

Using Conda to manage R/Python environments and packages

This tutorial is applicable to Unix-like machines (e.g., linux and MacOS). Conda is available in both distributions of [anaconda](#) and [miniconda](#).

Conda vs Rstudio

- **Conda** is designed to manage multiple R/Python environments and their associated packages. Each environment may have a different version of R/Python, as well as different versions of packages. Conda can also be used to install built software, such as gcc or samtools, which is specially convenient for a non-root user, since one has to typically compile and install a software from source code without Conda.
- **Rstudio** is an IDE (integrated development environment) widely used for R users. Within it, one can easily install R packages. It can also be used as a Python IDE, but may not be optimized compared to other Python IDEs (e.g., PyCharm). Rstudio relies on a R environment, which can be provided by Conda. You can switch R environments (3.6 vs 4.0) using Conda and relaunch Rstudio so that Rstudio can have a different R and its associated packages.

Anaconda vs Miniconda

- **Anaconda** can be thought of as the data scientist's hardware store. It's got everything you need. From tools for exploring datasets, to tools for modelling them, to tools for visualizing what you've found. Everyone can access the hardware store and all the tools inside.
- **Miniconda** is the workbench of a data scientist. Every workbench starts clean with only the bare necessities. But as a project grows, so do the number of tools on the workbench. They get used, they get changed, they get swapped. Each workbench can be customised however a data scientist wants. One data scientist's workbench may be completely different to another, even if they're on the same team.
- **Use anaconda:** if you're after a one size fits all approach which works out of the box for most projects and have 3 GB of space on your computer.
- **Use miniconda:** if you don't have 3 GB of space on your computer and prefer a setup that has only what you need.

Download the installer

- [Anaconda installer](#)
- [Miniconda installer](#)

Installation

1. Open a terminal and `cd` to the directory where the installer locates.

2. Miniconda:

```
bash Miniconda3-latest-Linux-x86_64.sh
```

3. Anaconda:

```
bash Anaconda-latest-Linux-x86_64.sh
```

4. Follow the prompts on the installer screens. If you are unsure about any setting, accept the defaults. You can change them later.
5. To make the changes take effect, close and then re-open your terminal window.
6. Test your installation. In your terminal window or Anaconda Prompt, run the command `conda list`. A list of installed packages appears if it has been installed correctly.
7. Should I add conda to the macOS or Linux PATH?

During installation, you will be asked “Do you wish the installer to initialize conda by running `conda init`?” We recommend “yes”. If you enter “no”, then conda will not modify your shell scripts at all. In order to initialize after the installation process is done, first run `source <path to conda>/bin/activate` and then run `conda init`.

Managing environments

With conda, you can create, export, list, remove, and update environments that have different versions of Python, R and/or packages installed in them. Switching or moving between environments is called activating the environment.

1. To create an environment:

```
conda create --name myenv
```

Replace myenv with the environment name.

2. To create an environment with a specific version of Python:

```
conda create -n myenv python=3.8
```

3. Activating an environment:

```
conda activate myenv
```

Replace myenv with the environment name or directory path.

4. Deactivating an environment:

```
conda deactivate
```

5. Determining your current environment:

```
conda info --envs
```

6. Viewing a list of your environments:

```
conda info --envs
```

OR

```
conda env list
```

7. Using pip in an environment:

```
8. conda install -n myenv pip
```

```
9. conda activate myenv
```

```
pip <pip_subcommand>
```

10. Removing an environment:

```
conda remove --name myenv --all
```

You may instead use

```
conda env remove --name myenv
```

Using R language with Conda

With conda, you can easily install the R programming language and over 6,000 commonly used R packages for data science.

1. Create a new conda environment with all the r-essentials conda packages built from CRAN:

```
conda create -n r_env r-essentials r-base
```

Replace `r_env` with your environment name.

2. Activate the environment:

```
conda activate r_env
```

3. Install R packages using conda: when using conda to install R packages, you will need to add r- before the regular package name.

```
conda install -c r r-{name_of_package}
```

4. You can install a bioconductor package:

```
conda install -c bioconda bioconductor-{name_of_package}
```

For example:

```
conda install -c bioconda bioconductor-deseq2
```

5. Update all of the packages and their dependencies with one command:

```
conda update r-caret
```