数学実習 2011 年度第 5 回
担当：中口悦史

今週の講義メモ：統計学の復習，Pascal のファイル入出力と関数と擬似乱数の利用

Pascal のファイル入出力

ファイルへのアクセス

- ファイル用変数の宣言：text 型
- ファイルを割り当てる： assign 文
  assign(text 型変数，’ファイル名’)
- ファイルを閉じる： close 文
  close(assign 文で割り当てた text 型変数)

ファイルからのデータの読み込み

- ファイルの先頭に読み込み位置を合わせる： reset 文
  reset(assign 文で割り当てた text 型変数)
- ファイルの内容を読み込む： read 文，readln 文
  read(text 型変数，変数並び)： 今の読み込み位置から変数並びに応じてデータを読み込む
  readln(text 型変数，変数並び)： さらに，その後のデータを読み飛ばして，次の行頭へ
- ファイルの中の，現在の位置を知る
  - eof(text 型変数)： 読み込み位置がファイルの終わり (end of file) にあれば true，そうでなければ false
  - eoln(text 型変数)： 読み込み位置がどこかの行末 (end of line) にあれば true，そうでなければ false

ファイルへのデータの書き出し

- 書き込むためにファイルを空にする： rewrite 文
  rewrite(assign 文で割り当てた text 型変数)
- ファイルにデータを書き込む： write 文，writeln 文
  write(text 型変数，出力並び)： 今の位置から出力並びに応じてデータを書き込む
  writeln(text 型変数，出力並び)： さらに，その後に改行して，次の行頭へ

ファイル入出力に関する注意

プログラム先頭の program 文のカッコ () 内にも，入出力ファイルに対応する変数名を並べないといけない。

Pascal の関数

作り方： プログラム本文の前の宣言・定義部で，プログラム本体と似た書き方で定義する。

```
function 関数名 ( 仮引数並び ): 結果型 ;
begin 関数本体 end;
```

本体の記述の中に，関数名への代入文が必要：呼び出された際に返す値を決める。

名前： プログラム名，変数・配列名と同じルールに従って，既定のものと同じでなければ，自由に決めていい。

仮引数： 引数名:型名 をセミコロンで区切って並べる。値の受け渡しに必要なパラメータ。内部変数としても使える。

使い方（呼出し）： 名前に続けて，渡すデータを引数並びに対応した順に，並べて書く。

Pascal の擬似乱数の利用

擬似乱数とは

乱数らしく見える数列を，適当な生成ルールを使って，コンピュータで再現するもの。

擬似乱数の呼出しと初期化

- `random`： 0.0 以上 1.0 未満の実数の一様分布に従う擬似乱数。real 型の値を取る関数。
- `Randomize`： 擬似乱数生成を初期化する文。

統計学

代表的な確率分布

二項分布 $B(n,p)$ 成功確率 p の試行を n 回行ったとき，k 回が成功する確率。平均 np，分散 $np(1-p)$

$$f(k) = \begin{cases} {}_nC_k\, p^k (1-p)^{n-k}, & k = 0,1,\ldots,n \\ 0, & \text{その他} \end{cases}$$

一様分布 $U(a,b)$ a から b までの値を等確率で取る。平均 $\frac{a+b}{2}$，分散 $\frac{(b-a)^2}{12}$

$$f(x) = \begin{cases} \frac{1}{b-a}, & a \le x < b \\ 0, & \text{その他} \end{cases}$$

正規分布 $N(\mu,\sigma^2)$ 平均 μ，分散 σ^2，標準偏差 σ

$$f(x) = \frac{1}{\sqrt{2\pi}\,\sigma}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

母集団と標本

母集団： 統計の対象とし，同じ統計に従うと仮定する，集団の全体
標本 ： 母集団から抽出された（とみなされる）個体群

標本の特性値 標本数 n，確率変数 X に対する標本値 $\{x_1, x_2, \ldots, x_n\}$ とする。
中央値（メディアン）： 標本値を昇順に並べたときに中央にくる値
平均 （標本平均） ： $\bar{X} = \{\sum_{i=1}^{n} x_i\}/n$
分散 （標本分散） ： $V = \left\{\sum_{i=1}^{n} (x_i - \bar{X})^2\right\}/n$
不偏分散（母分散の不偏推定値）： $U = \left\{\sum_{i=1}^{n} (x_i - \bar{X})^2\right\}/(n-1)$
母平均 μ からの分散 ： $S^2 = \left\{\sum_{i=1}^{n} (x_i - \mu)^2\right\}/n$

推定と検定 標本値から母集団の特性値を調べる。推論の方向が逆向き。
点推定 標本の特性値をそのまま母集団の特性値として採用する。
区間推定 信頼度 $(1-\alpha)$（危険率 α）で母集団の特性値を含む区間 (信頼区間) を，標本の特性値から計算する。
仮説検定 母集団の特性値に関する仮説 (帰無仮説) を立て，それが有意水準 (p-value) α で棄却されうるかどうか，標本の特性値を元に検討する。

母平均 μ の区間推定・仮説検定 母集団が正規分布に従うと仮定したとき

母分散 σ^2 が既知の場合：$\dfrac{\bar{X}-\mu}{\sqrt{\sigma^2/n}}$ が標準正規分布 $N(0,1)$ に従うことを用いる。

母分散 σ^2 が未知，かつ大標本の場合：$\dfrac{\bar{X}-\mu}{\sqrt{U/n}}$ が標準正規分布 $N(0,1)$ に従うことを用いる。

母分散 σ^2 が未知，かつ小標本の場合：$\dfrac{\bar{X}-\mu}{\sqrt{U/n}}$ が自由度 $(n-1)$ の t 分布に従うことを用いる。

今週のサンプルプログラム

テキストファイルはすべてプログラムファイルと同じフォルダ (計算機実習室のパソコンの場合は，マイドキュメントあるいはその中の Pascal フォルダ) に置いてください。また読み込み側のファイル (data1.txt など) は TeraPad などテキストエディタで適切なデータを入力して，作成しておいてください。

(5a) [Renshu5a.pas] テキストファイル data1.txt の内容をそのまま表示

```
program Renshu5a( input, output, file1 );
var file1: text;
    c: char;
begin
  assign( file1, 'data1.txt' );     { ファイルを file1 に割り当てる }
  reset( file1 );                   { file1 から読み込む準備をする }
  while not eof( file1 ) do begin       { file1 の終わり (eof) に来るまで }
    while not eoln( file1 ) do begin    { file1 の各行の行末 (eoln) まで }
      read( file1, c ); { file1 から 1 文字を読み込んで c に保存する }
      write( c )        { c の内容を表示する }
    end;
    readln( file1 );    { file1 の読み込み位置を改行する }
    writeln()           { 表示を改行する }
  end;
  close( file1 )    { file1 を閉じる }
end.
```

テキストファイル data1.txt の例：

```
0.1 1.25
0.2 1.38
0.3 1.61
0.4 1.75
0.5 2.01
```

(5b) [Renshu5b.pas] プログラムで生成された数値データを，テキストファイル data2.txt へ書き出す

```
program Renshu5b( input, output, file2 );
var file2: text;
    x: real;
    n, i: integer;
begin
  assign( file2, 'data2.txt' ); { ファイルを file2 に割り当てる }
  rewrite( file2 );             { file2 に書き出す準備をする }
  n := 100;
  for i := 1 to n do begin
    x := i * 0.1;               { データ生成 }
    writeln( file2, x );        { x の値を file2 に書き出す }
    writeln( x )                { x の値を表示する }
  end;
  close( file2 )    { file2 を閉じる }
end.
```

(5c) [Renshu5c.pas] キーボードから数値データを入力して，テキストファイル data3.txt へ書き出す

```
program Renshu5c( input, output, file3 );
var file3: text;
    n: integer;
begin
  assign( file3, 'data3.txt' ); { ファイルを file3 に割り当てる }
  rewrite( file3 );             { file3 に書き出す準備をする }
  repeat
    read( n );              { キーボードから 1 個の整数を読み込み n に保存する }
    writeln( file3, n )     { n の値を file3 に書き出す }
  until ( n = -1 );         { n の値が-1 だったら読み込みを止める }
  close( file3 )    { file3 を閉じる }
end.
```

(5d) [Renshu5d.pas] テキストファイル data4.txt から，各行の先頭に書かれてある 2 個の数値データだけを読み込んで，別のテキストファイル data5.txt へ書き出して，同時に表示

```
program Renshu5d( input, output, file1, file2 );
var file1, file2: text;
    x, y: real;
begin
  assign( file1, 'data4.txt' ); { ファイルを file1 に割り当てる }
  reset( file1 );               { file1 から読み込む準備をする }
  assign( file2, 'data5.txt' ); { ファイルを file2 に割り当てる }
  rewrite( file2 );             { file2 に書き出す準備をする }
  while not eof( file1 ) do begin   { file1 の終わり (eof) に来るまで }
    readln( file1, x, y );
      { file1 の各行から先頭の 2 個の実数値を読み込み x と y に保存する }
    writeln( file2, x, ' ', y );    { x と y の値を file2 に書き出して改行する }
    writeln( x, ' ', y );   { x と y の値を表示して改行する }
  end;
  close( file1 );   { file1 を閉じる }
  close( file2 )    { file2 を閉じる }
end.
```

テキストファイル data4.txt の例：data1.txt と同様。

(5e) [Renshu5e.pas] テキストファイル data6.txt から，各行の先頭に書かれてある 1 個の数値データだけを読み込んで，配列に保存して，データの個数とその和を計算して，表示

```
program Renshu5e( input, output, file3 );
var file3: text;
    a: array [1..100] of real;  { 扱うデータ数より十分多い配列を用意 }
    b: real;
    ndata, i: integer;
begin
  assign( file3, 'data6.txt' ); { ファイルを file3 に割り当てる }
  reset( file3 );               { file3 から読み込む準備をする }

  { file3 からデータを読み込む }
  i := 0;
```

```
  while not eof( file3 ) do begin   { file3 の終わり (eof) に来るまで }
    i := i + 1; { 行を数える = データの個数を数える }
    readln( file3, a[i] )
      { file3 の各行から先頭の 1 個の実数値を読み込み a[i] に保存する }
  end;
  ndata := i;   { 最新の i の値 = データの個数を ndata に保存 }
  writeln( 'データの個数：', ndata );

  { データの和 }
  b := 0;
  for i := 1 to ndata do b := b + a[i];
  writeln( 'データの和：', b );

  close( file3 )    { file3 を閉じる }
end.
```

テキストファイル data6.txt の例：data1.txt と同様。

(5f) [Renshu5f.pas] 擬似乱数生成関数 random の使用例 (1): 一様乱数の生成

```
program Renshu5f( input, output );
var r: real;
    n, i: integer;
begin
  n := 100; { 生成する乱数の個数 }
  Randomize;
  for i := 1 to n do begin
    r := random;    { r は一様分布 U(0,1) に従う乱数 }
    writeln( r )
  end
end.
```

(5g) [Renshu5g.pas] 擬似乱数生成関数 random の使用例 (2): f 面サイコロ関数

```
program Renshu5g( input, output );
var n, f, k, i: integer;

{ m 面サイコロ関数 : 1 から m までの値を等確率で取る乱数}
function dice( m: integer ): integer;
  begin   dice := round( random * m + 0.5 )   end;
    { dice := trunc( random * m ) + 1 と書いても同じ }

begin
  f := 6;   { サイコロの面の数 = 乱数の取りうる値の数 }
  n := 100; { 生成する乱数の個数 }
  Randomize;
  for i := 1 to n do begin
    k := dice( f );     { k := round( random * f + 0.5 ) と書いても同じ }
    writeln( k )
  end
end.
```

(5h) [Renshu5h.pas] 擬似乱数生成関数 random の使用例 (3): 正規乱数関数 (Box-Muller 法簡易版)

```
program Renshu5h( input, output );
var x, mu, sigma: real; n, i: integer;

{ Normal RANDom number generated by Box-Muller scheme }
function nrand: real;
  var t, u: real;
  begin
    t := sqrt( -2 * ln( 1.0-random ) );
    u := 2 * PI * random;
    nrand := t * cos( u )
  end; { end of definition of nrand }

begin
  mu := 33;     { 乱数の平均 }
  sigma := 3;   { 乱数の標準偏差 }
  n := 100;     { 生成する乱数の個数 }
  Randomize;
  for i := 1 to n do begin
    x := sigma * nrand + mu;    { x は正規分布 N(mu,sqr(sigma)) に従う乱数 }
    writeln( x )
  end
end.
```

(5i) [Renshu5i.pas] 擬似乱数生成関数 random の使用例 (4): ベルヌーイ試行

```
program Renshu5i( input, output );
var p, r: real;
    m, n, c, i, j: integer;
begin
  p := 0.25; { 成功する (コインの表が出る) 確率 }
  m := 100;  { 実験 (標本) の回数 }
  n := 1000; { 1 回の実験に含まれる試行の回数 }
  Randomize;
  for j := 1 to m do begin { 実験 (標本) }
    c := 0; { 成功の回数を数える }
    for i := 1 to n do begin { 試行 }
      r := random;  { r は一様乱数 U(0,1) に従う乱数 }
      if ( r < p ) then c := c + 1 { r<p となる確率は p -> 成功とみなす }
    end;
    writeln( c )    { c は二項分布 B(n,p) に従う乱数 }
  end
end.
```