## 」言語による

## 行列数式表現の回帰分析のプログラム

一般的に回帰分析は、目的変数 y と説明変数 x との間に以下のような関係が成り立っているとすると、

$$\begin{cases} y_{i} = \beta_{0} + \beta_{1}x_{i1} + \dots + \beta_{j}x_{ij} + \dots + \beta_{m}x_{im} + \epsilon_{1} \\ \vdots \\ y_{i} = \beta_{0} + \beta_{1}x_{i1} + \dots + \beta_{j}x_{ij} + \dots + \beta_{m}x_{im} + \epsilon_{i} \\ \vdots \\ y_{n} = \beta_{0} + \beta_{1}x_{i1} + \dots + \beta_{j}x_{ij} + \dots + \beta_{m}x_{im} + \epsilon_{n} \end{cases} \dots \dots (1)$$

ここで、「 $\varepsilon_i$ 」によって、表される項は、線形関係だけでは説明できない諸々の変動部分を吸収してると考えられる誤差項と呼ばれていて、

 $\lceil \sum \varepsilon_i = 0 \rfloor$  と仮定している。

【参考文献:1996 森北出版 J 言語による統計学的叡分析 P102 鈴木義一郎著】 まず、現データを次のように定義する

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1j} & \cdots & x_{1m} \\ 1 & x_{21} & x_{22} & \cdots & x_{2j} & \cdots & x_{2m} \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ 1 & x_{i1} & x_{i2} & \cdots & x_{ij} & \cdots & x_{im} \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nj} & \cdots & x_{nm} \end{bmatrix}$$
  $\Rightarrow \&V \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{bmatrix} \dots (2)$ 

(2)式に実際のデータを当てはめると、以下のようになる。

保険会社売上高(昭和61年4月~1年)				
会社名	売上	外務員数	内務員数	
日本	332	759	137	
第一	209	568	113	
住友	208	612	106	
明治	128	385	81	
朝日	106	335	81	
三井	85	257	54	
安田	79	160	67	
太陽	64	133	34	
東邦	52	77	38	
協栄	42	134	24	
第百	40	100	29	
千代田	40	111	30	
大同	35	47	22	
布告	27	93	28	
団体	15	34	20	

$$X = \begin{bmatrix} 1 & 759 & 137 \\ 1 & 568 & 113 \\ 1 & 612 & 106 \\ 1 & 385 & 81 \\ 1 & 335 & 81 \\ 1 & 257 & 54 \\ 1 & 160 & 67 \\ 1 & 133 & 34 \\ 1 & 77 & 38 \\ 1 & 134 & 24 \\ 1 & 100 & 29 \\ 1 & 111 & 30 \\ 1 & 47 & 22 \\ 1 & 93 & 28 \\ 1 & 34 & 20 \end{bmatrix}, Y = \begin{bmatrix} 332 \\ 209 \\ 208 \\ 128 \\ 106 \\ 85 \\ 79 \\ 42 \\ 40 \\ 35 \\ 27 \\ 15 \end{bmatrix}$$
 .....(3)

このデータをもとに、重回帰分析による売上高を推定するモデルを策定してみる。

行番	J Script By j64-807 Android	
001	NB.重回帰分析	
002	NB. By R.Toribe	==重回帰のデータ部分===
003	n=:15 NB.系列数	
004	m=:2 NB.Val 数	г3321
005	NB.目的変数データの定義	209
006	y1=:332 209 208 128 106 85	208
007	y2=:79 64 52 42 40 40 35 27 15	128
008	Y=:15 1\$y1,y2	106
009	NB.説明変数データ・各行のの定義	85
010	x01=:759 137	79
011	x02=:568 113	$Y = \begin{bmatrix} 64 \\ 52 \end{bmatrix}$
012	x03=:612 106	52
013	x04=:385 81	42 40
014	x05=:335 81	40
015	x06=:257 54	35
016	x07=:160 67	27
017	x08=:133 34	15
018	x09=:77 38	
019	x10=:134 24	[1 759 137
020	x11=:100 29	1 568 113
021	x12=:111 30	1 612 106 1 385 81
022	x13=:47 22	1 385 81 1 335 81
023	x14=:93 28	1 257 54
024	x15=:34 20	1 160 67
025	NB.説明変数・各行を結合して行列を作る	$X = \begin{bmatrix} 1 & 133 & 34 \end{bmatrix}$
026	NB.J 言語は行列演算が極めて得意です	1 77 38
027	X=:(n,m)\$x01,x02,x03,x04,x05,x06,x07,x08	1 134 24
028	,x09,x10,x11,x12,x13,x14,x15	1 100 29
029	NB.説明変数に常数項を加える	1 111 30 1 47 22
030	NB.常数項を加えると、予測の精度が上がる	1 93 28
031	X1=:1&, "1 X	$\begin{bmatrix} 1 & 33 & 20 \\ 1 & 34 & 20 \end{bmatrix}$
032	NB.」言語のプログラムは、行列表記の計算式で	1 31 20
033	NB.幾つかの論理的規則に従って、作成すれば簡潔	
034	に出来る。	===重回帰の要部分===
035	NB.」言語で計算した結果	_
036	NB.行列 X の転置行列を作る	  X' :Xの転置行列
037	X1t=: :X1	IN I NV/TAIE 11/1
038	NB.行列 X と X の転置を掛ける(分散共分散)	
039	X1tX1=:X1t +/ . * X1	X'X :分散共分散行列
040	NB.X の分散共分散行列の逆行列を作る	
041	InvX1tX1=:%.X1tX1	X'X  <sup>-1</sup> :逆行列計算
042	NB.重回帰分析の予測式のパラメータを推定する	<b>□帰係数推定</b>
043	NB.回帰係数を計算する	

```
Beta=:(%.(X1t +/ . *X1)) +/ . *(X1t +/ .
                                             \begin{bmatrix} \boldsymbol{b_0} \\ \boldsymbol{b_i} \end{bmatrix} = [X'X]^{-1}X'Y
044
    *Y)
045
                                            NB. 予測パラメータは Beta(j=0,1,…,m の順に表
046
047
    示されている)
    MultiReg=:3 : 0
048
    NB.ppq=: 9 |: 10
049
    NB.pps=: 9 |: 11
050
                                            =====出力部分======
051
    NB.
052
    wr =: 1!:2&2
                                            1!:2&2 は出力の外部関数
053 NB.計算
   │a=: '★重回帰分析・モデル出力★ '
054
055 wr a
056 | mdl 1=:'・線形回帰モデル式'
057 mdl 2=: ' Y=\beta0+\beta1X1+\beta2x2+.....'
   b=:'・目的変数=売上高(v)'
058
059
   wr Y
060 c=: '説明変数=変数 1(x1)、変数 2(x2)、変数
061
   3(x3)'
                                               出力文字等を定義
062 wr c
063 | wr ":X1
064 d=: '★推定回帰係数・常数項設定モデル★'
065 NB.常数項1の2
066 e=: '推定バラメータ'
067
   wr e
068
   | txt=:'β0= ';'β1= ';'β2= ';'β3= '
069 txt2=: '常数項のパラメータ'; '第1項のパラメー
   タ ';'第2項のパラメータ ';'第3項のパラメー
070
    ター
071
072
                                            ===係数出力部分=====
    NB.バラメータ数だけ」を繰り返し出力
073
074
   m=:m+1
                                            while.\simend.
075
    mc=:0
                                              m > mc の間 loop
076
    while. m > mc
                                              txt2のmc 項目
077
      do. q=:>mc{ txt2
078
      wr q,8j3":mc{ Beta
                                              Betaのmc 項目を出力
079
      mc = : mc + 1
                                              8i3 は出力形式
080
   end.
                                              Fortran の F8.3 と同様
081
082 f=: '★計算終了'
                                            ===========
083 | wr f
084
                                            ==ここから実際に出力された
001
   MultiReg ''
002 ★重回帰分析・モデル出力★
                                            部分==
003
    332
```

```
004
    209
005
    208
006
   128
007
   106
800
    85
009
    79
010
    64
                                            目的変数Y
011
   52
012
   42
013
    40
014
    40
015
   35
016
   27
017
   15
   常数項(x0)、変数 1(x1)、変数 2(x2)
018
019 | 1 759 137
020
   1 568 113
021
   1 612 106
022 1 385 81
023
   1 335 81
024 1 257 54
025 | 1 160 67
026 1 133 34
                                             常数項と説明変数
027 1 77 38
028 1 134 24
029 1 100 29
   1 111 30
030
   1 47 22
031
   1 93 28
032
033 1 34 20
                                          ==推定された重回帰係数=
   |推定バラメータ
034
035 常数項のパラメータ _6.781
036 第1項のパラメータ 0.296
037 第 2 項のパラメータ 0.505
038 ★計算終了
```

本重回帰分析の基本は、統計や数学の教科書を参考にして頂きますが、

とりあえず、式(2)、(3)から下記の式(4)を導きます。

$$\begin{bmatrix}
\hat{\beta}_{0} \\
\hat{\beta}_{1} \\
\vdots \\
\hat{\beta}_{m}
\end{bmatrix} = \begin{bmatrix}
n & \sum x_{11} & \cdots & \sum x_{1m} \\
\sum x_{11} & \sum x_{22}^{2} & \cdots & \sum x_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
\sum x_{mm} & \sum x_{m2} & \cdots & \sum x_{mm}^{2}
\end{bmatrix}^{-1} \begin{bmatrix}
\sum y \\
\sum x_{1}y \\
\vdots \\
\sum x_{n}y
\end{bmatrix} \quad \dots \quad (4)$$

そのためには、式(1)をもとに、β で偏微分して、Y の実測値と推定値の差の自乗が最小値 (ゼロ)のいなるように β を推定する方程式が式(4)となります。

ここで、要となるのは、式) 4) の、下記をとり出し式 (5) としてこれを眺めると、

$$\begin{bmatrix} n & \sum x_{11} & \cdots & \sum x_{1m} \\ \sum x_{11} & \sum x_{22}^{2} & \cdots & \sum x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \sum x_{mm} & \sum x_{m2} & \cdots & \sum x_{mm}^{2} \end{bmatrix} \dots \dots \dots (5)$$

この式は、式(3)の X 転置行列 X' と X の積であることが分かります。これは一般に分散共分散行列です。

また、式(4)の、

$$\begin{bmatrix} \sum_{x_1 y} y \\ \sum_{x_n y} \vdots \\ \sum_{x_n y} \end{bmatrix} \dots \dots \dots (6)$$

これは、(3)の X 転置行列 X' と Y の積で出来ています。これを、式(6)とします。

つまり、式(6)は、以下のようになります。

$$\begin{bmatrix} \sum y \\ \sum x_1 y \\ \vdots \\ \sum x_n y \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{m1} \\ 1 & x_{21} & \cdots & x_{m2} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & x_{n1} & \cdots & x_{mn} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \cdots \cdots (6)$$

これらをまとめると、重回帰分析の行列表記は、以下のようになります。

$$X'Y = X'X\beta$$

から、

$$\beta = [X'X]^{-1}X'Y\cdots\cdots(7)$$

が導かれる。

つまり、J 言語では式(7) があれば、プログラムが書ける。

2019/08/01 原稿作成:坂戸にて。

2019/08/02 報告会場: 亀有・明神の湯にて。塩分のあるいい湯でした。