

What is R?

R is an integrated statistical environment offering:

- extensive data and statistical analysis tools
- comprehensive linear and non-linear modelling facilities
- graphical facilities for data analysis and display
- a powerful programming language

What are S and S-PLUS?

- S was an ISE developed at Bell Labs (1976)
- S-PLUS is the commercial version of S
- R is a free reimplement of S (1995)

Finding R

- already installed on lab computers
- The Comprehensive R Archive Network (CRAN)
cran.stat.sfu.ca Or cran.r-project.org
Get the appropriate "Precompiled Binary Distribution."

1

Getting Help

- Use `help()`:

```
> help(q)
quit                package:base                R Documentation

Terminate an R Session

Description:

The function 'quit' or its alias 'q' terminate
the current R session.

Usage:

quit(save = "default", status = 0, runLast = TRUE)

[ . . . ]
:
```

At the ":" prompt, press SPACE for more, b to go back, or q to quit.
- Use `help.search()`:

```
> help.search("time series")
avgp(GeneTS)                Average Periodogram for
                             Multiple (Genetic) Time Series

[ . . . ]
```
- Use `help.start()` to launch web browser.
- Or, just visit www.r-project.org, and look at the "Documentation."

3

Using R: Starting and Stopping

Under UNIX or Linux:

```
[yourdir]$ mkdir project1
[yourdir]$ cd project1
[project1]$ R

R : Copyright 2004, The R Foundation for Statistical Computing
Version 2.0.1 (2004-11-15), ISBN 3-900051-07-0

[ . . . ]

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.

> q()
Save workspace image? [y/n/c]: y
[project1]$ ls -a
./          ../          .RData      .Rhistory
[project1]$
```

Using R: A Sample Session

- Follow the "Introduction to R" link on the course webpage.
- Try it out (either on a version installed at home or in MSRC).
- If you make mistakes: arrow keys move through and edit command history.

2

Objects in R

Everything in R (a vector of numbers, a list, a data frame, the result of a regression) is an *object* whose *class* determines how it behaves.

- Numbers (actually, numeric vectors):

```
> x <- 4
> class(x)
[1] "numeric"
> print(x)
[1] 4
> summary(x)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
     4         4         4         4         4         4
> plot(x)
>
```
- Fitted models:

```
> l <- lm(demand ~ Time, data=BOD)
> class(l)
[1] "lm"
> print(l)
[ . . . ]
Coefficients:
(Intercept)          Time
          8.521          1.721
> summary(l)
[ . . . ]
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   8.5214    2.6589   3.205  0.0328 *
Time          1.7214    0.6387   2.695  0.0544 .
[ . . . ]
> plot(l)
> unclass(l)
[ . . . ]
>
```

4

Saving Across Sessions

```
[project1]$ R
[ . . . ]
> x <- round(runif(10))
> x
[1] 0 1 1 0 0 1 1 0 1 0
> l <- lm(demand ~ Time, data=BOD)
> coef(l)
(Intercept)      Time
 8.521429      1.721429
> objects()
[1] "l" "x"
> q()
Save workspace image? [y/n/c]: y
[project1]$

      Many years pass ...

[project1]$ R
[ . . . ]
> objects()
[1] "l" "x"
> x
[1] 0 1 1 0 0 1 1 0 1 0
> coef(l)
(Intercept)      Time
 8.521429      1.721429
> rm(x)
> objects()
[1] "l"
>
```

5

Non-Numeric Vectors

- logical (boolean) vectors

```
> x <- round(10*runif(5))
> x
[1] 1 6 3 3 4
> x > 3
[1] FALSE TRUE FALSE FALSE TRUE
> l <- x > 3
> l
[1] FALSE TRUE FALSE FALSE TRUE
> x[l]
[1] 6 4
> sum(l)
[1] 2
>
> plant.height
[1] 13.41 NA 13.95 11.17 NA
> is.na(plant.height)
[1] FALSE TRUE FALSE FALSE TRUE
> plant.height[is.na(plant.height)] <- 0
> plant.height
[1] 13.41 0.00 13.95 11.17 0.00
>
```

- character vectors

```
> gender <- c("M", "M", "F", "M", "F", "F", "M")
> gender
[1] "M" "M" "F" "M" "F" "F" "M"
> table(gender)
gender
F M
3 4
>
```

7

Vectors

- building with c

```
> x <- c(4.5, -1, 3e-2, sqrt(15))
> x
[1] 4.500000 -1.000000 0.030000 3.872983
> x[4]
[1] 3.872983
>
```

- vector arithmetic

```
> x^2+1
[1] 21.2500 2.0000 1.0009 16.0000
> mean(x)
[1] 1.850746
>
```

- sequences using ":" and seq

```
> 4:10
[1] 4 5 6 7 8 9 10
> 8:2
[1] 8 7 6 5 4 3 2
> 1:5*10
[1] 10 20 30 40 50
> seq(-2, 9, by=2)
[1] -2 0 2 4 6 8
>
```

- (pseudo-)random vectors

```
> rnorm(5, mean=2, sd=.1)
[1] 1.885288 2.107558 2.056270 2.048590 1.955928
> rpois(3, lambda=6)
[1] 3 4 4
>
```

6

Indexing Vectors

- by a vector of positive integers

```
> x
[1] 0 8 9 7 4 2 10 0 2 1
> x[4]
[1] 7
> x[c(2,4,8)]
[1] 8 7 0
>
```

- by a vector of negative integers

```
> x[c(-1,-2,-9)]
[1] 9 7 4 2 10 0 1
>
```

- by a logical vector

```
> x[c(F,F,F,T,F,F,F,T,T,T)]
[1] 7 0 2 1
>
```

- by a vector of character strings

```
> age <- c(10,2,15)
> names(age) <- c("Nancy", "Bill", "Anne")
> age
Nancy Bill Anne
 10 2 15
> age[c("Nancy", "Anne")]
Nancy Anne
 10 15
>
```

8

Factors

- Character vector with allowed set of levels:

```
> gender
[1] "M" "M" "F" "M" "F" "F" "M"
> gender <- factor(gender)
> gender
[1] M M F M F F M
Levels: F M
> gender[2] <- "G"
Warning message: invalid factor level [ . . . ]
> gender
[1] M <NA> F M F F M
Levels: F M
> gender[2] <- "F"
> gender
[1] M F F M F F M
Levels: F M
>
> meal <- factor(c("chicken","beef","beef","chicken"),
                 levels=c("chicken","beef","vegetarian"))
> table(meal)
meal
  chicken    beef vegetarian
      2         2         0
>
```

- Ordered factors, too:

```
> treatment <- ordered(c("placebo","lowdose","lowdose",
                          "placebo","highdose"),
                       levels=c("placebo","lowdose","highdose"))
> treatment
[1] placebo lowdose lowdose placebo highdose
Levels: placebo < lowdose < highdose
>
```