**Creating, Managing, and Exporting Conda Environments**

Author: Noelle Kosarek

**What is a Conda Environment?**

A conda environment is an isolated, self-contained directory that holds specific versions of Python, R, or other languages along with specific versions of packages and dependencies. Conda environments allow you to manage and maintain separate working environments for different projects without interference between them. I like to think of conda environments as a room in a lab containing all of the instruments you need perfectly calibrated and ready to go. You can imagine a situation in which you need Special\_Package\_X version 1.1.1 for Project A, but then for Project B you need Special\_Package\_X version 1.1.2. You wouldn’t want to uninstall and reinstall those package versions when you need to switch between projects. This is precisely where conda environments are useful, among several other advantages:

1. Dependency management: Different projects may require different versions of packages or even different versions of R or Python. Conda environments allow you to maintain these separately without conflicts.
2. Reproducibility: As Data Scientists, we strive for consistency and reproducibility, using conda environments allows us to recreate the exact environment on another system, ensuring the same results. We can even trade environments with the greater scientific community on our GitHub.
3. Isolation: Each environment is independent, so changes in one do not affect others, preventing issues caused by package version mismatches.
4. Flexibility: Conda environments support packages from many different programming languages, making them useful for projects and workflows that require multiple languages.
5. Custom Environments: You can create custom environments tailored for specific tasks, not just individual projects.

**What will I need to install to create a Conda environment?**

To install Anaconda, follow this instructions at this [link](https://www.anaconda.com/docs/getting-started/anaconda/install). After installing Anaconda, navigate to your terminal.

**How Do I Create A Conda Environment?**

When you open a fresh Terminal you will see that the start of the prompt line begins with “(base)”. This means that your installation of conda was successful!

A screenshot of a computer

AI-generated content may be incorrect.

Now try typing in the following:

**conda –version**

This should return something like “conda 24.7.1”. This confirms that your conda installation worked and you are ready to create your first conda environment. To create your first conda environment enter this line of code:

**conda create -n my\_first\_env python=3.6.0**

Running this line of code will create a conda environment called “my\_first\_env” with python version 3.6.0 installed within it. Note, solving conda environments sometimes takes awhile, so don’t worry if it buffers for a bit, especially if you are installing Python and R along with a ton of other specific packages.

**How Do I Activate A Conda Environment?**

Once your conda environment has finished building, you are ready to activate! To activate this conda environment simply enter the following line:

**conda activate my\_first\_env**

You will notice that “(base)” at the beginning of the prompt line has been replaced with “(my\_first\_env)”. This means that you have properly activated your conda environment.

Now enter this line:

**python**

You will notice that Python has now been activated within your conda environment. Take a look at A computer code with white text

AI-generated content may be incorrect.the version of python, it should match the version we installed in the environment.

**How Do I Modify A Conda Environment?**

So far we have installed a conda environment with just Python version 3.6.0. Let’s add a python package called numpy to our environment. Numpy is a python package the provides support for large, multi-dimensional arrays and matrices. It also possesses a collection of math functions to conduct operations on these arrays and matrices. To install numpy in our environment type the following line while the environment is still activated:

**conda install numpy**

Note, if you wish to install a specific version of numpy you can add to the above command like so:

**conda install numpy=1.19.5**

So long as that command looks like it ran go ahead and type in “python” again. Once python pops up, type in the following line:

**import numpy as np**

If that line of code ran, then you successfully installed numpy!

Now that you know how to install one package, try using your new powers to install the python package “**pandas**”.

**How Can I Install Predefined Environments I Find on The Internet?**

Aside from funny videos of animals, the internet is full of wonderful conda environments, many of which live on people’s GitHub repositories. You can imagine a situation in which you find the perfect tool that will help you analyze your data on GitHub. Luckily, the author of that tool has deposited a “.yaml” file. Usually this file is called something like “environment.yaml”. You can download these files and install that conda environment right on your machine without having to wrestle with installing each package with the correct version arduously!

In this lesson I have added a dummy “evironment.yaml” file that installs R and ggplot2. Download that file to your Desktop, or somewhere on your local machine, and navigate to that location in your terminal (use your “cd” skills here!). Once there enter the line of code:

**conda env create --file environment.yaml**

Once this environment is done solving, type in the following:

**conda activate my\_first\_R\_env**

Once the environment is activate type in “R”. When R pops up, type in the following:

**library(dplyr)**

If that package loaded, then you successfully installed R and the package you wanted install from the environment file!

**What If I Want to Share The Environment I Built With My Friends?**

If you would like to share conda environments with your team or put them on your GitHub, you can simply run the following line in the environment you currently have activated:

**conda env export --file environment.yml**

This will generate a .yml file in your current working directory! Try this out with the “my\_first\_env” we created in this demo.