Apollo_hypothesis_testing

October 7, 2022

1 Business Case: Apollo Hospitals - Hypothesis Testing

1.1 Business context

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.

1.2 Problem Statement

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

The given dataset consists of the following columns.

- *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

1.3 Solution approach (additional views)

We will being with data import and basic EDA (including uni-variate and bi-variate analysis). We will then perform missing value and outlier analysis/treatment. Since the focus of this case-study is on hypothesis testing, we will evaluate various outliers treatment options with their pros and cons. We will then attempt to answer a few questions using various hypothesis tests. We will also attempt to identify significant factors impacting hospitalization charges and confirm the findings with the help of hypothesis tests and visualization plots. Finally, we will summarize the high level insights and potential recommendations.

2 Solution

[117]:
```
#common imports
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
import warnings

warnings.filterwarnings('ignore')
sns.set_theme()
```

2.1 Read Data

[118]:
```
data = df = pd.read_csv("data/apollo_hospitals.csv", index_col=0)
```

[119]:
```
df.head()
```

[119]:
```
   age     sex smoker     region  viral load  severity level  \
0   19  female    yes  southwest        9.30               0
1   18    male     no  southeast       11.26               1
2   28    male     no  southeast       11.00               3
3   33    male     no  northwest        7.57               0
4   32    male     no  northwest        9.63               0

   hospitalization charges
0                    42212
1                     4314
2                    11124
3                    54961
4                     9667
```

[120]:
```
df.tail()
```

[120]:
```
      age     sex smoker     region  viral load  severity level  \
1333   50    male     no  northwest       10.32               3
```

```
1334   18  female     no  northeast       10.64               0
1335   18  female     no  southeast       12.28               0
1336   21  female     no  southwest        8.60               0
1337   61  female    yes  northwest        9.69               0

      hospitalization charges
1333                    26501
1334                     5515
1335                     4075
1336                     5020
1337                    72853
```

[121]:

```
df.shape
```

[121]:

```
(1338, 7)
```

[122]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 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: 83.6+ KB
```

[123]:

```
#check for missing values
df.isna().sum()
```

[123]:

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

[124]:

```
df.nunique()
```

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

```
[125]: df.describe(include='all')
```

```
[125]:                 age   sex smoker     region   viral load  severity level  \
       count   1338.000000  1338   1338       1338  1338.000000     1338.000000
       unique          NaN     2      2          4          NaN             NaN
       top             NaN  male     no  southeast          NaN             NaN
       freq            NaN   676   1064        364          NaN             NaN
       mean      39.207025   NaN    NaN        NaN    10.221233        1.094918
       std       14.049960   NaN    NaN        NaN     2.032796        1.205493
       min       18.000000   NaN    NaN        NaN     5.320000        0.000000
       25%       27.000000   NaN    NaN        NaN     8.762500        0.000000
       50%       39.000000   NaN    NaN        NaN    10.130000        1.000000
       75%       51.000000   NaN    NaN        NaN    11.567500        2.000000
       max       64.000000   NaN    NaN        NaN    17.710000        5.000000

               hospitalization charges
       count               1338.000000
       unique                      NaN
       top                         NaN
       freq                        NaN
       mean               33176.058296
       std                30275.029296
       min                 2805.000000
       25%                11851.000000
       50%                23455.000000
       75%                41599.500000
       max               159426.000000
```

```
[126]: # check how categorical variables are distributed across various levels
       n = df.shape[0]
       for col in ['sex', 'smoker', 'region', 'severity level']:
           df_vc = df[col].value_counts().reset_index()
           df_vc['% records'] = np.round((df_vc[col] * 100) / n ,2)
           print(df_vc)
           print('\n')
```

```
  index  sex  % records
0  male  676      50.52
```

```
1  female  662      49.48
```

```
  index  smoker  % records
0    no    1064      79.52
1   yes     274      20.48
```

```
       index  region  % records
0  southeast     364      27.20
1  southwest     325      24.29
2  northwest     325      24.29
3  northeast     324      24.22
```

```
   index  severity level  % records
0      0             574      42.90
1      1             324      24.22
2      2             240      17.94
3      3             157      11.73
4      4              25       1.87
5      5              18       1.35
```

Observations

1. The data-set has 1338 rows and 7 columns.
2. The data-set has **no missing values**.
3. 'sex' is a dichotomous nominal categorical variable. There are almost equal number of male and female patients.
4. 'smoker' is a dichotomous nominal categorical variable. Around 20% of the patients are 'smokers'.
5. 'region' is a nominal categorical variables with 4 levels - 'southeast', 'southwest', 'northeast', and 'northwest'. There are almost equal number of patients across all regions.
6. 'severity level' is a nominal categorical variable with 6 levels - level-0 to level-5. The most common severity level is *level-0* (at 43%), followed by *level-1* (24%), *level-2*(18%), and level-3(11.7%). The number of patients with *level-4* and *level-5* are very low (< 2% each).
7. 'age' is a continuous variable with values ranging from 18 to 64 years.
8. 'viral load' is a continuous variable with values ranging from 5.3 to 17.7
9. 'hospitalization charges' is a **'dependent' or 'target'** continuous variables with values ranging from 2805 to 159426.

2.2 Creating age-bins categorical variable

[127]:

```
age_bins = list(range(17,67,3))
df['age_bins'] = pd.cut(df['age'], bins=age_bins)

df[['age', 'age_bins']].head(10)
```

[127]:

```
   age  age_bins
0   19  (17, 20]
1   18  (17, 20]
2   28  (26, 29]
3   33  (32, 35]
4   32  (29, 32]
5   31  (29, 32]
6   46  (44, 47]
7   37  (35, 38]
8   37  (35, 38]
9   60  (59, 62]
```

2.3 Converting categorical variables to category type

[128]:

```
#convert season, weather, hoiday, and workingday to cateogrical columns
for col in ['sex', 'region', 'severity level', 'smoker']:
    df[col] = df[col].astype('category')

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1338 entries, 0 to 1337
Data columns (total 8 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   category
 6   hospitalization charges  1338 non-null   int64
 7   age_bins                 1338 non-null   category
dtypes: category(5), float64(1), int64(2)
memory usage: 49.8 KB
```

[ ]:

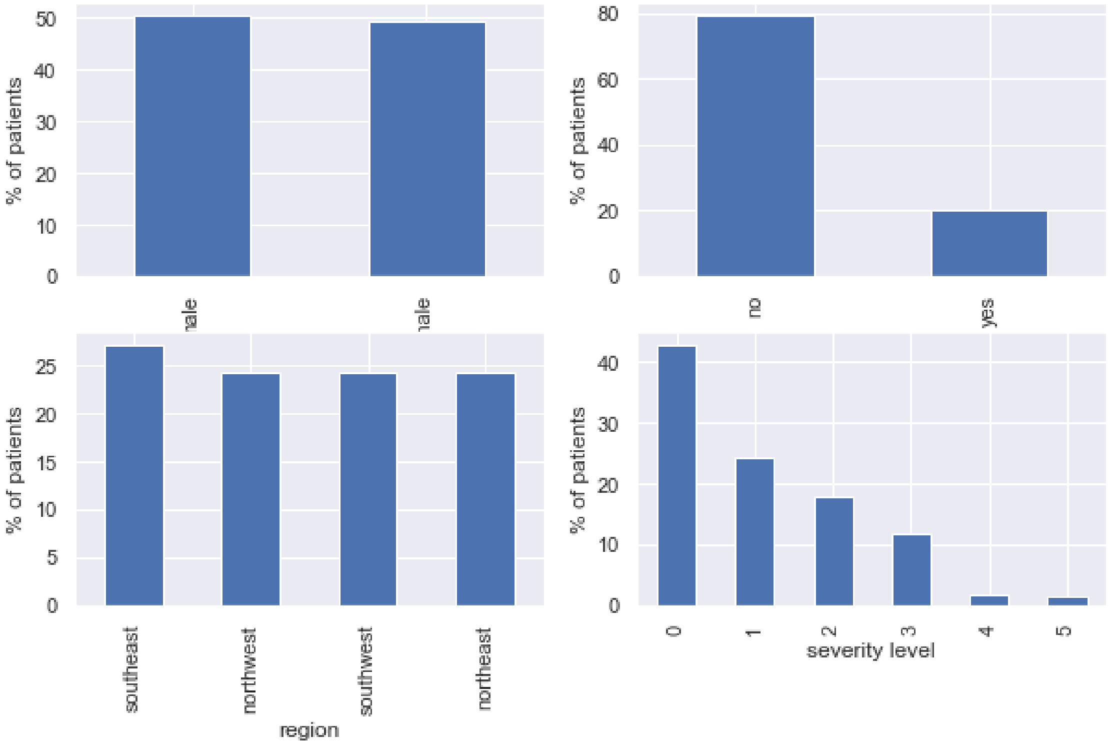
```

```

2.4 Univariate Analysis

[129]:
```
cols = {
    'cat' : ['sex', 'smoker', 'region', 'severity level'],
    'cont': ['age', 'viral load', 'hospitalization charges']
}
```

2.4.1 Categorical variables

[130]:
```
fig, ax = plt.subplots(2, 2, figsize=(10, 6))

df['sex'].value_counts(normalize=True).mul(100).plot(kind='bar', xlabel='sex',␣
 ↪ylabel='% of patients' ,ax = ax[0][0])
df['smoker'].value_counts(normalize=True).mul(100).plot(kind='bar',␣
 ↪xlabel='smoker', ylabel='% of patients', ax = ax[0][1])
df['region'].value_counts(normalize=True).mul(100).plot(kind='bar',␣
 ↪xlabel='region', ylabel='% of patients', ax = ax[1][0])
df['severity level'].value_counts(normalize=True).mul(100).plot(kind='bar',␣
 ↪xlabel='severity level', ylabel='% of patients', ax = ax[1][1])

plt.show()
```



Observations

1. There are almost equal number of male and female patients.
2. Around 20% of the patients are 'smokers'.
3. There are almost equal number of patients across all regions.
4. The most common severity level is *level-0* (at 43%), followed by *level-1* (24%), *level-2*(18%), and level-3(11.7%). The number of patients with *level-4* and *level-5* are very low (< 2% each).

2.4.2 Continuous variables

[131]:

```
#print statistial summary
df.describe()
```

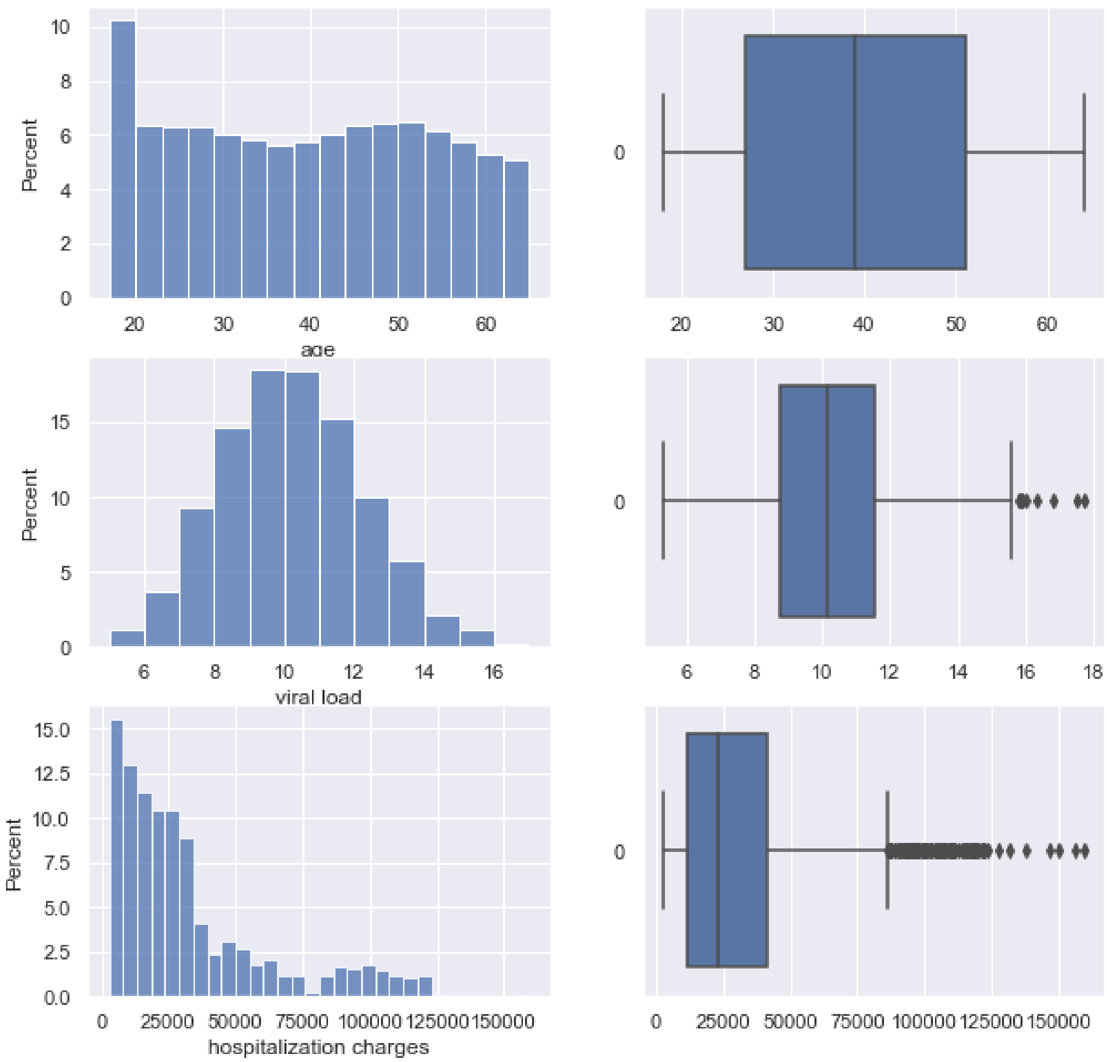
[131]:

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

[132]:

```
fig, ax = plt.subplots(3, 2, figsize=(10, 10))

sns.histplot(data=df['age'], bins=age_bins, stat='percent', ax=ax[0][0])
sns.boxplot(data=df['age'], orient="horizontal", ax=ax[0][1])

sns.histplot(df['viral load'], bins = list(range(5, 18, 1)), stat='percent',␣
 ↪ax=ax[1][0])
sns.boxplot(data = df['viral load'], orient="horizontal", ax=ax[1][1])

sns.histplot(df['hospitalization charges'], stat='percent', ax=ax[2][0])
sns.boxplot(data = df['hospitalization charges'], orient="horizontal",␣
 ↪ax=ax[2][1])

print(f'skew of "age": {df["age"].skew()}')
print(f'skew of "viral load": {df["viral load"].skew()}')
print(f'skew of "hospitalization charges": {df["hospitalization charges"].
 ↪skew()}')
```

```
skew of "age": 0.05567251565299186
skew of "viral load": 0.2835261022309636
skew of "hospitalization charges": 1.5158803706226045
```



Observations

1. 'age' is fairly uniformly distributed especially in the range 20-56 years. The IQR range = (27, 51). Median age is 39 years and mean age value is 39.02 years. The maximum number of patients (around 10%) fall in the age bracket 18-20, followed by age bracket 50-53 (around 6.5%). Overall, however, **age variable has close to zero skew and zero outliers**.

2. 'viral load' distribution look **fairly symmetrical visually** and it has **negligible skewness**. The IQR range = (8.76, 11.56). The median and mean viral load values are 10.13 and 10.22 respectively. However it has **a few outliers on the right tail of its distribution**.

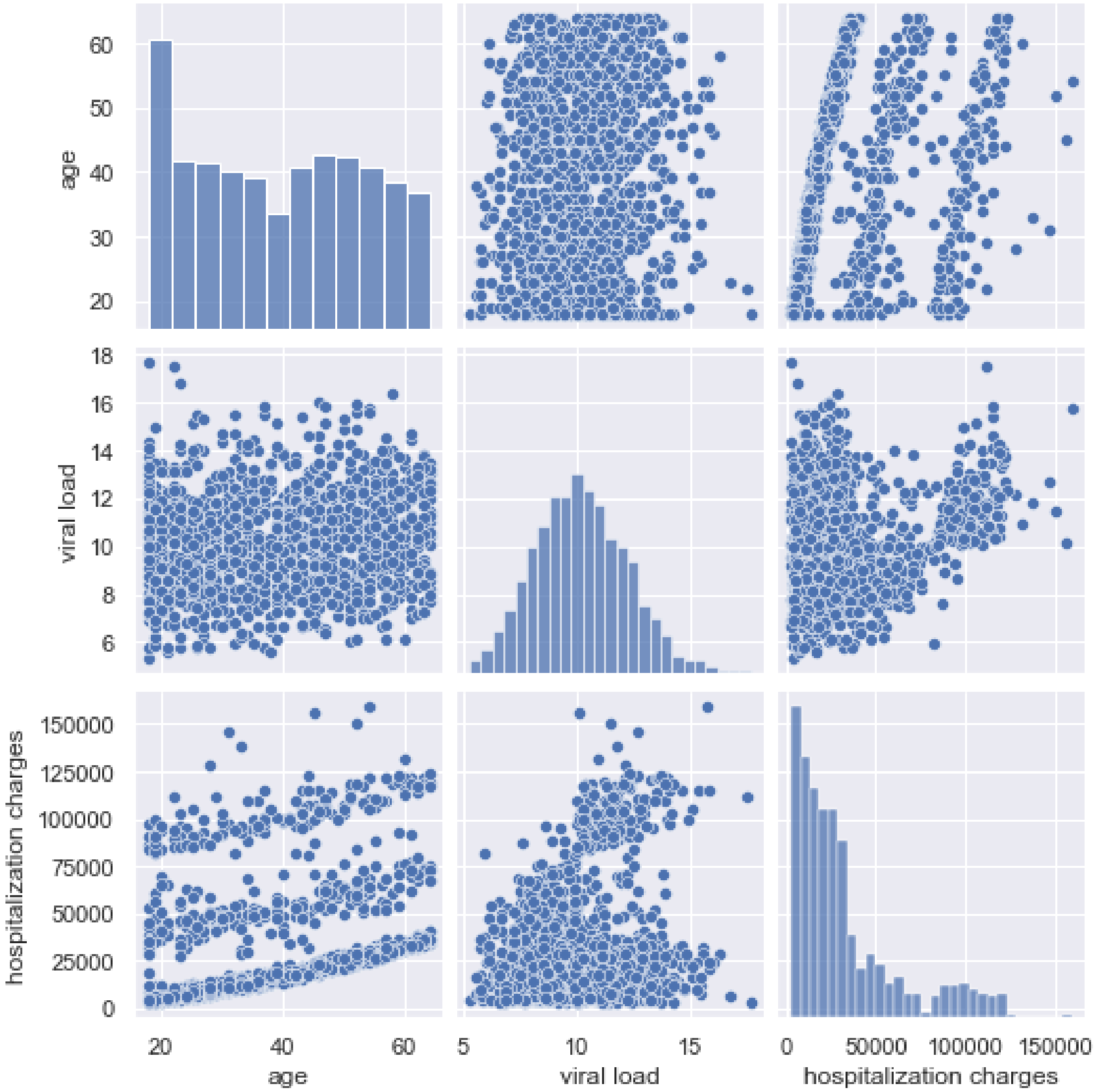
3. 'hospitalization charges' seem **positively skewed (skew factor = 1.51)** and has **several outliers on the right tail** of its distribution. The IQR range = (11851, 41599.5). The median and mean hospitalization charges are 23455 and 33176 respectively. We can observe that mean value is significantly higher than the median value which signifies positive skewness.

Note: We discuss the outliers further in the outliers treatment section.

2.5 Bivariate Analysis

2.5.1 pair plot and correlation (for continuous variables)

```
[133]: #pair plot
       sns.pairplot(df)
       plt.show()
```

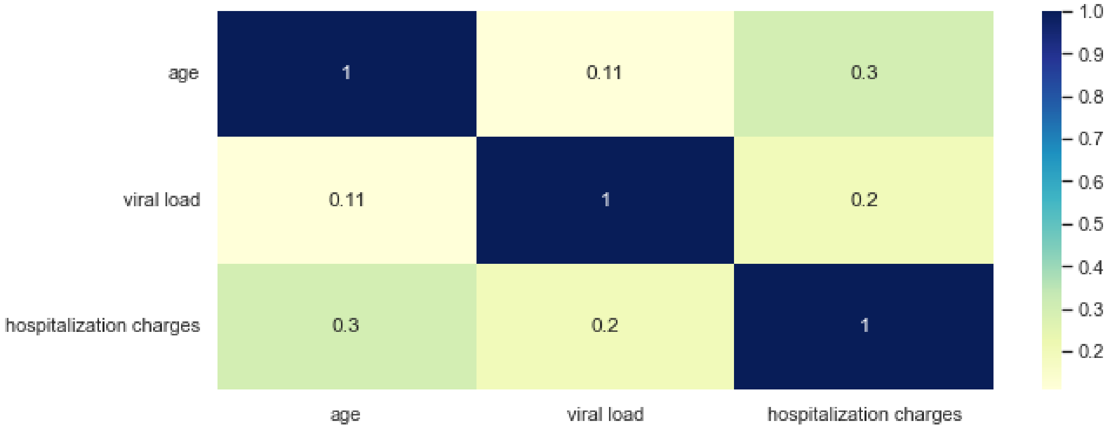


```
[134]: corr_df = df.corr(method='pearson')
       corr_df
```

```
[134]:                              age  viral load  hospitalization charges
       age                     1.000000    0.109300                 0.299008
       viral load              0.109300    1.000000                 0.198388
```

```
hospitalization charges  0.299008   0.198388                 1.000000
```

```
[135]: plt.figure(figsize=(10,4))
       sns.heatmap(corr_df, cmap="YlGnBu", annot=True)
       plt.show()
```



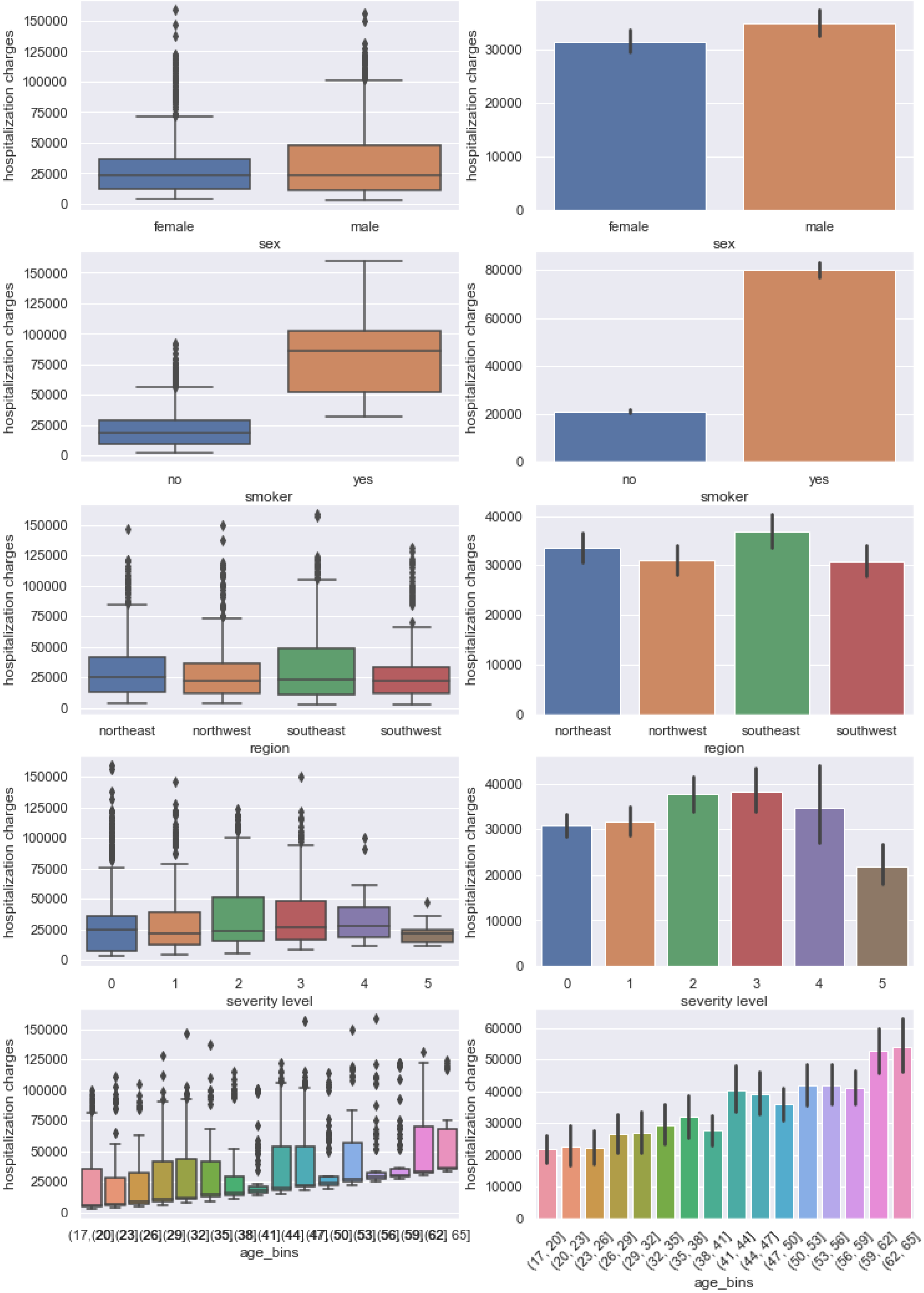
Observations

1. There is a weak positive correlation (~0.3) between ‘age’ and ‘hospitalization charges’.
2. The correlation between ‘viral load’ and ‘hospitalization charges’ is negligible. Similarly the correlation between the ‘viral load’ and ‘age’ is negligible as well.

2.5.2 Hospitalization charges by categorical features

```
[136]: fig, ax = plt.subplots(5, 2, figsize=(12, 18))

       sns.boxplot(x='sex', y='hospitalization charges', data=df, ax=ax[0][0])
       sns.barplot(x='sex', y='hospitalization charges', data=df, ax=ax[0][1])

       sns.boxplot(x='smoker', y='hospitalization charges', data=df, ax=ax[1][0])
       sns.barplot(x='smoker', y='hospitalization charges', data=df, ax=ax[1][1])

       sns.boxplot(x='region', y='hospitalization charges', data=df, ax=ax[2][0])
       sns.barplot(x='region', y='hospitalization charges', data=df, ax=ax[2][1])

       sns.boxplot(x='severity level', y='hospitalization charges', data=df,␣
        ↪ax=ax[3][0])
       sns.barplot(x='severity level', y='hospitalization charges', data=df,␣
        ↪ax=ax[3][1])

       plt.xticks(rotation=45)
       sns.boxplot(x='age_bins', y='hospitalization charges', data=df, ax=ax[4][0])
sns.barplot(x='age_bins', y='hospitalization charges', data=df, ax=ax[4][1])
```

```
[136]: <AxesSubplot:xlabel='age_bins', ylabel='hospitalization charges'>
```



[137]:

```
print(df.groupby('sex')['hospitalization charges'].agg(['median', 'mean']))
print(df.groupby('smoker')['hospitalization charges'].agg(['median', 'mean']))
print(df.groupby('region')['hospitalization charges'].agg(['median', 'mean']))
print(df.groupby('severity level')['hospitalization charges'].agg(['median',␣
 ↪'mean']))
print(df.groupby('age_bins')['hospitalization charges'].agg(['median', 'mean']))
```

```
         median          mean
sex
female  23532.5  31423.945619
male    23424.0  34891.884615
         median          mean
smoker
no      18363.5  21085.675752
yes     86141.0  80125.572993
            median          mean
region
northeast  25144.0  33515.966049
northwest  22414.0  31043.941538
southeast  23235.5  36838.541209
southwest  21996.0  30867.332308
                 median          mean
severity level
0               24642.5  30914.940767
1               21209.5  31827.935185
2               23162.5  37683.908333
3               26501.0  38388.305732
4               27584.0  34626.680000
5               21474.0  21965.000000
           median          mean
age_bins
(17, 20]    5506.0  21783.722892
(20, 23]    6777.5  22636.035714
(23, 26]    8472.0  22183.452381
(26, 29]   11085.0  26407.301205
(29, 32]   11822.5  26827.000000
(32, 35]   14617.0  29408.194805
(35, 38]   16370.0  31939.306667
(38, 41]   17992.0  27625.253165
(41, 44]   20058.0  40156.432099
(44, 47]   22063.0  39022.310345
(47, 50]   24474.0  35873.720930
(50, 53]   27068.5  41651.209302
(53, 56]   29165.5  41755.675000
(56, 59]   30582.0  41019.671053
(59, 62]   33677.0   52639.768116
(62, 65]   36187.0   53856.488889
```

Observations

1. The median ‘hospitalization charges’ for male and female patients are both around 25K. However, the mean ‘hospitalization charges’ for ‘male’ patients (around 34.8K) is somewhat higher than that of ‘female’ patients (around 31.4K). We may need a hypothesis test to determine if this difference is statistically significant (refer to additional test section).
2. There is a significant visual difference between both the mean and median values of ‘hospitalization charges’ for ‘smoker’ and ‘non-smoker’ patients. Thus it appears that ‘smoking’ may be a significant factor in determining ‘hospitalization charges’. We will confirm this in the statistical test section.
3. The median ‘hospitalization charges’ across all four regions are around 25K. However, the mean ‘hospitalization charges’ across four regions are somewhat different with ‘southeast’ being highest at 36.8K and ‘southwest’ being lowest at 30.8K. We may need a hypothesis test to determine if the differences in means are statistically significant (refer to additional tests section).
4. The lowest median ‘hospitalization charges’ is 21.2K for severity level-1 and highest median charge is 27.5K for level-4. Similarly, highest mean charge is 38.3K for level-3 while lowest mean charge is 21.9K for severity level-5. We may need a hypothesis test to determine if the differences in means are statistically significant (refer to additional test section).
5. The lowest median ‘hospitalization charges’ is 5.5K for age group (17-21] and highest median ‘hospitalization charges’ is 36K for (62,65] age group. Similarly, lowest and highest mean values are 21.7K and 53.8K for age groups (17,21] and (62-65] respectively.

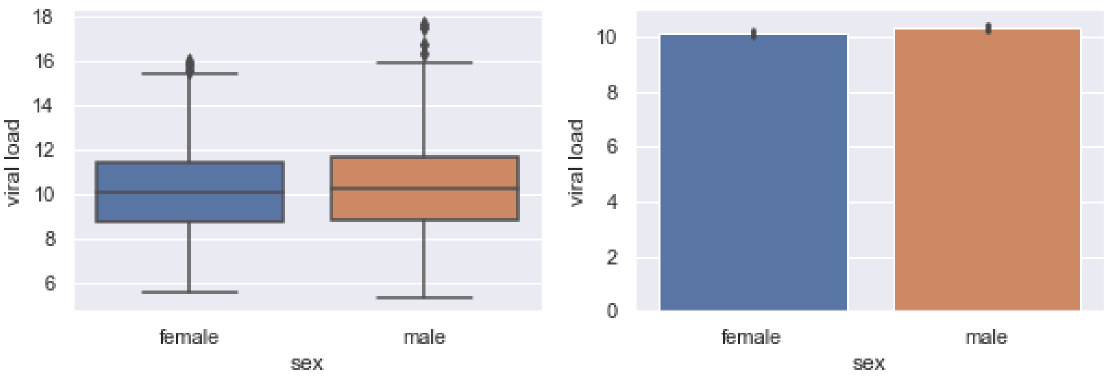
2.5.3 Viral load by gender

[138]:

```
fig, ax = plt.subplots(1, 2, figsize=(10, 3))
sns.boxplot(x='sex', y='viral load', data=df, ax=ax[0])
sns.barplot(x='sex', y='viral load', data=df, ax=ax[1])
```

[138]:

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



Observations

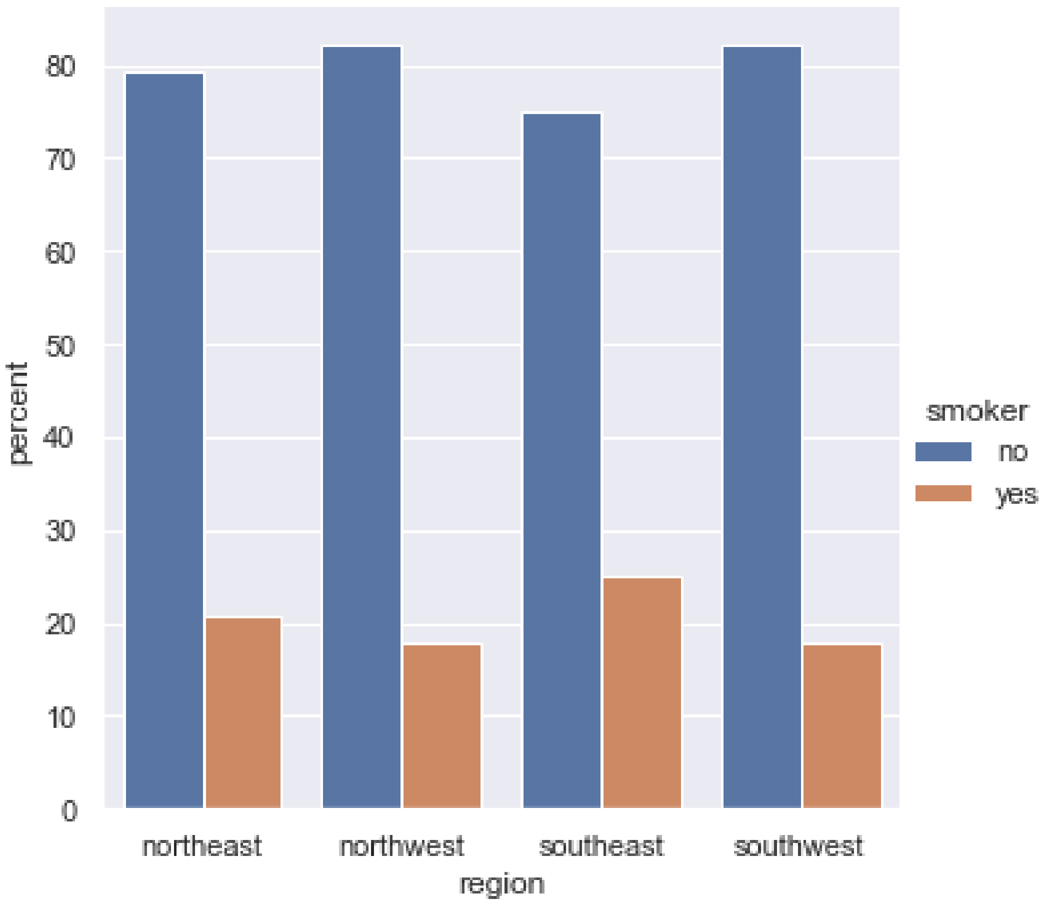
The median and mean values of ‘viral load’ for both female and male patients appear equal visually and hence gender doesn’t seem to be a significant factor in determining viral load value. We will confirm this in the statistical tests section.

2.5.4 Proportion of Smoker patients for different regions

```
[139]: df_smoke_by_region = df.groupby('region')['smoker'].
        ↪value_counts(normalize=True).mul(100).rename('percent').reset_index().
        ↪rename(columns={'level_1': 'smoker'})
       print(df_smoke_by_region)
       df_smoke_by_region.pipe((sns.catplot,'data'),␣
        ↪x='region',y='percent',hue='smoker',kind='bar')
```

```
      region smoker    percent
0  northeast     no  79.320988
1  northeast    yes  20.679012
2  northwest     no  82.153846
3  northwest    yes  17.846154
4  southeast     no  75.000000
5  southeast    yes  25.000000
6  southwest     no  82.153846
7  southwest    yes  17.846154
```

```
[139]: <seaborn.axisgrid.FacetGrid at 0x22d575e6400>
```



Observations

The proportion of smoker patients across the four regions ranges from 17.84% to 25%. We will use hypothesis test to confirm if this difference is significant.

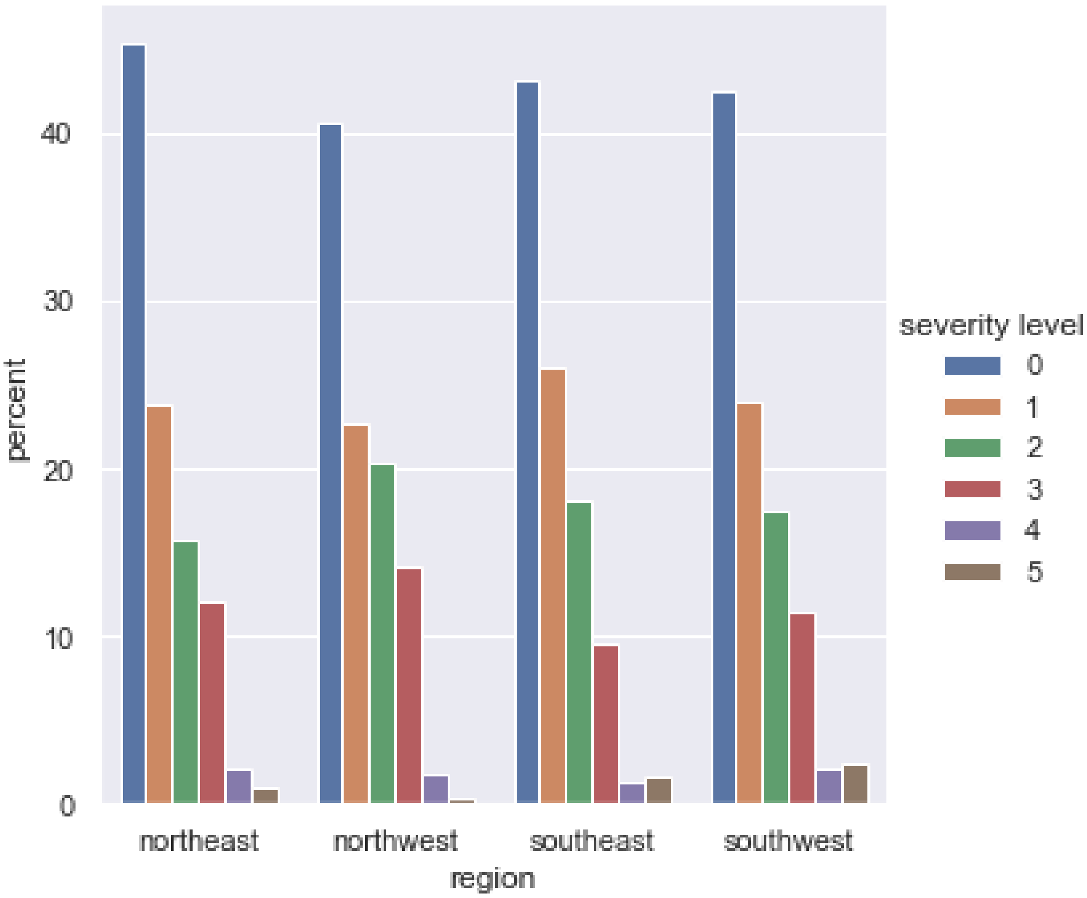
2.5.5 Severity level of patients for different regions

[140]:
```
df_sev_by_region = df.groupby('region')['severity level'].
 ↪value_counts(normalize=True).mul(100).rename('percent').reset_index().
 ↪rename(columns={'level_1': 'severity level'})
print(df_sev_by_region)
df_sev_by_region.pipe((sns.catplot,'data'),␣
 ↪x='region',y='percent',hue='severity level',kind='bar')
```

```
       region  severity level    percent
0   northeast               0  45.370370
1   northeast               1  23.765432
2   northeast               2  15.740741
3   northeast               3  12.037037
```

```
4   northeast              4   2.160494
5   northeast              5   0.925926
6   northwest              0  40.615385
7   northwest              1  22.769231
8   northwest              2  20.307692
9   northwest              3  14.153846
10  northwest              4   1.846154
11  northwest              5   0.307692
12  southeast              0  43.131868
13  southeast              1  26.098901
14  southeast              2  18.131868
15  southeast              3   9.615385
16  southeast              5   1.648352
17  southeast              4   1.373626
18  southwest              0  42.461538
19  southwest              1  24.000000
20  southwest              2  17.538462
21  southwest              3  11.384615
22  southwest              5   2.461538
23  southwest              4   2.153846
```

```
[140]: <seaborn.axisgrid.FacetGrid at 0x22d5763ff10>
```



Observation: Visually, the count of patients with different severity levels appear similar across region. However we may need statistical test to confirm this.

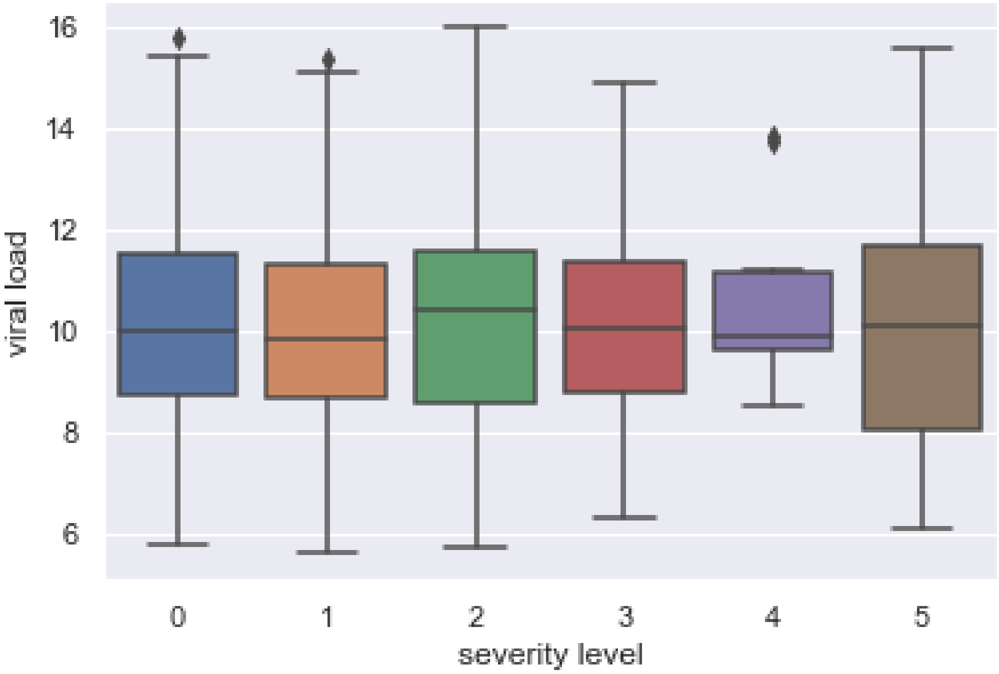
2.5.6 Viral load of female patients for different severity levels

[141]:

```
df_female = df[df['sex'] == 'female']
print(df_female.groupby('severity level')['viral load'].agg(['sum', 'count',␣
 ↪'mean']))
sns.boxplot(x='severity level', y='viral load', data=df_female)
```

```
                    sum  count       mean
severity level
0               2924.89    289  10.120727
1               1582.76    158  10.017468
2               1215.80    119  10.216807
3                781.24     77  10.145974
4                117.12     11  10.647273
5                 81.65      8  10.206250
```

```
[141]: <AxesSubplot:xlabel='severity level', ylabel='viral load'>
```



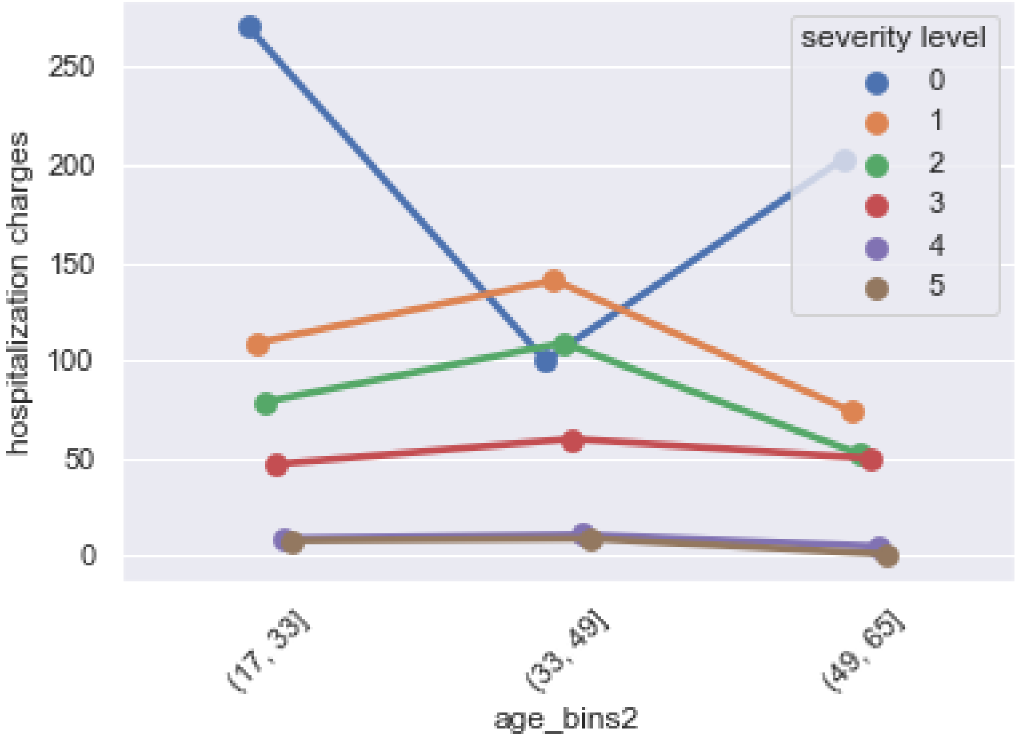
Observations

The mean viral load values for female patients with different severity levels appear close (10.01 to 10.64). We will confirm if the mean value differences are significant in the statistical tests section. "

2.5.7 Severity level and age

```
[142]: age_bins2 = range(17,67,16)
       df['age_bins2'] = pd.cut(df['age'], bins=age_bins2)
       df[['age', 'age_bins', 'age_bins2']]

       plt.xticks(rotation=45)
       sns.pointplot(data=df, x='age_bins2', y='hospitalization charges',␣
        ↪estimator=lambda x: len(x), dodge=True, hue='severity level')
```

```
[142]: <AxesSubplot:xlabel='age_bins2', ylabel='hospitalization charges'>
```



Observation - The young (17-33) and elder (49-65) patient groups have considerably higher number of severity level 0 patients than the patients with severity level 1, 2, or 3. The middle age group, however, has lower number of severity level 0 patients than patients with severity level 1 and 2.

2.6 Outliers analysis

We consider as outliers all the values which fall outside of the interval **[q1 - 1.5 * IQR, q3 + 1.5 * IQR]**, where q1 is 25% percentile value, q3 is 75% percentile value, and IQR is interquartile range which is equal to (q3-q1). We find outliers for the original features as well as for log-transformed, sqrt-transformed, and cube-root transformed data.

```
[143]: def findoutliers(arr):
           q3 = np.percentile(arr, 75)
           q1 = np.percentile(arr, 25)
           iqr = q3-q1
           ulim = q3 + 1.5*iqr
           llim = q1 - 1.5*iqr
           return pd.Series([True if((ele > ulim) or (ele < llim)) else False for ele␣
        ↪in arr])

       def makepositive(s, pos_val=0.01):
           return s.transform(lambda val: val if(val > 0) else pos_val)
```

[144]:
```
outliers = []
transformations = [
    ('  original', lambda s:s),
    ('sqrt', lambda s: makepositive(s)**(1/2)),
    ('cuberoot', lambda s: makepositive(s)**(1/3)),
    (' log', lambda s: np.log(makepositive(s)))]

total_n = df.shape[0]
for col in ['age', 'viral load','hospitalization charges']:
    for trans_name, trans_fn in transformations:
        ret = findoutliers(trans_fn(df[col]))
        outliers_n = ret.sum()
        outliers.append([col, trans_name, outliers_n, np.round((outliers_n /␣
 ↪total_n) * 100, 2)])

print(f'total number of rows in the dataset: {total_n}')
outliers_df = pd.DataFrame(data=outliers, columns=['column', 'transformation',␣
 ↪'outliers count', 'outliers as % of total rows'])

outliers_df = outliers_df.set_index(keys=['column', 'transformation'])
outliers_df.unstack()
```

```
total number of rows in the dataset: 1338
```

[144]:
```
                        outliers count                     \
transformation                original  log cuberoot sqrt
column
age                                  0    0        0    0
hospitalization charges            139    0        4   16
viral load                           9    8        3    3

                        outliers as % of total rows
transformation                             original  log cuberoot  sqrt
column
age                                            0.00  0.0     0.00  0.00
hospitalization charges                       10.39  0.0     0.30  1.20
viral load                                     0.67  0.6     0.22  0.22
```

Observations

1. The table above shows outlier percentage for each continuous column of the original data-set as well as the log transformed, square-root transformed, and cube root transformed features. **There are no outliers in the age column. Hospitalization charges on the other hand has 139 outliers (around 10.39%). Viral load has 9 outliers (around 0.67%).**
2. We also apply log, sqrt, and cube-root transformations on the data-set before counting the number of outliers. We observe that **log transformation is very effective here in eliminating the outliers in hospitalization charges column.**

2.7 Outliers treatment

There are a few potential options in dealing with outliers data, each having specific pros and cons.

1. Removing outliers - If the outliers represent noise/error and form a relatively small portion of the actual data, we can consider removing/trimming them. Another valid reason to remove outliers could be when we know that it doesn't belong to the target population under study. In the current data-set, however, the percentage of outliers is rather large (10.39% for hospitalization charges for the combined percentage including viral load could be higher).

2. Replacing outliers(Winsorization) - The other option could be to replace the outliers with the suitable percentile value of the data.

3. Apply log transformation - As observed in the previous section, the log transformation seems quite effective in removing/reducing the number of outliers significantly (especially for hospitalization charges). If the further intended statistical analysis can work effectively with log-transformed data, we can consider choosing this option. However, there are certain statistical analysis techniques which may not quite yield the same result on log-transformed data as on the original data. For instance, the t-test on the log-transformed data compares geometric means, not the (usual) arithmetic means.

In this case-study, we choose not to treat outliers for the statistical tests for the following reasons.

1. We can remove outliers if they represent either noise/erroneous data or if they do not represent the target population. However, in the absence of any additional details, it is difficult to know if the outliers are legitimate data values or if they represent one of the conditions noted earlier.
2. The outliers form a considerable portion of data-set (>10%), so removing them may produce statistical results which are not reflective of the actual population.
3. While winsorization can be one potential alternative, it can distort the sample distribution which may impact parametric hypothesis tests.
4. We observed that log transformation can effectively treat outliers. However, as noted above, t-test on the log-transformed data is not quite the same as t-test on the actual data. While t-test on the actual data compares arithmetic means, t-test on log-transformed data effectively compares their geometric means. While this is a fairly popular approach, for this case-study, we will refrain from using it.
5. Other two good solutions are **A.** use of non-parametric tests which are usually robust against outliers and the violation of normality, or **B**. bootstrapping approach to compute sampling distribution parameters which is independent of any assumptions about the underlying distribution. In this case-study, we rely on the use of non-parametric tests wherever possible.

NOTE: For the sake of completeness, we will create a copy of the original data-set and replace outliers with appropriate percentile values (option 2). This data-set can be used as necessary for further analysis. In the code below, we evaluate various percentile values for winsorization and select the lowest value which addresses all the outliers.

[145]:

```
import random
from scipy.stats.mstats import winsorize
```

```
outliers = []
transformations = [
    ('    original', lambda s:s),
    ('   90% winsorize', lambda s:winsorize(s,(0.05, 0.05))),
    ('  80% winsorize', lambda s:winsorize(s,(0.1, 0.1))),
    (' 78% winsorize', lambda s:winsorize(s,(0, 0.11))),
]

total_n = data.shape[0]
for col in ['hospitalization charges', 'viral load']:
    for trans_name, trans_fn in transformations:
        ret = findoutliers(trans_fn(df[col]))
        outliers_n = ret.sum()
        outliers.append([col, trans_name, outliers_n, np.round((outliers_n /␣
 ↪total_n) * 100, 2)])

print(f'total number of rows in dataset: {total_n}')
outliers_df = pd.DataFrame(data=outliers, columns=['column', 'winsorization',␣
 ↪'outliers count', 'outliers as % of total rows'])

outliers_df = outliers_df.set_index(keys=['column', 'winsorization'])
outliers_df.unstack()
```

```
total number of rows in dataset: 1338
```

[145]:

```
                        outliers count                                \
winsorization                 original   90% winsorize   80% winsorize
column
hospitalization charges            139             139             139
viral load                           9               0               0

                                       outliers as % of total rows  \
winsorization            78% winsorize                    original
column
hospitalization charges              0                       10.39
viral load                           0                        0.67


winsorization              90% winsorize   80% winsorize  78% winsorize
column
hospitalization charges            10.39           10.39            0.0
viral load                          0.00            0.00            0.0
```

[146]:

```
# At around 78% winsorization, we see zero outliers for both hospitalization␣
 ↪charges and viral load variables. So we will use this value.
df_outlier_treated = df.copy()
```

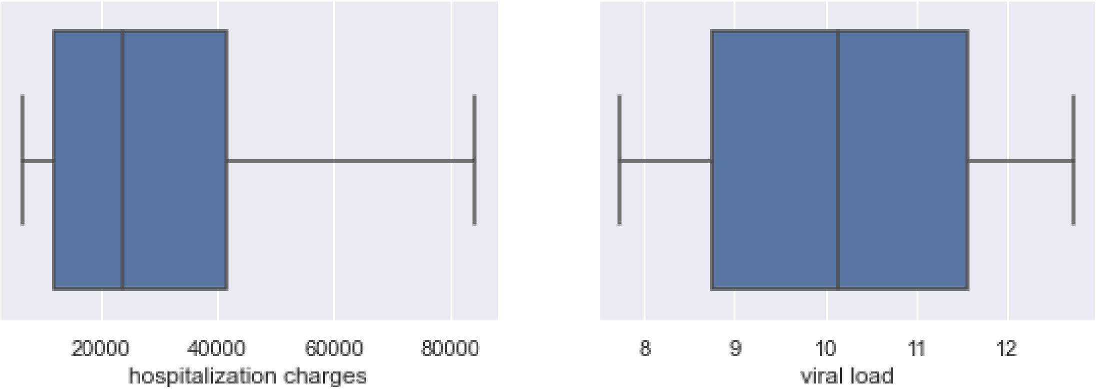
```
#dataset with outliers winsorized
for col in ['viral load', 'hospitalization charges']:
    df_outlier_treated[col] = winsorize(df_outlier_treated[col],(0.11,0.11))
```

[147]:

```
#visualize datasets with outliers treated (winsorized or trimmed)

fig, ax = plt.subplots(1,2, figsize=(10,3))

sns.boxplot(x='hospitalization charges', data=df_outlier_treated, ax=ax[0])
#sns.boxplot(x='hospitalization charges', data=df_outlier_treated2, ax=ax[0][1])

sns.boxplot(x='viral load', data=df_outlier_treated, ax=ax[1])
```

[147]:

```
<AxesSubplot:xlabel='viral load'>
```



2.8 Statistical tests

In this section, we attempt to answer the following questions using statistical tests.

1. Is hospitalization charge for smoker patients *greater than* non-smoker patients?
2. Is viral load of female patients *different* from that of male patients?
3. A. Is the proportion of smoking significantly different across different regions?

   B. Is the severity level significantly different across different regions?
4. Is the mean viral load of women with 0 Severity level , 1 Severity level, and 2 Severity level the same?

For each scenario, we test relevant assumptions before choosing an appropriate hypothesis test. We define a few utility functions below.

[148]:

```
### Utility functions

import scipy.stats as stats
```

```
from statsmodels.graphics.gofplots import qqplot

#helper function to perform normality test
#for each dataset, it plots its histogram, boxplot, and QQ plot
#it also prints Shapiro-Wilk metrics
#in addition, additional transformation functions (such as log, sqrt etc) can␣
 ↪be supplied in t_arr
def testnorm(data, title, t_arr = []):
    arr = [('', lambda x: x)] if (t_arr == None or len(t_arr) == 0) else t_arr
    cnt = len(arr)

    fig = plt.figure(figsize=(15, cnt*3.5))
    subfig = fig.subfigures(nrows=cnt, ncols=1)
    res = [] #to hold shapiro-wilk results

    for i in range(cnt):
        item = arr[i]
        text = title + ' ' + item[0]
        fn = item[1]
        tr_data = pd.Series([fn(ele) for ele in data])

        figref = subfig[i] if (cnt > 1) else subfig

        figref.suptitle(text)
        ax = figref.subplots(nrows=1, ncols=3)
        sns.histplot(tr_data, kde=True, ax=ax[0])
        sns.boxplot(x=tr_data, ax=ax[1])
        qqplot(tr_data, line='s', ax = ax[2])

        res.append(stats.shapiro(tr_data))

    plt.show()

    print('\nShapiro-Wilk Test metrics')
    for i in range(cnt):
        print(f'{title} {arr[i][0]} : {res[i]}')
```

2.8.1 Test 1 - Is hospitalization charge for smoker patients *greater than* non-smoker patients?

In this scenario, we have two groups; smokers and non-smokers. In order to determine the appropriate hypothesis tests, We first check the following assumptions.

Assumptions

1. Observations in each sample independent and identically distributed (iid) - We assume that this assumption holds.
2. Both samples (smoking and non-smoking patients) have homogeneous variance - This is a

requisite assumption for parametric two sample t-test. We check this below.

3. Observations within both the samples are normally distributed - This is a requisite assumption for parametric two sample t-test. We check this below.

[149]:
```
#create two samples for smoker and non-smoker groups
data = df
sample_s = data[data['smoker'] == 'yes']['hospitalization charges']
sample_ns = data[data['smoker'] == 'no']['hospitalization charges']
```

Assumption check for variance homogeneity

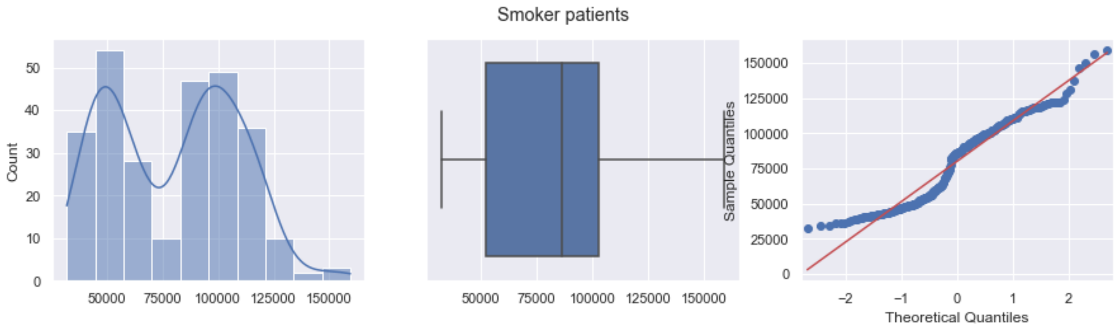
[150]:
```
var_s = np.var(sample_s, ddof=1)
var_ns = np.var(sample_ns, ddof=1)

print(f'variance of charges for smoker patients sample: {var_s}, variance of␣
 ↪charges for nonsmoking patients sample: {var_ns}')

#levene's test
print("\nLevene's test to check if population variances are equal (alpha=0.05)")
print('H0: population variances are equal')
print('H1: population variances are not equal')
print(f"Levene's test metric: {stats.levene(sample_s, sample_ns)}")
```

```
variance of charges for smoker patients sample: 832547033.7254218, variance of
charges for nonsmoking patients sample: 224533906.64452907

Levene's test to check if population variances are equal (alpha=0.05)
H0: population variances are equal
H1: population variances are not equal
Levene's test metric: LeveneResult(statistic=332.6132009308764,
pvalue=1.5595259401311176e-66)
```

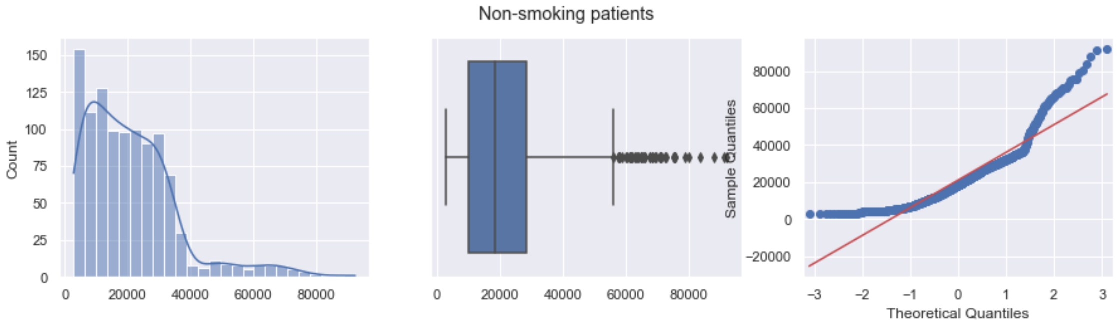
Conclusion - P-value from Levene's test is very low (<0.05), and hence we reject the null hypothesis of variance equality. So **the assumption of variance homogeniety does not hold.**

Assumption check for normality

[151]:
```
testnorm(sample_s, 'Smoker patients')
testnorm(sample_ns, 'Non-smoking patients')
```



```
Shapiro-Wilk Test metrics
Smoker patients  : ShapiroResult(statistic=0.939551830291748,
pvalue=3.6248792856241607e-09)
```



```
Shapiro-Wilk Test metrics
Non-smoking patients  : ShapiroResult(statistic=0.8728622794151306,
pvalue=1.4454367844287343e-28)
```

Conclusion - Clearly, **'hospitalization charges' for 'smoking' and 'non-smoking' patient samples are not normally distributed. So the normality assumption doesn't hold**.(confirmed by qq-plot and shapiro-welk test metrics)

Choosing appropriate hypothesis test. Since the assumptions of normality and variance homogeneity do not hold, two sample t-test ideally should not be used in this case. For this case-study, however, we will continue to use two sample t test. We will also Mann Whitney U Test which is a non parametric test suitable to compare distribution of two samples when the assumption of normality does not hold.

A. Two-sample independent t-test (hospitalization charges ~ smoker) H0: The means of the two samples are equal. That is, mean hospitalization charge for smoker patients = mean hospitalization charge for non-smoker patients.

H1: The mean hospitalization charge for smoking patients > mean hospitalization charge of non-smoking patients

confidence level - 95%, significance level alpha = 5% (0.05), right tail test

```
[152]: stats.ttest_ind(sample_s, sample_ns, equal_var=False, alternative='greater')
```

```
[152]: Ttest_indResult(statistic=32.751856578287196, pvalue=2.9454726753868796e-103)
```

Conclusion - The reported p-value is extremely small (<0.05), hence, we reject the null hypothesis of equal means. **Thus, at 95% confidence level, we can conclude that mean hospitalization charge for smoking patients is greater than that of non-smoking patients.**

B. Mann whitney test **Additional assumption check:** Observations in each sample (smoker and non-smoker) can be ranked. We can clearly rank each observation based on the value of hospitalization charge, and hence, this assumption holds.

Hypothesis test setup

H0: The distributions of hospitalization charges for smoker and non-smoker patients are equal.

H1: the distributions are not equal.

confidence level - 95%, significance level alpha = 5% (0.05), right tail test

```
[153]: stats.mannwhitneyu(sample_s, sample_ns, alternative='greater')
```

```
[153]: MannwhitneyuResult(statistic=284132.5, pvalue=2.6407031043303346e-130)
```

conclusion - The reported p-value is extremely small (<0.05), hence, we reject the null hypothesis of equal means. **Thus, at 95% confidence level, we can conclude that mean hospitalization charge for smoking patients is greater than that of non-smoking patients.**

Insights

As proven by both parametric and non-parametric hypothesis tests, smoking is a significant factor in determining hospitalization charges. Patients who smoke are likely to have greater hospitalization charges than those patients who do not smoke.

2.8.2 Test 2 - Is viral load of female patients *different* from that of male patients?

In this scenario, we have two groups; male patients and female patients. In order to determine the appropriate hypothesis tests, We first check the following assumptions.

Assumptions

1. Observations in each sample independent and identically distributed (iid) - We assume that this assumption holds.
2. Both samples (smoking and non-smoking patients) have homogeneous variance - This is a requisite assumption for parametric two sample t-test. We check this below.
3. Observations within both the samples are normally distributed - This is a requisite assumption for parametric two sample t-test. We check this below.

[154]:

```
#create two samples for male and female patients
data = df
sample_m = data[data['sex'] == 'male']['viral load']
sample_f = data[data['sex'] == 'female']['viral load']
```

Assumption check for variance homogeneity

[155]:

```
var_m = np.var(sample_m, ddof=1)
var_f = np.var(sample_f, ddof=1)

print(f'variance of viral load for male patients sample: {var_m}, variance of␣
 ↪charges for female patients sample: {var_f}')

#levene's test
print("\nLevene's test to check if population variances are equal (alpha=0.05)")
print('H0: population variances are equal')
print('H1: population variances are not equal')
print(f"Levene's test metric: {stats.levene(sample_m, sample_f)}")
```

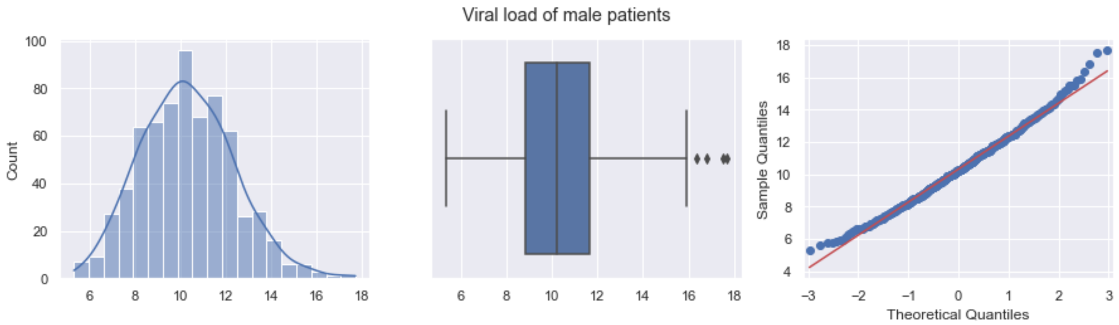
```
variance of viral load for male patients sample: 4.189755370370371, variance of
charges for female patients sample: 4.061844158123506

Levene's test to check if population variances are equal (alpha=0.05)
H0: population variances are equal
H1: population variances are not equal
Levene's test metric: LeveneResult(statistic=0.0038754151966871046,
pvalue=0.9503708012456551)
```

Conclusion - Levene's test reports a very high P-val (>0.05) and hence we fail to reject the null hypothesis of variance equality. **Thus, we conclude at 95% confidence level that variances of viral load of male and female patients are equal. In other words, the assumption of variance homogeneity holds.**

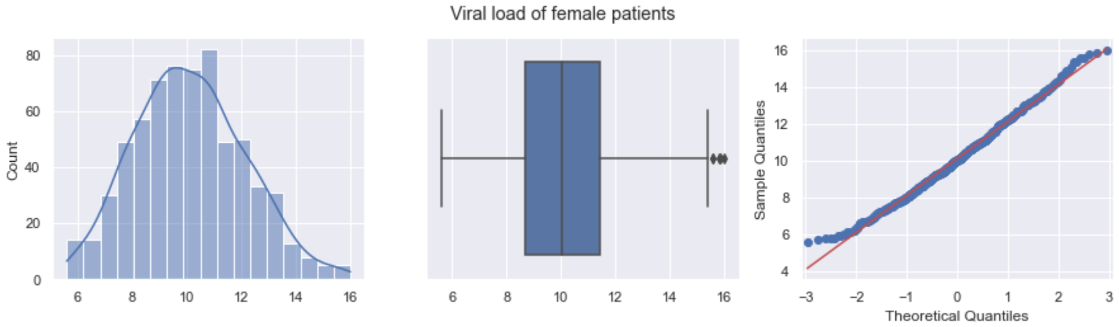
Assumption of normality

[156]:

```
testnorm(sample_m, 'Viral load of male patients')
testnorm(sample_f, 'Viral load of female patients')
```



```
Shapiro-Wilk Test metrics
Viral load of male patients  : ShapiroResult(statistic=0.9930650591850281,
pvalue=0.003189612179994583)
```



```
Shapiro-Wilk Test metrics
Viral load of female patients  : ShapiroResult(statistic=0.9930474162101746,
pvalue=0.003624602919444442)
```

Conclusion: - While the Shapiro-Wilk test result p-values are slightly less than 0.05, visually, in QQ plot we can observe that viral load distribution is fairly normal. Thus we can consider normality assumption to hold true.

Choosing appropriate test Since the assumption of variance homogeneity holds and the distribution of viral load for both samples appear fairly symmetric and normal, we can use two independent sample t-test for this scenario.

Two-sample independent t-test (viral load ~ sex) **H0:** The means of the two samples are equal. That is, mean viral load value male patients = mean viral load value for female patients.

H1: The mean viral load for male patients != mean viral load value for female patients

confidence level - 95%, significance level alpha = 5% (0.05), two tailed test

```
[157]: stats.ttest_ind(sample_m, sample_f, equal_var=True)
```

```
[157]: Ttest_indResult(statistic=1.695711164450323, pvalue=0.0901735841670204)
```

Conclusion - The reported p-value (~0.09) is > significance level 0.05, and hence, we fail to reject the null hypothesis of equal means. **Thus, at 95% confidence level, we can conclude that there is no significance difference in the viral load values of male and female patients.**

2.8.3 Test 3.A - Is the proportion of smoking significantly different across different regions?

In this scenario, we need to compare two categorical variables-smoker and region- and determine if they are independent. We can use chi-square test of independence if the following assumptions hold.

Assumptions

1. Observations used in the calculation of the contingency table are independent. - We assume the observations are indepedent.
2. The value of each cell of the contingency table should be at-least 5. - We check this below.

```
[158]: #calculate observed freq of patients
       obs_freq = pd.crosstab(index=df['smoker'], columns=df['region'])
       obs_freq
```

```
[158]: region  northeast  northwest  southeast  southwest
       smoker
       no            257        267        273        267
       yes            67         58         91         58
```

Observation - Each cell of the contigency table has value > 5. Thus we can use chi-square test of independence.

Chi-square test of independence **H0:** smoker and region are indepedent variables.

H1: smoker and region are not independent variables.

confidence level - 95%, significance level alpha = 5% (0.05)

```
[159]: stat, pval, dof, expected = stats.chi2_contingency(obs_freq)
       print(pval)
```

```
0.06171954839170547
```

Conclusion: - As the reported chi-square test p-value (0.0617) > 0.05, we fail to reject the null hypothesis of variable independence. **Thus, at 95% confidence level, we can conclude that smoker and region are independent variables. In other words, region does't determine number/proportion of smokers.**

2.8.4 Test 3.B - Is severity level of patients significantly different across different regions?

In this scenario, we need to compare two categorical variables-severity level and region- and determine if they are independent. We can use chi-square test of independence if the following assumptions hold.

Assumptions

1. Observations used in the calculation of the contingency table are independent. - We assume the observations are indepedent.

2. The value of each cell of the contingency table should be at-least 5. - We check this below.

[160]:
```
#calculate observed freq of patients
obs_freq = pd.crosstab(index=df['severity level'], columns=df['region'])
obs_freq
```

[160]:
```
region          northeast  northwest  southeast  southwest
severity level
0                     147        132        157        138
1                      77         74         95         78
2                      51         66         66         57
3                      39         46         35         37
4                       7          6          5          7
5                       3          1          6          8
```

Note: We observe that for severity level 5, northeast and northwest region values are <5. For simplicity, we remove severity level 5 altogether from our analysis.

[161]:
```
obs_freq = obs_freq[0:5]
obs_freq
```

[161]:
```
region          northeast  northwest  southeast  southwest
severity level
0                     147        132        157        138
1                      77         74         95         78
2                      51         66         66         57
3                      39         46         35         37
4                       7          6          5          7
```

Chi-square test of independence **H0:** severity level and region are indepedent variables.

H1: severity level and region are not independent variables.

confidence level - 95%, significance level alpha = 5% (0.05)

[162]:
```
stat, pval, dof, expected = stats.chi2_contingency(obs_freq)
print(pval)
```

```
0.8320194219201974
```

Observation - The reported p-value for chi-square test is 0.83 (>0.05), and therefore we fail to reject the null hypothesis of variable independence. **Thus at 95% confidence level, we can conclude that severity level and region are independent variables.**

2.8.5 Test 4 - Is the mean viral load of women with 0 Severity level , 1 Severity level, and 2 Severity level the same?

In this scenario, 'severity level' is a categorical variable with 3 levels (0, 1, and 2). We need to compare viral load means for female patients with each severity level, thus viral load is the response variable. We can use one-way ANOVA test for this purpose (one factor with three levels) if the following assumptions hold true.

Assumptions

1. Observations in each sample independent and identically distributed (iid) - We assume that this assumption holds.
2. All samples (corresponding to severity level 0, 1, and 2) have homogeneous variance - This is a requisite assumption for parametric one-way ANOVA. We check this below.
3. Observations within all three samples are normally distributed - This is a requisite assumption for parametric one-way ANOVA. We check this below.
4. Response variable residuals are normally distributed (or approximately normally distributed) - We will check this post ANOVA test.

[163]:

```
#create three samples for female patients corresponding to severity level 0, 1, and 2

#prepare dataframe for female patients with sev level 0,1 or 2
data = df[(df['sex'] == 'female') & ((df['severity level'] == 0) | (df['severity level'] == 1) | (df['severity level'] == 2))]

sample_s0 = data[data['severity level'] == 0]['viral load']
sample_s1 = data[data['severity level'] == 1]['viral load']
sample_s2 = data[data['severity level'] == 2]['viral load']
```

Assumption check for variance homogeneity

[164]:

```
var_s0 = np.var(sample_s0, ddof=1)
var_s1 = np.var(sample_s1, ddof=1)
var_s2 = np.var(sample_s2, ddof=1)

print(f'variance of viral load for female patients with severity level 0: {var_s0}')
print(f'variance of viral load for female patients with severity level 1: {var_s1}')
print(f'variance of viral load for female patients with severity level 2: {var_s2}')

#levene's test
print("\nLevene's test to check if population variances are equal (alpha=0.05)")
print('H0: population variances are equal')
print('H1: population variances are not equal')
print(f"Levene's test metric: {stats.levene(sample_s0, sample_s1, sample_s2)}")
```

```
variance of viral load for female patients with severity level 0:
3.9564046784890428
variance of viral load for female patients with severity level 1:
```

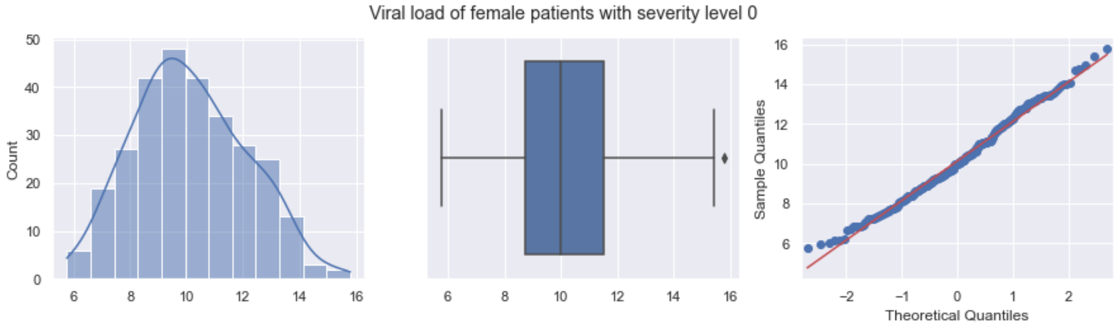
```
3.7212929130049175
variance of viral load for female patients with severity level 2:
4.8827151402934055

Levene's test to check if population variances are equal (alpha=0.05)
H0: population variances are equal
H1: population variances are not equal
Levene's test metric: LeveneResult(statistic=0.9435131022565071,
pvalue=0.38987253596513605)
```

Conclusion: - The P-value reported by Levene's test is 0.39 (> 0.05), and hence, we fail to reject the null hypothesis of equal variances. **Thus, with 95% confidence level, we can conclude that the assumption of equal variances holds**

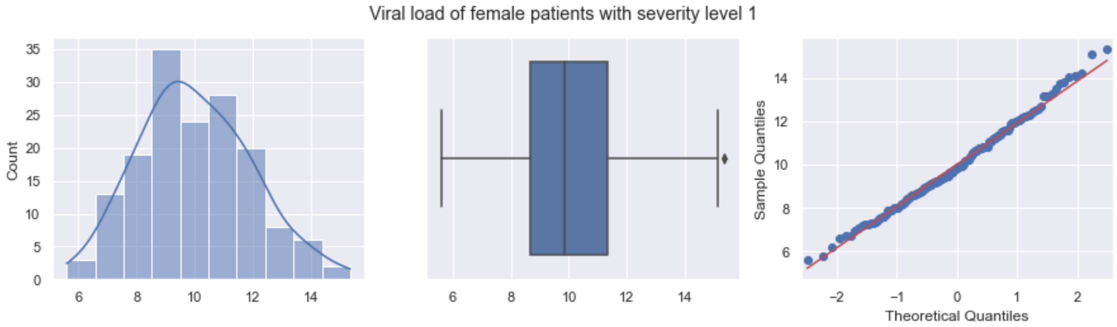
Assumption of normality

[165]:

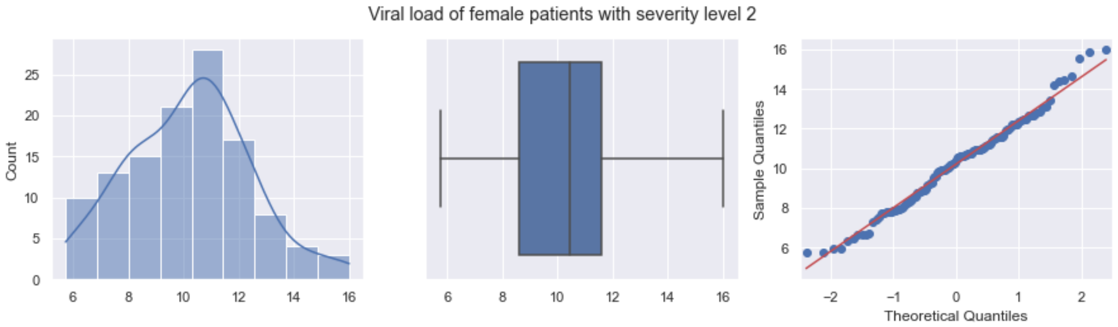
```
testnorm(sample_s0, 'Viral load of female patients with severity level 0')
testnorm(sample_s1, 'Viral load of female patients with severity level 1')
testnorm(sample_s2, 'Viral load of female patients with severity level 2')
```



```
Shapiro-Wilk Test metrics
Viral load of female patients with severity level 0  :
ShapiroResult(statistic=0.9896610379219055, pvalue=0.038132064044475555)
```



```
Shapiro-Wilk Test metrics
Viral load of female patients with severity level 1  :
ShapiroResult(statistic=0.9921300411224365, pvalue=0.539344072341919)
```



```
Shapiro-Wilk Test metrics
Viral load of female patients with severity level 2  :
ShapiroResult(statistic=0.9860238432884216, pvalue=0.25857919454574585)
```

Conclusion: - The reported pvalues from Shapiro-Wilk test for severity level 1 sample and severity level 2 samples are > 0.05, and therefore we fail to reject the null hypothesis of normality. However, the pvalue for severity level 0 sample is 0.038. While this value is < 0.05, it's quite close to the alpha threshold. Also, the QQ plot visually confirms that the observations in severity level 0 sample are almost normally distributed. Considering this, we can consider observations in each sample to be roughly normally distributed. **Thus the assumption of normality holds.**

Choosing appropriate test Since all the necessary assumptions hold true, we will use One-way ANOVA to determine if the mean viral load value in female patients corresponding to severity level 0, 1, and 2 are statistically different.

One-way ANOVA (viral load ~ severity level) **H0:** All population means are equal. That is, mean viral load value in female patients is similar across three severity levels.

H1: Not all population means are equal. In other words, mean viral load value is different for at-least one severity level.

confidence level - 95%, significance level alpha = 5% (0.05)

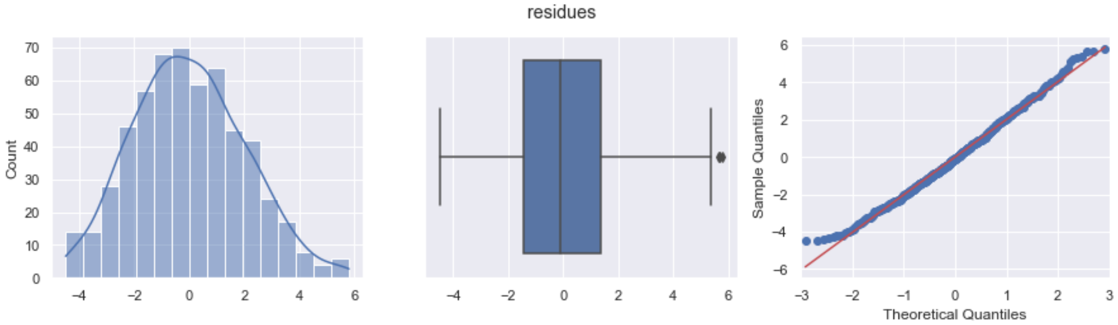
```
[166]: #compute ANOVA using statsmodel
import statsmodels.api as sm
from statsmodels.formula.api import ols

df_anova = data.copy()
df_anova = df_anova.rename(columns={'viral load': 'viral_load', 'severity level': 'sev_level'})
```

```
model = ols("viral_load ~ C(sev_level)", data=df_anova).fit()
aov_table = sm.stats.anova_lm(model, typ=2)
print(aov_table)
```

```
                   sum_sq     df         F    PR(>F)
C(sev_level)     6.852692    5.0  0.335506  0.715119
Residual      2299.847921  563.0       NaN       NaN
```

Residue analysis

```
[167]: testnorm(model.resid, 'residues')
```

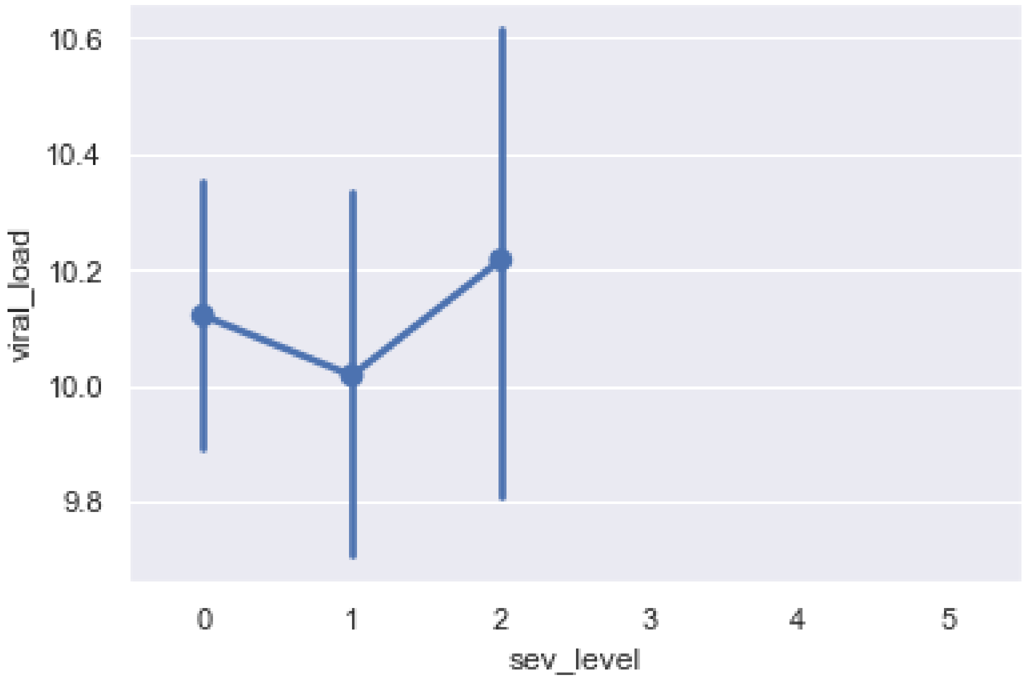


```
Shapiro-Wilk Test metrics
residues  : ShapiroResult(statistic=0.9933659434318542,
pvalue=0.013526226393878 46)
```

Conclusion: While the Shapiro-wilk test pvalue for residues is < 0.05, visually through QQ plot we can confirm that residues are fairly normally distributed. Thus the results of this ANOVA test can be considered valid.

mean effect plot for viral load ~ severity level

```
[168]: sns.pointplot(y="viral_load", x='sev_level', ci=95, data=df_anova)
       plt.show()
```



Conclusion:

Based on the output of the one way ANOVA test, we see that p-value(0.71) > 0.05, and therefore we fail to reject the null hypothesis of mean equality. This means that there indicates that the differences in the viral load value in female patients across three severity levels are statistically not significant. We can also visually see this in the bi-variate graph between viral load and severity level. **In other words, severity level (0,1, or 2) is not a signigicant factor in predicting viral load value in female patients (at significance level 0.05)**

2.9 Additional statistical tests

In this section, we carry out a few more statistical tests to determine factors which are significant in determinining hospitalization charges.

1. Is Gender a significant factor in determining hospitalization charges?
2. Is Region a significant factor in determining hospitalization charges?
3. Is Age a significant factor in determining hospitalization charges?
4. Is Severity level a significant factor in determining hospitalization charges?
5. Is smoking a significant factor in determining hospitalization charges? (we already covered this in the previous section and found smoking to be a significant factor)

NOTE: Since hospitalization charges is a positively skewed variable, it is quite likely that the observations (that is 'hospitalization charges') in in the sub-samples formed in each of the scenarios above will also be positively skewed, and therefore not normally distributed. **Therefore, in this section, we will prefer using non-parametric tests over parametric tests.**

Additional Test 1: Is Gender a significant factor in determining hospitalization charges? In this scenario, we have two samples-female patients and male patients. We will use **Mann whitney test** to determine if the distributions of the two samples are similar.

Hypothesis test setup

H0: The distributions of hospitalization charges for male and female patients are equal.

H1: the distributions are not equal.

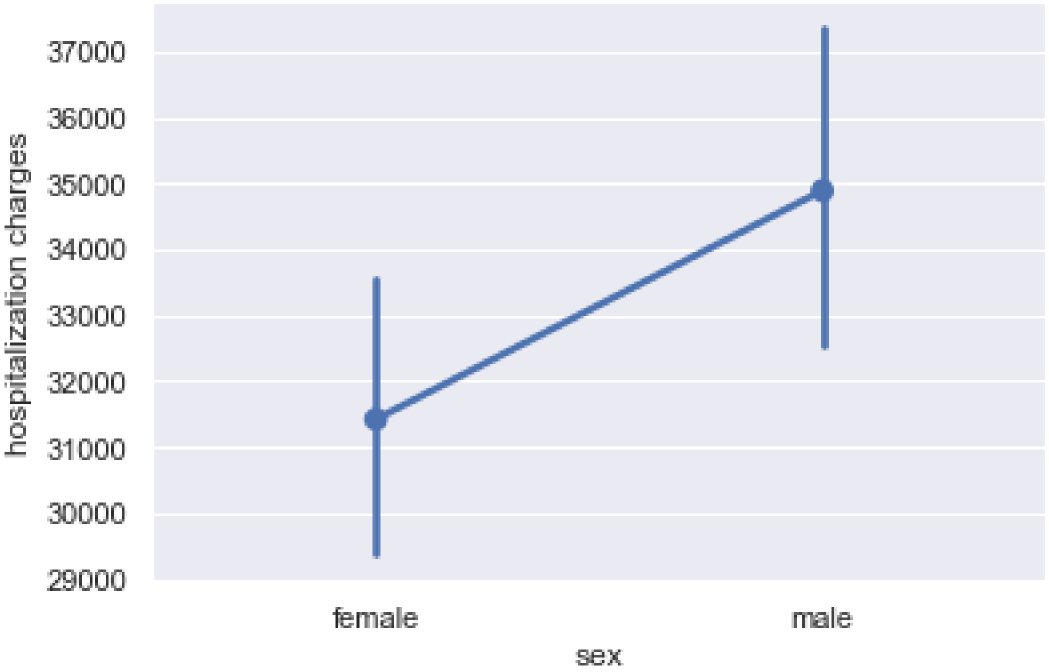
confidence level - 95%, significance level alpha = 5% (0.05)

[169]:
```
data = df
stats.mannwhitneyu(sample_m, sample_f)
```

[169]:
```
MannwhitneyuResult(statistic=235319.0, pvalue=0.10178463776495861)
```

mean effect plot for hospitalization charge ~ Gender

[170]:
```
sns.pointplot(y="hospitalization charges", x='sex', ci=95, data=data)
plt.show()
```



Conclusion: As pvalue(0.1) > 0.05, we fail to reject the null hypothesis of samples having equal distributions. The mean-effect plot for gender and hospitalization charges also confirms this. **Thus, at 95% confidence level, we can conclude that gender is not a significant factor in determining hospitalization charges.**

Additional Test 2: Is Region a significant factor in determining hospitalization charges? In this scenario, we have four samples correpsonding to each of the regions-northeast, northwest, southeast, southwest. We will use **Kruskal-Wallis H Test** to determine if the distributions of the four samples are similar.

Hypothesis test setup

H0: The distributions of hospitalization charges for samples corresponding to four regions are equal.

H1: At-least one sample's distribution is not equal.

confidence level - 95%, significance level alpha = 5% (0.05)

[171]:

```
sample_se = data[data['region'] == 'southeast']['hospitalization charges']
sample_sw = data[data['region'] == 'southwest']['hospitalization charges']
sample_ne = data[data['region'] == 'northeast']['hospitalization charges']
sample_nw = data[data['region'] == 'northwest']['hospitalization charges']

stats.kruskal(sample_se, sample_sw, sample_ne, sample_nw)
```

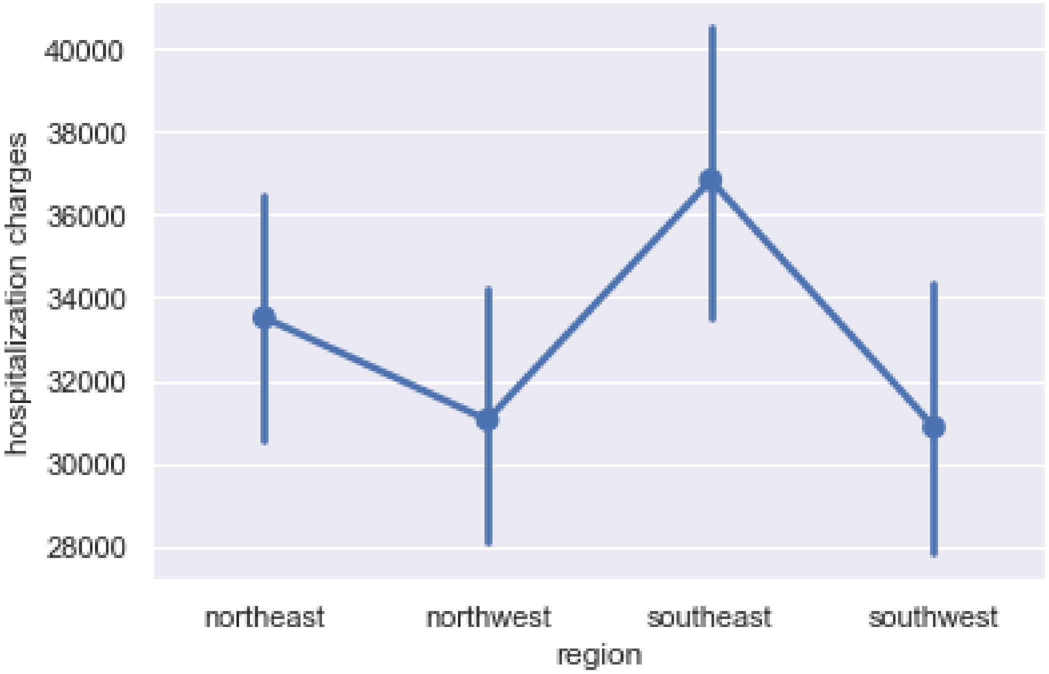
[171]:

```
KruskalResult(statistic=4.735721548178828, pvalue=0.19220376640477613)
```

mean effect plot for hospitalization charge ~ region

[172]:

```
sns.pointplot(y="hospitalization charges", x='region', ci=95, data=data)
plt.show()
```



Conclusion: As pvalue(0.19) > 0.05, we fail to reject the null hypothesis of samples having equal distributions. The mean-effect plot for region and hospitalization charges also confirms this. **Thus, at 95% confidence level, we can conclude that region is not a significant factor in determining hospitalization charges.**

Additional Test 3: Is Age a significant factor in determining hospitalization charges? Since Age is a continuous feature, we use binning to create a categorical variable. We will use 'age_binned' created earlier. The size of each bin is 3 years. We will use **Kruskal-Wallis H Test** to determine if the distributions hospitalization charge across all age groups is similar.

Hypothesis test setup

H0: The distributions of hospitalization charges across different age groups are equal.

H1: At-least one sample's distribution is not equal.

confidence level - 95%, significance level alpha = 5% (0.05)

[173]:

```
gb = data.groupby('age_bins')
samples_by_age = [gb.get_group(grp)['hospitalization charges'] for grp in gb.
 ↪groups]

stats.kruskal(*samples_by_age)
```

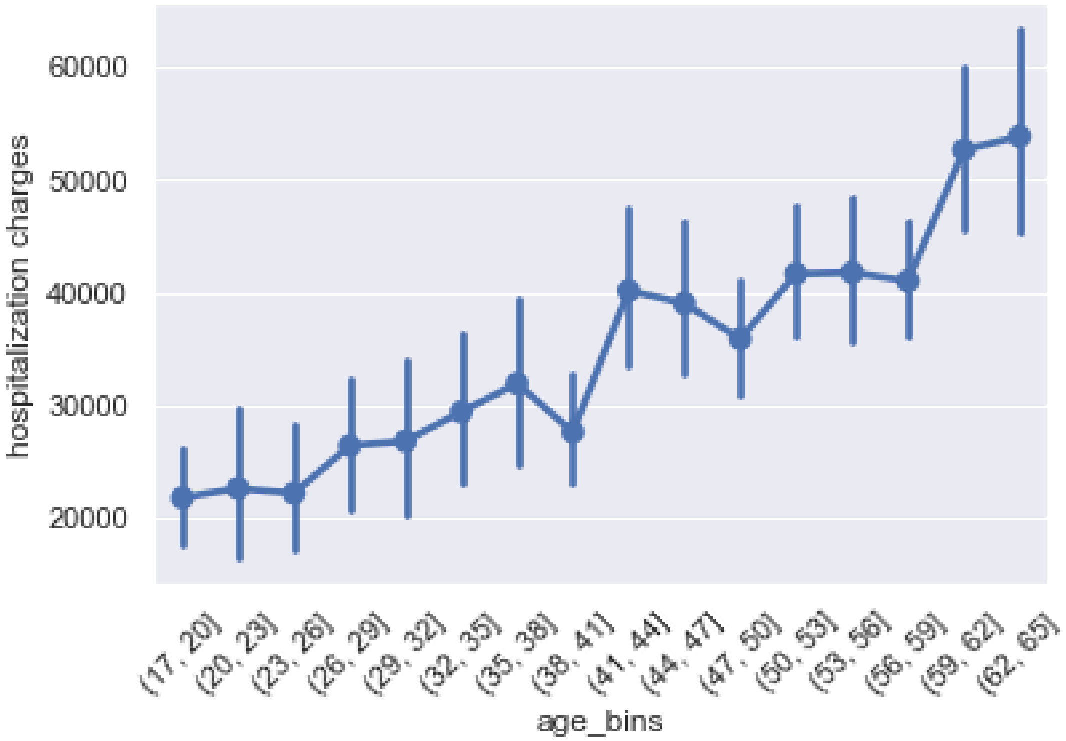
[173]:

```
KruskalResult(statistic=386.70010821896113, pvalue=4.2955091728791697e-73)
```

mean effect plot for hospitalization charge ~ age_bins

[174]:

```
sns.pointplot(y="hospitalization charges", x='age_bins', ci=95, data=data)
plt.xticks(rotation=45)
plt.show()
```



Conclusion: The p-value of kruskal-Wallis test is very small (<0.05), and hence we reject the null hypothesis of equal distributions. **Thus, with 95% confidence level we can conclude that age is a significant factor in determining hospitalization charges.** In the mean-effect plot we can clearly see that in general, mean value of hospitalization charge increase with the increase in age group values ((23,26], (38,41], (44-50] etc being a few exceptions).

Additional Test 4: Is Severiy level a significant factor in determining hospitalization charges? We will use **Kruskal-Wallis H Test** to determine if the distributions hospitalization charge across all severity levels is similar.

Hypothesis test setup

H0: The distributions of hospitalization charges across different severity levels are equal.

H1: At-least one sample's distribution is not equal.

confidence level - 95%, significance level alpha = 5% (0.05)

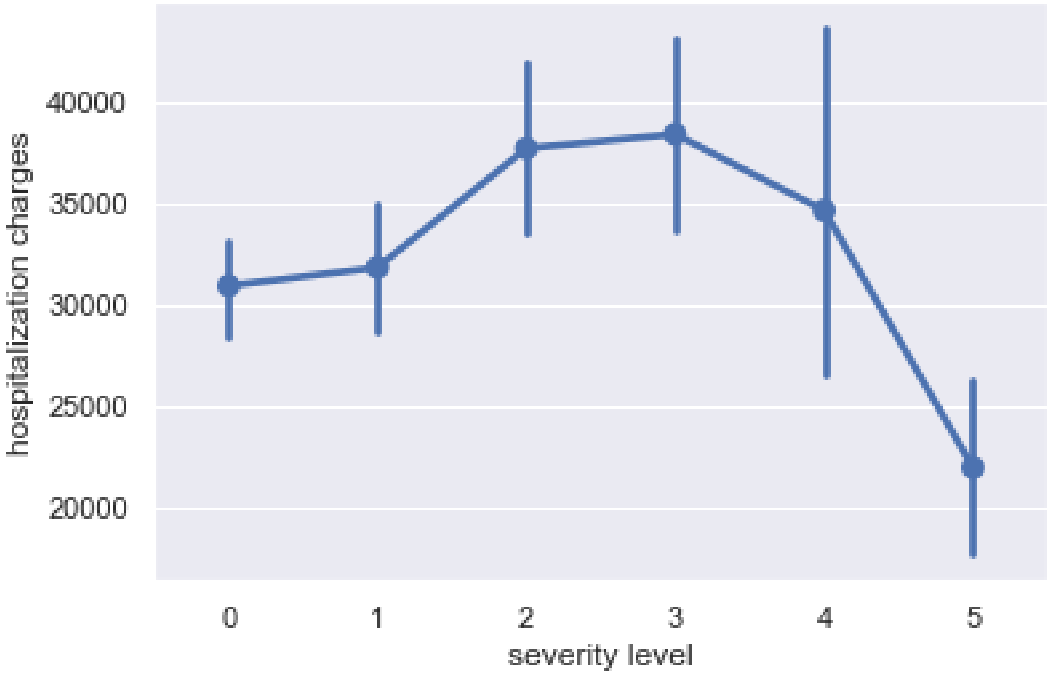
[175]:
```
gb = data.groupby('severity level')
samples_by_sev = [gb.get_group(grp)['hospitalization charges'] for grp in gb.
 ↪groups]

stats.kruskal(*samples_by_sev)
```

[175]:
```
KruskalResult(statistic=29.487198158531086, pvalue=1.860373239489511e-05)
```

mean effect plot for hospitalization charge ~ severity level

[176]:

```
sns.pointplot(y="hospitalization charges", x='severity level',ci=95, data=data)
plt.show()
```



Conclusion: As p-value of Kruskal-Wallis test is very low(< 0.05), we reject the null hypothesis of equal distributions. **Thus at 95% confidence level, we can conclude that severity level is a significant factor in determining hospitalizaton charges.** Also, we can observe in the mean-effect plot that the mean hospitalization charge increase from severity level 0 to severity level 3. Thereafter it drops for severity level 4 and sees further considerable drop for severity level 5. The highest mean hospitalization charge is for severity level 3 and lowest is for severity level 5.

2.10 Business insights and Recommendations

In this section, we summarize high level business insights and potential recommendations(if any). For more detailed observations, please refer the individual sections on each tests/analysis.

Business Insights

1. Distribution of patients

    - There are almost equal number of male and female patients.
    - The number of patients across four regions - ‘southeast’. ‘southwest’, ‘northeast’, and ‘northwest’ are almost equal.
    - Around 20% of the patients are ‘smokers’.

    - Age group 18-20 years has the highest number of patients (around 10% of the total patients). In the remaining age groups (ranging from 21-23 to 63-65), the number of

patients are more or less equally distributed (each group having around 6% of patients). The median and mean patient age is 39 years.
  - The most common severity level is *level-0* (at 43%), followed by *level-1* (24%), *level-2*(18%), and level-3(11.7%). The number of patients with *level-4* and *level-5* are very low (< 2% each).

2. Factors affecting hospitalization charges

   - Patient's smoking habit is a significant factor in predicting hospitalization charges. The mean hospitalization charge for smoking patients is greater than that of non-smoking patients.

   - Patients' age is a significant factor in determining hospitalization charges. Generally, the mean value of hospitalization charge increase with the increase in age group values ((23,26], (38,41], (44-50] etc being a few exceptions). We also observe a weak positive correlation between age and hospitalization charges. - Severity level is a significant factor in determining hospitalizaton charges. The mean hospitalization charge increase from severity level 0 to severity level 3. Thereafter it drops for severity level 4 and sees further considerable drop for severity level 5. The highest mean hospitalization charge is for severity level 3 and lowest is for severity level 5.

   - Patient's gender and region are not significant factors in predicting hospitalization charges.

3. Other insights

   - Gender is not a significant factor in determining viral load value.

   - Viral load value of female patients across severy levels 0, 1, and 2 are similar.

   - The proportion smoking patients doesn't vary across regions.

   - The proportion of patients with specific severity level does not vary across region. Thus region and severity levels are independent.
   - The young (17-33) and elder (49-65) patient groups have considerably higher number of severity level 0 patients than the patients with severity level 1, 2, or 3. The middle age group (34-48), however, has lower number of severity level 0 patients than patients with severity level 1 and 2.

Recommendations:

1. Around 20% of total patients are smokers, and smoking patients tend to incur higher hospitalization charges, which indicates that they, on average, undergo more complex, expensive treatment. If possible, the hospital can form strategy to devise more specialized but affordable treatment package for smoking patients.

2. The highest number of patients (around 10%) fall in 18-20 years group, and half the patient population is less than or equal to 39 years of age. Assuming that younger patients, on average, may need less advance treatment, the hospital management may consider setting up separate wards/centers for patients who may not need advance treatment and thus freeing up resources for other critical patients.

3. Generally speaking, treatment cost goes up with patient's age. Similarly, severity level determines treatment cost. The mean hospitalization charge increase from severity level 0 to severity level 3. Thereafter it drops for severity level 4 and sees further considerable drop for severity level 5. The highest mean hospitalization charge is for severity level 3 and lowest is for severity level 5. 67% of the patients have severity level 0 or 1. 29% patients have severity level 2 or 3. There are less than 4% patients having severity level 4 or 5. Based on the details shared above, hospital management can consider forming patient profiles based on their age, viral load, severity level, likely treatment required, and then can procure necessary equipment, medicines, and allocate resources (medical staff, doctors etc) accordingly.