

What are the possible directories I might have in my R package?

Required

- **R:** contains the source code for your functions
- **man:** contains the .Rd files documenting your functions

Optional

- **vignettes:** Contains the R Markdown (.Rmd) or Sweave (.Rnw) sources for any vignettes (AKA tutorials). Often contains the compiled HTML or PDF versions as well.
- **data:** Contains datasets saved as R environments (.RData). A user will be able to load these using the `data` function. Any datasets in here also need to be documented in their own .Rd files; see <http://mirror.las.iastate.edu/CRAN/doc/manuals/r-release/R-exts.html#Documenting-data-sets>.
- **inst:** Anything that you put in here will be in the package directory once the package is installed on your computer.
- **inst/doc:** (That is, inside the inst folder, another folder called doc.) If you have vignettes, a copy will automatically go here when the package is built. You can put any other type of documentation in here, for example an MS Word document.
- **inst/extdata:** If you have datasets that you want to install with the R package that aren't saved as .RData files, you can put them here. Plain text, csv, etc. You can also put scripts and whatever else in here.

Do I need a vignette for the final project?

Only if the help files (.Rd files) by themselves aren't sufficient to explain how to use the package.

How do I indicate that my package depends on another package?

In your DESCRIPTION file, you should have a line that starts with "Imports:" and lists any packages that your package needs in order to run. This used to be "Depends:", hence if you look in "Writing R Extensions" you'll see some examples that make it look like you should use "Depends:". But generally "Imports:" is what you want. If there's a package that your package uses but in a more optional way, you can list it under "Suggests:".

Anyway, in summary:

- Imports if you just need a few functions from a package
- Depends if you need all the functions from a package, and want to have the whole package namespace loaded when your source code is interpreted
- Suggests if you only need functions from a package under certain circumstances

For packages listed in Imports, you'll have to do one of two things:

In your R code, if you use a function from another package, you'll have to also give the package name since the whole package has not been loaded into the namespace. For example, if I want to use the `nj` function from the package `ape`, I should write it as `ape::nj(x)` instead of just `nj(x)`. (This is also what you should do for packages listed in Suggests.)

OR

In your `NAMESPACE` file, add some `importFrom` statements. The first argument is the package name and then any arguments after that are functions that you want to import from that package. For example, if I wanted to use `nj` and `axisPhylo` from the package `ape`, I would add the line `importFrom("ape", "nj", "axisPhylo")`.

There are a lot of functions in the `stats`, `utils`, `graphics`, and `grDevices` packages that we take for granted as always being available when we use R. However, these packages are not loaded into your package namespace by default. When you run your package check, you may get a message suggesting additional `importFrom` lines that you should add to your `NAMESPACE` file.

How do I include a dataset with my package?

As you may have noticed in the description of package directories, there are two main ways to do it.

If you want to have the dataset available already as an R object, you can save an image of it to an `.RData` file (using the `save` function) and put it in the “data” directory in your package. For this type of dataset, you should write an `.Rd` file to document it. Then if you want to use it in your `.Rd` examples or your vignettes, you can just include the line `data(mydataset)` (where `mydataset` is the name of the object) and the object will be loaded into the environment.

Perhaps you want to have the dataset in a file instead, because part of your tutorial is going to show how to read the file. In this case you want to put that file in the `inst/extdata` directory. Then when the package is installed, it will have a folder called `extdata`, which will contain your dataset. To access the file from your `.Rd` example sections or vignettes, include the line

```
mypath <- system.file("extdata", "myfile.txt", package = "mypackage")
```

(Where `myfile.txt` is the name of the file, and `mypackage` is the name of your package.) This will produce a full path to the file wherever it is stored on the computer after package installation. You can then run

```
mydata <- read.table(mypath)
```

etc.

As a third option, if you have data files that are only used by vignettes, you can put them in the “vignettes” directory. The vignettes directory will be the working directory when the vignette is compiled.

Should I make a Sweave or R Markdown vignette?

It is up to you, depending on your preference for how the output looks. Sweave is supported by the base R installation.

For R Markdown, you will have to 1) have knitr installed, 2) add a line to the DESCRIPTION file, `VignetteBuilder: knitr`, 3) add knitr to your Imports line in the DESCRIPTION, and 4) add the line `%\VignetteEngine{knitr::knitr}` to the beginning of your R Markdown document. See <http://mirror.las.iastate.edu/CRAN/doc/manuals/r-release/R-exts.html#Non-002dSweave-vignettes>.