

## Lecture 9 - R Software

**Ashutosh Rajput**  
Department of Mathematics

Rajdhani College  
University of Delhi

<https://rfidcard.cards>

You can include the instruction `file.choose()` as part of your `scan()` command.

Note: The `file.choose()` instruction is useful because you can select files from different directories without having to alter the working directory or type the names in full.

This opens a browser-type window where you can navigate to and select the file you want to read:

```
> data7 = scan(file.choose())  
Read 15 items  
> data7  
[1] 23.0 17.0 12.5 11.0 17.0 12.0 14.5 9.0 11.0 9.0 12.5 14.5 17.0 8.0 21.0
```

If you have text or the items are separated by other characters, you use the what = and sep = instructions as appropriate, like so:

```
> data8 = scan(file.choose(), what = 'char', sep = ',')  
Read 12 items  
> data8  
[1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct"  
[12] "Nov" "Dec"  
[13]
```

## The `read.csv()` Command

In most cases, we will have prepared data in a spreadsheet. When you have more complex data it is better to use a new command: `read.csv()`

This command looks for a CSV file and reads the enclosed data into **R**. You can add a variety of additional instructions to the command.

For example: `read.csv(file, sep = ',', header = TRUE, row.names)`

Instructions: • Replace the file with the required filename as before.

- By default the separator is set to a comma but you can alter this if you need to.
- This command expects the data to be in columns, and for each column to have a helpful name. Therefore, the instruction `header = TRUE`, by default reads the first row of the CSV file and sets this as a name for each column.

Note: You can override this with `header = FALSE`.

- The `row.names` part allows you to specify row names for the data; generally this will be a column in the dataset (the first one is most usual and sensible). You can set the row names to be one of the columns by setting `row.names = n`, where `n` is the column number.

Consider, the data table given below:

Name	Protocol	Ports
Load balancer 1	HTTP	80
Load balancer 2	HTTP	80
Load balancer 3	HTTPS	80
Load balancer 4	HTTP	80
Load balancer 5	HTTPS	80

Perform the following steps:

1. If you have a file saved in a proprietary format (for example, XLS), save the data as a CSV file instead.
2. Now assign the file a sensible name and use the `read.csv()` command as follows:

```
> fw = read.csv(file.choose(), header = TRUE)
```

3. The data would be look like

```
> fw
```

Name	Protocol	Ports
Load Balancer 1	HTTP	80
Load Balancer 2	HTTP	80
Load Balancer 3	HTTPS	80
Load Balancer 4	HTTP	80
Load Balancer 5	HTTPS	80