Count on SAS® to Count Your Census
Mark Mihalyo, Community Care Behavioral Health Organization, Pittsburgh, PA

ABSTRACT
In health care, especially in a patient facility setting, knowing your daily census is essential information. More importantly, being able to examine that daily census across multiple months, or even over a year period, can provide valuable trending data for a facility and other interested parties. By putting SAS® to work, producing a census report across time can be completed in undemanding fashion. Constructing well-defined iterative logic allows the user to employ SAS’s ability for compiling large amounts of data, and from that produce comprehensive reports. When you add in the flexibility that a MACRO can provide, SAS is off and running to put together a census report that will trend, average, and be one that you can ‘count on.’

INTRODUCTION
Community Care Behavioral Health Organization (Community Care) is a member of the Insurance Services Division of the University Of Pittsburgh Medical Center (UPMC). Community Care is the only Pennsylvania-based, not-for profit, licensed behavioral health managed care organization (BH-MCO) created to serve HealthChoices programs (Medicaid) throughout Pennsylvania. Community Care manages behavioral healthcare for nearly 600,000 Medicaid managed care enrollees. Community Care was designed primarily to serve the needs of public sector consumers, their families, and their communities.

Community Care is committed to supporting the recovery of our members. Recovery is promoted through care management, provider education, physical and behavioral health integration, family involvement, and the development of new clinical initiatives.

Detailed analyses to measure outcomes are a very important aspect for Community Care’s ability to monitor the success of member interventions. SAS is an integral tool for Community Care’s Decision Support team in developing these analyses, and exploring significant amounts of data and information by efficient and manageable methods.

THE TASK
The task for this project was to develop a program that can deliver census data across multiple months utilizing provider claims, collected and submitted by patient facilities. Examples of these facilities are hospitals, residential treatment, and drug and alcohol treatment units. The objective for monitoring data like the daily census, or number of daily admissions, across a period of time is that trends can be established. The ability to view increases/decreases in this data is valuable information to hospitals and companies like Community Care.

The most important factor in establishing a program that can accomplish this type of task is allowing for flexibility. The program user needs to be able to easily adjust the timeframes for running data. For instance, there may be interest in what the census was across fiscal year 2010 as compared to fiscal year 2012, or calendar year 2009 vs 2011. The program is set up in a manner that allows for quick and efficient date changes.

The other consideration is the ability to work with large data sets, like provider medical claims data. The goal is to ultimately eliminate the need for manual calculation when it comes to census or admission reports. Situations do exist where individual daily reports that collect census data are tediously added together to generate a complete longitudinal report that SAS can achieve in a fraction of the time. The key is finding ways to take significant amounts of data and produce the ‘routine’ type of information that help drive decisions.
THE SOLUTION

The programming solution to collect the described census data is to be creative by using %DO statements embedded in %IF %THEN and %ELSE statements. By generating dates within a loop the user has the ability to take large data sets and isolate the timeframes that are needed to be measured. Once the program has isolated these timeframes, episodes can be established that show the length of time a patient was hospitalized, or time spent in a facility setting. After the episodes are formulated the data is now set up to do quick daily tabulations in each month for the census or admission count. These steps will be demonstrated by building a simple ARRAY and using PROC TRANSPOSE to establish data sets for multiple months that can merged in to one data set and makes for uncomplicated reporting.

Because multiple MACROs are used in the program using the %INCLUDE statement in a separate program nicely organizes all the macro parameters and the macro variables that are used in one convenient place.

The macros in this code help launch the structure needed so that each section of the program creates the data sets that build off each other. The macros along with the Do, If/Then, Else statements make for powerful tools that produce daily totals, over multiple months, and between comparison periods. SAS makes pulling this type of data easy with a couple simple statements that resourcefully gather and structures only the information you need to generate your report.

The code in this section below walks you through this process and the daily census is generated from inpatient mental health hospital claims:

****************************************************************************** Inpatient Mental Health Bed Day Annual Census Count ******************************************************************************

Code Example Used to Run the Program that Generates the Census Count:

%let sm=7;
%let sd=1;
%let sy=2011;
%let em=6;
%let ed=30;
%let ey=2012;
%let smy=2011;
%let emy=2012;
%include "My SAS files\SAS_data\Census_Daily_Admits_Days\census_ttls_exhibit.sas";
%complt_data (ipmh, facility1);
%cens (ipmh, 2011);
%cens (ipmh, 2012);
%cens1 (ipmh);
%prjct (ipmh, Inpatient Mental Health, 114 124 144);
ods pdf close; - closes out the report section if one is included
quit;
******************************************************************************

Sample Data:

<table>
<thead>
<tr>
<th>Patient_id</th>
<th>Episode_From_Dat</th>
<th>Episode_To_Dat</th>
<th>Billing_Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXXXXXXX</td>
<td>08/15/2011</td>
<td>09/01/2011</td>
<td>0124</td>
</tr>
<tr>
<td>XXXXXXXX</td>
<td>11/20/2011</td>
<td>12/05/2011</td>
<td>0124</td>
</tr>
<tr>
<td>XXXXXXXX</td>
<td>01/10/2012</td>
<td>02/10/2012</td>
<td>0124</td>
</tr>
<tr>
<td>XXXXXXXX</td>
<td>03/01/2012</td>
<td>03/18/2012</td>
<td>0124</td>
</tr>
</tbody>
</table>
PROGRAM LOGIC:

%macro complt_data(grp, cnty);

**************Bring in Claims Data that will Used to Run The Census***************

data hx_data1;
set data.&grp._detail;
where service_code in (....) - include codes that pull the specific facility services run;

data &grp._complt_ttls;
set hx_data1;
run;

******Keep only necessary data that will run through program***********************

proc sort data=&grp._complt_ttls
out=&grp._complt_ttls1(keep=member_id billing_code epi_from_dat epi_to_dat) nodupkey;
by member_id billing_code epi_from_dat epi_to_dat;
run;
%mend complt_data;

**********Logic will insert the beginning date or end date for the Measure Period********
**********if the member’s date starts/ends before/after measure period**********

%macro cens(loc, per);

data month_totals;
set &loc._complt_ttls1;

if epi_from_dat < mdy(&sm,&sd,&sy) then
episode_start1 = mdy(&sm,&sd,&sy);
else episode_start1 = epi_from_dat;

if epi_to_dat > mdy(&em,&ed,&ey) then
episode_end1 = mdy(&em,&ed,&ey);
else episode_end1 = epi_to_dat;
format episode_start1 episode_end1 date07.;
run;

*******Create Date Spans for all Episodes in the Period they occur*******
*******This logic will find the Days and Month where the member was*******
*******active in the reported Level of Care**************

%if &per=&sy %then %do;
%do mn=&sm %to 12;
%let day1=1;
%let day2=31;
%if ($mn=4 or $mn=6 or $mn=9 or $mn=11) %then %do;
%let day2=30;
%end;
%else %if ($mn=2) %then %do;
%let day2=28;
%end;

Use the Macro Variable from the ‘Report Run’ program to apply the start date or end date if needed.

Use the %DO statement here to accomplish the following task:

1. Runs a loop that will create date variables that are used to find any patient episodes in the month that is being run during the iterations of the loop.
2. Creates individual data sets for each month that is in the date span being pulled. The individual month data will later be merged to form one data set for the entire measure period.
data elig&mn._&sy.;
set month_totals;
if((mdy(&mn, &day1, &sy) <= episode_start1 <= mdy(&mn, &day2, &sy))
or (mdy(&mn, &day1, &sy) <= episode_end1 <= mdy(&mn, &day2, &sy))
or (mdy(&mn, &day1, &sy) >= episode_start1 and mdy(&mn, &day2, &sy) <= episode_end1)
or (mdy(&mn, &day1, &sy) <= episode_start1 and mdy(&mn, &day2, &sy) >= episode_end1));
run;

data elig&mn._&sy.a;
set elig&mn._&sy;
if episode_start1 >= mdy(&mn, &day1, &sy) then
   episode_start2 = episode_start1;
else episode_start2 = mdy(&mn, &day1, &sy);
if episode_end1 >= mdy(&mn, &day2, &sy) then
   episode_end2 = mdy(&mn, &day2, &sy);
else episode_end2 = episode_end1;
start_day=day(episode_start2);
end_day=day(episode_end2);
fomat episode_start2 episode_end2 date07.;
run;

data days&mn._&sy;
set elig&mn._&sy.a;
array census{&day2} Day1 - Day&day2;
%do d=1 %to &day2;
   if start_day <= &d <= end_day then
      census{&d}=1;
   else census{&d}=0;
%end;
run;

proc means data=days&mn._&sy(keep=Day1 - Day&day2) noprint;
var Day1 - Day&day2;
output out=sum&mn._&sy (drop=_type_ _freq_) sum=;
run;

proc transpose data=sum&mn._&sy out=sum&mn._&sy.a;
var Day1 - Day&day2;
run;

data sum&mn._&sy.a(rename={_NAME_="days" COL1=census&mn});
set sum&mn._&sy.a;
run;

data sum&mn._&sy.a;
set sum&mn._&sy.a;
label days="Days";
census1="Jan-&smy"
census2="Feb-&smy"
census3="Mar-&smy"
census4="Apr-&smy"
census5="May-&smy"
census6="Jun-&smy"
census7="Jul-&smy"
census8="Aug-&smy"
census9="Sep-&smy"
census10="Oct-&smy"
census11="Nov-&smy"
census12="Dec-&smy";

1. Now that individual patient episodes are established use an ARRAY to create columns for each day of the month and count a member as 1 for each day they are present in the facility setting.

2. The PROC MEANS is then used to sum the total members counted for each day of the month.

3. Finally, The PROC TRANSPOSE conveniently converts the ‘days of the month’ columns (created by the ARRAY) to individual rows for each month.
run;

data &loc._census&sy;
merge &do j=&sm &to 12;
sum&j._&sy.a
&end;
run;
&end;
&end;

%else %do;
%do mn=1 &to &em;
%let day1=1;
%let day2=31;
%if (&mn=4 or &mn=6 or &mn=9 or &mn=11) %then %do;
%let day2=30;
%end;
%else %if (&mn=2) %then %do;
%let day2=28;
%end;

%let cens1 (loc2);
%if &sy ~= &ey %then %do;

data elig&m.&ey;
set month_totals;
if ((mdy(&mn, &day1, &ey) <= episode_start1 <= mdy(&mn, &day2, &ey))
or (mdy(&mn, &day1, &ey) <= episode_end1 <= mdy(&mn, &day2, &ey))
or (mdy(&mn, &day1, &ey) >= episode_start1 and mdy(&mn, &day2, &ey) <= episode_end1))
or (mdy(&mn, &day1, &ey) <= episode_start1 and mdy(&mn, &day2, &ey) >= episode_end1));
run;

data elig&m._a;
set elig&m._;

************ Run Second Period - if one exist as in the case of a Fiscal Year - i.e. 7/1/2011 through 6/30/2012******
%mend cens;

********** end loop to create monthly iterations of days spent in level of care ************
.
.
.
.
.
.
************ Merge both sets of data together for full census **************
%macro cens1 (loc2);
%if &sy ~= &ey %then %do;

data &loc2._ttl_census;
merge &loc2._census&sy
&loc2._census&ey;
run;
%end;
%else %do;

data &loc2._ttl_census;
set &loc2._census&sy;
run;
%end;
%mend cens1;
%macro prjct(loc3, svc, cde);

*************** Now Get Totals for each Month ***********************

proc means data=&loc3._ttl_census noprint;
var census&sm -- census&em;
output out=sum_census (drop=_type_ _freq_) sum=;
format census&sm -- census&em 7.;
run;

***************************************************************

proc format;
value $mont_dy 'Day1'='1'
'Day2'='2'
'Day3'='3'
'Day4'='4'
'Day5'='5'
'...
'Day31'='31'
'Total'='Total';
run;

data &loc3._ttl_census_a;
length days $10;
set &loc3._ttl_census
sum_census;
if days=' ' then days="Total";
run;

data &loc3._ttl_census_a;
set &loc3._ttl_census_a;
format days $mont_dy.;run;

***************************************************************

* A Simple ODS Report code can generate a comprehensive report that can detail the census data across time - Add in a graph to provide a visual representation of the census trend over time.*

%mend prjct;
## THE RESULTS

**Example: Daily Census Counts for FY 2012**  
*Inpatient Facility Provider Daily Census*  
*Data Source: Medical Billing Claims*

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>95</td>
<td>117</td>
<td>109</td>
<td>112</td>
<td>98</td>
<td>87</td>
<td>84</td>
<td>105</td>
<td>126</td>
<td>112</td>
<td>100</td>
<td>110</td>
</tr>
<tr>
<td>2</td>
<td>95</td>
<td>111</td>
<td>106</td>
<td>112</td>
<td>96</td>
<td>79</td>
<td>88</td>
<td>106</td>
<td>121</td>
<td>118</td>
<td>106</td>
<td>104</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>110</td>
<td>102</td>
<td>123</td>
<td>95</td>
<td>80</td>
<td>97</td>
<td>103</td>
<td>106</td>
<td>108</td>
<td>106</td>
<td>110</td>
</tr>
<tr>
<td>4</td>
<td>108</td>
<td>104</td>
<td>105</td>
<td>121</td>
<td>101</td>
<td>87</td>
<td>95</td>
<td>92</td>
<td>107</td>
<td>102</td>
<td>104</td>
<td>117</td>
</tr>
<tr>
<td>5</td>
<td>115</td>
<td>109</td>
<td>108</td>
<td>113</td>
<td>97</td>
<td>92</td>
<td>90</td>
<td>96</td>
<td>116</td>
<td>106</td>
<td>103</td>
<td>116</td>
</tr>
<tr>
<td>6</td>
<td>120</td>
<td>106</td>
<td>113</td>
<td>112</td>
<td>97</td>
<td>92</td>
<td>95</td>
<td>103</td>
<td>112</td>
<td>103</td>
<td>113</td>
<td>112</td>
</tr>
<tr>
<td>7</td>
<td>113</td>
<td>104</td>
<td>112</td>
<td>111</td>
<td>106</td>
<td>87</td>
<td>94</td>
<td>100</td>
<td>108</td>
<td>93</td>
<td>120</td>
<td>105</td>
</tr>
<tr>
<td>8</td>
<td>115</td>
<td>121</td>
<td>105</td>
<td>110</td>
<td>111</td>
<td>87</td>
<td>94</td>
<td>105</td>
<td>110</td>
<td>93</td>
<td>119</td>
<td>100</td>
</tr>
<tr>
<td>9</td>
<td>109</td>
<td>118</td>
<td>100</td>
<td>113</td>
<td>99</td>
<td>87</td>
<td>103</td>
<td>106</td>
<td>114</td>
<td>101</td>
<td>117</td>
<td>90</td>
</tr>
<tr>
<td>10</td>
<td>109</td>
<td>114</td>
<td>94</td>
<td>119</td>
<td>98</td>
<td>90</td>
<td>104</td>
<td>111</td>
<td>104</td>
<td>119</td>
<td>94</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>121</td>
<td>117</td>
<td>101</td>
<td>124</td>
<td>99</td>
<td>90</td>
<td>108</td>
<td>113</td>
<td>102</td>
<td>116</td>
<td>104</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>117</td>
<td>108</td>
<td>111</td>
<td>122</td>
<td>91</td>
<td>96</td>
<td>101</td>
<td>105</td>
<td>126</td>
<td>102</td>
<td>106</td>
<td>100</td>
</tr>
<tr>
<td>13</td>
<td>113</td>
<td>99</td>
<td>111</td>
<td>117</td>
<td>94</td>
<td>101</td>
<td>99</td>
<td>113</td>
<td>117</td>
<td>98</td>
<td>108</td>
<td>99</td>
</tr>
<tr>
<td>14</td>
<td>108</td>
<td>101</td>
<td>114</td>
<td>114</td>
<td>100</td>
<td>109</td>
<td>97</td>
<td>118</td>
<td>120</td>
<td>95</td>
<td>118</td>
<td>98</td>
</tr>
<tr>
<td>15</td>
<td>114</td>
<td>110</td>
<td>109</td>
<td>107</td>
<td>95</td>
<td>102</td>
<td>94</td>
<td>115</td>
<td>114</td>
<td>95</td>
<td>116</td>
<td>100</td>
</tr>
<tr>
<td>16</td>
<td>102</td>
<td>105</td>
<td>107</td>
<td>110</td>
<td>93</td>
<td>110</td>
<td>99</td>
<td>108</td>
<td>116</td>
<td>108</td>
<td>109</td>
<td>91</td>
</tr>
<tr>
<td>17</td>
<td>109</td>
<td>93</td>
<td>105</td>
<td>118</td>
<td>95</td>
<td>108</td>
<td>98</td>
<td>110</td>
<td>117</td>
<td>108</td>
<td>113</td>
<td>94</td>
</tr>
<tr>
<td>18</td>
<td>117</td>
<td>94</td>
<td>100</td>
<td>119</td>
<td>83</td>
<td>116</td>
<td>102</td>
<td>106</td>
<td>124</td>
<td>103</td>
<td>108</td>
<td>96</td>
</tr>
<tr>
<td>19</td>
<td>120</td>
<td>92</td>
<td>104</td>
<td>114</td>
<td>74</td>
<td>122</td>
<td>94</td>
<td>112</td>
<td>130</td>
<td>102</td>
<td>97</td>
<td>91</td>
</tr>
<tr>
<td>20</td>
<td>114</td>
<td>88</td>
<td>116</td>
<td>110</td>
<td>76</td>
<td>118</td>
<td>88</td>
<td>118</td>
<td>127</td>
<td>104</td>
<td>96</td>
<td>95</td>
</tr>
<tr>
<td>21</td>
<td>109</td>
<td>93</td>
<td>113</td>
<td>105</td>
<td>86</td>
<td>109</td>
<td>86</td>
<td>120</td>
<td>119</td>
<td>101</td>
<td>104</td>
<td>109</td>
</tr>
<tr>
<td>22</td>
<td>107</td>
<td>102</td>
<td>119</td>
<td>96</td>
<td>92</td>
<td>103</td>
<td>91</td>
<td>115</td>
<td>113</td>
<td>104</td>
<td>110</td>
<td>111</td>
</tr>
<tr>
<td>23</td>
<td>111</td>
<td>99</td>
<td>115</td>
<td>93</td>
<td>89</td>
<td>92</td>
<td>99</td>
<td>119</td>
<td>105</td>
<td>117</td>
<td>113</td>
<td>104</td>
</tr>
<tr>
<td>24</td>
<td>115</td>
<td>100</td>
<td>113</td>
<td>105</td>
<td>84</td>
<td>70</td>
<td>102</td>
<td>123</td>
<td>97</td>
<td>115</td>
<td>110</td>
<td>110</td>
</tr>
<tr>
<td>25</td>
<td>126</td>
<td>108</td>
<td>115</td>
<td>101</td>
<td>83</td>
<td>68</td>
<td>104</td>
<td>117</td>
<td>117</td>
<td>102</td>
<td>102</td>
<td>115</td>
</tr>
<tr>
<td>26</td>
<td>125</td>
<td>108</td>
<td>124</td>
<td>102</td>
<td>92</td>
<td>73</td>
<td>102</td>
<td>117</td>
<td>111</td>
<td>105</td>
<td>118</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>128</td>
<td>107</td>
<td>131</td>
<td>100</td>
<td>95</td>
<td>81</td>
<td>101</td>
<td>126</td>
<td>102</td>
<td>109</td>
<td>107</td>
<td>115</td>
</tr>
<tr>
<td>28</td>
<td>120</td>
<td>107</td>
<td>127</td>
<td>94</td>
<td>100</td>
<td>83</td>
<td>97</td>
<td>131</td>
<td>106</td>
<td>103</td>
<td>109</td>
<td>119</td>
</tr>
<tr>
<td>29</td>
<td>119</td>
<td>120</td>
<td>124</td>
<td>90</td>
<td>100</td>
<td>83</td>
<td>100</td>
<td>-</td>
<td>107</td>
<td>108</td>
<td>117</td>
<td>118</td>
</tr>
<tr>
<td>30</td>
<td>107</td>
<td>112</td>
<td>120</td>
<td>89</td>
<td>101</td>
<td>84</td>
<td>108</td>
<td>-</td>
<td>-110</td>
<td>117</td>
<td>113</td>
<td>110</td>
</tr>
<tr>
<td>31</td>
<td>110</td>
<td>114</td>
<td>-</td>
<td>98</td>
<td>-</td>
<td>80</td>
<td>104</td>
<td>-</td>
<td>-111</td>
<td>-107</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Total</td>
<td>3491</td>
<td>3289</td>
<td>3338</td>
<td>3374</td>
<td>2808</td>
<td>2866</td>
<td>3018</td>
<td>3100</td>
<td>3521</td>
<td>3149</td>
<td>3391</td>
<td>3155</td>
</tr>
</tbody>
</table>

## CONCLUSION

Even when working with a large volume of data using a straightforward approach to programming, along with some of SAS's unique tools, allows a user to effectively manage and manipulate information. The programming does not have to be complex, but rather well thought out for the best methods to isolate only the data that is needed to produce the results that one is looking for in a report. This paper has explored how features, like iterative logic, MACRO variables and statements, arrays, and conditional logic are all resourceful tools in summarizing considerable amounts of data. This is the appealing feature SAS offers by its ability to breakdown data into small, easy to use data sets without extensive programming. It's the reason to count on SAS for your counts!

## REFERENCES

SAS Programming III: Advanced Techniques and Efficiencies 2006, SAS Institute, Cary, NC, USA
ACKNOWLEDGMENTS

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 registered trademarks or trademarks of their respective companies.

CONTACT INFORMATION:

Mark Mihalyo, MHMS
Community Care Behavioral Health Organization
One Chatham Center
Suite 700
Pittsburgh, PA 15219
Work Phone: 412-402-8769
Fax: 412-454-2177
Email: mihalyomj@ccbh.com