Code - Setup

Mark Asch - ML-PREP

2025

Program

Setup the code environment for the Basic and Advanced Courses.

1. Python (conda),
   (a) packages for ML
   (b) packages for GIS
2. Seismic wave propagation
3. R and RStudio (optional)
4. Octave (optional)

PRINCIPLES

The 3 Principles of Coding

1. Coding is Understanding

   (a) if you don't code, you don't/won't understand (coding helps to understand, good code enables good understanding)
   (b) if you understand, then you can code (coding requires perfect undertanding)

2. Coding is Learning

   (a) if you learn well, then you can code well
   (b) if you code well, then you'll learn (from/with the code)

3. Coding is Tennis

   (a) you cannot code by just watching
   (b) if you code, then you're a player (in the game of life/science/research/industry/etc.), otherwise you will remain a spectator...

3 Principles of Coding

- Coding is Understanding
- Coding is Learning
- Coding is Tennis

PYTHON

Basic Python Installation

- We recommend the use of the conda environment.
- If not already installed on your laptop, please download a version (mini or full) from the conda website—see the links on the CSU course software webpage.

Conda is an open-source package management system and environment management system that runs on Windows, macOS, and Linux. Conda quickly installs, runs, and updates packages and their dependencies. Conda easily creates, saves, loads, and switches between environments on your local computer.

Python/Conda Environments - Machine Learning

Note

For the duration of the training, we will require a number of specific python packages. For this, we will create and use tailor-made conda environments.

```
conda create -n prep_ml python=3
conda activate prep_ml
conda install jupyterlab numpy matplotlib pandas
conda install scikit-learn
conda install pytorch torchvision -c pytorch
pip install islp
.
.
jupyter notebook
.
.
conda deactivate
conda env list
```

Test Python Installation

Note

Before starting any of the labs, we must test our python environment.

- Launch a jupyter notebook.
- Test pytorch:

```
import torch
x = torch.rand(5, 3)
print(x)
```

- Test sklearn:

```
import sklearn
print(sklearn.__version__)
```

Python/Conda Environments - GIS

Note

For the duration of the training, we will require a number of specific python packages. For this, we will create and use tailor-made conda environments.

```
conda create -n prep_gis python=3
conda activate prep_gis
conda install --file requirements.txt
.
.
jupyter notebook
.
.
conda deactivate
conda env list
```

Writing python code

- Here, we will use Jupyter Notebooks, which are very good for communicating results and research.

- Real coders, however, use far more sophisticated code editing applications, such as:

  ⇒ Visual Studio - developed by Microsoft, but freely available.
  ⇒ PyCharm - for professional developers.
  ⇒ Spyder - comes inside the anaconda bundle.

- Note that chatGPT and other generative transformers, can be used as a coding assistant - but beware, the code generated should be thoroughly debugged and tested before use! There are very often serious errors and bugs, except in the simplest cases...

Wave Propagation

Software Environemt

For introductory training, we will require:

- python libraries for numerical analysis and plotting
- spectral element codes for 2D and 3D seismic wave propagation

Note

All details for code installation are provided on the training github, in the seismic wave propagation section.

R

R and Rstudio

- Follow instructions on the CSU course software webpage to:

1. Download and install R.

2. Then, download and install Rstudio—please respect the order.

Rstudio Notebooks

- It is highly recommended (reproducible research) to use the notebook capabilities of Rstudio
- Note that slide presentations can equallly be produced, as can markdown output for GitHub
- Please follow the instructions and examples in the documents:

  ⇒ `learn_NB_EDA.pdf`

- **Note**: on the first launch of a notebook, Rstudio will need to automatically download numerous packages for the typesetting. Please make sure that this process terminates successfully!

OCTAVE

Octave

Note

- Octave is a free, open-source version of Matlab, containing all its basic functionality except some specialized toolboxes.
- Octave can readily be integrated as a Jupyter kernel, thus enabling notebook output.

1. Download Octave from the GNU site: `https://octave.org/`
2. (Optional) Install the Jupyter kernel and update the path to the Octave executable—please consult the course software webpage for this.

References

1. Please consult the list provided on the website: CODE REFERENCES