4. Matrix

A matrix is a two-dimensional array.

Create a matrix from vector data with `matrix(data=NA, nrow=1, ncol=1, byrow=FALSE, dimnames=NULL)`, where `byrow` tells whether to fill the matrix from `data` by row (or by column, the default), and `dimnames` is `NULL` or a list of two vectors containing row and column names. e.g.

```r
m = matrix(data=1:12, nrow=3, ncol=4, byrow=TRUE)
kids = matrix(data=c(c(1,2,6,7,9,11), c(1,5,100,100,100,100)),
               nrow=2, ncol=6, byrow=TRUE,
               dimnames=list(c("Age","#Toys"),
                             c("Teresa","Margaret","Monica","Andrew","Mary","Philip"))
)
```

Two other ways to create matrices are by combining columns with `cbind(...)` or rows with `rbind(...)`, getting data from vector, matrix, or data frame arguments in . . . :

- `cbind(...)` combines columns into a matrix; e.g. `cbind(m, 101:103)`
- `rbind(...)` combines rows into a matrix; e.g. `rbind(m, 101:104)`

For a matrix `m` (this paragraph helps with Connect Four),

- `row(m)` is a matrix of row numbers of elements of `m` (depends on `m`’s dimensions, not data)
- `col(m)` is a matrix of column numbers of elements of `m`; e.g.

```r
row(m) == col(m) # main diagonal
r = 2; c = 3
m[row(m) - col(m) == r - c] # diagonal through (r, c)
```

For matrices `A` and `B` and vectors `b` and `x`,

- `A * B` is an element-wise product
- `A %*% B` is the usual matrix product, and `A %*% x` is the usual matrix-vector product
- `solve(a=A, b=b)` gives the solution `x` to the system of linear equations, `A x = b`; e.g.

```r
A = matrix(data=1:4, nrow=2, ncol=2)
b = c(7, 10)
(x = solve(a=A, b=b))
A %*% x # check: is it b?
```

The `Matrix` package has more: 
[http://cran.r-project.org/web/packages/Matrix/Matrix.pdf](http://cran.r-project.org/web/packages/Matrix/Matrix.pdf).