# Data Science tools (1.22)

* **Repository**- the collection of files of all versions of all files
  + Local: the repository on your machine
  + Remote: the repository on a server somewhere else (github.com)
* **Sandbox**- label the changes in a file for inclusion in the repository
  + Allows you. To keep some files or changes only in the sandbox, while other are going to be added to the repo.
* **Commit**- add all staged changes to the local repo
  + This is where you add message explaining why these changes were made
* **Push**- adds repo to your GitHub where calibrators can see it
* **Clone**- create a local copy of the central repo
  + The first time you create a copy it’s called a clone. Any further copies/changes are called pull
* **Pull**- add changes from the central repo the local repo
* **Branch**- a copy of the code where you develop new features without interfering with the code that has already been developed.
  + Useful when working with multiple calibrators
  + Main branch- tested/released code
  + Develop branch- newly developed code
* **Pull request**- submit a branch for review
  + Receive feedback from the maintainers
* **Merge**- include the changes from one branch into another
* Git is not meant for large data files (<100 MB)

Options for large files:

* **Git large file storge** (GLFS)
  + < 2 GB
* **Git Annex**
  + Complicated to setup
  + Hard to maintain
* **Datalad**
  + Uses git-annex internally.
  + Slightly easier to maintain
  + OpenNeuro
* **DataJoint**- explicit data analysis pipelines
  + Good to save your preprocessing script so you can run on it any raw file
  + Everything is inside a database and your analysis adds extra lines
* Setup a structured folder for all your data
  + Data/project1
  + Data/project2

**MATLAB**

* Centralized biannual releases associated add-ons

Projects:

* Startup
  + Check for files, start parallel workers, connect to a vpn, write something to a log?
  + Set the search path
  + Shutdown
  + Upgrade
  + Export
* Alternative: startup.m
* Container
  + Virtual computer with all software needed for a data analysis project
  + Mainly open-source software
  + Docker (workstation running linux, windows, macOS)
  + singularityCE (HPC)
* Docker Examples
  + Docker pull hello-world
  + Dockerfile
    - Instructions to build an image (=set up the computer)
    - Written by maintainer/developer
  + Image
    - Template for a container
    - Read-only
    - Complete installation
  + Container
    - Docker “boots” the image to create a. running instance on your workstation
    - Interact with it via browser, terminal, GUI
    - Read-only, all data are lost once the container shut down
    - When to use it?
    - multiple interacting software packages (e.g., Matlab + python)
    - multiple machines, users
    - read only
    - share data?
    - disadvantage: more work to set up but you only set it up once
  + Volumes & bind mounts
    - External disks for data persistence
  + When to use environments?
    - Only MATLAB or only python
    - Easy to setup
      * Hidden dependencies
      * Read/write
  + **Dumb parallelism**: running the same non-interactive code several time (25 experiments, 5000 neurons)