Lecture 4

Hisam Sabouni, PhD

March 2020

OLS using Matrices

Heavily borrowed from: https://web.stanford.edu/~mrosenfe/soc_meth_proj3/matrix_OLS_NYU_notes.pdf

Now that we have a general idea of how to use Ordinary Least Squares we are going to re-formulate the OLS estimator using Matrix algebra. This will allow us to more easily write out the properties of the estimator when dealing with several variables (more x's!!). We can write out the same equations we have been dealing with as follows:

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}_{n\ x\ 1} = \begin{bmatrix} 1 & x_{11} & x_{12} & x_{13} & \dots & x_{1j} \\ 1 & x_{21} & x_{22} & x_{23} & \dots & x_{2j} \\ 1 & x_{n1} & x_{n2} & x_{n3} & \dots & x_{dj} \end{bmatrix}_{n\ x\ j} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_j \end{bmatrix}_{jx1} + \begin{bmatrix} \epsilon_0 \\ \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix}_{n\ x\ 1}$$

Which can be written as simply

$$Y = X\beta + \epsilon$$

Now previously we showed that the ordinary least squares solution is the set of β's that minimizes the residual sum of squares. In matrices we can write this out as:

$$\sum_i \hat{\epsilon}_i^2 = \epsilon'\epsilon = \begin{bmatrix} \epsilon_0 & \epsilon_1 & \dots & \epsilon_n \end{bmatrix}_{1\ x\ n} \begin{bmatrix} \epsilon_0 \\ \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix}_{n\ x\ 1} = \begin{bmatrix} \epsilon_0 * \epsilon_0 + \epsilon_1 * \epsilon_1 + \dots + \epsilon_n * \epsilon_n \end{bmatrix}_{1\ x\ 1}$$

Going back to our matrix formulation we then have that:

$$\epsilon'\epsilon = (Y - X\hat{\beta})'(Y - X\hat{\beta})$$

$$= Y'Y - \hat{\beta}'X'Y - Y'X\hat{\beta} + \hat{\beta}'X'X\hat{\beta}$$

$$= Y'Y - 2\hat{\beta}'X'Y + \hat{\beta}'X'X\hat{\beta}$$

where this development uses the fact that the transpose of a scalar is the scalar i.e. $Y'X\hat{\beta} = (Y'X\hat{\beta})' = \beta'X'Y$

To find the $\hat{\beta}$ that minimizes the sum of squared residuals, we need to take the derivative with respect to $\hat{\beta}$:

$$\frac{\partial \epsilon'\epsilon}{\partial \hat{\beta}} = -2X'Y + 2X'X\hat{\beta} = 0$$

$$\frac{\partial \epsilon'\epsilon}{\partial \hat{\beta}} = X'X\hat{\beta} = X'Y$$

$$X'X\hat{\beta} = X'Y$$

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

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

Where I is an identity matrix (a matrix of all 0's off diagonal and 1's along the diagnoal; basically 1 in matrix algebra).

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

With this formulation we now are able to estimate a vector of $\hat{\beta}$ using matrices! This formulation allows us to have up to j indpendent variables!

Using this formulation we can also derive a number of implied properties of Ordinary Least Squares that had previously skipped over. For example, if we go back to:

$$X'X\hat{\beta} = X'Y$$

If we substitue in for $Y = X\hat{\beta} + \epsilon$, we have:

$$X'X\hat{\beta} = X'(X\hat{\beta} + \epsilon)$$

$$X'X\hat{\beta} = X'X\hat{\beta} + X'\epsilon$$

$$X'X\hat{\beta} = X'X\hat{\beta} + X'\epsilon$$

$$0 = X'\epsilon$$

This tells us the following:

1) The observed values of X are uncorrelated with the residuals.

$X'\epsilon$ implies that every column of x_j of X, $x_j'\epsilon = 0$. In other words, each regressor has zero sample correlation with the residuals.

2) The sum of the residuals is zero.

If there is a constant, then the first column in X (i.e. X_1) will be a column of ones. This means that for the first element in the $X'\epsilon$ vector must be zero.

3) The sample mean of the residuals is zero (from 2, just divide by n).

4) The regression hyperplane passes through the means of the observed values.

This follows from the fact that $\bar{\epsilon} = 0$. Recall that $\epsilon = Y - X\hat{\beta}$, dividing by n we have $\bar{\epsilon} = \bar{Y} - \bar{X}\hat{\beta} = 0 \rightarrow \bar{Y} = \bar{X}\hat{\beta}$. This shows that the regression hyperplane goes through the point of means of the data.

5) The predicted values of Y are uncorrelated with the residual (Same proof we did for deriving R-Squared).

6) The mean of the predicted Y's for the sample will equal the mean of the observed Y's i.e. $\bar{\hat{Y}} = \bar{y}$

Gauss-Markov Assumptions

Everything we have derived up to this point tell us about the sample properties of $\hat{\beta}$. If we want to make statistical inference about the population properties of β we need to make a series of assumptions. These assumptions are known as the Gauss-Markov Assumptions.

1. $Y = X\beta + \epsilon$

This assumption states that there is a linear relationship between Y and X. Recall that when we say linear relationship we are stating the relationship is linear in the *parameters*.

2. X is an $n\ x\ j$ matrix of full rank.

This assumption states that there is no perfect multicollinearity. In other words, the columns of X are linearly independent. This assumption is known as the identification condition. In essence, we are stating that each X_j is not a linear combination of the other $X_i\ \forall\ i \neq j$.

3. $E(\epsilon|X) = 0$

$$E\begin{bmatrix}\epsilon_1|X\\ \epsilon_2|X\\ \vdots\\ \epsilon_n|X\end{bmatrix} = \begin{bmatrix}E(\epsilon_1)\\ E(\epsilon_2)\\ \vdots\\ E(\epsilon_n)\end{bmatrix} = \begin{bmatrix}0\\ 0\\ \vdots\\ 0\end{bmatrix}$$

This assumption - the zero conditional mean assumption - states that the disturbances average out to 0 for any value of X. Put differently, no observations of the independent variables convey any information about the expected value of the disturbance. The assumption implies that $E(Y) = X\beta$. This is important since it essentially says that we get the mean function correct.

4. $E(\epsilon\epsilon'|X) = \sigma^2 I$

This captures the familiar assumption of homoskedasticity and no autocorrelation. To see why, start with the following:

$$E(\epsilon\epsilon'|X) = E\begin{bmatrix}\epsilon_1|X \\ \epsilon_2|X \\ \vdots \\ \epsilon_n|X\end{bmatrix}\begin{bmatrix}\epsilon_1|X & \epsilon_2|X & \dots & \epsilon_n|X\end{bmatrix}$$

$$E(\epsilon\epsilon'|X) = \begin{bmatrix}\epsilon_1^2|X & \epsilon_1\epsilon_2|X & \dots & \epsilon_1\epsilon_n|X \\ \epsilon_1\epsilon_2|X & \epsilon_2^2|X & \dots & \epsilon_2\epsilon_n|X \\ \vdots & \vdots & \ddots & \vdots \\ \epsilon_1\epsilon_2|X & \epsilon_2\epsilon_1|X & \dots & \epsilon_n^2|X\end{bmatrix}$$

The assumption of homoskedasticity states that the variance of ϵ_i^2 is the same σ_i^2 for all i ($Var(\epsilon_i|X) = \sigma^2$). The assumption of no autocorrelation (uncorrelated errors) means that $Cov(\epsilon_i, \epsilon_j|X) = 0$, knowing something about the disturbance term for one observation tells us nothing about the disturbance term for any other observation. With these assumptions, we have:

$$E(\epsilon\epsilon'|X) = \begin{bmatrix}\sigma^2 & 0 & \dots & 0 \\ 0 & \sigma^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma^2\end{bmatrix} = \sigma^2\begin{bmatrix}1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1\end{bmatrix} = \sigma^2 I$$

5. X may be fixed or random, but must be generated by a mechanism that is unrelated to ϵ.

6. $\epsilon|X \sim N(0, \sigma^2 I)$

Gauss-Markov Theorem

The Gauss-Markov Theorem states that, conditional on assumptions 1-5, there will be no other linear and unbiased estimator of the β coefficients that has a smaller sampling variance. In other words, the OLS estimator is the Best Linear, Unbiased and Efficient estimator (BLUE). How do we know this?

Proof that $\hat{\beta}$ is an unbiased estimator of β.

We know from earlier that $\hat{\beta} = (X'X)^{-1}X'Y$ and that $Y = X'\beta + \epsilon$:

$$\hat{\beta} = (X'X)^{-1}X'(X'\beta + \epsilon)$$

$$\hat{\beta} = \beta + (X'X)^{-1}X'\epsilon$$

Therefore in expectation:

$$E(\hat{\beta}) = E(\beta + (X'X)^{-1}X'\epsilon) = E(\beta) + E((X'X)^{-1}X'\epsilon) = \beta + (X'X)^{-1}E(X'\epsilon)$$

As long as $E(\epsilon) = 0$ and that $(X'X)^{-1}X'$ is constant (non-random), we have that:

$$E(\hat{\beta}) = \beta$$

Proof that $\hat{\beta}$ is a linear estimator of β.

We saw earlier that:

$$\hat{\beta} = \beta + (X'X)^{-1}X'\epsilon$$

We can restate this as $\hat{\beta} = \beta + A\epsilon$, where $A = (X'X)^{-1}X'$. This implies that $\hat{\beta}$ is a linear function of the disturbances.

Variance of $\hat{\beta}$

$$Var(\hat{\beta}|X) = E((\hat{\beta} - \beta)(\hat{\beta} - \beta)'|X)$$

Recall that:

$$\hat{\beta} = \beta + (X'X)^{-1}X'\epsilon \rightarrow \hat{\beta} - \beta = (X'X)^{-1}X'\epsilon$$

$$Var(\hat{\beta}|X) = E(((X'X)^{-1}X'\epsilon)((X'X)^{-1}X'\epsilon)'|X)$$

$$E((X'X)^{-1}X'\epsilon\epsilon'X(X'X)^{-1}|X) = (X'X)^{-1}X'E(\epsilon\epsilon'|X)X(X'X)^{-1}$$

Now, under the Gauss-Markov Assumptions, we have that $E(\epsilon\epsilon'|X) = \sigma^2 I$. This is a critical assumption of homoscedasticity. There are other estimators of $E(\epsilon\epsilon'|X)$, such as the Robust Huber of White estimator.

Under the Gauss-Markov Assumptions:

$$(X'X)^{-1}X'E(\epsilon\epsilon'|X)X(X'X)^{-1} = (X'X)^{-1}X'(\sigma^2 I)X(X'X)^{-1}$$

$$Var(\hat{\beta}|X) = (X'X)^{-1}X'(\sigma^2 I)X(X'X)^{-1} = (\sigma^2 I)(X'X)^{-1}X'X(X'X)^{-1} = \sigma^2 I(X'X)^{-1}$$

Note that the sample estimator of σ^2 is simply $\hat{\sigma}^2 = \frac{\epsilon'\epsilon}{n-j}$.

Given our proof of unbiasedness, proof of the variance, and the assumption that $\epsilon|X \sim N(0, \sigma^2 I)$, we now can state that:

$$\hat{\beta} \sim N(\beta, \sigma^2 I(X'X)^{-1})$$

A Robust Estimator:

$$E((X'X)^{-1}X'\epsilon\epsilon'X(X'X)^{-1}|X) = (X'X)^{-1}X'\epsilon\epsilon'X(X'X)^{-1}$$

```
# Print lots of decimals
options(scipen = 10)
# Make two explanatory variables
set.seed(1)
X_1 <- rnorm(10000, 25, 5)
X_2 <- rnorm(10000, 50, 20)
# Create a true data generating process
Y <- 210 + X_1 * 0.8 + X_2 * 0.2 + rnorm(10000,
    0, 5)
X_mat <- cbind(X_1, X_2)
```

```r
# Lets take a sample from the population data
sample_index <- sample(1:nrow(X_mat), 1000)
# Subset out the X data
X_mat_sample <- X_mat[sample_index, ]
# Subset out the Y data
Y_sample <- Y[sample_index]
# Use R function to estimate regression
base_r_linear_model <- lm(Y_sample ~ X_mat_sample)
summary(base_r_linear_model)
```

Call: lm(formula = Y_sample ~ X_mat_sample)

Residuals: Min 1Q Median 3Q Max -16.2609 -3.4699 0.1287 3.5138 15.3840

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) 210.408682 0.935036 225.03 <2e-16 ***X_mat_sampleX_1 0.785371 0.032899 23.87 <2e-16*** X_mat_sampleX_2 0.198186 0.008173 24.25 <2e-16 *** — Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ’’ 0.05 ‘.’ 0.1 ’ ’ 1

Residual standard error: 5.109 on 997 degrees of freedom Multiple R-squared: 0.5325, Adjusted R-squared: 0.5316 F-statistic: 567.9 on 2 and 997 DF, p-value: < 2.2e-16

```r
# Now lets do everything manually Add a column
# of 1's for the intercept
X <- cbind(rep(1, nrow(X_mat_sample)), X_mat_sample)
# Estimate beta hat (X'X)^-1 X'Y solve takes
# the inverse in R. t() is the transpose. %*%
# is a dot product
beta_hat <- solve(t(X) %*% X) %*% t(X) %*% Y_sample
# Get fitted values
y_hat <- X %*% beta_hat
# Extract the estimated residual term
residual_term <- as.numeric(Y_sample - y_hat)
# Calculate sample variance
sample_variance_residual <- as.numeric((t(residual_term) %*%
    residual_term)/(nrow(X) - ncol(X)))
# Create the estimated covariance matrix for
# the variance of estimated beta
```

```
beta_variance <- sample_variance_residual * solve(t(X) %*%
    X)
# Extract the standard error of our estimated
# betas
beta_se <- sqrt(diag(beta_variance))
our_estimated_results <- cbind(beta_hat, beta_se)

our_estimated_results
```

```
              beta_se
210.4086824 0.93503610
```

X_1 0.7853713 0.03289929 X_2 0.1981864 0.00817253

```
# Check if matches R output :)
summary(base_r_linear_model)
```

Call: lm(formula = Y_sample ~ X_mat_sample)

Residuals: Min 1Q Median 3Q Max -16.2609 -3.4699 0.1287 3.5138 15.3840

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) 210.408682 0.935036 225.03 <2e-16 ***X_mat_sampleX_1 0.785371 0.032899 23.87 <2e-16*** X_mat_sampleX_2 0.198186 0.008173 24.25 <2e-16 *** — Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ’’ 0.05 ‘.’ 0.1 ’ ’ 1

Residual standard error: 5.109 on 997 degrees of freedom Multiple R-squared: 0.5325, Adjusted R-squared: 0.5316 F-statistic: 567.9 on 2 and 997 DF, p-value: < 2.2e-16

```
# Another term for the robust standard errors
# is a sandwhich estimator# Lets also generate
# the robust estimator
bread <- solve(t(X) %*% X)
meat <- t(X) %*% diag(residual_term)^2 %*% X
RW <- bread %*% meat %*% bread
RW <- RW * (nrow(X)/(nrow(X) - ncol(X)))
RW
```

```
                       X_1             X_2
 0.902766669 -0.02806093839 -0.00341922140
```

X_1 -0.028060938 0.00109405655 0.00001114015 X_2 -0.003419221 0.00001114015 0.00006263587

```r
# OR use a built in R package
# install.packages(c('sandwich','lmtest'))
library(sandwich)
library(lmtest)
white_var <- vcovHC(base_r_linear_model, type = "HC1")
white_var
```

```
             (Intercept) X_mat_sampleX_1 X_mat_sampleX_2
```

(Intercept) 0.902766669 -0.02806093839 -0.00341922140 X_mat_sampleX_1 -0.028060938 0.00109405655 0.00001114015 X_mat_sampleX_2 -0.003419221 0.00001114015 0.00006263587