

# Preliminaries in R

---

## The “package” system

---

# R packages

Your original download of R is only a starting point:

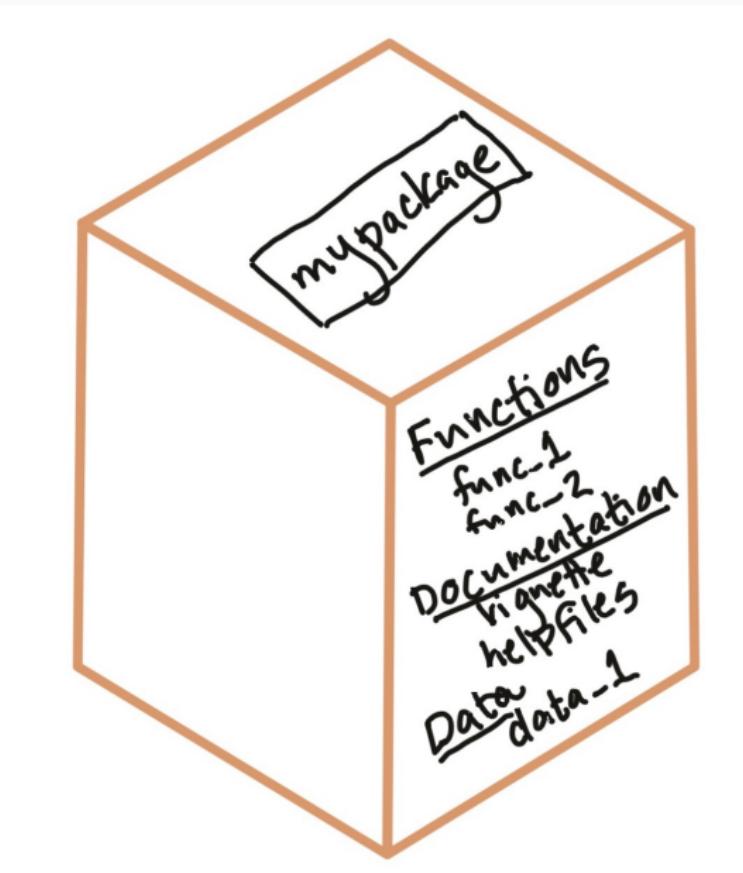


# R packages

To take full advantage of R, you'll want to add on packages:



# What's in a package?



# R packages

You can get packages to add-on to your version of R from:

- CRAN
- Bioconductor (specifically for bioinformatics-related packages)
- GitHub
- Your friends and collaborators
- Make them yourself

The Comprehensive R Archive Network (CRAN) is the primary source of R packages, with thousands of available packages. Each of the packages available on CRAN has a unique name and has passed some broad checks.



**Dirk Eddelbuettel** @eddelbuettel · 27 Jan 2017

Big congratulations to @gbwanderson whose new package 'hurricaneexposure' just became package 10,000 on CRAN !!

**CRAN Package Updates** @CRANberriesFeed

9999 packages on CRAN right now, so imagine dozens of R nerds hanging in suspense waiting for the package to make it 10k ...

2

35

93



# Installing from CRAN

The main way you will install new packages will be with the `install.packages` function. By default, this installs packages from CRAN.

For example, to get the package “phonenumbers”, you could use:

```
install.packages(pkgs = "phonenumbers")
```



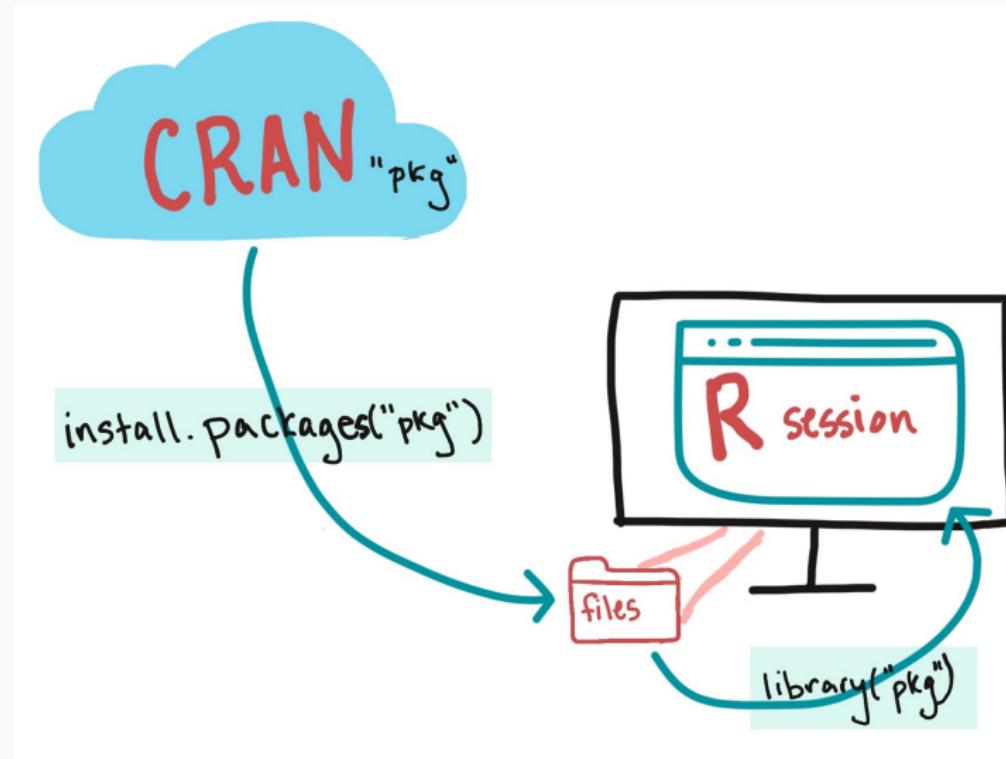
# Installing from CRAN

**Installing** a package downloads it (usually from CRAN) to your computer.

You will need to be online to install a package, because it downloads the package from an online repository.

Once you install a package, you do not need to install it again. It's on your computer. (At least, until the package maintainer creates a new version.)

## Installing versus loading



**Install** a package (with `install.packages()`) to get it onto your computer. **Load** it (with `library()`) to get it into your R session.

## How to use a package you've installed

1. Open an R session
2. Load the package
3. Use the package's vignette and helpfiles to figure out how to use the package (optional)
4. Use functions in the package

## Loading an installed package

Once you have a package, you can load it to an R session using the `library()` function.

```
library(package = "phonenumbers")
```

Once it's loaded, you can use all its functions.

```
message <- "HelloWorld"  
letterToNumber(value = message)
```

```
## [1] "4355696753"
```

## Package vignettes

Many packages will come with a “vignette”, or a tutorial on how to use the package. These are very helpful tools for figuring out how to use a package.

To get a list of all the vignettes a package has, use the `vignette` function, specifying a package name with `package`:

```
vignette(package = "phonenumbers")
```

## Package vignettes

To open a vignette once you know its name, you can also use the vignette function. For example, to open the vignette called "phonenumbers" for the phonenumbers package, run:

```
vignette(topic = "phonenumbers",
          package = "phonenumbers")
```

Many packages only have one vignette, with the same name as the package. In that case, the vignette topic usually has the same name as the whole package, so you can open the vignette using a shorter call:

```
vignette(topic = "phonenumbers")
```

## Function helpfiles

Packages include helpfiles for all the functions they intend for people to use.

You can access the helpfile for a function using ? followed by the function name. For example, to get the helpfile for letterToNumber, you can run:  
?letterToNumber