

Lecture 3-b

Least Squares – Review of Linear Algebra

Brooks (4th edition): Chapter 3

© R. Susmel, 2022 (for private use, not to be posted/shared online).

Review – Overall Summary

- In the first four lectures, we focused on studying one RV at a time, say stock returns, and learning about its distribution, for example, using descriptive statistics and testing hypothesis about its moments.

- Last class we changed the focus to functional linear relations between RVs: y (**dependent variable**) & \mathbf{x} (vector of **explanatory variables**):

$$y_i = \alpha + \beta_1 x_{1,i} + \beta_2 x_{2,i} + \dots + \beta_k x_{k,i} + \varepsilon_i \quad i = 1, 2, \dots, N$$

where α & the β 's are parameters to be estimated and ε_i is the error term or *disturbance*. We think of ε_i as the effect of individual i variation that is not “controlled for” with the x_i 's. ε_i is part of the model.

- We call the above equation the **Data Generating Process (DGP)**. The DGP represents the model that generates the observed data.

Review – Linear Model: One Variable Case

- **DGP:** (linear): $y_i = \alpha + \beta x_i + \varepsilon_i, \quad i = 1, 2, \dots, N.$

If we assume $E[\varepsilon_i] = 0$, we have:

$$E[y_i] = \alpha + \beta E[x_i].$$

Example: The CAPM relates excess return of asset i , $y_i = r_{i,t} - r_f$, to the excess return of the market, $x_i = r_{m,t} - r_f$. In equilibrium:

$$E[(r_{i,t} - r_f)] = \beta_i E[(r_{m,t} - r_f)],$$

where β_i is the sensitivity of asset i to market risk.

- **CAPM DGP:** $y_i = \alpha + \beta x_i + \varepsilon_i, \quad i = 1, 2, \dots, N.$

where α & β are parameters to be estimated. Once we estimate α & β , we can test the CAPM for IBM, since according to the CAPM $\alpha = 0$.

Review – Linear Model: Multivariate Case

- In the CAPM DGP, only “the market” explains excess returns for any asset i . But, there are other models (DGPs) for excess returns with more explanatory variables, for example, the 3-factor **Fama-French model**: Market, SMB (size factor), and HML (book-to-market).

The 3-factor FF model represents a **multivariate model** for asset i :

Fama-French DGP: $y_i = \alpha + \beta_1 x_{1,i} + \beta_2 x_{2,i} + \beta_3 x_{3,i} + \varepsilon_i$

- Though not necessary correct, we usually think of y as the *endogenous* variable and x as the *exogenous* variable determined “outside” the model.

Goal: Estimation of population parameters α & β to learn the DGP.

Review – Least Squares Estimation

- We start with a model, a linear **DGP**:

$$y_i = \beta_1 x_{1,i} + \beta_2 x_{2,i} + \beta_3 x_{3,i} + \dots + \beta_k x_{k,i} + \varepsilon_i$$

we estimate its parameters by *Ordinary Least Squares* (**OLS**). That is we estimate the vector $\beta = \{\beta_1, \beta_2, \dots, \beta_k\}$ by minimizing

$$S(\mathbf{x}; \theta) = \sum_{i=1}^T \varepsilon_i^2 = \sum_i^T (y_i - \beta_1 x_{1,i} - \beta_2 x_{2,i} - \dots - \beta_k x_{k,i})^2$$

- We take first derivatives with respect to $\beta_1, \beta_2, \dots, \beta_k$. Then, we set them equal to 0 & get k f.o.c. Finally, we solve for \mathbf{b} (OLS estimator).
- For the $k = 2$, and assuming $x_{1,i} = 1$, the f.o.c. are:

$$(\beta_1): \sum_i^T (y_i - b_1 - b_2 x_i) = 0 \quad (1)$$

$$(\beta_2): \sum_i^T (y_i x_i - b_1 x_i - b_2 x_i^2) = 0 \quad (2)$$

Review – OLS & One Variable Case: Derivation

- Next, we solve for b_1 & b_2 , the OLS estimators.

$$b_1 = \bar{y} - b_2 \bar{x}$$

$$b_2 = \frac{\sum_i^T (y_i - \bar{y}) x_i}{\sum_i^T (x_i - \bar{x}) x_i} = \frac{\text{cov}(y_i, x_i)}{\text{var}(x_i)}$$

- Interpretation of coefficients

- b_1 estimates the *constant* of the regression: IBM excess returns in excess of Market excess returns. In the CAPM, it should be 0 ($= \alpha_i$).

- b_2 estimates the *slope* of the regression. In the CAPM: β_i

$$\frac{\delta y_i}{\delta x_i} = \beta_i = \frac{\text{cov}(r_{i=IBM,t} - r_f, r_{m,t} - r_f)}{\text{var}(r_{m,t} - r_f)}$$

That is, if Market excess returns increase by one 1%, then IBM excess returns are expected to increase by $b_2 (= \beta_i)$ units (say, $b_2\%$).

Review – OLS & One Variable Case: CAPM

- Interpretation of coefficients

- b_1 estimates the *constant* of the regression: IBM excess returns in excess of Market excess returns. In the CAPM, it should be 0 ($= \alpha_i$).

- b_2 estimates the *slope* of the regression. In the CAPM: β_i

$$\frac{\delta y_i}{\delta x_i} = \beta_i = \frac{\text{cov}(r_{i=IBM,t} - r_f, r_{m,t} - r_f)}{\text{var}(r_{m,t} - r_f)}$$

That is, if Market excess returns increase by one 1%, then IBM excess returns are expected to increase by $b_2 (= \beta_i)$ units (say, $b_2\%$). The β_{IBM} also tells us if IBM is riskier ($\beta_{IBM} > 1$) or safer ($\beta_{IBM} < 1$) than the market.

Review – OLS & One Variable Case: CAPM

- Conditional Prediction

Suppose analysts estimate that Market excess returns are 2%, then, we estimate (or predict, given the 2% value for Market excess returns):

$$\text{Predicted [IBM excess returns] } (r_{m,t} - r_f) = .02 = b_1 + b_2 * .02.$$

We will call the Predicted $y_i = \hat{y}_i = \text{fitted value}$.

Review – OLS: CAPM – R Estimation

Example (continuation):

```
> summary(fit_ibm_capm)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	-0.005791	0.002487	-2.329	0.0202 *	b_1
xMkt_RF	0.895774	0.053867	16.629	<2e-16 ***	$b_2 = \hat{\beta}_{CAPM}$

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.05887 on 567 degrees of freedom

Multiple R-squared: 0.3278, Adjusted R-squared: 0.3266

F-statistic: 276.5 on 1 and 567 DF, p-value: < 2.2e-16

Interpretation of b_1 :

b_1 = constant. The additional IBM return, after excess market returns are incorporated, is -0.58%. Under the CAPM, b_1 should be close to 0.

Review – OLS: CAPM – R Estimation

Example (continuation):

Interpretation of b_2 : In addition, the estimate of β_{IBM} ($\beta < 1$) implies that IBM is less volatile (“safer”) than the market.

Conditional prediction of IBM excess returns:

Suppose market excess returns increase are 2%, then we predict IBM excess returns = $-0.005791 + 0.895774 * .02 = 0.01212$ (1.21%).

Note: According to the CAPM, IBM underperformed:

- IBM excess returns (CAPM) = $0.895774 * \text{mean}(\text{Mkt_RF})$
 $= 0.895774 * 0.0056489 = 0.0050601$
- IBM excess returns (sample) = $\text{mean}(\text{ibm_x}) = -0.00073141$

Review – OLS: Multivariate Case

- The CAPM is a particular case of what in financial theory we call “*factor models*.” Factors represent the systematic component that drives the cross-section of returns over time. For example, a *k-factor model* for excess returns is given by:

$$r_{i,t} - r_f = \alpha_i + \beta_1 f_{1,t} + \beta_2 f_{2,t} + \dots + \beta_k f_{k,t} + \varepsilon_{i,t}$$

where $f_{j,t}$ is the j (common) factor at time t , and constant over i , and $\varepsilon_{i,t}$ represents the idiosyncratic component of asset i .

- The higher the exposure –i.e., β_i – the higher the expected compensation.
- The CAPM has only one factor: market excess returns (“*the market*”).

Review – OLS: Multivariate Case

- LS is a general estimation method. It allows any functional form for the relation between y_i and x_i . It also allows y_i to be related to many explanatory variables, like multi-factor models for excess returns.

In this lecture, we cover the case where the DGP is **linear**. We assume a linear system with k independent variables and T observations. That is,

$$y_i = \beta_1 x_{1,i} + \beta_2 x_{2,i} + \dots + \beta_k x_{k,i} + \varepsilon_i, \quad i = 1, 2, \dots, T$$

The whole system (for all i) is:

$$\begin{aligned} y_1 &= \beta_1 x_{11} + \beta_2 x_{12} + \dots + \beta_k x_{k1} + \varepsilon_1 \\ y_2 &= \beta_1 x_{12} + \beta_2 x_{22} + \dots + \beta_k x_{k2} + \varepsilon_2 \\ &\dots \quad \dots \quad \dots \quad \dots \\ y_T &= \beta_1 x_{1T} + \beta_2 x_{2T} + \dots + \beta_k x_{kT} + \varepsilon_T \end{aligned}$$

Review – OLS: Multivariate Case

- It is cumbersome to write the whole system. Using linear algebra, we can rewrite the system in a more compact and simplify derivations.

Example: Using vector & matrix notation, we write the system as:

$$y = f(\mathbf{X}, \boldsymbol{\theta}) + \boldsymbol{\varepsilon} = \mathbf{X} \boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

Notation: \mathbf{y} , $\boldsymbol{\beta}$ & $\boldsymbol{\varepsilon}$ are vectors:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_T \end{bmatrix}, \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_T \end{bmatrix}, \quad \& \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_k \end{bmatrix}$$

\mathbf{X} is a matrix:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{k1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1T} & x_{2T} & \cdots & x_{kT} \end{bmatrix}$$

Linear Algebra: Brief Review – Matrix

- Life (& notation) becomes easier with linear Algebra. Concepts:

- A Matrix.

A matrix is a set of elements, organized into rows and columns

$$\begin{array}{c} \text{columns} \\ \text{rows} \end{array} \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

- a and d are the diagonal elements.
- b and c are the off-diagonal elements.
- Matrices are like plain numbers in many ways: they can be added, subtracted, and, in some cases, multiplied and inverted (divided).

Linear Algebra: Matrices and Vectors

Examples:

$$A = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix}; \quad b = [b_1 \quad b_2 \quad b_3]$$

- Dimensions of a matrix: numbers of rows by numbers of columns. The Matrix **A** is a 2x2 matrix, **b** is a 1x3 matrix.
- A matrix with only 1 column or only 1 row is called a *vector*.
- If a matrix has an equal numbers of rows and columns, it is called a *square* matrix. Matrix **A**, above, is a square matrix.
- Usual Notation:
 Upper case letters \Rightarrow matrices
 Lower case \Rightarrow vectors

15

Linear Algebra: Matrices – Information

- Information is described by data. A tool to organize the data is a list, which we call a vector. Lists of lists are called matrices. That is, we organize the data using matrices, say, **X**.
- We think of the elements of **X** as data points (“data entries”, “observations”), in economics, we usually have numerical data.
- We store the data in rows. In a $T \times k$ matrix, **X**, over time we build a database:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{k1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1T} & x_{2T} & \cdots & x_{kT} \end{bmatrix} \quad \begin{array}{l} \text{row } 1 = k \text{ entries at time } 1 \\ \text{row } T = k \text{ entries at time } T \end{array}$$

- Once the data is organized in matrices it can be easily manipulated: multiplied, added, etc. (This is what Excel does very well).

16

Linear Algebra: Matrices in Econometrics

- We want to estimate a model: $y = f(x_1, x_2, \dots, x_k)$. We collect data, T (or N) observations, on a dependent variable, y , and on k explanatory variables, X .
- Usual notation: vectors are column vectors: y & x_j are $T \times 1$ vectors:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_T \end{bmatrix} \quad \& \quad x_j = \begin{bmatrix} x_{j1} \\ x_{j2} \\ \vdots \\ x_{jT} \end{bmatrix} \quad j = 1, \dots, k$$

X is a $T \times k$ matrix:
$$X = \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{k1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1T} & x_{2T} & \cdots & x_{kT} \end{bmatrix}$$

Its columns are the k $T \times 1$ vectors x_j . It is common to treat x_1 as vector of ones, $\mathbf{1}$.

17

Linear Algebra: Matrices in Econometrics

- In general, we import matrices (information) to our programs.

Example: In R, we use the **read** function, usually followed by the type of data we are importing. Below, we import a comma separated values (csv) file with monthly CPIs and exchange rates for 20 different countries, then we use the **read.csv** function:

```
PPP_da <-
read.csv("http://www.bauer.uh.edu/rsusmel/4397/ppp_m.csv", head=TRUE, sep=
",")
```

The **names()** function describes the headers of the file imported (41 headers):

```
> names(PPP_da)
[1] "Date"   "BG_CPI" "IT_CPI" "GER_CPI" "UK_CPI"
[6] "SWED_CPI" "DEN_CPI" "NOR_CPI" "IND_CPI" "JAP_CPI"
[11] "KOR_CPI" "THAI_CPI" "SING_CPI" "MAL_CPI" "KUW_CPI"
[16] "SUAD_CPI" "CAN_CPI" "MEX_CPI" "US_CPI"  "EGY_CPI"
[...]
```

18

Linear Algebra: Matrices in Econometrics

Example (continuation):

The **summary()** function provides some stats of variables imported:

```
>summary(PPP_da)
      Date      BG_CPI      IT_CPI      GER_CPI
1/15/1971: 1  Min.   :19.77  Min.   : 5.90  Min.   :31.20
1/15/1972: 1  1st Qu.:49.32  1st Qu.:32.25  1st Qu.:57.17
1/15/1973: 1  Median :69.91  Median :67.30  Median :75.30
1/15/1974: 1  Mean    :67.92  Mean    :60.14  Mean    :72.29
1/15/1975: 1  3rd Qu.:89.40  3rd Qu.:89.65  3rd Qu.:91.17
1/15/1976: 1  Max.    :109.71  Max.    :103.50  Max.    :106.60
(Other) :588
```

We extract a variable from the matrix by the name of file followed by \$ and the header of variable:

```
x_chf <- PPP_da$CHF_USD          # extract CHF/USD exchange rate data
```

We can transform the vector x_chf. For example, for % changes:

```
T <- length(x_chf)                # length of CHF/USD exchange rate data
lr_chf <- log(x_chf[-1]/x_chf[-T]) # create log returns (changes) for the CHF/USD
```

19

Linear Algebra: Special Matrices

- *Identity Matrix, I*: A square matrix with 1's along the diagonal and 0's everywhere else. Similar to scalar "1":

$$\mathbf{A} * \mathbf{I} = \mathbf{A}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- *Null matrix, 0*: A matrix in which all elements are 0's. Similar to scalar "0":

$$\mathbf{A} * \mathbf{0} = \mathbf{0}$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

- Both are *diagonal* matrices \Rightarrow off-diagonal elements are zero.
- Both are examples of *symmetric* matrices. That is, element a_{ij} is equal to element a_{ji} . (Later, we'll see $\mathbf{A} = \mathbf{A}^T$). For example:

$$\mathbf{A} = \begin{bmatrix} 2 & 5 & 9 \\ 5 & -1 & 0 \\ 9 & 0 & 1 \end{bmatrix} \quad \text{is a symmetric matrix.}$$

20

Linear Algebra: Multiplication

- We want to multiply two matrices: $\mathbf{A} * \mathbf{B}$. But, multiplication of matrices requires a *conformability condition*.
- Conformability condition: The column dimensions of the lead matrix \mathbf{A} must be equal to the row dimension of the lag matrix \mathbf{B} .
- If \mathbf{A} is an $(m \times n)$ and \mathbf{B} an $(n \times p)$ matrix (\mathbf{A} has the same number of columns as \mathbf{B} has rows), then we define the product of \mathbf{AB} :
 $\mathbf{AB} = \mathbf{C}$ is $(m \times p)$ matrix with its ik^{th} element is $c_{ik} = \sum_{j=1}^n a_{ij} b_{jk}$.

Example: Given $\mathbf{a}(1 \times 2)$ and $\mathbf{B}(2 \times 3)$, we compute \mathbf{aB} :

$$\begin{aligned} \mathbf{aB} &= [a_{11} \ a_{12}] \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix} = \mathbf{c} = [c_{11} \ c_{12} \ c_{13}] \\ &= [a_{11}b_{11} + a_{12}b_{21} \quad a_{11}b_{12} + a_{12}b_{22} \quad a_{11}b_{13} + a_{12}b_{23}] \end{aligned}$$

- Dimensions: $\mathbf{a}(1 \times 2), \mathbf{B}(2 \times 3) \Rightarrow \mathbf{c}(1 \times 3)$

21

Linear Algebra: Multiplication

Example: We want to multiply $\mathbf{A}(2 \times 2)$ and $\mathbf{B}(2 \times 2)$, where \mathbf{A} has elements a_{ij} and \mathbf{B} has elements b_{jk} . Recall the ik^{th} element is

$$c_{ik} = \sum_{j=1}^{n=2} a_{ij} b_{jk}$$

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 7 & 9 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 2 & 1 \\ 7 & 9 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 4 = 2 * 1 + 1 * 2 & 3 = 2 * 0 + 1 * 3 \\ 25 = 7 * 1 + 9 * 2 & 27 = 7 * 0 + 9 * 3 \end{bmatrix}$$

$$\mathbf{C}_{2 \times 2} = \mathbf{A}_{2 \times 2} * \mathbf{B}_{2 \times 2}$$

- Dimensions: $\mathbf{A}(2 \times 2), \mathbf{B}(2 \times 2) \Rightarrow \mathbf{C}(2 \times 2)$, a square matrix.

22

Linear Algebra: Multiplication

Example: We want to multiply \mathbf{X} (2×2) and $\boldsymbol{\beta}$ (2×1), where \mathbf{X} has elements x_{ij} and \mathbf{b} has elements β_j :

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{21} \\ x_{12} & x_{22} \end{bmatrix} \quad \& \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix}$$

We compute

$$\mathbf{y} = \mathbf{X} \boldsymbol{\beta}$$

Recall the i^{th} element is

$$y_i = \sum_{j=1}^{n=2} x_{ij} \beta_j$$

Then,

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_{11} & x_{21} \\ x_{12} & x_{22} \end{bmatrix} * \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} x_{11} \beta_1 + x_{21} \beta_2 \\ x_{12} \beta_1 + x_{22} \beta_2 \end{bmatrix}$$

- Dimensions: $\mathbf{X}(2 \times 2)$, $\boldsymbol{\beta}(2 \times 1) \Rightarrow \mathbf{y}(2 \times 1)$, a row vector.

23

Linear Algebra: Transpose

- The transpose of a matrix \mathbf{A} is another matrix \mathbf{A}^T (also written \mathbf{A}') created by any one of the following equivalent actions:
 - write the rows (columns) of \mathbf{A} as the columns (rows) of \mathbf{A}^T
 - reflect \mathbf{A} by its main diagonal to obtain \mathbf{A}^T

Example: $A = \begin{bmatrix} 3 & 8 & -9 \\ 1 & 0 & 4 \end{bmatrix} \Rightarrow A' = \begin{bmatrix} 3 & 1 \\ 8 & 0 \\ -9 & 4 \end{bmatrix}$

- If \mathbf{A} is a $m \times n$ matrix $\Rightarrow \mathbf{A}^T$ is a $n \times m$ matrix.
- $(\mathbf{A}')' = \mathbf{A}$
- Conformability changes unless the matrix is square.
- $(\mathbf{AB})' = \mathbf{B}'\mathbf{A}' \Rightarrow (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})' = \mathbf{y}' - \boldsymbol{\beta}'\mathbf{X}'$

24

Linear Algebra: Transpose – Example (\mathbf{X}')

- In econometrics, an important matrix is $\mathbf{X}'\mathbf{X}$. Recall \mathbf{X} (usually, the matrix of k independent explanatory variables):

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{k1} \\ x_{12} & x_{22} & \cdots & x_{k2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1T} & x_{2T} & \cdots & x_{kT} \end{bmatrix} \quad \text{a } (T \times k) \text{ matrix}$$

Then,

$$\mathbf{X}' = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1T} \\ x_{21} & x_{22} & \cdots & x_{2T} \\ \vdots & \vdots & \ddots & \vdots \\ x_{k1} & x_{k2} & \cdots & x_{kT} \end{bmatrix} \quad \text{a } (k \times T) \text{ matrix}$$

25

Linear Algebra: Math Operations

- Addition, Subtraction, Multiplication

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a+e & b+f \\ c+g & d+h \end{bmatrix} \quad \text{Just add elements}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} - \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a-e & b-f \\ c-g & d-h \end{bmatrix} \quad \text{Just subtract elements}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae+bg & af+bh \\ ce+dg & cf+dh \end{bmatrix} \quad \text{Multiply each row by each column and add}$$

$$k \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} ka & kb \\ kc & kd \end{bmatrix} \quad \text{Multiply each element by the scalar}$$

26

Linear Algebra: Math Operations – Examples

- Matrix addition
$$\begin{bmatrix} 2 & 1 \\ 7 & 9 \end{bmatrix} + \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 5 & 2 \\ 7 & 11 \end{bmatrix}$$
$$A_{2 \times 2} + B_{2 \times 2} = C_{2 \times 2}$$
- Matrix subtraction
$$\begin{bmatrix} 2 & 1 \\ 7 & 9 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 5 & 6 \end{bmatrix}$$
- Matrix multiplication
$$\begin{bmatrix} 2 & 1 \\ 7 & 9 \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 4 & 3 \\ 25 & 27 \end{bmatrix}$$
$$A_{2 \times 2} \times B_{2 \times 2} = C_{2 \times 2}$$
- Scalar multiplication
$$\frac{1}{8} \begin{bmatrix} 2 & 4 \\ 6 & 1 \end{bmatrix} = \begin{bmatrix} 1/4 & 1/2 \\ 3/4 & 1/8 \end{bmatrix}$$

27

27

Linear Algebra: Math Operations – $\epsilon'\epsilon$

- In LS estimation, we minimize a sum of square errors

$$S(x_i, \beta) = \sum_{i=1}^T \epsilon_i^2$$

Since ϵ is $T \times 1$ vector, we use linear algebra to write the sum of squares of its elements as (dot product of 2 $T \times 1$ vectors):

$$S(x_i, \beta) = \sum_{i=1}^T \epsilon_i^2 = \epsilon' \epsilon \quad (\text{a scalar})$$

Check:

$$\epsilon' \epsilon = [\epsilon_1 \ \epsilon_2 \ \dots \ \epsilon_T] * \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_T \end{bmatrix} = [\epsilon_1^2 + \epsilon_2^2 + \dots + \epsilon_T^2] = \sum_{i=1}^T \epsilon_i^2$$

Thus, we pick β to minimize:

$$S(x_i, \beta) = \epsilon' \epsilon = (y - X\beta)' (y - X\beta)$$

Linear Algebra: Math Operations – $\mathbf{X}'\mathbf{X}$

- A special matrix in econometrics, $\mathbf{X}'\mathbf{X}$ (a $k \times k$ matrix, with $k = 2$):

- $\mathbf{X} (T \times 2) = \begin{bmatrix} x_{11} & x_{21} \\ x_{12} & x_{22} \\ \vdots & \vdots \\ x_{1T} & x_{2T} \end{bmatrix}$

- $\mathbf{X}' (2 \times T) = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1T} \\ x_{21} & x_{22} & \cdots & x_{2T} \end{bmatrix}$ Sum of squares of x_i ($i = 1, 2$)

$$\begin{aligned} \mathbf{X}'\mathbf{X} (2 \times 2) &= \begin{bmatrix} \sum_{i=1}^T x_{1i}^2 & \sum_{i=1}^T x_{2i}x_{1i} \\ \sum_{i=1}^T x_{1i}x_{2i} & \sum_{i=1}^T x_{2i}^2 \end{bmatrix} = \sum_{i=1}^T \begin{bmatrix} x_{1i}^2 & x_{2i}x_{1i} \\ x_{2i}x_{1i} & x_{2i}^2 \end{bmatrix} \\ &= \sum_{i=1}^T \begin{bmatrix} x_{1i} \\ x_{2i} \end{bmatrix} \begin{bmatrix} x_{1i} & x_{2i} \end{bmatrix} \quad \text{Sum of cross products of } x_1 \text{ } x_2 \\ &= \sum_{i=1}^T \mathbf{x}_i \mathbf{x}_i' \end{aligned}$$

29

Linear Algebra: Math Operations – $\mathbf{X}'\mathbf{X}$

- In general, with a $k \times k$ $\mathbf{X}'\mathbf{X}$ matrix:

- $\mathbf{X} (T \times k) = \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{k1} \\ x_{12} & x_{22} & \cdots & x_{k2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1T} & x_{2T} & \cdots & x_{kT} \end{bmatrix}$ & $\mathbf{X}' = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1T} \\ x_{21} & x_{22} & \cdots & x_{2T} \\ \vdots & \vdots & \ddots & \vdots \\ x_{k1} & x_{k2} & \cdots & x_{kT} \end{bmatrix}$

- $\mathbf{X}'\mathbf{X} (k \times k) = \begin{bmatrix} \sum_{i=1}^T x_{1i}^2 & \sum_{i=1}^T x_{1i}x_{2i} & \cdots & \sum_{i=1}^T x_{1i}x_{ki} \\ \sum_{i=1}^T x_{2i}x_{1i} & \sum_{i=1}^T x_{2i}^2 & \cdots & \sum_{i=1}^T x_{2i}x_{ki} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^T x_{ki}x_{1i} & \sum_{i=1}^T x_{ki}x_{2i} & \cdots & \sum_{i=1}^T x_{ki}^2 \end{bmatrix} =$

$$= \sum_{i=1}^T \begin{bmatrix} x_{1i}^2 & \cdots & x_{1i}x_{ki} \\ \vdots & \ddots & \vdots \\ x_{ki}x_{1i} & \cdots & x_{ki}^2 \end{bmatrix} \quad \text{(a symmetric matrix)}$$

30

Linear Algebra: Inverse of a Matrix

- Identity matrix: $\mathbf{AI} = \mathbf{A}$, where $I_j = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$

Notation: I_j is a $j \times j$ identity matrix.

- Given \mathbf{A} ($m \times n$), the matrix \mathbf{B} ($n \times m$) is a *right-inverse* for \mathbf{A} iff $\mathbf{AB} = \mathbf{I}_m$
- Given \mathbf{A} ($m \times n$), the matrix \mathbf{C} ($n \times m$) is a *left-inverse* for \mathbf{A} iff $\mathbf{CA} = \mathbf{I}_n$
- **Theorem:** If \mathbf{A} ($m \times n$), has both a *right-inverse* \mathbf{B} and a *left-inverse* \mathbf{C} , then $\mathbf{C} = \mathbf{B} = \mathbf{A}^{-1}$

Note:

- If \mathbf{A} has both a right and a left inverse, it is a **square matrix** ($m=n$). It is called *invertible*. We say “the matrix \mathbf{A} is *non-singular*.”
- This matrix, \mathbf{A}^{-1} , is unique.
- If $\det(\mathbf{A}) \neq 0 \Rightarrow \mathbf{A}$ is non-singular.

31

Linear Algebra: Symmetric Matrices

Definition:

If $\mathbf{A}' = \mathbf{A}$, then \mathbf{A} is called a *symmetric* matrix.

- In many applications, matrices are often symmetric. For example, in statistics the *correlation matrix* and the *variance covariance matrix*.
- Symmetric matrices play the same role as real numbers do among the complex numbers.
- We can do calculations with symmetric matrices like with numbers: for example, we can solve $\mathbf{B}^2 = \mathbf{A}$ for \mathbf{B} if \mathbf{A} is symmetric matrix (& \mathbf{B} is square root of \mathbf{A} .) This is not possible in general.
- $\mathbf{X}'\mathbf{X}$ is symmetric. It plays a very important role in econometrics.

32

Linear Algebra: Operations in R

- Many ways to create a vector (`c`, `2:7`, `seq`, `rep`, etc) or a matrix (`c`, `cbind`, `rbind`). We use **`c()`**, the **combine function**:

```
v1 <- c(1, 3, 8)           # a (3x1) vector (vectors are usually treated as a column list)
> v1
[1] 1 3 8
A <- matrix(c(1, 2, 3, 7, 8, 9), ncol = 3) # a (2x3) matrix
> A
      [,1] [,2] [,3]
[1,]    1    3    8
[2,]    2    7    9
B <- matrix(c(1, 3, 1, 1, 2, 0), nrow = 3)
> B
      [,1] [,2]
[1,]    1    1
[2,]    3    2
[3,]    1    0
```

33

Linear Algebra: Operations in R

- Now, we use **`rbind`** to create **A** and **`cbind`** to create **B**

```
v1 <- c(1, 3, 8)           # a (3x1) vector
v2 <- c(2, 7, 9)
A <- rbind(v1, v2)
> A
      [,1] [,2] [,3]
v1    1    3    8
v2    2    7    9
v3 <- c(1, 3, 1)
v4 <- c(1, 2, 0)
B <- cbind(v3, v4)
> B
      v3 v4
[1,]    1    1
[2,]    3    2
[3,]    1    0
```

34

Linear Algebra: Operations in R

- Matrix addition/subtraction: $+/ -$ –element by element.

- Matrix multiplication: $\%*\%$

`C <- A%*%B` #A is 2x3; B is 3x2 \Rightarrow C is 2x2

`> C`

`[,1] [,2]`

`[1,] 18 7`

`[2,] 32 16`

- Scalar multiplication: $*$

`> 2 * C`

elementwise multiplication of C by scalar 2

`[,1] [,2]`

`[1,] 36 14`

`[2,] 64 32`

Note: Usually, matrices will be data –i.e., read as input.

35

Linear Algebra: Operations in R

- Dot product “ \bullet ” is a function that takes pairs of vectors and produces a number. For vectors \mathbf{c} & \mathbf{z} , it is defined as:

$$\mathbf{c} \bullet \mathbf{z} = \mathbf{c}'\mathbf{z} = \mathbf{z}'\mathbf{c} = c_1 * z_1 + c_2 * z_2 + \dots + c_n * z_n = \sum_{i=1}^n c_i z_i$$

- Dot product with 2 vectors: $\mathbf{v1} \bullet \mathbf{v2}$ produces a sum of the elementwise multiplied elements of both vectors

`> t(v1) %*% v2`

v1 <- c(1, 3, 8) & v2 <- c(2, 7, 9)

`[,1]`

`[1,] 95`

- Dot product with a vector itself: $\mathbf{v1} \bullet \mathbf{v1}$ produces a sum of the square elements of vector

`> t(v1) %*% v1`

v1 <- c(1, 3, 8) $\Rightarrow 1^2 + 3^2 + 8^2 = 1+9+64$

`[,1]`

`[1,] 74`

36

Linear Algebra: Operations in R

- Dot product with \mathbf{i} (a vector of ones): sum of elements of vector

```
i <- c(1,1,1)           # define a unit vector
> t(i) %*% v1            # v1 <- c(1, 3, 8)
[1,] 12
```

- Product of 2 vectors: $\mathbf{v1}$ & $\mathbf{t(v2)}$: \mathbf{A} (3x3) matrix.

```
> v1 %*% t(v2)           # v2 <- c(2, 7, 9)  --a (3x1) vector x (1x3) vector
[1,] [2,] [3,]
[1,]  2  7  9
[2,]  6 21 27
[3,] 16 56 72
```

Property of dot product: If the dot product of two vectors is equal to zero, then the vectors are *orthogonal* (perpendicular or “ \perp ”) vectors. We interpret this as “the vectors are *uncorrelated*.”

37

Linear Algebra: Operations in R

- Matrix transpose: \mathbf{t}

```
> t(B)                  #B is 3x2 => t(B) is 2x3
[1,] [2,] [3,]
[1,]  1  3  1
[2,]  1  2  0
```

```
> B
      v3 v4
[1,]  1  1
[2,]  3  2
[3,]  1  0
```

- $\mathbf{X'X}$ (a symmetric matrix)

```
> t(B) %*% B            # command crossprod(B) is more efficient
[1,] [2,]
[1,] 11  7
[2,]  7  5
```

- Determinant: \mathbf{det}

```
> det(t(B) %*% B)      # Matrix has to be square. If det(A)=0 => A non-invertible
[1] 6
```

38

Linear Algebra: Operations in R

- $(X'X)^{-1}$: Inverse: **solve**

```
> solve(t(B)%*%B)           #Matrix inside solve() has to be square
      [,1] [,2]
[1,] 0.8333333 -1.166667
[2,] -1.166667 1.833333
```

- Take the diagonal elements of a matrix A: **diag()**

```
> diag(solve(t(B)%*%B))
[1] 0.8333333 1.833333
```

- Square root of (positive) elements of a matrix A: **sqrt()**

```
> sqrt(diag(solve(t(B)%*%B)))
      v3      v4
0.9128709 1.3540064
```

39

Linear Algebra: Example 1 – Linear DGP

- We have a model (the **DGP**) that linearly relates a dependent variable, y , k explanatory variables, \mathbf{X} , and an error term, $\boldsymbol{\varepsilon}$. There are k unknown parameters, $\boldsymbol{\beta}$. We have T observations of y and \mathbf{X} .

- **DGP:**

$$y_i = \sum_{j=1}^k x_{ji} \beta_j + \varepsilon_i \quad i = 1, 2, \dots, T.$$

Or

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_T \end{bmatrix} = \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{k1} \\ x_{21} & x_{22} & \cdots & x_{k1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1T} & x_{2T} & \cdots & x_{kT} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_k \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_T \end{bmatrix}$$

Or using matrix notation:

$$\mathbf{y} = \mathbf{X} \boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

where \mathbf{y} & $\boldsymbol{\varepsilon}$ are $(T \times 1)$; \mathbf{X} is $(T \times k)$; and $\boldsymbol{\beta}$ is $(k \times 1)$.

40

Linear Algebra: Example 2 – Linear System

- Assume an economic model as system of linear equations with:
 a_{ij} parameters, where $i = 1, \dots, m$ rows, $j = 1, \dots, n$ columns
 x_i endogenous variables (n),
 d_i exogenous variables and constants (m).

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = d_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = d_2 \\ \dots \quad \dots \quad \dots \quad \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = d_m \end{cases}$$

- We can write this system using linear algebra notation: $\mathbf{A} \mathbf{x} = \mathbf{d}$

$$\begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ \dots \\ d_m \end{bmatrix}$$

$\mathbf{A} = (m \times n)$ matrix

$\mathbf{x} = (n \times 1)$ column vector

$\mathbf{d} = (m \times 1)$ column vector

- Q: What is the nature of the set of solutions to this system?

41

Linear Algebra: Example 2 – Linear System

- System of linear equations: $\mathbf{A} \mathbf{x} = \mathbf{d}$

where

$\mathbf{A} = (m \times n)$ matrix of parameters

\mathbf{x} = column vector of endogenous variables ($n \times 1$)

\mathbf{d} = column vector of exogenous variables and constants ($m \times 1$)

- Solve for \mathbf{x}^*

- Theorem:** Given \mathbf{A} ($m \times n$) invertible. Then, the equation $\mathbf{A} \mathbf{x} = \mathbf{d}$ has one and only one solution for every \mathbf{d} ($m \times 1$). That is, there is a unique \mathbf{x}^* .

$$\Rightarrow \mathbf{x}^* = \mathbf{A}^{-1} \mathbf{d}$$

Example: In practice, we avoid computing \mathbf{A}^{-1} , we solve a system.

```
A <- matrix(c(1, 1, 5, 7, 9, 11, 10, 10, 14), ncol = 3) # check det(A) for singularity (det(A)=-72)
d <- c(2, 5, 2)
> solve(A,d)
[1] -0.7222222 1.5000000 -0.7777778
```

42

Linear Algebra: Linear Dependence and Rank

- A set of vectors is *linearly dependent* if any one of them can be expressed as a linear combination of the remaining vectors; otherwise, it is linearly independent.

- Formal definition: Linear independence (LI)

The set $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k\}$ is called a *linearly independent* set of vectors iff

$$c_1 \mathbf{u}_1 + c_2 \mathbf{u}_2 + \dots + c_k \mathbf{u}_k = \mathbf{0} \quad \Rightarrow \quad c_1 = c_2 = \dots = c_k = 0.$$

Notes:

- Dependence prevents solving a system of equations (\mathbf{A} is not invertible). More unknowns than independent equations.
- The number of linearly independent rows or columns in a matrix is the *rank* of a matrix ($\text{rank}(\mathbf{A})$).
- If \mathbf{A} , a $(k \times k)$ square matrix, has $\text{rank}(\mathbf{A}) = k$, then \mathbf{A} is invertible.

43

Linear Algebra: Linear Dependence and Rank

Examples:

$$\begin{aligned} (1) \quad & v'_1 = [5 \quad 12] \\ & v'_2 = [10 \quad 24] \\ & \mathbf{A} = \begin{bmatrix} 5 & 10 \\ 12 & 24 \end{bmatrix} = \begin{bmatrix} v'_1 \\ v'_2 \end{bmatrix} \quad (\text{a } 2 \times 2 \text{ matrix}) \\ & 2v'_1 - v'_2 = \mathbf{0} \quad \Rightarrow \text{rank}(\mathbf{A}) = 1 \quad \Rightarrow \text{cannot invert } \mathbf{A} \end{aligned}$$

$$\begin{aligned} (2) \quad & v_1 = \begin{bmatrix} 2 \\ 7 \end{bmatrix}; v_2 = \begin{bmatrix} 1 \\ 8 \end{bmatrix}; v_3 = \begin{bmatrix} 4 \\ 5 \end{bmatrix}; \\ & \mathbf{A} = \begin{bmatrix} 2 & 1 & 4 \\ 7 & 8 & 5 \end{bmatrix} \\ & 3v'_1 - 2v'_2 = [6 \quad 21] - [2 \quad 16] \\ & \quad \quad \quad = [4 \quad 5] = v'_3 \\ & 3v'_1 - 2v'_2 - v'_3 = \mathbf{0} \quad \Rightarrow \text{rank}(\mathbf{A}) = 2 \end{aligned}$$

44

Linear Algebra: Rules for Vector Derivatives

(1) Linear function: $\mathbf{y} = f(\mathbf{z}) = \omega + \mathbf{z}' \boldsymbol{\gamma} = \omega + z_1 \gamma_1 + \dots + z_k \gamma_k$
 where \mathbf{z} and $\boldsymbol{\gamma}$ are k -dimensional vectors and ω is a constant.

• Then,
$$\nabla f(\mathbf{z}) = \begin{bmatrix} \frac{\partial f(\mathbf{z})}{\partial z_1} \\ \vdots \\ \frac{\partial f(\mathbf{z})}{\partial z_k} \end{bmatrix} = \begin{bmatrix} \gamma_1 \\ \vdots \\ \gamma_k \end{bmatrix} = \boldsymbol{\gamma} \quad (k \times 1 \text{ vector})$$

(2) Quadratic form: $\mathbf{q} = f(\mathbf{z}) = \mathbf{z}' \mathbf{A} \mathbf{z}$ (a scalar)
 where \mathbf{z} is $k \times 1$ vector and \mathbf{A} is a $k \times k$ matrix.

• Then,
$$\nabla f(\mathbf{z}) = \mathbf{A}' \mathbf{z} + \mathbf{A} \mathbf{z}$$

If \mathbf{A} is symmetric, then $\nabla f(\mathbf{z}) = (\mathbf{A}' + \mathbf{A}) \mathbf{z} = 2 \mathbf{A} \mathbf{z}$ ($k \times 1$ vector)

45

Least Squares Estimation with Linear Algebra

• Let's assume a linear system with k independent variables and T observations. That is,

$$y_i = \beta_1 x_{1i} + \beta_2 x_{2i} + \dots + \beta_k x_{ki} + \varepsilon_i, \quad i = 1, 2, \dots, T$$

The whole system (for all i) is:

$$\begin{aligned} y_1 &= \beta_1 x_{11} + \beta_2 x_{21} + \dots + \beta_k x_{k1} + \varepsilon_1 \\ y_2 &= \beta_1 x_{12} + \beta_2 x_{22} + \dots + \beta_k x_{k2} + \varepsilon_2 \\ &\dots \quad \dots \quad \dots \quad \dots \\ y_T &= \beta_1 x_{1T} + \beta_2 x_{2T} + \dots + \beta_k x_{kT} + \varepsilon_T \end{aligned}$$

Using linear algebra we can rewrite the system as:

$$\mathbf{y} = \mathbf{X} \boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

Least Squares Estimation with Linear Algebra

- Using linear algebra notation: $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$

Vectors will be column vectors: \mathbf{y} , \mathbf{x}_j , and $\boldsymbol{\varepsilon}$ are $T \times 1$ vectors:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_T \end{bmatrix} \Rightarrow \mathbf{y}' = [y_1 \ y_2 \ \dots \ y_T]$$

$$\mathbf{x}_j = \begin{bmatrix} x_{j1} \\ \vdots \\ x_{jT} \end{bmatrix} \Rightarrow \mathbf{x}_j' = [x_{j1} \ x_{j2} \ \dots \ x_{jT}]$$

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_T \end{bmatrix} \Rightarrow \boldsymbol{\varepsilon}' = [\varepsilon_1 \ \varepsilon_2 \ \dots \ \varepsilon_T]$$

$$\mathbf{X} \text{ is a } T \times k \text{ matrix.} \Rightarrow \mathbf{X} = [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_k]$$

Least Squares Estimation with Linear Algebra

- Assume $f(\mathbf{X}, \theta)$ is linear: $f(\mathbf{X}, \theta) = \mathbf{X}\boldsymbol{\beta}$

- Objective function:
$$\begin{aligned} S(\mathbf{x}_i, \boldsymbol{\beta}) &= \sum_i^T \varepsilon_i^2 = \boldsymbol{\varepsilon}'\boldsymbol{\varepsilon} = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})' (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \\ &= (\mathbf{y}' - \boldsymbol{\beta}'\mathbf{X}') (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \\ &= \mathbf{y}'\mathbf{y} - \mathbf{y}'\mathbf{X}\boldsymbol{\beta} - \boldsymbol{\beta}'\mathbf{X}'\mathbf{y} + \boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta} \\ &= \mathbf{y}'\mathbf{y} - 2 \boldsymbol{\beta}'\mathbf{X}'\mathbf{y} + \boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta} \\ &= (\mathbf{c} - 2 \boldsymbol{\beta}'\mathbf{d} + \boldsymbol{\beta}'\mathbf{A}\boldsymbol{\beta}) \end{aligned}$$

- 1st derivative w.r.t. $\boldsymbol{\beta}$:
$$\begin{aligned} \frac{\partial S(\mathbf{x}_i, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}'} &= (-2 \mathbf{d} + 2 \mathbf{A} \boldsymbol{\beta}) \\ &= -2 \mathbf{X}'\mathbf{y} + 2 \mathbf{X}'\mathbf{X} \boldsymbol{\beta} \quad (k \times 1 \text{ vector}) \end{aligned}$$

- F.o.c. (normal equations): $\mathbf{X}'\mathbf{y} - (\mathbf{X}'\mathbf{X}) \boldsymbol{\beta} = \mathbf{0} \Rightarrow (\mathbf{X}'\mathbf{X}) \boldsymbol{\beta} = \mathbf{X}'\mathbf{y}$

Least Squares Estimation with Linear Algebra

- F.o.c. (normal equations): $\mathbf{X}'\mathbf{y} - (\mathbf{X}'\mathbf{X})\mathbf{b} = \mathbf{0} \Rightarrow (\mathbf{X}'\mathbf{X})\mathbf{b} = \mathbf{X}'\mathbf{y}$
- Assuming $(\mathbf{X}'\mathbf{X})$ is non-singular –i.e., invertible–, we solve for \mathbf{b} :

$$\Rightarrow \mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'\mathbf{y} \quad (\text{a } k \times 1 \text{ vector})$$

Notes: 1. \mathbf{b} is called the Ordinary Least Squares (**OLS**) estimator.
 2. We can use the determinant to check if $\mathbf{X}'\mathbf{X}$ is non-singular.

Remark: Technically, we still need to check the Second Order condition, we need the 2nd derivative to be positive for a minimum:

$$\frac{\partial^2 S(x_i, \beta)}{\partial \beta \partial \beta'} = 2 \mathbf{X}'\mathbf{X},$$

which is a **positive definite (pd)** matrix, the counterpart to positive numbers for matrices.

$\Rightarrow \mathbf{b}$ is a minimum!

Least Squares Estimation with Linear Algebra

- \mathbf{X} is a $T \times k$ matrix. Its columns are the k $T \times 1$ vectors \mathbf{x}_k . It is common to treat \mathbf{x}_1 as vector of ones:

$$\mathbf{x}_1 = \begin{bmatrix} x_{11} \\ \vdots \\ x_{1T} \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \Rightarrow \mathbf{x}_1' = [1 \ 1 \ \dots \ 1] = \mathbf{i}'$$

This vector of ones represent the usual constant in the model. Then,

$$\mathbf{X} = \begin{bmatrix} 1 & x_{21} & \cdots & x_{k1} \\ 1 & x_{22} & \cdots & x_{k1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{2T} & \cdots & x_{kT} \end{bmatrix}$$

OLS Estimation – Example in R: IBM returns

Example: CAPM Model for IBM monthly returns:

```
SFX_da <- read.csv("http://www.bauer.uh.edu/rsusmel/4397/Stocks_FX_1973.csv",
head=TRUE, sep=",")

x_ibm <- SFX_da$IBM
x_Mkt_RF <- SFX_da$Mkt_RF      # Market (CRSP) excess returns (in %)
x_RF <- SFX_da$RF              # Risk-free rate (in %)

T <- length(x_ibm)             # Data size
lr_ibm <- log(x_ibm[-1]/x_ibm[-T])
Mkt_RF <- x_Mkt_RF[-1]/100
RF <- x_RF[-1]/100

ibm_x <- lr_ibm - RF           # IBM Excess returns
T <- length(ibm_x)             # Data size adjusted by 1 observation
x0 <- matrix(1,T,1)           # vector of 1s (Tx1)
x <- cbind(x0, Mkt_RF)         # Matrix X (Tx2)
```

OLS Estimation – Example in R: IBM returns

Example (continuation): CAPM Model for IBM returns:

```
b <- solve(t(x)%*% x)%*% t(x)%*% y      #  $\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$  (OLS regression)
> b
      [1]
-0.005791039
Mkt_RF 0.895773564
```

Note: We got these coefficient before, using the `lm()` function:

```
fit_ibm_capm <- lm(x_ibm ~ Mkt_RF)
```