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,1) # 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 )
```

Calculate the estimate as $\hat{\beta} = (X'X)^{-1}X'Y =$ (1.5814469, 1.7342763, -0.9400743, 0.9138568).

Residual

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

```
ehat = Y - X %*% bhat

print( t(X) %*% ehat )
```

```
##                [,1]
## [1,]  7.327472e-15
## [2,]  2.997602e-15
## [3,] -8.992806e-15
## [4,] -4.735795e-16
```

Notice that

- $\sum_{i=1}^{n} e_i = 21.777548$,
- $\sum_{i=1}^{n} \hat{e}_i =$ 7.77156e-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,]  0.667517949  0.667517949  0.667517949
##  [2,] -0.336382202 -0.336382202 -0.336382202
##  [3,] -0.133736534 -0.133736534 -0.133736534
##  [4,] -0.596426384 -0.596426384 -0.596426384
##  [5,]  1.262455967  1.262455967  1.262455967
##  [6,]  0.551886966  0.551886966  0.551886966
##  [7,] -0.952286535 -0.952286535 -0.952286535
##  [8,] -0.385143812 -0.385143812 -0.385143812
##  [9,]  0.181789197  0.181789197  0.181789197
## [10,]  0.598669410  0.598669410  0.598669410
## [11,] -0.002564011 -0.002564011 -0.002564011
## [12,]  2.025378301  2.025378301  2.025378301
## [13,] -0.120820260 -0.120820260 -0.120820260
## [14,] -0.069722732 -0.069722732 -0.069722732
## [15,] -0.419084147 -0.419084147 -0.419084147
## [16,] -0.454572441 -0.454572441 -0.454572441
## [17,]  0.051679428  0.051679428  0.051679428
## [18,] -0.814093355 -0.814093355 -0.814093355
## [19,] -1.600689232 -1.600689232 -1.600689232
## [20,]  0.546144425  0.546144425  0.546144425
```

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 )
```

$(\hat{\beta}_3, \hat{\beta}_4)$ = (-0.9400743, 0.9138568), which is the same as the counterpart in $\hat{\beta}$ = (1.5814469, 1.7342763, -0.9400743, 0.9138568).

```
# 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,]  0.667517949  0.667517949
##  [2,] -0.336382202 -0.336382202
##  [3,] -0.133736534 -0.133736534
##  [4,] -0.596426384 -0.596426384
##  [5,]  1.262455967  1.262455967
```

```
##  [6,]  0.551886966  0.551886966
##  [7,] -0.952286535 -0.952286535
##  [8,] -0.385143812 -0.385143812
##  [9,]  0.181789197  0.181789197
## [10,]  0.598669410  0.598669410
## [11,] -0.002564011 -0.002564011
## [12,]  2.025378301  2.025378301
## [13,] -0.120820260 -0.120820260
## [14,] -0.069722732 -0.069722732
## [15,] -0.419084147 -0.419084147
## [16,] -0.454572441 -0.454572441
## [17,]  0.051679428  0.051679428
## [18,] -0.814093355 -0.814093355
## [19,] -1.600689232 -1.600689232
## [20,]  0.546144425  0.546144425
```