UQ Winter School - The Basics, session 1

Kathryn Kemper

Objective

This practical session will review the definitions and matrix algebra operations outlined in the handout `1_matrixAlgebra`, and be a basic introduction to using R.

Important skills in R include writing loops, defining matrices and matrix manipulations; and making basic plots of data.

Matrix Algebra

Defining matrices in R is relatively straight-forward. Just need to remember it fills rows then across columns when inputting the data, and to define either the number of rows or columns. We will input each of the matrices in the worksheet.

```
A = matrix(c(1,3,2,1,3,-1,-2,0,1),ncol=3) ; A
```

```
##      [,1] [,2] [,3]
## [1,]    1    1   -2
## [2,]    3    3    0
## [3,]    2   -1    1
```

```
B = matrix(c(1,0,1,0,1,1),nrow=2) ; B
```

```
##      [,1] [,2] [,3]
## [1,]    1    1    1
## [2,]    0    0    1
```

```
C = diag(2) ; C
```

```
##      [,1] [,2]
## [1,]    1    0
## [2,]    0    1
```

```
D = matrix(c(2,0,1,1,4,2),ncol=2) ; D
```

```
##      [,1] [,2]
## [1,]    2    1
## [2,]    0    4
## [3,]    1    2
```

```
E = t(D) ; E
```

```
##      [,1] [,2] [,3]
## [1,]    2    0    1
## [2,]    1    4    2
```

```r
F = matrix(c(3,1,0,2),nrow=2) ; F
```

```
##      [,1] [,2]
## [1,]    3    0
## [2,]    1    2
```

```r
G = matrix(c(1,3),nrow=2) ; G
```

```
##      [,1]
## [1,]    1
## [2,]    3
```

```r
H = diag(3) ; H
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    1    0
## [3,]    0    0    1
```

```r
I = matrix(c(1,3,0,2,6,0),nrow=3) ; I
```

```
##      [,1] [,2]
## [1,]    1    2
## [2,]    3    6
## [3,]    0    0
```

Note above how to transpose a matrix, `t()`, and the short-hand for defining an identity matrix. Note that the `diag()` function in R can have two uses, depending on the argument provided to the function. By providing a single number, as above, the function returns an identity matrix with the given dimension. If the argument is a square matrix, it will return the diagonal elements of the matrix.

Next we need the dimensions of the matrices. The dimensions can be found in R using the `dim()` function.

```r
  #matrix dimensions
   dim(A) ; dim(B) ; dim(C)
```

```
## [1] 3 3

## [1] 2 3

## [1] 2 2
```

```r
   dim(D) ; dim(E) ; dim(F)
```

```
## [1] 3 2

## [1] 2 3

## [1] 2 2
```

```r
   dim(G) ; dim(H) ; dim(I)
```

```
## [1] 2 1

## [1] 3 3

## [1] 3 2
```

There are a few special matrices in the examples, including 2 identity matrices (C, H) and 2 square matrices (A, F).

The rank of a matrix is the number of independent rows. If a matrix is not 'full rank' (meaning that the rank of the matrix is less than the largest dimension of the matrix) then it does not have a unique inverse. Can you identify the matrix that is not full rank? Calculating the inverse of a matrix is important when solving systems of equations, e.g. when estimating regression slopes. The rank of a matrix can be found using `qr()$rank` in R.

If the matrix is not full rank alternative methods to calculate the inverse (i.e. a generalized inverse) need to be used.

```
# the matrix that is not full rank is matrix 'I', can you see why?
 qr(I)$rank
```

```
## [1] 1
```

Next check the answers for the matrix multiplications, note that matrix multiplication has a special symbol, `%*%`.

```
2*B
```

```
##      [,1] [,2] [,3]
## [1,]    2    2    2
## [2,]    0    0    2
```

```
# B%*%C, non-conformable
B%*%D
```

```
##      [,1] [,2]
## [1,]    3    7
## [2,]    1    2
```

```
# C%*%D, non-conformable
B-E
```

```
##      [,1] [,2] [,3]
## [1,]   -1    1    0
## [2,]   -1   -4   -1
```

```
t(C)
```

```
##      [,1] [,2]
## [1,]    1    0
## [2,]    0    1
```

```
A%*%H
```

```
##      [,1] [,2] [,3]
## [1,]    1    1   -2
## [2,]    3    3    0
## [3,]    2   -1    1
```

Which matrices are invertible? Inverses are only calculable for square matrices, therefore only A and F potentially have inverses. We can use `solve()` in R to calculate the inverse and show its properties.

```
#calculate the inverse
 Ainv=solve(A)
 Finv=solve(F)
#show that a matrix multiplied by its inverse gives an identity matrix.
 A%*%Ainv
```

```
##      [,1]         [,2] [,3]
## [1,]    1 0.000000e+00    0
## [2,]    0 1.000000e+00    0
## [3,]    0 8.326673e-17    1
```

```r
Finv%*%F
```

```
##      [,1] [,2]
## [1,]    1    0
## [2,]    0    1
```

The final useful function to know for matrix manipulate in R is using `cbind()` and `rbind()` to combine matrices. Try out these functions to join some of your matrices together.