Eco311 Optional Reading: Matrix Algebra

(Jing Li, Miami University)

1. For a simple regression $y = \beta_0 + \beta_1 x + u$, we can derive a simple formula for the OLS estimate of slope coefficient
$$\hat{\beta}_1 = \frac{S_{x,y}}{S_x^2} \tag{1}$$
That formula cannot be applied to a multiple regressor. To derive a compact formula for multiple regressions, we need to make use of matrix algebra.

2. Recall that a multiple regression for the i-th observation is
$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_k x_{ik} + u_i, \quad (i = 1, ...n) \tag{2}$$
and the *estimated* regression is
$$y_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + ... + \hat{\beta}_k x_{ik} + \hat{u}_i, \quad (i = 1, ...n) \tag{3}$$
Do not confuse unknown coefficient β with its estimate $\hat{\beta}$; do not confuse error term u with residual $\hat{u}$

3. We obtain the OLS estimates $\hat{\beta}$ by taking partial derivatives of $RSS = \sum_i \hat{u}_i^2$ with respect to each $\hat{\beta}$ and set them to zeros. The j-th *first order condition* is
$$\frac{\partial RSS}{\partial \hat{\beta}_j} = 0 \Rightarrow \sum_{i=1}^{n} x_{ij}\hat{u}_i = 0, \quad (j = 0, 1, ...k) \tag{4}$$
There are in total $k + 1$ first order conditions, which form a *system* of $k + 1$ equations.

4. With matrix multiplication (google it), we can neatly write that system of equations as
$$X'\widehat{U} = 0 \tag{5}$$

where

$$\widehat{U} = \begin{pmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \vdots \\ \hat{u}_n \end{pmatrix} = Y - X\hat{\beta},\ Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix},\ X = \begin{pmatrix} 1 & x_{11} & \dots & x_{1k} \\ 1 & x_{21} & \dots & x_{2k} \\ \vdots & \vdots & \dots & \vdots \\ 1 & x_{n1} & \dots & x_{nk} \end{pmatrix},\ \hat{\beta} = \begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_k \end{pmatrix} \tag{6}$$

and X' is the *transpose* of X. Notice that the first column of X matrix is the constant term.

5. It follows that

$$X'\widehat{U} = 0 \Rightarrow X'(Y - X\hat{\beta}) = 0 \Rightarrow X'Y = (X'X)\hat{\beta} \tag{7}$$

and finally assuming X has full column rank,

$$\hat{\beta} = (X'X)^{-1}(X'Y) \tag{8}$$

where $(X'X)^{-1}$ is the *inverse* of $(X'X)$.

6. Formula (8) may be the most important formula for regression analysis. It shows how to compute the OLS estimates based on the vector of Y and matrix of X. Statistical packages such as R and Stata use formula (8) to report the OLS estimate. Formula (1) is a special case of formula (8) when $k = 1$.

7. Moreover we can show the *variance-covariance matrix* of OLS estimates (assuming homoskedasticity) is

$$var(\hat{\beta}) = \sigma^2(X'X)^{-1} \tag{9}$$

where $\sigma^2 = \frac{RSS}{n-k-1}$ is the variance of regression, and σ is called *residual standard error* in R. Note that the dimension for $var(\hat{\beta})$ is $(k+1)$ by $(k+1)$. The square root of *diagonal* term of $var(\hat{\beta})$ is standard error, which can be used to compute t value, confidence interval, and p value.

8. For an illustration, consider Galton data in the mosaicData package. The results from the canned function **lm** are below

```
> library(mosaicData)
> data(Galton)
```

```
> attach(Galton)
> m = lm(height~father+mother+sex)
> summary(m)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 15.34476    2.74696   5.586 3.08e-08 ***
father       0.40598    0.02921  13.900  < 2e-16 ***
mother       0.32150    0.03128  10.277  < 2e-16 ***
sexM         5.22595    0.14401  36.289  < 2e-16 ***
---
Residual standard error: 2.154 on 894 degrees of freedom
Multiple R-squared:  0.6397,Adjusted R-squared:  0.6385
F-statistic:   529 on 3 and 894 DF,  p-value: < 2.2e-16
```

9. Next we try to use matrix algebra to duplicate some main results. First, we create the Y vector, the X matrix. Then we apply formula (8)

```
> y = height
> n = length(y)
> x = cbind(rep(1,n), father, mother, as.numeric(sex=="M"))
> k = ncol(x)
> xpx = t(x)%*%x
> xpy = t(x)%*%y
> xpxinv = solve(xpx)
> bhat = xpxinv%*%xpy
> bhat
             [,1]
       15.3447600
father  0.4059780
mother  0.3214951
        5.2259513
> coef(m)
(Intercept)      father      mother        sexM
 15.3447600   0.4059780   0.3214951   5.2259513
```

We obtain the same estimate as **lm** function. In R, functions **t** and **solve** produce transpose and inverse, respectively. Matrix multiplication is done by $\% * \%$. This vector of $\hat{\beta}$ is so important that R function **coef** can be used to get the vector.

10. After that, we can obtain standard error, t value, residual standard error, r squared

```
> uhat = y-x%*%bhat
> rss = sum(uhat^2)
> sig2 = rss/(n-k)
> vcov = sig2*xpxinv
> se = sqrt(diag(vcov))
> se
               father     mother
2.74696121 0.02920696 0.03128178 0.14400791
> t = bhat/se
> t
            [,1]
        5.586085
father 13.900044
mother 10.277392
       36.289335

> rse = sqrt(sig2)
> rse
[1] 2.154326

> rsq = 1 - sum(uhat^2)/sum((y-mean(y))^2)
> rsq
[1] 0.6396752
```

We are able to duplicate those corresponding **lm** results.

11. The benefit of learning matrix algebra cannot be overstated. For instance, for the hypothesis of $H_0 : R\beta = c$, there is a *Wald test* whose matrix formula is

$$Wald = (R\hat{\beta} - c)'[R\sigma^2(X'X)^{-1}R']^{-1}(R\hat{\beta} - c) \tag{10}$$

You don't have to rely on a canned function in R to get the Wald statistic. You can do it yourself with matrix algebra. For Galton example, the R codes to obtain Wald test for $H_0 : \beta_1 = 0, \beta_2 = 0, \beta_3 = 0$ are below

```
> R = rbind(c(0,1,0,0), c(0,0,1,0), c(0,0,0,1))
> q = nrow(R)
> wald = t(R%*%bhat)%*%solve(R%*%vcov%*%t(R))%*% (R%*%bhat)
> F = wald/q
> F
         [,1]
[1,] 529.0317
```

The F statistic 529 is based on the Wald test, and is reported by lm function. The null hypothesis implies that the model is useless (all x have zero coefficients). We reject this hypothesis because the p value is less than 0.05.