

Read the error messages!

could not find function “%>%”

This is a classic error that can happen with %>% or with any other function

Two main cases for this error:

CASE 1 :

you misspelled the function's name:

```
rmom(1) # instead rnorm(1)
## Error in rmom(1): could not find function "rmom"
```

Correction :

How can you fix it? By correcting the misspelling.

```
rnorm(1)
```

CASE 2 :

Or (in most case) you forgot to load the package that contains the fonction .

```
iris %>% select(Species)
## Error in iris %>% select(Species) :
## could not find function "%>%"
```

Correction :

Don't forget to install and **launch** both `library(tidyverse)` and others `library(whatever)` at the top of your scripts and Rmd.

```
library(dplyr)
iris %>% select(Species)
```

Error: unexpected SPECIAL in “%>%”

```
library(dplyr)
iris
%>% select(Species)
## Error: unexpected SPECIAL in %>%
## 2: iris
## 3: %>%
##      ^
```

You misplaced a line break.

Correction :

%>% should never be at the start of the code line, you'll need a line break *after* and not *before* the %>%.

```
library(dplyr)
```

```
iris %>%
  select(Species)
```

Error: unexpected 'else' in "else"

```
if ( 1 != 1 ) {print("youpi")}
else { print("coucou")}
## Error: unexpected 'else' in "else"
## 1: if ( 1 != 1 ) {print("youpi")}
## 2: else
##      ^
```

Same cause as last issue.
You misplaced a line break.

Correction :

You need to check that the `else` is correctly associated with the former line.
R interprets the code one line at a time, and when `if` doesn't have an associated `else`, which makes sense, make sure that R understands that your instruction is finished.

```
if ( 1 != 1 ) {print("youpi")}
  } else { print("coucou")}
```

object 'dataset' not found

```
library(dplyr)
dataset %>% select(Species)
## Error in eval(lhs, parent, parent) : object 'dataset' not found
```

The object you're working on doesn't exist.

Correction :

Misspelling? Or make sure you launched and assigned before.

there is no package called 'diplry'

```
library(diplry)
## Error in library(diplry): there is no package called 'diplry'
```

Happens in 2 cases:

CASE 1 :

You misspelled the library name

Correction :

```
library(dplyr)
```

CASZ 2 :

The package hasn't be installed

Correction :

Launch `install.package("dplyr")` in the terminal will fix this.

non-numeric argument to binary operator

```
df + 1
## Error in df + 1 : non-numeric argument to binary operator
```

This one can be pretty confusing.

Correction :

Make sure your arithmetic operation make sense, and that it's on the right object. Make sure this object contains what you expected it to contain.

In the example above, `df` is a function, and not a number, causing the error.

Problem with `filter()` input `..1`.

```
library(dplyr)
iris %>% filter(Species = "setosa")
## Error: Problem with `filter()` input `..1`.
## x Input `..1` is named.
## i This usually means that you've used `=` instead of `==`.
## i Did you mean `Species == "setosa"`?
```

Le message d'erreur contient la solution,

Correction :

The correction is in the error message. You put only one `=`, but a condition shall be written with `==`.

```
library(dplyr)
iris %>% filter(Species == "setosa")
```

Error: unexpected symbol in:

```
library(dplyr)
iris %>% filter(Species == "setosa"
iris %>% filter(Species == "setosa")
## Error: :3:1: unexpected symbol
## 2: iris %>% filter(Species == "setosa"
## 3: iris
##      ^
```

This error message usually occurs when you send an unfinished line, where you forgot a closing bracket, accolade or quotation mark. In this particular case, R waits till the end of your instruction. but instead of only sending the missing bracket (or accolade/quotation mark), you are sending the whole instruction again. Even trickier in the Rmd as you don't think to look at the terminal to check what's been indeed sent to R.

Correction :

It's already fixed! Now that R sent you an error message, relaunch your instruction, it'll work.

```
library(dplyr)
```

```
iris %>% filter(Species == "setosa")
)
```

**Operation not allowed without an active reactive context.
(You tried to do something that can only be done from
inside a reactive expression or observer.)**

See below

**Can't access reactive value 'go' outside of reactive
consumer. Do you need to wrap inside reactive() or
observer()?**

```
library(shiny)
ui <- fluidPage(
  actionButton("go", "go")
)
server <- function(input, output, session) {
  res <- input$go
}
shinyApp(ui, server)
```

You use shiny and you tried using `input$` outside `observe()`, `observeEvent()`, `reactive()`, `render()`. It's just not possible. Don't worry, literally 100% of the people before you did the mistake! 😊

Correction :

You can put an `observe()` around your code.

But please! Both `observe()` and `reactive()` shouldn't be used in your shiny apps (we we talk about this one day..), so try and look for a better solution...:)

```
library(shiny)
ui <- fluidPage(
  actionButton("go", "go")
)
server <- function(input, output, session) {
  r <- reactiveValues(x=NULL)
  observeEvent( TRUE ,once= TRUE, {
    r$x <- input$go
  })
}
shinyApp(ui, server)
```

Any other annoying recurring mistake to share? Please write below in comments!