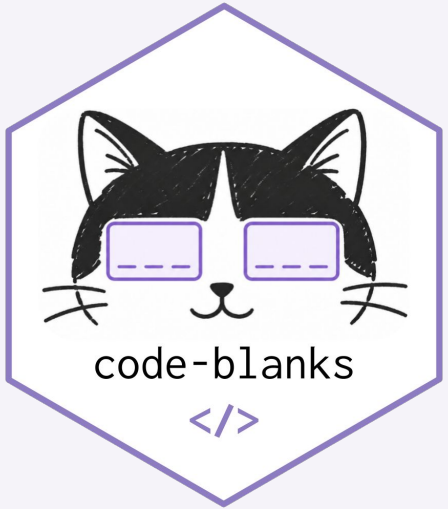
Learning through tinkering

A scaffolded approach to introducing programming to data science students using **code blanks**

Anna Fergusson
University of Auckland | Waipapa Taumata Rau
Aotearoa New Zealand
a.fergusson@auckland.ac.nz

THE UNIVERSITY OF AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND



The introductory-level data science education challenge

Tasks that **introduce computer programming** need to **support students to integrate statistical and computational thinking** to learn from/with data, models, and technologies (Fergusson, 2023) and empower students to engage in personally or socially meaningful projects (Fergusson & Hüsing, 2026).

Code blanks is a scaffolded way of presenting code where only **selected parts are editable**, with code aspects **progressively removed and replaced with blanks** based on an **analysis of the internal structure of the learning content** (Renkl et al., 2004).

Approach is consistent with using programming plans that provide the code needed to **produce a required output for a purpose**, with scaffolding that **directs learners' attention to the specific parts** of the code that could be adapted or extended (Cunningham et al., 2021).

What are code blanks?

▶ Run Code | ↺ Reset Code | 📋 Copy Code

```
weather_data <- get_nasa_weather_data(location = "Auckland",
                                      start_date = "2023-01-01",
                                      end_date = "2025-01-01")

ggplot(data = weather_data) +
  geom_line(aes(x = date,
                y = temperature_C,
                colour = location))
```

Only part of the code that can be changed

Teaching context: DATASCI 100

DATASCI 100 Data science for everyone

Module 1: Producing data-based communications involving visualisations

Module 2: Implementing algorithms for discovering and analysing features of data

Module 3: Creating data science products using informal generative models

Module 4: Developing models from data to make predictions and classifications

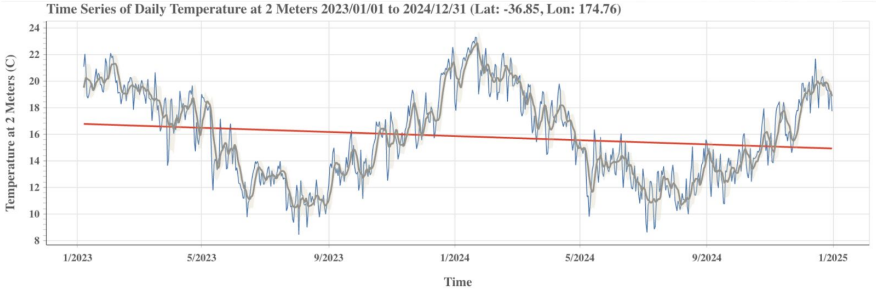
"…. Across the course, students will be introduced to a wide range of data science applications and be supported to think creatively, ethically, and responsibly about how to use data, models, and technology to produce solutions to identified problems or challenges."

Example one: Connecting GUI actions with code actions

Earlier in the learning activities for that week, students had engaged with the NASA climate data website (POWER) to access and visualise temperature data for Auckland.

How's the weather in Auckland?

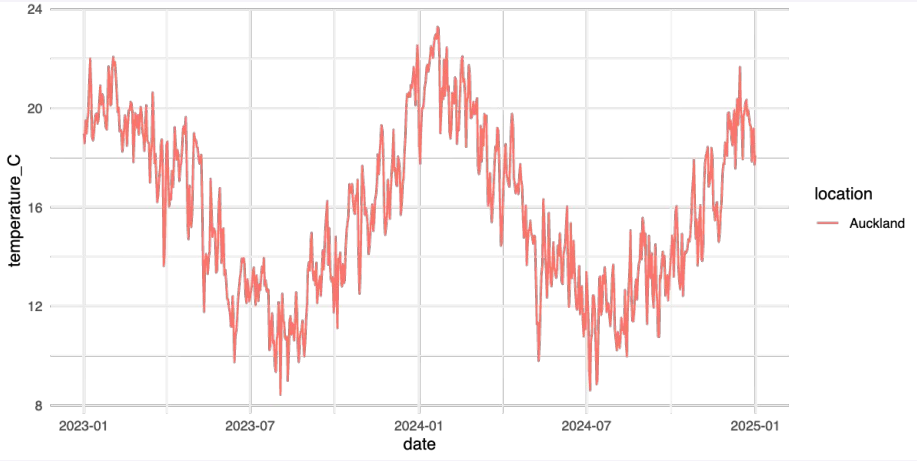
The visualisation below was produced by the NASA data access viewer and displays the temperature in Auckland, along with some extra modelling components such as the red trend line (we won't worry about these additional modelling components for now).



▶ Run Code | ↺ Reset Code | Copy Code

```
weather_data <- get_nasa_weather_data(location = "Auckland",
                                      start_date = "2023-01-01",
                                      end_date = "2025-01-01")

ggplot(data = weather_data) +
  geom_line(aes(x = date,
                y = temperature_C,
                colour = location))
```



Example two: Introduce new programming concepts

▶ Run Code | ↺ Reset Code | 📋 Copy Code

```
weather_data <- get_nasa_weather_data(location = Auckland ,


ggplot(data = weather_d
  geom_line(aes(x = dat
                y = tem
                colour
```

```
Error:
object 'Auckland' not f
```

▶ Run Code | ↺ Reset Code | 📋 Copy Code

```
weather_data <- get_nasa_weather_data(location = c("Auckland", "Christchurch") ,
                                      start_date = "2023-01-01",
                                      end_date = "2025-01-01")

ggplot(data = weather_data) +
  geom_line(aes(x = date,
                y = temperature_C,
                colour = location))
```

ⓘ **Note**

The `c()` function was used here to supply more than one location.

Example three: Structured creativity

▶ Run Code | ↺ Reset Code | 📋 Copy Code

```
weather_data <- get_nasa_weather_data(location = "Auckland",
                                      start_date = ,
                                      end_date = )

weather_data %>%
  ggplot() +
  geom_line(aes(x = date,
                y = temperature_C,
                colour = location)) +
  labs(title = ,
       subtitle = ,
       caption = ) +
  scale_x_date(date_breaks = "3 months",
               date_labels = "%d-%b\n%Y",
               date_minor_breaks = "1 month")
```

General design approach for using code blanks

Scaffolded code interactions require **tinkering with the programming plan**:

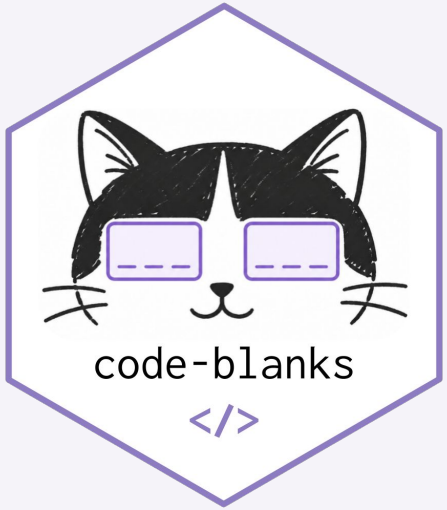
- Give an instruction to run code and see what happens
- Provide some explanation or motivation to tinker with the code
- Students make changes to the code and reflect on what is produced
- Students write down their ideas or respond to specific questions

Use the same programming plan (code example) to introduce and try out new computational knowledge.

At the same time, try to keep integrating statistical thinking as well (link back to purpose).

Code blanks in action!

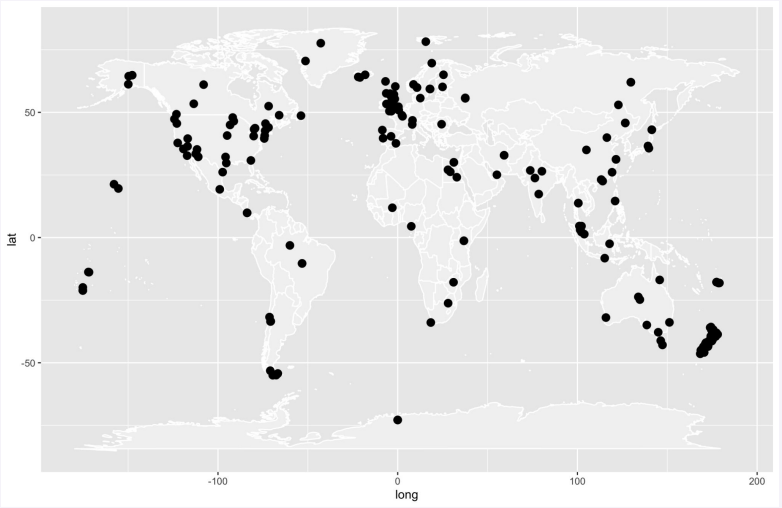
Demo lab task from DATASCI 100
annafergusson.online/sdss2026/



{code-blanks} quarto extension
github.com/annafergusson/code-blanks

How effective is the code blanks approach?

Nearly all students were able to complete the **overall project for course**, which involved **using and adapting code** to create an app to **predict which of two car park buildings to use,** based on data they had collected over time and a decision tree model they had developed.



All the locations explored by 37 students in the first lab

We are currently analysing:

- student interaction data
- weekly reflection answers
- responses to test and exam questions

“With **ChatGPT** so readily available, **any code I've been stuck on I tend to resort to that** through the rest of my degree. But the **coding knowledge required** in these labs was **more of an understanding than an ability to straight write code**, which I think meant that **I engaged with it much more** than I have in many other courses.”

Student reflection from DATASCI 100

Sparking Joy and Discovery in a World of AI?

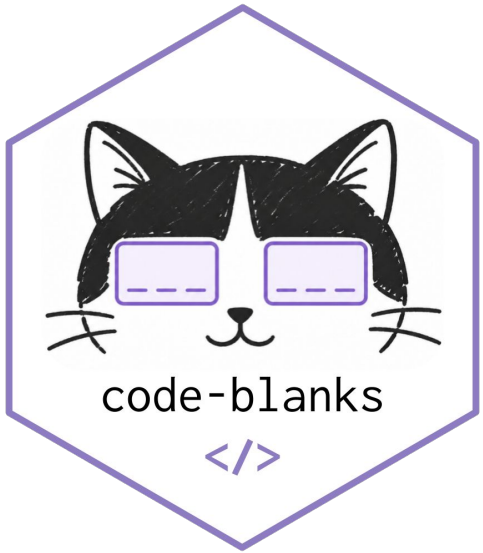
“I found it interesting that there was a clear difference in humidity patterns between Guangzhou and Moscow. **Guangzhou stays humid throughout the year, while Moscow has much more seasonal variation**, especially drier winters. **I used to live in China, so this made a lot of sense. I still remember how constantly humid it felt,** especially in the south. I’ve studied weather-related data before in past courses, but this exercise helped reinforce that knowledge and **reminded me how useful it is to connect data with real-life experiences.**”

“When I did the Final Lab Challenge above I originally **left the start and end date fields blank to see what would happen**, and I thought it was kind of interesting that **something in the background selected a year-long time period for me**. I would have assumed that with no time period selected, I would have created a graph of the entire time period we had data for, but this wasn't the case.”

“I **really enjoyed using R** and coding to make the data visualisations i**nstead of just clicking some buttons.** I found it interesting that **we could write what we want to see and then make it happen.”**

Student reflections from DATASCI 100 Lab task 1A

Links & resources



- Demo code blanks task (DATASCI 100 Lab task 1A)
- {code-blanks} Quarto extension

Learning through tinkering:

A scaffolded approach to introducing programming to data science students by using **code blanks**

Anna Fergusson
University of Auckland | Waipapa Taumata Rau
Aotearoa New Zealand
a.fergusson@auckland.ac.nz

THE UNIVERSITY OF AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND

References

Cunningham, K., Ericson, B. J., , Agrawal Bejarano, R., & Guzdial, M.. (2021). Avoiding the turing tarpit: Learning conversational programming by starting from code's purpose. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–15, Yokohama Japan, May. ACM. **Link to paper**

Fergusson, A.. (2023). Designing positive first experiences with coding for introductory-level data science students. In *Proceedings of the Satellite Conference of the International Association for Statistics Education (IASE) — IASE 2023: Fostering Learning of Statistics and Data Science, Toronto, Canada*. **Link to paper**

Fergusson, A., & Hüsing, S. (2026). Learning through tinkering: Designing interactive worked examples for introductory-level data science students [extended abstract]. *2026 Symposium on Data Science and Statistics, Milwaukee, Wisconsin.* American Statistical Association. **Link to extended abstract**

Renkl, A., Atkinson, R. K., & Große, C.S. (2004). How fading worked solution steps works – a cognitive load perspective. *Instructional Science, 32(*1/2):59–82. **Link to paper**