001 example

Ananda Biswas

Getting estimates by lm()

```
X <- matrix(data = c(3, 4, 2, 4, 1, 3), nrow = 3, ncol = 2, byrow = FALSE)

X

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

```
y <- c(9.8, 9.1, 7.0)

y

## [1] 9.8 9.1 7.0
```

```
lm(y ~ X-1)

##
## Call:
## lm(formula = y ~ X - 1)
##
## Coefficients:
##     X1      X2
## 2.0378  0.9411
```

- see video 5 to know why X-1

```
model.matrix(lm(y ~ X-1))

##   X1 X2
## 1  3  4
## 2  4  1
## 3  2  3
## attr(,"assign")
## [1] 1 1
```

Getting estimates by solving normal equations

```
M <- t(X) %*% X

M
```

```
##      [,1] [,2]
## [1,]   29   22
## [2,]   22   26
```

```
N <- t(X) %*% y

N
```

```
##      [,1]
## [1,] 79.8
## [2,] 69.3
```

```
solve(M, N)
```

```
##           [,1]
## [1,] 2.0377778
## [2,] 0.9411111
```

Compare the estimates by these two methods. See that they are almost same.

lm() function is more stable than the other method.

Let us have a case where the matrix X is singular.

```
X = matrix(data = rep(c(3, 4, 2), 2), nrow = 3, ncol = 2, byrow = FALSE)

X
```

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

```
M <- t(X) %*% X

M
```

```
##      [,1] [,2]
## [1,]   29   29
## [2,]   29   29
```

```r
N <- t(X) %*% y

N
```

```
##       [,1]
## [1,] 79.8
## [2,] 79.8
```

```r
solve(M, N)
```

```
## Error in solve.default(M, N): Lapack routine dgesv:  system is exactly singular:
U[2,2] = 0
```

- This will throw an error as X is singular.

```r
lm(y ~ X-1)
```

```
##
## Call:
## lm(formula = y ~ X - 1)
##
## Coefficients:
##    X1     X2
## 2.752     NA
```

But lm() function does not throw error and gives estimates.

NA implies the system has infinitely many solutions.

lm() reports one particular solution.

lm() has solved coefficient of X1 i.e. the first column of X by arbitrarily setting coefficient of X2 to 0.

So lm() is preferable.