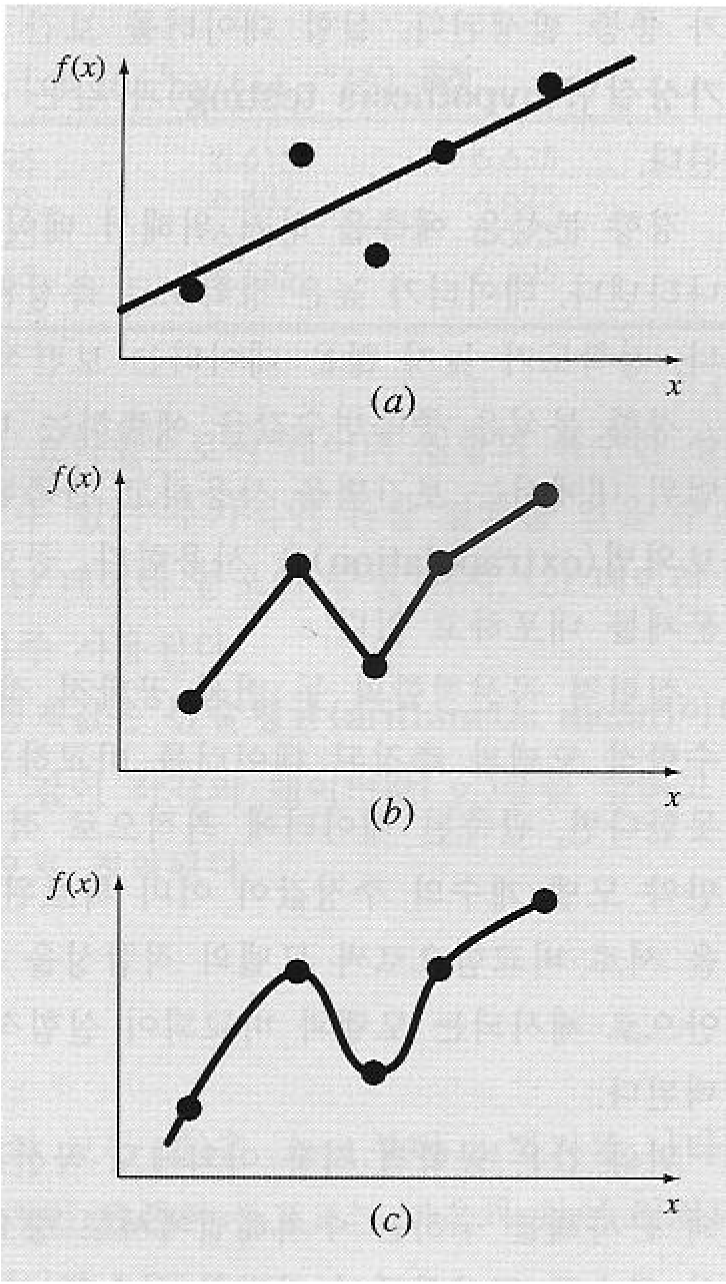
5. 보간법과 회귀분석

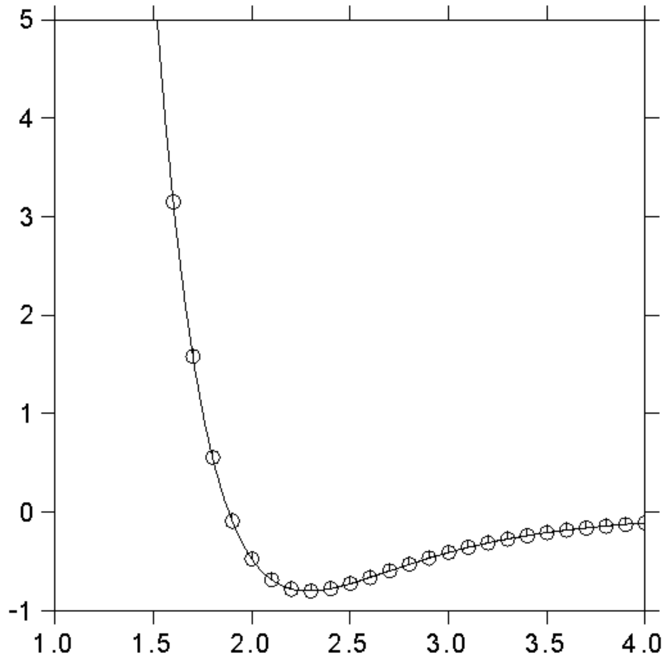
Interpolation and Regression



1. 보간법 (Interpolation)

1.1 서론

- 응용 예 : interatomic pair-wise interaction



- Taylor Series (*one-point approximation*)를 사용할 수 없는 이유 (Taylor Series doesn't work)
  Approximate $f(x)=1/x$ at $x=3$, using a Taylor expansion at $x=1$.

$$f^{(n)}(x) = (-1)^n n! x^{-n-1}$$

$$P_n(x) = \sum_{k=0}^{n} \frac{f^{(k)}(1)}{k!}(x-1)^k = \sum_{k=0}^{n} (-1)^k (x-1)^k$$

n	0	1	2	3	4	5	6	7
$P_n(3)$	1	−1	3	−5	11	−21	43	−85

1.2 Newton

- Taylor Series 와의 관계 (finite divided difference)

Forward difference $$f'(x_i)=\frac{f(x_{i+1})-f(x_i)}{x_{i+1}-x_i}+O(x_{i+1}-x_i)$$

Backward difference $$f'(x_i)=\frac{f(x_i)-f(x_{i-1})}{h}+O(h)$$

Centered difference $$f'(x_i)=\frac{f(x_{i+1})-f(x_{i-1})}{2h}-O(h^2)$$

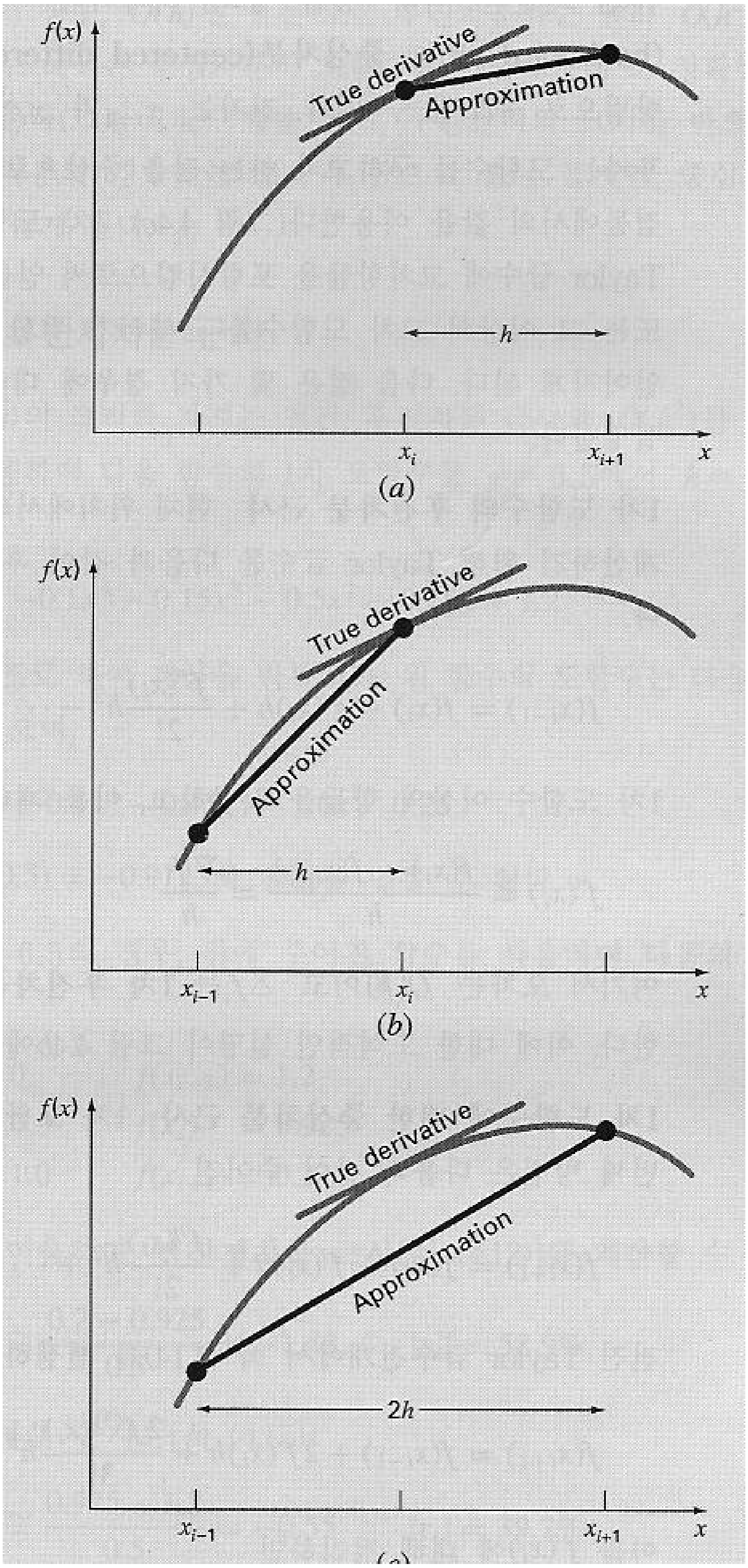
Example) 다음 함수의 1 차도함수를 x=0.5 에서 h=0.5, 0.25 로 위 세가지 방법을 사용하여 계산하라. (참값 -0.9125)
Find the derivative at x=0.5 with h=0.5, 0.25 using the above-mentioned three different method (answer: -0.9125)

$$f(x)=-0.1x^4-0.15x^3-0.5x^2-0.25x+1.2$$

고차 도함수의 유한차분 근사 (higher order derivative)

$$f''(x_i)=\frac{f(x_{i+1})-2f(x_i)+f(x_{i-1})}{h^2}+O(h^2)$$

$$f''(x_i)\cong\frac{\dfrac{f(x_{i+1})-f(x_i)}{h}-\dfrac{f(x_i)-f(x_{i-1})}{h}}{h}$$



f(x)
True derivative
Approximation
h
x
(a)
f(x)
True derivative
Approximation
h
x
(b)
f(x)
True derivative
Approximation
2h
x

- 1 차 (선형) 보간법 (first-order interpolation)

$$f_1(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}\ (x - x_0)$$

Forward difference $$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} + O(x_{i+1} - x_i)$$

Example) 선형 보간법을 사용하여 ln2 를 x = 1~6, x = 1~4 구간에서 계산하고 구간 크기의 영향을 조사하라.
(0.3583519, 0.4620981 vs. 0.69314718)

- 2 차 보간법 (second-order interpolation)

$$f_2(x) = b_o + b_1(x - x_0) + b_2(x - x_0)(x - x_1)$$

$$b_o = f(x_o)$$

$$b_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

$$b_2 = \frac{\dfrac{f(x_2) - f(x_1)}{x_2 - x_1} - \dfrac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_o}$$

Example) x = 1, 4, 6 에서의 값에 기반을 둔 2 차 보간법을 사용하여 ln2 값을 구하라.
Find the value of ln 2 based on the followings
ln 1 = 0
ln 4 = 1.386294
ln 6 = 1.791759

$$f_2(x) = 0 + 0.4620981(x-1) + -0.0518731(x-1)(x-4)$$

= 0.5658444

- Newton 보간 다항식의 일반화 (generalization of Newton method)

$$f_n(x) = b_o + b_1(x - x_0) + \cdots + b_n(x - x_0)(x - x_1)\cdots(x - x_{n-1})$$

$$b_o = f(x_o)$$

$$b_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = f[x_1, x_0]$$

$$b_2 = \frac{\dfrac{f(x_2) - f(x_1)}{x_2 - x_1} - \dfrac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_o} = f[x_2, x_1, x_0]$$

$$\vdots$$

$$b_n = f[x_n, x_{n-1}, \cdots, x_1, x_0]$$

$$f[x_i, x_j, x_k] = \frac{f[x_i, x_j] - f[x_j, x_k]}{x_i - x_k}$$

$$f[x_n, x_{n-1}, \cdots, x_1, x_0] = \frac{f[x_n, x_{n-1}, \cdots, x_1] - f[x_{n-1}, x_{n-2}, \cdots, x_0]}{x_n - x_0}$$

$$f_n(x) = f(x_0) + (x - x_0)f[x_1, x_0] + (x - x_0)(x - x_1)f[x_2, x_1, x_0]$$

$$+ \cdots + (x - x_0)(x - x_1)\cdots(x - x_{n-1})f[x_n, x_{n-1}, \cdots, x_0]$$

x	$f(x)$	First Divided Differences	Second Divided Differences	Third Divided Differences
x_0	$f[x_0]$			
		$f[x_0, x_1] = \frac{f[x_1]-f[x_0]}{x_1-x_0}$		
x_1	$f[x_1]$		$f[x_0, x_1, x_2] = \frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0}$	
		$f[x_1, x_2] = \frac{f[x_2]-f[x_1]}{x_2-x_1}$		$f[x_0, x_1, x_2, x_3] = \frac{f[x_1,x_2,x_3]-f[x_0,x_1,x_2]}{x_3-x_0}$
x_2	$f[x_2]$		$f[x_1, x_2, x_3] = \frac{f[x_2,x_3]-f[x_1,x_2]}{x_3-x_1}$	
		$f[x_2, x_3] = \frac{f[x_3]-f[x_2]}{x_3-x_2}$		$f[x_1, x_2, x_3, x_4] = \frac{f[x_2,x_3,x_4]-f[x_1,x_2,x_3]}{x_4-x_1}$
x_3	$f[x_3]$		$f[x_2, x_3, x_4] = \frac{f[x_3,x_4]-f[x_2,x_3]}{x_4-x_2}$	
		$f[x_3, x_4] = \frac{f[x_4]-f[x_3]}{x_4-x_3}$		$f[x_2, x_3, x_4, x_5] = \frac{f[x_3,x_4,x_5]-f[x_2,x_3,x_4]}{x_5-x_2}$
x_4	$f[x_4]$		$f[x_3, x_4, x_5] = \frac{f[x_4,x_5]-f[x_3,x_4]}{x_5-x_3}$	
		$f[x_4, x_5] = \frac{f[x_5]-f[x_4]}{x_5-x_4}$		
x_5	$f[x_5]$			

```
Subroutine NewtInt (x,y,n,xi,yint,ea)
    LOCAL fdd(n,n)
    DO i = 0, n
        fdd(i,0) = y(i)
    END DO
    DO j = 1, n
        DO i = 0, n-j
            fdd(i,j) = (fdd(i+1,j-1)-fdd(i,j-1)) / (x(i+1)-x(i))
        END DO
    END DO
    xterm = 1
    yint(0) = fdd(0,0)
    DO order = 1, n
        xterm = xterm * (xi - x(order-1))
        yint(order) = yint(order-1) + fdd(0,order) * xterm
        ea(order) = yint(order) - yint(order-1)
    END DO
END NewtInt
```

1.3 Lagrange Polynomials

$$f_n(x) = \sum_{k=0}^{n} L_{n,k}(x)\ f(x_k)$$

$$L_{n,k}(x) = \prod_{\substack{j=0 \\ j \neq k}}^{n} \frac{x - x_j}{x_k - x_j}$$

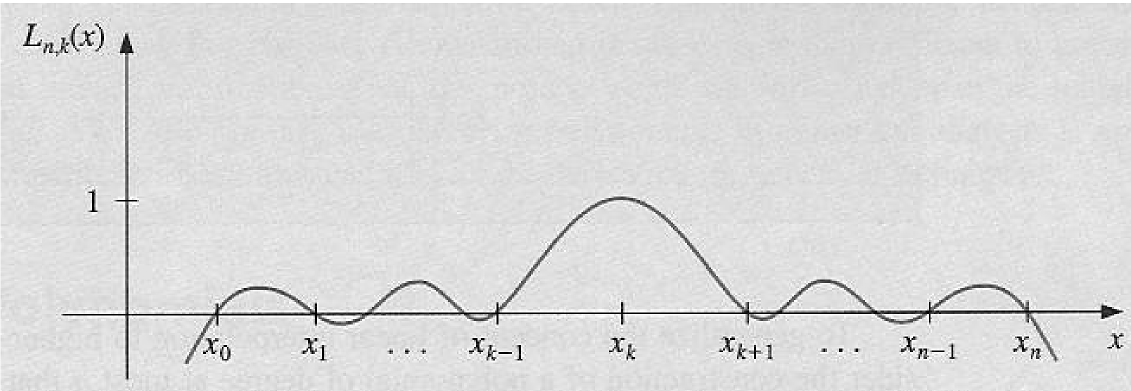
- 선형 보간 (n = 1) first-order

$$f_1(x) = \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1)$$

- 2 차 보간 second-order

$$f_2(x) = \frac{x - x_1}{x_0 - x_1} \frac{x - x_2}{x_0 - x_2} f(x_0) + \frac{x - x_0}{x_1 - x_0} \frac{x - x_2}{x_1 - x_2} f(x_1)$$
$$+ \frac{x - x_0}{x_2 - x_0} \frac{x - x_1}{x_2 - x_1} f(x_2)$$

$$L_{n,k}(x) = \frac{(x - x_0) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_n)}{(x_k - x_0) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_n)}$$



$L_{n,k}(x)$는 $x = x_k$ 에서는 1, 나머지 x 값에서는 0 이 되는 n차 함수이다.

⇒ $f_n(x)$는 모든 n+1 개의 데이터 점들을 정확하게 통과하는 유일한 n차 다항식이 된다.

nth Lagrange Interpolating Polynomial

$$P_n(x) = f(x_0)L_{n,0}(x) + \cdots + f(x_n)L_{n,n}(x) = \sum_{k=0}^{n} f(x_k)L_{n,k}(x),$$

where

$$L_{n,k}(x) = \frac{(x - x_0)(x - x_1)\cdots(x - x_{k-1})(x - x_{k+1})\cdots(x - x_n)}{(x_k - x_0)(x_k - x_1)\cdots(x_k - x_{k-1})(x_k - x_{k+1})\cdots(x_k - x_n)}$$

for each $k = 0, 1, \ldots, n$.

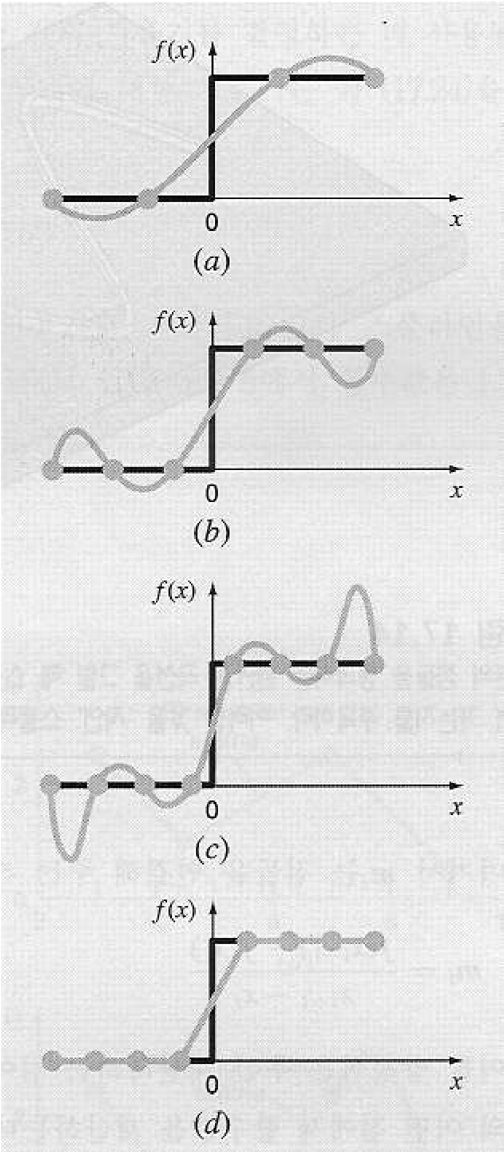
- N 차 결과를 n+1 차에 사용하도록 개선 → Neville's method

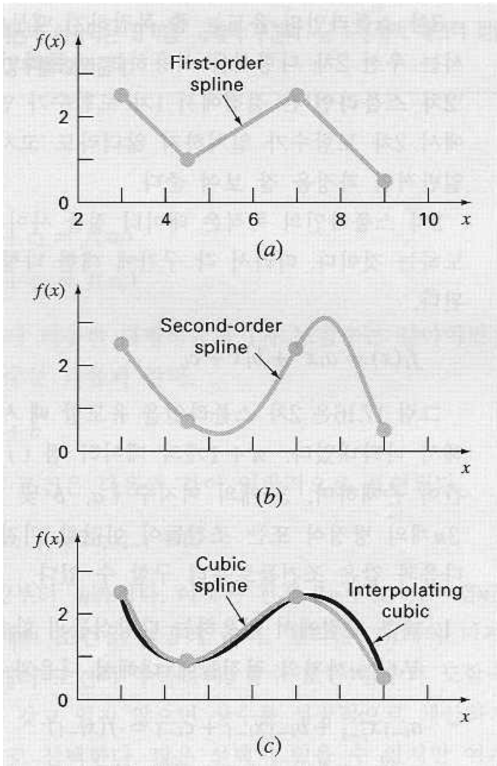
Example) 1, 2 차 Lagrange 보간 다항식을 이용하여 ln2 값을 구하라.
Find the value of ln 2 using 1st and 2nd order Lagrange method.
ln 1 = 0
ln 4 = 1.386294
ln 6 = 1.791759
1, 2 차 Newton 보간 다항식의 결과와 비교하라.
Compare with those from Newton method.

Example) Lagrange 보간 다항식은 Newton 제차분 보간 다항식으로부터 유도될 수 있는 것임을 보여라.
(1 차인 경우를 고려해 볼 것)
Show that the Lagrange polynomials could be derived from the Newton polynomials. (Consider 1st order)

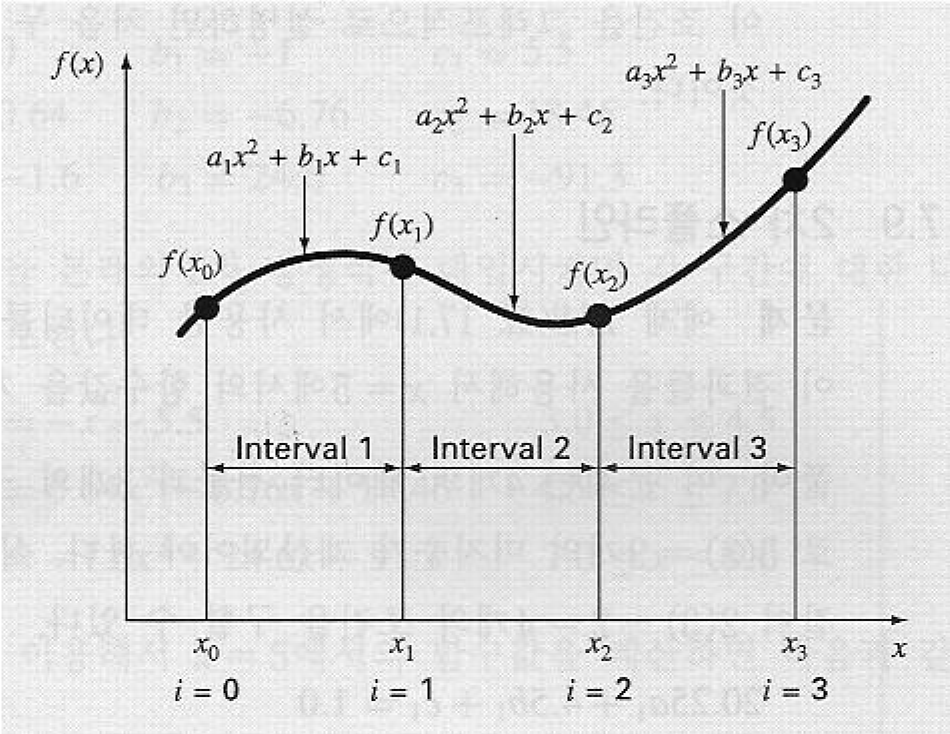
```
Subroutine Lagrng (x,y,n,xi)
    sum = 0
    DO i = 0, n
        product = y(i)
        DO j = 0, n
            IF i ≠ j THEN
                product = product * (x – x(j)) / (x(i) – x(j))
            ENDIF
        END DO
        sum = sum + product
    END DO
    Lagrng = sum
END Lagrng
```

1.4 Spline (Piecewise polynomial approximation)





- 2 차 스플라인 2nd order spline



for n+1 data point

$f_i(x) = a_i x^2 + b_i x + c_i$; 총 미지수 개수 (total unknowns) = $3n$

1. 내부 절점에서 이웃하는 다항식들의 함수 값이 같아야 한다.
   The values of neighboring polynomials on each inner node point should be the same.

$$a_{i-1}x_{i-1}^2 + b_{i-1}x_{i-1} + c_{i-1} = f(x_{i-1})$$

$$a_i x_{i-1}^2 + b_i x_{i-1} + c_i = f(x_{i-1}) \qquad (2n-2)$$

2. 첫번째와 마지막 함수는 반드시 끝점을 통과해야만 한다.
   The first and last polynomial should pass the endpoints.

$$a_1 x_0^2 + b_1 x_0 + c_0 = f(x_0)$$

$$a_n x_n^2 + b_n x_n + c_n = f(x_n) \qquad (2)$$

3. 내부 절점에서 이웃한 다항식들의 1 차 도함수는 같아야 한다.
   The values of 1st order derivatives of neighboring polynomials on each inner node point should be the same.

$$2a_{i-1}x_{i-1} + b_{i-1} = 2a_i x_{i-1} + b_i \qquad (n-1)$$

4. 2 차 도함수가 첫 번째 데이터 점에서 0 이라고 가정한다.
   The 2nd derivative on the first data point is assumed to be zero,

$$a_1 = 0 \qquad (1)$$

Example) 다음 데이터를 2 차 스플라인으로 적합시키고 x=5 에서의 값을 구하라. (Find the value at x=5).

x	$f(x)$
3.0	2.5
4.5	1.0
7.0	2.5
9.0	0.5

$$20.25a_1 + 4.5b_1 + c_1 = 1.0$$
$$20.25a_2 + 4.5b_2 + c_2 = 1.0$$
$$49a_2 + 7b_2 + c_2 = 2.5$$
$$49a_3 + 7b_3 + c_3 = 2.5$$

$$9a_1 + 3b_1 + c_1 = 2.5$$
$$81a_3 + 9b_3 + c_3 = 0.5$$

$$9a_1 + b_1 = 9a_2 + b_2$$
$$14a_2 + b_2 = 14a_3 + b_3$$

$$\begin{array}{lll} a_1 = 0 & b_1 = -1 & c_1 = 5.5 \\ a_2 = 0.64 & b_2 = -6.76 & c_2 = 18.46 \\ a_3 = -1.6 & b_3 = 24.6 & c_3 = -91.3 \end{array}$$

$$\begin{array}{ll} f_1(x) = -x + 5.5 & 3.0 \le x \le 4.5 \\ f_2(x) = 0.64x^2 - 6.76x + 18.46 & 4.5 \le x \le 7.0 \\ f_3(x) = -1.6x^2 + 24.6x - 91.3 & 7.0 \le x \le 9.0 \end{array}$$

$$f_2(5) = 0.64(5)^2 - 6.76(5) + 18.46 = 0.66$$

- 3 차 (cubic) 스플라인 (Cubic spline)

$$f_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i$$

1. 함수 값은 내부 절점에서 같아야 한다. [2*n*−2]
2. 첫 번째와 마지막 함수는 양 끝점을 통과해야 한다. [2]
3. 내부 절점에서 1 차 도함수는 같아야 한다. [*n*−1]
4. 내부 절점에서 2 차 도함수도 같아야 한다. [*n*−1]
5. 양 끝점에서의 2 차 도함수는 0 이라고 가정한다. [2]
   (2 차 도함수가 0 이 아니라면 그 정보로 조건을 대체)

Method 1

$$f_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

Method 2

각 구간 함수의 2 차 도함수를 다음과 같이 1 차 Lagrange 보간 다항식으로 표현.

$$f_i''(x) = f_i''(x_{i-1})\frac{x - x_i}{x_{i-1} - x_i} + f_i''(x_i)\frac{x - x_{i-1}}{x_i - x_{i-1}}$$

각 함수 값이 $x = x_{i-1}$ 에서 $f(x_{i-1})$, $x = x_i$ 에서 $f(x_i)$ 값을 가져야 한다는 2 개의 조건으로부터 적분 상수 값을 구하고 다음 표현식을 얻는다.

$$\begin{aligned} f_i(x) = {} & \frac{f_i''(x_{i-1})}{6(x_i - x_{i-1})}(x_i - x)^3 + \frac{f_i''(x_i)}{6(x_i - x_{i-1})}(x - x_{i-1})^3 \\ & + \left[\frac{f(x_{i-1})}{x_i - x_{i-1}} - \frac{f''(x_{i-1})(x_i - x_{i-1})}{6}\right](x_i - x) \\ & + \left[\frac{f(x_i)}{x_i - x_{i-1}} - \frac{f''(x_i)(x_i - x_{i-1})}{6}\right](x - x_{i-1}) \end{aligned}$$

From $f_{i+1}'(x_i) = f_i'(x_i)$

$$\begin{aligned} & (x_i - x_{i-1})f''(x_{i-1}) + 2(x_{i+1} - x_{i-1})f''(x_i) + (x_{i+1} - x_i)f''(x_{i+1}) \\ & = \frac{6}{x_{i+1} - x_i}[f(x_{i+1}) - f(x_i)] + \frac{6}{x_i - x_{i-1}}[f(x_{i-1}) - f(x_i)] \end{aligned}$$

Example) 다음 데이터를 3 차 스플라인으로 적합시키고 x=5 에서의 값을 구하라. (Find the value at x=5).

x	$f(x)$
3.0	2.5
4.5	1.0
7.0	2.5
9.0	0.5

$$(4.5-3)f''(3)+2(7-3)f''(4.5)+(7-4.5)f''(7)$$
$$=\frac{6}{7-4.5}(2.5-1)+\frac{6}{4.5-3}(2.5-1)$$

8 $f''(4.5)$ + 2.5 $f''(7)$ = 9.6
2.5 $f''(4.5)$ + 9 $f''(7)$ = −9.6

$f''(4.5)$ = 1.67909
$f''(7)$ = −1.53308

$$f_1\ (x)=\frac{1.67909}{6(4.5-3)}(x-3)^3+\frac{2.5}{4.5-3}(4.5-x)$$
$$+\left[\frac{1}{4.5-3}-\frac{1.67909(4.5-3)}{6}\right](x-3)$$

$$f_1(x)=0.186566(x-3)^3+1.666667(4.5-x)+0.246894(x-3)$$
$$f_2(x)=0.111939(7-x)^3-0.102205(x-4.5)^3-0.299621(7-x)+1.638783(x-4.5)$$
$$f_3(x)=-0.127757(9-x)^3+1.761027(9-x)+0.25(x-7)$$

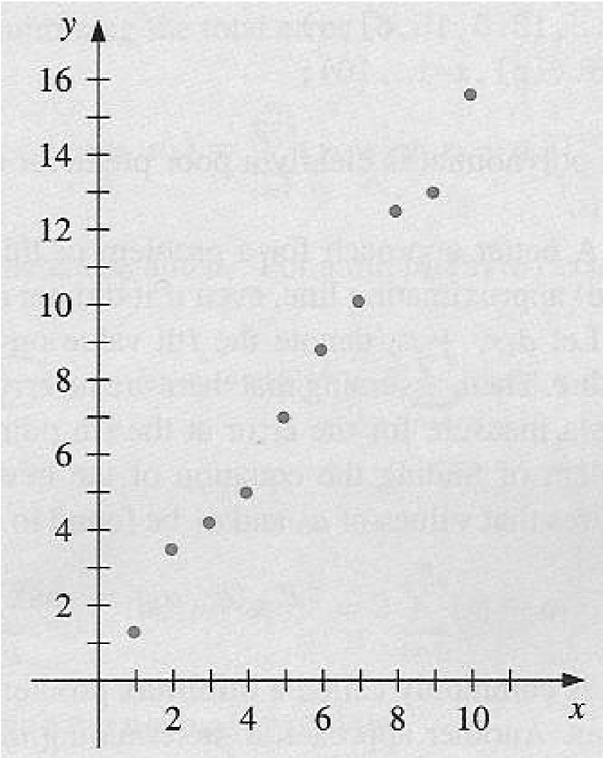
$$f_2(5)=1.102886$$

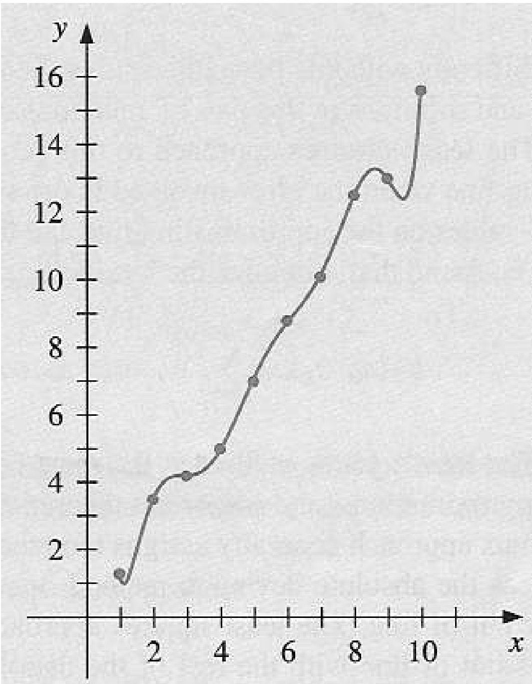
```
SUBROUTINE Spline (x,y,n,xu,yu,dy,d2y)
        LOCAL e(n), f(n), g(n), r(n), d2x(n)
        CALL Tridiag(x,y,n,e,f,g,r)
        CALL Gauss
        CALL Interpol(x,y,n,d2x,xu,yu,dy,d2y)
END Spline



SUBROUTINE Tridiag (x,y,n,e,f,g,r)
        f(1) = 2 * (x(2)-x(0))
        g(1) = (x(2)-x(1))
        r(1) = 6/(x(2)-x(1)) * (y(2)-y(1)) + 6/(x(1)-x(0)) * (y(0)-y(1))
        DO i = 2, n-2
            e(i) = (x(i) - x(i-1))
            f(i) = 2 * (x(i+1) - x(i-1))
            g(i) = (x(i+1) - x(i))
            r(i) = 6/(x(i+1)-x(i))*(y(i+1)-y(i)) + 6/(x(i)-x(i-1)) * (y(i-1)-y(i))
        ENDDO
        e(n-1) = (x(n-1) - x(n-2))
        f(n-1) = 2 * (x(n) - x(n-2))
        r(n-1) = 6/(x(n)-x(n-1)) * (y(n)-y(n-1))
                + 6/(x(n-1)-x(n-2)) * (y(n-2)-y(n-1))
END Tridiag
```

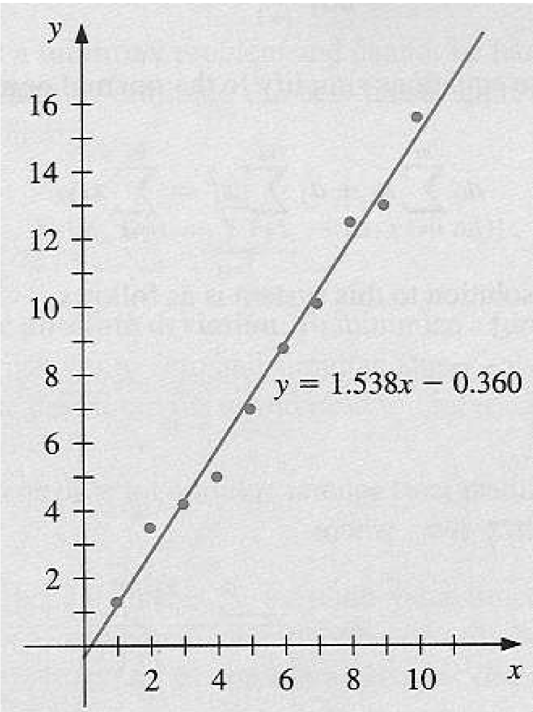
```
SUBROUTINE Interpol (x,y,n,d2x,xu,yu,dy,d2y)
        flag = 0
        i = 1
        DO
            IF xu >= x(i-1) AND xu =< x(i) THEN
                c1 = d2x(i-1)/6/(x(i)-x(i-1))
                c2 = d2x(i)/6/(x(i)-x(i-1))
                c3 = y(i-1)/(x(i)-x(i-1)) - d2x(i-1) * (x(i) - x(i-1))/6
                c4 = y(i)/(x(i)-x(i-1)) - d2x(i) * (x(i) - x(i-1))/6
                t1 = c1 * (x(i) - xu)^3
                t2 = c2 * (xu - x(i-1))^3
                t3 = c3 * (x(i) - xu)
                t4 = c4 * (xu - x(i-1))
                yu = t1 + t2 + t3 + t4
                t1 = -3 * c1 * (x(i)-xu)^2
                t2 = 3 * c2 * (xu - x(i-1))^2
                t3 = - c3
                t4 = c4
                dy = t1 + t2 + t3 + t4
                t1 = 6 * c1 * (x(i) - xu)
                t2 = 6 * c2 * (xu - x(i-1))
                d2y = t1 + t2
                flag = 1
            ELSE
                I = i + 1
            ENDIF
            IF i = n + 1 OR flag = 1 EXIT
        ENDDO
        IF flag = 0 THEN
          Print " outside range "
          Pause
        ENDIF
        END Interpol
```

2. 회귀분석 (Regression)

x_i	y_i	x_i	y_i
1	1.3	6	8.8
2	3.5	7	10.1
3	4.2	8	12.5
4	5.0	9	13.0
5	7.0	10	15.6

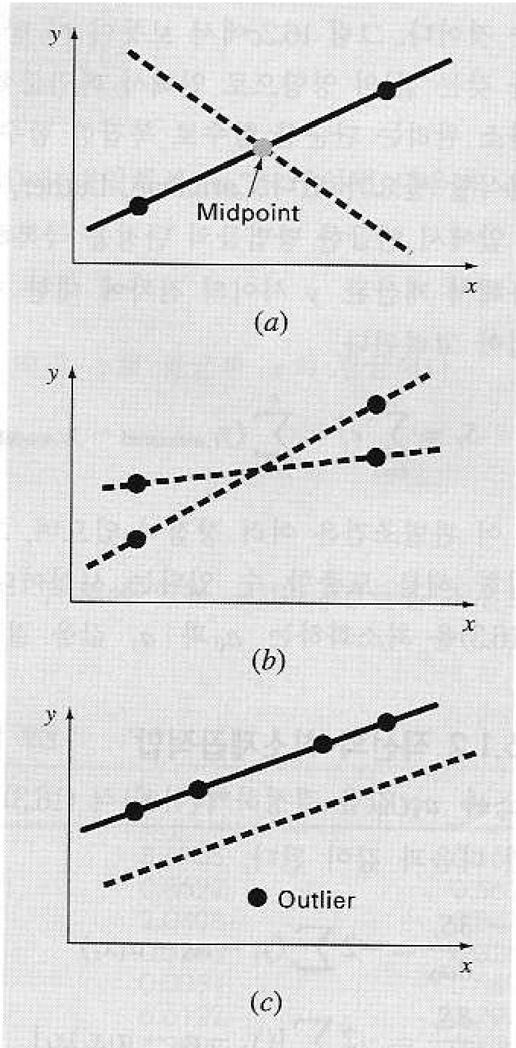






2.1 Discrete Least Square

- 선형회귀분석 도출 과정



회귀분석에 부적절한 "최적" 판별조건의 예들. (a) 잔차의 합을 최소화함, (b) 잔차의 절대값의 합을 최소화함, (c) 각 점의 최대오차값을 최소화함.

(a) Minimizing sum of deviation
(b) Minimizing sum of absolute deviation
(c) Minimizing maximum deviation (minimax)

- 선형 회귀분석 (Minimizing sum of squares of deviation)

$$y = a_0 + a_1 x + e$$

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

$$\frac{\partial S_r}{\partial a_0} = -2 \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i) = 0$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum_{i=1}^{n} x_i (y_i - a_0 - a_1 x_i) = 0$$

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

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

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

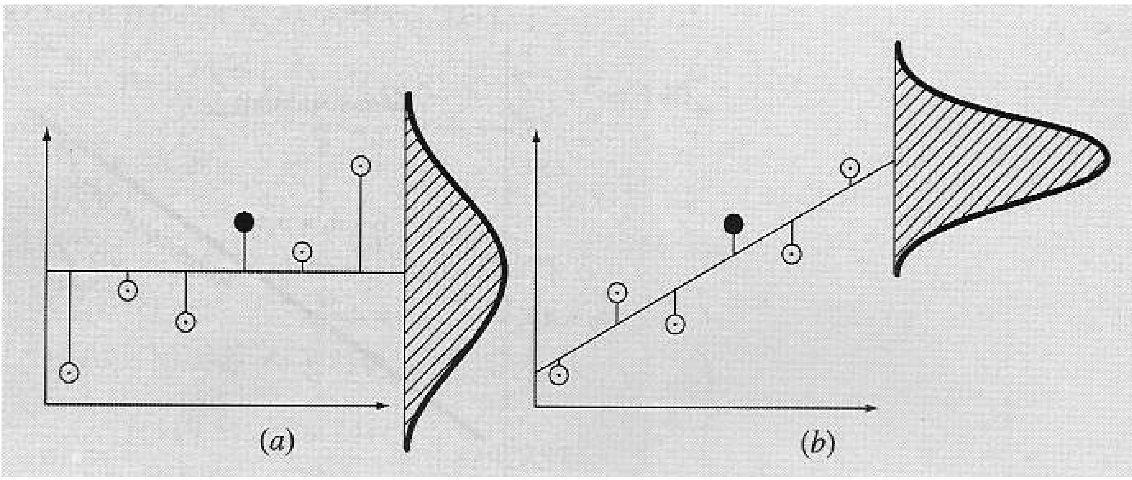
$$a_0 = \bar{y} - a_1 \bar{x}$$

x_i	y_i	x_i^2	$x_i y_i$	$P(x_i) = 1.538x_i - 0.360$
1	1.3	1	1.3	1.18
2	3.5	4	7.0	2.72
3	4.2	9	12.6	4.25
4	5.0	16	20.0	5.79
5	7.0	25	35.0	7.33
6	8.8	36	52.8	8.87
7	10.1	49	70.7	10.41
8	12.5	64	100.0	11.94
9	13.0	81	117.0	13.48
10	15.6	100	156.0	15.02
55	81.0	385	572.4	$E = \sum_{i=1}^{10} (y_i - P(x_i))^2 \approx 2.34$

- 오차의 정량화 Error evaluation

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

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



결정계수 (Coefficient of determination) $r^2 = \dfrac{S_t - S_r}{S_t}$

S_t : y 의 평균에 대한 제곱합

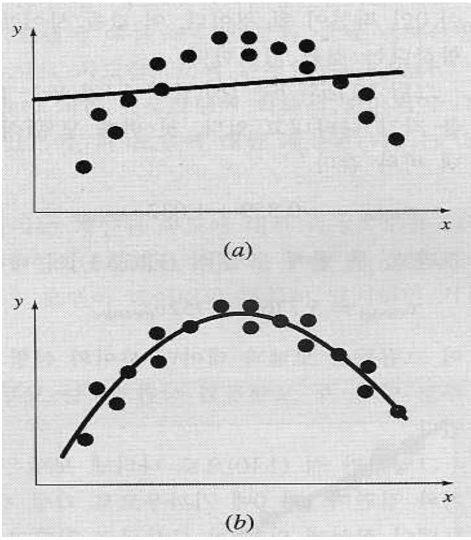
sum of squares of deviation wrt. mean y value

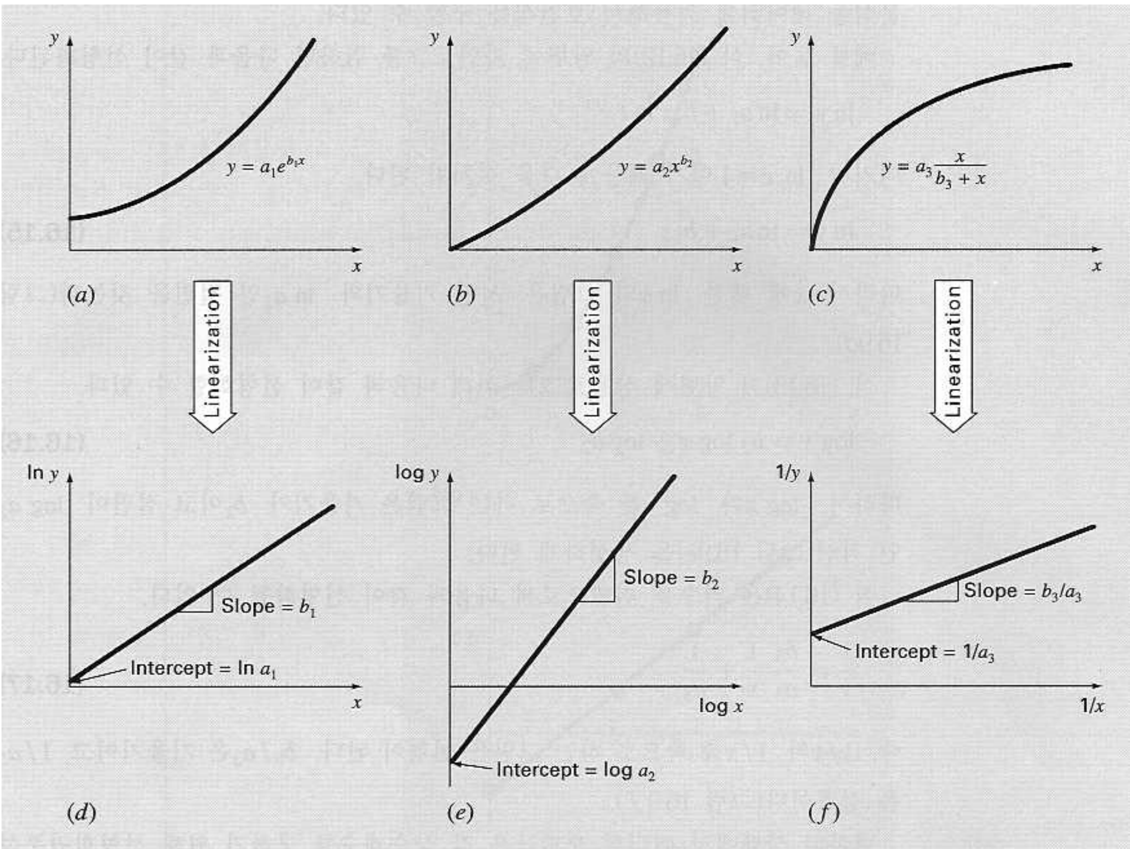
상관계수 (Correlation coefficient) r

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

```
SUB Regress (x, y, n, a1, a0, syx, r2)
    sumx = 0; sumxy = 0; st = 0
    sumy = 0; sumx2 = 0; sr = 0
    DO I = 1, n
       sumx = sumx + x(i)
       sumy = sumy + y(i)
       sumxy = sumxy + x(i)*y(i)
       sumx2 = sumx2 + x(i)*x(i)
    ENDDO
    xm = sumx / n
    ym = sumy / n
    a1 = (n*sumxy - sumx*sumy)/(n*sumx2 - sumx*sumx)
    a0 = ym - a1*xm
    DO i = 1, n
       st = st + (y(i) - ym)^2
       sr = sr + (y(i) - a1*x(i) - a0)^2
    ENDDO
    syx = (sr/(n-2))^0.5
    r2 = (st - sr) / st
END Regress
```

● 비선형 관계식의 선형화 (Linearization of nonlinear relation)





(a) 지수 방정식, (b) 멱 방정식, (c) 포화성장률 방정식. (d), (e), (f)는 간단한 변환에 의한 선형 방정식.

2.2 다항식 회귀분석 (polynomial regression)

$$y = a_0 + a_1 x + a_2 x^2 + e$$

$$S_r = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i - a_2 x_i^2)^2$$

$$\frac{\partial S_r}{\partial a_0} = -2 \sum (y_i - a_0 - a_1 x_i - a_2 x_i^2)$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum x_i (y_i - a_0 - a_1 x_i - a_2 x_i^2)$$

$$\frac{\partial S_r}{\partial a_2} = -2 \sum x_i^2 (y_i - a_0 - a_1 x_i - a_2 x_i^2)$$

$$(n) a_0 + \left(\sum x_i\right) a_1 + \left(\sum x_i^2\right) a_2 = \sum y_i$$

$$\left(\sum x_i\right) a_0 + \left(\sum x_i^2\right) a_1 + \left(\sum x_i^3\right) a_2 = \sum x_i y_i$$

$$\left(\sum x_i^2\right) a_0 + \left(\sum x_i^3\right) a_1 + \left(\sum x_i^4\right) a_2 = \sum x_i^2 y_i$$

Example)

x_i	y_i	$(y_i - \bar{y})^2$	$(y_i - a_0 - a_1x_i - a_2x_i^2)$
0	2.1	544.44	0.14332
1	7.7	314.47	1.00286
2	13.6	140.03	1.08158
3	27.2	3.12	0.80491
4	40.9	239.22	0.61951
5	61.1	1272.11	0.09439
Σ	152.6	2513.39	3.74657

$$m = 2 \qquad \sum x_i = 15 \qquad \sum x_i^4 = 979$$
$$n = 6 \qquad \sum y_i = 152.6 \qquad \sum x_i y_i = 585.6$$
$$\bar{x} = 2.5 \qquad \sum x_1^2 = 55 \qquad \sum x_i^2 y_i = 2488.8$$
$$\bar{y} = 25.433 \qquad \sum x_i^3 = 225$$

$$(n)a_0 + \left(\sum x_i\right) a_1 + \left(\sum x_i^2\right) a_2 = \sum y_i$$
$$\left(\sum x_i\right) a_0 + \left(\sum x_i^2\right) a_1 + \left(\sum x_i^3\right) a_2 = \sum x_i y_i$$
$$\left(\sum x_i^2\right) a_0 + \left(\sum x_i^3\right) a_1 + \left(\sum x_i^4\right) a_2 = \sum x_i^2 y_i$$

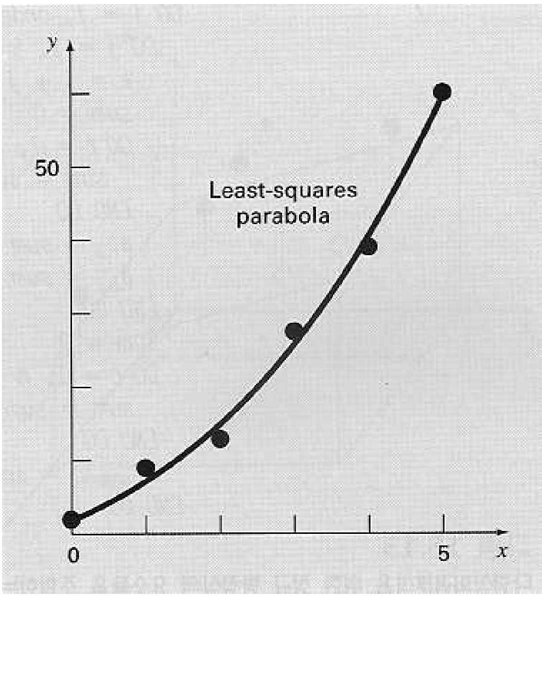
$$\begin{bmatrix} 6 & 15 & 55 \\ 15 & 55 & 225 \\ 55 & 225 & 979 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \end{Bmatrix} = \begin{Bmatrix} 152.6 \\ 585.6 \\ 2488.8 \end{Bmatrix}$$

$$y = 2.47857 + 2.35929x + 1.86071x^2$$

$$s_{y/x} = \sqrt{\frac{3.74657}{6-3}} = 1.12$$

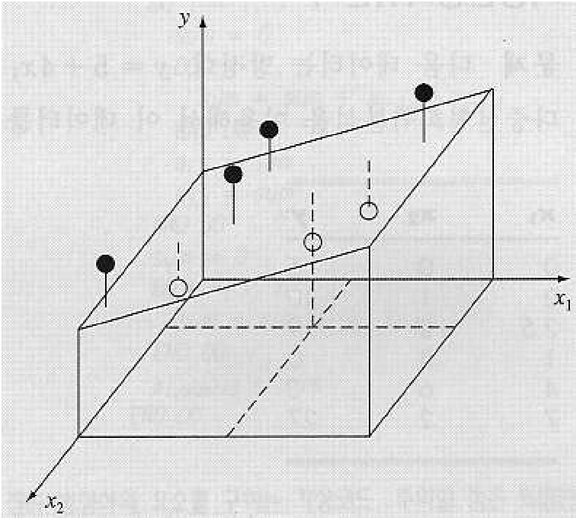
$$r^2 = \frac{2513.39 - 3.74657}{2513.39} = 0.99851$$

$$r = 0.99925$$



2.3 다중 선형 회귀분석 (multiple linear regression)

$$y = a_0 + a_1 x_1 + a_2 x_2 + e$$



$$S_r = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_{1i} - a_2 x_{2i})^2$$

$$\frac{\partial S_r}{\partial a_0} = -2 \sum (y_i - a_0 - a_1 x_{1i} - a_2 x_{2i})$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum x_{1i} (y_i - a_0 - a_1 x_{1i} - a_2 x_{2i})$$

$$\frac{\partial S_r}{\partial a_2} = -2 \sum x_{2i} (y_i - a_0 - a_1 x_{1i} - a_2 x_{2i})$$

$$\begin{bmatrix} n & \Sigma x_{1i} & \Sigma x_{2i} \\ \Sigma x_{1i} & \Sigma x_{1i}^2 & \Sigma x_{1i} x_{2i} \\ \Sigma x_{2i} & \Sigma x_{1i} x_{2i} & \Sigma x_{2i}^2 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \end{Bmatrix} = \begin{Bmatrix} \Sigma y_i \\ \Sigma x_{1i} y_i \\ \Sigma x_{2i} y_i \end{Bmatrix}$$

2.4 선형최소제곱의 일반화 (Generalization of linear least-square)

$$y = a_0 z_0 + a_1 z_1 + a_2 z_2 + \cdots + a_m z_m + e$$

{Y}=[Z]{A}+{E}

$$[Z] = \begin{bmatrix} z_{01} & z_{11} & \cdots & z_{m1} \\ z_{02} & z_{12} & \cdots & z_{m2} \\ \vdots & \vdots & \vdots & \vdots \\ z_{0n} & z_{1n} & \cdots & z_{mn} \end{bmatrix}$$

m: 모델 변수의 수 (number of variables)
n: data point 의 수 (number of data points)
$\{Y\}^T = [y_1\ y_2 \cdots y_n]$; y 의 관측값
$\{A\}^T = [a_0\ a_1 \cdots a_n]$; 미지계수의 열백터
$\{E\}^T = [e_1\ e_2 \cdots e_n]$; 잔차의 열벡터

$$S_r = \sum_{i=1}^{n} \left(y_i - \sum_{j=0}^{m} a_j z_{ji} \right)^2$$

$$\left[[Z]^T[Z]\right]\{A\} = \left\{[Z]^T\{Y\}\right\}$$

※ 단순선형회귀분석, 다항식회귀분석, 다중회귀분석 등
세가지 접근 방식이 모두 동일하며 같은 행렬식으로
간단하게 표현할 수 있다.

미지계수 {A}는 역행렬법으로 다음과 같이 구할 수 있다.

$$\{A\} = \left[[Z]^T[Z]\right]^{-1}\left\{[Z]^T\{Y\}\right\}$$

2.5 비선형 회귀분석 (Gauss-Newton 방법)

$$f(x) = a_0(1 - e^{-a_1 x}) + e$$

$$y_i = f(x_i) + e_i$$

Taylor 전개

$$f(x_i)_{j+1} = f(x_i)_j + \frac{\partial f(x_i)_j}{\partial a_0}\Delta a_0 + \frac{\partial f(x_i)_j}{\partial a_1}\Delta a_1$$

j : 초기 가정값 (initial), $j+1$: 예측 (prediction)

$\Delta a_0 = a_{0,j+1} - a_{0,j}$ $\Delta a_1 = a_{1,j+1} - a_{1,j}$

원래의 비선형 모델을 선형 모델로 변환 (nonlinear -> linear)

$$y_i - f(x_i)_j = \frac{\partial f(x_i)_j}{\partial a_0}\Delta a_0 + \frac{\partial f(x_i)_j}{\partial a_1}\Delta a_1 + e_i$$

{D}=[Zj]{ΔA}+{E}

$$[Z_j] = \begin{bmatrix} \partial f_1/\partial a_0 & \partial f_1/\partial a_1 \\ \partial f_2/\partial a_0 & \partial f_2/\partial a_0 \\ \vdots & \vdots \\ \partial f_n/\partial a_0 & \partial f_n/\partial a_0 \end{bmatrix}$$

$$\{D\} = \begin{Bmatrix} y_1 - f(x_1) \\ y_2 - f(x_2) \\ \vdots \\ y_n - f(x_n) \end{Bmatrix} \qquad \{\Delta A\} = \begin{Bmatrix} \Delta a_0 \\ \Delta a_1 \\ \vdots \\ \Delta a_m \end{Bmatrix}$$

선형최소제곱을 적용

$[[Z_j]^T[Z_j]]\{\Delta A\} = \{[Z]^T\{D\}\}$

$a_{0,j+1} = a_{0,j} + \Delta a_0$ $a_{1,j+1} = a_{1,j} + \Delta a_1$

Example) $f(x;a_0,a_1)=a_0(1-e^{-a_1x})$

X	0.25	0.75	1.25	1.75	2.25
Y	0.28	0.57	0.68	0.74	0.79

$$\frac{\partial f}{\partial a_0}=1-e^{-a_1x} \qquad \frac{\partial f}{\partial a_1}=a_0x\ e^{-a_1x}$$

$$[Z_0]=\begin{bmatrix} 0.2212 & 0.1947 \\ 0.5276 & 0.3543 \\ 0.7135 & 0.3581 \\ 0.8262 & 0.3041 \\ 0.8946 & 0.2371 \end{bmatrix} \qquad [Z_0]^T[Z_0]=\begin{bmatrix} 2.3193 & 0.9489 \\ 0.9489 & 0.4404 \end{bmatrix}$$

$$\left[[Z_0]^T[Z_0]\right]^{-1}=\begin{bmatrix} 3.6397 & -7.8421 \\ -7.8421 & 19.1678 \end{bmatrix}$$

$$\{D\}=\begin{Bmatrix} 0.28-0.2212 \\ 0.57-0.5276 \\ 0.68-0.7135 \\ 0.74-0.8262 \\ 0.79-0.8946 \end{Bmatrix}=\begin{Bmatrix} 0.0588 \\ 0.0424 \\ -0.0335 \\ -0.0862 \\ -0.1046 \end{Bmatrix} \qquad [Z_0]^T\{D\}=\begin{bmatrix} -0.1533 \\ -0.0365 \end{bmatrix}$$

$$\Delta A=\begin{Bmatrix} -0.2714 \\ 0.5019 \end{Bmatrix} \qquad \begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix}=\begin{Bmatrix} 1.0 \\ 1.0 \end{Bmatrix}+\begin{Bmatrix} -0.2714 \\ 0.5019 \end{Bmatrix}=\begin{Bmatrix} 0.7286 \\ 1.5019 \end{Bmatrix}$$

반복 계산을 통해 얻은 최종 결과

a_0 = 0.79186, a_1 = 1.6751, 잔차제곱합: 0.000662

▶ 프로그램에서는 편미분을 위해 차분방정식 사용

$$\frac{\partial f}{\partial a_k}\cong\frac{f(x_i;a_0,\cdots,a_k+\delta a_k,\cdots,a_m)-f(x_i;a_0,\cdots,a_k,\cdots,a_m)}{\delta a_k}$$

▶ 수렴이 늦고 진동이 심하며, 수렴이 보장되지 않는 단점.
비선형 최적화 기법을 통해 잔차 제곱합을 최소화하도록 매개변수를 조절

$$S_r=\sum_{i=1}^{n}\left(y_i-a_0(1-e^{-a_1x_i})\right)^2$$