



SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

Approved by AICTE, New Delhi, Affiliated to Anna University, Chennai

Accredited by NAAC-UGC with 'A++' Grade (Cycle III) &

Accredited by NBA (B.E - CSE, EEE, ECE, Mech & B.Tech.IT)

COIMBATORE-641 035, TAMIL NADU



UNIT II

Packages and Libraries

Packages are collections of R functions, data, and compiled code in a well-defined format, created to add specific functionality. There are 10,000+ user contributed packages and growing.

There are a set of **standard (or base) packages** which are considered part of the R source code and automatically available as part of your R installation. Base packages contain the **basic functions** that allow R to work, and enable standard statistical and graphical functions on datasets; for example, all of the functions that we have been using so far in our examples.

The directories in R where the packages are stored are called the **libraries**. The terms *package* and *library* are sometimes used synonymously and there has been [discussion](#) amongst the community to resolve this. It is somewhat counter-intuitive to *load a package* using the `library()` function and so you can see how confusion can arise.

You can check what libraries are loaded in your current R session by typing into the console:

```
sessionInfo() #Print version information about R, the OS and attached or loaded packages  
  
# OR  
  
search() #Gives a list of attached packages
```

Previously we have introduced you to functions from the standard base packages. However, the more you work with R, you will come to realize that there is a cornucopia of R packages that offer a wide variety of functionality. To use additional packages will require installation. Many packages can be installed from the [CRAN](#) or [Bioconductor](#) repositories.

Helpful tips for package installations

- Package names are case sensitive!
- At any point (especially if you've used R/Bioconductor in the past), in the console R may ask you if you want to **“update any old packages by asking Update all/some/none? [a/s/n]:”**. If you see this, type **“a”** at the prompt and hit Enter to update any old packages. *Updating packages can sometimes take awhile to run.* If you are short on time, you can choose **“n”** and proceed. Without updating, you run the risk

of conflicts between your old packages and the ones from your updated R version later down the road.

- If you see a message in your console along the lines of “binary version available but the source version is later”, followed by a question, “**Do you want to install from sources the package which needs compilation? y/n**”, type **n** for no, and hit enter.

Package installation from CRAN

CRAN is a repository where the latest downloads of R (and legacy versions) are found in addition to source code for thousands of different user contributed R packages.



CRAN
[Mirrors](#)
[What's new?](#)
[Task Views](#)
[Search](#)

[About R](#)
[R Homepage](#)
[The R Journal](#)

[A3](#)
[abbyR](#)
[abc](#)
[ABCanalysis](#)
[abc.data](#)
[abcdeFBA](#)
[ABCOptim](#)
[ABCp2](#)
[abcrf](#)

Available CRAN Packages By Name

[A](#)[B](#)[C](#)[D](#)[E](#)[F](#)[G](#)[H](#)[I](#)[J](#)[K](#)[L](#)[M](#)[N](#)[O](#)[P](#)[Q](#)[R](#)[S](#)[T](#)[U](#)[V](#)[W](#)[X](#)[Y](#)[Z](#)

Accurate, Adaptable, and Accessible Error Metrics for Predictive Models
Access to Abby Optical Character Recognition (OCR) API
Tools for Approximate Bayesian Computation (ABC)
Computed ABC Analysis
Data Only: Tools for Approximate Bayesian Computation (ABC)
ABCDE_FBA: A-Biologist-Can-Do-Everything of Flux Balance Analysis with this package
Implementation of Artificial Bee Colony (ABC) Optimization
Approximate Bayesian Computational Model for Estimating P2
Approximate Bayesian Computation via Random Forests

Packages for R can be installed from the [CRAN](#) package repository using the `install.packages` function. This function will download the source code from on the CRAN mirrors and install the package (and any dependencies) locally on your computer.

An example is given below for the `ggplot2` package that will be required for some plots we will create later on. Run this code to install `ggplot2`.

```
install.packages("ggplot2")
```

Package installation from Bioconductor

Alternatively, packages can also be installed from [Bioconductor](#), another repository of packages which provides tools for the analysis and comprehension of high-throughput **genomic data**. These packages includes (but is not limited to) tools for performing statistical analysis, annotation packages, and accessing public datasets.



There are many packages that are available in CRAN and Bioconductor, but there are also packages that are specific to one repository. Generally, you can find out this information with a Google search or by trial and error.

To install from Bioconductor, you will first need to install BiocManager. *This only needs to be done once ever for your R installation.*

```
# DO NOT RUN THIS!
```

```
install.packages("BiocManager")
```

Now you can use the `install()` function from the `BiocManager` package to install a package by providing the name in quotations.

Here we have the code to install `ggplot2`, through Bioconductor:

```
# DO NOT RUN THIS!
```

```
BiocManager::install("ggplot2")
```

The code above may not be familiar to you - it is essentially using a new operator, a double colon `::` to execute a function from a particular package. This is the syntax: `package::function_name()`.

Package installation from source

Finally, R packages can also be installed from source. This is useful when you do not have an internet connection (and have the source files locally), since the other two methods are retrieving the source files from remote sites.

To install from source, we use the same `install.packages` function but we have additional arguments that provide *specifications to change from defaults*:

```
# DO NOT RUN THIS!
```

```
install.packages("~/Downloads/ggplot2_1.0.1.tar.gz", type="source", repos=NULL)
```

Loading libraries

Once you have the package installed, you can **load the library** into your R session for use. Any of the functions that are specific to that package will be available for you to use by simply calling the function as you would for any of the base functions. *Note that quotations are not required here.*

```
library(ggplot2)
```

You can also check what is loaded in your current environment by using `sessionInfo()` or `search()` and you should see your package listed as:

```
other attached packages:  
[1] ggplot2_2.0.0
```

In this case there are several other packages that were also loaded along with `ggplot2`.

`install.packages("light")`



`library("light")`



Images sourced from <https://www.wikihow.com/Change-a-Light-Bulb>