

# Data Visualization with R Git tutorial

Ariane Ducellier

University of Washington - Fall 2025

# Creating a new repository

Create a new repository on GitHub.

Go to Settings > Collaborators to add new people to the repository.

Accept the invitation that you have received by email.

On your terminal, clone the repository.

```
git clone "https://github.com/MyName/example.git"
```

# Modifying the content of the repository

Check the changes in the repository.

```
git status
```

Track a new or a modified file.

```
git add README.md
```

Create a new commit with all the tracked changes.

```
git commit -m "My new commit"
```

Push the changes to the remote repository.

```
git push
```

# Creating a conflict

What happens if you try to push your local changes on the remote while the remote contains modifications that you do not have locally?

```
git push
```

```
To https://github.com/ArianeDucellier/example.git
! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/ArianeDucellier/example.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

# Solving the conflict

Fetch the latest changes from the remote

```
git pull
```

```
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 985 bytes | 89.00 KiB/s, done.
From https://github.com/ArianeDucellier/example
 201c2aa..e307922  main      -> origin/main
hint: You have divergent branches and need to specify how to reconcile them.
hint: You can do so by running one of the following commands sometime before
hint: your next pull:
hint:
hint:   git config pull.rebase false  # merge
hint:   git config pull.rebase true   # rebase
hint:   git config pull.ff only       # fast-forward only
hint:
hint: You can replace "git config" with "git config --global" to set a default
hint: preference for all repositories. You can also pass --rebase, --no-rebase,
hint: or --ff-only on the command line to override the configured default per
hint: invocation.
fatal: Need to specify how to reconcile divergent branches.
```

# Solving the conflict

Fetch the latest changes from the remote

```
git config pull.rebase false  
git pull
```

```
Auto-merging README.md  
CONFLICT (content): Merge conflict in README.md  
Automatic merge failed; fix conflicts and then commit the result.
```

# Solving the conflict

Solve the conflict by merging the two versions of the file.

```
git status
```

```
On branch main
Your branch and 'origin/main' have diverged,
and have 1 and 1 different commits each, respectively.
  (use "git pull" to merge the remote branch into yours)

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
```

Now we can get back to git add/commit/push.

# Additional resources on Git

Software carpentry tutorial:

<https://swcarpentry.github.io/git-novice/>