Getting Started with R

August 20, 2013

Outline

- **Introduction**
  - **What is R ?**
  - **Install R**
- **R Console and Script Editor**
  - **Assigning objects and functions**
  - **R as a calculator**
- **Documentation**

What is R?

- R is an open-source environment for statistical computing and visualization. It is the product of an active movement among statisticians for a powerful, programmable, portable, and open computing environment, applicable to the most complex and sophisticated problems, as well as "routine" analysis, without any restrictions on access or use.
  - Performs a variety of simple and advanced statistical methods
  - Produces high quality graphics
  - R is a computer language so we can write new functions that extends R's uses
- R was initially written by Ross Ihaka and Robert Gentleman at the University of Auckland in Auckland, New Zealand (hence the name).
- The official R home page is http://www.R-project.org

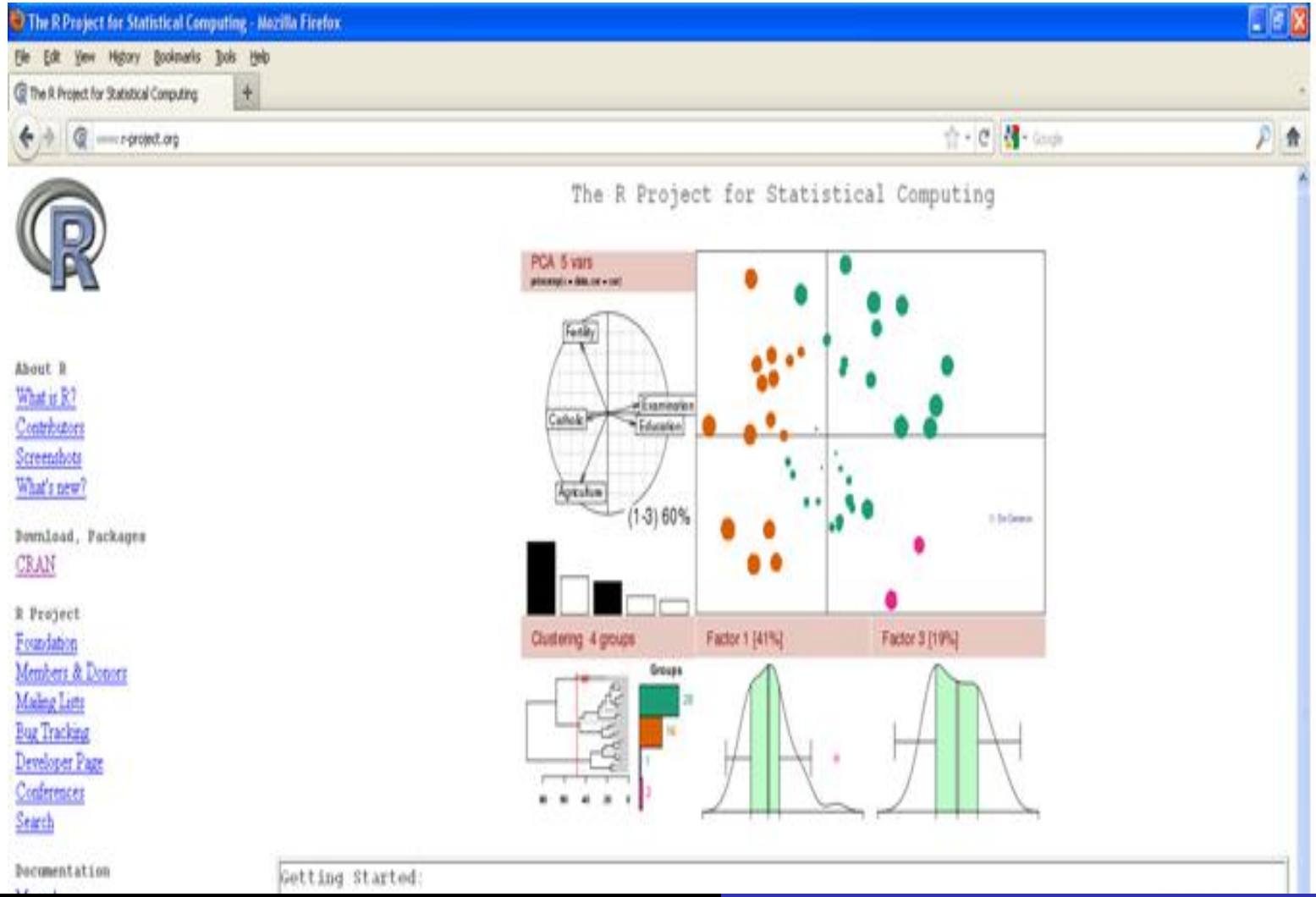
Install R

- The R system can be installed on, Windows, Mac or Linux
- You will want to install the base system.
- In addition to the base system there are user contributed add-on packages. Packages are a collection of functions, examples and documentation that usually focus on a specific task.
- The base system contains some packages. To install an additional package, say survival, be connected to the Internet and type
  $>$ install.packages("survival")
- You will be asked to select the mirror site nearest to you, after that everything is automatic.
- Before using the contents of the package we need to load it,
  $>$ library(survival)
- See the R website for a complete list of contributed packages

Installing R

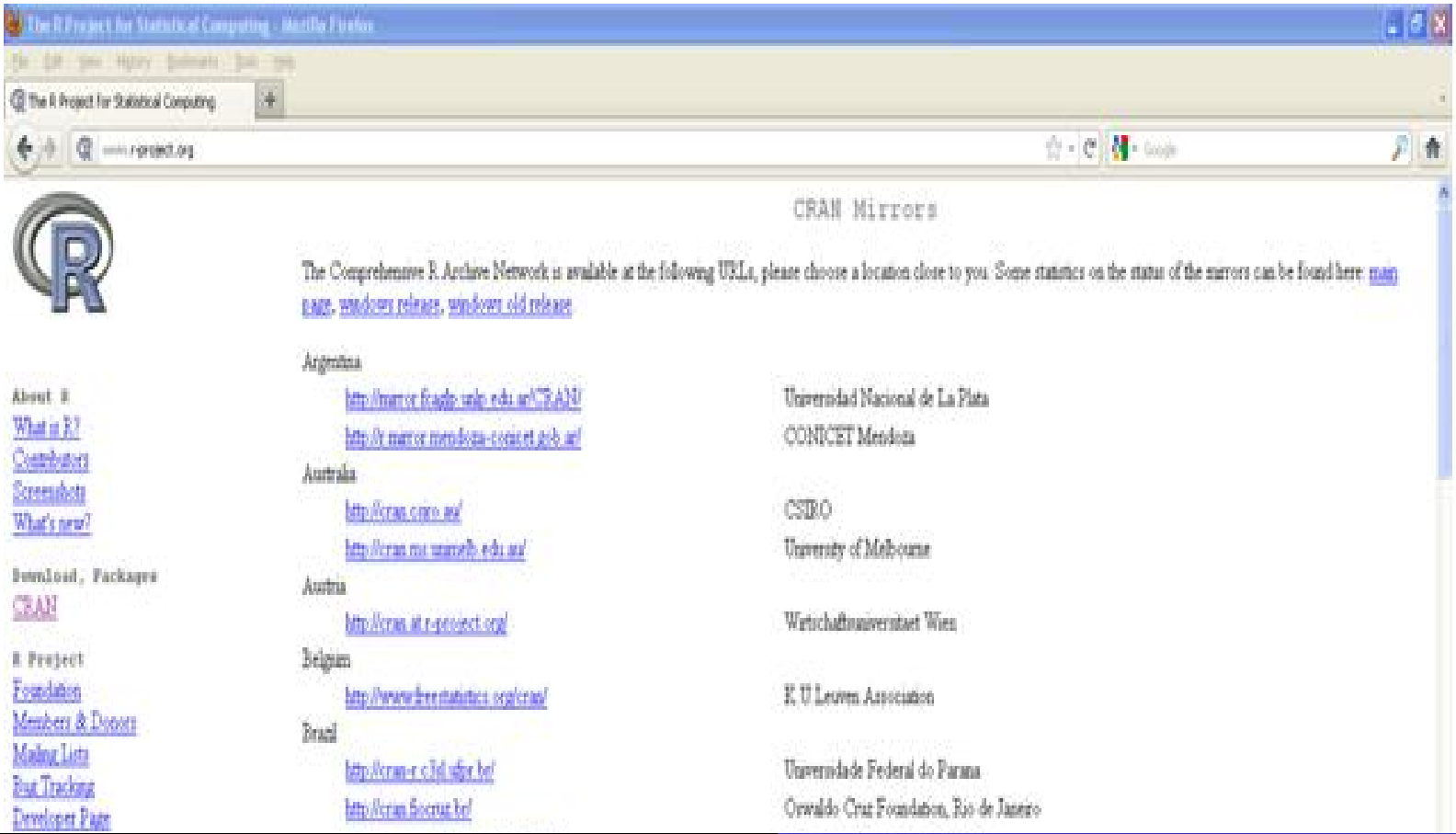
Visit www.r-project.org

The homepage appears as below.



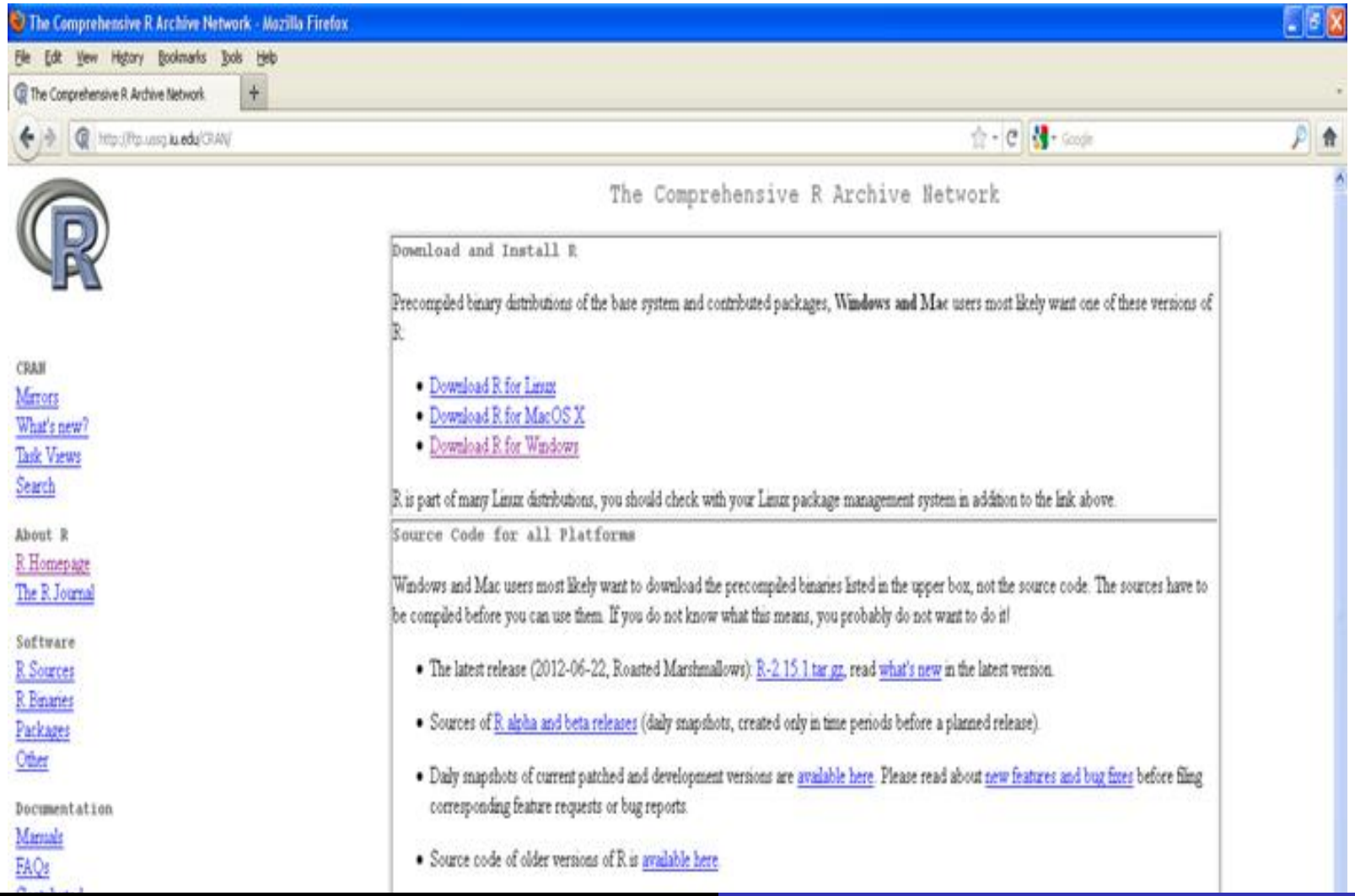
Installing R

The Comprehensive R Archive Network (CRAN) allows selection of a regional computer network from which you can download R. If you click on the CRAN link, you will be shown a list of network servers all over the world. Choose the mirror site closer to you.



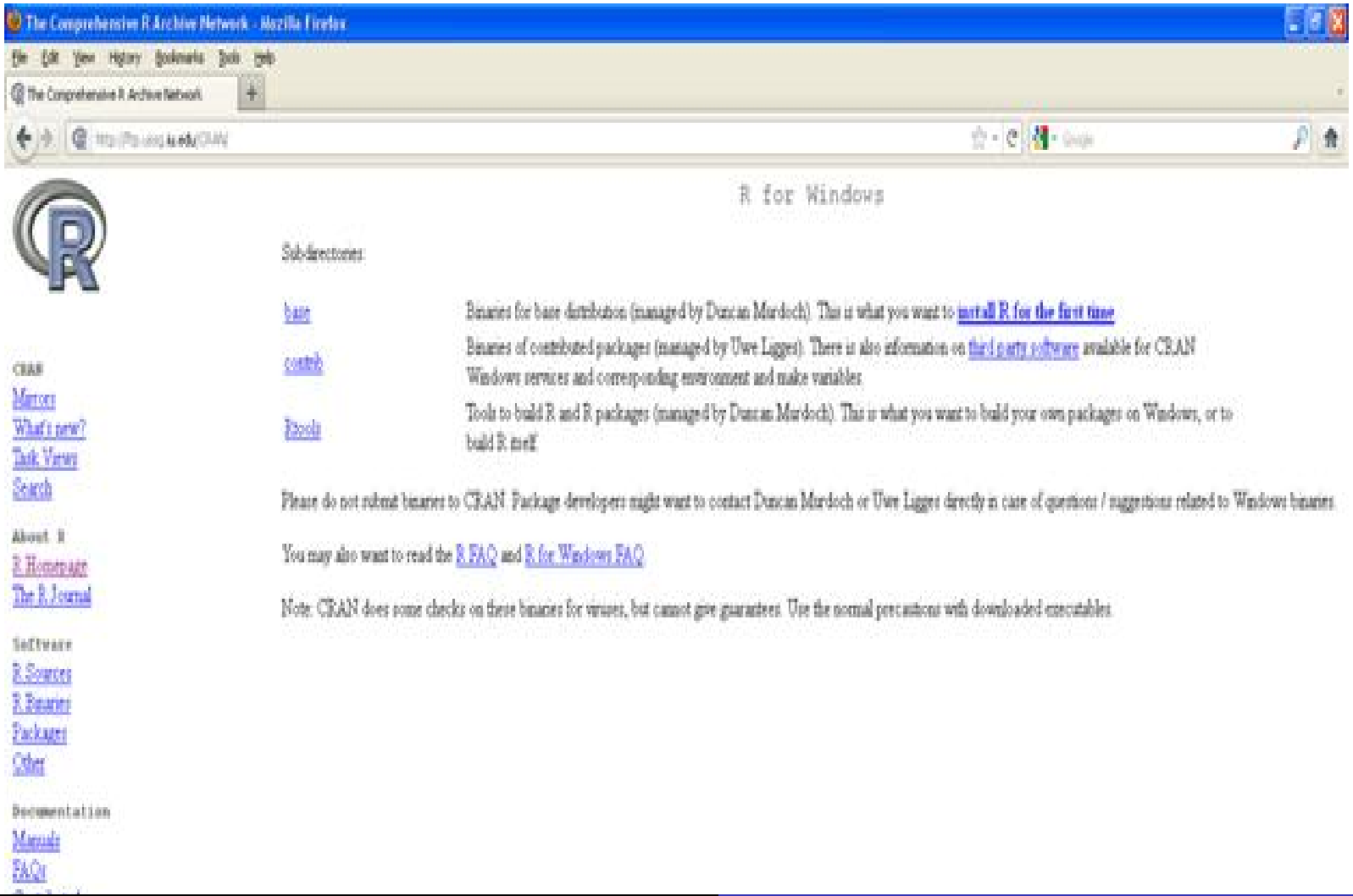
Installing R

If we choose Indiana University the webpage shown below will appear.



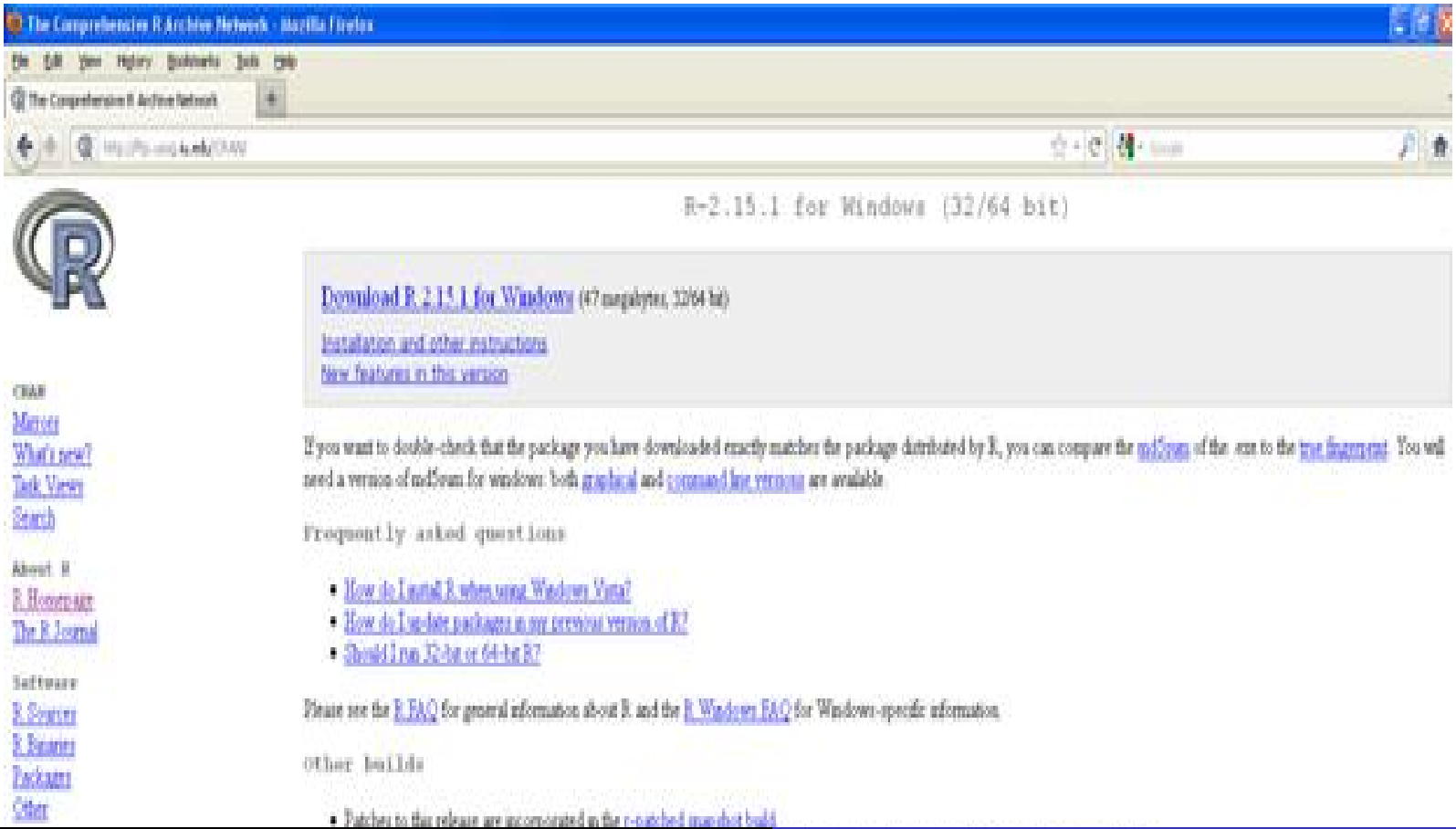
Installing R

If you click on Download R for Windows the page below will appear.



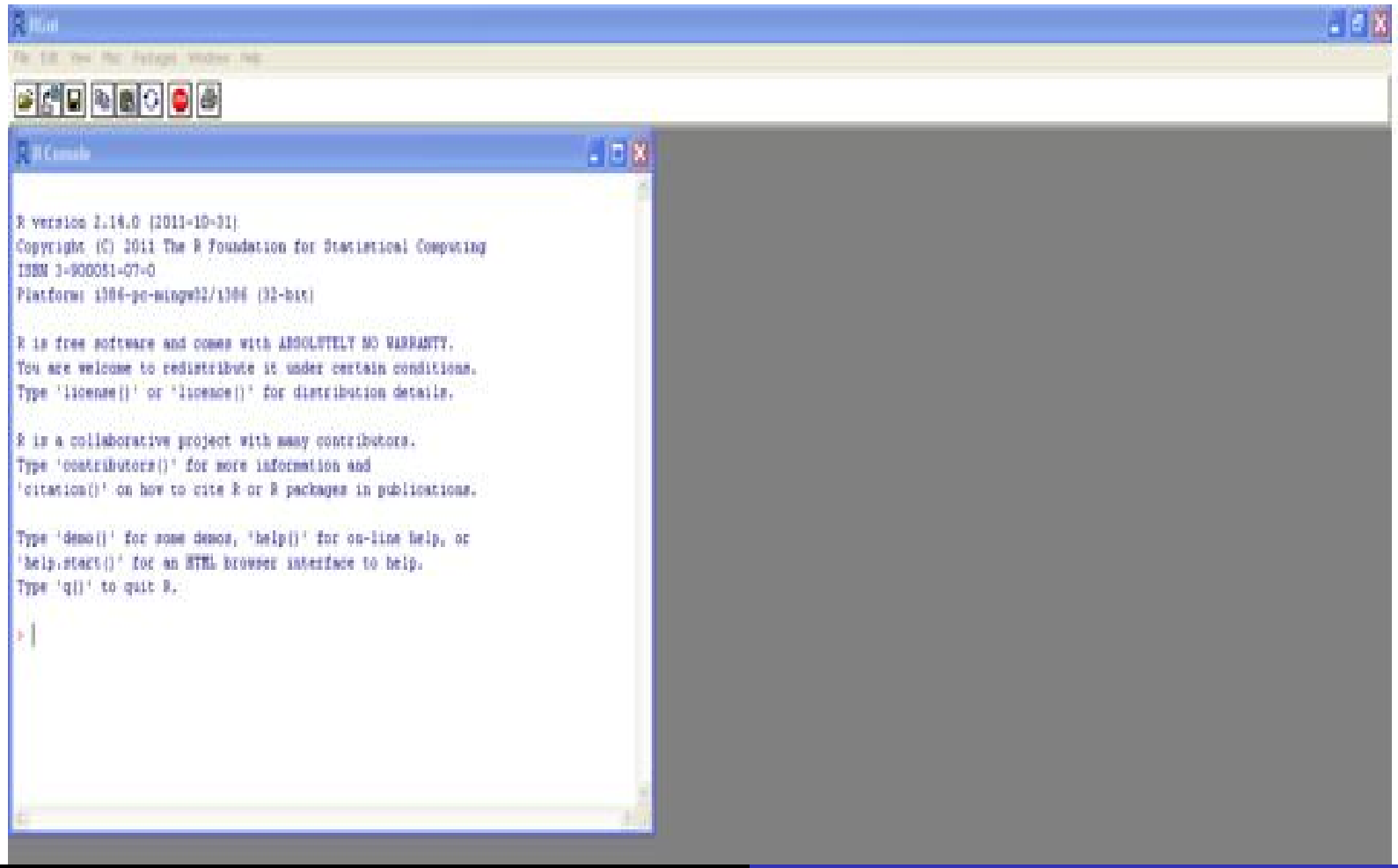
Installing R

If you click on base the page below will appear where you can download Download R 2.15.1 for Windows which is 47 megabytes, 32/64 bit. Note that this is the latest version at the time of writing this note , and you may see a more recent versions.



Installing R

R can be started by double-clicking the desktop shortcut icon R or by going to Start$->$Program$->$ *R*.
The R startup window appears as below



R Console

The R Console is where computations are performed. After starting R, you will be looking at a console where you interact with R

- An expression is inputed into the console and the expression is evaluated. Depending on the expression, the system may respond by outputting results to the console or creating a graph in a new window. Then another expression is inputed and evaluated.
- An R session is this interaction between you and the system
- To get the last expression entered use the up arrow
- To get the value of the last evaluated expression type .Last.value
- Press Esc to stop evaluating the current expression

R as a calculator

Enter a math expression and the result is displayed to the console

Binary Operators	$+$	$-$	$*$	$/$	$\wedge$	%%
Math Functions	abs	sqrt	log	exp	log10	factorial
Trig Functions	sin	cos	tan	asin	acos	atan
Rounding	round	ceiling	floor	trunc	signif	zapsmall

Example:

```
> 5+4
        [1] 9
> 6*7
        [1] 42
> 7^3
        [1] 343
> 20-15
        [1] 5
> 20%%7
        [1] 6
> ##%% is for modular arithmetic
> abs(5)
        [1] 5
> abs(-5)
        [1] 5
> factorial(4)
        [1] 24
```

Objects and Functions

In R We normally create objects and then perform functions on those objects
For example, Assign an object a name "x" using either
x $<-$ object
x $=$ object
Call a function by
function name(list of arguments separated by commas) : Example: mean, median, var, summary etc.

- Each function has a set of formal arguments some with default values
- A function call can include any subset of the complete argument list
- When specifying values for an argument use an $=$.

R is CASE SENSITIVE

Example

Suppose we want to calculate the mean of the scores 0, 5, 7, 9, 1, 2, 8. First we assign the vector of numbers a name say "x" and then call the function mean().

```
>  x = c(0,5,7,9,1,2,8)
> mean(x)
[1] 4.571429
> mean(X)
Error in mean(X) : object 'X' not found
> Mean(x)
Error: could not find function "Mean"
```

Example

Suppose we want to sort a vector y so that the numbers are descending. By default R will sort ascending so I need to change the formal argument decreasing to TRUE (the default value for decreasing is FALSE)

```
> y <- c(4,2,0,9,5,3,10)
> y
[1]  4  2  0  9  5  3 10
> sort(y)
[1]  0  2  3  4  5  9 10
> sort(y,decreasing=TRUE)
[1] 10  9  5  4  3  2  0
```

Script Editor

The script editor is used for writing programs in R.

- To start a new script, click File > New Script
- The easiest way to run code is keyboard shortcuts
- To run part of a program, highlight the lines you want and hit Ctrl+R
- To run an entire program, select all the code then run, Ctrl+A then Ctrl+R
- To comment a line of code use a #

Comments are notes that help users understand portions of your code. They are also useful reminders to yourself of what was done and why it was done. Including meaningful comments in code is a major part of writing good programs, this semester we will practice commenting our programs.

Working Directory

When we load/save datasets, load source files or save graphs we will need to specify the file path. To avoid typing the path every time we can specify a working directory.
To set the working directory click File $>$ Change dir... or type
$>$setwd(file path)

Package Documentations

Packages are collections of R functions, data, and compiled code in a well-defined format. The directory where packages are stored is called the library. R comes with a standard set of packages. Others are available for download and installation. Once installed, they have to be loaded into the session to be used.

```
.libPaths() # get library location
library() # see all packages installed
search() # see packages currently loaded
```

Adding Packages

One can expand the types of analyses by adding other packages. A complete list of contributed packages is available from CRAN.
on MS Windows

- Choose Install Packages from the Packages menu.
- Select a CRAN Mirror. (e.g. USA(IN))
- Select a package. (e.g. survival)
- Then use the library(package) function to load it for use. (e.g. library(survival))

Workspace

- The workspace is where all the objects you create during a session are located.
- When you close R you will be prompted to "Save workspace image?" If you say yes, the next time you open R from the same working directory the workspace will be restored. That is all of the objects you created during your last session will still be available. Also see save.image()
- Depending on what you are working on, it is usually wiser to write a script and then run the entire script with each new session. This way you know exactly what objects you have created; sometimes lingering objects can cause errors in programs. If a particular object is very important then save it to a file.

Managing the Workspace

- To list what variables you have created in the current session, ls()
- To see what libraries and dataframes are loaded, search()
- To see what libraries have been installed, library()
- To remove an object, rm(object names)
- To remove all objects, rm(list=ls())

THANK YOU