

Universal ggThemeAssist: An Interactive Shiny App for Custom ggplot2 Theming

Group 16

Data Visualization Project 2

June 5, 2025

Outline

- 1 Introduction
- 2 Approach & Technologies
- 3 App Structure
- 4 Demonstration
- 5 Discussion & Limitations
- 6 Conclusion

Introduction

Motivation

- Crafting publication-quality ggplot2 plots often requires manual tweaking of theme parameters.
- Users iterate through theme settings (font size, background, grid, colors) by editing code and re-running plots.
- Existing GUI tools or prebuilt themes are limited:
 - Cannot save/load custom configurations on-the-fly.
 - Lack reproducible guidance for consistent workflows.
- **Goal:** Provide an interactive Shiny app that:
 - Allows uploading any CSV dataset.
 - Offers flexible geometry and theming controls.
 - Persists user-defined theme configurations (load/save/delete).
 - Guides users toward reproducible plots.

Approach & Technologies

Why Shiny?

- **Reactivity:** Instantaneous plot updates when inputs change.
- **Widgets:** `selectInput`, `sliderInput`, `checkboxInput`, `fileInput` provide dynamic UI.
- **Ease of Deployment:** Cross-platform and runs in any modern web browser.
- **User Experience:** No need to re-run scripts—users interact with controls directly.

Why ggplot2?

- Grammar-of-graphics paradigm: intuitive layering (`ggplot()` + `geom_*()` + `theme_*()`).
- Users familiar with `ggplot2` can extend app-generated visuals.
- Built-in theme functions (`theme_minimal`, `theme_dark`, `theme_grey`, etc.).
- Easy to apply additional plot transformations (log scales, coordinate flip).

JSON for Persistence

- Store custom theme configurations as JSON in `saved_themes.json`.
- `jsonlite` maps R lists \leftrightarrow JSON effortlessly.
- Enables loading, deleting, and sharing of theme files.
- Human-readable and cross-language compatible.

App Structure

Key Components: Helper Functions

`save_theme_to_file(name, config)`

- Reads existing JSON (if any), adds/overwrites `themes[[name]] = config`, writes back.
- Returns `TRUE` on success, `FALSE` on error.

`load_themes_from_file()`

- Parses `saved_themes.json` into an R list; returns empty list if file missing or invalid.

`delete_theme_from_file(name)`

- Removes the entry `themes[[name]]`, rewrites JSON; returns `TRUE/FALSE`.

Key Components (cont.)

`apply_theme_config(conf, session)`

- Updates Shiny inputs to match `conf$base_theme`, `conf$base_size`, `conf$show_grid`, `conf$flip_coords`.
- Uses `updateSelectInput`, `updateSliderInput`, `updateCheckboxInput`.

`generate_geom(input, colorvar, fillcol)`

- Returns appropriate `geom_point`, `geom_line`, `geom_col`, `geom_density`, or `geom_histogram` based on user inputs.
- Handles manual color/fill if `colorvar == "none"`.

We will now switch to a live demonstration of the app.

Discussion & Limitations

Discussion of Results

- **Interactive Workflow:**

- Users iteratively adjust theme settings without re-running entire scripts.
- Real-time feedback fosters experimentation with fonts, colors, and layout.

- **Reproducibility:**

- The app logic ensures that themes and plot settings can be reproduced via Shiny code.

- **Theme Library:**

- Saving and loading themes enables consistent styling across multiple figures or team members.

- **Performance:**

- Responsive for small-to-medium datasets (up to tens of thousands of rows).
- Larger datasets may incur slight delays in UI rendering.

Limitations (Part 1)

- **Column Name Constraints:**

- `aes_string()` requires syntactically valid names (no spaces or leading numbers).
- Users must ensure data column names are R-friendly or sanitize them manually.

- **Limited Geometry Options:**

- Only five geoms are available: scatter, line, bar, density, histogram.
- No built-in support for boxplots, violin plots, faceting, or multiple layers.

Limitations (Part 2)

- **Static Theme Elements:**

- Cannot adjust axis text angles, legend positions, facet styles, or subtitle fonts via UI.
- Users must edit underlying code for fine-grained theming.

- **Concurrency & Storage:**

- Multiple R sessions writing to `saved_themes.json` can cause race conditions.
- A file-locking or database-based solution could mitigate this.

- **Fixed Output Options:**

- Download always produces a PNG (800x720; 72 dpi).
- No UI control for PDF, SVG, or custom DPI/dimensions.

Future Enhancements (Part 1)

1 Expanded Geometry Support

- Add `geom_boxplot`, `geom_violin`, `geom_smooth`, facetting (`facet_wrap`, `facet_grid`).
- Allow layering multiple geoms (e.g., scatter + smoothing).

2 Advanced Theme Controls

- Expose axis text angles, legend position, subtitle font size.
- Provide color palette picker (e.g., `viridis`, `RColorBrewer`).
- Let users adjust panel/grid colors interactively.

Future Enhancements (Part 2)

3 Robust Persistence

- Migrate from JSON to SQLite for concurrency safety.
- Version-control themes: store a timestamp and optional description per theme.
- Export/import theme collections (JSON or SQL dump).

4 Flexible Output Formats

- Allow users to choose PNG/PDF/SVG and specify resolution/dimensions on-the-fly.
- “Save Plot Package” option: generate a self-contained archive bundling the plot and theme metadata.

5 Enhanced Data Handling

- Support different delimiters (tab, semicolon) and Excel files (.xlsx).
- Provide a module for converting columns to numeric/factor types as needed.
- Use DT for interactive data tables with large datasets.

Conclusion

Conclusion

- **Universal ggThemeAssist** bridges the gap between writing custom ggplot2 theme code and using a graphical interface.
- Users gain:
 - Rapid, interactive theme experimentation via Shiny's reactivity.
 - The ability to save, share, and reload custom themes using JSON.
 - Consistent, publication-quality plots without manual code rewrites.
- By combining Shiny, ggplot2, and JSON persistence, the app provides an efficient workflow for creating polished data visualizations.
- Future enhancements will expand geometry support, expose more theme elements, and improve persistence mechanisms.

Thank you!

Questions?