



# **SNS COLLEGE OF ENGINEERING**

Kurumbapalayam (Po), Coimbatore – 641 107

**An Autonomous Institution**

Accredited by NAAC – UGC with 'A' Grade

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

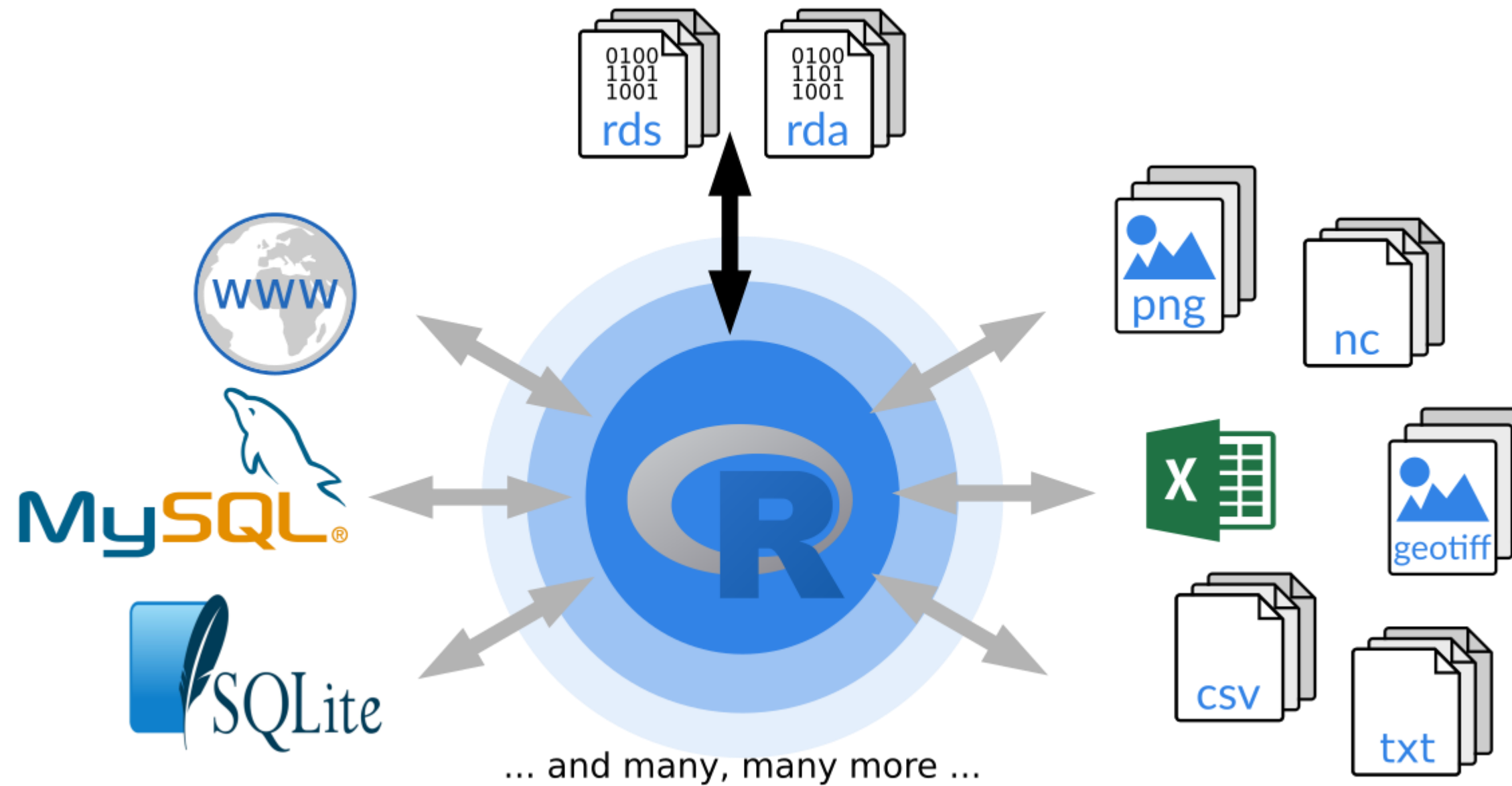


## **DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY**

**COURSE NAME :19CS407 DATA ANALYTICS WITH R**  
**II YEAR /IV SEMESTER**

**Unit 5- DATA VISUALIZATION USING R**

**Topic : Reading and getting data into R (External Data)**





# Reading Files in R Programming

- ✓ So far the operations using the R program are done on a prompt/terminal which is not stored anywhere.
- ✓ But in the software industry, most of the programs are written to store the information fetched from the program.
- ✓ One such way is to store the fetched information in a file. So the two most common operations that can be performed on a file are:
  - ✓ **Importing/Reading Files in R**
  - ✓ **Exporting/Writing Files in R**



# Reading Files in R Programming Language



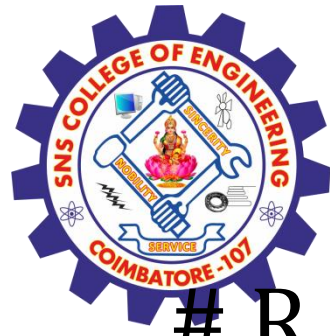
- When a program is terminated, the entire data is lost. Storing in a file will preserve our data even if the program terminates.
- If we have to enter a large number of data, it will take a lot of time to enter them all. However, if we have a file containing all the data, we can easily access the contents of the file using a few commands in R.
- You can easily move your data from one computer to another without any changes. So those files can be stored in various formats.
- It may be stored in .txt(tab-separated value) file, or in a tabular format i.e .csv(comma-separated value) file or it may be on internet or cloud.
- R provides very easier methods to read those files.



# File reading in R

- One of the important formats to store a file is in a text file. R provides various methods that one can read data from a text file.
- **read.delim()**: This method is used for reading “tab-separated value” files (“.txt”). By default, point (“.”) is used as decimal points.
- Syntax: **read.delim(file, header = TRUE, sep = “\t”, dec = “.”, ...)**
- Parameters:
  - file: the path to the file containing the data to be read into R.
  - header: a logical value. If TRUE, read.delim() assumes that your file has a header row, so row 1 is the name of each column. If that’s not the case, you can add the argument header = FALSE.
  - sep: the field separator character. “\t” is used for a tab-delimited file.
  - dec: the character used in the file for decimal points.





## Example



# R program reading a text file

```
# Read a text file using read.delim()  
myData = read.delim("geeksforgeeks.txt", header = FALSE)  
print(myData)
```

### **Output:**

*1 A computer science portal for geeks.*

Note: The above R code, assumes that the file “geeksforgeeks.txt” is in your current working directory. To know your current working directory, type the function `getwd()` in R console.



## read.delim2()

**read.delim2():** This method is used for reading “tab-separated value” files (“.txt”). By default, point (“,”) is used as decimal points.

Syntax: **read.delim2(file, header = TRUE, sep = “\t”, dec = “,”, ...)**

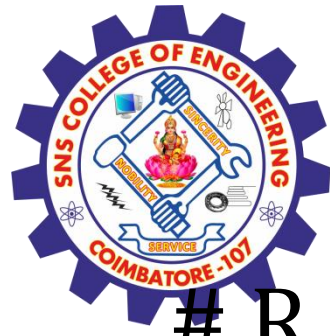
Parameters:

**file:** the path to the file containing the data to be read into R.

**header:** a logical value. If TRUE, read.delim2() assumes that your file has a header row, so row 1 is the name of each column. If that’s not the case, you can add the argument header = FALSE.

**sep:** the field separator character. “\t” is used for a tab-delimited file.

**dec:** the character used in the file for decimal points.



## Example



# R program reading a text file

```
# Read a text file using read.delim2
```

```
myData = read.delim2("geeksforgeeks.txt", header = FALSE)
```

```
print(myData)
```

Output:

```
1 A computer science portal for geeks.
```





## Choose

**file.choose()**: In R it's also possible to choose a file interactively using the function **file.choose()**, and if you're a beginner in R programming then this method is very useful for you.

Example:

```
# R program reading a text file using file.choose()
```

```
myFile = read.delim(file.choose(), header = FALSE)
```

```
# If you use the code above in RStudio
```

```
# you will be asked to choose a file
```

```
print(myFile)
```

Output:

1 A computer science portal for geeks.



## read\_tsv

***read\_tsv()***: This method is also used for to read a tab separated (“\t”) values by using the help of readr package.

*Syntax: read\_tsv(file, col\_names = TRUE)*

*Parameters:*

***file***: the path to the file containing the data to be read into R.

***col\_names***: Either TRUE, FALSE, or a character vector specifying column names. If TRUE, the first row of the input will be used as the column names.



## Example:

```
# R program to read text file
# using readr package
# Import the readr library
library(readr)
# Use read_tsv() to read text file
myData = read_tsv("geeksforgeeks.txt", col_names = FALSE)
print(myData)
```

### Output:

```
# A tibble: 1 x 1
  X1
```

```
1 A computer science portal for geeks.
```

Note: You can also use `file.choose()` with `read_tsv()` just like before.

```
# Read a txt file
myData <- read_tsv(file.choose())
```



# Reading one line at a time



`read_lines()`: This method is used for the reading line of your own choice whether it's one or two or ten lines at a time. To use this method we have to import reader package.

Syntax: **`read_lines(file, skip = 0, n_max = -1L)`**

Parameters:

file: file path

skip: Number of lines to skip before reading data

n\_max: Numbers of lines to read. If n is -1, all lines in the file will be read.

```
# R program to read one line at a time
```

```
# Import the readr library  
library(readr)
```

```
# read_lines() to read one line at a time  
myData = read_lines("geeksforgeeks.txt", n_max = 1)  
print(myData)
```

```
# read_lines() to read two line at a time  
myData = read_lines("geeksforgeeks.txt", n_max = 2)  
print(myData)
```

**Output:**

```
[1] "A computer science portal for geeks."
```

```
[1] "A computer science portal for geeks."
```

```
[2] "Geeksforgeeks is founded by Sandeep Jain Sir."
```



# Reading the whole file

**read\_file():** This method is used for reading the whole file. To use this method we have to import reader package.

Syntax: **read\_lines(file)**

file: the file path

```
# R program to read the whole file
```

```
# Import the readr library  
library(readr)
```

```
# read_file() to read the whole file  
myData = read_file("geeksforgeeks.txt")  
print(myData)
```

**Output:**

```
[1] "A computer science portal for geeks.\r\nGeeksforgeeks is founded by  
Sandeep Jain Sir.\r\nI am an intern at this amazing platform."
```





# Reading a file in a table format



Another popular format to store a file is in a tabular format. R provides various methods that one can read data from a tabular formatted data file.

`read.table()`: `read.table()` is a general function that can be used to read a file in table format. The data will be imported as a data frame.

Syntax: `read.table(file, header = FALSE, sep = "", dec = ".")`

Parameters:

**file**: the path to the file containing the data to be imported into R.

**header**: logical value. If TRUE, `read.table()` assumes that your file has a header row, so row 1 is the name of each column. If that's not the case, you can add the argument `header = FALSE`.

**sep**: the field separator character

**dec**: the character used in the file for decimal points.

```
# R program to read a file in table format
```

```
# Using read.table()
```

```
myData = read.table("basic.csv")
```

```
print(myData)
```

**Output:**

```
1 Name,Age,Qualification,Address
```

```
2 Amiya,18,MCA,BBS
```

```
3 Niru,23,Msc,BLS
```

```
4 Debi,23,BCA,SBP
```

```
5 Biku,56,ISC,JJP
```



# read.csv()

**read.csv():** read.csv() is used for reading “comma separated value” files (“.csv”). In this also the data will be imported as a data frame.

Syntax: **read.csv(file, header = TRUE, sep = “,”, dec = “.”, ...)**

Parameters:

**file:** the path to the file containing the data to be imported into R.

**header:** logical value. If TRUE, read.csv() assumes that your file has a header row, so row 1 is the name of each column. If that’s not the case, you can add the argument header = FALSE.

**sep:** the field separator character

**dec:** the character used in the file for decimal points.

```
# R program to read a file in table format
```

```
# Using read.csv()
```

```
myData = read.csv("basic.csv")
```

```
print(myData)
```

Output:

```
Name Age Qualification Address
```

```
1 Amiya 18 MCA BBS
```

```
2 Niru 23 Msc BLS
```

```
3 Debi 23 BCA SBP
```

```
4 Biku 56 ISC JJP
```



# read.csv2()

**read.csv2():** read.csv() is used for variant used in countries that use a comma “,” as decimal point and a semicolon “;” as field separators.

Syntax: **read.csv2(file, header = TRUE, sep = “;”, dec = “,”, ...)**

Parameters:

file: the path to the file containing the data to be imported into R.

header: logical value. If TRUE, read.csv2() assumes that your file has a header row, so row 1 is the name of each column. If that’s not the case, you can add the argument header = FALSE.

sep: the field separator character

dec: the character used in the file for decimal points.

```
# R program to read a file in table format
```

```
# Using read.csv2()
```

```
myData = read.csv2("basic.csv")
```

```
print(myData)
```

Output:

```
Name.Age.Qualification.Address
```

```
1      Amiya,18,MCA,BBS
```

```
2      Niru,23,Msc,BLS
```

```
3      Debi,23,BCA,SBP
```

```
4      Biku,56,ISC,JJP
```



# file.choose()



**file.choose():** You can also use file.choose() with read.csv() just like before.

Example:

```
# R program to read a file in table format
```

```
# Using file.choose() inside read.csv()
```

```
myData = read.csv(file.choose())
```

```
# If you use the code above in RStudio
```

```
# you will be asked to choose a file
```

```
print(myData)
```

Output:

	Name	Age	Qualification	Address
1	Amiya	18	MCA	BBS
2	Niru	23	Msc	BLS
3	Debi	23	BCA	SBP
4	Biku	56	ISC	JJP



## read\_csv()

**read\_csv():** This method is also used for to read a comma (“,”) separated values values by using the help of readr package.

Syntax: read\_csv(file, col\_names = TRUE)

Parameters:

file: the path to the file containing the data to be read into R.

col\_names: Either TRUE, FALSE, or a character vector specifying column names. If TRUE, the first row of the input will be used as the column names.

**# R program to read a file in table format**

**# using readr package**

**# Import the readr library**

**library(readr)**

**# Using read\_csv() method**

**myData = read\_csv("basic.csv", col\_names = TRUE)**

**print(myData)**

Output:

Parsed with column specification:

```
cols(  
  Name = col_character(),  
  Age = col_double(),  
  Qualification = col_character(),  
  Address = col_character()  
)  
# A tibble: 4 x 4  
  Name Age Qualification Address
```

```
1 Amiya 18 MCA BBS  
2 Niru 23 Msc BLS  
3 Debi 23 BCA SBP  
4 Biku 56 ISC JJP
```





# Reading a file from the internet



It's possible to use the functions `read.delim()`, `read.csv()` and `read.table()` to import files from the web.

Example:

```
# R program to read a file from the internet
```

```
# Using read.delim()
```

```
myData
```

```
read.delim("http://www.sthda.com/upload/boxplot_format.txt")
```

```
print(head(myData))
```

Output:

Nom variable Group

1 IND1 10 A

2 IND2 7 A

3 IND3 20 A

4 IND4 14 A

5 IND5 14 A

6 IND6 12 A



# Assessment 1





# References



1. João Moreira, Andre Carvalho, Tomás Horvath – “A General Introduction to Data Analytics” – Wiley -2018
2. <https://www.geeksforgeeks.org/reading-files-in-r-programming/>

**Thank You**