Tutorial 01 — Instructor notes

ENVX2001 – Applied Statistical Methods

Semester 1

Contents

Companion document for tutors running Tutorial 01. Mirrors each exercise with talking points, expected output at each checkpoint, and common problems.

Student document: Tutorial 01

Before you start

Get these sorted before students arrive:

- Have RStudio open with a fresh session.
- Download all four CSV files from the student tutorial page. You only need one for your demo, but students will pick different ones and ask about theirs.
- Decide which dataset you'll demo with. Ghibli is the default in the student doc, so it's easiest to follow along with.
- Skim the student document if you haven't recently.

Exercise 1: Set up your project

What to do

Demo creating an RStudio Project: **File** > **New Project…** > **New Directory** > **New Project**. Name it `tutorial01-demo` or similar. Create a `data/` folder in the Files pane and move your CSV into it.

What to say

- RStudio Projects set the working directory automatically, so `data/myfile.csv` works without a full path.
- This is how they should set up every lab and assignment — one project per task.
- Show where the `.Rproj` file lives and what it does (sets the working directory, nothing more).

Expected output

The Files pane shows the project folder with a `data/` subfolder containing one CSV file.

Common issues

⚠ Warning

CSV ends up in Downloads instead of `data/`. Students download the file but forget to move it. Show them how to drag the file from their Downloads folder into the `data/` folder. On Windows, the "More > Set As Working Directory" option in the Files pane sometimes confuses things — stick to the system file manager.

⚠ Warning

`Data` instead of `data`. Case matters on macOS and Linux. Not a problem on Windows, but good to keep it lowercase from the start.

Exercise 2: Create your document

What to do

Demo **File** > **New File** > **Quarto Document…** > **Create Empty Document**. Paste in the YAML header, save as `report.qmd`, and render with Cmd+Shift+K (Mac) or Ctrl+Shift+K (Windows/Linux).

CODE

```
---
title: "Exploring My Dataset"
author: "Your Name"
```

```yaml
date: today
format:
  html:
    theme: cosmo
    toc: true
---
```

What to say

- Walk through each YAML field: `title` and `author` are plain text, `date: today` auto-fills the current date, `format` controls the output.
- **Indentation matters** in YAML. `html:` must be indented under `format:`, and `theme:` under `html:`. Two spaces per level, no tabs.
- Render the empty doc first. This shows students the feedback loop early: edit, save, render, check.

Expected output

An HTML page with the title, author name, today's date, and an empty body. The table of contents sidebar is visible but empty.

Common issues

⚠ Warning

YAML indentation errors. The most common problem in this exercise. Students copy-paste the YAML and indentation gets mangled. Tell them to delete and retype the spaces if rendering fails.

⚠ Warning

"Quarto Document" option missing. Student has an old RStudio version or Quarto isn't installed. They need RStudio 2022.07 or later. As a workaround, they can create a plain text file and save it as `.qmd`.

Exercise 3: Write about your data

What to do

Demo adding markdown text below the YAML. Type a heading, some bold text, a bullet list. Use the example from the student doc or make up your own.

What to say

- Show the syntax live: ## for headings, `**bold**`, `*italic*`, `-` for bullets.
- Let students write their own content. Don't dictate what they write — this is their chance to practise markdown without worrying about code.
- Remind them to leave a blank line before headings and between paragraphs. Quarto won't render a heading properly if there's no blank line above it.
- Render after writing to show the table of contents populating.

Expected output

Formatted text with headings, bold, italic, and a bullet list. The table of contents sidebar lists the new sections.

Common issues

⚠ Warning

No blank line before a heading. The heading renders as plain text or merges with the paragraph above. Show what this looks like so students can recognise it next time.

⚠ Warning

Students unsure what to write. Tell them to just describe their dataset: what's in it, what looks interesting, what surprised them. Doesn't need to be long.

Exercise 4: Load and display your data

What to do

Demo adding the setup chunk first (`library(tidyverse)` with `message: false`), then a data-loading chunk. Render and show the output. Pick whichever dataset you like for your demo — the tabs below have the code for all four, but you only need one.

Code for each dataset

Ghibli

```
CODE
library(tidyverse)
my_data <- read_csv("data/ghibli_films.csv")
my_data
```

Columns: `title`, `year`, `rating`, `runtime`

Spotify

```
CODE
library(tidyverse)
my_data <- read_csv("data/spotify_top50.csv")
my_data
```

Columns: `title`, `artist`, `streams_millions`, `genre`

AFL

```
CODE
library(tidyverse)
my_data <- read_csv("data/afl_teams.csv")
my_data
```

Columns: `team`, `wins_2024`, `losses_2024`, `premierships`, `founded`

Fast food

```
CODE
library(tidyverse)
my_data <- read_csv("data/fast_food.csv")
my_data
```

Columns: `name`, `rating`, `price_range`, `locations_au`

What to say

- `library(tidyverse)` loads the packages we need. `read_csv()` comes from readr, which is part of the tidyverse.
- The `message: false` and `warning: false` chunk options hide startup messages. Show what happens without them so students understand why they're useful.
- `read_csv()` uses a relative path from the project root. That's why setting up the project first matters.

Expected output

The setup chunk produces no visible output (messages suppressed). The data chunk prints a tibble showing all rows and columns.

Common issues

⚠ Warning

"could not find function 'read_csv'" Student forgot to run the setup chunk or didn't include `library(tidyverse)`. Show the error so they learn to recognise it.

⚠ Warning

"file does not exist" Wrong file name (capitalisation, typo) or the CSV isn't in `data/`. Have them check the Files pane — the file name there is exactly what they need to type.

⚠ Warning

`read.csv()` instead of `read_csv()` Works fine, but returns a data.frame instead of a tibble. Not worth correcting in week 1 — just mention that `read_csv()` is the tidyverse version and we'll use it through the semester.

Exercise 5: Create a chart

What to do

Demo building a ggplot bar chart. Type it line by line so students can follow along. Render and show the result. Use the same dataset you picked for Exercise 4 — the other variants are here for reference if students ask.

Code for each dataset

Ghibli

CODE

```
ggplot(my_data, aes(x = reorder(title, rating), y = rating)) +
  geom_col() +
  coord_flip() +
  labs(x = NULL, y = "Rating")
```

Spotify

CODE

```
ggplot(my_data, aes(x = reorder(title, streams_millions), y = streams_millions)) +
  geom_col() +
  coord_flip() +
  labs(x = NULL, y = "Streams (millions)")
```

AFL

CODE

```
ggplot(my_data, aes(x = reorder(team, premierships), y = premierships)) +
  geom_col() +
  coord_flip() +
  labs(x = NULL, y = "Premierships")
```

Fast food

CODE

```
ggplot(my_data, aes(x = reorder(name, rating), y = rating)) +
  geom_col() +
  coord_flip() +
  labs(x = NULL, y = "Rating")
```

What to say

- Walk through each line: `aes()` maps columns to axes, `reorder()` sorts the bars, `geom_col()` draws them, `coord_flip()` flips to horizontal, `labs()` sets axis labels.
- Without `reorder()`, bars appear in alphabetical order. Show this if you have time.
- Students using non-Ghibli datasets need to swap column names. Point them to the margin note in the student doc, or to the tabset above.

Expected output

A horizontal bar chart sorted from lowest to highest, with a caption underneath. Quarto auto-numbers the figure because of the `fig-chart` label.

Common issues

⚠ Warning

Wrong column names. Students using Spotify try `rating` instead of `streams_millions`, or AFL students use `name` instead of `team`. The error says "object not found". Tell them to check their column names from the data output in Exercise 4.

⚠ Warning

Forgot to update `labs()`. The chart renders fine but the y-axis says "Rating" when it should say "Streams (millions)" or "Premierships". Not a breaking error, but worth catching.

⚠ Warning

Bars are unsorted. Student wrote `title` instead of `reorder(title, rating)`. The bars appear alphabetically.

Exercise 6: Add a formatted table

What to do

Demo adding a `knitr::kable()` chunk with `echo: false`. Render and show the clean table. Then go back to the chart chunk from Exercise 5 and add `echo: false` there too — render again to show the difference.

CODE
```
knitr::kable(my_data)
```

This code works the same for all four datasets.

What to say

- `echo: false` hides the source code from the rendered output. The reader sees the table but not the R code.
- This is how you'd present results in a report or assignment — code hidden, output clean.
- `knitr::kable()` produces a simple formatted table. Fancier options exist (gt, flextable), but kable is enough for now.
- Show `echo: true` (the default) alongside `echo: false` if you want to make the difference obvious.

Expected output

A formatted table with a caption and no visible code. If the chart chunk also has `echo: false`, the document reads like a finished report.

Common issues

⚠ Warning

Chunk option typo. `echo: False` (capital F) or `echo = false` (equals sign instead of colon). Quarto chunk options use `#| key: value` syntax — lowercase, colon, space.

Putting it all together

What to do

This section is tutor-led. Demo switching the output format live:

1. Change the `format` section of the YAML to Word output:

   ```
   CODE
   format:
     docx:
       toc: true
   ```

   Render and show the Word document.

2. Change it to Typst (PDF) output:

   ```
   CODE
   format:
     typst:
       toc: true
   ```

   Render and show the PDF.

3. Switch back to HTML.

What to say

- Same content, different format. Only the YAML changes.
- Typst is Quarto's newer PDF engine. It doesn't need a LaTeX installation, which saves a lot of setup pain.
- Word output is handy for collaborating with people who don't use R or Quarto.
- Students can list multiple formats in the YAML, but that's a topic for later weeks.

Expected output

The same document rendered as HTML, Word, and PDF. Content is identical; only styling differs.

Common issues

⚠ Warning

YAML indentation when switching formats. Students who copy-paste the format block can end up with mixed indentation. Same fix as before: two spaces, no tabs, `toc:` indented under the format name.