Writing your first documentation PR

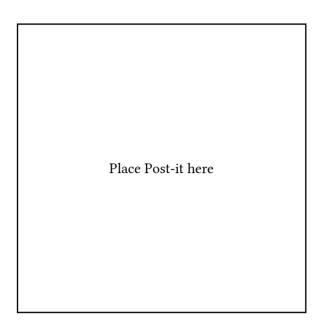
The goal of this worksheet is to guide you through the steps needed to successfully create your first PR (pull request). Empty spaces have been placed on this sheet for your notes.

☐ If Vaccinated: FALSE, consider running usethis::qit vaccinate().

1. Find and claim an issue

Browse the Post-its, looking for something of interest. Since you've never done a PR before, we recommend starting with a documentation issue since there are slightly fewer moving parts. We also encourage you to team up with a (new) friend and tackle something together!

Once you've picked a Post-it, open the issue and read the details in full. At this point, you might discover the issue is outside your wheelhouse; if so, no problem, just return the Post-it to the wall and try again.



| Get the source Fork and clone the | ce locally ne repository using this syn | ntax: | |
|----------------------------------------------------------------|----------------------------------------------------------|-----------------------|-------------|
| | te_from_github(" create_from_github("tidy | / yverse/ggplot2") |) |
| □ pak::local_inst packages. | all_dev_deps() to make s | sure you've got th | e necessary |
| | locument() to make sure yo s before you make any chan | • | g necessary |
| C | ols::document() fails or pr your setup and you should | Č | omething is |

3. Make the change

| ☐ Create a new branch for your changes to live with |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>- usethis::pr_init("") - (e.g usethis::pr_init("add-snapshot-tests"))</pre> |
| ☐ Make your change. |
| \square use devtools::document() to update the docs. |
| □ Commit the change. □ Check changed files in the git pane. □ Press Commit button. □ Write a short message (<80 characters) detailing the change. use the magic fixes keyword, e.g. "Fixes #123", to link & close your issue. |
| ☐ Press Commit button. |

5. Wait for review

A person from the tidyverse team will try and review your issue as quickly as possible, reading through your PR and offering suggestions for how to improve it. If you want to work on something else while you wait, use $pr_pause()$ to pause your pull request while you work on other stuff. It's possible the reviewer might make changes directly to your review - if that happens you can use $pr_pull()$ to get their code back onto your computer.

If you've committed a bigger PR, it might take a bit longer to review, and we might not be able to complete it on the day. Don't worry if this happens to you! We really appreciate your contribution and just need a bit more time to take it in.

4. Submit your PR

Push and make a PR with usethis::pr_push().

☐ Make the title descriptive:

- Not good: "Working on issue 45"
- Good: "Use stricter regexes for package and file name checks"
- ☐ Example with descriptive title and description containing magic keywords: https://github.com/r-lib/usethis/pull/742
- ☐ Write your PR number on the Post-it and move it to the review wall.
- ☐ Ring the gong to celebrate your successful submission!

Wrap up

At the end of the day, make sure that you've reinstalled the CRAN versions of any packages you might have installed development versions of. You can find a list of dev versions with this code:

```
subset(
  pak::lib_status(),
  as.numeric(package_version(version)[, 4]) >= 9000,
  c(package, version)
)
```