In [67]:
```
## importing important libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings

warnings.filterwarnings("ignore")
```

In [68]:
```
df = pd.read_csv("insurance.csv")
```

In [69]:
```
df
```

Out[69]:

	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

EDA

In [70]:
```
## Number of rows & columns
df.shape ## Data has 7 rows & 1338 rows
```

Out[70]:
```
(1338, 7)
```

In [71]:
```
## Top 5 rows
df.head()
```

Out[71]:

	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

In [72]:
```
## Information of each column
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
```

In [73]:
```
## Statistical measures of columns with quantitative data
df.describe()
```

Out[73]:

	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

In [74]:
```
## Checking for null values
df.isnull().sum().sum() ## There are no null values in the data
```

Out[74]:
```
np.int64(0)
```

In [75]:
```
## Checking for duplicate values
df.duplicated().sum() ## There is one duplicate record
```

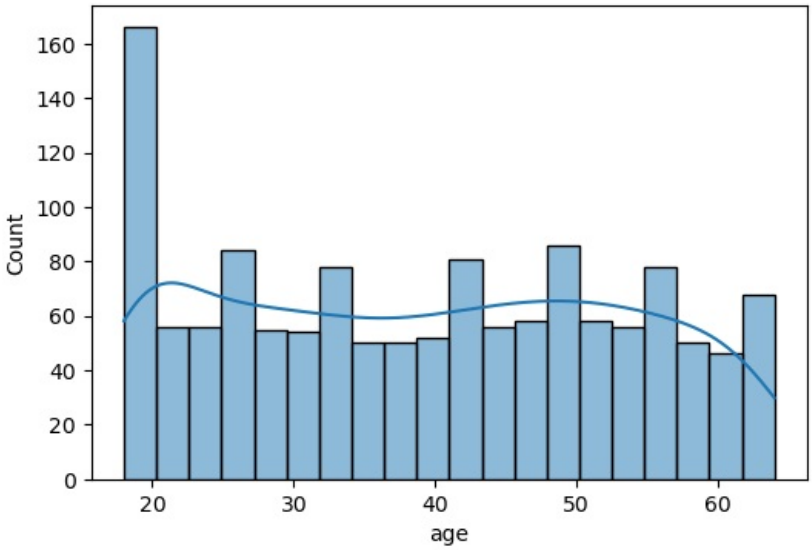
Out[75]:
```
np.int64(1)
```

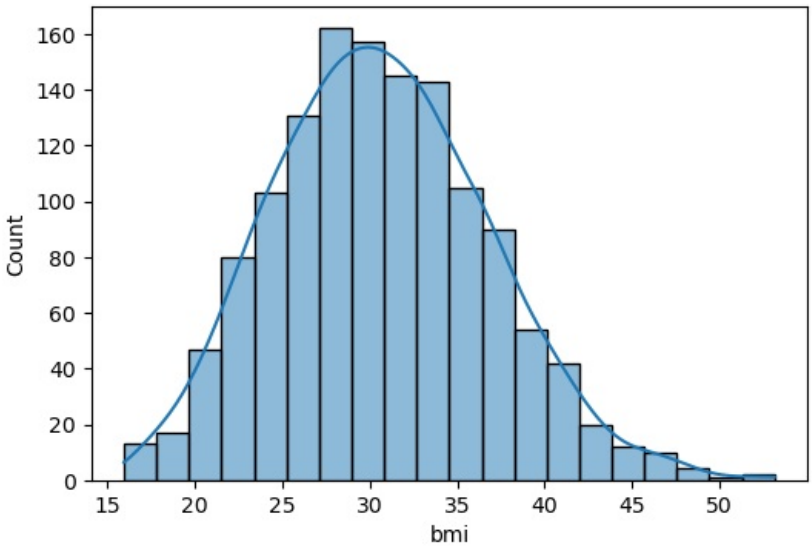
In [76]:
```
df[df.duplicated()] ## This is the duplicate record
```

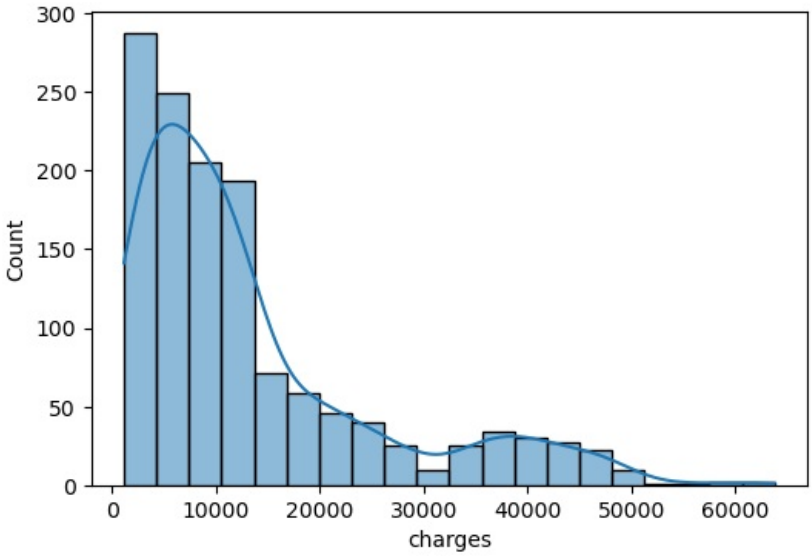
Out[76]:

	age	sex	bmi	children	smoker	region	charges
581	19	male	30.59	0	no	northwest	1639.5631

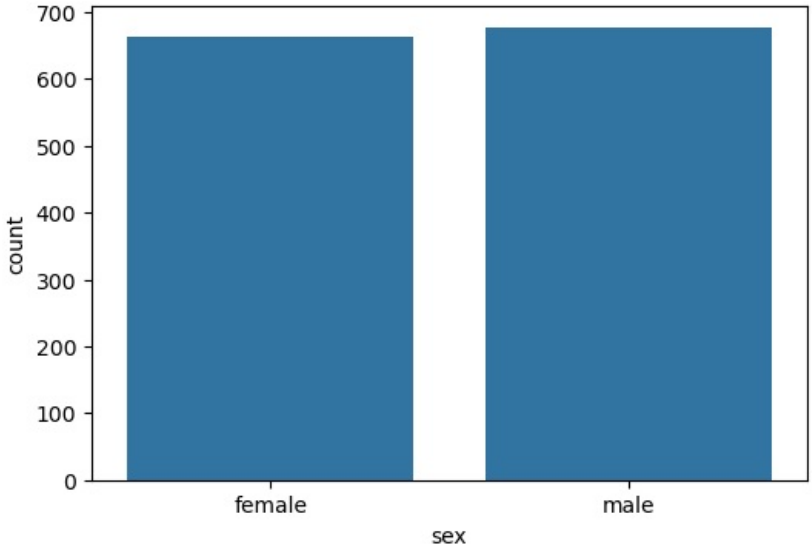
In [77]:
```
## Histogram of all continuous numerical columns
numeric_columns = ["age","bmi","charges"]
for col in numeric_columns:
    plt.figure(figsize=(6,4))
    sns.histplot(df[col],kde=True,bins=20)
```

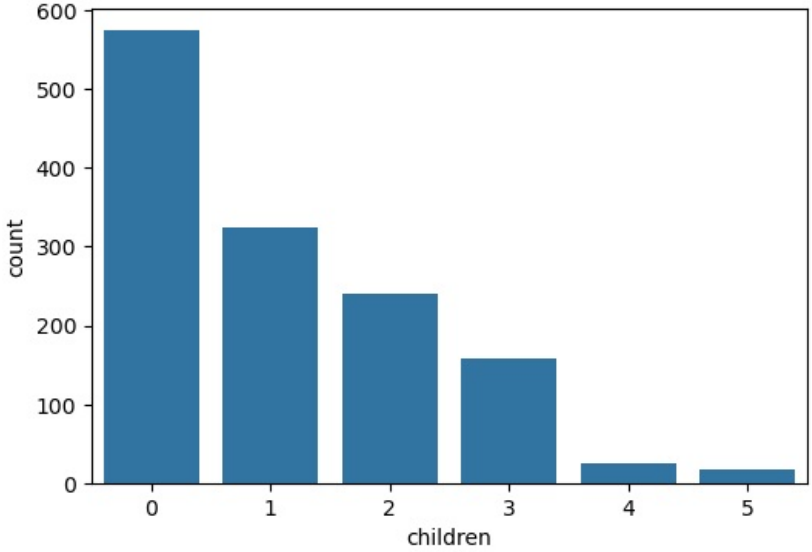


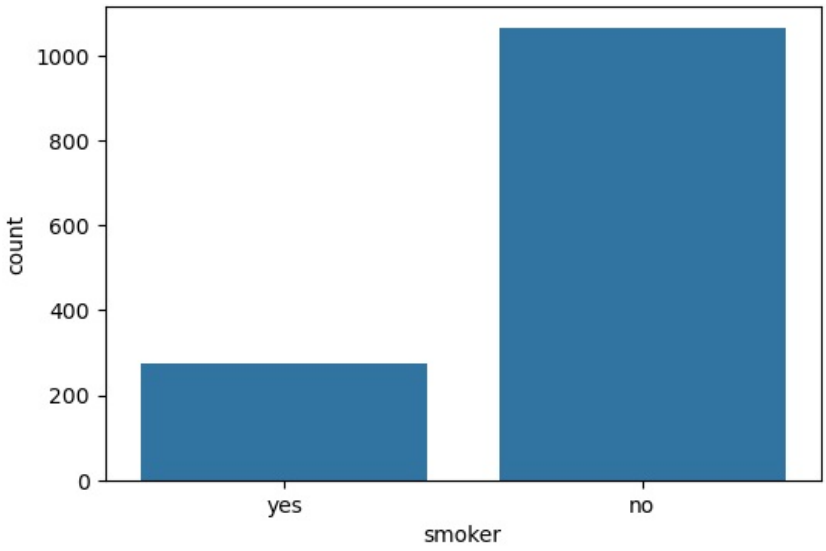


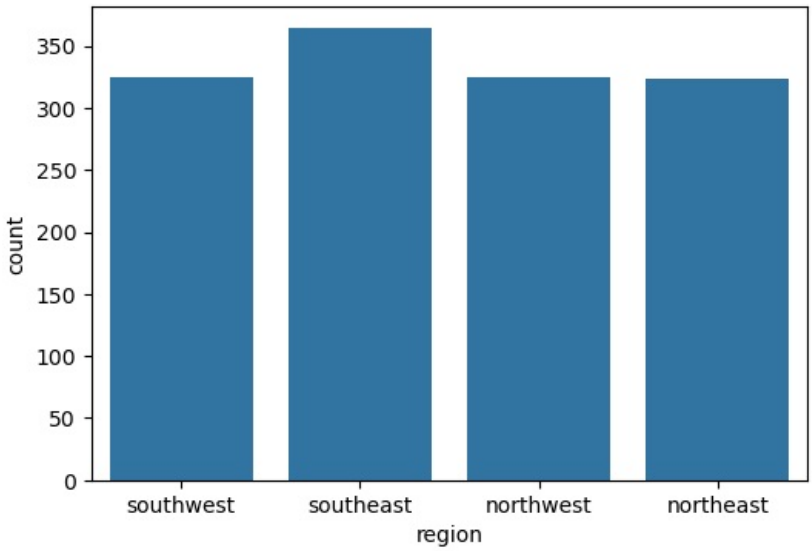


In [78]:
```
## countplot of all categorical columns
cat_columns = ["sex","children","smoker","region"]
for col in cat_columns:
    plt.figure(figsize=(6,4))
    sns.countplot(x=col,data=df)
```

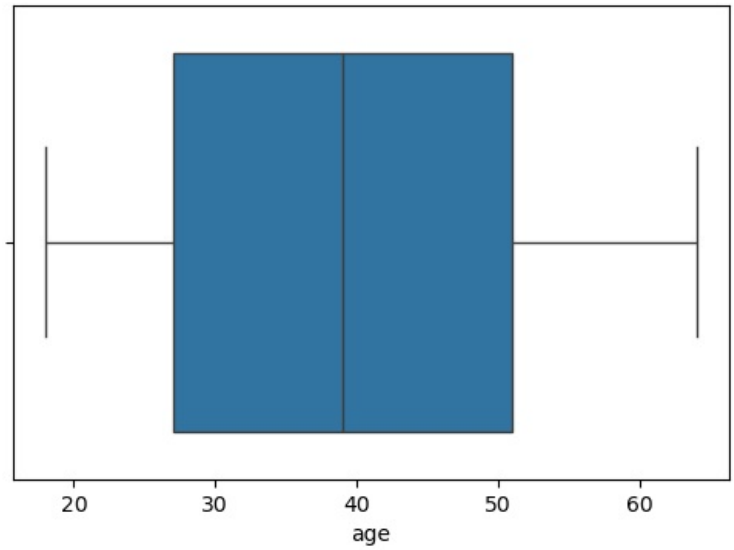


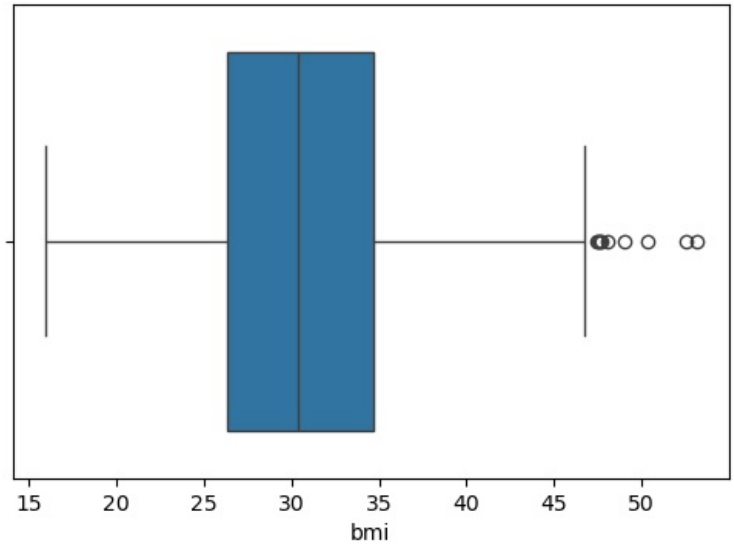


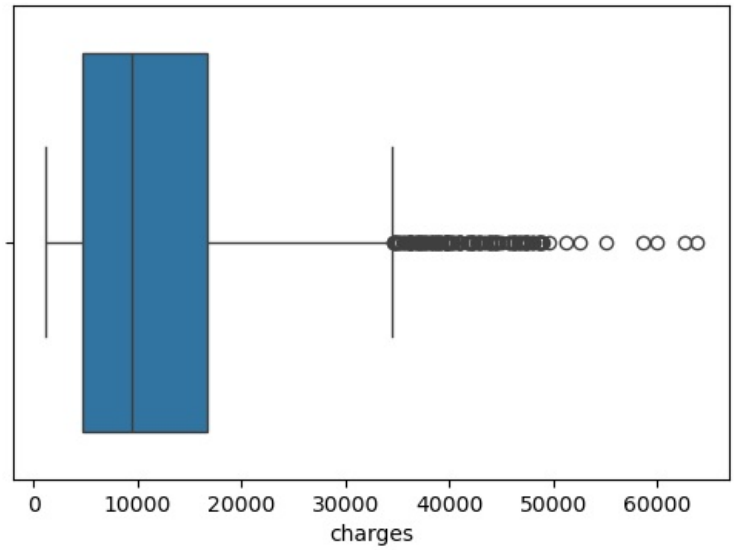




In [79]:
```
## boxplot of all continuous numerical columns
for col in numeric_columns:
    plt.figure(figsize=(6,4))
    sns.boxplot(x=df[col])
```

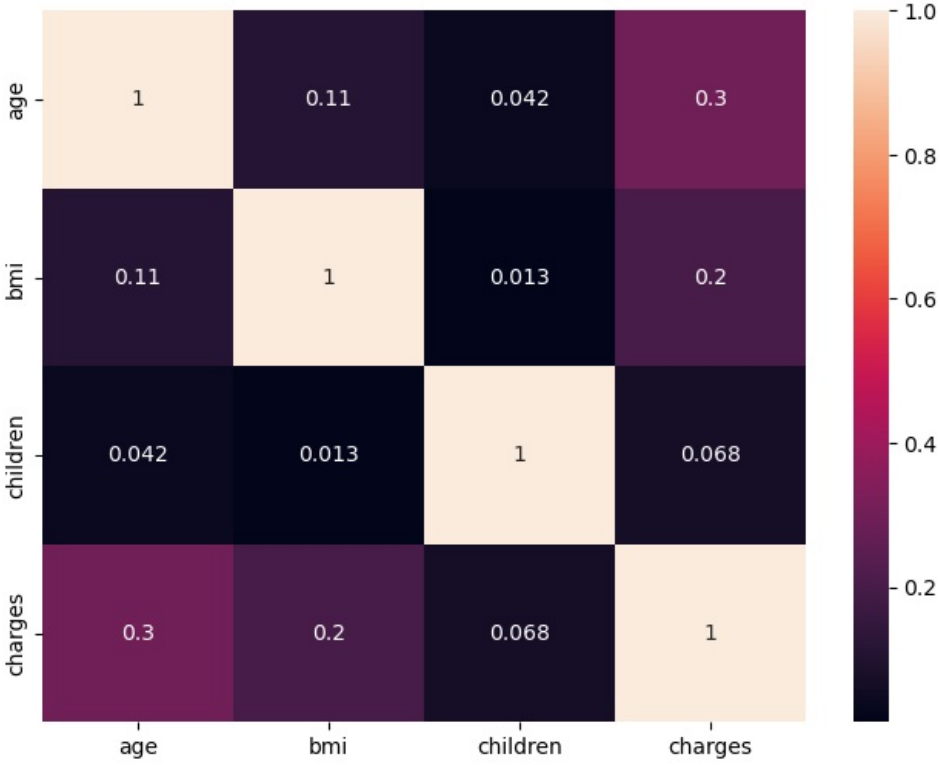






In [80]:
```
## Correllation through heatmap
plt.figure(figsize=(8,6))
sns.heatmap(df.corr(numeric_only=True),annot=True)
```

Out[80]: <Axes: >



Data Cleaning & Processing

In [81]:
```
## Copying the data
cleaned_data = df.copy()
```

In [82]:
```
## removing duplicates
cleaned_data.drop_duplicates(inplace=True)
```

In [83]:
```
cleaned_data
```

Out[83]:

	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

1337 rows × 7 columns

In [84]:
```
cleaned_data.dtypes ## need to convert object data type to int (data encoding)
```

Out[84]:
```
age           int64
sex          object
bmi         float64
children      int64
smoker       object
region       object
charges     float64
dtype: object
```

In [85]:
```
cleaned_data["sex"].value_counts()
```

Out[85]:
```
sex
male      675
female    662
Name: count, dtype: int64
```

In [86]:
```
## conerting sex column into int (0 for male & 1 for female)
cleaned_data["sex"] = cleaned_data["sex"].apply((lambda v: 0 if v == "male" else 1))
```

In [87]:
```
## conerting smoker column into int (0 for no & 1 for yes)
cleaned_data["smoker"] = cleaned_data["smoker"].apply((lambda v: 0 if v == "no" else 1))
```

In [88]:
```
cleaned_data.rename(columns={"sex":"Isfemale","smoker":"Issmoker"},inplace=True)
```

In [89]:
```
cleaned_data
```

Out[89]:

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

1337 rows × 7 columns

In [90]:
```
## one hot encoding of region column
cleaned_data = pd.get_dummies(cleaned_data,columns = ['region'])
```

In [91]:
```
cleaned_data = cleaned_data.astype(int)
```

In [92]:
```
cleaned_data
```

Out[92]:

	age	lsfemale	bmi	children	lssmoker	charges	region_northeast	region_northwest	region_southeast	region_southwest
0	19	1	27	0	1	16884	0	0	0	1
1	18	0	33	1	0	1725	0	0	1	0
2	28	0	33	3	0	4449	0	0	1	0
3	33	0	22	0	0	21984	0	1	0	0
4	32	0	28	0	0	3866	0	1	0	0
...	...	...	...	...	...	...	...	...	...	...
1333	50	0	30	3	0	10600	0	1	0	0
1334	18	1	31	0	0	2205	1	0	0	0
1335	18	1	36	0	0	1629	0	0	1	0
1336	21	1	25	0	0	2007	0	0	0	1
1337	61	1	29	0	1	29141	0	1	0	0

1337 rows × 10 columns

Feature engineering & extraction

In [93]:
```
## adding a new Feature called bmi category
cleaned_data['bmi_category'] = pd.cut(
    cleaned_data['bmi'],
    bins=[0, 18.5, 24.9, 29.9, float('inf')],
    labels=['Underweight', 'Normal', 'Overweight', 'Obese']
)
```

In [94]:
```
cleaned_data
```

Out[94]:

	age	Isfemale	bmi	children	Issmoker	charges	region_northeast	region_northwest	region_southeast	region_southwest	b
0	19	1	27	0	1	16884	0	0	0	1	
1	18	0	33	1	0	1725	0	0	1	0	
2	28	0	33	3	0	4449	0	0	1	0	
3	33	0	22	0	0	21984	0	1	0	0	
4	32	0	28	0	0	3866	0	1	0	0	
...	...	...	...	...	...	...	...	...	...	...	
1333	50	0	30	3	0	10600	0	1	0	0	
1334	18	1	31	0	0	2205	1	0	0	0	
1335	18	1	36	0	0	1629	0	0	1	0	
1336	21	1	25	0	0	2007	0	0	0	1	
1337	61	1	29	0	1	29141	0	1	0	0	

1337 rows × 11 columns

In [95]:
```
## one hot encoding of region column
cleaned_data = pd.get_dummies(cleaned_data,columns = ['bmi_category'])
```

In [96]:
```
cleaned_data = cleaned_data.astype(int)
```

In [97]:
```
cleaned_data
```

Out[97]:

	age	Isfemale	bmi	children	Issmoker	charges	region_northeast	region_northwest	region_southeast	region_southwest	b
0	19	1	27	0	1	16884	0	0	0	1	
1	18	0	33	1	0	1725	0	0	1	0	
2	28	0	33	3	0	4449	0	0	1	0	
3	33	0	22	0	0	21984	0	1	0	0	
4	32	0	28	0	0	3866	0	1	0	0	
...	...	...	...	...	...	...	...	...	...	...	
1333	50	0	30	3	0	10600	0	1	0	0	
1334	18	1	31	0	0	2205	1	0	0	0	
1335	18	1	36	0	0	1629	0	0	1	0	
1336	21	1	25	0	0	2007	0	0	0	1	
1337	61	1	29	0	1	29141	0	1	0	0	

1337 rows × 14 columns

In [98]:
```
cleaned_data.columns
```

Out[98]:
```
Index(['age', 'Isfemale', 'bmi', 'children', 'Issmoker', 'charges',
       'region_northeast', 'region_northwest', 'region_southeast',
       'region_southwest', 'bmi_category_Underweight', 'bmi_category_Normal',
       'bmi_category_Overweight', 'bmi_category_Obese'],
      dtype='object')
```

In [99]:
```
from sklearn.preprocessing import StandardScaler
cols = ['age','bmi','children']
scaler = StandardScaler()

cleaned_data[cols] = scaler.fit_transform(cleaned_data[cols])
```

In [100…
```
cleaned_data
```

Out[100…

	age	Isfemale	bmi	children	Issmoker	charges	region_northeast	region_northwest	region_southeast	region_s
0	-1.440418	1	-0.517949	-0.909234	1	16884	0	0	0	
1	-1.511647	0	0.462463	-0.079442	0	1725	0	0	1	
2	-0.799350	0	0.462463	1.580143	0	4449	0	0	1	
3	-0.443201	0	-1.334960	-0.909234	0	21984	0	1	0	
4	-0.514431	0	-0.354547	-0.909234	0	3866	0	1	0	
...	...	...	...	...	...	...	...	...	...	
1333	0.767704	0	-0.027743	1.580143	0	10600	0	1	0	
1334	-1.511647	1	0.135659	-0.909234	0	2205	1	0	0	
1335	-1.511647	1	0.952670	-0.909234	0	1629	0	0	1	
1336	-1.297958	1	-0.844753	-0.909234	0	2007	0	0	0	
1337	1.551231	1	-0.191145	-0.909234	1	29141	0	1	0	

1337 rows × 14 columns

In [101…

```
from scipy.stats import pearsonr

# ---------------------------------
# Pearson Correlation Calculation
# ---------------------------------

# List of features to check against target
selected_features = [
'age', 'Isfemale', 'bmi', 'children', 'Issmoker',
       'region_northeast', 'region_northwest', 'region_southeast',
       'region_southwest', 'bmi_category_Underweight', 'bmi_category_Normal',
       'bmi_category_Overweight', 'bmi_category_Obese']

correlations = {
    feature: pearsonr(cleaned_data[feature], cleaned_data['charges'])[0]
    for feature in selected_features
}
correlation_df = pd.DataFrame(list(correlations.items()), columns=['Feature', 'Pearson Correlation'])
correlation_df.sort_values(by='Pearson Correlation', ascending=False)
```

Out[101…

	Feature	Pearson Correlation
4	Issmoker	0.787234
0	age	0.298309
12	bmi_category_Obese	0.200348
2	bmi	0.196236
7	region_southeast	0.073577
3	children	0.067390
5	region_northeast	0.005946
6	region_northwest	-0.038695
8	region_southwest	-0.043637
9	bmi_category_Underweight	-0.050599
1	Isfemale	-0.058046
10	bmi_category_Normal	-0.104042
11	bmi_category_Overweight	-0.120601

In [108…

```
cat_features = [
    'Isfemale', 'Issmoker',
    'region_northwest', 'region_southeast', 'region_southwest','region_northeast',
    'bmi_category_Normal', 'bmi_category_Overweight', 'bmi_category_Obese','bmi_category_Underweight'
]
```

In [109…

```
from scipy.stats import chi2_contingency
import pandas as pd

alpha = 0.05

cleaned_data['charges_bin'] = pd.qcut(cleaned_data['charges'], q=4, labels=False)
chi2_results = {}
```

```
for col in cat_features:
    contingency = pd.crosstab(cleaned_data[col], cleaned_data['charges_bin'])
    chi2_stat, p_val, _, _ = chi2_contingency(contingency)
    decision = 'Reject Null (Keep Feature)' if p_val < alpha else 'Accept Null (Drop Feature)'
    chi2_results[col] = {
        'chi2_statistic': chi2_stat,
        'p_value': p_val,
        'Decision': decision
    }

chi2_df = pd.DataFrame(chi2_results).T
chi2_df = chi2_df.sort_values(by='p_value')
chi2_df
```

Out[109…

	chi2_statistic	p_value	Decision
lssmoker	848.219178	0.0	Reject Null (Keep Feature)
region_southeast	15.998167	0.001135	Reject Null (Keep Feature)
lsfemale	10.258784	0.01649	Reject Null (Keep Feature)
bmi_category_Obese	8.515711	0.036473	Reject Null (Keep Feature)
region_northeast	6.438442	0.092122	Accept Null (Drop Feature)
region_southwest	5.091893	0.165191	Accept Null (Drop Feature)
bmi_category_Overweight	4.25149	0.235557	Accept Null (Drop Feature)
bmi_category_Normal	3.708088	0.29476	Accept Null (Drop Feature)
bmi_category_Underweight	3.37403	0.337471	Accept Null (Drop Feature)
region_northwest	1.13424	0.768815	Accept Null (Drop Feature)

In [ ]: