# 8 Must-Have Git Commands for Data Scientists

**By [Soner Yildirim](https://www.kdnuggets.com/author/soner-yildirim" \o "Posts by Soner Yildirim), Data Science Enthusiast.**

Source: <https://www.kdnuggets.com/2021/10/8-git-commands-data-scientists.html>

Por do sol no horizonte

Descrição gerada automaticamente com confiança média

Photo by [Chang Duong](https://unsplash.com/@iamchang?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText) on [Unsplash](https://unsplash.com/s/photos/team-work?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText).

*Git is a must-have skill for data scientists. Maintaining your development work within a version control system is absolutely necessary to have a collaborative and productive working environment with your colleagues. This guide will quickly start you off in the right direction for contributing to an existing project at your organization.*

After a long period of hard work and dedication, you have landed your first job as a data scientist. The orientation and getting-familiar-with-the-environment period is over. You are now expected to work on real-life projects.

You are assigned a task to write a function that performs a particular task in a project. Your function will be a part of an existing project that is currently running.

You cannot just write the function in your local working environment and share it with an email. It should be implemented in the project. You need to “merge” your function to the current codebase.

In most cases, you will not be the only one who contributes to a project. Consider each contributor is responsible for writing a small part of a project. Without a proper and efficient system, it would be a burdensome and tedious task to combine the parts. As the project gets bigger and bigger, it would be impossible to maintain the process of combining these small parts.

Thankfully, we have Git, which provides a highly practical and seamless operation to track all the changes in a project.

Git is a version control system. It maintains a history of all changes made to the code. The changes are stored in a special database called “repository,” also known as “repo.”

In this article, we will go over 8 basic yet fundamental git commands.

## 1. git clone

Git clone creates a copy of the project in your local working environment. You just need to provide a path for the project. This path can be copied from the project main on the hosting service such as GitLab and GitHub.

# clone with HTTPS

git clone [https://gitlab.com/\*\*\*\*\*\*\*](https://gitlab.com/*******)

# clone with SSH

git clone git@gitlab.com:\*\*\*\*\*\*\*

## 2. git branch

Once you clone the project to your local machine, you only have the master branch. You should make all the changes on a new branch that can be created using the git branch command.

git branch mybranch

Your branch is the copy of the master branch until you make any changes.

## 3. git switch

Creating a new branch does not mean that you are working on the new branch. You need to switch to that branch.

git switch mybranch

You are now on the “mybranch” branch, and you can start making changes.

## 4. git status

It provides a brief summary of the current status. You will see what branch you are working on. It also shows if you have made any changes or anything to commit.

git status

On branch mybranch

nothing to commit, working tree clean

## 5. git add

When you make changes in the code, the branch you work on becomes different from the master branch. These changes are not visible in the master branch unless you take a series of actions.

The first action is the git add command. This command adds the changes to what is called the staging area.

git add

Diagrama

Descrição gerada automaticamente

*Basic git workflow (image by author).*

## 6. git commit

It is not enough to add your updated files or scripts to the staging area. You also need to “commit” these changes using the git commit command.

The important part of the git commit command is the message part. It briefly explains what has been changed or the purpose of this change.

There is not a strict set of rules to write commit messages. The message should not be lengthy, but it should clearly explain what the change is about. I think you will get used to it as you gain experience using git.

git commit -m "Your message"

## 7. git push

The add and commit methods make the changes in your local git repository. In order to store these changes in a remote branch (i.e., master branch), you first need to push your code.

It is worth mentioning that some IDEs like PyCharm allow for committing and pushing from the user interface. However, you still need to know what each command does.

After your branch is pushed, you will see a link in the terminal that will take you to the hosting service website (e.g., GitHub, GitLab). The link will open a page where you can create a merge request.

A merge request is asking the maintainer of the project to “merge” your code to the master branch. The maintainer will first review your code. If the changes are OK, your code will be merged.

The maintainer might also abort your branch and restore the master branch.

## 8. git pull

The purpose of using a version control system is to maintain a project with many contributors. Thus, while you are working on a task in your local branch, there might be some changes in the remote branch.

The git pull command is used for making your local branch up to date. You should use the git pull command to update your local working directory with the latest files in the remote branch.

## Conclusion

We have covered 8 basic, yet fundamental git commands. There are many more git commands that you will need to learn. The ones in this article will be a good start.