In [60]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [31]:
```
path = "C:/Users/srish/downloads/archive1/insurance.csv"
df = pd.DataFrame(pd.read_csv(path))
```

In [14]:
```
df.head()
```

Out[14]:

	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

Initial Inspection

In [16]:
```
print("Columns : ",len(df.columns),"Rows : ", len(df.iloc[:,1]))
```

```
Columns :  7 Rows :  1338
```

In [17]:
```
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 [18]:
```
df.describe()
```

Out[18]:

	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

Handling Missing Values

1. Check null values present in the dataset

In [19]:
```
df.isnull().sum()
```

Out[19]:
```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

Inference : there is no null values present in the dataset

Encoding categorical data into OneHotEncoding

In [32]:
```
df = pd.get_dummies(df,columns = ['smoker','sex','region'],drop_first = True)
```

In [33]:
```
df.head()
```

Out[33]:

	age	bmi	children	charges	smoker_yes	sex_male	region_northwest
0	19	27.900	0	16884.92400	True	False	False
1	18	33.770	1	1725.55230	False	True	False
2	28	33.000	3	4449.46200	False	True	False
3	33	22.705	0	21984.47061	False	True	True
4	32	28.880	0	3866.85520	False	True	True

Feature Engineering

In [34]:
```
def age_group(age):
    if age <= 18:
        return "Child"
    elif age <= 60:
        return "Adult"
    else:
        return "Senior"
def bmi_grp(bmi):
    if bmi < 18.5:
        return "underweight"
    elif bmi > 18.5 and bmi < 24.9:
        return "normal"
    elif bmi > 25.0 and bmi < 29.9:
        return "overweight"
    elif bmi > 30.0 and bmi < 34.9:
        return "obese"
    else:
        return "extremely obese"
df['bmi_encoded'] = df['bmi'].apply(bmi_grp).map({'underweight' : 0,'normal' :
df['age_grp_encoded'] = df['age'].apply(age_group).map({'Child' : 0,"Adult" :
```

Final

In [39]:
```
df.head(10)
```

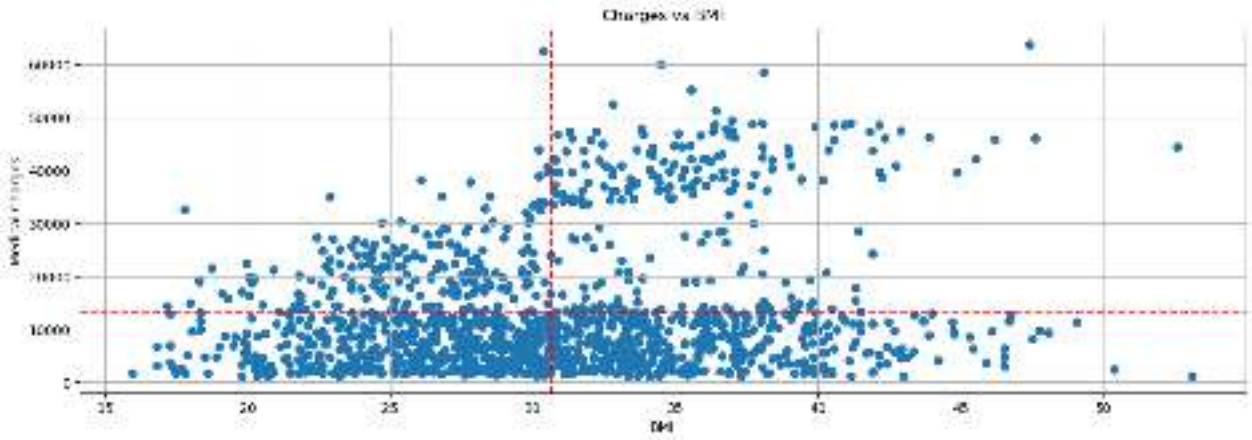
Out[39]:

	age	bmi	children	charges	smoker_yes	sex_male	region_northwest
0	19	27.900	0	16884.92400	True	False	False
1	18	33.770	1	1725.55230	False	True	False
2	28	33.000	3	4449.46200	False	True	False
3	33	22.705	0	21984.47061	False	True	True
4	32	28.880	0	3866.85520	False	True	True
5	31	25.740	0	3756.62160	False	False	False
6	46	33.440	1	8240.58960	False	False	False
7	37	27.740	3	7281.50560	False	False	True
8	37	29.830	2	6406.41070	False	True	False
9	60	25.840	0	28923.13692	False	False	True

EDA

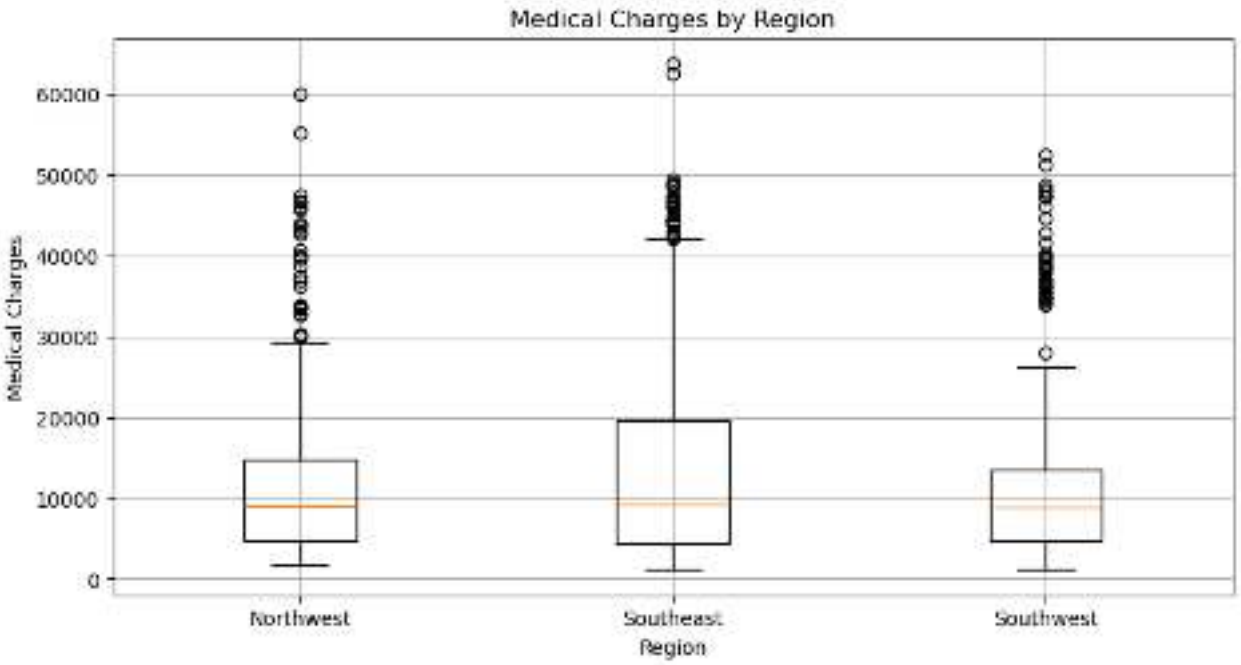
In [57]:
```
#bmi vs charges
```

```
plt.figure(figsize = (16,5))
plt.scatter(df['bmi'], df['charges'])
plt.axhline(df['charges'].mean(),color = 'red',linestyle = '--')
plt.axvline(df['bmi'].mean(),color = 'red',linestyle = '--')
plt.xlabel('BMI')
plt.ylabel('Medical Charges')
plt.title('Charges vs BMI')
plt.grid()
plt.show()
```



```
<Figure size 640x480 with 0 Axes>
```

In [61]:
```
regions = ['region_northwest', 'region_southeast', 'region_southwest']
region_charges = [
    df[df['region_northwest'] == True]['charges'],
    df[df['region_southeast'] == True]['charges'],
    df[df['region_southwest'] == True]['charges']
]
plt.figure(figsize=(10, 5))
plt.boxplot(region_charges, labels=['Northwest', 'Southeast', 'Southwest'])
plt.xlabel('Region')
plt.ylabel('Medical Charges')
plt.title('Medical Charges by Region')
plt.grid()
plt.show()
```



Inference :

bmi vs charges graph shows a non-linear realtionshop between BMI and medical charges and the graph below the distribution of charges and median charges in different regions.

What does one row in your dataset represent in the real world?

Ans - One row represents the details(age,sex,region), lifestyle factors and health indicators.

Which column in your dataset is most useful for decision-making and why?

Ans - smoker (yes/no) as it significantly effects the health of the individual and finally impacting the net medical charges of the individuals.

Which column would you remove before ML modeling? Justify your choice.

Ans - raw categorical columns may be removed after encoding.

In [66]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 11 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   age               1338 non-null   int64
 1   bmi               1338 non-null   float64
 2   children          1338 non-null   int64
 3   charges           1338 non-null   float64
 4   smoker_yes        1338 non-null   bool
 5   sex_male          1338 non-null   bool
 6   region_northwest  1338 non-null   bool
 7   region_southeast  1338 non-null   bool
 8   region_southwest  1338 non-null   bool
 9   bmi_encoded       1338 non-null   int64
 10  age_grp_encoded   1338 non-null   int64
dtypes: bool(5), float64(2), int64(4)
memory usage: 69.4 KB
```

What type of bias might exist in your dataset?

Ans - It has lifetsyle bias as it assumes everyone and uniform access to healthcare and insurance across regions.

Is your dataset more suitable for classification or regression? Why?

Ans - The dataset is more suitable for regression as the taget values are continuous numerical value which makes the main purpose to predict an exact insurance cost.

In [ ]: