Data Analysis with R:
Day 2

Sonja Hartnack, Terence Odoch & Muriel Buri

October 2017

Creating and assigning objects in R

Objects are assigned values using $< -$, an arrow formed out of $<$ and $-$. For example, the following command assigns the value 1 to the object `a`.

```
a <- 1 # ALWAYS use "gets" assignment operator!
# a = 1 # DO NOT USE the equal sign as the assignment operator!
```

After this assignment, the object `a` contains the value 1. Another assignment to the same object will change the content.

```
a <- 5
```

Examples of assigned objects: Single number

```
a <- 1
b <- 2
c <- a + b # c = 3
c

## [1] 3
```

Examples of assigned objects: Vector

```
a <- c(1, 2, 3, 4, 5)
b <- 1
c <- a + b
c

## [1] 2 3 4 5 6
```

Examples of assigned objects: Model

```r
anova_model <- aov(weight ~ feed, data = chickwts)
summary(anova_model)

##             Df Sum Sq Mean Sq F value   Pr(>F)
## feed         5 231129   46226   15.37 5.94e-10 ***
## Residuals   65 195556    3009
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Examples of assigned objects: Dataframe

```
bac <- bacteria
str(bac) # $ week: int  0 2 4 11 0 2 6 11 0 2 ...

## 'data.frame': 220 obs. of  6 variables:
##  $ y   : Factor w/ 2 levels "n","y": 2 2 2 2 2 2 1 2 2 2 ...
##  $ ap  : Factor w/ 2 levels "a","p": 2 2 2 2 1 1 1 1 1 1 ...
##  $ hilo: Factor w/ 2 levels "hi","lo": 1 1 1 1 1 1 1 1 2 2 ...
##  $ week: int  0 2 4 11 0 2 6 11 0 2 ...
##  $ ID  : Factor w/ 50 levels "X01","X02","X03",..: 1 1 1 1 2 2 2 2 3 3 ...
##  $ trt : Factor w/ 3 levels "placebo","drug",..: 1 1 1 1 3 3 3 3 2 2 ...

bac_sub <- subset(bac, week == 2)
str(bac_sub) # $ week: int  2 2 2 2 2 2 2 2 2 2 ...

## 'data.frame': 44 obs. of  6 variables:
##  $ y   : Factor w/ 2 levels "n","y": 2 2 2 2 2 2 1 2 2 2 ...
##  $ ap  : Factor w/ 2 levels "a","p": 2 1 1 2 2 1 1 2 2 2 ...
##  $ hilo: Factor w/ 2 levels "hi","lo": 1 1 2 2 2 2 1 1 2 1 ...
##  $ week: int  2 2 2 2 2 2 2 2 2 2 ...
##  $ ID  : Factor w/ 50 levels "X01","X02","X03",..: 1 2 3 4 5 6 7 8 9 11 ...
##  $ trt : Factor w/ 3 levels "placebo","drug",..: 1 3 2 1 1 2 3 1 1 1 ...
```

Structure of a R objects

The `str` function displays the `str`ucture of an R object. One line for each ”basic” structure is displayed.

```
## 'data.frame': 44 obs. of  6 variables:
##  $ y   : Factor w/ 2 levels "n","y": 2 2 2 2 2 2 1 2 2 2 ...
##  $ ap  : Factor w/ 2 levels "a","p": 2 1 1 2 2 1 1 2 2 2 ...
##  $ hilo: Factor w/ 2 levels "hi","lo": 1 1 2 2 2 2 1 1 2 1 ...
##  $ week: int  2 2 2 2 2 2 2 2 2 2 ...
##  $ ID  : Factor w/ 50 levels "X01","X02","X03",..: 1 2 3 4 5 6 7 8 9 11 ...
##  $ trt : Factor w/ 3 levels "placebo","drug",..: 1 3 2 1 1 2 3 1 1 1 ...
```

Exercise 4

Data types in R

- numeric

```
data(ToothGrowth)
ToothGrowth$len[1:6]

## [1]  4.2 11.5  7.3  5.8  6.4 10.0

class(ToothGrowth$len[1:6])

## [1] "numeric"
```

- integers

```
bacteria$week[1:6]

## [1]  0  2  4 11  0  2

class(bacteria$week[1:6])

## [1] "integer"
```

- (un/ordered) factor

```
chickwts$feed[1:6]

## [1] horsebean horsebean horsebean horsebean horsebean horsebean
## Levels: casein horsebean linseed meatmeal soybean sunflower

levels(chickwts$feed)[1:3]

## [1] "casein"    "horsebean" "linseed"
```

Data types in R: Ordered Factors

Ordinal variables are represented as ordered factors:

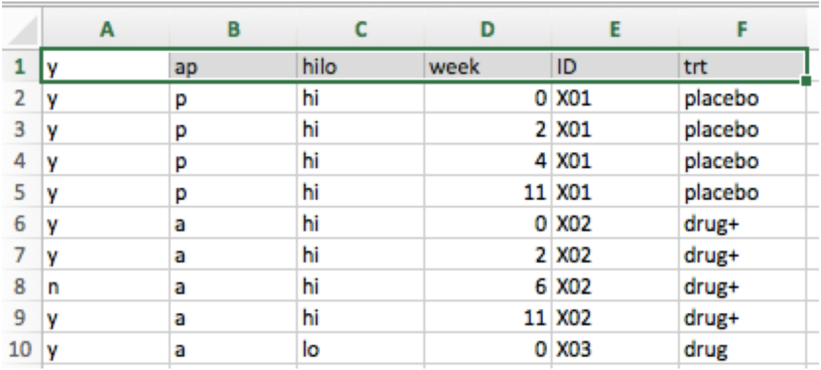
```r
bac_growth <- c("none", "+", "++", "+", "+++", "+", "none") # vector
bac_growth <- factor(bac_growth, levels = c("none", "+", "++", "+++"),
               order = TRUE)
bac_growth

## [1] none +    ++   +    +++  +    none
## Levels: none < + < ++ < +++

#
mood <- c("OK", "Well", "Super", "Super", "Don't ask", "OK") # vector
mood <- factor(mood, levels = c("Don't ask", "Well", "OK", "Super"),
               order = TRUE)
mood

## [1] OK        Well      Super     Super     Don't ask OK
## Levels: Don't ask < Well < OK < Super
```

Exercise 5

Exercise 6

Rules for importing data into R (from Excel)

- First row of excel sheet contains **variable names**: `y, ap, hilo, week, ID, trt`.
- Columns of excel sheet represent **variables**.
- Rows of excel sheet represent **observations per individual** (except for the first row).



	A	B	C	D	E	F
1	y	ap	hilo	week	ID	trt
2	y	p	hi	0	X01	placebo
3	y	p	hi	2	X01	placebo
4	y	p	hi	4	X01	placebo
5	y	p	hi	11	X01	placebo
6	y	a	hi	0	X02	drug+
7	y	a	hi	2	X02	drug+
8	n	a	hi	6	X02	drug+
9	y	a	hi	11	X02	drug+
10	y	a	lo	0	X03	drug

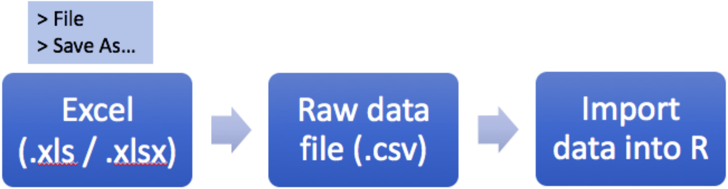
Rules for naming variables

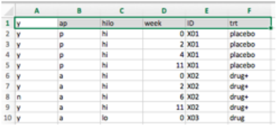
Variable names should ..

- start with a letter (not a number): `y`, `ap`, `hilo`, `week`, `ID`, `trt`
- longer variables names should be separated with dots: `time.in.weeks`
- do not use special characters, such as /, #, @, &, ⋆, ...

How to import Excel files into R?

Three major steps: Excel file preparation



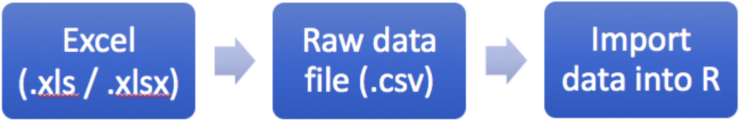


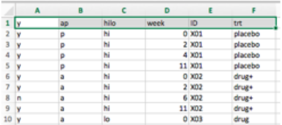
	A	B	C	D	E	F
1	y	ap	hilo	week	ID	trt
2	y	p	hi	0	X01	placebo
3	y	p	hi	2	X01	placebo
4	y	p	hi	4	X01	placebo
5	y	p	hi	11	X01	placebo
6	y	a	hi	0	X02	drug+
7	y	a	hi	2	X02	drug+
8	n	a	hi	6	X02	drug+
9	y	a	hi	11	X02	drug+
10	y	a	lo	0	X03	drug

How to import Excel files into R?

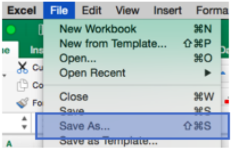
Three major steps: Excel file preparation

> File
> Save As...



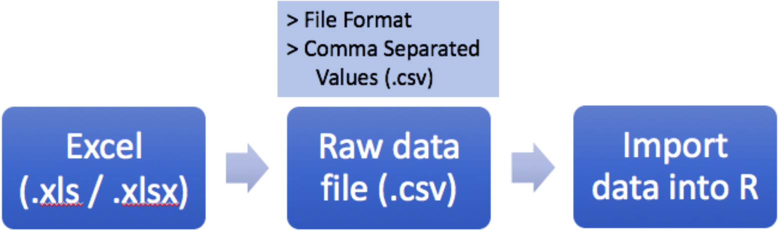


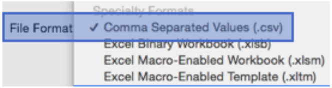
	A	B	C	D	E	F
1	y	ap	hilo	week	ID	trt
2	y	p	hi	0	X01	placebo
3	y	p	hi	2	X01	placebo
4	y	p	hi	4	X01	placebo
5	y	p	hi	11	X01	placebo
6	y	a	hi	0	X02	drug+
7	y	a	hi	2	X02	drug+
8	n	a	hi	6	X02	drug+
9	y	a	hi	11	X02	drug+
10	y	a	lo	0	X03	drug



How to import Excel files into R?

Three major steps: Save raw data file as .csv





Specialty Formats

File Format ✓ Comma Separated Values (.csv)

Excel Binary Workbook (.xlsb)

Excel Macro-Enabled Workbook (.xlsm)

Excel Macro-Enabled Template (.xltm)

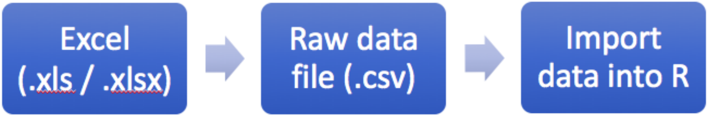
How to import Excel files into R?

Three major steps: Import data into R

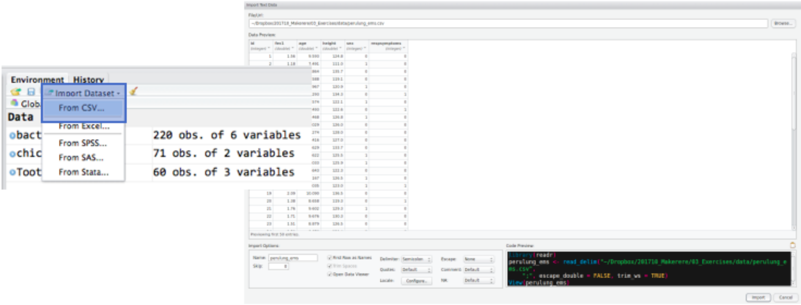
How to import Excel files into R?

Three major steps: Import data into R

> Import Dataset
> From CSV …
> Browse
> Check "Import Options"
> Check "Data Preview"
> Copy / Paste "Code Preview"



Excel (.xls / .xlsx) → Raw data file (.csv) → Import data into R



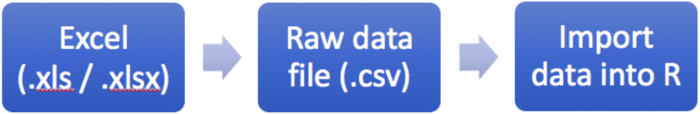
Data
bact 220 obs. of 6 variables
chic 71 obs. of 2 variables
Toot 60 obs. of 3 variables

```
library(readr)
perulung_ems <- read_delim("~/Dropbox/201710_Makerere/03_Exercises/data/perulung_e
ms.csv",
    ";", escape_double = FALSE, trim_ws = TRUE)
View(perulung_ems)
```

How to import Excel files into R?

Three major steps: Import data into R

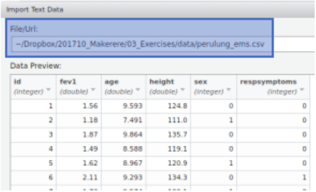
> Import Dataset
> From CSV …
> Browse
> Check "Import Options"
> Check "Data Preview"
> Copy / Paste "Code Preview"



Excel (.xls / .xlsx)

Raw data file (.csv)

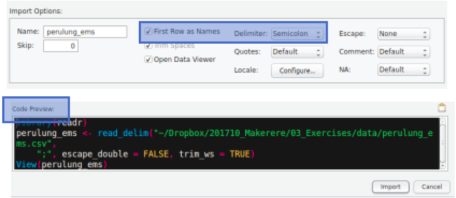
Import data into R



File/Url: ~/Dropbox/201710_Makerere/03_Exercises/data/perulung_ems.csv

Data Preview:

id (integer)	fev1 (double)	age (double)	height (double)	sex (integer)	respsymptoms (integer)
1	1.56	9.593	124.8	0	0
2	1.18	7.491	111.0	1	0
3	1.87	9.864	135.7	0	0
4	1.49	8.588	119.1	0	0
5	1.62	8.967	120.9	1	0
6	2.11	9.293	134.3	0	1



Import Options:

Name: perulung_ems

Skip: 0

☑ First Row as Names

☑ Trim Spaces

☑ Open Data Viewer

Delimiter: Semicolon

Quotes: Default

Locale: Configure...

Escape: None

Comment: Default

NA: Default

Code Preview:

```
library(readr)
perulung_ems <- read_delim("~/Dropbox/201710_Makerere/03_Exercises/data/perulung_e
ms.csv",
    ";", escape_double = FALSE, trim_ws = TRUE)
View(perulung_ems)
```

How to import Excel files into R?

```r
# Import .csv file with the help of the read.csv function
# Be sure to add sep = ";" so that we separate the columns.
lung <- read.csv("C:\\Users\\Exercises\\data\\perulung_ems.csv", sep = ";")
head(lung)
str(lung)
```

Exercise 7: perulung

Data from a study of lung function among children living in a deprived suburb of Lima, Peru. Data taken from Kirkwood and Sterne, 2nd edition.
Variables:

- `fev1`: in liter, ”forced expiratory volume in 1 second” measured by a spirometer. This is the maximum volume of air which the children could breath out in 1 second
- `age`: in years
- `height`: in cm
- `sex`: 0 = girl, 1 = boy
- `respsymp`: respiratory symptoms experienced by the child over the previous 12 months