Site-Specific Reports: Comparing each part to the whole using numbers and figures

Jenny Yu, Lewin-TAG, Inc., San Francisco, CA

Abstract:

You have data that involves a number of sites, and it's time to report the information for each individual site compared to the whole. It is easy to compare each site with the whole by creating a dummy variable to indicate the individual site versus to whole. A simple trick in the DATA step allows you to make this comparison. When the number of sites is large, it is essential to automate the production of the tables and graphs. To do this, you will need a number of macro features and a lot of attention to potential pitfalls. This paper will show how to use SAS to automate a series of tables and graphs, for each site and for all sites simultaneously. The main tools are the FREQ, SUMMARY, and GCHART procedures and various macro features. Special problems arise when subdividing a graph into subgroups and displaying the corresponding sample sizes; details of a solution to this problem will be provided.

Introduction:

Once you have an analytic dataset that contains the final variables to be reported, and want to compare each site to the whole, you are still not ready to graph. Two critical steps must be taken before actually using SAS/GRAPH: 1) Create a new dataset (based on the final analytic file) using a simple trick in the DATA step to allow this type of comparison; and 2) assuming that proportions are to be reported, create output datasets using PROC FREQ, which contain the actual proportions/counts to graph. Also, if you plan to display sample sizes for each graph, a third step of using either PROC SUMMARY, MEANS, or UNIVARIATE is also required. Finally, when graphing, use the macro language not only to automate the graphs themselves, but other components of the report such as the titles, labels, and displaying sample sizes.

1) Use the DATA step to create a dummy variable to compare each site to the whole.

If you have k sites and n observations, you will need to create a new dataset that contains n(k+1) observations, in order to efficiently compare one site to the whole. In the following example, there are 3 sites, so data temp01 will contain 3n+n=4n observations:

```
data temp01;
  set wuss.grphdata;
  all=0;
  output;
  all=1;
  grphsite=1;
  output;
  do grphsite=5 to 6;
    output;
  end;
run;
```

The variable GRPHSITE indicates the site, and the dummy variable ALL indicates whether an observation is part of the 'all sites' group (ALL=1), or the site being graphed (ALL=0). This way, each original observation appears in the dataset once when ALL=0, with GRPHSITE equal to the site from which the observation came, and then appears again with ALL=1 once for each site.

The following is an example dataset of 6 observations, 3 test sites to help visualize how this data step technique works:

PATID	GRPHSITE	ALL
001	1	0
002	1	0
003	5	0
004	5	0
005	6	0
006	6	0
001	1	1
002	1	1
003	1	1
004	1	1
005	1	1

```
006     1          1
001     2          1
002     2          1
003     2          1
004     2          1
..etc
```

With your dataset formatted this way, comparing site #1 to all sites is simple. By choosing where GRPHSITE=1, when ALL=0, the observations are from site#1. When ALL=1, then observations are from all sites.

Once your dataset is ready, sort by GRPHSITE and ALL.

Three examples of graphs with HIV data, along with their code, are explained here.

The general format to create each of the three graphs is similar. All graphs start out by creating output datasets from a PROC FREQ or PROC MEANS. Then, PROC SUMMARY is used to create macro variables that display sample sizes. Finally, PROC GCHART is used to create the final graph.

Chart #1 – Reporting Age Strata

a) Creating an output dataset

Once the dataset with k(n+1) observations with the 'ALL' flag has been created, create an output dataset in PROC FREQ, that contains the frequencies of the age strata. Run this *BY GRPHSITE and ALL.*

```
proc freq data=temp01 noprint;
  by grphsite all;
  tables age/out=g1;
  format age agefmt.;
run;
```

Since percentages among the age strata are to be charted, we are interested in the PERCENT variable of the output dataset. PERCENT is not in percent format, so it must divided by 100 to graph appropriately, and formatted to PERCENT8.. Then, sort by GRPHSITE and ALL.

```
data g_1;
    set g1;
  propn=percent/100;
  label propn='Proportion';
  format propn percent8.;
run;
```

```
proc sort data=g_1;
  by grphsite all;
  run;
```

b) Displaying sample sizes, site names

G_1 will be the input dataset for PROC GCHART. As Chart #1 shows, the subtitle contains the name of the site as well as the sample sizes. Rather than hard-coding these names and numbers, using PROC SUMMARY and creating macro variables allows automation for multiple graphs and multiple sites. To display sample sizes, two macro variables are created, which contain the sample sizes of site #1 and all sites for the first graph:

```
%let i=1;
proc summary nway data=g_1 (where grphsite=&i);
  class all;
  var count;
  output out=out&i sum=num;
run;

data _null_;
  set out&i;
  if all=1 then call symput('num_1',put(num,3.));
  else if all=0 then call
      symput('num_0',put(num,3.));
  run;
```

&NUM_0 is the count for site 1 alone, and &NUM_1 is the count for all sites.

Use these macro variables in the title statement, along with the site name, which is also a macro variable:

```
%let site1=Site #1;
title3 "&&site&i (n=&num_0) and All Sites
       (n=&num_1)";
```

Using &&SITE&I lets you automate the title for many site names, and &num_0 and &num_1 lets you automate the sample sizes for each graph.

Once you have created the output dataset, modified the proportions to the proper format, and created macro variables to display the sample sizes, you are now ready to graph the information. These examples are vertical bar charts (with one exception of a horizontal bar chart). In Charts #1-3, the following code is

used, along with the axis, pattern, and title statements. Except for the legend and axis statements, the rest of the code is incorporated into a macro:

```
legend1 cframe=white   cborder=black
      label=(position=(top left) 'Sites');
axis1 label=(angle=90) value=(height=0.4)
      minor=none major=(height=0.4) order=(0 to 1
      by 0.10);
axis2 label=(' ' color=white height=0.4)
      value=none minor=none major=none;
axis3 label=(' ');
axis4 label=(' ' color=white height=0.4)
order=(0 to 1 by 0.10) value=(height=0.4)
      minor=none major=(height=0.4);


%macro g(grphfile=, title=, title2=, out2=,
       psumvar=,   color2=white, d=, sumvar=,
       fmt=, crit=);
..

pattern1 v=empty  c=blue;
pattern2 v=r1 c=red;
title1 h=1 " &title ";
title2 h=0.5 " &title2 " color=&color2;
title3 h=0.5 "&&site&i (n=&num_0) and All Sites
            (n=&num_1)";

proc gchart data=g_1;
  vbar all/discrete group=&d subgroup=all
    sumvar=&sumvar sum patternid=subgroup
    space=0
    legend=legend1 axis=axis1 maxis=axis2 ;
    by grphsite;
    where grphsite=&i &crit;
    format all site&i.fmt.  &d &fmt  ;
  run;
  quit;
```

The VBAR statement has many options listed. Following is listed a brief explanation of each and how it contributes to the graph:

DISCRETE – interprets numerical values as categorical, rather than displaying the numerical codes on a continuum.
GROUP – variable of interest to be charted (e.g., age, CD4 count, etc)
SUBGROUP – subgroups within the strata of the GROUP variable. By using ALL in the VBAR statement, *and* in the subgroup option, you will produce the graphs shown in these examples.
SPACE - space between subgroups.
LEGEND – legend created in the program.
AXIS – y-axis for vertical charts, x-axis for horizontal charts.
MAXIS – midpoint axis. Axis where groups are aligned (x-axis for vertical charts, y-axis for horizontal charts).

The first chart displays age strata, by site #1 and all sites, with a legend, and sample sizes in the title.

Chart #2 – Displaying means

This second graph reports means rather than percentages. Therefore, create the output dataset from PROC MEANS. For convenience, I rounded the means to one decimal:

```
* CHART #2 - Graphing Means *;
*---------------------------;
proc means data=temp01 noprint;
  by testsite all;
  class disstat;
  var numvis98;
  output out=g3 n=n mean=mean;
  run;
data g_3;
  set g3;
  mean=round(mean,0.1);
  label mean='Mean # Office Visits in 1998';
  run;
```

Compared to Chart #1, there are some slight modifications. In the PROC SUMMARY, to get the sample size, use the variable MEAN instead of COUNT for the N option. In the VBAR statement of the PROC GCHART, your SUMVAR becomes MEAN rather than the PERCENT. Finally, you must modify the ORDER option of your axis statement to display the correct unit on your axis:

```
axis1 label=(angle=90)
 major=(height=0.25) order=(0 to 10 by 2);
```

Chart #3: Graphing subgroups within subgroups

This last example is the most challenging. There is already a subgroup in the graph - the ALL variable. But another subgroup within the ALL variable needs to be graphed as well. This graph will be reporting the distribution of the

patients' current regimen type, split up into whether the patient is naïve or not naïve to the regimen, comparing each site to the whole.

When running PROC FREQ to create your output dataset, create a crosstab of regimen type (REGTYPE) by whether or not the patient is naïve (REGNAIVE).

```
*  GRAPH #3 :  Subgrouping  ;
*--------------------------;
proc freq data=temp01 noprint;
  by testsite all;
  tables regtype*regnaive/missing out=g4;
  run;
```

The following DATA step creates a variable SUB_G, to represent each stratum of REGTYPE: Creating one variable to capture these 4 levels simplifies the graphing procedure :

```
  data g4a;
    set g4 ;
    sub_g=regnaive+2*all;
    run;

/* SUB_G=0 if not naive, site-specific
 SUB_G=1 if not naive, all sites
 SUB_G=2 if naive, site specific
 SUB_G=3 if naive, all sites  */
```

Formats need to be created for SUB_G, to be displayed in the legend. Remember that formats are limited to only 16 characters.

```
proc format;
  value nv1_f    0="#1, not naive"
                 1="#1, naive"
                 2="All, not naive"
                 3="All, naive";
  value nv5_f    0="#5, not naive"
/*etc.. .*/
 run;
```

Other variations of chart data:

- If there are overlapping categories that do not add up to 100%, use separate tables statements in PROC FREQ, creating an output dataset for each table, concatenate these output datasets into one, and summarize these categories into one variable.
- If these overlapping categories have different denominators, create extra macro variables to display these sample sizes.
- If there are categories with 0 count, use the SPARSE option in PROC FREQ.
- Finally, if site numbers are sequential, use %DO to run a macro loop.

Conclusion

When reporting site data, there are many details to consider a priori:

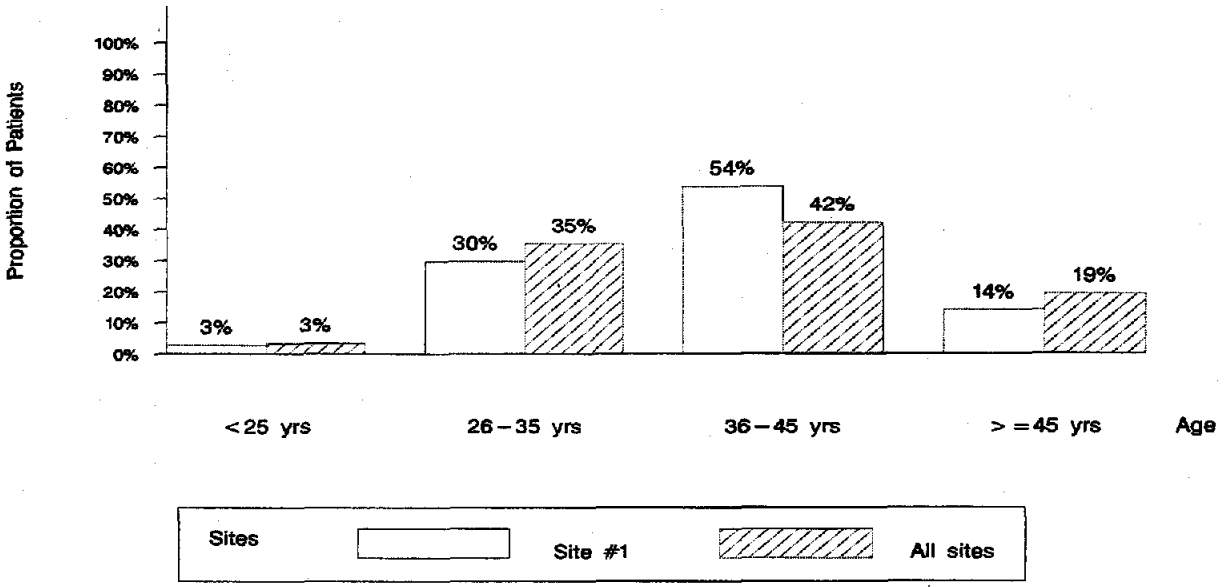
- How the data will be reported: proportions, frequencies, means
- How the data will be presented – tables or graphs. The techniques described in this paper are also required to report the information in tables (except the graph part).
  - Comparing one site to the whole – use the DATA step to make this comparison simple.
  - Displaying sample sizes for each graph/table – when reported with proportions, gives the reader an idea of the counts.
  - Reporting subgroups within subgroups – pay close attention to the extra steps required to solve this problem.

Contact for author at:

Jenny Yu, MPH
Statistical Analyst
Lewin-TAG, Inc.
490 Second St, Ste 201
San Francisco, CA 94107
(415) 495-8966 x280
jyu@lewin.com

Chart #1 Age Distribution

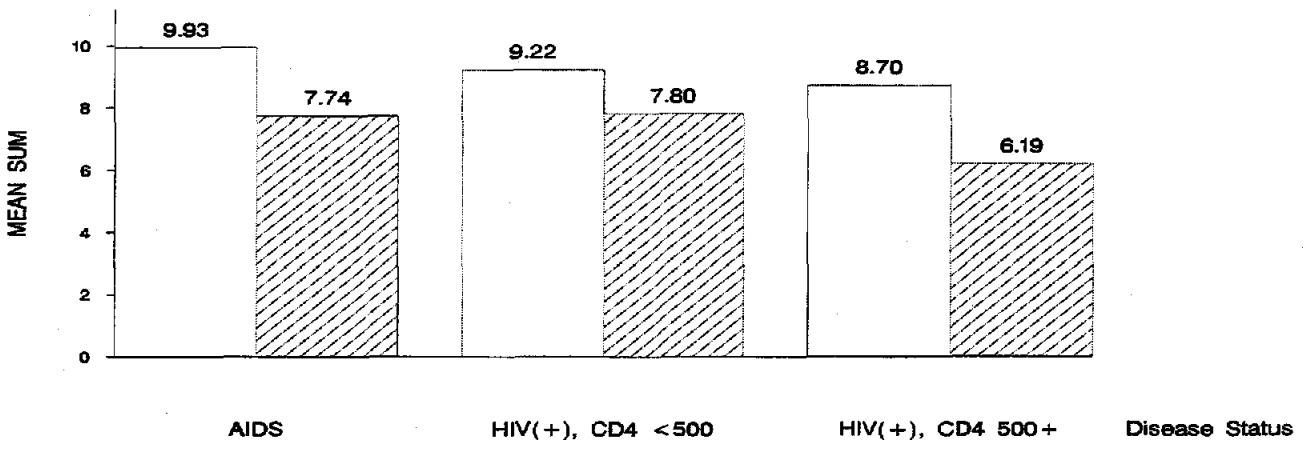
Site #1 (n=108) and All Sites (n=626)

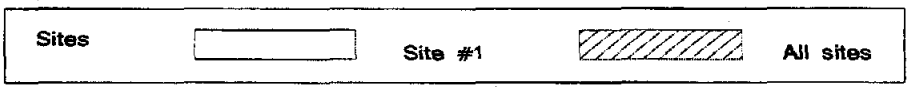


NOTE: Can put many footnotes here

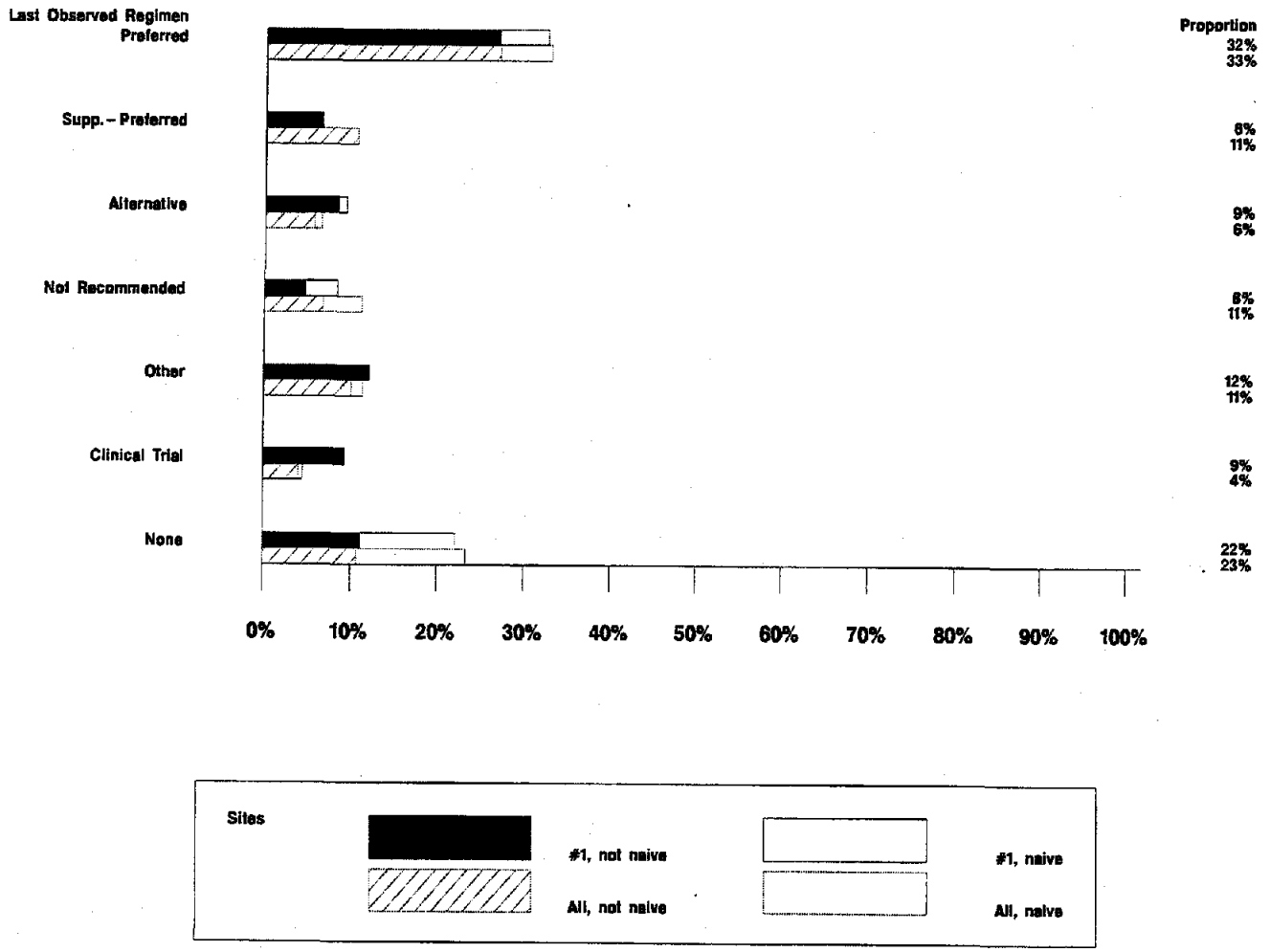
Chart #2 1998 Annual Rate of Office Visits by Disease Status* –

Site #1 (n= 28) and All Sites (n= 22)





NOTE: Can put many footnotes here

Chart #3 Antiretroviral Regimen Classification

Site #1 (n=108) and All Sites (n=628)