**R packages**

[ALL PLAGIARISED]

A package is a collection of functions, data, and code conveniently provided in a nice, complete format for you. A package is not to be confused with a library (these two terms are often conflated in colloquial speech about R). A library is the place where the package is located on your computer. To think of an analogy, a library is, well, a library… and a package is a book within the library. The library is where the books/packages are located.

Each package is developed and published by the R community at large and deposited in repositories. A repository is a central location where many developed packages are located and available for download.

There are three big repositories:  
1. [**CRAN (Comprehensive R Archive Network):**](https://cran.r-project.org/web/packages/) R’s main repository (>12,100 packages available!)  
2. [**BioConductor:**](https://bioconductor.org/packages/release/BiocViews.html#___Software) A repository mainly for bioinformatic-focused packages  
3. [**GitHub:**](https://github.com/collections) A very popular, open-source repository (not R specific!)

*Searching for packages:*

There are a few different avenues for exploring packages. First, CRAN groups all of its packages by their functionality/topic into 35 “themes.” It calls this its [“Task view.”](https://cran.r-project.org/web/views/) Second, there is a great website, **[RDocumentation,](https://www.rdocumentation.org/" \t "_blank)** which is a search engine for packages and functions from CRAN, BioConductor, and GitHub (ie: the big three repositories). And Google.

*Installing packages:*

Graphical user interface, text

Description automatically generated

**Installing from Bioconductor**  
The BioConductor repository uses their own method to [install packages](https://www.bioconductor.org/install/). First, to get the basic functions required to install through BioConductor, use: source("https://bioconductor.org/biocLite.R")

This makes the main install function of BioConductor, biocLite(), available to you. Following this, you call the package you want to install in quotes, between the parentheses of the biocLite command, like so: biocLite("GenomicFeatures")

**Installing from GitHub**  
This is a more specific case that you probably won’t run into too often. In the event you want to do this, you first must find the package you want on GitHub and take note of both the package name AND the author of the package. Check out [this guide](http://kbroman.org/pkg_primer/pages/github.html) for installing from GitHub, but the general workflow is:

1. install.packages("devtools") - only run this if you don’t already have devtools installed. If you’ve been following along with this lesson, you may have installed it when we were practicing installations using the R console
2. library(devtools) - more on what this command is doing immediately below this
3. install\_github("author/package") replacing “author” and “package” with their GitHub username and the name of the package.

**General tips**

There is an order to loading packages - some packages require other packages to be loaded first (**dependencies**). That package’s manual/help pages will help you out in finding that order, if they are picky.

* Checking what packages you have installed: library()
* Check what packages require update: old.packages()
* Update all packages: update.packages()
* Update a specific packages: install.packages(“packagename)

**Unloading packages**

Sometimes you want to unload a package in the middle of a script - the package you have loaded may not play nicely with another package you want to use.

To unload a given package you can use the detach() function. For example, detach("package:ggplot2", unload=TRUE) would unload the ggplot2 package

**Uninstalling packages**

If you no longer want to have a package installed, you can simply uninstall it using the function remove.packages(). For example, remove.packages("ggplot2")

**Help function**

In the console, you can use the help() function to access a package’s help files. Try help(package = "ggplot2") and you will see all of the *many* functions that ggplot2 provides.

If you still have questions about what functions within a package are right for you or how to use them, many packages include **“vignettes.”** These are extended help files, that include an overview of the package and its functions, but often they go the extra mile and include detailed examples of how to use the functions in plain words that you can follow along with to see how to use the package. To see the vignettes included in a package, you can use the browseVignettes() function. For example, let’s look at the vignettes included in ggplot2:browseVignettes("ggplot2") .