## R TUTORIAL - VTT (Lecture 1)

Oguzhan (Ouz) Gencoglu

VTT Technical Research Center of Finland

27 April 2016

#### Outline

- 1 What is R? Why/Why not R? Practical Issues.
  - What exactly is this R & RStudio thing?
  - The Good, The Bad & the Ugly (Sides of R)
  - Practical Issues
- 2 Basic Examples
  - Assignment, Lists, Arrays etc.
  - Data Frames & Data Wrangling
  - 'apply' functions

### Outline

- 1 What is R? Why/Why not R? Practical Issues.
  - What exactly is this R & RStudio thing?
  - The Good, The Bad & the Ugly (Sides of R)
  - Practical Issues
- 2 Basic Examples
  - Assignment, Lists, Arrays etc.
  - Data Frames & Data Wrangling
  - 'apply' functions

What exactly is this R thing?

R is a language and environment for statistical computing and graphics. It provides a wide variety of statistical (linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is highly extensible.

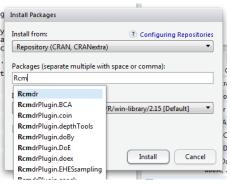


#### Outline

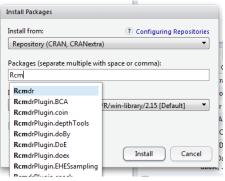
- 1 What is R? Why/Why not R? Practical Issues.
  - What exactly is this R & RStudio thing?
  - The Good, The Bad & the Ugly (Sides of R)
  - Practical Issues
- 2 Basic Examples
  - Assignment, Lists, Arrays etc.
  - Data Frames & Data Wrangling
  - 'apply' functions

The Good, The Bad & the Ugly (Sides of R)

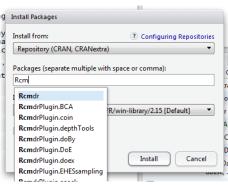
Developed by statisticians & scientists



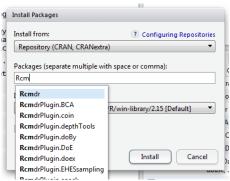
- Developed by statisticians & scientists
- Free (Speech & Sandwich)<sup>g</sup>



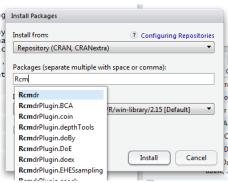
- Developed by statisticians & scientists
- Free (Speech & Sandwich)<sup>9</sup>
- Most comprehensive statistical analysis environment



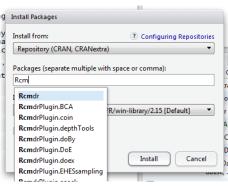
- Developed by statisticians & scientists
- Free (Speech & Sandwich)<sup>g</sup>
- Most comprehensive statistical analysis environment
- Outstanding graphical capabilities



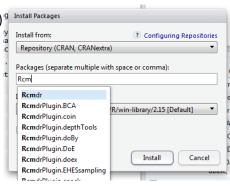
- Developed by statisticians & scientists
- Free (Speech & Sandwich)<sup>9</sup>
- Most comprehensive statistical analysis environment
- Outstanding graphical capabilities
- Almost 5000 packages



- Developed by statisticians & scientists
- Free (Speech & Sandwich)<sup>g</sup>
- Most comprehensive statistical analysis environment
- Outstanding graphical capabilities
- Almost 5000 packages
- Cross-platform



- Developed by statisticians & scientists
- Free (Speech & Sandwich)<sup>g</sup>
- Most comprehensive statistical analysis environment
- Outstanding graphical capabilities
- Almost 5000 packages
- Cross-platform
- Active user group



The Good, The Bad & the Ugly (Sides of R)

② Developed by statisticians& scientists

```
<?xel version="1.0" encoding="UTF-8"?>
although it does not require it.
The XML package provides general facilities for reading and writing XML documents within R. A description of the facilities of
details and examples. Package StatDataML on CRAN is one example building on XML
NB: XML is available as a binary package for Windows, normally from the "CRAN extras" repository (which is selected by def
Next: Importing from other statistical systems, Previous: Introduction, Up: Top. [Contents][Index]
                                                                                                            2 Spreadshe
· Variations on read table
. Fixed-width-format files

    Data Interchange Format (DIF):

· Re-shaping data:
· Flat contingency table
In Export to text file, we saw a number of variations on the format of a spreadsheet-like text file, in which the data are presented
Next: Fixed-width-format files, Previous: Spreadsheet-like data. Up: Spreadsheet-like data. [Contents][Index]
2.1 Variations on read.table
The function read, table is the most convenient way to read in a rectangular grid of data. Because of the many possibilities, ther
Beware that read, table is an inefficient way to read in very large numerical matrices; see scan below.
Some of the issues to consider are
      If the file contains non-ASCII character fields, ensure that it is read in the correct encoding. This is mainly an issue for re-
             read.table("file.dat", filetocoding="latin1")
      Note that this will work in any locale which can represent Latin-1 strings, but not many Greek-Russian/Chinese/Japanese
      We recommend that you specify the teaster argument explicitly. Conventionally the header line has entries only for the co
      presented with a file that has a (possibly empty) header field for the row labels, read it in by something like
              read table("file.det", header a TRUE, row.names a 1)
      Column names can be given emplicitly via the col. cases: explicit names override the header line (if present)
   3 Separator
      Normally looking at the file will determine the field separator to be used, but with white-space separated files there may b
      "\e". Note that the choice of separator affects the input of quoted strings.
      If you have a tab-delimited file containing empty fields be sure to use sep = "\e".
      By default character strings can be quoted by either '*' or '.', and in each case all the characters up to a matching quote as
      argument. For sep = "\n" the default is changed to quote = ""
      If no separator character is specified, quotes can be escaped within quoted strings by immediately preceding them by '\','.
      If a separator character is specified, quotes can be escaped within quoted strings by doubling them as is conventional in sp
```

The Good, The Bad & the Ugly (Sides of R)

- Developed by statisticians
   & scientists
- ② Learning Curve

(bull version\*1.4" encodings\*070-4\*))
although it does not require it.

The XLL perdage provides general facilities for reoding and writing XML documents within R. A description of the facilities of cedenia and examples. Package SurfilateML on CRAN is one example building on XML.

NB: XML is available as a binary package for Windows, normally from the "CRAN extras" repository (which is selected by def

Next: Importing from other statistical systems, Previous: Introduction, Up: Top. [Contents][Index]

#### • Variations on read table:

- Fixed-width-format files:
   Data Interchange Format (DIF):
- Using scan directly:
- Re-shaping data:
  Flat contingency tables:

In Export to text files we saw a number of variations on the format of a spreadsheet-like text file, in which the data are presented

Next: Fixed-width-format files, Previous: Spreadsheet-like data, Up: Spreadsheet-like data. [Contents][Index]

2.1 Variations on read.table

The function read, table is the most convenient way to read in a rectangular grid of data. Because of the many possibilities, ther Bernare that read, table is an inefficient way to read in very large numerical matrices: see scan below. Some of the issues to consider are:

#### 1 Encoding

If the file contains non-ASCII character fields, ensure that it is read in the correct encoding. This is mainly an issue for recreated the left of the det., filetrocoling="latini")

Note that this will work in any locale which can represent Latin-1 strings, but not many Greek/Russian Chinese/Japanese

We recommend that you specify the beater argument emplicitly. Conventionally the header line has entries only for the co presented with a file that has a (possibly empty) header field for the row labels, read it in by something like read-table("file.det", badder = "BML, remonent = 1).

 $Column \ names \ can be given \ explicitly \ via the \ col. \ names; \ explicit names \ override \ the \ header \ line \ (if \ present).$ 

Separator
 Normally looking at the file will determine the field separator to be used, but with white-space separated files there may b "te". Note that the choice of separator affects the input of quoted strings.

If you have a tab-delimited file containing empty fields be sure to use  $_{\text{tap}} = \text{``} \backslash \text{t}^*.$ 

#### Quoting By definite character strings can be quoted by either 'm' or 'n', and in each case all the characters up to a matching quote as argument. For sep = "'or' the definite is characted to quote = "".

If no separator character is specified, quotes can be escaped within quoted strings by unmediately preceding them by  $\langle \gamma_i \rangle$ .

If a separator character is specified, quotes can be escaped within quoted strings by doubling them as is conventional in sg.

2 Spreadshe

What exactly is this R & RStudio thing?
The Good, The Bad & the Ugly (Sides of R)
Practical Issues

## What is R? Why/Why not R? Practical Issues.

The Good, The Bad & the Ugly (Sides of R)

- Developed by statisticians
   & scientists
- ② Learning Curve
- Memory and Speed Issues for certain tasks

```
(bull version*1.0" exceding="01rt-4"):
although it does not require it.

The XML peckage provides general facilities for reading and writing XML documents within R. A description of the facilities of celestia and examples Peckage StriPhraML on CRAN'ss one example building on XML.
```

NB: XML is available as a binary package for Windows, normally from the 'CRAN extras' repository (which is selected by def

Next: Importing from other statistical systems, Previous: Introduction, Up: Top. [Contents][Index]

#### Variations on read table:

- Fixed-width-format files:
   Data Interchange Format (DIF):
- Using scan directly:
   Re-shaping data:
- Flat contingency tables:

In Export to text file, we saw a number of variations on the format of a spreadsheet-like text file, in which the data are presented.

Next: Food-width-format files. Previous: Social-sheet-like data. Use Social-sheet-like data. [Contents] [Index]

2.1 Variations on read.table

The function read, table is the most convenient way to read in a rectangular grid of data. Because of the many possibilities, ther Bernare that read, table is an inefficient way to read in very large numerical matrices: see scae below. Some of the issues to consider rea

#### ome of the issues to conside

- Encoung
   If the file contains non-ASCII character fields, ensure that it is read in the correct encoding. This is mainly an issue for recreed-table("file.det", filetrocoling"latin(").
- Note that this will work in any locale which can represent Latin-1 strings, but not many Greek Russian Chinese Japanese
- We recommend that you specify the bester argument explicitly. Conventionally the header line has entries only for the co presented with a file that has a (possibly empty) header field for the row labels, read it in by something like read to be the first of the first property of the fir
- $Column \ names \ can be given \ explicitly \ via the \ col. \ names; \ explicit names \ override \ the \ header \ line \ (if \ present).$
- Separator
   Normally looking at the file will determine the field separator to be used, but with white-space separated files there may b
  "te". Note that the choice of separator affects the input of quoted strings.
- If you have a tab-delimited file containing empty fields be sure to use sep = "\t^\*.
- Quoting

  By definit character strings can be quoted by either "" or "", and in each case all the characters up to a matching quote as argument. For sep = "\n" the definit is charged to quote = "".
- If no separator character is specified, quotes can be escaped within quoted strings by immediately preceding them by  $^{\wedge},^{\vee}$ .

  If a separator character is specified, quotes can be escaped within quoted strings by doubling them as is conventional in sg.

The Good, The Bad & the Ugly (Sides of R)

- Developed by statisticians
   & scientists
- ② Learning Curve
- ③ Ugly syntax (my opinion)
  - compare Python

And versions 1.0° accessing "NOT-8").

The XML package provides general ficilities for resding and writing XML documents within R. A description of the ficilities c desith and example, Package SeptEmble, on CRAN to one example building on XML.

NB: XML is available as a bunny pockage for Windows, normally from the 'CRAN extras' repository (which is selected by def Next Importung from other attitution) available sevents. Previous Introduction, Up: Too [Contents Hindex]

2 Spreadshe

- Variations on read table:
   Fixed-width-format files:
   Data Interchange Format (DIF):
- Using scan directly:
   Re-shaping data:
- Flat contingency tables:
   In Export to text file; we saw a number of variations on the format of a spreadsheet-like text file, in which the data are presented.
- Next: Fixed-width-format files. Previous: Spreadsheet-like-data. Up: Spreadsheet-like-data. [Contents][Index]
  2.1 Variations on read-table.
- The function read, table is the most convenient way to read in a rectangular grid of data. Because of the many possibilities, ther Beruze that read, table is an inefficient way to read in very large numerical matrices, see scan below. Some of the issues to consider are:
- Encoding
   Encoding Brock Common ASCII character fields, ensure that it is read in the correct encoding. This is mainly an issue for recreed table: "file-det", fileteroodine-"latina")
- Note that this will week in any locale which can represent Latin-1 strings, but not many Greek/Russian/Chinese/Japanese
- We recommend that you specify the beafer argument explicitly. Conventionally the header line has entries only for the co presented with a file that has a (possibly empty) header field for the row labels, read it in by something like read, while "file, after", beafer = "Bux", remeases = 1".
- Column names can be given explicitly via the col.names; explicit names override the header line (if present).

  3. Separator
- Normally looking at the file will determine the field separator to be used, but with white-space separated files there may b
  "\ver". Note that the choice of separator affects the input of quoted strings.

  If you have a tab-definited file containing empty fields be sure to use \*sp = "\ver".
  - Quoting By definite character strings can be quoted by either "" or "", and in each case all the characters up to a matching quote as argument. For sep. = "\n" the definit is charged to quote = "".
- If no separator character is specified, quotes can be escaped within quoted strings by immediately preceding them by  $(\gamma_i)$ .

  If a separator character is specified, quotes can be escaped within quoted strings by doubling them as is conventional in st
  - 4 D > 4 A > 4 B > 4 B > 9 Q

## What is R? Why/Why not R? Practical Issues.

The Good, The Bad & the Ugly (Sides of R)

- Developed by statisticians
   & scientists
- ② Learning Curve
- ③ Ugly syntax (my opinion)
  - compare Python
- © Documentation style

```
The XML package provides general facilities for reading and writing XML documents within R. A description of the facilities of
details and examples. Package StatDataML on CRAN is one example building on XML
NB: XML is available as a binary package for Windows, normally from the "CRAN extras" repository (which is selected by def
Next: Importing from other statistical systems, Previous: Introduction, Up: Top. [Contents][Index]
                                                                                                           2 Spreadshe
· Variations on read table
· Fixed-width-format files

    Data Interchange Format (DIF):

· Re-shaping data:
· Flat contingency tables
In Export to text files we saw a number of variations on the format of a spreadsheet-like text file, in which the data are presented
Next: Fixed-width-format files, Previous: Spreadsheet-like data. Up: Spreadsheet-like data. [Contents][Index]
2.1 Variations on read.table
The function read, table is the most convenient way to read in a rectangular grid of data. Because of the many possibilities, ther
Beware that read table is an inefficient way to read in very large numerical matrices: see scan below
Some of the issues to consider are
      If the file contains non-ASCII character fields, ensure that it is read in the correct encoding. This is mainly an issue for re-
             read.table("file.dat", filetocoding="latin1")
      Note that this will work in any locale which can represent Latin-1 strings, but not many Greek-Russian/Chinese/Japanese
      We recommend that you specify the teaster argument explicitly. Conventionally the header line has entries only for the co
      presented with a file that has a (possibly empty) header field for the row labels, read it in by something like
              read table("file.det", header a TRUE, row.names a 1)
      Column names can be given emplicitly via the col. cases: explicit names override the header line (if present)
```

Normally looking at the file will determine the field separator to be used, but with white-space separated files there may b

By default character strings can be quoted by either '\*' or '-', and in each case all the characters up to a matching quote as

If no separator character is specified, quotes can be escaped within quoted strings by immediately preceding them by  $\langle \gamma_i \rangle$ If a separator character is specified, quotes can be escaped within quoted strings by doubling them as is conventional in sp

"\t". Note that the choice of separator affects the input of quoted strings.

If you have a tab-delimited file containing empty fields be sure to use sap = "\t".

argument. For sep = "\n" the default is changed to quote = ""

#### Outline

- 1 What is R? Why/Why not R? Practical Issues.
  - What exactly is this R & RStudio thing?
  - The Good, The Bad & the Ugly (Sides of R)
  - Practical Issues
- 2 Basic Examples
  - Assignment, Lists, Arrays etc.
  - Data Frames & Data Wrangling
  - 'apply' functions

# What is R? Why/Why not R? Practical Issues. Starting the Journey

R - http://ftp.sunet.se/pub/lang/CRAN/

- R http://ftp.sunet.se/pub/lang/CRAN/
- RStudio IDE http://www.rstudio.com/

- R http://ftp.sunet.se/pub/lang/CRAN/
- RStudio IDE http://www.rstudio.com/
- Case sensitive

- R http://ftp.sunet.se/pub/lang/CRAN/
- RStudio IDE http://www.rstudio.com/
- Case sensitive
- Expressions are printed, assignments not

- R http://ftp.sunet.se/pub/lang/CRAN/
- RStudio IDE http://www.rstudio.com/
- Case sensitive
- Expressions are printed, assignments not
- Assigning with <- (or ->)

- R http://ftp.sunet.se/pub/lang/CRAN/
- RStudio IDE http://www.rstudio.com/
- Case sensitive
- Expressions are printed, assignments not
- Assigning with <- (or ->)
- Commenting with #

- R http://ftp.sunet.se/pub/lang/CRAN/
- RStudio IDE http://www.rstudio.com/
- Case sensitive
- Expressions are printed, assignments not
- Assigning with <- (or ->)
- Commenting with #
- Indexing starts from 1

- R http://ftp.sunet.se/pub/lang/CRAN/
- RStudio IDE http://www.rstudio.com/
- Case sensitive
- Expressions are printed, assignments not
- Assigning with <- (or ->)
- Commenting with #
- Indexing starts from 1
- NaN (not a number), NA (not available)

#### Outline

- 1 What is R? Why/Why not R? Practical Issues
  - What exactly is this R & RStudio thing?
  - The Good, The Bad & the Ugly (Sides of R)
  - Practical Issues
- 2 Basic Examples
  - Assignment, Lists, Arrays etc.
  - Data Frames & Data Wrangling
  - 'apply' functions

introduction.R control\_structures.R

Assignment

# introduction.R control\_structures.R

- Assignment
- Indexing

# introduction.R control\_structures.R

- Assignment
- Indexing
- Arrays

# introduction.R control\_structures.R

- Assignment
- Indexing
- Arrays
- List

## introduction.R control structures.R

- Assignment
- Indexing
- Arrays
- List
- Control structures

#### Outline

- 1 What is R? Why/Why not R? Practical Issues
  - What exactly is this R & RStudio thing?
  - The Good, The Bad & the Ugly (Sides of R)
  - Practical Issues
- 2 Basic Examples
  - Assignment, Lists, Arrays etc.
  - Data Frames & Data Wrangling
  - 'apply' functions

#### data\_frames.R data\_wrangling.R

Creating DFs

#### data\_frames.R data\_wrangling.R

- Creating DFs
- Using DFs

#### data\_frames.R data\_wrangling.R

- Creating DFs
- Using DFs
- Aggregation

### data\_frames.R data\_wrangling.R

- Creating DFs
- Using DFs
- Aggregation
- Reshaping

#### Outline

- 1 What is R? Why/Why not R? Practical Issues
  - What exactly is this R & RStudio thing?
  - The Good, The Bad & the Ugly (Sides of R)
  - Practical Issues

### 2 Basic Examples

- Assignment, Lists, Arrays etc.
- Data Frames & Data Wrangling
- 'apply' functions

## 'apply' functions

 $apply\_functions.R$ 

lapply

## 'apply' functions

### $apply\_functions.R$

- lapply
- sapply

## 'apply' functions

#### apply\_functions.R

- lapply
- sapply
- tapply

## Coming Up

