```
import pandas as pd
import numpy as np
import seaborn as sns
```

```
import matplotlib.pyplot as plt
```

```
data=pd.read_csv("/content/insurance.csv")
```

```
data.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 data View recommended plots

```
data.tail()
```

	age	sex	bmi	children	smoker	region	charges
1333	50	male	30.97	3	no	northwest	10600.5483
1334	18	female	31.92	0	no	northeast	2205.9808
1335	18	female	36.85	0	no	southeast	1629.8335
1336	21	female	25.80	0	no	southwest	2007.9450
1337	61	female	29.07	0	yes	northwest	29141.3603

```
data.shape
```

```
(1338, 7)
```

```
data.isnull().sum()
```

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

```
df1=data.drop(columns=['bmi','children','charges','region'],axis=1)
```

```
df1
```

	age	sex	smoker
0	19	female	yes
1	18	male	no
2	28	male	no
3	33	male	no
4	32	male	no
...	...	...	...
1333	50	male	no
1334	18	female	no
1335	18	female	no
1336	21	female	no
1337	61	female	yes

1338 rows × 3 columns

Next steps: Generate code with `df1` View recommended plots

```
IQR=df1['age'].quantile(0.75)-df1['age'].quantile(0.25)
```

```
IQR
```

```
24.0
```

```
lowerBound=df1['age'].quantile(0.75)-(1.5*IQR)
```

```
lowerBound
```

```
15.0
```

```
upperBound=df1['age'].quantile(0.75)+(1.5*IQR)
```

```
upperBound
```

```
87.0
```

```
from sklearn.preprocessing import LabelEncoder
```

```
le=LabelEncoder()
```

```
df1.sex=le.fit_transform(df1.sex)
df1.smoker=le.fit_transform(df1.smoker)
```

```
df1
```

	age	sex	smoker
0	19	0	1
1	18	1	0
2	28	1	0
3	33	1	0
4	32	1	0
...	...	...	...
1333	50	1	0
1334	18	0	0
1335	18	0	0
1336	21	0	0
1337	61	0	1

1338 rows × 3 columns

```
x=df1.iloc[:,:-1].values
y=df1.iloc[:,-1].values
```

```
x[0]
```

```
array([19,  0])
```

```
y[0]
```

```
1
```

```
from sklearn.model_selection import train_test_split
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=12)
```

```
x_train.shape
```

```
(1070, 2)
```

```
x_test.shape
```

```
(268, 3)
```

```
from sklearn.preprocessing import StandardScaler
```

```
sc=StandardScaler()
```

```
x_train=sc.fit_transform(x_train)
```

```
x_test=sc.transform(x_test)
```

```
x_train[0]
```

```
array([-1.47165638, -1.01506676])
```

```
x_test[0]
```

```
array([0.35323689, 0.98515688])
```

```
from sklearn.linear_model import LogisticRegression
```

```
model=LogisticRegression()
model.fit(x_train,y_train)
```

```
▾ LogisticRegression
LogisticRegression()
```

```
pred=model.predict(x_test)
```

```
pred
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0])
```

```
y_test
```

```
array([0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,
       0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
       1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0,
       0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1,
       0, 0, 0, 0])
```

```
df1.tail()
```

	age	sex	smoker
1333	50	1	0
1334	18	0	0
1335	18	0	0
1336	21	0	0
1337	61	0	1

```
model.predict([[50,1]])
```

```
array([0])
```

```
from sklearn.metrics import accuracy_score
```

```
accuracy_score(y_test,pred)
```

```
0.8134328358208955
```

```
#
```

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.