| This file contains information and sample SAS programs to create a permanent SAS dataset for users who want to use SAS in processing the MEPS data provided in this PUF release.  There are two ways to create a permanent SAS dataset, using either the SAS transport data file (H160IF1.SSP) or the ASCII data file (H160IF1.DAT)  |
|--|
| •  |
| transport data file (H160IF1.SSP) or the ASCII data file (H160IF1.DAT) supplied in this PUF release. Section A provides a sample SAS program for the first alternative, which is to convert the SAS transport data file to a   |
| regular SAS dataset using the SAS PROCedure: XCOPY. Section B provides a sample SAS program for the second alternative, which is to read data from the ASCII data file using a SAS DATA step with INFILE, INPUT, and LABEL statements. Section C explains format-related SAS statements that a user may  |
| optionally use when working with the SAS dataset. Examples of SAS programs (DATA step or PROC) are provided in all three sections, primarily for the benefit of inexperienced users. Section D contains complete SAS statements that must be used in the programs described in Sections B and C.   |
| INCLUDED BELOW ARE NOTES APPLICABLE TO USERS OF SAS VERSION 8 OR HIGHER.   |
| The sample SAS programs provided in Sections A and B show how to create a permanent SAS dataset from the data files provided in this PUF release.  |
| A. A Sample SAS Program for Converting the SAS Transport File to a Permanent SAS Dataset  The SAS PROCedure XCOPY will read a SAS transport file and convert the   |
| data to regular SAS format, storing the output in a permanent SAS dataset. This permanent SAS dataset can then be used for all future processing and analyses.   |
| Below is a sample SAS program that can be used to convert the SAS transport file to a permanent SAS dataset (in a Windows environment, with SAS V8 or higher).   |
| LIBNAME PUFLIB 'C:\MEPS\SASDATA'; FILENAME IN1 'C:\MEPS\DOWNLOAD\H160IF1.SSP';   |
| PROC XCOPY IN=IN1 OUT=PUFLIB IMPORT; RUN;  SAS transport files, SAS data files, and SAS program files each should be   |
| stored in separate locations (directory names). Storing different types of SAS files in one location can cause errors with converting or retrieving data.  |
| Below are SAS statements to print a list of variables and a few sample records from the permanent SAS dataset:  PROC CONTENTS DATA=PUFLIB.H160IF1;   |
| TITLE 'List of Variables in MEPS H160IF1 SAS Dataset'; RUN; PROC PRINT DATA=PUFLIB.H160IF1 (OBS=20);   |
| TITLE 'First 20 Observations in MEPS H160IF1 SAS Dataset'; RUN;  The LIBNAME statement tells SAS the location (directory name) to store the  |
| permanent SAS dataset which is output by PROC XCOPY. The FILENAME statement tells SAS the location (complete directory and file name) of the input SAS transport data file.  |
| NOTES:  1) If you have an error reading a SAS data file you created, the problem may be a result of where you are storing and/or how you are retrieving  |
| the data. First check the data library for multiple releases of SAS files (e.g., V8 or higher with file extensions of '.SAS7BDAT' and V6 with file extensions of '.SD2') stored in the same location.  |
| a) You can avoid errors when reading these files by including<br>the SAS release within the LIBNAME statement - e.g.,<br>LIBNAME PUFLIB V8 'C:\MEPS\SASDATA';<br>or  |
| b) Store SAS data files with different file extensions such as .SD2 and .SAS7BDAT, in separate folders (do not co-mingle V8 and V6   |
| files in the same folder); or  c) When importing transport files, output the SAS dataset to  |
| <pre>a different library than the one which contains the downloaded SAS transport file - e.g.,</pre>   |
| PROC XCOPY IN=IN1 OUT=PUBLIB IMPORT; RUN;  2) The names used in the LIBNAME and FILENAME statements shown  |
| above (i.e., PUFLIB, IN1) are arbitrary; they are only temporary aliases.  |
| 3) The directory and file names used in the LIBNAME and FILENAME<br>statements shown above are Windows syntax and may need to be<br>modified for other operating systems such as UNIX, MAC/OS, VMS, or<br>OS/2.  |
| 4) H160IF1 is the internal SAS dataset name (also the PC file name, without the extension) prior to the creation of the SAS transport data file. After running PROC XCOPY, the output SAS dataset assumes the same   |
| dataset name (or file name). Hence, in the example above, a file named H160IF1.SAS7BDAT will be created under the C:\MEPS\SASDATA directory when PROC XCOPY runs successfully.   |
| 5) The SAS transport file H160IF1.SSP was created from a SAS V9 data file, using PROC COPY. This file has been tested for use with SAS V8 or higher. This file may work with earlier versions of SAS, although it has not been tested with those versions. Users who are   |
| SAS, although it has not been tested with those versions. Users who are unable to use this SAS transport file should instead convert the ASCII data file H160IF1.DAT to a SAS dataset as described in Section B.   |
| B. A Sample SAS Program for Converting the ASCII Data File to a Permanent SAS Dataset  |
| The complete SAS statements (INPUT and LABEL) included in Section D are intended to save time for those users wishing to create a permanent SAS dataset from the H160IF1.DAT ASCII data file. These statements must be used in combination with other SAS statements to create the appropriate SAS   |
| in combination with other SAS statements to create the appropriate SAS program, as shown below. To use the statements provided in Section D to create a SAS program, you will need an ASCII text editor. If you are using an interactive form of SAS (Windows, UNIX, OS2, etc.), use the editor provided as  |
| part of the SAS software.  Following is a sample SAS program that will convert the ASCII data file to SAS format:  |
| LIBNAME PUFLIB 'C:\MEPS\SASDATA'; FILENAME IN1 'C:\MEPS\DOWNLOAD\H160IF1.DAT';   |
| DATA PUFLIB.H160IF1; INFILE IN1 LRECL=59; INPUT; * to user: insert the complete INPUT statement that is  |
| provided in Section D;  LABEL; * to user: insert the complete LABEL statement that is provided in Section D;  RUN;   |
| Here is an explanation of the SAS statements used in the program above.  |
| LIBNAME statement: This tells SAS the location (directory name) of the permanent SAS dataset.  FILENAME statement: This tells SAS the location of the input ASCII data file.   |
| DATA statement: This signifies the beginning of a SAS DATA step and specifies the output SAS dataset, referencing the LIBNAME entry (PUFLIB) and assigning an internal SAS dataset name (H160IF1). In the example, after the successful  |
| completion of the DATA step, a PC file named H160IF1.SAS7BDAT would have been created in the C:\MEPS\SASDATA directory.  INFILE statement: This tells SAS the location (directory and file name) of the  |
| input ASCII data file. Also provided is the logical record length (59 bytes), with the default of RECFM=V implied when this parameter is omitted. LRECL and RECFM are optional parameters in the INFILE statement. With regard   |
| to these options, please note the following: the ASCII data file H160IF1.DAT contains a 2-byte carriage return/line feed at the end of each record. When converting to a PC-SAS file, the LRECL option should be used to specify the record length to avoid use of a default record length by PC-SAS. If the   |
| RECFM=V option is used, the LRECL option must be specified as the logical record length (e.g., 59 for H160IF1.DAT). If RECFM=F is used, then the LRECL value must be specified as the logical record length plus 2 (61 for   |
| H160IF1.DAT). Note that if the RECFM option is omitted, then the default option of RECFM=V is automatically used, and LRECL should be specified as the logical record (59 for H160IF1.DAT).  |
| INPUT statement: This specifies the input record layout, giving names and the beginning and ending column positions for data items (which become SAS variables) in the ASCII data file (H160IF1.DAT). Variable type (numeric or character) is also defined via the INPUT statement.  |
| LABEL statement: This associates descriptive names with the SAS variables.   |
| RUN statement: This tells SAS to execute all commands up to this point.  See Section A.1 above for tips on retrieving and storing the permanent SAS data files.  |
|  |
| C. Optional Format-related SAS Statements  |
| If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this:  |
| If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this:  LIBNAME PUFLIB 'C:\MEPS\SASDATA';  PROC FORMAT LIBRARY=PUFLIB; VALUE; * to user: insert the complete set of VALUE statements found  |
| If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this:  LIBNAME PUFLIB 'C:\MEPS\SASDATA';  PROC FORMAT LIBRARY=PUFLIB; VALUE; * to user: insert the complete set of VALUE statements found in Section D; VALUE;   |
| If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this:  LIBNAME PUFLIB 'C:\MEPS\SASDATA';  PROC FORMAT LIBRARY=PUFLIB; VALUE; * to user: insert the complete set of VALUE statements found in Section D; VALUE; RUN;  Below is an example of how to use the SAS formats defined by the PROC FORMAT  |
| If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this:  LIBNAME PUFLIB 'C:\MEPS\SASDATA';  PROC FORMAT LIBRARY=PUFLIB;  VALUE; * to user: insert the complete set of VALUE statements found in Section D;  VALUE;  RUN;   |
| If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this:  LIBNAME PUFLIB 'C:\MEPS\SASDATA';  PROC FORMAT LIBRARY=PUFLIB; VALUE; * to user: insert the complete set of VALUE statements found in Section D; VALUE; RUN;  Below is an example of how to use the SAS formats defined by the PROC FORMAT procedure:  LIBNAME PUFLIB 'C:\MEPS\SASDATA'; OPTIONS FMTSEARCH=(PUFLIB);  PROC FREQ DATA=PUFLIB.H160IF1; TABLES / LIST MISSING;   |
| If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this:  LIBNAME PUFLIB 'C:\MEPS\SASDATA';  PROC FORMAT LIBRARY=PUFLIB; VALUE; * to user: insert the complete set of VALUE statements found in Section D; VALUE; RUN;  Below is an example of how to use the SAS formats defined by the PROC FORMAT procedure:  LIBNAME PUFLIB 'C:\MEPS\SASDATA'; OPTIONS FMTSEARCH=(PUFLIB); PROC FREQ DATA=PUFLIB.H160IF1;   |
| If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this:  LIBNAME PUFLIB 'C:\MEPS\SASDATA';  PROC FORMAT LIBRARY=PUFLIB; VALUE; * to user: insert the complete set of VALUE statements found in Section D; VALUE; RUN;  Below is an example of how to use the SAS formats defined by the PROC FORMAT procedure:  LIBNAME PUFLIB 'C:\MEPS\SASDATA'; OPTIONS FMTSEARCH=(PUFLIB);  PROC FREQ DATA=PUFLIB.H160IF1; TABLES / LIST MISSING; FORMAT varnam1 fmtnam1. Varnam2 fmtnam2; * to user: substitute varnam1 and fmtnam1 with actual variable names and format names;   |
| If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this:  LIBNAME PUFLIB 'C:\MEPS\SASDATA';  PROC FORMAT LIBRARY=PUFLIB; VALUE; * to user: insert the complete set of VALUE statements found in Section D; VALUE; RUN;  Below is an example of how to use the SAS formats defined by the PROC FORMAT procedure:  LIBNAME PUFLIB 'C:\MEPS\SASDATA'; OPTIONS FMTSEARCH=(PUFLIB);  PROC FREQ DATA=PUFLIB.H160IF1; TABLES / LIST MISSING; FORMAT varnam1 fmtnam1. Varnam2 fmtnam2; * to user: substitute varnam1 and fmtnam1 with actual variable names and format names; * Insert the FORMAT statement provided in Section D, if you are using all the variables in the TABLES statement; TITLE 'Frequency Distributions'; RUN;  Here is an explanation of the SAS statements used above.  LIBNAME statement: This tells SAS the location (directory name) of the SAS  |
| If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this:  LIBNAME PUFLIB 'C:\MEPS\SASDATA';  PROC FORMAT LIBRARY=PUFLIB; VALUE; * to user: insert the complete set of VALUE statements found in Section D; VALUE; RUN;  Below is an example of how to use the SAS formats defined by the PROC FORMAT procedure:  LIBNAME PUFLIB 'C:\MEPS\SASDATA'; OPTIONS FMTSEARCH=(PUFLIB);  PROC FREQ DATA=PUFLIB.H160IF1; TABLES / LIST MISSING; FORMAT varnam1 fmtnam1. Varnam2 fmtnam2; * to user: substitute varnam1 and fmtnam1 with actual variable names and format names; * Insert the FORMAT statement provided in Section D, if you are using all the variables in the TABLES statement; TITLE 'Frequency Distributions'; RUN;  Here is an explanation of the SAS statements used above.  |
| If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this:  LIBNAME PUFLIB 'C:\MEPS\SASDATA';  PROC FORMAT LIBRARY=PUFLIB; VALUE; * to user: insert the complete set of VALUE statements found in Section D; VALUE; RUN;  Below is an example of how to use the SAS formats defined by the PROC FORMAT procedure:  LIBNAME PUFLIB 'C:\MEPS\SASDATA'; OPTIONS FMTSEARCH=(PUFLIB);  PROC FREQ DATA=PUFLIB.H160IF1; TABLES / LIST MISSING; FORMAT varnam1 fmtnam1. Varnam2 fmtnam2; * to user: substitute varnam1 and fmtnam1 with actual variable names and format names; * Insert the FORMAT statement provided in Section D, if you are using all the variables in the TABLES statement; TITLE 'Frequency Distributions'; RUN;  Here is an explanation of the SAS statements used above.  LIBNAME statement: This tells SAS the location (directory name) of the SAS format library. Please note that SAS datasets (file name extension is 'SAS7BDAT' for SAS V8 or higher and 'SD2' for SAS V6) and format libraries (file name extension is 'SAS7BDAT' for SAS V8 or higher and 'SD2' for SAS V6) and format libraries (file name extension is 'SAS7BDAT' for SAS V8 or higher and 'SD2' for SAS V8 or higher and 'SC2' for SAS V6) can be stored under the same directory.  OPTIONS FMTSEARCH=: This specifies the SAS format library.   |
| If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this:  LIBNAME PUFLIB 'C:\MEPS\SASDATA';  PROC FORMAT LIBRARY=PUFLIB;  VALUE; * to user: insert the complete set of VALUE statements found in Section D;  VALUE;  RUN;  Below is an example of how to use the SAS formats defined by the PROC FORMAT procedure:  LIBNAME PUFLIB 'C:\MEPS\SASDATA';  OPTIONS FMTSEARCH=(PUFLIB);  PROC FREQ DATA=PUFLIB.H160IF1;  TABLES / LIST MISSING;  FORMAT Varnam1 fmtnam1. Varnam2 fmtnam2; * to user: substitute varnam1 and fmtnam1 with actual variable names and format names;  * Insert the FORMAT statement provided in Section D, if you are using all the variables in the TABLES statement;  TITLE 'Frequency Distributions';  RUN;  Here is an explanation of the SAS statements used above.  LIBNAME statement: This tells SAS the location (directory name) of the SAS format library. Please note that SAS datasets (file name extension is 'SAS7BDAT' for SAS V8 or higher and 'SD2' for SAS V6) and format libraries (file name extension is 'SAS7BCAT' for SAS V8 or higher and 'SC2' for SAS V6) can be stored under the same directory.  |
| If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this:  LIBNAME PUFLIB 'C:\MEPS\SASDATA';  PROC FORMAT LIBRARY=PUFLIB;  VALUE; * to user: insert the complete set of VALUE statements found in Section D;  VALUE;  RUN;  Below is an example of how to use the SAS formats defined by the PROC FORMAT procedure:  LIBNAME PUFLIB 'C:\MEPS\SASDATA';  OPTIONS FMTSEARCH=(PUFLIB);  PROC FORMAT PUFLIB.H160IFI;  TABLES / LIST MISSING;  FORMAT varnaml fmtnaml. Varnam2 fmtnam2;  * to user: substitute varnaml and fmtnam1 with actual variable names and format names;  * Insert the FORMAT statement provided in Section D, if you are using all the variables in the TABLES statement;  TITLE 'Frequency Distributions';  RUN;  Rere is an explanation of the SAS statements used above.  LIBNAME statement: This tells SAS the location (directory name) of the SAS format library. Please note that SAS datasets (file name extension is 'SAS7BDAT' for SAS V8 or higher and 'SC2' for SAS V6) and format libraries (file name extension is 'SAS7BCAT' for SAS V8 or higher and 'SC2' for SAS V6) can be stored under the same directory.  OPTIONS FMTSEARCH=: This specifies the SAS procedure that will make SAS formats according to VALUE statements. Formats will be stored in a file named FORMATS.SAS7BCAT. Please note that the option 'LIBRARY=' can be omitted if the user does not want to create a permanent SAS format library. When simpl 'PROC FORMAT; is used, the formats are defined only for the duration of the batch SAS program or an interactive SAS session.   |
| If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this:  LIBNAME PUFLIB 'C:\MEPS\SASDATA';  PROC FORMAT LIBRARY=PUFLIB;  VALUE; * to user: insert the complete set of VALUE statements found in Section D;  VALUE;  RUN;  Below is an example of how to use the SAS formats defined by the PROC FORMAT procedure:  LIBNAME PUFLIB 'C:\MEPS\SASDATA';  OPTIONS FMTSEARCH=(PUFLIB);  PROC FREQ DATA=PUFLIB.H160IF1;  TABLES / LIST MISSING;  FORMAT varnam1 fmtnam1. Varnam2 fmtnam2;  * to user: substitute varnam1 and fmtnam1 with actual variable names and format names;  * Insert the FORMAT statement provided in Section D, if you are using all the variables in the TABLES statement;  TITLE 'Prequency Distributions';  RUN;  Here is an explanation of the SAS statements used above.  LIBNAME statement: This tells SAS the location (directory name) of the SAS format library. Please note that SAS datasets (file name extension is 'SASTBDAT' for SAS v8 or higher and 'SC2' for SAS v6) can format libraries (file name extension is 'SASTBCAT' for SAS v8 or higher and 'SC2' for SAS v6) can be stored under the same directory.  OPTIONS FMTSEARCH=: This specifies the SAS format library.  PROC FORMAT statement: This identifies the SAS procedure that will make SAS formats according to VALUE statements. Formats will be stored in a file named FORMATS.SASTBCAT. Please note that the option 'LIBRARY=' can be omitted if the user does not want to create a permanent SAS format library. VALUE statements. Tormats will be stored in a file named FORMATS.SASTBCAT. Please note that the option 'LIBRARY=' can be omitted if the user does not want to create a permanent SAS format library when simpl 'PROC FORMAT,' is used, the formats are defined only for the duration of the batch SAS program or an interactive SAS session.  VALUE statement: This gives a) names to formats; and b) descriptive labels for individual values, or range of values. The f                                    |
| If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this:  LIBNAME PUFLIB 'C:\MEPS\SASDATA';  PROC FORMAT LIBRARY=PUFLIB;  VALUE; * to user: insert the complete set of VALUE statements found in Section D;  VALUE;  RUN;  RUN;  Below is an example of how to use the SAS formats defined by the PROC FORMAT procedure:  LIBNAME PUFLIB 'C:\MEPS\SASDATA'; OPTIONS FMTSEARCH=(PUFLIB);  PROC FREQ DATA=PUFLIB.H160IF1; TABLES / LIST MISSING; FORMAT varnam1 fmtnam1. Varnam2 fmtnam2;  * to user: substitute varnam1 and fmtnam1 with actual variable names and format names;  * Insert the FORMAT statement provided in Section D, if you are using all the variables in the TABLES statement; TITLE 'Frequency Distributions'; RUN;  Here is an explanation of the SAS statements used above.  LIBNAME statement: This tells SAS the location (directory name) of the SAS format library. Please note that SAS datasets (file name extension is 'SAS7BDAT' for SAS V8 or higher and 'SD2' for SAS V6) and format libraries (file name extension is 'SAS7BDAT' for SAS V8 or higher and 'SAS V8 or higher and 'SC2' for SAS V6) can be stored under the same directory.  OPTIONS FMTSEARCH=: This specifies the SAS format library.  PROC FORMAT statement: This identifies the SAS procedure that will make SAS formats according to VALUE statements. Formats will be stored in a file named FORMATS.SAS7BCAT. Please note that the option 'LIBRARY=' can be omitted if the user does not want to create a permanent SAS format library.' when simpl'PROC FORMAT; is used, the formats are defined only for the duration of the batch SAS program or an interactive SAS session.  VALUE statement: This gives a) names to formats; and b) descriptive labels for individual values, or range of values. The format names can then be invoked   |
| If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this:  LIBNAME PUFLIB 'C:\MEPS\SASDATA';  PROC FORMAT LIBRARY=PUFLIB; VALUE; * to user: insert the complete set of VALUE statements found in Section D; VALUE; RUN;  Below is an example of how to use the SAS formats defined by the PROC FORMAT procedure:  LIBNAME PUFLIB 'C:\MEPS\SASDATA'; OPTIONS FWITSEARCH=(PUFLIB);  PROC FREQ DATA-PUFLIB.H160IF1; TABLES / LIST MISSING; FORMAT varnam1 fintnam1. Varnam2 fmtnam2; * to user: substitute varnam1 and fmtnam1 with actual variable names and format names; * Insert the FORMAT statement provided in Section D, if you are using all the variables in the TABLES statement; TITLE 'Frequency Distributions'; RUN;  Here is an explanation of the SAS statements used above.  LIBNAME statement: This tells SAS the location (directory name) of the SAS format library. Please note that SAS datasets (file name extension is 'SASTBDAT' for SAS V8 or higher and 'SD2' for SAS V6) and format libraries (file name extension is 'SASTBDAT' for SAS V8 or higher and 'SD2' for SAS V6) can be stored under the same directory.  OPTIONS FMTSEARCH=: This specifies the SAS procedure that will make SAS formats according to VALUE statements. Formats will be stored in a file named FORMATS. SASTBCAT. Please note that the Option 'LIBRARY=' can be omitted if the user does not want to create a permanent SAS format library. When simpl PROC FORMAT; is used, the formats are defined only for the duration of the batch SAS program or an interactive SAS session.  VALUE statement: This gives a) names to formats; and b) descriptive labels for individual values, or range of values. The format names can then be invoked using a FORMAT statement is used. The input SAS dataset is specified   |
| If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this:  LIBNAME PUFLIB 'C:\MEPS\SASDATA';  PROC FORMAT LIBRARY=PUFLIB;  VALUE, * to user: insert the complete set of VALUE statements found in Section D;  VALUE, RUN;  Below is an example of how to use the SAS formats defined by the PROC FORMAT procedure:  LIBNAME PUFLIE 'C:\MEPS\SASDATA';  OPTIONS FMTSEARCH=(PUFLIB);  PROC FREQ DATA=PUFLIB.HIGOIF1;  TABLES / LIST MISSING;  FORMAT Varnam1 fmtnam1. Varnam2 fmtnam2;  * to user: substitute varnam1 and fmtnam1 with actual variable names and format names;  * Insert the FORMAT statement provided in Section D, if you are using all the variables in the TABLES statement;  TITLE 'Prequency Distributions';  RUN;  Here is an explanation of the SAS statements used above.  LIBNAME statement: This tells SAS the location (directory name) of the SAS format library. Please note that SAS datasets (file name extension is 'SASTBOAT' for SAS V8 or higher and 'SD2' for SAS V6) and format libraries (file name extension is 'SASTBOAT' for SAS V8 or higher and 'SC2' for SAS V6) can be stored under the same directory.  OPTIONS FMTSEARCH=: This specifies the SAS format library.  PROC FORMAT statement: This identifies the SAS procedure that will make SAS formats according to VALUE statements. Formats will be stored in a file named of TROMATS.SASTBCAT 'Please note that the option 'LIBRANE' can be omitted if the user does not want to create a permanent SAS format library. When simpl 'PROC FORMAT statement: This identifies the SAS procedure that will make SAS formats according to VALUE statements. Formats will be stored in a file named extension is case of the stored of the st                                 |
| If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this:  LIENAME PUPLIB 'C:\MEPS\SASDATA';  PROC FORMAT LIERARY=PUPLIE;  VALUE; * to user: insert the complete set of VALUE statements found in Soction D;  VALUE;  VALUE;  RUN;  Below is an example of how to use the SAS formats defined by the PROC FORMAT procedure:  LIENAME PUPLIB 'C:\MEPS\SASDATA';  OPTIONS FWISEARCH=(PUPLIB);  PROC FREQ DATA=PUPLIB.H.B.GIFT;  TABLES / LIST MISSING;  FORMAT varnaml fmtnaml. Varnam2 fmtnam2,  * to user: substitute varnaml and fmtnami with actual variable names and format names;  * Insert the FORMAT statement;  TITLE 'Frequency Distributions';  RUN;  Here is an explanation of the SAS statements used above.  LIBNAME statement: This tells SAS the location (directory name) of the SAS format library. Please note that SAS datasets (file name extension is 'SASTSMAT' for SAS V8 or higher and 'SU2' for SAS V6) and format libraries (file name extension is 'SASTSMAT' for SAS V8 or higher and 'SU2' for SAS V6) and format libraries (file name extension is 'SASTSMAT' for SAS V8 or higher and 'SU2' for SAS V6) and format libraries (file name extension is 'SASTSMAT' for SAS V8 or higher and 'SU2' for SAS V6) and format libraries (file name extension is 'SASTSMAT' for SAS V8 or higher and 'SU2' for SAS V6) and format library.  PROC FORMAT statement: This identifies the SAS format library.  PROC FORMAT; is used, the formats are defined only for the duration of the batch SAS program or an interactive SAS session.  VALUE statement: This gives a) names to formats; and b) descriptive labels for individual values, or range of values. The format names can then be invoked using a FORMAT statement if desired.  PROC FREQ statement: This identifies the SAS procedure that generates frequency distributions of variables specified in the TABLES statement, formatted in the 'DATA=' option.  FORMAT statement: This associates existing formats with                              |
| If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this:  LIBNAME PUPLIE 'C:\MEPS\SASDATA';  PROC FORMAT LIBRARY=PUPLIE;  VALUE; * to user: insort the complete set of VALUE statements found in Saction D;  VALUE; * to user: insort the complete set of VALUE statements found in Saction D;  VALUE; * many control of the SAS formats defined by the PROC FORMAT procedure:  LIBNAME PUPLIE 'C:\MEPS\SASDATA';  OPTIONS PRISEARCH=(FUPLIE);  PROC FORMAT DISTANCE (FUPLIE);  PROC FORMAT STATEMENT OF THE MISSING:  FORMAT MANUELLE, HIGOTI;  TABLES / LIST MISSING:  FORMAT Anames;  FORMAT Anames;  FORMAT STATEMENT OF THE TABLES STATEMENT OF THE ACTUAL VARIABLE AND AS THE MISSING:  all the variables in the TABLES STATEMENT;  RUN;  Here is an explanation of the SAS statements used above.  LIBNAME Statement: This tells SAS the location (directory name) of the SAS format library. Please note that SAS datasets (file name extension is 'SASTBOAT' for SAS VG or higher and 'SC2' for SAS VG) can be stored under the same directory.  OPTIONS FMISERACH=: This specifies the SAS format library.  PROC FORMAT Statement: This identifies the SAS procedure that will make SAS formats according to VALUE statements. Formats will be stored in a file name fORMATS.SASTBOAT, Please note that the option 'IJBRARY' can be omitted if the user does not want to create a permanent SAS format library. When simpl PROC FORMAT statement: This gives a) names to formats; and b) descriptive labels for individual values, or range of values. The formats and b) descriptive labels for individual values, or range of values. The format names can then be invoked using a FORMAT statement: This identifies the SAS procedure that generates frequency distributions of variables specified in the TABLES statement;  FORC FROS patatement: This descriptive SAS session.  VALUE statement: This statement is used. The input SAS dataset is specified in the 'DATAP' option.  FORMA                           |
| If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this:  LIBNAME PUFLIE 'C:\MEPS\SASDATA';  PROC FORMAT LIBRARY=PUFLIE;  VALUE; * to user: insert the complete set of VALUE statements found in Section D;  VALUE; * to user: insert the complete set of VALUE statements found in Section D;  VALUE; * to user: statement;  RUR;  PROC FORMAT BUFLIE 'C:\MEPS\SASDATA';  OPTIONS FHISEARCH=(PUFLIE);  PROC FRED DATA=PUFLIE, HisOFIP;  TABLES / LIST MISSING;  FORMAT varname funthamal. varnama? fmtnama?;  * to user: substitute varnaml and fmtnaml with actual variable names and format names;  * Insert the FORMAT statement provided in Section D, if you are using all the variables in the TABLES statement;  TITLE 'rrequency Distributions;  RUR;  Here is an explanation of the SAS statements used above.  LIBNAME statement: This tells SAS the location (directory name) of the SAS format library. Please note that SAS datasets (file name extension is 'SASS'BDAT' for SAS '80 rhigher and 'SD2' for SAS '80 and format libraries (file name extension is 'SASS'BDAT' for SAS '80 rhigher and 'SD2' for SAS '80 and format library. Please note that Greetory.  OPTIONS PHISEARCH=: This specifies the SAS procedure that will make SAS formats according to VALUE statements. Formats will be stored in a file named FORMATS. SAS'BCAT' please note that the option 'LIERARY=' can be omitted if the user does not want to create a permanent SAS format library. When simpl 'PROC FORMAT' is used, the formats are defined only for the duration of the batch SAS program or an interactive SAS secsion.  VALUE statement: This gives a) names to formats, and b) descriptive labels for individual values, or range of values. The format same can then be invoked using a FORMAT statement if desired.  PROC FREO statement: This associates existing formats with variables. When using this statement: This associates existing formats with variables. When using th                                     |
| If a user wants to use formats for the SAS variables, a SAS format library must first be created. Below is a SAS program that will accomplish this:  LIBRAME PULLIB 'C:\MEPS\SASDATA';  PROC FORMAT LIBRARY=PUPLIB; VALUE; * to user: insert the complete set of VALUE statements found in Section D; VALUE;  PROC FORMAT LIBRARY=PUPLIB; VALUE;  Below is an example of how to use the SAS formats defined by the PROC FORMAT procedure:  LIBRAME PUPLIB 'C:\MEPS\SASDATA'; OPTIONS PMYSSARGE=(PUPLIB);  PROC FREQ DATA=PUPLIB.H1601F1; TASLES / LIST MISSING; FORMAT varnami fantami. Varnam2 fmtnam2; * to user: substitute varnam1 and fmtnam1 with actual variable names and format names. set the YOMANT statement; FILLE 'Prequency Distributions; * to user: substitute varnam1 and fmtnam2 with actual variable names and format names. set the YOMANT statement; FILLE 'Prequency Distributions; * NUN;  Were is an explanation of the SAS statements used above.  LIBNAME statement: This tells SAS the location (directory name) of the SAS format library. Please note that SAS datasets (file name extension is 'SASTEDAT' for SAS 'V8 or higher and 'SD2' for SAS 'V6) and format libraries (file name extension is 'SASTEDAT' for SAS 'V8 or higher and 'SD2' for SAS 'V6) can be stored under the same directory.  OPTIONS FWISEARCH=: This specifies the SAS procedure that will make SAS formats according to VALUE statements. Formats will be stored in a file name FORMATS.SASTECAT. Please note that the option 'LIBRARY=' can be omitted if the user does not want to create a permanent SAS format library.  PROC FORMAT statement: This identifies the SAS procedure that will make SAS formats according to VALUE statements. Formats will be stored in a file name format a specified in the 'DATA and any statement is used. The format names can then be invoked using a FORMAT statement: This identifies the SAS procedure that generates frequency distributions of variables specified in the 'DATA and the permanent SAS dataset is specified   |
| If a user wants to use formats for the SAS variables, a SAS format library must first be czeated. Below is a SAS program that will accomplish this:  LIBNAME PUFLIB 'C:\MEDRA\SADATA';  PROC FORMAT LIBRARY-PUFLIB; VALUE; * to user: insert the complete set of VALUE statements found in Section D; VALUE;  VALUE;  **NOW;  Below is an example of how to use the SAS formats defined by the PROC FORMAT procedure;  LIBNAME PUFLIB 'C:\MEDRA\SADATA'; OPTIONS PYMSTARCH=(PUFLIB);  PROC FREE DATA-PUFLIB.H160TF1; TABLES / LIST HISSING; POURMAT varnam! fathman! . varnam2 fmtnam2;  ** to user: substitute varnam1 and fmtnam1 with actual variable names and format names.  ** Insert the PORMAT statement provided in Section D, if you are using all the variables in the TABLES statement; TITLE Frequency Distributions;  **RUN;  Bere is an explanation of the SAS statements used above.  LIBHAME statement: This tells SAS the location (directory name) of the SAS format library. Please note that SAS datasets (file name extension is 'SASTMAN' for SAS v8 or higher and 'SD2' for SAS v6) and format libraries (file name extension is 'SASTMAN' for SAS v8 or higher and 'SD2' for SAS v6) and format library.  PROC FORMAT statement: This identifies the SAS format library.  PROC FORMAT statement: This identifies the SAS format library.  PROC FORMAT; is used, the formats are defined only for the duration of the batch SAS proyer does not want to create a permanent SAS format library. Who simpl 'PROC FORMAT; is used, the formats are defined only for the duration of the batch SAS proyer on or interactive SAS session.  VALUE statement: This identifies the SAS procedure that will make SAS format socording to VALUE statements. Pormats will be stored in a file named FORMATS. SASTBCAT. Please note that the option 'LIBRARY' can be omitted  If the user formats interactive SAS session.  VALUE statement: This identifies the SAS procedure that generates frequency distributions of variables specified in the TABLES statement, formatte                                     |
| If a user wants to use formats for the SAS variables, a SAS format library must first be created. Relow is a SAS program that will accomplish this:  LIBNAME FURLIB 'C:\MEPS\SASDATA';  PROC FORMAT LIBRARY-PUPLIB; VALUE; * to user: insert the complete set of value statements found in Section D; VALUE; * to user: insert the complete set of value statements found in Section D; VALUE; * RUN; RUN;  Below is an example of how to use the SAS formats defined by the PROC FORMAT procedure:  LIBRAME FURLIB 'C:\MEPS\SASDATA'; OPTIONS FYMESHAGE[EVUTLB); PROC FREO RAWA-PUPLIB RIGGLE); FARMES / LISE WISSING; TORNAT varnaml fastnaml varnam2 [mtnam2,;  * to user: substitute varnam1 and fmtnam1 with actual variable names and format names;  * Insert the FORMAT statement provided in Section D, if you are using all the variables in the TABLES statement; TITLE 'Frequency Distributions; RUN; RUN; RUN; Rer is an explanation of the SAS statements used above.  LIBRANE Statement: This talls SAS the location (directory name) of the SAS format library. Please note that SAS datamets (file name extension is 'SASTBANE' for SAS VS on higher and 'SC2' for SAS VS) can be stored under the same directory.  OPTIONS PHISEARCH: This specifies the SAS format library.  PROC FORMAT statement: This identifies the SAS format library.  PROC FORMAT Statement: This identifies the SAS procedure that will make SAS formats according to VALUE statements. Formats will be stored in a file name format according to VALUE statements. Formats will be stored in a file name back of the SAS STORMS of the SAS STORMS of the SAS FORMAT SASSASCAT. Flease note that the option 'LIBRARY' can be omitted if the user does not want to create a permanent SAS format library. When simpl FROK FORMAY is used, the formats are defined only for the duration of the batch SAS program or an interactive SAS session.  PROC FORMAT statement: This identifies the SAS procedure that generates frequency distributions of variables specified in the TABLES statement,                                      |
| If a user wants to use formats for the SAS variables, a SAS format library must first be created. Relow is a SAS program that will accomplish this:  LIBNAME PUPLIB 'C:\MEMS\ASADATA';  PROC FORMAT LIBRARY-PUPLID;  VATUE;  **to user: insert the complete set of VALUE statements found in section D;  VATUE;  NUR;  Relow is an example of how to use the SAS formats defined by the PROC FORMAT procedure:  LIBNAME PUPLIB 'C:\MEMS\ASADATA';  OPPTONS FMYSRARACH=(DUPLIB);  PROC FRED DATA-PUPLIB INSERING;  PROME THE DATA-PUPLIB HOFF!;  TABLES / LISE MYSSING;  PROME THE DATA-PUPLIB HOFF!;  TABLES / LISE MYSSING;  PROME THE DATA-PUPLIB HOFF!;  TABLES / LISE MYSSING;  PROME THE ACT THE TABLES STATEMENT;  TOTAL Proquency Distributions;  ELEVANCE STATEMENT;  LIBRAME STATEMENT PROGRAMS STATEMENT;  TOTAL Proquency Distributions;  RIN;  But is an explanation of the SAS Statements used above.  LIBRAME Statement This tolks SAS the location (directory near) of the SAS statement is an explanation of this program that SAS Statements (file name extension is 'SASTPORT' to SAS VO or higher and 'SC2' for SAS VO') can be stored under the same directory.  OPTIONS PHYSEARCH=: this specifies the SAS format library.  PROC POWNAT statement: This tolis statements remeats will be stored in a file name POWAMIS SASTSCAT. Please note that the option 'LISEARY* can be omitted if the user does not vant to create a permanent SAS format library.  PROC POWNAT statement: this identifies the SAS procedure that will make SAS formats according to VALUE statements. Promess will be stored in a file name POWAMIS SASTSCAT. Please note that the option 'LISEARY* can be omitted if the user does not vant to create a permanent SAS format library. When simple 'PROC POWAMI' is used, the formats are defined only for the duration of the batch SAS program or an interactive SAS session.  NALUE statement: This identifies the SAS procedure that generates and the statement is a power of analyses that you are doing. It is recommended that you  |
| If a user wants to use formats for the SAS variables, a SAS format library must first be created. Relow is a SAS program that will accomplish this:  LIDRAME FUFLIS 'C:\MEPS\SASDATA';  PROC FORMAT LIRRAW=0PUFLIS;  VALUE; 'to user; insert the complete set of VALUE statements found in Section D;  VALUE;  BRH;  BRI;  BROW is an example of how to use the SAS formats defined by the PROC FORMAT procedure:  LIMRAME FUFLIS 'C:\MEPS\SASDATA';  OPTIONS FWISEARCH=(RUPLIS);  PROC FRED DATA-POFLID.HIGOFF; TABLES / LIST XISSING;  FORMAT varnami fathonal. Varnam2 fathoma2;  **TOTAL 'FORQUERY Distributions;  RING;                             |
| If a user wants to use formets for the SAS variables, a SAS formet library must first be created. Delow is a SAS program that will accomplish this:  LIDRAME FULIS 'C:\MEPS\SASDATA';  PROC TORNAT LIBRARY-FULID;  VALUE; * to user: insert the complete set of VALUE statements found in Saction D;  VALUE;  DELOW is an example of how to use the SAS formats defined by the DMCC FORMAT processor.  LIBRANE BURLIG 'C:\MEPS\SASDATA';  OFFICES PRISEARCH-(C:\MEPS\SASDATA';  OFFICES PRISEARCH-(C:\MEPS\SASDATA');  PROC FREQ OATA-FULID.HISSING;  FORMAT variable into TASING Statement;  TITLE 'Frequency District versand and feminal with actual variable names and format names;  * Insert the FORMAT statement provided in Section D, if you are using all the variables in the TASING Statement;  TITLE 'Frequency Districtions';  NOTH the variables in the TASING Statements used above.  LIBRANE statement: This tells SAS the location (directory name) of the SAS format library. Please note that SAS datasets (file name extension is TASING STATE, TASING STATE, THE STATE OF TASING STATE STATE THE STATE OF TASING STATE ST              |
| If a user wants to use formats for the SAS variables, a SAS format library must first the created. Below is a SAS program that will accomplish this:  LINEMADE PUTLE 'CTIMES'ASSORTA';  PROC FORMAT LIBRARYSHUPLIS;  VALUE; * to user: insert the complete set of VALUE statements found in Section D;  VALUE;  ROW,;  ROW,;  ROW,;  Below is an example of how to use the SAS formats defined by the PROC FORMAT procedure:  LIDRANE PUTLED 'CTIMES'ASSORTA';  POPPIONS PRESHARGE-(PUBLIS);  PROC FROM ORD-RUPLED HISSING;  PORMAT variant fantami. Variant2 fantama?;  * to user: substitute variant and intends with actual variable names and format names;  * the programmy binary variant fantami. Variant2 fantama?;  * to user: substitute variant and intends with actual variable names and format names;  * the programmy binary variant fantamin. Variant2 fantamin;  * To the ser: substitute variant and intends with actual variable names and format names;  * the programmy binary bushions;  * TO THE VALUE STATE THE PORMAT Statement provided in Section D, if you are using all the variables in the TAMINS statement provided in Section D, if you are using all the variables in the TAMINS statement in the variables in the TAMINS statement in the variables in the TAMINS statement in the section of the SAS statements used above.  LIDRANE statement: This lells SAS the location (directory name) of the SAS (fine and exclusion is 'SASTORAT' for SAS V0 or higher and 'SC2' for SAS V0) and format libraries of the same directory.  **PORTIONS TRYERACCE* This specifies the SAS format library.  **PORTIONS TRYERACCE* This specifies the SAS format library.  **PORTIONS TRYERACCE* This specifies the SAS format library. when simplify the provider according to VALUE statements. Formats will be stored in a file name of the section of the same directory and the same statement in th   |
| If a user wants to use formats for the SAS variables, a SAS format library most first be created. Below is a SAS program that will accomplish this:  LEDBARE PORTED ("A VERNEW SASSARA";  PROC FORMAT LIGRARY-PUPLID;  VALUE; * to user; insert the complete set of VALUE statements found in SALUE;  **VALUE; * to user; insert the complete set of VALUE statements found in SALUE;  **VALUE; * To user; insert the complete set of VALUE statements found in SALUE;  **VALUE; * TO USER SALUE                               |
| If a user wants to use formets for the SAS veriables, a SAS format library must first be created. Below is a SAS program that will accomplish this:  IIOMARE SUPLID ("AURENCANDATA";  PROC FORMAT LIBRARY-PUFLID;  VALUE; to user; insert the complete set of VALUE statements found in Salue;  VALUE;  ROW;  Notice is an example of how to use the SAS formets defined by the PROC PURMAY PROCESS.  PROBACE / LIGH WIGHING PROBACE;  **COMPTIONS PRESEASEL*(PUPLIS);  PROC FREED ALTA-SUPLIS HIGHING;  **CO user; substitute various and finished with actual variable names and forman names;  **CO user; substitute various and finished with actual variable names and forman in the same of the SAS statement;  BING;  **SUPERING THE TRACES statement;  **CO INTERPRETATION PROBACTION;  **ROW;  **SUPERING THE TRACES STATEMENT;  **CO INTERPRETATION STATEMENT STATEMENT;  **SUPERING THE ALTA-SUPLIS HIGHER SAS FORMAN STATEMENT;  **SUPLIS THE ALTA-SUPLIS HIGHER SAS FORMAN STATEMENT;  **CO INTERPRETATION STATEMENT;  **SUPLIS THE ALTA-SUPLIS STATEMENT;  **SUPL                             |
| If a user wants to use formats for the SAS variables, a SAS format library mast first be created. Select is a SAS program that will accomplish this?  ILIBRADY PURLING 'CHEMPSARABURA';  FROG FORMAT LIBRADY-PURLING;  FROG FORMAT LIBRADY-PURLING;  FROM FORMAT LIBRADY-PURLING;  FROM FORMAT LIBRADY-PURLING;  VALUE;  **Note of the second of the to use the SAS formats defined by the FROC FORMAT procedure.  **PROF FORMAT LIBRADY-PURLING;  **PROF PURL DECEMPERATED FOR A SAS FORMATS AND A SAS FORMAT     |
| The state works to use formats for the SAS variables, a SAS formet library must first be created. Delow is a SAS program that will accomplish this?  LESHAND FULL ("CHAUSARDIAN")  PROC PORMAT LIBRARY-PULLIN;  VALUE  **SACCITY 19 " " to later Insert the Complete set of VALUE statements Cound to Section 19; " " to later Insert the Complete set of VALUE statements Cound to Section 19; " " TO Later Insert the Complete set of VALUE statements Cound to Section 19; " " TO Later Insert Type To Complete Section 19; " " " " " " " " " " " " " " " " " " "   |
| Tis user wonts to use formats for the SAS variables, a SAS format library mant first he constant, selbs is a SAS program that will accomplish this:  LIBRARY MUPULE 'CLAMPERSASSAYA';  PROC FORMAT LIBRARY AND THE SAS ASSAYA';  PROC FORMAT LIBRARY HOPERS,  VARIAN, 1 to users insort the complish not of VARIAN statements found in Section D;  VARIAN, 2 to users insort the complish not of VARIAN statements found in Section D;  PARTY OF THE SAS ASSAYA';  PROC 18 OR DATA-PULLAN HORSING;  **Instant the 1054Nt; Variand fetnam2;, 7  PROC 18 OR DATA-PULLAN HORSING;  **Instant the 1054Nt; Variand fetnam2;, 7  PROC 18 OR DATA-PULLAN HORSING;  **Instant the 1054Nt; Variand fetnam2;, 7  **PROC 18 OR DATA-PULLAN HORSING;  **PROC 18 OR DATA-PULLAN HORSING;  **Instant the 1054Nt; Variand fetnam2;, 7  **Instant statement; This talls 885 the location (directory rame) of the 885 format library. Please note that the 1054Nt; Variand fetnam3;, 8  **PROC 1958Nt; Variand fetnam3;, 8  **PROC 1958Nt; Please note that the option (TriBanys) can be entitled for the 1054Nt; Variand fetnam3;, 8  **PROC 1958Nt; Please note that the option (TriBanys) can be entitled for the 1054Nt; Variand fetnam3;, 8  **PROC 1958Nt; Please note that the option (TriBanys) can be entitled for the 1054Nt; Variand fetnam3;, 8  **PROC 1958Nt; Please note that the option (TriBanys) can be entitled for the 1054Nt; Variand fetnam3;, 8  **PROC 1958Nt; Plant (TriBanys) can be entitled for the 1054Nt  |
| If a user weath to use formate for the SAS variables, a SAS format library must first be created. Sciencia a ship program that will accomplish this:  LINEAR FUNDAM CLAREAGE PURLER;  PROC FORMAT LIBRACE TO USER'S INSERT THE COMPLETE STATEMENT TO USE TO US |
| If a user wants to use formats for the SAS variables, a SAS format library main firm to created, helds in a shall program that will accomplish this:  LIBRADE PULLD: "C'ANDERSASSANIA";  proof PURMAT LIBRADE THE THE THE THE THE THE THE THE THE TH   |
| If a user wants to use formats for the SAS variables, a SAS format library mast lists be created, below is add company that will accomplish this library library that below is add company that will accomplish this library. FROM FORMAT LIBRARY PRINTERS, and the complete set of VANDE statements found to see that the complete set of VANDE statements found to see that the complete set of VANDE statements found to see that the complete set of VANDE statements found to see that the complete set of VANDE CAMERO CAME |

2 = '2 OPAT'
3 = '3 EROM'
4 = '4 STAZ'
5 = '5 DVIS'
6 = '6 OMED'
7 = '7 HVIS'
8 = '8 PMED'

VALUE \$EVNTIDX

VALUE PANEL

'0' < - HIGH = 'VALID ID'

17 = '17 PANEL 17' 18 = '18 PANEL 18'