

(. / . / . / . / . /)

Using R language with Anaconda

With Anaconda, you can easily install the R programming language and over 6,000 commonly used R packages for data science. You can also create and share your own custom R packages.

i Note

When using conda to install R packages, you will need to add `r-` before the regular package name. For instance, if you want to install `rbokeh`, you will need to use `conda install r-rbokeh` or for `rJava`, type `conda install r-rjava`.

The R Essentials bundle contains approximately 200 of the most popular R packages for data science, including the `IRKernel`, `dplyr`, `shiny`, `ggplot2`, `tidyr`, `caret`, and `nnet`. It is used as an example in the following guides.

R is the default interpreter installed into new environments. You can specify the R interpreter with the `r-base` package. Unless you change the R interpreter, conda will continue to use the default interpreter in each environment.

To run the commands below on Windows, use Start - Anaconda Prompt. On macOS or Linux, open a terminal.

Updating R packages

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

```
conda update r-caret
```

- If a new version of a package is available in the R channel, you can use `conda update` to update specific packages.

Creating and sharing custom R bundles

Creating and sharing custom R bundles is similar to creating and sharing conda packages.

EXAMPLE: Create a simple custom R bundle metapackage named “Custom-R-Bundle” that contains several popular programs and their dependencies:

```
conda metapackage custom-r-bundle 0.1.0 --dependencies r-irkernel jupyter r-ggplot2 r-dplyr --summary "My
```


Share the new metapackage by uploading it to your channel on [Anaconda Cloud \(https://anaconda.org\)](https://anaconda.org):

```
conda install anaconda-client
anaconda login
anaconda upload custom-r-bundle-0.1.0-0.tar.bz2
```

Anyone can now access your custom R bundle from any computer:

```
conda install -c <your anaconda.org username> custom-r-bundle
```

Creating an environment with R

 v: latest ▼

- [Download and install Anaconda \(https://docs.anaconda.com/anaconda/install/\)](https://docs.anaconda.com/anaconda/install/).

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

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

3. Activate the environment:

```
conda activate r_env
```

4. List the packages in the environment:

```
conda list
```

The list shows that the package r-base is installed and r is listed in the build string of the other R packages in the environment.

Anaconda Navigator, the Anaconda graphical package manager and application launcher, creates R environments by default.

Creating a new environment with R

When creating a new environment, you can use R by explicitly including r-base in your list of packages.

With conda 4.6:

```
conda create -n r-environment r-essentials r-base
conda activate r-environment
```

Mirroring the R channel

Many Anaconda Enterprise customers maintain a local mirror of the R channel.

When mirroring the R channel for the first time, clean the existing packages by running the command `anaconda-server-sync-conda` with the option `--clean`.

Uninstalling R Essentials

To uninstall the R Essentials package, run: `conda remove r-essentials`

i Note

This removes only R Essentials and disables R language support. Other R language packages are not removed.

Using MRO with Anaconda

If you prefer to use the Microsoft R Open (MRO) platform with Anaconda, as opposed to R, you can switch the default R interpreter from R to MRO. To get MRO, you need to explicitly include `mro-base`. Anaconda will maintain an archive of MRO packages but will not update MRO packages. Support for MRO packages will be on a case-by-case basis.

If you are using MRO, it is recommended to migrate to R. Follow the [migration directions](#) ([../using-mro/#rst-migration-directions](#)).


Switch the default R interpreter from R to MRO

Run `conda info` and check your version of conda. If your version of conda is below 4.6, run `conda update conda` to update conda to the latest version.

Run:

```
conda config --system --set pinned_packages _r-mutex=*=anacondar*
```

The default R interpreter will switch from R to MRO.

 v: latest ▼

To learn more about how to use MRO with Anaconda, see [Using MRO language with Anaconda](#) ([../using-mro/](#)).

Resources

Here are some additional resources on using Anaconda with the R programming language:

- R Language packages available for use with Anaconda ([../.../packages/r-language-pkg-docs/](#))—There are hundreds of R language packages now available and several ways to get them.
- Navigator tutorial ([../.../navigator/tutorials/](#))—Use the R programming language with Anaconda Navigator. The Anaconda Navigator graphical interface (GUI) makes it easy for even new users to use and run the R language in a Jupyter Notebook.
- Using R packages with Anaconda and Cloudera CDH ([../.../anaconda-scale/cloudera-cdh/](#))—Anaconda Scale provides resource management tools to easily deploy Anaconda across a cluster. It helps you manage multiple conda environments and packages, including Python and R language, on bare-metal or cloud-based clusters.
- Webinar: Anaconda for R Users (<https://speakerdeck.com/chdoig/anaconda-for-r-users>)—Download the slides from the webinar to see how Anaconda makes package, dependency and environment management easy with R language and other Open Data Science languages.

« [Configuring a shared package cache \(../shared-pkg-cache/\)](#)

[Using MRO language with Anaconda » \(../using-mro/\)](#)

[Docs Home \(../.../..\)](#)

[Anaconda Home \(https://www.anaconda.com/\)](https://www.anaconda.com/)

[More Help & Support \(https://www.anaconda.com/support\)](https://www.anaconda.com/support)

© 2020 Anaconda, Inc.

All Rights Reserved.

[Privacy Policy \(https://www.anaconda.com/privacy-policy/\)](https://www.anaconda.com/privacy-policy/)