

川柳で やさしくいざなう J 言語

統計数理研究所 鈴木 義一郎

- § 1 総和
- § 2 アレイの形とアイテム数
- § 3 (算術)平均
- § 4 調和平均
- § 5 最小値・最大値・範囲
- § 6 どこで最小・最大となるか?
- § 7 切捨て・切上げ・四捨五入
- § 8 分散と標準偏差
- § 9 変動係数
- § 10 偏差値
- § 11 分類と集計
- § 12 ボックス(<)とオープン(>)
- § 13 複合データの平均
- § 14 複合データの分散
- § 15 相関比
- § 16 メディアン
- § 17 四分位偏差
- § 18 回帰直線
- § 19 分散共分散行列
- § 20 相関行列

§ 1 総和

データに プラス・スラッシュ(+ /) 合計値

局所定義は イコールピリ(= .) イコールコロンの(= :)で 大局定義

```
sum=:+/
```

【総和を求める(片側形の)関数を定義している(『=:』)は大局定義】

```
MIN2=.2.1 1.5 2.9 3 3.3 3.7 4.7 4.2 4.8 4.9 3 2.9 1.1 1.8 2 4 5 3.3 0.5 0.9 0.5_1.4
_0.4 2.3 3.9 8.4 7.2 6.7
```

【東京地区の2月の最低気温のデータを『MIN2』に挿入(局所定義)している。MIN2は数学では、いわゆる「ベクトル」であるが、J言語では「リスト」と呼称する】

```
sum MIN2
```

86.8

【MIN2というリストの総和を与えている】

```
]M=.4 7 $ MIN2
```

```
2.1 1.5 2.9 3 3.3 3.7 4.7
4.2 4.8 4.9 3 2.9 1.1 1.8
2 4 5 3.3 0.5 0.9 0.5
_1.4 _0.4 2.3 3.9 8.4 7.2 6.7
```

【『\$』の両側形は「形(Shape)」と呼ばれる演算子で、MIN2というリストを、4×7の行列(J言語では「テーブル」と呼称)に変形している】 +/ M

```
6.9 9.9 15.1 13.2 12.9 13.7
```

【テーブルに『+/』を演算すると、縦方向の総和が得られる】

```
+/"1 M
```

```
21.2 22.7 16.2 26.7
```

【テーブルを横方向に足したければ、『+/"1』のように「ランク」を指定する】

..... 「J」言語メモ

J言語の最も特徴的な点は、動詞、名詞、副詞といったように、通常の言語と同様に「品詞」という概念が対応していることである。

```
1+2+3+4+5
```

+ / 1 2 3 4 5

15

【『 / 』は『 + 』という動詞の右に配置し、“右引数の間に + を挿入する”という副詞的な役割を果たしている。結局、『 + / 』は「全ての要素を足す」という動詞になる】

なお「ランク」を指定しない『 + / M 』の結果は、縦に足したかにみえるが、実は
(2.1 1.5 2.9 3 3.3 3.7 4.7)+(4.2 4.8 4.9 3 2.9 1.1 1.8)
+(2 4 5 3.3 0.5 0.9 0.5)+(-1.4 -0.4 2.3 3.9 8.4 7.2 6.7)
という4つのリストを加えた結果を表示しているのである。

§ 2 アレイの形とアイテム数

形なき たった1つは「アトム」なり アトムが並んで「リスト」を作る
上から下 リスト集めりゃ「テーブル」さ それを集めて 一般「アレイ」
引数の 低次のランクの 全てのものを「1セル」「2セル」などと呼ぶ
演算は 1つ低次の セル相手 これを名づけて「アイテム」と呼ぶ
テーブルなどの「形」を示す ドル(\$)マーク アイテム数は シャープ(#)さん

\$ 2.1

(empty!)

【ただ1個の要素は、数学でいう「スカラー」に相当し、「J」言語では「アトム」と呼ぶ。
ベクトル(Jではリスト)ではなく、形がないので空(')が返される】

```
$ MIN2          $ M=.4 7$ MIN2
28              4 7
```

【『\$』は、各ランクの要素の数を返す演算子で、結局、引数の「形」を示している】

2.1

1

【『#』は、アイテム(引数より1つランクの低いセル)の数を出力する演算子で、特にアトムのアイテムはそれ自身と定義されているから、アイテム数は1個という結果を返す】

```
# MIN2          # M
28              4
```

【リストに対するアイテムはアトムであるから、要素の個数を出力している。また引数が
テーブルの場合のアイテムはリストであるから、リストの個数を出力している】

..... 「J」言語メモ

「名詞」には、「ランク(rank)」という概念が対応、単独の数値「2.1」や文字「a」などは「アトム」で数学のスカラーに対応、ランクは0である。ランク1のものは「リスト」と呼ばれ、数学の(横)ベクトルに対応し、行列に対応するのが「テーブル」で、ランクは2である。さらにランク3以上のものもあり、一般に「アレイ(array)」と呼ばれる。

(rank=:\$@)\$2.1	rank M
0	2
rank MIN2	rank i .2 3 4
1	3

アレイの形は

\$ M

4 7

といったように、『\$』という演算子の片側形で与えられる。この形を示す数値の位置を左から順に、0軸、1軸、…… というように呼称する。

§ 3 (算術)平均

動詞が 3 つ 並んだときは 左右が先で 中の動詞は 3 番手(フォーク)

```
mean=:+/%
```

【算術平均を求める(片側形の)関数を定義している】

```
]LEFT=.+/ MIN2
```

86.8

```
]RIGHT=.# MIN2
```

28

【MIN2 というリストのアイテムの数(アトムの数)を与えている】

```
LEFT % RIGHT
```

3.1

【『%』は割算を行う両側動詞で、「86.8÷28」の値を求めている】

```
mean MIN2
```

3.1

【上の結果と同じで、MIN2 というリストの(算術)平均の値を求めている】

```
mean M
```

1.725 2.475 3.775 3.3 3.775 3.225 3.425

【テーブルに『mean』を演算すると、縦方向の(実はベクトルの)平均が得られる】

```
mean"1 M
```

3.02857 3.24286 2.31429 3.81429

【テーブルの 1 軸(横)方向の平均は、『mean"1』のように「ランク 1」を指定する。つまり、M というテーブルの 1 - セルであるリストを引数にするから、リストのアイテムであるアトムに対して「mean」という動詞が作動し、1 軸(横)方向の平均を求めている】

..... 「J」言語メモ

あるアレイの軸の右のほうから k 番目まで取り出した要素を「k - セル」とか「ランク k のセル」という。例えば、M というアレイの「2.1 1.5 2.9 3 3.3 3.7 4.7」などは、1 つの「1 - セル」である。また、M の「2 - セル」は M 自身で、1 つしかない。「0 - セル」は必ずアトムで、「1 - セル」はリストである。特に、アレイのランクより 1 つだけ小さいランクのセルを、特に「アイテム」という。一般に、動詞の作用する対象はアイテム

であるが、「 ” 1 」という副詞をつけると、「 1 - セル」を引数とみなして演算を行うことになる。

また J 言語では、3 種類の動詞が連なったものを「フォーク(Fork)」と呼ぶ。一般に

$$(f g h) x$$

のように右側だけに「引数」のある場合を「片側形」といい

$$x (f g h) y$$

のように両側に「引数」のある場合を「両側形」と呼んでいる。そして演算の順序は

$x (と y)$ に h という動詞をオペレートした結果を右引数に

$x (と y)$ に f という動詞をオペレートした結果を左引数に

最後に真中にある g という “ 両側動詞 ” をオペレートする

といった手順で実行される(特に中央の動詞「 g 」は必ず両側形でなければならない!)。

§ 4 調和平均

片側の動詞を順に結ぶのが アンド(&)やアット(@)の接続詞
アンダー(&.)で 2つの動詞を連結すれば 逆演算が 付加される

例えば、行きは時速 60、帰りは 40 で往復したとする。平均時速を

$$(60+40)/2 = 50$$

のように算術平均を算出することで正しいだろうか。いま移動距離を D としてみると、行きの所要時間と帰りの所要時間は、それぞれ

$$D/60, D/40$$

であるから、これらの合計が往復の所要時間になる。したがって平均時速は

$$2D / \{D/60 + D/40\} = 2 / \{1/60 + 1/40\} = 1 / \{[(1/60) + (1/40)]/2\}$$

のように算出しなければならない。つまり、逆数の算術平均を求めてから、その答えの逆数を求めてやればよいことが分かる。このような値は「調和平均」と呼ばれている。

$$[\{(1/60) + (1/40)\}/2] = 0.0208333$$

であるから、この値の逆数は

$$\% 0.0208333$$

48.0001

となる(正確には 48)。そこで J 言語では、逆数を求める『%(片側形)』を用いて

$$\% \text{mean} \% 60 40$$

48

のように算出される。また、2つの動詞を右から順に演算するように連結した合成関数を与える『&』や『@』という「接続詞」を用いて

$$\% \& \text{mean} \% 60 40$$

48

$$\% @ \text{mean} \% 60 40$$

48

のようにしても同じ結果が得られる。さらに、『&.』という接続詞を用いれば

$$\text{mean} \& . \% 60 40$$

48

【『mean&.%』は『%@mean&%』と同じ演算を行う(%の逆演算は%である!)】

..... 「J」言語メモ

「 月 日、晴れ。山へ登った。頂上でお弁当を食べて遊んだ。山から降りた。」

この文章では

f = 「山へ登る」

g = 「頂上で弁当を食べて遊ぶ」

h = 「山から降りる」

という3つの動詞が、上から順に実行されている。つまり『 $h@g&f$ 』という合成関数がオペレートされたことになる。

ところで、山へ登りっぱなしという人はいなくて、必ず山を降りるはずである。「山から降りる」という動詞は、「山へ登る」前の状態に戻ることになるので、hはfという動詞の「逆演算」になる。J言語の場合、fの逆演算は『 $f\wedge_1$ 』で与えられる。そこで、関数『 $g\&.f$ 』が『 $(f\wedge_1)@g&f$ 』と同じ働きをするように、「アンダー(under)」という接続詞『 $\&.$ 』(Jのヒット商品!)が用意されているのである。

§ 5 最小値・最大値・範囲

小にチョン(<.) 両側形は 最小値 大にチョン(>.)なら 最大値

```
2 <. 3          2 >. 3
2                3
```

【『<.’』という演算子の両側形は、左右の引数の小さい方の値を取り出す働きをする。また『>.’』は、大きい方の要素を取り出す両側演算子である】

さらに、平成9年の2月の最低気温に対して

```
<./ MIN2          >./ MIN2
_1.4              8.4
```

【最も低かったときの気温と、最も高いときの気温を出力している】

さらに

```
range=.>./-<./
という関数を定義してやれば
range MIN2
```

9.8

といったように、最大値と最小値の差である「範囲」の値が出力される。

さらに、4 × 7のテーブルMにも適用してみると

```
range"1 M
3.2 3.8 4.5 9.8
```

といった結果が得られる。最後の7日間で気温の変動の大きかったことなどが分かる。

..... 「J」言語メモ

J言語に用意されている『>.’』などは、原始動詞(atomic verb)と呼ばれている。これに対して、ユーザーが勝手に定義した動詞を代動詞(proverb)と名づけて区別することもある。各原始動詞の働きは固有のものであるのに対して、代動詞の働きの方は定義次第で如何様にも変化し得るということになる。

『/』は「インサート(insert)」と呼ばれる副詞で、「動詞を引数のアイテム間に挿入する」働きをする。先に定義した『sum=:+/'』のような関数にも、この『/』という副詞が使われていた。一般の副詞も、その左にある動詞を修飾して、結果も動詞となる。

また上で定義した関数に、引数を入れないで「リターン・キー」を押してみると
range

```
      -  
>./          <./
```

といったように、関数のカラクリがボックスで囲まれた形で表示される。この表示から、『range』という関数は3つの動詞の連なった「フォーク」であることが分かる。

日が26日であることを示している】

.....「J」言語メモ

『』』は「右引数を取り出す」という演算子で、「(]=<./)y」=「y=<./ y」のように演算を行い、これも「フック」である。

両側関数 g と片側関数 f の合成関数 『gf』が、両側形として機能する場合には

$$x(gf)y = xg(fy)$$

また片側形の場合には

$$(gf)y = yg(fy)$$

のような演算を行う場合に、接続詞を用いない2連動詞を「フック(Hook)」という。

§ 7 切捨て・切上げ・四捨五入

小にピリ(<.) 片側形は 切捨てる 大にピリ(>.)なら 切上げる
小コロン(<:) 片側形で 1を引く 大にコロン(>:)は 1を加える
動詞と名詞を アンド(&)で結べば 新たな動詞を 作り出す(@は不可!)

最低気温のデータに、『<.’』や『>.’』の片側形を適用してみると

<.’ M	>.’ M
2 1 2 3 3 3 4	3 2 3 3 4 4 5
4 4 4 3 2 1 1	5 5 5 3 3 2 2
2 4 5 3 0 0 0	2 4 5 4 1 1 1
_2 _1 2 3 8 7 6	_1 0 3 4 9 8 7

といった結果が出力される。つまり、切捨てや切上げによる整数値を与える。ここで注意しなければならない点は、負の場合には、正の場合と多少異なることである。

<.’ _1.4	>.’&.’ - _1.4
_2	_2
>.’ _1.4	<.’&.’ - _1.4
_1	_1
(<.’@+&0.5) 1.4	(<.’@+&0.5) 1.5
1	2

【「0.5」を加えてから切捨てているので、「四捨五入する関数」を与えている】

<: M	>: M
1.1 0.5 1.9 2 2.3 2.7 3.7	3.1 2.5 3.9 4 4.3 4.7 5.7
3.2 3.8 3.9 2 1.9 0.1 0.8	5.2 5.8 5.9 4 3.9 2.1 2.8
1 3 4 2.3 _0.5 _0.1 _0.5	3 5 6 4.3 1.5 1.9 1.5
_2.4 _1.4 1.3 2.9 7.4 6.2 5.7	_0.4 0.6 3.3 4.9 9.4 8.2 7.7

【『<:.’』や『>:.’』の片側形は「1」を引いたり加えたりするだけでさほど利用価値はない】

..... 「 J 」 言語メモ

『+』という動詞に「0.5」という定数を『&』という接続詞で連結すると

(+&0.5) 1.4	(0.5&+) 1.4
1.9	1.9

といったように、「“0.5”を加える」という動詞になる。また定数と動詞を逆順にしても同じ結果を与える。ところが『&』の代わりに『@』を使ってみると

(+@0.5) 1.4
domain error

のようにエラーになる。

さらに、次のようにしても「“0.5”を加える」という動詞(フォーク)が定義できる。

(]+0.5"_)1.4
1.9

つまり、『"_"』を適当な定数に付加すると、「その定数を取り出す」という動詞になる。

§ 8 分散と標準偏差

スターコロン(*:)は 平方値 逆演算の 平方根なら パーセントコロン(%:)
キャップ(:)はなんとも 不思議な動詞 何もしないで フォークを作る
並んだ動詞は 右からフォーク 残った動詞と またフォーク(5連以上の動詞)

2月の最低気温の平均は

```
mean MIN2
```

3.1

であった。そこで、「平均偏差」と呼ばれる値は次のように与えられる。

```
4 7$(]-mean) MIN2
```

```
_1 _1.6 _0.2 _0.1 0.2 0.6 1.6  
1.1 1.7 1.8 _0.1 _0.2 _2 _1.3  
_1.1 0.9 1.9 0.2 _2.6 _2.2 _2.6  
_4.5 _3.5 _0.8 0.8 5.3 4.1 3.6
```

【平均値からの「偏差」を出力している】

```
4 7$*(]-mean) MIN2
```

```
1 2.56 0.04 0.01 0.04 0.36 2.56  
1.21 2.89 3.24 0.01 0.04 4 1.69  
1.21 0.81 3.61 0.04 6.76 4.84 6.76  
20.25 12.25 0.64 0.64 28.09 16.81 12.96
```

【『*:』の片側形は「平方値」で、上記結果を平方している】

```
mean *: (]-mean) MIN2          %: mean *(]-mean) MIN2  
4.83286                          2.19838
```

【偏差平方値の平均を求めていて、これが「分散」に他ならない。また、『*:』の片側形は「平方根」で、「分散」の平方根である「標準偏差」を出力している】

一般に、分散や標準偏差を求める関数は次のように定義される。


```
var=: [:mean[:*:] -mean
```

```
sdev=:[:%:var
```

..... 「 J 」 言語メモ

『[:』は「キャップ(cap)」と呼ばれている動詞で、実際には“何も演算を行わない”という極めて特殊な演算子である。

```
var MIN2
```

```
4.83286
```

```
%: var MIN2
```

```
2.19838
```

```
([:%:var) MIN2
```

```
2.19838
```

【「[:%:var) y」は「%: var y」と同じ演算で、形式的にはフォークの1種に他ならない。
『[:』を除いてしまうと、「(%:var) y」はフックとなり、「(]%:var) y」と同じ演算になり、異なる演算結果を与えてしまう】

§ 9 変動係数

タシット(Tacit)で定義するのが 醍醐味さ J 特有の 面白さ
演算を 右から順に 作動さす ことも可能さ エクスプリシット(Explicit)

5 年間隔の自殺者数の年次データで、男子を S M、女子を S F という変数に入力する。

SM(Suicide of Male)

9820 13836 11506 8330 8761 11744 12769 15356 12316 13540

SF(Suicide of Female)

6491 8641 8637 6114 6967 8231 7773 8027 7772 6976

mean SM	mean SF
11797.8	7562.9
sdev SM	sdev SF
2151.97	837.462

【女子の自殺者のほうが少なく、標準偏差も男子の自殺者数の方の変動が激しい】

100*(sdev%mean)SM	100*(sdev%mean)SF
18.2404	11.0733

【標準偏差を平均で割って 100 倍した値で、変動係数(%表示)と呼ばれるものである】

```
coefv=:3 : '100*(%:mean*:y.-m)%m=.mean y.'
```

coefv SM	coefv SF
18.2404	11.0733

【明示的定義による関数を用いても同じ結果が得られる。女子の自殺者数の変動係数は、男子の場合より変動が少ない】

..... 「 J 」 言語メモ

これまでは、原始動詞や副詞等の “ プリミティブ ” を直接連結する形で、いろいろな関

数を定義してきた。このような「関数型定義」の方法は、「Tacit Definition」と呼ばれている。ところがJ言語には、「Explicit Definition」と呼ばれる、もう一つ別の方法で関数を定義することもできる。これを「明示的定義」と名づける。

まず、『coefv』といった動詞を定義するときには、

```
coefv=.3 : '***** .....
```

のように「3 :」の後に「シングルクォーテーション(')」で囲むことと、右引数の変数として「y.」を用いることである（左引数の変数は「x.」で、変数の文字の後に「ピリオド(.)」をつけないとエラーになる）。この関数のカラクリは

```
coefv
```

```
3 : 100*(%:mean*:y.-m)%m=.mean y.
```

のように表示される。

§ 10 偏差値

ダブルクォート(") ピリで数値化 コロンで文字化 書式も与える スグレモノ
左指定の個数だけ take({.})は取りで drop({.})は除く 便利な両側 動詞なり

```
T
55 53 51 49 47 45 43 41 39 37 35
【 11人の成績のデータ(特に降順になっていなくともよい)】
(mean,sdev) T
45 6.32456
【 Tの平均と標準偏差を求めている】
(]-mean) T
10 8 6 4 2 0 _2 _4 _6 _8 _10
【 Tの各要素から平均の45を引いた値で、「平均偏差」を与えている】
]T0=.6{.(stand=:(]-mean)%sdev) T
1.58114 1.26491 0.948683 0.6324556 0.316228 0
【平均偏差を標準偏差で割った(標準化した)値の先頭から6個までを表示している】
50+10*T0
65.8114 62.6491 59.48683 56.324556 53.16228 50
【直上の結果を10倍してから50を加えた値で、いわゆる「偏差値」である】
```

```
hensati=:3 : '50+10*(y.-mean y.)%sdev y.'
```

【偏差値を求める関数の明示的定義(Explicit Definition)】

```
0.1 ": hensati T
65.8 62.6 59.5 56.3 53.2 50.0 46.8 43.7 40.5 37.4 34.2
【『hensati』を用いて求めた偏差値を、書式関数『":』で1桁で四捨五入した値)】
```

..... 「J」言語メモ

```
0.2 ": 6{.hensati T
65.81 62.65 59.49 56.32 53.16 50.00
【 T1 を小数点以下2桁目で四捨五入した結果を表示(結果は文字列になっている!)】
```

```
1 + 0.2 ": 65.8114          1 + ". 0.2 ": 65.8114
domain error                66.81
```

【文字に数値は足せない！ 数値化する演算子『."]』をかぶせればエラーにならない】

また、次のような両側関数を明示的定義で行ってみると

```
round=:4 :'".(x.%10)":y.'
```

```
1 round 65.8154          2 round 65.8154
65.8                    65.82
```

といったように、左引数で指定した小数点の位置で「四捨五入」する関数が得られる。

§ 1 1 分類と集計

だぶってる データは消せと ニョロにピリ(~.) コロン(~:)つけば 不等記号
データを 分類するなら イコール(=)の 片側形を 使えばいい

```
]D=.2 ([:/:~[:>.%~) MIN2
0 0 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 4 4 5
```

【『~』は「左右の引数を交換して演算する」という副詞である。したがってDは、MIN2
というデータを2で割ってから切り上げて整数値にし、昇順に並べた値である】

```
]S=.~. D
0 1 2 3 4 5
```

【「引数の重複要素を排除する」という演算子『~.(Nub)』で、結果はSのようになる】

```
]T=. = D
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
```

【『=』の片側形は、“Self-classify”と呼ばれる演算子で、「=y」は「(~.=/)y」と同じ演算を行う。なお、「S=/D」は「S="0 1 D」という演算と同じである】

```
+/"1 T
2 7 11 5 2 1
```

【Dの系列に含まれる{0,1,2,3,4,5}の個数を出力している】

```
(~.,[:+/"1=)D
0 1 2 3 4 5
2 7 11 5 2 1
```

【Dをクラスの幅に2をとって分類した時の集計結果が出力される】
そこで、左引数で与えたクラス幅で分類・集計する関数を次のように定義する。

```
classfy=:[:(~.,[:+/"1=)]:/:~[:>.%~
```

2 classify MIN2
0 1 2 3 4 5
2 7 11 5 2 1

2 classify MIN3
2 3 4 5 6 7
3 9 11 4 3 1

..... 「 J 」 言語メモ

2 - ~ 3
1

3 - 2
1

【引数が両側の場合、『 ~ 』は左右の引数を交換して演算するという働きをする】

- / ~ i.2
0 _1
1 0

(i.2) - / i.2
0 _1
1 0

【引数が片側だけの場合、『 - / ~ y 』は「 y - / y 」のように左にも引数を挿入し演算する】

§ 1 2 ボックス(<)とオープン(>)

ボックス(<)で 囲めば全てが アトムに変身 オープン(>)使って 蘇生する
 ボックスで 囲み連結 セミコロン(;) 片側形なら リストに変身!

(AA=.<A=.1+i.5),BB=.<B=.6+i.4

```

- -
1 2 3 4 5      6 7 8 9
- -

```

【『<』の片側形は、引数の全体を“ボックス”で囲むという演算子である】

(rank AA),rank=.\$@\$ BB

0 0

【全体をボックス囲んだ要素はランクが0で、つまり「アトム」になる】

> AA

1 2 3 4 5

【『>』は『<』の逆演算で、「オープン(open)」と呼ばれる演算子である】

> AA, BB

1 2 3 4 5

6 7 8 9 0

【長さが不揃いの場合には、短い方の要素に対しては“0”が付加される】

<"0 C=.i.2 3

<"1 C

0 1 2

0 1 2 3 4 5

3 4 5

【ランク0でアトム、ランク1でリストに対し“ボックス”で囲む】

A;B

(<A),<B

1 2 3 4 5

6 7 8 9

1 2 3 4 5 6 7 8 9

-- |

【『;』の両側形は、「左右の引き数をボックスで囲んで接続する」という動詞である】

A;BB

(<A),BB

1 2 3 4 5 6 7 8 9

1 2 3 4 5 6 7 8 9

【右引数が『BB』のようにボックスの場合には、『A;BB』は『(<A),<BB』にはならない！】 ;/
A

1 2 3 4 5

【Aというリストの各アイテム(アトム)間に『;』を挿入している】

; <"0 C
0 1 2 3 4 5

> <"0 C
0 1 2
3 4 5

【『;』の片側形は、オープン(>)と類似しているが、出力結果は全てリストになる】

§ 1.3 複合データの平均

数値・文字 ボックス表示も みな「名詞」「動詞」が働き 結果出す
計算を まとめて演算 したければ レベル(L:)やイーチ(>)を 使えばいい

```
A;B                                mean L:0 (A;B)
-
1 2 3 4 5   6 7 8 9                3   7.5
-
```

【『L:0』という副詞で演算レベルを指定すると、ボックスの中ごとに平均を算出する】

```
mean&.> A;B                        <mean&> A;B
3   7.5                             3   7.5
```

【『>』でボックスを開いて演算し、『&.>』で接続しているので逆演算『<』している】

```
mean&> A;B
3 7.5
```

【『&』で接続すると逆演算『<』を行わないから、開いた状態で結果を表示する】

```
mean (A,B)                          (3+7.5)%2
5                                     5.25
```

【「Aの平均とBの平均の平均」は、複合データ{A,B}の平均と一致しない】

```
ABM=.5 3;4 7.5
```

【データAのデータ数と平均、そしてデータBのデータ数と平均をボックスで囲んだもの】

```
(* /, { . } ) &> ABM
15 5
30 4
```

【『{ . }』は先頭要素の取り出し。1列目にA,Bの総和、2列目にA,Bのデータ数を出力】

```
([:+ / (* /, { . } ) &>) ABM          % / 45 9
```

【複合データ{A, B}の総和Sとデータの総数Nを、右は「 $S \div N$ 」与えている】

そこで、加重平均を求める関数を次のように定義する。

```
wmean=:[:%/[:+/(*/,{.)&>
```

```
wmean ABM
```

5

【複合データ(A, B)の平均を与えている】

§ 1 4 複合データの分散

ヘッド({.})で先頭 テイル({:})で末尾 要素取り出す 片側動詞

オープン(>)は 動詞でアンド(&)は 接続詞 イーチ(&>)にすれば 副詞に変身

```
var&> A;B;A,B
```

```
2 1.25 6.66667
```

【データ A, B の分散、そして複合データ{A, B}の分散の値を与えている】

```
(2+1.25)%2          wmean 5 2;4 1.25
1.625                1.66667
```

【「Aの分散とBの分散の平均」は、複合データ{A, B}の分散とは一致しない。また、“加重平均”しても一致しない】

一般に、{データ数 m, 平均 MA, 分散 VA}のデータ A と、{データ数 n, 平均 MB, 分散 VB}のデータ B に対して、複合データ{A, B}の平均は

$$M = M(A, B) = \{mMA + nMB\} / (m+n)$$

のように各平均の加重平均である。さらに

$$WV = WV(A, B) = \{mVA + nVB\} / (m+n)$$

のように各分散の加重平均で定義されるものを、「級内分散」という。また

$$BV = BV(A, B) = \{m(MA-M)^2 + n(MB-M)^2\} / (m+n)$$

と定義されるものを「級間分散」という。これらの分散の合計が複合データの分散になる。

$$]MAB=.5 3;4 7.5$$

```
5 3    4 7.5
```

【データ A のデータ数と平均、そしてデータ B のデータ数と平均をボックスで囲んだもの】

```
({:&<-wmean) MAB          ]Q=.*:({:&<-wmean) MAB
_2 2.5                    4 6.25
]M=.{:&> MAB              (+/M*Q)%+/M
5 4                        5
```

そこで、データ数と平均から級間分散を求める関数を

```
bvar=:3 :'+/m**:({:&<-wmean)y.)/m=.{:&>y.'
```

のように定義する。{A, B}に対する級間分散は

```
bvar 5 3;4 7.5
```

5

【この値を分散の加重平均(WV)に加えると、複合データ(A, B)の分散に一致する】

さらに、データ数、平均、分散を与えて級内分散と級間分散を求める関数は

```
wbvar=:3 :'(wmean 0 2{L:0 y.},bvar):L:0 y.'
```

のように定義される。

```
wbvar 5 3 2;4 7.5 1.25
```

1.66667 5

【{A, B}に対する級内分散と級間分散を出力し、この合計が複合データの分散になる】

§ 1 5 相 関 比

「動詞」との 出会いひたすら 待つ「副詞」 右にこだわる 「接続詞」

MIN3

6.7 4.6 5.2 4 5 6.7 13.4 9.1 6.6 4.6 7.7 6.1 4.6 9.2 6.2 4.3
3.5 5.6 6.9 8 6.3 8.4 5.4 4.2 3 10.4 8.4 6.3 12 10.4 7.9

MIN4

8 11.1 10.5 10.7 13.3 12.6 10.9 10.3 12.9 12.6 10.5 8.8 9.5 12 10.6
8.9 9.9 11.2 11 12.1 13.5 14.5 9.9 9.6 11.1 11.8 11.3 14.1 13.2 16.1

【東京地区の平成9年3月と4月の最低気温のデータ、2月の気温 MIN2 は § 1】

]MV23=.(#,mean,var)L:0 MIN2;MIN3

28 3.1 4.83286 31 6.79677 6.04741

【2月、3月のデータ数、平均、分散である】

(;+/)wbvar MV23

5.47101 3.4077 8.87821

【2月、3月の気温のデータに対する級内分散、級間分散、そして全分散の値】

一般に、第 i 組のデータのデータ数、平均、分散をそれぞれ n_i, M_i, V_i とするとき ($i = 1, 2, \dots, k$)、複合データの平均、級内分散、級間分散は

$$M = \{n_1M_1+n_2M_2+ \dots +n_kM_k\}/N, \quad N = n_1+n_2+ \dots +n_k$$

$$WV = \{n_1V_1+n_2V_2+ \dots +n_kV_k\}/N$$

$$BV = \{n_1(M_1-M)^2+n_2(M_2-M)^2+ \dots +n_k(M_k-M)^2\}/N$$

のように与えられる。さらに

$$H = BV/\{WV+Bv\}$$

と定義される値を、「相関比」と呼び、データ間の“分離度”を示す尺度である。

に対しては

({:%/})wbvar MV23

0.383806

【2月、3月の気温のデータの級間分散を全分散で割った値で、相関比を与えている】

データから直接相関比を求める関数は次のように定義できる。

```
corratio=:3 :'+/(#&>y.)**:m-mean&>y.)%+/*:y-m=.mean y=.;y.'
```

```
corratio MIN2;MIN3          corratio MIN3;MIN4
0.383806                    0.533642
corratio MIN2;MIN4          corratio MIN2;MIN3;MIN4
0.811806                    0.706266
```

【2月と3月ではかなりの部分が重なっているため0.4以下だが、3月と4月では0.5以上、さらに2月と4月では0.8以上と完全に分離していることが分かる】

§ 16 メディアン

データを昇順にする グレードアップ(/:~) グレードダウン(\:~)は降順に
不景気で 売上げ半減 マイナスコロン(-:) プラスコロン(+:)で 所得倍増
左指定 インデックスの 要素選択 動詞はフロム({)の 両側形

```
2 14$ MD=.:~ MIN2
-1.4 -0.4 0.5 0.5 0.9 1.1 1.5 1.8 2 2.1 2.3 2.9 2.9 3
3 3.3 3.3 3.7 3.9 4 4.2 4.7 4.8 4.9 5 6.7 7.2 8.4
```

【 2月の最低気温のデータを 『/:~』 という演算子により昇順並べている 】

```
]A=.([:>.:[:-:#) MIN2          ]B=.(<:, -) A
14                                13 _14
```

【 MIN2 のアイテム数の半分を切上げて整数値にしたものがAで、またBにはAから1を引いた値と符号を変えた値とを連結している 】

```
B { MD=.:~ MIN2          -:@+/ B{ MD
3 3                        3
```

【 MD から 14 番目と末尾から 14 番目の要素を取り出し(先頭からの軸の指定は、“ 1 3 ” が通常の 14 番目になる！)、これらを足して 『-:』 で半分になっている 】

```
([:(<:, -)[:>.:[:-:#) MIN3      -:@+/ 15 _16{/ :~ MIN3
15 _16                            6.3
```

【 3月の気温は奇数個のデータで、インデクスは中央(この場合は 1 6 番目)になる 】

そこで、メディアンを求める関数を次のように定義する。

```
median=:[:-:@+/([:(<:, -)[:>.:[:-:#){/ :~
median&< MIN2;MIN3;MIN4
3 6.3 11.4
```


【2月、3月、4月の気温のデータのメディアン値を与えている】

..... 「J」言語メモ

『{』の両側形は、「選択(from)」と呼ばれている演算子で、

2{1 2 3 4 5 6	_2{1 2 3 4 5 6
3	5

【左引数が2なら先頭から3番目の要素、また _2の場合は末尾から2番目の要素】

]G=.:D=.5 2 4 7 1 8 3 6	G { D
4 1 6 2 0 7 3 5	1 2 3 4 5 6 7 8

【『/:』という演算子の片側形は昇順の「インデックス」を与えている。つまり G の左端の4は、Dの要素の最小値1は0から数えて4番目(通常の5番目)の位置にあることを示している。さらに 『/:~D』 = 『D/:D』は、『G{D』と同じ演算で、Dを昇順に並べ替える】

【第3四分位点(Q3)と第1・四分位点(Q1)の値を算出している】

```
(qdev=:[:(-/%+ /)][:+/weight*index{ }] "1[:(|.,:)]/:~)&> |  
0.5 0.5 0.5 0.5
```

【四分位偏差 (Q3 - Q1)/(Q3 + Q1) を算出している】

```
qdev&> MIN2;MIN3;MIN4  
0.487805 0.292308 0.108197
```

【2月、3月、4月の最低気温のデータの四分位偏差を算出している】

..... 「J」言語メモ

左右の引数に関係なく常に定数を取り出す関数は、0から9までと_1から_9までの整数に「コロンの(:)」をつけて定義できる。例えば『]-1:』は「1を引く」という演算子『<:』と同じになる。また「0.5を加える」という場合には『]+0.5"_』のように定義する。つまり「_」を適当な定数に付加すると、「その定数を取り出す」という動詞になる。

§ 1 8 回帰直線

行列の 割算行う パーセントピリ(%.) 片側形なら 逆行列

```
X=.43.05 38.25 35.68 34.68 31.57
```

【5つの都市(札幌・山形・東京・大阪・鹿児島)の緯度のデータ】

```
Y=_.8.9 _4.2 0.5 2.2 2.4
```

【5つの都市の1月の最低気温の平年値のデータ】

```
]X1=.(1:,.])X
```

```
1 43.05
```

```
1 38.25
```

```
1 35.68
```

```
1 34.68
```

```
1 31.57
```

【『, .』は『縦連結(Stitch)』という演算子で、引数のアイテムとの連結を行うから、1がXというリストの各要素ごとに連結し、結果は5 × 2のテーブルになる】

```
Y % . X1
```

```
38.2955 _1.08867
```

【回帰直線の切片と勾配を与えている】

```
Y (regline=.[%.1:,.]) X
```

```
38.2955 _1.08867
```

【上と同じ結果で、回帰直線の切片と勾配を与える】

```
SM=.98 138 115 83 88 117 128 154 123 142
```

```
SF=.65 86 86 61 70 82 78 80 78 72
```

【1950年から1995年までの5年間隔の男子と女子の自殺者数(単位千人)】

```
UE=.1.2 2.5 1.7 1.2 1.1 1.9 2.0 2.6 2.1 3.1
```

【対応する年次の完全失業率のデータ】

```
SM regline UE
```

```
SF regline UE
```

54.878 32.84664

62.8534 5.64257

【勾配の値から、失業率が0.1%上がるごとに男子の自殺者が328人増加する。また女子の場合は、失業率の変動によってそうは影響されない】

.....「J」言語メモ

regline

[%.

1: ,.]

という表示からも分かるように、「1:,.]」の部分がまずフォークで、さらにこの動詞が前2つの動詞と接続詞なしで連結されて再びフォークになる。

§ 19 分散共分散行列

棒にコロンの(:) 片側形なら アレイの軸の 順序をソックリ 入れ替える

```
BWH                                mean BWH
93 53 94                            87 58 91
84 61 88
84 55 91
```

【 BWH は 3 人の女性の左から、バスト(B) , ウェスト(W) , ヒップ(H)のデータ】

```
S; |:S=.(.)-"1 mean) BEH          V;(V=.(|:+/ .*]S)%#BWH

 6  0  3   6 _3 _3                54  0 27   18  0  9
_3  3 _3   0  3 _3                0 18 _9   0  6 _3
_3 _3  0   3 _3  0                27 _9 18   9 _3  6
```

【 S は 『"1』 とランク 1 を指定しているので、BWH のリストごとに平均を引いている。また、V は 『|:]』 による S の転置と S の “ 行列積 ” で、さらに V をデータのアイテム数で割った値は分散共分散行列になる】

そこで、分散共分散行列を求める関数を次のように定義する。

```
vcmat=:([:(|:+/ .*])-"1 mean)%#

vcmat BWH
18  0  9
 0  6 _3
 9 _3  6

vcmat SM, .SF, .UE
492.84 108.02 13.086
108.02  92.56  2.248
13.086  2.248  0.3984
```

【男子と女子の自殺者数と完全失業率のデータに対する分散共分散を示している】

..... 「 J 」 言語メモ

『|:』は、「転置(Transpose)」と呼ばれる演算子で、軸の順序を入れ替える

(:);|:)i.2 2

(|:);|:)i.2 2 2

0 1 0 2

2 3 1 3

0 1 0 4

2 3 2 6

4 5 1 5

6 7 3 7

ランク3のアレイでは、(0,1,2) (2,1,0)のように軸が入れ替わる。したがって、
「i.2 3 4」というアレイを転置したもののランクは『\$ |: i.2 3 4=4 3 2』となる。

§ 2 0 相関行列

DotProd(+/.*) ランク合わせて 行列算 チョー重宝な プリミティブ

```
]S=.(- "1 mean) BWH          mean *: S
6 0 3          18 6 6
_3 3 _3          %: mean *: S
_3 _3 0          4.24264 2.44949 2.44949
```

【変数ごとの平均からの偏差とその平方値の平均、そしてその平方根を与えている】

そこで、各変数ごとの標準偏差を求める関数と標準化する関数を次のように定義する。

```
stand:=(- "1 mean)"1 sdev [ sdev=:[%:[:mean[:*:] -"1 mean

stand BWH          ([:(|: +/ .*])stand) BWH
1.41421          0 1.22474          3 0 2.59808
0 1.22474 _1.22474          0 3 _1.5
_0.707107 _1.22474          0 2.59808 _1.5 3
```

【 BWH というデータを標準化したものと、その転置した行列との “ 行列積 ” を出力】

そこで、相関行列を求める関数を次のように定義する。

```
corm=:([:(|: +/ .*])stand)%#

corm BWH          corm SM, .SF, .UE
1 0 0.866025          1 0.504665 0.933887
0 1 _0.5          0 1 0.369393
0.866025 _0.5 1          0.933887 0.369393 1
```

【 BWH というデータに対する相関行列と、男子と女子の自殺者数と完全失業率のデータに対する相関行列を示している】

..... 「 J 」 言語メモ

```
(i.2 3)+/ .*(|:i.2 3)      (|:i.2 3)+/ .* i.2 3
5 14                        9 12 15
14 50                        12 17 22
                               15 22 29
```

【『+/ .*』という演算子は、行列同士の“掛け算”を行う】

```
$ M+/ .*|:M=.i.2 3 4      $(i.2 3 4)+/ .*i.4 2
2 3 3 2                    2 3 2
```

【ランク 3 以上の一般アレイに対しても演算が可能である】

```
$(i.2 3 4)+/ .*i.3 2
length error
```

【左の引数の最後のランクと、右の引数の最初のランクが一致しないとエラーになる】