

# Producing Clinical Laboratory Shift Tables From ADaM Data

Rao Bingi, Octagon Research Solutions, Wayne, PA

## ABSTRACT

Shift tables are very important in clinical trial studies. A shift table is a table that displays the number of subjects who are low, normal or high at baseline and the shift at post-dose. Writing code for a shift table is an easier task, if ADaM datasets are used rather than raw datasets. It represents cross frequencies of baseline values with post-dose values. The produced shift table gives safety information of the subjects during the clinical study.

This paper provides information about producing laboratory shift tables from ADaM datasets in clinical trial studies.

The SAS product used in this paper is SAS® BASE version 9.2 running on a WINDOWS environment.

## INTRODUCTION

All safety related data is collected in a clinical trial study for safety analysis. The laboratory data is one of the most important parts of safety analysis in any drug discovery study. The data mapping team and SAS programmers in the clinical programming department support the clinical data analysis and reporting process. The data mapping team analyzes and creates SDTM datasets from raw datasets. The SAS Programming team creates the ADaM datasets and generates reports in a clinical study. ADaM data structure is such that, it has all the necessary variables needed to create reporting code and this makes the job easier for programmers.

There are several summary tables commonly used to produce laboratory results. Descriptive statistics, clinically significance, shift tables and toxicity grade summaries are some of the summary tables that are commonly used. The shift table is one of the most frequently requested in a clinical study by statisticians or clinicians.

The site that provides the laboratory results usually produces the normal ranges for numeric lab tests. The study team analyzes the lab results based on normal range and then flags or categorizes whether it is low, normal or high.

The main purpose of shift tables in any clinical trial is to determine how the categorical result varies from baseline to post-dose.

### Figure 1 Sample report from shift table program

Clinicians are interested in seeing how the hemoglobin value shifts from baseline to post-dose in two different treatment groups. The report below represents the number of subjects who are low, normal, or high at baseline and at post-dose together.

	Treatment 1 (N=145) Baseline				Treatment 2 (N=199) Baseline			
Parameter Post Dose	Low	Normal	High	Total	Low	Normal	High	Total
Hemoglobin (G/L)								
Low	13 (9.4%)	4 (2.9%)	0 (0.0%)	17 (12.2%)	5 (2.7%)	6 (3.3%)	0 (0.0%)	11 (6.0%)
Normal	12 (8.6%)	110 (79.1%)	0 (0.0%)	122 (87.8%)	3 (1.6%)	169 (91.8%)	1 (0.5%)	173 (94.0%)
High	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
Total	25 (18.0%)	114 (82.0%)	0 (0.0%)	139 (100.0%)	8 (4.3%)	175 (95.1%)	1 (0.5%)	184 (100.0%)

#### Data used to produce Shift Table

The following ADSL and ADLB ADaM datasets are used in creation of shift table and help with understanding the programming process.

#### ADSL

	USUBJID	SAFFL	TRT01PN	TRT01P
1	US50-B01-012-0001	Y	2	Treatment 2
2	US50-B01-012-0002	Y	2	Treatment 2
3	US50-B01-012-0003	Y	2	Treatment 2
4	US50-B01-012-0004	Y	2	Treatment 2
5	US50-B01-012-0005	Y	1	Treatment 1

#### ADLB

	USUBJID	PARAM	ANLFL	AVALCAT1	AVALCT1N	BASECAT1	BASECT1N	VISIT	VISITNUM	ATPTN	ATPT	SAFFL	TRT01PN
1	US50-B01-012-0001	HEMOGLOBIN (g/L)	Y		.		.	SCREENING	0	1	Screening	Y	2
2	US50-B01-012-0001	HEMOGLOBIN (g/L)	Y		.		.	DAY 1	1	2	Pre-dose	Y	2
3	US50-B01-012-0001	HEMOGLOBIN (g/L)	Y	LOW	1			DAY 1	1	3	Baseline	Y	2
4	US50-B01-012-0001	HEMOGLOBIN (g/L)	Y	NORMAL	2	LOW		DAY 1	1	4	Post-dose	Y	2
5	US50-B01-012-0002	HEMOGLOBIN (g/L)	Y		.		.	SCREENING	0	1	Screening	Y	2

#### UNDERSTANDING THE PROCESS

It is critical to understand which datasets and variables are to be used in creating the shift table. To get a better understanding let's look at the ADLB dataset above. The key variables are trt01pn, param, atptn, avalct1n, and basecat1n. These variables are used to compare the values of hemoglobin at baseline and at post-dose. The variables AVALCAT1 and BASECAT1 have values 'LOW', 'NORMAL' or 'HIGH'. The corresponding numeric variables are AVALCAT1N and BASECAT1N. The clinician wants to see how many subjects who had a 'NORMAL' at Baseline and 'NORMAL' at post-dose. Subjects keep their values ('LOW','NORMAL','HIGH') unchanged from baseline to post-dose then that drug is a good indicator for safety. This can be demonstrated by using a shift table.

## STEPS TO CREATE A LAB SHIFT TABLE

Generating a shift table involves calculating Big N's, performing Freqs to get lab counts, calculating denominator and producing output by using PROC REPORT. The first step is to make sure that ADaM dataset ADLB (Lab Analysis dataset) has variables lab result (AVALCAT1) and baseline result (BASECAT1) and their corresponding numeric variables (AVALCAT1N, BASECAT1N). If the baseline value is not defined for each record in ADLB then create a baseline dataset and merge with the ADLB dataset. Each lab test should have an original result category and baseline result category.

The subsequent steps are involved in creating above Shift table.

### STEP I: CALCULATING BIG N

```
libname derlib "c:\shift";

%global n1 n2;

data adsl1;
    set derlib.adsl;
    where saffl='Y';
run;

%macro pop();

    %do i=1 %to 2;
        proc sql noprint;
            select count(distinct usubjid) into :n&i
            from adsl1
            where trt01pn=&i;
            quit;
        %end;

%mend pop;

%pop;
%put &n1;
%put &n2;
```

The ADL is the analysis dataset for each subject level. Initially subset the ADL dataset with safety subjects only. Count number of subjects participating in each treatment group. The treatment groups are Treatment 1 and Treatment 2 in this study. In this case the counts are: n1= 145 and n2=199. These counts are used to display total number of subjects in each treatment group at top level of the report.

### STEP II: REARRANGING LAB DATA

```
data lb1(keep=usubjid param paramcd atptn atpt visit visitnum avalctln
    avalcat1 basecat1 basectln trt01pn);
    set derlib.adlb;
    where saffl='Y' and anlfl='Y' and lbcat='HEMATOLOGY' and paramcd='HGB'
        and 4 le atptn and not missing(avalctln) and not missing(basectln) ;
run;

proc format;
    value avalc
        1="LOW"
        2="NORMAL"
        3="HIGH"
        4="TOTAL" ;
    value atpt
```

```

                                4="Post-dose";
run;

proc sort data=lb1;
    by usubjid param atptn;
run;

data lb1;
    set lb1;
    output;
    avalct1n=4;
    output;
run;

data lb1;
    set lb1;
    output;
    basect1n=4;
    output;
run;

```

Subset the ADLB dataset for Hemoglobin Lab parameters, safety subjects where analysis flag is equal to 'Y' and at post-dose.

To display data values in the order of 'Low', 'Normal', 'High and 'Total', then these values need to be defined in the format procedure as shown in the code above. Attach this format to the final dataset in the final step. The orders of these values are shown in the final report. The horizontal total and vertical total is calculated in above last two data steps.

### STEP III: PERFORMING COUNT

Now to see the total number of subjects who are low, normal, or high and total at baseline and at post-dose for each treatment group. This can be accomplished by using PROC FREQ as shown below:

```

proc freq data=lb1 noprint;
    tables trt01pn*param*atptn*avalct1n*basect1n /
    out=count(drop=percent where=(not missing(avalct1n) and not
    missing(basect1n)));
run;

```

The above code gives count for each category and horizontal, vertical totals at baseline and shift at post-dose for each treatment group, A portion of output from "count" dataset is shown below:

	TRT01PN	PARAM	ATPTN	AVALCT1N	BASECT1N	COUNT
1	1	HEMOGLOBI	4	1	1	13
2	1	HEMOGLOBI	4	1	2	4
3	1	HEMOGLOBI	4	1	4	17
4	1	HEMOGLOBI	4	2	1	12
5	1	HEMOGLOBI	4	2	2	110

### STEP IV: CREATING DUMMY DATASET & MERGING WITH ACTUAL DATA

```

proc sort data=lb1 out=dummy(keep=param) nodupkey;
by param;
run;
data dummy1;
    set dummy;
    by param;

```

```

do trt01pn=1 to 2;
  do atptn=4 to 4;
    do avalct1n=1 to 4;
      do basect1n=1 to 4;
        output;
      end;
    end;
  end;
end;
run;

```

Creat a dummy dataset that contains a row for each Base value (1,2,3 and 4) and corresponding post-dose value (1,2,3 and 4). This is nothing but creating a frame.

Let's see how this dummy dataset looks:

	PARAM	TRT01PN	ATPTN	AVALCT1N	BASECT1N
1	HEMOGLOBIN (g/L)	1	4	1	1
2	HEMOGLOBIN (g/L)	1	4	1	2
3	HEMOGLOBIN (g/L)	1	4	1	3
4	HEMOGLOBIN (g/L)	1	4	1	4
5	HEMOGLOBIN (g/L)	1	4	2	1
6	HEMOGLOBIN (g/L)	1	4	2	2
7	HEMOGLOBIN (g/L)	1	4	2	3
8	HEMOGLOBIN (g/L)	1	4	2	4

```

proc sort data=count;
by param trt01pn atptn avalct1n basect1n;
run;

proc sort data=dummy1;
by param trt01pn atptn avalct1n basect1n;
run;
data combined;
merge dummy1 count;
by param trt01pn atptn avalct1n basect1n;
run;

```

Merge the dummy data with the actual data to get continuous data like base values (1,2,3 and 4) for each post-dose value.

This is how the combined dataset looks:

	PARAM	TRT01PN	ATPTN	AVALCT1N	BASECT1N	COUNT
1	HEMOGLOBIN (g/L)	1	4	1	1	13
2	HEMOGLOBIN (g/L)	1	4	1	2	4
3	HEMOGLOBIN (g/L)	1	4	1	3	.
4	HEMOGLOBIN (g/L)	1	4	1	4	17

## STEP V: REARRANGING AND TRANSPOSING DATA

```
data combined;
  set combined;
  by param trt01pn atptn avalct1n basect1n;
  if trt01pn=2 and basect1n=1 then basect1n=5;
  else if trt01pn=2 and basect1n=2 then basect1n=6;
  else if trt01pn=2 and basect1n=3 then basect1n=7;
  else if trt01pn=2 and basect1n=4 then basect1n=8;
run;
proc sort data=combined;
  by param atptn avalct1n basect1n;
run;
proc transpose data=combined out=final(drop=_name_ _label_) prefix=col;
  by param atptn avalct1n;
  var count;
  id basect1n;
run;
```

Assign baseline categories to 5,6,7,8 for treatment group 2, to see the two-treatment group's baseline values on the top of the report.

Finally transpose the data by count and basect1n as an id.

The output of the transposed data is as follows:

	PARAM	ATPTN	AVALCT1N	COL1	COL2	COL3	COL4	COL5	COL6	COL7	COL8
1	HEMOGLOBIN (g/L)	4	1	13	4	.	17	5	6	.	11
2	HEMOGLOBIN (g/L)	4	2	12	110	.	122	3	169	1	173
3	HEMOGLOBIN (g/L)	4	3	.	.	.	.	.	.	.	.
4	HEMOGLOBIN (g/L)	4	4	25	114	.	139	8	175	1	184

## STEP VI : CALCULATING DENOMINATOR COUNT

```
data denomi;
  set final;
  by param;
  where avalct1n=4;
  retain rownum;
  if first.param then rownum+1;
run;

%global l1 r1 ;

Data _null_;
  Set denomi;

  if n(col4)=1 then
  Call symput('l1'|trim(left(rownum)),trim(left(col4)));
  else Call symput('l1'|trim(left(rownum)),0);

  if n(col8)=1 then
  Call symput('r1'|trim(left(rownum)),trim(left(col8)));
  else Call symput('r1'|trim(left(rownum)),0);
run;
```

The denominator count for each treatment group in shift table is total number of subjects who had both baseline and post-dose records per each lab test. This can be accomplished by using above code. In this case the values of total (avalct1n=4) and columns 4 and 8 are assigned to macro variables 'l1' and 'r1'. These counts('l1','r1') are used in the denominator portion of the total number of subjects in each cell to calculate percentages. In this case the counts are: l1=139 and r1=184.

## STEP VII: CALCULATING PERCENTAGE

Percentages for each cell are the total number of subjects divided by denominator count (l1 or r1) in that treatment group and multiplied by 100. This calculation is as follows:

```

data final1(drop=col1 col2 col3 col4 col5 col6 col7 col8 i);
  length tot1 tot2 tot3 tot4 tot5 tot6 tot7 tot8 $15;
  set final;
  array treat{8} col1 col2 col3 col4 col5 col6 col7 col8;
  array tot{8} tot1 tot2 tot3 tot4 tot5 tot6 tot7 tot8;
  do i=1 to 8;
    if i <= 4 then do;
      if treat{i} ne . then tot{i}=trim(left(put(treat{i},best.))) || " (" ||
        trim(left(put((treat{i}/&l1)*100,5.1))) || "%)"; end;
    else if i > 4 then do;
      if treat{i} ne . then tot{i}=trim(left(put(treat{i},best.))) || " (" ||
        trim(left(put((treat{i}/&r1)*100,5.1))) || "%)"; end;
      if treat{i} eq . and (&l1 ne 0 or &r1 ne 0) then tot{i}="0 (0.0%) ";
    end;
  end;

run;

data final1;
  set final1;
  length atpt $50 avalcat1 $15;
  atpt=strip(put(atptn,atpt.));
  avalcat1=strip(put(avalct1n,avalc.));

run;

data final2;
  length avalcat1 $20;
  set dummy
    final1;
  by param;
  if first.param then avalcat1=param;
  avalcat1=propcase(avalcat1);

run;

```

## STEP VIII: PRODUCING FINAL REPORT WITH PROC REPORT

The "final" dataset is ready to generate the shift table by using PROC REPORT. Let's look at the final code and then output:

```

ods rtf file = "c:\shift\shift.rtf" bodytitle;

proc report data=final2 missing center nowd headline headskip split='|'
  style(report)={outputwidth=9 in}
style(header)={just=center}/*{protectspecialchars=off}*/;
  column param atptn avalcat1
    ("Treatment 1|(N=%trim(&n1))|Baseline|" tot1 tot2 tot3 tot4 )
    ("Treatment 2|(N=%trim(&n2))|Baseline|" tot5 tot6 tot7 tot8);
  define param / noprint order ;
  define atptn / order noprint ;
  define avalcat1 / display "Parameter|Post Dose" style=[just=left
    cellwidth=0.75in asis=on];

```

```

define tot1 /display "Low" style(column)=[just=center cellwidth=0.5in];
define tot2 /display "Normal" style(column)=[just=center cellwidth=0.5in];
define tot3 /display "High" style(column)=[just=center cellwidth= 0.5in];
define tot4 /display "Total" style(column)=[just=center cellwidth= 0.5in];
define tot5 /display "Low" style(column)=[just=center cellwidth=0.5in];
define tot6 /display "Normal" style(column)=[just=center cellwidth=0.5in];
define tot7 /display "High" style(column)=[just=center cellwidth= 0.5in];
define tot8 /display "Total" style(column)=[just=center cellwidth= 0.5in];

compute before atptn / style=[just=center cellwidth= 0.5in];
  line " ";
endcomp;

break after param/page;

title 'Shift from Baseline in Hemoglobin By Treatment';
title2 'Safety Population';

run;

ods rtf close;

```

**STEP IX: FINAL SHIFT TABLE**

***Shift from Baseline in Hemoglobin By Treatment  
Safety Population***

Parameter Post Dose	Treatment 1 (N=145) Baseline				Treatment 2 (N=199) Baseline			
	Low	Normal	High	Total	Low	Normal	High	Total
Hemoglobin (G/L)								
Low	13 (9.4%)	4 (2.9%)	0 (0.0%)	17 (12.2%)	5 (2.7%)	6 (3.3%)	0 (0.0%)	11 (6.0%)
Normal	12 (8.6%)	110 (79.1%)	0 (0.0%)	122 (87.8%)	3 (1.6%)	169 (91.8%)	1 (0.5%)	173 (94.0%)
High	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
Total	25 (18.0%)	114 (82.0%)	0 (0.0%)	139 (100.0%)	8 (4.3%)	175 (95.1%)	1 (0.5%)	184 (100.0%)

## CONCLUSION

Shift tables are very helpful in observing changes from one time point to another time point. We looked at the logic of shift table, SAS code and output to accomplish the same.

## REFERENCES

CDISC Analysis Data Model Version 2.1

SAS V9.2 Online Documentation

## CONTACT INFORMATION

Author Name	Rao Bingi
Company	Octagon Research Solutions, Inc
Address	585 E Swedesford Road
City State Zip	Wayne, PA 19087
Work Phone	(610) 535-6500 Ext 5573
Email	<a href="mailto:rbingi@octagonresearch.com">rbingi@octagonresearch.com</a>

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.

## APPENDIX 1 - Code to Produce Lab Shift Table

```
libname derlib "c:\shift";
options validvarname=upcase nonumber nodate nofmterr nocenter;

%global n1 n2;
data adsl1;
    set derlib.adsl;
    where saffl='Y';
run;
%macro pop();
%do i=1 %to 2;
    proc sql noprint;
        select count(distinct usubjid) into :n&i-:n&i
            from adsl1
            where trt01pn=&i;
    quit;
%end;
%mend pop;
%pop;
%put &n1; %put &n2;

data lb1(keep=usubjid param paramcd atptn atpt visit visitnum avalctln
    avalcat1 basecat1 basectln trt01pn);
    set derlib.adlb;
    where saffl='Y' and anlfl='Y' and lbcat='HEMATOLOGY' and paramcd='HGB'
        and 4 le atptn and not missing(avalctln) and not missing(basectln);
run;
proc format;
    value avalc
        1="LOW"
        2="NORMAL"
        3="HIGH"
        4="TOTAL" ;
    value atpt
        4="Post-dose";
run;
proc sort data=lb1;
by usubjid param atptn;
run;
data lb1;
    set lb1;
    output;
    avalctln=4;
    output;
run;
data lb1;
    set lb1;
    output;
    basectln=4;
    output;
run;
proc freq data=lb1 noprint;
    tables trt01pn*param*atptn*avalctln*basectln / out=count(drop=percent
        where=(not missing(avalctln) and not missing(basectln)));
run;
proc sort data=lb1 out=dummy(keep=param) nodupkey;
    by param;
run;
data dummy1;
    set dummy;
    by param;
    do trt01pn=1 to 2;
        do atptn=4 to 4;
```

```

                do avalctln=1 to 4;
                    do basectln=1 to 4;
                        output;
                    end;
                end;
            end;
        end;
run;

proc sort data=count;
    by param trt01pn atptn avalctln basectln;
run;
proc sort data=dummy1;
    by param trt01pn atptn avalctln basectln;
run;
data combined;
    merge dummy1 count;
    by param trt01pn atptn avalctln basectln;
run;
data combined;
    set combined;
    by param trt01pn atptn avalctln basectln;
    if trt01pn=2 and basectln=1 then basectln=5;
    else if trt01pn=2 and basectln=2 then basectln=6;
    else if trt01pn=2 and basectln=3 then basectln=7;
    else if trt01pn=2 and basectln=4 then basectln=8;
run;
proc sort data=combined;
    by param atptn avalctln basectln;
run;
proc transpose data=combined out=final(drop=_name_ _label_) prefix=col;
    by param atptn avalctln;
    var count;
    id basectln;
run;
data denomi;
    set final;
    by param;
    where avalctln=4;
    retain rownum;
    if first.param then rownum+1;
run;
%global l1 r1 ;

Data _null_;
    Set denomi;
    if n(col4)=1 then
        Call symput('l' || trim(left(rownum)), trim(left(col4)));
    else Call symput('l' || trim(left(rownum)), 0);

    if n(col8)=1 then
        Call symput('r' || trim(left(rownum)), trim(left(col8)));
    else Call symput('r' || trim(left(rownum)), 0);
run;

options linesize=256;
data final1(drop=col1 col2 col3 col4 col5 col6 col7 col8 i);
    length tot1 tot2 tot3 tot4 tot5 tot6 tot7 tot8 $15;
    set final;
    array treat{8} col1 col2 col3 col4 col5 col6 col7 col8;
    array tot{8} tot1 tot2 tot3 tot4 tot5 tot6 tot7 tot8;
    do i=1 to 8;
        if i <= 4 then do;

```

```

        if treat{i} ne . then tot{i}=trim(left(put(treat{i},best.))) || " (" ||
            trim(left(put((treat{i}/&l1)*100,5.1))) || "%)"; end;
    else if i > 4 then do;
        if treat{i} ne . then tot{i}=trim(left(put(treat{i},best.))) || " (" ||
            trim(left(put((treat{i}/&r1)*100,5.1))) || "%)"; end;
        if treat{i} eq . and (&l1 ne 0 or &r1 ne 0) then tot{i}="0 (0.0%) ";
    end;
run;
data final1;
    set final1;
    length atpt $50 avalcat1 $15;
    atpt=strip(put(atptn,atpt.));
    avalcat1=strip(put(avalct1n,avalc.));
run;
data final2;
    length avalcat1 $20;
    set dummy
        final1;
    by param;
    if first.param then avalcat1=param;
    avalcat1=propcase(avalcat1);
run;

ods rtf file = "c:\shift\shift.rtf" bodytitle;

proc report data=final2 missing center nowd headline headskip split='|'
    style(report)={outputwidth=9 in} style(header)={just=center};
column param atptn avalcat1
    ("Treatment 1|(N=%trim(&n1))|Baseline|" tot1 tot2 tot3 tot4 )
    ("Treatment 2|(N=%trim(&n2))|Baseline|" tot5 tot6 tot7 tot8);
define param / noprint order ;
define atptn / order noprint ;
define avalcat1 / display "Parameter| Post Dose"
    style=[just=left cellwidth=0.75in asis=on] ;
define tot1 /display "Low"
    style(column)=[just=center cellwidth=0.5in] ;
define tot2 /display "Normal"
    style(column)=[just=center cellwidth=0.5in] ;
define tot3 /display "High"
    style(column)=[just=center cellwidth= 0.5in] ;
define tot4 /display "Total"
    style(column)=[just=center cellwidth= 0.5in] ;
define tot5 /display "Low"
    style(column)=[just=center cellwidth=0.5in] ;
define tot6 /display "Normal"
    style(column)=[just=center cellwidth=0.5in] ;
define tot7 /display "High"
    style(column)=[just=center cellwidth= 0.5in] ;
define tot8 /display "Total"
    style(column)=[just=center cellwidth= 0.5in] ;

compute before atptn / style=[just=center cellwidth= 0.5in] ;
    line " ";
endcomp;

break after param/page;

title 'Shift from Baseline in Hemoglobin By Treatment';
title2 'Safety Population';
run;

ods rtf close;

```