Introduction to R programming for data science – day 6

Dr. Francesca Finotello
Medical University of Innsbruck, Austria

Read and write text files

Reading tables with *read.table*

Plain text files in table format can be loaded into R as data.frames using the function read.table

```
read.table(file,             # Input file
    header = FALSE,          # Use first row as column names?
    sep = "",                # Column separator in the input file
    row.names = 1,           # Column containing the row names
    nrows = -1,              # Number of rows to be read
    skip = 0,                # Number of rows to be skipped
    check.names = TRUE,      # Check and fix column names?
                             # (e.g. "123-A" --> "X123.A")
    stringsAsFactors = FALSE, # Save strings as factors?
    ...)
```

Once you load a table into R, you can check how it looks like using head and tail, and check its dimensions with dim

Reading tables with *read.delim* and *read.csv*

Depending on the column separator in the input file, read.delim and read.csv can be also used to import files in table format

```
read.delim(file,
      header = TRUE,
      sep = "\t", ...)

read.csv(file,
      header = TRUE,
      sep = ",", ...)
```

The are both based on the read.table function, but they use different paremeter settings (e.g. “sep”)

Save data in plain text files

Data.frames and matrices can be saved into text files

```
( DF <- data.frame(name=c("Mary", "John", "Lisa"),
  age=c(19, 30, 20),
  city=c("New York", "Seattle", "New York")) )
```

```
##   name age     city
## 1 Mary  19 New York
## 2 John  30  Seattle
## 3 Lisa  20 New York
```

```
write.table(DF,
  quote = FALSE,
  sep = "\t",
  row.names = FALSE,
  col.names = TRUE,
  file =  "../Data/Day3_Friends_table.txt")
```


Read and write Excel files

The *xlsx* package

The xlsx package provides R functions to handle Excel files (97/2000/XP/2003/2007 formats)

It is available on CRAN and can be installed from the "Packages" window or by executing

```
install.packages("xlsx")
```

Once installed, it can be loaded with library, which is the function to load (installed) R packages into R

```
library("xlsx"")
```


Read Excel files

The read.xlsx function from the xlsx package can be used to read Excel files

```
read.xlsx(file,           # File to be read
  sheetIndex,             # Number of the sheet to be read
  sheetName = NULL,       # Character indicating the sheet name
  startRow = NULL,        # First row to be read
  endRow = NULL,          # Last row to be read
  header = TRUE,          # Does the first row contain column names?
  keepFormulas = FALSE,   # Display formulae (instead of results)?
  ...)
```

Write Excel files

The write.xlsx function from the xlsx package can be used to write Excel files

```
write.xlsx(x,             # Table to be written
  file,                   # Path to the output file
  sheetName = "Sheet1",   # Character indicating the sheet name
  col.names = TRUE,       # Write column names?
  row.names = TRUE,       # Write row names?
  ...)
```


Excel and gene names

Gene name errors are widespread in the scientific literature

Mark Ziemann, Yotam Eren and Assam El-Osta

Genome Biology 2016 **17**:177
https://doi.org/10.1186/s13059-016-1044-7
Published: 23 August 2016

Abstract

The spreadsheet software Microsoft Excel, when used with default settings, is known to convert gene names to dates and floating-point numbers. A programmatic scan of leading genomics journals reveals that approximately one-fifth of papers with supplementary Excel gene lists contain erroneous gene name conversions.

The *readr* package

The readr package from tidyverse provides functions to read files with different formats:

- read_csv: comma separated (CSV) files
- read_tsv: tab separated files
- read_delim: general delimited files
- read_fwf: fixed width files
- read_table: tabular files where columns are separated by white-space
- read_log: web log files