

ABC Macro and Performance Chart with Benchmarks Annotation

Jing Li, AQAF, Birmingham, AL

ABSTRACT

The achievable benchmark of care (ABCTM) approach identifies the performance of the top 10% of healthcare providers as the benchmark. Comparing individual performance with peer performance should be a powerful motivator to change behavior and improve quality of healthcare. This paper presents a SAS Macro to calculate ABC, also presents the utilization of Annotate data set to label ABCs on individual clinician's performance bar chart.

INTRODUCTION

Benchmarking is generally considered to be an important tool for quality improvement. Achievable Benchmarks of Care (ABCs), calculated from existing performance data, represent attainable levels of excellence for a performance measure. ABCs allow individual healthcare provider to compare the outcomes for their patients relative to overall performance of the "peer-group".

As the Quality Improvement Organization (QIO) for the state of Alabama, AQAF works in partnership with physicians, hospitals, nursing homes and home health agencies to improve the quality of health care in Alabama. Historically, a provider's performance has been compared to the "average" performance of "peer-group" for assessment. To help healthcare providers set attainable goal and make them more involved and proactive in quality improvement, we include both peer-group averages and ABCs in individual performance charts, which is a powerful motivator for providers to change behavior and provide appropriate care to patients.

We are working with different healthcare providers on almost 50 clinical measures, and for each measure we have similar data layout: provider #, measure, period, numerator, denominator, and rate. Using a separate module would make the programming and the debugging a lot easier. By utilizing SAS Macro, we created the ABC Macro to calculate ABCs. Once ABCs are calculated, we use Annotate data set to label ABCs on providers' performance charts to enhance graphics output.

ABC MACRO

This Macro has 8 parameters:

benchmark(lib=,dataset=,ProvVar=,NumVar=,DenVar=,measure=,period=, range=)

This paper will use diabetes mellitus (DM) patients receiving annual eye examinations as an example for illustration. Performance is measured by the proportion of DM patients having claims data reflecting an eye exam. Assume you have a LIB named SESUG, in which there is a SAS data set called Diab_Eye_2006Q3 with the following layout.

Provider	MEASURE_PERIOD	Eye_NUM	Eye_DEN	Eye_PCT
#1	2006Q3	25	46	54.35%
#2	2006Q3	33	47	70.21%
...	...	...	...	...
State	2006Q3	41293	81867	50.44%

The key steps for ABC calculation are: determine eligible patient's breakpoint, calculate the Bayesian Adjusted Performance Fraction (APF), and identify 10% Bayesian adjusted performance and any tied performance. The ABC Macro utilizes Macro variables, retain variable, and PROC SQL procedure to calculate ABC and save it into a SAS data set. Below are the steps for ABC Macro. See appendix for complete program.

ABC Macro Calculation Steps

First, create a copy of data set, renaming fields for calculation.

Step 1: Determine eligible patient's breakpoint. Assign the value to a Macro variable.

Step 2: Create provider data set, excluding the record with overall data, like state average.

Step 3: Calculate the Bayesian Adjusted Performance Fraction (APF) for each provider.

Step 4: Rank all provider data in descending order of their APF.

Step 5: Use retain variable to calculate cumulative count of patients in rank ordered data.

Step 6: Identify 10% Bayesian adjusted performance. Assign the corresponding APF to a Macro variable.

Step 7: Create top 10% (tied) performance data set, use PROC SQL to calculate benchmark and save it into a SAS data set.

Call ABC Macro

```
%let path = H:\SESUG\; %let lib_name = SESUG; libname &lib_name "&path"; run;
%include "&path.ABC.sas" /source2; /* to include ABC macro*/
%benchmark(lib=&lib_name, dataset=diab_eye_2006Q3, ProvVar=provider,
NumVar=eye_num, DenVar=eye_den, measure=Diab_Eye, period=2006Q3, range=State);
```

ANNOTATE GRAPHICS

The Annotate facility enables us to generate an Annotate data set with graphics commands, from which we can produce graphics output to enhance graphs generated from SAS/GRAPH procedures. Our goal is to have a vertical bar chart showing individual healthcare provider performance and peer-group average performance over time, with ABCs labeled for each time period. See appendix for Annotate graphics example. We now have one provider chart Macro for each clinical measure. Same as above, we use DM patients receiving annual eye examination as an example for illustration. Assume you have a LIB named SESUG, in which there are SAS data sets called Diab_Eye_PCT and Diab_Eye_ABCs with the following layouts:

Diab_Eye_PCT

Prov_Num	Prov_Name	Period	PCT
#001	Physician A	Baseline	58.46%
...	...	...	...
State	State	Baseline	49.00%
...	...	Interim1	...

Diab_Eye_ABCs

Measure	Period	Benchmark
Diab_Eye	Baseline	0.7324
Diab_Eye	Interim1	0.7443
Diab_Eye	Interim2	0.7429
...	...	...

The Diab_Eye_Chart Macro creates one PDF file for each provider and it is easy to be called:
%Diab_Eye_Chart(ProvNum);

Create Annotate data set

In order to create Annotate graphics, we first need to build an Annotate data set of graphics commands using the Annotate variables and functions. The Annotate facility labels benchmark for each period. The coordinates that position the labels are derived from the values of the GCHART procedure variables Period (the group variable) and Prov_Name (the chart / midpoint variable). These variables are assigned to the corresponding Annotate variable. The Annotate data set benchmark_anno has the following features:

Annotate Function: LABEL

Annotate Variables: MIDPOINT, POSITION, GROUP, XSYS, YSYS, Y, WHEN, SIZE, STYLE, COLOR, TEXT

Below is the Annotate data set code section in Diab_Eye_Chart Macro.

```
data benchmark_anno;
merge &ProvNum._Diab_Eye (in=x) /* subset from Diab_Eye_PCT */
    &lib_name..Diab_Eye_ABCs;
by period;
format benchmark percent8.2;
length function $ 8 style $ 20 color $ 8;
retain function 'label' position '6' xsys ysys '2' when 'a';
```

```

style='swissb'; color='blue'; size=0.8;
group=period; midpoint=prov_name; y=benchmark;
text=cat('Benchmark ',left(put(benchmark,percent8.2)));
if x and provider = "&ProvNum" then output;
run;

```

Generate a vertical bar chart and assign the Annotate data set to the VBAR statement

Now we have Annotate data set generated, next step is to submit a SAS/GRAPH procedure to produce the graphics output. We use PROC GCHART with ANNO function to create bar chart. Below is the PROC GCHART code section in Diab_Eye_Chart Macro.

```

proc gchart data=&ProvNum._Diab_Eye;

  vbar prov_name / type=mean sumvar=pct frame

  maxis=axis1 raxis=axis2 autoref clipref cautohref=CXC0C0C0

  legend=legend1 group=period g100 gaxis=axis3 subgroup=prov_name

  anno=benchmark_anno discrete;

run; quit;

```

By calling Diab_Eye_Chart Macro, we could have 500 providers' charts in 2 minutes. Our quality staffs carry the charts to healthcare providers during their site visits. Usually these providers pay immediate attention to the chart and express willingness to be the benchmark.

CONCLUSION

Having a separate ABC Macro, which is well programmed, the Macro can be accessed from any project as long as the data set is generated. Using this approach, we did avoid one single long program for each clinical measure and provided more efficient way of programming. Using Annotate facility to enhance graphics output, we provided healthcare providers more impressive and informative charts for quality improvement.

REFERENCES

ABC™ User Manual, Dr. Catarina Kiefe, Center for Outcomes and Effectiveness Research and Education (COERE), University of Alabama at Birmingham

ACKNOWLEDGMENTS

Thanks to Bart Prevallet at AQAF who encouraged me to write this paper and provided editorial advice.

CONTACT INFORMATION

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

Jing Li
 AQAF
 Two Perimeter Park South, Suite 200 West
 Birmingham, AL 35243
 Phone: (205) 970-1600 ext. 3514
 Fax: (205) 970-1624
 E-mail: jli@aqaf.com

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.

The Achievable Benchmark of Care (ABC™) system is trademarked by the University of Alabama at Birmingham

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

APPENDIX

ABC MACRO Program

```
/* Macro to calculate ABC benchmark */
%macro benchmark(lib=,
                 dataset=,
                 ProvVar=,
                 NumVar=,
                 DenVar=,
                 measure=,
                 period=,
                 range=); /* 'State', 'Nation', or project name like 'SOW' */
/* Rename variable name to calculate benchmark */
data data_var_rename (rename=(&ProvVar=provider &NumVar=num &DenVar=den));
set &lib..&dataset.;
run;
/* Step 1: Determine eligible patients breakpoint for clinical indicator */
data _null_;
set data_var_rename;
    if provider = "&Range" then
        call symput('break_point',put(round(den/10,1),6.));
run;
/* Step 2: Create provider data set */
data raw;
set data_var_rename;
    if provider ne "&Range";
run;
/* Step 3: Calculate Bayesian Adjusted Performance Fraction (APF) */
data apf;
set raw;
    format apf_n 3. apf_d 3. apf_rate percent8.2;
    apf_n = num + 1;
    apf_d = den + 2;
    apf_rate = round(apf_n/apf_d,.0001);
run;
```

```

/* Step 4: Rank data in descending order of Bayesian Estimator */
proc sort data=apf;
by descending apf_rate descending den;
run;

/* Step 5: Calculate cumulative number of patients in rank ordered dataset */
data apf;
set apf;
    retain cum_patient;
    cum_patient + den;
run;

/* Step 6: Identify cut-off provider and corresponding rate */
data apf;
set apf;
    retain i;
    if cum_patient ge &break_point then i+1;
run;
data _null_;
set apf;
    if i = 1 then call symput('break_percent',put(apf_rate,6.4));
run;

/* Step 7: Ceate cut-off subset and calculate "paired mean" - Benchmark */
data apf;
set apf;
    if apf_rate ge &break_percent;
run;
proc sql;
    create table benchmark_&measure._&period as
    select "&range." as Range, "&measure." as Measure, "&period." as period,
        round(sum(num)/sum(den),.0001) as Benchmark format percent8.2
    from apf;
quit;
%mend benchmark;

```

Example of Annotate Graphics

