M7 Matrices and their properties

Miranda Mortlock and Ting Tian

2 November 2016

Looking at R and matrix formation, properties and calculations

These are important concepts for working with data

They are useful in designing and analysing the results

They are relevant as types of matrices such as Design, Correlation, Covariance and for constructing contrasts

Example: Making a matrix from scratch using vectors

We will make a matrix with the following data

40 34 56 78
30 45 78 23
67 45 23 40
12 56 78 68

Reading data in as vector

Start at top left and put each number consecutively ###### into one vector. The c() can be interpreted as combine ###### or concantonate values within the brackets

```
cells <- c(40, 34, 56, 78, 30, 45, 78, 23, 67, 45, 23, 40, 12, 56, 78, 68)
rnames <- c("R1", "R2", "R3", "R4")
cnames <- c("C1", "C2", "C3", "C4")
```

Making a matrix from a vector

The size is given as number of rows (nrow)

and number of columns (ncol)

```
Mymatrix4 <- matrix(cells, nrow=4, ncol=4, byrow=TRUE,
                    dimnames=list(rnames, cnames))
Mymatrix4
```

```
##    C1 C2 C3 C4
## R1 40 34 56 78
## R2 30 45 78 23
## R3 67 45 23 40
## R4 12 56 78 68
```

Checking on the structure of your matrix, confirming it is a matrix

```r
str(Mymatrix4)
```

```
##  num [1:4, 1:4] 40 30 67 12 34 45 45 56 56 78 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:4] "R1" "R2" "R3" "R4"
##   ..$ : chr [1:4] "C1" "C2" "C3" "C4"
```

```r
class(Mymatrix4)
```

```
## [1] "matrix"
```

Making a matrix directly in R

Useful properties of matrices and their relevance to statistics

Transpose a matrix

```r
TrMymatirx4<-t(Mymatrix4)
TrMymatirx4
```

```
##    R1 R2 R3 R4
## C1 40 30 67 12
## C2 34 45 45 56
## C3 56 78 23 78
## C4 78 23 40 68
```

A Diagonal matrix

```r
diag.matrix<-Mymatrix4
diag.matrix[!diag.matrix==diag(diag.matrix)]<-0
diag.matrix
```

```
##    C1 C2 C3 C4
## R1 40  0  0  0
## R2  0 45  0  0
## R3  0  0 23  0
## R4  0  0  0 68
```

Triangular matrices

```r
uptri<-Mymatrix4
lowtri<-Mymatrix4
uptri[!upper.tri(Mymatrix4,diag=TRUE)]<-0
lowtri[!lower.tri(Mymatrix4,diag=TRUE)]<-0
uptri
```

```
##    C1 C2 C3 C4
## R1 40 34 56 78
## R2  0 45 78 23
## R3  0  0 23 40
## R4  0  0  0 68
```

```
lowtri
```

```
##    C1 C2 C3 C4
## R1 40  0  0  0
## R2 30 45  0  0
## R3 67 45 23  0
## R4 12 56 78 68
```

The Identity matrix

```
Imatrix<-matrix(1,4,4,byrow=TRUE,
                   dimnames=list(rnames, cnames))
Imatrix[upper.tri(Imatrix)]<-0
Imatrix[lower.tri(Imatrix)]<-0
Imatrix
```

```
##    C1 C2 C3 C4
## R1  1  0  0  0
## R2  0  1  0  0
## R3  0  0  1  0
## R4  0  0  0  1
```

Create an Identity matrix using a vector

```
elements<-c(1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1)
Imatrix1<-matrix(elements,4,4,byrow=TRUE,
                   dimnames=list(rnames, cnames))
Imatrix1
```

```
##    C1 C2 C3 C4
## R1  1  0  0  0
## R2  0  1  0  0
## R3  0  0  1  0
## R4  0  0  0  1
```

Create an Identity matrix using diag command

```
I<-diag(1,4,4)
I
```

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

Inverting a matrix

```
invmatrix<-solve(Mymatrix4)
invmatrix
```

```
##             R1          R2           R3           R4
## C1  0.01198842  0.01077589  0.006772325 -0.021379925
## C2 -0.03627424 -0.01382992  0.021861927  0.033426493
## C3  0.01228786  0.02093958 -0.014440375 -0.012683065
## C4  0.01366240 -0.01453122 -0.002635096  0.005499332
```

To check the result use invmatrix,

```
I<-invmatrix%*%Mymatrix4
I
```

```
##               C1           C2            C3            C4
## C1  1.000000e+00 0.000000e+00  0.000000e+00  0.000000e+00
## C2 -3.330669e-16 1.000000e+00 -4.440892e-16 -4.440892e-16
## C3  0.000000e+00 0.000000e+00  1.000000e+00  2.220446e-16
## C4  5.551115e-17 1.110223e-16 -5.551115e-17  1.000000e+00
```

Product of 2 matrices "%*% "means product of matrices"

```
I1<-Mymatrix4%*%invmatrix
I1
```

```
##              R1           R2            R3            R4
## R1 1.000000e+00 2.220446e-16  0.000000e+00 -2.220446e-16
## R2 1.665335e-16 1.000000e+00 -2.706169e-16 -1.665335e-16
## R3 0.000000e+00 0.000000e+00  1.000000e+00 -1.110223e-16
## R4 2.220446e-16 0.000000e+00  5.551115e-17  1.000000e+00
```

A determinant, makes a scalar value

Consider a 2 by 2 matrix with elements:

a b c d

say the values are

1 2 3 4

Then the Determinant is (a x d) - (b x 3) = -6

```r
detA<-det(Mymatrix4)
detA
```

```
## [1] -6912840
```

```r
detB<-det(invmatrix)
detB
```

```
## [1] -1.446583e-07
```

```r
a<-det(Imatrix)
a
```

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

```r
b<-detA*detB
b
```

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

A note: if the determinant is not equal to zero it is ###### nonsingular. In this case the inverse of the maxtrix ###### exists. So if A is non singular then A^{-1} exists such that $AA^{-1} = A^{-1}A = \mathbf{I}$.

Eigenvalues & Eigenvectors

```r
####to create a 4 by 4 matrix A
cell<-c(2,6,8,4,3,8,10,7,5,4,1,2,7,9,3,5)
A<-matrix(cell,4,4)
A
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    2    3    5    7
## [2,]    6    8    4    9
## [3,]    8   10    1    3
## [4,]    4    7    2    5
```

Use function eigen to obtain eigenvalues and eigenvectors

```r
eigval<-eigen(A)$values
eigvec<-eigen(A)$vectors
eigval
```

```
## [1] 21.276850 -5.329183  1.229098 -1.176764
```

```r
eigvec
```

```
##            [,1]        [,2]       [,3]       [,4]
## [1,] -0.3888814 -0.62164648 -0.4762871  0.6224778
## [2,] -0.6243249 -0.02586646  0.2254270 -0.4720895
## [3,] -0.5247323  0.77536702 -0.7118611 -0.5430872
## [4,] -0.4285390  0.10813218  0.4643136  0.3077485
```

Where eigvec is a 4 by 4 matrix whose columns contain

the eigenvectors of matrix A,

to test if det(A-lamda*Identity matrix)=0

```
lamda1<-eigval[1]
lamda2<-eigval[2]
lamda3<-eigval[3]
lamda4<-eigval[4]
```

eigval is a vector including 4 eigvalues

I is a 4 by 4 identitiy matrix

```
det(A-lamda1*I)
```

```
## [1] -4.27633e-10
```

```
det(A-lamda2*I)
```

```
## [1] -4.603921e-12
```

```
det(A-lamda3*I)
```

```
## [1] -8.638045e-14
```

```
det(A-lamda4*I)
```

```
## [1] -9.758992e-14
```

```
eigv1<-eigvec[,1]
```

The first eigenvector

```
A%*%eigv1
```

```
##             [,1]
## [1,]  -8.274172
## [2,] -13.283668
## [3,] -11.164650
## [4,]  -9.117960
```

```
lamda1*eigv1
```

```
## [1]  -8.274172 -13.283668 -11.164650  -9.117960
```

Covariance matrix and Correlation marix

```r
covMx<-cov(A)
covMx
```

```
##           [,1]      [,2]      [,3]      [,4]
## [1,]  6.666667  7.333333 -3.333333 -2.666667
## [2,]  7.333333  8.666667 -4.333333 -3.333333
## [3,] -3.333333 -4.333333  3.333333  4.000000
## [4,] -2.666667 -3.333333  4.000000  6.666667
```

Where diagonal values are the variance of each column vector, and off-diagonal values are the covariance between pairs of column vectors

```r
col1<-A[,1]
col2<-A[,2]
col3<-A[,3]
col4<-A[,4]
covMx11<-var(col1)
covMx11
```

```
## [1] 6.666667
```

Variances and covariance

```r
covMx22<-var(col2)
covMx22
```

```
## [1] 8.666667
```

```r
covMx33<-var(col3)
covMx33
```

```
## [1] 3.333333
```

```r
covMx44<-var(col4)
covMx44
```

```
## [1] 6.666667
```

```r
covMx21<-cov(col1,col2)
covMx21
```

```
## [1] 7.333333
```

Correlation matrix

```r
corMx<-cor(A)
corMx
```

```
##            [,1]       [,2]       [,3]       [,4]
## [1,]  1.0000000  0.9647638 -0.7071068 -0.4000000
## [2,]  0.9647638  1.0000000 -0.8062258 -0.4385290
## [3,] -0.7071068 -0.8062258  1.0000000  0.8485281
## [4,] -0.4000000 -0.4385290  0.8485281  1.0000000
```

Where off-diagonal entries are the correlation between pairs of column vectors

Design matrices

A contrast matrix

Some contrasts are available in package car

making contrasts is possible to customise

an orthoganol contrast