In [2]:
```
import numpy as np
data = np.genfromtxt("diabetes2.csv", delimiter=',', skip_header=1, dtype=int)
data
```

Out[2]:
```
array([[  6, 148,  72, ...,   0,  50,   1],
       [  1,  85,  66, ...,   0,  31,   0],
       [  8, 183,  64, ...,   0,  32,   1],
       ...,
       [  5, 121,  72, ...,   0,  30,   0],
       [  1, 126,  60, ...,   0,  47,   1],
       [  1,  93,  70, ...,   0,  23,   0]])
```

In [3]:
```
import pandas as pd      # built on top of numpy
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns    # built on top of matplotlib
from pandas.api.types import CategoricalDtype # enables specifying categorical aget
ype below
```

In [4]:
```
diabetes = pd.read_csv("diabetes2.csv") #Reading the dataset in a dataframe using P
andas
print diabetes.head(5)
print
diabetes.info()
```

```
   preg  plas  pres  skin  insu  mass   pedi  age  class
0     6   148    72    35     0  33.6  0.627   50      1
1     1    85    66    29     0  26.6  0.351   31      0
2     8   183    64     0     0  23.3  0.672   32      1
3     1    89    66    23    94  28.1  0.167   21      0
4     0   137    40    35   168  43.1  2.288   33      1

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
preg     768 non-null int64
plas     768 non-null int64
pres     768 non-null int64
skin     768 non-null int64
insu     768 non-null int64
mass     768 non-null float64
pedi     768 non-null float64
age      768 non-null int64
class    768 non-null int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

In [5]:
```
diabetes.describe()
```

Out[5]:

	preg	plas	pres	skin	insu	mass	pedi	a
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.0000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.24088
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.76023
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.00000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.00000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.00000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.00000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.00000

In [6]:
```
diabetes2 = diabetes.copy(deep=True)
# use replace as pure function:
diabetes2['plas'] = diabetes['plas'].replace(0,np.NaN)
diabetes2['pres'] = diabetes['pres'].replace(0,np.NaN)
diabetes2['skin'] = diabetes['skin'].replace(0,np.NaN)
diabetes2['insu'] = diabetes['insu'].replace(0,np.NaN)
diabetes2['mass'] = diabetes['mass'].replace(0,np.NaN)

#use replace as mutator by setting arg inplace=True
#diabetes2['pedi'].replace(0,np.NaN, inplace=True)
#diabetes.describe()
diabetes2.describe()
```

Out[6]:

	preg	plas	pres	skin	insu	mass	pedi	a
count	768.000000	763.000000	733.000000	541.000000	394.000000	757.000000	768.000000	768.0000
mean	3.845052	121.686763	72.405184	29.153420	155.548223	32.457464	0.471876	33.24088
std	3.369578	30.535641	12.382158	10.476982	118.775855	6.924988	0.331329	11.76023
min	0.000000	44.000000	24.000000	7.000000	14.000000	18.200000	0.078000	21.00000
25%	1.000000	99.000000	64.000000	22.000000	76.250000	27.500000	0.243750	24.00000
50%	3.000000	117.000000	72.000000	29.000000	125.000000	32.300000	0.372500	29.00000
75%	6.000000	141.000000	80.000000	36.000000	190.000000	36.600000	0.626250	41.00000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.00000

In [7]:
```
diabetes_group = diabetes2.groupby(['class'])
diabetes_group.describe()
```

Out[7]:

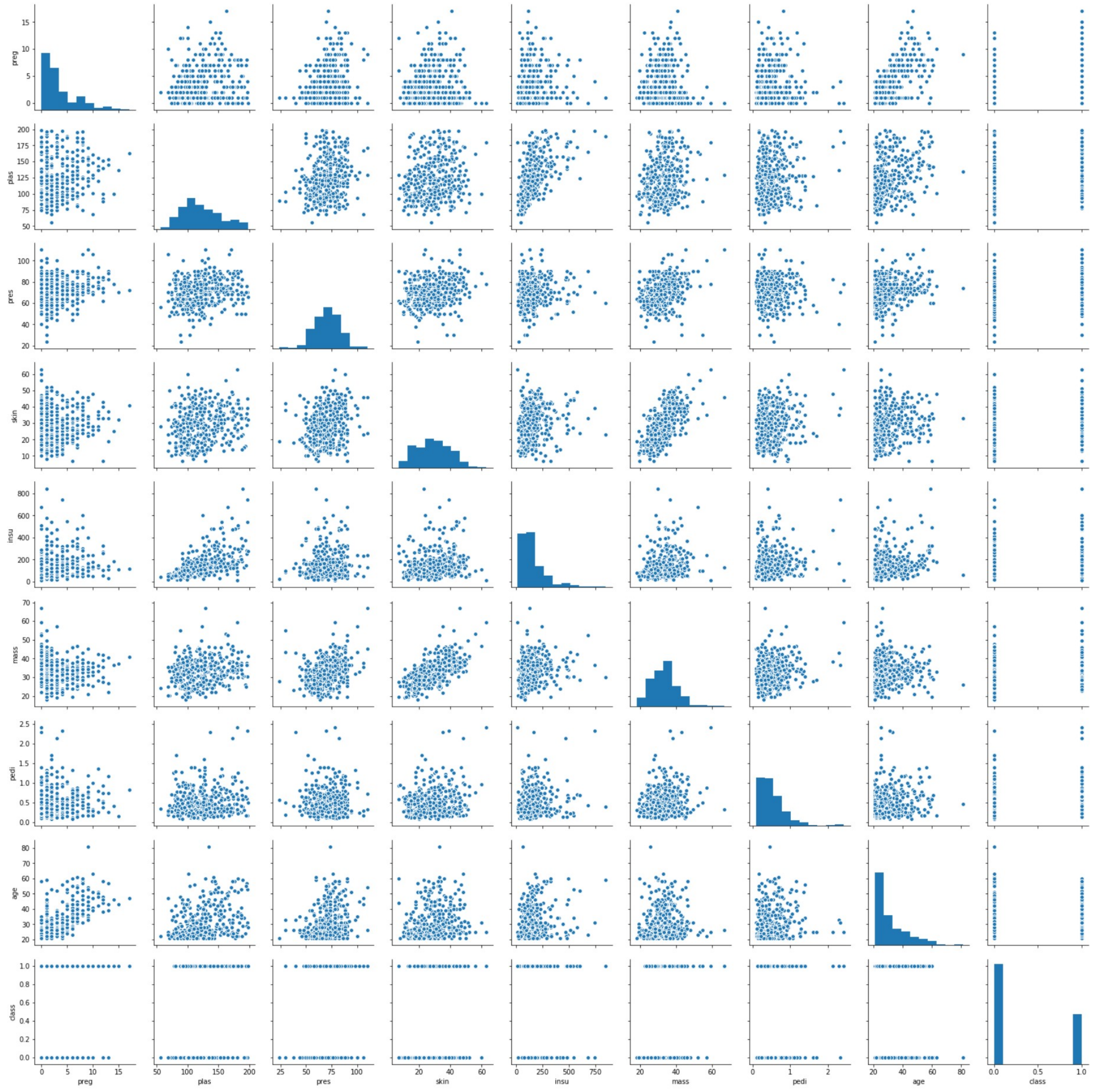
	age								insu		...	pres		skin
	count	mean	std	min	25%	50%	75%	max	count	mean	...	75%	max	cour
class														
0	500.0	31.190000	11.667655	21.0	23.0	27.0	37.0	81.0	264.0	130.287879	...	78.0	122.0	361.(
1	268.0	37.067164	10.968254	21.0	28.0	36.0	44.0	70.0	130.0	206.846154	...	84.0	114.0	180.(

2 rows × 64 columns

In [8]:
```
%matplotlib inline
```

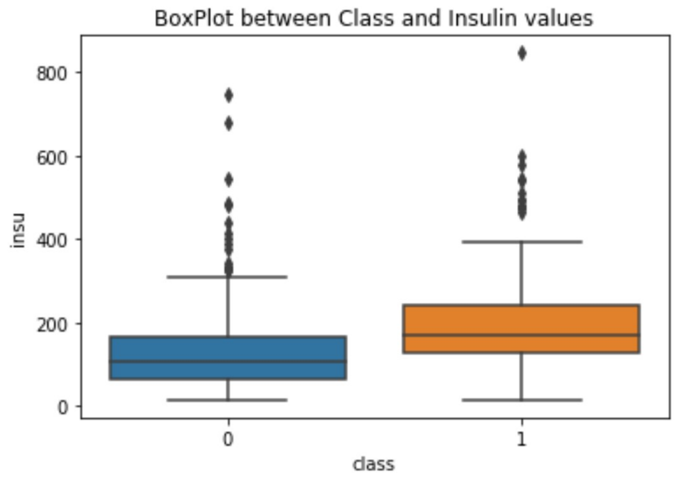
```
In [9]: #sns.pairplot(df, dropna=True) #Error if there are NAs in the data
        plot_diabetes = diabetes2.dropna()
        sns.pairplot(plot_diabetes) ### To look at the correlation between each variables f
        rom the plots
                                   ###we can see insu and pres being correlated also age an
        d pressure correlated

Out[9]: <seaborn.axisgrid.PairGrid at 0x81fa400>
```



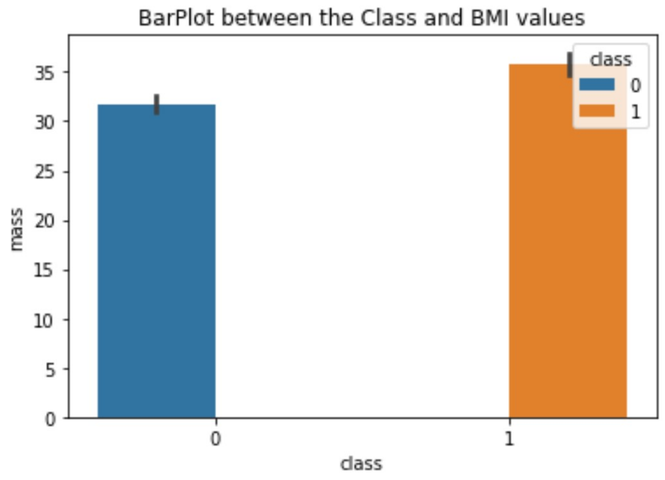
In [10]:
```
boxplot = sns.boxplot(x="class", y="insu", data=plot_diabetes) #This shows the plas
ma values for people with diabetes(Class 1) is greater than people without diabetes
(Class 2)
plt.title('BoxPlot between Class and Insulin values')
```

Out[10]:
```
<matplotlib.text.Text at 0x16241e80>
```



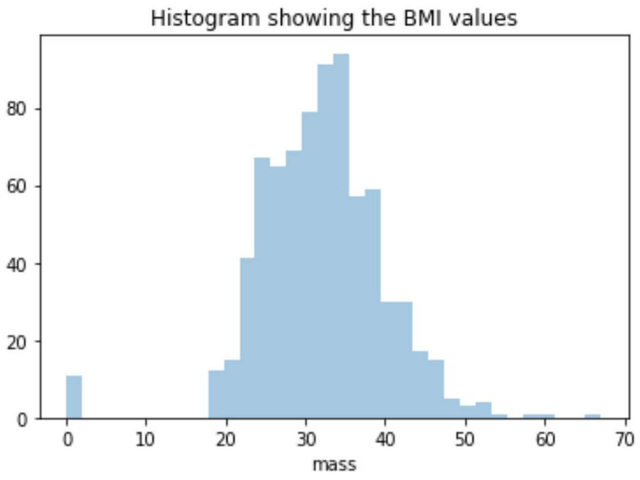
In [11]:
```
sns.barplot(x="class", y="mass", hue="class", data=plot_diabetes)### This shows the
BMI for the people with diabetes(class 1) is
plt.title("BarPlot between the Class and BMI values")    ### greater than people wi
thout diabetes(class 0)
```

Out[11]:
```
<matplotlib.text.Text at 0x16ad7978>
```



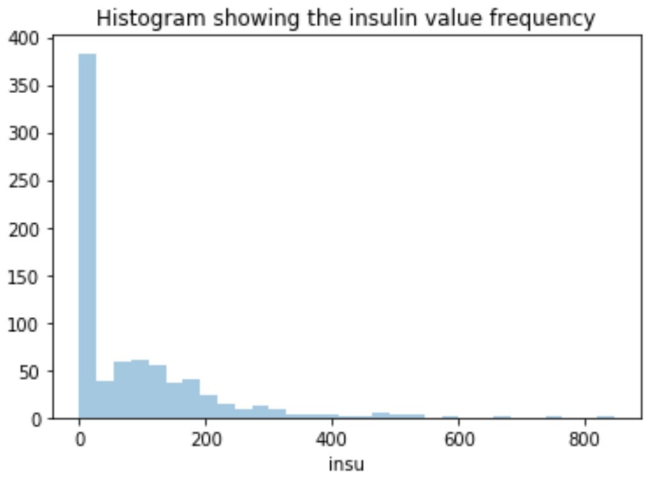
In [12]:
```
sns.distplot(diabetes.mass,kde=False) ### This shows the mean BMI will be close to
30
plt.title("Histogram showing the BMI values")
```

Out[12]:
```
<matplotlib.text.Text at 0xdcaeb00>
```



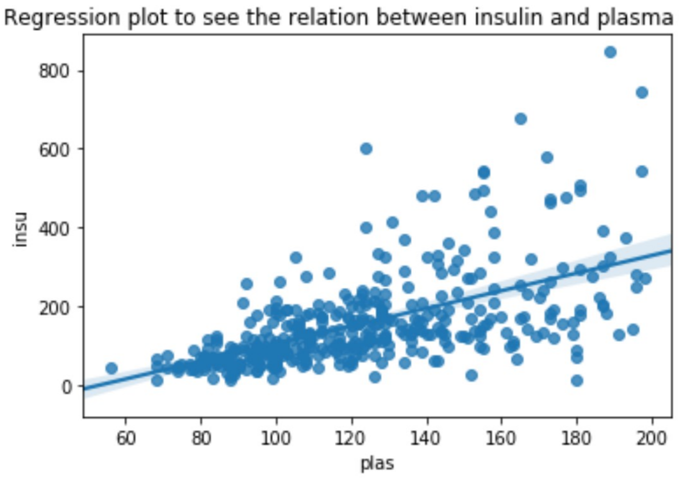
In [13]:
```
sns.distplot(diabetes.insu,kde=False) ### This shows there are many records with in
sulin values as 0
plt.title("Histogram showing the insulin value frequency")
```

Out[13]:
```
<matplotlib.text.Text at 0x17103400>
```



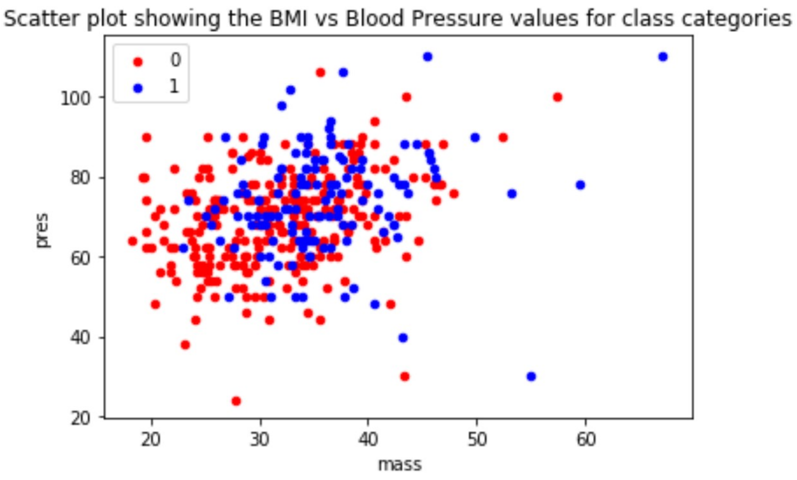
```
In [14]: sns.regplot(x="plas", y="insu", data=plot_diabetes)  ########################### Th
         is shows the insu and plas levels are highly correlated
         #sns.regplot(x="pres", y="mass",data=plot_diabetes)  ###########################
         plt.title('Regression plot to see the relation between insulin and plasma values')

Out[14]: <matplotlib.text.Text at 0x1740e5c0>
```



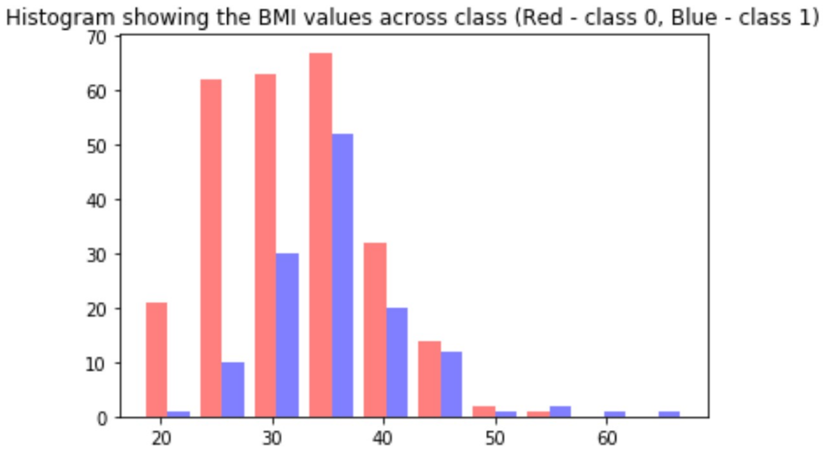
```
In [15]: fig, ax = plt.subplots()
         colors =   {0:'red', 1:'blue'}
         grouped = plot_diabetes.groupby('class')
         for key, group in grouped:
             group.plot(ax=ax, kind='scatter', x='mass', y='pres', label=key, color=colors[k
         ey])
         plt.title("Scatter plot showing the BMI vs Blood Pressure values for class categori
         es")### This shows that clearly most of class 0 are on the lower end of the spectru
         m for plas and insu levels.

Out[15]: <matplotlib.text.Text at 0x175d5400>
```



```
In [16]: plt.hist([plot_diabetes['mass'][plot_diabetes['class']==0],plot_diabetes['mass'][pl
         ot_diabetes['class']==1]], color=['r','b'], alpha=0.5)
         plt.title('Histogram showing the BMI values across class (Red - class 0, Blue - cla
         ss 1)')

Out[16]: <matplotlib.text.Text at 0x18ecfe10>
```


Histogram showing the BMI values across class (Red - class 0, Blue - class 1)