In [646…

```python
import numpy as np
import pandas as pd
import seaborn as sns
import math
import random
import matplotlib.pyplot as plt
from sklearn import preprocessing
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import classification_report
from itertools import product
%matplotlib inline
import pylab
from pylab import rcParams
import statsmodels.api as sm

import statistics
from scipy import stats
import sklearn
import warnings
warnings.filterwarnings('ignore')
import matplotlib as mpl
#import data to dataframe
medical_df = pd.read_csv('C:/Users/MichaelRupert/Downloads/e9d8sm5uf8df75k650df/medical_raw_data.csv');
list(medical_df.columns)
print(medical_df.columns)
```

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

In [647…

```python
#Drop unndeed columns
medical_df = medical_df.drop(columns=['CaseOrder', 'Customer_id', 'Interaction',
                                      'UID', 'County','Timezone', 'Job', 'Education', 'Employment',
                                      'Children', 'Zip','Doc_visits',
                                      'Initial_days'], axis =1)
medical_df.columns
```

Out[647…

```
Index(['Unnamed: 0', 'City', 'State', 'Lat', 'Lng', 'Population', 'Area',
       'Age', 'Income', 'Marital', 'Gender', 'ReAdmis', 'VitD_levels',
       'Full_meals_eaten', 'VitD_supp', 'Soft_drink', 'Initial_admin',
       'HighBlood', 'Stroke', 'Complication_risk', 'Overweight', 'Arthritis',
       'Diabetes', 'Hyperlipidemia', 'BackPain', 'Anxiety',
       'Allergic_rhinitis', 'Reflux_esophagitis', 'Asthma', 'Services',
       'TotalCharge', 'Additional_charges', 'Item1', 'Item2', 'Item3', 'Item4',
       'Item5', 'Item6', 'Item7', 'Item8'],
      dtype='object')
```

In [648…

```python
#Describe the data in the dataset
medical_df.describe().transpose()
```

Out[648…

	count	mean	std	min	25%	50%	75%	max
Unnamed: 0	10000.0	5000.500000	2886.895680	1.000000	2500.750000	5000.500000	7500.250000	10000.000000
Lat	10000.0	38.751099	5.403085	17.967190	35.255120	39.419355	42.044175	70.560990
Lng	10000.0	-91.243080	15.205998	-174.209690	-97.352982	-88.397230	-80.438050	-65.290170
Population	10000.0	9965.253800	14824.758614	0.000000	694.750000	2769.000000	13945.000000	122814.000000
Age	7586.0	53.295676	20.659182	18.000000	35.000000	53.000000	71.000000	89.000000
Income	7536.0	40484.438268	28664.861050	154.080000	19450.792500	33942.280000	54075.235000	207249.130000
VitD_levels	10000.0	19.412675	6.723277	9.519012	16.513171	18.080560	19.789740	53.019124
Full_meals_eaten	10000.0	1.001400	1.008117	0.000000	0.000000	1.000000	2.000000	7.000000
VitD_supp	10000.0	0.398900	0.628505	0.000000	0.000000	0.000000	1.000000	5.000000
Overweight	9018.0	0.709137	0.454186	0.000000	0.000000	1.000000	1.000000	1.000000
Anxiety	9016.0	0.322316	0.467389	0.000000	0.000000	0.000000	1.000000	1.000000
TotalCharge	10000.0	5891.538261	3377.558136	1256.751699	3253.239465	5852.250564	7614.989701	21524.224210
Additional_charges	10000.0	12934.528586	6542.601544	3125.702716	7986.487642	11573.979365	15626.491033	30566.073130
Item1	10000.0	3.518800	1.031966	1.000000	3.000000	4.000000	4.000000	8.000000
Item2	10000.0	3.506700	1.034825	1.000000	3.000000	3.000000	4.000000	7.000000
Item3	10000.0	3.511100	1.032755	1.000000	3.000000	4.000000	4.000000	8.000000
Item4	10000.0	3.515100	1.036282	1.000000	3.000000	4.000000	4.000000	7.000000
Item5	10000.0	3.496900	1.030192	1.000000	3.000000	3.000000	4.000000	7.000000
Item6	10000.0	3.522500	1.032376	1.000000	3.000000	4.000000	4.000000	7.000000
Item7	10000.0	3.494000	1.021405	1.000000	3.000000	3.000000	4.000000	7.000000
Item8	10000.0	3.509700	1.042312	1.000000	3.000000	3.000000	4.000000	7.000000

In [649…

```python
#Check for duplicates
Is_dups_bool = medical_df.duplicated()
print("Are there any duplicates? " + str(Is_dups_bool.value_counts()))
```

```
Are there any duplicates? False    10000
dtype: int64
```

In [650…

```python
#check for zeros
zeros = (medical_df["Population"] == 0).sum()
print(zeros)
```

```
109
```

In [651…

```python
#drop records with zeros in population field
count = 0
zeroList = []
for x in medical_df["Population"]:
    if x ==0:
        zeroList.append(count)
    count = count + 1
medical_df.drop(zeroList, axis=0, inplace=True)
medical_df.reset_index
medical_df.describe().transpose()
zeros = (medical_df["Population"] == 0).sum()
print(zeros)
```

```
0
```

In [652…

```python
#encode the data with numerics
my_dict3={"Yes": 1,"No": 0}
my_dict4={"Male": 1,"Female": 2,"Prefer not to answer": 0}
my_dict5={"Low":1,"Medium":2,"High":3}

medical_df_V2 = medical_df.replace({"ReAdmis": my_dict3,"Gender": my_dict4,"Soft_drink": my_dict3,
                    "HighBlood": my_dict3,
                    "Stroke": my_dict3,"Complication_risk":my_dict5,"Arthritis":my_dict3,
                    "Hyperlipidemia": my_dict3,"BackPain": my_dict3,"Diabetes":my_dict3,
                    "Allergic_rhinitis": my_dict3,"Reflux_esophagitis": my_dict3,"Asthma": my_dict3})
medical_df_V2.describe().transpose()
```

Out[652…

	count	mean	std	min	25%	50%	75%	max
Unnamed: 0	9891.0	5004.544839	2885.109958	1.000000	2507.500000	5004.000000	7502.500000	10000.000000
Lat	9891.0	38.761388	5.402855	17.967190	35.268175	39.440770	42.049850	70.560990
Lng	9891.0	-91.215769	15.176267	-174.209690	-97.319560	-88.375090	-80.444955	-65.290170
Population	9891.0	10075.072086	14869.065326	1.000000	730.000000	2859.000000	14161.000000	122814.000000
Age	7506.0	53.331601	20.664683	18.000000	35.000000	53.000000	71.000000	89.000000
Income	7466.0	40485.233793	28659.104817	154.080000	19454.922500	33947.305000	54107.657500	207249.130000
Gender	9891.0	1.480942	0.540864	0.000000	1.000000	2.000000	2.000000	2.000000
ReAdmis	9891.0	0.366899	0.481983	0.000000	0.000000	0.000000	1.000000	1.000000
VitD_levels	9891.0	19.415365	6.723257	9.519012	16.514963	18.078917	19.792348	53.019124
Full_meals_eaten	9891.0	1.000506	1.008006	0.000000	0.000000	1.000000	2.000000	7.000000
VitD_supp	9891.0	0.398847	0.628892	0.000000	0.000000	0.000000	1.000000	5.000000
Soft_drink	7446.0	0.258125	0.437633	0.000000	0.000000	0.000000	1.000000	1.000000
HighBlood	9891.0	0.409362	0.491741	0.000000	0.000000	0.000000	1.000000	1.000000
Stroke	9891.0	0.199070	0.399321	0.000000	0.000000	0.000000	0.000000	1.000000
Complication_risk	9891.0	2.123142	0.730667	1.000000	2.000000	2.000000	3.000000	3.000000
Overweight	8921.0	0.709562	0.453990	0.000000	0.000000	1.000000	1.000000	1.000000
Arthritis	9891.0	0.358103	0.479467	0.000000	0.000000	0.000000	1.000000	1.000000
Diabetes	9891.0	0.272975	0.445511	0.000000	0.000000	0.000000	1.000000	1.000000
Hyperlipidemia	9891.0	0.337883	0.473012	0.000000	0.000000	0.000000	1.000000	1.000000
BackPain	9891.0	0.411586	0.492146	0.000000	0.000000	0.000000	1.000000	1.000000
Anxiety	8916.0	0.321781	0.467186	0.000000	0.000000	0.000000	1.000000	1.000000
Allergic_rhinitis	9891.0	0.393287	0.488504	0.000000	0.000000	0.000000	1.000000	1.000000
Reflux_esophagitis	9891.0	0.413305	0.492452	0.000000	0.000000	0.000000	1.000000	1.000000
Asthma	9891.0	0.289859	0.453720	0.000000	0.000000	0.000000	1.000000	1.000000
TotalCharge	9891.0	5893.668940	3378.780141	1256.751699	3256.755457	5855.357730	7615.923547	21524.224210
Additional_charges	9891.0	12945.165184	6541.957948	3125.702716	7997.923737	11581.934090	15626.601495	30566.073130
Item1	9891.0	3.519968	1.032378	1.000000	3.000000	4.000000	4.000000	8.000000
Item2	9891.0	3.506723	1.035678	1.000000	3.000000	3.000000	4.000000	7.000000
Item3	9891.0	3.512587	1.033082	1.000000	3.000000	4.000000	4.000000	8.000000
Item4	9891.0	3.514811	1.037058	1.000000	3.000000	4.000000	4.000000	7.000000
Item5	9891.0	3.497826	1.029823	1.000000	3.000000	3.000000	4.000000	7.000000
Item6	9891.0	3.523607	1.032399	1.000000	3.000000	4.000000	4.000000	7.000000
Item7	9891.0	3.494692	1.023311	1.000000	3.000000	3.000000	4.000000	7.000000
Item8	9891.0	3.508543	1.042476	1.000000	3.000000	3.000000	4.000000	7.000000

In [653…

```python
#find fields and columns with nan values
nan_values = medical_df_V2.isnull()
nan_columns = nan_values.any()
columns_with_nan = medical_df_V2.columns[nan_columns].tolist()
print(columns_with_nan)
```

```
['Age', 'Income', 'Soft_drink', 'Overweight', 'Anxiety']
```

In [654…

```python
# Replace this set of clumns with a trinary value
medical_df_V2[['Soft_drink', 'Overweight',
             'Anxiety']] = medical_df_V2[['Soft_drink', 'Overweight', 'Anxiety']].fillna(value = 2)
```

In [655…

```python
#recheck for nulls in those columns
nan_values = medical_df_V2.isnull()
nan_columns = nan_values.any()
columns_with_nan = medical_df_V2.columns[nan_columns].tolist()
print(columns_with_nan)
```

```
['Age', 'Income']
```

In [656…

```python
#replace continuous values with randomly chosen values of similar tupe either integer or floats
count = 0;
for x in medical_df_V2["Age"]:
    randAge = random.randint(18,89)
    if math.isnan(x):
        medical_df_V2.iat[count,medical_df_V2.columns.get_loc("Age")] = randAge
    count = count + 1
    randAge = 0
count = 0
for x in medical_df_V2["Income"]:
    randInc = random.uniform(200.0,208000.0)
    if math.isnan(x):
        medical_df_V2.iat[count,medical_df_V2.columns.get_loc("Income")] = randInc
    count = count + 1
    randAge = 0
count = 0

nan_values = medical_df_V2.isnull()
nan_columns = nan_values.any()
columns_with_nan = medical_df_V2.columns[nan_columns].tolist()
print(columns_with_nan)
medical_df_V2.describe().transpose()
```

```
[]
```

Out[656…

	count	mean	std	min	25%	50%	75%	max
Unnamed: 0	9891.0	5004.544839	2885.109958	1.000000	2507.500000	5004.000000	7502.500000	10000.000000
Lat	9891.0	38.761388	5.402855	17.967190	35.268175	39.440770	42.049850	70.560990
Lng	9891.0	-91.215769	15.176267	-174.209690	-97.319560	-88.375090	-80.444955	-65.290170
Population	9891.0	10075.072086	14869.065326	1.000000	730.000000	2859.000000	14161.000000	122814.000000
Age	9891.0	53.362046	20.621395	18.000000	36.000000	53.000000	71.000000	89.000000
Income	9891.0	55823.829533	47187.667128	154.080000	22166.561077	40602.500000	73437.545767	207901.277703
Gender	9891.0	1.480942	0.540864	0.000000	1.000000	2.000000	2.000000	2.000000
ReAdmis	9891.0	0.366899	0.481983	0.000000	0.000000	0.000000	1.000000	1.000000
VitD_levels	9891.0	19.415365	6.723257	9.519012	16.514963	18.078917	19.792348	53.019124
Full_meals_eaten	9891.0	1.000506	1.008006	0.000000	0.000000	1.000000	2.000000	7.000000
VitD_supp	9891.0	0.398847	0.628892	0.000000	0.000000	0.000000	1.000000	5.000000
Soft_drink	9891.0	0.688707	0.841932	0.000000	0.000000	0.000000	1.000000	2.000000
HighBlood	9891.0	0.409362	0.491741	0.000000	0.000000	0.000000	1.000000	1.000000
Stroke	9891.0	0.199070	0.399321	0.000000	0.000000	0.000000	0.000000	1.000000
Complication_risk	9891.0	2.123142	0.730667	1.000000	2.000000	2.000000	3.000000	3.000000
Overweight	9891.0	0.836114	0.577234	0.000000	0.000000	1.000000	1.000000	2.000000
Arthritis	9891.0	0.358103	0.479467	0.000000	0.000000	0.000000	1.000000	1.000000
Diabetes	9891.0	0.272975	0.445511	0.000000	0.000000	0.000000	1.000000	1.000000
Hyperlipidemia	9891.0	0.337883	0.473012	0.000000	0.000000	0.000000	1.000000	1.000000
BackPain	9891.0	0.411586	0.492146	0.000000	0.000000	0.000000	1.000000	1.000000
Anxiety	9891.0	0.487211	0.668603	0.000000	0.000000	0.000000	1.000000	2.000000
Allergic_rhinitis	9891.0	0.393287	0.488504	0.000000	0.000000	0.000000	1.000000	1.000000
Reflux_esophagitis	9891.0	0.413305	0.492452	0.000000	0.000000	0.000000	1.000000	1.000000
Asthma	9891.0	0.289859	0.453720	0.000000	0.000000	0.000000	1.000000	1.000000
TotalCharge	9891.0	5893.668940	3378.780141	1256.751699	3256.755457	5855.357730	7615.923547	21524.224210
Additional_charges	9891.0	12945.165184	6541.957948	3125.702716	7997.923737	11581.934090	15626.601495	30566.073130
Item1	9891.0	3.519968	1.032378	1.000000	3.000000	4.000000	4.000000	8.000000
Item2	9891.0	3.506723	1.035678	1.000000	3.000000	3.000000	4.000000	7.000000
Item3	9891.0	3.512587	1.033082	1.000000	3.000000	4.000000	4.000000	8.000000
Item4	9891.0	3.514811	1.037058	1.000000	3.000000	4.000000	4.000000	7.000000
Item5	9891.0	3.497826	1.029823	1.000000	3.000000	3.000000	4.000000	7.000000
Item6	9891.0	3.523607	1.032399	1.000000	3.000000	4.000000	4.000000	7.000000
Item7	9891.0	3.494692	1.023311	1.000000	3.000000	3.000000	4.000000	7.000000
Item8	9891.0	3.508543	1.042476	1.000000	3.000000	3.000000	4.000000	7.000000

In [657…

```python
#Remane misleading and ambiguous columns
medical_df_V2.rename(columns = {'Income' : 'Household_Income',
                                'TotalCharge': 'Daily_Average_Charges',
                                'Item1':'Timely_Admission','Item2':'Timely_Treatment','Item3':'Timely_Visits',
                                'Item4':'Reliability','Item5':'Options','Item6':'Hours_Treatment',
                                'Item7':'Courteous_Staff','Item8':'Active_Listening'},inplace = True)
medical_df_V2.drop(columns = ['City', 'State', 'Area','Marital', 'Gender',
                              'Initial_admin', 'Services', 'Options','Full_meals_eaten'], axis=0, inplace=True)
medical_df_V2.describe().transpose()
```

Out[657…

	count	mean	std	min	25%	50%	75%	max
Unnamed: 0	9891.0	5004.544839	2885.109958	1.000000	2507.500000	5004.000000	7502.500000	10000.000000
Lat	9891.0	38.761388	5.402855	17.967190	35.268175	39.440770	42.049850	70.560990
Lng	9891.0	-91.215769	15.176267	-174.209690	-97.319560	-88.375090	-80.444955	-65.290170
Population	9891.0	10075.072086	14869.065326	1.000000	730.000000	2859.000000	14161.000000	122814.000000
Age	9891.0	53.362046	20.621395	18.000000	36.000000	53.000000	71.000000	89.000000
Household_Income	9891.0	55823.829533	47187.667128	154.080000	22166.561077	40602.500000	73437.545767	207901.277703
ReAdmis	9891.0	0.366899	0.481983	0.000000	0.000000	0.000000	1.000000	1.000000
VitD_levels	9891.0	19.415365	6.723257	9.519012	16.514963	18.078917	19.792348	53.019124
VitD_supp	9891.0	0.398847	0.628892	0.000000	0.000000	0.000000	1.000000	5.000000
Soft_drink	9891.0	0.688707	0.841932	0.000000	0.000000	0.000000	1.000000	2.000000
HighBlood	9891.0	0.409362	0.491741	0.000000	0.000000	0.000000	1.000000	1.000000
Stroke	9891.0	0.199070	0.399321	0.000000	0.000000	0.000000	0.000000	1.000000
Complication_risk	9891.0	2.123142	0.730667	1.000000	2.000000	2.000000	3.000000	3.000000
Overweight	9891.0	0.836114	0.577234	0.000000	0.000000	1.000000	1.000000	2.000000
Arthritis	9891.0	0.358103	0.479467	0.000000	0.000000	0.000000	1.000000	1.000000
Diabetes	9891.0	0.272975	0.445511	0.000000	0.000000	0.000000	1.000000	1.000000
Hyperlipidemia	9891.0	0.337883	0.473012	0.000000	0.000000	0.000000	1.000000	1.000000
BackPain	9891.0	0.411586	0.492146	0.000000	0.000000	0.000000	1.000000	1.000000
Anxiety	9891.0	0.487211	0.668603	0.000000	0.000000	0.000000	1.000000	2.000000
Allergic_rhinitis	9891.0	0.393287	0.488504	0.000000	0.000000	0.000000	1.000000	1.000000
Reflux_esophagitis	9891.0	0.413305	0.492452	0.000000	0.000000	0.000000	1.000000	1.000000
Asthma	9891.0	0.289859	0.453720	0.000000	0.000000	0.000000	1.000000	1.000000
Daily_Average_Charges	9891.0	5893.668940	3378.780141	1256.751699	3256.755457	5855.357730	7615.923547	21524.224210
Additional_charges	9891.0	12945.165184	6541.957948	3125.702716	7997.923737	11581.934090	15626.601495	30566.073130
Timely_Admission	9891.0	3.519968	1.032378	1.000000	3.000000	4.000000	4.000000	8.000000
Timely_Treatment	9891.0	3.506723	1.035678	1.000000	3.000000	3.000000	4.000000	7.000000
Timely_Visits	9891.0	3.512587	1.033082	1.000000	3.000000	4.000000	4.000000	8.000000
Reliability	9891.0	3.514811	1.037058	1.000000	3.000000	4.000000	4.000000	7.000000
Hours_Treatment	9891.0	3.523607	1.032399	1.000000	3.000000	4.000000	4.000000	7.000000
Courteous_Staff	9891.0	3.494692	1.023311	1.000000	3.000000	3.000000	4.000000	7.000000
Active_Listening	9891.0	3.508543	1.042476	1.000000	3.000000	3.000000	4.000000	7.000000

In [658…

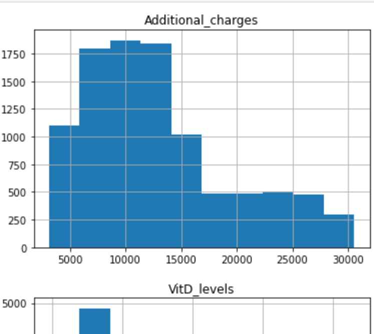
```python
medical_df_V2.columns
```

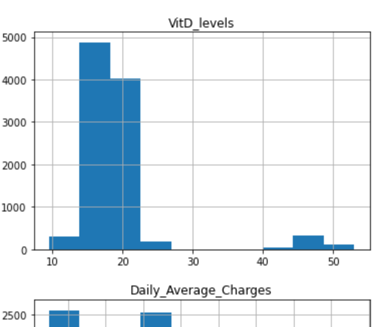
Out[658…

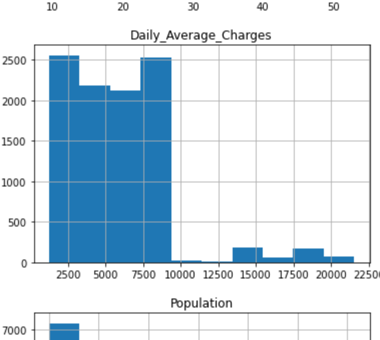
```
Index(['Unnamed: 0', 'Lat', 'Lng', 'Population', 'Age', 'Household_Income',
       'ReAdmis', 'VitD_levels', 'VitD_supp', 'Soft_drink', 'HighBlood',
       'Stroke', 'Complication_risk', 'Overweight', 'Arthritis', 'Diabetes',
       'Hyperlipidemia', 'BackPain', 'Anxiety', 'Allergic_rhinitis',
       'Reflux_esophagitis', 'Asthma', 'Daily_Average_Charges',
       'Additional_charges', 'Timely_Admission', 'Timely_Treatment',
       'Timely_Visits', 'Reliability', 'Hours_Treatment', 'Courteous_Staff',
       'Active_Listening'],
      dtype='object')
```

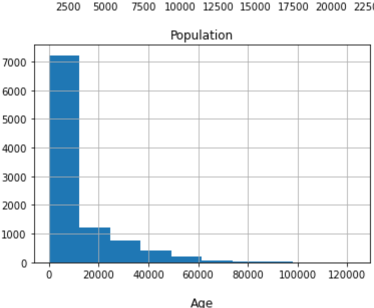
In [659…

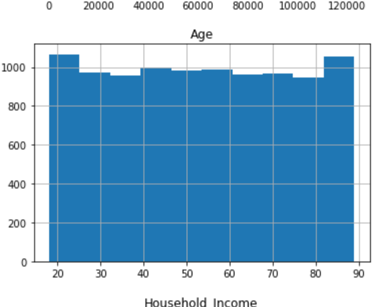
```python
Continuous ={'Lat', 'Lng', 'Population',  'Age',
       'Household_Income',  'VitD_levels',  'Daily_Average_Charges', 'Additional_charges'}

Categorical ={'VitD_supp', 'Soft_drink', 'HighBlood',
       'Stroke', 'Complication_risk', 'Overweight', 'Arthritis', 'Diabetes',
       'Hyperlipidemia', 'BackPain', 'Anxiety', 'Allergic_rhinitis',
       'Reflux_esophagitis', 'Asthma', 'Timely_Admission', 'Timely_Treatment',
       'Timely_Visits', 'Reliability', 'Hours_Treatment', 'Courteous_Staff',
       'Active_Listening'}

for x in Continuous:
    medical_df_V2[[x]].hist()
```

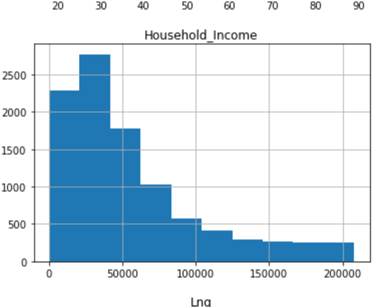


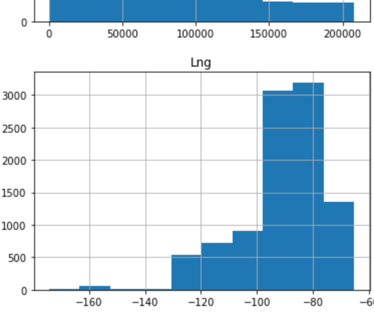


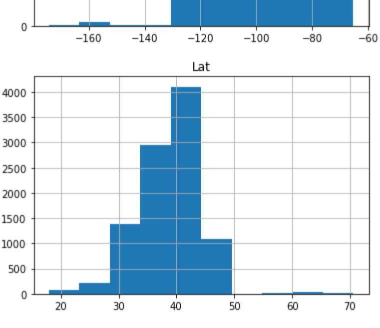










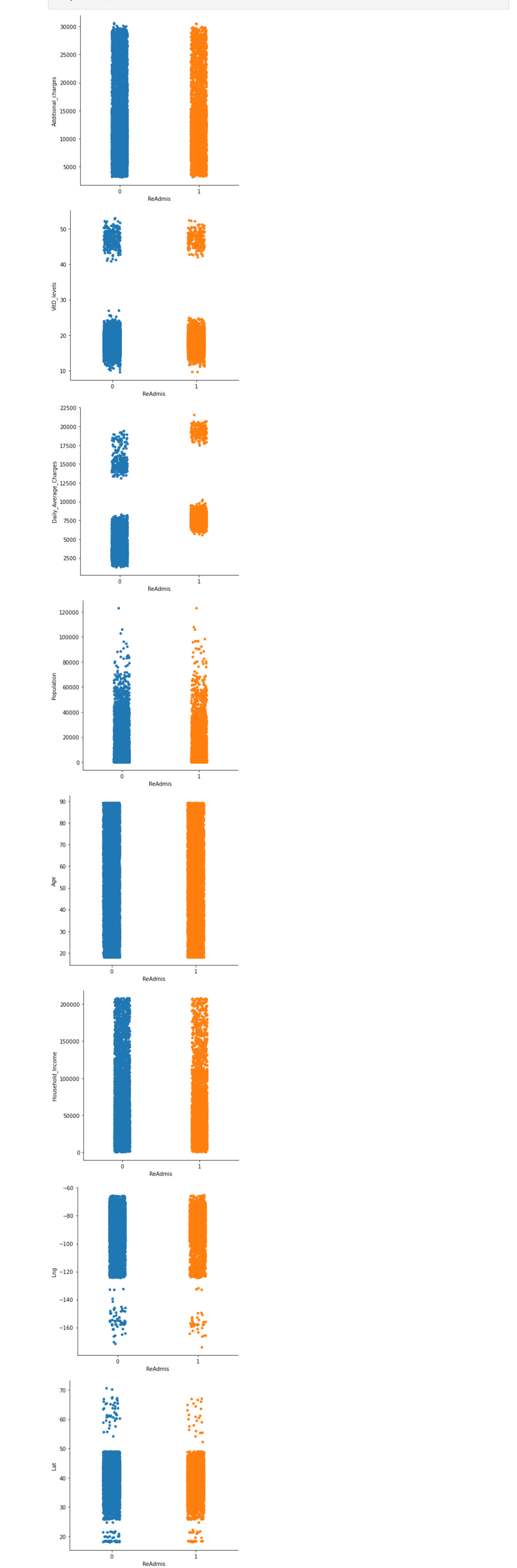


In [660…

```python
for x in  Categorical:
    table = pd.pivot_table(medical_df_V2.groupby([x]).size().reset_index(),values=0,index=x, columns=[x],aggfur
    table.plot(kind='bar',stacked=True)
    plt.show()
```

In [661…

```python
for cont in Continuous:
    sns.catplot(y=cont, x="ReAdmis", data=medical_df_V2)
    plt.show()
```



In [662…

```python
for  cat in Categorical:
    tbl = pd.pivot_table(medical_df_V2.groupby(['ReAdmis',cat]).size().reset_index(),
                         values=0,index='ReAdmis',columns=[cat],aggfunc=np.sum)
    tbl.plot(kind='bar',stacked=True)
```

In [663…

```python
medical_df_V2.to_csv("Prepared_medicalData_D208_Task2.csv", index=False)
```

In [664…

```python
medical_df_V2.columns
```

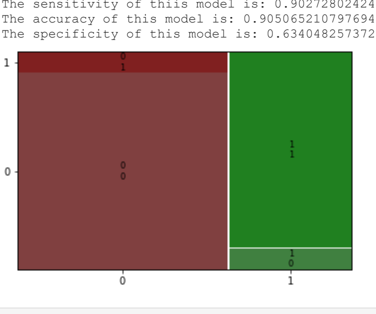
Out[664…
```
Index(['Unnamed: 0', 'Lat', 'Lng', 'Population', 'Age', 'Household_Income',
       'ReAdmis', 'VitD_levels', 'VitD_supp', 'Soft_drink', 'HighBlood',
       'Stroke', 'Complication_risk', 'Overweight', 'Arthritis', 'Diabetes',
       'Hyperlipidemia', 'BackPain', 'Anxiety', 'Allergic_rhinitis',
       'Reflux_esophagitis', 'Asthma', 'Daily_Average_Charges',
       'Additional_charges', 'Timely_Admission', 'Timely_Treatment',
       'Timely_Visits', 'Reliability', 'Hours_Treatment', 'Courteous_Staff',
       'Active_Listening'],
      dtype='object')
```

In [665…
```python
#build an initial model using logit
model_ReAdmis_V1 = sm.Logit(medical_df_V2['ReAdmis'],medical_df_V2[['Lat', 'Lng', 'Population', 'Age', 'Househo
       'VitD_levels', 'VitD_supp', 'Soft_drink', 'HighBlood',
       'Stroke', 'Complication_risk', 'Overweight', 'Arthritis', 'Diabetes',
       'Hyperlipidemia', 'BackPain', 'Anxiety', 'Allergic_rhinitis',
       'Reflux_esophagitis', 'Asthma', 'Daily_Average_Charges',
       'Additional_charges', 'Timely_Admission', 'Timely_Treatment',
       'Timely_Visits', 'Reliability', 'Hours_Treatment', 'Courteous_Staff',
       'Active_Listening']]).fit()
print(model_ReAdmis_V1.params)
print(model_ReAdmis_V1.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.213818
         Iterations 9
Lat                      0.003802
Lng                      0.000435
Population               0.000003
Age                     -0.001773
Household_Income         0.000001
VitD_levels             -0.642733
VitD_supp               -0.054509
Soft_drink               0.031811
HighBlood               -0.172691
Stroke                   0.129099
Complication_risk       -0.274730
Overweight               0.017370
Arthritis               -0.219184
Diabetes                -0.165793
Hyperlipidemia          -0.148815
BackPain                -0.106858
Anxiety                 -0.125126
Allergic_rhinitis       -0.263622
Reflux_esophagitis      -0.165699
Asthma                  -0.110501
Daily_Average_Charges    0.001862
Additional_charges       0.000007
Timely_Admission         0.054570
Timely_Treatment         0.092855
Timely_Visits           -0.030254
Reliability              0.068076
Hours_Treatment         -0.058190
Courteous_Staff         -0.026422
Active_Listening        -0.066533
dtype: float64
                           Logit Regression Results
==============================================================================
Dep. Variable:                ReAdmis   No. Observations:                 9891
Model:                          Logit   Df Residuals:                     9862
Method:                           MLE   Df Model:                           28
Date:                Mon, 30 May 2022   Pseudo R-squ.:                  0.6747
Time:                        21:11:17   Log-Likelihood:                -2114.9
converged:                       True   LL-Null:                       -6501.2
Covariance Type:            nonrobust   LLR p-value:                     0.000
=========================================================================================
                            coef    std err          z      P>|z|      [0.025      0.975]
-----------------------------------------------------------------------------------------
Lat                       0.0038      0.006      0.615      0.538      -0.008       0.016
Lng                       0.0004      0.002      0.187      0.852      -0.004       0.005
Population             2.655e-06   2.69e-06      0.988      0.323   -2.61e-06    7.92e-06
Age                      -0.0018      0.003     -0.667      0.505      -0.007       0.003
Household_Income       1.031e-06    8.6e-07      1.199      0.230   -6.54e-07    2.72e-06
VitD_levels              -0.6427      0.016    -40.229      0.000      -0.674      -0.611
VitD_supp                -0.0545      0.063     -0.869      0.385      -0.177       0.068
Soft_drink                0.0318      0.047      0.680      0.497      -0.060       0.124
HighBlood                -0.1727      0.126     -1.370      0.171      -0.420       0.074
Stroke                    0.1291      0.098      1.315      0.189      -0.063       0.322
Complication_risk        -0.2747      0.054     -5.132      0.000      -0.380      -0.170
Overweight                0.0174      0.066      0.261      0.794      -0.113       0.148
Arthritis                -0.2192      0.082     -2.685      0.007      -0.379      -0.059
Diabetes                 -0.1658      0.087     -1.898      0.058      -0.337       0.005
Hyperlipidemia           -0.1488      0.084     -1.778      0.075      -0.313       0.015
BackPain                 -0.1069      0.079     -1.345      0.179      -0.263       0.049
Anxiety                  -0.1251      0.058     -2.141      0.032      -0.240      -0.011
Allergic_rhinitis        -0.2636      0.080     -3.289      0.001      -0.421      -0.107
Reflux_esophagitis       -0.1657      0.080     -2.076      0.038      -0.322      -0.009
Asthma                   -0.1105      0.087     -1.272      0.203      -0.281       0.060
Daily_Average_Charges     0.0019    4.4e-05     42.265      0.000       0.002       0.002
Additional_charges     7.281e-06   1.12e-05      0.651      0.515   -1.46e-05    2.92e-05
Timely_Admission          0.0546      0.056      0.972      0.331      -0.055       0.165
Timely_Treatment          0.0929      0.052      1.782      0.075      -0.009       0.195
Timely_Visits            -0.0303      0.047     -0.641      0.522      -0.123       0.062
Reliability               0.0681      0.039      1.750      0.080      -0.008       0.144
Hours_Treatment          -0.0582      0.047     -1.248      0.212      -0.150       0.033
Courteous_Staff          -0.0264      0.042     -0.625      0.532      -0.109       0.056
Active_Listening         -0.0665      0.041     -1.626      0.104      -0.147       0.014
=========================================================================================
```

In [666…
```python
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix
from statsmodels.graphics.mosaicplot import mosaic
# Split the dataset into a training and testing set. Using an 80/20 testing/training split.
actual_response1 = medical_df_V2["ReAdmis"]

predicted_response1 = np.round(model_ReAdmis_V1.predict())

outcomes1 = pd.DataFrame({"actual_response": actual_response1,"predicted_response": predicted_response1})
print(outcomes1.value_counts(sort=False))
print(model_ReAdmis_V1.pred_table())
mosaic(model_ReAdmis_V1.pred_table())
TN1 = model_ReAdmis_V1.pred_table()[0,0]
TP1 = model_ReAdmis_V1.pred_table()[1,1]
FN1 = model_ReAdmis_V1.pred_table()[1,0]
FP1 = model_ReAdmis_V1.pred_table()[0,1]

sensitivity1 = TP1/(FN1 +TP1)
accuracy1 = (TN1 + TP1)/(TN1 + FN1 + FP1 + TP1)
specificity1 = TN1/(TN1 + TP1)
print("The sensitivity of thiis model is: " + str(sensitivity1))
print("The accuracy of this model is: " + str(accuracy1))
print("The specificity of this model is: " +str(specificity1))
```

```
actual_response  predicted_response
0                0.0                   5676
                 1.0                    586
1                0.0                    353
                 1.0                   3276
dtype: int64
[[5676.  586.]
 [ 353. 3276.]]
The sensitivity of thiis model is: 0.9027280242491045
The accuracy of this model is: 0.9050652107976949
The specificity of this model is: 0.6340482573726541
```



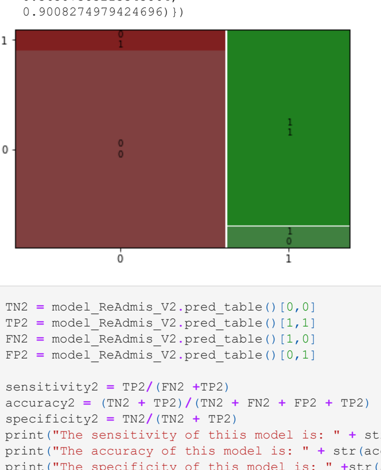
In [667…
```python
model_ReAdmis_V2 = sm.Logit(medical_df_V2['ReAdmis'],medical_df_V2[['VitD_levels','Complication_risk',
                                                                      'Arthritis','Anxiety','Allergic_rhinitis',
                                                                      'Reflux_esophagitis',
                                                                      'Daily_Average_Charges']]).fit()
print(model_ReAdmis_V2.params)
print(model_ReAdmis_V2.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.215313
         Iterations 9
VitD_levels             -0.636669
Complication_risk       -0.259786
Arthritis               -0.204376
Anxiety                 -0.110353
Allergic_rhinitis       -0.248485
Reflux_esophagitis      -0.154311
Daily_Average_Charges    0.001852
dtype: float64
                           Logit Regression Results
==============================================================================
Dep. Variable:                ReAdmis   No. Observations:                 9891
Model:                          Logit   Df Residuals:                     9884
Method:                           MLE   Df Model:                            6
Date:                Mon, 30 May 2022   Pseudo R-squ.:                  0.6724
Time:                        21:11:17   Log-Likelihood:                -2129.7
converged:                       True   LL-Null:                       -6501.2
Covariance Type:            nonrobust   LLR p-value:                     0.000
=========================================================================================
                            coef    std err          z      P>|z|      [0.025      0.975]
-----------------------------------------------------------------------------------------
VitD_levels              -0.6367      0.016    -40.924      0.000      -0.667      -0.606
Complication_risk        -0.2598      0.045     -5.783      0.000      -0.348      -0.172
Arthritis                -0.2044      0.080     -2.548      0.011      -0.362      -0.047
Anxiety                  -0.1104      0.058     -1.915      0.055      -0.223       0.003
Allergic_rhinitis        -0.2485      0.079     -3.150      0.002      -0.403      -0.094
Reflux_esophagitis       -0.1543      0.078     -1.977      0.048      -0.307      -0.001
Daily_Average_Charges     0.0019   4.34e-05     42.670      0.000       0.002       0.002
=========================================================================================
```

In [668…
```python
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix
from statsmodels.graphics.mosaicplot import mosaic
# Split the dataset into a training and testing set. Using an 80/20 testing/training split.
actual_response2 = medical_df_V2["ReAdmis"]

predicted_response2 = np.round(model_ReAdmis_V2.predict())

outcomes2 = pd.DataFrame({"actual_response": actual_response2,"predicted_response": predicted_response2})
print(outcomes2.value_counts(sort=False))
print(model_ReAdmis_V2.pred_table())
mosaic(model_ReAdmis_V2.pred_table())
```

```
actual_response  predicted_response
0                0.0                   5679
                 1.0                    583
1                0.0                    349
                 1.0                   3280
dtype: int64
[[5679.  583.]
 [ 349. 3280.]]
```

Out[668…
```
(<Figure size 432x288 with 3 Axes>,
 {('0', '0'): (0.0, 0.0, 0.629951043488452, 0.9038858017191709),
  ('0', '1'): (0.0, 0.9072080608553835, 0.629951043488452, 0.0927919391446164),
  ('1', '0'): (0.6349261678665614,
   0.0,
   0.36507383213343864,
   0.09585024292131765),
  ('1', '1'): (0.6349261678665614,
   0.09917250205753027,
   0.36507383213343864,
   0.9008274979424696)})
```



In [669…
```python
TN2 = model_ReAdmis_V2.pred_table()[0,0]
TP2 = model_ReAdmis_V2.pred_table()[1,1]
FN2 = model_ReAdmis_V2.pred_table()[1,0]
FP2 = model_ReAdmis_V2.pred_table()[0,1]

sensitivity2 = TP2/(FN2 +TP2)
accuracy2 = (TN2 + TP2)/(TN2 + FN2 + FP2 + TP2)
specificity2 = TN2/(TN2 + TP2)
print("The sensitivity of thiis model is: " + str(sensitivity2))
print("The accuracy of this model is: " + str(accuracy2))
print("The specificity of this model is: " +str(specificity2))
```

```
The sensitivity of thiis model is: 0.9038302562689446
The accuracy of this model is: 0.9057729248812051
The specificity of this model is: 0.6338877106819958
```

In [ ]: