

Installation of R in Linux System (Ubuntu)



Prepared by-
Anisur Rahman
Research Fellow
Bioinformatics Division
National Institute of Biotechnology
Email: arriyaz.nstu@gmail.com

1. Create a new R environment and install r-base and some important packages all together

```
conda create -y -n r-env r-essentials r-base
```

You should add **r-essentials** along with **r-base** as it will not only install all the important r packages but also help during **IRkernel** installation.

1.1. To install a specific R version, let's say R 3.6

```
conda create -n r-env3.6 r-essentials r-base=3.6
```

To know more about installing specific versions of a tool you can see [conda cheat sheet](#).

2. Add IRkernel for jupyter notebook or jupyterlab

After installing R you may want to run your R codes in jupyter notebooks. In that case, you have to install the R kernel. This kernel will make your R available for jupyter notebook or lab.

Let's say you install **R version 4.0.1** in the **r-env** environment. So, to make this version of R accessible for jupyter notebook/lab you have to install **IRkernel** in this environment.

First, activate the environment.

```
conda activate r-env
```

Now run the following command sequentially.

2.1. Install necessary dependencies:

```
sudo apt-get install libzmq3-dev libcurl4-openssl-dev libssl-dev  
jupyter-core jupyter-client
```

When you have run this command once. If you create a new conda environment and want to install IRkernel for a new R version inside that environment you don't have to run this command again.

2.2. Now, run R in the terminal:

```
R
```

You will see a typical R console in the terminal. Now we will install our kernel in this console.

2.3. Install IRkernel with some necessary packages:

```
install.packages(c('repr', 'IRdisplay', 'IRkernel'), type = 'source')
```

2.4. Make kernel is available to Jupyter:

Here is the trick. If we install kernelspecs with default names, every time it will replace the previous kernel from a different conda environment.

The aim of creating a conda environment is to manage different versions of tools (in this case different R versions) simultaneously.

So, you have to install different kernels with different names for every version of R. In that case we will be able to run r codes in a different version of R from the jupyter lab/notebook.

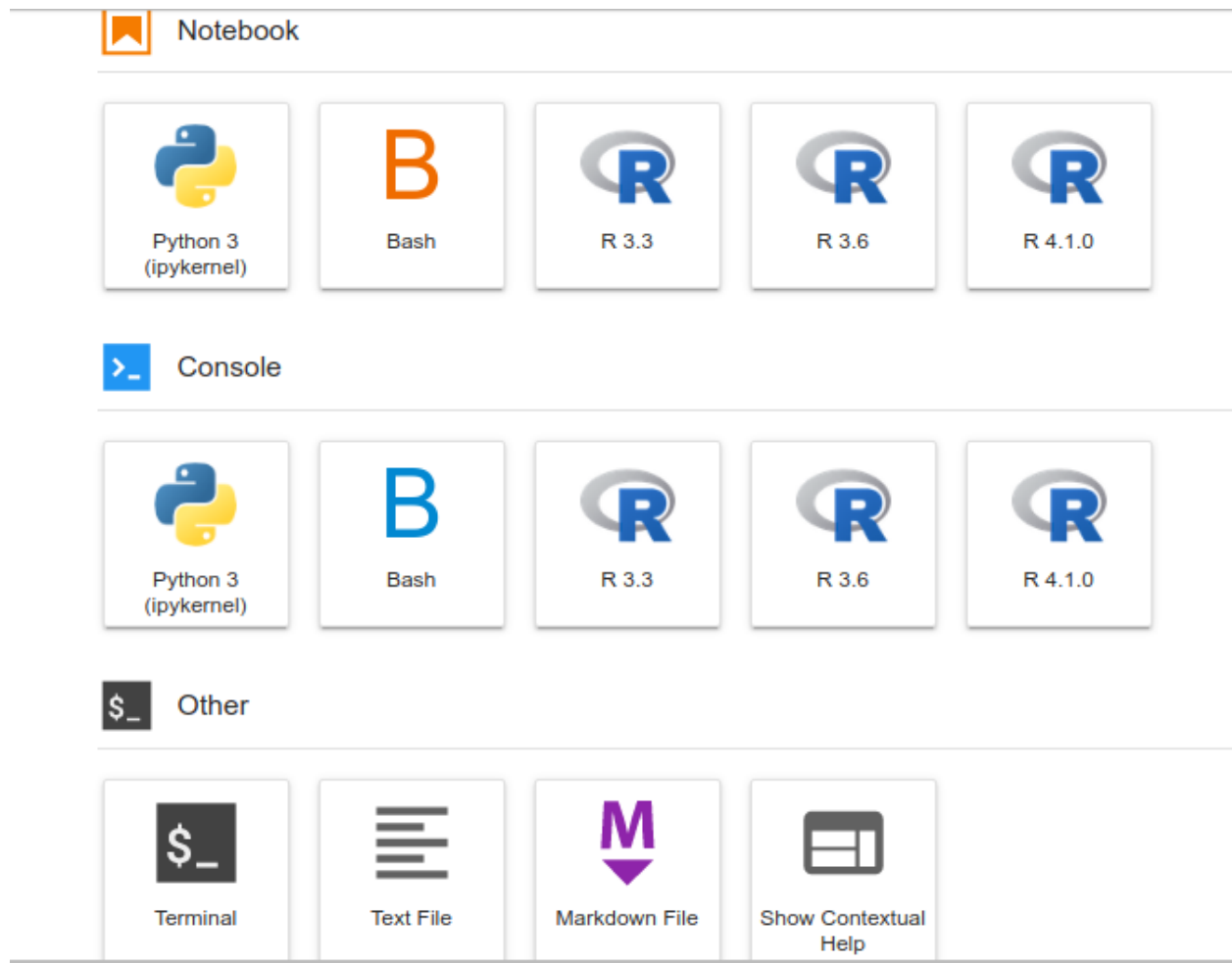
Even we don't have to activate that particular conda environment to run R inside that environment. We can just run jupyter lab/notebook from conda base and select a particular R kernel.

Here, we have installed R 4.0.1 inside the r-env environment, so for simplicity, we will give kernel name according to environment name and displayname of the kernel according to R version (it is not a specific rule, I am doing this for my ease of use)

```
IRkernel::installspec(name = 'ir-env', displayname = 'R 4.0.1')
```

Obviously run this command in the R console, not in bash terminal.

After installing the kernel in this way you can access this kernel from jupyter lab/notebook launcher display. Example of typical launcher display:



NB: If you install another R version in another conda environment, don't forget to change the **name** and **displayname** option in the above code.

For more details about IRkernel installation, see these three links below:

1. [Developer instruction](#)
2. [Github link of the IRkernel package](#)
3. [Man page](#)

3. Uninstall a kernel from jupyter

First, activate that specific environment from which you want to remove the kernel
Then run the following code in the terminal

```
jupyter kernelspec list
```

You will see available kernel and their paths as below

```
$ jupyter kernelspec list
Available kernels:
  bash          /home/arriyaz/.local/share/jupyter/kernels/bash
  ir-env        /home/arriyaz/.local/share/jupyter/kernels/ir-env
  ir-env3       /home/arriyaz/.local/share/jupyter/kernels/ir-env3
  ir33          /home/arriyaz/.local/share/jupyter/kernels/ir33
  python3       /home/arriyaz/miniconda3/share/jupyter/kernels/python3
```

To remove a specific kernel, let's say **ir33** from above image run the following code;

```
jupyter kernelspec uninstall ir33
```

Or,

```
jupyter kernelspec remove ir33
```

4. Once you install r via conda, you will be able to install packages either by,

```
install.packages("PackageName")
```

Or by,

```
conda install -c channelname r-packagename
```

5. Completely remove R along with installed packages from Ubuntu

The following code will remove R that was installed by **sudo apt**, not by conda.

To remove core-r

```
sudo apt-get remove r-base-core  
sudo apt-get remove r-base  
sudo apt-get autoremove
```

But The above code will not remove the installed packages.

To remove all installed packages first run the following code from the terminal, to get the list of folders where all the packages were installed:

```
R -e '.libPaths()'
```

The above code will give output as like below:

```
> .libPaths()  
[1] "/home/arriyaz/R/x86_64-pc-linux-gnu-library/4.1"  
[2] "/usr/local/lib/R/site-library"  
[3] "/usr/lib/R/site-library"  
[4] "/usr/lib/R/library"
```

Now you have to remove/delete each of the directories by **rm** command along with the **-r** flag.
For example;

```
sudo rm -r /home/arriyaz/R/x86_64-pc-linux-gnu-library/4.1
```

Run such a command for every directory in the list.

Thus you will completely remove R as well as installed packages from your ubuntu.