References

AI tools in programming

Scientific workflows: Tools and Tips 🛠️

Dr. Selina Baldauf

2024-01-18

What is this lecture series?

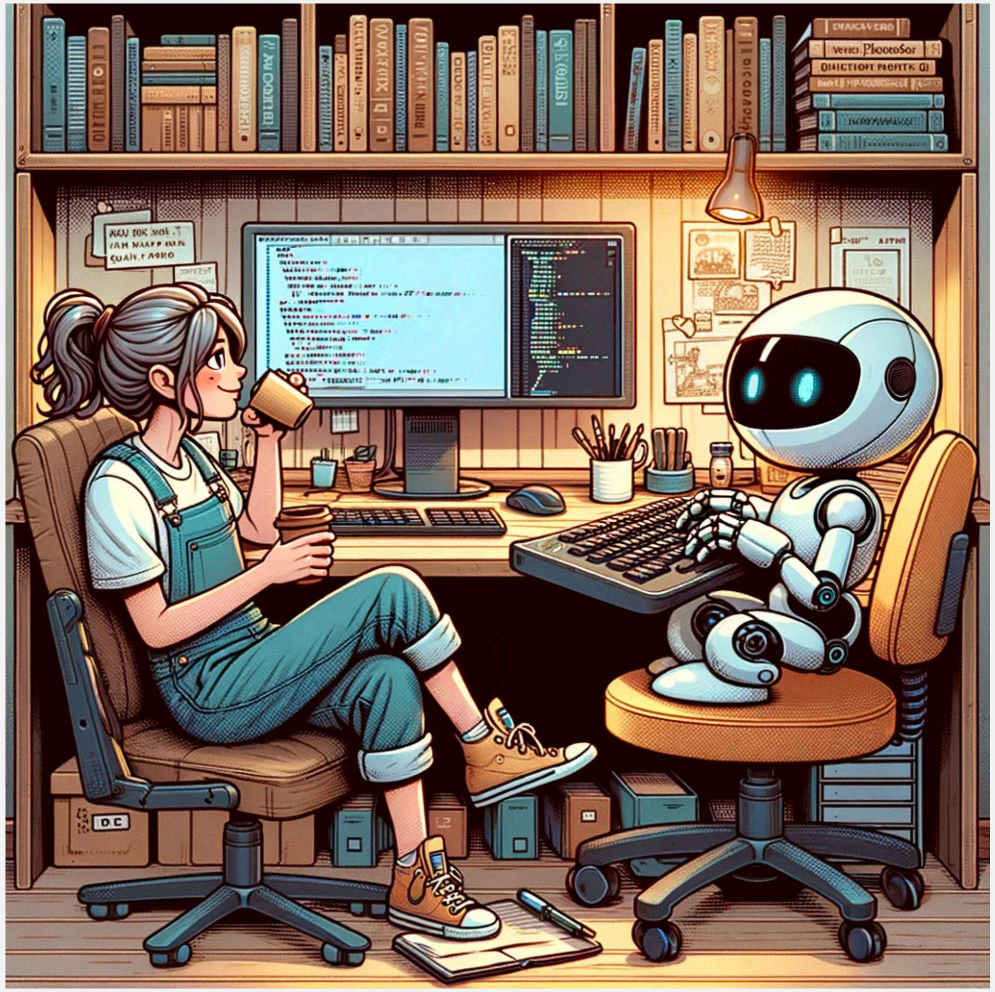
Scientific workflows: Tools and Tips 🛠️

📅 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

Motivation

- AI tools assist programmers with
  - Coding
  - Debugging
  - Learning
  - ...
- Higher productivity and efficiency
- More motivation



Overview of tools

- **Browser-based chat bots** (ChatGPT, Bard, ...)
  - General-purpose
- **Data-analysis tools** (Data analyst GPT, RTutor, ...)
  - Upload data and ask questions about it
  - Download the code that was used for the results
- **Integrated AI tools** (GitHub Copilot, Codium AI, ...)
  - Integrated directly in programming environment
  - Real-time suggestions, chat, debugging, ...

Today

- Focus on **integrated AI tools**
  - How to use GitHub Copilot to
    - Speed up your coding
    - Improve your code
    - Learn
- Concerns when using AI tools
- **Main goal:** Motivate you to try out tools and find out what fits your workflow
- Find other tools on the website

Now You

- ❓ What is your **main programming language**
- ❓ Which **IDE (programming environment)** do you use
- ❓ Which **AI tools for programming** did you already try

Integrated AI tools for programming

Mainly GitHub Copilot

GitHub Copilot

- Cloud-based AI tool by Github and OpenAI
- Model based on GPT-4 and OpenAI's Codex
  - Specifically trained on source code
- Basic idea: Plugin for your IDE to integrate Copilot
- Works best for well-represented languages (Python, JS, ...)

How to get GitHub Copilot

See lecture website for step-by-step guide and more information.

It's really easy, but you need:

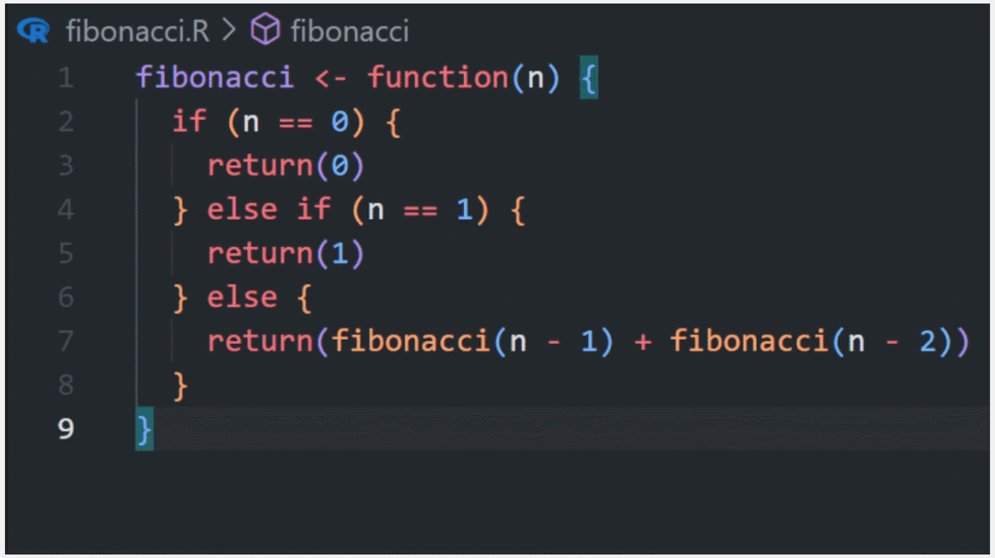
- GitHub Account
- Active GH Copilot subscription (10$ per month)
  - Get it for free as an academic with an educational account
- IDE that supports Copilot
  - Full support: Visual Studio (Code), Vim, Neovim, JetBrains IDEs (e.g. PyCharm)
  - Limited support: RStudio, ?

Using GitHub Copilot

Demo of the main features and use cases

Inline code suggestions

- Copilot tries to predict what you want to do next
- Suggestions are based on the context
  - Previous code
  - Comments
  - Variable and function names
  - …



```r
fibonacci <- function(n) {
  if (n == 0) {
    return(0)
  } else if (n == 1) {
    return(1)
  } else {
    return(fibonacci(n - 1) + fibonacci(n - 2))
  }
}
```

Get better suggestions

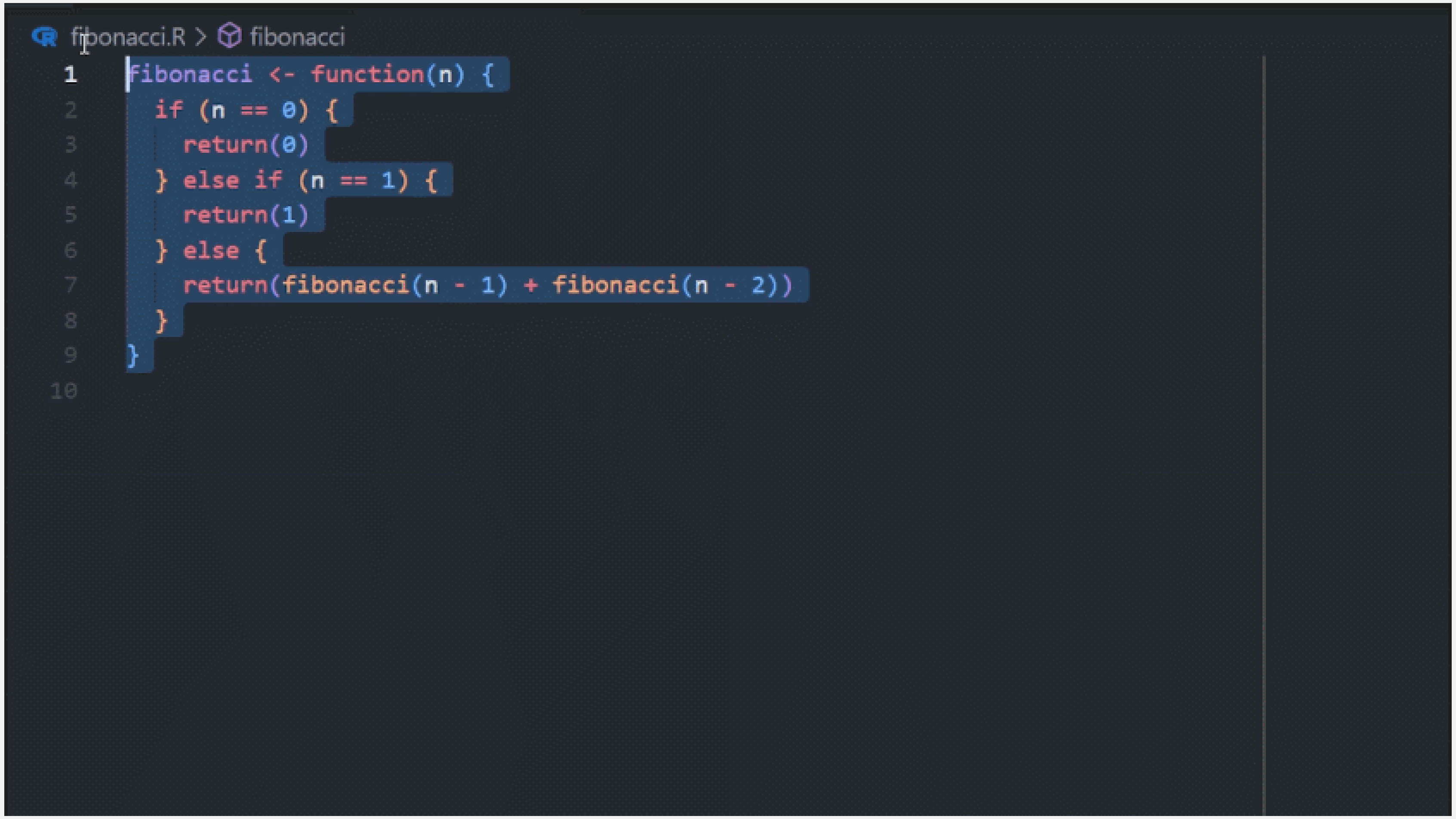
- Provide context
  - Open other files
  - Add top level comments explaining the purpose of the script
  - Name variables and functions properly
  - Copy-paste sample code and delete it later
- Be consistent
  - "Garbage in, garbage out"
  - Have a nice and consistent coding style

Nice side effect of using Copilot: More good-practice coding

Chat

- Ask and give commands regarding:
  - Highlighted lines of code
  - The whole script or project
- Preset commands starting with `/`
  - `/fix`: fix problems in your code
  - `/doc`: get documentation
  - `/explain`: explain this code
  - `/test`: write unit tests
  - `/new`: create new projects or scripts with code

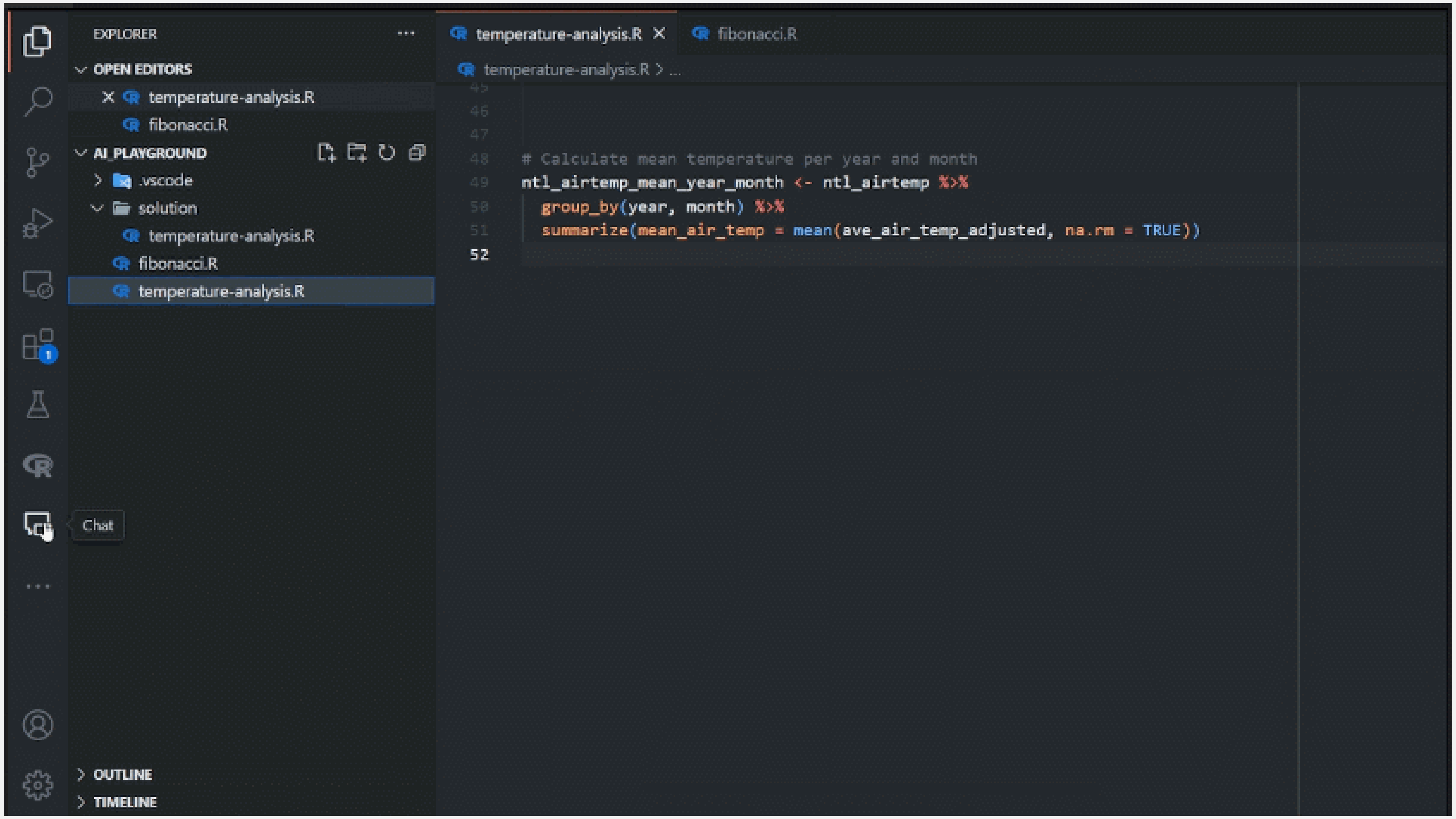
/fix with in-line chat



fibonacci.R > fibonacci

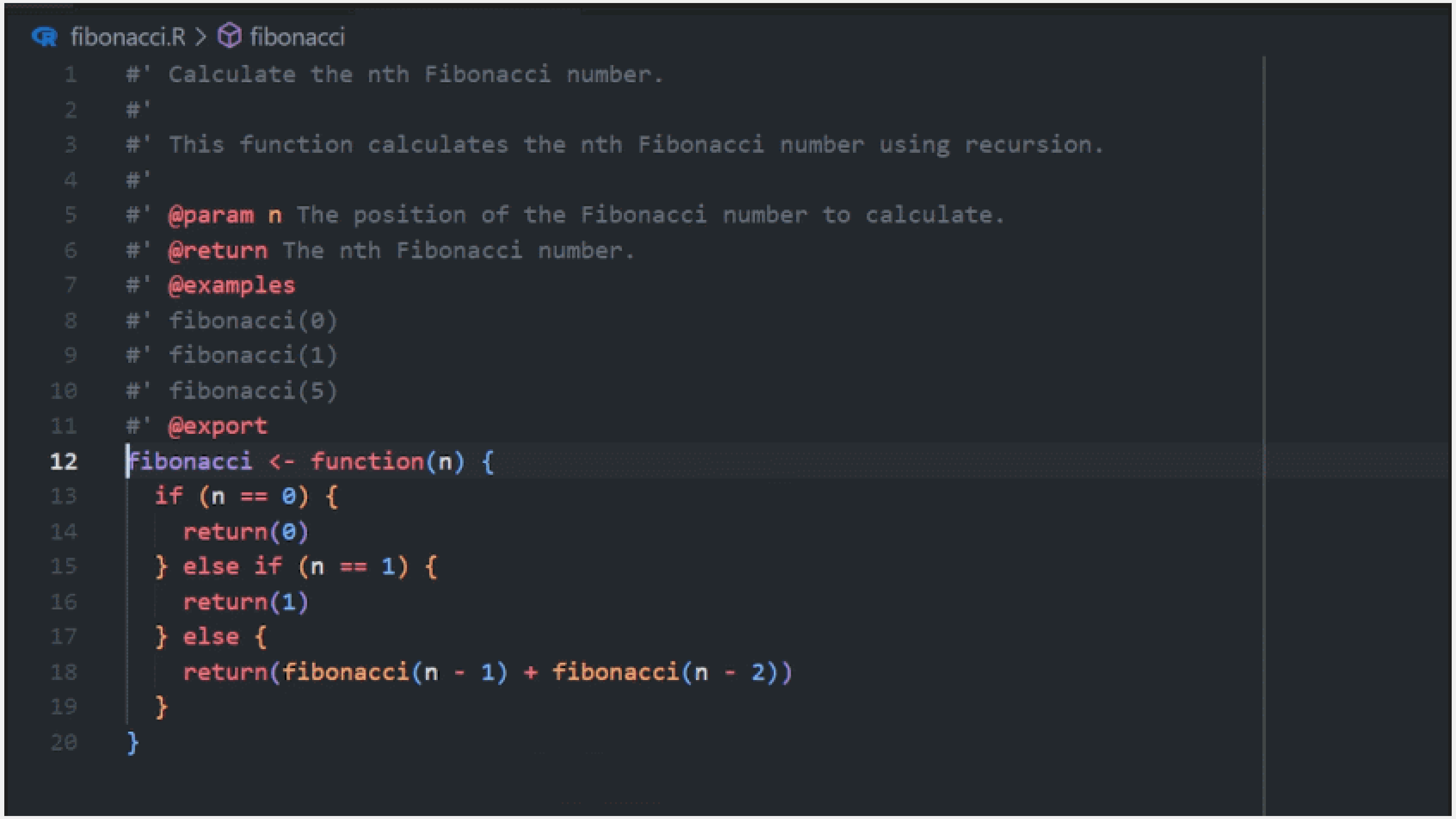
```r
fibonacci <- function(n) {
  if (n == 0) {
    return(0)
  } else if (n == 1) {
    return(1)
  } else {
    return(fibonacci(n - 1) + fibonacci(n - 2))
  }
}
```

/fix with chat in the sidebar



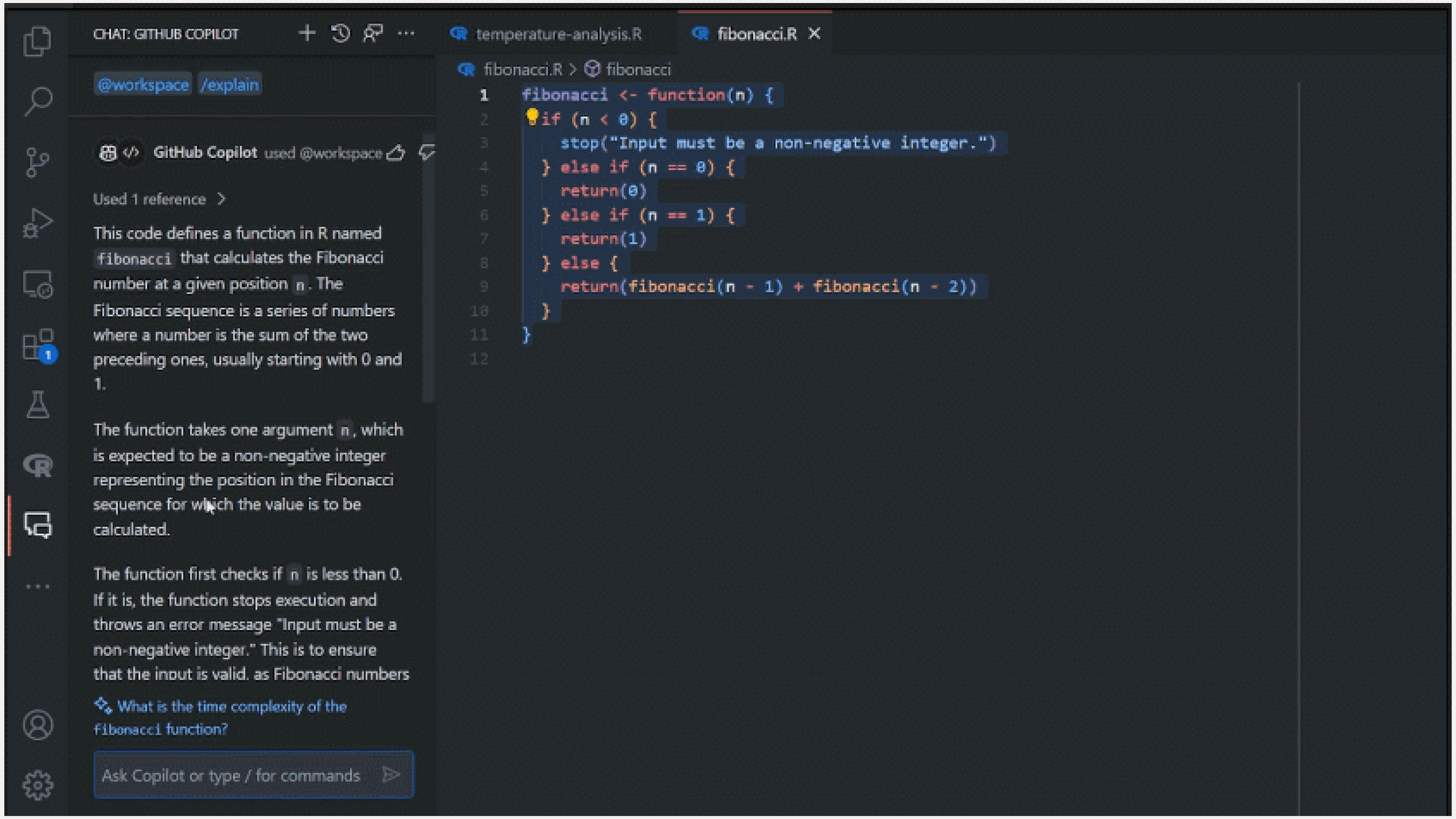
```
# Calculate mean temperature per year and month
ntl_airtemp_mean_year_month <- ntl_airtemp %>%
  group_by(year, month) %>%
  summarize(mean_air_temp = mean(ave_air_temp_adjusted, na.rm = TRUE))
```

/doc with specific documentation standard



```r
#' Calculate the nth Fibonacci number.
#'
#' This function calculates the nth Fibonacci number using recursion.
#'
#' @param n The position of the Fibonacci number to calculate.
#' @return The nth Fibonacci number.
#' @examples
#' fibonacci(0)
#' fibonacci(1)
#' fibonacci(5)
#' @export
fibonacci <- function(n) {
  if (n == 0) {
    return(0)
  } else if (n == 1) {
    return(1)
  } else {
    return(fibonacci(n - 1) + fibonacci(n - 2))
  }
}
```

/explain



This code defines a function in R named `fibonacci` that calculates the Fibonacci number at a given position `n`. The Fibonacci sequence is a series of numbers where a number is the sum of the two preceding ones, usually starting with 0 and 1.

The function takes one argument `n`, which is expected to be a non-negative integer representing the position in the Fibonacci sequence for which the value is to be calculated.

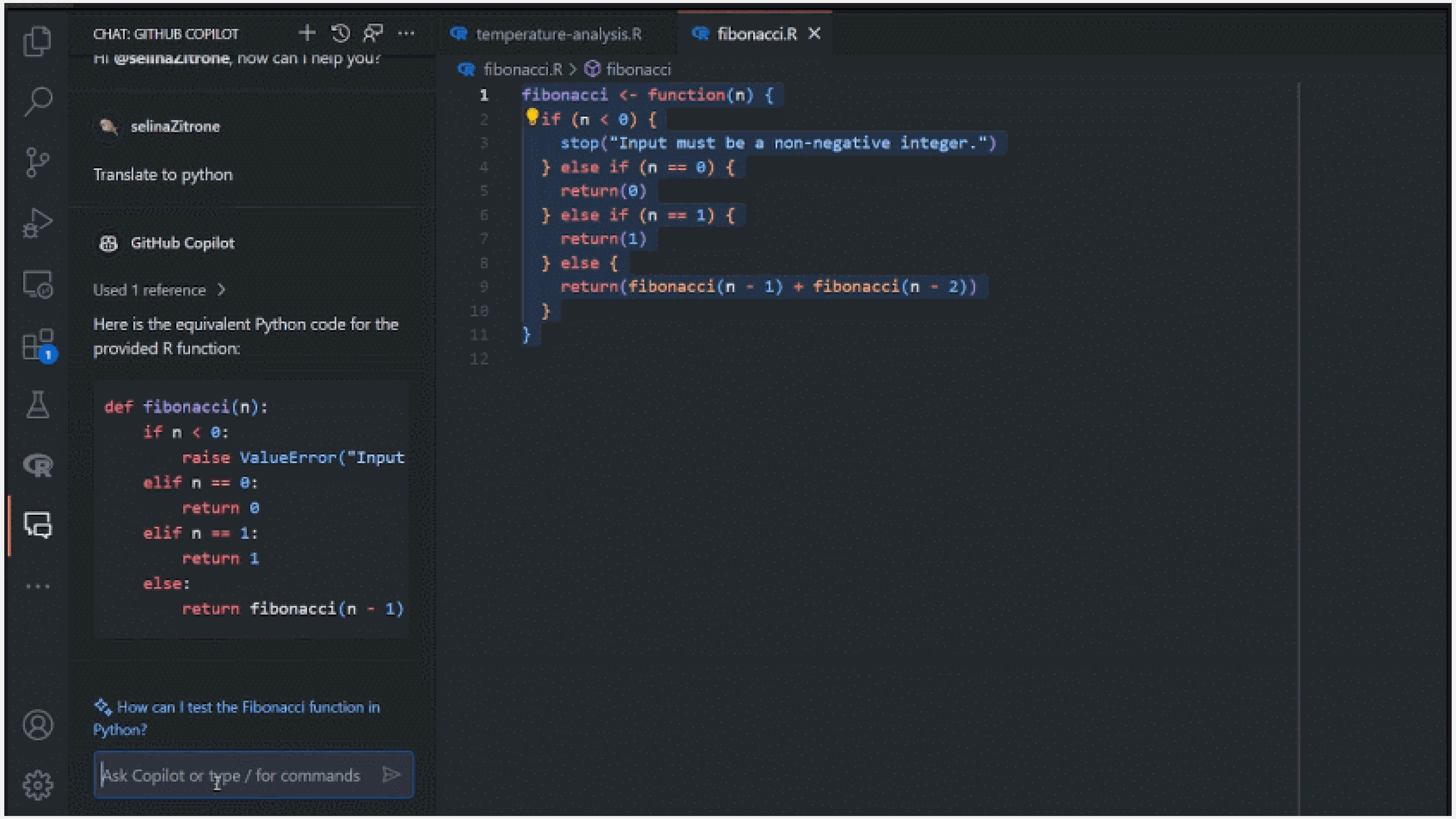
The function first checks if `n` is less than 0. If it is, the function stops execution and throws an error message "Input must be a non-negative integer." This is to ensure that the input is valid, as Fibonacci numbers

What is the time complexity of the `fibonacci` function?

Ask Copilot or type / for commands

```r
fibonacci <- function(n) {
  if (n < 0) {
    stop("Input must be a non-negative integer.")
  } else if (n == 0) {
    return(0)
  } else if (n == 1) {
    return(1)
  } else {
    return(fibonacci(n - 1) + fibonacci(n - 2))
  }
}
```

Translate code



Hi @selinaZitrone, how can I help you?

selinaZitrone

Translate to python

GitHub Copilot

Used 1 reference >

Here is the equivalent Python code for the provided R function:

```python
def fibonacci(n):
    if n < 0:
        raise ValueError("Input
    elif n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fibonacci(n - 1)
```

How can I test the Fibonacci function in Python?

Ask Copilot or type / for commands

```r
fibonacci <- function(n) {
  if (n < 0) {
    stop("Input must be a non-negative integer.")
  } else if (n == 0) {
    return(0)
  } else if (n == 1) {
    return(1)
  } else {
    return(fibonacci(n - 1) + fibonacci(n - 2))
  }
}
```

Codium AI as an alternative

- No inline code suggestions
- Great functionality to
  - Explain code
  - Suggestsions improve and enhance code
  - Generate tests
- Not in RStudio, but in VS code and many other IDEs
- Free for personal use (for now)

Concerns to consider

- Privacy
  - Chose whether your prompts and suggestions will be used by Github (`Github -> Seetings -> Copilot -> Policies`)
- Plagiarism
  - Block suggestions matching public code (`Github -> Seetings -> Copilot -> Policies`)
- Ethical concerns
  - For-profit tool trained on open-source
- Environmental concerns
  - Water and enery usage

Usage guidelines

- No definite guidelines, but see examples on lecture website
- Responsibility
  - You are responsible for your scientific output
  - Stay critical, double-check
- Transparency
  - Make clear for which tasks you used which AI
- Know relevant guidelines
  - Journals
  - Your university
- Don't use AI in exams

Summary

- AI tools for programming can be extremely useful
- Try different tools and find the ones you like
- Think about concerns
- Learn about relevant guidelines
- Development is fast, so keep up
- Check out the lecture website if you want to get started

Next lecture

Topic t.b.a.

15th February 4-5 p.m. Webex

Subscribe to the mailing list

For topic suggestions and/or feedback send me an email

Thank you for your attention :)

Questions?

Thanks to Anne Lewerentz for support with the preparation.

References

- Experiment on programmer efficiency with AI tools
- GitHub Copilot
- GitHub Copilot privacy FAQ
- GitHub Copilot Docs: Useful information and guides on how to use Copilot
- Prompt engineering with GitHub Copilot
- Codium AI

Guidelines

- DFG Rules on the use of AI particularly for proposals
- Nature living guidelines on responsible use of generative AI in research
- EU AI Act
- Universities (German)
  - FU Berlin “Eckpunktepapier” (German)
  - TU Berlin on AI: Mainly about AI in teaching but contains some general links to other guidelines