Paper PR02

De-Identification of Clinical Trials Data Demystified

Jack Shostak, Duke Clinical Research Institute (DCRI), Durham, NC

ABSTRACT

This paper discusses the de-identification and anonymization of clinical trials data. The reasons why you would want to de-identify and anonymize clinical trials data and the regulations that define the task will be discussed. Then a SAS based approach to clinical trials data de-identification will be presented. Finally, some de-identification challenges will be mentioned. The SAS code provided in this paper was developed with BASE SAS version 8.2 on UNIX, but the code should be compatible or portable to other operating systems and future releases of SAS.

INTRODUCTION

Why would you want to de-identify and anonymize your clinical trials data? For the purposes of this paper we will define "de-identified data" as clinical trial data that contain no individually identifiable health information and "anonymized" as clinical trial data for which there is no way to link the data back to a subject. Here are some excellent reasons to de-identify and anonymize clinical trial data:

GOOD FOR SOCIETY

The main reason behind de-identifying and anonymizing clinical trials data is that it can then be used more broadly by researchers for the benefit of public health. The vast stores of clinical trials data could be brought out from proprietary or hidden data warehouses and studied for the benefit of society as a whole.

NIH MANDATE

As of October 1, 2003, the National Institutes of Health (NIH) under the United States Department of Health and Human Services mandates that all research projects seeking more than $500,000 in a given year must have a data sharing plan[1]. This means that the NIH expects researchers to share their source data, and not just the statistical results of their analysis, for other researchers to use. Although the NIH provides researchers with broad flexibility in how they share their data, the data must be shared in a way that protects the confidentiality of the research subjects. The techniques identified in this paper will satisfy this requirement.

INSTITUTIONAL DIRECTIVE

De-identifying and anonymizing clinical trials data fits in well with the directive of academic research organizations. Where I work at the Duke Clinical Research Institute, it is our mission to improve evidence-based medicine and, "To develop and share knowledge that improves the care of patients around the world through innovative clinical research."[2] Our focus is on sharing knowledge, and data when possible, in order to improve the public health of all people.

MINIMIZES REGULATORY BURDEN ON PATIENT DATA PRIVACY

The Food and Drug Administration, the "Common Rule" of the United States Department of Health and Human Services Office of Civil Rights, and the Privacy Rule of the Health Insurance Portability and Accountability Act of 1996 largely regulate the privacy of data from clinical trials research. We will look at these regulations in more detail in the next section or this paper to determine how to de-identify and anonymize clinical trials data, but the point is that there are regulations in place that will not allow you to release individually identifiable health information to the public. If you de-identify and anonymize clinical trials data then these laws likely no longer regulate the data and you are much more able to disseminate your data for other uses.

REGULATIONS

In this section of the paper, we will examine the laws in the United States that govern the privacy of clinical trials data and what it means to de-identify and anonymize the data.

"COMMON RULE"

45CFR46, also known as the "Common Rule", applies to "all **research** involving **human subjects** conducted, supported or otherwise subject to regulation by any federal department or agency which takes appropriate administrative action to make the policy applicable to such research."[3] I added the bolding for emphasis because we will examine how "research" and "human subjects" are defined. The Common Rule defines "research" in 45CFR46.102(d) as "a systematic investigation, including research development, testing and evaluation, designed to develop or contribute to generalizable knowledge."[4] 45CFR46.102(f) goes on to define "human subjects" as:

> *"Human subject* means a living individual about whom an investigator conducting research obtains
>
> (1) Data through intervention or interaction with the individual, or
>
> (2) Identifiable private information.
>
> *Intervention* includes both physical procedures by which data are gathered (for example, venipuncture) and manipulations of the subject or the subject's environment that are performed for research purposes. Interaction includes communication or interpersonal contact between investigator and subject. *Private information* includes

> information about behavior that occurs in a context in which an individual can reasonably expect that no observation or recording is taking place, and information which has been provided for specific purposes by an individual and which the individual can reasonably expect will not be made public (for example, a medical record). Private information must be individually identifiable (i.e., the identity of the subject is or may readily be ascertained by the investigator or associated with the information) in order for obtaining the information to constitute research involving human subjects."[5]

So, depending on the study site, clinical trials research is governed by the Common Rule and requires Institutional Review Board (IRB) review and approval prior to conducting research. However, if clinical trials data are de-identified and anonymized then they are no longer data on "human subjects" and therefore are exempt from Common Rule regulation for further use. They key with the Common Rule is that the data must be made anonymous in addition to it being de-identified. The clinical trials data can be considered anonymized if it cannot be linked back to the subject in any way. So, it is insufficient to simply scramble a subject identifier. To anonymize the data you must scramble the subject identifier and then discard the key that links the new subject identifier with the original subject identifier. Data "anonymization" is an IRB construct that originated from the National Bioethics Advisory Committee. The concept of anonymization can be found in "RESEARCH INVOLVING HUMAN BIOLOGICAL MATERIALS: ETHICAL ISSUES AND POLICY GUIDANCE" found at http://www.georgetown.edu/research/nrcbl/nbac/hbm.pdf (last accessed January 28, 2006). The concept of anonymization with respect to the Common Rule is spelled out in some detail in the "Guidance on Research Involving Coded Private Information or Biological Specimens" which can be found at the Department of Health and Human Services Office for Human Research Protections at http://www.hhs.gov/ohrp/humansubjects/guidance/cdebiol.pdf (last accessed January 28, 2006).

HIPAA PRIVACY RULE

45CFR Parts 160 and 164, also known as the HIPAA Privacy Rule that went into effect on April 14, 2003, is part of the Health Insurance Portability and Accountability Act of 1996 and has an impact on clinical trials databases when they are used for research. Because HIPAA included tools to make it easier for providers to bill insurance companies for health care, the Privacy Rule was included to make sure that there was adequate protection of your private information. According to 45CFR160.102, the Privacy Rule applies to "covered entities" which are:

> (1) A health plan.
> (2) A health care clearinghouse.
> (3) A health care provider who transmits any health information in electronic form in connection with a transaction covered by this subchapter.[6]

Although you may not work under a "covered entity", keep in mind that you may need to exchange data with a covered entity and therefore this regulation will come into play. Also understand that data submitted to the NIH under a data sharing plan will generally need to be HIPAA Privacy Rule compliant.

To do research on data under the HIPAA Privacy Rule, you generally need to either have an authorization from the subject to use their protected health information or you need to have your research reviewed and granted a waiver of authorization by an IRB or Privacy Board. [There are Privacy Rule exceptions for research on decedents, reviews preparatory to research, and grandfathered data that you can read more about in the Privacy Rule and at http://privacyruleandresearch.nih.gov/.] Initially, clinical trials databases are covered for research use of their protected health information from a HIPAA authorization included during the informed consent process and the original IRB approved protocol. However, for research not covered by an authorization of the use of the protected health information granted under the original protocol, you can pursue one of the following approaches:

1. Get a waiver of authorization from an IRB or Privacy Board.
2. Have a qualified statistician certify and document that there is a very small risk that use of the protected health information could lead to a subject being identified.
3. Use "Safe Harbor" de-identified data sets of the clinical trials data. [Note that the HIPAA Privacy Rule "Safe Harbor" is not the same as the "Safe Harbor" framework defined by the United States in order to comply with the European Directive on Data Protection.]
4. Use "limited" data sets of the clinical trials data and enter into a data use agreement with the new researcher.

All four approaches are valid research approaches for data use under the HIPAA Privacy Rule, but I will focus in this paper primarily on option #3, which is the "Safe Harbor" approach to using data in a HIPAA Privacy Rule compliant way. The reason for doing this is that it is the most direct way that you can ensure HIPAA Privacy Rule compliance. Option #1 above requires obtaining permission of a waiver of authorization from an IRB or Privacy Board. Option #2 is simply not as well defined as the "Safe Harbor" method and therefore seems inherently a bit more risky. Option #4 is a subset of the "Safe Harbor" method which may be necessary especially if some protected health information is required by the researcher, but it also requires the additional burden of creating and maintaining a data use agreement with the researcher.

"SAFE HARBOR" METHOD OF DE-IDENTIFICATION

"De-identification" of protected health information (PHI) is defined broadly in 45CFR164.514(a) as it "does not identify an individual and with respect to which there is no reasonable basis to believe that the information can be used to identify an individual is not individually identifiable health information."[7] The "Safe Harbor" method means of protected health information data de-identification is defined in 45CFR164.514(b)(2) which states that the following things need to be removed from the

data:

> The following identifiers of the individual or of relatives, employers, or household members of the individual, are removed:
>
> (A) Names;
>
> (B) All geographic subdivisions smaller than a State, including street address, city, county, precinct, ZIP code, and their equivalent geocodes, except for the initial current publicly available data from the Bureau of the Census:
>
> (1) The geographic unit formed by combining all ZIP codes with the same three initial digits contains more than 20,000 people; and
>
> (2) The initial three digits of a ZIP code for all such geographic units containing 20,000 or fewer people is changed to 000.
>
> (C) All elements of dates (except year) for dates directly related to an individual, including birth date, admission date, discharge date, date of death; and all ages over 89 and all elements of dates (including year) indicative of such age, except that such ages and elements may be aggregated into a single category of age 90 or older;
>
> (D) Telephone numbers;
>
> (E) Fax numbers;
>
> (F) Electronic mail addresses;
>
> (G) Social security numbers;
>
> (H) Medical record numbers;
>
> (I) Health plan beneficiary numbers;
>
> (J) Account numbers;
>
> (K) Certificate/license numbers;
>
> (L) Vehicle identifiers and serial numbers, including license plate numbers;
>
> (M) Device identifiers and serial numbers;
>
> (N) Web Universal Resource Locators (URLs);
>
> (O) Internet Protocol (IP) address numbers;
>
> (P) Biometric identifiers, including finger and voice prints;
>
> (Q) Full face photographic images and any comparable images; and
>
> (R) Any other unique identifying number, characteristic, or code, except as permitted by paragraph (c) of this section; and
>
> The covered entity does not have actual knowledge that the information could be used alone or in combination with other information to identify an individual who is a subject of the information.[8]

The good news is that most traditional clinical trials do not collect much protected health information. If the protected health information were collected, the majority of the above 18 items could just be deleted from the data to de-identify it. However, dates identified above in item “C” are critical in clinical trials research and the databases are full of them. One way to deal with dates is to create a days from event variable to replace them. For instance, you could replace all date fields with the number of days from randomization, which would still maintain the chronology of events while de-identifying precisely when they happened. The medical record indicated in item “H” could be the subject number in the database that maps back to the CRF. The HIPAA Privacy Rule allows you to assign a new code for the patient, but it cannot be derived from the subject number or the subject’s data in any way. So, you could just assign a random subject identifier to each subject in your database.

DIFFERENCE BETWEEN HIPAA DE-IDENTIFIED AND “COMMON RULE” ANONYMIZED

- The HIPAA Privacy Rule allows you to create and keep (and protect) a key that maps your old subject identifiers to the new de-identified identifiers. The “Common Rule” states that the key must be destroyed for the data to be anonymous.
- The “Common Rule” allows for new de-identified subject identifiers to be derived from old subject identifiers but the HIPAA Privacy Rule prohibits this.
- The “Common Rule” states that anonymized data should not contain any identifiable private information, but the HIPAA Privacy Rule lists 18 more specific data types to be excluded. The “Common Rule” generally allows for ZIP codes and dates to remain in anonymized data.

OTHER RESTRICTIONS ON DATA SHARING

There may be other restrictions on data de-identification with regards to some clinical trials. Some divisions of the NIH, such as the NHLBI and the NIMH, have their own data sharing policies. These policies are generally very much like the HIPAA Privacy Rule, but they may add further restrictions on the data that must be considered. Also, federally funded clinical trials involving alcohol and drug abuse data fall under the jurisdiction of federal law 42CFR2, “CONFIDENTIALITY OF ALCOHOL AND DRUG ABUSE PATIENT RECORDS”. Finally, and by no means least important, is the proprietary nature of about half of all clinical trials conducted. Pharmaceutical companies have the legal rights to their data and they may not wish to share the data even after they have been de-identified and anonymized.

APPROACHES TO USING CLINICAL TRIALS DATA FOR RESEARCH

Now that the regulations have been summarily reviewed, how should you go about using clinical trials data for research?

Assuming that you don't already have patient authorization and IRB approval for the research, should you anonymize, de-identify, use a limited data set approach, or get IRB consent for the waiver of authorization? That depends in part on business needs. For instance, if you are looking at researching data from a clinical trial that will likely not be used for further research purposes, then perhaps obtaining an IRB approved protocol for the research is the most cost effective approach. However, if the clinical trials data could be used for many different future research studies then perhaps an anonymized and de-identified clinical trial database would be more efficient. As with most things, I don't believe that one size fits all and the potential future value of the clinical trial database should be considered when deciding on the best approach to how to prepare the data for research.

A SAS BASED APPROACH TO DE-IDENTIFICATION AND ANONYMIZATION

This section presents an example of a SAS based effort to data de-identification and anonymization. To make this paper more readable, all of the SAS code discussed has been attached at the very end of this paper. As a prerequisite to using these macros, I assume you have a folder full of clinical trial data sets, you know what your primary patient identifier variable is, and you know of a key trial reference date that all subjects would have (such as randomization date).

%DEIDNUM FOR SUBJECT IDENTIFIERS

The %deidnum SAS macro is designed so that you can drop the original subject number identifier and still be able to link a subject's data across all SAS data sets in a SAS libref. For every unique subject number identifier in a SAS libref, %deidnum creates a unique 6 digit random number in a variable called "deidnum". The macro then takes all data sets and attaches the "deidnum" variable to them. Finally, a data set called "deidnum" is created that contains only the original subject number identifier and the "deidnum" variable. This "deidnum" key data set can be destroyed if you wish to anonymize data so that there is no way to link the original subject number of subjects to the new "deidnum" identifiers, or you can store the "deidnum" file away in escrow if you just want to de-identify your data. If you wish to anonymize your data, you may wish to set the random number generator seed in %deidnum to "0" so that the starting seed is based on the time when the program is run. That would help further obfuscate how the "deidnum" variable was created since the source code would not contain the direct algorithm for how the "deidnum" variable was created.

%DATESTOSTUDYDAY FOR DATES

The %datestostudyday SAS macro takes all SAS formatted date and datetime variables for a SAS libref and converts them to a subject specific days from reference date equivalent. The macro needs you to specify a SAS date variable as the reference date. All SAS dates in the target libref are then converted to integer offsets and SAS datetimes are converted to floating point number offsets. The newly created studyday variable will have the same name as the old date variable and the SAS label for the variable with be prefixed by, "Studyday of ". So, if the SAS label was "Birth Date" the new label would be "Studyday of Birth Date". Please note that this macro requires that all dates to be de-identified must have a SAS date or datetime format. This macro knows all date and datetime formats through SAS 9.1.3.

%DROPVARS TO DROP VARIABLES

The %dropvars SAS macro simply drops all listed variables that exist anywhere across an entire SAS libref. The only exception to this is that %dropvars ignores the "deidnum" SAS data set because I did not want to drop the original subject identifier from that data set in case that "deidnum" was to be a de-identification key data set to be stored in escrow.

SAMPLE RUN

I begin with a small set of sample data sets that can be described by the following PROC CONTENTS and PROC PRINT information:

DEMOG Source Dataset Contents

Variable	Type	Format	Label
Subjid	Char		Subject Identifier
Invsite	Char		Site
Birthdt	Num	DATE9.	Birth Date
Height	Num		Height (in)
Randodt	Num	DATE9.	Randomization Date
Ssn	Char		SSN
Weight	Num		Weight (lbs)

INFUSION Source Dataset Contents

Variable	Type	Format	Label
subjid	Char		Subject Identifier
invsite	Char		Site
infusdt	Num	DATETIME19.	Drug Infusion Datetime

PRINT of DEMOG

subjid	invsite	randodt	birthdt	height	weight	ssn
100-001	100	01JAN2006	22JAN1964	60.1	120.3	111-11-1111
102-002	102	01FEB2006	24JUL1997	48.3	60.4	222-22-2222
103-003	103	01MAR2006	11APR1999	42.9	50.2	333-33-3333

PRINT of INFUSION

subjid	invsite	infusdt
100-001	100	01FEB2006:12:00:00
102-002	102	01MAR2006:00:00:00
103-003	103	01APR2006:06:00:00

I then identify “subjid” in the “DEMOG” data set as my original subject number identifier and the “randodt” variable in the “DEMOG” data set as my reference date to calculate study days from. I call the three macros as follows:

```
**** CREATE DEIDENTIFIED SUBJECT ID.;
%deidnum(inlibref=source,         /* source libref */
         outlibref=deid,          /* target libref */
         subjid=subjid,           /* original subject identifier */
         seed=0)                  /* starting seed for random number generation */
**** CREATE DEIDENTIFIED DATES.;
%datestostudyday(inlibref=deid,   /* source libref */
                 outlibref=deid,  /* target libref */
                 subjid=subjid,   /* subject identifier */
                 refset=demog,    /* reference date dataset identifier */
                 refvar=randodt)  /* reference date variable */
**** DROP IDENTIFYING DATA FIELDS.;
%dropvars(inlibref=deid,               /* source libref */
          outlibref=deid,              /* target libref */
          dropvars=subjid invsite ssn)  /* variables to drop */
```

Which results in the following PROC CONTENTS from the “deid” libref:

DEIDNUM Dataset Contents		
Variable	**Type**	**Label**
deidnum	Num	Deidentified Subject Identifier
subjid	Char	Subject Identifier

DEMOG De-Identified Dataset Contents		
Variable	**Type**	**Label**
deidnum	Num	Deidentified Subject Identifier
birthdt	Num	Studyday of Birth Date
randodt	Num	Studyday of Randomization Date
weight	Num	Weight (lbs)
height	Num	Height (in)

INFUSION De-Identified Dataset Contents			
Variable	**Type**	**Format**	**Label**
deidnum	Num		Deidentified Subject Identifier
infusdt	Num	13.5	Studyday of Drug Infusion Datetime

Which also results in the following PROC PRINT from the “deid” libref:

PRINT of DEIDNUM	
deidnum	**subjid**
1702	102-002
71796	100-001
81284	103-003

PRINT of DEMOG					
deidnum	**randodt**	**height**	**weight**	**birthdt**
1702	0	48.3	60.4	-3114
71796	0	60.1	120.3	-15320
81284	0	42.9	50.2	-2516

PRINT of INFUSION	
deidnum	**infusdt**
1702	28.00000
71796	31.50000
81284	31.25000

If I wanted to anonymize the database, I could just destroy the “DEIDNUM” data set and I would no longer be able to link “deidnum” back to “subjid”. Note how the fields to be dropped have been dropped across all data sets and how the SAS formatted dates and datetime variables have been converted to “study day” equivalents. “randodt” is always zero for all patients because it was the reference date specified. The above SAS code doesn’t address the “Safe Harbor” requirement 45CFR164.514(b)(2) for item C, but it would be easy enough to truncate age (and birth date study day) at 90 years either with a DATA step or another generic SAS macro call.

HAS DE-IDENTIFYING MY DATA REDUCED ITS VALUE?

You need to consider the value of the resulting database when de-identifying your database. For instance, say you decide that investigator number could be used to identify subjects because some of the investigator sample sizes are small and unique and those sample sizes have been publicly published already. So you decide to drop investigator number from your de-identified database. However, you know that researchers will want to use investigator site as a covariate in their analysis of the de-identified database. This problem can possibly be handled by using the HIPAA “limited dataset” approach, where the researcher is given “limited datasets” which drop obvious subject identifiers and they sign a data use agreement with the data provider. Please refer to the HIPAA Privacy Rule for further details on “limited datasets” and data use agreements.

IS MY DATA TRULY DE-IDENTFIED?

Although the previous example works towards de-identification of clinical trials data, the question arises whether clinical trial data can be truly and completely de-identified. The HIPAA Privacy Rule concludes section 45CFR164.514(b)(2) by saying that, "Any other unique identifying number, characteristic, or code" should be removed and that, "The covered entity does not have actual knowledge that the information could be used alone or in combination with other information to identify an individual who is a subject of the information."[9] This is basically a catch-all set of requirements that says even though you may drop the listed data items in the "Safe Harbor" method that you still must be sure that you censor any other data that may identify a subject. For instance, site number can be identifying if you have a very small site in a multi-center trial. Comment text and free text fields in general could contain subject identifying information. Unique adverse events and verbatim terms could be identifying.

So, no matter what automated tools are brought to bear on the task of de-identifying clinical trials data, I strongly suggest that they not be used in isolation. A qualified statistician, clinician, or data analyst should review the de-identified clinical trial database to ensure that no individually identifiable health information remains.

CONCLUSION

This paper discusses the regulations around clinical trial data de-identification and anonymization and provides an example of a SAS based approach to de-identify and anonymize a clinical trial database. I hope that those that need to provide this type of de-identified data find regulatory references in this paper that may be of use and that some ideas can be gathered from the sample SAS programs provided. With recent events surrounding regulated drugs and patient safety, the NIH requirement for data sharing, and the advances in information technology and the Internet, I feel that data sharing and de-identification of clinical trials data will grow ever important in the coming years as a fundamental pillar towards improving public healthcare.

REFERENCES

[1] National Institutes of Health. "FINAL NIH STATEMENT ON SHARING RESEARCH DATA." February 26, 2003. Available from http://grants.nih.gov/grants/guide/notice-files/NOT-OD-03-032.html accessed on January 28, 2006.

[2] Duke Clinical Research Institute home page at http://www.dcri.duke.edu/ accessed on January 28, 2006.

[3], [4], [5] Code of Federal Regulations. "TITLE 45 - PUBLIC WELFARE - DEPARTMENT OF HEALTH AND HUMAN SERVICES - PART 46 - PROTECTION OF HUMAN SUBJECTS". October 1, 2005. Available from http://www.gpoaccess.gov/cfr/index.html accessed on January 28, 2006.

[6], [7], [8], [9] Code of Federal Regulations. "TITLE 45 - PUBLIC WELFARE **-** DEPARTMENT OF HEALTH AND HUMAN SERVICES **-** Parts 160 and 164 - Standards for Privacy of Individually Identifiable Health Information; Final Rule**"**. October 1, 2005. Available from http://www.gpoaccess.gov/cfr/index.html accessed on January 28, 2006.

ACKNOWLEDGMENTS

Thanks and warm regards go to Dr. Doc Muhlbaier and Toddie Stewart of the Duke Clinical Research Institute for their review of the text in this manuscript and their guidance on patient data privacy related issues.

RECOMMENDED READING

[1] "NIH Data Sharing Policy" and related documentation found at http://grants.nih.gov/grants/policy/data_sharing/ accessed on January 28, 2006.

[2] The US Code of Federal Regulations including the HIPAA Privacy Rule at 45CFR Parts 160 and 164, the "Common Rule" at 45CFR46 found at http://www.gpoaccess.gov/cfr/ accessed on January 28, 2006.

[3] "HIPAA Privacy Rule, Information for Researchers" found at http://privacyruleandresearch.nih.gov/ accessed on January 28, 2006.

CONTACT INFORMATION

Any comments that you may have are valued and encouraged. Contact the author at:

Jack Shostak
Duke Clinical Research Institute
Address P.O Box 17969
City state ZIP Durham, NC 27715
Work Phone: 919-668-8832
Email: jack.shostak@duke.edu

SAS SOURCE CODE FOLLOWS …

```
**** DEIDNUM SAS MACRO ************************************************
**** This SAS macro takes a set of SAS datasets and assigns a random unique
**** 6 digit subject identifier called DEIDNUM to all datasets.  A dataset called
**** DEIDNUM is also created which can be used to re-identify subjects later.
****
**** PARAMETERS:  INLIBREF = Libref pointing to where the datasets come from.
****              OUTLIBREF = Libref pointing to where to write the new datasets.
****              SUBJID = Unique subject identifier found in INLIBREF datasets.
****              SEED = Starting seed for random number generator for DEIDNUM.
****                     Use seed=0 if you wish to anonymize datasets.
**** NOTES:  To anonymyze datasets, you need to run this macro with seed=0 and
****         destroy the DEIDNUM dataset when de-identification process is complete.
****         This macro assumes that you do not already have a dataset or
****         a variable called DEIDNUM in your source data.  If the source dataset
****         does not have the SUBJID variable, then a copy will not be sent to
****         the OUTLIBREF and a warning will be provided for that file.
************************************************************************;
%macro deidnum(inlibref=,outlibref=,subjid=,seed=0);
options ls=256;

**** GET ALL DATASET NAMES THAT HAVE THE SUBJID VARIABLE.;
proc sql noprint;
    select unique memname into :datasets separated by '|'  from dictionary.columns
    where upcase(libname)="%upcase(&inlibref)" and upcase(memtype)="DATA" and
          upcase(name)="%upcase(&subjid)";
    select count(unique memname) into :dataset_count from dictionary.columns
    where upcase(libname)="%upcase(&inlibref)" and upcase(memtype)="DATA" and
          upcase(name)="%upcase(&subjid)";
quit;

**** SET ALL DATASETS TOGETHER TO GET UNIQUE SUBJECTS.;
data deidnum;
    set %do i = 1 %to &dataset_count;
            %upcase(&inlibref).%scan(&datasets,&i,"|")(keep=&subjid)
        %end;  ;

    keep &subjid;
run;

proc sort
    data=deidnum
    nodupkey;
        by &subjid;
run;

**** CREATE DEIDNUM WHICH IS A 6 DIGIT DEIDENTIFIED SUBJECT ID.;
data deidnum;
    set deidnum;
        by &subjid;

        keep &subjid deidnum;
        deidnum = floor(100000 * ranuni(&seed));

        **** CREATE LABEL FOR DEIDNUM;
        call symput("deidnumlabel","Deidentified " || put(vlabel(&subjid),$243.));
run;

**** ENSURE THAT DEIDNUM IS UNIQUE.;
%let deidnum=NO;
%do %until(&deidnum=YES);
    proc sort
        data=deidnum;
```

```
            by deidnum;
    run;

    %let deidnum=YES;
    data deidnum;
        set deidnum;
            by deidnum;

            if not(first.deidnum and last.deidnum) then
                do;
                    call symput('deidnum','NO');
                    if last.deidnum then
                        deidnum = deidnum + 1;
                end;
    run;
%end;

**** ASSIGN DEIDNUM VARIABLE LABEL.;
proc datasets;
    modify deidnum(label="Deidentified patient identifier file");
    label deidnum = "&deidnumlabel";
    copy out=&outlibref;
         select deidnum;
run;

**** JOIN DEIDNUM WITH ALL OTHER DATASETS.;
proc sort
    data=deidnum;
        by &subjid;
run;

**** GET ALL DATASET NAMES AND COUNT AND PLACE INTO MACRO PARAMETERS.;
proc sql noprint;
    select memname into :datasets separated by '|'  from dictionary.members
      where upcase(libname)="%upcase(&inlibref)" and upcase(memtype)="DATA";
    select count(memname) into :dataset_count from dictionary.members
      where upcase(libname)="%upcase(&inlibref)" and upcase(memtype)="DATA";
quit;

%do i = 1 %to &dataset_count;
    **** DETERMINE IF SUBJECT IDENTIFIER IS PRESENT.;
    %let SubjectIdPresent=NO;
    data _null_ ;
        set sashelp.vcolumn ;
        where upcase(libname) = upcase("&inlibref") and
              upcase(memname) = upcase("%scan(&datasets,&i,'|')");

        if upcase(name) = upcase("&subjid") then
           call symput('SubjectIdPresent','YES');
    run;

    %if &SubjectIdPresent=YES %then
        %do;
            proc sort
                data=&inlibref..%scan(&datasets,&i,"|")
                out=%scan(&datasets,&i,"|");
                    by &subjid;
            run;

            data %scan(&datasets,&i,"|");
                merge %scan(&datasets,&i,"|") (in=_indomain_)
                      deidnum;
                    by &subjid;
```

```
                    **** KEEP ONLY RECORDS IN THE SOURCE DATASET.;
                    if _indomain_;
            run;

            proc sort
                data=%scan(&datasets,&i,"|")
                out=&outlibref..%scan(&datasets,&i,"|");
                    by deidnum;
            run;
        %end;
    %else %if &SubjectIdPresent=NO %then
        %do;
            data _null_;
                put "WARN" "ING: Dataset %scan(&datasets,&i,"|") does not have key
identifier &subjid and will not be copied to &outlibref";
            run;
        %end;
%end;

%mend deidnum;


**** DATESTOSTUDYDAY SAS MACRO ************************************************
**** This SAS macro takes a set of SAS datasets and replaces variables formatted
**** as SAS dates or datetimes as a deidentified offset from a reference date.
**** The offset is calculated as the source date minus the reference date.  For
**** dates the result is an integer.  For datetimes the result is a floating point
**** number formatted with 5 significant digits.
****
**** PARAMETERS:  INLIBREF = Libref pointing to where the datasets come from.
****              OUTLIBREF = Libref pointing to where to write the new datasets.
****              SUBJID = Unique subject identifier found in INLIBREF datasets.
****              REFSET = SAS dataset from INLIBREF containing REFVAR.
****              REFVAR = SAS variable name for the reference date.  For example,
****                       RANDODT (randomization date) or DOSDT (dosing date).
**** NOTES:   If the REFVAR is not unique within a SUBJID, then this program will
****          pick the first nonmissing date.  For this code to work, the SAS dates
****          and datetimes must have a valid SAS format.  This programs knows of
****          SAS formats up through SAS v9.1.3.  If a source dataset does not have
****          the SUBJID variable, then dates will not be processed.
*******************************************************************************;

%macro datestostudyday(inlibref=,outlibref=,subjid=,refset=,refvar=);
options ls=256;

**** GET REFERENCE DATE;
proc sort
    data=&inlibref..&refset(keep=&subjid &refvar)
    out=_ReferenceDate_;
        by &subjid &refvar;
run;

**** ENSURE A SINGLE REFERENCE DATE RECORD PER SUBJECT;
data _ReferenceDate_;
    set _ReferenceDate_;
        by &subjid &refvar;

        where &refvar ne .;

        keep &subjid _ReferenceDate_;

        if not (first.&subjid and last.&subjid) then
            put "NOTE: TAKING THE FIRST DATE AS REFERENCE DATE " &subjid= &refvar=;
        if first.&subjid;
```

```
        _ReferenceDate_ = &refvar;
run;

**** GET DATASET, VARIABLE, SAS FORMAT, AND SAS LABEL FROM SOURCE DATA.;
proc sql
    noprint;
    create table _DatesToConvert_ as
        select memname, name, format, label as rawlabel
        from sashelp.vcolumn
        where upcase(libname)="%upcase(&inlibref)"
        order by memname, name;
quit;

**** KEEP ONLY VARIABLES THAT ARE FORMATTED AS SAS DATES OR DATETIMES.;
data _DatesToConvert_;
    set _DatesToConvert_;
        by memname name;

        **** DETERMINE DATE OR DATETIME FIELD.  FORMATS VALID THROUGH SAS V9.1.3;
        if upcase(compress(format,"0123456789.")) in
            ('DATE','DAY','DDMMYY','DOWNAME','EURDFDD','EURDFDE','EURDFDN',
             'EURDFDWN','EURDFMN','EURDFMY','EURDFWDX','EURDFWKX','HDATE',
             'HEBDATE','JULDAY','JULIAN','MINGUO','MMDDYY','MMYY','MONNAME',
             'MONTH','MONYY','NENGO','NLDATE','NLDATEMN','NLDATEW','NLDATEWN',
             'NLDATMW','PDJULG','PDJULI','QTR','QTRR','WEEKDATE','WEEKDATX',
             'WEEKDAY','WEEKU','WEEKV','WEEKW','WORDDATE','WORDDATX','YEAR',
             'YYMM','YYMMDD','YYMON','YYQ','YYQR') then
            _DateType_ = 'DATE    ';
        else if upcase(compress(format,"0123456789.")) in
            ('DATEAMPM','DATETIME','DTDATE','DTMONYY','DTWKDATX','DTYEAR',
             'DTYYQC','EURDFDT','NLDATM','NLDATMAP','NLDATMTM','TOD') then
            _DateType_ = 'DATETIME';

        **** KEEP ONLY DATE OR DATETIME VARIABLES.;
        if _DateType_ ne '';

        **** DEFINE NEW STUDYDAY VARIABLE LABELS.;
        length label $ 256;
        label = "Studyday of " || trim(tranwrd(translate(rawlabel,"'",'"'),"'","''"));
run;

**** PUT ALL DATASETS NAMES AND COUNT OF THEM INTO MACRO PARAMETERS.;
proc sql noprint;
    select distinct memname into :datasets separated by '|'  from _DatesToConvert_;
    select count(distinct memname) into :dataset_count from _DatesToConvert_;
quit;

**** LOOP THROUGH DATASETS AND CONVERT DATE FIELDS TO STUDYDAY EQUIVALENTS.;
%do i = 1 %to &dataset_count;

    **** PUT ALL VARIABLE NAMES, LABELS, AND DATE TYPES INTO MACRO PARAMETERS.;
    proc sql noprint;
        select name into :datevariables separated by '|'
            from _DatesToConvert_
            where upcase(memname)="%upcase(%scan(&datasets,&i,"|"))";
        select label into :datelabels separated by '|'
            from _DatesToConvert_
            where upcase(memname)="%upcase(%scan(&datasets,&i,"|"))";
        select _DateType_ into :datetypes separated by '|'
            from _DatesToConvert_
            where upcase(memname)="%upcase(%scan(&datasets,&i,"|"))";
        select count(name) into :datevariable_count
            from _DatesToConvert_
```

```
            where upcase(memname)="%upcase(%scan(&datasets,&i,"|"))";
    quit;

    **** DETERMINE IF SUBJECT IDENTIFIER IS PRESENT.;
    %let SubjectIdPresent=NO;
    data _null_ ;
        set sashelp.vcolumn ;
        where upcase(libname) = upcase("&inlibref") and
              upcase(memname) = upcase("%scan(&datasets,&i,'|')");

        if upcase(name) = upcase("&subjid") then
           call symput('SubjectIdPresent','YES');
    run;

    %if &SubjectIdPresent=YES %then
      %do;
        proc sort
          data=&inlibref..%scan(&datasets,&i,"|")
          out=%scan(&datasets,&i,"|");
            by &subjid;
        run;

        **** JOIN REFERENCE DATE WITH DATASET.;
        data &outlibref..%scan(&datasets,&i,"|");
          merge _ReferenceDate_
              %scan(&datasets,&i,"|")(in = _indomain_ rename=(
              %do j = 1 %to &datevariable_count;
                 %scan(&datevariables,&j,"|") = _TEMP%scan(&datevariables,&j,"|")
              %end;));
            by &subjid;

            if _indomain_;

            drop _ReferenceDate_;

            **** FOR EACH DATE FIELD, CALCULATE THE STUDYDAY AND DROP OLD DATE VAR.;
            %do j = 1 %to &datevariable_count;
               %if %scan(&datetypes,&j,"|") = DATE %then
                  %do;
                     %scan(&datevariables,&j,"|") =
                           _TEMP%scan(&datevariables,&j,"|") - _ReferenceDate_;
                     format %scan(&datevariables,&j,"|");
                  %end;
               %else %if %scan(&datetypes,&j,"|") = DATETIME %then
                  %do;
                     %scan(&datevariables,&j,"|") =
                          (_TEMP%scan(&datevariables,&j,"|")/86400) - _ReferenceDate_;
                     format %scan(&datevariables,&j,"|") 13.5;
                  %end;

               label %scan(&datevariables,&j,"|") = "%scan(&datelabels,&j,'|')";
               drop _TEMP%scan(&datevariables,&j,"|");
            %end;
        run;
      %end;
    %else %if &SubjectIdPresent=NO %then
      %do;
          data _null_;
            put "WARN" "ING: Dataset %scan(&datasets,&i,"|") does not have key identifier
&subjid so dates will not be deidentified";
          run;
      %end;
%end;
%mend datestostudyday;
```

```
**** DROPVARS SAS MACRO ***********************************************************
**** This SAS macro takes a set of SAS datasets and drops all variables requested.
****
**** PARAMETERS:  INLIBREF = Libref pointing to where the datasets come from.
****              OUTLIBREF = Libref pointing to where to write the new datasets.
****              DROPVARS = Space delimited list of SAS variable names to be
****                         dropped from the OUTLIBREF library.  The macro assumes
****                         that you have at least one variable to drop.
**** NOTES: This macro ignores any dataset called DEIDNUM when dropping variables.
*************************************************************************;
%macro dropvars(inlibref=,outlibref=,dropvars=);
options ls=256;

**** GET ALL DATASETS AND PLACE INTO MACRO PARAMETERS;
proc sql noprint;
    select memname into :datasets separated by '|'  from dictionary.members
      where upcase(libname)="%upcase(&inlibref)" and upcase(memtype)="DATA" and
            upcase(memname) ne "DEIDNUM";
    select count(memname) into :dataset_count from dictionary.members
      where upcase(libname)="%upcase(&inlibref)" and upcase(memtype)="DATA" and
            upcase(memname) ne "DEIDNUM";
quit;

%do i = 1 %to &dataset_count;
    **** CREATE VARIABLE DROP STRING;
    data _null_;
        %let j=1;
        dropstring =
        %do %until(%scan(&dropvars,&j,' ')= );
            %let op=%sysfunc(open(&inlibref..%scan(&datasets,&i,'|')));
            %if %sysfunc(varnum(&op, %scan(&dropvars,&j,' '))) > 0 %then
                "%scan(&dropvars,&j,' ') " || ;
            %let rc=%sysfunc(close(&op));
            %let j=%eval(&j+1);
        %end;
        " ";

        if dropstring ne '' then
            call symput("dropstring","drop " || dropstring);
        else
            call symput("dropstring"," ");
    run;

    **** IF DROP VARIABLES EXIST, DROP THEM.  OTHERWISE SIMPLY COPY FILE.;
    data &outlibref..%scan(&datasets,&i,"|");
       set &inlibref..%scan(&datasets,&i,"|");

            %if &dropstring ne  %then
                &dropstring;;
    run;
%end;

%mend dropvars;
```