```
from google.colab import files
uploaded = files.upload()
```

Choose files insurance.csv

- **insurance.csv**(text/csv) - 55628 bytes, last modified: 23/12/2023 - 100% done

```
Saving insurance.csv to insurance.csv
```

```
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
```

```
insurance_dataset = pd.read_csv('insurance.csv')
```

```
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

```
insurance_dataset.shape
```

```
(1338, 7)
```

```
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
```

```
insurance_dataset.isnull().sum()
```

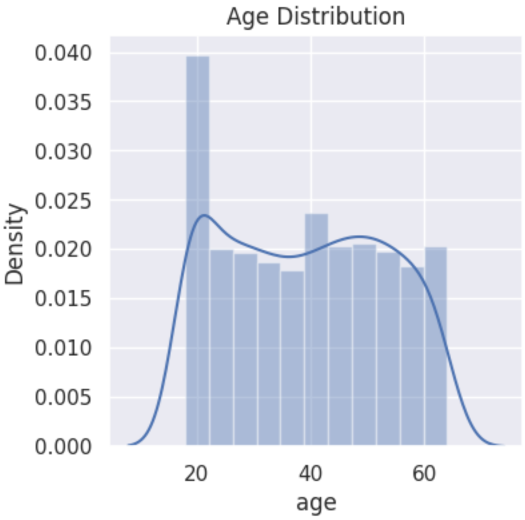
```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

```
insurance_dataset.describe()
```

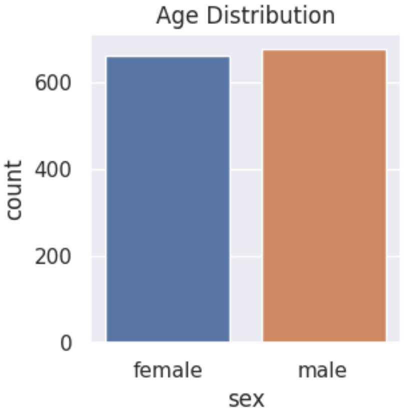
	age	bmi	children	charges
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.663397	1.094918	13270.422265
std	14.049960	6.098187	1.205493	12110.011237

```
sns.set()
plt.figure(figsize=(4,4))
sns.distplot(insurance_dataset['age'])
plt.title('Age Distribution')
plt.show()
```

```
<ipython-input-14-dcbb337c802d>: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['age'])
```



```
plt.figure(figsize=(3,3))
sns.countplot(x='sex',data=insurance_dataset)
plt.title("Age Distribution")
plt.show()
```

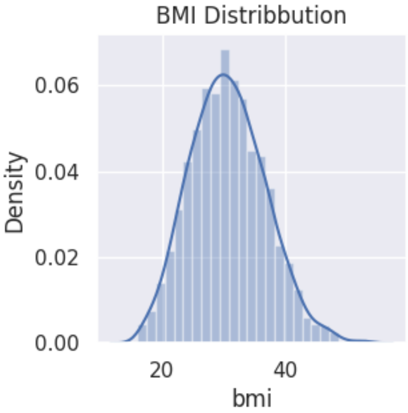


```
insurance_dataset['sex'].value_counts()
```

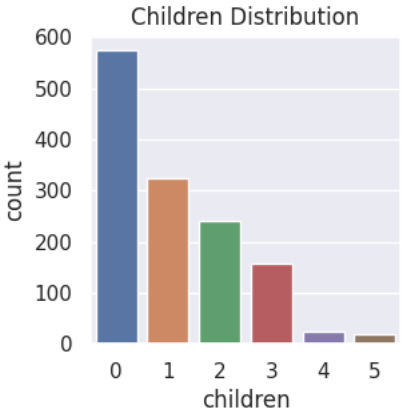
```
male      676
female    662
Name: sex, dtype: int64
```

```
plt.figure(figsize=(3,3))
sns.distplot(insurance_dataset['bmi'])
plt.title('BMI Distribbution')
plt.show()
```

```
<ipython-input-19-697a1bfc74b6>:2: 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'])
```



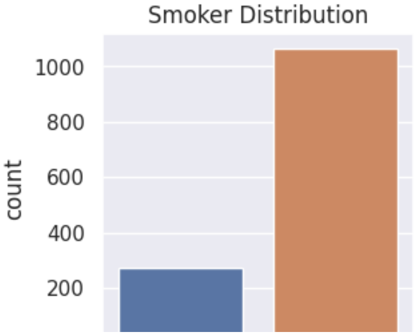
```
plt.figure(figsize=(3,3))
sns.countplot(x='children',data=insurance_dataset)
plt.title('Children Distribution')
plt.show()
```



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

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

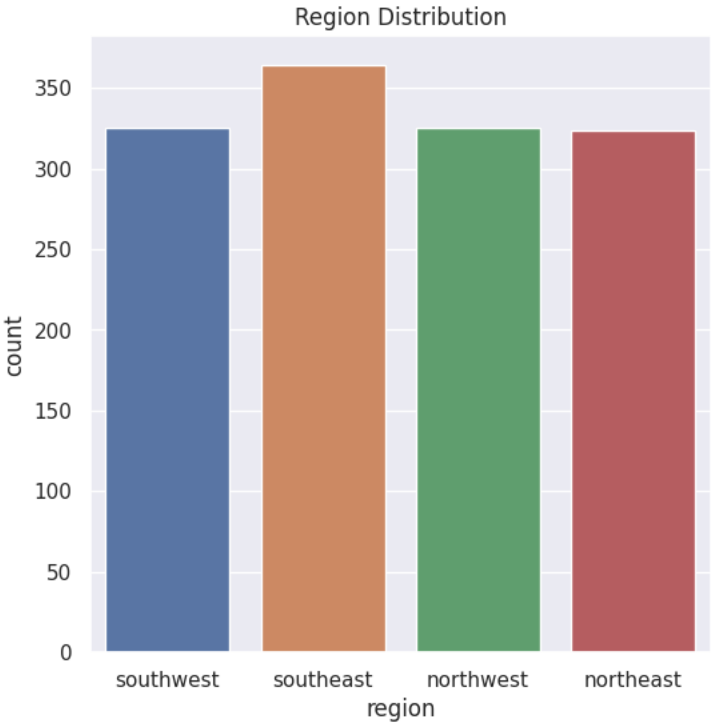
```
plt.figure(figsize=(3,3))
sns.countplot(x='smoker',data=insurance_dataset)
plt.title('Smoker Distribution')
plt.show()
```



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

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

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

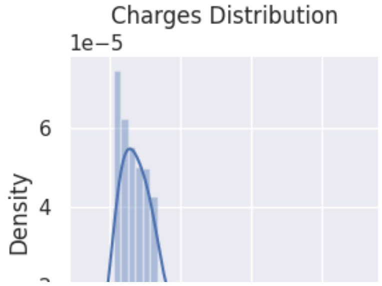


```
insurance_dataset['region'].value_counts()
```

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

```
plt.figure(figsize=(3,3))
sns.distplot(insurance_dataset['charges'])
plt.title('Charges Distribution')
plt.show()
```

```
<ipython-input-29-f56306017149>:2: 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'])
```



```
# STEP 01 ENCODING
```

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

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

```
insurance_dataset.replace({'region':{'southeast':0,'southwest':1,'northeast':2,'northwest':3}},inplace=True)
```

```
# STEP 2 SPLITTING
```

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

```
print(X)
print(Y)
```

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

[1338 rows x 6 columns]
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
```

```
# STEP 3 SPLITTING
```

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

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

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

```
regressor = LinearRegression()
```

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

```
▾ LinearRegression
LinearRegression()
```

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

```
r2_train = metrics.r2_score(Y_train,training_data_prediction)
print(f"R2 Train = {r2_train:.2f}")
```

```
R2 Train = 0.76
```

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

```
r2_test = metrics.r2_score(Y_test,testing_data_prediction)
print(f"R2 Test = {r2_test:.2f}")
```

```
R2 Test = -0.86
```

```
my_data = (31,1,25.74,0,1,0)
numpy_array = np.asarray(my_data)
reshaped_data=numpy_array.reshape(1,-1)
prediction=regressor.predict(reshaped_data)
print(prediction[0])
```

```
27345.39235140915
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but LinearRegression was
  warnings.warn(
```