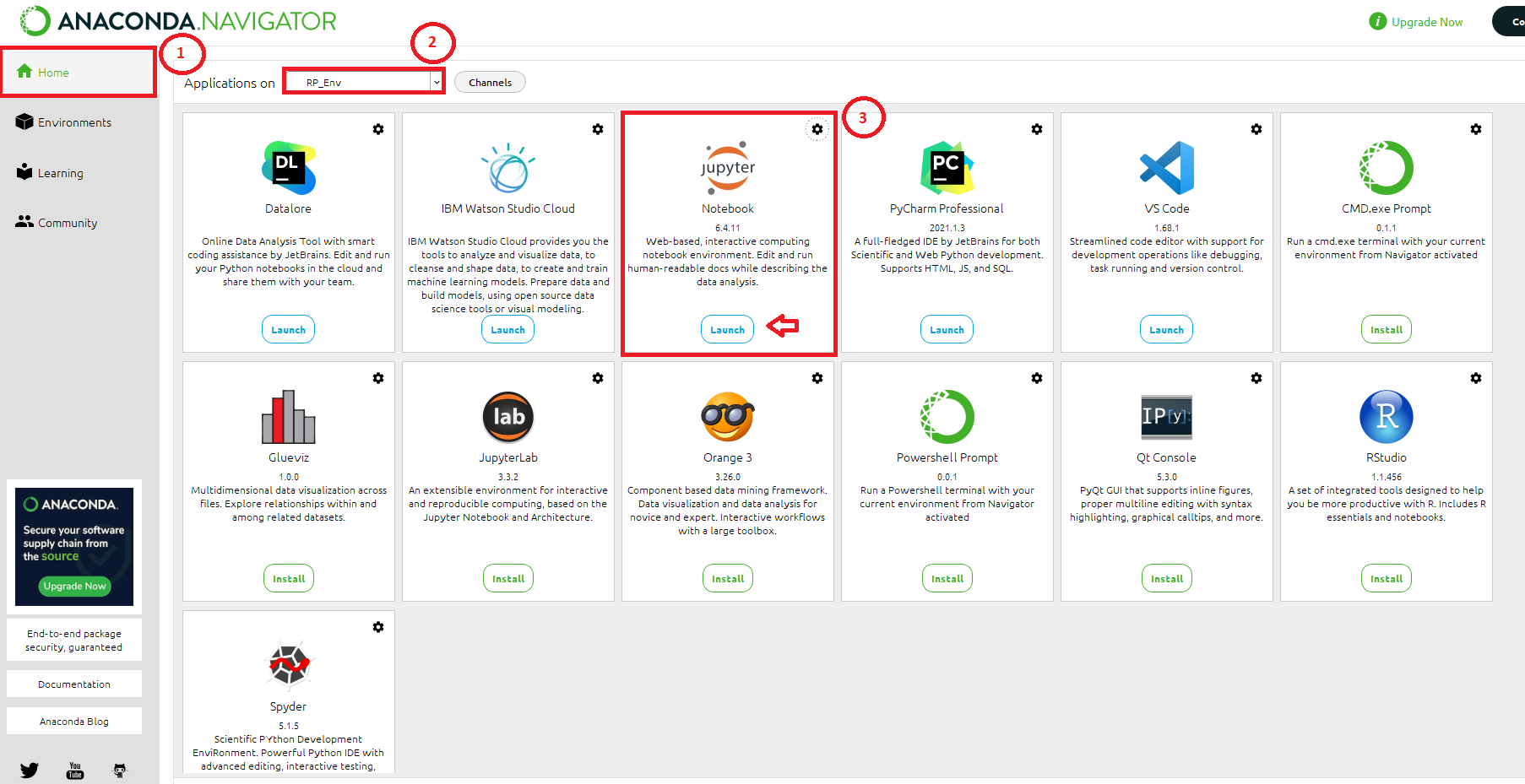


**Step 3:** Click **Save Notebook** or **Save Notebook as** to save the notebook from the **File** menu.

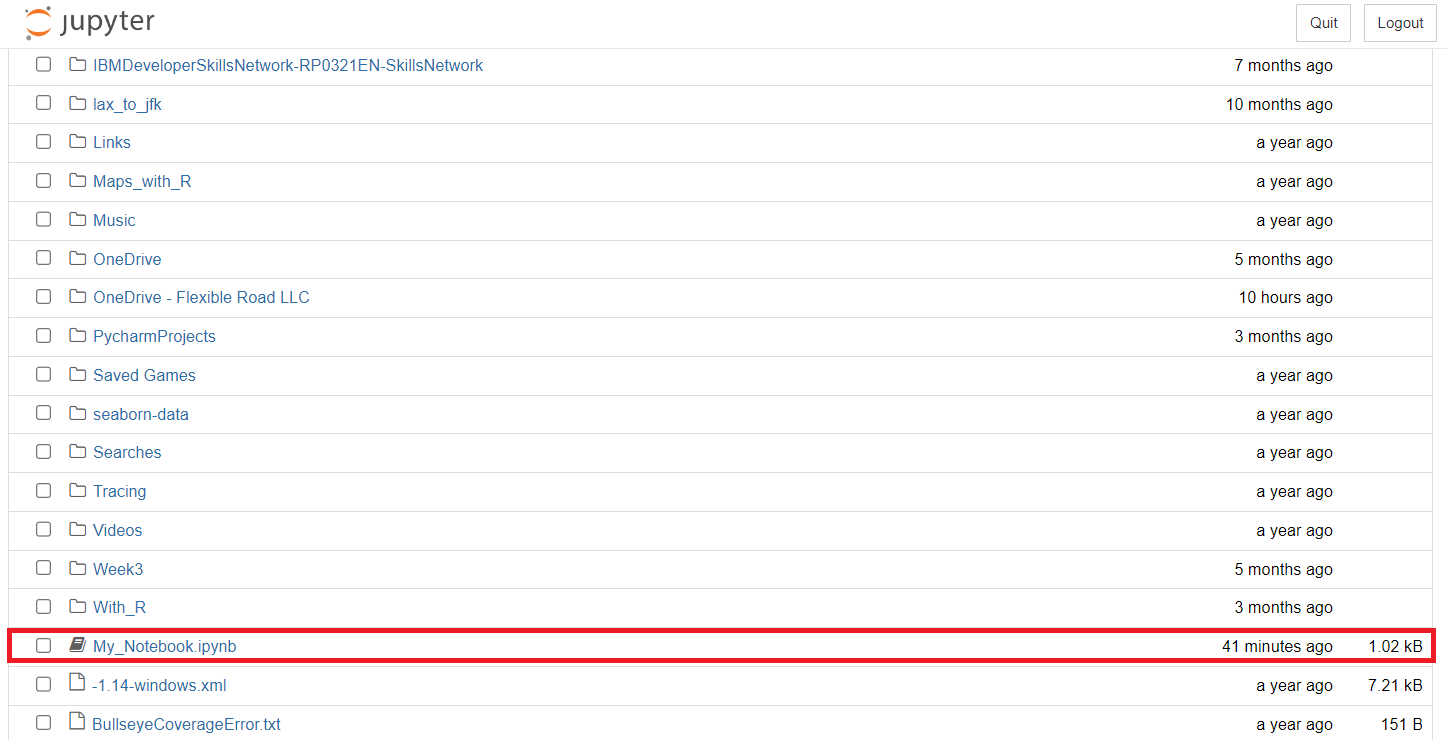


**5. Open the recently created notebook.**

**Step 1**: Open **Anaconda Navigator** from the Windows **Start** menu and **launch Jupyter**.



**Step 2**: Go to the **directory** where you **saved** your file and **click** to open it.



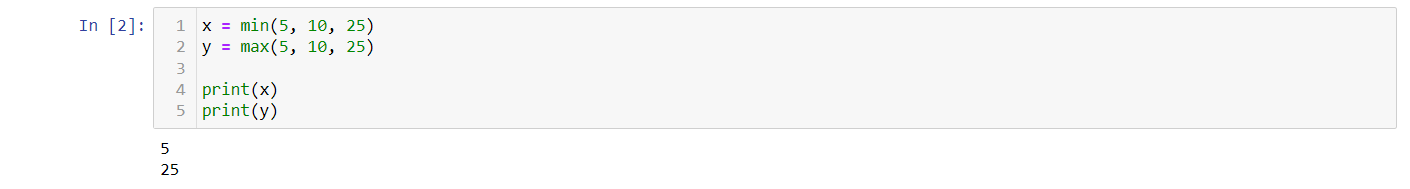
**Practice Exercise**

**Let us try executing simple math operations**

**Problem 1: Find the minimum and maximum values.**

1. 1
2. 2
3. 3
4. 4
5. 5
6. x = min(5, 10, 25)
7. y = max(5, 10, 25)
8. print(x)
9. print(y)

Copied!

Output

**Problem 2: Find the value of 4 to the power 3.**

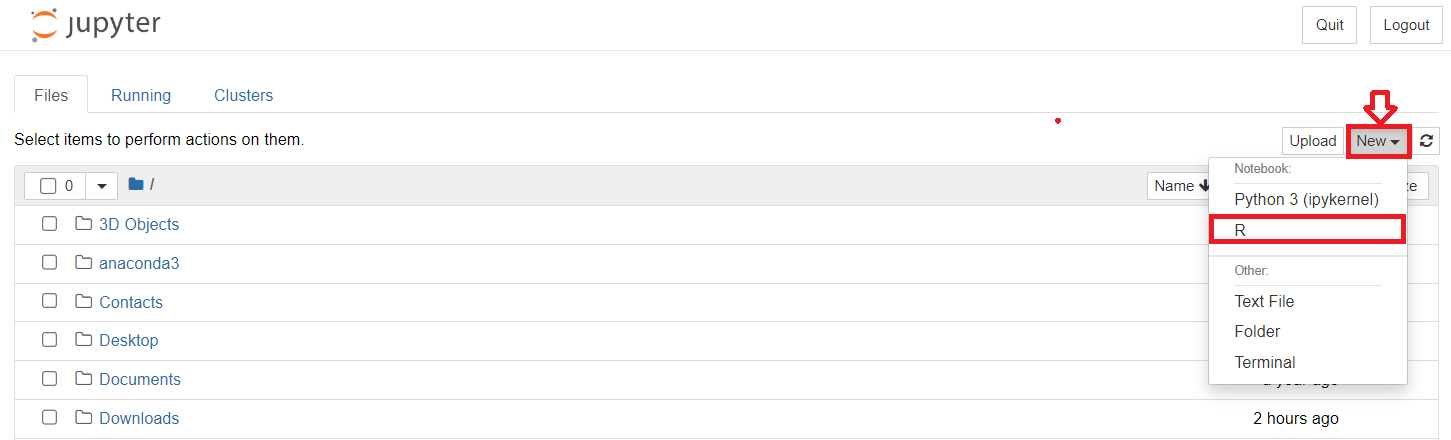
1. 1
2. 2
3. 3
4. x = pow(4, 3)
5. print(x)

Copied!

Output

**Exercise 4: Create and execute R Jupyter Notebook**

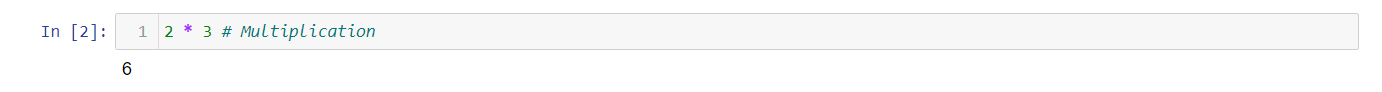
**Select the kernel and create a Notebook.**



**Problem 1: Find the Multiplication of 2 numbers.**

1. 1
2. 2 \* 3 # Multiplication

Copied!

Output

**Problem 2: Find the Subtraction of 2 numbers.**

1. 1
2. 4 - 1 # Subtraction

Copied!

Output

**Problem 3: Add 2 to the given number.**

1. 1
2. 2
3. a <- 1 # Assigning 1 to the variable called "a"
4. a + 2 # Adding 2

Copied!

Output

**Problem 4: Create a data frame**

1. 1
2. 2
3. 3
4. 4
5. 5
6. df = data.frame(Emp\_Name = c("Jai", "David", "Michael"),
7. Job\_role = c("Manager", "Team Lead", "Developer" )
8. )
9. print(df)

Copied!

Output

**Congratulations! You have learned how to download and install Anaconda on your local machine and create a new environment. You have also created a Jupyter Notebook and saved it.**

### Jupyter Notebooks on the Internet

There are thousands of interesting Jupyter Notebooks available on the internet for you to learn from. One of the best sources is: <https://github.com/jupyter/jupyter/wiki>

It is important to notice that you can download such notebooks to your local computer or import them to a cloud based notebook tool so that you can rerun, modify, and apply what's explained in the notebook.

Very often, Jupyter Notebooks are already shared in a rendered view. This means that you can look at them as if they were running locally on your machine. But sometimes, folks only share a link to the Jupyter file (which you can make out by the \*.ipynb extension). In this case, you can pick the URL to that file and paste it to the NB-Viewer => <https://nbviewer.jupyter.org/>

The list of Jupyter Notebooks provides you with a huge collection of materials to explore. Therefore, it might be useful to give you some pointers to interesting notebooks. You have covered some examples with data in the labs. Let's highlight some useful data that further explores data science. In addition, as we have covered different tasks in data science, we will also provide a sample notebook for each task.

First, you start with exploratory data analysis, for which this notebook is highly recommended: <https://nbviewer.jupyter.org/github/Tanu-N-Prabhu/Python/blob/master/Exploratory_data_Analysis.ipynb>

For data integration/cleansing at a smaller scale, the python library\_pandas\_is often used. For this task, you can have a look at this notebook: <https://towardsdatascience.com/data-cleaning-with-python-using-pandas-library-c6f4a68ea8eb>

If you want to know more about clustering, have a look at this notebook: <https://nbviewer.jupyter.org/github/temporaer/tutorial_ml_gkbionics/blob/master/2%20-%20KMeans.ipynb>

And finally, if you want an in-depth notebook on the\_iris\_dataset, have a look at this: <https://www.kaggle.com/lalitharajesh/iris-dataset-exploratory-data-analysis>

# Module 4 Summary

Congratulations! You have completed this module. At this point in the course, you know:

* Jupyter Notebooks are used in Data Science for recording experiments and projects.
* Jupyter Lab is compatible with many files and Data Science languages.
* There are different ways to install and use Jupyter Notebooks.
* How to run, delete, and insert a code cell in Jupyter Notebooks.
* How to run multiple notebooks at the same time.
* How to present a notebook using a combination of Markdown and code cells.
* How to shut down your notebook sessions after you have completed your work on them.
* Jupyter implements a two-process model with a kernel and a client.
* The notebook server is responsible for saving and loading the notebooks.
* The kernel executes the cells of code contained in the Notebook.
* The Jupyter architecture uses the NB convert tool to convert files to other formats.
* Jupyter implements a two-process model with a kernel and a client.
* The Notebook server is responsible for saving and loading the notebooks.
* The Jupyter architecture uses the NB convert tool to convert files to other formats.
* The Anaconda Navigator GUI can launch multiple applications on a local device.
* Jupyter environments in the Anaconda Navigator include JupyterLab and VS Code.
* You can download Jupyter environments separately from the Anaconda Navigator, but they may not be configured properly.
* The Anaconda Navigator GUI can launch multiple applications.
* Additional open-source Jupyter environments include JupyterLab, JupyterLite, VS Code, and Google Colaboratory.
* JupyterLite is a browser-based tool.