

{saperlipopette}, un paquet R
pour progresser en Git
en toute sérénité

Maëlle Salmon (rOpenSci, cynkra)

<https://rr2024-maelle.netlify.app/>

Git, outil de contrôle de version

De `script-vraiment-final-2024-05-27.R` à un vrai historique.

Pas du R mais très utile, voire indispensable, pour écrire du code.

Sans Git

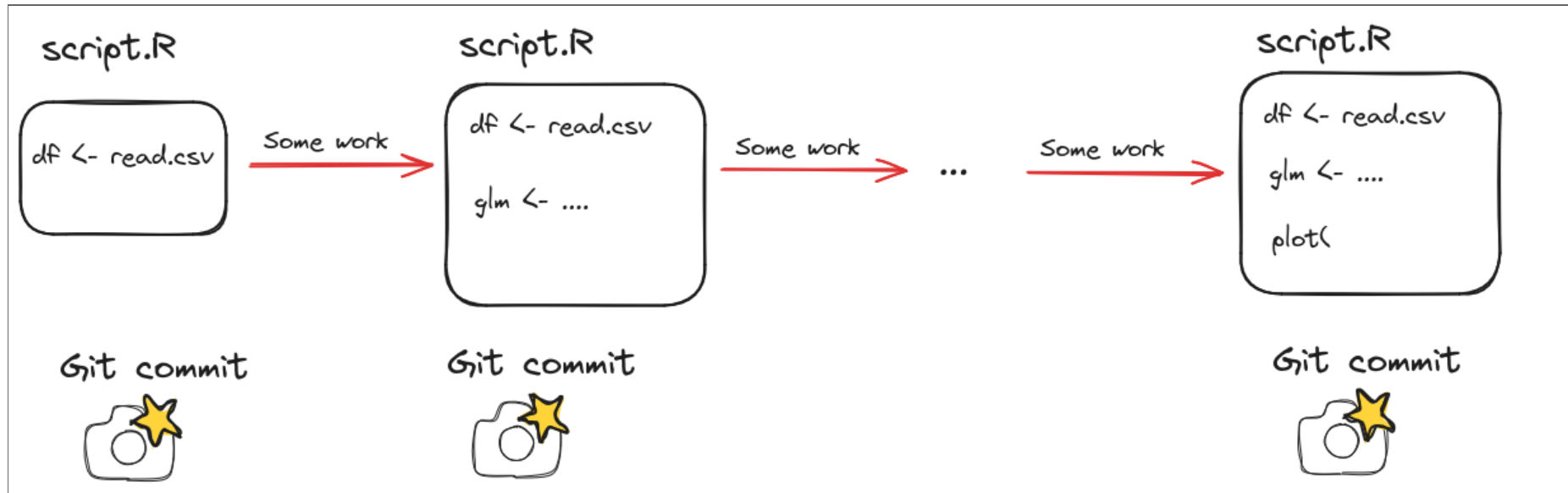
Jour 1 : script.R

Jour 2 : crée le modèle statistique, script2.R

...

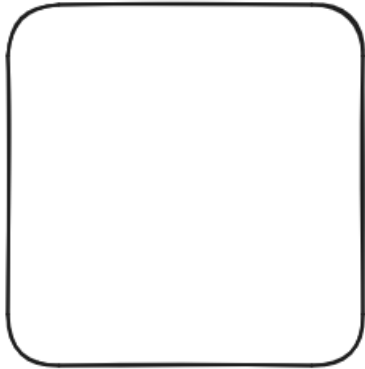
Jour 42 : rajoute les graphiques, script-final-2024-06-03.R (40ème copie du fichier 😁)

Avec Git : ce qu'on fait



Avec Git : ce qu'on a

script.R



.git



Time

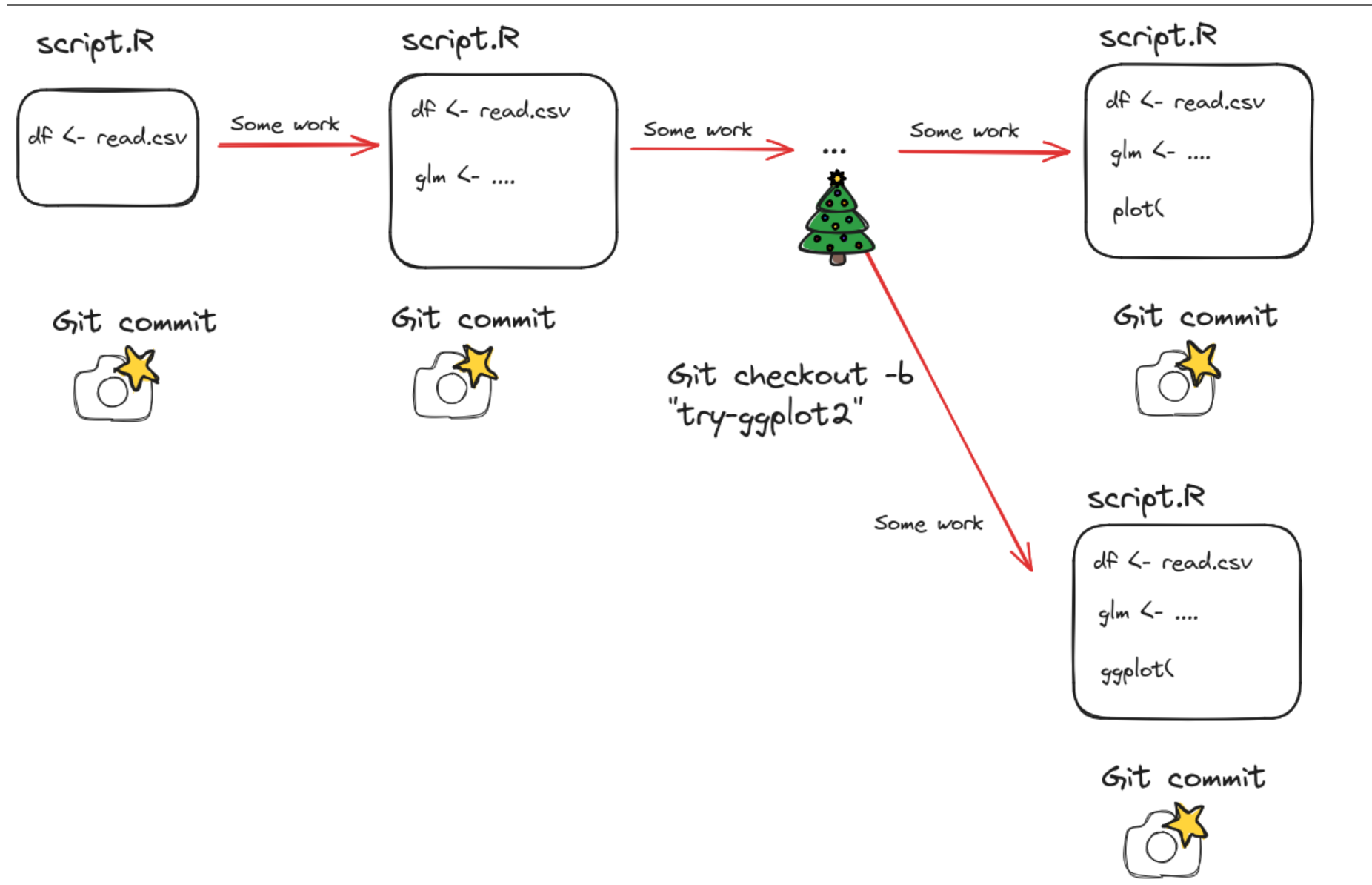
Instantanés



Sans Git

Je veux essayer ggplot2.

Avec Git



Git, c'est pratique

- Moins de perte de travail ;
- Historique à consulter ;
- Essais dans des branches.

Git, la base

- Créer des instantanés avec `git commit`.
- Travailler sur une plateforme comme GitHub.
- Créer des branches pour expérimenter.

Git, la base

- **<https://happygitwithr.com/>**
- **<https://wizardzines.com/zines/oh-shit-git/>**
- **<https://store.wizardzines.com/products/how-git-works>**

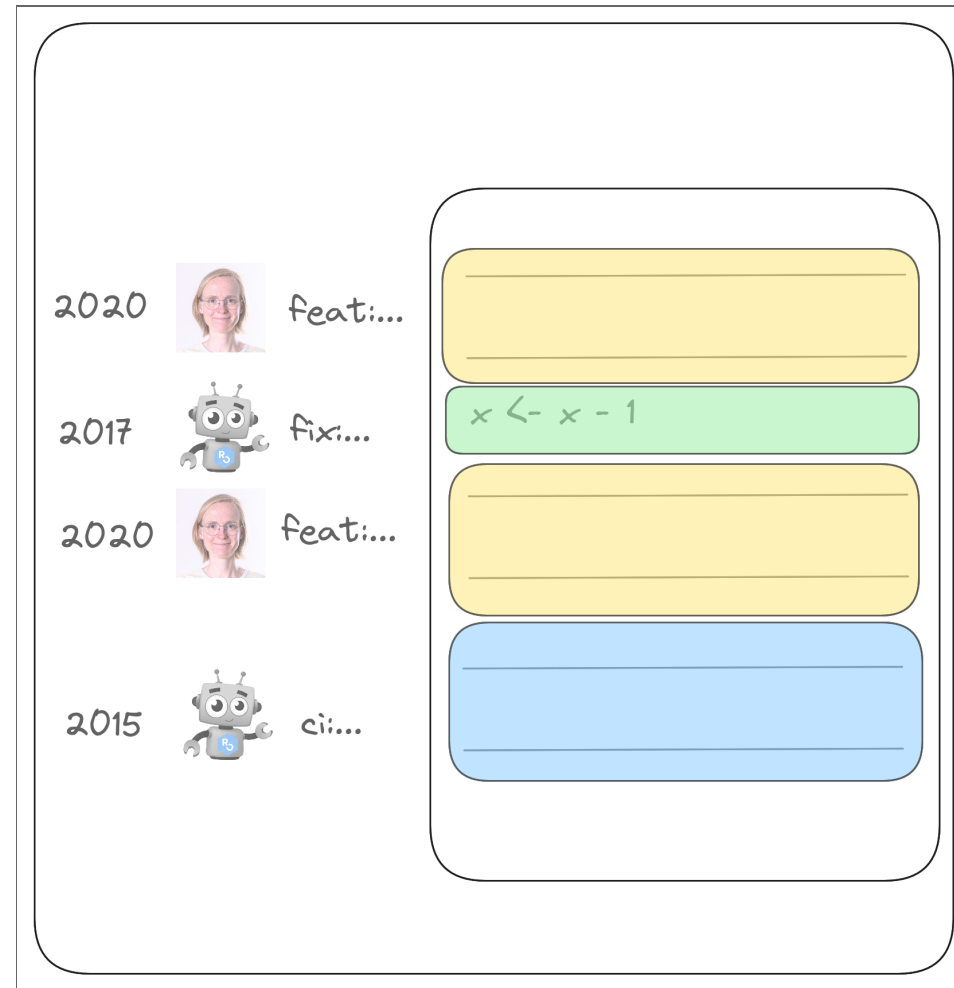
Git, idéal du commit petit et informatif

Mais pourquoi ?!

Une ligne mystérieuse

$$x \leftarrow x - 1$$

Git blame (explain)



Git blame: cliquer sur l'instantané...

Git blame: cliquer sur l'instantané...

“Sauve un tas de fichiers avant le dej 🍝”

145 fichiers modifiés avec 2 624 additions et 2 209 suppressions.

Git blame: cliquer sur l'instantané...

| “fix: adapter le code à l'indexation à partir de 0”

2 fichiers modifiés avec 3 additions et 2 suppressions.

Une mauvaise idée

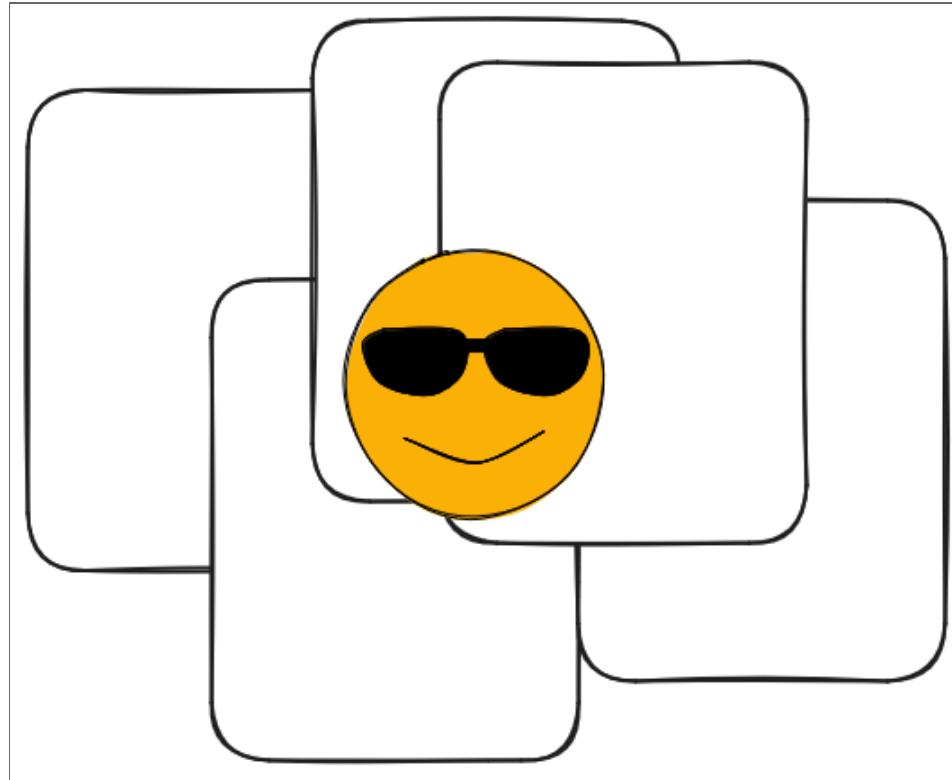
Oh non, cette idée d'il y a 7 instantanés est mauvaise. Est-ce que...

- j'enlève les changements à la main ;
- j'utilise Git revert sur l'instantané pas terrible ?

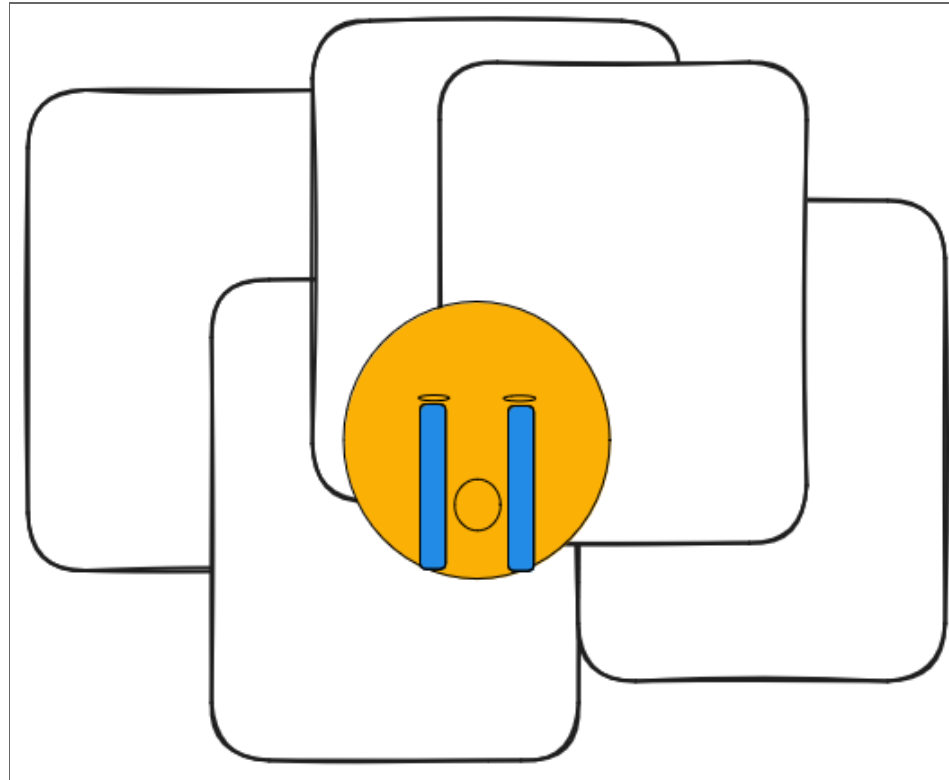
Git revert

Ne marche que si les instantanés sont petits.

Vous faisant un truc avec votre code il y a trois jours



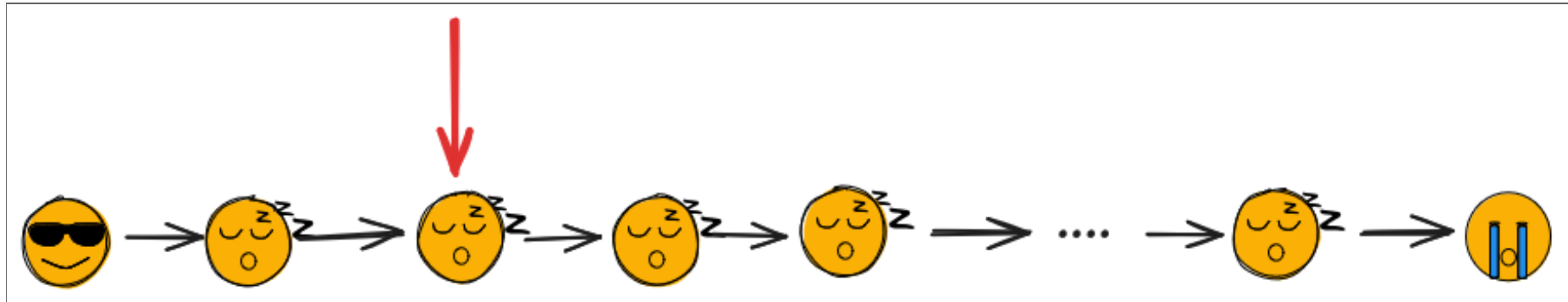
Vous faisant ce truc avec votre code aujourd'hui



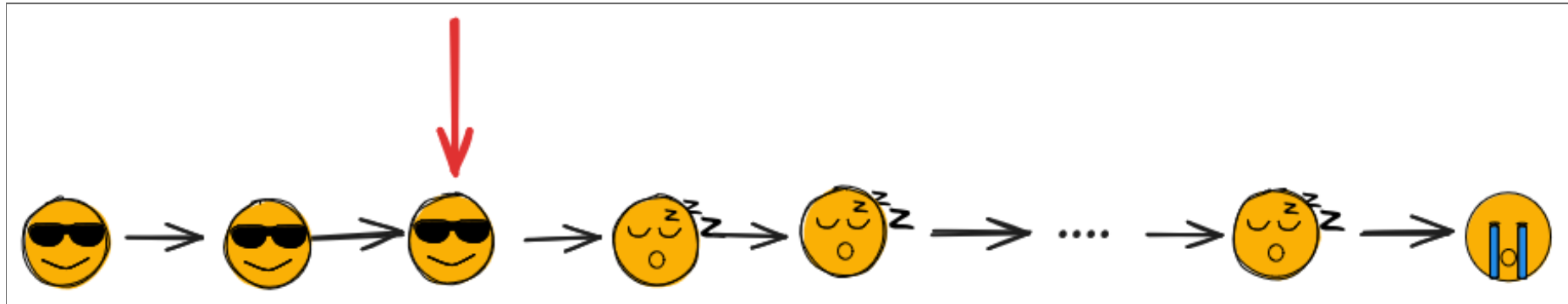
Git bisect: historique Git



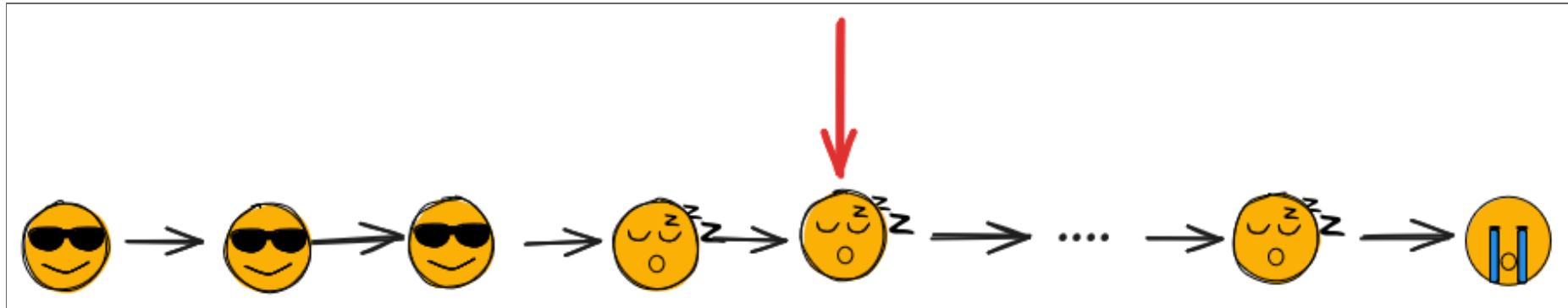
Git bisect: exploration optimale des instantanés



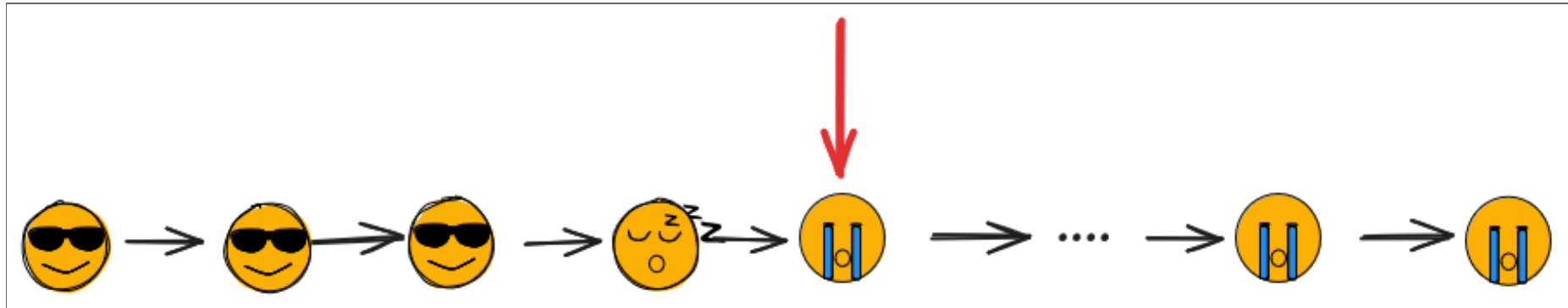
Git bisect: exploration optimale des instantanés



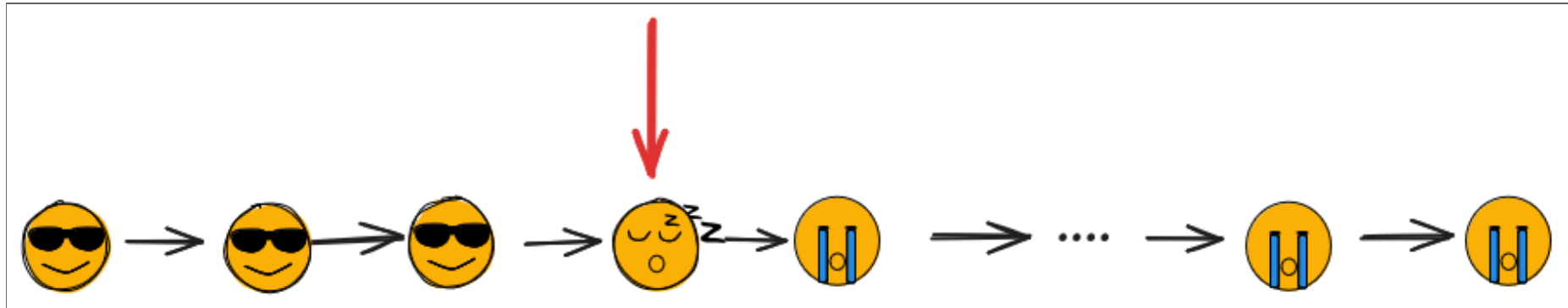
Git bisect: exploration optimale des instantanés



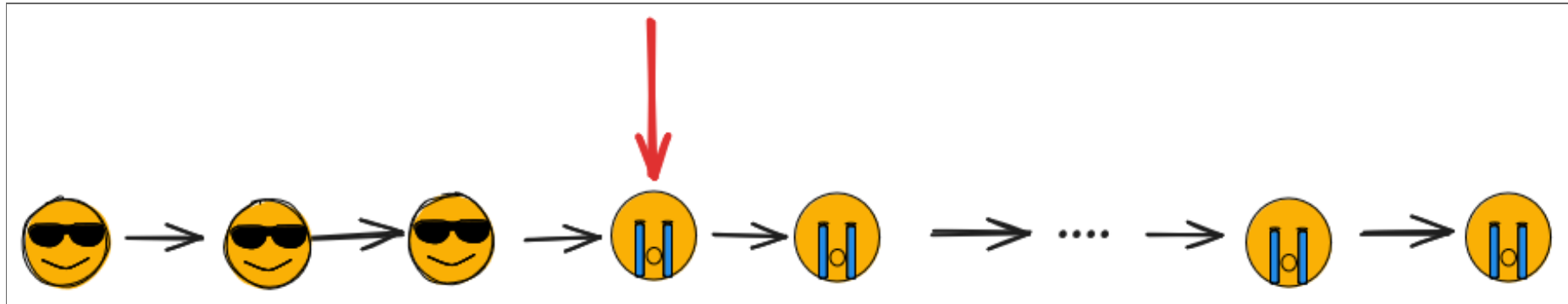
Git bisect: exploration optimale des instantanés



Git bisect: exploration optimale des instantanés



Git bisect: exploration optimale des instantanés



Résultat de Git bisect : un instantané !

Résultat de Git bisect : un instantané !

“Sauve un tas de fichiers avant le sport 💪”

145 fichiers modifiés avec 2 624 additions et 2 209 suppressions.

Résultat de Git bisect : un instantané !

| “refactor: commence à utiliser du YAML”

2 fichiers modifiés avec 3 additions et 2 suppressions.

Git, ce qu'il faut savoir

Git est utile dès le début mais encore mieux...

- Bien créer son historique ;
- Bien l'utiliser.

Mon déclic

Lecture de “Git in practice” de Mike McQuaid

Comment partager ma joie de mieux connaître Git ?

Une idée a germé...

- “Oh shit Git” de Katie Sylor-Miller : documentation de solutions simples à de vrais soucis.
- replex : une manière de reproduire une situation.

Créer des exercices Git reproductibles facilement !

{saperlipopette}

- Un nom inspiré de “Oh shit, Git”, en plus mignon 😊
- 12 exercices Git inspirés de “Oh shit, Git” et de mon expérience personnelle. ✨
- Pas pour les grand·e·s débutant·e·s.

Changer le message du dernier instantané

`saperlipopette::exo_latest_message()`

```
1 library("saperlipopette")
2 parent_path <- "../testdir" # withr::local_tempdir()
3 path <- exo_latest_message(parent_path)
```

Changer le message du dernier instantané

En ouvrant R, dans `path`

```
1 ✖ "Oh shit, I need to change the message on my last commit!"  
2 ✖ I made a typo!  
3 ⓘ See <https://ohshitgit.com/#change-last-commit-message>  
4 ⓘ For more help use tip()
```

Changer le message du dernier instantané

```
saperlipopette::exo_latest_message()
```

```
1 log <- gert::git_log(repo = path)
2 log
```

```
# A tibble: 2 × 6
  commit          author time          files
merge message
* <chr>          <chr>  <dtm>          <int>
<lg1> <chr>
1 d3d35a17a880eee777868a5fbe99ce... Jane ... 2023-12-15 16:25:00      1
FALSE "add a..."
2 e227ecc55e421f70b6e30602e6a2ee... Jane ... 2023-12-15 16:25:00      2
FALSE "First..."
```

```
1 log$message
```

```
[1] "add a new document\n" "First commit\n"
```

Changer le message du dernier instantané

`tip()`, dans `path`

- 1 • Examine Git history.
- 2 • Notice 'documennt' should be 'document'! A typo!
- 3 • ``git commit --amend``
- 4 • Examine Git history.

Changer le message du dernier instantané

Résolution avec les outils de notre choix (Git en ligne de commande, clic-bouton dans RStudio IDE, etc.)

Travail futur : proposer une correction.

Exercices de saperlipopette : Apprendre à faire un meilleur historique

- mettre un changement sur la bonne branche (`git cherry-pick + git reset`),
- réécrire l'historique d'une branche (`git rebase -i`),
- séparer un tas de changements en plusieurs instantanés (`git add --patch`),
- ...

Exercices de saperlipopette : Apprendre à utiliser l'historique

- Enlever les fichiers et dossiers non suivis (`git clean`),
- Chercher l'instantané qui a introduit un bug (`git bisect`),
- Enlever un certain changement (`git revert`).

Pourquoi c'est bien 😊

- Des exercices réalistes avec nos outils habituels ...
- Dans un dossier temporaire où on ne peut rien casser d'important !
- Avec des conseils en plus accessibles via `tip()`

Quelles conditions d'utilisation

- S'entraîner tout·e seul·e ;
- Faire une démo à des apprenant·e·s avant qu'il·elle·s ne refassent l'exercice sur leurs machines.

Comment ça marche ?

- Création de fichiers et d'historique Git. {fs}, {gert}
- Copie d'un fichier en tant que `.Rprofile`, listé dans `.gitignore`. Ce fichier a les messages d'entrée et la fonction `tip()` !

Saperlipopette, entraînez-vous à Git !

<https://rr2024-maelle.netlify.app/>

<https://maelle.github.io/saperlipopette>