Optimal Data Analysis
2013, Vol. 2, Release 1 (September 27, 2013), 2-6

2155-0182/11/$3.00

How to Create an ASCII Input Data File for UniODA and CTA Software

Fred B. Bryant & Patrick R. Harrison
Loyola University Chicago

UniODA and CTA software require an ASCII (unformatted text) file as input data. Arguably the most difficult task an operator faces in conducting analyses is converting the original data file from (a) whatever software package was used to enter the data, into (b) an ASCII file for analysis. This article first highlights critical issues concerning missing data, variable labels, and variable types that users must address in order to convert their data into an ASCII file for analysis using ODA software. Specific steps needed to convert a data set from its original file-type into a space-delimited ASCII file are then discussed. The process of converting data into ASCII files for use as input data is illustrated for three leading statistical software packages: SPSS, SAS, and STATISTICA.

ODA-based software such as UniODA and CTA requires only a few easy-to-understand control commands to conduct powerful, accurate non-linear modeling. Ironically, given the simplicity of ODA software syntax, the most difficult task for users to complete in conducting analyses is often the creation of an ASCII data file for ODA software to analyze. But, with a little forethought and attention to detail, this critical task is simple and straightforward.

Researchers often use statistics software such as the Statistical Package for the Social Sciences (SPSS), Statistical Analysis System (SAS), or STATISTICA, for example, to enter and process raw data. In contrast, ODA software requires a delimited ASCII file (i.e., an unformatted text file) as input data. Typically, spaces, tabs or commas are used to separate the data entries in an ASCII file. These delimiters enable ODA software to read input data in free form without requiring operators to specify formatting, making it easier to implement analysis. This paper explains how to create ASCII files which use spaces as delimiters separating data entries.

Initial Issues to Resolve

Before converting data from the original file type into an ASCII file, three basic issues must be resolved: (1) the handling of missing data; (2) the creation of variable labels having usable format; and (3) the transformation of alphabetic string variables into a quantified form which CTA can analyze.

Missing data. An important point to keep in mind is the fact that ODA software will not treat a blank space in an ASCII data file as a missing value, but instead will skip over a blank

2155-0182/11/$3.00

space and use the next numeric value in the data set to stand in for the missing value. Therefore, *before* converting their data into an ASCII file for ODA analysis, users must replace any system-missing "blank" values (e.g., a "." in SPSS or SAS) with a numeric value designating a missing response. This conversion can easily be accomplished by using the original statistical software package to recode each variable so as to replace whatever values were originally used to indicate missing data with a chosen, global missing-value indicator (e.g., -999). Note, however, that this task must be done *before* one converts the original data set into an ASCII file.

Observed values are often missing for at least some cases on one or more variables in a data set. Researchers typically use blank spaces to indicate missing values in the original data file or use one or more specific numeric values (e.g., -9, 99, 999) to designate missing responses for different variables. Before creating an ASCII file, users must first convert the value(s) which are being used to represent missing values for each variable—such as blank spaces or numeric values—into a *single numeric value* to be used to designate *all missing values* for every variable in the data set.

Yarnold and Soltysik[1] emphasized the importance of specifying missing values when creating ASCII files for analysis by ODA: "A very important point that one cannot overlook is that *all system missing data must be changed to a specified missing numeric value* prior to analysis via ODA" (p. 55). A popular choice for a universal missing-value indicator is -999. Of course, using -999 as a marker for missing values assumes that this is not a valid response for any variables included in the data set. If the value -999 is a valid response for any variable in the data set, then a value which is not a valid response should be selected instead.

*Saving variable labels in usable format***.** In UniODA and CTA software, variable labels may be no longer than eight characters, so users of general-purpose statistical software, such as SPSS, SAS or STATISTICA, should ensure that the variable names consist of no more than eight characters before exporting their data set for ODA. Otherwise, ODA software will be unable to read the names of variables having more than eight characters and will produce an error message. Alternatively, users can export variable names longer than eight characters, and use a text file editor to truncate variable names to a maximum of eight characters in the exported ASCII file. However, changing variable names to a maximum of eight characters in the original source data file makes it easier to verify the accuracy of the exported ASCII data file, when comparing descriptive statistics from the original and exported data sets.

*Transforming alphabetic variables into numeric form***.** Some variables in the original data set may consist of alphabetic or "string" values, rather than numbers. Examples of such alphabetic variables are gender (e.g., "male" or "female"), religious affiliation (e.g., "Catholic," "Protestant," "Jewish," "Buddhist," "Muslim," or "none"), or ethnicity (e.g., "White," "Black," "Hispanic," "Asian," or "Other"). To analyze such string variables in ODA, users must first recode each alphabetic value of the variable into a numeric value (e.g., "female"=0, "male"=1; "White"=1, "Black"=2, "Hispanic"=3, "Asian"= 4, "Other"=5). After converting all alphabetic values to numeric values, users should save the data file, carefully noting which variable is the class variable, which attributes are ordered, and which attributes are categorical (string variables typically reflect the latter). *The ASCII data file to be analyzed by UniODA and CTA software should contain only numeric values, delimited by spaces*.

Additional considerations. To streamline the analysis as much as possible, it is recommended that users delete any unnecessary variables from the original data file before exporting the data as an ASCII file. Thus, users should eliminate any variable which is neither a class variable nor an attribute in the analysis (e.g., ID number). Excluding unused variables will make the ASCII data file as small as possible and will minimize the time required to obtain final CTA results. Alternatively, one can choose to export only a subset of the variables from the full data set when constructing an ASCII data set.

For the UniODA and CTA programs to access the data file, users should assign the exported ASCII file a name that is no more than 8 characters, followed by a dot and 3 characters (e.g., CTA_RUN1.DAT). Finally, if applicable, users should cut and paste the variable labels from the first line of the exported ASCII data file into a separate ASCII file, to serve as part of the VARIABLES control command in the syntax file for the UniODA or CTA programs.

Examples of Syntax Files for Exporting a Source Data File as an ASCII Data File

Driven by both pull-down menus and syntax, ***SPSS*** is perhaps the most commonly used statistical program in academia. Imagine an SPSS data file (ODAdata.sav) containing 20 variables, 12 of which are to be exported into a space-delimited ASCII data file (ASCIIdat.dat) for analysis by ODA software.

After opening *ODAdata.sav* in SPSS, the SAVE TRANSLATE command may be used to convert the active SPSS data file into a space-delimited ASCII data file, as follows:

```
SAVE TRANSLATE OUTFILE='C:\Documents
and Settings\localuser\Desktop\ASCIIdat.dat'
  /TYPE=CSV
  /MAP
  /REPLACE
  /FIELDNAMES
  /TEXTOPTIONS DELIMITER=' '
  /KEEP=v1 v2 v3 v4 v5 v6 v7
         v8 v9 v10 v11 v12.
```

Here, the subcommand: *OUTFILE='C:\Documents and Settings\localuser\Desktop\ASCIIdat.dat'* is used to instruct SPSS to save the exported ASCII file (which we have named ASCIIdat.dat) to the Windows desktop. Users should alter this subcommand to specify the correct path to the folder on their hard drive where they wish to save the ASCII file.

The */TYPE=CSV* subcommand specifies that the exported data file will be in text-file (ASCII) format.

The */MAP* subcommand displays in the SPSS output a list of the variables and the number of cases exported in the ASCII data file.

The */REPLACE* subcommand gives SPSS permission to overwrite an existing ASCII file of the same name. Because the default is not to overwrite an existing ASCII file, SAVE TRANSLATE will not overwrite an existing file without an explicit REPLACE subcommand. If users wanted to prevent the possibility of overwriting an existing ASCII file, then they could omit the */REPLACE* subcommand.

The */FIELDNAMES* subcommand is used to instruct SPSS to write variable names separated by a delimiter (see below) in the first row of the ASCII data file. As noted earlier, before implementing CTA, users should cut and paste the variable labels from the first line of the exported ASCII data file into a separate ASCII file, to serve as part of the VARIABLES control command in the syntax file for ODA programs.

The */TEXTOPTIONS DELIMITER=' '* subcommand instructs SPSS to employ a blank space (empty column) to delimit or separate variable names and data values in the exported ASCII file.

The */KEEP* subcommand may be used to export to the ASCII data file either: (a) all of the variables in the active SPSS data (by specifying */KEEP=ALL)*; or (b) a subset of the variables in the in the active SPSS data (by specifying */KEEP=<variable names separated by spaces>*, as in the example above). Also, the */KEEP* subcommand may be used to change the order in which the variables appear in the ASCII data file, by using a particular order to list these variables in the */KEEP* subcommand.

Copyright 2013 by Optimal Data Analysis, LLC
2155-0182/11/$3.00

SAS is another popular statistical software program, which is widely used in business analytic settings for operations research, data mining, and predictive modeling. Imagine a SAS data file ODADdat containing 20 variables, 12 of which are to be exported into a space-delimited ASCII data file (ASCIIdat.dat) for analysis by ODA software.

In SAS, the PUT command may be used to convert the active SAS data file into an ASCII data file, as follows:

DATA ODAdata2;
SET ODAdata;
FILE 'C:\Documents and Settings\localuser\Desktop\ASCIIdat.dat';
PUT v1 v2 v3 v4 v5 v6 v7 v8 v9 v10 v11 v12;
RUN;

The DATA command begins the process of data restructuring in SAS. The command *DATA ODAdata2* instructs SAS not to overwrite the active SAS data set (i.e., ODAdata), but to give the new, restructured data set the name ODAdata2 (later changed to ASCIIdat.dat using the *FILE* command).

The *SET* command reads all variables and observations from the SAS input data set.

The *FILE* command renames the restructured data set and writes the contents of the active data set to an external ASCII file.

The *PUT* command outputs the listed variables to the ASCII data specified in the *FILE* command.

The *RUN* command has SAS process the set of commands listed in the syntax file.

Note that the above SAS commands do *not* output the variable names to the first line of the ASCII data file. However, users can use the following commands to enter variable names on line 1 of the ASCII file (followed by space-delimited data):

DATA ODAdata2 (keep= *v1 v2 v3 v4 v5 v6 v7 v8 v9 v10 v11 v12);*
SET ODAdata;
FORMAT v1 v2 v3 v4 v5 v6 v7 v8 v9 v10 v11 v12 10.6;
PROC EXPORT DATA=ODAdata2 OUTFILE ='C:\Documents and Settings\localuser\Desktop \ASCIIdat.dat' DBMS=DLM REPLACE;
RUN;

Note that this method requires that a FORMAT command is used to prevent rounding of exported values, as indicated. The FORMAT command uses the value of “10.6” to tell the SAS program to allot a total of 10 spaces with 6 decimal points for each exported variable.

STATISTICA is another popular data analysis program, commonly used in healthcare, financial services, insurance, and consumer product industries. Imagine a STATISTICA data file named ODAdata.sta containing 20 variables, 12 of which are to be exported into a space-delimited ASCII data file (ASCIIdat.dat) for analysis by ODA software.

With STATISTICA the Windows drop-down menu may be used to export the active data set into a comma-delimited ASCII data file by first opening the data file (*ODAdata.sta*). To export only 12 of the 20 variables in the data set, first delete the variables which will not be exported. Then click on “Save As…” under the File command on the top left-hand side of the main Data Editor screen. In the Save Data As window, click on the down arrow to the right of the “Save as type” box, and select “Text file (*.txt).” Users should then specify the name of the ASCII output file in the “File name box” using the *.txt extension (e.g., *ODAdata.txt*) and the location in which to save this file, and click on the Save command. STATISTICA will respond by warning users that the data file “may contain features that will be lost when saved as text” and asking them if they “want to export the Spreadsheet in this format.” Users should click on “Yes.”

STATISTICA will also display a smaller window giving users the option of specifying the particular “field separator” to use as a delimiter in the ASCII file (users should click on

Copyright 2013 by Optimal Data Analysis, LLC
2155-0182/11/$3.00

"Space"), and writing the variable names separated by the delimiter in the first row of the ASCII data file. Finally, to create the ASCII space-delimited data file, users should click on the Save command. As noted earlier, before implementing ODA software, users should cut and paste the variable labels from the first line of the exported ASCII data file into a separate ASCII file, to serve as part of the VARIABLES command in the syntax file for ODA programs.

Verifying the Accuracy of the ASCII Data File Before Running UniODA or CTA

Before running UniODA or CTA, it is essential first to check the accuracy of the ASCII data file by comparing it to the original data set. To check the accuracy of the exported ASCII file in relation to the original (source) data file, follow the following six steps.

First, run descriptive statistics on the variables in the original data file, using the statistical software employed to enter the raw data originally (e.g., SAS, SPSS, etc.).

Second, replace all blanks and other missing data values with a valid value, such as -999, which will be used to designate missing values in ODA software.

Third, export the original (source) data file into an ASCII format, making sure to export the variable labels on the first line of the ASCII data file. We recommend exporting data as a space-delimited ASCII file.

Fourth, import the exported ASCII file back into the original statistical software (e.g., in SPSS use the "Read Text Data" option beneath the "File" drop-down menu).

Fifth, after importing the exported ASCII data file, use the statistical software to designate values of -999 as missing.

Finally, run descriptive statistics, and compare the results for equivalence with the initial set of descriptive statistics.

If the first and second sets of descriptive statistics are identical, then one can be confident of having accurately exported the original data set into an ASCII format. If the two sets of descriptive statistics based on the original and imported ASCII data do not match perfectly, then pinpoint the source of the problem and repeat the process until perfect correspondence is obtained. Although there are countless mistakes one can make when converting an original data set into a space-delimited ASCII data file, the most common errors include forgetting to change blank data entries to a specific missing numeric value, forgetting to convert alphabetic string variables into numbers, exporting variable names that exceed the maximum of 8 characters, and failing to export all of the variables from the original data set that one wishes to analyze.

References

[1] Yarnold PR, Soltysik RC (2005). *Optimal data analysis: A guidebook with software for Windows*. APA Books, Washington, DC, 2005.

Author Notes

Correspondence should be sent to Fred B. Bryant at: Department of Psychology, Loyola University Chicago, 6525 North Sheridan Road, Chicago, IL, 60626. Email: fbryant@luc.edu.