Git Commands

Contents

[Running Git 1](#_Toc129858020)

[Git Commands 2](#_Toc129858021)

[Username and email address 2](#_Toc129858022)

[Displaying username and email address 2](#_Toc129858023)

[Configuring username and email address 2](#_Toc129858024)

[Syncing Repositories 2](#_Toc129858025)

[Cloning 2](#_Toc129858026)

[Retrieving Data From Remote Branch 2](#_Toc129858027)

[Pushing Local Changes to Remote Branch 3](#_Toc129858028)

[Branch Management 3](#_Toc129858029)

[Creating and switching to branches 3](#_Toc129858030)

[Merging Branches 3](#_Toc129858031)

[Conflict Resolution 3](#_Toc129858032)

[Remote Branches 4](#_Toc129858033)

[Deleting a Branch 4](#_Toc129858034)

[Undoing Changes 4](#_Toc129858035)

[Undo public changes 4](#_Toc129858036)

[Fix the last commit message 4](#_Toc129858037)

[Undo local changes that have been saved but not committed 4](#_Toc129858038)

[Undo local changes that have been committed but not pushed 4](#_Toc129858039)

[Undoing a reversion that went wrong 4](#_Toc129858040)

[How to sync your latest local changes to the main branch 4](#_Toc129858041)

# Running Git

**Visual Studio Code**

* Switch to Terminal
* Open Git Bash; PowerShell won't work
* Start typing Git commands

**RStudio:**

* Switch to Terminal
* Start typing Git commands

**Command Prompt**:

* Use *cd* to set the local repository as your working folder
* Start typing Git commands

# Git Commands

## Username and email address

### Displaying username and email address

git config user.name

git config user.email

### Configuring username and email address

git config --global user.name "Payman Alemi"

git config --global user.email "Payman.Alemi@waterboards.ca.gov"

## Syncing Repositories

**Name of remote repository:** TutorialsBySamoora

**Branch I want to fetch**: main; *the branch names ARE case-sensitive; main and Main are not the same*

**Local repository location:** C:\Users\palemi\Documents\Github\TutorialsBySamoora

**In R:** Open the R project tied to the local repository location and switch to Terminal; if you don't have an R project for this; create one.

**In VSCode:** Open the local repository location and switch to Terminal and use Git Bash.

**In Command Prompt:** change to the local repository location by typing:

*cd: C:\Users\palemi\Documents\Github\TutorialsBySamoora*

### Cloning

#Syntax#

git clone --branch <branchname> <remote-repo-url>

#our remote repository URL is <https://github.com/CAWaterBoardDataCenter/DWRAT_DataScraping.git>

#Example#

git clone –main <https://github.com/CAWaterBoardDataCenter/DWRAT_DataScraping.git>

### Pulling Data From Remote Branch

**Method 1:**

git fetch origin main *#fetches data from remote branch*

git merge origin/main main *#merges the remote repository (origin/main) with the local repository (main)*

**Method 2:**

git pull origin main *#fetches and merges changes from the remote repository main into the local repository main in one action*

### Pushing Local Changes to Remote Branch

git add \* *(or)* git add . *#Both options grab all the files in the current folder*

git commit -m "Enter your comments about the code change"

git push origin main *#pushes changes to the remote branch main hosted on Github*

### Variations of Git Add

You consider the relative file path beginning at the root folder of Git. In the example, the root folder is C:\users\palemi\Github\Weather\_and\_Climate\_R while the file is contained here: "C:\Users\palemi\Documents\Github\Weather\_and\_Climate\_R\code\scripts\CIMIS\_Scraper.R".

Hence the relative file path is code\scripts\CIMIS\_Scraper.R

**Syntax:** git add [file path]

**Example:** git add code/Scripts/CIMIS\_Scraper.R

## Branch Management

### Detecting the current branch

## git branch #tells you the active branch

### Creating and switching to branches

**Method 1:**

git branch Payman *#creates the Payman branch*

git checkout Payman *#switches to the Payman branch*

**Method 2:**

git checkout -b Payman *#creates and switches to the Payman branch in one command*

### Forced Switching

Sometimes git will prevent you from switching to another branch because you have unstaged changes in your working directory. If you want to override git and force the switch, use:

git checkout -f Payman #forces git to switch to the Payman branch

git checkout --force Payman #another way of forcing a switch

### Merging Branches

git checkout main *#switches to the main branch*

git merge Payman *#merges the Payman branch with the master branch*

### Aborting a merge

git merge –abort

### Conflict Resolution

git status *#determines the files in conflict*

Open the conflicting files and edit the highlighted section until it reflects the state you want in the merged result.

### Remote Branches

**Listing existing remotes:**

git remote -v

**Adding new remotes**

*Syntax:* git remote add <branchname> <URL>

*Example:* git remote add Payman <https://github.com/CAWaterBoardDataCenter/DWRAT_DataScraping.git>

### Deleting a Branch

git branch -d Payman

## Undoing Changes

<https://github.blog/2015-06-08-how-to-undo-almost-anything-with-git/>

### Undo public changes

git revert <SHA> #undoes the commit ID specified

### Fix the last commit message

git commit --amend -m

### Undo local changes that have been saved but not committed

git checkout -- <bad filename>

### Undo local changes that have been committed but not pushed

git reset <last good SHA> #contents remain on disk but removed from repository's history.

git reset --hard <last good SHA> #contents removed from disk.

### Undoing a reversion that went wrong

Sometimes a reversion can accidentally delete stuff you didn't intend; to abort the reversion, run:

git revert --abort

## Syncing your latest local changes to the main branch

## *These steps bypass the GitHub pull request process that I normally recommend.*

* 1. Stage your latest code changes in git (you could be using RStudio, VSCode, or Command Prompt to access git)

git add \*

* 1. Commit your latest code changes

git commit -m “Type message describing your changes”

* 1. Push your latest code changes to the remote version of your branch

git push origin Payman

* 1. After git gives you a confirmation message, double-check that your changes were applied by visiting the remote version of your branch on the GitHub repository.
  2. Return to your git editor.
  3. Switch to the main branch; if git doesn’t allow you to switch, force the switch.

git checkout -main

git checkout -f main #override git if git initially blocks the switch

* 1. Pull the changes from your branch into the main branch

git pull origin Payman

* 1. You might get blocked due to conflicts; if so, you may need to manually resolve the conflicts.

git reset --hard origin/master –to force a pull from origin/main into your local branch; overrides all conflicts