Project Name: Healthcare PGP

Project Task: Week 1

In [ ]:
```
'''
Data Exploration:
1. Perform descriptive analysis. Understand the variables and their corresponding va
a value of zero does not make sense and thus indicates missing value:
• Glucose
• BloodPressure
• SkinThickness
• Insulin
• BMI

2. Visually explore these variables using histograms. Treat the missing values accor

3. There are integer and float data type variables in this dataset. Create a count (
the data types and the count of variables.
'''
```

In [2]:
```
import numpy as np
import pandas as pd


import matplotlib.pyplot as plt
from matplotlib import style
import seaborn as sns

%matplotlib inline
```

In [3]:
```
df=pd.read_csv(r'C:\Users\Anuj Bhalla\Downloads\Project_2\Project 2\Healthcare - Dia
df.head()
```

Out[3]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	6	148	72	35	0	33.6	0.627	5(
1	1	85	66	29	0	26.6	0.351	31
2	8	183	64	0	0	23.3	0.672	32
3	1	89	66	23	94	28.1	0.167	21
4	0	137	40	35	168	43.1	2.288	33

Descriptive Analysis

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
```

```
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

In [5]:
```
df.isnull().any()
```

Out[5]:
```
Pregnancies                 False
Glucose                     False
BloodPressure               False
SkinThickness               False
Insulin                     False
BMI                         False
DiabetesPedigreeFunction    False
Age                         False
Outcome                     False
dtype: bool
```

In [6]:
```
df.describe()
```

Out[6]:

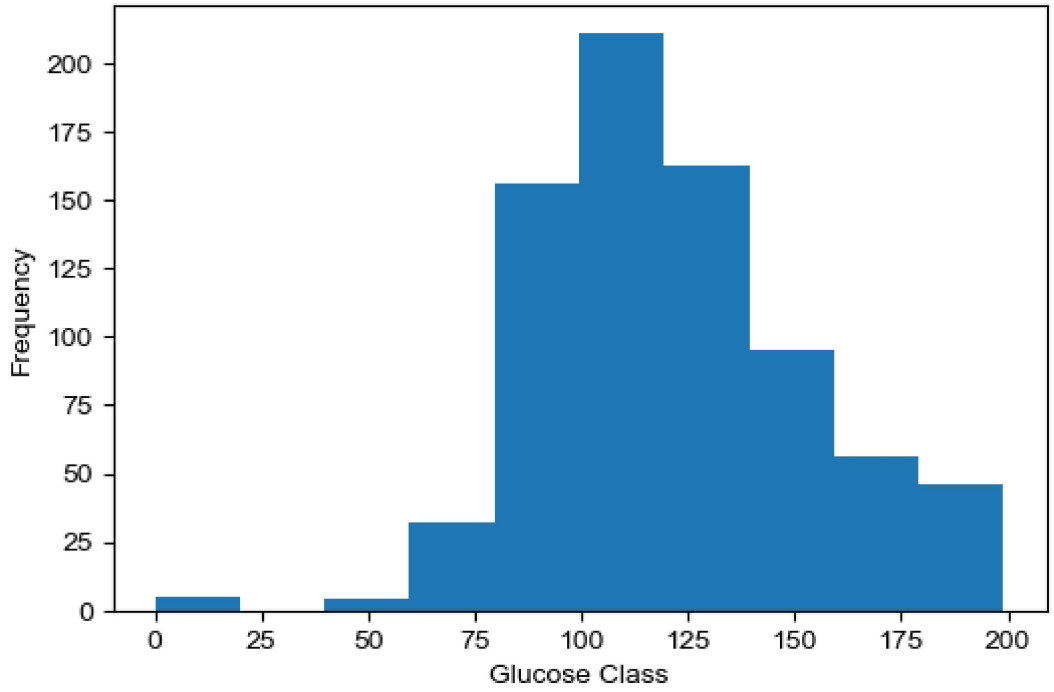
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedig
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	

In [67]:
```
### As per the above results, it is observed that the Zero value for Glucose, Blood
## make sense in this dataset and considered as missing values.
```

Visually explore these variables using histograms. Treat the missing values accordingly.

In [7]:
```
plt.figure(figsize=(6,4),dpi=100)
plt.xlabel('Glucose Class')
df['Glucose'].plot.hist()
sns.set_style(style='darkgrid')
print("Mean of Glucose level is :-", df['Glucose'].mean())
print("Datatype of Glucose Variable is:",df['Glucose'].dtypes)
```

```
Mean of Glucose level is :- 120.89453125
Datatype of Glucose Variable is: int64
```

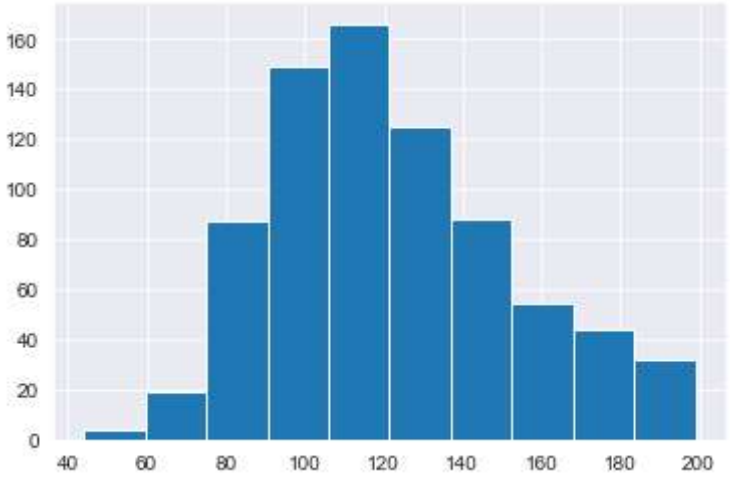


I am treating missing values which is basically 0 by mean of Glucose level. This is because we can see from histogram most of observation have Glucose level between 100 and 120.

In [8]:
```
df['Glucose']=df['Glucose'].replace(0,df['Glucose'].mean())
```

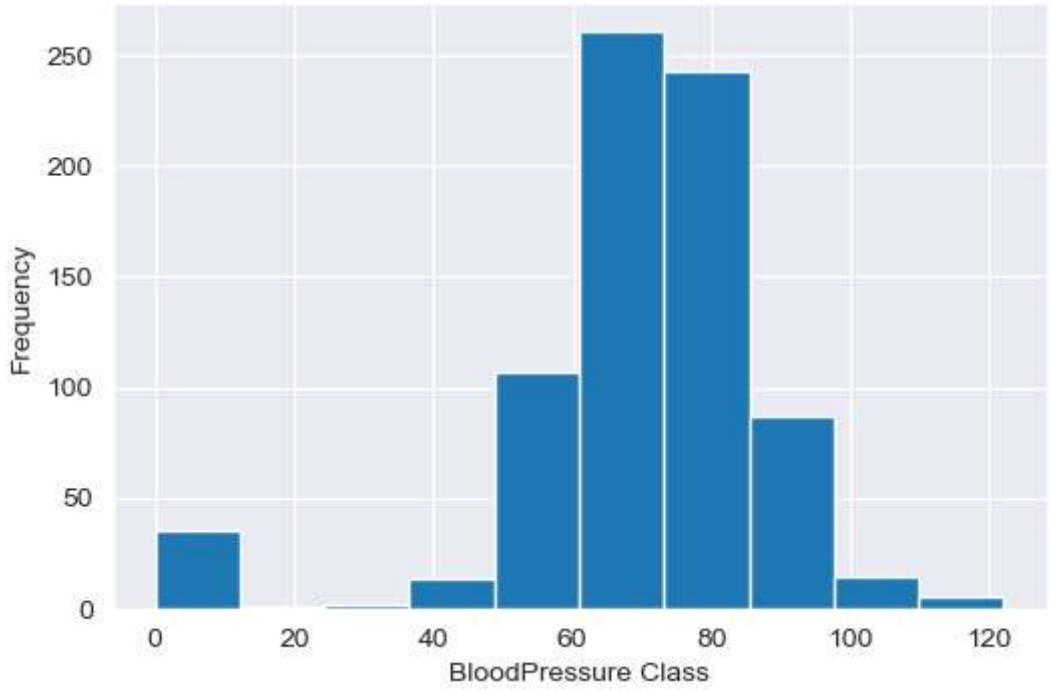
In [80]:
```
plt.hist(df['Glucose'])
```

Out[80]:
```
(array([  4.,  19.,  87., 149., 166., 125.,  88.,  54.,  44.,  32.]),
 array([ 44. ,  59.5,  75. ,  90.5, 106. , 121.5, 137. , 152.5, 168. ,
        183.5, 199. ]),
 <BarContainer object of 10 artists>)
```



In [10]:
```
plt.figure(figsize=(6,4),dpi=100)
plt.xlabel('BloodPressure Class')
df['BloodPressure'].plot.hist()
sns.set_style(style='darkgrid')
print("Mean of BloodPressure level is :-", df['BloodPressure'].mean())
print("Datatype of BloodPressure Variable is:",df['BloodPressure'].dtypes)
```

```
Mean of BloodPressure level is :- 69.10546875
Datatype of BloodPressure Variable is: int64
```

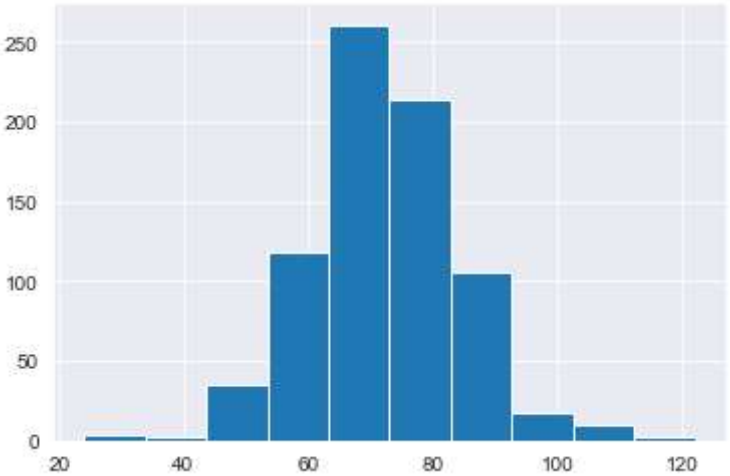


I am treating missing values which is basically 0 by mean of BloodPressure level. This is because we can see from histogram most of observation have BP level between 70 and 80.

In [11]:
```
df['BloodPressure']=df['BloodPressure'].replace(0,df['BloodPressure'].mean())
```

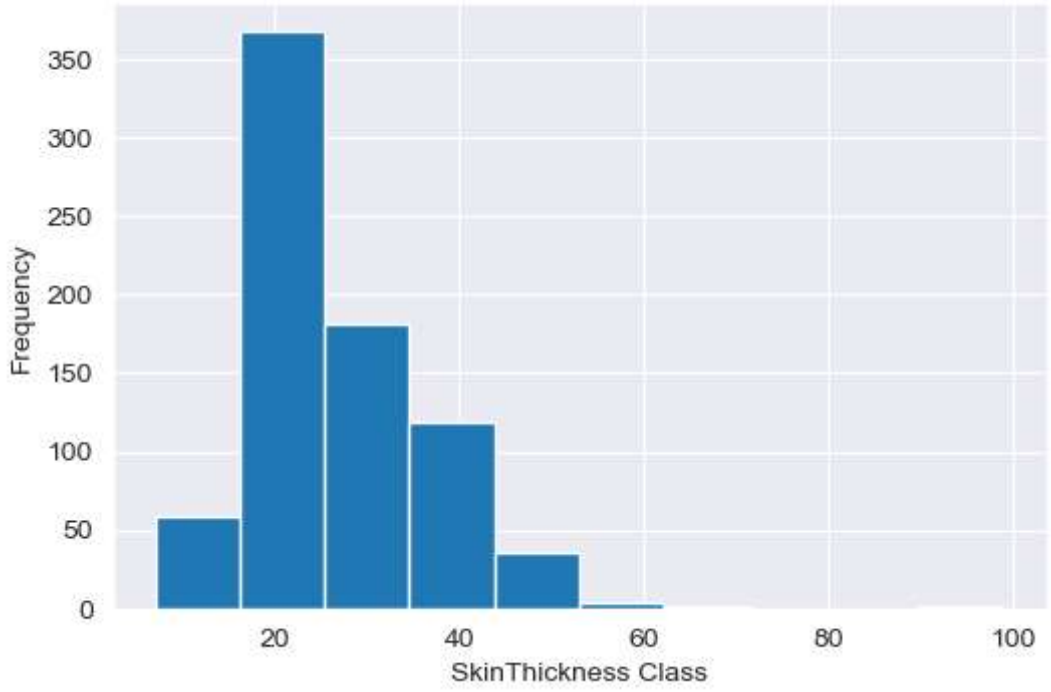
In [81]:
```
plt.hist(df['BloodPressure'])
```

Out[81]:
```
(array([  3.,   2.,  35., 118., 261., 214., 105.,  18.,  10.,   2.]),
 array([ 24. ,  33.8,  43.6,  53.4,  63.2,  73. ,  82.8,  92.6, 102.4,
        112.2, 122. ]),
 <BarContainer object of 10 artists>)
```



In [79]:
```
plt.figure(figsize=(6,4),dpi=100)
plt.xlabel('SkinThickness Class')
df['SkinThickness'].plot.hist()
sns.set_style(style='darkgrid')
print("Mean of SkinThickness is :-", df['SkinThickness'].mean())
print("Datatype of SkinThickness Variable is:",df['SkinThickness'].dtypes)
```

```
Mean of SkinThickness is :- 26.606479220920118
Datatype of SkinThickness Variable is: float64
```

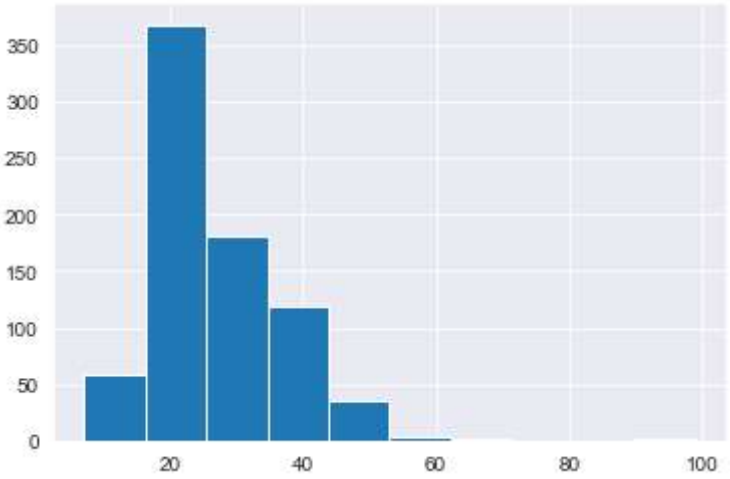


I am treating missing values which is basically 0 by mean of SkinThickness. This is because we can see from histogram most of observation have SkinThickness between 20 and 30.

In [14]:
```
df['SkinThickness']=df['SkinThickness'].replace(0,df['SkinThickness'].mean())
```

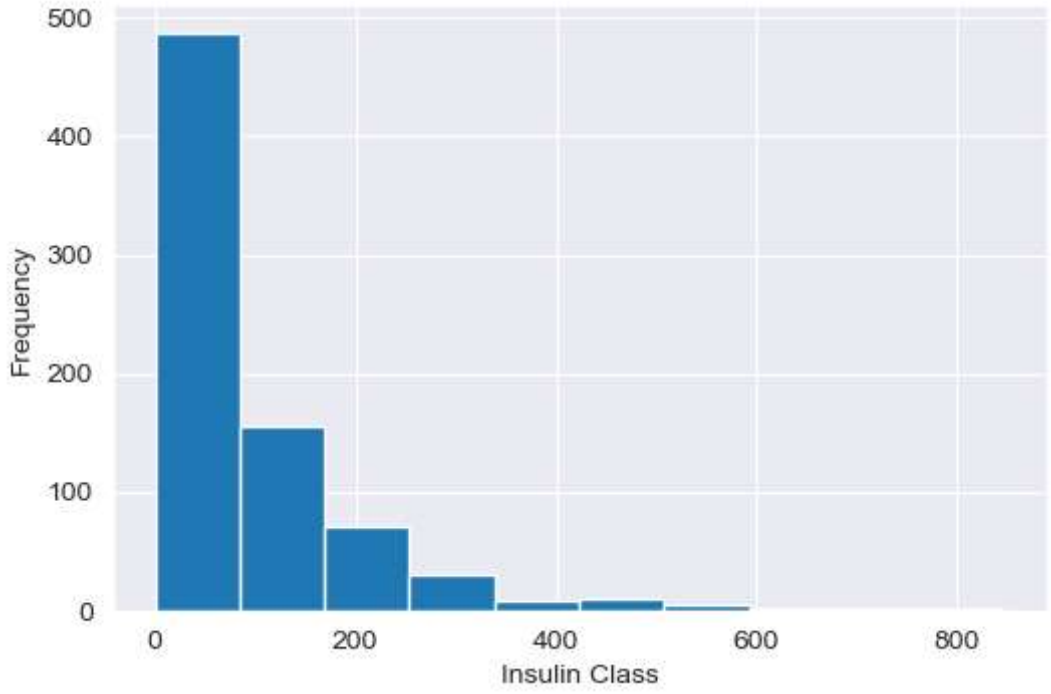
In [82]:
```
plt.hist(df['SkinThickness'])
```

Out[82]:
```
(array([ 59., 368., 181., 118.,  36.,   4.,   1.,   0.,   0.,   1.]),
 array([ 7. , 16.2, 25.4, 34.6, 43.8, 53. , 62.2, 71.4, 80.6, 89.8, 99. ]),
 <BarContainer object of 10 artists>)
```



In [16]:
```
plt.figure(figsize=(6,4),dpi=100)
plt.xlabel('Insulin Class')
df['Insulin'].plot.hist()
sns.set_style(style='darkgrid')
print("Mean of Insulin is :-", df['Insulin'].mean())
print("Datatype of Insulin Variable is:",df['Insulin'].dtypes)
```

```
Mean of Insulin is :- 79.79947916666667
Datatype of Insulin Variable is: int64
```

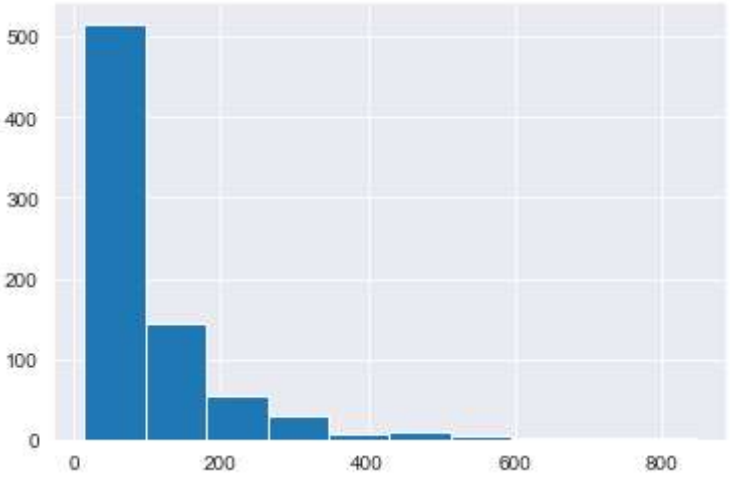


I am treating missing values which is basically 0 by mean of Insulin.

In [17]:
```
df['Insulin']=df['Insulin'].replace(0,df['Insulin'].mean())
```

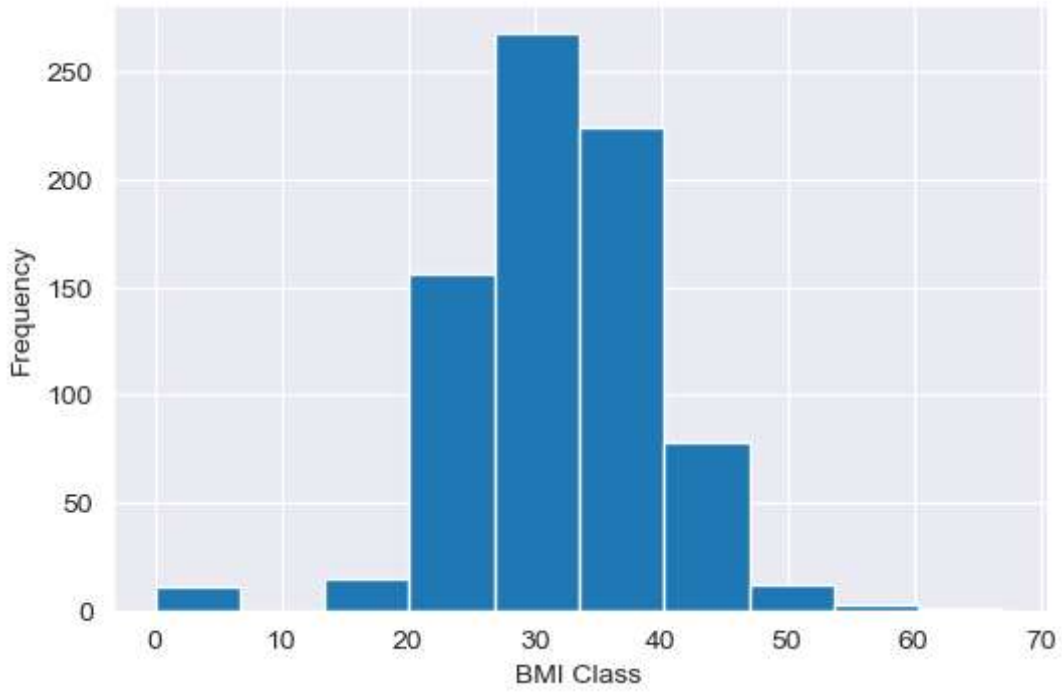
In [83]:
```
plt.hist(df['Insulin'])
```

Out[83]:
```
(array([516., 143.,  55.,  29.,   7.,  10.,   4.,   1.,   2.,   1.]),
 array([ 14. ,  97.2, 180.4, 263.6, 346.8, 430. , 513.2, 596.4, 679.6,
        762.8, 846. ]),
 <BarContainer object of 10 artists>)
```



In [19]:
```
plt.figure(figsize=(6,4),dpi=100)
plt.xlabel('BMI Class')
df['BMI'].plot.hist()
sns.set_style(style='darkgrid')
print("Mean of BMI is :-", df['BMI'].mean())
print("Datatype of BMI Variable is:",df['BMI'].dtypes)
```

```
Mean of BMI is :- 31.992578124999977
Datatype of BMI Variable is: float64
```

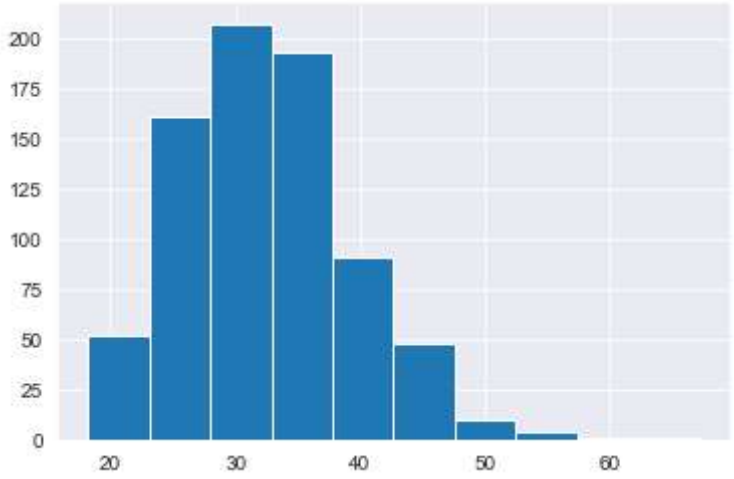


I am treating missing values which is basically 0 by mean of BMI

In [20]:
```
df['BMI']=df['BMI'].replace(0,df['BMI'].mean())
```

In [84]:
```
plt.hist(df['BMI'])
```

Out[84]:
```
(array([ 52., 161., 207., 193.,  91.,  48.,  10.,   4.,   1.,   1.]),
 array([18.2 , 23.09, 27.98, 32.87, 37.76, 42.65, 47.54, 52.43, 57.32,
        62.21, 67.1 ]),
 <BarContainer object of 10 artists>)
```



In [22]:
```
df.describe()
```

Out[22]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedig
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	
mean	3.845052	121.681605	72.254807	26.606479	118.660163	32.450805	
std	3.369578	30.436016	12.115932	9.631241	93.080358	6.875374	
min	0.000000	44.000000	24.000000	7.000000	14.000000	18.200000	
25%	1.000000	99.750000	64.000000	20.536458	79.799479	27.500000	
50%	3.000000	117.000000	72.000000	23.000000	79.799479	32.000000	

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedig
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	

There are integer and float data type variables in this dataset. Create a count (frequency) plot describing the data types and the count of variables.

In [23]:
```
freq=pd.DataFrame(df.apply(lambda x: x.value_counts()).T.stack(), columns=["Count"])
freq
```

Out[23]:

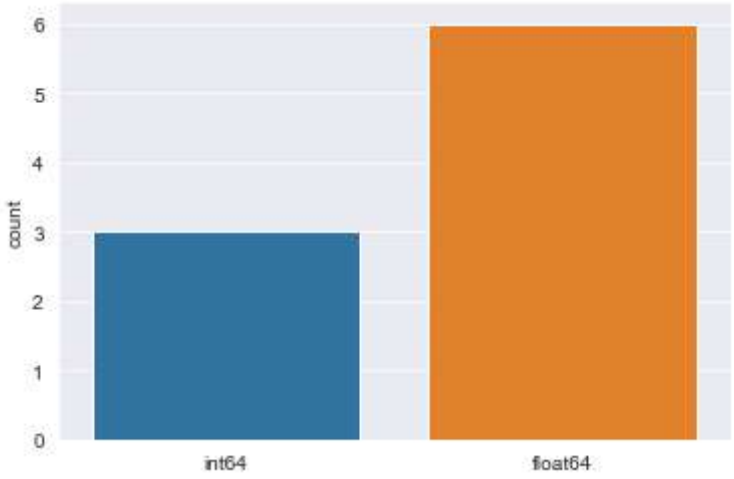
		Count
Pregnancies	**0.0**	111.0
	1.0	135.0
	2.0	103.0
	3.0	75.0
	4.0	68.0
...	**...**	...
Age	**70.0**	1.0
	72.0	1.0
	81.0	1.0
Outcome	**0.0**	500.0
	1.0	268.0

1256 rows × 1 columns

In [24]:
```
import matplotlib.pyplot as plt
import seaborn as sns

sns.countplot(df.dtypes.map(str))
plt.show()
```

```
C:\Users\Anuj Bhalla\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWa
rning: Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an exp
licit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [ ]: