In [1]:

```
#import all packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
import seaborn as sns
```

In [2]:

```
#Importing the medical data file
df = pd.read_csv(r"C:\Users\arjun\OneDrive\Desktop\WGU\D206\medical_raw_data.csv")
```

In [3]:

```
#data profiling to see if data was imported correctly
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 53 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Unnamed: 0          10000 non-null  int64
 1   CaseOrder           10000 non-null  int64
 2   Customer_id         10000 non-null  object
 3   Interaction         10000 non-null  object
 4   UID                 10000 non-null  object
 5   City                10000 non-null  object
 6   State               10000 non-null  object
 7   County              10000 non-null  object
 8   Zip                 10000 non-null  int64
 9   Lat                 10000 non-null  float64
 10  Lng                 10000 non-null  float64
 11  Population          10000 non-null  int64
 12  Area                10000 non-null  object
 13  Timezone            10000 non-null  object
 14  Job                 10000 non-null  object
 15  Children            7412 non-null   float64
 16  Age                 7586 non-null   float64
 17  Education           10000 non-null  object
 18  Employment          10000 non-null  object
 19  Income              7536 non-null   float64
 20  Marital             10000 non-null  object
 21  Gender              10000 non-null  object
 22  ReAdmis             10000 non-null  object
 23  VitD_levels         10000 non-null  float64
 24  Doc_visits          10000 non-null  int64
 25  Full_meals_eaten    10000 non-null  int64
 26  VitD_supp           10000 non-null  int64
 27  Soft_drink          7533 non-null   object
 28  Initial_admin       10000 non-null  object
 29  HighBlood           10000 non-null  object
 30  Stroke              10000 non-null  object
 31  Complication_risk   10000 non-null  object
 32  Overweight          9018 non-null   float64
 33  Arthritis           10000 non-null  object
 34  Diabetes            10000 non-null  object
 35  Hyperlipidemia      10000 non-null  object
 36  BackPain            10000 non-null  object
 37  Anxiety             9016 non-null   float64
 38  Allergic_rhinitis   10000 non-null  object
 39  Reflux_esophagitis  10000 non-null  object
 40  Asthma              10000 non-null  object
 41  Services            10000 non-null  object
 42  Initial_days        8944 non-null   float64
 43  TotalCharge         10000 non-null  float64
 44  Additional_charges  10000 non-null  float64
 45  Item1               10000 non-null  int64
 46  Item2               10000 non-null  int64
 47  Item3               10000 non-null  int64
 48  Item4               10000 non-null  int64
 49  Item5               10000 non-null  int64
 50  Item6               10000 non-null  int64
 51  Item7                10000 non-null  int64
 52  Item8                10000 non-null  int64
dtypes: float64(11), int64(15), object(27)
memory usage: 4.0+ MB
```

In [4]:

```
#steps to removing duplicates
# 1 Check to see how many rows are duplicated. I will check using the Customer_id,
# 2 Check to see how many are duplicated for each
# 3 Removal of the duplicates from the dataset
```

In [5]:

```
# 1 Check to see how many rows are duplicated. I will check using the CaseOrder, Cu
#Duplicates in CaseOrder
df.CaseOrder.duplicated()
```

Out[5]:

```
0       False
1       False
2       False
3       False
4       False
        ...  
9995    False
9996    False
9997    False
9998    False
9999    False
Name: CaseOrder, Length: 10000, dtype: bool
```

In [6]:

```
#Duplicates in Customer_id
df.Customer_id.duplicated()
```

Out[6]:

```
0       False
1       False
2       False
3       False
4       False
        ...  
9995    False
9996    False
9997    False
9998    False
9999    False
Name: Customer_id, Length: 10000, dtype: bool
```

In [7]:

```
#Duplicates in Interaction
df.Interaction.duplicated()
```

```
Out[7]: 0       False
        1       False
        2       False
        3       False
        4       False
                ...
        9995    False
        9996    False
        9997    False
        9998    False
        9999    False
        Name: Interaction, Length: 10000, dtype: bool
```

```
In [8]: #Duplicates in UID
        df.UID.duplicated()
```

```
Out[8]: 0       False
        1       False
        2       False
        3       False
        4       False
                ...
        9995    False
        9996    False
        9997    False
        9998    False
        9999    False
        Name: UID, Length: 10000, dtype: bool
```

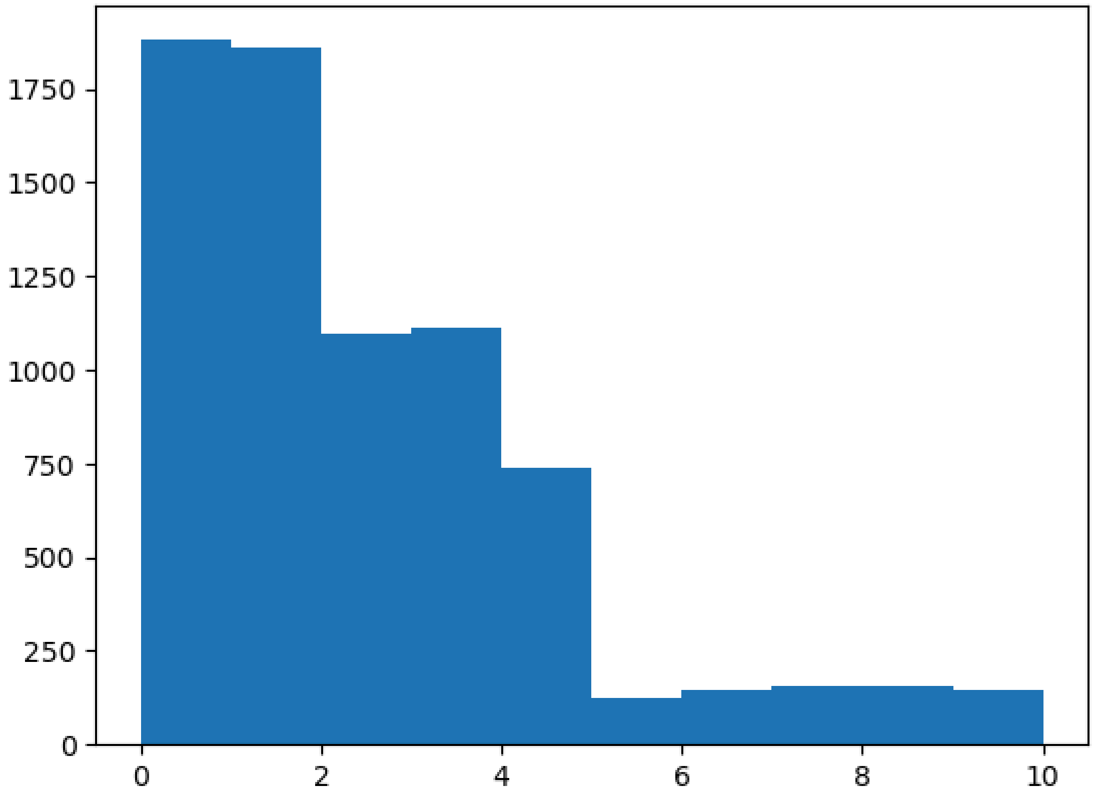
```
In [9]: # Steps to clean for missing data
        # 1 Detect which columns have missing data
        # 2 impute missing data
        # 3 Verify if the missing data has been corrected.
```

```
In [10]: #1 Detect which columns have missing data
         df.isnull().sum()
```

Out[10]:

```
Unnamed: 0               0
CaseOrder                0
Customer_id              0
Interaction              0
UID                      0
City                     0
State                    0
County                   0
Zip                      0
Lat                      0
Lng                      0
Population               0
Area                     0
Timezone                 0
Job                      0
Children              2588
Age                   2414
Education                0
Employment               0
Income                2464
Marital                  0
Gender                   0
ReAdmis                  0
VitD_levels              0
Doc_visits               0
Full_meals_eaten         0
VitD_supp                0
Soft_drink            2467
Initial_admin            0
HighBlood                0
Stroke                   0
Complication_risk        0
Overweight             982
Arthritis                0
Diabetes                 0
Hyperlipidemia           0
BackPain                 0
Anxiety                984
Allergic_rhinitis        0
Reflux_esophagitis       0
Asthma                   0
Services                 0
Initial_days          1056
TotalCharge              0
Additional_charges       0
Item1                    0
Item2                    0
Item3                    0
Item4                    0
Item5                    0
Item6                    0
Item7                    0
Item8                    0
dtype: int64
```

In [11]:
```
#impute data into Children variable
plt.hist(df['Children'])
plt.show()
```



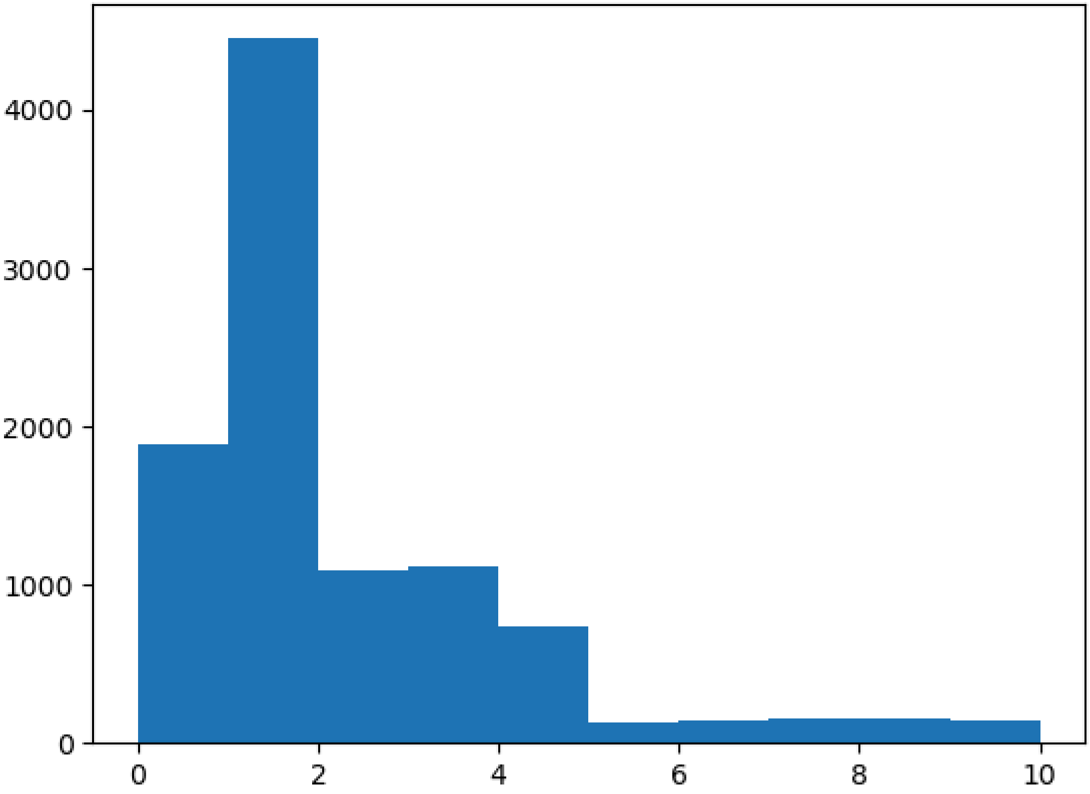
In [12]:
```
#impute using median
df['Children'].fillna(df['Children'].median(), inplace= True)
```

In [13]:
```
df.isnull().sum()
```

```
Out[13]: Unnamed: 0               0
         CaseOrder                0
         Customer_id              0
         Interaction              0
         UID                      0
         City                     0
         State                    0
         County                   0
         Zip                      0
         Lat                      0
         Lng                      0
         Population               0
         Area                     0
         Timezone                 0
         Job                      0
         Children                 0
         Age                   2414
         Education                0
         Employment               0
         Income                2464
         Marital                  0
         Gender                   0
         ReAdmis                  0
         VitD_levels              0
         Doc_visits               0
         Full_meals_eaten         0
         VitD_supp                0
         Soft_drink            2467
         Initial_admin            0
         HighBlood                0
         Stroke                   0
         Complication_risk        0
         Overweight             982
         Arthritis                0
         Diabetes                 0
         Hyperlipidemia           0
         BackPain                 0
         Anxiety                984
         Allergic_rhinitis        0
         Reflux_esophagitis       0
         Asthma                   0
         Services                 0
         Initial_days          1056
         TotalCharge              0
         Additional_charges       0
         Item1                    0
         Item2                    0
         Item3                    0
         Item4                    0
         Item5                    0
         Item6                    0
         Item7                    0
         Item8                    0
         dtype: int64
```

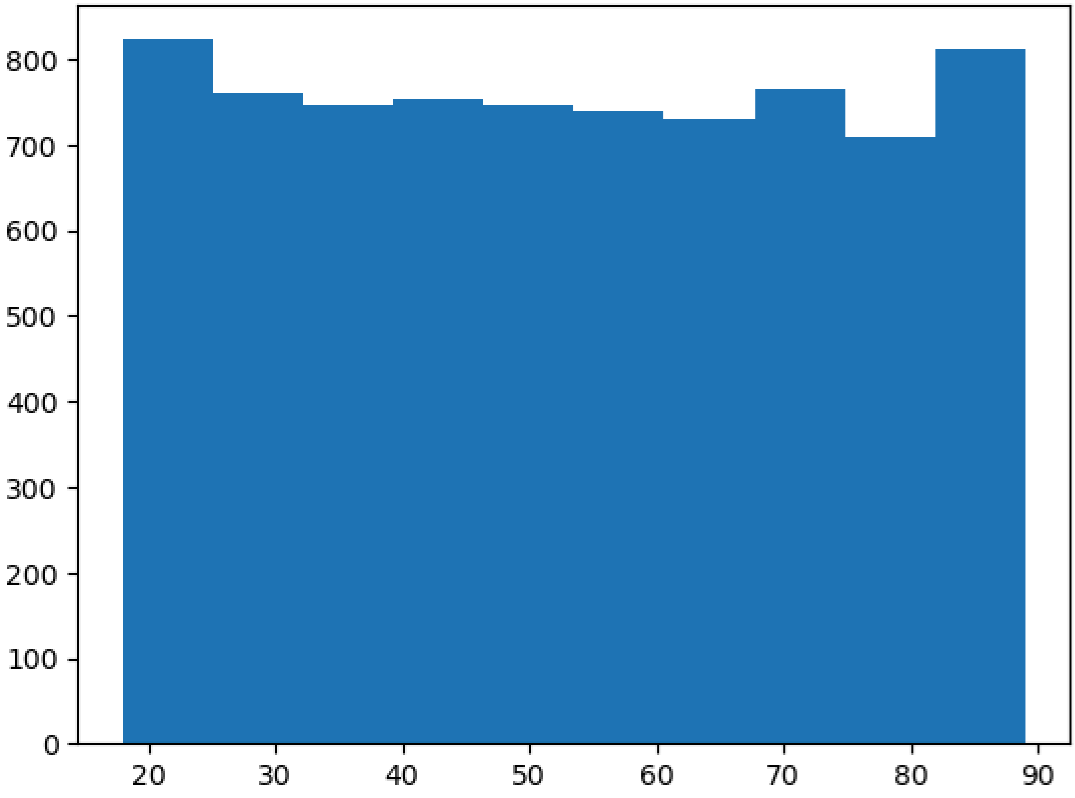
In [14]:

```
plt.hist(df['Children'])
plt.show()
```



In [15]:

```
#impute data into Age variable
plt.hist(df['Age'])
plt.show()
```



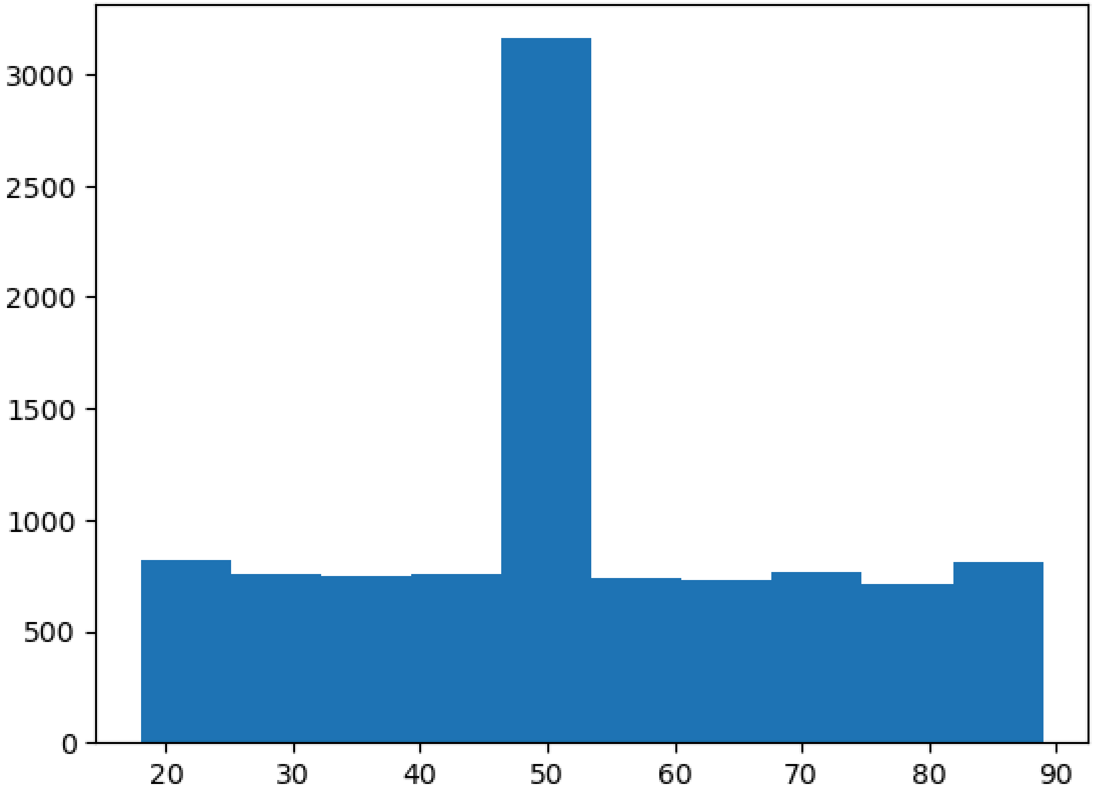
In [16]:
```
#impute age using mean due to uniform distribution
df['Age'].fillna(df['Age'].mean(), inplace= True)
```

In [17]:
```
df.isnull().sum()
```

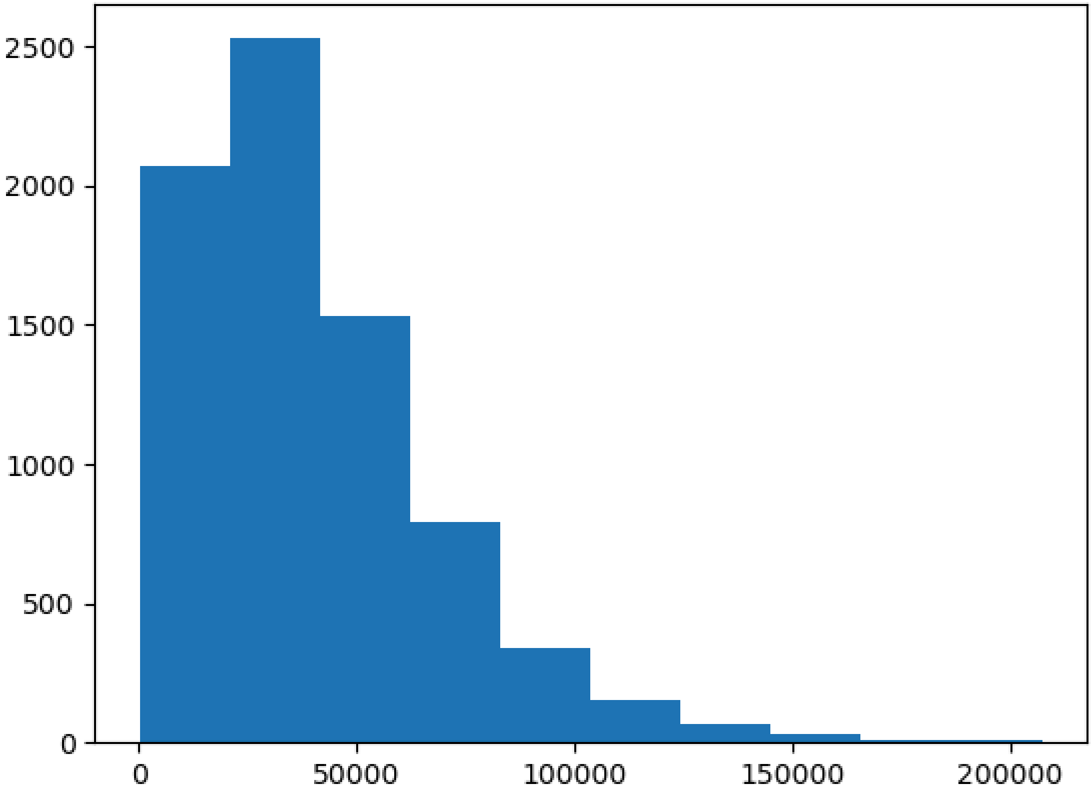
Out[17]:

```
Unnamed: 0               0
CaseOrder                0
Customer_id              0
Interaction              0
UID                      0
City                     0
State                    0
County                   0
Zip                      0
Lat                      0
Lng                      0
Population               0
Area                     0
Timezone                 0
Job                      0
Children                 0
Age                      0
Education                0
Employment               0
Income                2464
Marital                  0
Gender                   0
ReAdmis                  0
VitD_levels              0
Doc_visits               0
Full_meals_eaten         0
VitD_supp                0
Soft_drink            2467
Initial_admin            0
HighBlood                0
Stroke                   0
Complication_risk        0
Overweight             982
Arthritis                0
Diabetes                 0
Hyperlipidemia           0
BackPain                 0
Anxiety                984
Allergic_rhinitis        0
Reflux_esophagitis       0
Asthma                   0
Services                 0
Initial_days          1056
TotalCharge              0
Additional_charges       0
Item1                    0
Item2                    0
Item3                    0
Item4                    0
Item5                    0
Item6                    0
Item7                    0
Item8                    0
dtype: int64
```

In [18]:
```
#verify Age variable
plt.hist(df['Age'])
plt.show()
```



In [19]:
```
#impute data into the Income variable
plt.hist(df['Income'])
plt.show()
```

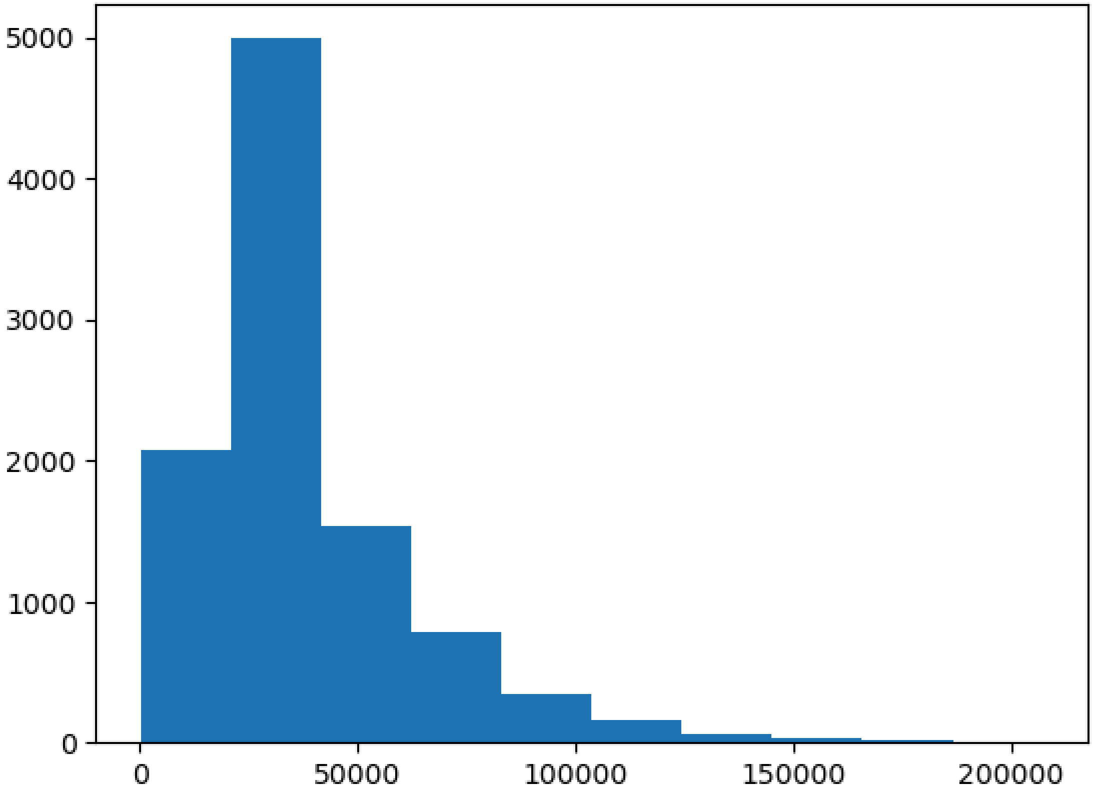


In [20]:
```
#impute Income using median due to skewed right distribution
df['Income'].fillna(df['Income'].median(),inplace= True)
```

In [21]:
```
df.isnull().sum()
```

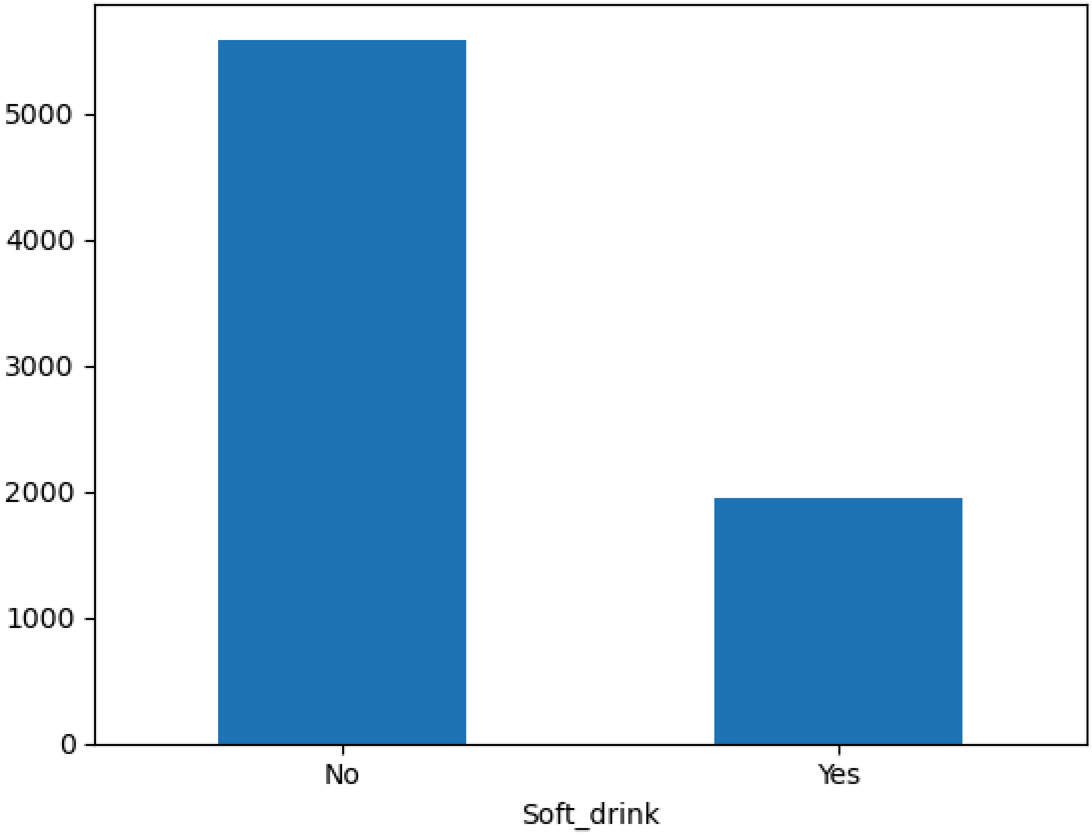
```
Out[21]: Unnamed: 0               0
         CaseOrder                0
         Customer_id              0
         Interaction              0
         UID                      0
         City                     0
         State                    0
         County                   0
         Zip                      0
         Lat                      0
         Lng                      0
         Population               0
         Area                     0
         Timezone                 0
         Job                      0
         Children                 0
         Age                      0
         Education                0
         Employment               0
         Income                   0
         Marital                  0
         Gender                   0
         ReAdmis                  0
         VitD_levels              0
         Doc_visits               0
         Full_meals_eaten         0
         VitD_supp                0
         Soft_drink            2467
         Initial_admin            0
         HighBlood                0
         Stroke                   0
         Complication_risk        0
         Overweight             982
         Arthritis                0
         Diabetes                 0
         Hyperlipidemia           0
         BackPain                 0
         Anxiety                984
         Allergic_rhinitis        0
         Reflux_esophagitis       0
         Asthma                   0
         Services                 0
         Initial_days          1056
         TotalCharge              0
         Additional_charges       0
         Item1                    0
         Item2                    0
         Item3                    0
         Item4                    0
         Item5                    0
         Item6                    0
         Item7                    0
         Item8                    0
         dtype: int64
```

In [22]:
```
#verify Income variable
plt.hist(df['Income'])
plt.show()
```



In [23]:
```
#impute data into the Soft_drink variable
df['Soft_drink'].value_counts().plot.bar(rot=0)
```

Out[23]:
```
<Axes: xlabel='Soft_drink'>
```



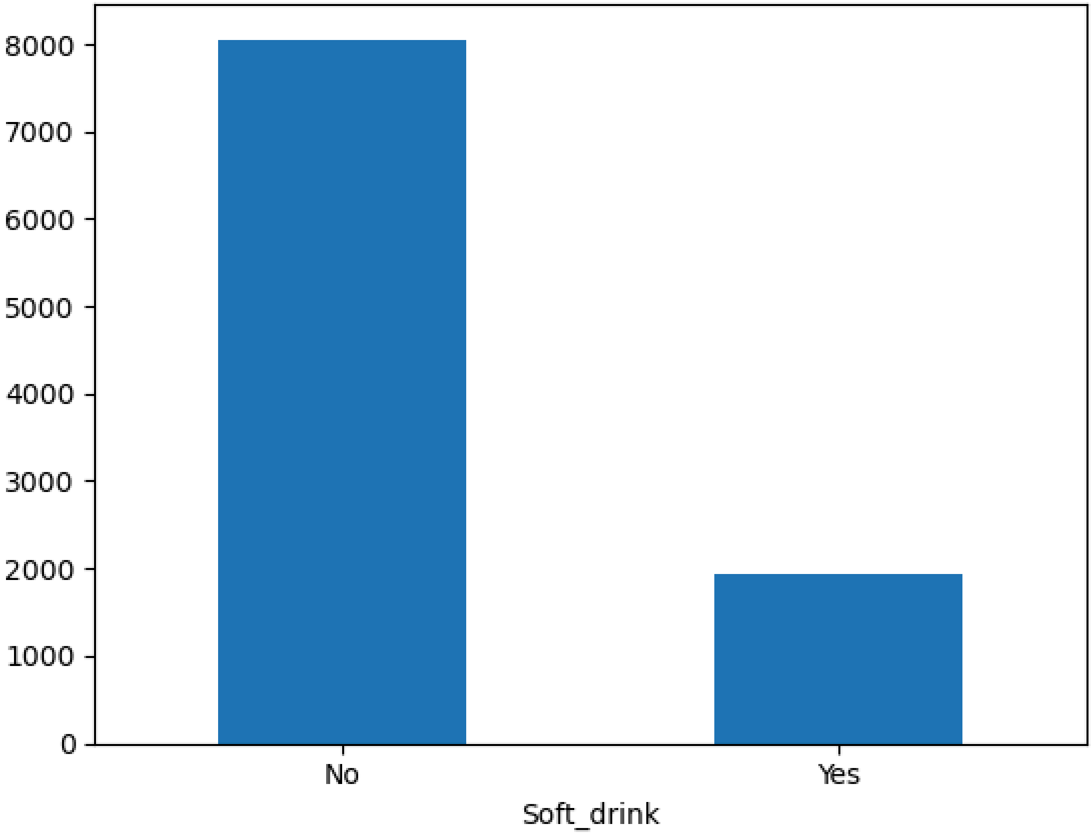
In [24]:
```
#impute Soft_drink using mode due to categorical data
df['Soft_drink'] = df['Soft_drink'].fillna(df['Soft_drink'].mode()[0])
```

In [25]:
```
#verify Soft_drink variable
df.isnull().sum()
```

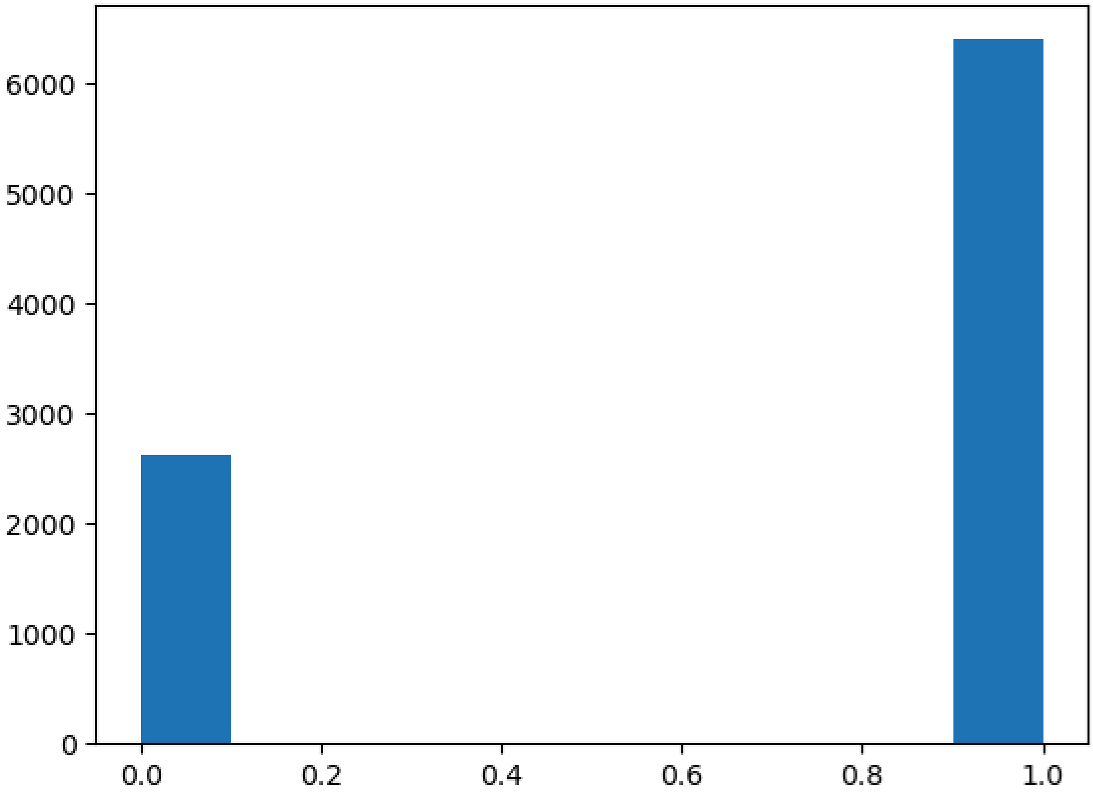
```
Out[25]: Unnamed: 0               0
         CaseOrder                0
         Customer_id              0
         Interaction              0
         UID                      0
         City                     0
         State                    0
         County                   0
         Zip                      0
         Lat                      0
         Lng                      0
         Population               0
         Area                     0
         Timezone                 0
         Job                      0
         Children                 0
         Age                      0
         Education                0
         Employment               0
         Income                   0
         Marital                  0
         Gender                   0
         ReAdmis                  0
         VitD_levels              0
         Doc_visits               0
         Full_meals_eaten         0
         VitD_supp                0
         Soft_drink               0
         Initial_admin            0
         HighBlood                0
         Stroke                   0
         Complication_risk        0
         Overweight             982
         Arthritis                0
         Diabetes                 0
         Hyperlipidemia           0
         BackPain                 0
         Anxiety                984
         Allergic_rhinitis        0
         Reflux_esophagitis       0
         Asthma                   0
         Services                 0
         Initial_days          1056
         TotalCharge              0
         Additional_charges       0
         Item1                    0
         Item2                    0
         Item3                    0
         Item4                    0
         Item5                    0
         Item6                    0
         Item7                    0
         Item8                    0
         dtype: int64
```

```
In [26]: df['Soft_drink'].value_counts().plot.bar(rot=0)
```

```
Out[26]: <Axes: xlabel='Soft_drink'>
```



```
In [27]: #impute data into the Overweight variable
         plt.hist(df['Overweight'])
         plt.show()
```



In [28]:

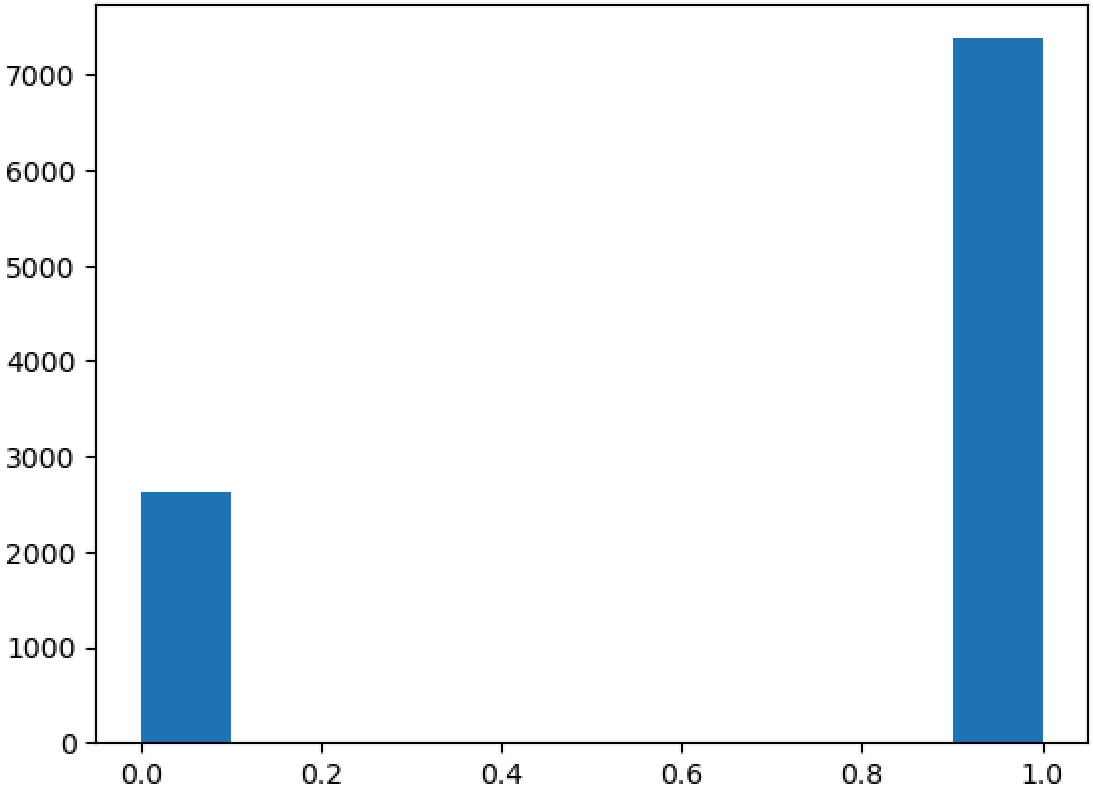
```
#impute Overweight using mode due to categorical data
df['Overweight'] = df['Overweight'].fillna(df['Overweight'].mode()[0])
```

In [29]:

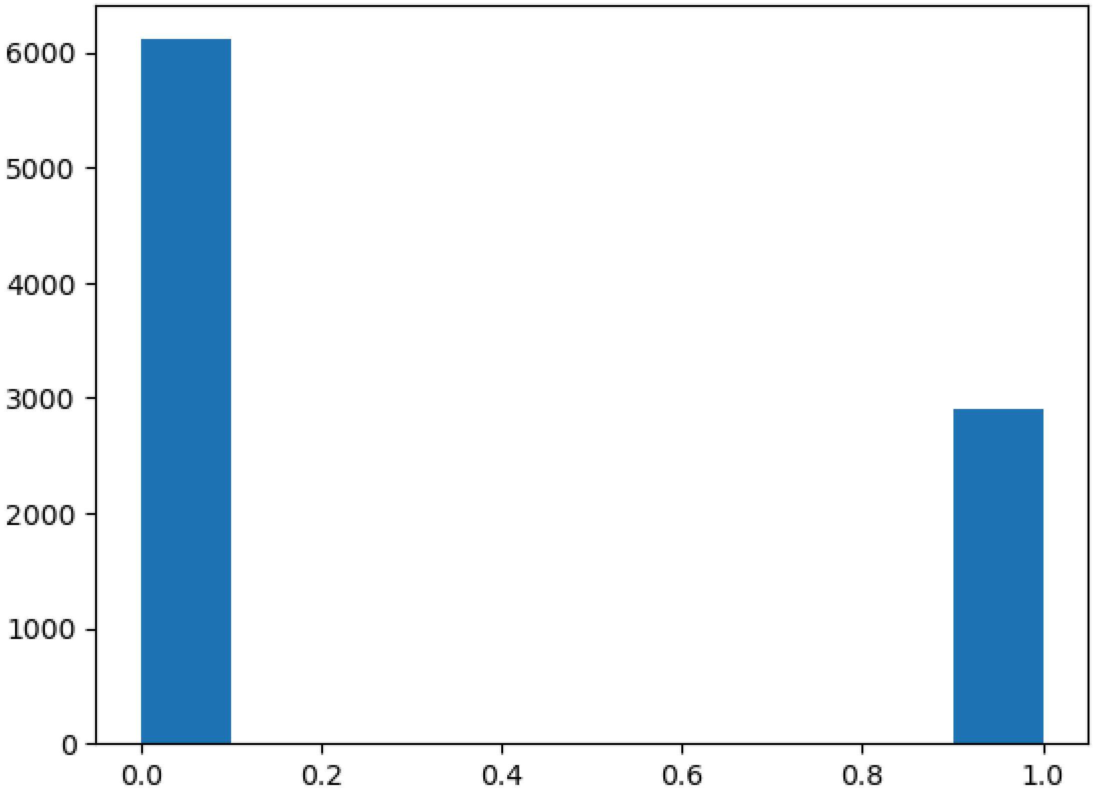
```
#verify Overweight variable
df.isnull().sum()
```

```
Out[29]: Unnamed: 0               0
          CaseOrder                0
          Customer_id              0
          Interaction              0
          UID                      0
          City                     0
          State                    0
          County                   0
          Zip                      0
          Lat                      0
          Lng                      0
          Population               0
          Area                     0
          Timezone                 0
          Job                      0
          Children                 0
          Age                      0
          Education                0
          Employment               0
          Income                   0
          Marital                  0
          Gender                   0
          ReAdmis                  0
          VitD_levels              0
          Doc_visits               0
          Full_meals_eaten         0
          VitD_supp                0
          Soft_drink               0
          Initial_admin            0
          HighBlood                0
          Stroke                   0
          Complication_risk        0
          Overweight               0
          Arthritis                0
          Diabetes                 0
          Hyperlipidemia           0
          BackPain                 0
          Anxiety                984
          Allergic_rhinitis        0
          Reflux_esophagitis       0
          Asthma                   0
          Services                 0
          Initial_days          1056
          TotalCharge              0
          Additional_charges       0
          Item1                    0
          Item2                    0
          Item3                    0
          Item4                    0
          Item5                    0
          Item6                    0
          Item7                    0
          Item8                    0
          dtype: int64
```

```
In [30]: #Verify imputation of data into the Overweight variable
         plt.hist(df['Overweight'])
         plt.show()
```



```
In [31]: #impute data into the Anxiety variable
         plt.hist(df['Anxiety'])
         plt.show()
```



In [32]:

```
#impute Anxiety using mode due to categorical data
df['Anxiety'] = df['Anxiety'].fillna(df['Anxiety'].mode()[0])
```

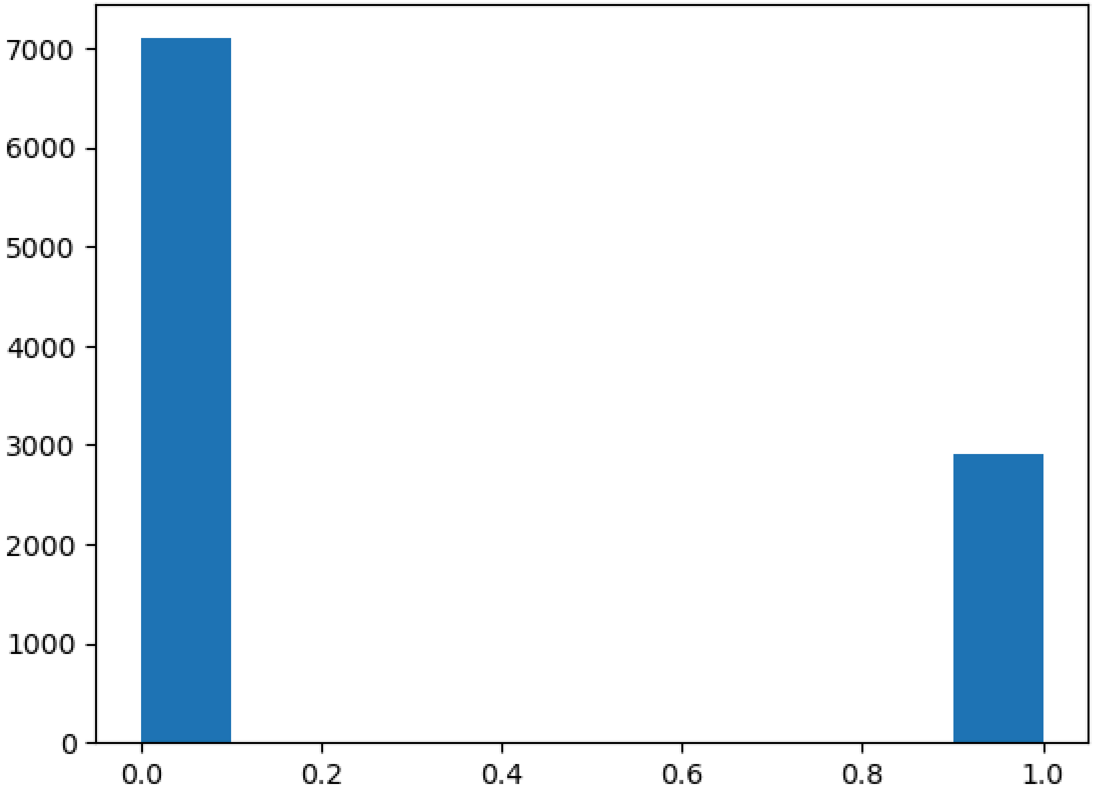
In [33]:

```
#verify Anxiety variable
df.isnull().sum()
```

```
Out[33]: Unnamed: 0               0
         CaseOrder                0
         Customer_id              0
         Interaction              0
         UID                      0
         City                     0
         State                    0
         County                   0
         Zip                      0
         Lat                      0
         Lng                      0
         Population               0
         Area                     0
         Timezone                 0
         Job                      0
         Children                 0
         Age                      0
         Education                0
         Employment               0
         Income                   0
         Marital                  0
         Gender                   0
         ReAdmis                  0
         VitD_levels              0
         Doc_visits               0
         Full_meals_eaten         0
         VitD_supp                0
         Soft_drink               0
         Initial_admin            0
         HighBlood                0
         Stroke                   0
         Complication_risk        0
         Overweight               0
         Arthritis                0
         Diabetes                 0
         Hyperlipidemia           0
         BackPain                 0
         Anxiety                  0
         Allergic_rhinitis        0
         Reflux_esophagitis       0
         Asthma                   0
         Services                 0
         Initial_days          1056
         TotalCharge              0
         Additional_charges       0
         Item1                    0
         Item2                    0
         Item3                    0
         Item4                    0
         Item5                    0
         Item6                    0
         Item7                    0
         Item8                    0
         dtype: int64
```

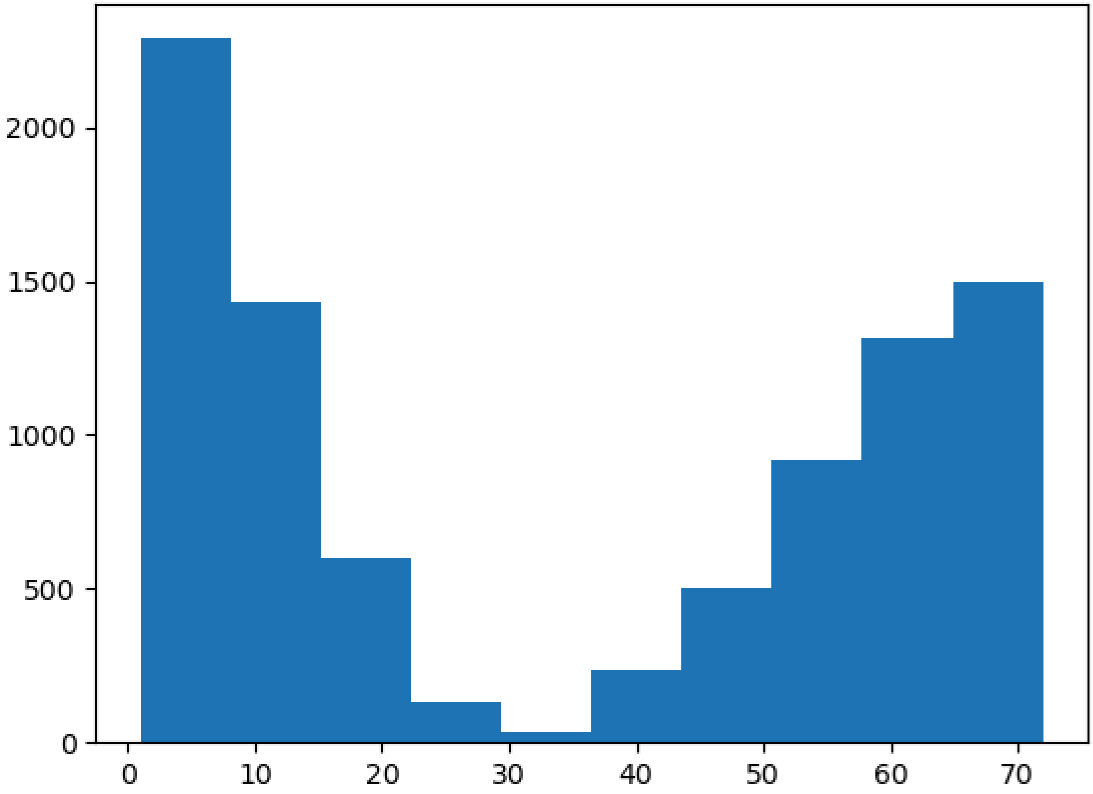
In [34]:

```
#Verify imputation of data into the Anxiety variable
plt.hist(df['Anxiety'])
plt.show()
```



In [35]:

```
#impute data into the Initial_days variable
plt.hist(df['Initial_days'])
plt.show()
```



In [36]:

```
#impute Initial_days using mode due to bimodal distribution
df['Initial_days'] = df['Initial_days'].fillna(df['Initial_days'].mode()[0])
```

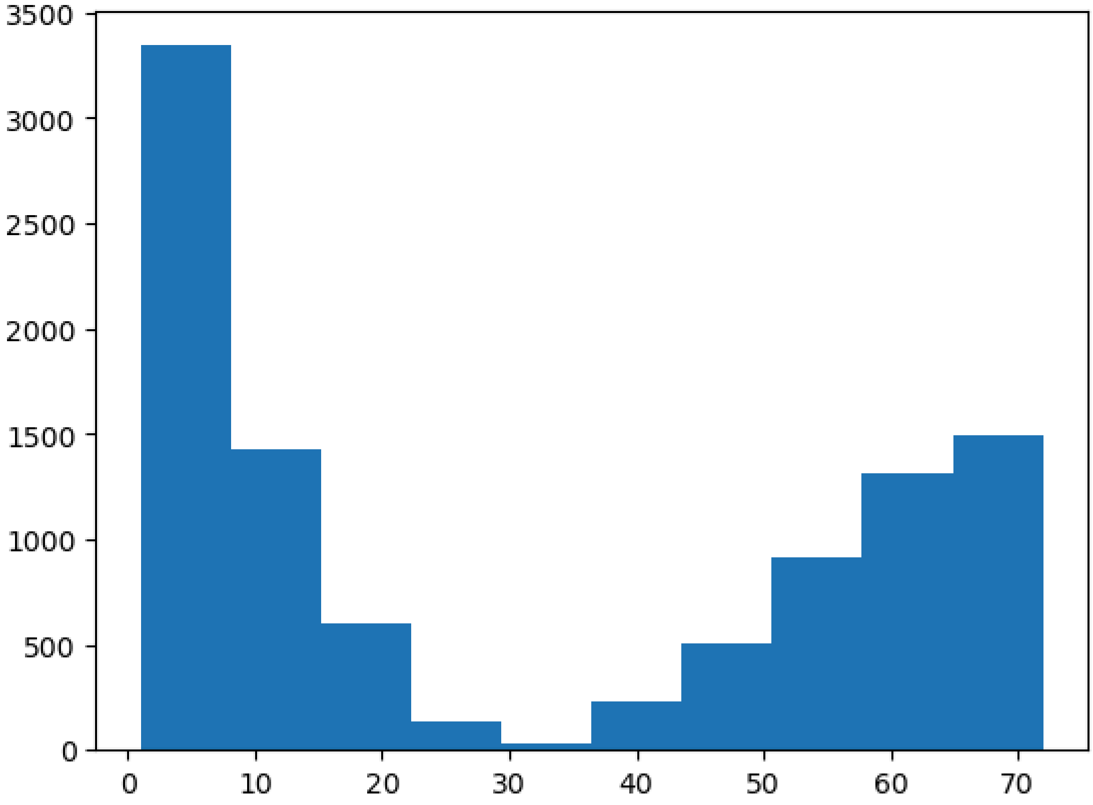
In [37]:

```
#verify Initial_days variable
df.isnull().sum()
```

```
Out[37]: Unnamed: 0            0
         CaseOrder             0
         Customer_id           0
         Interaction           0
         UID                   0
         City                  0
         State                 0
         County                0
         Zip                   0
         Lat                   0
         Lng                   0
         Population            0
         Area                  0
         Timezone              0
         Job                   0
         Children              0
         Age                   0
         Education             0
         Employment            0
         Income                0
         Marital               0
         Gender                0
         ReAdmis               0
         VitD_levels           0
         Doc_visits            0
         Full_meals_eaten      0
         VitD_supp             0
         Soft_drink            0
         Initial_admin         0
         HighBlood             0
         Stroke                0
         Complication_risk     0
         Overweight            0
         Arthritis             0
         Diabetes              0
         Hyperlipidemia        0
         BackPain              0
         Anxiety               0
         Allergic_rhinitis     0
         Reflux_esophagitis    0
         Asthma                0
         Services              0
         Initial_days          0
         TotalCharge           0
         Additional_charges    0
         Item1                 0
         Item2                 0
         Item3                 0
         Item4                 0
         Item5                 0
         Item6                 0
         Item7                 0
         Item8                 0
         dtype: int64
```

In [38]:

```python
#Verify imputation of data into the Initial_days variable
plt.hist(df['Initial_days'])
plt.show()
```

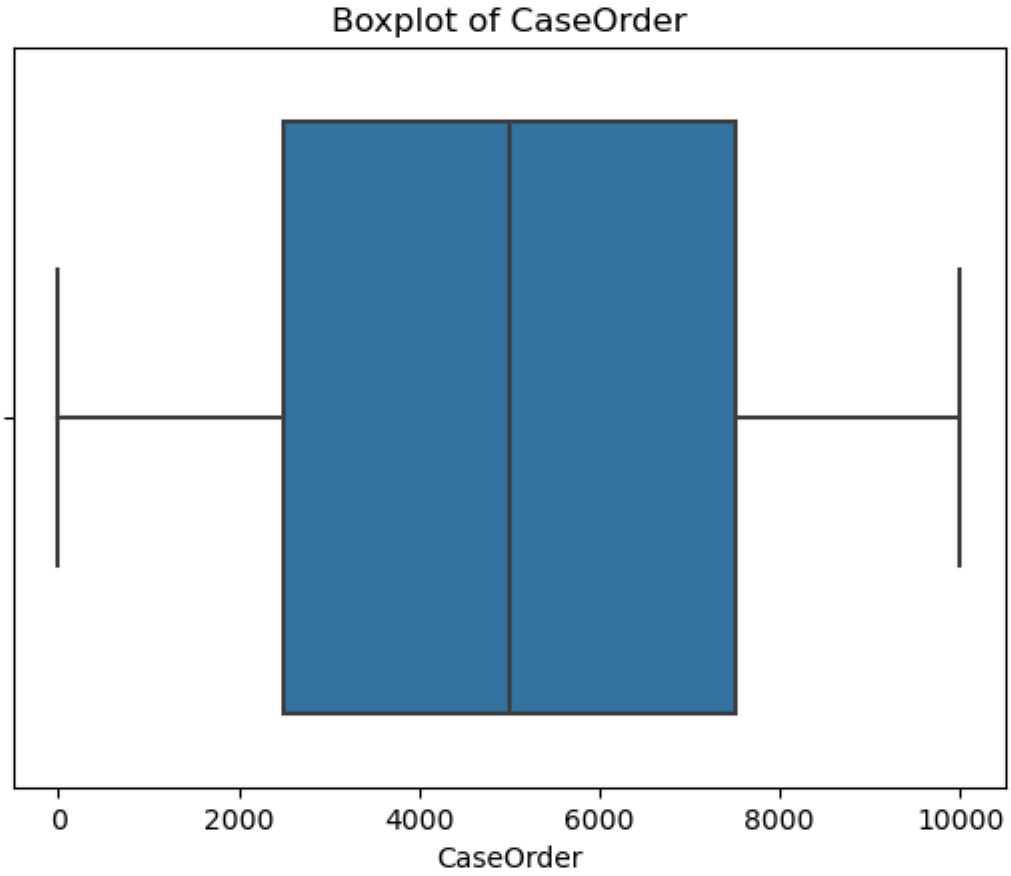


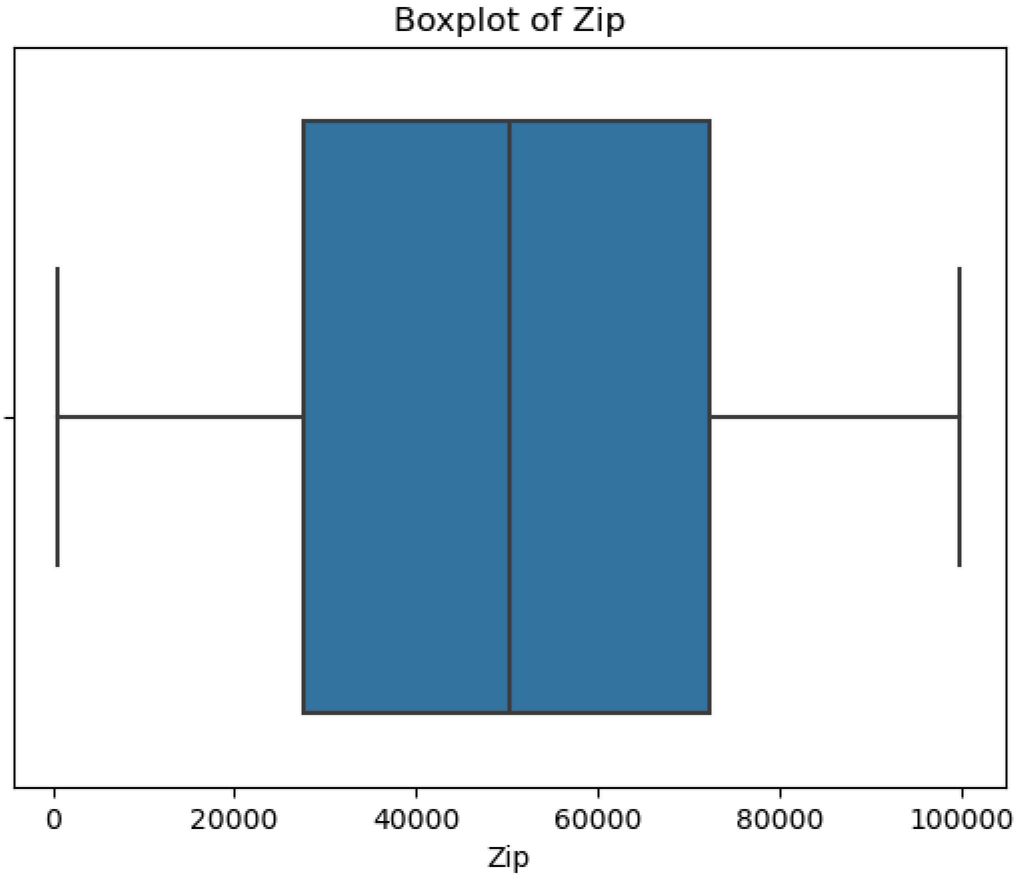
In [42]:

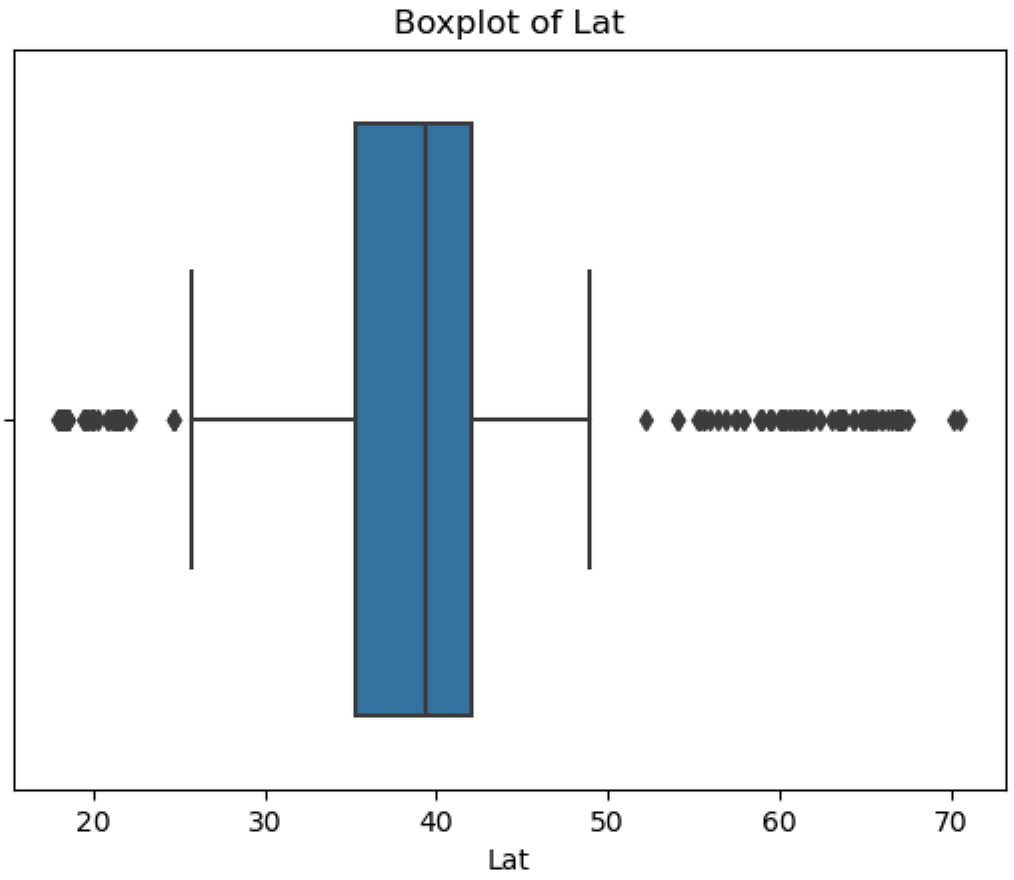
```python
#placing numerical columns in an array
outlier_col = ['CaseOrder', 'Zip', 'Lat', 'Lng', 'Population', 'Children', 'Age', '
```

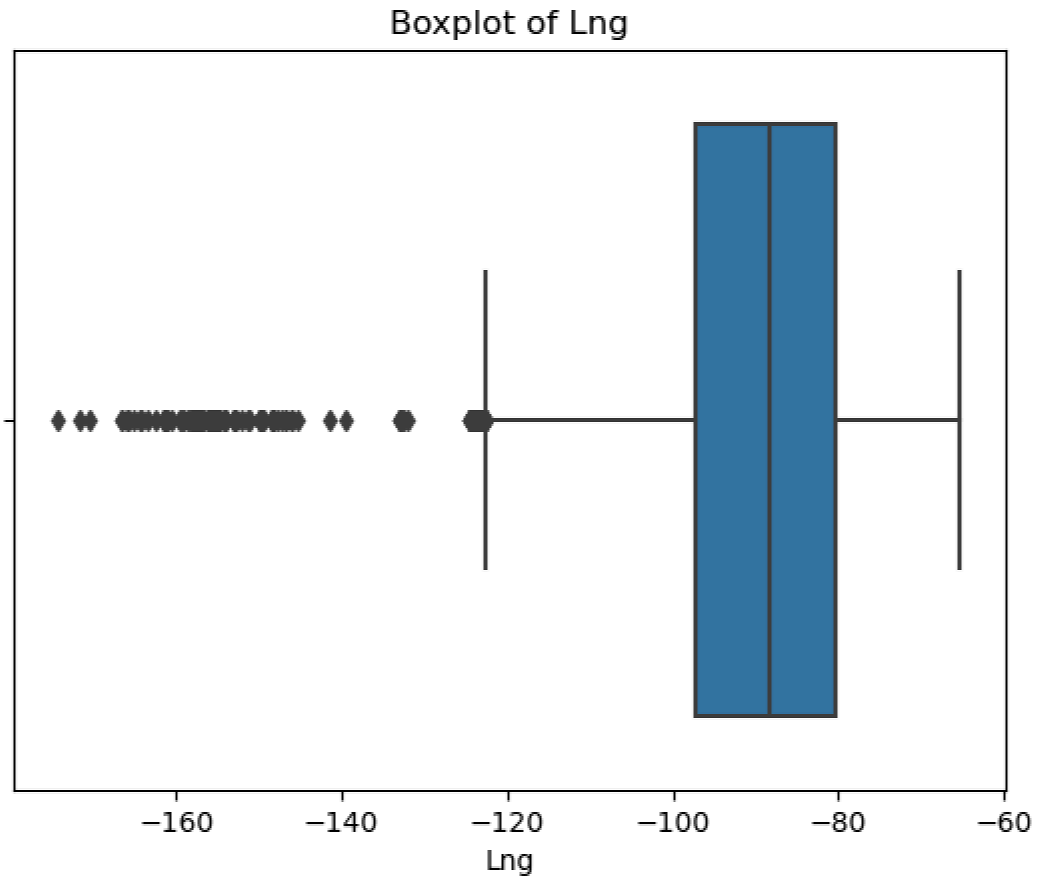
In [43]:

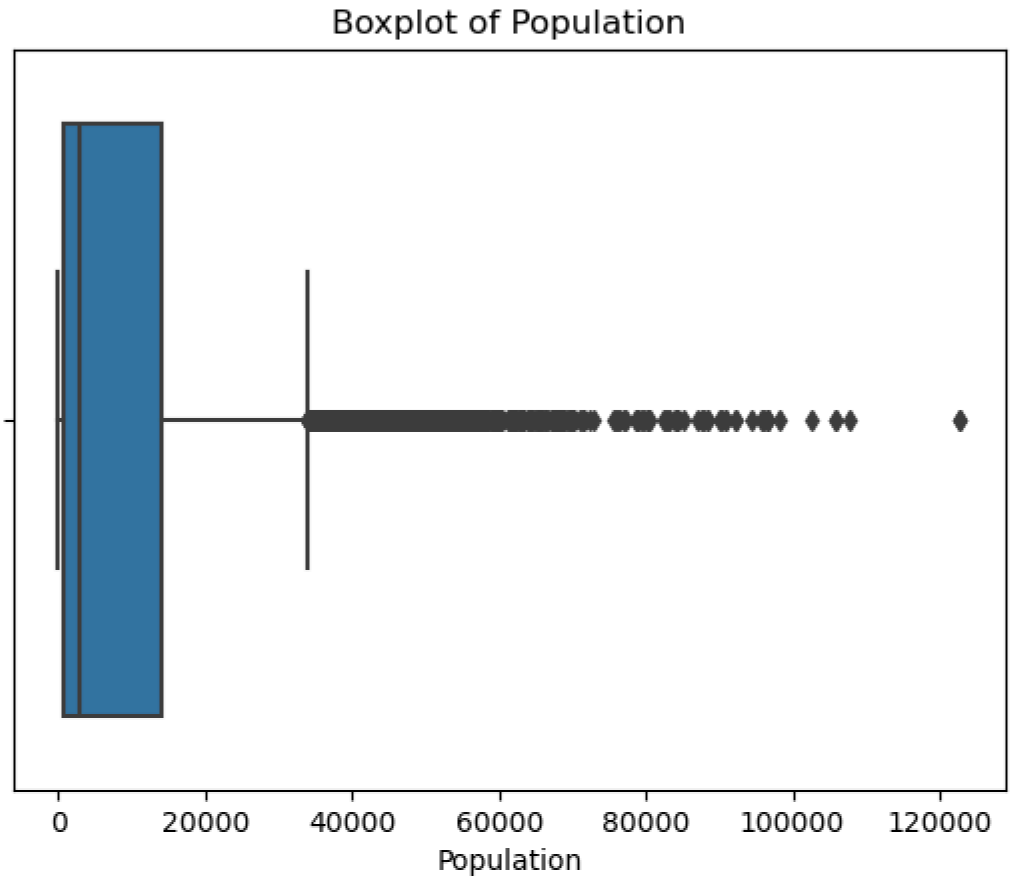
```python
# creating boxplot graphs for each numeric variable
for i in outlier_col:
    plt.figure()
    sns.boxplot(x=i, data=df)
    plt.title(f'Boxplot of {i}')
    plt.show()
```

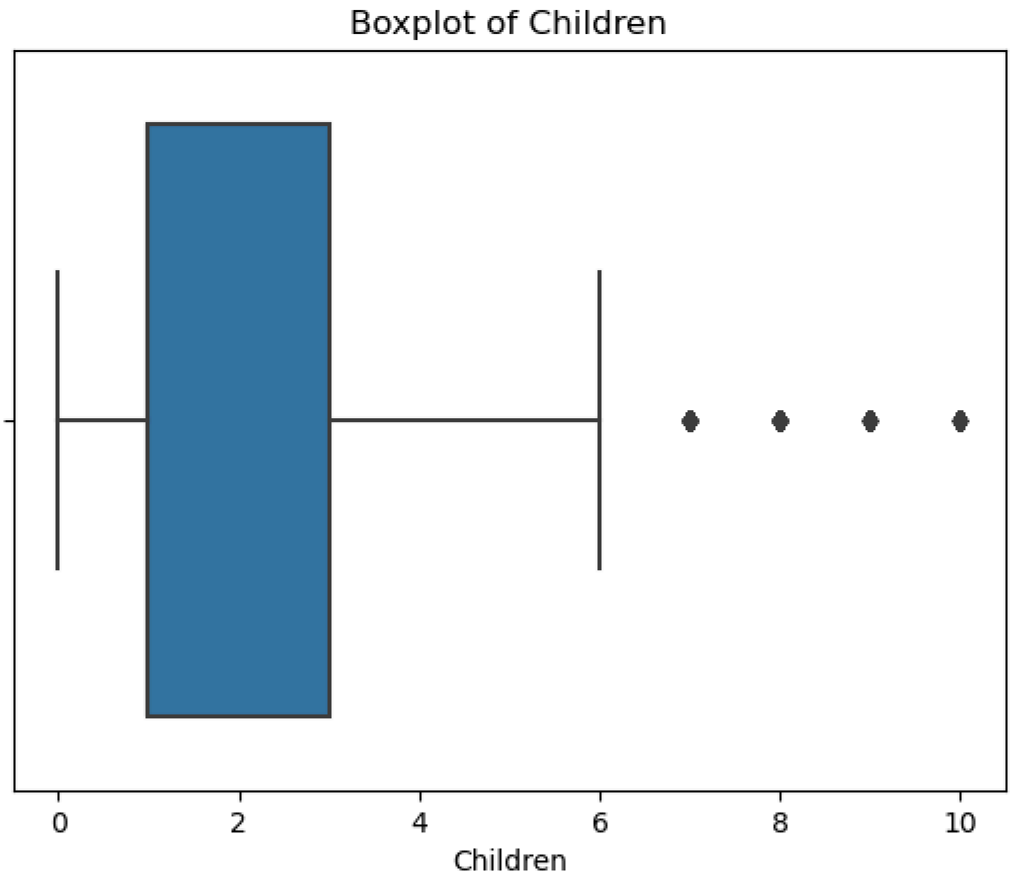

Boxplot of CaseOrder
0
2000
4000
6000
8000
10000
CaseOrder

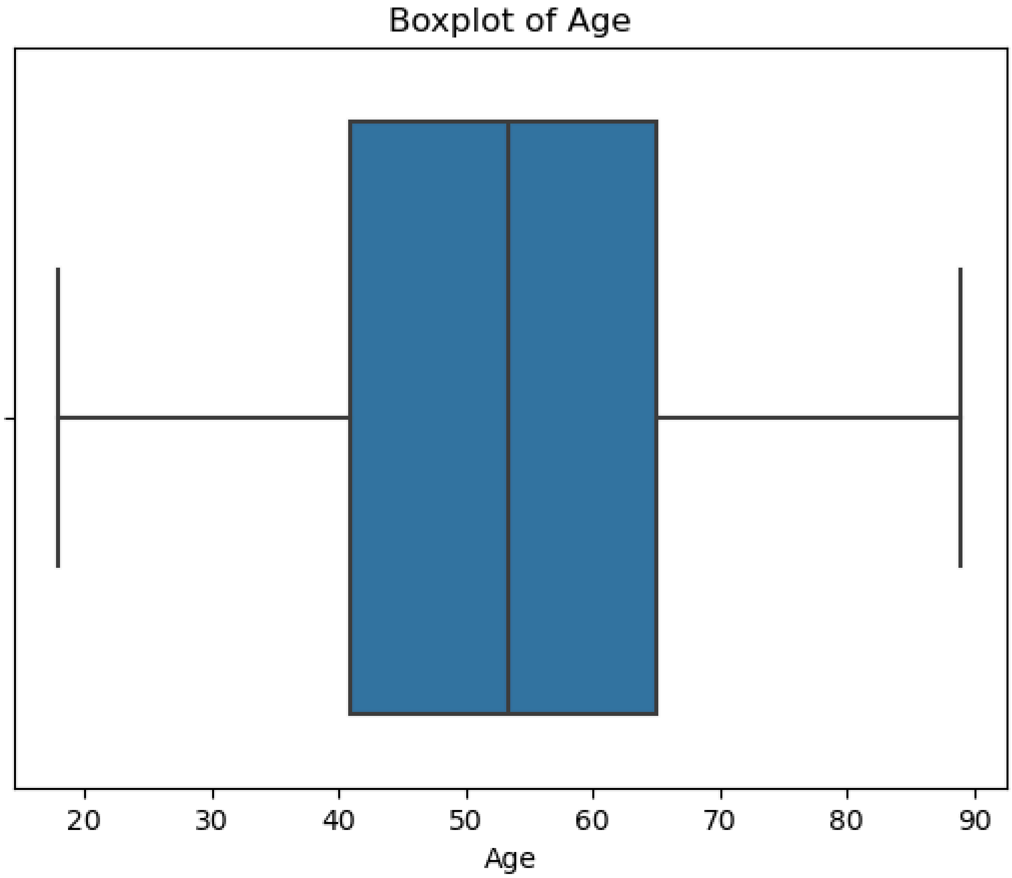

Boxplot of Zip
0
20000
40000
60000
80000
100000
Zip

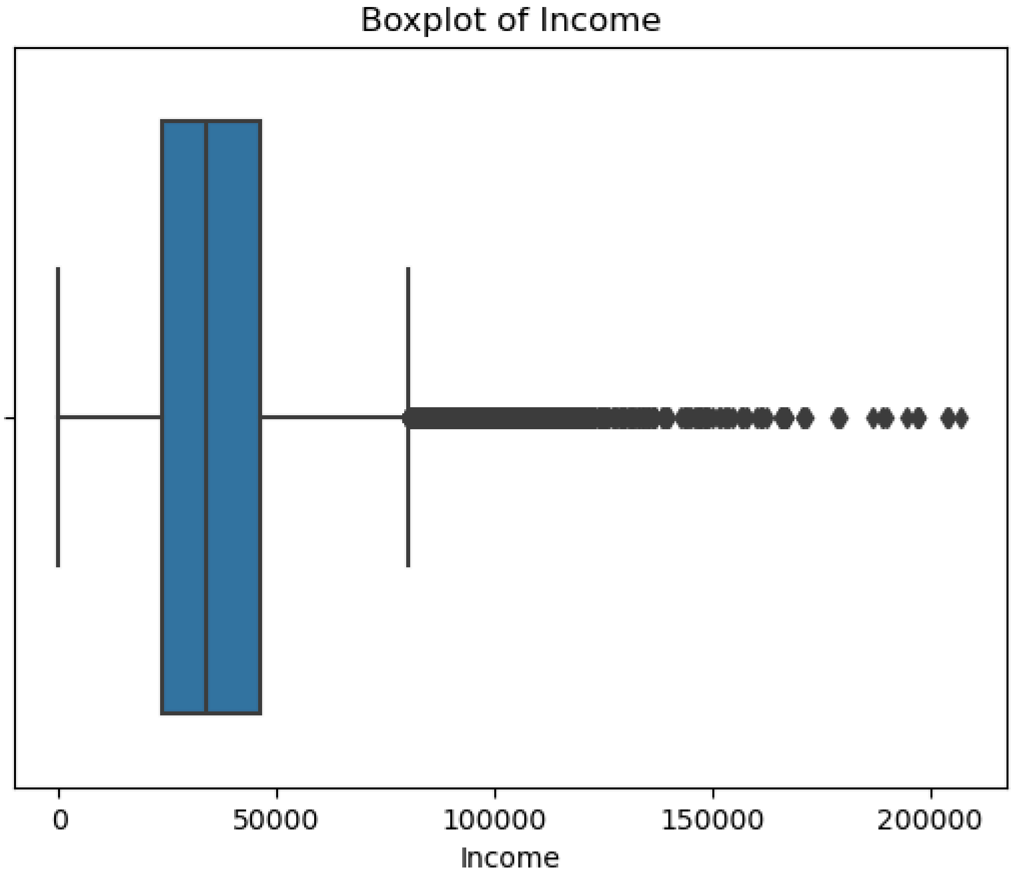

Boxplot of Lat
20
30
40
50
60
70
Lat

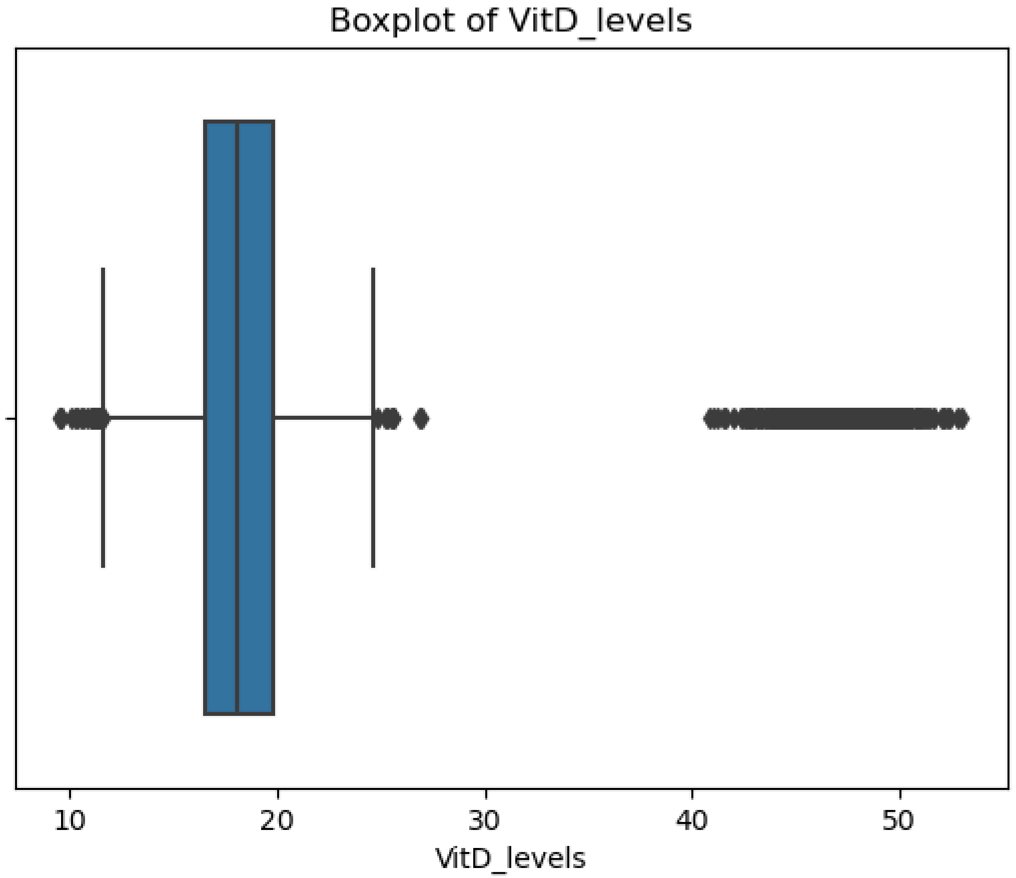

Boxplot of Lng
−160
−140
−120
−100
−80
−60
Lng

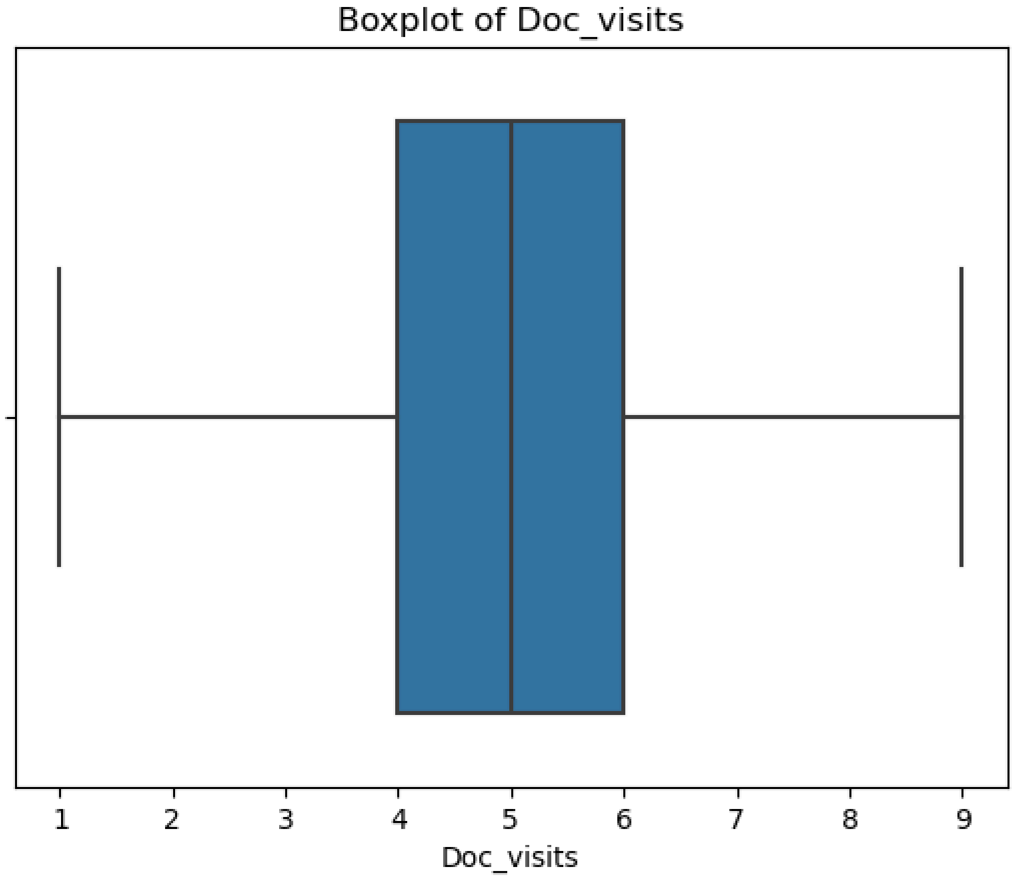

Boxplot of Population
0
20000
40000
60000
80000
100000
120000
Population

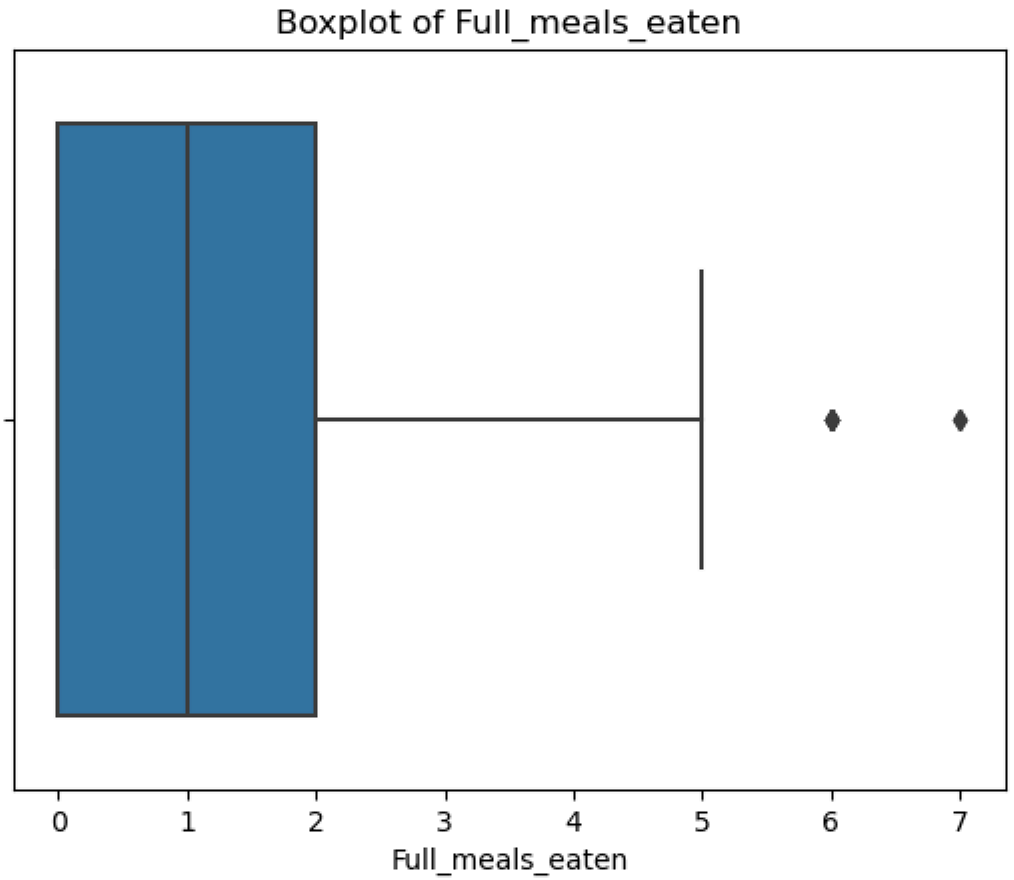

Boxplot of Children
0
2
4
6
8
10
Children

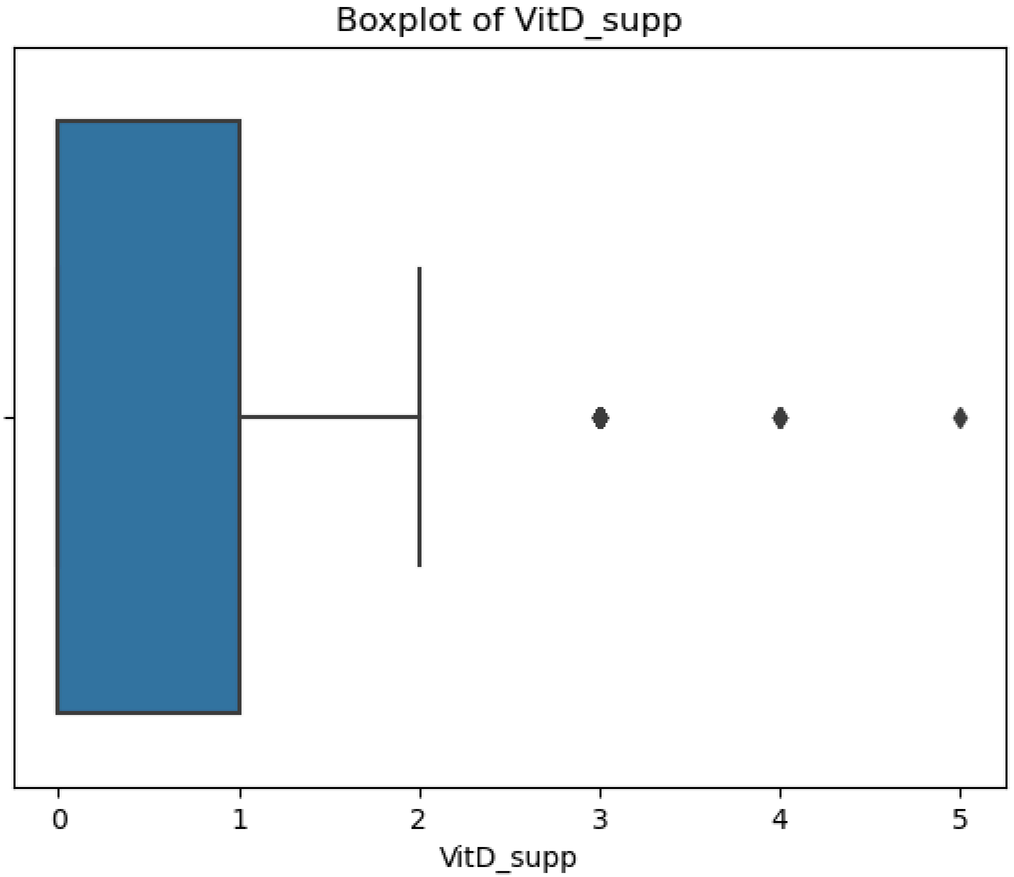

Boxplot of Age
20
30
40
50
60
70
80
90
Age

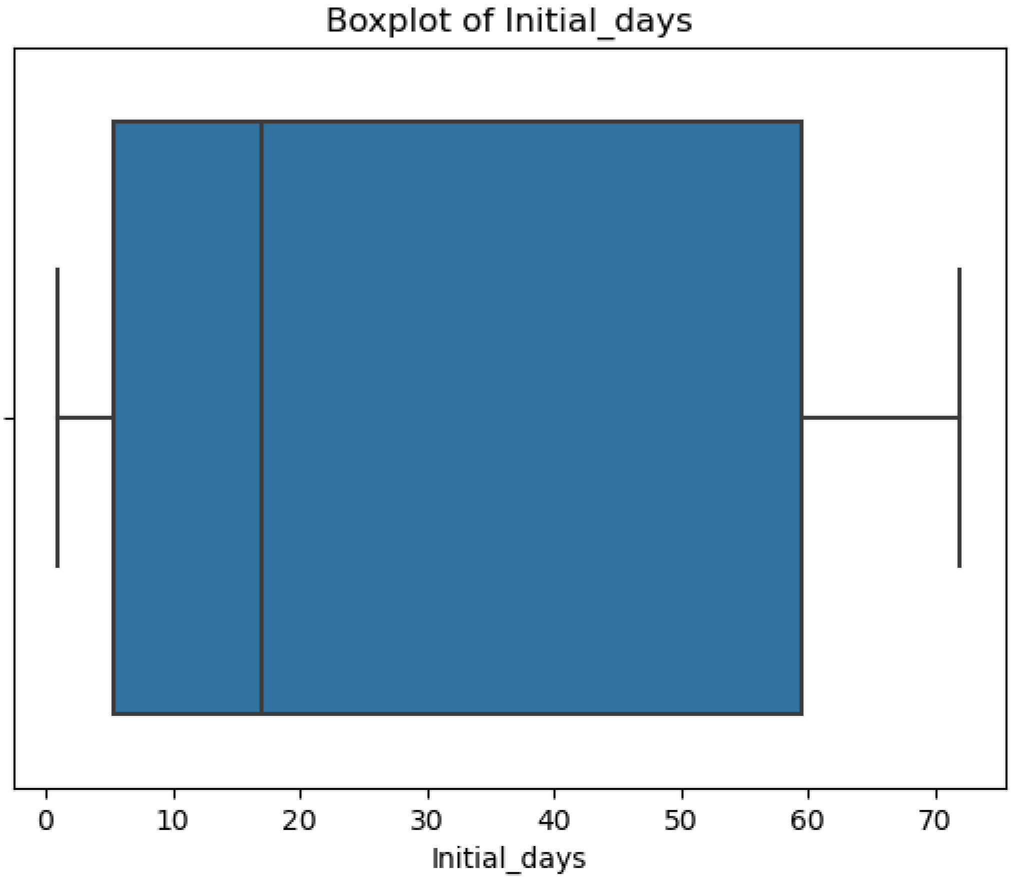

Boxplot of Income
0
50000
100000
150000
200000
Income

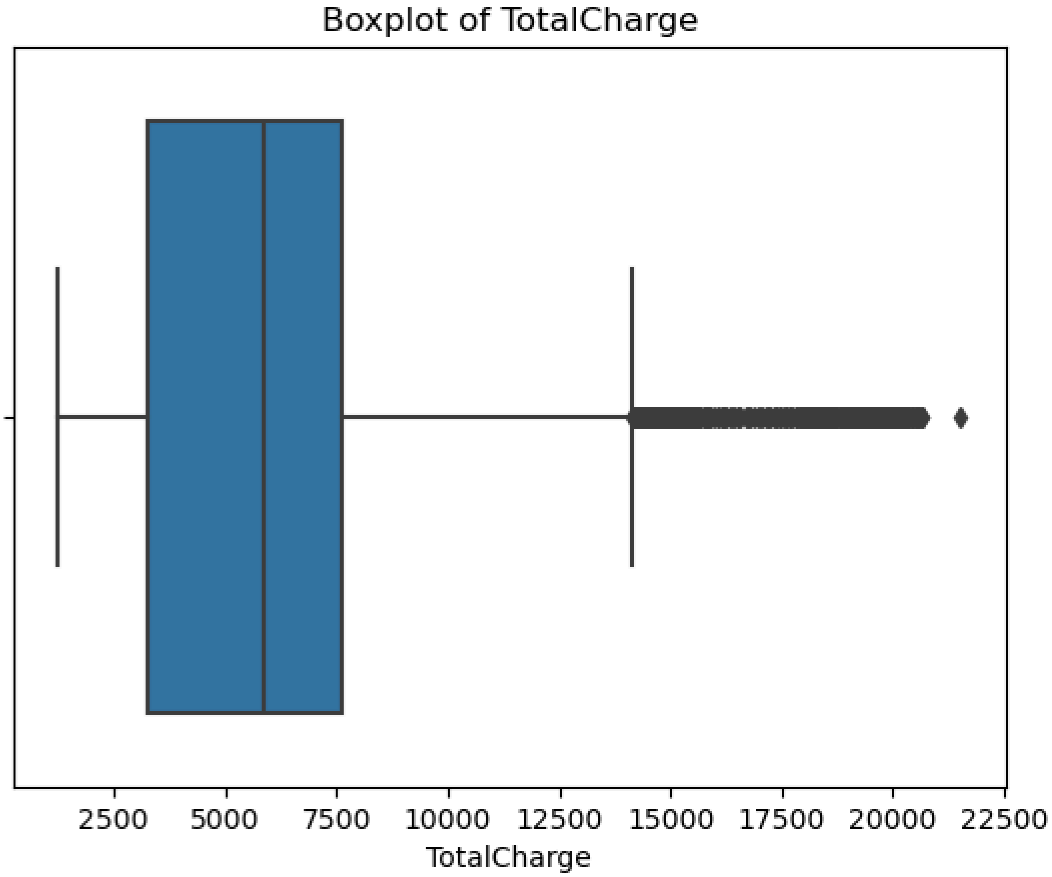

Boxplot of VitD_levels
10
20
30
40
50
VitD_levels

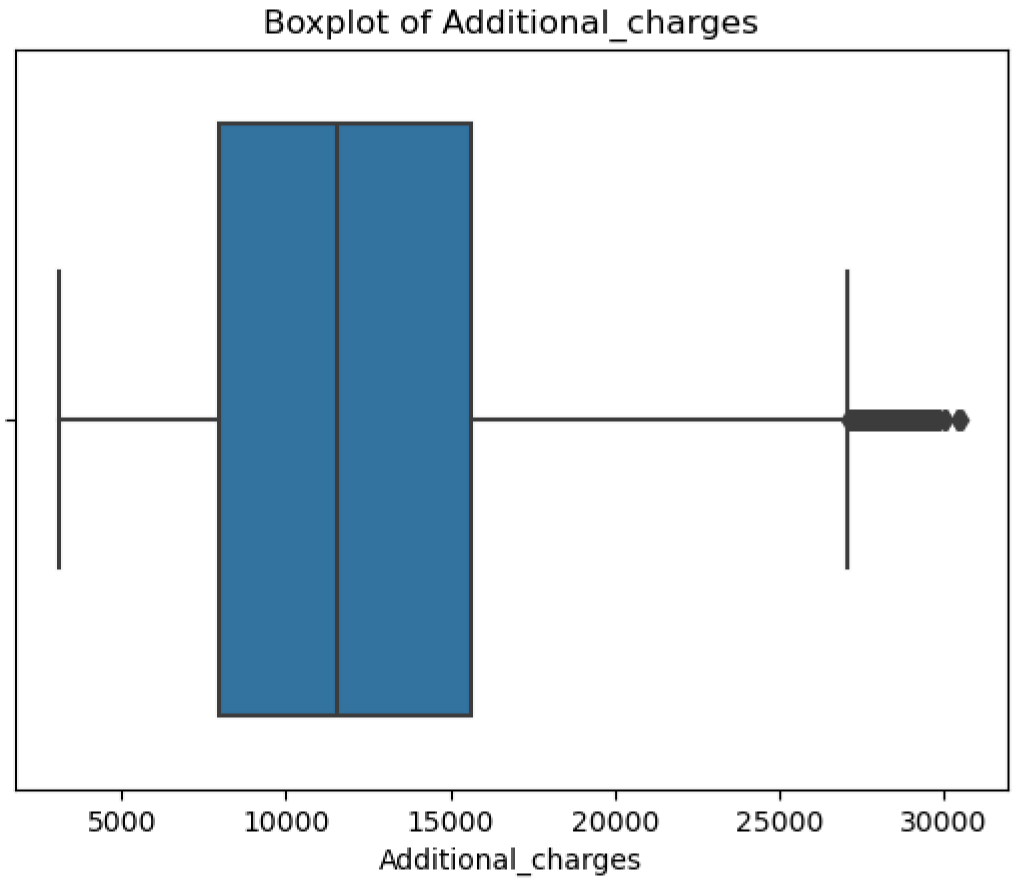

Boxplot of Doc_visits
1
2
3
4
5
6
7
8
9
Doc_visits


Boxplot of Full_meals_eaten
0
1
2
3
4
5
6
7
Full_meals_eaten


Boxplot of VitD_supp
0
1
2
3
4
5
VitD_supp


Boxplot of Initial_days
0
10
20
30
40
50
60
70
Initial_days


Boxplot of TotalCharge
2500
5000
7500
10000
12500
15000
17500
20000
22500
TotalCharge

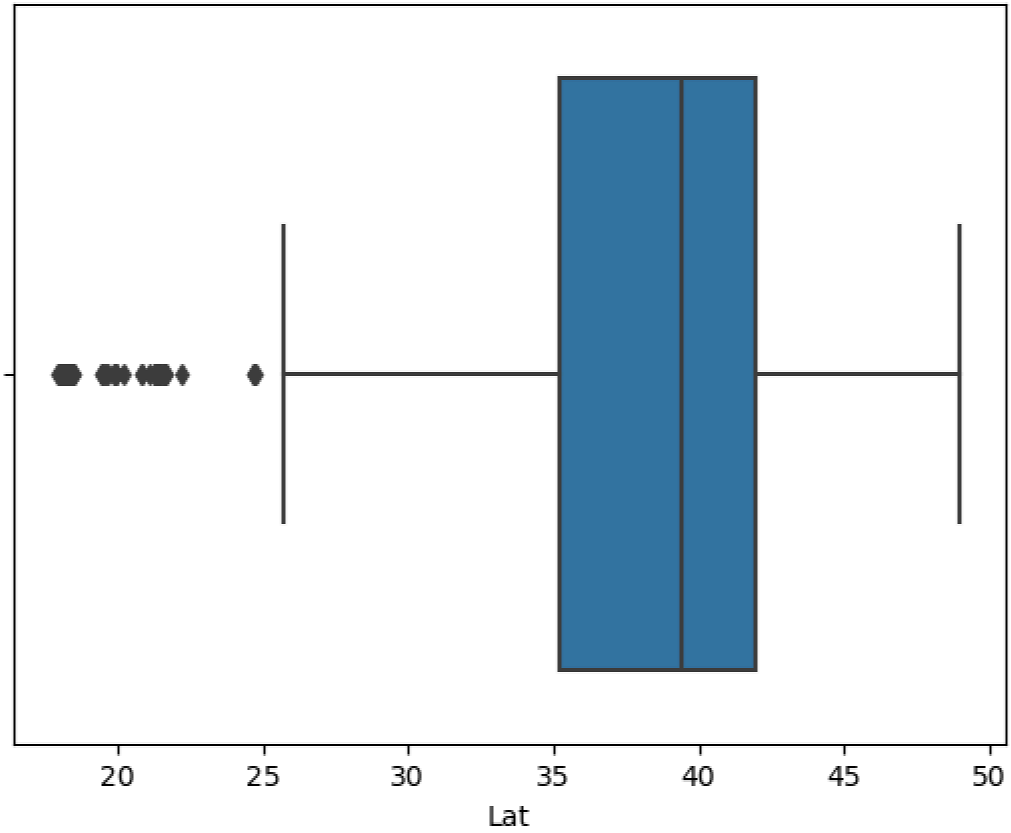


In [39]:

```
#impute outliers with the median for Lat
df['Lat'] = np.where(df['Lat'] >50, np.nan , df['Lat'])
```

In [40]:

```
boxplot=sns.boxplot(x='Lat',data=df)
```



In [41]:
```
df['Lat'] = np.where(df['Lat'] <25, np.nan , df['Lat'])
```

In [42]:
```
df['Lat'].fillna(df['Lat'].median(), inplace = True)
```

In [43]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 53 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Unnamed: 0          10000 non-null  int64  
 1   CaseOrder           10000 non-null  int64  
 2   Customer_id         10000 non-null  object 
 3   Interaction         10000 non-null  object 
 4   UID                 10000 non-null  object 
 5   City                10000 non-null  object 
 6   State               10000 non-null  object 
 7   County              10000 non-null  object 
 8   Zip                 10000 non-null  int64  
 9   Lat                 10000 non-null  float64
 10  Lng                 10000 non-null  float64
 11  Population          10000 non-null  int64  
 12  Area                10000 non-null  object 
 13  Timezone            10000 non-null  object 
 14  Job                 10000 non-null  object 
 15  Children            10000 non-null  float64
 16  Age                 10000 non-null  float64
 17  Education           10000 non-null  object 
 18  Employment          10000 non-null  object 
 19  Income              10000 non-null  float64
 20  Marital             10000 non-null  object 
 21  Gender              10000 non-null  object 
 22  ReAdmis             10000 non-null  object 
 23  VitD_levels         10000 non-null  float64
 24  Doc_visits          10000 non-null  int64  
 25  Full_meals_eaten    10000 non-null  int64  
 26  VitD_supp           10000 non-null  int64  
 27  Soft_drink          10000 non-null  object 
 28  Initial_admin       10000 non-null  object 
 29  HighBlood           10000 non-null  object 
 30  Stroke              10000 non-null  object 
 31  Complication_risk   10000 non-null  object 
 32  Overweight          10000 non-null  float64
 33  Arthritis           10000 non-null  object 
 34  Diabetes            10000 non-null  object 
 35  Hyperlipidemia      10000 non-null  object 
 36  BackPain            10000 non-null  object 
 37  Anxiety             10000 non-null  float64
 38  Allergic_rhinitis   10000 non-null  object 
 39  Reflux_esophagitis  10000 non-null  object 
 40  Asthma              10000 non-null  object 
 41  Services            10000 non-null  object 
 42  Initial_days        10000 non-null  float64
 43  TotalCharge         10000 non-null  float64
 44  Additional_charges  10000 non-null  float64
 45  Item1               10000 non-null  int64  
 46  Item2               10000 non-null  int64  
 47  Item3               10000 non-null  int64  
 48  Item4               10000 non-null  int64  
 49  Item5               10000 non-null  int64  
 50  Item6               10000 non-null  int64  
 51  Item7               10000 non-null  int64  
 52  Item8               10000 non-null  int64  
dtypes: float64(11), int64(15), object(27)
memory usage: 4.0+ MB
```

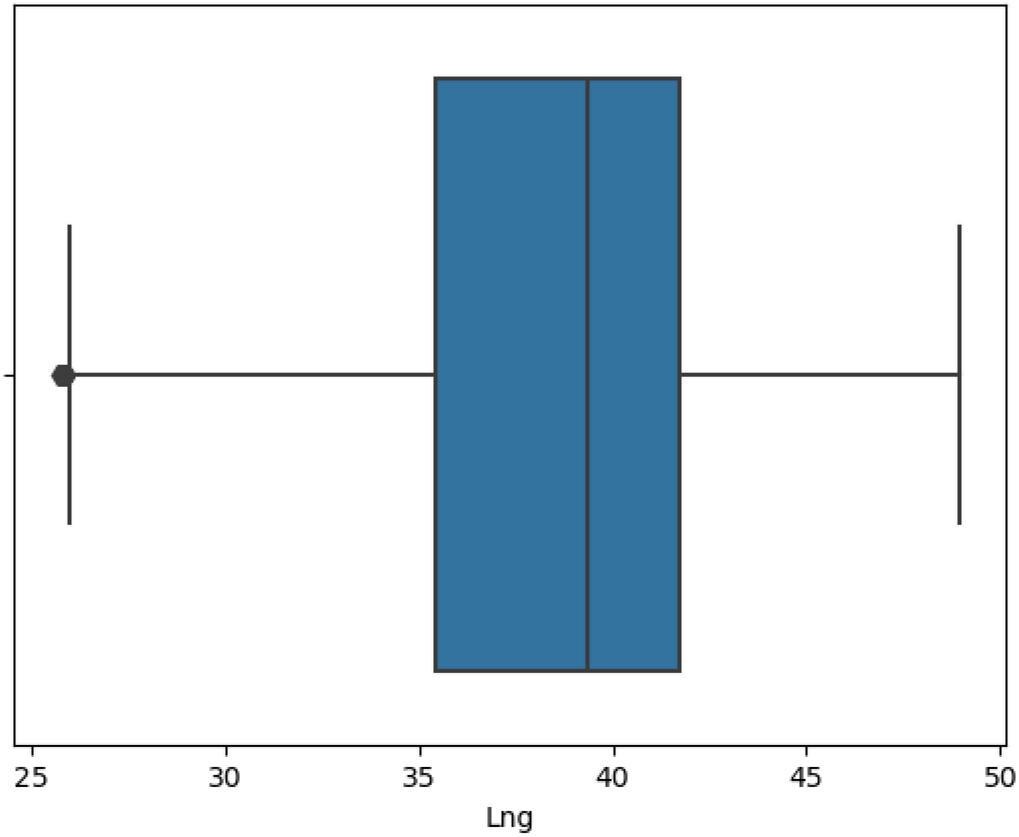
In [44]:

```
#impute outliers with the median for Lng
df['Lng'] = np.where(df['Lng'] < -122, np.nan , df['Lat'])
df['Lng'].fillna(df['Lng'].median(), inplace = True)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 53 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Unnamed: 0          10000 non-null  int64  
 1   CaseOrder           10000 non-null  int64  
 2   Customer_id         10000 non-null  object 
 3   Interaction         10000 non-null  object 
 4   UID                 10000 non-null  object 
 5   City                10000 non-null  object 
 6   State               10000 non-null  object 
 7   County              10000 non-null  object 
 8   Zip                 10000 non-null  int64  
 9   Lat                 10000 non-null  float64
 10  Lng                 10000 non-null  float64
 11  Population          10000 non-null  int64  
 12  Area                10000 non-null  object 
 13  Timezone            10000 non-null  object 
 14  Job                 10000 non-null  object 
 15  Children            10000 non-null  float64
 16  Age                 10000 non-null  float64
 17  Education           10000 non-null  object 
 18  Employment          10000 non-null  object 
 19  Income              10000 non-null  float64
 20  Marital             10000 non-null  object 
 21  Gender              10000 non-null  object 
 22  ReAdmis             10000 non-null  object 
 23  VitD_levels         10000 non-null  float64
 24  Doc_visits          10000 non-null  int64  
 25  Full_meals_eaten    10000 non-null  int64  
 26  VitD_supp           10000 non-null  int64  
 27  Soft_drink          10000 non-null  object 
 28  Initial_admin       10000 non-null  object 
 29  HighBlood           10000 non-null  object 
 30  Stroke              10000 non-null  object 
 31  Complication_risk   10000 non-null  object 
 32  Overweight          10000 non-null  float64
 33  Arthritis           10000 non-null  object 
 34  Diabetes            10000 non-null  object 
 35  Hyperlipidemia      10000 non-null  object 
 36  BackPain            10000 non-null  object 
 37  Anxiety             10000 non-null  float64
 38  Allergic_rhinitis   10000 non-null  object 
 39  Reflux_esophagitis  10000 non-null  object 
 40  Asthma              10000 non-null  object 
 41  Services            10000 non-null  object 
 42  Initial_days        10000 non-null  float64
 43  TotalCharge         10000 non-null  float64
 44  Additional_charges  10000 non-null  float64
 45  Item1               10000 non-null  int64  
 46  Item2               10000 non-null  int64  
 47  Item3               10000 non-null  int64  
 48  Item4               10000 non-null  int64  
 49  Item5               10000 non-null  int64  
 50  Item6               10000 non-null  int64  
 51  Item7                 10000 non-null  int64
 52  Item8                 10000 non-null  int64
dtypes: float64(11), int64(15), object(27)
memory usage: 4.0+ MB
```

In [45]:

```
boxplot=sns.boxplot(x='Lng',data=df)
```



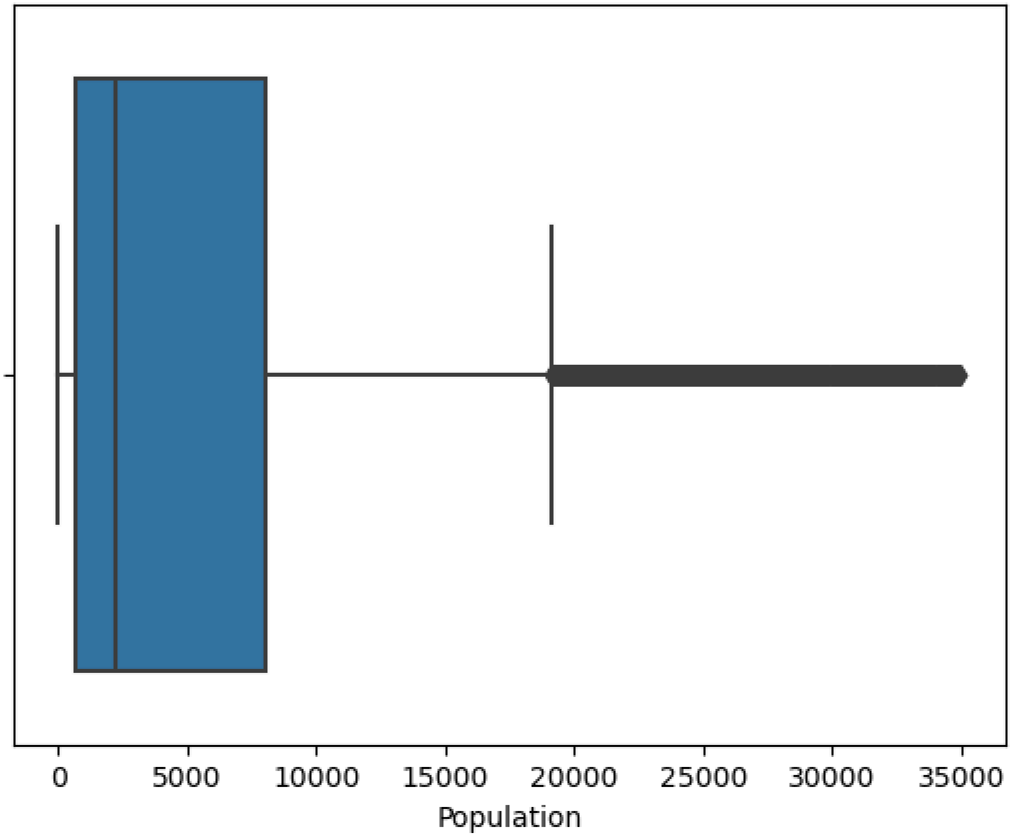
In [46]:

```
#impute outliers with the median for Population
df['Population'] = np.where(df['Population'] > 35000, np.nan , df['Population'])
df['Population'].fillna(df['Population'].median(), inplace = True)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 53 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Unnamed: 0          10000 non-null  int64  
 1   CaseOrder           10000 non-null  int64  
 2   Customer_id         10000 non-null  object 
 3   Interaction         10000 non-null  object 
 4   UID                 10000 non-null  object 
 5   City                10000 non-null  object 
 6   State               10000 non-null  object 
 7   County              10000 non-null  object 
 8   Zip                 10000 non-null  int64  
 9   Lat                 10000 non-null  float64
 10  Lng                 10000 non-null  float64
 11  Population          10000 non-null  float64
 12  Area                10000 non-null  object 
 13  Timezone            10000 non-null  object 
 14  Job                 10000 non-null  object 
 15  Children            10000 non-null  float64
 16  Age                 10000 non-null  float64
 17  Education           10000 non-null  object 
 18  Employment          10000 non-null  object 
 19  Income              10000 non-null  float64
 20  Marital             10000 non-null  object 
 21  Gender              10000 non-null  object 
 22  ReAdmis             10000 non-null  object 
 23  VitD_levels         10000 non-null  float64
 24  Doc_visits          10000 non-null  int64  
 25  Full_meals_eaten    10000 non-null  int64  
 26  VitD_supp           10000 non-null  int64  
 27  Soft_drink          10000 non-null  object 
 28  Initial_admin       10000 non-null  object 
 29  HighBlood           10000 non-null  object 
 30  Stroke              10000 non-null  object 
 31  Complication_risk   10000 non-null  object 
 32  Overweight          10000 non-null  float64
 33  Arthritis           10000 non-null  object 
 34  Diabetes            10000 non-null  object 
 35  Hyperlipidemia      10000 non-null  object 
 36  BackPain            10000 non-null  object 
 37  Anxiety             10000 non-null  float64
 38  Allergic_rhinitis   10000 non-null  object 
 39  Reflux_esophagitis  10000 non-null  object 
 40  Asthma              10000 non-null  object 
 41  Services            10000 non-null  object 
 42  Initial_days        10000 non-null  float64
 43  TotalCharge         10000 non-null  float64
 44  Additional_charges  10000 non-null  float64
 45  Item1               10000 non-null  int64  
 46  Item2               10000 non-null  int64  
 47  Item3               10000 non-null  int64  
 48  Item4               10000 non-null  int64  
 49  Item5               10000 non-null  int64  
 50  Item6               10000 non-null  int64  
 51  Item7               10000 non-null  int64
 52  Item8               10000 non-null  int64
dtypes: float64(12), int64(14), object(27)
memory usage: 4.0+ MB
```

In [47]:

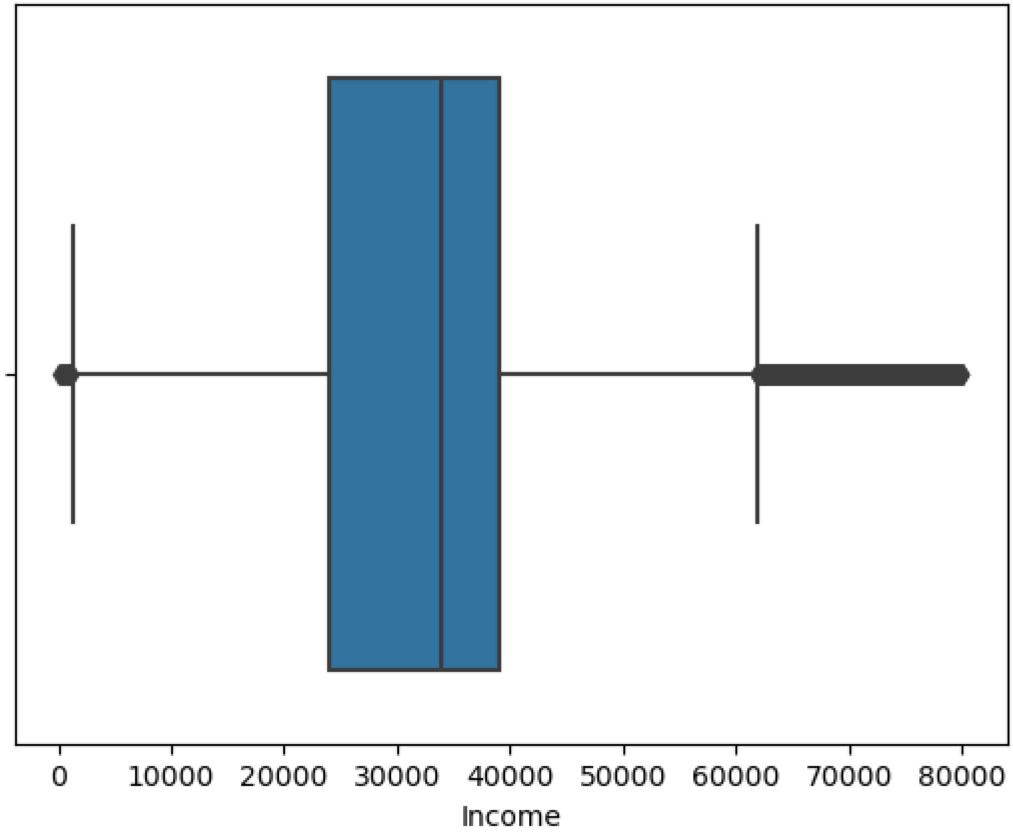
```
boxplot=sns.boxplot(x='Population',data=df)
```



In [48]:

```
#impute outliers with the median for Income
df['Income'] = np.where(df['Income'] > 80000, np.nan , df['Income'])
df['Income'].fillna(df['Income'].median(), inplace = True)
df.info()
boxplot=sns.boxplot(x='Income',data=df)
```

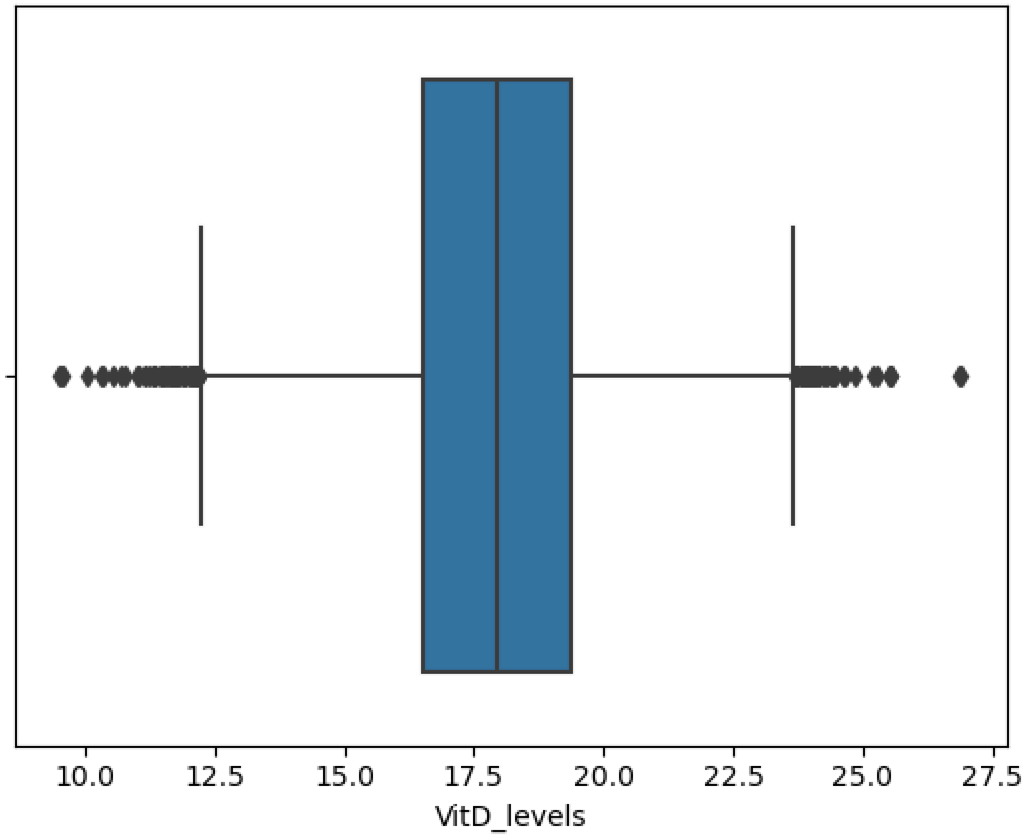
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 53 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Unnamed: 0          10000 non-null  int64  
 1   CaseOrder           10000 non-null  int64  
 2   Customer_id         10000 non-null  object 
 3   Interaction         10000 non-null  object 
 4   UID                 10000 non-null  object 
 5   City                10000 non-null  object 
 6   State               10000 non-null  object 
 7   County              10000 non-null  object 
 8   Zip                 10000 non-null  int64  
 9   Lat                 10000 non-null  float64
 10  Lng                 10000 non-null  float64
 11  Population          10000 non-null  float64
 12  Area                10000 non-null  object 
 13  Timezone            10000 non-null  object 
 14  Job                 10000 non-null  object 
 15  Children            10000 non-null  float64
 16  Age                 10000 non-null  float64
 17  Education           10000 non-null  object 
 18  Employment          10000 non-null  object 
 19  Income              10000 non-null  float64
 20  Marital             10000 non-null  object 
 21  Gender              10000 non-null  object 
 22  ReAdmis             10000 non-null  object 
 23  VitD_levels         10000 non-null  float64
 24  Doc_visits          10000 non-null  int64  
 25  Full_meals_eaten    10000 non-null  int64  
 26  VitD_supp           10000 non-null  int64  
 27  Soft_drink          10000 non-null  object 
 28  Initial_admin       10000 non-null  object 
 29  HighBlood           10000 non-null  object 
 30  Stroke              10000 non-null  object 
 31  Complication_risk   10000 non-null  object 
 32  Overweight          10000 non-null  float64
 33  Arthritis           10000 non-null  object 
 34  Diabetes            10000 non-null  object 
 35  Hyperlipidemia      10000 non-null  object 
 36  BackPain            10000 non-null  object 
 37  Anxiety             10000 non-null  float64
 38  Allergic_rhinitis   10000 non-null  object 
 39  Reflux_esophagitis  10000 non-null  object 
 40  Asthma              10000 non-null  object 
 41  Services            10000 non-null  object 
 42  Initial_days        10000 non-null  float64
 43  TotalCharge         10000 non-null  float64
 44  Additional_charges  10000 non-null  float64
 45  Item1               10000 non-null  int64  
 46  Item2               10000 non-null  int64  
 47  Item3               10000 non-null  int64  
 48  Item4               10000 non-null  int64  
 49  Item5               10000 non-null  int64  
 50  Item6               10000 non-null  int64  
 51  Item7                10000 non-null  int64  
 52  Item8                10000 non-null  int64  
dtypes: float64(12), int64(14), object(27)
memory usage: 4.0+ MB
```



In [49]:

```
#impute outliers with the median for Vitamin D levels
df['VitD_levels'] = np.where(df['VitD_levels'] > 30, np.nan , df['VitD_levels'])
df['VitD_levels'].fillna(df['VitD_levels'].median(), inplace = True)
df.info()
boxplot=sns.boxplot(x='VitD_levels',data=df)
```

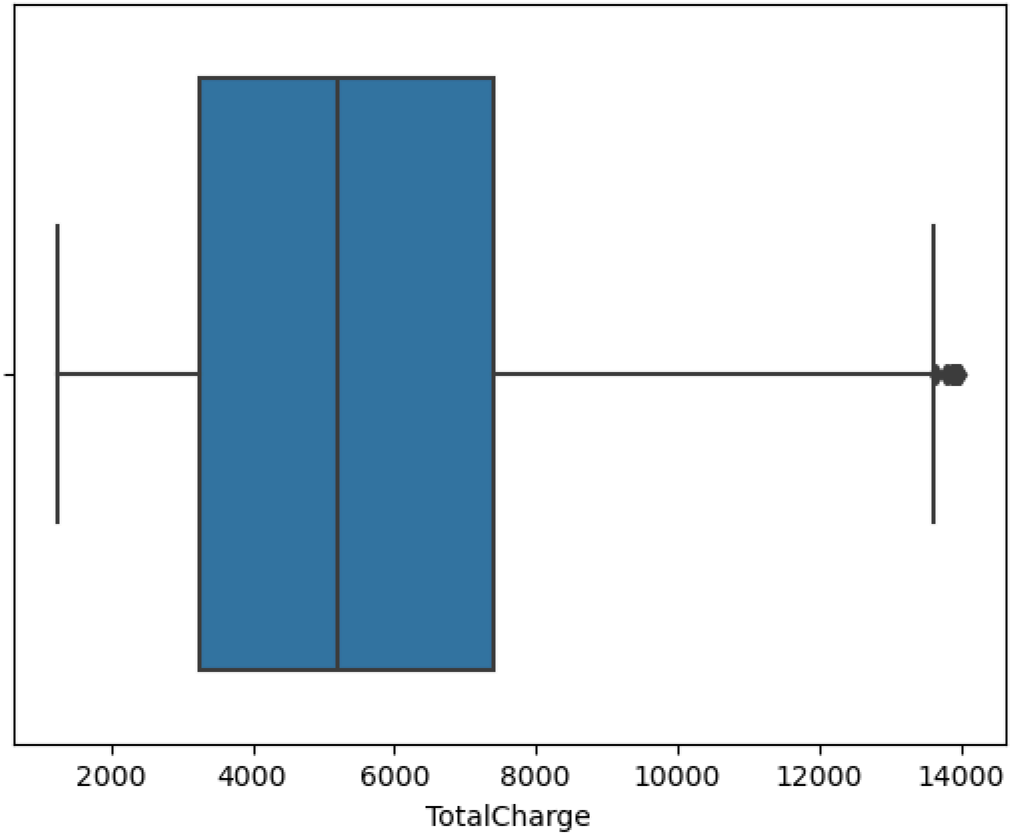
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 53 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Unnamed: 0          10000 non-null  int64  
 1   CaseOrder           10000 non-null  int64  
 2   Customer_id         10000 non-null  object 
 3   Interaction         10000 non-null  object 
 4   UID                 10000 non-null  object 
 5   City                10000 non-null  object 
 6   State               10000 non-null  object 
 7   County              10000 non-null  object 
 8   Zip                 10000 non-null  int64  
 9   Lat                 10000 non-null  float64
 10  Lng                 10000 non-null  float64
 11  Population          10000 non-null  float64
 12  Area                10000 non-null  object 
 13  Timezone            10000 non-null  object 
 14  Job                 10000 non-null  object 
 15  Children            10000 non-null  float64
 16  Age                 10000 non-null  float64
 17  Education           10000 non-null  object 
 18  Employment          10000 non-null  object 
 19  Income              10000 non-null  float64
 20  Marital             10000 non-null  object 
 21  Gender              10000 non-null  object 
 22  ReAdmis             10000 non-null  object 
 23  VitD_levels         10000 non-null  float64
 24  Doc_visits          10000 non-null  int64  
 25  Full_meals_eaten    10000 non-null  int64  
 26  VitD_supp           10000 non-null  int64  
 27  Soft_drink          10000 non-null  object 
 28  Initial_admin       10000 non-null  object 
 29  HighBlood           10000 non-null  object 
 30  Stroke              10000 non-null  object 
 31  Complication_risk   10000 non-null  object 
 32  Overweight          10000 non-null  float64
 33  Arthritis           10000 non-null  object 
 34  Diabetes            10000 non-null  object 
 35  Hyperlipidemia      10000 non-null  object 
 36  BackPain            10000 non-null  object 
 37  Anxiety             10000 non-null  float64
 38  Allergic_rhinitis   10000 non-null  object 
 39  Reflux_esophagitis  10000 non-null  object 
 40  Asthma              10000 non-null  object 
 41  Services            10000 non-null  object 
 42  Initial_days        10000 non-null  float64
 43  TotalCharge         10000 non-null  float64
 44  Additional_charges  10000 non-null  float64
 45  Item1               10000 non-null  int64  
 46  Item2               10000 non-null  int64  
 47  Item3               10000 non-null  int64  
 48  Item4               10000 non-null  int64  
 49  Item5               10000 non-null  int64  
 50  Item6               10000 non-null  int64  
 51  Item7                10000 non-null  int64  
 52  Item8                10000 non-null  int64  
dtypes: float64(12), int64(14), object(27)
memory usage: 4.0+ MB
```



In [50]:

```
#impute outliers with the median for TotalCharge
df['TotalCharge'] = np.where(df['TotalCharge'] > 14000, np.nan , df['TotalCharge'])
df['TotalCharge'].fillna(df['TotalCharge'].median(), inplace = True)
df.info()
boxplot=sns.boxplot(x='TotalCharge',data=df)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 53 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Unnamed: 0          10000 non-null  int64
 1   CaseOrder           10000 non-null  int64
 2   Customer_id         10000 non-null  object
 3   Interaction         10000 non-null  object
 4   UID                 10000 non-null  object
 5   City                10000 non-null  object
 6   State               10000 non-null  object
 7   County              10000 non-null  object
 8   Zip                 10000 non-null  int64
 9   Lat                 10000 non-null  float64
 10  Lng                 10000 non-null  float64
 11  Population          10000 non-null  float64
 12  Area                10000 non-null  object
 13  Timezone            10000 non-null  object
 14  Job                 10000 non-null  object
 15  Children            10000 non-null  float64
 16  Age                 10000 non-null  float64
 17  Education           10000 non-null  object
 18  Employment          10000 non-null  object
 19  Income              10000 non-null  float64
 20  Marital             10000 non-null  object
 21  Gender              10000 non-null  object
 22  ReAdmis             10000 non-null  object
 23  VitD_levels         10000 non-null  float64
 24  Doc_visits          10000 non-null  int64
 25  Full_meals_eaten    10000 non-null  int64
 26  VitD_supp           10000 non-null  int64
 27  Soft_drink          10000 non-null  object
 28  Initial_admin       10000 non-null  object
 29  HighBlood           10000 non-null  object
 30  Stroke              10000 non-null  object
 31  Complication_risk   10000 non-null  object
 32  Overweight          10000 non-null  float64
 33  Arthritis           10000 non-null  object
 34  Diabetes            10000 non-null  object
 35  Hyperlipidemia      10000 non-null  object
 36  BackPain            10000 non-null  object
 37  Anxiety             10000 non-null  float64
 38  Allergic_rhinitis   10000 non-null  object
 39  Reflux_esophagitis  10000 non-null  object
 40  Asthma              10000 non-null  object
 41  Services            10000 non-null  object
 42  Initial_days        10000 non-null  float64
 43  TotalCharge         10000 non-null  float64
 44  Additional_charges  10000 non-null  float64
 45  Item1               10000 non-null  int64
 46  Item2               10000 non-null  int64
 47  Item3               10000 non-null  int64
 48  Item4               10000 non-null  int64
 49  Item5               10000 non-null  int64
 50  Item6               10000 non-null  int64
 51  Item7               10000 non-null  int64
 52  Item8               10000 non-null  int64
dtypes: float64(12), int64(14), object(27)
memory usage: 4.0+ MB
```



In [51]:

```
#change boolean categorical to numeric
# create an array of all the variables that are needed to be converted
#create a dictionary for converting the values
#create a for loop changing all of them to numeric
var_cat = ['HighBlood', 'Stroke', 'Arthritis', 'Diabetes', 'Hyperlipidemia', 'BackP
dict_var = {'numeric':{'No':0, 'Yes':1}}
for i in var_cat:
    df['numeric'] = df[i]
    df.replace(dict_var, inplace = True)
```

In [52]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 54 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Unnamed: 0          10000 non-null  int64  
 1   CaseOrder           10000 non-null  int64  
 2   Customer_id         10000 non-null  object 
 3   Interaction         10000 non-null  object 
 4   UID                 10000 non-null  object 
 5   City                10000 non-null  object 
 6   State               10000 non-null  object 
 7   County              10000 non-null  object 
 8   Zip                 10000 non-null  int64  
 9   Lat                 10000 non-null  float64
 10  Lng                 10000 non-null  float64
 11  Population          10000 non-null  float64
 12  Area                10000 non-null  object 
 13  Timezone            10000 non-null  object 
 14  Job                 10000 non-null  object 
 15  Children            10000 non-null  float64
 16  Age                 10000 non-null  float64
 17  Education           10000 non-null  object 
 18  Employment          10000 non-null  object 
 19  Income              10000 non-null  float64
 20  Marital             10000 non-null  object 
 21  Gender              10000 non-null  object 
 22  ReAdmis             10000 non-null  object 
 23  VitD_levels         10000 non-null  float64
 24  Doc_visits          10000 non-null  int64  
 25  Full_meals_eaten    10000 non-null  int64  
 26  VitD_supp           10000 non-null  int64  
 27  Soft_drink          10000 non-null  object 
 28  Initial_admin       10000 non-null  object 
 29  HighBlood           10000 non-null  object 
 30  Stroke              10000 non-null  object 
 31  Complication_risk   10000 non-null  object 
 32  Overweight          10000 non-null  float64
 33  Arthritis           10000 non-null  object 
 34  Diabetes            10000 non-null  object 
 35  Hyperlipidemia      10000 non-null  object 
 36  BackPain            10000 non-null  object 
 37  Anxiety             10000 non-null  float64
 38  Allergic_rhinitis   10000 non-null  object 
 39  Reflux_esophagitis  10000 non-null  object 
 40  Asthma              10000 non-null  object 
 41  Services            10000 non-null  object 
 42  Initial_days        10000 non-null  float64
 43  TotalCharge         10000 non-null  float64
 44  Additional_charges  10000 non-null  float64
 45  Item1               10000 non-null  int64  
 46  Item2               10000 non-null  int64  
 47  Item3               10000 non-null  int64  
 48  Item4               10000 non-null  int64  
 49  Item5               10000 non-null  int64  
 50  Item6               10000 non-null  int64  
 51  Item7                 10000 non-null  int64
 52  Item8                 10000 non-null  int64
 53  numeric               10000 non-null  int64
dtypes: float64(12), int64(15), object(27)
memory usage: 4.1+ MB
```

In [53]:

```
#Check if it worked
df['Stroke']
```

Out[53]:

```
0        No
1        No
2        No
3       Yes
4        No
       ... 
9995     No
9996     No
9997     No
9998     No
9999     No
Name: Stroke, Length: 10000, dtype: object
```

In [54]:

```
#round the VitD_levels, Initial_days, TotalCharge, and Additional_charge to 2 place
df = df.round({'VitD_levels': 2, 'Initial_days': 2,'TotalCharge': 2,'Additional_cha
```

In [55]:

```
df['TotalCharge']
```

Out[55]:

```
0       3191.05
1       4214.91
2       2177.59
3       2465.12
4       1885.66
         ...   
9995    6651.24
9996    7851.52
9997    7725.95
9998    8462.83
9999    8700.86
Name: TotalCharge, Length: 10000, dtype: float64
```

In [ ]:

```
#steps taken for PCA
#define features/variables for PCA
#Normalize data and apply PCA
#PCA loadings
#selecting PCs
```

In [56]:

```
#selecting continuous variables
pca_col = df[['Lat', 'Lng', 'Population', 'Children', 'Age', 'Income', 'VitD_levels
```

In [57]:

```
#Normalize data
pca_normalized = (pca_col - pca_col.mean())/pca_col.std()
```

In [58]:

```
#Applying PCA
pca = PCA(n_components=pca_col.shape[1])
```

In [59]:
```
print(pca)
```

```
PCA(n_components=13)
```

In [60]:
```
pca.fit(pca_normalized)
```

Out[60]:
```
▾ PCA
PCA(n_components=13)
```

In [61]:
```
df_pca = pd.DataFrame(pca.transform(pca_normalized),
columns = ['PC1','PC2','PC3','PC4','PC5','PC6','PC7','PC8','PC9','PC10','PC11','PC1
```

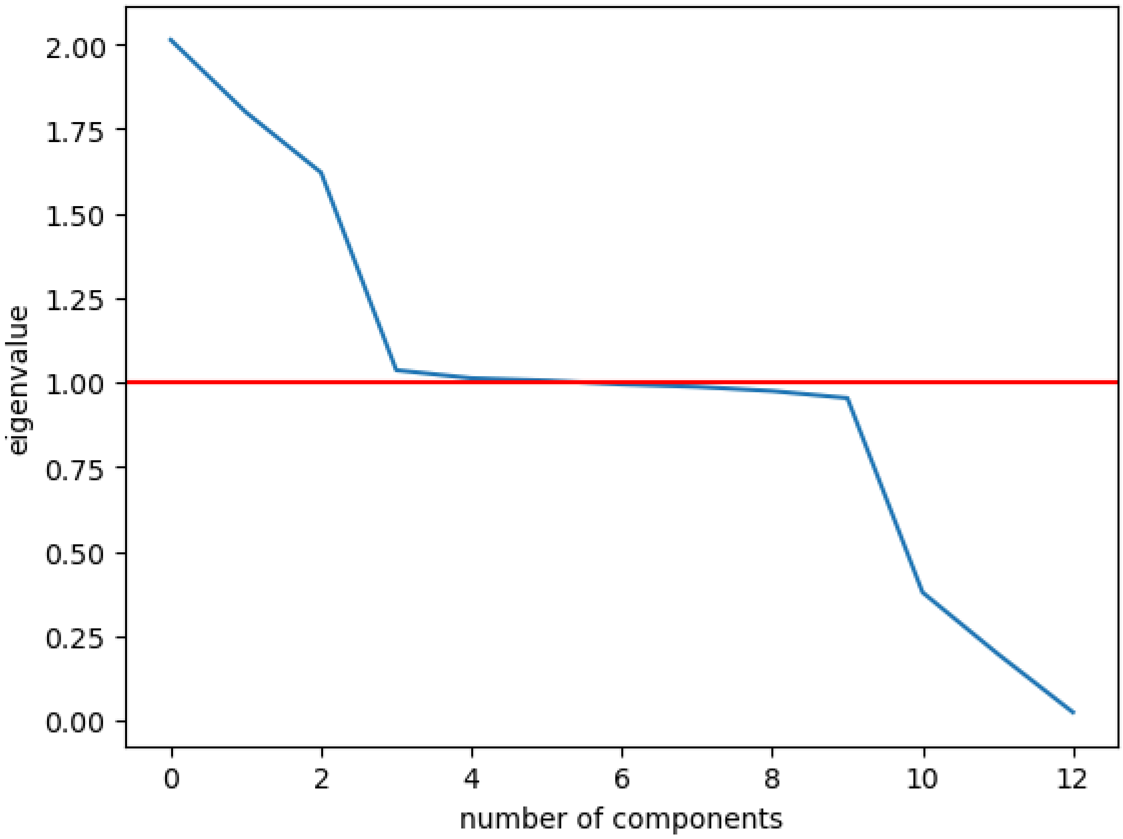
In [62]:
```
#PCA Loadings
loadings = pd.DataFrame(pca.components_.T,
                        columns= ['PC1','PC2','PC3','PC4','PC5','PC6','PC7','PC8','
                        index = pca_col.columns)
loadings
```

Out[62]:

| | PC1 | PC2 | PC3 | PC4 | PC5 | PC6 | P |
|---|---|---|---|---|---|---|---|
| **Lat** | -0.688881 | 0.090087 | 0.005902 | -0.020168 | 0.011733 | 0.028989 | -0.0528 |
| **Lng** | -0.689853 | 0.091700 | 0.007406 | -0.017607 | 0.008570 | 0.029425 | -0.0470 |
| **Population** | 0.181661 | -0.003183 | -0.025145 | -0.169625 | 0.043939 | 0.193604 | -0.3712 |
| **Children** | 0.007908 | 0.005243 | 0.011403 | 0.123882 | 0.032642 | 0.890027 | -0.2674 |
| **Age** | 0.018017 | 0.076730 | 0.701364 | -0.009192 | 0.038900 | -0.008535 | -0.0059 |
| **Income** | 0.002998 | -0.001692 | 0.001013 | -0.486053 | -0.347674 | -0.117513 | -0.2362 |
| **VitD_levels** | 0.005916 | 0.048511 | 0.015771 | 0.299371 | -0.695813 | 0.230925 | 0.3652 |
| **Doc_visits** | -0.011559 | -0.009666 | 0.014245 | -0.279159 | -0.541270 | -0.084112 | -0.4302 |
| **Full_meals_eaten** | -0.004441 | -0.024515 | 0.037558 | 0.556597 | -0.286246 | -0.213568 | -0.1224 |
| **VitD_supp** | -0.001700 | 0.038639 | 0.012404 | -0.490496 | -0.106798 | 0.217701 | 0.6275 |
| **Initial_days** | 0.085627 | 0.693399 | -0.088100 | 0.002814 | 0.055428 | -0.033248 | -0.0479 |
| **TotalCharge** | 0.090274 | 0.697039 | -0.071925 | 0.024273 | -0.027377 | -0.016387 | -0.0078 |
| **Additional_charges** | 0.021820 | 0.079633 | 0.701630 | -0.017189 | 0.010465 | -0.001133 | -0.0125 |

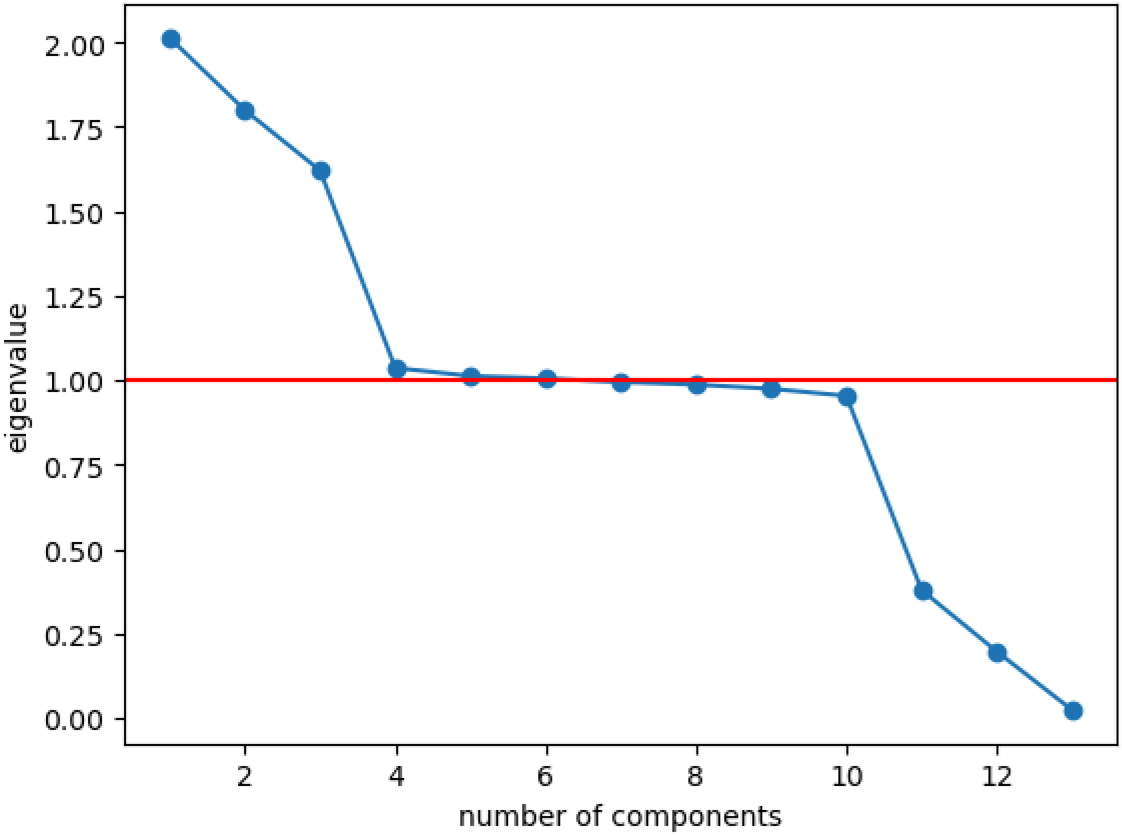
In [63]:
```
#Selecting PCs
cov_matrix = np.dot(pca_normalized.T, pca_normalized) / pca_col.shape[0]
eigenvalues = [np.dot(eigenvector.T,np.dot(cov_matrix,eigenvector)) for eigenvector

plt.plot(eigenvalues)
plt.xlabel('number of components')
plt.ylabel('eigenvalue')
plt.axhline(y=1, color = 'red')
plt.show()
```



In [64]:

```
#relabelling the axia
plt.plot(np.arange(1,len(eigenvalues)+1),eigenvalues, marker='o')
plt.xlabel('number of components')
plt.ylabel('eigenvalue')
plt.axhline(y=1, color = 'red')
plt.show()
```



```
In [65]: #exporting as csv file
         df.to_csv(r'C:\Users\arjun\OneDrive\Desktop\WGU\D206\D206csv.csv')
```

```
In [ ]:
```