Version Control Training

Ghislain Nono Gueye, Ph.D.

The Project

Develop a suite of functions (i.e. a package) that will help professors in their class-management tasks (e.g. computing grades, visualizing class performance, ...).

Guidelines

- The **activities** below will make you produce new code or alter already exisiting code. Commit every step in the process using thoughtful commit messages.
- If you come up with ideas to improve the code, commit these changes too.
- Push your local repo to Github after each activity.

Activity 1 (learning goals: local repository, remote repository, commit, push, diff)

- Create a function which assigns a letter grade based on a numeric grade (`assign_letter_grade()`).
  - the first version of the function should use `if()`
  - the second version of the function should use `cut()`
  - the final version of the function should use `dplyr::case_when()`
- Push your local repository to Github.
- Visualize the **diffs** between your commits.

Activity 2 (same learning goals as activity 1)

- Create a function to impute `NA` values.
  - The first version of the function should be on a single vector such that it has to be used with several `mutate()` functions.
  - The second version of the function should used a scoped version of `mutate()` (i.e. `mutate_*()`).

Activity 3 (learning goals: pull request, branch, merge)

- Create a function which computes weight averages (`compute_weighted_average()`).
  - The first version of the function should use a pure base `R` approach (with (a1+a2+a3)/3).
  - A collaborator on the project must send a **pull request** and improve the code by using the `mean()` function instead.
  - The second version of the function should use the `tidyverse` approach.

Activity 4 ()

- Create a function to drop the lowest score in a group of evaluations (`drop_lowest()`).
  - Version 1:
  - Version 2: