

# Prise en main de GitHub

Anissa Saumtally

02-01-2023

*“Github est votre ami, exigeant mais indispensable” XT (2022)*

# Objectif de la session :

## Démystifier GitHub

- Se familiariser avec le processus de travail avec GitHub
- Installer l'outil pour une prise en main rapide
- Être capable de travailler sur un projet existant sur GitHub

# Qu'est-ce que Git(Hub) ?



Préface : Il existe plusieurs plateformes qui permettent de travailler avec Git. On utilisera ici GitHub :

- Parce que c'est là que l'organisation OFCE sur Git est installée
- Fonctionnalités user-friendly : application plutôt que lignes de commandes, outils collaboratifs...
- “Git” permet de travailler avec un système de **version control**
  - On garde en mémoire plusieurs instances choisies d'un travail
  - On peut suivre les changements effectués
  - On peut comparer les instances des fichiers entre elles

# Pourquoi versionner son travail ?

Sans Git

^	Size	Kind
22_EA_AS_final_2.qmd	29 KB	Quarto...
22_EA_AS_final_VRAIE.qmd	29 KB	Quarto...
22_EA_AS_Final.qmd	29 KB	Quarto...
22_EA_AS.qmd	29 KB	Quarto...
22_EA.qmd	29 KB	Quarto...

# Qu'est-ce que Git(Hub) ?



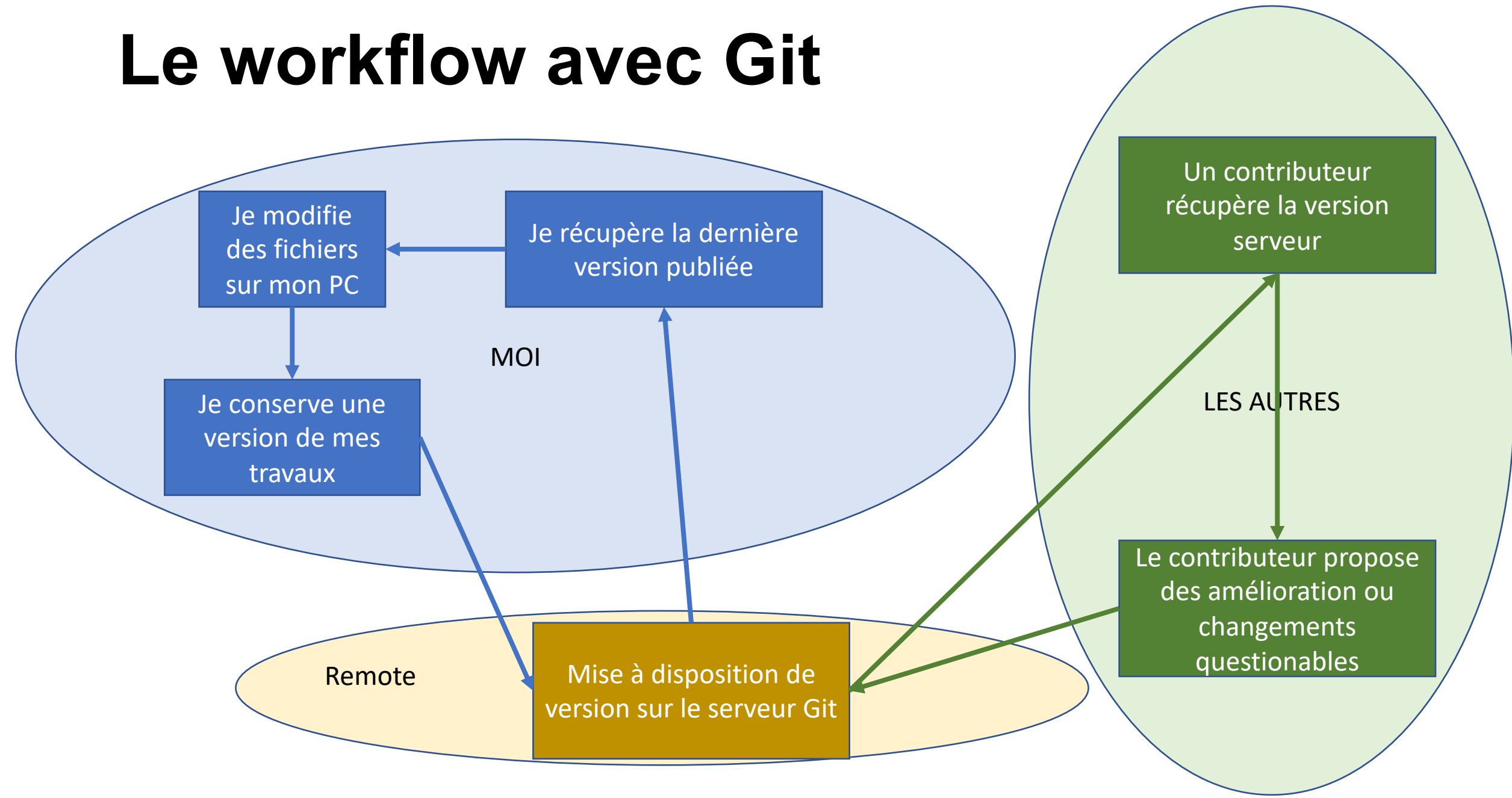
- Un outil utile pour soi-même ...
  - L'ensemble d'un projet "**repository**" dans un seul endroit, stocké sur le cloud "**remote**", qui garde un historique des différentes versions
- ... Mais aussi pour le travail collaboratif
  - Un même dossier, mais chacun sur sa version en local
  - On choisit le moment où l'on souhaite publier ses changements
  - Git se charge de rassembler (tant qu'il peut) les versions des utilisateurs
  - Quelques fonctionnalités intéressantes : le mode projet, les issues tickets ..

# Quels types de travaux ?



- Tout type de travail impliquant du code, dans n'importe quel langage :
  - LaTeX, R, Quarto, Markdown, Eviews, Html ...
- Travail seul comme à plusieurs :
  - Le vous de demain n'est pas la même personne que le vous d'avant-hier
- Versionning limité pour les fichiers type Excel et Word :
  - On ne pourra pas visualiser directement les changements
  - La gestion de versions reste cependant plus propre : pas de fichiers multiples aux 150 suffixes
  - Pas de conflits à l'ouverture simultanée des fichiers : chacun dispose d'une version en local
  - La publication est choisie par les utilisateurs : possibilité de garder un ordre dans les modifications des fichiers
- Et bien plus encore : Sites Web directement mis à jour et publiés via Git

# Le workflow avec Git



# Le jargon GitHub

- **Repository (ou “repo”)** : Le dossier de travail d'un projet qui contient l'ensemble des scripts et autres fichiers annexes qui va être géré par Git (avec son historique de versions)
  - *“L'ensemble des graphiques du PB sont disponibles sur le repo du PB éponyme”*
- **Commit** : les différentes instances du travail sauvegardées par un utilisateur, peut s'utiliser en verbe pour désigner l'action de créer un commit.
  - *“Mon dernier commit contient les changements de couleurs sur les graphiques et un nouveau paragraphe”*



# Le jargon GitHub

- **Remote** : La version sur le cloud du repository visible pour ceux qui y ont accès
- **Push** : L'envoi de ses commit sur le remote pour mise à disposition générale
  - *“J’ai push(-é?) mes derniers commits sur le remote, tu peux récupérer la dernière version”*
- **Pull** : La récupération de la dernière version du remote
  - *“Tu peux faire un pull pour voir les derniers résultats que j’ai pushé à l’instant”*
- **Branche** : Fonctionnalité permettant de travailler sur des versions alternatives d’un même repo. Chaque branche est hermétique : les avancées des autres branches ne sont pas visibles et n’interviendront qu’à la demande explicite de l’utilisateur (**“merge”**)
  - Ex : *“La branche ‘GOAT’ propose une version interactive des graphiques de la branche Master”*

# Les étapes du stockage d'un fichier



**J'ai modifié le script, il est  
sauvegardé sur mon ordinateur  
dans le répertoire de travail**

Stage

Commit to main

**Je souhaite archiver cette  
version du script sur au moins  
mon ordinateur**

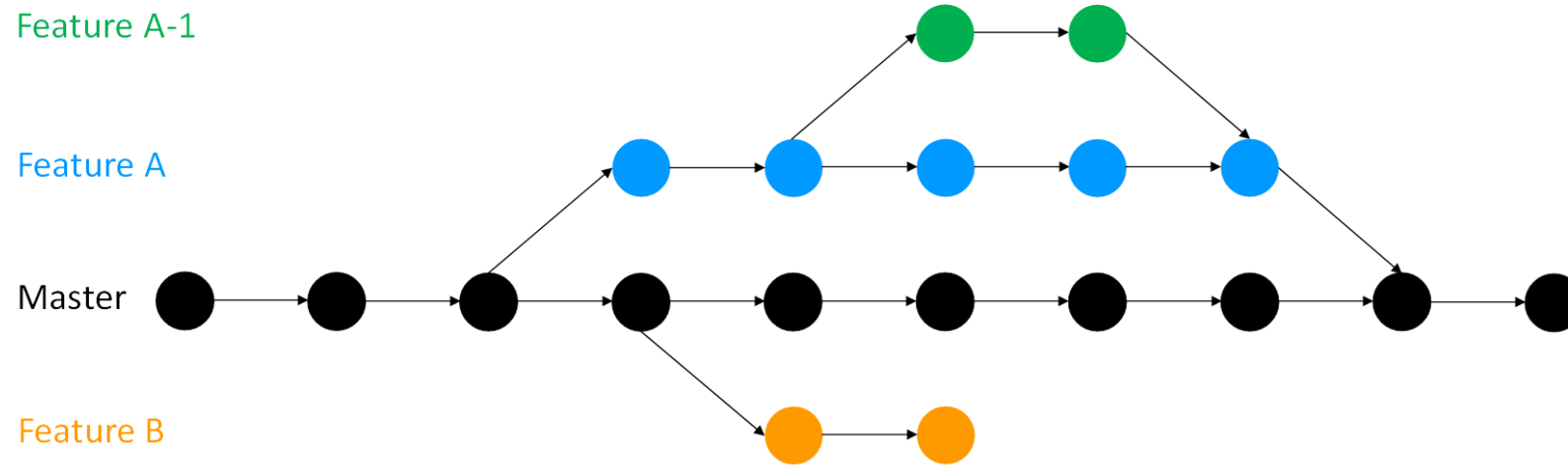
Commit

↑ **Push origin**  
Last fetched 16 minutes ... 1 ↑

**Je veux publier cette version  
pour la rendre accessible aux  
autres via le cloud (stockage  
sur le remote)**

Pushed

# Le concept de branches Git



- La branche “Master” sert de branche de référence
- Une branche permet de travailler sur des projets précis qui pourront être ajoutés à la branche Master (ou pas) (Feature A, Feature A-1)
- Certaines branches peuvent servir à faire des tests sans avoir vocation à être intégrées à la branche Master (Feature B)

# Le travail collaboratif

Quelques fonctionnalités :

- Un repository peut être public (le monde y a accès) ou privé (seuls les membres de l'organisation ou moi-même y ont accès)
  - Fonctionnalités limitées en mode privé
- Différents niveaux de permissions peuvent être établis (ie: restrictions d'écriture sur les branches)
- Le mode projet et les issues tickets

# GitHub en mode gestion de projet

<> Code Issues 23 Pull requests 2 Actions **Projects 4** Wiki Security Insights Settings

**R-ThreeME**  
Updated on Dec 1, 2022

Filter cards

+ Add

7 To do

• Add bridge name in R\_calibration.mdl et R\_list

#36 opened by fredericreynes

• Improvement of fonction contrib (ermeeth)

#34 opened by fredericreynes

• Calibration: missing variables

#33 opened by fredericreynes

enhancement

• @over by default in dynamo

#31 opened by fredericreynes

2 In progress

• R solver: Correct program when calibration error

#28 opened by fredericreynes

• Add export\_to\_eviews function in Ermeeth

#30 opened by fredericreynes

2 Done

✓ \*\*Dynamo bugs: Condition in Equations: \*\*

#25 opened by fredericreynes

✓ Change file structure

#26 opened by fredericreynes

# Le travail collaboratif

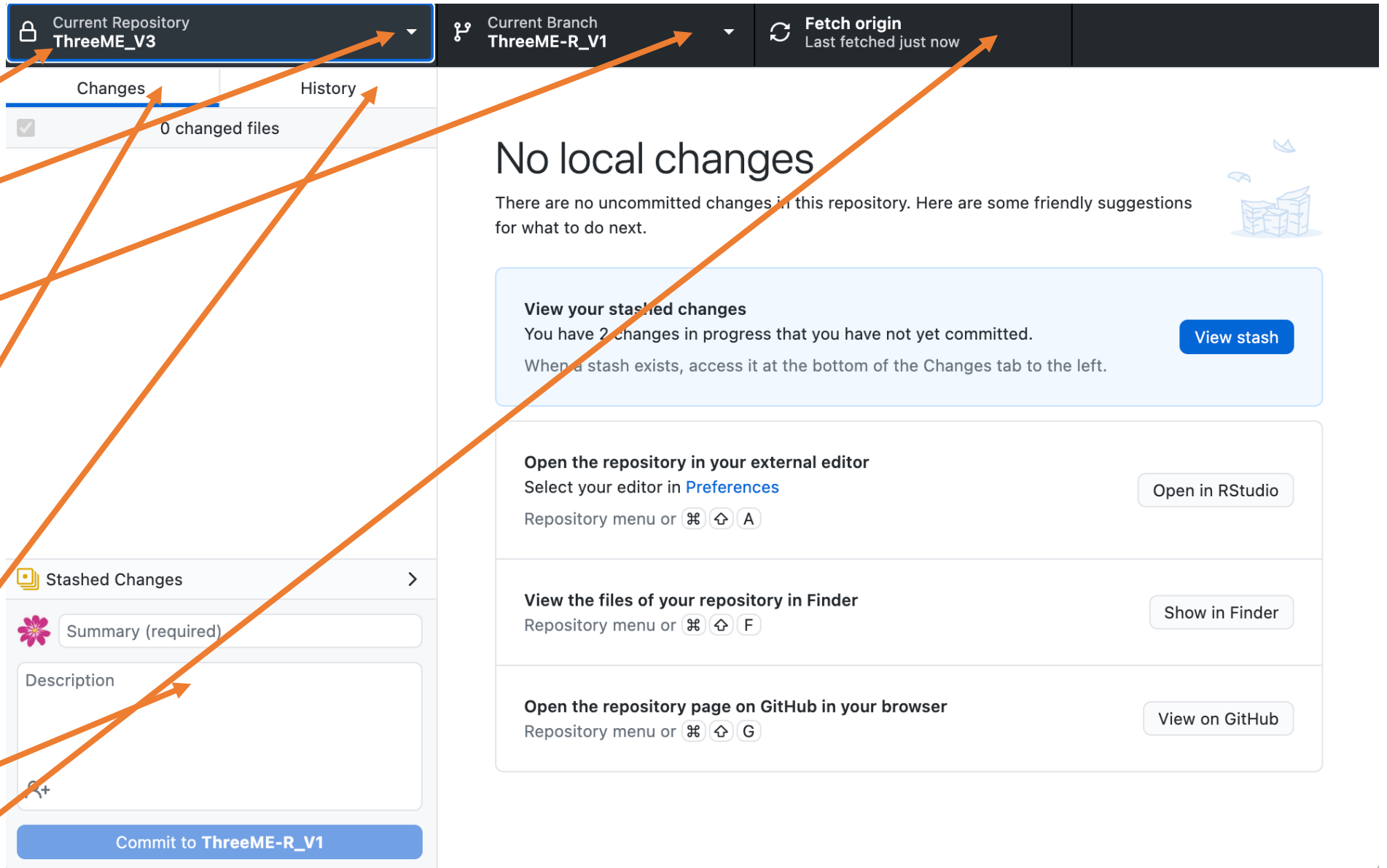
- À la récupération des fichiers (Pull) après un Push venant d'une autre personne, Git essaie de compiler les changements pour n'avoir qu'un seul fichier
    - Souvent ça fonctionne, et les changements des contributeurs apparaissent dans mes fichiers
    - Parfois ça échoue, Git ne sait pas comment intégrer les deux versions : cela crée un conflit qu'il faut absolument résoudre avant de pouvoir continuer le travail (on verra ça dans la démo)
- Assurance d'avoir toujours une version cohérente du repo.

# Installation de GitHub

- Si cela n'est pas déjà le cas : créer un compte sur [www.github.com](https://www.github.com)
  - Optionnel : choisir un avatar sympa
  - Si vous n'êtes pas membre de l'organisation GitHub **“OFCE Sciences Po Paris”**, **nous faire signe pour recevoir une invitation**
- Télécharger et installer GitHub Desktop : <https://desktop.github.com>

# L'interface GitHub Desktop

- Repository ouvert
- Voir la liste de mes repositories
- Les différences branches du repository ouvert
- Les changements sauvegardés en local par rapport au dernier commit
- L'historique des changements sur la branche
- Faire un commit
- Mettre à jour (push/pull)








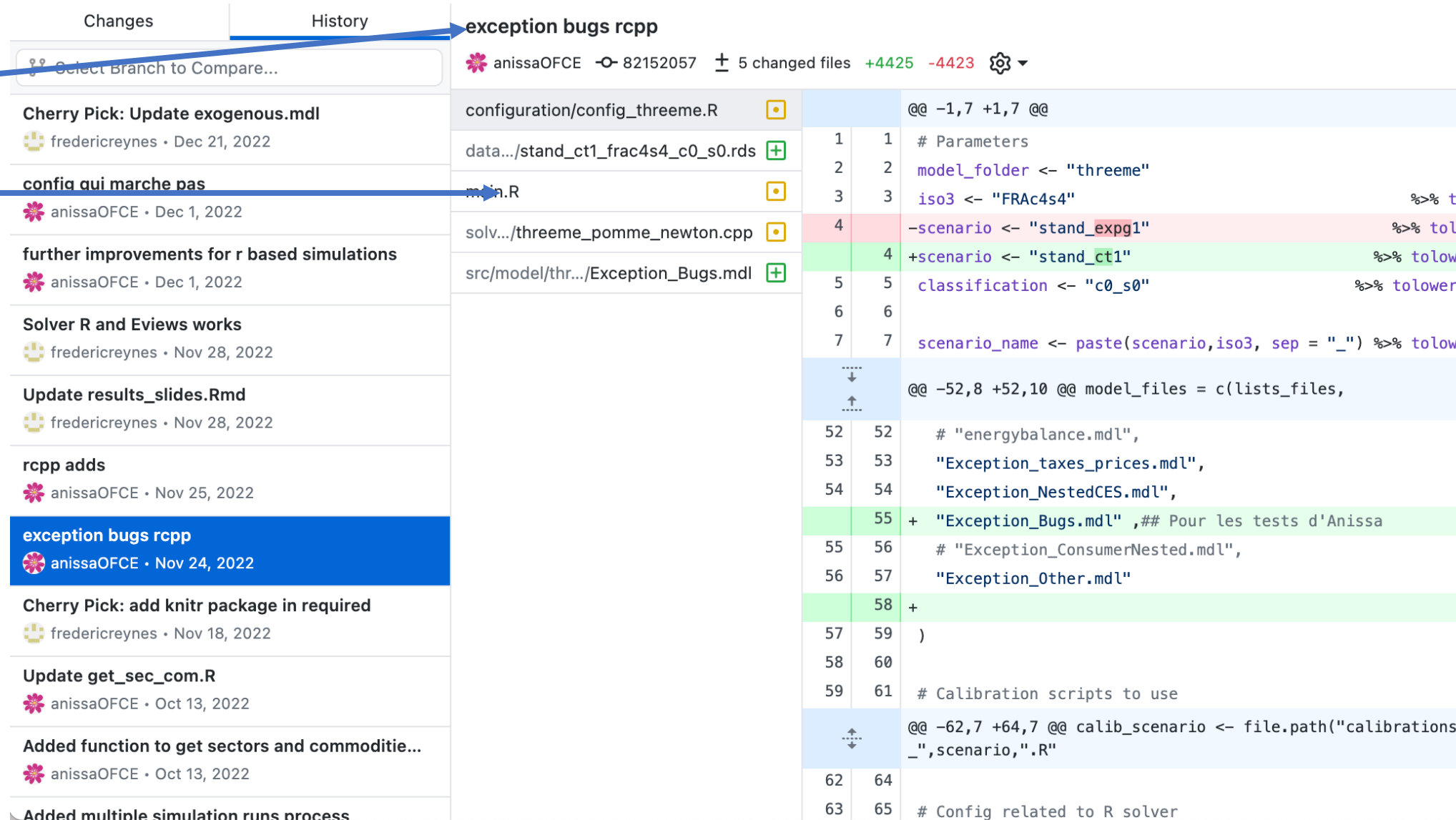
# Visualiser les changements de version

- Nom de la version (nom du commit)

- Liste des fichiers modifiés dans cette version

-  : ce fichier a été modifié
-  : ce fichier a été ajouté
-  : ce fichier a été supprimé

- Ligne surlignée verte : ajout de texte dans le fichier
- Ligne surlignée rouge : suppression de texte dans le fichier



Changes History

Select Branch to Compare...

Cherry Pick: Update exogenous.mdl  
fredericreynes • Dec 21, 2022

config qui marche pas  
anissaOFCE • Dec 1, 2022

further improvements for r based simulations  
anissaOFCE • Dec 1, 2022

Solver R and Eviews works  
fredericreynes • Nov 28, 2022

Update results\_slides.Rmd  
fredericreynes • Nov 28, 2022

rcpp adds  
anissaOFCE • Nov 25, 2022

**exception bugs rcpp**  
anissaOFCE • Nov 24, 2022

Cherry Pick: add knitr package in required  
fredericreynes • Nov 18, 2022

Update get\_sec\_com.R  
anissaOFCE • Oct 13, 2022

Added function to get sectors and commoditie...  
anissaOFCE • Oct 13, 2022

Added multiple simulation runs process

exception bugs rcpp  
anissaOFCE 82152057 5 changed files +4425 -4423

configuration/config\_threeme.R

data.../stand\_ct1\_frac4s4\_c0\_s0.rds

src/model/thr.../Exception\_Bugs.mdl

@@ -1,7 +1,7 @@

# Parameters

model\_folder <- "threeme"

iso3 <- "FRAc4s4"

-scenario <- "stand\_expg1"

+scenario <- "stand\_ct1"

classification <- "c0\_s0"

scenario\_name <- paste(scenario, iso3, sep = "\_")

@@ -52,8 +52,10 @@ model\_files = c(lists\_files,

# "energybalance.mdl",

"Exception\_taxes\_prices.mdl",

"Exception\_NestedCES.mdl",

+ "Exception\_Bugs.mdl" ,## Pour les tests d'Anissa

# "Exception\_ConsumerNested.mdl",

"Exception\_Other.mdl"

)

# Calibration scripts to use

@@ -62,7 +64,7 @@ calib\_scenario <- file.path("calibrations", scenario, ".R")

# Config related to R solver

# Démarrage rapide : créer un repository

Current Repository  
PlopTest

Filter

Recent

ThreeME V3

Add

Current Branch  
main

Clone Repository...

Create New Repository...

Add Existing Repository...

1 – Je crée un nouveau repository

2 – Je publie ce repository en appuyant sur ce bouton



**Publish repository**

Publish this repository to GitHub

Publish Repository



GitHub.com

GitHub Enterprise

Name

PlopTest

Description

☒ Keep this code private

Organization

✓ None

fosem

OFCE

ThreeME-org

Create a New Repository

Name

NouvoRepO

Description

Local Path

/Users/139804/Work

Choose...

☐ Initialize this repository with a README

Git Ignore

None

License

None

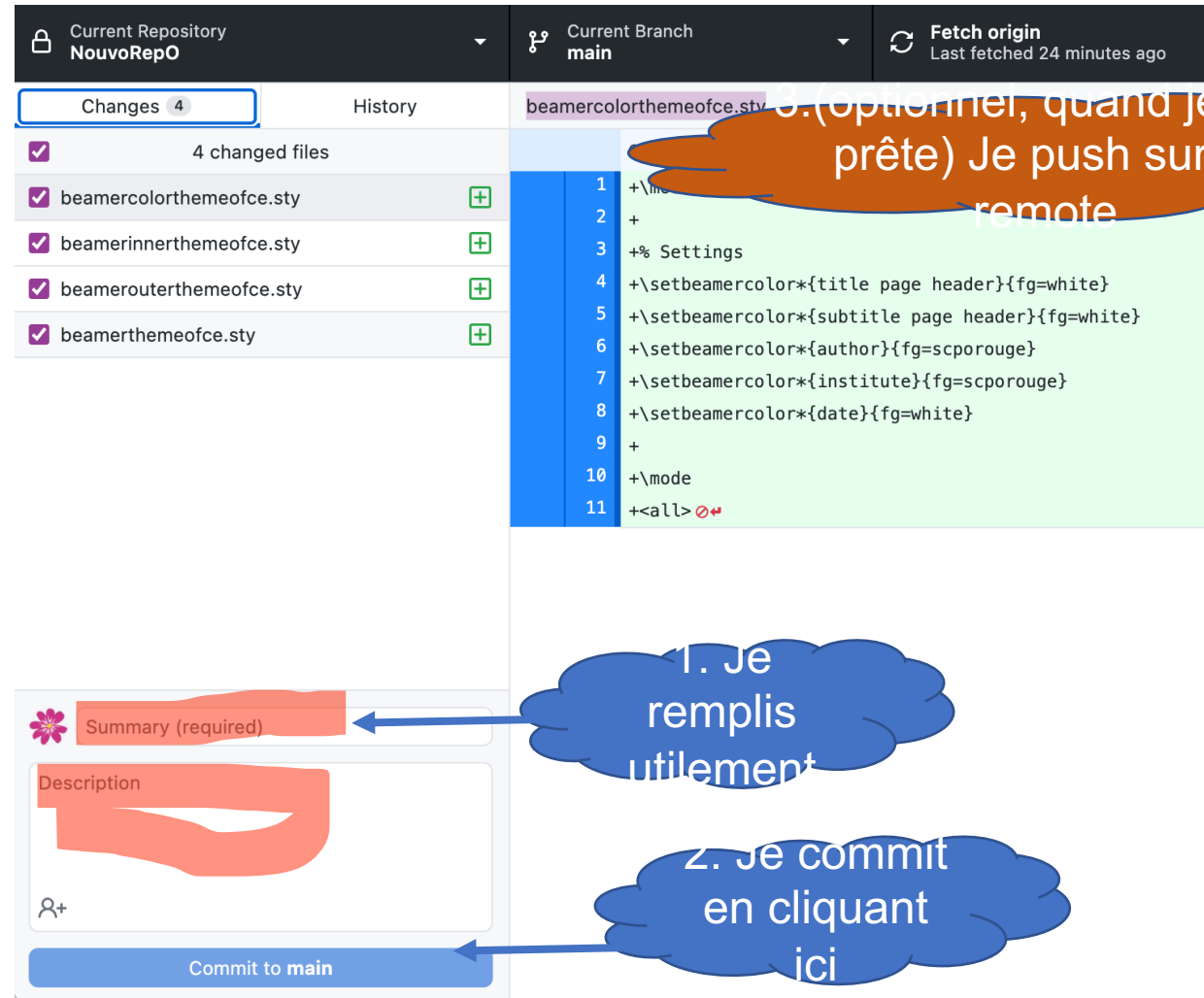
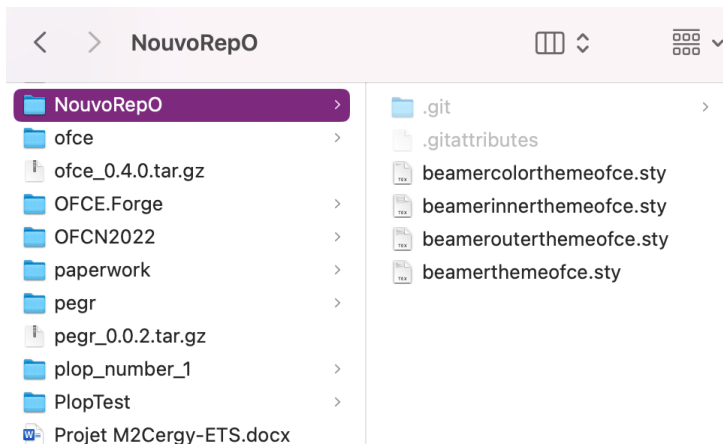
Cancel

Create Repository

Emplacement  
du dossier sur  
mon ordinateur

# Démarrage rapide : mon premier commit

- Je créer un fichier que je mets dans mon dossier local du repository
- Les changement du repository apparaissent dans l'onglet "changes"



# Des bonnes pratiques à avoir

- Faire des commit fréquents et utiles
  - Ne pas oublier de commenter et de nommer ses commits de façon utile
- Vérifier régulièrement les ajouts des collaborateurs pour ne pas être trop en retard
- Communiquer pour éviter les conflits
  - Utiliser un éditeur de script (Visual code)
- Pour faire des tests ou pour un première prise en main : chacun sa branche
- Pas obligé de tout commit :
  - Rejet de changement
  - .gitignore

Update results\_slides.Rmd  
fredericreynes • Jun 8, 2022

apres formation  
anissaOFCE • Jun 8, 2022

Update results\_slides.Rmd  
fredericreynes • Jun 8, 2022

Update results\_slides.Rmd  
fredericreynes • Jun 7, 2022

Update results\_slides.html  
fredericreynes • Jun 7, 2022

Update results\_slides.Rmd  
fredericreynes • Jun 7, 2022

New data  
fredericreynes • Jun 7, 2022

Update codenames\_c25\_s26.R  
fredericreynes • Jun 7, 2022

Update results\_slides.Rmd  
fredericreynes • Jun 5, 2022

Update results\_slides.Rmd  
fredericreynes • Jun 5, 2022



Simplify model path  
fredericreynes • Sep 12, 2022

Clean up  
fredericreynes • Sep 11, 2022

Start result training  
fredericreynes • Sep 11, 2022

Update 02.1-calib.mdl  
fredericreynes • Sep 11, 2022

Load list file separately  
fredericreynes • Sep 11, 2022

Correct calib in 02.1-calib.mdl  
fredericreynes • Sep 11, 2022

Add list file in training  
fredericreynes • Sep 11, 2022

Add Walrasian model  
fredericreynes • Sep 11, 2022

Update config and main  
fredericreynes • Sep 11, 2022

Change traning model file names  
fredericreynes • Sep 11, 2022



# La démo en live

- Recap des interfaces : Web et Desktop (bonus : Rstudio)

## Mode Solo :

- Créer un repository
- Faire plusieurs commit et push
- Récupérer un repository existant
- Créer une branche

## Mode Coopératif (ou pas) :

- Voir et récupérer les ajouts des autres
- Gérer un conflit
- Merger une branche
- Travail préliminaire sur la publication (avec <https://hypothes.is/> )

# Git : La bonne résolution à tenir pour 2023 !

- N'hésitez pas à vous lancer !
- Aide sympathique et sans jugement, disponible au bureau ThreeME
- Merci pour votre attention 