In [1]:

```
#libraries used
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn
import missingno as msno
from scipy import stats
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
```

In [2]:

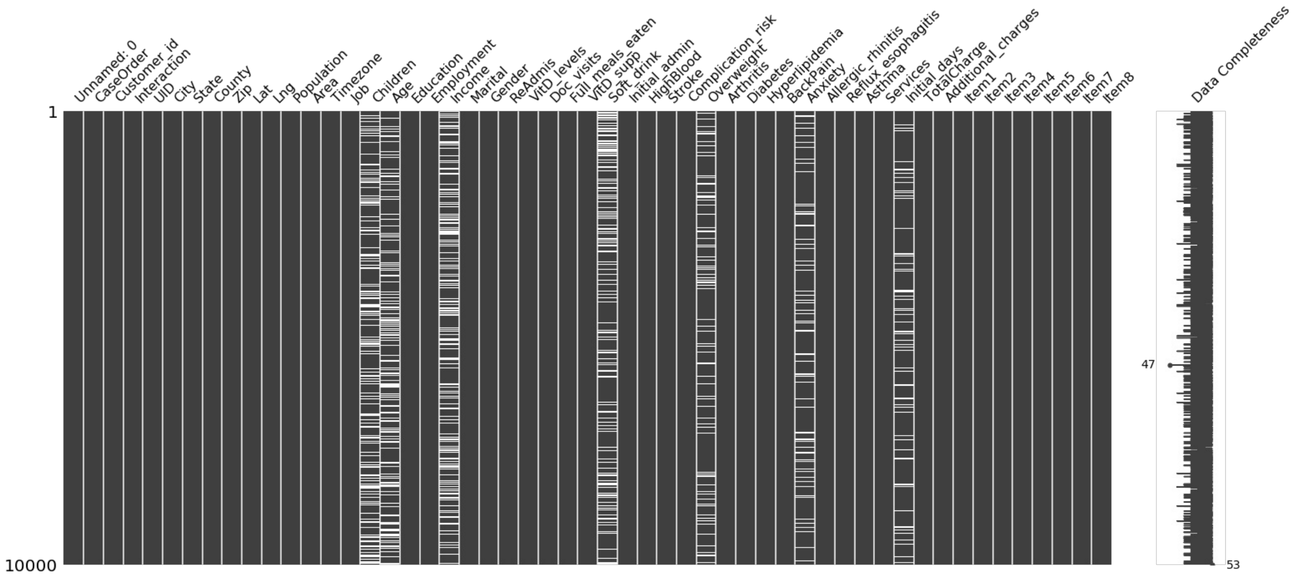
```
#import datafile
data = pd.read_csv('C:/Users/ericy/Desktop/medical_raw_data.csv')
```

In [3]:

```
# Graph variables to visualize missing data
msno.matrix(data, labels=True)
```

Out[3]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f8228e9a08>
```



In [4]:

```
#find out shape, create 'Index' variable, Drop 'Unnamed:0'.
data.shape
data['Index'] = pd.Series(range(0, 10000))
#Note https://appdividend.com/2020/06/01/pandas-dataframe-drop-method-in-python/ for .drop() method in PA
data.drop(['Unnamed: 0'], axis=1, inplace=True)
#Move 'Index' to beginning of data.
column_to_move = data.pop('Index')
data.insert(0, 'Index', column_to_move)
data.head()
```

Out[4]:

	Index	CaseOrder	Customer_id	Interaction	UID	City	State	County	Zip	Lat	...	T
0	0	1	C412403	8cd49b13-f45a-4b47-a2bd-173ffa932c2f	3a83ddb66e2ae73798bdf1d705dc0932	Eva	AL	Morgan	35621	34.34960	...	3
1	1	2	Z919181	d2450b70-0337-4406-bdbb-bc1037f1734c	176354c5eef714957d486009feabf195	Marianna	FL	Jackson	32446	30.84513	...	4.
2	2	3	F995323	a2057123-abf5-4a2c-abad-8ffe33512562	e19a0fa00aeda885b8a436757e889bc9	Sioux Falls	SD	Minnehaha	57110	43.54321	...	2
3	3	4	A879973	1dec528d-eb34-4079-adce-0d7a40e82205	cd17d7b6d152cb6f23957346d11c3f07	New Richland	MN	Waseca	56072	43.89744	...	2.
4	4	5	C544523	5885f56b-d6da-43a3-8760-83583af94266	d2f0425877b10ed6bb381f3e2579424a	West Point	VA	King William	23181	37.59894	...	1.

5 rows × 53 columns

In [5]:

```
#Double check which variables have null values
data.isnull().any()
```

Out[5]:

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

In [6]:

```
#Total null cases in each variable
data.isnull().sum()
#Variables to address with null values: Children, Age, Income, Soft_drink, Overweight, Anxiety, Initial_days.
```

Out[6]:

```
Index                    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 [7]:

```
#Rename Item1-Item8 variables to names provided in data's supplemental PDF.
#Also rename 'CaseOrder', 'ReAdmis', 'HighBlood', 'BackPain', and 'TotalCharge' to have the same syntax in variab
le naming across the dataset
#Note https://re-thought.com/guide-to-renaming-columns-with-python-pandas/ for .rename() method
data.rename(columns={'CaseOrder':'Case_order','ReAdmis':'Readmis','HighBlood':'High_blood','BackPain':'Back_pain'
,'TotalCharge':'Total_charge','Item1':'Timely_admission', 'Item2':'Timely_treatment', 'Item3':'Timely_visits', 'I
tem4':'Reliability', 'Item5':'Options', 'Item6':'Hours', 'Item7':'Courteous', 'Item8':'Active_listen'}, inplace=T
rue)
```

In [8]:

```
#Check that variables were correctly renamed
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 53 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Index               10000 non-null  int64  
 1   Case_order          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  High_blood          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  Back_pain           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  Total_charge        10000 non-null  float64
 44  Additional_charges  10000 non-null  float64
 45  Timely_admission    10000 non-null  int64  
 46  Timely_treatment    10000 non-null  int64  
 47  Timely_visits       10000 non-null  int64  
 48  Reliability         10000 non-null  int64  
 49  Options             10000 non-null  int64  
 50  Hours               10000 non-null  int64  
 51  Courteous           10000 non-null  int64  
 52  Active_listen       10000 non-null  int64  
dtypes: float64(11), int64(15), object(27)
memory usage: 4.0+ MB
```

In [9]:

```
#Address missing values in 'Soft_drink'
print(data['Soft_drink'])
data['Soft_drink'].fillna(0, inplace=True)
data.isnull().sum()
```

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

Out[9]:

```
Index                    0
Case_order               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               0
Initial_admin            0
High_blood               0
Stroke                   0
Complication_risk        0
Overweight             982
Arthritis                0
Diabetes                 0
Hyperlipidemia           0
Back_pain                0
Anxiety                984
Allergic_rhinitis        0
Reflux_esophagitis       0
Asthma                   0
Services                 0
Initial_days          1056
Total_charge             0
Additional_charges       0
Timely_admission         0
Timely_treatment         0
Timely_visits            0
Reliability              0
Options                  0
Hours                    0
Courteous                0
Active_listen            0
dtype: int64
```

In [10]:

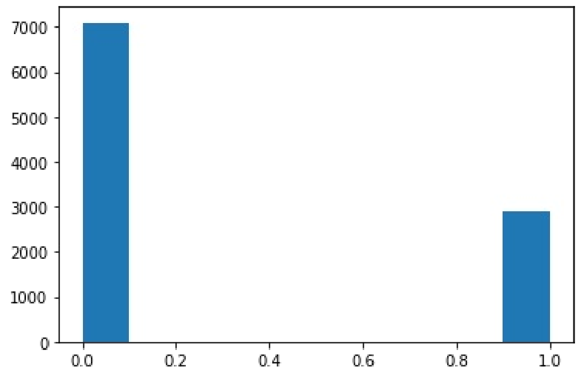
```
#Address missing values in 'Anxiety'
```

In [11]:

```
data['Anxiety'].fillna(0, inplace=True)
plt.hist(data['Anxiety'])
```

Out[11]:

```
(array([7094.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
        2906.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <a list of 10 Patch objects>)
```



In [12]:

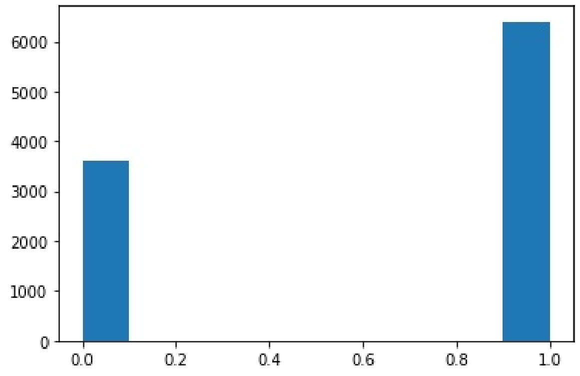
```
#Address missing values in 'Overweight'
```

In [13]:

```
data['Overweight'].fillna(0, inplace=True)
plt.hist(data['Overweight'])
```

Out[13]:

```
(array([3605.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
        6395.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <a list of 10 Patch objects>)
```



In [14]:

```
# Check for null values
data.isnull().sum()
```

Out[14]:

```
Index                    0
Case_order               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               0
Initial_admin            0
High_blood               0
Stroke                   0
Complication_risk        0
Overweight               0
Arthritis                0
Diabetes                 0
Hyperlipidemia           0
Back_pain                0
Anxiety                  0
Allergic_rhinitis        0
Reflux_esophagitis       0
Asthma                   0
Services                 0
Initial_days          1056
Total_charge             0
Additional_charges       0
Timely_admission         0
Timely_treatment         0
Timely_visits            0
Reliability              0
Options                  0
Hours                    0
Courteous                0
Active_listen            0
dtype: int64
```

In [15]:

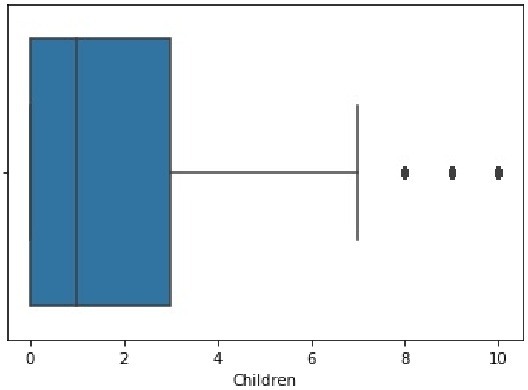
```
#Address null values in 'Children'.
```

In [16]:

```
seaborn.boxplot(data['Children'])
```

Out[16]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f822f5b4c8>
```

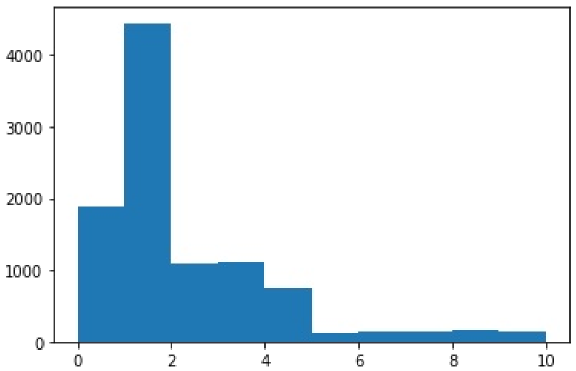


In [17]:

```
data['Children'].fillna(data['Children'].median(), inplace=True)
plt.hist(data['Children'])
```

Out[17]:

```
(array([1880., 4446., 1094., 1113.,  739.,  126.,  145.,  154.,  157.,
         146.]),
 array([ 0.,  1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.]),
 <a list of 10 Patch objects>)
```



In [18]:

```
data.isnull().sum()
```

Out[18]:

```
Index                    0
Case_order               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               0
Initial_admin            0
High_blood               0
Stroke                   0
Complication_risk        0
Overweight               0
Arthritis                0
Diabetes                 0
Hyperlipidemia           0
Back_pain                0
Anxiety                  0
Allergic_rhinitis        0
Reflux_esophagitis       0
Asthma                   0
Services                 0
Initial_days          1056
Total_charge             0
Additional_charges       0
Timely_admission         0
Timely_treatment         0
Timely_visits            0
Reliability              0
Options                  0
Hours                    0
Courteous                0
Active_listen            0
dtype: int64
```

In [19]:

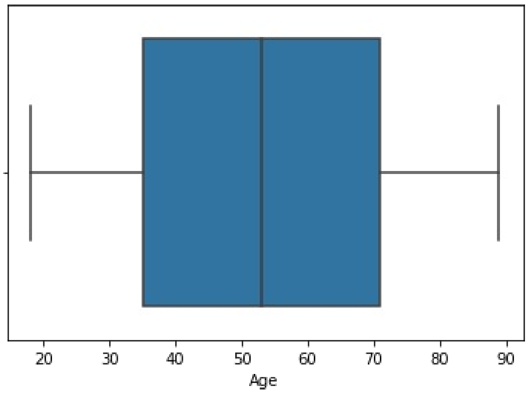
```
#Address null values in 'Age'.
```

In [20]:

```
seaborn.boxplot(data['Age'])
```

Out[20]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f82246f508>
```



In [21]:

```
data['Age'].fillna(method='backfill', inplace=True)
data.isnull().sum()
data['Age'].fillna(method='ffill', inplace=True)
data.isnull().sum()
```

Out[21]:

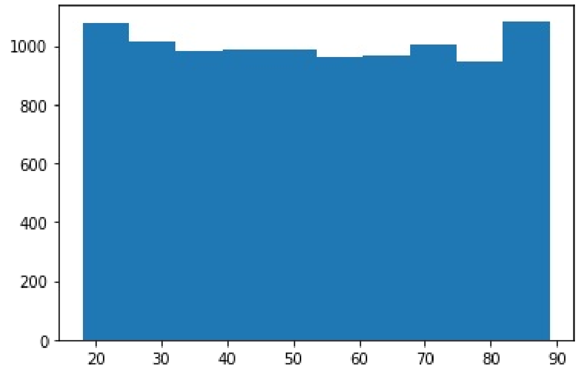
```
Index                    0
Case_order               0
Customer_id              0
Interaction              0
UID                      0
City                     0
State                    0
County                   0
Zip                      0
Lat                      0
Lng                      0
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               0
Initial_admin            0
High_blood               0
Stroke                   0
Complication_risk        0
Overweight               0
Arthritis                0
Diabetes                 0
Hyperlipidemia           0
Back_pain                0
Anxiety                  0
Allergic_rhinitis        0
Reflux_esophagitis       0
Asthma                   0
Services                 0
Initial_days          1056
Total_charge             0
Additional_charges       0
Timely_admission         0
Timely_treatment         0
Timely_visits            0
Reliability              0
Options                  0
Hours                    0
Courteous                0
Active_listen            0
dtype: int64
```

In [22]:

```
plt.hist(data['Age'])
```

Out[22]:

```
(array([1075., 1013.,  984.,  985.,  985.,  962.,  965., 1004.,  944.,
        1083.]),
 array([18. , 25.1, 32.2, 39.3, 46.4, 53.5, 60.6, 67.7, 74.8, 81.9, 89. ]),
 <a list of 10 Patch objects>)
```

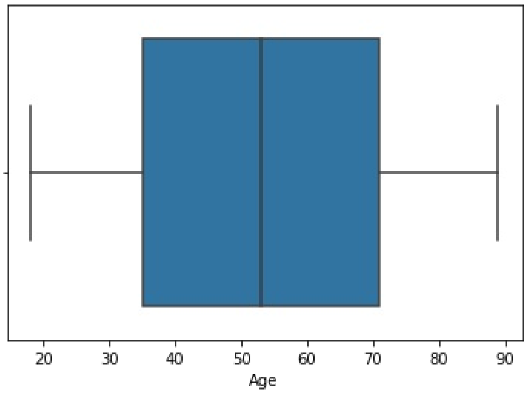


In [23]:

```
seaborn.boxplot(data['Age'])
```

Out[23]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f823f98648>
```



In [24]:

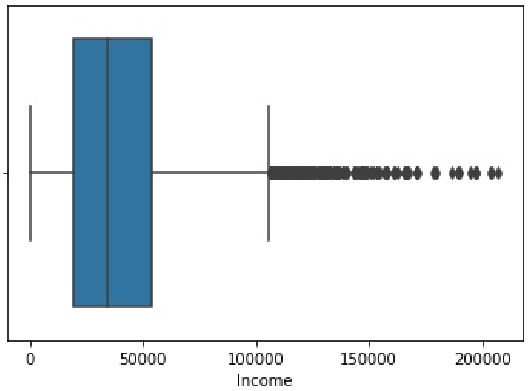
```
#Address null values in 'Income'
```

In [25]:

```
seaborn.boxplot(data['Income'])
```

Out[25]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f82400db48>
```



In [26]:

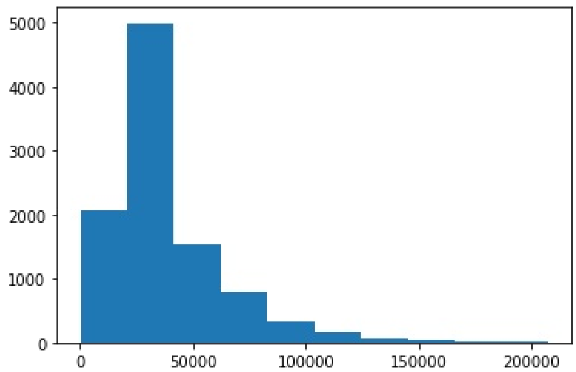
```
data['Income'].fillna(data['Income'].median(), inplace=True)
```

In [27]:

```
plt.hist(data['Income'])
```

Out[27]:

```
(array([2068., 4990., 1532.,  790.,  340.,  156.,   67.,   34.,   12.,
          11.]),
 array([1.54080000e+02, 2.08635850e+04, 4.15730900e+04, 6.22825950e+04,
        8.29921000e+04, 1.03701605e+05, 1.24411110e+05, 1.45120615e+05,
        1.65830120e+05, 1.86539625e+05, 2.07249130e+05]),
 <a list of 10 Patch objects>)
```

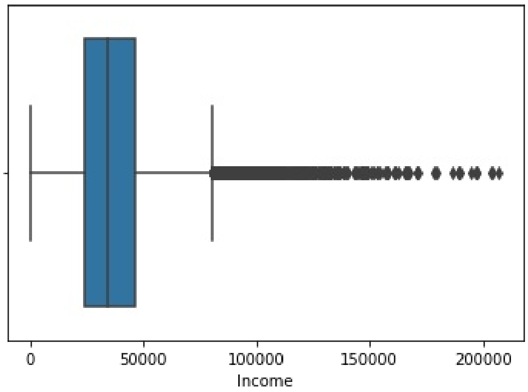


In [28]:

```
seaborn.boxplot(data['Income'])
```

Out[28]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f82446c808>
```



In [29]:

```
#Check data for null values.
data.isnull().sum()
```

Out[29]:

```
Index                    0
Case_order               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
High_blood               0
Stroke                   0
Complication_risk        0
Overweight               0
Arthritis                0
Diabetes                 0
Hyperlipidemia           0
Back_pain                0
Anxiety                  0
Allergic_rhinitis        0
Reflux_esophagitis       0
Asthma                   0
Services                 0
Initial_days          1056
Total_charge             0
Additional_charges       0
Timely_admission         0
Timely_treatment         0
Timely_visits            0
Reliability              0
Options                  0
Hours                    0
Courteous                0
Active_listen            0
dtype: int64
```

In [30]:

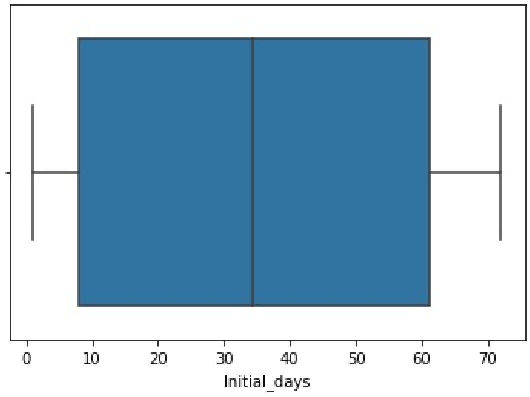
```
#Begin analyzing 'Initial_days'
```

In [31]:

```
seaborn.boxplot(data['Initial_days'])
```

Out[31]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f8258b7b88>
```

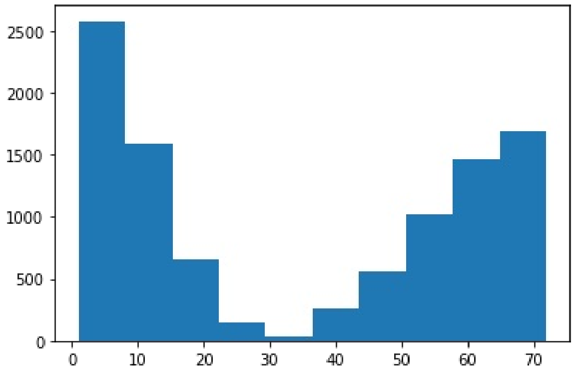


In [32]:

```
data['Initial_days'].fillna(method='backfill', inplace=True)
plt.hist(data['Initial_days'])
```

Out[32]:

```
(array([2577., 1591.,  662.,  145.,   32.,  256.,  559., 1022., 1464.,
        1692.]),
 array([ 1.00198092,  8.09993146, 15.197882  , 22.29583253, 29.39378307,
        36.49173361, 43.58968415, 50.68763469, 57.78558522, 64.88353576,
        71.9814863 ]),
 <a list of 10 Patch objects>)
```

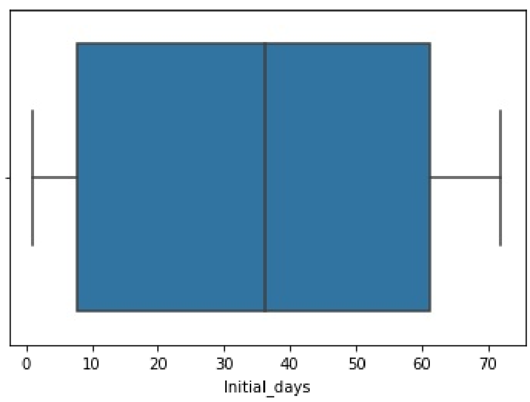


In [33]:

```
seaborn.boxplot(data['Initial_days'])
```

Out[33]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f82596da88>
```

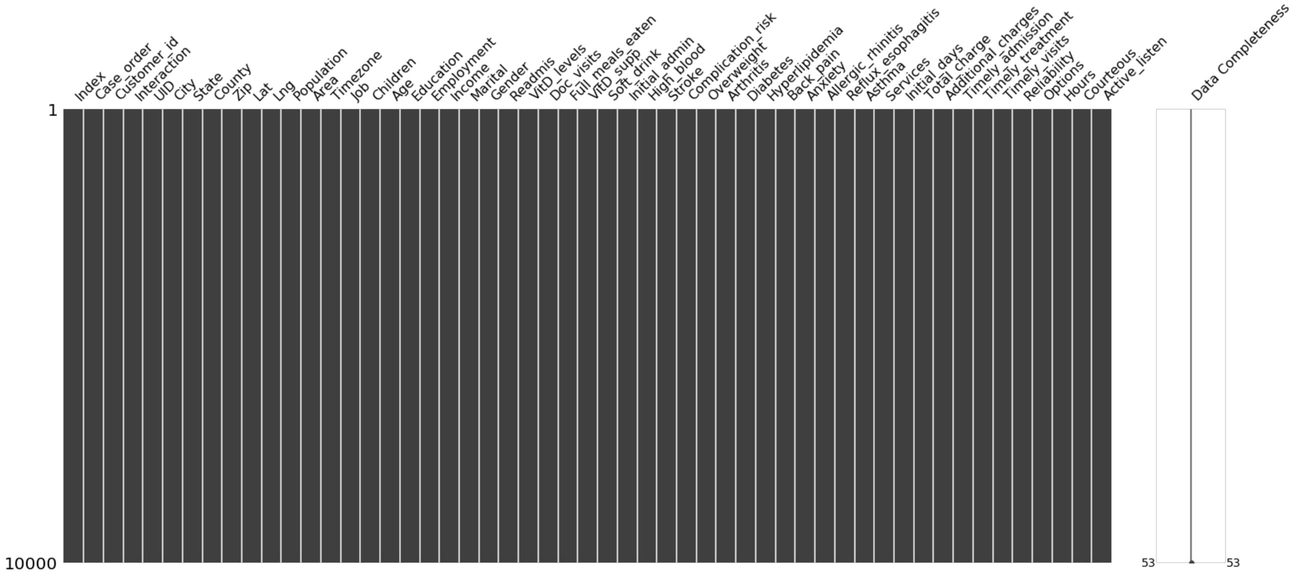


In [34]:

```
msno.matrix(data, labels=True)
```

Out[34]:

<matplotlib.axes._subplots.AxesSubplot at 0x1f8260755c8>



In [35]:

```
data.isnull().sum()
```

Out[35]:

```
Index                 0
Case_order            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
High_blood            0
Stroke                0
Complication_risk     0
Overweight            0
Arthritis             0
Diabetes              0
Hyperlipidemia        0
Back_pain             0
Anxiety               0
Allergic_rhinitis     0
Reflux_esophagitis    0
Asthma                0
Services              0
Initial_days          0
Total_charge          0
Additional_charges    0
Timely_admission      0
Timely_treatment      0
Timely_visits         0
Reliability           0
Options               0
Hours                 0
Courteous             0
Active_listen         0
dtype: int64
```

In [36]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 53 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Index               10000 non-null  int64  
 1   Case_order          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  High_blood          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  Back_pain           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  Total_charge        10000 non-null  float64
 44  Additional_charges  10000 non-null  float64
 45  Timely_admission    10000 non-null  int64  
 46  Timely_treatment    10000 non-null  int64  
 47  Timely_visits       10000 non-null  int64  
 48  Reliability         10000 non-null  int64  
 49  Options             10000 non-null  int64  
 50  Hours               10000 non-null  int64  
 51  Courteous           10000 non-null  int64  
 52  Active_listen       10000 non-null  int64  
dtypes: float64(11), int64(15), object(27)
memory usage: 4.0+ MB
```

In [37]:

```
data.head()
```

Out[37]:

	Index	Case_order	Customer_id	Interaction	UID	City	State	County	Zip	Lat	...
0	0	1	C412403	8cd49b13-f45a-4b47-a2bd-173ffa932c2f	3a83ddb66e2ae73798bdf1d705dc0932	Eva	AL	Morgan	35621	34.34960	...
1	1	2	Z919181	d2450b70-0337-4406-bdbb-bc1037f1734c	176354c5eef714957d486009feabf195	Marianna	FL	Jackson	32446	30.84513	...
2	2	3	F995323	a2057123-abf5-4a2c-abad-8ffe33512562	e19a0fa00aeda885b8a436757e889bc9	Sioux Falls	SD	Minnehaha	57110	43.54321	...
3	3	4	A879973	1dec528d-eb34-4079-adce-0d7a40e82205	cd17d7b6d152cb6f23957346d11c3f07	New Richland	MN	Waseca	56072	43.89744	...
4	4	5	C544523	5885f56b-d6da-43a3-8760-83583af94266	d2f0425877b10ed6bb381f3e2579424a	West Point	VA	King William	23181	37.59894	...

5 rows × 53 columns

In [38]:

```
plt.hist(data['Readmis'])
```

Out[38]:

```
(array([6331.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
        3669.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <a list of 10 Patch objects>)
```

In [39]:

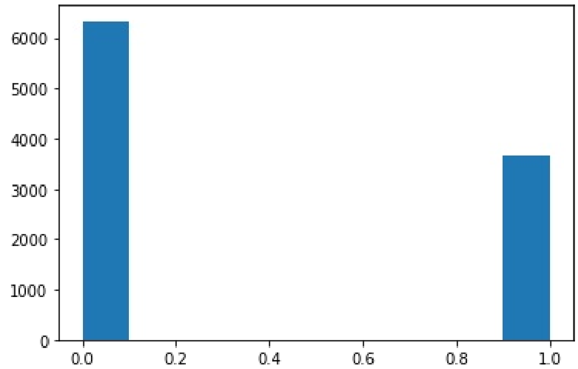
```
#Reexpression of 'Readmis' data as numeric
data['Readmis'] = data['Readmis'].astype(str)
data['Readmis'].replace(('Yes','No'), (1,0), inplace=True)
```

In [40]:

```
plt.hist(data['Readmis'])
```

Out[40]:

```
(array([6331.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
        3669.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <a list of 10 Patch objects>)
```



In [41]:

```
#Reexpression of 'Soft_drink' data as numeric
data['Soft_drink'] = data['Soft_drink'].astype(str)
data['Soft_drink'].replace(('Yes','No'), (1,0), inplace=True)
```

In [ ]:

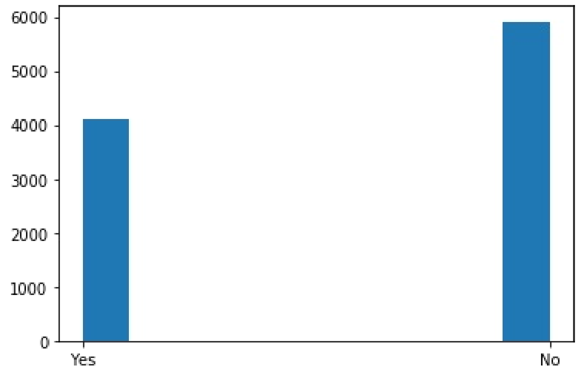
```

```

In [42]:

```
plt.hist(data['High_blood'])
```

Out[42]:

```
(array([4090.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
        5910.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <a list of 10 Patch objects>)
```



In [43]:

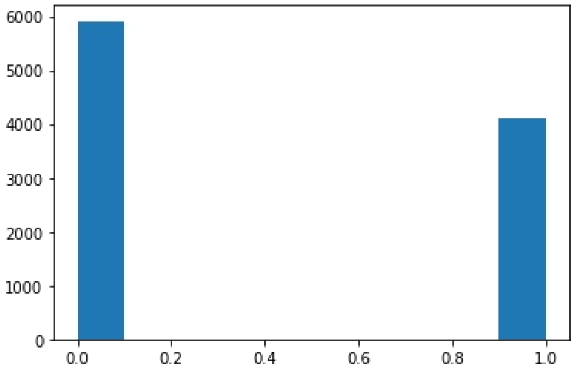
```
#Reexpression of 'High_blood' data as numeric
data['High_blood'] = data['High_blood'].astype(str)
data['High_blood'].replace(('Yes','No'), (1,0), inplace=True)
```

In [44]:

```python
plt.hist(data['High_blood'])
```

Out[44]:

```
(array([5910.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
        4090.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <a list of 10 Patch objects>)
```



In [45]:

```python
#Reexpression of 'Stroke' data as numeric
plt.hist(data['Stroke'])
```

Out[45]:

```
(array([8007.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
        1993.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <a list of 10 Patch objects>)
```

In [46]:

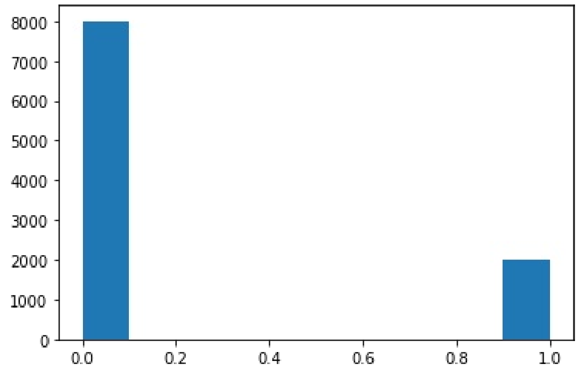
```python
#Reexpression of 'Stroke' data as numeric
data['Stroke'] = data['Stroke'].astype(str)
data['Stroke'].replace(('Yes','No'), (1,0), inplace=True)
```

In [47]:

```
plt.hist(data['Stroke'])
```

Out[47]:

```
(array([8007.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
        1993.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <a list of 10 Patch objects>)
```

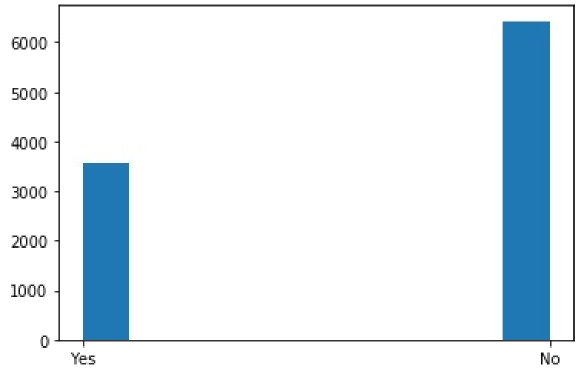


In [48]:

```
#Reexpression of 'Arthritis' data as numeric
plt.hist(data['Arthritis'])
```

Out[48]:

```
(array([3574.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
        6426.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <a list of 10 Patch objects>)
```



In [49]:

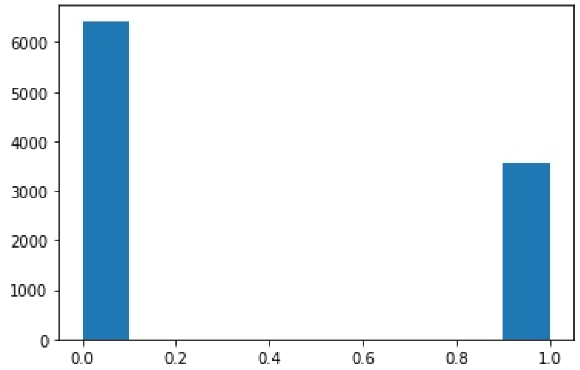
```
#Reexpression of 'Arthritis' data as numeric
data['Arthritis'] = data['Arthritis'].astype(str)
data['Arthritis'].replace(('Yes','No'), (1,0), inplace=True)
```

In [50]:

```
plt.hist(data['Arthritis'])
```

Out[50]:

```
(array([6426.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
        3574.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <a list of 10 Patch objects>)
```

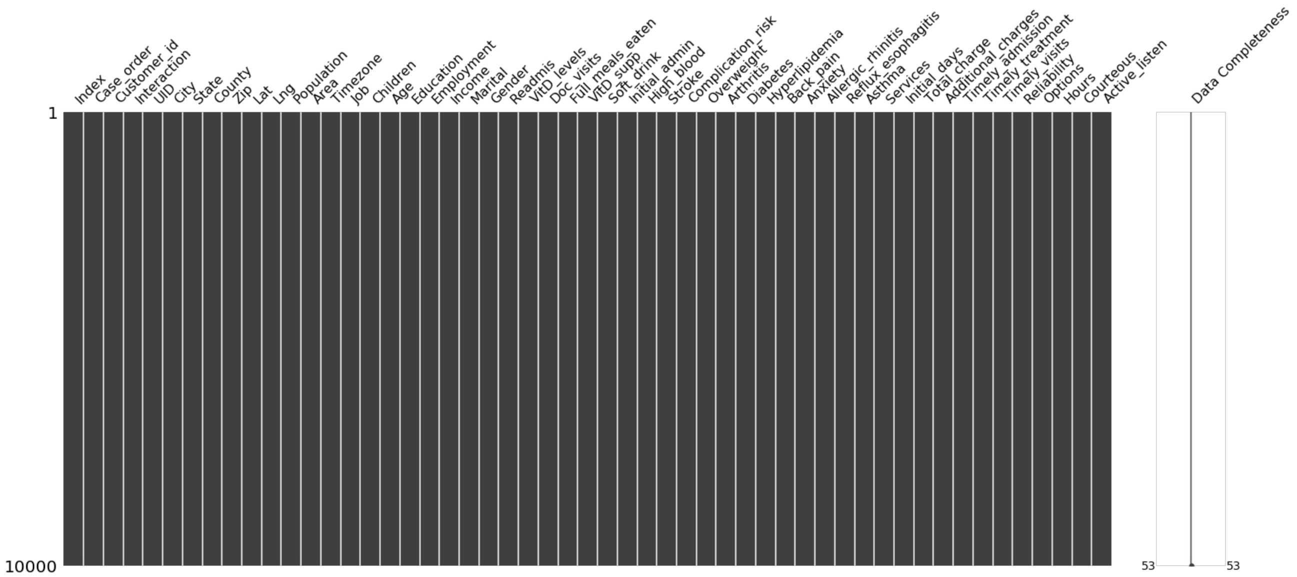


In [51]:

```
msno.matrix(data, labels=True)
```

Out[51]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f825fdbb48>
```



In [52]:

```
data.info()
```

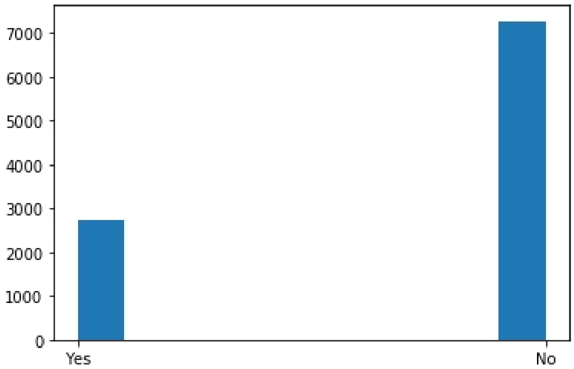
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 53 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Index               10000 non-null  int64  
 1   Case_order          10000 non-null  int64  
 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  int64  
 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  High_blood          10000 non-null  int64  
 30  Stroke              10000 non-null  int64  
 31  Complication_risk   10000 non-null  object 
 32  Overweight          10000 non-null  float64
 33  Arthritis           10000 non-null  int64  
 34  Diabetes            10000 non-null  object 
 35  Hyperlipidemia      10000 non-null  object 
 36  Back_pain           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  Total_charge        10000 non-null  float64
 44  Additional_charges  10000 non-null  float64
 45  Timely_admission    10000 non-null  int64  
 46  Timely_treatment    10000 non-null  int64  
 47  Timely_visits       10000 non-null  int64  
 48  Reliability         10000 non-null  int64  
 49  Options             10000 non-null  int64  
 50  Hours               10000 non-null  int64  
 51  Courteous           10000 non-null  int64  
 52  Active_listen       10000 non-null  int64  
dtypes: float64(11), int64(19), object(23)
memory usage: 4.0+ MB
```

In [53]:

```
plt.hist(data['Diabetes'])
```

Out[53]:

```
(array([2738.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
        7262.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <a list of 10 Patch objects>)
```



In [54]:

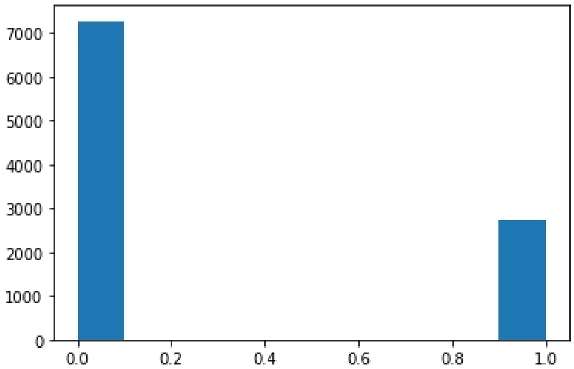
```
#Reexpression of 'Diabetes' data as numeric
data['Diabetes'] = data['Diabetes'].astype(str)
data['Diabetes'].replace(('Yes','No'), (1,0), inplace=True)
```

In [55]:

```
plt.hist(data['Diabetes'])
```

Out[55]:

```
(array([7262.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
        2738.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <a list of 10 Patch objects>)
```



In [56]:

```
#Reexpression of 'Hyperlipidemia' data as numeric
plt.hist(data['Hyperlipidemia'])
```

Out[56]:

```
(array([6628.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
        3372.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <a list of 10 Patch objects>)
```

In [57]:

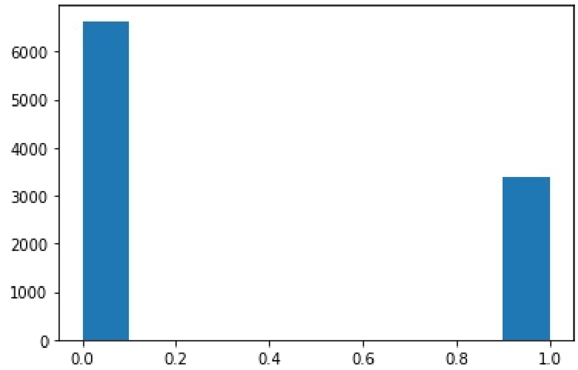
```
data['Hyperlipidemia'] = data['Hyperlipidemia'].astype(str)
data['Hyperlipidemia'].replace(('Yes','No'), (1,0), inplace=True)
```

In [58]:

```
plt.hist(data['Hyperlipidemia'])
```

Out[58]:

```
(array([6628.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
        3372.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <a list of 10 Patch objects>)
```

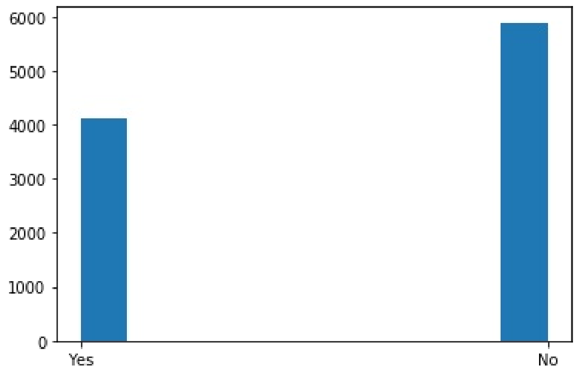


In [59]:

```
#Reexpression of 'Back_pain' data as numeric
plt.hist(data['Back_pain'])
```

Out[59]:

```
(array([4114.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
        5886.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <a list of 10 Patch objects>)
```



In [60]:

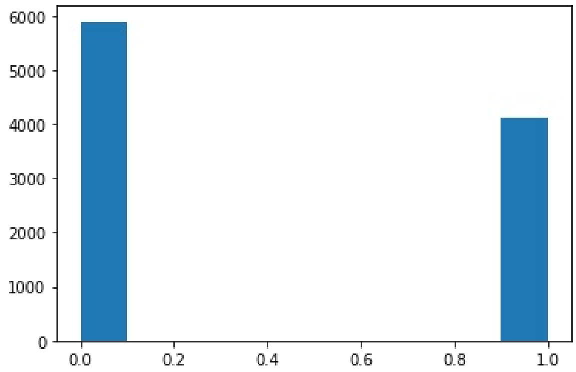
```
data['Back_pain'] = data['Back_pain'].astype(str)
data['Back_pain'].replace(('Yes','No'), (1,0), inplace=True)
```

In [61]:

```
plt.hist(data['Back_pain'])
```

Out[61]:

```
(array([5886.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
        4114.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <a list of 10 Patch objects>)
```

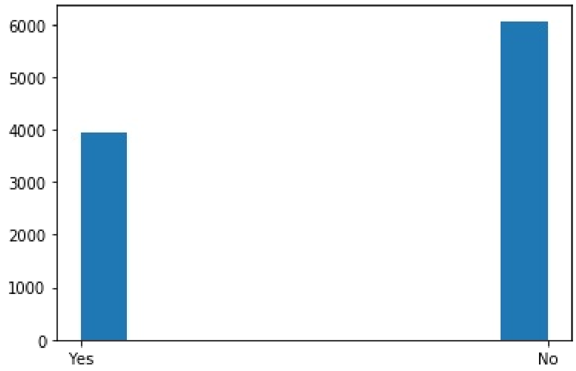


In [62]:

```
#Reexpression of 'Allergic_rhinitis' as numeric
plt.hist(data['Allergic_rhinitis'])
```

Out[62]:

```
(array([3941.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
        6059.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <a list of 10 Patch objects>)
```



In [63]:

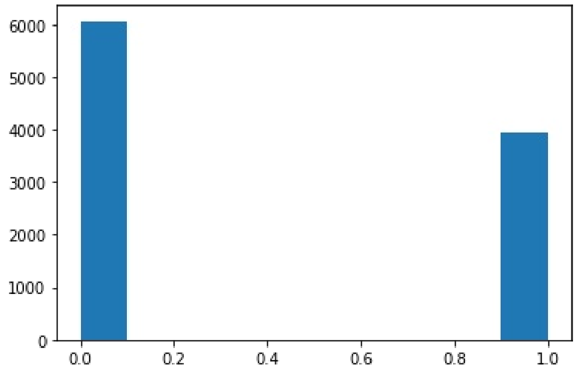
```
data['Allergic_rhinitis'] = data['Allergic_rhinitis'].astype(str)
data['Allergic_rhinitis'].replace(('Yes','No'), (1,0), inplace=True)
```

In [64]:

```
plt.hist(data['Allergic_rhinitis'])
```

Out[64]:

```
(array([6059.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
        3941.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <a list of 10 Patch objects>)
```



In [65]:

```
#Reexpression of 'Reflux_esophagitis' data as numeric
plt.hist(data['Reflux_esophagitis'])
```

Out[65]:

```
(array([5865.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
        4135.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <a list of 10 Patch objects>)
```

In [66]:

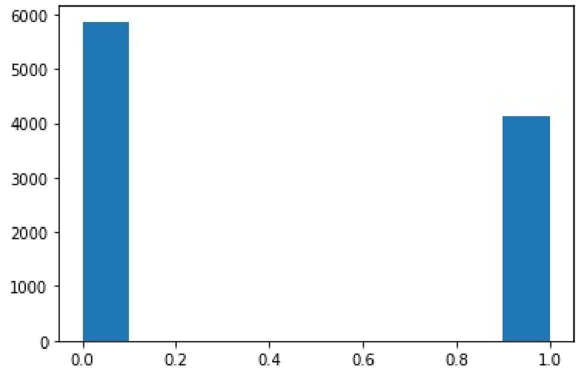
```
data['Reflux_esophagitis'] = data['Reflux_esophagitis'].astype(str)
data['Reflux_esophagitis'].replace(('Yes','No'), (1,0), inplace=True)
```

In [67]:

```
plt.hist(data['Reflux_esophagitis'])
```

Out[67]:

```
(array([5865.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
        4135.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <a list of 10 Patch objects>)
```

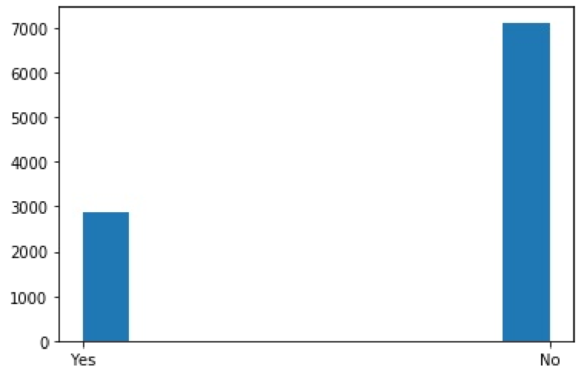


In [68]:

```
#Reexpression of 'Asthma' data as numeric.
plt.hist(data['Asthma'])
```

Out[68]:

```
(array([2893.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
        7107.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <a list of 10 Patch objects>)
```



In [69]:

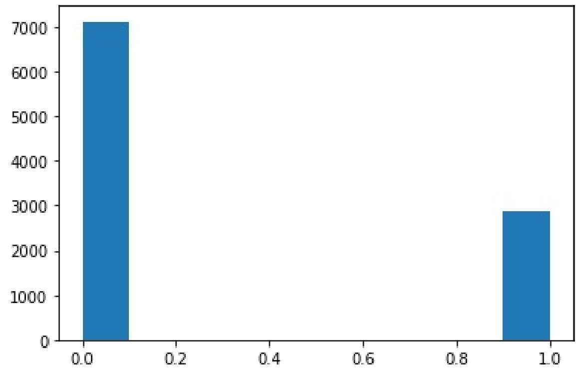
```
data['Asthma'] = data['Asthma'].astype(str)
data['Asthma'].replace(('Yes','No'), (1,0), inplace=True)
```

In [70]:

```
plt.hist(data['Asthma'])
```

Out[70]:

```
(array([7107.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
        2893.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <a list of 10 Patch objects>)
```

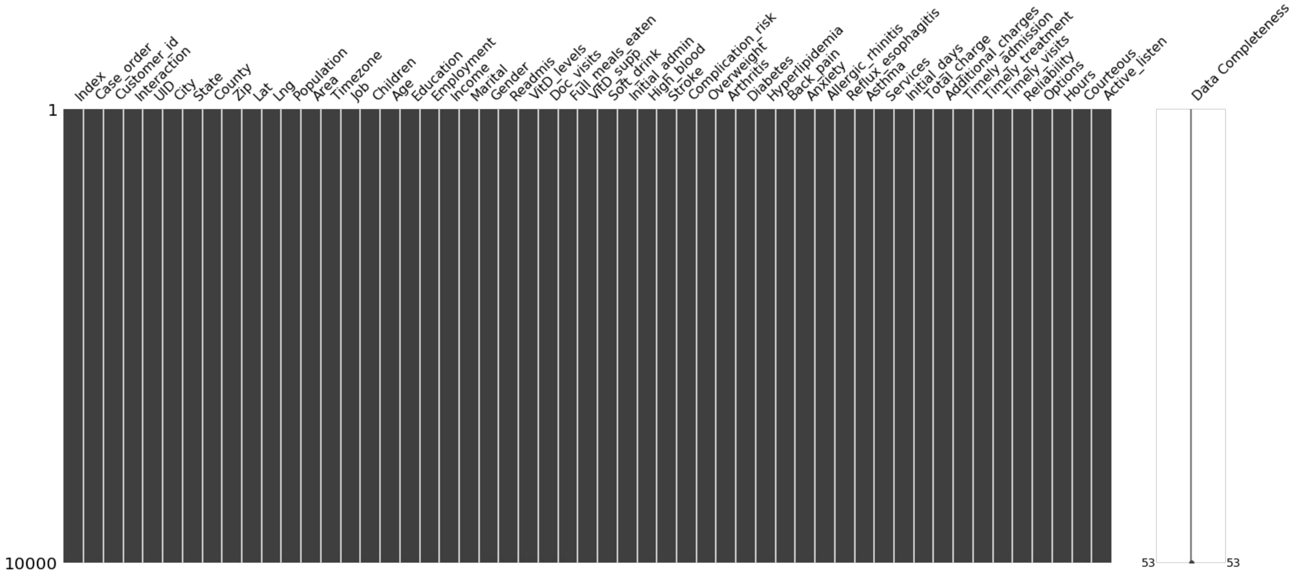


In [71]:

```
msno.matrix(data, labels=True)
```

Out[71]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f827311c08>
```



In [72]:

```
data.info()
```

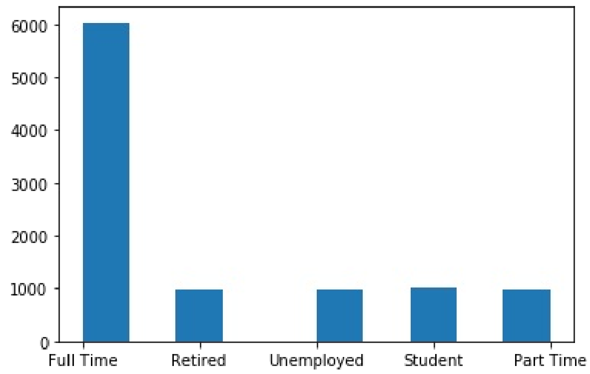
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 53 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Index               10000 non-null  int64  
 1   Case_order          10000 non-null  int64  
 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  int64  
 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  High_blood          10000 non-null  int64  
 30  Stroke              10000 non-null  int64  
 31  Complication_risk   10000 non-null  object 
 32  Overweight          10000 non-null  float64
 33  Arthritis           10000 non-null  int64  
 34  Diabetes            10000 non-null  int64  
 35  Hyperlipidemia      10000 non-null  int64  
 36  Back_pain           10000 non-null  int64  
 37  Anxiety             10000 non-null  float64
 38  Allergic_rhinitis   10000 non-null  int64  
 39  Reflux_esophagitis  10000 non-null  int64  
 40  Asthma              10000 non-null  int64  
 41  Services            10000 non-null  object 
 42  Initial_days        10000 non-null  float64
 43  Total_charge        10000 non-null  float64
 44  Additional_charges  10000 non-null  float64
 45  Timely_admission    10000 non-null  int64  
 46  Timely_treatment    10000 non-null  int64  
 47  Timely_visits       10000 non-null  int64  
 48  Reliability         10000 non-null  int64  
 49  Options             10000 non-null  int64  
 50  Hours               10000 non-null  int64  
 51  Courteous           10000 non-null  int64  
 52  Active_listen       10000 non-null  int64  
dtypes: float64(11), int64(25), object(17)
memory usage: 4.0+ MB
```

In [73]:

```
#Reexpress 'Employment' data as numeric.
plt.hist(data['Employment'])
```

Out[73]:

```
(array([6029.,    0.,  980.,    0.,    0.,  983.,    0., 1017.,    0.,
         991.]),
 array([0. , 0.4, 0.8, 1.2, 1.6, 2. , 2.4, 2.8, 3.2, 3.6, 4. ]),
 <a list of 10 Patch objects>)
```



In [74]:

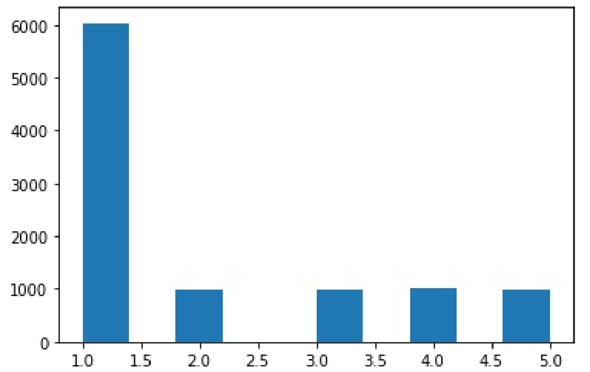
```
data['Employment'] = data['Employment'].astype(str)
data['Employment'].replace(('Full Time','Retired', 'Unemployed', 'Student', 'Part Time'), (1, 2, 3, 4, 5), inplac
e=True)
```

In [75]:

```
plt.hist(data['Employment'])
```

Out[75]:

```
(array([6029.,    0.,  980.,    0.,    0.,  983.,    0., 1017.,    0.,
         991.]),
 array([1. , 1.4, 1.8, 2.2, 2.6, 3. , 3.4, 3.8, 4.2, 4.6, 5. ]),
 <a list of 10 Patch objects>)
```

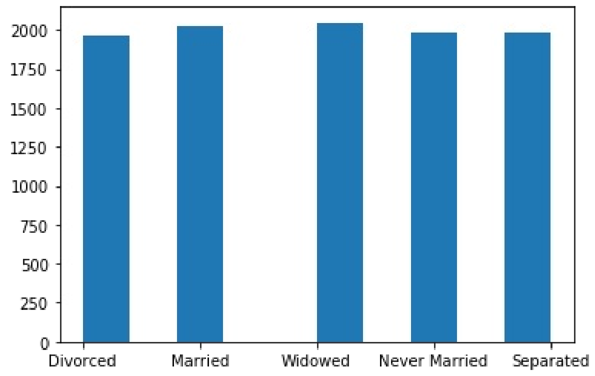


In [76]:

```
#Reexpress 'Marital' data as numeric
plt.hist(data['Marital'])
```

Out[76]:

```
(array([1961.,    0., 2023.,    0.,    0., 2045.,    0., 1984.,    0.,
        1987.]),
 array([0. , 0.4, 0.8, 1.2, 1.6, 2. , 2.4, 2.8, 3.2, 3.6, 4. ]),
 <a list of 10 Patch objects>)
```



In [77]:

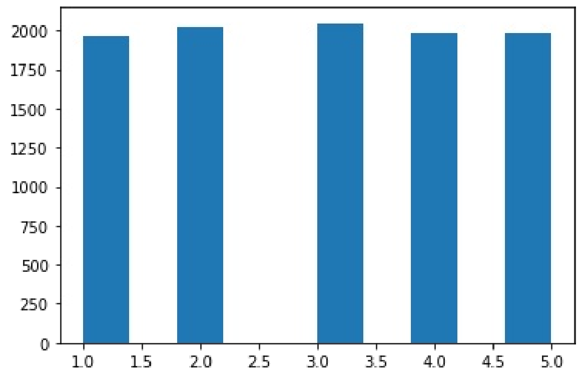
```
data['Marital'] = data['Marital'].astype(str)
data['Marital'].replace(('Divorced','Married', 'Widowed', 'Never Married', 'Separated'), (1, 2, 3, 4, 5), inplace
=True)
```

In [78]:

```
plt.hist(data['Marital'])
```

Out[78]:

```
(array([1961.,    0., 2023.,    0.,    0., 2045.,    0., 1984.,    0.,
        1987.]),
 array([1. , 1.4, 1.8, 2.2, 2.6, 3. , 3.4, 3.8, 4.2, 4.6, 5. ]),
 <a list of 10 Patch objects>)
```

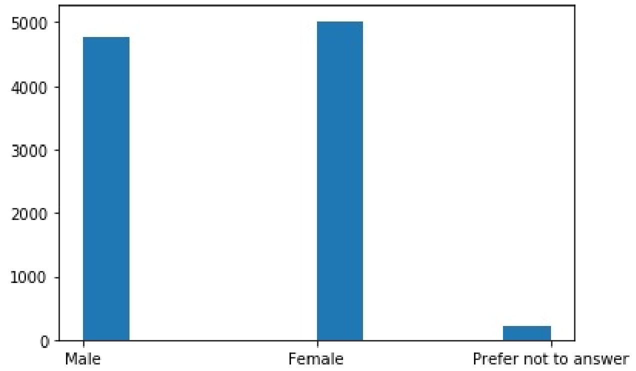


In [79]:

```
#Reexpress 'Gender' data as numeric
plt.hist(data['Gender'])
```

Out[79]:

```
(array([4768.,    0.,    0.,    0.,    0., 5018.,    0.,    0.,    0.,
         214.]),
 array([0. , 0.2, 0.4, 0.6, 0.8, 1. , 1.2, 1.4, 1.6, 1.8, 2. ]),
 <a list of 10 Patch objects>)
```



In [80]:

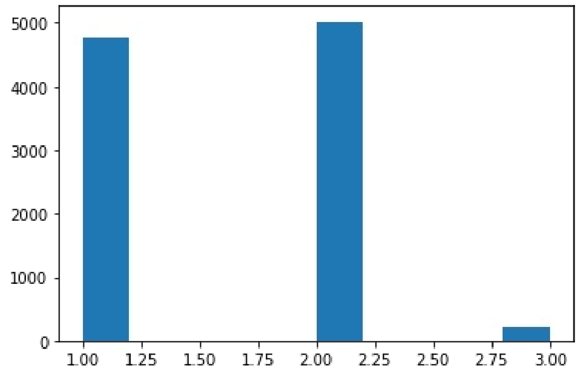
```
data['Gender'] = data['Gender'].astype(str)
data['Gender'].replace(('Male','Female', 'Prefer not to answer'), (1, 2, 3), inplace=True)
```

In [81]:

```
plt.hist(data['Gender'])
```

Out[81]:

```
(array([4768.,    0.,    0.,    0.,    0., 5018.,    0.,    0.,    0.,
         214.]),
 array([1. , 1.2, 1.4, 1.6, 1.8, 2. , 2.2, 2.4, 2.6, 2.8, 3. ]),
 <a list of 10 Patch objects>)
```

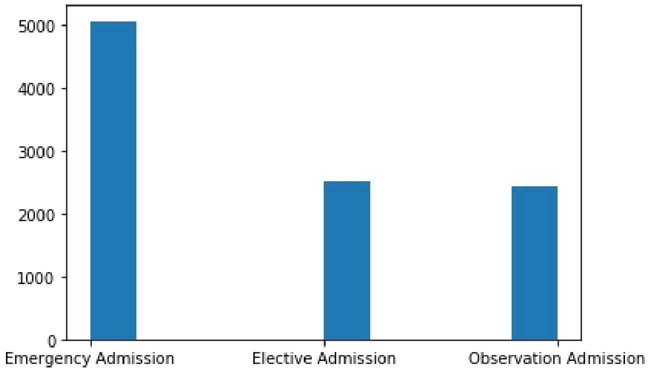


In [82]:

```
#Reexpress 'Initial_admin' as numeric data
plt.hist(data['Initial_admin'])
```

Out[82]:

```
(array([5060.,    0.,    0.,    0.,    0., 2504.,    0.,    0.,    0.,
        2436.]),
 array([0. , 0.2, 0.4, 0.6, 0.8, 1. , 1.2, 1.4, 1.6, 1.8, 2. ]),
 <a list of 10 Patch objects>)
```



In [83]:

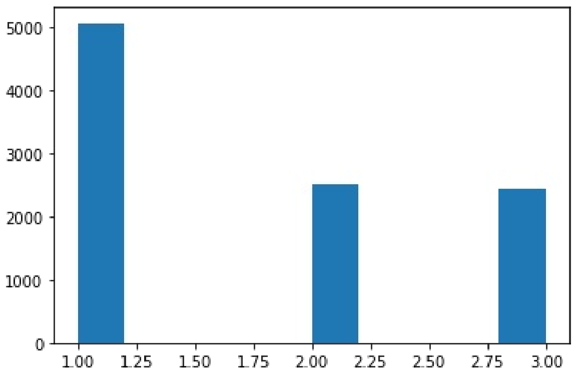
```
data['Initial_admin'] = data['Initial_admin'].astype(str)
data['Initial_admin'].replace(('Emergency Admission','Elective Admission', 'Observation Admission'), (1, 2, 3), i
nplace=True)
```

In [84]:

```
plt.hist(data['Initial_admin'])
```

Out[84]:

```
(array([5060.,    0.,    0.,    0.,    0., 2504.,    0.,    0.,    0.,
        2436.]),
 array([1. , 1.2, 1.4, 1.6, 1.8, 2. , 2.2, 2.4, 2.6, 2.8, 3. ]),
 <a list of 10 Patch objects>)
```

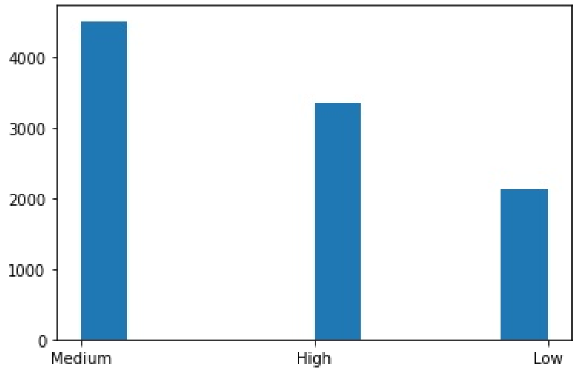


In [85]:

```
#Reexpress 'Complication_risk' data as numeric
plt.hist(data['Complication_risk'])
```

Out[85]:

```
(array([4517.,    0.,    0.,    0.,    0., 3358.,    0.,    0.,    0.,
        2125.]),
 array([0. , 0.2, 0.4, 0.6, 0.8, 1. , 1.2, 1.4, 1.6, 1.8, 2. ]),
 <a list of 10 Patch objects>)
```



In [86]:

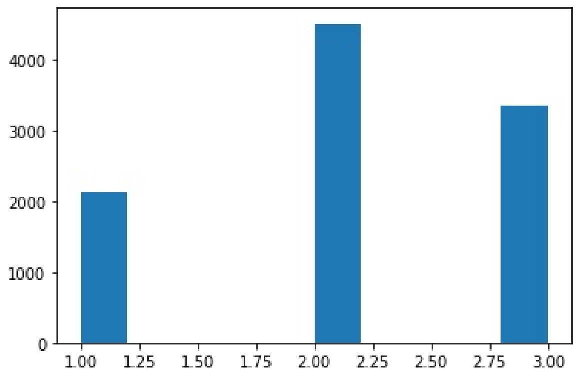
```
data['Complication_risk'] = data['Complication_risk'].astype(str)
data['Complication_risk'].replace(('Low','Medium', 'High'), (1, 2, 3), inplace=True)
```

In [87]:

```
plt.hist(data['Complication_risk'])
```

Out[87]:

```
(array([2125.,    0.,    0.,    0.,    0., 4517.,    0.,    0.,    0.,
        3358.]),
 array([1. , 1.2, 1.4, 1.6, 1.8, 2. , 2.2, 2.4, 2.6, 2.8, 3. ]),
 <a list of 10 Patch objects>)
```

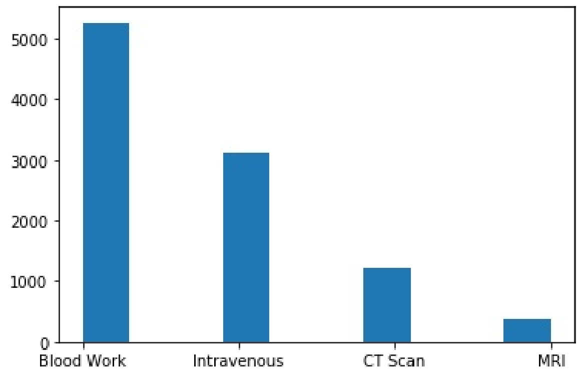


In [88]:

```
#Reexpress 'Services' data as numeric
plt.hist(data['Services'])
```

Out[88]:

```
(array([5265.,    0.,    0., 3130.,    0.,    0., 1225.,    0.,    0.,
         380.]),
 array([0. , 0.3, 0.6, 0.9, 1.2, 1.5, 1.8, 2.1, 2.4, 2.7, 3. ]),
 <a list of 10 Patch objects>)
```



In [89]:

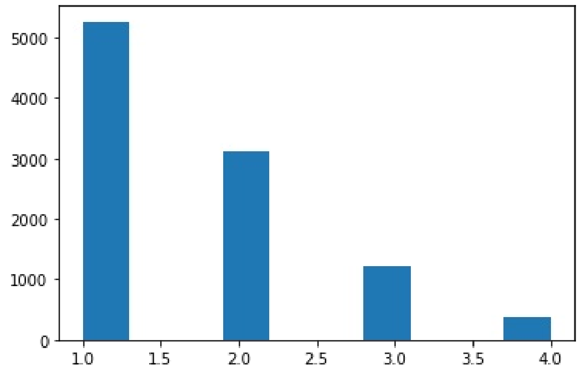
```
data['Services'] = data['Services'].astype(str)
data['Services'].replace(('Blood Work','Intravenous', 'CT Scan', 'MRI'), (1, 2, 3, 4), inplace=True)
```

In [90]:

```
plt.hist(data['Services'])
```

Out[90]:

```
(array([5265.,    0.,    0., 3130.,    0.,    0., 1225.,    0.,    0.,
         380.]),
 array([1. , 1.3, 1.6, 1.9, 2.2, 2.5, 2.8, 3.1, 3.4, 3.7, 4. ]),
 <a list of 10 Patch objects>)
```

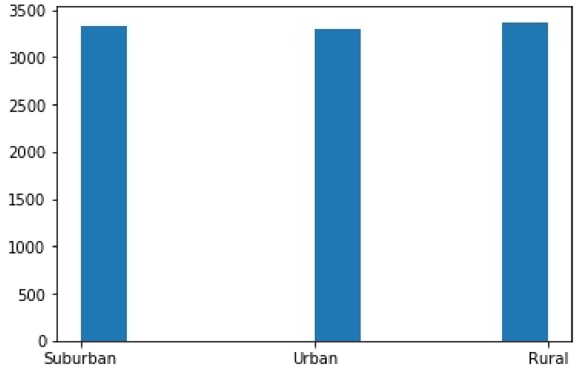


In [91]:

```
#Reexpress 'Area' data as numeric
plt.hist(data['Area'])
```

Out[91]:

```
(array([3328.,    0.,    0.,    0.,    0., 3303.,    0.,    0.,    0.,
        3369.]),
 array([0. , 0.2, 0.4, 0.6, 0.8, 1. , 1.2, 1.4, 1.6, 1.8, 2. ]),
 <a list of 10 Patch objects>)
```



In [92]:

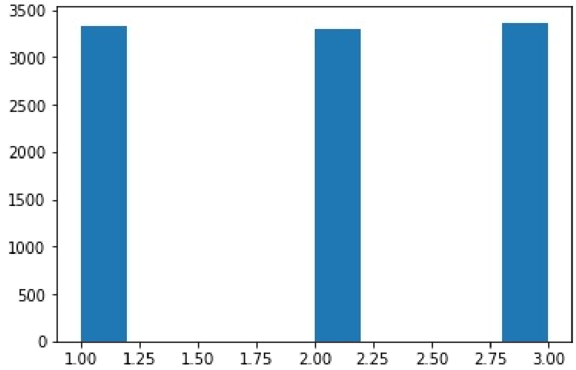
```
data['Area'] = data['Area'].astype(str)
data['Area'].replace(('Suburban','Urban', 'Rural'), (1, 2, 3), inplace=True)
```

In [93]:

```
plt.hist(data['Area'])
```

Out[93]:

```
(array([3328.,    0.,    0.,    0.,    0., 3303.,    0.,    0.,    0.,
        3369.]),
 array([1. , 1.2, 1.4, 1.6, 1.8, 2. , 2.2, 2.4, 2.6, 2.8, 3. ]),
 <a list of 10 Patch objects>)
```

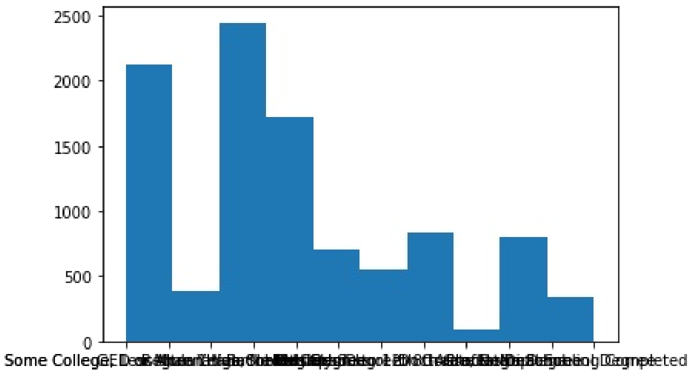


In [94]:

```python
#Reexpress 'Education' data as numeric
plt.hist(data['Education'])
```

Out[94]:

```
(array([2126.,  389., 2444., 1724.,  701.,  552.,  832.,   94.,  797.,
         341.]),
 array([ 0. ,  1.1,  2.2,  3.3,  4.4,  5.5,  6.6,  7.7,  8.8,  9.9, 11. ]),
 <a list of 10 Patch objects>)
```



In [95]:

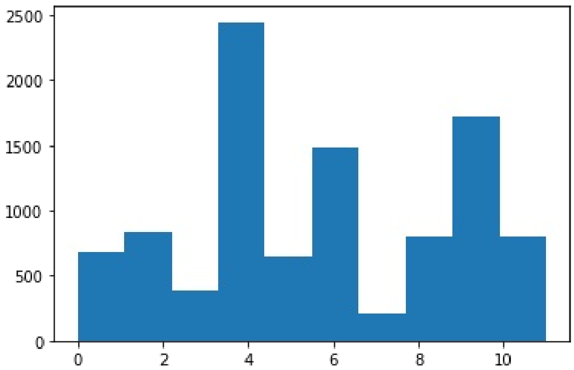
```python
data['Education'] = data['Education'].astype(str)
data['Education'].replace(('No Schooling Completed', 'Nursery School to 8th Grade', '9th Grade to 12th Grade, No Diploma', 'GED or Alternative Credential', 'Regular High School Diploma', 'Some College, Less than 1 Year', 'Some College, 1 or More Years, No Degree', 'Professional School Degree', 'Associate\'s Degree', 'Bachelor\'s Degree', 'Master\'s Degree', 'Doctorate Degree'), (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11), inplace=True)
```

In [96]:

```python
plt.hist(data['Education'])
```

Out[96]:

```
(array([ 685.,  832.,  389., 2444.,  642., 1484.,  208.,  797., 1724.,
         795.]),
 array([ 0. ,  1.1,  2.2,  3.3,  4.4,  5.5,  6.6,  7.7,  8.8,  9.9, 11. ]),
 <a list of 10 Patch objects>)
```



In [97]:

```python
data['Education'].unique()
```

Out[97]:

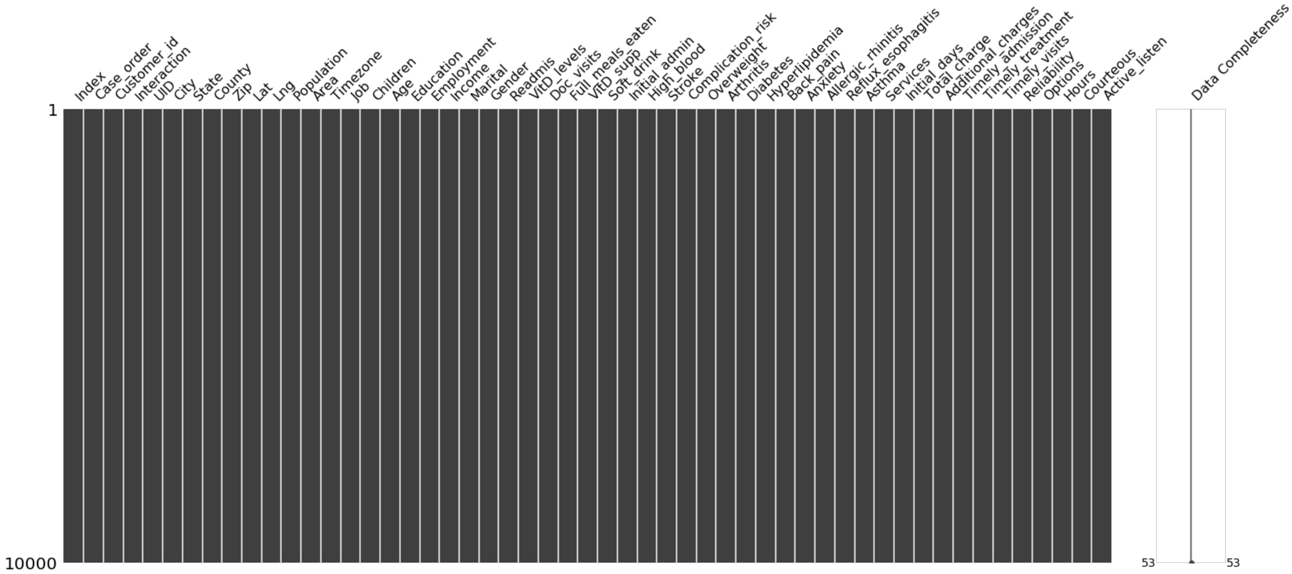
```
array([ 5,  6,  3,  4,  9, 10,  1,  2, 11,  8,  7,  0], dtype=int64)
```

In [98]:

```
msno.matrix(data, labels=True)
```

Out[98]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f82916b5c8>
```



In [99]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 53 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Index               10000 non-null  int64  
 1   Case_order          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  int64  
 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  int64  
 18  Employment          10000 non-null  int64  
 19  Income              10000 non-null  float64
 20  Marital             10000 non-null  int64  
 21  Gender              10000 non-null  int64  
 22  Readmis             10000 non-null  int64  
 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  int64  
 29  High_blood          10000 non-null  int64  
 30  Stroke              10000 non-null  int64  
 31  Complication_risk   10000 non-null  int64  
 32  Overweight          10000 non-null  float64
 33  Arthritis           10000 non-null  int64  
 34  Diabetes            10000 non-null  int64  
 35  Hyperlipidemia      10000 non-null  int64  
 36  Back_pain           10000 non-null  int64  
 37  Anxiety             10000 non-null  float64
 38  Allergic_rhinitis   10000 non-null  int64  
 39  Reflux_esophagitis  10000 non-null  int64  
 40  Asthma              10000 non-null  int64  
 41  Services            10000 non-null  int64  
 42  Initial_days        10000 non-null  float64
 43  Total_charge        10000 non-null  float64
 44  Additional_charges  10000 non-null  float64
 45  Timely_admission    10000 non-null  int64  
 46  Timely_treatment    10000 non-null  int64  
 47  Timely_visits       10000 non-null  int64  
 48  Reliability         10000 non-null  int64  
 49  Options             10000 non-null  int64  
 50  Hours               10000 non-null  int64  
 51  Courteous           10000 non-null  int64  
 52  Active_listen       10000 non-null  int64  
dtypes: float64(11), int64(33), object(9)
memory usage: 4.0+ MB
```

In [ ]:

```

```

In [100]:

```
data.to_csv('C:/Users/ericy/Desktop/D206_clean.csv')
```

In [101]:

```
#Round 'Income' case entries
data['Income'].round()
data['Income'] = data['Income'].astype('int64')
```

In [102]:

```
data['Income'].head()
```

Out[102]:

```
0    86575
1    46805
2    14370
3    39741
4     1209
Name: Income, dtype: int64
```

In [103]:

```
#Round 'VitD_levels' case entries
data['VitD_levels'].round()
data['VitD_levels'] = data['VitD_levels'].astype('int64')
data['VitD_levels'].head()
```

Out[103]:

```
0    17
1    18
2    17
3    17
4    16
Name: VitD_levels, dtype: int64
```

In [104]:

```
#Round 'Initial_days' case entries
data['Initial_days'].round()
data['Initial_days'] = data['Initial_days'].astype('int64')
data['Initial_days'].head()
```

Out[104]:

```
0    10
1    15
2     4
3     1
4     1
Name: Initial_days, dtype: int64
```

In [105]:

```
#Round 'Total_charge' case entries
data['Total_charge'].round()
data['Total_charge'] = data['Total_charge'].astype('int64')
data['Total_charge'].head()
```

Out[105]:

```
0    3191
1    4214
2    2177
3    2465
4    1885
Name: Total_charge, dtype: int64
```

In [106]:

```
#Round 'Additional_charges' case entries
data['Additional_charges'].round()
data['Additional_charges'] = data['Additional_charges'].astype('int64')
data['Additional_charges'].head()
```

Out[106]:

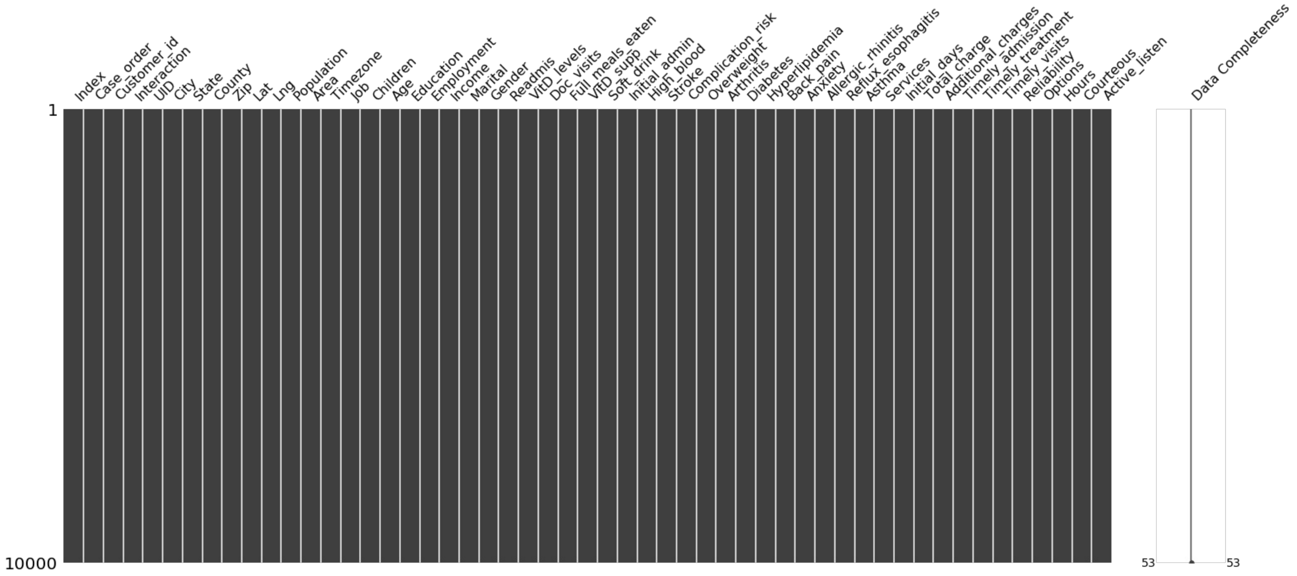
```
0    17939
1    17612
2    17505
3    12993
4     3716
Name: Additional_charges, dtype: int64
```

In [107]:

```
msno.matrix(data, labels=True)
```

Out[107]:

<matplotlib.axes._subplots.AxesSubplot at 0x1f829306d88>



In [108]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 53 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Index               10000 non-null  int64  
 1   Case_order          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  int64  
 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  int64  
 18  Employment          10000 non-null  int64  
 19  Income              10000 non-null  int64  
 20  Marital             10000 non-null  int64  
 21  Gender              10000 non-null  int64  
 22  Readmis             10000 non-null  int64  
 23  VitD_levels         10000 non-null  int64  
 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  int64  
 29  High_blood          10000 non-null  int64  
 30  Stroke              10000 non-null  int64  
 31  Complication_risk   10000 non-null  int64  
 32  Overweight          10000 non-null  float64
 33  Arthritis           10000 non-null  int64  
 34  Diabetes            10000 non-null  int64  
 35  Hyperlipidemia      10000 non-null  int64  
 36  Back_pain           10000 non-null  int64  
 37  Anxiety             10000 non-null  float64
 38  Allergic_rhinitis   10000 non-null  int64  
 39  Reflux_esophagitis  10000 non-null  int64  
 40  Asthma              10000 non-null  int64  
 41  Services            10000 non-null  int64  
 42  Initial_days        10000 non-null  int64  
 43  Total_charge        10000 non-null  int64  
 44  Additional_charges  10000 non-null  int64  
 45  Timely_admission    10000 non-null  int64  
 46  Timely_treatment    10000 non-null  int64  
 47  Timely_visits       10000 non-null  int64  
 48  Reliability         10000 non-null  int64  
 49  Options             10000 non-null  int64  
 50  Hours               10000 non-null  int64  
 51  Courteous           10000 non-null  int64  
 52  Active_listen       10000 non-null  int64  
dtypes: float64(6), int64(38), object(9)
memory usage: 4.0+ MB
```

In [109]:

```
data.to_csv('C:/Users/ericy/Desktop/D206_clean.csv')
```

In [110]:

```
#Convert 'Children', 'Age', 'Education', 'Readmis', 'Soft_drink', 'High_blood', 'Stroke',
#'Overweight', 'Arthritis', 'Diabetes', and 'Anxiety' to int64 datatype
data['Children'] = data['Children'].astype('int64')
data['Age'] = data['Age'].astype('int64')
data['Education'] = data['Education'].astype('int64')
data['Readmis'] = data['Readmis'].astype('int64')
data['Soft_drink'] = data['Soft_drink'].astype('int64')
data['High_blood'] = data['High_blood'].astype('int64')
data['Stroke'] = data['Stroke'].astype('int64')
data['Overweight'] = data['Overweight'].astype('int64')
data['Arthritis'] = data['Arthritis'].astype('int64')
data['Diabetes'] = data['Diabetes'].astype('int64')
data['Anxiety'] = data['Anxiety'].astype('int64')
```

In [111]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 53 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Index               10000 non-null  int64
 1   Case_order          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  int64
 13  Timezone            10000 non-null  object
 14  Job                 10000 non-null  object
 15  Children            10000 non-null  int64
 16  Age                 10000 non-null  int64
 17  Education           10000 non-null  int64
 18  Employment          10000 non-null  int64
 19  Income              10000 non-null  int64
 20  Marital             10000 non-null  int64
 21  Gender              10000 non-null  int64
 22  Readmis             10000 non-null  int64
 23  VitD_levels         10000 non-null  int64
 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  int64
 28  Initial_admin       10000 non-null  int64
 29  High_blood          10000 non-null  int64
 30  Stroke              10000 non-null  int64
 31  Complication_risk   10000 non-null  int64
 32  Overweight          10000 non-null  int64
 33  Arthritis           10000 non-null  int64
 34  Diabetes            10000 non-null  int64
 35  Hyperlipidemia      10000 non-null  int64
 36  Back_pain           10000 non-null  int64
 37  Anxiety             10000 non-null  int64
 38  Allergic_rhinitis   10000 non-null  int64
 39  Reflux_esophagitis  10000 non-null  int64
 40  Asthma              10000 non-null  int64
 41  Services            10000 non-null  int64
 42  Initial_days        10000 non-null  int64
 43  Total_charge        10000 non-null  int64
 44  Additional_charges  10000 non-null  int64
 45  Timely_admission    10000 non-null  int64
 46  Timely_treatment    10000 non-null  int64
 47  Timely_visits       10000 non-null  int64
 48  Reliability         10000 non-null  int64
 49  Options             10000 non-null  int64
 50  Hours               10000 non-null  int64
 51  Courteous           10000 non-null  int64
 52  Active_listen       10000 non-null  int64
dtypes: float64(2), int64(43), object(8)
memory usage: 4.0+ MB
```

In [112]:

```
data.to_csv('C:/Users/ericy/Desktop/D206_clean.csv', index=False)
data.to_csv('C:/Users/ericy/Desktop/data_z.csv', index=False)
data.to_csv('C:/Users/ericy/Desktop/PCA_Ready.csv', index=False)
```

In [113]:

```
#Assign variable to dataset for calculating z scores
#Calculate Z-Scores for all quantitative variables
dataz = pd.read_csv('C:/Users/ericy/Desktop/data_z.csv')
dataz.info()
dataz['Age_z'] = stats.zscore(data['Age'])
Agez = dataz.query('Age_z > 3 | Age_z < -3')
Agez.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 53 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Index               10000 non-null  int64  
 1   Case_order          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  int64  
 13  Timezone            10000 non-null  object 
 14  Job                 10000 non-null  object 
 15  Children            10000 non-null  int64  
 16  Age                 10000 non-null  int64  
 17  Education           10000 non-null  int64  
 18  Employment          10000 non-null  int64  
 19  Income              10000 non-null  int64  
 20  Marital             10000 non-null  int64  
 21  Gender              10000 non-null  int64  
 22  Readmis             10000 non-null  int64  
 23  VitD_levels         10000 non-null  int64  
 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  int64  
 28  Initial_admin       10000 non-null  int64  
 29  High_blood          10000 non-null  int64  
 30  Stroke              10000 non-null  int64  
 31  Complication_risk   10000 non-null  int64  
 32  Overweight          10000 non-null  int64  
 33  Arthritis           10000 non-null  int64  
 34  Diabetes            10000 non-null  int64  
 35  Hyperlipidemia      10000 non-null  int64  
 36  Back_pain           10000 non-null  int64  
 37  Anxiety             10000 non-null  int64  
 38  Allergic_rhinitis   10000 non-null  int64  
 39  Reflux_esophagitis  10000 non-null  int64  
 40  Asthma              10000 non-null  int64  
 41  Services            10000 non-null  int64  
 42  Initial_days        10000 non-null  int64  
 43  Total_charge        10000 non-null  int64  
 44  Additional_charges  10000 non-null  int64  
 45  Timely_admission    10000 non-null  int64  
 46  Timely_treatment    10000 non-null  int64  
 47  Timely_visits       10000 non-null  int64  
 48  Reliability         10000 non-null  int64  
 49  Options             10000 non-null  int64  
 50  Hours               10000 non-null  int64  
 51  Courteous           10000 non-null  int64  
 52  Active_listen       10000 non-null  int64  
dtypes: float64(2), int64(43), object(8)
memory usage: 4.0+ MB
```

Out[113]:

Index	Case_order	Customer_id	Interaction	UID	City	State	County	Zip	Lat	...	Additional_charges	Timely_admission	Timely_treat

0 rows × 54 columns

In [114]:

```python
dataz['Children_z'] = stats.zscore(dataz['Children'])
Childrenz = dataz.query('Children_z > 3 | Children_z < -3')
Childrenz.sort_values(['Children_z'], ascending = False)
```

Out[114]:

	Index	Case_order	Customer_id	Interaction	UID	City	State	County	Zip	La
16	16	17	O377757	7faf0261-bc66-489a-a8ba-fec333485254	728333940561457a9feba1e1dc763258	Blythe	CA	Riverside	92225	33.7464
1093	1093	1094	U798396	ded17fc4-27d2-4fce-a7e4-c3b27427ff0b	9d56a350bcbd02ad66629ce06080ef32	Rock Hill	SC	York	29730	34.8867
6484	6484	6485	A961890	00730262-8847-4a35-9a79-f8d247ee57e8	ef304b3a546a70fb216abde377fbe688	Loomis	CA	Placer	95650	38.8128
2124	2124	2125	T948257	17f13f8d-8c16-47eb-9e6c-c5635f97dcc8	672cf906ec22dbec40674cb78f837e44	Mullen	NE	Hooker	69152	42.1092
2121	2121	2122	E859932	2d86af54-e54e-485f-8d06-01de27a966d9	432f2a96712dfd4d2f56965c4805bb68	Madison	AR	St. Francis	72359	35.0228
...	...	...	...	...	...	...	...	...	...	..
6112	6112	6113	Z417502	05902a2c-b76f-4ab6-b46d-857c58cf6da7	30695977947a77889822d153007e8eb2	Chicago	IL	Cook	60653	41.8192
6174	6174	6175	M717683	14ea5131-9ce6-4417-8e3f-99467287ff45	47525dc30f2e86cabbaad3e3bd4a7406	Plainview	AR	Yell	72857	34.8577
2524	2524	2525	A541545	78c32463-77cd-4121-8d1c-e6764c6757fa	1c3e490f7ad5a81bf8f8ba7937e2221b	Lonsdale	MN	Rice	55046	44.4486
2487	2487	2488	L515011	e5b1b81c-917f-434f-84ae-ed6ffb281020	fced486b7771e55cfb7d800cb0e0fc04	Beavercreek	OR	Clackamas	97004	45.2511
9999	9999	10000	I569847	bc482c02-f8c9-4423-99de-3db5e62a18d5	95663a202338000abdf7e09311c2a8a1	Coraopolis	PA	Allegheny	15108	40.4999

303 rows × 55 columns

In [115]:

```python
dataz['Income_z'] = stats.zscore(dataz['Income'])
Incomez = dataz.query('Income_z > 3 | Income_z < -3')
Incomez.sort_values(['Income_z'], ascending = False)
```

Out[115]:

	Index	Case_order	Customer_id	Interaction	UID	City	State	County	Zip	L
8386	8386	8387	C817840	41770631-ff8b-4e71-9631-f369b04d2125	f81e41a00a41c04266e666361ad49a33	Phoenix	AZ	Maricopa	85044	33.342
841	841	842	F304162	cbd20767-266b-470b-9bd7-9b8aab96da38	3424165edc18b296b6ec24d69101a2a9	Galloway	WV	Barbour	26349	39.235
8598	8598	8599	C730234	bb1cdec6-187d-40ac-bcb2-1544f5bb4b1d	609d3ae46250dffa60021c1f62169869	Haywood	VA	Madison	22722	38.461
6406	6406	6407	J423842	fe003dd7-d9b2-4cc0-b446-fc0c48cdabea	b481a4d89ab6871d664e7f917393a5ba	Scranton	PA	Lackawanna	18504	41.425
1778	1778	1779	T848406	3c57ca24-c58c-45b0-a96f-928187a615d0	73fffc542bdeb8f39051413f55972023	Mowrystown	OH	Highland	45155	39.039
...	...	...	...	...	...	...	...	...	...	
7697	7697	7698	S906499	2d880cc6-37c3-4b7c-90d7-6fc3074d19eb	c4a72a9475c3d22d40ae6d483e8a5867	Lynnfield	MA	Essex	1940	42.534
3702	3702	3703	D875126	736613d8-eb00-488f-8e93-2c3f7d939c0f	0e9eb923d8ddf8ecca0db017fa1e99d8	Byron	MN	Olmsted	55920	44.013
86	86	87	E681129	78216a6f-87fe-45a8-8e76-a8abbe2adff2	ad555647d06822a4a1259340c2e21c5f	Caroleen	NC	Rutherford	28019	35.280
3017	3017	3018	Y624229	b920591f-01cb-4320-af69-d62ca61ae8e2	886eb2d72c29fe788232f6b36bd37f08	Sulphur Springs	AR	Benton	72768	36.476
2507	2507	2508	S453939	004cdcae-763f-4fe8-82c9-46bab6bf7011	37a0b6995354a733063ab3566e171a67	Selma	NC	Johnston	27576	35.582

180 rows × 56 columns

In [116]:

```
dataz['VitD_levels_z'] = stats.zscore(dataz['VitD_levels'])
VitD_levels_z = dataz.query('VitD_levels_z > 3 | VitD_levels_z < -3')
VitD_levels_z.sort_values(['VitD_levels_z'], ascending = False)
```

Out[116]:

	Index	Case_order	Customer_id	Interaction	UID	City	State	County	Zip	Lat	..
1963	1963	1964	J288779	d643d57b-cebb-4556-ac38-8b339b85175d	7305ac02547eb73b8a5e30855b602e99	Jean	NV	Clark	89019	35.76620	..
3473	3473	3474	Y739652	6dbae289-4c4d-4157-9fbf-4bc4665c12fd	0f46805163c147c5bc70ef76b46be56a	Concord	CA	Contra Costa	94521	37.95603	..
2615	2615	2616	S997798	8888fd85-4442-48f6-924d-858c30e733d0	4876750cae50b72e92b19e2213b1371c	Harris	MO	Sullivan	64645	40.29741	..
7157	7157	7158	L397900	85cc282c-0b16-404b-8f15-6b7ac633c2d6	2b091704732658b36d1a37c3674e69a0	Jobstown	NJ	Burlington	8041	40.03788	..
1306	1306	1307	B77596	e19f375b-b1ea-44b9-a0e3-bcf3bf4b4bc1	bcd4395e7916ffa8e6659f9c563f56ea	Holualoa	HI	Hawaii	96725	19.62925	..
...	...	...	...	...	...	...	...	...	...	...	..
786	786	787	U179768	1ca91d4f-9f12-4095-907a-3c81afb93207	266a7c39f532ce5455c4ac68a615d003	Hatteras	NC	Dare	27943	35.21097	..
7270	7270	7271	M212963	b9e0709b-a602-41a5-b3b4-229576f57952	758d7f97dea2ebe1c585733bd100e86f	Augusta	GA	Richmond	30905	33.41474	..
5688	5688	5689	Q71266	2179cd1f-a3b0-4ee7-a53a-35a3632bf291	ba620f7005481bb1641cbac29720efea	Duarte	CA	Los Angeles	91010	34.14074	..
2946	2946	2947	T519902	50542ca6-d2bb-4ead-a3c1-d8194eaab696	23bc2fc77e42f0a89e7bbef583f9cd9d	Dayton	WA	Columbia	99328	46.25660	..
8197	8197	8198	U547343	6150f8d8-e206-462b-bd81-930c7fb8aef1	cf0e28edb667f9e5166f0287a7e5ef07	Old Bethpage	NY	Nassau	11804	40.75874	..

500 rows × 57 columns

In [117]:

```
dataz['Doc_visits_z'] = stats.zscore(dataz['Doc_visits'])
Doc_visits_z = dataz.query('Doc_visits_z > 3 | Doc_visits_z < -3')
Doc_visits_z.sort_values(['Doc_visits_z'], ascending = False)
```

Out[117]:

	Index	Case_order	Customer_id	Interaction	UID	City	State	County	Zip	Lat	...
962	962	963	A518996	a38c4ad7-323f-41f2-9a08-ff17743aaa53	4112b686f622313e4d247da0b9a2afb4	Uvalde	TX	Uvalde	78801	29.35664	...
2766	2766	2767	N924859	5a334d2f-a78d-4165-aa83-d368bb82fa48	a18d1b5abef353e496b4d25926c0d213	Walton	OR	Lane	97490	44.00425	...
5645	5645	5646	H849940	d5b2f306-7c65-4ad0-8d9b-3b144bd20c34	e5e3073cdab4a0e7ba03a174660cb5b2	Faber	VA	Nelson	22938	37.86065	...
5756	5756	5757	Q856766	abf1c636-143b-4f87-a663-82c1ac92bbd2	5046bbcc46fbcccf4f6b76c7e5b71082	Toronto	OH	Jefferson	43964	40.48617	...
6017	6017	6018	Z448538	a44eb330-7119-4c47-a0c0-356b2d481587	42e4405346a43a9c60a2acf63718f235	Collins	WI	Manitowoc	54207	44.08782	...
6498	6498	6499	D695903	7e305136-22ea-4d2d-a92d-39132c1bf66b	994f0adf59f7a7d9bb53bb296058be3b	Douglas	OK	Garfield	73733	36.25360	...
6942	6942	6943	W120936	2d981e21-86cd-4880-b731-a5c0d5a2c2bb	2fd0a3063b109969d378e61c19c081c0	Noonan	ND	Divide	58765	48.87743	...
7143	7143	7144	K252805	dc0772b4-e146-492f-8537-8c02679d553f	bf9248b12adbe35b728debdf7f00b68e	El Paso	TX	El Paso	79907	31.70750	...

8 rows × 58 columns

In [118]:

```
dataz['Full_mealz'] = stats.zscore(dataz['Full_meals_eaten'])
Full_mealz = dataz.query('Full_mealz > 3 | Full_mealz < -3')
Full_mealz.sort_values(['Full_mealz'], ascending = False)
```

Out[118]:

	Index	Case_order	Customer_id	Interaction	UID	City	State	County	Zip	
958	958	959	Y657696	c7a8a8b7-5d61-4d95-872f-58f3bb589c09	30703ca82ae5ed6da3addcd421777c38	Sebastopol	CA	Sonoma	95472	38.3
4709	4709	4710	F767195	7da332b0-bc0f-4486-a973-c960376154aa	bda40730190467bcfc2b4ce70b727a71	Leopold	MO	Bollinger	63760	37.2
9986	9986	9987	Z630066	1ed0ed27-4965-4252-85ea-dd7ed73bd51a	f132eca4af3b1c955d89a213096ef88a	Perry	IA	Dallas	50220	41.8
7217	7217	7218	M529189	73802f3c-f978-4b67-ba2c-844140ac7e41	b200ab915ceeee74bceab2dc0ad39121	Ashton	IA	Osceola	51232	43.3
6068	6068	6069	Z447871	56d5bab6-25c0-4c30-a3f9-822ff27def3e	2db6d1a15351eaa51b4370872cafab51	Constableville	NY	Lewis	13325	43.5
1231	1231	1232	J394932	e625f515-a366-4b95-8e88-9dc3afda79d8	ea252a0d3bcd2272a60a658b7cf21b29	Bay Shore	NY	Suffolk	11706	40.7
2184	2184	2185	H40270	30bfc529-4c99-4244-b3fa-b1828e591622	e8b301a00be4e22f809745e50b684b28	Waynesville	GA	Brantley	31566	31.1

8144	8144	8145	G557244	25861106-d9bf-4744-8f7f-952bdef14ace	eff4a060f579f85142c6ea7ca3884435	Mangham	LA	Richland	71259	32.2
6083	6083	6084	T927706	1d3b5fc2-3a3b-4138-a2f8-b65e93d26125	cb97fffaa14fa2a4499cdaf32ae81f22	Highland	MI	Oakland	48356	42.6
6694	6694	6695	C327638	e9cfdc20-d85d-4c1f-9e35-a0714c341760	64b2d0f9910d479999267b0ecc142da2	Davison	MI	Genesee	48423	43.0
6802	6802	6803	G952688	1f4bf3cd-6419-4b0e-b1e1-91b44601ff79	10fb646b9851135d5578c03983ce924c	Hillpoint	WI	Sauk	53937	43.3
8326	8326	8327	P966922	ba29b074-2909-4ac0-ae8c-3d98132c1bb5	a0aefe75fb9316a55e02d0f11bed7c73	Hillsboro	MD	Caroline	21641	38.9
5859	5859	5860	I304713	a2a1010e-bdd1-4d85-95f8-ca765bd1777d	d01e611cc208e9b2ed5306fc36bba740	Lincoln	NE	Lancaster	68510	40.8
8902	8902	8903	L332623	65bd9f6f-5f20-4e8b-b155-c448edb96e4f	d6d5dab162b78d68aa561e09a763f5a8	Virginia Beach	VA	Virginia Beach	23456	36.7
8994	8994	8995	N415828	e089be60-b2ba-4d32-a086-77c5895e6516	c5367056ea7b4fcdcd2907135bee3e79	Odessa	TX	Ector	79766	31.7
9067	9067	9068	I917390	962d0ec6-27e9-4010-b59a-af649d02a475	3ed7b9e4969092fd2b6d49c3b421d494	Canaseraga	NY	Allegany	14822	42.4
9220	9220	9221	C513727	91d192f4-9f40-47d3-8a35-4d6c0b4dcb0a	0c5424d0c0c4877f43d9129966ba4a81	East Montpelier	VT	Washington	5651	44.2
6026	6026	6027	M688413	fc55b7d4-5d5f-47ce-872d-19b28001adf4	2eb28fd3fae40faeb134019216fe4f9b	Huger	SC	Berkeley	29450	33.0
550	550	551	K368670	ee6e63d6-b073-4f56-9751-5933049da455	8d4f1906f9ce5eff77b1d790fa7ed95f	Fort Covington	NY	Franklin	12937	44.9
5711	5711	5712	G268057	3abc06f2-a987-4242-a08d-ff8cadf45365	e6825550973f181bca9c8a8e5f02ef81	Mercer Island	WA	King	98040	47.5
5597	5597	5598	V944194	38f18892-2a10-41c2-8e11-49bf96c4bd1f	99679c97db9658487cdea25b37a173f9	Bradford	VT	Orange	5033	44.0
697	697	698	F454155	1592fb46-79d4-4b63-8ce9-f25374e8c8d4	467b7a7e36f1a274388dfe83f47fb2ba	Diamond	MO	Newton	64840	37.0
5367	5367	5368	O11669	0ae8ebb4-846a-4c59-acc5-7e2c90640cfe	6c43df9bb0d357055173d45dd8907ecf	Rumely	MI	Alger	49826	46.3
4902	4902	4903	X275889	46cca6f8-5e68-4662-8b41-a82c03d97719	084fc0fd364584d5f88ac080fac3f087	Aurora	MN	St. Louis	55705	47.4
4345	4345	4346	Q413439	46264e69-39c9-41af-9876-02ee2159ad63	2a86f887f3306f030dc96d91f57b07ba	Grand Lake	CO	Grand	80447	40.2
2919	2919	2920	Y483255	3af744a4-3e8f-4baa-b3fe-402953395fef	a67c72e0926293aaa5ec33727f37a7e5	Laotto	IN	Noble	46763	41.2
2877	2877	2878	I684405	4f10c57b-d053-4ea6-b52c-0313f3f130b2	6330a5d3563e97a21e3ed67cf941f7a7	Tranquillity	CA	Fresno	93668	36.6
2746	2746	2747	M406925	3d584b8b-8269-4ef5-8f9c-4e2d934eabea	c97a081eb0f5d3bf57b64eba126732e9	Kent	OH	Portage	44243	41.1

2652	2652	2653	B388915	4eec34de-c681-4901-b640-854cc7f32ebe	df2c4002e9d0dcb589b326b997d9c762	Boonville	CA	Mendocino	95415	39.0
2315	2315	2316	E105778	234af304-1c20-4578-9251-0648e8126ead	4814abcb5f4f0f2fe482feb98ac27f98	Crawford	WV	Lewis	26343	38.8
1456	1456	1457	V65457	e8c0a2d8-05df-4c82-a3c5-e93ee72ac9b5	a131bb7a298e31324f6a7f99c2714d24	Syracuse	NY	Onondaga	13214	43.0
1148	1148	1149	F466335	bfa9aa23-ec57-4b95-abbf-4402007b0a5b	755781fcb9b85ed2e5e675895fa810bb	Spring Glen	NY	Ulster	12483	41.6
5543	5543	5544	C451388	3e4e410f-f60e-4966-a3bd-1f604dafbf35	431621904d47fb4feee40a28bd421c0a	San Antonio	TX	Comal	78266	29.6

33 rows × 59 columns

In [119]:

```
dataz['VitD_suppz'] = stats.zscore(dataz['VitD_supp'])
VitD_suppz = dataz.query('VitD_suppz > 3 | VitD_suppz < -3')
VitD_suppz.sort_values(['VitD_suppz'], ascending = False)
```

Out[119]:

	Index	Case_order	Customer_id	Interaction	UID	City	State	County	Zip	
3131	3131	3132	A693543	c2eef231-ba8b-4f2a-b7bb-5722189fbe4b	19716f3f690b579b5dcdb771550f9b5c	Washington	KS	Washington	66968	39.82
2715	2715	2716	P60898	f66b928a-6de9-4043-b2d2-8cf0bfac0b34	52e8f2a6e67c326ce495e89cf8e3391a	Bainbridge	IN	Putnam	46105	39.76
9091	9091	9092	A771264	4676ed64-981d-48c8-bf78-6706c592e4fd	f77af82fc41b30951e69e437feb63ca5	Rio Nido	CA	Sonoma	95471	38.52
1342	1342	1343	X97640	c6680d61-0228-44dc-bac6-7f71492b5daf	15fc59be69381309db87a023bce971cd	Franklin Square	NY	Nassau	11010	40.70
2533	2533	2534	H623137	7dffab81-3be0-4a66-8aad-4d04a9e08ed9	c7b63686ec434c9059203bb28fe3cea1	Lonsdale	MN	Rice	55046	44.44
...	...	...	...	...	...	...	...	...	...	
4398	4398	4399	P241002	c43abb2f-d03f-4f94-b85c-b94f9b87ba8e	1a1d72d68cbe08b21e7a2e69b0db05a6	Honolulu	HI	Honolulu	96816	21.29
4406	4406	4407	Y884211	6559cede-c035-452a-a0cf-41f6ea72ffda	fd9e9c25844f187a3903254ac48a87b4	Glasco	NY	Ulster	12432	42.04
4567	4567	4568	M822122	3824bf42-5578-4c6a-bddf-e7c653e57fd8	10f56b5cfa41d592fca272b95903a775	Oklahoma City	OK	Oklahoma	73107	35.48
4844	4844	4845	L06840	07c93832-d655-440e-b039-030796cb9d72	48bbbf13517022e7f004b71473afddcf	Hartford	SD	Minnehaha	57033	43.61
9982	9982	9983	O64996	07ffe436-a1a2-4b37-96b0-2602ffb1ad6f	b0df4c12776c7d9efceb9fcc67d0262e	Atlantic City	NJ	Atlantic	8401	39.37

70 rows × 60 columns

In [120]:

```
dataz['Initial_days_z'] = stats.zscore(dataz['Initial_days'])
Initial_days_z = dataz.query('Initial_days_z > 3 | Initial_days_z < -3')
Initial_days_z.sort_values(['Initial_days_z'], ascending = False)
```

Out[120]:

Index	Case_order	Customer_id	Interaction	UID	City	State	County	Zip	Lat	...	Courteous	Active_listen	Age_z	Children_z	Incon

0 rows × 61 columns

In [121]:

```
dataz['Total_charge_z'] = stats.zscore(dataz['Total_charge'])
Total_charge_z = dataz.query('Total_charge_z > 3 | Total_charge_z < -3')
Total_charge_z.sort_values(['Total_charge_z'], ascending = False)
```

Out[121]:

	Index	Case_order	Customer_id	Interaction	UID	City	State	County	Zip	
8800	8800	8801	I804892	3fc45464-51e3-4182-ba05-0e960ddff205	815273eb63baa4ef16b596d02cfb92de	Weyanoke	LA	West Feliciana	70787	30.96
9005	9005	9006	G175531	56066f7f-5a32-4732-965a-9e92af4ffb0b	331b2187466de52359ae522cb8b48f8f	Lampe	MO	Stone	65681	36.55
5244	5244	5245	Y654860	11e8adb4-b54b-4672-b122-16b2efd33943	00a99d284dcb8bafd8bae0ee83314e77	Elko	GA	Houston	31025	32.34
5453	5453	5454	N354417	3af75e11-a617-493d-aef8-c799d51ad04b	cf2d345d5c28fd232a947fff963496c7	Elkins	NH	Merrimack	3233	43.42
9159	9159	9160	G163318	d4613a36-798a-4258-9106-e7769c8b9e13	c6f3559b9c756e76dabad184af795a4e	Sunnyvale	TX	Dallas	75182	32.80
...	...	...	...	...	...	...	...	...	...	
1963	1963	1964	J288779	d643d57b-cebb-4556-ac38-8b339b85175d	7305ac02547eb73b8a5e30855b602e99	Jean	NV	Clark	89019	35.76
2999	2999	3000	V552831	0d02edda-1ffe-4388-a126-a9f073151711	ede05577efbb3df8b43befa16a7fb9bf	East Hampstead	NH	Rockingham	3826	42.88
1847	1847	1848	E836671	9beaf0ae-8bb0-4788-baf7-9b1936f94e5a	7a4b3164770176c1f1df08872cf835d3	Rockville	IN	Parke	47872	39.75
3350	3350	3351	M549619	7f6cd69d-9319-4ac2-b7de-8a011a9525c5	b34915d6185285bb08a88575c22176fe	Towanda	IL	McLean	61776	40.56
527	527	528	I117310	39114acf-c2f5-4971-8fbc-88d830367e98	512e1a052e1043902a81b535e80ae309	Portland	OR	Clackamas	97267	45.40

276 rows × 62 columns

In [122]:

```
dataz['Additional_charges_z'] = stats.zscore(dataz['Additional_charges'])
Additional_charges_z = dataz.query('Additional_charges_z > 3 | Additional_charges_z < -3')
Additional_charges_z.sort_values(['Additional_charges_z'], ascending = False)
```

Out[122]:

Index	Case_order	Customer_id	Interaction	UID	City	State	County	Zip	Lat	...	Age_z	Children_z	Income_z	VitD_levels_z	Doc

0 rows × 63 columns

In [123]:

```
dataz['Population_z'] = stats.zscore(dataz['Population'])
Population_z = dataz.query('Population_z > 3 | Population_z < -3')
Population_z.sort_values(['Population_z'], ascending = False)
```

Out[123]:

	Index	Case_order	Customer_id	Interaction	UID	City	State	County	Zip	Lat
3024	3024	3025	W840448	7c8ccd98-1619-4492-99a7-b1dd82a713be	02cd4f72ff3415f684ab0847e47feffd	Katy	TX	Harris	77449	29.83556
9662	9662	9663	Y770582	7096d230-358f-4244-b05c-70aa3143572f	be3f3df437c3b7b114dc7d24b1a48bfc	Katy	TX	Harris	77449	29.83556
5965	5965	5966	Q787284	121150ca-a1fc-4ba7-aa48-6b7893d0eb0e	8356f8d77795cf648b1f66b4af5f1577	Houston	TX	Harris	77084	29.82641
767	767	768	E632881	e7758807-cc96-4396-a8c5-ff54d26882cc	3008f82476a1bca85459d1b3270a3f8f	Pacoima	CA	Los Angeles	91331	34.25563
7686	7686	7687	N145589	9ec70eec-90f7-4ba5-a266-df39435d1cd2	6b797b8b5e27596ef3475e8f57156ead	Pacoima	CA	Los Angeles	91331	34.25563
...	...	...	...	...	...	...	...	...	...	...
3185	3185	3186	T770780	21bd9512-ce0a-4dad-aae7-ada9c372032c	16c3b9e31c642011e9b28d4e8b091722	Kalispell	MT	Flathead	59901	48.22816
3819	3819	3820	R649606	ef1c915b-6122-43e1-a288-81fdb6adc8cb	b298e046e927404b221d90dce841db2a	Kalispell	MT	Flathead	59901	48.22816
6796	6796	6797	N911416	f2b9a623-b285-48c4-b0ea-af4d2224a904	e96e51aef8bb754c395bea931debcaeb	Mason	OH	Warren	45040	39.35199
964	964	965	U840422	3a1e8fac-deea-4914-87b4-e24a4c37233f	29198a8aad2c9d4f6b66429f61f1cf40	West Chester	PA	Chester	19382	39.92809
288	288	289	Q451442	77d2a41e-8515-4c2c-8fcb-0bc170b2cbe2	115c7675bbc8d7adb2ffe728a8a06403	Fort Washington	MD	Prince George's	20744	38.75403

218 rows × 64 columns

In [124]:

```
dataz['Zip_z'] = stats.zscore(dataz['Zip'])
Zip_z = dataz.query('Zip_z > 3 | Zip_z < -3')
Zip_z.sort_values(['Zip_z'], ascending = False)
```

Out[124]:

Index	Case_order	Customer_id	Interaction	UID	City	State	County	Zip	Lat	...	Income_z	VitD_levels_z	Doc_visits_z	Full_meal

0 rows × 65 columns

In [125]:

```
dataz['Lat_z'] = stats.zscore(dataz['Lat'])
Lat_z = dataz.query('Lat_z > 3 | Lat_z < -3')
Lat_z.sort_values(['Lat_z'], ascending = False)
```

Out[125]:

	Index	Case_order	Customer_id	Interaction	UID	City	State	County	Zip	Lat
960	960	961	L207471	3f59f2e7-e47d-41f5-9c69-a28435694872	8bd4402de2b9aaa9d398ddc2834f694a	Atqasuk	AK	North Slope	99791	70.56099
2282	2282	2283	Z462873	fef4cded-5810-4c43-b849-49ede612900c	292e98f84603bfcbbb8ab779578df8c3	Anchorage	AK	North Slope	99510	70.13850
4772	4772	4773	S598156	c8f0beab-fbe6-4c6e-96b1-04f973b16a8d	17301ae1a06453897f5863e10637ebd3	Venetie	AK	Yukon-Koyukuk	99781	67.47706
3836	3836	3837	M299873	b88c011f-aa2f-41dc-8633-5072c27a181b	d1b3b5734eca4799a52f296afdc93f81	Ambler	AK	Northwest Arctic	99786	67.17316
9141	9141	9142	P944084	ec4415b4-b579-490a-a7af-50e195f79efe	9de234b2402c0d5365f66861c99bc292	Bettles Field	AK	Yukon-Koyukuk	99726	67.11836
...	...	...	...	...	...	...	...	...	...	...
2013	2013	2014	D675480	30eae952-c151-4c25-9858-10ece8691ca2	62f14eded5c5606c559d00af81a5b057	Guayanilla	PR	Guayanilla	656	18.05280
2249	2249	2250	E748476	26b84dcf-ae87-4ea1-8cb7-4a9566877a26	76e8858db9fe7ad330140776f0b4e524	Ponce	PR	Ponce	730	18.03091
944	944	945	I293001	c5314d07-5984-4572-b727-cb484d00b67e	cdf21e87d6f3fe781ee55d08278d5132	Salinas	PR	Salinas	751	18.01023
5813	5813	5814	Q527299	3cca64fe-7391-48e4-b7a3-8e0a72d14561	5fa7855743b0bcec3db78d7a13f2e6b7	Boqueron	PR	Cabo Rojo	622	17.99174
4873	4873	4874	B702637	5066e481-8c4d-4e4d-988f-80135e832d0f	f1b458365728af1ea3392e965436559c	Aguirre	PR	Salinas	704	17.96719

144 rows × 66 columns

In [126]:

```
dataz['Lng_z'] = stats.zscore(dataz['Lng'])
Lng_z = dataz.query('Lng_z > 3 | Lng_z < -3')
Lng_z.sort_values(['Lng_z'], ascending = False)
```

Out[126]:

	Index	Case_order	Customer_id	Interaction	UID	City	State	County	Zip	Lat	.
378	378	379	U534288	6ceb0811-1275-44aa-8299-a9dd9d5ceab1	84d5d4366a34af0dfab077b864ccf94d	Yakutat	AK	Yakutat	99689	59.52058	.
5611	5611	5612	I630264	a25209fd-76aa-44b3-86b8-90c267e4f164	c1fce98dfc966e6d942561864ff64926	Northway	AK	Southeast Fairbanks	99764	63.38147	.
627	627	628	C106587	0bfad232-90c5-4073-8ec7-f5ea37f8dc3c	2f73f44bbab256a40c98564ae3127121	Central	AK	Yukon-Koyukuk	99730	65.61511	.
4772	4772	4773	S598156	c8f0beab-fbe6-4c6e-96b1-04f973b16a8d	17301ae1a06453897f5863e10637ebd3	Venetie	AK	Yukon-Koyukuk	99781	67.47706	.
6760	6760	6761	I277334	e09b15f3-c030-4fc2-a836-683d7903c01a	f38746b1cc1d220ec70e086bcde4fb6f	Cordova	AK	Valdez-Cordova	99574	60.63146	.
...	...	...	...	...	...	...	...	...	...	...	.
8841	8841	8842	R937496	91a6430c-84d2-41c3-bca0-afe9fa0cd27a	a528a4e8683ad5fa167ad4410a6b8a78	Brevig Mission	AK	Nome	99785	65.34195	.
1150	1150	1151	M44338	3f241261-2e26-4597-ac3c-230396f60da0	26ae468de550e93a8fb0dd8c5992605d	Brevig Mission	AK	Nome	99785	65.34195	.
65	65	66	Q660046	3ade4df3-2168-40df-9929-66b232d3a8a3	e81f2ce7a34173a2e91ea2914648290c	Savoonga	AK	Nome	99769	63.67959	.
965	965	966	W154018	ba7dc969-1349-415d-9fe7-8878e9a80434	3093ad47d782be083a1ebcae81481d1d	Gambell	AK	Nome	99742	63.75233	.
7336	7336	7337	N152385	06d49f7f-b2d7-49c6-ad52-dd900f46d977	35bf54f7d86d8641807014088208 75df	Atka	AK	Aleutians West	99547	52.22953	.

98 rows × 67 columns

In [127]:

```
dataz['Options_z'] = stats.zscore(dataz['Options'])
Options_z = dataz.query('Options_z > 3 | Options_z < -3')
Options_z.sort_values(['Options_z'], ascending = False)
```

Out[127]:

	Index	Case_order	Customer_id	Interaction	UID	City	State	County	Zip	Lat
371	371	372	V913617	6a9f9ede-dce6-4941-aec4-f0d9a960cf1c	7575470a2ba3f0559cd44366c66b1854	Rocky Ford	CO	Otero	81067	37.93805
2444	2444	2445	G520259	5af62758-6ef8-4fc9-87e0-0eb8e6c3e1d1	806f7f7d0ea6cb96b677f21155914b05	Tuttle	ND	Kidder	58488	47.17813
2751	2751	2752	C510896	6d3ca2ab-ff80-4312-a022-f5c1cdf97e1c	e0313ac2c67615ed26ac09ff85844277	Duncans Mills	CA	Sonoma	95430	38.46139
2901	2901	2902	M319118	a6b42670-d106-4294-bcf4-ef73404bb837	fd2f7b6a79b6c107c6a58d80ab2f93e2	Knox	PA	Clarion	16232	41.22118
3784	3784	3785	W908780	0af8874a-9626-4d8c-8622-d8472d5bbd05	c42b740687893082d4d1138a8301c99c	Newry	SC	Oconee	29665	34.72472
4322	4322	4323	C969452	2b3d7773-2381-413a-a900-043f47866d5c	ba9c8eabdd06457a3663e0c0cb73e52d	Columbus	GA	Muscogee	31903	32.41475
4754	4754	4755	A11402	363a9ecc-abe0-4a65-873c-78c1951a2494	e6eeaf589832325a9030ce3f1145158c	Miami	FL	Miami-Dade	33178	25.85803
4881	4881	4882	G449875	09208922-733e-4204-8933-6aaa8be4e705	2afac6a1d982922dd1f717d4a6634595	Lebanon	VA	Russell	24266	36.86436
5209	5209	5210	G807667	a9e2a880-2624-4bae-b79f-810eb8b05317	3b3cab0e4ba8ebe3ed6f7cf75973eb10	Avon	CT	Hartford	6001	41.79071
5992	5992	5993	E395420	a418c42c-e7e0-4405-ae49-1fa7871fc14a	d5b67f3c0d5527c9dfca6a21848be574	Warrendale	PA	Allegheny	15086	40.66541
7227	7227	7228	A751122	9b3b0e27-28a1-44e8-a8a9-16027b5f6af3	a222c4a72bf61f4bc01c528707005d24	Pasadena	TX	Harris	77503	29.70217
8100	8100	8101	Z697522	d2d2fd80-a1b0-4991-a4c0-d2df1e35d9fc	0973f36b623000e861747c5d3b18e97f	Cardwell	MO	Dunklin	63829	36.03861
8151	8151	8152	Q252120	fe3bb0ad-7432-4a8a-8102-81b099200ae5	7607342de1d353f3f3e7e9360b1a5874	Winter Garden	FL	Orange	34787	28.48236

13 rows × 68 columns

In [128]:

```
dataz['Timely_admission_z'] = stats.zscore(dataz['Timely_admission'])
Timely_admission_z = dataz.query('Timely_admission_z > 3 | Options_z < -3')
Timely_admission_z.sort_values(['Timely_admission_z'], ascending = False)
```

Out[128]:

	Index	Case_order	Customer_id	Interaction	UID	City	State	County	Zip	La
6790	6790	6791	C605737	3d3ed28d-f5df-494b-bd11-cd27c77093d3	50c72f0c71254a277cedc7d19336ee69	Pe Ell	WA	Lewis	98572	46.5512
116	116	117	Q253368	de7c4cbc-75a8-45ca-871b-fbd377786202	961865602af02e4cfeb19e4e67ba1bf7	Kittanning	PA	Armstrong	16201	40.8091
420	420	421	I30274	b5f9cc4d-c321-4d66-a04c-4eab054a39b6	62ec6ebc42411a51bcc19efcfbcf67ea	Faucett	MO	Buchanan	64448	39.5998
2356	2356	2357	L130335	6aa7c824-8804-4c2d-ba17-fe9c15686edf	9163bfdc3c246f323a899122f82f2359	Trinidad	CO	Las Animas	81082	37.1786
3772	3772	3773	Z199638	c62b19e0-1701-4c6a-81d8-3720875d458c	5309a64c2a22a27d6951843b7566a7cf	Mc Intosh	FL	Marion	32664	29.4455
5016	5016	5017	R426838	e3615fee-cc1e-4cea-abd7-5d6182fa3813	9c036eb794dc11c924a12760a3f302f7	Indianapolis	IN	Marion	46254	39.8489
5298	5298	5299	H509222	86e7bd57-33fc-499a-9b4c-7e5edbcdd169	b9a6ac0eda10b24ccdd0d59f13a0e8e0	Skyforest	CA	San Bernardino	92385	34.2147
5375	5375	5376	U499841	9a159a22-d40d-4b9b-9c47-c646dd9ecb89	a3ee73d52d794f63a01dceca701a3c98	Lima	OH	Allen	45806	40.6752
5949	5949	5950	Y669279	ce23eb44-1118-4449-b02c-b2db863e068a	7aa2d9e58477acae0acf56b48d3cb75c	Chugwater	WY	Platte	82210	41.7466
6488	6488	6489	J302887	c92e8738-f1a4-4f2c-81ba-72d2c8ec6dfb	497602cac02a66b78bc74089098cd212	Alma	KS	Wabaunsee	66401	38.9701
7431	7431	7432	R89456	5861dc08-c0ef-4c11-a0b9-8bd9fb8f5d93	dfd40be8f524c0a7212b149a952d414b	Waukegan	IL	Lake	60087	42.4034

11 rows × 69 columns

In [129]:

```
dataz['Timely_treatment_z'] = stats.zscore(dataz['Timely_treatment'])
Timely_treatment_z = dataz.query('Timely_treatment_z > 3 | Options_z < -3')
Timely_treatment_z.sort_values(['Timely_treatment_z'], ascending = False)
```

Out[129]:

	Index	Case_order	Customer_id	Interaction	UID	City	State	County	Zip	La
501	501	502	I780387	635e0f1f-1535-4b1d-9898-8339acdea07a	0b522fa019f9c845ab508d1107670413	Dublin	OH	Franklin	43016	40.0985
1764	1764	1765	N221105	d08c7a9d-11d7-4923-9d0a-c29c5fa47050	326a6a697b2873e5c84c3e8ff988a779	Greeleyville	SC	Williamsburg	29056	33.6051
5016	5016	5017	R426838	e3615fee-cc1e-4cea-abd7-5d6182fa3813	9c036eb794dc11c924a12760a3f302f7	Indianapolis	IN	Marion	46254	39.8489
5247	5247	5248	K348432	fa2b59a9-f62e-4b99-a436-f48056aaba05	bf488781a1c46f8634685e3754017d23	Dearborn Heights	MI	Wayne	48125	42.2779
5298	5298	5299	H509222	86e7bd57-33fc-499a-9b4c-7e5edbcdd169	b9a6ac0eda10b24ccdd0d59f13a0e8e0	Skyforest	CA	San Bernardino	92385	34.2147
6000	6000	6001	W425417	0ea98b00-1c7d-4f83-a0be-d6803f1d70b5	5712e1276c2d0df9c87c814557130ee7	Fort Irwin	CA	San Bernardino	92310	35.2614
7431	7431	7432	R89456	5861dc08-c0ef-4c11-a0b9-8bd9fb8f5d93	dfd40be8f524c0a7212b149a952d414b	Waukegan	IL	Lake	60087	42.4034
8326	8326	8327	P966922	ba29b074-2909-4ac0-ae8c-3d98132c1bb5	a0aefe75fb9316a55e02d0f11bed7c73	Hillsboro	MD	Caroline	21641	38.9177
8376	8376	8377	O962318	f3427c5f-7c7d-4ebc-926b-a66c8761c047	8e5566f675e2add9866ca24d88bdb879	Welda	KS	Anderson	66091	38.1739
9113	9113	9114	C804661	dc1b957c-348b-41f7-88d2-d6366e8bf0b6	c3d926798a7c16afdfc9abc2ebe345c1	Bayview	ID	Kootenai	83803	48.0363
9352	9352	9353	B573266	e031c243-b356-41ae-92ee-46a1f3a8d793	f70d725f7037eafce1b89ab1000710d8	Encinitas	CA	San Diego	92024	33.0561
9763	9763	9764	T741340	39753426-4e17-4d66-a135-87a4367840ad	8b253260c77c08fcaa54aea8e2f91d70	Nicholson	PA	Lackawanna	18446	41.6450

12 rows × 70 columns

In [130]:

```
dataz['Timely_visits_z'] = stats.zscore(dataz['Timely_visits'])
Timely_visits_z = dataz.query('Timely_visits_z > 3 | Timely_visits_z < -3')
Timely_visits_z.sort_values(['Timely_visits_z'], ascending = False)
```

Out[130]:

	Index	Case_order	Customer_id	Interaction	UID	City	State	County	Zip	Lat
8822	8822	8823	H579237	33326e08-9f62-4159-8d6a-d66545d8f4c5	f526923b83632e506fb60fb12e0e2e5f	Hoisington	KS	Barton	67544	38.58229
1028	1028	1029	E875190	999b36db-926b-4b88-894d-ecaa90dee332	b7e732e4a621c935ed640d6b46cc5a0a	Battle Creek	MI	Calhoun	49015	42.27127
1642	1642	1643	D685434	496fb29f-7556-430d-a720-7f4d24c4b75f	7cfcb06672f5b69c126fa537e2e80646	Eola	IL	DuPage	60519	41.77789
2939	2939	2940	F633638	f991a423-956e-47b3-8688-70ad78315bb6	3e4b4f0b30b7cd50d3cb71fe6350d348	Wyncote	PA	Montgomery	19095	40.08597
3805	3805	3806	N763358	7ab39276-865f-4445-8fe3-366fb7043dc4	f508f7c105d64011562326584f6e4e89	Cass Lake	MN	Cass	56633	47.31969
4050	4050	4051	C64476	2a818f63-e4bc-407b-8ae7-dd36f3578230	b01fac561a372c34255d2d607569fd14	Horatio	SC	Sumter	29062	33.99475
4407	4407	4408	H470636	c0994b61-454e-4c42-8617-26304aa9d717	c57d48f888a7783080049f4246196487	Mine Hill	NJ	Morris	7803	40.87768
6686	6686	6687	R295268	6ea0af09-7536-41ff-9800-c09fbc6a668b	33129b04dbe49d9623900e922ffe1e55	Sylmar	CA	Los Angeles	91342	34.31515
8964	8964	8965	R85226	e3703132-3e0a-46a1-9250-1824d2c7ad55	eca777ec1754a973bc33e553c9b0d055	Norwich	CT	New London	6360	41.54884
9113	9113	9114	C804661	dc1b957c-348b-41f7-88d2-d6366e8bf0b6	c3d926798a7c16afdfc9abc2ebe345c1	Bayview	ID	Kootenai	83803	48.03638
9528	9528	9529	O612221	a1e51d59-d286-4d7c-a310-f61354ea0ae3	09e5eac102d16fe00214670cff2f281e	Brookfield	MO	Linn	64628	39.79720
9827	9827	9828	V442531	7e0ff2ee-5b10-426b-8d7c-0f9cecf0dbaa	cfaee561a749c8b9348a674e96fff4bc	Hurley	NY	Ulster	12443	41.93393

12 rows × 71 columns

In [131]:

```
dataz['Reliability_z'] = stats.zscore(dataz['Reliability'])
Reliability_z = dataz.query('Reliability_z > 3 | Reliability_z < -3')
Reliability_z.sort_values(['Reliability_z'], ascending = False)
```

Out[131]:

	Index	Case_order	Customer_id	Interaction	UID	City	State	County	Zip	Lat
448	448	449	X12279	791a7ee3-0c9b-4b43-9c24-2b33d43bbe6f	34127cd5ef45302ae320eb5c4cd1818f	Eastlake Weir	FL	Marion	32133	29.02018
2101	2101	2102	I382969	df50efa2-6a57-4501-8230-d72650de2c52	4506338ab2653e7ef05edb1df50d1374	Columbia	MD	Howard	21046	39.17356
3178	3178	3179	B356505	8cda92fd-73c7-4074-8f1e-00cacd66538e	94cc5113be1dae40d7a06b76499d4cb6	Paul	ID	Minidoka	83347	42.73392
3225	3225	3226	X349857	b15f90d5-2def-4729-9a96-7e73e5a9b184	60bdc5a688500d47c205bdf4ae6d87f7	Deer Park	TX	Harris	77536	29.69839
4211	4211	4212	M801409	47e2d486-cbd5-4219-8270-b8ac05a831d7	8a62458f602cbebeb69774ec0172f9ed	Dixie	WV	Fayette	25059	38.23311
4776	4776	4777	E632070	a82d11e2-68b8-475d-bbab-322fda5b882b	7d77e330fbafe63f9088310532c27d5e	Columbus	TX	Colorado	78934	29.69375
5300	5300	5301	F18171	c4d5dff8-cabe-4839-aff4-9e6fdbfeafd5	4d492e879a993a3d12711b2370c1d0be	Torrance	CA	Los Angeles	90504	33.86682
6461	6461	6462	M335375	56dcfbe9-d2b6-432d-a3f2-ee53be0ede3e	c1cc82e04b82f27848576181f7337602	Olney	MT	Flathead	59927	48.57210
6983	6983	6984	U605143	1d5bb548-aefc-4895-b20a-c8a6baf60b48	eeed90779ca7d42bc79169c9014bfb45	Pembroke Township	IL	Kankakee	60958	41.06492
7585	7585	7586	S186662	da77edc9-494f-42c8-a439-c60d962cfe86	7cc42283a16ad06f41222c90101fbfe4	Pleasureville	KY	Henry	40057	38.38941
9708	9708	9709	I863574	c61ad302-84d7-498b-b744-f02cfae51a0e	1545a4a865cfa40dc797acd623f5bd37	Gordon	GA	Twiggs	31031	32.87127
9798	9798	9799	X960973	6494dcf6-0e77-4093-9687-9598ae0f7e50	b84cc9a7f1028b58b10e02b6461c7529	Axis	AL	Mobile	36505	30.94264

12 rows × 72 columns

In [132]:

```
dataz['Hours_z'] = stats.zscore(dataz['Hours'])
Hours_z = dataz.query('Hours_z > 3 | Hours_z < -3')
Hours_z.sort_values(['Hours_z'], ascending = False)
```

Out[132]:

	Index	Case_order	Customer_id	Interaction	UID	City	State	County	Zip	Lat
565	565	566	D442431	f7c46c99-70fe-4d6c-bdf5-67349d4e7ef7	20cc938a8b12edfd67faddf51db079e4	Burnside	IA	Webster	50521	42.34623
1755	1755	1756	P17573	f3addeec-cdbc-455b-972d-a53ca3e1ec88	4504b498855a2c2a64f1f455244336aa	Winnsboro	SC	Fairfield	29180	34.36739
1952	1952	1953	Q450603	a8a6aa7d-3bb0-46e7-9b1a-c52180587d63	e443edee809fd9937a2167b060af46b3	Nanjemoy	MD	Charles	20662	38.43516
2574	2574	2575	F525478	6a38e9cb-b2fd-4044-8f71-2793507c28e5	66e1f4663fe790b3ec24c900ebf0edb3	Beaver Bay	MN	Lake	55601	47.23577
2871	2871	2872	L172909	cb794d21-e46a-4f93-8071-431d8f8857f4	864b1053b47da42f8439efb5ec2e6b0b	Fayetteville	GA	Fayette	30214	33.49170
4141	4141	4142	D232618	80e82d9f-5ceb-460f-8cb5-610ac12927cc	32a5fbdf11647d8cf4ca3903d5371d51	Conway	NC	Northampton	27820	36.41563
4808	4808	4809	I840751	463c85ed-291f-4e56-a036-5dc319bfdb08	35814159827104bc8d42fe74a3c74837	Little Neck	NY	Queens	11363	40.77268
6790	6790	6791	C605737	3d3ed28d-f5df-494b-bd11-cd27c77093d3	50c72f0c71254a277cedc7d19336ee69	Pe Ell	WA	Lewis	98572	46.55120
7359	7359	7360	S363644	6fd69439-6c22-4e89-a13e-68349730c50d	08ab920b9c49d1263bfe4203b0251cac	Vero Beach	FL	Indian River	32968	27.58700
7553	7553	7554	R463541	98f8f4af-679c-4f63-a578-91f1818f407d	d9cfcc2c206eaa0f5343e3cd1d176f13	Rutherford	TN	Gibson	38369	36.13228

10 rows × 73 columns

In [133]:

```
dataz['Courteous_z'] = stats.zscore(dataz['Courteous'])
Courteous_z = dataz.query('Courteous_z > 3 | Courteous_z < -3')
Courteous_z.sort_values(['Courteous_z'], ascending = False)
```

Out[133]:

	Index	Case_order	Customer_id	Interaction	UID	City	State	County	Zip	Lat
599	599	600	T536145	e4fe184f-c28a-416a-816c-b9f1898f7d73	d8c1c6ac065390f252cdb698708233df	Hawthorn	PA	Clarion	16230	41.02099
2010	2010	2011	T101729	7ba6e56a-01bb-4ebb-a630-a394d2c730d6	f43629df63b8350dc12a8586bf35c690	Caldwell	ID	Canyon	83607	43.70795
3790	3790	3791	E382593	82eb36cd-68b4-41a5-92c2-a92824d2ce8d	384cb41d65137a4e243abaa659eeb543	Hampton	VA	Hampton	23664	37.07528
4850	4850	4851	E444065	6a02074b-e1bb-4685-a767-3df69665a270	d64310eb316104566c7cd992783f3431	Tatitlek	AK	Valdez-Cordova	99677	60.89214
6646	6646	6647	Z03012	44b36d8d-7ac5-4368-919f-b33144ad9542	c55ce718a887e70b8ccbf845c58584c8	Roseland	NJ	Essex	7068	40.82071
7527	7527	7528	Q775427	44a210d6-c9f0-490d-b544-55fce9e8f50a	2729d115709a2096811a62e42c6e04f1	Horton	AL	Marshall	35980	34.17625
7843	7843	7844	E472114	22957ec2-6c15-4bbf-bc0e-2f4a2f7a6f05	9de913184e45ee14815a8dbd467f102b	East Bridgewater	MA	Plymouth	2333	42.03515
8142	8142	8143	C831750	0c68fb1f-8acc-4d23-a4c7-9db30c5976ce	be947f4225d7856c32d58286bb1b463a	Charleston	SC	Charleston	29414	32.83802
8165	8165	8166	I670859	e9dec8d8-c2e1-4ed4-91ae-390090dc45b2	440b7ad9e4e5bf020215b7ed6d7f670d	Springfield	IL	Sangamon	62701	39.80082
8209	8209	8210	J805835	5ba4461e-d984-4d40-979a-e7433bfce4ef	7b193c59128eaa3b856febd43dca4222	Hamburg	NJ	Sussex	7419	41.15321
8720	8720	8721	U975030	8ef249c1-5b93-42d6-b105-2b7c93a49ea5	53ac1f6c55358c8352a3b8d05680c525	Algoma	WI	Kewaunee	54201	44.62083

11 rows × 74 columns

In [134]:

```
dataz['Active_listen_z'] = stats.zscore(dataz['Active_listen'])
Active_listen_z = dataz.query('Active_listen_z > 3 | Active_listen_z < -3')
Active_listen_z.sort_values(['Active_listen_z'], ascending = False)
```

Out[134]:

	Index	Case_order	Customer_id	Interaction	UID	City	State	County	Zip	
248	248	249	F210779	e9693fd1-0d38-494a-8961-038d929066ee	3820c94c0e8e198124716e61e4a0f674	Oklahoma City	OK	Oklahoma	73102	35.47
898	898	899	S435495	4d71a2be-b91c-40a7-9db0-a0b973154826	6a90d9c6b5cf447735de5bae988dcef6	Harviell	MO	Butler	63945	36.63
1096	1096	1097	O879050	3ea27c2a-3a58-43a2-bde2-52a5a1a2b014	8e4e762fa2f47dc5cdfc42e49fb2688e	Gadsden	AL	Etowah	35903	34.02
1402	1402	1403	Z958874	6a3115dc-e21f-4fe4-ad62-1dbf4d7d8e9f	40ca6a31050d74f2eaef8bc217311d4c	Fleming	PA	Centre	16835	40.90
2054	2054	2055	M400514	5f863a0c-008e-4933-be8a-95ae7ff7c1fa	542cfb80a4610dbb641a4d8e7995f924	Hague	VA	Westmoreland	22469	38.07
2736	2736	2737	M174545	44d3b166-135f-4f8e-a546-d489a7cc2b29	eed6c940042850d71911cb55a61aa0c0	Sylvania	OH	Lucas	43560	41.70
3300	3300	3301	V574050	1ca7a786-bd34-44ea-8051-c4a746ec9e62	9cea7c94a4ee90ceb62b74ead6315f6a	South Bloomingville	OH	Hocking	43152	39.39
3395	3395	3396	U246066	d579c126-57e0-40ef-9343-52bf53f71f93	b0e56f03b1aae1846c3966bde3b4f0a5	Coloma	WI	Waushara	54930	44.02
5949	5949	5950	Y669279	ce23eb44-1118-4449-b02c-b2db863e068a	7aa2d9e58477acae0acf56b48d3cb75c	Chugwater	WY	Platte	82210	41.74
6508	6508	6509	T191666	b9de5930-e19e-46a9-b0ce-7311bb4e9ca7	5aef9f1f998d472b7db14e77564228f2	Seymour	MO	Webster	65746	37.14
8326	8326	8327	P966922	ba29b074-2909-4ac0-ae8c-3d98132c1bb5	a0aefe75fb9316a55e02d0f11bed7c73	Hillsboro	MD	Caroline	21641	38.91
9799	9799	9800	Z246842	4afec6c9-9a63-4710-b72f-898f65fdd4e9	158ae21bf8d8ba5501015416cdc6ee9d	Normal	IL	McLean	61761	40.52

12 rows × 75 columns

In [135]:

```
dataz.to_csv('C:/Users/ericy/Desktop/data_z.csv')
```

In [136]:

```
dataz.shape
```

Out[136]:

```
(10000, 75)
```

In [137]:

```
dataz.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 75 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Index                 10000 non-null  int64  
 1   Case_order            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  int64  
 13  Timezone              10000 non-null  object 
 14  Job                   10000 non-null  object 
 15  Children              10000 non-null  int64  
 16  Age                   10000 non-null  int64  
 17  Education             10000 non-null  int64  
 18  Employment            10000 non-null  int64  
 19  Income                10000 non-null  int64  
 20  Marital               10000 non-null  int64  
 21  Gender                10000 non-null  int64  
 22  Readmis               10000 non-null  int64  
 23  VitD_levels           10000 non-null  int64  
 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  int64  
 28  Initial_admin         10000 non-null  int64  
 29  High_blood            10000 non-null  int64  
 30  Stroke                10000 non-null  int64  
 31  Complication_risk     10000 non-null  int64  
 32  Overweight            10000 non-null  int64  
 33  Arthritis             10000 non-null  int64  
 34  Diabetes              10000 non-null  int64  
 35  Hyperlipidemia        10000 non-null  int64  
 36  Back_pain             10000 non-null  int64  
 37  Anxiety               10000 non-null  int64  
 38  Allergic_rhinitis     10000 non-null  int64  
 39  Reflux_esophagitis    10000 non-null  int64  
 40  Asthma                10000 non-null  int64  
 41  Services              10000 non-null  int64  
 42  Initial_days          10000 non-null  int64  
 43  Total_charge          10000 non-null  int64  
 44  Additional_charges    10000 non-null  int64  
 45  Timely_admission      10000 non-null  int64  
 46  Timely_treatment      10000 non-null  int64  
 47  Timely_visits         10000 non-null  int64  
 48  Reliability           10000 non-null  int64  
 49  Options               10000 non-null  int64  
 50  Hours                 10000 non-null  int64  
 51  Courteous             10000 non-null  int64  
 52  Active_listen         10000 non-null  int64  
 53  Age_z                 10000 non-null  float64
 54  Children_z            10000 non-null  float64
 55  Income_z              10000 non-null  float64
 56  VitD_levels_z         10000 non-null  float64
 57  Doc_visits_z          10000 non-null  float64
 58  Full_mealz            10000 non-null  float64
 59  VitD_suppz            10000 non-null  float64
 60  Initial_days_z        10000 non-null  float64
 61  Total_charge_z        10000 non-null  float64
 62  Additional_charges_z  10000 non-null  float64
 63  Population_z          10000 non-null  float64
 64  Zip_z                 10000 non-null  float64
 65  Lat_z                 10000 non-null  float64
 66  Lng_z                 10000 non-null  float64
 67  Options_z             10000 non-null  float64
 68  Timely_admission_z    10000 non-null  float64
 69  Timely_treatment_z    10000 non-null  float64
 70  Timely_visits_z       10000 non-null  float64
 71  Reliability_z         10000 non-null  float64
 72  Hours_z               10000 non-null  float64
 73  Courteous_z           10000 non-null  float64
 74  Active_listen_z       10000 non-null  float64
dtypes: float64(24), int64(43), object(8)
memory usage: 5.7+ MB
```

In [138]:

```
#PCA
med = pd.read_csv('C:/Users/ericy/Desktop/pca_1.csv')
```

In [139]:

```
med.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 13 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Doc_visits          10000 non-null  int64
 1   VitD_supp           10000 non-null  int64
 2   Initial_days        10000 non-null  int64
 3   Total_charge        10000 non-null  int64
 4   Additional_charges  10000 non-null  int64
 5   Timely_admission    10000 non-null  int64
 6   Timely_treatment    10000 non-null  int64
 7   Timely_visits       10000 non-null  int64
 8   Reliability         10000 non-null  int64
 9   Options             10000 non-null  int64
 10  Hours               10000 non-null  int64
 11  Courteous           10000 non-null  int64
 12  Active_listen       10000 non-null  int64
dtypes: int64(13)
memory usage: 1015.8 KB
```

In [140]:

```
#Define variables for PCA
med = med[['Doc_visits','VitD_supp','Initial_days','Total_charge','Additional_charges','Timely_admission','Timely
_treatment','Timely_visits','Reliability','Options','Hours','Courteous','Active_listen']]
```

In [141]:

```
#Normalize data - scales data
med_normalized = (med-med.mean())/med.std()
```

In [142]:

```
pca = PCA(n_components=med.shape[1])
```

In [143]:

```
pca.fit(med_normalized)
```

Out[143]:

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

In [144]:

```
med_pca = pd.DataFrame(pca.transform(med_normalized),
                       columns=['PC1','PC2','PC3','PC4','PC5','PC6','PC7','PC8','PC9','PC10','PC11','PC12','PC13']
)
```

In [145]:

```
loadings = pd.DataFrame(pca.components_.T,
                       columns =['PC1','PC2','PC3','PC4','PC5','PC6','PC7','PC8','PC9','PC10','PC11','PC12','PC13
'],
                       index=med.columns)
loadings
```

Out[145]:

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11
Doc_visits	0.007069	-0.002075	-0.013897	0.551134	-0.750294	0.362101	-0.019941	-0.026724	-0.025656	0.007265	-0.010368
VitD_supp	-0.004949	0.019578	0.034108	0.545353	0.650617	0.524840	0.030928	0.030766	0.013852	-0.003816	0.010258
Initial_days	-0.016866	0.425164	0.562897	-0.043657	-0.030162	0.021512	-0.011771	-0.006577	0.000766	-0.007445	0.031354
Total_charge	-0.014241	0.440002	0.552356	-0.010146	-0.022778	-0.003120	-0.000712	-0.014128	0.000295	0.011957	-0.028224
Additional_charges	0.003986	0.034518	0.020067	0.629137	0.092341	-0.768731	0.004383	-0.041244	0.005803	0.014262	-0.015063
Timely_admission	0.454784	-0.232816	0.184023	0.002111	0.007337	-0.003075	-0.095714	-0.076403	-0.010802	0.086216	0.181731
Timely_treatment	0.428496	-0.226595	0.186167	0.004032	0.004817	-0.002444	-0.146858	-0.134481	-0.062202	0.102062	0.625524
Timely_visits	0.395301	-0.228728	0.188430	-0.004630	0.027615	0.010274	-0.204619	-0.212429	-0.238900	-0.433423	-0.620798
Reliability	0.152243	0.437651	-0.346530	-0.019568	0.047671	0.027333	-0.365196	-0.361566	-0.387968	0.483537	-0.113822
Options	-0.190134	-0.463555	0.355510	-0.004262	-0.000015	0.000493	0.124501	0.058344	-0.132365	0.694576	-0.307619
Hours	0.410398	0.134827	-0.093755	-0.005404	-0.014136	0.013554	-0.050728	0.061982	0.796740	0.266844	-0.274555
Courteous	0.356642	0.150254	-0.089585	0.013667	-0.017869	-0.029628	0.035179	0.846287	-0.335176	0.068621	-0.060967
Active_listen	0.312688	0.137892	-0.094741	-0.018968	-0.013051	0.008766	0.879324	-0.270498	-0.151259	0.040836	-0.037450

In [146]:

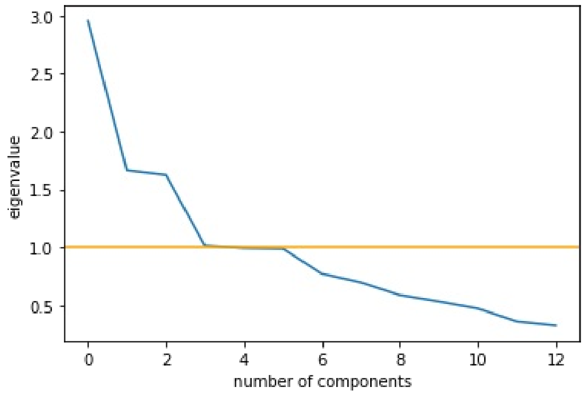
```
cov_matrix = np.dot(med_normalized.T, med_normalized) / med.shape[0]
```

In [147]:

```
eigenvalues = [np.dot(eigenvector.T, np.dot(cov_matrix, eigenvector)) for eigenvector in pca.components_]
```

In [148]:

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



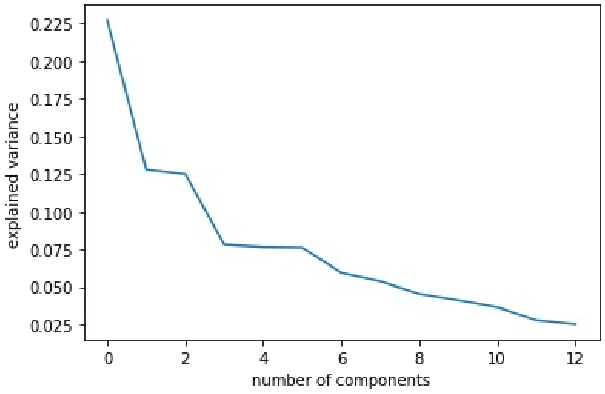
In [149]:

```
print(eigenvalues)
```

```
[2.953561847701104, 1.664527005604655, 1.6253016059708472, 1.015751765687237, 0.9944920712433167, 0.
9897973674624058, 0.7730497224229762, 0.698131569259065, 0.588691460987571, 0.5337901719644559, 0.47
36831400386705, 0.361101692090996, 0.3268205795665951]
```

In [150]:

```
plt.plot(pca.explained_variance_ratio_)
plt.xlabel('number of components')
plt.ylabel('explained variance')
plt.show()
```



In [ ]: