Econ 673 Lecture 2, 2017-01-05

January 5, 2017 2:37 PM

~matrix prints position for row/column, vector doesn't.

Note, if you subset a vector from matrix, R automatically drops attribute dim().

e.g. X[1, c(1,2)] → vector.

if you want to keep as 1x2 matrix.

X[1, c(1,2), drop = FALSE] → matrix.

eg. X[X[,1] < 0,]

select row, produce corresponding columns.

e.g. X[c(1,2), c(1,2)][2,2]

order operation.

eg. matrix(1:24, 6, 4)[1:2, 1:2].

Operations: *, -, +, /, ^

element operation.

$$A_{2\times2} * B_{22} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{11}b_{21} & a_{22}b_{22} \end{bmatrix}$$

$$A_{2\times2} + \underset{\sim}{C} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} + \begin{bmatrix} c_1 & c_1 \\ c_2 & c_2 \end{bmatrix}.$$

$\underset{\sim}{C}$: 2 element vector.

if C is 3 element vector. $\Rightarrow \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} + \begin{bmatrix} c_1 & c_3 \\ c_2 & c_1 \end{bmatrix}$.

ex. $R_{T\times N}$ stock returns
3000 x 1000

$R_F \rightarrow$ 3000 vectors of risk-free rates.

Non-R way

$R - (R_f - C)$.
$T\times1$ $1\times N$.

R-way

$R - R_f$.

Matrix Operations

$A_{50\times5}\ B_{5\times10} = C_{50\times10}$.

$A \,\%*\%\, B$

multiplication of matrix.

Matrix Inversion

solve(A) = A^{-1}

Matrix Transpose

t(A) → transpose (A').

ex. $X_{100 \times 5}$ $\quad Y = X\beta + u$.

$\hat{\beta} = (X'X)^{-1}X'y$.

$X'X \leftarrow$ t(X) %*% X.

$X'y \leftarrow$ t(X) %*% y.

$\hat{\beta} = (X'X)^{-1}X'y \leftarrow$ solve(XX) %*% Xy.

Faster Way: XX ← crossprod(X) ← upper triangular matrix.

Xy ← crossprod(X, y)

3rd type of object: List. → collection of elements of possible diff types.

is(mylist) ⇒ "list", "vector".

Accessing the elements

1) mylist[[1]]

2) mylist$name.

when unique, partial name works too.

3) Subsetting: mylist[c(1,4)].

If you don't know names: use names(mylist).

names() works on all objects.

↳ everything is a list!

this is why res$coef works!

Data.Frame:

≈ matrix Data $_{n \times k}$ → the k variables could be different.

ex.

Income	Province	Conservative
"numerical"	"characters"	"logical"

If order is the same, Data$income is the same as Data[,1].

4th type of object : Functions.

Loops: X100x1

S ← 0

① for (i in 1:100).

{ S ← S + X[i] }.

② i = 1

while (i ≤ 100) {

S ← S + X[i]

i ← i + 1 }.

③ i = 1

while (TRUE)

{ S ← S + X[i]

i ← i + 1.

if (i > 100)

break }.

function:

use source file.

→write all functions in a single file. (procedure. R).

ex.1
```
mysum ← function(x)
  { n ← length(x)
    for (i in 1:n){
      S ← S + x[i]}.
    return(s).
    }..
```

ex.2
```
myplot ← function(f, a, b, np=100, ...).
  {x ← seq(a, b, length.out = np).
   y ← f(x, ...).   ~ pass argument.
   plot(x, y, main="my plot", xlab="x", ylab="y", type="l")
    {.
```

```
myplot(dchisq, 1, 10, df=4)
        ↓
   y=dchisq(x, df=4).
```