

# **Basic R Programming**

## **(and how to talk to a computer)**

**Arporn Wangwiwatsin, PhD**

**Department of Biochemistry, Faculty of Medicine, Khon Kaen University**  
**Cholangiocarcinoma Research Institute, Khon Kaen University**  
**Khon Kaen University International Phenome Laboratory**

# Today plan

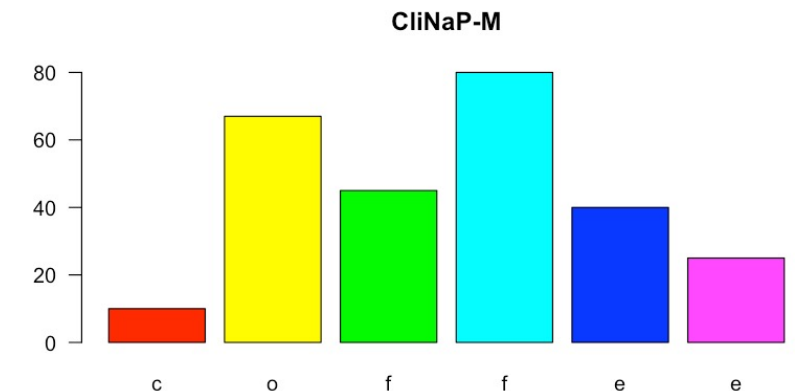
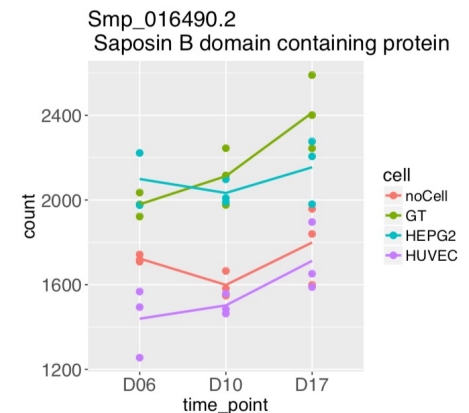
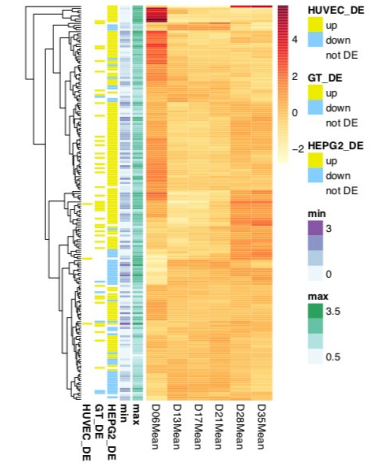
- Overview of R (~10 mins)
- Hands-on with R programming (~20 mins)

# Tomorrow

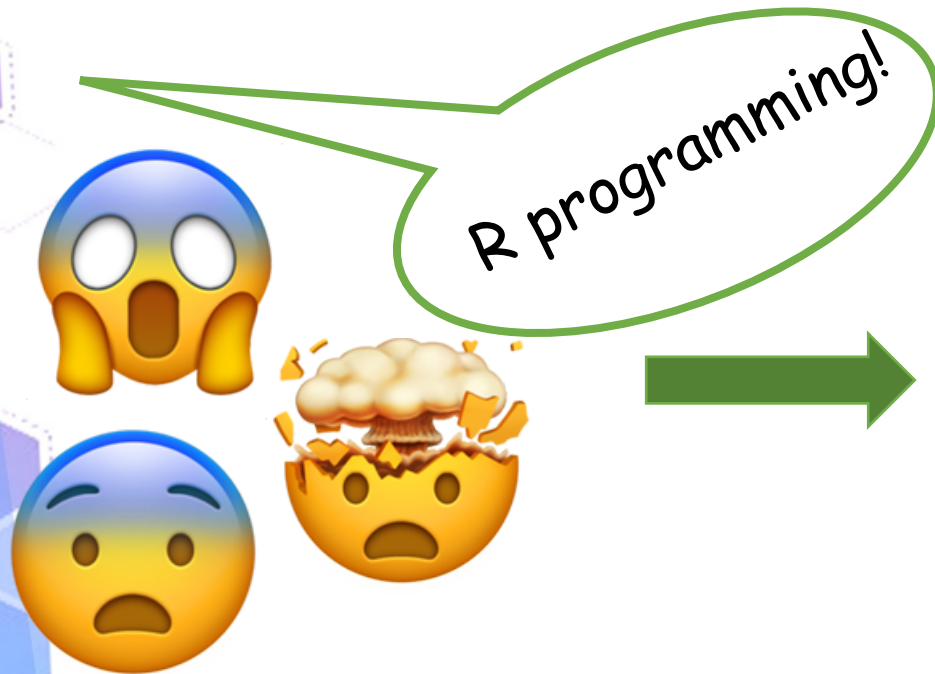
- Final point before NMR data analysis session (~5 mins)

# What is R, and what is it for?

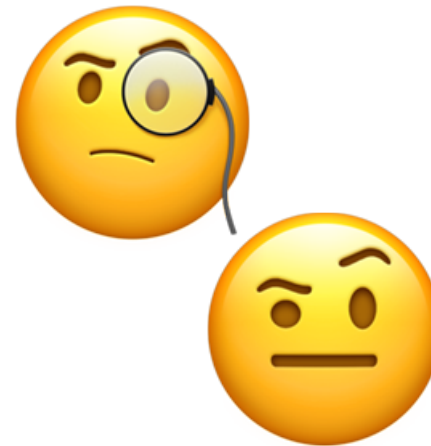
- **A language** for talking to computer
- Data exploration and **visualisation**
- Can produce publication-quality **plots**
- Can perform mathematical and **statistical analysis**
- Use **command line**
  - Allow better **reproducibility** / pipeline development
  - Fast if re-analysis is required
- **Freeware** + >1000s community developers
- Work on Linux, Windows, MacOS



# What this session is



Before



After

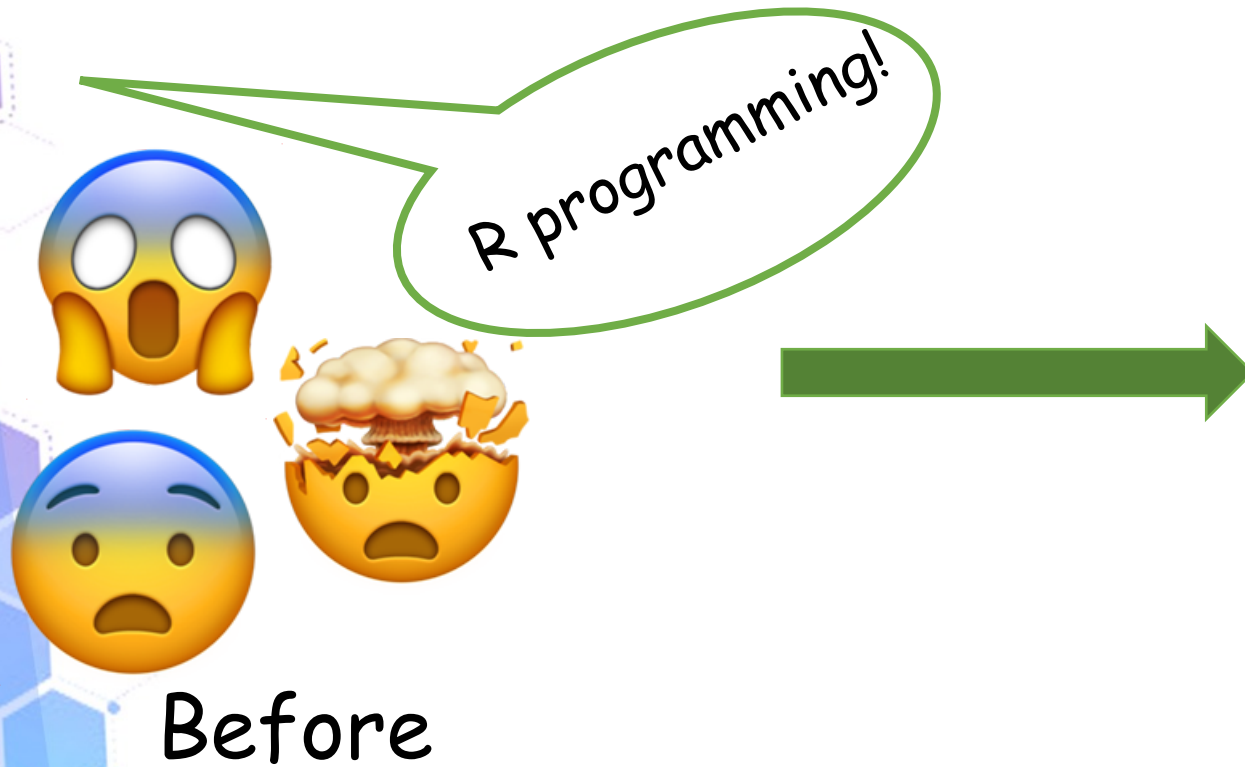
"I can read simple R scripts"  
"I know names of some R functions"



And maybe

"I want to write my own R script"  
"I know where to 'copy' sample R scripts"

# What this session is NOT



After

"I can write my own R scripts from scratch"



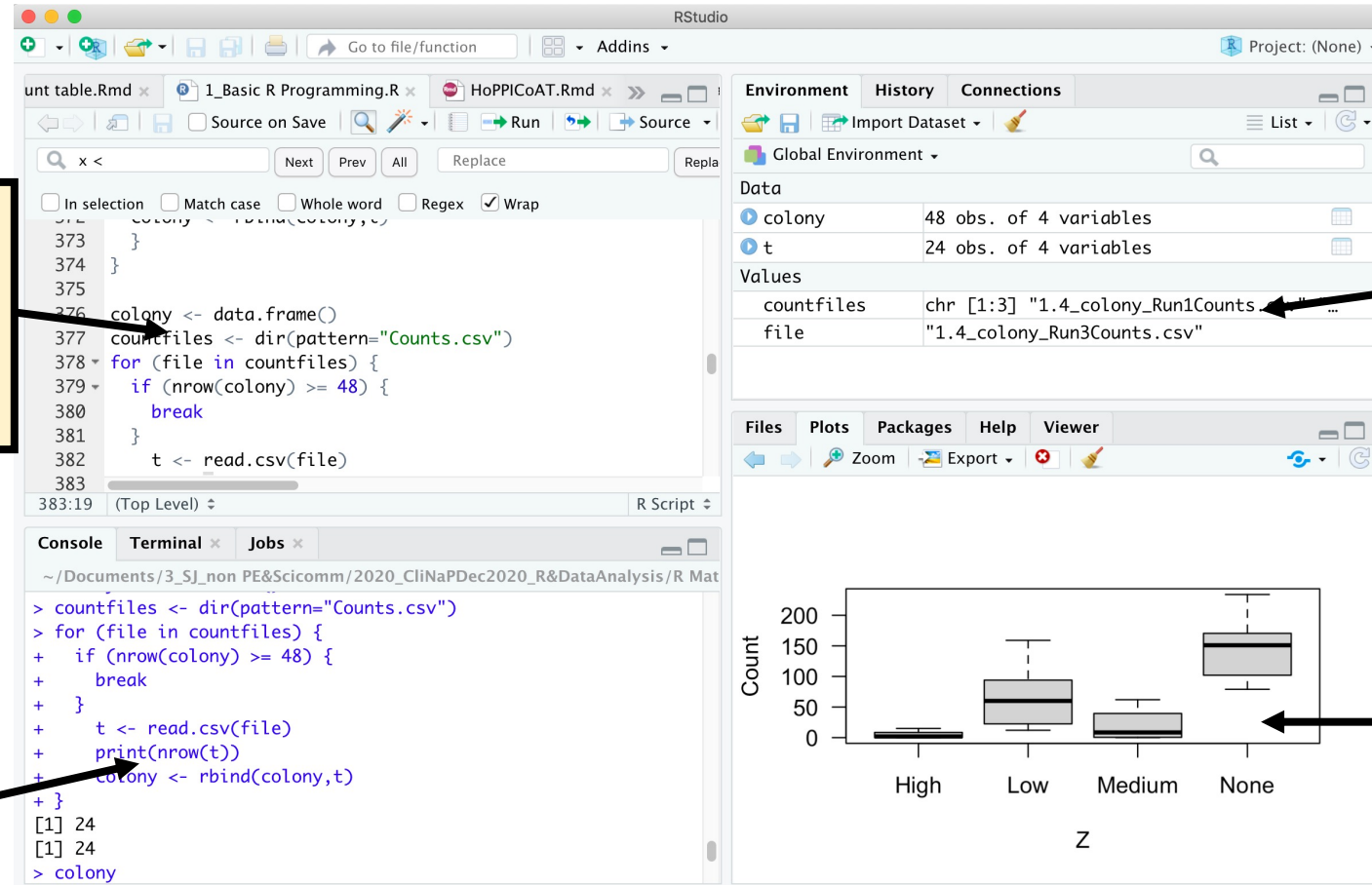
# A convenient way to use R



Store and edit  
your script here

“Environment”  
listing your data  
objects

Scripts that were  
run + text output +  
error message  
appear here



Graphical output  
appear here

# R is a (computer) language

- Languages have **nouns**
- Languages have **verbs**
- Languages have **syntax (grammar)**

# R is a (computer) language

- Languages have nouns

- In R, we call this “**Objects**” or “**variables**”

“**Objects**” or “**variables**” come in various shapes

- Vector (1-dimension)
- Dataframe (2-dimension / table)
- Matrix (2-dimension / table with one data type)
- List (collection of multiple objects)

- Objects store **data**.

“**Data**” can be...

- **Number** (numerical)
- **Character** (text)
- **Factor** (grouping)
- **TRUE / FALSE** (logical)



# R is a (computer) language

- Languages have verbs

- In R, we call this “**functions**” or “**commands**”
- Functions do something to the input object(s) and give some output
- Output can be **displayed** on console, or **stored** in another object
- Functions can have “**options**” to **fine tune** its performance

Adjective/Adverb

(“**Options**” are also known as “**arguments**”)

# R is a (computer) language

- Languages have syntax (grammar)

R code normally looks like this...

```
some_function(input_object, options)
```

Verb                      Noun                      Adjective/Adverb

Code like this usually gives output on the screen, as text or graphic

# R is a (computer) language

- Languages have syntax (grammar)

R code normally looks like this...

```
some_function(input_object, options)
```

Verb                      Noun                      Adjective/Adverb

Code like this usually gives output on the screen, as text or graphic

```
= some_function(input_object, options)
```

                    Verb                      Noun                      Adjective/Adverb

**<-** or **=** symbols can be used to assign value to an object.

Code like this put output in the **newObject**. Usually, nothing will be displayed on screen

# R is a (computer) language

- Languages have syntax (grammar)

R code normally looks like this...

```
some_function(input_object, options)
```

Verb                      Noun                      Adjective/Adverb

Code like this usually gives output on the screen, as text or graphic

```
newObject = some_function(input_object, options)
```

                    =                      Verb                      Noun                      Adjective/Adverb

Code like this put output in the **newObject**. Usually, nothing will be displayed on screen

**<-** or **=** symbols can be used to assign value to an object.

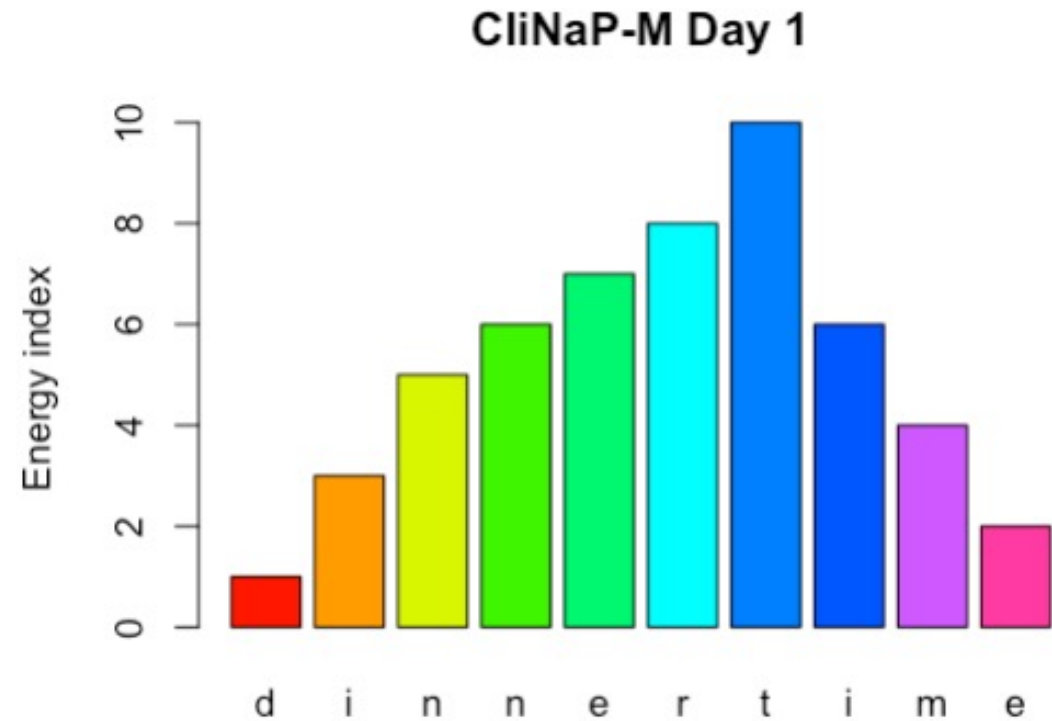
# Group exercise

Create this image by modifying the command below

```
data <- c(1,3,5,6,7,8,10,6,4,2)

__plot(height = data, names.arg =
c(__,'__','__','__','__','__','__','__','__','__'),
__),

col = rainbow(__),
__ = "CliNaP-M Day 1",
__ = "_____")
```







# R is a (computer) language

- Languages have verbs

- In R, we call this “**functions**” or “**commands**”
- Functions do something to the input object(s) and give some output
- Output can be **displayed** on console, or **stored** in another object
- Functions can have “**options**” to **fine tune** its performance

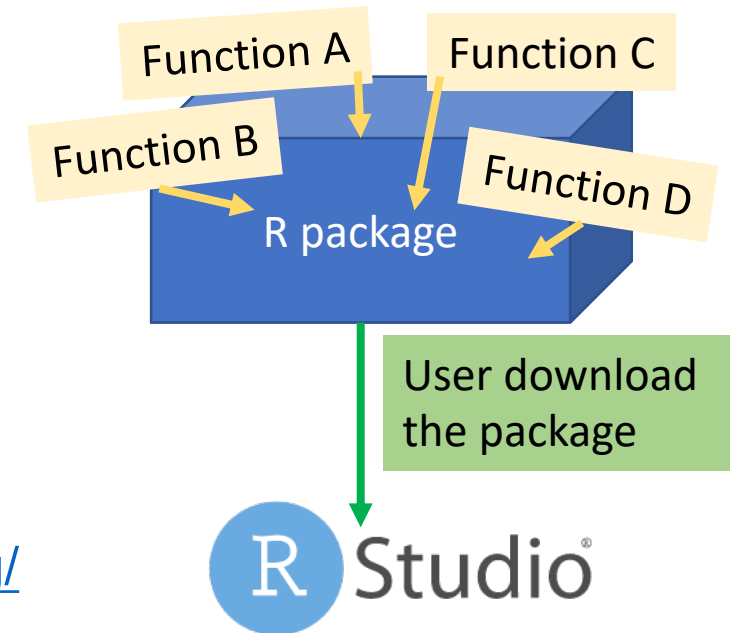
Adjective/Adverb

(“**Options**” are also known as “**arguments**”)

# R is a (computer) language

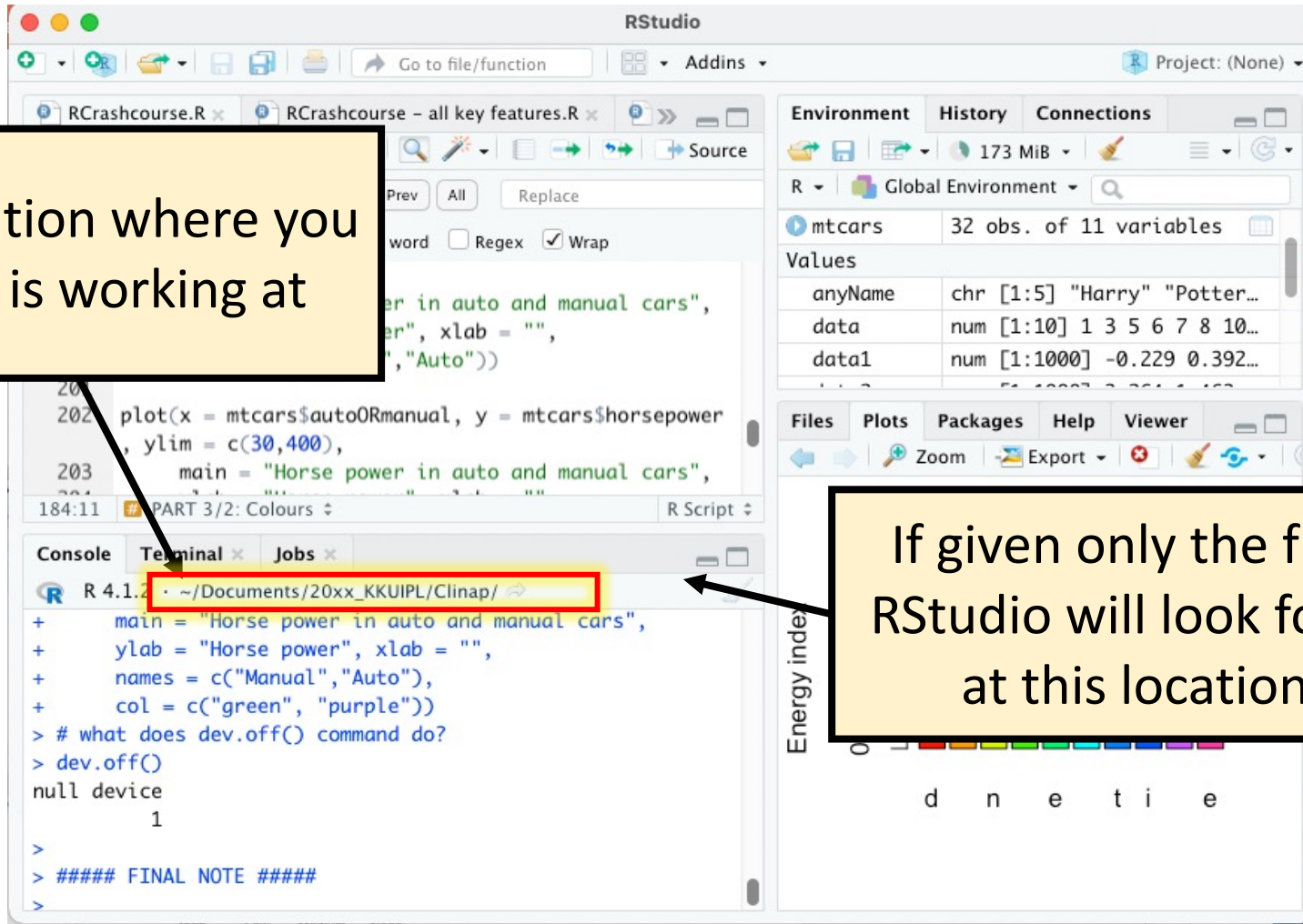
- Languages have verbs

- R comes with many built-in functions, but these might not be enough for some work, or will require writing a very long code.
- Nice people (i.e. developers and programmers) have created many useful **packages** that contain add-on **functions** for R.
- Most of the packages for bioinformatics work can be downloaded from
  - <https://cran.r-project.org/> or <https://www.bioconductor.org/>
- Once downloaded, you would run command **library(name of your package)** every time you restart RStudio



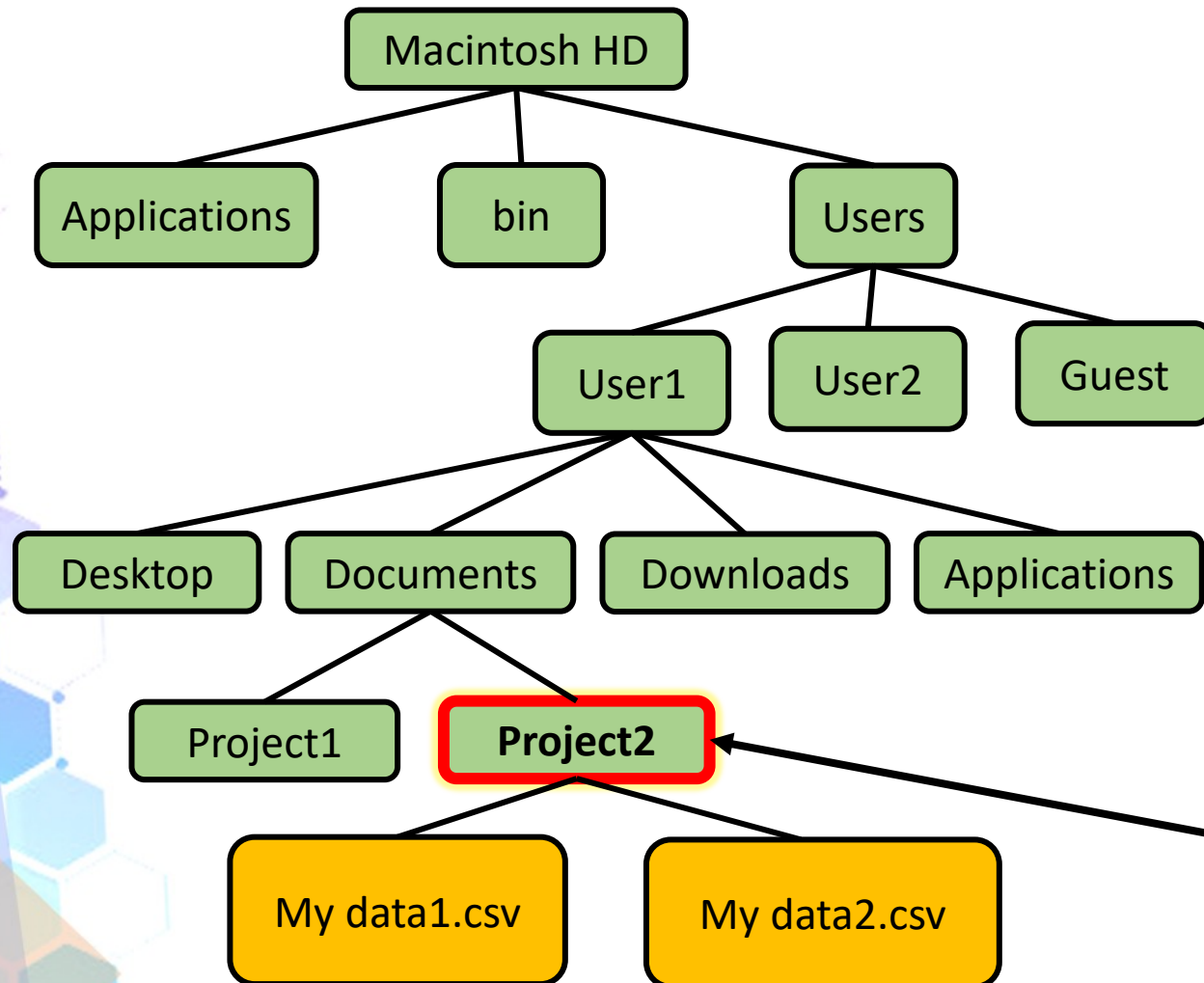
# Make sure Rstudio can find your data

This is the location where you and RStudio is working at



If given only the file name, RStudio will look for that file at this location ONLY

# File address – full path



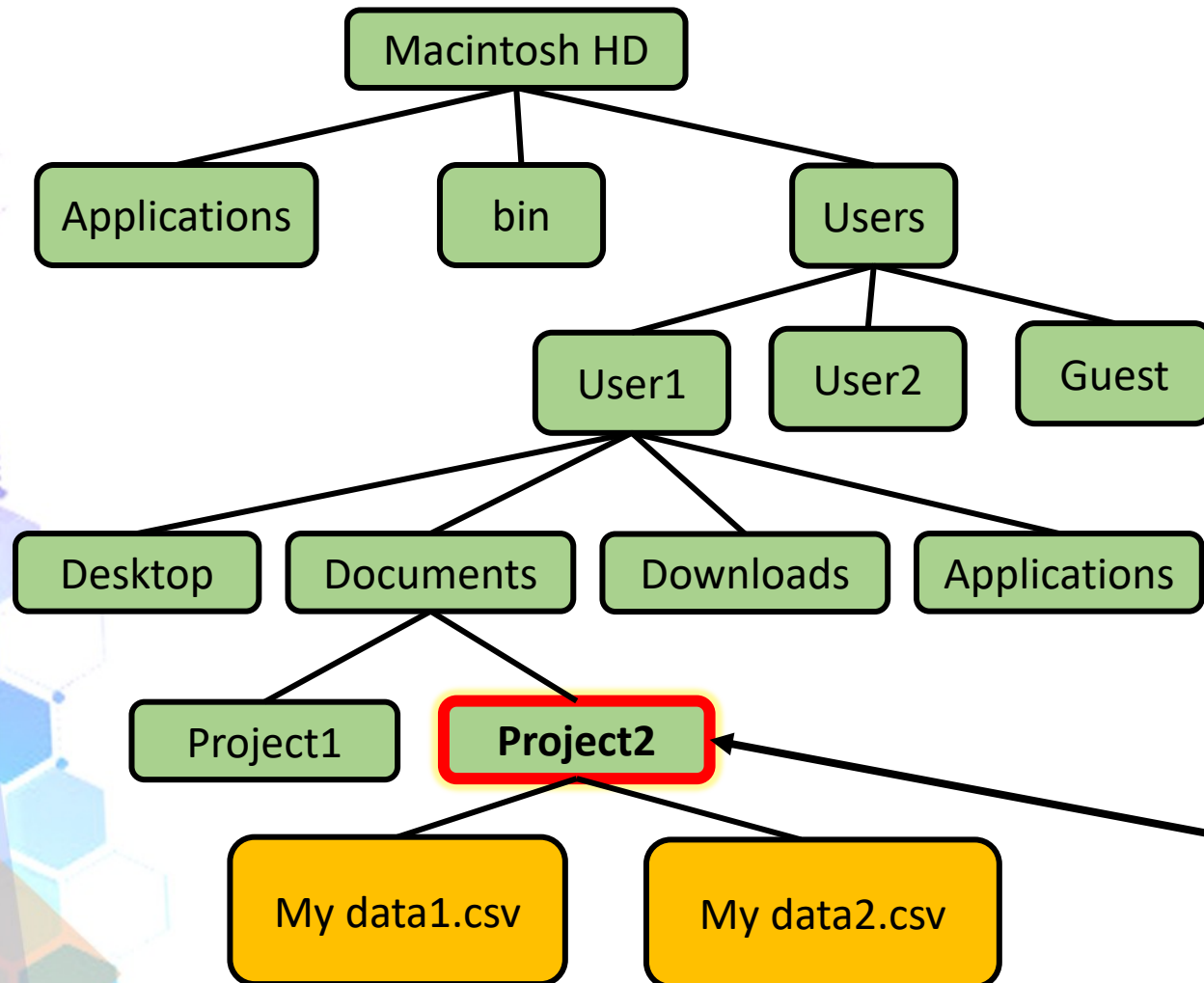
 Folder

 File

A **full path** to **Project2** folder would be  
(address)

**/Macintosh HD/Users/User1/  
Documents/Project2/**

# File address – full path



 Folder

 File

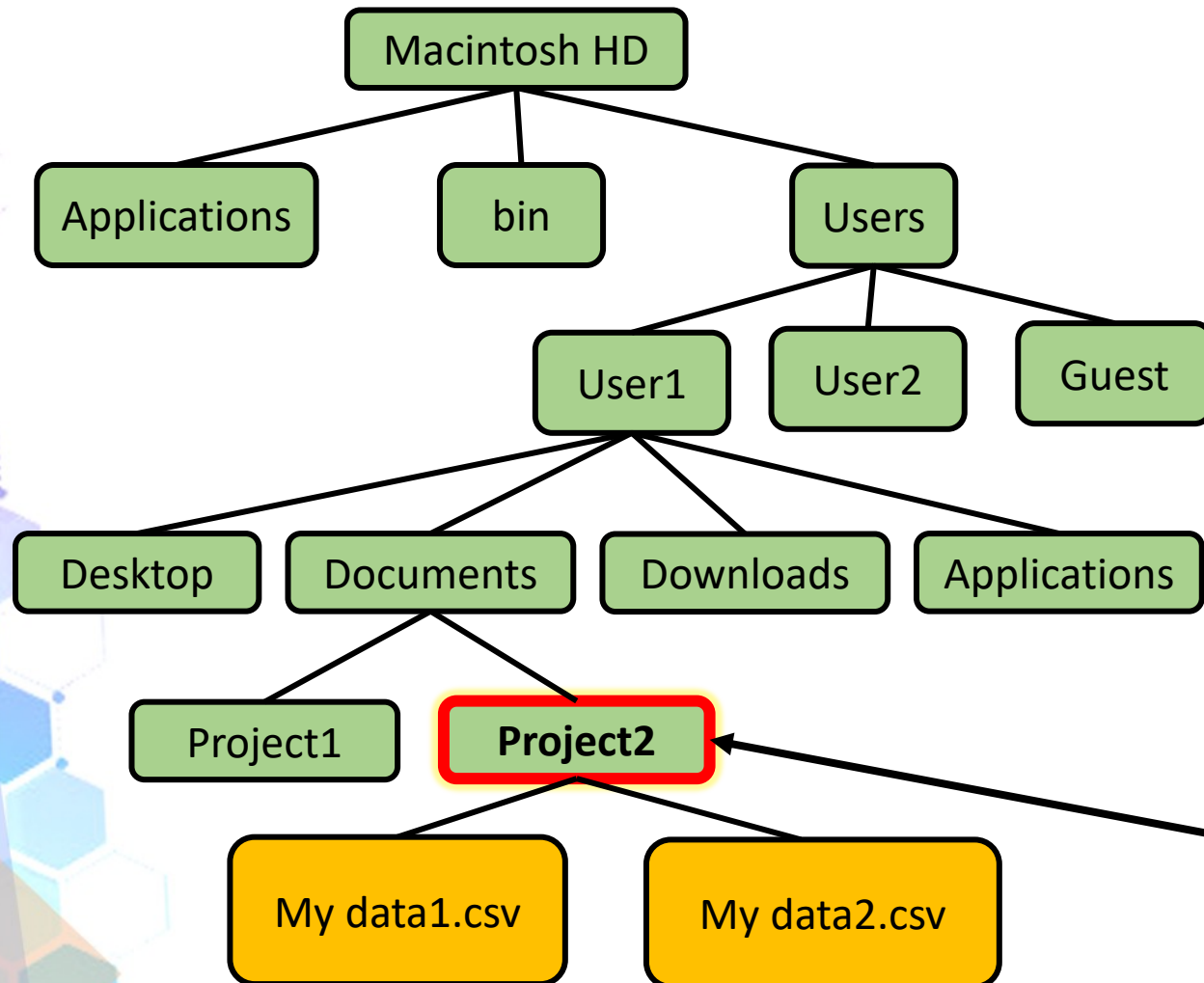
A **full path** to **Project2** folder would be (address)

**/Macintosh HD/Users/User1/  
Documents/Project2/**

Or

**~/Documents/Project2/**

# File address – full path



 Folder

 File

- This part can be replace with ~
- ~ means your HOME ADDRESS

A **full path** to **Project2** folder would be (address)

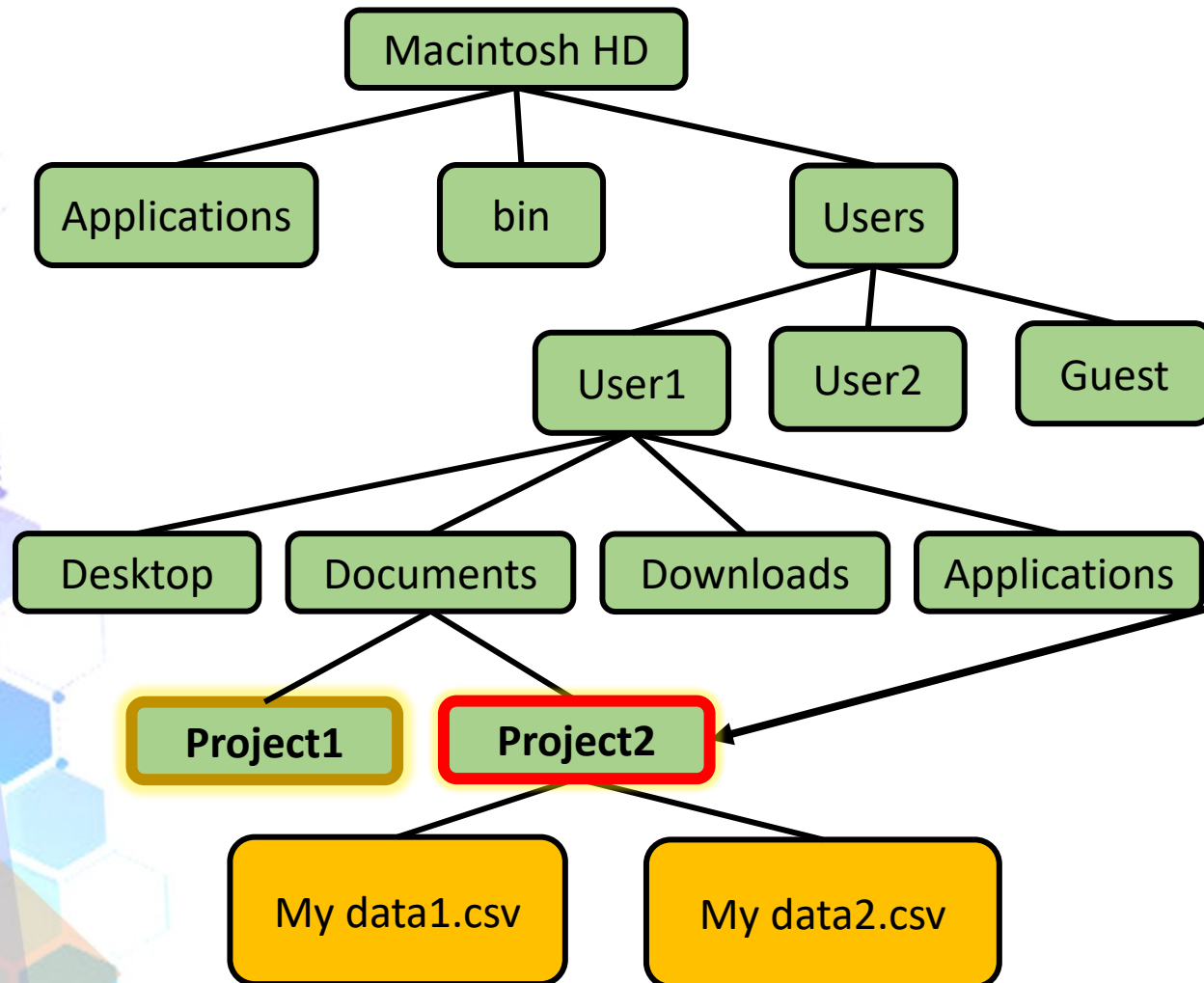
**/Macintosh HD/Users/User1/ Documents/Project2/**

Or

**~/Documents/Project2/**



# File address – relative path



 Folder

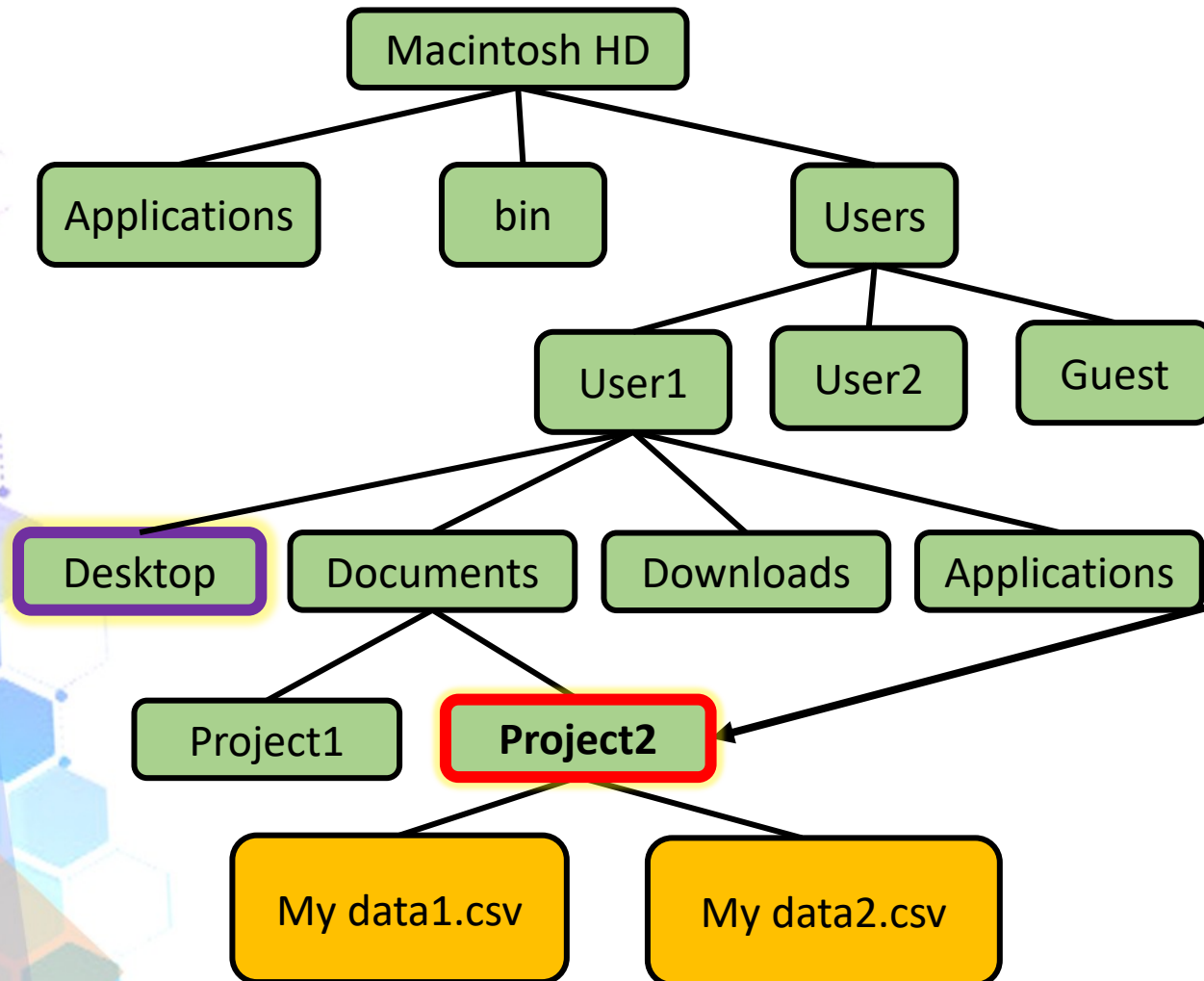
 File

Let's say you are inside **Project1** folder, your **relative path** to **Project2** would be

**../Project2/**

**../** means going up 1 step

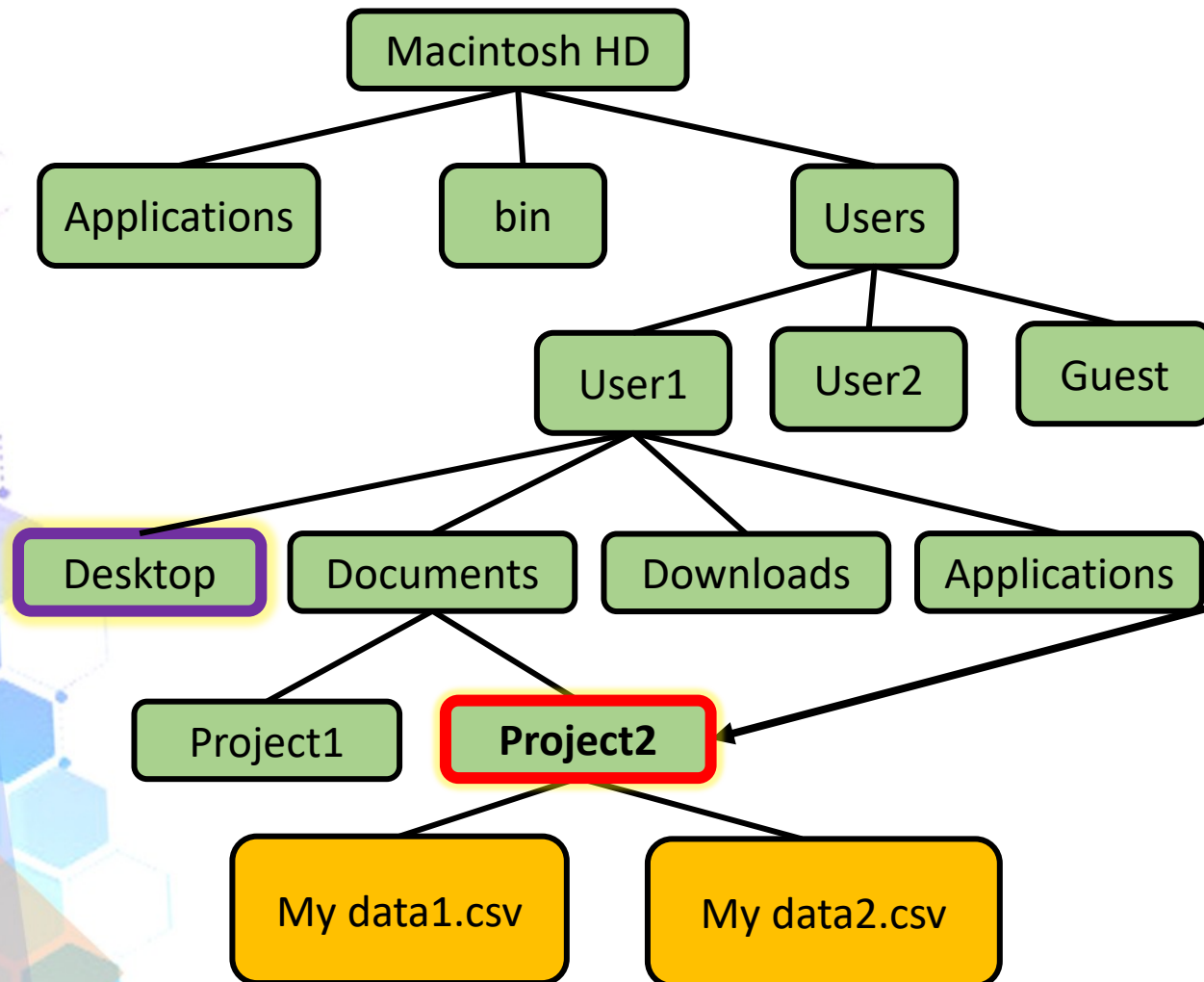
# File address – relative path



Say, you are inside **Project2** folder, what would be your **relative path** to **Desktop**?

???

# File address – relative path



 Folder

 File

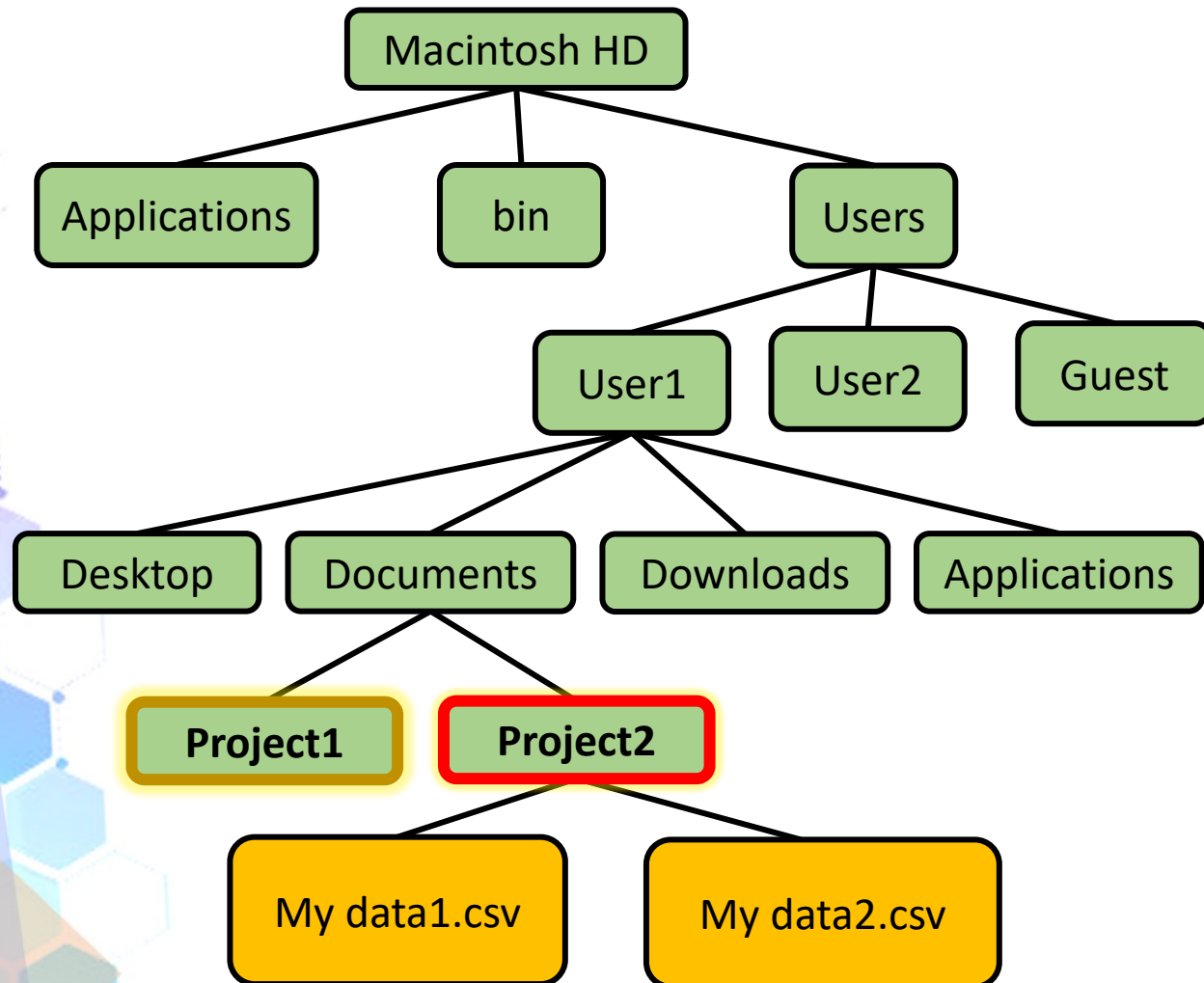
Say, you are inside **Project2** folder, what would be your **relative path** to **Desktop**?

**../.. / Desktop**

Up to **Documents**, up to **User1**, then to **Desktop**

\*\*When in doubt, always use full path\*\*

# Make sure Rstudio can find your data



Say, you are inside **Project1**, but you want to use the file **My data1.csv** which is inside **Project2**

```
read.csv("../Project2/My data1.csv")
```

Or

```
read.csv("/Macintosh  
HD/  
_____/My data1.csv")
```