# What is version control?

A version control system is a tool that manages changes made to the files and directories in a project. Many version control systems exist; this lesson focuses on one called Git, which is used by many of the data science tools covered in our other lessons. Its strengths are:

* Nothing that is saved to Git is ever lost, so you can always go back to see which results were generated by which versions of your programs.
* Git automatically notifies you when your work conflicts with someone else's, so it's harder (but not impossible) to accidentally overwrite work.
* Git can synchronize work done by different people on different machines, so it scales as your team does.

Version control isn't just for software: books, papers, parameter sets, and anything that changes over time or needs to be shared can and should be stored and shared using something like Git.

Which of the following does Git do?

##### Answer the question

50 XP

##### Possible Answers

* Keep track of changes to files.

press

* Notice conflicts between changes made by different people.

press

* Synchronize files between different computers.

press

* All of the above. **(A)**

press

# Where does Git store information?

Each of your Git projects has two parts: the files and directories that you create and edit directly, and the extra information that Git records about the project's history. The combination of these two things is called a repository.

Git stores all of its extra information in a directory called .git located in the root directory of the repository. Git expects this information to be laid out in a very precise way, so you should never edit or delete anything in .git.

Suppose your home directory /home/repl contains a repository called dental, which has a sub-directory called data. Where is information about the history of the files in /home/repl/dental/data stored?

##### Instructions

50 XP

##### Possible Answers

* /home/repl/.git
* /home/repl/dental/.git **(A)**
* /home/repl/dental/data/.git
* None of the above.

Yes: all of the information about a repository is stored under its root directory.

# How can I check the state of a repository?

When you are using Git, you will frequently want to check the status of your repository. To do this, run the command git status, which displays a list of the files that have been modified since the last time changes were saved.

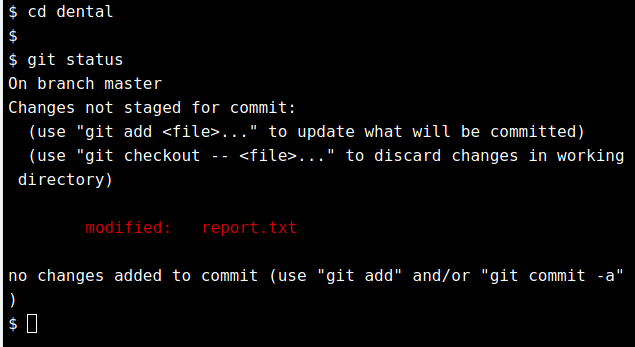
You have been put in the dental repository. Use git status to discover which file(s) have been changed since the last save. Which file(s) are listed?

##### Instructions

50 XP

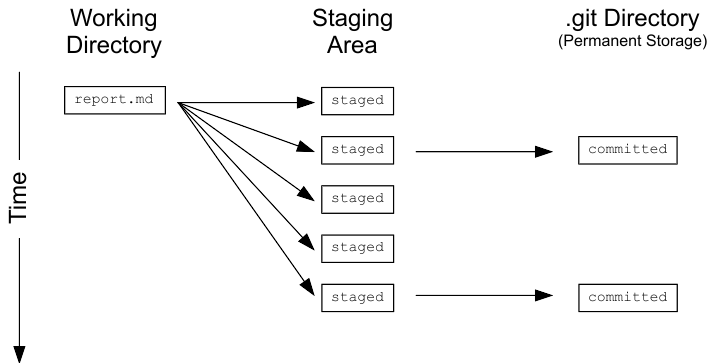
##### Possible Answers

* data/summer.csv.
* report.txt. **(A)**
* Neither of the above.
* Both of the above.



# How can I tell what I have changed?

Git has a staging area in which it stores files with changes you want to save that haven't been saved yet. Putting files in the staging area is like putting things in a box, while committing those changes is like putting that box in the mail: you can add more things to the box or take things out as often as you want, but once you put it in the mail, you can't make further changes.



git status shows you which files are in this staging area, and which files have changes that haven't yet been put there. In order to compare the file as it currently is to what you last saved, you can use git diff filename. git diff without any filenames will show you all the changes in your repository, while git diff directory will show you the changes to the files in some directory.

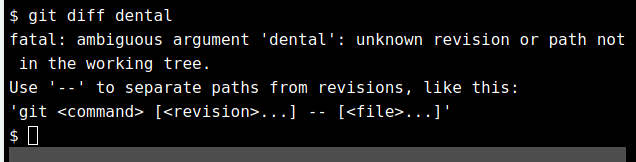
##### Instructions

100 XP

##### Instructions

100 XP

You have been put in the dental repository. Use git diff to see what changes have been made to the files.



# What is in a diff?

A diff is a formatted display of the differences between two sets of files. Git displays diffs like this:

diff --git a/report.txt b/report.txt

index e713b17..4c0742a 100644

--- a/report.txt

+++ b/report.txt

@@ -1,4 +1,5 @@

-# Seasonal Dental Surgeries 2017-18

+# Seasonal Dental Surgeries (2017) 2017-18

+# TODO: write new summary

This shows:

* The command used to produce the output (in this case, diff --git). In it, a and b are placeholders meaning "the first version" and "the second version".
* An index line showing keys into Git's internal database of changes. We will explore these in the next chapter.
* --- a/report.txt and +++ b/report.txt, wherein lines being removed are prefixed with - and lines being added are prefixed with +.
* A line starting with @@ that tells where the changes are being made. The pairs of numbers are start line and number of lines (in that section of the file where changes occurred). This diff output indicates changes starting at line 1, with 5 lines where there were once 4.
* A line-by-line listing of the changes with - showing deletions and + showing additions (we have also configured Git to show deletions in red and additions in green). Lines that haven't changed are sometimes shown before and after the ones that have in order to give context; when they appear, they don't have either + or - in front of them.

Desktop programming tools like [RStudio](https://www.rstudio.com/) can turn diffs like this into a more readable side-by-side display of changes; you can also use standalone tools like [DiffMerge](https://sourcegear.com/diffmerge/) or [WinMerge](http://winmerge.org/).

You have been put in the dental repository. Use git diff data/northern.csv to look at the changes to that file. How many lines have been added or removed?

##### Instructions

50 XP

##### Instructions

50 XP

##### Possible Answers

* None.
* 1. **(A)**
* 2.
* 20.

