

统计计算
FORTRAN 程序集

下 册

辽宁省计委计算中心
辽宁省气象研究所
沈阳计算机厂

统计计算FORTRAN语言程序集

前 言

在电子计算机诞生的四十年来，它已被广泛用于数值计算、信息处理和自动控制等方面，使用范围也日趋扩大。电子计算机以惊人的速度渗透到工业、农业、国防、航天、教育、商业、气象、地质、医疗、文艺、体育、交通运输、科学研究，甚至于日常生活等领域。

现代化的社会是信息化社会。人们为了充分利用各种情报，要在“浩如烟海”的信息海洋中，找出我们所需的信息，并将它们传出去，用以往的手工方式就不行了，那样就犹如大海捞针，并贻误时机。所以，人们必须借助电子计算机技术和现代通讯技术，来实现现代信息管理系统，因此，凡是有信息的地方，就有电子计算机，整个人类社会已经与电子计算机密切接近了。

在计算机问世以后的十年左右的时间中，出现了FORTRAN (FORMULA TRANSLATOR，即公式翻译)语言，已广泛用于科学计算、数理统计等许多领域中。FORTRAN语言在科学和统计计算的应用过程中积累了许多算法，有通用的算法，也有针对解决某一类专门问题而编制的程序包。显然把这些算法汇编成册，不仅可以避免重复性劳动，提高计算机的使用效率，而且可以缩短计算工作的周期，有利于计算机的推广使用。我们编写这本书，主要是在统计计算方面，通过几年来的工作实践，我们从大量的多元统计资料中收集整理并编写了大量有实际应用价值的程序。该套程序集，全部都在计算机上调试通过的，而且大部分程序已应用到实践中，取得了良好的效果。

本程序集共分十一章，包括六十多个通用程序。编写格式均包括下列几个部分：

一、数学方法描述

这一部分主要是简要叙述该程序所使用的计算方法或者计算步骤以及过程中所做的某些处理，并列出具体的计算公式。我们的叙述力求简明，不去追求数学上的完整性。对于想进一步了解所用的计算方法的读者，可在给出的参考资料中找到它们的出处。

二、功能

简单地叙述该程序的用途和适用范围。

三、程序使用说明

这是程序说明的重点，力图对于如何使用该程序有一个完整而准确的叙述。

者清楚怎样使用。在这一部分中共分二项，分别介绍如下：

1. 输入参数：列出程序中所需要的原始数据的标识符的名称和它们所代表的意义。

2. 输出参数：列出程序中运算的输出结果标识符的名称和它们所代表的意义。

四、源程序

给出用FORTRAN语言编写的程序。为了能使更多的计算机使用程序集中的程序，程序均按基本FORTRAN语言编写。任何一台计算机只要配有基本FORTRAN语言或者标准FORTRAN语言编译程序，均可使用。但是输入、输出语句和文件管理系统语句依赖于计算机操作系统。所以读者可以根据所使用的计算机的具体规定稍加改动便可使用本程序集中各类程序。

五、例题

给出具体的例子和计算结果，说明怎样使用该程序。编写例题的出发点是帮助读者理解该程序的使用方法和检验程序之用。

虽然所有程序都在计算机上进行了运行和验算，但由于水平有限，书中难免有错误和问题，请读者批评指正。

本书全文由中国科学院沈阳计算技术研究所付所长、付研究员朴致淳，付研究员李木等同志审阅，并提出了宝贵意见，在此表示衷心感谢。

编著者 王兴玉 冯耀煌

编辑 卢毅

一九八四年十二月

下 册

第六章 判别分析	1
§ 6.1 两组判别分析程序.....	1
§ 6.2 多类训练迭代判别分析程序.....	8
§ 6.3 多级逐步判别程序.....	16
§ 6.4 从因子初选到精选 (逐步判别) 程序.....	36
第七章 变量场分析	55
§ 7.1 建立丰比雷夫系数场程序.....	55
§ 7.2 建立时间权重系数场程序.....	61
§ 7.3 按车氏展开做变量场的预报程序.....	73
§ 7.4 按自然正交函数展开做变量场的预报程序.....	91
§ 7.5 趋势面分析程序.....	112
第八章 多因素分析	131
§ 8.1 因子分析程序.....	131
§ 8.2 对应分析程序.....	150
§ 8.3 主成分分析程序.....	160
第九章 时间序列分析	179
§ 9.1 纯自回归模型的建模程序.....	179
§ 9.2 ARMA (n, m) 的建模程序.....	184
§ 9.3 ARMA (p, q) 模型的平稳预报程序.....	210
§ 9.4 非线性门限自回归模型的建模和预报程序.....	214
第十章 线性规划	224
§ 10.1 对偶单纯形法程序.....	224
§ 10.2 修正单纯形法程序.....	229
§ 10.3 整数线性规划程序.....	233
§ 10.4 线性规划的匈牙利解法程序.....	244
§ 10.5 多目标线性规划程序.....	249
第十一章 宽行打圈程序	282
§ 11.1 一元线性回归散点图程序.....	282
§ 11.2 打类各类曲线的程序.....	287
§ 11.3 统计分析直方图程序.....	293
§ 11.4 判别分析图程序.....	297

第六章 判 别 分 析

§ 6.1 二组判别分析程序

一、数学方法描述

设有 P 个判别变量, A 类有 n_1 个样本, B 类有 n_2 个样本, 还有 n_3 个样本为待判样本, 其数据形式如下:

	X_1	X_2		X_P
A	$X_{11}(A)$	$X_{21}(A)$		$X_{P1}(A)$
	$X_{12}(A)$	$X_{22}(A)$		$X_{P2}(A)$
	$\vdots$	$\vdots$		$\vdots$
	$X_{1n_1}(A)$	$X_{2n_1}(A)$		$X_{Pn_1}(A)$
B	$X_{11}(B)$	$X_{21}(B)$		$X_{P1}(B)$
	$X_{12}(B)$	$X_{22}(B)$		$X_{P2}(B)$
	$\vdots$	$\vdots$		$\vdots$
	$X_{1n_2}(B)$	$X_{2n_2}(B)$		$X_{Pn_2}(B)$
待 判	X_{11}	X_{21}		X_{P1}
	X_{12}	X_{22}		X_{P2}
	$\vdots$	$\vdots$		$\vdots$
	X_{1n_3}	X_{2n_3}		X_{Pn_3}

二组判别计算步骤如下:

1、求线性判别函数

$$y = \sum_{i=1}^P c_i x_i \quad (1)$$

其中 c_i ($i=1, 2, \dots, P$) 为判别系数。

要使判别函数 y 能反映预报量的类别, 必须根据以下两个原则:

- 1) 要使预报量为 A 类的平均值 $\bar{y}(A)$ 与 B 类的平均值 $\bar{y}(B)$ 有较大的区别, 即 $(\bar{y}(A) - \bar{y}(B))^2$ 数值要大。
- 2) 同类的数值要相对集中, 即所有 $y(A)$ 或 $y(B)$ 的数据要比较接近, 即

2、计算判别指标

$$Y_c = \frac{n_1 \bar{y}(A) + n_2 \bar{y}(B)}{n_1 + n_2}$$

3、计算检验统计量

$$F = \left(\frac{n_1 \cdot n_2}{(n_1 + n_2)(n_1 + n_2 - 2)} \right) \left(\frac{n_1 + n_2 - p - 1}{p} \right) D^2$$

$$\text{其中 } D^2 = (n_1 + n_2 - 2) \sum_{i=1}^p c_i d_i$$

F自由度为P, $n_1 + n_2 - p - 1$

4、计算判别样本判别值

$$y_k = \sum_{i=1}^p C_i X_{ik} \quad (k = 1, 2, \dots, n_3)$$

二、程序功能

本程序可以根据两组不同类型的P个判别变量的数据,计算出判别函数和判别指标,并能对若干个独立样本的属性进行判别分析。

三、程序使用说明

1、输入参数:

- LP 自变量个数
- LP₁ 其值为LP+1
- N₁ A类样本个数
- N₂ B类样本个数
- N₃ 待判样本个数
- A_i 存放A类样本数据的LP*N₁个元素的二维数组
- B_i 存放B类样本数据的LP*N₂个元素的二维数组
- Z 存放待判样本数据的LP*N₃个元素的二维数组

2、工作单元:

- XA 存放A类各个判别变量均值的一维数组
- XB 存放B类各个判别变量均值的一维数组
- S₁ 存放A类样本协方差矩阵元素的二维数组,最后放方程组的增广矩阵,LP₁列存放判别系数
- S₂ 存放B类样本协方差矩阵元素的二维数组

3、输出参数:

S_i 第 LP_i 列 存判别系数
 YA A类样本的综合指标
 YB B类样本的综合指标
 YC 判别指标
 F 检验统计量
 Y 存放待判样本的判别值的一维数组

四、源程序

```

    DIMENSION A1 (4, 5) , B1 (4, 5) , Z (4, 8) , S1 (4, 5) ,
* S2 (4, 5) , XA (4) , XB (4) , Y (8)
    READ (105, 1) LP, LP1, N1, N2, N3
1  FORMAT (5I5)
    DO 10 I=1, LP
        READ (105, 5) (A1 (I, J) , J=1, N1) , (B1 (I, J) , J=1,
* N2) , (Z (I, J) , J=1, N3)
5  FORMAT (16F5. 2)
10  CONTINUE
    WRITE (108, 11) ( (A1 (I, J) , I=1, LP) , J=1, N1)
11  FORMAT (5X, 'A1, ' / (8X, 4F10. 2) )
    WRITE (108, 12) ( (B1 (I, J) , I=1, LP) , J=1, N2)
12  FORMAT (5X, 'B1, ' / (8X, 4F10. 2) )
    WRITE (108, 13) ( (Z (I, J) , I=1, LP) , J=1, N3)
13  FORMAT (5X, 'Z, ' / (8X, 4F10. 2) )
    CALL DA (LP, LP1, N1, N2, N3, A1, B1, Z, S1, S2, XA, XB,
* YA, YB, YC, Y, F, 1. E-10)
    STOP
    END

    SUBROUTINE DA (LP, LP1, N1, N2, N3, A1, B1, Z, S1, S2,
* XA, XB, YA, YB, YC, Y, F, EPS)
    DIMENSION A1 (LP, N1) , B1 (LP, N2) , Z (LP, N3) , S1 (LP,
* LP1) , S2 (LP, LP1) , XA (LP) , XB (LP) , Y (N3)
    CALL CM (LP, LP1, N1, A1, XA, S1)
    CALL CM (LP, LP1, N2, B1, XB, S2)
    DO 50 I=1, LP
        S1 (I, LP1) =XA (I) -XB (I)
    DO 50 J=1, LP

```

```

50 S1 (I, J) =S1 (I, J) +S2 (I, J)
   LP2=LP-1
   DO 55 I=1, LP2
   I1=I+1
   DO 55 J=I1, LP
55 S1 (J, I) =S1 (I, J)
   CALL GS (LP, LP1, S1, EPS, KOD)
   WRITE (108, 61) KOD
61 FORMAT (5X, 'KOD=' , I5)
   WRITE (108, 62) (S1 (I, LP1) , I, I=1, LP)
62 FORMAT (5X, 'Y=' , 4 (F12. 8, 1HX, I1, 1H+ ) )
   YA=0.
   YB=0.
   DO 70 I=1, LP
   YA=YA+S1 (I, LP1) *XA (I)
70 YB=YB+S1 (I, LP1) *XB (I)
   P=FLOAT (LP)
   Q1=FLOAT (N1)
   Q2=FLOAT (N2)
   YC= (Q1*YA+Q2*YB) / (Q1+Q2)
   D=0
   DO 80 I=1, LP
80 D=D+S1 (I, LP1) * (XA (I) -XB (I) )
   F=Q1*Q2* (Q1+Q2-P-1) *D/ (P* (Q1+Q2) )
   WRITE (108, 60) YA, YB, YC, F
60 FORMAT (/5X, 'YA=' , F8. 5, 5X, 'YB=' , F8. 5, 5X,
   * 'YC=' , F8. 5, 5X, 'F=' , F8. 5, /)
   DO 100 J=1, N3
   Y (J) =0
   DO 90 I=1, LP
90 Y (J) =Y (J) +S1 (I, LP1) *Z (I, J)
   ID=2
   IF (Y (J) . GE. YC) ID=1
   WRITE (108, 98) J, Y (J) , ID
98 FORMAT (5X, 'Y ( ' , I3, ' ) =' , F12. 8, 5X, ' ( ' , I1, ' ) ' )
100 CONTINUE
   RETURN

```

```

END
SUBROUTINE CM (L, L1, N, A, X, S)
DIMENSION A (L, N) , X (L) , S (L, L1)
DO 20 I=1, L
X (I) = 0.
DO 15 J=1, N
15 X (I) =X (I) +A (I, J)
X (I) =X (I) /FLOAT (N)
DO 20 J=1, N
20 A (I, J) =A (I, J) -X (I)
DO 25 I=1, L
DO 25 J=1, L
S (I, J) = 0.
DO 25 K=1, N
25 S (I, J) =S (I, J) +A (I, K) *A (J, K)
RETURN
END
SUBROUTINE GS (N, N1, A, EP, KOD)
DIMENSION A (N, N1)
DO 10 K=1, N
DO 20 IO=K, N
IF (ABS (A (IO, K) ) -EP) 20, 20, 100
20 CONTINUE
KOD=1
RETURN
100 IF (IO, EQ. K) GO TO 200
DO 30 J=K, N1
T=A (K, J)
A (K, J) =A (IO, J)
A (IO, J) =T
30 CONTINUE
200 T=1/A (K, K)
DO 40 J=K, N
40 A (K, J+1) =T*A (K, J+1)
IN=N-1
IF (K, EQ. N) GO TO 400
DO 10 I=IO, IN

```

```

DO 10 J=K, N
10 A (I+1, J+1) = A (I+1, J+1) - A (I+1, K) * A (K, J+1)
400 DO 50 IK=2, N
    I=N1-1K
    DO 50 J=I, IN
50 A (I, N1) = A (I, N1) - A (I, J+1) * A (J+1, N1)
    KOD=0
    RETURN
END

```

五、例 题

LP=4, LP₁=5, N₁⁻=5, N₂⁻=5, N₃⁻=6

原始数据和计算结果如下:

A ₁	X ₁	X ₂	X ₃	X ₄
	13.85	2.79	7.80	49.60
	22.31	4.67	12.31	47.80
	28.82	4.63	16.18	62.15
	15.29	3.54	7.50	43.20
	28.79	4.90	16.12	58.10

B ₁				
	2.18	1.06	1.22	20.60
	3.85	.80	4.06	47.10
	11.40	.00	3.50	.00
	3.66	2.42	2.14	15.10
	12.10	.00	5.68	.00

Z ₁				
	8.85	3.38	5.17	26.10
	28.60	2.40	1.20	127.00
	20.70	6.70	7.60	30.80
	7.90	2.40	4.30	33.20
	3.19	3.20	1.43	9.90
	12.40	5.10	4.43	24.60

KOD=0

Y = .5929X₁ + .5240X₂ - 1.0736X₃ + .0907X₄

YA=6.94979 YB=2.32124 YC=4.63552 F=14.46419

Y (1) = 3.83408 (2)

$$\begin{aligned}
Y(2) &= 28.44205 & (1) \\
Y(3) &= 10.41646 & (1) \\
Y(4) &= 4.33527 & (2) \\
Y(5) &= 2.93044 & (2) \\
Y(6) &= 7.49847 & (1)
\end{aligned}$$

§ 6.2 多类训练迭代判别分析程序

一、数学方法描述

记 $X = (x_1, x_2, \dots, x_p)'$ 为 P 个指标的样本向量, 其中每个指标可以是离散的代码, 也可以是连续变量。

设从 g 个总体 ($g \geq 2$) 中分别抽取 n_i ($i = 1, 2, \dots, g$) 个样本, 令 $n = \sum_{i=1}^g n_i$ 。这 n 个

向量按随机的次序进行排列和编号。假设这些样本可以用线性判别函数完全无错判地分类, 这时我们建立 g 个线性判别函数

$$u_i(X) = X'W(i) \quad (1)$$

其中 $i = 1, 2, \dots, g$, w 为系数向量, $w = (w_1, w_2, \dots, w_p)'$ 。下标 (i) 表示第 i 个总体。

这时我们规定下列分类规则, 如果

$$u_i(X) = \max_{1 \leq i \leq g} u_i(X) \quad (2)$$

则判 $X \in H_a$ 。这里 H_a 表示第 a 个总体。

训练迭代不是一次就能求出系数向量的。具体算法是: 先给出任一初始向量 $W_{(i)}^{(0)}$, 逐个检查每一总体 H_i 中的样本 X 是否满足

$$X'W_{(i)}^{(0)} \geq X'W_{(j)}^{(0)} \quad (3)$$

这里上角 (0) 表示第 0 次迭代, 又 $j = 1, 2, \dots, g$, $j \neq i$ 。如 $j = 1$ 时, 有

$$X'W_{(i)}^{(0)} < X'W_{(1)}^{(0)}$$

则对 $W(i)$ 和 $W(l)$ 同时进行调整

$$W_{(i)}^{(1)} = W_{(i)}^{(0)} + CX \quad (4)$$

$$W_{(l)}^{(1)} = W_{(l)}^{(0)} - CX \quad (5)$$

C 是常数, 取法详见后面说明。这样不断地调整, 从第 t 次迭代到第 $t+1$ 次迭代, 均利用同样形式的公式

$$W_{(i)}^{(t+1)} = W_{(i)}^{(t)} + CX \quad (6)$$

$$W_{(l)}^{(t+1)} = W_{(l)}^{(t)} - CX \quad (7)$$

直到全部样本都满足 (3) 式为止。

这里需要说明, 以上算法的前提是假设可以用线性判别函数将全部样本无错判地分类, 这一前提不是任何情况下都存在的。如果不满足, 则迭代过程不收敛。

C 称为调整系数, 它的选取方法有以下两种:

(1) 取一固定的正数, 最方便是令 $C=1$ 。

(2) 取 C 为样本的函数, 我们记 X_g 为第 g 个样本向量, 又 $[\cdot]$ 表示对括号内的数取整数, 并取

$$C = \left\lceil \frac{W_{(i)}^{(t)} X_g}{X_g' X_g} \right\rceil \quad (8)$$

程序中采用第二种方法, 但限定当按上式获得的 $C < 1$ 时取 $C=1$ 。

由于训练迭代法的收敛性不是无条件的, 因而程序中规定了迭代次数的最高界限, 当迭代超过 200 次仍不收敛, 则宣布计算失败。

二、程序功能

本程序用训练迭代法, 使建立的判别函数对每个已预分类样本都不错判, 然后对于待判样本进行判别。本程序可用于两类及多类判别。

三、程序使用说明

1、输入参数:

- N 分类的样本数
- M 变量数
- M_1 预分类数
- N_1 待判样本数
- JK 最高迭代次数

X 已分类样本原始数据矩阵 (N行M列)

ICL 已分类样本的预分类号

X₁ 待判样本的数据矩阵 (N₁行M列)

2、输出参数:

除X、ICL、X₁外, 还有

KK 迭代次数, 如果KK>JK, 迭代失败, 结果作废。

W 判别系数矩阵 (M₁行M列)

V 已知样本判别得分矩阵 (N行M₁列)

R₁ 判别样本判别得分矩阵 (N₁行M₁列)

IJ 判别结果分类号。

四、源程序

```
DIMENSION X (24, 9) , ICL (24) , X1 (15, 9) , W (3, 9) ,  
*R (24) , V (24, 3) , R1 (15, 3)  
READ (105, 1) N, M, M1, N1, JK  
1 FORMAT (5I5)  
DO 5 I=1, N  
  READ (105, 2) (X (I, J) , J=1, M)  
2 FORMAT (24F3, 0)  
  WRITE (108, 3) I, (X (I, J) , J=1, M)  
3 FORMAT (2X, 'X' , I2, ': ' /5X, 24F5, 1)  
5 CONTINUE  
  READ (105, 6) (ICL (I) , I=1, N)  
6 FORMAT (40I2)  
  WRITE (108, 7) (ICL (I) , I=1, N)  
7 FORMAT (2X, 40I3)  
  IF (N1, EQ, 0) GO TO 10  
  DO 9 I=1, N1  
    READ (105, 2) (X1 (I, J) , J=1, M)  
    WRITE (108, 8) I, (X1 (I, J) , J=1, M)  
8 FORMAT (2X, 'N1=' , I2, ': ' /5X, 24F5, 1)  
9 CONTINUE  
10 CONTINUE  
  KK=0  
  DO 14 I=1, M1  
    DO 14 K=1, N  
      IF (ICL (K) , NE, I) GO TO 14
```

```

DO 13 J=1, M
13 W (I, J) =X (K, J)
14 CONTINUE
50 IK=0
   KK=KK+1
   IF (KK. GT. JK) GO TO 100
   DO 200 I=1, N
   R (I) =0.
   DO 16 J=1, M
16 R (I) =R (I) +W (ICL (I) , J) *X (I, J)
   DO 18 K=1, M1
   RJ=0.
   DO 20 J=1, M
20 RJ=RJ+W (K, J) *X (I, J)
   IF (R (I) . GE. RJ) GO TO 99
   C=CC (I, K, W, X, M)
   IK=IK+1
   DO 23 J=1, M
   W (ICL (I) , J) =W (ICL (I) , J) +C*X (I, J)
   W (K, J) =W (K, J) -C*X (I, J)
23 CONTINUE
99 CONTINUE
18 CONTINUE
200 CONTINUE
   IF (IK. NE. 0) GO TO 50
100 CONTINUE
   WRITE (108, 101) KK
101 FORMAT (10X, 'KK=' , I3)
   DO 30 I=1, M1
   WRITE (108, 32) I, (W (I, J) , J=1, M)
32 FORMAT (2X, I2, 2X, 10 (2X, F8. 2) )
30 CONTINUE
   DO 34 I=1, N
   DO 34 J=1, M1
   V (I, J) =0.
   DO 34 K=1, M
   V (I, J) =V (I, J) +X (I, K) *W (J, K)

```

```

34 CONTINUE
DO 35 I=1, N
P=V (I, 1)
IJ=1
DO 36 J=2, M1
IF (V (I, J) . LE. P) GO TO 36
P=V (I, J)
IJ=J
36 CONTINUE
WRITE (108, 52) I, (V (I, J) , J=1, M1) , IJ
52 FORMAT (2X, I2, 2X, 3F10. 2, 5X, I1)
35 CONTINUE
IF (N1. EQ. 0) GO TO 300
DO 40 I=1, N1
DO 40 K=1, M1
R1 (I, K) = 0.
DO 40 J=1, M
R1 (I, K) =R1 (I, K) + W (K, J) *X1 (I, J)
40 CONTINUE
DO 45 I=1, N1
Q=R1 (I, 1)
IJ=1
DO 42 J=2, M1
IF (R1 (I, J) . LE. Q) GO TO 42
Q=R1 (I, J)
IJ=J
42 CONTINUE
WRITE (108, 52) I, (R1 (I, J) , J=1, M1) , IJ
45 CONTINUE
300 CONTINUE
STOP
END
FUNCTION CC (I, K, W, X, M)
DIMENSION W (3, 9) , X (24, 9)
F=0.
G=0.
DO 10 J=1, M

```

```

      F=F+W (K, J) *X (I, J)
      G=G+X (I, J) **2
10  CONTINUE
      F=ABS (F)
      CC=AINTE (F/G)
      IF (CC. LT. 1. ) CC=1.
      RETURN
      END

```

五、例 题

$N=24$, $M=9$, $M_1=3$, $N_1=15$, $JK=200$

X 、 ICL 及 X_1 的原始数据如下:

X_1 :

1	3.0	1.0	3.0	6.0	3.0	1.0	3.0	3.0	3.0
2	6.0	5.0	7.0	5.0	6.0	4.0	5.0	5.0	3.0
3	5.0	3.0	5.0	5.0	5.0	2.0	3.0	5.0	3.0
4	3.0	2.0	4.0	6.0	4.0	3.0	7.0	5.0	3.0
5	2.0	2.0	3.0	6.0	4.0	2.0	6.0	5.0	3.0
6	3.0	2.0	4.0	6.0	5.0	2.0	5.0	5.0	3.0
7	3.0	2.0	3.0	5.0	5.0	2.0	3.0	5.0	3.0
8	3.0	2.0	4.0	5.0	4.0	1.0	3.0	5.0	2.0
9	4.0	2.0	5.0	6.0	3.0	1.0	4.0	5.0	2.0
10	5.0	1.0	6.0	7.0	2.0	1.0	4.0	3.0	2.0
11	2.0	1.0	3.0	6.0	3.0	1.0	4.0	3.0	1.0
12	2.0	2.0	2.0	5.0	6.0	1.0	4.0	3.0	1.0
13	6.0	3.0	6.0	5.0	4.0	2.0	3.0	3.0	2.0
14	2.0	.0	3.0	6.0	.0	.0	.0	2.0	1.0
15	1.0	.0	1.0	6.0	.0	.0	.0	2.0	1.0
16	4.0	3.0	5.0	5.0	5.0	2.0	4.0	3.0	.0
17	7.0	6.0	7.0	5.0	6.0	5.0	4.0	3.0	.0
18	4.0	4.0	4.0	5.0	7.0	3.0	4.0	3.0	.0
19	4.0	2.0	5.0	6.0	4.0	2.0	4.0	3.0	.0
20	5.0	3.0	6.0	5.0	5.0	3.0	4.0	3.0	.0
21	4.0	3.0	5.0	5.0	5.0	3.0	4.0	3.0	.0
22	3.0	1.0	3.0	6.0	3.0	1.0	3.0	3.0	.0
23	4.0	3.0	4.0	5.0	6.0	2.0	3.0	3.0	.0
24	3.0	2.0	3.0	5.0	5.0	2.0	4.0	3.0	.0