```
4190  %macro mcartest
4191  (indata= ems.recodeptsd           /* Input DATASET name */
4192  ,testvars= sc1 ptsd1_1 ptsd1_2 ptsd1_3 ptsd1_4 ptsd1_5 ptsd1_6 ptsd1_7 ptsd1_8 ptsd1_9
4192! ptsd1_10 ptsd1_11 ptsd1_12 ptsd1_13
4193   ptsd1_14 ptsd1_15 ptsd1_16 ptsd1_17         /* SPECIFY VARIABLE SET FOR THE MCAR TEST */
4194  ,misscode= .      /* SPECIFY THE MISSING VALUE CODE */
4195  );
4196  /*************************************************************************************
4196! ************************
4197  * This SAS macro implements the chi-square test for a missing completely at random (MCAR)
4197! mechanism, as *
4198  * outlined in Little's (1998) JASA article. Note that the macro requires SAS version 8.2
4198! (or higher) because *
4199  * PROC MI is used to obtain ML estimates of the covariance matrix and mean vector. * *
4200  *************************************************************************************
4200! ************************/
4201
4202  %local numvars ;
4203  %let numvars = %sysfunc(countw(&testvars));
4204
4205  data one;
4206    set &indata (keep=&testvars);
4207    array m[&numvars] &testvars ;
4208    array r[&numvars] r1 - r&numvars ;
4209
4210    do i = 1 to &numvars;
4211      if m[i] = &misscode then m[i] = .;
4212      r[i] = not missing(m[i]);
4213    end;
4214    drop i;
4215  run;
4216
4217  proc sort;
4218    by r1-r&numvars;
4219  run;
4220
4221  proc mi data = one nimpute = 0 noprint;
4222    var &testvars;
4223    em outem = emcov;
4224  run;
4225
4226  proc iml;
4227
4228  use one;
4229  read all var {&testvars} into y;
4230  read all var {%do i = 1 %to &numvars; r&i %end;} into r;
4231  use emcov;
4232  read all var {&testvars} into em;
4233
4234  mu = em[1,];
4235  sigma = em[2:nrow(em),];
4236
4237  /* ASSIGN AN INDEX VARIABLE DENOTING EACH CASE'S PATTERN */
4238
4239  jcol = j(nrow(y), 1 , 1);
4240
4241  do i = 2 to nrow(y);
```

```
4242    rdiff = r[i,] - r[i - 1,];
4243    if max(rdiff) = 0 & min(rdiff) = 0 then jcol[i,] = jcol[i - 1,];
4244    else jcol[i,] = jcol[i - 1,] + 1;
4245  end;
4246
4247  /* NUMBER OF DISTINCT MISSING DATA PATTERNS */
4248
4249  j = max(jcol);
4250
4251  /* PUT THE NUMBER OF CASES IN EACH PATTERN IN A COL VECTOR M */
4252  /* PUT THE MISSING DATA INDICATORS FOR EACH PATTERN IN A MATRIX RJ */
4253
4254  m = j(j, 1, 0);
4255  rj = j(j, ncol(r), 0);
4256
4257  do i = 1 to j;
4258    count = 0;
4259    do k = 1 to nrow(y);
4260      if jcol[k,] = i then do;
4261        count = count + 1;
4262      end;
4263      if jcol[k,] = i & count = 1 then rj[i,] = r[k,];
4264      m[i,] = count;
4265    end;
4266  end;
4267
4268  /* COMPUTE D^2 STATISTIC FOR EACH J PATTERN */
4269
4270  d2j = j(j, 1, 0);
4271
4272  do i = 1 to j;
4273
4274  /* OBSERVED VALUES FOR PATTERN J */
4275    yj = y[loc(jcol = i),loc(rj[i,] = 1)];
4276
4277  /* VARIABLE MEANS FOR PATTERN J */
4278    ybarobsj = yj[+,]/nrow(yj);
4279
4280  /* D = P X Pj MATRIX OF INDICATORS (SEE P. 1199) */
4281    Dj = j(ncol(y), rj[i,+], 0);
4282
4283    count = 1;
4284    do k = 1 to ncol(rj);
4285      if rj[i,k] = 1 then do;
4286        Dj[k, count] = 1;
4287        count = count + 1;
4288      end;
4289    end;
4290
4291  /* REDUCE EM ESTIMATES TO CONTAIN OBSERVED ELEMENTS */
4292    muobsj = mu * Dj;
4293    sigmaobsj = t(Dj) * sigma * Dj;
4294
4295  /* THE CONTRIBUTION TO THE D^2 STATISTIC FOR EACH OF THE J PATTERNS */
4296    d2j[i,] = m[i,] * (ybarobsj - muobsj) * inv(sigmaobsj) * t(ybarobsj - muobsj);
4297
4298  end;
```

```
4299
4300  /* THE D^2 STATISTIC */
4301  d2 = d2j[+,];
4302
4303  /* DF FOR D^2 */
4304  df = rj[+,+] - ncol(rj);
4305  p = 1 - probchi(d2,df);
4306
4307  /* PRINT ANALYSIS RESULTS */
4308  file print;
4309  put "Number of Observed Variables = " (ncol(rj)) 3.0;
4310  put "Number of Missing Data Patterns = " (j) 3.0; put;
4311  put "Summary of Missing Data Patterns (0 = Missing, 1 = Observed)"; put;
4312  put "Frequency | Pattern | d2j"; put;
4313  do i = 1 to nrow(rj);
4314    put (m[i,]) 6.0 " | " @;
4315    do j = 1 to ncol(rj);
4316      put (rj[i,j]) 2.0 @;
4317    end;
4318    put " | " (d2j[i,]) 8.6;
4319  end;
4320  put;
4321  put "Sum of the Number of Observed Variables Across Patterns (Sigma psubj) = " (rj[+,+])
4321! 5.0; put;
4322  put "Little's (1988) Chi-Square Test of MCAR"; put;
4323  put "Chi-Square (d2) = " (d2) 10.3;
4324  put "df (Sigma psubj - p) = " (df) 7.0;
4325  put "p-value = " (p) 10.3;
4326
4327  %mend mcartest;
4328
4329  %mcartest
4330  quit;

NOTE: There were 2683 observations read from the data set EMS.RECODEPTSD.
NOTE: The data set WORK.ONE has 2683 observations and 36 variables.
NOTE: DATA statement used (Total process time):
      real time           0.03 seconds
      cpu time            0.01 seconds



NOTE: There were 2683 observations read from the data set WORK.ONE.
NOTE: The data set WORK.ONE has 2683 observations and 36 variables.
NOTE: PROCEDURE SORT used (Total process time):
      real time           0.01 seconds
      cpu time            0.01 seconds



NOTE: The EM algorithm (MLE) converges in 4 iterations.
NOTE: The data set WORK.EMCOV has 19 observations and 20 variables.
NOTE: PROCEDURE MI used (Total process time):
      real time           0.04 seconds
      cpu time            0.06 seconds
```

```
NOTE: IML Ready
ERROR: (execution) Matrix has not been set to a value.

 operation : [ at line 4329 column 1
 operands  : y, _TEM1002, _TEM1005
y    2683 rows     18 cols    (numeric)
_TEM1002       1 row     741 cols    (numeric)

_TEM1005       0 row       0 col     (type ?, size 0)


 statement : ASSIGN at line 4329 column 1
NOTE: Exiting IML.
NOTE: The SAS System stopped processing this step because of errors.
NOTE: PROCEDURE IML used (Total process time):
      real time           0.14 seconds
      cpu time            0.12 seconds

4331  run;
```