Business Problem

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.

Objective:

- To find which variables are significant in predicting the reason for hospitalization for different regions.
- Analysisng how well some variables like viral load, smoking, Severity Level describe the hospitalization charges.

Concept used:

- Uni-Variate Analysis
- Bi-Variate Analysis
- 2-sample t-test
- ANNOVA
- Chi-square

Basic data exploration

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

from scipy.stats import norm, chi2, f # distributions

from scipy.stats import ttest_ind, ttest_rel, f_oneway, kruskal # numerical vs categorical
from scipy.stats import chisquare, chi2_contingency # categorical features
from scipy.stats import pearsonr, spearmanr # numeric vs numeric

from scipy.stats import kstest # cdf

from statsmodels.distributions.empirical_distribution import ECDF
# Empirical CDF
```

```
df = pd.read_csv('/content/scaler_apollo_hospitals.csv')
```

```
df
```

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

1338 rows × 8 columns

```
df.shape
```

```
(1338, 8)
```

```
df.info()
```

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

```
df = df.drop(df.columns[0],axis=1)
```

```
df.isna().sum()
```

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

```
df.nunique()
```

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

```
df.dtypes
```

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

```
df['severity level']= df['severity level'].astype('object')
```

```
cat_cols = list(df.dtypes[df.dtypes =='object'].index)
num_cols = list(df.dtypes[df.dtypes !='object'].index)
cat_cols,num_cols
```

```
(['sex', 'smoker', 'region', 'severity level'],
 ['age', 'viral load', 'hospitalization charges'])
```

```
for i in cat_cols:
  print(df[i].value_counts())
```

```
male      676
female    662
Name: sex, dtype: int64
no     1064
yes     274
Name: smoker, dtype: int64
southeast    364
southwest    325
northwest    325
northeast    324
Name: region, dtype: int64
0    574
1    324
2    240
3    157
4     25
5     18
Name: severity level, dtype: int64
```

Univariate Data Analysis

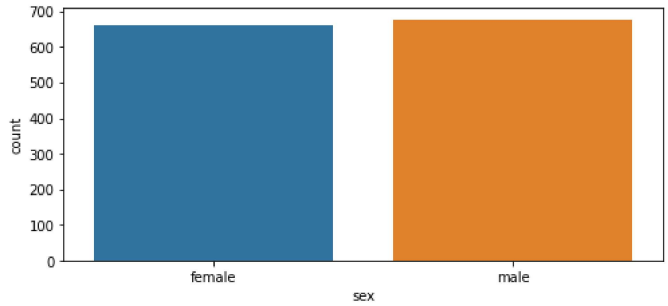
```
df2=df.copy()
cat_count = df2[cat_cols].melt().groupby(['variable', 'value'])[['value']].size().reset_index(name='Counts')
s = df2[cat_cols].melt().variable.value_counts()
cat_count['Percent'] = cat_count['Counts'].div(cat_count['variable'].map(s)).mul(100).round().astype('int')
cat_count.groupby(['variable', 'value','Counts','Percent']).first()

# there are just less than 5% patients with severity level 4 or 5.
```

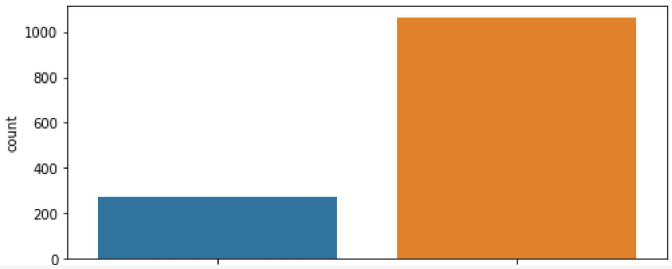
variable	value	Counts	Percent
region	northeast	324	24
	northwest	325	24
	southeast	364	27
	southwest	325	24
severity level	0	574	43
	1	324	24
	2	240	18
	3	157	12
	4	25	2
	5	18	1
sex	female	662	49
	male	676	51
smoker	no	1064	80
	yes	274	20

```
df.describe(include='object').T
```

	count	unique	top	freq
sex	1338	2	male	676
smoker	1338	2	no	1064
region	1338	4	southeast	364
severity level	1338	6	0	574

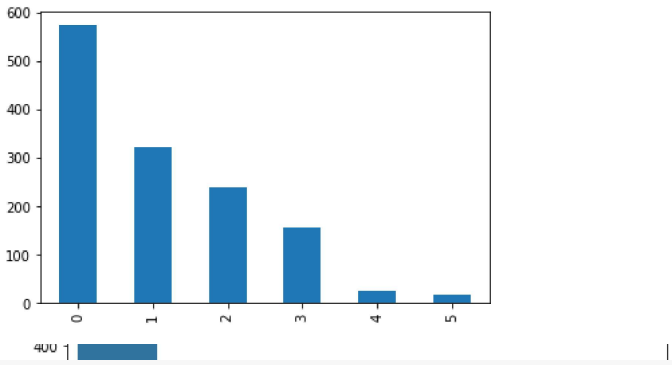
```
plt.figure(figsize = [8,16])
for i in range (len(cat_cols)):
  plt.subplot(len(cat_cols),1, i+1)
  sns.countplot(data=df, x=cat_cols[i])
```






```
1 df['severity level'].value_counts().plot(kind='bar')
```

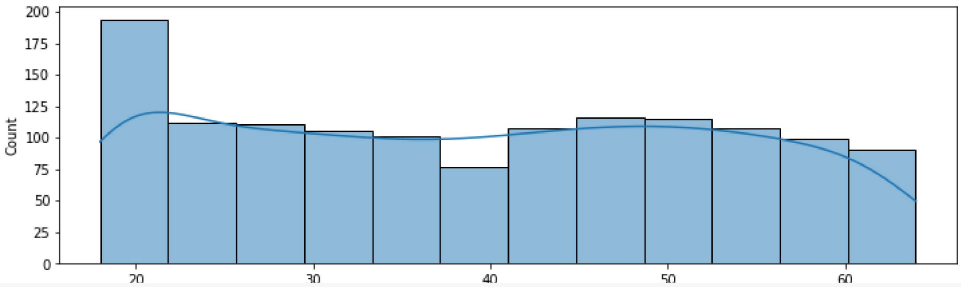
```
<AxesSubplot:>
```



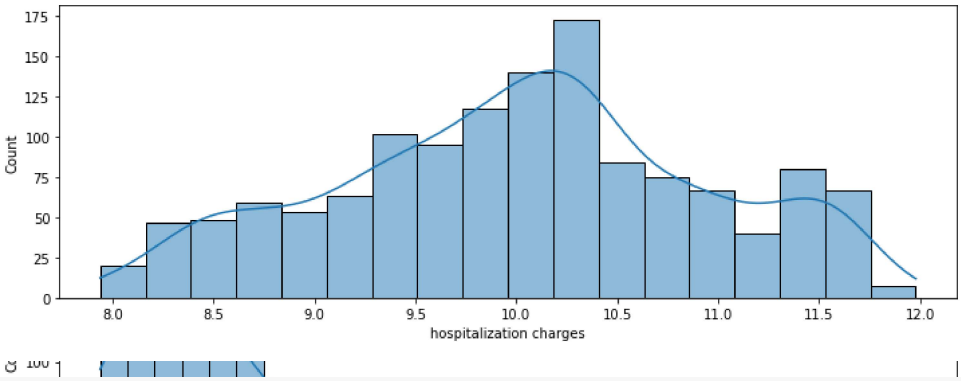

```
1 df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
age	1338.0	39.207025	14.049960	18.00	27.0000	39.00	51.0000	64.00
viral load	1338.0	10.221233	2.032796	5.32	8.7625	10.13	11.5675	17.71
hospitalization charges	1338.0	33176.058296	30275.029296	2805.00	11851.0000	23455.00	41599.5000	159426.00

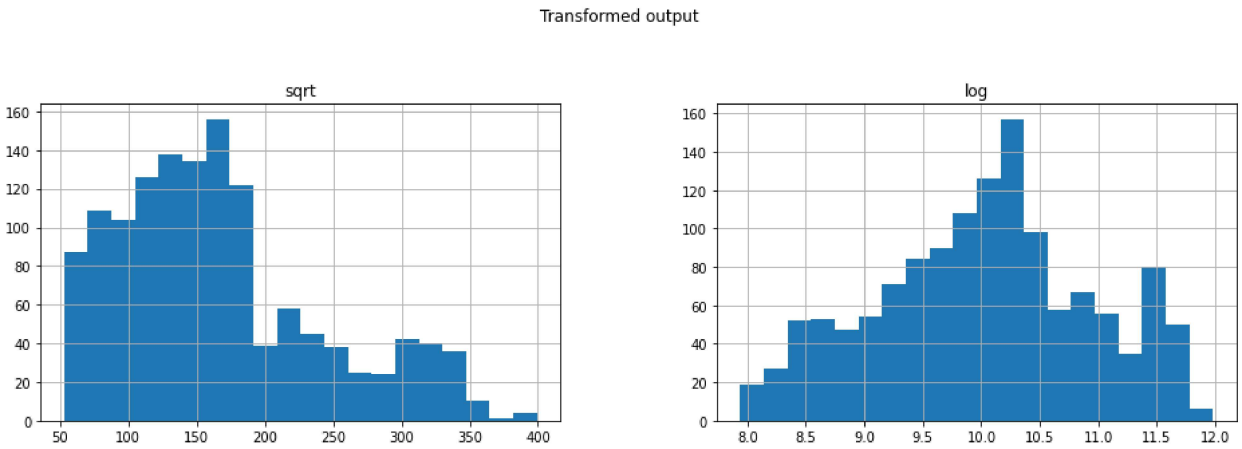
```
1 plt.figure(figsize = [12,12])
2 for i in range (len(num_cols)):
3   plt.subplot(len(num_cols),1, i+1)
4   sns.histplot(data=df, x=num_cols[i], kde=True)
```



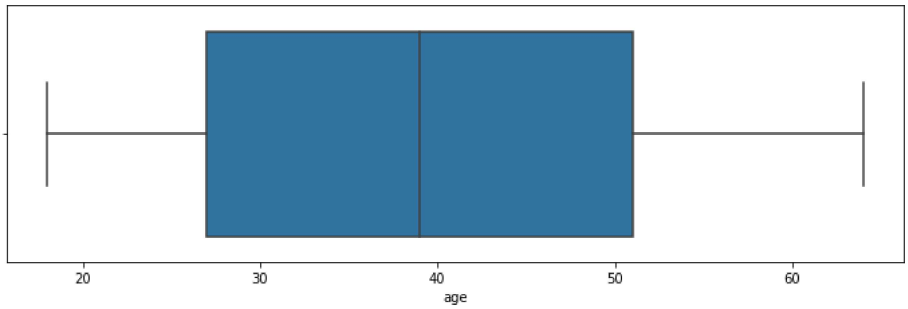

```
plt.figure(figsize = [12,4])
sns.histplot(np.log(df['hospitalization charges']),kde=True)
plt.show()
```

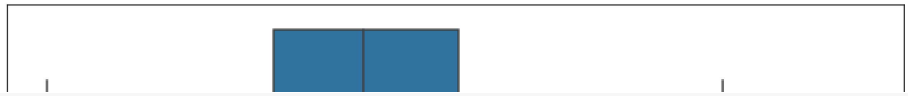



```
trans = df['hospitalization charges'].transform([np.sqrt, np.log])
trans.hist(bins=20, layout=(2,2),figsize=(16,10))
plt.suptitle('Transformed output')
plt.show()
```




```
plt.figure(figsize = [12,12])
for i in range (len(num_cols)):
  plt.subplot(len(num_cols),1, i+1)
  sns.boxplot(data=df, x=num_cols[i])
```



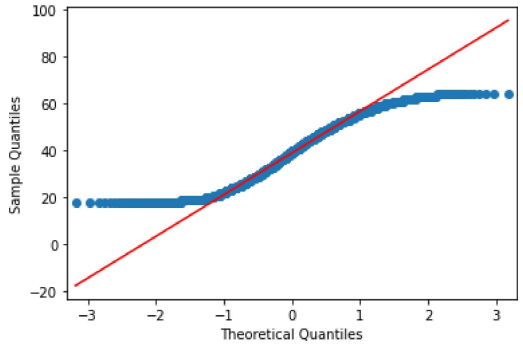


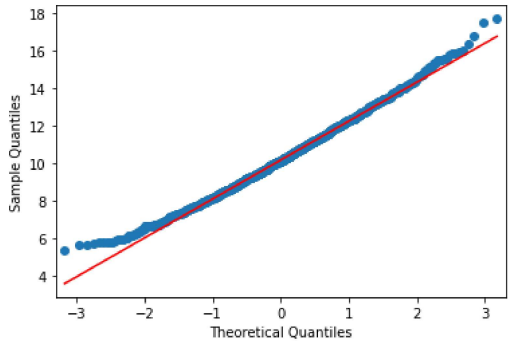
```
for i in (num_cols):
  print(i, round(df[i].skew(),1))

# windspeed distributioni is right skewed, means it has some outliers in right.
```

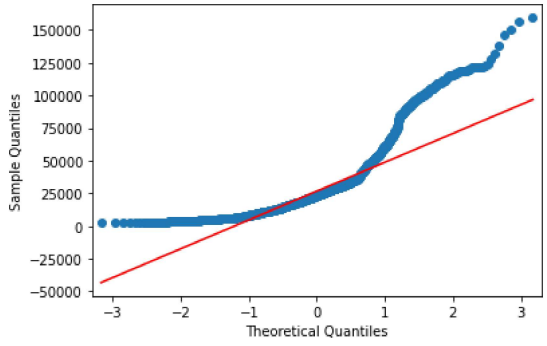
```
age 0.1
viral load 0.3
hospitalization charges 1.5
```

```
import statsmodels.api as sm
import matplotlib.pyplot as plt
plt.figure(figsize = [22,5])
for i in (num_cols):
  sm.qqplot(df[i], line ='q')

# hospitalization charge values clearly do not follow the red line,
# which is an indication that they do not follow a normal distribution.
```

```
<Figure size 1584x360 with 0 Axes>
```








```
for i in num_cols:
  R_whisker = np.percentile(df[i],75)+(np.percentile(df[i],75)-np.percentile(df[i],25))*1.5
```

```
  outliers_percentage = df[df[i]>R_whisker].index.size/df.index.size*100
  print(f'Percentage of outliers in {i} column = ', round(outliers_percentage,2))
```

```
Percentage of outliers in age column =  0.0
Percentage of outliers in viral load column =  0.67
Percentage of outliers in hospitalization charges column =  10.39
```

```
for i in num_cols:
  q1=df[i].quantile(0.25)
  q3=df[i].quantile(0.75)
  IQR=q3-q1
  outliers_percentage = df[((df[i]<(q1-1.5*IQR)) | (df[i]>(q3+1.5*IQR)))].index.size/df.index.size*100
  print(f'Percentage of outliers in {i} column = ', round(outliers_percentage,2))
```

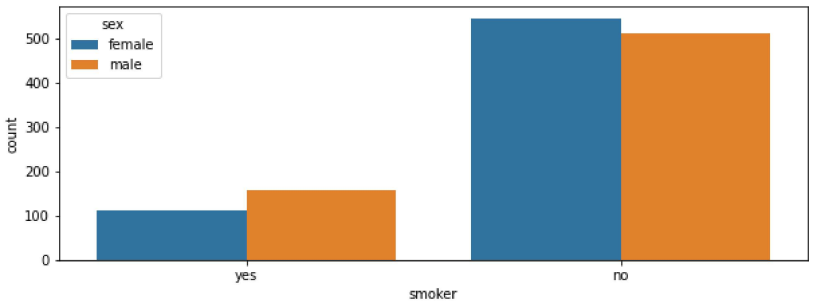
```
Percentage of outliers in age column =  0.0
Percentage of outliers in viral load column =  0.67
Percentage of outliers in hospitalization charges column =  10.39
```

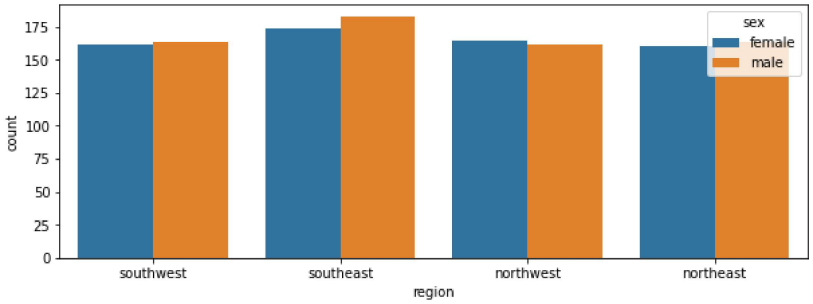
```
q1=df['viral load'].quantile(0.25)
q3=df['viral load'].quantile(0.75)
IQR=q3-q1
df = df[~((df['viral load']<(q1-1.5*IQR)) | (df['viral load']>(q3+1.5*IQR)))]

#·viral·load·column·has·less·than·1%·outliers so we will remove them using IQR method.
```

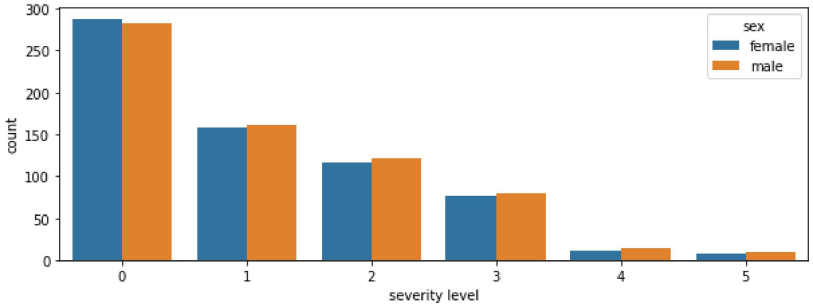
```
p99=df['hospitalization charges'].quantile(0.95)
df = df[~((df['hospitalization charges']<0) | (df['hospitalization charges']>p99))]

# hospitalization charges column has more than 10% outliers so we will just remove extreme 1% of them using percentile method.
```

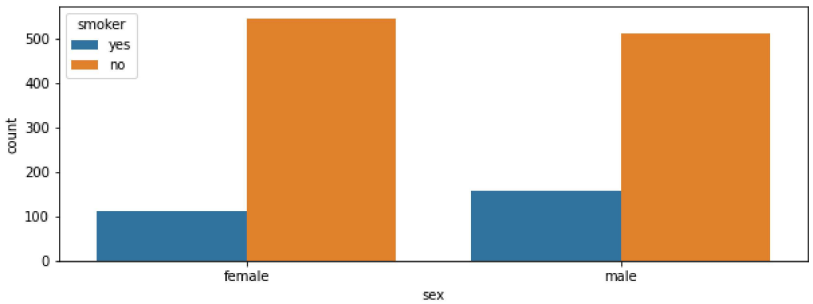
Bivariate Data Analysis

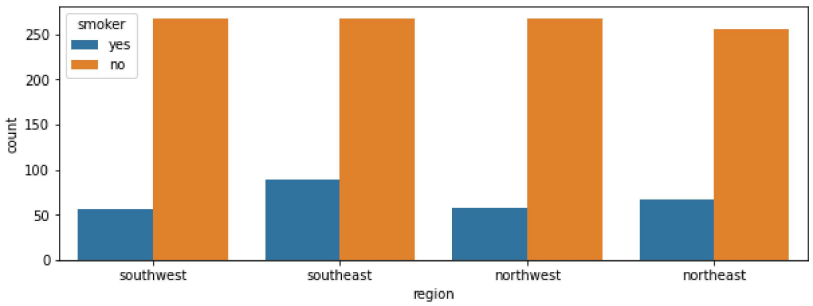
```
for j in range (len(cat_cols)):
  for i in range (len(cat_cols)):
    if i!=j:
      plt.figure(figsize = [10,16])
      plt.subplot(len(cat_cols), 1, i+1)
      sns.countplot(data=df, x=cat_cols[i],hue=cat_cols[j])
      plt.show()
```

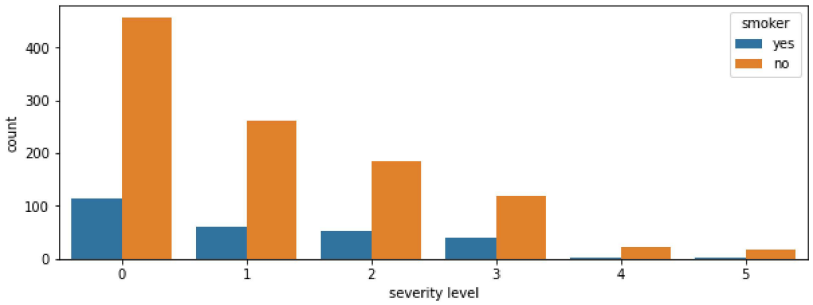

sex
female
male
count
smoker
yes
no

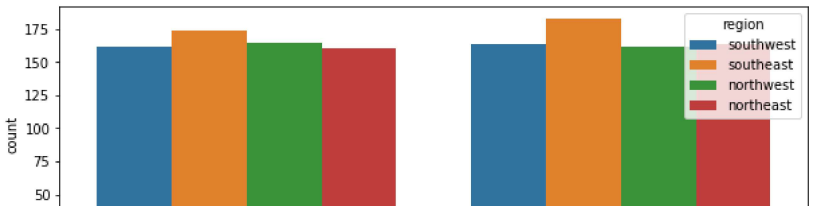

sex
female
male
count
region
southwest
southeast
northwest
northeast

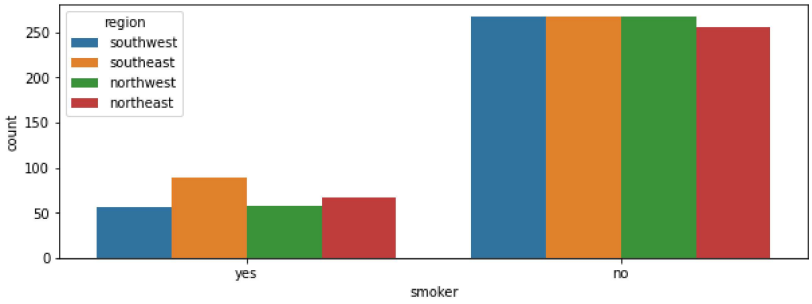

sex
female
male
count
severity level

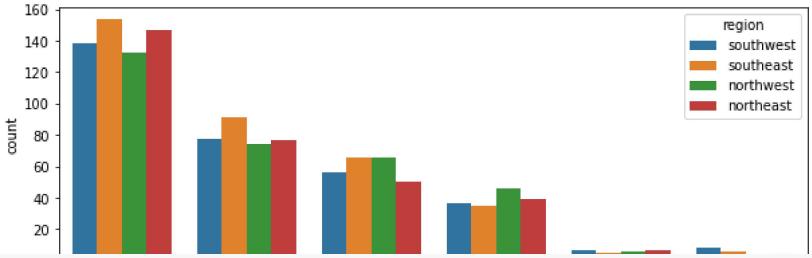

smoker
yes
no
count
sex
female
male


smoker
yes
no
count
region
southwest
southeast
northwest
northeast

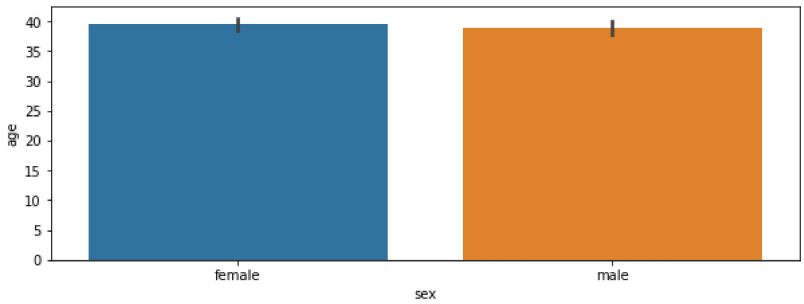

smoker
yes
no
count
severity level

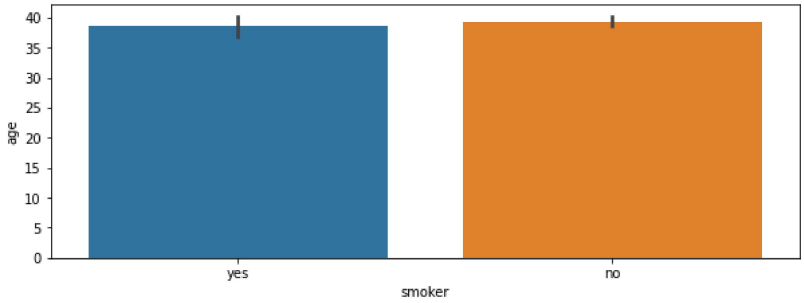

region
southwest
southeast
northwest
northeast
count

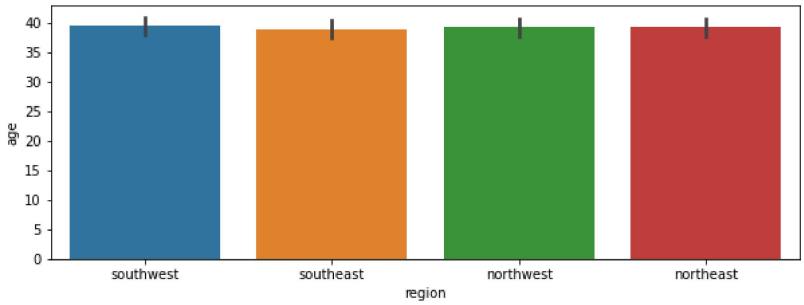


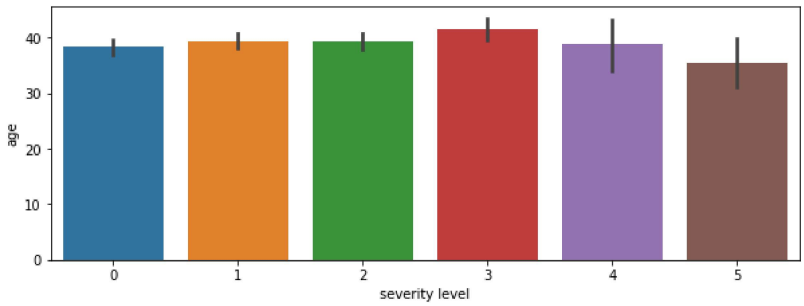


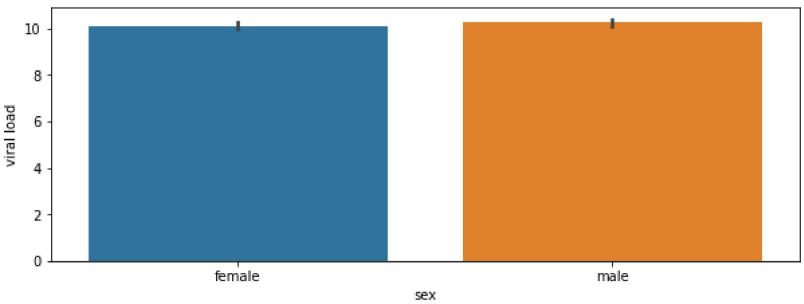

```
for j in range(len(num_cols)):
  for i in range (len(cat_cols)):
    plt.figure(figsize = [10,16])
    plt.subplot(len(cat_cols), 1, i+1)
    sns.barplot(data=df, x=cat_cols[i],y=num_cols[j])
    plt.show()
```

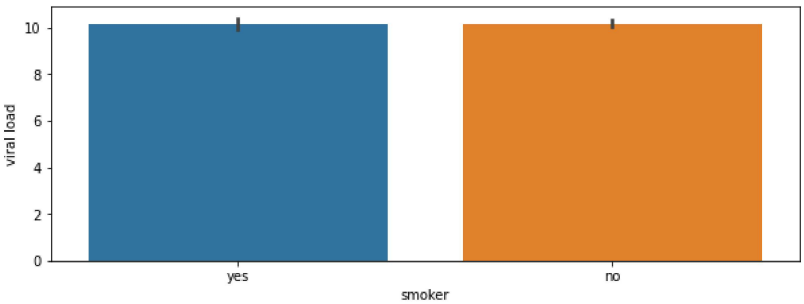

age
sex
female
male

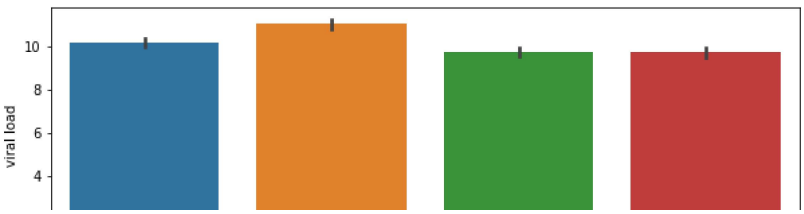

age
smoker
yes
no

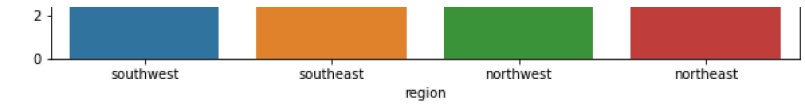

age
region
southwest
southeast
northwest
northeast

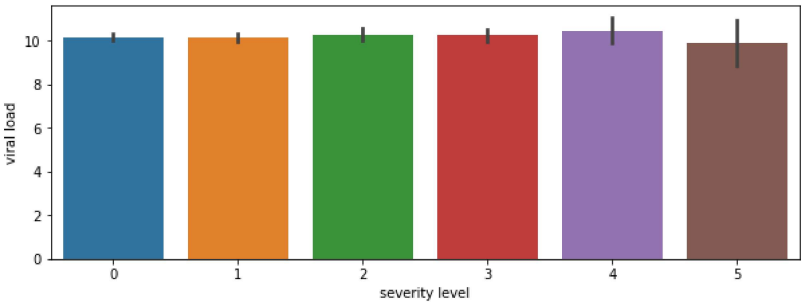

age
severity level


viral load
sex
female
male


viral load
smoker
yes
no

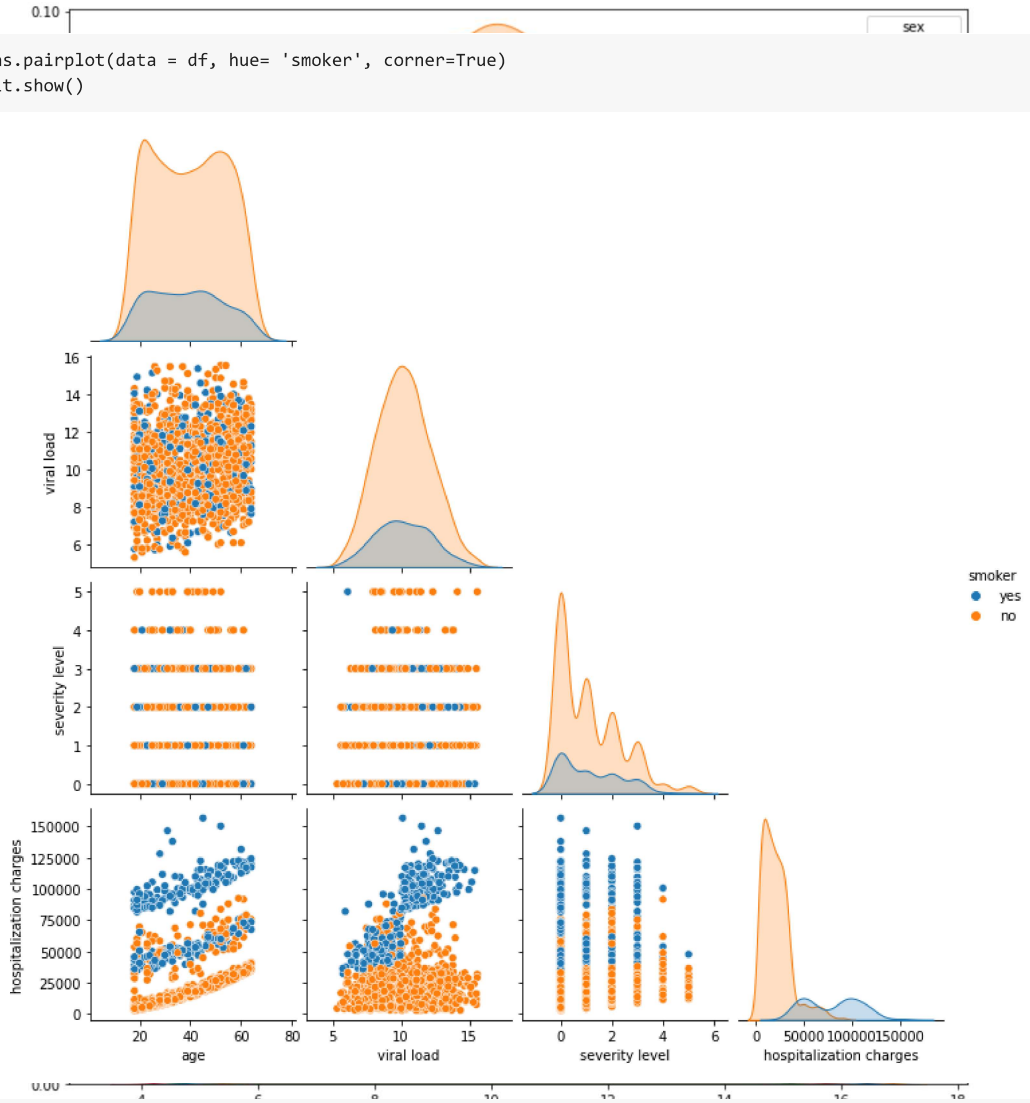

viral load





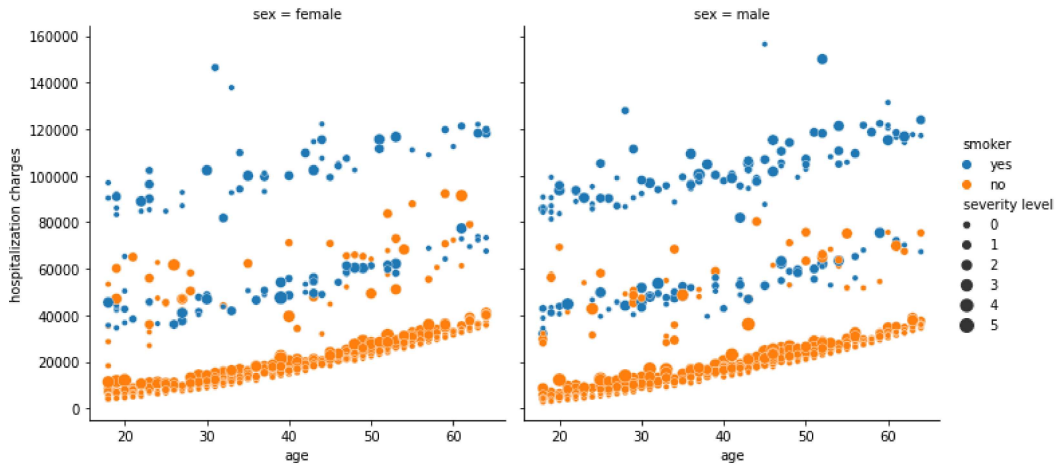

```
plt.figure(figsize = [12,20])
for j in range(len(cat_cols)):
  plt.subplot(len(cat_cols), 1,j+1)
  sns.kdeplot(data=df,x='viral load',hue=cat_cols[j],fill=True)
```

```
sns.pairplot(data = df, hue= 'smoker', corner=True)
plt.show()
```




```
sns.relplot(data=df, y="hospitalization charges", x="age", hue='smoker', size="severity level", sizes=(20, 100), col="sex")

# There are more male who is smoker.
```

```
<seaborn.axisgrid.FacetGrid at 0x7fc047fe50a0>
```



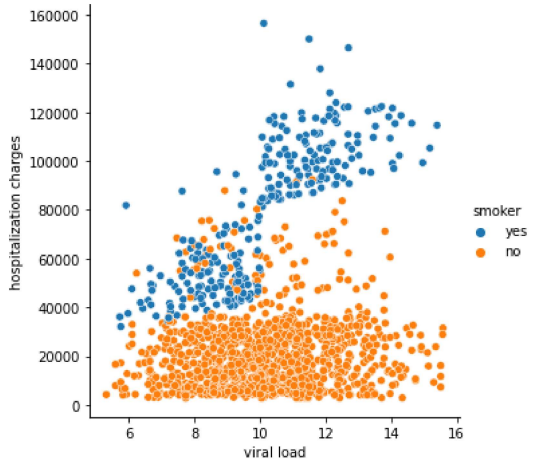

```
df[df['smoker']=='yes'].describe().T
```

	count	mean	std	min	25%	50%	75%	max
age	271.0	38.523985	13.932057	18.00	27.000	38.00	49.000	64.0
viral load	271.0	10.167970	2.013258	5.73	8.685	10.12	11.675	15.4
hospitalization charges	271.0	79588.354244	28461.379883	32074.00	51899.500	85758.00	102342.000	156482.0

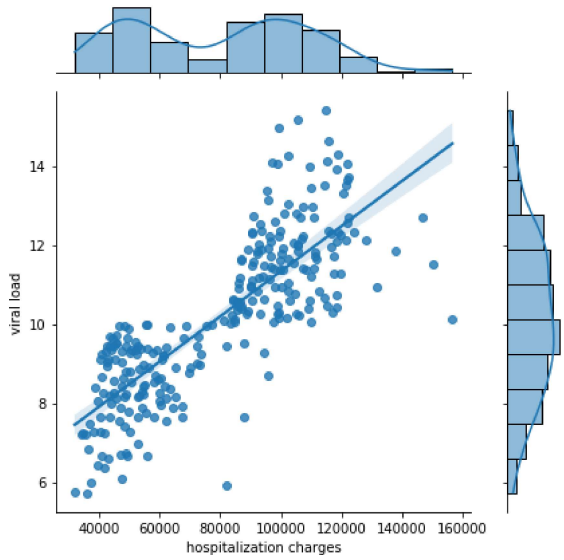
```
df[df['smoker']=='no'].describe().T
```

	count	mean	std	min	25%	50%	75%	max
age	1058.0	39.378072	14.078600	18.00	27.00	40.0	52.000	64.00
viral load	1058.0	10.182108	1.964027	5.32	8.77	10.1	11.455	15.58
hospitalization charges	1058.0	21105.423440	15007.155103	2805.00	9971.25	18353.5	28407.750	92277.00

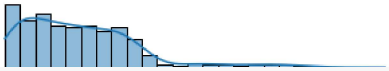
```
sns.relplot(data=df, y="hospitalization charges", x="viral load", hue='smoker')
plt.show()
```



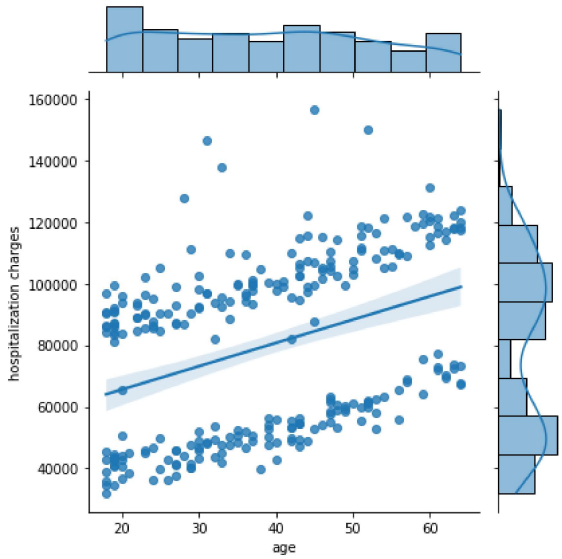

```
sns.jointplot(x='hospitalization charges', y='viral load', data=df[df['smoker']=='yes'], kind='reg')
plt.tight_layout()
plt.show()
```



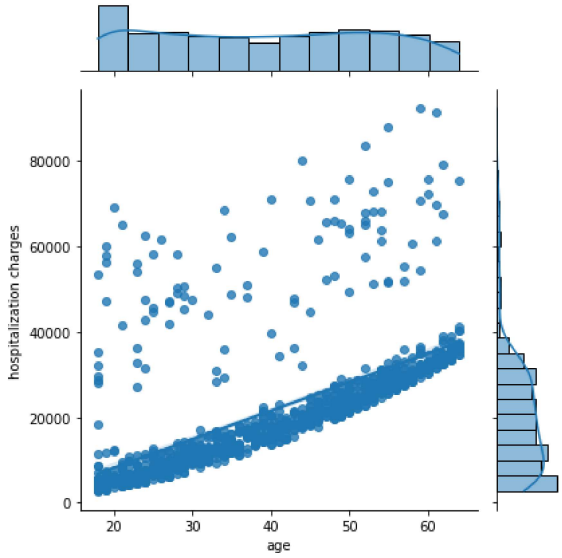

```
sns.jointplot(x='hospitalization charges', y='viral load', data=df[df['smoker']=='no'], kind='reg')
plt.tight_layout()
plt.show()
```



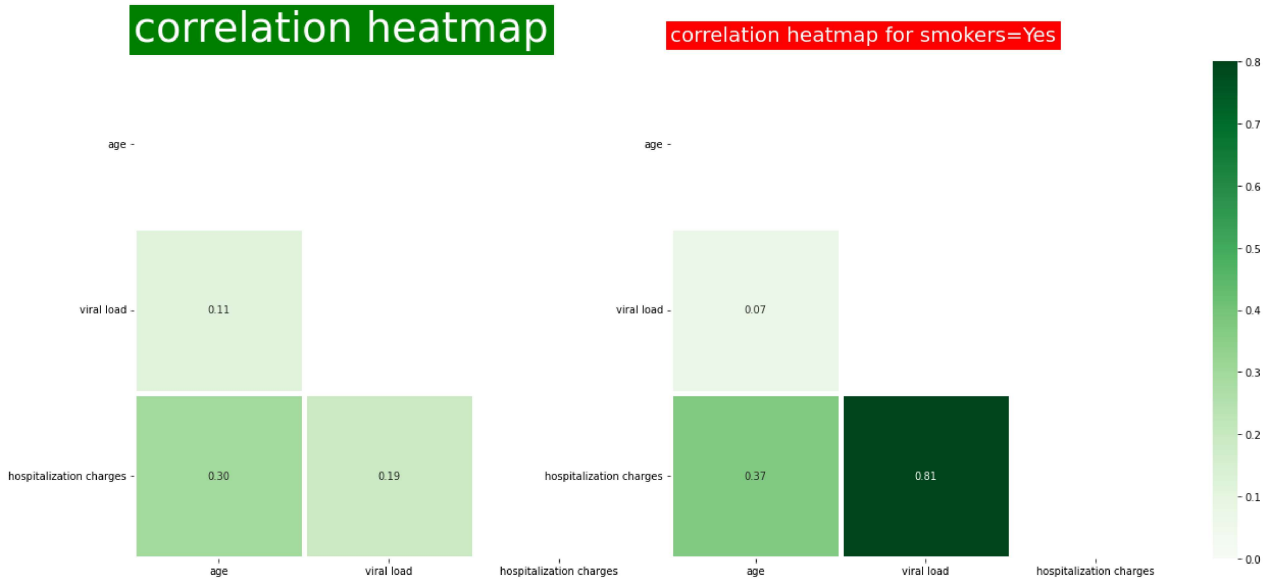
```
sns.jointplot(y='hospitalization charges', x='age', data=df[df['smoker']=='yes'], kind='reg')
plt.tight_layout()
plt.show()
```




```
sns.jointplot(y='hospitalization charges', x='age', data=df[df['smoker']=='no'], kind='reg')
plt.tight_layout()
plt.show()
```




```
fig=plt.figure(figsize = [18,8])
df_somoker = df.copy()[df['smoker']=='yes']

plt.subplot(1, 2, 1)
mask = np.triu(np.ones_like(df.corr()))
sns.heatmap(df.corr().round(2), cmap= "Greens", annot=True, mask=mask, fmt=".2f", linewidths=5, vmin=0, vmax=0.8)
plt.tight_layout()
plt.title('correlation heatmap', color='w',fontsize=40, fontweight = 'normal',backgroundcolor = 'g', pad = 20, loc='left')
plt.yticks(rotation = 0)

plt.subplot(1, 2, 2)
mask = np.triu(np.ones_like(df_somoker.corr()))
sns.heatmap(df_somoker.corr().round(2), cmap= "Greens", annot=True, mask=mask, fmt=".2f", linewidths=5, vmin=0, vmax=0.8)
plt.tight_layout()
plt.title('correlation heatmap for smokers=Yes', color='w',fontsize=20, fontweight = 'normal',backgroundcolor = 'r', pad = 20, loc='le
plt.yticks(rotation = 0)
plt.show()

# In general there is no correlation greater than 0.5 percent between any two variables,
# But if we see correlation for onle smoker patients then,
# we can observe that there is highly correlation between viral load and hospitalization charge
# and also there is correlation between age and hospitalization charge
```



Missing values treatment & Outlier treatment

```
# dataset has no missing values, checked in first section.
# Outiler has treated in second section with IQR method (for few outliers) and with percentile methods (for many outliers).
```

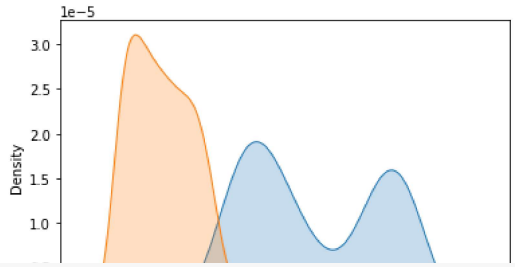
Hypothesis Testing

```
def Hypothesis_testing(Samples,alpha,alternative,mu,crosstab):

  if alternative=='two-sided':
    H0 = 'means are equal'
  elif alternative== 'less':
      H0 = 'Sample1>=Sample2'
  elif alternative=='greater':
    H0 = 'Sample1<=Sample2'
  else:
    print('Check the alternative input')
    exit()
  H0_chi_stat = 'obs and exp values are same, proportions are similar'
  H0_chi2_contingency = 'variables are independet, liklihoods are similar'
  if not all(Samples):
    print('Check Samples detail input')
    exit()

  if type(Samples[0][0])==list:

    if sum(Samples[0][0])-sum(Samples[0][1])<0.001:
      print('Chi square test for goodness of fit (obs,exp)')
      from scipy.stats import chisquare
      test, p_val = chisquare(*Sample1)
      print("p-value is: " + str(p_val))
      print('null hypothesis: ',H0_chi_stat)
      if p_val <= alpha:
          print('We can reject the null hypothesis')
      else:
          print('We can accept the null hypothesis')

    else:
      print('Chi square test for indepedence (contingency)')
      from scipy.stats import chi2_contingency
      test, p_val, dof, expected_val = chi2_contingency(Sample1)
      print("p-value is: " + str(p_val))
      print('null hypothesis: ',H0_chi2_contingency)
      if p_val <= alpha:
          print('We can reject the null hypothesis')
      else:
          print('We can accept the null hypothesis')

```

```
  elif len(Samples)>2:
    print(len(Samples), 'Sample','Anova f test')
    from scipy.stats import f_oneway
    f_test, p_val = f_oneway(*Samples)
    print("p-value is: " + str(p_val))
    print('null hypothesis: ',H0)
    if p_val < alpha:
        print("We can reject the null hypothesis")
    else:
        print("We can accept the null hypothesis")

  elif len(Samples)>1:
    print(len(Samples), 'Sample','t-test')
    from scipy.stats import ttest_ind
    t_test,p_val = ttest_ind(*Samples)
    print("p-value of independent t-test is: ", p_val)
    print('null hypothesis: ',H0)
    if p_val < alpha:
        print("We can reject the null hypothesis")
    else:
        print("We can accept the null hypothesis")

    if len(Samples[0])==len(Samples[1]):
      from scipy.stats import ttest_rel
      t_test, p_val = ttest_rel(*Samples, alternative=alternative)
      print("p-value of the paried t-test is: ", p_val)
      print('null hypothesis: ',H0)
      if p_val < 0.05:
          print("We can reject the null hypothesis")
      else:
          print("We can accept the null hypothesis")


  elif len(Samples)==1:
    print(len(Samples), 'Sample','t test')
    from scipy.stats import ttest_1samp
    t_stat, p_val = ttest_1samp(Samples[0],mu, alternative=alternative)
    print("P-value is: ", p_val)
    print('null hypothesis: ',H0)
    if p_val < 0.05:
        print(" We can reject the null hypothesis")
    else:
        print("We can accept the null hypothesis")

  import matplotlib.pyplot as plt
  import seaborn as sns
  if type(Sample1[0])==list:
    crosstab.plot(kind="bar",figsize=(8,5))
    print(crosstab)
  else:
    for samp in (Samples):
      sns.kdeplot(samp,fill=True)
    plt.show()

    import numpy as np
    for i in range(len(Samples)):
      if alternative=='less':
        print(f'mean of Sample{i+1}:',np.mean(Samples[i]).round(2), '       95% Range [5%-100%] :', [np.percentile(Samples[i],5).round
      elif alternative=='greater':
        print(f'mean of Sample{i+1}:',np.mean(Samples[i]).round(2), '       95% Range [0%-95%] :', [np.min(Samples[i]).round(2),np.per
      else:
        print(f'mean of Sample{i+1}:',np.mean(Samples[i]).round(2), '       95% Range [2.5%-97.7%] :', [np.percentile(Samples[i],2.5).
```

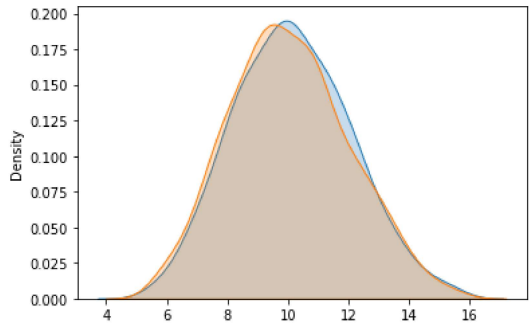
```
# null hypothesis = hospitalization charge of people who do smoking is lesser than those who don't.
# alternative hypothesis = hospitalization charge of people who do smoking is greater than those who don't.

Sample1=df[df['smoker']=='yes']['hospitalization charges'].tolist()
Sample2=df[df['smoker']=='no']['hospitalization charges'].tolist()
Samples=[Sample1,Sample2]
alpha = 0.05
alternative = 'greater'   # 'two-sided'  # 'less' , 'greater'
mu = None
crosstab=None
Hypothesis_testing(Samples,alpha,alternative,mu,crosstab)

# conclusion: hospitalization charge of people who do smoking is greater than those who don't.
```

```
2 Sample t-test
p-value of independent t-test is:  5.209104236115724e-207
null hypothesis:  Sample1<=Sample2
We can reject the null hypothesis
```




```
# null hypothesis =  viral load of females is same as males.
# null hypothesis =  viral load of females is different as males.

Sample1=df[df['sex']=='male']['viral load'].tolist()
Sample2=df[df['sex']=='female']['viral load'].tolist()
Samples=[Sample1,Sample2]
alpha = 0.05
alternative = 'two-sided'  # 'two-sided'  # 'less' , 'greater'
mu = None
crosstab=None
Hypothesis_testing(Samples,alpha,alternative,mu,crosstab)

# conclusion: viral load of females is same as males
```

```
2 Sample t-test
p-value of independent t-test is:  0.3144637521964963
null hypothesis:  means are equal
We can accept the null hypothesis
```



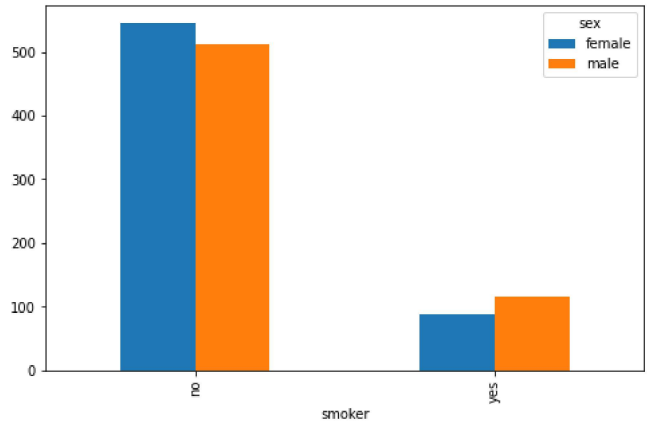

```
mean of Sample1: 10.13        95% Range [2.5%-97.7%] : [6.64, 14.07]
mean of Sample2: 10.02        95% Range [2.5%-97.7%] : [6.47, 13.98]
```

```
# null hypothesis =  proportion of smoking significantly similar across different regions.
# alternative hypothesis =  proportion of smoking significantly different across different regions.

import pandas as pd
crosstab = pd.crosstab(index=df['smoker'], columns=df['region'])   # Sample1 = crosstab.values.tolsit()
Sample1=crosstab.values.tolist()
Samples=[Sample1]
mu = None
alpha = 0.05
alternative = 'two-sided'  # 'less' , 'greater'
Hypothesis_testing(Samples,alpha,alternative,mu,crosstab)

# Conclusion: proportion of smoking significantly similar across different regions.
```

```
Chi square test for indepedence (contingency)
p-value is: 0.558169723528082
null hypothesis:  variables are independet, liklihoods are similar
We can accept the null hypothesis
region  northeast  northwest  southeast  southwest
smoker
```

```
# null hypothesis =  proportion of smoking significantly similar across different gender.
# alternative hypothesis =  proportion of smoking significantly different across different gender.

import pandas as pd
crosstab = pd.crosstab(index=df['smoker'], columns=df['sex'])   # Sample1 = crosstab.values.tolsit()
Sample1=crosstab.values.tolist()
Samples=[Sample1]
mu = None
alpha = 0.05
alternative = 'two-sided'  # 'less' , 'greater'
Hypothesis_testing(Samples,alpha,alternative,mu,crosstab)

# conclusion: we can say that there are more males who are smoker.
```

```
Chi square test for indepedence (contingency)
p-value is: 0.03245297207395062
null hypothesis:  variables are independet, liklihoods are similar
We can reject the null hypothesis
sex     female  male
smoker
no         546   512
yes         88   116
```

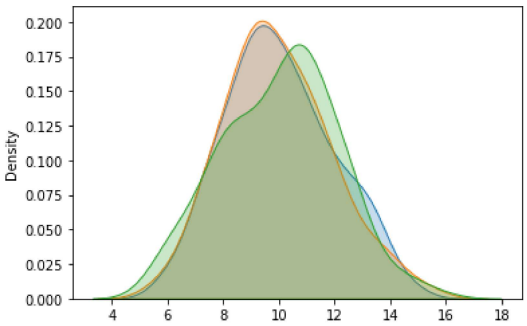



```
# null hypothesis =  proportion of smoking significantly similar across different severity level.
# alternative hypothesis =  proportion of smoking significantly different across different severity level.

import pandas as pd
crosstab = pd.crosstab(index=df['smoker'], columns=df['severity level'])   # Sample1 = crosstab.values.tolsit()
Sample1=crosstab.values.tolist()
Samples=[Sample1]
mu = None
alpha = 0.05
alternative = 'two-sided'  # 'less' , 'greater'
Hypothesis_testing(Samples,alpha,alternative,mu,crosstab)

# conclusion: we can not say that people who smoke has higher severity level.
```

```
Chi square test for indepedence (contingency)
p-value is: 0.5963357865679622
null hypothesis:  variables are independet, liklihoods are similar
We can accept the null hypothesis
```

```
# null hypothesis =  the mean viral load of women with 0 Severity level, 1 Severity level, and 2 Severity level are the same.
# null hypothesis =  the mean viral load of women with 0 Severity level, 1 Severity level, and 2 Severity level are different.

cf=df['sex']=='female'
c0=df['severity level']==0
c1=df['severity level']==1
c2=df['severity level']==2

Sample1=df[(cf) & (c0)]['viral load'].tolist()
Sample2=df[(cf) & (c1)]['viral load'].tolist()
Sample3=df[(cf) & (c2)]['viral load'].tolist()
Samples=[Sample1,Sample2,Sample3]
alpha = 0.05
alternative = 'two-sided'  # 'two-sided'  # 'less' , 'greater'
mu = None
crosstab=None
Hypothesis_testing(Samples,alpha,alternative,mu,crosstab)

# conclusion: the mean viral load of women with 0 Severity level, 1 Severity level, and 2 Severity level are the same.
```

```
3 Sample Anova f test
p-value is: 0.90641731132652
null hypothesis:  means are equal
We can accept the null hypothesis
```




```
mean of Sample1: 10.02        95% Range [2.5%-97.7%] : [6.73, 13.79]
mean of Sample2: 9.94        95% Range [2.5%-97.7%] : [6.68, 14.06]
mean of Sample3: 10.03        95% Range [2.5%-97.7%] : [5.97, 14.26]
```

```
# hospitalization charges column is not normally distributed
# we can perform nonparametric statistical hypothesis tests

# null hypothesis = hospitalization charge of people who do smoking is lesser than those who don't.
# alternative hypothesis = hospitalization charge of people who do smoking is greater than those who don't.

from scipy.stats import mannwhitneyu
sample1 = df[df['smoker']=='yes']['hospitalization charges'].tolist()
sample2 = df[df['smoker']=='no']['hospitalization charges'].tolist()
stat, p = mannwhitneyu(sample1, sample2)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
 print('Probably the same distribution, accept null hypothesis.')
else:
 print('Probably different distributions, reject null hypothesis.')
```

```
stat=208428.500, p=0.000
Probably different distributions, reject null hypothesis.
```

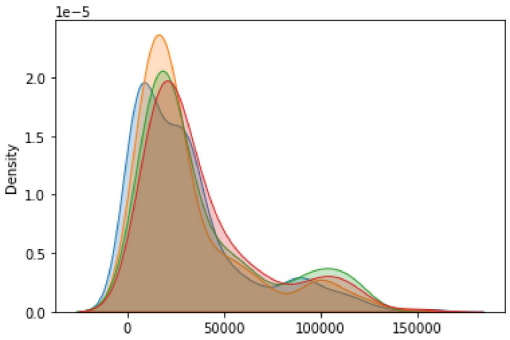
```
# hospitalization charges column is not normally distributed
# we can perform nonparametric statistical sypothesis tests

# null hypothesis = hospitalization charges are similar across different severity level.
# alternative hypothesis = hospitalization charges are different across different severity level.

from scipy.stats import mannwhitneyu
sample1 = df[df['severity level']==0]['hospitalization charges'].tolist()
sample2 = df[df['severity level']==1]['hospitalization charges'].tolist()
sample3 = df[df['severity level']==2]['hospitalization charges'].tolist()
sample4 = df[df['severity level']==3]['hospitalization charges'].tolist()
stat, p = kruskal(sample1, sample2, sample3, sample4)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
 print('Probably the same distribution, accept null hypothesis.')
else:
 print('Probably different distributions, reject null hypothesis.')

```

```
# Conclusion: hospitalization charges are different across different severity level.
```

```
stat=27.000, p=0.000
Probably different distributions, reject null hypothesis.
```

```
# parametric test is not applicable hospitalization charges,
# but we are just comparing the result with previous non parametric test.

sample1 = df[df['severity level']==0]['hospitalization charges'].tolist()
sample2 = df[df['severity level']==1]['hospitalization charges'].tolist()
sample3 = df[df['severity level']==2]['hospitalization charges'].tolist()
sample4 = df[df['severity level']==3]['hospitalization charges'].tolist()
Samples = [sample1, sample2, sample3, sample4]
alpha = 0.05
alternative = 'greater'  # 'two-sided'  # 'less' , 'greater'
mu = None
crosstab=None
Hypothesis_testing(Samples,alpha,alternative,mu,crosstab)

# Conclusion: hospitalization charges are different across different severity level.
```

```
4 Sample Anova f test
p-value is: 0.0035397453106568765
null hypothesis:  Sample1<=Sample2
We can reject the null hypothesis
```




```
mean of Sample1: 30914.94        95% Range [0%-95%] : [2805, 97698.45]
mean of Sample2: 31827.94        95% Range [0%-95%] : [4278, 102023.6]
mean of Sample3: 37683.91        95% Range [0%-95%] : [5760, 110514.25]
mean of Sample4: 38388.31        95% Range [0%-95%] : [8608, 106125.4]
```

Business Insights and Recommendations

```
# There are less patients who are smoker.
# 80% patients does not smoke.
# There are more number of males who are smoker than females.
# there are just less than 5% patients with higher severity level 4 or 5.
# there 43% patients with very low severity level 0.
# most patients belong to southeast region.
# maximum hospitalization charge has paid by patient is 159426.
# minimum hospitalization charge has paid by patient is 2805.
# age column is showing continuous uniform distribution.
# viral load column is showing continuous normal distribution.
# hospitalization charge column is showing lognormal distribution.
# dataset conatins no missing values.
# age column has no outliers present in dataset.
# viral load column has less than 1% outliers, we removed.
# hospitalization charge column has more than 10% outliers, removed 1%.
# In general there is no correlation between any two variables,
# But if we see correlation for onle smoker patients then,
# we can observe that there is highly correlation between viral load and hospitalization charge
# and also there is correlation between age and hospitalization charge
# using parametic test we conclueded that:
# hospitalization charge of people who do smoking is greater than those who don't.
# proportion of smoking significantly similar across different regions.
# viral load of females is same as males.
# we can say that there are more males who are smoker.
# we can not say that people who smoke has higher severity level.
# the mean viral load of women with 0 Severity level, 1 Severity level, and 2 Severity level are the same.
# hospitalization charges are different across different severity level.
```

✓ 0s completed at 11:24 PM