情報処理論2a

8章「回帰分析」

(最小自乗法)

中島康彦

§8. 1 今日の作業ディレクトリを作る

1. % cd ⇒ ホームディレクトリへ移動
2. % mkdir chap20 ⇒ ディレクトリchap20を作成
3. % cd chap20 ⇒ ディレクトリchap20へ移動
4. % netscapeを使ってdata20をchap20へダウンロード
5. % tar xvf data20 ⇒ サンプルデータの複写

```
squares.c
squares.in
```

§8. 2 回帰モデル

実際のデータ(x_iおよびy_i)に基づく仮説

- $y=\alpha+\beta x+u$
  - y … 被説明変数
  - x … 説明変数
  - α, β … 回帰係数(未知数)
  - u … 無作為撹乱項(確率変数)

回帰係数の推定(最小自乗法の場合)

- $u_i=y_i-(\alpha+\beta x_i)$
- Σu_i^2 を最小にするαおよびβを求める

§8. 3 最小自乗法

- $\Sigma (y_i-(\alpha+\beta x_i))^2$ をα, βに関して偏微分

$$-2\Sigma(y_i-(\alpha+\beta x_i)) = 0 \quad \cdots (1)$$
$$-2\Sigma x_i(y_i-(\alpha+\beta x_i)) = 0 \quad \cdots (2)$$

- すなわち

$$N\alpha=\Sigma y_i-\beta\Sigma x_i \quad \cdots (3)$$
$$N\Sigma x_iy_i=N\alpha\Sigma x_i+N\beta\Sigma x_ix_i \quad \cdots (4)$$

- αを(4)に代入してβを求める

$$N\Sigma x_iy_i=(\Sigma y_i-\beta\Sigma x_i)\Sigma x_i+N\beta\Sigma x_ix_i$$

$$\beta = \frac{N\Sigma x_iy_i-\Sigma x_i\Sigma y_i}{N\Sigma x_ix_i-\Sigma x_i\Sigma x_i}$$

$$\alpha = (\Sigma y_i-\beta\Sigma x_i)/N$$

§8. 4 最小自乗法の具体化

```
x, y:       標本(入力)
n:          標本数(初期値0)
sum_x:      xiの累計(初期値0)
sum_x2:     xixiの累計(初期値0)
sum_y:      yiの累計(初期値0)
sum_xy:     xiyiの累計(初期値0)
a, b:       回帰係数(出力)

while (scanf("%d %d", &x, &y) == 2) {
  sum_x  += (double)x;
  sum_y  += (double)y;
  sum_x2 += (double)x*(double)x;
  sum_xy += (double)x*(double)y;
  n++;
}

b = (n*sum_xy - sum_x*sum_y)/(n*sum_x2-sum_x*sum_x);
a = (sum_y - b*sum_x)/n;
```

§8. 5 プログラム(squares.c)

```
static char RcsId[] = "$Header$";
#include <stdio.h>
main()
{
  int x, y;
  int n = 0;
  double sum_x  = 0.0;
  double sum_x2 = 0.0;
  double sum_y  = 0.0;
  double sum_xy = 0.0;
  double a, b;

  while (scanf("%d %d", &x, &y) == 2) {
    sum_x  += (double)x;
    sum_y  += (double)y;
    sum_x2 += (double)x*(double)x;
    sum_xy += (double)x*(double)y;
    n++;
  }

  b = (n*sum_xy - sum_x*sum_y)/(n*sum_x2-sum_x*sum_x);
  a = (sum_y - b*sum_x)/n;
  printf("a=%g b=%g\n", a, b);
  exit(0);
}
```

§8. 6 コンパイルと実行

1．コンパイルする.

```
% gcc squares.c -o squares
```

2．実行

```
% ./squares
2 12
3 13
e

a=10 b=1
```

3．実行

```
% ./squares
2 102
3 103
e

a=100 b=1
```

§8. 6 コンパイルと実行(続き)

4．ファイルの確認

```
% cat squares.in

13 243517440
14 243814400
15 244264960
16 244736000
19 246824960
26 251013120
33 255068160
```

x: 2000年8月1日からの経過日数(13は8月13日のデータ)
y: 経済学部ホームページ上のデータ量(バイト)

5．ファイルから入力する

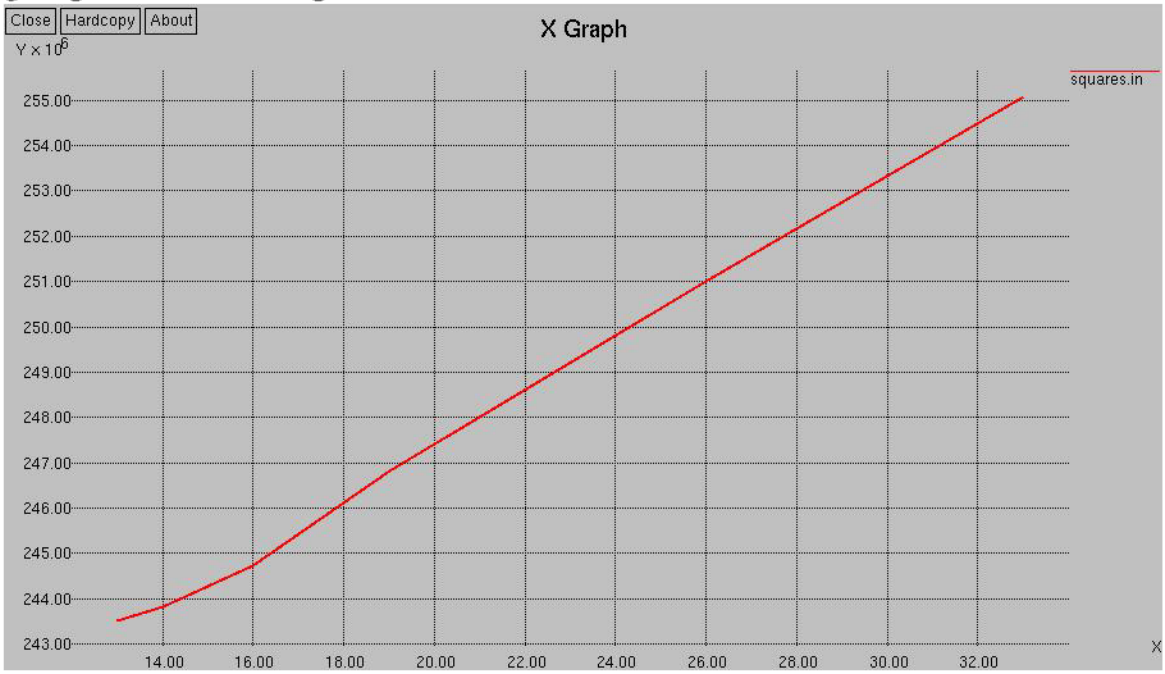
```
% ./squares < squares.in

a=2.35514e+08 b=592935
```

§8. 7 グラフ表示

X Window Systemを起動した後

▶ xgraph -lw 2 squares.in



§8. 8 例題

1日あたり何バイト増加していると考えられるか答えよ.

2000年12月31日のデータ量を推定せよ.

ヒント: 以前に作成したdays2.cを利用すると, 日数計算が簡単である.

§8. 9 今日の課題

「今日は8章と9章を1度に講義するので，今日の課題は9章分のみ」

宛先: nakashim@econ.kyoto-u.ac.jp
件名: unix2-学生番号

今日はここまで