

ExcelToShiny: An R Package for Automating Shiny Dashboard Creation from Excel Templates

Lily Clements ¹

¹ IDEMS International

DOI: [10.21105/joss.07876](https://doi.org/10.21105/joss.07876)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Sehrish Kanwal](#)  

Reviewers:

- [@YaoxiangLi](#)
- [@devarops](#)

Submitted: 13 September 2024

Published: 21 July 2025

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

ExcelToShiny is an open-source, user-friendly tool that automates the creation of Shiny dashboards from structured Excel templates. Designed to simplify the process of building interactive web applications in R, ExcelToShiny allows users to convert Excel spreadsheets directly into customisable, fully functional dashboards with minimal coding. This solution promotes reproducibility and transparency while significantly reducing the technical barrier for users seeking data-driven insights through visualisations.

By automating the traditionally labour-intensive process of dashboard generation, ExcelToShiny ensures that even non-technical users can engage with dynamic data visualisations. The package plays a role in promoting accessibility and flexibility in data presentation, enabling users to focus on decision-making rather than complex programming tasks.

Statement of Need

Interactive dashboards provide a dynamic platform to explore and communicate research data, making it easier to identify patterns, trends, and anomalies. These insights can be acted upon in real-time, supporting decision-making for a current ongoing experiment, data interpretation, or in subsequent research phases ([Dutta, 2024](#)). However, constructing these dashboards from scratch in Shiny can be complex even for those with experience in coding in R ([Konrad, 2020](#)). For non-technical users, or those unfamiliar with Shiny, this challenge often leads to reliance on static reports or external proprietary tools like Tableau and Power BI ([Koldus, 2024](#)). While these tools are user-friendly, they come with limitations, such as cost and a lack of reproducibility ([Microsoft Corporation, 2024](#); [Tableau Software, 2024](#)).

Researchers can face difficulties in creating reproducible workflows for interactive data visualisations, especially when technical expertise is a barrier. Static reports limit data exploration ([Koldus, 2024](#)), and proprietary software tools, while convenient, are not as easily adaptable or transparent enough for scientific research needs ([Biswas, 2021](#)).

ExcelToShiny addresses these gaps by offering an R-based alternative that automates the conversion of Excel templates into Shiny dashboards. By reading a user-defined Excel structure, ExcelToShiny dynamically generates the corresponding UI and server components for a Shiny application. This approach allows users to create dashboards that are reproducible, transparent, and easy to share, with the flexibility to modify and adapt them as needed within the R ecosystem. Additionally, by lowering the technical barrier, ExcelToShiny allows researchers from various fields to engage with large datasets, perform real-time analysis, and generate data-driven insights without the need for extensive coding.

By streamlining the creation of Shiny dashboards, ExcelToShiny directly supports research activities by making it easier for researchers to interact with data, visualise results, and share reproducible analysis in a transparent, flexible format.

State of the Field

Several R packages exist for creating Shiny applications, each designed for different use cases and levels of technical expertise. For instance, `golem` promotes modularisation and best practices for large-scale applications but requires a deeper understanding of Shiny's architecture. Similarly, `flexdashboard` enables the creation of interactive dashboards with minimal explicit Shiny code, though it relies on RMarkdown and R, which can be a barrier for users with limited programming experience.

While these tools are powerful, they introduce complexities that may not suit all users. In contrast, `ExcelToShiny` lowers the entry barrier by leveraging structured Excel templates, making it particularly well-suited for those who prefer spreadsheet-based interfaces for data management and visualisation. By bridging the gap between non-technical users and interactive dashboard creation, `ExcelToShiny` offers a more accessible alternative to traditional Shiny development approaches.

Functionality and Features

`ExcelToShiny` leverages a simple, structured Excel format to automatically generate Shiny dashboards, making the process highly accessible. The package interprets Excel template elements like tabs, graphs, tables, and menus, converting them into Shiny UI components and server logic with minimal user input. The dashboards can have regular display pages, display pages with tabs, or have a page that downloads the data. Furthermore, while much of the process is automated, users can further customise their dashboards, enabling fine-tuning of both layout and behaviour to meet specific requirements without needing to dive into R code.

Finally, the use of structured templates ensures that dashboards can be easily replicated, modified, and audited, enhancing transparency in data-driven processes. The package integrates with existing R workflows, making it straightforward to incorporate into larger projects, combining it with other packages for a comprehensive analysis.

Usage

`ExcelToShiny` is simple to use, with the workflow beginning from an Excel template that defines the dashboard's layout. Users provide the template, and the package generates the dashboard based on the structure defined in the Excel file.

Example: Building a Basic Dashboard

To create a dashboard from a spreadsheet template, first design a file with the necessary structure for the dashboard, including tabs, charts, and tables. The next task is to read in the Excel Template and your data to display needed for the dashboard. You can do data manipulation at this stage to the data, if needed. Finally, using the `build_shiny` function, the user can build their shiny dashboard.

```
library(ExcelToShiny)

# Load the Excel Template
excel_file <- "path_to_template.xlsx"

# Create the Shiny Dashboard
shiny_app <- build_shiny(title = "Testing App",
                        data_list = excel_file,
                        data_frame = data_file)
```

This process generates and launches a Shiny dashboard based on the specified spreadsheet template, requiring minimal programming expertise.

Future Goals

The ongoing development of ExcelToShiny aims to extend its functionality and usability. Improvements include optimisation changes on the R code to improve its efficiency, and adding additional features for further flexibility in the Shiny Dashboards.

Hosted on GitHub, ExcelToShiny encourages contributions from the community to ensure the package continues to evolve and meet the diverse needs of its users. This community-driven development fosters an open-source, collaborative approach to refining the tool.

Conclusion

ExcelToShiny lowers the barriers to entry for users without extensive programming experience in creating Shiny Dashboards. By automating the generation of dashboards from Excel templates, it ensures accessibility, reproducibility, and transparency in data visualisation. As the package evolves, it aims to serve as a tool for researchers, analysts, and decision-makers in a wide range of fields, providing an easy-to-use, powerful solution for data presentation.

References

- Biswas, D. (2021). *Tableau vs Shiny: Which one should you pick for data visualisation*. <https://analyticsindiamag.com/ai-origins-evolution/tableau-vs-shiny-which-one-should-you-pick-for-data-visu>
- Dutta, S. (2024). *Business dashboards decoded: Visualizing success in data-driven decision-making*. <https://www.sprinkledata.com/blogs/business-dashboards-decoded-visualizing-success-in-da>
- Koldus, M. (2024). *Beyond static reporting: Interactive dashboards*. <https://www.gooddata.com/blog/beyond-static-repoting-interactive-dashboards/>.
- Konrad, M. (2020). *Developing a complex R Shiny app – the good, the bad and the ugly*. <https://datascience.blog.wzb.eu/2020/02/27/developing-a-complex-r-shiny-app/>.
- Microsoft Corporation. (2024). *Power BI*. <https://www.microsoft.com/en-gb/power-platform/products/power-bi/>.
- Tableau Software. (2024). *Tableau*. <https://www.tableau.com/>