Using R for statistical analyses

Robert Bauer

Warnemünde, 04/17/2012

Leibniz Institute for
Baltic Sea Research
Warnemünde

Day 1 - Agenda:

- What is R?
- Installation of R and RStudio
- Importing Data
- Accessing Data
- Vector Operations
- Plots
- Exercises
- Help
- Literature

What is R?

"R is a freely available programming language for statistical analyses and graphics, much like S-Plus or SAS"

R's Strengths

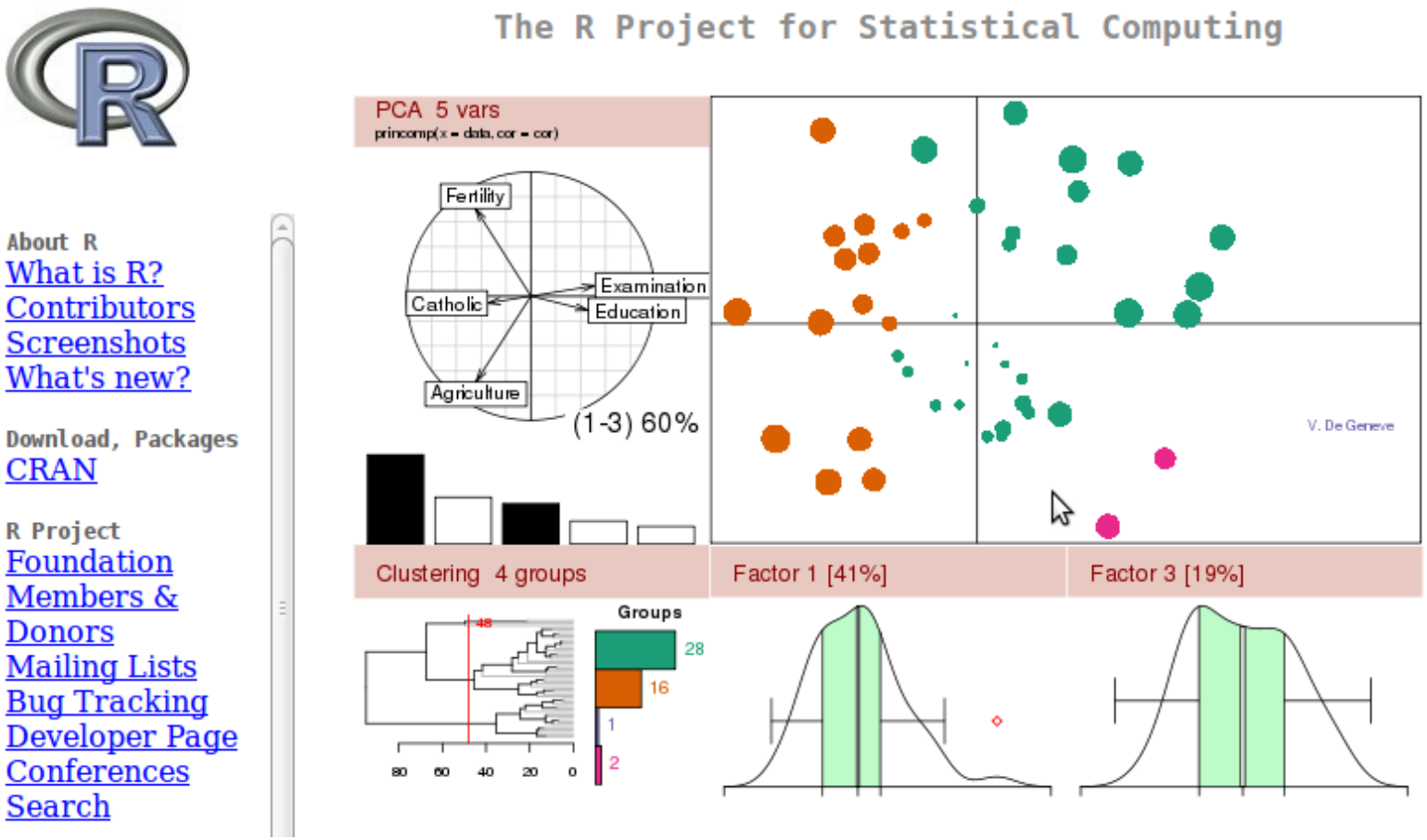
- Data management & manipulation
- Statistics
- Graphics
- Programming language (calculations repeatable)
- available for Windows, Linux and Mac OS X
- Free
- comes with lots of functions, still extendable
- no need to know all functions, but to know how to find them!
- HELP: Active user community

R's Weaknesses

- Not very user friendly at start
- No commercial support
- Slower than other programming languages (Java, C++)

R-Homepage

http://www.r-project.org/



Installation of R

- required version: R 2.11.1 or higher

  Windows:

  - Download "base distribution" from CRAN, e.g. http://mirrors.softliste.de/cran/bin/windows/base/R-2.15.0-win.exe
  - run default installation process

  Ubuntu:

  - Run: System -> Administration -> Synaptic Package Manager
  - search for "r-base" using the Quick-Filter and proceed installation process

Installation of RStudio

http://rstudio.org/

- Download -> RStudio Desktop
- Choose recommended version

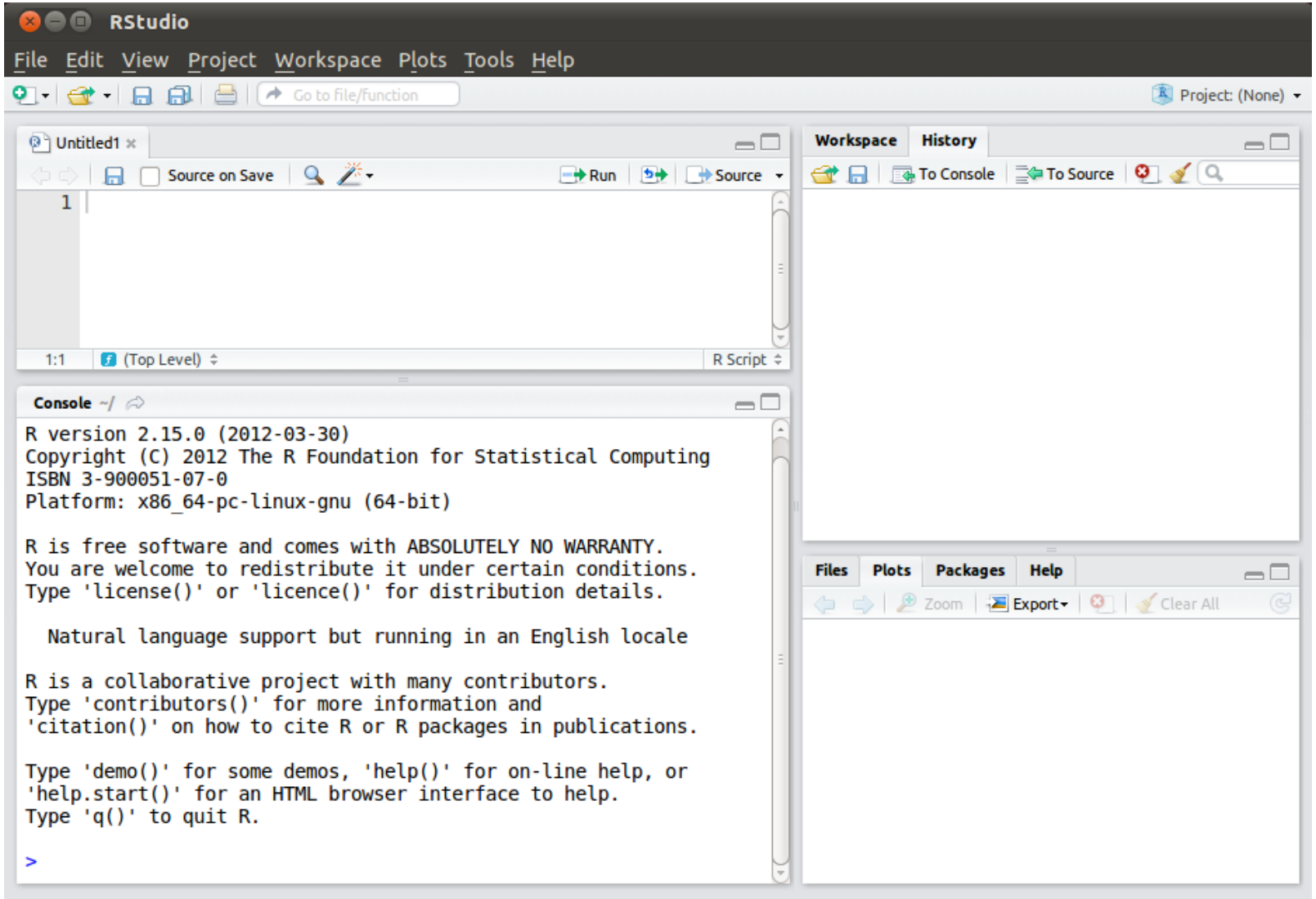
Windows:

- RStudio 0.95.265 - Windows XP/Vista/7
- run default installation process

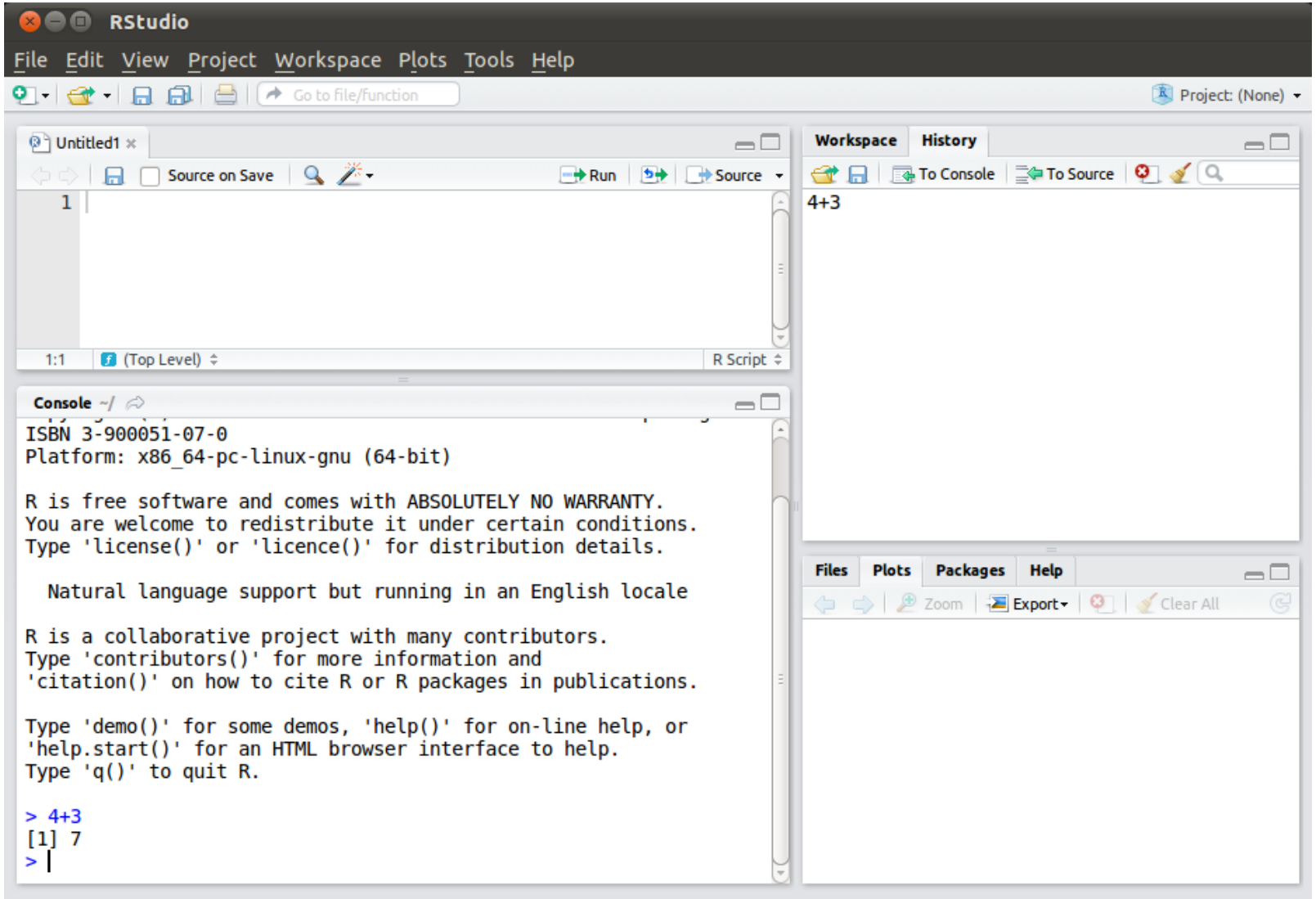
Ubuntu:

- RStudio 0.95.265 - Debian 6+/Ubuntu 10.04+ (64-bit)
- Open deb package with "Ubuntu Software Center"

Getting Started!



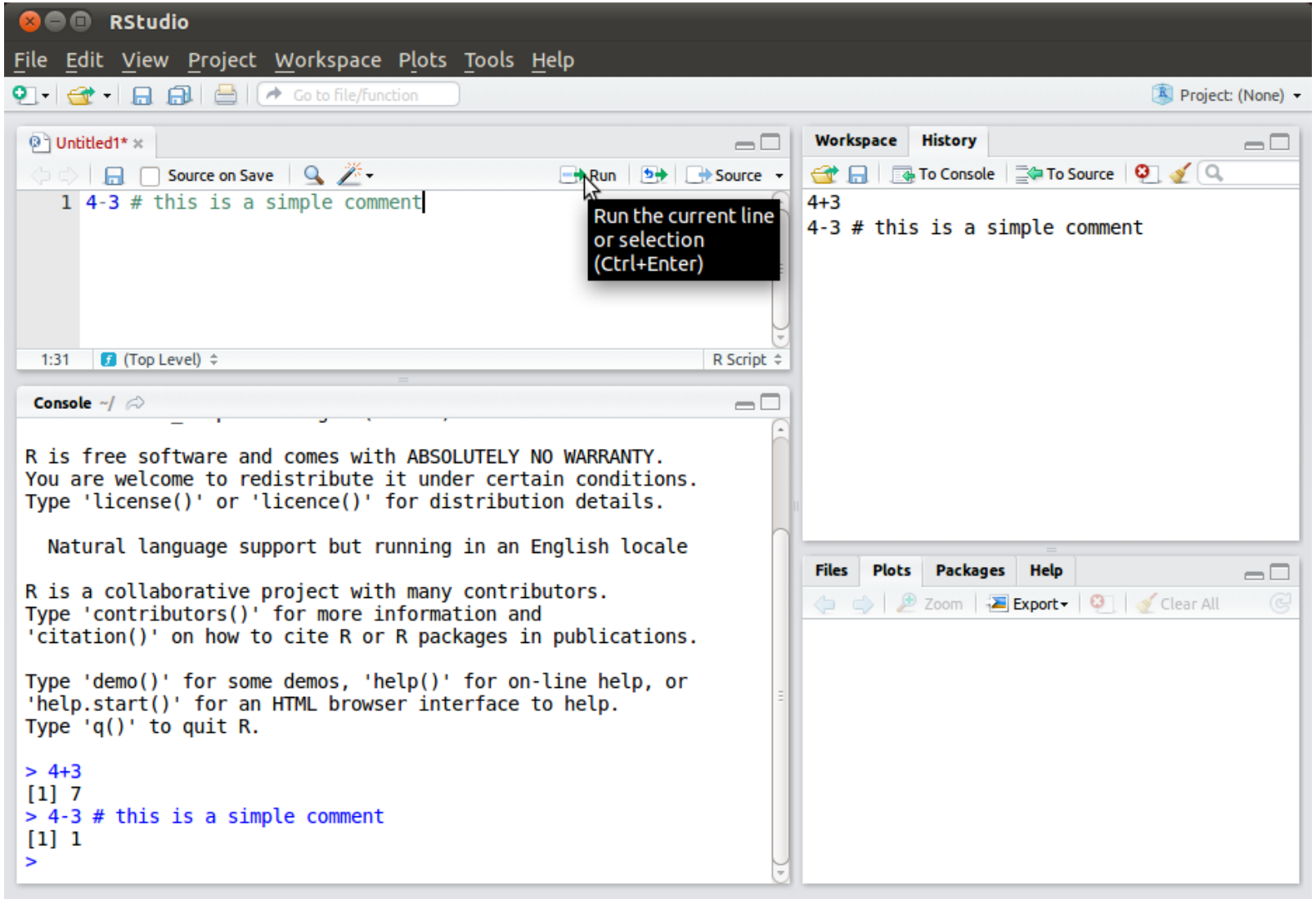
Console & History



Arithmetic Operators

Operator	Description	Example
`+`	addition	`4+3`
`-`	subtraction	`4-3`
`*`	multiplication	`4*3`
`^ or **`	exponentiation	`4^3`
`/`	division	`4/3`
`%%`	modulus (`x` mod `y`)	`4%%3`
`%/%`	integer division	`4%/%3`

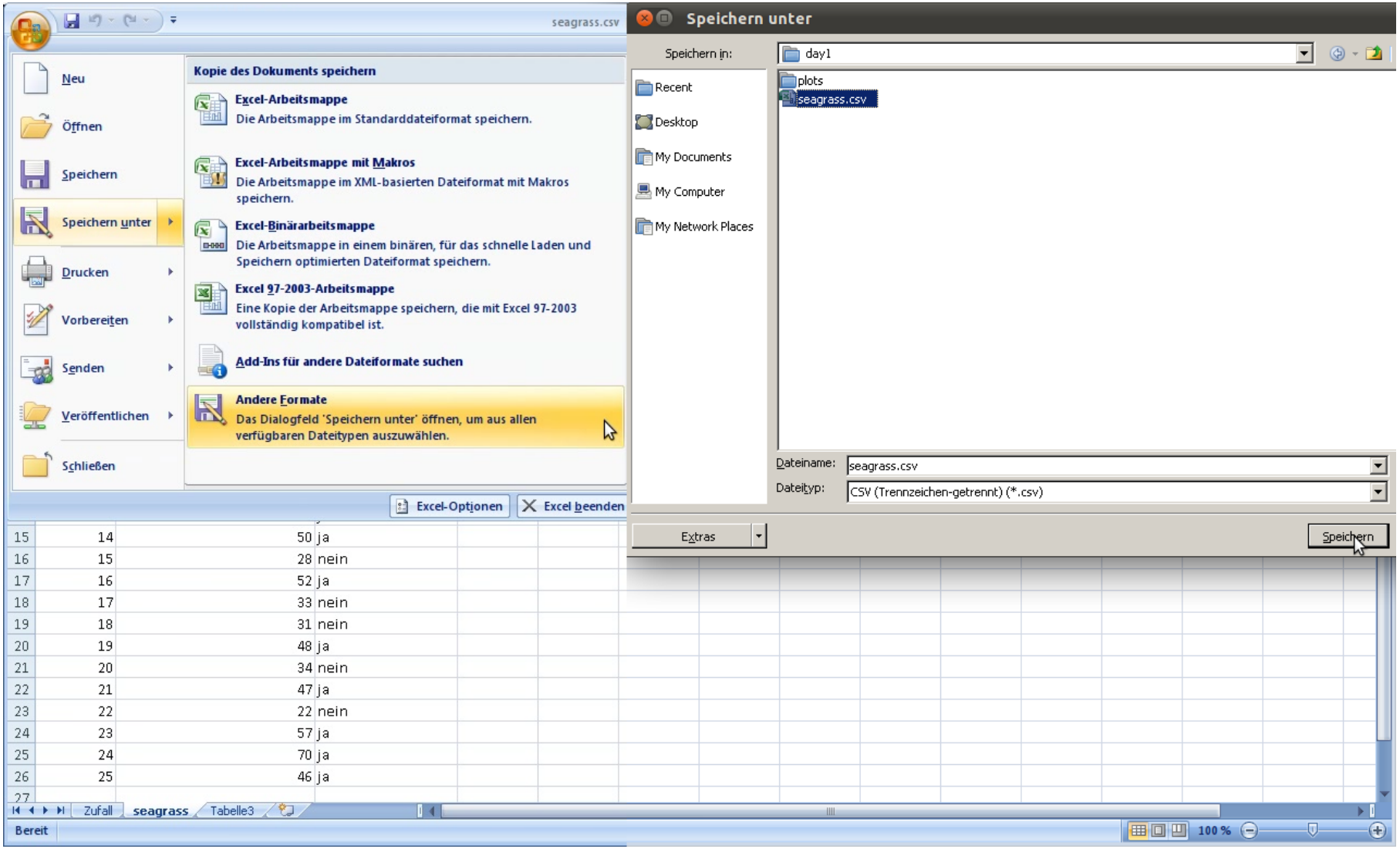
Run Code from Scripts - Another Way to Execute Commands



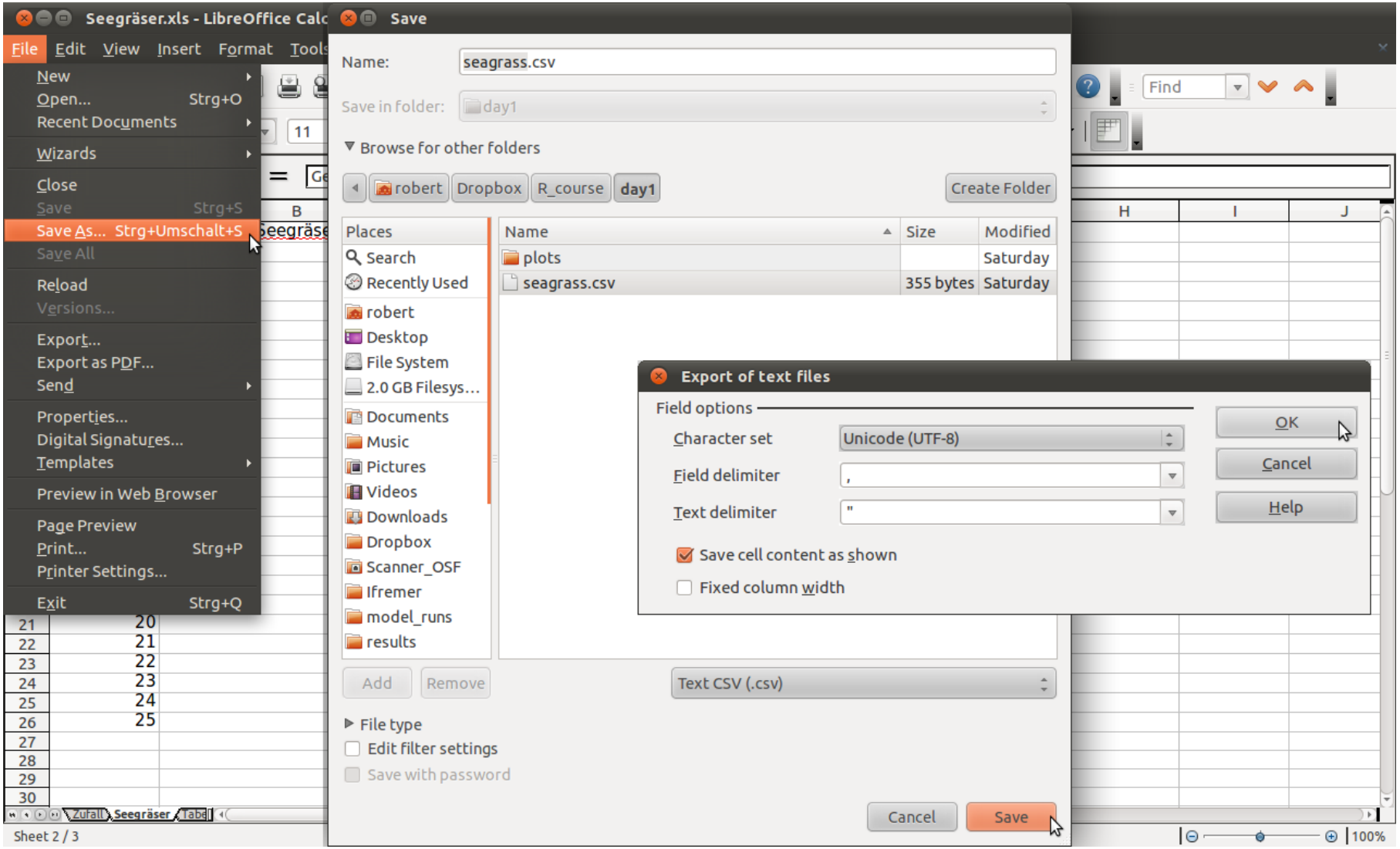
Arithmetic Functions

Operator	Description	Example
`sqrt()`	square root	`sqrt(4)`
`sum(), prod()`	sum, product	`prod(3,4)`
`exp()`	exponential function	`exp(1)`
`log()`	natural logarithm	`log(exp(4))`
`log(x, base)`	base specific logarithm	`log(4,10)`

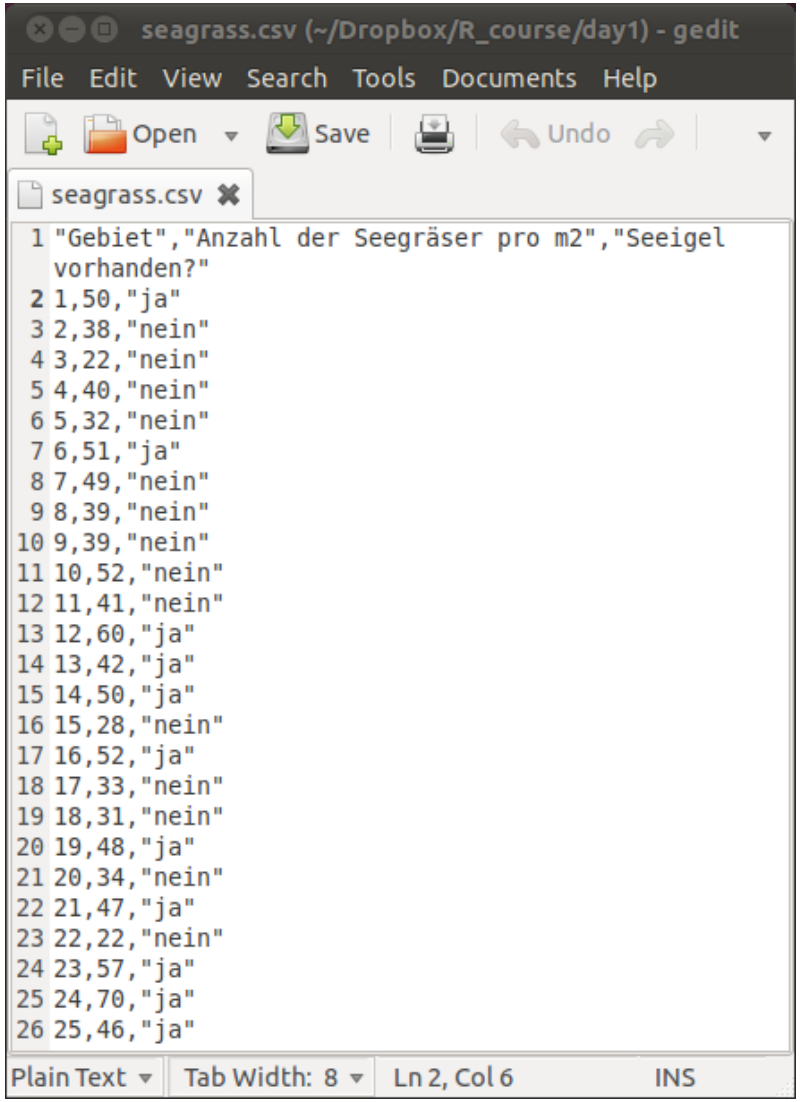
Importing CSV-Files



Importing CSV-Files



CSV-Files


seagrass.csv (~/Dropbox/R_course/day1) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
seagrass.csv
1 "Gebiet","Anzahl der Seegräser pro m2","Seeigel vorhanden?"
2 1,50,"ja"
3 2,38,"nein"
4 3,22,"nein"
5 4,40,"nein"
6 5,32,"nein"
7 6,51,"ja"
8 7,49,"nein"
9 8,39,"nein"
10 9,39,"nein"
11 10,52,"nein"
12 11,41,"nein"
13 12,60,"ja"
14 13,42,"ja"
15 14,50,"ja"
16 15,28,"nein"
17 16,52,"ja"
18 17,33,"nein"
19 18,31,"nein"
20 19,48,"ja"
21 20,34,"nein"
22 21,47,"ja"
23 22,22,"nein"
24 23,57,"ja"
25 24,70,"ja"
26 25,46,"ja"
Plain Text
Tab Width: 8
Ln 2, Col 6
INS

Data Import

```
read.table(file, header = FALSE, sep = "", dec = ".")
```

Argument	Description
`file`	path incl. filename
`header`	TRUE: read the first line as a header of column names
`sep`	character separating values, (default: “” = white space)
`dec`	character indicating decimal points

further arguments and explanations `?read.table`

First Session

```
seagrass <- read.table("path_to_file/seagrass.csv", header=T
    , sep=',', dec=".") # load dataframe
```

First Session

```
seagrass <- read.table("path_to_file/seagrass.csv", header=T
    , sep=',', dec=".") # load dataframe
```

```
setwd("~/Dropbox/R_course/day1") # set working directory
seagrass <- read.table("seagrass.csv", header=T, sep=',',
    dec=".") # load dataframe
```

First Session

```
seagrass <- read.table("path_to_file/seagrass.csv", header=T
    , sep=',', dec=".") # load dataframe

setwd("~/Dropbox/R_course/day1") # set working directory
seagrass <- read.table("seagrass.csv", header=T, sep=',',
    dec=".") # load dataframe
```

```
# accsess data:
seagrass # return entire table in console
View(seagrass) # open table in editor
head(seagrass) # show first 10 rows incl. header
```

First Session

```
seagrass <- read.table("path_to_file/seagrass.csv", header=T
    , sep=',', dec=".") # load dataframe

setwd("~/Dropbox/R_course/day1") # set working directory
seagrass <- read.table("seagrass.csv", header=T, sep=',',
    dec=".") # load dataframe

# accsess data:
seagrass # return entire table in console
View(seagrass) # open table in editor
head(seagrass) # show first 10 rows incl. header
```

```
colnames(seagrass) <- c("area", "n", "urchins") # rename
    columns
head(seagrass)
```

Selecting a single column - 3 ways to Rome

```
seagrass[,1] # selecting first column
seagrass$area # selecting column name
attach(seagrass) # load each column of dataframe as vector
area
```

Basic Vector Functions

Function	Description
`length()`	length of vector
`unique()`	unique values of a vector
`min(), max()`	minimum, maximum
`mean(), median()`	average, `median`
`sd(), var()`	standard deviation, variance
`sum(), prod()`	sum, product
`cumsum(), cumprod()`	cumulative sum/ product
`range()`	min() and max()
`summary()`	min, 1st quart., median, mean, 3rd quart., max

e.g. `length(area)`

Subsetting Data

```
seagrass$n[seagrass$urchins == "ja"]
n[urchins == "ja"] # seagrass density at areas with sea
    urchins
subset(n, urchins == "ja")
subset(seagrass, urchins == "ja" & area > 40)
which(n > 40) # index of areas where n > 40
```

Logical Operators

Operator	Description	Example (a=2, b=1)
<	less than	a<b → FALSE
<=	less than or equal to	a<=b → FALSE
>	greater than	a>b → TRUE
>=	greater than or equal to	a>=b → TRUE
==	equal to	a==b → FALSE
!=	not equal to	a!=b → TRUE
|	or	a|b → TRUE
&	and	a&b → FALSE
isTRUE(x)	test if x is TRUE	isTRUE(a) → FALSE

Exercises

1. In which areas was the seagrass density less than 30 or greater than 50?
2. What is the range of the seagrass density per area type?
3. What is the average seagrass density for areas with and without sea urchins?

Results

```
# 1.In which areas was the seagrass density less than 30 or
    greater than 50?
> which(n < 30 | n > 50) # index of areas!
[1]  3  6 10 12 15 16 22 23 24

# 2.What is the range of the seagrass density per area type?
> range(n[urchins == "ja"])
[1] 42 70
> range(n[urchins == "nein"])
[1] 22 52

# 3.What is the average seagrass density for areas with and
    without sea urchins?
> mean(n[urchins == "ja"])
[1] 52.09091

> mean(n[urchins == "nein"])
[1] 35.71429
```

First Plots

```
attach(seagrass) # load each column of dataframe as vector
boxplot(n[urchins == "ja"], n[urchins == "nein"], data =
    seagrass) # boxplot of specified categories
```

First Plots

```
attach(seagrass) # load each column of dataframe as vector
boxplot(n[urchins == "ja"], n[urchins == "nein"], data =
    seagrass) # boxplot of specified categories
```

```
boxplot(n~urchins, data = seagrass) # boxplot of all
    categories given in urchins
```

First Plots

```
attach(seagrass) # load each column of dataframe as vector
boxplot(n[urchins == "ja"], n[urchins == "nein"], data =
    seagrass) # boxplot of specified categories

boxplot(n~urchins, data = seagrass) # boxplot of all
    categories given in urchins
```

```
boxplot(n~urchins, data = seagrass, las=1) # rotate y-axis
    values by 90 degrees
```

First Plots

```
attach(seagrass) # load each column of dataframe as vector
boxplot(n[urchins == "ja"], n[urchins == "nein"], data =
    seagrass) # boxplot of specified categories

boxplot(n~urchins, data = seagrass) # boxplot of all
    categories given in urchins

boxplot(n~urchins, data = seagrass, las=1) # rotate y-axis
    values by 90 degrees
```

```
boxplot(n~urchins, data = seagrass, las=1, xlab="sea urchins
    ", ylab="seagrass density", names=c("available", "not
    available")) # set axes labels & name boxplot categories
```

Exercises

- Import "Fish.csv" (mind not available values!)
- Which was the maximum, which was the minimum size of each caught species?
- Create box-plots of the length distribution of both species and both sexes
- Create species specific box-plots of the full, empty and liver weight
- Define a new vector: Hepta Somatic Index $HSI = \frac{liver\ weight * 100}{total\ weight}$
- What is the median & the range of the HSI per species?

Help

Functions

```
help("mean")
?mean
```

Official Manuals

http://cran.r-project.org/manuals.html
An Introduction to R
R Data Import/Export
R Installation and Administration
Writing R Extensions

Frequently asked questions (FAQ)

http://cran.r-project.org/doc/manuals/R-FAQ.html

Mailing List

https://stat.ethz.ch/mailman/listinfo/r-help

Literature

Uwe Ligges - Programmieren mit R. Springer-Verlag, Heidelberg, 2005.
(free copy available on:
http://dx.doi.org/10.1007/978-3-540-79998-6)