

In-Class Problem Set: Exploring Movie Data with Distribution and Color (R + GitHub)

Goal. Use real movie data to practice visualizing distributions, comparing groups, and encoding multiple variables in a single plot. You will pull the dataset from GitHub, build a reproducible workflow, generate several plots, interpret what they show, and submit your work via GitHub.

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 code (`ggsave`); do not use screenshots.
- Git commands must be run in the **Terminal tab**, not the R Console.

Mini codebook (use this; do not guess)

Each row represents one movie. Relevant variables include:

- `budget`: Production budget in USD (0 if unavailable).
- `revenue`: Worldwide box office revenue in USD (0 if unavailable).
- `director`: Director of the movie.
- `runtime`: Movie length in minutes.
- `vote_average`: Average user rating (0–10).
- `vote_count`: Number of user votes.
- `popularity`: Popularity score based on user engagement.

Questions

1. Pull the movie dataset from GitHub (proof required).

- (a) In the **Terminal tab**, run:

```
git status  
git pull
```

- (b) Confirm the movie dataset exists in your repo (location specified in the course GitHub).

- (c) **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 movie file.

2. Load and summarize the dataset.

- (a) Load the movie dataset into an object named `df`.
- (b) Summarize the dataset to understand its structure.

Suggested edit: Use `dim(df)`, `names(df)`, and a focused summary of `budget` and `revenue`.

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

- number of rows and columns,
- the range of `budget`,
- the range of `revenue`.

3. Plot distributions: movie budget and revenue.

Create two histograms:

- one for `budget`,
- one for `revenue`.

Suggested edit (important):

- Use consistent bin widths.
- Decide whether to include or exclude zero values, and state your choice.

Save the plots as:

- `figures/budget_hist.png`
- `figures/revenue_hist.png`

4. Identify top-grossing directors and compare revenue.

- (a) Identify the **top three directors** by total box office revenue (sum of `revenue`).
- (b) Subset the data to movies directed by these three directors.
- (c) Create a **boxplot** showing the distribution of `revenue` for each director.

Save the plot as:

`figures/revenue_by_director.png`

Suggested edit: Ensure the director names are readable and the y-axis is clearly labeled in USD.

5. Scatter plot with size and category encodings.

Create a scatter plot with:

- x-axis: `budget`
- y-axis: `revenue`

Then:

- Choose **one additional quantitative variable** (e.g., `popularity` or `vote_count`) and map it to **point size**.
- Choose **one categorical variable** (e.g., `original_language` or a simplified genre indicator) and map it to **color**.

Suggested edit:

- Use a color palette that makes category differences clear.
- Avoid using size in a way that hides smaller-budget films.

Save the plot as:

`figures/budget_revenue_scatter.png`

6. Interpretation (write-up required).

In `outputs/writeup.md`, write 10–14 sentences addressing:

- What do the budget and revenue histograms reveal about the movie industry?
- How do revenues differ across the top-grossing directors?
- In the scatter plot, what relationships are most visually salient?
- How do size and color encodings change what is easy or hard to see?

7. Push your work to GitHub (proof required).

- (a) In the **Terminal tab**, run:

```
git status  
git add .  
git commit -m "Movie data visualization lab"  
git push
```

- (b) **Proof (write-up):** Paste:

- the output of `git status` after committing (clean working tree), and
- the output of `git log -1`.

Optional challenge (if you finish early)

Choose one plot and create an alternative version optimized for a **general public** audience rather than an expert audience. In 5–7 sentences, explain:

- what design choices you changed,
- what information you simplified or emphasized,
- and why these changes are appropriate for a public-facing visualization.

Checklist (before you leave)

- `scripts/lab.R` runs top-to-bottom
- `outputs/writeup.md` exists and includes interpretations + proofs
- Required figures exist in `figures/`
- Work is committed and pushed to GitHub