SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)
Approved by AICTE, New Delhi, Affiliated to Anna University, Chennai
Accredited by NAAC-UGC with 'A++' Grade (Cycle III) &
Accredited by NBA (B.E - CSE, EEE, ECE, Mech & B.Tech.IT)
COIMBATORE-641 035, TAMIL NADU

UNIT II

Importing and Exporting Data in R

First, make sure that you are working in the right directory- the directory that you want to be. To see the current working directory, type getwd() into the console

```
getwd()
## [1] "/Users/Ercan/Desktop/R Books/RInto_Ercan"
```

You can change the working directory by typing setwd():

```
setwd("/Users/Ercan/Desktop/Beginning R")

getwd()
```

Alternatively, just select Session > Change Working Directory in Rstudio and navigate to the directory you want to make your new working directory.

Finally, type in dir() to see all the documents in the current directory.

The definitive guide for importing data in R is the R Data Import/Export manual available at R Data Import/Export manual(https://cran.r-project.org/doc/manuals/R-data.pdf)

Importing .csv, .txt, .delim files

Read .csv format (comma separated values)

To read data from a CSV file use either read.table or read.csv functions, the latter being a wrapper around the former. Suppose that you have a data set health.csv in *your working directory* and you want to introduce this to R:

```
# use read.table
health <- read.table(file = "health.csv", header = TRUE, sep = ",")
# or use read.csv
health <- read.csv("health.csv")
# the result is a date.frame
class(health)
```

```
## [1] "data.frame"
roster <- read.csv("roster.csv")
```

To learn about the content of a data frame:

```
str(roster)
## 'data.frame':    13 obs. of  6 variables:
##  $ Jersey  : int  0 1 3 5 10 12 15 20 21 33 ...
##  $ Name    : Factor w/ 13 levels "Ajukwa, Austin",..: 11 1 8 2 3 6 5 12 7 13 ...
##  $ Position: Factor w/ 3 levels "C","F","G": 3 3 3 2 3 3 2 3 3 2 ...
##  $ Inches  : int  74 78 74 79 75 73 80 72 76 80 ...
##  $ Pounds  : int  190 205 205 215 200 205 205 165 205 245 ...
##  $ Class   : Factor w/ 4 levels "freshman","junior",..: 1 4 2 4 1 3 1 2 3 2 ...
```

To view your data without editing them, you can use the View command:

```
View(roster)
```

Read .txt format

Suppose that you have a data set health.txt in *your working directory* and you want read this into R:

```
read.table("health.txt", sep = " ")
```

Importing .xls, xlsx, .sav, .sas Files

- Use the package readxl to read in **.xls** and **.xlsx** files.

```
install.packages("readxl")
library(readxl)
read_excel("health.xlsx")
```

- Use the package haven to read in **SPSS**, **Stata** and **SAS** files

```
install.packages("haven")
library(haven)
read_spss("Dataset.sav")
read_dta("Dataset.dta")
read_sas("Dataset.sas7bdat")
```

Importing Data from the Web

Suppose there is some data at `http://www.something.com/data.csv’ We can import this as

```
getLink <- "http://www.something.com/data.csv"


myData <- read.table(file = getLink, header = TRUE, sep = ",")
```

Unlike read.table, read_excel cannot read data directly from the Internet, and thus the files must be downloaded first. We could do this by visiting a browser or we can stay within R and use download.file.

```
# download the data.
# Note that this will download to the current wd, but you can change it by specifying a path to "destfile"
download.file(url='http://www.something.com/data.xlsx',
            destfile='excelData.xlsx', method='curl')
# read data
excelData <- read_excel("excelData.xlsx")
```

R Binary Files

When working with other R programmers, a good way to pass around data-or any R objects such as variables and functions—is to use RData files. These are binary files that represent R objects of any kind. They can store a single object or multiple objects and can be passed among Windows, Mac and Linux without a problem.

First, let’s create an RData file, remove the object that created it and then read it back into R:

```
health <- read.table(file = "health.csv", header = TRUE, sep = ",")
# save the health.frame to disk
save(health, file="health.rdata")
# remove health from memory
rm(health)
# read it from the rdata file
load("health.rdata")
# check if it exists now
head(health)
##   id gender state age health1 health2 health3 health4 health5 health6
## 1  1      M     1  51       1       4       2       1       4       5
## 2  2      F     3  35       2       3       3       2       3       4
## 3  3      F     1  29       5       2       4       2       1       3
## 4  4      M     1  21       5       1       5       4       2       1
## 5  5      M     2  56       2       4       2       4       3       3
## 6  6      M     3  72       1       5       4       2       4       5
```