

Conda Command Prompt

Dr. Saad Laouadi

Econometrics and Data Science Academy

Outline

① Introduction

② conda info

conda info examples

③ conda version

④ conda install

conda install examples

⑤ conda update

conda update examples

⑥ conda create

conda create options

conda create examples

conda create python version

⑦ conda activate

⑧ conda deactivate

conda info -e

⑨ list command

conda list examples

⑩ conda remove

conda remove examples

⑪ conda search

conda search examples

⑫ conda clean

conda clean examples

⑬ conda config

conda config modifiers

conda config examples

Introduction

Conda is a powerful package manager and environment manager that you use with:

- ① *Command line commands at the Anaconda Prompt for Windows.*
- ② *In a terminal window for macOS or Linux.* ¹

¹source

Getting Conda Information

info command

- **info** command displays information about the current active environment, conda and python versions, channels ... etc.
- **info** command has several options:
 - **conda info -h** for the help page.
 - **conda info -a** or **--all**: displays information about everything.
 - **conda info --base**: displays the path of the base environment.
 - **conda info -e** or **--envs**: list of available conda environments.

Info Command Examples

- 1 Show information.

```
root$ conda info
```

- 2 Display the base path about

```
root$ conda info --base
```

- 3 Show information about everything

```
root$ conda info --all
```

Conda Version

- You can check which version of conda is installed on your system with the following command.

```
root$ conda -V
```

Or

```
root$ conda --version
```

Installing Packages with Conda

install command

- **install** command installs a package or a list of packages in a specified environment.
- **install** command has several options:
 - **conda install -h** for the help page.
 - **conda install -n or --name:** install packages in the specified environment. This has to be followed with the env name and the package or list of packages to be installed.
 - **conda install packname -y or --yes:** to install without asking for confirmation.

Installing Packages Examples

- 1 Install a package.

```
root$ conda install scipy
```

- 2 Install a list of packages without asking for confirmation

```
root$ conda install scipy numpy sympy --yes
```

- 3 Installing packages in a specified environment.

```
root$ conda install --name myenv numpy --yes
```


Update Conda

- Updating all the installed packages in the environment is needed from time to time.
- The **conda update** command has few options:
 - **conda update --all** or **conda update --update-all** which updates all the packages in the environment
 - **conda --update-all:** which does the same.
 - **conda update -h:** for help.
 - **conda update -n:** followed by the environment name you aim to update its packages.
 - **conda update --only-deps:** This updates only the dependencies.
 - **conda update -y** or **--yes:** This will update without asking for confirmation.

Here are the commands in the shell:

```
root$ conda update -h
```

Updating Packages Examples

- 1 Update all packages.

```
root$ conda update --all
```

- 2 Update a specific environment

```
root$ conda update -n myenv --all
```

- 3 Update dependencies without asking for confirmation

```
root$ conda update --only-deps --all -y
```

Creating Virtual Environments with Conda

create command

- **create** command allows you to create a new conda environment.
- You can specify a list of packages to be installed when creating the conda environment.
- Working with different environments gives you flexibility to work with different projects, and a specific environment for each project.
- Having a specific environment for each project reduces the risk of breaking some other project.
- When installing new packages, the author(s) may recommend installing them on a different environment.
- Try to build the habit of working with different environments.

Conda Create Options

The `conda create` command has several options, we mention the most common

- **`conda create -h`**: to show the help page of this command.
- **`conda create -n` or **`--name`****: this allows you to create a new environment, which must be followed with the name of the new environment.
- **`conda create -y` or **`--yes`****

Example Command of New Conda Envs

- Create a new conda environment called netenv.

```
root$ conda create -n netenv
```

- Create a new conda environment with a specific packages.

```
root$ conda create --name netenv tensorflow pandas
```

- Create a new conda environment with a specific packages without asking for confirmation (add --yes option).

```
root$ conda create --name netenv tensorflow pandas --yes
```

Create New Conda Environment with Specific python Version

- Create a new conda environment with python 3.9, proceed without asking for confirmation.

```
root$ conda create --yes --name netenv python=3.9
```

- Here is a complicated command: create a new env with python 3.9 from a file without asking for confirmation. ²

```
root$ conda create --file plist.txt --yes --name netenv python=3.9
```

²It's personal preference that I use --option like --name or --yes

Conda Environment Activation

- After creating a conda environment, you need to **activate** it in order to use it.
- Each time you log out of the command line interpreter, the environment will deactivate automatically, so if you need to use a specific environment for a specific project, you'll have to activate that again.
- To activate a conda environment use the following syntax:

```
root$ conda activate envname
```

Conda Environment Deactivation

- Usually, you will have different projects and different conda environments.
- You may need to switch from an environment to another or go back to base one. Thus, you need to deactivate the active environment.
- To deactivate a conda environment, run the following command:

```
root$ deactivate
```

Yes just deactivate!

Listing Conda Environments

- Creating conda environments with meaningful names will be useful in the long run.
 - What if you don't remember the name of a specific environment or how many of them are they?
 - Getting information about all available conda environments will be more than helpful.
- Hence the following command `conda info -e` or `conda info --envs`:

```
root$ conda info -e
```

Listing

list command

- **list** command list the installed packages in the active or specified environment.
- the main command is **conda list** without any arguments lists all installed packages.
- **list** command has several options:
 - **conda list -h** for the help page.
 - **conda list -n or --name:** plus the name of the environment will list all the installed packages in that environment.
 - **conda list -e or --export:** this will export the installed package to a text file which can be used to create a conda environment using the option **conda create --file**

Listing Packages Examples

- 1 List all the packages in the current environment.

```
root$ conda list
```

- 2 list all packages in the specified environment.

```
root$ conda list -n envname
```

- 3 Export the installed packages in the current environment.

```
root$ conda list --export > packlist.txt
```

- 4 Create a new environment using the exported file.

```
root$ conda create --name envname --file packlist.txt
```

Removing packages

remove command

- **remove** command removes a package or a list of packages in a specified environment.
- **remove** command has several options:
 - **conda remove -h** for the help page.
 - **conda remove -n or --name:** removes packages in the specified environment. This has to be followed with the env name and the package or list of packages to be removed.
 - **conda remove packname-y or --yes:** to remove packages without asking for confirmation.

Remove Command Examples

- 1 remove a package

```
root$ conda remove scipy
```

- 2 Remove a list of packages without asking for confirmation

```
root$ conda remove scipy numpy sympy --yes
```

- 3 Removing an entire environment. You should pass the option **--all**.

```
root$ conda remove --name myenv --all --yes
```

Search for Packages

Search Command

- **search** command searches for packages and reports the associated information when found.
- **search** command has several options:
 - **conda search -h** for the help page.
 - **conda search packname --envs**: searches for the specified package in the current environment (or all environments in the system when using the admin mode).
 - **conda search packname -i or --info**: searches and reports all the information about the package.

Listing Packages Examples

- 1 Search for a specific package

```
root$ conda search numpy
```

- 2 Search for package in the current environment.

```
root$ conda search numpy --envs
```

- 3 Search for a package in environment and report all information about it.

```
root$ conda search numpy --envs --info
```

Cleaning Packages and Caches

Clean Command

- **clean** removes unused packages and caches.
- **clean** command has several options:
 - **conda clean -h** for the help page.
 - **conda clean -a --all**: Remove index cache, lock files, unused cache packages, and tarballs.
 - **conda clean -t or --tarballs**: Remove cached package tarballs.
 - **conda clean -l or --logfiles**: Remove logfiles.
 - **conda clean -y or --yes** add this option with another option for stopping asking for confirmation.

Cleaning Packages Examples

- 1 Clean everything

```
root$ conda clean --all
```

- 2 Clean logfiles without asking for confirmation

```
root$ conda clean -logfiles --yes
```

- 3 Clean index Caches and tarballs

```
root$ conda clean --index-caches  
root$ conda clean --tarballs
```

Conda Configuration

Config Command

- **conda config** this command is intended for programmer and not for novices.
- **conda config** modifies the configuration stored in the file **.condarc**
- **conda config** has several options:
 - **conda config -h** for the help page.
 - **conda config --show**: Shows information for all configuration values.
 - **conda config--show-sources**: Displays all identified configuration sources..
 - **conda config --describe**: Describes given configuration parameters. If no arguments given, show information for all configuration parameters.

Conda Config Modifiers

- **conda config --get** get a configuration value.
- **conda config --append** Add one configuration value to the end of a list key.
- **conda config --prepend or --add** Add one configuration value to the beginning of a list key.
- **conda config --set:** Set a boolean or a string key.
- **conda config --remove:** Remove a configuration value from a list key. This removes all instances of the value.
- **conda config --remove-key:** Remove a configuration key (and all its values)..

Config Examples

- 1 Show all configuration information.

```
root$ conda config --show
```

- 2 Display all identified configuration sources:

```
root$ conda config --show-sources
```

- 3 Describe all available configuration options:

```
root$ conda config --describe
```

- 4 Add the conda-canary channel:

```
root$ conda config --add channels conda-canary
```

Copyright Information

This slide show was prepared by **Dr. Saad Laouadi**. Independent Researcher . It is licensed under a **Creative Commons Attribution-ShareAlike 4.0 International License**. Use any part of it as you like and share the result freely.