

Data requests, quickly

Elisa L. Priest
The Cooper Institute, Dallas, Texas
University of North Texas School of Public Health, Fort Worth, Texas

ABSTRACT

As completion of a research study nears, investigators want to begin preliminary analysis of their data. Often, a data manager receives multiple data requests at once from different collaborators. I will give examples of code developed for a clinical trial that allows a data manager to produce complex datasets from multiple data tables with useful excel data dictionaries, and fulfill customized data requests with minimal effort during the hectic final stage of a study.

INTRODUCTION

Our organization primarily conducts NIH funded clinical trials. These trials range in size from twenty persons to hundreds of persons. Different types of data are collected from participants at many stages of the trial and include forms filled out by study staff, questionnaires completed by the participants on site or at home, measurement data collected during laboratory testing, and data collected during the intervention. In addition, study staff continually monitor participants' progress through the trial and record information such as group assignment, visit completion, participant chart location, and contact information.

Data management staff use PC SAS to produce varying reports and analyses on combinations of variables from all of the above data sources. In an effort to stay organized, we developed a system of SAS files for our project that allows a data manager to produce these datasets with a minimal amount of new code.

GOALS

We had several goals in the development of this system. Primarily, we needed to reduce the amount of coding time that it took to produce customized datasets with code books from the project data. Similarly, we wanted to reduce the amount of redundant code written for the project. Finally, we wanted to track all data requests that were completed and re-create any data request at any given time.

ORGANIZATION

Folder Organization

The first key component of our system is the organization of our project folders. All folders and files are kept on a shared drive so that authorized staff may perform data management activities. We organize our SAS files under the project name to facilitate archiving of all relevant files after trial completion. Our data management folder structure consists of two primary divisions: SAS, which stores all .SAS files, and Data,

which stores all the SAS datasets used in the project. (Figure 1) Our SAS files are further divided into folders by relevance to the study: Data Cleaning, Data Requests, Dataset Creation, Descriptive, Formats, Intervention, Measurements, and Questionnaires. Data Cleaning, Data Requests, and Dataset Creation all create and manipulate data from multiple sources. Data Cleaning files are used for monitoring the project and data quality, Data Requests create data sets for project collaborators, and Dataset Creation files create and update the main analysis dataset for the project.

The remaining folders store code that create data tables from one source of data. This is the key to reducing redundant SAS code for dataset creation. For example, the Descriptive folder holds four files: 1) general participant tracking information 2) Project Group assignment 3) Chart tracking and 4) Descriptive information from the participant screening. Each of these four files produces one data table that can be combined to produce analysis datasets.

Another folder of note is the Formats folder. This folder contains one file which defines all the libraries for the project and another which defines all the filenames for the project. This is another key to reducing time spent coding because all the library and file pathways can be defined and updated in one location. The code below is an excerpt from that assigns all the files for the Descriptive folder and a file from the Measurements folder.

```
*Project Descriptive Folder;
FILENAME TRACKING ' \PROJECT\SAS\DESCRIPTIVE\TRACKING.SAS';
FILENAME RANDOMIZ ' \PROJECT\SAS\DESCRIPTIVE\RANDOMIZATION.SAS';
FILENAME CHART ' \PROJECT\SAS\DESCRIPTIVE\CHARTTRACKING.SAS';
FILENAME PHONE ' \PROJECT\SAS\DESCRIPTIVE\SCREENING.SAS';

*Project Measurements Folder;
FILENAME BLOOD0 ' \Project\SAS\MEASUREMENTS\BLOOD0.SAS';
```

Finally, the resulting permanent data files are stored in either the Project\SAS\DATA\Datasets folder or the \Data Requests folder depending on the data.

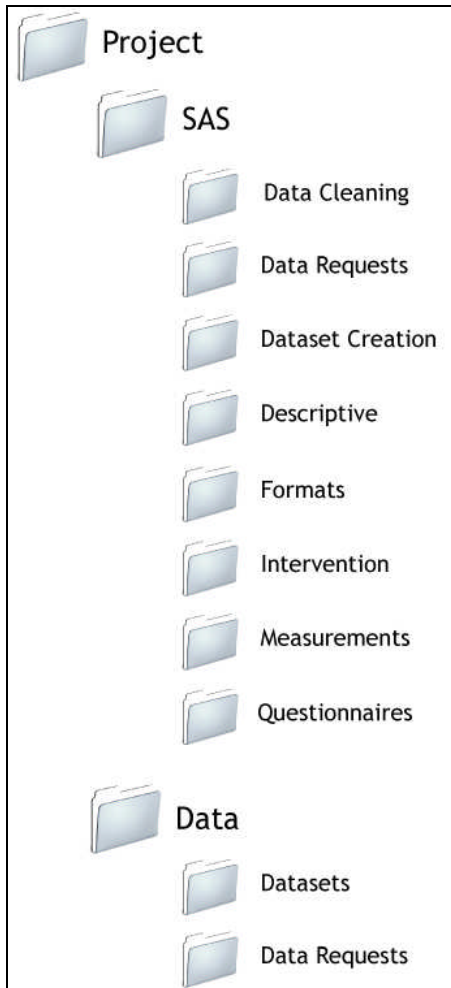


Figure 1. Folder Organization

File Organization

All SAS files are kept organized by placing them into the appropriate folders and by using meaningful names. In addition, each file contains a header with information about the creation of the code, the input and output data, and a table of contents which summarizes the major sections of the code. Each .sas file in the Descriptive, Formats, Intervention, Measurements, and Questionnaires folders creates temporary data sets complete with variable labels. This ensures that the data set is a complete self-contained module that is updated from the original data each time the code is run. The code below is an example header from the file blood0.sas in the measurements folder. This file is responsible for creating a dataset of all the blood data from the project.

```
*****
*****
TITLE: PROJECT DATASET: BLOOD0
FOLDER: MEASUREMENTS\BLOOD0.SAS
AUTHOR: Elisa L. Priest
CREATED: FRIDAY, NOVEMBER 12 2004
LAST MODIFIED: Wednesday, July 19, 2006
```

```
INPUTS: ACCESS DATABASE "PROJECT\PROJECTDATABASE.mdb" TABLE: BLOODTEST
OUTPUTS: WORK.BLOOD0
TEMPORARY: WORK.BLOOD00
MACROS USED: NONE
```

Dataset Creation

The next key to meeting our data management goals is the creation of a main analysis dataset. This dataset facilitates the production and re-creation of the customized analysis datasets requested by study collaborators.

The main analysis dataset uses the following three lines of code below the header and table of contents to allow access to all the SAS files and libraries in the project.

```
****Create Main Dataset for Project*****;
*Include these 3 lines to access all Project SAS files;
FILENAME PROJECTFILE "\PROJECT\SAS\FORMATS\PROJECTFILES.SAS";
%INCLUDE PROJECTFILE; *defines all files*;
%INCLUDE PROJECTLIB; *defines all libraries*;
```

Next, we continue to use the %INCLUDE macro to call the different code modules. Since we are creating the main analysis dataset, we will call most of our SAS files and then merge them. For example, the following code excerpt includes the tracking information for our participants and the blood data mentioned before. This method is also applicable when creating highly customized datasets for reports or data cleaning.

```
%INCLUDE TRACKING;
%INCLUDE BLOOD0;
```

Next, we merge all of the temporary data sets created by the code modules to create the main analysis dataset. This dataset is saved in the DATASETS library at \Project\Data\Datasets and named PROJECTDATA with the date.

```
** PROJECT DATASET CREATION*****;
DATA WORK.PROJECTDATA0;
MERGE      WORK.TRACKING (IN=A)
          WORK.BLOOD ;
BY ID;
IF A;
RUN;

DATA DATASETS.PROJECTDATA07152006;
SET WORK.PROJECTDATA0;
RUN;
```

Finally, we create a main codebook to serve a reference for all collaborators. This codebook is created using PROC CONTENTS out to a SAS dataset which is exported to an EXCEL file. The EXCEL format allows collaborators who use an alternate statistical package to view the available variables. In addition, collaborators can request variables for an analysis data by cutting and pasting variable names directly from the EXCEL file to an email. This ensures that they receive the requested variables and saves time and guesswork for the data managers fulfilling the request.

```

** PROJECT CODEBOOK CREATION*****;
PROC CONTENTS DATA=WORK.PROJECTDATA0 VARNUM OUT=WORK.PROJECTLABELS0;
TITLE1 'PROJECT DATASET';
RUN;

DATA WORK.PROJECTLABELS;
SET WORK.PROJECTLABELS0;
KEEP VARNUM NAME LENGTH TYPE LABEL FORMAT FORMATL FORMATD;
RUN;

PROC SORT;
BY VARNUM;
RUN;

PROC EXPORT DATA= WORK.PROJECTLABELS
OUTFILE=
"PROJECT\SAS\DATA\DATASETS\PROJECTCODEBOOK07152006.xls"
DBMS=EXCEL2000 REPLACE;
RUN;

```

Data requests

After we have used modular SAS code to create a main analysis dataset and codebook for our project data, data requests are generally quick to fill. Once a collaborator has completed a data request form with the variable names copied directly from the EXCEL codebook, the variables are subsetting from the most updated main analysis dataset. In the following code, a request for four variables is filled from the main analysis dataset created on 07/15/2006. A permanent copy of the data request dataset is saved in the DATAREQ library at \Project\Data\Data Requests. This permanent copy allows us to re-create the data request from the original data. In addition, a data request codebook may be created and saved with the dataset using the method described previously.

```

****Data Request for 4 variables from Main Project Dataset***;
DATA WORK.DATAREQUEST;
SET DATASETS.PROJECTDATA07152006;
KEEP
ID STATUSFOLLOWUP STATUSINTERVENTION WITHDRAW
Run;

DATA DATAREQ.DATAREQUEST08152006;
SET WORK.DATAREQUEST;
RUN;

```

CONCLUSION

In conclusion, we have created a system for organizing PC SAS code for an NIH clinical trial. This method relies on basic file and folder organization and the use of modular SAS code to create a main analysis dataset and customized data requests. The modular code can also be used for creating datasets for reporting and data cleaning. In addition, we produce an EXCEL codebook to facilitate data requests. Our system has worked well to organize our growing project data which currently consists of 32 code modules that create 136 temporary datasets and a main analysis dataset with over 1300 variables.

*This research was supported by the National Heart, Lung, and Blood Institute grant HL66262.

CONTACT INFORMATION

Elisa L. Priest

Director, Clinical Trials Data Management

The Cooper Institute

12330 Preston Road

Dallas, TX 75230

972-341-3254

elisapriest@hotmail.com