

# Topic 2: Programming with R

2023-08-08

In this topic, you will learn about :

- Flow control
- Introduction to function
- Packages, libraries, and repositories

## Packages

### Packages

In R programming, a package is a collection of functions, data, and documentation that extends the capabilities of R. Packages provide a way to organize and distribute R code, making it easier for users to access additional functionality beyond what comes with the base R installation.

You can install and load packages to use their functions and data in your R scripts.

Example: Installing and Loading a Package

```
# Installing a package (only needs to be done once)  
install.packages("dplyr")  
  
# Loading the package (needs to be done in each R session)  
library(dplyr)
```

## Libraries

### Libraries

In R, the term “library” is often used interchangeably with “package.” When you load a package using the `library()` function, you are essentially loading the package’s functions and data into your current R session. After loading a package, you can access its functions directly by their names.

Example: Using Functions from a Loaded Package

```
# Assuming 'dplyr' package is installed and loaded  
# Using the 'filter' function from the 'dplyr' package  
filtered_data <- filter(my_data_frame, column_name > 10)
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

## Repositories

R packages are hosted in repositories, which are collections of packages accessible through the internet. The Comprehensive R Archive Network (CRAN) is the primary and most widely used repository for R packages. CRAN hosts thousands of packages, and you can install packages directly from there using the `install.packages()` function.

Example: Installing a Package from CRAN

```
# Installing the 'ggplot2' package from CRAN  
install.packages("ggplot2")
```

Apart from CRAN, there are other repositories like Bioconductor (specializing in bioinformatics packages) and GitHub, where developers share R packages.

Example: Installing a Package from GitHub

```
# Installing a package from GitHub using the 'remotes' package  
# First, install 'remotes' if not already installed  
install.packages("remotes")  
  
# Load the 'remotes' package  
library(remotes)  
  
# Install the package from GitHub  
install_github("username/repo")
```

Using packages in R makes it easier to leverage the work of other developers, collaborate, and build on top of existing functionality. It is a powerful mechanism for extending R's capabilities and enhancing your data analysis, visualization, and programming tasks.