**IMPORT DATA FROM EXCEL**

**(Sources: https://www.datacamp.com/community/tutorials/r-tutorial-read-excel-into-r)**

### **Best Practices**

* Before loading your Excel files and spreadsheets into R, make sure that your data is well prepared to be imported. If you would neglect to do this, you might experience problems when using the R functions later.

Here’s a list of some best practices to help avoid any issues with reading your Excel files and spreadsheets into R:

* The first row of the spreadsheet is usually reserved for the header, while the first column is used to identify the sampling unit;
* Avoid names, values or fields with blank spaces. Otherwise, each word will be interpreted as a separate variable, resulting in errors that are related to the number of elements per line in your data set;
* If you want to concatenate words, do this by inserting a “.”. For example: Working.Time
* Short names are preferred over longer names;
* Try to avoid using names that contain symbols such as ?, $,%, ^, &, \*, (, ), -, #, ?, ,, <, >, /, |, \, [ ,] ,{, and };
* Delete any comments that you have made in your Excel file to avoid extra columns or NA's to be added to your file; and
* Make sure that any missing values in your data set are indicated with NA

### **Saving Your Data**

Make sure that your data is saved in Excel. This allows you to revisit the data later to edit, to add more data or to change them, preserving the formulas that may be used to calculate the data, etc.

Microsoft Excel offers many options to save your file: besides the default extension .xls or .xlsx, you can go to the File tab, click on “Save As” and select one of the extensions that are listed as the “Save as Type” options.

The most common extensions to save datasets are .csv and .txt (as tab-delimited text file). Depending on the saving option that you choose, your data set’s fields are separated by commas or tabs. These symbols are then called the “*field separator characters*” of your data set.

## Preparatory Work in R

You may need to set your working directory in R. To do this, try to find the current working directory at the moment by use of the command: getwd()

Then, change the path so that it includes the folder where you have stored your dataset:

 setwd("<location of your dataset>")

For example:

setwd("D:/Program Problem/EU Plus Project/WP3/Course 9 \_ Data Analytic/Power Point/Data")

By executing this command, R now knows exactly in which folder you’re working.

Let build this file “Cost\_Time of Engine Tune-up” in folder named “Data” using Excel and save it in .csv and .txt formats

|  |  |  |
| --- | --- | --- |
| **Invoice** | **Cost** | **Working Time** |
| 1 | 91 | 38 |
| 2 | 71 | 32 |
| 3 | 104 | 33 |
| 4 | 85 | 38 |
| 5 | 62 | 37 |
| 6 | 78 | 30 |
| 7 | 69 | 32 |
| 8 | 74 | 32 |
| 9 | 97 | 34 |
| 10 | 82 | 34 |
| 11 | 93 | 32 |
| 12 | 72 | 33 |
| 13 | 62 | 39 |
| 14 | 88 | 37 |
| 15 | 98 | 31 |
| 16 | 57 | 37 |
| 17 | 89 | 34 |
| 18 | 68 | 38 |
| 19 | 68 | 40 |
| 20 | 101 | 32 |
| 21 | 75 | 33 |
| 22 | 66 | 33 |
| 23 | 97 | 36 |
| 24 | 83 | 39 |
| 25 | 79 | 38 |
| 26 | 52 | 32 |
| 27 | 75 | 35 |
| 28 | 105 | 33 |
| 29 | 68 | 37 |
| 30 | 105 | 37 |
| 31 | 99 | 39 |
| 32 | 79 | 38 |
| 33 | 77 | 39 |
| 34 | 71 | 39 |
| 35 | 79 | 37 |
| 36 | 80 | 39 |
| 37 | 75 | 32 |
| 38 | 65 | 38 |
| 39 | 69 | 34 |
| 40 | 69 | 38 |
| 41 | 97 | 38 |
| 42 | 72 | 36 |
| 43 | 80 | 32 |
| 44 | 67 | 33 |
| 45 | 62 | 32 |
| 46 | 62 | 32 |
| 47 | 76 | 39 |
| 48 | 109 | 38 |
| 49 | 74 | 31 |
| 50 | 73 | 37 |

## Loading your Spreadsheets and Files into R

### **Basic R Commands**

The following commands are all part of R’s Utils package, which is one of the core and built-in packages that contains a collection of utility functions. You will see that these basic functions focus on getting Excel spreadsheets into R, rather than the Excel files themselves.

#### **read.table( ):**

Used to import tab-delimited text file or \*.txt file

Example: read the file and assign to the data frame named “df”

 df <- read.table("Cost\_Time of Engine Tune-up.txt", header = TRUE)

If you did not change working directory:

df <- read.table("D:/Program Problem/EU Plus Project/WP3/Course 9 \_ Data Analytic/Power Point/Data/Cost\_Time of Engine Tune-up.txt", header=TRUE)

Then you can do some operation with each vector (i.e., column) in “df”. For example:

mean(df$Cost): find the mean of “Cost” vector in “df”

Note: if your data set did not have a header, R will provide some attributes for it, namely V1, V2, V3, V4, and V5

#### **read.csv( ) and read.csv2( ):**

The read.csv() and read.csv2() are used to read in .csv format. As described before, read.csv() is used when the separator is a comma while read.csv2() is used when the separator is a semicolon.

Tip: find out what separator symbol is used in your .csv file by opening it in a text editor such as Notepad.

### **Using “Import Dataset” in RStudio (look at top-right corner or use “File” menu)**

For .txt file: use “From Text (base)”: it is better to untick the box “Strings as factors” in the wizard (see pages 49, 50 of the book “Hands-on Programming with R” to understand factors, and page 59 to understand why strings should not be converted to factors automatically)

For .csv file: use “From Text(readr)”

For .xlsx file: use “From Excel”