Linear Regression

Berlin Chen
Department of Computer Science & Information Engineering
National Taiwan Normal University

Reference:

1. *Applied Numerical Methods with MATLAB for Engineers*, Chapter 14 & Teaching material

Chapter Objectives (1/2)

- Familiarizing yourself with some basic descriptive statistics and the normal distribution
- Knowing how to compute the slope and intercept of a best fit straight line with linear regression
- Knowing how to compute and understand the meaning of the coefficient of determination and the standard error of the estimate

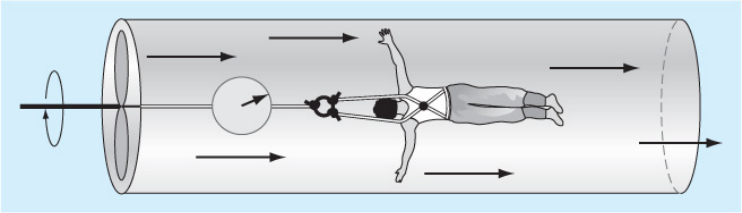


FIGURE 14.1
Wind tunnel experiment to measure how the force of air resistance depends on velocity.

$$F = cv\ ?$$

$$F = cv^2\ ?$$

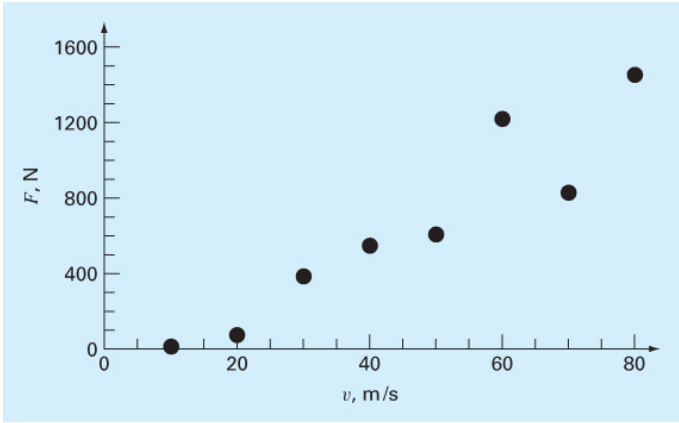


FIGURE 14.2
Plot of force versus wind velocity for an object suspended in a wind tunnel.

Chapter Objectives (2/2)

- Understanding how to use transformations to linearize nonlinear equations so that they can be fit with linear regression
- Knowing how to implement linear regression with MATLAB

Statistics Review: Measures of Location

- **Arithmetic mean**: the sum of the individual data points (y_i) divided by the number of points n:

$$\bar{y} = \frac{\sum y_i}{n}$$

- **Median**: the midpoint of a group of data

- **Mode**: the value that occurs most frequently in a group of data

Statistics Review: Measures of Spread

- **Standard deviation**:

$$s_y = \sqrt{\frac{S_t}{n-1}}$$

  - where S_t is the sum of the squares of the data residuals:

$$S_t = \sum (y_i - \bar{y})^2$$

  - and n-1 is referred to as the *degrees of freedom*

- **Variance**:

$$s_y^2 = \frac{\sum (y_i - \bar{y})^2}{n-1} = \frac{\sum y_i^2 - \left(\sum y_i\right)^2 / n}{n-1}$$

- **Coefficient of variation (c.v.)**:

$$\text{c.v.} = \frac{s_y}{\bar{y}} \times 100\%$$

Histogram

- A histogram provides a simple visual representation of the data
- A histogram is constructed by sorting the measurements into intervals, or bin
- In a histogram, the units of measurement are plotted on abscissa and the frequency of occurrence of each interval is plotted on the ordinate

Normal Distribution

FIGURE 14.3
A histogram used to depict the distribution of data. As the number of data points increases, the histogram often approaches the smooth, bell-shaped curve called the normal distribution.

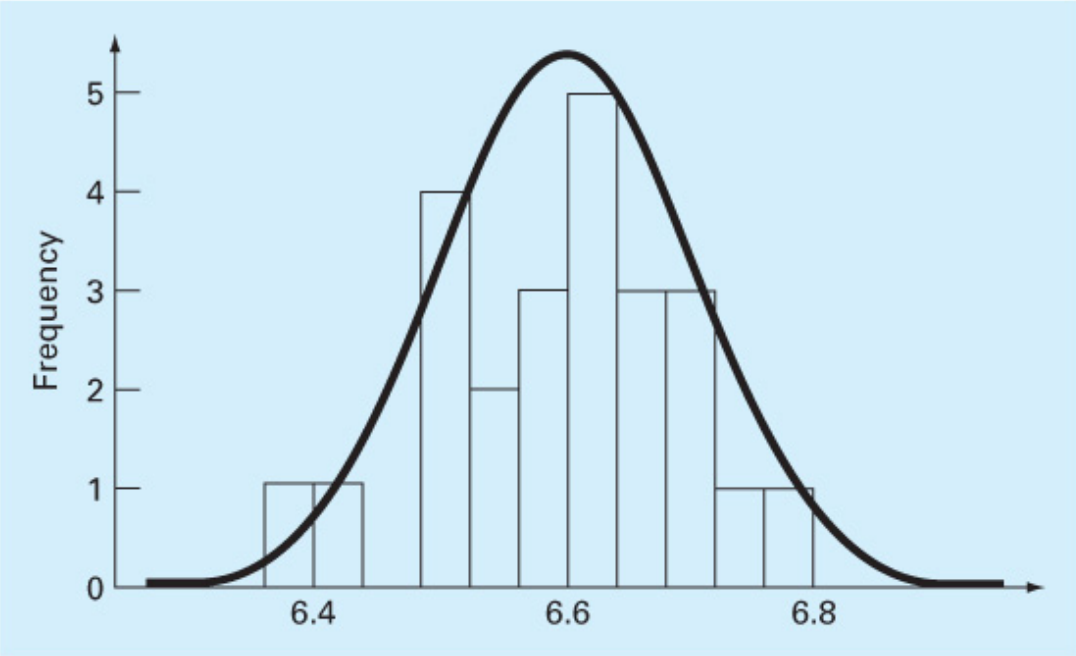
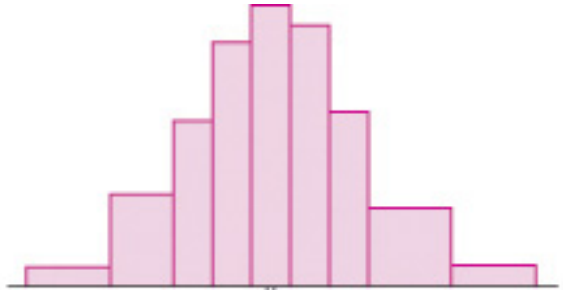


TABLE 14.2 Measurements of the coefficient of thermal expansion of structural steel.

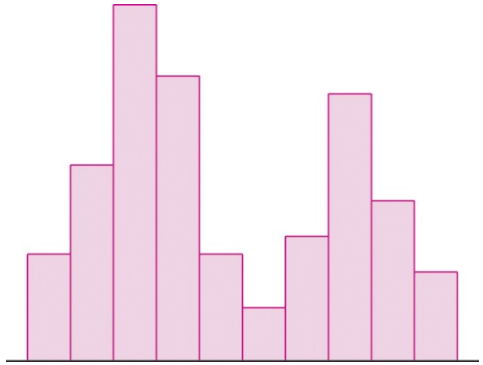
6.495	6.595	6.615	6.635	6.485	6.555
6.665	6.505	6.435	6.625	6.715	6.655
6.755	6.625	6.715	6.575	6.655	6.605
6.565	6.515	6.555	6.395	6.775	6.685

More on Histograms (1/2)

- Unimodal and Bimodal Histograms



- A unimodal histogram

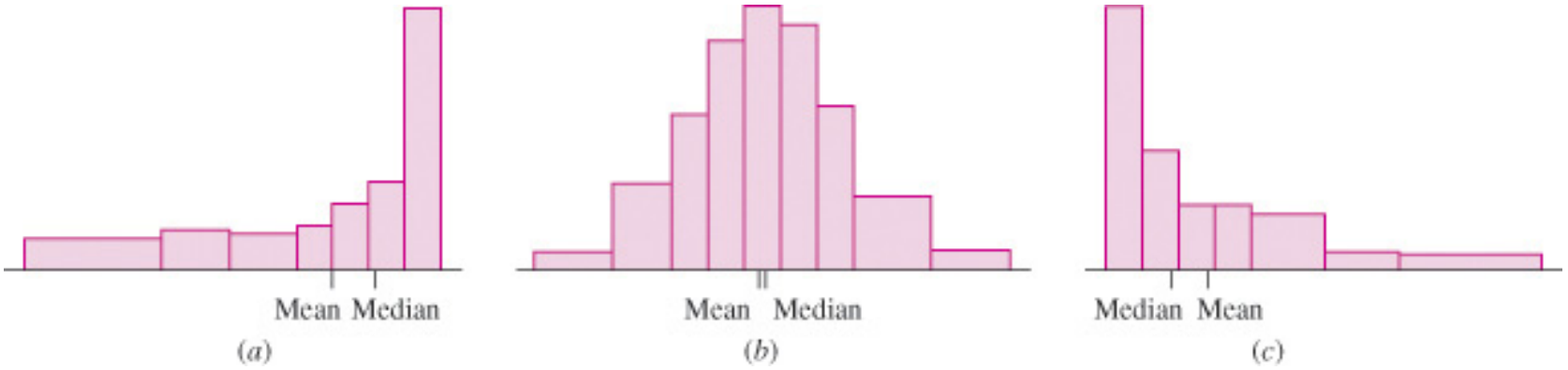


- A bimodal histogram

  - A bimodal histogram, in some cases, indicates that the sample can be divides into two subsamples that differ from each other in some scientifically important way

More on Histograms (2/2)

- Histograms that are not symmetric are referred to as **skewed**
- A histogram with a long left-hand tail is said to be **skewed to the left**, or **negatively skewed**
  - E.g., grades of the numerical methods course are left skewed (?)
- A histogram with a long right-hand tail is said to be **skewed to the right**, or **positively skewed**
  - E.g., incomes are right skewed (?)



- skewed to the left
- nearly symmetric
- skewed to the right

Descriptive Statistics in MATLAB

- MATLAB has several built-in commands to compute and display descriptive statistics. Assuming some column vector *s*
  - `mean(s), median(s), mode(s)`
    - Calculate the mean, median, and mode of *s*. `mode` is a part of the statistics toolbox.
  - `min(s), max(s)`
    - Calculate the minimum and maximum value in *s*.
  - `var(s), std(s)`
    - Calculate the variance and standard deviation of *s*
- **Note** - if a matrix is given, the statistics will be returned for each column

Histograms in MATLAB

- `[n, x] = hist(y, x)`
  - *n* is the number of elements in each bin; *x* is a vector specifying the midpoint of each bin; *y* is the vector being analyzed
- `[n, x] = hist(y, m)`
  - Determine the number of elements in each bin of data in *s* using ***m* bins**. *x* will contain the centers of the bins. The default case is *m*=10
- `hist(y, x)` or `hist(y, m)` or `hist(y)`
  - With no output arguments, `hist` will actually produce a histogram bar plot with 10 bins determined automatically based on the range of values in *y*

Histogram Example

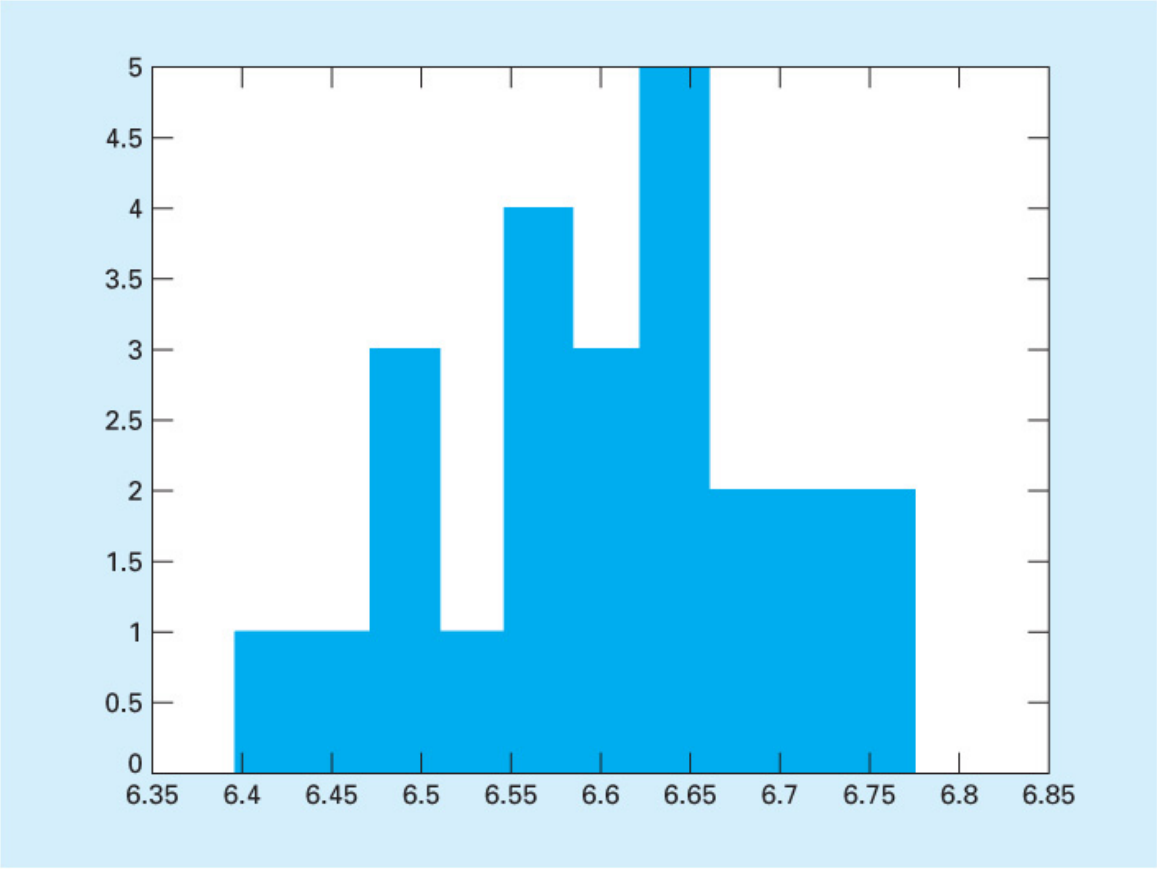


FIGURE 14.4
Histogram generated with the MATLAB `hist` function.

Linear Least-Squares Regression

- **Linear least-squares regression** is a method to determine the "best" coefficients in a linear model for given data set
- "Best" for least-squares regression means minimizing the sum of the squares of the *estimate* residuals. For a straight line model, this gives:

$$S_r = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2$$

  - n is the total number of points
- This method will yield a unique line for a given set of data

Least-Squares Fit of a Straight Line

- Using the model:

$$y = a_0 + a_1 x$$

  - The **slope** and **intercept** producing the best fit can be found using:

$$\frac{\partial S_r}{\partial a_0} = -2\sum\left(y_i - a_0 - a_1 x_i\right)$$

$$\frac{\partial S_r}{\partial a_1} = -2\sum\left[\left(y_i - a_0 - a_1 x_i\right)x_i\right]$$

$$n\,a_0 + \left(\sum x_i\right)a_1 = \sum y_i$$

$$\left(\sum x_i\right)a_0 + \left(\sum x_i^2\right)a_1 = \sum x_i y_i$$

Normal Equations

$$a_1 = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - \left(\sum x_i\right)^2}$$

$$a_0 = \bar{y} - a_1 \bar{x}$$

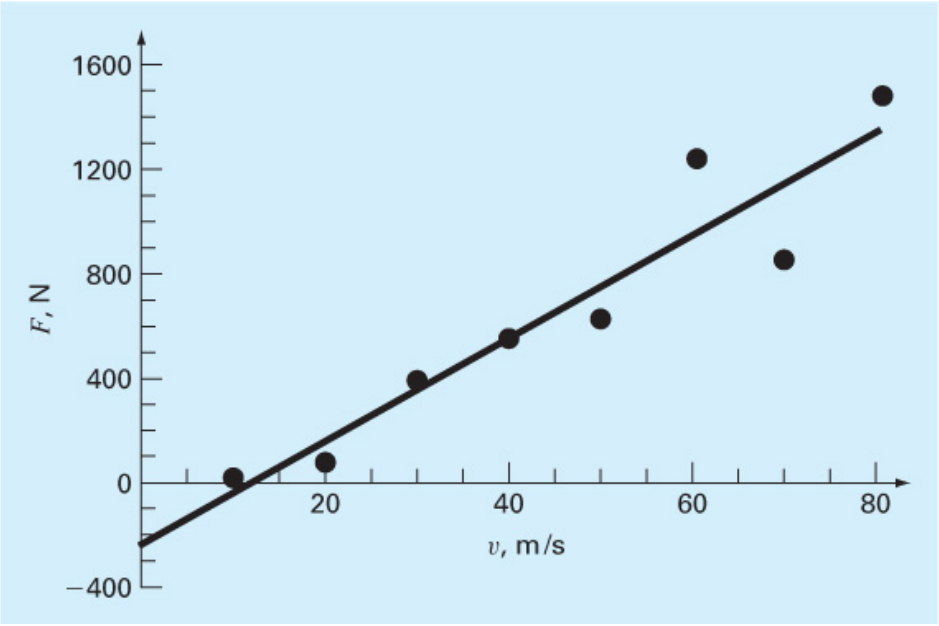
An Example

	V (m/s)	F (N)		
i	x_i	y_i	$(x_i)^2$	$x_i y_i$
1	10	25	100	250
2	20	70	400	1400
3	30	380	900	11400
4	40	550	1600	22000
5	50	610	2500	30500
6	60	1220	3600	73200
7	70	830	4900	58100
8	80	1450	6400	116000
Σ	360	5135	20400	312850

$$a_1 = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - \left(\sum x_i\right)^2} = \frac{8(312850) - (360)(5135)}{8(20400) - (360)^2} = 19.47024$$

$$a_0 = \bar{y} - a_1 \bar{x} = 641.875 - 19.47024(45) = -234.2857$$

$$F_{est} = -234.2857 + 19.47024v$$



Quantification of Error

- Recall for a straight line, the sum of the squares of the estimate residuals:

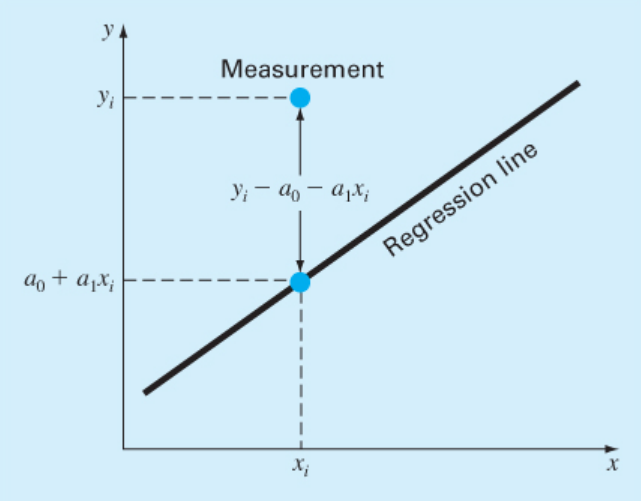


FIGURE 14.9
The residual in linear regression represents the vertical distance between a data point and the straight line.

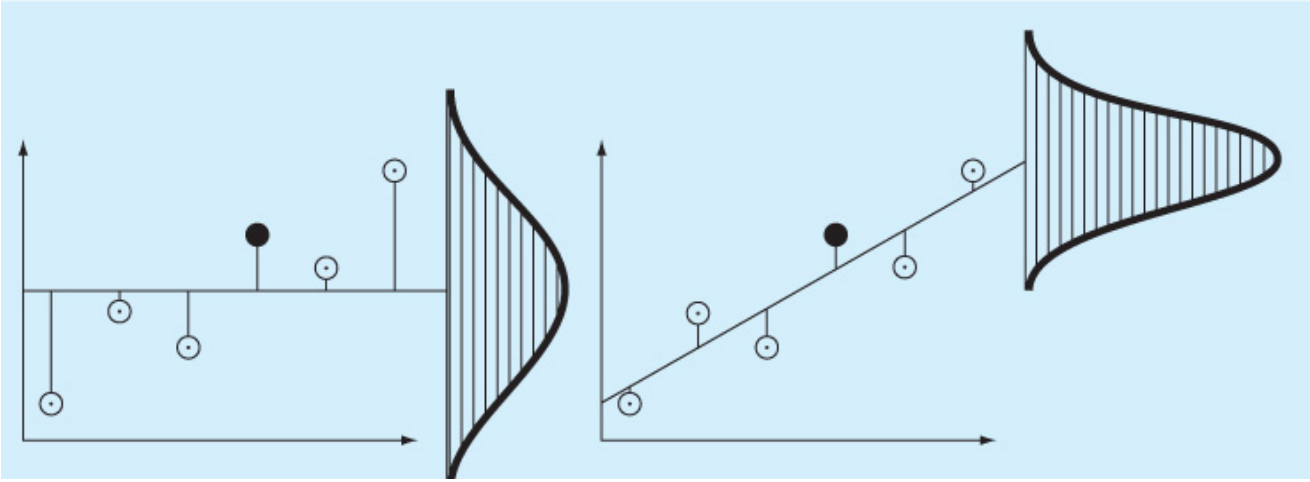
$$S_r = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2$$

- ***Standard error of the estimate:***
  - Quantify the spread around the regression line

$$s_{y/x} = \sqrt{\frac{S_r}{n-2}}$$

Standard Error of the Estimate

- Regression data showing (a) the spread of data around the mean of the dependent data and (b) the spread of the data around the best fit line:



- The reduction in spread, as those going from (a) to (b), represents the improvement due to linear regression

Coefficient of Determination (1/2)

- The ***coefficient of determination*** r^2 is the difference between the sum of the squares of the data residuals and the sum of the squares of the estimate residuals, normalized by the sum of the squares of the data residuals:

$$r^2 = \frac{S_t - S_r}{S_t}$$

- r^2 represents the percentage of the original uncertainty explained by the model
- For a perfect fit, S_r=0 and r^2=1
- If r^2=0, there is no improvement over simply picking the mean
- If r^2<0, the model is *worse* than simply picking the mean!

Coefficient of Determination (2/2)

- An alternative formulation for *r* is more convenient for computer implementation:

$$r = \frac{n\sum(x_i y_i) - (\sum x_i)(\sum y_i)}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}}$$

An Example

	V (m/s)	F (N)			
i	x_i	y_i	$a_0+a_1x_i$	$(y_i-\bar{y})^2$	$(y_i-a_0-a_1x_i)^2$
1	10	25	-39.58	380535	4171
2	20	70	155.12	327041	7245
3	30	380	349.82	68579	911
4	40	550	544.52	8441	30
5	50	610	739.23	1016	16699
6	60	1220	933.93	334229	81837
7	70	830	1128.63	35391	89180
8	80	1450	1323.33	653066	16044
Σ	360	5135		1808297	216118

$$F_{est} = -234.2857 + 19.47024v$$

$$S_t = \sum (y_i - \bar{y})^2 = 1808297$$

$$S_r = \sum (y_i - a_0 - a_1 x_i)^2 = 216118$$

$$s_y = \sqrt{\frac{1808297}{8-1}} = 508.26$$

$$s_{y/x} = \sqrt{\frac{216118}{8-2}} = 189.79$$

$$r^2 = \frac{1808297 - 216118}{1808297} = 0.8805$$

88.05% of the original uncertainty has been explained by the linear model

Paradox

- Four data sets consisting of 11 points each have the same best-fit equation y=3+0.5x

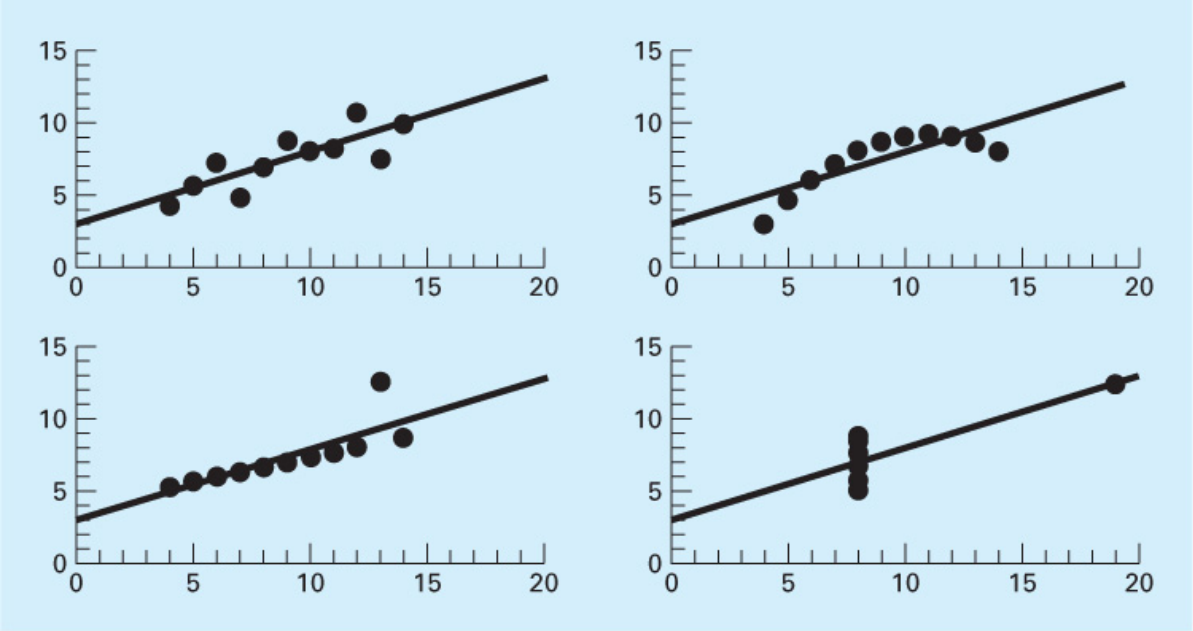


FIGURE 14.12
Anscombe's four data sets along with the best-fit line, $y = 3 + 0.5x$.

  - *You should always inspect a plot of the data along with your regression curve*

Nonlinear Relationships

- Linear regression is predicated on the fact that the relationship between the dependent and independent variables is linear - this is not always the case
- Three common examples are:

$$\text{exponential equation}: \quad y = \alpha_1 e^{\beta_1 x}$$

$$\text{power equation}: \quad y = \alpha_2 x^{\beta_2}$$

$$\text{saturation - growth - rate equation}: \quad y = \alpha_3 \frac{x}{\beta_3 + x}$$

Linearization of Nonlinear Relationships

- One option for finding the coefficients for a nonlinear fit is to linearize it. For the three common models, this may involve taking logarithms or inversion:

Model	Nonlinear	Linearized
exponential :	$y = \alpha_1 e^{\beta_1 x}$	$\ln y = \ln \alpha_1 + \beta_1 x$
power :	$y = \alpha_2 x^{\beta_2}$	$\log y = \log \alpha_2 + \beta_2 \log x$
saturation - growth - rate :	$y = \alpha_3 \frac{x}{\beta_3 + x}$	$\frac{1}{y} = \frac{1}{\alpha_3} + \frac{\beta_3}{\alpha_3} \frac{1}{x}$

Transformation Examples

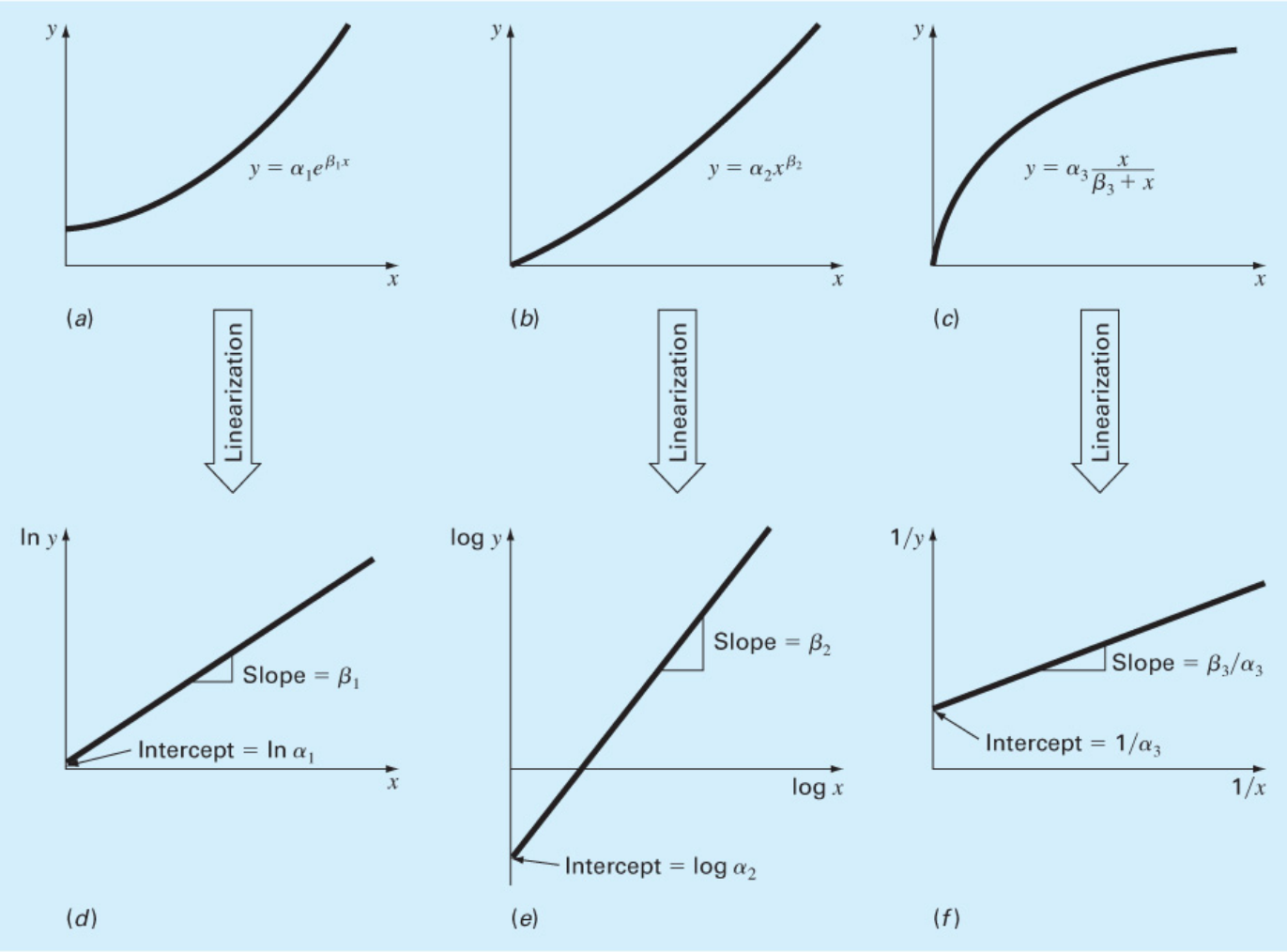


FIGURE 14.13
(*a*) The exponential equation, (*b*) the power equation, and (*c*) the saturation-growth-rate equation. Parts (*d*), (*e*), and (*f*) are linearized versions of these equations that result from simple transformations.

Linear Regression Program

FIGURE 14.15
An M-file to implement linear regression.

```
function [a, r2] = linregr(x,y)
% linregr: linear regression curve fitting
%    [a, r2] = linregr(x,y): Least squares fit of straight
%              line to data by solving the normal equations

% input:
%   x = independent variable
%   y = dependent variable
% output:
%   a = vector of slope, a(1), and intercept, a(2)
%   r2 = coefficient of determination

n = length(x);
if length(y)~=n, error('x and y must be same length'); end
x = x(:); y = y(:);     % convert to column vectors
sx = sum(x); sy = sum(y);
sx2 = sum(x.*x); sxy = sum(x.*y); sy2 = sum(y.*y);
a(1) = (n*sxy-sx*sy)/(n*sx2-sx^2);
a(2) = sy/n-a(1)*sx/n;
r2 = ((n*sxy-sx*sy)/sqrt(n*sx2-sx^2)/sqrt(n*sy2-sy^2))^2;
% create plot of data and best fit line
xp = linspace(min(x),max(x),2);
yp = a(1)*xp+a(2);
plot(x,y,'o',xp,yp)
grid on
```

MATLAB Functions

- MATLAB has a built-in function **`polyfit`** that fits a least-squares nth order polynomial to data:
  - `p = polyfit(x, y, n)`
    - `x`: independent data
    - `y`: dependent data
    - `n`: order of polynomial to fit
    - `p`: coefficients of polynomial
      $f(x)=p_1x^n+p_2x^{n-1}+\ldots+p_nx+p_{n+1}$
- MATLAB's **`polyval`** command can be used to compute a value using the coefficients.
  - `y = polyval(p, x)`