What is R?

- Statistical computer language similar to S-plus
- Interpreted language (like Matlab)
- Has many built-in (statistical) functions
- Easy to build your own functions
- Good graphic displays
- Extensive help files

Strengths

- Many built-in functions
- Can get other functions from the internet by downloading libraries
- Relatively easy data manipulations

Weaknesses

- Not as commonly used by non-statisticians
- Not a compiled language, language interpreter can be very slow, but allows to call own C/C++ code

R, Statistics and Bio-Statistics

- Packaging: a crucial infrastructure to efficiently produce, load and keep consistent software libraries from (many) different sources / authors
- Statistics: most packages deal with statistics and data analysis
- State of the art: many statistical researchers provide their methods as R packages
- Bioconductor: an open source and open development software project for the analysis and comprehension of genomic data.
http://www.bioconductor.org/

Starting and stopping R

- Starting
  - Windows: **`Double click on the R icon`**
  - Unix/Linux: type **`R (or the appropriate path on your machine)`**
- Stopping
  - **`Type q()`**
  - `q()` is a function execution
  - Everything in R is a function
  - q merely returns the content of the function

Writing R code

- Can input one line of code at a time into R
- Can write many lines of code in any of your favorite text editors and run all at once
  - Simply paste the commands into R
  - Use function source("path/yourscript"), to run in batch mode the codes saved in file "yourscript" (use options(echo=T) to have the commands echoed)

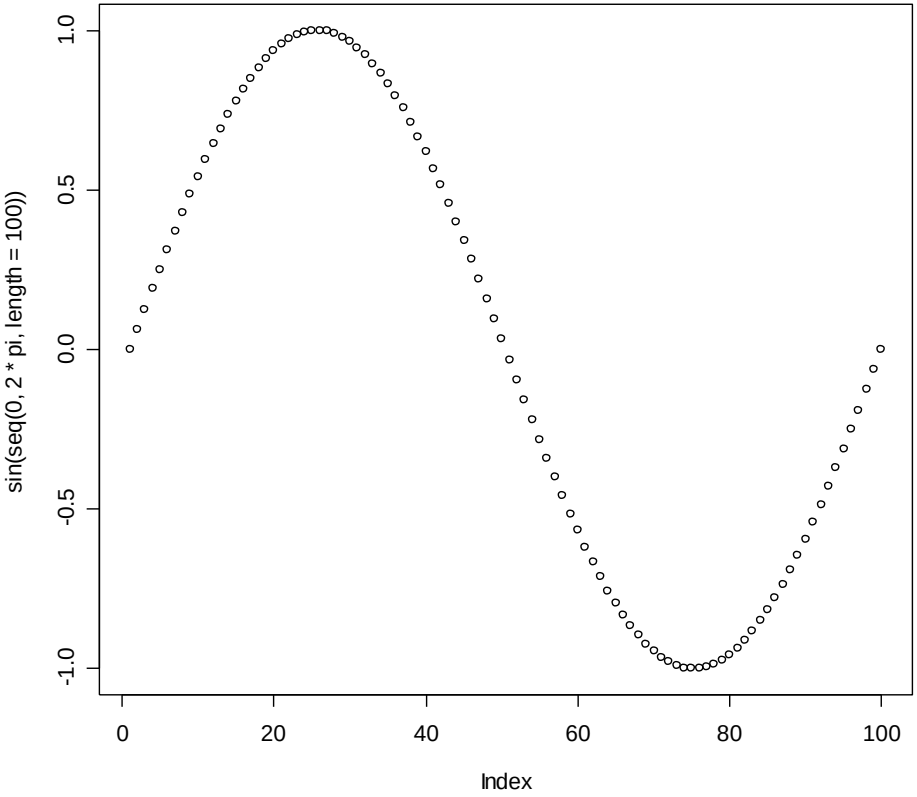
R as a Calculator

```
> log2(32)
[1] 5

> sqrt(2)
[1] 1.414214

> seq(0, 5, length=6)
[1] 0 1 2 3 4 5

> plot(sin(seq(0,
 2*pi, length=100)))
```



Recalling Previous Commands

- In WINDOWS/UNIX one may use the **`arrow up key`** or **`history()`**

- Given the history window, one can copy some commands and then paste them into the console window

Language layout

- Three types of statement
  - expression: it is evaluated, printed, and the value is lost, for example (3+5).
  - assignment: passes the value to a variable but the result is not printed automatically, for example (out<-3+5).
  - comment: (# This is a comment)

Naming conventions

- Use any roman letters, digits, underline, and '.' (non-initial position)
- Avoid using system names: c, q, s, t, C, D, F, I, T, diff, mean, pi, range, rank, tree, var
- Variable names are case sensitive

Arithmetic operations and functions

- Most operations in R are similar to Excel and calculators
- Basic: +(add), -(subtract), *(multiply), /(divide)
- Exponentiation: ^
- Remainder or modulo operator: %%
- Matrix multiplication: %*%
- sin(x), cos(x), cosh(x), tan(x), asin(x), acos(x), acosh(x), atan(x)
- sqrt(x), abs(x), ceiling(x), floor(x), trunc(x)
- exp(x), log(x, base=e), log10(x),
- max(), min()

Defining new variables

- Assignment symbol, use “<-” (or =)
- Scalars

  >scal<-6

  >value<-7
- Vectors; using c() to enter data

  >whales<-c(74,122,235,111,292,111,211,133,16,79)

  >simpsons<-c(“Homer", “Marge", “Bart“, “Lisa”, ”Maggie”)

Use functions on a vector

- Most functions work on vectors exactly as we would want them to do
  - >sum(whales)
  - >length(whales)
  - >mean(whales)
  - sort(), min(), max(), range(), diff()
- Vectorization of (arithmetic) functions
  - >wales - mean(whales)
  - Other arithmetic funs: sin(), cos(), exp(), log(), ^, sqrt()
- Example: The variance VAR(X)
- Finding help
  - >help(funname) or ? funname
  - >help.search("keyword") or ??keyword

Create structured data

- Simple sequences
  - >1:10
  - >rev(1:10)
  - >10:1
- Arithmetic sequence
  - >a=1; h=4; n=5
  - >a+h*(0:(n-1))
- Repeated numbers
  - >rep(1,10)
  - >rep(1:3,3)

>seq(1,9,by=2)
>seq(1,9,length=5)

Matrix

- There are several ways to make a matrix
- To make a 2x3 (2 rows, 3 columns) matrix of 0's:

  >mat<-matrix(0,2,3)
- To make the following matrix:

71	172
73	169
69	160
65	130

>mat2<-rbind(c(71,172),c(73,169),c(69,160),c(65,130))

>mat3<-cbind(c(71,73,69,65),c(172,169,160,130))

- To make the following matrix:
  - mat4<-matrix(1:10,2,5, byrow=T)

1	2	3	4	5
6	7	8	9	10

Accessing data by using index

- Accessing individual observations

  >whales[2]

- Slicing

  >whales[1:4]

- Negative indices

  >whales[-1]

- Logical values

  >whales[whales>100]

  >which(whales>100)

Ways to manipulate a vector/matrix

- x[1]
- x[length(x)]
- x[i]
- x[c(2,3)]
- x[-c(2,3)]
- x[1]=5
- x[c(1,4)]=c(2,3)
- x[indices]=y
- x[x<3]

- mat[,2]
- mat[2,]
- mat[1:3,1]
- mat[c(2,4),]
- mat[-c(2,4),]

Create logical vectors by conditions

- Logical operators: <, <=, >, >=, ==, !=
- Comparisons
  - Vectors: AND &; OR |
- Examples
  - X=1:5
  - X<5; X>1
  - X >1 & X <5; X >1 | X <5;
  - X >1 && X < 5; X >1 || X < 5

Missing values

- R codes missing values as NA
- is.na(x) is a logical function that assigns a T to all values that are NA and F otherwise

```
>x[is.na(x)]<-0
>mean(x, na.rm=TRUE)
```

Manage the work environment

- ls() list all the objects(var, fun, etc) you defined before in a given environment
- Get and set working directory

  >getwd()

  >setwd("working/directory/path")
- Save and load working environment

  >save.image(file="filename.RData")

  >load(file="filename.RData")

Reading in other sources of data

- Use R's built-in libraries and data sets

```
>range(lynx) #lynx is a built-in dataset
>library(MASS) # load a library
>data(survey)  # load a dataset in the library
```

Read formatted data

- Read data from formatted data files, e.g. a file of numbers from a single file, a table of numbers separated by space, comma, tab etc, with or without header

  >whale=scan(file="whale.txt")

  "whale.txt":

  74 122 235 111 292 111 211 133 156 79

  >whale=read.table(file="whale.txt", header=FALSE)

  >read.table(file=url("http://www.math.csi.cuny.edu/Using R/Data/whale.txt"),header=T) # read from internet

Data frame

- read.table() stores data in a *data frame;* it likes a matrix, each column corresponds to a different var, but can be of different types

- Access var in a dataset: $, attach()

```
>library(MASS); data(Sitka)
>names(Sitka)  #variable names
>length(Sitka$tree)
```

```
>attach(Sitka)
>summary(tree)
>detach(Sitka)
```

Additional reference

- Pdf document,
  - http://cran.r-project.org/doc/manuals/R-intro.pdf