

# ETC5512: AI tips

## Table of contents

Tips for Using AI to code (without cheating) . . . . .	1
<b>Some final remarks</b>	<b>3</b>

### Tips for Using AI to code (without cheating)

#### 1. Use AI to explain code concepts

If you encounter a coding concept that you don't understand ask AI to explain it to you.

For example, "Can you explain how data frames work in R?" or "What's the difference between a list and a vector in R?"

This helps you understand the fundamentals without compromising your learning.

#### 2. Request explanations of code

If you find examples of code you can ask AI to break down how it works line by line.

This improves your comprehension of code and its structure. Never copy and paste code chunks you do not understand. This is a form of plagiarism and does not accurately represent your skills.

#### 3. Guide AI to your coding level

If AI provides you with code you don't understand don't copy it. Instead, ask AI to try again with simpler code that is better suited for a beginner.

For example, "I don't understand that code. Can you help me to use simpler functions from the tidyverse that I can understand."

#### **4. Get code syntax help, not full code snippets**

When you know what function you need, but can't remember the exact coding syntax, AI can be a quick reference.

It's better to ask "Remind me, what is the syntax for creating a ggplot histogram?" rather than "Write me code to produce a histogram." You must be able to explain any code you write.

#### **5. Find a function to help solve a specific problem**

Often you know what you want to do but don't know what functions exist. AI can help you find out about different functions. I've learnt about lots of new functions this way!

Try creating a small example of your problem and ask what functions could help you solve it. e.g. "Is there a simple function to convert a string "2025-Mar-01" to a date format?"

#### **6. Create toy examples of your problem (smallest possible version), then ask for help**

If you are stuck with something, break the problem down into its simplest possible building blocks. Identify the building block you need help with. Then convert this to the simplest possible example that shows your problem – a toy example.

Asking AI for help on toy examples you create, is very different to asking AI to solve a question directly from your assignment. You can adapt the AI solution from the toy problem to your assignment without breaching academic integrity.

Explaining and creating toy problems is also a great way to boost your understanding.

#### **7. Use AI to understand error messages**

R error messages can be cryptic. Paste the error message to AI and ask for a plain-language explanation of what's wrong and general approaches to fix it. Then attempt to fix your code yourself, without asking for the exact solution.

#### **8. Debugging your code**

If your code isn't working, always try debugging it yourself first. If after investigating the error message you are still stuck, try sharing part of your code with AI and ask for help identifying the issue rather than requesting a complete rewrite. Example: "My code isn't working. Can you help me spot which line might be causing the error?"

## 9. Generate practice problems

I love this one. Ask AI to create custom practice exercises. You may like to share material from class or your assignments to get additional practice.

## 10. Ask AI to review your code and for best practice

Ask about AI about coding conventions or efficiency improvements.

For example: “What are some R coding best practices I could implement for writing more maintainable code?” or “How could I improve my code for readability?”

This helps you develop good habits without short cutting the learning process.

## Some final remarks

### ! Remember: AI is not a substitute for thinking

The goal is to use AI as a learning assistant, not as a substitute for developing your own skills.

Always check your unit’s policy on AI use, and when in doubt, consult with your instructor about appropriate AI use for coursework.

You can review this unit’s policy in our first lecture along with the links to Monash policy and support resources.

### ! AI use statements

To be transparent about your AI use, keep a record of your prompts and share them alongside the AI responses when you submit your assignments. Also be clear about how you used AI.

Here I already had an idea about tips I wanted to share, and I used AI to help build out my ideas and adapted the tips to be suitable for this context.

Below is my prompt:

I am writing up to 10 tips about how to use AI to code in R for beginners. These tips should be for students who are learning, and who should not be cheating. The students are at a university level

While I used AI to help me write this! It’s 75% me and 25% AI. I also take full accountability for the work. It is also important I don’t pass this off as entirely my own work.

Look at the AI use statement examples provided alongside your assignment