Setup Guide: VS Code + Stata Kernel

Jesús Villota Miranda

Methods for Time Series
CEMFI

Introduction

Welcome to the setup guide for the "Methods for Time Series" course at CEMFI! This guide will help you set up everything you need to run Stata code in Jupyter notebooks using Visual Studio Code (VS Code) as your code editor.

This setup allows you to combine the powerful statistical capabilities of Stata with the interactive notebook environment of Jupyter, making it easier to learn and practice time series analysis techniques.

What you'll get:

- VS Code installed and configured for data science
- Python environment with Jupyter and required packages
- Stata kernel for Jupyter notebooks
- Ability to run Stata code in interactive notebooks

We'll cover the steps for Windows, macOS, and Linux. Follow the sections for your operating system (OS). If you're unsure what OS you have, check your computer's settings.

Step 1: Install Stata

Stata is the statistical software that will perform the actual time series analysis. You need to have Stata installed before setting up the Jupyter kernel.

For Windows:

1. Go to `https://www.stata.com` and click on "Buy" or "Contact Sales" to purchase or get access to Stata.
2. Download the Windows installer for your version (StataMP, StataSE, or StataIC).
3. Run the downloaded installer and follow the on-screen instructions.
4. Note the installation directory (typically `C:\Program Files\Stata17\` or similar).
5. Launch Stata once to complete the license activation process.

For macOS:

1. Go to `https://www.stata.com` and click on "Buy" or "Contact Sales" to purchase or get access to Stata.
2. Download the macOS installer for your version (StataMP, StataSE, or StataIC).
3. Open the downloaded `.dmg` file and drag Stata to your Applications folder.
4. Launch Stata from Applications to complete the license activation.
5. Note the installation path: `/Applications/Stata/StataMP.app` (or StataSE.app/StataIC.app).

For Linux:

1. Go to `https://www.stata.com` and click on "Buy" or "Contact Sales" to purchase or get access to Stata.
2. Download the Linux installer for your architecture (32-bit or 64-bit).
3. Extract the downloaded file: `tar -xzf stata17-linux64.tar.gz`
4. Move to desired location: `sudo mv stata17 /usr/local/`
5. Set permissions: `sudo chown -R $(whoami) /usr/local/stata17`
6. Launch Stata to complete license activation: `/usr/local/stata17/xstata`

Verify Stata Installation

To verify your Stata installation, try running a simple command:

```
# On Windows (in Command Prompt):
"C:\Program Files\Stata17\StataMP.exe" -e "display \"Hello from Stata\""

# On macOS (in Terminal):
/Applications/Stata/StataMP.app/Contents/MacOS/stata-mp -e "display \"Hello from Stata\""

# On Linux (in Terminal):
/usr/local/stata17/stata-mp -e "display \"Hello from Stata\""
```

You should see Stata start and display the message.

Step 2: Install VS Code

VS Code is a free code editor where you'll write and run your Jupyter notebooks.

1. Go to the official website: `https://code.visualstudio.com/download`.
2. Download the installer for your OS (it will auto-detect, but select Windows, Mac, or Linux if needed).
3. Run the downloaded file and follow the on-screen instructions to install. Accept the defaults if you're unsure.
4. Once installed, open VS Code to make sure it launches. It should look like a blank window with a menu bar at the top.

Step 3: Install Python and UV

We'll use UV to manage Python packages and create an isolated environment for our project.

For Windows:

1. Open PowerShell (search for "PowerShell" in your start menu and run it as administrator by right-clicking > Run as administrator).
2. Copy and paste this command, then press Enter:

   ```
   powershell -ExecutionPolicy ByPass -c "irm https://astral.sh/uv/install.ps1 | iex"
   ```

3. Follow any prompts. It may ask you to restart PowerShell or your computer.
4. To verify: Reopen PowerShell and run `uv --version`. You should see a version number.

For macOS:

1. Open Terminal (press Command + Space, type "Terminal", and open it).
2. Copy and paste this command, then press Enter:

   ```
   curl -LsSf https://astral.sh/uv/install.sh | sh
   ```

3. If you don't have curl, use: `wget -qO- https://astral.sh/uv/install.sh | sh` (but curl is usually pre-installed).
4. Follow any prompts. It may add UV to your PATH.
5. To verify: In Terminal, run `uv --version`. You should see a version number.

For Linux:

1. Open your Terminal (search for "Terminal" in your applications menu).
2. Copy and paste this command, then press Enter:

   ```
   curl -LsSf https://astral.sh/uv/install.sh | sh
   ```

3. If you don't have curl, install it first (e.g., on Ubuntu: `sudo apt update && sudo apt install curl`).
4. Follow any prompts.
5. To verify: In Terminal, run `uv --version`. You should see a version number.

Install Python

Now that UV is installed, use it to download and install Python:

1. Run this command:

   ```
   uv python install 3.11
   ```

2. UV will download and install Python automatically (it may take a few minutes depending on your internet).
3. To verify: Run `uv python list`. You should see Python 3.11 listed as installed.

Step 4: Set Up the Project Environment

Download Course Materials

First, you need to download the course materials:

1. Go to `https://github.com/jesusvillotamiranda/TA_Time_Series_Methods`
2. Click the green "Code" button and select "Download ZIP"
3. Extract the ZIP file to a folder on your computer (e.g., `TA_Time_Series_Methods`)

Initialize Python Environment

1. Open VS Code
2. Go to File > Open Folder... and select the extracted `TA_Time_Series_Methods` folder
3. Open a terminal in VS Code: Terminal > New Terminal
4. In the terminal, navigate to the project folder (it should already be there)
5. Run this command to install all required packages:

   ```
   uv sync
   ```

   This will create a virtual environment and install all dependencies listed in `pyproject.toml`.
6. Activate the virtual environment:

   ```
   # On Windows:
   .venv\Scripts\activate

   # On macOS/Linux:
   source .venv/bin/activate
   ```

Step 5: Install and Configure Stata Kernel

Now we'll install the Stata kernel that allows Jupyter to communicate with Stata.

1. Make sure you're in the project directory and the virtual environment is activated
2. Run this command to install the Stata kernel:

   ```
   uv run python -m stata_kernel.install
   ```

3. The installer will prompt you for the Stata executable path. Enter the path based on your OS and Stata version:

   Windows examples:

   - StataMP: `C:\Program Files\Stata17\StataMP.exe`
   - StataSE: `C:\Program Files\Stata17\StataSE.exe`
   - StataIC: `C:\Program Files\Stata17\StataIC.exe`

macOS examples:

- StataMP: `/Applications/Stata/StataMP.app/Contents/MacOS/stata-mp`
- StataSE: `/Applications/Stata/StataSE.app/Contents/MacOS/stata-se`
- StataIC: `/Applications/Stata/StataIC.app/Contents/MacOS/stata-ic`

Linux examples:

- StataMP: `/usr/local/stata17/stata-mp`
- StataSE: `/usr/local/stata17/stata-se`
- StataIC: `/usr/local/stata17/stata`

4. The installer will test the connection and confirm successful setup.

Verify Kernel Installation

To verify the Stata kernel is properly installed:

```
jupyter kernelspec list
```

You should see `stata` in the list of available kernels.

Step 6: Set Up VS Code for Jupyter

1. Install the Python extension for VS Code:
   - Click the Extensions icon on the left sidebar (it looks like four squares)
   - Search for "Python" (by Microsoft)
   - Click Install on the top result
2. Install the Jupyter extension:
   - In the Extensions sidebar, search for "Jupyter"
   - Install the "Jupyter" extension by Microsoft
3. Configure Python interpreter:
   - At the bottom left of VS Code, click on the Python version (it might say "Select Interpreter" if none is chosen)
   - Search for the one in your project's `.venv` folder
   - Select the interpreter that shows your project path

Step 7: Test the Setup

Let's create a test notebook to verify everything works:

1. In VS Code, create a new file: File > New File
2. Save it as `test_stata.ipynb`
3. VS Code should automatically open it in Jupyter notebook format
4. Click on "Select Kernel" in the top right of the notebook
5. Choose "Stata" from the list of available kernels

6. Create a new cell and enter this Stata code:

```
clear
set obs 100
gen x = rnormal()
gen y = 2 + 0.5*x + rnormal()
reg y x

```

7. Run the cell (Shift + Enter or the run button)
8. You should see Stata output with the regression results

Step 8: Open Course Notebooks

Now you can open the actual course notebooks:

1. Navigate to `sessions/session_1/` in the file explorer
2. Open any `.ipynb` file (e.g., `01_autocorrelation_correlation.ipynb`)
3. Make sure the kernel is set to "Stata" (top right corner)
4. You can now run the cells and follow along with the course material

Troubleshooting

Common Issues and Solutions

Problem: "stata_kernel not found" or import errors
Solution:

```
# Make sure you're in the project directory
cd TA_Time_Series_Methods

# Reinstall dependencies
uv sync

# Reinstall the kernel
uv run python -m stata_kernel.install
```

Problem: Stata kernel doesn't appear in VS Code
Solution:

```
# Check if kernel is installed
jupyter kernelspec list

# If not listed, reinstall
uv run python -m stata_kernel.install

# Restart VS Code
```

Problem: "Stata executable not found" errors
Solution: Verify the Stata executable path you provided during installation:

- Test the path manually in terminal/command prompt
- Make sure Stata launches with the exact path you provided

- Reinstall the kernel with the correct path

Problem: Kernel dies or notebook crashes
Solution:

- Check that your Stata license is valid and activated
- Verify Stata can run independently (not through Jupyter)
- Try restarting VS Code and the notebook
- Check that you have enough system memory

Problem: Can't find Stata executable
Solution: Find your Stata installation:

```
# On Windows (in PowerShell):
Get-ChildItem -Recurse -Path "C:\Program Files\Stata*" -Include "Stata*.exe"

# On macOS (in Terminal):
find /Applications -name "stata*" -type f 2>/dev/null

# On Linux (in Terminal):
find /usr/local -name "stata*" -type f 2>/dev/null
```

Operating System Specific Issues

Windows:

- Make sure you run PowerShell as administrator for UV installation
- If you get permission errors, try running VS Code as administrator
- Check that your antivirus isn't blocking the Stata executable

macOS:

- If you get "command not found" for stata-mp, make sure Stata is in /Applications/Stata/
- You might need to allow the Stata app in Security & Privacy settings
- Try launching Stata from Applications first to complete any setup

Linux:

- Make sure the Stata executable has proper permissions: `chmod +x /path/to/stata`
- You might need to install additional libraries: `sudo apt install libxt6 libxmu6`
- Check that your display manager supports X11 if using GUI Stata

Quick Reference Commands

Here's a summary of the key commands you'll need:

```
# Navigate to project
cd /path/to/TA_Time_Series_Methods

# Install/update packages
uv sync
```

```
# Activate virtual environment
# Windows: .venv\Scripts\activate
# macOS/Linux: source .venv/bin/activate

# Install Stata kernel
uv run python -m stata_kernel.install

# List available kernels
jupyter kernelspec list

# Launch Jupyter Lab (alternative to VS Code)
uv run jupyter lab
```