

# Stochastic Methods + Lab

Session 1  
Sep. 2, 2019

Prof. Sören Petrat

Office: 112, Research I

## Organization:

- syllabus
- website
- class: Mon., 14:15-15:30 and 15:45-17:00  
Th., 14:15-15:30  
no practical difference between lab and lecture slots
- weekly homework assignments / programming (starting Mon., Sep. 9)
  - ↳ download and upload and grading via git (see later)
  - ↳ due a week later before class
  - ↳ only best 8 out of 11 assignments count for total grade  
=> no late hand-ins, no excuses (except illness longer than a week)
  - ↳ solutions discussed in class
  - ↳ note: I check for copying, respect Academic Integrity
- grade: 80% HW  
20% final take-home exam  
(same grade for class and lab)

- TA: Tuba Masur

↳ will grade the assignments

↳ weekly office hour to ask general questions

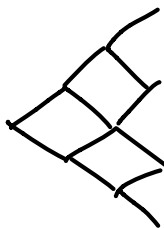
↳ ask her questions about grading first

- topics:

• introduction to git and scientific python

• basics of finance (interest, cash flows, bond, immunization, options)

• binomial tree models



• Brownian motion



• stochastic integrals and ODEs

• Black-Scholes eq.

• time series analysis

• some extra topics not related to financial math (depending on time)

- books: • Cyvv (main reference)

• Etheridge (but later may more mathematically involved than this class)

Bring laptop to class

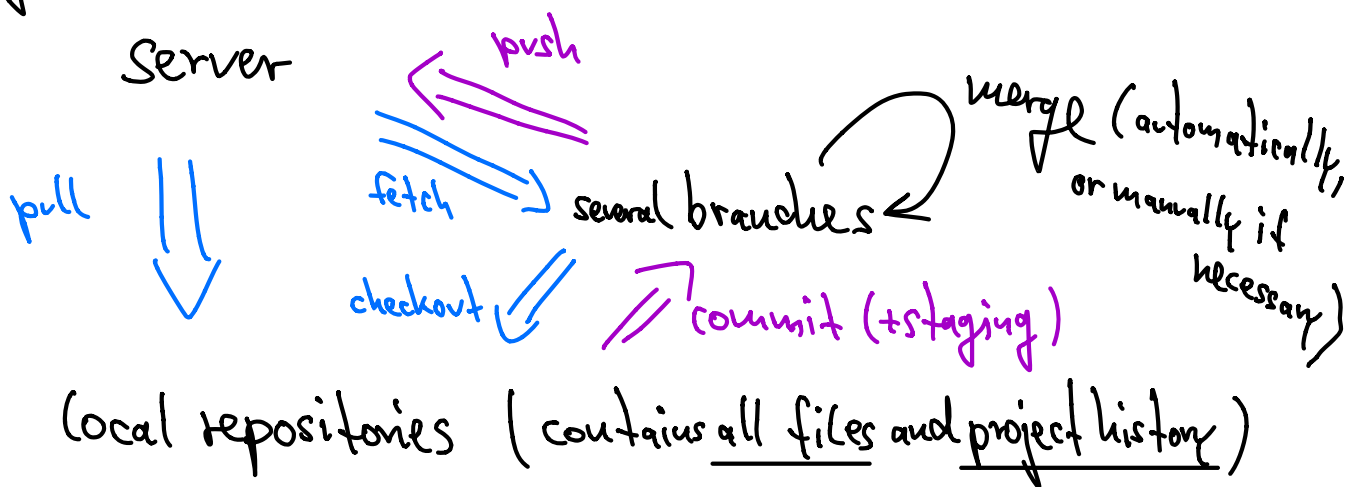
# 0. Introduction to git and scientific python

## 0.1 git

- software (free + open source)
- project development software
  - ↳ version control, change tracking
  - ↳ speed, non-linear workflow (file merging etc.)
  - ↳ used for software development (Linux, recently Windows, some google, ...)
  - ↳ useful for (large) scientific collaborations

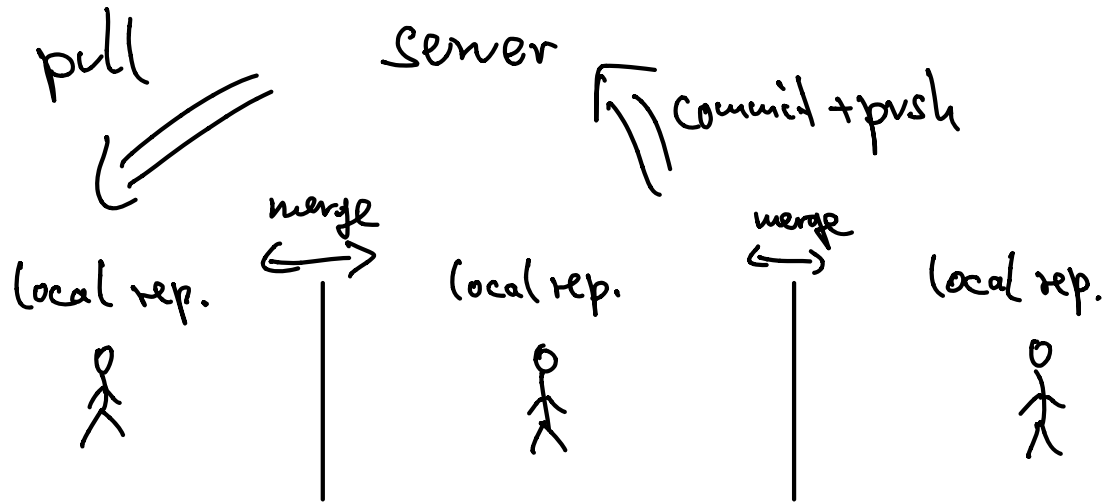
hosting server (file storage etc.): bitbucket

rough overview of workflow:



Ex.: • scientific collaboration

smaller projects: one branch usually sufficient



• this class

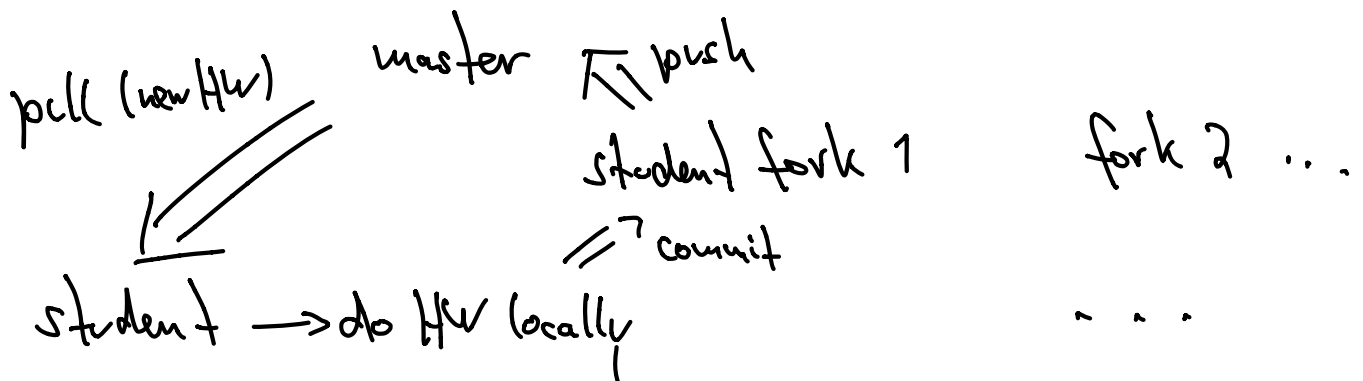
master branch: all official course material (HW)

each student: • separate private fork of project

⇒ get all files from master, but add own private work

instructor/TA: write access to all project forks

student: read access to master



student: pull master → do work → stage and commit → push

To Do:

- set up git using "Intro to git for academics" (link on website)

↳ download and install git

↳ configure git

↳ bitbucket account

sign up with Jacobs email address → academic account  
→ so I can find you

↳ fork and clone repository

↳ own project branch

↳ to your local machine

give spetrat and goekcetubamasur write access