The R Book

Chapter 2: Essentials of the R Language
Session 5

Logical arithmetic

```
x <- 3
y <- 5
```

```
x < y
[1] TRUE
```

```
x > y
[1] FALSE
```

```
x == y
[1] FALSE
```

```
x == y
[1] FALSE
```

```
x != y
[1] TRUE
```

```
x < y & x == y
[1] FALSE
```

```
x < y | x == y
[1] TRUE
```

Table 2.3. Logical operations.

Symbol	Meaning
!	logical NOT
&	logical AND
|	logical OR
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
==	logical equals (double =)
!=	not equal
&&	AND with IF
||	OR with IF
xor(x,y)	exclusive OR
isTRUE(x)	an abbreviation of identical(TRUE,x)

Logical Arithmetic

```
(x <- 0:6)
[1] 0 1 2 3 4 5 6

x<4
[1]  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE
```

Logical Arithmetic

Example: generating simplified factor levels

```
(treatment<-letters[1:5])
[1] "a" "b" "c" "d" "e"

(t2<-factor(1+(treatment=="b")+2*(treatment=="c")
+2*(treatment=="d")))
[1] 1 2 3 3 1

Levels: 1 2 3
```

Logical Arithmetic

First of all...

factor: categorical variable in R

```
sex <-
factor(c('male','male','male','female','male','femal
e','female','male','male','female'))
sex
 [1] male   male   male   female male   female
female male   male   female
Levels: female male
```

Logical Arithmetic

```
treatment <- factor(c("a","b","c","d","e"))
treatment
[1] a b c d e
Levels: a b c d e
```

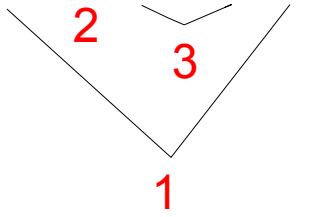
Logical Arithmetic

```
treatment <- factor(c("a","b","c","d","e"))

treatment

[1] a b c d e

Levels: a b c d e
```




```
(t2<-factor(1+(treatment=="b")+2*(treatment=="c")
+2*(treatment=="d")))
[1] 1 2 3 3 1
Levels: 1 2 3
```

```
t2<-factor(1+(treatment=="b")+2*(treatment=="c")
+2*(treatment=="d"))
```

```
treatment
[1] a b c d e
Levels: a b c d e
```

```
treatment == "b"
[1] FALSE  TRUE FALSE FALSE FALSE
```

R coerces this into `0 1 0 0 0`

```
1 + (treatment == "b")
[1] 1 2 1 1 1
```

```
t2<-factor(1+(treatment=="b")+2*(treatment=="c")
+2*(treatment=="d"))
```

```
treatment
[1] a b c d e
Levels: a b c d e
```

```
treatment == "c"
[1] FALSE FALSE  TRUE FALSE FALSE
```

```
2*(treatment=="c")
[1] 0 0 2 0 0
```

1 2 1 1 1

```
1 + (treatment == "b") + 2*(treatment=="c")
[1] 1 2 3 1 1
```

```
t2<-factor(1+(treatment=="b")+2*(treatment=="c")
+2*(treatment=="d"))
```

```
treatment
[1] a b c d e
Levels: a b c d e
```

```
treatment == "d"
[1] FALSE FALSE FALSE  TRUE FALSE
```

```
2*(treatment=="d")
[1] 0 0 0 2 0
```

1 2 3 1 1

```
1 + (treatment == "b") + 2*(treatment=="c") +
2*(treatment=="d")
[1] 1 2 3 3 1
```

```
t2<-factor(1+(treatment=="b")+2*(treatment=="c")
+2*(treatment=="d"))
```

```
treatment
[1] a b c d e
Levels: a b c d e

t2
[1] 1 2 3 3 1
Levels: 1 2 3
```

sample(sample,size,replace=,prob=)

```
y
[1]  8  3  5  7  6  6  8  9  2  3  9  4 10  4 11

sample(y)
[1]  3  8  4  6  6  7  5  2  9 10 11  8  4  9  3

sample(y, replace=T)
[1] 7 8 4 7 7 2 4 3 9 5 8 6 6 3 9
```

sampling with replacement

```
sample(y,5)
[1]  6 10  8  3  8
```

specifying size

```
x <- 1:10
p <- c(1, 2, 3, 4, 5, 5, 4, 3, 2, 1)
sample(x,4,prob = p)
[1] 9 1 4 7
```

specifying the probability of an element to be sampled

Matrices

Some ways of creating a matrix:

1) use `matrix()` to create the matrix from scratch

```
X<-matrix(c(1,0,0,0,1,0,0,0,1),nrow=3)
X
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1
```

2) use `matrix()` to convert from a vector

```
vector<-c(1,2,3,4,4,3,2,1)
X<-matrix(vector,byrow=T,nrow=2)
X
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    4    3    2    1
```

By default, filled up column by column!

Matrices

3) use `dim()` to convert from a vector

```
vector
[1] 1 2 3 4 4 3 2 1


dim(vector)<-c(4,2)


vector
     [,1] [,2]
[1,]    1    4
[2,]    2    3
[3,]    3    2
[4,]    4    1
```

`dim()` has no `byrow=` attribute but you can transpose the result:

```
 t(vector)
     [,1] [,2] [,3]
[,4]
[1,]    1    2    3
4
[2,]    4    3    2
1
```

Matrices

Labelling the rows and columns of a matrix:

Reminder: labelling the elements of a vector

```
 vector
[1] 1 2 3 4 4

 names(vector) <- c("a","b","c","d","e")
 vector
a b c d e
1 2 3 4 4
```

Matrices

Labelling the rows and columns of a matrix:

1) use rownames() or colnames()

```
X<-matrix(rpois(20,1.5),nrow=4)
X
     [,1] [,2] [,3] [,4] [,5]
[1,]    2    1    0    2    3
[2,]    0    1    0    3    0
[3,]    1    0    0    1    1
[4,]    0    3    3    1    4

drug.names<-c("aspirin", "paracetamol", "nurofen", "hedex", "placebo")
colnames(X)<-drug.names
X
     aspirin paracetamol nurofen hedex placebo
[1,]       1           1       1     2       0
[2,]       1           1       0     3       0
[3,]       0           0       1     0       0
[4,]       1           1       3     0       3
```

Matrices

Labelling the rows and columns of a matrix:

2) use `rownames()` or `colnames()` and specify the `prefix=` attribute

```
rownames(X)<-rownames(X,do.NULL=FALSE,prefix="Trial.")
X
        aspirin paracetamol nurofen hedex placebo
Trial.1       1           1       1     2       0
Trial.2       1           1       0     3       0
Trial.3       0           0       1     0       0
Trial.4       1           1       3     0       3
```

Matrices

Labelling the rows and columns of a matrix:

3) use `dimnames()`

```
 drug.names
[1] "aspirin"     "paracetamol" "nurofen"     "hedex"
  "placebo"


 dimnames(X) = list(NULL, drug.names)
 X
     aspirin paracetamol nurofen hedex placebo
[1,]       1           2       1     3       3
[2,]       3           0       2     1       3
[3,]       3           3       2     2       1
[4,]       2           2       1     2       3
```

Matrices

Calculations on the rows and columns of a matrix:

```
 X
        aspirin paracetamol nurofen hedex placebo
Trial.1       1           2       1     3       3
Trial.2       3           0       2     1       3
Trial.3       3           3       2     2       1
Trial.4       2           2       1     2       3
```

```
 mean(X[,5])
[1] 2.5

 var(X[4,])
[1] 0.5

 rowSums(X)
Trial.1 Trial.2 Trial.3 Trial.4
     10       9      11      10

colSums(X)
    aspirin paracetamol     nurofen       hedex     placebo
          9           7           6           8          10
```

Matrices

Calculations on the rows and columns of a matrix:

```
 X
        aspirin paracetamol nurofen hedex placebo
Trial.1       1           2       1     3       3
Trial.2       3           0       2     1       3
Trial.3       3           3       2     2       1
Trial.4       2           2       1     2       3
```

```
 rowMeans(X)
Trial.1 Trial.2 Trial.3 Trial.4
    2.0     1.8     2.2     2.0

 colMeans(X)
    aspirin paracetamol     nurofen       hedex     placebo
       2.25        1.75        1.50        2.00        2.50

 apply(X,2,mean)
    aspirin paracetamol     nurofen       hedex     placebo
       2.25        1.75        1.50        2.00        2.50
```

Matrices

Exercice:

use `apply()` to independently shuffle each column of `X`!

```
 X
        aspirin paracetamol nurofen hedex placebo
Trial.1       1           2       1     3       3
Trial.2       3           0       2     1       3
Trial.3       3           3       2     2       1
Trial.4       2           2       1     2       3
```

```
 X.shuffled
     aspirin paracetamol nurofen hedex placebo
[1,]       1           3       1     1       3
[2,]       3           2       2     3       1
[3,]       2           0       2     2       3
[4,]       3           2       1     2       3
```

Matrices

Solution:

```
 X.shuffled <- apply(X,2,sample)
 X.shuffled
     aspirin paracetamol nurofen hedex placebo
[1,]       1           3       1     1       3
[2,]       3           2       2     3       1
[3,]       2           0       2     2       3
[4,]       3           2       1     2       3
```

Matrices

Grouping rows together using `rowsum()`

```
 X
        aspirin paracetamol nurofen hedex placebo
Trial.1       1           2       1     3       3
Trial.2       3           0       2     1       3
Trial.3       3           3       2     2       1
Trial.4       2           2       1     2       3
```

```
 group <- c("A","B","B","A")

 rowsum(X,group)
  aspirin paracetamol nurofen hedex placebo
A       3           4       2     5       6
B       6           3       4     3       4
```

Matrices

Adding rows and columns to a matrix:

```
 X
         aspirin paracetamol nurofen hedex placebo
Trial.1        1           2       1     3       3
Trial.2        3           0       2     1       3
Trial.3        3           3       2     2       1
Trial.4        2           2       1     2       3
```

```
 X <- rbind(X,apply(X,2,mean))
 X <- cbind(X,apply(X,1,var))
 X
         aspirin paracetamol nurofen hedex placebo
Trial.1    1.00        2.00     1.0     3     3.0 1.00000
Trial.2    3.00        0.00     2.0     1     3.0 1.70000
Trial.3    3.00        3.00     2.0     2     1.0 0.70000
Trial.4    2.00        2.00     1.0     2     3.0 0.50000
           2.25        1.75     1.5     2     2.5 0.15625
```