

0.8 ジュニア版 データのグラフ化

今までは関数を使う、または、自ら作成してデータを分析することのさわりに取り組みましたが、ここではデータをグラフ化することをやってみましょう。まずは、時系列データをプロットすることから始めて、平均の応用である移動平均というものをとってそれらをプロットさせましょう。また、時系列ではない度数を見るグラフとして、ヒストグラムを利用して度数分布や累積度数分布を見てみましょう。最後に、ヒストグラムをならしてスムーズにしたようなカーネル密度関数について少しふれましょう。

時系列データ

原始的に Copy&Paste だけで株価データを GAUSS のプログラム上に取り入れるには次のようになります。

【手順】

- 1) Yahoo Japan 等で公開されている株価の時系列をカーソルで黒くして Copy する
- 2) Excel などのワークシートソフト上に Paste する
- 3) いくつかある系列のうち、終値の列だけを選択して Copy する
- 4) もう 1 枚ワークシートを開いてそこに Paste する
- 5) 終値 1 列のデータをテキスト形式で保存する
- 6) 保存されたデータを開いて Copy&Paste して GAUSS のエディター上で扱う

以下では、まずデータをこれまでと同じようにプログラム上で読み込んでから、この場合一番上の行が最近で、一番下が一番遠い過去になっていますから、順序を逆にします。そうしてから、x 軸を任意の続き番号（例えば 1 から 50 まで）、y 軸をこれら株価の終値として、x y グラフで描いてみます。

プログラム

```
new;
```

```
cls;
```

```
let data[50,1]=
```

```
60
```

```
61
```

```
60
```

```
62
```

```
60
```

```
63
```

```
64
```

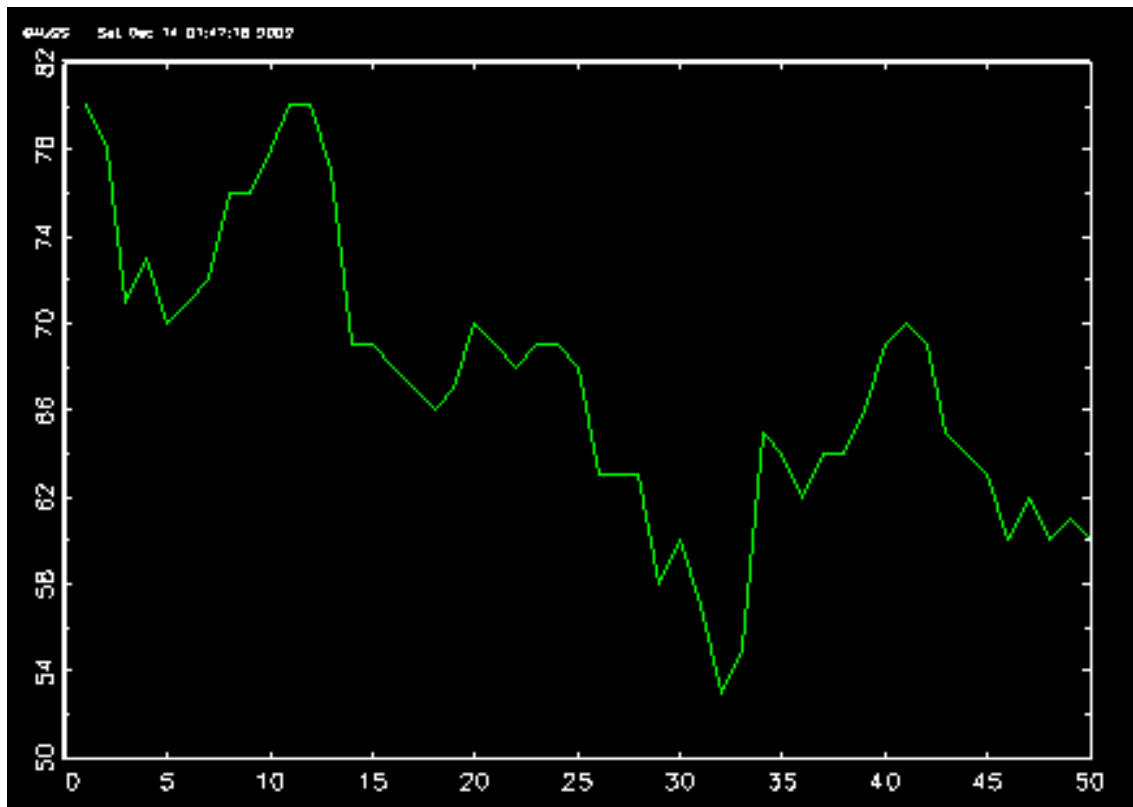
```
65
```

```
69
```

70
69
66
64
64
62
64
65
55
53
57
60
58
63
63
63
68
69
69
68
69
70
67
66
67
68
69
69
77
80
80
78
76
76
72
71

```
70  
73  
71  
78  
80  
;  
data=rev(data);
```

```
library pgraph;  
graphset;  
xy(seqa(1,1,50),data);  
グラフ表示
```



プログラム各行の説明

1 行目 メモリを初期化する

2 行目 画面をクリアする

3 ~ 5 4 行目 変数 data に 50×1 のディメンションでセミコロンまでのスペースに書かれているデータを取り込んで入れる

5 5 行目 組込み関数 rev を用いて、変数 data を上と下を逆にする

- 5 7 行目 グラフを描く機能の入った pgraph というライブラリを呼び出す
- 5 8 行目 ライブラリ pgraph の各種設定のグローバル変数を初期化する
- 5 9 行目 x 軸に 1 から 1 ステップの 50 個のシーケンスを割り当て、y 軸には変数 data の内容を割り当てて、x y グラフで描く

前半部分のプログラム上でのデータの取り込みはこれまでと同じ形式です。株式データなどは直近のものが上にきていることが多いので、その場合には、上（列の 1 行目）が古くて、下（列の最終行）が新しくするように上下を反転させるのには、組込み関数 rev を使いその中に data を入れて変換されたものをあらためて data と置き直します（置き直す変数名は違うものでもかまいません）。各種グラフを描くには、GAUSS では通常の組込み関数では対応していなくて、外部のライブラリという形の組込み関数群を呼び出して利用可能な状態にしておいてから、例えば x y グラフなどの描画を行ないます。少し面倒ですが、

```
library pgraph;  
graphset;
```

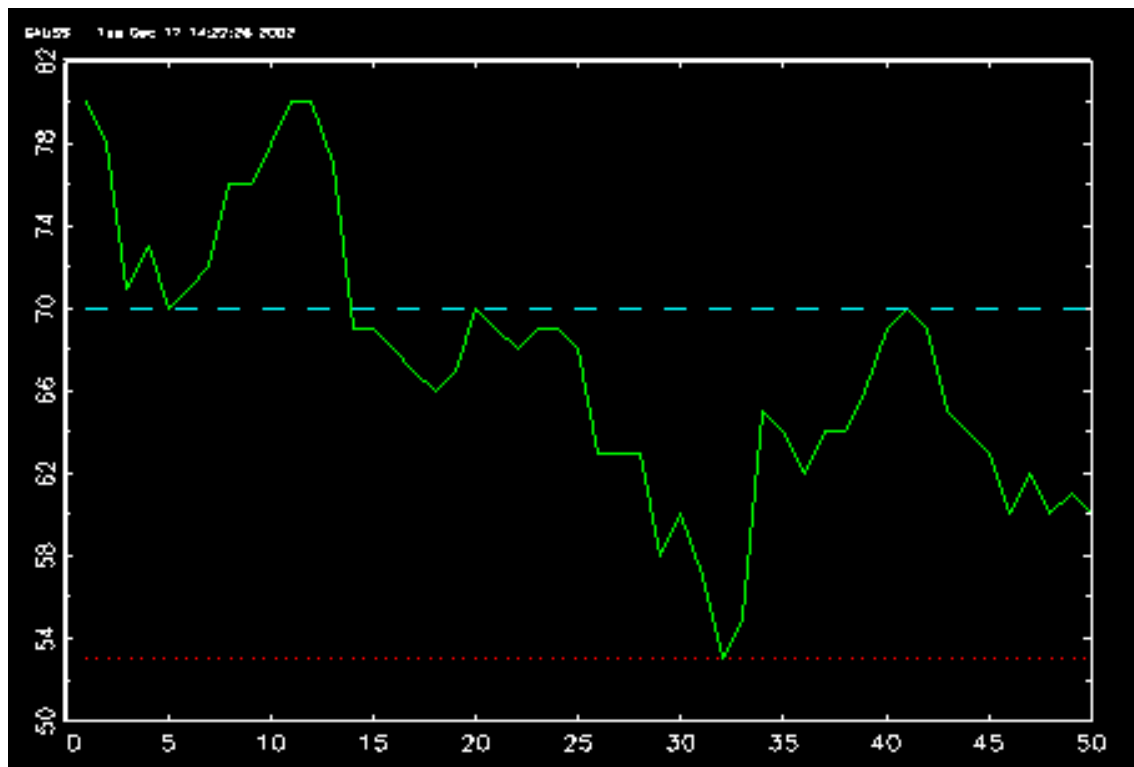
の 2 つの命令が必要なだけです。pgraph という名前のグラフ描画関係の組込み関数がたくさんはあったライブラリを呼び出して利用可能にして、次にその各種設定に使われるグローバル変数と呼ばれるものを初期化します。これだけです。この 2 つは覚えておいてください。これらがなければ x y グラフ他各種のグラフを書こうとしても動かないか、または画面が意図しないものになります。最後に、x y グラフで、その第 1 要素に x 座標となるべき列ベクトルを入れます。ここでは仮に 1 から 50 までのシーケンスとしましょう。その第 2 要素には y 座標となる変数 data の内容を入れます。GAUSS は自動描画させるパッケージソフトウェアではなくて、設定もすべて自分でしなければなりません。x 軸と y 軸の長さ（それぞれの列の行数）は必ず一致する必要があります。この場合 50 で共通です。以上がグラフの中心となる x y グラフの描き方の基礎となる部分です。

次に、上の時系列グラフに基準線を入れてやりましょう。例えば、70 という上限線と 53 という下限線をグラフに書き入れたいときには、

```
xy(seqa(1,1,50),data~70*ones(50,1)~53*ones(50,1));
```

50 × 1 のすべて 1 からなる列ベクトルを利用して、それに任意の数をかけてやることで、同じ数からなる列ベクトルを作成します。そうして作成した列ベクトルを y 軸側に 2 列目の変数、3 列目の変数...という具合に水平方向に ~ の記号でマージしてやれば、もとのデータも含めて 3 列からなる y 軸側のデータ行列ができます。これによって、x 軸側は共通として、3 つの系列のグラフが一度に同じ画面に描かれます。基準線は 1 本であっても、はたまた 3 本以上であっても同じ要領でできます。

グラフ表示



株価グラフでよく使う5日間および25日間移動平均を計算して、基準線と同じように元データに水平方向にマージすることでグラフ表示してやりましょう。

プログラム

(同上の部分)

```
data=rev(data);
```

```
library pgraph;
```

```
graphset;
```

```
xy(seqa(1,1,50),data~ma5(data)~ma25(data));
```

```
proc ma5(x);
```

```
  local m,i,j;
```

```
  m=zeros(rows(x),cols(x));
```

```
  i=5;
```

```
  do while i<=rows(x);
```

```
    j=1;
```

```
    do while j<=5;
```

```
      m[i,.]=m[i,.]+x[i-5+j,.];
```

```

        j=j+1;
    endo;
    i=i+1;
end;
m=m/5;
m[1:4,.]=miss(m[1:4,.],0);
retp(m);
endp;

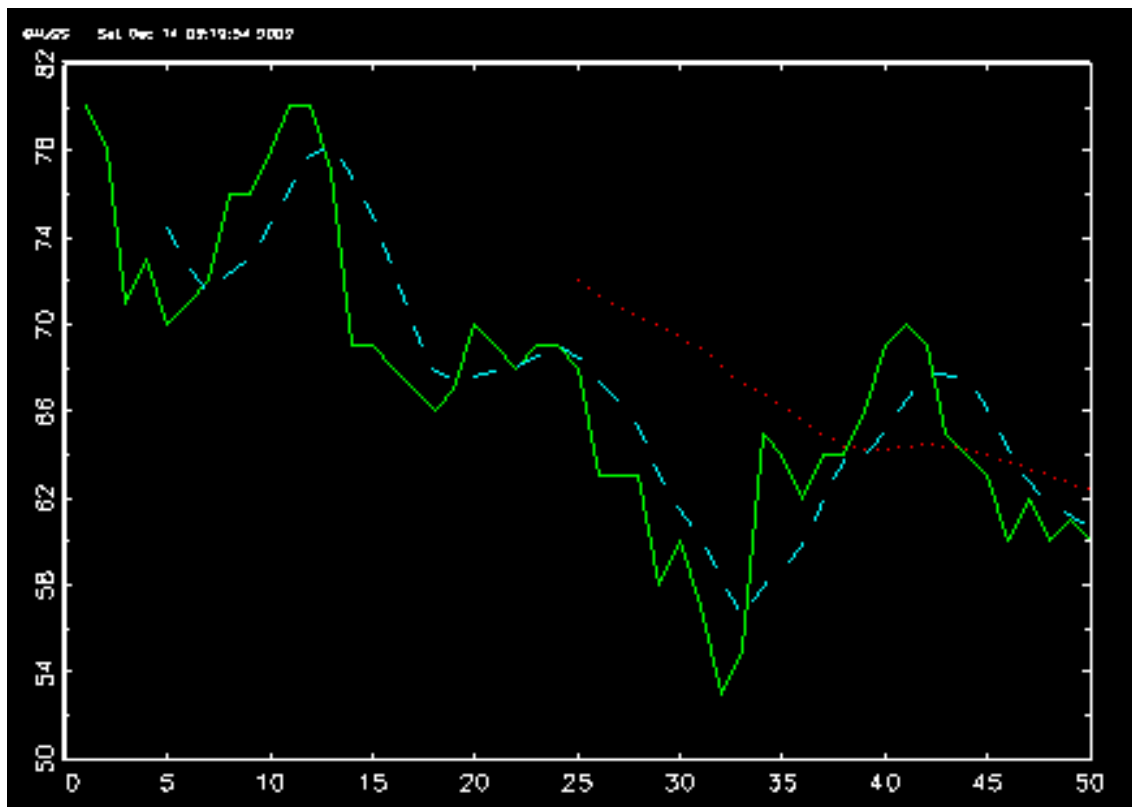
```

```

proc ma25(x);
    local m,i,j;
    m=zeros(rows(x),cols(x));
    i=25;
    do while i<=rows(x);
        j=1;
        do while j<=25;
            m[i,.]=m[i,.]+x[i-25+j,.];
            j=j+1;
        end;
        i=i+1;
    end;
    m=m/25;
    m[1:24,.]=miss(m[1:24,.],0);
    retp(m);
endp;

```

グラフ表示



ボリンジャーバンド

移動平均の応用として、単純移動平均のまわりに を求めて、例えば 20 期の移動平均に対して逐次的に

$$\text{Upper Band} = \text{MA} + 2$$

$$\text{Lower Band} = \text{MA} - 2$$

なる上限と下限を求めるものです。ここで、 は GAUSS の標準機能の $n - 1$ で割るものではなくて、 n で割るバージョンを使うことにします。

プログラム

(同上の部分)

```
data=rev(data);
```

```
library pgraph;
```

```
graphset;
```

```
xy(seqa(1,1,50),data~Bollinger(data));
```

```
proc Bollinger(x);
```

```
  local m,i,j,sig;
```

```
/* MA20 calculation */
```

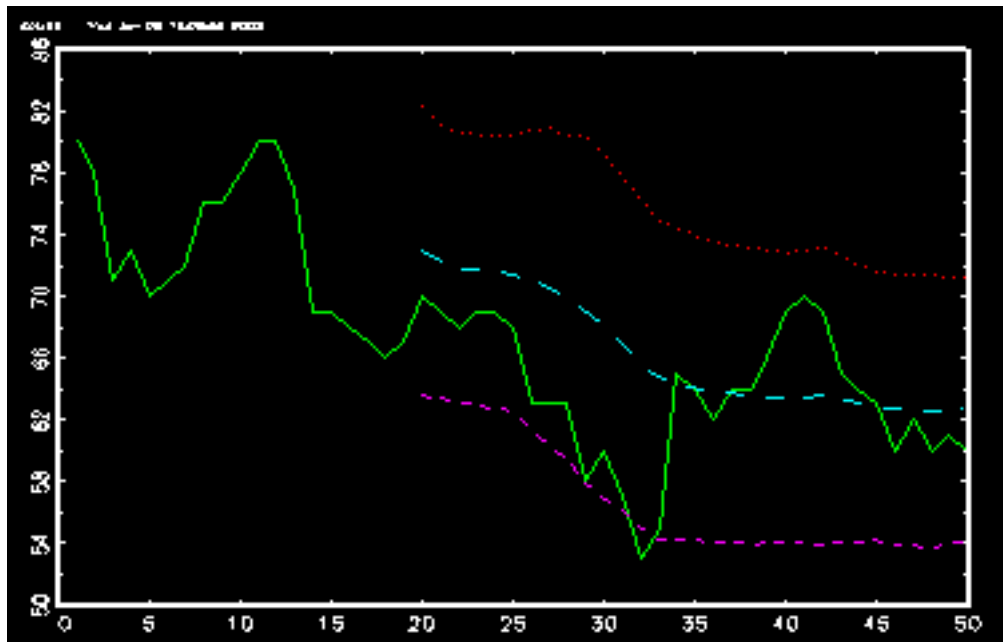
```

m=zeros(rows(x),cols(x));
i=20;
do while i<=rows(x);
    j=1;
    do while j<=20;
        m[i,.]=m[i,.]+x[i-20+j,.];
        j=j+1;
    endo;
    i=i+1;
endo;
m=m/20;
m[1:19,.]=miss(m[1:19,.],0);
/* sigma calculation */
sig=zeros(rows(x),cols(x));
i=20;
do while i<=rows(x);
    sig[i]=sd(x[i-19:i,.]);
    i=i+1;
endo;
retp(m~(m+2*sig)~(m-2*sig));
endp;

proc sd(x);
    local n,sum,i,xbar,s2,s;
    n=rows(x);
    sum=0;
    i=1;
    do while i<=n;
        sum=sum+x[i,.];
        i=i+1;
    endo;
    xbar=sum/n;
    s2=sumc((x-xbar)^2)/n;    @ Divided by n. @
    s=sqrt(s2);
    retp(s);
endp;

```


グラフ表示



上のプログラムの procedure の前半はこれまでの 5 期と 25 期の移動平均のプログラムを 20 に修正しただけです。後半は、その下に置いている sd という n で割るバージョンの を求める proc を用いて、 $i - 19$ から i までの 20 データの を逐次的に計算したものです。最後に、移動平均の系列 m に 2 倍した を足し引きしたものをマージして、移動平均と上限と下限の 3 列の変数行列をリターンとして外部に返しています。冒頭で、もとの系列データとマージして、合計 4 つの系列を x y グラフで表示させています。

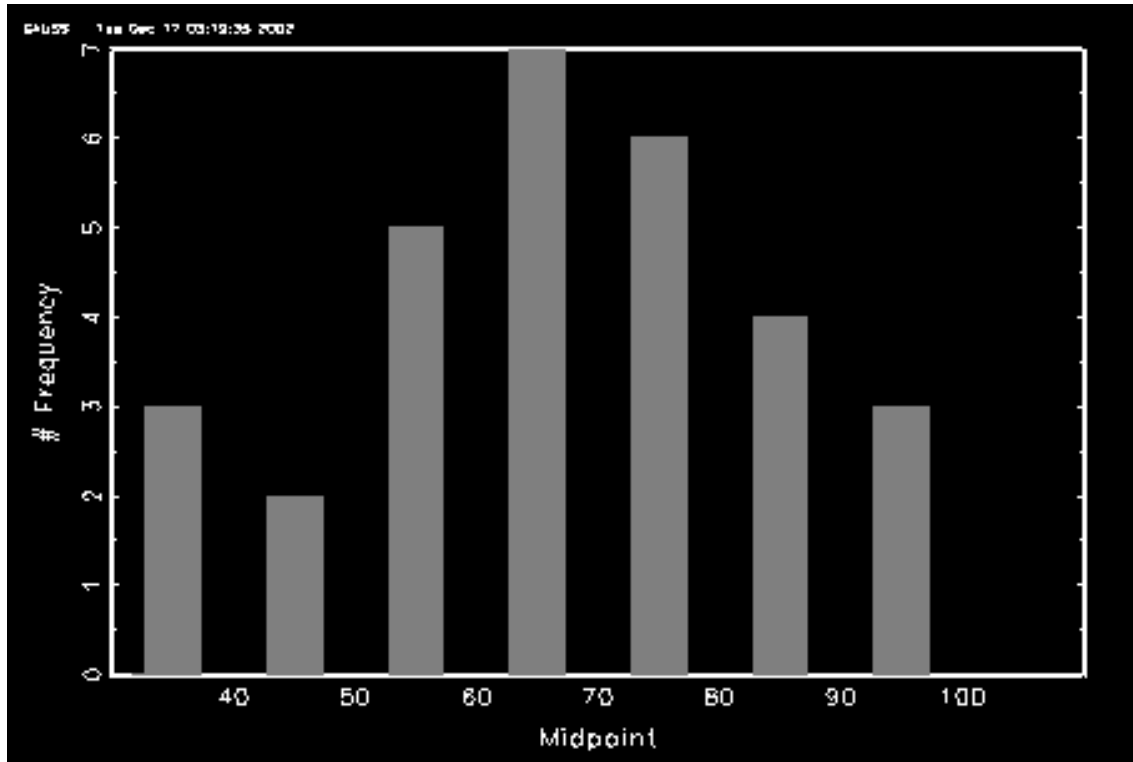
度数分布

プログラム

```
new;  
cls;  
data={90,62,38,81,55,40,78,100,79,65,  
      60,97,62,83,70,48,90,58,74,68,  
      75,60,48,70,95,65,77,55,40,73};
```

```
library pgraph;
graphset;
v={30,40,50,60,70,80,90,100};
call hist(data,v);
```

グラフ表示



上のヒストグラムは縦軸が度数ですが、パーセントにするには hist のかわりに histp を使い全く同じようにグラフ化します。v のところを各階級の区切り値からなるベクトルではなくて、ただ単に 7 とか 8 とか数値を 1 つ入れると、その数の階級からなるヒストグラムが自動的にグラフ化されます。

プログラム

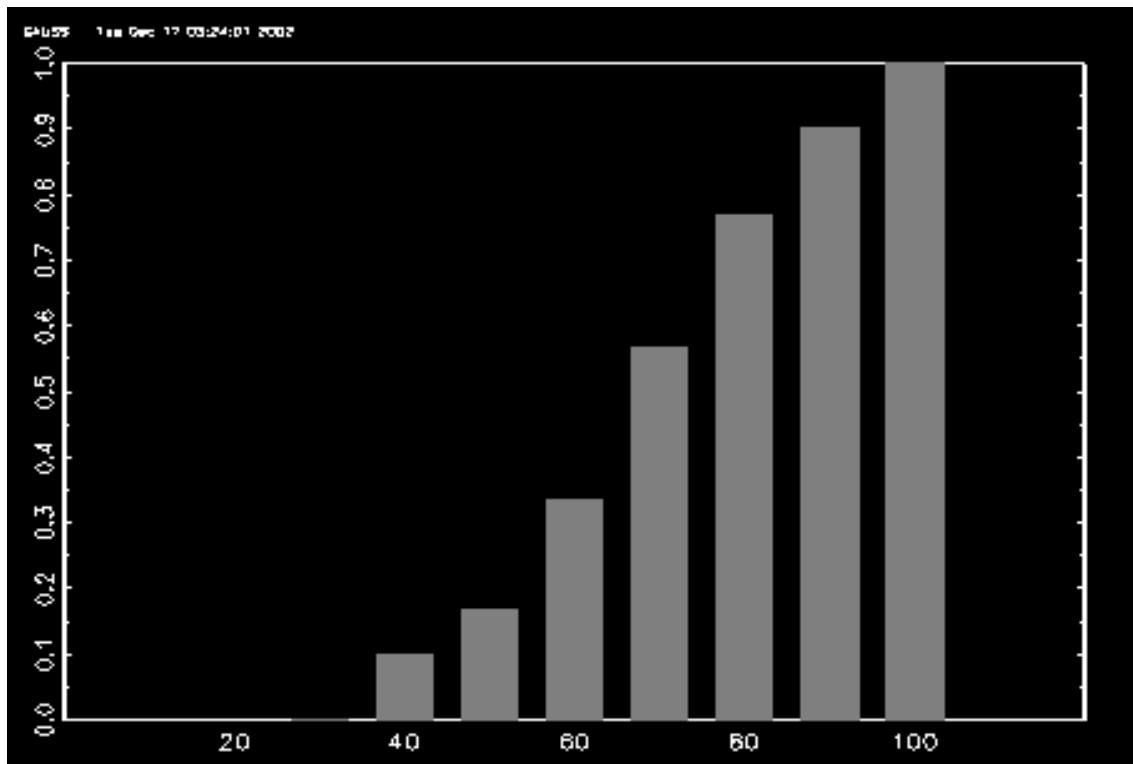
```
new;
cls;
data={90,62,38,81,55,40,78,100,79,65,
      60,97,62,83,70,48,90,58,74,68,
      75,60,48,70,95,65,77,55,40,73};
library pgraph;
```

```

graphset;
pqgwin many;
v={30,40,50,60,70,80,90,100};
{b,m,freq}=hist(data,v);
histf(cumsumc(freq)./sumc(freq),v);

```

グラフ表示



カーネル密度

プログラム

```

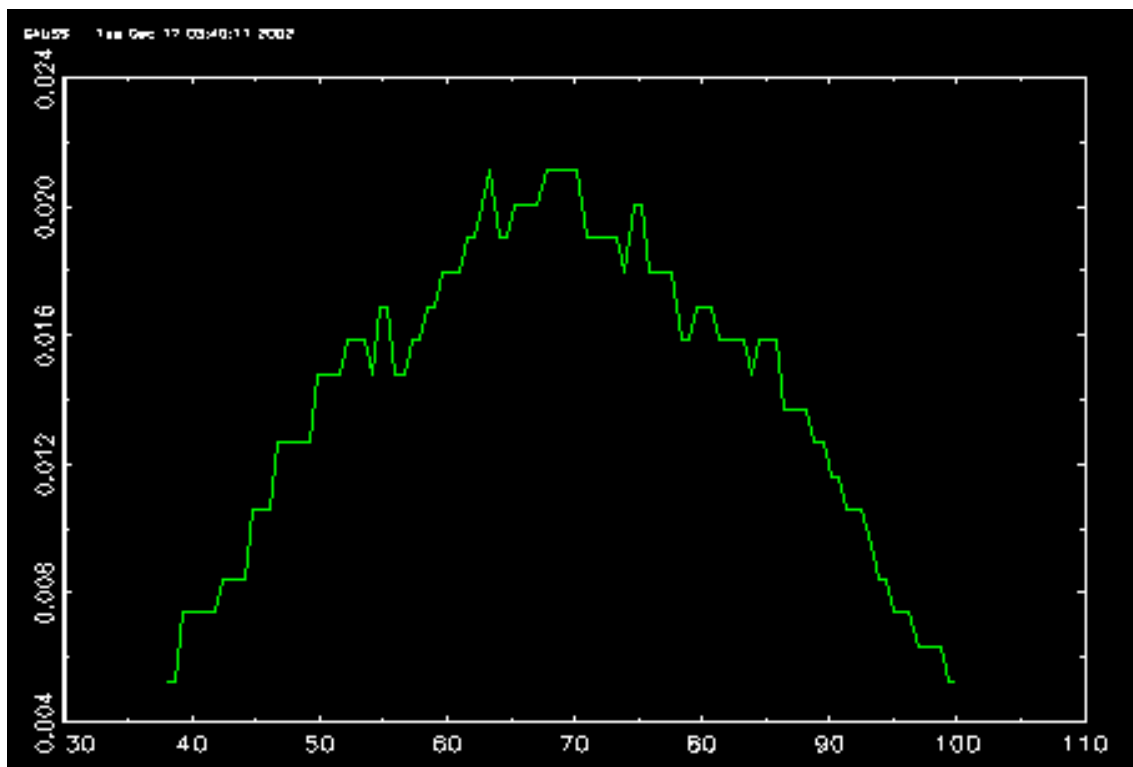
new;
cls;
data={90,62,38,81,55,40,78,100,79,65,
      60,97,62,83,70,48,90,58,74,68,
      75,60,48,70,95,65,77,55,40,73};

```

```
call plotdens(data);
```

```
proc plotdens(x);  
    local n,h,points,kern,i;  
    n=rows(x);  
    x=sortc(x,1);  
    h=1.059*stdc(x)*n^(-0.2)*1.740;      @ One of the ways to get a right band h @  
    points=seqa(x[1],(x[n]-x[1])/100,101);  
    kern=zeros(101,1);  
    i=1;  
    do while i<=101;  
        kern[i]=ukernel(points[i],x,h);  
        i=i+1;  
    endo;  
    library pgraph;  
    graphset;  
    xy(points,kern);  
    retp(kern);  
endp;  
  
proc ukernel(x0,x,h);  
    local u,kv;  
    u=(x0-x)/h;  
    kv=0.5*(abs(u).<=1);  
    retp( sumc(kv/(rows(x)*h)) );  
endp;
```

グラフ表示



練習問題

【問 1】

Yahoo Japan 等で百円前後の株価の銘柄について、最初に示した取り込み方で GAUSS 上に置いた上で、そのグラフを描こう。

【問 2】

都道府県別になっている政府統計を何か使って、相対度数分布と累積度数分布の 2 つのグラフを描いてみよう。

【問 3】

カーネル密度の計算では簡単化のためフラットな一様カーネルが用いられた。これをガウスカーネルで描くとどうなるだろうか（ヒント：一番下に置く `ukernel` は下記の `gkernel` に変更する。また、1.740 の定数がついているところは、1.740 を 1 に変更する）

```
proc gkernel(x0,x,h);
  local u,kv;
  u=(x0-x)/h;
```

```
kv=pdfn(u);  
retp( sumc(kv/(rows(x)*h)) );  
endp;
```

発展

本編 [1.4](#) [1.5](#) [1.6](#)