

Some more R, Git, and R Markdown and Quarto tips

MRC IEU Code Clinic
07/05/2025
Tom Palmer

Overview

- Some more R Markdown and Quarto tips
 - Seven tips for making Quarto revealjs presentations
 - Creating R Markdown and Quarto tutorial documents
 - Creating multilanguage Quarto documents (i.e., R and Stata executed code in same doc)
- Some more R tips
 - Creating a reproducible environment in R without using renv - CRAN snapshot repositories for the win
- Some more Git, GitHub Desktop (and Lazygit), and GitHub tips
 - Making suggestions on a GitHub PR
 - Amending a previous commit message in GitHub Desktop
 - Amending a previous commit in GitHub Desktop
 - Amending previous commits and commits messages in Lazygit - interactive rebase made "easy"

Some more R Markdown and Quarto tips

Seven tips for making Quarto presentations

- Quarto revealjs docs <https://quarto.org/docs/presentations/revealjs/>
- <https://remlapmot.github.io/post/2025/quarto-revealjs-tips/>

Creating R Markdown and Quarto tutorial documents

- R Markdown:
- Quarto: <https://remlapmot.github.io/post/2025/quarto-conditional-content/>

Creating multilanguage Quarto documents

- Technically Quarto documents may only have 1 engine
- Workaround: use `{{< embed >}}` shortcode to include chunks and output from documents using other engines
- <https://remlapmot.github.io/post/2025/multi-engine-quarto/>
- Setting up the nbstata Jupyter kernel in a uv venv
 - uv docs (including installation instructions) <https://docs.astral.sh/uv/>

Some more R tips

Reproducible R environments using CRAN snapshot repos without using renv/pak/pacman

- (Without using renv or similar)
- Simply note the date at the top of your script in a comment
- Then run `update.packages()`
- (Assume using current version of R)
- Then if you need to recreate this environment use a snapshot from <https://packagemanager.posit.co/client/#/repos/cran/setup>
- Nb. RSPM = PPPM = P3M

Set up your environment to install R packages from cran

Select from the following options, then follow the customized instructions below to complete your setup.

Operating System: what system are you using the packages on?

- ☐ macOS
☒ Windows
☐ Linux

Snapshots: do you want to freeze package versions to enhance reproducibility?

- ☐ No, install the most recent packages available
☒ Yes, always install packages from the date I choose

025

February 2025						March 2025												
TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT
	1	2	3	4						1								1
7	8	9	10	11	2	3	4	5	6	7	8	2	3	4	5	6	7	8
14	15	16	17	18	9	10	11	12	13	14	15	9	10	11	12	13	14	15
21	22	23	24	25	16	17	18	19	20	21	22	16	17	18	19	20	21	22
28	29	30	31		23	24	25	26	27	28		23	24	25	26	27	28	29
												30	31					

The URL below encodes the following snapshot references:

- cran [Last update: Mar 20, 2025 12:00 AM UTC]

Environment: where are you using R?

- ☒ RStudio IDE
☐ Posit Workbench
☐ Posit Connect
☐ Some other R outside of RStudio IDE

Repository URL:

<https://packagemanager.posit.co/cran/2025-03-20>

Copy

Windows

Package Manager
compatible with v
optionally selecte

Frozen to March

Package Manager
as of March 20, 20

Freezing to a spec
when re-installing
always match, or
from a previous p

RStudio IDE

Get more info on

Repository URL

Use this URL to in
Manager directly.

- `install.packages(c("package1", "package2"), repos = "https://packagemanager.posit.co/cran/2025-03-20")`
- Snapshots created at midnight - so you might need tomorrow's date
- The Public Posit Package Manager now only CRAN snapshotting service (used to be MRAN as well)
- Many companies advocated this approach: RevolutionR, Cynkra (e.g., <https://github.com/cynkra/cynkrathis>)
- Can be used within renv

cynkra RSPM snapshots

cynkra makes use of certain RStudio Package Manager (RSPM) snapshots across projects. Usually each R Version is tied to a snapshot near it's release date. If a snapshot is considered unstable due to certain R package version clashes, additional snapshots for specific R versions can be listed.

- For a GitHub only package you can record the most recent commit and then include its installation as:

```
remotes::install_github('MRCIEU/TwoSampleMR@6268c2c')
```

- Or you could save a copy of the repo on the day you installed it
- Or you can download the package from an r-universe - see apis page
 - <https://mrcieu.r-universe.dev/apis>

Download snapshot

[Download](#)

Download a full copy of the cran-like repository with docs, to mirror or backup.

Optional filters (defaults include everything):

<input type="checkbox"/> Source packages	<input type="checkbox"/> Binaries for R-4.5
<input type="checkbox"/> Windows binaries	<input type="checkbox"/> Binaries for R-4.4
<input type="checkbox"/> MacOS binaries	<input type="checkbox"/> Binaries for R-4.3
<input type="checkbox"/> Linux binaries	
<input type="checkbox"/> WebAssembly binaries	
<input type="checkbox"/> Package docs	

- alspac
- BESIDEMR
- bpbounds
- BWMR
- CAMeRa
- cause
- causl
- CheckSumStats
- cit
- CIVMR

Some more Git, GitHub Desktop (and
Lazygit), and GitHub tips

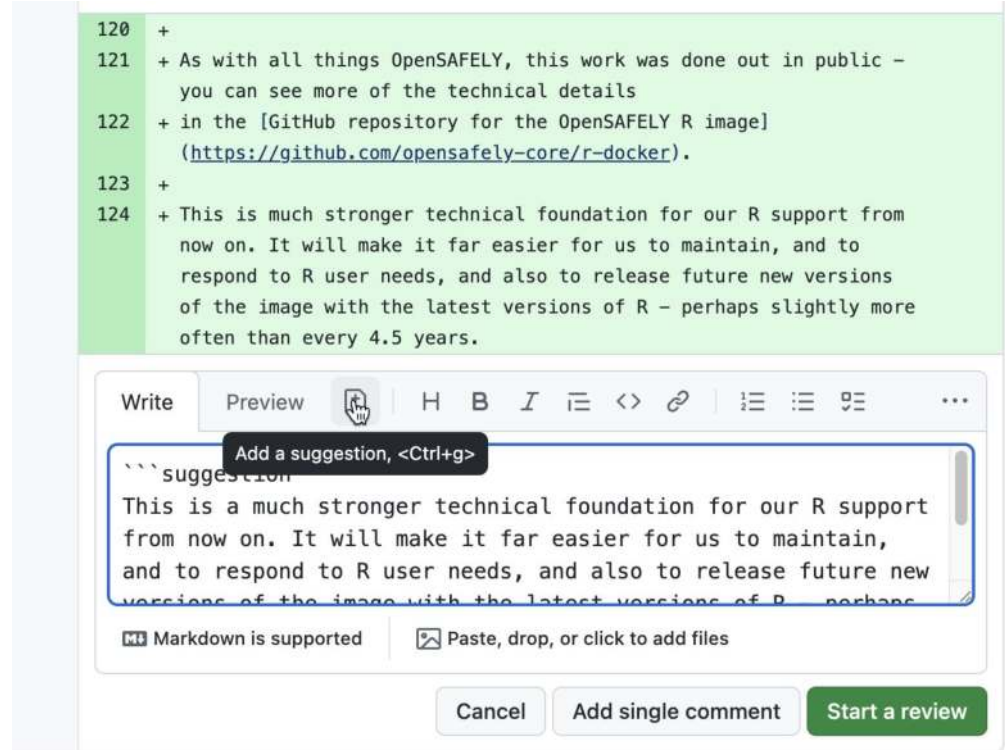
Making suggestions on a GitHub PR

Click + on LHS for a line (or Shift click for several lines)

Then button third right in popup is Add a suggestion; and has Markdown syntax

```suggestion

```



The screenshot shows a GitHub Pull Request interface. At the top, a green diff view shows changes to a file. Line 120 has a '+' sign. Line 121 has a '+' sign and the text 'As with all things OpenSAFELY, this work was done out in public - you can see more of the technical details'. Line 122 has a '+' sign and the text 'in the [GitHub repository for the OpenSAFELY R image] (<https://github.com/opensafely-core/r-docker>)'. Line 123 has a '+' sign. Line 124 has a '+' sign and the text 'This is much stronger technical foundation for our R support from now on. It will make it far easier for us to maintain, and to respond to R user needs, and also to release future new versions of the image with the latest versions of R - perhaps slightly more often than every 4.5 years.'

Below the diff view, a 'Write' button is visible. To its right is a 'Preview' button, followed by a button with a hand icon. A tooltip 'Add a suggestion, <Ctrl+g>' is shown over this button. Below the buttons is a text area containing the text 'This is a much stronger technical foundation for our R support from now on. It will make it far easier for us to maintain, and to respond to R user needs, and also to release future new versions of the image with the latest versions of R - perhaps'. Below the text area are two checkboxes: 'Markdown is supported' (checked) and 'Paste, drop, or click to add files'. At the bottom are three buttons: 'Cancel', 'Add single comment', and 'Start a review'.

Amending previous commit message in GitHub Desktop

- <https://remlapmot.github.io/post/2025/amend-commit-messages/>

Interactive rebase made "easy" in Lazygit

- My Git levels of attainment
 - Beginner: branches, push and pull between local and GitHub, make PR
 - Medium: squash, reorder, resolve merge conflicts, stash, worktrees
 - Advanced: interactive rebase
- Interactive rebase at the command line is very very hard (see next slide)
<https://opensource.com/article/20/3/lazygit>
- Lazygit installation instructions:
<https://github.com/jesseduffield/lazygit?tab=readme-ov-file#installation>
- Launch by running lazygit in Terminal

Interactive rebase at the command line is hard (don't do it)

The user experience of an interactive rebase on the Git command line is a horror story that belongs in a Stephen King novel. To do something as simple as amending an old commit requires following these steps (do not let your kids read this without parental guidance):

1. Stash the changes you want to apply with **git stash**.
2. Copy the SHA of the commit you want to amend.
3. Begin the rebase with **git rebase --interactive <commit-SHA>^**.
4. (This is where the screams start...) A TODO file opens in Vim, where you'll need to find your commit and replace **pick** on the line with **edit**.
5. Save the file.
6. Unstash your changes with **git stash pop**.
7. Amend the commit with **git commit --amend**.
8. Continue the rebase with **git rebase --continue**.

Just thinking back to the days of CLI terror gives me heart palpitations.

Lazygit in action

```
[1]-Status
(rebasig) TwoSampleMR -> 53ea41

[2]-Files - Worktrees - Submo
NEWS.md

[3]-Commits (devel)
e99444dd TP ○ Delete use of mag
53ea4107 TP ○ Update NEWS.md
d171ac9f TP ○ devtools::documen
b96dac52 TP ○ Add parentheses
e7b251cc TP ○ Remove curly brac
f559a992 TP ○ Add parentheses

[4]-Diff files (53ea410 Updat
M NEWS.md

[5]-Stash
3w On multivariable: !!GitHub_D

View rebase options: m | Stage: <space> | Commit: c | Edit: e | Stash: s | -
```

Staged changes

```
diff --git a/NEWS.md b/NEWS.md
index 2512823f..3bb32e75 100644
--- a/NEWS.md
+++ b/NEWS.md
@@ -2,7 +2,7 @@

 (Release date 2025-##-##)

-* The use of the **magrittr** pipe, '%>%', has been replaced
with the use of the native pipe, '>'. As such the minimum
required version of R has been increased to R 4.1.0, however,
this should not cause any users to need to upgrade their version
of R because **TwoSampleMR** has required a more recent version
of R than R 4.1.0 for some time. We think that the oldest
version of R **TwoSampleMR** will install on is R 4.3.2.
+* The use of the **magrittr** pipe, '%>%', has been replaced
with the use of the native pipe, '>'.

# TwoSampleMR v0.6.15
```

Command log

```
git add -- NEWS.md
```

```
[1]-Status
(rebasig) TwoSampleMR -> 53ea41

[2]-Files - Worktrees - Submo
M NEWS.md

[3]-Commits (devel)

[4]-Diff files (53ea410 Updat
M NEWS.md

[5]-Stash
3w On multivariable: !!GitHub_D

View rebase options: m | Cancel: <esc>
```

Commit summary

```
diff --git a/NEWS.md b/NEWS.md
index 2512823f..3bb32e75 100644
--- a/NEWS.md
+++ b/NEWS.md
@@ -2,7 +2,7 @@

 (Release date 2025-##-##)

-* The use of the **magrittr** pipe, '%>%', has been replaced
with the use of the native pipe, '>'. As such the minimum
required version of R has been increased to R 4.1.0, however,
this should not cause any users to need to upgrade their version
of R because **TwoSampleMR** has required a more recent version
of R than R 4.1.0 for some time. We think that the oldest
version of R **TwoSampleMR** will install on is R 4.3.2.
+* The use of the **magrittr** pipe, '%>%', has been replaced
with the use of the native pipe, '>'.

# TwoSampleMR v0.6.15
```

Commit description

```
Press <tab> to toggle focus, <c-o> to open menu
```

Press <a-enter> to commit

Rewording and editing commits in Lazygit

- To edit previous commit messages in Lazygit
 - Navigate to Reflog
 - r - Reword - to edit previous commit message
 - Move to message box (<Tab>), alt + <Enter> to commit
- To edit previous commits in Lazygit
 - Move to Reflog [4]
 - Move to commit want to edit
 - e - Edit - to edit previous commit
 - Press <Enter> on commit want to edit
 - e - Edit
 - Pops you into vim - make edits - then save and quit :wq
 - Move back to top left Files pane
 - Space - stage changed file/s
 - Now continue rebase - m - rebase options - <Enter> or c - continue
 - Move to Commit pane - Escape - and should see edited commit
 - Now probably want to edit commit message of edited commit - Move to Reflog - r - Reword commit message of squashed commit

Summary

- Slightly random collection of R, Quarto, and Git tips
- I hope something was interesting/useful