Part A Simulation and Statistical programming HT15

Geoff Nicholls

Lecture 12: reference slides for matrices in R

R commands for matrices and vectors

Here are some slides of R commands for matrices and vectors. Please refer back to them in the practical as needed.

Vectors and Matrices in R

Matrices can be constructed using the functions `matrix()`, `cbind()` or `rbind()`.

```
matrix(data, nrow, ncol)
# data is a vector of nrow*ncol values
```

```
cbind(d1, d2, ...., dm)
# d1, ....  , dm are vectors (columns)
```

```
rbind(r1, r2, ....,rn)
# r1, ...  , rn are vectors (rows)
```

Accessing elements

If `X` is a matrix we can access the element in the ith row and jth column using `X[i,j]`

We can access the ith row using `X[i,]`, and the jth column using `X[,j]`.

These commands result in an answer that is a vector with no dimension information kept. If we want to maintain the result as a row or column vector we use the option `drop = FALSE`
i.e. `X[i, ,drop = FALSE]` and `X[,j,drop = FALSE]`

If we want to find which elements satisfy a certain property we can use the `which()` command.

```
which(X >= 0, arr.ind = TRUE)
```

Matrix properties

There are a few useful functions that return basic properties of matrices

`dim()`	returns the number of rows and columns
`det()`	returns the determinant of a square matrix
`diag()`	returns the diagonal entries of a matrix
	OR turns a vectors into a diagonal matrix. `sum(diag())` can be used to calculate the trace.
`t()`	returns the transpose of a matrix
`upper.tri()`	returns a matrix of logical elements with `TRUE` for the upper triangular elements.
`lower.tri()`	returns a matrix of logical elements with `TRUE` for the lower triangular elements.
`eigen()`	Computes eigenvalues and eigenvectors of real or complex matrices.

Matrix arithmetic

`X + Y`	element-wise addition (matrices must conform)
`X + 2`	addition of 2 to each element of `X`
`X * Y`	element-wise multiplication (matrices must conform)
`X * 2`	multiplication of each element of `X` by 2
`X %*% Y`	matrix multiplication (matrices must conform)
`crossprod(Y,X)`	calculates $Y^T X$ efficiently
`crossprod(Y)`	calculates $Y^T Y$
`solve(X)`	returns the inverse of a square matrix.
`solve(X,b)`	Solves a system of linear equations $X\theta = b$
`backsolve(A,b)`	Solves a system of linear equations $A\theta = b$ where the coefficient matrix A is upper or lower triangular.