

NAG Library Routine Document

G02LBF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of *bold italicised* terms and other implementation-dependent details.

1 Purpose

G02LBF fits an orthogonal scores partial least squares (PLS) regression by using Wold's iterative method.

2 Specification

```

SUBROUTINE G02LBF (N, MX, X, LDX, ISX, IP, MY, Y, LDY, XBAR, YBAR,      &
                  ISCALE, XSTD, YSTD, MAXFAC, MAXIT, TAU, XRES, LDXRES,  &
                  YRES, LDYRES, W, LDW, P, LDP, T, LDT, C, LDC, U, LDU,  &
                  XCV, YCV, LDYCV, IFAIL)
INTEGER              N, MX, LDX, ISX(MX), IP, MY, LDY, ISCALE, MAXFAC,  &
                  MAXIT, LDXRES, LDYRES, LDW, LDP, LDT, LDC, LDU,      &
                  LDYCV, IFAIL
REAL (KIND=nag_wp) X(LDX,MX), Y(LDY,MY), XBAR(IP), YBAR(MY), XSTD(IP),  &
                  YSTD(MY), TAU, XRES(LDXRES,IP), YRES(LDYRES,MY),    &
                  W(LDW,MAXFAC), P(LDP,MAXFAC), T(LDT,MAXFAC),        &
                  C(LDC,MAXFAC), U(LDU,MAXFAC), XCV(MAXFAC),          &
                  YCV(LDYCV,MY)

```

3 Description

Let X_1 be the mean-centred n by m data matrix X of n observations on m predictor variables. Let Y_1 be the mean-centred n by r data matrix Y of n observations on r response variables.

The first of the k factors PLS methods extract from the data predicts both X_1 and Y_1 by regressing on a t_1 column vector of n scores:

$$\begin{aligned}\hat{X}_1 &= t_1 p_1^T \\ \hat{Y}_1 &= t_1 c_1^T, \quad \text{with } t_1^T t_1 = 1,\end{aligned}$$

where the column vectors of m x -loadings p_1 and r y -loadings c_1 are calculated in the least squares sense:

$$\begin{aligned}p_1^T &= t_1^T X_1 \\ c_1^T &= t_1^T Y_1.\end{aligned}$$

The x -score vector $t_1 = X_1 w_1$ is the linear combination of predictor data X_1 that has maximum covariance with the y -scores $u_1 = Y_1 c_1$, where the x -weights vector w_1 is the normalised first left singular vector of $X_1^T Y_1$.

The method extracts subsequent PLS factors by repeating the above process with the residual matrices:

$$\begin{aligned}X_i &= X_{i-1} - \hat{X}_{i-1} \\ Y_i &= Y_{i-1} - \hat{Y}_{i-1}, \quad i = 2, 3, \dots, k,\end{aligned}$$

and with orthogonal scores:

$$t_i^T t_j = 0, \quad j = 1, 2, \dots, i - 1.$$

Optionally, in addition to being mean-centred, the data matrices X_1 and Y_1 may be scaled by standard deviations of the variables. If data are supplied mean-centred, the calculations are not affected within numerical accuracy.

4 References

Wold H (1966) Estimation of principal components and related models by iterative least squares *In: Multivariate Analysis* (ed P R Krishnaiah) 391–420 Academic Press NY

5 Parameters

- 1: N – INTEGER *Input*
On entry: n , the number of observations.
Constraint: $N > 1$.
- 2: MX – INTEGER *Input*
On entry: the number of predictor variables.
Constraint: $MX > 1$.
- 3: X(LDX, MX) – REAL (KIND=nag_wp) array *Input*
On entry: $X(i, j)$ must contain the i th observation on the j th predictor variable, for $i = 1, 2, \dots, N$ and $j = 1, 2, \dots, MX$.
- 4: LDX – INTEGER *Input*
On entry: the first dimension of the array X as declared in the (sub)program from which G02LBF is called.
Constraint: $LDX \geq N$.
- 5: ISX(MX) – INTEGER array *Input*
On entry: indicates which predictor variables are to be included in the model.
 $ISX(j) = 1$
 The j th predictor variable (with variates in the j th column of X) is included in the model.
 $ISX(j) = 0$
 Otherwise.
Constraint: the sum of elements in ISX must equal IP.
- 6: IP – INTEGER *Input*
On entry: m , the number of predictor variables in the model.
Constraint: $1 < IP \leq MX$.
- 7: MY – INTEGER *Input*
On entry: r , the number of response variables.
Constraint: $MY \geq 1$.
- 8: Y(LDY, MY) – REAL (KIND=nag_wp) array *Input*
On entry: $Y(i, j)$ must contain the i th observation for the j th response variable, for $i = 1, 2, \dots, N$ and $j = 1, 2, \dots, MY$.
- 9: LDY – INTEGER *Input*
On entry: the first dimension of the array Y as declared in the (sub)program from which G02LBF is called.
Constraint: $LDY \geq N$.

- 10: XBAR(IP) – REAL (KIND=nag_wp) array Output
On exit: mean values of predictor variables in the model.
- 11: YBAR(MY) – REAL (KIND=nag_wp) array Output
On exit: the mean value of each response variable.
- 12: ISCALE – INTEGER Input
On entry: indicates how predictor variables are scaled.
 ISCALE = 1
 Data are scaled by the standard deviation of variables.
 ISCALE = 2
 Data are scaled by user-supplied scalings.
 ISCALE = -1
 No scaling.
Constraint: ISCALE = -1, 1 or 2.
- 13: XSTD(IP) – REAL (KIND=nag_wp) array Input/Output
On entry: if ISCALE = 2, XSTD(*j*) must contain the user-supplied scaling for the *j*th predictor variable in the model, for *j* = 1, 2, ..., IP. Otherwise XSTD need not be set.
On exit: if ISCALE = 1, standard deviations of predictor variables in the model. Otherwise XSTD is not changed.
- 14: YSTD(MY) – REAL (KIND=nag_wp) array Input/Output
On entry: if ISCALE = 2, YSTD(*j*) must contain the user-supplied scaling for the *j*th response variable in the model, for *j* = 1, 2, ..., MY. Otherwise YSTD need not be set.
On exit: if ISCALE = 1, the standard deviation of each response variable. Otherwise YSTD is not changed.
- 15: MAXFAC – INTEGER Input
On entry: *k*, the number of latent variables to calculate.
Constraint: $1 \leq \text{MAXFAC} \leq \text{IP}$.
- 16: MAXIT – INTEGER Input
On entry: if MY = 1, MAXIT is not referenced; otherwise the maximum number of iterations used to calculate the *x*-weights.
Suggested value: MAXIT = 200.
Constraint: if MY > 1, MAXIT > 1.
- 17: TAU – REAL (KIND=nag_wp) Input
On entry: if MY = 1, TAU is not referenced; otherwise the iterative procedure used to calculate the *x*-weights will halt if the Euclidean distance between two subsequent estimates is less than or equal to TAU.
Suggested value: TAU = 1.0E-4.
Constraint: if MY > 1, TAU > 0.0.
- 18: XRES(LDXRES, IP) – REAL (KIND=nag_wp) array Output
On exit: the predictor variables' residual matrix X_k .

- 19: LDXRES – INTEGER *Input*
On entry: the first dimension of the array XRES as declared in the (sub)program from which G02LBF is called.
Constraint: $LDXRES \geq N$.
- 20: YRES(LDYRES, MY) – REAL (KIND=nag_wp) array *Output*
On exit: the residuals for each response variable, Y_k .
- 21: LDYRES – INTEGER *Input*
On entry: the first dimension of the array YRES as declared in the (sub)program from which G02LBF is called.
Constraint: $LDYRES \geq N$.
- 22: W(LDW, MAXFAC) – REAL (KIND=nag_wp) array *Output*
On exit: the j th column of W contains the x -weights w_j , for $j = 1, 2, \dots, \text{MAXFAC}$.
- 23: LDW – INTEGER *Input*
On entry: the first dimension of the array W as declared in the (sub)program from which G02LBF is called.
Constraint: $LDW \geq \text{IP}$.
- 24: P(LDP, MAXFAC) – REAL (KIND=nag_wp) array *Output*
On exit: the j th column of P contains the x -loadings p_j , for $j = 1, 2, \dots, \text{MAXFAC}$.
- 25: LDP – INTEGER *Input*
On entry: the first dimension of the array P as declared in the (sub)program from which G02LBF is called.
Constraint: $LDP \geq \text{IP}$.
- 26: T(LDT, MAXFAC) – REAL (KIND=nag_wp) array *Output*
On exit: the j th column of T contains the x -scores t_j , for $j = 1, 2, \dots, \text{MAXFAC}$.
- 27: LDT – INTEGER *Input*
On entry: the first dimension of the array T as declared in the (sub)program from which G02LBF is called.
Constraint: $LDT \geq N$.
- 28: C(LDC, MAXFAC) – REAL (KIND=nag_wp) array *Output*
On exit: the j th column of C contains the y -loadings c_j , for $j = 1, 2, \dots, \text{MAXFAC}$.
- 29: LDC – INTEGER *Input*
On entry: the first dimension of the array C as declared in the (sub)program from which G02LBF is called.
Constraint: $LDC \geq \text{MY}$.
- 30: U(LDU, MAXFAC) – REAL (KIND=nag_wp) array *Output*
On exit: the j th column of U contains the y -scores u_j , for $j = 1, 2, \dots, \text{MAXFAC}$.

- 31: LDU – INTEGER *Input*
On entry: the first dimension of the array U as declared in the (sub)program from which G02LBF is called.
Constraint: $LDU \geq N$.
- 32: XCV(MAXFAC) – REAL (KIND=nag_wp) array *Output*
On exit: XCV(*j*) contains the cumulative percentage of variance in the predictor variables explained by the first *j* factors, for $j = 1, 2, \dots, \text{MAXFAC}$.
- 33: YCV(LDYCV, MY) – REAL (KIND=nag_wp) array *Output*
On exit: YCV(*i, j*) is the cumulative percentage of variance of the *j*th response variable explained by the first *i* factors, for $i = 1, 2, \dots, \text{MAXFAC}$ and $j = 1, 2, \dots, \text{MY}$.
- 34: LDYCV – INTEGER *Input*
On entry: the first dimension of the array YCV as declared in the (sub)program from which G02LBF is called.
Constraint: $LDYCV \geq \text{MAXFAC}$.
- 35: IFAIL – INTEGER *Input/Output*
On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.
 For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this parameter, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**
On exit: IFAIL = 0 unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL = 0 or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL = 1

On entry, element $\langle \text{value} \rangle$ of ISX is invalid.

On entry, ISCALE = $\langle \text{value} \rangle$.

Constraint: ISCALE = -1 or 1.

On entry, MX = $\langle \text{value} \rangle$.

Constraint: $MX > 1$.

On entry, MY = $\langle \text{value} \rangle$.

Constraint: $MY \geq 1$.

On entry, N = $\langle \text{value} \rangle$.

Constraint: $N > 1$.

IFAIL = 2

On entry, IP = $\langle \text{value} \rangle$ and MX = $\langle \text{value} \rangle$.

Constraint: $1 < IP \leq \text{MX}$.

On entry, $LDC = \langle value \rangle$ and $MY = \langle value \rangle$.
Constraint: $LDC \geq MY$.

On entry, $LDP = \langle value \rangle$ and $IP = \langle value \rangle$.
Constraint: $LDP \geq IP$.

On entry, $LDT = \langle value \rangle$ and $N = \langle value \rangle$.
Constraint: $LDT \geq N$.

On entry, $LDU = \langle value \rangle$ and $N = \langle value \rangle$.
Constraint: $LDU \geq N$.

On entry, $LDW = \langle value \rangle$ and $IP = \langle value \rangle$.
Constraint: $LDW \geq IP$.

On entry, $LDX = \langle value \rangle$ and $N = \langle value \rangle$.
Constraint: $LDX \geq N$.

On entry, $LDXRES = \langle value \rangle$ and $N = \langle value \rangle$.
Constraint: $LDXRES \geq N$.

On entry, $LDY = \langle value \rangle$ and $N = \langle value \rangle$.
Constraint: $LDY \geq N$.

On entry, $LDYCV = \langle value \rangle$ and $MAXFAC = \langle value \rangle$.
Constraint: $LDYCV \geq MAXFAC$.

On entry, $LDYRES = \langle value \rangle$ and $N = \langle value \rangle$.
Constraint: $LDYRES < N$.

On entry, $MAXFAC = \langle value \rangle$ and $IP = \langle value \rangle$.
Constraint: $1 \leq MAXFAC \leq IP$.

On entry, $MY = \langle value \rangle$ and $MAXIT = \langle value \rangle$.
Constraint: if $MY > 1$, $MAXIT > 1$.

On entry, $TAU = \langle value \rangle$.
Constraint: if $MY > 1$, $TAU > 0.0$.

IFAIL = 3

On entry, IP is not equal to the sum of ISX elements: $IP = \langle value \rangle$, $\text{sum}(\text{ISX}) = \langle value \rangle$.

IFAIL = -99

An unexpected error has been triggered by this routine. Please contact NAG.
See Section 3.8 in the Essential Introduction for further information.

IFAIL = -399

Your licence key may have expired or may not have been installed correctly.
See Section 3.7 in the Essential Introduction for further information.

IFAIL = -999

Dynamic memory allocation failed.
See Section 3.6 in the Essential Introduction for further information.

7 Accuracy

In general, the iterative method used in the calculations is less accurate (but faster) than the singular value decomposition approach adopted by G02LAF.

8 Parallelism and Performance

G02LBF is not threaded by NAG in any implementation.

G02LBF makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

G02LBF allocates internally $(n + r)$ elements of real storage.

10 Example

This example reads in data from an experiment to measure the biological activity in a chemical compound, and a PLS model is estimated.

10.1 Program Text

```

Program g02lbfe

!      G02LBF Example Program Text

!      Mark 25 Release. NAG Copyright 2014.

!      .. Use Statements ..
      Use nag_library, Only: g02lbf, nag_wp, x04caf
!      .. Implicit None Statement ..
      Implicit None
!      .. Parameters ..
      Integer, Parameter          :: nin = 5, nout = 6
!      .. Local Scalars ..
      Real (Kind=nag_wp)         :: tau
      Integer                     :: i, ifail, ip, iscale, ldc, ldp, ldt, &
                                   ldu, ldw, ldx, ldxres, ldy, ldycv, &
                                   ldyres, maxfac, maxit, mx, my, n
      Character (80)              :: fmt
!      .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable :: c(:,,:), p(:,,:), t(:,,:), u(:,,:), &
                                   w(:,,:), x(:,,:), xbar(:), xcv(:), &
                                   xres(:,,:), xstd(:), y(:,,:), ybar(:), &
                                   ycv(:,,:), yres(:,,:), ystd(:)
      Integer, Allocatable          :: isx(:)
!      .. Intrinsic Procedures ..
      Intrinsic                    :: count
!      .. Executable Statements ..
      Write (nout,*) 'G02LBF Example Program Results'
      Write (nout,*)
      Flush (nout)

!      Skip heading in data file
      Read (nin,*)

!      Read in the problem size
      Read (nin,*) n, mx, my, iscale, maxfac

      ldx = n
      ldy = n
      Allocate (x(ldx,mx),isx(mx),y(ldy,my))

!      Read in data
      Read (nin,*)(x(i,1:mx),y(i,1:my),i=1,n)

```

```

!      Read in variable inclusion flags
      Read (nin,*) isx(1:mx)

!      Calculate IP
      ip = count(isx(1:mx)==1)

      ldxres = n
      ldyres = n
      ldt = n
      ldc = my
      ldu = n
      ldycv = maxfac
      ldw = ip
      ldp = ip
      Allocate (xbar(ip),ybar(my),xstd(ip),ystd(my),xres(ldxres,ip), &
        yres(ldyres,ip),w(ldw,maxfac),p(ldp,maxfac),t(ldt,maxfac), &
        c(ldc,maxfac),u(ldu,maxfac),xcv(maxfac),ycv(ldycv,my))

!      Use suggested values for control parameters
      maxit = 200
      tau = 1.0E-4_nag_wp

!      Fit a PLS model
      ifail = 0
      Call g02lbf(n,mx,x,ldx,isx,ip,my,y,ldy,xbar,ybar,yscale,xstd,ystd, &
        maxfac,maxit,tau,xres,ldxres,yres,ldyres,w,ldw,p,ldp,t,ldt,c,ldc,u, &
        ldu,xcv,ycv,ldycv,ifail)

!      Display results
      ifail = 0
      Call x04caf('General',' ',ip,maxfac,p,ldp,'x-loadings, P',ifail)
      Write (nout,*)
      Flush (nout)
      ifail = 0
      Call x04caf('General',' ',n,maxfac,t,ldt,'x-scores, T',ifail)
      Write (nout,*)
      Flush (nout)
      ifail = 0
      Call x04caf('General',' ',my,maxfac,c,ldc,'y-loadings, C',ifail)
      Write (nout,*)
      Flush (nout)
      ifail = 0
      Call x04caf('General',' ',n,maxfac,u,ldu,'y-scores, U',ifail)
      Write (nout,*)
      Write (nout,*) 'Explained Variance'
      Write (nout,*) ' Model effects   Dependent variable(s)'
      Write (fmt,99999) '( ', my + 1, '(F12.6,3X))'
      Write (nout,fmt)(xcv(i),ycv(i,1:my),i=1,maxfac)

99999 Format (A,I0,A)
      End Program g02lbfe

```

10.2 Program Data

G02LBF Example Program Data

15	15	1	1	4						
										: N, MX, MY, SCALE, MAXFAC
-2.6931	-2.5271	-1.2871	3.0777	0.3891						
-0.0701	1.9607	-1.6324	0.5746	1.9607						
-1.6324	0.5740	2.8369	1.4092	-3.1398	0.00					
-2.6931	-2.5271	-1.2871	3.0777	0.3891						
-0.0701	1.9607	-1.6324	0.5746	0.0744						
-1.7333	0.0902	2.8369	1.4092	-3.1398	0.28					
-2.6931	-2.5271	-1.2871	3.0777	0.3891						
-0.0701	0.0744	-1.7333	0.0902	1.9607						
-1.6324	0.5746	2.8369	1.4092	-3.1398	0.20					
-2.6931	-2.5271	-1.2871	3.0777	0.3891						
-0.0701	0.0744	-1.7333	0.0902	0.0744						
-1.7333	0.0902	2.8369	1.4092	-3.1398	0.51					
-2.6931	-2.5271	-1.2871	2.8369	1.4092						
-3.1398	0.0744	-1.7333	0.0902	0.0744						


```

-1.7333  0.0902  2.8369  1.4092 -3.1398  0.11
-2.6931 -2.5271 -1.2871  3.0777  0.3891
-0.0701 -4.7548  3.6521  0.8524  0.0744
-1.7333  0.0902  2.8369  1.4092 -3.1398  2.73
-2.6931 -2.5271 -1.2871  3.0777  0.3891
-0.0701  0.0744 -1.7333  0.0902  0.0744
-1.7333  0.0902 -1.2201  0.8829  2.2253  0.18
-2.6931 -2.5271 -1.2871  3.0777  0.3891
-0.0701  2.4064  1.7438  1.1057  0.0744
-1.7333  0.0902  2.8369  1.4092 -3.1398  1.53
-2.6931 -2.5271 -1.2871  0.0744 -1.7333
 0.0902  0.0744 -1.7333  0.0902  0.0744
-1.7333  0.0902  2.8369  1.4092 -3.1398 -0.10
 2.2261 -5.3648  0.3049  3.0777  0.3891
-0.0701  0.0744 -1.7333  0.0902  0.0744
-1.7333  0.0902  2.8369  1.4092 -3.1398 -0.52
-4.1921 -1.0285 -0.9801  3.0777  0.3891
-0.0701  0.0744 -1.7333  0.0902  0.0744
-1.7333  0.0902  2.8369  1.4092 -3.1398  0.40
-4.9217  1.2977  0.4473  3.0777  0.3891
-0.0701  0.0744 -1.7333  0.0902  0.0744
-1.7333  0.0902  2.8369  1.4092 -3.1398  0.30
-2.6931 -2.5271 -1.2871  3.0777  0.3891
-0.0701  2.2261 -5.3648  0.3049  2.2261
-5.3648  0.3049  2.8369  1.4092 -3.1398 -1.00
-2.6931 -2.5271 -1.2871  3.0777  0.3891
-0.0701 -4.9217  1.2977  0.4473  0.0744
-1.7333  0.0902  2.8369  1.4092 -3.1398  1.57
-2.6931 -2.5271 -1.2871  3.0777  0.3891
-0.0701 -4.1921 -1.0285 -0.9801  0.0744
-1.7333  0.0902  2.8369  1.4092 -3.1398  0.59 : End of observations
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 : ISX

```

10.3 Program Results

G02LBF Example Program Results

x-loadings, P

	1	2	3	4
1	-0.6708	-1.0047	0.6505	0.6169
2	0.4943	0.1355	-0.9010	-0.2388
3	-0.4167	-1.9983	-0.5538	0.8474
4	0.3930	1.2441	-0.6967	-0.4336
5	0.3267	0.5838	-1.4088	-0.6323
6	0.0145	0.9607	1.6594	0.5361
7	-2.4471	0.3532	-1.1321	-1.3554
8	3.5198	0.6005	0.2191	0.0380
9	1.0973	2.0635	-0.4074	-0.3522
10	-2.4466	2.5640	-0.4806	0.3819
11	2.2732	-1.3110	-0.7686	-1.8959
12	-1.7987	2.4088	-0.9475	-0.4727
13	0.3629	0.2241	-2.6332	2.3739
14	0.3629	0.2241	-2.6332	2.3739
15	-0.3629	-0.2241	2.6332	-2.3739

x-scores, T

	1	2	3	4
1	-0.1896	0.3898	-0.2502	-0.2479
2	0.0201	-0.0013	-0.1726	-0.2042
3	-0.1889	0.3141	-0.1727	-0.1350
4	0.0210	-0.0773	-0.0950	-0.0912
5	-0.0090	-0.2649	-0.4195	-0.1327
6	0.5479	0.2843	0.1914	0.2727
7	-0.0937	-0.0579	0.6799	-0.6129
8	0.2500	0.2033	-0.1046	-0.1014
9	-0.1005	-0.2992	0.2131	0.1223
10	-0.1810	-0.4427	0.0559	0.2114
11	0.0497	-0.0762	-0.1526	-0.0771
12	0.0173	-0.2517	-0.2104	0.1044
13	-0.6002	0.3596	0.1876	0.4812

14	0.3796	0.1338	0.1410	0.1999
15	0.0773	-0.2139	0.1085	0.2106

y-loadings, C

	1	2	3	4
1	3.5425	1.0475	0.2548	0.1866

y-scores, U

	1	2	3	4
1	-1.7670	0.1812	-0.0600	-0.0320
2	-0.6724	-0.2735	-0.0662	-0.0402
3	-0.9852	0.4097	0.0158	0.0198
4	0.2267	-0.0107	0.0180	0.0177
5	-1.3370	-0.3619	-0.0173	0.0073
6	8.9056	0.6000	0.0701	0.0422
7	-1.0634	0.0332	0.0235	-0.0151
8	4.2143	0.3184	0.0232	0.0219
9	-2.1580	-0.2652	0.0153	0.0011
10	-3.7999	-0.4520	0.0082	0.0034
11	-0.2033	-0.2446	-0.0392	-0.0214
12	-0.5942	-0.2398	0.0089	0.0165
13	-5.6764	0.5487	0.0375	0.0185
14	4.3707	-0.1161	-0.0639	-0.0535
15	0.5395	-0.1274	0.0261	0.0139

Explained Variance

Model effects	Dependent variable(s)
16.902124	89.638060
29.674338	97.476270
44.332404	97.939839
56.172041	98.188474
