In-Class Problem Set: Color Encodings with Senate Ideology Data (R + GitHub *or* Canvas)

Goal. Practice using color intentionally (not decoratively) by visualizing ideology in the U.S. Senate across four time points. You will (i) obtain the data from the course materials (GitHub *or* Canvas), (ii) subset to the target Senates, (iii) make one plot for each time point with color encoding ideology, (iv) interpret what you see, and (v) submit via **GitHub *or* Canvas**.

What to submit (GitHub *or* Canvas).

- A script file: `scripts/lab.R`
- A short write-up: `outputs/writeup.md`
- Four figures (one per time point) saved to `figures/`

If you submit via Canvas, upload the same files listed above as individual files (or upload a single zip that preserves the folder structure).

Rules (read carefully).

- Work inside an **R Project**.
- Use a **sequential, hard-coded workflow** (no user-defined functions).
- Your plots must:
  - set a non-default plot background color,
  - use colors that are intuitive for the task,
  - use an accessibility-conscious palette (colorblind-friendly),
  - and **explicitly justify your color scale choice** (sequential vs diverging) in your write-up.
- Save outputs using `ggsave()` (no screenshots).
- If you choose the **GitHub submission option**, Git commands go in the **Terminal tab** (not the R Console).

Submission options

You may submit this assignment using **either** of the following methods:

- **GitHub submission (recommended):** Commit and push your work to your GitHub repository. You will include Git proof (`git status` and `git log`) in your write-up.
- **Canvas submission:** Upload the required files directly to Canvas. You do *not* need to use GitHub if you choose this option.

Both submission methods are graded using the same rubric.

Mini codebook (use this; do not guess)

- **Which Senates to use.** For this problem set, use the following Congress numbers to represent the four target time points:

  - **1990 → 101st Congress**
  - **2000 → 106th Congress**
  - **2010 → 111th Congress**
  - **2020 → 116th Congress**
- **What DW-NOMINATE is.** DW-NOMINATE is an ideology scaling procedure based on roll-call voting. In most datasets:
  - `dwnom1` is the primary (left–right) ideological dimension,
  - `dwnom2` is a secondary ideological dimension (often smaller and context-dependent).
- **What you will plot.** Your scatterplots should use the two ideology dimensions (typically `dwnom1` on x and `dwnom2` on y), and **color points by ideology** (typically `dwnom1`) unless your course codebook specifies a different ideology column.
- **Color scale rule (required).**
  - If the ideology variable has meaningful sign around 0 (e.g., negative vs positive), use a **diverging** scale centered at 0.
  - If you treat ideology as magnitude only (no meaningful center), use a **sequential** scale.

  In your write-up, state which rule you used and why.

Questions

1. **Get the Senate ideology data (proof required).**
   (a) Choose **one** method:
   - **GitHub option:** In the **Terminal tab**, run:

     ```
     git status
     git pull
     ```
   - **Canvas option:** Download the Senate ideology file from Canvas and place it in your project `data/` folder.

   (b) Confirm the Senate ideology file exists in your project (the exact file name/path is in the course materials; for most of you it will be in `data/`).
   (c) **Pseudo-code (do not make this perfect; fill in blanks):**

   ```
   # confirm you are in the project root
   __________()

   # list files in the data folder
   __________("data")

   # (optional) search for the Senate file name pattern
   # _________("data", pattern="_____")
   ```

   (d) **Proof (write-up):** In `outputs/writeup.md`, paste:
   - the output of `getwd()` from inside your R Project, and
   - the output of `list.files("data")` showing the Senate file.
2. **Set up your reproducible workflow (folders + script).**
   (a) Ensure your project contains these folders (create them if missing):
   - `scripts/`
   - `outputs/`
   - `figures/`

   (b) Create a script named `scripts/lab.R`. All code for this problem set must live in this script.

(c) **Pseudo-code (structure only):**

```
# from the project root:
__________("scripts")
__________("outputs")
__________("figures")

# confirm what is in the root folder
__________()
```

(d) **Suggested edit (important):** At the top of `scripts/lab.R`, include:

- a short header comment describing what the script does,
- `library(...)` calls,
- `set.seed(123)`.

(e) **Pseudo-code (script header):**

```
# ---------------------------------
# Script name: __________
# Purpose: ________________________
# ---------------------------------

library(__________)
library(__________)

set.seed(_____)
```

(f) **Proof (write-up):** paste the output of `list.files()` from your project root.

3. **Load and summarize the dataset.**

(a) In `scripts/lab.R`, load the dataset into an object named `df`.

(b) **Pseudo-code (intentionally incomplete):**

```
# load data (fill in filename)
df <- read.csv("data/__________.csv")

# quick checks (pick at least two)
dim(df)
names(df)
head(df)

# check the ideology columns exist
# summary(df$______)
# summary(df$______)
```

(c) Summarize the dataset in a way that supports your next steps. At minimum include:

- `dim(df)`
- `names(df)`
- a focused summary of the key columns you will use (time/Congress, chamber, ideology).

(d) **Proof (write-up):** Report:

- number of rows and columns,
- the column you will use for Congress/time,
- the column you will use to identify the Senate chamber (if applicable),

- the ideology columns you will use (e.g., `dwnom1`, `dwnom2`).

4. **Subset to the four target Senates (required).**
   (a) Subset the data so it contains only **Senate** observations for the following Congresses:

$$\{101, 106, 111, 116\}.$$

   (b) Save the subset as `df4`.
   (c) **Pseudo-code (leave blanks):**

```
# choose the Congress variable name
# unique(df$_______)

df4 <- df %>%
  filter(_________ == "Senate") %>%     # or the correct chamber coding
  filter(_________ %in% c(101, 106, ____, ____))

# quick checks
table(df4$_________)      # Congress counts
# table(df4$_________)    # chamber counts (if applicable)
```

   (d) **Proof (write-up):** Include counts that confirm:
   - only these four Congresses appear in `df4`, and
   - `df4` contains only Senate observations (not House).

5. **Make four plots (one per Congress), with color encoding ideology (required).**
   For each of the four Congresses, create a scatterplot using the two DW-NOMINATE dimensions:
   - x-axis: `dwnom1`
   - y-axis: `dwnom2`
   - color: ideology (typically `dwnom1`)

   Required design constraints (integrated). Each plot must:
   - set a non-default plot background color,
   - use an accessibility-conscious palette,
   - use a color scale that is **intuitive for the task**:
     - **diverging centered at 0** if ideology sign matters, or
     - **sequential** if you treat ideology as magnitude only,
   - label axes and the legend clearly (what the variable is).

   Save figures to `figures/` with clear names, for example:
   - `figures/senate_101.png`
   - `figures/senate_106.png`
   - `figures/senate_111.png`
   - `figures/senate_116.png`

   Pseudo-code (intentionally incomplete; do not perfect this):

```
# pick ONE Congress first, then repeat for others
# congress_id <- ____

p <- ggplot(df4 %>% filter(_________ == ____),
            aes(x = ________, y = ________, color = ________)) +
  geom_point(__________) +
  labs(title = "Senate ideology: ____ Congress",
```

```
      x = "________", y = "________", color = "________") +
  theme_minimal() +
  theme(plot.background = element_rect(fill = "________", color = NA))

# choose a palette + scale type (diverging or sequential)
# scale_color___________(...)

ggsave("figures/senate____.png", p, width = ____, height = ____)
```

6. **Interpretation (write-up required).**
   In `outputs/writeup.md`, write 10–14 sentences answering:
   - What does color represent in your plots (which variable, which direction, what range)?
   - Compare the earliest vs latest Congress in your set: what changed in ideological separation and dispersion?
   - **Color-scale justification (required):** Did you use a sequential or diverging scale? Why does that choice match the meaning of ideology in your plot?
   - Accessibility: state one concrete decision you made to improve accessibility (palette choice, contrast with background, labeling).
7. Choose **one** of your four Senate plots to evaluate.
   - Upload the image file to a color-blindness testing website. You may use any reputable tool, such as: `https://www.color-blindness.com/coblis-color-blindness-simulator/`, or another simulator discussed in class
   - Examine how your plot appears under at least **two** simulated conditions (e.g., deuteranopia, protanopia, tritanopia).
   - If the plot is **not** clearly readable under at least one simulation, revise your color scale and re-export the plot. You may:
     - change the palette,
     - adjust the midpoint or limits of the scale,
     - change the background color,
     - or adjust point size / opacity.
   - Save the final version (even if unchanged) using the *same filename* so it overwrites the original plot.
   - **Write-up (required):** In `outputs/writeup.md`, write 6–9 sentences describing:
     - which plot you tested and which simulator you used,
     - which color-vision deficiencies you examined,
     - what problems (if any) you observed,
     - what changes you made *or* why you decided no change was necessary,
     - and one general lesson about color accessibility you learned from this process.
8. **Submit your work (GitHub *or* Canvas) + proof required.**
   (a) **Choose ONE submission method:**
   - **GitHub option:** Commit and push your work to GitHub.
   - **Canvas option:** Upload `scripts/lab.R`, `outputs/writeup.md`, and the four figure files from `figures/` to Canvas.

   (b) **Pseudo-code (submission skeleton; fill in blanks):**

```
# GitHub option (Terminal tab)
git status
git add ______
git commit -m "__________"
```

`git push`

(c) **Proof (write-up):**
- If using **GitHub**: paste `git status` (clean working tree) and `git log -1`.
- If using **Canvas**: paste `list.files("scripts")`, `list.files("outputs")`, `list.files("figures")`, and write one sentence stating you submitted via Canvas.

Optional challenge (if you finish early)

Create a second version of one Congress plot that changes exactly **one** design element:
- Switch sequential ↔ diverging (and explain why the alternative is worse), *or*
- Keep the same palette but change the background color and explain how contrast changes readability.

In 5–7 sentences, argue which version is better for (i) a general audience and (ii) an expert audience.

Checklist (before you leave)

- `scripts/lab.R` exists and runs top-to-bottom
- `outputs/writeup.md` exists and includes interpretation + proofs
- Four figures saved in `figures/` (101, 106, 111, 116)
- Work is either committed and pushed to GitHub *or* uploaded to Canvas