Paper 036-2019

Order, Order! Four Ways to Reorder Your Variables, Ranked by Elegance and Efficiency

Louise S. Hadden, Abt Associates Inc.

ABSTRACT

SAS® practitioners are frequently required to present variables in an output data file in a particular order, or standards may require variables in a production data file to be in a particular order. This paper and presentation offer several methods for reordering variables in a data file, encompassing both DATA step and procedural methods. Relative efficiency and elegance of the solutions will be discussed.

INTRODUCTION

SAS provides us with numerous methods to control all types of SAS output, including SAS data files, data tables in other formats, and ODS output. This paper focuses solely on output SAS data files (which may then be used to generate SAS data files and other types of output from SAS processes), and specifically on DATA step and PROC SQL methods. This short paper and presentation is suitable for all SAS practitioners at all levels of expertise. Attendees will gain a greater understanding of the processes by which SAS assigns variable attributes, including variable/column order within a data file, and how to obtain information on variable attributes – and in the process, learn how to reorder variables within a SAS data file.

BACKGROUND

Edits made in the process of peer review can sometimes backfire – what seems like an innocuous change can have a ripple effect downstream. In my case, a change to using PROC SQL to reorder variables instead of the RETAIN statement “broke” a long standing routine involving an XML data delivery to a client, and the reviewer could not figure out why. In the process of troubleshooting, this paper was conceived. Pluses and minuses of the methods discussed will illuminate several SAS processes, emphasizing the fact that SAS statements never operate in a vacuum.

KNOW THY DATA

It is always important to understand fully and explore the inputs to SAS-created output. SAS has provided users with a variety of possibilities in terms of determining locations of variables or columns (and other important details comprising metadata). These possibilities include, but are not limited to:

1. The CONTENTS Procedure
2. The DATASETS PROCEDURE
3. SAS Dictionary Tables (used with PROC SQL)
4. SASHELP Views (can be used with PROC SQL and with the SAS DATA step)
5. %SYSFUNC (through MACRO functions)
6. “V” functions (through data file functions)

Any one of the processes listed above can produce the location of a given variable, which is our primary interest here.

```
* THE CONTENTS PROCEDURE;
PROC CONTENTS DATA=DD.HARRYPOTTER_LABELLED
    OUT=LABELLED_CONTENTS
    (KEEP=NAME TYPE LENGTH LABEL FORMAT INFORMAT VARNUM);
RUN;
```

For the purposes of expediency, an abridged CONTENTS procedure listing of a test data file is provided below, showing the column header "#", which is the label for the metadata variable "VARNUM". Note that an output data file is created from the CONTENTS procedure, with a KEEP statement that retains only the variables of interest. Please see the references at the end of the paper for more information on techniques to harness the power of SAS metadata.

Variables in Creation Order

#	Variable	Type	Len	Label
1	combined_name	Char	50	Full Name
2	firstname	Char	25	First Name
7	patronus	Char	25	Patronus
8	animal	Char	25	Pet
9	school	Char	50	School
16	deatheater	Num	8	Binary: Death Eater
17	bloodstatus	Char	10	Blood Status: Halfblood, Muggleborn, Squib, Pureblood

PLAYING COMPUTER

It is important to have an understanding of how SAS handles and defines records and variables in the data file, i.e. through the Program Data Vector (PDV), so as to manipulate data file attributes. The PDV builds data file observations during processing, storing data values in memory, and finally writes out both an output data file and a metadata data file containing variable attributes when a data file compiles.

In the listing above, 5 columns are displayed, including a column for the attribute of interest, VARNUM. The CONTENTS procedure only displays variable attributes that are present in the data file being analyzed, so a contents listing for a data file that does not have any assigned formats, informats or indices will not have columns for those attributes.

If we wish to change the order in which variables are presented in an existing data file, we need to know how SAS determines variable order under various conditions. In the DATA step, variable order is determined by the first mention of the variable as it passes through the PDV. DATA step statements that can set the order of variables include ARRAY, ATTRIB, FORMAT, INFORMAT, LENGTH and RETAIN. These statements MUST precede any SET, MERGE or UPDATE statements so as to accomplish the goal of reordering variables. The order in which variables are mentioned in the statement determines the order of the variables in the corresponding data file. If variables are not listed in the statement, variables that are mentioned will appear in the order that they are mentioned, remaining variables will appear in their original order in the originating data file (or INPUT statements in the case of data files being created).

In addition to DATA step statements, PROC SQL can also be utilized to reorder variables in an existing data file, in a similar fashion to the DATA STEP statements described above. PROC SQL SELECT and SELECT AS statements may be used to order, format, set the length of and rename variables in a new data file created from the existing data file.

Common reasons for wanting to reorder variables in a data file are the result of merging files (order will be common variables first, then variables in the original order of each merged file) or the use of ARRAY, ATTRIB, LENGTH or RETAIN statements in originating data files that "disordered" variables.

METHODS FOR RENAMING

As noted above, options for reordering variables include six data file statements (ARRAY, ATTRIB, FORMAT, INFORMAT, LENGTH and RETAIN) and PROC SQL. Examples of each of these techniques will be explicated below specifically in the context of reordering variables in a data file; DATA STEP statements and PROC SQL are very powerful tools that have utility in many different areas. It is not within the scope of a Coder's Corner paper to discuss all the options and interactions among the different statements; the author urges readers to consult the papers in the references section for more information.

PREPARING FILES FOR TESTING

Each method of reordering data requires the provision of metadata information in the statement. Rather than type in what could be a very extensive variable attribute list, SAS metadata can be used to build the statements. Some methods (RETAIN, FORMAT, INFORMAT, PROC SQL) require an ordered list of variables as input (note that additional metadata information can be used, but is not strictly necessary.) Others (ATTRIB, ARRAY, LENGTH) require additional information, such as the length of a variable, variable label, and/or type of variable. SAS metadata can be used to construct the statements in a variety of ways. Two methods of harnessing SAS metadata to build data-driven code follow. The precursor to these steps is the use of the LABELLED_CONTENTS data file created above, and sorted in various ways into different output data files.

```
* TEST FILE PRECURSOR STEPS: SORT OUT DATA SETS IN SPECIFIC ORDERS;
* VARIABLE NAME (NAME) IS KEY;
* POINT IS TO PROVIDE VARIABLE LISTS IN DIFFERENT ORDER;

*ALPHA ORDER;

PROC SORT DATA=LABELLED_CONTENTS OUT=LABELLED_BY_NAME;
    BY NAME;
RUN;

*DESCENDING POSITION;

PROC SORT DATA=LABELLED_CONTENTS OUT=LABELLED_BY_DVARNUM;
    BY DESCENDING VARNUM;
RUN;

*LENGTH ORDER;

PROC SORT DATA=LABELLED_CONTENTS OUT=LABELLED_BY_LENGTH;
    BY LENGTH;
RUN;

////////////////////////////////////////////////////////////////////////////////
```

```
* TEST FILE PREPARATION METHOD 1: MACRO LISTS FROM METADATA VIA PROC SQL INTO:;
* CREATES MACRO LISTS THAT CAN BE USED IN DATA SET STATEMENTS AND PROC SQL;

PROC SQL;
     SELECT NAME
     INTO :VARNUM_ORDER SEPARATED BY ' '
     FROM LABELLED_CONTENTS;
quit;

%PUT By Original Order = &VARNUM_ORDER;

PROC SQL;
     SELECT NAME
     INTO :NAME_ORDER SEPARATED BY ' '
     FROM LABELLED_BY_NAME;
quit;

%PUT By Alpha Order = &NAME_ORDER;
```

```
* TEST FILE PREPARATION METHOD 2: STRUCTURED INCLUDE FILE VIA DATA _NULL_;
* CREATES AN "INCLUDE" FILE TO BE USED IN LENGTH STATEMENTS USING SAS METADATA;
* NOTE CONDITIONALLY CREATES STRINGS BASED ON TYPE;

DATA _NULL_;
     FILE '.\STRING_ORDER.TXT' LRECL=50 PAD;
     LENGTH STRING_ORDER $ 50;
     SET LABELLED_BY_NAME;
     IF TYPE=2 THEN STRING_ORDER=CATX(' ',NAME,'$',LENGTH);
     ELSE IF TYPE=1 THEN STRING_ORDER=CATX(' ',NAME,LENGTH);
     PUT STRING_ORDER;
RUN;
```

```
* TEST FILE PREPARATION METHOD 3: STRUCTURED INCLUDE FILE VIA DATA _NULL_;
* CREATES AN "INCLUDE" FILE TO BE USED IN ARRAY STATEMENTS USING SAS METADATA;
* NOTE CONDITIONALLY CREATES SEPARATE MACRO VARIABLES BASED ON TYPE;

DATA CHARS NUMS;
    SET LABELLED_BY_DVARNUM;
    IF TYPE=2 THEN OUTPUT CHARS;
    ELSE IF TYPE=1 THEN OUTPUT NUMS;
RUN;

PROC SQL;
     SELECT NAME
     INTO :DVARNUM_ORDER_C SEPARATED BY ' '
     FROM CHARS;
QUIT;

PROC SQL;
     SELECT NAME
     INTO :DVARNUM_ORDER_N SEPARATED BY ' '
     FROM NUMS;
quit;

%PUT By Descending Varnum Char Order = &DVARNUM_ORDER_C;
%PUT By Descending Varnum Num Order = &DVARNUM_ORDER_N;
```

LENGTH AND ATTRIB STATEMENTS

An example of using the LENGTH statement (created as a %INCLUDE file in test file preparation method 2) is shown below. The LENGTH statement requires that a type designation (for character variables) and length be provided, thus SAS metadata was used in the creation of the %INCLUDE file above. Note that the LENGTH statement is *above* the set statement. The ATTRIB statement can be constructed in a similar fashion as the LENGTH statement using SAS metadata, and employed in the same manner.

```
* EXAMPLE 1: LENGTH STATEMENT (ATTRIB STATEMENT IS SIMILAR);

DATA DD.HARRYPOTTER_LENGTH
    (LABEL="Harry Potter Sample Data Set - Variables Ordered with Length
     Statement");
     LENGTH
     %INCLUDE '.\STRING_ORDER.TXT';
     SET DD.HARRYPOTTER_LABELLED;
RUN;

*PROC CONTENTS ON OUTPUT DATA SET;

PROC CONTENTS DATA=DD.HARRYPOTTER_LENGTH VARNUM;
RUN;
```

ARRAY STATEMENT

Using ARRAY statements created by using SAS metadata is similar to the LENGTH statement. Separate macro variable lists of character and numeric variables are created based on variable type in the metadata and deployed in the DATA step, again *above* the SET statement. The reason for separating

character and numeric variables is because arrays do not allow mixing of variable types. Additional instructions can be put on the ARRAY statements such as format information, and depending on the metadata values for type and format, you could have multiple ARRAY statements. In this example, no information beyond type is utilized.

```
* EXAMPLE 2: ARRAY STATEMENT;

DATA DD.HARRYPOTTER_ARRAY
    (LABEL="Harry Potter Sample Data Set - Variables Ordered with Array
    Statements");
    ARRAY CH (*) $ 50 &DVARNUM_ORDER_C. ;
    ARRAY NU (*) &DVARNUM_ORDER_N. ;
    SET DD.HARRYPOTTER_LABELLED;
RUN;

PROC CONTENTS DATA=DD.HARRYPOTTER_ARRAY VARNUM;
RUN;
```

RETAIN / FORMAT / INFORMAT STATEMENTS

The RETAIN statement is similar to the ARRAY statement in that variable metadata information is not required to reorder variables, just a list of the variables in the appropriate order. The FORMAT and INFORMAT statements behave in the same fashion. All three of these DATA STEP statements can perform additional operations, but in this context, they are not necessary and can complicate the task. For example, RETAIN is used to retain, or carry values across iterations of a DATA step. If RETAIN is used for this purpose, AND to reorder the data, there's a risk of errors and/or unexpected results. The ability of these statements to reorder variables is simply a by-product of how they operate.

```
* EXAMPLE 3: RETAIN / FORMAT / INFORMAT STATEMENTS;

DATA DD.HARRYPOTTER_RETAIN
    (LABEL="Harry Potter Sample Data Set - Variables Ordered with Retain
    Statement");
    RETAIN &NAME_ORDER.;
    SET DD.HARRYPOTTER_LABELLED;
RUN;

PROC CONTENTS DATA=DD.HARRYPOTTER_RETAIN VARNUM;
RUN;
```

PROC SQL

PROC SQL in the context of variable reordering performs in a similar fashion to the RETAIN / FORMAT / INFORMAT statements in that PROC SQL only REQUIRES a reordered list of variables, but allows the specification of additional attributes, such as type, length, format, etc. PROC SQL is also unique in that it allows you to rename or "alias" variables. Often, SAS practitioners wish to reorder variables in a data file to make reporting more straightforward. However, there are circumstances that require that output data (temporary or permanent SAS data files, Excel files, XML files, etc.) be ordered in a certain way. For example for XML files that are created using XML maps or schemas, case (upper, lower, proper or

camelcase), matters *as well as* variable order. PROC SQL allows for the renaming of variables to match XML case requirements in the reordering process.

Although the code for this example was not built programmatically, as with the other three examples, it could easily be.

```
* EXAMPLE 4: PROC SQL TO REORDER VARIABLES;
PROC SQL;
    CREATE TABLE DD.HARRYPOTTER_SQL AS SELECT
    animal
    ,bloodstatus
    ,combined_name
    ,da
    ,deatheater
    ,deceased as deadasadoornail
    ,firstname
    ,house
    ,lastname
    ,middlename
    ,ministryofmagic as MOM
    ,nickname
    ,oop as OrderOfthePhoenix
    ,patronus
    ,professor
    ,quidditch_wins
    ,school
    from dd.harrypotter_labelled
    ORDER BY lastname ;
QUIT;

PROC CONTENTS DATA=DD.HARRYPOTTER_SQL VARNUM;
RUN;
```

CONCLUSION

SAS provides practitioners with a wealth of SAS metadata regarding SAS data files. Coupled with an understanding of how the Program Data Vector (PDV) works, metadata can be used to construct both documentation of data file and variables attributes and data-driven code to modify data files. These modifications include reordering variable/column order within an output data file. Methods to reorder variables described are the use of six specific data file statements and PROC SQL. The PROC SQL and the ATTRIB statement methods provide the most flexibility and least amount of time spent coding to accomplish the functional outcome of reordering variables. PROC SQL edges out the ATTRIB statement with the ability to rename or “alias” variables within the same procedural statement.

REFERENCES

Chapman, Tasha. September 2017. “Creating a data dictionary using Base SAS®.” *Proceedings of the Western Users of SAS System 2017 Conference*, Long Beach, CA.

Choate, Paul A.. September 2006. “SAS®’s Various Varying Variables, or 1000+ Ways to Manipulate SAS Variables.” *Proceedings of the Western Users of SAS System 2006 Conference*, Irvine, CA.

Clapson, Andrew. March 2014. “Ordering Columns in a SAS® Dataset: Should you really RETAIN that?” *Proceedings of the SAS Global 2014 Conference*, Washington, DC.

Cody, Ron. 2007. *Learning SAS® by Example: A Programmer’s Guide*. Cary, NC: SAS Institute Inc.

Gilsen, Bruce. April 2010. “Tales from the Help Desk 4: Still More Solutions for Common SAS® Mistakes.” *Proceedings of the SAS Global 2010 Conference*, Seattle, WA.

Go, Imelda C.. September 2002. “Inside the DATA Step: Pearls of Wisdom for the Novice SAS Programmer.” *Proceedings of the Southeast SAS Users Group 2002 Conference*, Savannah, GA.

Hadden, Louise S.. April 2015. “Better Metadata Through SAS® II: %SYSFUNC, PROC DATASETS, and Dictionary Tables.” *Proceedings of the SAS Global 2015 Conference*, Dallas, TX.

Horstman, Joshua M. and Gilbert, Britney D.. April 2015. “Inside the DATA Step: Pearls of Wisdom for the Novice SAS Programmer.” *Proceedings of the SAS Global 2015 Conference*, Dallas, TX.

Hughes, Troy Martin. 2016. *SAS® Data Analytic Development: Dimensions of Software Quality*. Hoboken, New Jersey. John Wiley & Sons, Inc.

Hughes, Troy Martin, 2016. “SAS® Spontaneous Combustion: Securing Software Portability through Self – Extracting Code”. *Proceedings of the SAS Global 2016 Conference*, Las Vegas, NV.

Jain, Adish and Bachtell, Kate. April 2012. “Reordering Values within Observations: Beyond CALL SORTC(N).” *Proceedings of the SAS Global 2012 Conference*, Dallas, TX.

Olson, Diane. April 2013. “Developer Reveals: Extended Data Set Attributes.” *Proceedings of the SAS Global 2013 Conference*, San Francisco, CA.

SAS® 9.4 Language Reference: Concepts, Sixth Edition: Reordering Variables in SAS Output

Watson, Richann. September 2018. “Driving Miss Data”. *Proceedings of the Western Users of SAS Software 2018 Conference*, Sacramento, CA.

Wright, Philip A.. October 2009. “Using the Data Step’s ATTRIB Statement to both Manage and Document Variables in a SAS® Dataset (*lightly*).” *Proceedings of the Midwest SAS Users Group 2009 Conference*, Cleveland, OH.

Yiu, Tony. April 2013. “Reordering Columns after PROC TRANSPOSE (or anytime you want, really).” *Proceedings of the SAS Global 2013 Conference*, San Francisco, CA.

ACKNOWLEDGMENTS

The author would like to express her appreciation for the reciprocal exchange of ideas with her colleague Christianna Williams, which has inspired this paper and many other coding solutions and learning experiences over the years.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Louise S. Hadden
Abt Associates Inc.
617-349-2385
Louise_hadden@abtassoc.com
abtassociates.com

APPENDIX

Below follows code to create the sample data file. However, any data file available to the user will work with the techniques described in the paper. SASHELP.CLASS or SASHELP.HEART are frequently used for demonstration purposes.

```
options ps=55 ls=132 errorabend fullstimer nocenter nodate nonumber;

libname dd '.';
filename odsout '.';
run;

title1 'Create Harry Potter themed data file';
run;

data dd.harrypotter (label='Harry Potter themed data file for use in
demonstrating functions');
     length combined_name school bloodstatus $ 50 firstname middlename
lastname nickname house patronus animal $ 25 ;

     combined_name='Potter, Harry';
     firstname='Harry';
     middlename='James';
     lastname='Potter';
     school='Hogwarts';
     house='Gryffindor';
     patronus='Stag';
     nickname='';
     animal='Snowy Owl';
     quidditch_wins=9;
     professor=0;
     ministryofmagic=1;
     deceased=0;
     da=1;
     oop=0;
     deatheater=0;
     bloodstatus='Half-Blood';
     output;

     combined_name='Longbottom, Neville';
     firstname='Neville';
     middlename='';
     lastname='Longbottom';
     school='Hogwarts';
     house='Gryffindor';
     patronus='Non-Corporeal';
     nickname='';
     animal='Toad';
     quidditch_wins=.N;
     professor=1;
     ministryofmagic=0;
     deceased=0;
     da=1;
     oop=0;
     deatheater=0;
     bloodstatus='Pure Blood';
     output;
```

```
combined_name='Lovegood, Luna';
firstname='Luna';
middlename='';
lastname='Lovegood';
nickname='Loony';
school='Hogwarts';
house='Ravenclaw';
patronus='Hare';
animal='';
quidditch_wins=.N;
professor=0;
ministryofmagic=0;
deceased=0;
da=1;
oop=0;
deatheater=0;
bloodstatus='Pure Blood';
output;

combined_name='Weasley, Ronald';
firstname='Ronald';
middlename='Bilius';
lastname='Weasley';
school='Hogwarts';
house='Gryffindor';
nickname='Ron';
patronus='Jack Russell Terrier';
animal='Rat, then Owl';
quidditch_wins=6;
professor=0;
ministryofmagic=0;
deceased=0;
da=1;
oop=0;
deatheater=0;
output;

combined_name='Weasley, Fred';
firstname='Fred';
middlename='Fabian';
lastname='Weasley';
school='Hogwarts';
house='Gryffindor';
nickname='Gred';
patronus='';
animal='';
quidditch_wins=8;
professor=0;
ministryofmagic=0;
deceased=1;
da=1;
oop=0;
deatheater=0;
bloodstatus='Pure Blood';
output;
```

```
combined_name='Weasley, George';
firstname='George';
middlename='Gideon';
lastname='Weasley';
school='Hogwarts';
house='Gryffindor';
nickname='Forge';
patronus='';
animal='';
quidditch_wins=8;
professor=0;
ministryofmagic=0;
deceased=0;
da=1;
oop=0;
deatheater=0;
bloodstatus='Pure Blood';
output;

combined_name='Granger, Hermione';
firstname='Hermione';
middlename='Jean';
lastname='Granger';
school='Hogwarts';
house='Gryffindor';
patronus='Otter';
nickname='Hermy';
animal='Cat';
quidditch_wins=.N;
professor=0;
ministryofmagic=0;
deceased=0;
da=1;
oop=0;
deatheater=0;
bloodstatus='Muggle Born';
output;

combined_name='Weasley, Ginevra';
firstname='Ginevra';
middlename='Molly';
lastname='Weasley';
school='Hogwarts';
house='Gryffindor';
patronus='Horse';
nickname='Ginny';
animal='Puffskein';
quidditch_wins=25;
professor=0;
ministryofmagic=0;
deceased=0;
da=1;
oop=0;
deatheater=0;
bloodstatus='Pure Blood';
output;
```

```
combined_name='Weasley, Percival';
firstname='Percival';
middlename='Ignatius';
lastname='Weasley';
school='Hogwarts';
house='Gryffindor';
patronus='';
nickname='Percy';
animal='Rat, then Owl';
quidditch_wins=.N;
professor=0;
ministryofmagic=1;
deceased=0;
da=1;
oop=0;
deatheater=0;
bloodstatus='Pure Blood';
output;

combined_name='Malfoy, Draco';
firstname='Draco';
middlename='Lucius';
lastname='Malfoy';
school='Hogwarts';
house='Slytherin';
patronus='';
nickname='';
animal='';
quidditch_wins=2;
professor=0;
ministryofmagic=0;
deceased=0;
da=0;
oop=0;
deatheater=1;
bloodstatus='Pure Blood';
output;

combined_name='Black, Sirius';
firstname='Sirius';
middlename='Orion';
lastname='Black';
school='Hogwarts';
house='Gryffindor';
patronus='Black Dog';
nickname='Padfoot';
animal='';
quidditch_wins=.N;
professor=0;
ministryofmagic=0;
deceased=1;
bloodstatus='Pure Blood';
output;

combined_name='Black, Regulus';
firstname='Regulus';
middlename='Arcturus';
```

```
lastname='Black';
school='Hogwarts';
house='Slytherin';
patronus='';
nickname='';
animal='';
quidditch_wins=.N;
professor=0;
ministryofmagic=0;
deceased=1;
da=0;
oop=0;
deatheater=1;
bloodstatus='Pure Blood';
output;

combined_name='Dumbledore, Albus';
firstname='Albus';
middlename='Percival Wulfric Brian';
lastname='Dumbledore';
school='Hogwarts';
house='Gryffindor';
patronus='Phoenix';
nickname='';
animal='Phoenix';
quidditch_wins=.N;
professor=1;
ministryofmagic=0;
deceased=1;
da=0;
oop=1;
deatheater=0;
bloodstatus='Half Blood';
output;

combined_name='Tonks, Nymphadora';
firstname='Nymphadora';
middlename='';
lastname='Tonks';
school='Hogwarts';
house='Hufflepuff';
patronus='Wolf';
nickname='Tonks';
animal='';
professor=0;
ministryofmagic=0;
deceased=1;
da=0;
oop=1;
deatheater=0;
bloodstatus='Half Blood';
output;

combined_name='MacMillan, Ernie';
firstname='Ernie';
middlename='';
lastname='MacMillan';
```

```
school='Hogwarts';
house='Hufflepuff';
patronus='Boar';
nickname='';
animal='';
quidditch_wins=.N;
professor=0;
ministryofmagic=0;
deceased=0;
da=1;
oop=0;
deatheater=0;
bloodstatus='Half Blood';
output;

combined_name='Dumbledore, Aberforth';
firstname='Aberforth';
middlename='';
lastname='Dumbledore';
school='Hogwarts';
house='Gryffindor';
patronus='Goat';
nickname='';
animal='';
quidditch_wins=.N;
professor=0;
ministryofmagic=0;
deceased=0;
da=0;
oop=0;
deatheater=0;
bloodstatus='Half Blood';
output;

combined_name='Potter, James';
firstname='James';
middlename='';
lastname='Potter';
school='Hogwarts';
house='Gryffindor';
patronus='Stag';
nickname='Prongs';
animal='';
quidditch_wins=9;
professor=0;
ministryofmagic=0;
deceased=1;
da=0;
oop=0;
deatheater=0;
bloodstatus='Pure Blood';
output;

combined_name='Potter, Lily';
firstname='Lily';
middlename='';
lastname='Potter';
```

```
school='Hogwarts';
house='Gryffindor';
patronus='Doe';
nickname='';
animal='';
quidditch_wins=.N;
professor=0;
ministryofmagic=0;
deceased=1;
da=0;
oop=0;
deatheater=0;
bloodstatus='Muggle Born';
output;

combined_name='Snape, Severus';
firstname='Severus';
middlename='';
lastname='Snape';
school='Hogwarts';
house='Slytherin';
patronus='Doe';
nickname='Half-Blood Prince';
animal='';
quidditch_wins=.N;
professor=1;
ministryofmagic=0;
deceased=1;
da=0;
oop=1;
deatheater=1;
bloodstatus='Half Blood';
output;

combined_name='McGonagall, Minerva';
firstname='Minerva';
middlename='';
lastname='McGonagall';
school='Hogwarts';
house='Gryffindor';
patronus='Cat';
nickname='';
animal='';
quidditch_wins=.N;
professor=1;
ministryofmagic=0;
deceased=0;
da=0;
oop=0;
deatheater=0;
bloodstatus='Half Blood';
output;

combined_name='Lupin, Remus';
firstname='Remus';
middlename='';
lastname='Lupin';
```

```
school='Hogwarts';
house='Gryffindor';
patronus='Moon';
nickname='Moony';
animal='';
quidditch_wins=.N;
professor=1;
ministryofmagic=0;
deceased=1;
da=0;
oop=1;
deatheater=0;
bloodstatus='Half Blood';
output;

combined_name='Weasley, Arthur';
firstname='Arthur';
middlename='';
lastname='Weasley';
school='Hogwarts';
house='Gryffindor';
patronus='Weasel';
nickname='';
animal='';
quidditch_wins=.N;
professor=0;
ministryofmagic=1;
deceased=0;
da=0;
oop=1;
deatheater=0;
bloodstatus='Pure Blood';
output;

combined_name='Shacklebolt, Kingsley';
firstname='Kingsley';
middlename='';
lastname='Shacklebolt';
school='Hogwarts';
house='';
patronus='Lynx';
nickname='';
animal='';
quidditch_wins=.N;
professor=0;
ministryofmagic=1;
deceased=0;
da=0;
oop=1;
deatheater=0;
bloodstatus='Pure Blood';
output;

combined_name='Finnigan, Seamus';
firstname='Seamus';
middlename='';
lastname='Finnigan';
```

```
school='Hogwarts';
house='Gryffindor';
patronus='Fox';
nickname='';
animal='';
quidditch_wins=.N;
professor=0;
ministryofmagic=0;
deceased=0;
da=0;
oop=0;
deatheater=0;
bloodstatus='Half Blood';
output;

combined_name='Chang, Cho';
firstname='Cho';
middlename='';
lastname='Chang';
school='Hogwarts';
house='Hufflepuff';
patronus='Swan';
nickname='';
animal='';
quidditch_wins=4;
professor=0;
ministryofmagic=0;
deceased=0;
da=0;
oop=0;
deatheater=0;
bloodstatus='Half Blood';
output;

combined_name='Delacour, Fleur';
firstname='Fleur';
middlename='';
lastname='Delacour';
school='Beauxbatons Academy of Magic';
house='';
patronus='';
nickname='';
animal='';
quidditch_wins=.N;
professor=0;
ministryofmagic=0;
deceased=0;
da=0;
oop=0;
deatheater=0;
bloodstatus='Unknown';
output;

combined_name='Delacour, Gabrielle';
firstname='Gabrielle';
middlename='';
lastname='Delacour';
```

```
school='Beauxbatons Academy of Magic';
house='';
patronus='';
nickname='';
animal='';
quidditch_wins=.N;
professor=0;
ministryofmagic=0;
deceased=0;
da=0;
oop=0;
deatheater=0;
bloodstatus='Unknown';
output;

combined_name='Maxime, Olympe';
firstname='Olympe';
middlename='';
lastname='Maxime';
school='Beauxbatons Academy of Magic';
house='';
patronus='';
nickname='';
animal='';
quidditch_wins=.N;
professor=1;
ministryofmagic=0;
deceased=0;
da=0;
oop=0;
deatheater=0;
bloodstatus='Half Human';
output;

combined_name='Hagrid, Rubeus';
firstname='Rubeus';
middlename='';
lastname='Hagrid';
school='Hogwarts';
house='Gryffindor';
patronus='';
nickname='Hagrid';
animal='Acromantula';
quidditch_wins=.N;
professor=1;
ministryofmagic=0;
deceased=0;
da=0;
oop=0;
deatheater=0;
bloodstatus='Half Blood';
output;

combined_name='Krum, Victor';
firstname='Victor';
middlename='';
lastname='Krum';
```

```
     school='Durmstrang Institute';
     house='';
     patronus='';
     nickname='';
     animal='';
     quidditch_wins=85;
     professor=0;
     ministryofmagic=0;
     deceased=0;
     da=0;
     oop=0;
     deatheater=0;
     bloodstatus='Unknown';
     output;

     combined_name='Grindelwald, Gellert';
     firstname='Gellert';
     middlename='';
     lastname='Grindelwald';
     school='Durmstrang Institute';
     house='';
     patronus='';
     nickname='';
     animal='';
     quidditch_wins=.N;
     professor=0;
     ministryofmagic=0;
     deceased=1;
     da=0;
     oop=0;
     deatheater=0;
     bloodstatus='Pure or Half Blood';
     output;

     combined_name='Karkaroff, Igor';
     firstname='Igor';
     middlename='';
     lastname='Karkaroff';
     school='Durmstrang Institute';
     house='';
     patronus='';
     nickname='';
     animal='';
     quidditch_wins=.N;
     professor=1;
     ministryofmagic=0;
     deceased=1;
     da=0;
     oop=0;
     deatheater=1;
     bloodstatus='Pure Blood';
     output;

run;

ods rtf file='SampleData.rtf' path=odsout style=styles.journal;
```

```
proc print data=dd.harrypotter (obs=25) noobs;
title2 'Test created input data file';
run;

ods rtf close;
```