

# J言語を使って、実務を行う

2014/04/19

JAPLA 4月研究会資料

JAPLA 鳥邊錬太郎(Tribe Rentaro)

本来楽をしたい性分の私ですが、プログラムの作業はそれとは反対に、結構過酷な繰り返しや、指向が続き苦痛なものです。

ここでは、Jの式(Expression)やプログラム(Script)を使って作業するに当たり、私にとって、とても苦手な入力作業や、Jからの出力作業を今一度見直してみました。

ここでは、J初心者の方が、ある程度Jを習得して、実際に使用してみようと思ったときに簡単に利用できることを目標としました。

正直言いまして、私はまだJの初心者です。いろいろ学ばなければならないのに、何時も入口でモタモタしております。今回も毎度同じ苦悶から出発致します。

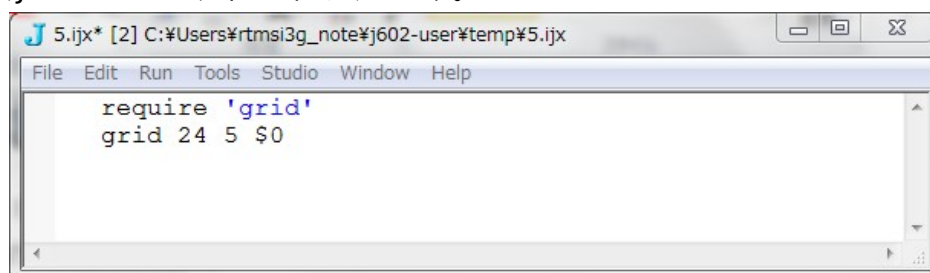
今回、ここで取り上げた例題は、JAPLAの副会長である鈴木義一郎先生の著書、「J言語による統計分析(森北出版株式会社)1996年」の一部から引用させて頂きました。

1. 以下のようなデータをEXCELに用意します。

	A	B	C	D	E	F
1	DataNo.	身長	体重	バスト	ウエスト	ヒップ
2	1	165	53	86	56	92
3	2	160	47	84	52	92
4	3	166	55	86	64	89
5	4	164	56	90	60	95
6	5	168	55	87	56	87
7	6	164	54	87	57	92
8	7	168	54	94	58	97
9	8	169	55	88	57	92
10	9	169	53	86	58	93
11	10	166	56	84	57	90
12	11	165	53	85	55	90
13	12	163	49	84	59	90
14	13	164	52	87	58	90
15	14	167	53	86	59	88
16	15	169	58	89	60	90
17	16	169	51	84	60	90
18	17	166	50	86	59	87
19	18	168	53	88	60	88
20	19	165	54	88	62	90
21	20	167	50	88	58	89
22	21	170	55	88	60	90
23	22	168	57	84	62	92
24	23	168	56	85	62	94
25	24	163	52	83	60	88
26						

表1 鈴木義一郎著 「J言語による統計解析」から、  
主成分分析より。EXCELデータ

1. まず、J Windowsに以下の式を入力します。



```
require 'grid'  
grid 24 5 $0
```

2. 24×5 のグリッドが現れます。

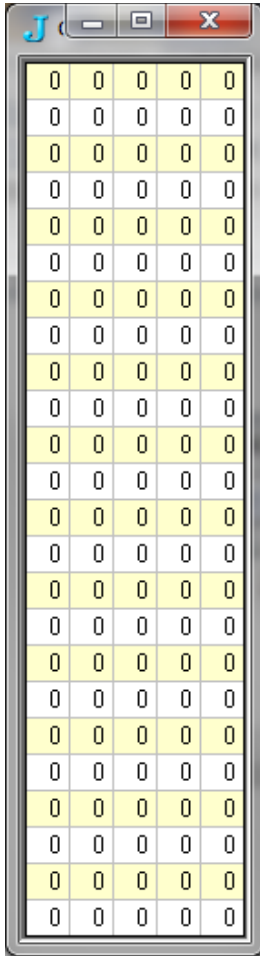


図1: 25×5

- 2.1. EXCEL の選択範囲をコピーします。
  - 2.2. J-Grid の左上 1 行 1 列のセル(赤枠)をクリック。
  - 2.3. 「**ctrl & V**」で、EXCEL のデータが、J-Grid に貼りつきます。
- ここで、本来なら、貼りついたデータを、変数 a に代入したいのですが、現在のところ、私の力量ではできません。

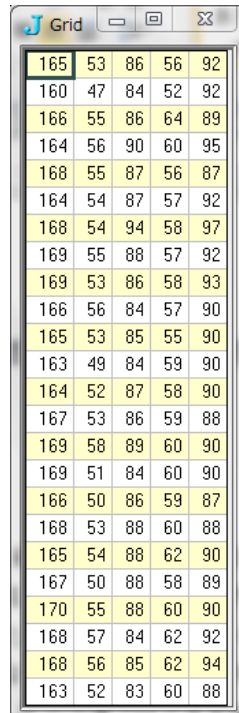


図2: データ貼付け

3. そこで、取り敢えず以下の方法で、変数に取り込みます。

EXCEL のデータをメモ帳に 1 列に並べ (データの区切りはスペース)、データの頭に「s=. 」をつけて、J-Windows に貼り付けると、

```
ja=. 24 5$ 165 53 86 56 92 160 47 84 52 92 166 55 86 64 89 164 56 90 60 95 168 55 87 56 87 164  
54 87 57 92 168 54 94 58 97 169 55 88 57 92 169 53 86 58 93 166 56 84 57 90 165 53 85 55 90 163  
49 84 59 90 164 52 87 58 90 167 53 86 59 88 169 58 89 60 90 169 51 84 60 90 166 50 86 59 87 168  
53 88 60 88 165 54 88 62 90 167 50 88 58 89 170 55 88 60 90 168 57 84 62 92 168 56 85 62 94 163  
52 83 60 88
```

表2: J-Windows に貼り付けるためのベタ打ちのデータ

J-Windows が以下のようになり、カーソルがデータの意版後ろになるので、このまま Enter を叩きます。

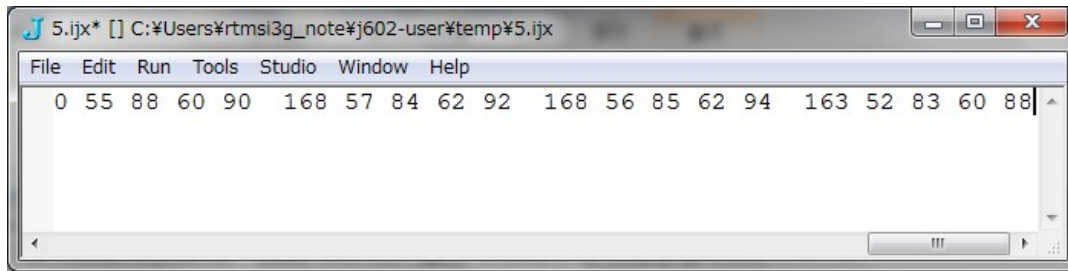


図3: データのベタ打ちを貼り付け、Enter を叩く

これで、データは変数「a」に取り込まれました。

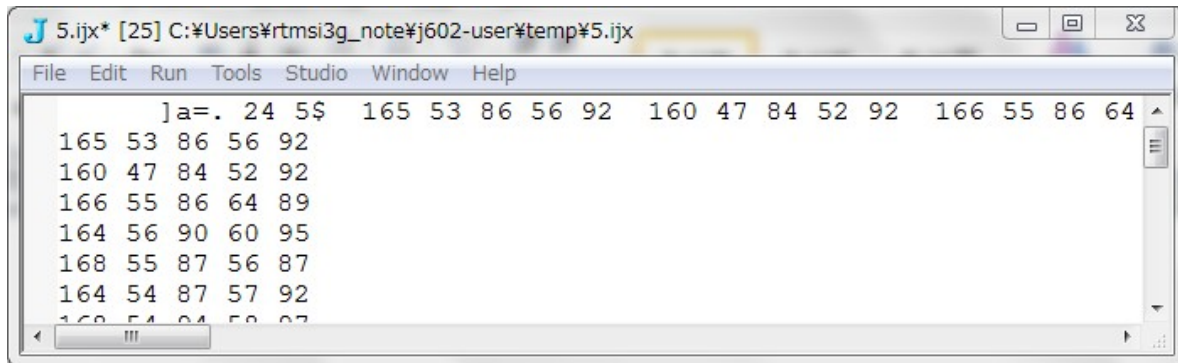


図4: 変数 a にベタ打ちデータが、24×5 のアレイで取り込まれる。

4. 主成分分析を計算しますが、ここでは計算の方法は論外ですので、経過のみを鈴木先生の本をなぞります。

A. 関数 itr

この関数は、固有値計算をするための前準備として、計算に必要な様々な行列などを作成する役割を持っているようです。この関数は、固有値計算の関数 mev から呼び出されています。

```
itr=.3 : 0
    m=.m+/. *m=.m%>./|,m=.m+/.
*m=.>{:y
    (v%>./|v=(.>{.y)+/ . *m);m
)
```

この関数の中で itr が使用されている。

```
mev=.3 : 0
    v=.v%+/*:v=.>{.itr^:20(1;y)
    (+/(y+/. *v)%v*#v),v
)
```

B. 関数 mev

この関数は、最大固有値 (第 1 番目) とその固有ベクトルを求めるものである。

C. 関数 red

```
NB.
red=.3 : 'y-({.lv)**/-}.lv=.mev y'
```

この関数は、2 番目以降の固有値とその固有ベクトルを順繰りに求めるものである。

さて、ここまでの関数は、実は鈴木先生はJの初心者に対して、固有値問題を解くその方法としてステップバイステップで、その仕組みと実例を使ってひも解いています。

D.関数 evs

この関数は、特に高次関数の場合、必ずしもすべての主成分を求める必要はなく、求めたい主成分の数を左変数に、当該相関係数行列を右変数で設定する、両側型を定義している。

```

evs=.3 : 0
  mev"2(red^(i.x))y
)
    
```

この関数を利用すると、Domain Error となる。どうやら左変数の x の使い方に問題があるらしい。

記、A、B、C の関数で、大まかな主成分が完成しました。

5. 上分析

10 ステップ程度のプログラムです。

VB や C などでのプログラムなら、数百ステップは必要でしょう。J 言語の威力です。

ここでは、その例を見てみよう。

鈴木先生は、小さなデータでそのロジックを検証しています。

本書の中で、まず計算の式を確認させるため、以下の関数を用いています。

```

mpbm=.3 : 0      NB.関数定義
  d=.%:(4*b=.*:1{y)+*:(a=.0{y)-c=.2{y
  e=-.:a+c+(1,_1)*d
  |:e,v,:%:1-*:v=.(1,_1)*%:b%b+*:a-e
)
mpbm 6 2 3      NB.実行結果
7 0.894427 0.447214
2 _0.447214 0.894427
    
```

これに対して、B.mev 関数を照らしてみると、

```

NB.mev 関数を使用して計算、第 1 固有値
JM2=.2 2$6 2 2 3      NB.実験行列の定義
6 2
2 3
  mev M2      NB.実行結果
7 0.894427 0.447214
    
```

さらに、mev と red を重ねて照らすと、

```

NB.第 2 固有値の計算
  mev red M2
2 _0.447214 0.894427
    
```

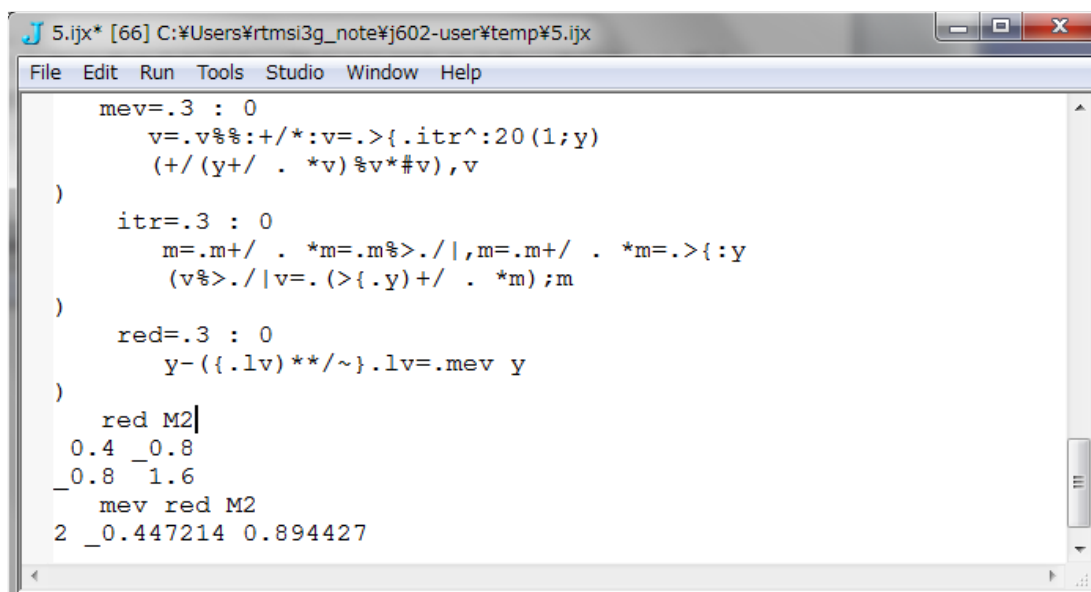
A,B,C の関数を小さなデータで検証出来ました。

初心者の方々は、J の式をいきなりこのような形で見せられても、何のことはわからない方が多いと思います。

J言語では、プログラムの式はアスキー文字の記号を使っているため、学習しないと判りにくいと思います。

ここでは、こんなものだと覚えて下さい。

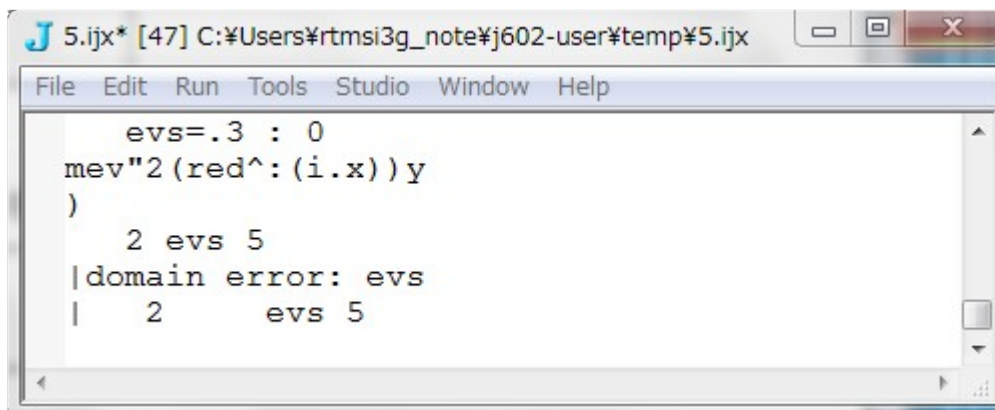
これらの式やプログラムは、Jのワークシートに入力します。Jの実行は式を書き、入力キーを叩くと直ちに実行する、インタープリター方式です。



```
J 5.ijx* [66] C:\Users\rtmsi3g_note\j602-user\temp\5.ijx
File Edit Run Tools Studio Window Help
mev=.3 : 0
v=.v%#:+/*:v=>{.itr^:20(1;y)
(+(y+/ . *v)%v*#v),v
)
itr=.3 : 0
m=.m+/ . *m=.m%>./|,m=.m+/ . *m=>{:y
(v%>./|v=.(>{.y)+/ . *m);m
)
red=.3 : 0
y-({.lv)**/~}.lv=.mev y
)
red M2|
0.4 _0.8
_0.8 1.6
mev red M2
2 _0.447214 0.894427
```

先ほどの、D 関数 evs ですが、そのままでは実行できないので、取り敢えず x の部分に求める固有値の数を直接設定して実行することにしました。

```
evs=.3 : 0
  mev"2(red^(i.x))y
)
```



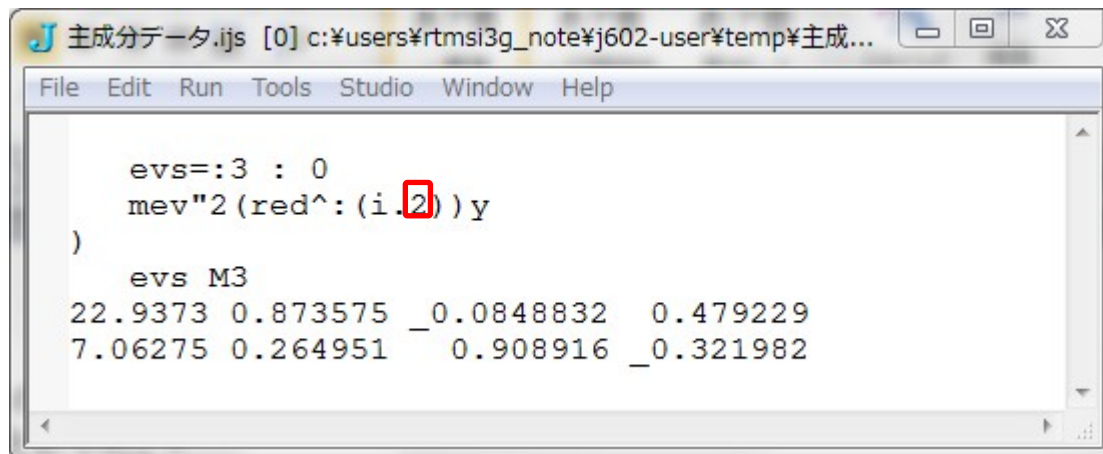
The screenshot shows a Jupyter Notebook window titled "5.ijx\* [47] C:\Users\rtmsi3g\_note\j602-user\temp\5.ijx". The code cell contains the following code:

```
evs=.3 : 0
mev"2(red^(i.x))y
)
```

The output shows a domain error:

```
2 evs 5
|domain error: evs
| 2      evs 5
```

その結果は、以下ようになります。



The screenshot shows a Jupyter Notebook window titled "主成分データ.ijs [0] c:\users\rtmsi3g\_note\j602-user\temp\主成...". The code cell contains the following code:

```
evs=:3 : 0
mev"2(red^(i.2))y
)
```

The output shows the eigenvalues and eigenvectors:

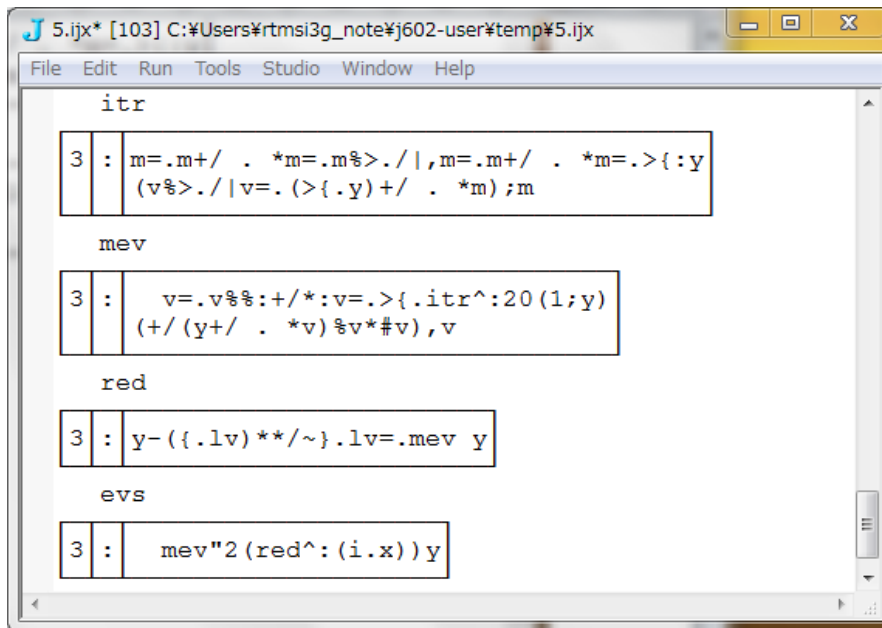
```
evs M3
22.9373 0.873575 _0.0848832 0.479229
7.06275 0.264951 0.908916 _0.321982
```

注意:この問題は、4月19日の研究会で西川先生、志村さんのご指摘で、  
「evs=:3 : 0」を「evs=4 : 0」で解決しました。両項式の宣言を忘れていました。  
昔の Version では、`evs=3 : 0` で正しかったらしい。2014年4月20日解決。  
:

6. では、ここでせっかく準備した、「美女のプロポーション」データを当てはめてみます。

図4で代入した、変数 a のデータを使用して、鈴木先生の関数に対応してみましょう。

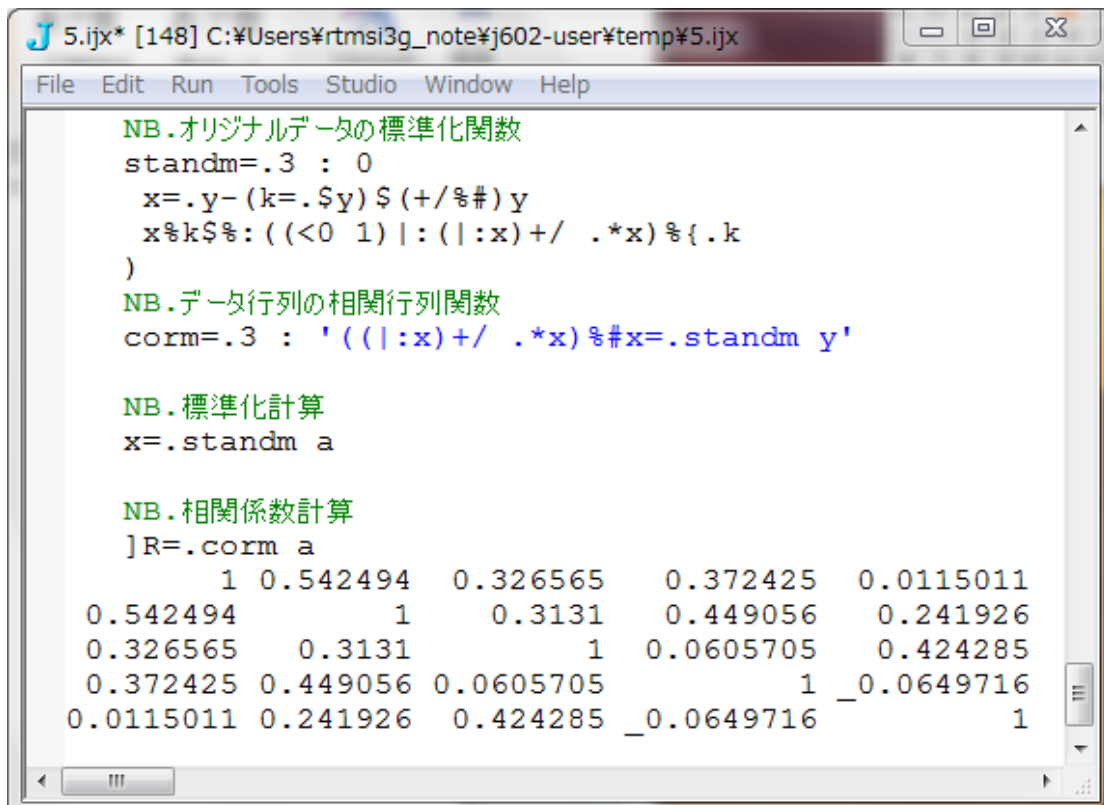
まず、4で表示した関数、を、J-Windows に記憶させる。



必要な関数はすべて、記憶されています。ただし、evs 関数は実行時に左変数を実数に変更する。

24×5 のデータは、変数 a にあります。

6.1.



## 6.2.主成分分析を実行

```
J 5.ijx* [168] C:\Users\rtmsi3g_note\j602-user\temp\5.ijx
File Edit Run Tools Studio Window Help

NB.evs関数を実行して、主成分を計算
evs R

evs=.3 : 0
    mev"2 (red^(i.5)) y
)
|
evs
3 : mev"2 (red^(i.5)) y

evs R
2.14898 0.521033 0.570928 0.419977 0.399578 0.25792
1.30735 _0.236935 _0.104455 0.483214 _0.488062 0.679155
0.675427 _0.534739 0.198549 _0.401878 0.513792 0.499144
0.509665 _0.27597 _0.481168 0.591852 0.551293 _0.195203
0.358586 0.557029 0.626247 0.279874 0.185005 0.43009
```

赤枠が固有値

青枠が固有ベクトル

統計学的解説は、次回の予定です。

ここでの問題は、外部からJ-Grid に貼り付けたデータをJの変数に取り込むことです。

何となく、require 'grid'の関数に、そういう操作ができる機能がありそうですが？

皆さんのお知恵を拝借したいです。

require 宣言を解除するには？

7.Jの計算結果をJ-Grid に出力する。

NB.主成分分析結果

result

```
2.14898 0.521033 0.570928 0.419977 0.399578 0.25792
1.30735 _0.236935 _0.104455 0.483214 _0.488062 0.679155
0.675427 _0.534739 0.198549 _0.401878 0.513792 0.499144
0.509665 _0.27597 _0.481168 0.591852 0.551293 _0.195203
0.358586 0.557029 _0.626247 _0.279874 0.185005 0.43009
```

NB.固有値

]eval=. {. |:result

```
2.14898 1.30735 0.675427 0.509665 0.358586
```

NB.固有ベクトル



```

]evec=.|:({).|:result)
0.521033 0.570928 0.419977 0.399578 0.25792
_0.236935 _0.104455 0.483214 _0.488062 0.679155
_0.534739 0.198549 _0.401878 0.513792 0.499144
_0.27597 _0.481168 0.591852 0.551293 _0.195203
0.557029 _0.626247 _0.279874 0.185005 0.43009

```

これらのJ-変数を Grid へ出力する

```

require 'grid'

grid result
grid eval
grid evec

```

その結果は、

result 主成分結果

2.148980	0.521033	0.570928	0.419977	0.399578	0.257920
1.307350	-0.236935	-0.104455	0.483214	-0.488062	0.679155
0.675427	-0.534739	0.198549	-0.401878	0.513792	0.499144
0.509665	-0.275970	-0.481168	0.591852	0.551293	-0.195203
0.358586	0.557029	-0.626247	-0.279874	0.185005	0.430090

eval 固有値

2.14898	1.30735	0.675427	0.509665	0.358586
---------	---------	----------	----------	----------

evec 固有ベクトル

0.521033	0.570928	0.419977	0.399578	0.257920
-0.236935	-0.104455	0.483214	-0.488062	0.679155
-0.534739	0.198549	-0.401878	0.513792	0.499144
-0.275970	-0.481168	0.591852	0.551293	-0.195203
0.557029	-0.626247	-0.279874	0.185005	0.430090

このJ-Gridと外部の接続は、J-Gridを選択して、「ctrl & C」でコピーできます。

これを、外部の EXCEL や Word に貼り付けてください。

ここで、問題は、J-Grid に変数の内容と説明の文字なども出力できないか?ということです。

恐らく、「配列の形、変形、連結」あたりのテクニックを使って Grid に出力すればよいかな?と思っています。

本テキストを **J-Windows** で再現するための参考です。これを **J-Windows** に貼り付けて、実際を確認できます。目視タイプは結構間違いのもとになります。

1. 関数

```
itr=.3 : 0
      m=.m+/. *m=.m%>./|,m=.m+/. *m=.>{:y
      (v%>./|v=(.>{:y)+/. *m);m
    )
```

2. 関数

```
mev=.3 : 0
      v=.v%: +/*:v=>{:itr^:20(1;y)
      (+/(y+/. *v)%v*#v),v
    )
```

3. 関数

```
red=.3 : 'y-({.lv)**/~}.lv=.mev y'
```

4. 関数

```
evs=.3 : 0
      mev"2(red^(i.x))y
    )
```

5. 関数テスト用相関係数(3×3)

```
JM3=.3 3$18 0 9 0 6 _3 9 _3 6
      18 0 9
      0 6 _3
      9 _3 6
```

6. 標準化関数

NB.オリジナルデータの標準化関数

```
standm=.3 : 0
      x=.y-(k=.$y)$(+/%#)y
      x%k$%:((<0 1)|:(|:x)+/. *x)%{.k
    )
```

7. 相関係数計算関数

NB.データ行列の相関行列関数

```
corm=.3 : '((|:x)+/. *x)%#x=.standm y'
```

8. 関数使用の計算例

NB.標準化計算

```
x=.standm a
```

NB.相関係数計算

```
JR=.corm a
      1      0.542494 0.326565 0.372425 0.0115011
      0.542494      1      0.31131 0.449056 0.241926
```

```
0.326565 0.3131 1 0.0605705 0.424285
0.372425 0.449056 0.0605705 1_ 0.0649716
0.0115011 0.241926 0.424285_0.0649716 1
```

9.両側型関数を片側型に変更(理由は DomainError のため)

NB.evs 関数の変更

```
evs=.3 : 0
    mev"2(red^(i.5))y
)
```

10.連続して、主成分を算出

NB.evs 関数を実行して、主成分を計算

```
evs R
2.14898 0.521033 0.570928 0.419977 0.399578 0.25792
1.30735 _0.236935 _0.104455 0.483214 _0.488062 0.679155
0.675427 _0.534739 0.198549 _0.401878 0.513792 0.499144
0.509665 _0.27597 _0.481168 0.591852 0.551293 _0.195203
0.358586 0.557029 _0.626247 _0.279874 0.185005 0.43009
```

11.オリジナルデータをJに取り込む

```
]a=. 24 5$ 165 53 86 56 92 160 47 84 52 92 166 55 86 64 89 164 56 90 60 95 168
55 87 56 87 164 54 87 57 92 168 54 94 58 97 169 55 88 57 92 169 53 86 58 93 166 56
84 57 90 165 53 85 55 90 163 49 84 59 90 164 52 87 58 90 167 53 86 59 88 169 58 89
60 90 169 51 84 60 90 166 50 86 59 87 168 53 88 60 88 165 54 88 62 90 167 50 88 58
89 170 55 88 60 90 168 57 84 62 92 168 56 85 62 94 163 52 83 60 88
165 53 86 56 92
160 47 84 52 92
166 55 86 64 89
164 56 90 60 95
168 55 87 56 87
164 54 87 57 92
168 54 94 58 97
169 55 88 57 92
169 53 86 58 93
166 56 84 57 90
165 53 85 55 90
163 49 84 59 90
164 52 87 58 90
167 53 86 59 88
169 58 89 60 90
169 51 84 60 90
166 50 86 59 87
168 53 88 60 88
165 54 88 62 90
```

167 50 88 58 89  
170 55 88 60 90  
168 57 84 62 92  
168 56 85 62 94  
163 52 83 60 88