Buisness Case:Apollo Hospitals - Hypothesis Testing

In [1]:

```
#Importing required libraries
import pandas as pd
import numpy as np
import  matplotlib . pyplot  as  plt
import seaborn as sns
from  numpy  import  NaN , nan , NAN
import statsmodels.api as sm
import warnings
warnings.filterwarnings("ignore")
from scipy import stats
from scipy.stats import levene
```

In [2]:

```
df=pd.read_csv("C:\\Downloads\\scaler_apollo_hospitals.csv")
```

In [3]:

```
#Checking shape of Data
df.shape
```

Out[3]:

```
(1338, 8)
```

In [4]:

```
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   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
```

We have datatype of int,objects and float

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

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

In [6]:
```
df['Unnamed: 0']
```

Out[6]:
```
0          0
1          1
2          2
3          3
4          4
        ... 
1333    1333
1334    1334
1335    1335
1336    1336
1337    1337
Name: Unnamed: 0, Length: 1338, dtype: int64
```

As we can see Unnamed doesnt carry any information so we can drop this column

In [7]:
```
df.drop('Unnamed: 0', inplace=True, axis=1)
```

In [8]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   age                      1338 non-null   int64  
 1   sex                      1338 non-null   object 
 2   smoker                   1338 non-null   object 
 3   region                   1338 non-null   object 
 4   viral load               1338 non-null   float64
 5   severity level           1338 non-null   int64  
 6   hospitalization charges  1338 non-null   int64  
dtypes: float64(1), int64(3), object(3)
memory usage: 73.3+ KB
```

In [9]:

```
#Converting object data type into category
Object_Data = ['sex','smoker','region']
for i in Object_Data:
    df[i] = df[i].astype("category")
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column                   Non-Null Count  Dtype   
---  ------                   --------------  -----   
 0   age                      1338 non-null   int64   
 1   sex                      1338 non-null   category
 2   smoker                   1338 non-null   category
 3   region                   1338 non-null   category
 4   viral load               1338 non-null   float64 
 5   severity level           1338 non-null   int64   
 6   hospitalization charges  1338 non-null   int64   
dtypes: category(3), float64(1), int64(3)
memory usage: 46.3 KB
```

In [10]:

```
#Analysing the Basic Matrix
df.describe(include = np.number )
```

Out[10]:

	age	viral load	severity level	hospitalization charges
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	10.221233	1.094918	33176.058296
std	14.049960	2.032796	1.205493	30275.029296
min	18.000000	5.320000	0.000000	2805.000000
25%	27.000000	8.762500	0.000000	11851.000000
50%	39.000000	10.130000	1.000000	23455.000000
75%	51.000000	11.567500	2.000000	41599.500000
max	64.000000	17.710000	5.000000	159426.000000

Mean and Median of age is approximately same so we can say that data is not skewed.
Maximum frequency of people is from southeast region.

In [11]:

```
df.describe(include = 'category' )
```

Out[11]:

	sex	smoker	region
count	1338	1338	1338
unique	2	2	4
top	male	no	southeast
freq	676	1064	364

In [12]:

```
#Checking for NULL Values
df.isnull().sum()/len(df)*100
```

Out[12]:

```
age                        0.0
sex                        0.0
smoker                     0.0
region                     0.0
viral load                 0.0
severity level             0.0
hospitalization charges    0.0
dtype: float64
```

There are no values in dataset

In [13]:

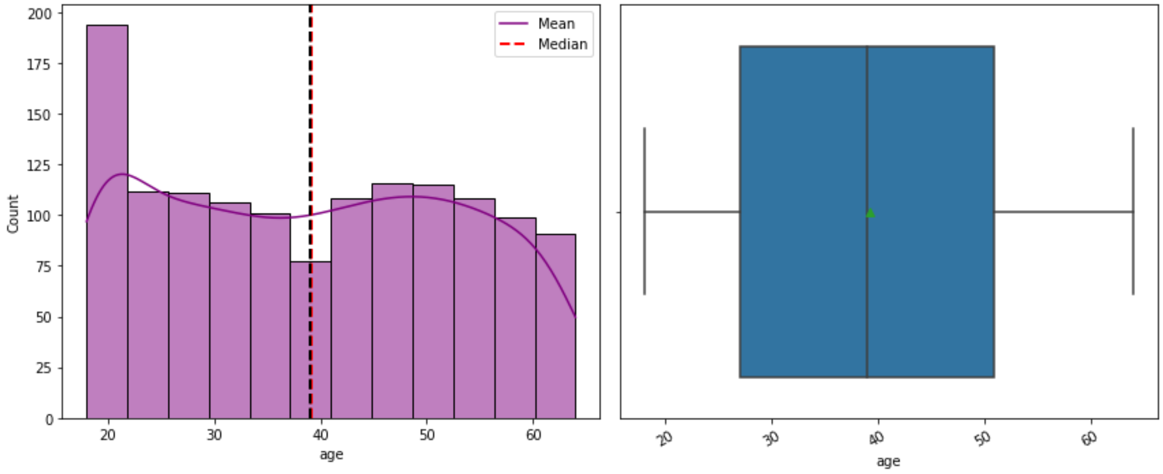
```
#Univariate Analysis for numerical features
def numerical_feature(col_data):
    fig,ax = plt.subplots(nrows=1,ncols=2,figsize=(12,5))
    sns.histplot(x = col_data, kde=True, ax=ax[0], color = 'purple')
    ax[0].axvline(col_data.mean(), color='r', linestyle='--',linewidth=2)
    ax[0].axvline(col_data.median(), color='k', linestyle='dashed', linewidth
    ax[0].legend({'Mean':col_data.mean(),'Median':col_data.median()})
    sns.boxplot(x=col_data, showmeans=True, ax=ax[1])
    plt.xticks(rotation = 30)
    plt.tight_layout()
    plt.show()
```

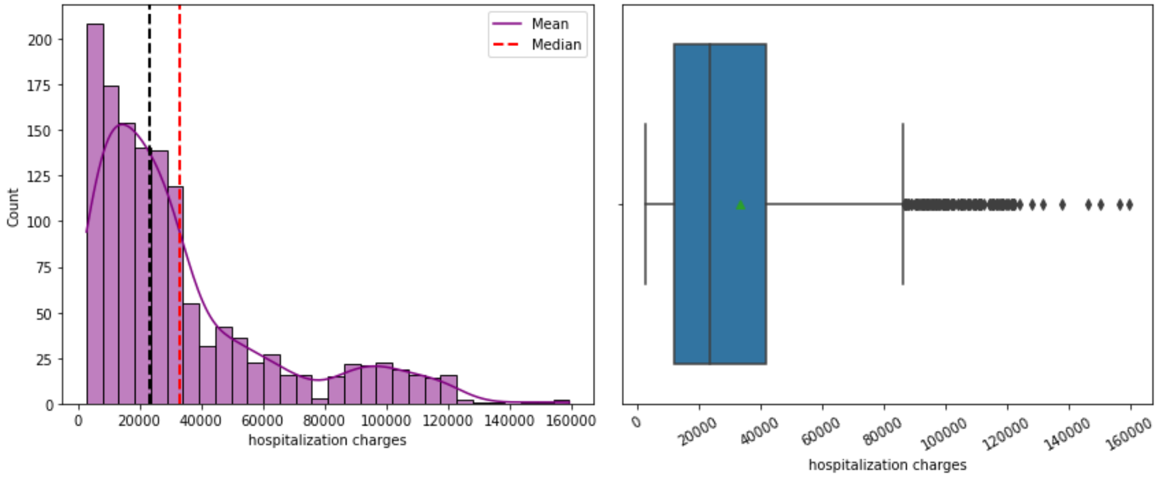
In [14]:

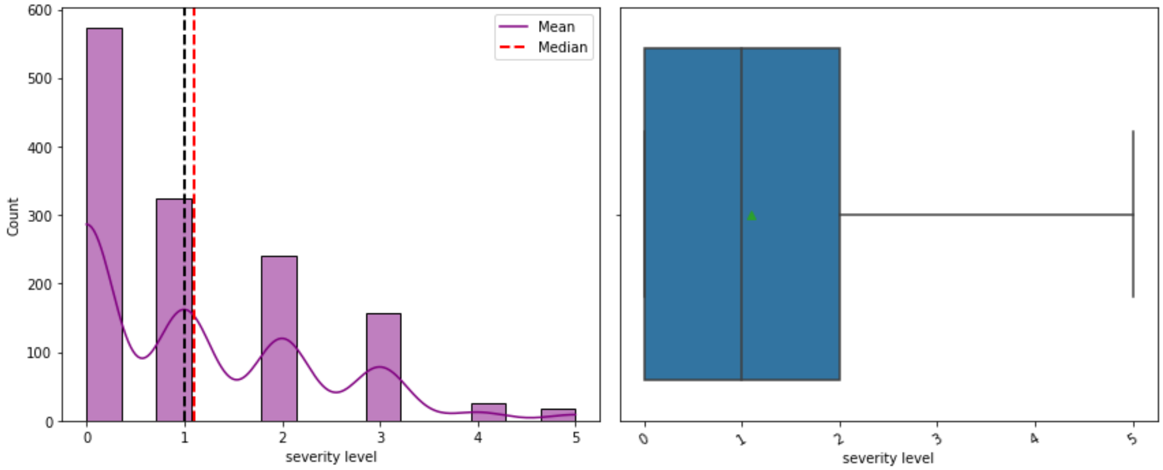
```
numerical_columns=['age', 'hospitalization charges','severity level']
```

In [15]:

```
for  i  in  numerical_columns :
    numerical_feature(df[i])
```







In [16]:

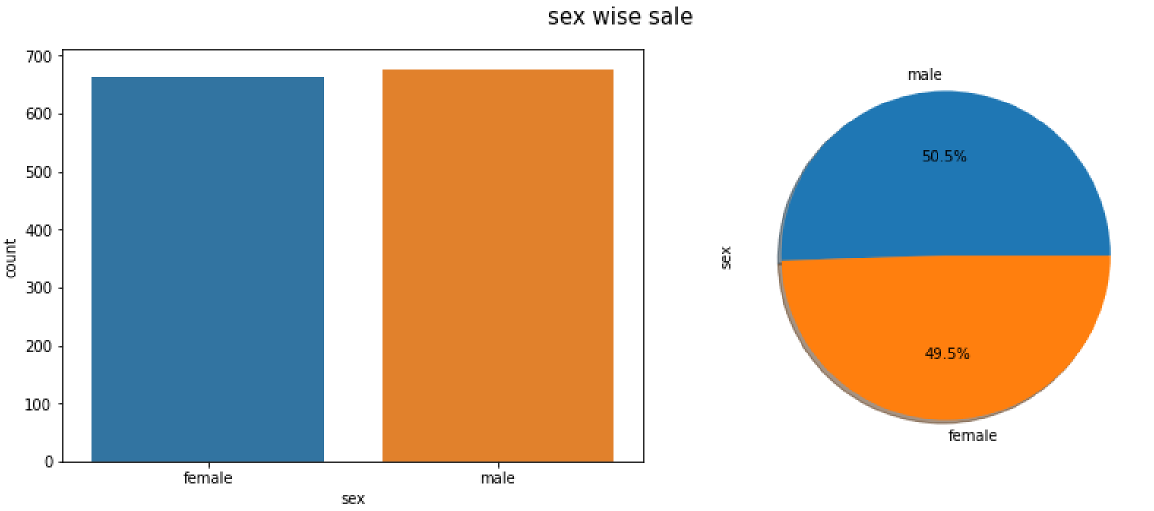
```
def categorical_features(col_data):
    fig,ax = plt.subplots(nrows=1,ncols=2,figsize=(12,5))
    fig.suptitle(col_data.name+' wise sale',fontsize=15)
    sns.countplot(col_data,ax=ax[0])
    col_data.value_counts().plot.pie(autopct='%1.1f%%',ax=ax[1], shadow = Tru
    plt.tight_layout()
```

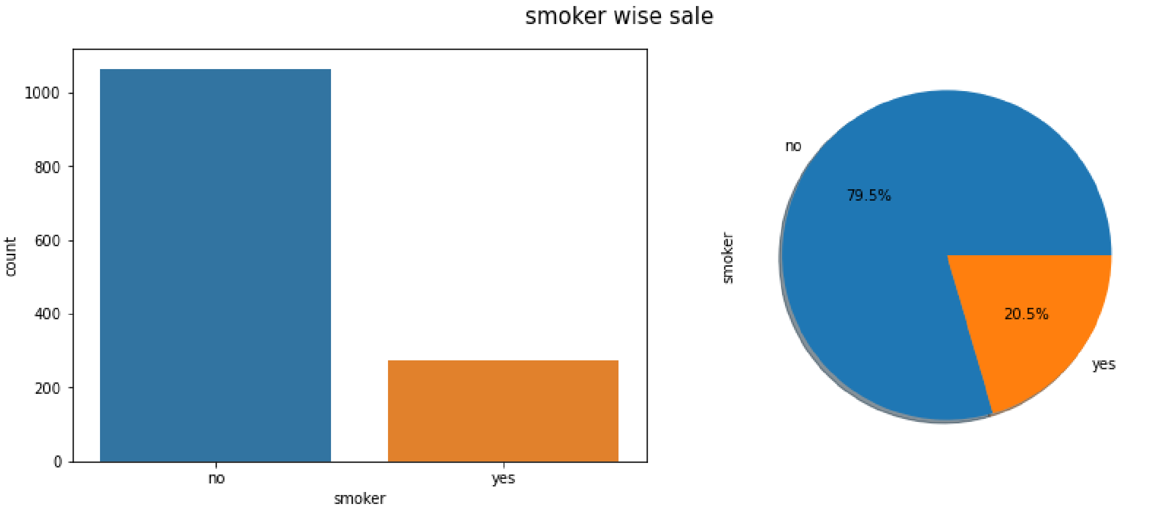
In [17]:

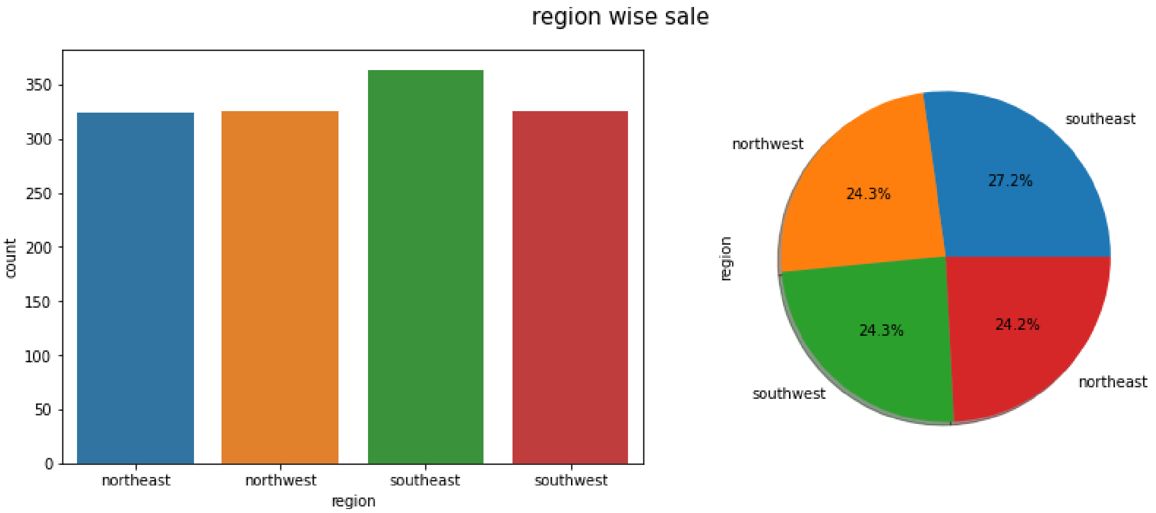
```
categorical_cols = ['sex', 'smoker', 'region']
```

In [18]:

```
for i in categorical_cols:
    categorical_features(df[i])
```



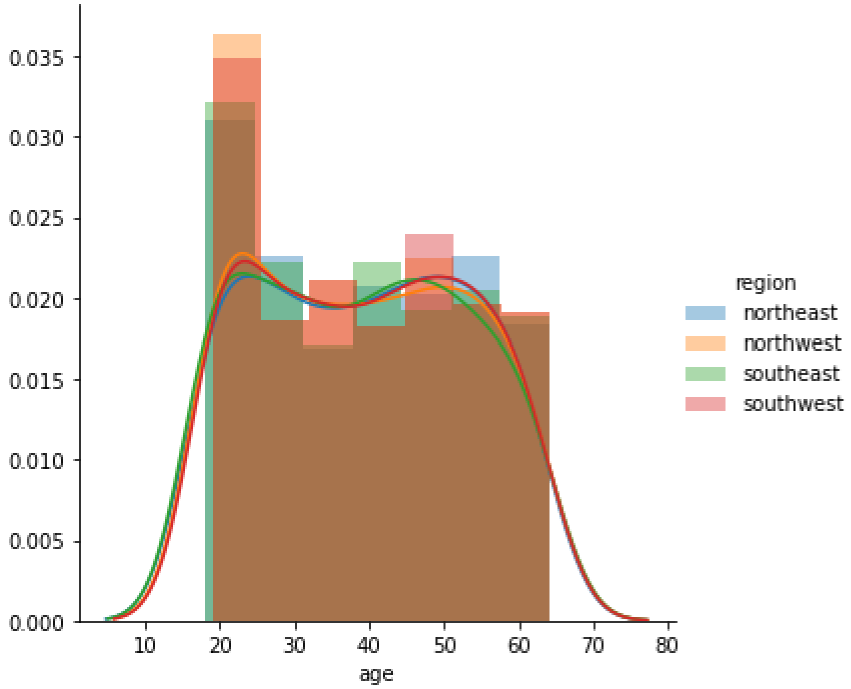




Male and Female addmited to the hospital are nearly same.
Out of total patients 79.5% are not smokers and rest are smokers.
Southeast has highest frequency and remaining regions has approxymatley same frequency.

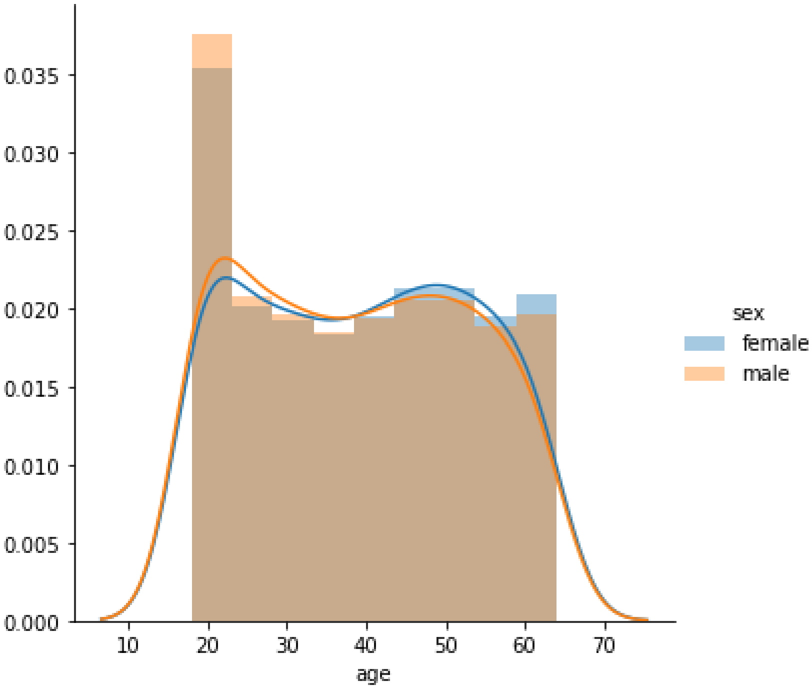
In [19]:

```
sns.FacetGrid(df,hue='region',size=5)\
   .map(sns.distplot,"age")\
   .add_legend();
plt.show()
```



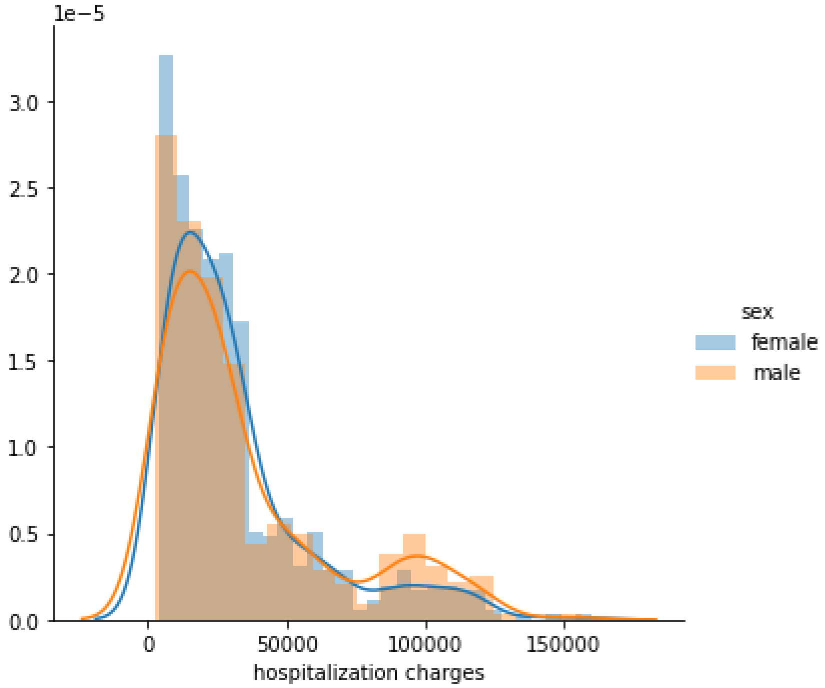
In [20]:

```
sns.FacetGrid(df,hue='sex',size=5)\
   .map(sns.distplot,"age")\
   .add_legend();
plt.show()
```



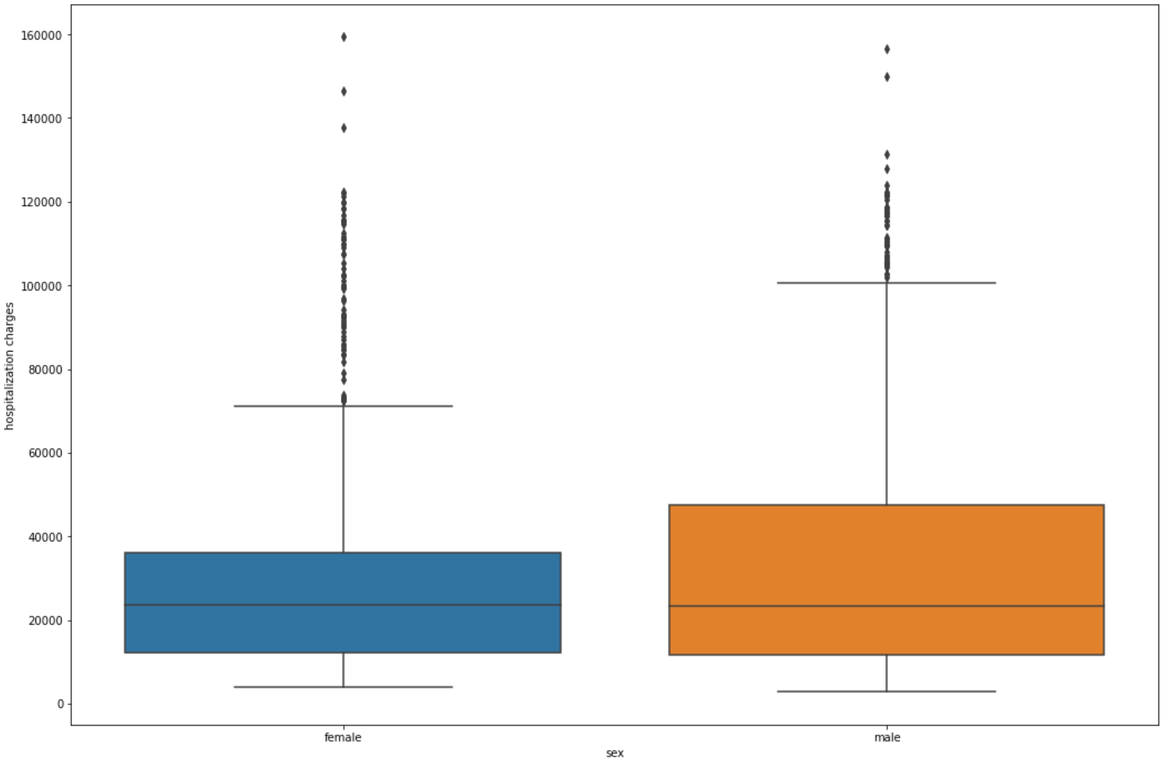
In [21]:

```
sns.FacetGrid(df,hue='sex',size=5)\
   .map(sns.distplot,"hospitalization charges")\
   .add_legend();
plt.show()
```



In [22]:

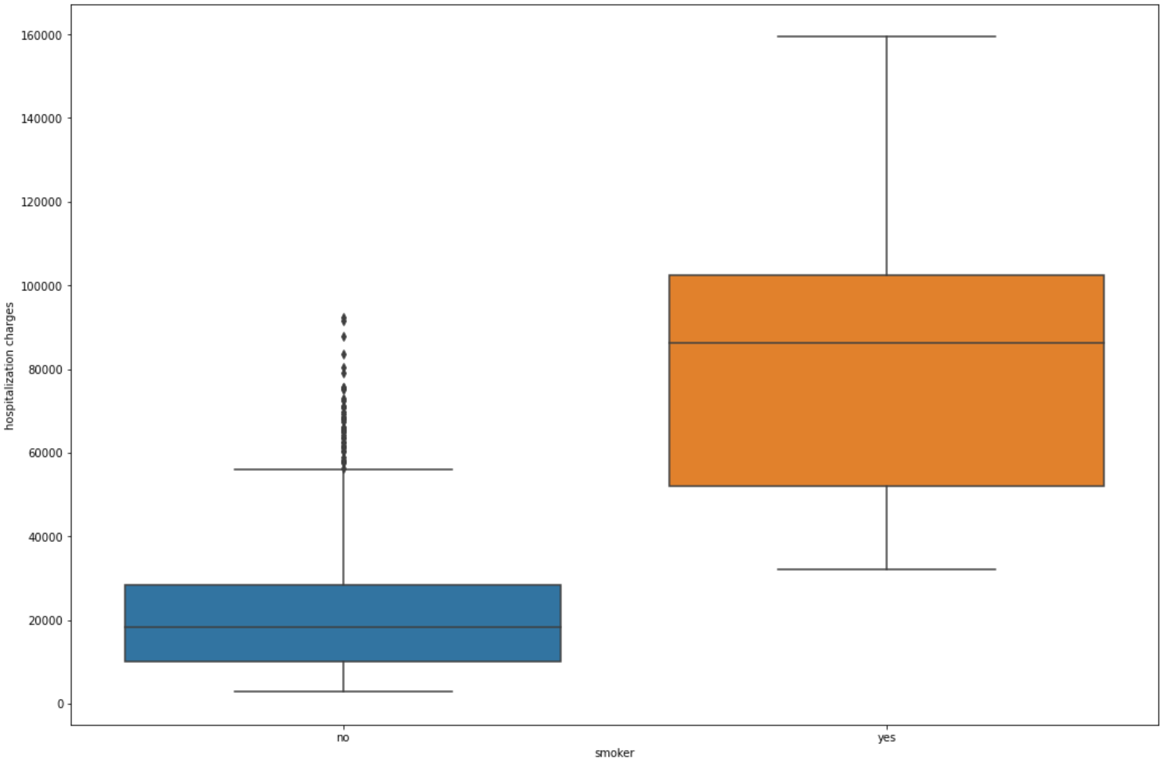
```
#Checking for outliers
plt.figure(figsize=(15,10))
sns.boxplot(x = 'sex', y = 'hospitalization charges', data = df)
plt.tight_layout(pad = 2)
```



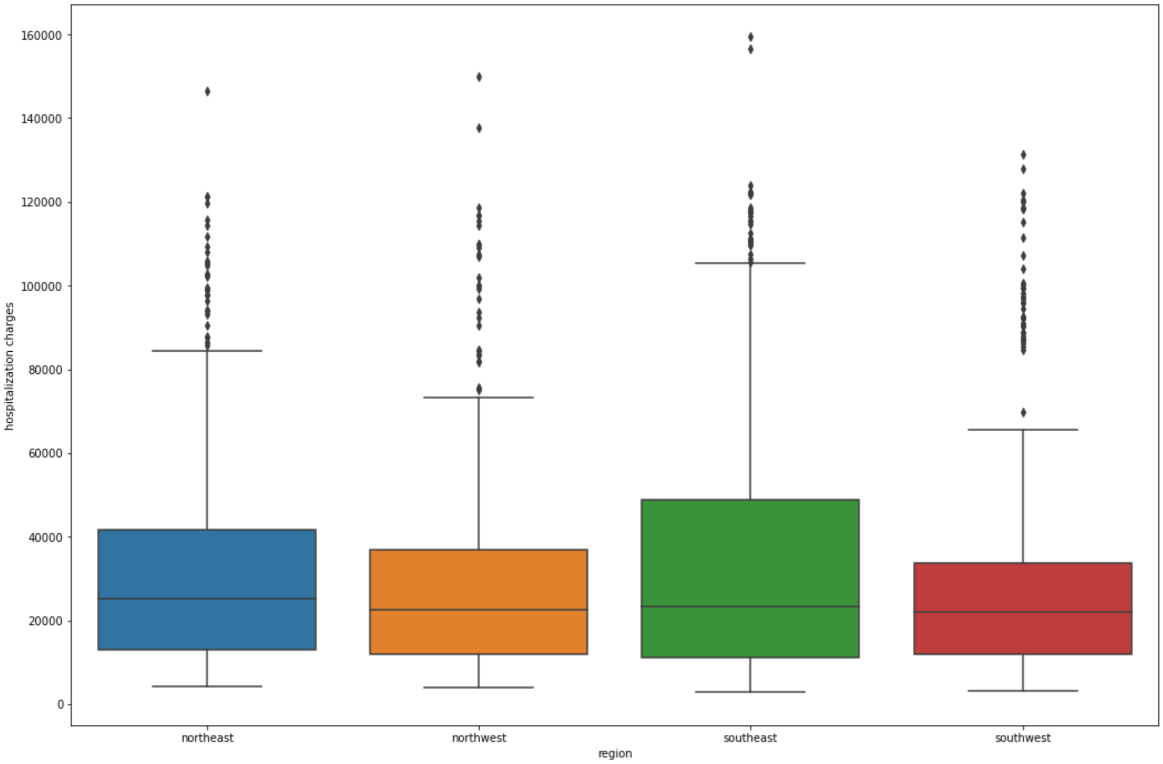
Median of charges seems to be similar visually we can verify same using hypothesis testing.

In [23]:

```
plt.figure(figsize=(15,10))
sns.boxplot(x = 'smoker', y = 'hospitalization charges', data = df)
plt.tight_layout(pad = 2)
```



In [24]:
```
plt.figure(figsize=(15,10))
sns.boxplot(x = 'region', y = 'hospitalization charges', data = df)
plt.tight_layout(pad = 2)
```



In [25]:
```
#Removing outliers
df1=df
numerical_cols = ['age', 'severity level','hospitalization charges']
q1 = df1[numerical_cols].quantile(0.25)
q3 = df1[numerical_cols].quantile(0.75)
iqr = q3 -q1
```

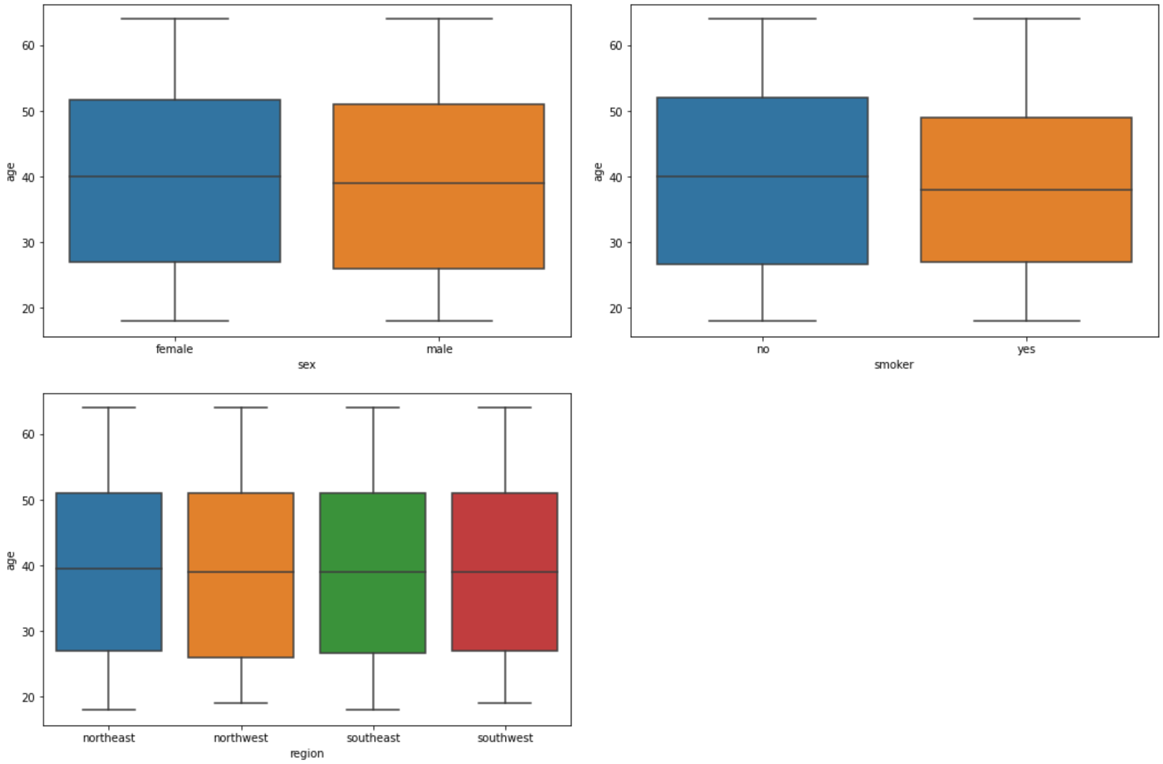
In [26]:
```
df1=df1[~((df1[numerical_cols]<q1-1.5*iqr) | (df1[numerical_cols]>q3+1.5*iqr)
df1= df1.reset_index(drop = True)
```

In [27]:
```
df1.shape[0] - df.shape[0]
```

Out[27]:
```
-139
```

In [28]:

```
plt.figure(figsize=(15,10))
for i,j in enumerate(categorical_cols):
    plt.subplot(2, 2, i+1)
    plt.subplots_adjust(hspace = 0.8)
    sns.boxplot(x = j, y = 'age', data = df)
    plt.tight_layout(pad = 2)
```



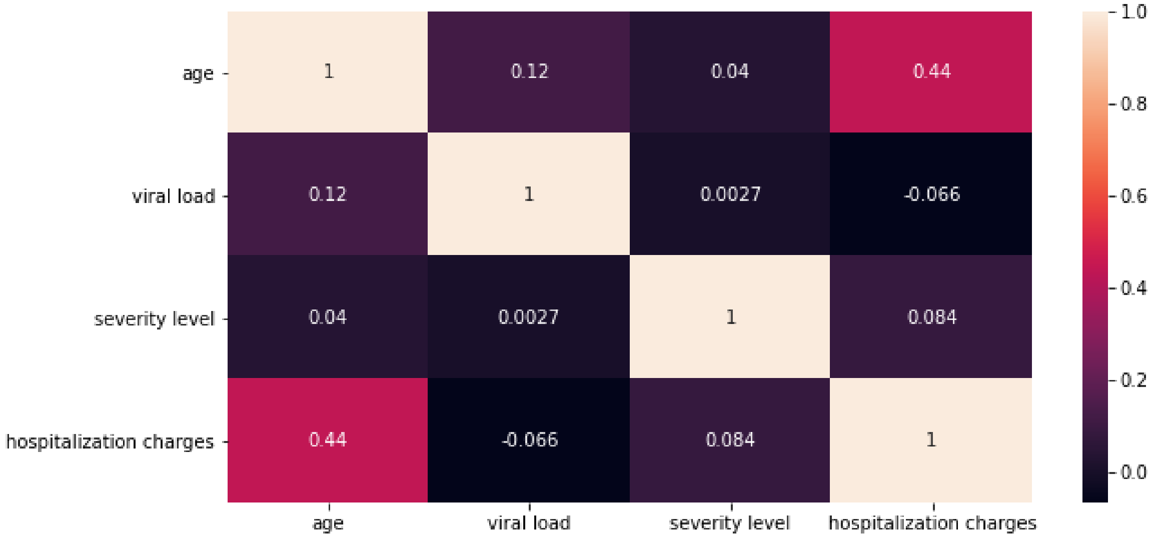
In [29]:

```
#Bivariate Analysis
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1199 entries, 0 to 1198
Data columns (total 7 columns):
 #   Column                   Non-Null Count  Dtype   
---  ------                   --------------  -----   
 0   age                      1199 non-null   int64   
 1   sex                      1199 non-null   category
 2   smoker                   1199 non-null   category
 3   region                   1199 non-null   category
 4   viral load               1199 non-null   float64 
 5   severity level           1199 non-null   int64   
 6   hospitalization charges  1199 non-null   int64   
dtypes: category(3), float64(1), int64(3)
memory usage: 41.5 KB
```

In [30]:

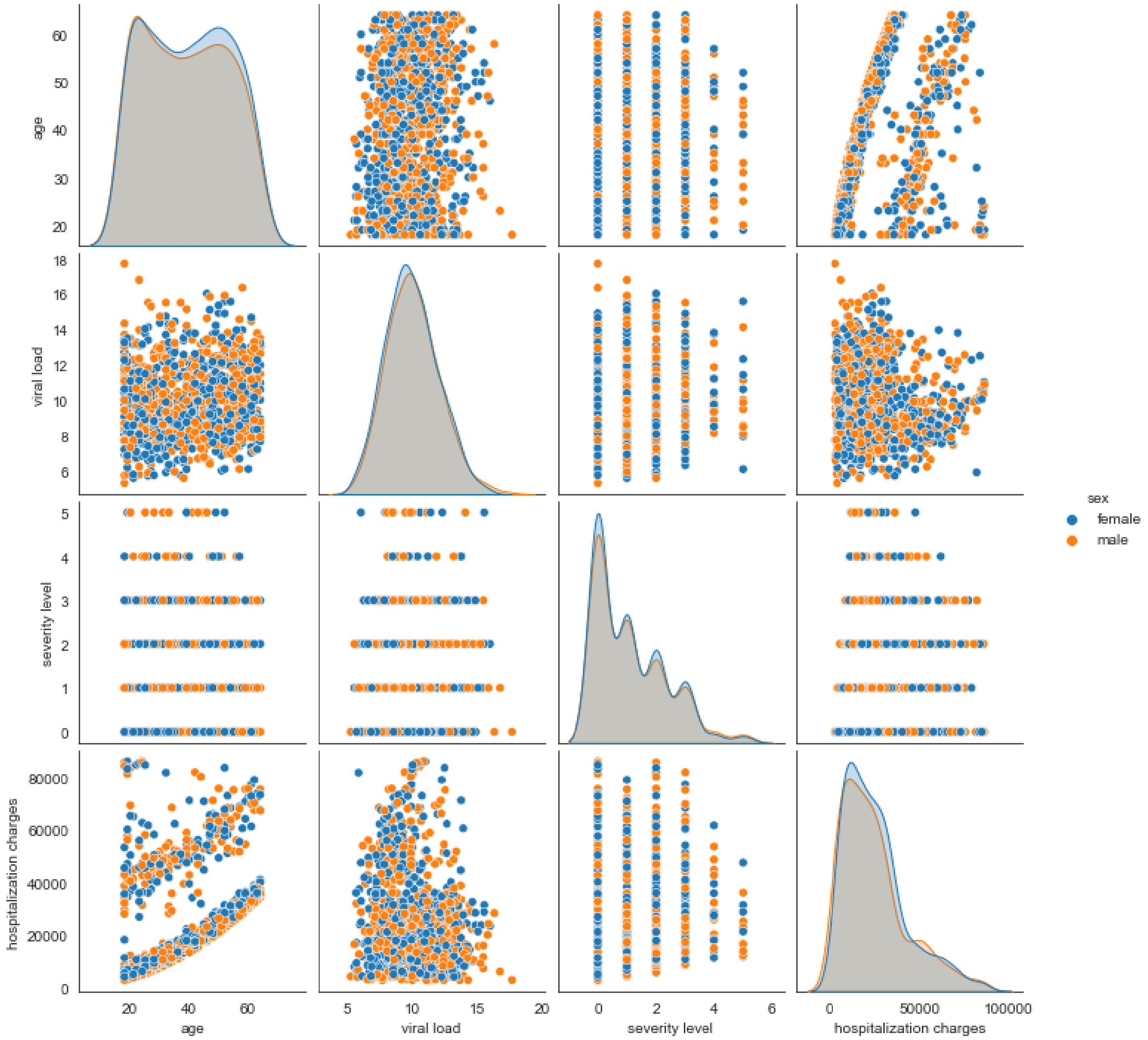
```
plt.figure(figsize = (10, 5))
sns.heatmap(df1.corr(),annot = True)
plt.yticks(rotation = 360)
plt.show()
```



Age and Hospitalization charges are correlated.
Severity level is not correlated

In [31]:

```
#Pair Plot
sns.set_style('white')
sns.pairplot(df1,hue='sex')
plt.show()
```



In [32]:

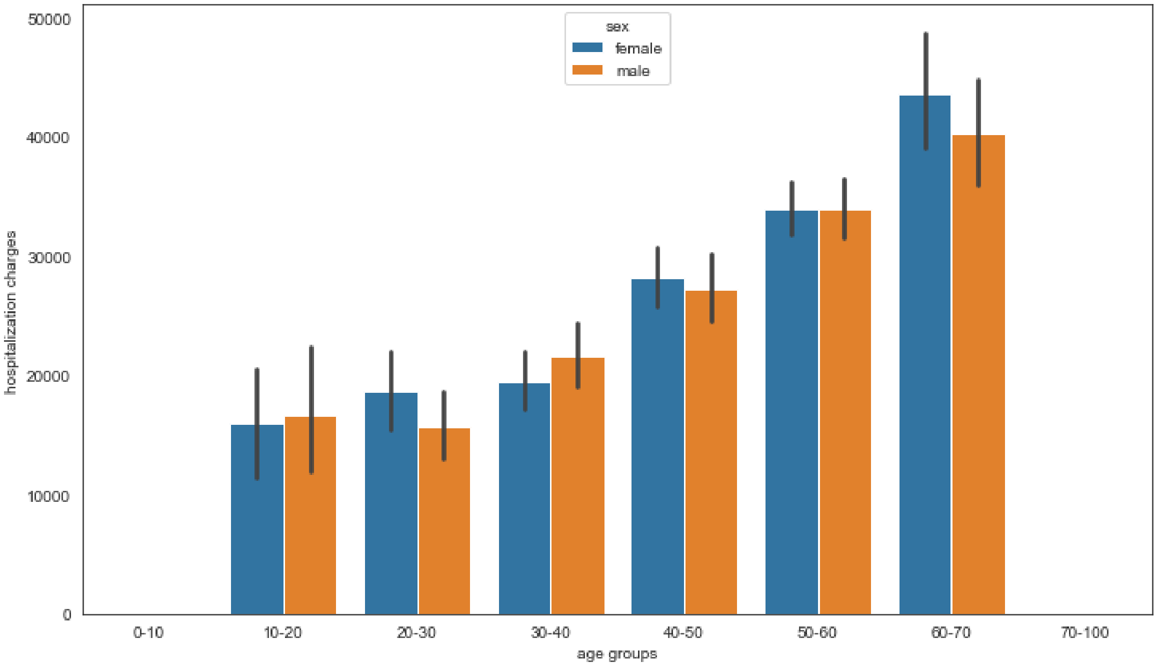
```
# Creating  age groups of persons.

bins = [0,10,20,30,40,50,60,70,100]
labels = ['0-10','10-20','20-30','30-40','40-50','50-60','60-70','70-100']
df1['age groups'] = pd.cut(x = df1['age'], bins = bins, labels = labels)
df1.head()
```

Out[32]:

	age	sex	smoker	region	viral load	severity level	hospitalization charges	age groups
0	19	female	yes	southwest	9.30	0	42212	10-20
1	18	male	no	southeast	11.26	1	4314	10-20
2	28	male	no	southeast	11.00	3	11124	20-30
3	33	male	no	northwest	7.57	0	54961	30-40
4	32	male	no	northwest	9.63	0	9667	30-40

In [33]:

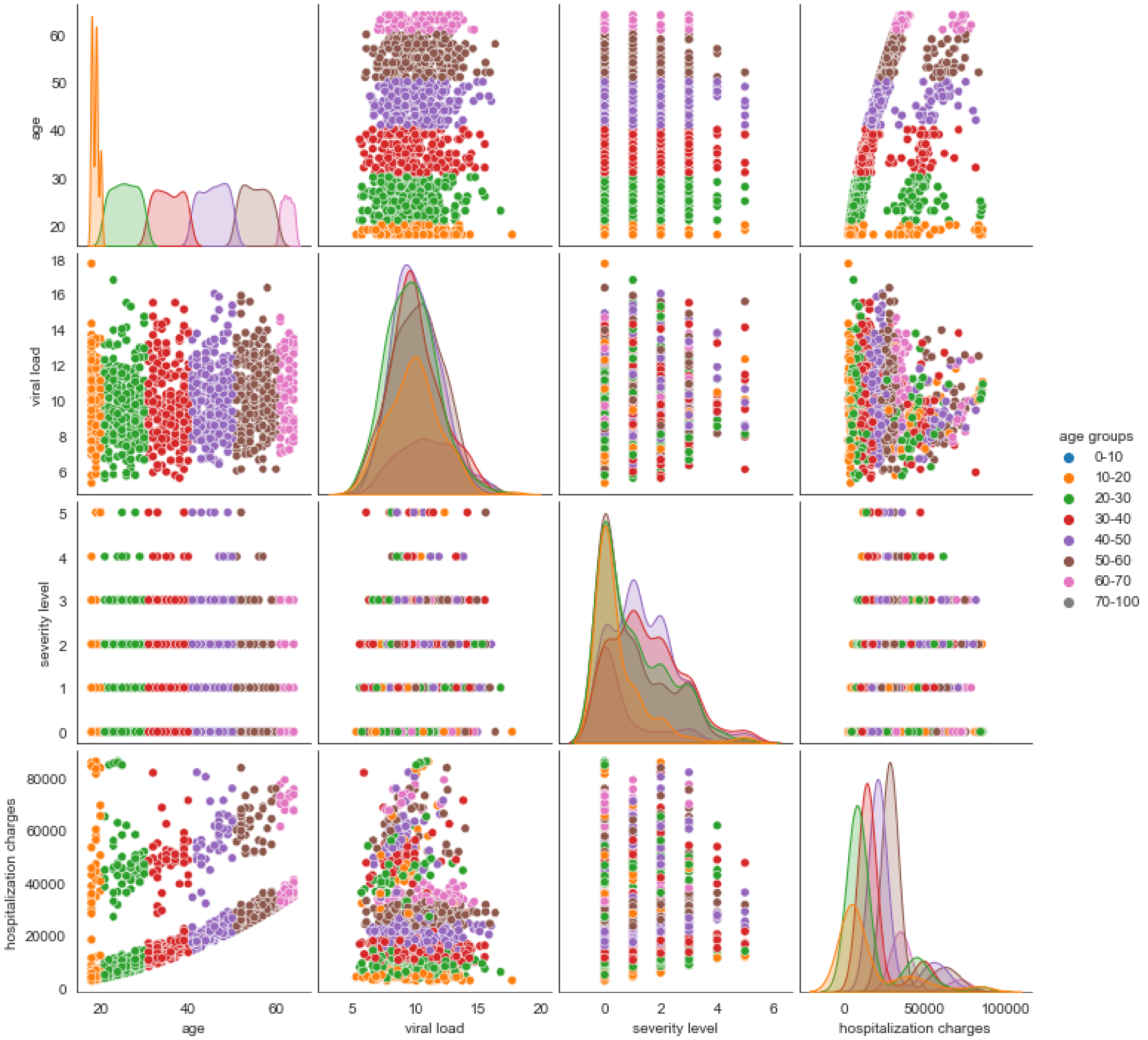
```
plt.figure(figsize=(12,7))
sns.barplot(x = 'age groups', y = 'hospitalization charges', data = df1, hue
plt.show()
```



As we can infer as ag e increases the hositalization charges increases.

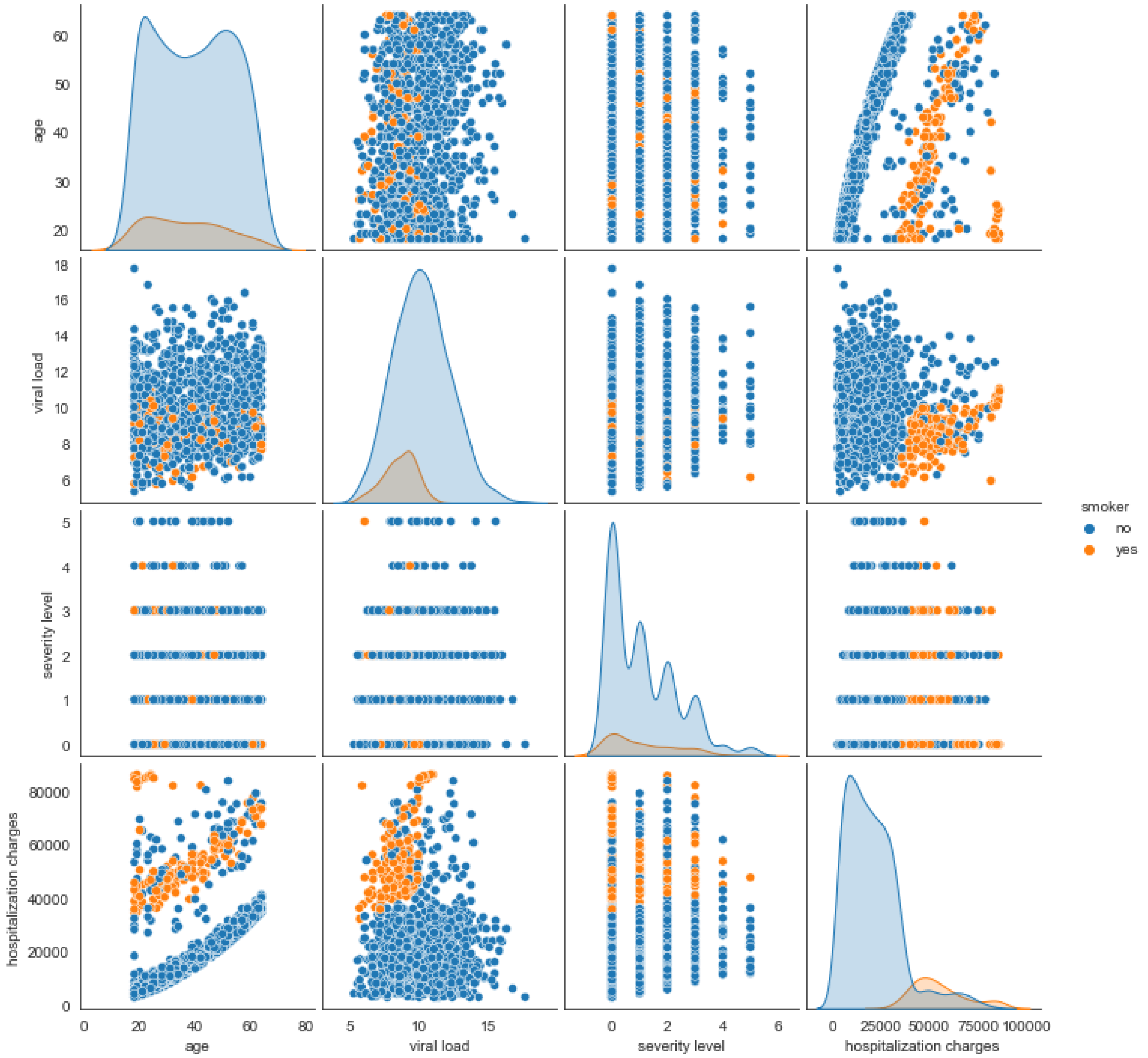
In [34]:

```
#Overview of data accordance with age group
#Pair Plot
sns.set_style('white')
sns.pairplot(df1,hue='age groups')
plt.show()
```



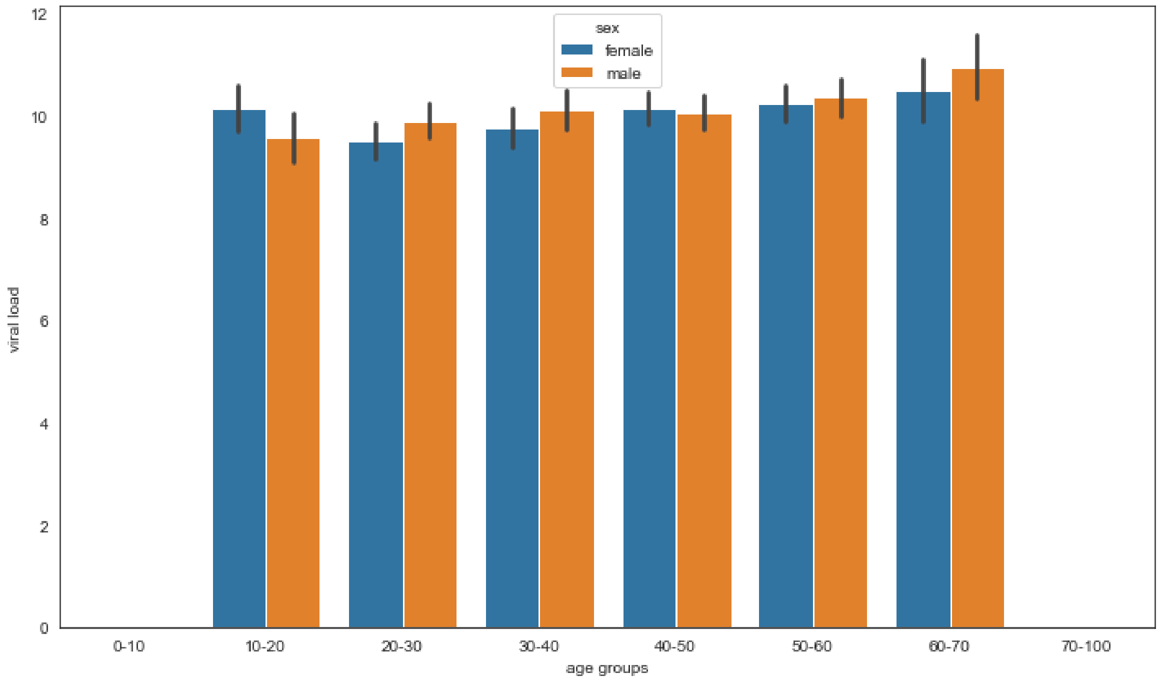
In [35]:

```
#Overview of data accordance with Smoker
#Pair Plot
sns.set_style('white')
sns.pairplot(df1,hue='smoker')
plt.show()
```



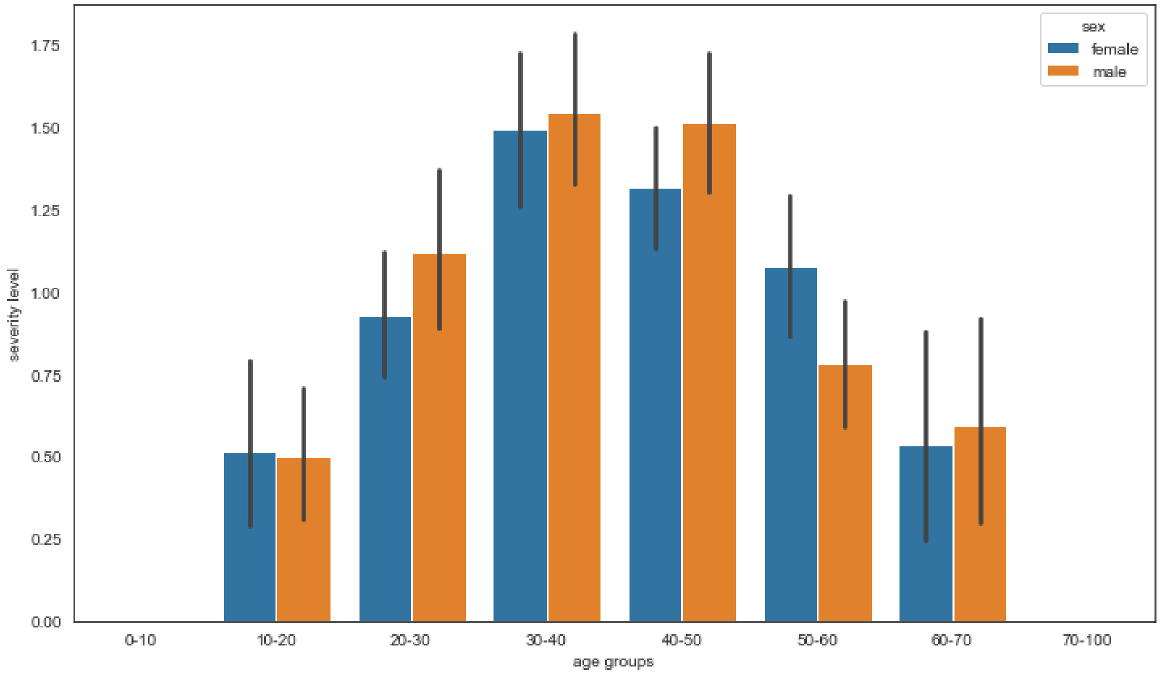
There are high imbalane of data for smoker and non smoker.
Patient who is smoker tends to pay more charges than non smoker.

In [36]:
```
plt.figure(figsize=(12,7))
sns.barplot(x='age groups',y='viral load',data=df1,hue='sex')
plt.show()
```



From above graph we can conclude, Viral load has nothing to do we age group.

In [37]:
```
plt.figure(figsize=(12,7))
sns.barplot(x='age groups',y='severity level',data=df1,hue='sex')
plt.show()
```



> We can observe severity level is higher in age group 30-50

Prove (or disprove) that the hospitalization of people who do smoking is greater than those who don't? (T-test Right tailed)

Two Sample t-test assumption.

> 1)Data values must be independent. Measurements for one observation do not affect measurements for any other observation.
> 2)Data in each group must be obtained via a random sample from the population.
> 3)Data in each group are normally distributed.
> 4)Data values are continuous.
> 5)The variances for the two independent groups are equal.

In [38]:

```
df1.shape
```

Out[38]:

```
(1199, 8)
```

In [39]:

```
df1.groupby(['smoker'])['hospitalization charges'].describe()
```

Out[39]:

	count	mean	std	min	25%	50%	75%	max
smoker								
no	1061.0	20889.284637	14541.903769	2805.0	9955.0	18344.0	28383.0	83680.0
yes	138.0	55035.586957	13792.707698	32074.0	44784.5	52197.0	62048.5	86182.0

In [40]:

```
smoker = df1[df1['smoker']== 'yes'] ['hospitalization charges'].sample(200,re
non_smoker = df1[df1['smoker']== 'no'] ['hospitalization charges'].sample(200
```

In [41]:

```
def shapiro_normality_check(series,alpha=0.05):
    a,p_value = stats.shapiro(series)
    print("Statistics",a, "p-value",p_value)

    # If p-value is not less than 0.05 then we fail to reject the null hypothe
    # If p-value is less than .05, we reject the null hypothesis.
    if p_value < alpha:
        print("We have sufficient evidence to say that the sample data does n
    else:
        print("We do not have sufficient evidence to say that sample data doe
```

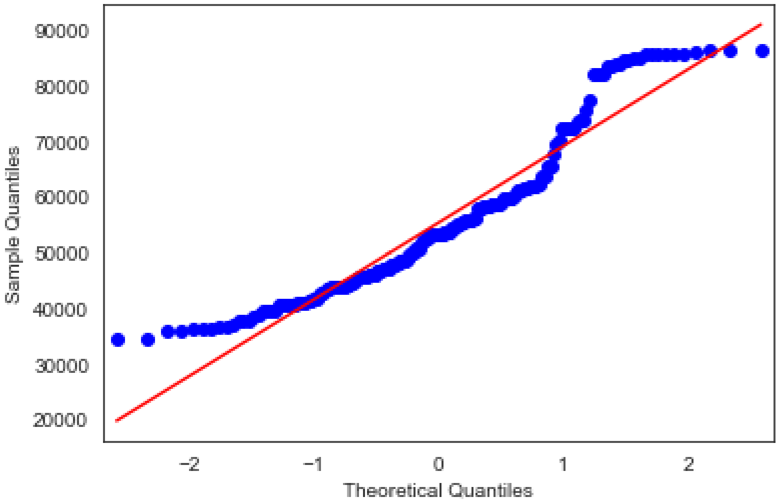
In [42]:
```
shapiro_normality_check(smoker)
```

```
Statistics 0.9164139032363892 p-value 3.2035083474823978e-09
We have sufficient evidence to say that the sample data does not come from
 a normal distribution
```

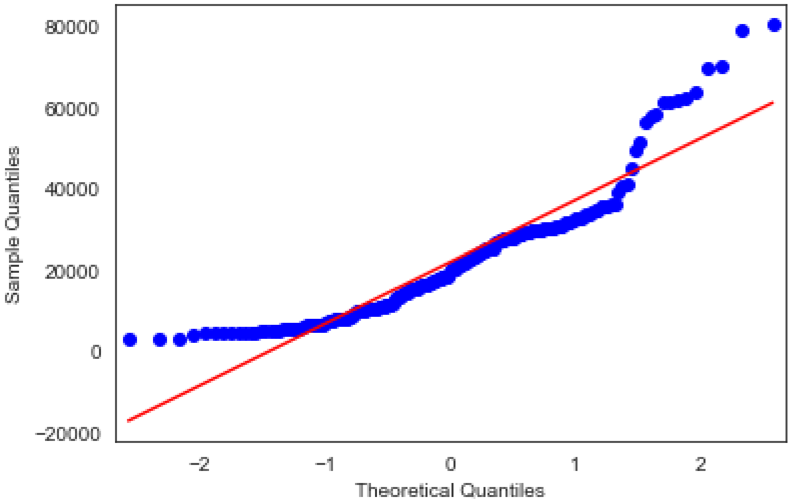
In [43]:
```
shapiro_normality_check(non_smoker)
```

```
Statistics 0.8812834620475769 p-value 1.8823820974178673e-11
We have sufficient evidence to say that the sample data does not come from
 a normal distribution
```

In [44]:
```
#Normality test using QQ plot
import statsmodels.api as sm
sm.qqplot(smoker, line = 's')
plt.show()
```



In [45]:
```
sm.qqplot(non_smoker, line = 's')
plt.show()
```



In [46]:

```
def  levene_var_check ( sample1 , sample2 , alpha = 0.05 ):
    a, p_value = stats.levene(sample1, sample2)
    print("p value = ", p_value)
    if p_value < alpha:
        print('We have sufficient evidence to say that the sample data does n
    else:
        print('We do not have sufficient evidence to say that the sample data
```

In [47]:

```
levene_var_check(smoker,non_smoker)
```

```
p value =  0.4315574308255927
We do not have sufficient evidence to say that the sample data does not hav
e equal variance.
```

In [48]:

```
alpha = 0.05
```

In [49]:

```
t_stat , p_value  =  stats . ttest_ind ( smoker , non_smoker , equal_var = Fa

onetail_pvalue = p_value/2
print("Test statistics = {},P value = {}, One Tail P-value = {}".format(t_sta
```

```
Test statistics = 22.967890698163973,P value = 9.936294041876319e-75, One T
ail P-value = 4.968147020938159e-75
```

In [50]:

```
if onetail_pvalue < alpha:
    print("P-value {} is less that alpha {}".format(onetail_pvalue,alpha))
    print("We have sufficient evidence to reject the Null hypothesis that Ave
else:
    print("P-value {} is greater that alpha {}".format(onetail_pvalue,alpha))
    print("We do not have sufficient evidence to reject the Null hypothesis t
```

```
P-value 4.968147020938159e-75 is less that alpha 0.05
We have sufficient evidence to reject the Null hypothesis that Average char
ges of smokers is less than or equal to non-smoker
```

In [51]:

```
t_stat,p_value = stats.ttest_ind(smoker,non_smoker,alternative='greater',equa
```

In [52]:

```
print("Test statistics = {} , One Tailed P-value = {} as specified that the a
```

```
Test statistics = 22.967890698163973 , One Tailed P-value = 4.9681470209381
59e-75 as specified that the alternative equal greater which means one tail
ed test
```

Prove (or disprove) with statistical evidence that the viral load of females is different from that of males (10 Points)

In [53]:
```
df1.groupby('sex')['viral load'].describe()
```

Out[53]:

sex	count	mean	std	min	25%	50%	75%	max
female	612.0	9.978186	1.981809	5.60	8.595	9.86	11.1875	16.02
male	587.0	10.087700	2.026193	5.32	8.610	9.94	11.3550	17.71

In [54]:
```
female_viral_load = df1[df1['sex'] == 'female']['viral load']
male_viral_load = df1[df1['sex'] == 'male']['viral load']
```

In [55]:
```
female_viral_load.shape[0],male_viral_load.shape[0]
```

Out[55]:
```
(612, 587)
```

In [56]:
```
female_viral_load_sample = df1[df1['sex'] == 'female']['viral load'].sample(5
male_viral_load_sample = df1[df1['sex'] == 'male']['viral load'].sample(500,
```

In [57]:
```
#Checking Variance
round(female_viral_load_sample.std()**2,2), round(male_viral_load_sample.std(
```

Out[57]:
```
(3.76, 4.24)
```

Normality Test:

> We will perform normality check using Shapiro test.
> The hypothesis of this test are:

> Null Hypothesis Ho - series is normal
> Alternative Hypothesis Ha - series is not normal

In [58]:
```
from scipy.stats import shapiro
def normality_check(series, alpha=0.05):
    _, p_value = shapiro(series)
    print(f'p value = {p_value}')
    if p_value >= alpha:
        print('We fail to reject the Null Hypothesis Ho')
    else:
        print('We reject the Null Hypothesis Ho')
```

In [59]:

```
normality_check(female_viral_load_sample)
print('-'*50)
normality_check(male_viral_load_sample)
```

```
p value = 8.879068627720699e-05
We reject the Null Hypothesis Ho
--------------------------------------------------
p value = 9.073790715774521e-05
We reject the Null Hypothesis Ho
```

In [60]:

```
from scipy.stats import levene
def variance_check(series1, series2, alpha=0.05):
    _, p_value = levene(series1, series2)
    print(f'p value = {p_value}')
    if p_value >= alpha:
        print('We fail to reject the Null Hypothesis Ho')
    else:
        print('We reject the Null Hypothesis Ho')
```

In [61]:

```
variance_check(female_viral_load_sample,male_viral_load_sample)
```

```
p value = 0.2404736516217631
We fail to reject the Null Hypothesis Ho
```

In [62]:

```
from scipy.stats import mannwhitneyu
test, p_val= mannwhitneyu(female_viral_load_sample,male_viral_load_sample)

if p_val >= 0.05:
    print('We fail to reject the Null Hypothesis Ho')
else:
    print('We reject the Null Hypothesis Ho')
```

```
We fail to reject the Null Hypothesis Ho
```

Normality test - Shapiro Wilk test -> Failed
Equality of Variance Test - Levene's Test -> Pass
Non-parametric Test for confirmation - Mann Whitney test -> Pass
Hence we can proceed for 2 sample t test

In [63]:

```
alpha = 0.05
t_stats, p_value = stats.ttest_ind(female_viral_load_sample,male_viral_load_s
print(f"p-value is {p_value}, test statistics is {t_stats}")
if p_value < alpha:
      print(f"Since p value {p_value} is less than alpha {alpha}, we reject t
else:
    print(f"We fail to reject the H0 and hence can say that the viral load of
```

```
p-value is 0.9697346404030945, test statistics is 0.037950623129076856
We fail to reject the H0 and hence can say that the viral load of females i
s same as that of males.
```