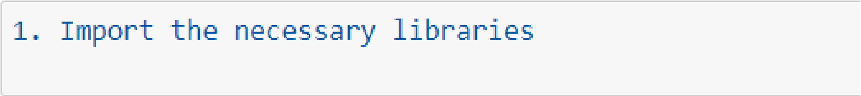
BUSINESS REPORT - HEALTHCARE INSURANCE

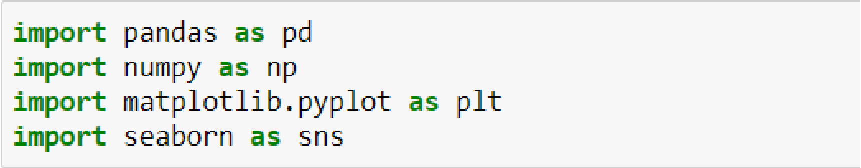
DONE BY: ASHWATH J

1. Import the necessary libraries

Libraries like pandas,numpy,matplotlib and seaborn are imported



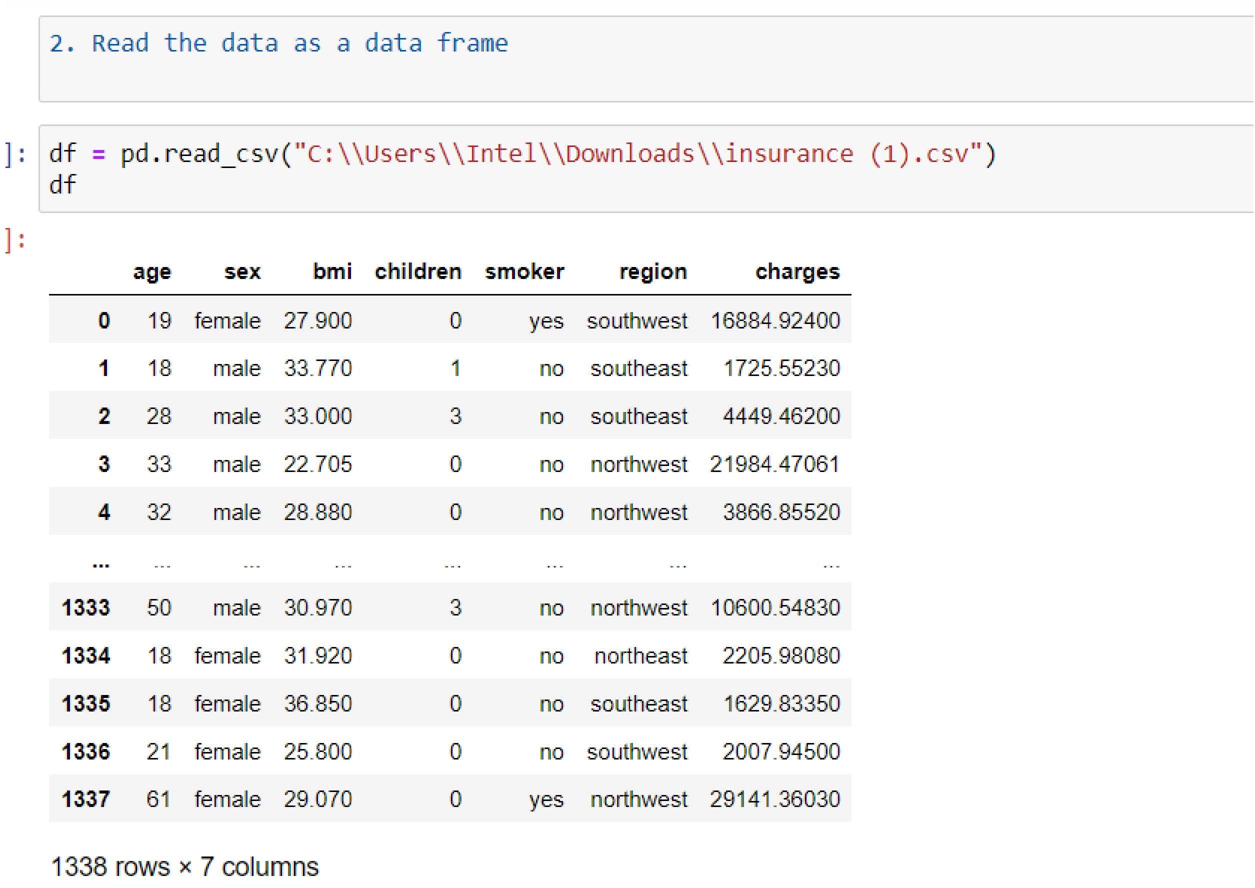
1. Import the necessary libraries



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

2. Read the data as a data frame

Read_csv method is used to read the csv file.



2. Read the data as a data frame

```python
]: df = pd.read_csv("C:\\Users\\Intel\\Downloads\\insurance (1).csv")
   df
```

]:

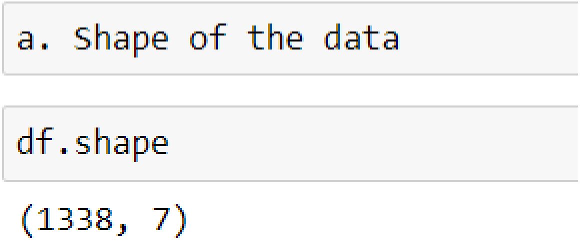
	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

BUSINESS REPORT - HEALTHCARE INSURANCE

3. Perform basic EDA which should include the following and print out your insights at every step.

a. Shape of the data

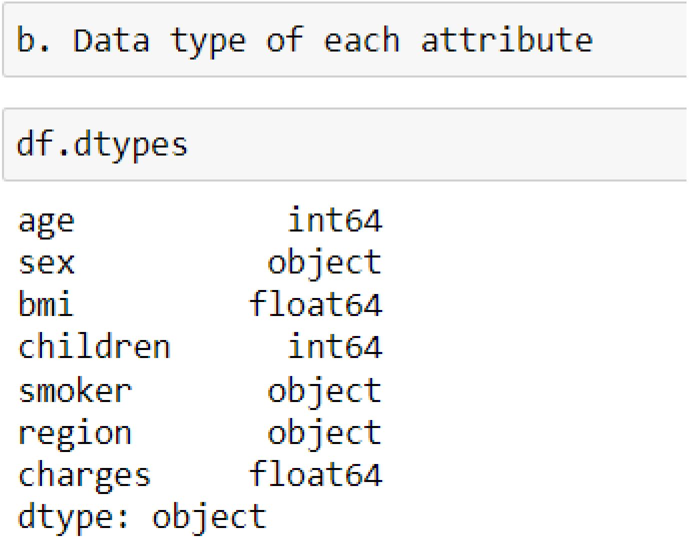


a. Shape of the data

```
df.shape
```

```
(1338, 7)
```

b. Data type of each attribute



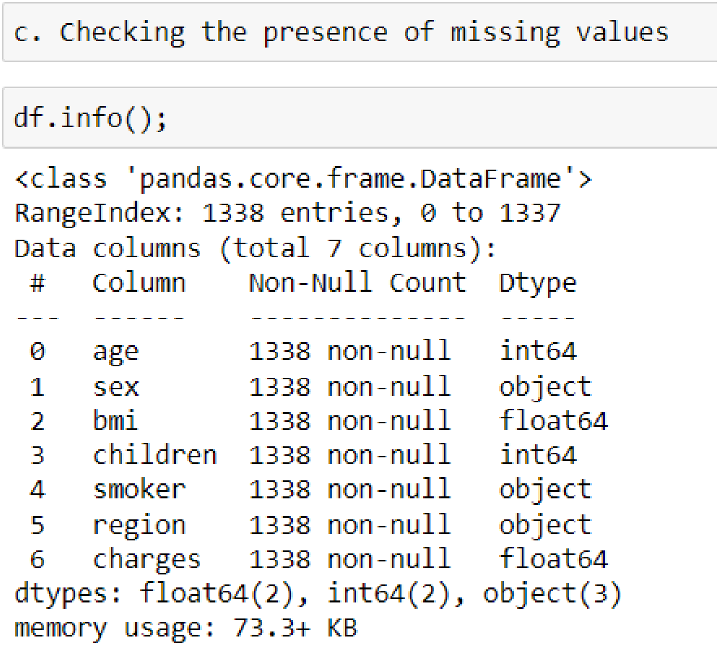
b. Data type of each attribute

```
df.dtypes
```

```
age           int64
sex          object
bmi         float64
children      int64
smoker       object
region       object
charges     float64
dtype: object
```

BUSINESS REPORT - HEALTHCARE INSURANCE

c. Checking the presence of missing values

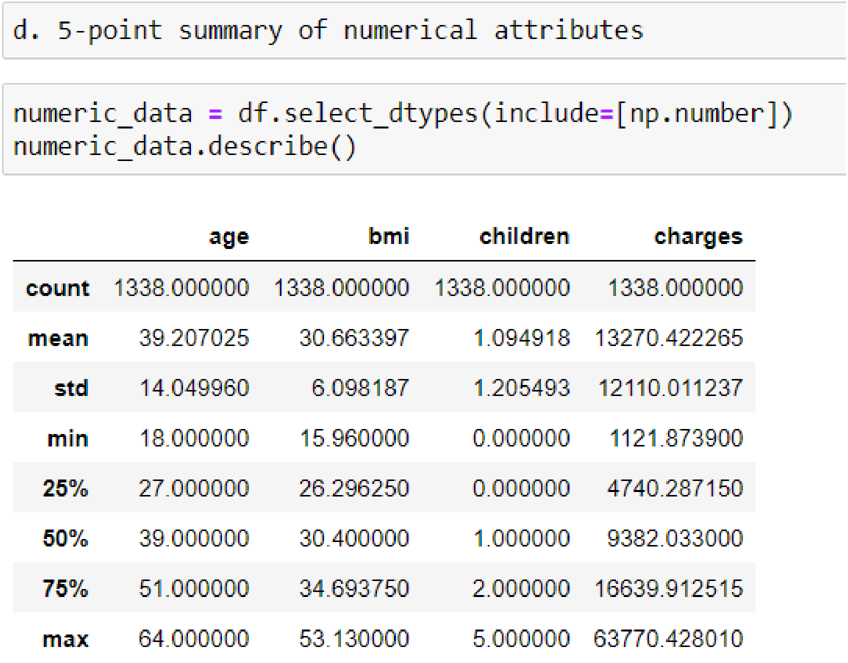


c. Checking the presence of missing values

```
df.info();
```

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

d. 5-point summary of numerical attributes

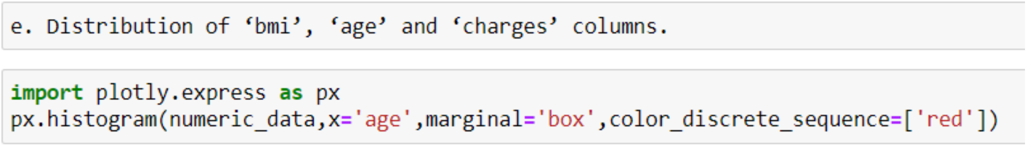


d. 5-point summary of numerical attributes

```
numeric_data = df.select_dtypes(include=[np.number])
numeric_data.describe()
```

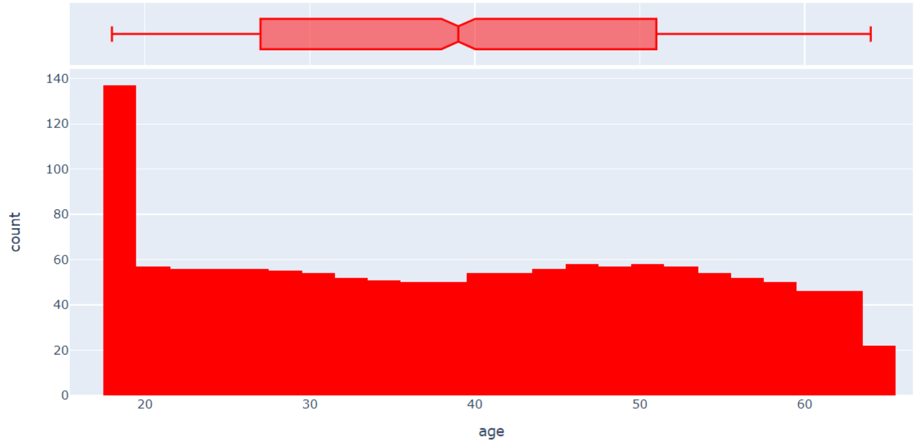
	age	bmi	children	charges
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.663397	1.094918	13270.422265
std	14.049960	6.098187	1.205493	12110.011237
min	18.000000	15.960000	0.000000	1121.873900
25%	27.000000	26.296250	0.000000	4740.287150
50%	39.000000	30.400000	1.000000	9382.033000
75%	51.000000	34.693750	2.000000	16639.912515
max	64.000000	53.130000	5.000000	63770.428010

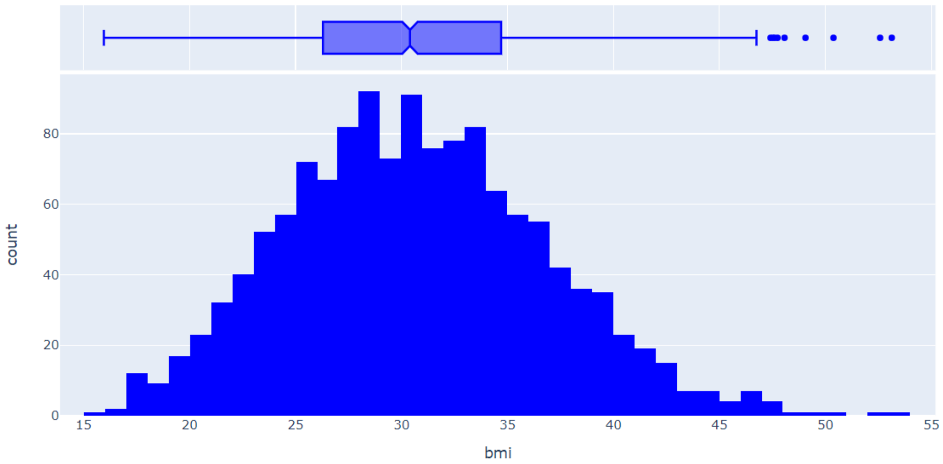
e. Distribution of ‘bmi’, ‘age’ and ‘charges’ columns.



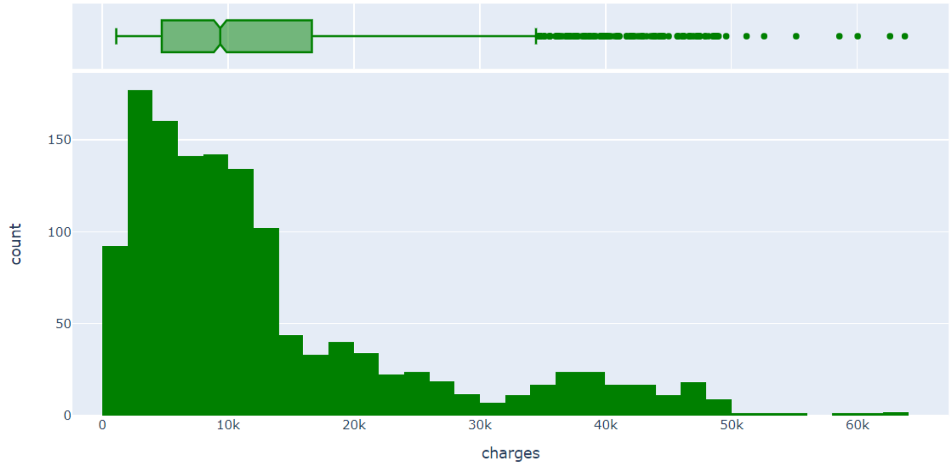
```
e. Distribution of ‘bmi’, ‘age’ and ‘charges’ columns.
```

```python
import plotly.express as px
px.histogram(numeric_data,x='age',marginal='box',color_discrete_sequence=['red'])
```

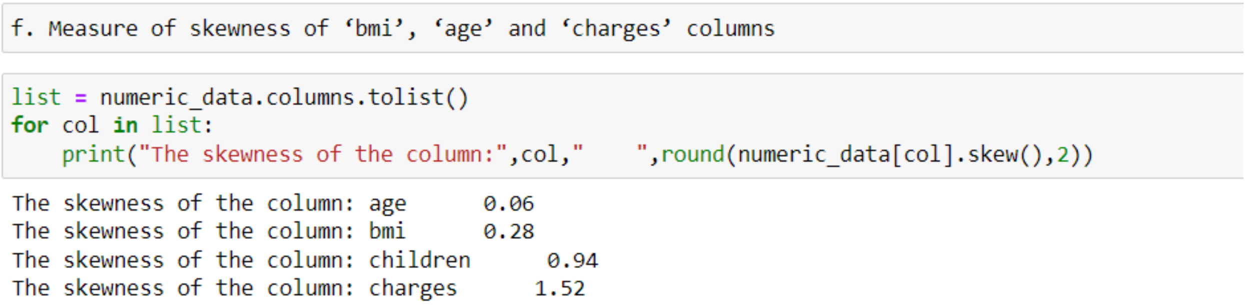




BUSINESS REPORT - HEALTHCARE INSURANCE



f. Measure of skewness of ‘bmi’, ‘age’ and ‘charges’ columns

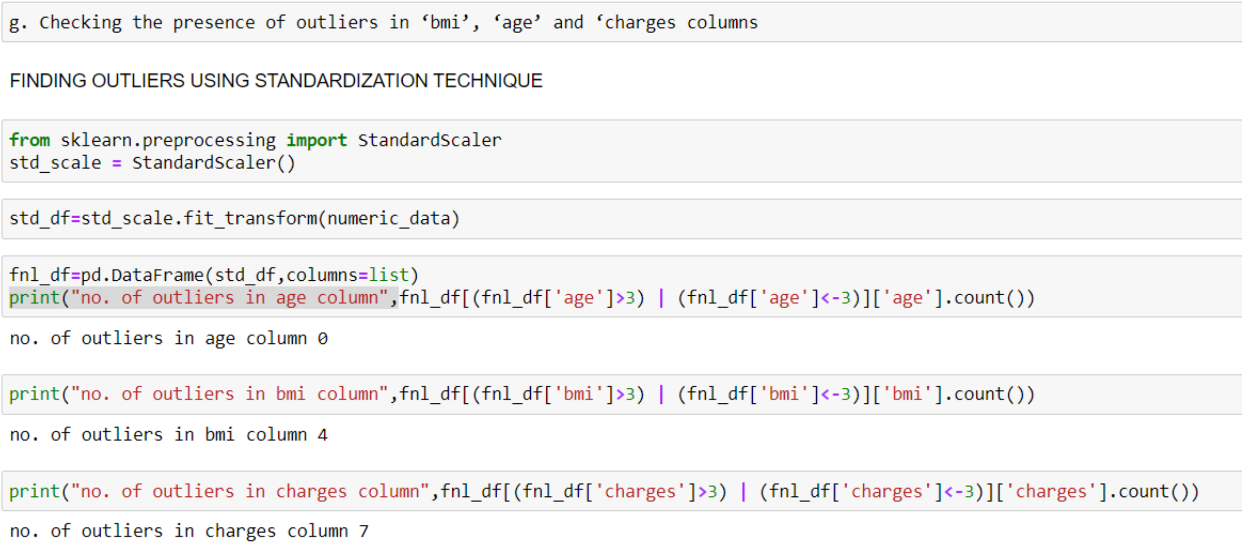


f. Measure of skewness of ‘bmi’, ‘age’ and ‘charges’ columns

```
list = numeric_data.columns.tolist()
for col in list:
    print("The skewness of the column:",col,"     ",round(numeric_data[col].skew(),2))
```

```
The skewness of the column: age       0.06
The skewness of the column: bmi       0.28
The skewness of the column: children       0.94
The skewness of the column: charges       1.52
```

g. Checking the presence of outliers in ‘bmi’, ‘age’ and ‘charges columns



g. Checking the presence of outliers in ‘bmi’, ‘age’ and ‘charges columns

FINDING OUTLIERS USING STANDARDIZATION TECHNIQUE

```
from sklearn.preprocessing import StandardScaler
std_scale = StandardScaler()
```

```
std_df=std_scale.fit_transform(numeric_data)
```

```
fnl_df=pd.DataFrame(std_df,columns=list)
print("no. of outliers in age column",fnl_df[(fnl_df['age']>3) | (fnl_df['age']<-3)]['age'].count())
```

```
no. of outliers in age column 0
```

```
print("no. of outliers in bmi column",fnl_df[(fnl_df['bmi']>3) | (fnl_df['bmi']<-3)]['bmi'].count())
```

```
no. of outliers in bmi column 4
```

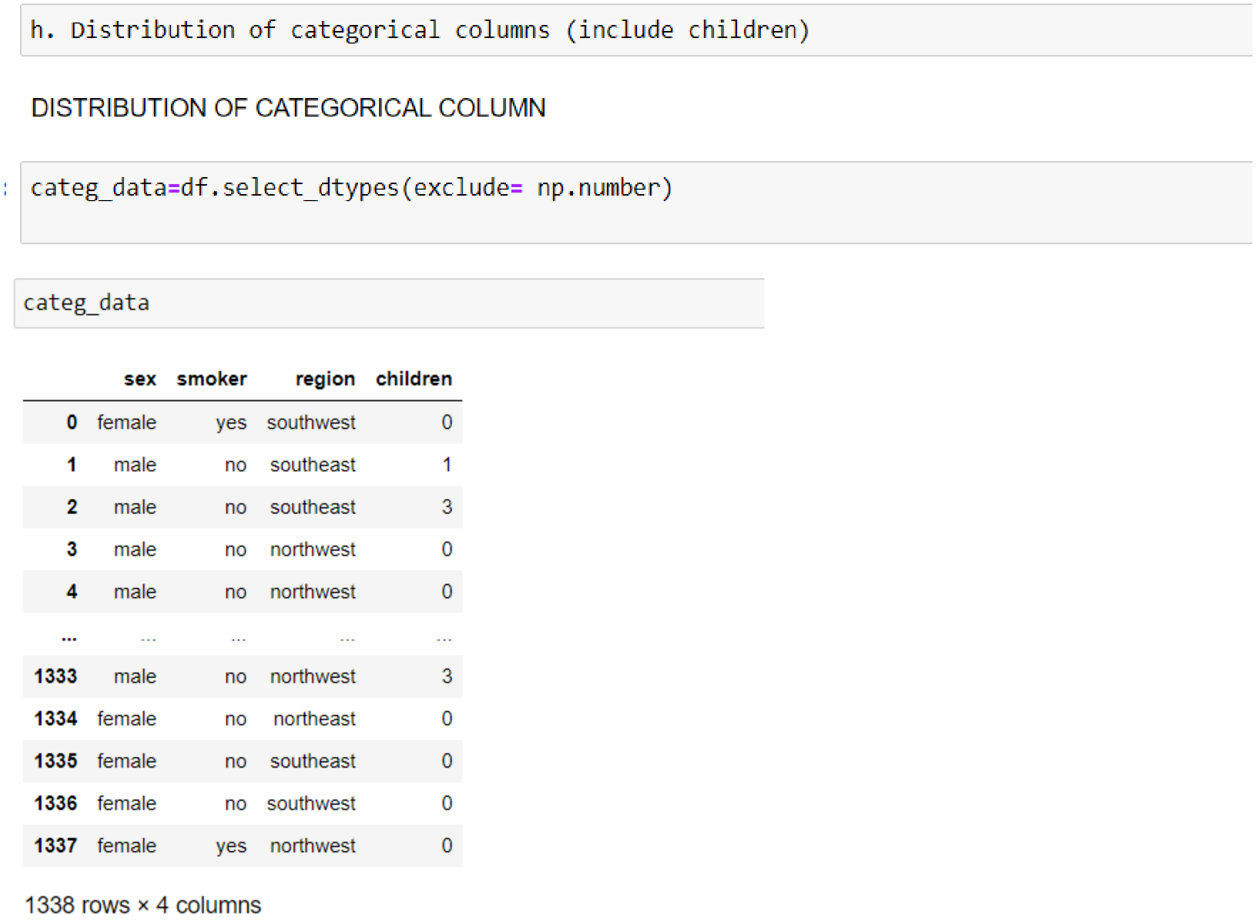
```
print("no. of outliers in charges column",fnl_df[(fnl_df['charges']>3) | (fnl_df['charges']<-3)]['charges'].count())
```

```
no. of outliers in charges column 7
```

**The data points above 3 standard deviations are considered as outliers.
Another method is using interquartile range.**

BUSINESS REPORT - HEALTHCARE INSURANCE

h. Distribution of categorical columns (include children)



```
h. Distribution of categorical columns (include children)
```

DISTRIBUTION OF CATEGORICAL COLUMN

```
categ_data=df.select_dtypes(exclude= np.number)
```

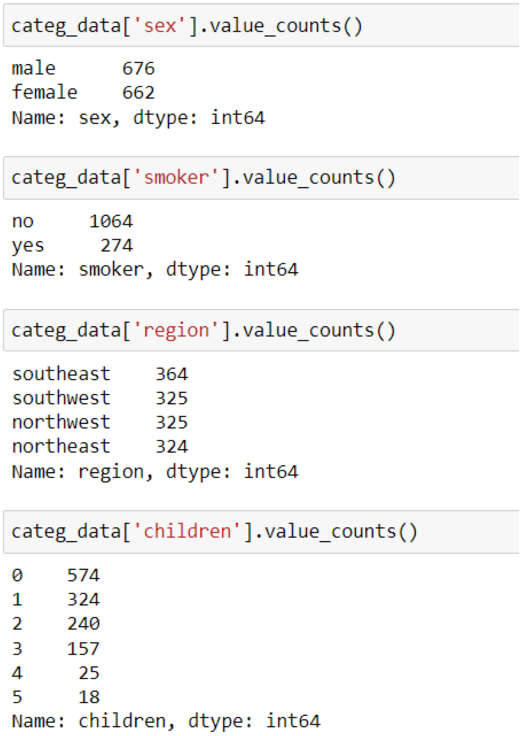
```
categ_data
```

	sex	smoker	region	children
0	female	yes	southwest	0
1	male	no	southeast	1
2	male	no	southeast	3
3	male	no	northwest	0
4	male	no	northwest	0
...	...	...	...	...
1333	male	no	northwest	3
1334	female	no	northeast	0
1335	female	no	southeast	0
1336	female	no	southwest	0
1337	female	yes	northwest	0

1338 rows × 4 columns

Individual values count is calculated for each column to find its distribution because 5 points doesn't make significance for categories.

BUSINESS REPORT - HEALTHCARE INSURANCE



```
categ_data['sex'].value_counts()
```

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

```
categ_data['smoker'].value_counts()
```

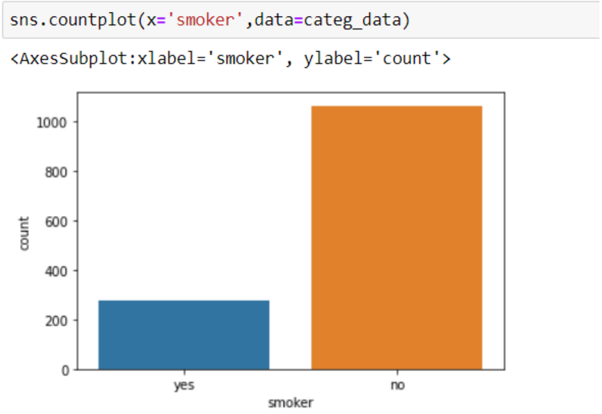
```
no     1064
yes     274
Name: smoker, dtype: int64
```

```
categ_data['region'].value_counts()
```

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

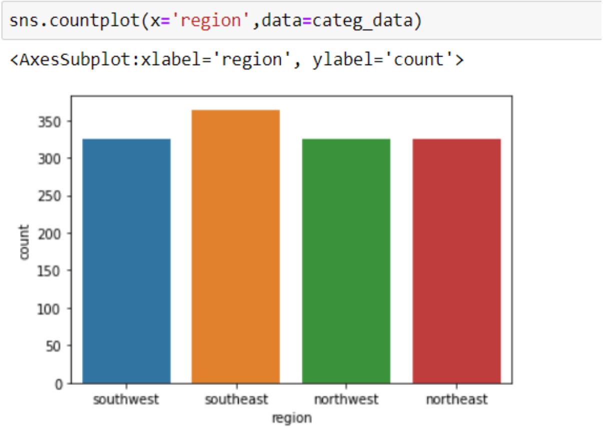
```
categ_data['children'].value_counts()
```

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



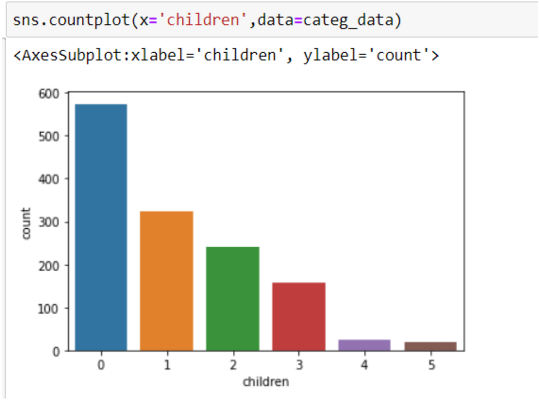
```
sns.countplot(x='smoker',data=categ_data)
```

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



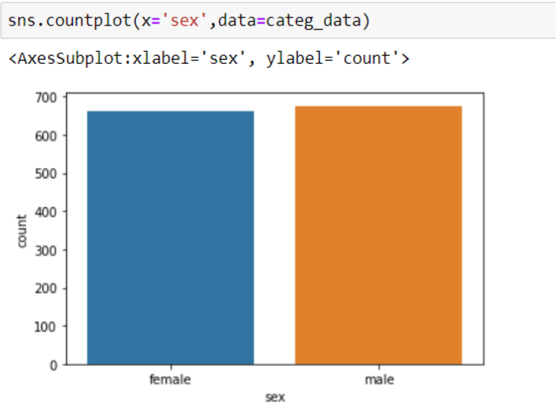
```
sns.countplot(x='region',data=categ_data)
```

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



```
sns.countplot(x='children',data=categ_data)
```

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

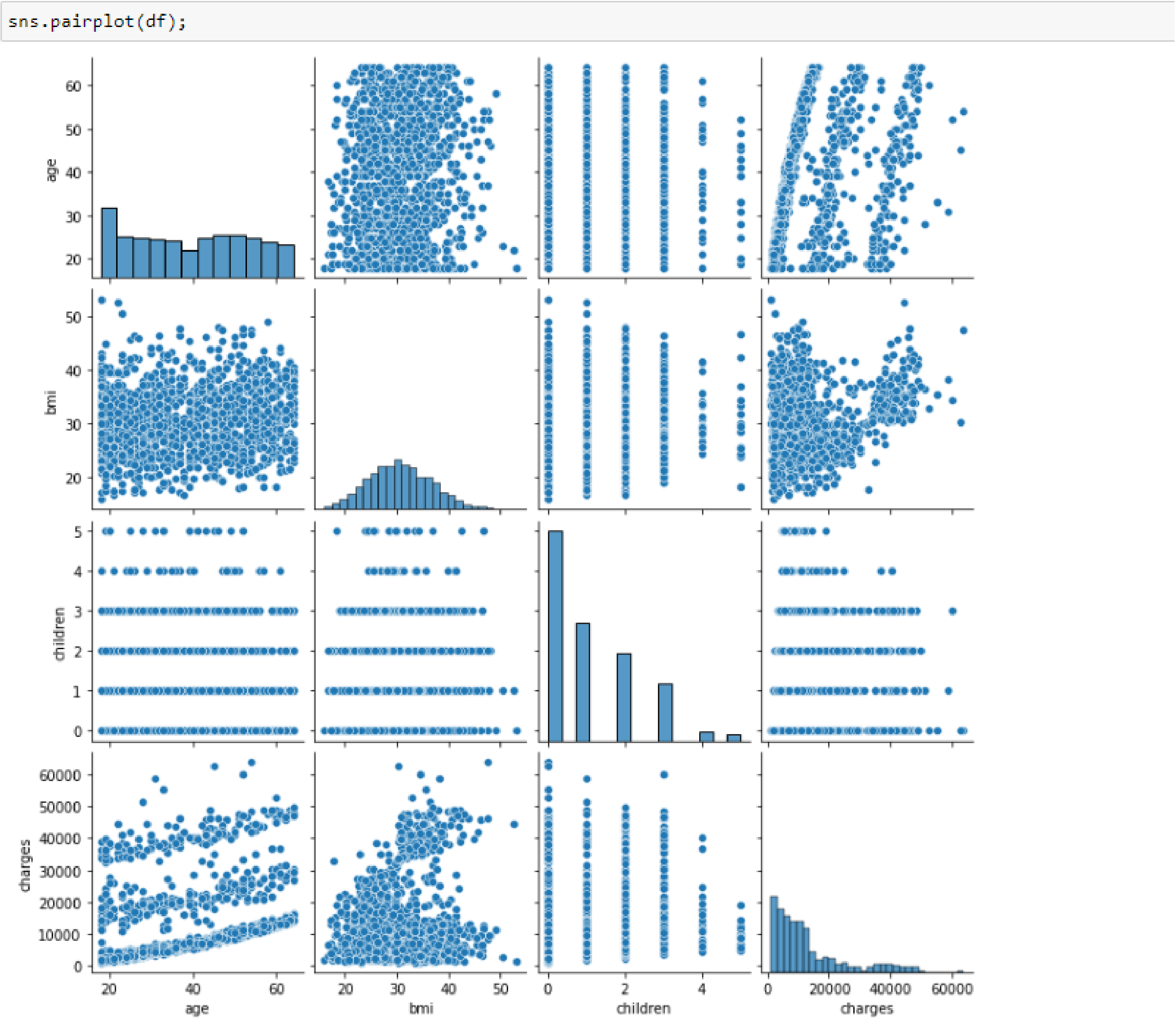


```
sns.countplot(x='sex',data=categ_data)
```

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

BUSINESS REPORT - HEALTHCARE INSURANCE

i. Pair plot that includes all the columns of the data frame



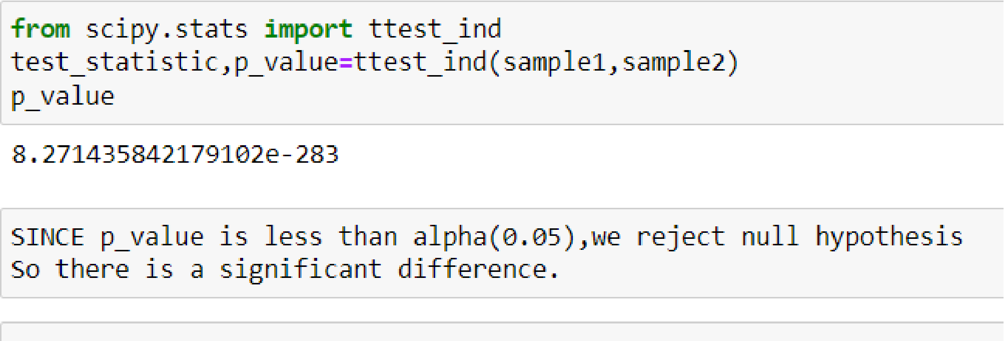
```
sns.pairplot(df);
```

BUSINESS REPORT - HEALTHCARE INSURANCE

4. Answer the following questions with statistical evidence

a) Do charges of people who smoke differ significantly from the people who don't?

T-test for independent events is used



```
from scipy.stats import ttest_ind
test_statistic,p_value=ttest_ind(sample1,sample2)
p_value
```

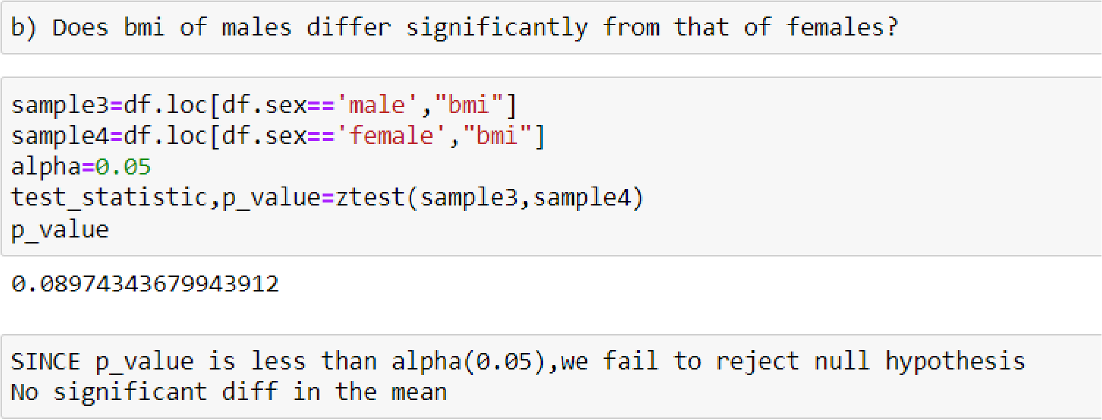
```
8.271435842179102e-283
```

SINCE p_value is less than alpha(0.05),we reject null hypothesis
So there is a significant difference.

Since the p_value is less than the significant value alpha(0.05) we reject the null hypothesis.

There is a significant difference in the charges for those who smoke and dont.

b) Does the BMI of males differ significantly from that of females?



b) Does bmi of males differ significantly from that of females?

```
sample3=df.loc[df.sex=='male',"bmi"]
sample4=df.loc[df.sex=='female',"bmi"]
alpha=0.05
test_statistic,p_value=ztest(sample3,sample4)
p_value
```

```
0.08974343679943912
```

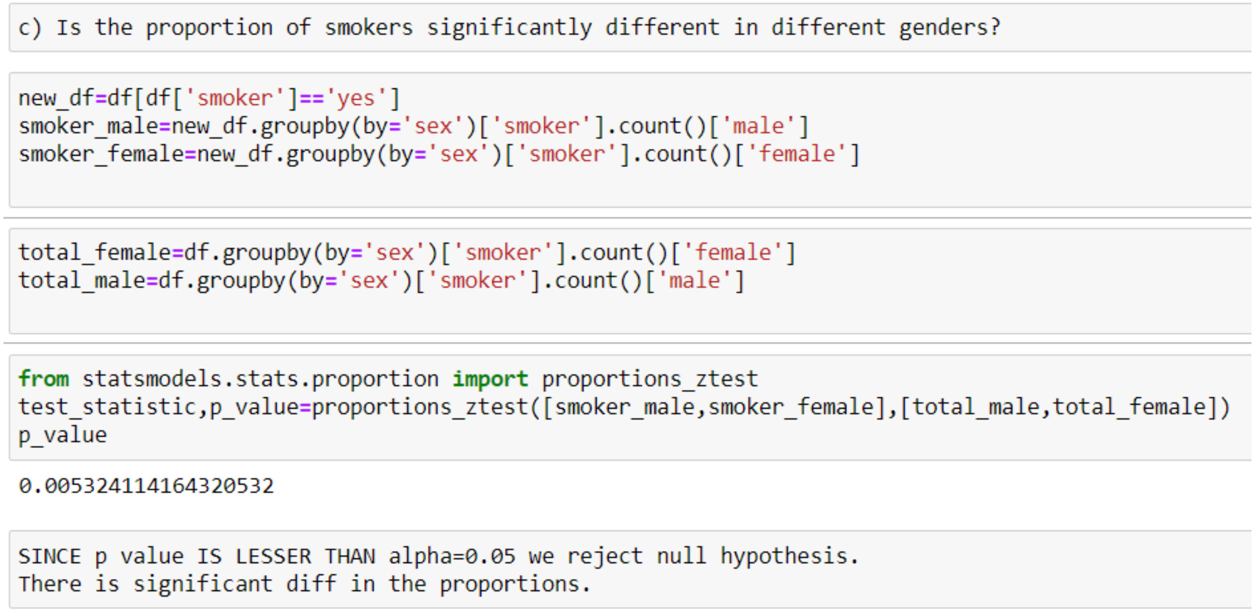
SINCE p_value is less than alpha(0.05),we fail to reject null hypothesis
No significant diff in the mean

Z-Test is done and the p_value is greater than alpha so we fail to reject the null hypothesis.

There is no significant difference in the bmi values on sex category.

BUSINESS REPORT - HEALTHCARE INSURANCE

c) Is the proportion of smokers significantly different in different genders?



```
c) Is the proportion of smokers significantly different in different genders?
```

```
new_df=df[df['smoker']=='yes']
smoker_male=new_df.groupby(by='sex')['smoker'].count()['male']
smoker_female=new_df.groupby(by='sex')['smoker'].count()['female']
```

```
total_female=df.groupby(by='sex')['smoker'].count()['female']
total_male=df.groupby(by='sex')['smoker'].count()['male']
```

```
from statsmodels.stats.proportion import proportions_ztest
test_statistic,p_value=proportions_ztest([smoker_male,smoker_female],[total_male,total_female])
p_value
```

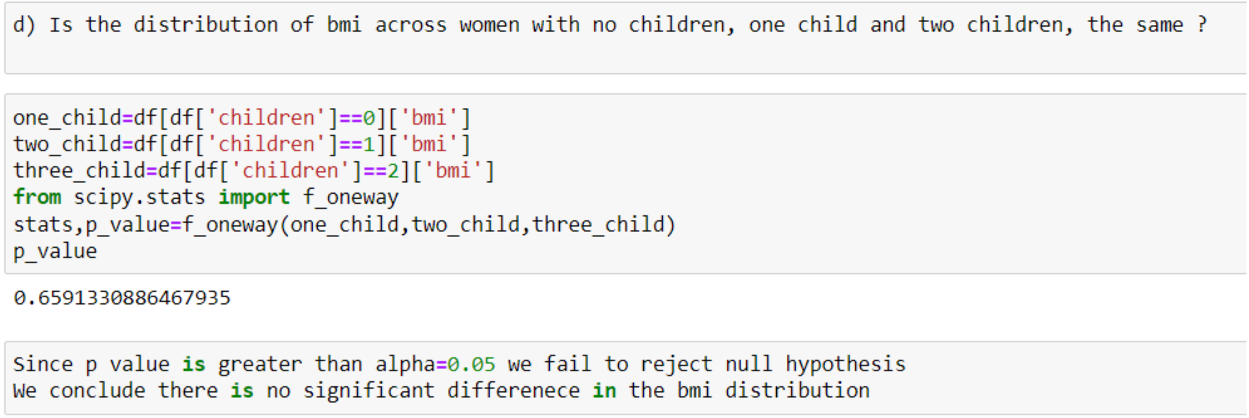
```
0.005324114164320532
```

```
SINCE p value IS LESSER THAN alpha=0.05 we reject null hypothesis.
There is significant diff in the proportions.
```

Since the p_value is less than alpha we reject the null hypothesis so the difference in the proportions are significant.

d) Is the distribution of bmi across women with no children, one child and two children, the same?

Since the sample is more than 2 ANOVA test is conducted



```
d) Is the distribution of bmi across women with no children, one child and two children, the same ?
```

```
one_child=df[df['children']==0]['bmi']
two_child=df[df['children']==1]['bmi']
three_child=df[df['children']==2]['bmi']
from scipy.stats import f_oneway
stats,p_value=f_oneway(one_child,two_child,three_child)
p_value
```

```
0.6591330886467935
```

```
Since p value is greater than alpha=0.05 we fail to reject null hypothesis
We conclude there is no significant differenece in the bmi distribution
```

Since p value is greater than alpha we fail to reject the null hypothesis.
So there is no difference in the bmi.