

Intro to Data Analysis

7 – 8 October 2024



2-day Course Plan

- **Day 1 (7 Oct 2024)**


- Anaconda walk-through
- Python basics
 - Variables, Operators
 - Built-in scalar types, data structures
 - Control flow statements
 - Functions
 - Errors and exceptions
 - Iterators
 - String manipulation
 - Modules and packages
- Numpy
 - Array operations
 - Indexing and slicing
 - Linear algebra

- **Day 2 (8 Oct 2024)**

- Pandas
 - Dataframe indexing and selection
 - Data cleaning and handling missing values
 - Data manipulation
 - Grouping and aggregation
- SciPy
 - SciPy Constants
 - Basic Linear Algebra
 - Optimization Techniques
 - Statistical Functions and Distributions
- Exploratory data analysis (EDA)
 - Reading data
 - Identifying missing values
 - Data cleaning
 - Extracting relevant features



Introduction to Data Analysis

1. The workshop involves exploring a real-life dataset, focusing on conducting exploratory data analysis (EDA) using essential libraries like Matplotlib, Seaborn, Pandas, Scikit-learn, and NumPy in Python.
 2. Learning Outcome: Identify potential problems within the data and subsequently design and implement a machine learning solution using Scikit-learn to address the identified issues.
- 

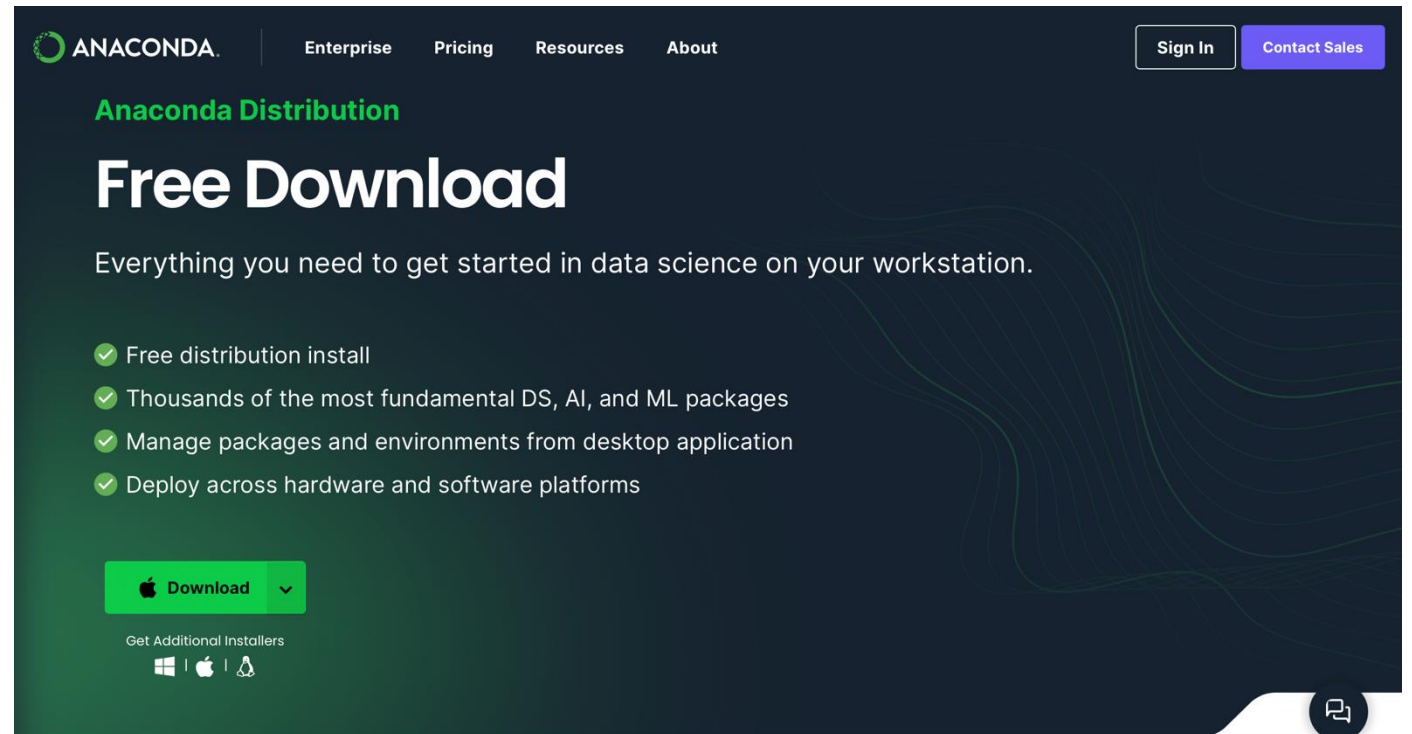
Installation & Practical Considerations

- Installing Python and the suite of libraries that enable scientific computing is pretty much straight-forward whether you use Windows or MacOS.
- Though there are various ways to install Python, the one that is recommended particularly for data science and analytics is via the cross-platform Anaconda distribution.



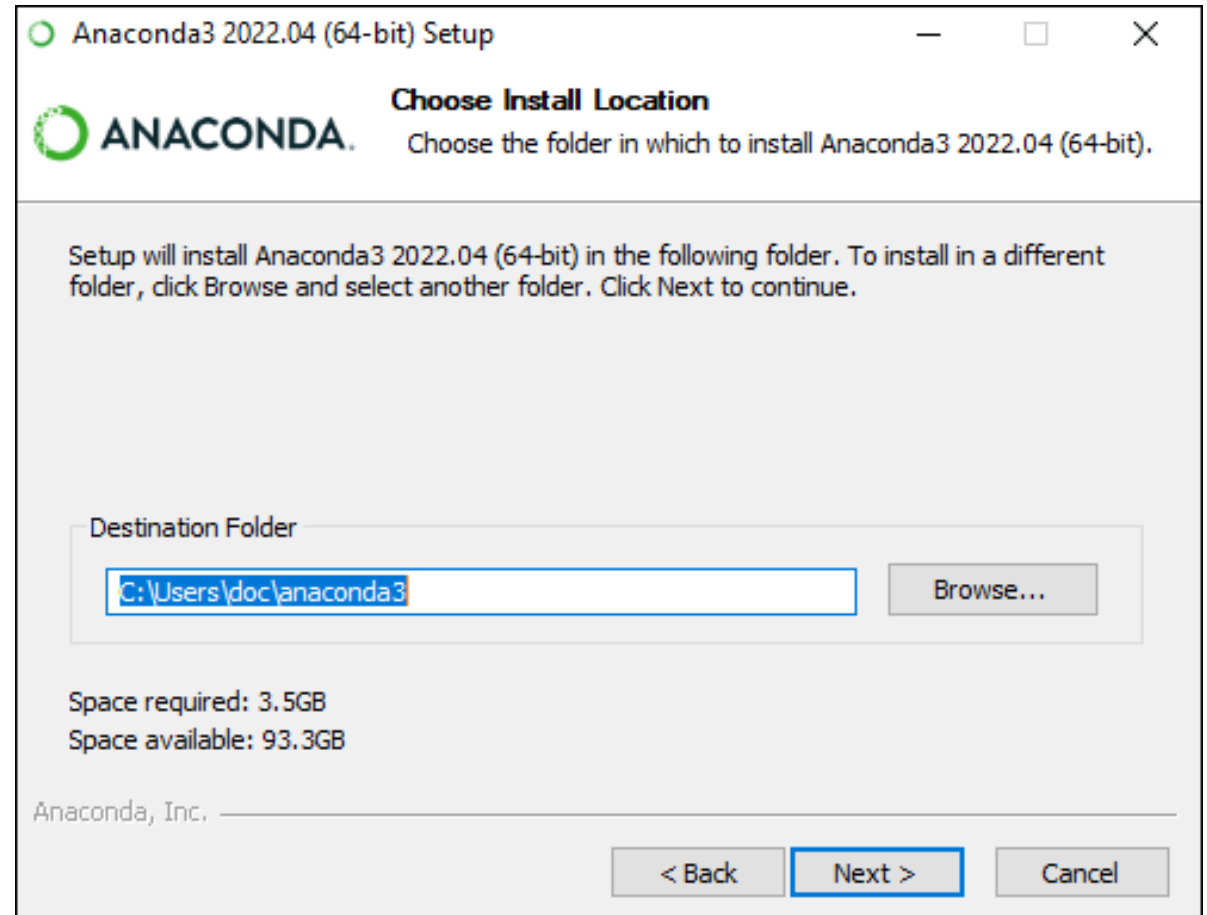
Download & Install Anaconda

- Go to the <https://www.anaconda.com/download> and download the installer.
 - The installer is pre-selected based on your platform.
- Save the installer to an easy location to locate (i.e. Desktop).



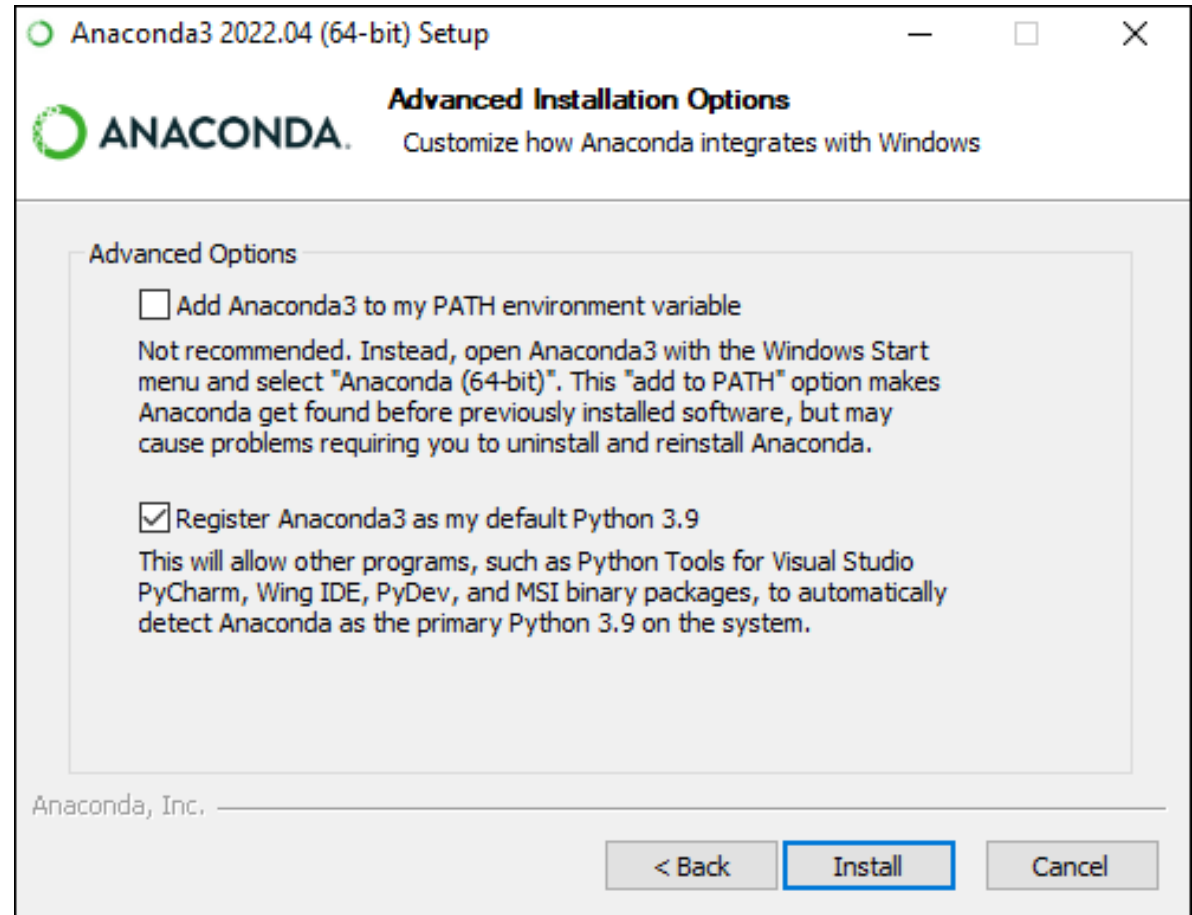
Download & Install Anaconda (Windows)

- Double-click the installer to launch.
 - To prevent permission errors, do not launch the installer from the **Favorites** folder.
- Click **Next**.
- Read the licensing terms and click **I Agree**.
- Select an install for **Just Me** unless you're installing for all users (which requires Windows Administrator privileges) and click **Next**.
- Select a destination folder to install Anaconda and click the Next button.
 - DO NOT install as Administrator unless admin privileges are required.



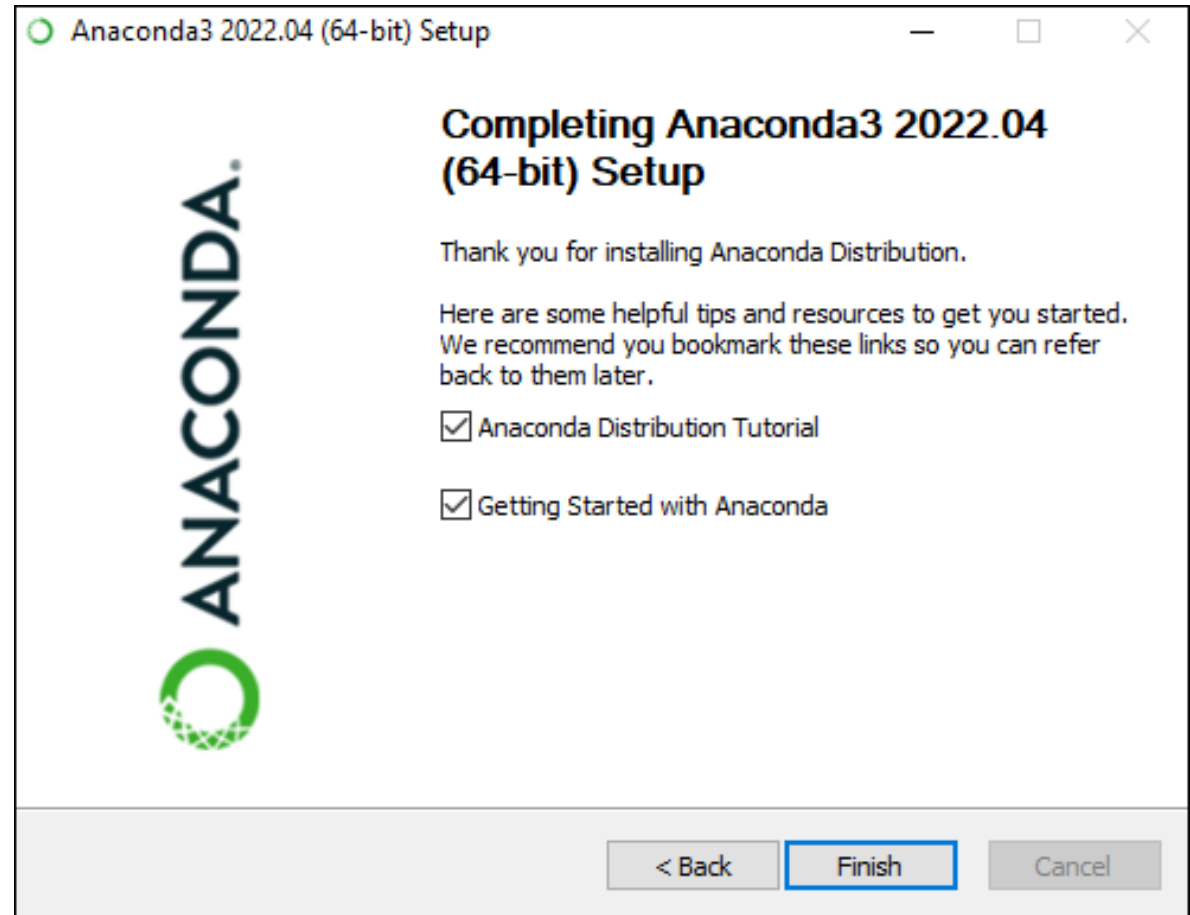
Download & Install Anaconda

- Choose whether to add Anaconda to your PATH environment variable.
- The recommended approach is NOT to add Anaconda to the PATH environment variable, since this can interfere with other software.
- Instead, use Anaconda software by opening **Anaconda Navigator** or the **Anaconda Prompt** from the Start Menu.



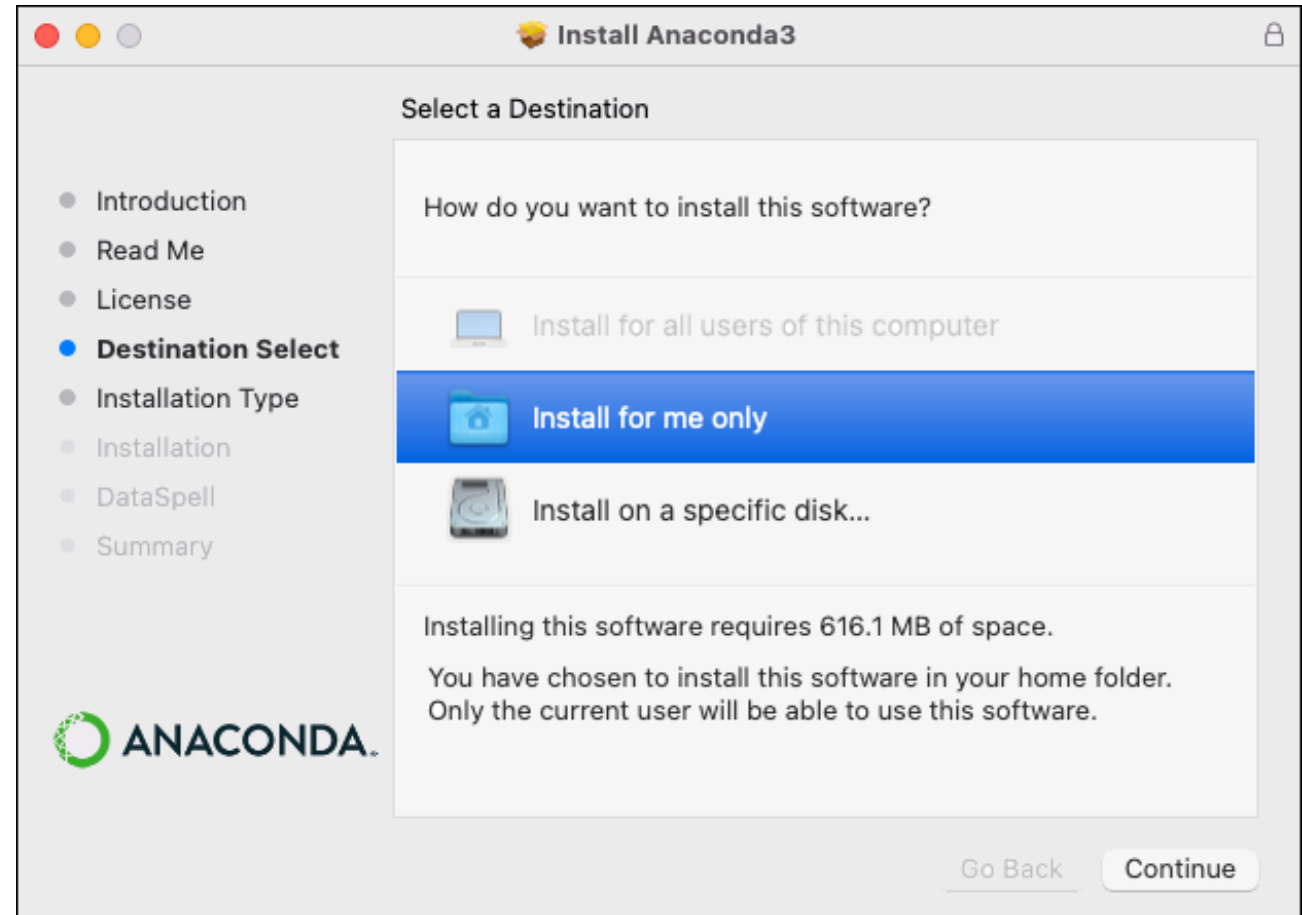
Download & Install Anaconda

- Choose whether to register Anaconda as your default Python.
- Unless you plan on installing and running multiple versions of Anaconda or multiple versions of Python, accept the default and leave this box checked.
- Click **Install**. If you want to watch the packages Anaconda is installing, click Show Details.
- Click **Next**.
- After a successful installation you will see the “Thanks for installing Anaconda” dialog box.

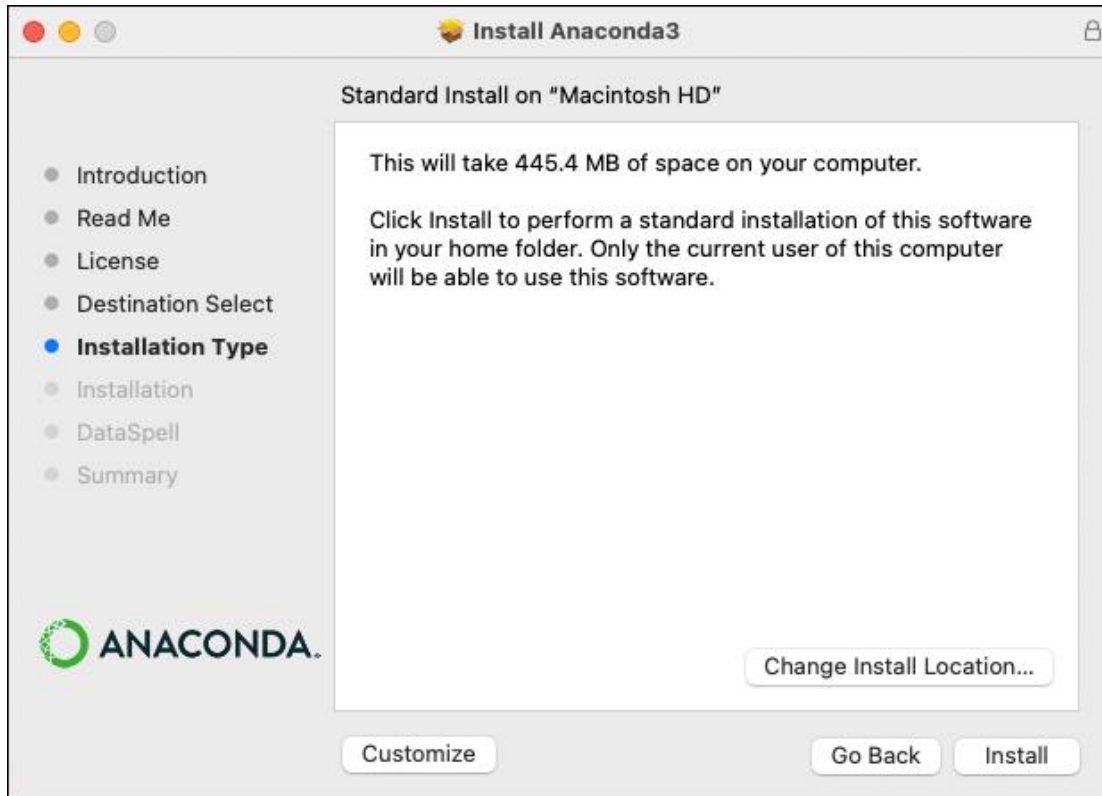


Download & Install Anaconda (MacOS)

- Double-click the downloaded file and click **Continue** to start the installation.
- Answer the prompts on the Introduction, Read Me, and License screens.
- You can click **Change Install Location** to install Anaconda Distribution for all users or on a specific disk.
- This is not recommended. **Install for me only** is the default (and recommended) selection. Click **Continue** to return to Installation Type.
 - If you get the error message “You cannot install Anaconda in this location,” reselect **Install for me only**.



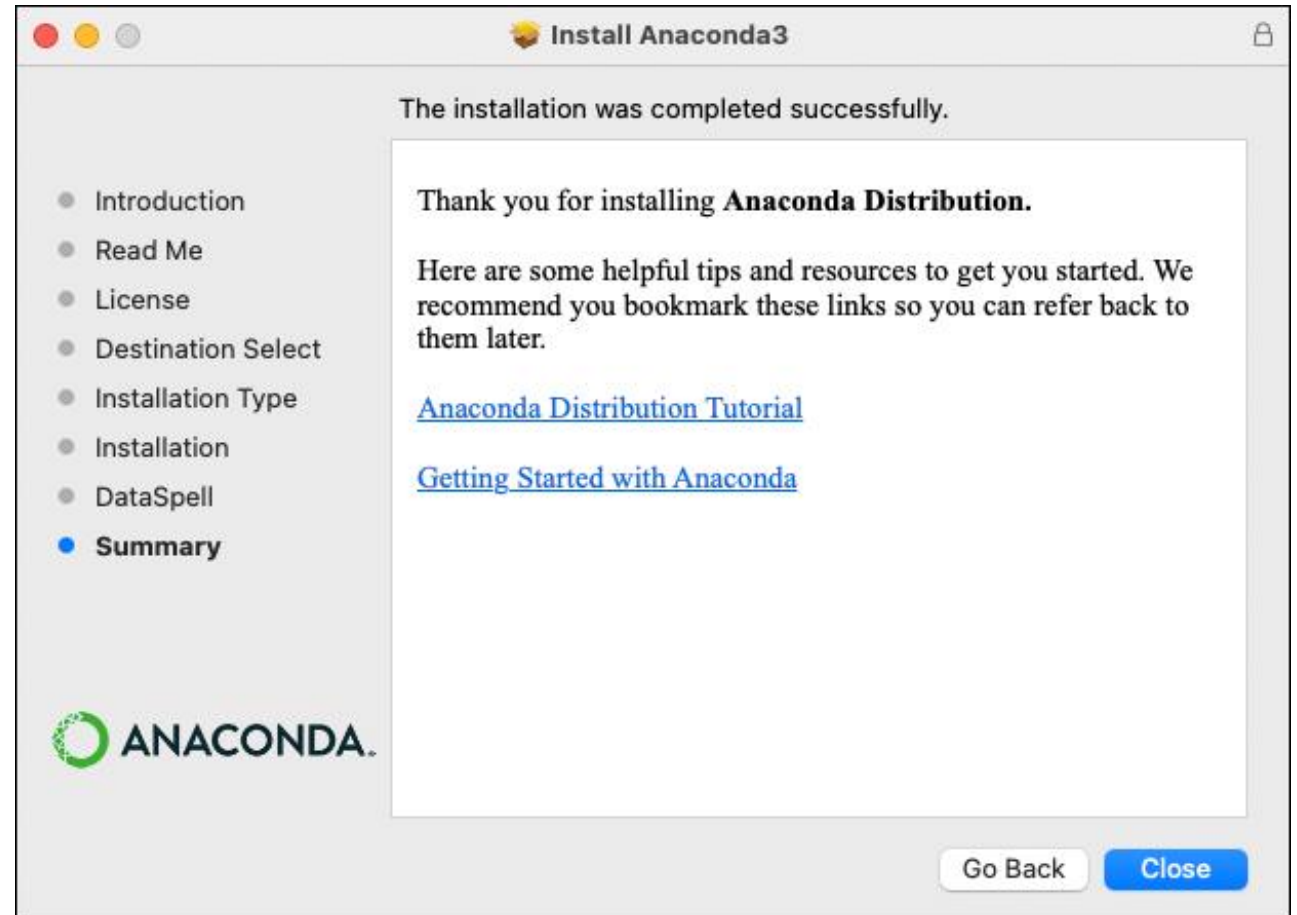
Download & Install Anaconda (MacOS)



- Click **Install** to install Anaconda in your ~/opt directory (recommended).
- Once the install is complete, click **Continue**.
- Since we will be using different IDE for programming in Python, we can skip DataSpell installation for now (click **Continue** on DataSpell screen).

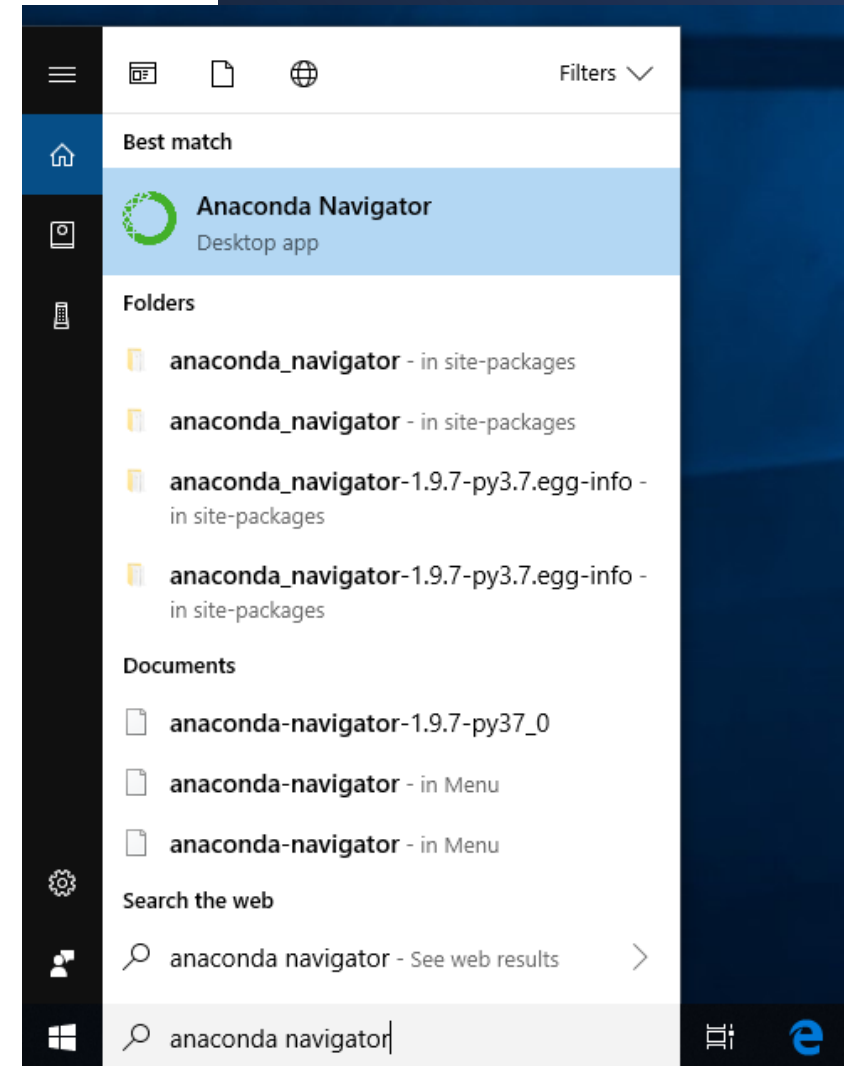
Download & Install Anaconda (MacOS)

- A successful installation displays the following screen:



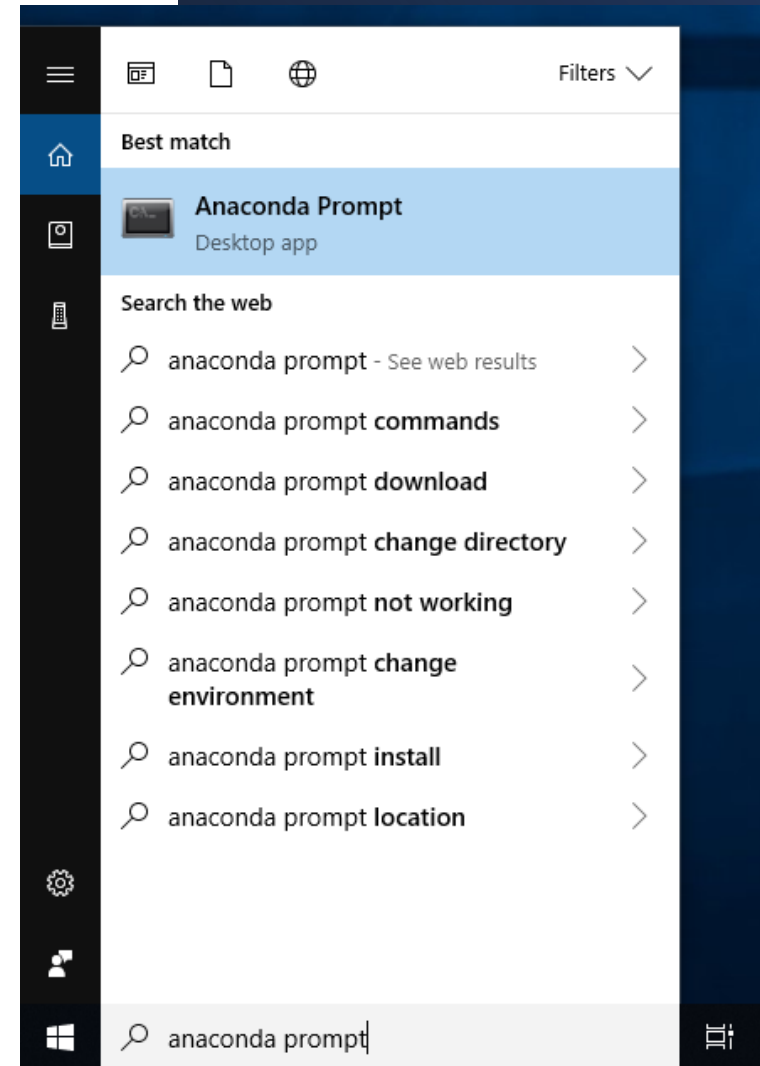
Verifying Installation: Anaconda Navigator

- You can confirm that Anaconda is installed and working with Anaconda Navigator or `conda`.
- Anaconda Navigator is a graphical user interface that is automatically installed with Anaconda.
- Navigator will open if the installation was successful.
- Windows: Click **Start**, search or select **Anaconda Navigator** from the menu.
- macOS: Click Launchpad, select Anaconda Navigator.



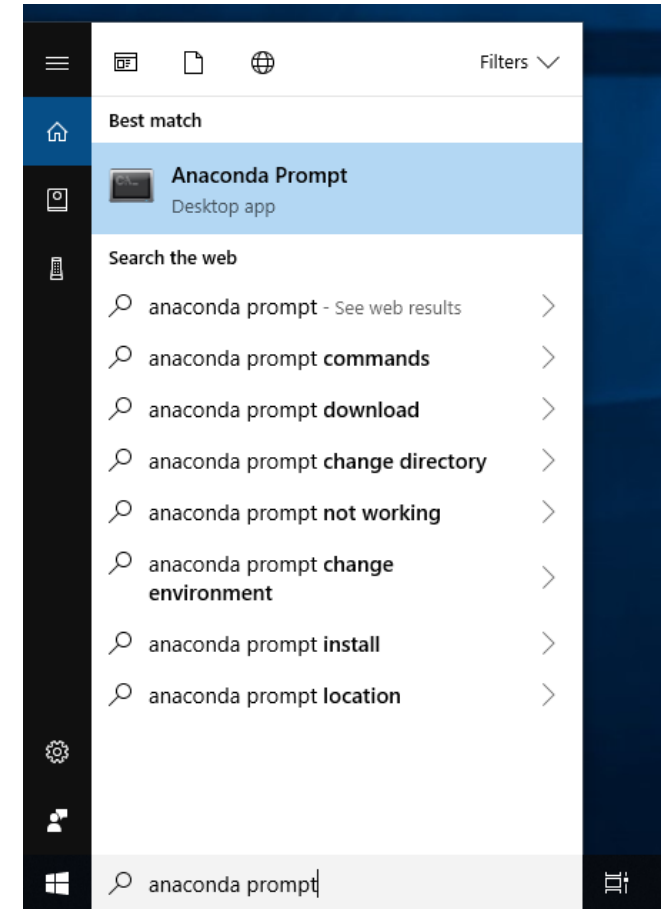
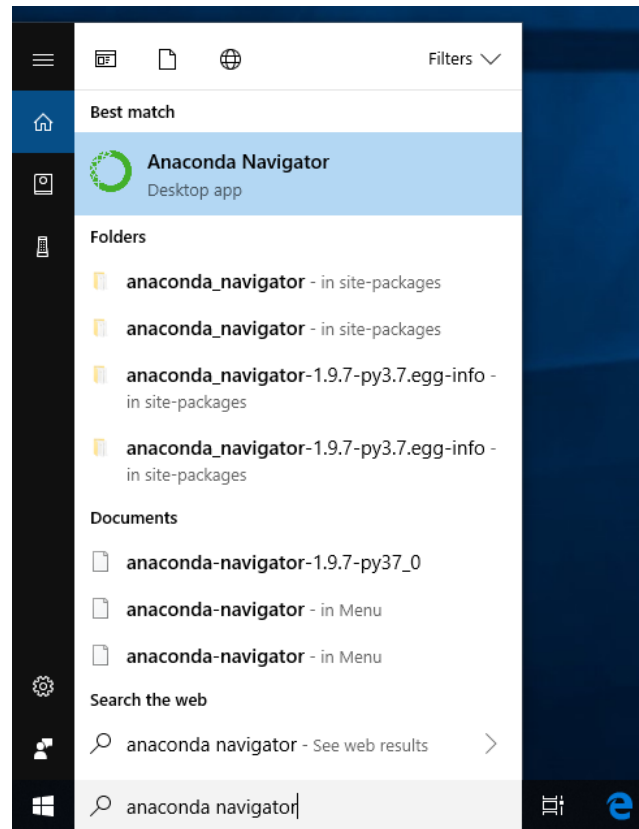
Verifying Installation: Conda

- You can use `conda` to verify the installation using Anaconda Prompt on Windows or terminal on macOS.
- To open Anaconda Prompt:
 - Windows: Click **Start**, search, or select **Anaconda Prompt** from the menu.
 - macOS: Cmd+Space to open **Spotlight Search** and type "Terminal" to open the program.
- Choose any of the following methods to verify:
 1. Enter `conda --version`. This will display the installed Anaconda version
 2. Enter the command `python`. This command runs the Python shell. If Anaconda is installed and working, the version information it displays when it starts up will include "Anaconda". To exit the Python shell, enter the command `quit()`.



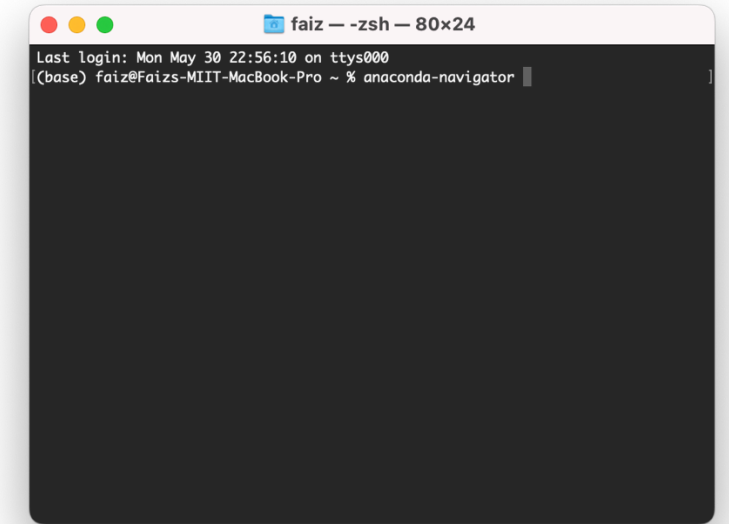
Starting Navigator on Windows

- Windows
 - From the Start menu, click the Anaconda Navigator desktop app.
 - Or from the Start menu, search for and open “Anaconda Prompt” and type the command `anaconda-navigator`.



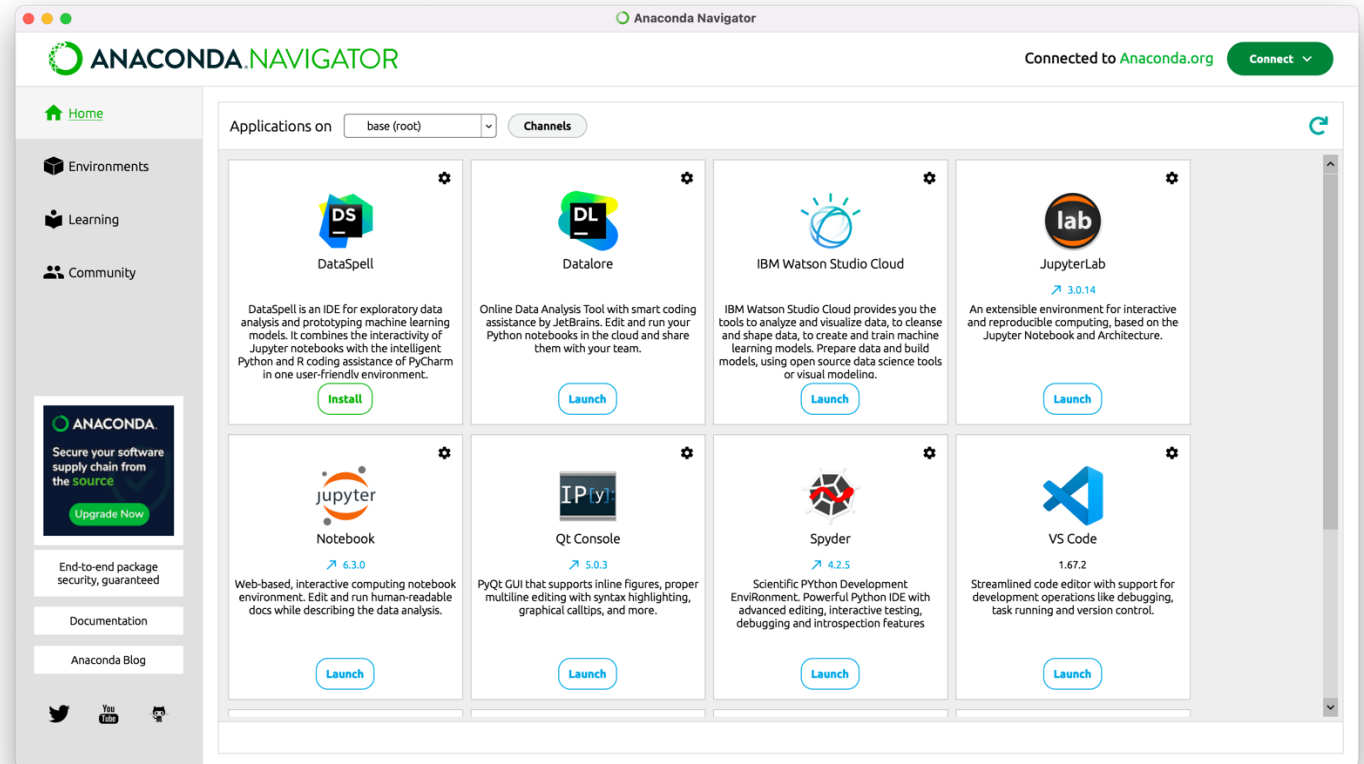
Starting Navigator on macOS

- MacOS
 - Open Launchpad, then click the Anaconda-Navigator icon.
 - Or open Launchpad and click the terminal icon. Then in terminal, type anaconda-navigator.



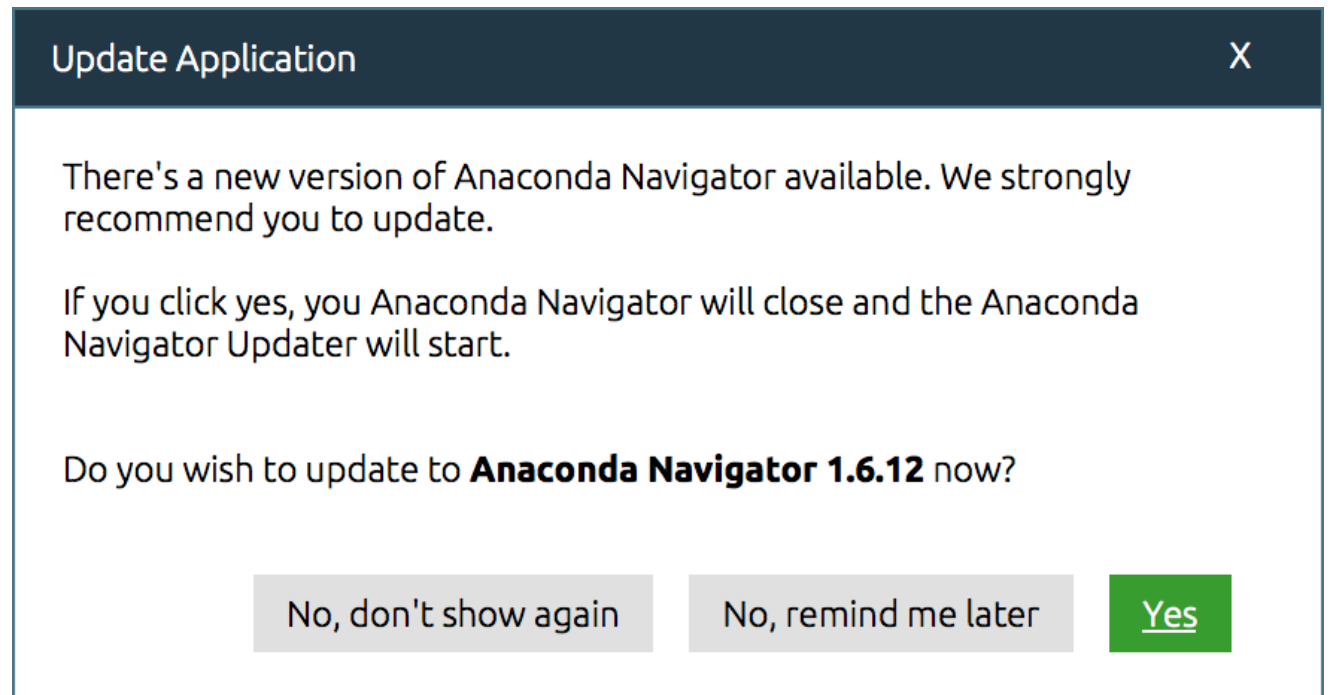
Anaconda Navigator: Overview

- Anaconda Navigator is a graphical user interface to the conda package and environment manager.



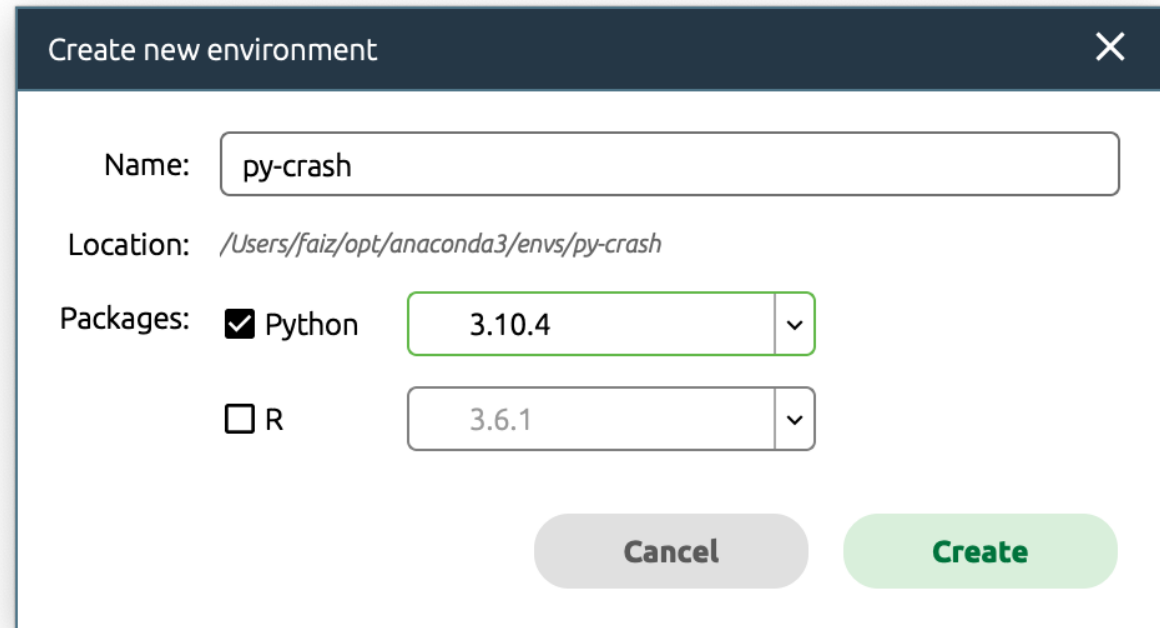
Managing Navigator

- Verify that Anaconda is installed and running on your system.
 - When Navigator starts up, it verifies that Anaconda is installed.
 - If Navigator does not start up, go back to Anaconda installation and make sure you followed all the steps.
- Check that Navigator is updated to the current version.
 - When you start Navigator, it automatically checks for a new version. If Navigator finds a new version, you will see a dialog box like this:
- Click the “Yes” button to update Navigator to the current version.



Managing Environments

- Navigator uses conda to create separate environments containing files, packages, and their dependencies that will not interact with other environments.
- Create a new environment named `py-crash` and install a package in it:
 1. In Navigator, click the **Environments** tab, then click the Create button. The **Create new environment** dialog box appears.
 2. In the **Environment** name field, type a descriptive name for your environment.
 3. Click **Create**. Navigator creates the new environment and activates it.

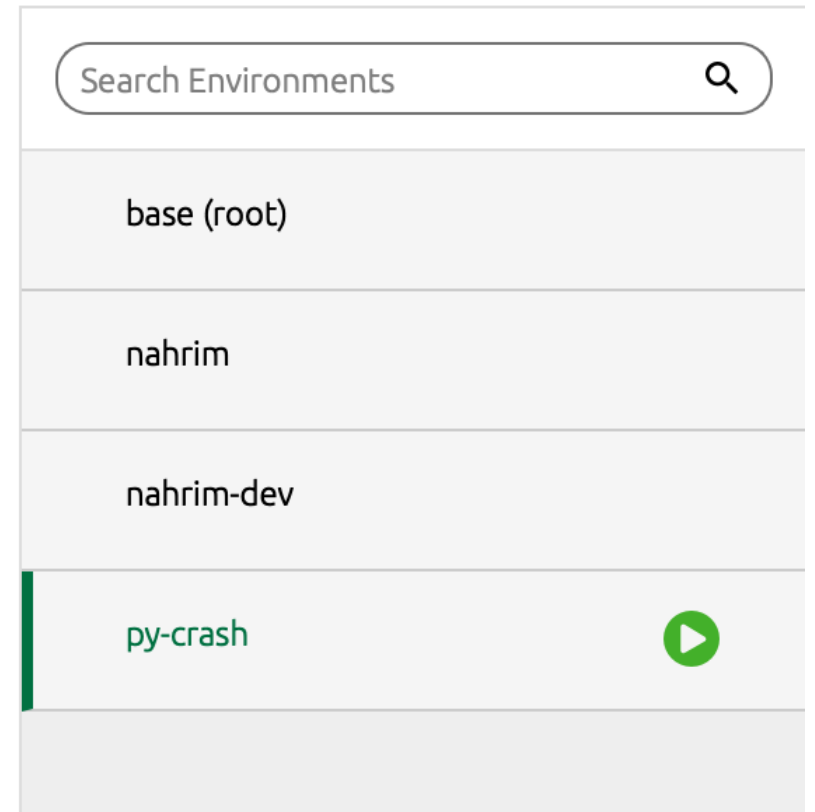


The screenshot shows a dialog box titled "Create new environment" with a close button (X) in the top right corner. The dialog contains the following fields and options:

- Name:** A text input field containing "py-crash".
- Location:** A text input field containing the path `/Users/faiz/opt/anaconda3/envs/py-crash`.
- Packages:** Two rows of package selection:
 - Row 1: A checked checkbox next to "Python", followed by a dropdown menu showing "3.10.4".
 - Row 2: An unchecked checkbox next to "R", followed by a dropdown menu showing "3.6.1".
- Buttons:** At the bottom right, there are two buttons: "Cancel" (a light gray button) and "Create" (a green button with white text).

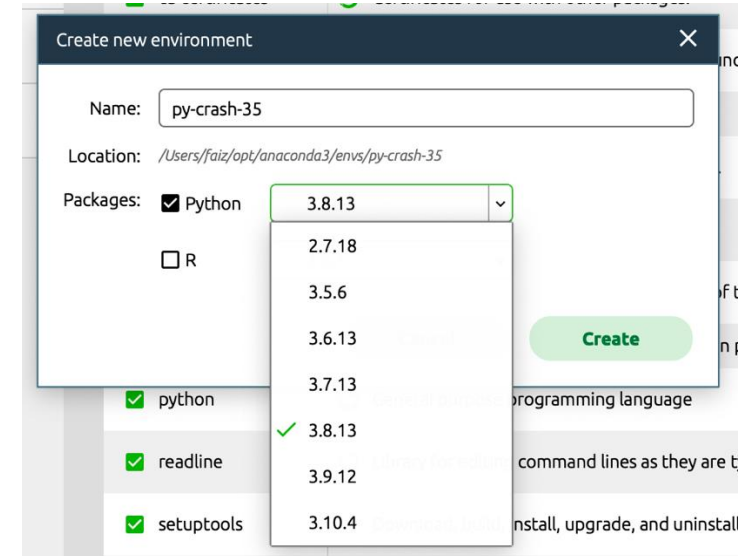
Managing Environments

- Now you have multiple environments, the default environment base (root), several others and the newly created `py-crash`.
- Switch between them (activate and deactivate environments) by clicking the name of the environment you want to use.
- Return to the other environment by clicking its name.

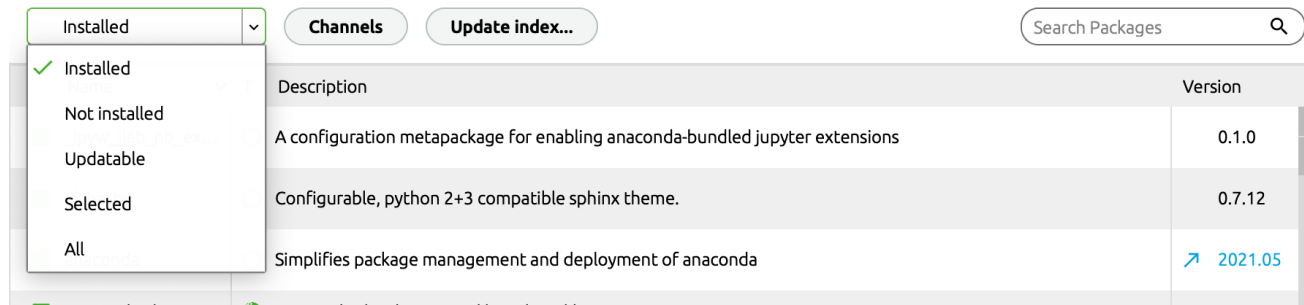
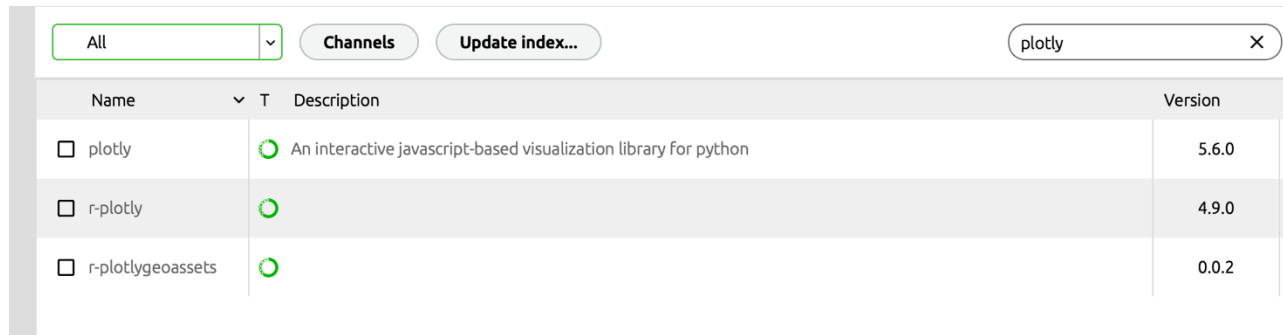


Managing Python

- When you create a new environment, Navigator installs the same Python version you used when you downloaded and installed Anaconda. If you want to use a different version of Python, for example Python 3.5, simply create a new environment and specify the version of Python that you want in that environment.
- Create a new environment named `py-crash-35` that contains Python 3.5:
 - In Navigator, click the **Environments** tab, then click the Create button.
 - The Create new environment dialog box appears.
 - In the Environment name field, type the descriptive name `py-crash-35` and select the version of Python you want to use from the Python Packages box (3.10, 3.9, 3.8, 3.7, 3.6, 3.5, or 2.7). Select a different version of Python than is in your other environments, base or snowflakes.
 - Click the Create button.
 - Activate the version of Python you want to use by clicking the name of that environment.



Managing Packages



- You can check which packages you have installed, check which are available, and look for a specific package and install it.
- To find a package you have already installed, click the name of the environment you want to search. The installed packages are displayed in the right pane.
- You can change the selection of packages displayed in the right pane at any time by clicking the drop-down box above it and selecting Installed, Not Installed, Updatable, Selected, or All.
- Check to see if a package you have not installed named “plotly” is available from the Anaconda repository (must be connected to the Internet). On the Environments tab, in the Search Packages box, type `plotly`, and from the Search Subset box select All or Not Installed.

Managing Packages

- To install the package into the current environment, check the checkbox next to the package name, then click the bottom Apply button.
- The newly installed program is displayed in your list of installed programs.

