Managing R Packages

Research Computing
UNIVERSITY OF COLORADO **BOULDER**

Be Boulder.

Managing R Packages

- Mea Trahan
- *Email: Daniel.Trahan@Colorado.edu*
- *RC Homepage: https://www.colorado.edu/rc*
- *RC Email: rc-help@colorado.edu*

- Slides available for download at:
- *https://github.com/ResearchComputing/RInstalls_Fall_2021*

Outline

- Problems with installing Packages with R
- Within R
  - Local installs
  - Problems
- Anaconda
  - Environments
  - Anaconda R packages
  - Mixing CRAN installs with Anaconda

Quick Note

- If there is nothing else you take away from this tutorial, **use Conda instead of base R**

R Packages

- R has many features by default but many user wish to expand what they can run using **R packages**
- R packages are a bundle of functions and utilities that can be loaded into your R environment by used the `library()` function.
- Types of packages you can expect to find include:
  - Genomic packages
  - Ploting packages
  - Parallel packages
  - And much much more!

Installs within R

- R offers an internal installation method that can easily be ran from Rstudio.

```
install.packages('<package-name>')
```

- By default applications will be installed in the location of your R install.
- Installations to another path can be performed with the command:

```
install.packages('<package-name>', lib="<local-path>")
```

- These installs can then be access with the command:

```
library(<package-name>, lib.loc="<local-path>")
```

R Demo: Installing

- Installing Stringr with CRAN

Problems with installing inside R

- Package conflicts
  - R does install dependencies required to build certain packages but does not keep track of them. Some issues may arise when installing different packages that use the same dependancies
- Difficult to debug
  - R will often report errors when installing a package that you might not know how to get around.
- No root privileges.
  - Root privileges might be required when installing an R package without specifying a install directory.
  - Requires management of directories…
  - Lot of work!

Conda

- Conda is the Anaconda foundation's package and environment manager.
  - Classically for Python
  - Grown rapidly over the last few years and has great R support too.
  - Package manager can be installed ether through a full anaconda download or through miniconda.
    - https://docs.conda.io/en/latest/miniconda.html
  - RC's preferred solution for managing your R installations.
- Conda environments carry over a lot of the benefits of custom R install locations but without needing to manage much.

Getting started with Conda

- Once installed, simply open a terminal. A `(base)` prefix will be set in front of your prompt.
  - Windows users: Open up the ‘Anaconda powershell prompt (miniconda3)’
- Once this is done print out the conda version to validate your install with the command:

```
conda -V
```

Conda enviornments.

- Conda operates with varying levels of environments.
  - (base) is top level environment.
  - Should not be modified.
  - Instead create a new environment!
- Create your new R environment with the command:

```
conda create –n <your-environment-name> r-essentials r-base
```

- Then load your new environment:

```
conda activate <your-environment-name>
```

Installing libraries with Conda

- Installing libraries follows a similar process as creation of an environment:

```
conda install <your-package>
```

- No package availability in base anaconda? Google!
  - Anaconda hosts a variety of different channels for your installations.
  - A lot of the time your package might be available in a different channel
  - Channels include:
    - Bioconda
    - Conda-Forge
    - Intel

```
conda install –c <channel> <your-package>
```

Basic conda commands

- List all conda packages

```
conda list
```

- List all conda enviornments

```
conda env list
```

- Install a conda package

```
conda install <package>
```

- Remove a conda package

```
conda remove <package>
```

Mixing Conda and R internal installs

- What if a package is unavailable on Conda and the only way to get it is through R?
- Mix and Match!
  - Since a conda environment is in a local directory. Global install instructions can be used.
  - No individual management of tool chains
  - No need for additional commands or variable setting.
- Since conda manages environments, install conda packages first

Conda and R Demo

- Installing Tidyr and Stringr using anaconda and R respectively…

Conda on Summit (1)

- Conda is also available to CURC users.
- Customizability for your local R installations with minimal effort!
- Standard module unavailable at the moment. Instead use:

```
source /curc/sw/anaconda3/latest
```

- Once loaded a `(base)` prompt will be placed on your command line.
  - One additional step…

Conda on Summit (2)

- Conda will automatically cache downloads in your home directory.
- Run this command to push the cache to a larger space:

```
nano ~/.condarc
```

- This will open up the nano text editor. Simply paste the lines:

```
pkgs_dirs:
  - /projects/$USER/.conda_pkgs
envs_dirs:
  - /projects/$USER/software/anaconda/envs
```

Take aways

- Conda is an environment manager
- Software sometimes sucks
- Use Conda over Basic R to manage your R modules
- R can be used in conjunction with your R internal installs

Questions?

Thank you!

- Please fill out the survey: http://tinyurl.com/curc-survey18
- Contact information: rc-help@Colorado.edu
- Slides: *https://github.com/ResearchComputing/RInstalls_Fall_2021*
- Documentation:

  https://curc.readthedocs.io/en/latest/software/python.html