Data Science Tools

Engineering productivity tips with Git, Bash and Vim

4.1.2 Main commands

☐ **Tracking status** – We can check previous changes made to the repository with the following commands:

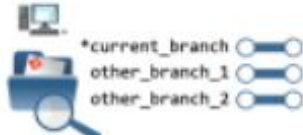| Action	Command	Illustration
Check status of modified file(s)	git status	
View last commits	git log --oneline	
Compare changes made between two commits	git diff commit_1 commit_2	
View list of local branches	git branch	

☐ **Canceling changes** – Canceling changes is done differently depending on the situation that we are in. The table below sums up the most common cases:

Case	Action	Command	Illustration
Unstaged	Revert file to last commit	git checkout -- file	
Staged	Remove file from staging area	git reset HEAD file	
Committed	Go back to a previous commit	git reset --hard prev_commit	

Engineering productivity tips with Git, Bash and Vim

4.1.3 Project structure

☐ **Structure of folders –** It is important to keep a consistent and logical structure of the project. One example is as follows:

```
my_project/
analysis/
    graph/
    notebook/
data/
    query/
    raw/

processed/
modeling/
    method/
    tests
README.md
```

Engineering productivity tips with Git, Bash and Vim

4.2 Working with Bash

☐ **Basic terminal commands** – The table below sums up the most useful terminal commands:

Category	Action	Command
Exploration	Display list of files (including hidden ones)	ls (-a)
	Show current directory	pwd
	Show content of file	cat path_to_file
	Show statistics of file (lines/words/characters)	wc path_to_file
File management	Make new folder	mkdir folder_name
	Change directory to folder	cd path_to_folder
	Create new empty file	touch filename
	Copy-paste file (folder) from origin to destination	scp (-R) origin destination
	Move file/folder from origin to destination	mv origin destination
	Remove file (folder)	rm (-R) path
Compression	Compress folder into file	tar -czvf comp_folder.tar.gz folder
	Uncompress file	tar -xzvf comp_folder.tar.gz
Miscellaneous	Display message	echo "message"
	Overwrite / append file with output	output > file.txt / output >> file.txt
	Execute command with elevated privileges	sudo command
	Connect to a remote machine	ssh remote_machine_address

Engineering productivity tips with Git, Bash and Vim

4.2 Working with Bash

☐ **Chaining** – It is a concept that improves readability by chaining operations with the pipe | operator. The most common examples are summed up in the table below:

Action	Command
Count number of files in a folder	ls path_to_folder | wc -l
Count number of lines in file	cat path_to_file | wc -l
Show last n commands executed	history | tail -n

☐ **Advanced search** – The find command allows the search of specific files and manipulate them if necessary. The general structure of the command is as follows:

```Bash
find path_to_folder/. [conditions] [actions]
```

The possible conditions and actions are summarized in the table below: history | tail -n

Category	Action	Command
Conditions	Certain names, regex accepted	-name 'certain_name'
	Certain file types (d/f for directory/file)	-type certain_type
	Certain file sizes (c/k/M/G for B/kB/MB/GB)	-size file_size -not [condition]
	Opposite of a given condition	
Actions	Delete selected files	-delete -print
	Print selected files	

Engineering productivity tips with Git, Bash and Vim

4.2 Working with Bash

☐ **Changing permissions** – The following command enables to change the permissions of a given file (or folder):

```Bash
chmod (-R) three_digits file
```

with three_digits being a combination of three digits, where:
• the first digit is about the owner associated to the file
• the second digit is about the group associated to the file
• the third digit is anyone irrespective of their relation to the file
Each digit is one of (0, 4, 5, 6, 7), and has the following meaning:

Representation	Binary	Digit	Explanation
---	000	0	No permission
r--	100	4	Only read permission
r-x	101	5	Both read and execution permissions
rw-	110	6	Both read and write permissions
rwx	111	7	Read, write and execution permissions

For instance, giving read, write, execution permissions to everyone for a given_file is done by running the following command:

```Bash
chmod 777 given_file
```

Engineering productivity tips with Git, Bash and Vim

4.2 Working with Bash

☐ **Terminal shortcuts –** The table below summarizes the main shortcuts when working with the terminal:

Action	Command
Search previous commands	Ctrl + r
Go to beginning / end of line	Ctrl + a / Ctrl + e
Remove everything after the cursor	Ctrl + k
Clear line	Ctrl + u
Clear terminal window	Ctrl + l

DID YOU LIKE THIS POST

Follow **SKILLSLASH** for more such valuable content

SKILL<S/ASH>