# Install R Packages

Sometimes you need to do analysis with functions that don't come pre-installed with R. In those cases, you need to install the package that contains the functions you need.

In order to use the functions within a package, you must both **install** the package and **load** the package. The instructions below will help you through this process.

# How do you know what packages you already have installed?

In **R for OSX** or **R GUI**, or **R Studio**, use the **library( )** command to print a list of available packages to the screen.

```
> library()
```

This will give you the name of each package installed on your computer, as well as a brief description of the functions in that package.

## Point and Click options

**R for OSX** has a "Packages & Data" menu. Packages & Data > Package Manager will provide a list of all the packages installed, and checkboxes indicate whether or not a package is loaded.

**R Studio** also has a "Packages" tab in one of its panels. Where this tab is located depends on how you have configured your R Studio layout. This tab lists all the packages installed, and checkboxes indicate whether or not a package is loaded.

**R GUI** does not have a point-and-click way to view installed packages.

# What functions are available in pre-installed packages?

If you would like to see which functions are available in a package you already have installed, you can use the **library( )** and **help( )** functions together to get this information. Take the "base" package as an example:

```
> library(help = base)
```

This lists all the functions available in "base". The "base" package is the heart of R, so it installed when you install R on your computer, and it is automatically loaded when you start R.

# Installing new packages

You may find that the pre-installed packages do not have the functions you need to use. There are many R packages you can download to expand the range of analysis you can perform. These packages, like R itself, are freely available to download.

Packages can be downloaded from the Comprehensive R Archive Network (CRAN). This is the same site where you downloaded R itself. However, instead of using a web browser to go to the CRAN website, you can

use R itself to initialize the download.

First, some terminology:

**Repository**: The service through which you search for and download packages. CRAN should have all of the packages you will need, but be aware that there exist other repositories for R packages.

**Mirror**: The CRAN server from which you will download your packages. There are many CRAN mirrors, all of which contain exactly the same files for download (hence mirror). Select one that is physically close to your current location.

**Dependency**: Some packages need to work with other packages in order to function. When downloading a new package, remember to download all the other packages required for it to function.

**Install Location**: Where do you want to install your package, and who do you want to have access to it? The default location is system-level, meaning that any user account will be able to load the package into R without having to reinstall. This setting will work fine for most any circumstance you'll encounter. But be aware that this is a setting you can change should the need arise.

The following examples illustrate how to install the package 'car', which is used in regression analysis.

## Instructions for R (OSX)

**Command Line** Use the **install.packages( )** function. Executing this function will bring up a window asking you to select a CRAN mirror (first time only). After you have selected the appropriate mirror, click "OK" and the package download will commence.

```
> install.packages('car')
```

**Point-and-Click** Use the "Packages & Data" menu at the top of the screen.

1. Select the "Package Installer" option, which will bring up the "R Package Installer" screen from which you can complete the installation.
2. Select your repository (set to "CRAN (binaries)").
3. Click on "Get List" to generate a list of packages available at that repository, or use the search bar to find your desired package.
4. Be sure that the "Install Dependencies" box is checked. Leave the "Install Location" as "At System Level". Select the desired package and click "Install Selected".

**Keyboard Shortcut** Option + Cmd + D will bring up the "R Package Installer" screen as described above. Follow the steps for **Point-and-Click**.

## Instructions for R Studio (Windows, OSX)

**Command Line** Use the **install.packages( )** function. Executing this function will bring up a window asking you to select a CRAN mirror (first time only). After you have selected the appropriate mirror, click "OK" and the package download will commence.

```
> install.packages('car')
```

**Point-and-Click** Use the "Tools" menu at the top of the screen.

1. Select “Install Packages”, which will bring up a screen from which you can search for your package.
2. Set the “Install from” pull-down menu to “Repository (CRAN)” (the default).
3. Also leave the “Install to Library” set to its default.
4. Check “Install dependencies” and click “Install”.

**Keyboard Shortcut** None.

### Instructions for R Gui (Windows)

**Command Line** Use the **install.packages( )** function. Executing this function will bring up a window asking you to select a CRAN mirror (first time only). After you have selected the appropriate mirror, click “OK” and the package download will commence.

```
> install.packages('car')
```

**Point-and-Click** Use the “Packages” menu at the top of the screen.

1. Select the “Install package(s)” option.
2. You will be prompted to select a CRAN mirror if you have not already (you can also select a CRAN mirror with the “Set CRAN mirror” option in the “Packages” menu).
3. After selecting your mirror, you will be given a list of packages. Select your desired package and click “OK”.

**Keyboard Shortcut** None.

## Loading packages

Why is it necessary to load installed packages? Think of packages as analagous to any other software on your computer. You may have Microsoft Word installed, but you can’t use it unless you open it first.

Two functions can be used to load packages:

```
> library('car')
> require('car')
```

While both will technically work, the preferred function is **library( )**. The **require( )** function is designed to work as a component of other functions you may write yourself. Consequently its behavior is slightly different. It is recommended that you use **library( )** when loading packages.

After you have loaded a package, you will have access to its documentation. You can generate a list of functions within that package as before:

```
> library(help=car)
```

You can also get the help documentation for a specific functions:

```
> help(scatterplot)
```