

L02: Tabulated data I/O

“Clu, we don't have much time left to find that file. This is top priority.”

Kevin Flynn, Tron

Things we cover in this session

- Reading tabulated data sets
- Accessing data values in data frames

Things you need for this session

- [W02-1: Reading CSV files](#)

Things to take home from this session

At the end of this session you should be able to

- read data from CSV files
- access individual cells, entire columns or rows of data frames

CSV I/O

In general, reading or writing from/to structured files with R is quite simple and straight forward. While other programming languages often require the explicit opening and closing of a file, the respective R functions called `read.table` and `write.table` are an all-round carefree package. As a result, these functions return a variable of type data frame which you can think of as a table in Excel or any other spreadsheet software.

🤖 Have a look at [C02-1 CSV I/O](#) now for a hands-on introduction.

For more information, you can also search the web, refer to Quick-R's [importing](#) and [exporting](#) data section or type `?read.table` or `?write.table` inside your R environment.

R has a variety of packages for reading and writing specific data sets. Before you start programming your own I/O routine, search the internet like “R reading XY files”

Accessing or modifying values in data frames

You already have noticed that when reading tabulated data into R, the variable in which the data is stored is of type data frame. In turn, each column of the data frame is of a certain data type.

To access individual values in R data structures you have to specify the section inside the respective

variable where the desired value(s) are stored. This sounds much more complicated than it is.

Imagine a table with a single row. Such a row would be a graphical representation of a vector in R. If you are interested into the third value, you will have a look into the third column. The same is true for R: you access the third value of a vector v by selecting it within square brackets like $v[3]$. For other data structures the same logic applies, also it might be necessary to use double brackets combined with single ones (e.g. $l[3]$ gives you the third value in the simple list l) or supply more than one dimension (e.g. $df[3,2]$ gives you the value of the third row and second column inside the data frame df).

🤔 Have a look at [C00-2 Data frame basics](#) now for a hands-on introduction for data frames.

In [C00-1 Vector basics](#) you find the corresponding explanations for vectors if you need them.

🤔 Introducing R data types and structures in more detail is beyond the scope of this school. For some more information on that topic, please refer to the excursus [E02-1: Data types and structures](#).

Time for practice

[W02-1: Reading CSV files](#)

From: <http://bisfogo.environmentalinformatics-marburg.de/> - **BIS-Fogo**

Permanent link: <http://bisfogo.environmentalinformatics-marburg.de/doku.php?id=en:learning:schools:s01:lecture-notes:ba-ln-02>

Last update: **2015/09/22 15:22**

