Dirty Data? Clean it up with SAS®

Lindsey Brown Philpot[1,2], Gabriela Cantu[1]
[1] Institute for Health Care Research and Improvement, Baylor Health Care System
[2]University of North Texas School of Public Health, Department of Health Management and Policy

ABSTRACT
Clinical trials data can be complex and integrate multiple data elements including demographic, laboratory, clinical, medication, and medical history. Although extremely valuable to the study the completeness and cleanliness of clinical trials data is often less than ideal. In order to be successful, clinical data managers must strategize methods to maintain data integrity and cleanliness. This presentation will focus on planning for and performing clinical trials data edit checks, cleaning and documentation.

Through a comprehensive planning process and a series of simple SAS procedures, dirty data can be transformed into usable and clinically informative datasets. A simple ARRAY can be used to reassign tricky variables into more useful formats. Utilization of PROC UNIVARIATE to produce continuous variable statistics allows data managers to identify out of range and unexpected values for clinical data. Additionally, PROC FREQ will allow data managers to check for inappropriate or incorrect values for categorical variables. Through a series of MACROS, the clinical researcher is able to execute these SAS procedures with minimal key strokes and repetition. SAS provides clinical data managers real time documentation of both data cleaning procedures and results.

INTRODUCTION
Clinical trials data can be complex and integrate multiple data elements including demographic, laboratory, clinical, medication, and medical history. Although extremely valuable to the study the completeness and cleanliness of clinical trials data is often less than ideal. In order to be successful, clinical data managers must strategize methods to maintain data integrity and cleanliness. This presentation will focus on planning for and performing clinical trials data edit checks, cleaning and documentation.

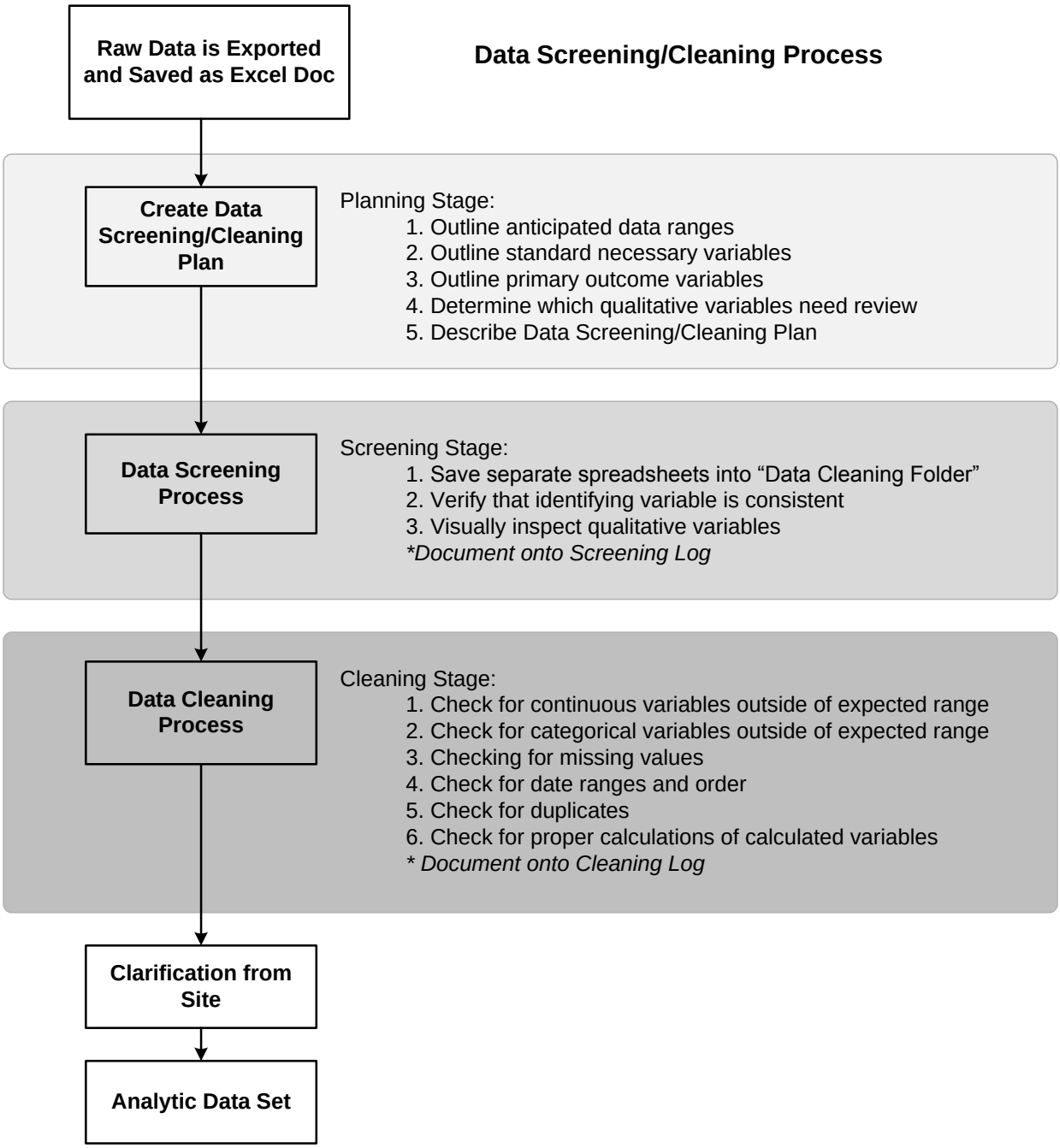
Through a comprehensive planning process and a series of simple SAS procedures, dirty data can be transformed into usable and clinically informative datasets. A simple ARRAY can be used to reassign tricky variables into more useful formats. Utilization of PROC UNIVARIATE to produce continuous variable statistics allows data managers to identify out of range and unexpected values for clinical data. Additionally, PROC FREQ will allow data managers to check for inappropriate or incorrect values for categorical variables. Through a series of MACROS, the clinical researcher is able to execute these SAS procedures with minimal key strokes and repetition. SAS provides clinical data managers real time documentation of both data cleaning procedures and results.

The following paper describes a series of steps, as outlined in the flow diagram (Figure 1) that have been adapted from Good Clinical Practice Guidelines and with the aid of Ron Cody's Data Cleaning Techniques book (Cody 2008).

OUTLINE
1. Planning: Creation of a Data Screening and Cleaning Plan
2. Data Screening: Preparing Datasets, Inspection of Qualitative Variables, and Verifying the Identifying Variable Name is Consistent Between Data Tables

3. Data Cleaning: Checking for Outlying Values, Missing Values, Inappropriate Date Ranges, Duplicates and Miscalculated Variables
4. Requesting Clarification from the Clinical Site
5. Preparing the Analytic Dataset

Figure 1. Flow-diagram of steps in Data Screening and Cleaning Process for Clinical Trials



1. PLANNING: CREATION OF A DATA SCREENING AND CLEANING PLAN

Documentation is the key to success when working with clinical trials data. Therefore, the first step to cleaning your clinical trials data is to create a Data Screening and Cleaning Plan. This Data Screening and Cleaning Plan is used throughout the process to guide the individual executing the plan, to document which steps are performed, as well as the provide a record of the results of each step along the data screening and cleaning journey. For ease of use, we used an Excel spreadsheet in which five tabs were employed:

- Plan Outline
- Screening Process
- Screening Log
- Cleaning Process
- Cleaning Log

	A	B	C	D	E
1	**Information to Plan for**	**Specifics**	**Complete/Incomplete**	**Completed By:**	**Date of Completion:**
2	**1. Outline anticipated data ranges within Data Dictionary (Continuous)**				
3	Vital Signs Baseline	weight_kg (Exp=>50 kg)	Complete	BM	10-Aug-12
4		temp (Exp=96.4-99.6)	Complete	BM	10-Aug-12
5	Blood Labs: CBC with Differential	cbc_wbc_result (Exp=4.0-10.0)	Complete	BM	10-Aug-12
6		cbc_rbc_result (Male - Exp=4.50-5.90) (Female - Exp=4.00-5.20)	Complete	BM	10-Aug-12
7	Follow Up Questionnaire	fu_today(Exp=1-10)	Complete	BM	10-Aug-12
8	**2. Outline anticipated data values within Data Dictionary (Categorical)**				
9	Demographics	Gender (Exp= 1 - Male, 2 - Female)	Complete	BM	10-Aug-12
10		Race (Exp= 1-Asian, 2-White, 3-Black, 4-Hawaiian/PI, 5-AM/AN, 6-Other)	Complete	BM	10-Aug-12
11	Vaccination History	flu_hx (Exp=1-Yes, 0-No)	Complete	BM	10-Aug-12
12	**3. Outline Standard Necessary Variables (Inclusion Criteria, Exclusion Criteria, Date of Consent)**				
13	Inclusion Variables	incl_weight, incl_age, incl_pe_lab, incl_no_infection, incl_no_vaccine, incl_summary (Exp=1)	Complete	BM	10-Aug-12
14	Exclusion Variables	excl_vein, excl_preg, excl_disease_hx, excl_autoimmune_hx, excl_infection, excl_hic, excl_steroid, excl_allergy_vaccine	Complete	BM	10-Aug-12
15	Consent	incl_consent (Exp=1), incl_consent_date (Exp=present)	Complete	BM	10-Aug-12
16	**4. Outline Primary Outcome Variables**				
17	Study Vaccination	vaccine_given, vaccine_reason, vaccine_type, vaccine_date, vaccine_time, vaccine_site, vaccine_site_specify	Complete	BM	10-Aug-12
18	Apheresis	apheresis_day, apheresis_date, apheresis_nd, apheresis_reason	Complete	BM	10-Aug-12
19	**5. Data Screening - Plan manual/visual verification**				
20	Save Separate Spreadsheets into "Data Cleaning Folder"	Q:\Research\Example\Data management\Data review and cleaning\Data Cleaning_01Aug2012	Complete	BM	6-Aug-12
21	Verify that identifying variable consistent bewteen extract speadsheets	Identifying variable="ssid"	Complete	BM	6-Aug-12
22	Visual inspection of all qualitative variables	Eligibility: Summary>exception_1_specify	Complete	BM	14-Aug-12
23	**6. Plan programmed data cleaning**				
24	Check for Outlying Numeric Variables	PROC UNIVARIATE/PROC FREQ	Complete	BM	10-Aug-12
25	Check for Missing Values	PROC PRINT	Complete	BM	10-Aug-12
26	Check Date Ranges	DATA STEP	Complete	BM	14-Aug-12
27	Check for Duplicates	PROC SORT NODUP	Complete	BM	14-Aug-12
28	Check for Proper Calculations	DATA STEP/PROC PRINT	Complete	BM	14-Aug-12

Figure 2. Screen Shot of Excel Spreadsheet Used to Document Data Screening and Cleaning Plan and Results.

During the planning process, it is important to consult with the principle investigator(s) and statistician(s) in order to determine which variables are most important for the study. A typical clinical trial can collect up to hundreds of data points, and screening and cleaning every variable may not be the best use of resources. Within the context of our studies, we have outlined six objectives to complete during the course of the data screening and cleaning process: 1) Outline anticipated data ranges for continuous variables, 2) Outline anticipated data ranges for categorical variables based on the data dictionary, 3) Outline any standard necessary variables, including complete inclusion and exclusion criteria, and noted date of informed consent, 4) Outline primary outcome variables, 5)Perform manual/visual verification of any qualitative variables, 6)Created a detailed data screening/cleaning plan that can be reused as date sets update. Each clinical trial project is unique, and can also be limited by the budget available to conduct all of the objectives listed above. Be sure to adjust objectives of your data screening and cleaning procedures to meet the needs of your specific clinical trial project.

2. DATA SCREENING

Once the data screening and cleaning plan has been created and reviewed, it is time to jump into the data. We have outlined the data screening and cleaning process into two distinct sections, which can be performed by individuals with different levels of SAS programming skills. The data screening process

can be performed outside of SAS completely, so it will only be discussed briefly within the context of this paper.

	A	B	
1	**Data Screening Process**	**Specifics**	**Status**
2	**Save Separate Spreadsheets into "Data Cleaning Folder"**	**Copy of "Database Exctracts\hipc08032011" into "Data Cleaning\Data Extract 8.3.2011"**	
3		Location: Q:\Research\Example\Data management\Data review and cleaning\Data Cleaning_01Aug2012	Complete
4	**Verify that identifying variable consistent bewteen extract speadsheets**		
5		Identifying variable="ssid"	Complete
6	**Visual inspection of all qualitative variables**		
7	Visual inspection of all qualitative variables	Eligibility: Summary>exception_1_specify	Complete
8		Eligibility: Summary>exception_2_specify	Complete
9		Eligibility: Summary>exception_3_specify	Complete
10		Demographics>race_other	Complete
11		Blood Labs: CBC>cbc_other_specify	Complete
12		Processing Profile>type_interpret_specify	Complete

Figure 3. Screen Shot of Specifics of Data Screening Process.

Data screening can be viewed as a data manager's initial interaction with and look at the data. We suggest saving a copy of each data table into a separate folder in order to protect against any accidental over-writing. The data screening process also involves verifying that the main identifying variable, or merging variable, is consistent between spreadsheets. Finally, the data screening process may involve a visual inspection of all qualitative variables that are deemed important by the primary investigator(s) or statistician(s).

	A	B	C	D	E	F	G	H
1	**Patient:**	**Form:**	**Visit:**	**Variable:**	**Modification Suggested By:**	**Modified Reason:**	**Modification Made (Yes/No):**	**Modification Made By:**
2	015	Medical history	Screening	Description-Menstral	DP	Misspelled, should be menstrual	Yes	DP
3	016	Medical history	Screening	Description-Arhoxicillin	DP	Misspelled, should be amoxicillin	Yes	DP

Figure 4. Screen Shot of Specifics of Data Screening Log.

All data discrepancies found during the data screening process should be noted on the Data Screening Log.

3. DATA CLEANING

Clinical trials data need to be reviewed to identify continuous and categorical values that are out of the expected range, missing values, incorrectly ordered dates, duplicate observations, and miscalculated values. Documentation of clinical trials data cleaning is equally as important as performing the data cleaning process.

The Cleaning Process spreadsheet is used by data managers to accurately document the data cleaning procedure and its results. Below is an example of a data Cleaning Process spreadsheet. The Microsoft Excel column headers contain categories such as, Data Cleaning Item, Programming Step, Table, Variables Verified, and Outcome. Each column serves as a guide to the data manager as they contemplate what variables need cleaning and what SAS PROC they will use to perform the cleaning process.

The Data Cleaning Item column is where the data manager lists out what stage of cleaning they will be performing: 1) Checking continuous variables for values outside of the expected range, 2) Checking for categorical variables outside of the expected range, 3) Checking for missing values, 4) Checking for date ranges that are incorrect or out of order, 5) Checking for duplicates, or 6) Checking for miscalculations of calculated variables.

The next column header, Programming Step is the data manager's opportunity to contemplate which SAS PROC they plan to use to achieve review of the listed data cleaning stage. To identify out of range variables data managers will use PROC UNIVARIATE for continuous variables and PROC FREQ for categorical variables. PROC UNIVARIATE will provide the mean, median, mode, IQR, minimum and maximum values. PROC FREQ will supply a listing of each unique value for the variable and the count and percentage of the time the variable response occurs. To check for missing values, they will use PROC PRINT coupled with a "WHERE" statement. Out of order date ranges can be identified by creating a "flag" in a SAS DATA STEP each time a date precedes or follows a study visit date incorrectly. To check for duplicates, PROC SORT and a NODUPKEY with a DUPOUT= statement will be used. Miscalculated variables will be flagged in a DATA STEP and then printed using PROC PRINT with a WHERE statement.

The following two column headers, Table and Variables Verified, are used to help the data manager identify which variables will be cleaned and in which tables these variables are currently located.

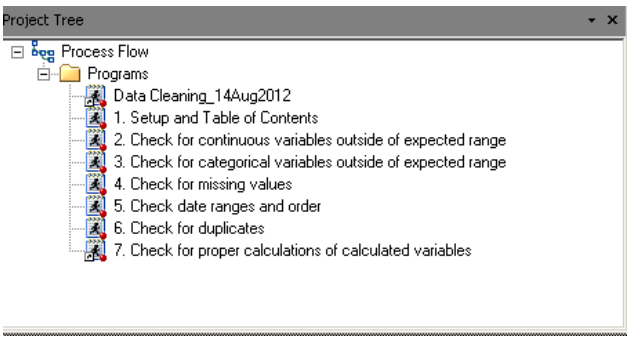
The last column header, Outcome, is arguably the most important. This column is where the data manager will record their cleaning process findings. Examples of outcomes listed below include whether a variable "Passed" or "Failed" and a record of any missing items or duplicate entries. It also allows for some more detailed information to be listed, perhaps a variables expectant normal range and the actual found range to demonstrate the discrepancy.

	A	B	C	D	E	F	G
1	Data Cleaning Item	Programming Step	Table	Variables Verified	Outcome		
2	Check for Outlying Numeric Variables	PROC UNIVARIATE			Pass/Fail	Detail	Normal Range
3		Example.vital_signs_baseline_format	vital_signs_baseline_format	Weight (kg)	Pass	Range (60.7-119.8)	>50 (Inclusion
4		Example.vital_signs_baseline_format	vital_signs_baseline_format	Temperature	Fail (See Cleaning Log)	Range (94.9-98.3)	96.4-99.6
56	Check for Outlying Numeric Variables	PROC FREQ			Pass/Fail	Detail	Normal Range
57		Example.adverse_events_format	adverse_events_format	ssid	Pass		
58		Example.adverse_events_format	adverse_events_format	ae_none	Pass		
59							
159	Check for Missing Values	PROC FREQ/PROC PRINT					
329		Example.pregnancy_test_format	pregnancy_test_format	ssid	No missing		
330		Example.pregnancy_test_format	pregnancy_test_format	study_event_oid	No missing		
331		Example.pregnancy_test_format	pregnancy_test_format	serum_preg_date	26 Missing and should be		
347	Ensure Proper formatting of Dates	DATA STEP					
348		Example.apheresis_format	_apheresis_format	apheresis_date	DATE9.		
349		Example.blood_labs_format	blood_labs_format	cbc_date	DATE9.		
350		Example.blood_labs_format	blood_labs_format	serum_date	DATE9.		
351		Example.blood_labs_format	blood_labs_format	ldh_date	DATE9.		
352		Example.blood_labs_format	blood_labs_format	magnesium_date	DATE9.		
365							
366	Check Date Ranges	DATA STEP					
367		Example.demographics_format	demographics_format	birth_date	Verified Birth date < Vaccination Date		
368		Example.medical_history_format	medical_history_format	medhx_date	Verified Medical History Date < Vaccination Date		
369		Example.vital_signs_baseline_format	vital_signs_baseline_format	vitals_date	Verified tt all date ranges fall between medhx_date and comp_date		
383	Check for Duplicates	PROC SORT NODUP					
384	Ensure no Duplicate entries				Pass		
385							
386	Verify calculation of specified variables	DATA STEP/PROC PRINT					
387	calculated variables should be confirm	Example.Physical_Exam_Baseline	Physical_Exam_Baseline	BMI	Pass		
388							

Cleaning Outline | Screening Process | **Cleaning Process** | Screening Log | Cleaning Log

Figure 5. Screenshot of Cleaning Process spreadsheet.

SAS Enterprise Guide provides an efficient platform to organize the appropriate programs related to the data cleaning process. The process flow and project tree below organizes the data manager's programs to reflect the steps which will unfold throughout the data cleaning process.



Program 1: The first SAS program listed in the project tree is designed to layout the following cleaning steps in a Table of Contents and also assign the permanent project libraries where datasets and output for this data cleaning process will be stored. This provides a permanent record to serve as documentation of the datasets that underwent the cleaning process.

```
*1. Set location to SAS Datasets
Location: Q:\Research\Example\Data management\Data extracts\17 June 2012\Analysis Data\Formatted
*************************************************************************;
*************************************************************************;
libname Example " Q:\Research\Example\Data management\Data management\Data extracts\Analysis Data\Unformatted";
libname format " Q:\Research\Example\Data management\Data extracts\14Sep2012\Archive\Analysis Data\Formatted";
```

Program 2: A SAS MACRO is used to efficiently check continuous variables for outliers. The MACRO runs PROC UNIVARIATE for a combination of datasets and variables and provides the minimum, maximum, mean, median, mode, and IQR for each variable. The data manager can utilize the Cleaning Outline to identify expected ranges of variables and verify whether the output from this MACRO is within these ranges. Out of range values, can be the result of data entry errors and can affect data in the analysis phase by skewing data above or below the true value.

```
*************************************************************************
*2. Check for continuous variables outside of expected range*******
*************************************************************************

%MACRO CONTINUOUS (DSN, ID, VAR1, VAR2);
PROC UNIVARIATE DATA=&DSN;
    ID &ID;
    VAR &VAR1 &VAR2 &VAR3;
RUN;
%MEND CONTINUOUS;

%CONTINUOUS (EXAMPLE.VITAL_SIGNS, SSID, WEIGHT, TEMP);
%CONTINUOUS (EXAMPLE.BLOOD_LABS, SSID, CBC_LYMPHOCYTE, CBC_PROTEIN;
```

After reviewing the output, the noted variables are labeled out of range in the Cleaning Process spreadsheet.

	A	B	C	D	E	F	G
1	Data Cleaning Item	Programming Step	Table	Variables Verified	Outcome		
2	Check for Outlying Numeric Variables	PROC UNIVARIATE			Pass/Fail	Detail	Normal Range
3		Example.vital_signs_baseline_format	vital_signs_baseline_format	Weight (kg)	Pass	Range (60.7-119.8)	>50 (Inclusion)
4		Example.vital_signs_baseline_format	vital_signs_baseline_format	Temperature	Fail (See Cleaning Log)	Range (94.9-98.3)	96.4-99.6

Figure 6. Screenshot of the Cleaning Process spreadsheet documenting checks for outlying numeric continuous variables

Program 3: The use of a second SAS MACRO with an embedded PROC FREQ statement is used to identify outliers of categorical variables. PROC FREQ creates a listing of all variable values and the number and percentage of times that value occurs. Again, a comparison can be made between the output and already identified expected ranges to locate any outlying values.

```
*****************************************************************
*3. Check for categorical variables outside of expected range*****
*****************************************************************
%MACRO CATEGORICAL (DSN);
PROC FREQ DATA= &DSN;
    TABLES _ALL_/NOCUM NOPERCENT;
RUN;
%MEND CATEGORICAL;
```

```
%CATEGORICAL (EXAMPLE.ADVERSE_EVENT);
%CATEGORICAL (EXAMPLE.APHERESIS);
%CATEGORICAL (EXAMPLE.ELIGIBILITY);
```

The variables that contain outlying values will be recorded as “failed” in the Cleaning Process spreadsheet and a description of the incorrect value will be placed in the Detail column. Normal or Expected Ranges of values should also be recorded in the Cleaning Process spreadsheet.

	A	B	C	D	E	F	G
1	Data Cleaning Item	Programming Step	Table	Variables Verified	Outcome		
56	Check for Outlying Numeric Variables	PROC FREQ			Pass/Fail	Detail	Normal Range
57		Example.adverse_events_format	adverse_events_format	ssid	Pass		
58		Example.adverse_events_format	adverse_events_format	ae_none	Pass		
59							

Figure 7. Screenshot of the Cleaning process spreadsheet documenting the check for outlying numeric categorical variables

Program 4: PROC PRINT coupled with a WHERE statement is used to check for missing values of numeric variables. Likewise, to check for missing character data, a WHERE statement with an open and closed quotation mark (‘’) would replace the period (.) below. Missing data can greatly affect a study’s outcomes by not providing a true representation of the results of the study.

```
*****************************************************************
*4. Check for missing values***************************************
*****************************************************************
```

```
/*Missing values found in serum ranges.*/
proc print data=example.blood_labs;
where serum_sodium_range eq . or
      serum_potassium_range eq . or
      serum_chloride_range  eq . or
      serum_co2_range eq . or
      serum_creatinine_range eq . or
      serum_bc_ratio_range eq . or
      serum_glucose_range eq . or
      serum_calcium_range eq . or
      serum_protein_range eq . or
      serum_albumin_range eq . or
      serum_globulin_range eq . or
      serum_tbil_range eq . or
      serum_alt_range eq . ;
var ssid study_event_oid serum_sodium_range serum_potassium_range
    serum_bc_ratio_range serum_glucose_range serum_calcium_range
    serum_tbil_range serum_alt_range ;
run;
```

Information regarding the variables that contain missing values are recorded in the Cleaning Process spreadsheet as seen in Figure 8.

	A	B	C	D	E	F	G
1	Data Cleaning Item	Programming Step	Table	Variables Verified	Outcome		
159	Check for Missing Values	PROC FREQ/PROC PRINT					
329		Example.blood_labs	blood_labs	ssid	No missing		
330		Example.blood_labs	blood_labs	serum_potassium_range	No missing		
331		Example.blood_labs	blood_labs	serum_alt_range	26 Missing and should be		

Figure 8. Screenshot of the Cleaning process spreadsheet documenting the check for missing variables

Program 5: In order to check for date ranges that inappropriately precede or follow study event dates, a "flag" is created using a combination of IF/THEN statements in a DATA STEP. Data managers are attempting to identify data entry errors. For example, if a participant's date of birth occurred after their first clinical trial visit, then that date of birth needs to be "flagged" as out of order and investigated. Any incorrectly ordered dates may raise questions about the integrity of the collected data.

```
*************************************************************************************************************
*5. Check date ranges and order*****************************************************************************
*************************************************************************************************************
/*identifying date fields where vaccination date or apheresis date incorrectly precede or follow an event
/*events tested here include birth, consent, medical history, study completion*/
Data work.flag;
 set work.all;
 If birth_date - vaccine_date > 0 then birth_flag =1;
 Else birth_flag=0;
 If medhx_date - vaccine_date > 0 then med_flag =1;
 Else med_flag=0;
 If incl_consent_date - vaccine_date > 0 then consent_flag =1;
 else consent_flag=0;
 If apheresis_date1 - vaccine_date < 0 then aph_flag =1;
 If apheresis_date2 - vaccine_date < 0 then aph_flag =1;
 Else flag=0;
 If comp_date - apheresis_date2 < 0 then ca_flag=1;
 else ca_flag=0;
 keep ssid birth_flag med_flag consent_flag aph_flag;
 run;
```

Dates that are flagged as out of order are recorded in the Cleaning Processes Log, as seen below.

	A	B	C	D	E	F	G
1	Data Cleaning Item	Programming Step	Table	Variables Verified	Outcome		
366	Check Date Ranges	DATA STEP					
367		Example.demographics_format	demographics_format	birth_date	Verified Birth date < Vaccination Date		
368		Example.medical_history_format	medical_history_format	medhx_date	Verified Medical History Date < Vaccination Date		
369		Example.vital_signs_baseline_format	vital_signs_baseline_format	vitals_date	tt all date ranges fall between medhx_date and comp_date		

Figure 9. Screenshot of the Cleaning process spreadsheet documenting the checking of date ranges and order.

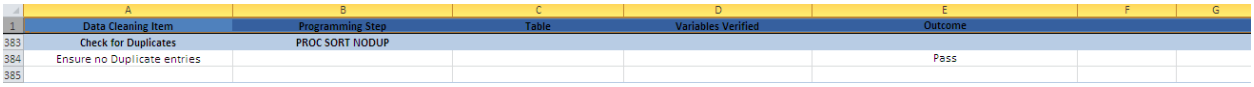
Program 6: To identify duplicate observations use the PROC SORT procedure with a NODUPKEY and DUPOUT= statement. The NODUPKEY will identify all duplicate records based on the variable listed in the BY statement. These duplicate observations will be moved to a new dataset specified in the DUPOUT= statement. The utilization of a SAS MACRO will make this process more efficient and allow duplicate entries in multiple datasets to be identified with minimal key strokes.

```
*************************************************************
*6. Check for duplicates************************************
*************************************************************
%macro nodup (dsn=,dsnout=);
title "Checking for Duplicates for &dsn";
proc sort data=&dsn dupout=&dsnout nodupkey;
    by ssid;
run;

proc print data=&dsnout;
    id ssid;
run;
title;
%mend nodup;
```

```
%nodup (dsn=example.adverse_events, dsnout=work.aedup);
%nodup (dsn=example.apheresis, dsnout=work.apheresisdup);
```

The information pertaining to duplicate observations is recorded in the Cleaning process spreadsheet as seen in Figure 10.



	A	B	C	D	E	F	G
1	Data Cleaning Item	Programming Step	Table	Variables Verified	Outcome		
383	Check for Duplicates	PROC SORT NODUP					
384	Ensure no Duplicate entries				Pass		
385							

Figure 10. Screenshot of the Cleaning process spreadsheet documenting the "check for duplicates"

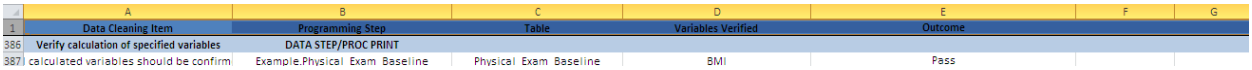
Program 7: Improperly calculated data can be the result of data entry error or information that is incorrectly calculated and recorded on the paper case report form. Checking the values of calculated variables will help eliminate incorrect data from entering the analytic dataset. A SAS DATA STEP creates a new calculated variable, in this example "BMI_CALC" from a known formula. An IF/THEN statement will then compare the newly calculated variable with value entered, stored in this example as "BMI." Values that do not match will be flagged by creating a third variable "BMI_FLAG". The data manager can then use PROC PRINT with a WHERE statement to identify all miscalculated variables.

```
 ************************************************************
 ************************************************************
 *7. Check for proper calculation of calculated variables****
 ************************************************************;

DATA Example.PHYSEXAMBASELINE;
     SET Example.PHYSEXAM;
     BMI_CALC=(WEIGHT*703)/(HEIGHT**2);
     IF BMI_CAL NE BMI THEN BMI_FLAG=1;
     RUN;

PROC PRINT DATA=Example.PHYSEXAMBASELINE;
     WHERE BMI_FLAG=1;
     VAR SSID BMI_FLAG BMI BMI_CALC;
     RUN;
```

Any values that do not match should be identified in the Cleaning Processes Log, as seen in Figure 11.



	A	B	C	D	E	F	G
1	Data Cleaning Item	Programming Step	Table	Variables Verified	Outcome		
386	Verify calculation of specified variables	DATA STEP/PROC PRINT					
387	calculated variables should be confirm	Example.Physical Exam Baseline	Physical Exam Baseline	BMI	Pass		

Figure 11. Screenshot of the Cleaning process spreadsheet documenting the verify calculation of specified variables.

4. REQUEST DATA CLARIFICATIONS FROM THE CLINICAL SITE

Following your data screening and cleaning procedures, you may have a list of questions regarding the content and completeness of your clinical trial data. Before you contact the clinical site, it is often helpful to consult the source document to verify whether a data-entry error has occurred. However, if the source documentation is not available, contacting the clinical site directly may be your only option.

Contacting the clinical site with these questions via data clarification requests is done through a custom built Access database at Baylor, but an Excel spreadsheet listing items in need of clarification can be just as effective. Be sure to indicate the study participant ID and a thorough description of the data field in question to the site in order to avoid any unnecessary confusion.

Once the clinical site has responded to your data clarification requests with sufficient information, it is typically recommended that the error be corrected directly within the data collection database. Changes to clinical data should be documented within the appropriate Data Screening or Cleaning Log.

5. PREPARING THE ANALYTIC DATASET

Now you are in the final stage of clinical trials data screening and cleaning. You have successfully gone through a thorough screening of qualitative variables, checked for continuous and categorical outliers, duplicates, missing values, miscalculated variables, and improperly ordered date variables.

In the final step, you will prepare your analytic dataset by performing a few brief programming steps to replace any missing data markers with "."s (Program 8), recode any inconsistently reported values (Program 9), and export your data into SAS data sets (Program 10) for use by your primary investigator(s) or statistician(s).

Program 8: Use a simple ARRAY statement to replace coded missing values from the clinical data set, in this case coded as "9999" to analytic missing values for the analytic data set (".") . Due to the nature of clinical trials data, the clinical site may not be able to provide all of the data requested for each patient. In this case, the data management team may designate a coded missing value (9999) to indicate that the site has been contacted, but is not able to provide the data point. However, a "9999" or other designated coded missing value may impact the findings of the study if not removed before analysis. Therefore, it is important for those preparing the analytic dataset to remove these flags before data analysis begins.

```
DATA Example;
    SET EHR_IMP.EHR_NURSING_1;
 ************************************************************************
 *8. Replace 9999 Value with "."*****************************************
 ************************************************************************;
    ARRAY REPLACE_MISSING {5}
    var1 var2 var3 var4 var5;
        DO i=1 TO 5;
            IF  REPLACE_MISSING {i}=9999 THEN REPLACE_NA {i}=MISSING;
        END;
 RUN;
```

Program 9: Use a series of IF/THEN statements to recode data entry inconsistencies into standard responses. The data provided by clinical trials is not always reported in a consistent format. For this example, blood type was reported in a free-text field. For each O-Positive study participant, their blood type could have been reported in any of the following ways: "O Positive", "O positive", "O Pos", or "O pos". Since each of these responses truly represents the same value, they can be collapsed into a consistent format using IF/THEN statements.

```
*************************************************************************************
*9. Corrections to the Processing Profile dataset*********************************
*************************************************************************************

DATA format.Example1;
     SET Example;
 IF TYPE_INTERPRET_SPECIFY = '0 Positive' THEN TYPE_INTERPRET_SPECIFY = 'O Positive';
 IF TYPE_INTERPRET_SPECIFY = '0 positive' THEN TYPE_INTERPRET_SPECIFY = 'O Positive';
 IF TYPE_INTERPRET_SPECIFY = '0 negative' THEN TYPE_INTERPRET_SPECIFY = 'O Negative';
 IF TYPE_INTERPRET_SPECIFY = 'O Pos' THEN TYPE_INTERPRET_SPECIFY = 'O Positive';
 IF TYPE_INTERPRET_SPECIFY = 'O pos' THEN TYPE_INTERPRET_SPECIFY = 'O Positive';
 IF TYPE_INTERPRET_SPECIFY = 'A pos' THEN TYPE_INTERPRET_SPECIFY = 'A Positive';
 IF TYPE_INTERPRET_SPECIFY = 'A positive' THEN TYPE_INTERPRET_SPECIFY = 'A Positive';
 IF TYPE_INTERPRET_SPECIFY = 'B Pos' THEN TYPE_INTERPRET_SPECIFY = 'B Positive';

proc freq data=format.Example1;
 table type_interpret_specify;
 run;
```

Program 10: Once you are comfortable with the quality and completeness of your dataset, you are ready to export your analytic datasets! A MACRO can be used to simplify the export process for future exports of ongoing data cleaning procedures.

```
*************************************************************************************
*10. Output of formatted datasets*************************************************
*************************************************************************************

%macro out (dsn=,dsnout=);

DATA &dsnout;
    SET &dsn;
RUN;
%mend out;

%out (dsn=Example_adverse_events, dsnout=format.Example_adverse_events_format);
%out (dsn=Example_blood_labs, dsnout=format.Example_blood_labs_format);
%out (dsn=Example_EndofStudy, dsnout=format.Example_EndofStudy_format);
```

Conclusion

Clinical data used in research projects can be messy and ill formatted. The first step should always include a well-documented screening and cleaning plan as well as a method by which documentation can take place. This paper shows how creation and organization of several programs into a SAS EG project can allow data managers to efficiently flag data for cleaning. Through simple SAS PROCS, DATA STEPS, MACROS and ARRAYS, data can be identified as an outlier, missing, incorrect, or as a duplicate. Additionally, SAS can provide an avenue to identify potentially needed data corrections and create uniform and formatted clinical analytic datasets.

Reference:

Cody, Ron. 2008. *Cody's Data Cleaning Techniques Using SAS®, Second Edition.* Cary, NC: SAS Institute Inc.

International Conference on Harmonisation, Guideline for Good Clinical Practice. Retrieved October 1, 2012, from http://ichgcp.net/ich-gcp-en.pdf.