In [1]:
```
import pandas as pd
import numpy as np
from datetime import datetime
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import plotly.express as px
import scipy.stats
from scipy.stats import ttest_1samp
from scipy.stats import ttest_ind
from sklearn.preprocessing import LabelEncoder
from sklearn.feature_selection import chi2
from dython.nominal import identify_nominal_columns
from dython.nominal import associations
from statistics import mean, stdev
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.model_selection import StratifiedKFold
from sklearn.model_selection import cross_val_score, KFold
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import SGDRegressor
from sklearn.pipeline import make_pipeline
from sklearn.model_selection import GridSearchCV
from sklearn import linear_model
from sklearn import datasets
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
import joblib
```

In [2]:
```
DF1 = pd.read_csv(r"D:\Data Science\Data Science Career Readiness Program\Datasets\Cap
DF1.head()
```

Out[2]:

	Customer ID	year	month	date	children	charges	Hospital tier	City tier	State ID
0	Id2335	1992	Jul	9	0	563.84	tier - 2	tier - 3	R1013
1	Id2334	1992	Nov	30	0	570.62	tier - 2	tier - 1	R1013
2	Id2333	1993	Jun	30	0	600.00	tier - 2	tier - 1	R1013
3	Id2332	1992	Sep	13	0	604.54	tier - 3	tier - 3	R1013
4	Id2331	1998	Jul	27	0	637.26	tier - 3	tier - 3	R1013

In [3]:
```
DF1.shape
```

Out[3]:
```
(2343, 9)
```

In [4]:
```
DF2 = pd.read_csv(r"D:\Data Science\Data Science Career Readiness Program\Datasets\Cap
DF2.head()
```

Out[4]:

	Customer ID	BMI	HBA1C	Heart Issues	Any Transplants	Cancer history	NumberOfMajorSurgeries	smoker
0	Id1	47.410	7.47	No	No	No	No major surgery	yes
1	Id2	30.360	5.77	No	No	No	No major surgery	yes
2	Id3	34.485	11.87	yes	No	No	2	yes
3	Id4	38.095	6.05	No	No	No	No major surgery	yes
4	Id5	35.530	5.45	No	No	No	No major surgery	yes

In [5]:
```
DF2.shape
```

Out[5]:
```
(2335, 8)
```

In [6]:
```
xl_file = pd.ExcelFile(r"D:\Data Science\Data Science Career Readiness Program\Dataset
```

In [7]:
```
DF3 = xl_file.parse('Sheet1')
DF3.head()
```

Out[7]:

	Customer ID	name
0	Id1	Hawks, Ms. Kelly
1	Id2	Lehner, Mr. Matthew D
2	Id3	Lu, Mr. Phil
3	Id4	Osborne, Ms. Kelsey
4	Id5	Kadala, Ms. Kristyn

In [8]:
```
DF3.shape
```

Out[8]:
```
(2335, 2)
```

In [9]:
```
HealthCare = pd.merge(DF1, DF2, on = "Customer ID", how = "inner").merge(DF3, on = "Cu
HealthCare
```

Out[9]:

	Customer ID	year	month	date	children	charges	Hospital tier	City tier	State ID	BMI	HBA1C	Hear Issue
0	Id2335	1992	Jul	9	0	563.84	tier - 2	tier - 3	R1013	17.580	4.51	N
1	Id2334	1992	Nov	30	0	570.62	tier - 2	tier - 1	R1013	17.600	4.39	N
2	Id2333	1993	Jun	30	0	600.00	tier - 2	tier - 1	R1013	16.470	6.35	N
3	Id2332	1992	Sep	13	0	604.54	tier - 3	tier - 3	R1013	17.700	6.28	N
4	Id2331	1998	Jul	27	0	637.26	tier - 3	tier - 3	R1013	22.340	5.57	N
...	...	...	...	...	...	...	...	...	...	...	...	
2330	Id5	1989	Jun	19	0	55135.40	tier - 1	tier - 2	R1012	35.530	5.45	N
2331	Id4	1991	Jun	6	1	58571.07	tier - 1	tier - 3	R1024	38.095	6.05	N
2332	Id3	1970	?	11	3	60021.40	tier - 1	tier - 1	R1012	34.485	11.87	ye
2333	Id2	1977	Jun	8	0	62592.87	tier - 2	tier - 3	R1013	30.360	5.77	N
2334	Id1	1968	Oct	12	0	63770.43	tier - 1	tier - 3	R1013	47.410	7.47	N

2335 rows × 17 columns

In [10]:
```
HealthCare.shape
```

Out[10]:
```
(2335, 17)
```

In [11]:
```
HealthCare.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2335 entries, 0 to 2334
Data columns (total 17 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Customer ID             2335 non-null   object 
 1   year                    2335 non-null   object 
 2   month                   2335 non-null   object 
 3   date                    2335 non-null   int64  
 4   children                2335 non-null   int64  
 5   charges                 2335 non-null   float64
 6   Hospital tier           2335 non-null   object 
 7   City tier               2335 non-null   object 
 8   State ID                2335 non-null   object 
 9   BMI                     2335 non-null   float64
 10  HBA1C                   2335 non-null   float64
 11  Heart Issues            2335 non-null   object 
 12  Any Transplants         2335 non-null   object 
 13  Cancer history          2335 non-null   object 
 14  NumberOfMajorSurgeries  2335 non-null   object 
 15  smoker                  2335 non-null   object 
 16  name                    2335 non-null   object 
dtypes: float64(3), int64(2), object(12)
memory usage: 328.4+ KB
```

In [12]:
```
HealthCare.describe()
```

Out[12]:

	date	children	charges	BMI	HBA1C
count	2335.000000	2335.000000	2335.000000	2335.000000	2335.000000
mean	15.563597	1.025696	13529.918034	30.972649	6.578998
std	8.720508	1.234754	11898.654299	8.742095	2.228731
min	1.000000	0.000000	563.840000	15.010000	4.000000
25%	8.000000	0.000000	5084.010000	24.600000	4.900000
50%	15.000000	0.000000	9630.910000	30.400000	5.810000
75%	23.000000	2.000000	16912.295000	36.300000	7.955000
max	30.000000	5.000000	63770.430000	55.050000	12.000000

In [13]:
```
HealthCare.isnull().sum()
```

Out[13]:
```
Customer ID              0
year                     0
month                    0
date                     0
children                 0
charges                  0
Hospital tier            0
City tier                0
State ID                 0
BMI                      0
HBA1C                    0
Heart Issues             0
Any Transplants          0
Cancer history           0
NumberOfMajorSurgeries   0
smoker                   0
name                     0
dtype: int64
```

In [14]:
```
HealthCare.columns
```

Out[14]:
```
Index(['Customer ID', 'year', 'month', 'date', 'children', 'charges',
       'Hospital tier', 'City tier', 'State ID', 'BMI', 'HBA1C',
       'Heart Issues', 'Any Transplants', 'Cancer history',
       'NumberOfMajorSurgeries', 'smoker', 'name'],
      dtype='object')
```

In [15]:
```
index_df = HealthCare[(HealthCare['Customer ID'] == '?') |
                      (HealthCare['year'] == '?') |
                      (HealthCare['month'] == '?') |
                      (HealthCare['date'] == '?') |
                      (HealthCare['children'] == '?') |
                      (HealthCare['charges'] == '?') |
                      (HealthCare['Hospital tier'] == '?') |
                      (HealthCare['City tier'] == '?') |
                      (HealthCare['State ID'] == '?') |
                      (HealthCare['BMI'] == '?') |
                      (HealthCare['HBA1C'] == '?') |
                      (HealthCare['Heart Issues'] == '?') |
                      (HealthCare['Any Transplants'] == '?') |
                      (HealthCare['Cancer history'] == '?') |
                      (HealthCare['NumberOfMajorSurgeries'] == '?') |
                      (HealthCare['smoker'] == '?') |
                      (HealthCare['name'] == '?')].index
```

In [16]:
```
HealthCare = HealthCare.drop(index_df)
HealthCare
```

Out[16]:

	Customer ID	year	month	date	children	charges	Hospital tier	City tier	State ID	BMI	HBA1C	Hear Issue
0	Id2335	1992	Jul	9	0	563.84	tier - 2	tier - 3	R1013	17.580	4.51	N
1	Id2334	1992	Nov	30	0	570.62	tier - 2	tier - 1	R1013	17.600	4.39	N
2	Id2333	1993	Jun	30	0	600.00	tier - 2	tier - 1	R1013	16.470	6.35	N
3	Id2332	1992	Sep	13	0	604.54	tier - 3	tier - 3	R1013	17.700	6.28	N
4	Id2331	1998	Jul	27	0	637.26	tier - 3	tier - 3	R1013	22.340	5.57	N
...	...	...	...	...	...	...	...	...	...	...	...	
2329	Id6	1962	Aug	4	0	52590.83	tier - 1	tier - 3	R1011	32.800	6.59	N
2330	Id5	1989	Jun	19	0	55135.40	tier - 1	tier - 2	R1012	35.530	5.45	N
2331	Id4	1991	Jun	6	1	58571.07	tier - 1	tier - 3	R1024	38.095	6.05	N
2333	Id2	1977	Jun	8	0	62592.87	tier - 2	tier - 3	R1013	30.360	5.77	N
2334	Id1	1968	Oct	12	0	63770.43	tier - 1	tier - 3	R1013	47.410	7.47	N

2325 rows × 17 columns

In [17]:
```
((2335-2325)/2335)*100
```

Out[17]:
```
0.4282655246252677
```

In [18]:
```
pd.value_counts(HealthCare['State ID'])
```

Out[18]:
```
R1013    609
R1011    574
R1012    572
R1024    159
R1026     84
R1021     70
R1016     64
R1025     40
R1023     38
R1017     36
R1019     26
R1022     14
R1014     13
R1015     11
R1018      9
R1020      6
Name: State ID, dtype: int64
```

In [19]:
```
State_dummy = pd.get_dummies(HealthCare['State ID']).iloc[:,0:3]
State_dummy
```

Out[19]:

	R1011	R1012	R1013
0	0	0	1
1	0	0	1
2	0	0	1
3	0	0	1
4	0	0	1
...	...	...	...
2329	1	0	0
2330	0	1	0
2331	0	0	0
2333	0	0	1
2334	0	0	1

2325 rows × 3 columns

In [20]:
```
HCI = pd.concat([HealthCare, State_dummy], axis = 1)
HCI
```

Out[20]:

	Customer ID	year	month	date	children	charges	Hospital tier	City tier	State ID	BMI	HBA1C	Hear Issue
0	Id2335	1992	Jul	9	0	563.84	tier - 2	tier - 3	R1013	17.580	4.51	N
1	Id2334	1992	Nov	30	0	570.62	tier - 2	tier - 1	R1013	17.600	4.39	N
2	Id2333	1993	Jun	30	0	600.00	tier - 2	tier - 1	R1013	16.470	6.35	N
3	Id2332	1992	Sep	13	0	604.54	tier - 3	tier - 3	R1013	17.700	6.28	N
4	Id2331	1998	Jul	27	0	637.26	tier - 3	tier - 3	R1013	22.340	5.57	N
...	...	...	...	...	...	...	...	...	...	...	...	
2329	Id6	1962	Aug	4	0	52590.83	tier - 1	tier - 3	R1011	32.800	6.59	N
2330	Id5	1989	Jun	19	0	55135.40	tier - 1	tier - 2	R1012	35.530	5.45	N
2331	Id4	1991	Jun	6	1	58571.07	tier - 1	tier - 3	R1024	38.095	6.05	N
2333	Id2	1977	Jun	8	0	62592.87	tier - 2	tier - 3	R1013	30.360	5.77	N
2334	Id1	1968	Oct	12	0	63770.43	tier - 1	tier - 3	R1013	47.410	7.47	N

2325 rows × 20 columns

In [21]:
```
HCI.NumberOfMajorSurgeries.unique()
```

Out[21]:
```
array(['1', 'No major surgery', '2', '3'], dtype=object)
```

In [22]:
```
HCI['NumberOfMajorSurgeries'] = pd.to_numeric(HCI['NumberOfMajorSurgeries'], errors =
```

In [23]:
```
HCI.NumberOfMajorSurgeries.unique()
```

Out[23]:
```
array([ 1., nan,  2.,  3.])
```

In [24]:
```
HCI.fillna({'NumberOfMajorSurgeries':0}, inplace = True)
HCI
```

Out[24]:

	Customer ID	year	month	date	children	charges	Hospital tier	City tier	State ID	BMI	HBA1C	Hear Issue
0	Id2335	1992	Jul	9	0	563.84	tier - 2	tier - 3	R1013	17.580	4.51	N
1	Id2334	1992	Nov	30	0	570.62	tier - 2	tier - 1	R1013	17.600	4.39	N
2	Id2333	1993	Jun	30	0	600.00	tier - 2	tier - 1	R1013	16.470	6.35	N
3	Id2332	1992	Sep	13	0	604.54	tier - 3	tier - 3	R1013	17.700	6.28	N
4	Id2331	1998	Jul	27	0	637.26	tier - 3	tier - 3	R1013	22.340	5.57	N
...	...	...	...	...	...	...	...	...	...	...	...	
2329	Id6	1962	Aug	4	0	52590.83	tier - 1	tier - 3	R1011	32.800	6.59	N
2330	Id5	1989	Jun	19	0	55135.40	tier - 1	tier - 2	R1012	35.530	5.45	N
2331	Id4	1991	Jun	6	1	58571.07	tier - 1	tier - 3	R1024	38.095	6.05	N
2333	Id2	1977	Jun	8	0	62592.87	tier - 2	tier - 3	R1013	30.360	5.77	N
2334	Id1	1968	Oct	12	0	63770.43	tier - 1	tier - 3	R1013	47.410	7.47	N

2325 rows × 20 columns

In [25]:
```
HCI.NumberOfMajorSurgeries.dtype
```

Out[25]:
```
dtype('float64')
```

In [26]:
```
HCI['NumberOfMajorSurgeries'] = HCI['NumberOfMajorSurgeries'].astype(int)
```

In [27]:
```
HCI.NumberOfMajorSurgeries.dtype
```

Out[27]:
```
dtype('int32')
```

In [28]:
```
HCI
```

Out[28]:

	Customer ID	year	month	date	children	charges	Hospital tier	City tier	State ID	BMI	HBA1C	Hear Issue
0	Id2335	1992	Jul	9	0	563.84	tier - 2	tier - 3	R1013	17.580	4.51	N
1	Id2334	1992	Nov	30	0	570.62	tier - 2	tier - 1	R1013	17.600	4.39	N
2	Id2333	1993	Jun	30	0	600.00	tier - 2	tier - 1	R1013	16.470	6.35	N
3	Id2332	1992	Sep	13	0	604.54	tier - 3	tier - 3	R1013	17.700	6.28	N
4	Id2331	1998	Jul	27	0	637.26	tier - 3	tier - 3	R1013	22.340	5.57	N
...	...	...	...	...	...	...	...	...	...	...	...	
2329	Id6	1962	Aug	4	0	52590.83	tier - 1	tier - 3	R1011	32.800	6.59	N
2330	Id5	1989	Jun	19	0	55135.40	tier - 1	tier - 2	R1012	35.530	5.45	N
2331	Id4	1991	Jun	6	1	58571.07	tier - 1	tier - 3	R1024	38.095	6.05	N
2333	Id2	1977	Jun	8	0	62592.87	tier - 2	tier - 3	R1013	30.360	5.77	N
2334	Id1	1968	Oct	12	0	63770.43	tier - 1	tier - 3	R1013	47.410	7.47	N

2325 rows × 20 columns

In [29]:
```
HCI.month.unique()
```

Out[29]:
```
array(['Jul', 'Nov', 'Jun', 'Sep', 'Dec', 'Aug', 'Oct'], dtype=object)
```

In [30]:
```
conditions = [(HCI['month'] == 'Jun'), (HCI['month'] == 'Jul'), (HCI['month'] == 'Aug
              (HCI['month'] == 'Sep'), (HCI['month'] == 'Oct'), (HCI['month'] == 'Nov')

choices = ['06', '07', '08', '09', '10', '11', '12']

HCI['month'] = np.select(conditions, choices, default = '')
```

In [31]:
```
HCI.month.unique()
```

Out[31]:
```
array(['07', '11', '06', '09', '12', '08', '10'], dtype=object)
```

In [32]:
```
HCI.head()
```

Out[32]:

	Customer ID	year	month	date	children	charges	Hospital tier	City tier	State ID	BMI	HBA1C	Heart Issues	Tra
0	Id2335	1992	07	9	0	563.84	tier - 2	tier - 3	R1013	17.58	4.51	No	
1	Id2334	1992	11	30	0	570.62	tier - 2	tier - 1	R1013	17.60	4.39	No	
2	Id2333	1993	06	30	0	600.00	tier - 2	tier - 1	R1013	16.47	6.35	No	
3	Id2332	1992	09	13	0	604.54	tier - 3	tier - 3	R1013	17.70	6.28	No	
4	Id2331	1998	07	27	0	637.26	tier - 3	tier - 3	R1013	22.34	5.57	No	

In [33]:
```
HCI['DOB'] = pd.to_datetime(HCI['year'].astype(str) + '-' + HCI['month'].astype(str) +
```

In [34]:
```
HCI.head()
```

Out[34]:

	Customer ID	year	month	date	children	charges	Hospital tier	City tier	State ID	BMI	...	Heart Issues	Transpla
0	Id2335	1992	07	9	0	563.84	tier - 2	tier - 3	R1013	17.58	...	No	
1	Id2334	1992	11	30	0	570.62	tier - 2	tier - 1	R1013	17.60	...	No	
2	Id2333	1993	06	30	0	600.00	tier - 2	tier - 1	R1013	16.47	...	No	
3	Id2332	1992	09	13	0	604.54	tier - 3	tier - 3	R1013	17.70	...	No	
4	Id2331	1998	07	27	0	637.26	tier - 3	tier - 3	R1013	22.34	...	No	

5 rows × 21 columns

In [35]:
```
now = pd.Timestamp('now')
HCI['DOB'] = HCI['DOB'].where(HCI['DOB'] < now, HCI['DOB'] - np.timedelta64(100, 'Y'))
HCI['age'] = (now - HCI['DOB']).astype('<m8[Y]')
HCI['age'] = HCI['age'].astype(int)
```

In [36]:
```
HCI.head()
```

Out[36]:

	Customer ID	year	month	date	children	charges	Hospital tier	City tier	State ID	BMI	...	Any Transplants	Ca his
0	Id2335	1992	07	9	0	563.84	tier - 2	tier - 3	R1013	17.58	...	No	
1	Id2334	1992	11	30	0	570.62	tier - 2	tier - 1	R1013	17.60	...	No	
2	Id2333	1993	06	30	0	600.00	tier - 2	tier - 1	R1013	16.47	...	No	
3	Id2332	1992	09	13	0	604.54	tier - 3	tier - 3	R1013	17.70	...	No	
4	Id2331	1998	07	27	0	637.26	tier - 3	tier - 3	R1013	22.34	...	No	

5 rows × 22 columns

In [37]:
```
HCI.drop(['year', 'month', 'date'], axis = 1, inplace = True)
HCI.head()
```

Out[37]:

	Customer ID	children	charges	Hospital tier	City tier	State ID	BMI	HBA1C	Heart Issues	Any Transplants	Cancer history	N
0	Id2335	0	563.84	tier - 2	tier - 3	R1013	17.58	4.51	No	No	No	
1	Id2334	0	570.62	tier - 2	tier - 1	R1013	17.60	4.39	No	No	No	
2	Id2333	0	600.00	tier - 2	tier - 1	R1013	16.47	6.35	No	No	Yes	
3	Id2332	0	604.54	tier - 3	tier - 3	R1013	17.70	6.28	No	No	No	
4	Id2331	0	637.26	tier - 3	tier - 3	R1013	22.34	5.57	No	No	No	

In [38]:
```
g = []

for text in HCI.name:
    if ' Mr. ' in text:
        g.append('M')
    elif 'Ms.' in text:
        g.append('F')
    elif 'Mrs.' in text:
        g.append('F')
    else:
        g.append('O')
```

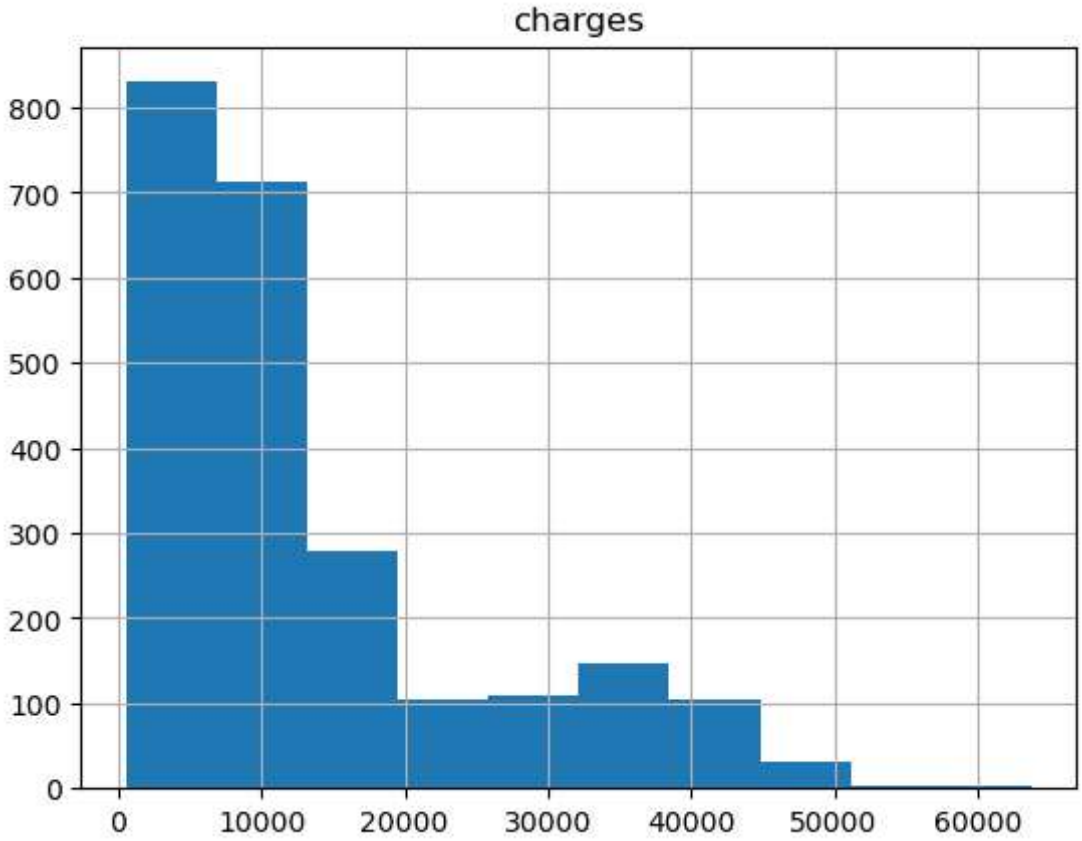
```
HCI["Gender"] = g
```

In [39]:
```
HCI.Gender.unique()
```

Out[39]:
```
array(['M', 'F'], dtype=object)
```

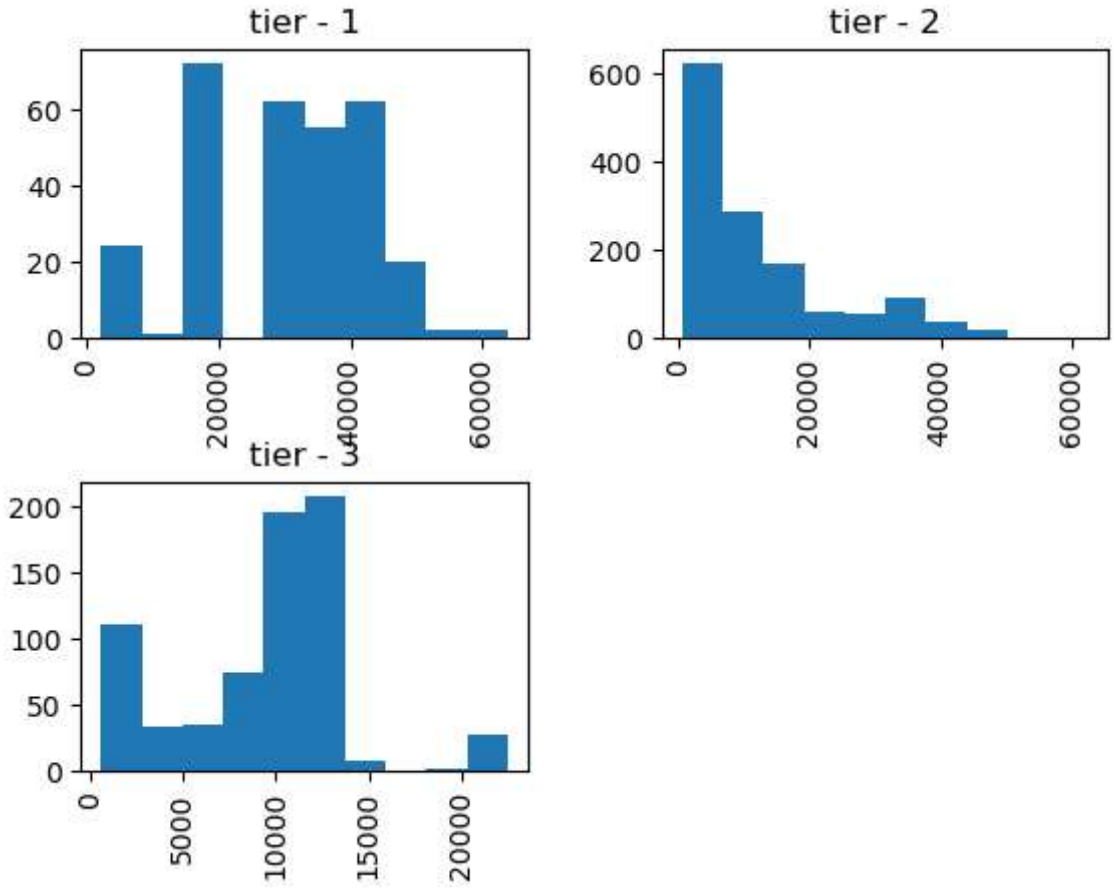
In [40]:
```
HCI.hist(column='charges')
```

Out[40]:
```
array([[<AxesSubplot:title={'center':'charges'}>]], dtype=object)
```



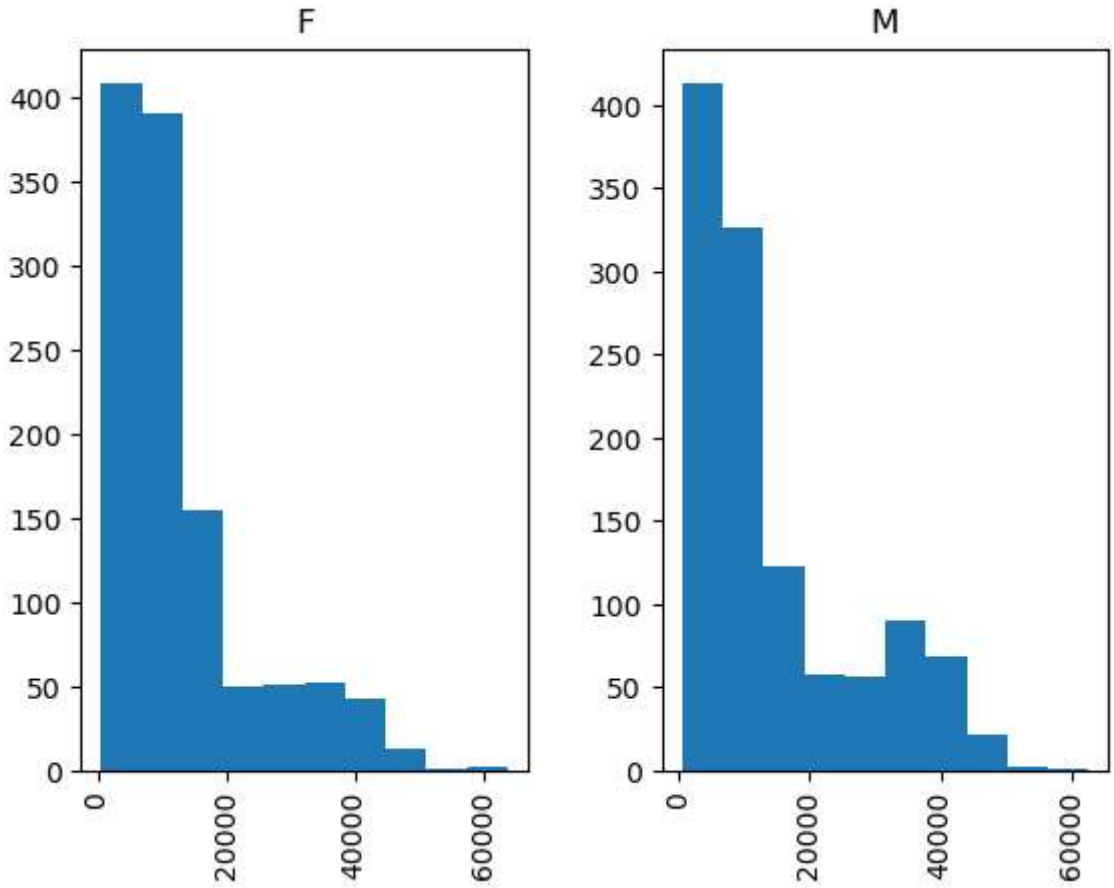
In [41]:
```
HCI.hist(column='charges', by = 'Hospital tier')
```

Out[41]:
```
array([[<AxesSubplot:title={'center':'tier - 1'}>,
        <AxesSubplot:title={'center':'tier - 2'}>],
       [<AxesSubplot:title={'center':'tier - 3'}>, <AxesSubplot:>]],
      dtype=object)
```



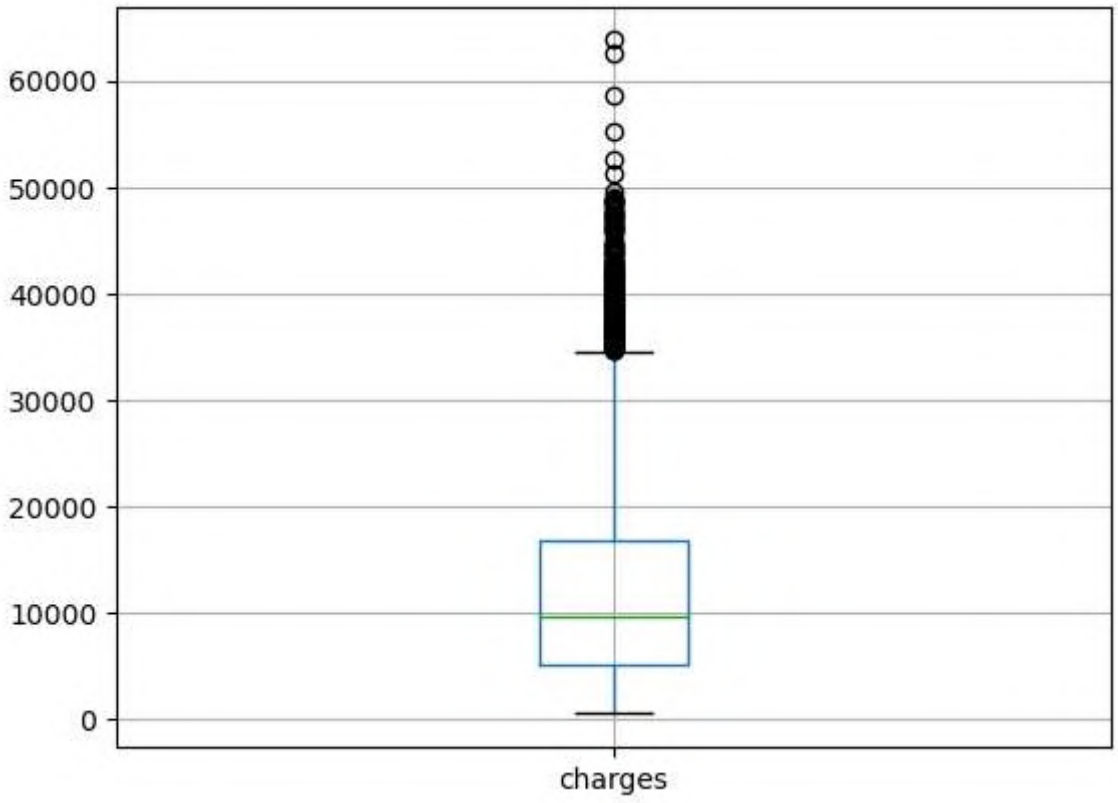
In [42]:
```
HCI.hist(column='charges', by = 'Gender')
```

Out[42]:
```
array([<AxesSubplot:title={'center':'F'}>,
       <AxesSubplot:title={'center':'M'}>], dtype=object)
```



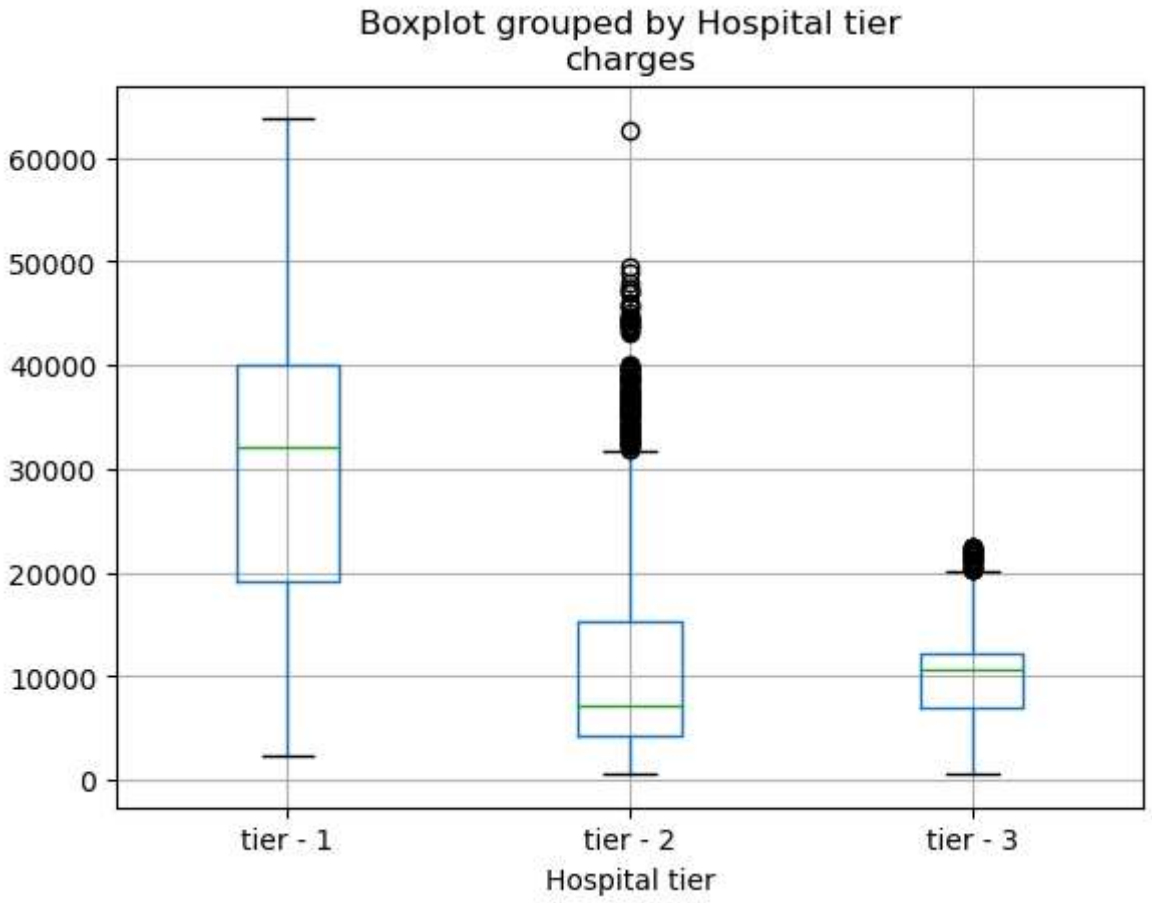
In [43]:
```
HCI.boxplot(column = 'charges')
```

Out[43]:
```
<AxesSubplot:>
```



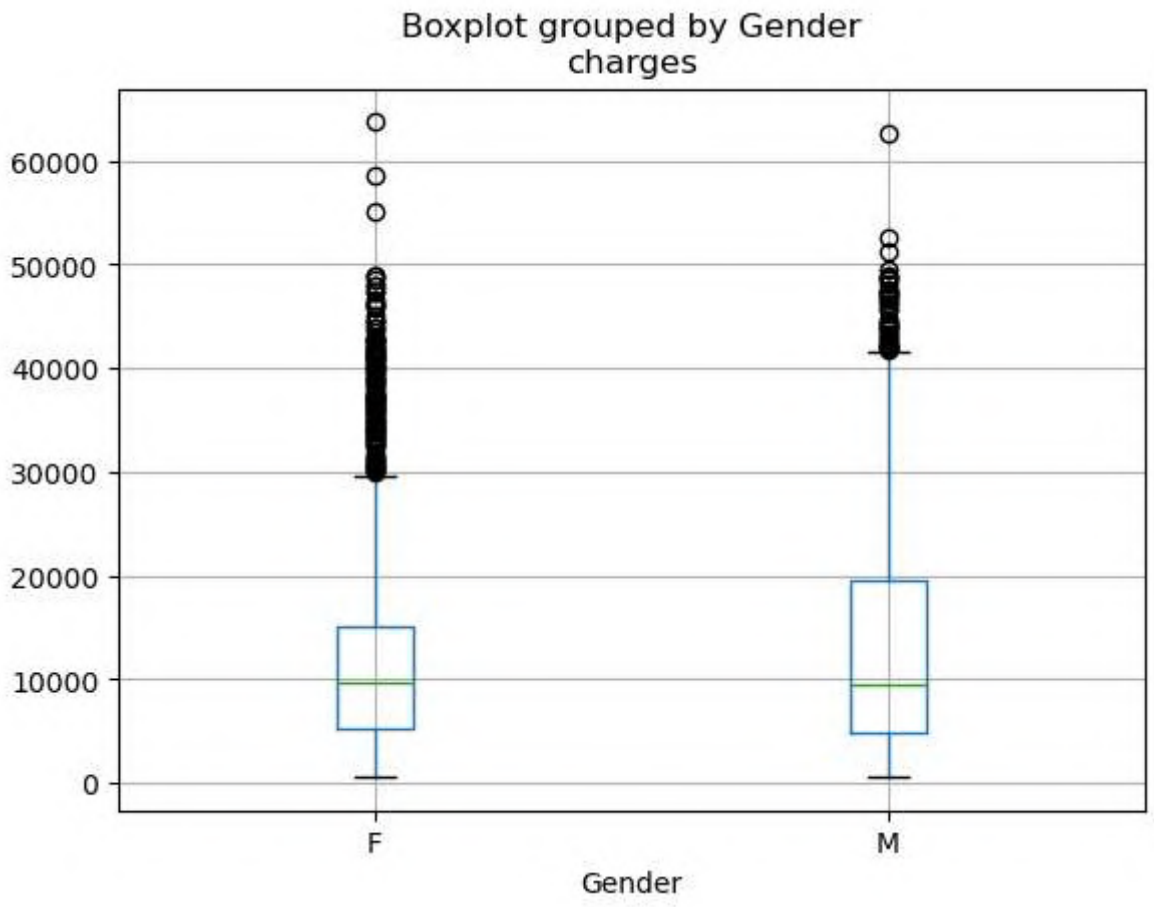
In [44]:
```
HCI.boxplot(column='charges', by = 'Hospital tier')
```

Out[44]:
```
<AxesSubplot:title={'center':'charges'}, xlabel='Hospital tier'>
```



In [45]:
```
HCI.boxplot(column='charges', by = 'Gender')
```

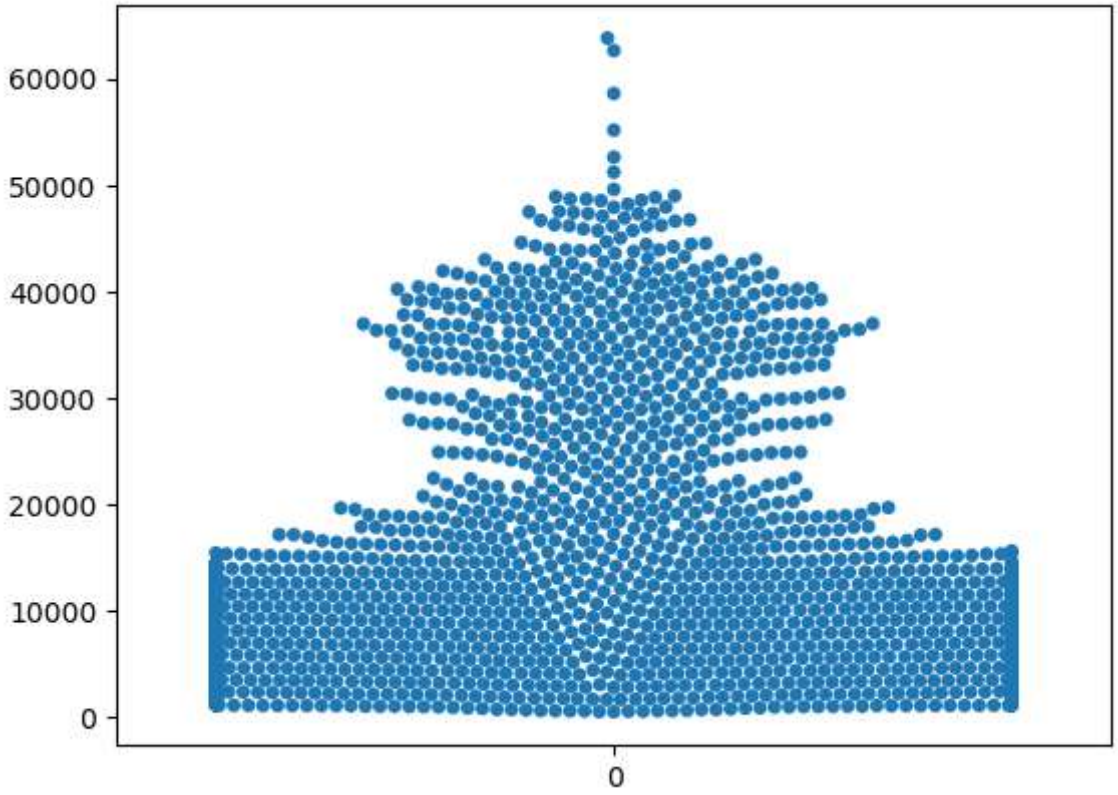
Out[45]:
```
<AxesSubplot:title={'center':'charges'}, xlabel='Gender'>
```



In [46]:
```
sns.swarmplot(data = HCI['charges'])
```

```
C:\Users\naray\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning:
42.6% of the points cannot be placed; you may want to decrease the size of the marker
s or use stripplot.
  warnings.warn(msg, UserWarning)
```

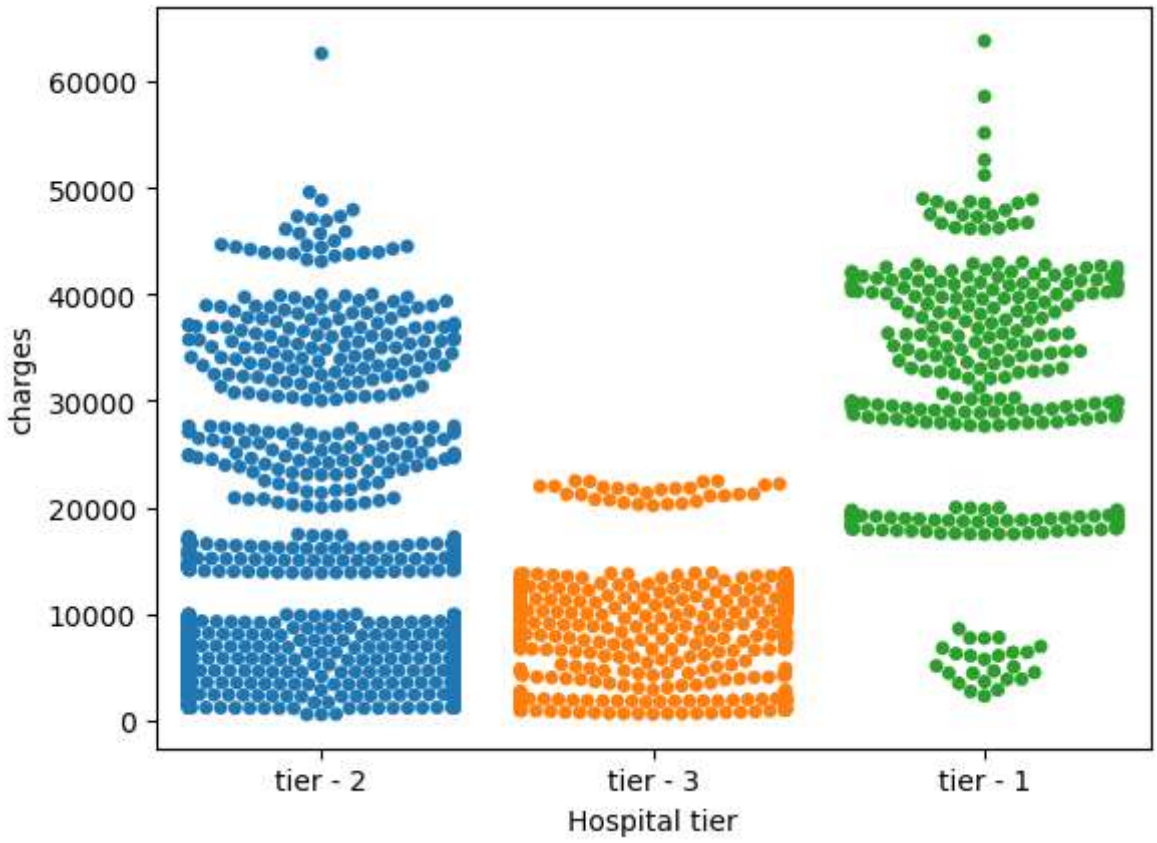
Out[46]:
```
<AxesSubplot:>
```



In [47]:
```
sns.swarmplot(data = HCI, x = 'Hospital tier', y = 'charges')
```

```
C:\Users\naray\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning:
66.3% of the points cannot be placed; you may want to decrease the size of the marker
s or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\naray\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning:
67.9% of the points cannot be placed; you may want to decrease the size of the marker
s or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\naray\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning:
16.7% of the points cannot be placed; you may want to decrease the size of the marker
s or use stripplot.
  warnings.warn(msg, UserWarning)
```

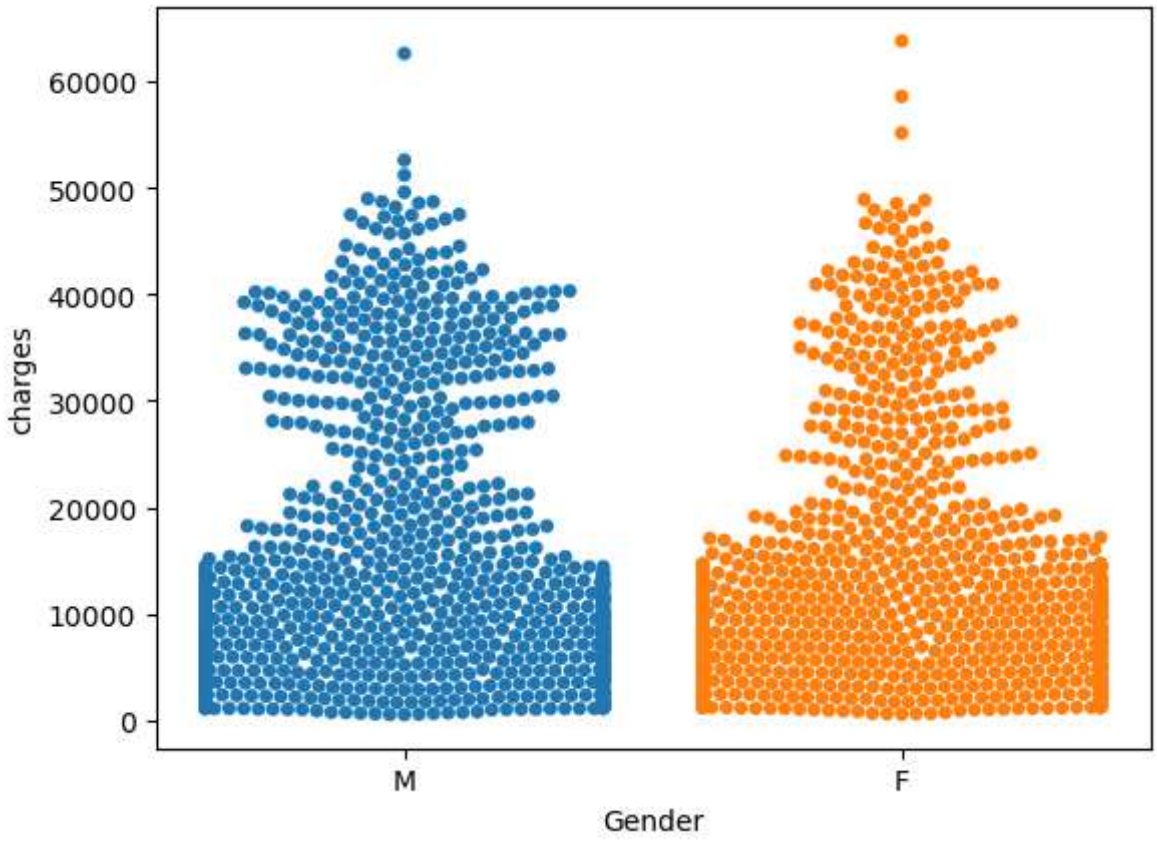
Out[47]:
```
<AxesSubplot:xlabel='Hospital tier', ylabel='charges'>
```



In [48]:
```
sns.swarmplot(data = HCI, x = 'Gender', y = 'charges')
```

```
C:\Users\naray\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning:
40.3% of the points cannot be placed; you may want to decrease the size of the marker
s or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\naray\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning:
46.3% of the points cannot be placed; you may want to decrease the size of the marker
s or use stripplot.
  warnings.warn(msg, UserWarning)
```

Out[48]:
```
<AxesSubplot:xlabel='Gender', ylabel='charges'>
```



In [49]:
```
px.line_polar(HCI, r='charges', theta='Hospital tier', line_close=True)
```

```
C:\Users\naray\anaconda3\lib\site-packages\plotly\express\_core.py:271: FutureWarnin
g: The frame.append method is deprecated and will be removed from pandas in a future
version. Use pandas.concat instead.
  trace_data = trace_data.append(trace_data.iloc[0])
```

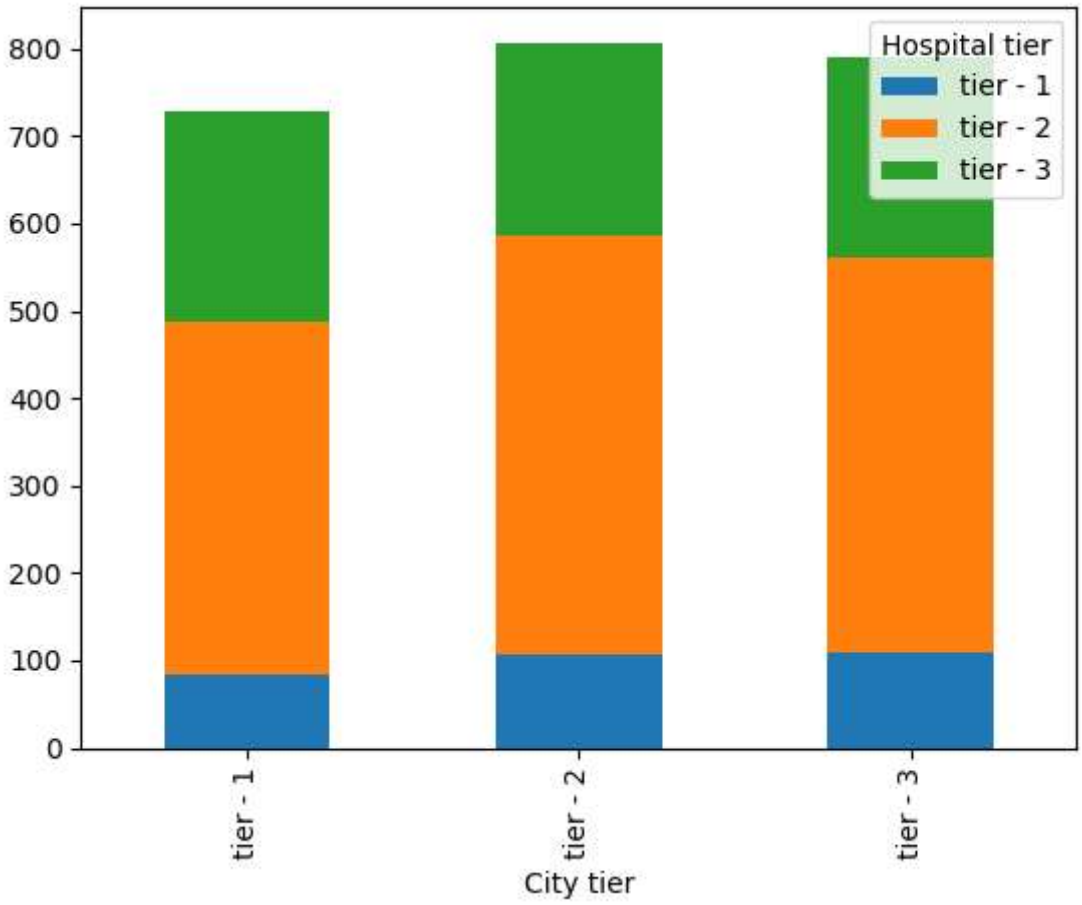
In [50]:
```
pd.crosstab(HCI['City tier'], HCI['Hospital tier'])
```

Out[50]:

City tier	tier - 1	tier - 2	tier - 3
tier - 1	85	403	241
tier - 2	106	479	222
tier - 3	109	452	228

In [51]:
```
HCI.groupby(['City tier', 'Hospital tier']).size().unstack().plot(kind='bar', stacked=
```

Out[51]:
```
<AxesSubplot:xlabel='City tier'>
```



Null Hypothesis: The average hospitalization costs for the three types of hospitals are not significantly different

In [52]:
```
H_tier_1 = HCI[HCI['Hospital tier'] == 'tier - 1']
H_tier_1.head()
```

Out[52]:

	Customer ID	children	charges	Hospital tier	City tier	State ID	BMI	HBA1C	Heart Issues	Any Transplants	Cancer history
238	Id2097	0	2302.30	tier - 1	tier - 2	R1011	20.80	4.87	yes	No	No
284	Id2051	0	2721.32	tier - 1	tier - 3	R1014	26.22	4.65	yes	No	Yes
302	Id2033	0	2867.12	tier - 1	tier - 3	R1013	27.94	6.08	No	No	Yes
369	Id1966	1	3490.55	tier - 1	tier - 2	R1015	32.49	4.00	yes	No	No
389	Id1946	0	3645.09	tier - 1	tier - 2	R1014	25.46	4.14	No	No	No

In [53]:
```
H_tier_2 = HCI[HCI['Hospital tier'] == 'tier - 2']
H_tier_2.head()
```

Out[53]:

	Customer ID	children	charges	Hospital tier	City tier	State ID	BMI	HBA1C	Heart Issues	Any Transplants	Cancer history	I
0	Id2335	0	563.84	tier - 2	tier - 3	R1013	17.58	4.51	No	No	No	
1	Id2334	0	570.62	tier - 2	tier - 1	R1013	17.60	4.39	No	No	No	
2	Id2333	0	600.00	tier - 2	tier - 1	R1013	16.47	6.35	No	No	Yes	
50	Id2285	0	1146.80	tier - 2	tier - 2	R1013	41.14	6.02	No	yes	No	
51	Id2284	0	1149.00	tier - 2	tier - 1	R1012	15.01	4.15	No	No	Yes	

In [54]:
```
H_tier_3 = HCI[HCI['Hospital tier'] == 'tier - 3']
H_tier_3.head()
```

Out[54]:

	Customer ID	children	charges	Hospital tier	City tier	State ID	BMI	HBA1C	Heart Issues	Any Transplants	Cancer history	N
3	Id2332	0	604.54	tier - 3	tier - 3	R1013	17.70	6.28	No	No	No	
4	Id2331	0	637.26	tier - 3	tier - 3	R1013	22.34	5.57	No	No	No	
5	Id2330	0	646.14	tier - 3	tier - 3	R1012	22.24	4.29	yes	No	No	
6	Id2329	0	650.00	tier - 3	tier - 3	R1013	17.07	5.22	No	No	Yes	
7	Id2328	0	650.00	tier - 3	tier - 3	R1013	17.82	5.26	yes	No	No	

In [55]:
```
f_test, p_val = scipy.stats.f_oneway(H_tier_1['charges'], H_tier_2['charges'], H_tier_
print("p-value is: ", p_val)
if p_val < 0.05:
    print(" We can reject the null hypothesis")
else:
    print("We can accept the null hypothesis")
```

```
p-value is:  1.7738221310852664e-179
 We can reject the null hypothesis
```

In [56]:
```
H_tier_1['charges'].mean()
```

Out[56]:
```
30131.995900000005
```

In [57]:
```
H_tier_2['charges'].mean()
```

Out[57]:
```
11875.88386056971
```

In [58]:
```
H_tier_3['charges'].mean()
```

Out[58]:
```
9487.456222865401
```

Null Hypothesis: The average hospitalization costs for the three types of cities are not significantly different

In [59]:
```
C_tier_1 = HCI[HCI['City tier'] == 'tier - 1']
C_tier_1.head()
```

Out[59]:

	Customer ID	children	charges	Hospital tier	City tier	State ID	BMI	HBA1C	Heart Issues	Any Transplants	Cancer history	I
1	Id2334	0	570.62	tier - 2	tier - 1	R1013	17.60	4.39	No	No	No	
2	Id2333	0	600.00	tier - 2	tier - 1	R1013	16.47	6.35	No	No	Yes	
12	Id2323	0	722.99	tier - 3	tier - 1	R1013	23.35	5.94	No	No	No	
14	Id2321	0	760.00	tier - 3	tier - 1	R1013	17.86	5.43	No	No	Yes	
20	Id2315	0	865.41	tier - 3	tier - 1	R1013	24.14	5.29	No	yes	No	

In [60]:
```
C_tier_2 = HCI[HCI['City tier'] == 'tier - 2']
C_tier_2.head()
```

Out[60]:

	Customer ID	children	charges	Hospital tier	City tier	State ID	BMI	HBA1C	Heart Issues	Any Transplants	Cancer history	I
8	Id2327	0	668.00	tier - 3	tier - 2	R1012	21.77	10.67	No	No	No	
10	Id2325	0	687.54	tier - 3	tier - 2	R1013	24.76	4.54	yes	No	No	
18	Id2317	0	773.54	tier - 3	tier - 2	R1013	20.47	5.81	yes	No	No	
19	Id2316	0	830.52	tier - 3	tier - 2	R1011	25.03	5.91	No	yes	No	
24	Id2311	0	964.71	tier - 3	tier - 2	R1013	25.19	5.64	yes	No	No	

In [61]:
```
C_tier_3 = HCI[HCI['City tier'] == 'tier - 3']
C_tier_3.head()
```

Out[61]:

	Customer ID	children	charges	Hospital tier	City tier	State ID	BMI	HBA1C	Heart Issues	Any Transplants	Cancer history	N
0	Id2335	0	563.84	tier - 2	tier - 3	R1013	17.58	4.51	No	No	No	
3	Id2332	0	604.54	tier - 3	tier - 3	R1013	17.70	6.28	No	No	No	
4	Id2331	0	637.26	tier - 3	tier - 3	R1013	22.34	5.57	No	No	No	
5	Id2330	0	646.14	tier - 3	tier - 3	R1012	22.24	4.29	yes	No	No	
6	Id2329	0	650.00	tier - 3	tier - 3	R1013	17.07	5.22	No	No	Yes	

In [62]:
```
f_test, p_val = scipy.stats.f_oneway(C_tier_1['charges'], C_tier_2['charges'], C_tier_
print("p-value is: ", p_val)
if p_val < 0.05:
    print(" We can reject the null hypothesis")
else:
    print("We can accept the null hypothesis")
```

```
p-value is:  0.23376344386881315
We can accept the null hypothesis
```

In [63]:
```
C_tier_1['charges'].mean()
```

Out[63]:
```
13009.97257887517
```

In [64]:
```
C_tier_2['charges'].mean()
```

Out[64]:
```
13471.919281288725
```

In [65]:
```
C_tier_3['charges'].mean()
```

Out[65]:
```
14045.312065906202
```

Null Hypothesis: The average hospitalization cost for smokers is not significantly different from the average cost for nonsmokers

In [66]:
```
smokers = HCI[HCI['smoker'] == 'yes']
smokers.head()
```

Out[66]:

	Customer ID	children	charges	Hospital tier	City tier	State ID	BMI	HBA1C	Heart Issues	Any Transplants	Canc histo
1511	Id824	2	12829.46	tier - 3	tier - 3	R1016	17.290	4.62	No	yes	N
1593	Id742	0	13747.87	tier - 3	tier - 2	R1018	21.565	4.95	No	yes	N
1599	Id736	0	13844.51	tier - 2	tier - 1	R1011	21.700	4.90	No	No	Y
1628	Id707	0	14283.46	tier - 2	tier - 2	R1024	21.660	4.37	No	yes	N
1646	Id689	2	14455.64	tier - 2	tier - 3	R1024	17.195	5.29	yes	No	N

In [67]:
```
non_smokers = HCI[HCI['smoker'] == 'No']
non_smokers.head()
```

Out[67]:

	Customer ID	children	charges	Hospital tier	City tier	State ID	BMI	HBA1C	Heart Issues	Any Transplants	Cancer history	N
0	Id2335	0	563.84	tier - 2	tier - 3	R1013	17.58	4.51	No	No	No	
1	Id2334	0	570.62	tier - 2	tier - 1	R1013	17.60	4.39	No	No	No	
2	Id2333	0	600.00	tier - 2	tier - 1	R1013	16.47	6.35	No	No	Yes	
3	Id2332	0	604.54	tier - 3	tier - 3	R1013	17.70	6.28	No	No	No	
4	Id2331	0	637.26	tier - 3	tier - 3	R1013	22.34	5.57	No	No	No	

In [68]:
```
t_test, p_val =  ttest_ind(smokers['charges'], non_smokers['charges'])
print("p-value is: ", p_val)
if p_val < 0.05:
    print(" We can reject the null hypothesis")
else:
    print("We can accept the null hypothesis")
```

```
p-value is:  0.0
 We can reject the null hypothesis
```

In [69]:
```
smokers['charges'].mean()
```

Out[69]:
```
32866.960226337425
```

In [70]:
```
non_smokers['charges'].mean()
```

Out[70]:
```
8409.199249592164
```

Null Hypothesis = Smoking and heart issues are independent

In [71]:
```
data = HCI[['Heart Issues', 'Any Transplants', 'Cancer history', 'smoker', 'Gender']]
data.head(10)
```

Out[71]:

	Heart Issues	Any Transplants	Cancer history	smoker	Gender
0	No	No	No	No	M
1	No	No	No	No	M
2	No	No	Yes	No	F
3	No	No	No	No	M
4	No	No	No	No	M
5	yes	No	No	No	F
6	No	No	Yes	No	F
7	yes	No	No	No	F
8	No	No	No	No	F
9	yes	No	Yes	No	M

In [72]:
```
for col in data.columns:
    le = LabelEncoder()
    data[col] = le.fit_transform(data[col])
data.head(10)
```

```
C:\Users\naray\AppData\Local\Temp\ipykernel_15296\3504758527.py:3: SettingWithCopyWar
ning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
```

Out[72]:

	Heart Issues	Any Transplants	Cancer history	smoker	Gender
0	0	0	0	0	1
1	0	0	0	0	1
2	0	0	1	0	0
3	0	0	0	0	1
4	0	0	0	0	1
5	1	0	0	0	0
6	0	0	1	0	0
7	1	0	0	0	0
8	0	0	0	0	0
9	1	0	1	0	1

In [73]:
```
X = data.drop(columns = ['Heart Issues', 'Any Transplants', 'Cancer history', 'Gender'
y = data['Heart Issues']
```

In [74]:
```
chi_scores, p_value = chi2(X, y)
chi_scores
```

Out[74]:
```
array([0.0942577])
```

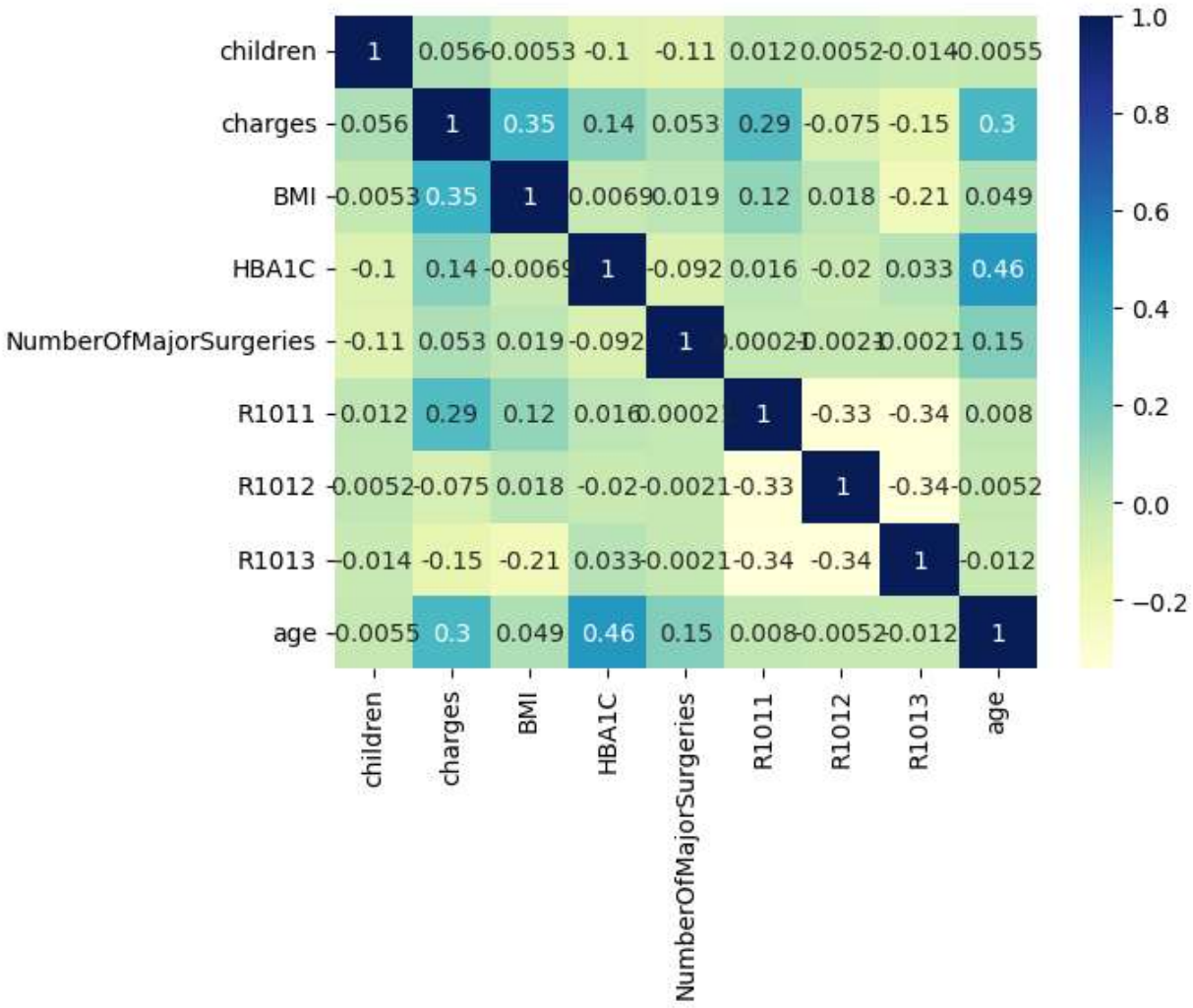
In [75]:
```
print("p-value is: ", p_value)
```

```
p-value is:  [0.75883259]
```

In [76]:
```
alpha = 0.05
if p_value <= alpha:
    print('We can reject the null hypothesis')
else:
    print('We can accept the null hypothesis')
```

```
We can accept the null hypothesis
```

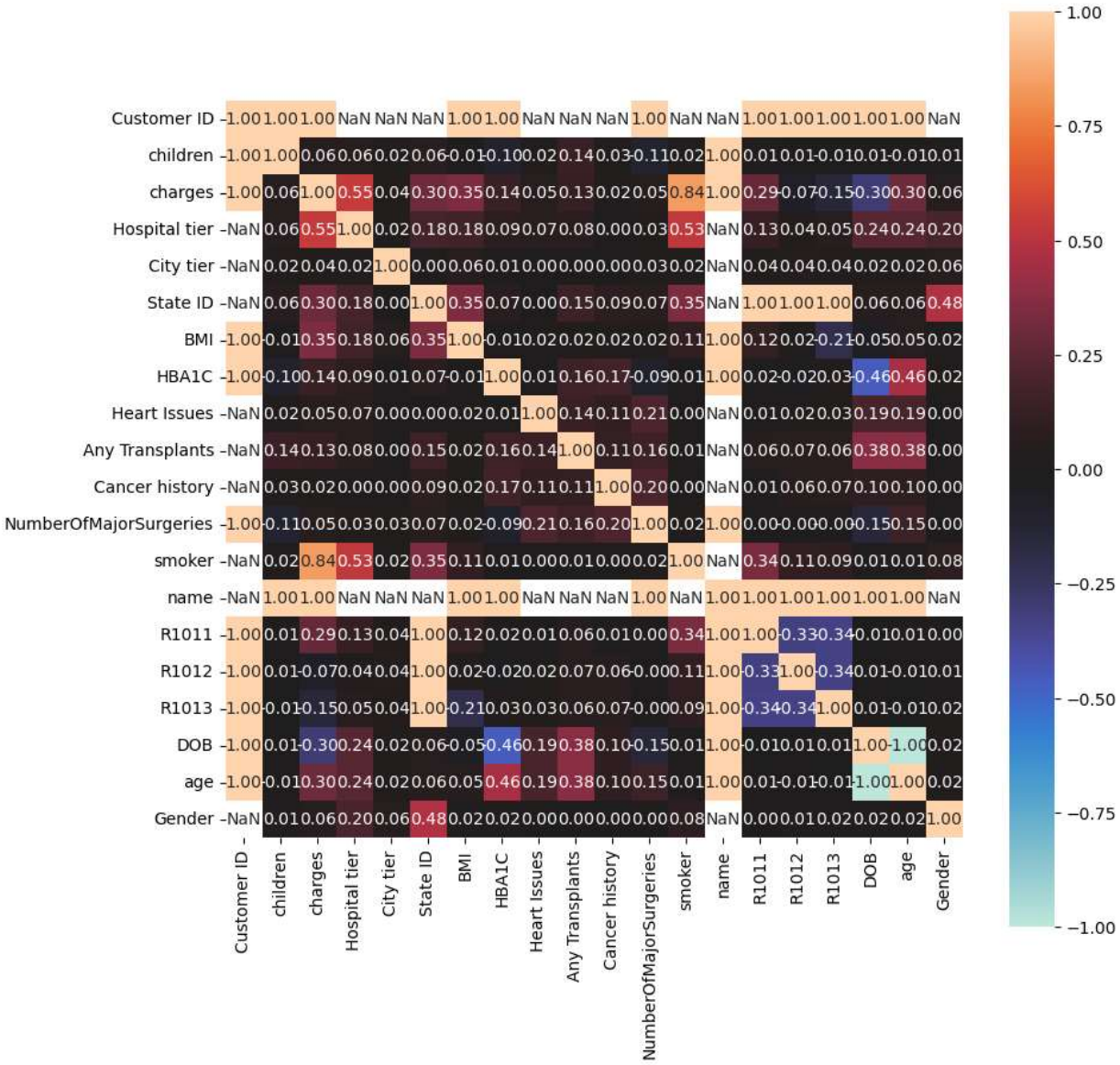
In [77]:
```
dataplot = sns.heatmap(HCI.corr(), cmap="YlGnBu", annot=True)
plt.show()
```



In [78]:
```
complete_correlation= associations(HCI, filename= 'complete_correlation.png', figsize=
```

```
C:\Users\naray\anaconda3\lib\site-packages\dython\nominal.py:137: RuntimeWarning:

Unable to calculate Cramer's V using bias correction. Consider using bias_correction=
False

C:\Users\naray\anaconda3\lib\site-packages\dython\nominal.py:137: RuntimeWarning:

Unable to calculate Cramer's V using bias correction. Consider using bias_correction=
False

C:\Users\naray\anaconda3\lib\site-packages\dython\nominal.py:137: RuntimeWarning:

Unable to calculate Cramer's V using bias correction. Consider using bias_correction=
False

C:\Users\naray\anaconda3\lib\site-packages\dython\nominal.py:137: RuntimeWarning:

Unable to calculate Cramer's V using bias correction. Consider using bias_correction=
False

C:\Users\naray\anaconda3\lib\site-packages\dython\nominal.py:137: RuntimeWarning:

Unable to calculate Cramer's V using bias correction. Consider using bias_correction=
False

C:\Users\naray\anaconda3\lib\site-packages\dython\nominal.py:137: RuntimeWarning:

Unable to calculate Cramer's V using bias correction. Consider using bias_correction=
False

C:\Users\naray\anaconda3\lib\site-packages\dython\nominal.py:137: RuntimeWarning:

Unable to calculate Cramer's V using bias correction. Consider using bias_correction=
False

C:\Users\naray\anaconda3\lib\site-packages\dython\nominal.py:137: RuntimeWarning:

Unable to calculate Cramer's V using bias correction. Consider using bias_correction=
False

C:\Users\naray\anaconda3\lib\site-packages\dython\nominal.py:137: RuntimeWarning:

Unable to calculate Cramer's V using bias correction. Consider using bias_correction=
False

C:\Users\naray\anaconda3\lib\site-packages\dython\nominal.py:137: RuntimeWarning:

Unable to calculate Cramer's V using bias correction. Consider using bias_correction=
False

C:\Users\naray\anaconda3\lib\site-packages\dython\nominal.py:137: RuntimeWarning:

Unable to calculate Cramer's V using bias correction. Consider using bias_correction=
False

C:\Users\naray\anaconda3\lib\site-packages\dython\nominal.py:137: RuntimeWarning:

Unable to calculate Cramer's V using bias correction. Consider using bias_correction=
False
```

```
C:\Users\naray\anaconda3\lib\site-packages\dython\nominal.py:137: RuntimeWarning:

Unable to calculate Cramer's V using bias correction. Consider using bias_correction=
False

C:\Users\naray\anaconda3\lib\site-packages\dython\nominal.py:137: RuntimeWarning:

Unable to calculate Cramer's V using bias correction. Consider using bias_correction=
False

C:\Users\naray\anaconda3\lib\site-packages\dython\nominal.py:137: RuntimeWarning:

Unable to calculate Cramer's V using bias correction. Consider using bias_correction=
False

C:\Users\naray\anaconda3\lib\site-packages\dython\nominal.py:137: RuntimeWarning:

Unable to calculate Cramer's V using bias correction. Consider using bias_correction=
False

C:\Users\naray\anaconda3\lib\site-packages\dython\nominal.py:137: RuntimeWarning:

Unable to calculate Cramer's V using bias correction. Consider using bias_correction=
False
```



In [79]:
```
Hospital_dummy = pd.get_dummies(HealthCare['Hospital tier']).iloc[:,0:3]
Hospital_dummy
```

Out[79]:

	tier - 1	tier - 2	tier - 3
0	0	1	0
1	0	1	0
2	0	1	0
3	0	0	1
4	0	0	1
...	...	...	...
2329	1	0	0
2330	1	0	0
2331	1	0	0
2333	0	1	0
2334	1	0	0

2325 rows × 3 columns

In [80]:
```
Hospital_dummy.rename(columns = {'tier - 1':'H_tier - 1', 'tier - 2':'H_tier - 2',
                                 'tier - 3':'H_tier - 3'}, inplace = True)
Hospital_dummy.head()
```

Out[80]:

	H_tier - 1	H_tier - 2	H_tier - 3
0	0	1	0
1	0	1	0
2	0	1	0
3	0	0	1
4	0	0	1

In [81]:
```
City_dummy = pd.get_dummies(HealthCare['City tier']).iloc[:,0:3]
City_dummy
```

Out[81]:

	tier - 1	tier - 2	tier - 3
0	0	0	1
1	1	0	0
2	1	0	0
3	0	0	1
4	0	0	1
...	...	...	...
2329	0	0	1
2330	0	1	0
2331	0	0	1
2333	0	0	1
2334	0	0	1

2325 rows × 3 columns

In [82]:
```
City_dummy.rename(columns = {'tier - 1':'C_tier - 1', 'tier - 2':'C_tier - 2',
                             'tier - 3':'C_tier - 3'}, inplace = True)
City_dummy.head()
```

Out[82]:

	C_tier - 1	C_tier - 2	C_tier - 3
0	0	0	1
1	1	0	0
2	1	0	0
3	0	0	1
4	0	0	1

In [83]:
```
Data = pd.concat([data, Hospital_dummy, City_dummy], axis = 1)
Data.head()
```

Out[83]:

	Heart Issues	Any Transplants	Cancer history	smoker	Gender	H_tier - 1	H_tier - 2	H_tier - 3	C_tier - 1	C_tier - 2	C_tier - 3
0	0	0	0	0	1	0	1	0	0	0	1
1	0	0	0	0	1	0	1	0	1	0	0
2	0	0	1	0	0	0	1	0	1	0	0
3	0	0	0	0	1	0	0	1	0	0	1
4	0	0	0	0	1	0	0	1	0	0	1

In [84]:
```
Data2 = HCI[['charges', 'children', 'BMI', 'HBA1C', 'NumberOfMajorSurgeries', 'R1011',
Data2.head()
```

Out[84]:

	charges	children	BMI	HBA1C	NumberOfMajorSurgeries	R1011	R1012	R1013	age
0	563.84	0	17.58	4.51	1	0	0	1	30
1	570.62	0	17.60	4.39	1	0	0	1	30
2	600.00	0	16.47	6.35	1	0	0	1	29
3	604.54	0	17.70	6.28	1	0	0	1	30
4	637.26	0	22.34	5.57	1	0	0	1	24

In [85]:
```
Health = pd.concat([Data, Data2], axis = 1)
Health.head()
```

Out[85]:

	Heart Issues	Any Transplants	Cancer history	smoker	Gender	H_tier - 1	H_tier - 2	H_tier - 3	C_tier - 1	C_tier - 2	C_tier - 3	charges
0	0	0	0	0	1	0	1	0	0	0	1	563.84
1	0	0	0	0	1	0	1	0	1	0	0	570.62
2	0	0	1	0	0	0	1	0	1	0	0	600.00
3	0	0	0	0	1	0	0	1	0	0	1	604.54
4	0	0	0	0	1	0	0	1	0	0	1	637.26

In [86]:
```
X = Health.drop(columns = ['charges'], axis = 1)
y = Health['charges']
```

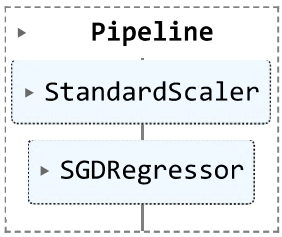
In [87]:
```
model = make_pipeline(StandardScaler(), SGDRegressor())
```

In [88]:
```
fold_scores = []
variable_importance = {}
```

In [89]:
```
kf = KFold(n_splits=5, shuffle=True, random_state=42)
for fold, (train_index, val_index) in enumerate(kf.split(X, y)):
    # Split the data into training and validation sets
    X_train, X_val = X.iloc[train_index], X.iloc[val_index]
    y_train, y_val = y.iloc[train_index], y.iloc[val_index]
```

In [90]:
```
model.fit(X_train, y_train)
```

Out[90]:



In [91]:
```
y_pred = model.predict(X_val)
rmse = mean_squared_error(y_val, y_pred, squared=False)
fold_scores.append(rmse)
```

In [92]:
```
coef = model.named_steps['sgdregressor'].coef_
variable_importance[fold] = {feature: abs(coef[i]) for i, feature in enumerate(X.colur
```

In [93]:
```
average_score = sum(fold_scores) / len(fold_scores)
```

In [94]:
```
print("Validation Scores (RMSE) for each fold:")
for fold, score in enumerate(fold_scores):
    print(f"Fold {fold+1}: {score}")

print("Average Validation Score (RMSE):", average_score)

print("Variable Importance Scores:")
for fold, importance in variable_importance.items():
    print(f"Fold {fold+1}:")
    for feature, score in importance.items():
        print(f"{feature}: {score}")
```

```
Validation Scores (RMSE) for each fold:
Fold 1: 4454.707586346708
Average Validation Score (RMSE): 4454.707586346708
Variable Importance Scores:
Fold 5:
Heart Issues: 44.26380750248429
Any Transplants: 119.11901674066364
Cancer history: 70.71944791110565
smoker: 8654.580843205544
Gender: 65.37153680386015
H_tier - 1: 1143.8302806339857
H_tier - 2: 193.99128520917512
H_tier - 3: 617.4014952234389
C_tier - 1: 10.042242698895304
C_tier - 2: 13.325823788635011
C_tier - 3: 23.267325681358983
children: 477.6373388851194
BMI: 2645.6206052363577
HBA1C: 343.2997405589134
NumberOfMajorSurgeries: 50.433004744674164
R1011: 349.8295089867437
R1012: 247.49963237953938
R1013: 531.400763806876
age: 3371.617100113446
```

In [95]:
```
X = Health.drop(columns = ['charges'], axis = 1)
y = Health['charges']
```

In [96]:
```
rf_model = RandomForestRegressor(random_state=42)
rf_scores = cross_val_score(rf_model, X, y, cv=5, scoring='neg_mean_squared_error')
rf_rmse_scores = (-rf_scores) ** 0.5
```

In [97]:
```
xgb_model = XGBRegressor(random_state=42)
xgb_scores = cross_val_score(xgb_model, X, y, cv=5, scoring='neg_mean_squared_error')
xgb_rmse_scores = (-xgb_scores) ** 0.5
```

In [98]:
```
rf_model.fit(X, y)
rf_feature_importance = pd.Series(rf_model.feature_importances_, index=X.columns).sort
```

In [99]:
```
xgb_model.fit(X, y)
xgb_feature_importance = pd.Series(xgb_model.feature_importances_, index=X.columns).sc
```

In [100…
```
print("Random Forest Cross-Validation Results (RMSE):")
for fold, score in enumerate(rf_rmse_scores):
    print(f"Fold {fold+1}: {score}")

print("XGBoost Cross-Validation Results (RMSE):")
for fold, score in enumerate(xgb_rmse_scores):
    print(f"Fold {fold+1}: {score}")
```

```
Random Forest Cross-Validation Results (RMSE):
Fold 1: 9861.23548143323
Fold 2: 4381.892781761607
Fold 3: 6401.4408518550035
Fold 4: 7935.139986137964
Fold 5: 18802.749808690798
XGBoost Cross-Validation Results (RMSE):
Fold 1: 9993.904692659493
Fold 2: 4400.279929435713
Fold 3: 5412.401886463171
Fold 4: 8195.021281578605
Fold 5: 18949.853489412002
```

In [101…
```
print("Random Forest Variable Importance:")
print(rf_feature_importance)

print("XGBoost Variable Importance:")
print(xgb_feature_importance)
```

```
Random Forest Variable Importance:
smoker                    0.700320
BMI                       0.126851
age                       0.094192
children                  0.015288
HBA1C                     0.014439
H_tier - 1                0.013009
R1011                     0.008030
R1013                     0.006258
H_tier - 3                0.005165
H_tier - 2                0.004534
Gender                    0.002371
C_tier - 2                0.001735
R1012                     0.001623
Cancer history            0.001319
NumberOfMajorSurgeries    0.001212
C_tier - 1                0.001148
C_tier - 3                0.001144
Heart Issues              0.001131
Any Transplants           0.000233
dtype: float64
XGBoost Variable Importance:
smoker                    0.868453
age                       0.027204
H_tier - 1                0.022359
BMI                       0.016588
R1013                     0.011123
R1011                     0.009623
H_tier - 2                0.009416
children                  0.006912
H_tier - 3                0.005487
R1012                     0.003321
Gender                    0.003170
HBA1C                     0.002910
Any Transplants           0.002594
NumberOfMajorSurgeries    0.002528
C_tier - 2                0.002118
C_tier - 3                0.001857
Cancer history            0.001774
C_tier - 1                0.001460
Heart Issues              0.001102
dtype: float32
```

In [102…

```
DATA = {
    'Heart Issues' : [0],
    'Any Transplants': [0],
    'Cancer history' : [1],
    'smoker': [1],
    'Gender' : [0],
    'H_tier - 1' : [1],
    'H_tier - 2' : [0],
    'H_tier - 3' : [0],
    'C_tier - 1' : [1],
    'C_tier - 2' : [0],
    'C_tier - 3' : [0],
    'children' : [2],
    'BMI' : [29.4],
    'HBA1C': [5.8],
    'NumberOfMajorSurgeries': [0],
    'R1011' : [1],
```

```
    'R1012' : [0],
    'R1013' : [0],
    'age' : [35]

}
input_df = pd.DataFrame(DATA)
```

In [103…
```
rf_prediction = rf_model.predict(input_df)
xgb_prediction = xgb_model.predict(input_df)
model_prediction = model.predict(input_df)
```

In [104…
```
model_prediction
```

Out[104]:
```
array([32389.61222781])
```

In [105…
```
xgb_prediction
```

Out[105]:
```
array([25640.902], dtype=float32)
```

In [106…
```
rf_prediction
```

Out[106]:
```
array([24089.9756])
```

In [107…
```
mean_prediction = (rf_prediction + xgb_prediction + model_prediction) / 3
```

In [108…
```
print("Estimated Hospitalization Cost: $", round(mean_prediction[0], 2))
```

```
Estimated Hospitalization Cost: $ 27373.5
```

In [ ]: