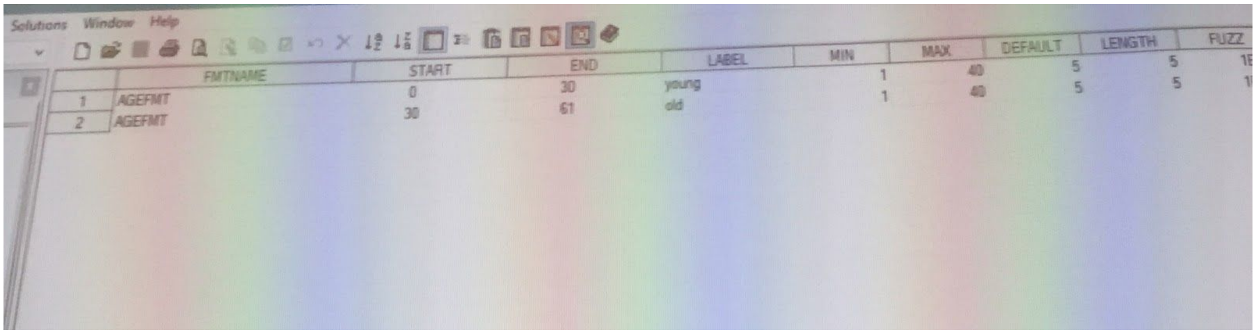
CLASS 31 (CONTINUED)

PROC FORMAT with the CNTLOUT= and CNTLIN= options:

USING A FORMAT TO CREATE A DATA SET [CNTLOUT= (control out)]:

```
proc formats lib=sasuser.formats cntlout=xx;
Select agefmt;   // this age format we have created earlier having the range 0-30 (young) and 31-60 (old) , control out will put the existing format into a dataset.//
Run;
```

Note : here ‘xx’ dataset is created having format name (here we created age format which will come under format name) , start value, end value (this is basically the range we have given like 0-30, 31-60), label (which will have the names we have given like ‘young’ for 0-30 and ‘old’ for 31-60).



	FMTNAME	START	END	LABEL	MIN	MAX	DEFAULT	LENGTH	FUZZ
1	AGEFMT	0	30	young	1	40	5	5	1E
2	AGEFMT	30	61	old	1	40	5	5	1

USING A SAS DATA SET TO CREATE A FORMAT [CNTLIN= (control in)]:

Control in is used to create format from a dataset and for that three key variables are required:
1. FMTNAME : a character variable that specifies name of format to create. Give format name in quotes.
2. LABEL: the label you wish to generate. Label for numeric should be numeric value.
3. START: the value you want to format into a label.

Suppose we have datasets ‘big’ and ‘small’ and we want look up for the values of ‘small’ dataset into the values of ‘big’ dataset. In this case we will use format. We can do this with merge and join also but proc format is efficient.

CODE:

```
Data big;
Input id;
Datalines;
1
2
3
4
5
6
7
8
9
10
;
Run;

Data small;
Input id;
Datalines;
1
2
3
;
Run;
```

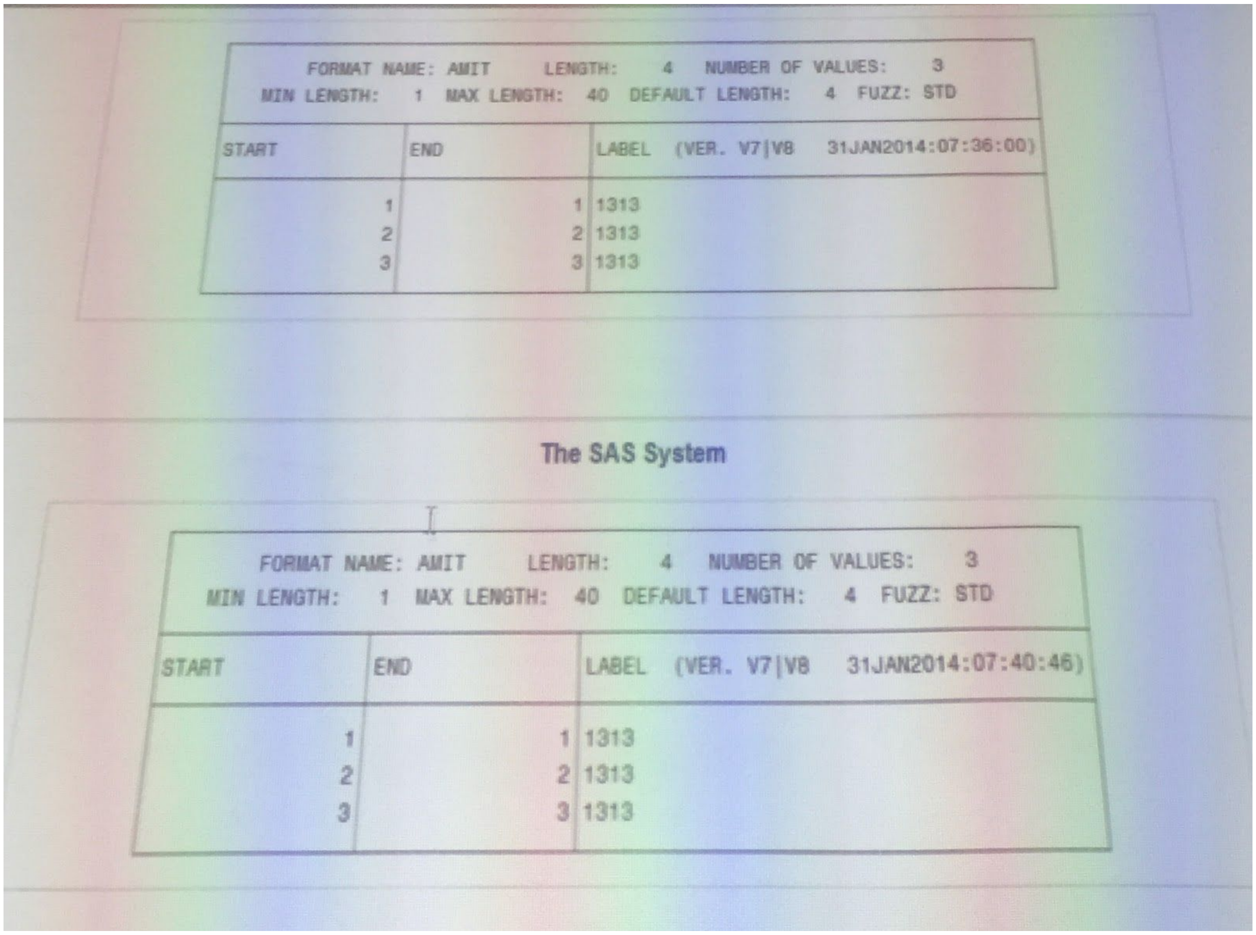
```
data x;
Set small;  // dataset ‘x’ is created and values from small is copied into it and the three key
variables required to create a format is used below //
fmtname=’amit’;   /* Give format name in quotes */
label=1313;          /* label for numeric should be numeric */
rename id=start;  /*start should have the values of key variable */
Run;
```

Note: Values of format name and label can be hard-cored.

```
 proc format cntlin=x;  /* here the dataset created , we will make it format using control in.
Format catalogue is created at this step, */
 Run;
```

---

data match;
Set big;
If put (id,amit.)=1313 then output match; // On ‘id’ (having 10 values) apply format ‘amit.’ (created above having 3 values , if id matches and the value is 1313 [which is equal to label] , then put it into match. Format compares the value and replace it with label value It is working like “inner join”)
Run;



```
   FORMAT NAME: AMIT      LENGTH:    4   NUMBER OF VALUES:    3
MIN LENGTH:    1  MAX LENGTH:  40  DEFAULT LENGTH:    4  FUZZ: STD
START              END                LABEL  (VER. V7|V8   31JAN2014:07:36:00)
                 1                  1 1313
                 2                  2 1313
                 3                  3 1313

                         The SAS System

   FORMAT NAME: AMIT      LENGTH:    4   NUMBER OF VALUES:    3
MIN LENGTH:    1  MAX LENGTH:  40  DEFAULT LENGTH:    4  FUZZ: STD
START              END                LABEL  (VER. V7|V8   31JAN2014:07:40:46)
                 1                  1 1313
                 2                  2 1313
                 3                  3 1313
```

Another Example: We want the observations which are present in ‘big’ dataset but not in ‘small’ dataset.

Data big;
Input id;
Datalines;
1
2

```
3
4
5
6
7
8
9
10
;
Run;

Data small;
Input id;
Datalines;
1
2
3
;
Run;
```

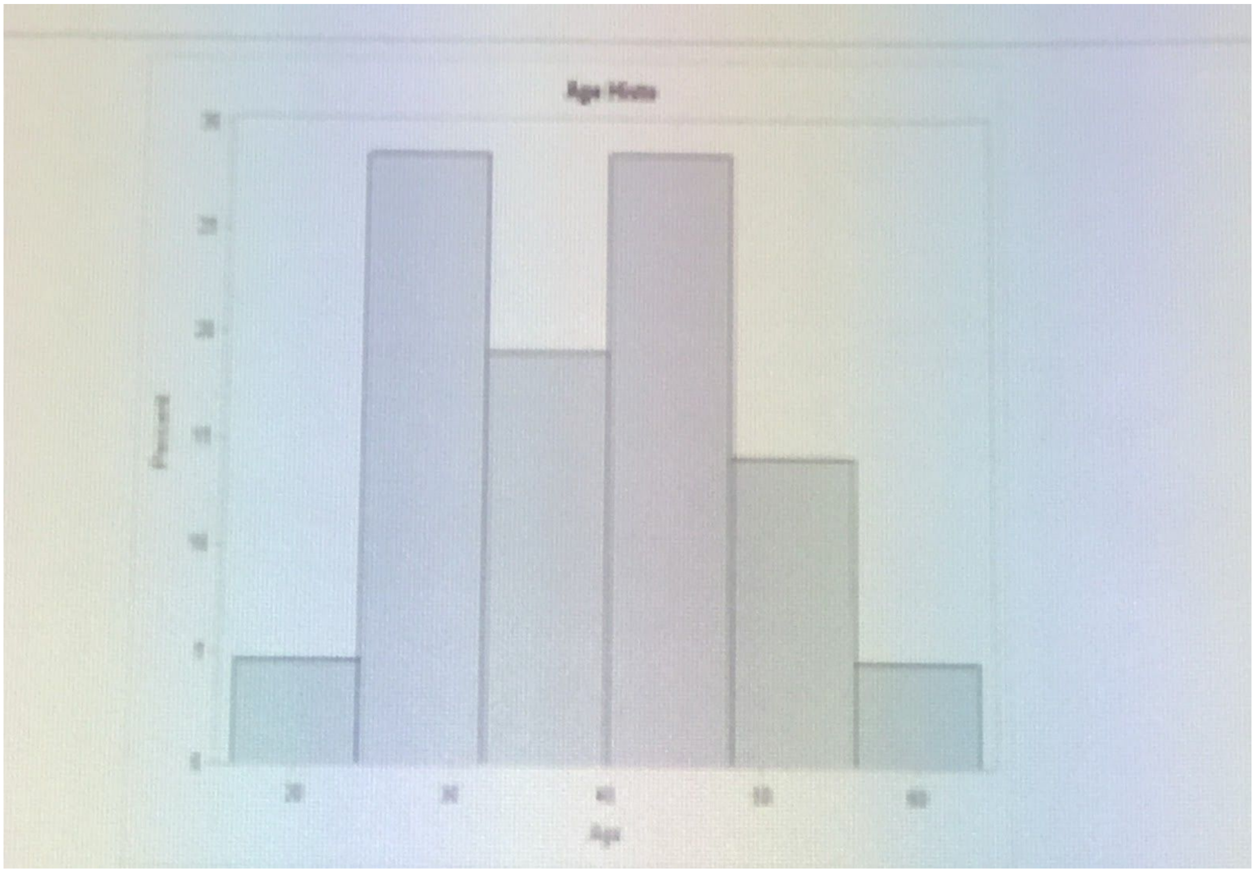
```
data x;
Set small;
fmtname='amit';    // same steps followed from above example //
label=1313;
rename id=start;
Run;
```

```
proc format cntlin=x;
 Run;
```

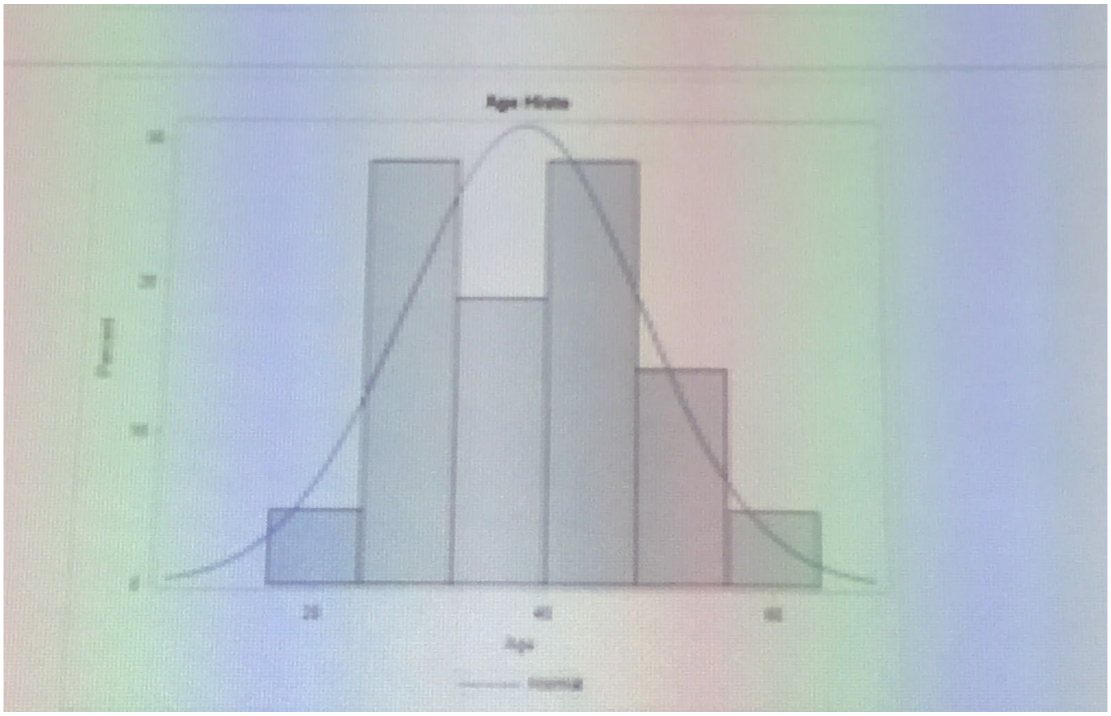
```
data match big_no_small;
 Set big;
If put (id,amit.)=1313 then output match;
Else if put (id,amit.) ne 1313 then output big_no_small;  // this will give the values from 4 to 10
from 'big' dataset that are not present in 'small' dataset , right join-inner join//
Run;
```

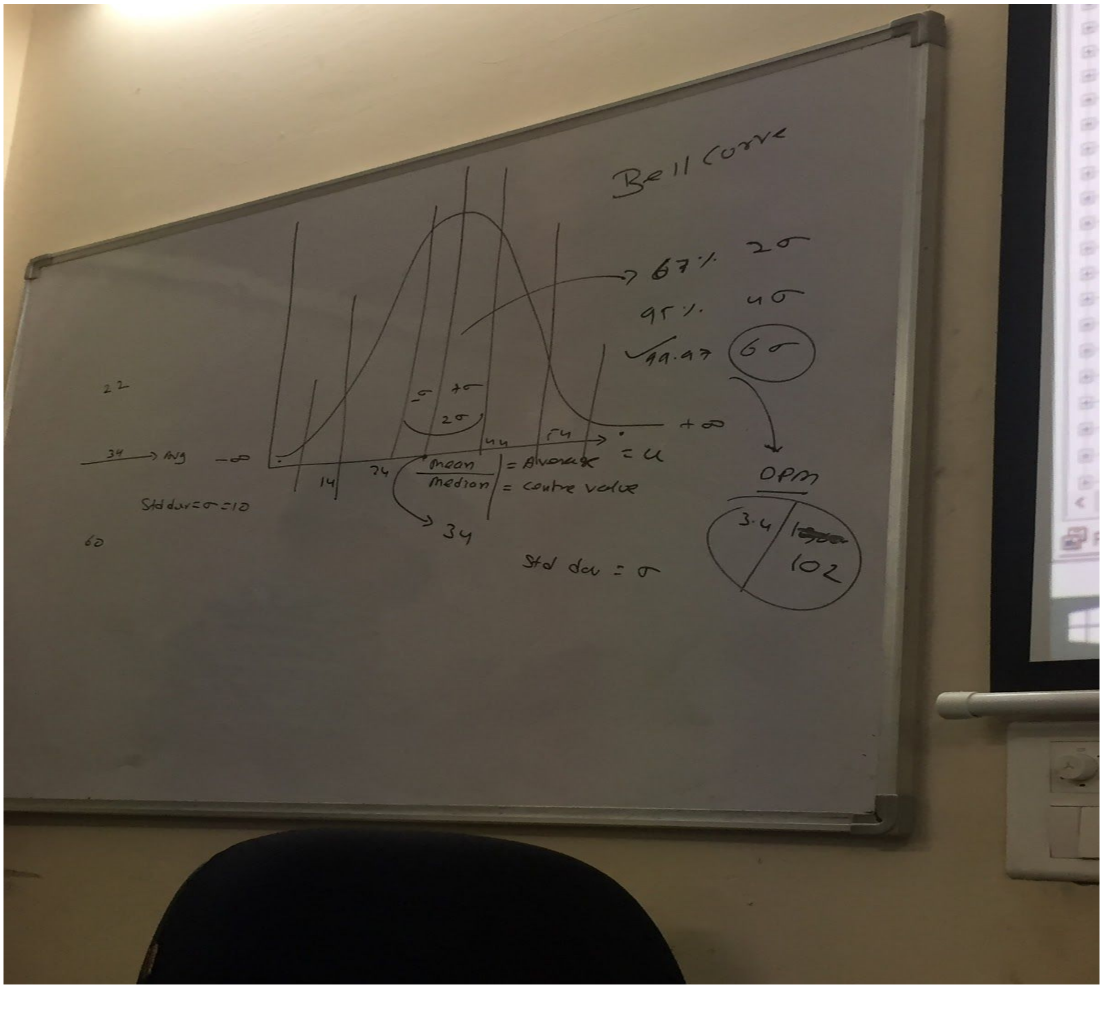
HISTOGRAMS: Histograms show the distribution of a continuous variable.

```
Proc SGPLOT DATA = sasuser.admit;
HISTOGRAM age;
TITLE “Age Histo”;
Run;
```



```
Proc SGPLOT DATA = sasuser.admit;
HISTOGRAM age;
Density age; // this will be indicated by bell-curve//
TITLE “Age Histo”;
Run;
```


Age Histo
30
20
Percent
10
0
20
40
60
Age
Normal

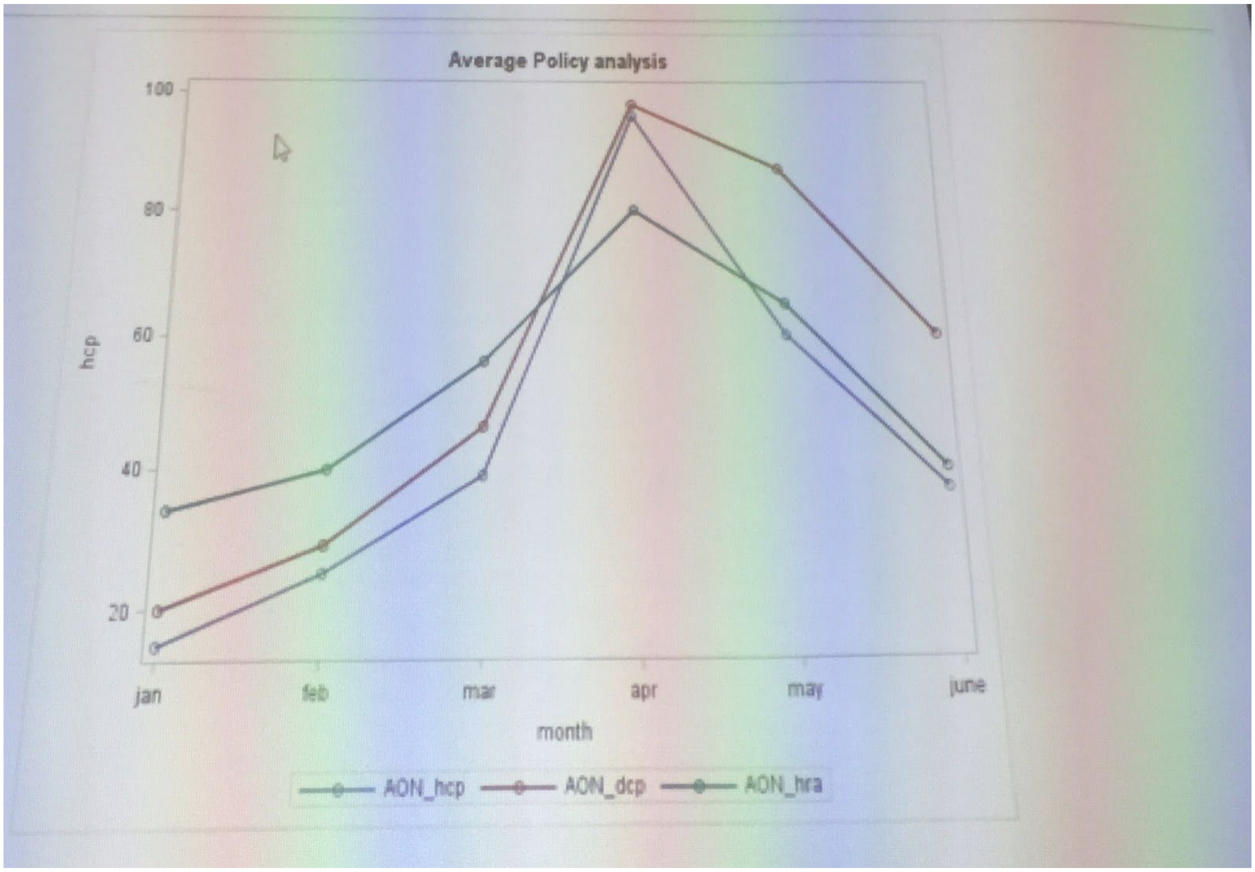

Bell curve
67% 2σ
95% 4σ
✓99.97 6σ
DPM
3.4 / 1~~000~~
102
-σ +σ
2σ
+∞
-∞
22
34 → Avg
Std dev = σ = 10
60
14
24
44
54
= u
mean = Average
median = centre value
34
Std dev = σ

SERIES PLOTS :In a series plot, the data points are connected by a line.Data for series plots must be sorted by the X variable.
The SGPLOT procedure produces a variety of graphs including bar charts, scatter plots, and line graphs.

CODE:

```
Data policy ;
Input month $ hcp (health care plan) dcp (dental care plan) hra (house rent allowance);
Cards;
Jan 15 20 34
Feb 25 29 40
Mar 39 46 56
Apr 96 98 80
May 60 87 65
June 37 60 40
;
Run;

Proc SGPLOT DATA =policy;
SERIES X = Month Y = hcp / LEGENDLABEL =’AON_hcp’ MARKERS
LINEATTRS=(THICKNESS=2);
SERIES X = Month Y = dcp / LEGENDLABEL =’AON_dcp’ MARKERS
LINEATTRS=(THICKNESS=2);
SERIES X = Month Y = hra / LEGENDLABEL =’AON_hra’ MARKERS
LINEATTRS=(THICKNESS=2);
Title ‘average policy analysis’;
Run;
```



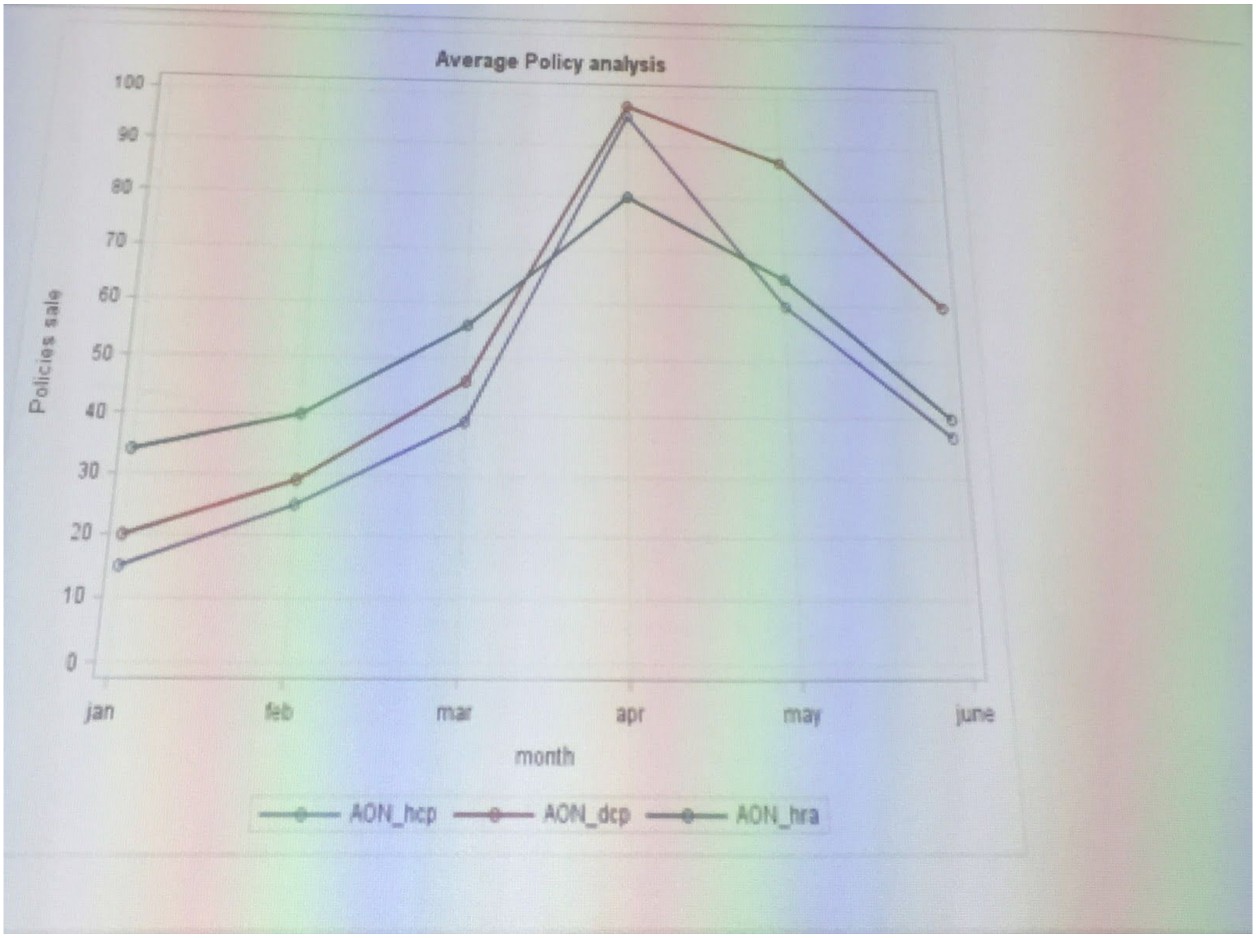
Note: From the line graph we can see that enrollments for plans are high in april month.

```
Data policy ;
Input month $ hcp dcp hra;
Cards;
Jan 15 20 34
Feb 25 29 40
Mar 39 46 56
Apr 96 98 80
May 60 87 65
June 37 60 40
;
Run;

Proc SGPLOT DATA =policy;
SERIES X = Month Y = hcp / LEGENDLABEL ='AON_hcp' MARKERS
LINEATTRS=(THICKNESS=2);
```

```
SERIES X = Month Y = dcp / LEGENDLABEL =’AON_dcp’ MARKERS
LINEATTRS=(THICKNESS=2);
SERIES X = Month Y = hra / LEGENDLABEL =’AON_hra’ MARKERS
LINEATTRS=(THICKNESS=2);
XAXIS TYPE = DISCRETE GRID; YAXIS LABEL =’Policies sale’ GRID VALUES = (0 TO 100
BY 10);
TITLE ‘Average policy analysis’;
Run;
```

Note: GRID VALUES = (0 TO 100 BY 10) - by this boxes will be created in the graph.

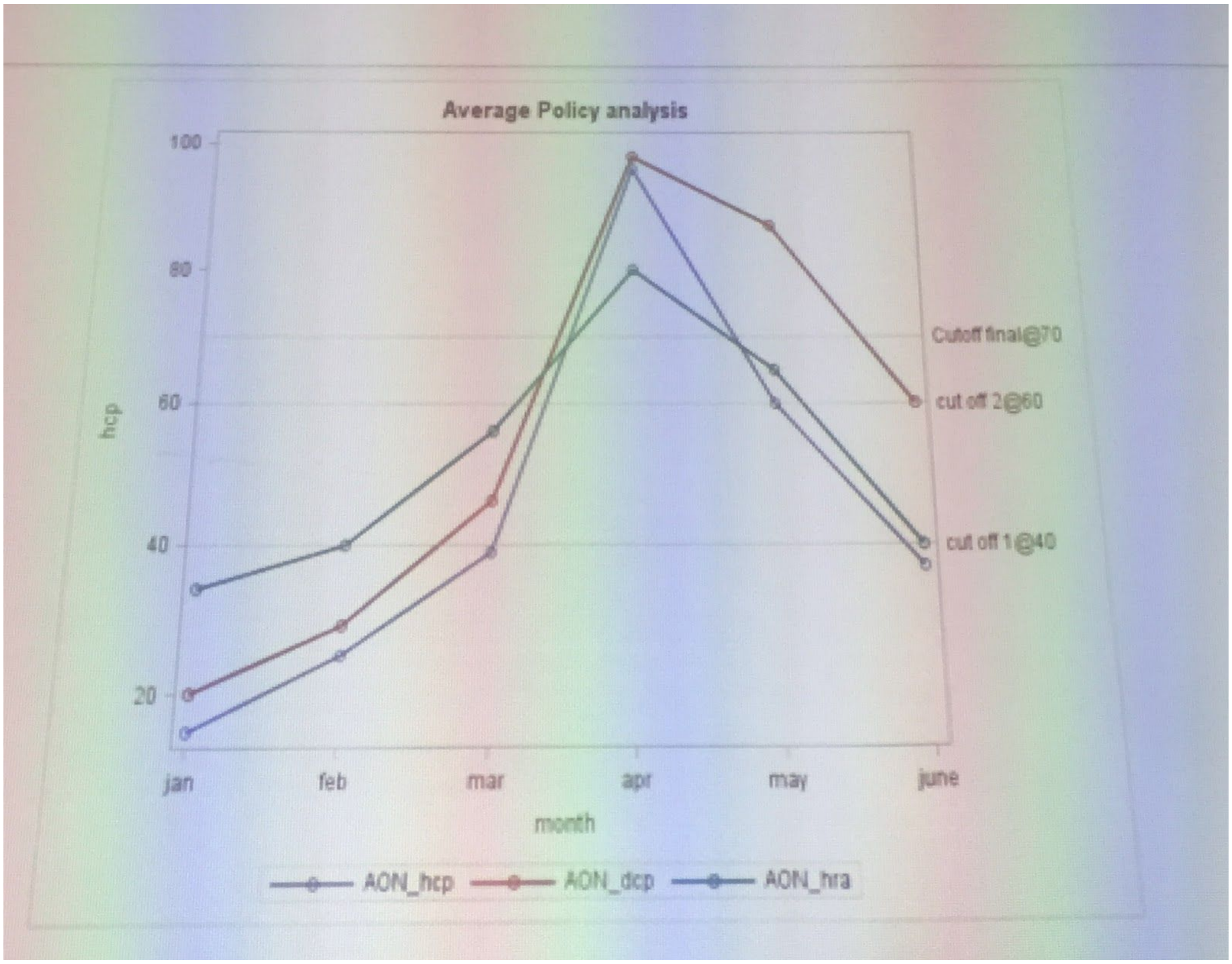


```
Data policy ;
Input month $ hcp dcp hra;
Cards;
Jan 15 20 34
Feb 25 29 40
Mar 39 46 56
Apr 96 98 80
```

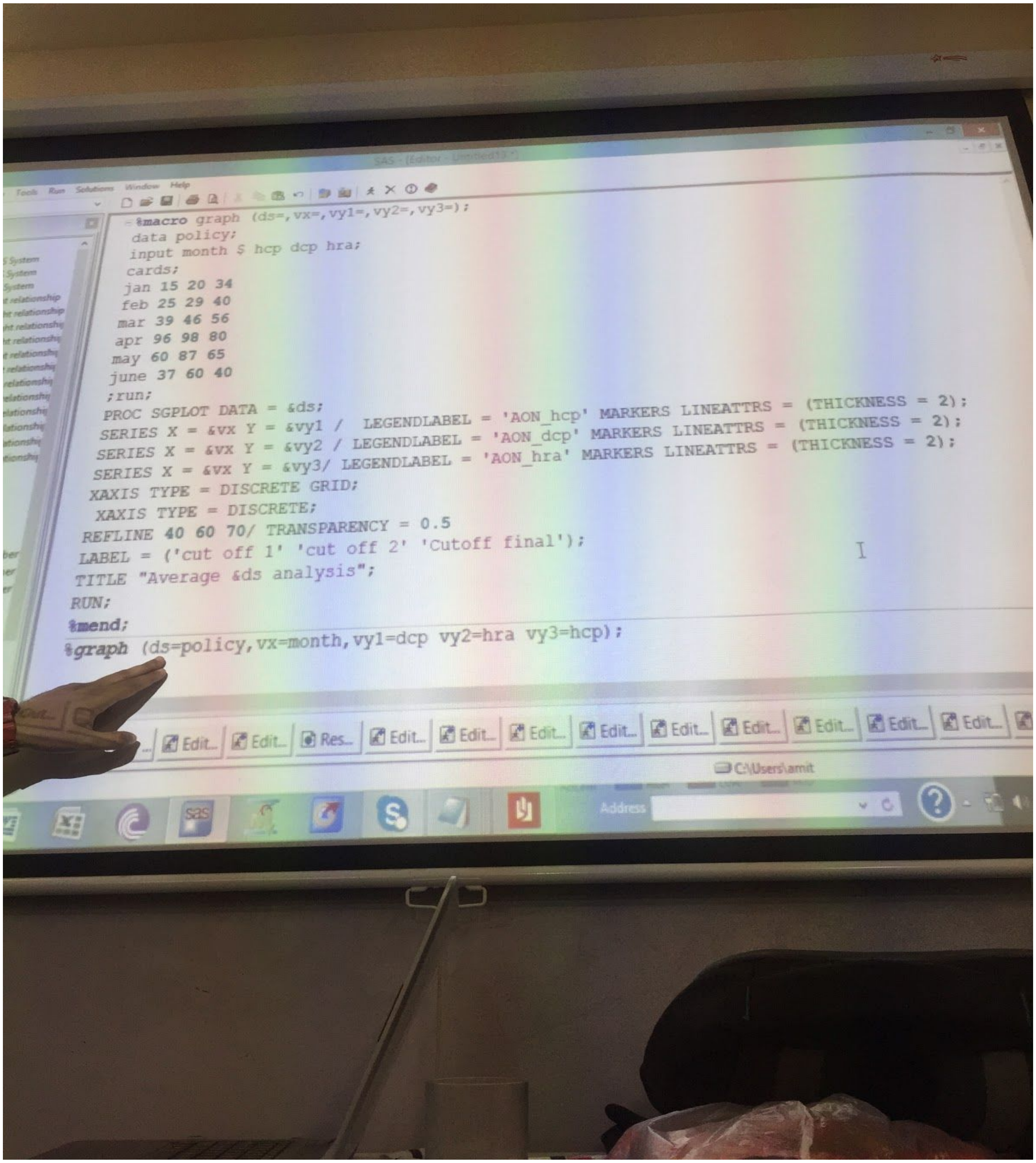
```
May 60 87 65
June 37 60 40
;
Run;

Proc SGPLOT DATA =policy;
SERIES X = Month Y = hcp / LEGENDLABEL =’AON_hcp’ MARKERS
LINEATTRS=(THICKNESS=2);
SERIES X = Month Y = dcp / LEGENDLABEL =’AON_dcp’ MARKERS
LINEATTRS=(THICKNESS=2);
SERIES X = Month Y = hra / LEGENDLABEL =’AON_hra’ MARKERS
LINEATTRS=(THICKNESS=2);
REFLINE 40 60 70/ TRANSPARENCY = 0.5
LABEL=(‘cut off 1@40’ ‘cut off 2@60’ ‘cut off final@70’);
TITLE ‘Average Policy analysis’;
Run;
```

Note:LABEL=('cut off 1@40' 'cut off 2@60' 'cut off final@70')- by this three cut-offs are defined which can be seen in the graph..



SAS macro we have created for the above example:



```
%macro graph (ds=,vx=,vy1=,vy2=,vy3=);
 data policy;
 input month $ hcp dcp hra;
 cards;
 jan 15 20 34
 feb 25 29 40
 mar 39 46 56
 apr 96 98 80
 may 60 87 65
 june 37 60 40
 ;run;
 PROC SGPLOT DATA = &ds;
 SERIES X = &vx Y = &vy1 /  LEGENDLABEL = 'AON_hcp' MARKERS LINEATTRS = (THICKNESS = 2);
 SERIES X = &vx Y = &vy2 / LEGENDLABEL = 'AON_dcp' MARKERS LINEATTRS = (THICKNESS = 2);
 SERIES X = &vx Y = &vy3/ LEGENDLABEL = 'AON_hra' MARKERS LINEATTRS = (THICKNESS = 2);
 XAXIS TYPE = DISCRETE GRID;
  XAXIS TYPE = DISCRETE;
 REFLINE 40 60 70/ TRANSPARENCY = 0.5
 LABEL = ('cut off 1' 'cut off 2' 'Cutoff final');
 TITLE "Average &ds analysis";
 RUN;
%mend;
%graph (ds=policy,vx=month,vy1=dcp vy2=hra vy3=hcp);
```