In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
import warnings
```

In [2]:
```
warnings.filterwarnings(action='ignore')
pd.options.mode.chained_assignment = None  # default='warn' ---- ignores false warning
```

In [3]:
```
df = pd.read_csv('medical_clean.csv')
```

In [4]:
```
df = df[['City', 'State',       'County',       'Zip',  'Lat',  'Lng',  'Population',
```

In [5]:
```
# check for duplicates and null values
print(df.loc[df.duplicated()])
print(df.isnull().sum())
```

```
Empty DataFrame
Columns: [City, State, County, Zip, Lat, Lng, Population, Area, TimeZone, Job, Children,
Age, Income, Marital, Gender, ReAdmis, VitD_levels, Doc_visits, Full_meals_eaten, vitD_s
upp, Soft_drink, Initial_admin, HighBlood, Stroke, Complication_risk, Overweight, Arthri
tis, Diabetes, Hyperlipidemia, BackPain, Anxiety, Allergic_rhinitis, Reflux_esophagitis,
Asthma, Services, Initial_days, TotalCharge, Additional_charges]
Index: []

[0 rows x 38 columns]
City                  0
State                 0
County                0
Zip                   0
Lat                   0
Lng                   0
Population            0
Area                  0
TimeZone              0
Job                   0
Children              0
Age                   0
Income                0
Marital               0
Gender                0
ReAdmis               0
VitD_levels           0
Doc_visits            0
Full_meals_eaten      0
vitD_supp             0
Soft_drink            0
Initial_admin         0
HighBlood             0
Stroke                0
```

```
Complication_risk     0
Overweight            0
Arthritis             0
Diabetes              0
Hyperlipidemia        0
BackPain              0
Anxiety               0
Allergic_rhinitis     0
Reflux_esophagitis    0
Asthma                0
Services              0
Initial_days          0
TotalCharge           0
Additional_charges    0
dtype: int64
```

In [6]:

```
# check for outliers and remove (appears VitD_levels contained the outliers)
print(df.shape)
df = df[(np.abs(stats.zscore(df.select_dtypes(include=np.number))) < 3).all(axis=1)]
print(df.shape)
print(df.head())
```

```
(10000, 38)
(9198, 38)
          City State        County    Zip       Lat       Lng  Population  \
0          Eva    AL        Morgan  35621  34.34960 -86.72508        2951
1     Marianna    FL       Jackson  32446  30.84513 -85.22907       11303
2   Sioux Falls    SD     Minnehaha  57110  43.54321 -96.63772       17125
3  New Richland    MN        Waseca  56072  43.89744 -93.51479        2162
4    West Point    VA  King William  23181  37.59894 -76.88958        5287

       Area          TimeZone                               Job  ...  \
0  Suburban   America/Chicago  Psychologist, sport and exercise  ...
1     Urban   America/Chicago      Community development worker  ...
2  Suburban   America/Chicago           Chief Executive Officer  ...
3  Suburban   America/Chicago               Early years teacher  ...
4     Rural  America/New_York       Health promotion specialist  ...

  Hyperlipidemia  BackPain  Anxiety Allergic_rhinitis Reflux_esophagitis  \
0             No       Yes      Yes               Yes                 No
1             No        No       No                No                Yes
2             No        No       No                No                 No
3             No        No       No                No                Yes
4            Yes        No       No               Yes                 No

  Asthma     Services  Initial_days  TotalCharge  Additional_charges
0    Yes   Blood Work     10.585770  3726.702860        17939.403420
1     No  Intravenous     15.129562  4193.190458        17612.998120
2     No   Blood Work      4.772177  2434.234222        17505.192460
3    Yes   Blood Work      1.714879  2127.830423        12993.437350
4     No      CT Scan      1.254807  2113.073274         3716.525786

[5 rows x 38 columns]
```

In [7]:

```
di = {'Yes': 1, 'No': 0}
di2 = {'Rural': 1, 'Suburban': 2, 'Urban': 3}
di3 = {'Divorced': 1, 'Married': 2, 'Widowed': 3, 'Never Married': 4, 'Separated': 5}
di4 = {'Male': 1, 'Female': 2, 'Nonbinary': 3}
di5 = {'Low': 1, 'Medium': 2, 'High': 3}
di6 = {'Blood Work': 1, 'Intravenous': 2, 'CT Scan': 3}
df = df.replace({'Area': di2, 'ReAdmis': di, 'Soft_drink': di, 'HighBlood': di,'Stroke'
```

```
print(df.head())
df.to_csv('initial_clean.csv')
```

```
           City State        County    Zip       Lat       Lng  Population  \
0           Eva    AL        Morgan  35621  34.34960 -86.72508        2951
1      Marianna    FL       Jackson  32446  30.84513 -85.22907       11303
2   Sioux Falls    SD     Minnehaha  57110  43.54321 -96.63772       17125
3  New Richland    MN        Waseca  56072  43.89744 -93.51479        2162
4    West Point    VA  King William  23181  37.59894 -76.88958        5287

   Area          TimeZone                               Job  ...  \
0     2   America/Chicago  Psychologist, sport and exercise  ...
1     3   America/Chicago      Community development worker  ...
2     2   America/Chicago           Chief Executive Officer  ...
3     2   America/Chicago               Early years teacher  ...
4     1  America/New_York       Health promotion specialist  ...

   Hyperlipidemia  BackPain  Anxiety Allergic_rhinitis Reflux_esophagitis  \
0               0         1        1                 1                  0
1               0         0        0                 0                  1
2               0         0        0                 0                  0
3               0         0        0                 0                  1
4               1         0        0                 1                  0

   Asthma  Services  Initial_days  TotalCharge  Additional_charges
0       1         1     10.585770  3726.702860        17939.403420
1       0         2     15.129562  4193.190458        17612.998120
2       0         1      4.772177  2434.234222        17505.192460
3       1         1      1.714879  2127.830423        12993.437350
4       0         3      1.254807  2113.073274         3716.525786

[5 rows x 38 columns]
```

In [8]:

```
df.hist(figsize = (16,16))
plt.savefig('hospital_pyplot.jpg')
plt.tight_layout()
plt.close()
print('Histogram done')
```

```
Histogram done
```

In [9]:

```
# bivariate analysis heatmap
ax = plt.subplots(figsize=(18,18))
ax = sns.heatmap(df.corr(), annot=True)
plt.savefig('heatmap_initial.jpg')
plt.close()
print('Initial heatmap done')
```

```
Initial heatmap done
```

In [10]:

```
# heatmap says we should focus on Lng, Zip, Population, Lat, Age, Additional_charges, H
dfReduced = df[['Zip',  'Lat',  'Lng',  'Population',   'Age', 'ReAdmis',       'HighBl
print(dfReduced.head())
dfReduced.to_csv('reduced_clean.csv')
```

```
     Zip       Lat       Lng  Population  Age  ReAdmis  HighBlood  \
0  35621  34.34960 -86.72508        2951   53        0          1
1  32446  30.84513 -85.22907       11303   51        0          1
2  57110  43.54321 -96.63772       17125   53        0          1
3  56072  43.89744 -93.51479        2162   78        0          0
4  23181  37.59894 -76.88958        5287   22        0          0
```

```
   Initial_days  TotalCharge  Additional_charges
0     10.585770  3726.702860        17939.403420
1     15.129562  4193.190458        17612.998120
2      4.772177  2434.234222        17505.192460
3      1.714879  2127.830423        12993.437350
4      1.254807  2113.073274         3716.525786
```

In [11]:

```
X = np.array(dfReduced[['Zip',  'Lat',  'Lng',  'Population', 'ReAdmis',        'HighBl
y = np.array(dfReduced[['Age']])
```

In [12]:

```
model = GaussianNB()
model.fit(X, y)
print(model)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_stat
np.savetxt('training.csv', y_train)
print(y_train)
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
classifier = GaussianNB()
classifier.fit(X_train, y_train)
y_pred  =  classifier.predict(X_test)
np.savetxt('prediction.csv', y_pred)
```

```
GaussianNB(priors=None, var_smoothing=1e-09)
[[20]
 [58]
 [69]
 ...
 [73]
 [44]
 [86]]
```

In [13]:

```
#confusion matrix accuracy stats
print('Accuracy: ', accuracy_score(y_test, y_pred))
print('Classification Report\n')
print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))
```

```
Accuracy:  0.025
Classification Report

              precision    recall  f1-score   support

          18       0.11      0.35      0.17        26
          19       0.17      0.29      0.21        28
          20       0.03      0.03      0.03        29
          21       0.00      0.00      0.00        27
          22       0.00      0.00      0.00        23
          23       0.08      0.13      0.10        30
          24       0.02      0.05      0.02        22
          25       0.00      0.00      0.00        27
          26       0.06      0.08      0.07        24
          27       0.00      0.00      0.00        25
          28       0.00      0.00      0.00        29
          29       0.00      0.00      0.00        14
          30       0.01      0.06      0.02        31
          31       0.04      0.03      0.04        30
          32       0.00      0.00      0.00        23
```

```
              33       0.00      0.00      0.00        22
              34       0.00      0.00      0.00        30
              35       0.00      0.00      0.00        25
              36       0.00      0.00      0.00        15
              37       0.00      0.00      0.00        28
              38       0.00      0.00      0.00        33
              39       0.00      0.00      0.00        23
              40       0.00      0.00      0.00        26
              41       0.03      0.02      0.03        42
              42       0.00      0.00      0.00        22
              43       0.00      0.00      0.00        30
              44       0.00      0.00      0.00        20
              45       0.00      0.00      0.00        30
              46       0.00      0.00      0.00        22
              47       0.00      0.00      0.00        30
              48       0.00      0.00      0.00        27
              49       0.00      0.00      0.00        16
              50       0.00      0.00      0.00        17
              51       0.00      0.00      0.00        22
              52       0.00      0.00      0.00        33
              53       0.00      0.00      0.00        22
              54       0.00      0.00      0.00        26
              55       0.00      0.00      0.00        23
              56       0.02      0.04      0.02        24
              57       0.00      0.00      0.00        28
              58       0.00      0.00      0.00        32
              59       0.00      0.00      0.00        32
              60       0.00      0.00      0.00        17
              61       0.00      0.00      0.00        23
              62       0.00      0.00      0.00        27
              63       0.00      0.00      0.00        26
              64       0.02      0.08      0.04        25
              65       0.00      0.00      0.00        31
              66       0.03      0.03      0.03        31
              67       0.00      0.00      0.00        22
              68       0.00      0.00      0.00        20
              69       0.00      0.00      0.00        23
              70       0.00      0.00      0.00        29
              71       0.00      0.00      0.00        25
              72       0.00      0.00      0.00        17
              73       0.08      0.03      0.04        32
              74       0.03      0.09      0.05        35
              75       0.00      0.00      0.00        19
              76       0.00      0.00      0.00        36
              77       0.10      0.12      0.11        17
              78       0.00      0.00      0.00        30
              79       0.00      0.00      0.00        29
              80       0.00      0.00      0.00        18
              81       0.07      0.12      0.09        25
              82       0.00      0.00      0.00        20
              83       0.00      0.00      0.00        32
              84       0.00      0.00      0.00        18
              85       0.05      0.04      0.05        25
              86       0.03      0.11      0.05        18
              87       0.05      0.03      0.04        31
              88       0.00      0.00      0.00        26
              89       0.00      0.00      0.00        25

    accuracy                           0.03      1840
   macro avg       0.01      0.02      0.02      1840
weighted avg       0.01      0.03      0.02      1840

[[9 4 1 ... 0 0 0]
 [7 8 2 ... 0 0 0]
 [8 3 1 ... 0 0 0]
```

```
...
[0 0 0 ... 1 0 2]
[0 0 0 ... 2 0 0]
[0 0 0 ... 1 0 0]]
```

In [ ]: