In [9]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LassoCV
from sklearn.model_selection import RepeatedKFold
from sklearn.metrics import mean_squared_error
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')

df = df[['City', 'State',       'County',       'Zip',  'Lat',  'Lng',  'Population',
```

In [4]:

```
# 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 [5]:

```
# 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 [6]:

```
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 [7]:

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

```
Histogram done
```

In [8]:

```python
# 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]:

```python
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', 'HighBlood',
y = np.array(dfReduced[['Age']])
```

In [12]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, random_state=
np.savetxt('training.csv', y_train)
np.savetxt('testing.csv', y_test)
```

In [13]:

```
cv = RepeatedKFold(n_splits=10, n_repeats=3, random_state=1)
model = LassoCV(alphas=np.arange(0, 1, 0.01), cv=cv, n_jobs=-1)
c = model.fit(X_train, y_train)
print(model.alpha_)
d = model.score(X_test, y_test)
print(d)
e = model.predict(X_test)
np.savetxt('predictAges.csv', e)

# mean squared error
print(mean_squared_error(y_test, e))
```

```
0.02
0.889454080053387
46.98629349317084
```

In [ ]: