

CS 482/682 MACHINE LEARNING

ANACONDA/SPYDER TUTORIAL

READ and follow each instruction and execute each one and **SIGN** at the end of the document stating everything worked as described.

Introduction

This tutorial will provide an introduction to Anaconda and Spyder. This is highly basic and you will have to explore and learn various features within Anaconda and related programs over the term.

What is Anaconda Navigator?

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® distribution that allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda.org or in a local Anaconda Repository. It is available for Windows, macOS, and Linux.

Downloading Anaconda

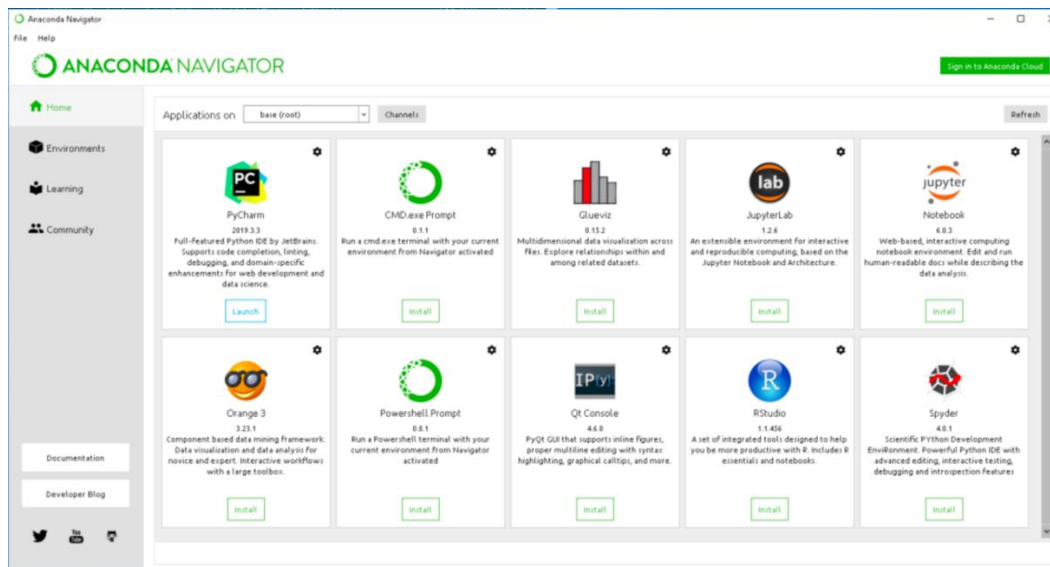
- Download Anaconda from this [link](#) (or search on Google: Anaconda download)
- Select the appropriate version for your operating system

Updating Anaconda

- If Anaconda is already installed, update Anaconda using the following command in the terminal:
`conda update anaconda`

Anaconda Navigator

- Open Anaconda Navigator



From here you can launch a variety of applications including Jupyter Notebooks and

What applications can I access using Navigator? (descriptions for each)

- [Jupyter Notebook](#): Popular data science IDE with Python and R. Code runs on a browser.
- [Spyder](#): Python IDE
- [Glueviz](#): interactive linked-view data visualization package for Python
- [Orange 3 App](#): data mining through visual programming or Python scripting
- [RStudio](#): R IDE
- Anaconda Prompt (Windows only)
- Anaconda PowerShell (Windows only)
- Advanced conda users can also [build their own Navigator applications](#).

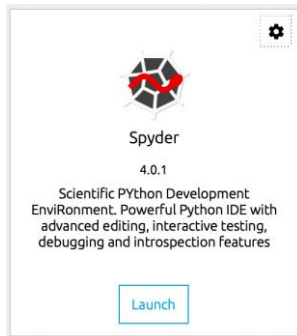
Further documentation and tutorials available from the [Anaconda website](#)

What is Spyder?

Spyder is a free and open source scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts. It features a unique combination of the advanced editing, analysis, debugging, and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package.

Launch Spyder

To launch Spyder through Anconda, select the launch icon from Anaconda Navigator:



Navigating Spyder

The screenshot shows the Spyder IDE interface. The main window is divided into three panes. The left pane is the "Editor", which contains a Python script named "numpy-tutorial.py". The right pane is the "Variable Explorer / Plots / Files" pane, which is currently showing the "Variable Explorer" tab. It displays a table of variables: "a" (Array of int64, (3, 5)) and "b" (Array of int64, (3,)). The bottom pane is the "IPython console", which shows the output of the script. Annotations in blue boxes point to each of these three panes.

Variable Explorer / Plots / Files

Editor

IPython console

Variable	Type	Size	Value
a	Array of int64	(3, 5)	[[0 1 2 3 4] [5 6 7 8 9] [10 11 12 13 14]]
b	Array of int64	(3,)	[6 7 8]

```
#!/usr/bin/env python3
# coding: utf-8
"""
Created on Tue Jan 26 17:38:00 2021
@author: timo
"""
import numpy as np
a = np.arange(15).reshape(3, 5)
print("inArray A:\n", a)
print("inArray A Size:\n", a.shape)
print("inArray A Dimensions:\n", a.ndim)
print("inArray A Data Type:\n", a.dtype.name)
print("inArray A Item Size:\n", a.itemsize)
print("inArray A Size:\n", a.size)
print("inArray A Type:\n", type(a))
b = np.array([6, 7, 8])
print("inArray B:\n", b)
print("inArray B Type:\n", type(b))
```

IPython 7.10.0 -- An enhanced Interactive Python.

```
In [1]: runfile('/Users/timo/Documents/KU/ML 101/numpy-tutorial.py', wdir='/Users/timo/Documents/KU/ML 101')
```

Array A:

```
[[ 0 1 2 3 4]
 [ 5 6 7 8 9]
 [10 11 12 13 14]]
```

Array A Size:

```
(3, 5)
```

Array A Dimensions:

```
2
```

Array A Data Type:

```
int64
```

Array A Item Size:

```
8
```

Array A Size:

```
15
```

Kite ready conda: base (Python 3.7.6) Line 8, Col 1 UTF-8 LF RW Mem 64%

The **Variable Explorer** in the Variable Explorer/Plots/Files allows you to interactively browse and manage the objects generated running your code.

Name	Type	Size	Value
data	Array of str128	(3, 3)	ndarray object of numpy module
df	DataFrame	(2, 2)	Column names: Col1, Col2
filename	str	1	/Users/Documents/spyder/sp...
x	float64	1	1.1235123099439
y	Array of float32	(3,)	[1. 2. 4.]

Variable explorer Help Plots Files

Console 1/A

```
Python 3.7.0 (default, Jun 28 2018, 07:39:16)
Type "copyright", "credits" or "license" for more information.

IPython 7.13.0 -- An enhanced Interactive Python.

In [1]: |
```

IPython console History

The **Plots** pane in the Variable Explorer/Plots/Files pane shows the static figures and images created during your session. It will show you plots from the **IPython Console**, produced by your code in the **Editor** or generated by the **Variable Explorer** allowing you to interact with them in several ways.

The top pane shows a code editor with the following Python code:

```
1 # Plot a terrain model and a polar plot side by side.
2
3 # Third party imports
4 import numpy as np
5 import matplotlib.pyplot as plt
6 import matplotlib.cm
7 import matplotlib.colors
8 import mpl_toolkits.mplot3d # Needed for 3D plots: Disable unused-l
9
10 # %% Plot final terrain model
11
12 # pylint: disable=member
13
14
15
16
17
18 def generate_polar_plot():
19     """Generate an example polar slice plot."""
20     # Compute pie slices
21     n_slices = 20
22     theta = np.linspace(0, 2 * np.pi, n_slices, endpoint=False)
23     radii = 10 * np.random.rand(n_slices)
24     width = np.pi / 4 * np.random.rand(n_slices)
25
26     plt.subplots(figsize=(15, 6))
27     ax1 = plt.subplot(1, 2, 1, projection='polar')
28     bars = ax1.bar(theta, radii, width=width, bottom=0.0)
29
30     # Use custom colors and opacity
31     for radius, plot_bar in zip(radii, bars):
32         plot_bar.set_facecolor(plt.cm.viridis(radius / 10.))
33         plot_bar.set_alpha(0.5)
34
35
36 def generate_dem_plot():
37     """Generate a 3D representation of a terrain DEM"""
38     dem_path = '/Users/juanis/Documents/Local/SpyderDocs/dem.npy'
39     with np.load(dem_path) as dem:
40         z_data = dem['elevation']
41         rows, cols = z_data.shape
42         x_data = np.linspace(dem['xmin'], dem['xmax'], ncols)
43         y_data = np.linspace(dem['ymin'], dem['ymax'], nrows)
44         x_data, y_data = np.meshgrid(x_data, y_data)
45
46         region = np.s_[5:50, 5:50]
47         x_region, y_region, z_region = (
48             x_data[region], y_data[region], z_data[region])
49
50     axes = plt.subplot(1, 2, 2, projection='3d')
```

The bottom pane shows the IPython console with the following commands:

```
In [3]: runfile('/Users/juanis/Documents/Local/SpyderDocs/image.py', wdir='/Users/juanis/Documents/Local/SpyderDocs')
/Users/juanis/Documents/Local/SpyderDocs/image.py:13: MatplotlibDeprecationWarning:
The examples_directory rcparam was deprecated in Matplotlib 3.0 and will be removed in 3.2. In the future
examples will be found relative to the 'datapath' directory.
  matplotlib.rcParams['examples.directory'] = './'
/Users/juanis/Documents/Local/SpyderDocs/image.py:17: MatplotlibDeprecationWarning:
The examples_directory rcparam was deprecated in Matplotlib 3.0 and will be removed in 3.2. In the future
examples will be found relative to the 'datapath' directory.
  with get_sample_data('ci.raw.gz') as datafile:

In [4]: runfile('/Users/juanis/Documents/Local/SpyderDocs/plot_example5.py', wdir='/Users/juanis/Documents/Local/SpyderDocs')

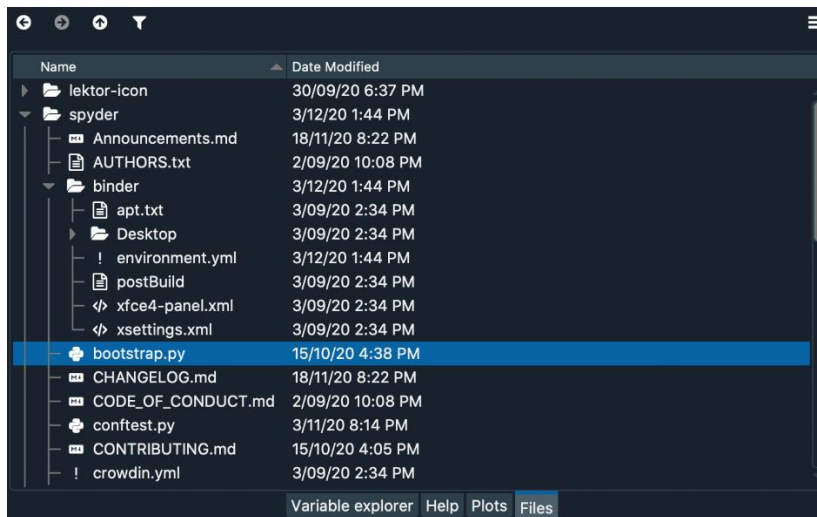
In [5]: %matplotlib inline

In [6]: runfile('/Users/juanis/Documents/Local/SpyderDocs/plot_example.py', wdir='/Users/juanis/Documents/Local/SpyderDocs')

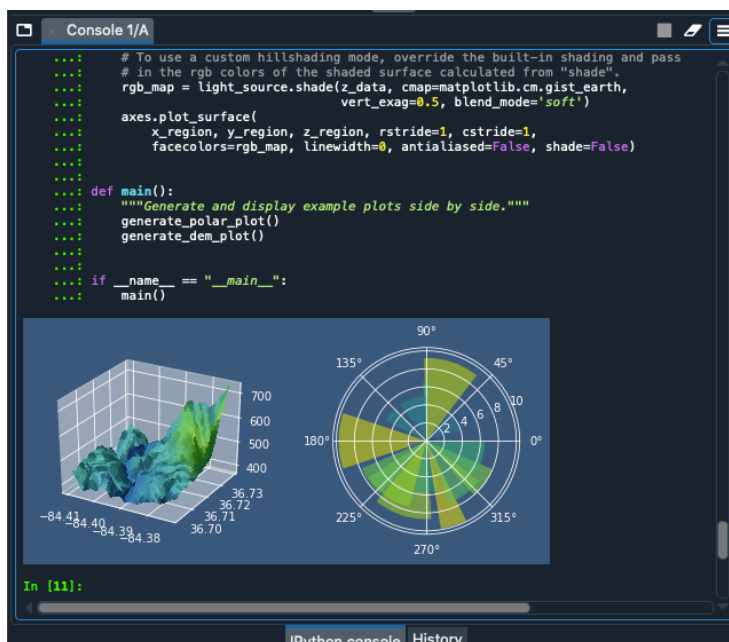
In [7]:
```

The right pane shows the Plots pane with two subplots: a polar plot and a 3D surface plot. The status bar at the bottom indicates: LSP Python ready, conda base, Python 3.7.4, Line 1, Col 1, ASCII, LF, RW, Mem 55%.

The **Files** pane in the Variable Explorer/Plots/Files is a filesystem and directory browser built right into Spyder. You can view and filter files according to their type and extension, open them with the **Editor** or an external tool, and perform many common operations.

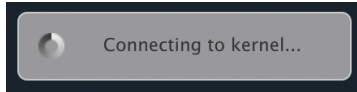


The **IPython Console** allows you to execute commands and enter, interact with and visualize data inside any number of fully featured **IPython** interpreters. Each console is executed in a separate process, allowing you to run scripts, interrupt execution and restart or terminate a shell without affecting the others or Spyder itself, and easily test your code in a clean environment without disrupting your primary session.



If you see the following error:

Stuck on...



Then...update Anaconda using the following command in the terminal:

```
conda update anaconda
```

Tips for Spyder

1. Ctrl + 1 - Comment/uncomment.
2. Ctrl + 4 - Comment a block of code.
3. Ctrl + 5 - Uncomment a block of code.

SIGN: I acknowledge that I have read the document above and experimented with each feature and they all work as described.

Dylan Lozon

Maddy Connell

Your Name