In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
```

In [2]:

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

Out[2]:

```
      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
```

In [3]:

```
df.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]:

```
df.shape
```

Out[4]:

```
(1338, 7)
```

In [6]:

```
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
```

In [7]:

```
df.isnull().sum()
```

Out[7]:

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

In [9]:

```
df.describe()
```

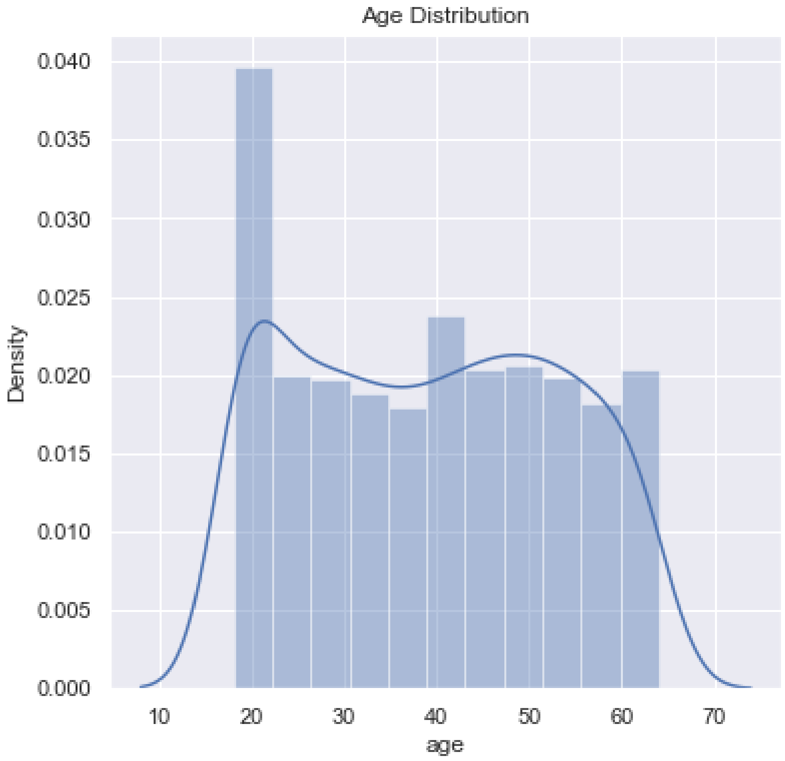
Out[9]:

| | 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 [11]:

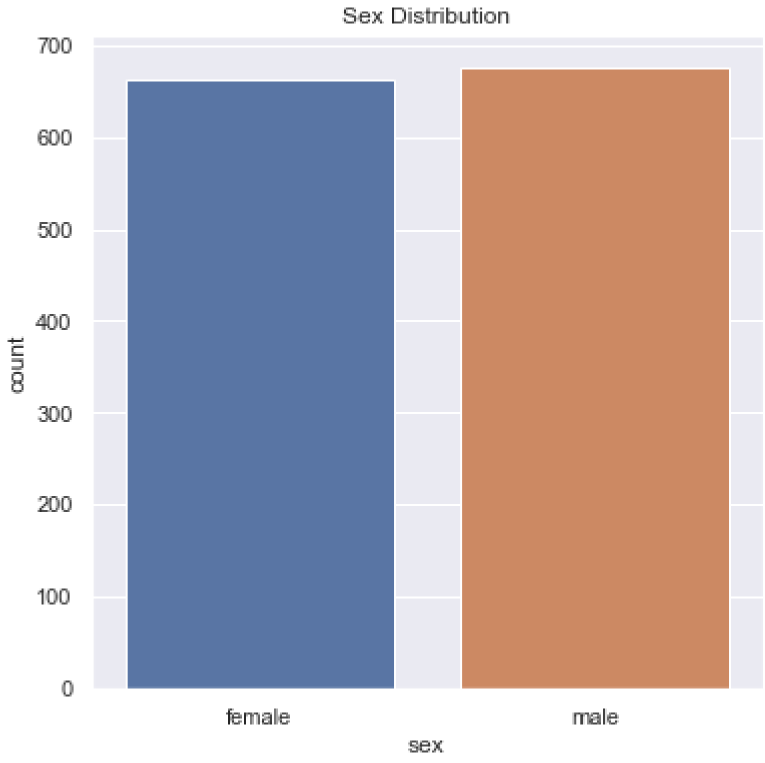
```
sns.set()
plt.figure(figsize=(6,6))
sns.distplot(df['age'])
plt.title('Age Distribution')
plt.show()
```

```
D:\Anaconda\lib\site-packages\seaborn\distributions.py:2619: FutureWarning:
`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 with
similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [12]:

```
plt.figure(figsize=(6,6))
sns.countplot(x='sex', data=df)
plt.title('Sex Distribution')
plt.show()
```



In [13]:

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

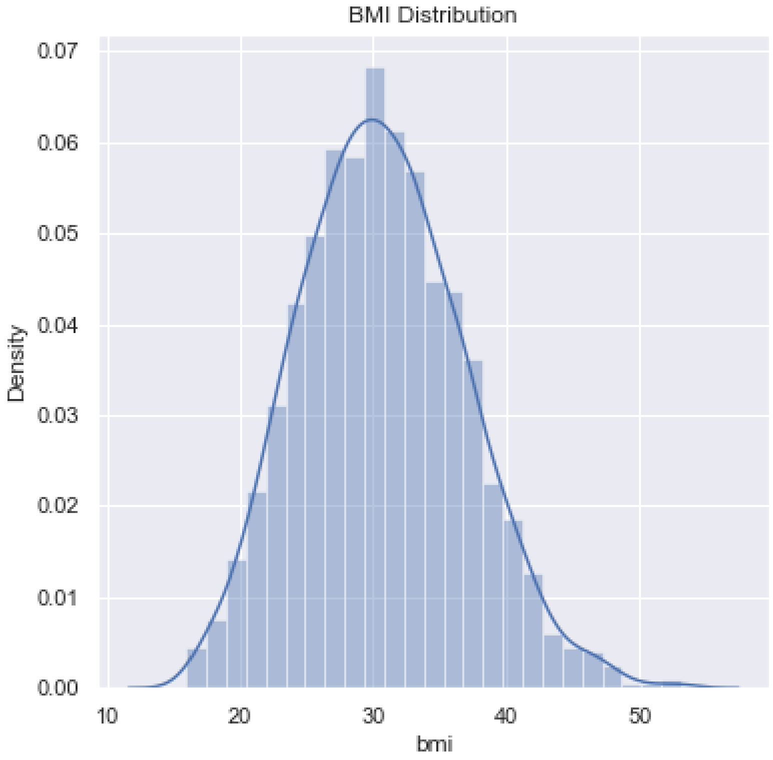
Out[13]:

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

In [14]:

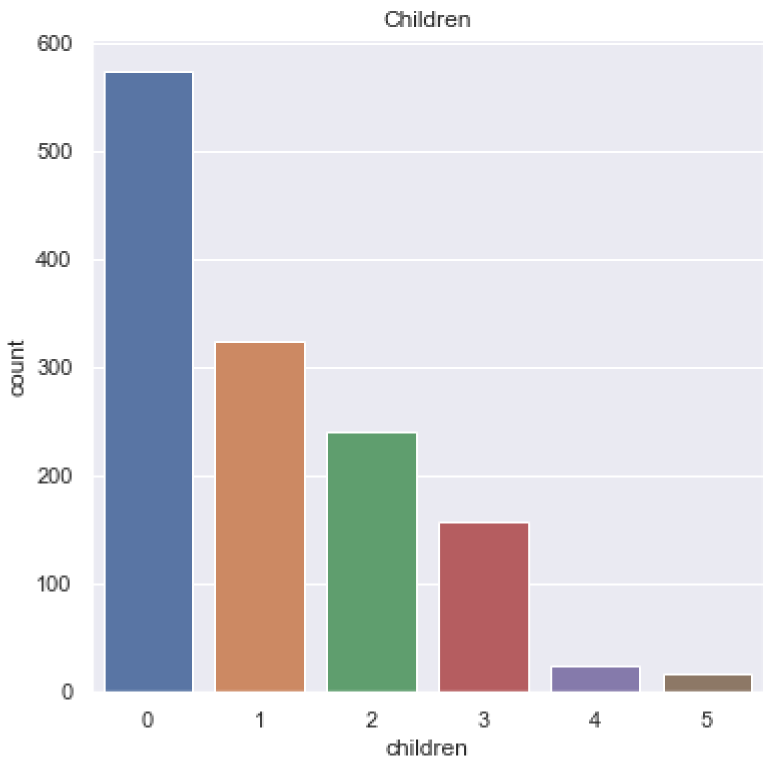
```
plt.figure(figsize=(6,6))
sns.distplot(df['bmi'])
plt.title('BMI Distribution')
plt.show()
```

```
D:\Anaconda\lib\site-packages\seaborn\distributions.py:2619: FutureWarning:
`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 with
similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [16]:

```
plt.figure(figsize=(6,6))
sns.countplot(x='children', data=df)
plt.title('Children')
plt.show()
```



In [17]:

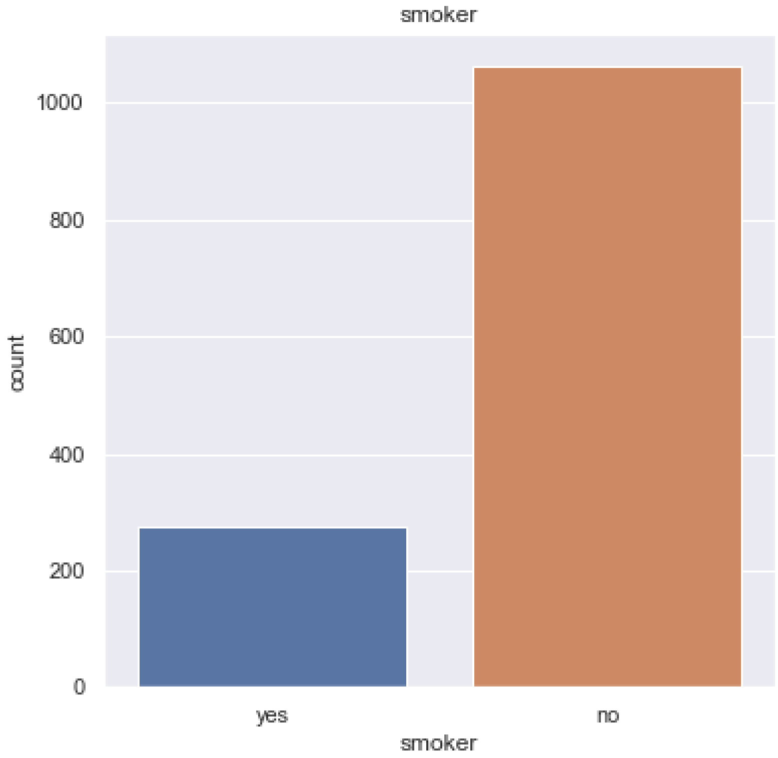
```
df['children'].value_counts()
```

Out[17]:

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

In [18]:

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



In [19]:

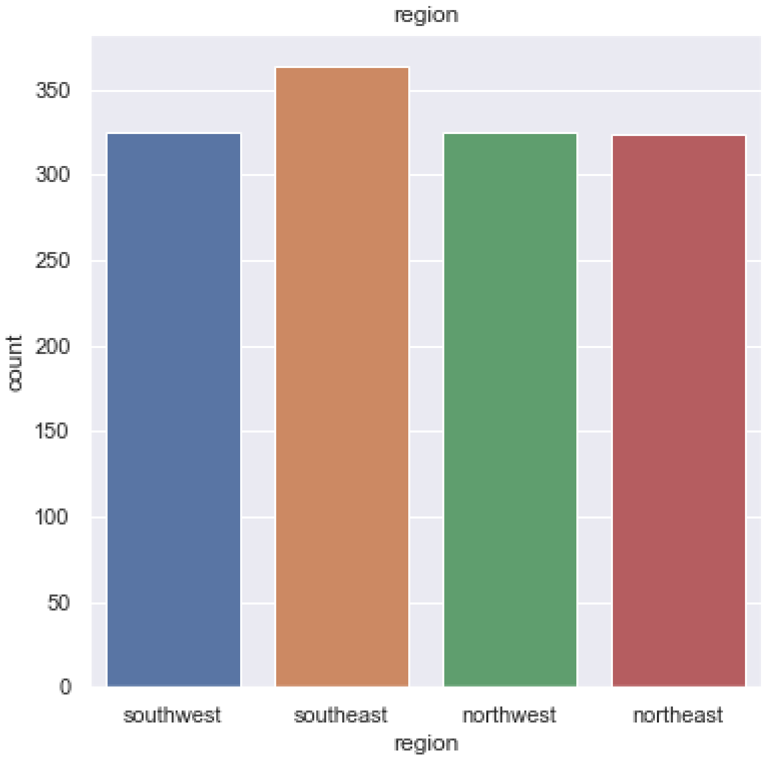
```
df['smoker'].value_counts()
```

Out[19]:

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

In [20]:

```
plt.figure(figsize=(6,6))
sns.countplot(x='region', data=df)
plt.title('region')
plt.show()
```



In [21]:

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

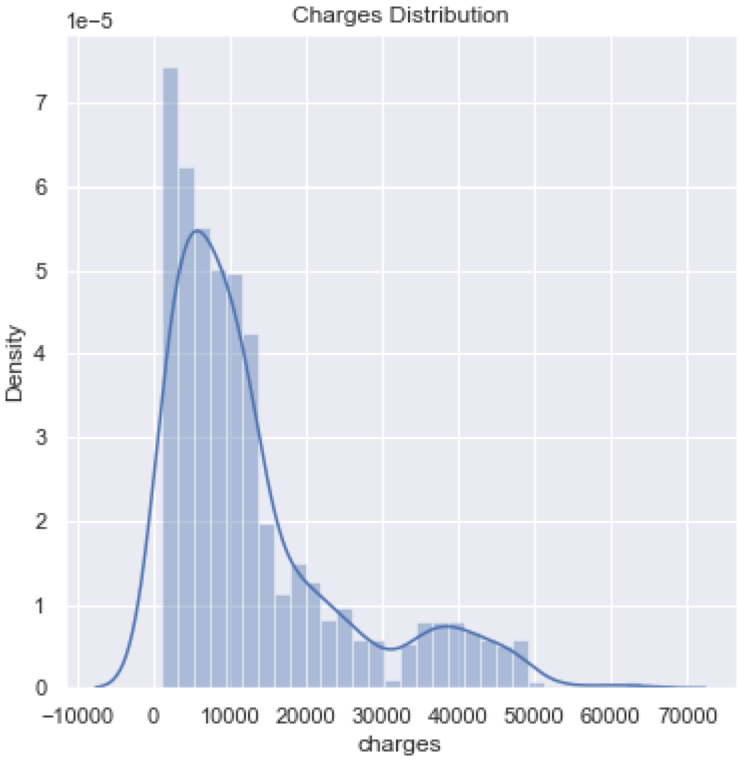
Out[21]:

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

In [22]:

```
plt.figure(figsize=(6,6))
sns.distplot(df['charges'])
plt.title('Charges Distribution')
plt.show()
```

```
D:\Anaconda\lib\site-packages\seaborn\distributions.py:2619: FutureWarning:
`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 with
similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [24]:

```
# encoding sex column
df.replace({'sex':{'male':0,'female':1}}, inplace=True)

3 # encoding 'smoker' column
df.replace({'smoker':{'yes':0,'no':1}}, inplace=True)

# encoding 'region' column
df.replace({'region':{'southeast':0,'southwest':1,'northeast':2,'northwest':3}}, inplace=Tr
```

In [25]:

```
X = df.drop(columns='charges', axis=1)
Y = df['charges']
```

In [27]:

```
print(X)
```

```
      age  sex     bmi  children  smoker  region
0      19    1  27.900         0       0       1
1      18    0  33.770         1       1       0
2      28    0  33.000         3       1       0
3      33    0  22.705         0       1       3
4      32    0  28.880         0       1       3
...   ...  ...     ...       ...     ...     ...
1333   50    0  30.970         3       1       3
1334   18    1  31.920         0       1       2
1335   18    1  36.850         0       1       0
1336   21    1  25.800         0       1       1
1337   61    1  29.070         0       0       3

[1338 rows x 6 columns]
```

In [28]:

```
print(Y)
```

```
0       16884.92400
1        1725.55230
2        4449.46200
3       21984.47061
4        3866.85520
           ...
1333    10600.54830
1334     2205.98080
1335     1629.83350
1336     2007.94500
1337    29141.36030
Name: charges, Length: 1338, dtype: float64
```

In [29]:

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)
```

In [30]:

```
print(X.shape, X_train.shape, X_test.shape)
```

```
(1338, 6) (1070, 6) (268, 6)
```

In [31]:

```
regressor = LinearRegression()
```

In [32]:

```
regressor.fit(X_train, Y_train)
```

Out[32]:

```
LinearRegression()
```

In [33]:

```
training_data_prediction =regressor.predict(X_train)
```

In [34]:

```
r2_train = metrics.r2_score(Y_train, training_data_prediction)
print('R squared vale : ', r2_train)
```

```
R squared vale :  0.751505643411174
```

In [35]:

```
test_data_prediction =regressor.predict(X_test)
```

In [36]:

```
r2_test = metrics.r2_score(Y_test, test_data_prediction)
print('R squared vale : ', r2_test)
```

```
R squared vale :  0.7447273869684077
```

In [37]:

```
input_data = (31,1,25.74,0,1,0)

# changing input_data to a numpy array
input_data_as_numpy_array = np.asarray(input_data)

# reshape the array
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = regressor.predict(input_data_reshaped)
print(prediction)

print('The insurance cost is USD ', prediction[0])
```

```
[3760.0805765]
The insurance cost is USD  3760.080576496046
```

In [ ]:

```

```