

# Installing R Jupyter Notebook via Anaconda Navigator

By Abzer Kelminal ([abzer.shah@uni-tuebingen.de](mailto:abzer.shah@uni-tuebingen.de))

---

## Contents:

<b>Anaconda Installation:</b>	<b>2</b>
<b>Navigating through the homepage:</b>	<b>5</b>
<b>Installing R kernel via Anaconda homepage:</b>	<b>6</b>
<b>Installing JupyterLab in an R-environment (Optional):</b>	<b>10</b>
<b>Troubleshooting:</b>	<b>14</b>
1) DLL Error in Anaconda Prompt:	14
2) Formatting the conda shell	14
3) Failed to add Anaconda to the system PATH:	15
4) R Kernel always busy in Jupyter Notebook:	16
5) Error during install.packages() in R Jupyter Notebook	18

# Anaconda Installation:

Anaconda is a desktop application that allows users to launch standard python programs without using the command line. <https://www.anaconda.com/>

## Step 1: Download the Anaconda installers

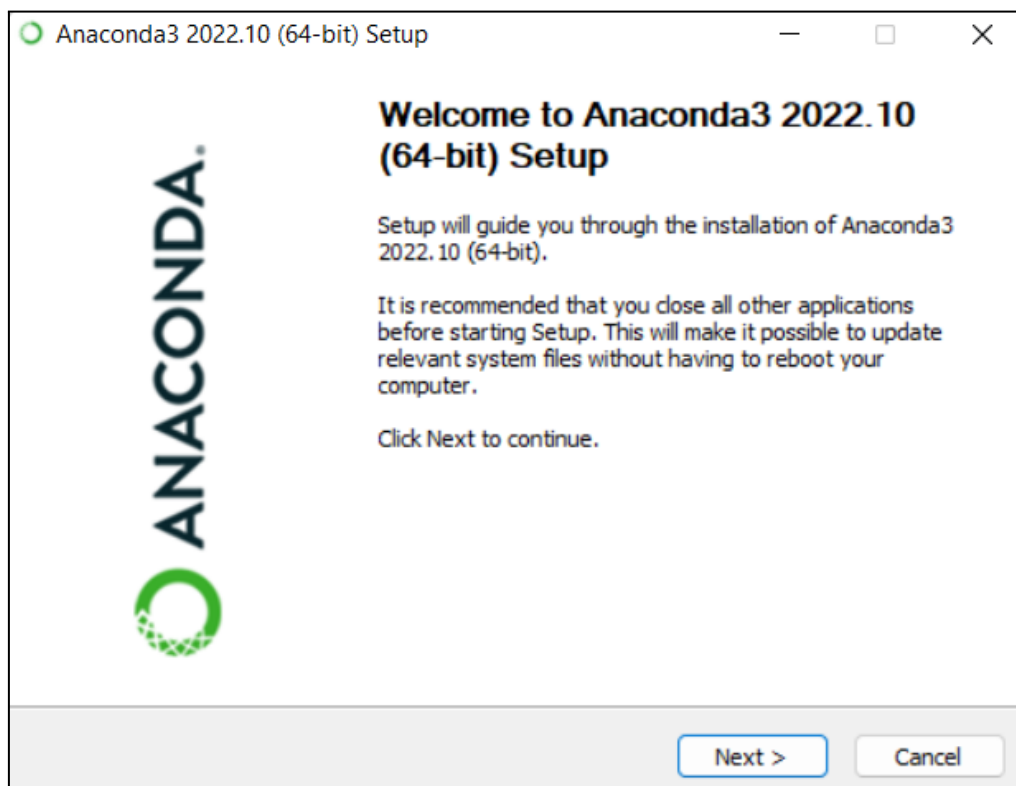
for different OS (Windows, macOS, Linux): [Anaconda](#) | [Anaconda Distribution](#)

## Step 2: Installation of Anaconda

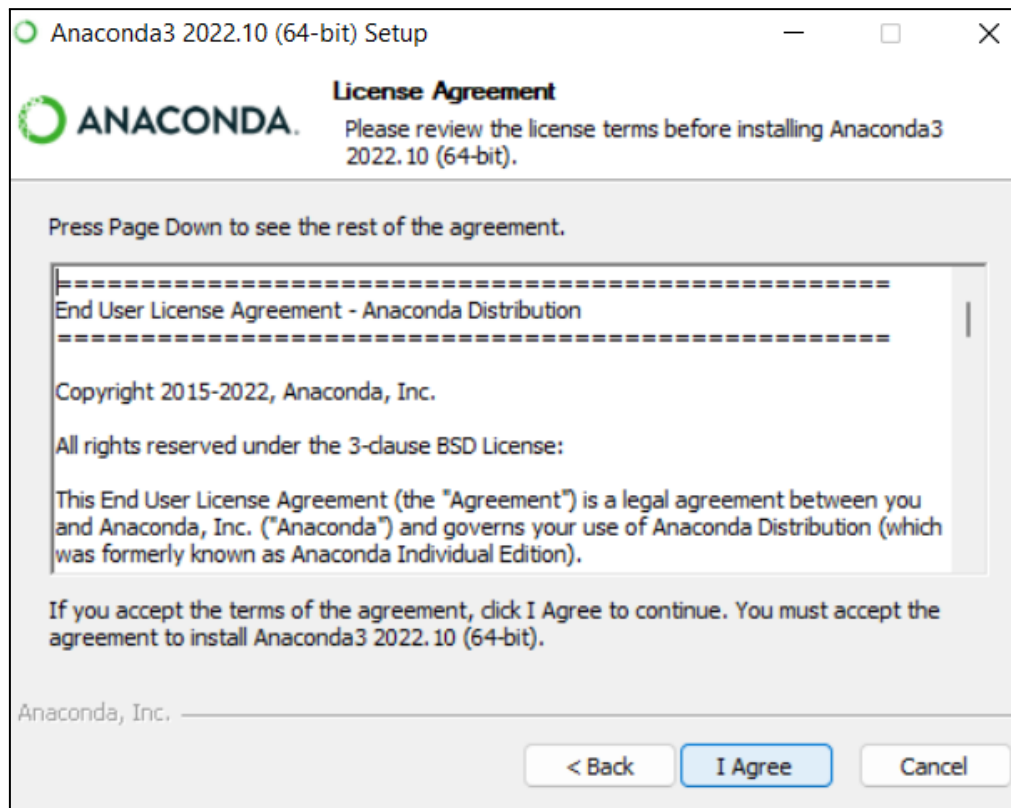
After downloading the respective **.exe file**, follow the steps to install the application: <https://docs.anaconda.com/anaconda/install/>

### Installation steps for Windows as follows:

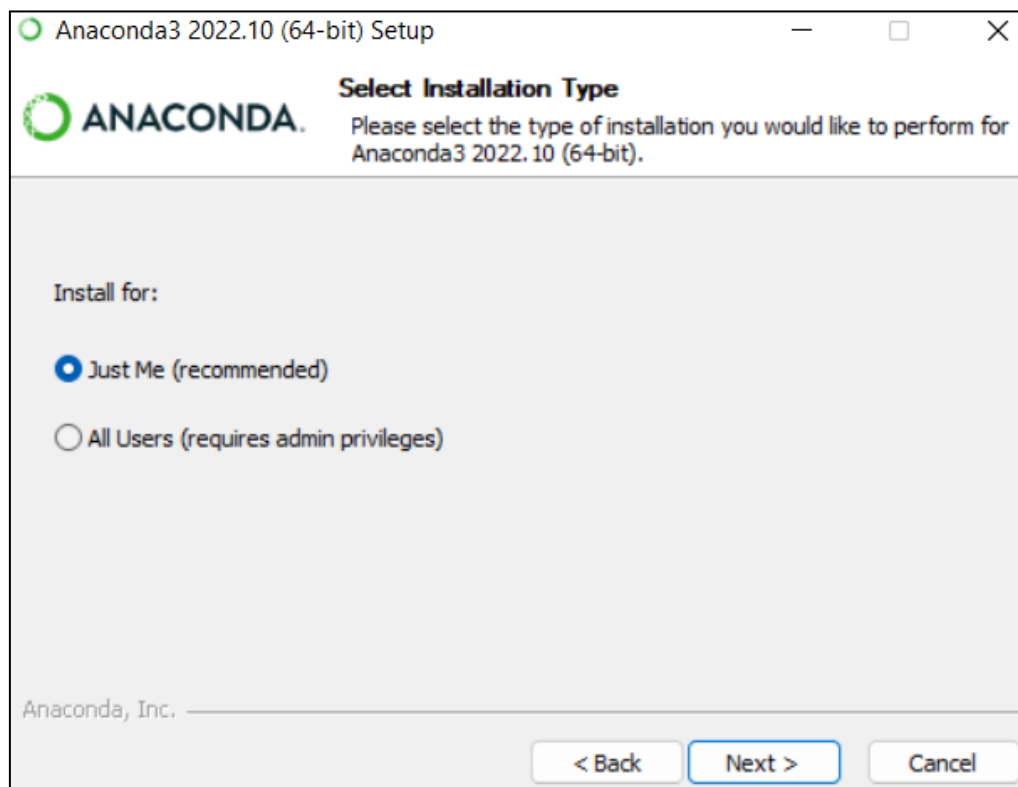
1. When you open the .exe file, the welcome tab will appear. **Click Next**



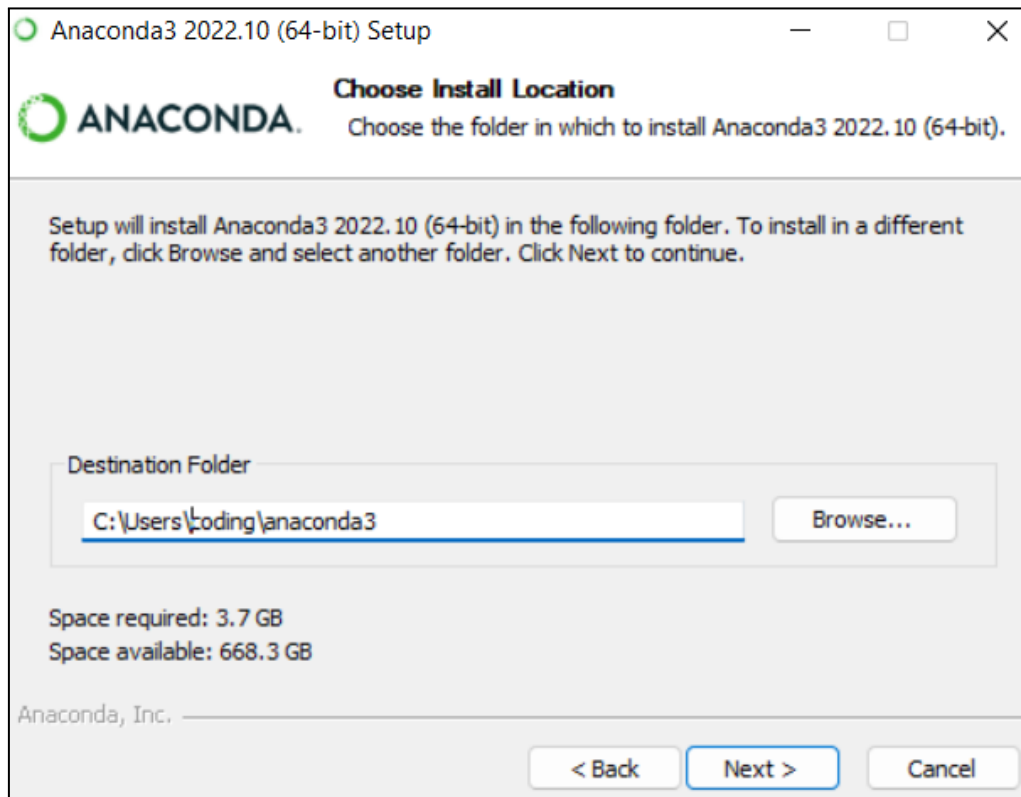
## 2. Click, I agree



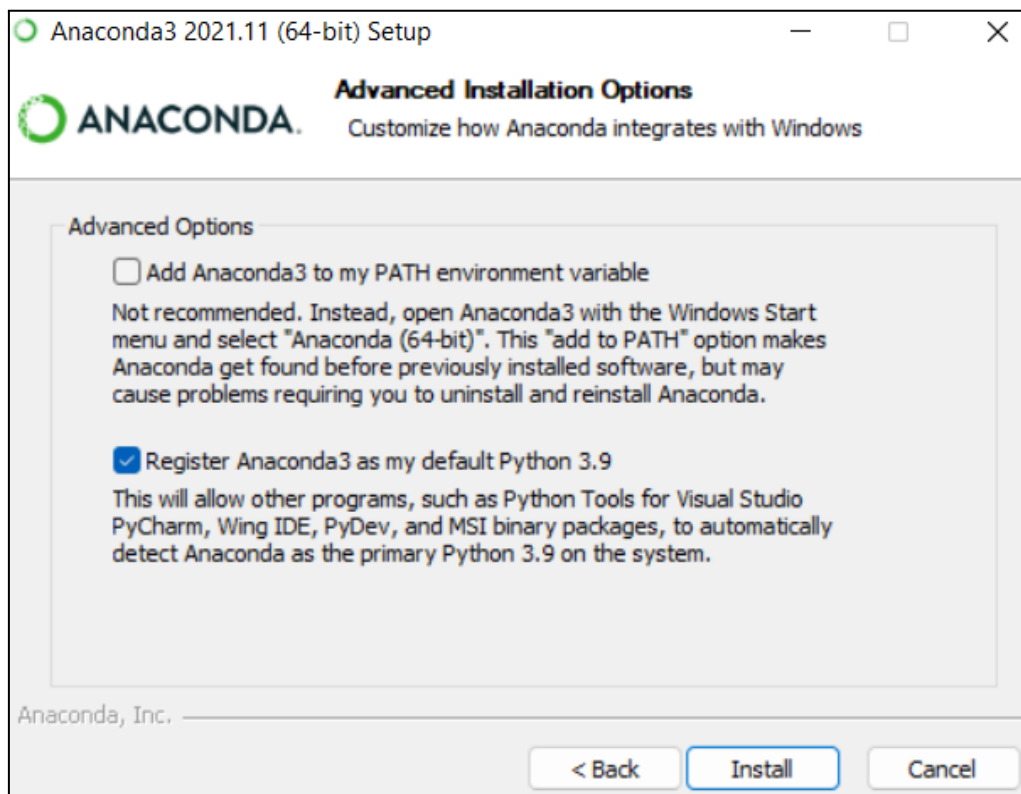
## 3. Install for 'Just Me' and click Next



4. Define a path for installing the folder: Usually, it is  
**C:\Users\username\anaconda2**

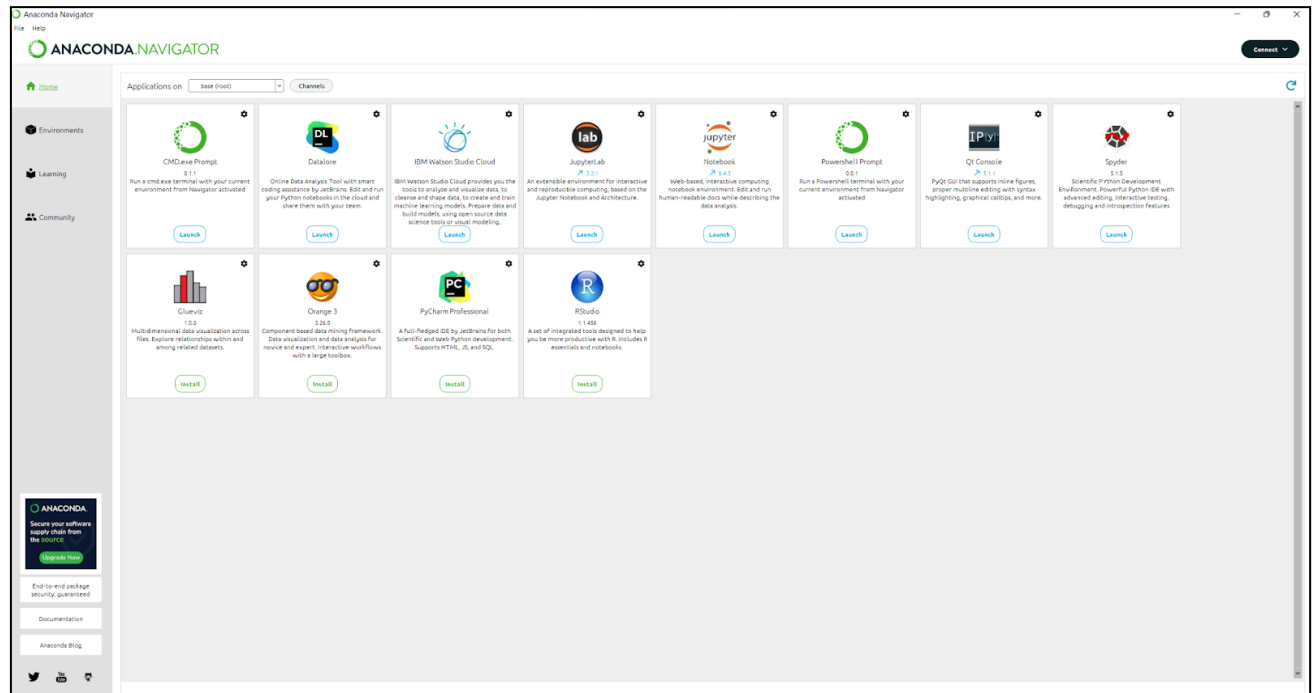


4. Register Anaconda3 as the default Python path and install.



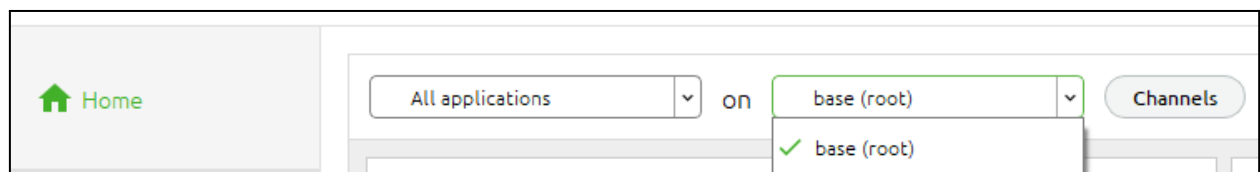
# Navigating through the homepage:

Once installed, the homepage of Anaconda will look like this:



**Critical:** When you get a notification for updating Anaconda, please update it to the latest version (2.3.2)

By default, the applications shown on the homepage such as JupyterLab, Jupyter Notebook homepage is on the **base(root)** environment. An environment is simply a directory or folder where all the packages are installed.



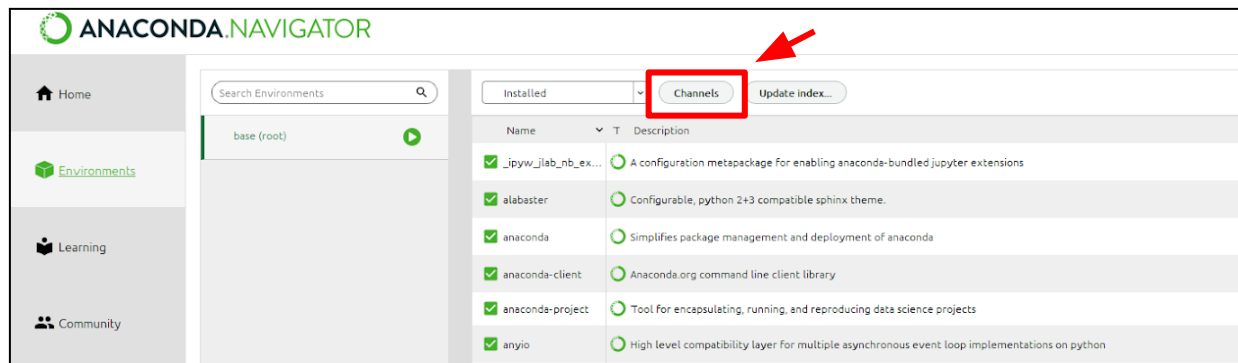
The base environment contains all the python packages as Anaconda, by default, works with Python kernels. When we open Jupyter Notebook or JupyterLab via the base environment, we can open/create notebooks with only Python kernels and not others.

In order to make the notebook recognize R codes, we need to install R Kernels onto our Anaconda environment. That way we can run/create Jupyter notebooks using R code.

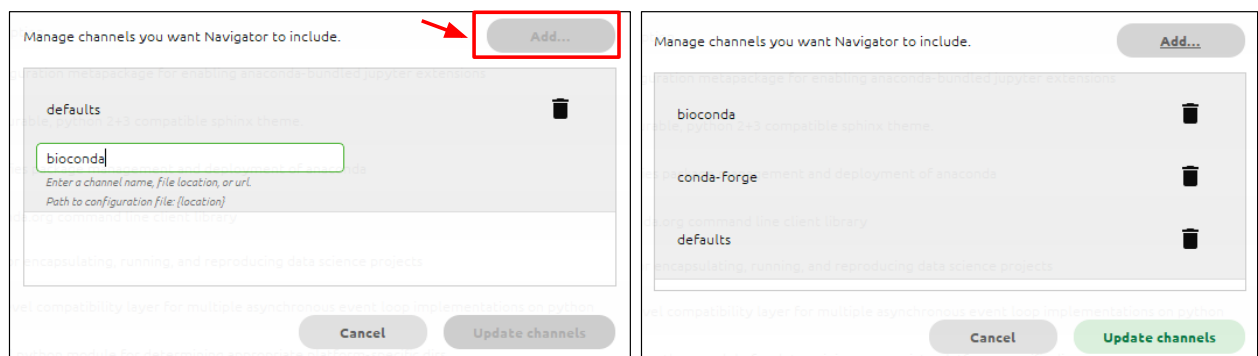
# Installing R kernel via Anaconda homepage:

We can create a new environment for R from the homepage.

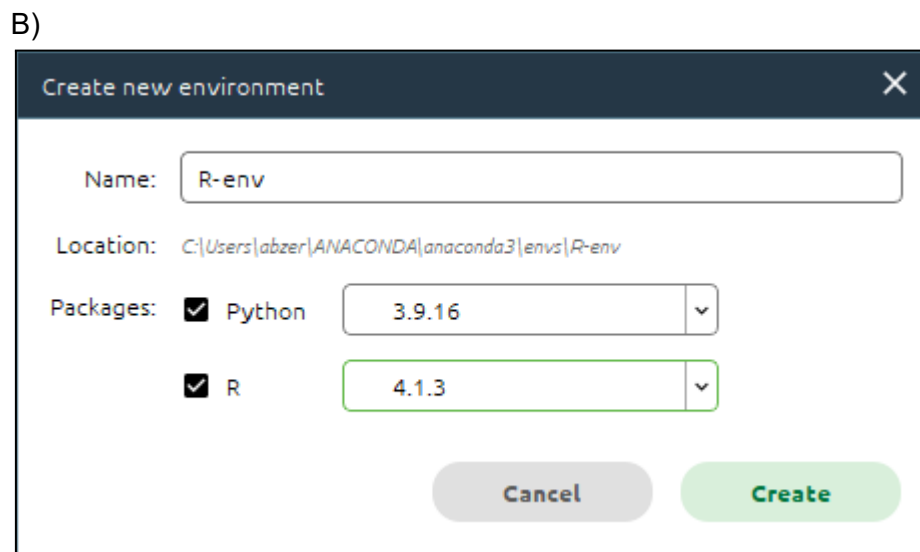
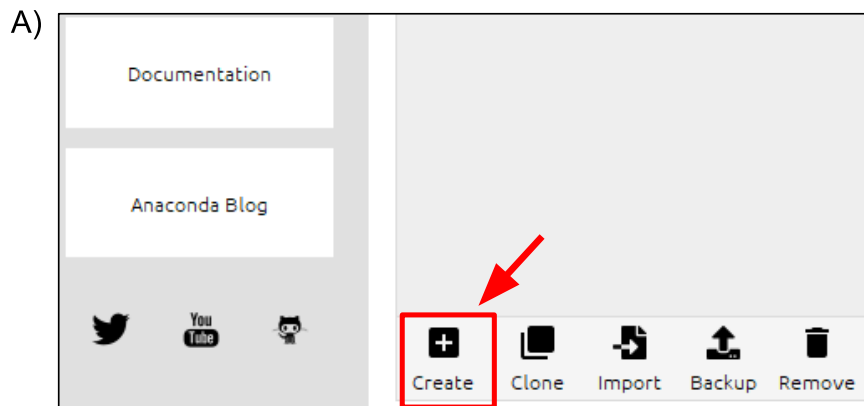
**Step 1:** Go to → Environments → channels. In addition to the default channel, we need to add the channels to download the R packages or dependencies. **conda-forge** contains the 'r-base' package (highest available R version 4.1.3) and **bioconda** contains many other bioinformatic packages.



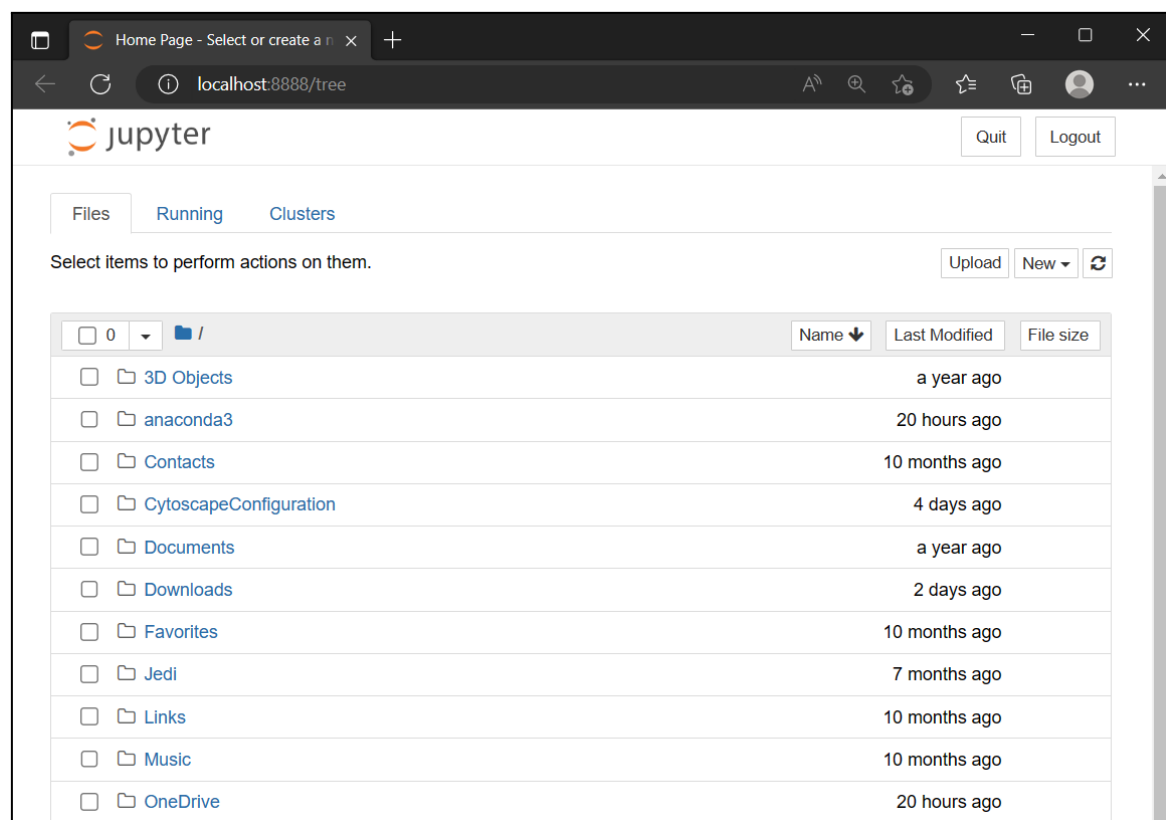
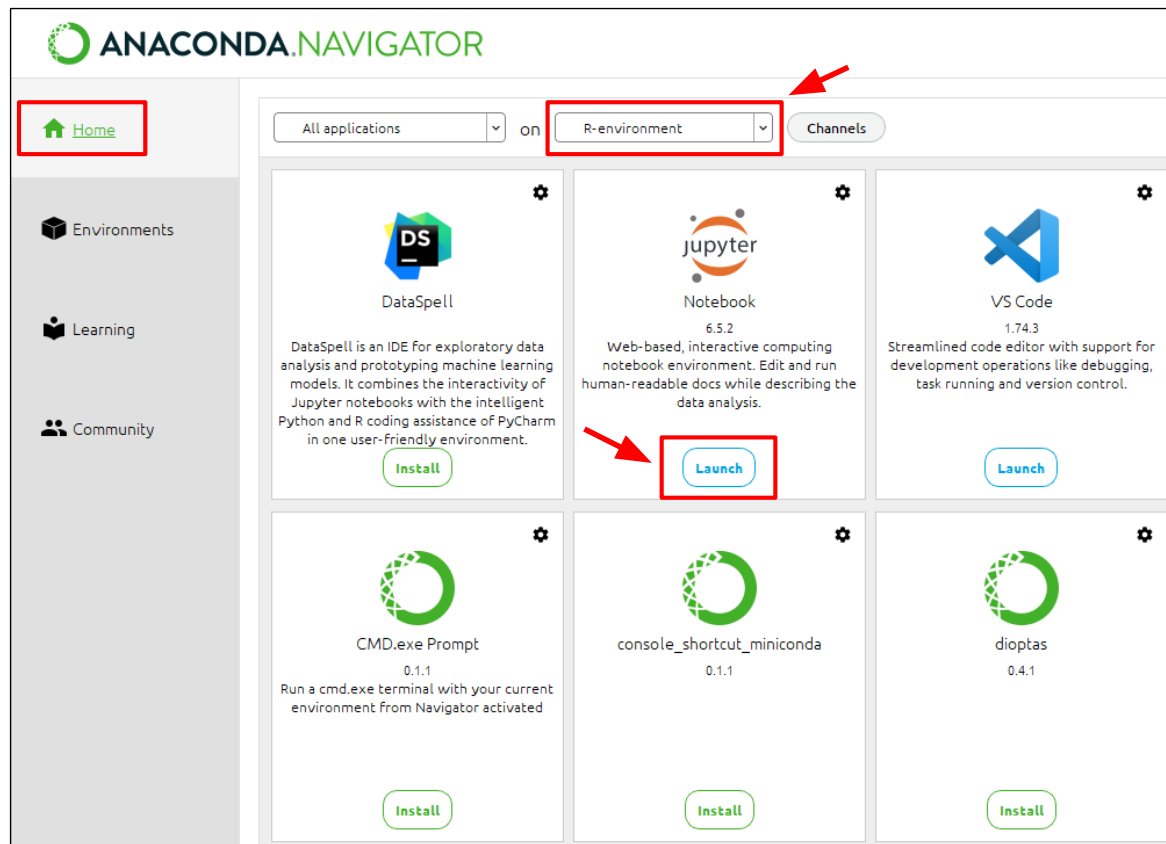
Click on Add → enter the channel name in the dialog box → Hit enter. Once all the channels are added. Press Update channels



**Step 2:** Go to → Environments → Create (found at the bottom of the page). This will open a new dialog box where we enter a name for the new environment (avoid spaces). **Include both Python and R packages.** Select the highest version available for R → Enter Create.



It will take a while to load all the packages. Once complete, we can again go to Home Page. As shown in the figure, choose the R-environment to use the applications. Then launch your Jupyter Notebook from there. In your default browser, a new tab will open, allowing you to navigate into the notebook.

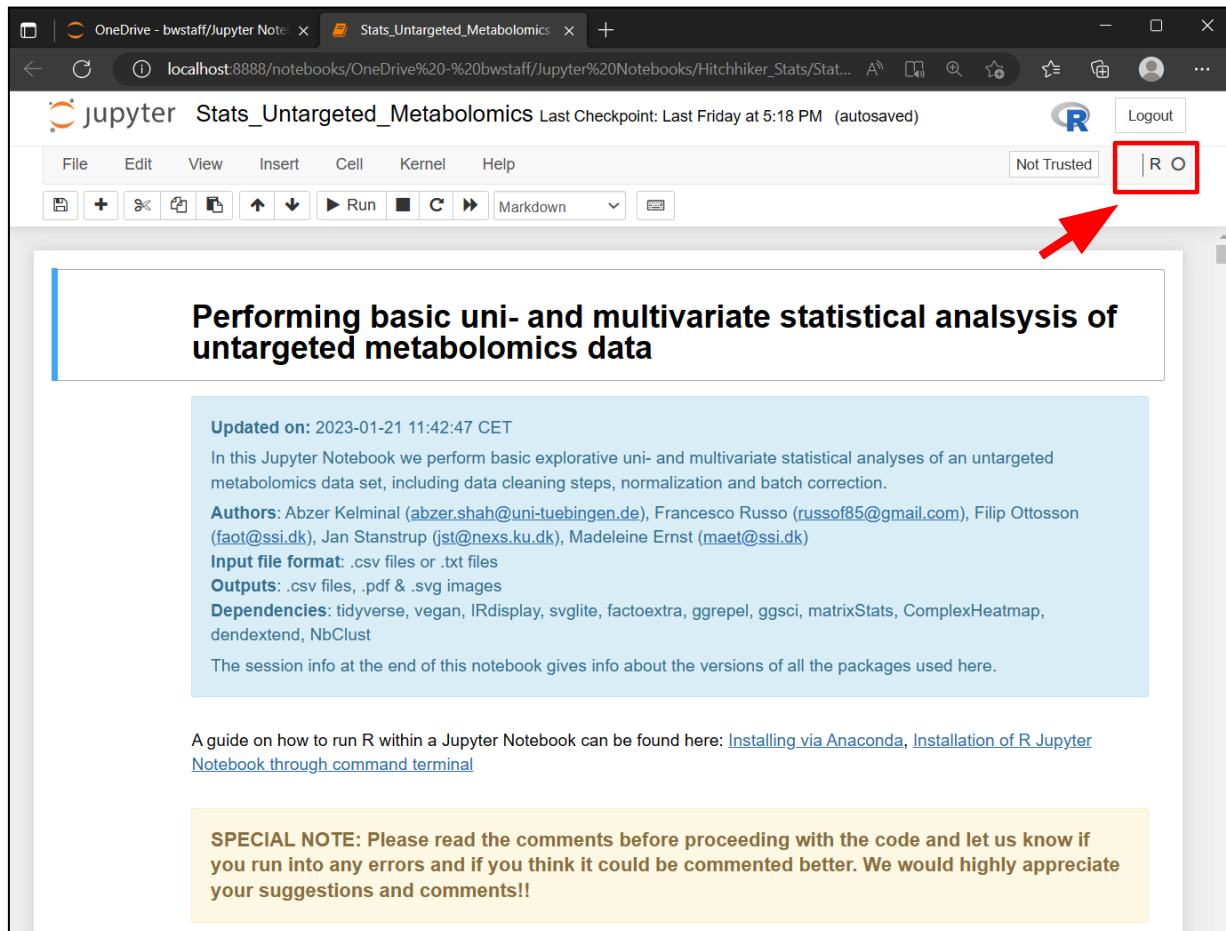


Once you open the Jupyter Notebook, on the upper right corner, you can see which kernel is used for the notebook (R or Python). Since we navigated via



“R-environment”, we have the R-kernel. We can also open Python notebooks in this environment.

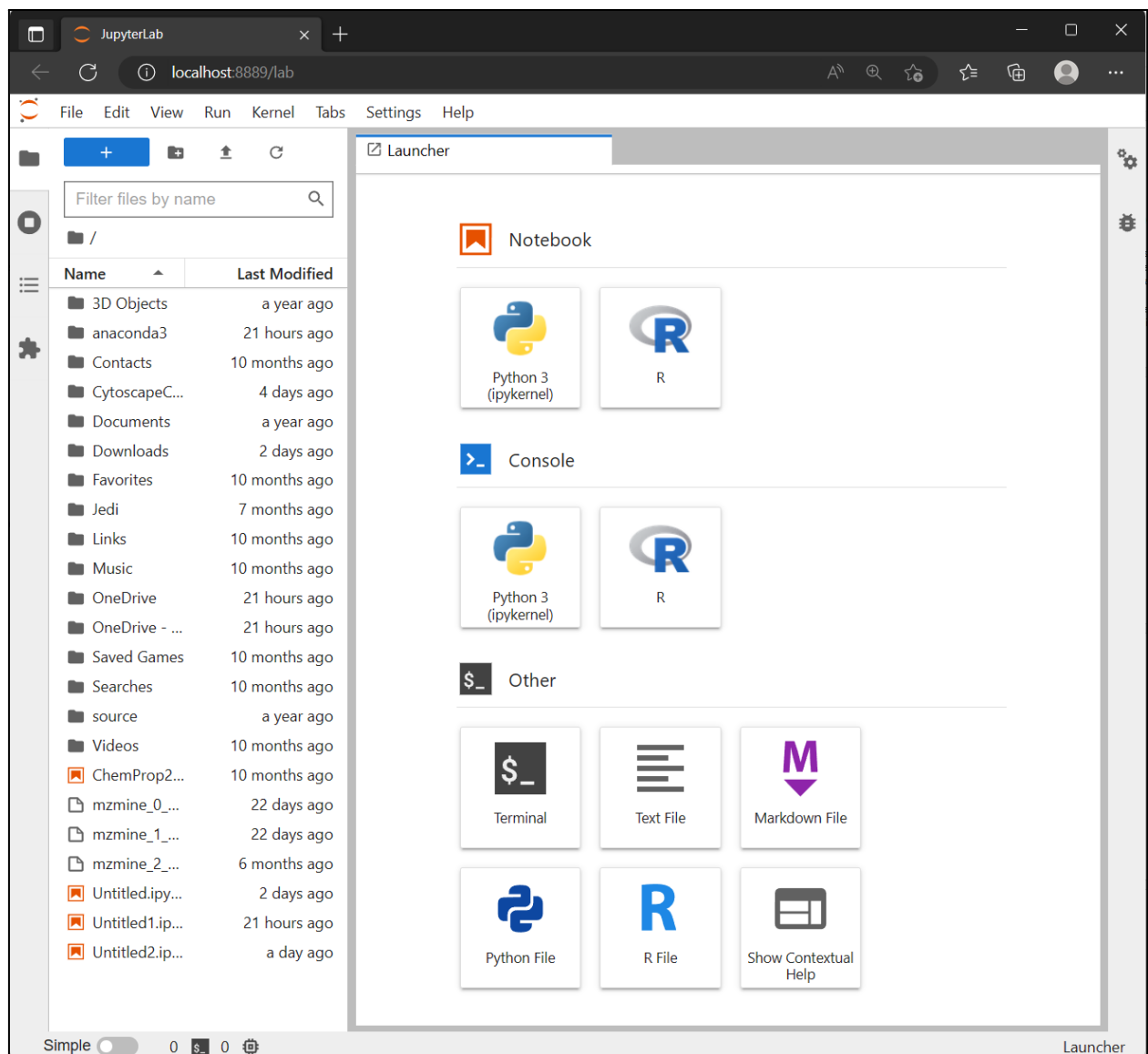
**Note:** If we launched the Jupyter notebook from the base repository, we will only have Python kernel and therefore, we cannot run the R-notebooks.



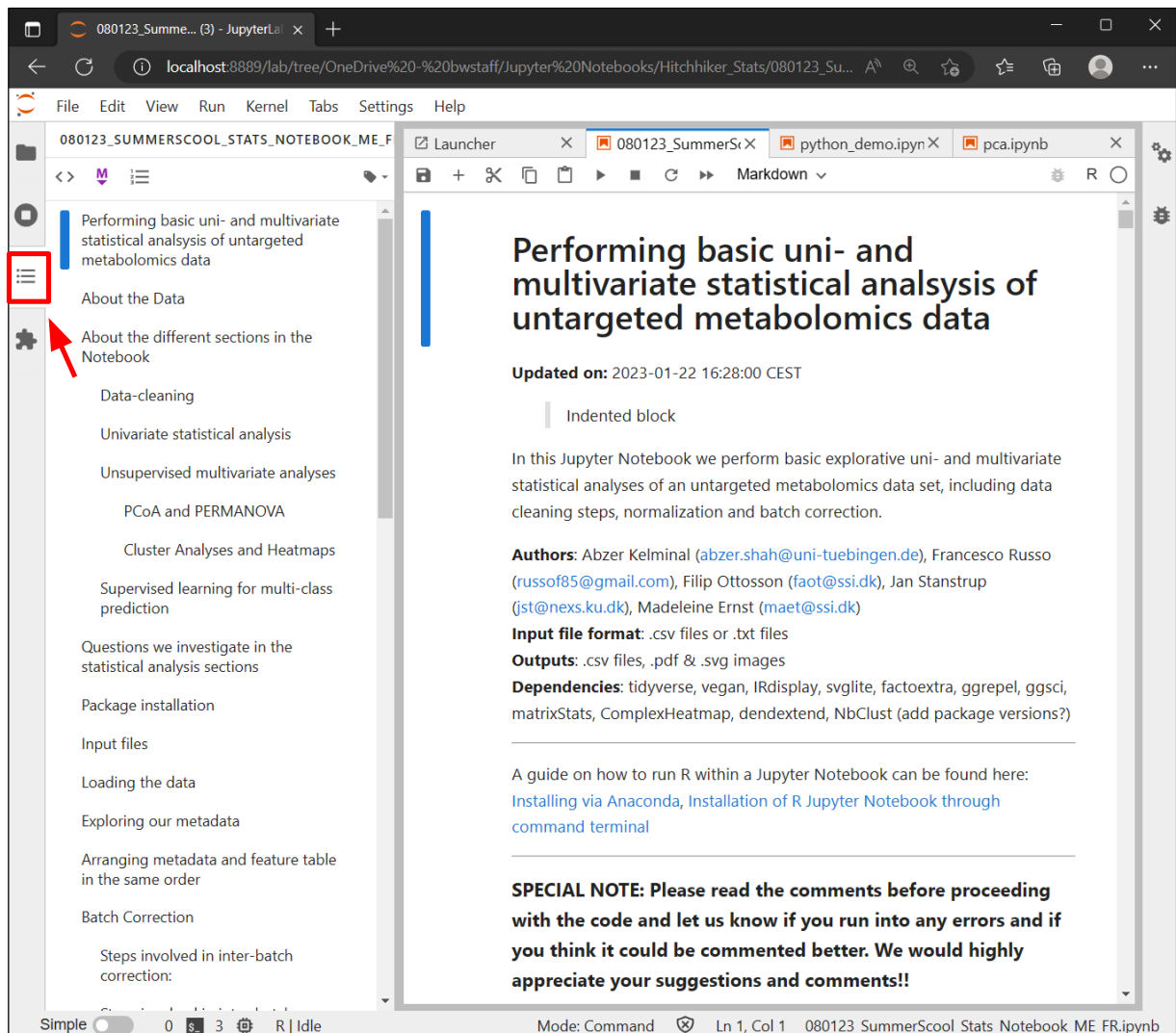
The screenshot shows a web browser window displaying a Jupyter Notebook titled "Stats\_Untargeted\_Metabolomics". The browser's address bar shows the URL "localhost:8888/notebooks/OneDrive%20-%20bwtstaff/Jupyter%20Notebooks/Hitchhiker\_Stats/Stat...". The Jupyter interface includes a top bar with the Jupyter logo, the notebook title, and a "Logout" button. Below this is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, and Help. A "Not Trusted" warning is visible on the right side of the menu bar. The "Kernel" menu is open, showing a list of available kernels: "Python 3" and "R". The "R" kernel is highlighted with a red box, and a red arrow points to it from the right. The main content area of the notebook displays the title "Performing basic uni- and multivariate statistical analysis of untargeted metabolomics data" in a large, bold font. Below the title is a light blue box containing metadata: "Updated on: 2023-01-21 11:42:47 CET", a description of the notebook's purpose, a list of authors with their email addresses, input and output file formats, and a list of dependencies. At the bottom of the notebook content, there is a yellow box with a "SPECIAL NOTE" asking users to read comments and provide feedback.

# Installing JupyterLab in an R-environment (Optional):

JupyterLab gives the flexibility to open multiple Jupyter notebooks at a time, similar to opening multiple tabs in a browser.

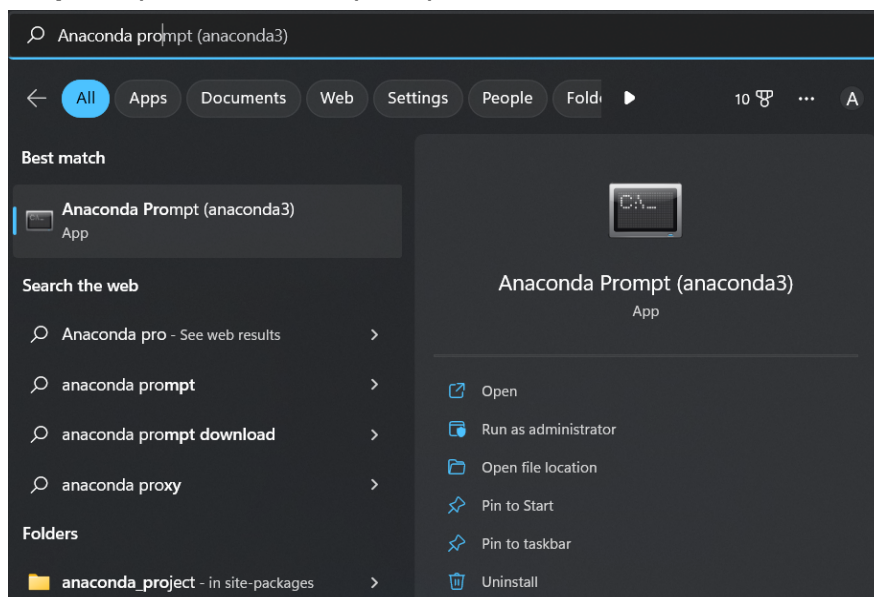


In addition to that, JupyterLab provides a table of contents, which makes it easy to toggle among different sections of the notebook.



The base environment comes with JupyterLab extension. In order to add it to our 'R-environment', we need to conda install it from Anaconda prompt

### Step 1: open Anaconda prompt



**Step 2:** Llist all the environments in your anaconda by typing the command 'conda env list':

```
(base) C:\Users\abzer>conda env list
# conda environments:
#
base                * C:\Users\abzer\anaconda3
R-environment        C:\Users\abzer\anaconda3\envs\R-environment
```

**Step 3:** Install Jupyterlab version 3.3.2 from conda-forge channel into our R-environment such as:

conda install -n "your\_environment\_name" -c "channel-name" "package-name"

```
(base) C:\Users\abzer>conda install -n R-environment -c conda-forge jupyterlab=3.3.2
```

In case you face inconsistency issues with other packages, some new packages will be downloaded and some will be downgraded automatically to solve the issue. **You will be asked whether to proceed or not. Enter y.** Your prompt will look like:

The following NEW packages will be INSTALLED:

babel	conda-forge/noarch::babel-2.11.0-pyhd8ed1ab_0
brotlipy	conda-forge/win-64::brotlipy-0.7.0-py39ha55989b_1005
certifi	conda-forge/noarch::certifi-2022.12.7-pyhd8ed1ab_0
charset-normalizer	conda-forge/noarch::charset-normalizer-2.1.1-pyhd8ed1ab_0
cryptography	conda-forge/win-64::cryptography-39.0.0-py39hb6bd5e6_0
json5	conda-forge/noarch::json5-0.9.5-pyh9f0ad1d_0
jupyterlab	pkgs/main/noarch::jupyterlab-3.3.2-pyhd3eb1b0_0
jupyterlab_server	conda-forge/noarch::jupyterlab_server-2.19.0-pyhd8ed1ab_0
pyopenssl	conda-forge/noarch::pyopenssl-23.0.0-pyhd8ed1ab_0
pysocks	conda-forge/noarch::pysocks-1.7.1-pyh0701188_6
pytz	conda-forge/noarch::pytz-2022.7.1-pyhd8ed1ab_0
pywin32	conda-forge/win-64::pywin32-304-py39h99910a6_2
requests	conda-forge/noarch::requests-2.28.2-pyhd8ed1ab_0
urllib3	conda-forge/noarch::urllib3-1.26.14-pyhd8ed1ab_0
win_inet_pton	conda-forge/noarch::win_inet_pton-1.1.0-pyhd8ed1ab_6

The following packages will be DOWNGRADED:

jupyter_server	2.1.0-pyhd8ed1ab_0 --> 1.23.5-pyhd8ed1ab_0
----------------	--

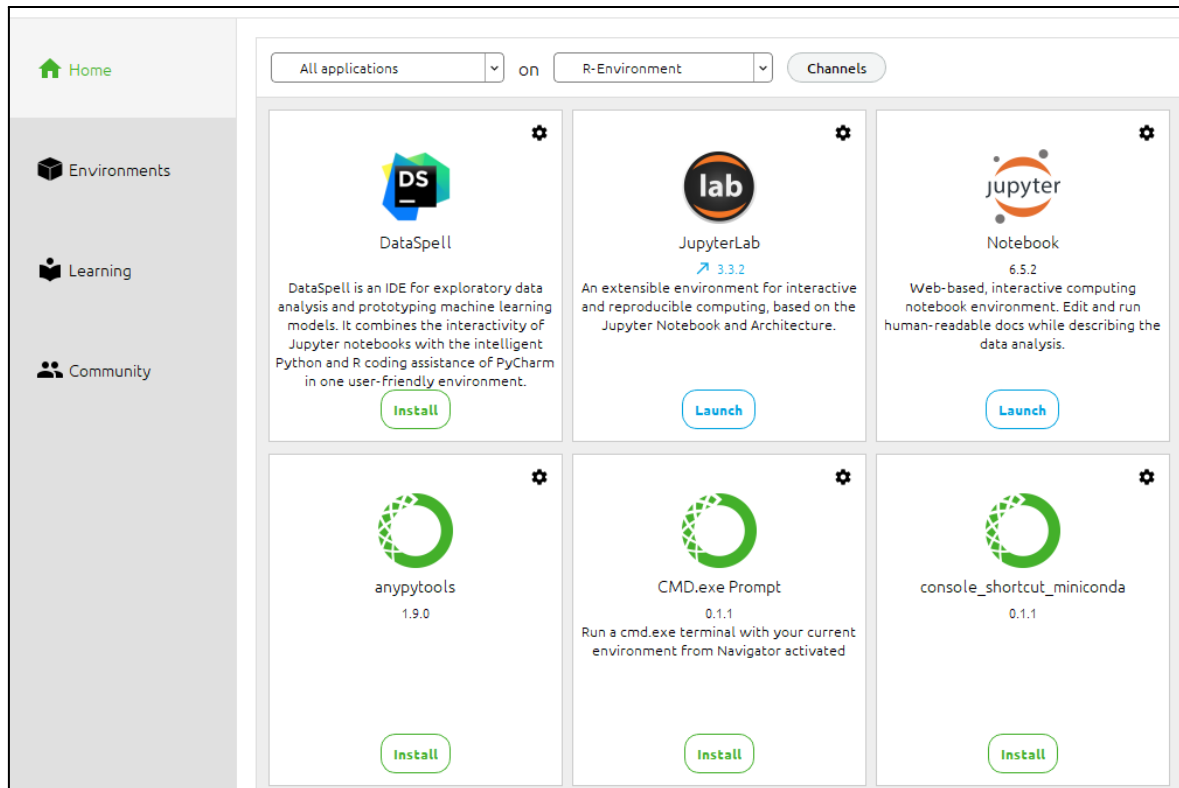
Proceed ([y]/n)? y

**Enter y**

Downloading and Extracting Packages

Preparing transaction: done  
Verifying transaction: done  
Executing transaction: done

Once installed you can close the prompt and open Anaconda Navigator → Homepage → Change the environment to R-Environment → You will see the JupyterLab widget → You can launch from there and continue to your notebook



# Troubleshooting:

## Critical:

Both pip and conda are package managers. Users who already have pip installed jupyter (with python kernel), try to install IRkernel with pip as well. It is not advised to mix up pip and conda installations. You can follow the example here: <https://developers.refinitiv.com/en/article-catalog/article/setup-jupyter-notebook-r>

For first-time users following this protocol: we are installing everything with conda (python as a part of the base environment and R as an additional environment). It is possible, however, to encounter certain errors, some of which are listed below:

## 1) DLL Error in Anaconda Prompt:

Error message:

```
KeyError: 'pkgs_dirs'  
ImportError: DLL load failed while importing shell: Can not find  
procedure.
```

While trying to install things from the Anaconda command prompt using conda install, sometimes this error might occur in Windows.

One way to solve the error is downgrading pywin32 from higher versions (304) to version 228. Pywin32 is a python extension for Windows.

```
C:\Users\abzer>pip install --upgrade pywin32==228
```

Then you can try installing the same package again. In most cases, it works.

## 2) Formatting the conda shell

By typing '**conda init**' command, we initialize or format the shell permanently. This will resolve the errors in some cases.

```
C:\Users\abzer>conda init
```

After the command, you will get a message to close the Anaconda prompt and open it again for changes to take place. Wait until you receive the message on the prompt, then close everything (Anaconda prompt as well as Navigator) and open it again.

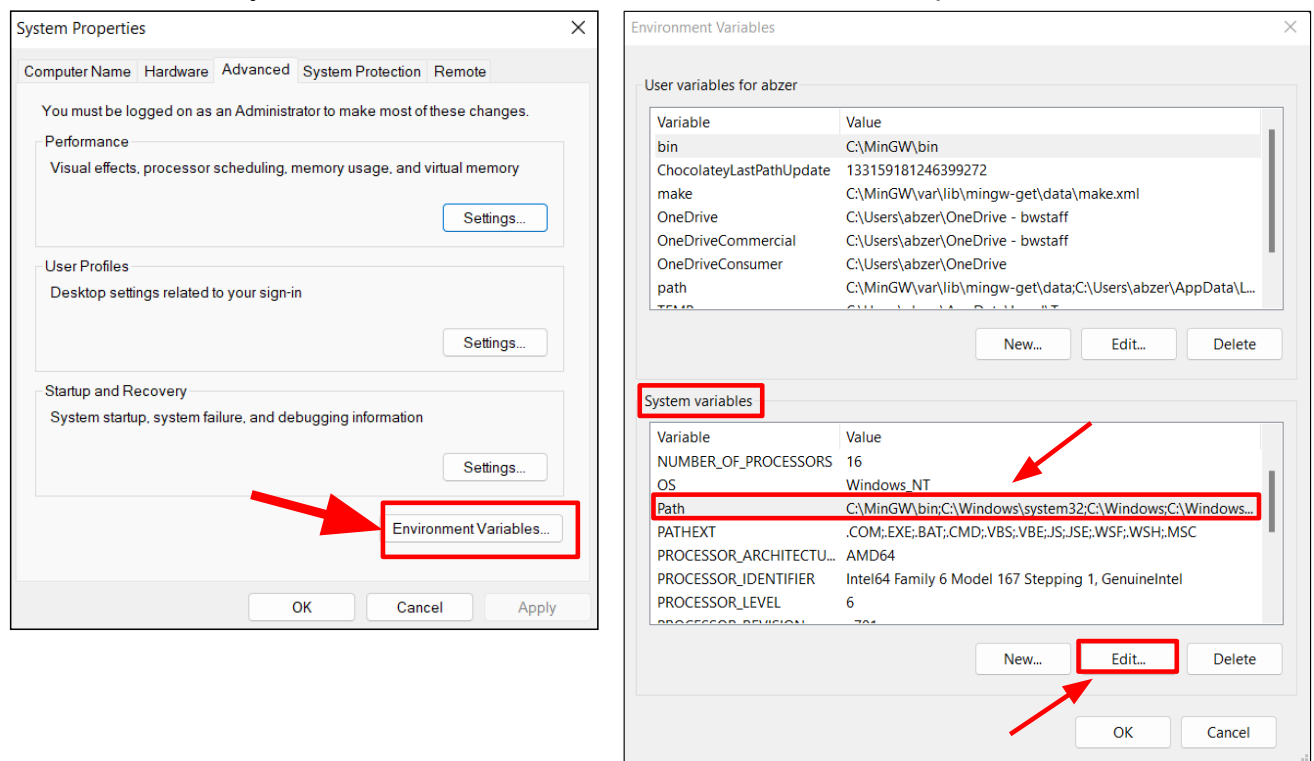
### 3) Failed to add Anaconda to the system PATH:

This is a common error in Windows. Check the environment variables in your PC to see if you have included the path of Anaconda scripts in the environment variables. By adding a path to the environment variable, we allow the system to access the variables from that particular application or environment.

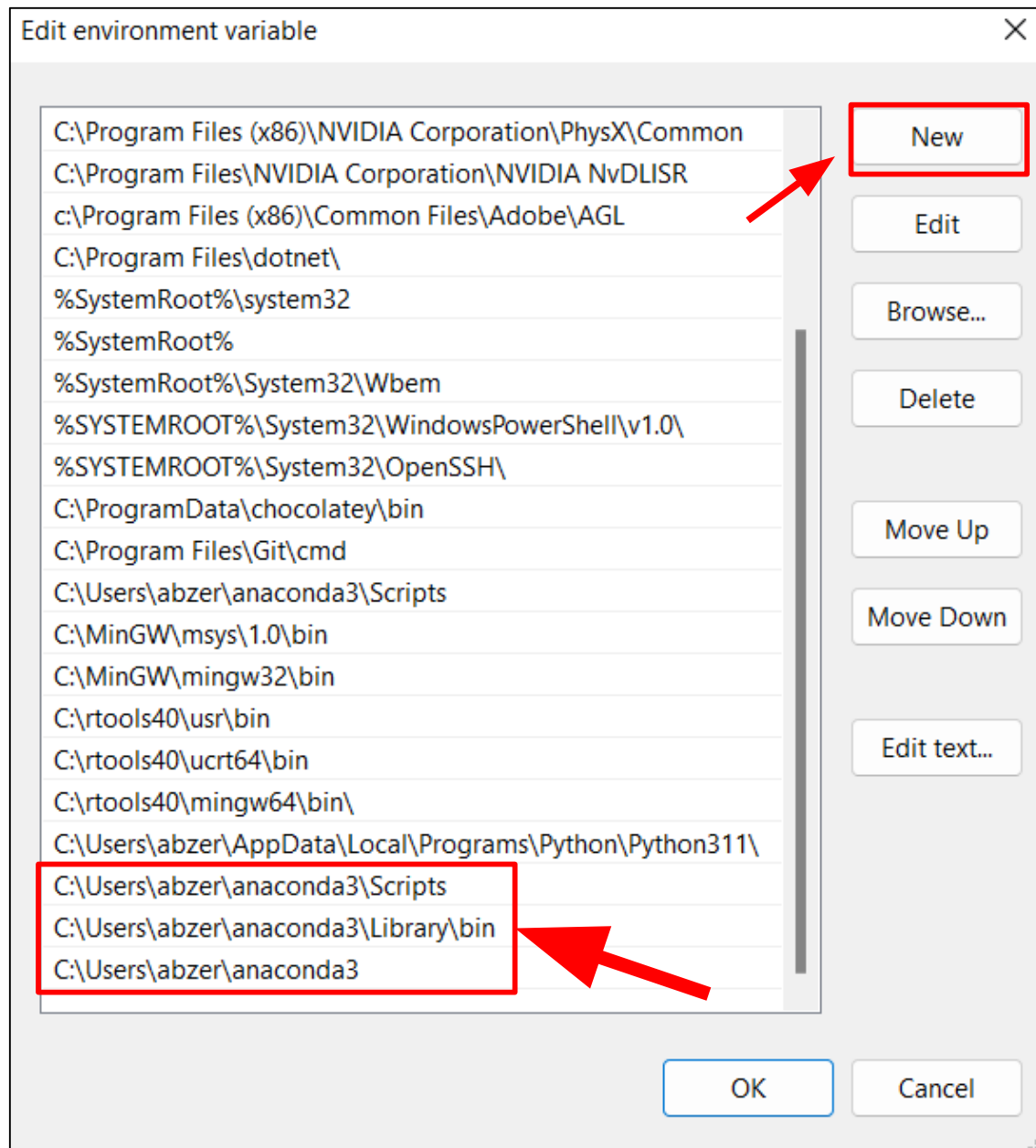
#### To look for Environment Variables in a Windows system:

Windows + I → System → About → Advanced Settings → Go to Environment variables:

As shown in the second image here, the Environmental variables will open a new window. On the system variables sections, click on the 'Path' and press 'Edit'.



This will open a new dialog box where we can look for the Anaconda path, if not available, add the paths as mentioned below:



Finally, press OK for changes to take place. And try to run Anaconda again.

There are several possible errors that can occur. For additional guidance on troubleshooting, please refer to Anaconda's user guide: <https://docs.anaconda.com/anaconda/user-guide/troubleshooting/>

#### 4) R Kernel always busy in Jupyter Notebook:

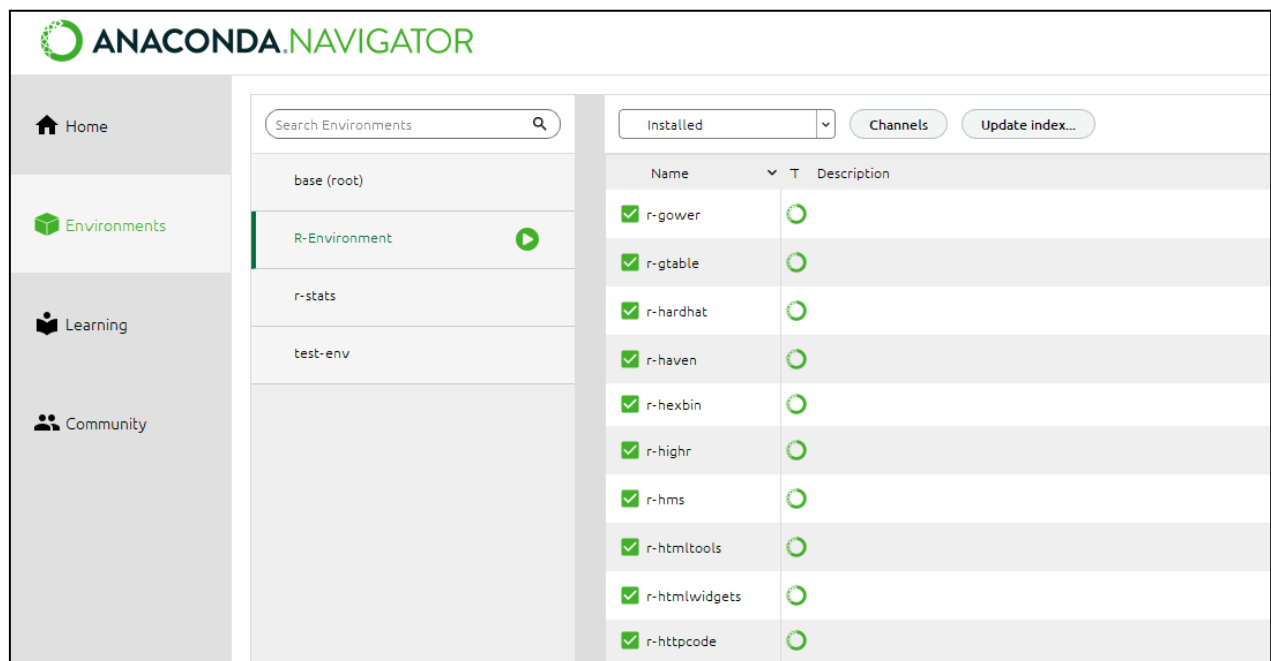
When opening R Jupyter Notebook in Anaconda, make sure to enter via the right environment. Opening R Notebook via the 'base' environment would cause a kernel error as the base environment does not have any R-packages or R kernel, it will not be able to recognize the R codes.



However, even after launching the Notebook via an R-environment, sometimes, one might run into a **'kernel busy' error**. This might be due to the 'IRkernel' (interactive R Kernel) not being visible to Jupyter Notebook.

For Anaconda users, we can inspect if we have IR kernel installed in our environment by the following steps:

Anaconda Homepage → Go to Environments → Select the R-Environment → Check at the right section containing the list of packages for 'r-irkernel'.



If it is not present we need to install it. If present, we need to make it visible to the Jupyter Notebook.

### To install IRkernel from the Anaconda prompt:

```
(base) C:\Users\abzer>conda install -n R-environment -c conda-forge  
r-irkernel
```

### To make the Kernel visible:

Step 1: Active the R-environment and enter R inside the R-environment

```
(base) C:\Users\abzer>conda activate R-environment  
(R-environment) C:\Users\abzer> R
```

This will open R in the Anaconda prompt like this:

```
R version 4.1.3 (2022-03-10) -- "One Push-Up"  
Copyright (C) 2022 The R Foundation for Statistical Computing  
Platform: x86_64-w64-mingw32/x64 (64-bit)
```

```
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.
```

```
Natural language support but running in an English locale
```

```
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.
```

```
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.
```

Enter the following in your R workspace:

```
> IRkernel::installspec(user = FALSE)
```

Finally, quit the R workspace:

```
> q()  
Save workspace image? [y/n/c]: n
```

## 5) Error during `install.packages()` in R Jupyter Notebook

As the first step of running an R Jupyter Notebook, we will install the packages necessary to run the notebook. For example, here, in the following example, when trying to install the package “xgboost” for supervised classification, it throws an error saying the installation had “non-zero exit status”, meaning the installation failed. Furthermore, we get a message that an older binary version is available, but the latest source version needs compilation.

For all the packages that were written in C, C++ or Fortran, their source file needs to be compiled (or translated) into an object file (binary file) for the computer to understand the task.

```
install.packages("xgboost")
```

```
There is a binary version available but the source version is later:
      binary  source needs_compilation
xgboost 1.6.0.1 1.7.3.1             TRUE
```

```
installing the source package 'xgboost'
```

```
Warning message in install.packages("xgboost"):
"installation of package 'xgboost' had non-zero exit status"
```

There are several ways to solve the compilation error:

1. One way is to just download the available binary version:

```
•[6]: install.packages("xgboost",
      available = available.packages(type = "win.binary",
      repos="https://cran.r-project.org"))

package 'xgboost' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
      C:\Users\abzer\AppData\Local\Temp\RtmpiateR6\downloaded_packages
```

2. Another way is to mention at the beginning of the notebook to install the available binary version. This way, we override the compilation step

```
options(install.packages.compile.from.source = "never")
```

```
install.packages("xgboost")
```

3. **(Advanced)** To get the latest version of a package, we can install the package directly through the Anaconda prompt into the respective environment

Before proceeding further, let's check the library path to see in which environment you are in Jupyter Notebook. This is useful when you have multiple environments.

```
.libPaths()
```

```
'C:/Users/abzer/anaconda3/envs/R-environment/Lib/R/library'
```

Then we can conda install the package from the respective channel (here, conda-forge)

```
(base) C:\Users\abzer>conda install -n R-environment -c conda-forge r-xgboost
```

Few new packages might be installed as the dependencies. In the Anaconda prompt you will be asked whether to proceed with the installation. Enter 'y'.

```
Proceed ([y]/n)? y
```

For Example: With xgboost, the previous error said the available binary version was 1.6.0.1, but the latest source version was 1.7.3.1. We installed the binary version 1.6.0.1 since we couldn't compile the latest source code. The latest available version installed with 'conda install' is 1.7.1.1 version.

[5]: # List all packages where an update is available  
old.packages()

A matrix: 9 × 6 of type chr

	Package	LibPath	Installed	Built	ReposVer	Repository	
	curl	curl	C:/Users/abzer/anaconda3/envs/R-Environment/Lib/R/library	4.3.3	4.1.3	5.0.0	https://cran.r-project.org/src/contrib
	future	future	C:/Users/abzer/anaconda3/envs/R-Environment/Lib/R/library	1.30.0	4.1.3	1.31.0	https://cran.r-project.org/src/contrib
	gargle	gargle	C:/Users/abzer/anaconda3/envs/R-Environment/Lib/R/library	1.2.1	4.1.3	1.3.0	https://cran.r-project.org/src/contrib
	glmnet	glmnet	C:/Users/abzer/anaconda3/envs/R-Environment/Lib/R/library	4.1-2	4.1.3	4.1-6	https://cran.r-project.org/src/contrib
	nlme	nlme	C:/Users/abzer/anaconda3/envs/R-Environment/Lib/R/library	3.1-161	4.1.3	3.1-162	https://cran.r-project.org/src/contrib
	sourcetools	sourcetools	C:/Users/abzer/anaconda3/envs/R-Environment/Lib/R/library	0.1.7	4.1.3	0.1.7-1	https://cran.r-project.org/src/contrib
	utf8	utf8	C:/Users/abzer/anaconda3/envs/R-Environment/Lib/R/library	1.2.2	4.1.3	1.2.3	https://cran.r-project.org/src/contrib
	xfun	xfun	C:/Users/abzer/anaconda3/envs/R-Environment/Lib/R/library	0.36	4.1.3	0.37	https://cran.r-project.org/src/contrib
	xgboost	xgboost	C:/Users/abzer/anaconda3/envs/R-Environment/Lib/R/library	1.7.1.1	4.1.3	1.7.3.1	https://cran.r-project.org/src/contrib