

Introduction to R

Working with data

Arni Magnusson

Hafro, 8 Nov 2010

Outline

1	Input/output	1
2	Data objects	2
3	Special objects	3
4	Manipulation	4

1 Input/output

Read file

```
scan("c:/shop/rivers.txt")
```

```
read.table("c:/shop/cars.txt", header=T)
```

```
read.csv("c:/shop/cars.csv")
```

Write file

```
dir.create("c:/shop/out")
```

```
write(rivers, "c:/shop/out/rivers2.dat")
```

```
write.table(cars, "c:/shop/out/cars2.dat", quote=F,  
            row.names=F)
```

```
write.csv(cars, "c:/shop/out/cars2.csv", quote=F,  
          row.names=F)
```

2 Data objects

Simple objects

```
integer  1      2      3
numeric  0.1    0.2    0.3
character "one"  "two"  "three"
logical  TRUE   FALSE  TRUE
```

Is x integer?

- `is.integer(x)`

Convert x to integer

- `as.integer(x)`

Unusual values

```
0          zero          x == 0
""         empty string  x == ""

NA         not available  is.na(x)
NULL      not defined    is.null(x)

Inf        large positive number  is.infinite(x)
-Inf      large negative number   is.infinite(x)

FALSE     not TRUE       !x
```

Vector and factor

```
numbers <- c(10, 20, 30)
strings <- c("ten", "twenty", "thirty")

vec <- c("West", "Center", "East", "West", "Center")
fac <- factor(vec)
ord <- ordered(vec, levels=c("West", "Center", "East"))

table(fac)
table(ord)

plot(fac)
plot(ord)
```

Matrix, data frame, table

```
matrix(c(10,20,30,40), ncol=2)

# numbers <- c(10, 20, 30)
# strings <- c("ten", "twenty", "thirty")
data.frame(one=numbers, two=strings)

mtcars
class(mtcars)
as.matrix(mtcars)

table(mtcars$cyl)
table(mtcars$am, mtcars$cyl)
```

Matrix, data frame, table

```
matrix all values of same mode (linear algebra)
data.frame data in columns (analysis, plots)
  default choice for statistical analysis
  supports y~x formula notation
  supports x$name column selection
table frequency table (view)
```

3 Special objects

List

```
list(one=rivers, two=TRUE, three=sleep, four=pi)
```

Date/time

```
ISOdate(2010, 11, 08) # 12:00 by default
```

```
ISOdate(2010, 11, 08, 23, 59, 59)
```

```
as.POSIXct("2010-11-08", tz="GMT") # 0:00 by default
```

```
as.POSIXct("2010-11-08 23:59:59", tz="GMT")
```

Formula

```
plot(mpg ~ cyl, data=mtcars)
```

```
lm(mpg ~ cyl, data=mtcars)
```

```
aggregate(mpg ~ cyl, data=mtcars, mean)
```

Other object types

Model results `lm, glm`

Time series data `ts`

Multipanel plot `trellis`

4 Manipulation

Object information

```
length(rivers)                    mode(WorldPhones)
dim(mtcars)                        class(WorldPhones)
nrow(mtcars)                      unclass(mtcars)
ncol(mtcars)                      attributes(mtcars)

names(islands)                    head(mtcars)
dimnames(mtcars)                 tail(mtcars)
rownames(mtcars)                 unique(mtcars$cyl)
colnames(mtcars)                 object.size(mtcars)
```

Names

```
v <- c(10, 20, 30)
names(v) <- c("one", "two", "three")
```

```
head(cars)
```

```
colnames(cars) <- c("s", "d")
```

```
dimnames
```

Logical expressions

`pi == 3`

`pi != 3`

`pi < 3`

`pi <= 3`

`pi > 3`

`pi >= 3`

Logical expressions

AND

`logical && logical`

`vector & vector`

OR

`logical || logical`

`vector | vector`

NOT

`!logical`

`!vector`

Logical expressions

`pi > 3`

`is.character(pi)`

`!is.character(pi)`

`pi > 3 && is.character(pi)`

`pi > 3 || is.character(pi)`

`!(pi > 3 || is.character(pi))`

Logical expressions

```
# numbers <- c(10, 20, 30)
# strings <- c("ten", "twenty", "thirty")
```

```
numbers >= 20
strings == "thirty"
numbers >= 20 | strings == "thirty"
numbers >= 20 & strings == "thirty"
```

```
any(numbers >= 20)
all(numbers >= 20)
```

Logical expressions

```
chickwts$feed=="soybean" | chickwts$feed=="casein"
```

```
chickwts$feed %in% c("soybean","casein")
```

Ways to subset

Vector (logical, integer, names)

```
islands[islands<20]
islands[1:3]
islands[c("Greenland","Iceland","Britain")]
```

Data frame (dollar, logical, integer, names)

```
cars$dist
cars[1,2]
cars[1:10,1]
cars[,1]
```

List (dollar, logical, integer, names)

```
z <- list(one=rivers, two=TRUE, three=sleep, four=pi)
z$two
```

Extract

Vector

```
v <- c(1, 3, 5, 7, 9)
v[1:3]
```

Data frame

```
x <- data.frame(num=v, char=letters[v])
x[1:3, "char"]
```

List

```
z <- list(one=rivers, two=TRUE, three=sleep, four=pi)
z$two
```

Replace

Vector

```
v <- c(1, 3, 5, 7, 9)
v[1:3] <- 0
v <- v[-(1:3)]
```

Data frame

```
x <- data.frame(num=v, char=letters[v])
x[1:3, "char"] <- ""
x <- x[-(1:3),]
```

List

```
z <- list(one=rivers, two=TRUE, three=sleep, four=pi)
z$two <- FALSE
z$two <- NULL
```

Subset summary

```
x[i]          x["name"]          x[c(T,F)]
      select elements from vector
x[i,]         x["name",]         x[c(T,F),] # row
x[,j]         x[, "name"]        x[,c(T,F)] # column
x[i, j]       x["name", "name"]  x[c(T,F), c(T,F)]
      select rows/columns/elements from matrix or data frame
x$name
      select column in data frame, or element in list
```

Subset summary

Combining logical expressions

```
USArrests[USArrests$UrbanPop>80 & USArrests$Rape<20,]
```

Generic

```
rep(10, 3)
rep(1:10, 3)
rep(1:10, each=3)
rep(1:10, length=22)

sample(month.abb, 10, replace=T)

sort(islands)
sort(islands, decreasing=T)

rev(rivers)
order(rivers) # rivers[order(rivers)]
rank(rivers)
```

Numbers

```
1:10
seq(1, 10, 0.5)
seq(1, 10, length=5)

rnorm(10, m=0, s=1)
runif(10, min=0, max=1)
rpois(10, lambda=1)

round(pi)

trunc(pi)

pi %% 1
```

String manipulation

```
nchar(month.name)

paste(month.abb[1], month.abb[3], sep="-")
paste(month.abb, collapse=".")

substring(month.abb, first=2, last=3)

grep("r", month.name)
month.abb[grep("r", month.name)]
month.abb[grep("r", month.name, invert=T)]

gsub("J", "Y", month.abb)
```

Bind, apply, transpose

```
v <- 1:10

cbind(v)
cbind(v^2, log(v))

rbind(v)
rbind(v^2, log(v))

apply(WorldPhones, 1, sum) # within row
apply(mtcars, 2, max)     # within column

t(WorldPhones)
```

Aggregate and tapply

```
aggregate(hp ~ cyl, data=mtcars, mean)
```

```
tapply(mtcars$hp, mtcars$cyl, mean)
```

```
z <- aggregate(qsec ~ cyl+am, data=mtcars, mean)
```

```
xtabs(qsec ~ cyl+am, data=z)
```

```
tapply(mtcars$qsec, list(mtcars$cyl,mtcars$am),  
       mean)
```