The Super Genius Guide to Generating Dummy Data
Brian Varney, COMSYS, Portage, MI

ABSTRACT
A common necessity in clinical development and programming is having representative data to program and develop against. Situations that can hamper this could be that the data has not been collected yet, the data is too sensitive to share or just lack of resources to prepare and provide the data from the source data tables. This paper is intended to provide methods to generate representative data whether one has the project data or only metadata in such a way that sensitive data is not revealed.

INTRODUCTION
Often times, development or programming is delayed because of the lack of representative data. This paper is intended to briefly present a few different types of and possible methods for generating sample data to use for development.

The audience for this paper should have at least an introductory knowledge of SAS/Base and SAS Macro in order to follow along.

At this time, I consider this paper a work in progress as I am sure I will receive good ideas and feedback. I plan on using the ideas and feedback to further mature this paper to be more complete and useful for other SAS users. I apologize in advance to any data that is offended by being referred to as “dummy data”.

ATTRIBUTES OF GOOD DUMMY DATA
This section intends to discuss attributes that can make your dummy data more useful for development.

DATA STRUCTURES MATCH
This is really a necessity. If the field names and attributes do not match, development will really be hampered.

DATA VALUES ARE VALID (OR PURPOSEFULLY, INVALID)
Development is much more meaningful if the data values are representative. For example, the field gender should be mostly M/F or Male/Female. If it makes sense, a few outliers, invalid values and missing values should be mixed in. If the data field is a comment field, the varying value lengths should be used in the data generation.

RELATIONSHIPS BETWEEN KEYS ARE REPRESENTATIVE
Primary keys and foreign key relationships between tables are important to represent in an accurate manner to have a legitimate development environment.

SECURITY OF SENSITIVE DATA IS UPHELD
If the data contains items such as social security numbers, names, addresses, etc., it is not recommended to try and use it for development or testing in a shared environment. Sensitive data needs to be closely monitored and should not be copied from the production environment.

THE DATA IS EASILY AVAILABLE TO SAS ENVIRONMENTS
There are certain data sets that come with SAS by default. Some of them depend on the modules that are licensed. Good development data should be easily available to the development, testing and production environments. There should not be data that is too sensitive to be made available to those who need it.
POSSIBLE SOURCES FOR DUMMY DATA

Since there are many situations and requirements around data, different types of dummy data are more appropriate in different situations.

SASHELP DATA SETS

As all SAS users know, the SASHELP library is always there. If one is providing some sample code to show SAS functionality from a training perspective, what better data set to use than one such as SASHELP.CLASS? Another option is to use some of the views against the dictionary tables such as SASHELP.VTABLE a.k.a. DICTIONARY.TABLES.

GENERATING EMPTY TABLES WITH PROC SQL

The following PROC SQL code can be used to create empty tables with the same structure as the source data tables.

```sql
proc sql noprint;
    create table <target_table> like <source_table>;
quit;
```

GENERATING FROM METADATA

By leveraging the dictionary tables containing the metadata about the production data, one can generate dummy data of the exact same structure.

A proof of concept macro

```sql
/*----------------------------------------------------------

Macro Name: dummy_data
Programmer: Brian Varney
Date: 5/22/2009

Purpose: Leverage meta data about identified data from the SAS Dictionary
tables and generate data with the same structure.

Parameters:

  source_meta:  dictionary_tables
  source_lib:  source library
  source_ds:  source data set
  target_lib:  target library
  target_ds:  target datasets
  numobs_override:  number of observations override

----------------------------------------------------------*/
%macro dummy_data(source_meta  =dictionary_tables,source_lib  =sashelp,source_ds  =$ALL$,target_lib  =,
target_ds  =,numobs_override =
);
```
proc sql;
  create table lib_meta as
  select *
  from dictionary.tables
  where libname=%upcase("&source_lib.") and memtype="DATA"
  %if &source_ds=$ALL$ %then
  %do;
  %end;
  %else %do;
      and   memname=%upcase("&source_ds.")
  %end;
  ;
quit;
proc sql;
  create table ds_meta as
  select *
  from dictionary.columns
  where libname  =%upcase("&source_lib.") and
      memname =%upcase("&&memname&z.");
quit;
proc sql noprint;
  select name,
       type,
       length,
       label,
       format,
       informat
  into :name1     - :name9999,
       :type1     - :type9999,
       :length1   - :length9999,
       :label1    - :label9999,
       :format1   - :format9999,
       :informat1 - :informat9999
  from ds_meta;

  select count(*) into :numvars
  from ds_meta;
quit;
data &target_lib..&&memname&z.
    (label="%nrbquote(&&memlabel&z.");
attrib
%do i=1 %to &numvars.;
   &&name&i.
      %if &&type&i.=num %then
      %do;
         length=&&length&i.
      %end;
      %else %do;
         length=$&&length&i.
      %end;
      label="%nrbquote(&&label&i.)"
      %if &&format&i. ne %then
      %do;
         format=&&format&i..
      %end;
   %end;;
%do i=1 %to &numvars.;
   %if &&type&i.=char %then
   %do;
      &&name&i.="A"||repeat('a',&&length&i.-1);
   %end;
   %else %do;
      &&name&i.=99;
   %end;
%end;
%do j=1 %to &&numobs&z.;
   output;
%end;
run;
%dummy_data(source_lib=sashelp,
   source_ds=$ALL$,
   target_lib=work,
   target_ds=,
   numobs_override=9)
   );
   - Actual Data

<table>
<thead>
<tr>
<th>Obs</th>
<th>Name</th>
<th>Sex</th>
<th>Age</th>
<th>Height</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alfred</td>
<td>M</td>
<td>14</td>
<td>69.0</td>
<td>112.5</td>
</tr>
<tr>
<td></td>
<td>Name</td>
<td>Sex</td>
<td>Age</td>
<td>Height</td>
<td>Weight</td>
</tr>
<tr>
<td>---</td>
<td>--------</td>
<td>-----</td>
<td>-----</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>2</td>
<td>Alice</td>
<td>F</td>
<td>13</td>
<td>56.5</td>
<td>84.0</td>
</tr>
<tr>
<td>3</td>
<td>Barbara</td>
<td>F</td>
<td>13</td>
<td>65.3</td>
<td>98.0</td>
</tr>
<tr>
<td>4</td>
<td>Carol</td>
<td>F</td>
<td>14</td>
<td>62.8</td>
<td>102.5</td>
</tr>
<tr>
<td>5</td>
<td>Henry</td>
<td>M</td>
<td>14</td>
<td>63.5</td>
<td>102.5</td>
</tr>
<tr>
<td>6</td>
<td>James</td>
<td>M</td>
<td>12</td>
<td>57.3</td>
<td>83.0</td>
</tr>
<tr>
<td>7</td>
<td>Jane</td>
<td>F</td>
<td>12</td>
<td>59.8</td>
<td>84.5</td>
</tr>
<tr>
<td>8</td>
<td>Janet</td>
<td>F</td>
<td>15</td>
<td>62.5</td>
<td>112.5</td>
</tr>
<tr>
<td>9</td>
<td>Jeffrey</td>
<td>M</td>
<td>13</td>
<td>62.5</td>
<td>84.0</td>
</tr>
<tr>
<td>10</td>
<td>John</td>
<td>M</td>
<td>12</td>
<td>59.0</td>
<td>99.5</td>
</tr>
<tr>
<td>11</td>
<td>Joyce</td>
<td>F</td>
<td>11</td>
<td>51.3</td>
<td>50.5</td>
</tr>
<tr>
<td>12</td>
<td>Judy</td>
<td>F</td>
<td>14</td>
<td>64.3</td>
<td>90.0</td>
</tr>
<tr>
<td>13</td>
<td>Louise</td>
<td>F</td>
<td>12</td>
<td>56.3</td>
<td>77.0</td>
</tr>
<tr>
<td>14</td>
<td>Mary</td>
<td>F</td>
<td>15</td>
<td>66.5</td>
<td>112.0</td>
</tr>
<tr>
<td>15</td>
<td>Philip</td>
<td>M</td>
<td>16</td>
<td>72.0</td>
<td>150.0</td>
</tr>
<tr>
<td>16</td>
<td>Robert</td>
<td>M</td>
<td>12</td>
<td>64.8</td>
<td>128.0</td>
</tr>
<tr>
<td>17</td>
<td>Ronald</td>
<td>M</td>
<td>15</td>
<td>67.0</td>
<td>133.0</td>
</tr>
<tr>
<td>18</td>
<td>Thomas</td>
<td>M</td>
<td>11</td>
<td>57.5</td>
<td>85.0</td>
</tr>
<tr>
<td>19</td>
<td>William</td>
<td>M</td>
<td>15</td>
<td>66.5</td>
<td>112.0</td>
</tr>
</tbody>
</table>

- Dummy Data

<table>
<thead>
<tr>
<th>Obs</th>
<th>Name</th>
<th>Sex</th>
<th>Age</th>
<th>Height</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Aaaaaaa</td>
<td>A</td>
<td>99</td>
<td>99</td>
<td>99</td>
</tr>
<tr>
<td>2</td>
<td>Aaaaaaa</td>
<td>A</td>
<td>99</td>
<td>99</td>
<td>99</td>
</tr>
<tr>
<td>3</td>
<td>Aaaaaaa</td>
<td>A</td>
<td>99</td>
<td>99</td>
<td>99</td>
</tr>
<tr>
<td>4</td>
<td>Aaaaaaa</td>
<td>A</td>
<td>99</td>
<td>99</td>
<td>99</td>
</tr>
<tr>
<td>5</td>
<td>Aaaaaaa</td>
<td>A</td>
<td>99</td>
<td>99</td>
<td>99</td>
</tr>
<tr>
<td>6</td>
<td>Aaaaaaa</td>
<td>A</td>
<td>99</td>
<td>99</td>
<td>99</td>
</tr>
<tr>
<td>7</td>
<td>Aaaaaaa</td>
<td>A</td>
<td>99</td>
<td>99</td>
<td>99</td>
</tr>
<tr>
<td>8</td>
<td>Aaaaaaa</td>
<td>A</td>
<td>99</td>
<td>99</td>
<td>99</td>
</tr>
<tr>
<td>9</td>
<td>Aaaaaaa</td>
<td>A</td>
<td>99</td>
<td>99</td>
<td>99</td>
</tr>
</tbody>
</table>

**GENERATING DATA WITH PROC SQL AND THE DATA STEP**

The data step is perfectly suited for generating data due to the flexibility of the functions available, the natural looping mechanism and the ability to output several data sets at once.

**Example using PROC SQL**

```plaintext
data patids;
   do patid=1 to 10;
      output;
```

5
end;
run;

data treatments;
  do treatment='Placebo','T1','T2';
    output;
  end;
run;

data visits;
  do visit='Baseline','Day 2','Day 3','Day 4';
    output;
  end;
run;

proc sql;
  create table all as
  select p.patid,
   case when mod(p.patid,2)=0 then "M" else "F" end as gender,
   t.treatment, v.visit,
   20.0 + normal(0)*1.8 as BMI,
   round(1.62+ normal(0)*.05,.01) as Height,
   round(calculated BMI* calculated height**2,.1) as Weight
  from patids p,
    treatments t,
    visits v;
  quit;

Resulting data for PATID=1

<table>
<thead>
<tr>
<th>Obs</th>
<th>patid</th>
<th>gender</th>
<th>treatment</th>
<th>visit</th>
<th>BMI</th>
<th>Height</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>F</td>
<td>Placebo</td>
<td>Baseline</td>
<td>21.6364</td>
<td>1.64</td>
<td>58.2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>F</td>
<td>Placebo</td>
<td>Day 2</td>
<td>21.2128</td>
<td>1.61</td>
<td>55.0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>F</td>
<td>Placebo</td>
<td>Day 3</td>
<td>17.4086</td>
<td>1.60</td>
<td>44.6</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>F</td>
<td>Placebo</td>
<td>Day 4</td>
<td>18.1382</td>
<td>1.59</td>
<td>45.9</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>F</td>
<td>T1</td>
<td>Baseline</td>
<td>23.4352</td>
<td>1.62</td>
<td>61.5</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>F</td>
<td>T1</td>
<td>Day 2</td>
<td>22.6535</td>
<td>1.56</td>
<td>55.1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>F</td>
<td>T1</td>
<td>Day 3</td>
<td>19.8473</td>
<td>1.50</td>
<td>44.7</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>F</td>
<td>T1</td>
<td>Day 4</td>
<td>20.1527</td>
<td>1.58</td>
<td>50.3</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>F</td>
<td>T2</td>
<td>Baseline</td>
<td>20.5718</td>
<td>1.54</td>
<td>48.8</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>F</td>
<td>T2</td>
<td>Day 2</td>
<td>20.7244</td>
<td>1.60</td>
<td>53.1</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>F</td>
<td>T2</td>
<td>Day 3</td>
<td>19.1138</td>
<td>1.58</td>
<td>47.7</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>F</td>
<td>T2</td>
<td>Day 4</td>
<td>16.6206</td>
<td>1.63</td>
<td>44.2</td>
</tr>
</tbody>
</table>
Example using the SAS Data Step

Using techniques borrowed from "Generating Data with the SAS Dataset" by Andrew Cary (SUGI 22), the code below uses the data step, do loops, arrays and the rantbl function to generate data with realistic variability that the user defines.

For example, in the first data step below, the data step generates a 1,000 patient demographic data set with a variable for gender. The values for gender take on the values listed in the array. Notice missing and invalid values can easily be included. The rantbl function allows one to specify the proportions desired for each value of gender.

```sas
data demo(keep=pat_id gender);
array gender_list[4] $10 _temporary_ ('Male','Female',' ','emale');
do pat_id=1 to 1000;
    gender=gender_list[rantbl(00 00, .45 .45 .45 .45, .48 .48 .48 .48, .06 .06 .06 .06, .01 .01 .01 .01)];
    output;
end;
run;
```

```sas
data demo(keep=pat_id gender)
   ae (keep=pat_id visit ae);
array gender_list[4] $10 _temporary_ ('Male','Female',' ','emale');
array ae_list[6] $50 _temporary_
   ('HEADACHE','BACKACHE','NAUSEA','VOMITING','BAD BREATH',' '); 
do pat_id=1 to 1000;
    gender=gender_list[rantbl(0, .45, .48, .06, .01)];
    output demo;
    do visits=0 to 5;
        ae=ae_list[rantbl(0, .4, .3, .2, .07, .02, .01)];
        output ae;
    end;
end;
run;
```

**GENERATEDATA WEB SITE**

This is a thin client tool to help generate a single table of dummy data.
EXCEL ADD-IN

This is an Excel Add-In that I found on the internet which helps generate a single table of dummy data.
If the data is not sensitive and there is data available, the SURVEYSELECT procedure can be used to generate samples from the actual data. This is typically appropriate if the actual data sources are too large to use for development.

EXAMPLES

Examples of the different approaches will be provided during the presentation.
CONCLUSION

Planning for the existence and/or creation of data for development and programming is an important issue to avoid the risk of delays on your project. If actual data is not available at the necessary stage of a project, the creation of dummy data to use for development is essential to build in as a task. The generated data can also be part of the deliverable to be used for future testing and benchmarking.

REFERENCES

SUGI 22 Paper “Generating Data With the SAS Data Step” by Andrew J. L.
http://www2.sas.com/proceedings/sugi22/CODERS/PAPER74.PDF

Web Based Tool for Generating Sample Data
http://generatedata.com

Excel Add-in for Generating Sample Data

CONTACT INFORMATION

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

    Brian Varney, Senior Technical Manager
    COMSYS
    5220 Lovers Lane
    Portage, MI 49002
    Phone:    269-553-5185
    Fax:      269-553-5101
    E-mail:   bvarney@comsys.com
    Web:      www.comsys.com/analytics

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.