

Capsule 1 – Concepts de base de GitHub et du versionnage

Hubert Cadieux et Arnaud Beaulé

Introduction (50 sec)

Bonjour tout le monde!

Vous êtes nombreux à découvrir la programmation et les outils de gestion de code dans le cadre de vos études et de vos recherches en sciences sociales.

Vous utilisez ces outils pour faire des analyses de données, pour écrire des articles, pour créer des rapports de recherche, ou encore pour gérer vos bases de données.

Mais pour pouvoir collaborer efficacement, gérer les différentes versions de vos fichiers, et assurer la reproductibilité de vos travaux, il faut des outils spécialisés.

Dans cette capsule, nous allons découvrir Git et GitHub, deux outils de *contrôle de version* qui vous permettront de mieux organiser et partager vos projets de recherche.

Définition de Git et GitHub (2 min)

Premièrement, qu'est-ce qu'un système de contrôle de version ?

Imaginez que vous travaillez sur votre mémoire de maîtrise. Vous travaillez dessus, vous l'envoyez à votre directrice pour de la rétroaction, elle vous renvoie ses commentaires, vous les appliquez et ainsi de suite.

Au bout d'un moment, vous vous retrouvez avec des fichiers qui s'appellent `memoire_final.docx`, `memoire_final_vraiment_final.docx`, `memoire_final_pour_vrai_cette_fois.docx`. Ça vous dit quelque chose ?

Un système de contrôle de version, c'est exactement l'outil qui résout ce problème. Il garde automatiquement un historique de toutes vos modifications, vous permet de revenir à n'importe

quelle version précédente, et surtout, il vous permet de voir exactement ce qui a changé entre deux versions. C'est exactement à ça que Git et GitHub servent.

Maintenant, la différence entre Git et GitHub. C'est important de bien comprendre ça.

Git, c'est le logiciel que vous installez sur votre ordinateur. C'est lui qui fait tout le travail de gestion des versions, qui suit vos modifications, qui crée l'historique. Git fonctionne entièrement en local sur votre machine.

GitHub, c'est une plateforme web qui utilise Git. C'est comme un Google Drive, mais spécialisé pour les projets de programmation et de recherche. GitHub vous permet de sauvegarder vos projets en ligne, de les partager avec d'autres personnes et de collaborer facilement, même à distance.

Ce qui rend GitHub particulièrement intéressant, c'est que son interface est conçue pour être accessible aux débutants. Sans GitHub, utiliser Git directement nécessiterait beaucoup plus de connaissances techniques et l'utilisation constante de la ligne de commande. GitHub rend donc l'utilisation de Git beaucoup plus facile et visuelle.

Si ça semble abstrait, ne vous inquiétez pas. Plus tard dans ces capsules, nous allons vous montrer concrètement ces différences pour que vous compreniez bien comment tout ça fonctionne en pratique.

Pourquoi utiliser Git/GitHub en sciences sociales ? (1 min 20 sec)

Maintenant, vous vous demandez probablement : "Pourquoi est-ce que moi, étudiant en sciences sociales, j'aurais besoin de ça ?" C'est une excellente question !

Premièrement, la traçabilité des modifications. Quand vous travaillez sur une analyse de données ou un projet de recherche, vous voulez pouvoir documenter exactement ce que vous avez fait, quand vous l'avez fait, et pourquoi. Git garde une trace complète de tous vos changements avec vos commentaires. Plus jamais de "je ne me rappelle plus pourquoi j'ai supprimé cette partie de l'analyse".

Deuxièmement, la collaboration et le partage. En sciences sociales, vous travaillez souvent en équipe - avec votre directeur ou directrice de recherche, des collègues étudiants ou dans des projets interdisciplinaires. GitHub vous permet de travailler ensemble sur les mêmes fichiers sans vous marcher sur les pieds, de partager facilement vos données et vos codes d'analyse.

Finalement, la reproductibilité. C'est un enjeu majeur en recherche aujourd'hui. Avec Git et GitHub, quelqu'un d'autre peut reprendre exactement votre travail, comprendre vos étapes, et reproduire vos résultats. C'est essentiel pour la crédibilité scientifique.

Vocabulaire de base (1 min 30)

- Repository (dépot)
 - Commit (validation/livraison)
 - Branch (branche)
 - Merge (fusion)
 - Pull request (demande de fusion)
-

Fonctionnement général du versionnage (1 min 30)

- Historique des modifications
 - Revenir à une version antérieure
 - Suivi des contributions individuelles
-

Installation de Git (2 min)

- Installation sur Windows, Mac et Linux
 - Introduction à GitHub Desktop (GUI) et au terminal (CLI)
 - Avantages et limites de chaque approche
-

Présentation de l'interface GitHub (2 min)

- Création de compte
- Navigation dans un dépôt (README, fichiers, historique)