**Module # 10 Building your own R package**

Package in R is a structured, standardized unit consisting of  
1. R code  
2. Documentation  
3. Data  
4. External code

Creating a package is initiated by generating a new blank project in R.  
Installing '**devtools**' and '**roxygen**' which help in creating packages and can be done by the following code:  
install.packages(“devtools”)  
install.packages(“roxygen2”)  
library(devtools)  
library(roxygen2)  
package.skeleton()

It can also be installed directly by implementing code   
devtools::install\_github("hadley/devtools")

It creates new files:

* Data, folder
* DESCRIPTION, information file
* Man, help folder
* R folder contains the . R files for functions
* And “read-and-delete-me” file

The ***Description*** consists of fundamental information regarding the package in the following sequence: The important and required fields are ‘Package’, ‘Version’, ‘License’, ‘Description’, ‘Title’, ‘Author’, and ‘Maintainer’ and other fields are optional.

Package: Veda  
Title: Function for Addition  
Version: 0.1.0  
Date: 2023-03-20  
Authors@R: c(person("Veda", "Developer", role = c("aut", "cre"),  
                     email = "vedar@some.domain.net"),  
                     person("A.", "User", role = "ctb",  
                     email = "A.User@whereever.net"))  
Author: Veda Developer [aut, cre],  
        A. User [ctb]  
Maintainer: Veda Developer <vedar@some.domain.net>  
Description: This package provides function for adding values in R.It returns the value after calling the function which when implemented can be used to count n number of digits.   
Depends: R (>= 3.1.0)  
License: CC0  
LazyData: true  
RoxygenNote: 7.2.3  
Built: R 4.2.2; ; 2023-03-21 03:56:59 UTC; windows

The required '**Package**' variable represents the package's identity. It should comprise (ASCII) letters, numerals, and dots, include at least two characters, begin with a letter, and not end with a dot.   
The required '**Title**' domain must include a brief summary of the package. A few package items may limit the tagline length to 65 characters. It should use title case (that is, capitalize the main words; tools::toTitleCase can assist you with this). It should not include markup, continuation lines, and period.   
The required '**Version**' indicates the package's edition. This is a string that includes a minimum of 2 (typically 3)  integers which are not negative are isolated by single '.' or '-' characters. The foundational form should be in the format of '0.01' or '0.01.0' or are treated as if they were '0.1-0'. This isn't a decimal number, so for example 0.9 < 0.75 since 9 < 75.  
The '**Date**' field is optional and displays the launch date of the current version of the package. Format of 'yyyy-mm-dd'  is highly advised.  
**'Authors@R**' is the machine-readable description of the author, maintainer, contributors, and others. Fields **'Author'**and **'Maintainer'** can be auto-generated from **'Authors@R**'. These two fields can be omitted if the latter is included.

Author: Veda Developer [aut, cre],  
        A. User [ctb]  
Maintainer: Veda Developer [vedar@some.domain.net](mailto:vedar@some.domain.net)

The required '**Description**' domain must contain a detailed description of the tasks the package performs. It should be understandable to the entire intended audience. One paragraph of what the package can perform. It can extend up to 80 characters or less. The following lines should be indented by four spaces.  
The '**Depends**' field will list the dependencies needed for loading the package. So, these will be installed when the main package is installed. These are loaded as part of its loading. It is visible to the user.  
The '**Imports**' field will list all the packages that package uses but are not attached to the search path. This means that your package can use the functions from the package, but you will need to call them through package:: function(). These are not loaded and are not visible to the user.  
'**License**' gives the option of the extent of usage of the package i.e whether it is public or limited to some. It should be specified in a standardized form. CCO 1.0 Universal is a dedication to the public waiving all the rights of the work. There are other licenses which dedicate the extent of usage of the package. There is no warranty of correctness or accountability.  
The **LazyData** package provides a different strategy in which datasets are inserted into memory just when required and stored for better accessibility in subsequent sessions. This is notably helpful with big sets of data projects or immersive work where there is the requirement to use subsets of the data at a period.

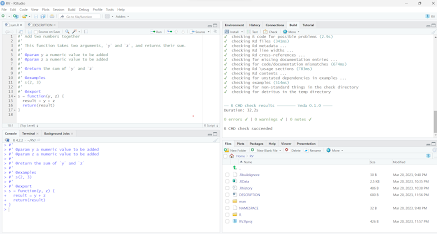
***Building Package:***

It is important to save the code and description accurately.

R file code:

#' Add two numbers together  
#'  
#' This function takes two arguments, `y` and `z`, and returns their sum.  
#'  
#' @param y a numeric value to be added  
#' @param z a numeric value to be added  
#'  
#' @return the sum of `y` and `z`  
#'  
#' @examples  
#' s(2, 3)  
#'  
#' @export  
s = function(y, z) {  
  result = y + z  
  return(result)  
}

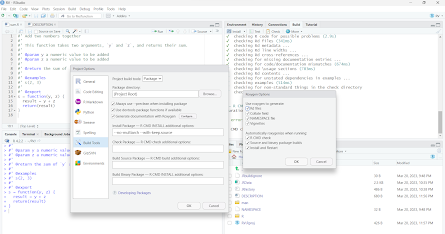
*Build tab and check* -allows error checking and building.

[](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEizixdeP5uymu_ML71_PDddJFElJOPH6H4A46i3roDs5R4czVX6sLRiBsQM3mf237zV8JV69mE1Lpme4dkS40t2AkkfLqLeh9J3tCVSmJUZsLF-YNWZP2AMmzGeddqNUwcfvQsPRQ04dLunhY-9lu9NlnyFir4SHUDpWzL8a-Aec2yy4zp5kdRldiZe/s1903/build.png)

*Roxygen-documentation* set up

* In the build tab, configure build tools is selected followed by turning on the generate documentation with roxygen.
* later in roxygen options, all options are turned on.
* It is important to start the code file with #’ which documents the code using the oxygen package followed by check, build and reload.

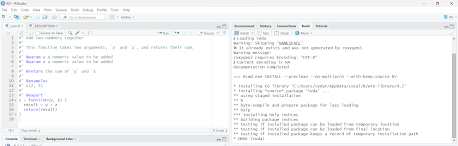
% Please edit documentation in R/sum.R  
\name{s}  
\alias{s}  
\title{Add two numbers together}  
\usage{  
s(y, z)  
}  
\arguments{  
\item{y}{a numeric value to be added}  
\item{z}{a numeric value to be added}  
}  
\value{  
the sum of `y` and `z`  
}  
\description{  
This function takes two arguments, `y` and `z`, and returns their sum.  
}  
\examples{  
s(2, 3)  
}

**[](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEj0i_xSuwB2UGWLOIv0obYUeJRM2i0jwg3wdXR5KvuagBy2qHw8oQrjVRh4BM-7NvU-s2erMBe6JIHRpwi6QFdlSZ12gdR4BQCDWI8vXHmhwZRPo92co8zrbRFrT3opxrKGM27tmGdlXeJksSKQTXqIidRy4rsrp8Exo7LD3F39w26onLYtUmwwA93H/s1917/roxy.png)**

#### 

While submitting to CRAN, a file which is in a.tar.gz is created.

* **build** command will collate the package.
* **install** command which tests if any errors and this creates Package in R.

[](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEjk49WrJsQRUDrsHfIgAOa0gi7tkklsrBD-BW9A41hvYShSsJ-fohcDC1vSppMDhbVTPIBNItYP51ItF02wpRU3x440-kLjrsBu902rd9RKdvfUGZs87BVeeL8LpipAYpNOfvj3SaEo2QMCmRMAtvqAJGsKPO0DBtYIdleECLCE-zGOBlaAkxnfTKFE/s1915/install.png)

**References**

Wickham, H. (2015). *R Packages*(Chapter 7)*.*

Matloff. N. (2011). *The Artof R programming*.(Chapter 13)

The R repository <https://cran.r-project.org/web/packages>