In R, we can read data from files stored outside the R environment.

```
# Get and print current working directory.

print(getwd())
```

```
# Get and print current working directory.

print(getwd())

# Set current working directory.

setwd("c://Users")
```

CSV file(comma separated value files)

The csv file is a text file in which the values in the columns are separated by a comma.

Reading a CSV File

Following is a simple example of read.csv() function to read a CSV file available in to current working directory -

```
data <- read.csv("alekhya.csv")

print(data)
```

```
data <- read.csv("alekhya.csv")

print(ncol(data))

print(nrow(data))
```

```
# Create a data frame.

data <- read.csv("alekhya.csv")
```

```
# Get the max salary from data frame.
sal <- max(data$salary)
print(sal)
```

Get the details of the person with max salary

```
# Create a data frame.
data <- read.csv("alekhya.csv")
# Get the max salary from data frame.
sal <- max(data$salary)
# Get the person detail having max salary.
retval <- subset(data, salary == max(salary))
print(retval)
```

Get all the people working in Finance department

```
# Create a data frame.
data <- read.csv("alekhya.csv")
retval <- subset( data, dept == "Finance")
print(retval)
```

Get the persons in Finace department whose salary is greater than 600

```
# Create a data frame.
data <- read.csv("input.csv")
info <- subset(data, salary > 600 & dept == "FICT")
print(info)
```

```
# Create a data frame.
data <- read.csv("input.csv")
info <- subset(data, salary > 100 & dept == "FICT")
print(info)
# Write filtered data into a new file.
write.csv(info,"output.csv",row.names = FALSE)
newdata <- read.csv("output.csv")
print(newdata)
```

Writing into a CSV File

R can create csv file form existing data frame. The write.csv() function is used to create the csv file. This file gets created in the working directory.

```
# Create a data frame.
data <- read.csv("input.csv")
info <- subset(data, salary > 100 & dept == "FICT")
print(info)
# Write filtered data into a new file.
```

```
write.csv(info,"output.csv")

newdata <- read.csv("output.csv")

print(newdata)


---------------------------------

R Excel file

install.packages("xlsx")

#Loading xlsx package

library("xlsx")

  # Reading the first worksheet in the file employee.xlsx.

excel_data<- read.xlsx("doc.xlsx", sheetIndex = 1)

print(excel_data)


library("xlsx")

#Creating data frame

emp.data<- data.frame(

  name = c("alekhya","madhu","yagnesh","rose","Yash"),

  salary = c(623.3,915.2,611.0,729.0,843.25),

  start_date = as.Date(c("2012-01-01", "2013-09-23", "2014-11-15", "2014-05-11","2015-
03-27")),

  dept = c("Operations","IT","HR","IT","Finance"))

   # Writing the first data set in employee.xlsxRscript

write.xlsx(emp.data, file = "doc.xlsx", col.names=TRUE,
row.names=TRUE,sheetName="Sheet2",append = TRUE)

# Reading the first worksheet in the file doc.xlsx.

excel_data<- read.xlsx("doc.xlsx", sheetIndex = 1)

print(excel_data)
```

```
# Reading the first worksheet in the file doc.xlsx.
excel_data<- read.xlsx("doc.xlsx", sheetIndex = 2)
print(excel_data)
```

R JSON File

JSON stands for JavaScript Object Notation.

The JSON file contains the data as text in a human-readable format. Like other files, we can also read and write into the JSON files. For this purpose, R provides a package named rjson, which we have to install with the help of the familiar command install.packages.

Install rjson package

By running the following command into the R console, we will install the rjson package into our current working directory.

```
install.packages("rjson")
```

Read the JSON file

Reading the JSON file in R is a very easy and effective process. R provide from JSON() function to extract data from a JSON file.

```
# Loading the package which is required to read JSON files.
```

```
library("rjson")

# Giving the input file name to the function fromJSON.
result <- fromJSON(file = "ja1.json")

# Printing the result.
print(result)
```

Converting JSON data to a Data Frame

R provide, as.data.frame() function to convert the extracted data into data frame

```
# Loading the package which is required to read JSON files.
library("rjson")

# Giving the input file name to the function fromJSON.
result <- fromJSON(file = "ja1.json")

# Converting the JSON record to a data frame.
data_frame <- as.data.frame(result)

#Printing JSON data frame
print(data_frame)
```

R XML File

Like HTML, XML is also a markup language which stands for Extensible Markup Language. It is developed by World Wide Web Consortium(W3C) to define the syntax for encoding documents which both humans and machine can read.
This file contains markup tags.

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

Creating XML File

We will create an xml file with the help of the given data.
We will save the following data with the .xml file extension to create an xml file. XML tags describe the meaning of data, so that data contained in such tags can easily tell or explain about the data.

```
# Loading the package required to read XML files.
library("XML")

# Also loading the other required package.
library("methods")

# Giving the input file name to the function.
```

```
result <- xmlParse(file = "xml_data.xml")


xml_data <- xmlToList(result)

print(xml_data)
```

Output

How to convert xml data into a data frame

```
# Loading the package required to read XML files.

library("XML")

# Also loading the other required package.

library("methods")

# Giving the input file name to the function xmlToDataFrame.

data_frame <- xmlToDataFrame("a1.xml")

#Printing the result

print(data_frame)
```