

# Project Proposal: Shiny Theme App – A User-Friendly Interface for Customizing ggplot2 Themes

Bui Huy Linh Phuc, Le Gia Duc, Ta Viet Thang

## High-Level Goal

To develop a Shiny web application that allows users to interactively change the theme and visual styling of any ggplot2 chart without needing to write code.

## Project Description and Motivation

While ggplot2 provides extensive options for customizing plot themes (e.g., minimal, classic, dark, light), users often find it tedious and unintuitive to tweak appearance manually via code, especially those new to R or data visualization.

This project aims to create an easy-to-use Shiny app where users can:

- Upload or generate a ggplot2 plot directly within the app.
- Select from preset themes (e.g., `theme_minimal()`, `theme_classic()`, `theme_dark()`).
- Adjust key theme elements interactively, such as font size, font family, plot background color, panel grid line visibility, axis text angle, and legend position.
- Preview changes live as they make adjustments.
- Download the customized plot as an image or save the underlying R code for reproducibility.

The app will feature an intuitive sidebar for controls and a main panel for real-time plot rendering, following best practices in Shiny UI design. Advanced users may also be given the option to fine-tune specific elements like title styling, axis formatting, and legend formatting.

This project is exciting because it bridges the gap between technical flexibility and user experience, making powerful R visualizations more accessible to non-programmers. It also introduces reactive programming concepts in Shiny and reinforces best practices in UI/UX design for data science tools.

## Weekly Plan and Team Member Tasks

- **Week 1:** Brainstorm app features, set up repository, install tools, submit proposal
  - Brainstorm app features: sample plots, theme selection (`theme_minimal`, `theme_classic`), font size adjustment (All)
  - Set up GitHub repository (Phuc)
  - Install R, Shiny, ggplot2 (Phuc)
  - Prepare sample dataset (Duc)
  - Draft and submit project proposal outlining app scope and deliverables (Thang)

*Deliverable:* GitHub repo, submitted project proposal.

- **Week 2:** Conduct peer reviews, address feedback, start coding app
  - Conduct peer reviews for two other teams' proposals via GitHub issues (Phuc, Duc)
  - Address feedback on own repo through commits (Phuc, Duc)
  - Create Shiny app with file upload (Duc)
  - Add theme dropdown (Phuc)
  - Draft presentation slides outline (Duc)
  - Draft report outline (Phuc)

*Deliverable:* Completed peer reviews, updated repo with feedback fixes, initial app code.

- **Week 3:** Revise proposal, add customization features
  - Review teaching team's feedback on proposal and resubmit revised version (All)
  - Add font size slider (Phuc)
  - Add legend position dropdown (Duc)
  - Continue drafting presentation slides (Thang)
  - Continue drafting report (Thang)

*Deliverable:* Revised proposal, app with customization features, draft slides and report.

- **Week 4:** Add download, style UI, advance deliverables

- Implement download button for plot as PNG (Thang)
- Apply basic UI styling with `shinythemes` (Thang)
- Draft presentation slides (Thang)
- Draft project report (Thang)
- Test app functionality with sample dataset (Thang)

*Deliverable:* App with download and basic styling, draft slides and report.

• **Week 5:** Finalize app, complete deliverables, submit package

- Finalize Shiny app functionality for upload, plot, and customization (All)
- Clean up code and add comments (Phuc)
- Update repo README (Phuc)
- Complete presentation slides (Thang)
- Complete project report (Duc)
- Submit project package (code, slides, report) (All)

*Deliverable:* Final app in repo, presentation slides, project report.