Version Controlling mit Git + Github + RStudio

Claudia Neuendorf & Benjamin Becker

18. August 2021

Interner IQB Workshop

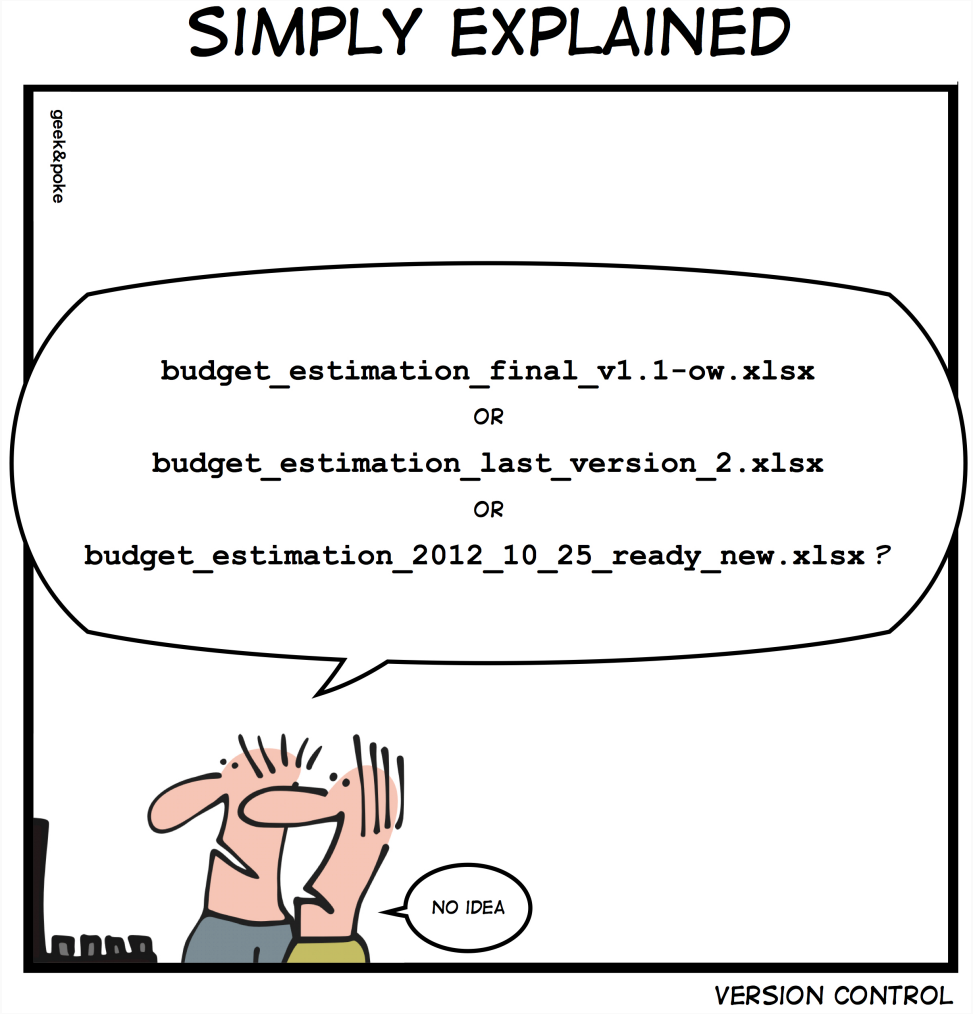
Disclaimer

Agenda

- Motivation
- Setup
- Work Flows
- Allgemeine Empfehlungen
- Ressourcen

Motivation

Motivation


SIMPLY EXPLAINED
geek&poke
budget_estimation_final_v1.1-ow.xlsx
OR
budget_estimation_last_version_2.xlsx
OR
budget_estimation_2012_10_25_ready_new.xlsx ?
NO IDEA
VERSION CONTROL

Motivation 1

Solo-Projekte

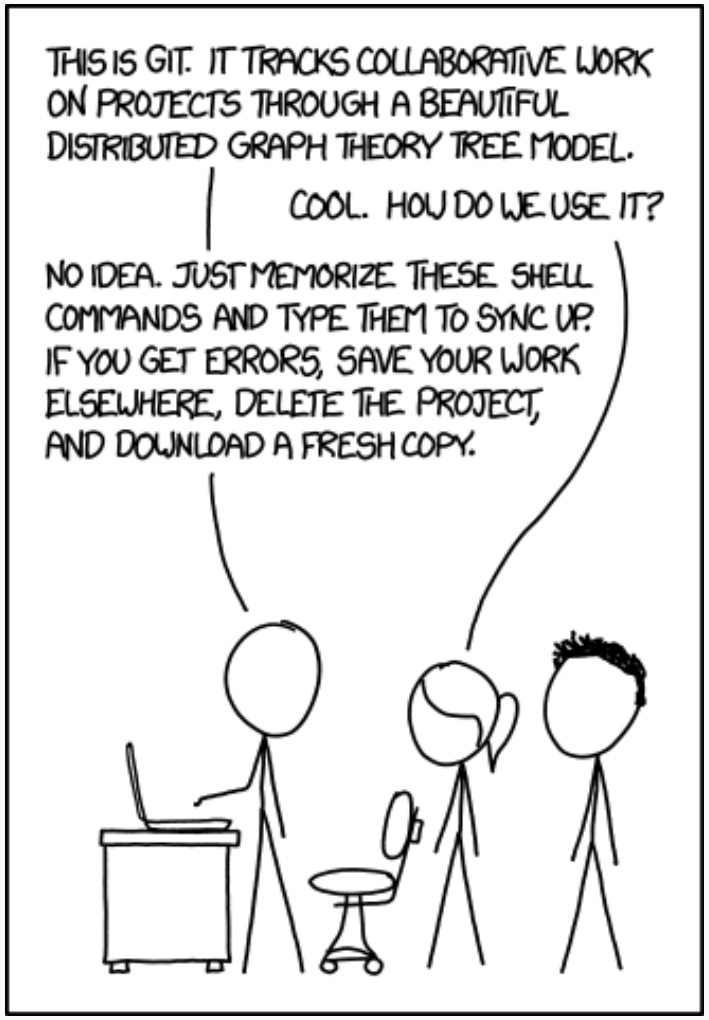
- Implementation einer 'Change History'
  - Was wurde geändert?
  - Wann wurde es geändert?
- Keine verrückten Dateinamen
- Kein Archiv-Ordner
- Zugänglichkeit für andere ('Open Science')
- Sicherung/Archivierung
- ...

Motivation 2

Kollaborationen

- Wer hat wann was geändert?
- Eindeutiger, aktueller Projektstand
- Keine nervigen Mail-Anhänge oder Austausch-Plattformen
- Erlaubt einfaches, paralleles Arbeiten
- Implementiert hierarchische Projekt-Strukturen (Absegnen von Änderungen)
- ...

Aber...

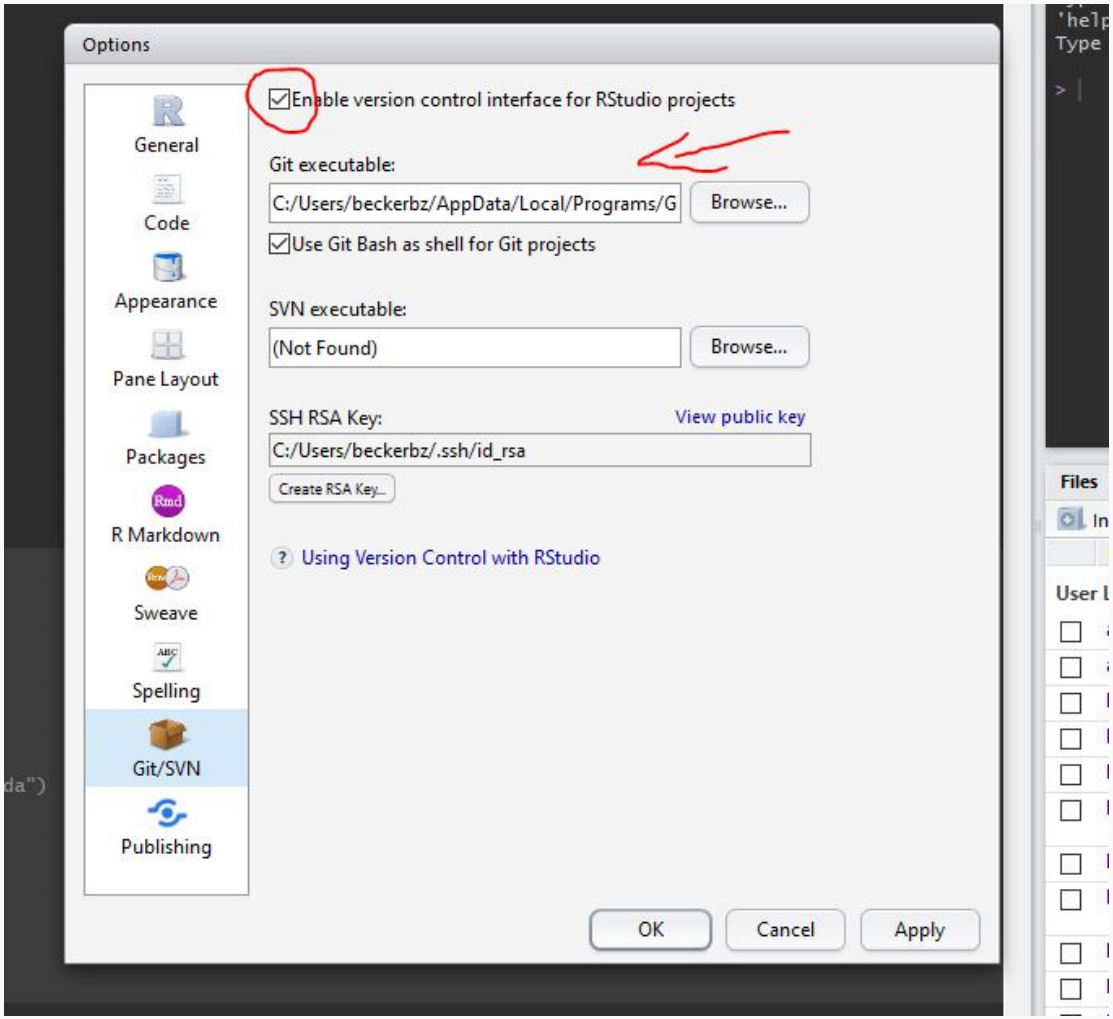

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

Setup

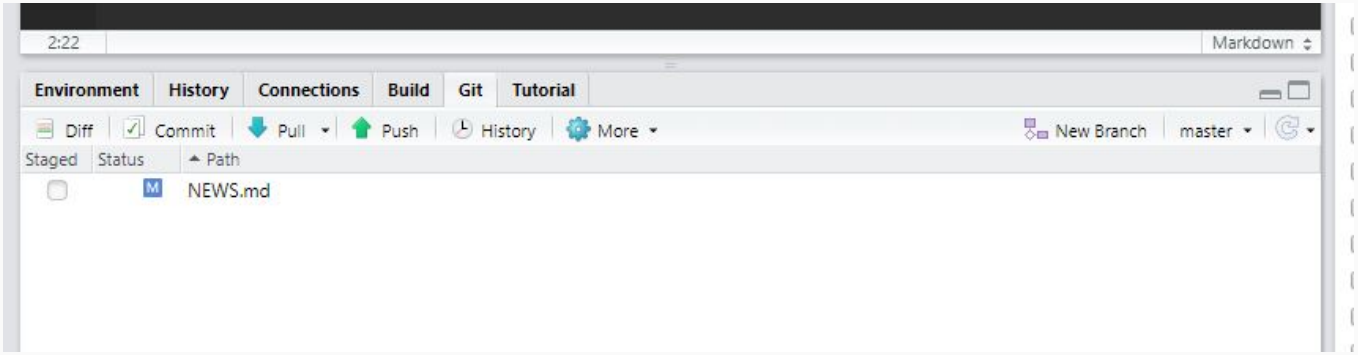
Voraussetzungen

- Git-Installation
- RStudio-Installation
  $\rightarrow$ Alternativen: Shell, Gitkraken, SmartGit, ...
- Account anlegen für Github
  $\rightarrow$ Alternativen: Bitbucket, Gitlab, ...
- Alles verbinden

RStudio


Options
General
Code
Appearance
Pane Layout
Packages
R Markdown
Sweave
Spelling
Git/SVN
Publishing
Enable version control interface for RStudio projects
Git executable:
C:/Users/beckerbz/AppData/Local/Programs/G
Browse...
Use Git Bash as shell for Git projects
SVN executable:
(Not Found)
Browse...
SSH RSA Key:
View public key
C:/Users/beckerbz/.ssh/id_rsa
Create RSA Key...
Using Version Control with RStudio
OK
Cancel
Apply
Files

RStudio


2:22
Markdown
Environment
History
Connections
Build
Git
Tutorial
Diff
Commit
Pull
Push
History
More
New Branch
master
Staged
Status
Path
M
NEWS.md

Work Flows

Work Flow 1

Erstellen eines Repositories

- Erstellen eines **Online-Repositorys** (z.B. auf Github)
  - Benutzen einer R-spezifischen .gitignore
  - Initialisieren mit einer kurzen Readme-Datei (.md)
- Repository lokal auf den PC **clonen** via neues Projekt
- R-Projekt wird automatisch in bestehendes Repository eingefügt

Exkurs: gitignore

- Einfache Text-Datei
- Welche Dateien sollen nicht getracked werden?
  $\rightarrow$ Diese dann nur lokal in ihrer aktuellen Version vorhanden!
- Optionen
  - Einzelne Dateien
  - Ordner
  - Bestimmte Datei-Typen
  - Kombinationen daraus
- Anwendungs-Beispiele
  - Große Dateien (Datensätze, Bilder, ...)
  - Hilfsdateien (z.B. bei LaTeX-Kompilierungen)

Work Flow 2

Arbeit an einem bestehenden Repositorium

- Davor: Lokales Repo synchronisieren (**pullen**) oder **clonen**
- Änderungen im lokalen Repository
  $\rightarrow$ Dateien erstellen/überschreiben/löschen
- Änderungen **stagen**
- Gestagede Änderungen **commiten** (aka neue Version)
- Einen oder mehrere Commits **pushen** (Online-Repository updaten)

Exkurs: mergen

Konflikte innerhalb von Textdateien

- Vor allem bei kollaborativem Arbeiten
- Eigene Änderungen stehen im Konflikt mit anderen Änderungen $\rightarrow$ Git beschwert sich und zeigt Konflikte auf
- Auswahl der gewünschten Änderungen
- Neu stagen, commiten und pushen

Exkurs: Branches

Mehrere parallele Projektversionen innerhalb eines Repos

- Vor allem im Bereich Software-Entwicklung
- z.B. ein stabiler & ein Development-Branch
- Nur bestimmte Änderungen in stabilen Branch übernehmen
- Achtung: RStudio UI stößt hier an seine Grenzen

Allgemeine Empfehlungen

Eure Eindrücke?

Allgemeine Empfehlungen

- So simpel wie möglich halten!
  - Wenn möglich keine Branches/Forks/Pull Requests
- Inhaltlich bedeutungsvolle Commits
- Repository schlank halten (keine großen Dateien)
- Vermeidet direktes Arbeiten über die Github Homepage

Resources

Git + RStudio Resourcen

- Small Intro (https://r-bio.github.io/intro-git-rstudio/)
- Happy Git with R (https://happygitwithr.com/)
- R Packages and Git (https://r-pkgs.org/git.html)

Allgemeine Git Resourcen

- Git Book (http://git-scm.com/book/en/v2)

Vielen Dank fuer Eure Aufmerksamkeit!

Vielen Dank fuer Eure Aufmerksamkeit!

Fragen? Anmerkungen?