

1.5 グラフィックス表示（入門編その2）

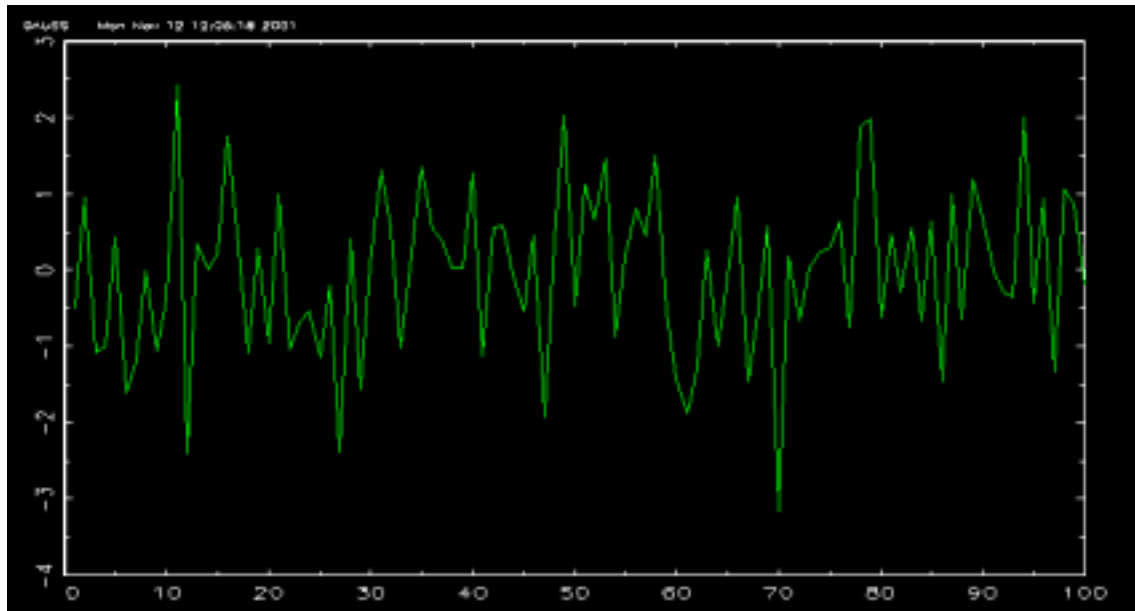
ver. 0.1

引き続きグラフを描く作業の入門を続けます。今回は `library pgraph;` で `pgraph` ライブラリーをプログラム上に呼び出して、`graphset;` によって、そこで使われるグローバル変数をリセットする。そして、`pgraph` 内の `xy(x,y);` という2次元のグラフの関数でグラフを描く作業をしました。さて、グラフのプロットなしのラインが `xy` グラフのデフォルトであるなど、少し戸惑うところがあったかもしれません。しかしながら、ラインのみが `xy` グラフのデフォルトであることは、次の例から、なぜそうなのかがわかるはずです。

プログラム

```
new; cls;  
library pgraph;  
graphset;  
y=rndn(100,1);  
x=seqa(1,1,100);  
xy(x,y);
```

グラフ表示



上のように、ファイナンシャル関係のようなグラフができます。ここで、`rndn(100,1)` は以前にも出てきたように `standard normal` な分布にふるまう乱数 100 個のデータを 1 列分発生させるもの。Seed が設定されていませんから、GAUSS 起動時からの時間が seed になって、その時々によって、データは変化します。新しく登場した `seqa(1,1,100)` は、グラフを

描く際によく使われるシーケンスの列ベクトルを生成する GAUSS の組込み関数の 1 つで、1 からステップ 1 ずつ 1 0 0 個の列ベクトルを生成します。すなわち、

```
1  
2  
.  
.  
100
```

という縦ベクトルデータを生成します。これを x 軸に使い、standard normal の乱数 100 個の縦ベクトルデータを y 軸に使い、1 対 1 で上から対応させます。それを xy(x,y); で 2 次元グラフにしたものです。もちろん、乱数 y のところは、以前にやったデータファイルの load という形で、実際のデータを取り込んで、それに 1 対 1 で seqa で作成した値を対応させて xy グラフを描くこともできます。見てもらってわかるように、ファイナンシャルデータのようなラインのみのグラフとなるのが xy グラフのデフォルト値です。これは、GAUSS が WINDOWS の前身の Dos の時代からファイナンシャルデータを使った計量経済学者の間で頻繁に使われてきた歴史的背景があって、xy グラフのデフォルトも、そのままそれらの人々が使いやすいようにラインのみのグラフの設定になっているのです。

ポイント シーケンスの列ベクトルを生成するには

seqa(始めの値,ステップの間隔,全体の個数)

例 a=seqa(0,1,1000)

この場合、0 から 1、....999 までの列 (縦) ベクトル

ここで seqa の a は、additive の a の意味です。同じような組込み関数に seqm というものもあって、その m は multiplicative の m で、第 2 要素の数だけ倍数していくものです。たとえば、seqm(1,2,100)であれば、1,2,4,8,16,32,64,128,...と 1 から 2 倍ずつ 1 0 0 個のシーケンスの列ベクトルを生成します。これも GAUSS 内部の組込み関数です。

ポイント 倍数シーケンスの列ベクトルを生成するには

seqm(始めの値,倍数,全体の個数)

例 a=seqm(1,10,10)

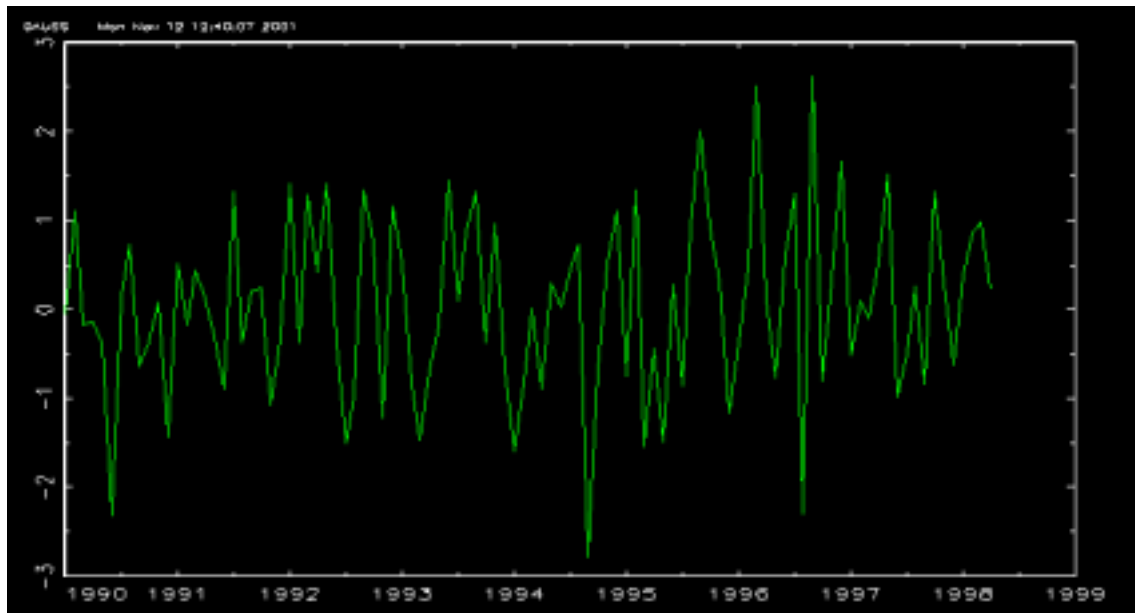
この場合、1 から 10,100,1000,....,10 の 9 乗までの列 (縦) ベクトル

1 から 1 刻みで 100 個というのは、実施のデータでは x 軸には非現実的です。そこで、ひとひねりして、1990 年から 1/12 年ごとに 100 月分とすると考えるとどうでしょうか。始めの値に 1990 を、1 ではなくて 1/12 という分数を間隔に、そして 100 個分です。

プログラム

```
new; cls;  
library pgraph;  
graphset;  
y=rndn(100,1);  
x=seqa(1990,1/12,100);  
xy(x,y);
```

グラフ表示



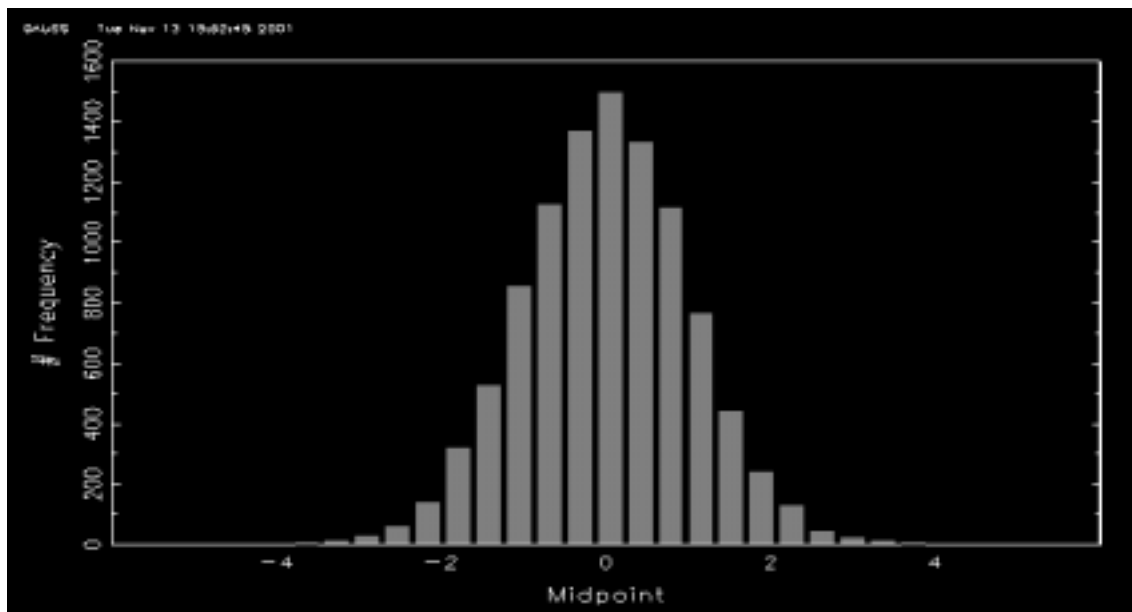
上のように、きれいに x 軸が年代になりました。十分にデータの数が大きければ、たいいていの場合、x 軸の各ラベルは 1 年おきまたは数年おき刻みに自動的にになります。分数 $1/12$ を計算して小数を入れる必要はありません。そのまま直接、seqa の第 2 要素に $1/12$ を入れば、月次刻みになります。4 半期データの場合には $1/4$ を入ればよいでしょう。

x y の 2 次元グラフ、x y z の 3 次元グラフのほかに、よく使われるものとして、ヒストグラムがあります。頻度を計算することなく、データの列シリーズと区切りの数を指定するだけで簡単に描けて、特に、データの分布を見るときに有効です。

プログラム

```
new; cls;  
library pgraph;  
graphset;  
y=rndn(10000,1);  
hist(y,21);
```

グラフ表示



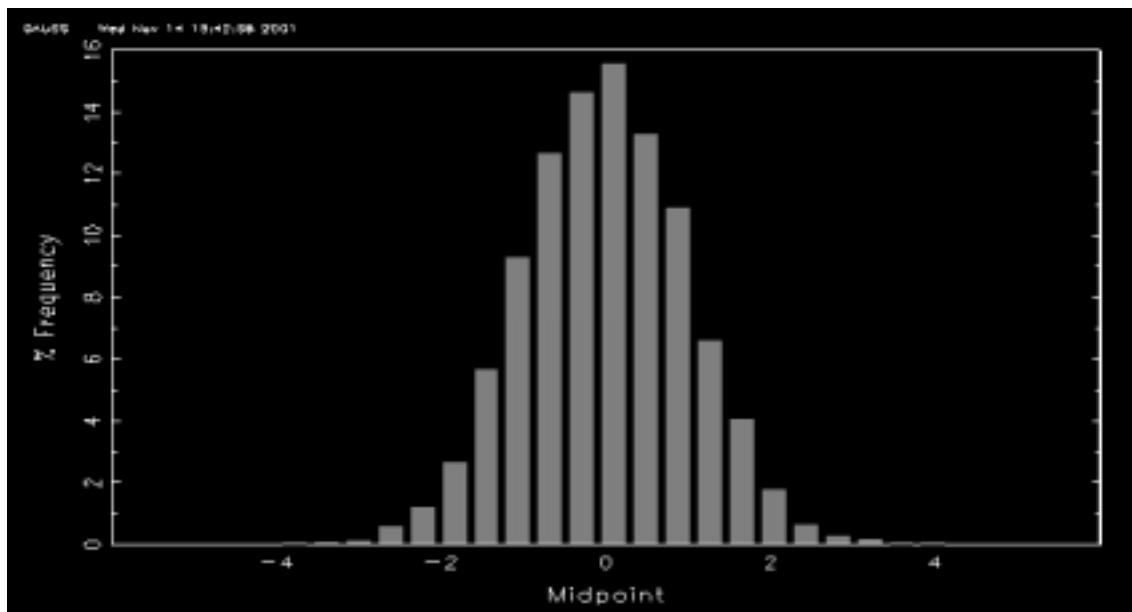
上のように、一連の手続きの後、`rndn(1000,1);`で standard normal にふるまう乱数 10000 個を一列に発生させます、それを、`hist(y,21);`で `y` という列ベクトルを 21 の区切りに分けて表示します。`hist` の第一要素は、表示させる列ベクトル。ここには、`xy` グラフと違って、行列を入れることはできません。第 2 要素は、ヒストグラムの区切りの数。経験的に、左右対称な分布をヒストグラムに描くときには、このきぎりの数は奇数個の方がより効果的でしょう。ほかの計量パッケージとは違って、この区切りの数を `GAUSS` は自動的に決定するわけではないので、この 21 の数のところをデータの数と分布の広がりに応じて、適宜変更する必要があります。何度か自分でグラフを試し描きするか、または、プログラムにより、区切りの数をいくつにするのかのルールをあらかじめ決めておくことも必要化もしれません。この `hist` で描かれるグラフは縦軸に頻度をとるものです。

同じヒストグラムでも、縦軸にパーセントをとる場合には、`histp` を使います。この `p` は `percentage` の `p` です。`hist` とまったく同様に扱うことができます。(ただし、これとは別の `histf` というグラフコマンドは、頻度を表すグラフではあるのですが、`hist` とはまったく違う変数の扱いをします。あまり使われることはないなので、ここでは省略します。)

プログラム

```
new; cls;
library pgraph;
graphset;
y=rndn(10000,1);
histp(y,21);
```

グラフ表示



上のように、縦軸が% Frequency にかわったと思います。hist も histp も両方とも、第 1 変数に行列はこれなくて、1 列の列ベクトルのみがきて、1 種類のヒストグラムを表示するだけです。重ね書きは、このコマンドだけではできません。なお、xy グラフや xyz グラフに使われた、ラベルやグラフ名を表示するコマンドなど、すべて同じように使えます。

ポイント 頻度ヒストグラムを描くには、hist(y , 区切りの数);

%ヒストグラムを描くには、histp(y , 区切りの数);

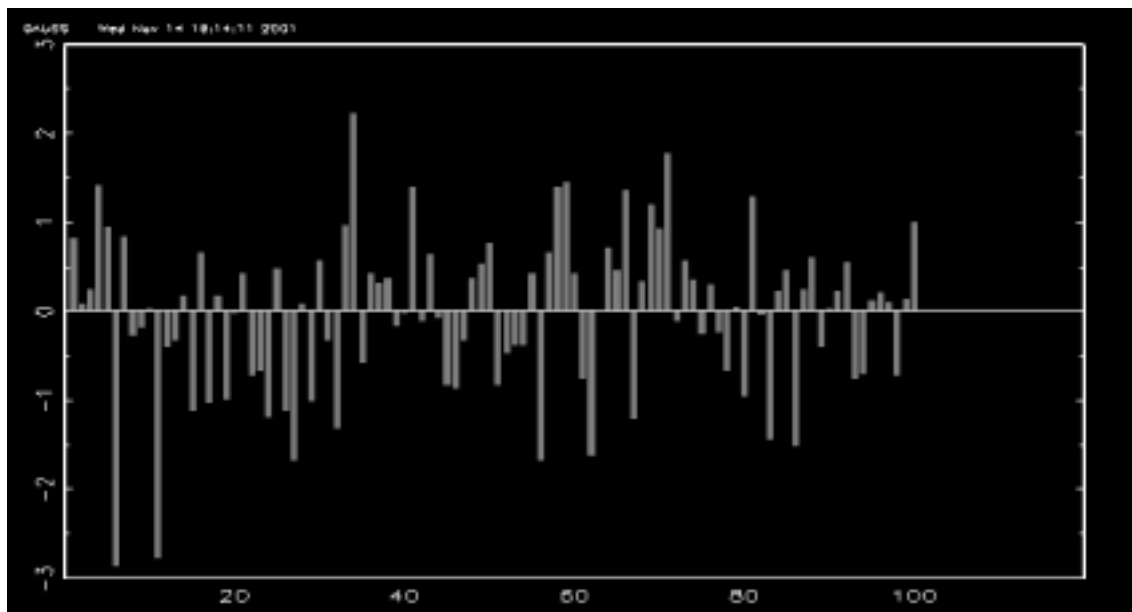
ただし、y は $M \times 1$ の列ベクトルのみ。区切り数は自分で設定する必要あり。

2 次元のグラフには、このほかに、xy グラフの変形バージョンとして、logxy(x,y)や logx(x,y)、それに loglog(x,y)、bar(x,y)があります。logy は y 軸を log スケールに変換したもの、同様に logx は x 軸スケールに変換したもの。loglog は x 軸 y 軸ともに log スケールに変換したものです。これらは x y グラフとまったく同じように使えます。bar(x,y)グラフは、すこし違いますので、ここで例を示しておきましょう。

プログラム

```
new; cls;
library pgraph;
graphset;
y=rndn(100,1);
x=seqa(1,1,100);
bar(x,y);
```

グラフ表示

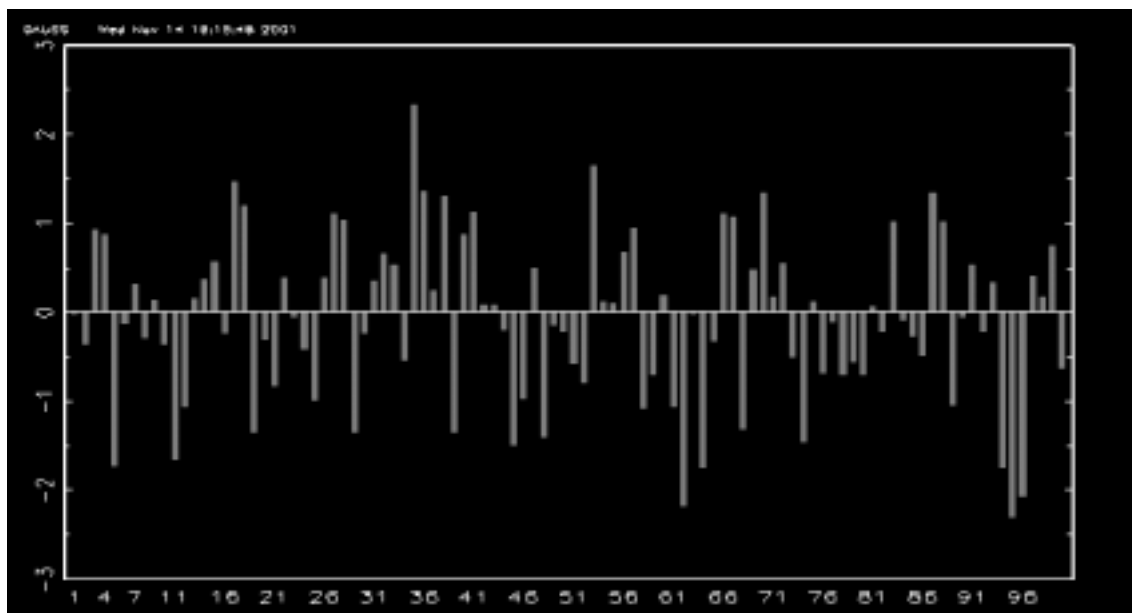


このグラフは、今回一番最初に表示したファイナンシャルライクな x y グラフと同じプログラムの $xy(x,y)$ を $bar(x,y)$ にかえただけで作成できます。そう意味で、 x 軸と y 軸のデータがそろっている場合には、 bar グラフは x y グラフとまったく同じように扱えます。ただ bar グラフは、 y 軸があたえられていなくても簡単に描けます。

プログラム変更（最後の2行のかわりに）

```
bar(0,y);
```

グラフ表示



基本的に上の2つのグラフは、若干横軸のふりかたが違う点と seed が異なる点を除き、まったく同じです `bar(0,y)`;と第1変数に0（ゼロ）を入れてやれば、x軸に相当するデータがあたえられていなくても、自動的に、1からyのデータの個数だけのシーケンスがx軸方向に生成されて、bar グラフが描かれます。

ポイント bar グラフを描くには、`bar(x,y)`;または `bar(0,y)`;

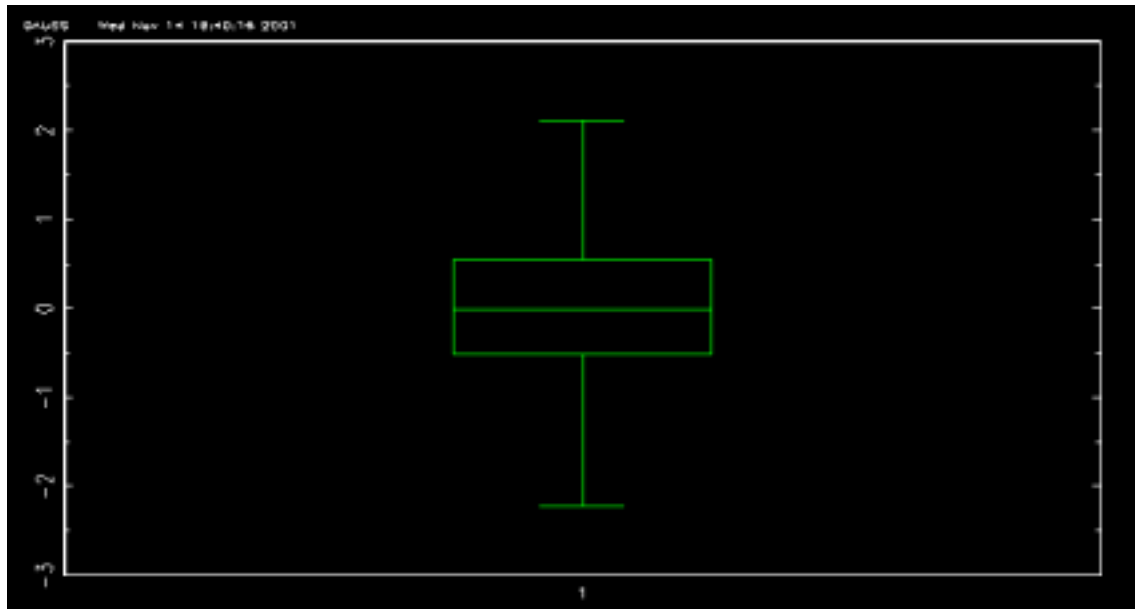
第1引数が0の場合には、x軸は1からデータ数だけの自動作成。

以上がよく使われるグラフでしたが、そのほかに、ファイナンシャル関係で使われるテクニカルなグラフとしては、box グラフがあります。

プログラム

```
new; cls;  
library pgraph;  
graphset;  
y=rndn(100,1);  
box(0,y);
```

グラフ表示



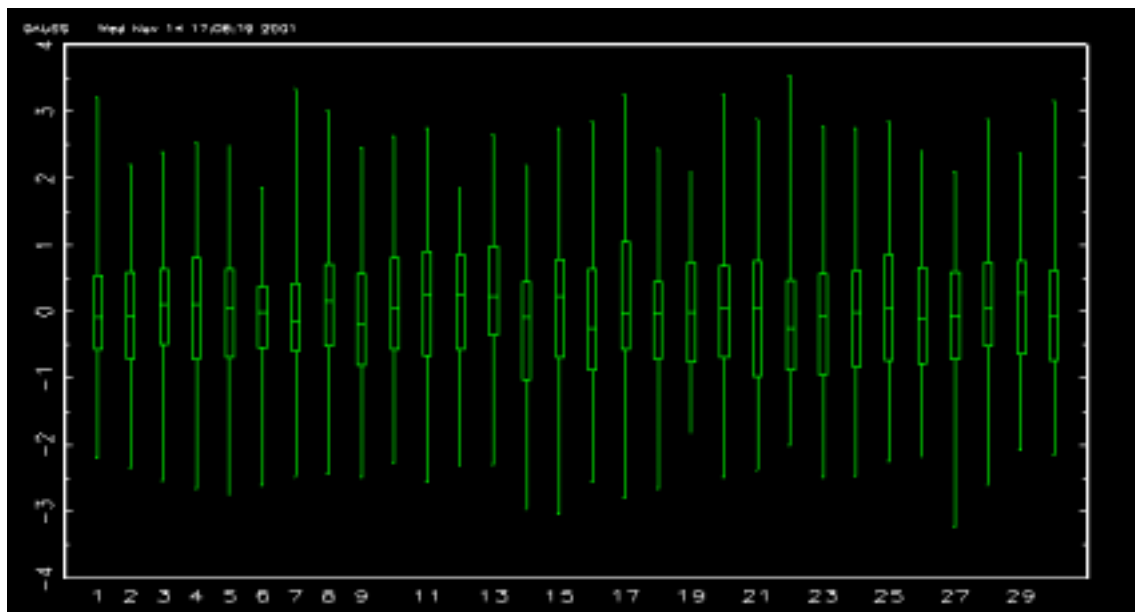
box グラフと同様なプログラムで、第1引数に0を入れます。すると、 $M \times 1$ の列ベクトルのシリーズの広がりを表す box グラフが1つだけ表示されます。データの最低値と最高値が上下に示され、ボックスは25th から75th の percentile で、中間の線は平均値です。

上の例は、データシリーズが1列分の特殊な例であって、実際には、yのところにN列分のデータが、例えば、為替や株価のデータでは、日次あるいは月次データとして入ってきます。その場合、yのところは $M \times 1$ の列ベクトルではなくて、 $M \times N$ のN日分あるいはN月分のデータ行列ということになります。rndnでstandard normalなデータを擬似的に描いてみましょう。下の例は、standard normal にふるまう乱数を30列分(30日分)100個ずつ発生させたものです。

プログラム

```
new; cls;  
library pgraph;  
graphset;  
y=rndn(100,30);  
box(0,y);
```

グラフ表示



実際には、例えば、それぞれの列に、1列目は第1日目、30列目は第30日目というふうに列データは30日分入ります。boxの第1引数の0は、seqaで1から30までシーケンスを生成してもかまいません。barグラフとどのように、boxグラフも、0が使える、x軸方向には自動座標設定が可能です。boxグラフは第2引数に、普通は、 $M \times N$ のNシリーズ分の行列がくることが、少し特殊です。ここでも、GAUSSではデータは、それぞれ列(縦)行列として扱われていますから注意が必要です。

ポイント box グラフを描くには、`box(x,y);`または`box(0,y);`

`y`には通常 $M \times N$ の N シリーズ分の行列データがくる。

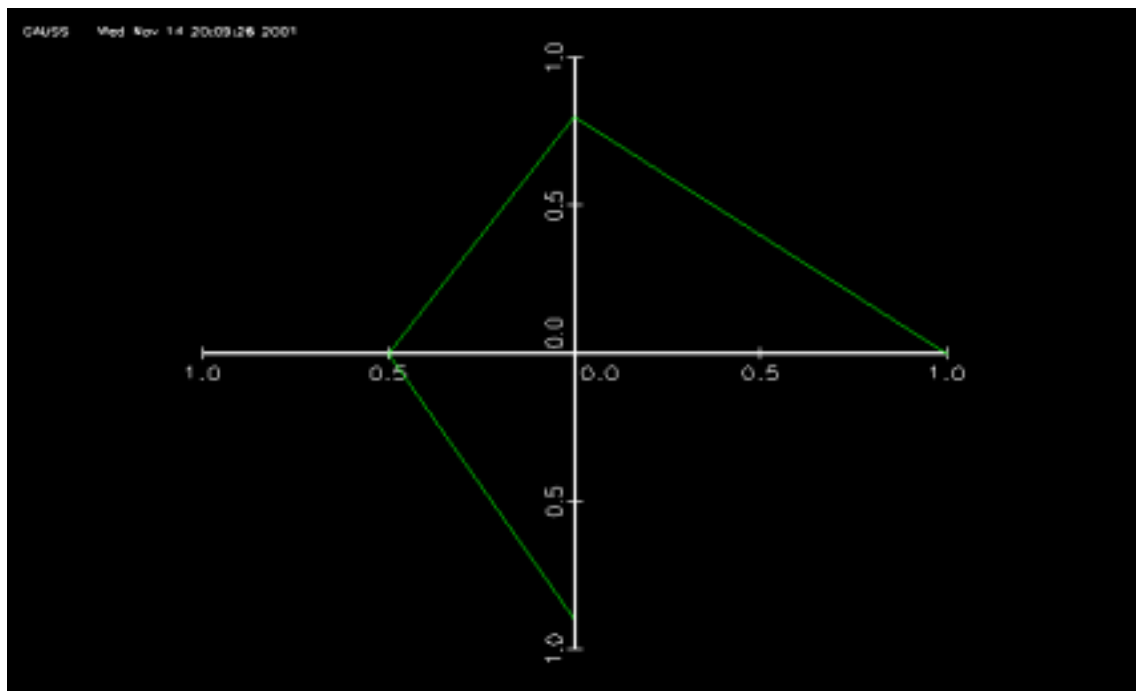
第1引数が0の場合には、 x 軸は1からデータ数だけの自動作成。

2次元のグラフには、あまり使われませんが、`polar` プロットという放射線状に奇跡をプロットするグラフもあります。詳しくはふれませんが、例だけを示しておきます。

プログラム

```
new; cls;  
library pgraph;  
graphset;  
y={10,8,5,9};  
r=seqa(0,pi/2,4);  
polar(y,r);
```

グラフ表示

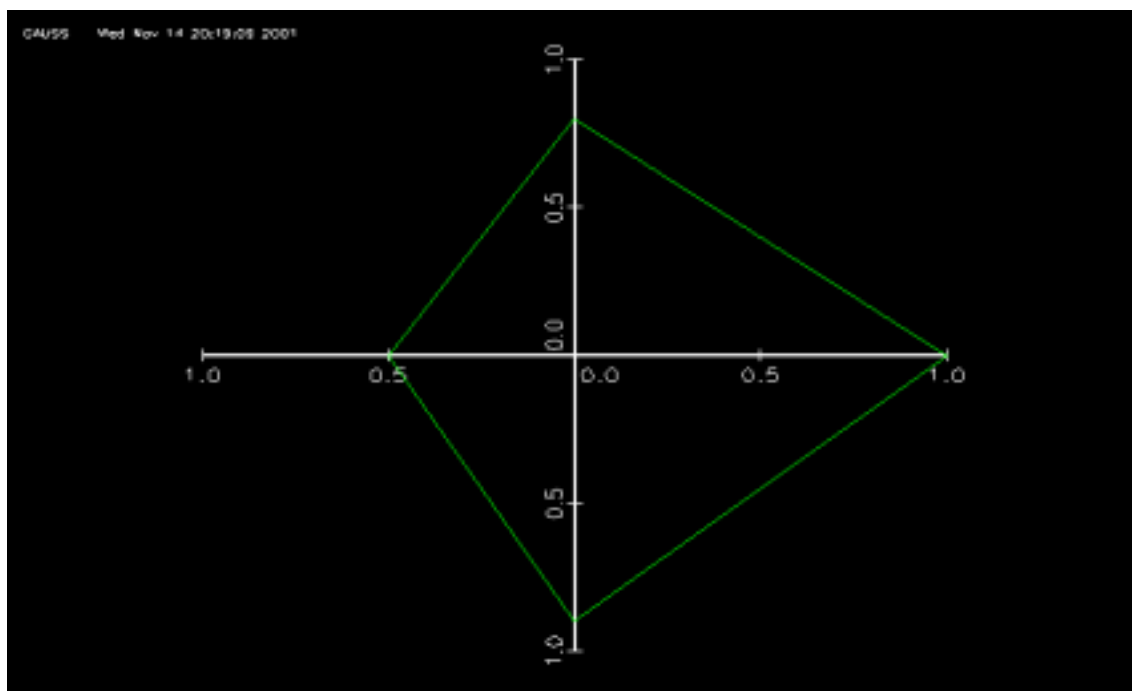


上のように放射状にプロットされました。 x y グラフと同様、デフォルトはラインのみになっています。`polar` の第1引数は、プロットされるデータの列ベクトル、または列ベクトルの集まりである行列データ。第2引数は、ラジアン表示の角度。`pi` は π を表す定数で、 $\pi/2$ ごとに0から左回りにプロットしています。なお、デフォルトの軸の目盛は、データ y の最大値を1としたときの各変数の割合。なお、0のところでグラフをつなげるには、

プログラム

```
new; cls;  
library pgraph;  
graphset;  
y={10,8,5,9,10};  
r=seqa(0,pi/2,5);  
polar(y,r);
```

グラフ表示



つまり、最後のデータを最初のデータと同じにして、2 だけ回転させてやると、当然のことながらグラフの始点と終点は一致します。y の最初の 10 を最後にもう 1 つもってきて、さらに $\pi/2$ だけ回しています。(列ベクトル y は横にカンマをつけて表示しています。)

最後に少し技術的な話をしますと、pgraph の p というのは、GAUSS のグラフィックインターフェイス Publication Quality Graphics の p からきています。実際には、この pgraph というライブラリーは普段は GAUSS のプログラム本体とは別に、GAUSS のフォルダーの下の src という各種ライブラリーがおさめられているフォルダーの中に入っています。詳しい設定や説明は、マニュアルの中にもあると同時に、この pgraph と名前のつくファイル群にも注釈として書かれています。(GAUSS の中から開くことが可能ですが、キーボードを押したりして変更してしまわないように、内容を見る際には細心の注意のこと。) 実際、pgraph のライブラリーファイルの中に、グローバル変数の初期設定値が代入の形で書かれていることが見てとれます。我々は、このグローバル変数を外から変更しているのです。