

Forschungsseminar – Social Data Science

Git Intro

Jonas Beste und Arne Bethmann



Professorship for Statistics & Methodology
HWS 2015

Was ist ein Repository?

- ▶ Eine Art Datenbank
- ▶ Protokolliert Veränderungen an Dateien
- ▶ Speichert neben Inhalten auch Zusatzinformationen: z. B. Wer und Wann
- ▶ Bezieht sich auf ein Verzeichnis in dem man Dateien für ein Projekt abgelegt hat und bearbeitet
- ▶ Gibt es *lokal* auf dem eigenen Rechner und *remote* z. B. bei GitHub

Repository erstmalig lokal anlegen

1. Klonen des Remote-Repositories um eine lokale Kopie anzulegen (`git clone`)
 - ▶ Dadurch wird sowohl eine Kopie der Datenbank (versteckt, Unterverzeichnis `.git`) angelegt,
 - ▶ als auch die Dateien und Verzeichnisse des Repositories lokal angelegt (Arbeitskopie aka working copy)
- ▶ Alternativ kann man auch ein ganz neues lokales Repository anlegen ohne auf ein Remote-Repository zurückzugreifen (`git init`)

Lokales Repository updaten

- ▶ Bevor nach einer Pause wieder mit der Arbeit begonnen wird sollten immer zwischenzeitliche Änderungen am Remote-Repository heruntergeladen werden
1. Lokales Repository auf den Stand des Remote-Repository bringen (`git pull`)

Der Workflow

Eigene Änderungen einpflegen

1. Eine oder mehrere Dateien bearbeiten
2. Einige oder alle der geänderten Dateien zum Einpflegen ins lokale Repository vormerken (aka staging, `git add`)
3. Einpflegen der Änderungen an den vorgemerkten Dateien in das lokale Repository (`git commit`)
 - ▶ Dazu muss ein Kommentar eingegeben werden, der möglichst knapp aber verständlich die Änderungen beschreibt
 - ▶ Es empfiehlt sich immer miteinander zusammenhängende Änderungen zu einem commit zusammenzufassen
4. Die Änderungen am lokalen Repository mit dem Remote-Repository abgleichen um die Änderungen für alle verfügbar zu machen (`git push`)

Zustand der Arbeitskopie anzeigen

1. Status der Änderungen an der Arbeitskopie ausgeben (`git status`)
 - ▶ Änderungen die im Staging-Bereich liegen (nach einem `git add`) und bei einem `git commit` committed würden
 - ▶ Änderungen an Dateien, die noch nicht im Staging-Bereich liegen (vor einem `git add`)
 - ▶ Dateien, die (noch) nicht von Git versioniert werden (was man per `git add` ändern kann)

Mit vermeintlichen Konflikten umgehen

1. Der Versuch Änderungen am lokalen Repository ins Remote-Repository zu übertragen (`git push`) scheitert, weil zwischenzeitlich Änderungen am Remote-Repository durchgeführt wurden
2. Änderungen am Remote-Repository lokal einpflegen und eigene Änderungen auf diesen Stand übertragen (`git pull --rebase`)

Mit echten Konflikten umgehen

1. Der Versuch ein rebase durchzuführen scheitert mit einem merge-Konflikt
2. Datei mit Konflikt bearbeiten um den Konflikt aufzulösen
3. Korrigierte Datei in den Staging-Bereich schieben (`git add`)
4. Das rebase fortsetzen (`git rebase --continue`)