# Directory interactions

# Tương tác với thư mục làm việc

In this page we cover common scenarios where you create, interact with, save, and import with directories (folders).

Trong chương này, chúng tôi sẽ bao gồm các tình huống phổ biến, nơi chúng ta sẽ tạo, tương tác, lưu và nhập với thư mục

## Preparation

## Chuẩn bị

### fs package

### Gói lệnh fs

The **fs** package is a **tidyverse** package that facilitate directory interactions, improving on some of the **base** R functions. In the sections below we will often use functions from **fs**.

Gói lệnh **fs** là một gói lệnh **tidyverse** làm thuận tiện cho việc tương tác với thư mục, cải thiện một số hàm trong **base R.** Trong các phần bên dưới, chúng ta sẽ dùng các hàm từ **fs.**

### Print directory as a dendrogram tree

### In thư mục dưới dạng cây biểu đồ dendrogram

Use the function dir\_tree() from **fs**.

Provide the folder filepath to path = and decide whether you want to show only one level (recurse = FALSE) or all files in all sub-levels (recurse = TRUE). Below we use here() as shorthand for the R project and specify its sub-folder “data”, which contains all the data used for this R handbook. We set it to display all files within “data” and its sub-folders (e.g. “cache”, “epidemic models”, “population”, “shp”, and “weather”).

Sử dụng hàm dir\_tree() từ gói lệnh **fs**.

Cung cấp đường dẫn tập tin, thư mục cho đối số path = và quyết định xem có hay không chúng ta muốn hiển thị chỉ ở một cấp độ (recurse = FALSE) hay tất cả các tập tin trong tất cả các thư mục con (recurse = TRUE). Bên dưới chúng ta sử dụng hàm here() để hiển thị kết quả dưới dạng viết tốc ký trong R và cụ thể “số liệu” của các thư mục con mà chứa tất các số liệu được sử dụng cho cuốn sổ tay này. Chúng ta đặt nó để hiển thị tất cả tập tin trong “số liệu” và các thư mục con (ví dụ: “bộ nhớ cache”, “mô hình dịch bênh”, “dân số”, “shp”, và “thời tiết”).

## List files in a directory

## Liệt kê các tập tin trong một thư mục

To list just the file names in a directory you can use dir() from **base** R. For example, this command lists the file names of the files in the “population” subfolder of the “data” folder in an R project. The relative filepath is provided using here() (which you can read about more in the [Import and export](#bookmark=id.2et92p0) page).

To list the full file paths of the directory’s files, you can use you can use dir\_ls() from **fs**. A **base** R alternative is list.files().

To get all the metadata information about each file in a directory, (e.g. path, modification date, etc.) you can use dir\_info() from **fs**.

This can be particularly useful if you want to extract the last modification time of the file, for example if you want to import the most recent version of a file. For an example of this, see the [Import and export](#bookmark=id.2et92p0) page.

Here is the data frame returned. Scroll to the right to see all the columns.

Để chỉ liệt kê các tên tập tin trong một thư mục, chúng ta có thể sử dụng hàm dir() trong **base** R. Ví dụ, lệnh này liệt kê tên các tập tin của các tập tin trong thư mục con tên “dân số / population” của thư mục “số liệu data” trong một dự án R. Đường dẫn tập tin liên quan được cung cấp bằng cách sử dụng hàm here() (đọc thêm trong chương [Nhập và xuất số liệu](#bookmark=id.2et92p0)).

Để liệt kê đường dẫn đầy đủ của các tập tin trong thư mục, chúng ta có thể sử dụng hàm dir\_ls() từ gói lệnh **fs**. Một cách dùng khác là sử dụng hàm list.files()trong **base** R.

Để lấy tất cả thông tin chi tiết của mỗi tập tin trong một thư mục (ví dụ: đường dẫn, ngày sửa đổi, .v.v.), chúng ta có thể sử dụng hàm dir\_info() từ gói lệnh **fs**.

Điều này có thể đặc biệt hữu ích nếu chúng ta muốn trích xuất thời gian sửa đổi sau cùng của tập tin, ví dụ như nếu chúng ta muốn nhập phiên bản mới nhất của tập tin. Ví dụ cho việc này, xem trong chương [Nhập và xuất số liệu](#bookmark=id.2et92p0).

Đây là khung số liệu được xuất ra. Cuộn qua phải để xem tất cả các cột.

## File information

## Thông tin của tập tin

To extract metadata information about a specific file, you can use file\_info() from **fs** (or file.info() from **base** R).

Here we use the $ to index the result and return only the modification\_time value.

Để trích xuất thông tin chi tiết của một tập tin cụ thể, chúng ta có thể sử dụng hàm file\_info() từ gói lệnh **fs** (hoặc hàm file.info() trong **base** R).

Ở đây, chúng ta sử dụng ký hiệu $ để chỉ rõ kết quả và xuất chỉ mỗi giá trị về thời gian sửa đổi (modification\_time).

## Check if exists

## Kiểm tra xem có tồn tại không

### R objects

### Đối tượng R

You can use exists() from **base** R to check whether an R object exists *within* R (supply the object name in quotes).

Note that some **base** R packages use generic object names like “data” behind the scenes, that will appear as TRUE unless inherit = FALSE is specified. This is one reason to not name your dataset “data”.

If you are writing a function, you should use missing() from **base** R to check if an argument is present or not, instead of exists().

Chúng ta có thể sử dụng hàm exists() trong **base** R để kiểm tra xem một đối tượng R có tồn tại bên trong môi trường R hay không (cung cấp tên đối tượng trong dấu ngoặc kép).

Lưu ý rằng một số gói lệnh trong **base** R sử dụng các tên chung làm tên cho các đối tượng như “data” trong môi trường R, mà kết quả kiểm tra sẽ là TRUE trừ khi chúng ta chỉ rõ inherit = FALSE. Đây là một lý do để không đặt tên cho bộ số liệu của chúng ta là “data”.

Nếu chúng ta đang viết một hàm, chúng ta nên sử dụng hàm missing() trong **base** R để kiểm tra xem có một đối số hiện diện hay không, thay vì dùng hàm exists().

### Directories

### Thư mục

To check whether a directory exists, provide the file path (and file name) to is\_dir() from **fs**. Scroll to the right to see that TRUE is printed.

An alternative is file.exists() from **base** R.

Để kiểm tra xem một thư mục có tồn tại không, cung cấp đường dẫn tập tin (và tên tập tin) đến hàm is\_dir() từ gói lệnh **fs**. Cuộn qua bên phải để thấy rằng TRUE được xuất ra.

Một cách làm khác là dùng hàm file.exists() trong **base** R.

### Files

### Tập tin

To check if a specific file exists, use is\_file() from **fs**. Scroll to the right to see that TRUE is printed.

A **base** R alternative is file.exists().

Để kiểm tra xem một tập tin cụ thể có tồn tại hay không, sử dụng hàm is\_file() trong **fs**. Cuộn qua bên phải để thấy rằng TRUE được xuất ra.

Một các làm khác là dung hàm file.exists() trong **base** R.

## Create

## Tạo mới

### Directories

### Thư mục

To create a new directory (folder) you can use dir\_create() from **fs**. If the directory already exists, it will not be overwritten and no error will be returned.

An alternative is dir.create() from **base** R, which will show an error if the directory already exists. In contrast, dir\_create() in this scenario will be silent.

Để tạo một thư mục mới, chúng ta có thể sử dụng hàm dir\_create() từ gói lệnh **fs**. Nếu thư mục đã tồn tại, thư mục này sẽ không bị ghi đè và không bị trả lại lỗi.

Một cách làm khác là dùng hàm dir.create() trong **base** R, mà sẽ hiển thị lỗi nếu thư mục đó đã tồn tại. Ngược lại, dùng hàm dir\_create() trong tình huống này sẽ là im lặng.

### Files

### Tập tin

You can create an (empty) file with file\_create() from **fs**. If the file already exists, it will not be over-written or changed.

A **base** R alternative is file.create(). But if the file already exists, this option will truncate it. If you use file\_create() the file will be left unchanged.

Chúng ta có thể tạo một tâp tin (rỗng) bằng hàm file\_create() từ gói lệnh **fs**. Nếu tập tin này đã tồn tại, nó sẽ không bị ghi đè hoặc thay đổi.

Một cách làm khác là dùng hàm file.create() trong **base** R, như nếu tập tin này đã tồn tại, tùy chọn này sẽ cắt xén tập tin đó. Nếu dùng hàm file\_create(),tập tin này sẽ được giữ nguyên.

### Create if does not exists

### Tạo nếu không tồn tại

UNDER CONSTRUCTION

ĐANG XÂY DỰNG

## Delete

## Xóa

### R objects

### Đối tượng R

Use rm() from **base** R to remove an R object.

Dùng hàm rm() trong **base** R để loại bỏ một đối tượng R.

### Directories

### Thư mục

Use dir\_delete() from **fs**.

Dùng hàm dir\_delete() từ gói lệnh **fs**.

### Files

### Tập tin

You can delete files with file\_delete() from **fs**.

Chúng ta có thể xóa các tập tin bằng cách dùng hàm file\_delete() từ gói lệnh **fs**.

## Running other files

## Vận hành các tập tin khác

### source()

To run one R script from another R script, you can use the source() command (from **base** R).

This is equivalent to viewing the above R script and clicking the “Source” button in the upper-right of the script. This will execute the script but will do it silently (no output to the R console) unless specifically intended. See the page on [Interactive console] for examples of using source() to interact with a user via the R console in question-and-answer mode.

Để thực hiện một tập lệnh R từ một tập lệnh R khác, chúng ta sử dụng lệnh source() (trong **base** R).

Điều này tương đương với việc xem tập lệnh R bên trên và nhấp vào nút “Nguồn/Source” ở phía trên bên phải của tập lệnh. Điều này sẽ thực hiện tập lệnh này nhưng sẽ thực hiện một cách âm thầm (không có kết quả đầu ra trong bảng điều khiển R) trừ khi có chủ ý cụ thể. Xem các ví dụ ở trang [Bảng điều khiển tương tác] về cách sử dụng source() để tương tác với người dùng qua bảng điều khiên R ở chế độ hỏi và trả lời.

### render()

render() is a variation on source() most often used for R markdown scripts. You provide the input = which is the R markdown file, and also the output\_format = (typically either “html\_document”, “pdf\_document”, “word\_document”, "")

See the page on [Reports with R Markdown](#bookmark=id.1pxezwc) for more details. Also see the documentation for render() [here](https://rmarkdown.rstudio.com/docs/reference/render.html) or by entering ?render.

Hàm render() là một biến thể của hàm source() mà thường được sử dụng nhất cho các tập lệnh của R markdown. Chúng ta cung cấp input = tập tin R markdown, và cũng vậy output\_format = (thường là “html\_document”, “pdf\_document”, “word\_document”, "")

Xem chương [Báo cáo với R Markdown](#bookmark=id.1pxezwc) để biết thêm chi tiết. Cũng vậy, xem tài liệu về render() [ở đây](https://rmarkdown.rstudio.com/docs/reference/render.html) hoặc bằng cách gõ ?render.

### Run files in a directory

### Vận hành các tập tin trong một thư mục

You can create a *for loop* and use it to source() every file in a directory, as identified with dir().

If you only want to run certain scripts, you can identify them by name like this:

Here is a [comparison](https://cran.r-project.org/web/packages/fs/vignettes/function-comparisons.html) of the **fs** and **base** R functions.

Chúng ta có thể tạo một *vòng lặp* và đưa nó vào hàm source() mỗi tập tin trong một thư mục, như được nhận dạng với hàm dir().

Nếu chúng ta chỉ muốn thực hiện một vài tập lệnh, chúng ta có thể xác định chúng bằng tên như sau:

Đây là một [so sánh](https://cran.r-project.org/web/packages/fs/vignettes/function-comparisons.html) giữa các hàm của **fs** và **base** R.

### Import files in a directory

### Nhập tập tin vào một thư mục

See the page on [Import and export](#bookmark=id.2et92p0) for importing and exporting individual files.

Also see the [Import and export](#bookmark=id.2et92p0) page for methods to automatically import the most recent file, based on a date in the file name *or* by looking at the file meta-data.

See the page on [Iteration, loops, and lists](#bookmark=id.4f1mdlm) for an example with the package **purrr** demonstrating:

* Splitting a data frame and saving it out as multiple CSV files
* Splitting a data frame and saving each part as a separate sheet within one Excel workbook
* Importing multiple CSV files and combining them into one dataframe
* Importing an Excel workbook with multiple sheets and combining them into one dataframe

Xem chương [Nhập và xuất số liệu](#bookmark=id.2et92p0) để nhập và xuất các tập tin riêng lẻ.

Cũng vậy, xem chương [Nhập và xuất số liệu](#bookmark=id.2et92p0) để biết các phương pháp nhập một cách tự động các tập tin gần đây nhất dựa trên ngày trong tên tập tin hoặc bằng cách xem các thông tin chi tiết của tập tin.

Xem một ví dụ trong chương [Lặp, vòng lặp, và danh sách](#bookmark=id.4f1mdlm) để thấy rõ cách sử dụng gói lệnh **purr**:

* Để tách một khung số liệu và lưu nó ra dưới dạng nhiều tập tin CSV
* Tách khung số liệu và lưu từng phần vào một trang riêng biệt trong một tập tin Excel
* Nhập nhiều tập tin CSV và kết hợp chúng thành một khung số liệu
* Nhập một tập tin Excel có nhiều trang riêng biệt và kết hợp chúng thành một khung số liệu

## base R

See below the functions list.files() and dir(), which perform the same operation of listing files within a specified directory. You can specify ignore.case = or a specific pattern to look for.

If a file is currently “open”, it will display in your folder with a tilde in front, like “~$hospital\_linelists.xlsx”.

Xem bên dưới các hàm list.files() và dir() thực hiện cùng một thao tác liệt kê các tập tin trong một thư mục cụ thể. Chúng ta có thể cụ thể cho đối số ignore.case = hoặc cho đối số pattern.

Nếu một tập tin hiện tại đang “mở”, nó sẽ hiện thị trong thư mục với dấu ngã phía trước, như là “~$hospital\_linelists.xlsx”.

## Resources

## Nguồn

<https://cran.r-project.org/web/packages/fs/vignettes/function-comparisons.html>