Apollo Hospitals - Hypothesis Testing

Apollo Hospitals was established in 1983, renowned as the architect of modern healthcare in India. As the nation's first corporate hospital, Apollo Hospitals is acclaimed for pioneering the private healthcare revolution in the country.

As a data scientist working at Apollo 24/7, the ultimate goal is to tease out meaningful and actionable insights from Patient-level collected data.

You can help Apollo hospitals to be more efficient, to influence diagnostic and treatment processes, to map the spread of a pandemic.

One of the best examples of data scientists making a meaningful difference at a global level is in the response to the COVID-19 pandemic, where they have improved information collection, provided ongoing and accurate estimates of infection spread and health system demand, and assessed the effectiveness of government policies.

How can you help here?

The company wants to know:

• Which variables are significant in predicting the reason for hospitalization for different regions

• How well some variables like viral load, smoking, Severity Level describe the hospitalization charges

Column Profiling

Age: This is an integer indicating the age of the primary beneficiary (excluding those above 64 years, since they are generally covered by the government).

Sex: This is the policy holder's gender, either male or female

Viral Load: Viral load refers to the amount of virus in an infected person's blood

Severity Level: This is an integer indicating how severe the patient is

Smoker: This is yes or no depending on whether the insured regularly smokes tobacco.

Region: This is the beneficiary's place of residence in Delhi, divided into four geographic regions - northeast, southeast, southwest, or northwest

Hospitalization charges: Individual medical costs billed to health insurance

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

from scipy import stats
import matplotlib_inline
matplotlib_inline.backend_inline.set_matplotlib_formats('svg')
# Setting the theme for the plots to be drawn ahead
sns.set_theme(style="whitegrid")
```

In [2]:
```
AP = pd.read_csv(r"H:\Scaler\Apollo Project\Apollo DataSet.csv")
```

In [3]:
```
AP.head()
```

Out[3]:

	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 [4]:
```
AP.shape
```

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

The dataset has 1338 rows and 8 columns

In [5]:
```
AP.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
```

There seem to be no null values present in the dataset

The column datatypes are of int64,float64 and object

The columns Unnamed,age,severity level and hospitalization charges are of int64 date type

The columns sex,smoker,region are of object datatype

The column viral load is of float 64 data type

In [6]:
```
AP.describe()
```

Out[6]:

	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

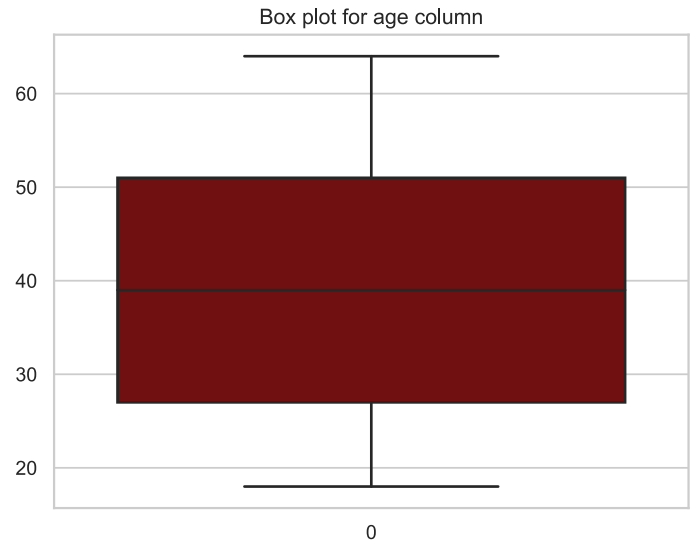
OUTLIER DETECTION

In [7]:
```
def find_outliers_IQR(df):
    q1 = df.quantile(0.25)
    q3 = df.quantile(0.75)
    IQR = q3-q1
    outliers = df[((df<(q1-1.5*IQR)) | (df>(q3+1.5*IQR)))]
    return outliers
```

In [8]:
```
#Checking outliers for age column

age_outliers = find_outliers_IQR(AP['age'])

print('number of outliers: '+ str(len(age_outliers)))

print('max outlier value: '+ str(age_outliers.max()))

print('min outlier values: '+ str(age_outliers.min()))
```

```
number of outliers: 0
max outlier value: nan
min outlier values: nan
```

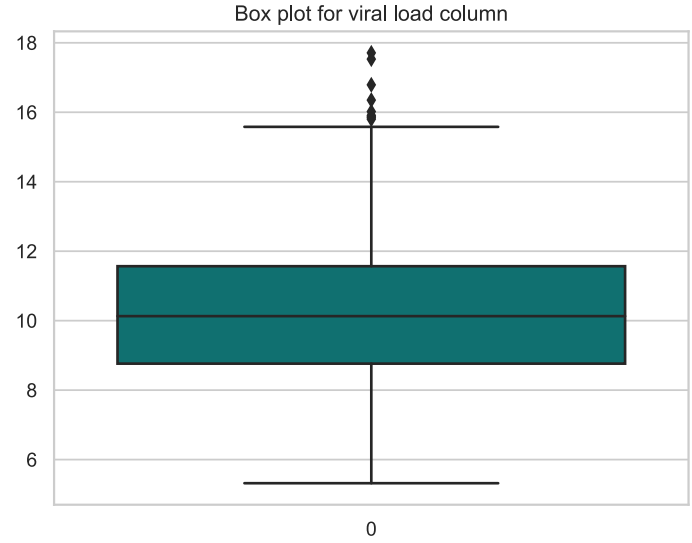
In [9]:
```
sns.boxplot(data=AP['age'], color='maroon')
plt.title("Box plot for age column")
plt.show()
```



In [10]:
```
#Checking outliers for viral load column

viralload_outliers = find_outliers_IQR(AP['viral load'])

print('number of outliers: '+ str(len(viralload_outliers)))

print('max outlier value: '+ str(viralload_outliers.max()))

print('min outlier values: '+ str(viralload_outliers.min()))
```

```
number of outliers: 9
max outlier value: 17.71
min outlier values: 15.8
```

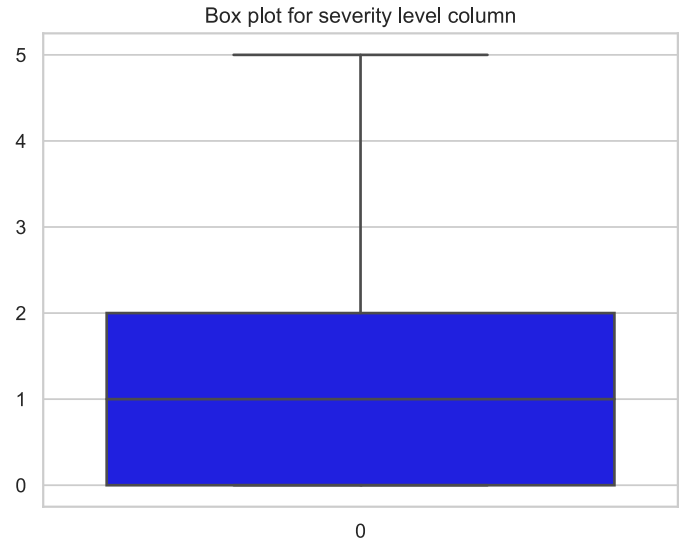
In [11]:
```
sns.boxplot(data=AP['viral load'],color='teal')
plt.title("Box plot for viral load column")
plt.show()
```



In [12]:
```
#Checking outliers for severity level column

severitylevel_outliers = find_outliers_IQR(AP['severity level'])

print('number of outliers: '+ str(len(severitylevel_outliers)))

print('max outlier value: '+ str(severitylevel_outliers.max()))

print('min outlier values: '+ str(severitylevel_outliers.min()))
```

```
number of outliers: 0
max outlier value: nan
min outlier values: nan
```

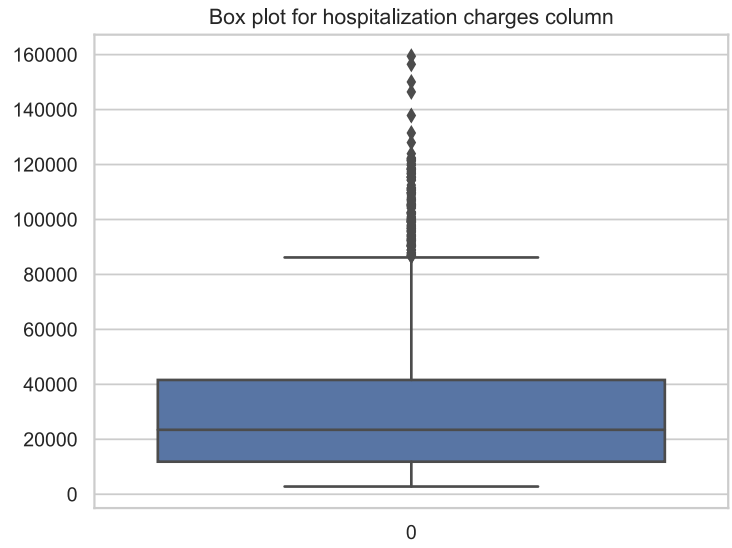
In [13]:
```
sns.boxplot(data=AP['severity level'],color='blue')
plt.title("Box plot for severity level column")
plt.show()
```



In [14]:
```
#Checking outliers for hospitalization charges column

hospitalizationcharges_outliers = find_outliers_IQR(AP['hospitalization charges'])

print('number of outliers: '+ str(len(hospitalizationcharges_outliers)))

print('max outlier value: '+ str(hospitalizationcharges_outliers.max()))

print('min outlier values: '+ str(hospitalizationcharges_outliers.min()))
```

```
number of outliers: 139
max outlier value: 159426
min outlier values: 86545
```

In [15]:
```
sns.boxplot(data=AP['hospitalization charges'])
plt.title("Box plot for hospitalization charges column")
plt.show()
```



The columns age and severity level have no outliers

The columns viral load and hospitalization charges have outliers

CHECKING THE CATEGORICAL VARIABLES

In [16]:
```
AP.head()
```

Out[16]:

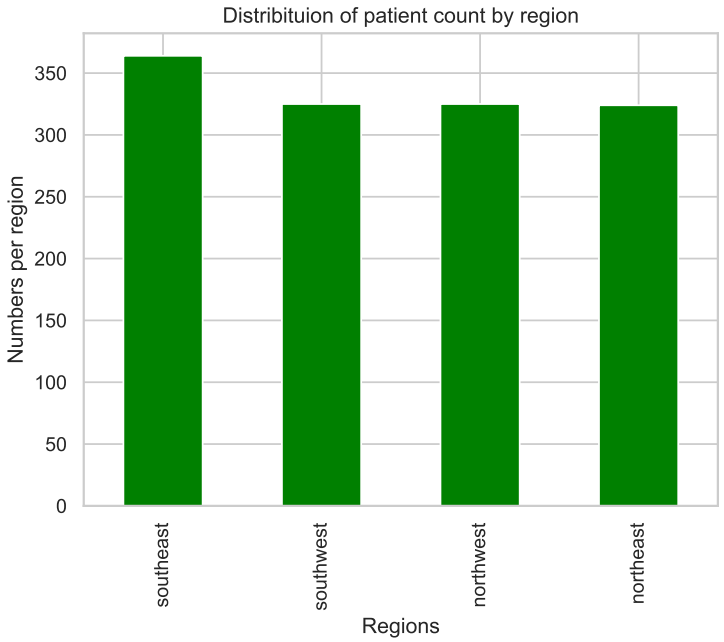
	Unnamed: 0	age	sex	smoker	region	viral load	severity level	hospitalization charges
0	0	19	female	yes	southwest	9.30	0	42212
1	1	18	male	no	southeast	11.26	1	4314
2	2	28	male	no	southeast	11.00	3	11124
3	3	33	male	no	northwest	7.57	0	54961
4	4	32	male	no	northwest	9.63	0	9667

In [17]:
```
AP['region'].value_counts()
```

Out[17]:
```
southeast    364
southwest    325
northwest    325
northeast    324
Name: region, dtype: int64
```

In [18]:
```
AP['region'].value_counts().plot(kind='bar',color = 'green')
plt.xlabel('Regions')
plt.ylabel('Numbers per region')
plt.title('Distribituion of patient count by region')
```

Out[18]:
```
Text(0.5, 1.0, 'Distribituion of patient count by region')
```



In [51]:
```
AP['region'].value_counts(normalize=True)
```

Out[51]:
```
southeast    0.272048
southwest    0.242900
northwest    0.242900
northeast    0.242152
Name: region, dtype: float64
```

Southeast has the highest count of patients.

Southwest and Northwest both are in second rank with equal number of patients.

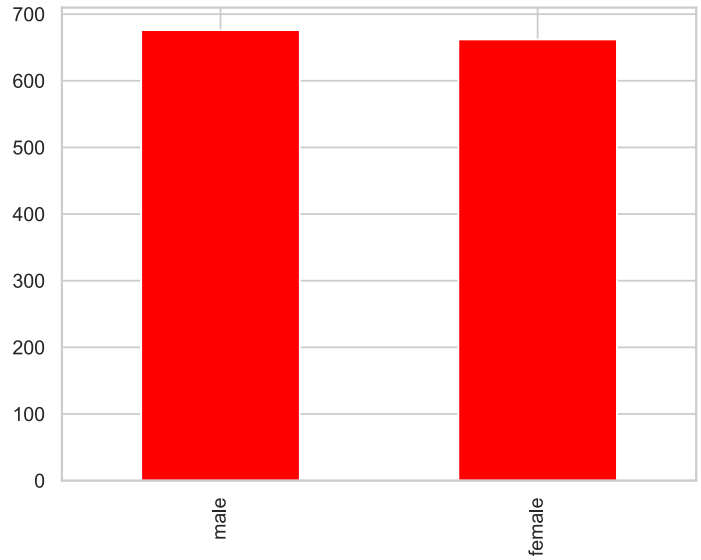
Northeast has the lowest number of patients

In [19]:
```
AP['sex'].value_counts()
```

Out[19]:
```
male      676
female    662
Name: sex, dtype: int64
```

In [20]:
```
AP['sex'].value_counts().plot(kind='bar',color='red')
```

Out[20]: <AxesSubplot:>

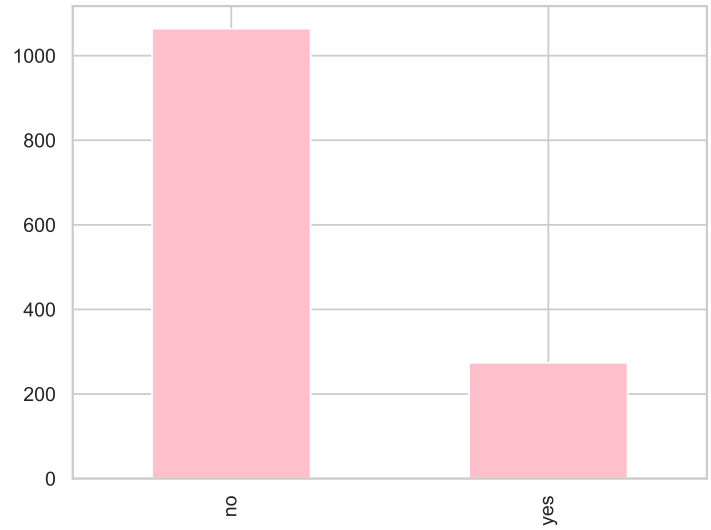


There are more male patients than female patients but the difference is not significant.

In [21]:
```
AP['smoker'].value_counts()
```

Out[21]:
```
no     1064
yes     274
Name: smoker, dtype: int64
```

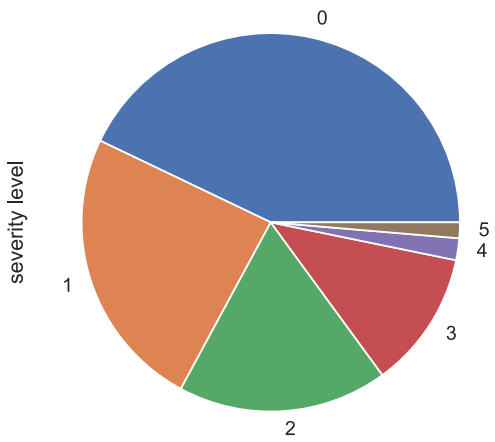
In [22]:
```
AP['smoker'].value_counts().plot(kind='bar',color='pink')
plt.show()
```



In [54]:
```
AP['severity level'].value_counts()
```

Out[54]:
```
0    574
1    324
2    240
3    157
4     25
5     18
Name: severity level, dtype: int64
```

In [56]:
```
AP['severity level'].value_counts().plot(kind='pie')
plt.show()
```



In [57]:
```
AP['severity level'].value_counts(normalize=True)
```

Out[57]:
```
0    0.428999
1    0.242152
2    0.179372
3    0.117339
4    0.018685
5    0.013453
Name: severity level, dtype: float64
```

Severity level 0 has the highest count followed by severity 1 and 2

In [63]:
```
AP['hospitalization charges'].value_counts()
```

Out[63]:
```
4593     2
35986    2
4099     2
23618    2
35003    2
        ..
70720    1
2873     1
92698    1
11846    1
72853    1
Name: hospitalization charges, Length: 1320, dtype: int64
```

In [46]:
```
APS = AP[AP['smoker']=='yes']
```

In [47]:
```
APNS =AP[AP['smoker']=='no']
```

In [48]:
```
len(APNS)/len(APS)
```

Out[48]:
```
3.883211678832117
```

In [49]:
```
len(APNS)/len(AP)
```

Out[49]:
```
0.7952167414050823
```

In [50]:
```
len(APS)/len(AP)
```

Out[50]:
```
0.20478325859491778
```

There is huge difference between the count of smoker and non smoker category of patients. Nearly 80 percent of the patients are non smokers and 20 percent are smokers

In [24]:
```
AP['age'].nunique()
```

Out[24]:
```
47
```

In [25]:
```
# Maximum and minimum values of 'age' column
print(f'Minimum Age available in data: {AP["age"].min()}, Maximum Age available in data: {AP["age"].max()}')
```

```
Minimum Age available in data: 18, Maximum Age available in data: 64
```

In [26]:
```
AP['age'].median()
```

Out[26]:
```
39.0
```

In [27]:
```
AP['age'].mean()
```

Out[27]:
```
39.20702541106129
```

In [28]:
```
AP['sex'].value_counts()
```

Out[28]:
```
male      676
female    662
Name: sex, dtype: int64
```

In [29]:
```
AP['smoker'].value_counts()
```

Out[29]:
```
no     1064
yes     274
Name: smoker, dtype: int64
```

In [31]:
```
AP['viral load'].nunique()
```

Out[31]:
```
462
```

In [35]:
```
print(f'Minimum Viral Load available in data: {AP["viral load"].min()},Maximum Viral Load available in data: {AP["viral load"].ma
```

```
Minimum Viral Load available in data: 5.32,Maximum Viral Load available in data: 17.71
```

In [52]:
```
AP["viral load"].mean()
```

Out[52]:
```
10.221233183856526
```

In [53]:
```
AP["viral load"].median()
```

Out[53]:
```
10.13
```

In [36]:
```
AP['severity level'].unique()
```

Out[36]:
```
array([0, 1, 3, 2, 5, 4], dtype=int64)
```

In [62]:
```
print(f'Minimum severity level available in data: {AP["severity level"].min()},Maximum severity level available in data: {AP["sev
```

```
Minimum severity level available in data: 0,Maximum severity level available in data: 5
```

In [38]:
```
print(f'Minimum Hospitalization Charge available in data: {AP["hospitalization charges"].min()},Maximum Hospitalization Charge av
```

```
Minimum Hospitalization Charge available in data: 2805,Maximum Hospitalization Charge available in data: 159426
```

Observations from data analysis

1) For the age column there are 47 unique values. The range is between 18 & 64 and there are no outliers.

2) The number of male and female patients are almost the same with count of male patients being 676 and female being 662.

3) Checking the smoker column there seems to be a huge difference in ratio with nearly 80 percent of the patients being

non smokers and 20 percent being smokers. The count of non smokers is 1064 The count of smokers is 274

4) Checking the region columns it can be seen that the data is almost evenly distributed.

REGION COUNT PERCENTAGE CONTRIBUTION southeast 364 0.272048 southwest 325 0.242900 northwest 325 0.242900 northeast 324 0.242152

5) The viral load column has the range between 5.32 and 17.71 There are about 9 outliers in the viral load column.

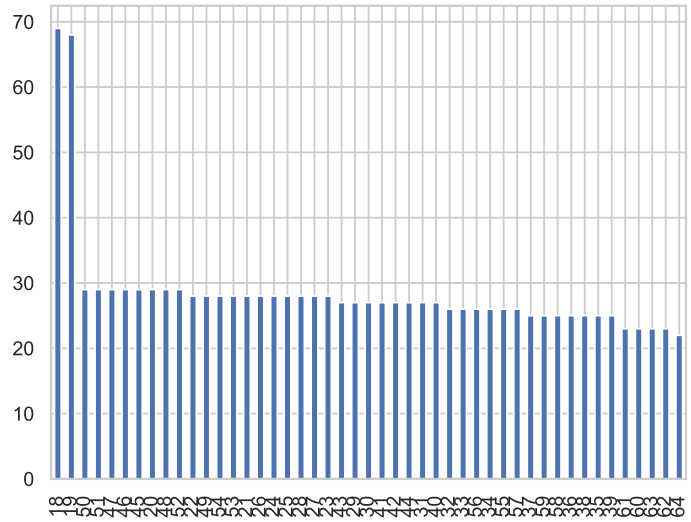
6) Severity level 0 has the highest count followed by severity 1 and 2

Severity Level Count Percentage contribution 0 574 0.428999
1 324 0.242152 2 240 0.179372 3 157 0.117339 4 25 0.018685
5 18 0.013453

7) Hospitalization charges has a minimum value of 2805 and maximum charge of 159426

Univariate analysis

In [78]:
```
AP['age'].value_counts().plot(kind='bar')
f = plt.figure()
plt.figure(figsize=(2, 2))

plt.show()
```



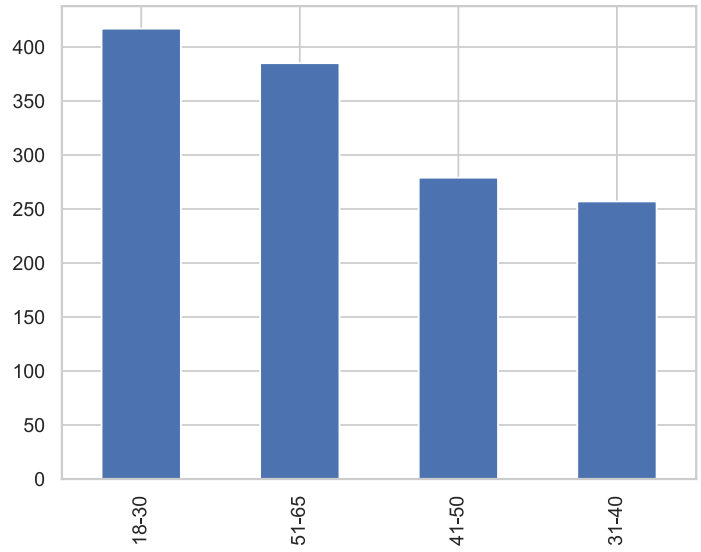

```
<Figure size 640x480 with 0 Axes>
```

```
<Figure size 200x200 with 0 Axes>
```

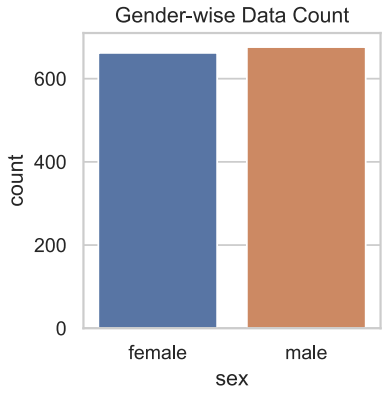
In [79]:
```
AP1 = pd.read_csv(r"H:\Scaler\Apollo Project\Apollo DataSet.csv")
bins = [17,30,40,50,65]
labels = ['18-30','31-40','41-50','51-65']
AP1['age']=pd.cut(AP1['age'],bins=bins,labels=labels,right=False)
AP1['age'].value_counts()
```

Out[79]:
```
18-30    417
51-65    385
41-50    279
31-40    257
Name: age, dtype: int64
```

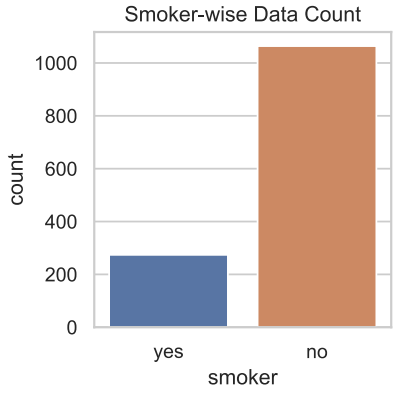
In [81]:
```
AP1['age'].value_counts().plot(kind='bar')
plt.show()
```



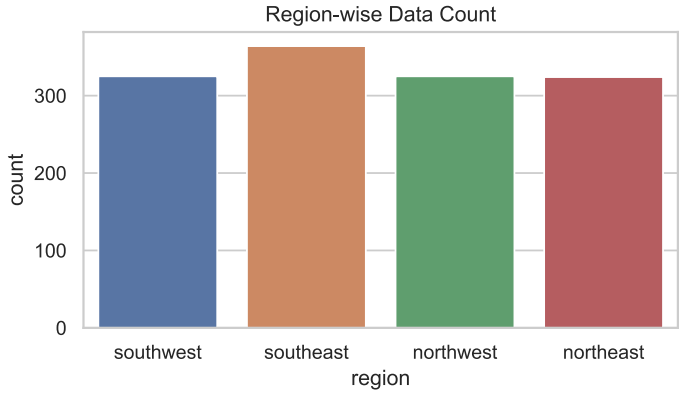
In [84]:
```
# Plot showing count of data of different gender
fig, ax = plt.subplots(figsize = (3,3))
sns.countplot(data = AP, x = 'sex', ax = ax).set(title = 'Gender-wise Data Count')
plt.show()
```



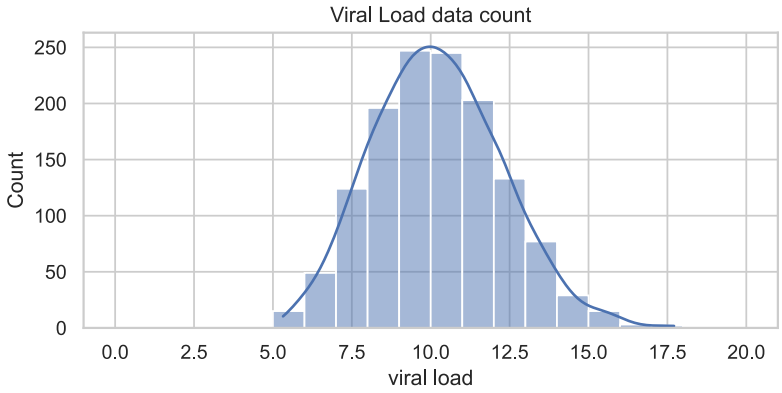
In [82]:
```
# Plot showing count of data of different 'smoker'
fig, ax = plt.subplots(figsize = (3,3))
sns.countplot(data = AP, x = 'smoker', ax = ax).set(title = 'Smoker-wise Data Count')
plt.show()
```



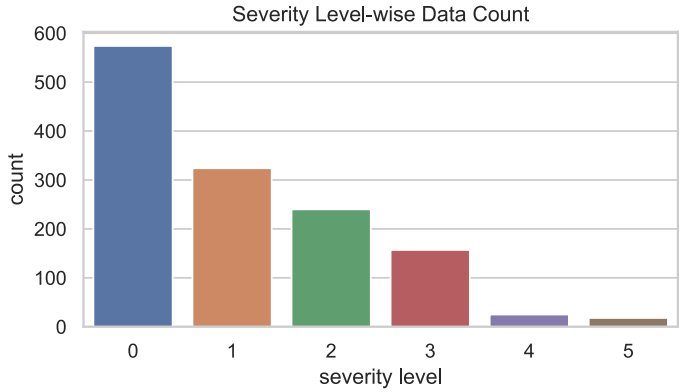
In [85]:
```
# Plot showing count of data of different region
fig, ax = plt.subplots(figsize = (6,3))
sns.countplot(data = AP, x = 'region', ax = ax).set(title = 'Region-wise Data Count')
plt.show()
```



In [88]:
```
Plot showing count of data of different viral load
g, ax = plt.subplots(figsize = (7,3))
s.histplot(data = AP, x = 'viral load', binwidth = 1, binrange = (0,20), ax = ax, kde = True).set(title = 'Viral Load data count')
t.show()
```



In [90]:
```
# Plot showing count of data of different severity level
fig, ax = plt.subplots(figsize = (6,3))
sns.countplot(data = AP, x = 'severity level', ax = ax).set(title = 'Severity Level-wise Data Count')
plt.show()
```



In [92]:
```
# Checking the percentage of Null Values available in the data
(100 * AP.isnull().sum())/len(AP)
```

Out[92]:
```
Unnamed: 0                0.0
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
```

In [93]:
```
# Checking the statistical description for numerical features
AP.describe()
```

Out[93]:

	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 [94]:
```
# Checking the statistical description of categorical features
AP.describe(include = object)
```

Out[94]:

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

Observations/Insights from checking null values and describing data

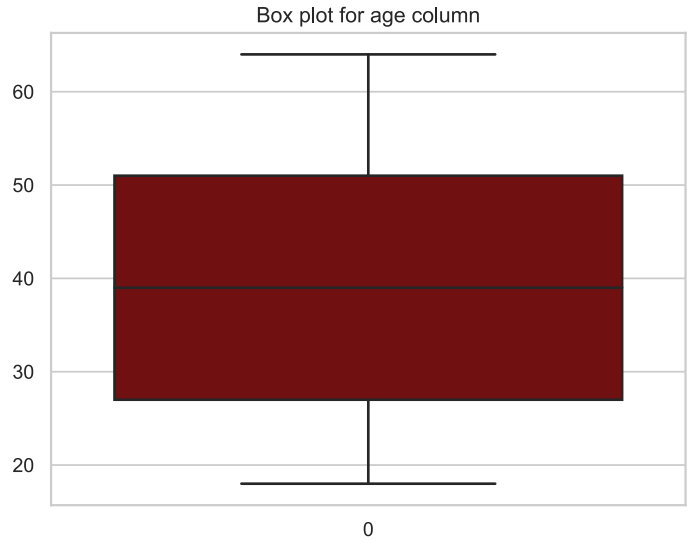
1. There are no missing values in our dataset.
2. The mean and median of features 'age' and 'viral load' are very close indicating there are no significant outliers in the data.
3. The mean and median of dependent variable 'hospitalization charges' has huge difference which indicates the presence of outliers in the data.

Outlier Treatment

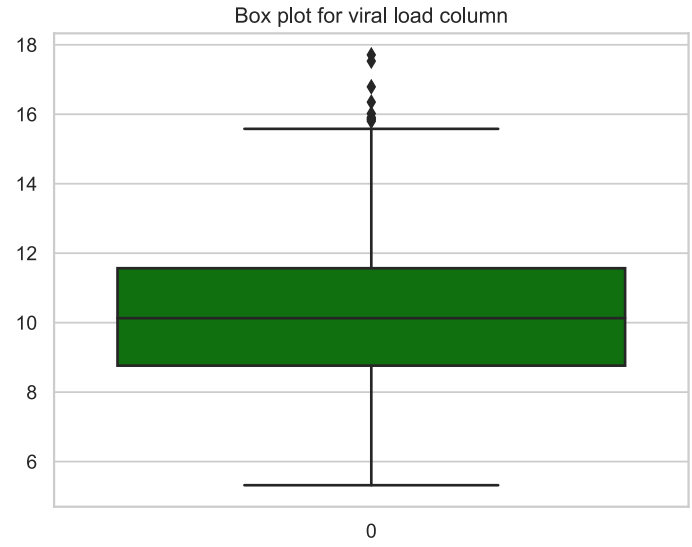
In [95]:
```
AP_copy = AP.copy(deep = True)
AP_copy.info()
```

```
<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
```

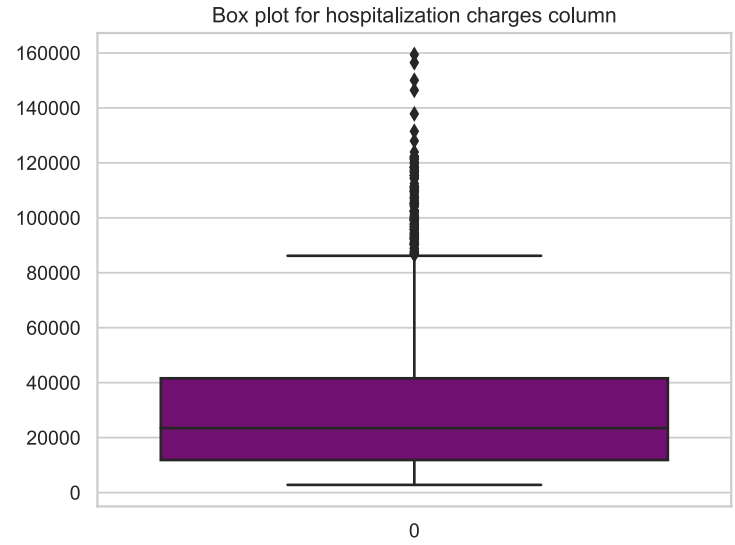
In [96]:
```
#Box plot for age column

sns.boxplot(data=AP['age'], color='maroon')
plt.title("Box plot for age column")
plt.show()
```



In [98]:
```
# Boxplot for 'viral load' column
sns.boxplot(data=AP['viral load'], color='green')
plt.title("Box plot for viral load column")
plt.show()
```



In [99]:
```
# Boxplot for 'hospitalization charges' column
sns.boxplot(data=AP['hospitalization charges'], color='purple')
plt.title("Box plot for hospitalization charges column")
plt.show()
```



In [101]:
```
# Treating the outliers using IQR Method
f_cols = ['viral load', 'hospitalization charges']
for i in f_cols:
    q1 = AP[i].quantile(0.25)
    q3 = AP[i].quantile(0.75)
    iqr = q3 - q1
    AP_copy = AP_copy[(AP_copy[i] > (q1 - 1.5*iqr)) & (AP_copy[i] < (q3 + 1.5*iqr))]
```

In [102]:
```
# Again checking the info of outlier treated dataset
AP_copy.info()
```

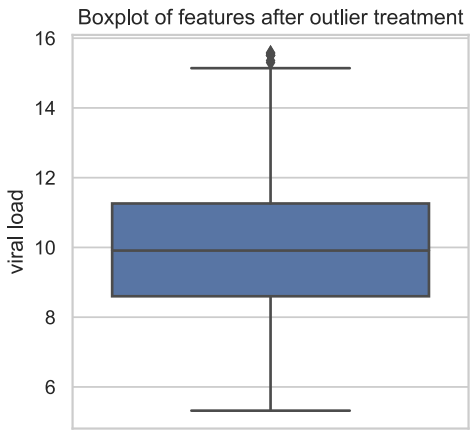
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1193 entries, 0 to 1337
Data columns (total 8 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Unnamed: 0              1193 non-null   int64  
 1   age                     1193 non-null   int64  
 2   sex                     1193 non-null   object 
 3   smoker                  1193 non-null   object 
 4   region                  1193 non-null   object 
 5   viral load              1193 non-null   float64
 6   severity level          1193 non-null   int64  
 7   hospitalization charges  1193 non-null   int64  
dtypes: float64(1), int64(4), object(3)
memory usage: 83.9+ KB
```

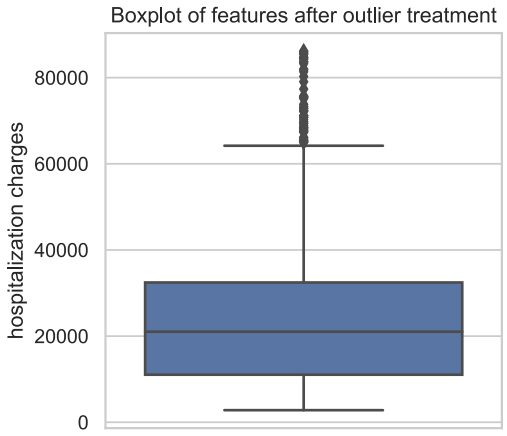
In [103]:
```
# Describing the final dataset
AP_copy.describe()
```

Out[103]:

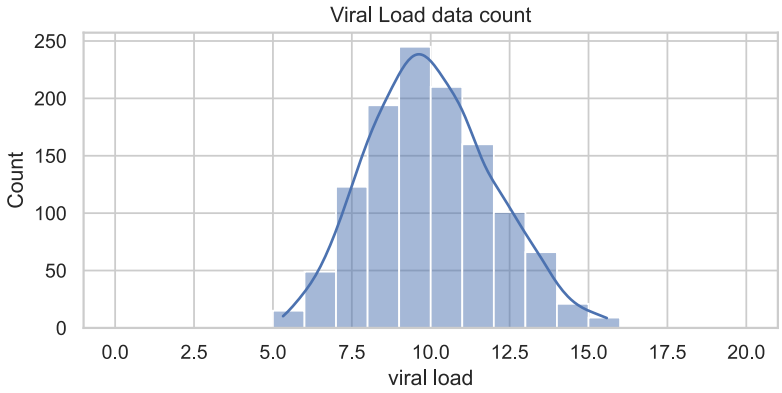
	Unnamed: 0	age	viral load	severity level	hospitalization charges
count	1193.000000	1193.000000	1193.000000	1193.000000	1193.000000
mean	669.999162	38.981559	9.999589	1.085499	24855.675608
std	384.593275	14.063482	1.955718	1.216249	18128.312143
min	0.000000	18.000000	5.320000	0.000000	2805.000000
25%	344.000000	26.000000	8.600000	0.000000	11038.000000
50%	669.000000	39.000000	9.910000	1.000000	21025.000000
75%	999.000000	51.000000	11.260000	2.000000	32448.000000
max	1337.000000	64.000000	15.580000	5.000000	86182.000000

In [104]:
```
# Checking the plots of the numerical features again after outlier treatment
for j in f_cols:
    fig, ax = plt.subplots(figsize = (4,4))
    sns.boxplot(data = AP_copy, y = j, ax = ax).set(title = 'Boxplot of features after outlier treatment')
    plt.show()
```

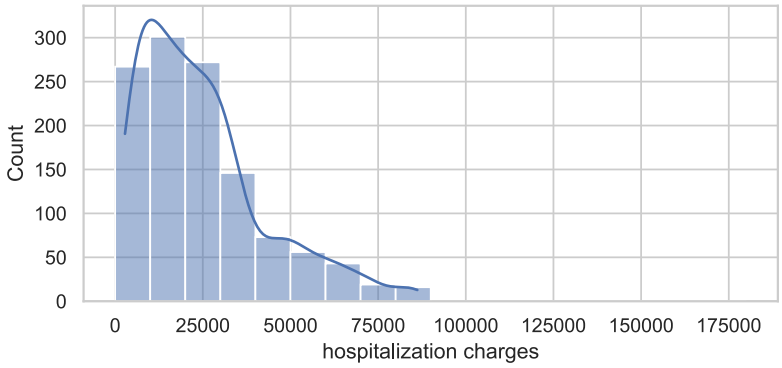




In [106]:
```
Outlier Treatment - Plot showing count of data of different viral load
= plt.subplots(figsize = (7,3))
tplot(data = AP_copy, x = 'viral load', binwidth = 1, binrange = (0,20), ax = ax, kde = True).set(title = 'Viral Load data count'
w()
```

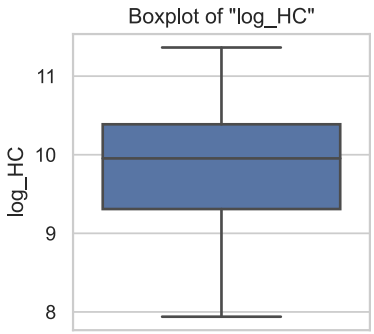


In [108]:
```
# After Outlier Treatment - Plot showing count of data of different Hospitalization Charges
fig, ax = plt.subplots(figsize = (7,3))
sns.histplot(data = AP_copy, x = 'hospitalization charges', binwidth = 10000, binrange = (0,180000), ax = ax, kde = True)
plt.show()
```

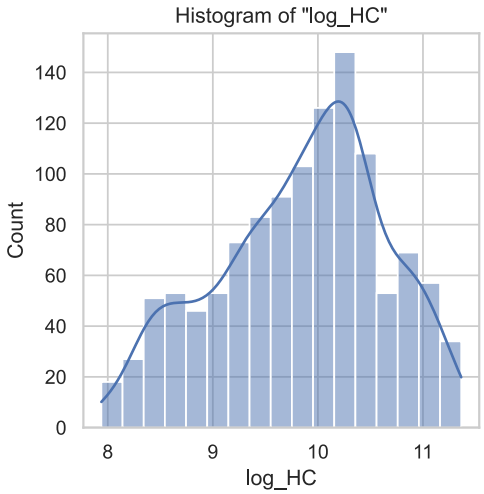


In [110]:
```
# Performing Log-Transformation of 'hospitalization charges' column
AP_copy['log_HC'] = np.log(AP_copy['hospitalization charges'])
```

In [111]:
```
# Boxplot showing 'log_HC' column
fig, ax = plt.subplots(figsize = (3,3))
sns.boxplot(data = AP_copy, y = 'log_HC', ax = ax).set(title = 'Boxplot of "log_HC"')
plt.show()
```



In [112]:
```
# Plot showing histogram of 'log_HC'
fig, ax = plt.subplots(figsize = (4,4))
sns.histplot(data = AP_copy, x = 'log_HC', ax = ax, kde = True).set(title = 'Histogram of "log_HC"')
plt.show()
```



Observations/Insights from Outlier Detection and Treatment

1. The plots showed that the features 'viral load' and 'hospitalization charges' had outliers.

2. After outlier treatment by IQR Method, the outliers vanished from 'viral load' while it got reduced in 'hospitalization charges' feature.

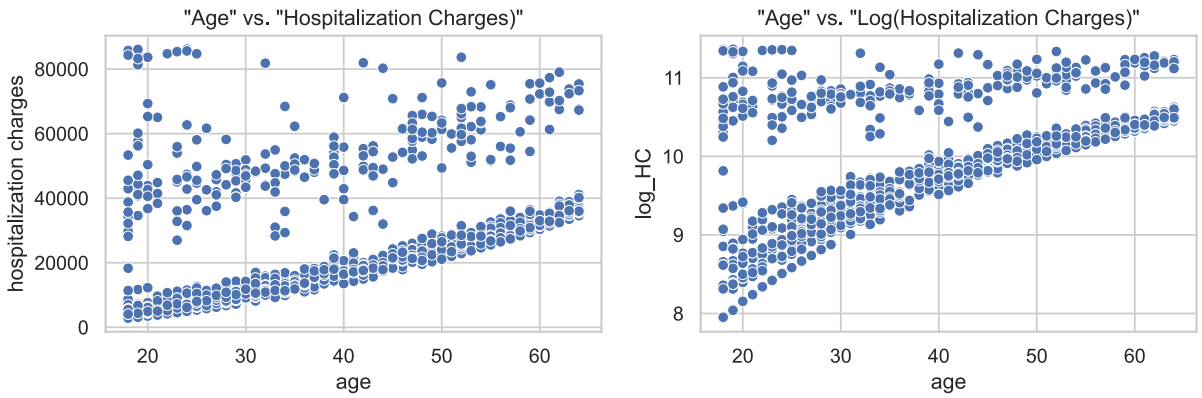
3. The histogram of feature 'hospitalization charges' showed that it is right-skewed. So we wanted to check if it can be converted into a Normalcurve upon log-transformation of 'hospitalization charges'.

4. In quick-view, the log-transformed feature of 'hospitalization charges' seems like Normal Distribution but it cannot be said with accuracy anda statistical test would give better understanding of this situation.

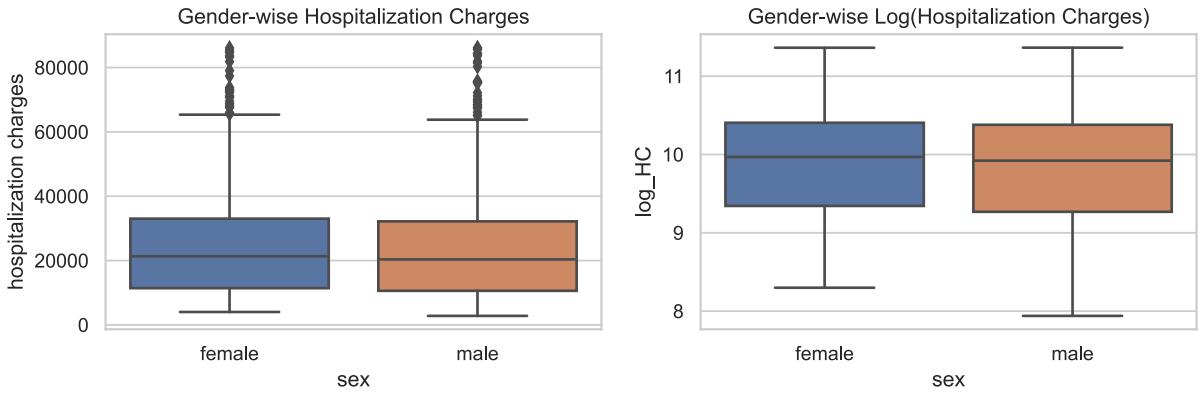
Bi-Variate Visual Analysis

Here the dependent variables are 'hospitalization charges' and independent variables are 'age', 'sex', 'smoker', 'region', 'viral load', we will tryand see the plots between dependent and independent variables.

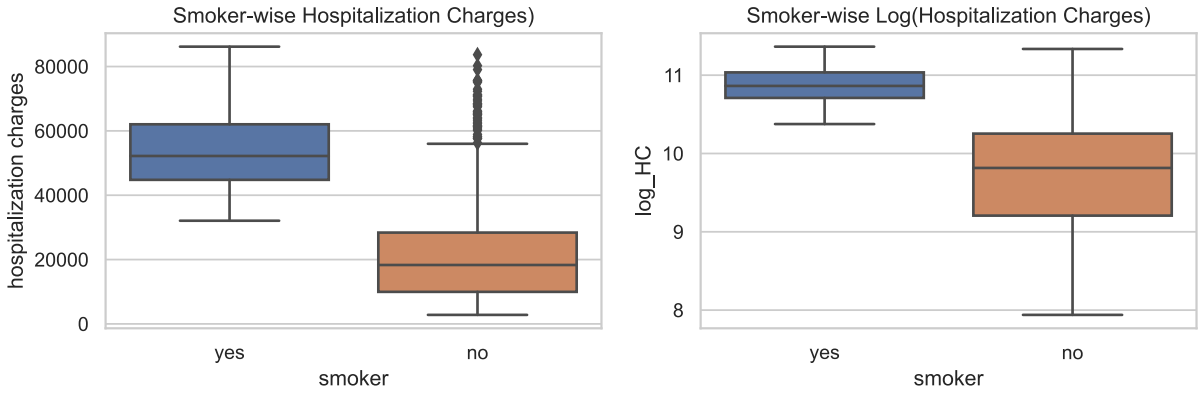
In [116]:
```
# Plot of 'age' vs. 'hospitalization charges' and 'age' vs. 'log_HC'
plt.figure(figsize = (11,3))
plt.subplot(121)
sns.scatterplot(data = AP_copy, x = 'age', y = 'hospitalization charges').set(title = '"Age" vs. "Hospitalization Charges)"')
plt.subplot(122)
sns.scatterplot(data = AP_copy, x = 'age', y = 'log_HC').set(title = '"Age" vs. "Log(Hospitalization Charges)"')
plt.show()
```



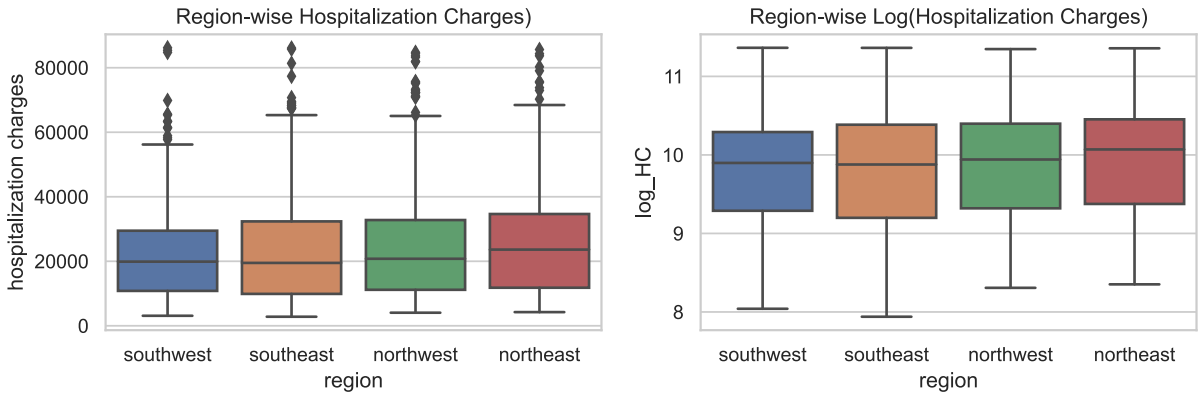
In [117]:
```
# Plot of 'sex' vs. 'hospitalization charges' and 'sex' vs. 'log_HC'
plt.figure(figsize = (11,3))
plt.subplot(121)
sns.boxplot(data = AP_copy, y = 'hospitalization charges', x = 'sex').set(title = 'Gender-wise Hospitalization Charges')
plt.subplot(122)
sns.boxplot(data = AP_copy, y = 'log_HC', x = 'sex').set(title = 'Gender-wise Log(Hospitalization Charges)')
plt.show()
```



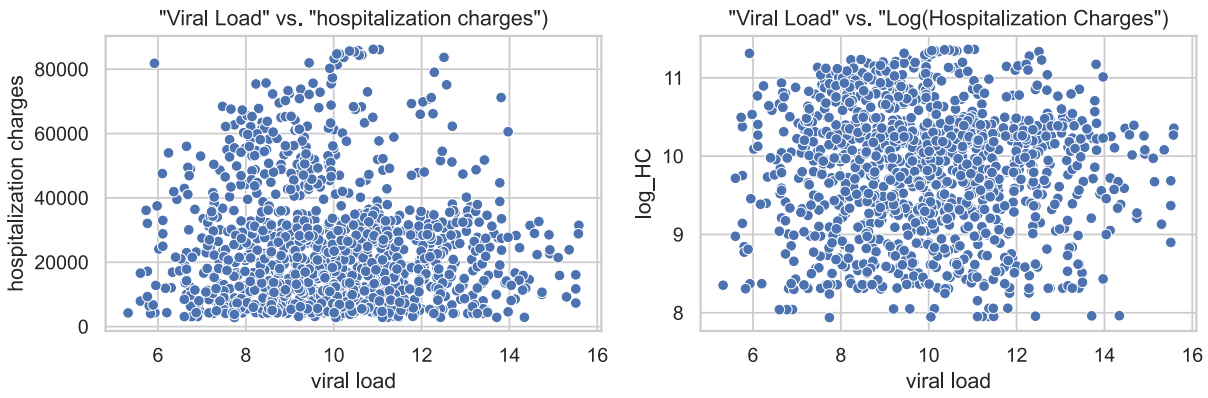
In [120]:
```
# Plot of 'smoker' vs. 'hospitalization charges' and 'smoker' vs. 'log_HC'
plt.figure(figsize = (11,3))
plt.subplot(121)
sns.boxplot(data = AP_copy, y = 'hospitalization charges', x = 'smoker').set(title = 'Smoker-wise Hospitalization Charges)')
plt.subplot(122)
sns.boxplot(data = AP_copy, y = 'log_HC', x = 'smoker').set(title = 'Smoker-wise Log(Hospitalization Charges)')
plt.show()
```



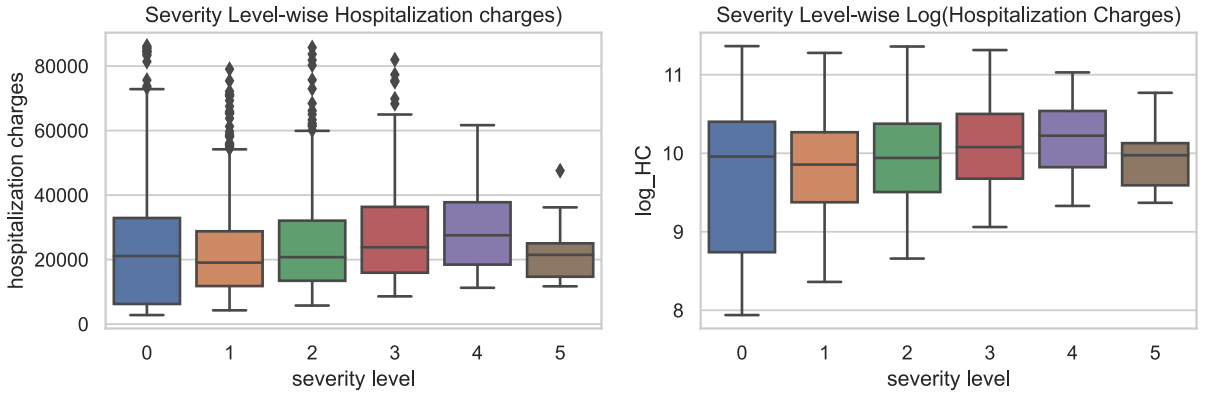
In [124]:
```
# Plot of 'region' vs. 'hospitalization charges' and 'region' vs. 'log_HC'
plt.figure(figsize = (11,3))
plt.subplot(121)
sns.boxplot(data =  AP_copy, y = 'hospitalization charges', x = 'region').set(title = 'Region-wise Hospitalization Charges)')
plt.subplot(122)
sns.boxplot(data =  AP_copy, y = 'log_HC', x = 'region').set(title = 'Region-wise Log(Hospitalization Charges)')
plt.show()
```



In [130]:
```
# Plot of 'viral load' vs. 'hospitalization charges' and 'viral load' vs. 'log_HC'
plt.figure(figsize = (11,3))
plt.subplot(121)
sns.scatterplot(data = AP_copy, x = 'viral load', y = 'hospitalization charges').set(title = '"Viral Load" vs. "hospitalization c
plt.subplot(122)
sns.scatterplot(data = AP_copy, x = 'viral load', y = 'log_HC').set(title = '"Viral Load" vs. "Log(Hospitalization Charges")')
plt.show()
```



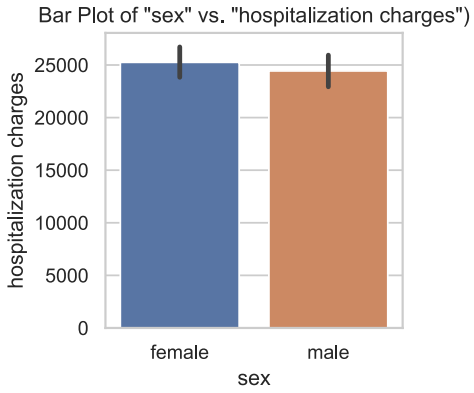
In [134]:
```
# Plot of 'severity level' vs. 'hospitalization charges' and 'severity level' vs. 'log_HC'
plt.figure(figsize = (11,3))
plt.subplot(121)
sns.boxplot(data = AP_copy, y = 'hospitalization charges', x = 'severity level').set(title = 'Severity Level-wise Hospitalizatior
plt.subplot(122)
sns.boxplot(data = AP_copy, y = 'log_HC', x = 'severity level').set(title = 'Severity Level-wise Log(Hospitalization Charges)')
plt.show()
```



In [135]:
```
# Calculating the Gender-wise mean hospitalization charges
AP_copy.groupby('sex')['hospitalization charges'].agg('mean')
```

Out[135]:
```
sex
female    25253.787234
male      24437.726804
Name: hospitalization charges, dtype: float64
```

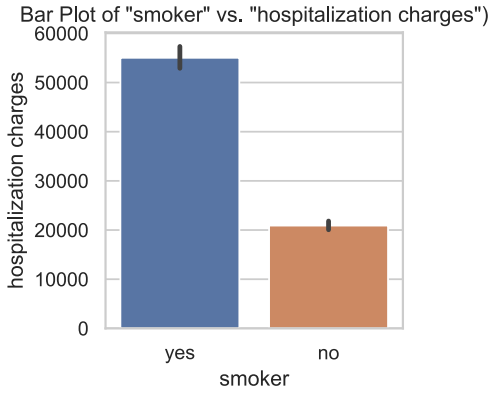
In [139]:
```
howing 'Sex' vs.'Hospitalization Charges'
t.subplots(figsize = (3,3))
data = AP_copy, x = 'sex', y = 'hospitalization charges', ax = ax).set(title = 'Bar Plot of "sex" vs. "hospitalization charges")')
```



In [140]:
```
# Calculating the Smoker-wise mean hospitalization charges
AP_copy.groupby('smoker')['hospitalization charges'].agg('mean')
```

Out[140]:
```
smoker
no     20907.971564
yes    55035.586957
Name: hospitalization charges, dtype: float64
```

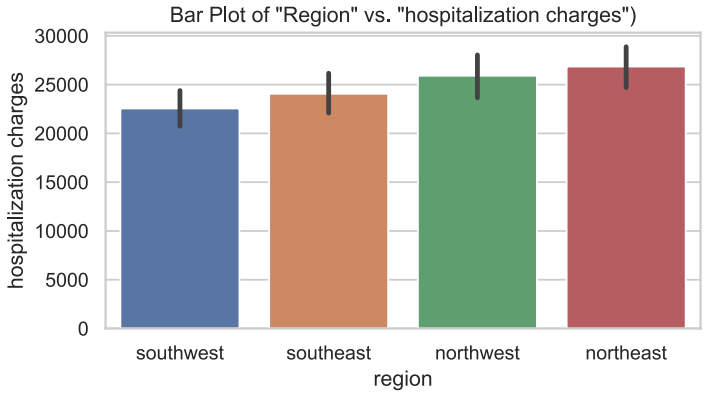
In [142]:
```
lot showing 'Smoker' vs.'Hospitalization Charges'
 = plt.subplots(figsize = (3,3))
plot(data = AP_copy, x = 'smoker', y = 'hospitalization charges', ax = ax).set(title = 'Bar Plot of "smoker" vs. "hospitalization
w()
```



In [143]:
```
# Calculating the Region-wise mean hospitalization charges
AP_copy.groupby('region')['hospitalization charges'].agg('mean')
```

Out[143]:
```
region
northeast    26851.705085
northwest    25908.986885
southeast    24060.347682
southwest    22553.615120
Name: hospitalization charges, dtype: float64
```

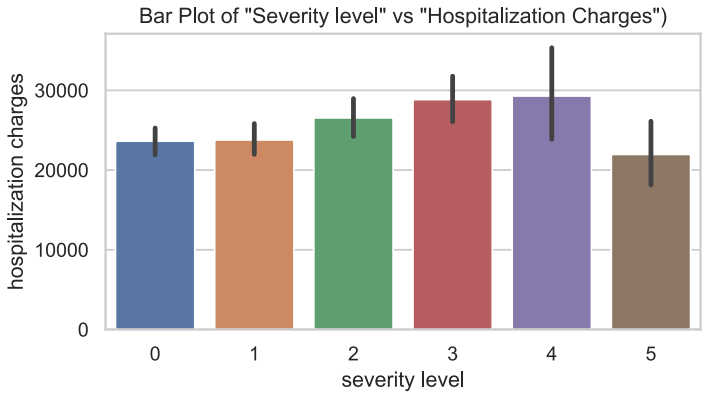
In [144]:
```
# Bar Plot showing 'Region' vs.'Hospitalization Charges'
fig, ax = plt.subplots(figsize = (6,3))
sns.barplot(data = AP_copy, x = 'region', y = 'hospitalization charges', ax = ax).set(title = 'Bar Plot of "Region" vs. "hospital
plt.show()
plt.show()
```



In [146]:
```
# Calculating the Severity Level-wise mean hospitalization charges
AP_copy.groupby('severity level')['hospitalization charges'].agg('mean')
```

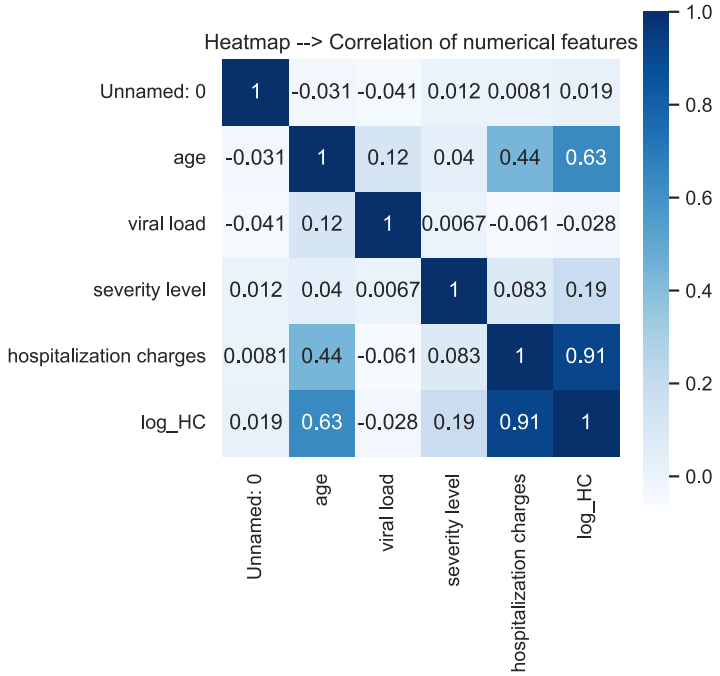
Out[146]:
```
severity level
0    23629.713462
1    23782.643599
2    26547.829268
3    28846.021739
4    29293.913043
5    21965.000000
Name: hospitalization charges, dtype: float64
```

In [148]:
```
# Bar Plot showing 'Severity Level' vs.'Hospitalization Charges'
fig, ax = plt.subplots(figsize = (6,3))
sns.barplot(data = AP_copy, x = 'severity level', y = 'hospitalization charges', ax = ax).set(title = 'Bar Plot of "Severity leve
plt.show()
```



Correlation and Heatmap

In [150]:
```
# Plotting the heatmap for the numerical features
fig, ax = plt.subplots(figsize = (5,5))
sns.heatmap(AP_copy.corr(method = 'pearson'), square = True, annot = True, cmap = 'Blues').set(title = 'Heatmap --> Correlation c
plt.show()
```



Observations/Insights from Visual Analysis

1. The Median Hospitalization Charges for both males and females is very close to each other depicting in quick-look that there is negligibledifference between them.

2. There is vast difference in the hospitalization charges of smokers as compared to non-smokers.

3. The median value of Hospitalization charges for northeast region seems higher than other regions, followed by northwest region whereas forsoutheast and southwest they look pretty much same.

4. From the plot it looks like the hospitalization charges for viral load around 10-12 are the highest but this needs to be studied further.

5. The median value of hospitalization charges seems highest for severity level 4, followed by severity level 3 and 5.

6. The boxplot of hospitalization charges for severity level 0 shows that there are huge outliers in the data, thereby leading to increase in thehospitalization charges.

7. The Mean Hospitalization Charges for Males and Females are 24437.72 and 25253.78 respectively, thereby showing that they are very closeto each other.

8. The Mean Hospitalization charges for Smokers is 55035.58 and that for Non-Smokers is 20907.97, thereby showing a huge differencebetween them.

9. The Mean Hospitalization Charges for Northeast region is 26851.70 which is the highest, followed by Northwest region with 25908.90 and islowest for Southwest region.

10. The Mean Hospitalization Charges for Severity Level 4 is 29293.90 which is the highest, followed by Severity Level 3 with 28846.02.

Statistical Analysis

Hypothesis Testing

A. Hypothesis Testing to Prove (or disprove) that the hospitalization of people who smoke is greater than those who don't.

Null Hypothesis (H0): The mean of hospitalization charges of people who smoke is SAME as those who don't smoke.

Alternate Hypothesis (Ha): The mean of hospitalization charges of people who smoke is GREATER than those who don't smoke.

Assumed Significance level (alpha): 0.05

This means that if the p-value of the tests is less than the assumed significance level, we will REJECT the Null Hypothesis and vice-versa.

Case-relevant Assumptions of T-Test

1. Data in each group should be NORMALLY Distributed.
2. Data values should be INDEPENDENT.
3. The VARIANCES of the two independent groups should be EQUAL.

Thus we will need to do NORMALITY Test as well as EQUI-VARIANCE Test.

For Normality, we will do SHAPIRO-WILK'S Test. For Equi-Variance, we will do LEVENE'S Test.

SHAPIRO-WILK'S TEST:

For Shapiro-Test:

Null Hypothesis (H0): Sample follows a Gaussian distribution.

Alternate Hypothesis (Ha): Sample does not follow a Gaussian distribution.

In [155]:
```
# Shapiro-Wilk's Test for 'log_HC'

test_stat_sh1, p_val_1 = stats.shapiro(AP_copy['log_HC'])
print(f"Test Statistics of Shapiro Test: {test_stat_sh1}, P-value of Shapiro Test: {p_val_1}")
if p_val_1 < 0.05:
    print('Sample follows a Gaussian Distribution')
else:
    print('Sample does not follow a Gaussian Distribution')
```

```
Test Statistics of Shapiro Test: 0.9769867062568665, P-value of Shapiro Test: 7.577407668338254e-13
Sample follows a Gaussian Distribution
```

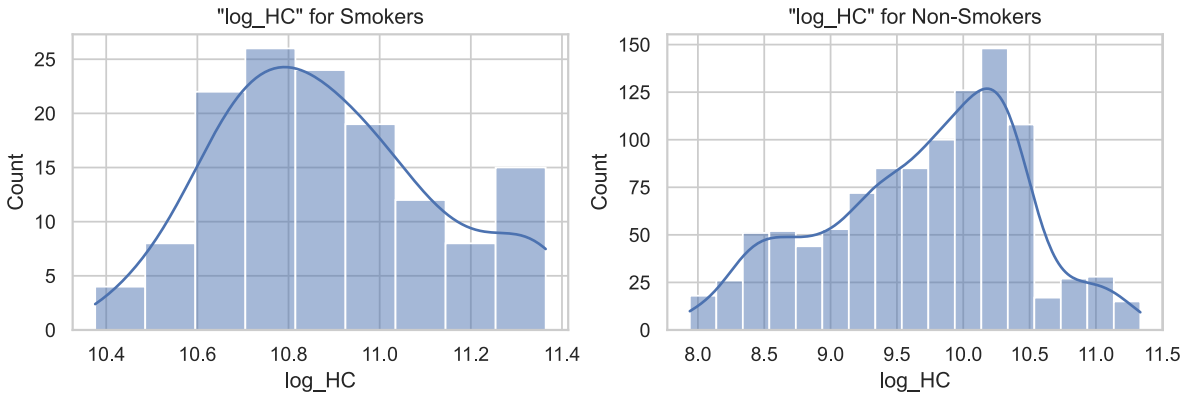
In [156]:
```
# Describing the feature 'log_HC' for smoker and non-smoker groups
AP_copy.groupby('smoker')['log_HC'].describe()
```

Out[156]:

smoker	count	mean	std	min	25%	50%	75%	max
no	1055.0	9.700799	0.738870	7.939159	9.206583	9.815366	10.253704	11.334755
yes	138.0	10.886460	0.239871	10.375801	10.709617	10.862768	11.035665	11.364217

In [157]:
```
temp1 = AP_copy[AP_copy['smoker'] == 'yes']
temp2 = AP_copy[AP_copy['smoker'] == 'no']
```

In [158]:
```
plt.figure(figsize = (11,3))
plt.subplot(121)
sns.histplot(data = temp1, x = 'log_HC', kde = True).set(title = '"log_HC" for Smokers')
plt.subplot(122)
sns.histplot(data = temp2, x = 'log_HC', kde = True).set(title = '"log_HC" for Non-Smokers')
plt.show()
```



LEVENE'S TEST:

For Levene's Test:

Null Hypothesis (H0): The variance of feature 'log_HC' for smoker and non-smoker will be equal.

Alternate Hypothesis (Ha): The variance of feature 'log_HC' for smoker and non-smoker will not be equal.

In [160]:
```
# Creating the groups of smokers and non-smokers by taking the minimum common number of samples
smoker_grp1 = AP_copy[AP_copy['smoker'] == 'yes']['log_HC'].sample(138)
smoker_grp2 = AP_copy[AP_copy['smoker'] == 'no']['log_HC'].sample(138)
# Levene's test for checking Equi-Variance of features mentioned
test_stat_l1, p_val_l1 = stats.levene(smoker_grp1, smoker_grp2, center = 'median')
print(f"Test Statistic for Levene's Test: {test_stat_l1}, P-value for Levene's Test: {p_val_l1}")
if p_val_l1 < 0.05:
    print('Both the samples do not have equal variance')
else:
    print('Both the samples have equal variance')
```

```
Test Statistic for Levene's Test: 92.96182027241173, P-value for Levene's Test: 3.936152694754359e-19
Both the samples do not have equal variance
```

Here in the Levene's Test, we get to know that the samples do not have equal variance.

So the assumptions of a T-Test does not follow here.

Thuswe will move on to a Non-Parametric Test here. The test to be used in this scenario is Kruskal-Wallis Test.

In [161]:
```
# Doing Kruskal-Wallis Test just to test the similarity between the distributions of the given samples
kr_test_stat_1, kr_p_val_1 = stats.kruskal(smoker_grp1, smoker_grp2)
print(f"Test Statistic for Kruskal-Wallis Test: {kr_test_stat_1}, P-value for Kruskal-Wallis Test: {kr_p_val_1}")
if kr_p_val_1 < 0.05:
    print("The Distributions of the given samples are NOT EQUAL")
else:
    print("The Distributions of the given samples are EQUAL")
```

```
Test Statistic for Kruskal-Wallis Test: 170.16690438331307, P-value for Kruskal-Wallis Test: 6.803196386933247e-39
The Distributions of the given samples are NOT EQUAL
```

Test Statistic for Kruskal-Wallis Test: 182.60428972435835, P-value for Kruskal-Wallis Test: 1.308640183772956e-41 The Distributions of the given samples are NOT EQUAL

In [162]:
```
# Doing a Right Tailed T-Test to test the Hypothesis even if the assumptions of T-Test failed.
t_test_stat_1, t_p_val_1 = stats.ttest_ind(smoker_grp1, smoker_grp2, alternative = 'greater')
print(f"Test Statistic for Right-Tailed T-Test: {t_test_stat_1}, P-value of Right-Tailed T-Test: {t_p_val_1}")
if t_p_val_1 < 0.05:
    print('Mean of hospitalization charges of people who smoke is GREATER than those who do not smoke')
else:
    print('Mean of hospitalization charges of people who smoke is SAME as those who do not smoke')
```

```
Test Statistic for Right-Tailed T-Test: 17.975657077976106, P-value of Right-Tailed T-Test: 1.460912057968161e-48
Mean of hospitalization charges of people who smoke is GREATER than those who do not smoke
```

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

Null Hypothesis (H0): The mean viral load of females is SAME as that of males.

Alternate Hypothesis (Ha): The mean viral load of females is DIFFERENT from that of males.

Assumed Significance level (alpha): 0.05

This means that if the p-value of the tests is less than the assumed significance level, we will REJECT the Null Hypothesis and vice-versa.

The plot for 'viral load' seems Gaussian in quick view but we will do a statistical test to confirm this. Since we have to prove that the viral load of females is different than those of males, we will need to do a "T-Test Two-Tailed".

But before that we will need to test the assumptions of a T-Test.

Case-relevant Assumptions of T-Test

1. Data in each group should be NORMALLY Distributed.
2. Data values should be INDEPENDENT.
3. The VARIANCES of the two independent groups should be EQUAL.

Thus we will need to do NORMALITY Test as well as EQUI-VARIANCE Test.

For Normality, we will do SHAPIRO-WILK'S Test. For Equi-Variance, we will do LEVENE'S Test.

SHAPIRO-WILK'S TEST:

For Shapiro-Test:

Null Hypothesis (H0): Sample follows a Gaussian distribution.

Alternate Hypothesis (Ha): Sample does not follow a Gaussian distribution.

In [165]:
```
# Shapiro-Wilk's Test for 'viral load'
test_stat_sh2, p_val_2 = stats.shapiro(AP_copy['viral load'])
print(f"Test Statistics of Shapiro Test: {test_stat_sh2}, P-value of Shapiro Test: {p_val_2}")
if p_val_2 < 0.05:
    print('Sample follows a Gaussian Distribution')
else:
    print('Sample does not follow a Gaussian Distribution')
```

```
Test Statistics of Shapiro Test: 0.992615818977356, P-value of Shapiro Test: 1.1277620615146589e-05
Sample follows a Gaussian Distribution
```

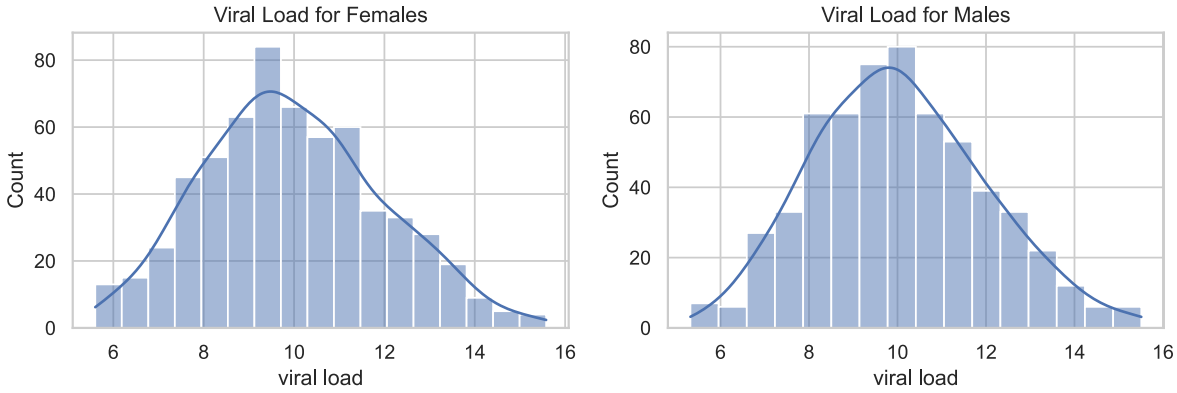
In [166]:
```
# Describing the Gender-wise 'viral load'
AP_copy.groupby('sex')['viral load'].describe()
```

Out[166]:

sex	count	mean	std	min	25%	50%	75%	max
female	611.0	9.968298	1.968264	5.60	8.5900	9.86	11.165	15.58
male	582.0	10.032440	1.943610	5.32	8.6025	9.94	11.285	15.51

In [168]:
```
temp3 = AP_copy[AP_copy['sex'] == 'female']
temp4 = AP_copy[AP_copy['sex'] == 'male']
```

In [169]:
```
plt.figure(figsize = (11,3))
plt.subplot(121)
sns.histplot(data = temp3, x = 'viral load', kde = True).set(title = 'Viral Load for Females')
plt.subplot(122)
sns.histplot(data = temp4, x = 'viral load', kde = True).set(title = 'Viral Load for Males')
plt.show()
```



LEVENE'S TEST:

For Levene's Test:

Null Hypothesis (H0): The variance of feature 'viral load' for females and males will be equal.

Alternate Hypothesis (Ha): The variance of feature 'viral load' for females and males will not be equal.

In [170]:
```
# Creating the groups of viral load of females and males by taking the minimum common number of samples
Gender_grp1 = AP_copy[AP_copy['sex'] == 'female']['viral load'].sample(582)
Gender_grp2 = AP_copy[AP_copy['sex'] == 'male']['viral load'].sample(582)
# Levene's test for checking Equi-Variance of features mentioned
test_stat_l2, p_val_l2 = stats.levene(Gender_grp1, Gender_grp2, center = 'median')
print(f"Test Statistic for Levene's Test: {test_stat_l2}, P-value for Levene's Test: {p_val_l2}")
if p_val_l2 < 0.05:
    print('Both the samples do not have equal variance')
else:
    print('Both the samples have equal variance')
```

```
Test Statistic for Levene's Test: 0.8020731672358477, P-value for Levene's Test: 0.37065972948637693
Both the samples have equal variance
```

Here the assumptions of T-Test have been satisfied and proved above.

Hence we will proceed with Two-Tailed T-Test (since we have to prove/disprove viral load of females is different from viral load of males).

In [171]:
```
# Doing a Two Tailed T-Test to test the Hypothesis.
t_test_stat_2, t_p_val_2 = stats.ttest_ind(Gender_grp1, Gender_grp2, alternative = 'two-sided')
print(f"Test Statistic for Two-Tailed T-Test: {t_test_stat_2}, P-value of Two-Tailed T-Test: {t_p_val_2}")
if t_p_val_2 < 0.05:
    print('Mean viral load of females is DIFFERENT from that of the males')
else:
    print('Mean viral load of females is SAME as that of the males')
```

```
Test Statistic for Two-Tailed T-Test: -0.6260344023293654, P-value of Two-Tailed T-Test: 0.5314153232636851
Mean viral load of females is SAME as that of the males
```

C. Is the proportion of smoking significantly different across different regions ?

Null Hypothesis (H0): Features 'smoker' and 'region' are INDEPENDENT of each other

(No association between'smoker' and 'region') i.e. Proportion of Smokers is SAME across all regions.

Alternate Hypothesis (Ha): Features 'smoker' and 'region' DEPEND on each other (There is association between 'smoker' and 'region') i.e. Proportion of Smokers is DIFFERENT across different regions.

Assumed Significance level (alpha): 0.05

This means that if the p-value of the tests is less than the assumed significance level, we will REJECT the Null Hypothesis and vice-versa.

Here we have to find association between two CATEGORICAL Variables.

Thus, we will perform a CHI-SQUARE Test here. But first we will check for the assumtions of Chi-Square Test.

Case-Relevant Assumptions for Chi-Square Test:

1. Both the features/variables are categorical.
2. All the observations are independent.
3. Cells in the Contingency table are Mutually Exclusive.
4. Expected value of cells should be 5 or greater in at least 80 % of the cells and that no cell should have expected value less than 1.

All the assumptions mentioned above are satisfied and hence we will proceed for the Chi-Square Test.

Describing the region-wise smoker

In [173]:
```
AP_copy.groupby('region')['smoker'].value_counts()
```

Out[173]:
```
region     smoker
northeast  no        256
           yes        39
northwest  no        267
           yes        38
southeast  no        267
           yes        35
southwest  no        265
           yes        26
Name: smoker, dtype: int64
```

In [176]:
```
# Preparing a Contingency Table of 'smoker' and 'region'
Q3_contingency = pd.crosstab(AP_copy['smoker'], AP_copy['region'])
chi_stat_3, p_val_3, dof_3, expected_val_3 = stats.chi2_contingency(Q3_contingency)
print(f"Test Statistic for Chi-Square Test: {chi_stat_3}, P-value for Chi-Square Test: {p_val_3}")
if p_val_3 < 0.05:
    print('Proportion of smokers is DIFFERENT across different regions')
else:
    print('Proportion of smokers is SAME across all regions')
```

```
Test Statistic for Chi-Square Test: 2.99680663546149, P-value for Chi-Square Test: 0.39211779235957156
Proportion of smokers is SAME across all regions
```

D. Is the mean viral load of women with 0 Severity level , 1 Severity level, and 2 Severity level the same?

Null Hypothesis (H0): Mean viral load of women is SAME across severity level 0, 1 and 2.

Alternate Hypothesis (Ha): Mean viral load of women is DIFFERENT across severity level 0, 1 and 2.

Assumed Significance level (alpha): 0.05This means that if the p-value of the tests is less than the assumed significance level, we will REJECT the Null Hypothesis and vice-versa.

Here we have to compare the means of more than two groups. The best test in thi scase is a ONE-WAY ANOVA. But first we need to check for theassumptions of ANOVA.

Case-relevant Assumptions of ANOVA Test

1. Data in each group should be NORMALLY Distributed.
2. Data values should be independent.
3. The Variances of all the different independent groups should be EQUAL.

Thus we will need to do NORMALITY Test as well as EQUI-VARIANCE Test.

For Normality, we will do SHAPIRO-WILK'S Test. For Equi-Variance, we will do LEVENE'S Test.

SHAPIRO-WILK'S TEST:

For Shapiro-Test:

Null Hypothesis (H0): Sample follows a normal distribution.Alternate Hypothesis

(Ha): Sample does not follow a normal distribution.

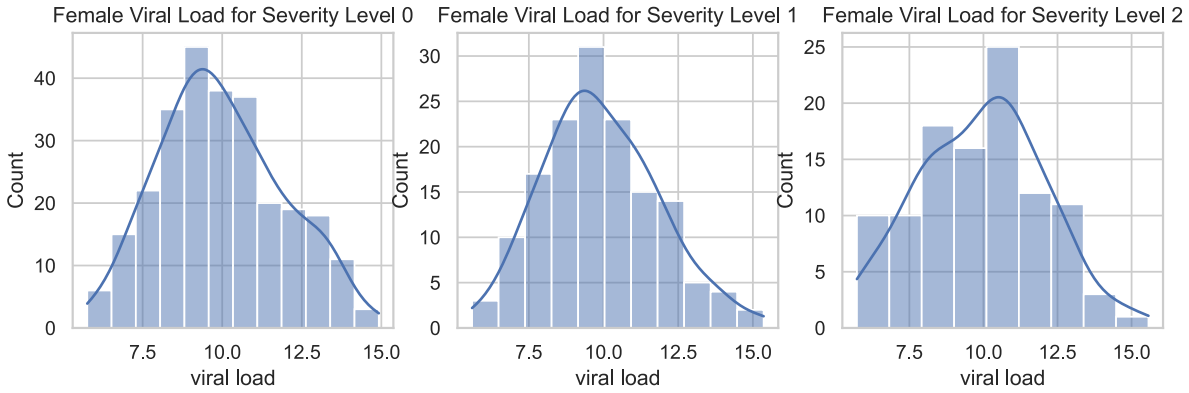
In [179]:
```
# Describing the severity level-wise 'viral load' of females

AP_copy[AP_copy['sex'] == 'female'].groupby('severity level')['viral load'].describe()[:3]
```

Out[179]:

severity level	count	mean	std	min	25%	50%	75%	max
0	269.0	9.967212	1.934359	5.76	8.610	9.70	11.1300	14.92
1	147.0	9.908844	1.918987	5.60	8.600	9.67	11.1300	15.36
2	106.0	9.945000	2.092305	5.73	8.345	10.06	11.2025	15.57

In [181]:
```
temp5 = AP_copy[(AP_copy['sex'] == 'female') & (AP_copy['severity level'] == 0)]
temp6 = AP_copy[(AP_copy['sex'] == 'female') & (AP_copy['severity level'] == 1)]
temp7 = AP_copy[(AP_copy['sex'] == 'female') & (AP_copy['severity level'] == 2)]
```

In [182]:
```
plt.figure(figsize = (11,3))
plt.subplot(131)
sns.histplot(data = temp5, x = 'viral load', kde = True).set(title = 'Female Viral Load for Severity Level 0')
plt.subplot(132)
sns.histplot(data = temp6, x = 'viral load', kde = True).set(title = 'Female Viral Load for Severity Level 1')
plt.subplot(133)
sns.histplot(data = temp7, x = 'viral load', kde = True).set(title = 'Female Viral Load for Severity Level 2')
plt.show()
```



LEVENE'S TEST:

For Levene's Test:

Null Hypothesis (H0): The variance of feature 'viral load' for females is EQUAL for severity level 0,1,2.

Alternate Hypothesis (Ha): The variance of feature 'viral load' for females is DIFFERENT for severity level 0,1,2.

In [194]:
```
# Levene's Test by taking minimum common number of samples
severity_grp1 = AP_copy[(AP_copy['sex'] == 'female') & (AP_copy['severity level'] == 0)]['viral load'].sample(86)
severity_grp2 = AP_copy[(AP_copy['sex'] == 'female') & (AP_copy['severity level'] == 1)]['viral load'].sample(86)
severity_grp3 = AP_copy[(AP_copy['sex'] == 'female') & (AP_copy['severity level'] == 2)]['viral load'].sample(86)
test_stat_l4, p_val_l4 = stats.levene(severity_grp1, severity_grp2, severity_grp3, center = 'median')
print(f"Test Statistic for Levene's Test: {test_stat_l4}, P-value for Levene's Test: {p_val_l4}")
if p_val_l4 < 0.05:
    print('At least one of the samples does not have equal variance')
else:
    print('All the samples have equal variance')
```

```
Test Statistic for Levene's Test: 0.7216090244139409, P-value for Levene's Test: 0.48695933562834814
All the samples have equal variance
```

The assumptions of ONE-WAY ANOVA have been satisfied here as is evident from Shapiro-Wilk's Test and Levene's Test.

Now we can proceed with the ONE-WAY ANOVA Test.

In [195]:
```
# Performing One-Way ANOVA for mentioned features
test_stat_an4, p_val_an4 = stats.f_oneway(severity_grp1,severity_grp2,severity_grp3)
print(f"Test Statistic for ANOVA Test: {test_stat_an4}, P-value for ANOVA Test: {p_val_an4}")
if p_val_an4 < 0.05:
    print('Mean viral load of women is DIFFERENT across severity level 0, 1 and 2')
else:
    print('Mean viral load of women is SAME across severity level 0, 1 and 2')
```

```
Test Statistic for ANOVA Test: 0.22641991207838744, P-value for ANOVA Test: 0.7975433354076947
Mean viral load of women is SAME across severity level 0, 1 and 2
```

Observations/Insights from Hypothesis Testing

1. The Mean Hospitalization Charges for people who smoke is GREATER than people who don't smoke.
2. The Mean Viral Load for Males and Females is SAME.
3. Proportion of Smokers is SAME across all regions.
4. The Mean Viral Load of Females is SAME for Severity Level 0, 1 and 2.

Recommendations

1. Since the mean hospitalization charges for smokers is very high as compared to non-smokers, special focus should be given to smokers by accompanying the hospitalization charges for smokers along with some post-treatment de-addiction programmes so that they can be allured to the hospitals for timely check-up and smoking de-addiction. This will help to come out of the addiction to smoking and also early detection of any dangerous diseases caused due to smoking is also possible by taking such measures.

2. Apollo Hospitals can organize health camps and de-addiction camps on a regular and timely basis in order raise the awareness of people and encourage them for their timely treatment. These camps can highight the various health issues which are a direct result of smoking and also the camps can show the kind of treatment and health routine that is required to come out of the addiction to smoking and recover from health ailments.

3. The mean hospitalization charges is high in Northeast region compared to other regions even when the proportion of smokers is same across all regions. To bring this hospitalization charge down and encourage more people to opt for regular check-ups, more diagnostic centers should be opened in densely populated areas and home-collection of test samples can be made available.

4. Since the mean hospitalization charges for severity level 4 is the highest, followed by severity level 3 which indicates that there is slight delay in response time in going for treatment by people as they went to the hospitals somewhat late for treatment, thereby leading to increment of severity level followed by hospitalization charges as well. Apollo Hospitals can use this scenario to encourage and make aware more people.

5. Since the mean viral load of females is same for severity levels 0,1 and 2, this signifies that females are more cautious about infections andare engaged in regular and timely check-ups. Apollo Hospital can advertise such scenarios and examples of early diagnosis and treatment toencourage more people to avoid severe cases.

6. To reduce the severity of infection, primarily due to smoking cases, regular and repeated test screenings/investigation should be encouragedand smokers should be made aware of the benefits of early detection of symptoms in these screenings which might lead to diminish theseverity level and thereby reducing the overall hospitalization charges.

In [ ]: