



Data screening and summarization

Contents

Data screening.....	2
Variables and scale types.....	3
Labels for categories.....	4
Frequencies, percentages, bar charts, histograms.....	4
Extension of the select cases command.....	6
Specification of missing values.....	7
Imputation of missing values.....	8
Simplified syntax for reading and saving matrices.....	13
Saving the raw data.....	13
New variables.....	14
Merging files.....	16

Most raw data for analyses are stored or downloaded from large files at data archives. The raw data file may contain many variables on many cases. Before doing more elaborate analysis of the data, it is important to do a careful data screening to check for coding errors or other mistakes in the data. Such data screening will also give a general idea of the character and the quality of the data.

PRELIS automatically does such data screening by determining for each variable the distinct data values present in the data and the number of each. If a variable has more than 15 distinct values, PRELIS will group them in intervals and determine the number in each interval.

Labels for categories

One can specify labels (names) for categories of categorical (ordinal) variables. This makes it easier to read and interpret univariate and bivariate distributions of categorical (ordinal) variables. The syntax for doing this is

```
CL varlist n1=c1ab1 n2 = c1ab2
```

where *varlist* is a list of variables by labels or numbers, *n1*, *n2*, ... are the numerical values in the data on these variables, and *c1ab1*, *c1ab2*, ... are the labels to be assigned to these numerical values. Each category label may contain up to four characters.

The following input file (**EX7B.PRL**) will read the six political efficacy items from the file **EX7.RAW** and assign category labels AS, A, D, DS, DK, and NA to the numerical values 1, 2, 3, 4, 8, and 9, respectively (AS = Agree Strongly, A = Agree, D = Disagree, DS = Disagree Strongly, DK = Don't Know, NA = No Answer).

EXAMPLE 7B

```
!Assigning Category Labels as follows
!AS = Agree Strongly, A = Agree, D = Disagree, DS = Disagree Strongly,
!DK = Don't Know, NA = No Answer
DA NI=6
LA;NOSAY VOTING COMPLEX NOCARE TOUCH INTEREST ! Note the ;
RA=EX7.RAW FO;(T142,6F2.0) ! Note the ;
CL NOSAY - INTEREST 1=AS 2=A 3=D 4=DS 8=DK 9=NA
OU
```

Note that one must know which numerical values are present in the data to do this. If these are unknown, screen the data first, as shown earlier in this document.

Each CL command contains a *varlist* listing all variables with the same category labels, followed by a listing defining category labels for numerical values. If the category labels do not fit on a line of 127 characters, repeat the same CL command and *varlist* and continue with the remaining category labels. If the *varlist* on the CL command is too long to fit on one line of 127 characters, write the remaining variables in *varlist* on a new CL command followed by the same list of category labels as for the previous CL command. Note that, if some of the variables are recoded, the category labels refer to the recoded values.

Frequencies, percentages, bar charts, histograms

For both univariate and bivariate distributions of categorical (ordinal) variables, frequencies are given in absolute as well as relative form (percentages). In addition, bar charts representing the univariate marginal distribution are given for each ordinal variable. For continuous variables, PRELIS gives a histogram in addition to the statistics. Tests of univariate and multivariate normality are also given for continuous variables.

The output for this example illustrates this:

Specification of missing values

To specify several values to be treated as missing in *all variables globally*, put `MI = a,b,c` on the `DA` command. To assign several missing values to a *specific set of variables*, write an `MI` command as follows.

```
MI = a,b,c,... varList
```

where *varList* is a list of variables and *a,b,c,...* are numerical values to be treated as missing values. If there is more than one missing value, there must be commas between the values, but not after the last missing value. Each of the *a,b,c,...* can be one of the following:

- A single value
- A range of values from *x* to *y* inclusive, denoted *x-y*.

The following input file (**EX7C.PRL**) reads the same data as before, and eliminates all cases with missing data. Category labels are then assigned to the data values that remain after listwise deletion and the data screening is done on this subsample.

EXAMPLE 7C

Listwise Elimination of Missing Values (DK and NA Responses)

```
DA NI=6 MI=8,9
```

```
LA;NOSAY VOTING COMPLEX NOCARE TOUCH INTEREST !Note the ;
```

```
RA FI=EX7.RAW FO;(T142,6F2.0) !Note the ;
```

```
CL NOSAY - INTEREST 1=AS 2=A 3=D 4=DS
```

```
OU
```

The following tables in the output file gives information about how the missing values are distributed over variables and cases.

Effective Sample Sizes

Univariate (in Diagonal) and Pairwise Bivariate (off Diagonal)

	NOSAY	VOTING	COMPLEX	NOCARE	TOUCH	INTEREST
	-----	-----	-----	-----	-----	-----
NOSAY	312					
VOTING	312	312				
COMPLEX	290	290	290			
NOCARE	312	312	290	312		
TOUCH	312	312	290	312	312	
INTEREST	312	312	290	312	312	312

Percentage of Missing Values

Univariate (in Diagonal) and Pairwise Bivariate (off Diagonal)

	NOSAY	VOTING	COMPLEX	NOCARE	TOUCH	INTEREST
	-----	-----	-----	-----	-----	-----
NOSAY	0.00					
VOTING	0.00	0.00				

COMPLEX	7.05	7.05	7.05			
NO CARE	0.00	0.00	7.05	0.00		
TOUCH	0.00	0.00	7.05	0.00	0.00	
INTEREST	0.00	0.00	7.05	0.00	0.00	0.00

Of the 312 cases in the data, there are only 282 without missing values, so this is the listwise sample on which the data screening is done.

Imputation of missing values

In PRELIS there are two ways of handling missing values: pairwise and listwise deletion. In many situations, particularly when values are missing not completely at random, these procedures are far from satisfactory (see, for example, Little & Rubin, 1987, and Rubin, 1987). PRELIS offers yet another possibility of handling missing values, namely by imputation, i.e., by substitution of real values for the missing values. The value to be substituted for the missing value for a case is obtained from another case that has a similar response pattern over a set of matching variables. To do this, include a line

IN (*Ivarlist*) (*Mvarlist*) VR = XN XL

in the input file, where *Ivarlist* is a set of variables whose missing values should be imputed and *Mvarlist* is a set of matching variables. VR, XN, and XL are explained below.

The imputation scheme is as follows. Let y_1, y_2, \dots, y_p denote the variables in *Ivarlist* and let x_1, x_2, \dots, x_q denote the variables in *Mvarlist*. To begin, let us assume that there is only a single variable y in *Ivarlist* whose missing values are to be imputed and that y is not included in *Mvarlist*. Let z_1, z_2, \dots, z_q be the standardized x_1, x_2, \dots, x_q , i.e. for each case c

$$z_{cj} = \left(x_{cj} - \bar{x}_j \right) / s_j \quad j = 1, 2, \dots, q,$$

where \bar{x}_j and s_j are the estimated mean and standard deviation of x_j . These are estimated from all complete data on x_j .

The imputation procedure is as follows.

1. Find the first case a with a missing value on y and no missing values on x_1, x_2, \dots, x_q . If no such case exists, imputation of y is impossible. Otherwise, proceed to impute the value y_a as follows.
2. Find *all* cases b which have no missing value on y and no missing values on x_1, x_2, \dots, x_q , and which minimizes

$$\sum_{j=1}^n (z_{bj} - z_{\alpha j})^2. \quad (1)$$

3. Two cases will occur

- If there is a single case b satisfying 2, y_a is replaced by y_b .
- Otherwise, if there are $n > 1$ matching cases b with the same minimum value, denote their y -values by $y_1^{(m)}, y_2^{(m)}, \dots, y_n^{(m)}$. Let

$$\bar{y}_m = (1/n) \sum_{i=1}^n y_i^{(m)}, \quad s_m^2 = [1/(n-1)] \sum_{i=1}^n (y_i^{(m)} - \bar{y}_m)^2,$$

be the mean and variance of the y -values of the matching cases. Then, imputation will be done only if

$$\frac{s_m^2}{s_y^2} < v, \quad (2)$$

where s_y^2 is the total variance of y estimated from all complete data on y , and v is the value VR specified on the MI command. This may be interpreted to mean that the matching cases predict the missing value with a reliability of at least $1 - v$. The default value of VR is VR = 0.5, i.e., $v = 0.5$. Larger values than this are not recommended. Smaller values may be used if one requires high precision in the imputation. For each value imputed, PRELIS gives the value of the variance ratio and the number of cases on which s_m^2 is based.

If condition (2) is satisfied, then y_a is replaced with the mean \bar{y}_m if y is continuous or censored, or with the value on the scale of y closest to \bar{y}_m if y is ordinal. Otherwise, no imputation is done and y_a is left as a missing value.

4. This procedure is repeated for the next case a for which y_a is missing, and so on, until all possible missing values on y have been imputed.

This procedure has the advantage that it gives the same results under linear transformation of the matching variables. Thus, if age is a matching variable, age can be in years or months, or represented by the year of birth, and the resulting imputed data will be the same in each case. Another advantage is that the results of the imputation will be the same regardless of the order of cases in the data.

If *lvarlist* contains several variables, they will be imputed in the order they are listed. This is of no consequence if no variables in *lvarlist* is included in *Mvarlist*. Ideally, *lvarlist* contains the variables with missing values and *Mvarlist* contains variables without missing values. However, PRELIS can also handle the case when some variables are included in both *varlists* and, in fact, the two *varlists* could be identical. If a variable is included in both, it is automatically excluded from the *Mvarlist* when its values are imputed.

In this case, the order of the variables in *lvarlist* can make a difference, since a variable already imputed can be used as matching variable when another variable is imputed.

Imputation of missing values should be done with utmost care and control, since missing values will be replaced by other values that will be treated as real observed values. If possible, use matching variables which are *not* to be used in the LISREL modeling. Otherwise, if the matching variables are included in the LISREL model, it is likely that the imputation will affect the result of the analysis. This should be checked by comparing with the result obtained without imputation.

For each variable to be imputed, PRELIS lists all the cases with missing values. If imputation is successful, it gives the value imputed, the number of matching cases and the variance ratio. If the imputation is not successful, it gives the reason for the failure. This can be either that no matching case was found or that the variance ratio was too large. The XN option on the IM command will make PRELIS list only successful imputations, and the XL option makes PRELIS skip the entire listing of cases. PRELIS always gives the number of missing values per variable, both before and after imputation.

Important note: Imputation is done after all recodings and other data transformations. If it is required to do recodings or other transformation on the imputed data, one can do this by running PRELIS twice. In the first run, do the imputation and save the raw data. In the second run, use the saved raw data after imputation and do the recoding or other transformations.

The following example (**EX7D.PRL**) reads the same data as used previously in this document. The missing values on each variable are imputed using all the other variables as matching variables. Cases with missing values are eliminated after imputation. Category labels are then assigned to the data values that remain after listwise deletion and the data screening is done of this subsample.

```
EXAMPLE 7D
Imputing Missing Values
DA NI=6 MI=8,9
LA;NOSAY VOTING COMPLEX NOCARE TOUCH INTEREST
RA FI=EX7.RAW FO;(T142,6F2.0)
IM (NOSAY - INTEREST) (NOSAY - INTEREST)
CL NOSAY - INTEREST 1=AS 2=A 3=D 4=DS
OU
```

The output file gives the following information concerning missing values and imputation.

Number of Missing Values per Variable

NOSAY	VOTING	COMPLEX	NOCARE	TOUCH	INTEREST
5	8	3	7	14	14

Imputations for NOSAY

```
Case 56 not imputed because of missing values for matching variables
Case 88 imputed with value 3 (Variance Ratio = 0.393), NM= 4
Case 99 not imputed because of missing values for matching variables
Case 229 not imputed because of missing values for matching variables
Case 274 imputed with value 3 (Variance Ratio = 0.315), NM= 11
```

Imputations for VOTING

Case 13 not imputed because of Variance Ratio = 2.312 (NM= 6)
Case 18 not imputed because of missing values for matching variables
Case 62 not imputed because of missing values for matching variables
Case 99 not imputed because of missing values for matching variables
Case 138 imputed with value 1 (Variance Ratio = 0.000), NM= 1
Case 180 not imputed because of missing values for matching variables
Case 188 not imputed because of missing values for matching variables
Case 257 imputed with value 2 (Variance Ratio = 0.324), NM= 13

Imputations for COMPLEX

Case 143 not imputed because of missing values for matching variables
Case 188 not imputed because of missing values for matching variables
Case 240 imputed with value 2 (Variance Ratio = 0.394), NM= 18

Imputations for NOCARE

Case 40 not imputed because of missing values for matching variables
Case 143 not imputed because of missing values for matching variables
Case 144 imputed with value 3 (Variance Ratio = 0.000), NM= 1
Case 206 not imputed because of missing values for matching variables
Case 229 not imputed because of missing values for matching variables
Case 233 imputed with value 3 (Variance Ratio = 0.000), NM= 1
Case 270 imputed with value 3 (Variance Ratio = 0.000), NM= 7

Imputations for TOUCH

Case 18 not imputed because of missing values for matching variables
Case 28 not imputed because of missing values for matching variables
Case 29 imputed with value 2 (Variance Ratio = 0.000), NM= 1
Case 37 imputed with value 2 (Variance Ratio = 0.000), NM= 2
Case 40 not imputed because of missing values for matching variables
Case 56 not imputed because of missing values for matching variables
Case 62 not imputed because of missing values for matching variables
Case 99 not imputed because of missing values for matching variables
Case 104 imputed with value 2 (Variance Ratio = 0.000), NM= 1
Case 143 not imputed because of missing values for matching variables
Case 188 not imputed because of missing values for matching variables
Case 203 not imputed because of missing values for matching variables
Case 209 not imputed because of Variance Ratio = 0.618 (NM= 5)
Case 238 imputed with value 1 (Variance Ratio = 0.000), NM= 1

Imputations for INTEREST

Case 12 not imputed because of Variance Ratio = 0.611 (NM= 3)
Case 18 not imputed because of missing values for matching variables
Case 28 not imputed because of missing values for matching variables
Case 48 imputed with value 2 (Variance Ratio = 0.000), NM= 2
Case 56 not imputed because of missing values for matching variables

Case 62 not imputed because of missing values for matching variables
 Case 64 imputed with value 3 (Variance Ratio = 0.000), NM= 1
 Case 67 imputed with value 2 (Variance Ratio = 0.000), NM= 1
 Case 99 not imputed because of missing values for matching variables
 Case 180 not imputed because of missing values for matching variables
 Case 188 not imputed because of missing values for matching variables
 Case 203 not imputed because of missing values for matching variables
 Case 206 not imputed because of missing values for matching variables
 Case 229 not imputed because of missing values for matching variables

Number of Missing Values per Variable After Imputation

NOSAY	VOTING	COMPLEX	NOCARE	TOUCH	INTEREST
3	6	2	4	10	11

Distribution of Missing Values

Total Sample Size(N) = 312

Number of Missing Values	0	1	2	3	4
Number of Cases	297	3	5	5	2

Effective Sample Sizes

Univariate (in Diagonal) and Pairwise Bivariate (off Diagonal)

	NOSAY	VOTING	COMPLEX	NOCARE	TOUCH	INTEREST
NOSAY	309					
VOTING	304	306				
COMPLEX	307	305	310			
NOCARE	306	302	307	308		
TOUCH	301	300	302	300	302	
INTEREST	301	300	300	299	298	301

Percentage of Missing Values

Univariate (in Diagonal) and Pairwise Bivariate (off Diagonal)

	NOSAY	VOTING	COMPLEX	NOCARE	TOUCH	INTEREST
NOSAY	0.96					
VOTING	2.56	1.92				
COMPLEX	1.60	2.24	0.64			
NOCARE	1.92	3.21	1.60	1.28		
TOUCH	3.53	3.85	3.21	3.85	3.21	
INTEREST	3.53	3.85	3.85	4.17	4.49	3.53

Fifteen data values were successfully imputed, two in NOSAY, two in VOTING, one in COMPLEX, three in NOCARE, four in TOUCH, and three in INTEREST. The listwise sample was increased from 282 to 297. Many cases could not be imputed because of missing values in the matching variables. Only a few cases

could not be imputed because of a variance ratio being too large. For more successful examples of imputation, see Jöreskog & Aish (1996).

Simplified syntax for reading and saving matrices

In PRELIS, `SM = filename` on the OU command is used to save the estimated covariance or correlation matrix in a file. One can also use any of the moment matrix names instead of SM to specify the filename to be used. Thus, one may use AM, CM, KM, MM, OM, PM, RM, or TM. This extension is analogous to LISREL syntax where the LISREL command `PM FI = filename` may be simplified to `PM = filename`. Similarly, in LISREL one can write `CM = filename`, `KM = filename`, `RM = filename`, `TM = filename`, `OM = filename`, etc. Furthermore, one can write `AM = filename` in PRELIS instead of `SA = filename`, and `AC = filename` in LISREL instead of `AC FI = filename`. Also, in PRELIS one can use the syntax `XX = filename` instead of `XX FI = filename`, where XX may be RA, LA, or FT.

Saving the raw data

Suppose the raw data contains many cases and very many variables written in a complicated format with lots of redundant spaces and digits. The file containing the raw data is very large and it takes a long time to read it each time. Suppose we wish to save the raw data on a small subset of the variables (and possibly also for a subset of the cases) in a compact form for future use and documentation. This can be done by reading the subset of variables by format or SD commands and saving the new raw data by putting `SR = filename` on the OU command. Instead of `SR = filename`, one can write `RA = filename` and the format of the new data file can be specified by `WI = width-of-format` and `ND = number-of-decimals`. For example, if the data on the selected variables are single-digit positive numbers, as is often the case, `WI = 1 ND = 0`, will save the data in one-column fields. This will save a lot of space and in future PRELIS runs, the data can be read much faster by the format (`nF1.0`). Note that PRELIS can write data in this form, whereas in FORTRAN this format can only be used to read data, not to write data.

The following input file (**EX7E.PRL**) saves the raw data on the six variables after imputation, recoding, and listwise deletion. The variables are recoded so they measure Efficacy from low to high. The new raw data is saved in the file **EFFICACY.RAW** in the format (`6F1.0`).

EXAMPLE 7E

Imputing Missing Values

Recoding All Variables from Low to High

Saving the Raw Data after Imputation, Recoding, and Listwise Deletion

DA NI=6 MI=8,9

LA;NOSAY VOTING COMPLEX NOCARE TOUCH INTEREST

RA FI=EX7.RAW FO;(T142,6F2.0)

IM (NOSAY - INTEREST) (NOSAY - INTEREST)

RE NOSAY - INTEREST OLD=1,2,3,4 NEW=4,3,2,1

CL NOSAY - INTEREST 1=DS 2=D 3=A 4=AS

OU RA=EFFICACY.RAW WI=2 ND=0

The file **EFFICACY.RAW** has 297 lines of 6 columns and contains the complete data on the six variables for further study.

New variables

Let x_1, x_2, \dots, x_k denote a set of variables whose labels have been defined. A new variable y can be defined as a function of already defined variables x_1, x_2, \dots, x_k . The general functional form is

$$y = \sum_i a_i \prod_{j=1}^k x_j^{b_j},$$

where a_i is any positive or negative constant and b_j is any constant including zero. Note that $b_j = 0$ means that y does not depend on x_j . If $b_j = 1$, it may be omitted.

New variables can be defined by the following line

```
NE vary = some function of old variables
```

where vary defines the label for the new variable. If vary is an already existing label, the new variable replaces the old. A separate NE command is used to define each new variable.

The formula defining the new variable is written in a straightforward way. Multiplication is specified by a * and exponentiation by ** or ^. Parentheses are not permitted, but this is no restriction because new variables can be functions of other new variables already defined. Division is not permitted, although exponents may be negative. Constants are written as usual. If they are written without a decimal point, they are taken to be integers.

The new variables will be computed for each case in the data. If any of the variables x_1, x_2, \dots, x_k defining the new variable is missing for a case, the value of the new variable will be missing for this case.

Examples:

```
NE var5=2.5*var1+var2-var3
NE var4=vr4-var2
NE var6=var1**2
NE var7=var1*var2
NE var8=var1+var2*vaar3-var2**2
NE var8=var8**2
```

The last two command illustrate how one can define var8 as the square of a quadratic expression.

New variables can also be transformed by power or log transformations or recoded. For example, if x is a defined variable, one may define two new variables $y_1 = \sqrt{1+x^2}$ and $y_2 = \log(1+x^2)$ with the commands:

```
NE Y1=1+X**2
NE Y2=1+X**2
PO Y1 GA=.5
LO Y2
```

Suppose a factor analysis of the six variables NOSAY, VOTING, COMPLEX, NOCARE, TOUCH, and INTEREST has confirmed that there are two factors, Efficacy and Responsiveness, that NOSAY, VOTING and COMPLEX are valid indicators of Efficacy, and that NOCARE, TOUCH, and INTEREST are valid indicators of Responsiveness (see Aish & Jöreskog, 1990). We want to compute scale scores for Efficacy and Responsiveness defined as the sum of their respective indicators and we want to compute the correlation between these scale scores and the original variables as well as the correlation between the two scale scores. Alternatively, one may want to compute factor scores defined as linear combinations of the observed variables with coefficients obtained as the factor scores regression coefficients estimated by LISREL (see Jöreskog & Sörbom, 1989, p 93).

To obtain the required scores, add two new variables Efficacy and Respons to the six original variables. The keyword NI specified on the DA command will be increased by one for each NE command defining a *new* label so that the new variables will automatically be included in the analysis. Then, the input file (**EX7F.PRL**) is as follows:

```
EXAMPLE 7F
Computing Scale Scores
DA NI=6
LA;NOSAY VOTING COMPLEX NOCARE TOUCH INTEREST
RA=EFFICACY.RAW FO;(6F2.0)
NEW EFFICACY = NOSAY + VOTING + COMPLEX
NEW RESPONS = NOCARE + TOUCH + INTEREST
OU MA=KM WP
```

The resulting correlation matrix is of order 8 x 8:

Correlation Matrix

	NOSAY	VOTING	COMPLEX	NOCARE	TOUCH	INTEREST	EFFICACY	RESPONS
NOSAY	1.000							
VOTING	0.326	1.000						
COMPLEX	0.242	0.289	1.000					
NOCARE	0.441	0.268	0.440	1.000				
TOUCH	0.337	0.167	0.315	0.663	1.000			
INTEREST	0.399	0.223	0.368	0.711	0.630	1.000		
EFFICACY	0.749	0.791	0.757	0.512	0.361	0.440	1.000	
RESPONS	0.442	0.241	0.415	0.929	0.888	0.910	0.490	1.000

Total Variance = 8.000 Generalized Variance = 0.134

Largest Eigenvalue = 4.549 Smallest Eigenvalue = -0.717988D-01

The last two rows of the correlation matrix give the correlations between the scale scores and the original variables, as well as the correlation between the scale scores themselves. Since no variables are declared continuous in the input file, all variables, including EFFICACY and RESPON, are treated as ordinal. As a result of the NE commands, EFFICACY and RESPON will have 10 categories with scores ranging from 3 to 12. Since MA = PM, the matrix of polychoric correlations is computed.

Other possibilities are to use MA = KM and treat all variables as continuous to obtain a matrix of product-moment correlations, or to use MA = PM and declare EFFICACY and RESPON as continuous variables to obtain a matrix of polychoric, polyserial, and product-moment (Pearson) correlations. We leave it to the reader to estimate different type of correlations and compare them.

Merging files

It is a common situation in practice when working with survey data that some variables are in one file and some are in another file. In particular, this is common in longitudinal or panel studies, where the data have been recorded in separate files for each occasion. In these situations, it is useful to be able to combine some variables from each file and store them in one file for further study.

Suppose that files fname1, fname2, ... contain different sets of variables, all measured on the same cases. To read the variables from different files, use the following type of DA and RA commands:

```
DA NI=n1,n2,...  
RA=fname1, fname2,...
```

where n1, n2, ... are the number of variables to read from the respective files. As with data from a single file, if no format is given in the input file or the data files, the variables will be read in free format. To read formatted data, put the format first in each data file or put the formats for each data file one after another after the RA command in the input file (in that case the FO option should be given on the RA command).

The files **PANUSA1.RAW** and **PANUSA2.RAW** contain the six Efficacy items previously discussed. The first file contains the data collected at the first occasion and the second file contain the data collected at the second occasion. Both files contain data on 933 cases. The following input file (**EX8.PRL**) will read the data from both files and store the data on all 12 variables in a new file **PANUSA.RAW**.

EXAMPLE 8: Merging two Files

```
DA NI=6,6  
LA  
NOSAY1 VOTING1 COMPLEX1 NOCARE1 TOUCH1 INTERES1  
NOSAY2 VOTING2 COMPLEX2 NOCARE2 TOUCH2 INTERES2  
RA=PANUSA1.RAW,PANUSA2.RAW FO  
(6F1.0)
```

```
(6F1.0)  
OU RA=PANUSA.RAW WI=1 ND=0
```

Note that two formats must follow the FO option on the RA command, and each must begin on a new line.