

Git & Versionierung

Christoph Rinne*

17. Dezember 2021

Inhaltsverzeichnis

1	Grundlegende Information	1
2	Git nutzen	1
2.1	Szenarien zur Nutzung	1
2.2	GIT-Plattform und GIT-Account	2
2.3	Git lokal einrichten	2

1 Grundlegende Information

Git ist eine freie Software für das verteilte Arbeiten an Dateien. Sie gewährleistet die Synchronisation und Kontrolle aller Änderungen oder kurz gesagt die Versionierung aller Dateien. Bei textbasierten Dateien, z.B. Markdown und csv-Tabellen, erfolgt dies auf Zeilenniveau. Die Kooperation an Projekten mit Texten, Daten, Abbildungen und Programmcode, z.B. R, ist damit ohne den sonst üblichen Aufwand und oft auftretende Verwirrung möglich.

Die Synchronisation erfolgt zu einem Server und einem dort betriebenen Dienst, hier sind zwei Webanwendungen sehr bekannt: [GitLab](#) und [GitHub](#). Die lokale Kontrolle wird ebenfalls durch Git gesteuert, dafür gibt es diverse grafische Oberflächen (GUI), z.B. [gitextensions](#). Auch [R-Studio](#) bietet eine vollständige *workflow* für die alltägliche Arbeit mit Git. Für das Schreiben auf dem Server ist eine Authentifizierung notwendig die meist über einen ssh-key erfolgt.

Bei der asymmetrischen Verschlüsselung ist ssh ein weit verbreiteter Standard und in allen gängigen Betriebssystemen integriert. Grundlegende Informationen zu dieser Verschlüsselung gibt es u.a. beim [BSI: Arten der Verschlüsselung](#). Eine Einführung zu ssh und den Befehlen finden Sie u.a. bei GitLab: [GitLab and SSH keys](#).

2 Git nutzen

2.1 Szenarien zur Nutzung

Das Naheliegendste ist es, GIT für das Entwickeln von Programmen zu nutzen. In einem archäologischen Kontext sind aber auch andere Szenarien und Kombinationen daneben denkbar.

2.1.1 Daten sammeln

Das Sammeln von Daten erfolgt lokal, auf dem einem PC, gerne in [MS Excel](#) alternativ in [LO Calc](#) oder in echten Datenbankmanagementsystemen wie MS Access oder SQLite. Wollen Sie diese Daten aber gemeinschaftlich erheben oder nutzen kommen Sie um einen Datenaustausch nicht umhin.

*Christian-Albrechts Universität zu Kiel, crinne@ufg.uni-kiel.de

Eine zentrale Datenhaltung kann innerhalb eines lokalen Netzwerkes mit Freigaben und in MS Access mit verknüpften Datenbanken (backend) oder in SQL mit angehängten Datenbeständen (attached) arbeiten. Das ist nicht wirklich kompliziert, setzt aber für eine performante Nutzung eine gute Datenstruktur voraus. Sobald Sie das lokale Netzwerk verlassen sind Sie auf eine Serverdatenbank (z.B. PostgreSQL) angewiesen und haben einen sehr hohen administrativen Aufwand auch bei Sicherheitsaspekten. Für Projekte einer überschaubaren Größe also keine Alternative.

Alternativ: sammeln Sie Daten als CSV-Tabellen und synchronisieren (pushen) Sie regelmäßig zum Repository. Wie Sie diese CSV-Tabelle lokal editieren ist fast egal, möglich sind MS Excel, LO Calc oder ein Texteditor. **Allerdings** ist der Wechsel zwischen MS Excel und LO Calc sehr umständlich und fehleranfällig, da MS Excel die Spalten mit einem “;” und LO Calc diese mit einem “,” trennt. Optional können Sie auch ein anderes Trennzeichen verwenden, z.B. . Allerdings werden diese Dateien mit der Endung .txt oder .csv nicht mehr automatisch richtig erkannt und beim Speichern muss in MS Excel jedes mal das Trennzeichen definiert werden während im LO Calc automatisch ein ‘,’ gesetzt wird. Der **Vorteil** ist die Versionierung durch GIT, also das Protokoll der vollzogenen Änderungen, und die zentrale Ablage auf einem Server um dessen Sicherheit und Zugangsberechtigungen Sie sich nicht unmittelbar kümmern müssen.

2.1.2 Texte Verfassen

Sofern Sie sich an die Arbeit mit Rstudio und das Schreiben von Markdown gewöhnt haben, können Sie über GIT natürlich auch gemeinsam Texte, Artikel und ganze Bücher mit *Bookdown* schreiben. Sie sollten innerhalb von RStudio nicht mit einzelnen Dateien sondern mit Projekten und Versionskontrolle arbeiten. In diesem Projekt können Sie dann die benötigten Daten (CSV) als auch die Literatur als BibTeX-Datei pflegen und in Ihrem Text einbinden. Tabellen, Analysen und resultierende Grafiken werden mittels R-Code auf Grundlage der vorliegenden Daten im Text generiert und sind stets aktuell. Natürlich ist dies eine grundlegend andere Arbeitsweise als eine gemeinsame Ablage in der Cloud und die Versionierung von Hand, selbst bei der Verwendung von Online Office. Es ist eine Umstellung mit anfänglichen Reibungsverlusten und benötigt Einarbeitungszeit.

Anmerkung: Wenn Sie dieses Thema *Online Texteditieren* interessiert, sehen Sie sich auch mal folgendes an [Fidus Writer](#).

2.2 GIT-Plattform und GIT-Account

2.3 Git lokal einrichten

Der Vorgang ist hier für Windows 10 beschrieben.

2.3.1 SSH

Sie können OpenSSH auf der Kommandozeile verwenden, es gehört zu den optionalen Programmen (optional features) oder PuTTY mit einer grafischen Oberfläche nutzen. Für beide Wege gibt es ergänzende Anleitungen im Netz (u.a. [hier](#)).

Laden Sie das Programmpaket (*.msi) von PuTTY herunter <https://www.chiark.greenend.org.uk> und installieren Sie es. Starten Sie dann das Programm **puttygen**. Die Vorgabe ist ein RSA-Schlüssel, belassen Sie dies und wählen Sie [Generate]. Danach müssen Sie die Maus über dem leeren Fensterausschnitt bewegen um einen zufälligen Schlüssel zu erzeugen. Im folgenden Fenster ergänzen Sie bitte das Passwort (Key passphrase) und prägen Sie es sich gut ein. Anschließend müssen Sie nacheinander mit [Save public key] und [Save private key] beide Schlüssel speichern. Der Standardorder dafür in Windows ist c:\Benutzer\<nutzernamen>\.ssh . Nennen Sie beide identisch, der öffentliche Teil hat die Endung .ppk der private Schlüssel hat keine.

2.3.2 GIT Extensions

Laden Sie sich die aktuelle Version (*.msi) von [Git Extensions](#) herunter.

Es werden weitere Programme benötigt die getrennt heruntergeladen und installiert werden müssen:

- [kdiff3](#) dient der wird nachfolgend für die Synchronisation (merge) verwendet. Installieren Sie mit den Vorgaben.

Installieren Sie nun Git Extensions und folgen Sie den Schritten der Installation mit den Vorgaben. Folgendes bedarf der Aufmerksamkeit:

- Select SSH Client: belassen Sie es auf der Vorgabe, aktuell OpenSSH
- Telemetry privacy policy: Im Prinzip handelt es sich um Geben und Nehmen. Sie nutzen die Software und geben Teile Ihres Nutzerverhaltens für die Optimierung derselben preis. Ihre Entscheidung.

Starten Sie Git Extensions. Vermutlich werden Sie zuerst einem Fenster zu den Einstellungen “Settings - Checkliste” konfrontiert, wenn nicht gehen Sie dorthin: Werkzeuge > Einstellungen.

- Benutzername und E-Mail-Adresse: Bitte entsprechendes eintragen
- Direkt darunter im selben Fenster fehlen weitere Angaben
- Merge-Werkzeuge: kdiff3
- Pfad zu Merge-Werkzeug: [Durchsuchen] und dort kdiff3 auswählen, die nächste Zeile zu den Befehlen sollte automatisch ergänzt werden.
- Wiederholen Sie den Vorgang für die Vergleichswerkzeuge.

Zeilenenden werden in Windows anders kodiert als in Linux/Mac. Aktivieren Sie deshalb die Option “Zeilennumbrüche im Windows-Format auschecken...”. Ansonsten würde jedes Zeilenende zu einer protokollierten Änderung.

2.3.3 RStudio & GIT