Getting Started

This document outlines steps for environment configuration on Windows and MacOS computers to run the exercises developed in the workshop repository.

1. Install Anaconda / Miniconda
2. Build the Conda Environment # for Jupyter Notebook exercises No. 1-6.
3. Install VSCode # for Jupyter Notebook; Notebooks may also be run in a browser.
4. Install QGIS # for review of generated datasets.
5. Install GRASS GIS # for Hydro Flowline Jupyter Notebook exercise No. 7.

Windows

Conda

- See Installing Miniconda — Anaconda documentation
- Start “Anaconda Prompt (miniconda3)” from the Windows Start Menu.
- Install git from the Anaconda channel.

```
conda install -c anaconda git
```

- Update the base environment from the default channel.

```
conda update -n base -c defaults conda
```

- Print the conda channels.

```
conda config --show channels
```

- Add the conda-forge channels.

```
conda config --append channels conda-forge
```

Build the Conda Workshop Environment on Windows

Download workshop material

Download workshop (https://github.com/rdzur/ncsu-workshop) materials, e.g., click the 

button to download a zip of the repository or copy the link to clone the repository.

- At the Anaconda command prompt, change directories to the location of extracted workshop material, for example:

```
cd C:\Temp\ncsu-workshop-main\
```

- An environment.yml configuration file is included to support environment creation.

- Run the following command to create a default environment called “ncsu-workshop” including the key packages: RVT and Ultralytics.

```
conda env create --file environment.yml
```

- The environment name can be overridden with the -n option as noted below for example with “tem-env” or whatever name may be preferred.

```
conda env create --file environment.yml -n tem-env
```

- Activate the new environment with the following command and continue with

```
conda activate ncsu-workshop
```

Configure the Environment

- On Windows, current Ultralytics installation requires downgrading to the previous PyTorch version 2.3.1 https://pytorch.org/get-started/previous-versions/

- Check the PyTorch versions within the environment.

```
conda list -n tem-env | findstr torch
```

- Uninstall the PyTorch versions within the environment.

```
pip uninstall torch
```

```
pip uninstall torchvision
```

CPU or GPU?

To confirm whether the computer has a discrete NVIDIA GPU, run the Nvidia System Management Interface (SMI) command. It returns card details and CUDA (Compute Unified Device Architecture) version.

```
nvidia-smi
```

Install CPU only PyTorch version for Ultralytics

- Next, if the computer only has CPU and no discrete NVIDIA graphics card, install the previous 2.3.1 PyTorch CPU version for Ultralytics to function on Windows.

```
pip install torch==2.3.1 torchvision==0.18.1 torchaudio==2.3.1 --index-url
https://download.pytorch.org/whl/cpu
```

Install GPU CUDA PyTorch version for Ultralytics

- If the computer is equipped with a discrete NVIDIA graphics card and supports the required architecture, install the previous 2.3.1 PyTorch 12.1 CUDA version for Ultralytics to function on Windows.

```
pip install torch==2.3.1 torchvision==0.18.1 torchaudio==2.3.1 --index-url
https://download.pytorch.org/whl/cu121
```

Check the Environment

When the environment install is complete, at the command prompt the environment can be evaluated by entering the Python prompt and attempting to load, for example, the libraries the following commands.

```
python

>>> import rvt.default

>>> import ultralytics

>>> exit()
```

The YOLO environment can also be checked at the command line interface (CLI). The following command returns YOLO syntax and demonstrates the installation is complete.

```
yolo
```

To check YOLO configuration CPU or Nvidia GPU, run the following which will display CPU, GPU and CUDA version.

```
yolo checks
```

Jupyter can be started by either typing the jupyter notebook or jupyter lab command to launch a local Jupyter server for running notebooks via web browser. Alternatively, notebooks may be run via Microsoft VS Code (Visual Studio Code) or another IDE (integrated development environment) as described in the section below.

```
jupyter notebook
```

VS Code

Once installed, VS Code can be started at the terminal or via the Windows Start menu.

```
code
```

Open the Workshop in VS Code; Install Extensions

In VS Code, use Explorer  to open the workshop folder. 

Selecting the first (01…) notebook in the explorer pane will display the notebook in the editor pane where at the upper right area use the select kernel button to use the Jupyter Kernel created in the Conda sections of this document. VS Code may prompt to install the Python and Jupyter extensions. Install those extensions. If not prompted to install these extensions, they may be found by searching through the Extension  panel.

Delete the .gitignore files

Remove .gitignore files from OPR, SVF, SVF_2x2_Multiprocess and LPC folders.

QGIS

Download and install QGIS. QGIS is only used to quickly review generated datasets.

Optional: Install the QuickMapServices QGIS Plugin

The QuickMapServices QGIS Plugin provides access to web map services such as OpenStreetMap and others added after plugin installation via Web > QuickMapServices > 

. Use the “More services”, “Get contributed pack” button.

QuickMapServices Plugin is not required to run any of the notebooks.

GRASS GIS

Download and install GRASS GIS.

GRASS GIS and Jupyter Integration

GRASS GIS is integrated with Jupyter notebooks and used for notebook exercise No. 7. Once GRASS GIS is launched, the GRASS Jupyter environment can be started from the GRASS Command Prompt with the `jupyter notebook` or `jupyter lab` command. When Jupyter starts up, local server URLs are printed in the terminal, and these URL can also be used as a VS Code kernel to run notebook No. 7 within VS Code.

MacOS

Conda

- Install Anaconda from Homebrew (https://brew.sh). This site recommends the following one-line install method.

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

- When brew finishes it prompts the next steps to add brew to the PATH, which may depend on the default shell. The example below is relevant to the bash shell. Modify the command below to substitute <account> with applicable username. Use the whoami command at the terminal to confirm username.

```
/bin/bash -c  (echo; echo 'eval "$(/opt/homebrew/bin/brew shellenv)"') >>
/Users/<account>/.bash_profile
```

```
eval "$(/opt/homebrew/bin/brew shellenv)"
```

- At this point, brew commands such as following should be operable at the terminal:

```
brew info
```

```
brew doctor
```

- Install anaconda and wget (for data download) with brew.

```
brew install --cask anaconda
```

```
brew install wget
```

- If zsh is the default shell, the following commands will be required to add anaconda to the PATH

```
echo 'export PATH="/opt/homebrew/anaconda3/bin:$PATH"' >> ~/.zshrc
```

```
source ~/.zshrc
```

- If the default shell is bash then modify the PATH to .bash_profile.

```
echo 'export PATH="/opt/homebrew/anaconda3/bin:$PATH"' >> ~/.bash_profile
```

```
source ~/.bash_profile
```

- Initialize Anaconda

```
conda init
```

- In zsh, possibly run the following to initialize Anaconda:

```
conda init zsh
```

- Close the terminal (shell) and reopen to see the base environment activated.

Build the Conda Workshop Environment on MacOS

Use the terminal prompt on macOS to set up a new environment for YOLO and RVT with associated dependencies. Then activate the environment.

- Navigate to github repository with yaml and click 

and copy link for git clone or download the zip.

- Git clone, following example below and modify <account> to appropriate username:

```
cd /Users/<account>/Documents
```

```
git clone https://github.com/rdzur/ncsu-workshop.git
```

- Alternatively, extract the downloaded zip folder and data and change directories to the folder location, for example to:

```
cd /Users/<account>/Documents/ncsu-workshop-main
```

- It may be useful to check the ownership of the ~/.conda directory and set the ownership to the current <account>.

```
sudo chown -R <account> ~/.conda
```

- An environment file (environment.yml) can be used to create the necessary environment packages. In the example below default name in environment.yml is called “ncsu-workshop” and can be changed to whatever name may be desired by overriding the name with the -n option.
- Edit the yml file with a text editor (or VS Code) to comment out the lines corresponding to specific libraries for the Windows installation by adding # at the beginning of the relevant lines:

```
# - m2-base
```

- Create the conda environment with the environment.yml file.

```
conda env create --file environment.yml -y
```

```
conda env create -n "any-name" --file environment.yml -y
```

- Activate the environment.

```
conda activate ncsu-workshop
```

Check the Environment

The environment may be evaluated by loading YOLO and RVT libraries in python.

```
python

>>> import rvt.default

>>> import ultralytics

>>> exit()
```

Notebooks on MacOS

The workshop notebooks can run as described in the sections above for VS Code (notebooks 01 through 06) and GRASS GIS (notebook 07). With data review available with QGIS. It is also important on MacOS to delete the .gitignore files as described above. Note: On MacOS .DS_Store files may also interfere if present in those folders.