NAME : Mansi Dwivedi
BRANCH : IT
UID : 2019140016
BATCH : B
COURSE : Data Analytics Lab
EXPERIMENT : 2

AIM : Using the SAS software to analyze statistical data.

PROBLEM STATEMENT : Study and understand the workings of SAS studio by referring to the online materials and documentation, etc., and then implement a small problem.

THEORY :

1) **What is SAS?**
    - SAS stands for Statistical Analysis Software. It offers business intelligence and data management software and services through cutting-edge analytics. SAS turns data into insight, which might offer a new angle on how to conduct business.
    - In contrast to other BI solutions on the market, SAS uses considerable programming to transform and analyse data rather than just a simple drag-and-drop method.
    - Over the years, SAS has expanded its product offering with a number of solutions. It offers solutions for issues including fraud prevention, data governance, data quality, big data analytics, text mining, and health science, among others. We can safely assume SAS has a solution for every business domain and hence the popularity.

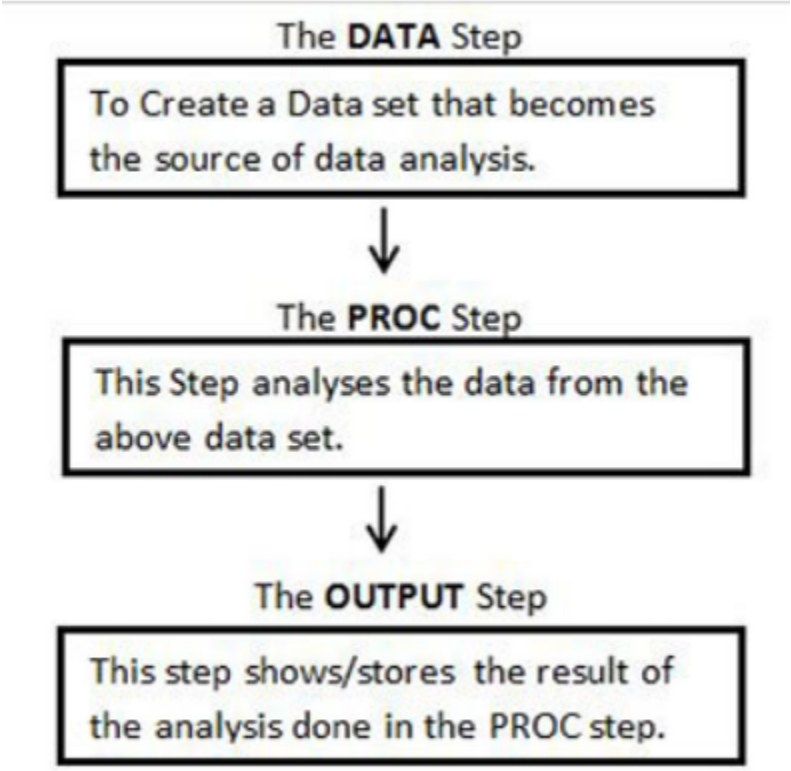
2) **Structure and Features of SAS?**
    - SAS has more than 20 components that include Base SAS (It is a core component which contains data management facility and a programming language for data analysis. It is also the most widely used), SAS/GRAPH (to create graphs, presentations for better understanding and showcasing the result in a proper format), SAS/STAT (to perform Statistical analysis with the variance analysis,

regression, multivariate analysis, survival analysis, and psychometric analysis, mixed model analysis) and much more.

- SAS provides a feature called libraries that are like storages. SAS gives us the option to make several libraries. There are only 8 characters in an SAS library. The two types of libraries are : Temporary or Work Library and Permanent Library.

3) SAS Program Structure

- In order to create a SAS program the following steps need to be followed :



CODE & OUTPUT:

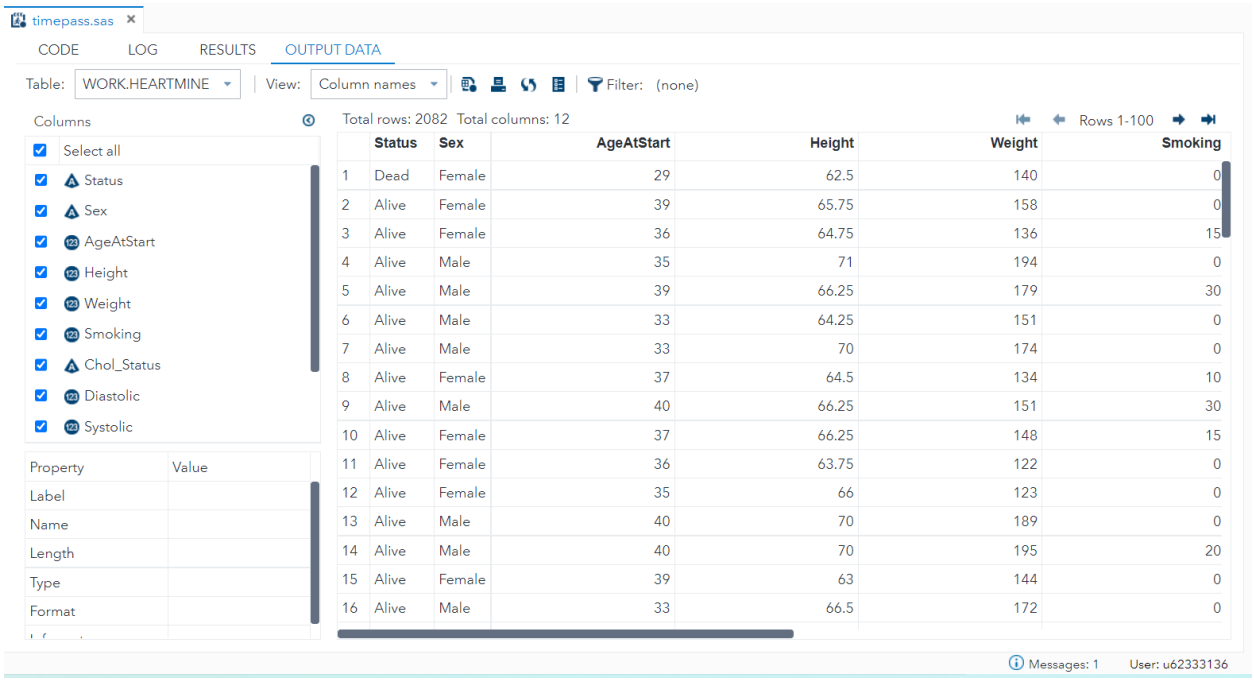
1) For the small problem that we had to implement I chose to use the already present database in the SAS studio (My Libraries -> SASHELP -> HEART) just made some modifications by including only the required columns and named the final table as 'heartmine' which according to the SAS structure got stored in the WORK folder. Here I have directly used the SQL procedures to process the SQL statements.

timepass.sas

CODE LOG RESULTS

```
1  PROC SQL;
2  create table heartmine as
3  SELECT Status, Sex, AgeAtStart, Height, Weight, Smoking, Chol_Status,Diastolic,Systolic,BP_Status,Weight_Status,Smoking_Status
4     FROM
5     SASHELP.HEART
6     WHERE AgeAtStart between 20 and 40
7  ;
8  RUN;
```

The final table looks like this :



	Status	Sex	AgeAtStart	Height	Weight	Smoking
1	Dead	Female	29	62.5	140	0
2	Alive	Female	39	65.75	158	0
3	Alive	Female	36	64.75	136	15
4	Alive	Male	35	71	194	0
5	Alive	Male	39	66.25	179	30
6	Alive	Male	33	64.25	151	0
7	Alive	Male	33	70	174	0
8	Alive	Female	37	64.5	134	10
9	Alive	Male	40	66.25	151	30
10	Alive	Female	37	66.25	148	15
11	Alive	Female	36	63.75	122	0
12	Alive	Female	35	66	123	0
13	Alive	Male	40	70	189	0
14	Alive	Male	40	70	195	20
15	Alive	Female	39	63	144	0
16	Alive	Male	33	66.5	172	0

2) Now before starting any kind of data analyzation I took care of the missing values in the dataset by in this case replacing them with the mean values of that column
   Null values in SAS studio are either represented using a '.' in the case of numeric values or some alphabetical characters. For example :

timepass.sas | SASHELP.CARS | WORK.HEARTMINE

View: Column names | Filter: (none)

Total rows: 2082 Total columns: 12 | Rows 1-100

	Status	Sex	AgeAtStart	Height	Weight	Smoking	Chol_Status
1	Dead	Female	34	.	.	0	High
2	Alive	Female	34	.	.	5	Desirable
3	Alive	Female	36	54.75	127	0	Desirable
4	Alive	Female	40	55.5	124	15	Borderline
5	Alive	Female	33	55.75	94	0	High
6	Alive	Female	33	56	118	0	Borderline
7	Alive	Male	33	56	148	0	High
8	Alive	Female	30	56.25	98	0	High

Replacing the null or blank values of all the rows by substituting mean values :

```
15
16 proc stdize data=work.heartmine
17     out=work.heartmine
18     reponly method=mean;
19 run;
20
```

Checking whether the exact values are getting substituted or not

timepass.sas | SASHELP.CARS | WORK.HEARTMINE

View: Column names | Filter: (none)

Total rows: 2082 Total columns: 12 | Rows 1-100

	Status	Sex	AgeAtStart	Height	Weight	Smoking	Chol_Status
1	Alive	Female	36	54.75	127	0	Desirable
2	Alive	Female	40	55.5	124	15	Borderline

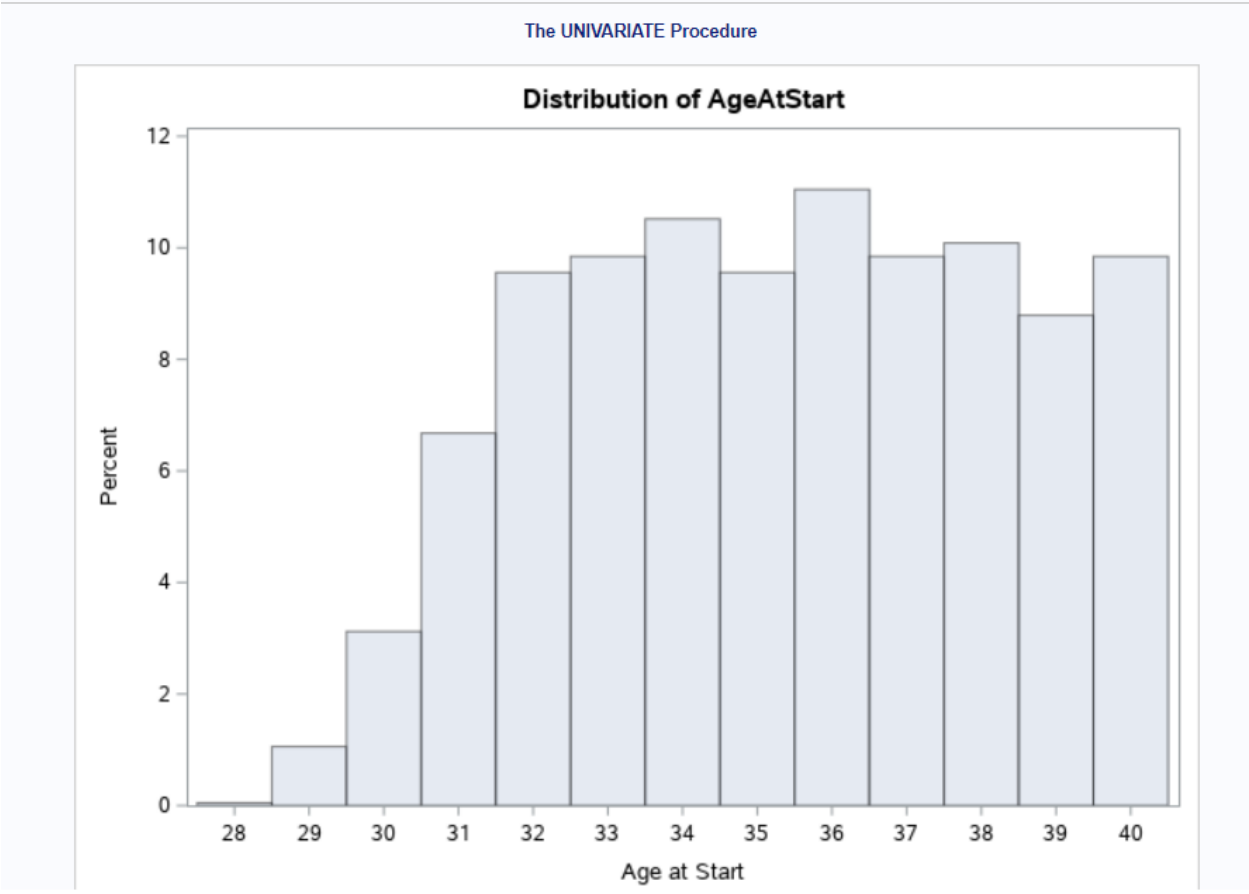
Then since the columns like Chol_Status, Weight_Status and Smoking_Status do not have numeric values in their case the missing values will have to be filled using a “None” filler, also not complicating the task by applying complex techniques for value filling.

```
20
21 PROC SQL;
22 UPDATE work.heartmine SET Chol_Status="None" where Chol_Status="";
23 PROC PRINT data = heartmine;
24 RUN;
25
26 PROC SQL;
27 UPDATE work.heartmine SET Weight_Status="None" where Weight_Status="";
28 PROC PRINT data = heartmine;
29 RUN;
30
31 PROC SQL;
32 UPDATE work.heartmine SET Smoking_Status="None" where Smoking_Status="";
33 PROC PRINT data = heartmine;
34 RUN;
```

Obs	Status	Sex	AgeAtStart	Height	Weight	Smoking	Chol_Status	Diastolic	Systolic	BP_Status	Weight_Status	Smoking_Status
1	Dead	Female	29	62.5000	140.000	0.0000	None	78	124	Normal	Overweight	Non-smoker
2	Alive	Female	39	65.7500	158.000	0.0000	High	80	128	Normal	Overweight	Non-smoker
3	Alive	Female	36	64.7500	136.000	15.0000	Desirable	80	112	Normal	Overweight	Moderate (6-15)
4	Alive	Male	35	71.0000	194.000	0.0000	Borderline	68	132	Normal	Overweight	Non-smoker
5	Alive	Male	39	66.2500	179.000	30.0000	Borderline	76	128	Normal	Overweight	Very Heavy (> 25)
6	Alive	Male	33	64.2500	151.000	0.0000	Borderline	68	108	Optimal	Overweight	Non-smoker
7	Alive	Male	33	70.0000	174.000	0.0000	Desirable	90	142	High	Overweight	Non-smoker
8	Alive	Female	37	64.5000	134.000	10.0000	Desirable	76	120	Normal	Normal	Moderate (6-15)
9	Alive	Male	40	66.2500	151.000	30.0000	Desirable	72	132	Normal	Overweight	Very Heavy (> 25)
10	Alive	Female	37	66.2500	148.000	15.0000	Desirable	78	110	Optimal	Overweight	Moderate (6-15)
11	Alive	Female	36	63.7500	122.000	0.0000	Desirable	84	132	Normal	Normal	Non-smoker
12	Alive	Female	35	66.0000	123.000	0.0000	Desirable	76	132	Normal	Normal	Non-smoker
13	Alive	Male	40	70.0000	189.000	0.0000	High	78	124	Normal	Overweight	Non-smoker
14	Alive	Male	40	70.0000	195.000	20.0000	Borderline	76	132	Normal	Overweight	Heavy (16-25)
15	Alive	Female	39	63.0000	144.000	0.0000	Desirable	80	120	Normal	Overweight	Non-smoker
16	Alive	Male	33	66.5000	172.000	0.0000	High	106	146	High	Overweight	Non-smoker
17	Alive	Male	31	68.7500	231.000	30.0000	Desirable	68	126	Normal	Overweight	Very Heavy (> 25)
18	Alive	Female	39	63.7500	120.000	0.0000	Desirable	80	130	Normal	Normal	Non-smoker
19	Alive	Female	38	62.0000	117.000	0.0000	Borderline	72	112	Optimal	Normal	Non-smoker
20	Alive	Female	40	64.5000	145.000	0.0000	Borderline	88	146	High	Overweight	Non-smoker
21	Alive	Female	34	68.7500	136.000	30.0000	High	86	134	Normal	Normal	Very Heavy (> 25)
22	Dead	Male	37	72.2500	153.000	30.0000	Borderline	76	116	Optimal	Normal	Very Heavy (> 25)
23	Alive	Female	31	65.2500	137.000	10.0000	Borderline	74	114	Optimal	Normal	Moderate (6-15)
24	Dead	Male	39	68.0000	175.000	5.0000	Borderline	84	122	Normal	Overweight	Light (1-5)
25	Alive	Female	34	64.5000	157.000	0.0000	High	78	154	High	Overweight	Non-smoker

3) Now that the data was in proper shape, next I used the SAS Data representation to view and analyze the data in a better manner. The first kind was plotting histograms which is a graphical display of data using bars of different heights. It groups the various numbers in the data set into many ranges. It also represents the estimation of the probability of distribution of a continuous variable. In SAS the **PROC UNIVARIATE** is used to create histograms.
Histogram for AgeAtStart variable - (Obtained the min and max value of the attribute by uilizing the GUI)

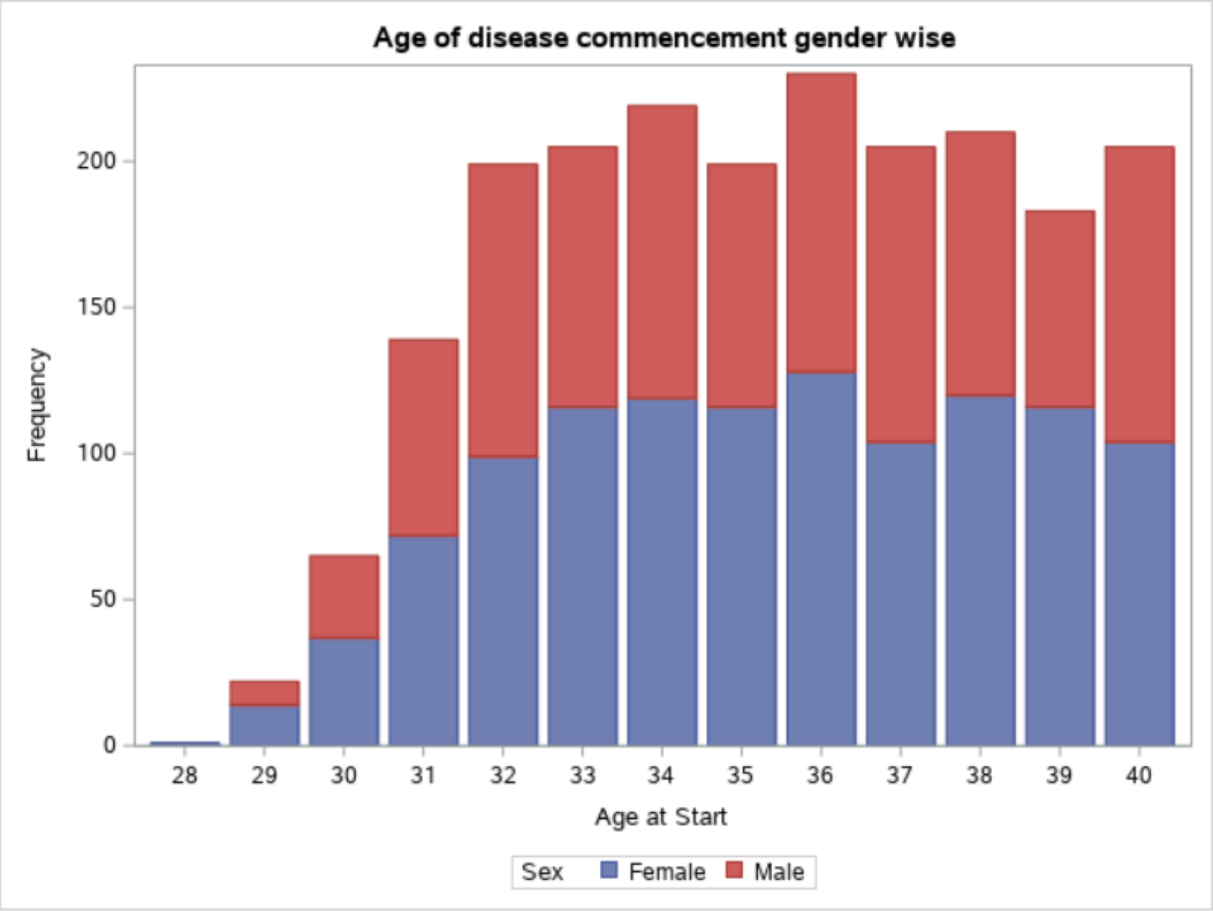
```
36 proc univariate data = work.heartmine;
37    histogram AgeAtStart
38    / midpoints = 28 to 40;
39 run;
```



Through the above output we can easily infer that as the age increases the chances of heart diseases increase with the peak being at 36 after which the curve dips down

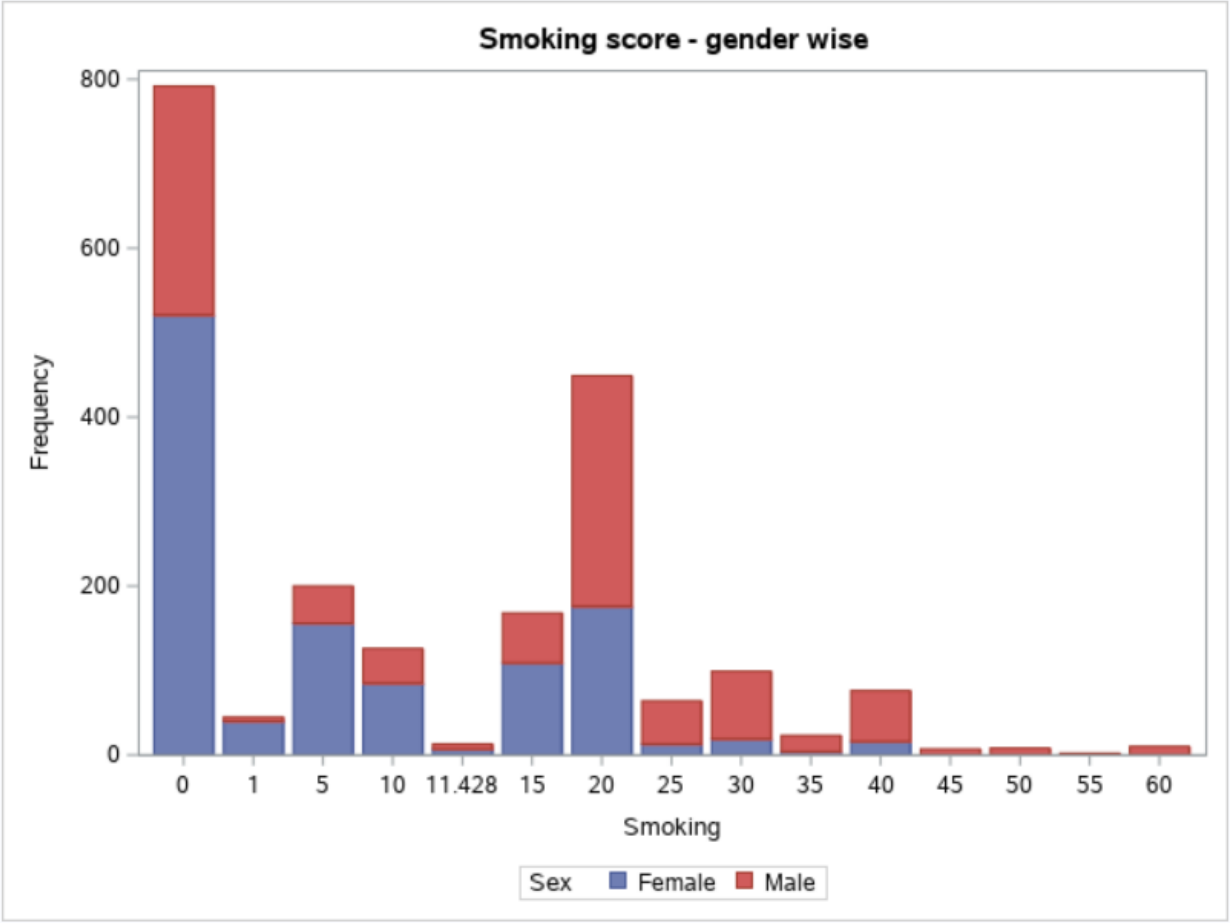
Stacked Bar Chart for comparing the number of males vs females who suffered the disease from a certain age :

```
42 proc SGPLOT data = work.heartmine;
43     vbar AgeAtStart /group = Sex ;
44     title 'Age of disease commencement gender wise';
45     run;
46 quit;
```



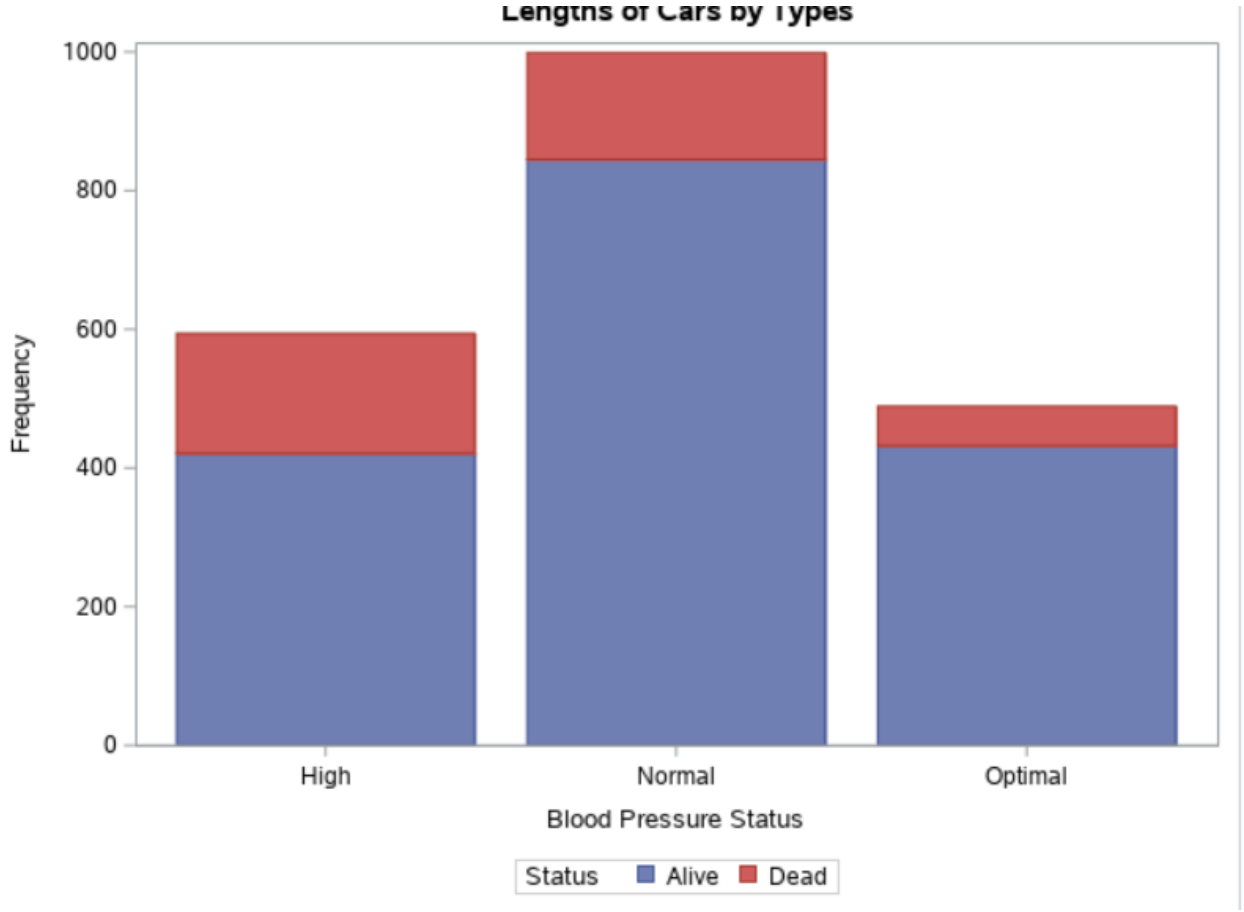
So here we can observe that there is not much of a difference in the numbers based on gender but in terms of conclusion the number of women patients are more at approximately all stages.

```
42 proc SGPLOT data = work.heartmine;
43     vbar Smoking /group = Sex ;
44     title 'Smoking score - gender wise';
45     run;
46 quit;
```



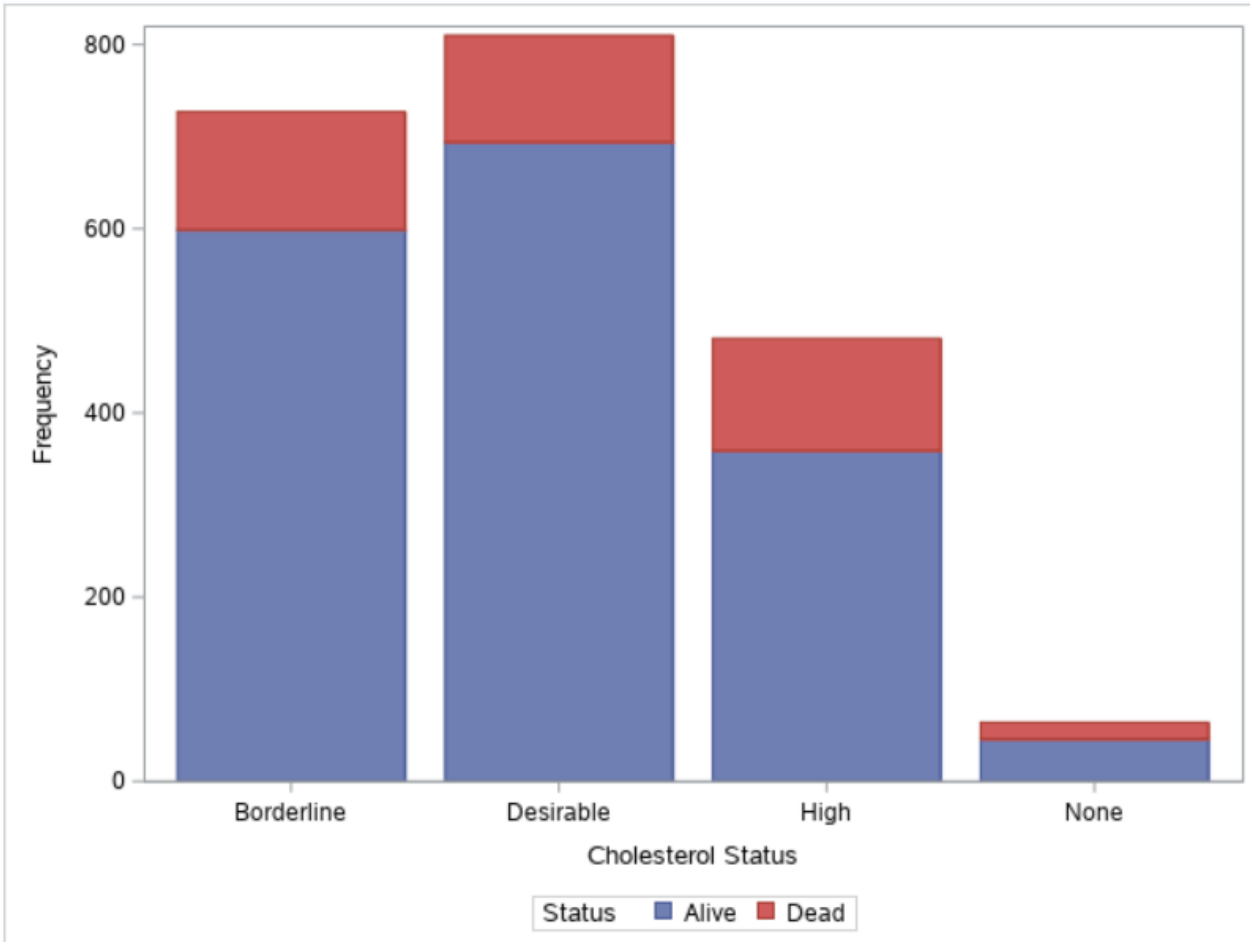
By looking at the above smoke vs gender plot we can clearly state that as the smoking score increases showing the increase in intensity the number of male smokers outweights the number of females eventually bringing it down to negligible.

```
136 proc SGPLOT data = work.heartmine;
137 vbar BP_Status /group = Status ;
138 title 'Lengths of Cars by Types';
139 run;
140 quit;
```

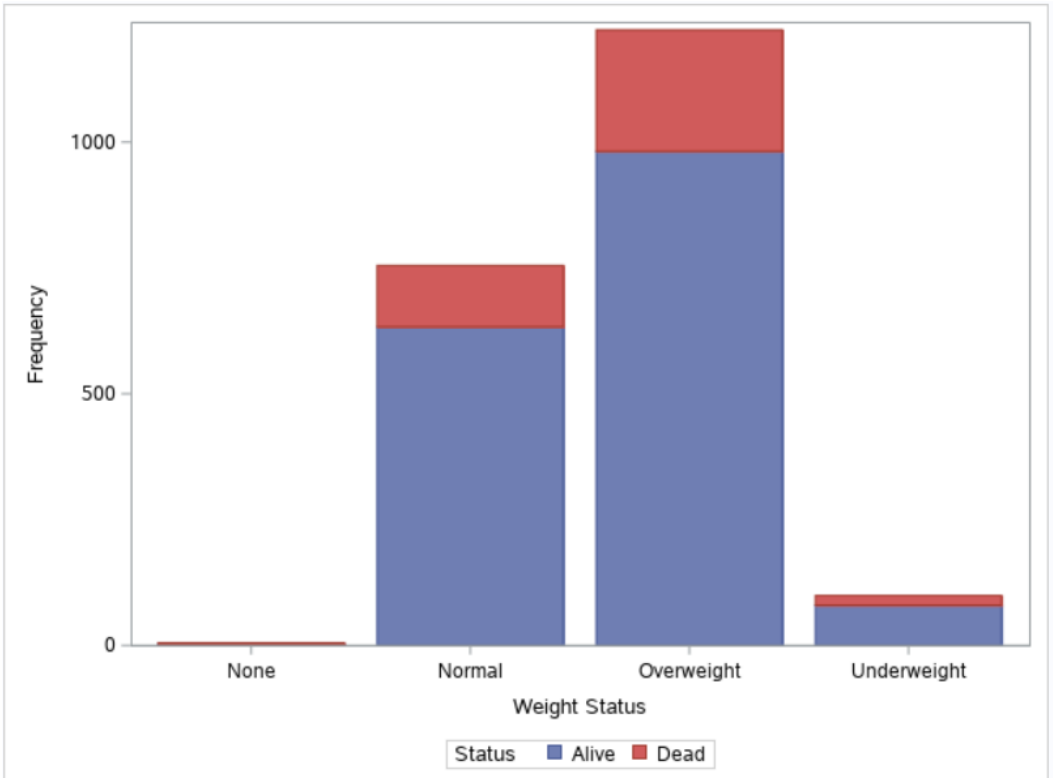


Here we can see that the BP status is definitely a factor in determining the status of the patient since we see that as intense the BP status becomes the frequency of people dying increases.

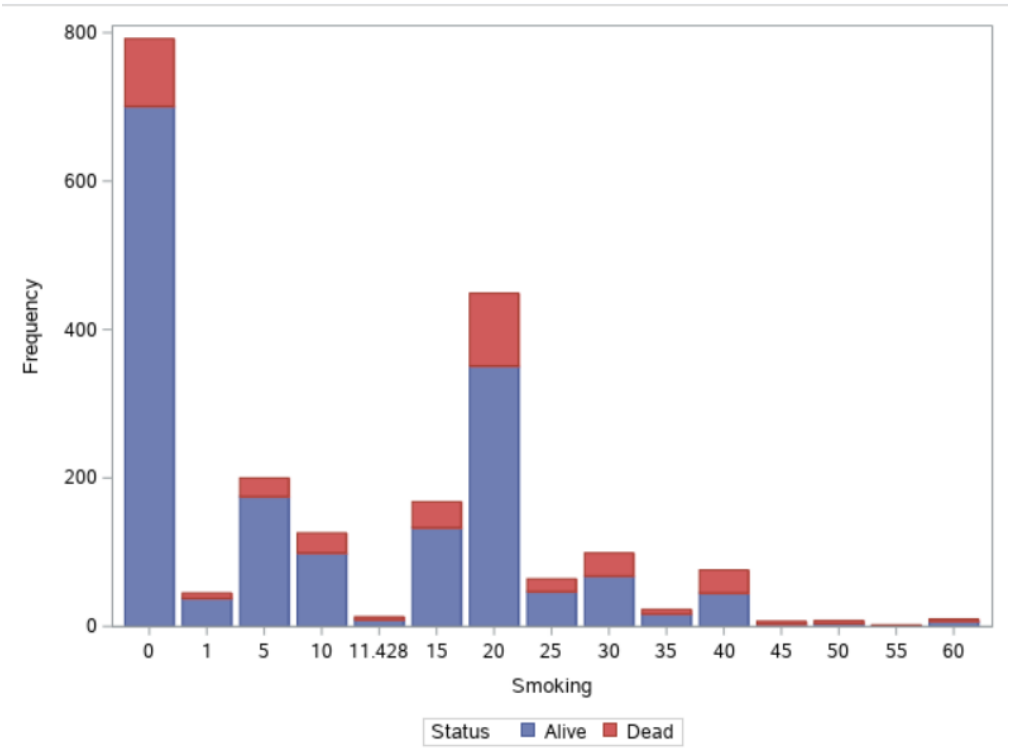
```
142 proc SGPLOT data = work.heartmine;
143 vbar Chol_Status /group = Status ;
144 run;
145 quit;
```



The Cholesterol status does not seem to affect the death of a person significantly.



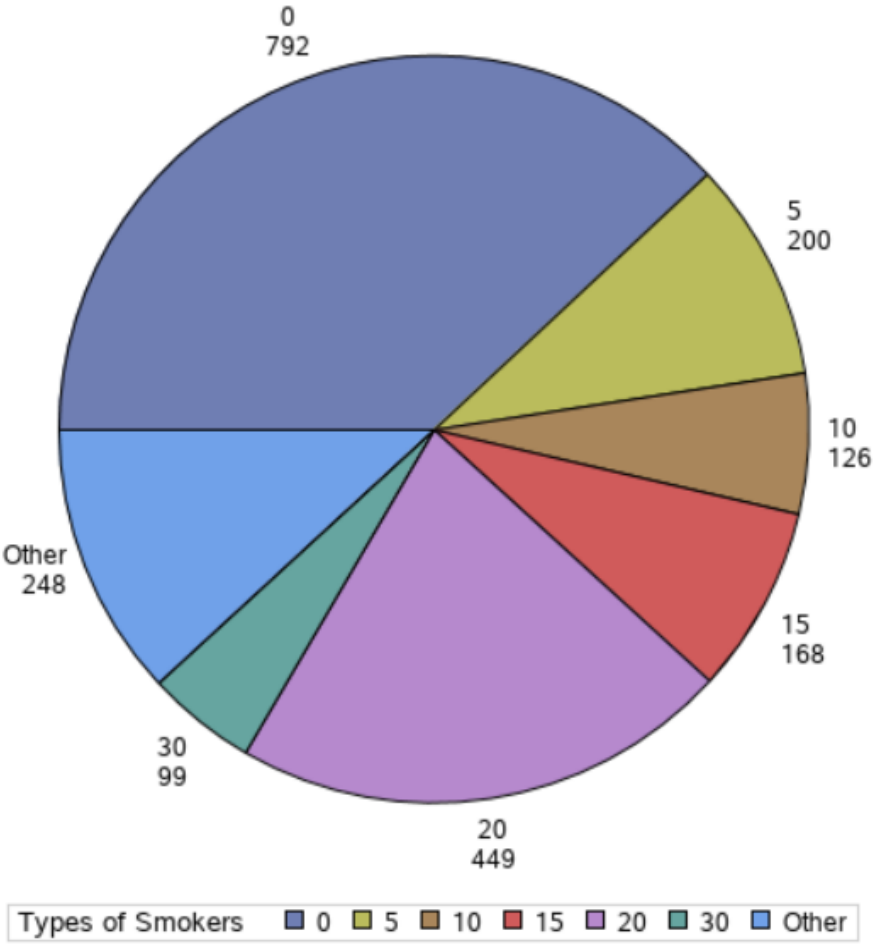
Even weights seems to be a valid factor in status determination since as the weight status increases the number of deaths also increase.



Smoking seems to play a role in determination of status since the numbers are increasing as the smoking level of the people increase.

Pie Chart for viewing the spread of smokers based on the smoking scores that represents the intensity of the smoke :

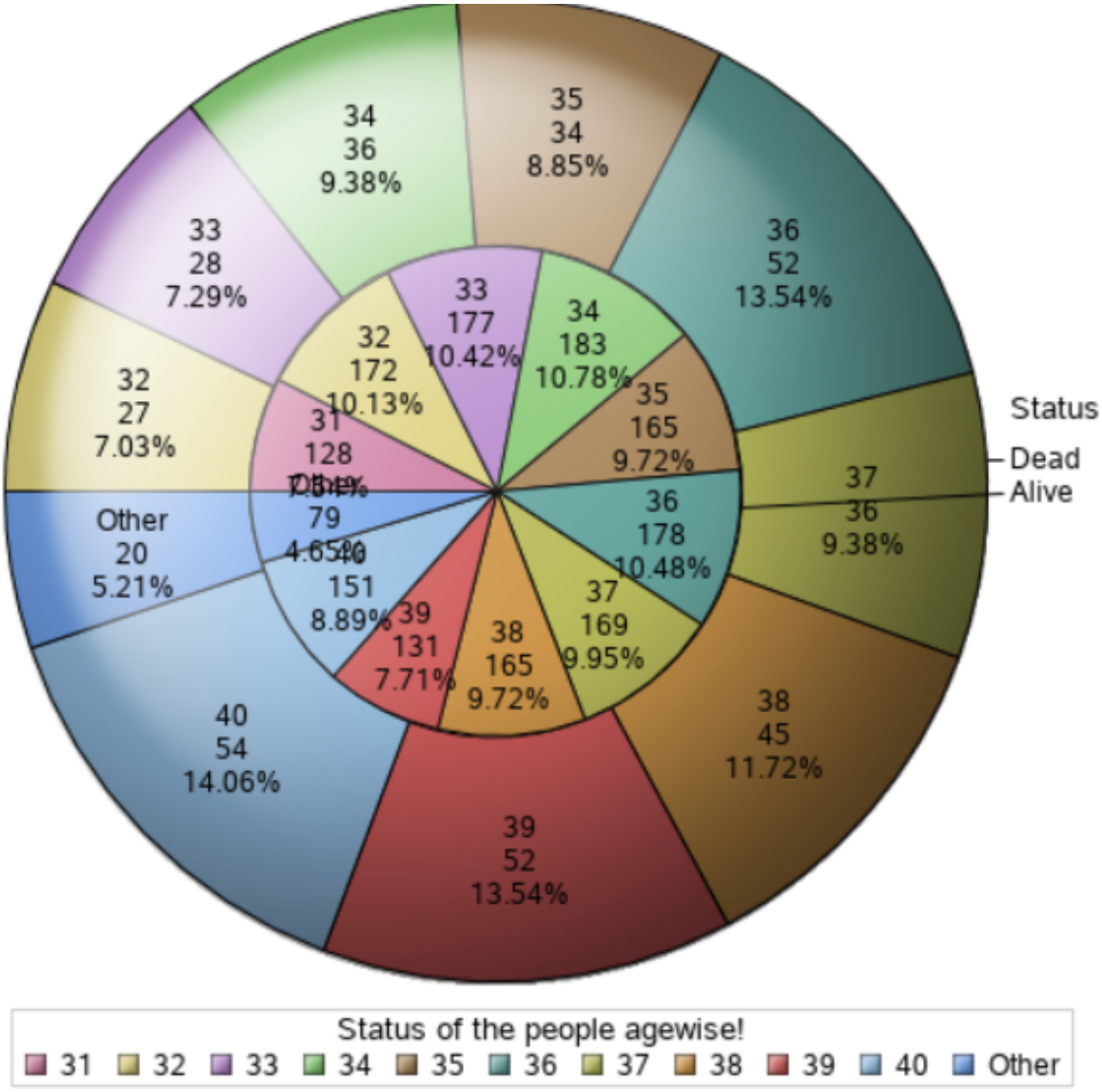
```
PROC SGRENDER DATA = heartmine TEMPLATE = pie;
RUN;
quit;
PROC TEMPLATE;
   DEFINE STATGRAPH pie;
      BEGINGRAPH;
         LAYOUT REGION;
            PIECHART CATEGORY = Smoking /
            DATALABELLOCATION = INSIDE
            DATALABELCONTENT = ALL
            CATEGORYDIRECTION = CLOCKWISE
            DATASKIN = SHEEN
            START = 180 NAME = 'pie';
            DISCRETELEGEND 'pie' /
            TITLE = 'Types of Smokers';
         ENDLAYOUT;
      ENDGRAPH;
   END;
RUN;
```



Grouped Pie Chart

In this pie chart the value of the variable presented in the graph is grouped with respect to another variable of the same data set. Each group becomes one circle and the chart has as many concentric circles as the number of groups available.

```
PROC TEMPLATE;
   DEFINE STATGRAPH pie;
      BEGINGRAPH;
         LAYOUT REGION;
            PIECHART CATEGORY = AgeAtStart / Group = Status
            DATALABELLOCATION = INSIDE
            DATALABELCONTENT = ALL
            CATEGORYDIRECTION = CLOCKWISE
            DATASKIN = SHEEN
            START = 180 NAME = 'pie';
            DISCRETELEGEND 'pie' /
            TITLE = 'Status of the people agewise!';
         ENDLAYOUT;
      ENDGRAPH;
   END;
RUN;
PROC SGRENDER DATA = heartmine
            TEMPLATE = pie;
RUN;
```

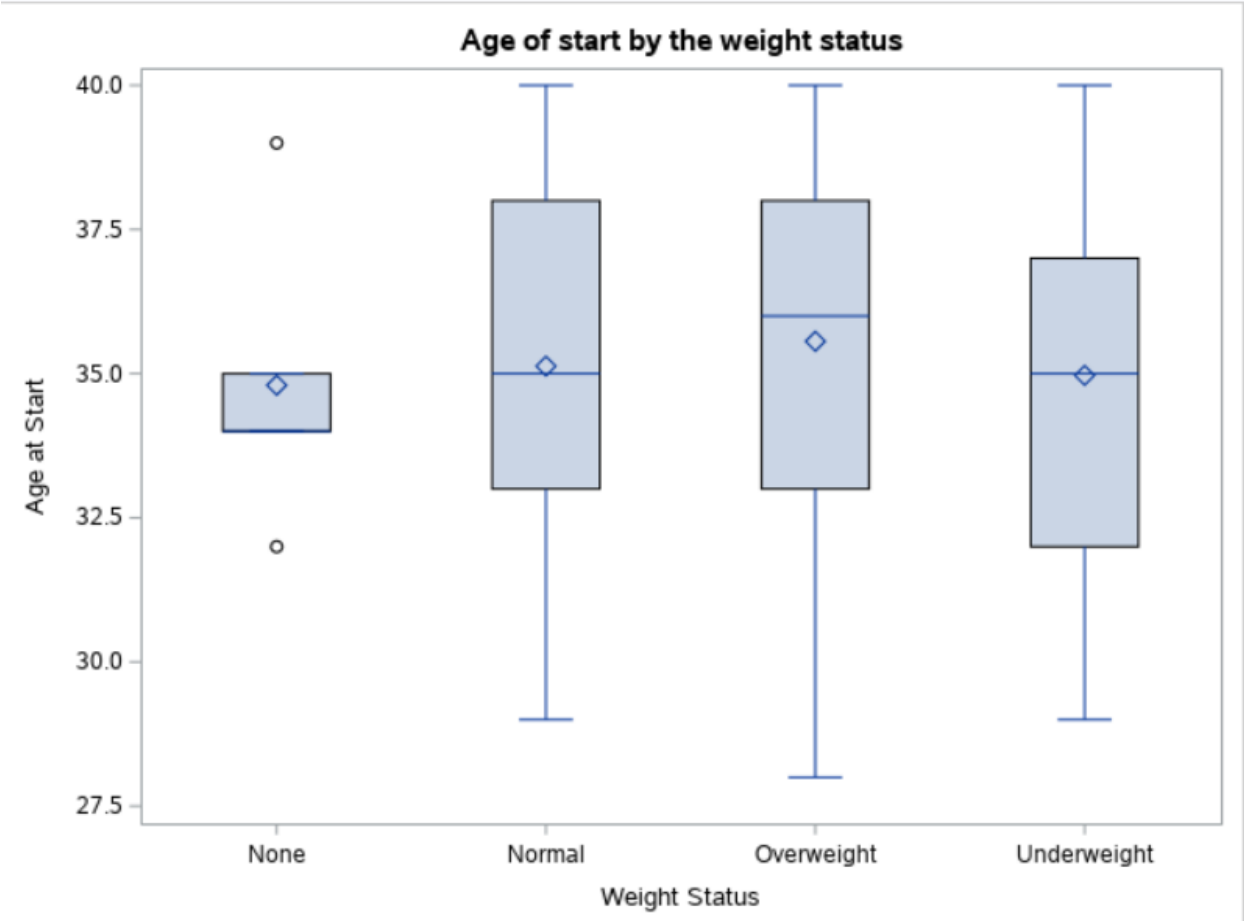


One example : The above chart shows that for people whose heart disease started at the age of 38, out of all about 9.72% people are still alive and 11.72% people are dead

Box Plots

In a simple Boxplot we choose one variable from the data set and another to form a category. The values of the first variable are categorized in as many number of groups as the number of distinct values in the second variable.

```
121 /* box plots */
122 PROC SGPLOT  DATA = work.heartmine;
123    VBOX AgeAtStart
124    / category = Weight_Status;
125
126    title 'Age of start by the weight status';
127 RUN;
```

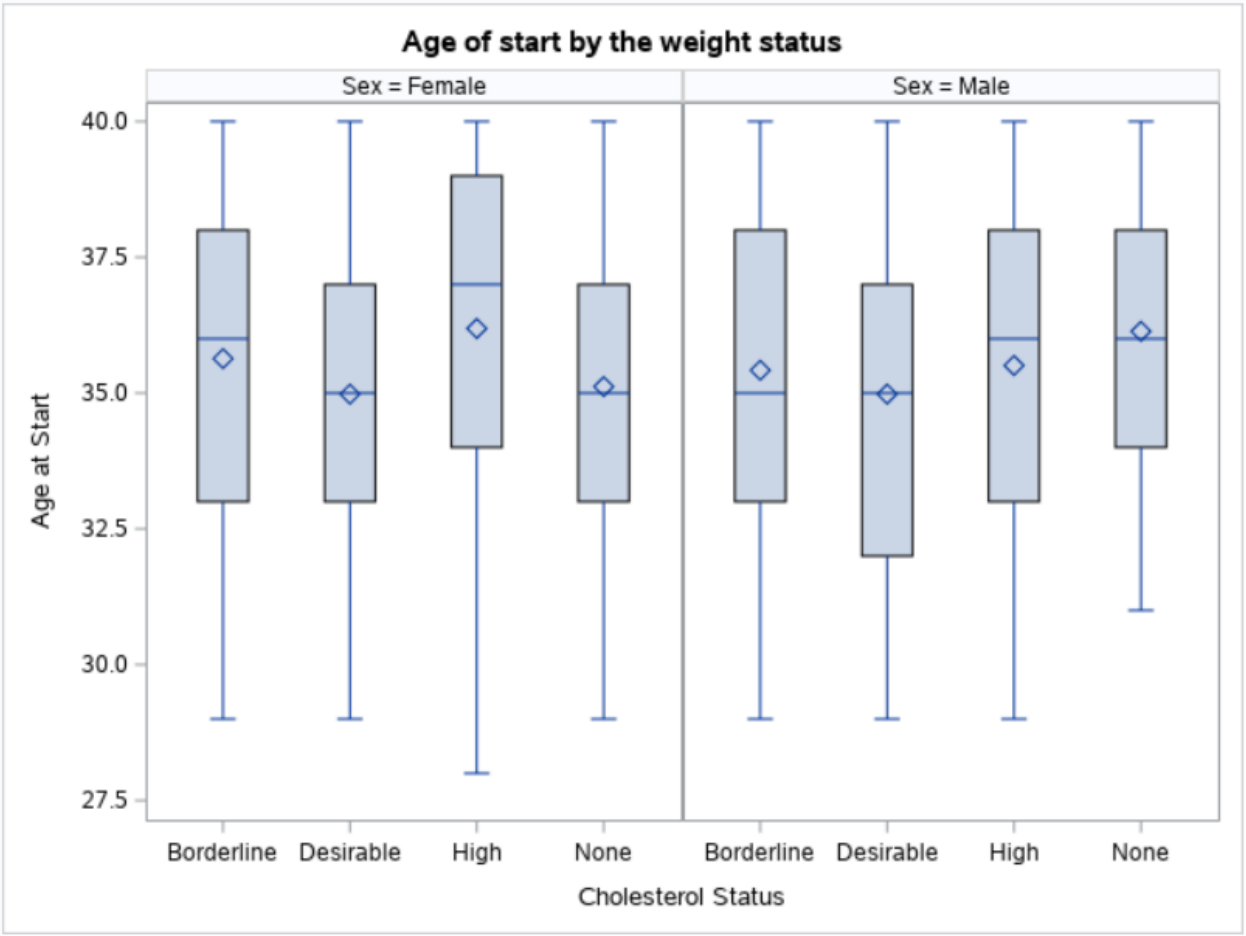


As shown above I have plotted the boxplot for the age of start of disease vs the weight status. In the none category we can observe the unknown outliers. The mean of values for all the categories of weight status more or less lies in the 35 o 37.5 year range only.

Box Plots in Vertical Panel

We can divide the Boxplots of a variable into many vertical panels(columns). Each panel holds the boxplots for all the categorical variables. But the boxplots are further grouped using another third variable which divides the graph into multiple panels.

```
129 PROC SGPANEL  DATA = work.heartmine;
130 PANELBY Sex;
131    VBOX AgeAtStart / category = Chol_Status;
132
133    title 'Age of start by the weight status';
134 RUN;
```



Here we can see that in all stages of cholesterol levels women always have a higher average age compared to men. And the population spread in case of females is always higher than men.

CONCLUSION :

After learning about SAS studio, its features, functionalities and also implementing them, I was able to conclude that :

- SAS provides us various functionalities like Data Management, Statistical Analysis, Report formation with perfect graphics etc that I have used in the above experiment.
- So in my case I loaded the already existing dataset by just making some changes in the original dataset to create my own data and then after cleaning the values, filling missing values I performed various kind of data analysis by plotting the various utility functions for plotting that SAS provides.