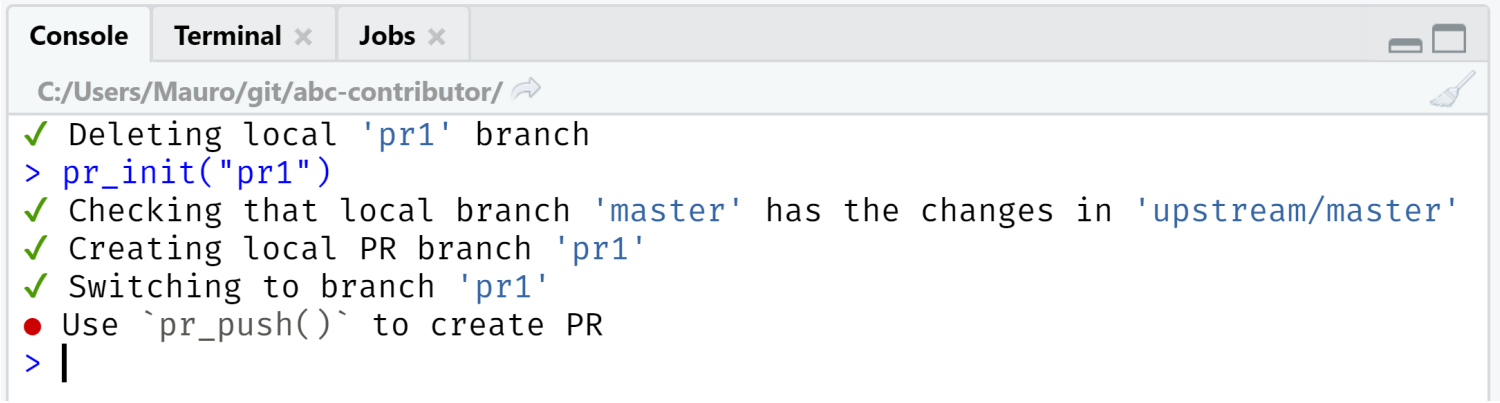
Interdependent pull requests

Situation

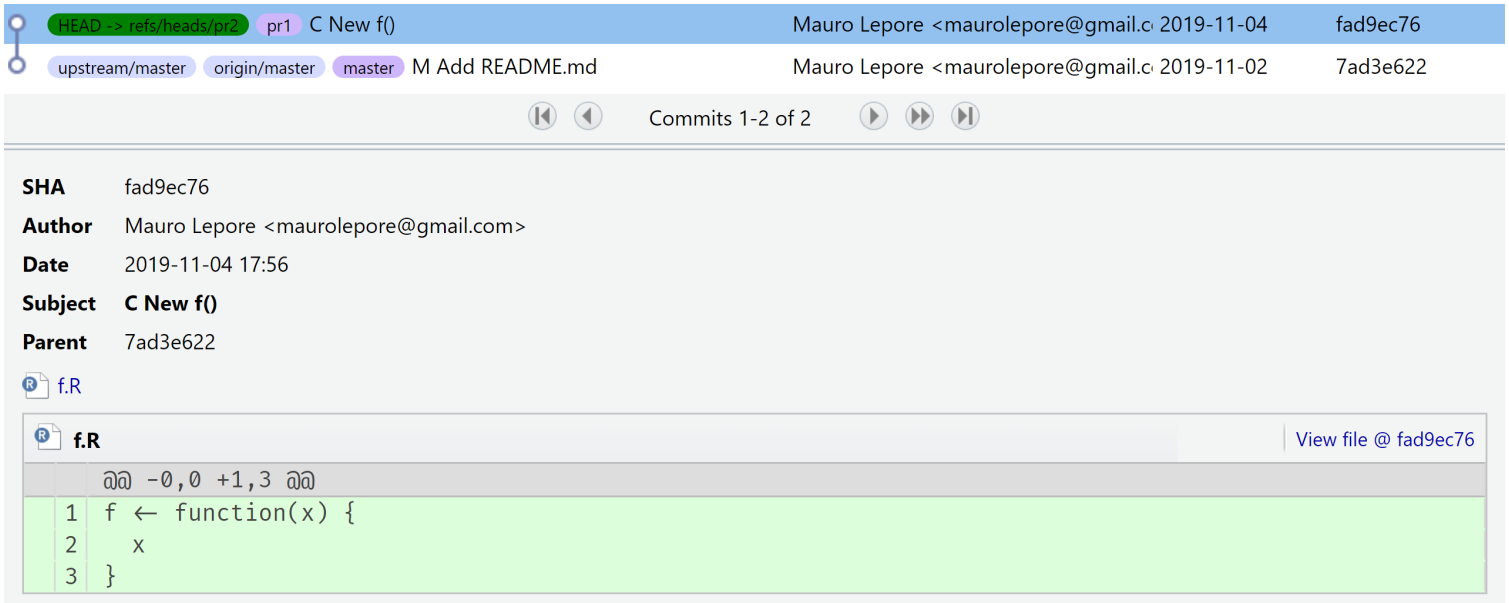
After the contributor submitted a first pull request (`pr1`), and before the maintainer merges it into the source repository, the contributor starts a second pull request (`pr2`) that depends on `pr1`.

The maintainer edits `pr1` and accepts it (squash-merges it into the upstream/master). But `pr2` now is a mess and the contributor must fix it before it can be merged.

pr_init("pr1")

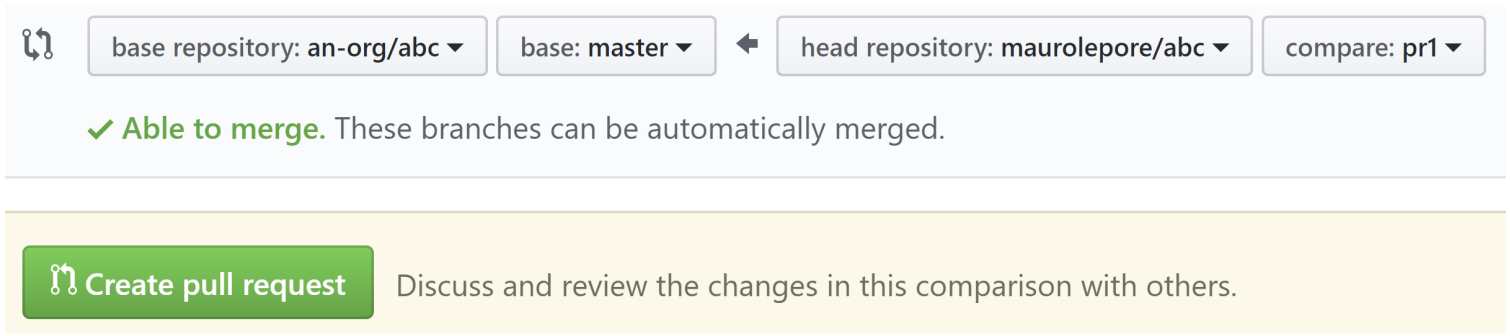

Console
Terminal
Jobs
C:/Users/Mauro/git/abc-contributor/
✓ Deleting local 'pr1' branch
> pr_init("pr1")
✓ Checking that local branch 'master' has the changes in 'upstream/master'
✓ Creating local PR branch 'pr1'
✓ Switching to branch 'pr1'
● Use `pr_push()` to create PR
>

New f()


HEAD -> refs/heads/pr2 pr1 C New f() Mauro Lepore <maurolepore@gmail.c 2019-11-04 fad9ec76
upstream/master origin/master master M Add README.md Mauro Lepore <maurolepore@gmail.c 2019-11-02 7ad3e622
Commits 1-2 of 2
SHA fad9ec76
Author Mauro Lepore <maurolepore@gmail.com>
Date 2019-11-04 17:56
Subject C New f()
Parent 7ad3e622
f.R
f.R
View file @ fad9ec76
@@ -0,0 +1,3 @@
1 f ← function(x) {
2 x
3 }

pr_push()

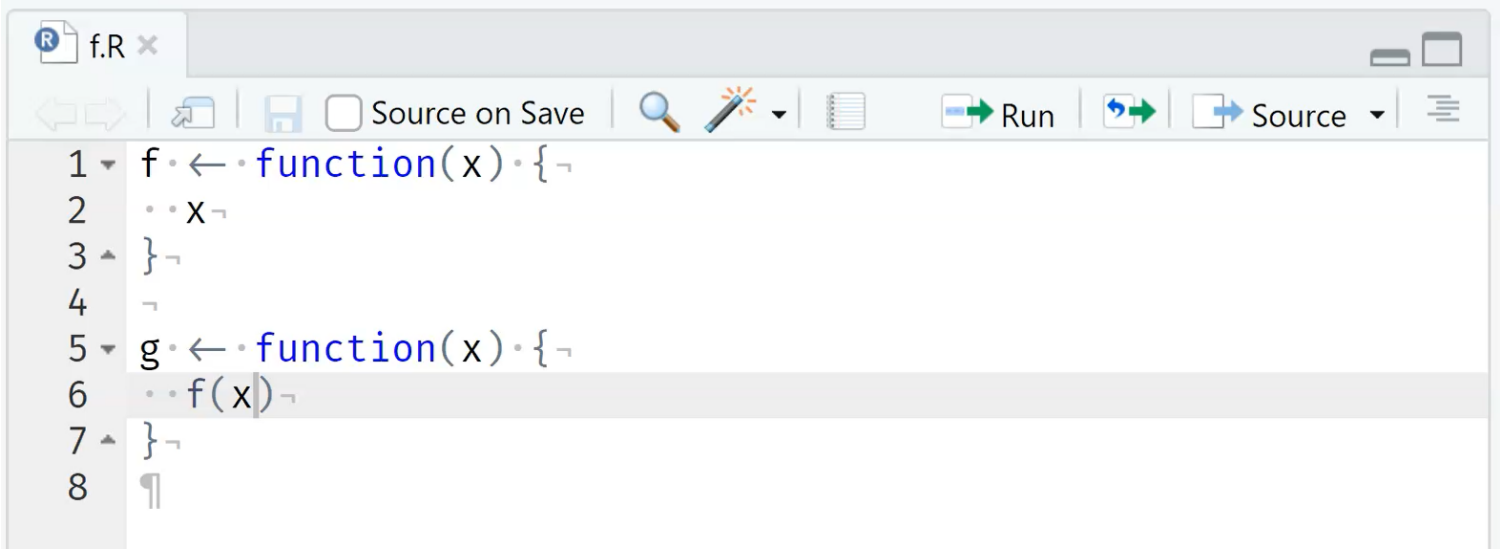
```
> pr_push()
✓ Pushing local 'pr1' branch to 'origin:pr1'
✓ Setting upstream tracking branch for 'pr1' to 'origin/pr1'
✓ Create PR at link given below
✓ Opening URL 'https://github.com/maurolepore/abc/compare/pr1'
>
```



pr_init("pr2")

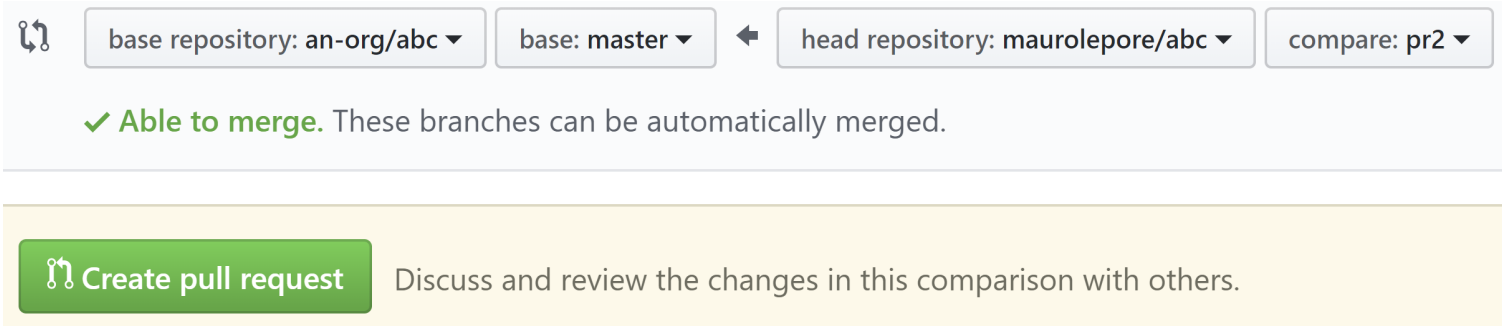
```
> pr_init("pr2")
✓ Setting active project to 'C:/Users/Mauro/git/abc-contributor'
✓ Checking that local branch 'master' has the changes in 'upstream/master'
Create local PR branch with non-master parent?

1: No
2: Absolutely
3: Absolutely not

Selection: 2
✓ Creating local PR branch 'pr2'
✓ Switching to branch 'pr2'
● Use `pr_push()` to create PR
>
```

New g()


f.R
Source on Save
Run
Source
1 f ← function(x) {
2 x
3 }
4
5 g ← function(x) {
6 f(x)
7 }
8

pr_push()

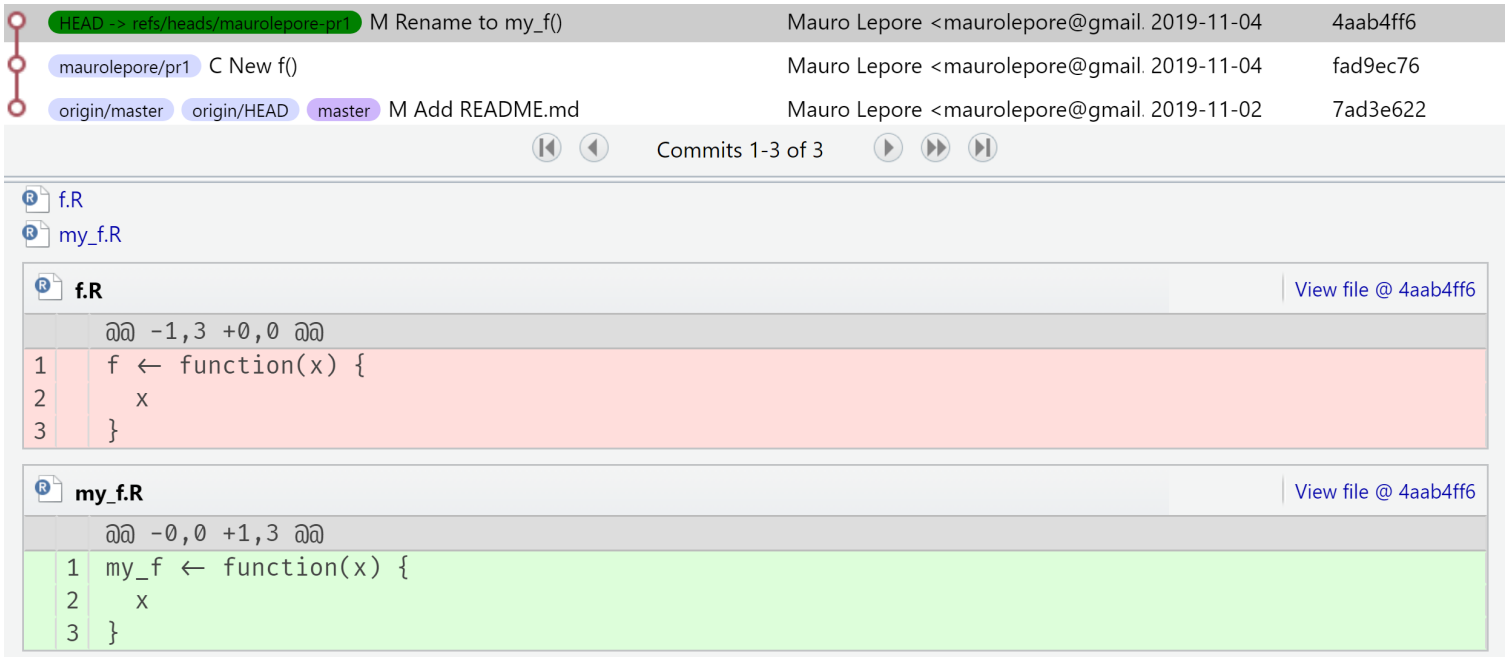
```
> pr_push()
✓ Pushing local 'pr2' branch to 'origin:pr2'
✓ Setting upstream tracking branch for 'pr2' to 'origin/pr2'
✓ Create PR at link given below
✓ Opening URL 'https://github.com/maurolepore/abc/compare/pr2'
>
```



`pr_fetch()` (Maintainer)

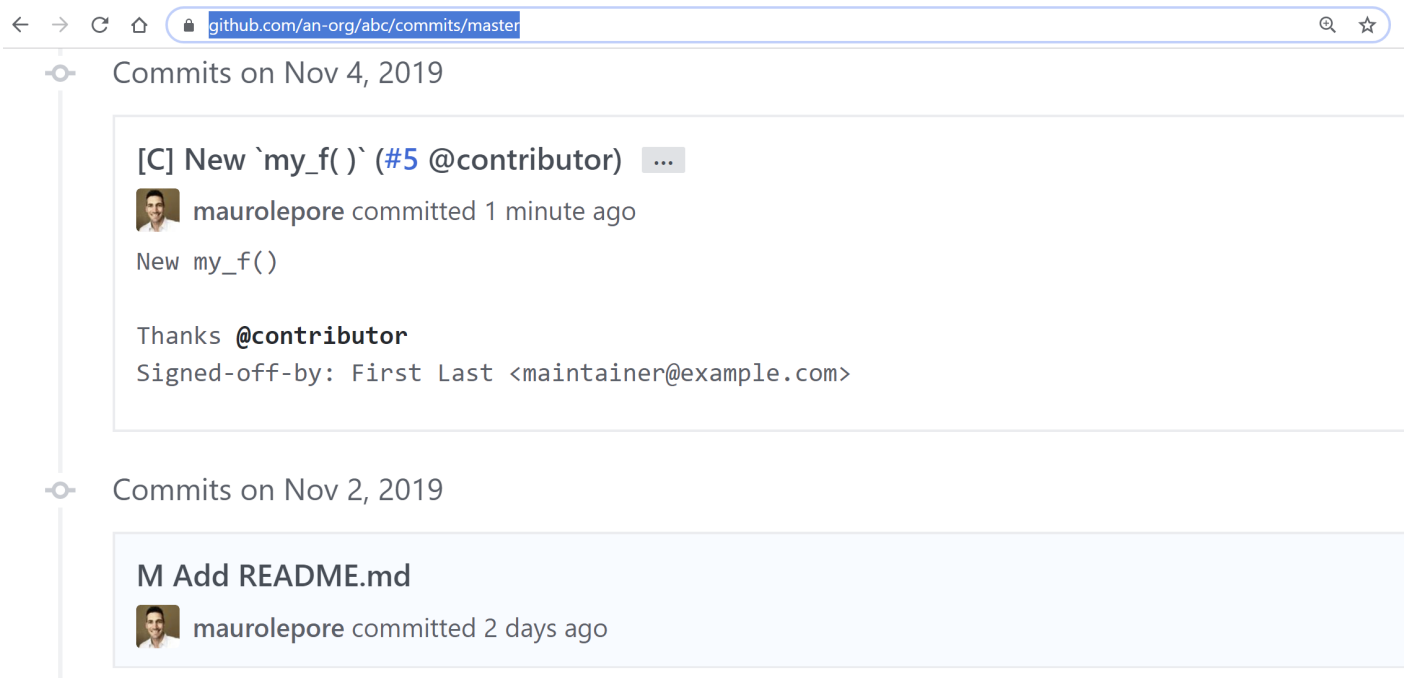
```
> library(usethis)
> pr_fetch(5)
✓ Checking out PR 'an-org/abc/#5' (@maurolepore): 'C New f()'
✓ Adding remote 'maurolepore' as 'git@github.com:maurolepore/abc.git'
✓ Creating local branch 'maurolepore-pr1'
✓ Setting upstream tracking branch for 'maurolepore-pr1' to 'maurolepore/pr1'
✓ Switching to branch 'maurolepore-pr1'
>
```

Edit **pr1** (Maintainer)

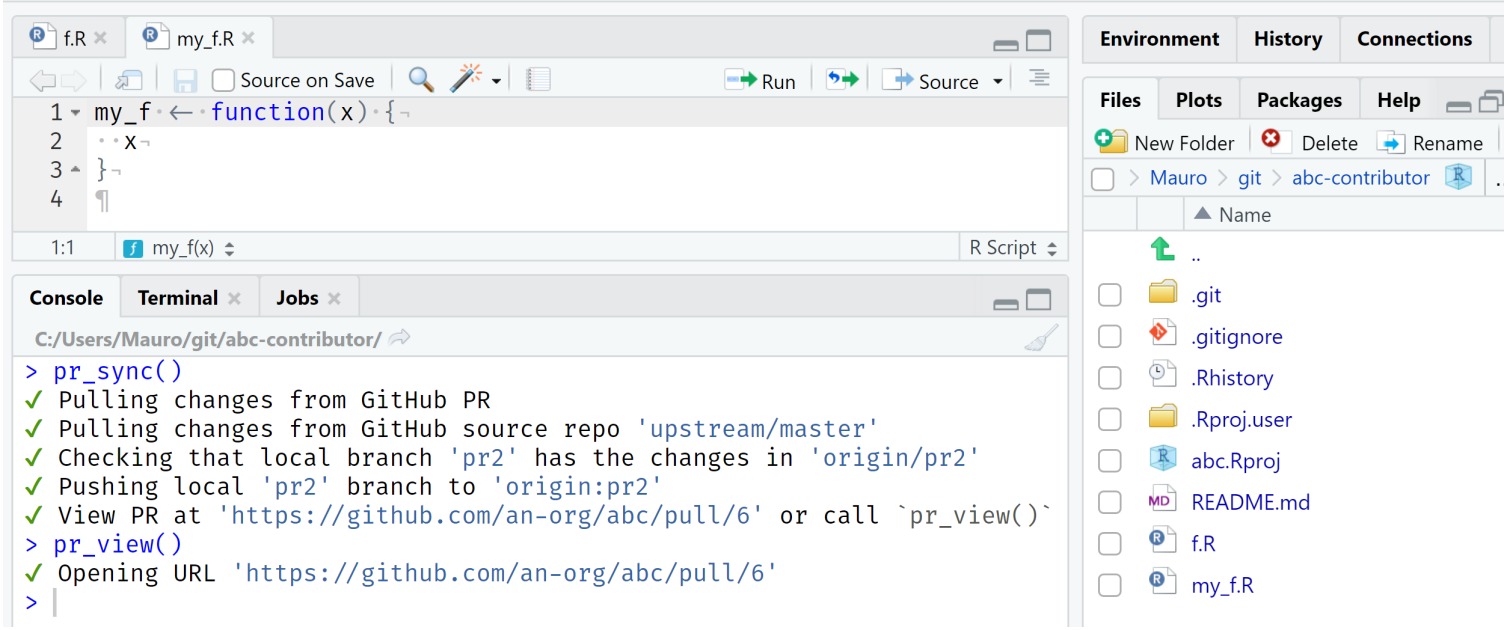


pr_push() (Maintainer)

```
> pr_push()
✓ Checking that local branch 'maurolepore-pr1' has the changes in 'maurolepore/pr1'
✓ Pushing local 'maurolepore-pr1' branch to 'maurolepore:pr1'
✓ View PR at 'https://github.com/an-org/abc/pull/5' or call `pr_view()`
>
```



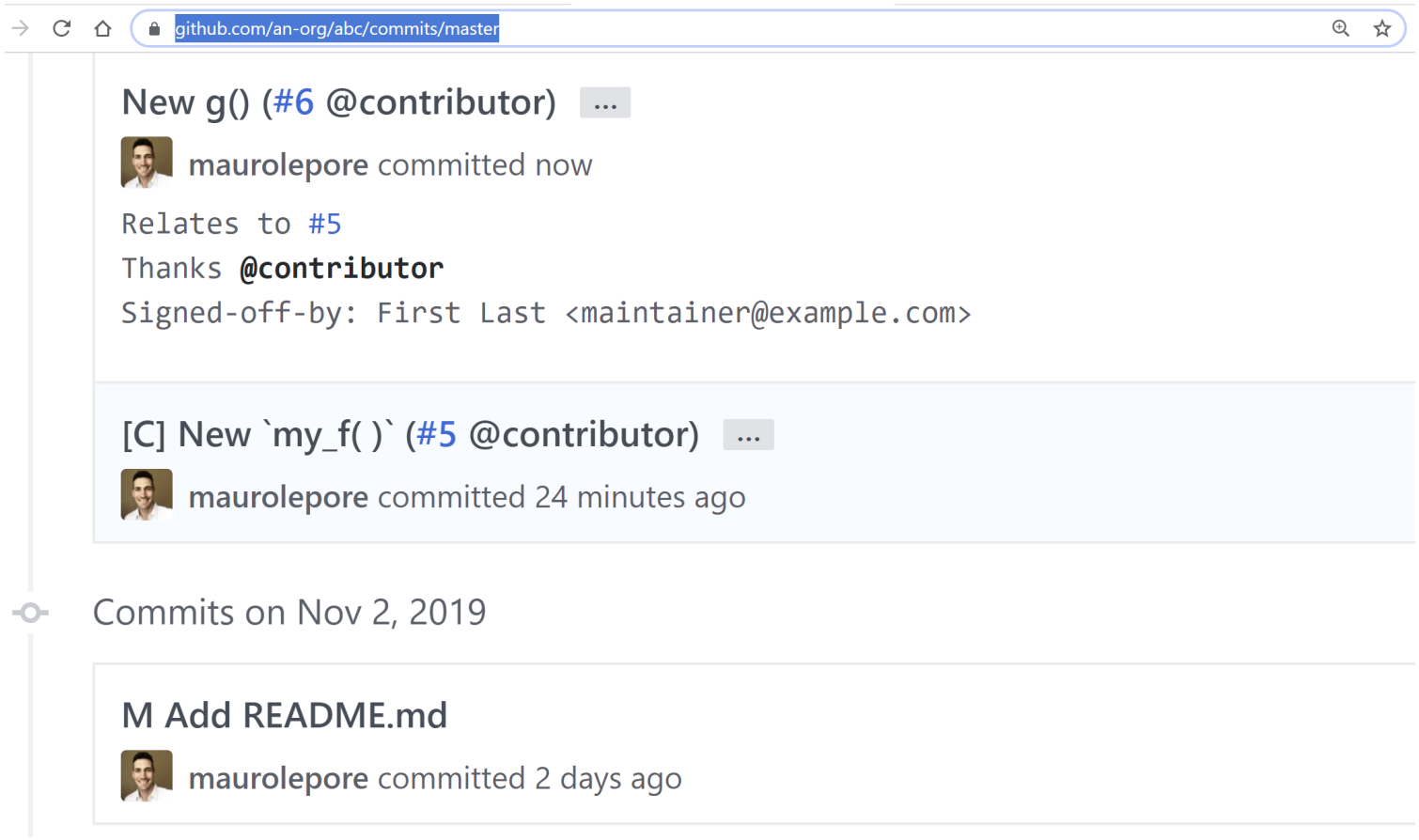
pr_sync()

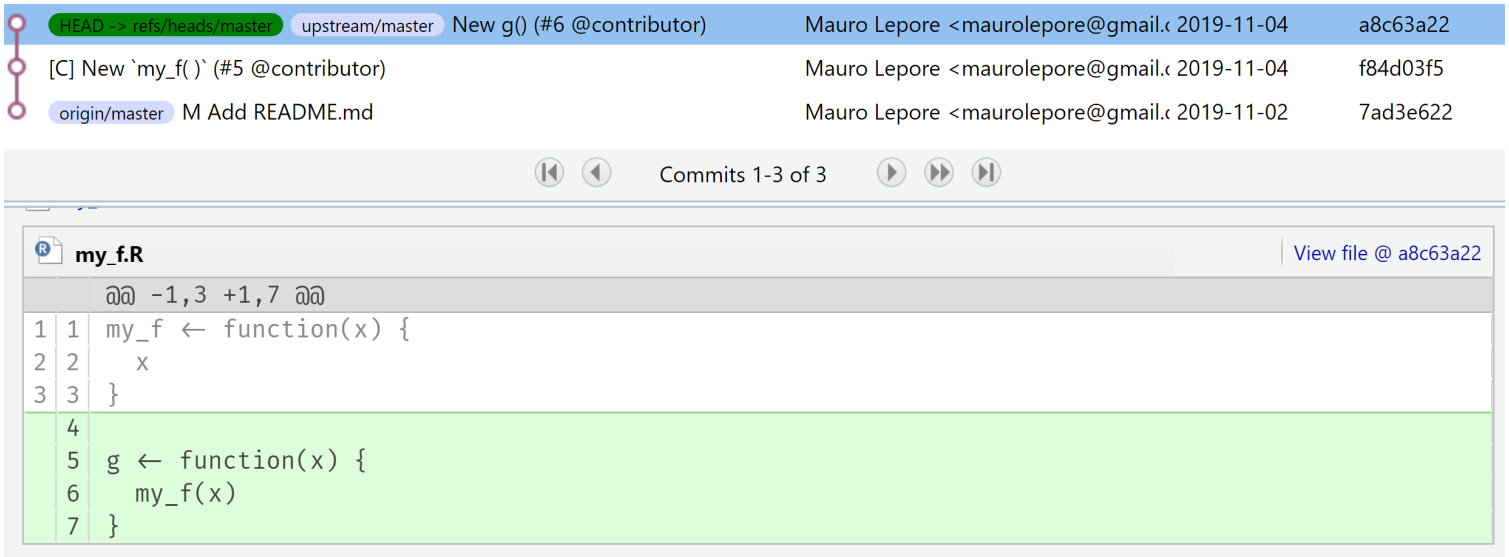

f.R
my_f.R
Source on Save
Run
Source
1 my_f ← function(x) {
2 x
3 }
4
1:1
my_f(x)
R Script
Console
Terminal
Jobs
C:/Users/Mauro/git/abc-contributor/
> pr_sync()
✓ Pulling changes from GitHub PR
✓ Pulling changes from GitHub source repo 'upstream/master'
✓ Checking that local branch 'pr2' has the changes in 'origin/pr2'
✓ Pushing local 'pr2' branch to 'origin:pr2'
✓ View PR at 'https://github.com/an-org/abc/pull/6' or call `pr_view()`
> pr_view()
✓ Opening URL 'https://github.com/an-org/abc/pull/6'
>
Environment
History
Connections
Files
Plots
Packages
Help
New Folder
Delete
Rename
Mauro
git
abc-contributor
Name
..
.git
.gitignore
.Rhistory
.Rproj.user
abc.Rproj
README.md
f.R
my_f.R

Fix pr2 then pr_push()

```
> pr_push()
✓ Checking that local branch 'pr2' has the changes in 'origin/pr2'
✓ Pushing local 'pr2' branch to 'origin:pr2'
✓ View PR at 'https://github.com/an-org/abc/pull/6' or call `pr_view()`
>
```

Squash-merge **pr2** (Maintainer)



pr_finish()

```
Console  Terminal ×  Jobs ×
C:/Users/Mauro/git/abc-contributor/
> pr_finish()
✓ Switching back to 'master' branch
✓ Pulling changes from GitHub source repo 'upstream/master'
✓ Deleting local 'pr2' branch
>
```