In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import norm,ttest_1samp,ttest_ind,ttest_rel
from statsmodels.stats.weightstats import ztest
from bioinfokit.analys import stat
from scipy import stats
from scipy.stats import t
from scipy.stats import chisquare,f_oneway,kruskal
from scipy.stats import ttest_ind_from_stats # Takes sample means, std, n and returns st
from scipy.stats import chi2
import statistics
import random
from scipy.stats import chi2_contingency
from statsmodels.distributions.empirical_distribution import ECDF
```

In [2]:

```
data = pd.read_csv("scaler_apollo_hospitals.csv")
data
```

Out[2]:

	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
...	...	...	...	...	...	...	...	...
1333	1333	50	male	no	northwest	10.32	3	26501
1334	1334	18	female	no	northeast	10.64	0	5515
1335	1335	18	female	no	southeast	12.28	0	4075
1336	1336	21	female	no	southwest	8.60	0	5020
1337	1337	61	female	yes	northwest	9.69	0	72853

1338 rows × 8 columns

In [3]:

```
data.shape
```

Out[3]:

```
(1338, 8)
```

In [4]:

```
data.dtypes
```

Out[4]:

```
Unnamed: 0                  int64
age                         int64
sex                        object
smoker                     object
region                     object
viral load                float64
severity level              int64
hospitalization charges     int64
dtype: object
```

In [5]:

```
data.describe()
```

Out[5]:

	Unnamed: 0	age	viral load	severity level	hospitalization charges
count	1338.000000	1338.000000	1338.000000	1338.000000	1338.000000
mean	668.500000	39.207025	10.221233	1.094918	33176.058296
std	386.391641	14.049960	2.032796	1.205493	30275.029296
min	0.000000	18.000000	5.320000	0.000000	2805.000000
25%	334.250000	27.000000	8.762500	0.000000	11851.000000
50%	668.500000	39.000000	10.130000	1.000000	23455.000000
75%	1002.750000	51.000000	11.567500	2.000000	41599.500000
max	1337.000000	64.000000	17.710000	5.000000	159426.000000

In [6]:

```
data.describe(include = ['object'])
```

Out[6]:

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

In [7]:

```
data["sex"].unique()
```

Out[7]:

```
array(['female', 'male'], dtype=object)
```

In [8]:

```
data["sex"].value_counts()
```

Out[8]:

```
male      676
female    662
Name: sex, dtype: int64
```

In [9]:

```
data["smoker"].unique()
```

Out[9]:

```
array(['yes', 'no'], dtype=object)
```

In [10]:

```
data["smoker"].value_counts()
```

Out[10]:

```
no     1064
yes     274
Name: smoker, dtype: int64
```

In [11]:

```
data["region"].unique()
```

Out[11]:

```
array(['southwest', 'southeast', 'northwest', 'northeast'], dtype=object)
```

In [12]:

```
data["region"].value_counts()
```

Out[12]:

```
southeast    364
southwest    325
northwest    325
northeast    324
Name: region, dtype: int64
```

In [18]:

```
data["severity level"].unique()
```

Out[18]:

```
array([0, 1, 3, 2, 5, 4], dtype=int64)
```

In [19]:

```
data["severity level"].value_counts()
```

Out[19]:

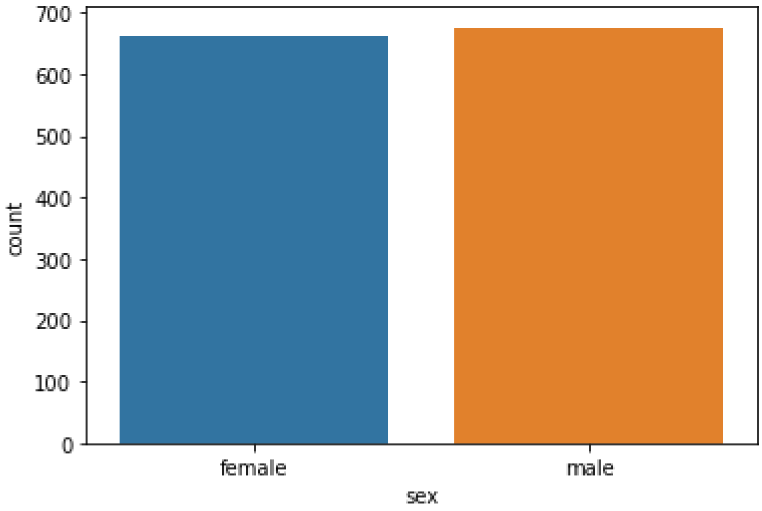
```
0    574
1    324
2    240
3    157
4     25
5     18
Name: severity level, dtype: int64
```

In [14]:

```
sns.countplot(x="sex",data=data)
```

Out[14]:

```
<AxesSubplot:xlabel='sex', ylabel='count'>
```

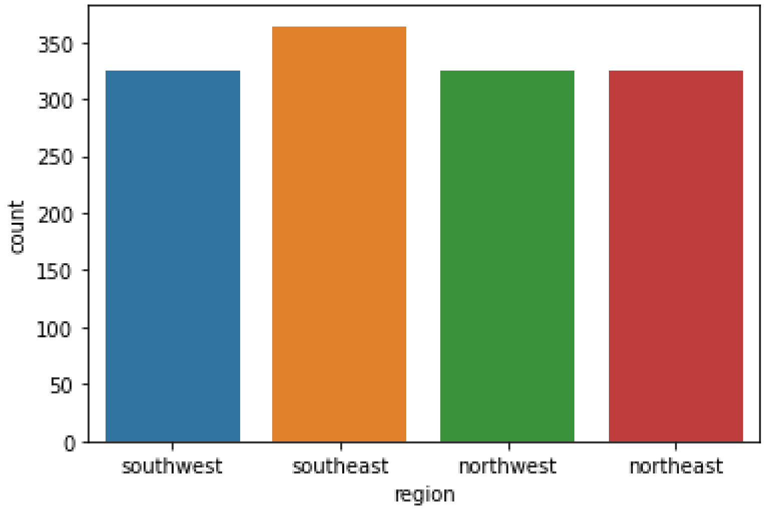


In [15]:

```
sns.countplot(x="region",data=data)
```

Out[15]:

```
<AxesSubplot:xlabel='region', ylabel='count'>
```

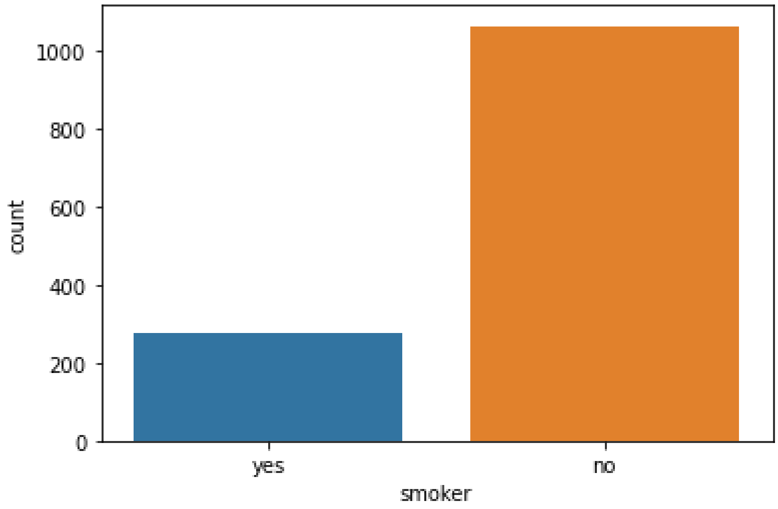


In [17]:

```
sns.countplot(x="smoker",data=data)
```

Out[17]:

```
<AxesSubplot:xlabel='smoker', ylabel='count'>
```

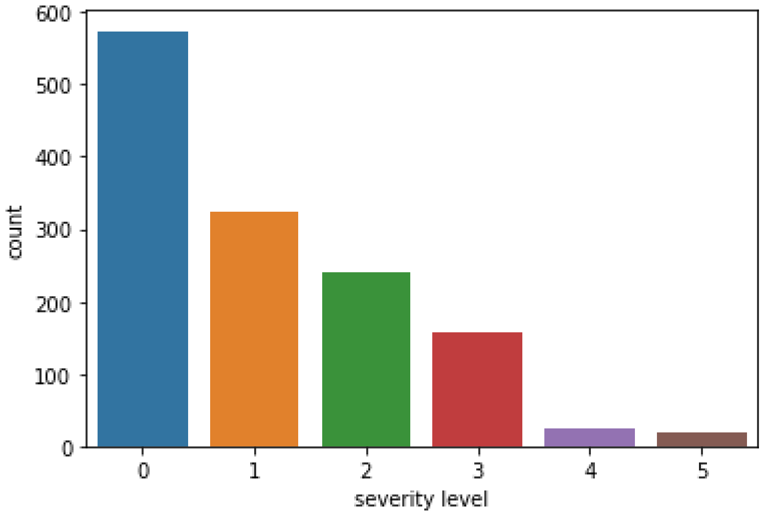


In [20]:

```
sns.countplot(x="severity level",data=data)
```

Out[20]:

```
<AxesSubplot:xlabel='severity level', ylabel='count'>
```

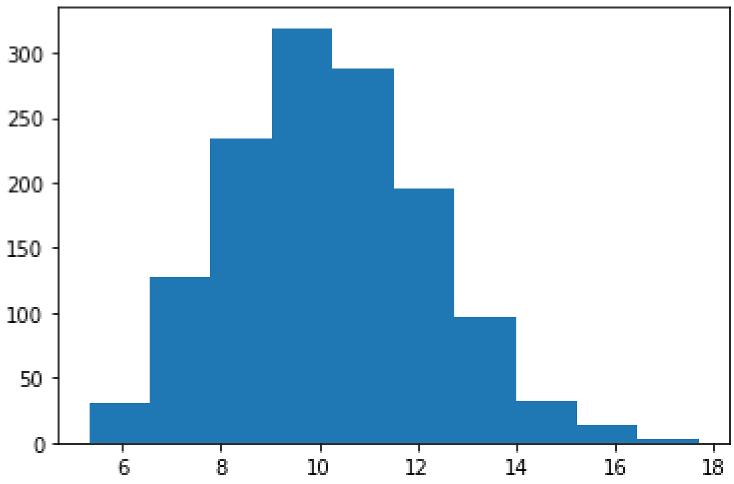


In [22]:

```
plt.hist(data["viral load"])
```

Out[22]:

```
(array([ 30., 127., 234., 319., 288., 195.,  96.,  32.,  14.,   3.]),
 array([ 5.32 ,  6.559,  7.798,  9.037, 10.276, 11.515, 12.754, 13.993,
        15.232, 16.471, 17.71 ]),
 <BarContainer object of 10 artists>)
```

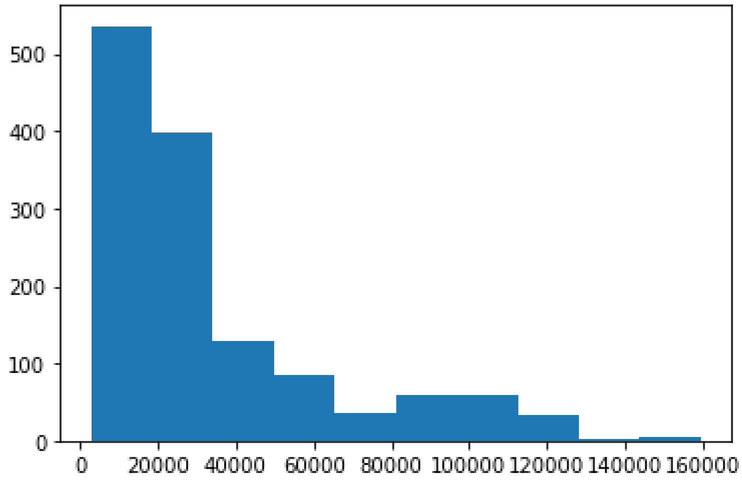


In [24]:

```
plt.hist(data["hospitalization charges"])
```

Out[24]:

```
(array([536., 398., 129.,  86.,  35.,  58.,  58.,  32.,   2.,   4.]),
 array([  2805. ,  18467.1,  34129.2,  49791.3,  65453.4,  81115.5,
         96777.6, 112439.7, 128101.8, 143763.9, 159426. ]),
 <BarContainer object of 10 artists>)
```

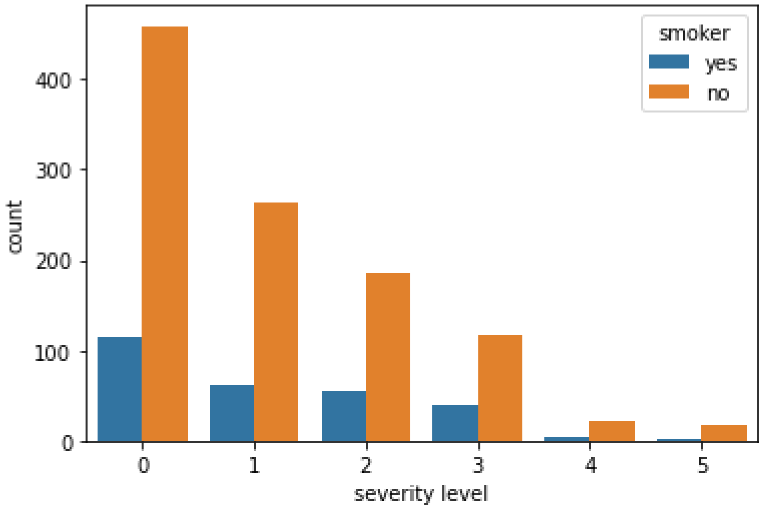


In [25]:

```
sns.countplot(x="severity level",hue="smoker",data=data)
```

Out[25]:

```
<AxesSubplot:xlabel='severity level', ylabel='count'>
```

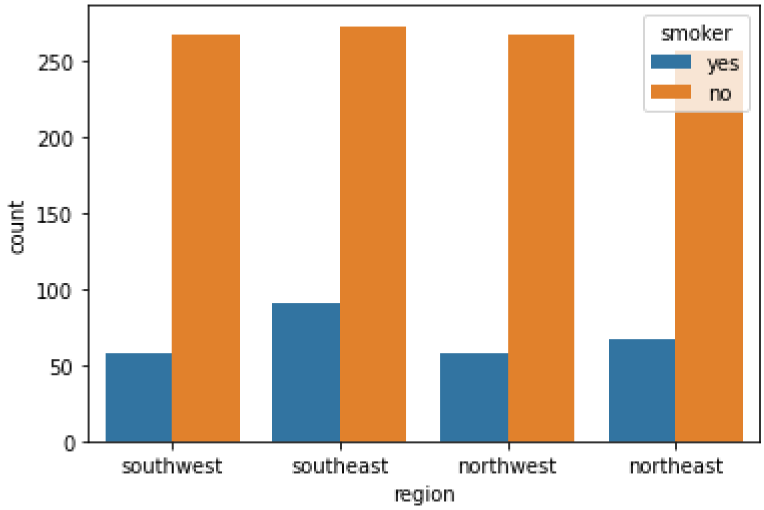


In [26]:

```
sns.countplot(x="region",hue="smoker",data=data)
```

Out[26]:

```
<AxesSubplot:xlabel='region', ylabel='count'>
```

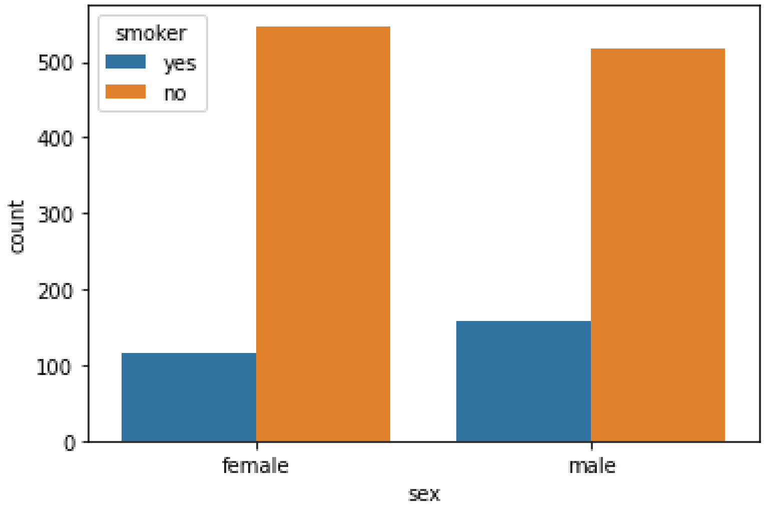


In [29]:

```
sns.countplot(x="sex",hue="smoker",data=data)
```

Out[29]:

```
<AxesSubplot:xlabel='sex', ylabel='count'>
```

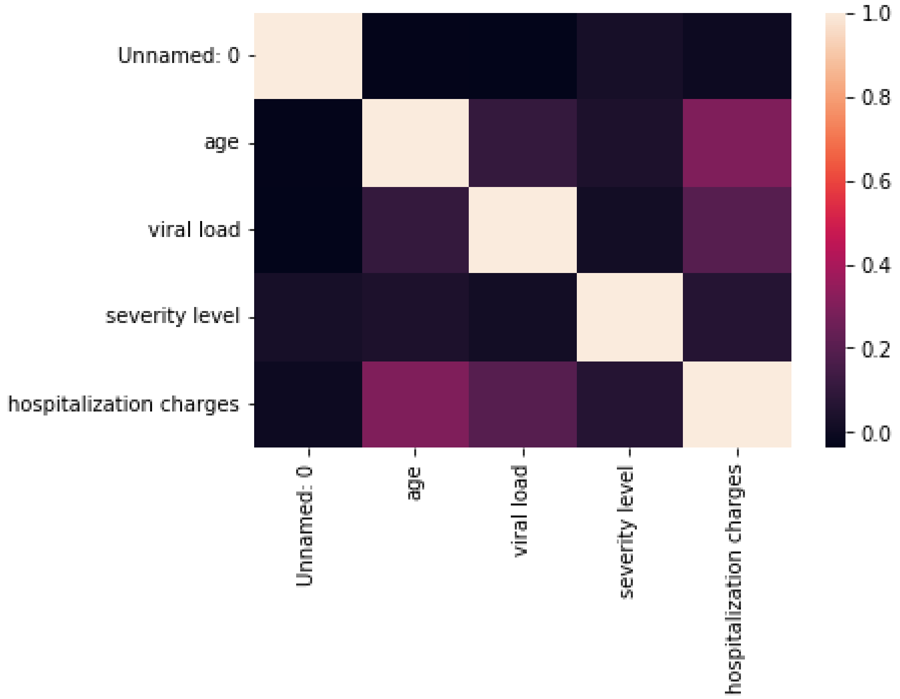


In [30]:

```
corr_matrix = data.corr()
sns.heatmap(corr_matrix)
```

Out[30]:

```
<AxesSubplot:>
```



In [31]:

```
data = data.drop("Unnamed: 0", axis='columns')
data
```

Out[31]:

	age	sex	smoker	region	viral load	severity level	hospitalization charges
0	19	female	yes	southwest	9.30	0	42212
1	18	male	no	southeast	11.26	1	4314
2	28	male	no	southeast	11.00	3	11124
3	33	male	no	northwest	7.57	0	54961
4	32	male	no	northwest	9.63	0	9667
...	...	...	...	...	...	...	...
1333	50	male	no	northwest	10.32	3	26501
1334	18	female	no	northeast	10.64	0	5515
1335	18	female	no	southeast	12.28	0	4075
1336	21	female	no	southwest	8.60	0	5020
1337	61	female	yes	northwest	9.69	0	72853

1338 rows × 7 columns

In [32]:

```
data.isnull().sum() # no missing values
```

Out[32]:

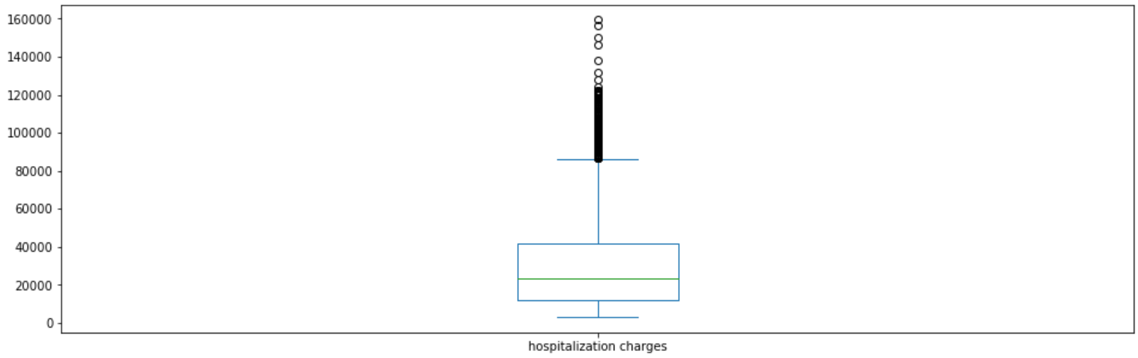
```
age                        0
sex                        0
smoker                     0
region                     0
viral load                 0
severity level             0
hospitalization charges    0
dtype: int64
```

In [33]:

```
plt.figure()
data['hospitalization charges'].plot.box(figsize=(16,5))
```

Out[33]:

```
<AxesSubplot:>
```

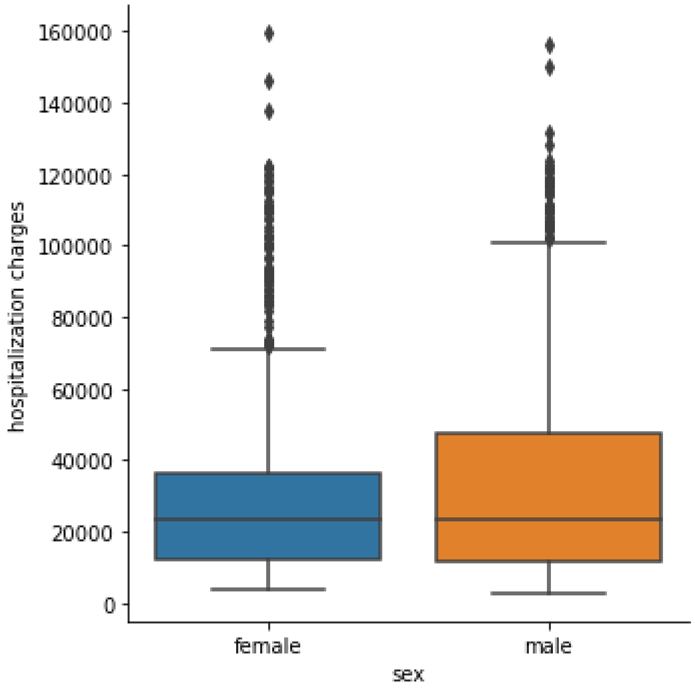


In [34]:

```
sns.catplot(data=data, x="sex", y="hospitalization charges", kind="box")
```

Out[34]:

```
<seaborn.axisgrid.FacetGrid at 0x2597bcde220>
```

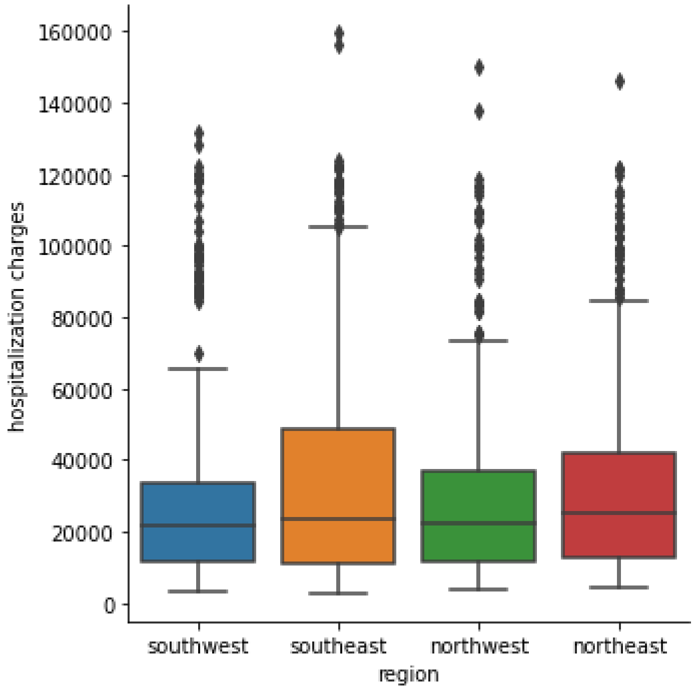


In [36]:

```
sns.catplot(data=data, x="region", y="hospitalization charges", kind="box")
```

Out[36]:

```
<seaborn.axisgrid.FacetGrid at 0x2597be582b0>
```

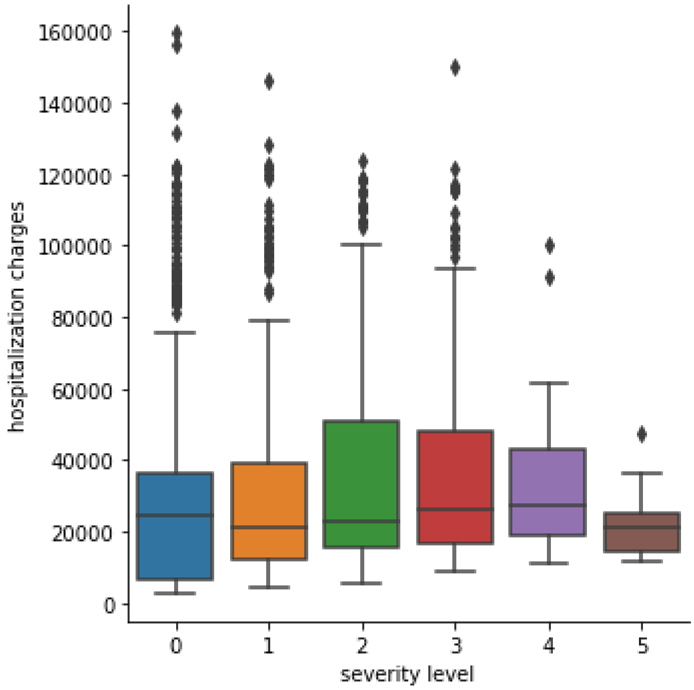


In [54]:

```
sns.catplot(data=data, x="severity level", y="hospitalization charges", kind="box")
```

Out[54]:

```
<seaborn.axisgrid.FacetGrid at 0x2597bf228b0>
```

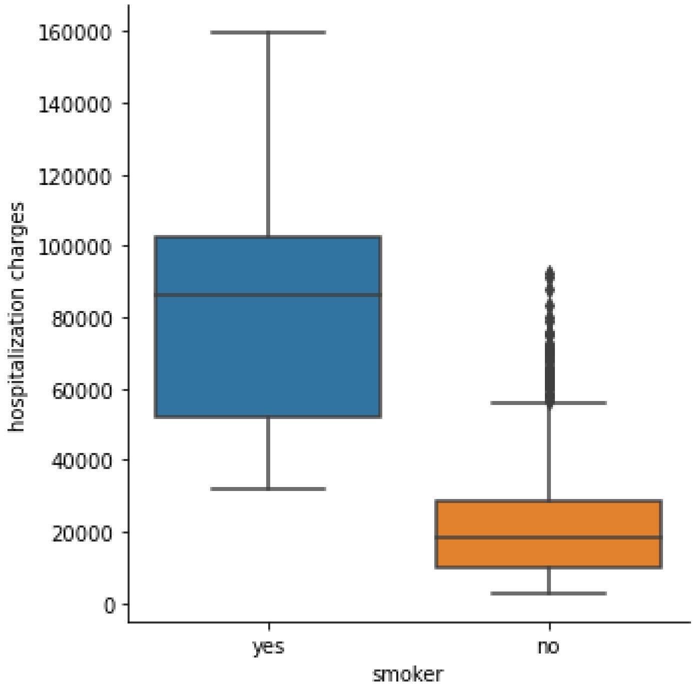


In [37]:

```
sns.catplot(data=data, x="smoker", y="hospitalization charges", kind="box")
```

Out[37]:

```
<seaborn.axisgrid.FacetGrid at 0x2597be98520>
```

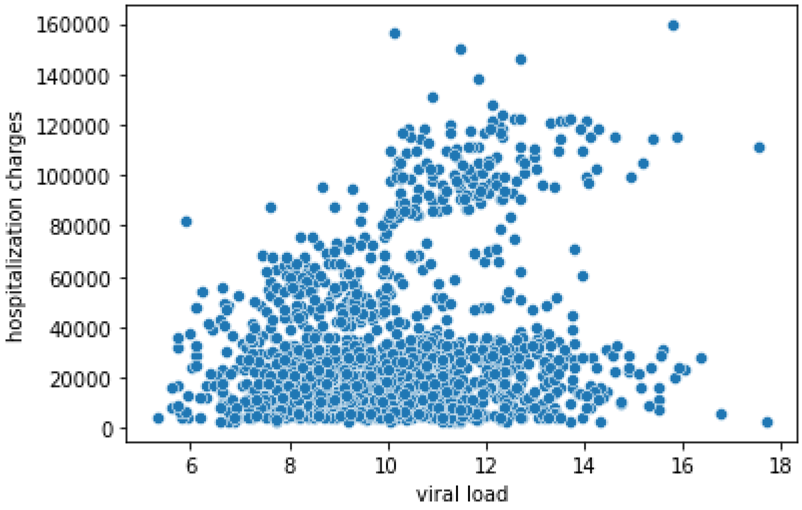


In [55]:

```
sns.scatterplot(x='viral load',y='hospitalization charges',data=data)
```

Out[55]:

```
<AxesSubplot:xlabel='viral load', ylabel='hospitalization charges'>
```



In [56]:

```
corr = data["viral load"].corr(data["hospitalization charges"])
corr
```

Out[56]:

```
0.1983875318556104
```

In [39]:

```
d1 = data[data["smoker"] == "yes"]["hospitalization charges"]
d2 = data[data["smoker"] == "no"]["hospitalization charges"]
stats.ttest_ind(d1, d2)
```

Out[39]:

```
Ttest_indResult(statistic=46.66489219013773, pvalue=8.275692527491989e-28
3)
```

Null Hypothesis : Mean of smoker and non-smoker hospitalization chargers are same Alternate Hypothesis : Mean of smoker is higher than of non Smoker since we are checking right tail test

In [44]:

```
## H0: u1 = u2
## Ha: u1 > u2
alpha = 0.05
test_stat, p_value = ttest_ind(d1, d2, alternative="greater") # right-tailed test
print("Test stat: ", test_stat)
print("p-value: ", p_value)

if p_value < alpha:
    print("Null rejected")
```

```
Test stat:  46.66489219013773
p-value:  4.1378462637459944e-283
Null rejected
```

Null hypothesis is rejected that means smoker hospitalization chargers are higher than non-smoker

In [45]:

```
## From the scratch implementation of ttest_ind
# EQUIVALENT FUNCTION: scipy.stats.ttest_ind
from scipy.stats import t

def ttest_ind_from_data(d1, d2, alternative="greater"):
    """
    d1: pandas Series
    d2: pandas Series
    alternative: {‘two-sided’, ‘less’, ‘greater’}, optional
    """
    n1 = len(d1)
    n2 = len(d2)

    m1 = d1.mean()
    m2 = d2.mean()

    s1 = d1.std()
    s2 = d2.std()

    df = n1 + n2 - 2

    s = np.sqrt((((n1-1)*(s1**2)) + ((n2-1)*(s2**2))) / (n1 + n2 - 2))

    t_stat = (m1 - m2) / (s*np.sqrt(1/n1+ 1/n2))

    if alternative == "two-sided":
        p_value = 2*(1 - t.cdf(t_stat, df=df))
    if alternative == "less":
        p_value = t.cdf(t_stat, df=df)
    if alternative == "greater":
        p_value = 1 - t.cdf(t_stat, df=df)
    print("T-stat = ", t_stat)
    print("P-value = ", p_value)
```

In [46]:

```
hypo = ttest_ind_from_data(d1,d2) #same above analysis in a different way
```

```
T-stat =  46.66489219013771
P-value =  0.0
```

In [47]:

```
d1 = data[data["sex"] == "female"]["viral load"]
d2 = data[data["sex"] == "male"]["viral load"]
stats.ttest_ind(d1, d2)
```

Out[47]:

```
Ttest_indResult(statistic=-1.695711164450323, pvalue=0.0901735841670204)
```

In [48]:

```
## From the scratch implementation of ttest_ind
# EQUIVALENT FUNCTION: scipy.stats.ttest_ind
from scipy.stats import t

def ttest_ind_from_data(d1, d2, alternative="two-sided"):
    """
    d1: pandas Series
    d2: pandas Series
    alternative: {‘two-sided’, ‘less’, ‘greater’}, optional
    """
    n1 = len(d1)
    n2 = len(d2)

    m1 = d1.mean()
    m2 = d2.mean()

    s1 = d1.std()
    s2 = d2.std()

    df = n1 + n2 - 2

    s = np.sqrt((((n1-1)*(s1**2)) + ((n2-1)*(s2**2))) / (n1 + n2 - 2))

    t_stat = (m1 - m2) / (s*np.sqrt(1/n1+ 1/n2))

    if alternative == "two-sided":
        p_value = 2*(1 - t.cdf(t_stat, df=df))
    if alternative == "less":
        p_value = t.cdf(t_stat, df=df)
    if alternative == "greater":
        p_value = 1 - t.cdf(t_stat, df=df)
    print("T-stat = ", t_stat)
    print("P-value = ", p_value)
```

In [49]:

```
## H0: u1 = u2
## Ha: u1 != u2
hypo = ttest_ind_from_data(d1,d2)
```

```
T-stat =  -1.695711164450404
P-value =  1.909826415832995
```

Failed to reject null hypothesis that means means of both genders are same and doesn't effect hospital charges

In [50]:

```
smoker_region = pd.crosstab(index=data["smoker"], columns=data["region"])
smoker_region
```

Out[50]:

region	northeast	northwest	southeast	southwest
smoker				
no	257	267	273	267
yes	67	58	91	58

In [51]:

```
# H0: smoker and region are independent
# Ha: dependant
chi2_contingency(smoker_region) # chistat, p-value, df, expected
```

Out[51]:

```
(7.34347776140707,
 0.06171954839170547,
 3,
 array([[257.65022422, 258.44544096, 289.45889387, 258.44544096],
        [ 66.34977578,  66.55455904,  74.54110613,  66.55455904]]))
```

Since alpha is 0.05 and the pvalue is 0.06 we say we fail to reject null hypothesis that means region doesn't affect the smoking habits

In [52]:

```
a = data[(data["sex"] == "female") * (data["severity level"] == 0)]["viral load"]
b = data[(data["sex"] == "female") * (data["severity level"] == 1)]["viral load"]
c = data[(data["sex"] == "female") * (data["severity level"] == 2)]["viral load"]
```

In [53]:

```
from scipy.stats import f_oneway
f_stat, p_value = f_oneway(a, b, c)
print("F-stat: ", f_stat)
print("p-value: ", p_value)
```

```
F-stat:  0.3355061434584082
p-value:  0.7151189650367746
```

Here null hypothesis is all three means are same and since pvalue is 0.71 and alpha is 0.05 we fail to reject null hypothesis that means severity level doesn't impact viral load

Business Insights

1) Hospital charges mostly depends on if a person is smoking or not

2) Smoking people have most hospital charges

3) Region Doesn't effect smoking habits

4) very less correlation between hospitalization charges and viral load

Recommendations

1) Hospital should focus more on if a person is smoker or non-smoker rather than geder or region

In [ ]:

```

```