Setting up a public dotfiles repo for sharing and tranferring

RG Thomas

2025-07-28

Table of contents

Figure 1: UCSD Geisel Library - A hub for research and academic discovery

The Geisel Library at UC San Diego, where research and innovation converge. **TEMPLATE NOTE***: Replace this hero image with one relevant to your specific topic while maintaining visual impact and professional appearance. Consider using high-quality images from your field, data visualizations, or compelling stock photos that set the tone for your content.*

1 Introduction

In this post, we'll explore [topic/technique/problem]. This is particularly relevant for [target audience] because [motivation/problem statement].

By the end of this post, you'll be able to:

- [Learning objective 1]
- [Learning objective 2]
- [Learning objective 3]

2 Prerequisites and Setup

3 System to setup new users or a new machine.

```
Common files typically shared in public dotfiles repositories include:

  Shell Configuration:
  - .bashrc, .bash_profile, .zshrc - shell startup scripts
  - .aliases - command aliases
  - .functions - custom shell functions

  Development Tools:
  - .gitconfig - Git configuration and aliases
  - .vimrc or .nvim/ - Vim/Neovim configuration
  - .tmux.conf - tmux terminal multiplexer settings

  System Configuration:
  - .ssh/config - SSH client configuration (without private keys)
  - .inputrc - readline input configuration
  - .editorconfig - editor formatting rules

  Package Managers:
  - Brewfile (macOS) - Homebrew package list
  - .npmrc - npm configuration

  Setup Scripts:
  - install.sh or setup.sh - automated installation scripts
  - Makefile - installation and management commands

  Popular additions:
  - .prompt - custom shell prompts
  - .exports - environment variables
  - .path - PATH modifications
  - Application-specific configs (VS Code, iTerm2, etc.)

  Most dotfiles repos exclude sensitive files like .ssh/id_rsa, .aws/credentials, or any file

> how do developers usually move sensitive files from one machine to another_

  Developers typically handle sensitive files through these methods:
```

```
Encrypted Storage:
- 1Password/Bitwarden - store SSH keys, API tokens, certificates
- Keybase - encrypted file sharing and storage
- Git-crypt - encrypt specific files within git repos

Cloud Services:
- iCloud/Google Drive with encryption for SSH keys
- Dropbox with additional encryption layer
- AWS S3/Google Cloud with client-side encryption

Manual Transfer:
- SCP/RSYNC over SSH for direct machine-to-machine transfer
- USB drives with encryption (LUKS, FileVault, BitLocker)
- AirDrop (macOS) for local network transfers

Key Management:
- SSH agent forwarding - use keys from one machine on another
- Hardware tokens (YubiKey) for portable authentication
- Cloud key management (AWS KMS, HashiCorp Vault)

Best Practices:
- Generate new keys per machine when possible
- Use separate keys for different purposes
- Never commit sensitive files to version control
- Use .gitignore patterns like *.pem, *.key, .env.local

Most experienced developers avoid moving sensitive files entirely, preferring to regenerate

<!-- Include this section if applicable -->

Before we begin, ensure you have the following:

**Required Packages:**
::: {.cell}

```{.r .cell-code}
# Install required packages if not already installed
install.packages(c("package1", "package2", "package3"))
```

:::
```

Load Libraries:

```
# Replace with your actual packages
# library(dplyr)
# library(ggplot2)
# library(readr)
```

Sample Data:

```
# Replace with your actual data loading
# data <- read_csv("your_data.csv")
# data <- mtcars  # Example with built-in data
```

4 Main Section 1: [Descriptive Heading]

[Explanation of first main concept]

```
# Replace with your actual example code
# result <- your_function(data)
# print(result)
```

4.1 Subsection 1.1: [More Specific Topic]

[More detailed explanation or variation]

Figure 2: Optional supporting visualization with descriptive caption

5 Main Section 2: [Implementation/Analysis]

[Detailed implementation or analysis]

```
# Replace with your actual advanced example
# advanced_result <- complex_analysis(data)
# summary(advanced_result)
```

5.1 Subsection 2.1: [Handling Edge Cases]

[Discussion of potential issues and solutions]

```
# Replace with your actual error handling code
# tryCatch({
#   risky_operation(data)
```

```
# }, error = function(e) {
#   message("Error handled: ", e$message)
# })
```

6 Main Section 3: [Results/Advanced Applications]

[Analysis of results or advanced applications]

```
# Replace with your actual final analysis
# final_plot <- ggplot(data, aes(x, y)) +
#   geom_point() +
#   theme_minimal()
# print(final_plot)
```
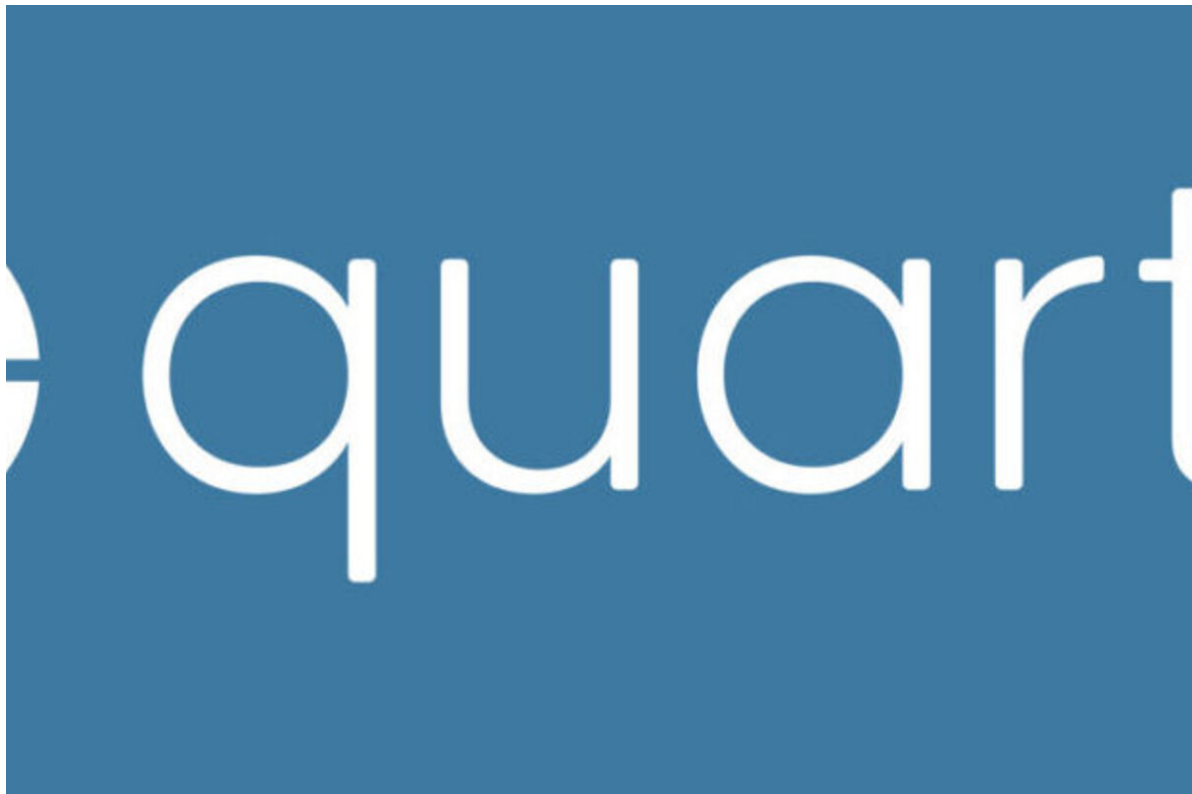

Figure 3: Data visualization workflow - from raw data to insights

TEMPLATE NOTE: This is a good location for a second large, compelling image. Consider using a relevant technical diagram, process flow, screenshot of results, or another high-impact

visual that supports your main content. This creates visual rhythm and helps maintain reader engagement throughout the post.

7 Main Section 4: [Performance/Comparison]

[Performance analysis or comparison with alternative approaches]

```
# Replace with your actual benchmarking code
# system.time(method1(data))
# system.time(method2(data))
```

8 Results and Key Findings

Our analysis revealed several key findings:

1. **[Key finding 1]**: [Brief explanation with numbers if applicable]
2. **[Key finding 2]**: [Brief explanation]
3. **[Key finding 3]**: [Brief explanation]

Figure 4: Research insights emerging from systematic analysis - like knowledge discovered in UC San Diego's academic environment

Just as the Geisel Library serves as a foundation for discovery, our analysis provides a solid foundation for understanding [your topic]. **TEMPLATE NOTE**: *This is the primary results image location - replace with your most important visualization, key findings chart, or compelling summary graphic that encapsulates your main conclusions.*

9 Limitations and Considerations

While this approach is effective, there are some important considerations:

9.1 Model Assumptions

- [**Assumption 1**]: [e.g., Linearity assumption - check with residual plots]
- [**Assumption 2**]: [e.g., Independence of observations]
- [**Assumption 3**]: [e.g., Homoscedasticity - constant variance]

9.2 Data Limitations

- **Sample size**: [Discussion of adequacy for conclusions]
- **Generalizability**: [Population this applies to vs. broader populations]
- **Missing data**: [How missing values were handled and potential bias]

9.3 Method Limitations

- [**Limitation 1**]: [Explanation and potential workarounds]
- [**Limitation 2**]: [When this approach may not be appropriate]
- **Performance considerations**: [Computational requirements, scalability]

10 Future Extensions

This work could be extended in several directions:

- [Extension idea 1]
- [Extension idea 2]
- [Extension idea 3]

11 Conclusion

In this post, we've demonstrated [brief summary of what was accomplished]. The key advantages of this approach are [main benefits].

Next Steps: - Try this technique with your own data - Experiment with different parameters - Explore the additional resources below

I encourage you to adapt this approach to your specific use case and share your experiences in the comments below.

12 References and Further Reading

12.1 Academic Literature

1. **Primary Research Papers:**

   - Wickham, H. (2014). "Tidy Data". *Journal of Statistical Software*, 59(10), 1-23. https://doi.org/10.18637/jss.v059.i10

- Breiman, L. (2001). "Random Forests". *Machine Learning*, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324
- [Your domain-specific paper]. Author, A. (Year). "Relevant Paper Title". *Journal Name*, Volume(Issue), pages. DOI

2. **Foundational Books:**

- Wickham, H., & Grolemund, G. (2017). *R for Data Science.* O'Reilly Media. https://r4ds.had.co.nz/
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning with Applications in R* (2nd ed.). Springer.
- [Your domain book]. Author, B. (Year). *Book Title.* Publisher.

3. **Statistical Methods:**

- Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time Series Analysis: Forecasting and Control* (5th ed.). Wiley.
- Gelman, A., & Hill, J. (2006). *Data Analysis Using Regression and Multi-level/Hierarchical Models.* Cambridge University Press.

12.2 Blog Posts and Tutorials

1. **Technical Blog Posts:**

- R-bloggers: "Advanced ggplot2 Techniques" - Comprehensive visualization strategies
- Simply Statistics: "The Role of Statistics in Data Science" - Foundational concepts
- Towards Data Science: "Machine Learning Best Practices" - Practical implementation guidance

2. **Package-Specific Tutorials:**

- Package creator's blog: "Introduction to [PackageName]" - Official guidance from package authors
- RStudio Blog: "New Features in [Package]" - Updates and best practices
- Stack Overflow: "Common [Package] Issues and Solutions" - Community troubleshooting

3. **Domain-Specific Applications:**

- Industry blog: "Real-world Application of [Method]" - Practical case studies
- Academic blog: "Methodological Considerations for [Technique]" - Research perspectives
- Practitioner blog: "Lessons Learned from [Project]" - Implementation insights

12.3 Technical Documentation

1. **Package Documentation:**

   - Package Reference Manual - Complete function documentation
   - Package Vignettes - Detailed usage examples
   - GitHub Repository - Source code and development issues

2. **Language and Framework Guides:**

   - R Language Definition - Official R documentation
   - Quarto Documentation - Publishing framework reference
   - RMarkdown Cookbook - Advanced document preparation

3. **Standards and Best Practices:**

   - Google's R Style Guide - Code formatting standards
   - rOpenSci Packages - Peer-reviewed R packages for research
   - CRAN Task Views - Domain-specific package collections

12.4 Community Resources

1. **Q&A and Discussion:**

   - Cross Validated - Statistical methodology discussions
   - Stack Overflow R Tag - Programming troubleshooting
   - RStudio Community - User support and discussions

2. **Social Learning:**

   - #rstats Twitter - Community updates and tips
   - R Weekly Newsletter - Curated R news and resources
   - R-Ladies Global - Inclusive R community and events

3. **Professional Networks:**

   - LinkedIn R Groups - Professional networking and job opportunities
   - Meetup R Groups - Local community events
   - UseR! Conference - Annual R user conference

12.5 Data Sources and Repositories

1. **Public Datasets:**

   - UCI Machine Learning Repository - Benchmark datasets
   - Kaggle Datasets - Community-contributed data

- [government data portal] - Domain-specific public data

2. **R Built-in Data:**

   - `datasets` package - Standard R datasets for examples
   - [Your specific dataset source] - Domain-relevant data repositories

12.6 Related Work and Extensions

1. **Methodological Extensions:**

   - Author, C. (Year). "Extension of [Your Method]". *Journal*, Volume(Issue), pages.
   - Author, D. (Year). "Comparative Analysis of [Related Methods]". *Conference Proceedings.*

2. **Applications in Other Domains:**

   - Author, E. (Year). "Application to [Different Field]". *Domain Journal*, Volume(Issue), pages.
   - Author, F. (Year). "Cross-disciplinary Perspectives on [Topic]". *Interdisciplinary Journal.*

Citation Note: When using ideas or code from these resources, please cite appropriately. For academic work, use standard citation formats. For blog posts and online resources, include the author, title, publication date, and URL.

13 Reproducibility Information

13.1 Data Availability

- **Dataset**: [Name and source of dataset used]
- **Access**: [How others can access the data - URL, package, etc.]
- **License**: [Data usage license and restrictions]

13.2 Code Repository

- **GitHub**: [Link to repository with complete analysis code]
- **Commit**: [Specific commit hash for reproducibility]
- **Environment**: [Docker image, renv lockfile, or environment specs]

13.3 Session Information

```
R version 4.5.0 (2025-04-11)
Platform: aarch64-apple-darwin20
Running under: macOS Sequoia 15.5

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/lib/libRlapack.dylib;

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

time zone: America/Los_Angeles
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

loaded via a namespace (and not attached):
 [1] compiler_4.5.0   fastmap_1.2.0    cli_3.6.5         tools_4.5.0
 [5] htmltools_0.5.8.1 yaml_2.3.10      rmarkdown_2.29   knitr_1.50
 [9] jsonlite_2.0.0   xfun_0.52        digest_0.6.37    rlang_1.1.6
[13] evaluate_1.0.3
```

14 Appendix: [Optional Detailed Information]

14.1 Appendix A: Complete Code

```
# Complete code for easy reproduction - replace with your actual code
# library(your_packages)
# data <- load_your_data()
# results <- your_analysis(data)
# plot(results)
```

14.2 Appendix B: Mathematical Details

For statistical posts, include relevant formulas using LaTeX notation:

Linear Regression Model:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip} + \epsilon_i$$

Model Evaluation Metrics: - **RMSE**: $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2}$ - **R-squared**: $R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$

[Additional mathematical explanations or derivations as needed]

14.3 Appendix C: Additional Data

[Additional tables, charts, or data summaries]

14.4 Share This Post

Found this helpful? Share it with your network:

- Twitter
- LinkedIn
- Reddit

14.5 Connect and Discuss

Have questions or suggestions? I'd love to hear from you:

- **Twitter**: @rgt47 - Quick questions and discussions
- **LinkedIn**: Ronald Glenn Thomas - Professional networking
- **GitHub**: rgt47 - Code, issues, and contributions
- **Email**: Contact through website - Detailed inquiries

Comments are enabled below via Utterances - join the discussion!

14.6 About the Author

Ronald (Ryy) Glenn Thomas is a biostatistician and data scientist at UC San Diego, specializing in statistical computing, machine learning applications in healthcare, and reproducible research methods. He develops R packages and conducts research at the intersection of statistics, data science, and clinical research.

Connect: *Website* | *ORCID* | *Google Scholar*