Course: Analytics, Machine Learning, and the Digital Economy

Code in R

Lecturer Radjabova Dilnora

Learning aims

- Basic use of R and R help
- How to give R commands
- R data structures
- Reading and writing data
- Some more R commands (exercises)

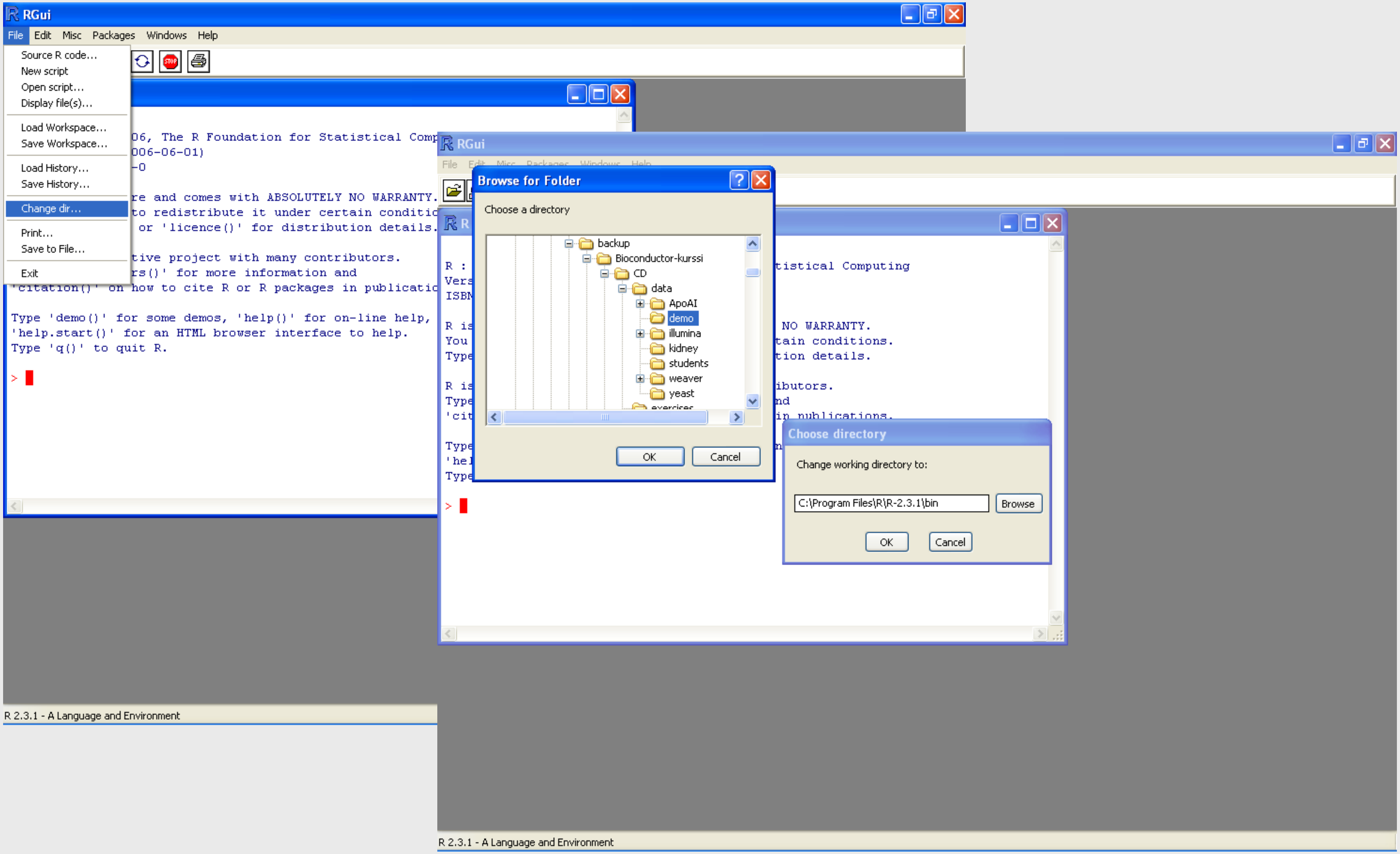
R project

- ”R is a free software environment for statistical computing and graphics” (http://www.r-project.org)
- ”Bioconductor is a software project for the analysis of genomic data” (http://www.bioconductor.org)
  - Currently works as an expansion to R

Packages

- R consists of a core and packages.
- Packages contain functions that are not available in the core.
- For example, Bioconductor code is distributed as several dozen of packages for R.
  - Software packages
  - Metadata (annotation) packages

Starting the work with R



```
06, The R Foundation for Statistical Comp
006-06-01)
-0

re and comes with ABSOLUTELY NO WARRANTY.
to redistribute it under certain conditio
 or 'licence()' for distribution details.

tive project with many contributors.
rs()' for more information and
'citation()' on how to cite R or R packages in publicatio

Type 'demo()' for some demos, 'help()' for on-line help,
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

R 2.3.1 - A Language and Environment

```
R :                                    tistical Computing
Vers
ISBN

R is                                    NO WARRANTY.
You                                    tain conditions.
Type                                   tion details.

R is                                   ibutors.
Type                                   nd
'cit                                   in publications.

Type
'hel
Type

>
```

Browse for Folder

Choose a directory

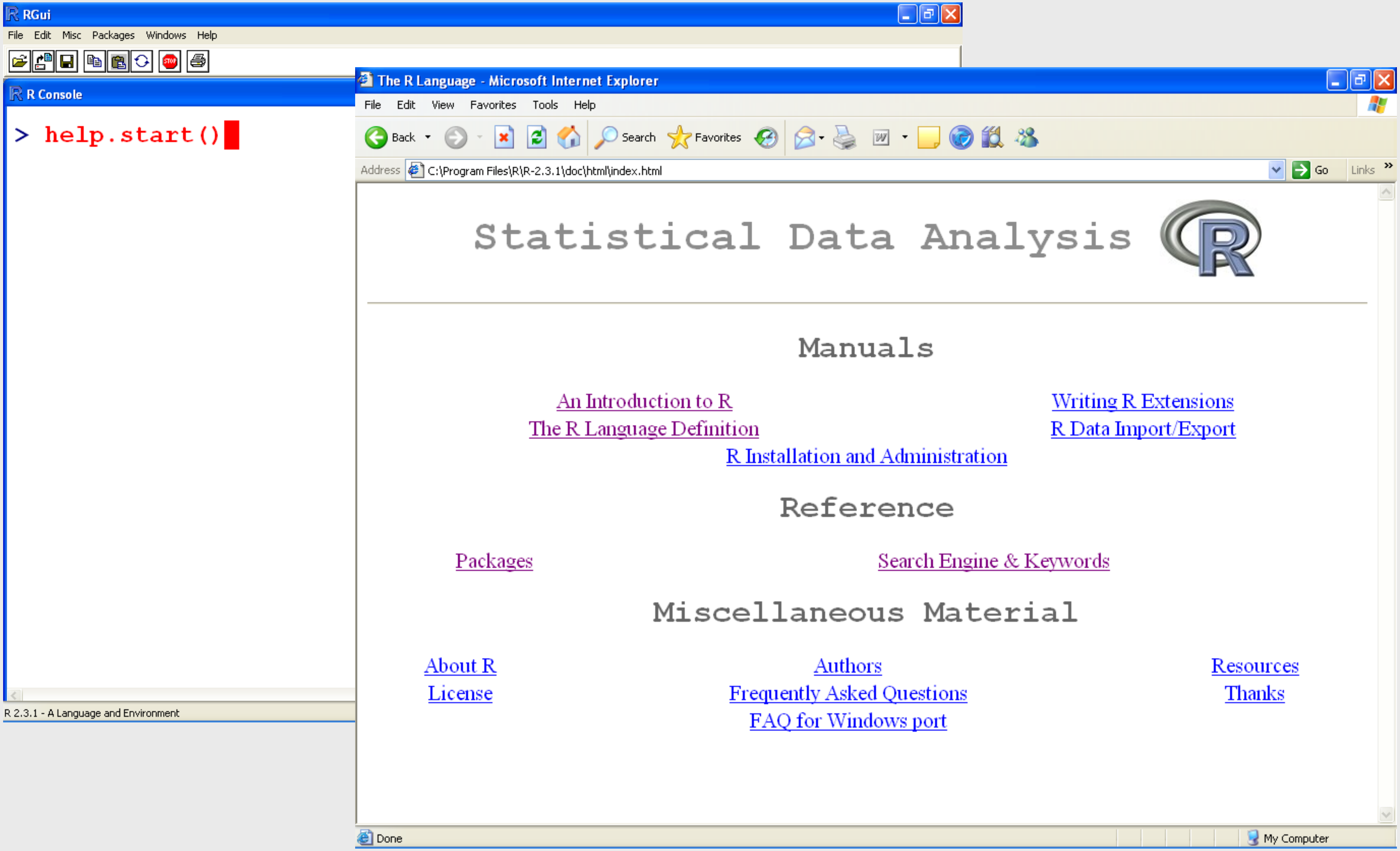
- backup
  - Bioconductor-kurssi
    - CD
      - data
        - ApoAI
        - demo
        - illumina
        - kidney
        - students
        - weaver
        - yeast
      - exercises

Choose directory

Change working directory to: C:\Program Files\R\R-2.3.1\bin

R 2.3.1 - A Language and Environment

Start help



```
> help.start()
```

Statistical Data Analysis

Manuals

An Introduction to R
The R Language Definition
Writing R Extensions
R Data Import/Export
R Installation and Administration

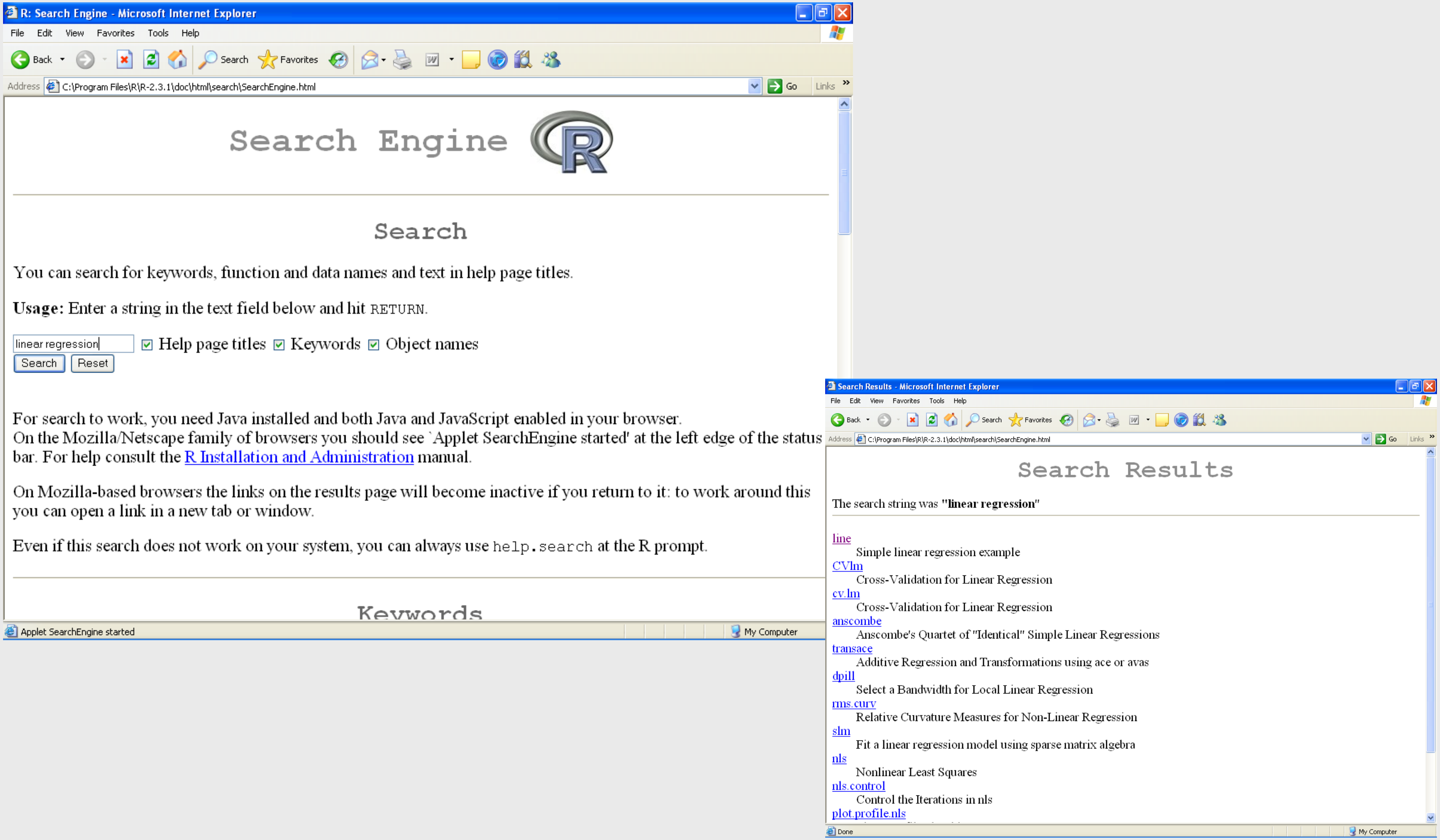
Reference

Packages
Search Engine & Keywords

Miscellaneous Material

About R
License
Authors
Frequently Asked Questions
FAQ for Windows port
Resources
Thanks

Help - Search engine



Address: C:\Program Files\R\R-2.3.1\doc\html\search\SearchEngine.html

Search Engine

Search

You can search for keywords, function and data names and text in help page titles.

Usage: Enter a string in the text field below and hit `RETURN`.

linear regression ☑ Help page titles ☑ Keywords ☑ Object names

Search Reset

For search to work, you need Java installed and both Java and JavaScript enabled in your browser.
On the Mozilla/Netscape family of browsers you should see `Applet SearchEngine started' at the left edge of the status bar. For help consult the R Installation and Administration manual.

On Mozilla-based browsers the links on the results page will become inactive if you return to it: to work around this you can open a link in a new tab or window.

Even if this search does not work on your system, you can always use `help.search` at the R prompt.

Keywords

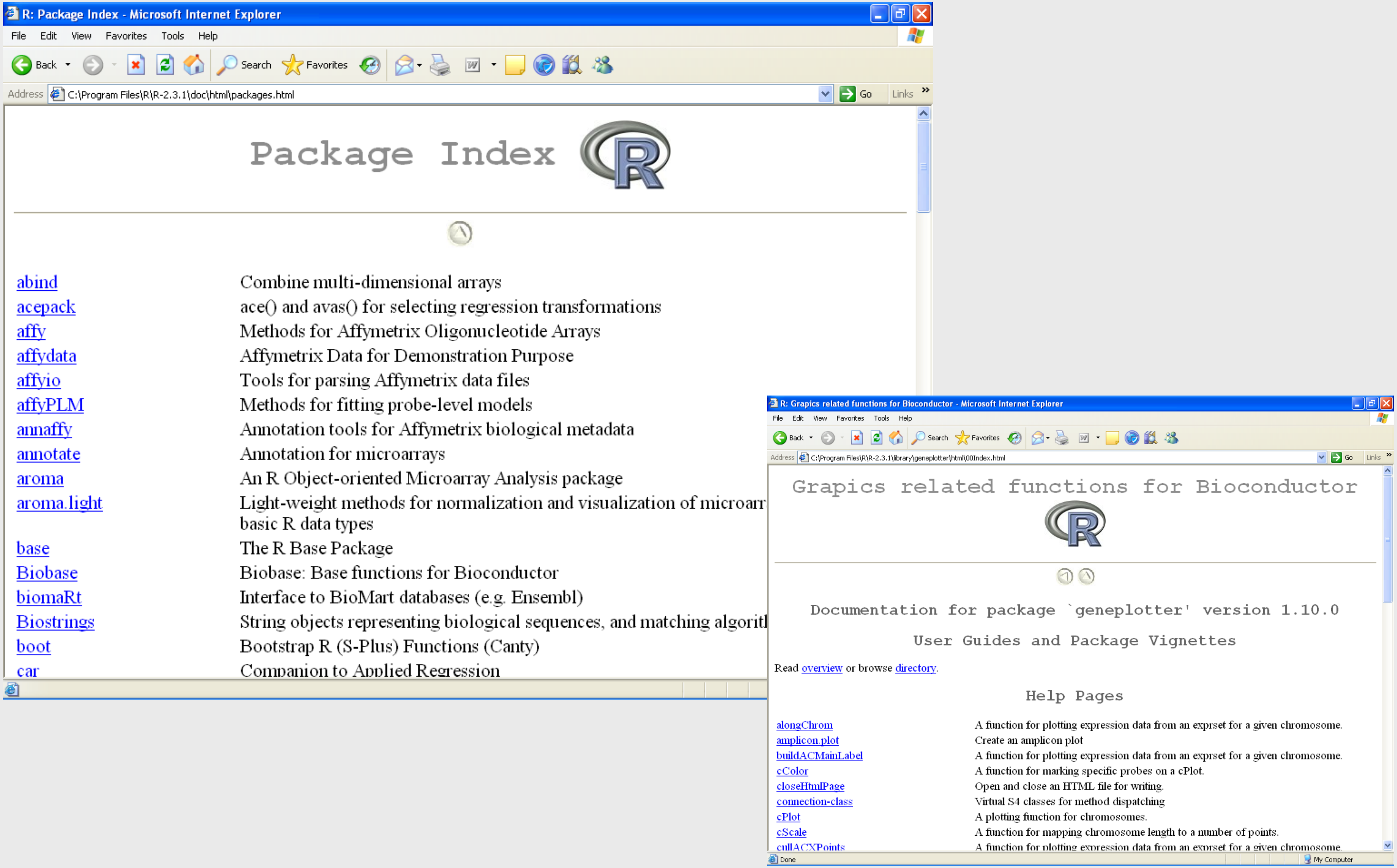
Address: C:\Program Files\R\R-2.3.1\doc\html\search\SearchEngine.html

Search Results

The search string was **"linear regression"**

line	Simple linear regression example
CVlm	Cross-Validation for Linear Regression
cv.lm	Cross-Validation for Linear Regression
anscombe	Anscombe's Quartet of "Identical" Simple Linear Regressions
transace	Additive Regression and Transformations using ace or avas
dpill	Select a Bandwidth for Local Linear Regression
rms.curv	Relative Curvature Measures for Non-Linear Regression
slm	Fit a linear regression model using sparse matrix algebra
nls	Nonlinear Least Squares
nls.control	Control the Iterations in nls
plot.profile.nls	

Help - packages



Package Index

abind	Combine multi-dimensional arrays
acepack	ace() and avas() for selecting regression transformations
affy	Methods for Affymetrix Oligonucleotide Arrays
affydata	Affymetrix Data for Demonstration Purpose
affyio	Tools for parsing Affymetrix data files
affyPLM	Methods for fitting probe-level models
annaffy	Annotation tools for Affymetrix biological metadata
annotate	Annotation for microarrays
aroma	An R Object-oriented Microarray Analysis package
aroma.light	Light-weight methods for normalization and visualization of microarr basic R data types
base	The R Base Package
Biobase	Biobase: Base functions for Bioconductor
biomaRt	Interface to BioMart databases (e.g. Ensembl)
Biostrings	String objects representing biological sequences, and matching algorit
boot	Bootstrap R (S-Plus) Functions (Canty)
car	Companion to Applied Regression

Grapics related functions for Bioconductor

Documentation for package `geneplotter' version 1.10.0

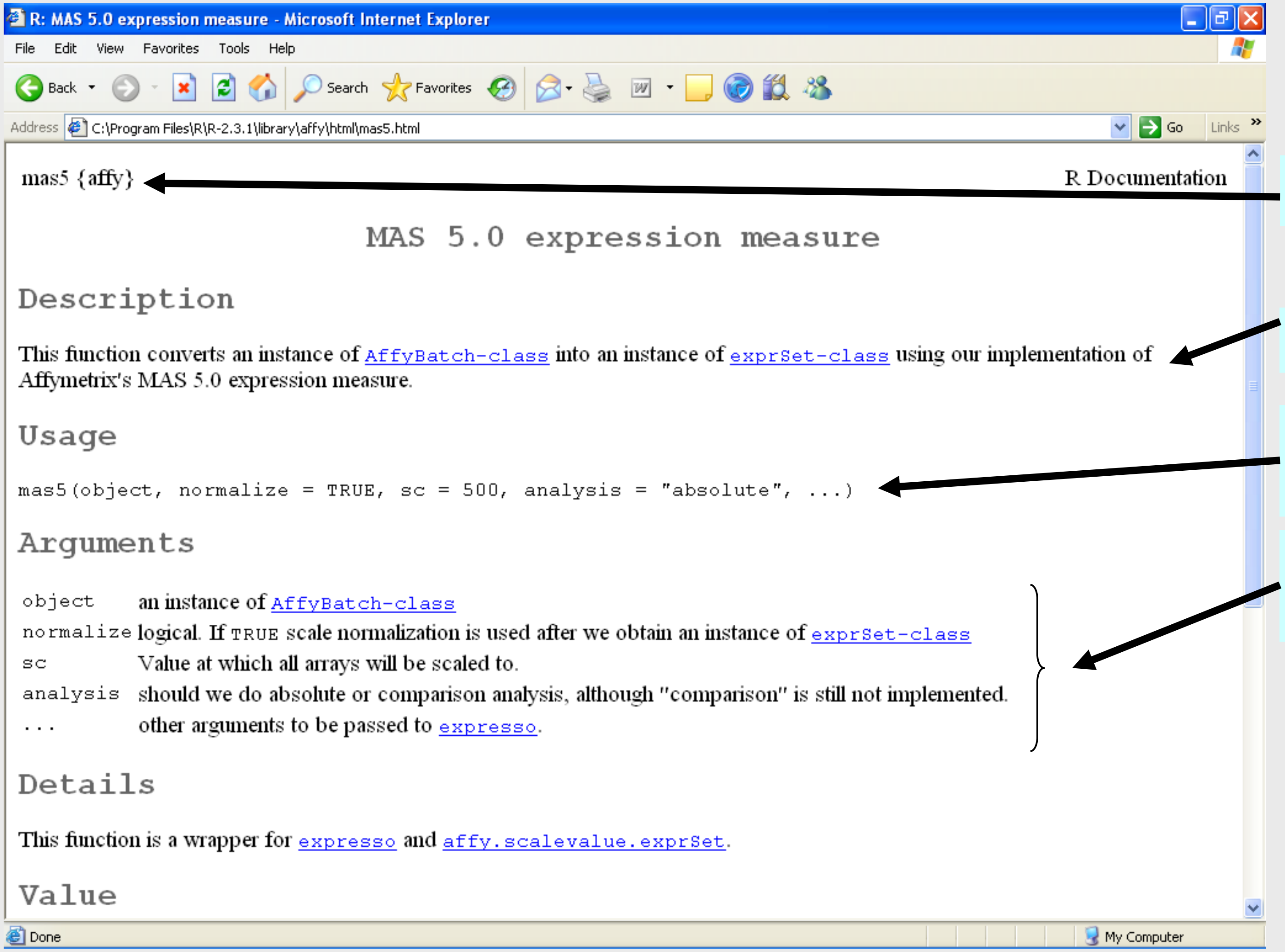
User Guides and Package Vignettes

Read overview or browse directory.

Help Pages

alongChrom	A function for plotting expression data from an exprset for a given chromosome.
amplicon.plot	Create an amplicon plot
buildACMainLabel	A function for plotting expression data from an exprset for a given chromosome.
cColor	A function for marking specific probes on a cPlot.
closeHtmlPage	Open and close an HTML file for writing.
connection-class	Virtual S4 classes for method dispatching
cPlot	A plotting function for chromosomes.
cScale	A function for mapping chromosome length to a number of points.
cullACXPoints	A function for plotting expression data from an exprset for a given chromosome.

Anatomy of a help file 1/2



mas5 {affy} R Documentation

MAS 5.0 expression measure

Description

This function converts an instance of AffyBatch-class into an instance of exprSet-class using our implementation of Affymetrix's MAS 5.0 expression measure.

Usage

```
mas5(object, normalize = TRUE, sc = 500, analysis = "absolute", ...)
```

Arguments

`object`	an instance of AffyBatch-class
`normalize`	logical. If `TRUE` scale normalization is used after we obtain an instance of exprSet-class
`sc`	Value at which all arrays will be scaled to.
`analysis`	should we do absolute or comparison analysis, although "comparison" is still not implemented.
`...`	other arguments to be passed to expresso.

Details

This function is a wrapper for expresso and affy.scalevalue.exprSet.

Value

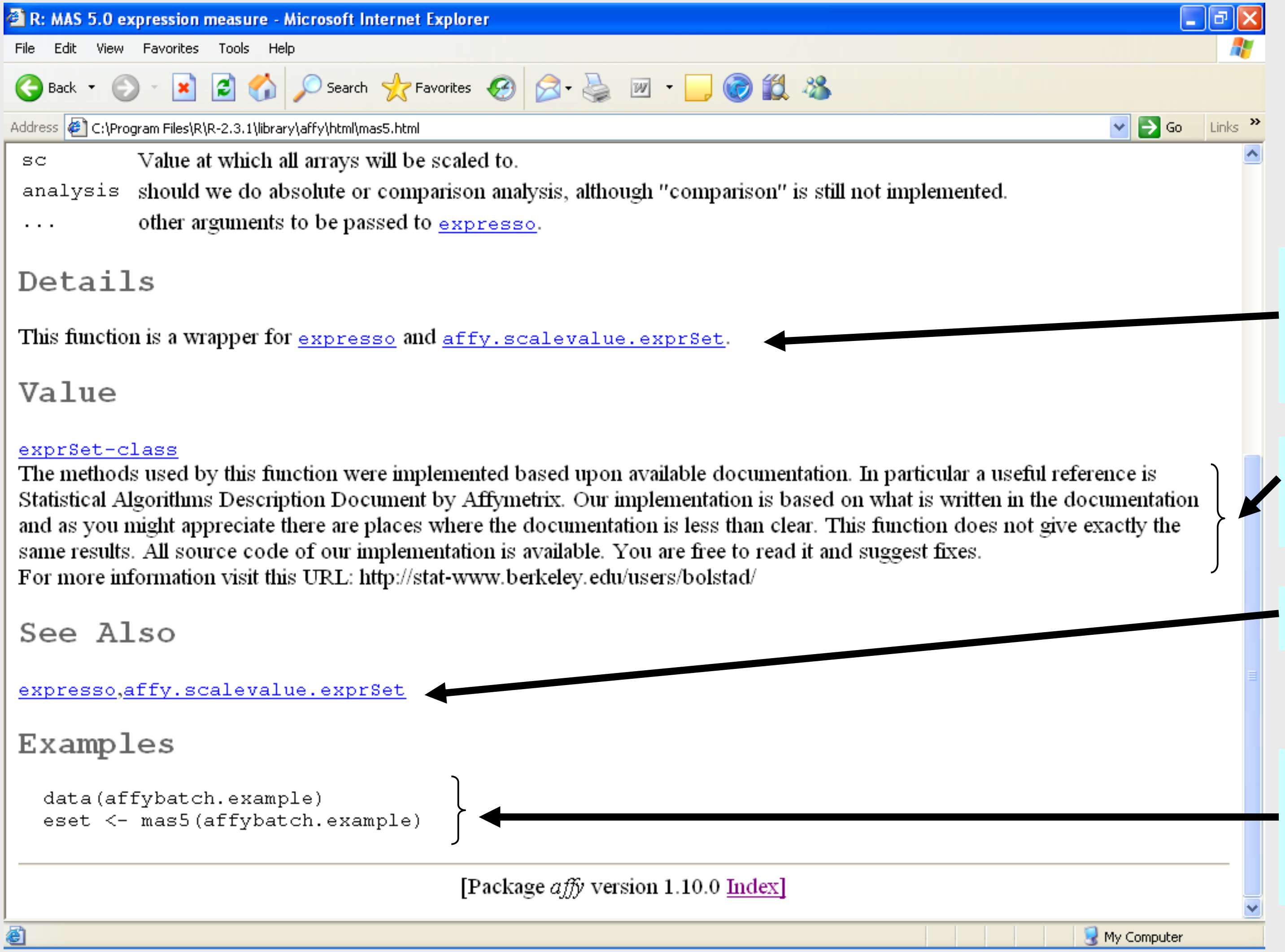
Function {package}

General description

Command and it’s argument

Detailed description of arguments

Anatomy of a help file 2/2



`sc`	Value at which all arrays will be scaled to.
`analysis`	should we do absolute or comparison analysis, although "comparison" is still not implemented.
`...`	other arguments to be passed to `expresso`.

Details

This function is a wrapper for `expresso` and `affy.scalevalue.exprSet`.

Value

`exprSet-class`
The methods used by this function were implemented based upon available documentation. In particular a useful reference is Statistical Algorithms Description Document by Affymetrix. Our implementation is based on what is written in the documentation and as you might appreciate there are places where the documentation is less than clear. This function does not give exactly the same results. All source code of our implementation is available. You are free to read it and suggest fixes.
For more information visit this URL: http://stat-www.berkeley.edu/users/bolstad/

See Also

`expresso`,`affy.scalevalue.exprSet`

Examples

```
data(affybatch.example)
eset <- mas5(affybatch.example)
```

[Package *affy* version 1.10.0 Index]

Description of how function actually works

What function returns

Related functions

Examples, can be run from R by: `example(mas5)`

Functions or commands in R 1/3

- To use a function in a package, the package needs to be loaded in memory.
- Command for this is library( ), for example:

```
library(affy)
```

- There are three parts in a command:
  - the command
  - brackets
  - Arguments inside brackets (these are not always present)

Functions or commands in R 2/3

- R is case sensitive, so take care when typing in the commands!
  - `library(affy)` works, but `Library(affy)` does not.
- Multiple commands can be written on the same line. Here we first remove missing values from the variable year, and then calculate it’s arithmetic average.
  - Writing:
    - na.omit(year)
    - mean(year)
  - Would be the same as
    - mean(na.omit(year))

Functions or commands in R 3/3

- Command can have many arguments. These are always giving inside the brackets.
- Numeric (1, 2, 3…) or logic (T/F) values and names of existing objects are given for the arguments without quotes, but string values, such as file names, are always put inside quotes. For example:
- `mas5(dat3, normalize=T, analysis=”absolute”)`

Data structures 1/6

- Vector
  - A list of numbers, such as (1,2,3,4,5)
  - R: `a<-c(1,2,3,4,5)`
    - Command c creates a vector that is assigned to object a
- Factor
  - A list of levels, either numeric or string
  - R: `b<-as.factor(a)`
    - Vector a is converted into a factor

Data structures 2/6

- Data frame
  - A table where columns can contain numeric and string values
  - R: `d<-data.frame(a, b)`
- Matrix
  - All columns must contain either numeric or string values, but these can not be combined
  - R: `e<-as.matrix(d)`
    - Data frame d is converted into a matrix e
  - R: `f<-as.data.frame(e)`
    - Matrix e is converted into a dataframe f

Data structures 3/6

- List
  - Contains a list of objects of possibly different types.
  - R: `g<-as.list(d)`
    - Converts a data frame d into a list g
- Class structures
  - Many of the Bioconductor functions create a formal class structure, such as an AffyBatch object.
  - They contain data in slots
  - Slots can be accessed using the @-operator:
    - dat2@cdfName

Data structures 4/6

- Some command need to get, for example, a matrix, and do not accept a data frame. Data frame would give an error message.
- To check the object type:
  - R: `class(d)`
- To check what fields there are in the object:
  - R: `d`
  - R: `str(d)`
- To check the size of the table/matrix:
  - R: `dim(d)`
- To check the length of a factor of vector:
  - R: `length(a)`

Data structures 5/6

- Some data frame related commands:
  - R: `names(d)`
    - Reports column names
  - R: `row.names(d)`
    - Reports row names
- These can also be used for giving the names for the data frame. For example:
  - R: `row.names(d)<-c("a","b","c","d","e")`
    - Letters from a to e are used as the row names for data frame d
    - Note the quotes around the string values!
  - R: `row.names(d)`

Data structures 5/6

- Naming objects:
  - Never use command names as object names!
  - If your unsure whether something is a command name, type to the comman line first. If it gives an error message, you’re safe to use it.
  - Object names can’t start with a number
  - Never use special characters, such as å, ä, or ö in object names.
  - Underscore (_) is not usable, use dot (.) instead:
    - Not acceptable: `good_data`
    - Better way: `good.data`
  - Object names are case sensitive, just like commands

Reading data 1/2

- Command for reading in text files is:

```
read.table(”suomi.txt”, header=T, sep=”\t”)
```

- This examples has one command with three arguments: file name (in quotes), header that tells whether columns have titles, and sep that tells that the file is tab-delimited.

Reading data 2/2

- It is customary to save the data in an object in R. This is done with the assignment operator (<-):

```
dat<-read.table(”suomi.txt”, header=T, sep=”\t”)
```

- Here, the data read from file suomi.txt is saved in an object dat in R memory.
- The name of the object is on the left and what is assigned to the object is on the right.
- Command `read.table( )` creates a data frame.

Using data frames

- Individual columns in the data frame can be accessed using one of the following ways:
  - Use its name:
    - `dat$year`
    - dat is the data frame, and year is the header of one of its columns. Dollar sign ($) is an opertaor that accesses that column.
  - Split the data frame into variables, and use the names directly:
    - `attach(dat)`
    - `year`
  - Use subscripts

Subscripts 1/2

- Subscripts are given inside square brackets after the object’s name:
  - `dat[,1]`
    - Gets the first column from the object dat
  - `dat[,1]`
    - Gets the first row from the object dat
  - `dat[1,1]`
    - Gets the first row and it’s first column from the object dat
- Note that `dat` is now an object, not a command!

Subscripts 2/2

- Subscripts can be used for, e.g., extracting a subset of the data:
  - `dat[which(dat$year>1900),]`
    - Now, this takes a bit of pondering to work out…
    - First we have the object dat, and we are accessing a part of it, because it’s name is followed by the square brackets
    - Then we have one command (`which`) that makes an evaluation whether the column year in the object dat has a value higher than 1900.
    - Last the subscript ends with a comma, that tells us that we are accessing rows.
    - So this command takes all the rows that have a year higher 1900 from the object dat that is a data frame.

Writing tables

- To write a table:
  - `write.table(dat, ”dat.txt”, sep=”\t”)`
  - Here an object dat is written to a file called dat.txt. This file should be tab-delimited (argument sep).
- To capture what is written on the screen:
  - `sink(”output.txt”)`
  - `dat`
  - `sink( )`
  - Here, output written on the screen should be written to a file output.txt instead. Contents of the object dat are written to the named file. Last, the file is closed.
  - Note that if you accidentally omit the last command, you’ll not be able to see any output on the screen, because output is still redirected to a file!

Quitting R

- Use command `q()` or menu choise File->Exit.
- R asks whether to save workspace image. If you do, all the object currently in R memory are written to a file .Rdata, and all command will be written a file .Rhistory.
- These can be loaded later, and you can continue your work from where you left it.
- Loading can be done after starting R using the manu choises File->Load Workspace and File->Load History.

In summary 1/2

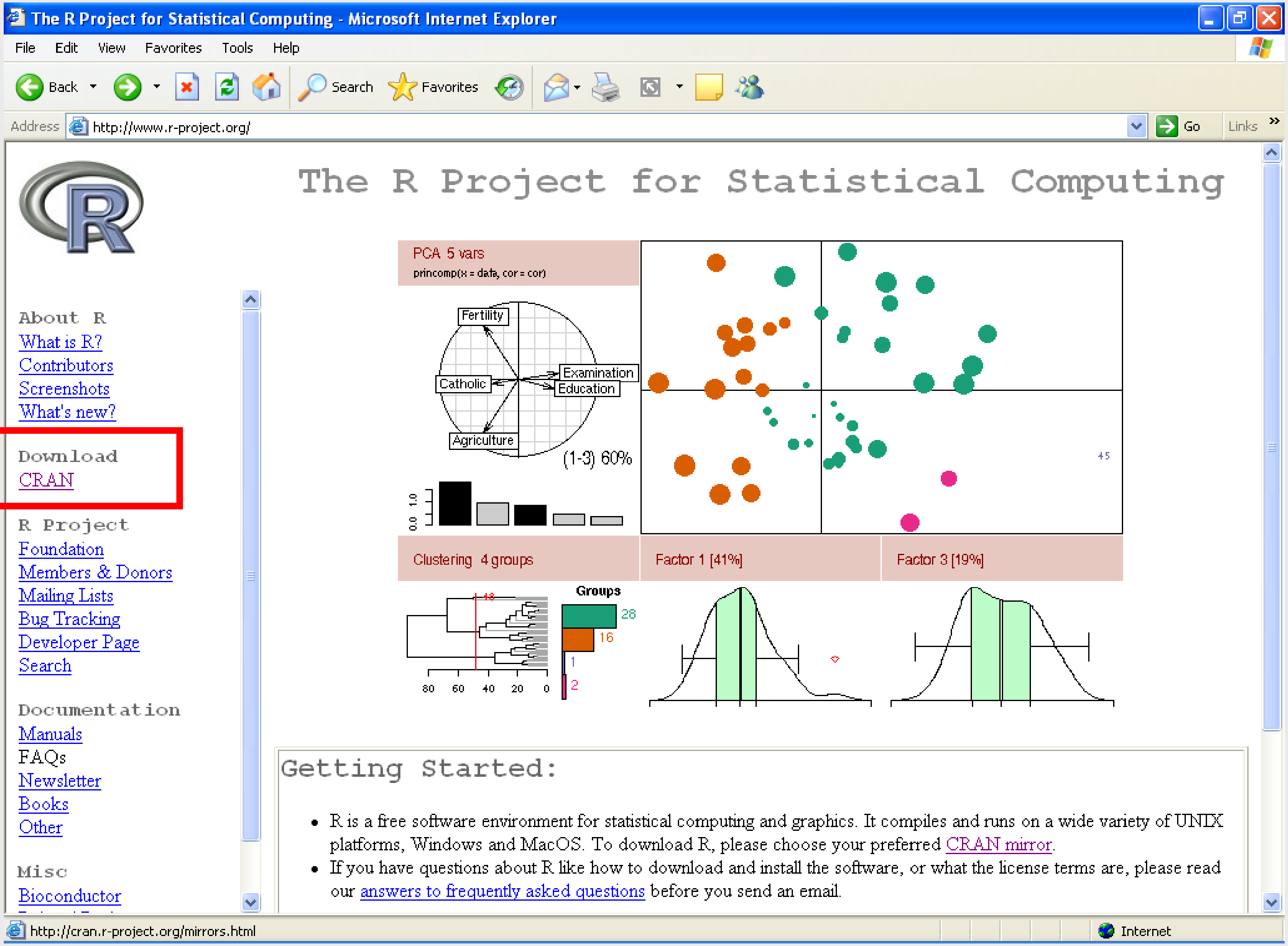
- Commands can be recognized from the brackets ”( )” that follow them. If you calculate how many bracket pairs there are, you’ll be able to identify the number of commands.
    - `pData(dat)<-pd`
- Assignment to an object is denoted by ”<-” or ”->” or ”=”. If you see a notation ”= =”, you’ll looking at a comparison operator.
    - Many other notations can be found from the documentation for the Base package or R.
- Table-like objects are often followed by square brackets ”[ ]”. Square never associate with commands, only objects.
    - `dat[,1]`
- Special characters $ and @ are used denoting individual columns in a data frame or an individual slot in a class type of an object, respectively.
    - `dat$year`
    - `dat2@cdfName`

In summary 2/2

- If you encounter a new command during the exercises, and you’d like to know what it does, please consult the documentation. All R commands are listed nowhere, and the only way to get to know new commands is to read the documentation files, so we’d like you to practise this youself.
- You’ll probably see command and notations that were not introduced in this talk. This in intentional, because we thought that these things are best handled on a situational basis. In such cases, please ask for more clarifications if needed.
- If you run into problems, please ask for help from the teachers. That’s why we are here!

Installing R

Downloading R

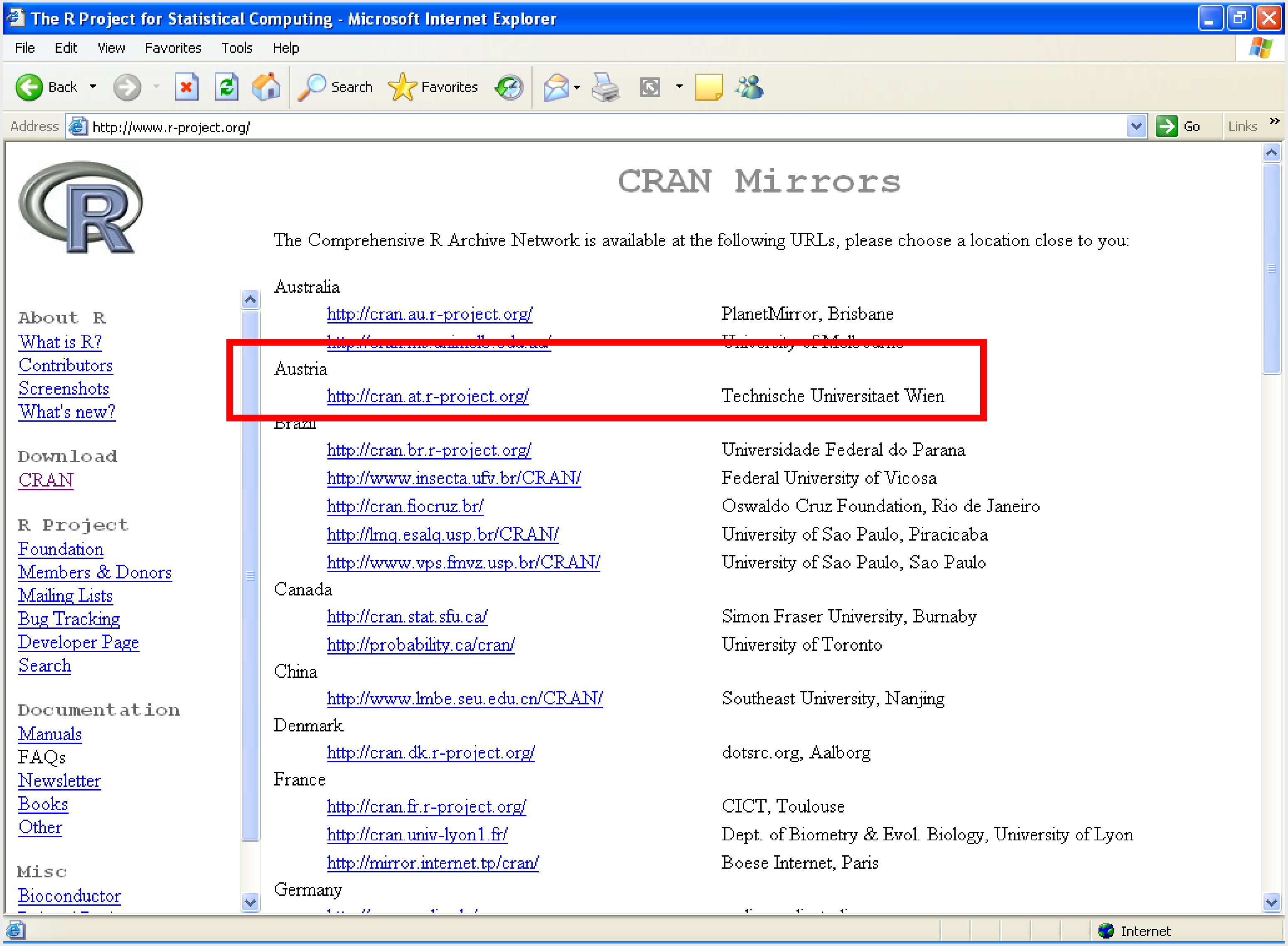


The R Project for Statistical Computing

Getting Started:

- R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To download R, please choose your preferred CRAN mirror.
- If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

Downloading R

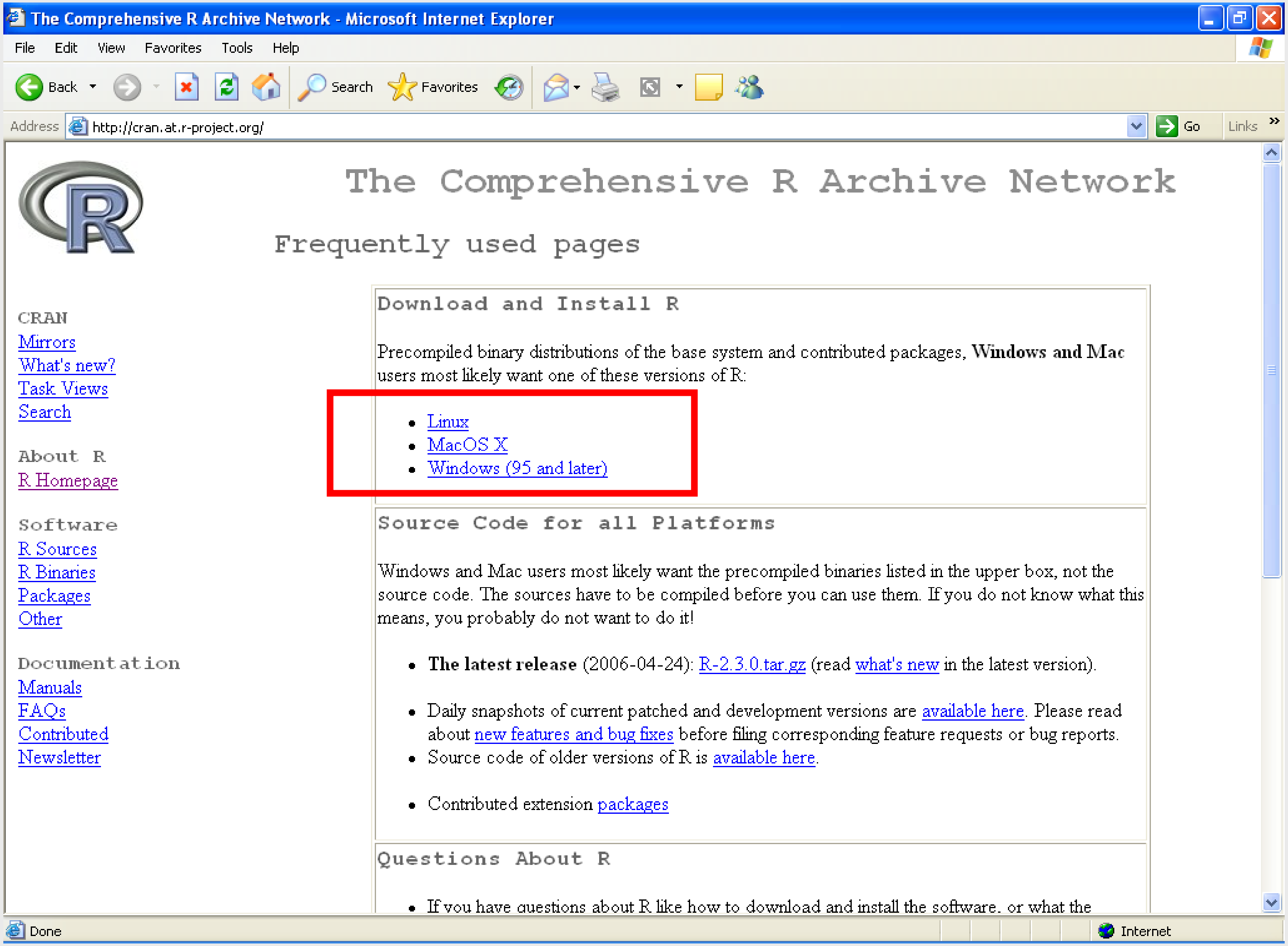


CRAN Mirrors

The Comprehensive R Archive Network is available at the following URLs, please choose a location close to you:

Australia	
http://cran.au.r-project.org/	PlanetMirror, Brisbane
http://cran.ms.unimelb.edu.au/	University of Melbourne
Austria	
http://cran.at.r-project.org/	Technische Universitaet Wien
Brazil	
http://cran.br.r-project.org/	Universidade Federal do Parana
http://www.insecta.ufv.br/CRAN/	Federal University of Vicosa
http://cran.fiocruz.br/	Oswaldo Cruz Foundation, Rio de Janeiro
http://lmq.esalq.usp.br/CRAN/	University of Sao Paulo, Piracicaba
http://www.vps.fmvz.usp.br/CRAN/	University of Sao Paulo, Sao Paulo
Canada	
http://cran.stat.sfu.ca/	Simon Fraser University, Burnaby
http://probability.ca/cran/	University of Toronto
China	
http://www.lmbe.seu.edu.cn/CRAN/	Southeast University, Nanjing
Denmark	
http://cran.dk.r-project.org/	dotsrc.org, Aalborg
France	
http://cran.fr.r-project.org/	CICT, Toulouse
http://cran.univ-lyon1.fr/	Dept. of Biometry & Evol. Biology, University of Lyon
http://mirror.internet.tp/cran/	Boese Internet, Paris
Germany	

Downloading R



The Comprehensive R Archive Network

Frequently used pages

CRAN
- Mirrors
- What's new?
- Task Views
- Search

About R
- R Homepage

Software
- R Sources
- R Binaries
- Packages
- Other

Documentation
- Manuals
- FAQs
- Contributed
- Newsletter

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- Linux
- MacOS X
- Windows (95 and later)

Source Code for all Platforms

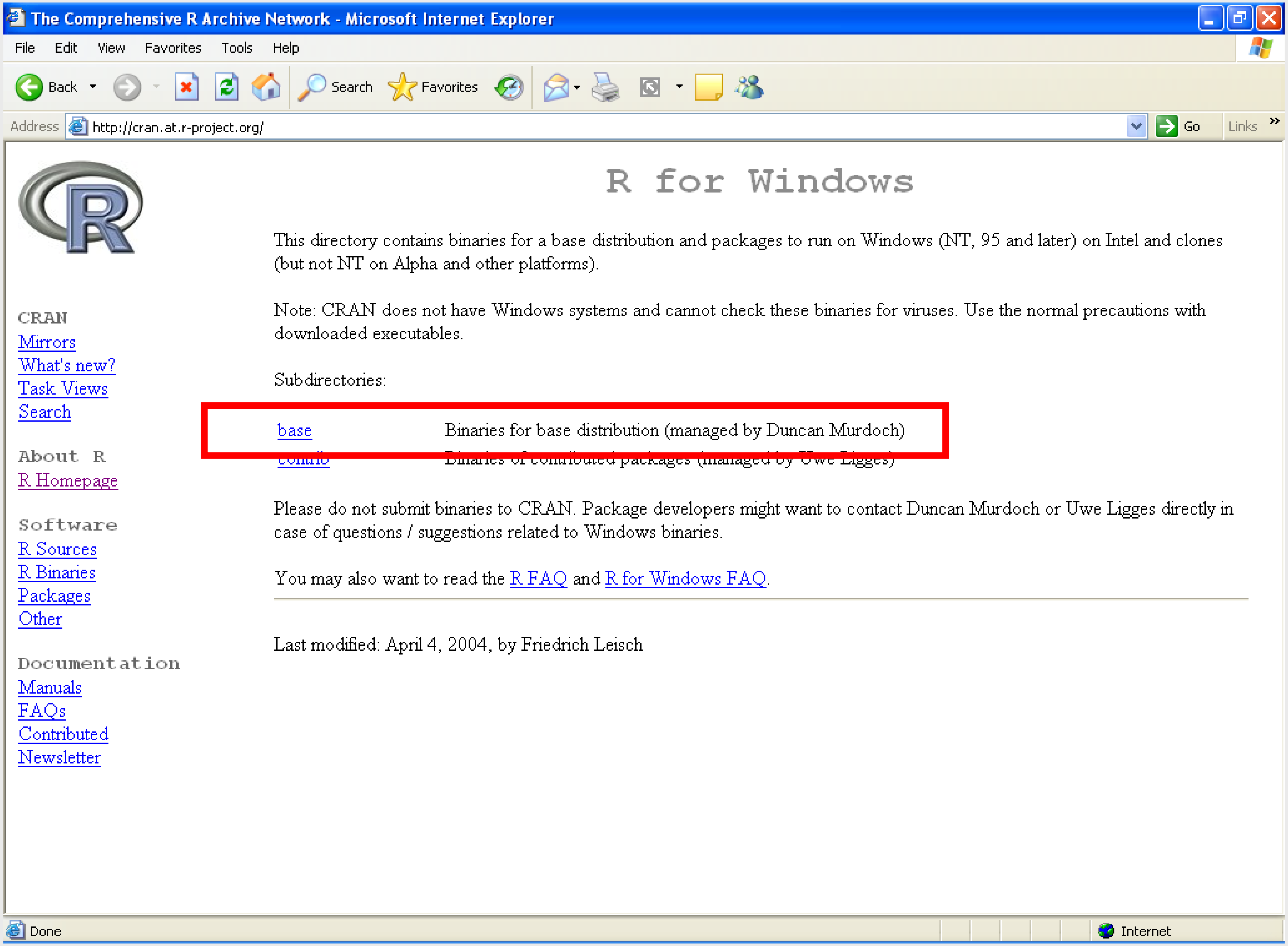
Windows and Mac users most likely want the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- **The latest release** (2006-04-24): R-2.3.0.tar.gz (read what's new in the latest version).
- Daily snapshots of current patched and development versions are available here. Please read about new features and bug fixes before filing corresponding feature requests or bug reports.
- Source code of older versions of R is available here.
- Contributed extension packages

Questions About R

- If you have questions about R like how to download and install the software, or what the

Downloading R



R for Windows

This directory contains binaries for a base distribution and packages to run on Windows (NT, 95 and later) on Intel and clones (but not NT on Alpha and other platforms).

Note: CRAN does not have Windows systems and cannot check these binaries for viruses. Use the normal precautions with downloaded executables.

Subdirectories:

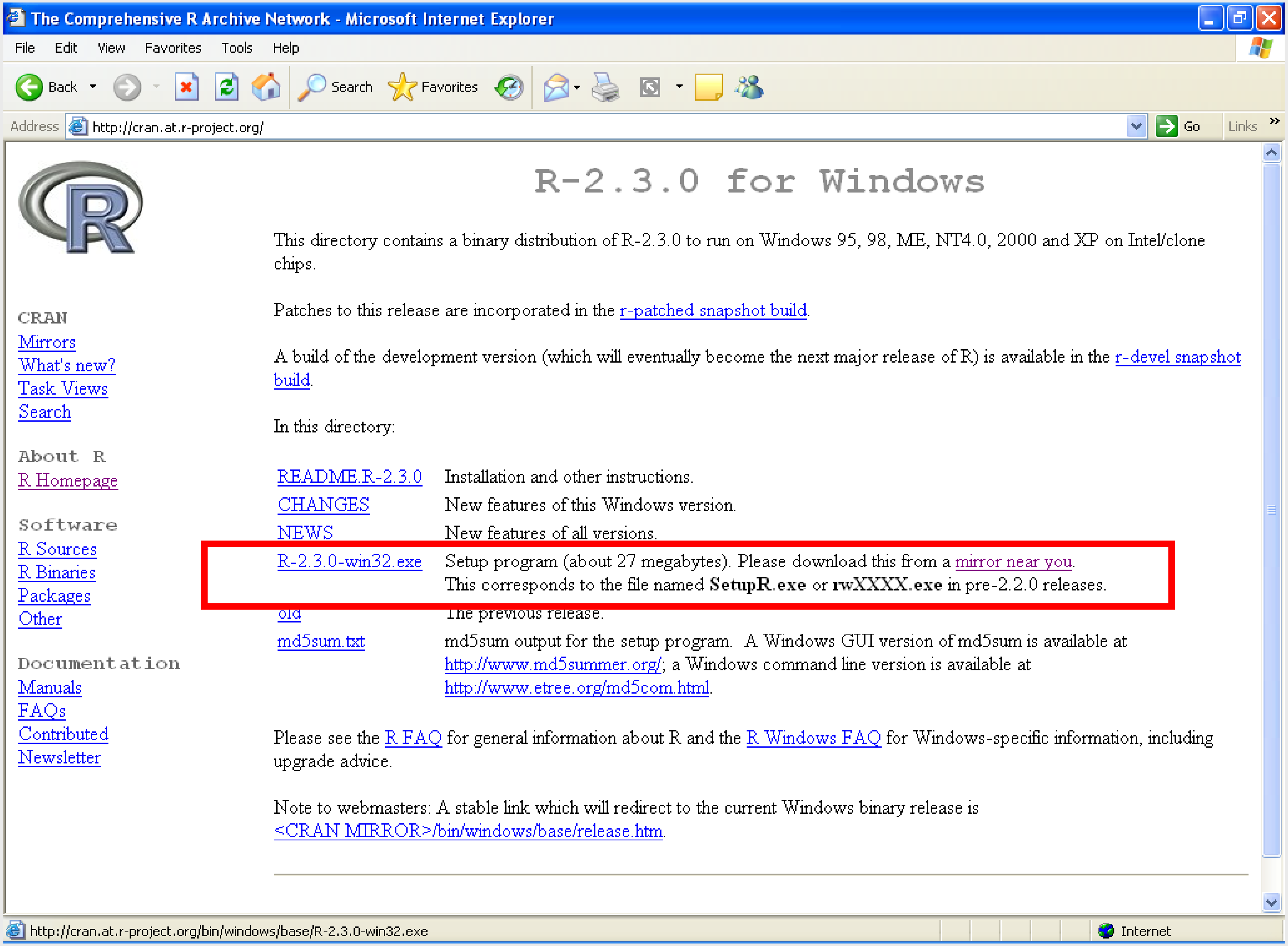
base	Binaries for base distribution (managed by Duncan Murdoch)
contrib	Binaries of contributed packages (managed by Uwe Ligges)

Please do not submit binaries to CRAN. Package developers might want to contact Duncan Murdoch or Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

You may also want to read the R FAQ and R for Windows FAQ.

Last modified: April 4, 2004, by Friedrich Leisch

Downloading R



R-2.3.0 for Windows

This directory contains a binary distribution of R-2.3.0 to run on Windows 95, 98, ME, NT4.0, 2000 and XP on Intel/clone chips.

Patches to this release are incorporated in the r-patched snapshot build.

A build of the development version (which will eventually become the next major release of R) is available in the r-devel snapshot build.

In this directory:

README.R-2.3.0	Installation and other instructions.
CHANGES	New features of this Windows version.
NEWS	New features of all versions.
R-2.3.0-win32.exe	Setup program (about 27 megabytes). Please download this from a mirror near you. This corresponds to the file named **SetupR.exe** or **rwXXXX.exe** in pre-2.2.0 releases.
old	The previous release.
md5sum.txt	md5sum output for the setup program. A Windows GUI version of md5sum is available at http://www.md5summer.org/; a Windows command line version is available at http://www.etree.org/md5com.html.

Please see the R FAQ for general information about R and the R Windows FAQ for Windows-specific information, including upgrade advice.

Note to webmasters: A stable link which will redirect to the current Windows binary release is <CRAN MIRROR>/bin/windows/base/release.htm.

Installing R for Windows

- Execute the R-2.3.0-win32.exe with administrator privileges
- Once the program is installed, run the R program by clicking on its icon

- R 2.2.1 with Bioconductor 1.7.0 is installed on corona.csc.fi, also
- R 2.3.1 is in works

Downloading Bioconductor

BioConductor is an open source and open development software project for the analysis and comprehension of genomic data.

project news

- 2006-04-06
  Changes in BioC Devel, March 2006
- 2006-02-08
  Changes in BioC Devel, February 2006

More...

QUICK LINKS

- What is it?
- Install - How To
- FAQ
- For Developers
- BioC Workshops
- How to up-load packages
- Metadata
- BioC News

Latest News

Bioconductor 1.8 released 27 April, 2006. This release is designed for R 2.3.0. View the packages here

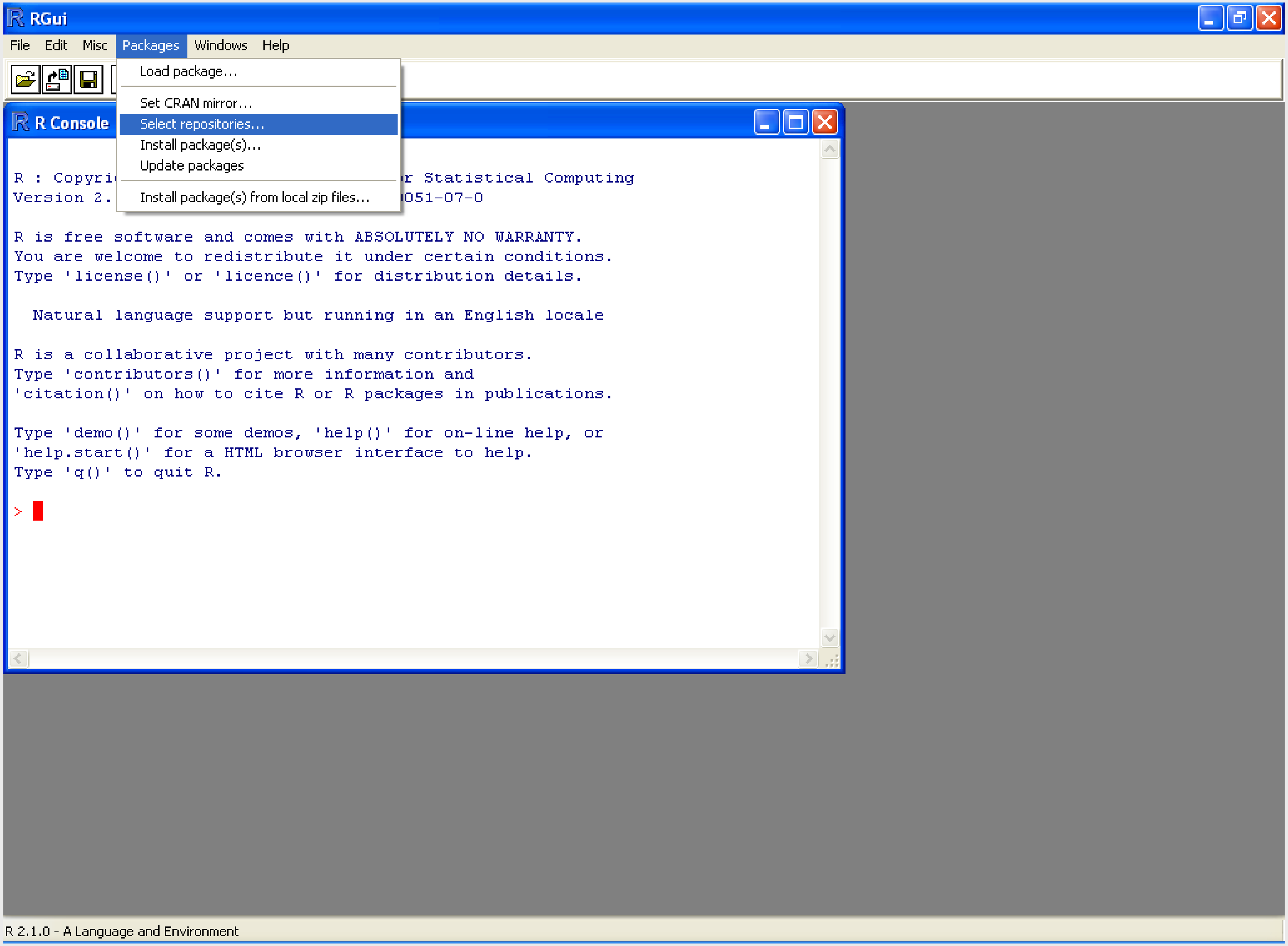
Upcoming: Computational and Statistical short course on Microarrays.
18 June - 23 June, 2006, Brixen-Bressanone (BZ), Italy.

Upcoming: Workshops on S, R and Bioconductor.
7 July - 8 July 2006, Auckland, New Zealand.

Annoucement: BioC2006 Conference

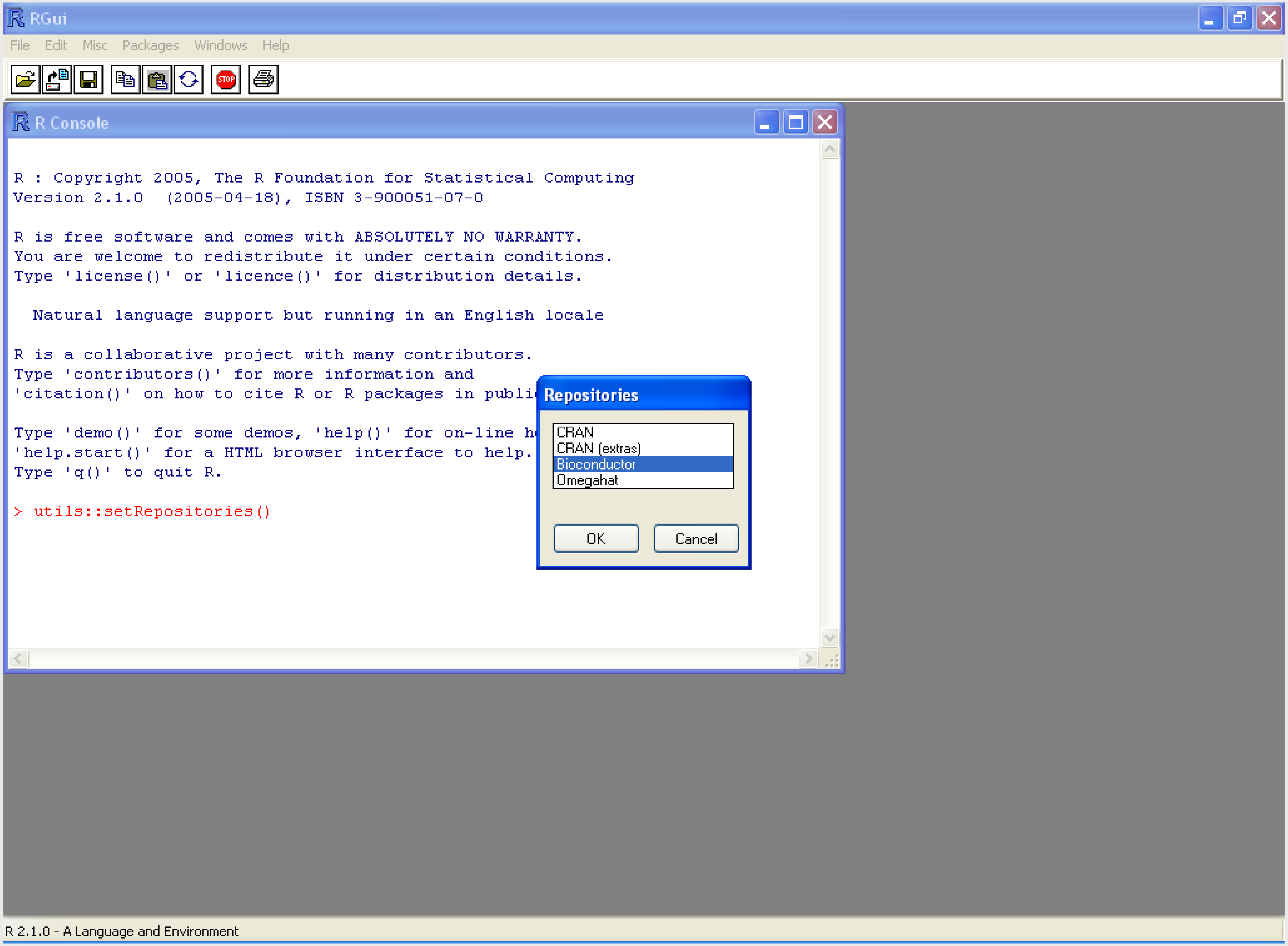
Bioconductor User and Developer Conference
August 3-4, 2006
Seattle, WA, USA
Detailed information

Installing Bioconductor



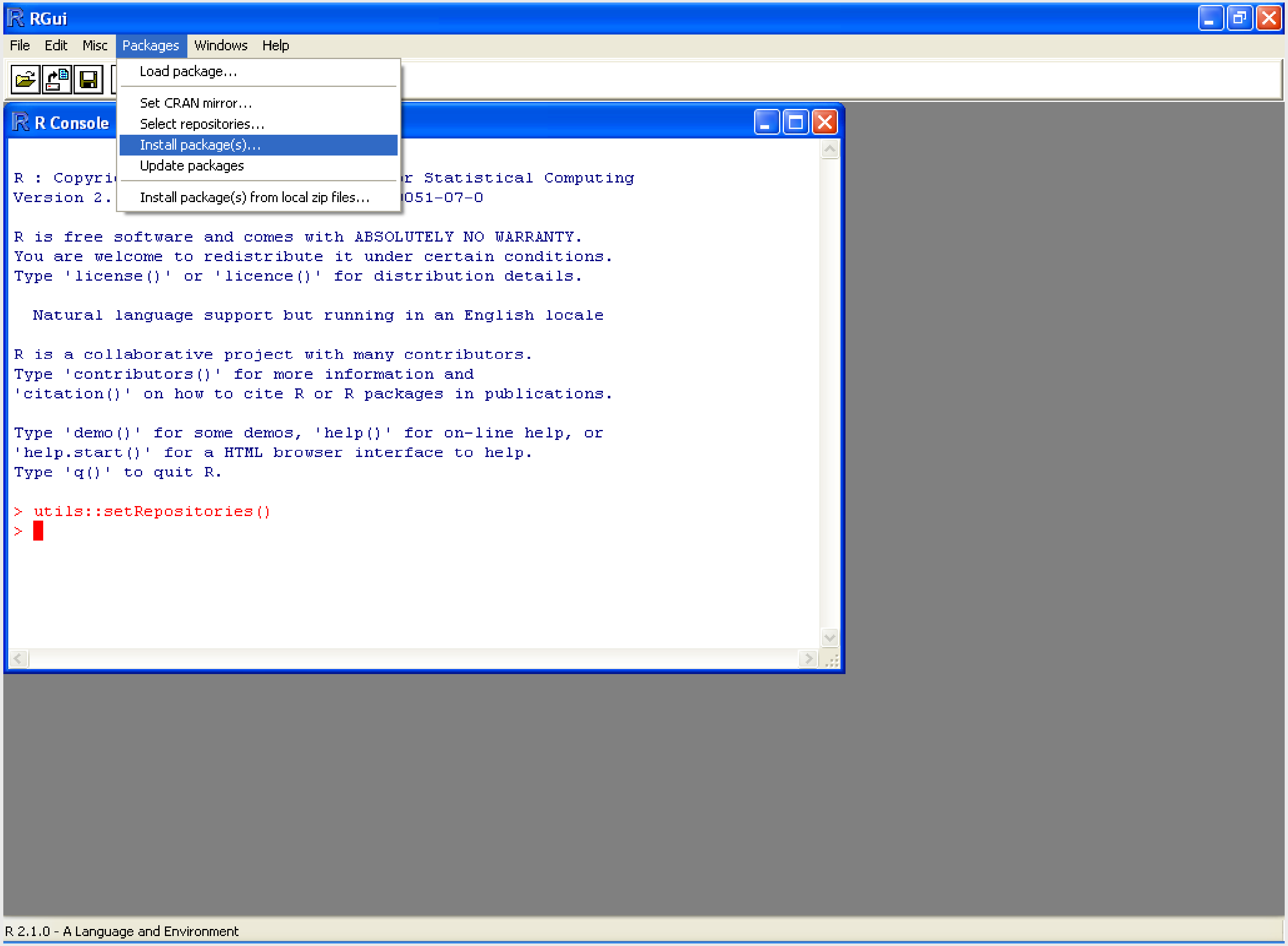
```
R : Copyri                                r Statistical Computing
Version 2.                                051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for a HTML browser interface to help.
Type 'q()' to quit R.

>
```

Installing Bioconductor



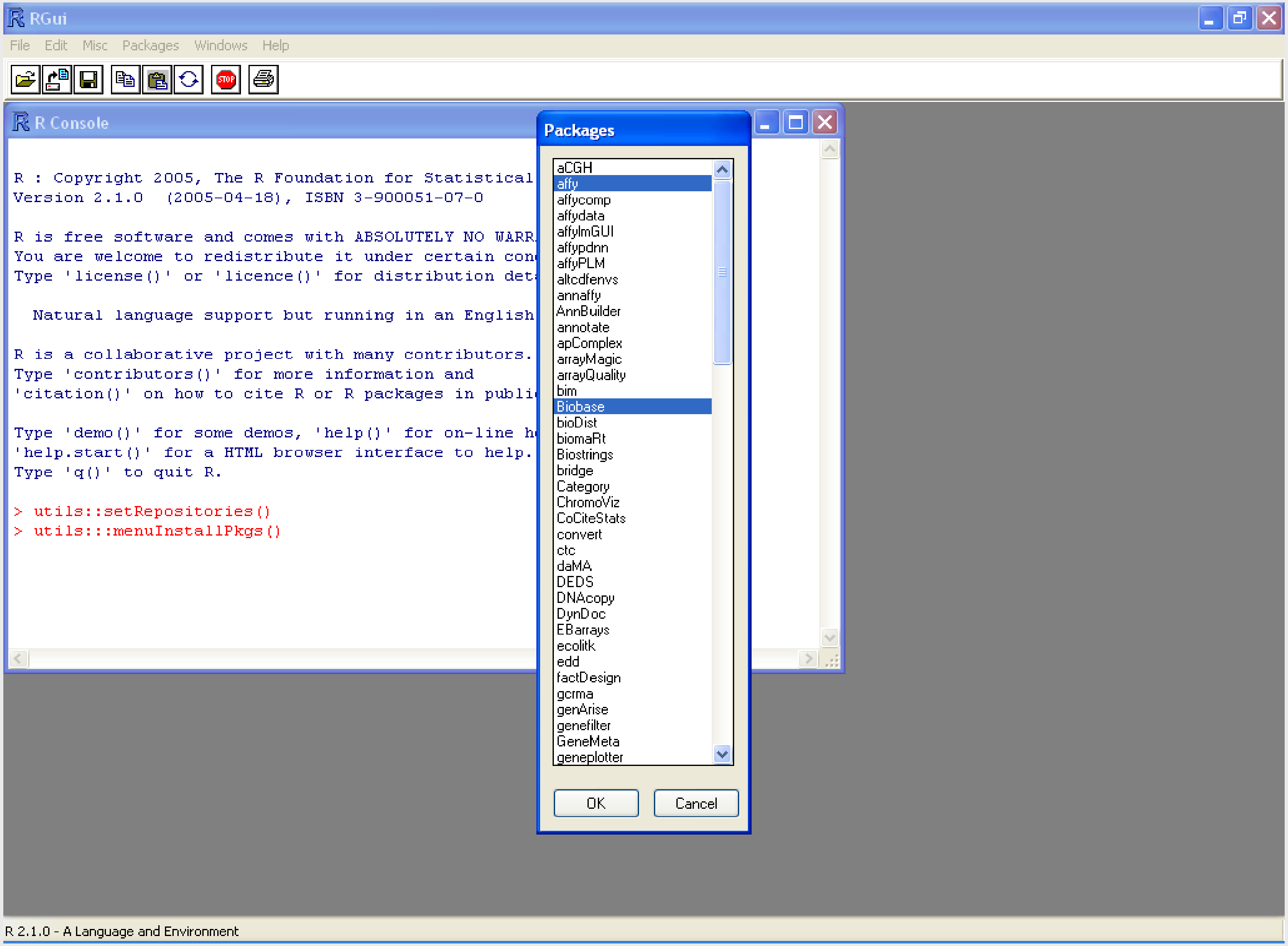
```
R : Copyright 2005, The R Foundation for Statistical Computing
Version 2.1.0  (2005-04-18), ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publi

Type 'demo()' for some demos, 'help()' for on-line h
'help.start()' for a HTML browser interface to help.
Type 'q()' to quit R.

> utils::setRepositories()
```

Installing Bioconductor



```
R : Copyri                          r Statistical Computing
Version 2.                          051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for a HTML browser interface to help.
Type 'q()' to quit R.

> utils::setRepositories()
> 
```

Installing Bioconductor



```
R : Copyright 2005, The R Foundation for Statistical
Version 2.1.0  (2005-04-18), ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARR
You are welcome to redistribute it under certain con
Type 'license()' or 'licence()' for distribution det

  Natural language support but running in an English

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publi

Type 'demo()' for some demos, 'help()' for on-line h
'help.start()' for a HTML browser interface to help.
Type 'q()' to quit R.

> utils::setRepositories()
> utils:::menuInstallPkgs()
```

Installing Bioconductor (the best way)

- Alternatively, you can install Bioconductor using a script:

```
source("http://www.bioconductor.org/biocLite.R")
biocLite()

biocLite(c(” "hgu133a", "hgu133acdf",
  "hgu133aprobe", "ygs98", "ygs98cdf",
  "ygs98probe")
```

Reference

- 1. Shalev-Shwartz and Ben-David. Understanding Machine Learning: From Theory to Algorithms (Cambridge University Press, 2014)
- 2. Daum´e. A Course in Machine Learning.
- 3. The Art of Statistics: How to Learn from Data by David Shpigelter
- 4. Learning From Data – January 1, 2012 by Yaser S. Abu-Mostafa (Author), Malik Magdon-Ismail (Author), Hsuan-Tien Lin (Author)
- 5. Statistics: The Art and Science of Learning from Data by Alan Agresti
- 6. Learning From Data: An Introduction To Statistical Reasoning by M.Glenber.
- 7. Statistics: Learning from Data (with JMP Printed Access Card) by Rocky Pek
- 8. The Elements of Statistical Learning by Gerim Garold
- 9. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems 2nd Edition
- by Aurélien Géron (Author)