5-1-2007

JMASM27: An Algorithm for Implementing Gibbs Sampling for 2PNO IRT Models (Fortran)

Yanyan Sheng
Southern Illinois University at Carbondale, ysheng@siu.edu

Todd C. Headrick
Southern Illinois University at Carbondale, headrick@siu.edu

JMASM27: An Algorithm for Implementing Gibbs Sampling for 2PNO IRT Models (Fortran)

Yanyan Sheng   Todd C. Headrick
Southern Illinois University-Carbondale

A Fortran 77 subroutine is provided for implementing the Gibbs sampling procedure to a normal ogive IRT model for binary item response data with the choice of uniform and normal prior distributions for item parameters. The subroutine requires the user to have access to the IMSL library. The source code is available at http://www.siu.edu/~epse1/sheng/Fortran/, along with a stand alone executable file.

Key words: IRT, two-parameter normal ogive model, MCMC, Gibbs sampling, Fortran.

Introduction

Item response theory (IRT) describes a probabilistic relationship between correct responses on a set of test items and a latent variable, where the influence of items and persons on the responses is modeled by distinct sets of parameters. Common IRT models include the two-parameter normal ogive (2PNO; Lawley, 1943, 1944; Lord, 1952, 1953a, 1953b) model such that the probability of person i obtaining a correct response for item j, where $i = 1, ..., n$ and $j = 1, ..., k$, is defined as

$$P(y_{ij} = 1) = \Phi(\alpha_j \theta_i - \gamma_j)$$

$$= \int_{-\infty}^{\alpha_j \theta_i - \gamma_j} \frac{1}{\sqrt{2\pi}} e^{\frac{-t^2}{2}} dt, \quad (1)$$

Yanyan Sheng is an Assistant Professor of Measurement and Statistics. Her areas of research interest are psychometrics, IRT and Bayesian hierarchical models, and adaptive testing. Email her at ysheng@siu.edu. Todd C. Headrick is Associate Professor of Statistics. Address: Section on Statistics and Measurement, Department of EPSE, 222-J Wham Building, Mail Code 4618, Southern Illinois University-Carbondale, IL, 62901. His areas of research interest are statistical computing, nonparametric statistics, and optimization. Email him at headrick@siu.edu.

where γ_j and α_j denote item parameters and θ_i denotes the continuous person trait. In the model, items are assumed to vary in terms of location, γ_j, as well as slope, α_j. Simultaneous estimation of both item and person parameters results in statistical complexities in the estimation task of IRT models, which have made estimation procedures a primary focus of psychometric research over decades (e.g., Birnbaum, 1969; Bock & Aitkin, 1981; Molenaar, 1995). Recent attention has been focused on Markov chain Monte Carlo (MCMC; e.g., Chib & Greenberg, 1995) techniques, which have demonstrated to be useful for complex estimation problems in many areas of applied statistics. Albert (1992) was the first to apply an MCMC algorithm, known as Gibbs sampling (Casella & George, 1992; Gelfand & Smith, 1990; Geman & Geman, 1984), to the 2PNO model, where he adopted non-informative priors for item parameters.

As Albert's (1992) focus was on investigating the applicability of Gibbs sampling to IRT, he did not specifically consider the situations where informative priors are adopted for item parameters. However, in some applications, they are more preferred than vague priors. For example, when comparing several candidate models, Bayes factors are commonly adopted in the Bayesian framework, but they are not defined with non-informative priors (Gelman et al., 2003). In this case, the program given by Albert (1992) does not provide a solution. Moreover, given that MCMC is computationally

demanding in drawing a sufficiently long chain to ensure convergence, a major problem in applied IRT is the accessibility of efficient MCMC programs. Researchers have either used WinBUGS (e.g., Bazán, Branco & Bolfarinez, 2006; DeMars, 2005) to implement MCMC for IRT models, or coded the sampler in S-Plus (e.g., Patz & Junker, 1999) or MATLAB (e.g., Albert, 1992). They noted that each execution consumed many hours, and hence was computationally expensive. This fact makes it impractical for users to utilize these programs for various applications of IRT. They further limit researchers in conducting Monte Carlo studies, or developing more complicated IRT models. It is then anticipated that Fortran will provide a better solution, as it is the fastest programming language for numerical computing (Brainerd, 2003).

In view of the above, the purpose of this article is to provide a Fortran subroutine that obtains the posterior estimates (and their associated standard errors) of item and person parameters in the 2PNO IRT model. The subroutine will have the option of specifying non-informative and informative priors for item parameters.

Methodology

The Gibbs Sampling Procedure

To implement Gibbs sampling to the 2PNO model defined in (1), a latent continuous random variable Z is introduced so that $Z_{ij} \sim N(\alpha_j \theta_i - \gamma_j, 1)$ (Albert, 1992; Tanner & Wong, 1987). With prior distributions assumed for θ_i and ξ_j, where $\xi_j = (\alpha_j, \gamma_j)'$, the joint posterior distribution of (θ, ξ) is hence

$$p(\theta, \xi \mid y) \propto f(y \mid Z) p(Z \mid \theta, \xi) p(\theta) p(\xi), \quad (2)$$

where $f(y \mid Z)$ is the likelihood function.

With a normal prior for θ_i and non-informative priors for α_j and γ_j so that $\theta_i \sim N(\mu, \sigma^2)$, $\alpha_j > 0$ and $p(\gamma_j) \propto 1$, the full conditional distributions of Z_{ij}, θ_i, and ξ_j can be derived in closed forms as follows:

$$Z_{ij} \mid \bullet \sim \begin{cases} N_{(0,\infty)}(\alpha_j \theta_i - \gamma_j, 1), & if \ y_{ij} = 1 \\ N_{(-\infty,0)}(\alpha_j \theta_i - \gamma_j, 1), & if \ y_{ij} = 0 \end{cases} \quad (3)$$

$$\theta_i \mid \bullet \sim N\left(\frac{\sum_j (Z_{ij} + \gamma_j)\alpha_j + \mu}{\frac{1}{\sigma^2} + \sum_j \alpha_j^2}, \frac{1}{\frac{1}{\sigma^2} + \sum_j \alpha_j^2} \right) \quad (4)$$

$$\xi_j \mid \bullet \sim N((\mathbf{x'x})^{-1}\mathbf{x'Z}_j, (\mathbf{x'x})^{-1})I(\alpha_j > 0), \quad (5)$$

where $\mathbf{x} = [\theta, -1]$. Alternatively, informative conjugate priors can be assumed for α_j and γ_j so that $\alpha_j \sim N_{(0,\infty)}(\mu_\alpha, \sigma_\alpha^2)$, $\gamma_j \sim N(\mu_\gamma, \sigma_\gamma^2)$. In this case, the full conditional distribution of ξ_j is derived as

$$\xi_j \mid \bullet \sim N((\mathbf{x'x} + \Sigma_\xi^{-1})^{-1}(\mathbf{x'Z}_j + \Sigma_\xi^{-1}\mu_\xi), \\ (\mathbf{x'x} + \Sigma_\xi^{-1})^{-1})I(\alpha_j > 0) \quad (6)$$

where $\mu_\xi = (\mu_\alpha, \mu_\gamma)'$ and $\Sigma_\xi = \begin{pmatrix} \sigma_\alpha^2 & 0 \\ 0 & \sigma_\gamma^2 \end{pmatrix}$. Hence, with starting values $\theta^{(0)}$ and $\xi^{(0)}$, observations $(Z^{(l)}, \theta^{(l)}, \xi^{(l)})$ can be simulated from the Gibbs sampler by iteratively drawing from their respective full conditional distributions specified in (3), (4) and (5) (or equations 3, 4, and 6). To go from $(Z^{(l-1)}, \theta^{(l-1)}, \xi^{(l-1)})$ to $(Z^{(l)}, \theta^{(l)}, \xi^{(l)})$, it takes three transition steps:

1. Draw $Z^{(l)} \sim p(Z \mid y, \theta^{(l-1)}, \xi^{(l-1)})$;
2. Draw $\theta^{(l)} \sim p(\theta \mid Z^{(l)}, \xi^{(l-1)})$;
3. Draw $\xi^{(l)} \sim p(\xi \mid Z^{(l)}, \theta^{(l)})$.

This iterative procedure produces a sequence of $(\theta^{(l)}, \xi^{(l)})$, $l = 0, …, L$. To reduce the effect of the starting values, early iterations in the Markov chain are set as burn-ins to be discarded. Samples from the remaining iterations are then used to summarize the posterior density of item

parameters ξ and ability parameters θ. As with standard Monte Carlo, with large enough samples, the posterior means of ξ and θ are considered as estimates of the true parameters. However, their standard deviations tend to underestimate the posterior standard deviations, as subsequent samples in Gibbs sampler are autocorrelated (e.g., Albert, 1992; Patz & Junker, 1999). One approach to calculating them is through batching (Ripley, 1987). That is, with a long chain of samples being separated into contiguous batches of equal length, the posterior mean and standard deviation for each parameter are then estimated to be the sample mean and standard deviation of these batch means respectively. Thus, the standard error of the estimate is a ratio of the standard deviation and the square root of the number of batches.

The Fortran Subroutine

The subroutine initially sets the starting values for the parameters, so that $\theta_i^{(0)} = 0$, $\alpha_i^{(0)} = 2$ and $\gamma_i^{(0)} = -\Phi^{-1}(\sum_i y_{ij}/n)\sqrt{5}$ (Albert, 1992). It then iteratively draws random samples for Z and θ from their respective full conditional distributions specified in (3) and (4) with $\mu = 0$ and $\sigma^2 = 1$. Samples for ξ_j are simulated either from (5), where uniform priors are assumed for ξ_j, or from (6), where normal priors are adopted with $\mu_\alpha = \mu_\gamma = 0$ and $\sigma_\alpha^2 = \sigma_\gamma^2 = 1$. The algorithm continues until all the L samples are simulated. It then discards the early burn-in samples, and computes the posterior estimates and standard errors for the model parameters, θ, α and γ, using batching.

For example, for a 2000-by-10 (i.e., $n =$ 2,000 and $k = 10$) dichotomous (0-1) data matrix simulated using the item parameters shown in the first two columns of Table 1, the Gibbs

sampler was implemented so that 10,000 samples were simulated with the first 5,000 taken to be burn-in. The remaining 5,000 samples were separated into 5 batches, each with 1,000 samples. Two sets of the posterior means for α and γ, as well as their standard errors, were obtained assuming the uniform or normal prior distributions described previously, and are displayed in the rest of the table. It is noted that the item parameters were estimated with enough accuracy and the two sets of posterior estimates differ only slightly from each other, signifying that the results are not sensitive to the choice of priors for ξ. For this example, each implementation took less than 13 minutes. Although 10,000 iterations are long enough for the Markov chain to reach the stationary distribution, one may easily increase the length of the chain to be as long as 50,000, which takes about 60-90 minutes for each execution.

Conclusion

This Fortran subroutine leaves it to the user to choose between uniform and normal priors for the item parameters, α and γ. In addition, the user can change the source code so that the prior distribution for θ_i assumes different location, μ and scale, σ^2. Similarly, μ_α, σ_α^2, and μ_γ, σ_γ^2 can be modified to reflect different prior beliefs on the distributions for the item parameters. It is noted that convergence can be assessed by comparing the marginal posterior mean and standard deviation of each parameter computed for every 1,000 samples after the burn-ins. Similar values provide a rough indication of similar marginal posterior densities, which further indicates possible convergence of the Gibbs sampler (Gelfand, Hills, Racine-Poon & Smith, 1990; Hoijtink & Molenaar, 1997).

Table 1. Posterior estimates and their standard errors for α and γ with uniform priors and normal priors.

Parameters		Posterior estimates			
		uniform priors		normal priors	
α	γ	$\hat{\alpha}$	$\hat{\gamma}$	$\hat{\alpha}$	$\hat{\gamma}$
		(SE)	(SE)	(SE)	(SE)
0.0966	-0.7997	0.1147	-0.8258	0.1120	-0.8223
		(.0009)	(.0010)	(.0012)	(.0006)
0.0971	-0.5321	0.1291	-0.5441	0.1285	-0.5449
		(.0003)	(.0006)	(.0022)	(.0004)
0.4589	0.8583	0.4412	0.9289	0.4416	0.9278
		(.0031)	(.0021)	(.0031)	(.0017)
0.9532	0.7237	1.1335	0.8906	1.1124	0.8801
		(.0088)	(.0032)	(.0074)	(.0041)
0.0771	-0.8184	0.0517	-0.8063	0.0510	-0.8056
		(.0005)	(.0005)	(.0007)	(.0007)
0.4891	-0.5834	0.4761	-0.6186	0.4726	-0.6188
		(.0023)	(.0006)	(.0023)	(.0004)
0.8599	0.3629	0.7960	0.3484	0.8015	0.3501
		(.0028)	(.0013)	(.0036)	(.0022)
0.9427	-0.9010	0.9230	-0.8661	0.9189	-0.8618
		(.0060)	(.0017)	(.0083)	(.0049)
0.2727	-0.9339	0.3981	-0.9497	0.3948	-0.9465
		(.0027)	(.0018)	(.0028)	(.0023)
0.6532	-0.3978	0.6562	-0.3785	0.6544	-0.3781
		(.0016)	(.0008)	(.0015)	(.0009)

References

Albert, J. H. (1992). Bayesian estimation of normal ogive item response curves using Gibbs sampling. *Journal of Educational Statistics, 17*, 251-269.

Bazán, J. L., Branco, M. D, & Bolfarinez, H. (2006). A Skew Item Response Model. *Bayesian Analysis, 1*, 861-892.

Birnbaum, A. (1969).Statistical theory for logistic mental test models with prior distribution of ability. *Journal of Mathematical Psychology, 6,* 258-276.

Bock, R. D., & Aitkin, M. (1981). Marginal maximum likelihood estimation of item parameters: application of an EM algorithm. *Psychometrika, 46*, 443-459.

Brainerd, W. (2003). The importance of Fortran in the 21st century. *Journal of Modern Statistical Methods, 2,* 14-15.

Casella, G., & George, E. I. (1992). Explaining the Gibbs sampler. *The American Statistician, 46*(3), 167-174.

Chib, S., & Greenberg, E. (1995). Understanding the Metropolis-Hastings algorithm. *The American Statistician, 49*(4), 327-335.

DeMars, C. E. (2005). *"Guessing" parameter estimates for multidimensional IRT models*. Paper presented at the annual meeting of the American Educational Research Association, Montreal.

Gelfand, A. E., & Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association, 85*, 398-409.

Gelfand, A. E., Hills, S. E., Racine-Poon, A., & Smith, A. F. M. (1990). Illustration of Bayesian inference in normal data models using Gibbs sampling. *Journal of the American Statistical Association, 85,* 315-331.

Geman, S. and D. Geman (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Analysis and Machine Intelligence* 6: 721-741.

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2003). *Bayesian data analysis*. Boca Raton: Chapman & Hall/CRC.

Hoijtink, H., & Molenaar, I. W. (1997). A multidimensional item response model: Constrained latent class analysis using posterior predictive checks. *Psychometrika*, *62*, 171-189.

Lawley, D. N. (1943). On problems connected with item selection and test construction. *Proceedings of the Royal Society of Edinburgh, 61*, 273-287.

Lawley, D. N. (1944). The factorial analysis of multiple item tests. *Proceedings of the Royal Society of Edinburgh, 62*, 74-82.

Lord, F.M. (1952). A theory of test scores. *Psychometric Monograph No. 7*.

Lord, F.M. (1953a). An application of confidence intervals and of maximum likelihood to the estimation of an examinee's ability. *Psychometrika, 18*, 57–75.

Lord, F.M. (1953b). The relation of test score to the trait underlying the test. *Educational and Psychological Measurement, 13*, 517–548.

Molenaar, I. W. (1995). Estimation of item parameters. In G. H. Fischer & I. W. Molenaar (Eds.), *Rasch models: Foundations, recent developments, and applications* (pp. 39–51). NewYork: Springer-Verlag.

Patz, R. J., & Junker, B. W. (1999). A straightforward approach to Markov chain Monte Carlo methods for item response models. *Journal of Educational and Behavioral Statistics, 24,* 146-178.

Ripley, B. D. (1987). *Stochastic simulation*. New York: Wiley.

Tanner, M. A., and W. H. Wong (1987). The calculation of posterior distribution by data augmentation (with discussion). *Journal of the American Statistical Association, 82*, 528-550.

Appendix

```
SUBROUTINE GSU2(Y, N, K, L, BURNIN, BN, UNIF, ITEM, PERSON)
C************************************************************************
C Y is the n-by-k binary item response data
C N is the number of subjects
C K is the test length (number of items)
C L is the number of iterations using Gibbs sampling
C BURNIN is the first number of iterations that are to be discarded
C BN is the number of batches
C UNIF is a 0-1 indicator with 0 specifying normal priors for item
C     parameters and 1 specifying uniform priors for them
C ITEM is a k-by-4 matrix of posterior estimates and standard errors
C     for item parameters
C PERSON is a n-by-2 matrix of posterior estimates and standard errors
C     for person abilities
C************************************************************************
```

```
      INTEGER    L, COUNT, IRANK, BURNIN, UNIF, INDX(2), BN,
     &      BSIZE, Y(N, K)
      REAL  A(K), G(K), TH(N), LP, MU, VAR, AV(L,K), GV(L,K), THV(N,L),
     &      PHAT(K), U, Z(N,K), V, MN, MSUM, PVAR,PMEAN, TT, X(N,2),
     &      XX(2,2), IX(2,2),ZV(N,1),XZ(2,1), AMAT(2,2), BZ(2,1), AMU,
     &      GMU, AVAR, GVAR, AGMU(2,1), AGVAR(2,2), SIGMA(2,2), BETA(1,2),
     &      BI(1,2), ITEM(K,4), PERSON(N,2), SUM1, SUM2, SUM3,
     &      M1, M2, M3, TOT1, TOT2, TOT3, SS1, SS2, SS3
      DOUBLE PRECISION    BB, TMP
C*****************************************************************************
C Connect to external libraries for normal (RNNOR) and uniform (RNUN)
C random number generator, inverse (ANORIN, DNORIN) and CDF (ANORDF,
C DNORDF) for the standard normal distribution, and Cholesky
C factorization (CHFAC) routines
C*****************************************************************************
   EXTERNAL  RNNOR, RNSET, RNUN, ANORDF, ANORIN,CHFAC, DNORDF, DNORIN
C*****************************************************************************
C Set initial values for item parameters a, g, and person ability theta so
C that a = 2, g = $-\Phi^{-1}(\sum_i y_{ij}/n)\sqrt{5}$ for all k items, and theta = 0 for all n
C persons.
C*****************************************************************************
      PHAT = SUM(Y, 1)
      DO 10 I = 1, K
        A(I) = 2.0
        G(I) = -ANORIN(PHAT(I)/N)*SQRT(5.0)
   10 CONTINUE
      DO 20 I = 1, N
        TH(I) = 0.0
   20 CONTINUE
C*****************************************************************************
C MU and VAR are the mean and the variance for the prior distribution of
C theta.
C*****************************************************************************
      MU = 0.0
      VAR = 1.0
C*****************************************************************************
C Start iteration
C*****************************************************************************
      COUNT = 0
      DO 30 IT = 1, L
        COUNT = COUNT + 1
C*****************************************************************************
C Update samples for Z from its normal posterior distributions
C*****************************************************************************
        DO 40 I = 1, N
        DO 40 J = 1, K
          LP  = TH(I) * A(J) - G(J)
          BB = ANORDF((0.0 - LP))
          CALL RNUN (1, U)
          TMP = BB*(1 - Y(I, J)) + (1 - BB)*Y(I, J)) * U + BB*Y(I, J)
          Z(I, J) = DNORIN(TMP) + LP
   40     CONTINUE
```

```
C***************************************************************************
C Update samples for theta from their normal posterior distributions
C***************************************************************************
          V = 1/SUM(A*A)
          PVAR = 1/(1/V + 1/VAR)
          DO 50 I = 1, N
           MSUM = 0.0
           DO 60 J = 1, K
             MSUM = MSUM+A(J)*(Z(I, J) + G(J))
   60        CONTINUE
           MN = MSUM*V
           PMEAN = (MN/V + MU/VAR)*PVAR
           CALL RNNOR(1,TT)
           TH(I) = TT*SQRT(PVAR) + PMEAN
           THV(I, COUNT) = TH(I)
   50      CONTINUE
C***************************************************************************
C Update samples for item parameters, a and g from their multivariate
C normal posterior distributions
C***************************************************************************
          DO 70 J = 1, 1
          DO 70 I = 1, N
            X(I, J)  =  TH(I)
   70      CONTINUE
          DO 80 J = 2, 2
          DO 80 I = 1, N
            X(I, J) = -1
   80      CONTINUE
          IF (UNIF = = 0) THEN
C***************************************************************************
C Specify the prior means (AMU, GMU) and variances (AVAR, GVAR)
C for a and g.
C***************************************************************************
            AMU = 0.0
            GMU = 0.0
            AVAR = 1.0
            GVAR = 1.0
C***************************************************************************
C Put the means and variances in vector and matrix format
C***************************************************************************
            AGMU(1, 1) = AMU
            AGMU(2, 1) = GMU
            AGVAR(1, 1) = AVAR
            AGVAR(2, 2) = GVAR
C***************************************************************************
C Call the matrix inversion routine.
```

```
C Invert matrix AGVAR with the inverse stored in SIGMA
C************************************************************************
           CALL MIGS(AGVAR, 2, SIGMA, INDX)
           XX = MATMUL(TRANSPOSE(X), X) + SIGMA
         ELSE IF (UNIF = = 1) THEN
           XX = MATMUL(TRANSPOSE(X), X)
         END IF
C************************************************************************
C Call the matrix inversion routine.
C Invert matrix XX with the inverse stored in IX
C************************************************************************
         CALL MIGS(XX, 2, IX, INDX)
C************************************************************************
C Call the Cholesky factorization routine. Compute the Cholesky
C factorization of the symmetric definite matrix IX and store the
C result in AMAT
C************************************************************************
         CALL CHFAC (2, IX, 2, 0.00001, IRANK, AMAT, 2)
         DO 90 J = 1, K
           DO 100 I = 1, N
               ZV(I, 1)=Z(I, J)
  100        CONTINUE
             IF (UNIF = = 0) THEN
               XZ = MATMUL(SIGMA, AGMU)+MATMUL(TRANSPOSE(X), ZV)
             ELSE IF (UNIF = = 1) THEN
               XZ = MATMUL(TRANSPOSE(X), ZV)
             END IF
               BZ = MATMUL(IX, XZ)
               A(J) = 0
             DO WHILE (A(J).LE.0)
               CALL RNNOR (2, BI)
               BETA = MATMUL(BI, AMAT)+TRANSPOSE(BZ);
               A(J) = BETA(1, 1)
               G(J) = BETA(1, 2)
              END DO
             AV(COUNT, J) = A(J)
             GV(COUNT, J) = G(J)
   90        CONTINUE
   30 CONTINUE
C************************************************************************
C Calculate the posterior means and SEs for a, g and theta and store them
C in ITEM and PERSON
C************************************************************************
      BSIZE=(L-BURNIN)/BN
      DO 110 J = 1, K
         COUNT = BURNIN
         TOT1 = 0.0
         TOT2 = 0.0
         SS1 = 0.0
         SS2 = 0.0
         DO 120 M = 1, BN
             SUM1 = 0.0
             SUM2 = 0.0
             DO 130 I = 1, BSIZE
               COUNT = COUNT + 1
               SUM1 = SUM1 + AV(COUNT, J)
               SUM2 = SUM2 + GV(COUNT, J)
```

```
130       CONTINUE
          M1 = SUM1/BSIZE
          M2 = SUM2/BSIZE
          TOT1 = TOT1 + M1
          TOT2 = TOT2 + M2
          SS1 = SS1 + M1*M1
          SS2 = SS2 + M2*M2
120   COUNTINE
      ITEM(J, 1) = TOT1/BN
      ITEM(J, 2) = SQRT((SS1 - (TOT1*TOT1/BN))/(BN -1))/SQRT(FLOAT(BN))
      ITEM(J, 3) = TOT2/BN
      ITEM(J, 4) = SQRT((SS2 - (TOT2*TOT2/BN))/(BN -1))/SQRT(FLOAT(BN))
110 CONTINUE

    DO 140 J = 1,N
      COUNT = BURNIN
      TOT3 = 0.0
      SS3 = 0.0
      DO 150 M = 1, BN
          SUM3 = 0.0
          DO 160 I = 1, BSIZE
            COUNT = COUNT + 1
            SUM3 = SUM3 + THV(J, COUNT)
160       CONTINUE
          M3 = SUM3/BSIZE
          TOT3 = TOT3 + M3
          SS3 = SS3 + M3*M3
150   CONTINUE
      PERSON(J, 1) = TOT3/BN
      PERSON(J, 2) = SQRT((SS3 - (TOT3*TOT3/BN))/(BN -1))/SQRT(FLOAT(BN))
140 CONTINUE

    RETURN
    END
```