

Data Visualization in Real World Data (RWD) and Health Economics Outcomes Research (HEOR) using Statistical Analysis Software (SAS)

Swapna Ambati, Eliassen Group LLC

ABSTRACT

Real-world data (RWD) analytics in Health economics outcomes research (HEOR) generates real-world evidence (RWE) to provide useful information that helps in decision-making in the healthcare setting and healthcare-related investment. RWE compliments clinical trial findings and may help fill knowledge gaps related to how a medication is used in the real world. Data visualization techniques help explore, synthesize, and communicate the results of research studies. In this paper, we will explore data visualization methods using SAS analysis software specifically for standard HEOR analysis such as patient cohort extraction steps, demographics, comorbidities, cost, utilization, treatment patterns, medication, adherence, incidence, prevalence, and survey related analysis. We will generate patient disposition charts, flow charts, bar charts, histograms, tree maps, bubble charts, pie charts, line charts, scatter plots, polar style chart, calendar charts, sankey style plots, 3D maps, stacked charts, butterfly charts, and so forth to study HEOR results specifically. This paper will focus on applying simpler techniques in SAS for visualizing the HEOR type of data. These techniques can be further updated to create much more advanced visualizations, automated macros, and to create the HEOR dashboards.

INTRODUCTION

HEOR provides important information regarding patient outcomes that helps us understand and make healthcare coverage and access decisions. HEOR includes studies in the fields of real-world evidence, patient-reported outcomes, meta-analyses, observational studies, post-market studies and clinical outcomes studies, and so forth. HEOR outcomes include cost analysis, cost burden, cost minimization, cost-effectiveness, cost-utility outcomes measures, clinical, demographic, psychologic, social factors, health-related Quality of Life (HR-QoL), satisfaction, compliance preferences, convenience, competitive market measures, and so forth [1].

Data Visualization (DV) is a graphical representation of the data analysis. HEOR data uses many Big Data databases which involves analyzing huge patient cohorts and hence DV will help specifically as an efficient method of communicating the results. Many HEOR studies also study longitudinal data and hence DV will be a very efficient tool to understand the time series data and create predictive models. In this paper, we will study DV in the random sample HEOR type of data by using Statistical Analysis Software (SAS). SAS software was developed by the SAS Institute and is used to access, manage, analyze, and report data. In this paper, we will apply SAS procedures and syntax from the SAS website [2], SAS support, [8] and SAS Blogs [9] to provide the DV results in HEOR. We will mostly use PROC SGPLOT to create the visualizations. Other SAS procedures that we will explore include PROC GCHART, PROC SGPANEL, PROC GMAP, PROC SGSCATTER, PROC SGRENDER, PROC GRADAR, PROC SURVEYLOGISTIC, PROC SURVEYFREQ, etc. we will also use ODS Graphics, PROC TEMPLATE, annotate datasets, and goptions to customize the options to report the data visualizations. PROC SGPANEL can be used to create a panel of graph cells. SAS Studio offers inbuilt SAS graphical procedures that can be also used to create graphical reports. SAS website [2] can be referred to for general syntax and all the options available for each of these SAS procedures. In this paper, we will either have sample data in excel that can be imported as a SAS dataset using PROC IMPORT procedure or create a SAS data set with variables having random values using RAND function in SAS. We will analyze either a patient level dataset with one record for each patient or a summary dataset that might show the sum, means, or any other summary stat associated with the variable of interest. The data reported in this paper is from a random sample and hence it is an example of how we can report the actual data but this paper data is not meant for drawing any conclusions.

The Statistical data will have continuous variables that are quantitative such as age, height, weight, laboratory measurements, length of stay (LOS), etc. while categorical data includes variables such as gender (nominal) and comorbidity Score (ordinal) [3]. It is important to use the correct data and the correct methods in visualizing the data, so the charts are not misleading [4][5]. When performing comparison among variables, we will use charts such as using bar graphs with same scales or column charts, etc., and while we will use line charts, column charts, calendar charts, etc. to understand comparison and change over time. To study and visualize the relationship among variables, we will use a scatter plot, bubble chart and, so forth. Histograms, line histograms, scatter and area charts will be used to understand the distribution of variables. The composition can be studied using display charts such as stacked bar charts, stacked area charts, pie charts. A Waterfall chart can be used to understand the distribution and percentage change from baseline [6] [7]. In this paper, we will study several data visualization examples of composition and distribution of one variable in the sample cohort patients, and at the same time, we will also perform a comparison, relation, composition, and distribution among more than one variable in the sample patient cohort.

PATIENT DISPOSITION

Most HEOR studies involve the creation of the patient cohort population of interest. The patient population is selected sometimes based on the diagnosis or sometimes based on the treatment received or procedure performed and so forth at a certain time. The patient population is often shown as a step-by-step selection process. This can be visualized in a patient disposition or a flow chart. The sample dataset is shown in Table 1. In Table 1 we have the patient selection in selection_step variable, frequency in N variable, percentage of main steps in PCT, percentages of sub-steps under PCT2. X_text and Y_text are variables imputed to represent X and Y coordinates on the graph where the selection_step variable should be shown. The variables with prefixes X_vector and Y_vector represent where the arrow starts and ends in the flow chart. Notice that step 4 is repeated in the sample dataset (highlighted) is because this step is connected to two different sub-steps with arrows directing to these two sub-steps. PROC TEMPLATE is used to customize the look of the flow chart which is displayed using PROC SGRENDER. Figure 1 shows the outputs as a patient disposition flow chart.

obs	Selection_step	N	PCT	PCT2	X_text	Y_text	X_vector_srt	X_vector_end	Y_vector_srt	Y_vector_end
1	1. Total patients in the database in study period	50000	100		500	100	800	800	90	85
2	2. Patients diagnosed with the disease in the study	15000	30		500	80	800	800	70	65
3	3. Patient enrolled	8000	16		500	60	800	800	50	45
4	4. Patients with eligible criteria (Final patient cohort)	5000	10		500	40	800	400	30	25
4	4. Patients with eligible criteria (Final patient cohort)	5000	10		500	40	800	1000	30	25
5	4a. Treatment A	2000		40	100	20				
	4b. Treatment B	1000		20	800	20				

Table 1: Selection step dataset with associated X and Y coordinates for text and arrow positions

SAS code to create a flow chart for sample patient cohort creation – (Table 1, Figure 1):

EXAMPLE 1:

/ Create text with the step description and the associated frequency and percentage in one variable */*

```
data sample_data2;
set sample_data;
if obs in (1,2,3,4) then do; /*Main steps*/
grp=cats(left(trimn(selection_step||"||N||"(N), "||pct||"(%)"))); end;
else do; /*Sub steps*/
grp=cats(left(trimn(selection_step||"||N||"(N), "||pct2||"(%)"))); end;
run;
```

/ Use PROC TEMPLATE to create text plot and specify X and Y coordinates from the above dataset*/*
/ Remove X and Y axis with display=none option */*

```
/* Overlay with vector plot having arrows */
```

```
proc template;  
define statgraph textplot;  
  begingraph; entrytitle "Patient Disposition Flow Chart"; layout overlay / xaxisopts=(display=none)  
  yaxisopts=(display=none) ;  
  textplot x=x_text y=y_text text=grp / name='textplot1' display=all  
  textattrs=(weight=bold) fillattrs=(transparency=0.9)  
  position=right splitpolicy=split splitchar="*";  
  vectorplot xorigin=x_vector_srt yorigin=y_vector_srt x=x_vector_end y=y_vector_end /  
  lineattrs=(pattern=solid thickness=1px color=black); discretelegend 'textplot1';  
  endlayout; endgraph;  
end; run;
```

```
/* Display with the above template for your data using PROC SGRENDER */
```

```
proc sgrender data=sample_data2 template=textplot;  
run;
```

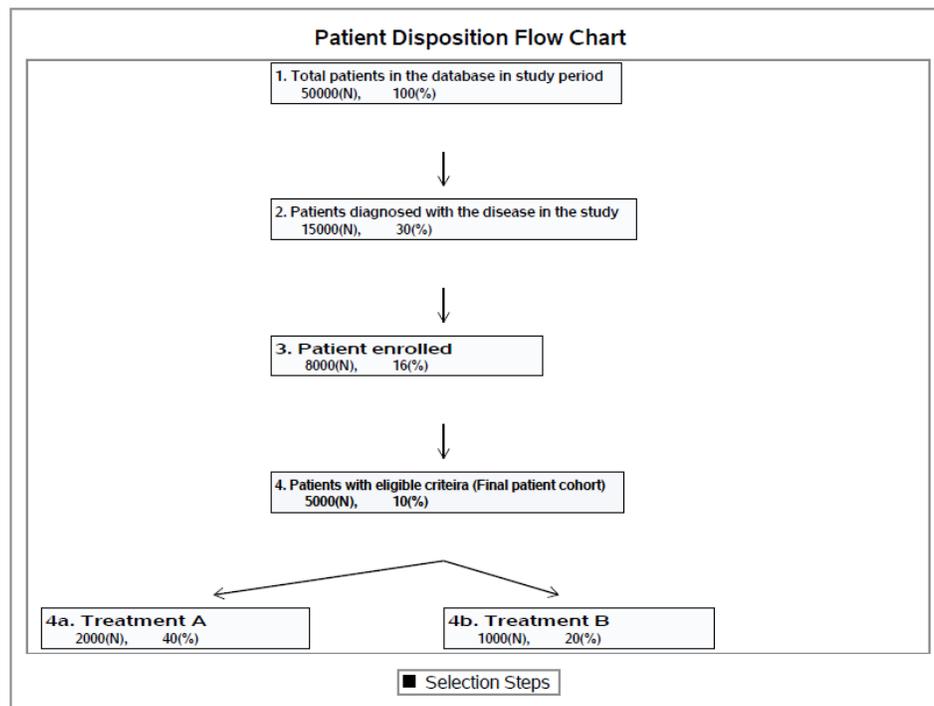


Figure 1: Patient disposition flow chart showing selection process for cohort patients in a study.

DEMOGRAPHICS

Demographics in HEOR studies provide us with statistical data that helps us understand the population of interest. In the HEOR studies, some of the common demographic characteristics include age groups, gender, income groups, geographical location, race, ethnicity, and type of insurance. HEOR projects may study the disparities among different demographic groups so that they can understand the specific demographic groups that are disproportionately impacted by the burden of disease or the burden of cost when compared to the overall population. In this paper, we will analyze some possible ways of visualizing the demographic data of some random sample patients.

AGE

Sample demographic patient-level data is shown in Table 2. Age is shown as a continuous variable. In this example, subcohort groups are assigned based on the treatment given. Age categories (categorical) will be created from the age (continuous) variable. In this example, we will visualize using a bar chart with PROC SGPLOT for both age and age group.

Patient_id	Cohort_group	State	Stateid	Age	Gender	Insurance_plan_type	Income
1	1	Maryland	24	2	F	HMO	20000
2	2	Alaska	2	5	M	PPO	50000
3	2	New Jersey	34	18	M	EPO	50000
4	1	Connecticut	9	25	M	POS	50000
5	1	Minnesota	27	44	M	PPO	50000
6	1	Colorado	8	75	M	PPO	50000
7	1	New Hampshire	33	75	M	PPO	50000
8	1	Virginia	51	2	M	PPO	20000
9	2	Delaware	10	2	M	PPO	0000
10	1	Illinois	17	5	F	HMO	0000

Table 2: Sample patient-level data with one record per patient and their corresponding demographics and subcohort group

SAS code to create a horizontal bar chart for age (Table 2, Figure 2):

Example 1:

```
Data age_data;
```

```
Set sample_data;
```

```
/* Assign categorical age groups and treatment subgroups */
```

```
If age <= 10 then age_group = "Child";
```

```
else if age <= 19 then age_group = "Teenager";
```

```
else if age <= 29 then age_group = "Young adult";
```

```
else if age <= 45 then age_group = "Adult";
```

```
else if age <= 59 then age_group = "Middle age"; else age_group = "Senior";
```

```
if cohort_group=1 then subcohort= "Treatment A";
```

```
else if cohort_group = 2 then subcohort= "Treatment B";
```

```
run;
```

```
/* Create age group plot using bar chart using PROC SGPLOT */
```

```
title "Frequency of Cohort Patients by Age Group";
```

```
proc sgplot data=age_data;
```

```
hbar age_group; /* hbar statement is used to display horizontal bar chart*/
```

```
xaxis label= "Number of Patients";
```

```
yaxis label = "Age Group";
```

```
run;
```

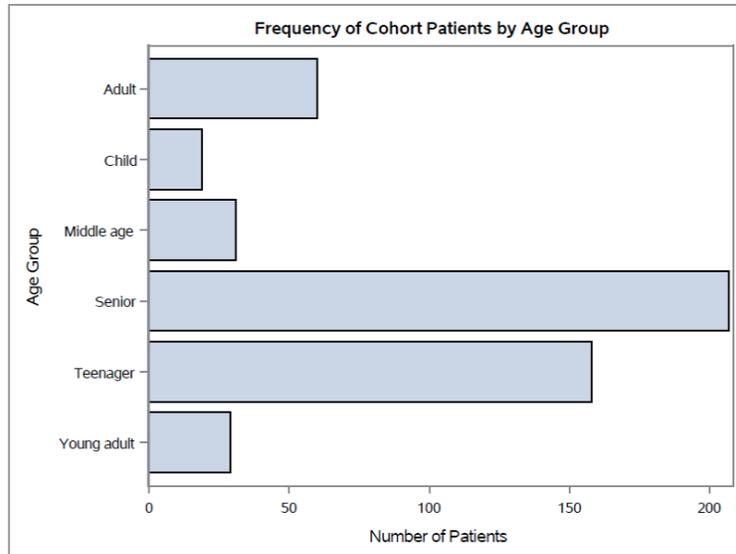


Figure 2: Horizontal bar chart with the frequency of cohort patients by age group

SAS code to create a vertical bar chart for mean age by the treatment group (Table 2, Figure 3):

Example 2:

/ Create mean age plot with age as continuous variable by subcohort groups using PROC SGPLOT */*

```

title "Mean Age of Patients by the Treatment groups";
proc sgplot data=age_data;
vbar subcohort / response=age stat=mean; /*vbar statement is used to display vertical bar chart*/
xaxis label= "Sub Cohorts";
yaxis label = "Mean Age";
run;

```

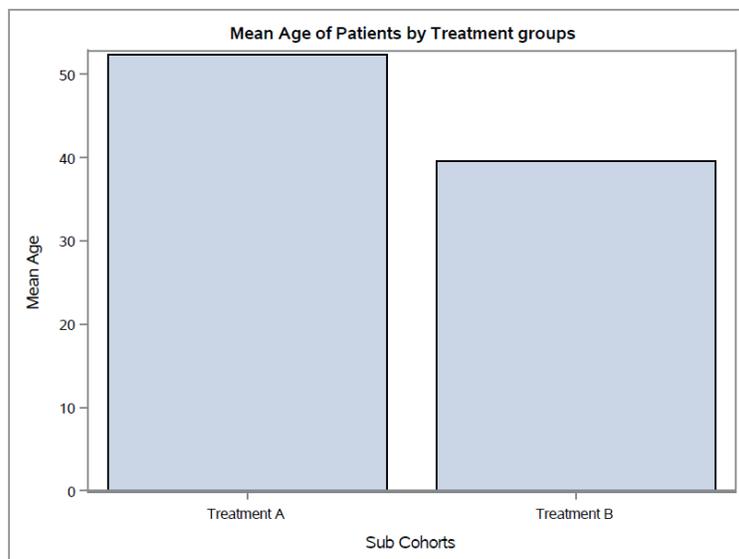


Figure 3: Vertical bar chart of mean age of the cohort patients by the treatment groups.

SAS code to create a vertical bar chart for mean age by the treatment group with precomputed limits (Table 2, Figure 4):

Example 3:

```
/* Create mean age plot using precomputed mean values and limits */
/* Use PROC MEANS procedure to get upper and lower limits */

proc means data=age_data alpha=.05 clm mean std;
class subcohort;
var age;
output out=classMean uclm=uclm lclm=lclm mean=mean;
run;

/* Create mean age with limits plot by subgroups using PROC SGPLOT */

title "Mean Age with upper and lower limits for Patients by Treatment groups";
proc sgplot data=classMean;
vbarparm category=subcohort response=mean /
limitlower=lclm limitupper=uclm;
xaxis label= "Sub Cohorts";
yaxis label = "Mean Age";
run;
```

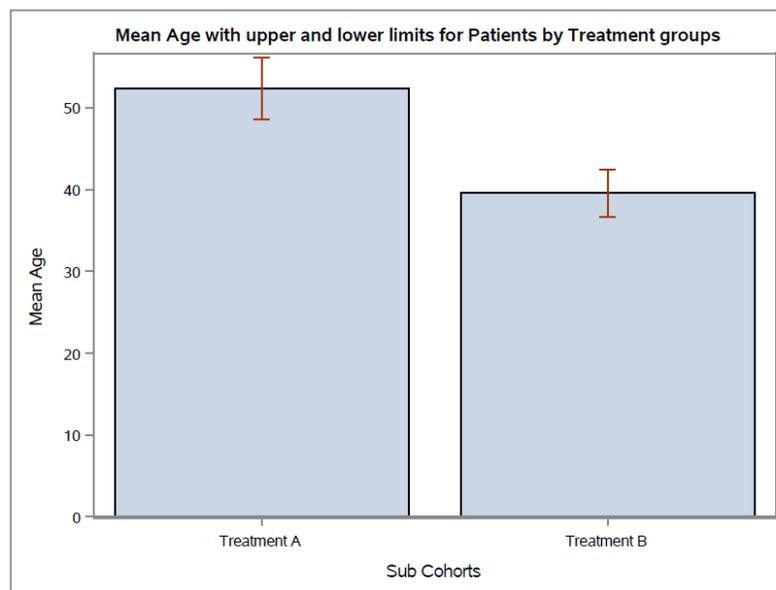


Figure 4: Vertical bar chart with precomputed mean age and precomputed limits for the cohort patients by the treatment groups

GENDER:

In this example, we will use the sample data from table 2 to create a pie chart to study the composition of gender groups within the overall cohort population using PROC GCHART. We will also create paneled charts using PROC SGPANEL to compare the age composition by different treatments with a separate panel for each gender.

SAS code to create a pie chart for gender (Table 2, Figure 5):

Example 1:

```
/* Create gender group pie chart with labels showing frequency and percentage using PROC GCHART*/  
/* Use GOPTIONS to control display settings */
```

```
Data gender_data;  
Set sample_data;  
/* Assign Gender and subgroups */  
If upcase(gender)='F' then gender_group = "Female    ";  
else if upcase(gender) = 'M' then gender_group = "Male  ";  
if cohort_group=1 then subcohort= "Treatment A";  
else if cohort_group = 2 then subcohort= "Treatment B";  
run;  
  
/* Create pie chart using PROC GCHART */  
  
title "Composition of Gender in the Overall Cohort patients";  
goptions device=png xpixels=500 ypixels=400 ctitle=gray33 ctext=gray33 htext=12pt ftext='albany  
amt/bold';  
pattern1 v=solid c=lightblue;  
pattern2 v=solid c=lightgrey;  
proc gchart data=gender_data;  
pie gender_group/ type=freq noheading /*Pie statement to create Pie chart*/  
angle=90 clockwise descending  
slice=inside value=inside percent=inside  
coutline=gray77;  
run;
```

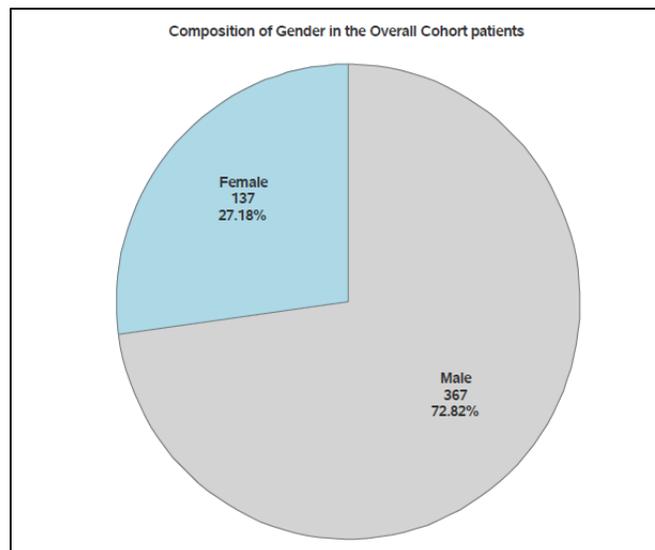


Figure 5: Pie chart showing percentage and frequency of each of the gender groups among the sample cohort patients.

SAS code to create a detailed pie chart for gender by subcohort (Table 2, Figure 6):

Example 2:

```

/* Create detailed pie chart using PROC GCHART */

title "Composition of Gender groups in each of the treatment groups";
goptions reset=all border;
quit;
proc gchart data=gender_data;
pie subcohort / detail=gender_group detail_percent=best detail_value=none detail_slice=best
detail_threshold=2 legend;
run;
quit;

```

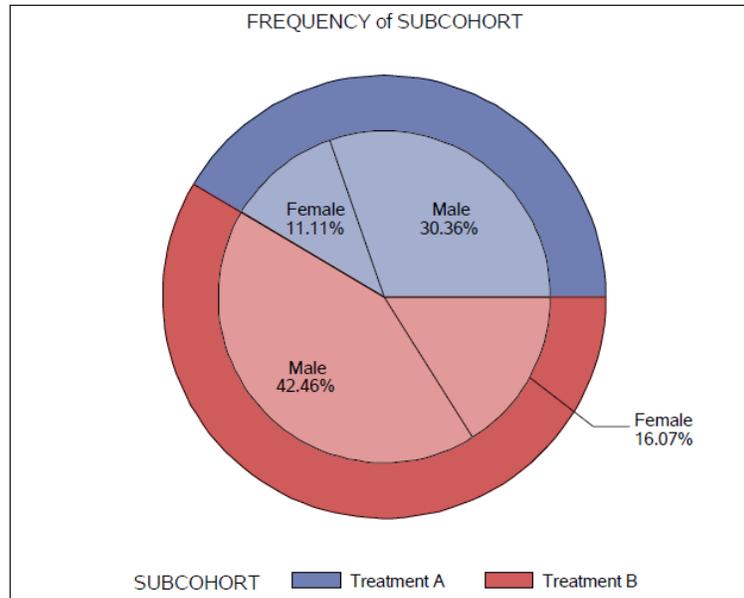


Figure 6: Detailed pie chart with percentages of gender within the treatment sub cohort groups.

SAS code to create a panel chart for gender by subcohort (Table 2, Figure 7):

Example 3:

```

/* Create mean age plot by treatment groups in separate panels for each gender using SGPANEL */

title "Mean Age of Patients by Treatment groups in separate panels for each gender";
/* Precompute mean values using PROC MEANS*/
proc means data=gender_data alpha=.05 clm mean std;
class gender_group subcohort;
var age;
output out=classMean uclm=uclm lclm=lclm mean=mean;
run;

/* Use PROC SGPANEL to display in separate panels by gender */

proc sgpanel data=classMean;
panelby gender_group;
hbarparm category=subcohort response=mean;
run;

```

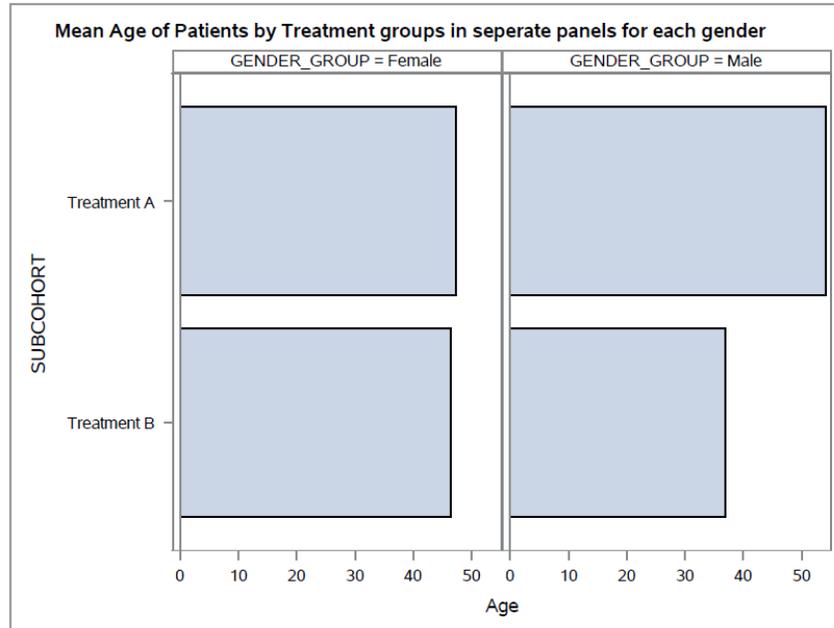


Figure 7: SG Panel graph with side-by-side comparative gender group panels showing mean age of patients among the treatment groups.

RACE AND ETHNICITY:

Like the pie chart, the donut chart can be used to investigate the composition of population. In this example, we will create donut chart by race and ethnicity. PROC GCHART is used to create the donut chart. Alternatively PROC SGPIE can also be used to create a customized donut chart in the newer SAS versions.

SAS code to create donut chart for race and ethnicity (Table 2. Figure 8):

Example 1:

```

/* Create Donut chart with PROC GCHART */

title "Composition of Race in the Cohort population";
goptions device=png xpixels=500 ypixels=400 ctitle=gray33 ctext=gray33 htext=12pt ftext='albany
amt/bold';
pattern1 v=solid c=lightbrown;
pattern2 v=solid c=lightblue;
proc gchart data=sample_data;
donut race/ type=freq noheading /*donut statement to create donut chart*/
angle=90 clockwise descending
slice=inside value=inside percent=inside
coutline=gray77;
run;

```

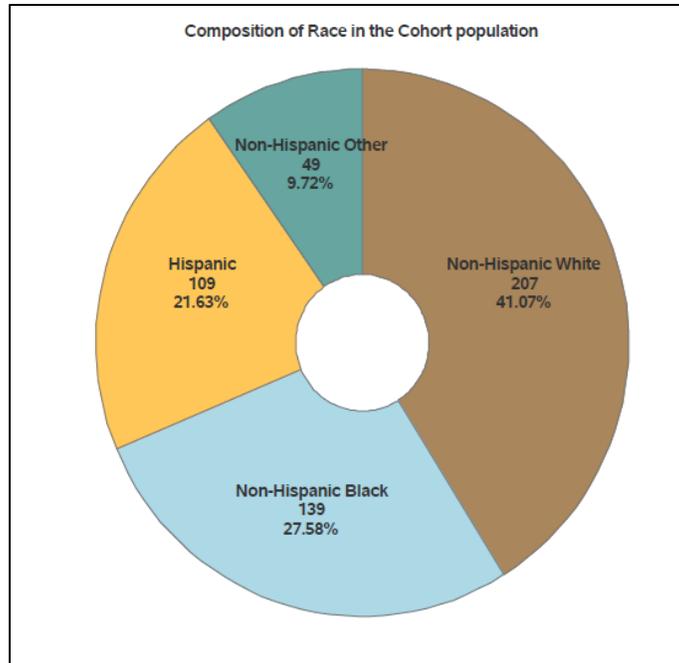


Figure 8: Donut chart with composition of race and ethnic groups among the cohort population.

TYPE OF INSURANCE:

The sample data from Table 2 is used to create a grouped cluster bar chart to compare multiple insurance categories by the treatments given using PROC SGPLOT.

SAS code to create grouped bar chart for insurance (Table 2, Figure 9):

Example 1:

/ Create numeric variable with patient count*/*

```
proc sql;
create table insurance_summary as
select insurance_plan_type, subcohort, count (distinct patient_id) as cnt from sample_data group by 1,2;
quit;
```

*/*Create Grouped Bar Chart using PROC SGPLOT*/*

```
title 'Frequency of patients by Insurance and Treatment';
proc sgplot data=insurance_summary noborder;
vbar insurance_plan_type / response=cnt group=subcohort groupdisplay=cluster
dataskin=gloss baselineattrs=(thickness=0);
xaxis label= 'Insurance Plan Type' display=(noline noticks);
yaxis label= 'Number of Patients' display=(noline) grid;
run;
```

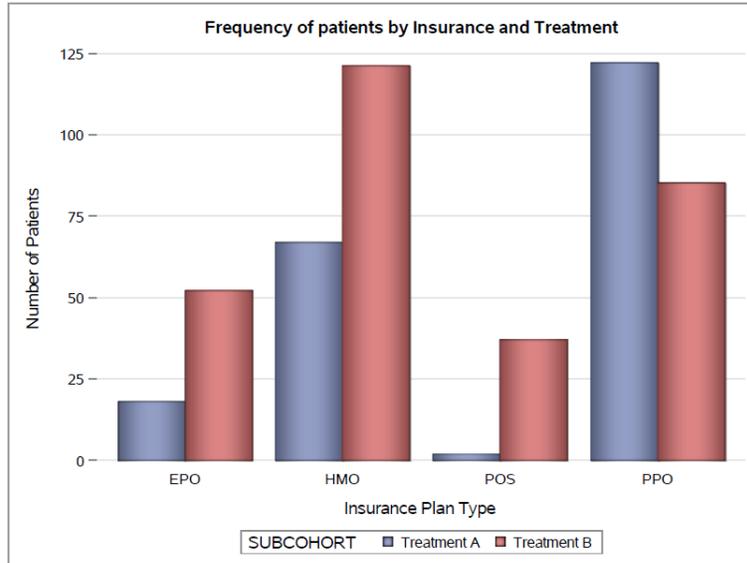


Figure 9: Grouped cluster bar chart showing patient count by insurance and treatment.

GEORGRAPHICAL LOCATION:

We will study the random sample selected patients based on the location by US states as well as by US regions. PROC GMAP uses the existing maps such as maps.US in the SAS library to create data visualizations as maps by countries, states etc. These existing maps can also be further customized, or a completely new map can be created to use in the PROC GMAP procedure to show visualizations by location as well as for other purposes. Annotate datasets such as dataset named 'center' (<https://support.sas.com/kb/25/558.html>) can be used to label the states in the US choro map.

SAS code to create the geographic location map by states (Table 2, Figure 10):

Example 1:

```

/* Get count of patients by state id */

proc sql;
create table demog as
select input(a.stateid,5.) as state, count(distinct patient_id) as cnt from sample_data a
group by 1,2;
quit;

/* Use PROC GMAP to display patients by states */

title 'Distribution of Cohort Patients by State';
proc gmap data = demog map=maps.us;
id state;
block cnt; /*Block map*/
run;
quit;
proc gmap data = demog_final map=maps.us;
id state;
choro cnt /annotate=center; /*Choro map with annotate center dataset to write labels */
run;
quit;

```

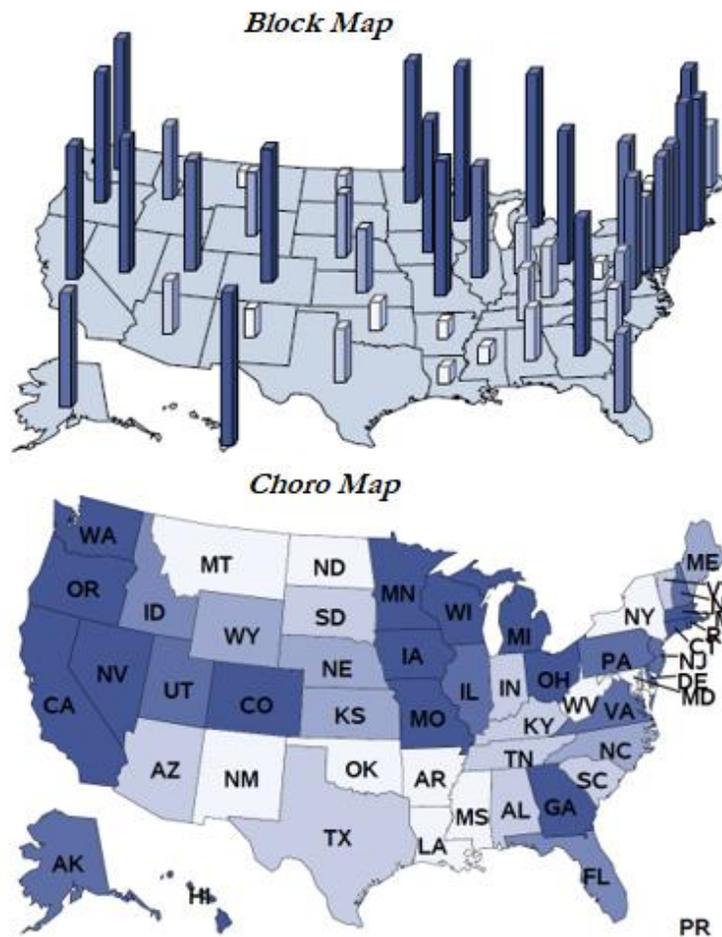


Figure 10: 3D map distribution of patients across different states in the US in the block map and 2D map distribution of patients in the choro map.

The standard maps.US dataset available in the SAS library is customized to create a region map as shown below.

SAS code to create the geographic location map by regions (Table 2, Figure 11):

Example 2:

```
/* Set the graphics environment */
```

```
goptions reset=all cback=white border htitle=12pt htext=10pt;
```

```
/* Create custom map data set STATES1 by combining states into regions from standard library maps.
US dataset */
```

```
data states1;
set maps.us;
select;
/* west */
when (state=53 or state=41 or state=16 or state=32 or state=06
```

```

or state=02 or state=15) region=1;
/* north central */
when (state=30 or state=46 or state=38 or state=31
or state=20 or state=56 or state=49
or state=08) region=2;
/* southwest */
when (state=04 or state=35 or state=48
or state=40) region=3;
/* midwest */
when (state=27 or state=19 or state=29 or state=55
or state=17 or state=26 or state=18
or state=39) region=4;
/* south */
when (state=05 or state=22 or state=01 or state=21 or state=47
or state=28 or state=13 or state=12 or state=45 or state=37
or state=51 or state=54) region=5;
/* new england */
otherwise region=6;
end;
run;

/* Sort by REGION STATE. STATE must be included in the sort to preserve the original order of the
points that comprise each state */

proc sort data=states1;
by region state;
run;

/* Remove the state boundaries since we only want one block per region */

proc gremove data=states1 out=map;
by region;
id state;
run;

data demog;
set sample_data;
select;
/* west */
when (stateid=53 or stateid=41 or stateid=16 or stateid=32 or stateid=06
or stateid=02 or stateid=15) region=1;
/* north central */
when (stateid=30 or stateid=46 or stateid=38 or stateid=31
or stateid=20 or stateid=56 or stateid=49
or stateid=08) region=2;
/* southwest */
when (stateid=04 or stateid=35 or stateid=48
or stateid=40) region=3;
/* midwest */
when (stateid=27 or stateid=19 or stateid=29 or stateid=55
or stateid=17 or stateid=26 or stateid=18
or stateid=39) region=4;
/* south */
when (stateid=05 or stateid=22 or stateid=01 or stateid=21 or stateid=47
or stateid=28 or stateid=13 or stateid=12 or stateid=45 or stateid=37
or stateid=51 or stateid=54) region=5;

```

```

/* new england */
otherwise region=6;
end;
run;

proc format;
value region
1="west"
2="north central"
3="southwest"
4="midwest"
5="south"
6="new england";
run;

/* Create patient frequency by region */

proc sql;
create table demog_final as
select region, count(distinct patient_id) as cnt from demog a
group by 1;
quit;

/* Use PROC GMAP with custom US map to show patient distribution */

title1 'Distribution of Cohort Patients by Region';
proc gmap data = demog_final map=map all;
note move=(20,88) font="albany amt/bold" height=4.5;
id region;
block cnt / description="Patient Count" ;
format region.;
run;
quit;

```

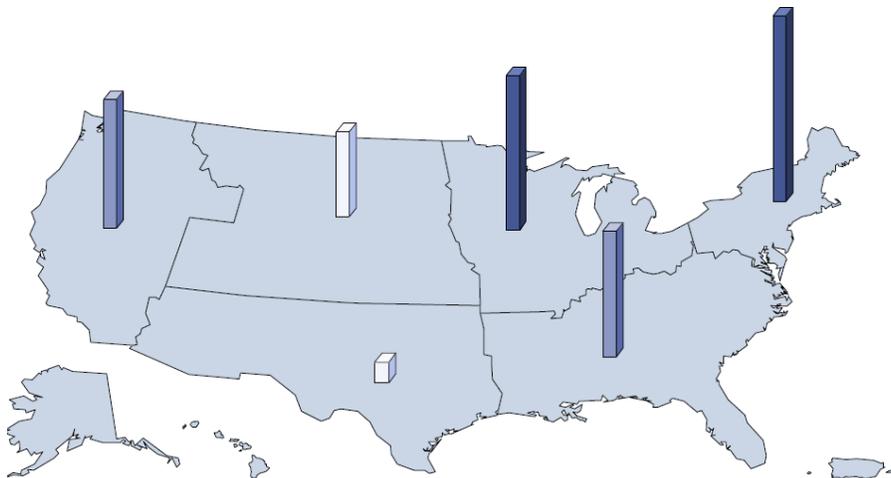


Figure 11: US region map chart showing patient's frequency distribution by region.

INCOME:

Demographic disparities are often studied in HEOR to understand if that demographic variable such as income influences the outcomes of interest. In this example, we will use PROC GMAP to study the income groups across different states in US using random sample data.

SAS code to create map chart for Income groups (Table 2, Figure 12):

Example 1:

```
/* Create average income per state from patient level income dataset*/
```

```
proc sql;  
create table demog as  
select input(a.stateid,5.) as state, avg(income) as income_total from sample_data a  
group by 1;  
quit;
```

```
/* Create proc format to list the Income groups in the map */
```

```
proc format;  
value income low-20000 = 'less than $20,000'  
20001-40000 = '$20,001-$40,000'  
40001-60000 = '$40,000-$60,000'  
60001-80000 = '$60,001-$80,000'  
80001-100000 = '$80,001-$100,000'  
100001-120000 = '$100,001-$120,000'  
120001-high = 'more than $120,000';  
run;
```

```
/* PROC GMAP to create income-based groups in the US map */
```

```
proc gmap data = demog map=maps.us;  
id state;  
choro income_total /discrete;  
format income_total income.;  
run;  
quit;
```

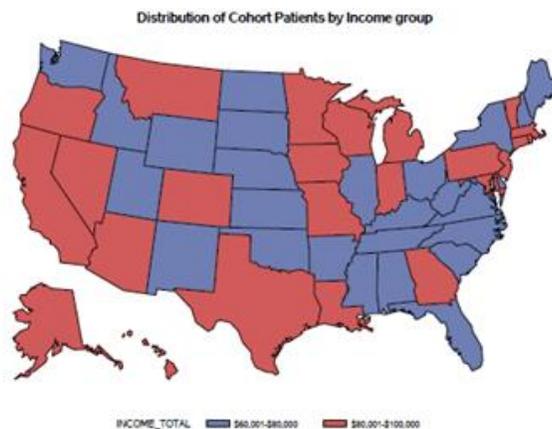


Figure 12: 2D income group map chart across the US states.

COMORBIDITIES:

CHARLSON COMORBIDITIES:

Many HEOR study analyses include understanding patient comorbidities. Comorbidity is defined as a patient's pre-existing medical condition of a patient, or the presence of one or more medical conditions known to affect the risk of mortality. Comorbid conditions are evaluated using the Charlson Comorbidity Index (CCI). The CCI is a method for measuring patient comorbidity based on the International Classification of Diseases (ICD) diagnoses codes of individual patients using database types such as claims or EHR. Each comorbidity category has an associated weight, based on the adjusted risk of one-year mortality, and the sum of all the weights results in a single comorbidity score for a patient. This score is sometimes used in decision making such as selecting candidates for surgery and so forth. In this section, we will look at sample random data and its visualization of patient numbers by these comorbidities in a comparative butterfly chart between male and female groups. A butterfly plot is an effective comparative bar chart between two groups to study the distribution of patients in variables such as comorbidities. In this example, we will compare the comorbidities of a patient with a diagnosis between the male and female groups. The sample data in Table 3 will be used to create a butterfly plot. This sample data is a summary level data with patient counts for each comorbidity and percentages of each comorbidity based on each total gender cohort population.

Baseline Charlson comorbidities	Diagnosis - X N=5400			
	Male N= 2340		Female N=3060	
	N	%		%
Myocardial infarction	30	0.09	43	43
Heart failure	44	0.3	16	16
Cerebrovascular disease	100	4.27	3	3
Peripheral vascular disease	70	0.38	60	60
Diabetes with complications	50	0.81	60	3
Diabetes		1.28		6

Table 3. Sample summary data for Charlson Comorbidities (N=Patient count, %=Percentage)

SAS code used to create butterfly chart for comorbidity comparison among gender (Table 3, Figure 13):

Example 1:

/ Create negative values for one of the groups, so we can display the chart as a butterfly. And create a variable zero for the reference point. - Also create a variable which has the total values of the two groups which will be used for sorting the dataset so we can display highest total value on the top. */*

```
data comorbidity;
set sample_data;
zero=0;
total=male+female;
male=-1*male;
run;
```

/ Create picture format for axes to be displayed with comma format. */*

```
proc format;
```

```

picture positive low-<0='000,000'
0<-high='000,000';
run;

/* Sort the dataset */

proc sort data=comorbidity;
by descending total;
run;

/* Create a butterfly plot with PROC SGPLOT with hbarparm statement with Y axis tick values.*/

ods graphics / reset width=7in height=5in imagename='Butterfly';
title 'Patient count by Comorbidity Group and Gender';
proc sgplot data=comorbidity noautolegend;
format male female positive.;
hbarparm category=comorbidity response=male / dataskin=gloss name='m'
fillattrs=graphdata1 /*datalabel=male datalabelattrs=(size=5) */ transparency=0 ;
hbarparm category=comorbidity response=female / dataskin=gloss name='f'
fillattrs=graphdata2 /*datalabel=female datalabelattrs=(size=5)*/ transparency=0;
scatter x=zero y=comorbidity / markerchar=comorbidity
markercharattrs=(size=5 weight=bold color=black); keylegend 'm' 'f';
xaxis values=(-150 to 150 by 10) display=(nolabel) grid offsetmin=0.05 offsetmax=0.05 ;
yaxis display=(noticks novalues nolabel);
run;

```

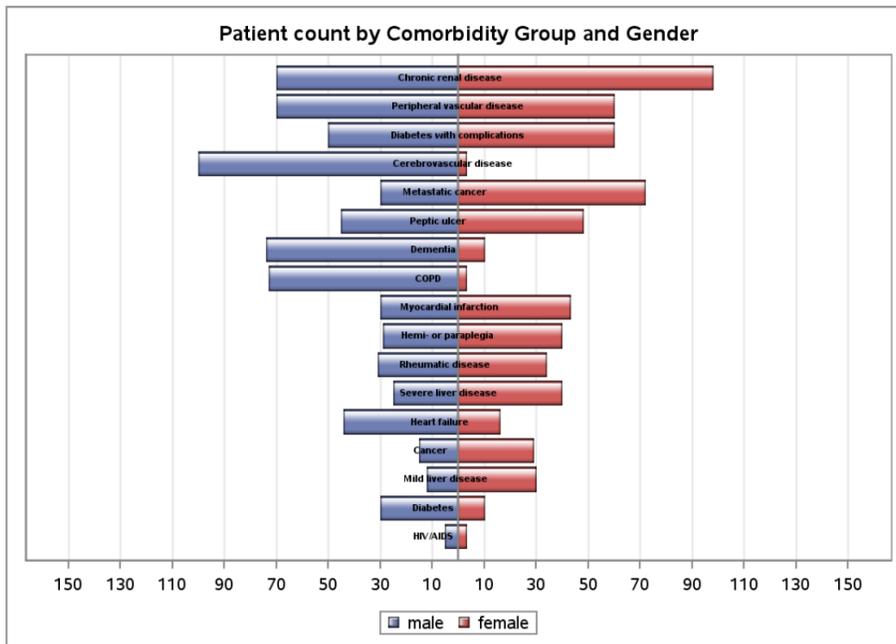


Figure 13: Patient count by comorbidity and gender in a butterfly comparative plot.

CCI score is assigned to each patient based on number of comorbidities associated with each patient. In the below example, sample patient level data in table 4 is used to create 3D Pie chart using PROC GCHART procedure. In this example we are not showing patients with more than 3 CCI score.

Patient_id	cci_score
1	0
2	1
3	2
4	2
5	2

Table 4: Patient level data with associated CCI score.

SAS code used to create the 3D pie chart with percentage of patients based on the CCI score (Table 4, Figure 14):

Example 2:

/ PROC GCHART to show composition of patients with different CCI score */*

```
proc gchart data=sample_data;
pie3d cci_score / sumvar=patient_id
coutline=black
explode=3
value=none
percent=arrow
slice=arrow
plabel=(font='Albany AMT/bold' h=1.3 color=depk);
run;
```

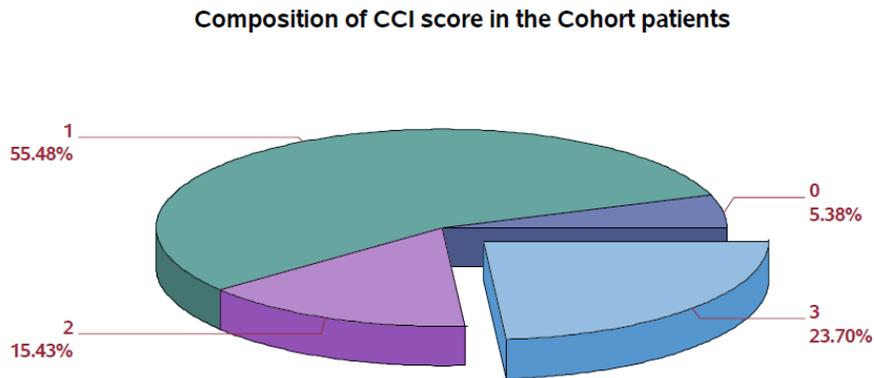


Figure 14: 3D sliced pie chart with the percentage of patients based on assigned CCI score 0,1,2 and 3.

TOP BASELINE DIAGNOSIS OR DIAGNOSIS RELATED GROUPS

In HEOR studies, sometimes top baseline comorbidities are studied to understand the pre-existing conditions that may affect the cohort population. Also, HEOR studies involve Diagnosis Related Groups (DRG) that are determined based on the characteristics of the patient population. These DRG's provide hospitals with information on costs incurred and expected services for the hospital stay or treatment services. In this example, we will look at how sample data in Table 5 shows how ranking changes among these comorbidities in the follow up years. In this sample data, ranking is assigned to the comorbidities

based on the selected sample of patients.

Top_diag	Rank	follow_up
ICD_A	1	1yr
ICD_B	3	1yr
ICD_C	4	1yr
ICD_D	6	1yr
ICD_E	2	1yr
ICD_F	7	1yr
ICD_G	10	1yr
ICD_H		1yr

Table 5: Sample summary data with top selected diagnosis based on ranking 1 through 10 in different follow up years.

SAS code used to create series line plot for comorbidity ranking over years (Table 5, Figure 15):

Example 3:

/ Create line plot showing change in ranking in the follow up years using PROC SGPLOT */*

```
proc sgplot data=sample_data noautolegend;
series x=follow_up y=rank / group=top_diag groupdisplay=cluster clusterwidth=0.1
name="scatter";
axis label="Follow up Year" ;
yaxis label="Top 10 Diagnosis" values=(0 to 12 by 1);
title "Change in Ranking of Top 10 Diagnosis in the Cohort patients";
keylegend / location=outside position=bottom title="Group";
run;
```

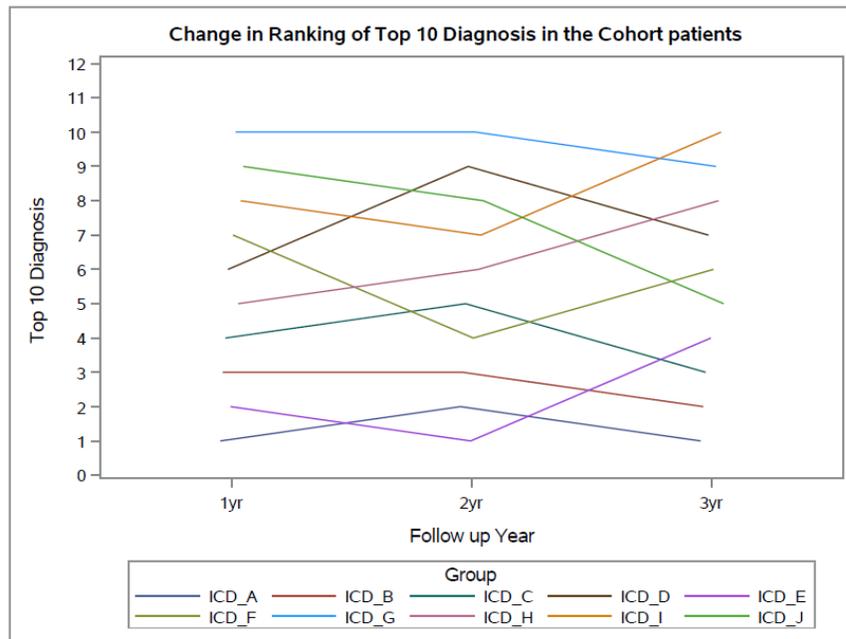


Figure 15: Series line plot showing top comorbid diagnosis ranking in follow up years.

COSTS AND UTILIZATION:

COSTS:

HEOR studies involve understanding the burden of the cost of the disease and its treatment by studying direct and indirect costs. This will help us understand the comparative cost-effectiveness and to build budget impact models. These costs can be medical, prescription (RX), inpatient (IP), outpatient (OP), emergency (ER), Intensive Care Unit (ICU) costs, direct costs, indirect costs, diagnosis-specific costs, treatment-specific costs and procedure-specific costs and so forth. In this example, sample data from Table 6 shows costs associated with each cost type and subgroup of the patient cohort population.

cost_type	costs	cohort_group
Inpatient	20000	A
Outpatient	7000	A
Pharmacy	4000	A
Inpatient	30000	B
Outpatient	5000	B
Pharmacy	6000	B
Inpatient	5000	C

Table 6: Sample summary data showing costs associated with cost type and subgroups.

SAS code used to stacked bar chart showing total costs (IP + OP + RX) (Table 6, Figure 16):

```
/* Picture format with comma and dollar sign */
```

```
proc format;  
picture costfmt low-high='000,009.99'  
(prefix='$');  
run;
```

```
/* Add pharmacy, inpatient and outpatient's costs to get total cost */
```

```
proc sql;  
create table totalcost as  
select a.cohort_group, sum(costs) as totalcost  
from sample_data a group by 1;  
quit;
```

```
/* Explicitly specify top segment based on the example. Here total cost value is displayed only the above  
the top segment pharmacy*/
```

```
proc sql;  
create table addtotal as  
select a.*, case when cost_type = "Pharmacy" then b.totalcost else . end as totalcost from sample_data  
a left join totalcost b on a.cohort_group = b.cohort_group  
order by cohort_group, cost_type;  
quit;
```

```
/* Create low and high values for each segment to stack above one another and display values in the middle of each associated segment */
```

```
data highlow;
set addtotal;
by cohort_group cost_type;
if first.cohort_group then _ybase=0;
yMin = _ybase;
yMax = actual_sum+_ybase;
ymid = (actual_sum/2)+_ybase;
output;
_ybase + actual_sum;
format actual_sum costfmt.;
format totalcost costfmt.
run;
```

```
/* Create labeled stacked bar chart using PROC SGPLOT */
```

```
proc sgplot data=highlow;
title 'Costs by cohort group and type in 1 year follow up';
highlow x=cohort_group low=ymin high=ymax / group=cost_type
type=bar lineattrs=(pattern=solid) highlabel=totalcost
labelattrs=(weight=bold);
scatter x=cohort_group y=ymid / markerchar=actual_sum;
xaxis label="Patient Sub cohort Groups";
yaxis grid offsetmin=0 label="Costs in 1 year follow up";
run;
```

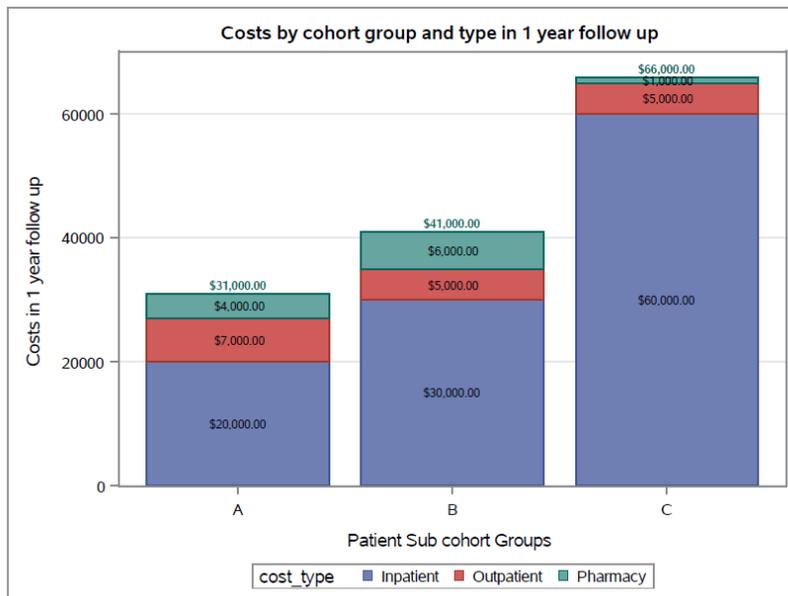


Figure 16: Stacked bar chart showing total costs of sample patients in 1 year follow up.

VISITS:

Like the costs, visits are analyzed in the utilization analysis of HEOR studies. In the sample data Table 7, Inpatient visits are counted in the data for different years. In example 1 and example 2, we study the

distribution of inpatient visits in the cohort population by using a histogram.

Year	Inpatient_Visits
2010	5000
2011	4500
2012	4000
2013	3200
2014	2000

Table 7: Sample summary table of inpatient visits by year of study.

SAS code to create histogram (Table 7, Figure 17):

Example 1:

/ Create histogram of inpatient visits using PROC SGPLOT */*

```

title "histogram of inpatient visits across all years";
proc sgplot data=sample_data;
  histogram inpatient_visits;
run;

```

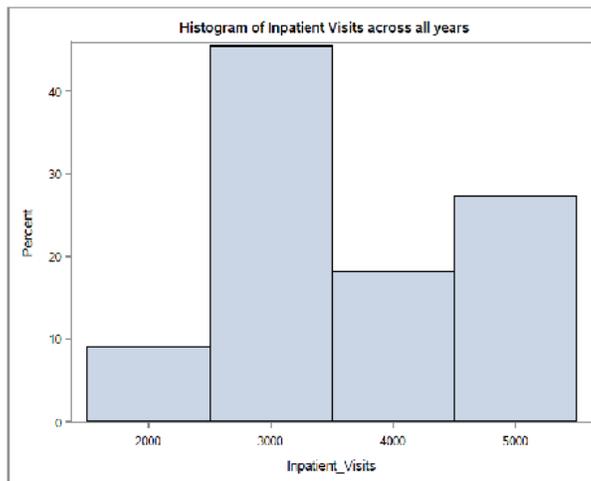


Figure 17: Histogram chart showing inpatient visit distribution in the cohort sample patients.

Patient_id	year	Visits
1	2010	200
2	2011	148
3	2012	129
4	2013	101
5	2014	140
6	2015	200

Table 8: Sample dataset for outpatient visits.

SAS code to create histogram (Table 8, Figure 18):

Example 2:

```

/* PROC UNIVARIATE to create plots */

title 'summary of outpatient visits by year';
proc univariate data = sample_data2 noprint;
class year;
histogram visits / normal (color = red)
cfill = ltgray
cext = blue;
inset n = 'number of outpatient visits' median (8.2) mean (8.2) std = 'standard deviation' (8.3)
/ position = ne; /*Inset option to show the box in the graph with statistical analysis*/
label year = 'year';
run;

```

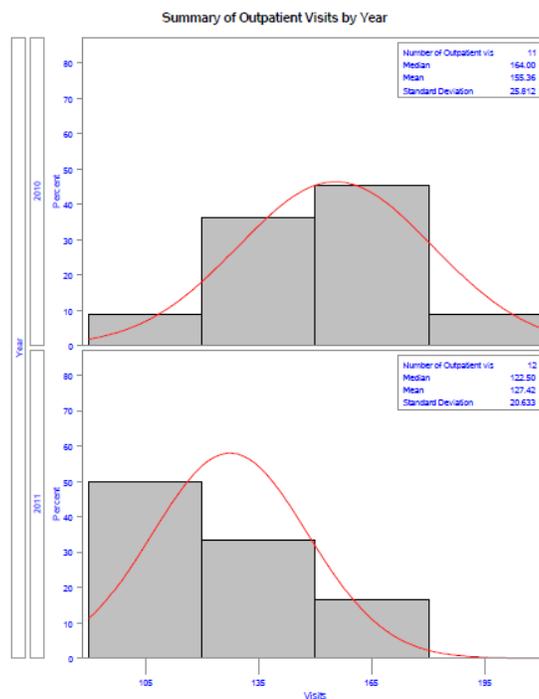


Figure 18: Histogram of outpatient visits by year showing descriptive analysis listed in the graph.

In the example 3, we will create custom map for the sample summary level dataset in Table 9 showing the inpatient visits by speciality. This custom map will be used in PROC GMAP to create a tree map chart.

Speciality	visits	pct_total	id_var
Emergency medicine	72	24	1
Family medicine	42	14	2
Internal medicine	18	6	3
Obstetrics and gynecology	48	16	4
Neurology	120	40	5

Table 9: Sample summary level dataset showing visits by speciality.

SAS code to create histogram (Table 9, Figure 19):

Example 3:

```
/* Create ID variable to match with ID variable of the custom map */
```

```
data sample_data;  
set sample_Data;  
id=id_var;  
run;
```

```
/* Create custom X and Y coordinates with id variable. These X and Y coordinates should represent the  
size and shape of the box representing the specialty in the tree map. It is important that the relative area  
of the specialty in the tree map represents the percentage (pct_total) in the sample dataset*/
```

```
data custommap;  
input id x y;  
datalines;  
1 4 0  
1 4 4  
1 10 4  
1 10 0
```

```
2 0 4  
2 0 6  
2 7 6  
2 7 4
```

```
3 7 4  
3 7 6  
3 10 6  
3 10 4
```

```
4 0 0  
4 0 4  
4 4 4  
4 4 0
```

```
5 0 6  
5 0 10  
5 10 10  
5 10 6;  
run;
```

```
proc format;  
value speciality  
1="Emergency medicine"  
2="Family medicine"  
3="Internal medicine"  
4="Obstetrics and gynecology"  
5="Neurology";  
run;
```

```
/* Create annotate dataset with labels and their corresponding x and y positions in the map so that labels  
are in the center of the shapes */
```

```

data labels;
length text $30;
retain xsys ysys '2' hsys '3' position '5' function 'label' size 2 style "'Tahoma/bo'" when 'a';
input text x y;
datalines;
Emergency_medicine 6 2
Family_medicine 3 5
Internal_medicine 9 5
Obstetrics_and_gynecology 2 1
Neurology 3 8;
run;

/* Use PROC GMAP to create the tree map */

proc gmap map=custommap
data=sample_Data;
format id speciality.;
id;
choro visits / discrete annotate=labels;
run;

```

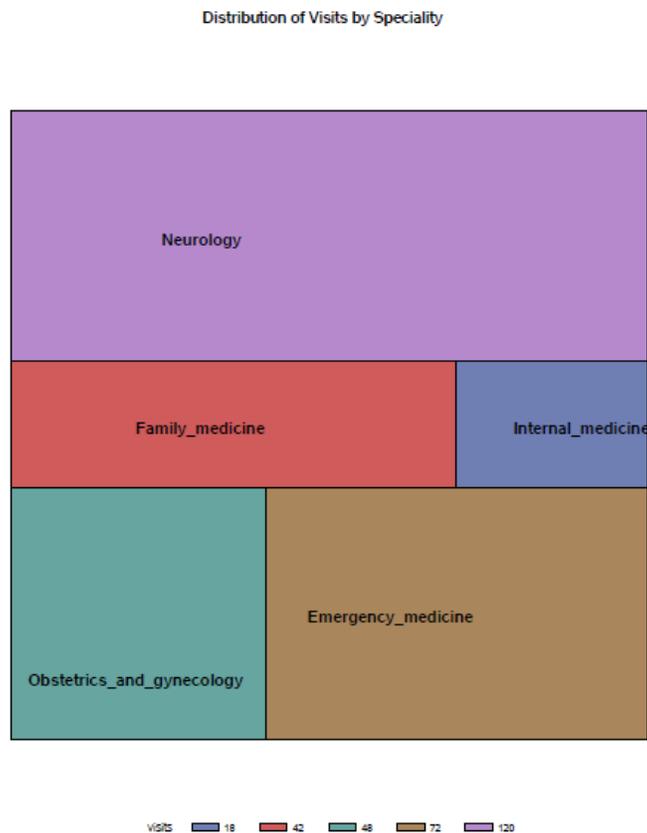


Figure 19: Tree map showing patient visits counts by specialty.

In the example 4 below, we will create a vector plot comparing the inpatient and outpatient visits across years using PROC SGPLOT. Vector plot can be customized for other purposes such as studying preferences among population. Like some other plots, vector plot can be overlaid with other plots.

Example 4:

Year	Inpatient_visits	Outpatient_visits
2010	500	3500
2011	600	3000
2012	700	2500
2013	1000	2000
2014	3000	1500
2015	3200	1200

Table 10: Sample summary dataset with inpatient and outpatient visits.

Sample SAS code to create vector plot (Table 10, Figure 20):

```
/* Create vector plot using PROC SGPLOT */
```

```
proc sgplot data=sample_data aspect=1 noborder nowall noautolegend subpixel;  
vector x=outpatient_visits y=inpatient_visits / group= year datalabel=year ;  
xaxis label="Outpatient visits";  
yaxis label="Inpatient visits";  
run;
```

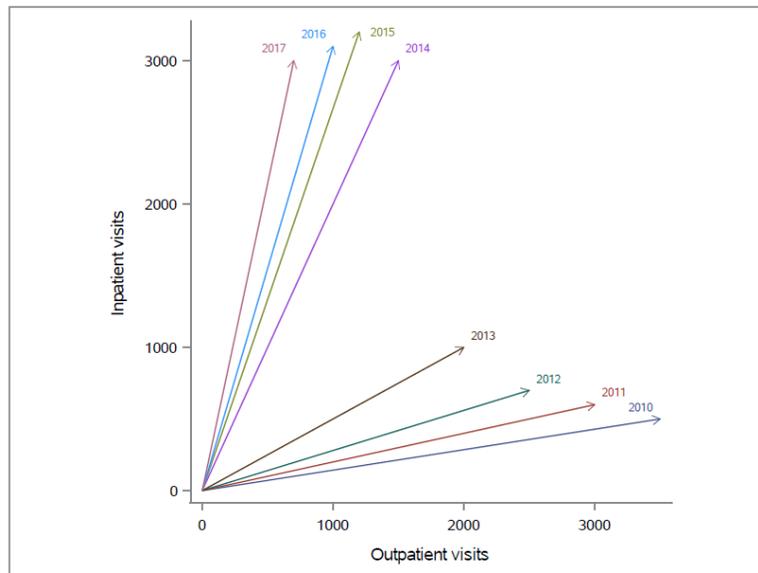


Figure 20: Vector plot comparing inpatient and outpatient visits across years of study.

HOSPITALIZATIONS:

Hospitalizations are counted in some studies as a part of the cost and utilization analysis. In the example 1, under hospitalizations, sample data in the Table 11 is created as a chart showing trends across months and years of patient hospitalizations.

No_of_hospitalizations	Year	Month
100	2019	Jan
150	2020	Jan
110	2019	Feb
200	2020	Feb
120	2019	Mar

Table 11: Summary sample data showing number of hospitalizations across months and years.

SAS code to create calendar chart for hospitalizations (Table 11, Figure 21):

Example 1:

/ PROC GRADAR chart showing number of hospitalizations */*

```
proc gradar data=sample_data;
chart month / freq=no_of_hospitalizations
calendar
overlayvar=year;
run;
quit;
```

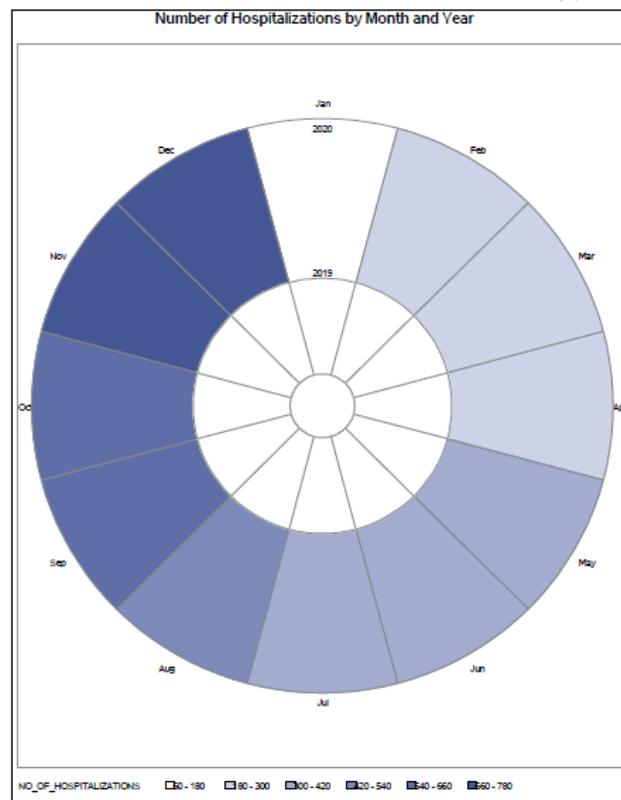


Figure 21: Calendar chart of number of hospitalizations across months and years.

In the example 2, we will create scatter plot to study the relation of three variables using PROC SGSCATTER and PROC SGPLOT. Sample data is shown in Table 12.

patient_id	age	hospitalization	preexisting_cond
1	18	200	yes
2	19	300	yes
3	20	350	no
4	35	400	yes
5	40	500	yes
6	45	550	yes
7	50	600	yes

Table 12: Patient level sample data for creating scatter plot.

SAS code to create scatter plot (Table 12, Figure 22 ad Figure 23):

Example 2:

```
/* Create scatter plot using PROC SGSCATTER */
```

```
title "difference in age vs hospitalizations by the presence on one or more preexisiting conditions";
proc sgscatter data=sample_data;
plot hospitalization * age
/ datalabel = preexisting_cond group = preexisting_cond;
run;
```

```
/* Create scatter plot using PROC SGPLOT */
```

```
proc sgplot data=sample_data;
scatter x=age y=hospitalization / group=preexisting_cond;
xaxis label = "age of the inpatient";
yaxis label = "number of hospitalizations";
run;
```

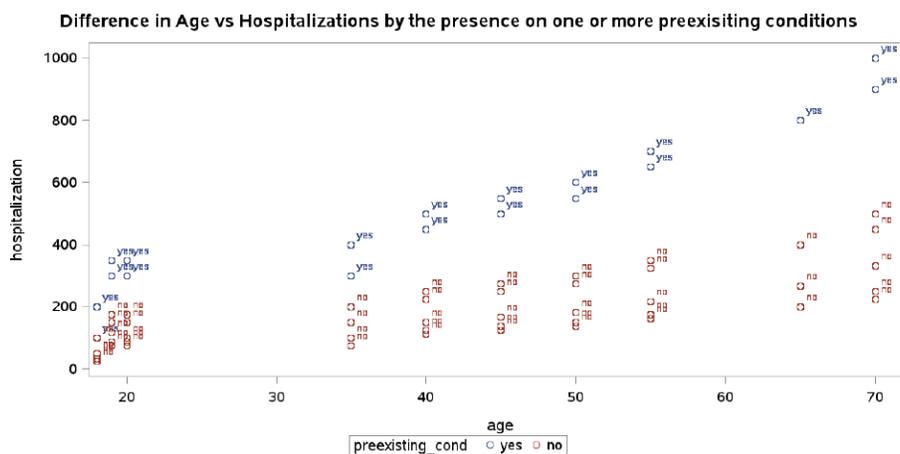


Figure 22: scatter plot using PROC SGSCATTER.

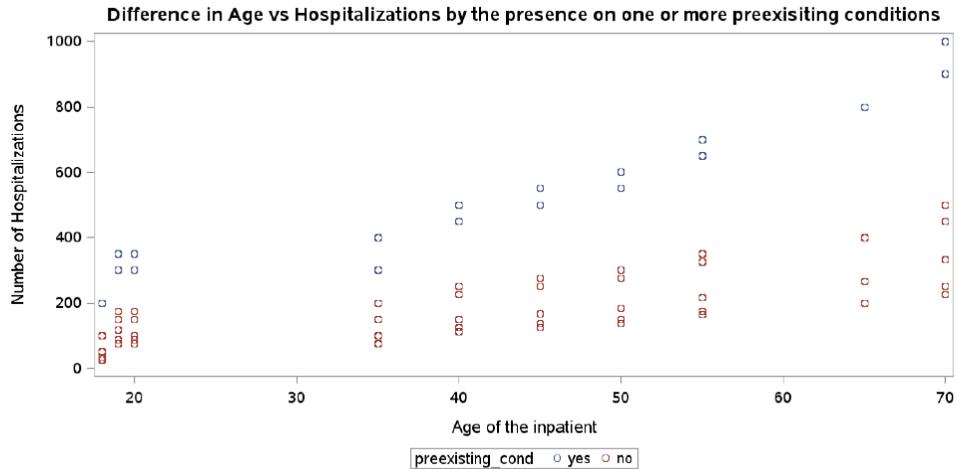


Figure 23: scatter plot using PROC SGPLOT.

READMISSION:

Readmission counts and readmission rates are studied in HEOR to understand the causes of increase in readmission in some studies. Readmission definitions can vary based on the study analysis. In this example, we will create spaghetti style plot by using series plot in PROC SGPLOT procedure to understand readmission rates based on the initial encounters i.e., initial hospitalizations and readmission hospitalizations. This example will study difference in readmission rates across several DRG's.

DRG	Encounter_type	Number
Aplasia	Initial encouners	120
Anemia	Initial encouners	100
COPD	Initial encouners	200
Diabetis	Initial encouners	250
CHD	Initial encouners	250
Asthma	Initial encouners	70
Aplasia	Readmi	50

Table 13: Sample summary data showing patients hospitalizations by Diagnosis Related Groups (DRG's)

Sample SAS code for creating spaghetti style plot (Table 13, Figure 24):

Example 1:

/ First calculate Readmission rates based on initial encounters and readmission hospitalizations. */*

```
proc sql;
create table in_enc as
select a.drg, number as initial_visits from import a where lower(encounter_type) = 'initial encouners';
quit;
```

```
proc sql;
create table readm as
select a.drg, number as readm_visits from import a where lower(encounter_type) = 'readmission';
```

```
quit;
```

```
proc sql;  
create table readm_rate as  
select a.drg, initial_visits, readm_visits, (readm_visits/initial_visits)*100 as readmrate  
from in_enc a left join readm b on a.drg=b.drg;  
quit;
```

```
proc sql;  
create table readmission as  
select a.*, number as visits, round(b.readmrate,1) as readr, trim(a.drg||put(calculated readr,10.)||"%") as  
label  
from import a left join readm_rate b on a.drg=b.drg;  
quit;
```

```
/*Create Spaghetti style line plot showing number of encounters and rate as a label for readmission –  
using PROC SGPLOT*/
```

```
title "Readmission Rate by Diagnosis Related Group (DRG)";  
proc sgplot data=readmission noautolegend;  
series x=encounter_type y=visits/ group= label curvelabel curvelabelattrs=(color=red)  
lineattrs=(thickness=4) curvelabelloc=inside curvelabelpos=start ;  
xaxis label="Encounter" ;  
yaxis label="Number of Visits" ;  
run;
```

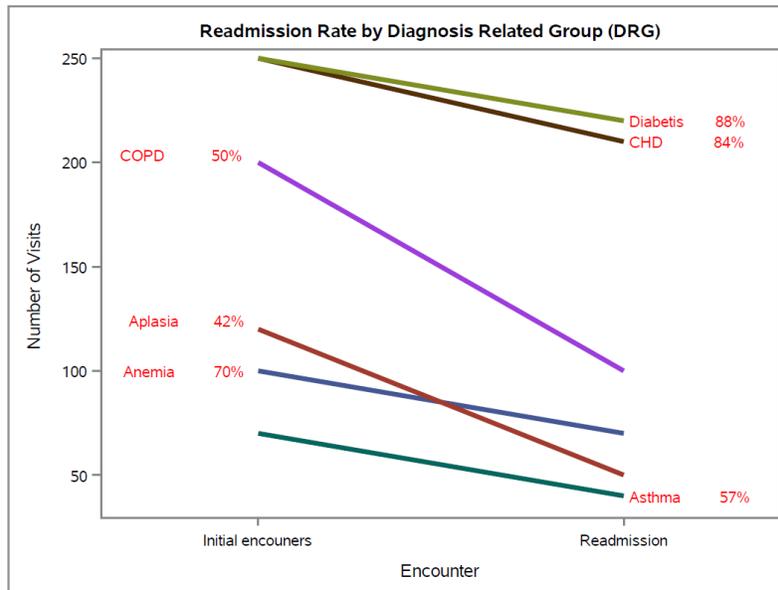


Figure 24: Spaghetti line plot showing number of inpatient stays and readmission rate across different Diagnosis Related Group (DRG).

LENGTH OF STAY:

Length of stay is studied many times for inpatient stay and hospitalization studies to understand the burden of disease and the cost due to specific diagnosis and so forth. In this example the length of stay from the sample data from Table 14 is shown as a bubble chart using PROC SGPLOT.

Patient_ID	LOS
1	5
2	2
3	7
4	6
5	2
6	2

Table 14: Patient-level data with the Length of Stay (LOS) variable.

Sample SAS code for creating bubble chart (Table 14, Figure 25):

Example 1:

```
/* Get count of patients by length of stay */
```

```
proc sql;
create table los_size as
select a.los, count(distinct patient_id) as patientcnt from los a
group by 1;
quit;
```

```
/* Create bubble plot with proportional bubble size depending on patient count using PROC SGPLOT */
```

```
title1 'length of hospital stay in the cohort patients';
proc sgplot data=los_size aspect=0.6 noborder;
bubble x=los y=patientcnt size=patientcnt/ proportional datalabel=patientcnt group = patientcnt
datalabelpos=center
transparency=0.4 datalabelattrs=(size=9 weight=bold) dataskin=sheen;
inset "bubble size represents patient count" / position=bottomright textattrs=(size=11);
yaxis grid label="number of inpatients";
xaxis grid label="length of stay";
run;
```

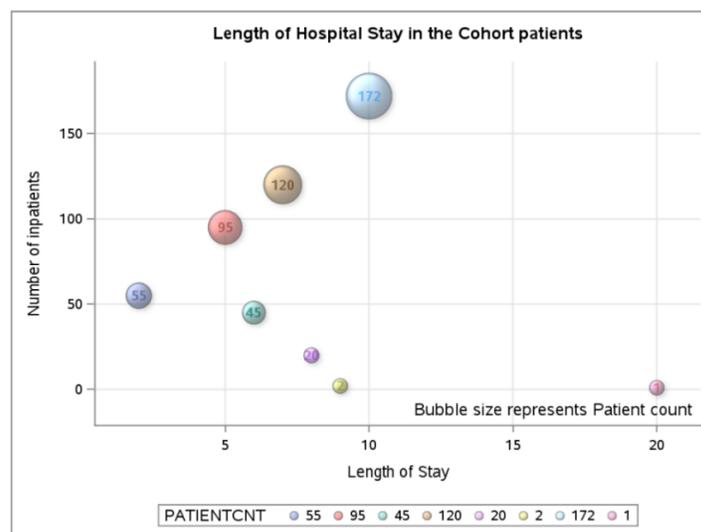


Figure 25: Bubble chart showing bubble size based on the number of patients with different lengths of stay.

LABORATORY DATA ANALYSIS:

Laboratory analysis involve studying patient's lab data which may include vitals such as Body Mass Index (BMI), Blood Pressure (BP). It may also include lab data related to diagnosis such as glucose, hemoglobin etc. Below examples analyze lab data by data visualization.

treatment_group	age_group	Glucose
A	18-25	100
B	26-35	100
C	35-45	110
D	45-50	90
	60+	

Table 15: Summary level sample data with glucose levels by treatment group and age group.

The sample data in Table 15 will be analyzed in the block plot. Block plot shows relation of their variables, patient subgroup based on the treatment, age group and glucose levels.

Sample SAS code for creating block chart (Table 15, Figure 26):

EXAMPLE 1:

```
/* Sort data by age group */
```

```
proc sort data=sample_data;  
by age_group;  
quit;
```

```
/* Create block map using PROC SGPLOT */
```

```
title "glucose levels in response to the different treatments";  
proc sgplot data=sample_data;  
block x=treatment_group block=age_group;  
series x=treatment_group y=glucose / markers;  
xaxis label="treatment group";  
yaxis label="glucose levels mg/dl";  
run;
```

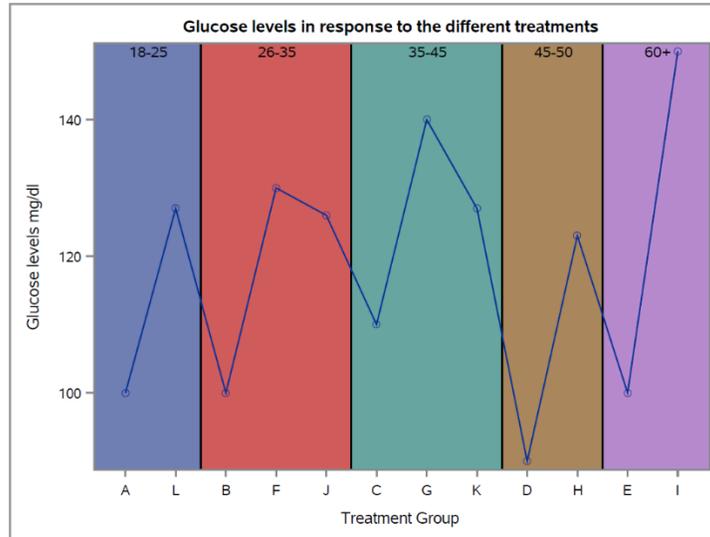


Figure 26: Block chart showing with each block showing patient number in each age group and treatment group with overlay of line plot showing glucose levels.

treatment_group	systolic	diastolic
A	126	80
B	120	80
C	130	90
D	120	95

Table 16: Summary level sample data with BP recordings by the treatment group.

The sample data in Table 16 will be analyzed in the High and Low plot. High and Low plot will enable us to see the two high and low readings in this case by the treatment group.

Sample SAS code for creating high low chart (Table 16, Figure 27):

EXAMPLE 2:

```
/* Sort data by age group */
```

```
proc sort data=sample_data;  
by treatment_group;  
quit;
```

```
/* Create high and low plot using PROC SGPLOT */
```

```
proc sgplot data=sample_data;  
highlow x=treatment_group high=systolic low=diastolic / type=bar  
groupdisplay=cluster highlabel=systolic lowlabel=diastolic lineattrs=graphoutlines  
dataskin=matte barwidth=0.3;  
xaxis label="Treatment Group";  
yaxis offsetmin=0 values=(50 to 150 by 20) label="Systolic and Diastolic readings";  
run;
```

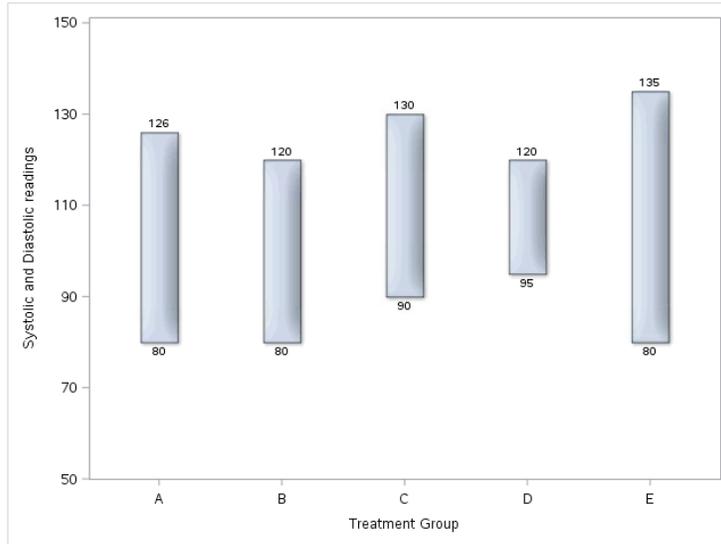


Figure 27: High and Low chart showing with each bar showing readings (systolic -high, diastolic-low) in each treatment group.

BMI	Pain_1	Pain_2	Pain_3	Pain_4	Pain_5	Pain_6	Pain_7	Pain_8	Pain_9	Pain_10
<18.5 (underweight)	25	35	40	50	53	56	57	60	70	67
18.5-<25 (normal weight)	30	37	44	55	57	58	60	65	75	80

Table 17: Patient level data with pain scale by BMI and gender groups.

The sample data in Table 17 will be analyzed in the heatmap. Below heat map shows intensity of colors representing patient count. Below chart is designed to analyze relation of two variables i.e. BMI and pain scale.

Sample SAS code for creating heat map (Table 17, Figure 28):

EXAMPLE 3:

/ Transform each pain variable to one pain scale variable */*

```

data sample_data2;
set sample_data1;
array pains[1:10] pain_1-pain_10;
do i = 1 to 10;
pain_scale=put(i, 8.);
count = pains(i);
output;
end;
drop pain_1 - pain_10 ;
run;

```

```

data sample_data2;

```

```

set sample_data2;
col=pain_scale;
row=bmi;
run;

/* Get counts of patients by pain scale-col and BMI -row*/

proc freq data=sample_data2; /* row and column marginals */
weight count;
tables row / out=t2 (drop=percent rename=(count=rowf row=mrow));
tables col / out=t3 (drop=percent rename=(count=colf col=mcol));
run;

/* Get maximum value of pain scale into macro variable to use in next step */
proc sql;
select max(colf) into: maxn
from t3 ;
quit;

/* Merge tables and create additional rows with minimum and maximum col – pain scale count*/
data tables;
if 0 then merge sample_data2 t2 t3;
low = 0;
x0 = 20;
if _n_ = 1 then do;
row = ' '; col = ' '; mrow = ' '; mcol = ' ';
count = 0; rowf = 0; colf = 0; output;
count = &maxn; rowf = &maxn; colf = &maxn; output;
end;
merge sample_data2 t2 t3;
output;
run;

/* Create a heat map showing correlation of two variables using PROC SGPLOT*/
title "the frequency of cohort patients by rheumatoid arthritis pain scale and bmi";

ods graphics on / width=5.8in height=5.0in;
proc sgplot data=tables noautolegend noborder;
title 'patient population by bmi and pain scale';
%let c = colormodel=(white cx6567bb cxbb68bb cxdd5522);
heatmapparm y=row x=col colorresponse=count / &c;
text y=row x=col text=count;
%let o = type=bar barwidth=1 nooutline &c colorresponse;
highlow y=mrow low=low high=rowf / x2axis &o=rowf;
text y=mrow x=x0 text=rowf / x2axis;
highlow x=mcol low=low high=colf / y2axis &o=colf;
text x=mcol y=x0 text=colf / y2axis;
xaxis label = "pain scale" display=( noticks noline) offsetmax=.32;
yaxis label = "bmi" display=( noticks noline) offsetmin=.32 reverse;
x2axis display=none offsetmin=.72 offsetmax=.03;
y2axis display=none offsetmin=.78 offsetmax=.12;
run;
ods pdf close;

```

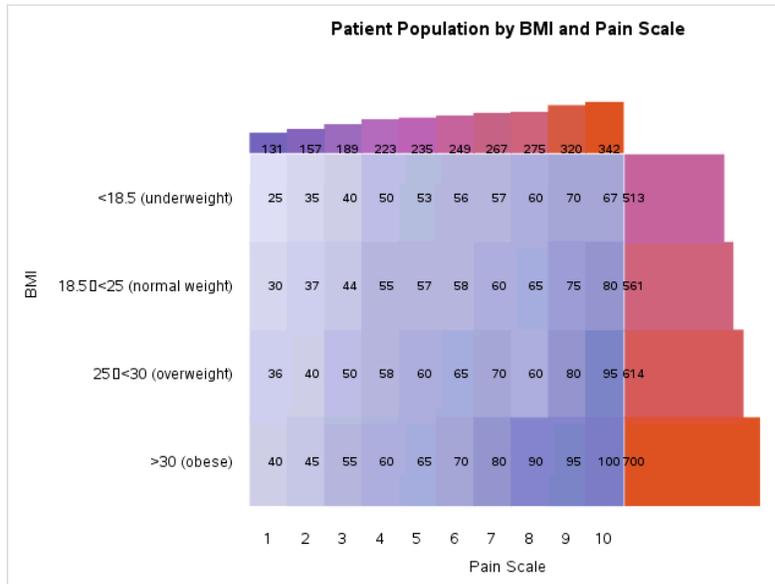


Figure 28: Heat map showing intensity of color by patient count in groups by pain scale and BMI. The graph also shows row and column totals.

TREATMENT:

Real world evidence or Post-market analysis on drugs and treatments in HEOR studies help us understand safety, preferences, associated cost, treatment patterns etc. Treatment patterns help us understand how different patients prefer different medications at different points of time. It helps us understand how different treatments and their patterns impact on patient's survival rates, costs associated and so forth.

Treatment	days_around_diag
A	-20
B	30
C	50
D	-10
E	20
F	40
G	70
H	90

Table 18: Summary level data by treatment group and days it started with respect to diagnosis date.

In some studies, patient's treatment start date is checked relative to the diagnosis start date. In this example, we will analyze which treatment groups started before the diagnosis and which ones after and how close to the diagnosis date.

SAS code to create bar chart with negative and positive y axis to show relative treatment start days with respect to index diagnosis (Table 18, Figure 29) :

EXAMPLE 1:

/ Create SAS bar chart with numeric variable using PROC SGPLOT */*

```
proc sgplot data=sample_data;
vbar treatment / response=days_around_diag; /* axis shows percentages instead of counts */
```

```
axis label="treatment type";
yaxis label="days between earliest treatment date to diagnosis date of cohort patients";
run;
```

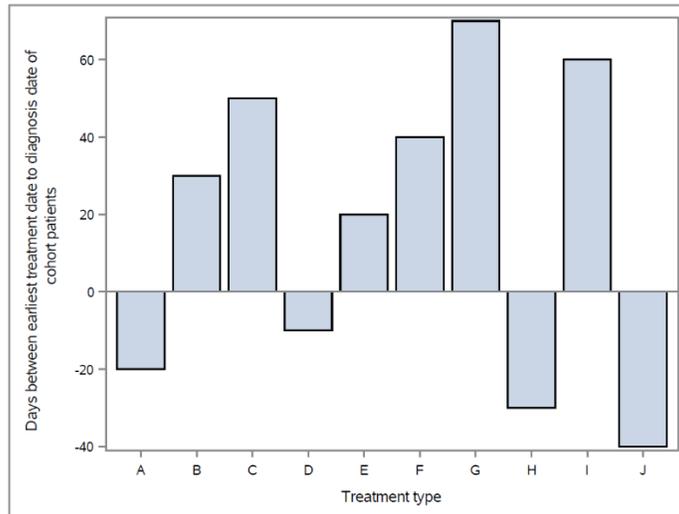


Figure 29: Bar chart showing days before and days after the diagnosis start date depending on the treatment type.

Month	Drug_pattern	survival_pct
0	continue	85
1	continue	80
2	continue	80
3	continue	79
4	continue	83
5	switch	83

Table 19: Summary level dataset showing survival percentage of patients with different treatment patterns over several months.

In example 2, we would visualize by series plot regarding patient's survival percentage if patient continued initial treatment, if patient switched to another type and if patient discontinued treatment by using PROC SGPLOT.

SAS code to create series line plot (Table 19, Figure 30):

EXAMPLE 2:

```
/* Create SAS bar chart with numeric variable using PROC SGPLOT */
```

```
proc sgplot data = sample_data2 noautolegend ;
series x=month y=continue / lineattrs=(color=orange)
curvelabel='continue treatment' curvelabelpos=max curvelabelloc=outside;
series x=month y=switch / lineattrs=(color=green)
```

```

curvelabel='switch treatment' curvelabelpos=max curvelabelloc=outside;
series x=month y=discontinue / lineattrs=(color=blue)
curvelabel='discontinue treatment' curvelabelpos=max curvelabelloc=outside;
xaxis label="follow up month" grid values = (0 to 12 by 1);
yaxis label="% survival" grid values = (0 to 100 by 10);
title "study results by treatment pattern group";
run;

```

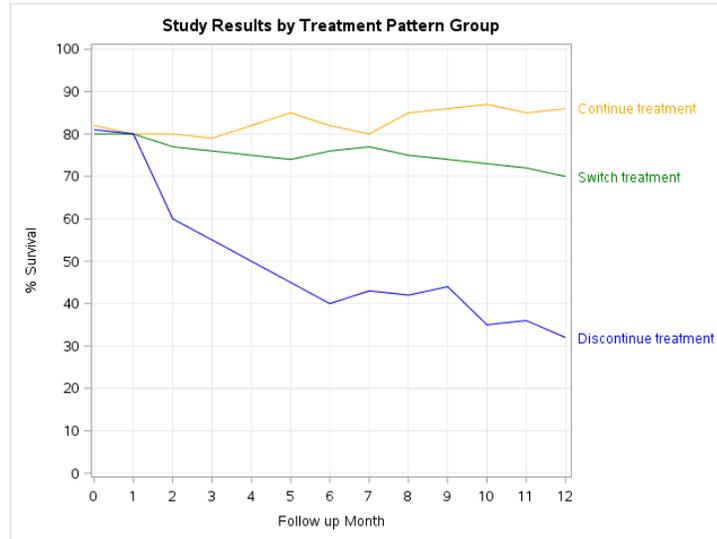


Figure 30: Series line plot comparing different treatment patterns.

RX_number	Days_of_supply
1	5
2	7
3	5
4	5
5	5

Table 20: Days of supply for the prescription sequence of medication in follow up period.

In example 3, we will see the accumulated summing pattern of days of supply over the follow up period. Waterfall chart helps us to understand how a total sum is calculated after sequence of additions and subtractions. This can also be applied in different other circumstances in several studies.

SAS code to create waterfall plot (Table 20, Figure 31):

EXAMPLE 3:

```
/* Create using PROC SGPLOT */
```

```

title "patient's total accumulated days of supply at the end of follow up period";
proc sgplot data=sample_data;
waterfall category=rx_number response=days_of_supply

```

```

/stat=sum;
xaxis label="prescription sequence in follow up period";
yaxis label="days of supply";
run;

```

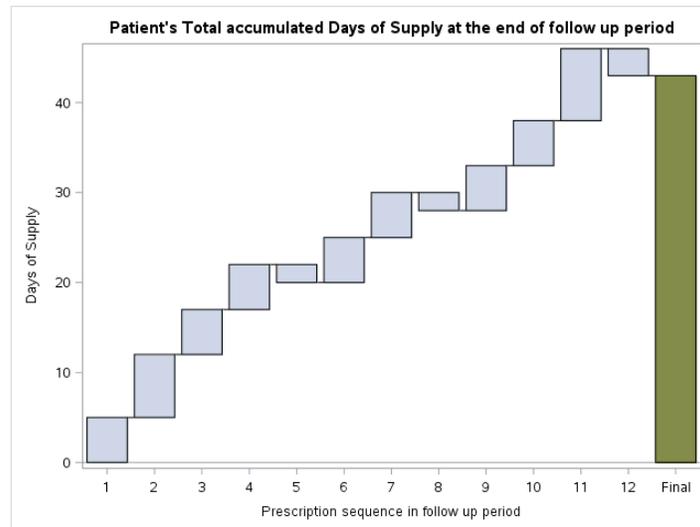


Figure 31: Waterfall chart showing cumulative sum of variable days of supply over 12 months.

Patient_id	Drug_1	Drug_2
1	A	A
2	A	B
3	A	C
4	B	B
5	B	B
6	B	B

Table 21: Patient level dataset with initial drug type and drug type after.

In example 4, we will study treatment patterns based on treatment switches and relative thickness of the series plot representing the number of patients switching in each type of pattern. This plot creates basic Sankey style plot. This method can be updated further to create advanced Sankey plot.

SAS code to create waterfall plot (Table 21, Figure 32):

EXAMPLE 4:

```

/* Identify switch pattern*/

data sample_data2;
set sample_data;
pattern=drug_1||drug_2;
run;

/* Create separate record for each level of switch - drug type */

```

```

proc sql;
create table step1 as
select patient_id, 'initial' as switch_no, drug_1 as drug_type, pattern
from sample_data2
union
select patient_id, 'first switch' as switch_no, drug_2 as drug_type, pattern
from sample_data2
;
quit;

/* Calculate totals for the percentages of each drug type within each switch */
proc sql;
select count(distinct patient_id) into: total
from step1;
quit;

/* Calculate patient counts by switch number and drug type */
proc sql;
create table drugtot as
select switch_no, drug_type , count(distinct patient_id) as drugtypetotal
from step1
group by 1,2
order by 1,2;
quit;

proc sql;
create table drugpct as
select a.* , (drugtypetotal/&total)*100 as pct1 from drugtot a ;
quit;

/*Calculate percentages of each switch pattern within each drug type -if adding more switches, calculate
percentages separately*/
/*this will be the thickness of series smooth connect between patterns*/
proc sql;
create table patterntot as
select switch_no, drug_type ,pattern, count(distinct patient_id) as patterntotal
from step1
group by 1,2,3
order by 1,2,3;
quit;

proc sql;
create table patternpct as
select a.* , b.drugtypetotal, (patterntotal/drugtypetotal)*100 as pct2,
pattern as grp from patterntot a left join drugtot b
on a.switch_no=b.switch_no and a.drug_type = b.drug_type
order by drug_type, grp;
quit;

/* Create high low values to create stacked bar chart */
data step2_highlow;
set drugpct;
by switch_no drug_type;
if first.switch_no then ybase=0;

```

```

ymin = ybase;
ymax = pct1+ybase;
yscatter = (pct1/2)+ybase; output;
ybase + pct1;
label
drug_type = "treatment group";
run;

```

```

proc sql;
create table step3 as
select a.* , b.pct2, b.grp , (pct1*pct2) *100 as actual_pct2, pattern
from step2_highlow a left join patternpct b
on a.switch_no=b.switch_no and a.drug_type = b.drug_type
order by drug_type, grp;
quit;

```

```

/* Create annotate dataset */
data getattrs;
input id $ 1 value $ 3-4 fillcolor $ 6-10 linecolor $ 11-14 linepattern $ 16-20;
cards;
c a blue blue solid
c b red red solid
d aa blue blue solid
d ba red red solid
d ab blue blue solid
d bb red red solid
;
run;

```

```

/* Create basic switch pattern plot – basic Sankey style where thickness represents the flow */
/* using PROC SGPLOT */

```

```

title "treatment switch patterns";
footnote1 "a -> a (50% of initial a) a -> b (50% of initial a)";
footnote2 "b -> a (66.6% of initial b) b -> a (33.3% of initial b)";
ods graphics / reset width=5in height=3in;
proc sgplot data=step3 dattrmap=getattrs;
highlow x=switch_no low=ymin high=ymax/ type=bar barwidth=0.1 group=drug_type
groupdisplay=overlay attrid=c;
xaxis label = "switch level 0=initial drug 1=first switch" values=('initial' 'first switch') ;
yaxis label = "percentage of patients" values =(0 to 100 by 20);
scatter x=switch_no y=yscatter/ markerchar= drug_type markerchar=pct1 markercharattrs=(size=15pt
color=black) ;
series x=switch_no y=yscatter / group=pattern transparency=0.5 smoothconnect attrid=d
thickresp=pct2;
run;

```

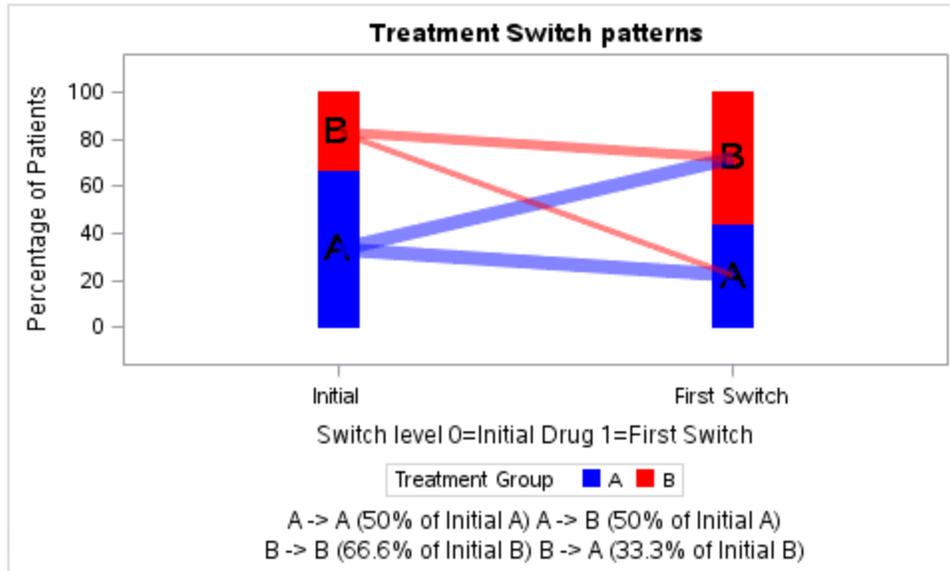


Figure 32: Sankey style plot to show treatment patterns.

SURVEY DATA ANALYSIS:

HEOR studies involve analyzing data from several big data sources such as medical claims, Electronic Health Records (EHR), and integrated databases so forth. On the other HEOR studies are also performed using survey data which involves collecting data from a sample of patients. The survey data collection can be a complex method which assigns appropriate weights and other variables such as strata and cluster to the population to make it representative of a larger sample. SAS has inbuilt survey methods that can be used to perform this survey analysis. In the below example, we will just show the syntax of how a sample survey data with assigned variables weight, stratum, and cluster (bold) could be used in the syntax shown to create data visualizations.

When analyzing continuous variables, PROC SURVEYMEANS can be used directly with plots option to create graphs.

SAS code to create survey continuous variable plot (Figure 33):

EXAMPLE 1:

```
/* Create graph for survey data continuous variable using PROC SURVEYMEANS */
proc surveymeans data = sample_survey_data sum plots=all;
stratum stratum;
cluster cluster;
weight weight;
var variable;
run;
```

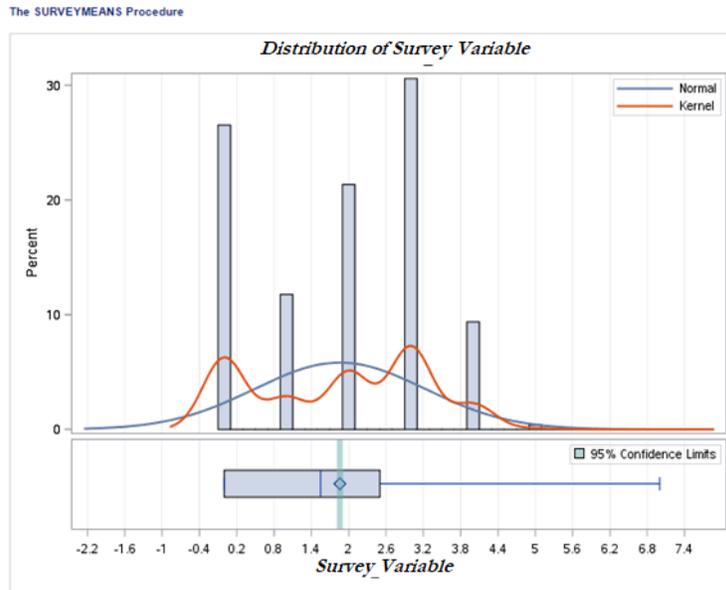


Figure 33: Distribution of survey data continuous variable.

When analyzing categorical variables, PROC SURVEYMEANS cannot be used directly to create graphs. The output from PROC SURVEYMEANS can be used in PROC SGPLOT to create graphs as shown below.

SAS code to create survey categorical plot (Figure 34):

EXAMPLE 2:

/ Create graph for survey data categorical variable using PROC SGPLOT*/*

```
proc sgplot data=meansoutput ;
band x=varlevel lower=lowerclmean upper=upperclmean / group=sex
transparency=.2 ; xaxis integer label="variable categories";
yaxis label="weighted mean" valuesformat=data;
series y=mean x=varlevel /group=grp datalabel ;
run ;
```

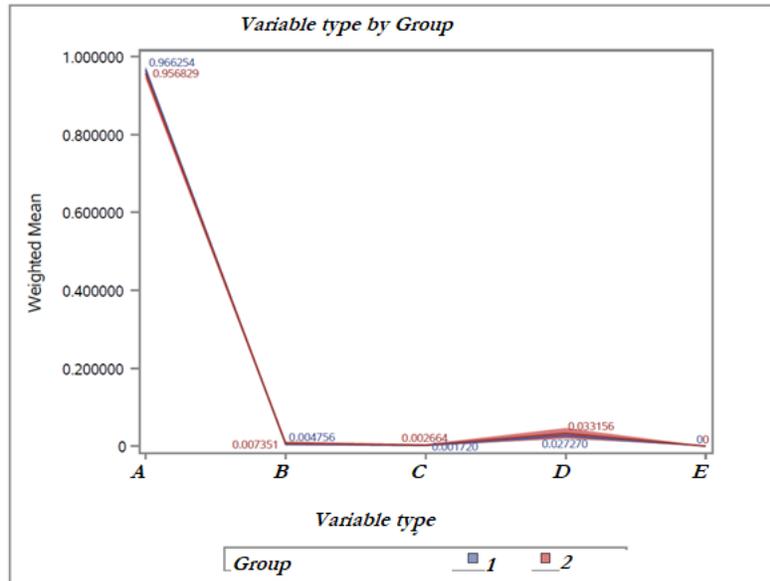


Figure 34: Weighted statistics for survey data categorical variable.

Odds ratio plot can be created for any data including survey data. In survey data, odds ratio plot can be created using PROC SURVEYFREQ with plots option specify oddsratioplot option.

SAS code to create survey odds ratio plot (Figure 35):

EXAMPLE 3:

/ Create odds ratio plot using PROC SURVEYFREQ */*

```
proc surveyfreq data=sample_survey data order=formatted;
stratum stratum;
cluster cluster;
weight weight;
tables group1*variable*group2 / or plots(only)=oddsratioplot relrisk ;
run ;
```

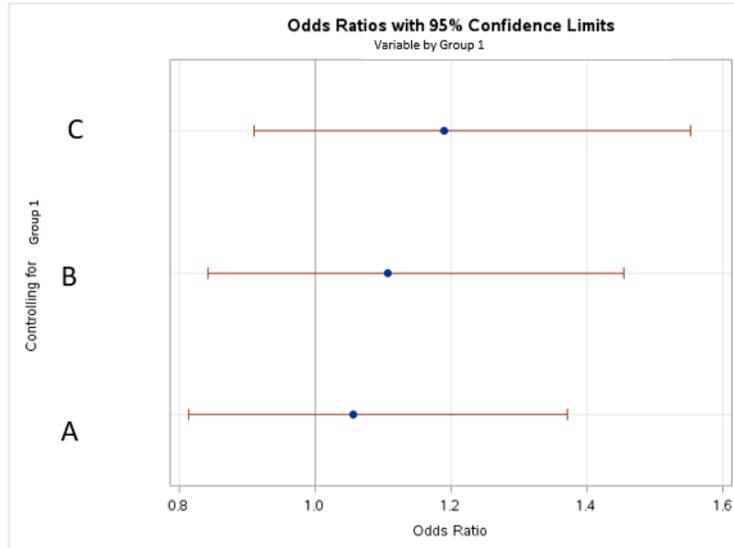


Figure 35: Odds ratio plot in survey data.

REFERENCES:

1. Robert L. Kane and David M. Radosevich, Conducting Health Outcomes Research
2. SAS
https://www.sas.com/en_us/home.html
<https://documentation.sas.com/?docsetId=helpcenterwlc&docsetTarget=home.htm&docsetVersion=1.0&locale=en>
3. Albert Spriestersbach 1, Bernd Röhrig, Jean-Baptist du Prel, Aslihan Gerhold-Ay, Maria Blettner, Descriptive statistics: the specification of statistical measures and their presentation in tables and graphs. Part 7 of a series on evaluation of scientific publications
4. Stephen R. Midway, Principles of Effective Data Visualization
5. Duke library for DV
<https://guides.library.duke.edu/datavis/topten>
6. Travis Murphy, Infographics by SAS
7. Robert Allison, Visualizing Data with SAS
8. SAS Support
<https://support.sas.com/en/support-home.html>
9. SAS Blogs
<https://blogs.sas.com/content/>

CONTACT INFORMATION:

Swapna Ambati
 Eliassen Group
 400 Atrium Drive, 4th Floor South
 Somerset, NJ-08873
sambati@eliassen.com

TRADEMARK

TRADEMARK SAS and all other SAS Institute Inc. products 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