SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

Re-accredited by NAAC with A+ grade, Accredited by NBA(CSE, IT, ECE, EEE & Mechanical)
Approvedy by AICTE, New Delhi, Recognized by UGC, Affiliated to Anna University, Chennai

Department of MCA

Packages in R Programming

Course: 16CA817 - Big Data Analytics

Unit V : R Programming

II Semester / I MCA

Department of MCA

Topic: Packages in R Programming

COURSE
19CAT702
Big Data Analytics

UNIT - V
R

CLASS
III Semester / II MCA

Packages in R Programming

•The package is an appropriate way to organize the work and share it with others.

•Typically, a package will include code (not only R code!), documentation for the package and the functions inside, some tests to check everything works as it should, and data sets.

Packages in R

Packages in R Programming language are a set of R functions, compiled code, and sample data. These are stored under a directory called "library" within the R environment. By default, R installs a group of packages during installation.

Once we start the R console, only the default packages are available by default. Other packages that are already installed need to be loaded explicitly to be utilized by the R program that's getting to use them.

What are Repositories?

A repository is a place where packages are located and stored so you can install packages from it.

Organizations and Developers have a local repository, typically they are online and accessible to everyone.

Some of the most popular repositories for R packages are:

CRAN: Comprehensive R Archive Network(CRAN) is the official repository, it is a network of ftp and web servers maintained by the R community around the world. The R community coordinates it, and for a package to be published in CRAN, the Package needs to pass several tests to ensure that the package is following CRAN policies.

Bioconductor: Bioconductor is a topic-specific repository, intended for open source software for bioinformatics. Similar to CRAN, it has its own submission and review processes, and its community is very active having several conferences and meetings per year in order to maintain quality.

Github: Github is the most popular repository for open source projects. It's popular as it comes from the unlimited space for open source, the integration with git, a version control software, and its ease to share and collaborate with others.

Install an R-Packages

There are multiple ways to install R Package, some of them are,

Installing Packages From CRAN:

For installing Package from CRAN we need the name of the package and use the following command:

install.packages("package name")Installing Package from CRAN is the most common and easiest way as we just have to use only one command. In order to install more than a package at a time, we just have to write them as a character vector in the first argument of the install.packages() function:

Example:

```
install.packages(c("vioplot", "MASS"))
```

Update, Remove and Check Installed Packages in R

To check what packages are installed on your computer, type this command:

```
installed.packages()
```

To update all the packages, type this command:

```
update.packages()
```

To update a specific package, type this command:

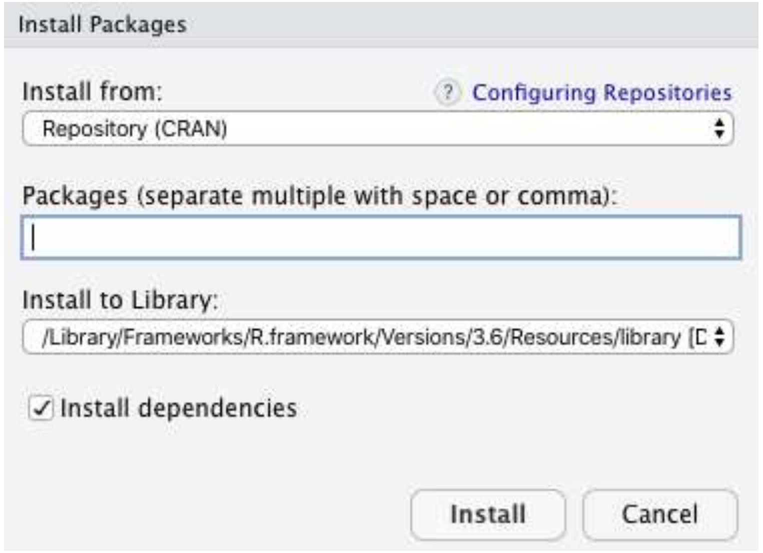
```
install.packages("PACKAGE NAME")
```

Installing Packages Using RStudio UI

- In R Studio goto Tools -> Install Package, and there we will get a pop-up window to type the package you want to install:

Installing Packages Using RStudio UI

Under Packages, type, and search Package which we want to install and then click on install button.



Install Packages

Install from:

Configuring Repositories

Repository (CRAN)

Packages (separate multiple with space or comma):

Install to Library:

/Library/Frameworks/R.framework/Versions/3.6/Resources/library [D

[✓] Install dependencies

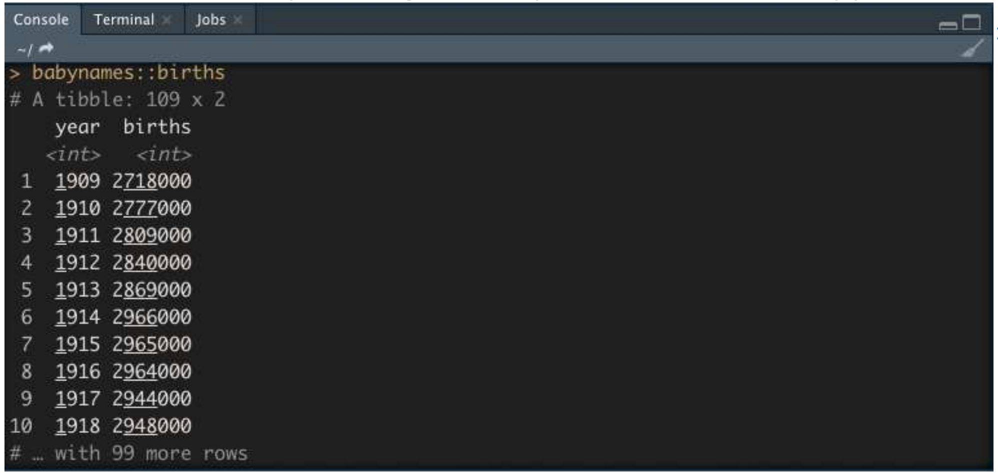
How to Load Packages in R Programming Language

When a package is installed, we are ready to use its functionalities. If we just need a sporadic use of a few functions or data inside a package we can access them with the following notation.

packagename::functionname()

Example: Let's access the births function of package babynames. Then type this command,

babynames::births



```
> babynames::births
# A tibble: 109 x 2
    year  births
   <int>   <int>
 1  1909 2718000
 2  1910 2777000
 3  1911 2809000
 4  1912 2840000
 5  1913 2869000
 6  1914 2966000
 7  1915 2965000
 8  1916 2964000
 9  1917 2944000
10  1918 2948000
# … with 99 more rows
```

Difference Between a Package and a Library

There is always confusion between a package and a library, and we find people calling libraries as packages.

library(): It is the command used to load a package, and it refers to the place where the package is contained, usually a folder on our computer.

Package: It is a collection of functions bundled conveniently. The package is an appropriate way to organize our own work and share it with others.

Load More Than One Package at a Time

We can just input a vector of names to the **install.packages()** function to install a package, in the case of the **library()** function, this is not possible. We can load a set of packages one at a time, or if you prefer, use one of the many work arounds developed by R users.

Unload a Package in R Language

To unload a given package, use the detach() function. The use will be:

```
detach("package:babynames", unload = TRUE)
```

R Packages

Packages are part of R programming and they are useful in collecting sets of R functions into a single unit. It also contains compiled code and sample data. All of these are kept stored in a directory called the "library" in the R environment. In this chapter you will learn about the concepts that are within R packages.

A package is a set of R functions and data-sets and the library is a folder on your system / computer which stores the files for those package(s).

Loading Packages in R

R Packages

Packages are part of R programming and they are useful in collecting sets of R functions into a single unit. It also contains compiled code and sample data. All of these are kept stored in a directory called the "library" in the R environment. In this chapter you will learn about the concepts that are within R packages.

A package is a set of R functions and data-sets and the library is a folder on your system / computer which stores the files for those package(s).

Loading Packages in R

R - Packages

R packages are a collection of R functions, complied code and sample data. They are stored under a directory called "library" in the R environment. By default, R installs a set of packages during installation.

More packages are added later, when they are needed for some specific purpose.

When we start the R console, only the default packages are available by default. Other packages which are already installed have to be loaded explicitly to be used by the R program that is going to use them.

Check Available R Packages

Get library locations containing R packages

```
.libPaths()
```

When we execute the above code, it produces the following result. It may vary depending on the local settings of your pc.

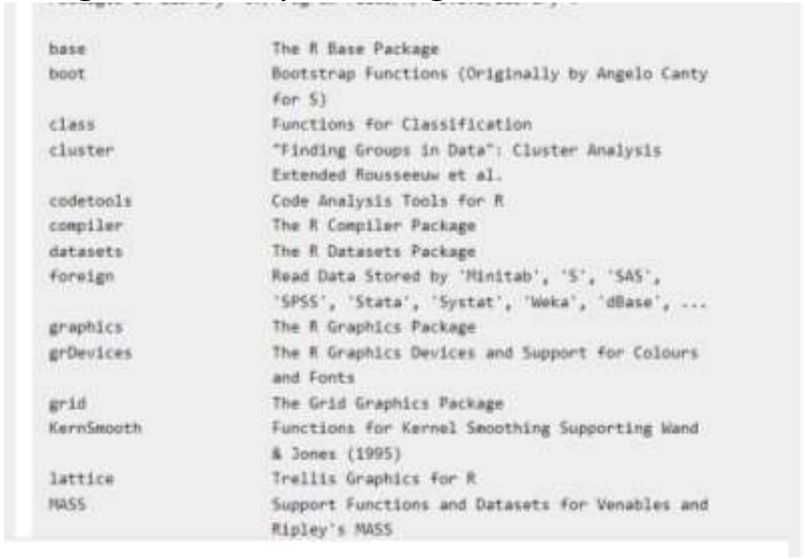
```
[2] "C:/Program Files/R/R-3.2.2/library"
```

Get the list of all the packages installed

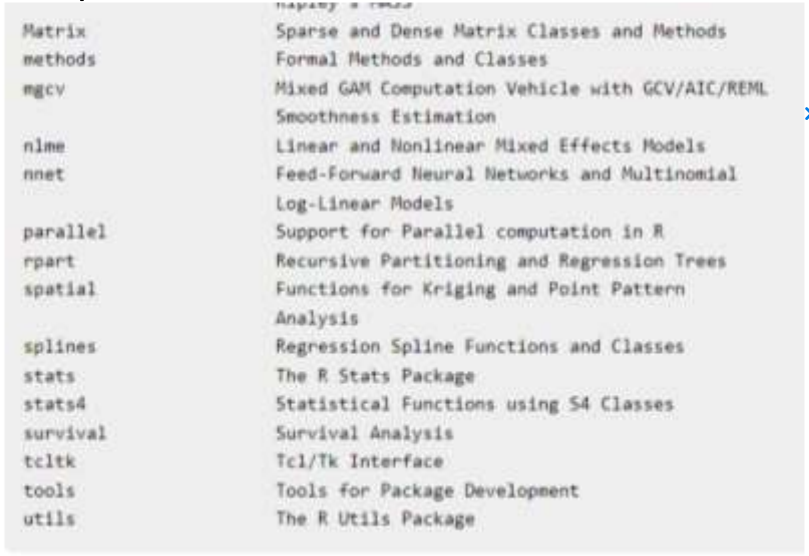
library()

When we execute the above code, it produces the following result. It may vary depending on the local settings of your pc.

Packages in library ‘C:/Program Files/R/R-3.2.2/library’:



```
base                    The R Base Package
boot                    Bootstrap Functions (Originally by Angelo Canty
                        for S)
class                   Functions for Classification
cluster                 "Finding Groups in Data": Cluster Analysis
                        Extended Rousseeuw et al.
codetools               Code Analysis Tools for R
compiler                The R Compiler Package
datasets                The R Datasets Package
foreign                 Read Data Stored by 'Minitab', 'S', 'SAS',
                        'SPSS', 'Stata', 'Systat', 'Weka', 'dBase', ...
graphics                The R Graphics Package
grDevices               The R Graphics Devices and Support for Colours
                        and Fonts
grid                    The Grid Graphics Package
KernSmooth              Functions for Kernel Smoothing Supporting Wand
                        & Jones (1995)
lattice                 Trellis Graphics for R
MASS                    Support Functions and Datasets for Venables and
                        Ripley's MASS
```



```
Matrix                  Sparse and Dense Matrix Classes and Methods
methods                 Formal Methods and Classes
mgcv                    Mixed GAM Computation Vehicle with GCV/AIC/REML
                        Smoothness Estimation
nlme                    Linear and Nonlinear Mixed Effects Models
nnet                    Feed-Forward Neural Networks and Multinomial
                        Log-Linear Models
parallel                Support for Parallel computation in R
rpart                   Recursive Partitioning and Regression Trees
spatial                 Functions for Kriging and Point Pattern
                        Analysis
splines                 Regression Spline Functions and Classes
stats                   The R Stats Package
stats4                  Statistical Functions using S4 Classes
survival                Survival Analysis
tcltk                   Tcl/Tk Interface
tools                   Tools for Package Development
utils                   The R Utils Package
```

Get the list of all the packages installed

•Get all packages currently loaded in the R environment

```
search()
```

When we execute the above code, it produces the following result. It may vary depending on the local settings of your pc.

```
[1] ".GlobalEnv" "package:stats" "package:graphics"
[4] "package:grDevices" "package:utils" "package:datasets"
[7] "package:methods" "Autoloads" "package:base"
```

Install a New Package

•There are two ways to add new R packages. One is installing directly from the CRAN directory and another is downloading the package to your local system and installing it manually.

•Install directly from CRAN

•The following command gets the packages directly from CRAN webpage and installs the package in the R environment. You may be prompted to choose a nearest mirror. Choose the one appropriate to your location.

•install.packages("Package Name") # Install the package named "XML".

```
install.packages("XML")
```

Install package manually

- Go to the link R Packages to download the package needed. Save the package as a .zip file in a suitable location in the local system.
- Now you can run the following command to install this package in the R environment.
- install.packages(file_name_with_path, repos = NULL, type = "source")
- # Install the package named "XML“

 install.packages("E:/XML_3.98-1.3.zip", repos = NULL, type = "source")

Load Package to Library

Before a package can be used in the code, it must be loaded to the current R environment. You also need to load a package that is already installed previously but not available in the current environment.

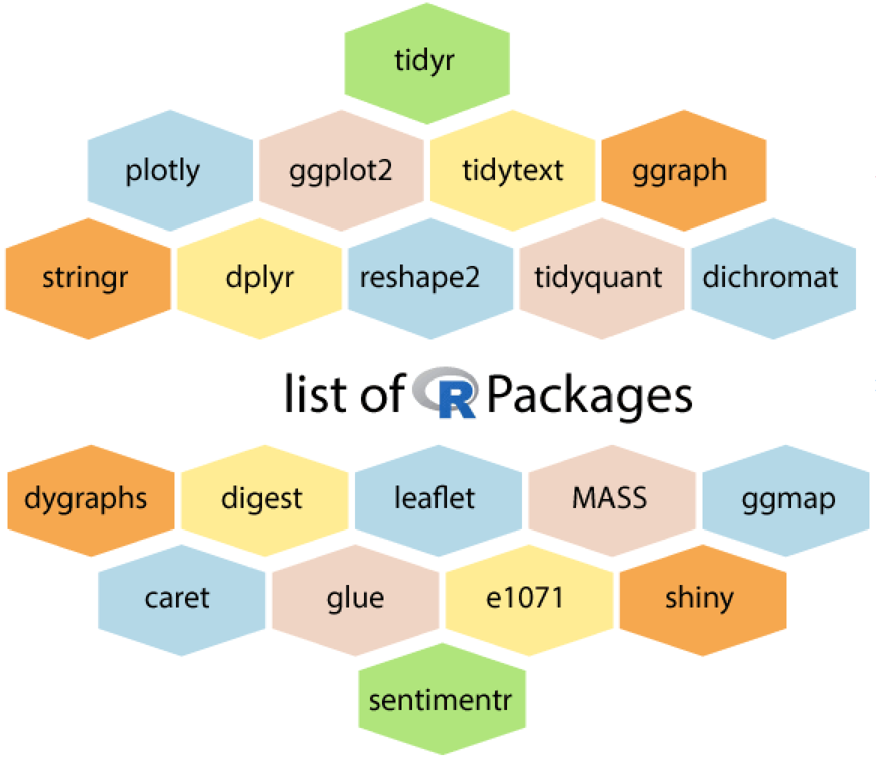
A package is loaded using the following command

library("package Name", lib.loc = "path to library") # Load the package named "XML"

install.packages("E:/XML_3.98-1.3.zip", repos = NULL, type = "source")

List of R packages

- R is the language of data science which includes a vast repository of packages. These packages appeal to different regions which use R for their data purposes. CRAN has 10,000 packages, making it an ocean of superlative statistical work. There are lots of packages in R, but we will discuss the important one.
- There are some mostly used and popular packages which are as follows:



List of R packages

1) tidyr

The word tidyr comes from the word tidy, which means clear. So the tidyr package is used to make the data' tidy'. This package works well with dplyr. This package is an evolution of the reshape2 package.

2) ggplot2

R allows us to create graphics declaratively. R provides the ggplot package for this purpose. This package is famous for its elegant and quality graphs which sets it apart from other visualization packages.

List of R packages

3) ggraph

R provides an extension of ggplot known as **ggraph**. The limitation of **ggplot** is the dependency on tabular data is taken away in ggraph.

4) dplyr

R allows us to perform data wrangling and data analysis. R provides the **dplyr** library for this purpose. This library facilitates several functions for the data frame in R.

5) tidyquant

The tidyquant is a financial package which is used for carrying out quantitative financial analysis. This package adds to the **tidyverse** universe as a financial package which is used for importing, analyzing and visualizing the data.

6) dygraphs

The dygraphs package provides an interface to the main JavaScript library which we can use for charting. This package is essentially used for plotting time-series data in R.

List of R packages

7) leaflet

For creating interactive visualization, R provides the leaflet package. This package is an open-source JavaScript library. The world's popular websites like the New York Times, Github and Flicker, etc. are using leaflet. The leaflet package makes it easier to interact with these sites.

8) ggmap

For delineating spatial visualization, the ggmap package is used. It is a mapping package which consists of various tools for geolocating and routing.

9) glue

R provides the glue package to perform the operations of data wrangling. This package is used for evaluating R expressions which are present within the string.

10) shiny

R allows us to develop interactive and aesthetically pleasing web apps by providing a shiny package. This package provides various extensions with HTML widgets, CSS, and JavaScript.

List of R packages

11) plotly

The plotly package provides online interactive and quality graphs. This package extends upon the JavaScript library -plotly.js.

12) tidytext

The tidytext package provides various functions of text mining for word processing and carrying out analysis through ggplot, dplyr, and other miscellaneous tools.

13) stringr

The stringr package provides simplicity and consistency to use wrappers for the 'stringi' package. The stringi package facilitates common string operations.

14) reshape2

This package facilitates flexible reorganization and aggregation of data using melt () and decast () functions.

15) dichromat

The R dichromat package is used to remove Red-Green or Blue-Green contrasts from the colors.

List of R packages

16) digest

The digest package is used for the creation of cryptographic hash objects of R functions.

17) MASS

The MASS package provides a large number of statistical functions. It provides datasets that are in conjunction with the book "Modern Applied Statistics with S."

18) caret

R allows us to perform classification and regression tasks by providing the caret package. CaretEnsemble is a feature of caret which is used for the combination of different models.

19) e1071

The e1071 library provides useful functions which are essential for data analysis like Naive Bayes, Fourier Transforms, SVMs, Clustering, and other miscellaneous functions.

20) sentimentr

The sentiment package provides functions for carrying out sentiment analysis. It is used to calculate text polarity at the sentence level and to perform aggregation by rows or grouping variables.

References

https://www.javatpoint.com/list-of-r-packages
https://www.geeksforgeeks.org/packages-in-r-programming/
https://www.tutorialspoint.com/r/r_packages.htm