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


#loading the data from csv file to a pandas dataframe
insurance_dataset = pd.read_csv('/content/insurance.csv')
insurance_dataset
```

| | 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

```
# First 5 rows of the dataset
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 |

```
# FInding no.of rows and columns
insurance_dataset.shape
```

```
(1338, 7)
```

```
# Information about the dataset
insurance_dataset.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
```

```
#Checking the missing values
insurance_dataset.isnull().sum()
```

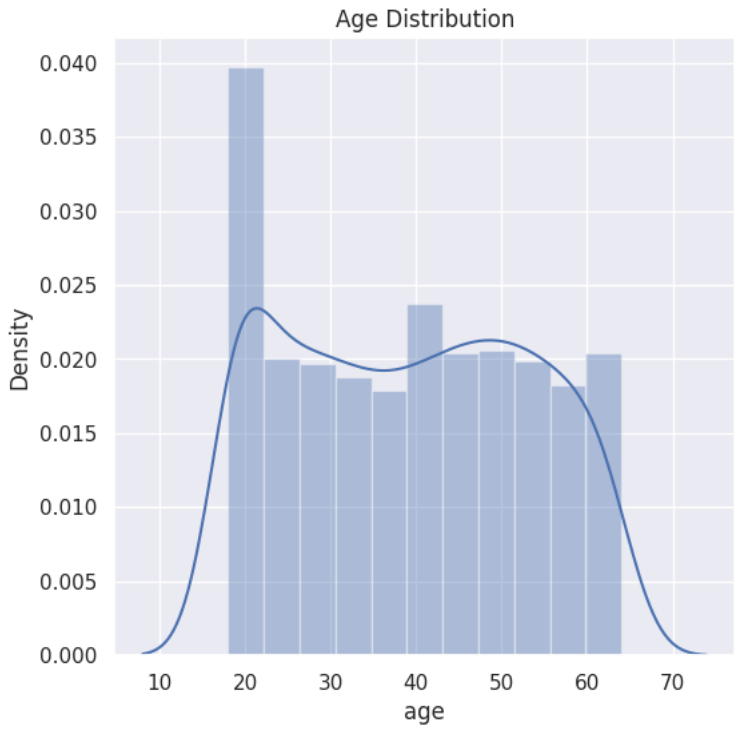
```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

```
# Statsitical measures of the dataset
insurance_dataset.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 |

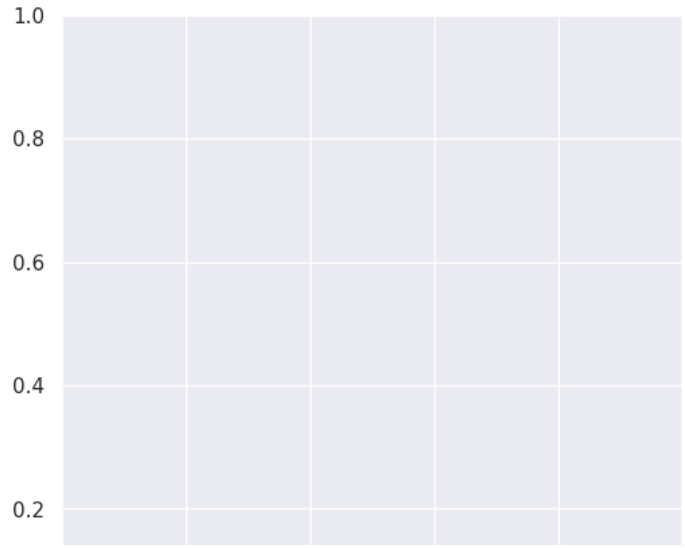
```
#Distribution of age value
sns.set()
plt.figure(figsize = (6,6))
sns.distplot(insurance_dataset['age'])
plt.title("Age Distribution")
plt.show()
```

```
<ipython-input-10-c8c9a2bbae5e>: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'])
```

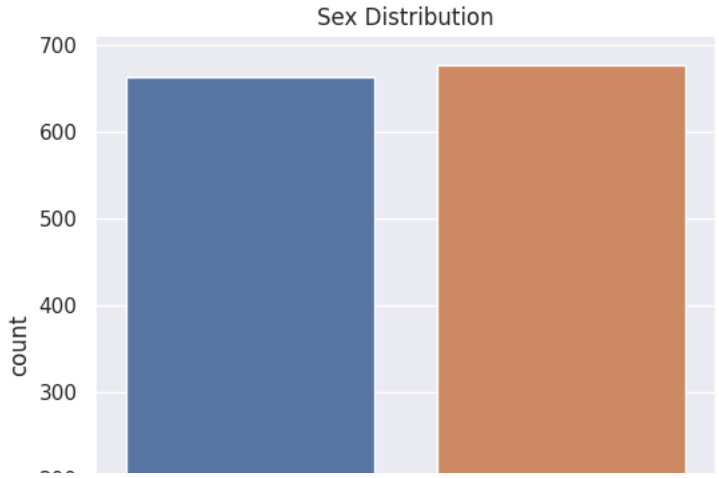


```
#Gender column
sns.set()
plt.figure(figsize = (6,6))
sns.distplot(insurance_dataset['sex'])
plt.title("sex Distribution")
plt.show()
```

```
<ipython-input-11-f1ce334ff304>: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['sex'])
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-11-f1ce334ff304> in <cell line: 4>()
      2 sns.set()
      3 plt.figure(figsize = (6,6))
----> 4 sns.distplot(insurance_dataset['sex'])
      5 plt.title("sex Distribution")
      6 plt.show()

 1 frames
/usr/local/lib/python3.10/dist-packages/pandas/core/series.py in __array__(self,
dtype)
    891               dtype='datetime64[ns]')
    892         """
--> 893         return np.asarray(self._values, dtype)
    894
    895     # ----------------------------------------------------------------------

ValueError: could not convert string to float: 'female'

SEARCH STACK OVERFLOW
```




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

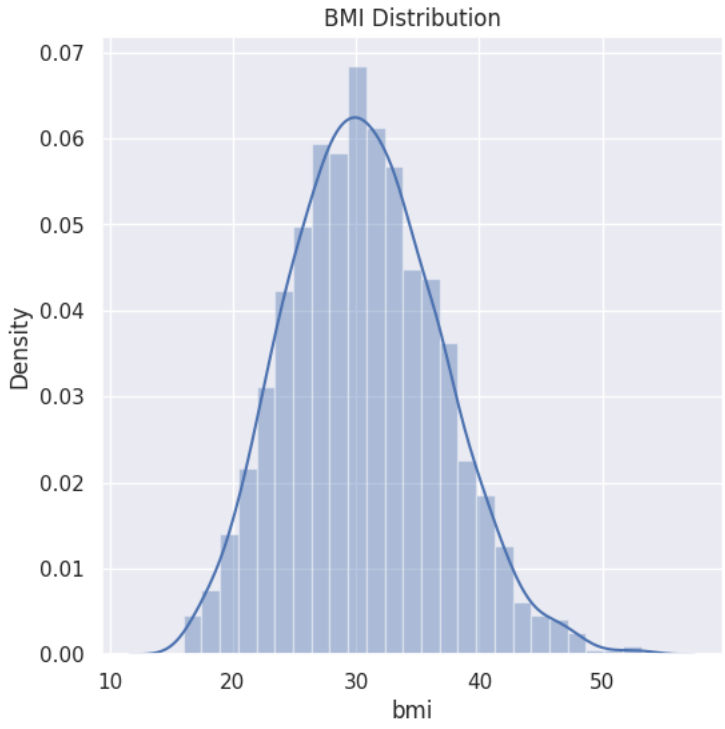



```
insurance_dataset['sex'].value_counts()
```

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

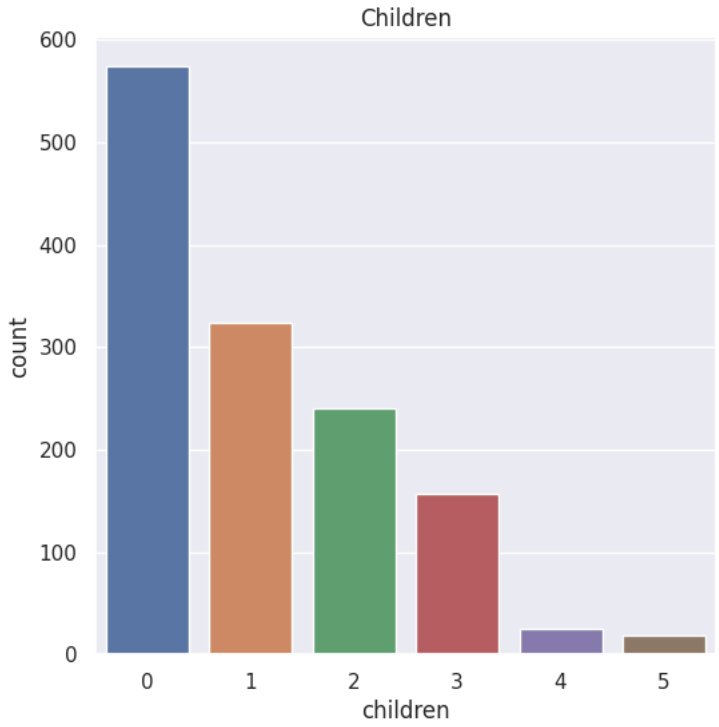
```
#Bmi Distribution
sns.set()
plt.figure(figsize = (6,6))
sns.distplot(insurance_dataset['bmi'])
plt.title("BMI Distribution")
plt.show()
```

```
<ipython-input-14-31866dc3d15b>: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['bmi'])
```

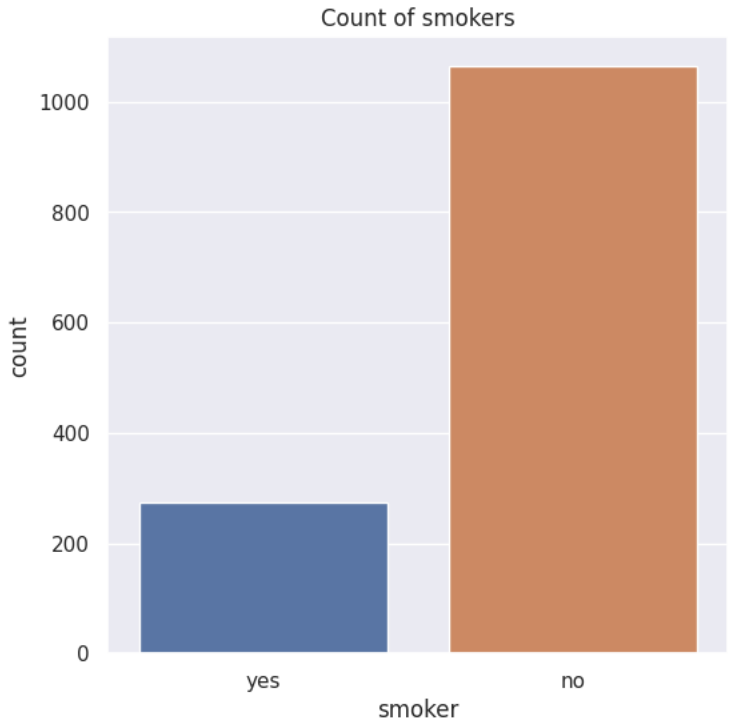



**Normal BMI Range 18.5 to 24.9**

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



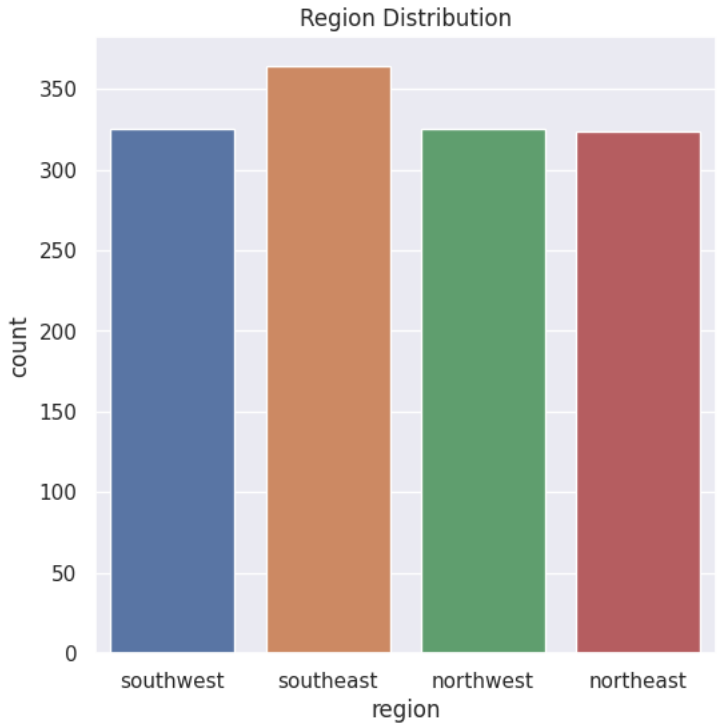

```
# Distribution for smoker
plt.figure(figsize= (6,6))
sns.countplot(x = 'smoker', data = insurance_dataset)
plt.title('Count of smokers')
plt.show()
```




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

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

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




```
# Charges Coulmn
plt.figure(figsize = (6,6))
sns.countplot(x = 'charges', data= insurance_dataset)
plt.title('Charges Distribution')
plt.show()
```

Data Preprocessing

2.00

Encoding the categorical features **bold text**

1.75

```
# 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' : 3}}, inplace = True)
```

**Splitting Features and Target**

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

Y = insurance_dataset['charges']
```

**Splitting the data into training and testing 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)
```

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

**Model Training**

```
# Calling Linear Regression Model
regressor = LinearRegression()

regressor.fit(X_train, Y_train)
```

```
LinearRegression
LinearRegression()
```

**Model Evaluation**

```
# Prediction on training data
trainig_data_prediction = regressor.predict(X_train)

# R Squared Value on training data
r2_train = metrics.r2_score(Y_train, trainig_data_prediction)
print("R Squared Value:" , r2_train)
```

```
R Squared Value: 0.751505643411174
```

```
# Prediction on testing data
testing_data_prediction = regressor.predict(X_test)

# R squared Value on testing data
r2_test = metrics.r2_score(Y_test, testing_data_prediction)
print("R Squared Value:" , r2_test)
```

```
R Squared Value: 0.7447273869684076
```

**Builidng a Predictive System**

```
input_data = (56,0,19.95,0,0,2)
```

```
#Chaning inp_data into a numpy array
input_data_as_numpy_array = np.asarray(input_data)

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


#Predicting the cost
prediction = regressor.predict(input_data_reshaped)
print("The Cost of your Model Insurance is:", prediction)
```

```
The Cost of your Model Insurance is: [32457.19553961]
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but LinearRegression was
  warnings.warn(
```