

# Projet : R avancé et introduction à Git

Raphaël Nedellec

## Projet de fin de classe

L'objectif du projet est le suivant :

- manipuler les différents concepts R vu lors des différents cours : fonctions et packages, documents `quarto`, et études reproductibles avec `renv` notamment.
- vous faire interagir avec git et github pour vous familiariser avec les outils et vous donner à voir aux potentialités de ces outils.

L'utilisation de git doit se faire de manière continue. Néanmoins, certaines questions vous demanderont d'exécuter certaines commandes à un moment donné. Ce n'est pas pour ça qu'il faut se restreindre à ces moments précis.

## Rendu

Le dossier projet sous github et le site seront le rendu, ainsi qu'un document `.qmd` compilé dans lequel vous préciserez les réponses aux questions qui ne pourront pas figurer dans le code.

- Le site devra être conforme aux demandes des questions suivantes.
- Le dossier projet devra être structuré comme attendu.

La date finale du rendu est le 10 mars 2024.

## Création d'un blog R quarto

À la fin du projet, vous aurez créé un site de blog perso en quarto, hébergé sur github. Votre premier post de blog sera issu d'un travail reproductible généré en R à partir du package développé dans les séances précédentes.

## Création du projet de site sous github

1. Créez un nouveau projet sous RStudio, que vous appellerez `my_blog`. Ce projet sera créé en utilisant le template `quarto blog`. Cochez les options `renv` et `git` pour les utiliser. Décrivez la structure du projet.
2. Personnalisez les documents `index.qmd` et `about.qmd a minima`. De la même manière, modifiez le post `posts/welcome/index.qmd`.
3. Créez un nouveau projet github, que vous appellerez `my_blog`.
4. De retour sur RStudio, ouvrez le terminal. Configurez les paramètres `user.name` et `user.email` en utilisant les instructions `git config user.name "<votre username>"` et `git config user.email "<votre email>"`. Vous pouvez utiliser le paramètre `--global` pour modifier ces valeurs pour tous vos projets par défaut `git config --global user.name "<votre username>"`.
5. Toujours dans le terminal, configurez le `remote` de manière à le faire pointer vers le projet github récemment créé. L'instruction devrait ressembler à `git remote add git@github.com:<votre username github>/my_blog.git` si vous utilisez un accès `ssh`.
6. Faites un `commit` de l'ensemble des fichiers, puis un `push`. Que voyez-vous désormais sous github ?
7. Quittez le projet RStudio avant de passer à la section suivante.

## Un package sous github

1. Reprenez votre package `olympicsWeather` développé dans l'exercice de la semaine 4. Votre package doit être fonctionnel et pouvoir s'installer ! Ouvrez le projet associé au package.
2. Créez un nouveau projet github, que vous appellerez `olympicsWeather`.
3. Initiez un projet git à l'aide de la commande `git init`. Configurez le `remote` de manière à le faire pointer vers le projet github récemment créé. L'instruction devrait ressembler à `git remote add git@github.com:<votre username github>/olympicsWeather.git`.
4. Ajoutez l'ensemble des fichiers du dossier R à la pile des fichiers en utilisant la commande `git add`.
5. Faites un premier `commit` intitulé "code R" en utilisant la commande `git commit -m "<insérez les infos du commit ici>"`. N'oubliez pas de faire un message informatif.
6. Ajoutez un fichier `.gitignore` dans lequel vous excluez les fichiers `'.Rproj.user'`, `'.Rhistory'`, `'.Rdata'`, `'.httr-oauth'`, `'.DS_Store'`.
7. Faites un `git status`. Ajoutez les fichiers restants à la pile avec un `git add`, puis faites un nouveau `commit`.
8. Synchronisez votre projet local avec le répertoire distant à l'aide de la commande `git push origin main`.
9. Fermez le projet RStudio et redémarrez un nouveau terminal R. Depuis ce terminal, installez le packages `remotes`. Que se passe t'il quand vous exécutez la commande

```
remotes::install_github("<votre username>/olympicsWeather")      (remplacez  
<votre username> par votre username github) ?
```

## Posts de blog

1. Ouvrez à nouveau votre projet de blog, `my_blog`. Ouvrez le fichier `posts/post-with-code/index.qmd`. Vérifiez que `renv` est bien actif en utilisant la fonction `renv::status()`. Installez le package `c("quarto")`.
2. Vous créez une première section de niveau `h2`, soit avec deux `##`, intitulée `Librairie olympicsWeather`. Après avoir introduit et présenté le package, vous décrirez comment installer le package en utilisant le package `remotes` depuis github. Vous utiliserez un bloc de code pour décrire les commandes. Attention, le code ne devra pas être exécuté !
3. Installez le package `olympicsWeather` depuis votre dépôt github en utilisant le package `remotes`. Utilisez `renv::snapshot()` et vérifiez que le package `olympicsWeather` est bien mentionné dans votre fichier `renv.lock` à la racine de votre projet de blog.
4. Modifiez votre post de blog `posts/post-with-code/index.qmd` de manière à ajouter une section pour chacune des requêtes suivantes : quelles sont les prévisions météo à 7 jours pour le stade de France ? Pour Tahiti ? Pour les coordonnées gps 43.276703, 5.334791 ?
5. Ajoutez un fichier `.gitignore` à la racine du dossier `posts/post-with-code/` pour ne pas traquer les fichiers indésirables comme `.Rhistory`, ou autres.
6. Dans le terminal ou dans la console R, faites une étape de preview de votre site complet. Par exemple, on pourra utiliser la commande `quarto preview` dans le terminal.
7. Vérifiez que vous ajoutez bien tous les documents et les dossiers à votre commit. Faire un commit, en particulier assurez-vous que votre dossier `_site` est bien commit.
8. Faites un git push et vérifiez que votre projet est à jour sur github.
9. Configurez github pages. Assurez-vous que votre projet est public. Configurez votre github pages à la manière de :

Assurez-vous que les pages pointent vers le sous dossier `_site/` et non `root`.

Si tout va bien, vous devriez avoir un github pages setup et un site web accessible ! Félicitations.

### Exercices des semaines 1 à 3

1. Créez un nouvel article de blog pour chacun des exercices des semaines 1 à 3.
2. Répétez le processus décrit précédemment pour uploader et commit l'ensemble des articles relatifs à ces exercices. N'oubliez pas de `quarto render` ou de `quarto preview` avant le dernier commit pour pousser une version à jour du site contenue dans le dossier `_site`.

### Bonus

Le processus utilisé dans le projet nécessite que le site soit construit préalablement au commit. Il est possible d'utiliser une `github action` pour construire le site dynamiquement à chaque commit, et faire en sorte que le code soit exécuté à nouveau à chaque modification de la branche principale. Essayez de mettre ce processus en place en utilisant les instructions décrites ici <https://quarto.org/docs/publishing/github-pages.html#example-knitr-with-renv>. Ceci à l'avantage de ne pas nécessiter l'exécution préalable et donc allège les commits, tout en assurant que le site soit toujours à jour !