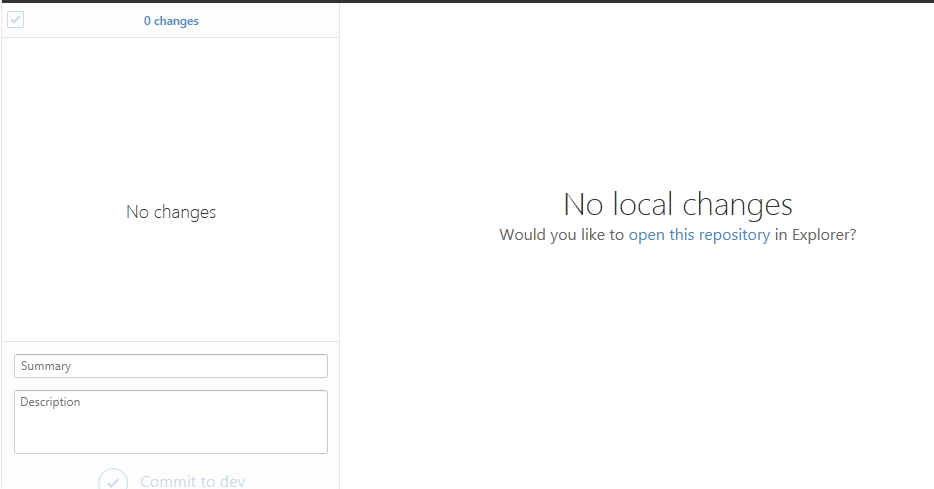
# Work Flow d’un déploiement sur origin/Master

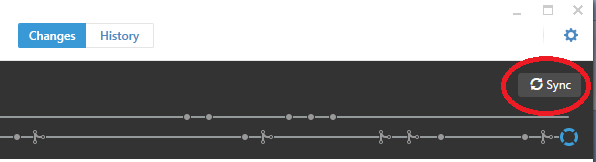
1. Se mettre à jour avec origin/dev

A . Avoir au préalable commité ses dernières modifications

**

B. Se mettre à jour

* Via l’interface GitHub avec le bouton de Synchronisation.

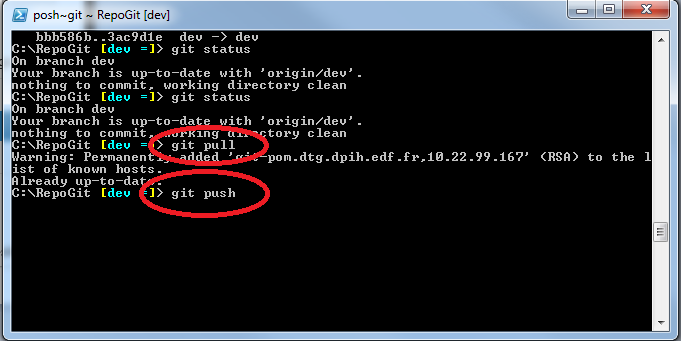
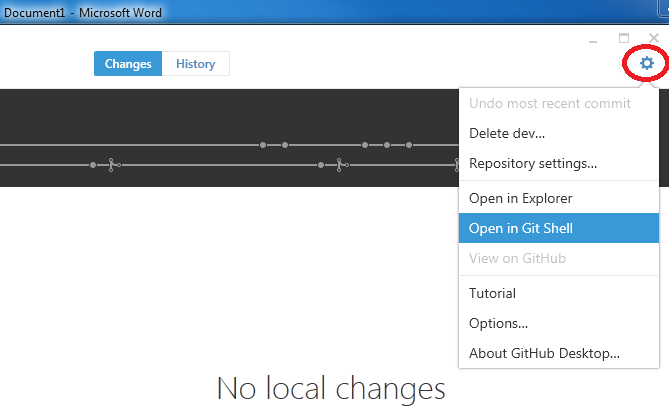


**Ou**

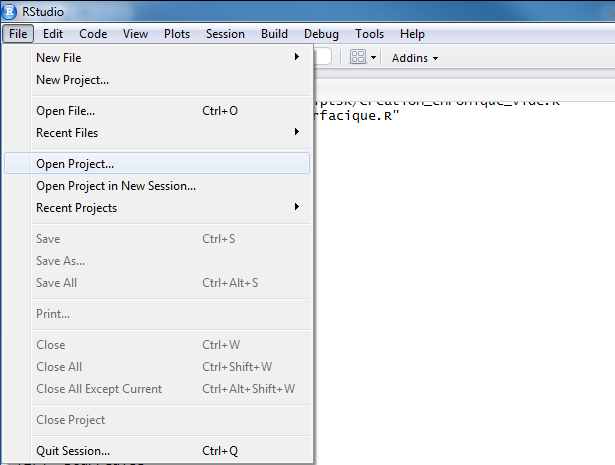
* Via le shell Git avec les commandes « Git pull ; Git push ».

**1**

**2**

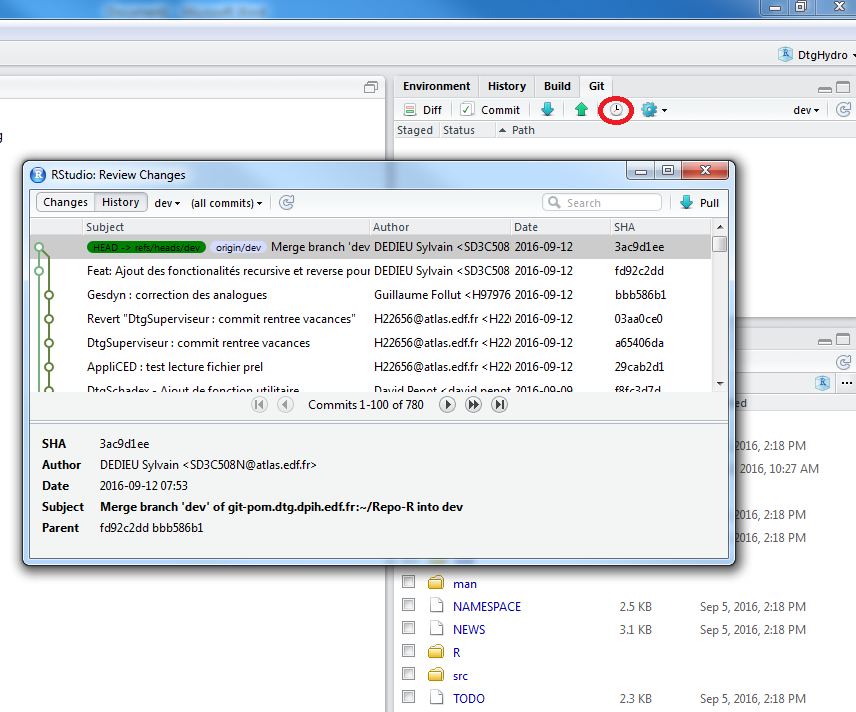
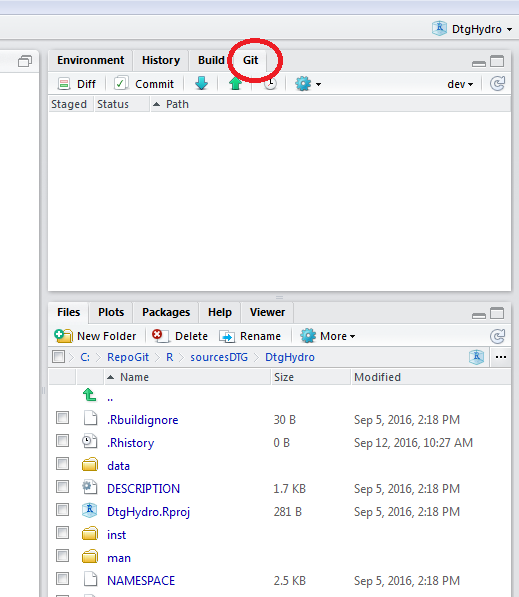


1. Ouvrir le projet concerné dans R Studio



1. Regarder l’historique des changements dans l’onglet « git », en sélectionnant le répertoire du projet

**1**

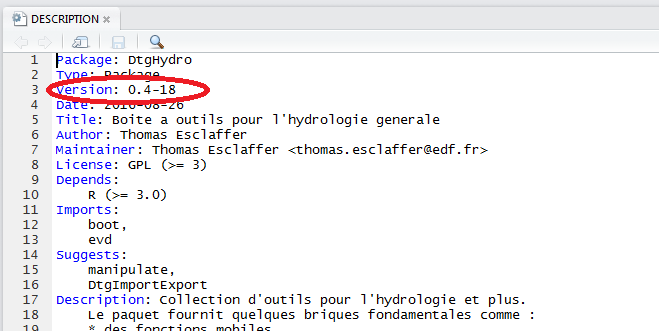
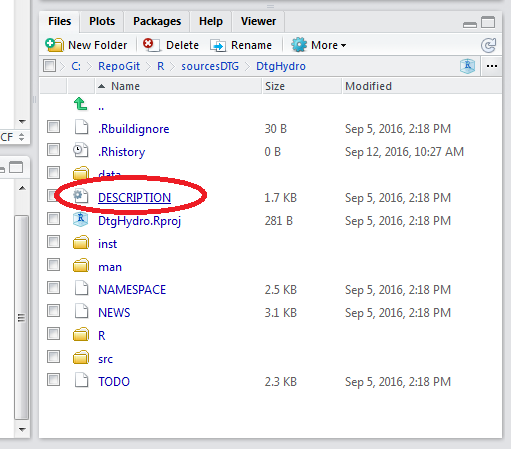


**2**

1. Vérifier que la description (n° de version) est à jour ainsi que le fichier NEWS
   1. Dans le fichier « DESCRIPTION »

**1**

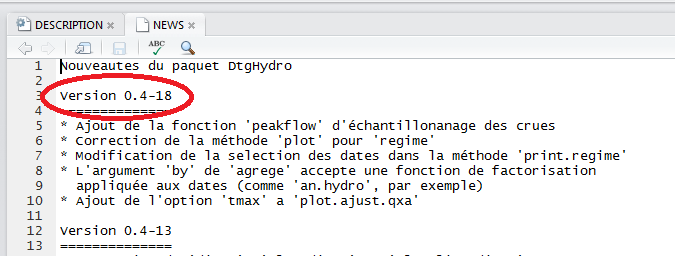
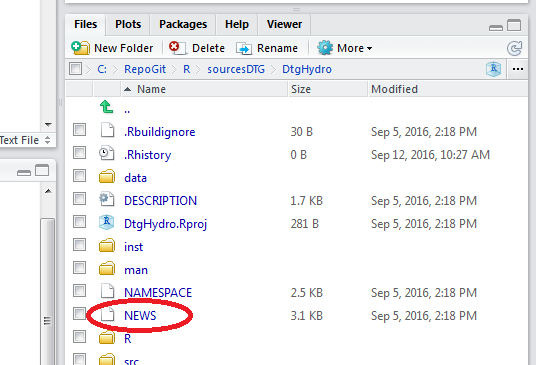
**2**



* 1. Dans le fichier « NEWS »

**1**

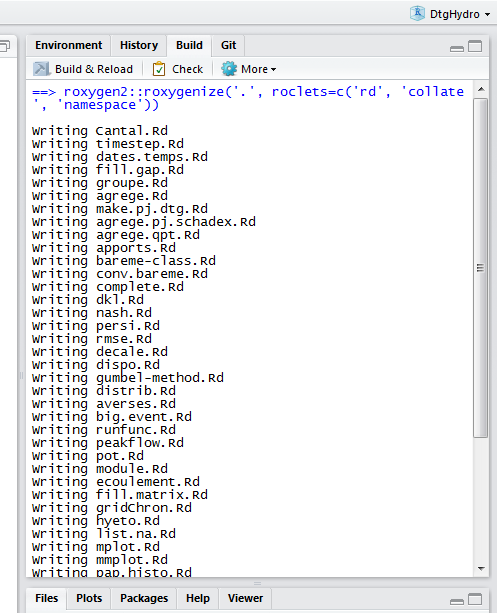
**2**



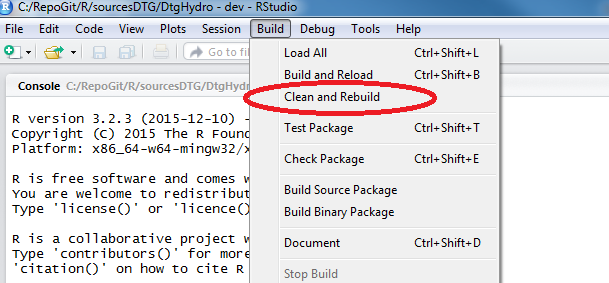
1. Relancer la documentation automatique (Ctrl + Shift + D)

**1**

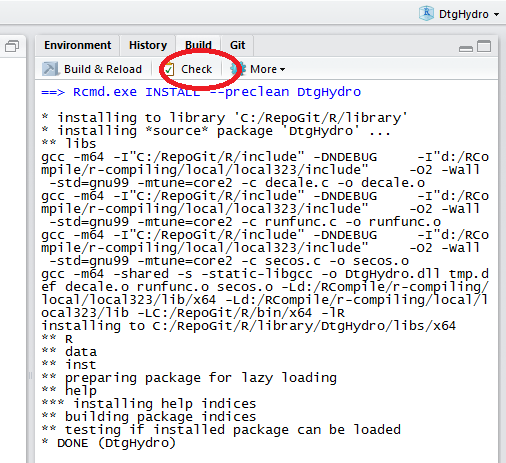
**2**

1. Effectuer un « Clean rebuild »



1. Cliquer sur « Check » et essayer de corriger les erreurs signalées



Note :

Certaines erreurs/notes/avertissements ne sont pas bloquantes :

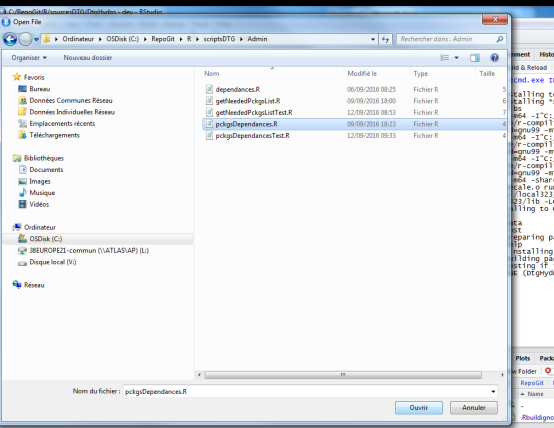
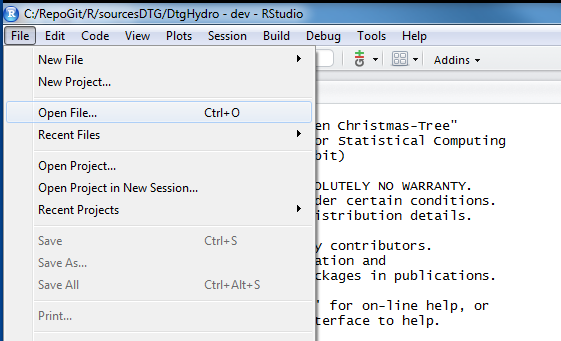
\* checking R files for non-ASCII characters ... WARNING: généralement accents dans la doc, on laisse passer

\* checking PDF version of manual ... WARNING / ERROR : la doc pdf n’a pas été générée : aucune importance

(liste à enrichir)

1. Exécuter le script « dépendance » pour vérifier quels sont les scripts/packages impactés
   1. Ouvrir le script « pckgsDependances.R » dans C:\RepoGit\R\scriptsDTG\Admin

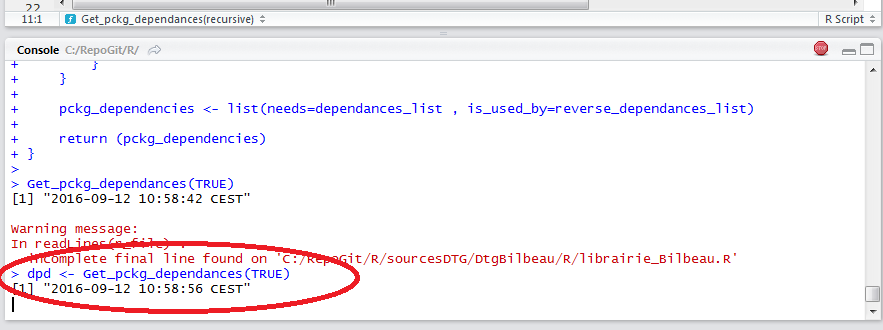
**2**



**1**

Note : Compter environ 10 minutes d’exécution

* 1. Exécuter la fonction Get\_pckg\_dependances(TRUE)

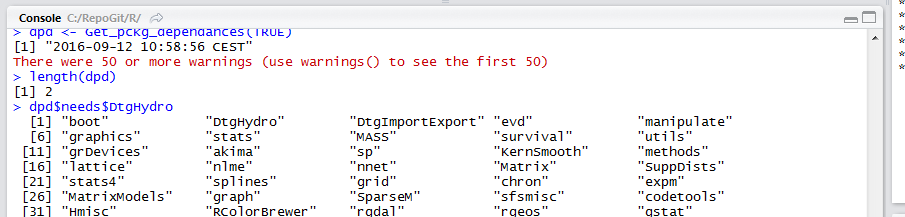


* 1. Rechercher dans le résultat du script le projet à déployer

Note : Si c’est un package, rechercher via son nom

Exemple : dpd$needs$DtgExemple

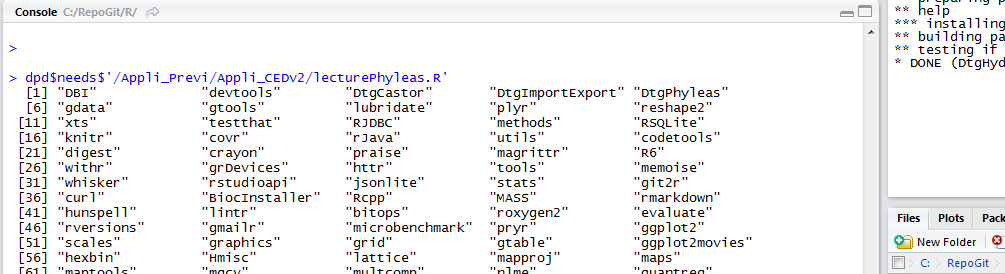
Ou dpd$ used\_by$DtgExemple

**

Si c’est un script, rechercher via son nom précédé de son chemin depuis scriptsDTG

Exemple : dpd$needs$‘/CHA/Calcul Débit SVR.R’

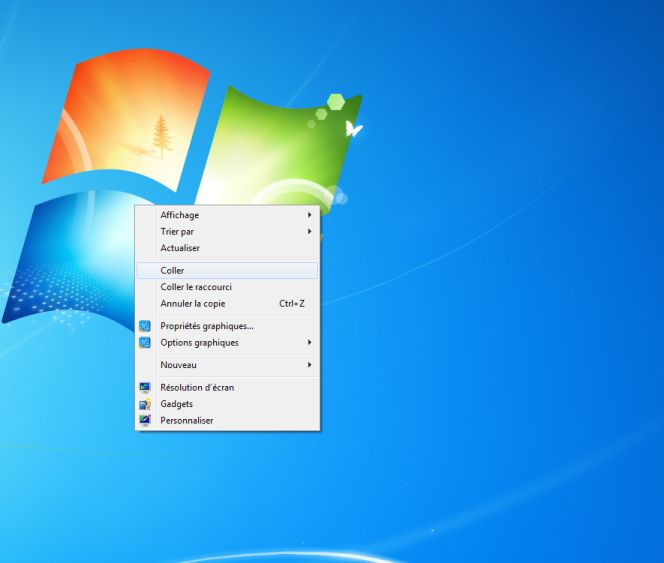
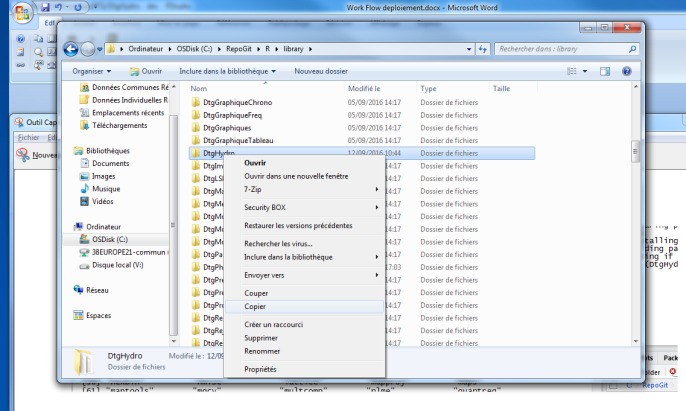
Note : l’emploi du ‘ pour entourer le nom est indispensable.



* 1. Vérifier que les packages et scripts ayant des dépendances avec le projet à déployer fonctionnent encore

1. Copier le répertoire repoGit/R/**library**/nomDuPackage dans un répertoire local

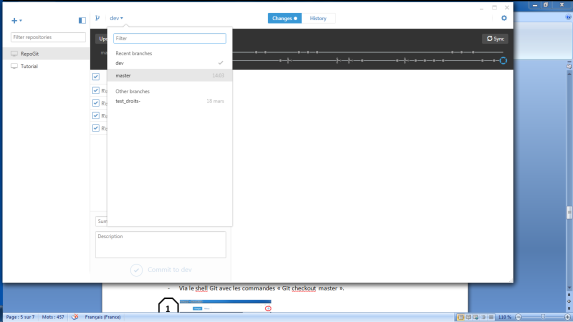
**2**



**1**

1. Basculer sur la branche master

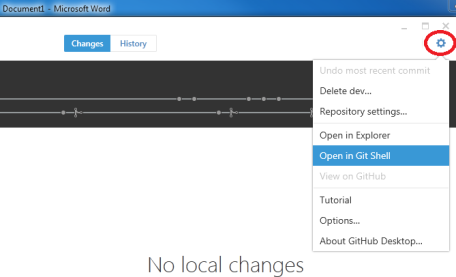
* Via l’interface GitHub en cliquant sur la branche Master.



**Ou**

* Via le shell Git avec les commandes « Git checkout master ».

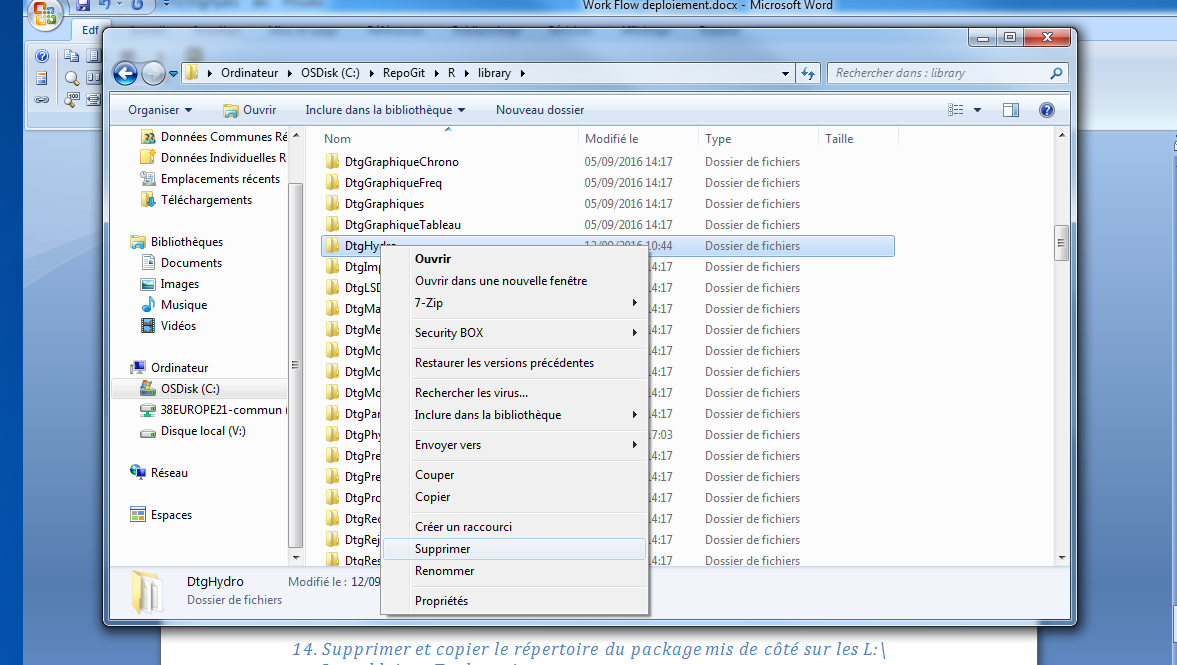
**1**

 Capture.PNG

**2**

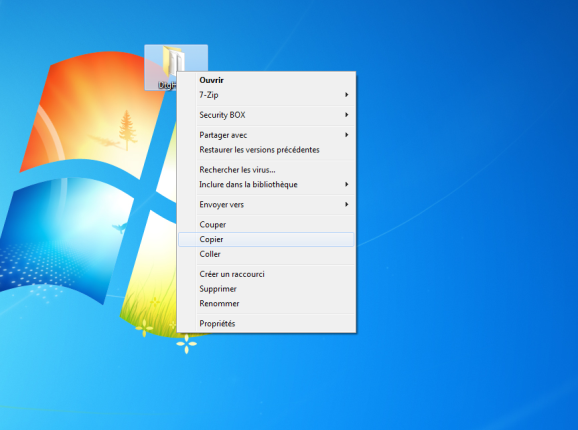
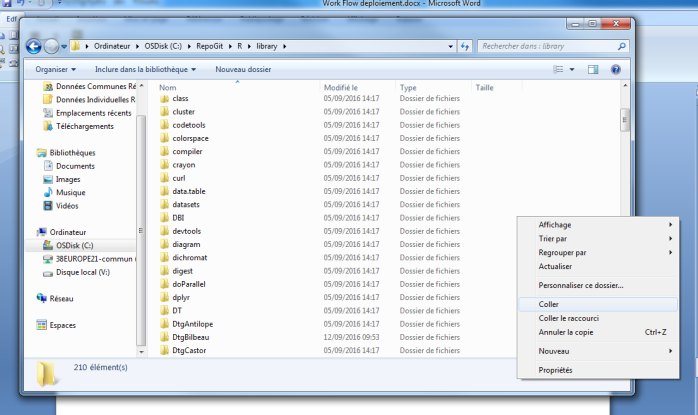
1. Supprimer le répertoire repoGit/R/**library**/nomDuPackage et coller celui mis de côté précédemment (ne pas remplacer directement, bien supprimer avant)

**1**

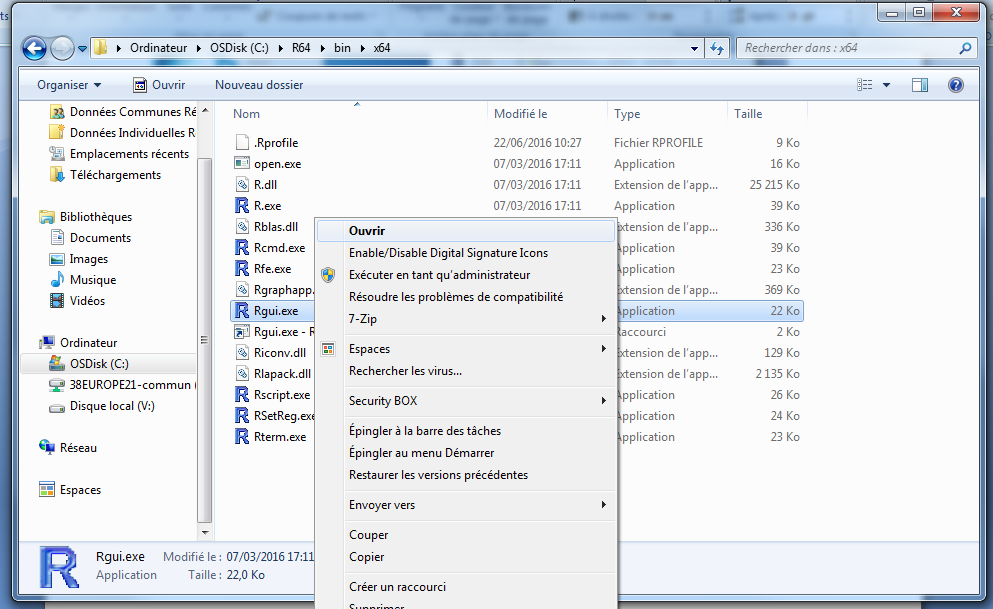


**3**

**2**

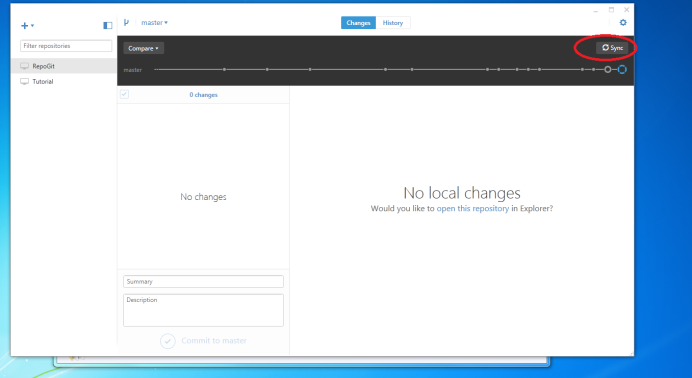
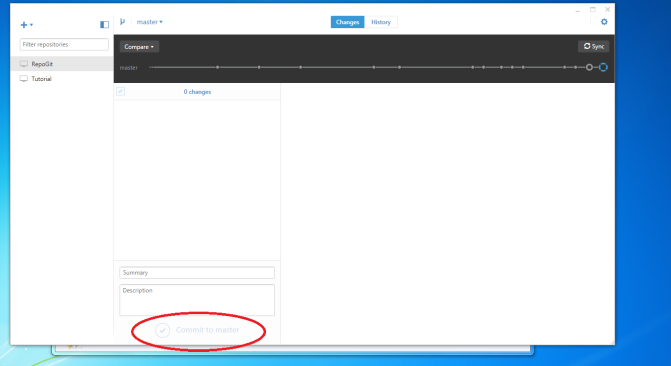
 

1. Ouvrir RGui et importer le package, tester une fonction (ou lancer les tests autos si présents)
   1. Ouvrir RGui dans C:\R64\bin\x64



1. Faire un commit sur master et push

* Via l’interface GitHub en cliquant sur commit to master après avoir bien renseigné les fichiers et la nature des modifications.



**Ou**

* Via le shell Git avec les commandes «Git pull ; Git push origin/master ».

1. Supprimer et copier le répertoire du package mis de côté sur les L:\ Grenoblois et Toulousains
2. Ouvrir RGui depuis chaque L:\ et importer le package, tester une fonction (ou lancer les tests autos si présents)