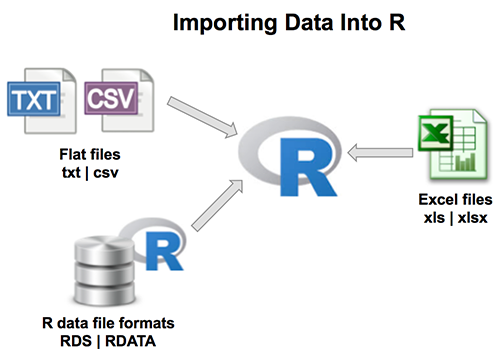
3. Demonstrate Importing and exporting Data.

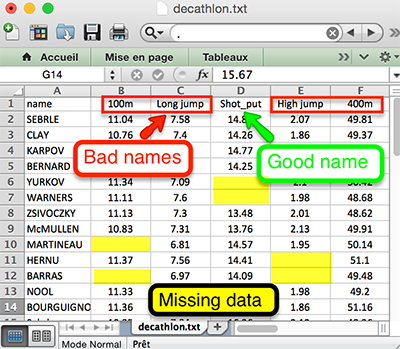
**Importing Data Into R**

Here, you’ll learn how to **import data** from **txt**, **csv**, **Excel** (xls, xlsx) into **R**.

* [Best practices in preparing data files for importing into R](http://www.sthda.com/english/wiki/best-practices-in-preparing-data-files-for-importing-into-r)
* [Reading data from txt|csv files: R base functions](http://www.sthda.com/english/wiki/reading-data-from-txt-csv-files-r-base-functions)
* [Fast Reading of Data From txt|csv Files into R: readr package](http://www.sthda.com/english/wiki/fast-reading-of-data-from-txt-csv-files-into-r-readr-package)
* [Reading data From Excel Files (xls|xlsx) into R](http://www.sthda.com/english/wiki/reading-data-from-excel-files-xls-xlsx-into-r)

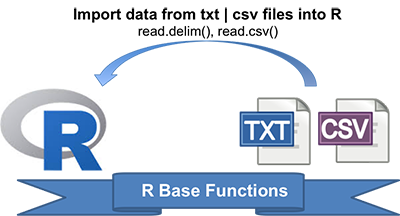


1. [**Best practices in preparing data files for importing into R**](http://www.sthda.com/english/wiki/best-practices-in-preparing-data-files-for-importing-into-r)



1. [**Reading data from txt|csv files: R base functions**](http://www.sthda.com/english/wiki/reading-data-from-txt-csv-files-r-base-functions)

* R base functions for importing data: **read.table**(), **read.delim**(), **read.csv**(), **read.csv2**()
* Reading a local file
* Reading a file from internet



# Read tab separated values

read.delim(file.choose())

# Read comma (",") separated values

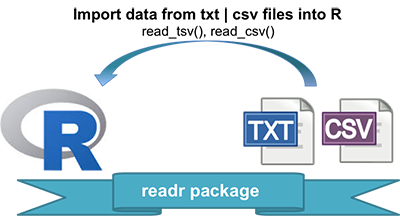
read.csv(file.choose())

# Read semicolon (";") separated values

read.csv2(file.choose())

1. [**Fast Reading of Data From txt|csv Files into R: readr package**](http://www.sthda.com/english/wiki/fast-reading-of-data-from-txt-csv-files-into-r-readr-package)

* Functions for reading txt|csv files: **read\_delim**(), **read\_tsv**(), **read\_csv**(), **read\_csv2**()
* Reading a file
  + Reading a local file
  + Reading a file from internet
  + In the case of parsing problems
* Specify column types
* Reading lines from a file: **read\_lines**()
* Read whole file: **read\_file**()



**library**("readr")

# Read tab separated values

read\_tsv(file.choose())

# Read comma (",") separated values

read\_csv(file.choose())

# Read semicolon (";") separated values

read\_csv2(file.choose())

1. [**Reading data From Excel Files (xls|xlsx) into R**](http://www.sthda.com/english/wiki/reading-data-from-excel-files-xls-xlsx-into-r)

* Copying data from Excel and import into R
* Importing Excel files into R using readxl package
* Importing Excel files using xlsx package



# Use readxl package to read xls|xlsx

**library**("readxl")

my\_data <- read\_excel("my\_file.xlsx")

# Use xlsx package

**library**("xlsx")

my\_data <- read.xlsx("my\_file.xlsx")

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())

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

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

[1] "/web/com"

This result depends on your OS and your current directory where you are working.

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 Finance

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))

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

[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)

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

[1] 843.25

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)

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

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)

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

id name salary start\_date dept

1 1 Rick 623.3 2012-01-01 IT

3 3 Michelle 611.0 2014-11-15 IT

6 6 Nina 578.0 2013-05-21 IT

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)

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

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)

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

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 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")

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)

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

X id name salary start\_date dept

1 3 3 Michelle 611.00 2014-11-15 IT

2 4 4 Ryan 729.00 2014-05-11 HR

3 5 NA Gary 843.25 2015-03-27 Finance

4 8 8 Guru 722.50 2014-06-17 Finance

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)

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

id name salary start\_date dept

1 3 Michelle 611.00 2014-11-15 IT

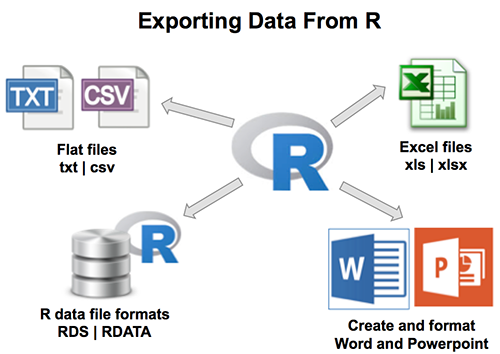
2 4 Ryan 729.00 2014-05-11 HR

3 NA Gary 843.25 2015-03-27 Finance

4 8 Guru 722.50 2014-06-17 Finance

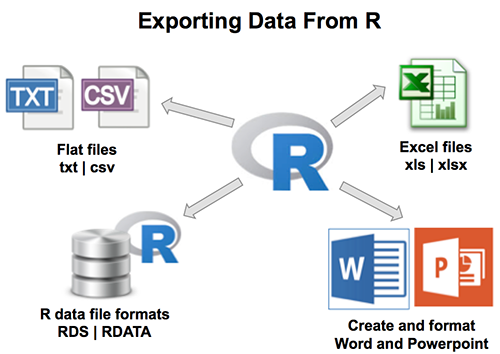
**Exporting data from R**

1. **Export Data From R to txt|csv|Excel files**
   * [Writing data from R to a txt|csv file: R base functions](http://www.sthda.com/english/wiki/writing-data-from-r-to-txt-csv-files-r-base-functions)
   * [Fast Writing of Data From R to txt|csv Files: readr package](http://www.sthda.com/english/wiki/fast-writing-of-data-from-r-to-txt-csv-files-readr-package)
   * [Writing data from R to Excel files (xls|xlsx)](http://www.sthda.com/english/wiki/writing-data-from-r-to-excel-files-xls-xlsx)
   * [Saving data into R data format: RDATA and RDS](http://www.sthda.com/english/wiki/saving-data-into-r-data-format-rds-and-rdata)
2. **Create and format word and powerpoint documents using R and ReporteRs package**:
   * [Create and format Word documents](http://www.sthda.com/english/wiki/create-and-format-word-documents-using-r-software-and-reporters-package)
   * [Create a Word document from a template file](http://www.sthda.com/english/wiki/create-a-word-document-from-a-template-file-using-r-software-and-reporters-package)
   * [Add a table into a Word document](http://www.sthda.com/english/wiki/add-a-table-into-a-word-document-using-r-software-and-reporters-package)
   * [Create and format PowerPoint documents](http://www.sthda.com/english/wiki/create-and-format-powerpoint-documents-from-r-software)
   * [Create an editable graph from R software to PowerPoint](http://www.sthda.com/english/wiki/create-an-editable-graph-from-r-software)



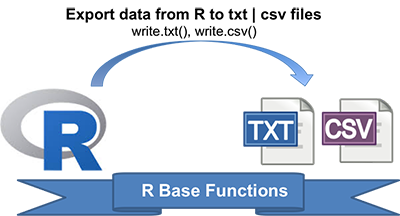
Here, you’ll learn how to **export data** from **R** to **txt**, **csv**, **Excel** (xls, xlsx) and R data file formats. Additionally, we’ll describe how to create and format **Word** and **PowerPoint** documents from R.

1. **Export Data From R to txt|csv|Excel files**
   * [Writing data from R to a txt|csv file: R base functions](http://www.sthda.com/english/wiki/writing-data-from-r-to-txt-csv-files-r-base-functions)
   * [Fast Writing of Data From R to txt|csv Files: readr package](http://www.sthda.com/english/wiki/fast-writing-of-data-from-r-to-txt-csv-files-readr-package)
   * [Writing data from R to Excel files (xls|xlsx)](http://www.sthda.com/english/wiki/writing-data-from-r-to-excel-files-xls-xlsx)
   * [Saving data into R data format: RDATA and RDS](http://www.sthda.com/english/wiki/saving-data-into-r-data-format-rds-and-rdata)
2. **Create and format word and powerpoint documents using R and ReporteRs package**:
   * [Create and format Word documents](http://www.sthda.com/english/wiki/create-and-format-word-documents-using-r-software-and-reporters-package)
   * [Create a Word document from a template file](http://www.sthda.com/english/wiki/create-a-word-document-from-a-template-file-using-r-software-and-reporters-package)
   * [Add a table into a Word document](http://www.sthda.com/english/wiki/add-a-table-into-a-word-document-using-r-software-and-reporters-package)
   * [Create and format PowerPoint documents](http://www.sthda.com/english/wiki/create-and-format-powerpoint-documents-from-r-software)
   * [Create an editable graph from R software to PowerPoint](http://www.sthda.com/english/wiki/create-an-editable-graph-from-r-software)



1. [**Writing data from R to a txt|csv file: R base functions**](http://www.sthda.com/english/wiki/writing-data-from-r-to-txt-csv-files-r-base-functions)

* R base functions for writing data: **write.table**(), **write.csv**(), **write.csv2**()
* Writing data to a file



# Loading mtcars data

data("mtcars")

# Write data to txt file: tab separated values

# sep = "\t"

write.table(mtcars, file = "mtcars.txt", sep = "\t",

row.names = TRUE, col.names = NA)

# Write data to csv files:

# decimal point = "." and value separators = comma (",")

write.csv(mtcars, file = "mtcars.csv")

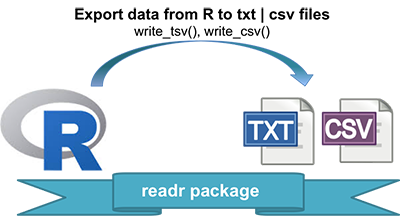
# Write data to csv files:

# decimal point = comma (",") and value separators = semicolon (";")

write.csv2(mtcars, file = "mtcars.csv")

1. **[Fast writing of Data From R to txt|csv Files: readr package](http://www.sthda.com/english/wiki/fast-writing-of-data-from-r-to-txt-csv-files-readr-package)**

* Installing and loading readr: **install.packages**(“readr”)
* readr functions for writing data: **write\_tsv**(), **write\_csv**()
* Writing data to a file



# Loading mtcars data

data("mtcars")

**library**("readr")

# Writing mtcars data to a tsv file

write\_tsv(mtcars, path = "mtcars.txt")

# Writing mtcars data to a csv file

write\_csv(mtcars, path = "mtcars.csv")

1. [**Writing data from R to Excel files (xls|xlsx)**](http://www.sthda.com/english/wiki/writing-data-from-r-to-excel-files-xls-xlsx)

* Installing xlsx package: **install.packages**(“xlsx”)
* Using xlsx package: **write.xlsx**()



**library**("xlsx")

# Write the first data set in a new workbook

write.xlsx(USArrests, file = "myworkbook.xlsx",

sheetName = "USA-ARRESTS", append = FALSE)

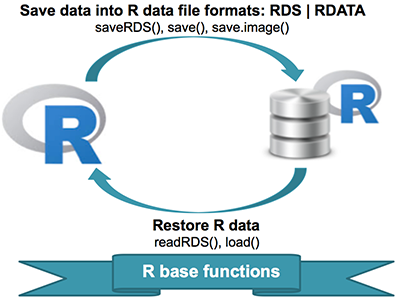
# Add a second data set in a new worksheet

write.xlsx(mtcars, file = "myworkbook.xlsx",

sheetName="MTCARS", append=TRUE)

1. [**Saving data into R data format: RDATA and RDS**](http://www.sthda.com/english/wiki/saving-data-into-r-data-format-rds-and-rdata)

* Save one object to a file: **saveRDS**(object, file), **readRDS**(file)
* Save multiple objects to a file: **save**(data1, data2, file), **load**(file)
* Save your entire workspace: **save.image**(), **load**()



1. Saving and restoring one single R object:

# Save a single object to a file

saveRDS(mtcars, "mtcars.rds")

# Restore it under a different name

my\_data <- readRDS("mtcars.rds")

1. Saving and restoring one or more R objects:

# Save multiple objects

save(data1, data2, file = "data.RData")

# To load the data again

load("data.RData")

1. Saving and restoring your entire [workspace](http://www.sthda.com/wiki/(unning-rstudio-and-setting-up-your-working-directory-easy-r-programming#use-r-inside-rstudio)):

# Save your workspace

save.image(file = "my\_work\_space.RData")

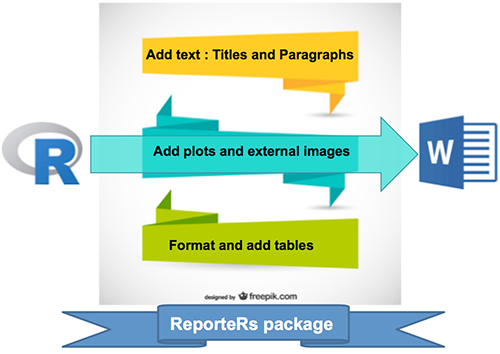
# Load the workspace again

load("my\_work\_space.RData")

1. [**Create and format Word documents with R and ReporteRs package**](http://www.sthda.com/english/wiki/create-and-format-word-documents-using-r-software-and-reporters-package)

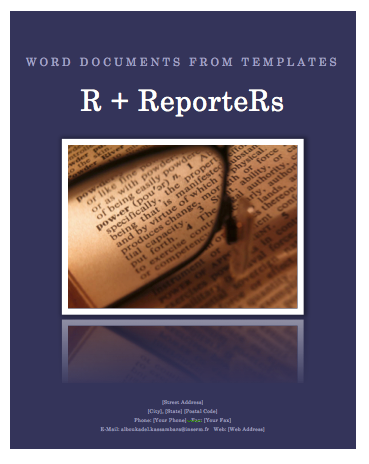
**ReporteRs** package, by David Gohel, provides easy to use functions to **write** and **format** **Word documents**. It can be also used to generate **Word document from a template file** with logos, fonts, etc. ReporteRs is Java-based solution, so it works on Windows, Linux and Mac OS systems.

* Install and load the ReporteRs R package
* Create a simple Word document
  + Add texts : title and paragraphs of texts
  + Format the text of a Word document using R software
  + Add plots and images
  + Add a table
  + Add lists : ordered and unordered lists
  + Add a footnote to a Word document
  + Add R scripts
* Add a table of contents into a Word document



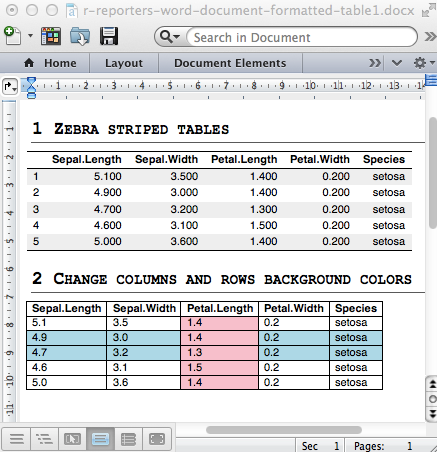
1. [**Create a Word document from a template file with R and ReporteRs package**](http://www.sthda.com/english/wiki/create-a-word-document-from-a-template-file-using-r-software-and-reporters-package)

* Quick introduction to ReporteRs package
* Create a Word document using a template file



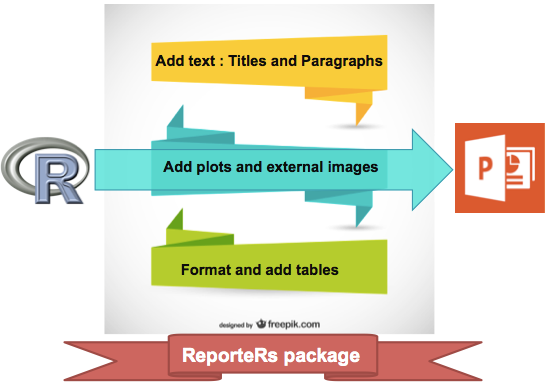
1. [**Add a table into a Word document with R and ReporteRs package**](http://www.sthda.com/english/wiki/add-a-table-into-a-word-document-using-r-software-and-reporters-package)

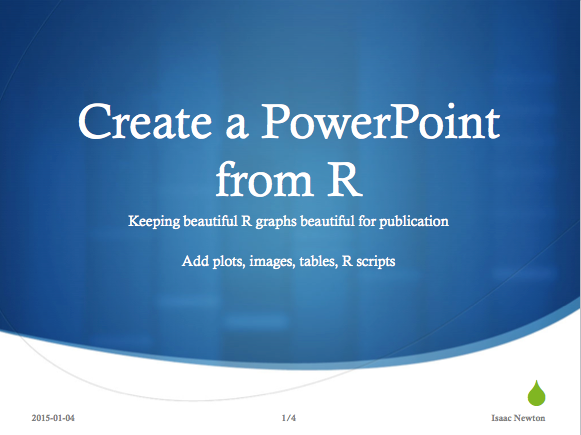
* Add a simple table
* Add a formatted table
  + Change the background colors of rows and columns
  + Change cell background and text colors
  + Insert content into a table : header and footer rows
* Analyze, format and export a correlation matrix into a Word document
* Powerpoint



1. [**Create and format PowerPoint documents with R and ReporteRs**](http://www.sthda.com/english/wiki/create-and-format-powerpoint-documents-from-r-software)

* Why is it important to be able to generate a PowerPoint report from R ?
  + Reason I : Many collaborators works with Microsoft office tools
  + Reason II : keeping beautiful R graphs beautiful for publications
* Install and load the ReporteRs package
* Create a simple PowerPoint document
  + Slide layout
  + Generate a simple PowerPoint document from R software
  + Format the text of a PowerPoint document
  + Add plots and images
  + Add a table
  + Add ordered and unordered lists
* Create a PowerPoint document from a template file





1. [**Create an editable graph from R to PowerPoint**](http://www.sthda.com/english/wiki/create-an-editable-graph-from-r-software)

* Case of base graphs
* Case of graphs generated using ggplot2

