In-Class Problem Set: Distributions, Q–Q Plots, and Faceting (R + GitHub)

Goal. Use real conflict data to practice diagnosing distributional shape using histograms, density plots, and Q–Q plots, including comparisons across groups using faceting. You will pull the data from GitHub, build a reproducible workflow, generate required plots, interpret what they show, and submit via GitHub.

Dataset. `battle_deaths` (1141 rows; 6 variables):

- `iso2c` (country code), `country`, `year`
- `battle_deaths` (numeric; battle-related deaths)
- `region` (categorical), `income` (categorical)

What to submit (in your GitHub repo).

- A script file: `scripts/lab.R`
- A short write-up: `outputs/writeup.md`
- Saved figures in `figures/` (see requirements below)

Rules.

- Work inside an **R Project**.
- Use a **sequential, hard-coded workflow** (no user-defined functions).
- Save figures using `ggsave()` (no screenshots).
- Git commands must be run in the **Terminal tab**, not the R Console.
- Unless your lecture explicitly did otherwise, treat missing values defensibly (state what you did).

Questions

Code snippet expectation. For Questions 2–6, include **small R snippets (2–4 lines)** near the relevant instructions as reminders of correct syntax. These are not full solutions—just scaffolding.

1. **Pull the data and set up your workflow (proof required).**
   (a) In the **Terminal tab**, run:

   ```
   git status
   git pull
   ```

   (b) Confirm the dataset file exists in your repo (path posted in the course repository).
   (c) Create the standard folder structure (if missing): `scripts/`, `outputs/`, `figures/`.
   (d) **Proof (write-up):** In `outputs/writeup.md`, paste:
   - the output of `getwd()`,
   - the output of `list.files()` from the project root, and
   - the output of `list.files("data")` showing the dataset file.

   R snippet (for proofs in your write-up):

```
getwd()
list.files()
list.files("data")
```

2. **Load and summarize `battle_deaths`.**
   (a) Load the dataset into an object named `df`.
   (b) Verify the key columns exist: `country`, `year`, `battle_deaths`, `region`, `income`.
   (c) Summarize the distribution of `battle_deaths` (min/median/mean/max is sufficient).
   (d) **Proof (write-up):** Report:
      - number of rows and columns,
      - the number of unique countries,
      - the range of years,
      - a short summary of `battle_deaths`.

   R snippet (load + quick checks):

   ```
   df <- read.csv("data/battle_deaths.csv")
   names(df)
   summary(df$battle_deaths)
   ```

   R snippet (proof statistics):

   ```
   dim(df)
   length(unique(df$country))
   range(df$year, na.rm = TRUE)
   ```

3. **Histogram of battle deaths (baseline).**
   Create a histogram of `battle_deaths`.
   - You must make a clear binning choice and state it (binwidth or number of bins).
   - If you restrict the x-axis (e.g., to reduce the influence of extreme values), you must state the rule you used.

   R snippet (histogram + save):

   ```
   p_hist <- ggplot(df, aes(x = battle_deaths)) +
    geom_histogram(bins = 40) +
    theme_classic()
   ggsave("figures/battle_deaths_hist.png", p_hist, width = 7, height = 5)
   ```

   Save as:

   `figures/battle_deaths_hist.png`

4. **Density plot of battle deaths (baseline).**
   Create a density plot of `battle_deaths`.
   - Use the same x-axis limits as your histogram (so the two are comparable).
   - Label the axes clearly.

   R snippet (density + same x limits idea):

   ```
   p_den <- ggplot(df, aes(x = battle_deaths)) +
   geom_density() + coord_cartesian(xlim = c(0, 5000)) +
   theme_classic()
   ggsave("figures/battle_deaths_density.png", p_den, width = 7, height = 5)
   ```

Save as:

`figures/battle_deaths_density.png`

5. **Q–Q plot (normality check).**
   Create a Q–Q plot comparing `battle_deaths` to a theoretical normal distribution.
   - Include a Q–Q reference line.
   - In your write-up, describe what kind of deviation you see (e.g., heavy right tail, skew).

   R snippet (correct ggplot Q–Q syntax):

```
p_qq <- ggplot(df, aes(sample = battle_deaths)) +
stat_qq() +
stat_qq_line() +
theme_classic()
ggsave("figures/battle_deaths_qq.png", p_qq, width = 7, height = 5)
```

   Save as:

   `figures/battle_deaths_qq.png`

6. **Faceting by income and region (required).**
   Create two sets of faceted distribution plots:
   (a) A faceted plot by **income**
   (b) A faceted plot by **region**
   For each set, choose **one** distribution geometry that was covered in lecture (histogram or density) and facet it. Your goal is to compare how distributional shape differs across groups.

   Required:
   - Use consistent axis limits across facets (so comparisons are meaningful).
   - If some facets are too sparse to interpret, state what you did (e.g., dropped very small groups, or noted limitations).

   R snippet (facet by income):

```
p_inc <- ggplot(df, aes(x = battle_deaths)) +
geom_histogram(bins = 30) +
facet_wrap(˜ income) +
theme_classic()
ggsave("figures/facet_income.png", p_inc, width = 9, height = 6)
```

   R snippet (facet by region):

```
p_reg <- ggplot(df, aes(x = battle_deaths)) +
geom_histogram(bins = 30) +
facet_wrap(˜ region) +
theme_classic()
ggsave("figures/facet_region.png", p_reg, width = 9, height = 6)
```

   Save as:
   - `figures/facet_income.png`
   - `figures/facet_region.png`

7. **Interpretation (write-up required).**
   In `outputs/writeup.md`, write 12–16 sentences addressing:

- What do the histogram and density plot suggest about skew and tail behavior?
- What does the Q–Q plot reveal (and why is it useful here)?
- Compare distributions by income: what differences (if any) stand out?
- Compare distributions by region: what differences (if any) stand out?
- Name one concrete plotting choice you made (bins, limits, faceting) and why it helped interpretability.

8. **Submit your work (GitHub or Canvas; proof required).**
   (a) Choose **one** submission path:
   - **GitHub path:** In the **Terminal tab**, run:

     ```
     git status
     git add .
     git commit -m "Distributions lab: battle_deaths faceting + QQ"
     git push
     ```

   - **Canvas path:** Upload `scripts/lab.R`, `outputs/writeup.md`, and required files from `figures/` to Canvas.

   (b) **Proof (write-up):** Paste:
   - if using GitHub: the output of `git status` after committing (clean working tree) and `git log -1`,
   - if using Canvas: a short note confirming upload date/time and the list of uploaded files.

Optional challenge (if you finish early): ggridges

Create a ridgeline density plot using `ggridges` for `battle_deaths` grouped by **income** *or* **region**.

- Save as `figures/ridgeline.png`.
- In 4–6 sentences, explain what the ridgeline plot makes easier (or harder) to compare relative to faceting.

R snippet (ridgeline starter):

```
library(ggridges)
ggplot(df, aes(x = battle_deaths, y = income)) + geom_density_ridges() + theme_classic()
```

Checklist (before you leave)

- `scripts/lab.R` runs top-to-bottom
- Required figures exist in `figures/`:
  - `battle_deaths_hist.png`, `battle_deaths_density.png`, `battle_deaths_qq.png`
  - `facet_income.png`, `facet_region.png`
- `outputs/writeup.md` includes interpretation + proofs
- Work is submitted (either pushed to GitHub or uploaded to Canvas)