

# Loading Data into R

Loading data into R requires R to interact with the file system of your computer. Your file system is a collection of folders that contain documents. For example, you may have a folder titled “Introduction to Biological Statistics” that contains the lecture slides and R scripts for this course. The folder may be on your Desktop, or in some other folder on your computer. Folders can also be called directories.

Your working directory is the folder in which R will look for files. When you ask R to load a data file, you need to tell R where to look. The manner in which you do this depends on your operating system (Windows/OSX/Linux), your particular software implementation of the R software (R/R Gui/R Studio) and your own personal preferences.

Below is a guide to loading data into R using several common methods. The examples use the `read.csv()` function, but you can replace that function with `read.table()` if your data are not in .csv format.

## Before we start, a note on naming conventions

File paths and file names that contain spaces are problematic. R, like other scripting languages, does not know how to interpret the white space between characters when spaces are present. Enclosing your file names and paths in quotes circumvents this problem, so you do not need to rename files. However, as a best practice, it is recommended that you start naming files and folders **without** spaces. This holds for all operating systems.

## Windows

### Software Independent Methods

#### Set your working directory to the directory containing your data.

##### 1. Check your current working directory

The following command, executed without arguments, will print the current working directory to screen.

```
> getwd()
```

##### 2. Change your working directory

If your current working directory is not the directory in which your data are saved, you’ll need to change your working directory. Specify your file path relative to your local disk. This is typically your C drive. Use either single forward slashes or double back slashes to separate directory names.

```
> setwd("C:/path/to/directory")
```

or

```
> setwd("C:\\path\\to\\directory")
```

### What if you don't know the file path?

Navigate to your data file using File Explorer. Right-click on the file and select the "Properties" option. This will bring up a window with several tabs containing information about your file. Find the tab named "General" and look for the line labeled "Location". This is your file path. You can copy and paste it into your R script, but remember to make sure you have the appropriate number and direction of slashes. You'll find more information below about properly formatting the file path for R.

## 3. Load your data

Pass your file name to the **read.csv()** function. R will search your working directory for that file, and load it into your workspace under the variable name you specify (in this example, "data"). The file name should be in quotes.

```
> data <- read.csv("file.csv")
```

### If you don't want to change your working directory...

You have a couple of options.

1. You can specify the full file name in the **read.csv()** function. You can use either double backslashes ("\\") or single forward slashes ("/"), depending on your preference. Only the double backslash option is shown below.

```
> data <- read.csv("C:\\path\\to\\directory\\file.csv")
```

2. You can open up a file browser window and select your data file using the following command.

```
> data <- read.csv(file.choose())
```

## Software Dependent Methods

The methods below are additional options available to users of these specific R interfaces.

### R Gui

#### Set your working directory to the directory containing your data.

R installs in Windows as R Gui, which has a few point-and-click options. You can change your working directory by navigating the R Gui menus.

File > Change Dir opens up a file browser. Select the directory that contains your data file. Click OK. Then load data using the **read.csv()** function.

```
> data <- read.csv("file.csv")
```

## R Studio

### Set your working directory to the directory containing your data.

**Point-and-Click** Use the Session menu at the top of the screen. Session > Set Working Directory > Choose Directory will open a file browser for you to select your working directory. Click “OK” when you’ve found the right one.

**Keyboard Shortcut** Ctl + Shift + K will also open a file browser for you to select your working directory. Click “OK” when you’ve found the right one.

### If you don’t want to change your working directory...

You can use the “Import Dataset” button. You’ll find it in the “Workspace” tab of R Studio. Click on “Import Dataset” and select “From Text File”. This will open up a file browser. Select the directory that contains your data file and click “Open”. A screen titled “Import Dataset” will appear. You can preview your dataset here, and assign the dataset a variable name. Click “Import”.

The “Tools” menu at the top of the screen also has an “Import Dataset” option. It works exactly as described above.

# OSX

## Software Independent Methods

### Set your working directory to the directory containing your data.

#### 1. Check your current working directory

The following command, executed without arguments, will print the current working directory to screen.

```
> getwd()
```

#### 2. Change your working directory

If your current working directory is not the directory in which your data are saved, you’ll need to change your working directory. Specify your file path relative to the Users directory.

```
> setwd("~/Users/Account Name/path/to/directory")
```

#### What if you don’t know the file path?

Navigate to your data file using Finder. Right click on the file and select the “Get Info” option. This will bring up a window that contains information about your file. Look for the line labeled “Where”. This is your file path.

Copy and paste it into your R script, but remember to make sure you have the appropriate direction of slashes. You'll find more information below about properly formatting the file path for R.

### 3. Load your data

Pass your file name to the **read.csv()** function. R will search your working directory for that file, and load it into your workspace under the variable name you specify (in this example, "data"). The file name should be in quotes.

```
> data <- read.csv("file.csv")
```

### If you don't want to change your working directory...

You have a couple of options.

1. You can specify the full file name in the **read.csv()** function.

```
> data <- read.csv("/Users/Account Name/path/to/directory/file.csv")
```

2. You can open up a file browser window and select your data file using the following command.

```
> data <- read.csv(file.choose())
```

## Software Dependent Methods

The methods below are additional options available to users of R and R Studio.

### R

**Point-and-Click** Use the Misc menu at the top of the screen. Misc > Change Working Directory will open a file browser for you to select your working directory. Click "OK" when you've found the right one.

**Keyboard Shortcut** Cmd + D will also open a file browser for you to select your working directory. Click "OK" when you've found the right one.

### R Studio

#### Set your working directory to the directory containing your data.

**Point-and-Click** Use the Tools menu at the top of the screen. Tools > Change Working Dir > Choose Directory will open a file browser for you to select your working directory. Click "OK" when you've found the right one.

**Keyboard Shortcut** Ctl + Shift + D will also open a file browser for you to select your working directory. Click "OK" when you've found the right one.

#### If you don't want to change your working directory...

You can use the "Import Dataset" button. You'll find it in the "Workspace" tab of R Studio. Click on "Import Dataset" and select "From Text File". This will open up a file browser. Select the directory that contains your

data file and click "Open". A screen titled "Import Dataset" will appear. You can preview your dataset here, and assign the dataset a variable name. Click "Import".