



# **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**

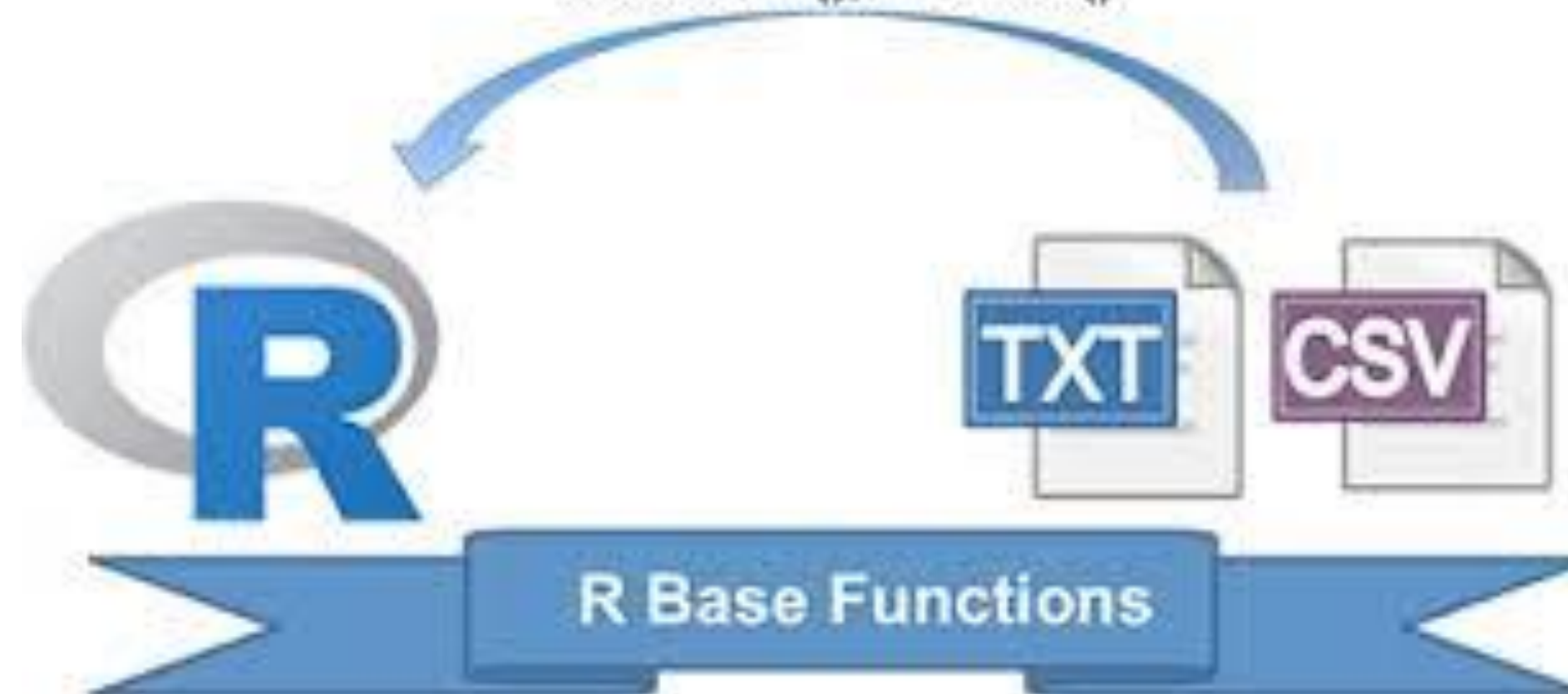
**COURSE NAME :19CS407 DATA ANALYTICS WITH R**  
**II YEAR /IV SEMESTER**

**Unit 5- DATA VISUALIZATION USING R**

**Topic : Using CSV files**

## Import data from txt | csv files into R

`read.delim()`, `read.csv()`





# Using CSV files

- ✓ In R, we can read data from files stored outside the R environment. We can also write data into files which will be stored and accessed by the operating system. R can read and write into various file formats like csv, excel, xml etc.
- ✓ In this chapter we will learn to read data from a csv file and then write data into a csv file. The file should be present in current working directory so that R can read it. Of course we can also set our own directory and read files from there.



# Getting and Setting the Working Directory



- You can check which directory the R workspace is pointing to using the `getwd()` function. You can also set a new working directory using `setwd()` function.

```
# Get and print current working directory.
```

```
print(getwd())
```

```
# Set current working directory.
```

```
setwd("/web/com")
```

```
# Get and print current working directory.
```

```
print(getwd())
```

## **OUTPUT:**

```
[1] "/web/com/1441086124_2016"
```

```
[1] "/web/com"
```



# Input as CSV File



The csv file is a text file in which the values in the columns are separated by a comma. Let's consider the following data present in the file named input.csv.

You can create this file using windows notepad by copying and pasting this data. Save the file as input.csv using the save As All files (\*.\*) option in notepad.

*id,name,salary,start\_date,dept*

*1,Rick,623.3,2012-01-01,IT*

*2,Dan,515.2,2013-09-23,Operations*

*3,Michelle,611,2014-11-15,IT*

*4,Ryan,729,2014-05-11,HR*

*5,Gary,843.25,2015-03-27,Finance*

*6,Nina,578,2013-05-21,IT*

*7,Simon,632.8,2013-07-30,Operations*

*8,Guru,722.5,2014-06-17,Finance*



# Reading a CSV File

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

```
data <- read.csv("input.csv")  
print(data)
```

When we execute the above code, it produces the following result –

	id	name	salary	start_date	dept
1	1	Rick	623.30	2012-01-01	IT
2	2	Dan	515.20	2013-09-23	Operations
3	3	Michelle	611.00	2014-11-15	IT
4	4	Ryan	729.00	2014-05-11	HR
5	NA	Gary	843.25	2015-03-27	Finance
6	6	Nina	578.00	2013-05-21	IT
7	7	Simon	632.80	2013-07-30	Operations
8	8	Guru	722.50	2014-06-17	



# Analyzing the CSV File

By default the `read.csv()` function gives the output as a data frame. This can be easily checked as follows. Also we can check the number of columns and rows.

```
data <- read.csv("input.csv")
print(is.data.frame(data))
print(ncol(data))
print(nrow(data))
```

## OUTPUT:

```
[1] TRUE
[1] 5
[1] 8
```

Once we read data in a data frame, we can apply all the functions applicable to data frames as explained in subsequent section.



# Get the maximum salary



```
# Create a data frame.
```

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

```
# Get the max salary from data frame.
```

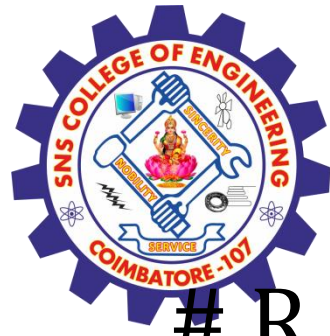
```
sal <- max(data$salary)
```

```
print(sal)
```

OUTPUT

```
[1] 843.25
```





## 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.
```



# Get the details of the person with max salary



We can fetch rows meeting specific filter criteria similar to a SQL where clause.

```
# Create a data frame.
```

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

OUTPUT:

```
id name salary start_date dept  
5 NA Gary 843.25 2015-03-27 Finance
```



# Get all the people working in IT department



*# Create a data frame.*

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

```
retval <- subset( data, dept == "IT")
```

```
print(retval)
```

*OUTPUT*

	<i>id</i>	<i>name</i>	<i>salary</i>	<i>start_date</i>	<i>dept</i>
<i>1</i>	<i>1</i>	<i>Rick</i>	<i>623.3</i>	<i>2012-01-01</i>	<i>IT</i>
<i>3</i>	<i>3</i>	<i>Michelle</i>	<i>611.0</i>	<i>2014-11-15</i>	<i>IT</i>
<i>6</i>	<i>6</i>	<i>Nina</i>	<i>578.0</i>	<i>2013-05-21</i>	<i>IT</i>



# Get the persons in IT department whose salary is greater than 600



```
# Create a data frame.
```

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

```
info <- subset(data, salary > 600 & dept == "IT")
```

```
print(info)
```

OUTPUT

	id	name	salary	start_date	dept
1	1	Rick	623.3	2012-01-01	IT
3	3	Michelle	611.0	2014-11-15	IT



# Get the people who joined on or after 2014



**# Create a data frame.**

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

```
retval <- subset(data, as.Date(start_date) > as.Date("2014-01-01"))
```

```
print(retval)
```

OUTPUT:

```
id name salary start_date dept
3 3 Michelle 611.00 2014-11-15 IT
4 4 Ryan 729.00 2014-05-11 HR
5 NA Gary 843.25 2015-03-27 Finance
8 8 Guru 722.50 2014-06-17 Finance
```



# Writing into a CSV File

R can create csv file from 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")
```

```
retval <- subset(data, as.Date(start_date) > as.Date("2014-01-01"))
```

*# Write filtered data into a new file.*

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

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

```
print(newdata)
```

OUTPUT

X	id	name	salary	start_date	dept
1	3	Michelle	611.00	2014-11-15	IT
2	4	Ryan	729.00	2014-05-11	HR
3	5	NA Gary	843.25	2015-03-27	Finance
4	8	Guru	722.50	2014-06-17	Finance



# Writing into a CSV File



Here the column X comes from the data set newper. This can be dropped using additional parameters while writing the file.

*# Create a data frame.*

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

```
retval <- subset(data, as.Date(start_date) > as.Date("2014-01-01"))
```

*# Write filtered data into a new file.*

```
write.csv(retval,"output.csv", row.names = FALSE)
```

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

```
print(newdata)
```

OUTPUT

```
id name salary start_date dept
1 3 Michelle 611.00 2014-11-15 IT
2 4 Ryan 729.00 2014-05-11 HR
3 NA Gary 843.25 2015-03-27 Finance
4 8 Guru 722.50 2014-06-17 Finance
```



# Assessment 1







# References



1. João Moreira, Andre Carvalho, Tomás Horvath – “A General Introduction to Data Analytics” – Wiley -2018
2. [https://www.tutorialspoint.com/r/r\\_csv\\_files.htm](https://www.tutorialspoint.com/r/r_csv_files.htm)

**Thank You**