Matthew Morgan
Student ID: 010471280
Data Mining - 209
Task 2: Classification Analysis
Western Governor's University

Part I: Research Question
A1. Research Question
Can I predict which patients are at risk of re-admission using a decision tree analysis so the hospital can take appropriate steps to reduce re-admissions?

A2. Goal of the Data Analysis
The primary goal of this data analysis is to develop a machine learning model using a decision tree to help the company identify patients who are at risk of re-admission.

Part II: Method Justification
B1. Explain Method from Part A1

A decision tree is a tree-like model that acts as a decision support tool. A decision tree is a sequence of if-else questions about individual features that's able to capture non-linear relationships between features and labels, and doesn't require feature scaling. This machine learning algorithm can also handle numerical and categorical data while also requiring less computational time. Because of these features, it is a good fit for my dataset.

B2. Summarize one assumption of your chosen classification model.

One assumption of the decision tree algorithm is that it is non-parametric. It doesn't make distribution assumptions on the data. Thus, decisions can deal with non-linear data, skewed data, multi-model data, and categorical, ordinal, and non-ordinal data efficiently. This characteristic makes the model easier to understand, interpret, and visualize. (Navlani, 2018)

B3. List Packages or Libraries Chosen

| Packages | Usage |
|---|---|
| Pandas | Importing data and data manipulation |
| Numpy | Provides array objects for calculations |
| Seaborn | For visualizations like correlation matrix |
| Matplotlib.pyplot | For visualizations like ROC curve |
| missingno | For visualizing missing data |
| Sklearn,preprocessing | To scale features |
| Skleanr.feature_selection | For feature selection |
| Sklearn.model_selection | For splitting data into train and test sets |
| Sklearn.pipeline | To assemble several steps that can be performed together while setting different parameters |
| sklearn.metrics | To import accuracy score and MSE |
| Scipy.stats | To run statistical calculations |
| sklearn.tree | To run the decision tree model |

In [22]:
```
# Data Analytics imports
import pandas as pd
import numpy as np

# Visualization imports
import seaborn as sns
import matplotlib.pyplot as plt
import missingno as msno

# Statistics imports
import scipy.stats as stats
from scipy.stats import skew, kurtosis
import statistics as stat

# scikit-learn imports
import sklearn
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error

# import decisiontreeclassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree

#Ignore warnings
import warnings
warnings.filterwarnings('ignore')
```

```
Part III: Data Preparation
C1. One Data Preprocessing Goal from A1
Removing white spaces, imputing missing data, converting binary (yes/no) variables into
quantitative (1/0) variables.

C2. Identify Initial Data Set Variables Used for Analysis
Variable #  Independent Variable                Data Type   Data Class
         1  Initial_days                        Continuous  Quantitative
         2  Services_CT_Scan                    Categorical Qualitative
         3  Children                            Continuous  Quantitative
         4  Services_Intravenous                Categorical Qualitative
         5  Population                          Continuous  Quantitative
         6  Initial_Admin_Emergency_admission   Categorical Qualitative


C3. Explain Each of the Steps Used to Prepare the Data for Analysis
1.  Load data into dataframe
2.  View the data to evaluate structure and types
3.  Detect null values
4.  Check for missing data
5.  Visualize data to check for outliers
6.  Convert categorical data to quantitative
7.  Rename columns from pd.get_dummies
8.  Visualize univariate stats from dataframe to ensure data quality
```

In [23]:
```
#Loading the CSV of the default dataset
df = pd.read_csv(r'C:\Users\mmorg\WGU\D209\medical_clean.csv')
```

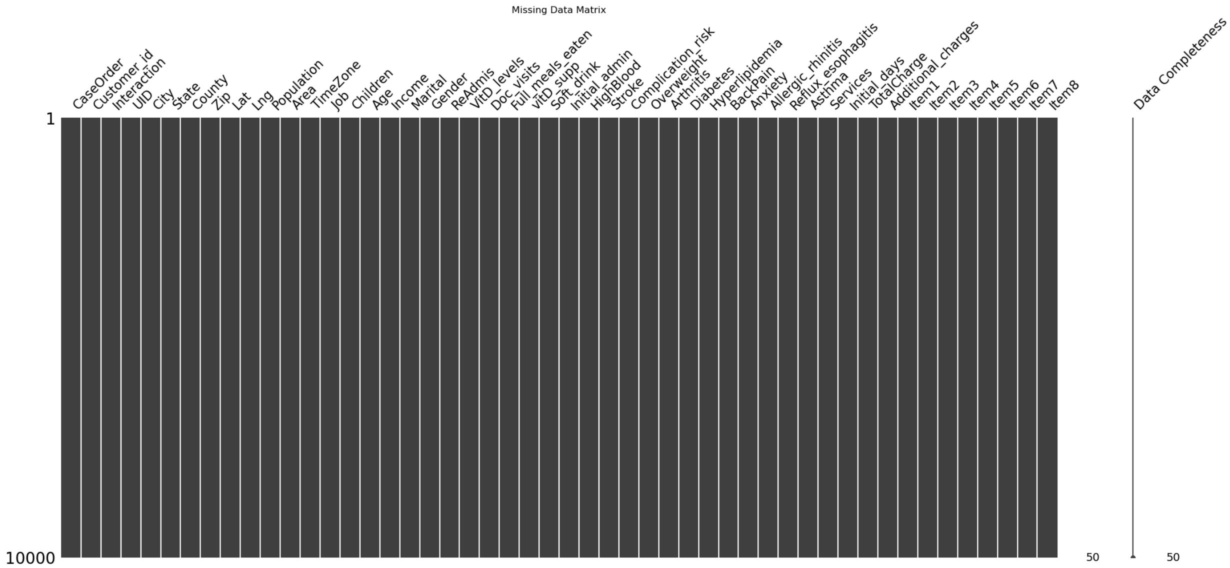
In [24]:
```
#Viewing Data to evaluate structure and types
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 50 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   CaseOrder           10000 non-null  int64  
 1   Customer_id         10000 non-null  object 
 2   Interaction         10000 non-null  object 
 3   UID                 10000 non-null  object 
 4   City                10000 non-null  object 
 5   State               10000 non-null  object 
 6   County              10000 non-null  object 
 7   Zip                 10000 non-null  int64  
 8   Lat                 10000 non-null  float64
 9   Lng                 10000 non-null  float64
 10  Population          10000 non-null  int64  
 11  Area                10000 non-null  object 
 12  TimeZone            10000 non-null  object 
 13  Job                 10000 non-null  object 
 14  Children            10000 non-null  int64  
 15  Age                 10000 non-null  int64  
 16  Income              10000 non-null  float64
 17  Marital             10000 non-null  object 
 18  Gender              10000 non-null  object 
 19  ReAdmis             10000 non-null  object 
 20  VitD_levels         10000 non-null  float64
 21  Doc_visits          10000 non-null  int64  
 22  Full_meals_eaten    10000 non-null  int64  
 23  vitD_supp           10000 non-null  int64  
 24  Soft_drink          10000 non-null  object 
 25  Initial_admin       10000 non-null  object 
 26  HighBlood           10000 non-null  object 
 27  Stroke              10000 non-null  object 
 28  Complication_risk   10000 non-null  object 
 29  Overweight          10000 non-null  object 
 30  Arthritis           10000 non-null  object 
 31  Diabetes            10000 non-null  object 
 32  Hyperlipidemia      10000 non-null  object 
 33  BackPain            10000 non-null  object 
 34  Anxiety             10000 non-null  object 
 35  Allergic_rhinitis   10000 non-null  object 
 36  Reflux_esophagitis  10000 non-null  object 
 37  Asthma              10000 non-null  object 
 38  Services            10000 non-null  object 
 39  Initial_days        10000 non-null  float64
 40  TotalCharge         10000 non-null  float64
 41  Additional_charges  10000 non-null  float64
 42  Item1               10000 non-null  int64  
 43  Item2               10000 non-null  int64  
 44  Item3               10000 non-null  int64  
 45  Item4               10000 non-null  int64  
 46  Item5               10000 non-null  int64  
 47  Item6               10000 non-null  int64  
 48  Item7               10000 non-null  int64  
 49  Item8               10000 non-null  int64  
dtypes: float64(7), int64(16), object(27)
memory usage: 3.8+ MB
```

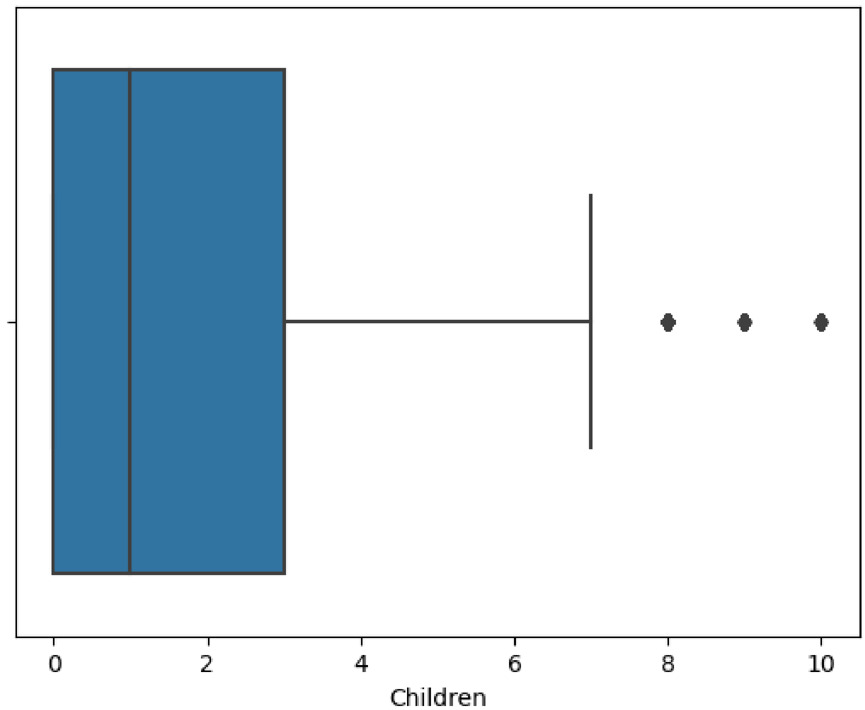
In [25]:
```
#Detect null values
print(df.isnull().sum())
```

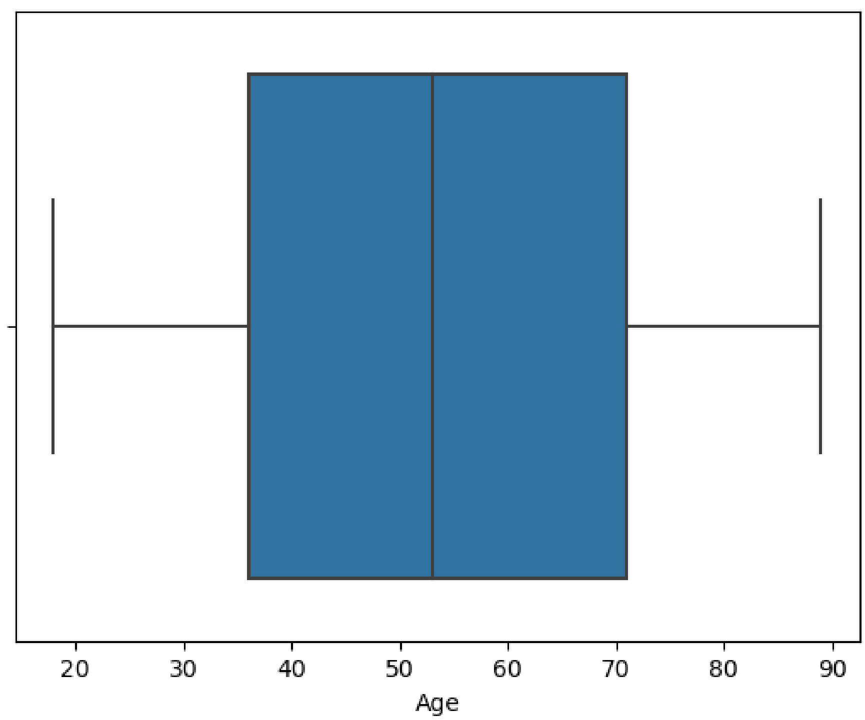
```
CaseOrder             0
Customer_id           0
Interaction           0
UID                   0
City                  0
State                 0
County                0
Zip                   0
Lat                   0
Lng                   0
Population            0
Area                  0
TimeZone              0
Job                   0
Children              0
Age                   0
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
```

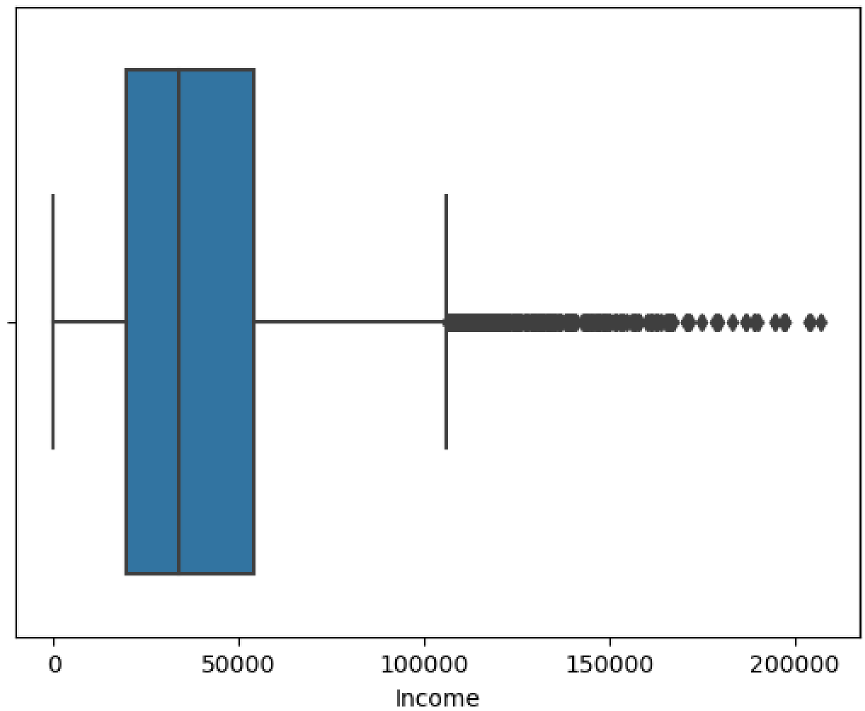
In [26]:
```
#Detect missing data
msno.matrix(df, labels=True)
plt.title('Missing Data Matrix')
plt.show()
```

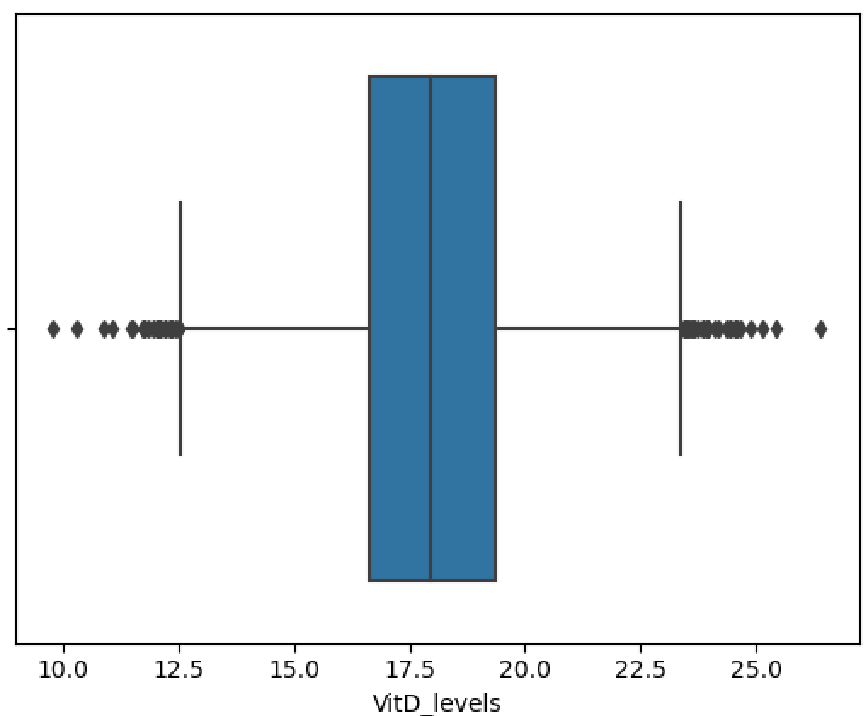


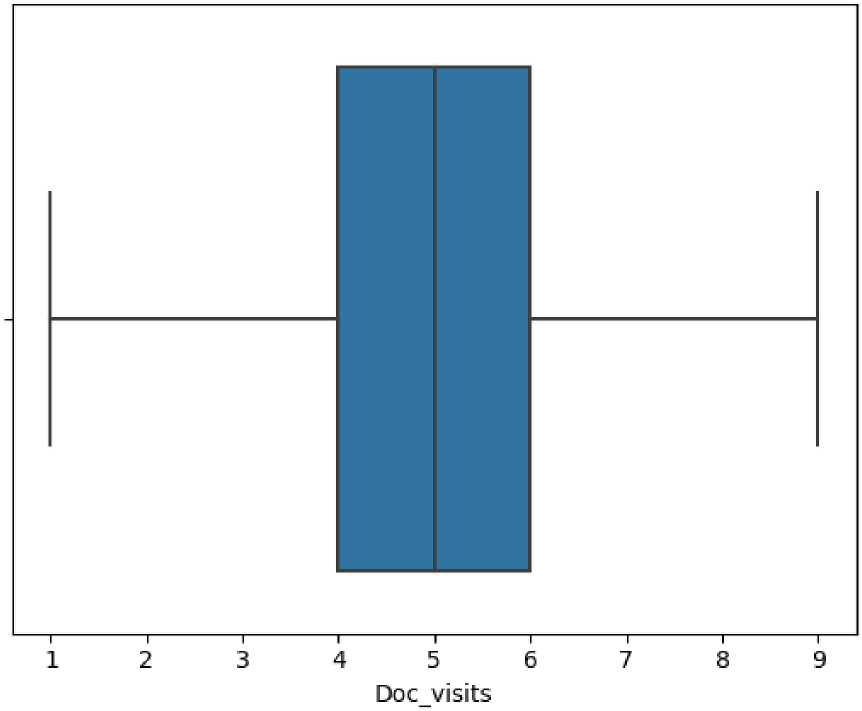
In [27]:
```
#Detection of outliers for quantitative values
boxplot=sns.boxplot(x='Children',data=df)
plt.show()
boxplot=sns.boxplot(x='Age',data=df)
plt.show()
boxplot=sns.boxplot(x='Income',data=df)
plt.show()
boxplot=sns.boxplot(x='VitD_levels',data=df)
plt.show()
boxplot=sns.boxplot(x='Doc_visits',data=df)
plt.show()
boxplot=sns.boxplot(x='Full_meals_eaten',data=df)
plt.show()
boxplot=sns.boxplot(x='vitD_supp',data=df)
plt.show()
boxplot=sns.boxplot(x='TotalCharge',data=df)
plt.show()
boxplot=sns.boxplot(x='Additional_charges',data=df)
plt.show()
boxplot=sns.boxplot(x='Item1',data=df)
plt.show()
boxplot=sns.boxplot(x='Item2',data=df)
plt.show()
boxplot=sns.boxplot(x='Item3',data=df)
plt.show()
boxplot=sns.boxplot(x='Item4',data=df)
plt.show()
boxplot=sns.boxplot(x='Item5',data=df)
plt.show()
boxplot=sns.boxplot(x='Item6',data=df)
plt.show()
boxplot=sns.boxplot(x='Item7',data=df)
plt.show()
boxplot=sns.boxplot(x='Item8',data=df)
plt.show()

#Visualizing distribution shapes
print('Children Original: ')
plt.hist(df['Children'])
plt.show()
print('Age Original: ')
plt.hist(df['Age'])
plt.show()
print('Income Original: ')
plt.hist(df['Income'])
plt.show()
print('Overweight Original: ')
plt.hist(df['Overweight'])
plt.show()
print('Anxiety Original: ')
plt.hist(df['Anxiety'])
plt.show()
print('Initial_days Original: ')
plt.hist(df['Initial_days'])
plt.show()
```

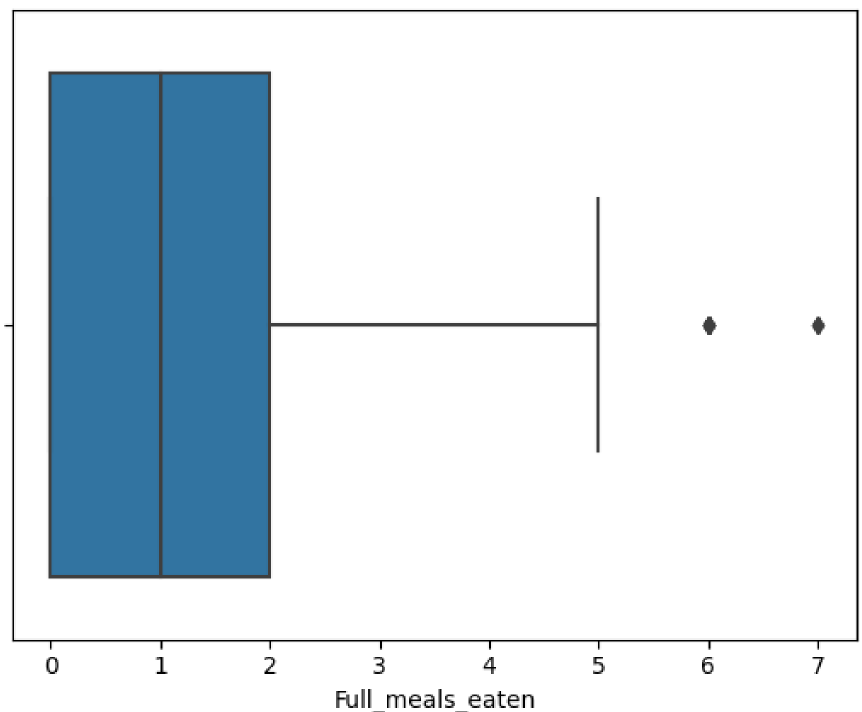

0
2
4
6
8
10
Children

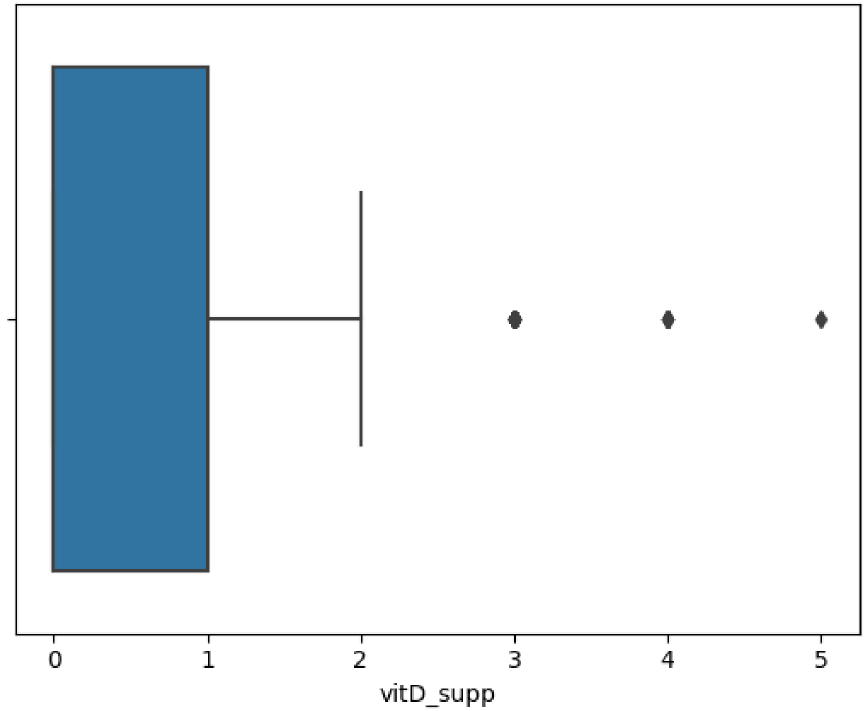

20
30
40
50
60
70
80
90
Age

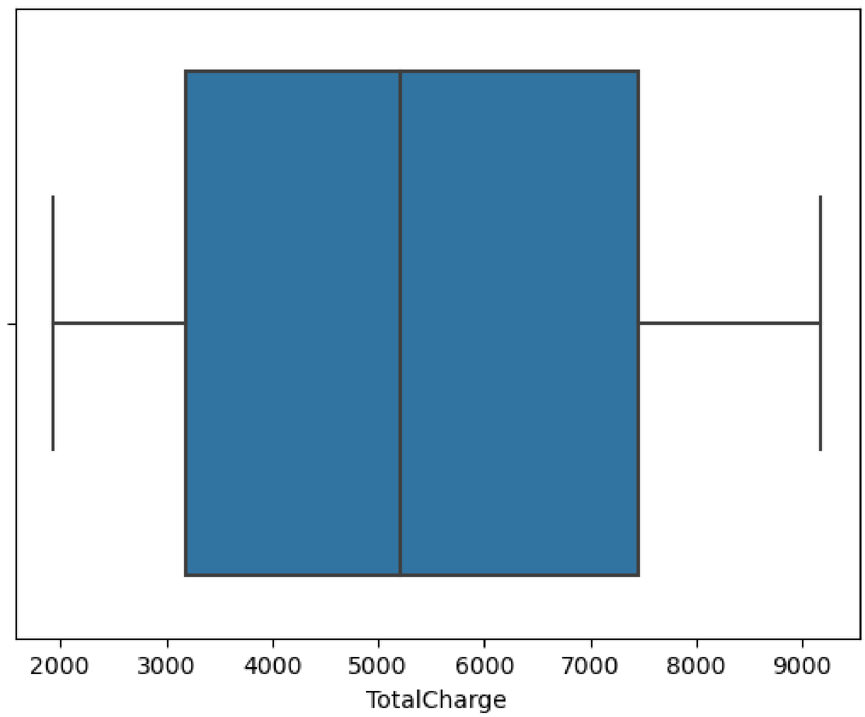

0
50000
100000
150000
200000
Income

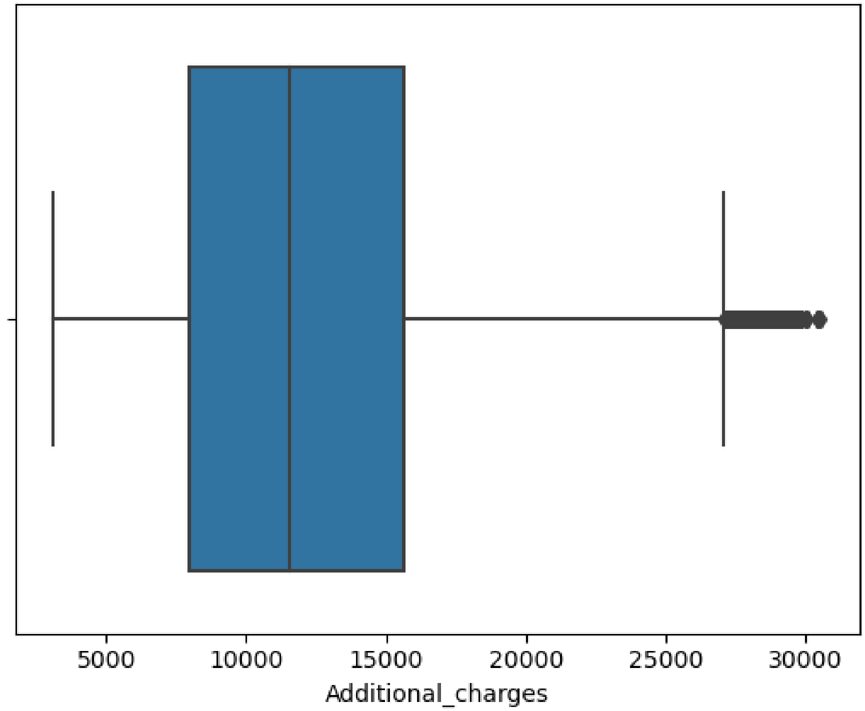

10.0
12.5
15.0
17.5
20.0
22.5
25.0
VitD_levels

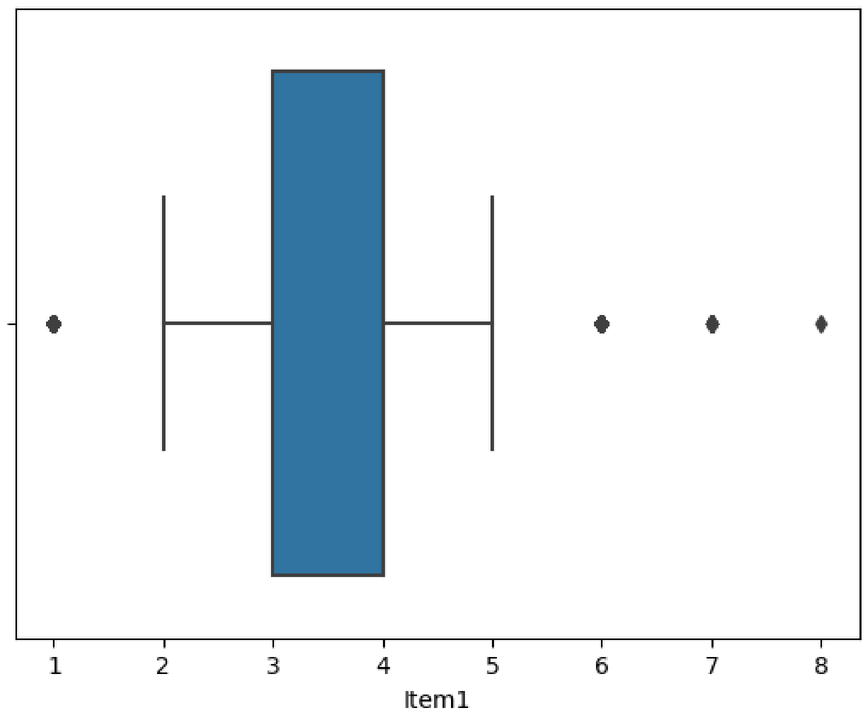

1
2
3
4
5
6
7
8
9
Doc_visits

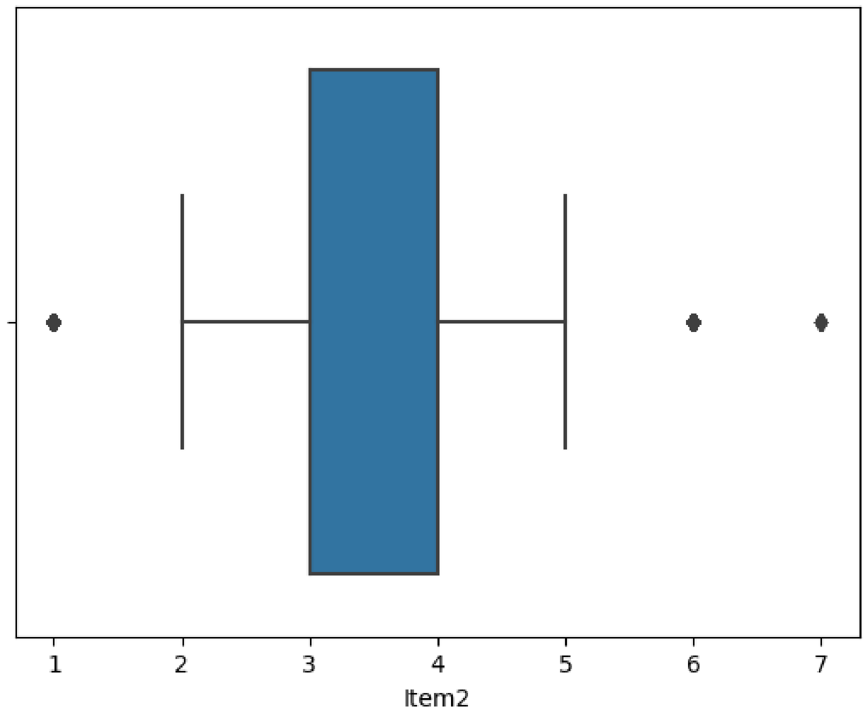

0
1
2
3
4
5
6
7
Full_meals_eaten

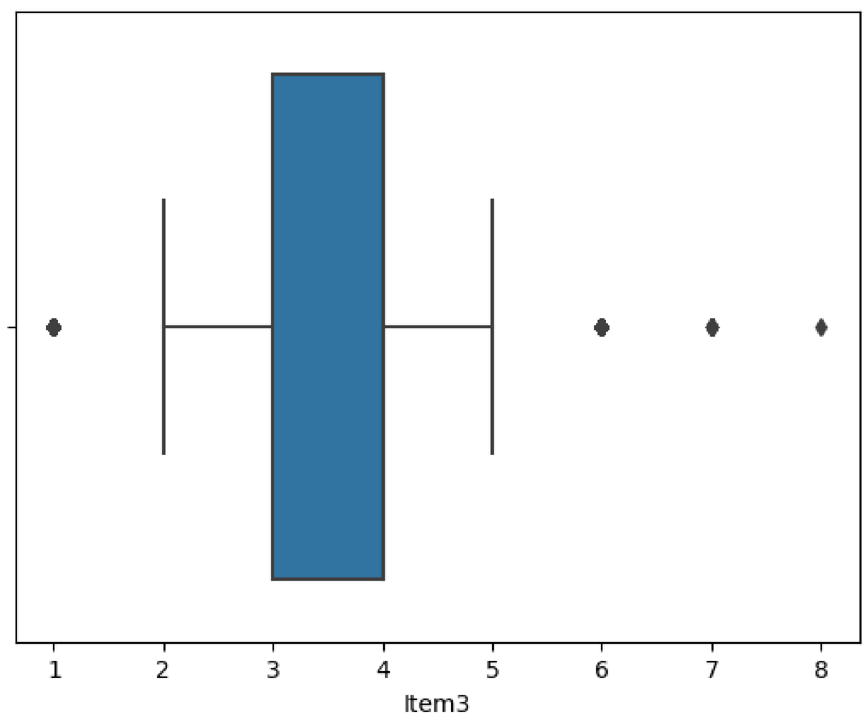

0
1
2
3
4
5
vitD_supp

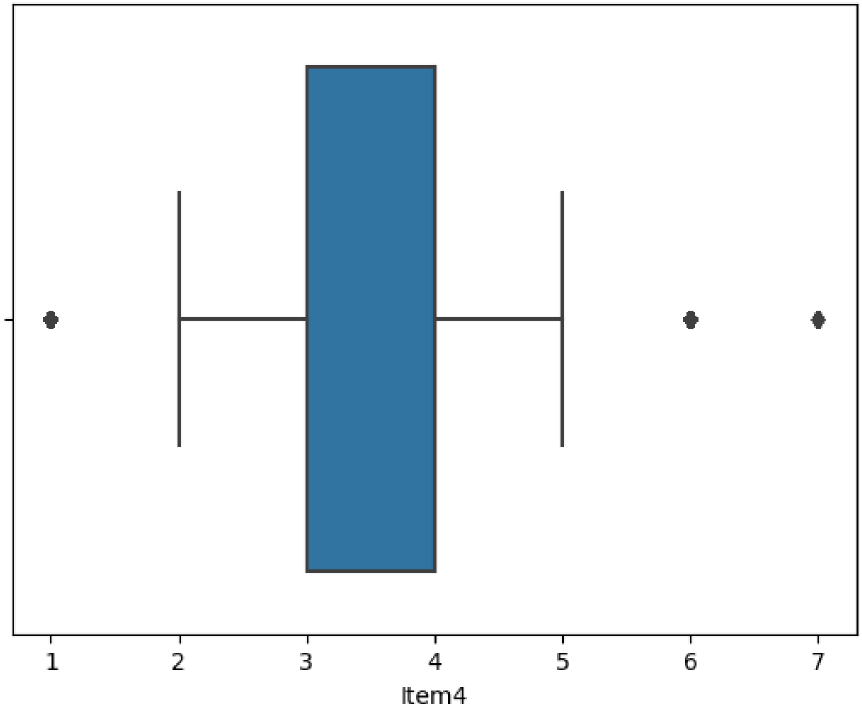

2000
3000
4000
5000
6000
7000
8000
9000
TotalCharge

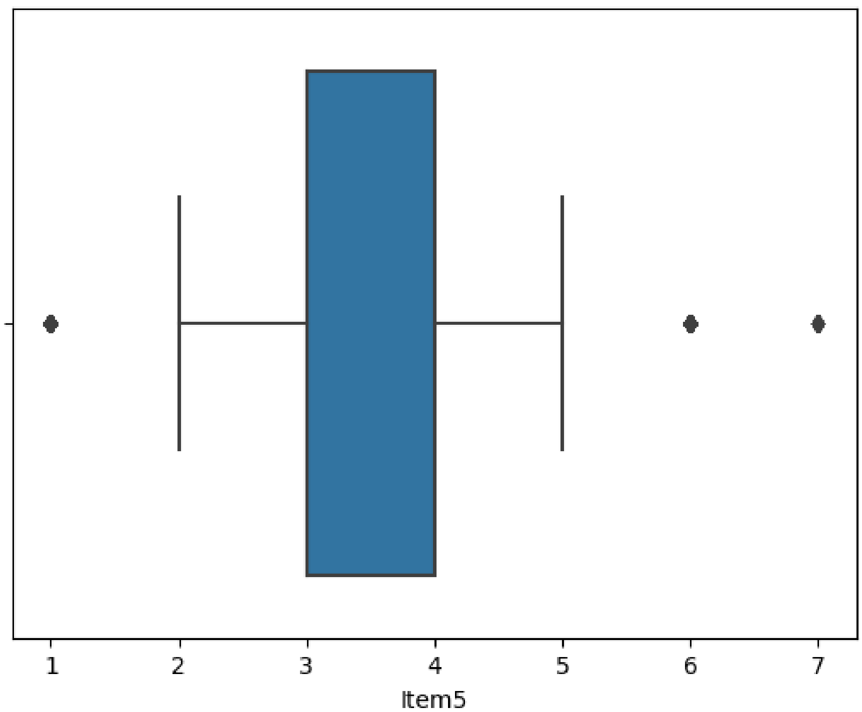

5000
10000
15000
20000
25000
30000
Additional_charges

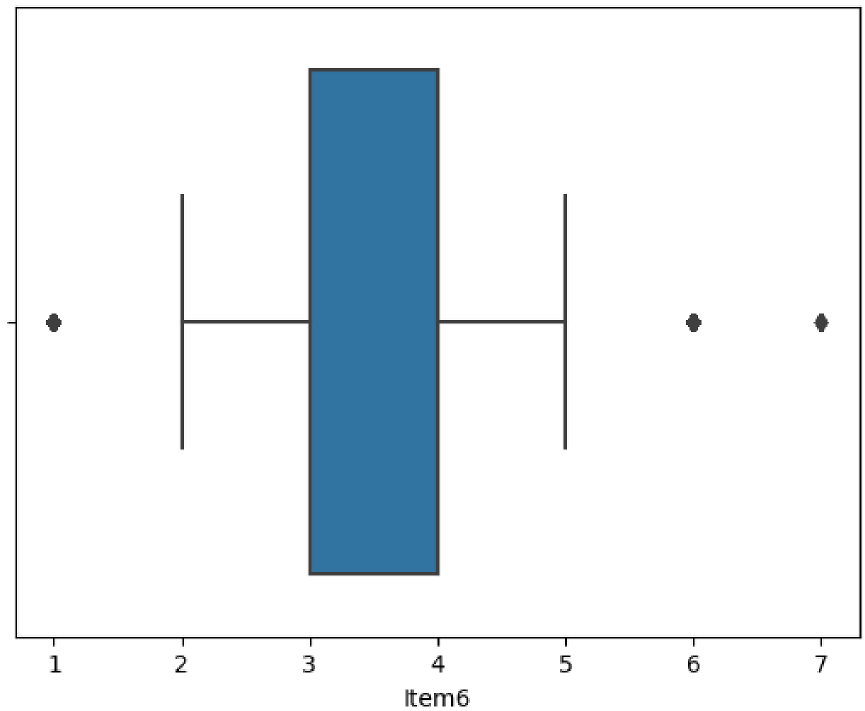

1
2
3
4
5
6
7
8
Item1

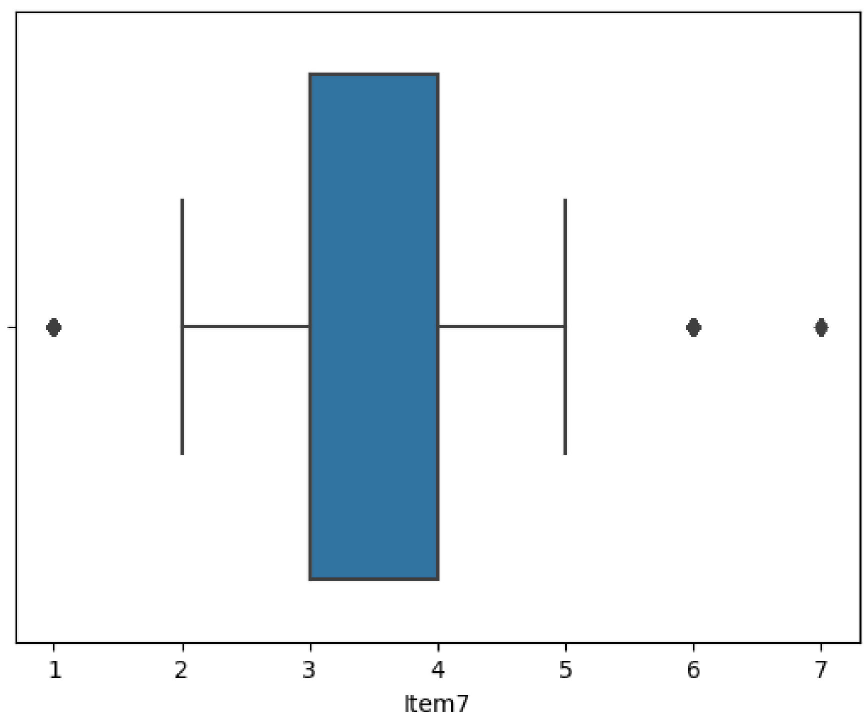

1
2
3
4
5
6
7
Item2

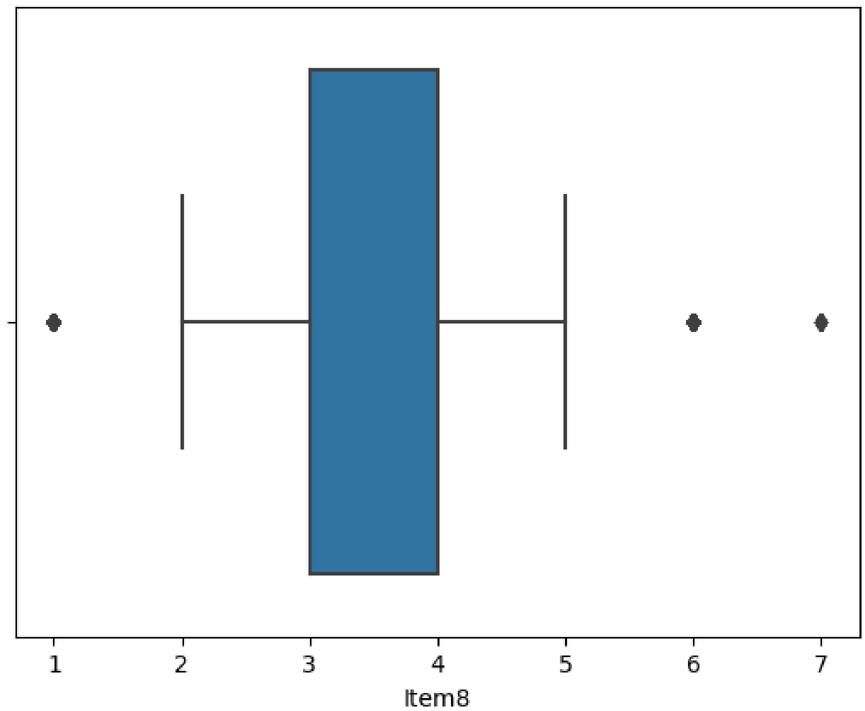

1
2
3
4
5
6
7
8
Item3


1
2
3
4
5
6
7
Item4

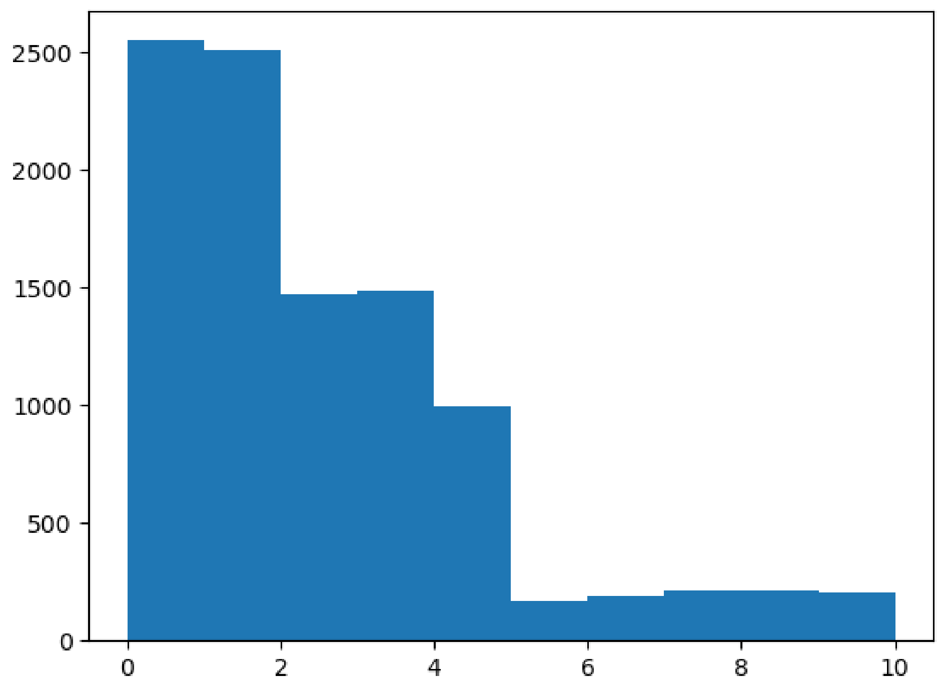

1
2
3
4
5
6
7
Item5


1
2
3
4
5
6
7
Item6

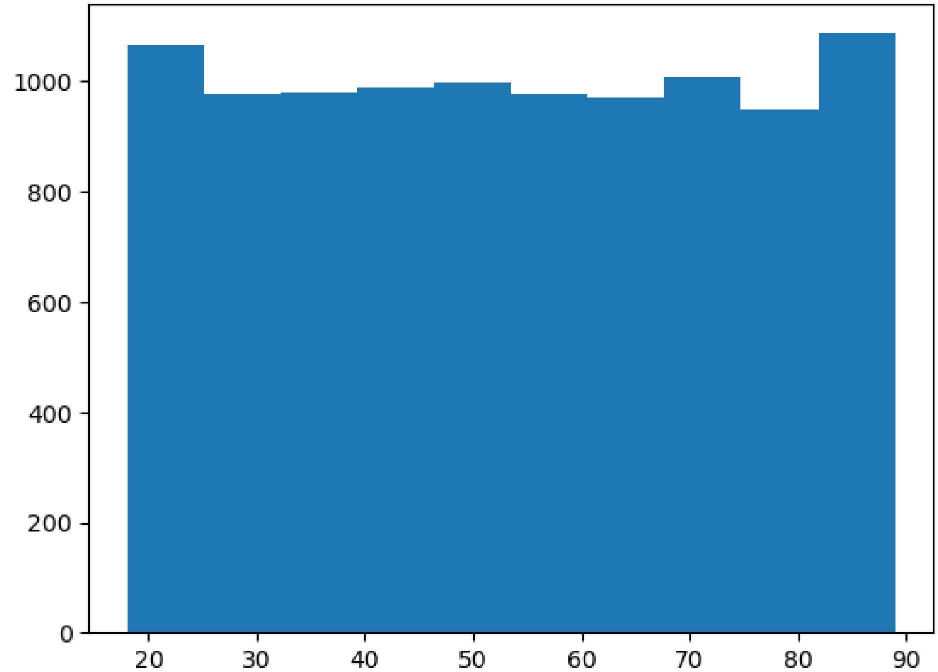

1
2
3
4
5
6
7
Item7



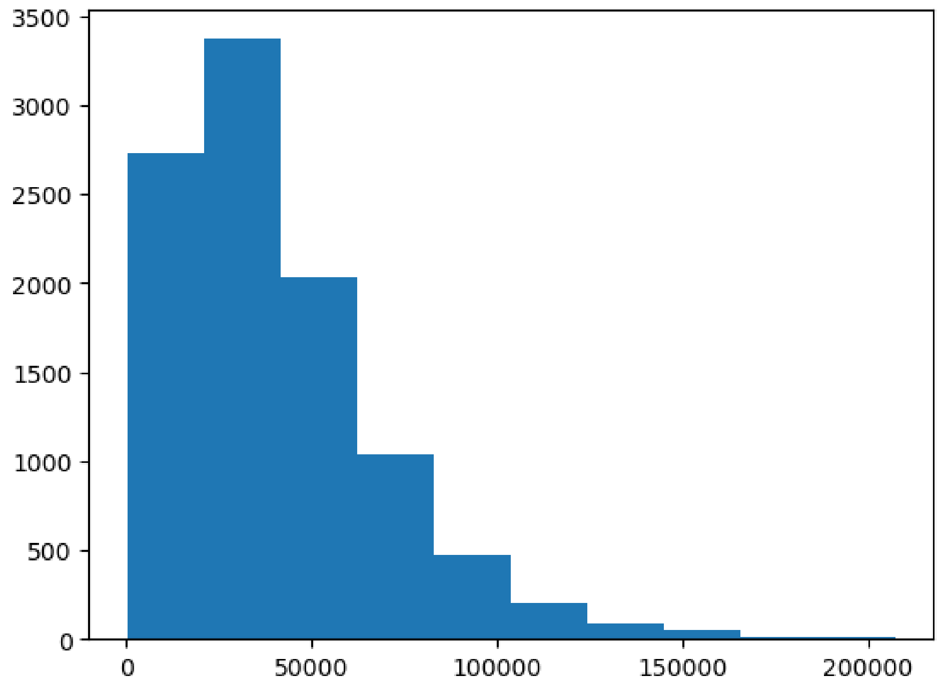

Children Original:



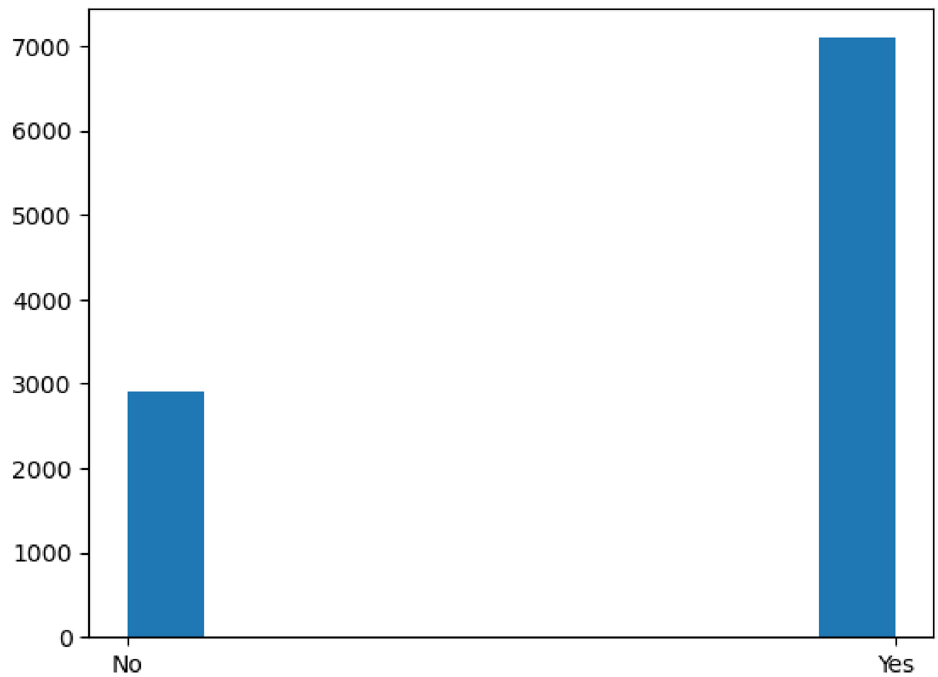

Age Original:



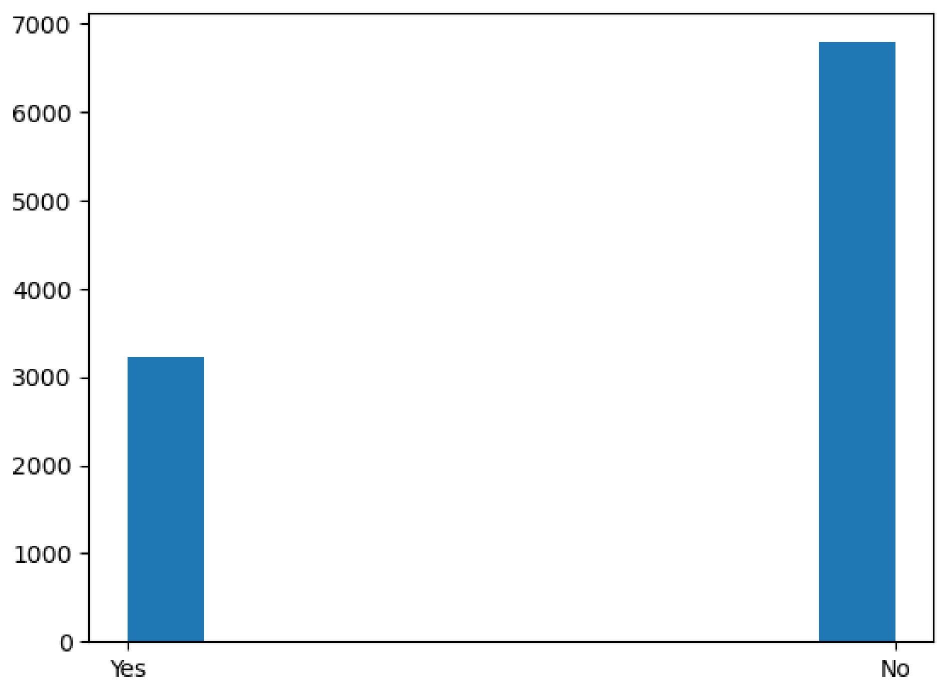

```
Income Original:
```



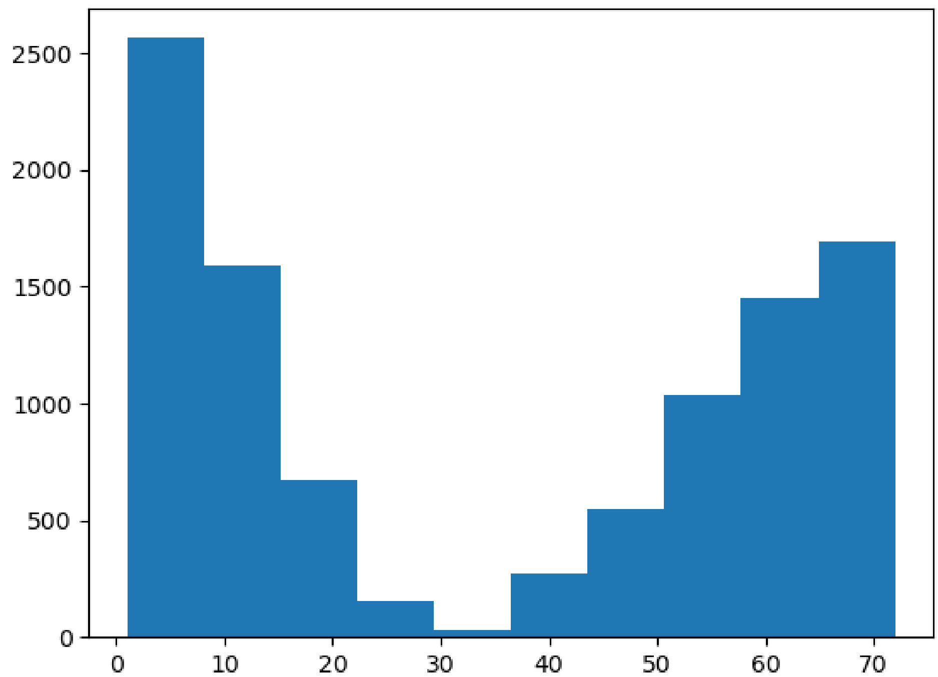

```
Overweight Original:
```




Anxiety Original:




Initial_days Original:


2500
2000
1500
1000
500
0
0
10
20
30
40
50
60
70

In [28]:
```
#Data Wrangling; turn categorical values into quantitative data
df['ReAdmis_numeric'] = df['ReAdmis']
dict_ReAdmis = {"ReAdmis_numeric": {"No": 0, "Yes": 1}}
df.replace(dict_ReAdmis, inplace=True)

df['Soft_drink_numeric'] = df['Soft_drink']
dict_Soft_drink = {"Soft_drink_numeric": {"No": 0, "Yes": 1}}
df.replace(dict_Soft_drink, inplace=True)

df['HighBlood_numeric'] = df['HighBlood']
dict_HighBlood = {"HighBlood_numeric": {"No": 0, "Yes": 1}}
df.replace(dict_HighBlood, inplace=True)

df['Stroke_numeric'] = df['Stroke']
dict_stroke = {"Stroke_numeric": {"No": 0, "Yes": 1}}
df.replace(dict_stroke, inplace=True)

df['Arthritis_numeric'] = df['Arthritis']
dict_arthritis = {"Arthritis_numeric": {"No": 0, "Yes": 1}}
df.replace(dict_arthritis, inplace=True)

df['Diabetes_numeric'] = df['Diabetes']
dict_diabetes = {"Diabetes_numeric": {"No": 0, "Yes": 1}}
df.replace(dict_diabetes, inplace=True)

df['Hyperlipidemia_numeric'] = df['Hyperlipidemia']
dict_hyperlipidemia = {"Hyperlipidemia_numeric": {"No": 0, "Yes": 1}}
df.replace(dict_hyperlipidemia, inplace=True)

df['BackPain_numeric'] = df['BackPain']
dict_backpain = {"BackPain_numeric": {"No": 0, "Yes": 1}}
df.replace(dict_backpain, inplace=True)

df['Allergic_rhinitis_numeric'] = df['Allergic_rhinitis']
dict_allergies = {"Allergic_rhinitis_numeric": {"No": 0, "Yes": 1}}
df.replace(dict_allergies, inplace=True)

df['Reflux_esophagitis_numeric'] = df['Reflux_esophagitis']
dict_reflux = {"Reflux_esophagitis_numeric": {"No": 0, "Yes": 1}}
df.replace(dict_reflux, inplace=True)

df['Asthma_numeric'] = df['Asthma']
dict_asthma = {"Asthma_numeric": {"No": 0, "Yes": 1}}
df.replace(dict_asthma, inplace=True)

df['Overweight_numeric'] = df['Overweight']
dict_Overweight = {"Overweight_numeric": {"No": 0, "Yes": 1}}
df.replace(dict_Overweight, inplace=True)

df['Anxiety_numeric'] = df['Anxiety']
dict_Anxiety = {"Anxiety_numeric": {"No": 0, "Yes": 1}}
df.replace(dict_Anxiety, inplace=True)

df = pd.get_dummies(df, columns=["Marital", "Services", "Gender", "Initial_admin", "Complicatio

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 76 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   CaseOrder                     10000 non-null  int64  
 1   Customer_id                   10000 non-null  object 
 2   Interaction                   10000 non-null  object 
 3   UID                           10000 non-null  object 
 4   City                          10000 non-null  object 
 5   State                         10000 non-null  object 
 6   County                        10000 non-null  object 
 7   Zip                           10000 non-null  int64  
 8   Lat                           10000 non-null  float64
 9   Lng                           10000 non-null  float64
 10  Population                    10000 non-null  int64  
 11  Area                          10000 non-null  object 
 12  TimeZone                      10000 non-null  object 
 13  Job                           10000 non-null  object 
 14  Children                      10000 non-null  int64  
 15  Age                           10000 non-null  int64  
 16  Income                        10000 non-null  float64
 17  ReAdmis                       10000 non-null  object 
 18  VitD_levels                   10000 non-null  float64
 19  Doc_visits                    10000 non-null  int64  
 20  Full_meals_eaten              10000 non-null  int64  
 21  vitD_supp                     10000 non-null  int64  
 22  Soft_drink                    10000 non-null  object 
 23  HighBlood                     10000 non-null  object 
 24  Stroke                        10000 non-null  object 
 25  Overweight                    10000 non-null  object 
 26  Arthritis                     10000 non-null  object 
 27  Diabetes                      10000 non-null  object 
 28  Hyperlipidemia                10000 non-null  object 
 29  BackPain                      10000 non-null  object 
 30  Anxiety                       10000 non-null  object 
 31  Allergic_rhinitis             10000 non-null  object 
 32  Reflux_esophagitis            10000 non-null  object 
 33  Asthma                        10000 non-null  object 
 34  Initial_days                  10000 non-null  float64
 35  TotalCharge                   10000 non-null  float64
 36  Additional_charges            10000 non-null  float64
 37  Item1                         10000 non-null  int64  
 38  Item2                         10000 non-null  int64  
 39  Item3                         10000 non-null  int64  
 40  Item4                         10000 non-null  int64  
 41  Item5                         10000 non-null  int64  
 42  Item6                         10000 non-null  int64  
 43  Item7                         10000 non-null  int64  
 44  Item8                         10000 non-null  int64  
 45  ReAdmis_numeric               10000 non-null  int64  
 46  Soft_drink_numeric            10000 non-null  int64  
 47  HighBlood_numeric             10000 non-null  int64  
 48  Stroke_numeric                10000 non-null  int64  
 49  Arthritis_numeric             10000 non-null  int64  
 50  Diabetes_numeric              10000 non-null  int64  
 51  Hyperlipidemia_numeric        10000 non-null  int64  
 52  BackPain_numeric              10000 non-null  int64  
 53  Allergic_rhinitis_numeric     10000 non-null  int64  
 54  Reflux_esophagitis_numeric    10000 non-null  int64  
 55  Asthma_numeric                10000 non-null  int64  
 56  Overweight_numeric            10000 non-null  int64  
 57  Anxiety_numeric               10000 non-null  int64  
 58  Marital_Divorced              10000 non-null  uint8  
 59  Marital_Married               10000 non-null  uint8  
 60  Marital_Never Married         10000 non-null  uint8  
 61  Marital_Separated             10000 non-null  uint8  
```

```
 62  Marital_Widowed                     10000 non-null  uint8
 63  Services_Blood Work                 10000 non-null  uint8
 64  Services_CT Scan                    10000 non-null  uint8
 65  Services_Intravenous                10000 non-null  uint8
 66  Services_MRI                        10000 non-null  uint8
 67  Gender_Female                       10000 non-null  uint8
 68  Gender_Male                         10000 non-null  uint8
 69  Gender_Nonbinary                    10000 non-null  uint8
 70  Initial_admin_Elective Admission    10000 non-null  uint8
 71  Initial_admin_Emergency Admission   10000 non-null  uint8
 72  Initial_admin_Observation Admission  10000 non-null  uint8
 73  Complication_risk_High              10000 non-null  uint8
 74  Complication_risk_Low               10000 non-null  uint8
 75  Complication_risk_Medium            10000 non-null  uint8
dtypes: float64(7), int64(29), object(22), uint8(18)
memory usage: 4.6+ MB
```

In [29]:
```
#Renaming columns from pd.get_dummies
df = df.rename({'Initial_admin_Elective Admission': 'Initial_admin_Elective_Admission',
                'Initial_admin_Emergency Admission': 'Initial_admin_Emergency_Admission',
                'Initial_admin_Observation Admission': 'Initial_admin_Observation_Admission',
                'Marital_Never Married': 'Marital_Never_Married',
                'Services_Blood Work': 'Services_Blood_Work',
                'Services_CT Scan': 'Services_CT_Scan'}, axis ='columns')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 76 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   CaseOrder                     10000 non-null  int64  
 1   Customer_id                   10000 non-null  object 
 2   Interaction                   10000 non-null  object 
 3   UID                           10000 non-null  object 
 4   City                          10000 non-null  object 
 5   State                         10000 non-null  object 
 6   County                        10000 non-null  object 
 7   Zip                           10000 non-null  int64  
 8   Lat                           10000 non-null  float64
 9   Lng                           10000 non-null  float64
 10  Population                    10000 non-null  int64  
 11  Area                          10000 non-null  object 
 12  TimeZone                      10000 non-null  object 
 13  Job                           10000 non-null  object 
 14  Children                      10000 non-null  int64  
 15  Age                           10000 non-null  int64  
 16  Income                        10000 non-null  float64
 17  ReAdmis                       10000 non-null  object 
 18  VitD_levels                   10000 non-null  float64
 19  Doc_visits                    10000 non-null  int64  
 20  Full_meals_eaten              10000 non-null  int64  
 21  vitD_supp                     10000 non-null  int64  
 22  Soft_drink                    10000 non-null  object 
 23  HighBlood                     10000 non-null  object 
 24  Stroke                        10000 non-null  object 
 25  Overweight                    10000 non-null  object 
 26  Arthritis                     10000 non-null  object 
 27  Diabetes                      10000 non-null  object 
 28  Hyperlipidemia                10000 non-null  object 
 29  BackPain                      10000 non-null  object 
 30  Anxiety                       10000 non-null  object 
 31  Allergic_rhinitis             10000 non-null  object 
 32  Reflux_esophagitis            10000 non-null  object 
 33  Asthma                        10000 non-null  object 
 34  Initial_days                  10000 non-null  float64
 35  TotalCharge                   10000 non-null  float64
 36  Additional_charges            10000 non-null  float64
 37  Item1                         10000 non-null  int64  
 38  Item2                         10000 non-null  int64  
 39  Item3                         10000 non-null  int64  
 40  Item4                         10000 non-null  int64  
 41  Item5                         10000 non-null  int64  
 42  Item6                         10000 non-null  int64  
 43  Item7                         10000 non-null  int64  
 44  Item8                         10000 non-null  int64  
 45  ReAdmis_numeric               10000 non-null  int64  
 46  Soft_drink_numeric            10000 non-null  int64  
 47  HighBlood_numeric             10000 non-null  int64  
 48  Stroke_numeric                10000 non-null  int64  
 49  Arthritis_numeric             10000 non-null  int64  
 50  Diabetes_numeric              10000 non-null  int64  
 51  Hyperlipidemia_numeric        10000 non-null  int64  
 52  BackPain_numeric              10000 non-null  int64  
 53  Allergic_rhinitis_numeric     10000 non-null  int64  
 54  Reflux_esophagitis_numeric    10000 non-null  int64  
 55  Asthma_numeric                10000 non-null  int64  
 56  Overweight_numeric            10000 non-null  int64  
 57  Anxiety_numeric               10000 non-null  int64  
 58  Marital_Divorced              10000 non-null  uint8  
 59  Marital_Married               10000 non-null  uint8  
 60  Marital_Never_Married         10000 non-null  uint8  
 61  Marital_Separated             10000 non-null  uint8  
```

```
 62  Marital_Widowed                     10000 non-null  uint8
 63  Services_Blood_Work                 10000 non-null  uint8
 64  Services_CT_Scan                    10000 non-null  uint8
 65  Services_Intravenous                10000 non-null  uint8
 66  Services_MRI                        10000 non-null  uint8
 67  Gender_Female                       10000 non-null  uint8
 68  Gender_Male                         10000 non-null  uint8
 69  Gender_Nonbinary                    10000 non-null  uint8
 70  Initial_admin_Elective_Admission    10000 non-null  uint8
 71  Initial_admin_Emergency_Admission   10000 non-null  uint8
 72  Initial_admin_Observation_Admission 10000 non-null  uint8
 73  Complication_risk_High              10000 non-null  uint8
 74  Complication_risk_Low               10000 non-null  uint8
 75  Complication_risk_Medium            10000 non-null  uint8
dtypes: float64(7), int64(29), object(22), uint8(18)
memory usage: 4.6+ MB
```

In [30]:
```
##Univariate Stats Dataframe
def unistats(df):
    output_df = pd.DataFrame(columns=['Count', 'Missing', 'Unique', 'Dtype', 'Numeric', 'Mean',

    for col in df:
        if pd.api.types.is_numeric_dtype(df[col]):
            output_df.loc[col] = [df[col].count(), df[col].isnull().sum(), df[col].nunique(), d
        else:
            output_df.loc[col] = [df[col].count(), df[col].isnull().sum(), df[col].nunique(), d
    return output_df.sort_values(by=['Numeric', 'Skew', 'Unique'], ascending=False)

df.drop(columns=['CaseOrder', 'Customer_id', 'Interaction', 'UID', 'State', 'County', 'Job', 'Z
print(unistats(df))
```

```
                                  Count  Missing  Unique    Dtype  Numeric  \
Gender_Nonbinary                  10000        0       2    uint8     True
Services_MRI                      10000        0       2    uint8     True
Services_CT_Scan                  10000        0       2    uint8     True
Population                        10000        0    5951    int64     True
vitD_supp                         10000        0       6    int64     True
Marital_Never_Married             10000        0       2    uint8     True
Marital_Separated                 10000        0       2    uint8     True
Stroke_numeric                    10000        0       2    int64     True
Marital_Married                   10000        0       2    uint8     True
Marital_Widowed                   10000        0       2    uint8     True
Children                          10000        0      11    int64     True
Income                            10000        0    9993  float64     True
Complication_risk_Low             10000        0       2    uint8     True
Initial_admin_Observation_Admission  10000        0       2    uint8     True
Initial_admin_Elective_Admission  10000        0       2    uint8     True
Soft_drink_numeric                10000        0       2    int64     True
Diabetes_numeric                  10000        0       2    int64     True
Full_meals_eaten                  10000        0       8    int64     True
Asthma_numeric                    10000        0       2    int64     True
Additional_charges                10000        0    9418  float64     True
Services_Intravenous              10000        0       2    uint8     True
Anxiety_numeric                   10000        0       2    int64     True
Complication_risk_High            10000        0       2    uint8     True
Hyperlipidemia_numeric            10000        0       2    int64     True
Arthritis_numeric                 10000        0       2    int64     True
ReAdmis_numeric                   10000        0       2    int64     True
Allergic_rhinitis_numeric         10000        0       2    int64     True
HighBlood_numeric                 10000        0       2    int64     True
BackPain_numeric                  10000        0       2    int64     True
Reflux_esophagitis_numeric        10000        0       2    int64     True
Complication_risk_Medium          10000        0       2    uint8     True
Gender_Male                       10000        0       2    uint8     True
Initial_days                      10000        0    9997  float64     True
TotalCharge                       10000        0    9997  float64     True
VitD_levels                       10000        0    9976  float64     True
Age                               10000        0      72    int64     True
Doc_visits                        10000        0       9    int64     True
Initial_admin_Emergency_Admission 10000        0       2    uint8     True
Services_Blood_Work               10000        0       2    uint8     True
Overweight_numeric                10000        0       2    int64     True

                                          Mean         Mode          Min  \
Gender_Nonbinary                      0.021400      0.00000     0.000000
Services_MRI                          0.038000      0.00000     0.000000
Services_CT_Scan                      0.122500      0.00000     0.000000
Population                         9965.253800      0.00000     0.000000
vitD_supp                             0.398900      0.00000     0.000000
Marital_Never_Married                 0.198400      0.00000     0.000000
Marital_Separated                     0.198700      0.00000     0.000000
Stroke_numeric                        0.199300      0.00000     0.000000
Marital_Married                       0.202300      0.00000     0.000000
Marital_Widowed                       0.204500      0.00000     0.000000
Children                              2.097200      0.00000     0.000000
Income                            40490.495160  14572.40000   154.080000
Complication_risk_Low                 0.212500      0.00000     0.000000
Initial_admin_Observation_Admission   0.243600      0.00000     0.000000
Initial_admin_Elective_Admission      0.250400      0.00000     0.000000
Soft_drink_numeric                    0.257500      0.00000     0.000000
Diabetes_numeric                      0.273800      0.00000     0.000000
Full_meals_eaten                      1.001400      0.00000     0.000000
Asthma_numeric                        0.289300      0.00000     0.000000
Additional_charges                12934.528587   3883.66416  3125.703000
Services_Intravenous                  0.313000      0.00000     0.000000
Anxiety_numeric                       0.321500      0.00000     0.000000
Complication_risk_High                0.335800      0.00000     0.000000
Hyperlipidemia_numeric                0.337200      0.00000     0.000000
```

```
Arthritis_numeric                      0.357400     0.00000     0.000000
ReAdmis_numeric                        0.366900     0.00000     0.000000
Allergic_rhinitis_numeric              0.394100     0.00000     0.000000
HighBlood_numeric                      0.409000     0.00000     0.000000
BackPain_numeric                       0.411400     0.00000     0.000000
Reflux_esophagitis_numeric             0.413500     0.00000     0.000000
Complication_risk_Medium               0.451700     0.00000     0.000000
Gender_Male                            0.476800     0.00000     0.000000
Initial_days                          34.455299    63.54432     1.001981
TotalCharge                         5312.172769  7555.45200  1938.312067
VitD_levels                           17.964262    15.26009     9.806483
Age                                   53.511700    47.00000    18.000000
Doc_visits                             5.012200     5.00000     1.000000
Initial_admin_Emergency_Admission      0.506000     1.00000     0.000000
Services_Blood_Work                    0.526500     1.00000     0.000000
Overweight_numeric                     0.709400     1.00000     0.000000

                                         Median           Max  \
Gender_Nonbinary                       0.000000      1.000000
Services_MRI                           0.000000      1.000000
Services_CT_Scan                       0.000000      1.000000
Population                          2769.000000 122814.000000
vitD_supp                              0.000000      5.000000
Marital_Never_Married                  0.000000      1.000000
Marital_Separated                      0.000000      1.000000
Stroke_numeric                         0.000000      1.000000
Marital_Married                        0.000000      1.000000
Marital_Widowed                        0.000000      1.000000
Children                               1.000000     10.000000
Income                             33768.420000 207249.100000
Complication_risk_Low                  0.000000      1.000000
Initial_admin_Observation_Admission    0.000000      1.000000
Initial_admin_Elective_Admission       0.000000      1.000000
Soft_drink_numeric                     0.000000      1.000000
Diabetes_numeric                       0.000000      1.000000
Full_meals_eaten                       1.000000      7.000000
Asthma_numeric                         0.000000      1.000000
Additional_charges                 11573.977735  30566.070000
Services_Intravenous                   0.000000      1.000000
Anxiety_numeric                        0.000000      1.000000
Complication_risk_High                 0.000000      1.000000
Hyperlipidemia_numeric                 0.000000      1.000000
Arthritis_numeric                      0.000000      1.000000
ReAdmis_numeric                        0.000000      1.000000
Allergic_rhinitis_numeric              0.000000      1.000000
HighBlood_numeric                      0.000000      1.000000
BackPain_numeric                       0.000000      1.000000
Reflux_esophagitis_numeric             0.000000      1.000000
Complication_risk_Medium               0.000000      1.000000
Gender_Male                            0.000000      1.000000
Initial_days                          35.836244     71.981490
TotalCharge                         5213.952000   9180.728000
VitD_levels                           17.951122     26.394449
Age                                   53.000000     89.000000
Doc_visits                             5.000000      9.000000
Initial_admin_Emergency_Admission      1.000000      1.000000
Services_Blood_Work                    1.000000      1.000000
Overweight_numeric                     1.000000      1.000000

                                            Std      Skew       Kurt
Gender_Nonbinary                       0.144721  6.615434  41.772323
Services_MRI                           0.191206  4.833456  21.366572
Services_CT_Scan                       0.327879  2.303141   3.305119
Population                         14824.758614  2.229959   5.880913
vitD_supp                              0.628505  1.550205   2.330763
Marital_Never_Married                  0.398815  1.512784   0.288572
Marital_Separated                      0.399042  1.510420   0.281425
```

```
Stroke_numeric                         0.399494  1.505705   0.267202
Marital_Married                        0.401735  1.482369   0.197456
Marital_Widowed                        0.403356  1.465500   0.147720
Children                               2.163659  1.448013   2.076321
Income                             28521.153293  1.405899   2.745690
Complication_risk_Low                  0.409097  1.405815  -0.023688
Initial_admin_Observation_Admission    0.429276  1.194810  -0.572544
Initial_admin_Elective_Admission       0.433265  1.152412  -0.672081
Soft_drink_numeric                     0.437279  1.109354  -0.769488
Diabetes_numeric                       0.445930  1.014712  -0.970553
Full_meals_eaten                       1.008117  1.009461   1.042727
Asthma_numeric                         0.453460  0.929485  -1.136285
Additional_charges                  6542.601544  0.831842  -0.142684
Services_Intravenous                   0.463738  0.806652  -1.349583
Anxiety_numeric                        0.467076  0.764483  -1.415849
Complication_risk_High                 0.472293  0.695470  -1.516625
Hyperlipidemia_numeric                 0.472777  0.688834  -1.525813
Arthritis_numeric                      0.479258  0.595206  -1.646059
ReAdmis_numeric                        0.481983  0.552412  -1.695180
Allergic_rhinitis_numeric              0.488681  0.433498  -1.812442
HighBlood_numeric                      0.491674  0.370238  -1.863296
BackPain_numeric                       0.492112  0.360153  -1.870664
Reflux_esophagitis_numeric             0.492486  0.351350  -1.876929
Complication_risk_Medium               0.497687  0.194137  -1.962703
Gender_Male                            0.499486  0.092914  -1.991765
Initial_days                          26.309341  0.070286  -1.754525
TotalCharge                         2180.393838  0.069661  -1.668267
VitD_levels                            2.017231  0.032435  -0.022112
Age                                   20.638538  0.005117  -1.189527
Doc_visits                             1.045734 -0.018563   0.025999
Initial_admin_Emergency_Admission      0.499989 -0.024005  -1.999824
Services_Blood_Work                    0.499322 -0.106165  -1.989127
Overweight_numeric                     0.454062 -0.922526  -1.149176
```

In [31]:
```
#C4. Cleaned Dataset:
# Provide a copy of the cleaned Data Set
df.to_csv(r'C:\Users\mmorg\WGU\D209\Cleaned209data.csv')
```

```
Part IV: Analysis
```

In [ ]:
```
D1. Split Data into Training and Test Data Sets and Provide the File(s)
```

In [32]:
```
#Set predictor variables & target variable
X = df.drop(['ReAdmis_numeric'], axis=1)
y = df["ReAdmis_numeric"]
```

In [33]:
```
SEED = 1

X_train, X_test, y_train, y_test = train_test_split(X,y,train_size=0.8,test_size=0.2,
                                                    random_state=15,stratify=y)
```

In [34]:
```
#export training and test set to csv files
X_train.to_csv(r'C:\Users\mmorg\WGU\D209\X_train.csv')
X_test.to_csv(r'C:\Users\mmorg\WGU\D209\X_test.csv')
y_train.to_csv(r'C:\Users\mmorg\WGU\D209\y_train.csv')
y_test.to_csv(r'C:\Users\mmorg\WGU\D209\y_test.csv')
```

```
D2. Describe Analysis Technique Used
```

A decision tree is a non-parametric supervised learning method that builds the models in a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The result is a tree with decision nodes and leaf nodes. A decision node has two or more branches. Leaf node represents a classification or decision. The topmost decision node in a tree corresponds to the best predictor called the root node. Decision trees can handle both categorical and numerical data.

In [ ]:
```
D3. Provide the code used to perform the classficiation analysis from part D2.
```

In [35]:
```
# Instantiate dt model
dt = DecisionTreeClassifier(max_depth=2, random_state=1)

# Fit dt to the training set
dt.fit(X_train,y_train)

# Predict the test set labels
y_pred = dt.predict(X_test)

# Evaluate the test-set accuracy
accuracy_score(y_test, y_pred)
```

Out[35]:
```
0.982
```

Part V: Data Summary and Implications
E1. Explain the accuracy and the mean squared error (MSE) of your prediction model.

The accuracy of the model is initially reported as 98.2% accurate. The MSE is also very low. Because the accuracy is high and the MSE is low, this analysis supports the model's high accuracy.

In [36]:
```
# Setting up model to calculate MSE for the training set
X_pred = dt.predict(X_train)

# Calculate the MSE for the training set
train_mse = mean_squared_error(y_train, X_pred)
print("Training MSE:", train_mse)

# Calculate the MSE for the test set
test_mse = mean_squared_error(y_test, y_pred)
print("Test MSE:", test_mse)
```

```
Training MSE: 0.022625
Test MSE: 0.018
```

E2. Discuss the results and implications of your classification analysis.

The accuracy of the model is initially reported as 98.2% accurate. The MSE is also very low. Because the accuracy is high and the MSE is low, this analysis supports the model's high accuracy. To make the model more accurate we can do hyperparameter tuning. Hyperparameter tuning could be used to perform variable analysis and finding the optimal values for max_depth. The model accuracy could be further improved by performing hyperparameter tuning on additional parameters and by using more labeled data to train the model.

E3. Discuss one limitation of your data analysis.

Decision tree analysis can be somewhat unstable. One small change in the data can result in a major change in the structure of the decision tree. (Taylor, 2023)

E4. Recommend a course of action for the real-world organizational situation from part A1 based on your results and implications discussed in part E2.

This model seems to be able to predict with 98.2% accuracy whether a patient will be re-admitted or not. When new patients are admitted their data should be fed into the model, and the resulting prediction can then be used to categorize the patient and their re-admission risk factor. This information can be useful in coming up with a patient-treament plan. Patients who are predicted for re-admission should be treated with a more intensive care plan than those who are predicted for no re-admission. The intensive care plan can hopefully reduce the chances of future re-admission and save the hospital money.

Part VI: Demonstration
F. Provide a Panopto video recording that includes a demonstration of the functionality of the code used for the analysis and a summary of the programming environment.

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=f8d2071d-7ce1-47d1-82cc-afcc018a4407

G. Record the web sources used to acquire data or segments of third-party code to support the analysis. Ensure the web sources are reliable.

H. Acknowledge sources, using in-text citations and references, for content that is quoted, paraphrased, or summarized.

Navlani, Avinash. “Decision Tree Classification in Python Tutorial.” datacamp, 28 Dec. 2018, www.datacamp.com/community/tutorials/decision-tree-classification-python. Accessed 10 Apr. 2022.

Taylor, Sebastian (2023, March 8). Decision Tree - Overview, Decision Types, Applications. Retrieve March, 18, 2023 from https://corporatefinanceinstitute.com/resources/data-science/decision-tree/.