

Data Science & AI for Economists

Lecture 5: Introduction to VS Code

Zhaopeng Qu

Business School, Nanjing University

October 08 2025



Roadmap

Today's Agenda

- 1. Why VS Code for Data Science?
 - From RStudio to VS Code
 - One Tool for Everything
- 2. Installing and Setting Up VS Code
 - Download and Installation
 - First Launch and Interface Tour
 - Essential Settings Configuration

Today's Agenda

- 3. Essential Extensions for Data Science
 - R Language Support
 - Git and GitHub Integration
 - Markdown and Documentation Tools
- 5. Copilot and AI Features
- 4. Using VS Code with R and GitHub
 - Running R Code in VS Code
 - Git Workflow Integration
 - Integrated Terminal Usage

Why VS Code for Data Science?

From RStudio to VS Code

Traditional Workflow

- RStudio for R programming
- Jupyter Notebooks for Python
- Stata do-file editor for Stata
- Separate text editor for other files
- GitHub Desktop or command line for Git

Problem

- Switching between multiple applications.
- Different keyboard shortcuts and interfaces.
- Harder to manage complex projects with multiple languages.
- AI assistants (e.g., Copilot) not integrated.

VS Code: One Tool for Everything

What is VS Code?

- Visual Studio Code is a free, open-source code editor developed by Microsoft
- Available for Windows, Mac, and Linux
- Lightweight but powerful
- Developed by Microsoft
- Actively maintained and updated

VS Code: One Tool for Everything

1. Multi-language Support

- R, Python, Stata, Julia, SQL...
- HTML, CSS,
- And many more...

2. Integrated Git/GitHub

- Built-in version control
- Visual diff tools
- Easy push/pull operations

3. Extensible

- Thousands of extensions
- Customize to your needs
- Active community

4. AI Coding Assistants

- Many AI tools available (e.g., GitHub Copilot)
- Helps write code faster and better
- Debugging support

A Typical Research Project Structure

```
research-project/  
├─ data/                # Raw and processed data  
│   ├─ raw/  
│   └─ processed/  
├─ scripts/            # R/Python/Stata scripts  
│   ├─ 01_cleaning.R  
│   ├─ 02_analysis.do  
│   └─ 03_visualization.py  
├─ outputs/            # Figures and tables  
│   ├─ figures/  
│   └─ tables/  
├─ paper/              # Manuscript files  
│   ├─ manuscript.md  
│   └─ references.bib  
└─ README.md           # Project documentation
```

- **VS Code** can handle all these files in one workspace! You can write R code, Python code, edit Markdown, manage Git, and run terminal commands without leaving VS Code.
 - It is especially useful for **AI-assisted coding** with GitHub Copilot(next week!).

Installing and Setting Up VS Code

Download and Installation

Step 1: Download VS Code

- Visit: <https://code.visualstudio.com/>
- Click the download button for your operating system
 - Windows: `.exe` installer
 - Mac: `.dmg` file
 - Linux: `.deb` or `.rpm` package

Download and Installation

Step 2: Install

- Run the installer
- Accept the license agreement
- Choose installation location (default is fine)
- **Important:** Check "Add to PATH" option
- Complete the installation

First Launch and Interface Tour

Main Components

1. Activity Bar (Left)

- Explorer: File navigation
- Search: Find in files
- Source Control: Git integration
- Extensions: Install add-ons
- (More icons as you install extensions)

2. Side Bar

- Shows content based on Activity Bar selection
- File tree when Explorer is selected
- Git changes when Source Control is selected

First Launch and Interface Tour

3. Editor Group

- Main area where you write code
- Can split into multiple panes
- Tabs for different files
- Syntax highlighting

5. Status Bar (Bottom)

- Current file info
- Git branch
- Line/column number
- Language mode

4. Panel (Bottom)

- Terminal: Run commands
- Output: See program output
- Problems: View errors and warnings
- Debug Console: For debugging

Essential Settings Configuration

Accessing Settings

- Method 1: File → Preferences → Settings
- Method 2: Keyboard shortcut
 - Windows/Linux: Ctrl + ,
 - Mac: Cmd + ,
- Method 3: Command Palette
 - Windows/Linux: Ctrl + Shift + P
 - Mac: Cmd + Shift + P
 - Type Preferences: Open Settings

Two Views

- UI View: User-friendly interface

Essential Settings Configuration

Recommended Settings for Students

1. Auto Save

Files: Auto Save → afterDelay

Files: Auto Save Delay → 1000

- Prevents losing work!
- Saves automatically after typing

2. Font and Display

Editor: Font Size → 14-16

Editor: Word Wrap → on

Editor: Line Numbers → on

- Better readability
- Easier to follow in class

3. Terminal Settings

Terminal > Integrated > Default Profile

Windows: Git Bash or PowerShell

Mac/Linux: bash or zsh

- Use the terminal you're comfortable with

4. Git Configuration

Git: Enabled → ✓

Git: Auto Fetch → ✓

Git: Confirm Sync → false

- Enable Git integration
- Auto-fetch updates from remote

Essential Settings Configuration

5. Format on Save

Editor: Format On Save → ✓

- Automatically format code when saving
- Keeps code clean and consistent

6. Tab Settings

Editor: Tab Size → 2 (for R)

Editor: Insert Spaces → ✓

- Consistent indentation
- Important for R code

7. Minimap

Editor: Minimap: Enabled → ✓

- Shows code overview on the right
- Helps navigate long files

8. Bracket Pair Colorization

Editor: Bracket Pair Colorization → ✓

- Colors matching brackets
- Helps track code structure

Essential Extensions for Data Science

What are Extensions?

- Extensions add functionality to VS Code
- Similar to packages in R
- Install from VS Code Marketplace

How to Install Extensions

1. Click Extensions icon in Activity Bar (or `Ctrl+Shift+X`)
2. Search for extension name
3. Click "Install" button
4. Reload VS Code if needed

Basic R in VS Code

- Download and install R and VS Code, we have done this last week.
- Install languageserver in R

```
install.packages("languageserver")
```

- Install R Extension in VS Code
 - Search R in the extension market
 - Install R Extension by REditorSupport
- In addition, to enhance the experience, you can install radian in Python(but you have to install it) and httpgd in R respectively.

```
install.packages("httpgd")
```

Using VS Code with R

Setting Up R in VS Code

Prerequisites

- R already installed (from Week 3)
- R extension installed in VS Code

Configuration Steps

1. Open VS Code
2. Open a directory for your R projects(`File → Open Folder`) very important!
3. Open Command Palette (`Ctrl+Shift+P` or `Cmd+Shift+P`)
4. Type `R: Create R Terminal`
5. Select **R executable path** if prompted(*you have to know where R is installed in your computer*)

Running R Code in VS Code

Create a New R Script

1. File → New File
2. Save as `.R` file (e.g., `analysis.R`)
3. VS Code automatically detects R language

Basic R Script Example

```
# Load libraries
library(ggplot2)
library(dplyr)
library(AER)

# Load data
data <- CASchools

# Quick exploration
summary(data)
head(data)
```

Running R Code in VS Code

Key Commands for Running Code

Run Current Line

- `Ctrl + Enter` (Windows/Linux)
- `Cmd + Enter` (Mac)
- Cursor moves to next line automatically

Run Selection

- Select multiple lines
- Press `Ctrl + Enter` or `Cmd + Enter`
- Only selected code runs

Running R Code in VS Code

Key Commands for Running Code

Run Entire File

- `Ctrl + Shift + S` (Windows/Linux)
- `Cmd + Shift + S` (Mac)
- Runs from top to bottom

Source File

- Type in R Terminal: `source("script.R")`
- Runs file without showing output

Advantages Over RStudio

RStudio Advantages

- Integrated environment for R
- Environment pane shows objects
- Plot viewer built-in
- Package management GUI
- R Markdown rendering
- **Recommendation:**
- For beginning R users, RStudio is still a great choice.
- For multi-language projects and advanced users, VS Code is more powerful.

VS Code Advantages

- Multi-language support (R + Python)
- Better Git integration
- More customizable
- AI coding assistants (Copilot)
- Lighter weight
- Can do everything in one tool

Using VS Code with GitHub

Git Integration in VS Code

Prerequisites

- Git installed on your computer (Week 2)
- GitHub account created (Week 2)
- Git configured with your name and email

Verify Git Installation

Open terminal in VS Code (**Ctrl + `** or **Cmd + `**):

```
git --version
git config --global user.name
git config --global user.email
```

If not configured

```
git config --global user.name "Your Name"
git config --global user.email "your.email@examp
```

Clone a Repository from GitHub

Method 1: Using Command Palette

1. Open Command Palette (`Ctrl+Shift+P` or `Cmd+Shift+P`)
2. Type `Git: Clone`
3. Enter repository URL (from GitHub)
4. Choose folder location
5. Open the cloned repository

Method 2: Using Terminal

```
cd ~/Documents
git clone https://github.com/username/repository.git
cd repository
```

Then open the folder in VS Code: File → Open Folder

Clone a Repository from GitHub

Example: Clone Course Repository

```
git clone your repository url in github
```

- This creates a local copy of the course materials
- You can now view and edit files locally
- You can pull updates from the instructor
- **Note:** You won't be able to push changes to the course repository (unless you're a collaborator)

Basic Git Workflow in VS Code

Source Control Panel

- Click Source Control icon in Activity Bar (3rd icon)
- Or press `Ctrl+Shift+G` (Windows/Linux) or `Cmd+Shift+G` (Mac)

The Git Workflow

1. Make changes to files
2. Review changes in Source Control panel
3. Stage changes (+ icon next to files)
4. Write commit message in text box
5. Commit (✓ icon)
6. Push to GitHub (cloud icon or sync button)

Basic Git Workflow in VS Code

Step-by-Step Example

1. Modify a File

- Open `README.md`
- Add some text
- Save the file (`Ctrl+S` or `Cmd+S`)
- File appears in Source Control panel

2. Review Changes

- Click on the file in `Source Control`
- See side-by-side diff view
- Red = deleted lines
- Green = added lines

Basic Git Workflow in VS Code

3. Stage Changes

- Click  icon next to file
- Or stage all: Click  next to "Changes"
- Staged files move to "Staged Changes"

4. Commit Changes

- Write descriptive commit message
- Good: "Add data cleaning script"
- Bad: "Update"
- Click  icon to commit

5. Push to GitHub

- Click sync button ( icon)
- Or click "..." → "Push"
- Authenticate if needed

Working with Branches

Create a New Branch

1. Click on branch name in Status Bar (bottom-left)
2. Click `Create new branch`
3. Enter branch name (e.g., `test-branch`)
4. Press Enter

Working with Branches

Switch Between Branches

1. Click on branch name in Status Bar
2. Select the branch you want to switch to
3. VS Code updates files to match that branch

Merge Branches

1. Switch to the branch you want to merge into (usually `main`)
2. Open Command Palette
3. Type `Git: Merge Branch`
4. Select the branch to merge
5. Resolve any conflicts if needed

Hands-on Practice

Practice Exercise

Task: Set Up Your Course Project Workspace

Goal: Create a well-organized project structure for this course

Step 1: Create Folder Structure

- Create new folders for your project work:

DSAI_YourTeamName/

- ├─ RawData/
- ├─ WorkData/
- ├─ SaveData/
- ├─ Figures/
- ├─ Scripts/
- ├─ Figures/
- ├─ Reports/
- └─ README.md

Practice Exercise

Step 2: Initialize Git Repository

Step 3: Create Initial Files

Step 4: write a simple r script

Step 5: Run the R Script

Step 6: Commit and Push to GitHub

Step 7: Verify on GitHub

Practice Exercise

Congratulations!

You have successfully:

- ✓ Set up VS Code for data science
- ✓ Created a project structure
- ✓ Written and run R code in VS Code
- ✓ Used Git and GitHub integration
- ✓ Pushed your work to GitHub