(ＪＡＰＬＡ 2006/08/5-7)

「チュートリアル セミナー」

帝京平成大学　鈴木義一郎

§１　整数列の生成

integer=:>:@i.　NB.整数列を生成する(片側形の)関数

i.5	1+i.5	>:i.5	(>:i.)5
0 1 2 3 4	1 2 3 4 5	1 2 3 4 5	1 1 1 1 1

【Ｊ言語では、指標原点が常に０に設定されるので、「i.」は０から始まる連続した整数を、右引数で与えた個数だけ生成する。そこで、右引数「>:」は「１を加える」という演算子で、前の結果に１を加えている。ところが、カッコをつけてしまうと「フック」となり、「5>:i.5」のように演算する。「>:」の両側形は「≧」で、５が右側の要素のいずれよりも大きいから、論理演算は真で１になる】

>:i.4 5 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20	「i.4 5」は、０から始まる連続した２０(＝４×５)個の整数を「４×５」の行列で生成する。さらに「>:」で、１から始まるようにしてある。
>:i._5 5 4 3 2 1	引数に負の値を入力すると、逆順の整数列が得られる
i:5 _5 _4 _3 _2 _1 0 1 2 3 4 5 i:_5 5 4 3 2 1 0 _1 _2 _3 _4 _5	「i:」という演算子の片側形は、右引数で与えた数の負の値から正の値までの整数列を生成する。また負の値を入力した場合には、逆順の結果を出力する。

…………「Ｊ」言語メモ ……………………

【合成関数ｇ・ｆが両側形として機能する場合】

次のように、３通りの結合のし方が考えられる。そこでＪ言語では

ｘ(ｇ＠ｆ)ｙ＝ ｇ(ｘ ｆ ｙ)	ｆが両側形でｇが片側形の場合
ｘ(ｇ＆ｆ)ｙ＝(ｆ ｘ)ｇ(ｆ ｙ) ｘ(ｇ ｆ)ｙ ＝ ｘ ｇ(ｆ ｙ)	ｆが片側形でｇが両側形の場合 両側形の場合の「フック(hook)」

【合成関数ｇ・ｆが片側形として機能する場合】

（ｇ＠ｆ）ｙ＝（ｇ＆ｆ）ｙ＝ｇ（ｆ ｙ） （ｇ ｆ）ｙ　＝　ｙ ｇ（ｆ ｙ）	ｆもｇも片側形の場合 ｆが片側形でｇが両側形の場合

§２　総和

sum=:+/　NB. 総和を求める(片側形の)関数

]D1=:>:i.5 1 2 3 4 5	D1 というベクトル（Ｊ言語では「リスト」という）を定義し、「] 」によりその内容を表示している。
sum D1 15	D1 というリストの総和を与えている。
]D2=:>:i.4 5 1　2　3　4　5 6　7　8　9 10 11 12 13 14 15 16 17 18 19 20	D2 という行列（Ｊ言語では「テーブル」という）を定義し、その内容を表示している。
sum D2 34 38 42 46 50	D2 のアイテムについての総和で、したがって０軸(縦)方向の和を与えている。
sum"1 D2 15 40 65 90	D2 のランク１のセル、したがって１軸(横)方向の和を与えている

…………「Ｊ」言語メモ ……………………

Ｊ言語では､動詞､名詞のように､通常の言語と同様に｢品詞｣という概念が対応している例えば､変数として定義された｢D1｣は名詞であり､また｢sum｣という関数の定義に用いた｢+/」，「] 」は、「全ての要素を足す」，「右引数を取り出す」といった動詞である。

特に名詞に関しては､｢ランク｣という概念が対応している｡単独の数値｢1.1｣や文字｢a｣などは｢アトム｣と呼ばれ､数学の場合のスカラーに対応し､ランクは０である｡ランク１の名詞は｢リスト｣と呼ばれ､数学の(横)ベクトルに対応する｡また行列に対応するのが｢テーブル｣で､ランクは２である｡さらにランクが３以上のものも考えられ､一般に｢アレイ｣と呼ばれる。アレイの形は

```
   $ D2
4 5
```

といったように､｢$｣という演算子で与えられる｡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。（中村先生は「ケシカラン」と嘆いている！）

i.3 1 0 1 2	なおＪ言語には、「縦ベクトル」に対応するものはなく、「3 1」という形のテーブルである。

§３　平均

mean=:+/%#　NB. 算術平均を求める(片側形の)関数

D1	# D1	+/ D1
1 2 3 4 5	5 【D1 というリストの個数】	15 【D1 というリストの総和】

(+/D1)%(#D1)	mean D1
3 【「15÷5」の値を求めている】	3 【D1 というリストの(算術)平均の値】

D2	mean D2
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20	8.5 9.5 10.5 11.5 12.5 【D2 のアイテムについて、つまり０軸(縦)方向の平均を求めている】 mean"1 D2 3 8 13 18 【D2 というテーブルの１ー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ーセル」とか「ランクｋのセル」という。例えばD2 というアレイの「1 2 3 4 5」などは、１つの「１ーセル」である。またD2 の「２ーセル」はD2 自身で、１つしかない。「０ーセル」は必ずアトムで、「１ーセル」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「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ーセル」を引数とみなして演算を行なうことになる。　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」と呼んでいる。一般に

(fgh) 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 (fgh) 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(両側形の)動詞をオペレートする

といった手順で実行される（中央の動詞「ｇ」は両側形でなければならない！）。

§４　幾何平均と調和平均

```
meang=:#%:*/  NB. 幾何平均を求める(片側形の)関数
meanh=:[:%[:mean%  NB. 調和平均を求める(片側形の)関数
```

D1 1 2 3 4 5	# D1 5 【D1 の各要素の個数】	*/ D1 120 【D1 の各要素の積】

(#D1)%:(*/D1) 2.60517 【120(*/D1)の５(#D1)乗根を与えている】	meang D1 2.60517 【D1 というリストの幾何平均の値】

([:mean%) D1 0.456667	「%」という演算子で D1 の各要素の逆数を求め、その平均を求めている。[[:」は“何もしない”という動詞である。
([:%[:mean%) D1 2.18978	直上の結果の逆数を与えている。つまり、D1 の調和平均を求めている。
meanh D1 2.18978	「（[:gh）」とう形のフォークでは「g」という中の動詞は片側形である。

…………「Ｊ」言語メモ ……………………

「〇月〇日、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だから、ｈはｆという動詞の「逆演算」で、Ｊ言語では、「f:_1」で与えられる。そこで、関数「ｇ&. ｆ」が「（f:_1)@g&f」と同じ働きをするように、アンダーと呼ばれる接続詞「&.」を用意しておけば、h という動詞は特記する必要がなくなることになる。（残念なことに、新バージョンになってから、この「&.」の“動き”がオカシクなってしまった！）

§５　最大値

max=:>./　NB. 右引数の最大値を与える(片側形の)関数

D1 1 2 3 4 5	「>.」という演算子の両側形は、左右の引数の大きいほうの値を取り出す働きをする。従って、４と５では５のほうが大きいから５を出力することになる。
4 >. 5 5	
3 >. 4 >. 5 5	４と５では５のほうが大きく、さらに３と５とではやはり５が大きいから、結果は５。
1>.2>.3>.4>.5 5	やはり右から順次計算すると結果は５となり、結局、右引数の最大値を与えている。
>./ D1 5	「/」で、D1 のアイテム間に「>.」を挿入するから、右引数の「最大値」を与える。
max D1 5	ユーザーが定義した「代動詞」を用いても同じ結果が得られる。

D2 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20	max D2 16 17 18 19 20 【この結果は縦方向に関しての最大値を求めているように見えるが、D2 のアイテムは、「1 2 3 4 5」，……，「16 17 18 19 20」といったリストになるから、アイテム(この場合にはリスト)のなかでの最大のものを求めている】 max"1 D2 5 10 15 20

…………「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。つまり、同じ内容のものを「max」ではなく「saidaiti」と表わしても、最大値を求めるという働きは同じである。各原始動詞の働きは固有のものであるのに対して、代動詞の働きのほうは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、「左側の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=:+/」や「meang=:#%:*/」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。

§６　範囲

range=:>./-<./　NB. 範囲を与える(片側形の)関数

D1	>./D1	<./D1
1 2 3 4 5	5 【D1 の各要素の「最大値」】	1 【D1 の各要素の「最小値」】

(>./D1)-(<./D1)	range D1
4	4

【「<.」は「>. 」と対をなす演算子で、左右の引数の小さいほうの値を取り出す働きをする。】

```
   D2
 1  2  3  4  5
 6  7  8  9 10
11 12 13 14 15
16 17 18 19 20
   range D2
15 15 15 15 15
```

【D2 のアイテムはリストになるから、アイテムのなかでの最大のものと最小のものとの差を与えている】

```
   range"1 D2
4 4 4 4
```

【D2 のランク１のセル、したがってリストの要素(横方向)についての範囲を求めている】

…………「Ｊ」言語メモ ……………………

上で定義した関数に、引数を入力しないで「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。

§７　平均偏差

```
mdev=:[:mean[:|(-mean)   NB. 平均偏差(絶対偏差の平均)を求める関数
```

D1	
1 2 3 4 5	
(-mean)D1	D1 というデータの偏差を求めている。「(-mean)y」は「y-mean y」
_2 _1 0 1 2	のように演算を行なう「フック」である。
-mean D1	ただしカッコがないと、平均を求めてから符合を反対にする。
_3	
([:|(-mean))D1	「|」は「絶対値をとる」という演算子で、偏差の絶対値(絶対偏差)
2 1 0 1 2	を求めている。
([:mean[:|(-mean))D1	絶対偏差の平均で、結局「平均偏差」を求めている。
1.2	
mdev D1	ユーザーの定義関数で平均偏差を求めている。
1.2	

…………「Ｊ」言語メモ ……………………

「|」という演算子の片側形は「絶対値をとる」という機能をもつが、両側形で用いると

```
   2 | D1=:1 2 3 4 5
1 0 1 0 1
```

といったように、左引数で割算したときの「剰余」を与える。

また、割算の「商(割った結果の整数部）」を求めるには

```
   2 <.@%~ D1
0 1 1 2 2
```

のようにすればよい。さらに

```
   2([*<. @%~)D1
0 2 2 4 4
```

であるから

```
   2(|+[*<.@%~)D1
1 2 3 4 5
```

のように、元の D1 に戻ってしまう。つまり

(A mod B)＋A×[B/A] = B

という関係を示している。

§８　分散と標準偏差

```
var=:[:mean[:*:(-mean)  NB. 分散を求める関数
sdev=:[:%:var  NB. 標準偏差(分散の平方根)を求める関数
```

```
   T ; Z
```

80 70 60 90 100	50 60 70 80 90

【太郎と次郎の５回のテストの成績】

【２組のデータをボックスで囲んでいる。これは、「ボックスで囲む」という演算子「<」を用いて、「(<T),<Z」としても同じ結果が得られる】

```
   > T;Z
80 70 60 90 100
50 60 70 80  90
```
【「>」は「<」の逆演算で、「ボックスを開く」という働きをする】

```
    (-mean) T
0 _10 _20 10 20
```
【Tから平均を引いて、「偏差」を求めている】

```
([:*:(-mean) T
0 100 400 100 400
```
【偏差を平方している。「*:」の片側形は「平方値」を与える】

```
   ([:mean[:*:(-mean) T
200
```
【直上の結果(偏差平方値)の平均を求めている】

```
   var T
200
```
【上と同じ結果で、つまりＴというデータの分散を求めている】

```
   var&> T;Z
200 200
```
【２組のデータの分散を同時に求めている】

```
   sdev&> T;Z
14.1421 14.1421
```
【２組のデータの標準偏差を同時に求めている】

…………「Ｊ」言語メモ ……………………

「ｆ&>」という関数は、ボックスで囲まれたデータを右引数で与えると、「ｆ」という演算を各ボックスごとに行う(「earch」と呼ばれている)。さらに、演算結果をボックスで囲んだ形にしておきたければ、「ｆ&.>」のように「&.」という接続詞に替えてやればよい。

```
   var&.> T;Z
```

200	200

また

```
   var&> T ; i.10
200 8.25
```

のように、データの個数が違っていても大丈夫である。

§９　分類されたデータの平均と標準偏差

```
meanc =:+/@:*/%+/@:{:   NB. ０行と１行で与えた分類データの平均を求める関数
sdevc=:[:%:[:+/{:*[:*:{.-meanc  NB. ０行と１行で与えた分類データの標準偏差
```

```
   TEST
40 42 44 46 48 50 52 54 56 58 60
 1  3  7 12 17 20 17 12  7  3  1
```

【最初の行が級代表値で、最後の行が度数を示す分類された合計 100 個のデータ】

```
   */ TEST
40 126 308 552 816 1000 884 648 392 174 60
```

【アイテム間の積を与えている】

+/@*/ TEST	+/@(*/) TEST	+/@:*/ TEST
40 126 308 552 816 1000 884 648 392 174 60	5000	5000

【上の結果の総和で、分類されたデータの総和を求めている】

```
   (+/@:*/%+/@:{:) TEST
50
```

50【上の総和をデータの総数「+/@:{:」で割った値で、分類されたデータの平均値】

```
   meanc TEST
50
```

50【上と同じ結果で、「meanc」は分類されたデータの平均を求める関数である】

```
   ([:mean{:#{.)TEST
50
```

50【「{:#{.」によって、(1 行)の個数だけ(0 行)の値をコピーするので、結局、100 個の分類されてないデータの平均をとっても同じ結果が得られる】

```
   ({.-meanc)TEST
_10 _8 _6 _4 _2 0 2 4 6 8 10
```

【分類されたデータの級代表値の平均からの偏差】

```
   ([:*:{.-meanc)TEST
100 64 36 16 4 0 4 16 36 64 100
```

【平均からの偏差の平方値を与えている】

```
   ({:*[:*:{.-meanc)TEST
100 192 252 192 68 0 68 192 252 192 100
```

【偏差の平方値に、度数を掛けているフォーク

```
   ([:+/{:*[:*:{.-meanc)TEST
1608
```

【上の結果の合計値で、分類されたデータの偏差平方和を与えている】

```
   (([:+/{:*[:*:{.-meanc)%+/@:{:)TEST
16.08
```

【偏差平方和をデータの総数「＋／＠{：」で割った値で、分散の値を与えている】

```
   sdevc TEST
4.00999
```

【直上の結果の平方根で、「sdevc」は分類されたデータの標準偏差を与える】

([:sdev{:#{.)TEST
4.00999
【「{:#{.」によって、100個の分類されてないデータの標準偏差をとっても同じ結果】

§１０　累積度数法による平均と標準偏差

```
sub=:+/¥&.|.  NB. rsk1 rsk2 を求めるための補助関数
rsk1=:[:+/sub  NB. 第１累積度数の和を求める関数
rsk2=:[:+/[:}.sub^:2 NB. 第２累積度数の和を求める関数
meanr=:{.@[+{:@[*([:<:rsk1%+/)@] NB. 等間隔分類されたデータの平均を求める関数
sdevr=:[:%:*:@[*((+:@rsk2+rsk1([-*:@[%+/@]))])%+/)@] NB. 分類データの標準偏差
```

```
   ]F=:{:TEST
1 3 7 12 17 20 17 12 7 3 1【級代表値の小さいほうからの度数分布】
   (sub=:+/¥&.|.)F
100 99 96 89 77 60 40 23 11 4 1
【後方からの度数の逐次和で、第１累積度数が与えられる】
   ]S=:rsk1 F
600【上の第１累積度数の合計をＳに挿入し表示している】
   40 2({:@[*([:<:rsk1%+/)@])F
10【「(S÷N)-1」に左引数の末尾の要素(級間隔)を掛けた値である】
   40 2 meanr F
50【上の結果に左引数の先頭の要素(最小の級代表値)を足せば平均値を与える】
   sub^:2 F
600 500 401 305 216 139 79 39 16 5 1【第１累積度数の逐次和で、第２累積度数】
   ]T=:rsk2 F
1701【上の第２累積度数で先頭の要素を取り除いた合計値をＴに挿入し表示している】
   (+:T)+S([-*:@:[%])N=.+/F
```

402【「$2T+S-S^2/N=+/F$」という値を計算している】

```
   %:(*:2)*((+:T)+S([-*:@:[%])N)%N
```

4.00999【「$(2T+S-S^2/N)$÷Ｎ」に級間隔の２乗を掛けてから平方根をとった値】

```
   2 sdevr F
```

4.00999【上と同じ結果で、等間隔で分類されたデータの標準偏差を与えている】

等間隔に分類されたデータの度数｛$f_1,f_1,\cdots\cdots\cdots,f_k$｝が与えられたとき、平均(M)や分散(V)は第１累積度数$\{g_j\}$と第２累積度数$\{h_j\}$を用いて次式で与えられる：

$$S = \sum_{i=1}^{k} g_i \; (g_j = \sum_{i=j}^{k} f_i \,; 1 \le j \le k) \quad ; \quad T = \sum_{i=2}^{k} h_i \; (h_j = \sum_{i=j}^{k} g_i \,; 2 \le j \le k)$$
$$M = x_1 + w\{S/n - 1\} \quad ; \quad V = w^2\{2T + S - S^2/n\}/n$$

§１１　偏差値

```
sdev=:[:%:[:mean[:*:(-mean=:+/%#)  NB. 標準偏差を求める関数
hnst=:50"_+10"_*(-mean)%sdev  NB. 偏差値を求める関数
meanc =:+/@:*/%+/@:{:  NB. ０行と１行で与えた分類データの平均を求める関数
sdevc=:[:%:[:+/{:*[:*:{.-meanc  NB. ０行と１行で与えた分類データの標準偏差
hnstc=:50"_+10"_*({.-meanc)%sdevc  NB. 分類データの偏差値を求める関数
```

```
   ]T=:45+2*i.11
45 47 49 51 53 55 57 59 61 63 65【11 人の点数のデータ】
   (mean,sdev)T
55 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 55 を引いた値で、平均からの偏差を与えている】
   6{.((-mean)%sdev)T
_1.58114 _1.26491 _0.948683 _0.632456 _0.316228 0
【上の偏差の値を標準偏差で割った結果の 6 個目の値で、「データの規準化」と呼ぶ】
   50+10*Z
34.1886  37.3509  40.5132  43.6754  46.8377  50  53.1623  56.3246  59.4868  62.6491
65.8114
【Z の 10 倍した値に 50 を加えた数値】
   hnst T
34.1886  37.3509  40.5132  43.6754  46.8377  50  53.1623  56.3246  59.4868  62.6491
65.8114
   0.1": hnst T
34.2 37.4 40.5 43.7 46.8 50.0 53.2 56.3 59.5 62.6 65.8
【直上の値を、書式関数「“:」を用いて１桁で”四捨五入“した値で、数値ではなく文字)】
```

]TEST=:T,:1 3 7 12 17 20 17 12 7 3 1	(meanc,sdevc)TEST 55 40.0999

```
45 47 49 51 53 55 57 59 61 63 65
 1  3  7 12 17 20 17 12  7  3  1
```

```
   ({.-meanc)TEST
_10 _8 _6 _4 _2 0 2 4 6 8 10
```

```
   50+10*Z=:(({.-meanc)%sdevc)TEST
47.5062  48.005  48.5037  49.0025  49.5012  50  50.4988  50.9975  51.4963  51.995
52.4938
   hnstc TEST
47.5062  48.005  48.5037  49.0025  49.5012  50  50.4988  50.9975  51.4963  51.995
52.4938
   0.1":hnstc TEST
47.5 48.0 48.5 49.0 49.5 50.0 50.5 51.0 51.5 52.0 52.5
```

§１２　変動係数

```
   Sdev=:[:%:[:mean[:*:(-mean=:+/%#)  NB. 標準偏差を求める関数
   coefv=:[:100" _+ sdev%mean  NB. 変動係数を求める関数
   coef=:3 :' 100*(%:mean*:y.-m)%m=.mean y.' NB. 明示的定義による変動係数
```

　SM【男子自殺者数（５年間隔の年次データ）】

9820 13836 11506 8330 8761 11744 12769 15356 12316 13540

　SW【女子自殺者数】

6491 8641 8637 6114 6967 8231 7773 8027 7772 6976

```
   mean L:0 SM;SW
```

11830.9	7562.9

【女子の自殺者のほうが少ない】

```
   sdev L:0 SM;SW
```

2334.55	837.462

【男子の自殺者数のほうの変動が激しい】

```
   100*(sdev%mean)SM
19.7327
```

【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 SM
19.7327
   coefv SW
11.0733
```

```
   coef SM
19.7327
   coef SW
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】

…………「Ｊ」言語メモ ……………………

これまでは、原子動詞や副詞等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、「coef」といった動詞を定義するときには、

coef=: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。（変数の文字の後に「ピリオド(.)」をつけないとエラーになる。）

§１３　メディアン

```
median=:[:-:@+/([:(<:,-)[:>.-:@#){/:~  NB. メディアンを求める関数
```

```
   /:~D=:5 2 4 7 1 8 3 6
1 2 3 4 5 6 7 8
```
【Dというリストを「/:~」により昇順形に並べている】

```
   ]M=:([:>.-:@#)D
4
```
【Dのアイテム数(個数)を半分にしてから、小数点以下を切捨ててＭに挿入する】

```
   ]I=:(<:,-)M
3 _4
```
【Ｍというインデックスの１を引いた値と符号を変えた値とを連結している】

```
   I{/:~D
4 5
```
【Dの昇順形の先頭から４番目と末尾から４番目の要素を取り出している(先頭からの軸の指定は、３軸が通常の４番目になる！)】

```
   -:@+/ 4 5
4.5
```
【右引数の要素を足してから「-:」で半分にしている。結局、平均を求めている】

```
   median D
4.5
```
【上と同じ結果で、「median」はデータの昇順形の４番目と５番目の要素の平均】

```
   median D,9
5
```
【奇数個のデータの場合には中央の(この場合は５番目の)要素を与えることになる】

```
   median >:i.10
5.5
```
【５番目と６番目の要素の平均を与えている】

…………「Ｊ」言語メモ ……………………

「/:」という演算子は、片側形で用いると

```
   /:D
4 1 6 2 0 7 3 5
```

のように昇順の「インデックス」を与えている。つまりこの結果の左端の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。

「{」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 3【先頭から３番目の要素】	_2{1 2 3 4 5 6 5【末尾から２番目の要素】

といったように、左引数で指定した数値の要素を右引数のリスト(一般にはアレイ)から選択する。ここで左引数がプラスの場合には、２は０から数えているので、先頭から３番目の要素が選択される。(「選択(from)」の左引数に、正数と負数を与えた場合には、順番に整合

性を欠いている点に注意を要する。）

§１４ 四分位数と四分位偏差

```
qtl1=:-:@+/@(>.@(<:,-&0.5)@(#%4:){/:~)  NB. 「第１四分位数」を与える関数
qtl3=:-:@+/@(-@>.@(],+&0.5)@(#%4:){/:~)  NB. 「第３四分位数」を与える関数
qtl=:qtl3-qtl1)%2;  NB. 右引数で与えたデータの四分位偏差を与える関数
```

```
   (#%4:)D=:5 2 4 7 1 8 3 6
2
```

【「4:」は左右の引数に関係なく、常に「4を取り出す」という動詞である。従って、「#%4:」はフォークで、「(#D)%4＝8%4＝2」のように演算する】

```
   ]A=:((#%4:){/:~)D
```

3【Dの昇順形の２番目（通常の３番目）の要素を選択してAに挿入している】

```
   (<:,-&0.5)A
```

2 2.5【3から１を引いた値と0.5引いた値とを取り出している】

```
   >.@(<:,-&0.5)A
```

2 3【上の値を「>.」という演算子で切り上げて整数にしている】

```
   1 2{/:~D
2 3
   -:@+/ 1 2{/:~D
```

2.5【Dの昇順形の1、2番目(通常の2、3番目)の要素を足してから半分にしている】

```
   -:@+/(>.@(<:,-&0.5)@(#%4:){/:~)D
2.5
   qtl1 D
```

2.5【上と同じ結果で、Dの昇順形の2、3番目の要素の平均で「第１四分位数」を与える】

```
   qtl1 >:i.10
```

3【データ数が10個のときには、小さいほうから３番目の要素が「第１四分位数」になる】

```
   qtl3 D
```

6.5【「第3四分位数」を求めるプログラムも同様に定義でき、結果は上記のようになる】

```
   qtl D
```

2【第3四分位数と第１四分位数の差の半分が「四分位偏差」である】

…………「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(:)」をつけて定義できる。しかし「0.5」を引く場合には「-&0.5」のように記述しなければならない。また、任意の数字を“動詞化”するには「10“_」や「o.2”_」といったよ

うにすればよい。

§１５ 平均差

```
meandif=:([:+/[:|[:,-/~)%2:*2:!#  NB. 右引数で与えたデータの平均差を与える
```

D1 1 2 3 4 5]A=:(2:*2:!#)D1 20	-/~ D1 0 _1 _2 _3 _4 1 0 _1 _2 _3 2 1 0 _1 _2 3 2 1 0 _1 4 3 2 1 0

【「2！5」」は２項係数「$_5C_2$」を与える演算子で、「2:*」で結果を 2 倍にしている】

【「-/~D1」は「D1-/D1」と同じで、D1 の各要素の引き算に関するクロス表が与えられる。なお「*/~>:i.9」とすると「九九の表」が得られる】

```
   ]B=:([:+/[:|[:,-/~) D1
40
```
【上の行列を「,」でベクトル化し、「|」で絶対値をとってから加えている】

```
   B % A
2
   meandif D1
2
```
【D1 というデータの「平均差係数」を与えている】

```
   meandif 2+D1
2
```
【データ全体をずらしても、平均差係数の値は変わらない（“係数”という表記は変だ！）

```
   sdev D1
1.41421
```
【平均差は標準偏差と同じような値になる】

```
   meandif 2.8 2.9 3 3.1 3.2
0.2
```
【変動幅が 10 分の 1 になれば、平均差の値も 10 分の 1 になる】

```
   sdev 2.8 2.9 3 3.1 3.2
0.141421
```
【標準偏差の値も 10 分の 1 になる】

…………「統計学」メモ ……………………

ｎ個のデータ $\{x_i, x_i, \cdots\cdots\cdots, x_n\}$ が与えられたとき

$$d = \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j=1}^{n}|x_i - x_j|$$

によって定義されるものを「平均差(係数)」と呼んでいる。標準偏差は“平均からのずれ”の平均であるが、この平均差のほうは“各データ間のずれ”の平均である。

§１６ 級間分散と級内分散

```
var=:[:mean*:@(-mean=:+/%#)  NB. 分散を求める関数
weight=:[:(%+/)#&>  NB. ウエイトを与える関数
varb=:[:+/weight*[:*:mean&>-mean@;  NB. 級間分散を求める関数
varw=:[:+/weight*var&>  NB. 級内分散を求める関数
```

KT;KZ【太郎と次郎の国語の成績(架空のデータ)】

80 70 60 90 100	50 60 70 80 90

#&>KT;KZ

5 5【2 組のデータ(リスト)の個数(アイテムの数)を与えている】

(%+/) #&> KT;KZ

0.5 0.5【２組のデータのウェイトを与えている。「%+/」はフックである】

mean&> KT;KZ 80 70【２組のデータの平均】	;KT;KZ 80 70 60 90 100 50 60 70 80 90

【２組のデータを合わせてリストにしている。「;」は「,@:＞」と同じである】

mean@; KT;KZ

75【複合データの平均】

(mean&>-mean@;)KT;KZ

5 _5【各組の平均から、複合データの平均を引いている。これもフォークである】

(weight*[:*:mean&>-mean@;)KT;KZ

12.5 12.5【上の結果の平方値にデータ数のウェイトを掛けている】

varb KT;KZ

25【直上の結果の合計で、「varb」は複数組のデータの級間分散を求める関数である】

Var&>KT;KZ 200 200【２組のデータの分散】	(weight*var&>)KT;KZ 100 100【分散の値にウエイトを掛けている】

varw KT;KZ

200【２組のデータの級内分散を求めている】

…………「統計学」メモ ……………………

m 個のデータの平均を $M(x)$，分散を $V(x)$ 、n 個のデータの平均を $M(y)$，分散を

$V(y)$ とするとき２組のデータを合わせた複合データの平均 $M(z)$ と分散 $V(z)$ は

$$M(z) = \alpha M(x) + (1-\alpha)M(y)$$
$$V(z) = \alpha V(x) + (1-\alpha)V(y) + \alpha\{M(x)-M(z)\}^2 + (1-\alpha)\{M(y)-M(z)\}^2$$

のように表される。ここで、$\alpha = m/(m+n)$ である。

§１７ 相関比

```
varb=:[:+/weight*[:*:mean&>-mean@;  NB. 級間分散を求める関数
varw=:[:+/weight*var&>  NB. 級内分散を求める関数
cratio=:varb([%+]varw  NB. 複数組のデータの相関比を求める関数
```

KT;KZ【太郎と次郎の国語の成績】

80 70 60 90 100	50 60 70 80 90

ST;SZ【太郎と次郎の数学の成績】

78 79 82 81 80	70 69 71 68 72

(mean,var)&.> KT;KZ;ST;SZ

80 200	70 200	80 2	70 2

【４組のデータの平均と分散を求めている】

varw KT;KZ 200	varb KT;KZ 25	var@; KT;KZ 225

【級内分散と級間分散の和は全体の分散と一致する】

25([%+)200

0.111111

【25%(25+200)の値を求めている】

cratio KT ; KZ 0.111111	cratio ST;SZ 0.925926

【数学の成績のほうが顕著な違いがある】

…………「統計学」メモ ……………………

「相関比」とは

$$H = \frac{BV\text{（級間分散）}}{WB\text{（級内分散）}+BV\text{（級間分散）}}$$

のように定義され、平均値間の差の相対的な尺度を与えるものである。

```
cratio KT ; KZ
0.111111
```

【国語のテストの成績では、太郎と次郎の間にはそれ程の差はない】

```
cratio ST;SZ
0.925926
```

【数学のテストの成績では、太郎と次郎の間には顕著な違いがある】

§１８ 順序尺度データの散布度

```
devdif=:[:+/}:*[:}.(+/¥^:2)&.|.   NB. u-関数を求める補助関数
ordv=:devdif%2:!+/   NB. 順序尺度データの散布度（ｄ）を求める関数
```

```
   F=:7 1 2 1
```
【力士の連敗のパターン分布で、勝ちが７、単敗、２連敗、３連敗が１，２，１度ある】
```
   ]H=:+/¥^:2&.|.F
19 8 4 1
```
【度数分布Ｆの第２累積度数を求めている】
```
   (}:F)*}.H
56 4 2
```
【Ｆの先頭の要素を落したリストとＨの末尾の要素を落したリストの積】
```
   ]U=:+/(}:F)*}.H
62
```
【上のリストの総和で、「ｕ関数」を求めてＵに挿入し表示している】

devdif F	]C=:(2:!+/)F	U % C
62	55	1.12727

【Ｕの値を異なる組合せの総数「n×(n−1)÷2」の値で割っている】
```
   ordv F
1.12727
```
【右上と同じ結果で、「ordv」は順序尺度データの散布度（ｄ）を求める関数である】

…………「統計学」メモ ……………………

順序尺度のカテゴリカルデータが与えられている場合に、ｉ番目のカテゴリーの観測度数を $\{f_i\}$ とする（i＝0,1, ……… ,m）。隣あったカテゴリー間の距離を１と考えると、「平均差」に対応するものは

$$d = \frac{1}{n(n-1)} \sum_{i \neq j}^{n} |i - j| f_i f_j$$

によって定義される。

そこで、第１、第２累積度数を

$$g_j = \sum_{i=j}^{k} f_i \; (1 \le j \le k) \quad ; \quad h_j = \sum_{i=j}^{k} g_i \; (2 \le j \le k)$$

のように定義すると、

$$\boxed{d = \frac{u}{n(n-1)} \quad ; u = \sum_{j=0}^{m} \sum_{j+1}^{m} (i-j) f f_j = \sum_{k=1}^{m} f_{k-1} h_k} \quad (n = \sum_{i-1}^{k} f_i)$$

で与えられるものが、順序尺度データの散布度を表わすことになる。

§１９ カテゴリカルデータの散布度

```
catu=:1:-+/@:*:%[:*:+/  NB. U係数を求める補助関数
catv=:[:%:(#%+/)*catu  NB. カテゴリカルデータの散布度(V)を求める関数
```

TAKA=:7 3 2 1 1【貴乃花の平成８年初場所の決まり手の分布(カテゴリー間に順序なし！)】

　]A=:([:*:+/)TAKA

196【TAKAの要素の総和(n)の平方値を求めている】

　]B=:([:+/*:)TAKA

64【TAKAの各要素の平方値の総和を求めている】

　A % B

0.326531

　(+/@:*:%[:*:+/)TAKA

0.326531【TAKAの要素の平方和(A)を要素の総和の平方値(B)で割る「フォーク」である】

　]U=:catu TAKA

0.673469

【TAKAというカテゴリカルデータの散布度(U係数)を与えている】

　(#%+/)TAKA

0.357143【TAKAのアイテム数を要素の総和で割るフォークである】

　]C=:%:@(#%+/)TAKA

0.597614

【上で求めた値、つまり「Ｋ(カテゴリー数)/Ｎ(データの総和)」の平方根を求めている】

　(%:@(#%+/)*catu)TAKA

0.490433【「catu」を演算した結果に$\sqrt{K/N}$を掛けた値である】

　catv TAKA

0.490433【上と同じ結果で、「U係数」を修正した「V係数」を与えている】

…………「統計学」メモ ……………………

ｉカテゴリーの観測度数数を$\{f_i\}$とする(i＝01,2, ……… ,k)とき、

$$\boxed{U = 1 - \frac{1}{n^2}\sum_{i-1}^{k} f_i^2} \quad (n = \sum_{i-1}^{k} f_i)$$

といったものが、平均差に対応する散布度と解釈できる。

しかし、観測数が多いときには、次のように修正した係数を用いるほうがよい。

$$V = \frac{\sqrt{k}}{\sqrt{n}}\left\{1 - \frac{1}{n^2}\sum_{i-1}^{k} f_i^{\,2}\right\}$$

【関数の script】

```
integer=:>:@i.  NB.整数列を生成する(片側形の)関数

sum=:+/  NB. 総和を求める(片側形の)関数
mean=:+/%#  NB. 算術平均を求める(片側形の)関数
meang=:#%:*/  NB. 幾何平均を求める(片側形の)関数
meanh=:[:%[:mean%  NB. 調和平均を求める(片側形の)関数
max=:>./  NB. 右引数の最大値を与える(片側形の)関数
range=:>./-<./  NB. 範囲を与える(片側形の)関数
mdev=:[:mean[:|(-mean)  NB. 平均偏差(絶対偏差の平均)を求める関数
var=:[:mean[:*:(-mean)  NB. 分散を求める関数
sdev=:[:%:var  NB. 標準偏差(分散の平方根)を求める関数
meanc =:+/@:*/%+/@:{:  NB. 0行と1行で与えた分類データの平均を求める関数
sdevc=:[:%:[:+/{:*[:*:{.-meanc  NB. 0行と1行で与えた分類データの標準偏差
sub=:+/¥&.|.  NB. rsk1 rsk2 を求めるための補助関数
rsk1=:[:+/sub  NB. 第1累積度数の和を求める関数
rsk2=:[:+/[:}.sub^:2 NB. 第2累積度数の和を求める関数
meanr=:{.@[+{:@[*([:<:rsk1%+/)@] NB. 等間隔分類されたデータの平均を求める関数
sdevr=:[:%:*:@[*((+:@rsk2+rsk1([-*:@[%+/@]))])%+/)@] NB. 分類データの標準偏差
hnst=:50”_+10”_*(-mean)%sdev  NB. 偏差値を求める関数
hnstc=:50”_+10”_*({.-meanc)%sdevc  NB. 分類データの偏差値を求める関数
coefv=:[:100”_+ sdevc%mean  NB. 変動係数を求める関数
median=:[:-:@+/([:(<:,-)[:>.-:@#){/:~  NB. メディアンを求める関数
qtl1=:-:@+/@(>.@(<:,-&0.5)@(#%4:){/:~)  NB. 「第1四分位数」を与える関数
qtl3=:-:@+/@(-@>.@(],+&0.5)@(#%4:){/:~)  NB. 「第3四分位数」を与える関数
qtl=:qtl3-qtl1)%2;  NB. 右引数で与えたデータの四分位偏差を与える関数
meandif=:([:+/[:|[:,-/~)%2:*2:!#  NB. 右引数で与えたデータの平均差を与える
weight=:[:(%+/)#&>  NB. ウエイトを与える関数
varb=:[:+/weight*[:*:mean&>-mean@;  NB. 級間分散を求める関数
```

```
varw=:[:+/weight*var&>  NB. 級内分散を求める関数
cratio=:varb([%+]varw  NB. 複数組のデータの相関比を求める関数
devdif=:[:+/}:*[:}.(+/¥^:2)&.|.  NB. u-関数を求める補助関数
ordv=:devdif%2:!+/  NB. 順序尺度データの散布度(ｄ)を求める関数
catu=:1:-+/@:*:%[:*:+/  NB. U係数を求める補助関数
catv=:[:%:(#%+/)*catu  NB. カテゴリカルデータの散布度(V)を求める関数
```