Algebra of OLS

Zhentao Shi

September 23, 2016

We demonstrate the OLS estimator and its algebraic properties

Generating data

```
n = 20 # sample size
K = 4  # number of paramters

b0 = as.matrix( c(0.5, 1, -1, 1) ) # the true coefficient

X = cbind(1, matrix( rnorm(n * (K-1)), nrow = n ) )  # the regressor matrix
e = rnorm(n) # the error term

Y = X %*% b0 + e # generate the dependent variable
```

After the data generation, we obtain an $n \times 1$ vector of Y and an $n \times K$ vector of X. Since the random generator seed is unspecified, the generated random variables are different every time we run the code.

OLS estimator

```
bhat = solve(t(X)%*%X, t(X) %*% Y ) # translate the formula into code
```

Calculate the estimate as $\hat{\beta} = (X'X)^{-1}X'Y =$ (0.6570774, 0.995053, -1.0213077, 0.8804432).

Residual

The residual $\hat{e} = Y - X'\hat{\beta}$. Verify $X'\hat{e} = 0$.

```
ehat = Y - X %*% bhat
print( t(X) %*% ehat )
```

```
##               [,1]
## [1,]  3.996803e-15
## [2,] -2.071260e-15
## [3,] -2.220446e-15
## [4,]  5.773160e-15
```

Notice that

- $\sum_{i=1}^{n} e_i = 3.9332269$,
- $\sum_{i=1}^{n} \hat{e}_i =$ 3.9968e-15.

Define P_X and M_X, and show $\hat{e} = M_X Y = M_X e$.

```
PX = X %*% solve( t(X) %*% X) %*% t(X)
MX = diag(rep(1,n)) - PX
print( cbind( ehat, MX %*% Y, MX %*% e) )
```

```
##              [,1]        [,2]        [,3]
##  [1,]  0.33177812  0.33177812  0.33177812
##  [2,]  0.21233754  0.21233754  0.21233754
##  [3,] -1.46626529 -1.46626529 -1.46626529
##  [4,]  1.42512627  1.42512627  1.42512627
##  [5,] -1.74135996 -1.74135996 -1.74135996
##  [6,] -0.06935491 -0.06935491 -0.06935491
##  [7,]  0.26358389  0.26358389  0.26358389
##  [8,]  0.53747283  0.53747283  0.53747283
##  [9,]  0.19033731  0.19033731  0.19033731
## [10,]  1.63498225  1.63498225  1.63498225
## [11,]  0.48002238  0.48002238  0.48002238
## [12,]  1.98243727  1.98243727  1.98243727
## [13,] -0.90346464 -0.90346464 -0.90346464
## [14,]  0.31947513  0.31947513  0.31947513
## [15,] -1.59282064 -1.59282064 -1.59282064
## [16,] -0.85626692 -0.85626692 -0.85626692
## [17,]  1.04236644  1.04236644  1.04236644
## [18,]  0.52075031  0.52075031  0.52075031
## [19,] -0.55995540 -0.55995540 -0.55995540
## [20,] -1.75118198 -1.75118198 -1.75118198
```

FWL Theorem

```
X1 = X[,1:2]
PX1 = X1 %*% solve( t(X1) %*% X1) %*% t(X1)
MX1 = diag(rep(1,n)) - PX1
X2 = X[,3:4]


bhat12 =  solve(t(X2)%*% MX1 %*% X2, t(X2) %*% MX1 %*% Y )
```

$(\hat{\beta}_3, \hat{\beta}_4)$ = (-1.0213077, 0.8804432), which is the same as the counterpart in $\hat{\beta}$ = (0.6570774, 0.995053, -1.0213077, 0.8804432).

```
# the residuls after purging out X1 is the same as that from the full regression
ehat12 = MX1 %*% Y - MX1 %*% X2 %*% bhat12
print(cbind(ehat, ehat12))
```

```
##              [,1]        [,2]
##  [1,]  0.33177812  0.33177812
##  [2,]  0.21233754  0.21233754
##  [3,] -1.46626529 -1.46626529
##  [4,]  1.42512627  1.42512627
##  [5,] -1.74135996 -1.74135996
```

```
##  [6,] -0.06935491 -0.06935491
##  [7,]  0.26358389  0.26358389
##  [8,]  0.53747283  0.53747283
##  [9,]  0.19033731  0.19033731
## [10,]  1.63498225  1.63498225
## [11,]  0.48002238  0.48002238
## [12,]  1.98243727  1.98243727
## [13,] -0.90346464 -0.90346464
## [14,]  0.31947513  0.31947513
## [15,] -1.59282064 -1.59282064
## [16,] -0.85626692 -0.85626692
## [17,]  1.04236644  1.04236644
## [18,]  0.52075031  0.52075031
## [19,] -0.55995540 -0.55995540
## [20,] -1.75118198 -1.75118198
```