Importing the necessary libraries

In [1]:
```
import pandas as pd
import numpy as np
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 [3]:
```
df = pd.read_csv('insurance.csv')
df
```

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
...	...	...	...	...	...	...	...
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 [4]:
```
df.head()
```

Out[4]:

	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

Number of rows and columns

In [6]:
```
df.shape
```

Out[6]:
```
(1338, 7)
```

Getting some information about the dataset

In [7]:
```
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
```

Categorical Feature:

Sex

Smoker

Region

Check for missing values

In [8]:
```
df.isnull().sum()
```

Out[8]:
```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

Data Analysis

In [10]:
```
df.describe()
```

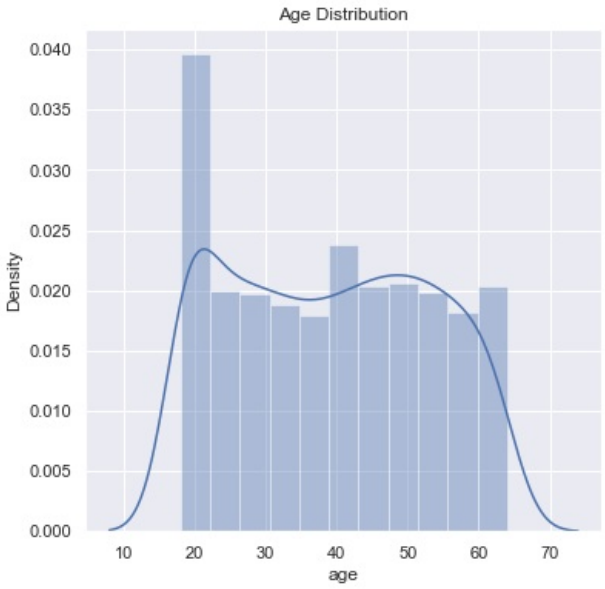
Out[10]:

	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

Distribution of age value

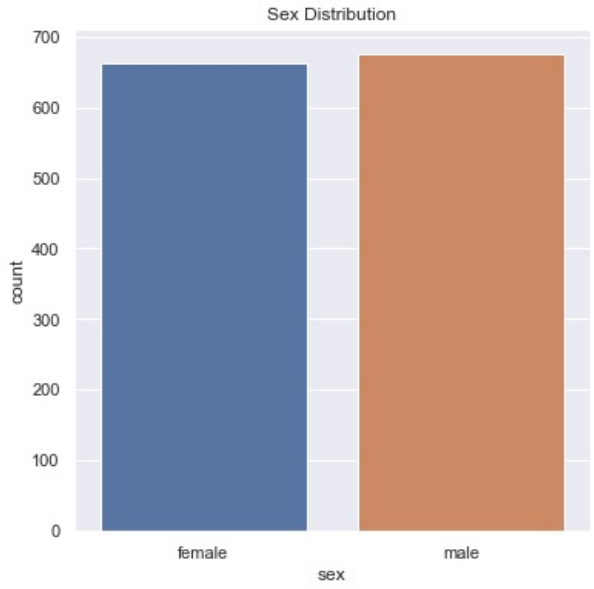
In [9]:
```
sns.set()
plt.figure(figsize = (6,6))
sns.distplot(df['age'])
plt.title('Age Distribution')
plt.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprec
ated 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)
```



Gender Column

In [13]:
```
plt.figure(figsize=(6,6))
sns.countplot(x = 'sex',data = df)
plt.title('Sex Distribution')
plt.show()
```



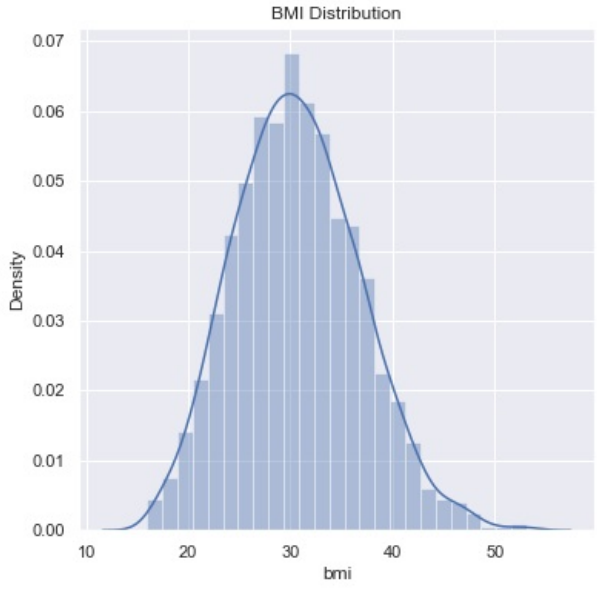
In [14]:
```
df ['sex'].value_counts()
```

Out[14]:
```
male      676
female    662
Name: sex, dtype: int64
```

Bmi Distribution

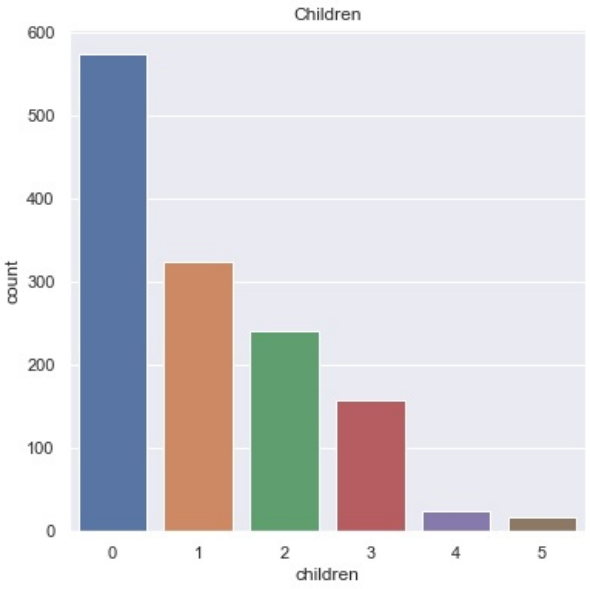
In [16]:
```
plt.figure(figsize=(6,6))
sns.distplot(df['bmi'])
plt.title('BMI Distribution')
plt.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprec
ated 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)
```



Children column

In [17]:
```
plt.figure(figsize = (6,6))
sns.countplot(x = 'children',data = df)
plt.title('Children')
plt.show()
```

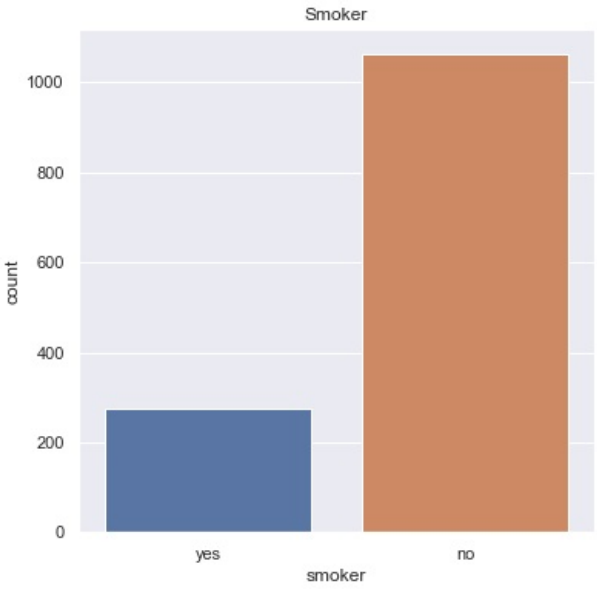


In [18]:
```
df['children'].value_counts()
```

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

Smoker column

In [19]:
```
plt.figure(figsize = (6,6))
sns.countplot(x = 'smoker',data = df)
plt.title('Smoker')
plt.show()
```

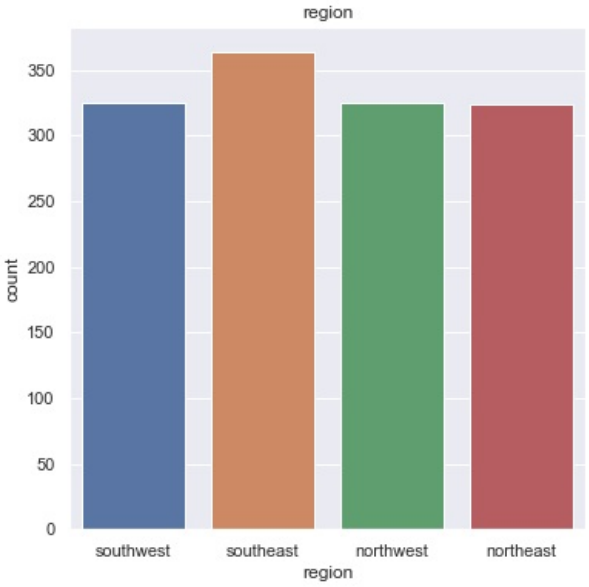


In [20]:
```
df['smoker'].value_counts()
```

Out[20]:
```
no     1064
yes     274
Name: smoker, dtype: int64
```

Region column

In [21]:
```
plt.figure(figsize = (6,6))
sns.countplot(x = 'region',data = df)
plt.title('region')
plt.show()
```



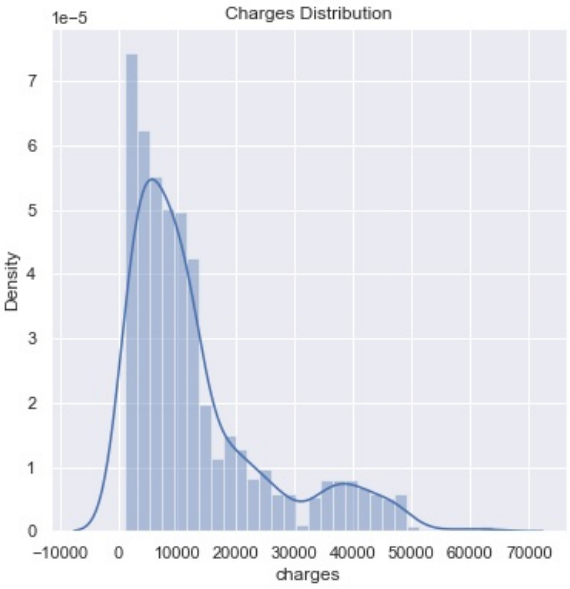
In [22]:
```
df['region'].value_counts()
```

Out[22]:
```
southeast    364
southwest    325
northwest    325
northeast    324
Name: region, dtype: int64
```

Bmi Distribution

In [23]:
```
plt.figure(figsize=(6,6))
sns.distplot(df['charges'])
plt.title('Charges Distribution')
plt.show()
```

```
C:\ProgramData\Anaconda3\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)
```



Encoding the columns

In [26]:
```
# encoding sex column
df.replace({'sex':{'male':0,'female':1}},inplace=True)

# 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 = True)
```

In [27]:
```
df
```

Out[27]:

	age	sex	bmi	children	smoker	region	charges
0	19	1	27.900	0	0	1	16884.92400
1	18	0	33.770	1	1	0	1725.55230
2	28	0	33.000	3	1	0	4449.46200
3	33	0	22.705	0	1	3	21984.47061
4	32	0	28.880	0	1	3	3866.85520
...	...	...	...	...	...	...	...
1333	50	0	30.970	3	1	3	10600.54830
1334	18	1	31.920	0	1	2	2205.98080
1335	18	1	36.850	0	1	0	1629.83350
1336	21	1	25.800	0	1	1	2007.94500
1337	61	1	29.070	0	0	3	29141.36030

1338 rows × 7 columns

splitting the feature and Target

In [30]:
```
X = df.drop(columns= 'charges',axis = 1)
Y = df['charges']
```

In [31]:
```
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 [32]:
```
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 [35]:
```
# splitting the data into Training data & Teating Data
X_train,X_test,Y_train , Y_test = train_test_split(X,Y,test_size = 0.2,random_state =2 )
```

In [36]:
```
print(X.shape,X_train.shape,X_test.shape)
```

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

Linear Regression

In [37]:
```
# Loding the Linear Regression model
regressor = LinearRegression()
```

In [38]:
```
regressor.fit(X_train,Y_train)
```

Out[38]:
```
LinearRegression()
```

Model Evaluation

In [39]:
```
training_data_prediction = regressor.predict(X_train)
```

In [40]:
```
training_data_prediction
```

Out[40]:
```
array([  478.49404197,  9317.75369733, 13193.79859142, ...,
       17327.55442479,  9600.51860822, 13753.18970971])
```

In [41]:
```
# R squared value
r2_train = metrics.r2_score(Y_train,training_data_prediction)
```

In [42]:
```
print('R squared value:',r2_train)
```

```
R squared value: 0.751505643411174
```

prediction on test data

In [43]:
```
test_data_prediction = regressor.predict(X_test)
```

In [44]:
```
test_data_prediction
```

Out[44]:
```
array([ 1520.59242161, 11570.5920178 , 10082.43849883,  2246.21754312,
        7881.28362035, 11081.50227956,  3538.24791808,   698.03224036,
       12223.4851558 ,  9611.93217623, 11657.51046259,  4891.0539656 ,
       29947.50192274,  -370.8384887 , 12401.36048618, 13243.21522903,
        3814.42216541,  7883.39384825, 29431.34485576,  2362.83672121,
       12505.50452609,  2256.75277238, 34468.01948464, 31742.4859866 ,
       30306.19118561,  9027.76110059,  1923.87420399, 15247.09503907,
        6542.61302531,  2104.79910554,  9484.36642532,  5794.91649267,
        4425.26853454,  5015.3811241 ,  9579.4545934 ,  4601.74838962,
       29875.58083252,  6797.04084444, 27239.25811383, 13999.0938259 ,
         313.55184653, 28415.75044713,  7886.54751277,  1478.09056648,
       10273.28966107,  8003.09003405, 11612.15283896,  8175.95966058,
       10753.45200738, 13802.18082647,  5740.90172027,  -737.13333209,
       26346.21771217, 37192.66032995,  7364.09646118, 17845.51752284,
        1412.63748094, 11042.48090545,  2159.33597148, 34066.1609094 ,
       11646.83178834,   874.98548929,  4020.66706965, 35913.0386546 ,
       -1034.71506651, 13963.49470486, 14840.86595147,  3395.11689253,
       12935.74119039, 11199.38639761, 11579.90265947, 16132.93772732,
       10183.88439249,  9888.34374983, 15157.35586536, 12377.94812939,
        4387.77863628,  3680.0942183 ,  5347.06219182, 13291.0174177 ,
        9158.24253865, 11935.82529104,  9522.10094863, 27668.10801212,
       12639.34008179,  3989.82506218, 38550.3600665 , 11191.86138788,
        8088.76475698, 11068.02157864, 10956.54972199, 15139.01708371,
       11077.7652618 , 13045.02707757,  5283.33522041, 25958.0327765 ,
        4962.43983078, 10543.57361001,  2709.95649343, 29007.79585973,
        6350.41196404,  3478.11303549,  2661.5079005 , 15990.91366368,
        7905.79980945, 10304.73937225,  9962.86575973,  5066.24762376,
       14869.35897203, 33752.1676117 ,  3761.88660755, 11521.18346955,
       24631.42819661, 14803.95189475,  1734.60861523, 10401.39588933,
        9202.60416666,  6288.03801508, 11838.14846799, 28871.88920869,
        6579.83915531,  7172.5493248 , 15845.7059381 , 16235.1462466 ,
        8251.21825771, 26323.60251235, 35303.7543364 , 11847.13682432,
        8073.11495528,  9326.25448529,  8467.39129356,  2933.9917805 ,
        3322.8695607 ,  4683.92759642,  8307.29448212,  8002.16943038,
        7053.31134868, 28990.07000293, 35181.28277884,  4167.15930146,
       27886.14685479,  4144.07006286,  6628.26922773, 13311.51217138,
        8025.49599525, 36451.54381063, 11784.84114664, 11347.89349827,
        8294.89578165,   524.38645586,  6503.27709943,  7165.34947975,
        4638.1194905 , 11666.09138657, 11630.93778466, 15478.52566732,
        5856.27738941, 27679.01778802,  1979.26736391, 11476.47168147,
       16974.37864533, 13934.2661456 ,  9520.8147517 ,  2269.28578271,
        4396.04458266,  8922.70311363, 19309.54145116, 28276.8594048 ,
       12676.31036501,  2965.72503913, 32305.95532934, 13107.14725741,
       32778.03744536, 34349.43983065, 11161.90211021,  7576.16565725,
        2633.64298278,  2362.83672121, 11656.06768299,  7884.51285855,
        2926.10661155,  1166.95403524, 31658.17342743,  7134.58660758,
        5557.65095352, 27325.26552208,  6609.80947788,  2654.92453849,
        7915.90908586, 35382.85588438,  7986.35556548,  4319.94677933,
        9477.98125702, 26872.46549002,  5713.52005266, 40198.16671135,
       37499.39947482, 12998.97434383, 26841.49272812, 11921.07008303,
       37470.06851291,  7403.67284293,  4214.20198795,  1961.81400965,
       14048.97433527, 14018.66010565,  2180.00417375, 35697.72795561,
       12791.22900693,  8748.61933066,  1132.66189998, 30647.68798314,
        3495.69714418,  3469.35222538, 12600.42201939, 15082.03691758,
       29668.01412306,   -90.72967482,  3183.27545559,  8454.89054624,
       39754.78580876,  7972.36417173, 35120.73194872, 27504.76077554,
       13731.00485102, 28889.95796905, 16499.4845035 ,  7606.95831393,
       16113.44475909,  7121.06385743, 10218.00060066,  3711.06528332,
        8798.98783422,  1921.16940112, 32853.72073048, 32064.68779053,
       14808.74127134, 11403.04577031,  1017.44053899,  6282.0954554 ,
       11157.27362218,  4173.88588937, 10981.18496951,  1055.76715878,
       34880.38727916, 32433.90662952, 10508.02880569, 26355.22189142,
       12805.78624032,  1722.14283127, 11198.49344957,  2425.6595318 ,
        7497.57207675, 10638.4733706 , 17092.44095263,  5779.81964596,
       10521.06603397,  -521.71832066, -2343.57982801,  1908.03764045,
       27783.94876666, 33997.01915615, 37349.83789264, -1669.89064998,
       15722.41952204, 36968.05564506, 12987.36484768, 34174.92279327])
```

In [46]:
```
# R squared value
r2_test = metrics.r2_score(Y_test,test_data_prediction)
```

In [47]:
```
print('R squared value:',r2_test)
```

```
R squared value: 0.7447273869684077
```

In [50]:
```
# Buliding a Predictive System
input_data = (31,1,25.74,0,1,0)

#changing input data to a numpy arrays
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)
```

```
[3760.0805765]
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature name
s, but LinearRegression was fitted with feature names
  warnings.warn(
```

In [51]:
```
print('The insurance cost is USD',prediction[0])
```

```
The insurance cost is USD 3760.080576496046
```

In [ ]: