In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

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

In [3]:

```
data.head()
```

Out[3]:

	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

In [4]:

```
data.shape
```

Out[4]:

```
(1338, 7)
```

In [5]:

```
data.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
```

In [6]:

```
data.isnull().sum()
```

Out[6]:

```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

In [7]:

```
data.describe()
```

Out[7]:

	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

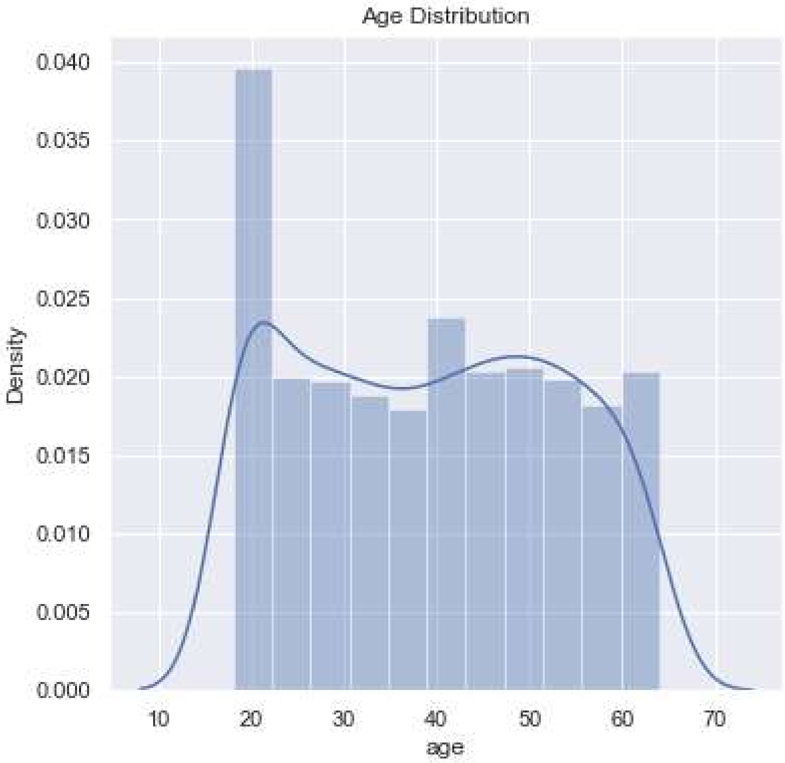
In [8]:

```
import seaborn as sns
import matplotlib.pyplot as plt
```

In [9]:

```
sns.set()
plt.figure(figsize=(6,6))
sns.distplot(data['age'])
plt.title('Age Distribution')
plt.show()
```

```
C:\Users\CHANDU DHAGE\anaconda3\lib\site-packages\seaborn\distributions.py:2557: F
utureWarning: `distplot` is a deprecated function and will be removed in a future
version. Please adapt your code to use either `displot` (a figure-level function w
ith similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

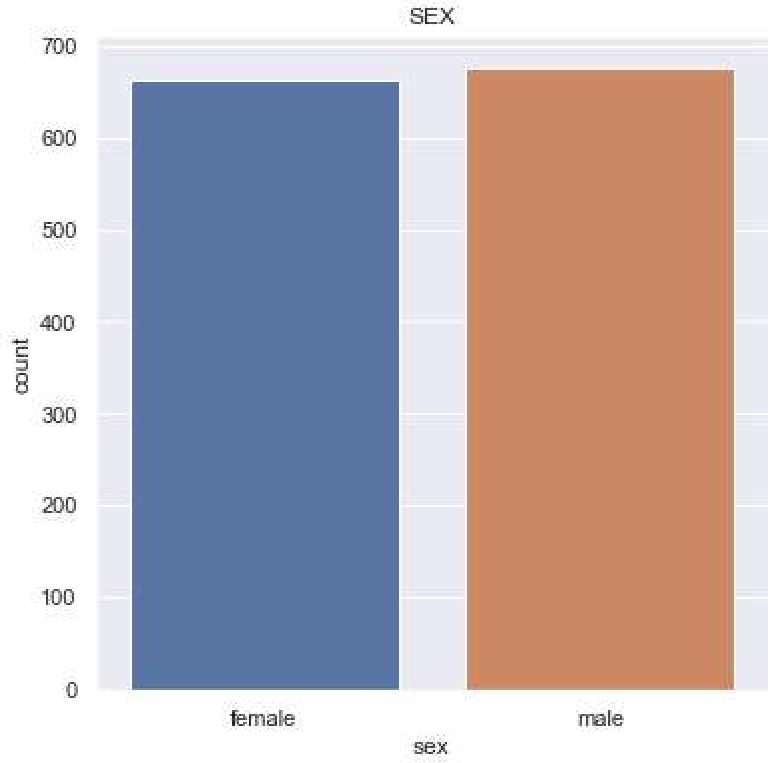


In [25]:

```
sns.set()
plt.figure(figsize =(6,6))
sns.countplot(x = 'sex', data = data)
plt.title('SEX')
plt.show
```

Out[25]:

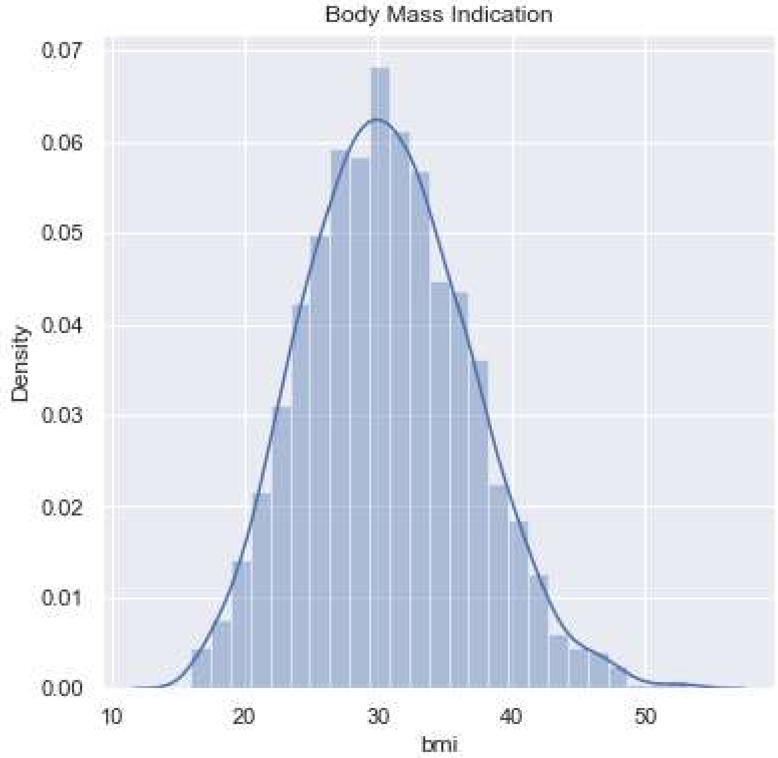
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [27]:

```
sns.set()
plt.figure(figsize = (6,6))
sns.distplot(data['bmi'])
plt.title('Body Mass Indication')
plt.show()
```

```
C:\Users\CHANDU DHAGE\anaconda3\lib\site-packages\seaborn\distributions.py:2557: F
utureWarning: `distplot` is a deprecated function and will be removed in a future
version. Please adapt your code to use either `displot` (a figure-level function w
ith similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

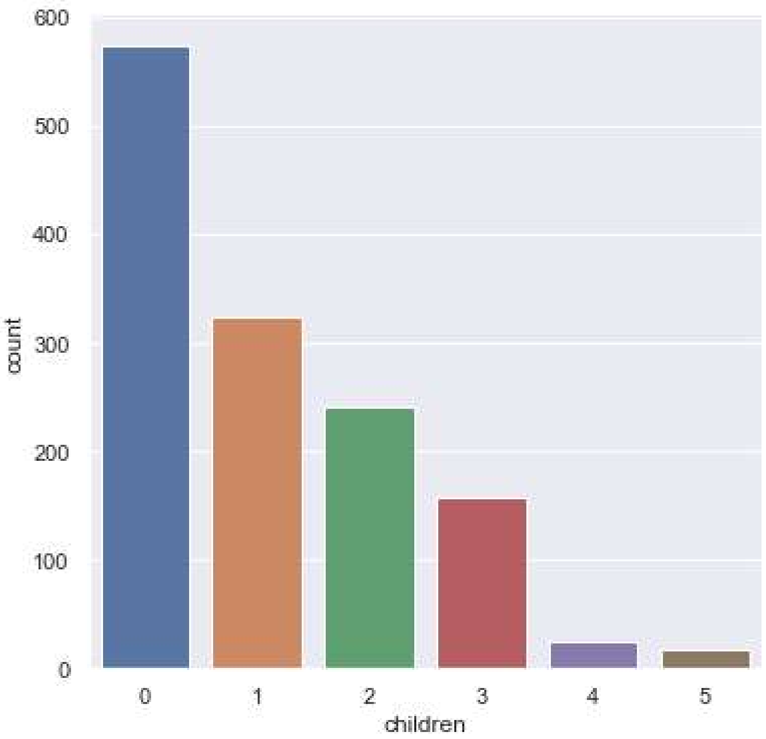


In [30]:

```
sns.set()
plt.figure(figsize=(6,6))
sns.countplot(x = 'children', data = data)
```

Out[30]:

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



In [36]:

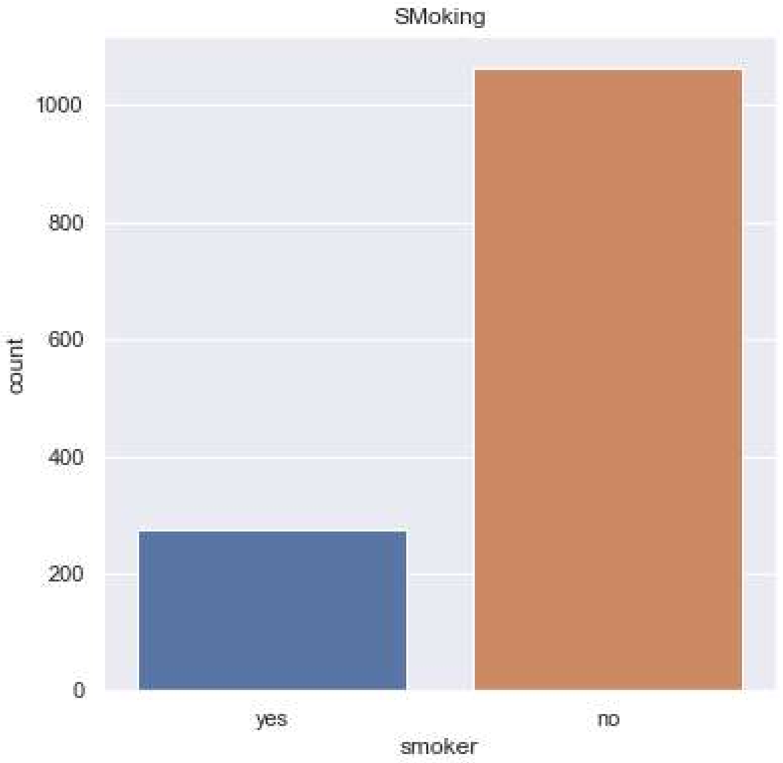
```
data['children'].value_counts()
```

Out[36]:

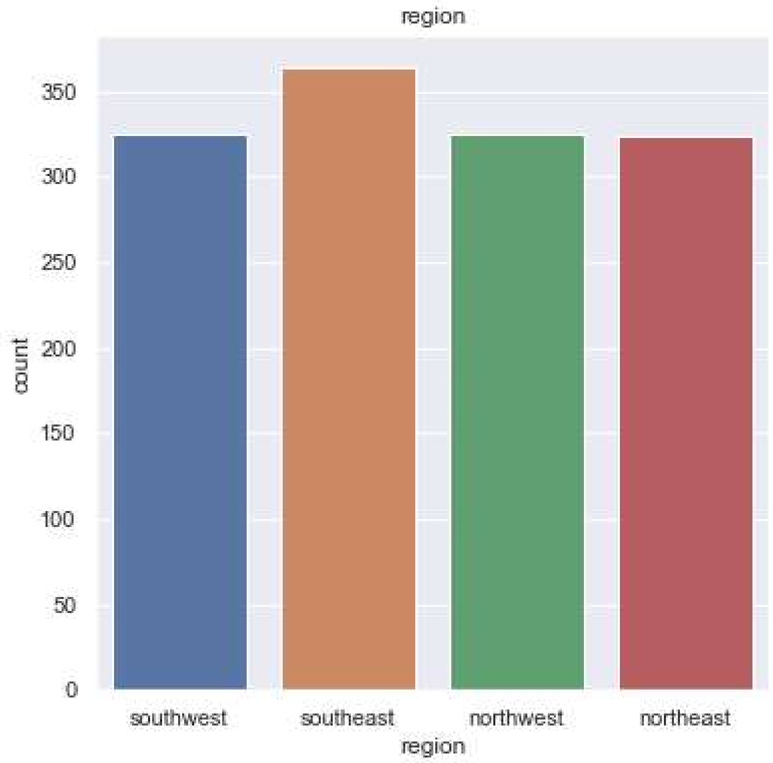
```
0    574
1    324
2    240
3    157
4     25
5     18
Name: children, dtype: int64
```

In [40]:

```
sns.set()
plt.figure (figsize =(6,6))
sns.countplot(x = 'smoker', data = data)
plt.title('SMoking')
plt.show()
```

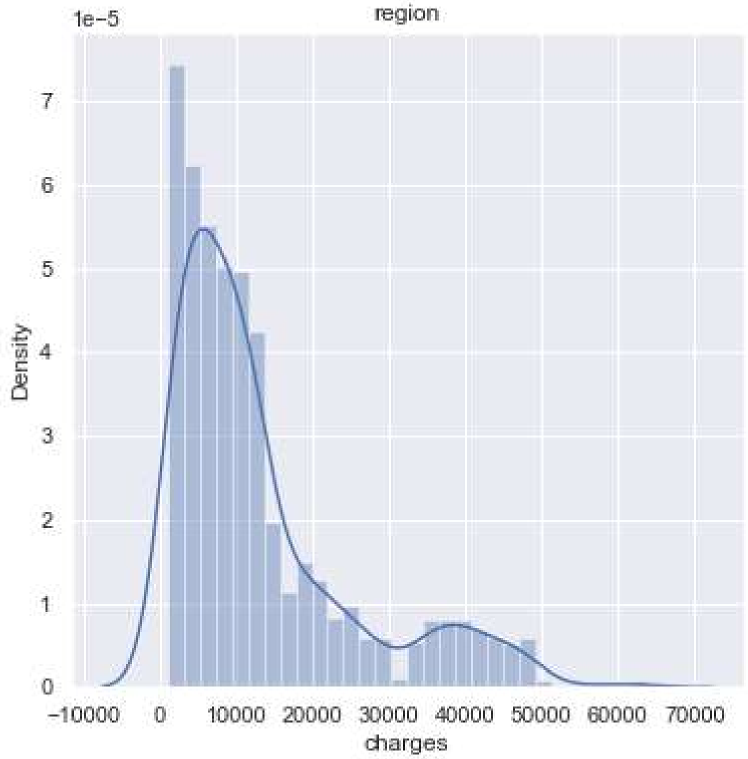


```
In [42]: sns.set()
         plt.figure (figsize =(6,6))
         sns.countplot(x = 'region', data = data)
         plt.title('region')
         plt.show()
```



```
In [44]: sns.set()
         plt.figure (figsize =(6,6))
         sns.distplot(data['charges'])
         plt.title('region')
         plt.show()
```

```
C:\Users\CHANDU DHAGE\anaconda3\lib\site-packages\seaborn\distributions.py:2557: F
utureWarning: `distplot` is a deprecated function and will be removed in a future
version. Please adapt your code to use either `displot` (a figure-level function w
ith similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [50]:
```
data.replace({'sex' : {'male': 0,'female': 1}}, inplace =True)
```

In [52]:
```
data.replace({'smoker' :{'yes':0, 'no':1}}, inplace =True)
```

In [53]:
```
data['region'].unique()
```

Out[53]:
```
array(['southwest', 'southeast', 'northwest', 'northeast'], dtype=object)
```

In [57]:
```
data.replace({'region':{'southwest' : 1, 'southeast':0, 'northeast':2, 'northwest'
```

In [59]:
```
data.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   int64  
 2   bmi       1338 non-null   float64
 3   children  1338 non-null   int64  
 4   smoker    1338 non-null   int64  
 5   region    1338 non-null   int64  
 6   charges   1338 non-null   float64
dtypes: float64(2), int64(5)
memory usage: 73.3 KB
```

In [61]:
```
x = data.drop(['charges'], axis =1)
```

In [ ]:
```
y = data['charges']
```

In [75]:
```
from sklearn.model_selection import train_test_split
```

In [80]:
```
x_train,x_test,y_train,y_test = train_test_split(x, y, random_state = 2, test_size
```

In [81]:
```
print(x_train.shape, x_test.shape, y_train.shape, y_test.shape)
```
```
(1070, 6) (268, 6) (1070,) (268,)
```

In [83]:
```
from sklearn.linear_model import LinearRegression
```

In [85]:
```
model = LinearRegression()
```

In [88]:
```
model.fit(x_train, y_train)
```
Out[88]:
```
LinearRegression()
```

In [91]:
```
predict = model.predict(x_train)
```

In [93]:
```
predict
```
Out[93]:
```
array([  478.49404197,  9317.75369733, 13193.79859142, ...,
       17327.55442479,  9600.51860822, 13753.18970971])
```

In [98]:
```
from sklearn import metrics
from sklearn.metrics import accuracy_score
```

In [102…
```
metrics.r2_score(y_train, predict)
```
Out[102]:
```
0.751505643411174
```

In [104…
```
predict2 = model.predict(x_test)
```

In [106…
```
metrics.r2_score(y_test, predict2)
```
Out[106]:
```
0.7447273869684077
```

In [112…
```
input = (31,1,25,74,0,1)

inputtonumpy = np.asarray(input)
```

In [113…
```
inputtoreshape = inputtonumpy.reshape(1,-1)
```

In [115…
```
new = model.predict(inputtoreshape)
```

In [116…
```
new
```
Out[116]:
```
array([70596.0307385])
```