

Applied AI Tools for Scientific Programming

Scientific workflows: Tools and Tips 

Selina Baldauf

2025-12-18

What is this lecture series?

Scientific workflows: Tools and Tips

 Every 3rd Thursday  4-5 p.m.  Webex

- One topic from the world of scientific workflows
- Material provided [online](#)
- If you don't want to miss a lecture [subscribe to the mailing list](#)
- For credit points: Send me a short message (Email or Webex)

Motivation

- Speed up repetitive tasks
- Learn new methods and languages
- Support debugging, documenting, refactoring, ...

AI tools for programming

- Browser-based chat bots (ChatGPT, etc.)
- Web-tools for data analysis (Data analyst GPT, JuliusAI, ...)
- **IDE-integrated AI tools** (GitHub Copilot, Codium AI, ...)

Integrated tools

- know your project structure
- see your code and console output
- respond in context while coding

Today

GitHub Copilot inside the IDE Positron

- How to use it
- Limitations and responsible use

Let's get started

GitHub Copilot in Positron

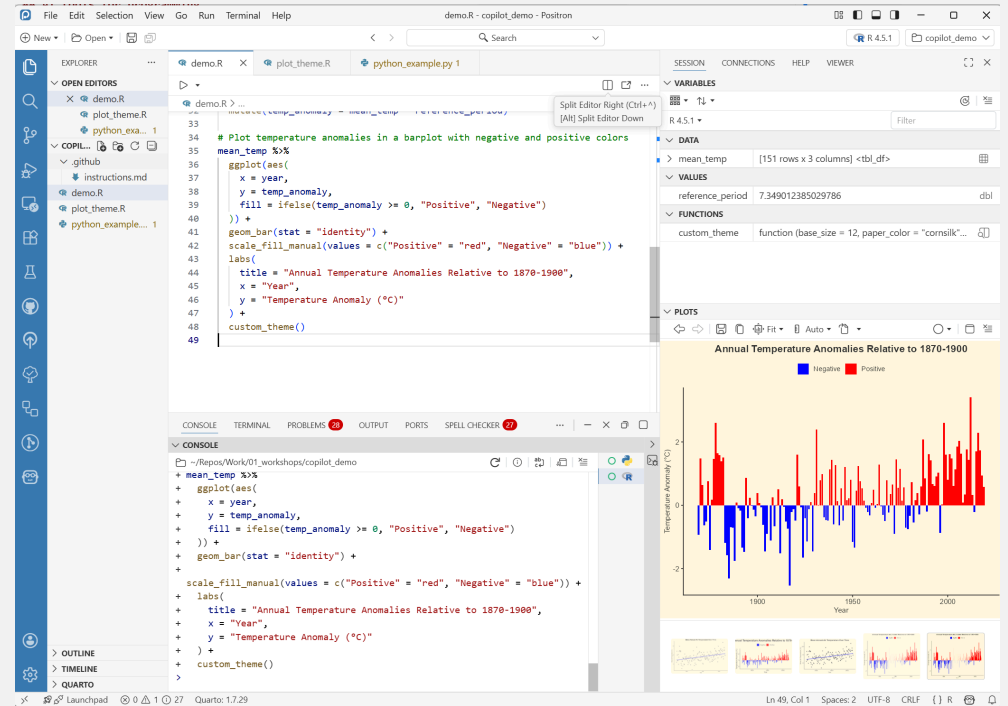
GitHub Copilot

- **Integrated** and **context-aware**
- **Standard** tool for different environments
- **Free** for academics with an educational account
 - See [lecture website](#) for guide on how to get it



Positron

- General-purpose data science IDE
- Built by the same team behind RStudio
- Based on VS Code
- Complete GitHub Copilot integration



Code auto-complete

Copilot predicts what you want to write based on

- Current code
- Open files
- Comments

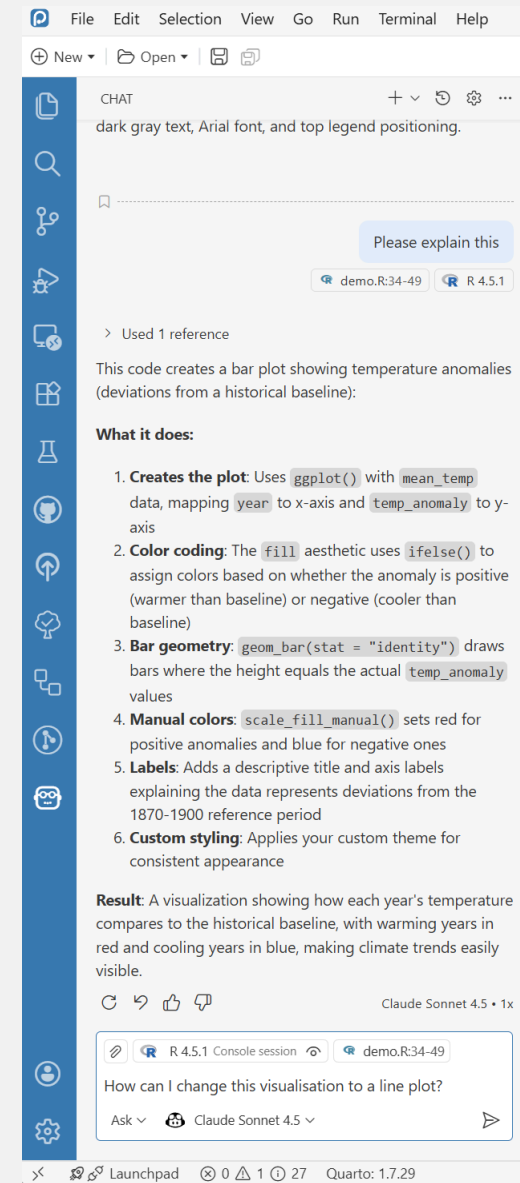
```
fibonacci.R > fibonacci
1  fibonacci <- function(n) {
2    if (n == 0) {
3      return(0)
4    } else if (n == 1) {
5      return(1)
6    } else {
7      return(fibonacci(n - 1) + fibonacci(n - 2))
8    }
9  }
```

Get better suggestions

- Provide context
 - Good comments and names for functions and variables
 - Open other relevant files
- Adopt a good coding style
 - GitHub Copilot will imitate your style
- Acceptance discipline
 - Don't auto-accept everything
 - Modify suggestions if needed

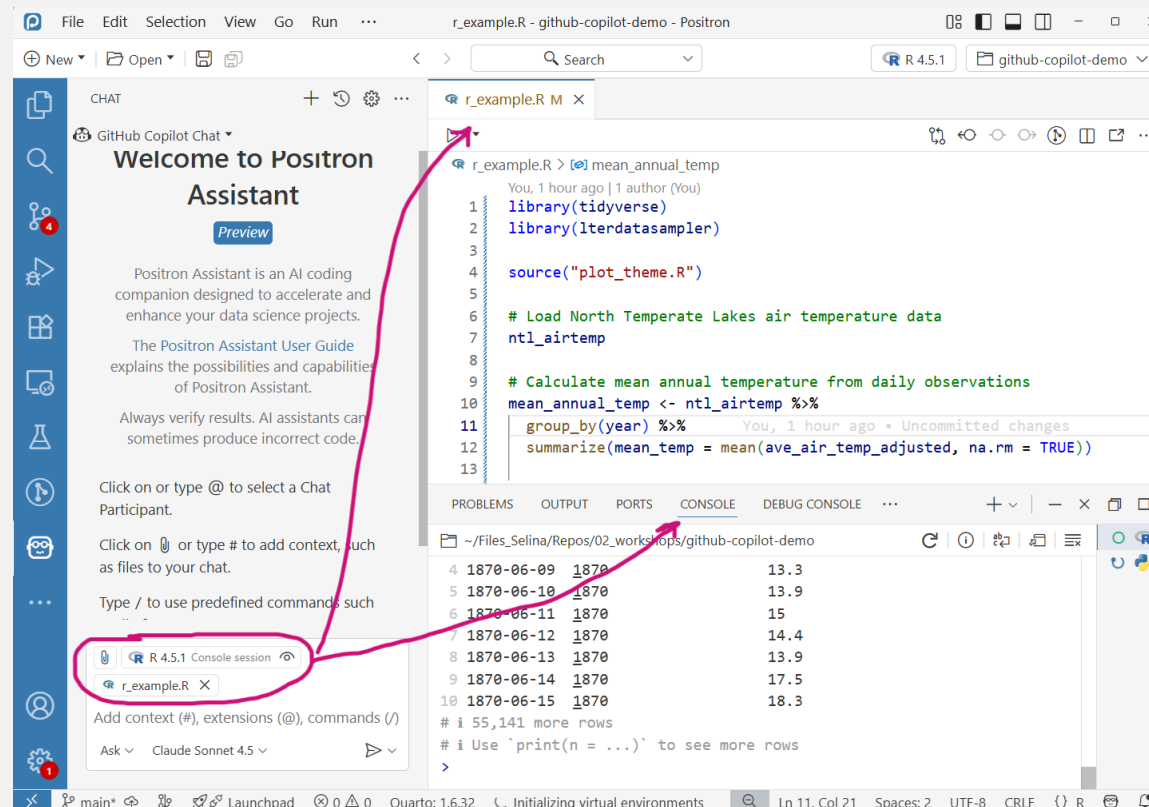
Sidebar Chat

- Tell GitHub Copilot what to do with your code
- Larger contexts and longer conversations
- Can see: files, console output, project workspace



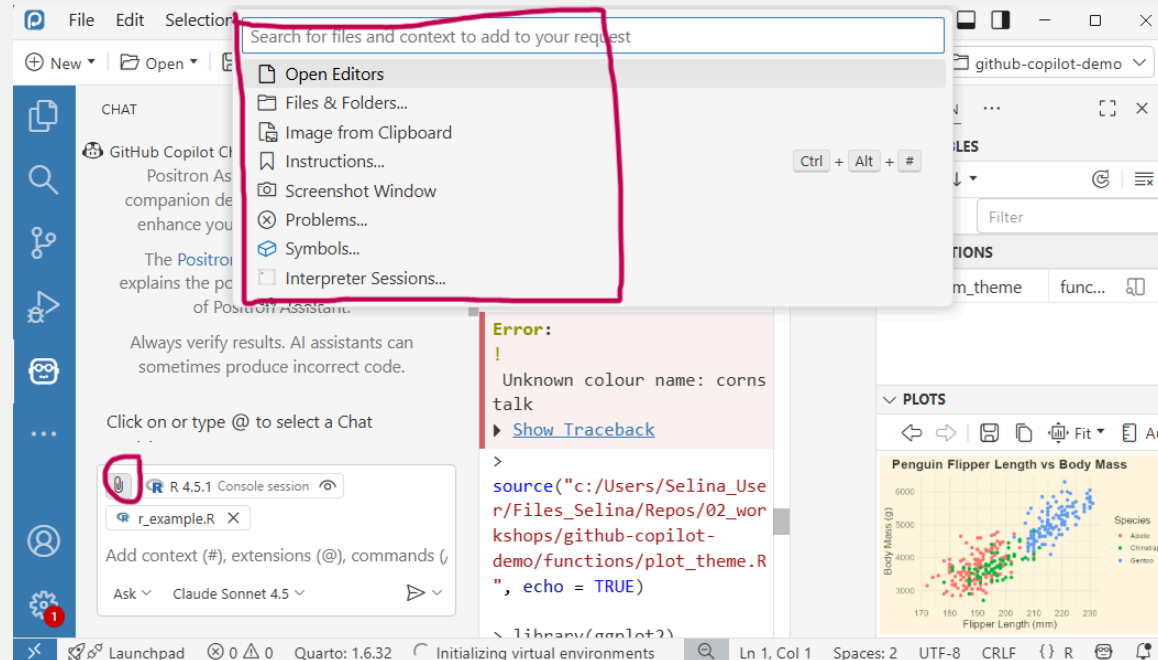
The default context

- Context is indicated at the top of the chat
- Default context: current file + console output
- If you highlight code, the highlighted code is added as context



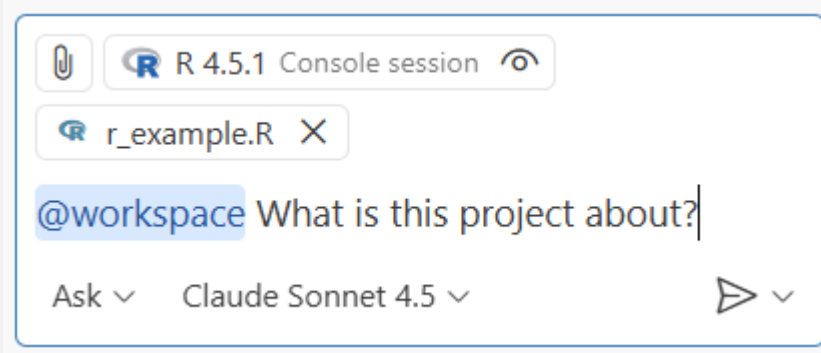
The “Add context” button

- Add other files and folders as context with the “Add context” button

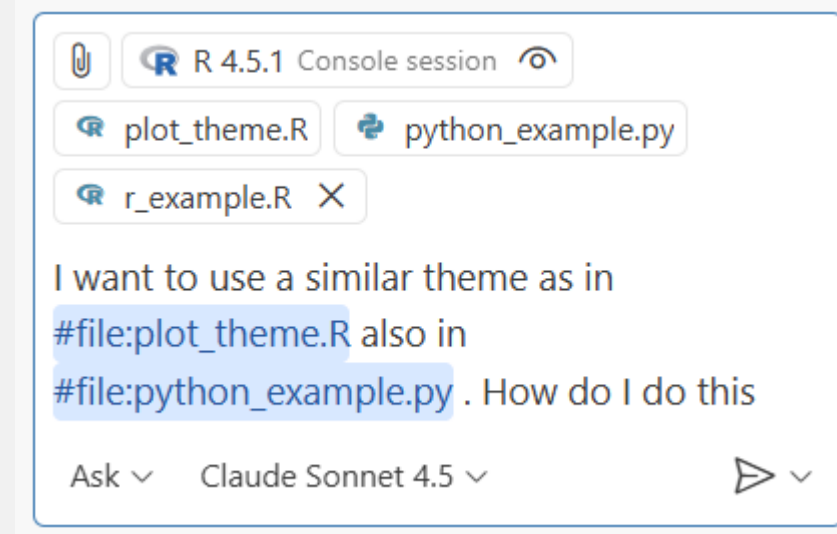


Use @, # or copy-paste

@workspace adds all files in the project



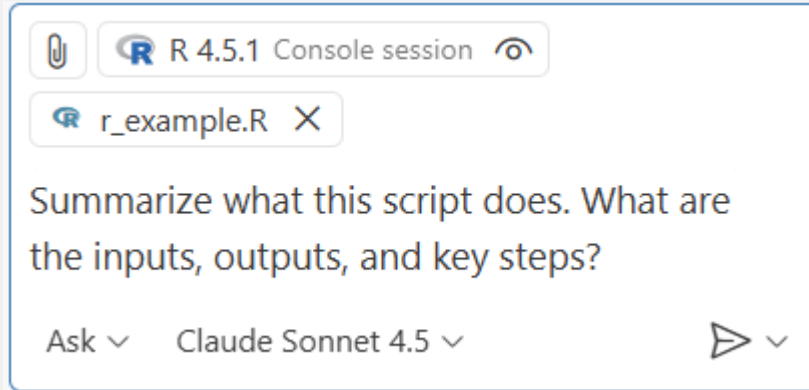
#file adds specific files to the context



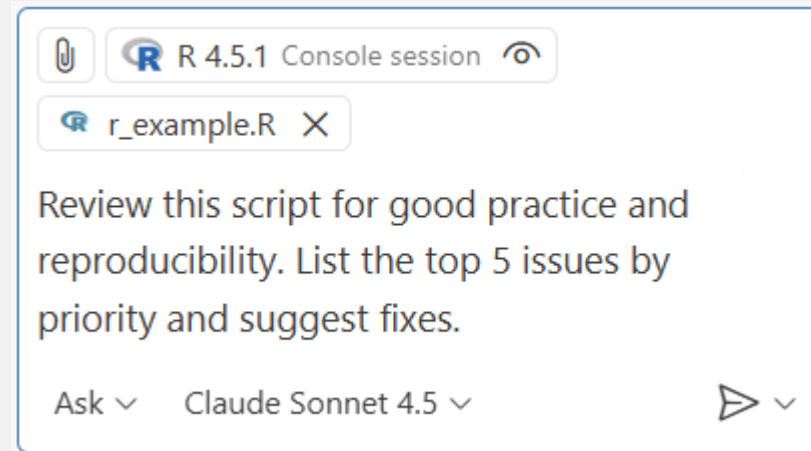
You can also copy paste images (e.g. screenshots) as context in the chatbox

Sidebar chat: prompt ideas

Overview of unfamiliar scripts

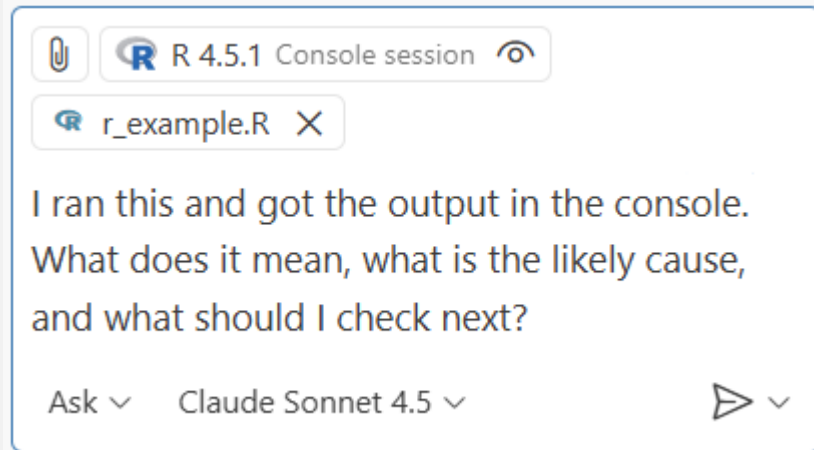


Review your own code

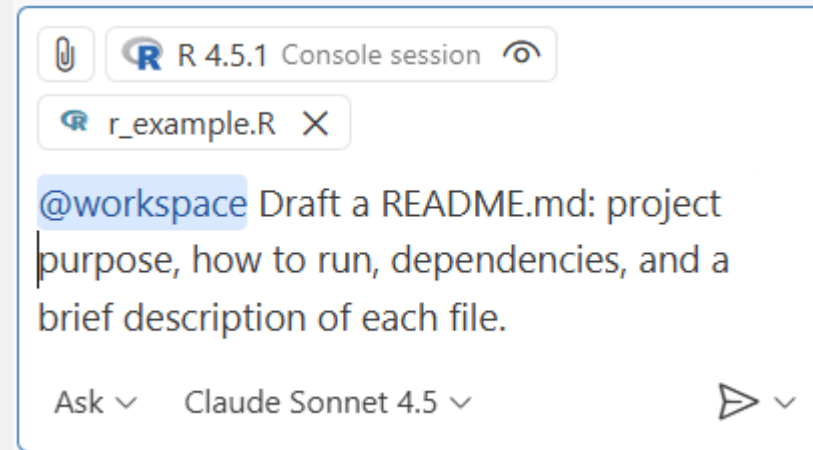


Sidebar chat: prompt ideas

Understand output (errors, warnings, surprising results)



Write project-level documentation draft



Advanced features

- Project-level instruction files
 - Give instructions that apply to the whole project
 - Special file in `/.github/instructions.md`
- **Agents:** Automate multi-step workflows (use with care)

Limitations and risks

- Can suggest
 - Outdated functions and code
 - Patterns that are common but not best practice
- GitHub Copilot has no understanding
- Predicts plausible code not necessarily correct code
 - Code can be subtly wrong or inappropriate
- Risk of over-reliance

Responsible use

- Don't treat AI as an authority: review and double-check
- Use version control as a safety net: review and trace changes
- Privacy: Code, comments and console output may be shared with model providers
 - Beware if you have sensitive data
- Check institutional and journal guidelines
- Transparency: Disclose your use of AI tools
- You are responsible for your scientific output

Conclusion

- GitHub Copilot can be a great **assistant**
- Support but not replace

Getting started

1. Get GitHub Copilot for free with your educational GitHub account
2. Install Positron
3. Setup GitHub Copilot in Positron

Check out the [lecture website](#) for all relevant information

Next lecture

Topic t.b.a.

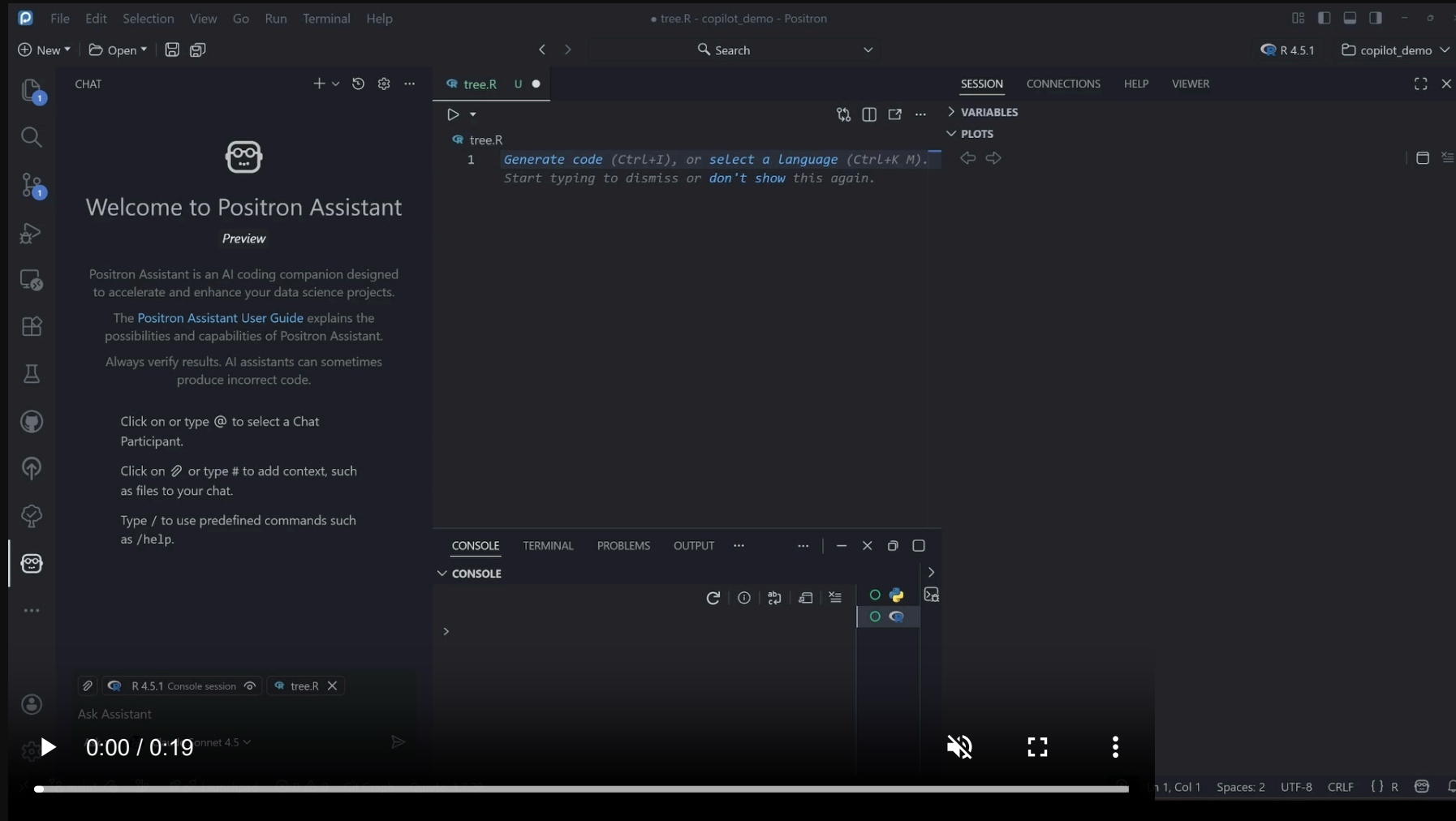
 15th January  4-5 p.m.  Webex

 [Subscribe to the mailing list](#)

 For topic suggestions and/or feedback [send me an email](#)

Thanks for your attention :)

Questions?



The screenshot displays the Positron IDE interface. On the left, the 'CHAT' panel is open, showing a 'Welcome to Positron Assistant' message. The message includes a 'Preview' section and instructions on how to interact with the assistant, such as using '@' to select a chat participant, '#' to add context, and '/' for predefined commands. At the bottom of the chat panel, there is a 'Ask Assistant' input field and a timer showing '0:00 / 0:19'. The main editor area shows a file named 'tree.R' with a single line of code: 'Generate code (Ctrl+I), or select a Language (Ctrl+K M). Start typing to dismiss or don't show this again.' The bottom status bar indicates the current line and column (1, Col 1), spacing (Spaces: 2), encoding (UTF-8), line ending (CRLF), and the active language (R).