R data I/O

Alberto Garfagnini

Università di Padova

R lecture 6

Data Input

- numbers can be inputed through the keyboard, from the Clipboard, from an external file on disk, or from an external file on the Web
- use the concatenate function for up to 10 numbers
- and scan() for typing or pasting data into a vector

```
y <- c (6,7,3,4,8,5,6,2)

tu <- scan()
%> 1: 6
%> 2: 3
%> 3: 4
%> 4: 2
%> 5:
%> Read 4 items
tu
%> [1] 6 3 4 2
```

- but the easiest way is to read data from a file (or from the Web), already shaped in a data frame format

Data Input using read.table()

- the read.table() function reads data from a local file and creates a data frame

```
data <- read.table("yield.txt",header=T)

data
%>    year wheat barley oats rye corn
%> 1  1980   5.9    4.4  4.1 3.8  4.4
%> 2  1981   5.8    4.4  4.3 3.7  4.1
%> 3  1982   6.2    4.9  4.4 4.1  4.0
...
%> 27 2006   8.0    5.9  6.0 6.1  4.5
%> 28 2007   7.2    5.7  5.5 5.7  3.9
%> 29 2008   8.3    6.0  5.8 6.1  4.4
```

- the parameter header = T tells R to use the first row as column names

```
names(data)
%> [1] "year"   "wheat"  "barley" "oats"   "rye"    "corn"

str(data)
%> 'data.frame':        29 obs. of  6 variables:
%>  $ year  : int  1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 ..
%>  $ wheat : num  5.9 5.8 6.2 6.4 7.7 6.3 7 6 6.2 6.7 ...
%>  $ barley: num  4.4 4.4 4.9 4.7 5.6 5 5.2 5 4.7 4.9 ...
%>  $ oats  : num  4.1 4.3 4.4 4.3 4.9 4.6 5.2 4.6 4.6 4.5 ...
%>  $ rye   : num  3.8 3.7 4.1 3.7 4.7 4.6 4.7 4.8 4.6 4.8 ...
%>  $ corn  : num  4.4 4.1 4 4.1 4.7 4.3 4.3 4.5 4.2 3.8 ...
```

Data Input using read.table()

- if the separator between variable names and data fields are not blanks or tabs, (\t), a different separator can be specified with the sep="," option

```
datav <- read.table("bowens.csv",sep=",", header=T)

str(datav)
%> 'data.frame':        733 obs. of  3 variables:
%>  $ place: Factor w/ 727 levels "Abingdon","Admoor␣Copse",..: 1 2 3 .
%>  $ east : int  50 60 48 70 59 60 60 59 61 60 ...
%>  $ north: int  97 70 87 73 65 65 63 66 63 67 ...
```

read.table() : separators and decimal points

- the default field separator character in read.table() is sep=" ": which identifies with one or more spaces, one or more tabs (\t), and one or more newlines (\n)
- for comma-separated fields use read.csv()
- for semicolon-separated fields use read.csv2()
- for tab-delimited fields with decimal points as a commas, use read.delim2()

```
File: bowens.csv
--------------------
|place,east,north  |
|Abingdon,50,97    |
|Admoor Copse,60,70|
|...               |
|Youlbury,48,3     |
--------------------

str(bw)
%> 'data.frame':   733 obs. of  3 variables:
%>  $ place: Factor w/ 727 levels "AERE␣Harwell",..: 2 3 1 4 5 ...
%>  $ east : int  50 60 48 70 59 60 60 59 61 60 ...
%>  $ north: int  97 70 87 73 65 65 63 66 63 67 ...
```

read.csv() and read.delim()

- additional functions to read a file in table format exist

```
> ?read.table
...
read.delim(file, header = TRUE, sep = "\t", quote = "\"",
␣␣␣␣␣␣␣␣␣␣␣dec␣=␣".",␣fill␣=␣TRUE,␣comment.char␣=␣"",␣...)
...
read.csv(file,␣header␣=␣TRUE,␣sep␣=␣",",␣quote␣=␣"\"",
         dec = ".", fill = TRUE, comment.char = "", ...)
...
read.csv2(file, header = TRUE, sep = ";", quote = "\"",
␣␣␣␣␣␣␣␣␣␣dec␣=␣",",␣fill␣=␣TRUE,␣comment.char␣=␣"",␣...)
...
read.delim2(file,␣header␣=␣TRUE,␣sep␣=␣"\t",␣quote␣=␣"\"",
            dec = ",", fill = TRUE, comment.char = "", ...)
```

- further detailed instructions in the 'R Data Import/Export' manual:

  https://cran.r-project.org/doc/manuals/r-release/R-data.html

The readr package

- is part of the core Tidyverse
- readr supports seven file formats with seven read_ functions:

- 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

```
readr::read_delim("yield.txt", delim='\t')
%- Column specification -----------------------
cols(
  year = col_double(),
  wheat = col_double(),
  barley = col_double(),
  oats = col_double(),
  rye = col_double(),
  corn = col_double()
)
```

https://readr.tidyverse.org/

Data Input from the Web and from DB

- R can read data form the network using HTTP by specifying the file URL

```
wc <- read.table("https://tinyurl.com/murders-txt", header=T)

str(wc)
%> 'data.frame':   50 obs. of  4 variables:
%>  $ state      : Factor w/ 50 levels "Alabama","Alaska",..: 1 2 ...
%>  $ population: int  3615 365 2212 2110 21198 2541 3100 ...
%>  $ murder     : num  15.1 11.3 7.8 10.1 10.3 6.8 3.1 6.2 ...
%>  $ region     : Factor w/ 4 levels "North.Central",..: 3 4 4 ...
```

- several packages available on CRAN to help R communicate with DBMSs:

  combining a unified 'front-end' package with a 'back-end' module, several common relational databases can be accessed (RMySQL, ROracle, RPostgreSQL and RSQLite)

- finally, R can read binary data files: NASA's HDF5 (Hierarchical Data Format, https://www.hdfgroup.org/HDF5/) and UCAR's netCDF data files (network Common Data Form, http://www.unidata.ucar.edu/software/netcdf/)

- and image files

Example: data Input from the Web

- let's retrieve the latest data on the COVID-19 Virus infection from the European Centers for Disease Control https://www.ecdc.europa.eu/en
- R can read data form the network using HTTP by specifying the file URL

European Centre for Disease Prevention and Control

An agency of the European Union

All topics: A to Z   News & events   Publications & data   Tools   About us

Coronavirus disease

The disease is rapidly spreading worldwide and the number of cases in Europe is rising with increasing pace in several affected areas.

Latest information on COVID-19 ▸

Coronavirus disease (COVID-19)   COVID-19: Social distancing measures   Antimicrobial resistance in zoonoses   2019/2020 influenza season

COVID-19

Several countries are now experiencing sustained local transmission of coronavirus disease 2019 (COVID-19), including Europe.

The COVID-19 pandemic is rapidly evolving, and outbreak investigations are ongoing. ECDC is closely monitoring this outbreak, providing risk assessments, public health guidance, and advice on response activities to EU Member States and the EU Commission.

- Latest situation update, epidemiological curve and global distribution

Example: data Input from the Web

- we download an EXCEL file
- we use the following packages: lubridate, curl and readxl

```
url <- "https://opendata.ecdc.europa.eu/"
fname <- "covid19/nationalcasedeath_eueea_daily_ei/xlsx"
target <- paste(url, fname, sep="")
message("target:", target)

tmp_file <- tempfile("data", "/tmp", fileext=ext)

tmp <- curl::curl_download(target , destfile=tmp_file)

data <- readxl::read_xlsx(tmp_file)
```

- data are imported in a tibble data structure

```
(data <- readxl::read_xlsx(tmp_file))
%> A tibble: 6,012 x 8
%>   DateRep               Day Month  Year Cases Deaths Countries. GeoId
%>   <dttm>              <dbl> <dbl> <dbl> <dbl>  <dbl> <chr>      <chr>
%> 1 2020-03-21 00:00:00    21     3  2020     2      0 Afghanistan   AF
%> 2 2020-03-20 00:00:00    20     3  2020     0      0 Afghanistan   AF
%> 3 2020-03-19 00:00:00    19     3  2020     0      0 Afghanistan   AF
%> 4 2020-03-18 00:00:00    18     3  2020     1      0 Afghanistan   AF
%> 5 2020-03-17 00:00:00    17     3  2020     5      0 Afghanistan   AF
%> 6 2020-03-16 00:00:00    16     3  2020     6      0 Afghanistan   AF
%> 7 2020-03-15 00:00:00    15     3  2020     3      0 Afghanistan   AF
%> 8 2020-03-11 00:00:00    11     3  2020     3      0 Afghanistan   AF
%> 9 2020-03-08 00:00:00     8     3  2020     3      0 Afghanistan   AF
%>10 2020-03-02 00:00:00     2     3  2020     0      0 Afghanistan   AF
% ... with 6,002 more rows
```