

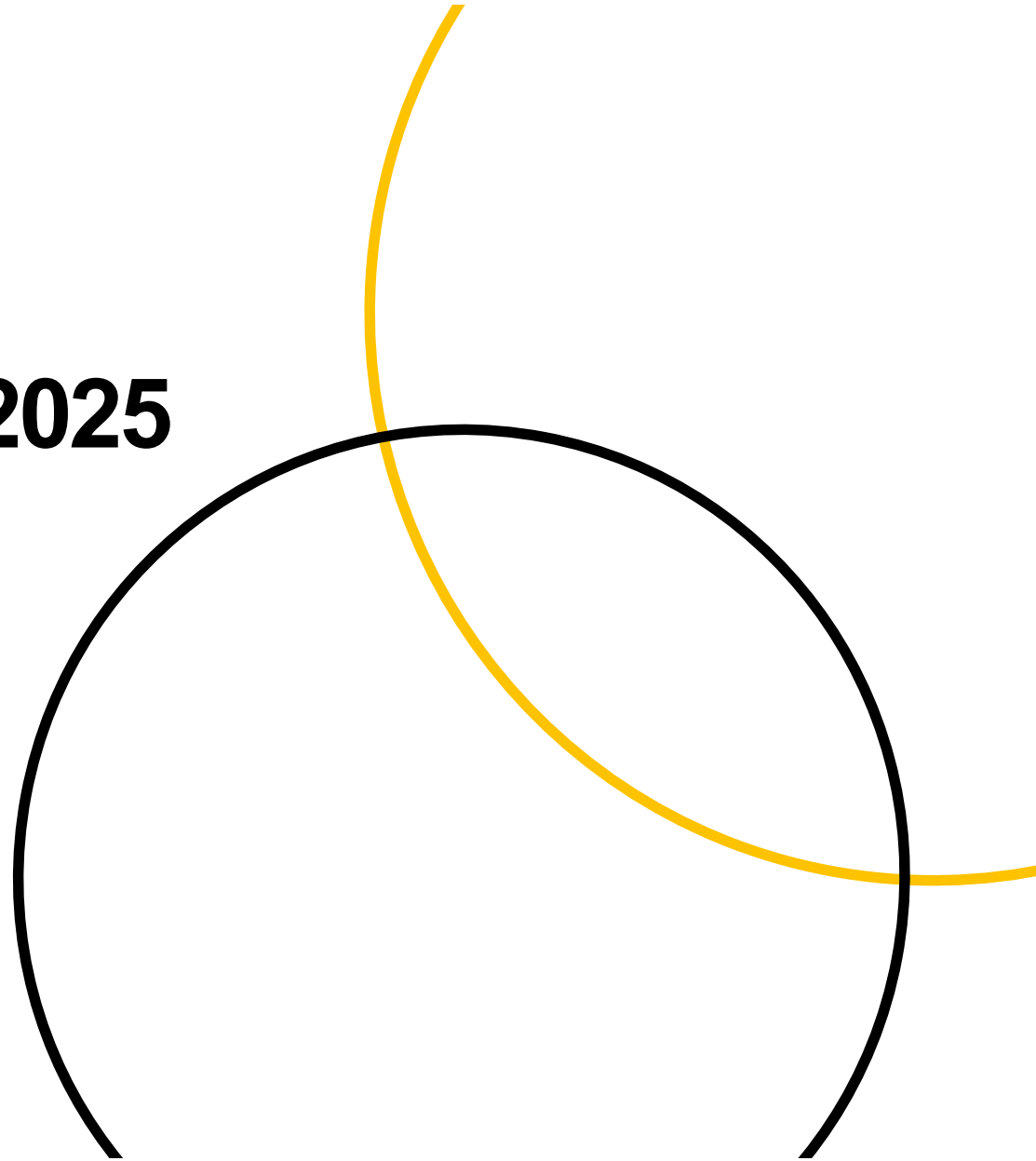
**wellcome
connecting
science**

Polygenic Risk Score Analyses Workshop 2025

Day 1a: GWAS & relevant Statistics

Isewon Itunuoluwa / Melek Chaouch

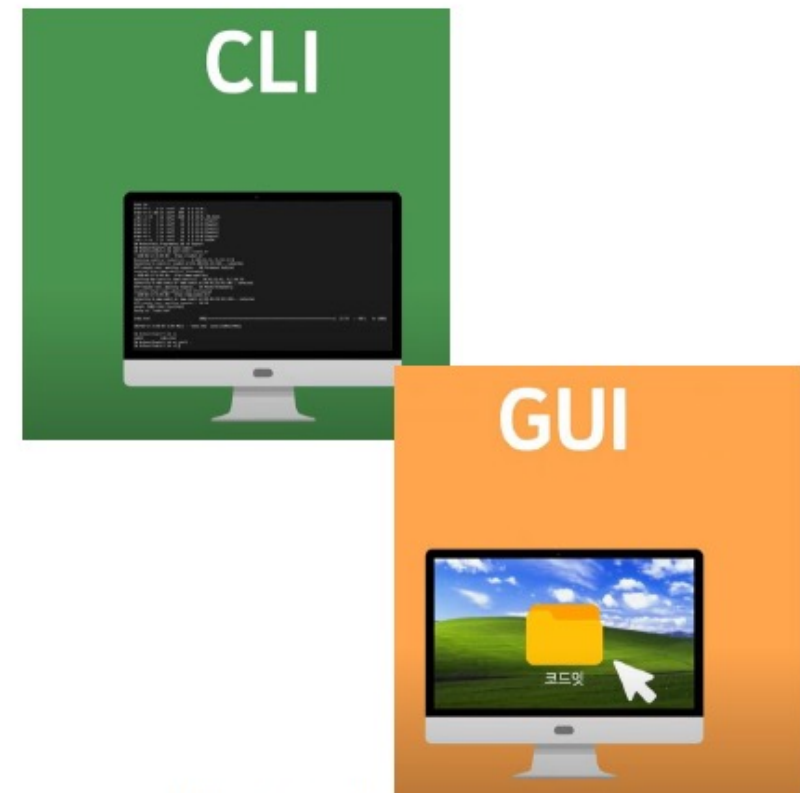
June 2025



What is a Shell?

- A computer program that provides a command-line interface, accessible from the console or a terminal.
- Alternative to using a graphical user interface (GUI).
- Allows users to control the computer using commands entered with a keyboard.
- The user enters commands as text input, which is then interpreted and executed by the shell.
- Linux and macOS : robust command-line interfaces

Command Line Interface (CLI)



<https://medium.com>

How to access the shell ?

- On a Mac or Linux machine, you can access a shell through a program called “Terminal”, which is already available on your computer.
- The Terminal is a window into which we will type commands.
- For scripting and automation, Linux commonly uses:
 - Bash (Bourne Again Shell) – Default scripting language for many Linux distributions.
 - Python – Widely used for automation and system administration.
 - Perl, Ruby, and Shell scripting languages – Used in various system tasks.

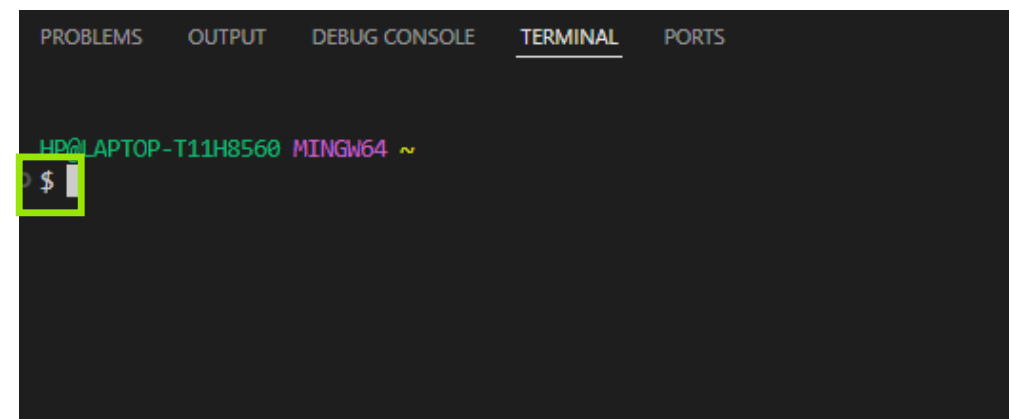
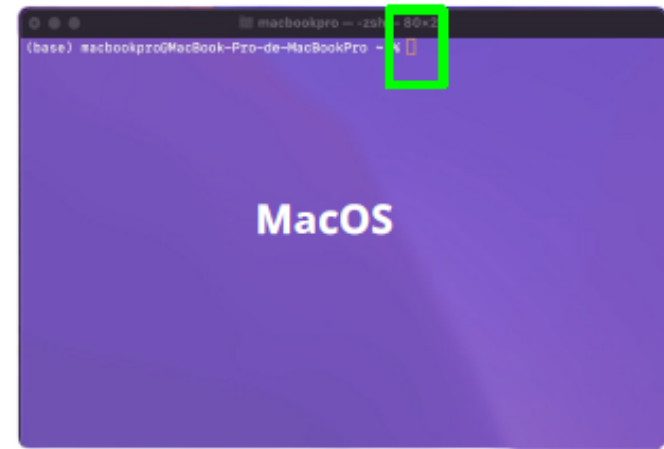
How to access the shell ?

- On a Mac or Linux machine, you can access a shell through a program called “Terminal”, which is already available on your computer.
- The Terminal is a window into which we will type commands.
- For scripting and automation, Linux commonly uses:
 - Bash (Bourne Again Shell) – Default scripting language for many Linux distributions.
 - Python – Widely used for automation and system administration.
 - Perl, Ruby, and Shell scripting languages – Used in various system tasks.

How to access the shell ?

- The dollar sign is a prompt,
- Prompt shows that the shell is waiting for input
- Your shell may use a different character as a prompt and may add information before the prompt.

When typing commands, either from these lessons or from other sources, do not type the prompt, only the commands that follow it !!!



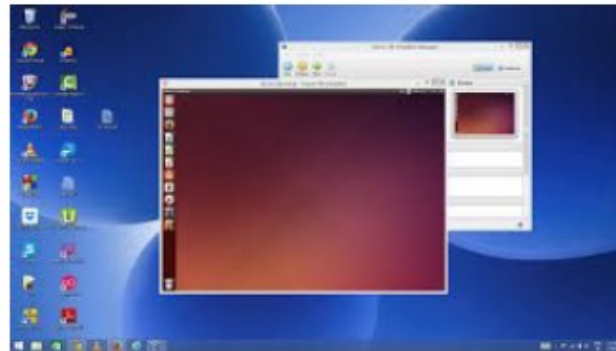
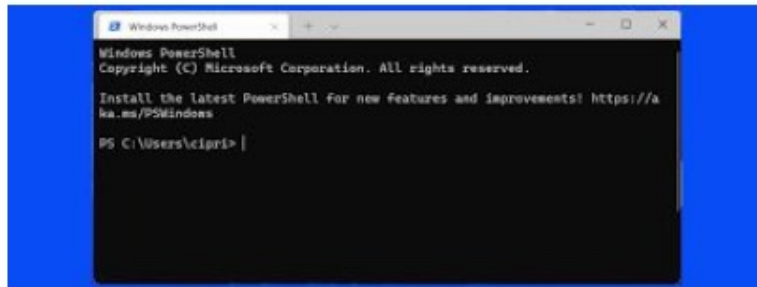
Shell for Windows users

- If you're using **Windows**, you'll need to :

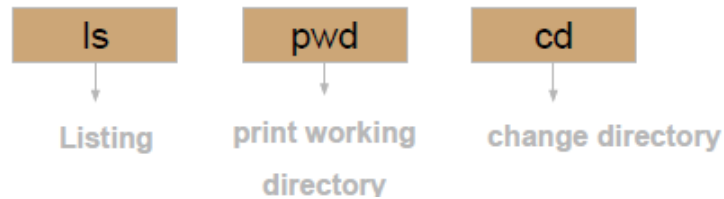
- download a separate program to access the shell.

VirtualBox is a free and open-source virtualization software that allows you to run multiple operating systems (like Linux) inside Windows as a virtual machine (VM).

- Install Ubuntu



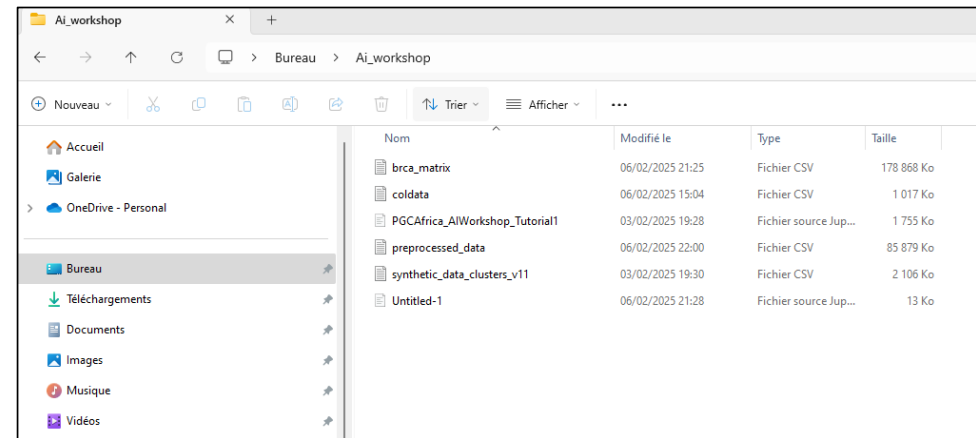
Navigating your file system



File system : The part of the operating system that manages files and directories.

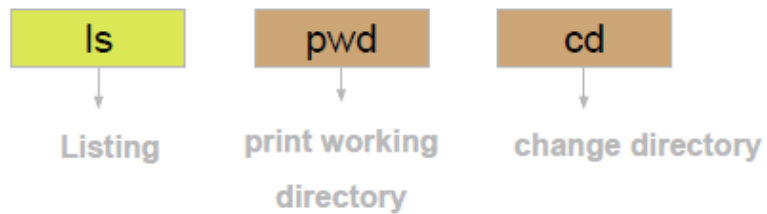
It organizes our data into :

- Files \Rightarrow hold information
- Directories (also called “folders”) \Rightarrow hold files or other directories



```
PS C:\Users\HP\Desktop> tree
Structure du dossier pour le volume Windows
Le numéro de série du volume est 3EE3-3E01
C:..
├── Adobe professionnel
│   ├── Adobe Acrobat X
│   └── Crack-MPT[h33t][eSpNs]
├── Ai_workshop
├── Office 2016
│   └── Office_2016_French_Language_Pack_-x64.MyEgy.Tv
│       ├── access.fr-fr
│       ├── catalog
│       ├── dcf.fr-fr
│       ├── excel.fr-fr
│       ├── groove.fr-fr
│       ├── hotfixes
│       ├── infopath.fr-fr
│       ├── lync.fr-fr
│       ├── office.fr-fr
│       ├── office32.fr-fr
│       ├── omui.fr-fr
│       ├── onenote.fr-fr
│       ├── osm.fr-fr
│       ├── osmux.fr-fr
│       ├── outlook.fr-fr
│       ├── pmui.fr-fr
│       ├── powerpoint.fr-fr
│       ├── project.fr-fr
│       └── proofing.fr-fr
```

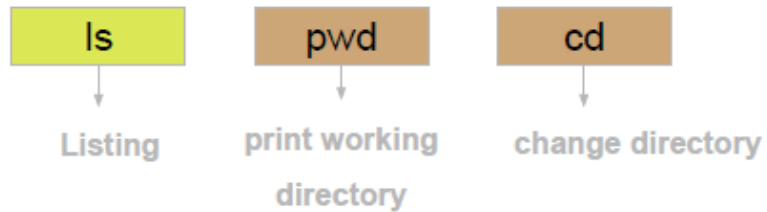
Navigating your file system



- `ls` : prints the names of the files and directories in the current directory in alphabetical order, arranged neatly into columns.
- `'*'` option : Command displays the all file type.

```
Jun 16 14:50
manager@PRS25: ~
(base) manager@PRS25:~$ mkdir my_file.txt
(base) manager@PRS25:~$ ls
BridgePRS      Music          PRSice_linux.zip  toy.map
Data           my_file.txt    PRSice.R          toy.ped
Desktop        Pictures       Public            TOY_TARGET_DATA.bed
Documents      plink_linux_x86_64_20201019.zip  R               TOY_TARGET_DATA.bim
Downloads      plink_linux_x86_64_20201019.zip.1 rstudio-2024.12.1-563-amd64.deb TOY_TARGET_DATA.fam
LICENSE        prettify       snap             TOY_TARGET_DATA.pheno
miniforge3     PRSicsx        Templates
Miniforge3-Linux-x86_64.sh PRSice_linux   TOY_BASE_GWAS.assoc
Videos
```


Navigating your file system



man (short for manual) displays detailed documentation (also referred as **man page** or man file) for bash commands.

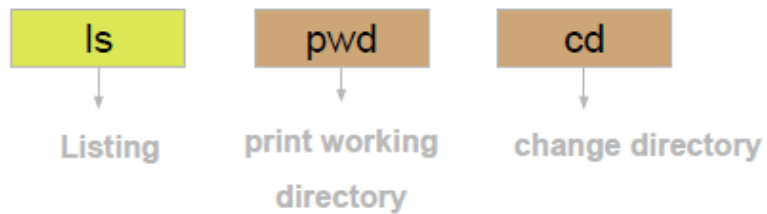
It is a powerful resource to explore bash commands, understand their usage and flags. Some manual files are very long.

You can scroll through the file using your keyboard's down arrow or use the **Space** key to go forward one page and the **b** key to go backwards one page.

When you are done reading, hit **q** to quit.

```
manager@PRS25: ~  
LS(1) User Commands LS(1)  
NAME  
ls - list directory contents  
SYNOPSIS  
ls [OPTION]... [FILE]...  
DESCRIPTION  
List information about the FILES (the current directory by default). Sort entries alphabetically if none of  
-cftuvSUX nor --sort is specified.  
Mandatory arguments to long options are mandatory for short options too.  
-a, --all  
do not ignore entries starting with .  
-A, --almost-all  
do not list implied . and ..  
--author  
with -l, print the author of each file  
-b, --escape  
print C-style escapes for nongraphic characters  
Manual page ls(1) line 1 (press h for help or q to quit)
```

Navigating your file system

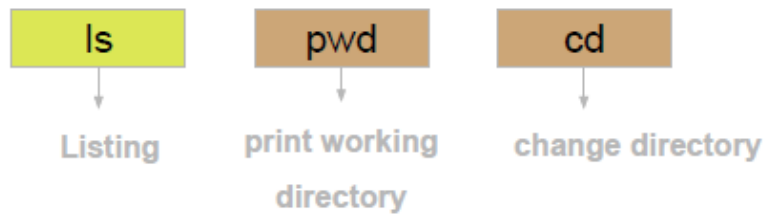


We can make the `ls` output more comprehensible by using the flag `-F`

- `ls -F` command lists files and directories with indicators.
 - Directories are shown with a trailing slash (`/`).
 - Executable files are marked with an asterisk (`*`).
- Regular files have no special indicator.
- `ls -a` : see hidden directory/file that start with `."`.

The hidden files/folders are usually system-related data, configuration settings, or temporary files.

Navigating your file system

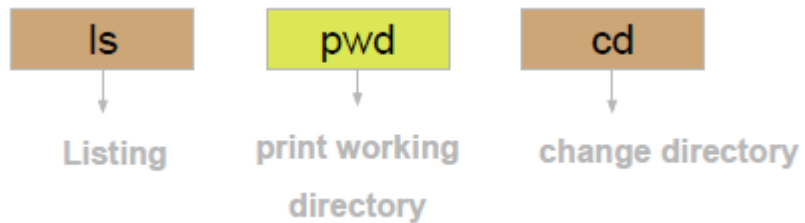


We can make the `ls` output more comprehensible by using the flag `-F`

- `ls -F` command lists files and directories with indicators.
 - Directories are shown with a trailing slash (`/`).
 - Executable files are marked with an asterisk (`*`).
- Regular files have no special indicator.
- `ls -a` : see hidden directory/file that start with `."`.

The hidden files/folders are usually system-related data, configuration settings, or temporary files.

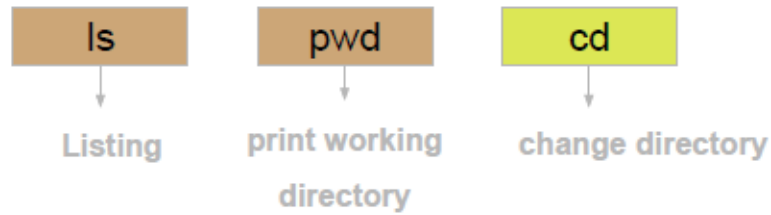
Navigating your file system



`pwd` : Prints the absolute path of the current working directory.

```
Jun 16 14:38
manager@PRS25: ~/Data
(base) manager@PRS25:~/Data$ pwd
/home/manager/Data
(base) manager@PRS25:~/Data$
```

Navigating your file system

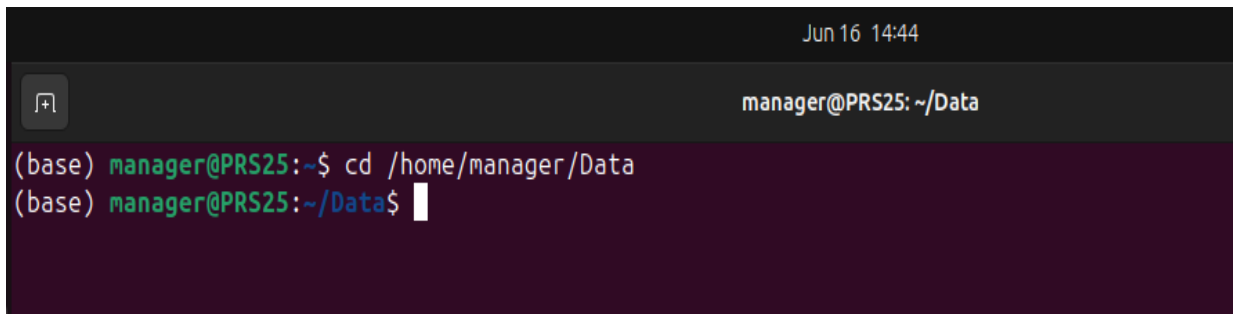


- The command to change locations in our file system is `cd`, followed by a directory name to change our working directory.
- Typing out file or directory names can waste a lot of time and it's easy to make typing mistakes.
- Instead, we can use tab complete as a shortcut.
- When you start typing out the name of a directory or file, then hit the Tab key, the shell will try to fill in the rest of the directory or file name.

```
Jun 16 14:39
manager@PRS25: ~/Data
(base) manager@PRS25:~$ cd Data/
(base) manager@PRS25:~/Data$
```

Full vs. Relative Paths

- The directories on the computer are arranged into a hierarchy.
- The `cd` command takes an argument which is a directory name.
- Directories can be specified using either a relative path or a full absolute path.

A terminal window with a dark background. The title bar at the top shows "Jun 16 14:44" on the right and a window icon on the left. Below the title bar, the text "manager@PRS25: ~/Data" is displayed. The terminal shows two lines of input: the first line is "(base) manager@PRS25:~\$ cd /home/manager/Data" and the second line is "(base) manager@PRS25:~/Data\$" with a cursor at the end.

```
(base) manager@PRS25:~$ cd /home/manager/Data
(base) manager@PRS25:~/Data$
```

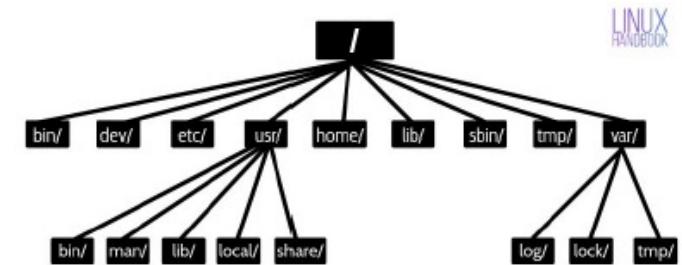
Absolute path, giving the full address from the home directory.

The very top of the hierarchy is a directory called `/` which is usually referred to as the root directory.

Relative path, giving only the address from the working directory.

Navigational Shortcuts : Root Directory

- The root directory (`/`) is the top-level directory of the file system.
- It contains critical system files and should not be directly modified.
- Avoid working in the root directory to prevent unintended system changes.
- The home directory is where users typically store their files.
- The tilde (`~`) is a shortcut that represents the user's home directory.
- The commands `cd`, and `cd ~` are very useful for quickly navigating back to your home directory.



Working with Files and Directories : create a folder

mkdir

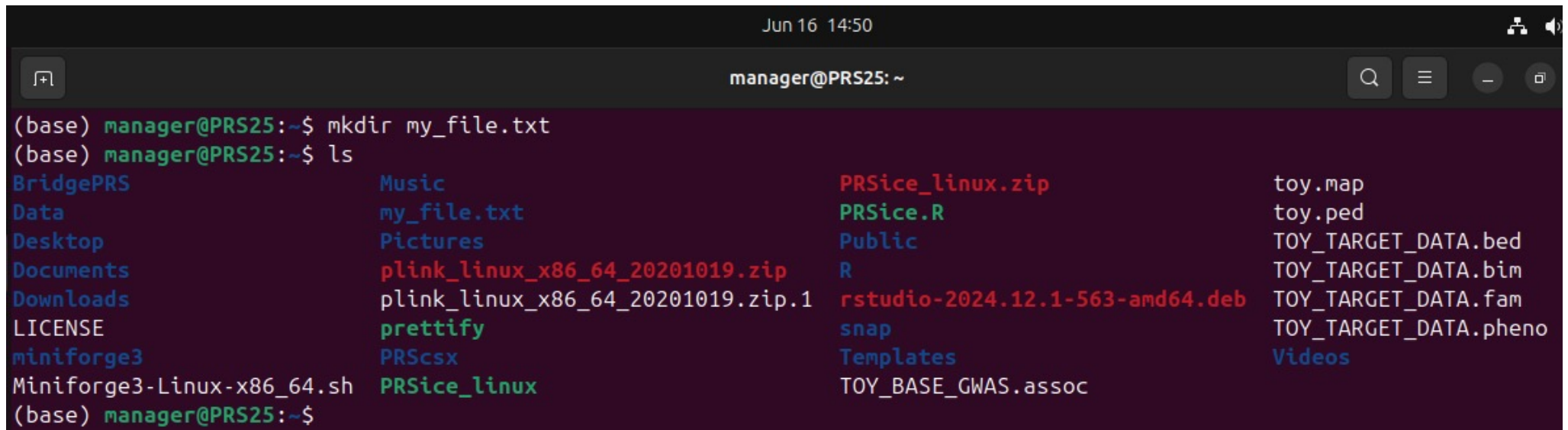
cp

mv

rm

The mkdir command is used to make a directory.

Enter mkdir followed by a space, then the directory name you want to create.

A terminal window titled 'manager@PRS25: ~' with a search bar and window controls. The terminal shows the execution of 'mkdir my_file.txt' and 'ls'. The 'ls' output lists various files and directories in a multi-column format, including 'BridgePRS', 'Data', 'Desktop', 'Documents', 'Downloads', 'LICENSE', 'miniforge3', 'Miniforge3-Linux-x86_64.sh', 'Music', 'my_file.txt', 'Pictures', 'plink_linux_x86_64_20201019.zip', 'plink_linux_x86_64_20201019.zip.1', 'prettify', 'PRScsx', 'PRsice_linux', 'PRsice_linux.zip', 'PRsice.R', 'Public', 'R', 'rstudio-2024.12.1-563-amd64.deb', 'snap', 'Templates', 'TOY_BASE_GWAS.assoc', 'toy.map', 'toy.ped', 'TOY_TARGET_DATA.bed', 'TOY_TARGET_DATA.bim', 'TOY_TARGET_DATA.fam', 'TOY_TARGET_DATA.pheno', and 'Videos'.

```
(base) manager@PRS25:~$ mkdir my_file.txt
(base) manager@PRS25:~$ ls
BridgePRS      Music          PRsice_linux.zip  toy.map
Data           my_file.txt    PRsice.R          toy.ped
Desktop        Pictures       Public            TOY_TARGET_DATA.bed
Documents      plink_linux_x86_64_20201019.zip  R               TOY_TARGET_DATA.bim
Downloads      plink_linux_x86_64_20201019.zip.1 rstudio-2024.12.1-563-amd64.deb TOY_TARGET_DATA.fam
LICENSE        prettify       snap              TOY_TARGET_DATA.pheno
miniforge3     PRScsx         Templates
Miniforge3-Linux-x86_64.sh PRsice_linux   TOY_BASE_GWAS.assoc
Videos
```

June 2025

Working with Files and Directories : copy file/folder

mkdir

cp

mv

rm

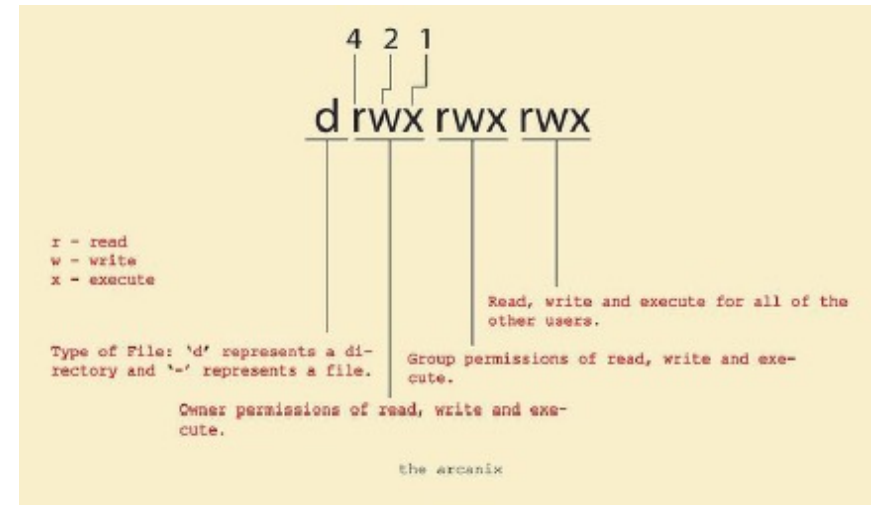
Copying Files : When working with computational data, it's important to keep a safe copy of that data that can't be accidentally overwritten or deleted.

For this lesson, our raw data is our txt files. We don't want to accidentally change the original files, so we'll make a copy of them

```
Jun 16 14:55
manager@PRS25: ~/Data/Day1a_Data
(base) manager@PRS25:~/Data/Day1a_Data$ cp GIANT_Height.txt GIANT_Height-copy.txt
(base) manager@PRS25:~/Data/Day1a_Data$ ls -l
total 295584
drwxrwxr-x 2 manager manager 4096 Jun 16 14:52 backup
-rw-r--r-- 1 manager manager 100849786 Jun 16 14:54 GIANT_Height-copy.txt
-rw-r--r-- 1 manager manager 100849786 Sep  6 2022 GIANT_Height_st.txt
-rw-r--r-- 1 manager manager 100849786 Jun  2 2018 GIANT_Height.txt
-rw-r--r-- 1 manager manager 98817 Jun 14 2022 GIANT_Height.txt.zip
-rw-r--r-- 1 manager manager 120 Jun  4 2018 Select.sample
-rw-r--r-- 1 manager manager 8519 Jun  7 2018 TAR.height
```

Working with Files and Directories : File permission

- Keep a safe copy of data that can't be accidentally overwritten or deleted.
- View the current permissions on a file using the -l (long) flag for the ls command.
- 10 slots in the permissions list: first character is file type, next 3 are owner permissions, next 3 are group permissions, last 3 are everyone else.



Working with Files and Directories : File permission

The "chmod" command modifies the read, write, and execute permissions of specified files and the search permissions of specified directories

```
chmod <operation> <file/dir>    # To change permissions of a file or
                                   directory.
chmod +r sample.sh               # Granting read permission to a file.
chmod +w sample.sh               # Granting write permission to a file.
chmod +x sample.sh               # Granting execute permission to a file.
```

```
chmod -r sample.sh               # Revoking read permission from a file
chmod -w sample.txt              # Revoking write permission from a file
chmod -x sample.txt              # Revoking execute permission from a
                                   file.

sudo chmod -R -r directory       # Revoking read permission from a
                                   directory recursively.
```

Working with Files and Directories: Moving / Renaming

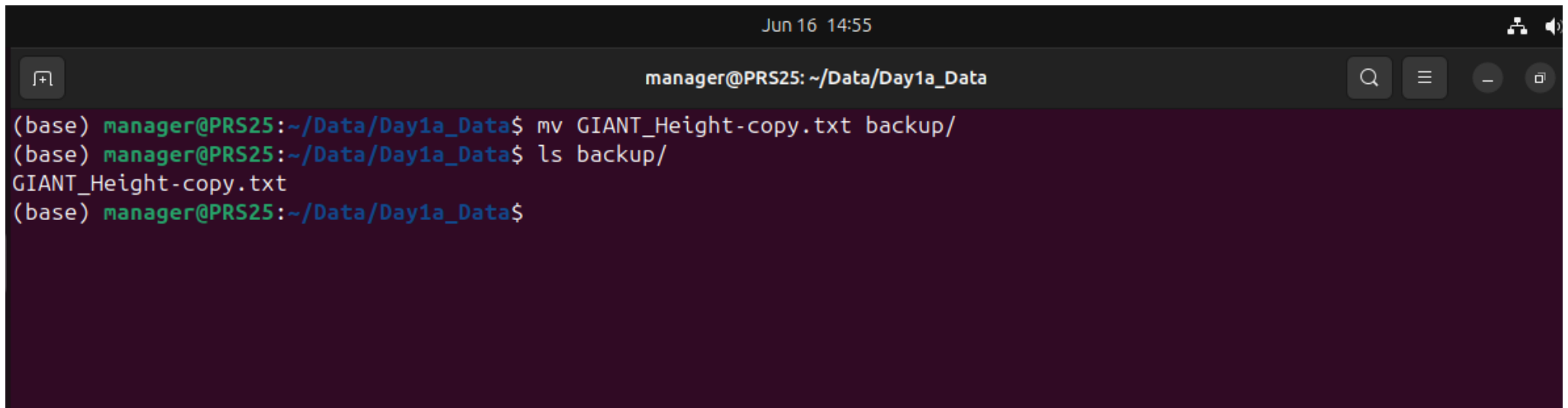
mkdir

cp

mv

rm

- We can now move our backup file to another directory/file using the command mv
- The mv command is also how you rename files.



```
Jun 16 14:55
manager@PRS25: ~/Data/Day1a_Data
(base) manager@PRS25:~/Data/Day1a_Data$ mv GIANT_Height-copy.txt backup/
(base) manager@PRS25:~/Data/Day1a_Data$ ls backup/
GIANT_Height-copy.txt
(base) manager@PRS25:~/Data/Day1a_Data$
```

Working with Files and Directories: Moving / Renaming

mkdir

cp

mv

rm

The `rm` command permanently removes the file. Be careful with this command. It doesn't just nicely put the files in the Trash. They're really gone.

By default, `rm` will not delete directories. You can tell `rm` to delete a directory using the `-r` (recursive) option. Let's delete the backup directory we just made.

Examining Files

cat

less

head

tail

wc

grep

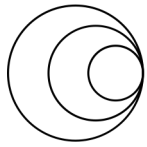
- One way to examine a file is to print out all of the contents using the program **cat**. This will print out all of the contents of the file to the screen.
- One way to examine a file is to print out all of the contents using the program cat. This will print out all of the contents of the file to the screen. **cat** is a terrific program, but when the file is really big, it can be annoying to use.
The program, **less**, is useful for this case. less opens the file as read only, and lets you navigate through it.
- The **wc** (word count) command is used to count the **numb**
- **grep** : Identifies sequences matching a specific pattern, We can use the -B argument for grep to return a specific number of lines before each match. The -A argument returns a specific number of lines after each matching line.

Filtering and Reshuffling Files

A very powerful feature of the terminal is the **awk** programming language, which allows us to extract subsets of a data file, filter data according to some criteria or perform arithmetic operations on the data. **awk** manipulates a data file by performing operations on its **columns** - this is extremely useful for scientific data sets because typically the columns features or variables of interest.

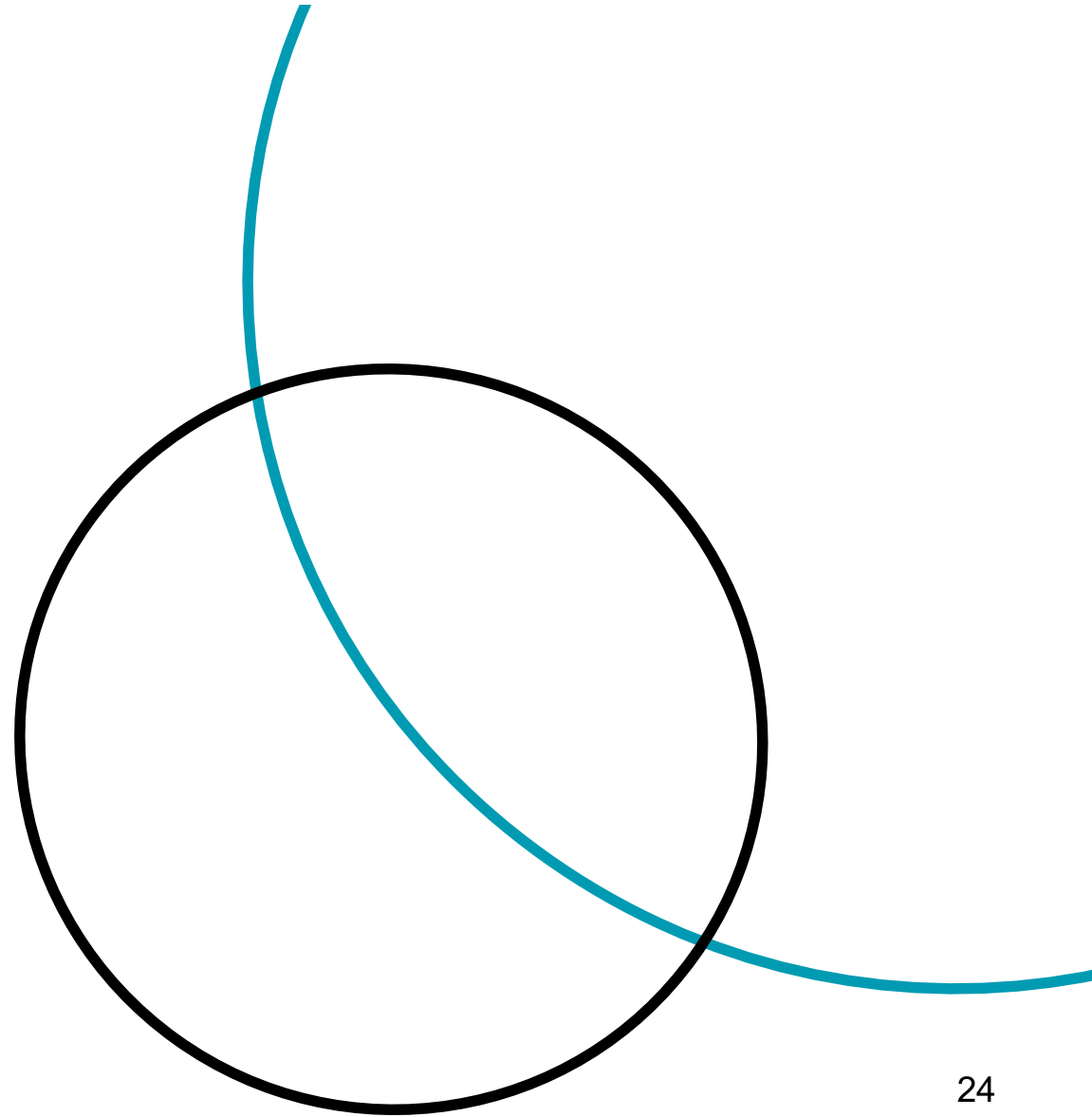
For example, we can use awk to produce a new results file that only contains SNP rsIDs (column 1), allele frequencies (column 4) and P -values (column 7) as follows:

```
awk '{ print $1,$4,$7}' GIANT_Height.txt > GIANT_Height_3cols.txt
```



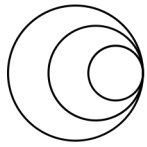
**wellcome
connecting
science**

questions?



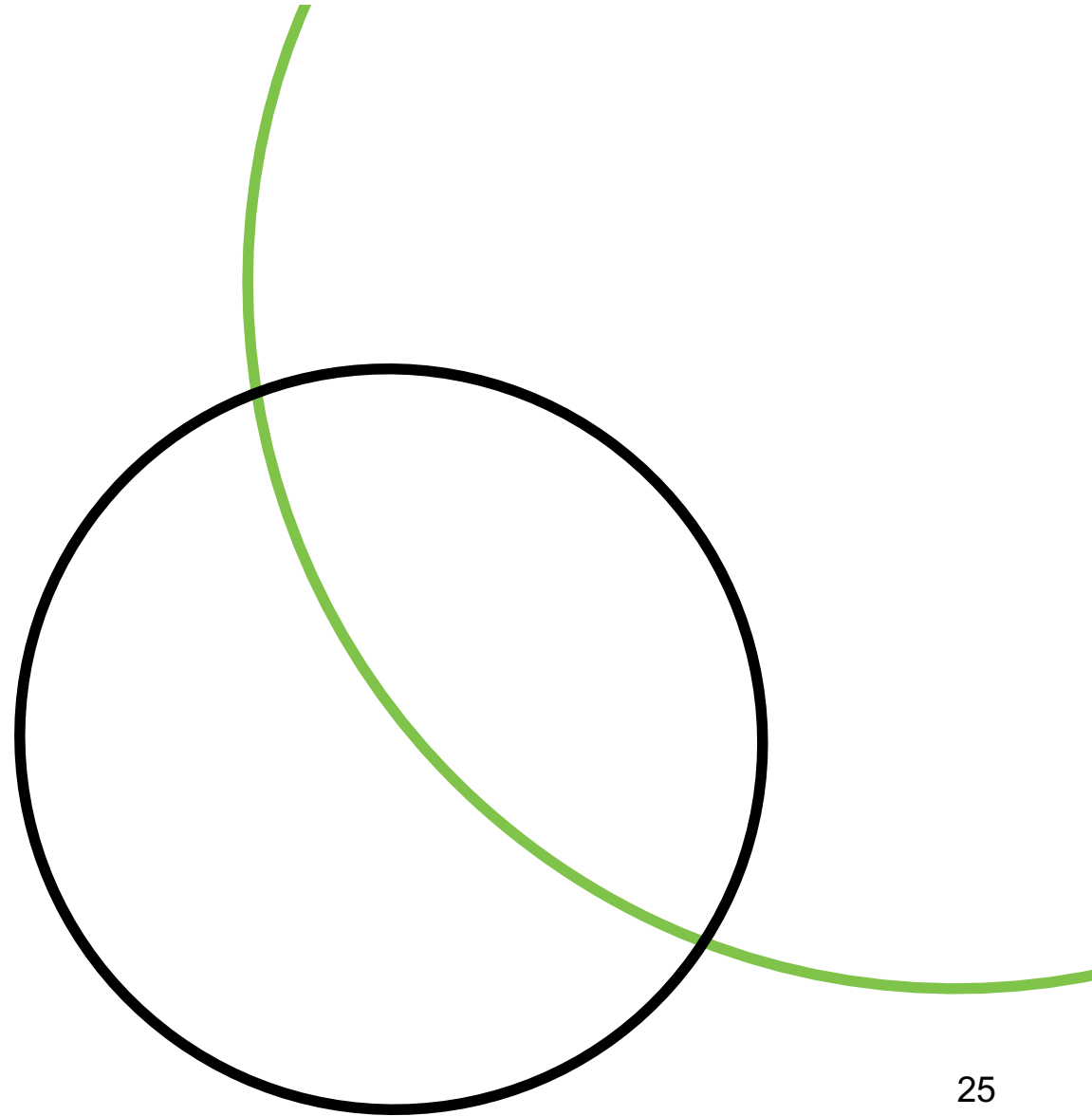
May 2021

wellcome
connecting
science
training | learning | engagement | society



**wellcome
connecting
science**

thanks!



May 2021

wellcome
connecting
science
training | learning | engagement | society