In [2]:
```
import pandas as pd
import numpy as np
import seaborn as sns
from matplotlib import pyplot as plt
from sklearn import preprocessing


import warnings
warnings.filterwarnings('ignore')
```

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

In [4]:
```
df.shape
#shape of data
```

Out[4]:
```
(1338, 8)
```

In [5]:
```
df.head(10)
```

Out[5]:

| | Unnamed: 0 | age | sex | smoker | region | viral load | severity level | hospitalization charges |
|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 19 | female | yes | southwest | 9.30 | 0 | 42212 |
| **1** | 1 | 18 | male | no | southeast | 11.26 | 1 | 4314 |
| **2** | 2 | 28 | male | no | southeast | 11.00 | 3 | 11124 |
| **3** | 3 | 33 | male | no | northwest | 7.57 | 0 | 54961 |
| **4** | 4 | 32 | male | no | northwest | 9.63 | 0 | 9667 |
| **5** | 5 | 31 | female | no | southeast | 8.58 | 0 | 9392 |
| **6** | 6 | 46 | female | no | southeast | 11.15 | 1 | 20601 |
| **7** | 7 | 37 | female | no | northwest | 9.25 | 3 | 18204 |
| **8** | 8 | 37 | male | no | northeast | 9.94 | 2 | 16016 |
| **9** | 9 | 60 | female | no | northwest | 8.61 | 0 | 72308 |

In [6]:
```
df.info()
#data type of all attributes
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 8 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Unnamed: 0               1338 non-null   int64  
 1   age                      1338 non-null   int64  
 2   sex                      1338 non-null   object 
 3   smoker                   1338 non-null   object 
 4   region                   1338 non-null   object 
 5   viral load               1338 non-null   float64
 6   severity level           1338 non-null   int64  
 7   hospitalization charges  1338 non-null   int64  
dtypes: float64(1), int64(4), object(3)
memory usage: 83.8+ KB
```

In [7]:
```
df['region'].value_counts()
#statistical summary of region can be inferred from this
```

Out[7]:
```
southeast    364
southwest    325
northwest    325
northeast    324
Name: region, dtype: int64
```

In [8]:
```
# conversion of categorical attributes to 'category'

#df =  pd.get_dummies(df, columns=["region"])
#one hot encoding for the region would make it more complex but can be done
```

```
#since sex and smoker are binary we can do the below encoding
df['sex'] = df['sex'].map({'female':1, 'male':0})
df['smoker'] = df['smoker'].map({'yes':1, 'no':0})
```

In [9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 8 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Unnamed: 0               1338 non-null   int64  
 1   age                      1338 non-null   int64  
 2   sex                      1338 non-null   int64  
 3   smoker                   1338 non-null   int64  
 4   region                   1338 non-null   object 
 5   viral load               1338 non-null   float64
 6   severity level           1338 non-null   int64  
 7   hospitalization charges  1338 non-null   int64  
dtypes: float64(1), int64(6), object(1)
memory usage: 83.8+ KB
```

In [10]:
```
df.columns
```

Out[10]:
```
Index(['Unnamed: 0', 'age', 'sex', 'smoker', 'region', 'viral load',
       'severity level', 'hospitalization charges'],
      dtype='object')
```

In [11]:
```
df.rename( columns={'Unnamed: 0':'Index'}, inplace=True )
```

In [12]:
```
#Missing Value Detection
df_nv = df.isna()
```

In [13]:
```
df_nv.value_counts()

#As we can see NO Null values
```

Out[13]:
```
Index  age    sex    smoker  region  viral load  severity level  hospitalization charges
False  False  False  False   False   False       False           False                      1338
dtype: int64
```

In [14]:
```
#statistical summary
df.describe()
```

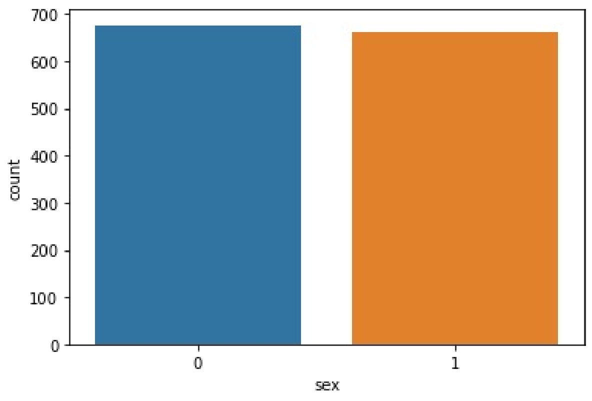
Out[14]:

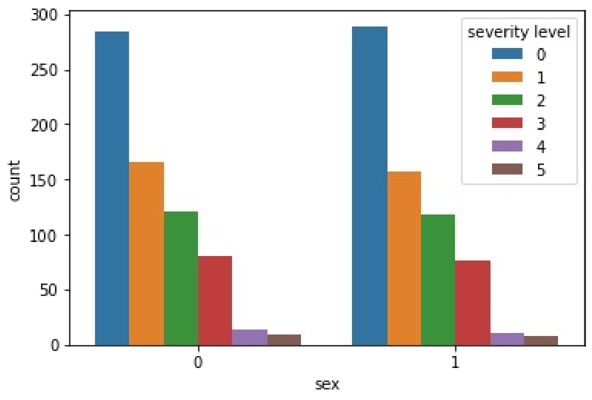
| | Index | age | sex | smoker | viral load | severity level | hospitalization charges |
|---|---|---|---|---|---|---|---|
| **count** | 1338.000000 | 1338.000000 | 1338.000000 | 1338.000000 | 1338.000000 | 1338.000000 | 1338.000000 |
| **mean** | 668.500000 | 39.207025 | 0.494768 | 0.204783 | 10.221233 | 1.094918 | 33176.058296 |
| **std** | 386.391641 | 14.049960 | 0.500160 | 0.403694 | 2.032796 | 1.205493 | 30275.029296 |
| **min** | 0.000000 | 18.000000 | 0.000000 | 0.000000 | 5.320000 | 0.000000 | 2805.000000 |
| **25%** | 334.250000 | 27.000000 | 0.000000 | 0.000000 | 8.762500 | 0.000000 | 11851.000000 |
| **50%** | 668.500000 | 39.000000 | 0.000000 | 0.000000 | 10.130000 | 1.000000 | 23455.000000 |
| **75%** | 1002.750000 | 51.000000 | 1.000000 | 0.000000 | 11.567500 | 2.000000 | 41599.500000 |
| **max** | 1337.000000 | 64.000000 | 1.000000 | 1.000000 | 17.710000 | 5.000000 | 159426.000000 |

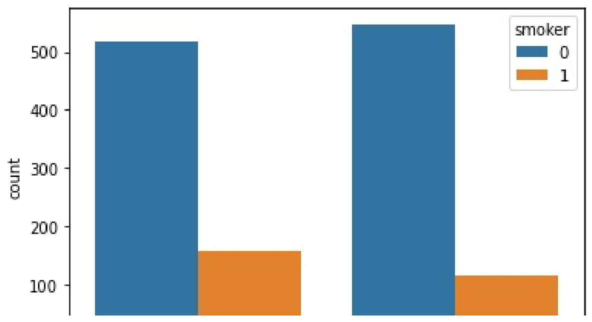
In [15]:
```
df.nunique()
```

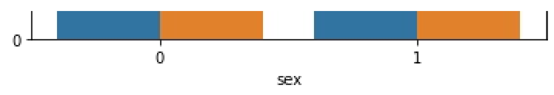
Out[15]:
```
Index                      1338
age                          47
sex                           2
smoker                        2
region                        4
viral load                  462
severity level                6
hospitalization charges    1320
dtype: int64
```

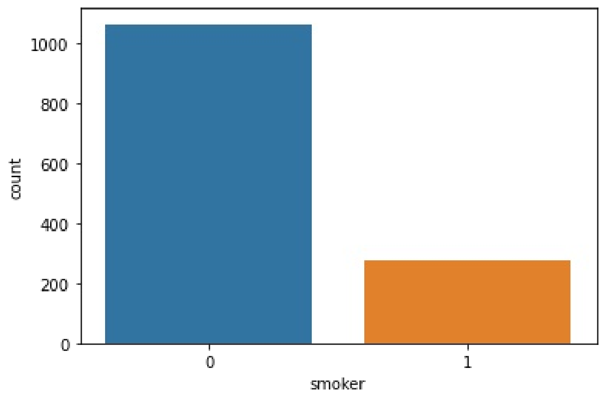
In [16]:
```
df['severity level'].value_counts()
```
Out[16]:
```
0    574
1    324
2    240
3    157
4     25
5     18
Name: severity level, dtype: int64
```

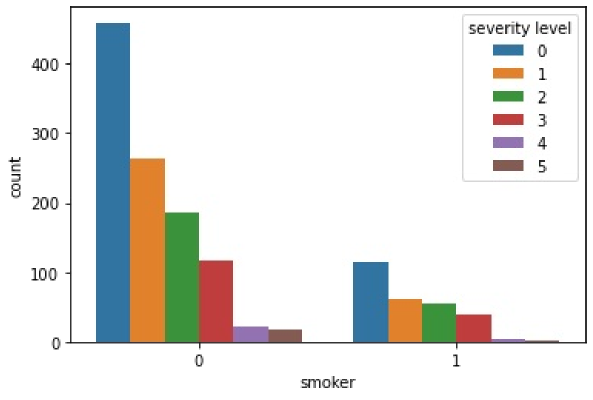
In [17]:
```
#Univariate Analysis

#Categorical valriables are : sex, smoker, region, severity load

#Sex
sns.countplot(x="sex", data=df)
```
Out[17]:
```
<AxesSubplot:xlabel='sex', ylabel='count'>
```


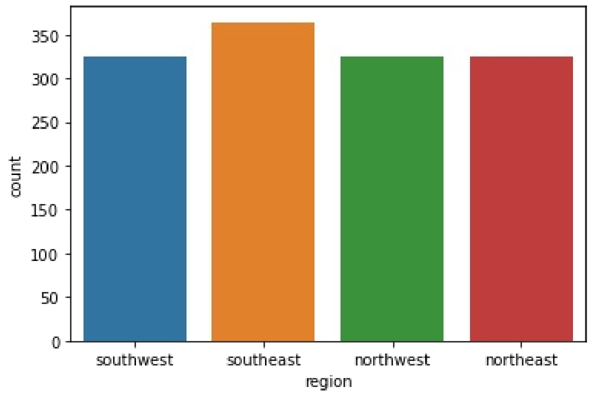
In [18]:
```
sns.countplot(x="sex", hue = "severity level", data=df)
```
Out[18]:
```
<AxesSubplot:xlabel='sex', ylabel='count'>
```


In [19]:
```
sns.countplot(x="sex", hue = "smoker", data=df)
```
Out[19]:
```
<AxesSubplot:xlabel='sex', ylabel='count'>
```


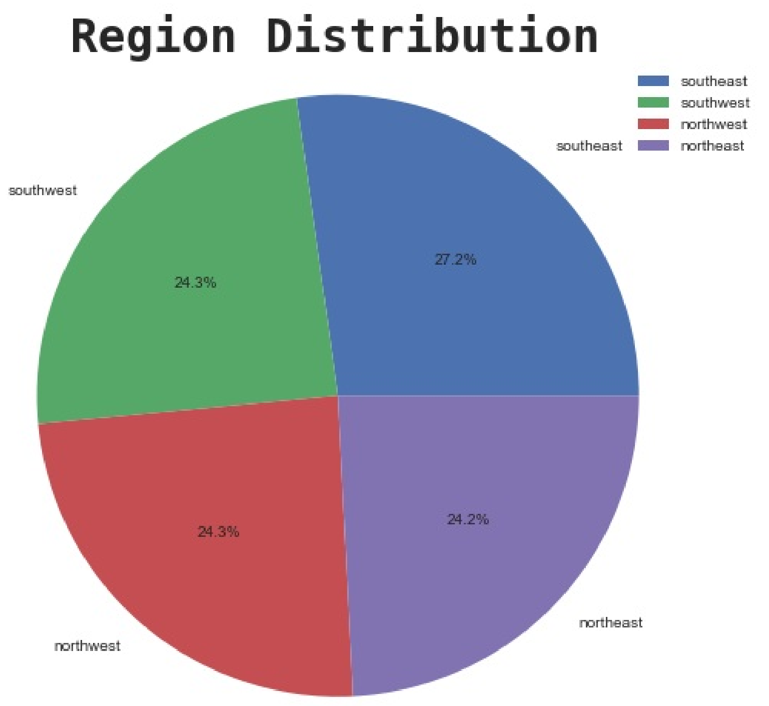


In [20]:
```
#smoker

sns.countplot(x="smoker", data=df)
```
Out[20]:
```
<AxesSubplot:xlabel='smoker', ylabel='count'>
```


In [21]:
```
sns.countplot(x="smoker", hue = "severity level", data=df)
```
Out[21]:
```
<AxesSubplot:xlabel='smoker', ylabel='count'>
```


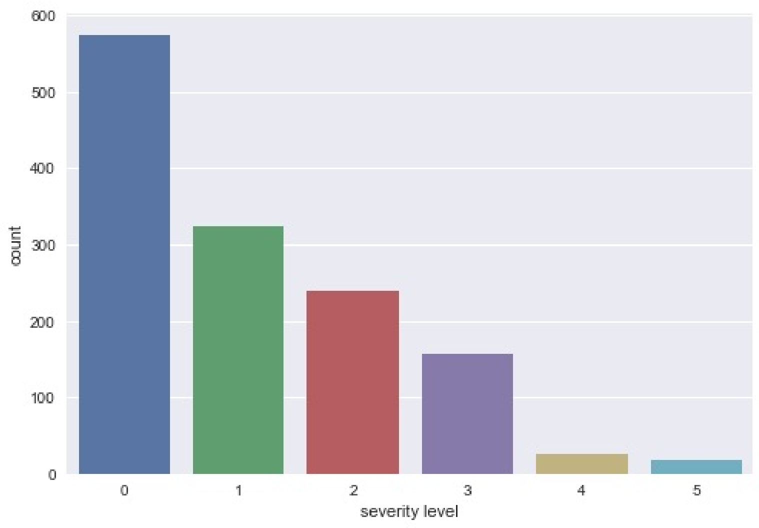
In [22]:
```
#region
sns.countplot(x="region", data=df)
```
Out[22]:
```
<AxesSubplot:xlabel='region', ylabel='count'>
```


In [23]:
```
ssn = df['region'].value_counts()
plt.style.use('seaborn')
plt.figure(figsize = (10, 8))
plt.pie(ssn.values, labels = ssn.index, autopct = '%1.1f%%')
plt.title('Region Distribution', fontdict = {'fontname' : 'Monospace','fontsize' : 30, 'fontweight' : 'bold'})
plt.legend()
plt.axis('equal')
plt.show()
```



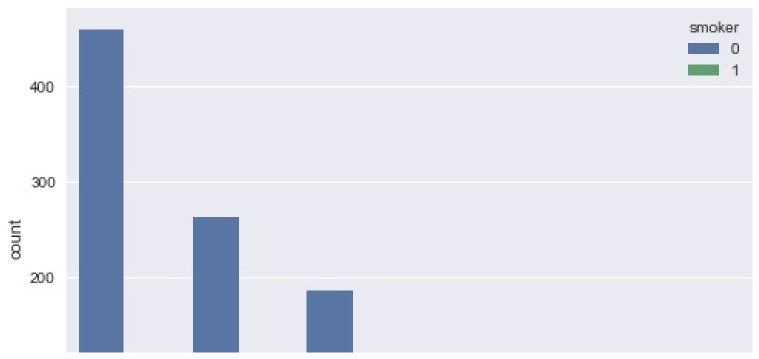
In [24]:
```
#severity load
sns.countplot(x="severity level", data=df)
```

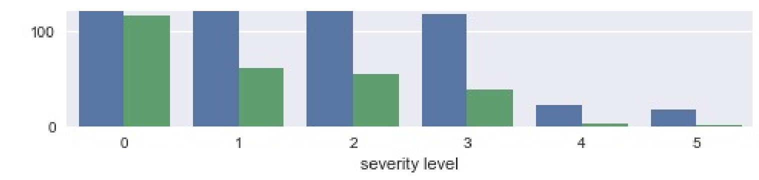
Out[24]:
```
<AxesSubplot:xlabel='severity level', ylabel='count'>
```



In [25]:
```
sns.countplot(x="severity level", hue = "smoker", data=df)
```

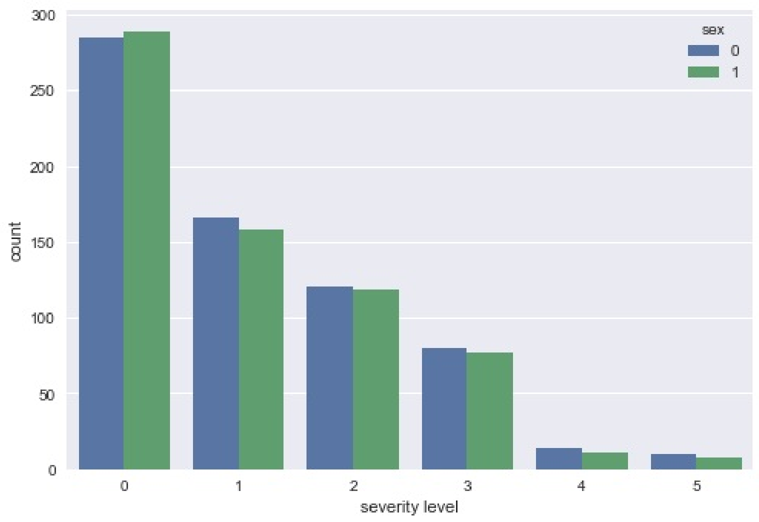
Out[25]:
```
<AxesSubplot:xlabel='severity level', ylabel='count'>
```



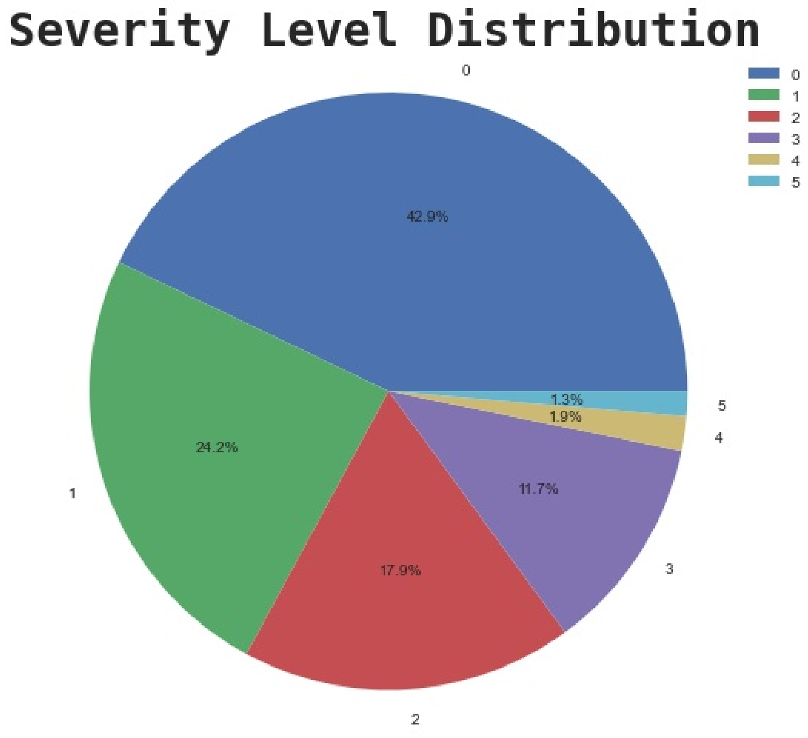


In [25]:
```
sns.countplot(x="severity level", hue = "sex", data=df)
```

Out[25]:
```
<AxesSubplot:xlabel='severity level', ylabel='count'>
```

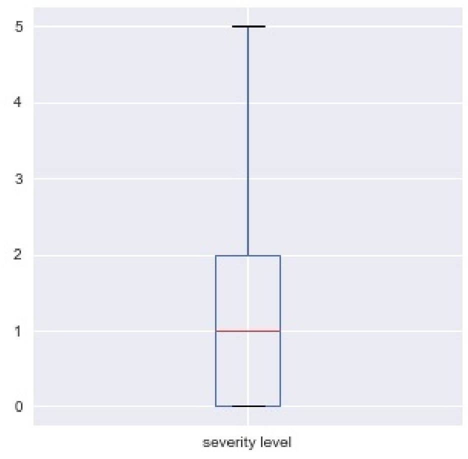


In [26]:
```
ssn = df['severity level'].value_counts()
plt.style.use('seaborn')
plt.figure(figsize = (10, 8))
plt.pie(ssn.values, labels = ssn.index, autopct = '%1.1f%%')
plt.title('Severity Level Distribution', fontdict = {'fontname' : 'Monospace','fontsize' : 30, 'fontweight' : 'bo
plt.legend()
plt.axis('equal')
plt.show()
```



In [27]:
```
cat_cols = ['severity level']

df[cat_cols].boxplot(figsize=(5,5))
```

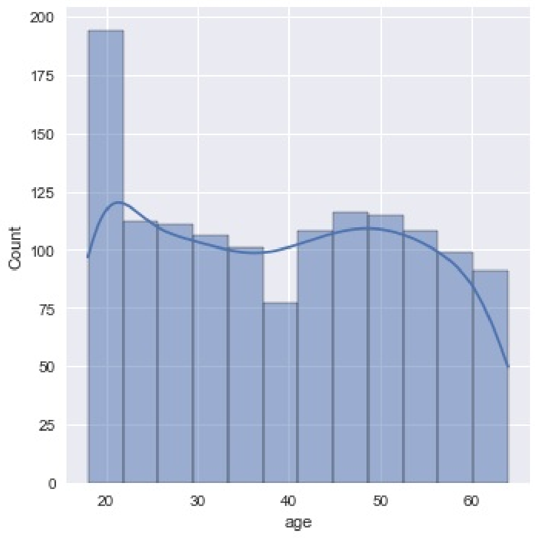
Out[27]:
```
<AxesSubplot:>
```



```
In [ ]:
```

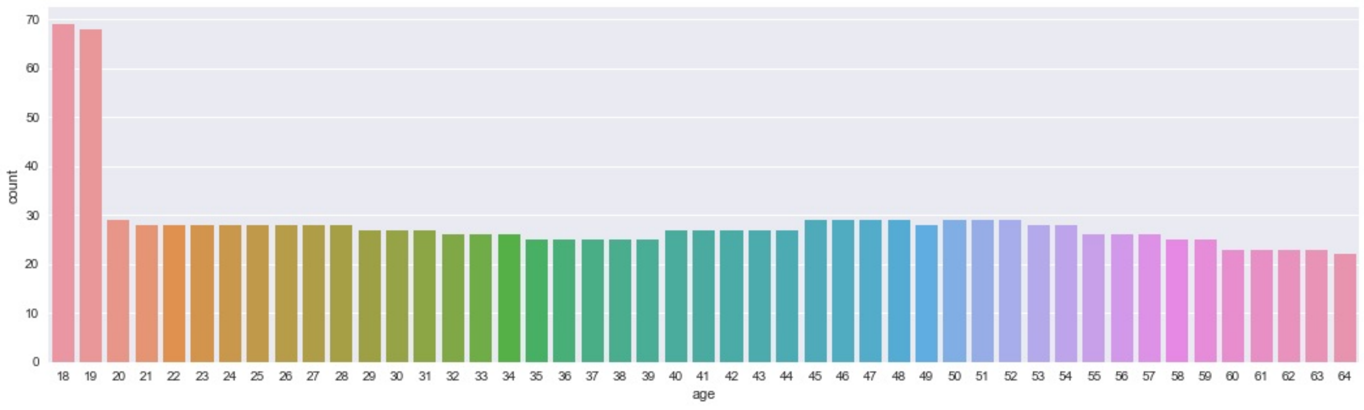
```
In [28]: #Continous variables are : age, viral load, hospitalization charges

         #age
         sns.displot( df['age'], kde = True)
```

```
Out[28]: <seaborn.axisgrid.FacetGrid at 0x2c342952fd0>
```



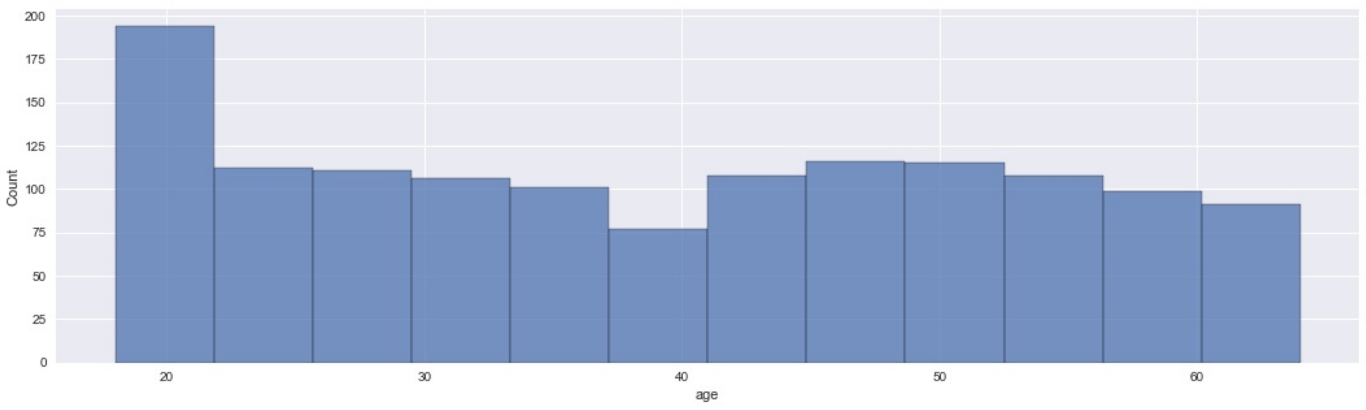
```
In [29]: fig, ax = plt.subplots(figsize=(18, 5))
         sns.countplot(x= 'age', data = df, ax= ax)
```

```
Out[29]: <AxesSubplot:xlabel='age', ylabel='count'>
```



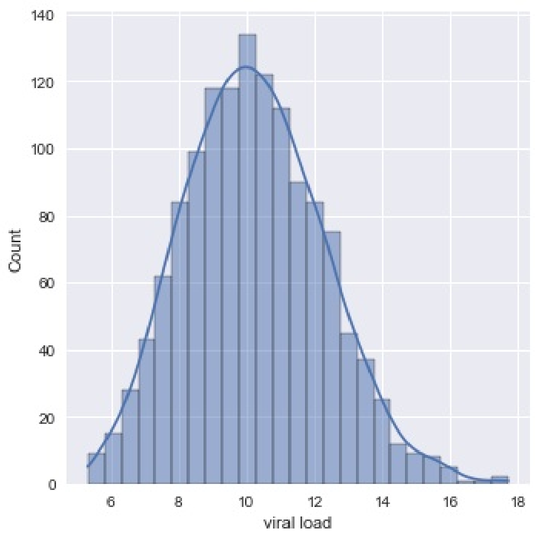
```
In [30]: fig, ax = plt.subplots(figsize=(18, 5))
         sns.histplot(data = df, x = 'age', ax=ax)
```

Out[30]:
```
<AxesSubplot:xlabel='age', ylabel='Count'>
```



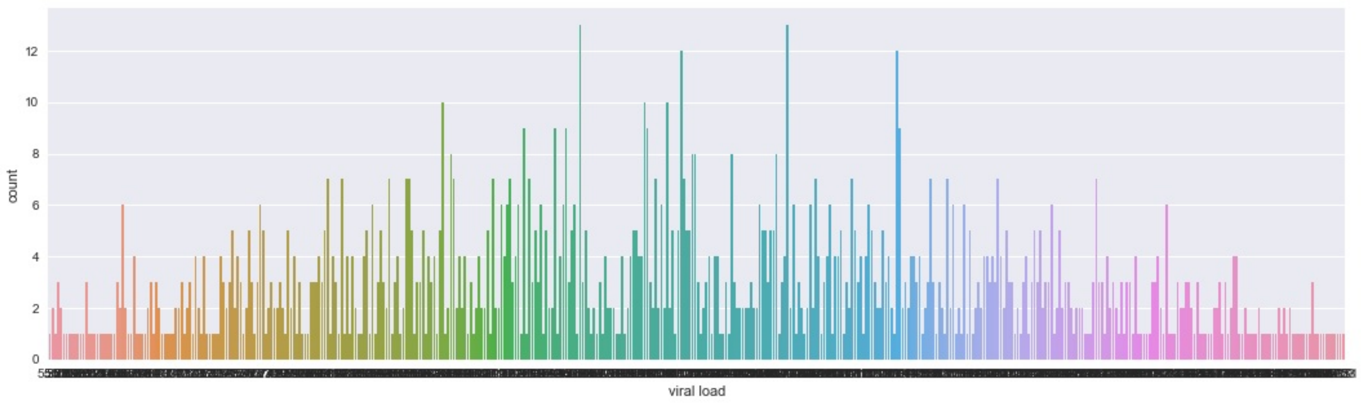
In [31]:
```
#viral load

sns.displot( df['viral load'], kde= True)
```

Out[31]:
```
<seaboarn.axisgrid.FacetGrid at 0x2c34295f040>
```



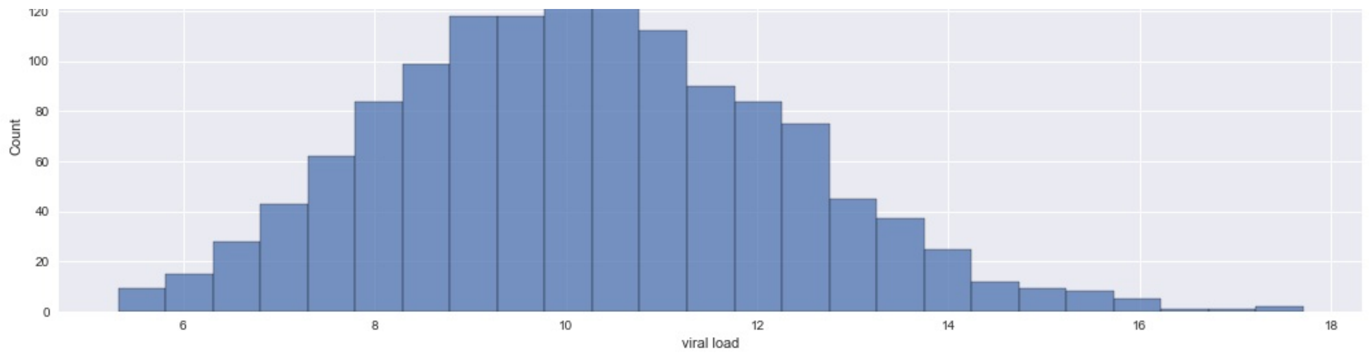
In [32]:
```
fig, ax = plt.subplots(figsize=(18, 5))
sns.countplot(x= 'viral load', data = df, ax= ax)
```

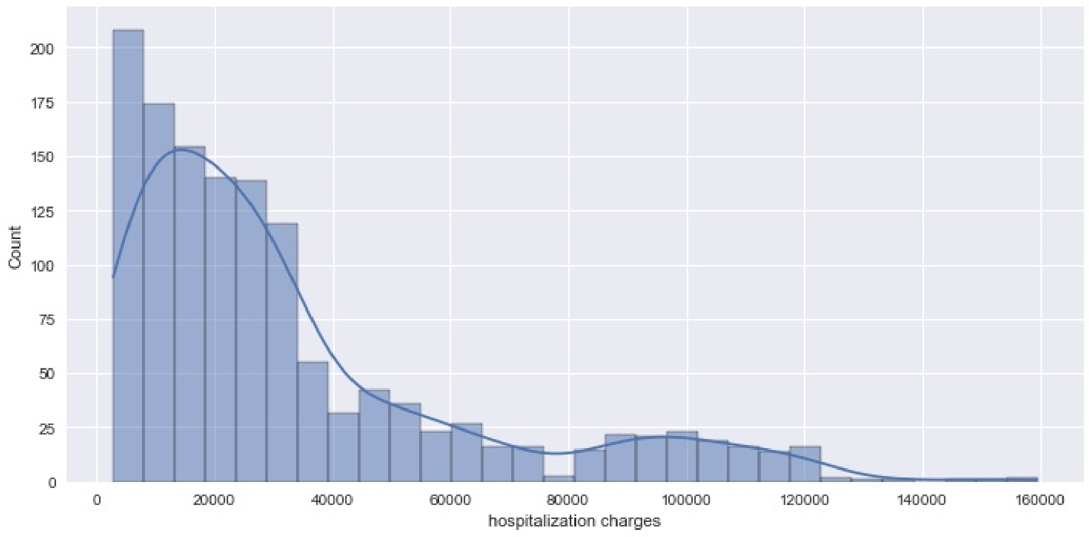
Out[32]:
```
<AxesSubplot:xlabel='viral load', ylabel='count'>
```

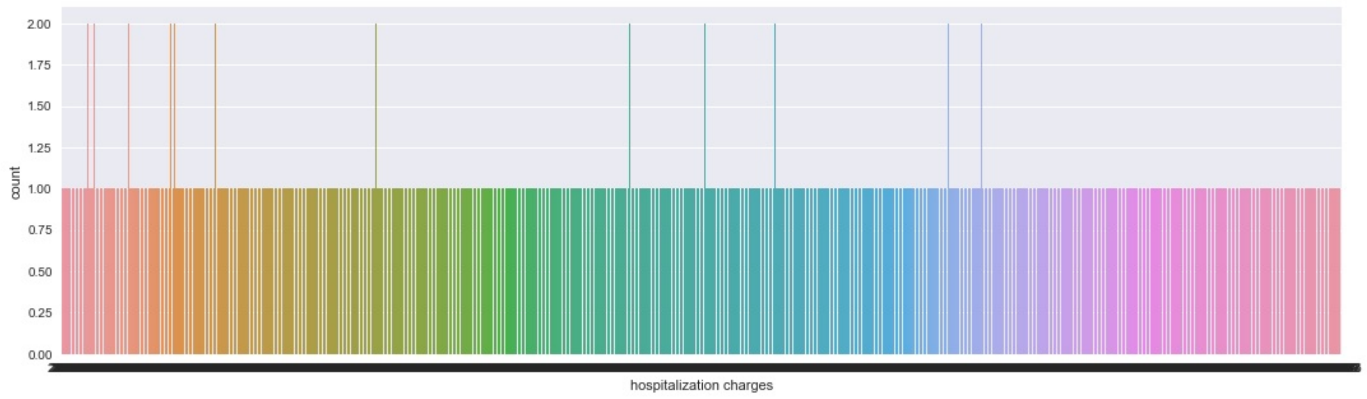


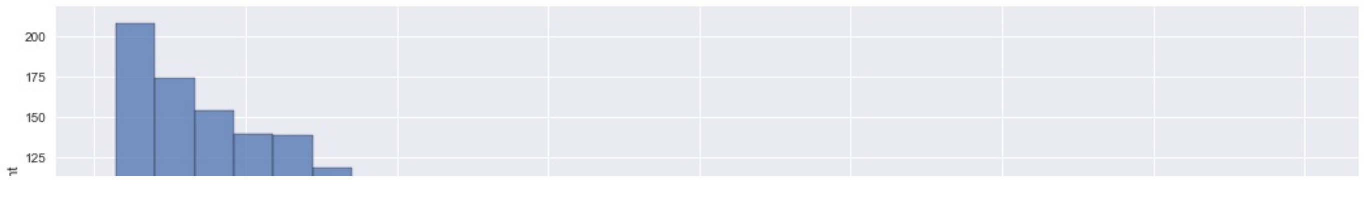
In [33]:
```
fig, ax = plt.subplots(figsize=(18, 5))
sns.histplot(data = df, x = 'viral load', ax=ax)
```

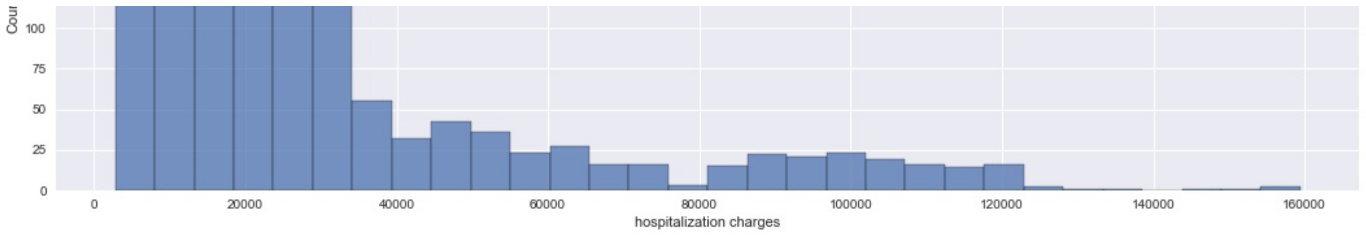
Out[33]:
```
<AxesSubplot:xlabel='viral load', ylabel='Count'>
```



In [34]:
```
#hospitalization charges

sns.displot( df['hospitalization charges'], kde= True, aspect = 2)
```
Out[34]:
```
<seaborn.axisgrid.FacetGrid at 0x2c342b44a60>
```


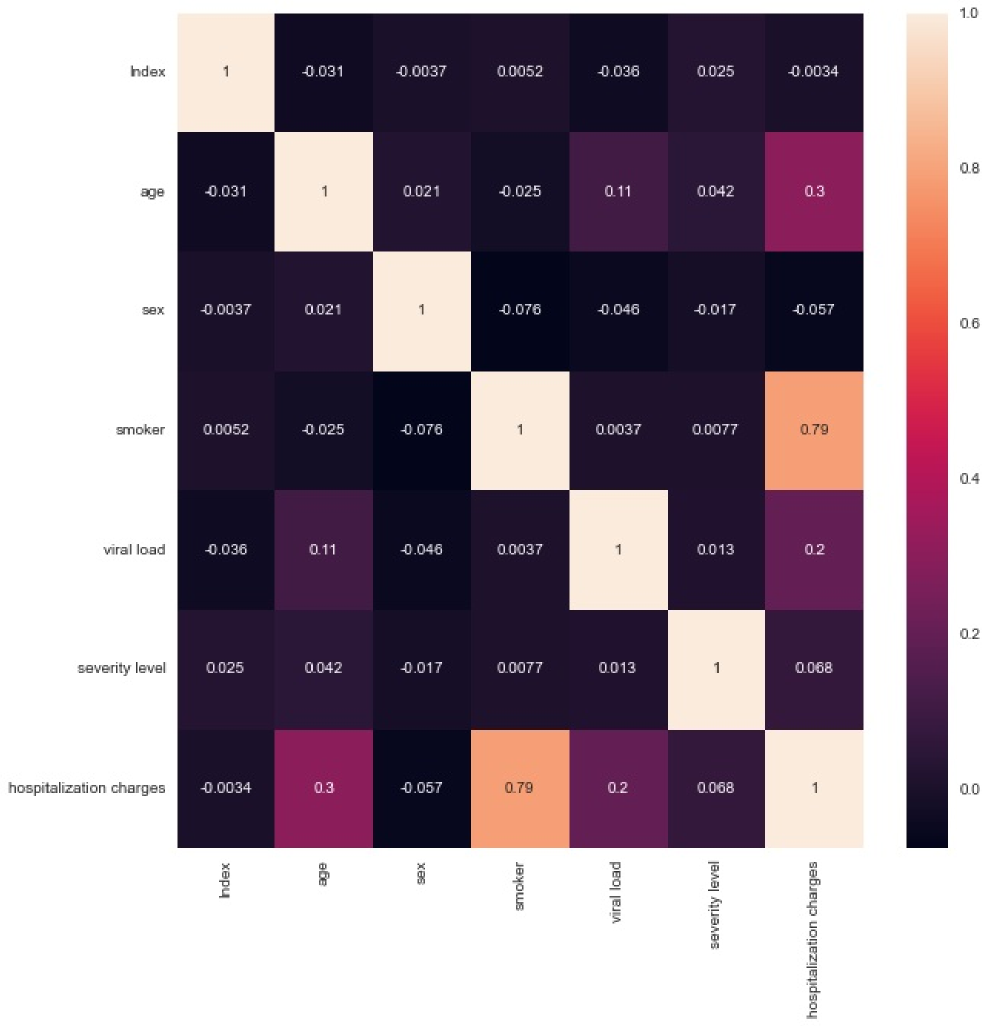
In [35]:
```
fig, ax = plt.subplots(figsize=(18, 5))
sns.countplot(x= 'hospitalization charges', data = df, ax= ax)
```
Out[35]:
```
<AxesSubplot:xlabel='hospitalization charges', ylabel='count'>
```


In [36]:
```
fig, ax = plt.subplots(figsize=(18, 5))
sns.histplot(data = df, x = 'hospitalization charges', ax=ax)
```
Out[36]:
```
<AxesSubplot:xlabel='hospitalization charges', ylabel='Count'>
```




In [37]:
```python
#Bivariate analysis

#Heatmap

#Since variables aren't normally distributed we do not consider Pearson correlation
fig, ax = plt.subplots(figsize=(10, 10))
Var_Corr = df.corr(method = 'pearson')
sns.heatmap(Var_Corr, xticklabels=Var_Corr.columns, yticklabels=Var_Corr.columns, annot=True, ax=ax)
```

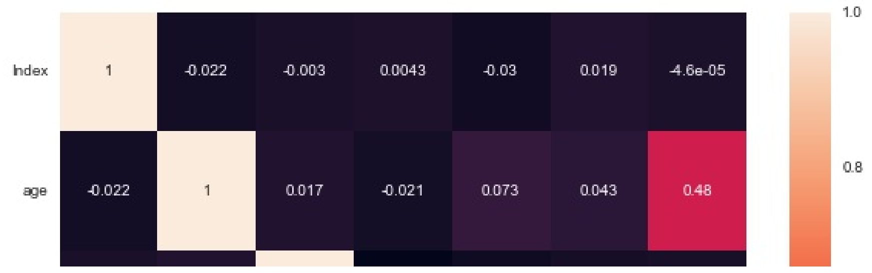
Out[37]:
```
<AxesSubplot:>
```



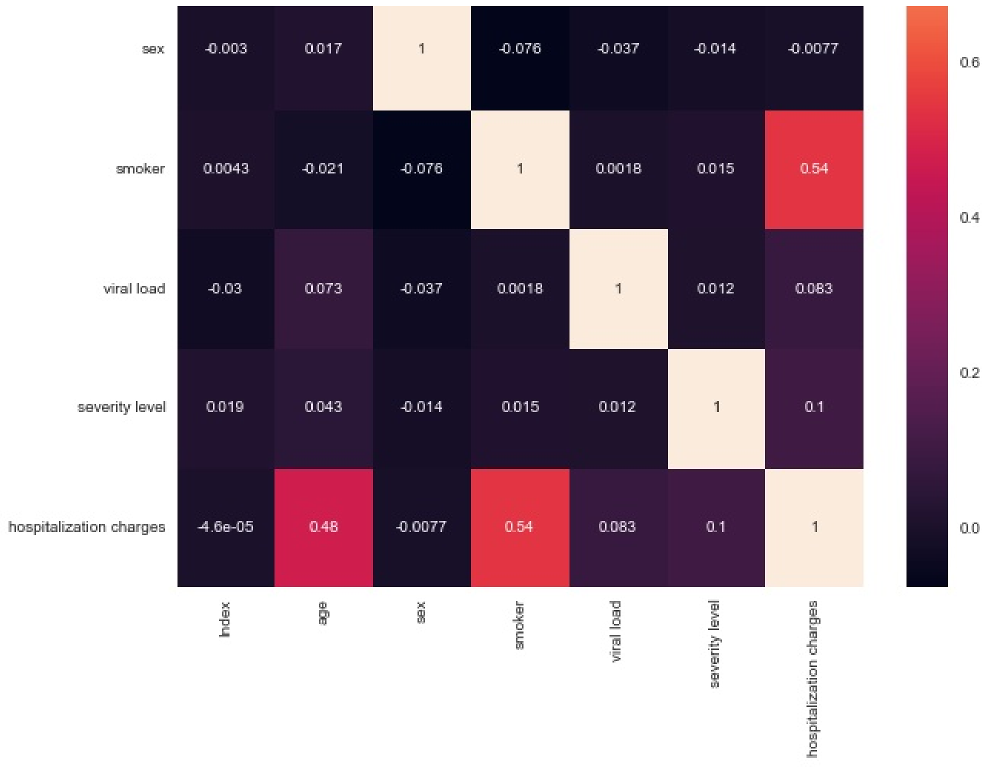
| | Index | age | sex | smoker | viral load | severity level | hospitalization charges |
|---|---|---|---|---|---|---|---|
| Index | 1 | -0.031 | -0.0037 | 0.0052 | -0.036 | 0.025 | -0.0034 |
| age | -0.031 | 1 | 0.021 | -0.025 | 0.11 | 0.042 | 0.3 |
| sex | -0.0037 | 0.021 | 1 | -0.076 | -0.046 | -0.017 | -0.057 |
| smoker | 0.0052 | -0.025 | -0.076 | 1 | 0.0037 | 0.0077 | 0.79 |
| viral load | -0.036 | 0.11 | -0.046 | 0.0037 | 1 | 0.013 | 0.2 |
| severity level | 0.025 | 0.042 | -0.017 | 0.0077 | 0.013 | 1 | 0.068 |
| hospitalization charges | -0.0034 | 0.3 | -0.057 | 0.79 | 0.2 | 0.068 | 1 |

In [38]:
```python
fig, ax = plt.subplots(figsize=(10, 10))
Var_Corr = df.corr(method = 'kendall')
sns.heatmap(Var_Corr, xticklabels=Var_Corr.columns, yticklabels=Var_Corr.columns, annot=True, ax=ax)
```

Out[38]:
```
<AxesSubplot:>
```



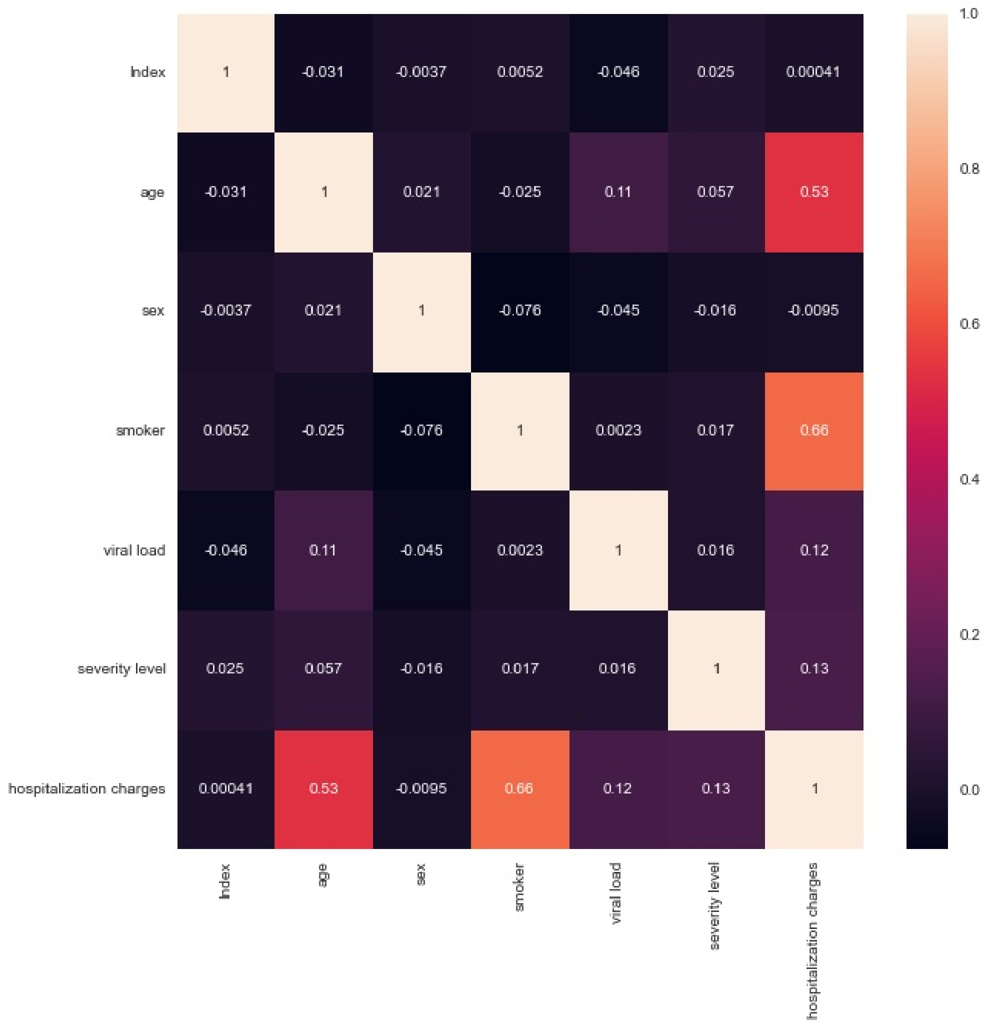
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Index | 1 | -0.022 | -0.003 | 0.0043 | -0.03 | 0.019 | -4.6e-05 |
| age | -0.022 | 1 | 0.017 | -0.021 | 0.073 | 0.043 | 0.48 |



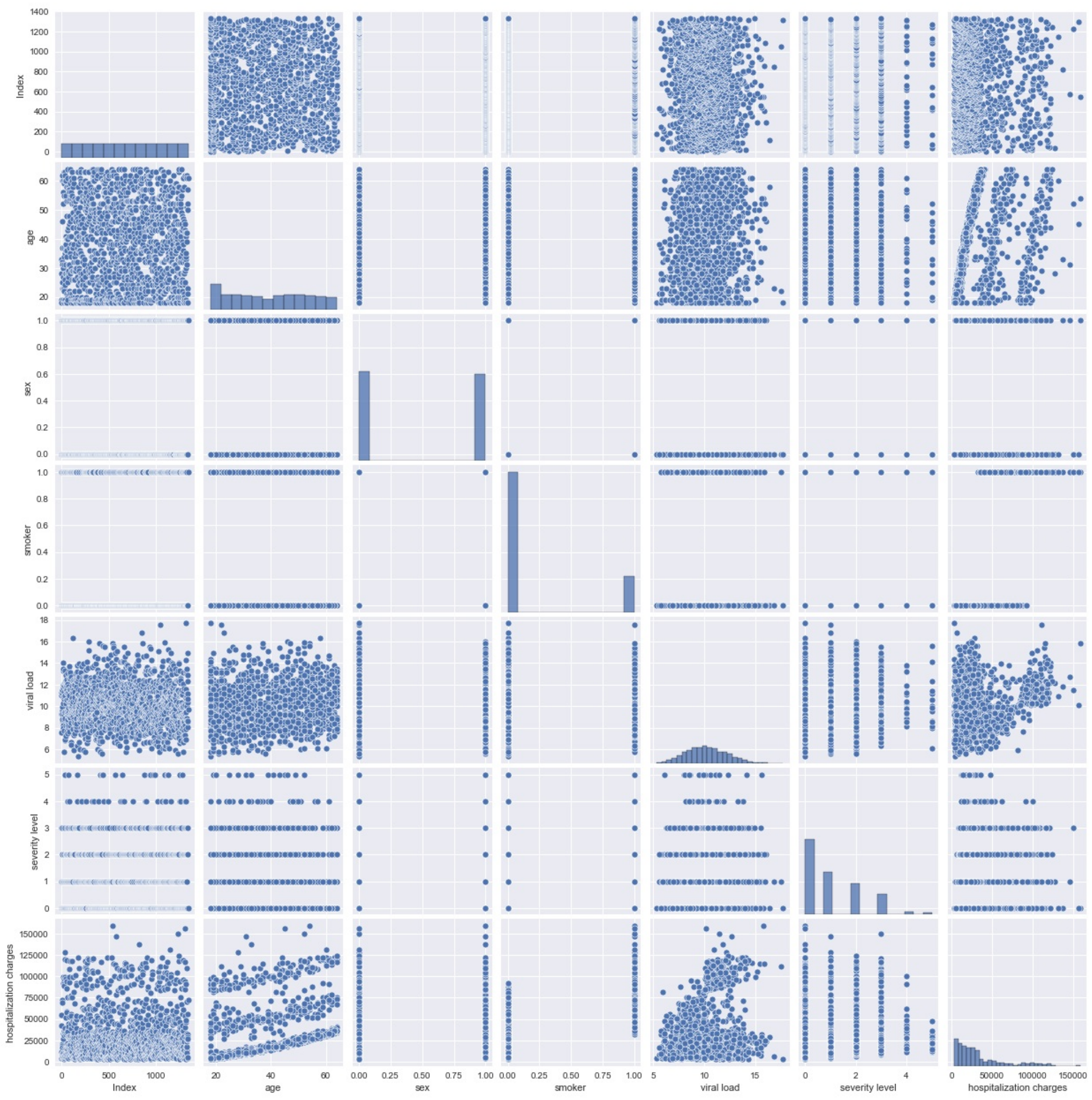
|  | Index | age | sex | smoker | viral load | severity level | hospitalization charges |
|---|---|---|---|---|---|---|---|
| sex | -0.003 | 0.017 | 1 | -0.076 | -0.037 | -0.014 | -0.0077 |
| smoker | 0.0043 | -0.021 | -0.076 | 1 | 0.0018 | 0.015 | 0.54 |
| viral load | -0.03 | 0.073 | -0.037 | 0.0018 | 1 | 0.012 | 0.083 |
| severity level | 0.019 | 0.043 | -0.014 | 0.015 | 0.012 | 1 | 0.1 |
| hospitalization charges | -4.6e-05 | 0.48 | -0.0077 | 0.54 | 0.083 | 0.1 | 1 |

In [39]:
```python
fig, ax = plt.subplots(figsize=(10, 10))
Var_Corr = df.corr(method = 'spearman')
sns.heatmap(Var_Corr, xticklabels=Var_Corr.columns, yticklabels=Var_Corr.columns, annot=True, ax=ax)


#Hospitalization charges and age
#Hospitalization charges and smoker
# Both show strong positive correlation
```

Out[39]:
```
<AxesSubplot:>
```

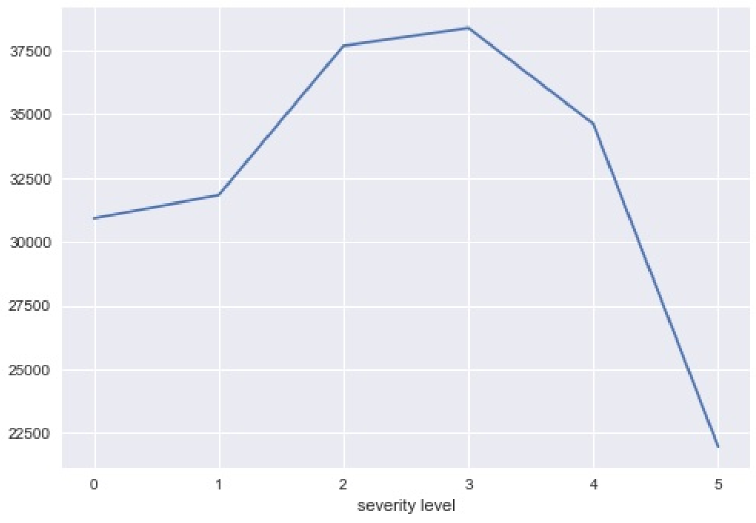


|  | Index | age | sex | smoker | viral load | severity level | hospitalization charges |
|---|---|---|---|---|---|---|---|
| Index | 1 | -0.031 | -0.0037 | 0.0052 | -0.046 | 0.025 | 0.00041 |
| age | -0.031 | 1 | 0.021 | -0.025 | 0.11 | 0.057 | 0.53 |
| sex | -0.0037 | 0.021 | 1 | -0.076 | -0.045 | -0.016 | -0.0095 |
| smoker | 0.0052 | -0.025 | -0.076 | 1 | 0.0023 | 0.017 | 0.66 |
| viral load | -0.046 | 0.11 | -0.045 | 0.0023 | 1 | 0.016 | 0.12 |
| severity level | 0.025 | 0.057 | -0.016 | 0.017 | 0.016 | 1 | 0.13 |
| hospitalization charges | 0.00041 | 0.53 | -0.0095 | 0.66 | 0.12 | 0.13 | 1 |

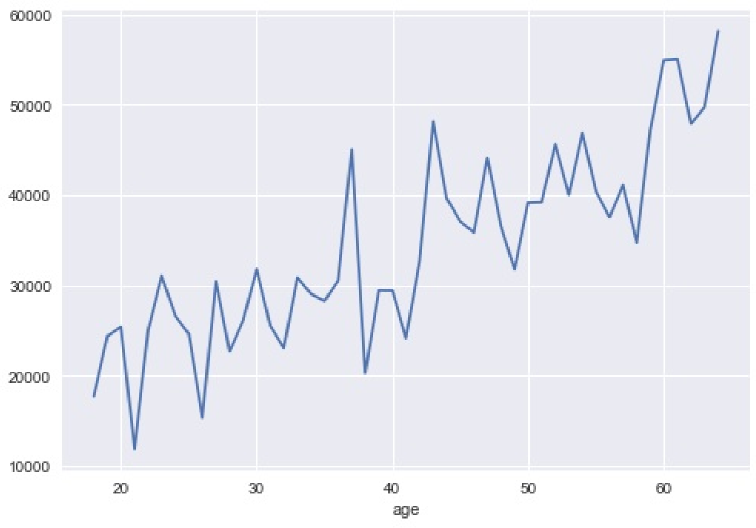
In [40]:
```
sns.pairplot(df)
```
Out[40]:
```
<seaborn.axisgrid.PairGrid at 0x2c3475f9880>
```


In [41]:
```
#Viral load by sex
df.groupby('sex')['viral load'].mean()
```
Out[41]:
```
sex
0    10.314423
1    10.126073
Name: viral load, dtype: float64
```

In [42]:
```
#hospitalization by smoking
df.groupby('smoker')['hospitalization charges'].mean()

#the difference for hospitalization charges for smoker and non smoker is significant
```
Out[42]:
```
smoker
0    21085.675752
1    80125.572993
Name: hospitalization charges, dtype: float64
```

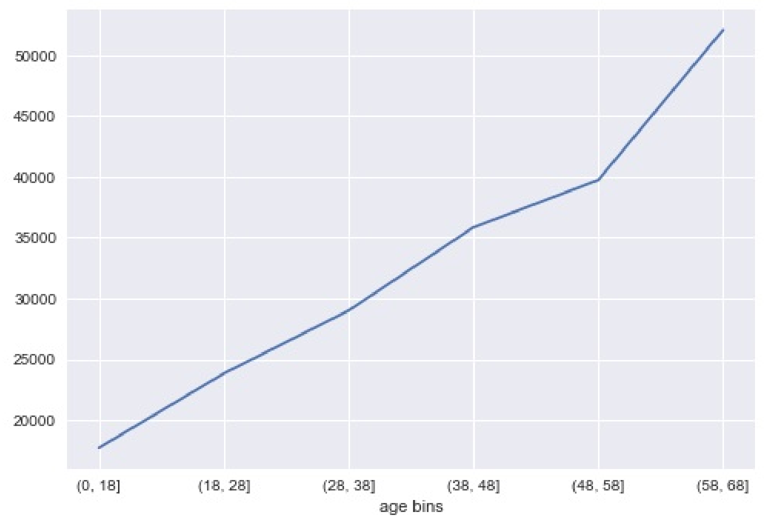
In [43]:
```
#hospitalization charges by severity level
df.groupby('severity level')['hospitalization charges'].mean().plot()
```
Out[43]:
```
<AxesSubplot:xlabel='severity level'>
```


In [44]:
```
#viral load by smoking category
df.groupby('smoker')['viral load'].mean()

#viral load is indepedent of whether patient is smoker or not as per this
```
Out[44]:
```
smoker
0    10.217378
1    10.236204
Name: viral load, dtype: float64
```

In [45]:
```
df.groupby('age')['hospitalization charges'].mean().plot()

#strongly correlated
```
Out[45]:
```
<AxesSubplot:xlabel='age'>
```


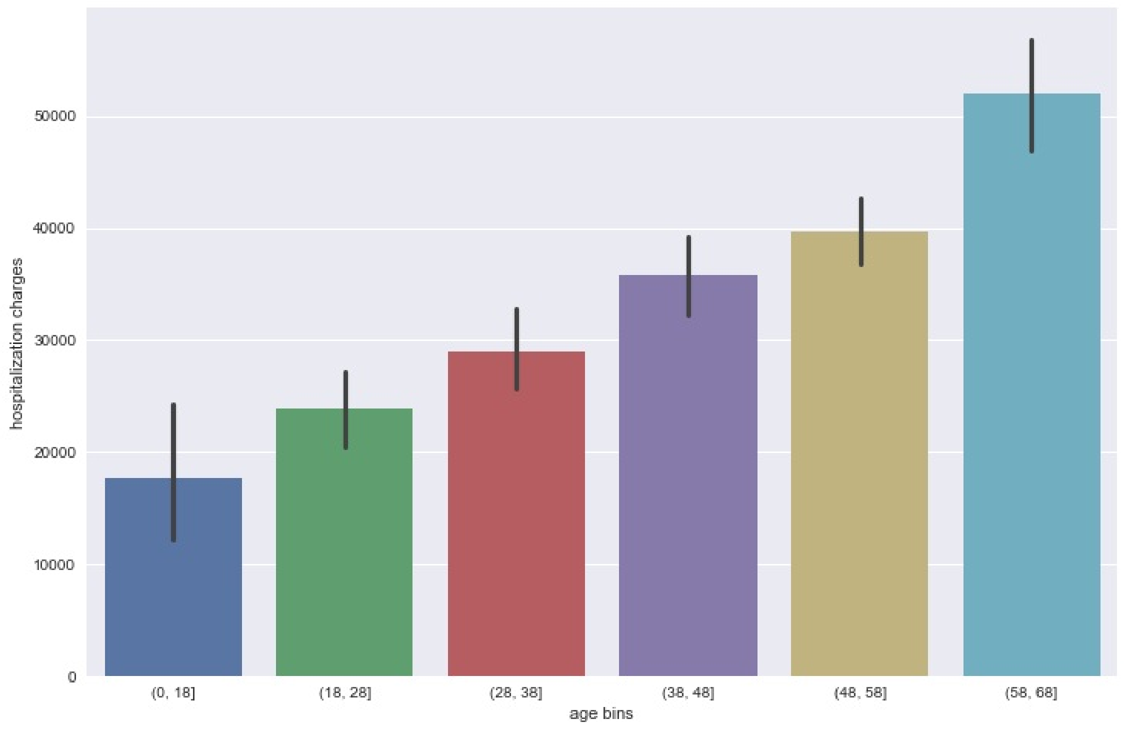
In [46]:
```
#grouping age into bins
bins = [0,18,28,38,48,58,68]
labels = ['0-18','18-28','28-38','38-48','48-58','58-68']
df['age bins'] = pd.cut(df['age'], bins, labels)
```
In [47]:
```
df.groupby('age bins')['hospitalization charges'].mean().plot()

#as age increases we see hospitalization charges increase linearly
```

Out[47]:
```
<AxesSubplot:xlabel='age bins'>
```



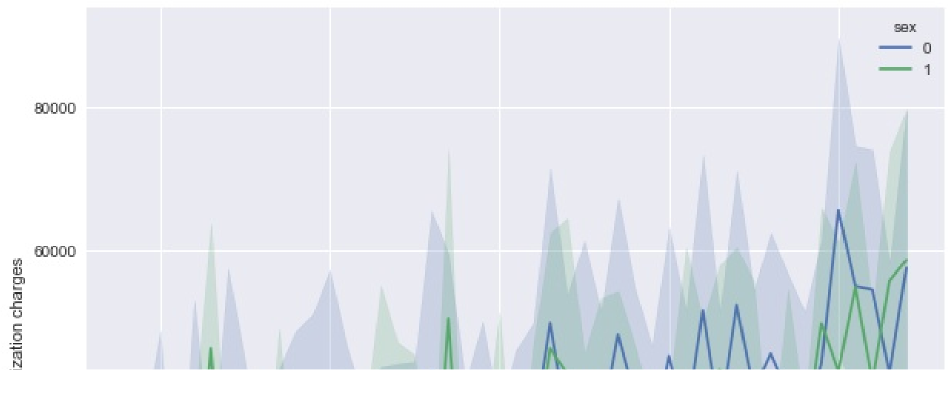
In [48]:
```python
plt.figure(figsize = (12,8))
sns.barplot(x='age bins', y = 'hospitalization charges', data = df)
```

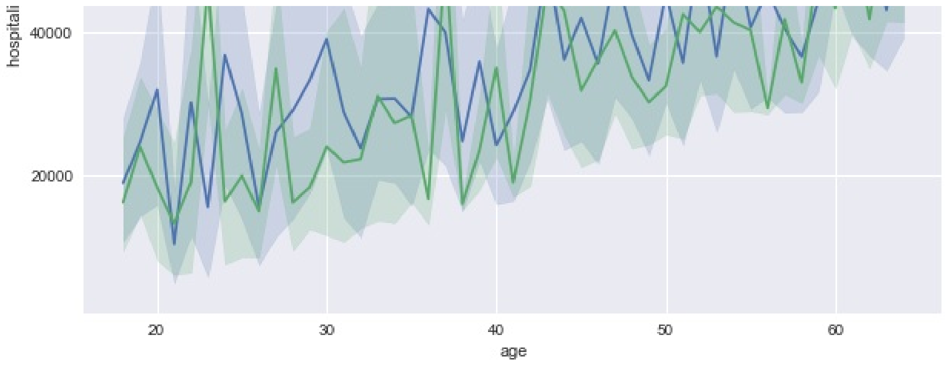
Out[48]:
```
<AxesSubplot:xlabel='age bins', ylabel='hospitalization charges'>
```



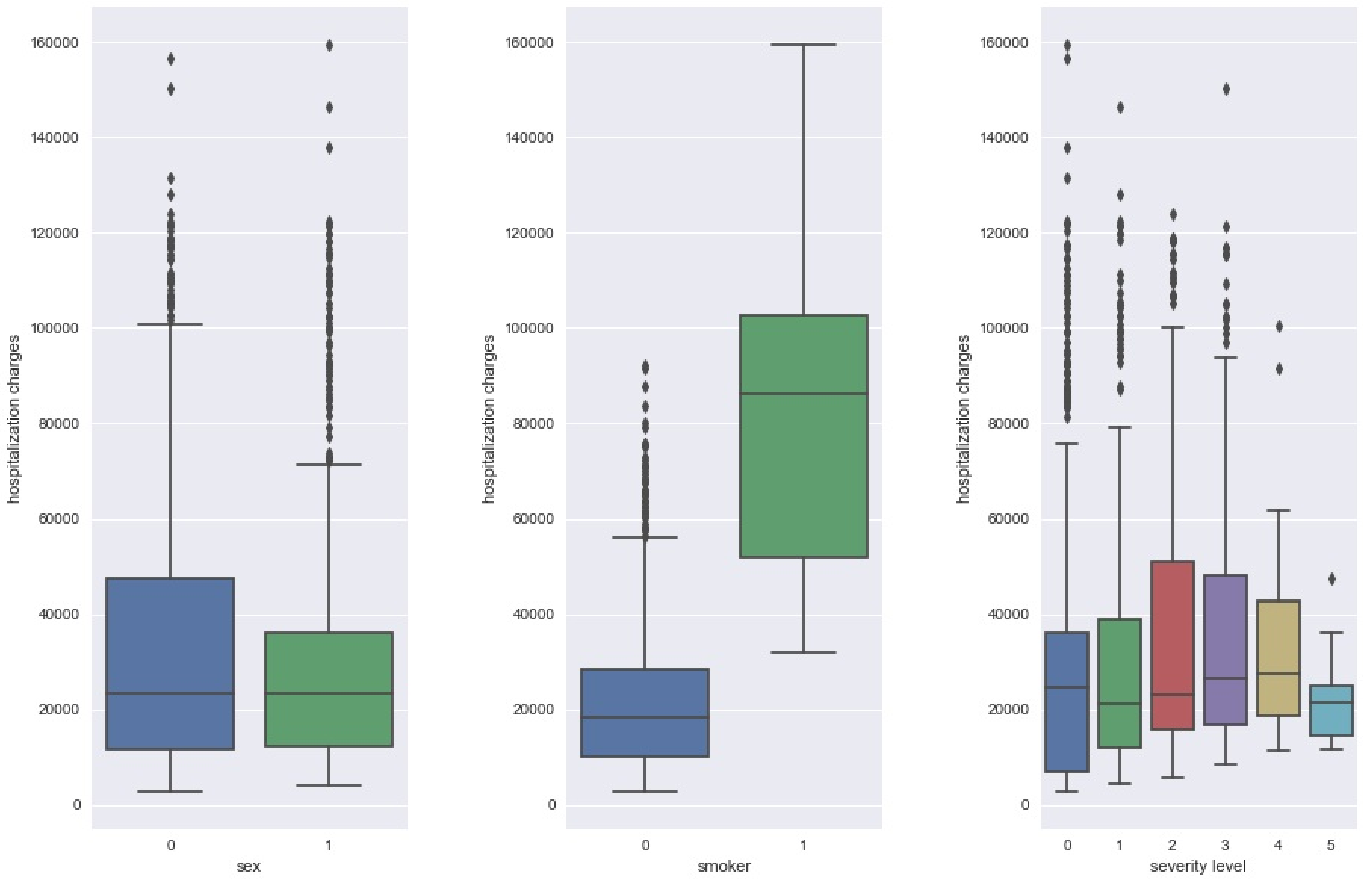
In [49]:
```python
plt.figure(figsize = (10,8))
sns.lineplot(x='age', y = 'hospitalization charges', hue = 'sex', data = df)

#not muhc of a difference between males and females in hospitalization charges vs age.
```

Out[49]:
```
<AxesSubplot:xlabel='age', ylabel='hospitalization charges'>
```



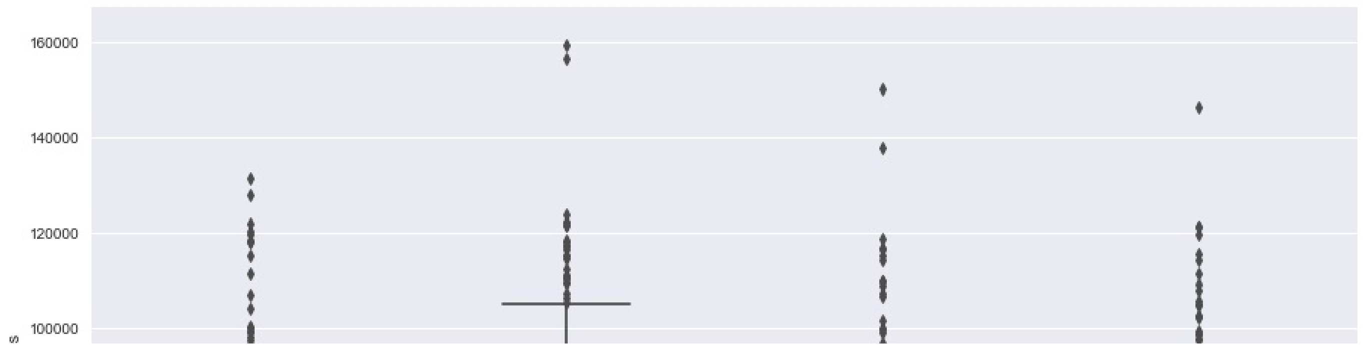


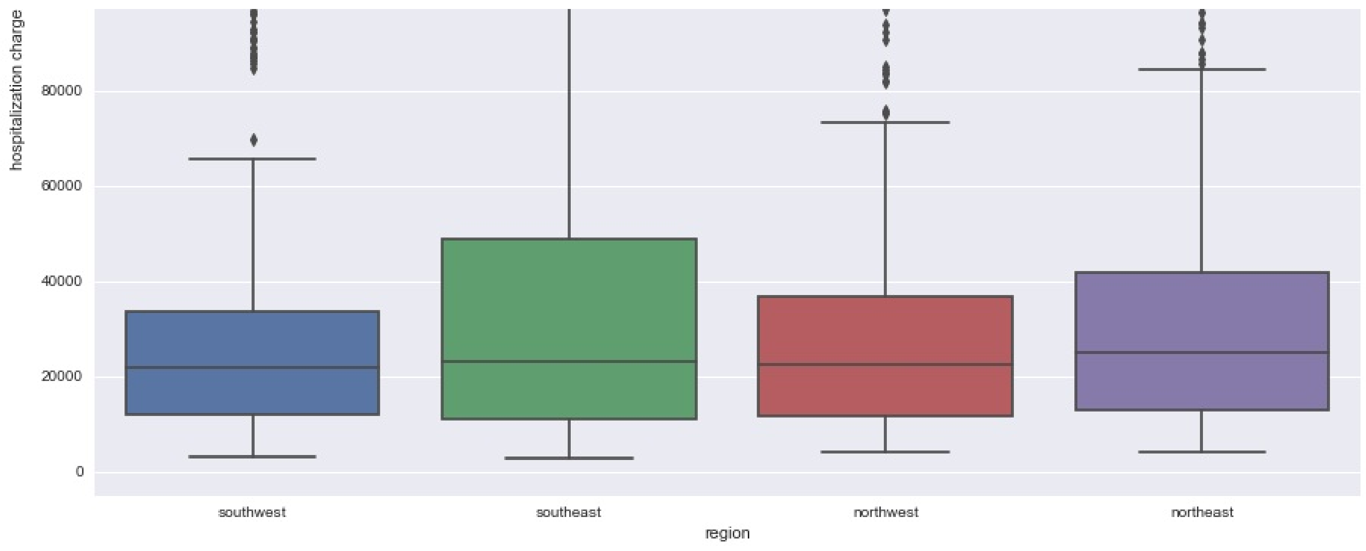
In [53]:
```
plt.figure(figsize = (15,10))
cat_cols = ['sex', 'smoker', 'severity level']
for i,j in enumerate(cat_cols):
    #row 1, col 3, index begins i is 1 based hence i+1
    plt.subplot(1,3,i+1)
    plt.subplots_adjust(wspace = 0.5, hspace = 2)
    sns.boxplot(x=j, y ='hospitalization charges', data=df)
```



In [51]:
```
#Except for smokers the hospitalization charges do not chage for across sex, severity level and region

plt.figure(figsize = (15,10))
sns.boxplot(x='region', y ='hospitalization charges', data=df)
```

Out[51]:
```
<AxesSubplot:xlabel='region', ylabel='hospitalization charges'>
```





```
In [52]: df.groupby(['sex','smoker','age bins'])['hospitalization charges'].mean().unstack()
```

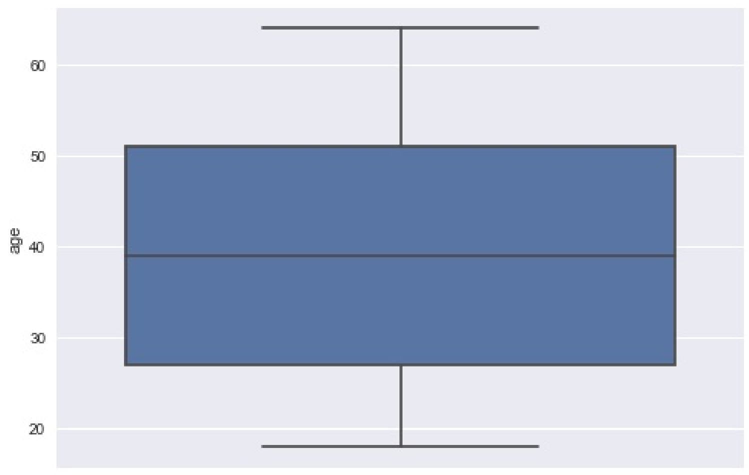
Out[52]:

| | age bins | (0, 18] | (18, 28] | (28, 38] | (38, 48] | (48, 58] | (58, 68] |
|---|---|---|---|---|---|---|---|
| sex | smoker | | | | | | |
| 0 | 0 | 6739.285714 | 9617.187500 | 16996.948454 | 21754.786408 | 29646.357798 | 36771.500000 |
| | 1 | 61948.750000 | 76063.710526 | 74800.942857 | 83855.457143 | 91621.370370 | 107589.625000 |
| 1 | 0 | 9291.862069 | 13081.352000 | 14050.326923 | 22979.908257 | 31545.153226 | 39288.660714 |
| | 1 | 67156.000000 | 62991.233333 | 77482.782609 | 76219.586207 | 90458.214286 | 93530.866667 |

```
In [ ]:
```

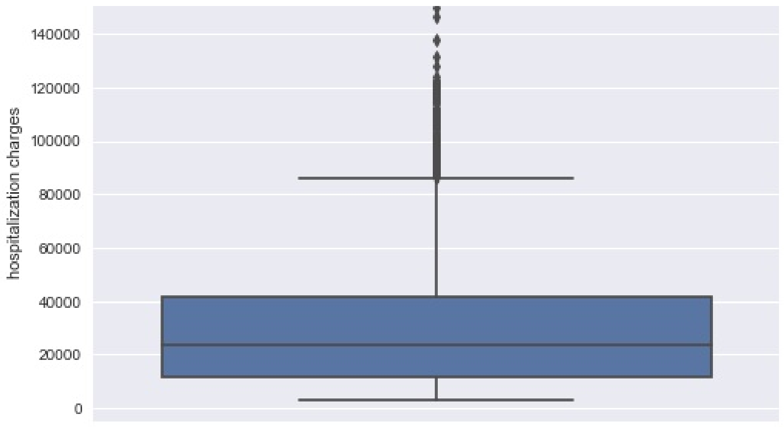
```
In [55]: #Missing value treatment

         #There are no missing values as we have seen earlier

         #Outlier Treatment

         # Hospitalization charges,  Age and Viral Load are the variables that we need to check for outlier treatment


         sns.boxplot(y=df['age'])
```

```
Out[55]: <AxesSubplot:ylabel='age'>
```



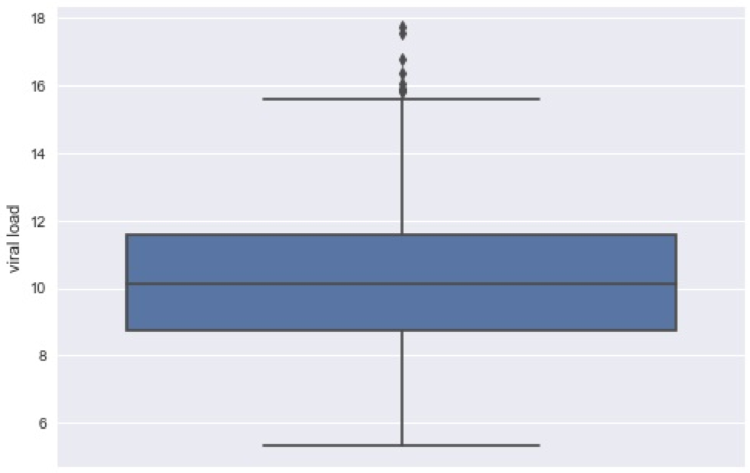
```
In [56]: sns.boxplot(y=df['hospitalization charges'])
```

```
Out[56]: <AxesSubplot:ylabel='hospitalization charges'>
```



In [57]:
```
sns.boxplot(y=df['viral load'])
```

Out[57]:
```
<AxesSubplot:ylabel='viral load'>
```

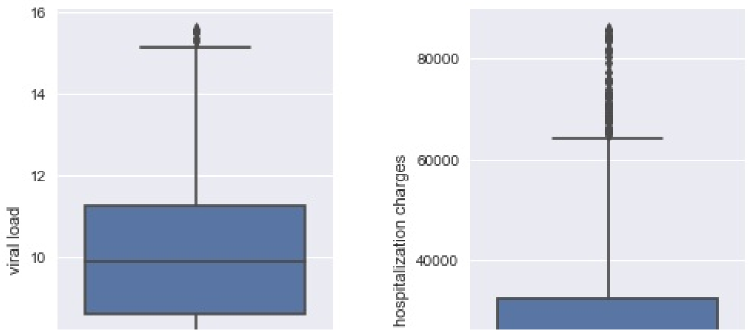


In [58]:
```
#Age does not have outliers as is clear from the boxplot and also the statistical summary drawn earlier
#we treat viral load and hospitalization charges for outlier treatment

o_cols = ['viral load', 'hospitalization charges']

for i in o_cols:
    Q1ss = df[i].quantile(0.25)
    Q3ss = df[i].quantile(0.75)
    IQRss = Q3ss - Q1ss
    df = df[ ~ ((df[i] < (Q1ss -  1.5 * IQRss)) | (df[i] > (Q3ss + 1.5 * IQRss))) ]
```

In [59]:
```
df.shape
```

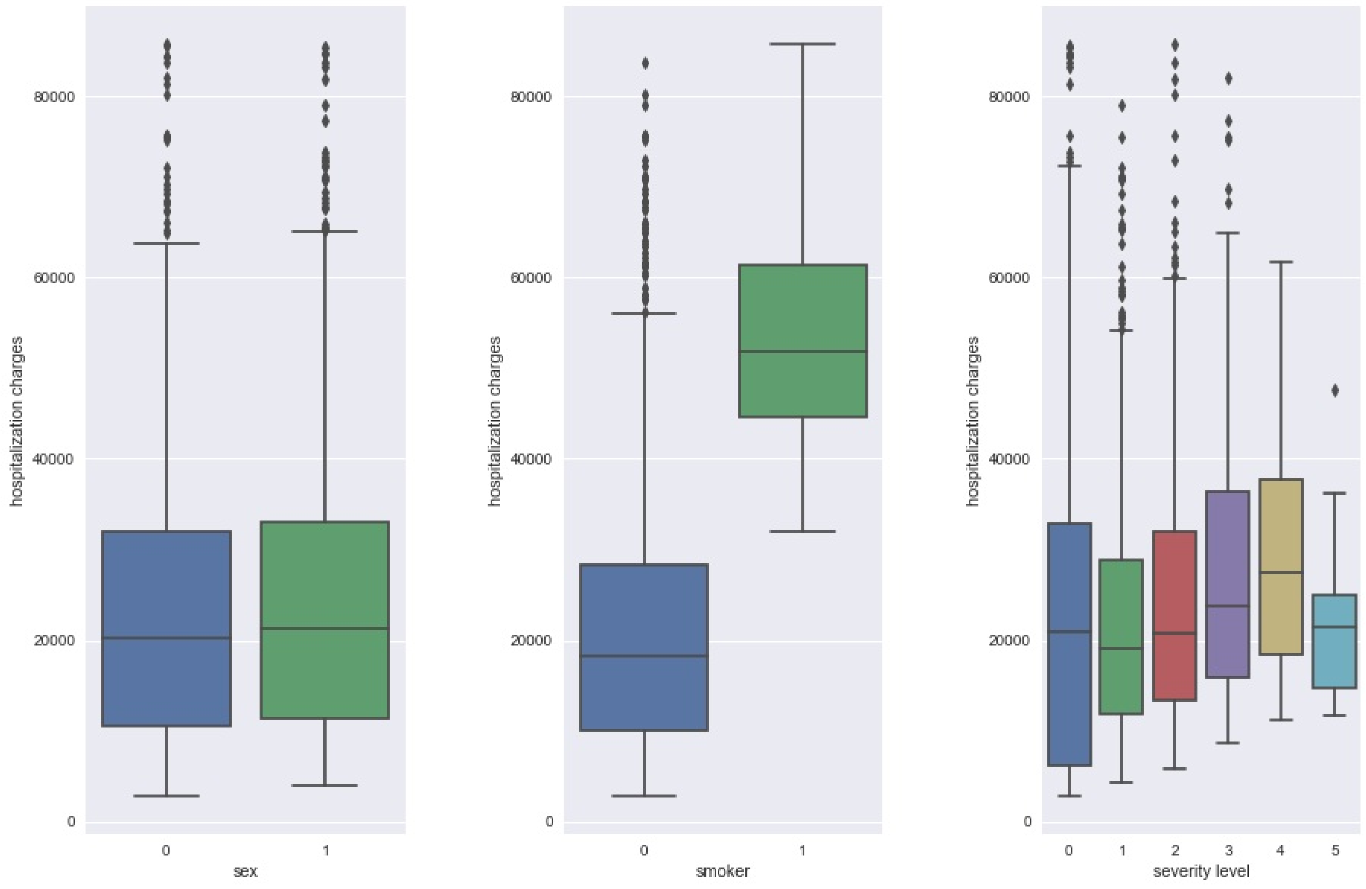
Out[59]:
```
(1191, 9)
```

In [62]:
```
for i,j in enumerate(o_cols):
    #row 1, col 3, index begins i is 1 based hence i+1
    plt.subplot(1,2,i+1)
    plt.subplots_adjust(wspace = 0.5, hspace = 2)
    sns.boxplot(y =df[j], data=df)
```



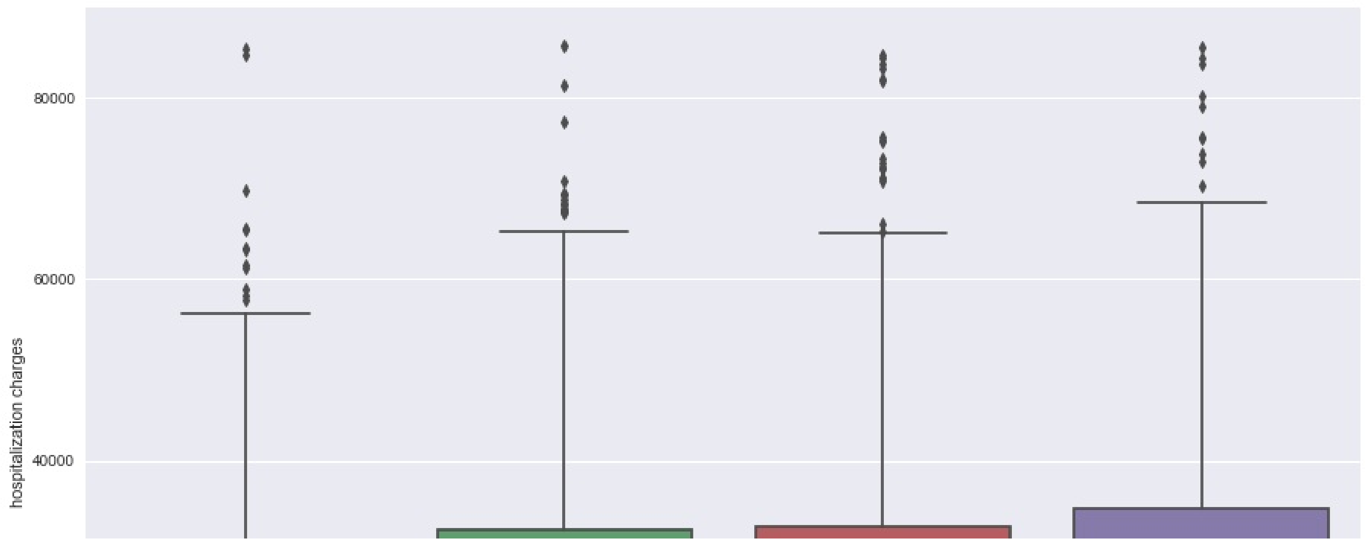
In [59]:
```
#There are still outliers present but their range and number has reduced significantly. Applying outlier treatmen
#again would lead to loss of data i.e. smaller sample size as more rows would be deleted
```

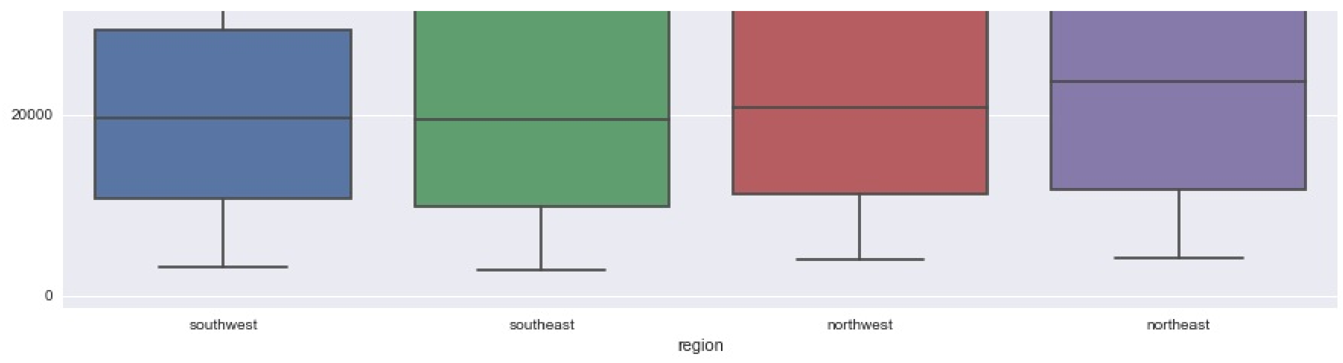
In [65]:
```
plt.figure(figsize = (15,10))
for i,j in enumerate(cat_cols):
    #row 1, col 3, index begins i is 1 based hence i+1
    plt.subplot(1,3,i+1)
    plt.subplots_adjust(wspace = 0.5, hspace = 2)
    sns.boxplot(x=j, y ='hospitalization charges', data=df)
```



In [66]:
```
plt.figure(figsize = (15,10))
sns.boxplot(x='region', y ='hospitalization charges', data=df)
```

Out[66]:
```
<AxesSubplot:xlabel='region', ylabel='hospitalization charges'>
```





In [67]:
```
#Hypothesis Testing
```

In [68]:
```
#Prove (or disprove) that the hospitalization charges of people who do smoking are greater than those who don't?

#We can do it using right tailed t test

smog = df.groupby('smoker')
```

In [69]:
```
smog['hospitalization charges'].describe()
```

Out[69]:

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **smoker** | | | | | | | | |
| **0** | 1055.0 | 20907.971564 | 14563.067125 | 2805.0 | 9962.5 | 18313.0 | 28387.5 | 83680.0 |
| **1** | 136.0 | 54578.154412 | 13360.849267 | 32074.0 | 44663.5 | 51899.5 | 61421.5 | 85758.0 |

In [70]:
```
smoker = df[df['smoker'] == 1]['hospitalization charges'].sample(136)
non_smoker = df[df['smoker'] == 0]['hospitalization charges'].sample(136)
```

In [71]:
```
smoker.reset_index(drop=True)
non_smoker.reset_index(drop=True)
```

Out[71]:
```
0        4262
1       15281
2       12210
3       11072
4       32660
        ...
131     17905
132      5756
133     31608
134     26761
135     24290
Name: hospitalization charges, Length: 136, dtype: int64
```

In [67]:
```
# 1) Prove (or disprove) that the hospitalization charges of people who do smoking are greater than those who dor
```

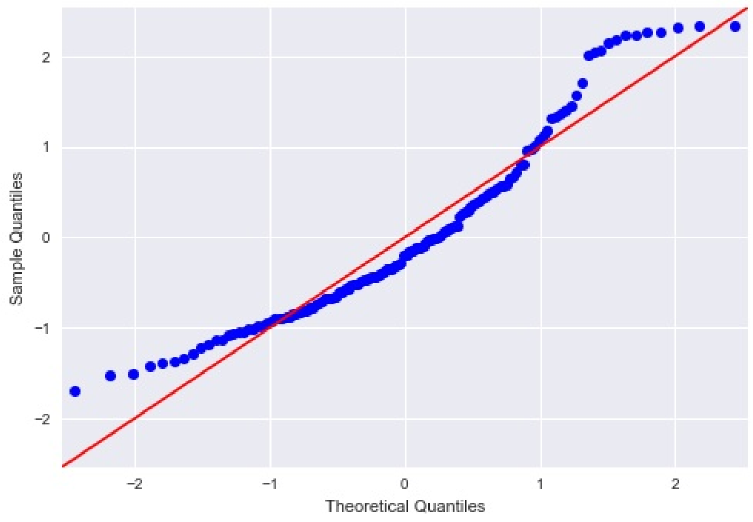
In [72]:
```
# We use Right Tailed t test

# Null hypothesis : The mean of hospitalization charges of smokers are non smokers is equal

# Alternate hypothesis : The mean of hospitalization charges of smokers is greater than non smokers

#Checkin the assumptions

from scipy import stats
import statsmodels.api as sm
from statsmodels.graphics.gofplots import qqplot
from scipy.stats import norm, uniform
from scipy.stats import levene

stats.shapiro(smoker)
```

Out[72]:
```
ShapiroResult(statistic=0.9291411638259888, pvalue=2.4488156213919865e-06)
```

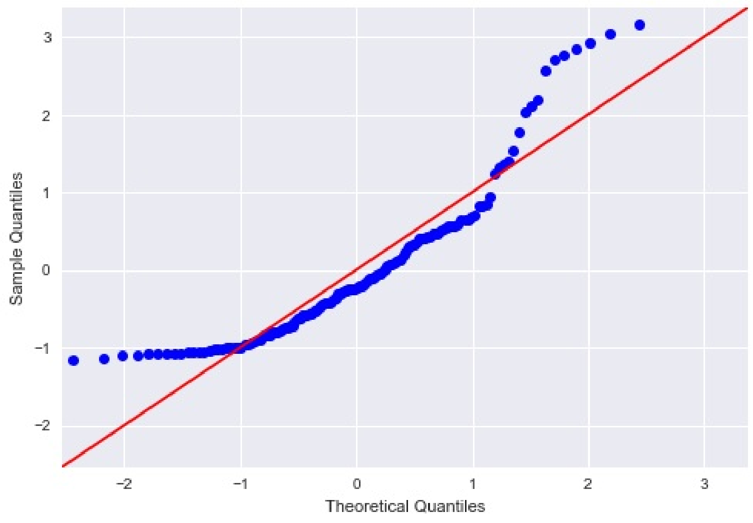
In [69]:
```
stats.shapiro(non_smoker)
```
Out[69]:
```
ShapiroResult(statistic=0.8702459931373596, pvalue=1.4824373950617087e-09)
```

In [70]:
```
#For both smoker and non smoker the p value are much smaller than 0.05 hence we know that they are NOT normally
#distributed. We further check this with q-q plot
```

In [71]:
```
qqplot(smoker,norm,fit=True,line="45")
plt.show()
```



In [72]:
```
qqplot(non_smoker,norm,fit=True,line="45")
plt.show()
```



In [73]:
```
#Standard deviations
np.var(smoker)
```
Out[73]:
```
177199702.73350984
```

In [74]:
```
np.var(non_smoker)
```
Out[74]:
```
280250246.6286765
```

In [73]:
```
#Std deviations are not equal. We can use levene's test to confirm . Levene's test gives a YES becasue it takes i
# a certain factor by which the standard devations can differ to be considered equal
alpha = 0.05
w_stats, p_value = levene(smoker, non_smoker, center = 'median')
```

In [74]:
```python
if p_value > alpha :
  print("Variances of smokers and non_smokers are equal")
else:
  print("Variances of smokers and non_smokers are NOT equal")
```
```
Variances of smokers and non_smokers are equal
```

In [76]:
```python
stats.shapiro(np.log(smoker))
```
Out[76]:
```
ShapiroResult(statistic=0.9721938967704773, pvalue=0.00695952819660306)
```

In [77]:
```python
stats.shapiro(np.log(non_smoker))
```
Out[77]:
```
ShapiroResult(statistic=0.9450490474700928, pvalue=3.225280670449138e-05)
```

In [78]:
```python
#Even the log of smoker and non_smoker aren't normally distributed
```

In [83]:
```python
#For smoker and non_smoker
#1) The distributiuons are not normal for both of them
#2) But their variances are equal

#Proceeding with the T test

def t_test(x,y,alternative):
            _, double_p = stats.ttest_ind(x,y)

            if alternative == 'two-sided':
                pval = double_p
            elif alternative == 'greater':
                if np.mean(x) > np.mean(y):
                    pval = double_p/2.
                else:
                    pval = 1.0 - double_p/2.
            elif alternative == 'less':
                if np.mean(x) < np.mean(y):
                    pval = double_p/2.
                else:
                    pval = 1.0 - double_p/2.
            return pval
```

In [86]:
```python
p_val  = t_test(smoker, non_smoker, 'greater')
print(p_val)
```
```
2.8995676943672616e-63
```

In [87]:
```python
stats.ttest_ind(smoker, non_smoker, alternative='greater')
```
Out[87]:
```
Ttest_indResult(statistic=22.23571094828444, pvalue=2.8995676943672616e-63)
```

In [88]:
```python
#As we can see on  calcualting from both ways p value is smaller than alpha =  0.05  . REJECT NULL HYPOTHESIS

if p_val < 0.05:
    print("We reject the null hypothesis ")
    print('''The means hospitalization charges for smokers and non smokers are not equal. The mean hospitalizatio
charges for smokers and greater than non smokers''')
else:
    print("The means hospitalization charges for smokers and non smokers are equal")
```
```
We reject the null hypothesis
The means hospitalization charges for smokers and non smokers are not equal. The mean hospitalization
charges for smokers and greater than non smokers
```

In [ ]:

In [89]:
```
#2) Prove (or disprove) with statistical evidence that the viral load of females is different from that of males

#We use two sided T test

# Null hypothesis : The mean (of) viral load of males and females is the same.

# Alternate hypothesis :  The mean (of) viral load of males and females are unequal i.e. not the same.

# Checkin the assumptions
```

In [90]:
```
df.groupby('sex')['viral load'].describe()
```

Out[90]:

| sex | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **0** | 581.0 | 10.030947 | 1.94495 | 5.32 | 8.600 | 9.940 | 11.2900 | 15.51 |
| **1** | 610.0 | 9.966541 | 1.96940 | 5.60 | 8.585 | 9.855 | 11.1725 | 15.58 |

In [105…
```
# male viral load
mvl =  df[ df['sex'] == 0]['viral load'].sample(581)

# female viral load
fvl = df[ df['sex'] == 1]['viral load'].sample(581)

#viral load irrespective of gender
tvl = df['viral load']
```

In [106…
```
mvl.reset_index(drop=True)
fvl.reset_index(drop=True)
```

Out[106…
```
0       9.47
1      12.28
2       9.97
3      14.13
4       7.22
       ...
576     9.70
577    10.54
578    11.97
579     5.76
580     9.69
Name: viral load, Length: 581, dtype: float64
```

In [107…
```
stats.shapiro(mvl)
```

Out[107…
```
ShapiroResult(statistic=0.9929239749908447, pvalue=0.007620655465871096)
```

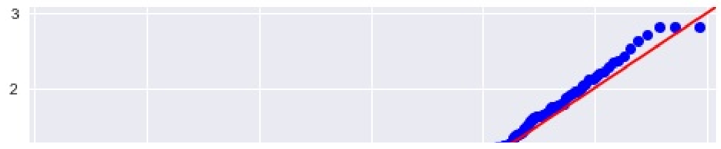
In [108…
```
stats.shapiro(fvl)
```

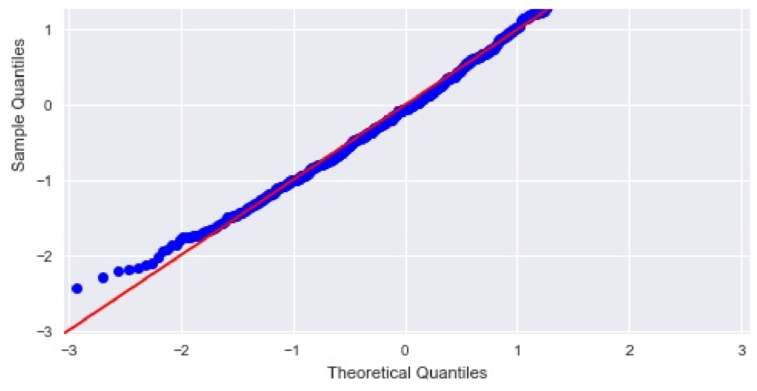
Out[108…
```
ShapiroResult(statistic=0.9918439388275146, pvalue=0.002799693727865815)
```

In [109…
```
stats.shapiro(tvl)
```

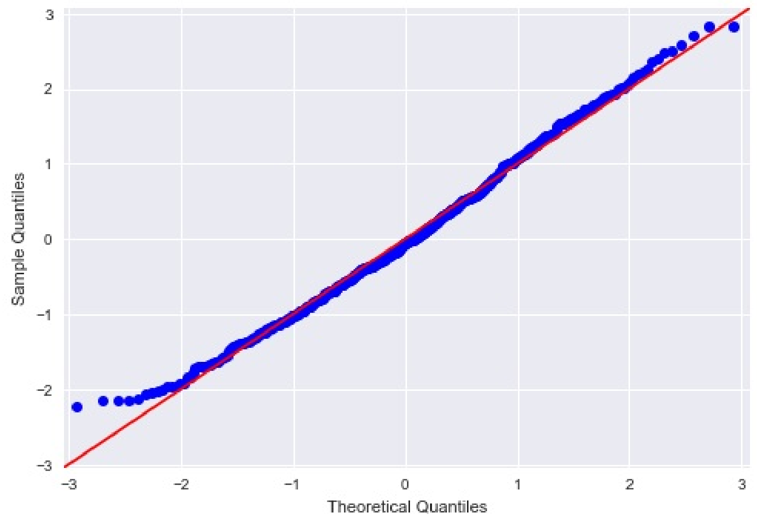
Out[109…
```
ShapiroResult(statistic=0.9924992918968201, pvalue=9.647324986872263e-06)
```

In [110…
```
#For both male viral load and female viral load the p value are much smaller than alpha 0.05 hence we know that t
#are NOT normally distributed. We further check this with q-q plot

qqplot(mvl,norm,fit=True,line="45")
plt.show()
```

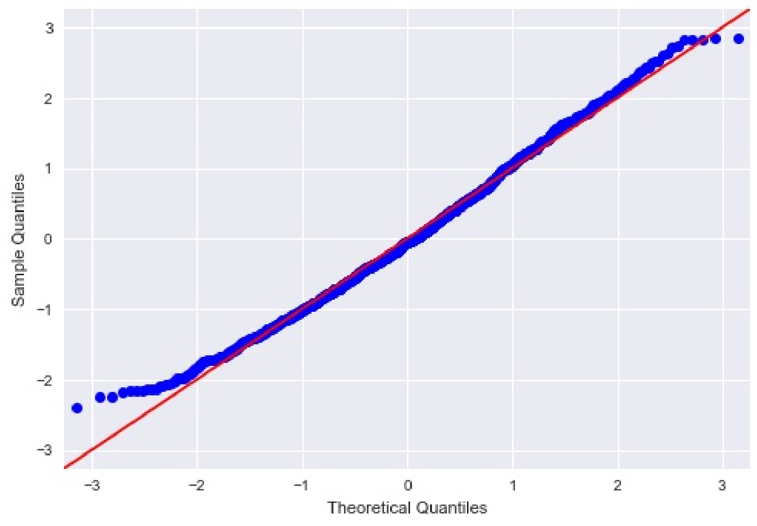




In [111…
```
qqplot(fvl,norm,fit=True,line="45")
plt.show()
```



In [112…
```
qqplot(tvl,norm,fit=True,line="45")
plt.show()
```



In [98]:
```
#From q-q- plots we can see that male as well as female viral loads are not normally distributed. Viral load irre
# gender is also not normally distributed
```

In [113…
```
stats.shapiro(np.log(mvl))
```

Out[113…
```
ShapiroResult(statistic=0.9944441914558411, pvalue=0.03279054909944534)
```

In [114…
```
stats.shapiro(np.log(fvl))
```

Out[114…
```
ShapiroResult(statistic=0.9927094578742981, pvalue=0.0062293680384755135)
```

In [ ]:
```
#Even their logs are not normally distributed hence this isn't a log normal distribution
```

In [189…
```
# We use levene's test to check for equal variances
w_stats, p_value = levene(mvl, fvl, center="median")
```

In [190…
```
if p_value > alpha :
  print("Variances of viral loads of males and females are equal")
else:
  print("Variances of viral loads of males and females are NOT equal")
```
```
Variances of viral loads of males and females are equal
```

In [105…
```
#Variances are equal.. lets confirm by calcualting the variances indivudually as well

np.var(mvl)
```
Out[105…
```
3.776321134846738
```

In [106…
```
np.var(fvl)
```
Out[106…
```
3.9323172007429767
```

In [109…
```
#Two sided T test on male viral load and female viral load

t_test(mvl, fvl, 'two-sided')
```
Out[109…
```
0.6826719822745506
```

In [110…
```
stats.ttest_ind(mvl, fvl, alternative='two-sided')
```
Out[110…
```
Ttest_indResult(statistic=0.4089225510074076, pvalue=0.6826719822745506)
```

In [ ]:
```
#p value is 0.68 hence we fail to reject the Null hypothesis
```

In [116…
```
#Is the proportion of smoking significantly different across different regions?

#We use a Chi sqaured test

#Null hpyothesis (H0) : Smoker and regions are independent variables i.e. mutually independent. There is no relat

#Alternate hypothesis (H1) : Both these variables are dependent

contingency= pd.crosstab(df['region'], df['smoker'])
contingency
```
Out[116…

| smoker | 0 | 1 |
|---|---|---|
| **region** | | |
| **northeast** | 256 | 39 |
| **northwest** | 267 | 38 |
| **southeast** | 267 | 34 |
| **southwest** | 265 | 25 |

In [117…
```
from scipy.stats import chi2_contingency


stat, p, dof, expected = chi2_contingency(contingency, correction = False)
```

In [118…
```
p
```
Out[118…
```
0.31791538258247426
```
In [119…
```
if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (Fail to reject H0)')
```
```
Independent (Fail to reject H0)
```
In [ ]:
```

```
In [140…
```
#Is the mean viral load of women with 0 Severity level, 1 Severity level, and 2 Severity level the same?

# H0 (Null Hypothesis): The mean viral load of women with 0 Severity level, 1 Severity level, and 2 Severity leve

# H1 (Alternate Hypothesis) : The mean viral loads of women with severity level 0, 1 and 2 are not the same

df.groupby(['sex', 'severity level'])['viral load'].mean()
```
Out[140…
```
sex  severity level
0    0                 10.034880
     1                 10.042113
     2                 10.026162
     3                 10.023881
     4                 10.239231
     5                  9.598000
1    0                  9.963209
     1                  9.908844
     2                  9.945000
     3                 10.014366
     4                 10.601000
     5                 10.206250
Name: viral load, dtype: float64
```
In [120…
```
#We can observe that the mean viral load of women with 0 Severity level, 1 Severity level, and 2 Severity level i
#Confirming the same with Anova
```
In [121…
```
df[df['sex'] == 1].groupby('severity level')['viral load'].describe()
```
Out[121…

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **severity level** | | | | | | | | |
| **0** | 268.0 | 9.963209 | 1.936862 | 5.76 | 8.6075 | 9.695 | 11.1300 | 14.92 |
| **1** | 147.0 | 9.908844 | 1.918987 | 5.60 | 8.6000 | 9.670 | 11.1300 | 15.36 |
| **2** | 106.0 | 9.945000 | 2.092305 | 5.73 | 8.3450 | 10.060 | 11.2025 | 15.57 |
| **3** | 71.0 | 10.014366 | 1.950361 | 6.33 | 8.5950 | 10.030 | 11.1750 | 14.90 |
| **4** | 10.0 | 10.601000 | 1.815063 | 8.53 | 9.6200 | 9.825 | 11.0350 | 13.82 |
| **5** | 8.0 | 10.206250 | 2.975480 | 6.10 | 8.0500 | 10.080 | 11.6625 | 15.58 |

In [151…
```
dfws = df[(df['sex'] == 1) &  (df['severity level'] <=2)]
```
In [154…
```
dfws.reset_index(drop=True)
```
Out[154…

| | Index | age | sex | smoker | region | viral load | severity level | hospitalization charges | age bins |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 19 | 1 | 1 | southwest | 9.30 | 0 | 42212 | (18, 28] |
| **1** | 5 | 31 | 1 | 0 | southeast | 8.58 | 0 | 9392 | (28, 38] |
| **2** | 6 | 46 | 1 | 0 | southeast | 11.15 | 1 | 20601 | (38, 48] |
| **3** | 9 | 60 | 1 | 0 | northwest | 8.61 | 0 | 72308 | (58, 68] |
| **4** | 11 | 62 | 1 | 1 | southeast | 8.76 | 0 | 69522 | (58, 68] |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **516** | 1331 | 23 | 1 | 0 | southwest | 11.13 | 0 | 26990 | (18, 28] |
| **517** | 1334 | 18 | 1 | 0 | northeast | 10.64 | 0 | 5515 | (0, 18] |
| **518** | 1335 | 18 | 1 | 0 | southeast | 12.28 | 0 | 4075 | (0, 18] |
| **519** | 1336 | 21 | 1 | 0 | southwest | 8.60 | 0 | 5020 | (18, 28] |
| **520** | 1337 | 61 | 1 | 1 | northwest | 9.69 | 0 | 72853 | (58, 68] |

521 rows × 9 columns

In [178…

```
#Checking assumption of Anova that each group sample is drawn from a normally distributed population

#for females taking into account the severity level 0,1, and 2 the viral load isn't normally distributed
stats.shapiro(dfws['viral load'])
```

Out[178…

```
ShapiroResult(statistic=0.9920216202735901, pvalue=0.006814346183091402)
```

In [180…

```
#for females taking into account all severity levels the viral load isn't normally distributed
stats.shapiro(df[df['sex'] == 1]['viral load'])
```

Out[180…

```
ShapiroResult(statistic=0.9914260506629944, pvalue=0.0013309171190485358)
```

In [183…

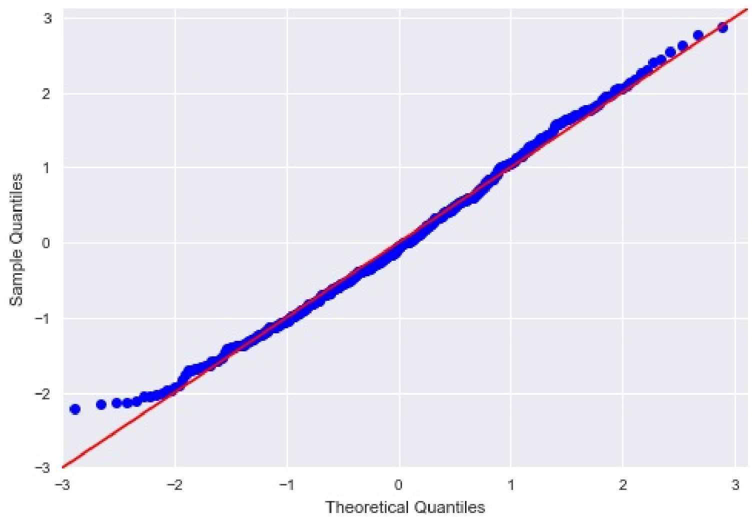
```
#female viral load for severity level 0,1 and 2 is not log normally distributed
stats.shapiro(np.log(dfws['viral load']))
```

Out[183…

```
ShapiroResult(statistic=0.9924635291099548, pvalue=0.009911485947668552)
```

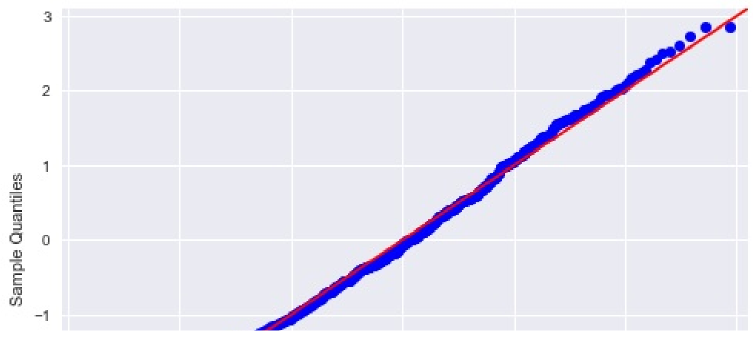
In [181…

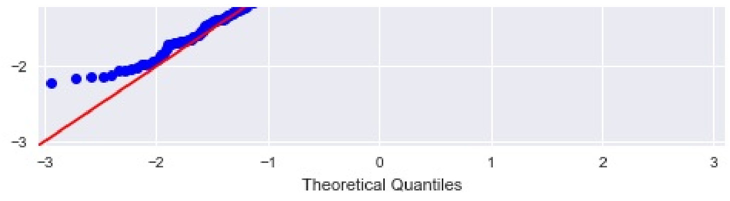
```
#We further confirm this qq plot

qqplot(dfws['viral load'],norm,fit=True,line="45")
plt.show()
```



In [182…

```
qqplot(df[df['sex'] == 1]['viral load'],norm,fit=True,line="45")
plt.show()
```





In [ ]:
```
#As is clear from the qq plot
# 1) The viral load for females with severity level 0,1 and 2 is not normally distributed
# 2) The viral load for females with taking into account all severity levels isn't normally distributed
```

In [186…
```
dws2 = []
for i in range(0,3):
    dws2i = df[(df["sex"] == 1) & (df['severity level'] == i)]["viral load"].sample(106)
    dws2.append(dws2i)
```

In [191…
```
#Checking for assumptions that all groups have same variance

w_stats, p_value = levene(dws2[0], dws2[1], dws2[2], center = 'mean')
```

In [192…
```
if p_value > alpha :
  print("Variances of viral load 0,1 and 2 are equal in females")
else:
  print("Variances of viral load 0,1 and 2 are NOT Equal in females")
```

```
Variances of viral load 0,1 and 2 are equal in females
```

In [194…
```
from scipy.stats import f_oneway


fstat, p_value = f_oneway(dws2[0], dws2[1], dws2[2])
```

In [196…
```
p_value
```

Out[196…
```
0.9561554023268642
```

In [197…
```
if p_value > alpha :
  print("Means of viral load 0,1 and 2 are equal in females. We accept the null hypothesis")
else:
  print("Means of viral load 0,1 and 2 are NOT Equal in females and we fail to reject the null hypothesis")
```

```
Means of viral load 0,1 and 2 are equal in females. We accept the null hypothesis
```

In [122…
```
df.groupby('region')['hospitalization charges'].mean()
```

Out[122…
```
region
northeast    26851.705085
northwest    25908.986885
southeast    23854.235880
southwest    22334.206897
Name: hospitalization charges, dtype: float64
```

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js