```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.m
```

Importing the Dependencis

```
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
```

Data collection and analysis

```
# load the data from csv file to a panda Dataframe
insurance_dataset = pd.read_csv('/content/Medical_insurance.csv')
```

```
#first five row of the dataframe
insurance_dataset.head()
```

	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

Next steps: Generate code with insurance_dataset | View recommended plots | New interactive sheet

```
# number of rows and columns
insurance_dataset.shape
```

```
(2772, 7)
```

```
#getting some information about dataset
insurance_dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2772 entries, 0 to 2771
```

```
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       2772 non-null   int64
 1   sex       2772 non-null   object
 2   bmi       2772 non-null   float64
 3   children  2772 non-null   int64
 4   smoker    2772 non-null   object
 5   region    2772 non-null   object
 6   charges   2772 non-null   float64
dtypes: float64(2), int64(2), object(3)
memory usage: 151.7+ KB
```

Categorial Features: -Sex -Smoker -Region

```
#checking for missing values
insurance_dataset.isnull().sum()
```

	0
age	0
sex	0
bmi	0
children	0
smoker	0
region	0
charges	0

dtype: int64

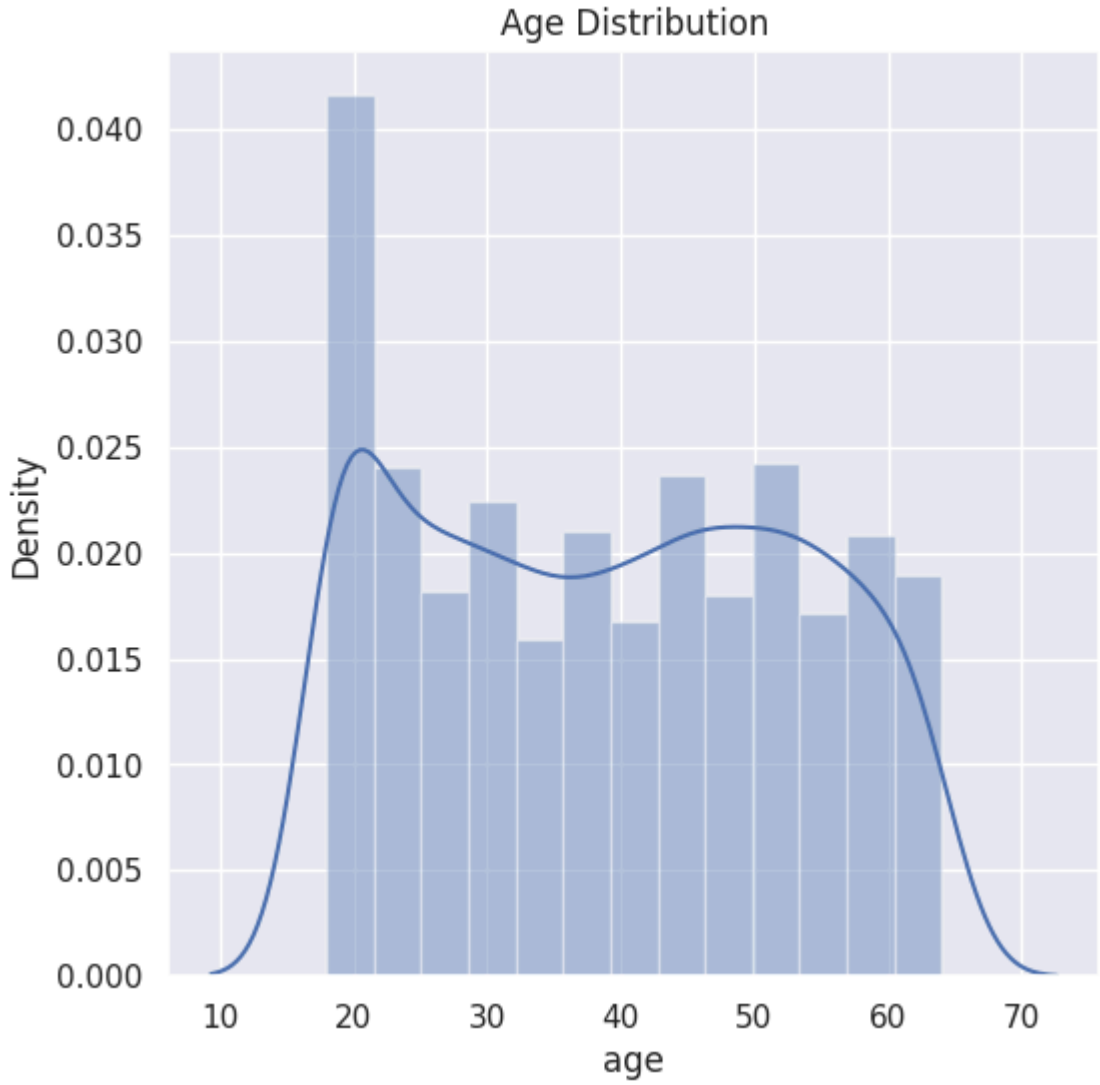
Data Analysis

```
# statical measures of the dataset
insurance_dataset.describe()
```

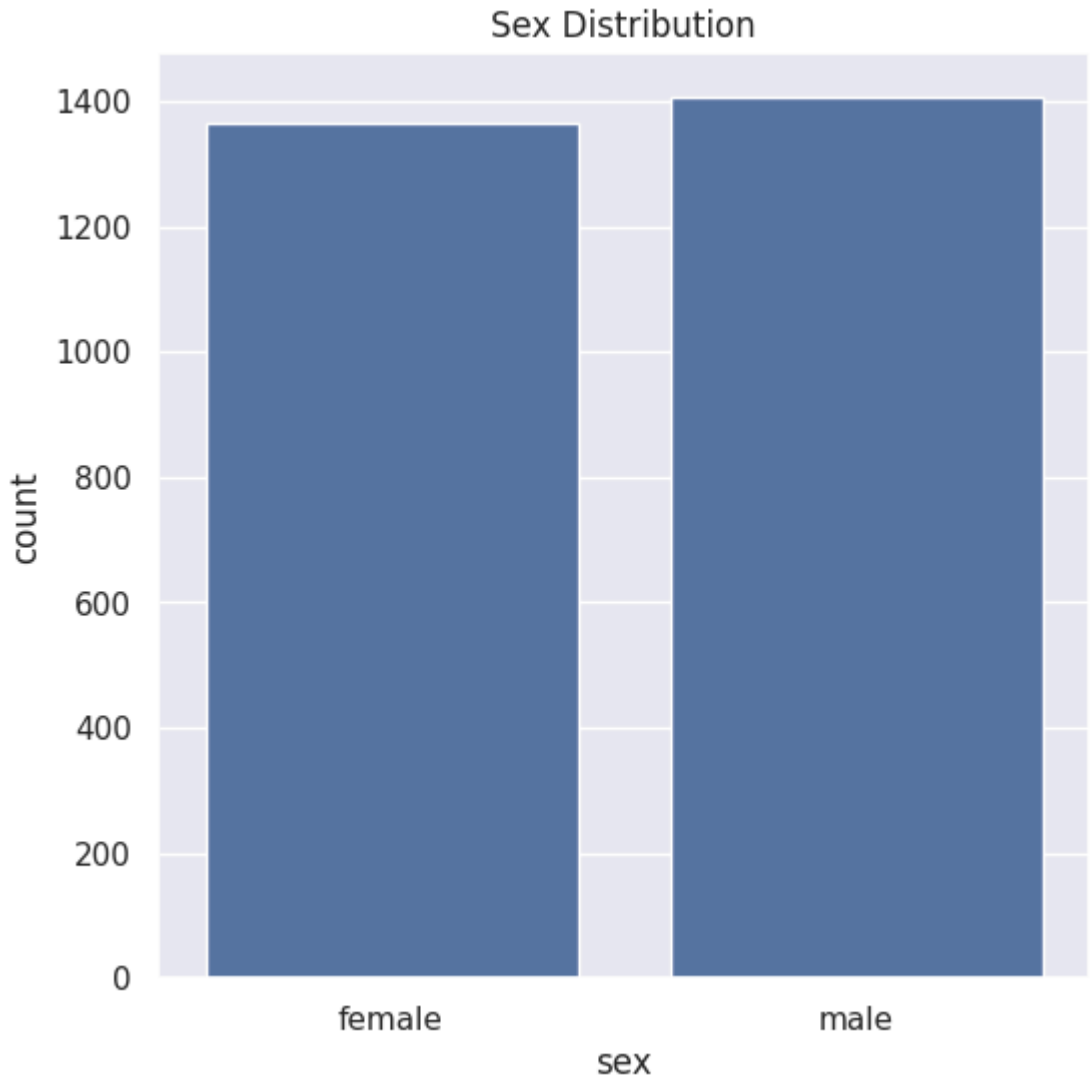
	age	bmi	children	charges
count	2772.000000	2772.000000	2772.000000	2772.000000
mean	39.109668	30.701349	1.101732	13261.369959
std	14.081459	6.129449	1.214806	12151.768945
min	18.000000	15.960000	0.000000	1121.873900
25%	26.000000	26.220000	0.000000	4687.797000
50%	39.000000	30.447500	1.000000	9333.014350
75%	51.000000	34.770000	2.000000	16577.779500
max	64.000000	53.130000	5.000000	63770.428010

```
#distribution of age value
sns.set()
plt.figure(figsize=(6,6))
sns.distplot(insurance_dataset['age'])
plt.title('Age Distribution')
plt.show()
```

```
<ipython-input-79-30bd7651d2c1>:4: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(insurance_dataset['age'])
```




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



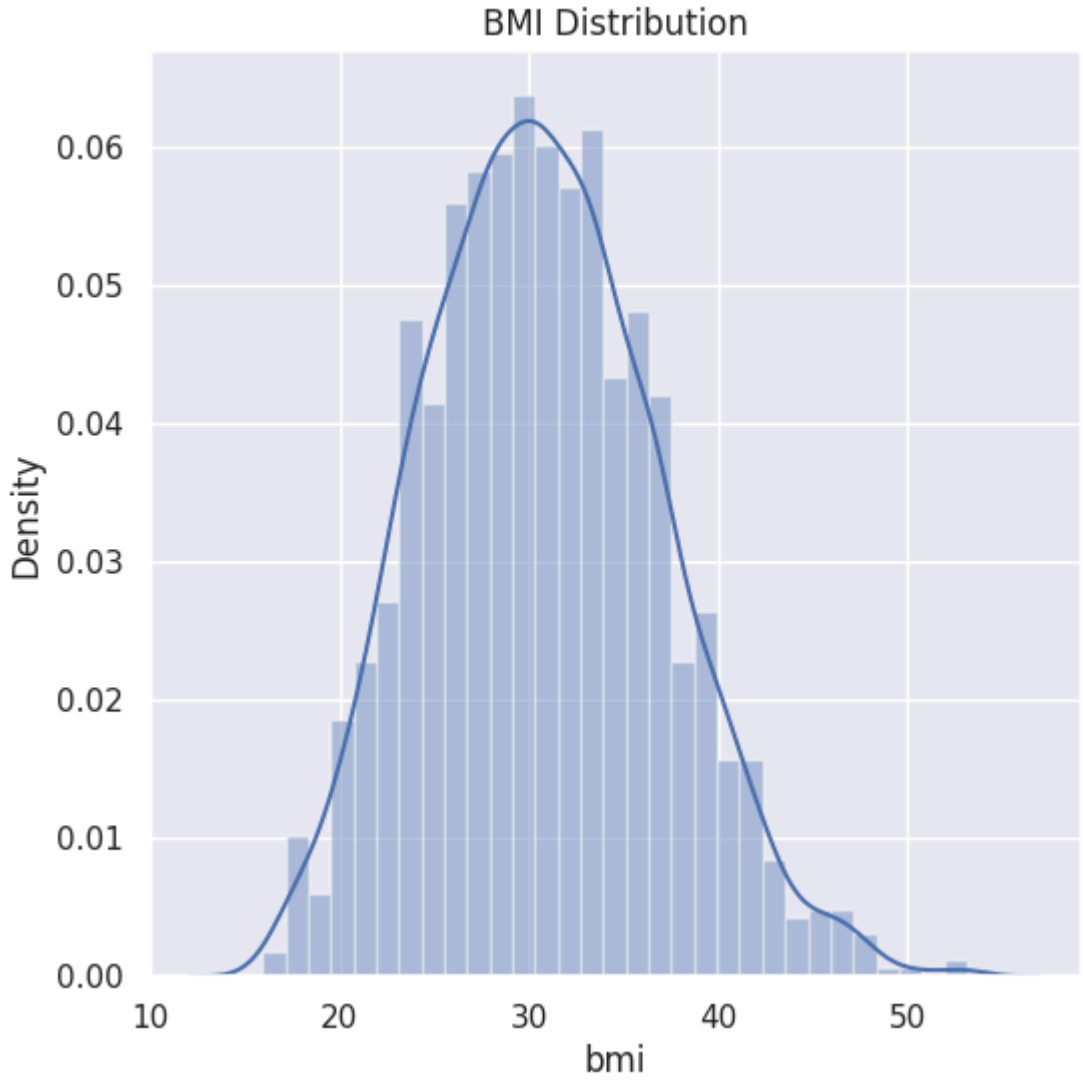

```
insurance_dataset['sex'].value_counts()
```

sex	count
male	1406
female	1366

dtype: int64

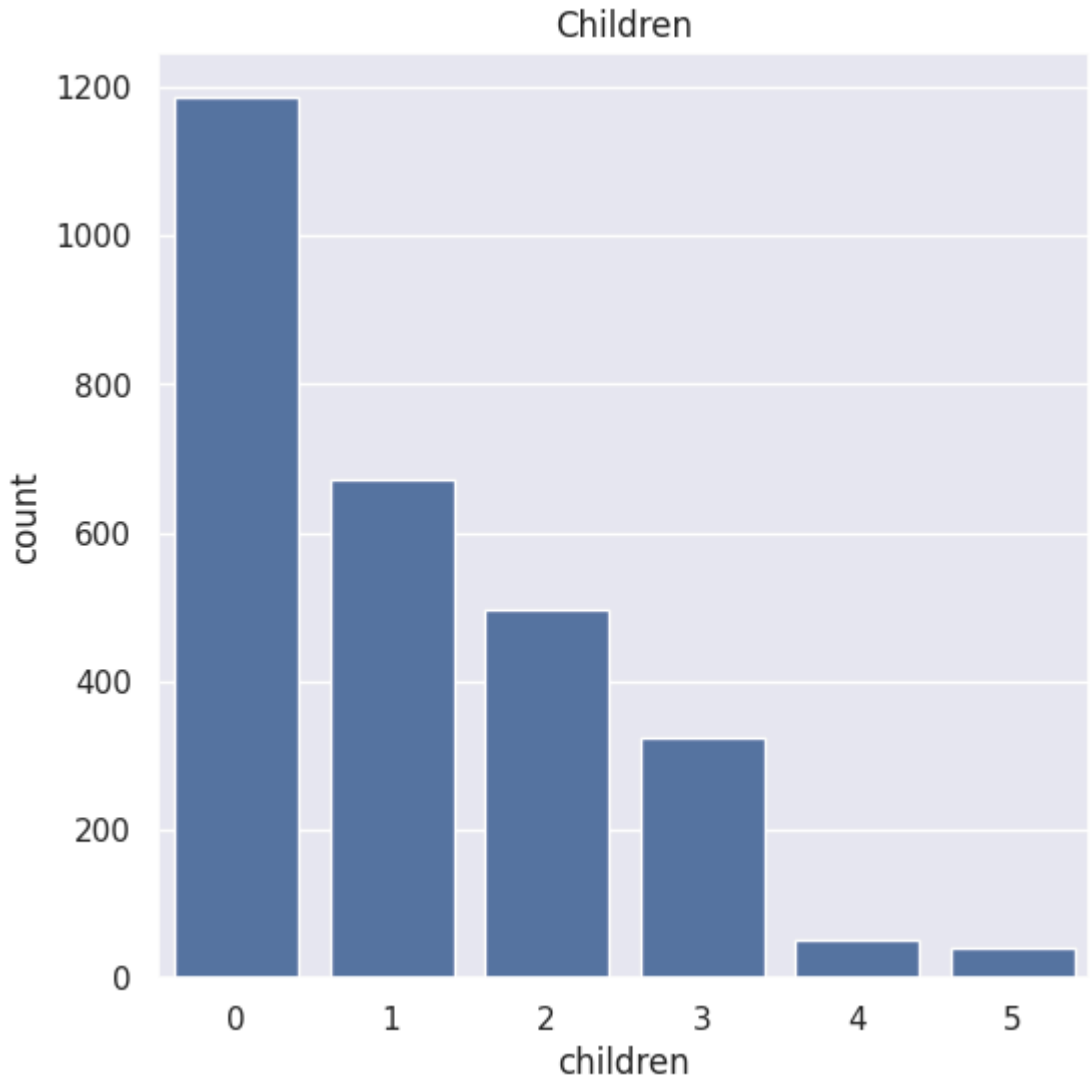
```
# BMI Distribution
plt.figure(figsize=(6,6))
sns.distplot(insurance_dataset['bmi'])
plt.title('BMI Distribution')
plt.show()
```

```
<ipython-input-82-de48435fbc01>:3: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(insurance_dataset['bmi'])
```



Normal Bmi Range --> 18.5 To 24.9

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

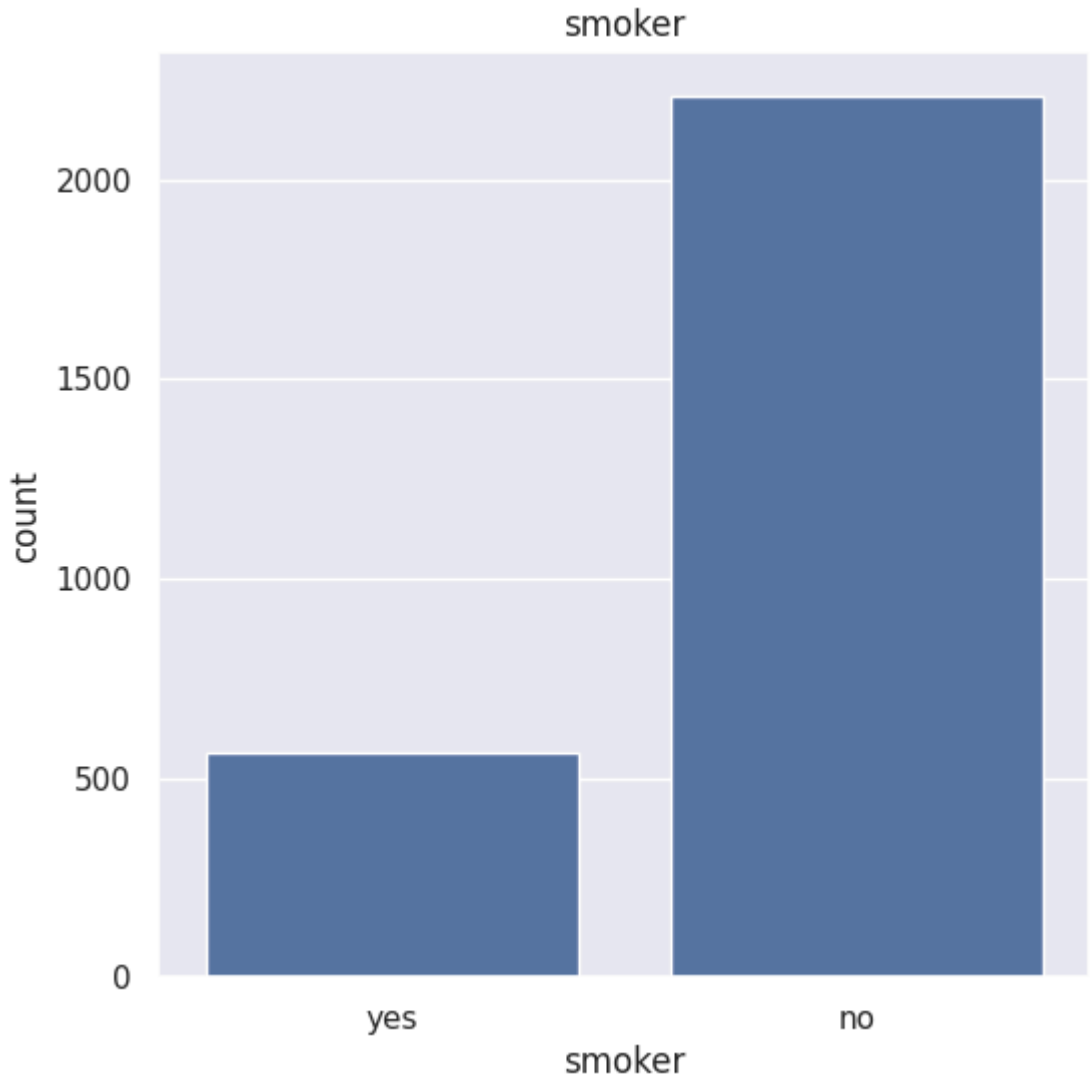



```
insurance_dataset['children'].value_counts()
```

children	count
0	1186
1	672
2	496
3	324
4	52
5	42

dtype: int64

```
#Smoker column
plt.figure(figsize=(6,6))
sns.countplot(x='smoker',data=insurance_dataset)
plt.title('smoker')
plt.show()
```

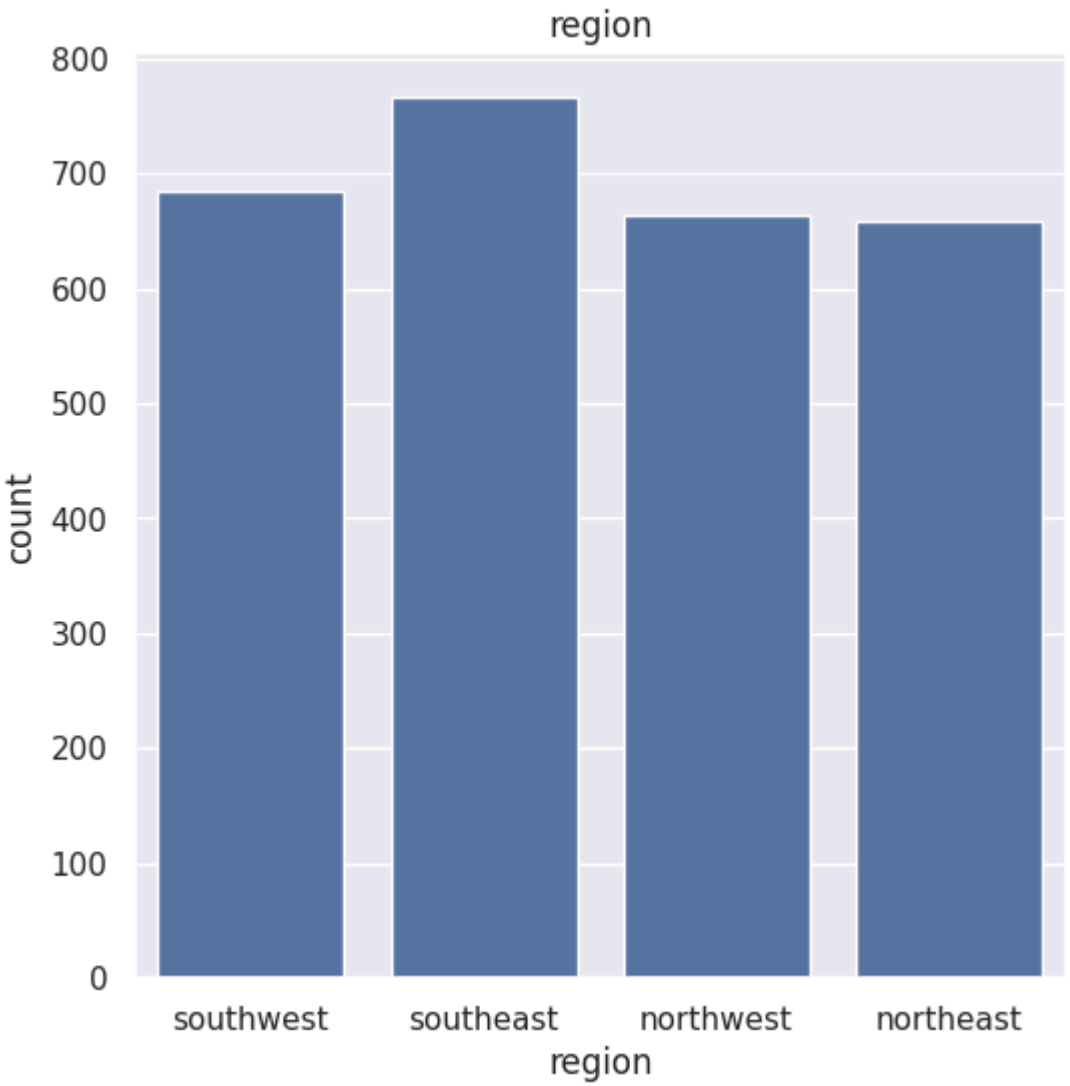



```
insurance_dataset['smoker'].value_counts()
```

smoker	count
no	2208
yes	564

dtype: int64

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



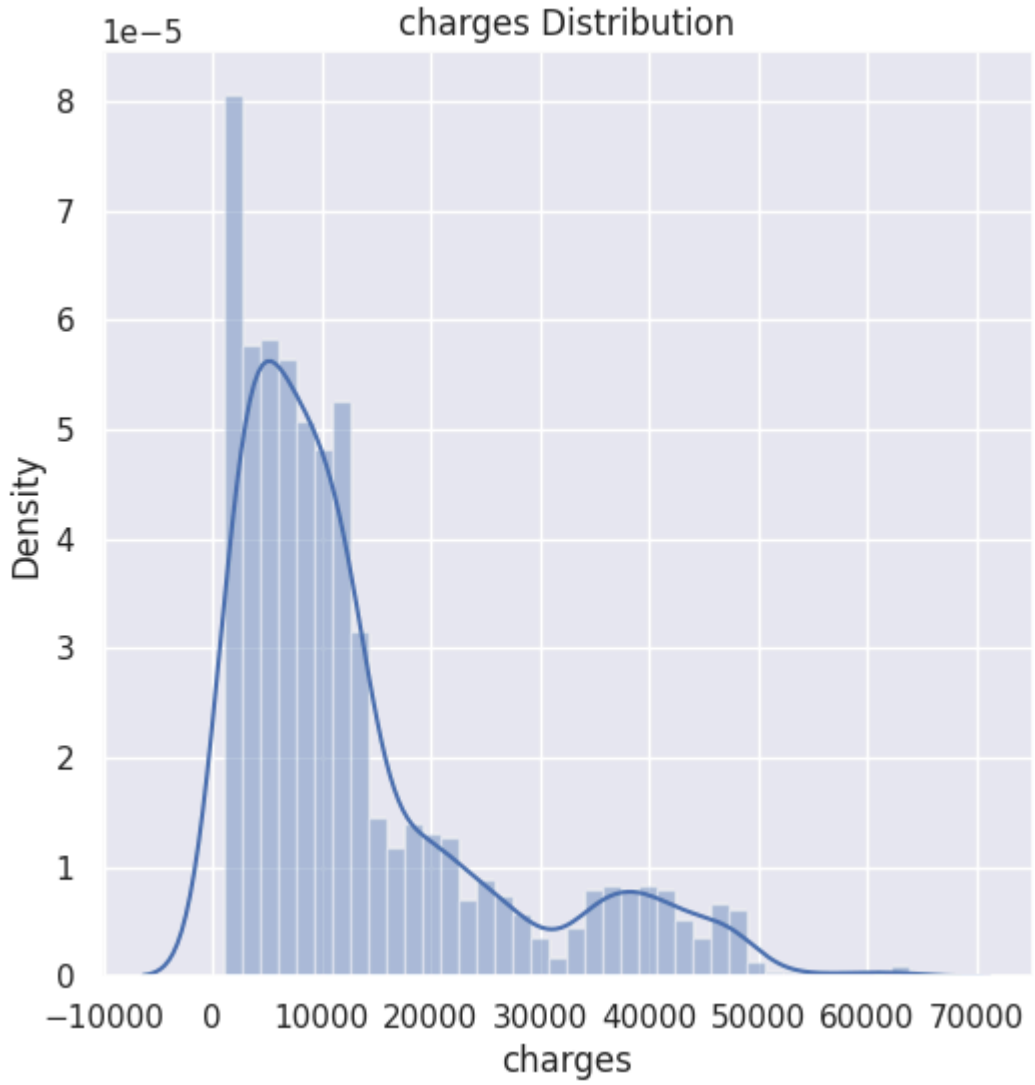

```
insurance_dataset['region'].value_counts()
```

region	count
southeast	766
southwest	684
northwest	664
northeast	658

dtype: int64

```
#  Distribution of chares values
plt.figure(figsize=(6,6))
sns.distplot(insurance_dataset['charges'])
plt.title('charges Distribution')
plt.show()
```

```
<ipython-input-89-10c36008c91b>:3: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(insurance_dataset['charges'])
```



Data Pre-Processing

Encoding the categorial feature

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

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

#encoding region column
```

```
insurance_dataset.replace({'region':{'southeast':0,'southwest':1,'northeast':2,'northwest
```

```
<ipython-input-90-5c813f3b4a07>:2: FutureWarning: Downcasting behavior in `replace` i
  insurance_dataset.replace({'sex':{'male':0, 'female':1}}, inplace=True)
<ipython-input-90-5c813f3b4a07>:5: FutureWarning: Downcasting behavior in `replace` i
  insurance_dataset.replace({'smoker':{'yes':0, 'no':1}}, inplace=True)
<ipython-input-90-5c813f3b4a07>:8: FutureWarning: Downcasting behavior in `replace` i
  insurance_dataset.replace({'region':{'southeast':0,'southwest':1,'northeast':2,'nor
```

Splitting the feature and Target

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

```
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
...   ...  ...     ...       ...     ...     ...
2767   47    1  45.320         1       1       0
2768   21    1  34.600         0       1       1
2769   19    0  26.030         1       0       3
2770   23    0  18.715         0       1       3
2771   54    0  31.600         0       1       1

[2772 rows x 6 columns]
```

```
print(Y)
```

```
0       16884.92400
1        1725.55230
2        4449.46200
3       21984.47061
4        3866.85520
           ...
2767     8569.86180
2768     2020.17700
2769    16450.89470
2770    21595.38229
2771     9850.43200
Name: charges, Length: 2772, dtype: float64
```

Splitting the data into Training data and Teasting Data

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

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

```
(2772, 6) (2217, 6) (555, 6)
```

Liner Regeression

```
#Loading the linear Regression Model
```