Ministry
of Justice

Coffee & Code: Conda

Calum Barnett

30 October 2019

What is Conda?

- Conda is an package, dependency and environment management system
- It is used to install packages and keep track of them in projects
- It is similar to packrat but is not specific to R

Why is Conda useful?

- Language agnostic
- Packages are pre-compiled
- Manages environments
- Facilitates reproducible analysis

When should I use Conda?

For projects that:

- involve multiple languages
- involve multiple people
- require reproducibility

How do I access Conda?

- Conda is available in RStudio to all users running **R-3.5.1** or later
- You interact with Conda using the command line
- Conda is not fully supported in JupyterLab

Basics

Basics

Check Conda is installed

```
$ conda --version
```

Basics

Get help

```
$ conda --help


# You can also use the “--help” tag with individual commands
```

Basics

List installed packages

```
# List all installed packages
$ conda list


# Search for a specific installed package
$ conda list | grep “pandas”
```

Basics

Search for packages

```
$ conda search r-dplyr


# R packages are prefixed with “r-”
# You can also search on anaconda.org


# Ex: search for a package using the command line
```

Basics

Install packages

```
# Install a package with default options

$ conda install r-dplyr


# Ex: install a package with default options
```

Basics

Install packages

```
# Install a specific version of a package
$ conda install pandas=0.24.0


# Ex: install a non-current version of a package
```

Basics

Install packages

```
# Install a package from a specific channel

$ conda install r-s3tools -c moj-analytical-services


# Ex: install a package from a specific channel
```

Basics

Uninstall packages

```
$ conda uninstall r-dplyr


# Ex: uninstall a package
```

Basics

Update packages

```
# Update a specific package
$ conda update r-dplyr


# Update all packages
$ conda update --update-all
```

Environment management

Environment management

Reset your environment

```
$ conda env export -n base | grep -v “^prefix: ” > /tmp/base.yml &&
  conda env update --prune -n rstudio -f /tmp/base.yml &&
  rm /tmp/base.yml


# Ex: reset your environment
```

Environment management

Fix r-pillar error

```
$ conda env export -n base | grep -v “^prefix: ” > /tmp/base.yml &&
  conda env update --prune -n rstudio -f /tmp/base.yml &&
  rm /tmp/base.yml
```

Environment management

Reset your environment

```
$ conda deactivate &&
  conda env remove -n rstudio &&
  rm -rf ~/.conda/envs/rstudio/ &&
  conda list --explicit -n root |
  grep -Ev "^@EXPLICIT|r-pillar" > /tmp/spec-file.txt &&
  conda env create -n rstudio -f /tmp/spec-file.txt &&
  conda activate rstudio


# Ex: reset your environment
```

Environment management

Export your environment

```
$ conda env export | grep -v “^prefix: ” > environment.yml


# Ex: install some packages and export your environment
```

Environment management

Load an environment

```
$ conda env update --prune -f environment.yml


# Ex: manually add a package to environment.yml

# Ex: load the environment

# Ex: verify the new package has been installed (conda list)
```

Next steps

Missing packages

Most of the main packages for R and Python are available through Conda. If a package is not available:

- install it using `pip` within a Conda environment (Python only)
- build and upload it to Anaconda yourself

Using Conda with apps

You can use Conda to install dependencies for apps:

- Store the `environment.yml` file in the GitHub repository
- Use the [`Dockerfile`]() from the [`conda`]() branch of the RShiny app template