

Getting Started: Python Time Series (with Jupyter & Conda)

A quick guide for beginners to run notebooks from the Python_for_Time_Series repository

Updated: 2025-10-31

1) Install & Verify

- Install Anaconda (recommended) from <https://www.anaconda.com/download>. Choose Python 3.12 (or newer).
- Open **Anaconda Prompt** (Windows) or **Terminal** (macOS/Linux).
- Verify Python & conda:

```
python --version  
conda --version
```

2) Get the Repository

- Option A (recommended): Clone with Git

```
git clone https://github.com/sulabola/Python_for_Time_Series.git  
cd Python_for_Time_Series
```
- Option B: Download ZIP from GitHub → Extract → cd into the extracted folder.

3) Create a Clean Environment (avoid breaking base)

Create a conda env with core packages (one-time):

```
conda create -n ts-env python=3.12 pandas numpy matplotlib scikit-learn jupyter -c  
conda-forge
```

Activate it:

```
conda activate ts-env
```

(Optional) Add the kernel to Jupyter so you can select it in notebooks:

```
python -m ipykernel install --user --name ts-env --display-name "Python (ts-env)"
```

*** Note: "ts-env" can be changed to any preferred name.

4) Launch Jupyter in Your Working Folder

If you keep notebooks in Downloads for now:

```
cd %USERPROFILE%\Downloads (Windows)
```

Or go to the repo folder, e.g.:

```
cd path\to\Python_for_Time_Series
```

Start Jupyter:

```
jupyter notebook
```

Then open the notebook you want. In Jupyter: Kernel → Change kernel → "Python (ts-env)".

5) If a Package is Missing (e.g., statsmodels)

Inside the activated environment (preferred):

```
conda install -c conda-forge statsmodels
```

Or, from within a notebook cell (same kernel):

```
!pip install statsmodels
```

Then: Kernel → Restart, and re-run the import.

6) Common Issues & Fixes

- ModuleNotFoundError: Make sure the notebook kernel matches your env: Kernel → Change kernel → Python (ts-env).

- pip vs conda: In conda envs, prefer conda for scientific packages. If you use pip, run:

```
python -m pip install <package>
```

- Jupyter not seeing your env: install an ipykernel (see step 3).

- Paths: Confirm which Python is used:

```
import sys; print(sys.executable)
```

7) Updating Packages (only when needed)

Safer via conda (inside ts-env):

```
conda update pandas numpy matplotlib scikit-learn jupyter
```

You can also add other time-series tools when needed:

```
conda install -c conda-forge statsmodels pmdarima arch prophet
```

8) Reproducibility (recommended)

Export your exact environment to share or re-create later:

```
conda env export --no-builds > environment.yml
```

Re-create on a new machine:

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

9) Suggested Project Structure (when you move out of Downloads)

Create a stable folder per project, e.g.:

```
C:\Users\<you>\Projects\TimeSeries\
```

Keep data in a 'data/' subfolder, notebooks in 'notebooks/', and scripts in 'src/'.

Start Jupyter from the project root so relative paths are clean.

Quick Checklist (Each Session)

1. Open Anaconda Prompt
2. conda activate ts-env
3. cd to your working folder (repo or Downloads)
4. jupyter notebook
5. Pick kernel: Python (ts-env)