Lecture 3-b
Least Squares – Review of Linear Algebra

Brooks (4^{th} edition): Chapter 3

Review – First Four Classes: Summary

- Defined a random variable (say, stock returns) and how to describe its distribution with moments (mean, variance, skewness, etc).

- Testing hypothesis (H_0) about the behavior of the RV. For example, the mean return of the S&P 500 is 0 (H_0: μ=0).

- Building Confidence Intervals: Assuming a distribution or a bootstrap. For example, compute a C.I. for Transaction Exposure in FX Markets and compute a VaR(1 - α).

- Basic R and simple applications.

Modeling a Dependent Variable as a Function

• So far, we focused on one RV only, say stock returns and learning about its distribution, for example, using descriptive statistics. In econometrics, we usually care about a functional relation between y, the *dependent variable,* and $\boldsymbol{x}$, a set (a vector!) of *explanatory variables.*

• In this lecture, we will **linearly** relate y to x & an *error term*, ε:

$$y_i = \alpha + \beta\, x_i + \varepsilon_i, \qquad i = 1, 2, ..., N$$

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

• We call the above equation the **Data Generating Process** (DGP).

Review – Linear Model: One Variable Case

• **DGP**: (linear): $y_i \;=\; \alpha + \beta\, x_i + \varepsilon_i, \qquad i = 1, 2, ..., N.$

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

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

Example: The CAPM posits a relation between the excess return of asset i, $y_i = r_{i,t} - r_f$, and the excess return of the market, $x_i = r_{m,t} - r_f$. In equilibrium, the CAPM states:

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

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

CAPM DGP: $y_i \;=\; \alpha + \beta\, x_i + \varepsilon_i, \qquad i = 1, 2, ..., N.$

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 example, we have that i's excess returns are only explained by the market. But, we could have used more variables, for example the 3 factors in the standard Fama-French model: Market, SMB (size factor), and HML (book-to-market).

The 3-factor FF model represents a **multivariate model** for IBM returns:

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

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

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

Review – Least Squares Estimation

• We relate a dependent variable y to a set of k explanatory variables $\boldsymbol{x}$. This function depends on unknown parameters, $\boldsymbol{\theta}$, which we want to estimate. The relation between y and $\boldsymbol{x}$ is not exact; there is an error, ε. We have T observations of y and $\boldsymbol{x}$.

$$y_i = f(\, x_{1,i}, x_{2,i}, \ldots, x_{k,i}; \boldsymbol{\theta}) + \varepsilon_i, \qquad i = 1, 2, ..., T.$$

• The functional form, $f(\boldsymbol{x}_i, \boldsymbol{\theta})$, is dictated by theory or experience. In this class, we mainly work with the **linear** case:

$$f(\boldsymbol{x}_i, \boldsymbol{\beta}) = \beta_1\ x_{1,i} + \beta_2\ x_{2,i} + \beta_3\, x_{3,i} + \ldots + \beta_k\ x_{k,i}.$$

• Now, we estimate the vector $\boldsymbol{\beta} = \{\beta_1, \beta_2, \ldots, \beta_k\}$ by minimizing

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

We call this estimator the *Ordinary Least Squares* (OLS) estimator.

Review – OLS for One Variable: Derivation

- LS estimation can be applied to any functional form. In this class, we use a **linear function**. In this lecture, we derive the OLS formulas for the simplest case: **one explanatory variable**. Then,

$$y_i = \beta_1 + \beta_2 x_i + \varepsilon_i \qquad \text{(two parameters } \Rightarrow k = 2\text{)}$$

Objective function:

$$S(\boldsymbol{x}; \beta_1, \beta_2) = \sum_i^T {\varepsilon_i}^2 = \sum_i^T (y_i - \beta_1 - \beta_2\, x_i)^2 =$$
$$= \{y_1 - \beta_1 - \beta_2\, x_1)^2 + \{y_2 - \beta_1 - \beta_2\, x_2)^2 + \ldots + \{y_T - \beta_1 - \beta_2\, x_T)^2$$

Taking first derivatives with respect to β_1 & β_2:

(β_1): $2 \sum_i^T (y_i - \beta_1 - \beta_2\, x_i)$ (-1) $\quad \Rightarrow -2 \sum_i^T (y_i - \beta_1 - \beta_2\, x_i)$

(β_2): $2 \sum_i^T (y_i - \beta_1 - \beta_2\, x_i)$ ($-x_i$) $\quad \Rightarrow -2 \sum_i^T (y_i\, x_i - \beta_1\, x_i - \beta_2 x_i^2)$

Review – OLS & One Variable Case: Derivation

- Now, we set f.o.c.'s

(β_1): $\sum_i^T (y_i - b_1 - b_2\, x_i) = 0 \qquad (1)$

(β_2): $\sum_i^T (y_i\, x_i - b_1\, x_i - b_2\, x_i^2) = 0 \qquad (2)$

Since we have $k = 2$, the f.o.c.'s form a 2x2 system of equations, the *normal equations.*

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

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

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

From (2): $\sum_i^T y_i x_i - (\bar{y} - b_2\, \bar{x}) \sum_i^T x_i - b_2 \sum_i^T x_i^2 = 0 = 0$

$\Rightarrow \sum_i^T y_i x_i - \bar{y} \sum_i^T x_i - b_2 \left(\sum_i^T x_i^2 - \bar{x} \sum_i^T x_i\right) = 0$

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

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 (= α_i).

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

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

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

Review – OLS & One Variable Case: CAPM

- Conditional Prediction

Suppose analysts estimate that Market excess returns are **10%,** then, we estimate (or predict, given the **10%** value for Market excess returns):

Predicted [IBM excess returns | $(r_{m,t} - r_f)$=**.10**] = $b_1 + b_2$ * **.10**.

We will call the Predicted $y_i = \hat{y}_i$ = 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 *
xMkt_RF      0.895774   0.053867  16.629   <2e-16 ***
---
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
```

b_1 (points to -0.005791)

$b_2 = \hat{\beta}_{CAPM}$ (points to 0.895774)

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:

b_2 = slope. If market excess returns increase by 1%, IBM excess returns increase by 0.90%. The estimate of β_{IBM} (β<1) implies that IBM is less volatile ("safer") than the market.

Conditional prediction of IBM excess returns:

Suppose market excess returns increase are 10%, then we predict IBM excess returns = **-0.005791** + **0.895774** * .10 = 0.08378 (**8.38%**).

Note: According to the CAPM, IBM underperformed:

- IBM excess returns (CAPM) = **0.895774** * mean(Mkt_RF)

  = **0.895774** * 0.0056489 = **0.0050601**

- IBM excess returns (sample) = mean(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:

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

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. and it 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 $f(x_i, \theta)$ is **linear**. We assume a linear system with k independent variables and T observations. That is,

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

The whole system (for all i) is:

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

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:

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

Notation: **y**, **β** & **ε** are vectors:

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

X is a matrix:

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

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

columns

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

• 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 ⇒ matrices
Lower case ⇒ vectors

17

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 Txk matrix, **X**, over time we build a database:

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

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

18

Linear Algebra: Matrices in Econometrics

- We want to estimate a model: $y = f(x_1, x_2, ..., x_k)$. We collect data, T (or N) observations, on a dependent variable, $\boldsymbol{y}$, and on k explanatory variables, **X**.

- Usual notation: vectors are column vectors: **y** & $\boldsymbol{x_j}$ are Tx1 vectors:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_T \end{bmatrix} \qquad \& \qquad \boldsymbol{x_j} = \begin{bmatrix} x_{j1} \\ x_{j2} \\ \vdots \\ x_{jT} \end{bmatrix} \qquad j = 1, ..., k$$

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

Its columns are the k Tx1 vectors $\boldsymbol{x_j}$. It is common to treat $\boldsymbol{x_1}$ as vector of ones, $\boldsymbol{\iota}$.

19

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“
[...]
```

20

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
```

21

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":

$$\mathbf{A} * \mathbf{I} = \mathbf{A} \qquad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

$$\mathbf{A} * \mathbf{0} = \mathbf{0} \qquad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

- Both are *diagonal* matrices ⇒ off-diagonal elements are zero.

- Both are examples of *symmetric* and *idempotent* matrices.
  - Symmetric: $\mathbf{A} = \mathbf{A}^T$
  - Idempotent: $\mathbf{A} = \mathbf{A}^2 = \mathbf{A}^3 = \ldots$

22

Linear Algebra: Multiplication

- We want to multiply two matrices: **A*****B**. But, multiplication of matrices requires a *conformability condition.*

- Conformability condition: The column dimensions of the lead matrix **A** must be equal to the row dimension of the lag matrix **B**.

- If **A** is an $(m$x$n)$ and **B** an $(n$x$p)$ matrix (**A** has the same number of columns as **B** has rows), then we define the product of **AB**:
AB = **C** is $(m$x$p)$ matrix with its ik^{th} element is $c_{ik} = \sum_{j=1}^{n} a_{ij}\, b_{jk}$.

- What are the dimensions of the vector, matrix, and result?

$$\boldsymbol{a}\mathbf{B} = [a_{11} a_{12}] \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix} = \boldsymbol{c} = [c_{11} \quad c_{12} \quad c_{13}]$$

$$= [a_{11}b_{11} + a_{12}b_{21} \quad a_{11}b_{12} + a_{12}b_{22} \quad a_{11}b_{13} + a_{12}b_{23}]$$

- Dimensions: a(1x2), B(2x3) ⇒ c(1x3)

23

Linear Algebra: Multiplication

Example: We want to multiply **A** (2x2) and **B** (2x2), where **A** has elements a_{ij} and **B** has elements b_{jk}. Recall the ik^{th} element is

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

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

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

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

$$C_{2x2} = A_{2x2} * \ B_{2x2}$$

- Dimensions: A(2x2), B(2x2) ⇒ C(2x2), a square matrix.

24

Linear Algebra: Multiplication

Example: We want to multiply **X** (2x2) and **β** (2x1), where **X** has elements x_{ij} and **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

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

Recall the i^{th} element is

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

Then,

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

- Dimensions: **X**(2x2), **β**(2x1) ⇒ $\boldsymbol{y}$(2x1), a row vector.

25

Linear Algebra: Transpose

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

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 **A** is a $m \times n$ matrix ⇒ $\mathbf{A}^{\mathrm{T}}$ is a $n \times m$ matrix.
- (**A'**)' = **A**
- Conformability changes unless the matrix is square.
- (**AB**)' = **B'A'**

26

Linear Algebra: Transpose – Example (X')

- In econometrics, an important matrix is **X'X**. Recall **X** (usually, the matrix of k independent explanatory variables):

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

Then,

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

27

Linear Algebra: Math Operations

- Addition, Subtraction, Multiplication

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

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

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

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

28

Linear Algebra: Math Operations – Examples

- Matrix addition

$$\begin{bmatrix} 2 & 1 \\ 7 & 9 \end{bmatrix} + \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 5 & 2 \\ 7 & 11 \end{bmatrix}$$
$$A_{2x2} + B_{2x2} = C_{2x2}$$

- 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

$$\begin{bmatrix} 2 & 1 \\ 7 & 9 \end{bmatrix} x \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 4 & 3 \\ 25 & 27 \end{bmatrix}$$
$$A_{2x2} \text{ x } B_{2x2} = C_{2x2}$$

- Scalar multiplication

$$\frac{1}{8}\begin{bmatrix} 2 & 4 \\ 6 & 1 \end{bmatrix} = \begin{bmatrix} 1/4 & 1/2 \\ 3/4 & 1/8 \end{bmatrix}$$

29

29

Linear Algebra: Math Operations – ε'ε

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

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

Since **ε** is Tx1 vector, we use linear algebra to write the sum of squares of its elements as (dot product of 2 Tx1 vectors):

$$S(x_i, \beta) = \sum_{i=1}^{T} \varepsilon_i^2 = \boldsymbol{\varepsilon}' \boldsymbol{\varepsilon}$$

Check:

$$\boldsymbol{\varepsilon}' \boldsymbol{\varepsilon} = [\varepsilon_1\ \varepsilon_2\ ....\ \varepsilon_T] * \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_T \end{bmatrix} = [\varepsilon_1^2 + \varepsilon_2^2 + ... + \varepsilon_T^2] = \sum_{i=1}^{T} \varepsilon_i^2$$

Thus, we pick **β** to minimize:

$$S(x_i, \beta) = \boldsymbol{\varepsilon}'\boldsymbol{\varepsilon} = (\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})' (\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})$$

Linear Algebra: Math Operations – X' X

- A special matrix in econometrics, **X'X** (a kxk matrix, with k = 2):

- $\boldsymbol{X}\,(T\text{x}2) = \begin{bmatrix} x_{11} & x_{21} \\ x_{12} & x_{22} \\ \vdots & \vdots \\ x_{1T} & x_{2T} \end{bmatrix}$

- $\boldsymbol{X}'\,(2\text{x}T) = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1T} \\ x_{21} & x_{22} & \cdots & x_{2T} \end{bmatrix}$

Sum of squares of x_i (i = 1, 2)

$$\boldsymbol{X}'\boldsymbol{X}\,(2\text{x}2) = \begin{bmatrix} \sum_{i=1}^{T} x_{1i}^2 & \sum_{i=1}^{T} x_{2i}x_{1i} \\ \sum_{i=1}^{T} x_{1i}x_{2i} & \sum_{i=1}^{T} x_{2i}^2 \end{bmatrix} = \sum_{i=1}^{T} \begin{bmatrix} x_{1i}^2 & x_{2i}x_{1i} \\ x_{2i}x_{1i} & x_{2i}^2 \end{bmatrix}$$

$$= \sum_{i=1}^{T} \begin{bmatrix} x_{1i} \\ x_{2i} \end{bmatrix} \begin{bmatrix} x_{1i} & x_{2i} \end{bmatrix}$$

Sum of cross products of $x_1\, x_2$

$$= \sum_{i=1}^{T} \boldsymbol{x}_i \boldsymbol{x}_i{}'$$

31

Linear Algebra: Math Operations – X' X

- In general, with a kxk **X'X** matrix:

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

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

$$= \sum_{i=1}^{T} \begin{bmatrix} x_{1i}^2 & \cdots & x_{1i}x_{ki} \\ \vdots & \ddots & \vdots \\ x_{ki}x_{1i} & \cdots & x_{ki}^2 \end{bmatrix}$$

(a symmetric matrix)

32

Linear Algebra: Inverse of a Matrix

• Identity matrix: **AI** = **A**, where $I_j = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$

Notation: $\boldsymbol{I}_j$ is a jxj identity matrix.

• Given **A** (mxn), the matrix **B** (nxm) is a *right-inverse* for **A** iff $\mathbf{AB}=\mathbf{I}_m$

• Given **A** (mxn), the matrix **C** (nxm) is a *left-inverse* for **A** iff $\mathbf{CA} = \mathbf{I}_n$

• **Theorem**: If **A** (mxn), has both a *right-inverse* **B** and a *left-inverse* **C**, then $\mathbf{C} = \mathbf{B} = \mathbf{A}^{-1}$

Note:

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

33

Linear Algebra: Symmetric Matrices

Definition:

If **A'** = **A**, then **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 **B** if **A** is symmetric matrix (& **B** is square root of **A**.) This is not possible in general.

• **X'X** is symmetric. It plays a very important role in econometrics.

34

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
```

35

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                             # a (2x3) matrix
   [,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                             # a (3x2) matrix
     v3 v4
[1,]  1  1
[2,]  3  2
[3,]  1  0
```

36

Linear Algebra: Operations in R

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

- Matrix multiplication: **%*%**

```
C <- A%*%B                    #A is 2x3; B is 3x2   ⇒ 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.

37

Linear Algebra: Operations in R

- Dot product "•" is a function that takes pairs of vectors and produces a number. For vectors **c** & **z**, it is defined as:

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

- Dot product with 2 vectors: v1 • 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: v1 • v1 produces a sum of the square elements of vector

```
> t(v1) %*% v1
     [,1]
[1,]   74
```

38

Linear Algebra: Operations in R

- Dot product with $\imath$ (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]
[1,]   12
```

- Product of 2 vectors: v1 & t(v2): 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
```

<u>Property of dot product</u>: If the dot product of two vectors is equal to zero, then the vectors are *orthogonal* (perpendicular *or* "⊥") vectors. We interpret this as "the vectors are *uncorrelated*."

39

Linear Algebra: Operations in R

- Matrix transpose: **t**

```
> t(B)                     #B is 3x2  => t(B) is 2x3          > B
     [,1] [,2] [,3]                                              v3 v4
[1,]    1    3    1                                          [1,]  1  1
[2,]    1    2    0                                          [2,]  3  2
                                                             [3,]  1  0
```

- **X'X** (a symmetric matrix)

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

- Determinant: **det**

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

40

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,] -1.1666667  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
```

41

Linear Algebra: Example 1 – Linear DGP

- There is a functional form relating a dependent variable, y, and k explanatory variables, **X**. The functional form is linear, but it depends on k unknown parameters, $\boldsymbol{\beta}$. The relation between y and **X** is not exact. There is an error, $\boldsymbol{\varepsilon}$. We have T observations of y and **X**.

- Then, the data is generated according to:

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

Or

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

Or using matrix notation:

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

42

Linear Algebra: Example 1 – Linear DGP

- Model: $\boldsymbol{y} = \mathbf{X}\,\boldsymbol{\beta} + \boldsymbol{\varepsilon}$

where $\boldsymbol{y}$ & $\boldsymbol{\varepsilon}$ are (Tx1); $\mathbf{X}$ is (Txk); and $\boldsymbol{\beta}$ is (kx1).

- We call this relation *data generating process* (DGP).

- Our goal this lecture: Estimate the unknown vector $\boldsymbol{\beta}$.

43

Linear Algebra: Example 2 – Linear System

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

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

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

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

$\boldsymbol{d}$ = (mx1) column vector

$\mathbf{A}$ = (mxn) matrix

$\mathbf{x}$ = (nx1) column vector

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

44

Linear Algebra: Example 2 – Linear System

- System of linear equations: **Ax** = $\boldsymbol{d}$

where
A = (mxn) matrix of parameters
x = column vector of endogenous variables (nx1)
$\boldsymbol{d}$ = column vector of exogenous variables and constants (mx1)

- Solve for **x***

- **Theorem**: Given **A** (mxn) invertible. Then, the equation **Ax** = $\boldsymbol{d}$ has one and only one solution for every $\boldsymbol{d}$ (mx1). That is, there is a unique x*.

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

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
```

45

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 $\{\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_k\}$ is called a *linearly independent* set of vectors iff

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

Notes:

- Dependence prevents solving a system of equations (**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 (rank(**A**)).

- If **A**, a (k x k) square matrix, has rank(**A**)= k, then **A** is invertible. 46

Linear Algebra: Linear Dependence and Rank

Examples:

(1) $v_1' = [5 \quad 12]$

$v_2' = [10 \quad 24]$

$\boldsymbol{A} = \begin{bmatrix} 5 & 10 \\ 12 & 24 \end{bmatrix} = \begin{bmatrix} v_1' \\ v_2' \end{bmatrix}$ (a 2x2 matrix)

$2v_1' - v_2' = \mathbf{0} \quad \Rightarrow rank(\boldsymbol{A}) = 1 \quad \Rightarrow$ cannot invert **A**

(2) $v_1 = \begin{bmatrix} 2 \\ 7 \end{bmatrix}; v_2 = \begin{bmatrix} 1 \\ 8 \end{bmatrix}; v_3 = \begin{bmatrix} 4 \\ 5 \end{bmatrix};$

$A = \begin{bmatrix} 2 & 1 & 4 \\ 7 & 8 & 5 \end{bmatrix}$

$3{v_1}' - 2{v_2}' = [6 \quad 21] - [2 \quad 16]$

$= [4 \quad 5] = {v_3}'$

$3v_1' - 2v_2' - {v_3}' = \mathbf{0} \quad \Rightarrow rank(\boldsymbol{A}) = 2$

47

Linear Algebra: Rules for Vector Derivatives

(1) Linear function: $\boldsymbol{y} = f(\boldsymbol{x}) = \boldsymbol{x}' \boldsymbol{\gamma} + \omega$

where $\boldsymbol{x}$ and $\boldsymbol{\gamma}$ are k-dimensional vectors and ω is a constant.

- We derive the gradient in matrix notation as follows:

1. Convert to summation notation: $f(\boldsymbol{x}) = \sum_i^k x_i \gamma_i$

2. Take partial derivative w.r.t. x_j: $\frac{\partial}{\partial x_j}\left[\sum_i^k x_i \gamma_i\right] = \gamma_j$

3. Put all the partial derivatives in a vector:

$$\nabla f(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \vdots \\ \frac{\partial f(x)}{\partial x_k} \end{bmatrix} = \begin{bmatrix} \gamma_1 \\ \vdots \\ \gamma_k \end{bmatrix}$$

4. Convert to matrix notation: $\nabla f(\boldsymbol{x}) = \gamma$

48

Linear Algebra: Rules for Vector Derivatives

(2) Quadratic form: $q = f(\boldsymbol{x}) = \boldsymbol{x}' \mathbf{A} \boldsymbol{x}$

where $\boldsymbol{x}$ is kx1 vector and **A** is a kxk matrix, with a_{ji} elements.

- Convert $\boldsymbol{x}' \mathbf{A} \boldsymbol{x}$ to summation notation:

$$f(\boldsymbol{x}) = \boldsymbol{x}' \begin{bmatrix} \sum_j^k a_{j1} x_j \\ \vdots \\ \sum_j^k a_{jk} x_j \end{bmatrix} = \sum_i^k \sum_j^k x_i \, a_{ji} \, x_j$$

- After taking derivatives and some algebra:

$$\nabla f(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \vdots \\ \frac{\partial f(x)}{\partial x_k} \end{bmatrix} = \begin{bmatrix} \sum_i^k x_i \, a_{i1} \\ \vdots \\ \sum_i^k x_i \, a_{ik} \end{bmatrix} + \begin{bmatrix} \sum_i^k a_{1i} \, x_i \\ \vdots \\ \sum_i^k a_{ki} \, x_i \end{bmatrix} = \mathbf{A}' \boldsymbol{x} + \mathbf{A} \boldsymbol{x}$$

If **A** is symmetric, then $\nabla f(\boldsymbol{x}) = (\mathbf{A}' + \mathbf{A}) \, \boldsymbol{x} = 2 \, \mathbf{A} \, \boldsymbol{x}$

49

Least Squares Estimation with Linear Algebra

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

$$y_i = \beta_1 \, x_{1i} + \beta_2 \, x_{2i} + \ldots + \beta_k \, x_{ki} + \varepsilon_i, \qquad i = 1, 2, \ldots., T$$

The whole system (for all i) is:

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

Using linear algebra we can rewrite the system as:

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

Least Squares Estimation with Linear Algebra

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

Vectors will be column vectors: $\mathbf{y}$, $\mathbf{x}_j$, and $\boldsymbol{\varepsilon}$ are Tx1 vectors:

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_T \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{y}' = [y_1\ y_2\ ....\ y_T]$$

$$\boldsymbol{x}_j = \begin{bmatrix} x_{j1} \\ \vdots \\ x_{jT} \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{x}_j' = [x_{j1}\ x_{j2}\ ....\ x_{jT}]$$

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

$\mathbf{X}$ is a Txk matrix. $\Rightarrow$ $\mathbf{X} = [\boldsymbol{x}_1\ \boldsymbol{x}_2\ ....\ \boldsymbol{x}_k]$

Least Squares Estimation with Linear Algebra

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

$$\begin{aligned} S(x_i, \boldsymbol{\beta}) &= \Sigma_i\, \varepsilon_i^2 = \boldsymbol{\varepsilon}'\boldsymbol{\varepsilon} = (\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})'\,(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}) \\ &= \boldsymbol{y}'\boldsymbol{y} - \boldsymbol{y}'\mathbf{X}\boldsymbol{\beta} - \boldsymbol{\beta}'\mathbf{X}'\boldsymbol{y} + \boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta} \\ &= \boldsymbol{y}'\boldsymbol{y} - 2\,\boldsymbol{\beta}'\mathbf{X}'\mathbf{y} + \boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta} \end{aligned}$$

- First derivative w.r.t. $\boldsymbol{\beta}$: $-2\,\mathbf{X}'\boldsymbol{y} + 2\,\mathbf{X}'\mathbf{X}\,\boldsymbol{\beta}$ (a kx1 vector)
- F.o.c. (normal equations): $\mathbf{X}'\boldsymbol{y} - (\mathbf{X}'\mathbf{X})\,\mathbf{b} = \mathbf{0}$ $\Rightarrow (\mathbf{X}'\mathbf{X})\,\mathbf{b} = \mathbf{X}'\boldsymbol{y}$
- Assuming $(\mathbf{X}'\mathbf{X})$ is non-singular –i.e., invertible-, we solve for $\mathbf{b}$:

$$\Rightarrow \mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1}\,\mathbf{X}'\boldsymbol{y} \qquad \text{(a } k\text{x1 vector)}$$

<u>Note</u>: **b** is called the Ordinary Least Squares (OLS) estimator.
(*Ordinary* = $f(\mathbf{X}, \theta)$ is linear)

Least Squares Estimation with Linear Algebra

- $\mathbf{X}$ is a Txk matrix. Its columns are the k Tx1 vectors $\boldsymbol{x}_k$. It is common to treat $\mathbf{x}_1$ as vector of ones:

$$\boldsymbol{x}_1 = \begin{bmatrix} x_{11} \\ \vdots \\ x_{1T} \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{x}_1' = [1\ 1\ ....\ 1] = \boldsymbol{\imath}'$$

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

Note: Recall the dot product: Post-multiplying a vector (1xT) $\boldsymbol{x}_k$ by $\boldsymbol{\imath}$ (or $\boldsymbol{\imath}'\ \boldsymbol{x}_k$) produces a scalar, the sum of all the elements of vector $\boldsymbol{x}_k$:

$$\boldsymbol{x}_k'\ \boldsymbol{\imath} = \boldsymbol{\imath}'\ \boldsymbol{x}_k = x_{k1} + x_{k2} + ... + x_{kT} = \sum_i^T x_{ki}$$

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                 # b = (X'X)^-1 X' 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)
```