In-Class Problem Set: Reproducible Visualization Workflow (R + GitHub)

Goal. Extend the code from the lecture slides to produce a small, reproducible workflow: load the provided dataset, choose variables using the codebook, create multiple figures using explicit mappings, and submit your work through GitHub.

What to submit (in your GitHub repo).

- A script file: `scripts/lab.R`
- A short write-up: `outputs/writeup.md`
- Saved figures: at least 4 image files in `figures/`
- (Optional but recommended) a log file: `outputs/log.txt`

Rules.

- Work inside an **R Project**.
- Use a **sequential, hard-coded workflow** (no user-defined functions).
- You may consult notes and documentation. If you use any external code, cite it in your write-up.

Questions

1. **Create an R Project (proof required).**
   (a) Create an R Project for this course on your computer.
   (b) **Proof:** In your `outputs/writeup.md`, include:
      - the output of `getwd()` run from inside the project, and
      - a screenshot showing the `.Rproj` file in your project folder *or* the RStudio Project name visible in the RStudio window.
2. **Load the provided dataset from the `data/` folder.**
   (a) Confirm the dataset file exists in `data/`. (Do not manually move it.)
   (b) Write code in `scripts/lab.R` to load it into R as an object named `df`.
   (c) **Proof:** In `outputs/writeup.md`, include:
      - the dimensions of `df` (rows × columns), and
      - the first 3 column names.
3. **Select variables using the codebook.**
   (a) Consult the dataset codebook and choose:
      - **two continuous** variables (numeric, meaningfully ordered with many values), and
      - **two categorical** variables (groups/labels).
   (b) In `outputs/writeup.md`, list the four variables and briefly justify (1 sentence each) why they are continuous vs categorical.
   (c) **Proof:** Include either `str(df)` output for the four variables *or* a small table showing each variable and its class/type.

4. **Create a reproducible folder structure + (optional) logging.**
   (a) Ensure these folders exist in your project:
   - `scripts/`
   - `outputs/`
   - `figures/`
   - `logs/` *(optional but recommended)*

   (b) In `scripts/lab.R`, add code that creates any missing directories (without errors).
   (c) **Proof:** In `outputs/writeup.md`, include `list.files()` output showing the folders.
   (d) **Optional (challenge):** Create a simple log file `outputs/log.txt` that records:
   - the current date/time,
   - the dataset filename loaded,
   - and the names of the four selected variables.

5. **Make three plot extensions + comment on them.**
   Using the lecture code as your baseline, create **three** distinct extensions (three separate figures). Each figure must include a caption in your write-up that explains:
   - what variables are mapped to what visual properties,
   - what comparison is easiest to make,
   - and one default choice you are accepting (or changing) and why.

   Your three extensions must come from different categories below (choose any three):
   (a) **Add a mapping:** map a categorical variable to `color` or `shape`.
   (b) **Change the mark:** switch to a different geometry appropriate for the variable types (e.g., boxplot for outcome vs group).
   (c) **Add an annotation layer (lightweight):** add a title + axis labels + a one-sentence caption in the write-up.
   (d) **Handle overplotting:** use transparency (`alpha`) and briefly explain why.
   (e) **Re-order categories:** reorder a categorical axis to improve interpretability (explain the ordering rule).

   Saving requirement: Save each plot to `figures/` using `ggsave()` (do not rely on screenshots). Name files clearly (e.g., `figures/plot1.png`, `figures/plot2.png`, `figures/plot3.png`).

6. **Public vs expert visualization + GitHub submission (proof required).**
   (a) Create **two** versions of the *same* visualization:
   - one intended for a **general public** audience, and
   - one intended for an **expert** audience.

   (b) In `outputs/writeup.md`, state at least **three decision rules** you used to adapt the design (e.g., labeling density, uncertainty/context, annotation, choice of scale, what to simplify vs keep).
   (c) **Challenge ideas (pick one):**
   - Add a short "limitations" note for the public version (1–2 sentences).
   - Add a technical note for the expert version describing a key default or transformation.
   - Add a small multiple (two panels) for the expert version only.
   - Add a deliberately "bad" version and write 3 bullets on why it misleads.

   (d) **GitHub requirement:** Commit and push your work.
   Proof: In `outputs/writeup.md`, include:
   - the output of `git status` *after* committing (showing a clean working tree), and
   - either a screenshot of your GitHub repo showing the latest commit *or* the commit hash and message.

Checklist (before you leave)

- `scripts/lab.R` exists and runs top-to-bottom
- `outputs/writeup.md` exists and includes required proofs
- At least 4 figures saved in `figures/` (3 extensions + 2 audience versions can overlap if you clearly label)
- Work is pushed to GitHub