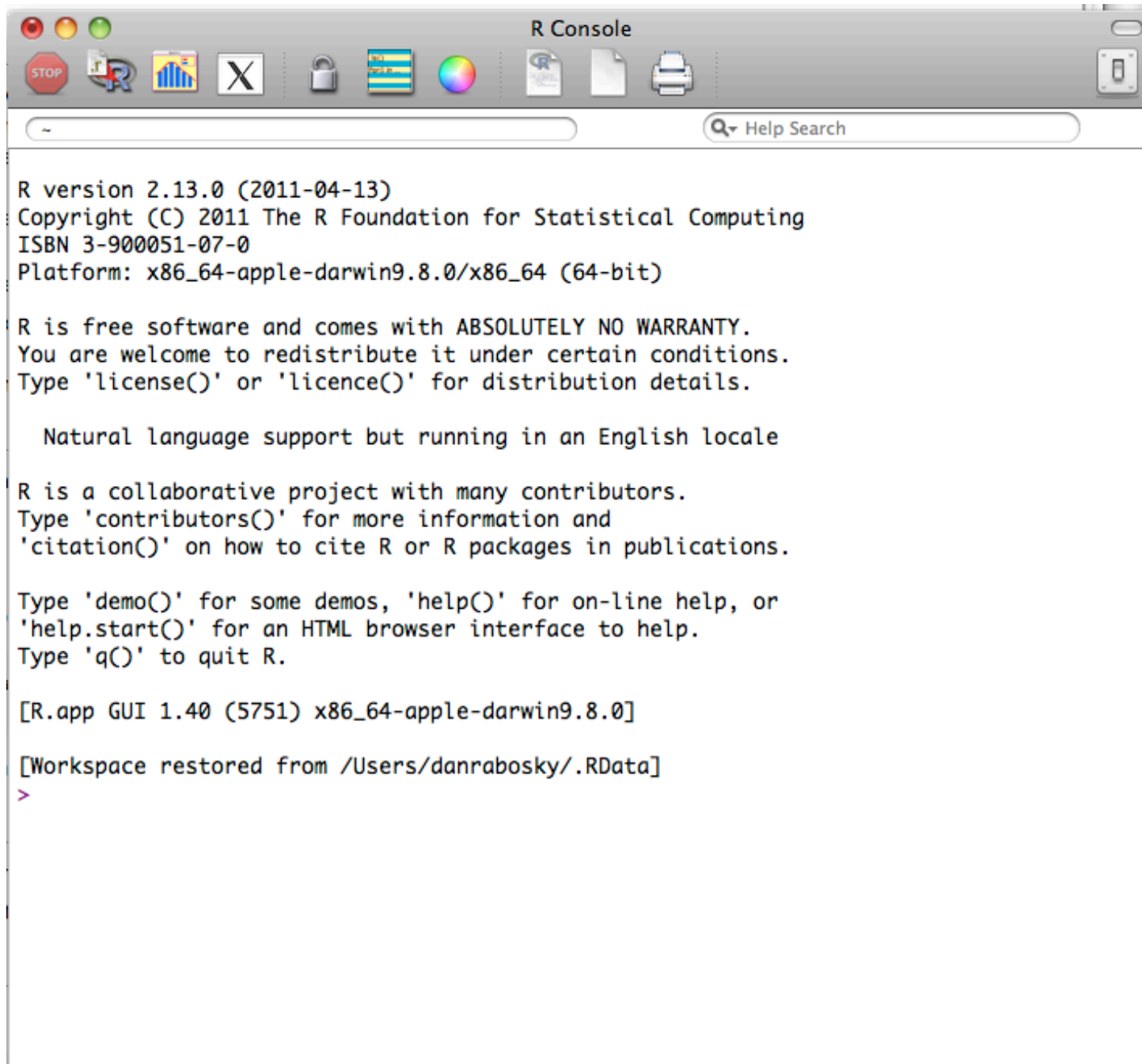# R: a multipurpose scientific programming environment for 95% of ecology & evolution

- Data manipulation
- Basic statistics
- Advanced statistics
- **Programming: simulation and modeling of data**
- Scripting: save and/or repeat your analyses exactly.
- Scientific graphics
- Speed: from **idea** to **results** in *hours* or less!

Limitations on R

- Learning curve

- Memory: limits on virtual memory

- Computational speed:
  - *some problems are not R friendly*

Getting started with R

R version 2.13.0 (2011-04-13)
Copyright (C) 2011 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-apple-darwin9.8.0/x86_64 (64-bit)

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 an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.40 (5751) x86_64-apple-darwin9.8.0]

[Workspace restored from /Users/danrabosky/.RData]
>

This is the R editor. Launch with CMD+N [mac]
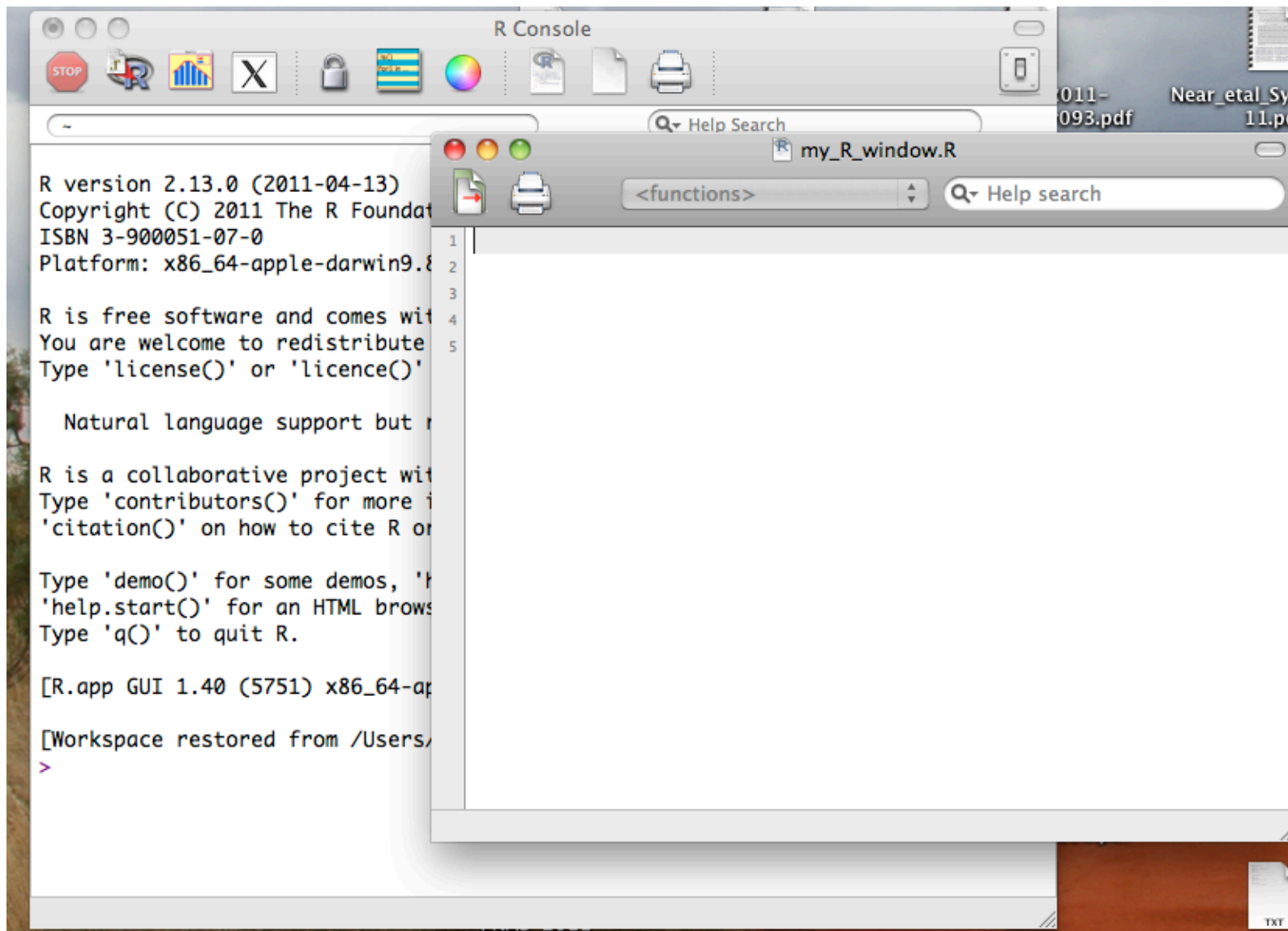
Structure of your R environment

Preloaded base stuff

Structure of your R environment

Preloaded base stuff

Datatypes & structures:
`numeric, character, factor, data.frame`

Functions: `dnorm(), lm(), list(), rm()`

Programming keywords:
`for, function, while, in`

Structure of your R environment

Preloaded base stuff

Stuff you create

Structure of your R environment

Preloaded base stuff

Variables you create:
```
x <- runif(10)
```

Data you load:
```
myData <- scan('mydata.txt')
```

Stuff you create

Structure of your R environment

Preloaded base stuff

Other people's stuff:
- packages and data
- "libraries"

Structure of your R environment

Preloaded base stuff

Other people's stuff:
- packages and data
- "libraries"

```
>library(ape)
>data(bird.orders)
>plot.phylo(bird.orders)
```

Structure of your R environment

Preloaded base stuff

Other people's stuff:
- packages and data
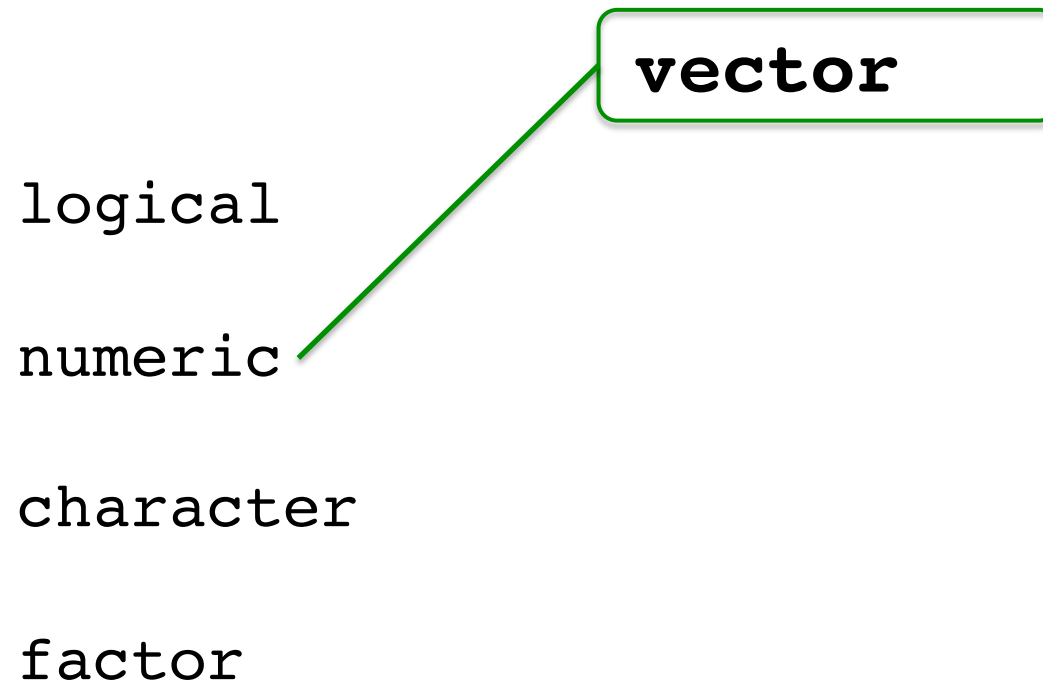**- "libraries"**

Stuff you create
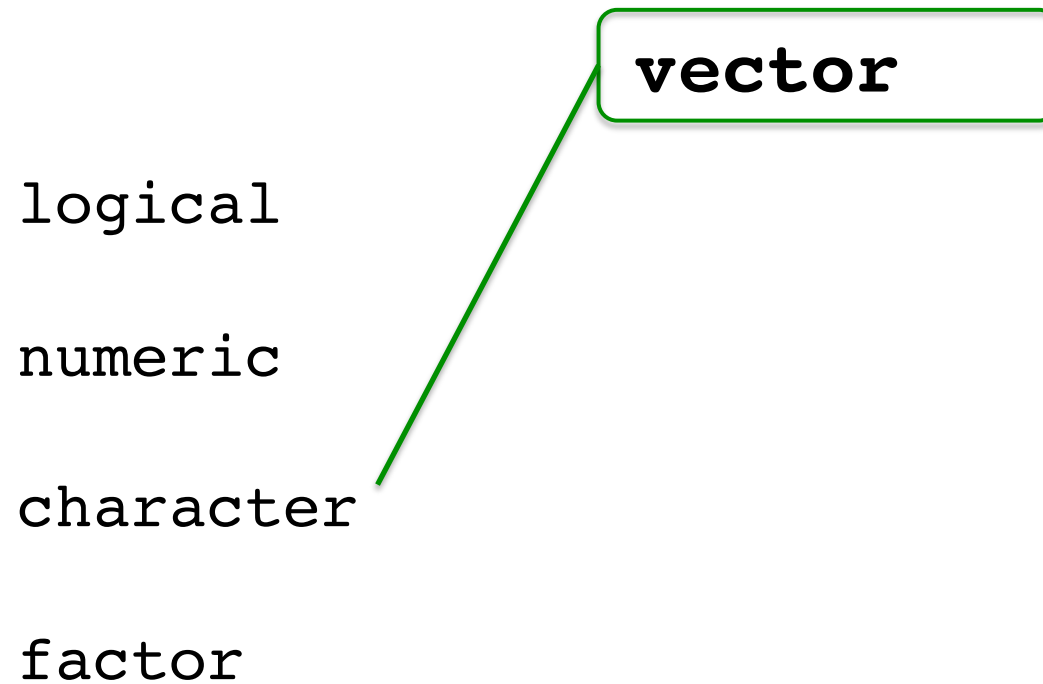
Basic data types

```
logical

numeric

character

factor
```

Basic data structures

vector

logical

numeric

character

factor

Basic data structures

vector

logical

numeric

character

factor

Data structures

logical

numeric

character

factor

vector

matrix
array

Data structures

logical

numeric

character

factor

vector

matrix
array

data.frame

Data structures

logical

numeric

character

factor

vector

matrix
array
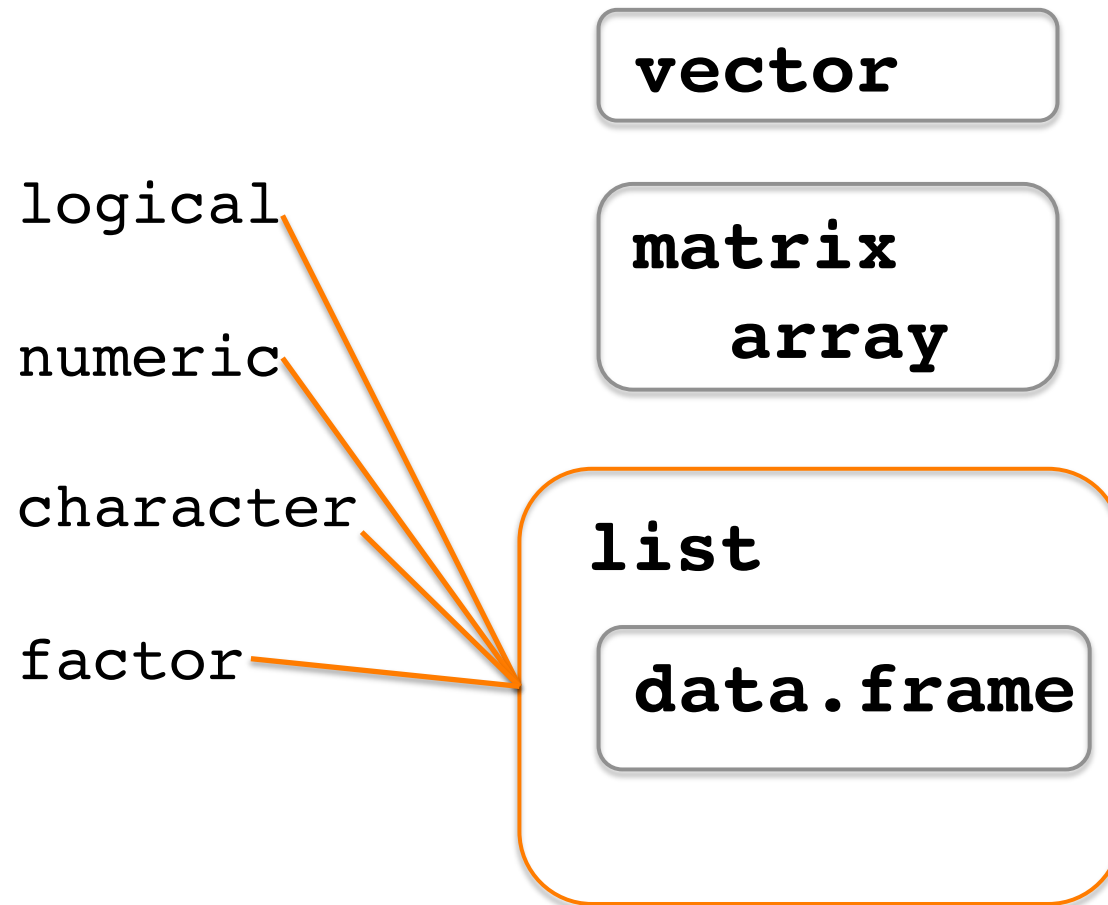
list

data.frame

Perform operations on data

logical

numeric

character

factor

vector

matrix
array

function
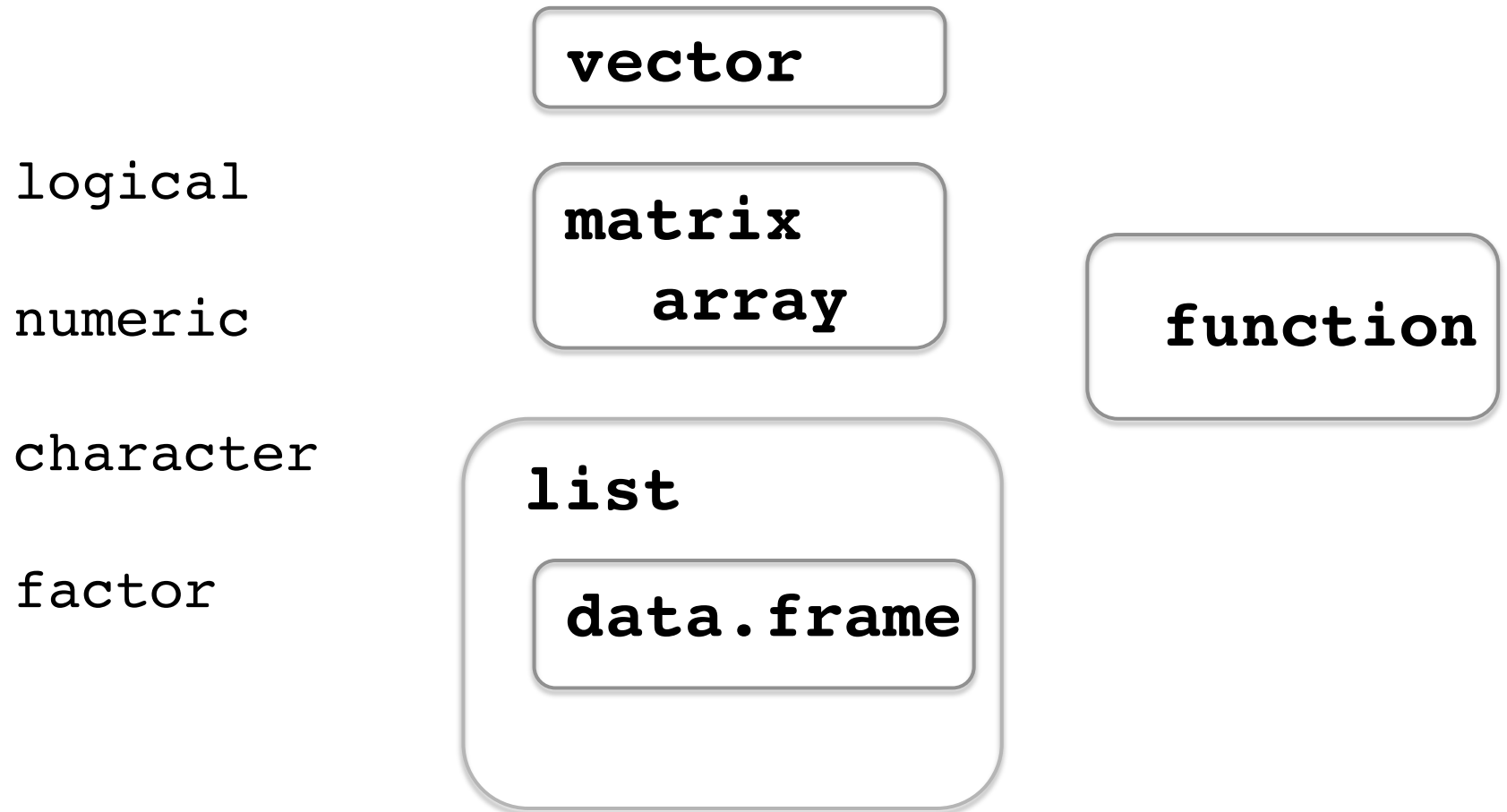
list

data.frame

Data types

logical

numeric

character

factor

Data "containers"

vector

**matrix
array**

list

data.frame

Operations

function

R commandments

Be aware of thy directory

- `dir()`
- `getwd()`
- `setwd(...)`
- `file.choose()`

Keep a clean workspace and working directory

- Remove all objects from workspace:

```
rm(list = ls())
```

When quitting R:
- DO NOT save your workspace.
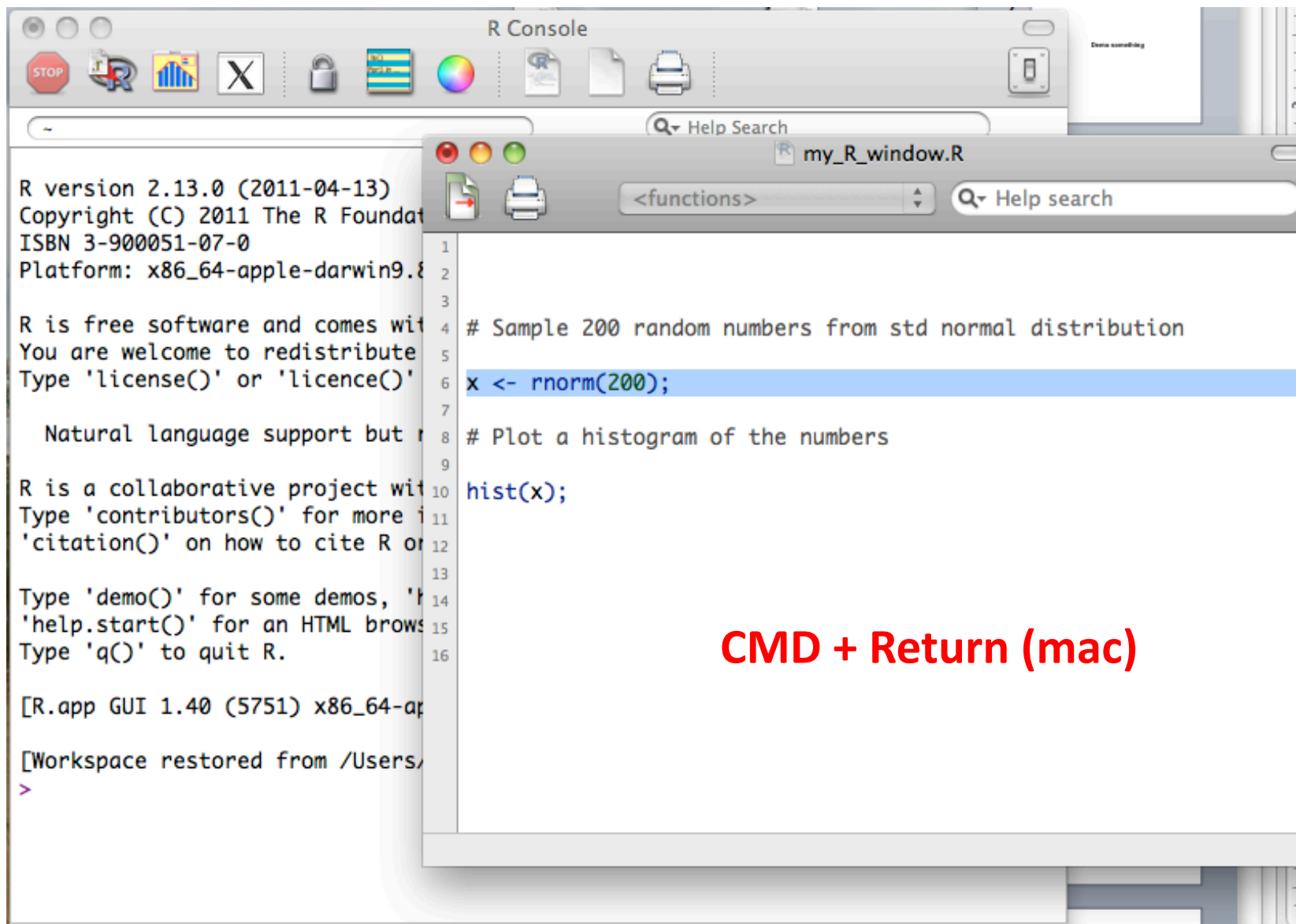- DO save your script files

Keep R updated

- *Failing to update R will lead to many cryptic problems.*

Use the Editor

- *Never, ever execute more than a single line of code at a time in the R console window*

- *Except for "quick looks" at data and results, almost all your work should be done in the Editor*

- *Save everything frequently*

Sourcing and executing code

Getting help

Type *?myFunction* for help on myFunction

- Numerous online resources

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

- R-help
- R-sig-phylo
- R-sig-ecology
- R-sig-mixed-models

Parentheses, brackets, braces

- *These are very different:* `[  ]  ,  {  },  (  )`

- *Brackets [ ] access data from matrices, vectors, and dataframes*
- *Braces { } are strictly used in programming*
- *Parentheses ( ) are required for function calls and to specify "order of operations" for calculations*

Semicolon

- Strongly recommended to end statements with a semicolon;

The "arrow operator" versus "equals sign"

- Assignments in R are made with <-

  ```
  x <- 10;
  ```

- Never use arrow in right-to-left direction, e.g.,

  ```
  10 -> x
  ```

  - equals sign:

    - 1. Function arguments:

    ```
    myData <- read.table(file = 'my_datafile.txt');
    ```

    - 2. Logical comparisons

Commenting your code

```
# Commented code looks like this

# This line will not be executed

# Put lots of comments in your code
```

The end