

R Programming Assignment-8

1. In the R programming language, how do we install packages?

ANS: R programming language doesn't come with all packages installed, and they need to be installed explicitly. In this article, we will discuss How to Install a Package in the R language.

Installing packages in R studio:

USING APPLICATION OPTIONS:

1. Open
2. Select tools
3. After selecting the tools you need to press install packages.
4. Here you need to give the package name you need to install.

Here we used expm. This function computes the exponential of a square matrix.

INSTALLING PACKAGES USING RGUI:

1. Open RGui
2. Select packages
3. Select install packages.
4. Select required package and click ok.

Package will be installed

2.What is the purpose of the with() function in R?

ANS:

with() function:

We often come across situations wherein we feel the need to build customized/user-defined functions to conduct out a certain operation. With R with() function, we can operate on R expressions as well as the process of calling that function in a single go!

That is with() function enables us to evaluate an R expression within the function to be passed as an argument. It works on data frames only. That is why the outcome of the evaluation of the R expression is done with respect to the data frame passed to it as an argument.

Syntax:

with(data-frame, R expression)

3.What is the purpose of the by() function in R?

ANS: The by() function in R is an easy function that allows us to group data within a data set, and perform mathematical functions on it.

It takes a vector, a string, a matrix, or a data frame as input and computes that data based on the mentioned functions.

Syntax:

The by() function takes the data as input and computes that based on a given function.

by(x,indices,FUN)

Where,

X = The input data frame.

Indices = It is the list of variables or the factors.

FUN = The function which needs to be applied for the variables/factors.

A Simple Example of by() function in R:

In this section, we are going to try out a simple example. For this purpose, we are using the 'iris' dataset. The reason to use this dataset is that it contains categorical data with respect to the numerical value.

Let's import the dataset by the following code.

CODE:

```
#importing data and assigning to variable df
df<-iris
#computes the mean for species categories in terms of petal.width
by(df$Petal.Width,list(df$Species),mean)
```

Output:

```
-----
: setosa
[1] 0.246
```

```
-----
: versicolor
[1] 1.326
```

```
-----
: virginica
[1] 2.026
-----
```

4.Create a programme that prints numbers from 1 to 100, with 'iNeuron' printed for multiples of 5?

ANS:

R Programming Code :

```
for (n in 1:100)
{
  if (n %% 5 == 0) {print("iNeroun")}
  else print(n)
}
```

5.How about writing an R application to read a .csv file?

ANS:

R CSV Files:

R provides functions to read and write to various file formats. In this R Tutorial, we shall look specifically into CSV Files.

We shall learn R functions to :

- [R Read CSV Files](#)
- [R Process CSV Files](#)
- [R Write CSV Files](#)

Example of a CSV File, that we use in the examples going forward, is given below :

```
Andrew,28,25.2
Mathew,23,10.5
Dany,49,11
Philip,29,21.9
John,38,44
```

Bing,23,11.5

Monica,29,45

You may refer [R Working Directory](#) to modify the path of R Workspace to point to the directory containing your input files (CSV Files)

Read CSV Files:

CSV Files are those files with values separated by commas in each row. Each row corresponds to a record or observation.

The syntax of function to read CSV File in R programming language is

`read.csv(<filename>).`



6.What does the factor variable stand for?

ANS: Factor in R is a variable used to categorize and store the data, having a limited number of different values. It stores the data as a vector of integer values. Factor in R is also known as a categorical variable that stores both string and integer data values as levels. Factor is mostly used in Statistical Modeling and exploratory data analysis with R.

In a dataset, we can distinguish two types of variables: **categorical** and **continuous**.

- In descriptive statistics for categorical variables in R, the value is limited and usually based on a particular finite group. For example, a categorical variable in R can be countries, year, gender, occupation.
- A continuous variable, however, can take any values, from integer to decimal. For example, we can have the revenue, price of a share, etc..

7.In R programming, how do you concatenate strings?

ANS: R – Concatenate Strings:

In this tutorial, we will learn how to Concatenate Strings in R programming Language.

To concatenate strings in r programming, use paste() function

The syntax of paste() function that is used to concatenate two or more strings.

```
paste(..., sep="", collapse=NULL)
```

While concatenating strings in R, we can choose the separator and number number of input strings. Following examples demonstrate different scenarios while concatenating strings in R using paste() function.

Example 1 – Concatenate Strings in R:

In this example, we will use paste() function with default separator, and concatenate two strings.

```
r_strings_concatenate_two.R
```

```
# Example R program to concatenate two strings
```

```
str1 = 'Hello'
```

```
str2 = 'World!'
```

```
# concatenate two strings using paste function
```

```
result = paste(str1,str2)
```

```
print (result)
```

Output:

```
$ Rscript r_strings_concatenate_two.R
```

```
[1] "Hello World!"
```

8. In R programming, what is the when() function?

ANS: **when** is a flavour of pattern matching (or an if-else abstraction) in which a value is matched against a sequence of condition-action sets. When a valid match/condition is found the action is executed and the result of the action is returned.

USAGE:

When(., ...)

Arguments:

.

the value to match against

...

formulas; each containing a condition as LHS and an action as RHS. named arguments will define additional values.

Value:

The value resulting from the action of the first valid match/condition is returned. If no matches are found, and no default is given, NULL will be returned.

Validity of the conditions are tested with **isTRUE**, or equivalently with **identical(condition, TRUE)**. In other words conditions resulting in more than one logical will never be valid. Note that the input value is always treated as a single object, as opposed to the **ifelse** function.