

# Introduction to R

## Using R

Arni Magnusson

Hafro, 8 Nov 2010

# Outline

- 1 Projects - organizing and archiving your work
- 2 Packages - using them and finding them
- 3 Scripts - editing scripts and functions
- 4 Help - R help system, manuals

# Outline

- 1 Projects - organizing and archiving your work
- 2 Packages - using them and finding them
- 3 Scripts - editing scripts and functions
- 4 Help - R help system, manuals

# One root folder for all projects

e:/project

2008 faroes  
schaefer

2009 shrimp  
tuna

2010 pan  
msc  
unu

In statistical computing, short paths without spaces are helpful:

- e:/project/msc/data/landings.dat

## Each project can be divided into four folders

**admin** documents related to the project  
proposal.doc, meetings.doc

**analysis** script.R, workspace.RData  
model.tpl, input.dat, output.rep  
overview.xls

**data** original.txt, original.csv

**report** manuscripts, slideshows, figures

# File types

`.R` R code (text file that can be pasted into R)  
script, function (with `#` comments)  
complex R object (list, factor w/levels)

`.RData` workspace  
large R object

`.txt` data

`.csv`

`.dat`

# Load and save

**Workspace**      project.RData    bigobject.RData

```
load(file)
save(ls(), file)
save(bigobj, file)
```

**Code**            function.R      script.R      object.R

```
source(file)
dump(obj, file)
```

# Backup

## System backup

Hafrö automatic backups on tape

## Revision control

Multiauthor projects (e.g. Gadget) are under revision control

## Directory backup

User makes a backup copy of a project directory: 'analysis.zip'

## File backup

User makes a backup copy of a file: 'script.old'

## Editor backup

Emacs can make automatic backups in a dedicated directory

## R backup

R can make automatic backups of '.RData' inside `.Last()`

# Backup

## Same disk

Undo mistakes, revisit old version

## External

Recover from catastrophes (disk failure, stolen laptop)

# Outline

- 1 Projects - organizing and archiving your work
- 2 Packages - using them and finding them**
- 3 Scripts - editing scripts and functions
- 4 Help - R help system, manuals

# Using packages

```
find("function")
```

```
search()
```

```
library()
```

```
library(pkg)
```

```
ls("package:pkg")
```

```
install.packages("pkg")
```

```
update.packages()
```

## Core packages

**base** data objects, input/output, maths, etc.

**datasets** example datasets

**graphics** traditional plots

**lattice** trellis plots

**stats** statistical models and tests

## Hafro packages

**geo** draw maps

**fjlst** survey data

**Logbooks** logbook data

**ROracleUI** Oracle queries

# The `gdata` package

`Args(fun)` show function args, similar to `args(fun)`

`env()` show all environments, similar to `search()`

`is.what(obj)` show all 'is.\*()' test results

`ll()` show all objects in workspace, similar to `ls()`

`ll(obj)` show all elements of a list, or columns in data frame,  
similar to `names(obj)`

`keep(obj)` remove all objects in workspace, except `obj`

## CRAN package repository

When you type

```
install.packages("pkg")
```

R downloads the package from CRAN

<http://cran.r-project.org>

main package repository (> 2600 packages)

use mirror: Scandinavia, UK, Ireland

<http://cran.r-project.org/web/views/>

<http://crantastic.org>

browse packages by topic and maintainer, with reviews

# Outline

- 1 Projects - organizing and archiving your work
- 2 Packages - using them and finding them
- 3 Scripts - editing scripts and functions**
- 4 Help - R help system, manuals

## Editing scripts and functions

**R** plain and simple  
no extra packages  
optional script window

**Editors** syntax highlighting  
closing parentheses and strings  
efficient text editing, keybindings, regexp  
handle many scripts, recent files, etc.

**IDE** send code to R  
features for writing R packages  
many editors have optional R add-on  
e.g. Emacs, Tinn-R, Eclipse

# history

```
history()
```

```
history(Inf)
```

## source

```
source("script.R")
```

Runs everything in the `script.R` file

Like copying everything from the script into the R window

# Outline

- 1 Projects - organizing and archiving your work
- 2 Packages - using them and finding them
- 3 Scripts - editing scripts and functions
- 4 Help - R help system, manuals

## Help pages

Structured web pages with links

**Arguments** things that determine the function output

**See Also** related functions

**Examples** code that can be copied from browser and pasted into R

## Help functions

```
args(log)
```

```
apropos("log")
```

```
help(log, help_type="html")
```

```
help(log)
```

```
?log
```

```
?["
```

```
help(package="pkg")
```

```
help.search("keyword")
```

```
??keyword
```

# Manuals

## An introduction to R (official user manual)

<http://mirrors.dotsrc.org/cran/doc/manuals/R-intro.pdf>

## User-contributed tutorials

<http://mirrors.dotsrc.org/cran/other-docs.html>

## FAQ

<http://mirrors.dotsrc.org/cran/doc/FAQ/R-FAQ.pdf>

## Technical manuals

<http://mirrors.dotsrc.org/cran/manuals.html>

## Source code

View source code of an R function

```
lm
```

```
logLik.lm
```

```
getAnywhere(logLik.lm)
```

Browse the entire R source code

```
http://svn.r-project.org/R/trunk/
```

Download the entire R source code

```
ftp://mirrors.dotsrc.org/cran/src/base/R-latest.tar.gz
```

```
$ svn co http://svn.r-project.org/R/trunk R
```

# Web search

## Normal web search

Your favorite search engine in a browser

## R web search

```
RSiteSearch("keyword")
```