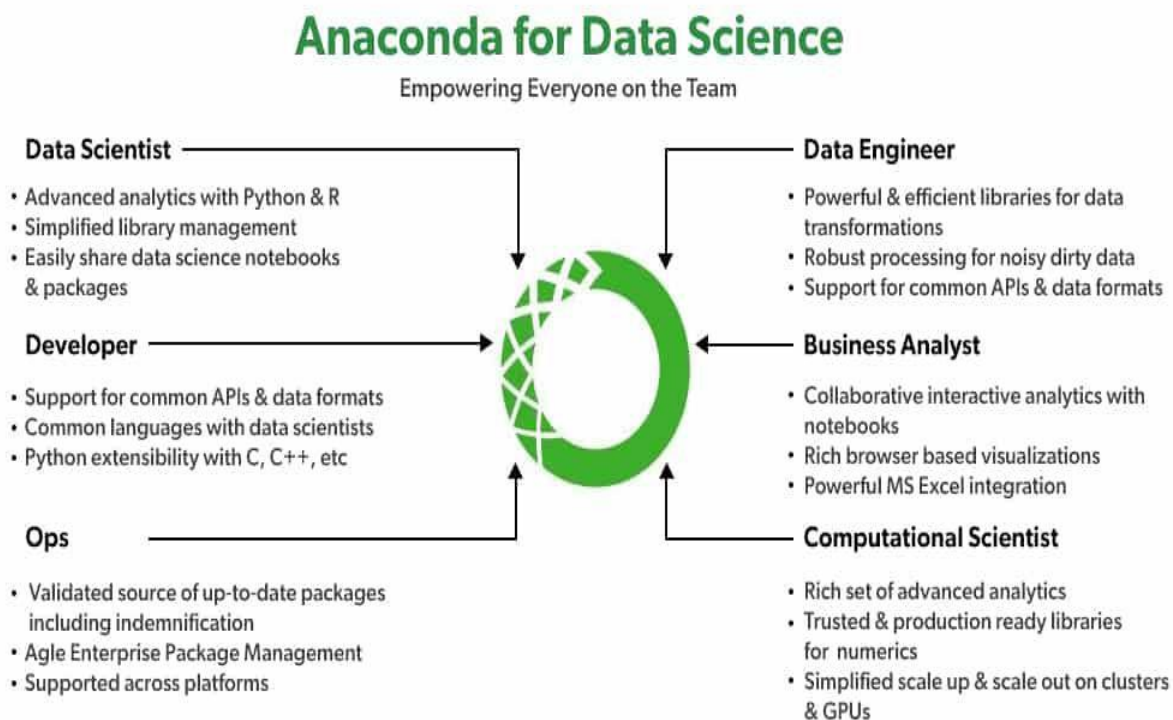


How to Setup Anaconda For Data Science?

Why Anaconda for Data Science?

Here are some important reasons why you should choose Anaconda for your next Data Science project:

- Ease of **Installation**,
- More than **1000** data science packages are available,
- Anaconda installs the latest Python 2 or 3 version in an isolated and activated environment so any installed Python version doesn't cause any issues for your projects,
- It's **noob-friendly!!** Yes, You don't need any prior coding/programming knowledge about the usual nerdy stuff that scares noobs away.



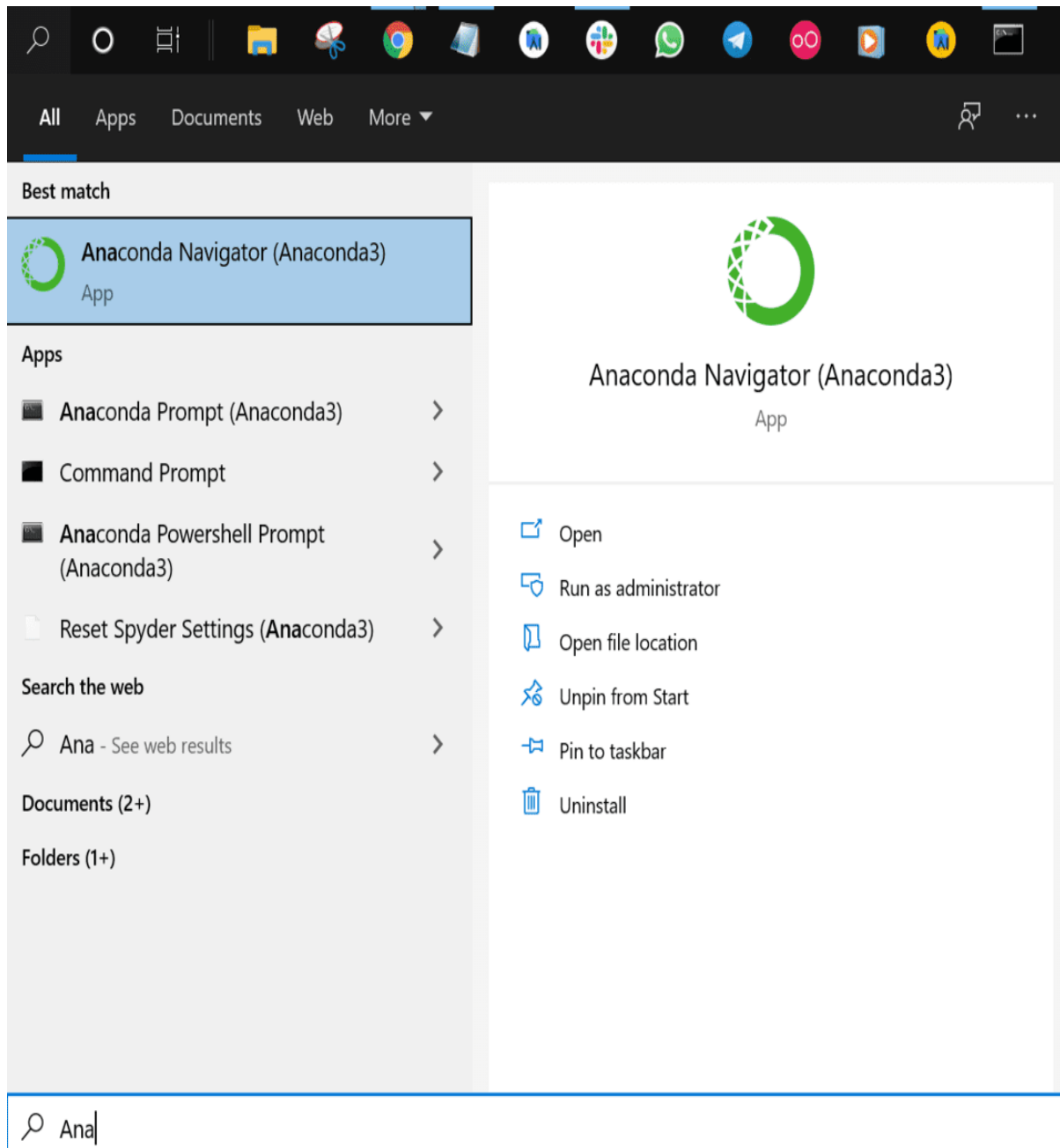
Installation Process

- To install Anaconda on Windows please refer to [How to install Anaconda on Windows?](#)
- To install Anaconda on Linux please refer to [How to install Anaconda on Linux?](#)

After successfully completing the installation process you are here now, the **Anaconda Navigator**.

Anaconda Navigator

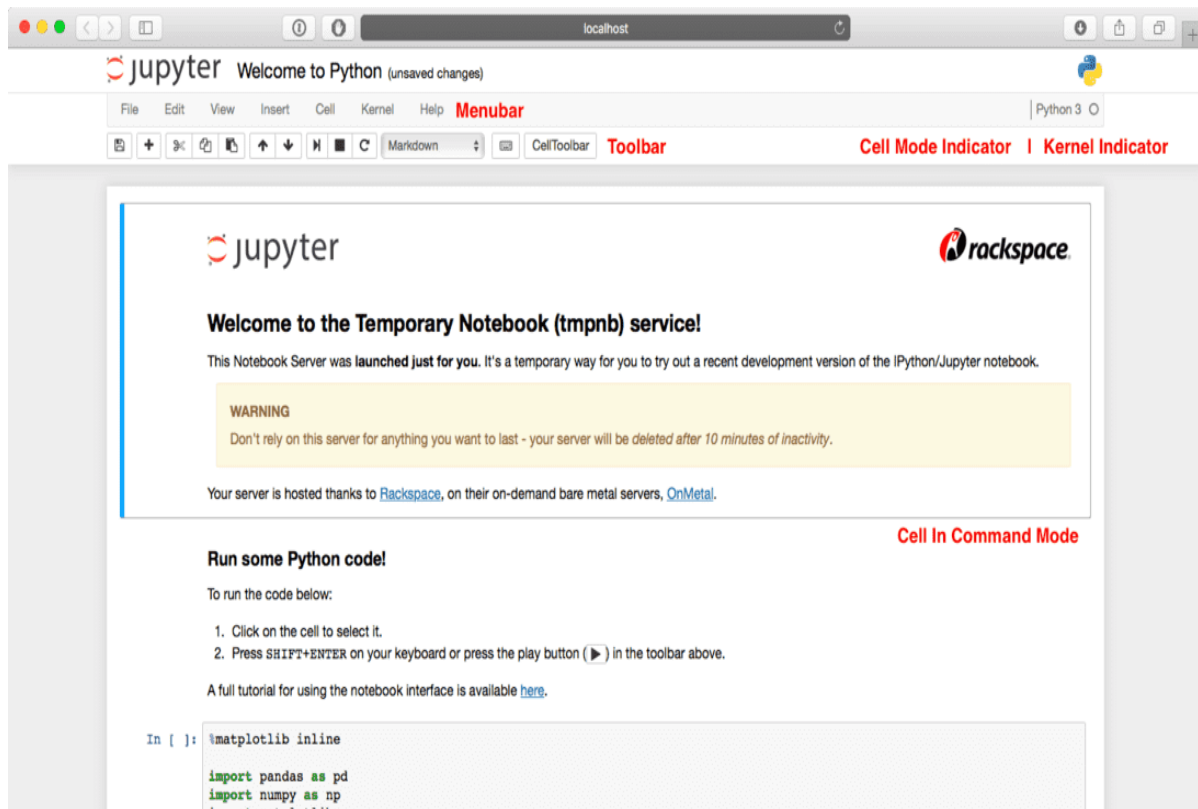
Anaconda Navigator is a graphical UI that is automatically installed with Anaconda. Navigator will open if the installation was successful. For Windows users, click Start, search, or select Anaconda Navigator from the menu as shown in the below image:



Anaconda Navigator contains lots of stuff inside it. So let's understand which stuff we need for our next data science project.

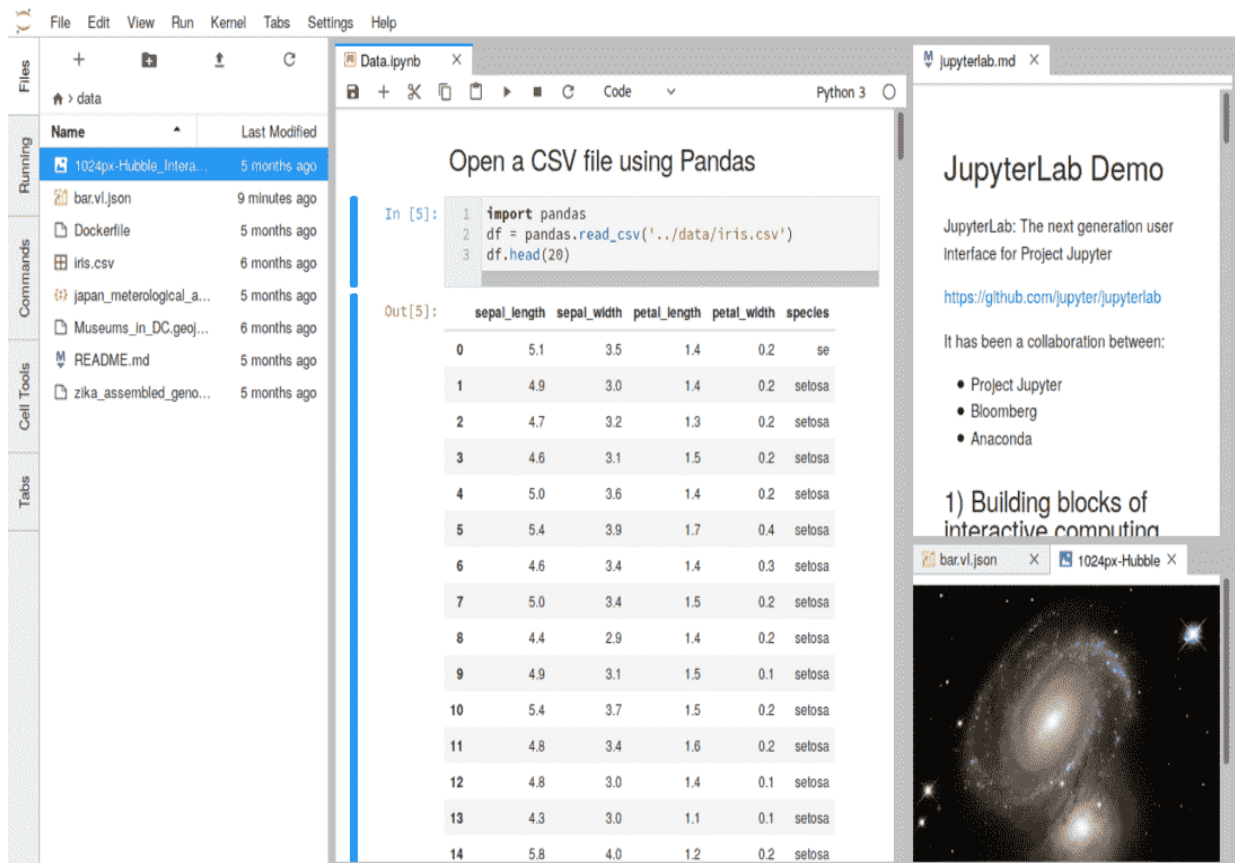
1. Jupyter Notebook

Jupyter Notebook is a web-based, interactive computing notebook environment. You can edit and run human-readable docs while describing the data analysis. The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. Uses include data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more. Below is a demo image to demonstrate how Jupyter Notebook UI looks like:



2. JupyterLab

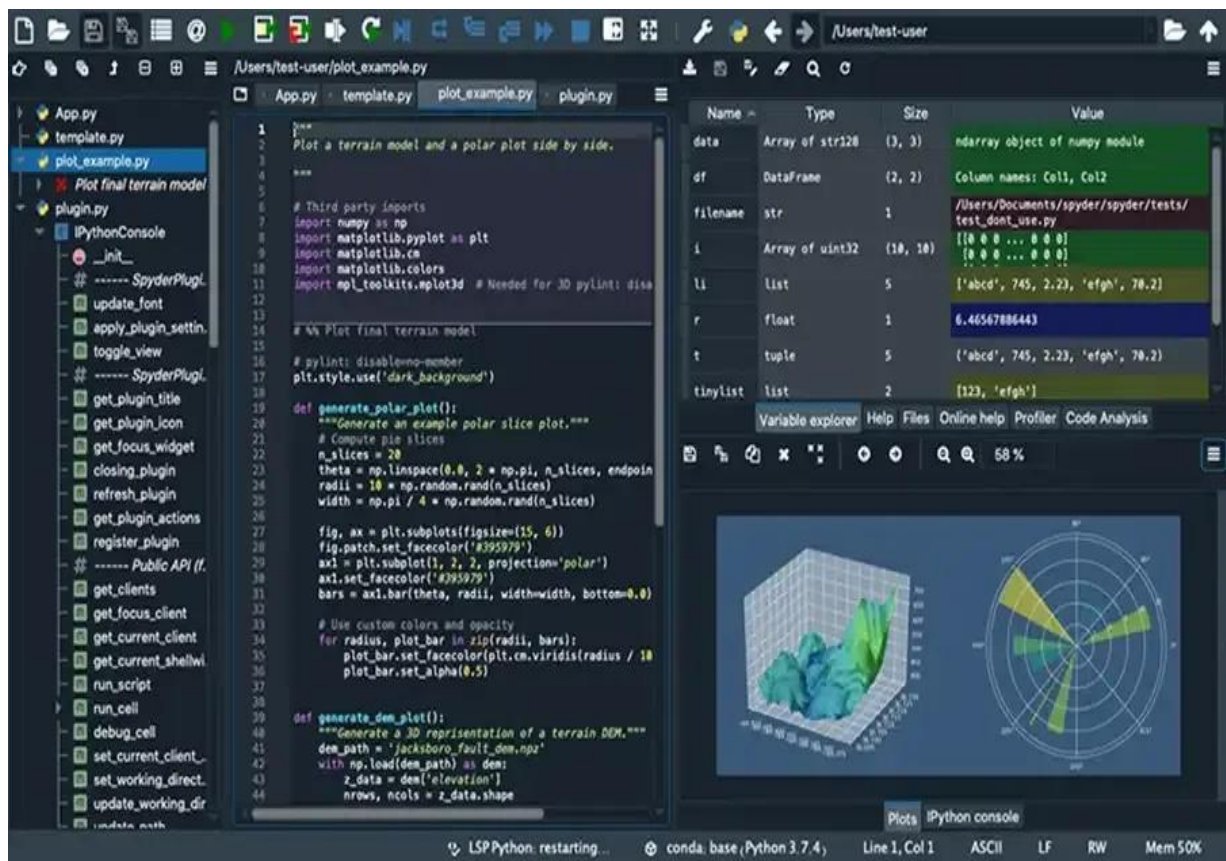
It's an extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture. JupyterLab enables to work with documents and activities such as Jupyter notebooks, text editors, terminals, and custom components in a flexible, integrated, and extensible manner. Below is a demo image to demonstrate how JupyterLab UI looks like:



Jupyterlab in anaconda

3. Spyder

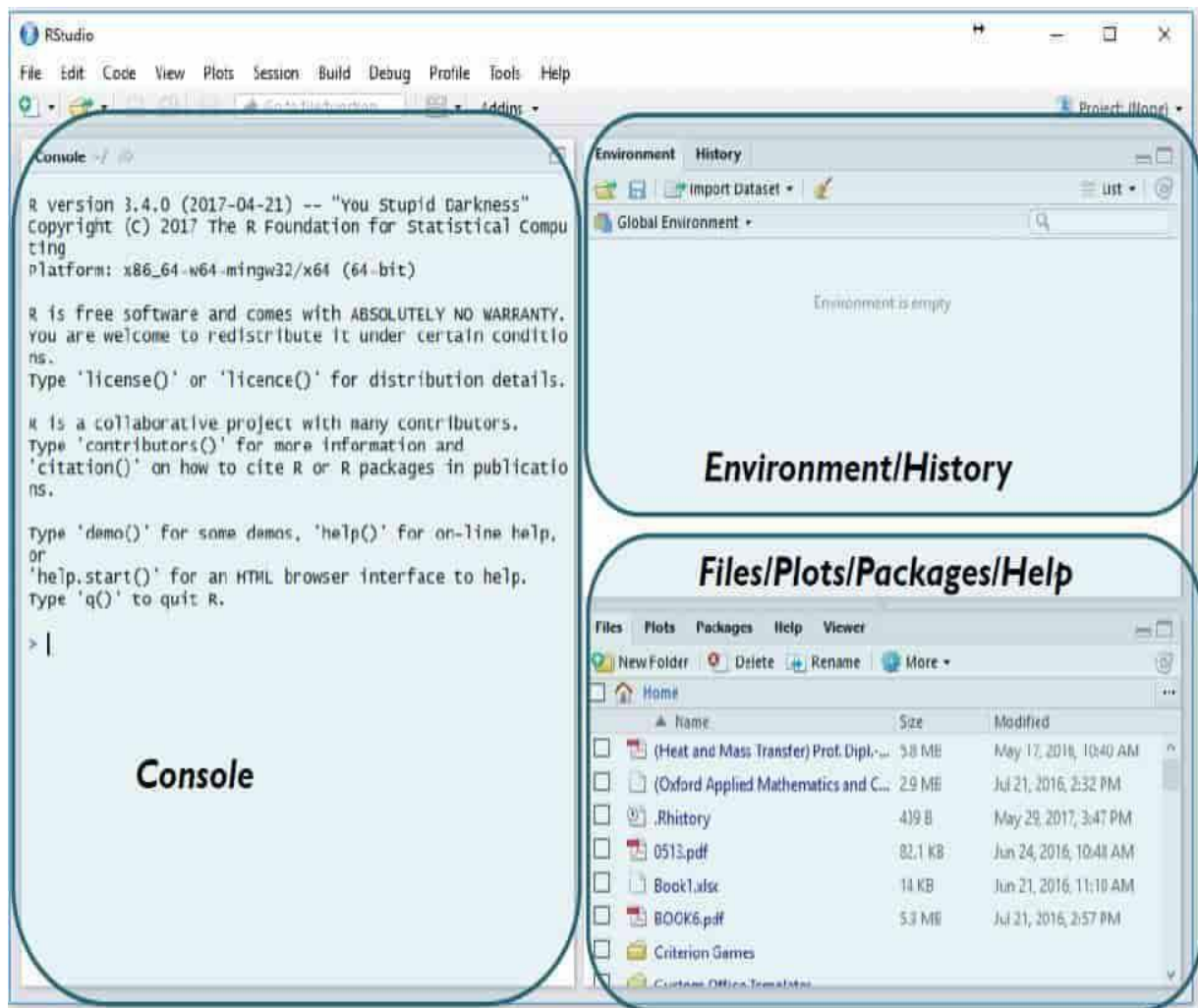
One of the most important and powerful Python IDE is Spyder. Spyder is another good open-source and cross-platform IDE written in Python. It is also called Scientific Python Development IDE, and it is the most lightweight IDE for Python. It is mainly used by data scientists who can integrate with Matplotlib, SciPy, NumPy, Pandas, Cython, IPython, SymPy, and other open-source software. Below is a demo image to demonstrate how Spyder UI looks like:



Spyder in Anaconda

4. RStudio

When it comes to the data science world then Python and R are the two most programming languages that come into our minds. R Studio is an integrated development environment(IDE) for the [R programming language](#). It provides literate programming tools, which basically allow the use of R scripts, outputs, text, and images in reports, Word documents, and even an HTML files. Below is a demo image to demonstrate how RStudio UI looks like:



R Studio in Anaconda lab

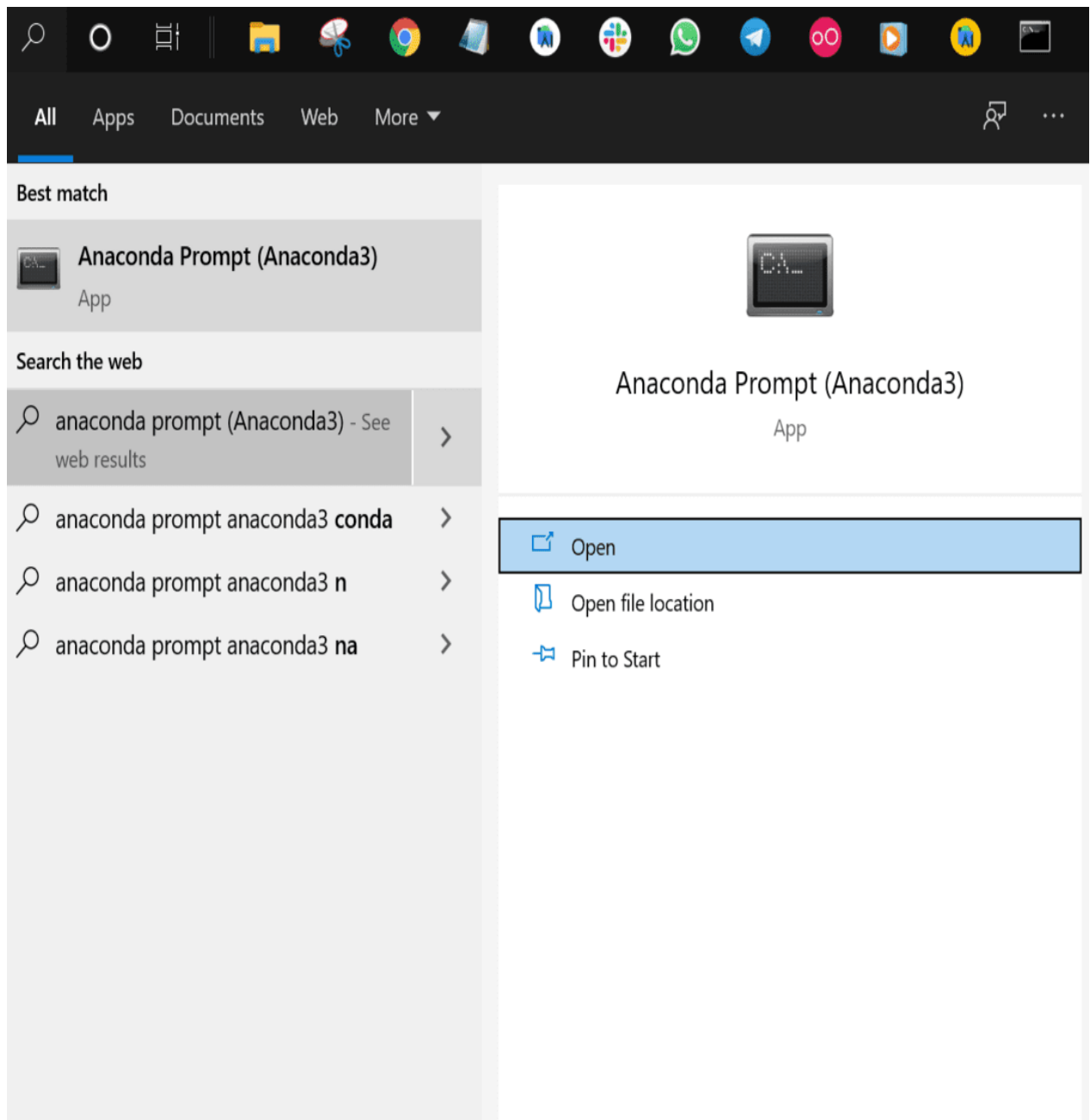
For more detail please refer to: [Introduction to R Studio](#)

Apart from these 4 important environments there also some other environments which are very useful for the data science project. Let's discuss them in brief.

1. **Datalore:** Datalore is an online data analysis tool with smart coding assistance by Jetbrains. One can edit and run the Python notebooks in the cloud and share them with the team.
2. **Glueviz:** Glueviz is a multidimensional data visualization across files. It is a Python library to explore relationships within and among related datasets.
3. **Orange 3:** Orange 3 is a component-based data mining framework. It is a powerful platform to perform data analysis and visualization, see data flow, and become more productive.
4. **IBM Watson Studio Cloud:** IBM Watson Studio Cloud provides the tools to analyze and visualize data, cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open-source data science tools or visualize modeling.

Conda

If you prefer using a command-line interface (CLI), you can use **conda** to verify the installation using Anaconda Prompt on Windows or a terminal on Linux and macOS. To open Anaconda Prompt in Windows: Click Start, search, or select Anaconda Prompt from the menu.



anaconda prompt (Anaconda3)

Anaconda prompt in window

Once you launch the prompt you will notice that the terminal now has (base) written in front of the computer name. It means that your base conda environment is set (meaning you're working globally for the whole user and not a specific environment).

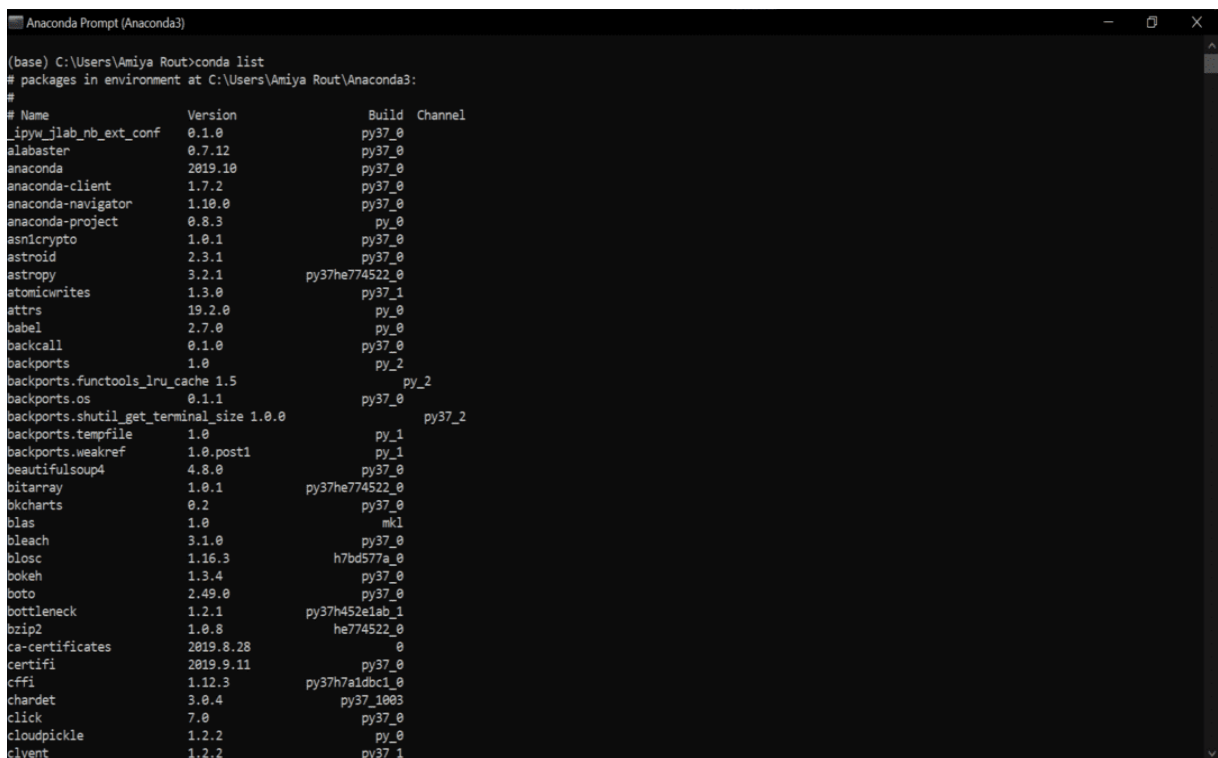


Now let's discuss some useful commands

1. View installed packages

To view all the installed packages please type the command as follows:

```
conda list
```

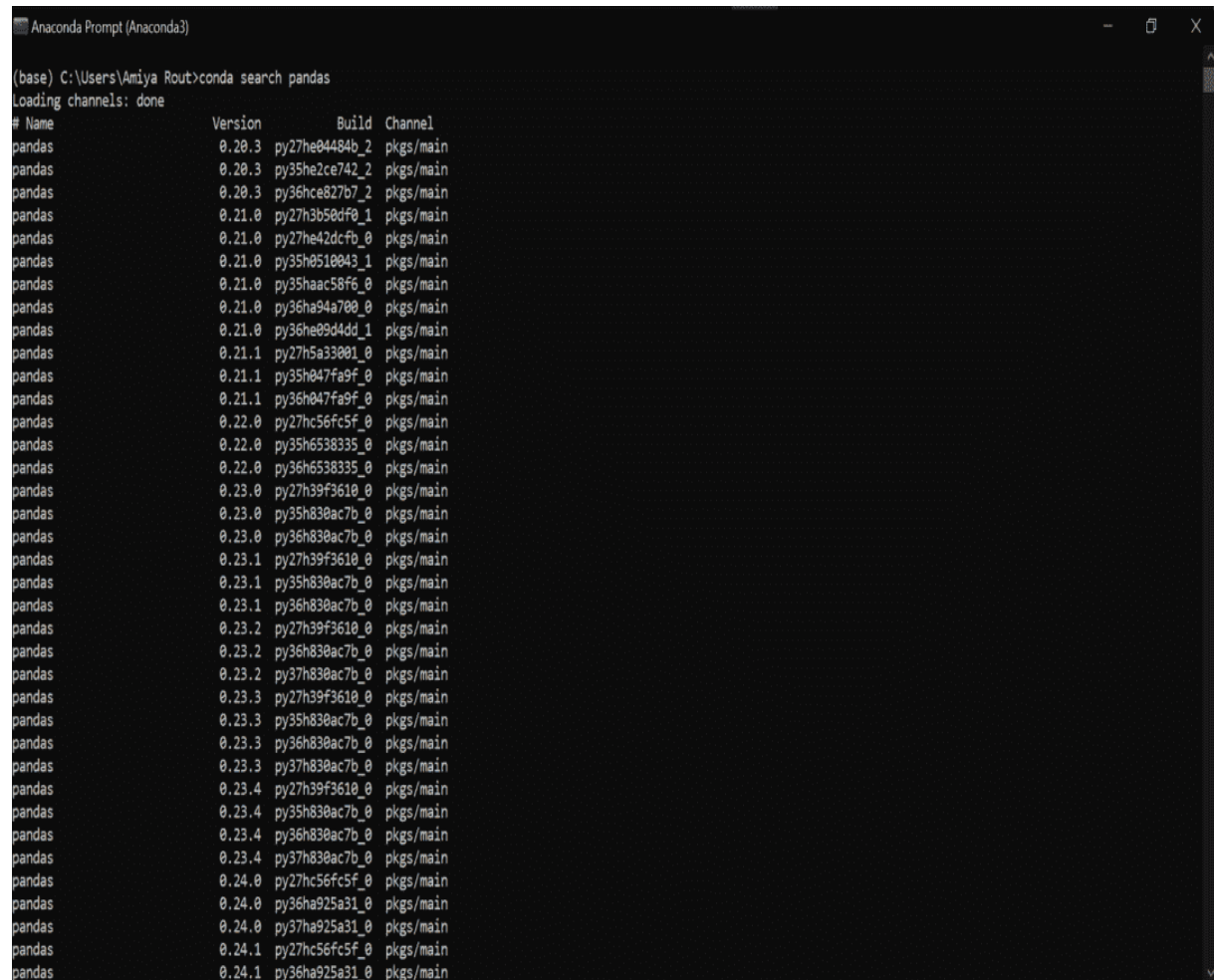


list of files in anaconda

2. Search and install a package

Let's say the user wants to install **pandas** but he/she does not know the version. The user can use the following command to search for its versions:

```
conda search pandas
```



The screenshot shows an Anaconda Prompt window with the command `(base) C:\Users\Amiya Rout>conda search pandas` entered. The output displays a list of pandas packages available across different channels, including their names, versions, build numbers, and the channel they belong to. The packages are listed in a table format with columns: #, Name, Version, Build, and Channel. The packages are sorted by version, showing various builds for versions 0.20.3, 0.21.0, 0.21.1, 0.22.0, 0.23.0, 0.23.1, 0.23.2, 0.23.3, 0.23.4, 0.24.0, 0.24.1, and 0.24.1.

#	Name	Version	Build	Channel
1	pandas	0.20.3	py27he04484b_2	pkgs/main
2	pandas	0.20.3	py35he2ce742_2	pkgs/main
3	pandas	0.20.3	py36hce827b7_2	pkgs/main
4	pandas	0.21.0	py27h3b50df0_1	pkgs/main
5	pandas	0.21.0	py27he42dcfb_0	pkgs/main
6	pandas	0.21.0	py35h0510043_1	pkgs/main
7	pandas	0.21.0	py35haac58f6_0	pkgs/main
8	pandas	0.21.0	py36ha94a700_0	pkgs/main
9	pandas	0.21.0	py36he09d4dd_1	pkgs/main
10	pandas	0.21.1	py27h5a33001_0	pkgs/main
11	pandas	0.21.1	py35h047fa9f_0	pkgs/main
12	pandas	0.21.1	py36h047fa9f_0	pkgs/main
13	pandas	0.22.0	py27hc56fc5f_0	pkgs/main
14	pandas	0.22.0	py35h6538335_0	pkgs/main
15	pandas	0.22.0	py36h6538335_0	pkgs/main
16	pandas	0.23.0	py27h39f3610_0	pkgs/main
17	pandas	0.23.0	py35h830ac7b_0	pkgs/main
18	pandas	0.23.0	py36h830ac7b_0	pkgs/main
19	pandas	0.23.1	py27h39f3610_0	pkgs/main
20	pandas	0.23.1	py35h830ac7b_0	pkgs/main
21	pandas	0.23.1	py36h830ac7b_0	pkgs/main
22	pandas	0.23.2	py27h39f3610_0	pkgs/main
23	pandas	0.23.2	py36h830ac7b_0	pkgs/main
24	pandas	0.23.2	py37h830ac7b_0	pkgs/main
25	pandas	0.23.3	py27h39f3610_0	pkgs/main
26	pandas	0.23.3	py35h830ac7b_0	pkgs/main
27	pandas	0.23.3	py36h830ac7b_0	pkgs/main
28	pandas	0.23.3	py37h830ac7b_0	pkgs/main
29	pandas	0.23.4	py27h39f3610_0	pkgs/main
30	pandas	0.23.4	py35h830ac7b_0	pkgs/main
31	pandas	0.23.4	py36h830ac7b_0	pkgs/main
32	pandas	0.23.4	py37h830ac7b_0	pkgs/main
33	pandas	0.24.0	py27hc56fc5f_0	pkgs/main
34	pandas	0.24.0	py36ha925a31_0	pkgs/main
35	pandas	0.24.0	py37ha925a31_0	pkgs/main
36	pandas	0.24.1	py27hc56fc5f_0	pkgs/main
37	pandas	0.24.1	py36ha925a31_0	pkgs/main

pandas in anaconda

To install a package type the following command

```
conda install pandas
```

```
Anaconda Prompt (Anaconda3)

(base) C:\Users\Amiya Rout>conda install pandas
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.

(base) C:\Users\Amiya Rout>
```

And the user wants to install **pandas with version 1.2.4** then use the following command to do so:

```
conda install pandas==1.2.4
```

3. Remove package

To remove the package please type the command as follows:

```
conda remove pandas
```

```
Anaconda Prompt (Anaconda3) - conda remove pandas

(base) C:\Users\Amiya Rout>conda remove pandas
Collecting package metadata (repodata.json): done
Solving environment: /
Warning: 2 possible package resolutions (only showing differing packages):
- defaults/win-64::libtiff-4.2.0-hd0e1b90_0, defaults/win-64::zstd-1.4.5-h04227a9_0
- defaults/win-64::libtiff-4.1.0-h56a325e_1, defaults/win-64::zstd-1.4.9-h19a0ad4done

## Package Plan ##

environment location: C:\Users\Amiya Rout\Anaconda3

removed specs:
- pandas

The following packages will be downloaded:
```

package	build	size
anyio-2.2.0	py37haa95532_1	125 KB
argon2-cffi-20.1.0	py37h2bbff1b_1	49 KB
async_generator-1.10	py37h28b3542_0	40 KB
attrs-20.3.0	pyhd3eb1b0_0	43 KB
babel-2.9.0	pyhd3eb1b0_0	5.5 MB
backcall-0.2.0	pyhd3eb1b0_0	13 KB
backports-1.0	pyhd3eb1b0_2	210 KB
backports.functools_lru_cache-1.6.4	pyhd3eb1b0_0	9 KB
backports.tempfile-1.0	pyhd3eb1b0_1	11 KB
beautifulsoup4-4.9.3	pyha847dfd_0	86 KB
bleach-3.3.0	pyhd3eb1b0_0	113 KB
broutlipy-0.7.0	py37h2bbff1b_1003	337 KB
ca-certificates-2021.4.13	haa95532_1	115 KB
certifi-2020.12.5	py37haa95532_0	141 KB
cffi-1.14.5	py37hcd4344a_0	220 KB
chardet-4.0.0	py37haa95532_1003	209 KB
click-7.1.2	pyhd3eb1b0_0	64 KB
colorama-0.4.4	pyhd3eb1b0_0	21 KB
conda-package-handling-1.7.3	py37h8cc25b3_1	721 KB
cryptography-3.4.7	py37h71e12ea_0	641 KB

Conda Remoe in Anaconda

