# Project to analyze the Insurace dataset to create a model that will predict the cost of medical insurance based on various input features

# Importing Libraries

In [1]:
```
import pandas as pd
```

In [2]:
```
import numpy as np
```

In [3]:
```
import seaborn as sns
```

In [4]:
```
import matplotlib.pyplot as plt
```

In [62]:
```
df = pd.read_csv('Downloads/Python without IBM 23.09.2024/insurance.csv')
df.head()
```

Out[62]:

| | 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 |

Imported All the libraries like Pandas, Numpy, Seaborn & matplot that will use in the Project of Insurance Data loaded as df

# 2. Check the shape of the data along with the data types of the column

In [8]:
```
shape = df.shape
print('df shape=' ,shape)
```

```
df shape= (1338, 7)
```

In [9]:
```
type = df.dtypes
print('df type=', type)
```

```
df type= age           int64
sex          object
bmi         float64
children      int64
smoker       object
region       object
charges     float64
dtype: object
```

In [10]:
```
null_values = df.isnull().sum()
null_values
```

Out[10]:
```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

In [69]:
```
df_stat = df.describe()
df_stat
```

Out[69]:

| | 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 |

# OBSERVATION :-

1. Insurance data (df) has 1338 Nos. of Rows & 7 Nos. of Coloumn. 2. In Insurance data (df) the columns sex, smoker, region has categorical values while age, bmi, children, charges has numerical values.

# 3. Check missing values in the dataset and find the appropriate measures to fill in the missing values

In [74]:
```
df_inf= df.info()
df_inf
```

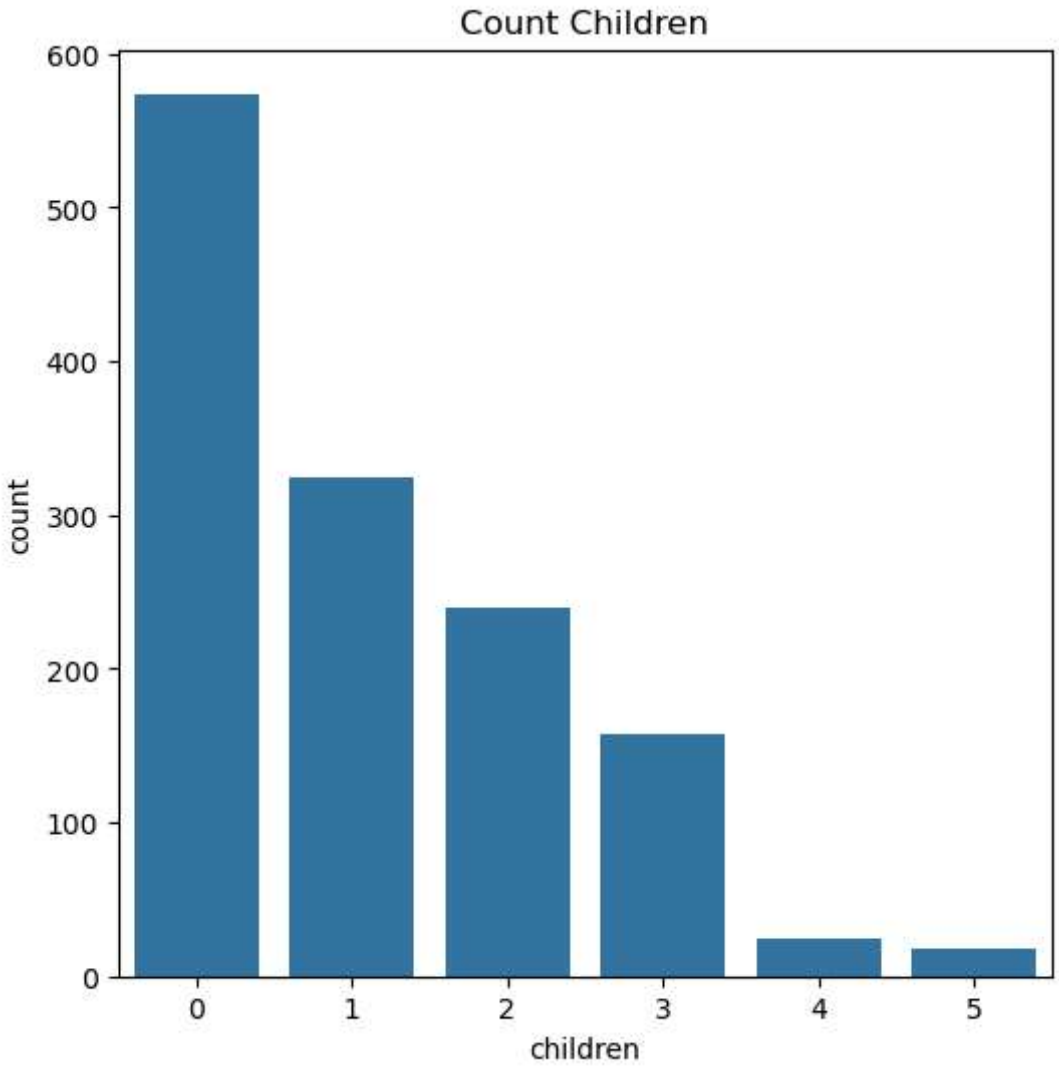
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1338 non-null   int64  
 1   sex       1338 non-null   object 
 2   bmi       1338 non-null   float64
 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
```

# OBSERVATION :-

Insurance data (df) does not consist any null value.

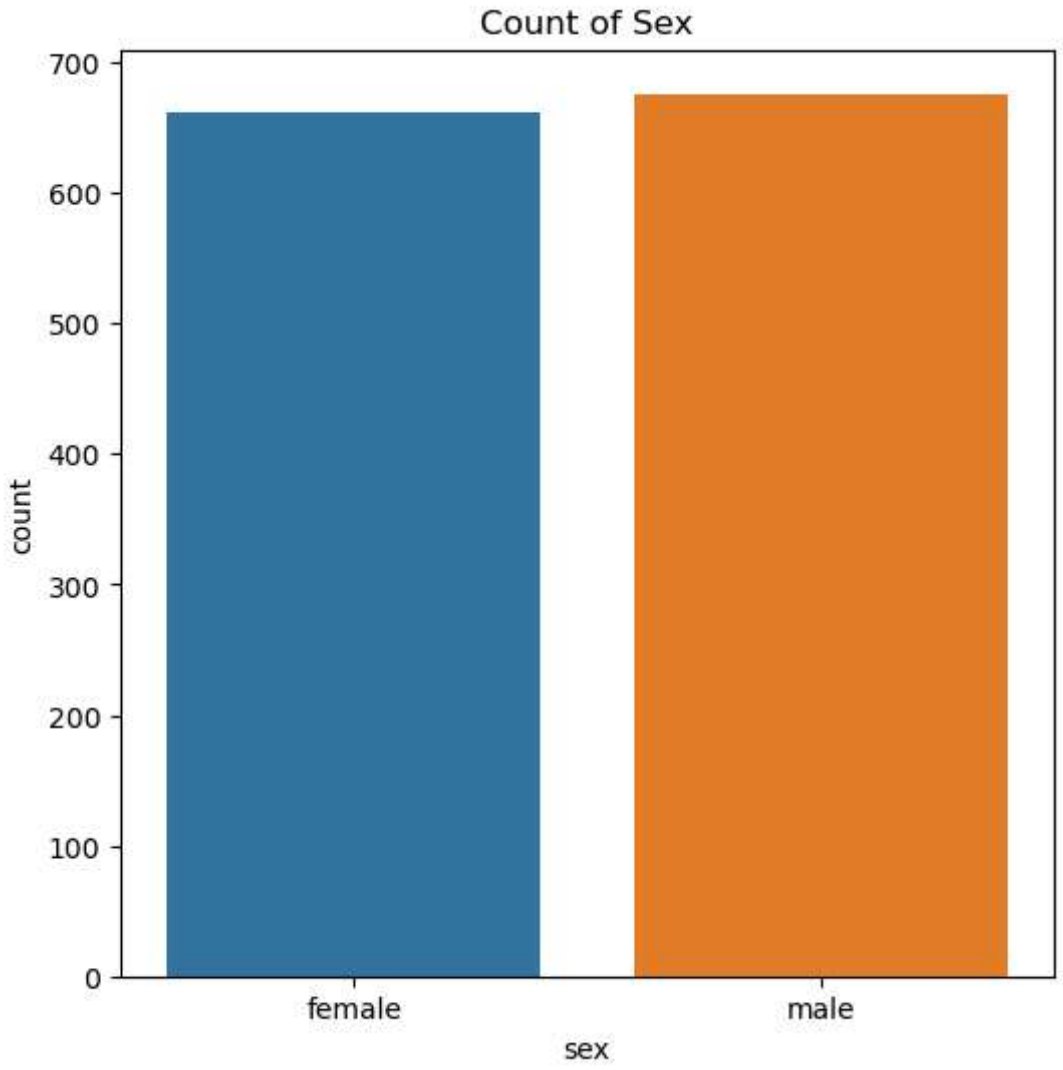
# 4. Explore the relationship between the feature and target column using a count plot of categorical columns and a scatter plot of numerical columns

In [85]:
```
plt.figure(figsize=(6,6))
sns.countplot(x ='children', data =df)
plt.title('Count Children')
plt.show()
df['children'].value_counts()
```



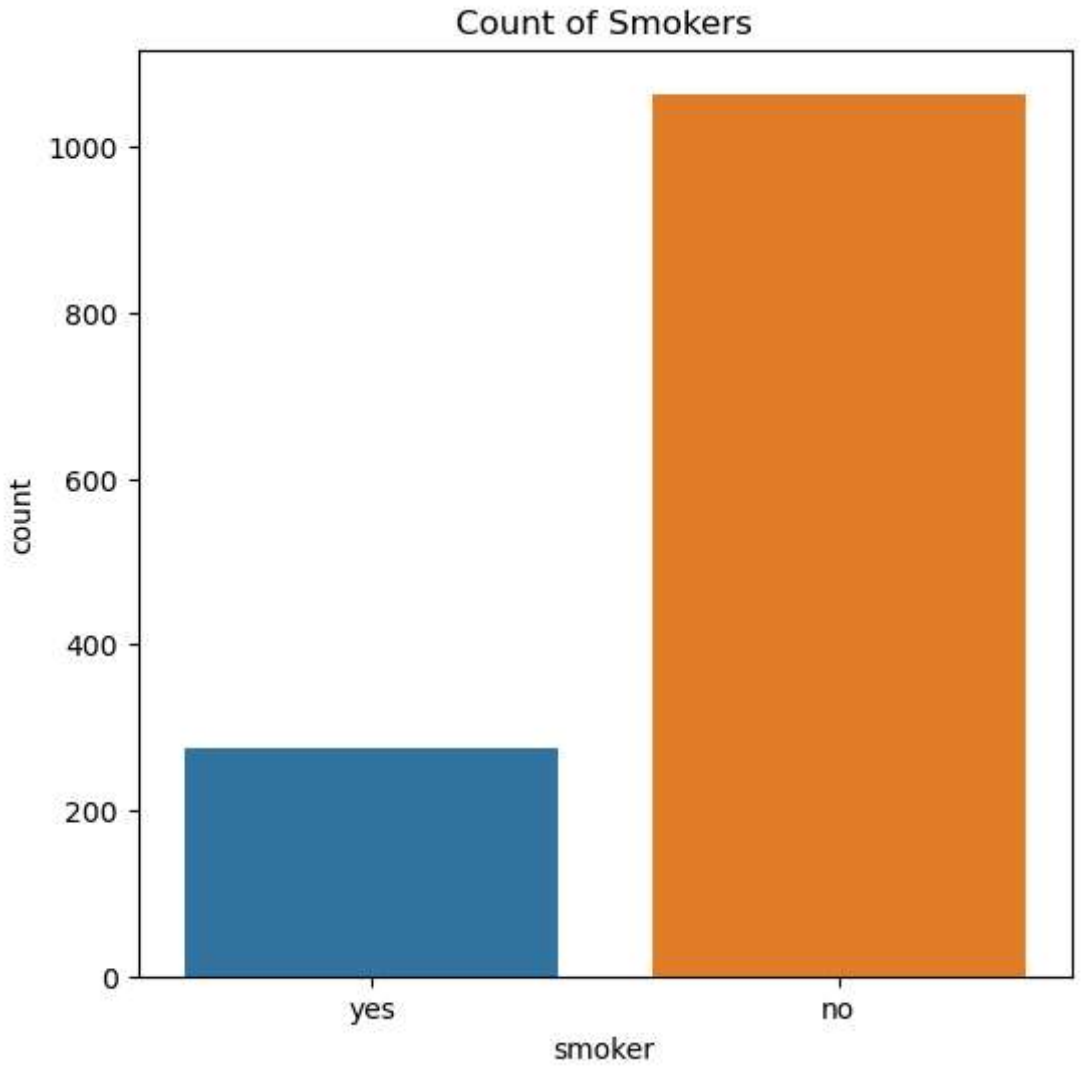

Out[85]:
```
children
0    574
1    324
2    240
3    157
4     25
5     18
Name: count, dtype: int64
```

In [26]:
```
plt.figure(figsize=(6,6))
sns.countplot(x='sex',data =df, hue ='sex')
plt.title('Count of Sex')
plt.show()
df['sex'].value_counts()
```



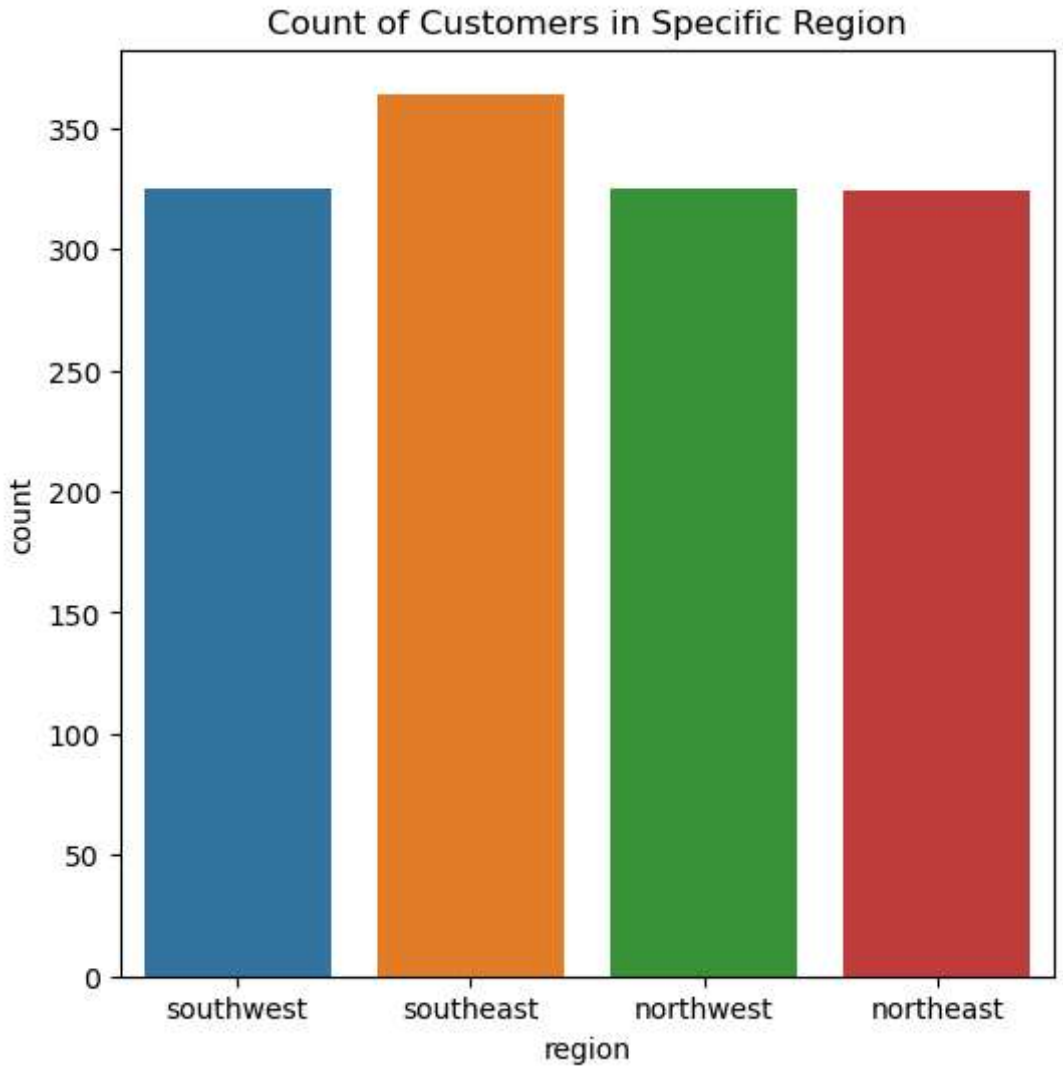

Out[26]:
```
sex
male      676
female    662
Name: count, dtype: int64
```

In [87]:
```
plt.figure(figsize=(6,6))
sns.countplot(x ='smoker', data =df, hue ='smoker')
plt.title('Count of Smokers')
plt.show()
df['smoker'].value_counts()
```




Out[87]:
```
smoker
no     1064
yes     274
Name: count, dtype: int64
```

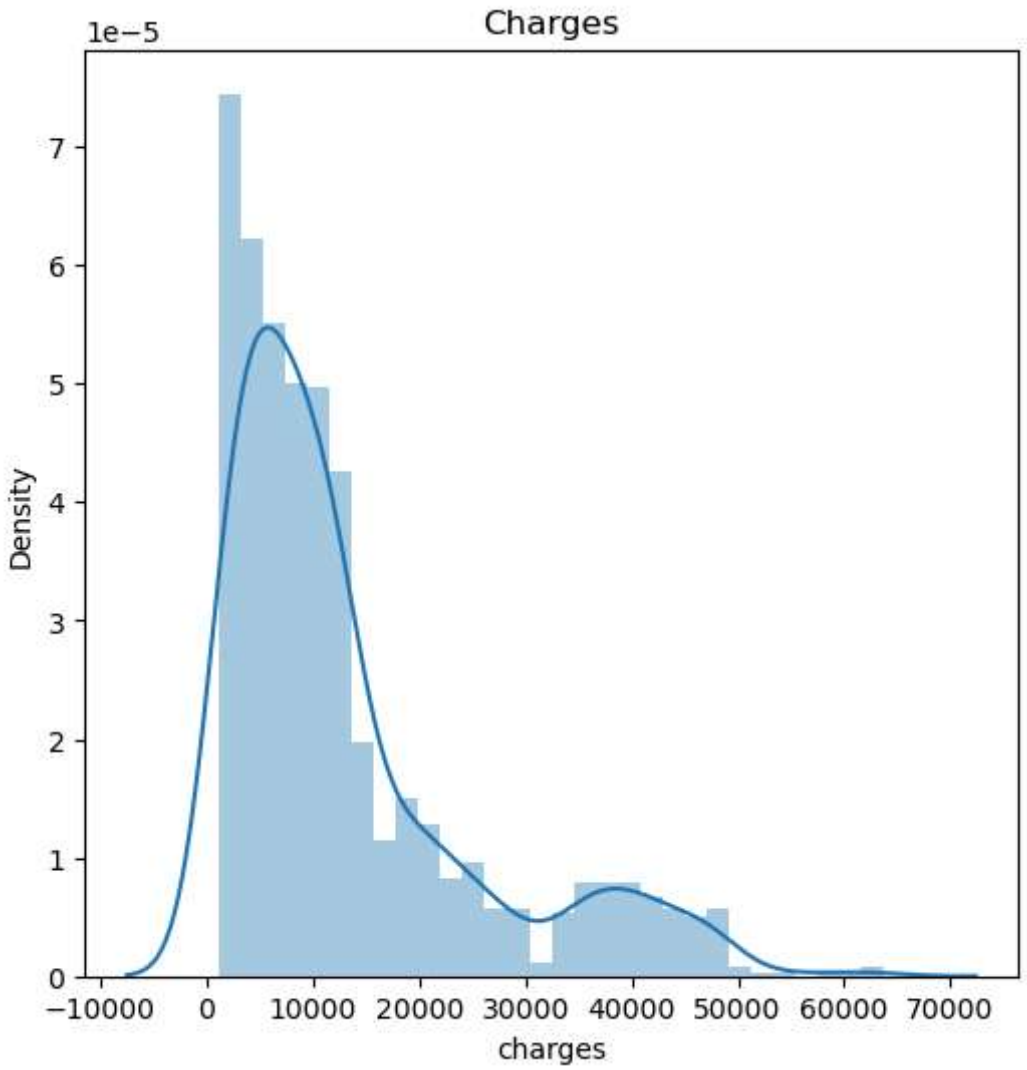
In [91]:
```
plt.figure(figsize=(6,6))
sns.countplot(x ='region', data =df, hue ='region')
plt.title('Count of Customers in Specific Region')
plt.show()
df['region'].value_counts()
```



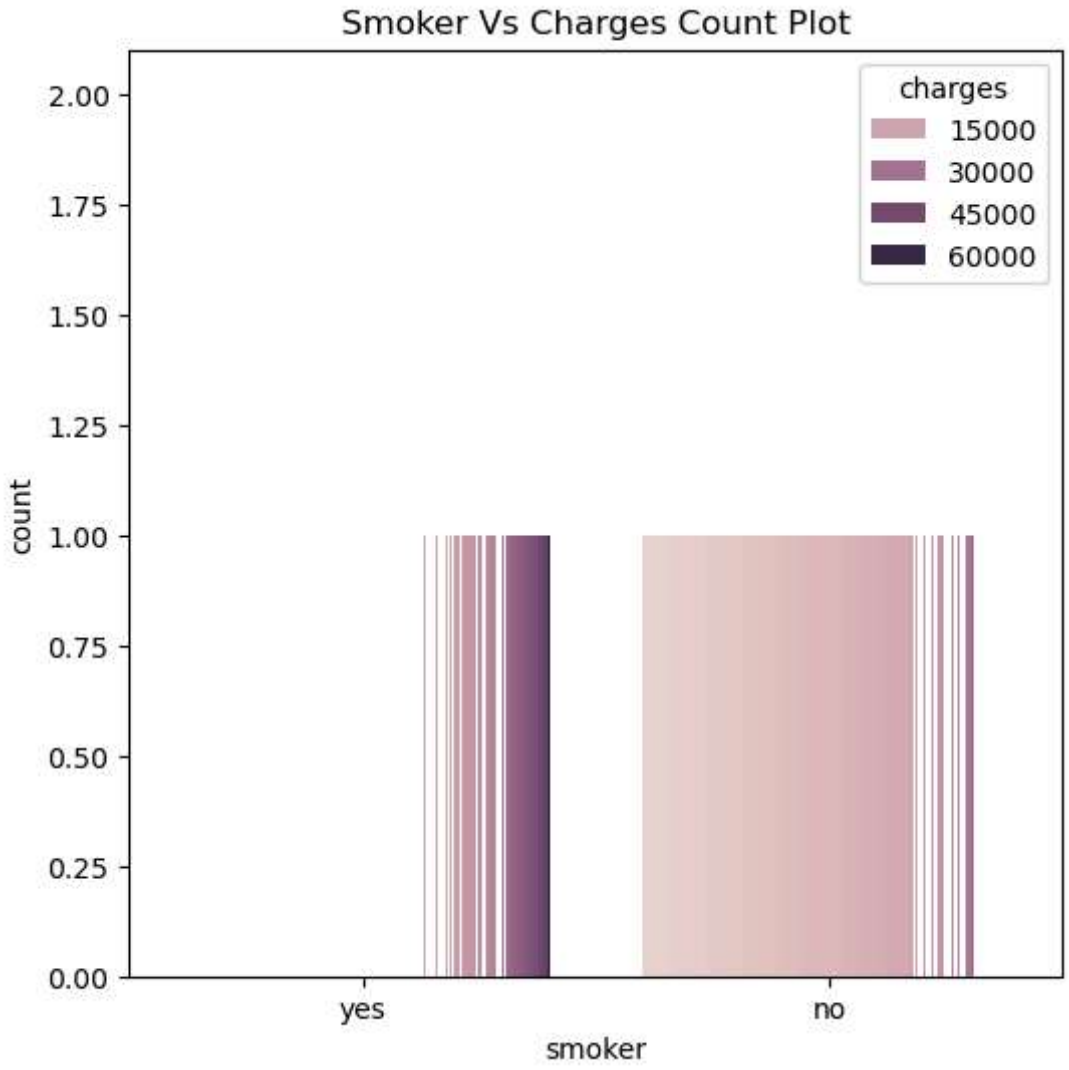

Out[91]:
```
region
southeast    364
southwest    325
northwest    325
northeast    324
Name: count, dtype: int64
```

In [30]:
```
plt.figure(figsize=(6,6))
sns.distplot(df ['charges'])
plt.title('Charges')
plt.show()
```

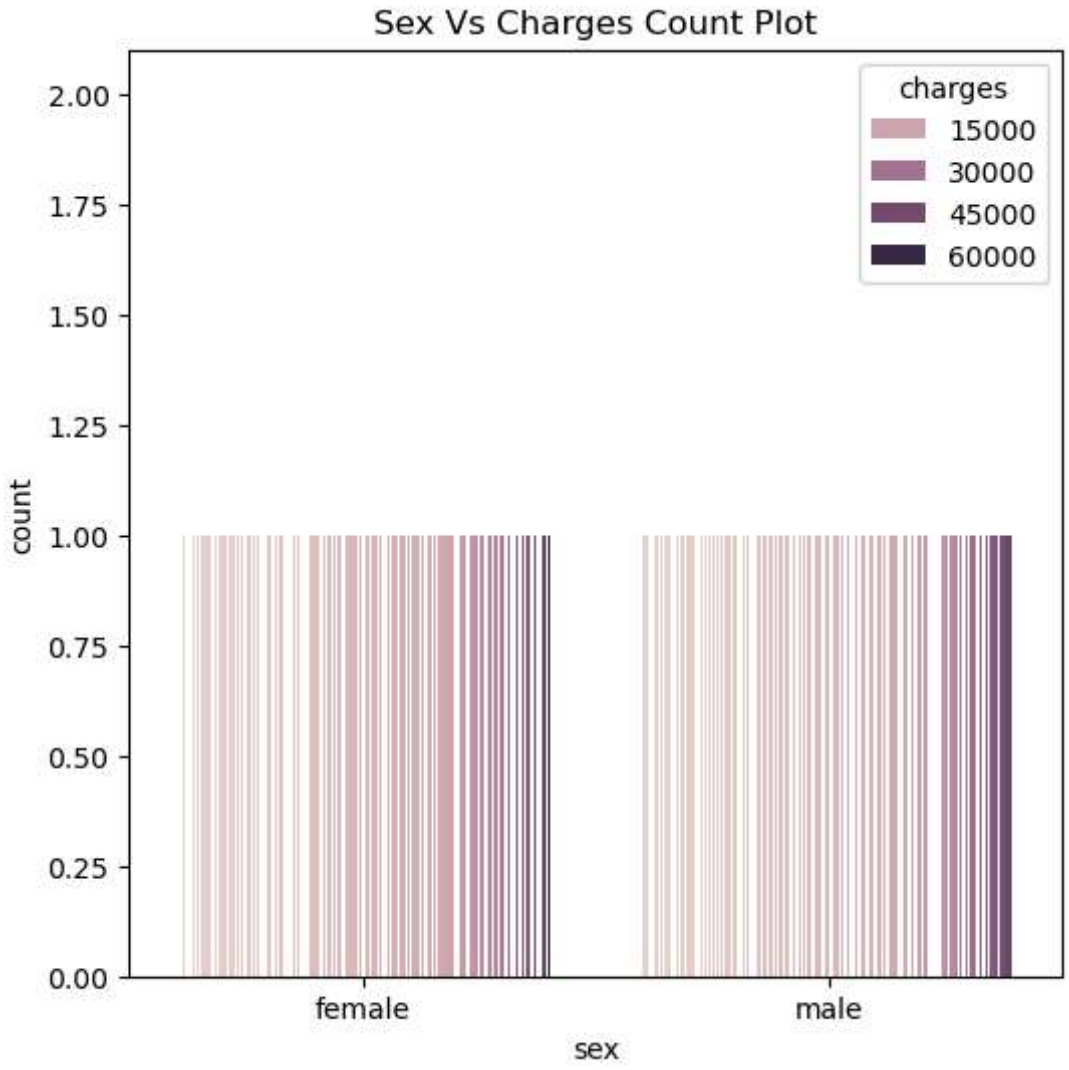
```
C:\Users\HP\AppData\Local\Temp\ipykernel_7052\2639158157.py:2: UserWarning:

`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(df ['charges'])
```



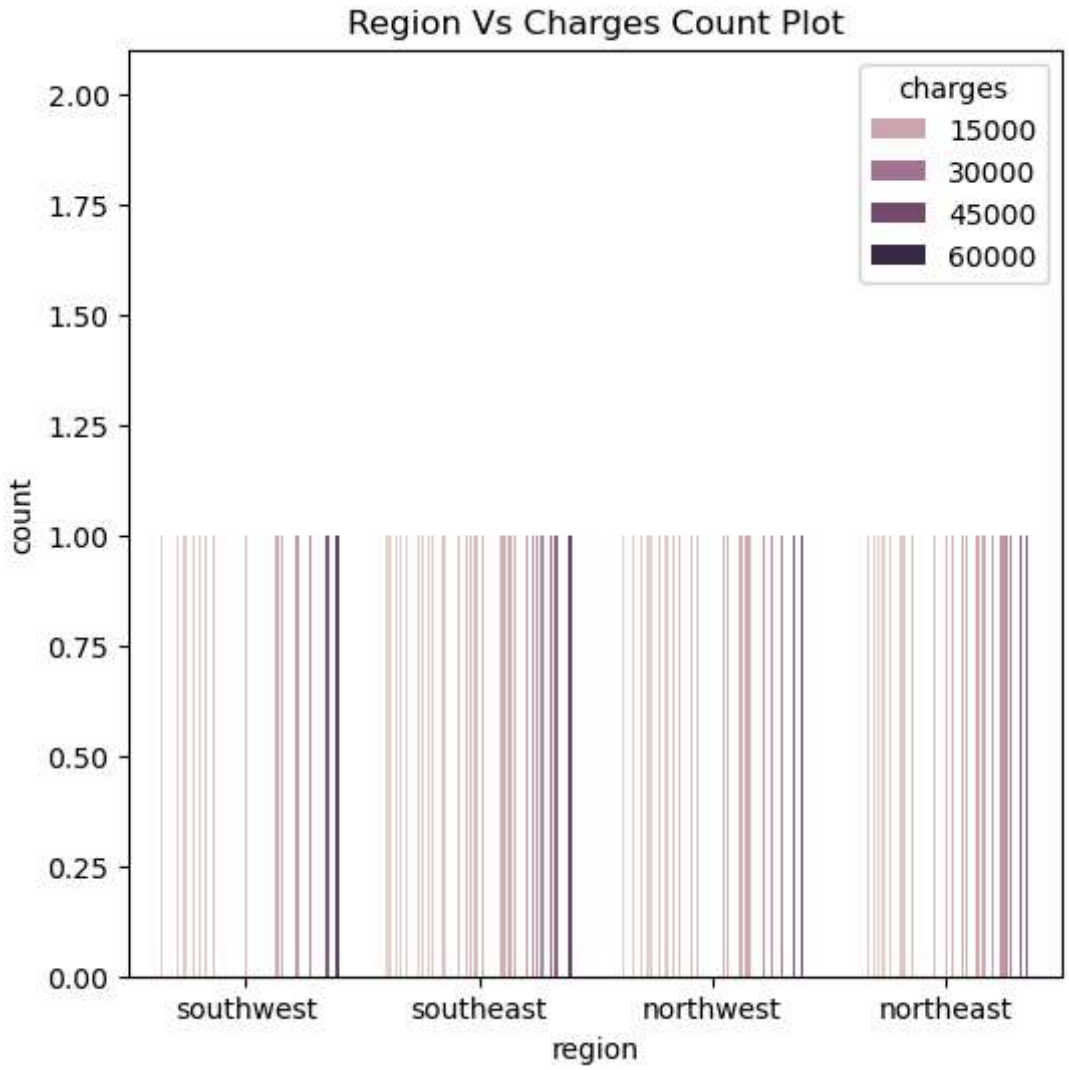

In [31]:
```
# 4
plt.figure (figsize =(6,6))
sns.countplot(data =df, x ='smoker', hue ='charges')
plt.title('Smoker Vs Charges Count Plot')
plt.show()
```




In [32]:
```
plt.figure(figsize =(6,6))
sns.countplot(data = df, x='sex', hue='charges')
plt.title('Sex Vs Charges Count Plot')
plt.show()
```

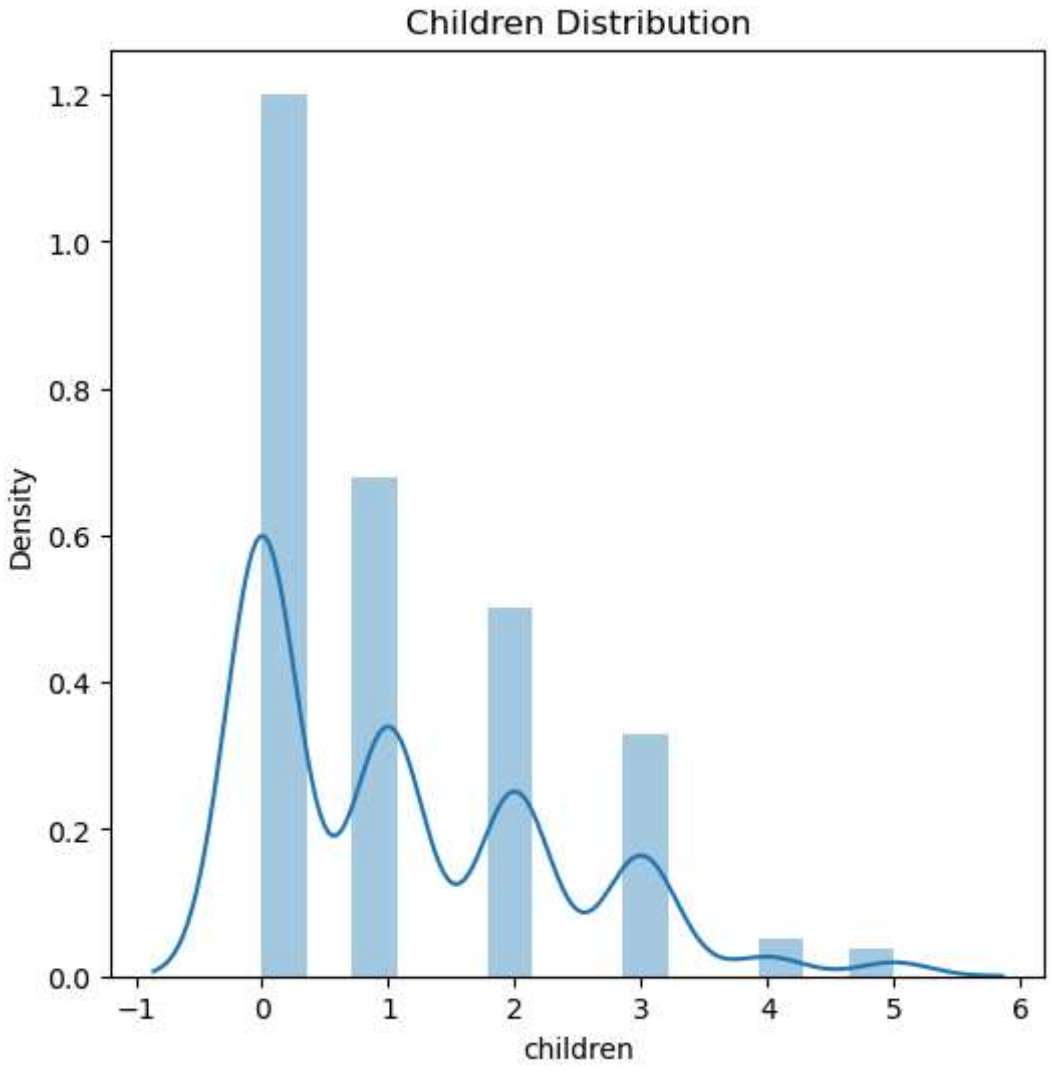



In [35]:
```
plt.figure(figsize=(6,6))
sns.countplot(data =df, x='region', hue='charges')
plt.title('Region Vs Charges Count Plot')
plt.show()
```



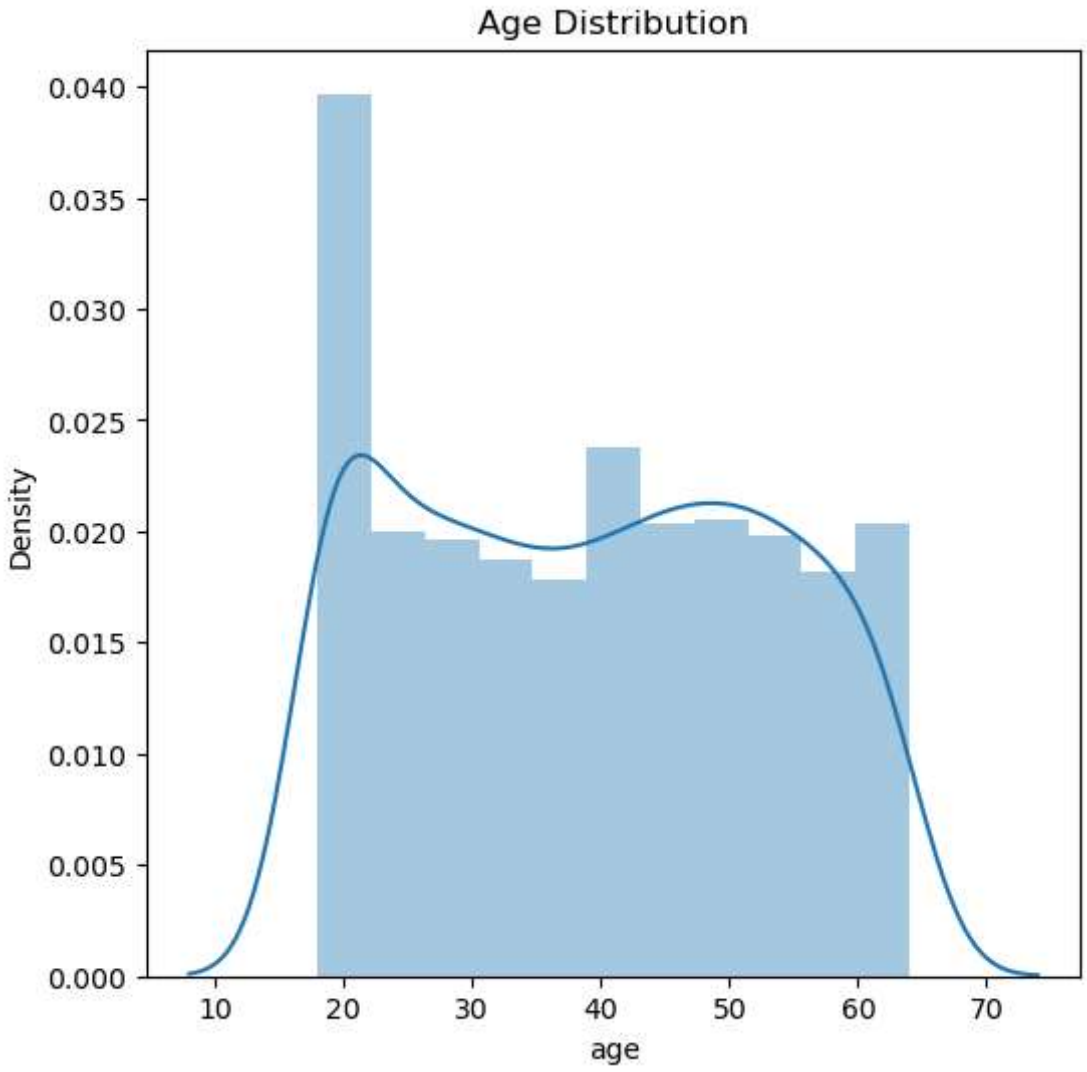

In [83]:
```
plt.figure(figsize=(6,6))
sns.distplot(df['children'])
plt.title('Children Distribution')
plt.show()
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_7052\3690289693.py: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(df['children'])
```



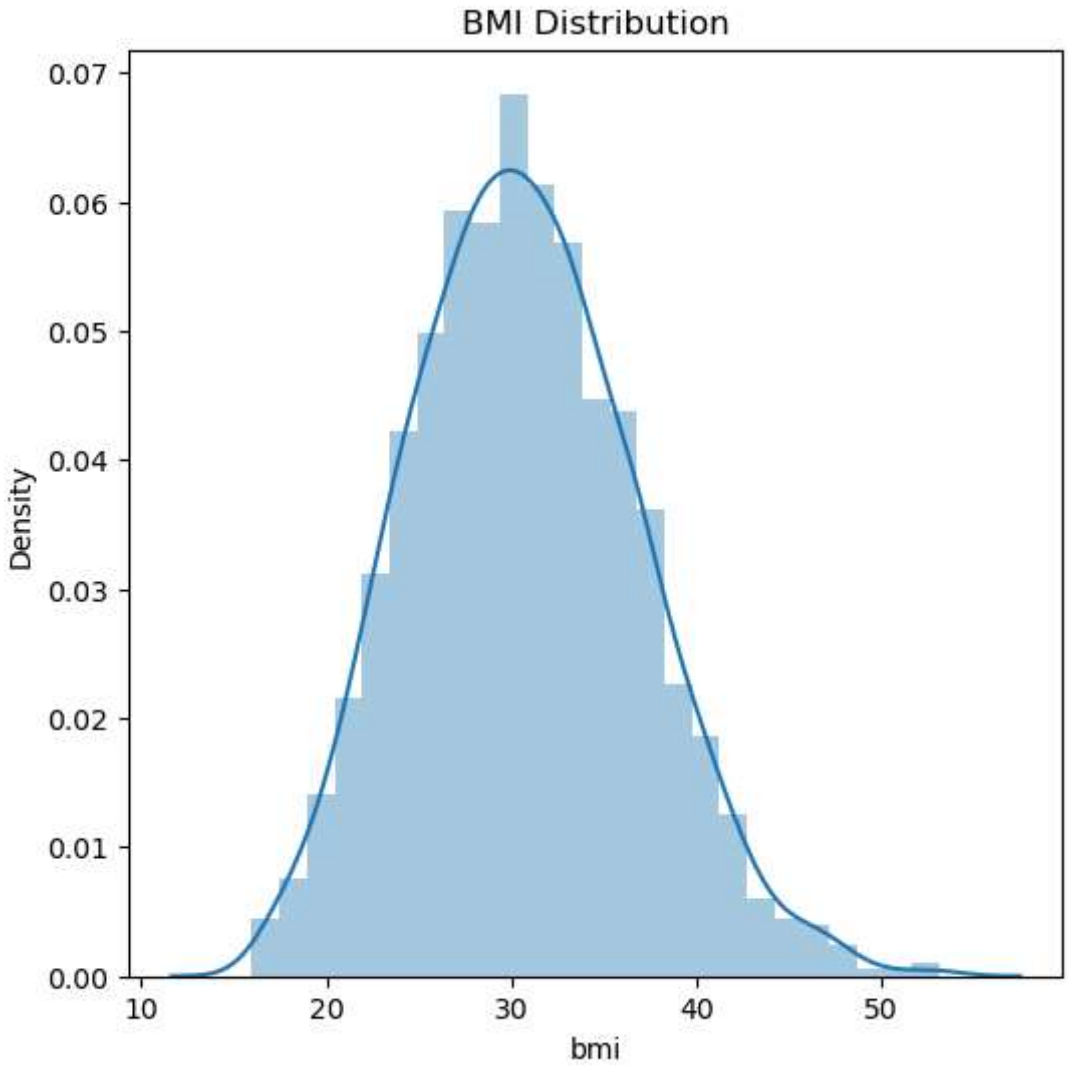

In [77]:
```
plt.figure(figsize =(6,6))
sns.distplot(df['age'])
plt.title('Age Distribution')
plt.show()
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_7052\755584957.py: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(df['age'])
```



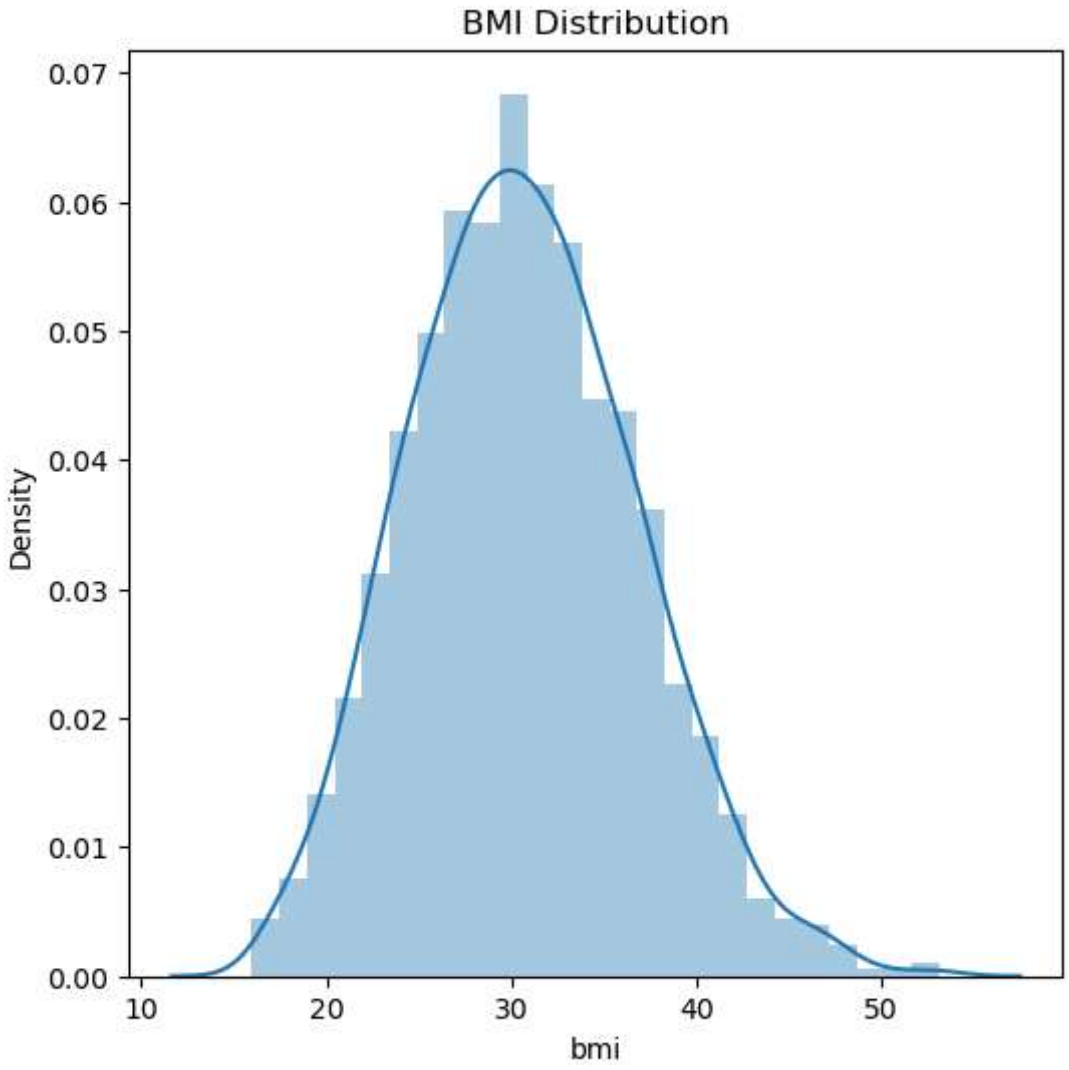

In [79]:
```
plt.figure(figsize=(6,6))
sns.distplot(df['bmi'])
plt.title('BMI Distribution')
plt.show()
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_7052\3969024274.py: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(df['bmi'])
```




In [81]:
```
plt.figure(figsize=(6,6))
sns.distplot(df['bmi'])
plt.title('BMI Distribution')
plt.show()
```

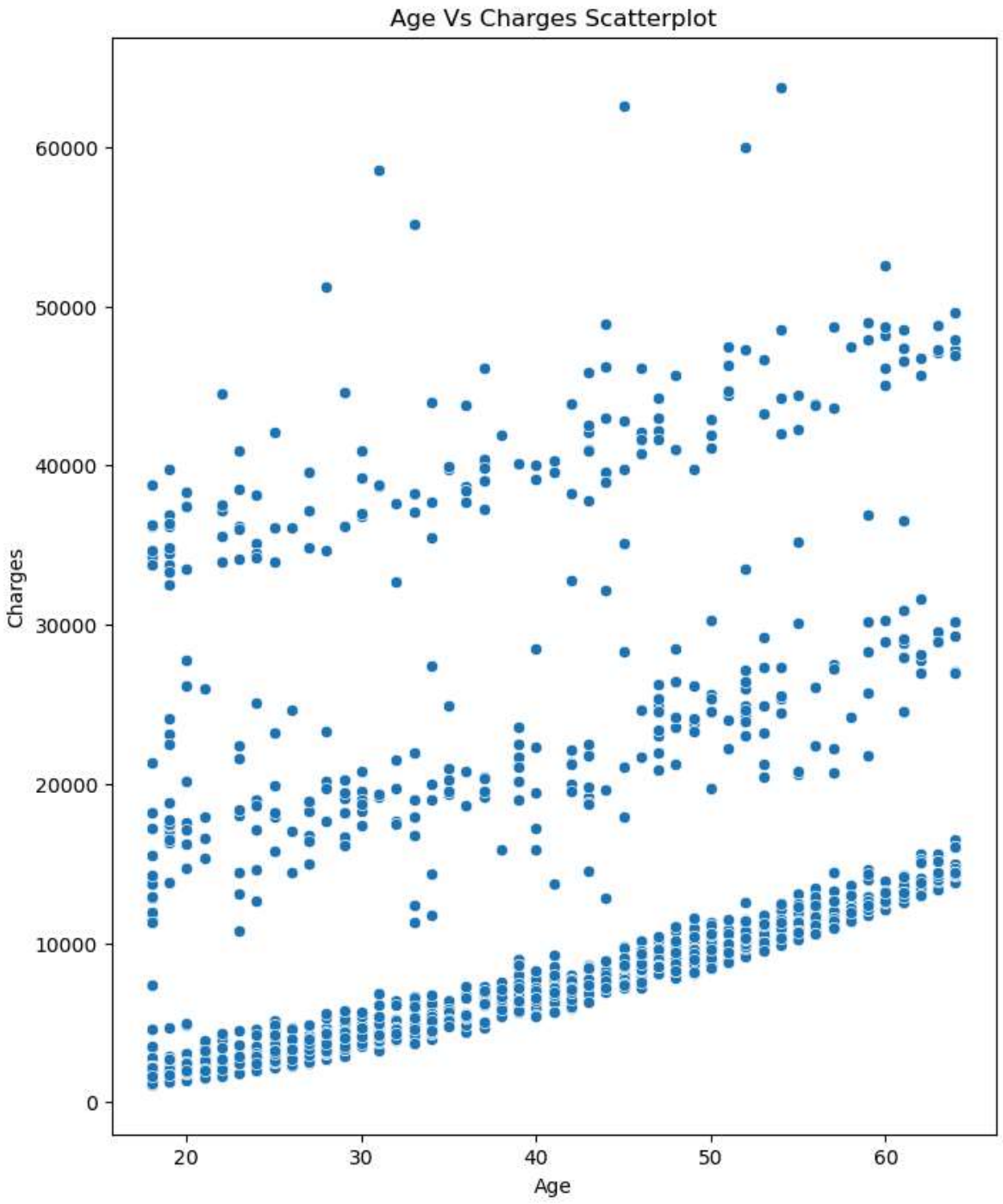
```
C:\Users\HP\AppData\Local\Temp\ipykernel_7052\3969024274.py:2: UserWarning:

`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(df['bmi'])
```



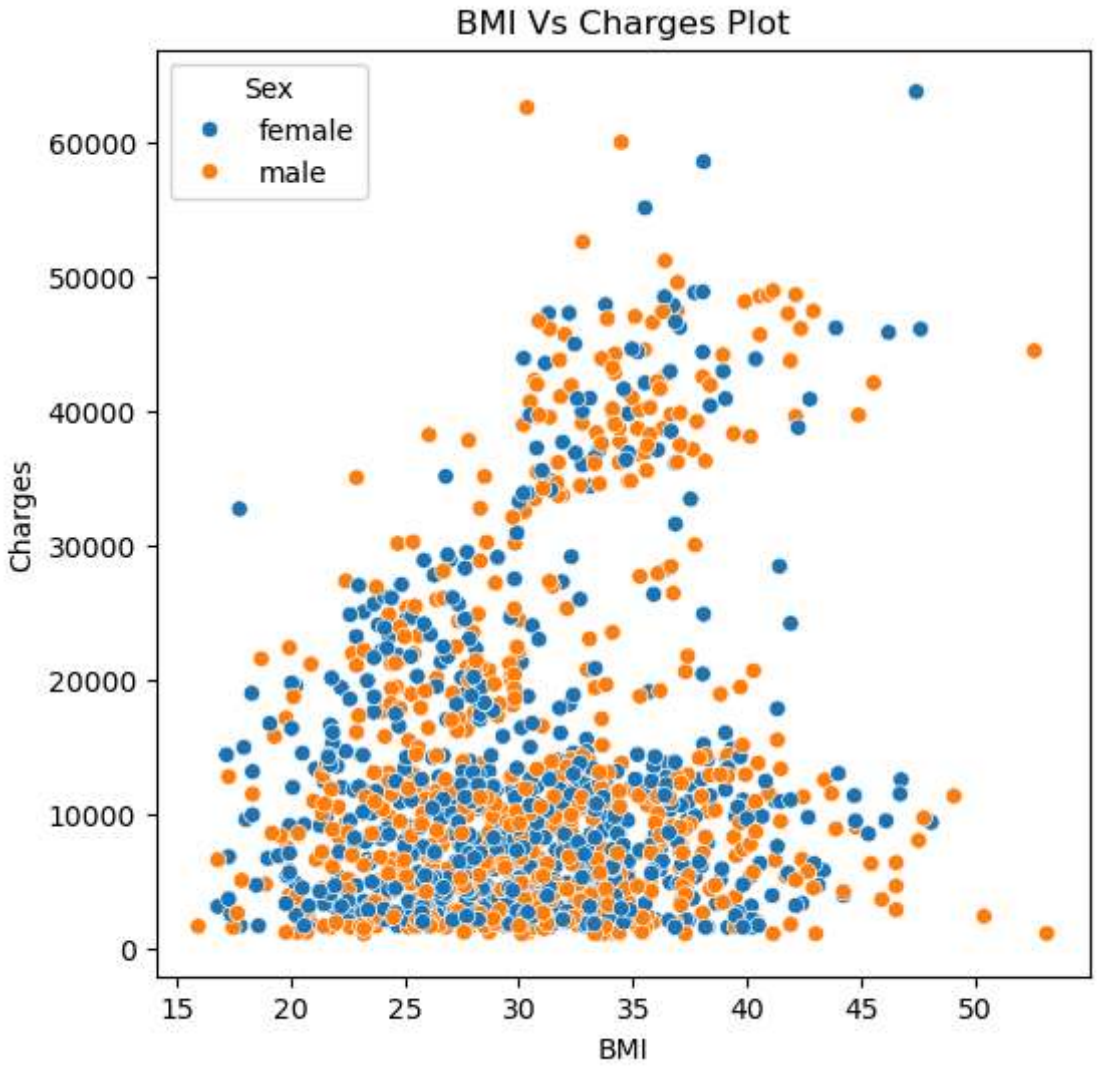

# OBSERVATIONS :-

1.Count Children : Provide a view of individuals count based on the number of children they have. 2.Count of Sex : Visualizes that in insured individuals No. of males are more than female. 3.Count of Smokers : Visualizes that participation of policy holding by Smokers is more than Non-smokers. 4.Count of Customers in Specific Region: Visualizes that No. of Policy holders are more in Southeast region compare to other three region. 5.Smoker Vs Charges Count Plot : Visualizes that Somewhere Premium Charges are high for Smokers compare to Non-smokers. 6.Sex Vs Charges Count Plot : Predicting that Somewhere Premium Charges are high for both male & female, but qty. of males are high to pay high Premium Charges compare to female. 7.Region Vs Charges Count Plot Shows : Visualizing that Premium Charges are high for Southwest & Southeast Region compare to other two.

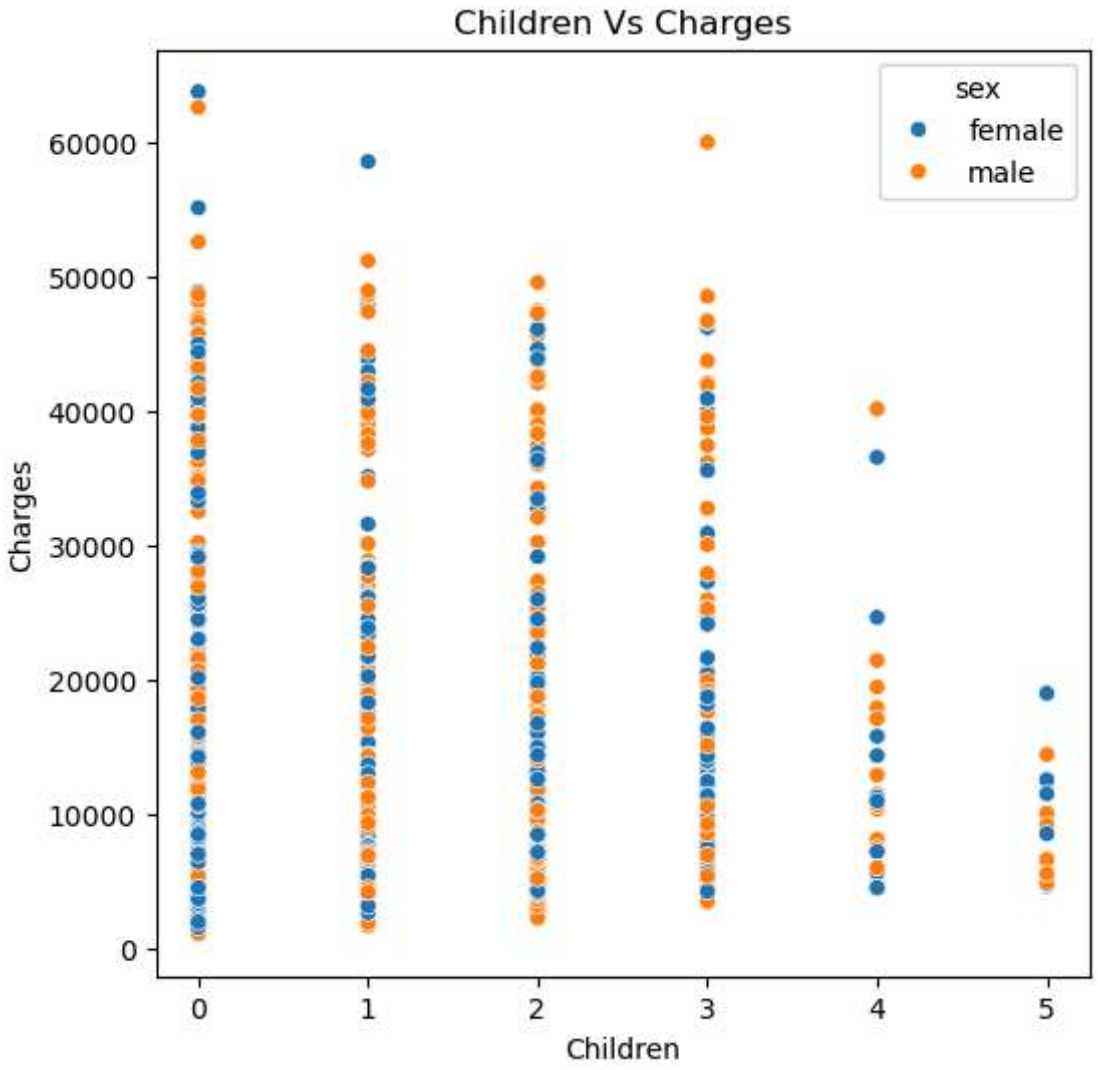
In [37]:
```
plt.figure(figsize=(8, 10))
sns.scatterplot(data=df, x='age', y='charges')
plt.title('Age Vs Charges Scatterplot')
plt.xlabel('Age')
plt.ylabel('Charges')
plt.show()
```




In [39]:
```
plt.figure(figsize=(6,6))
sns.scatterplot(x='bmi', y ='charges', data =df,hue ='sex')
plt.title('BMI Vs Charges Plot')
plt.xlabel('BMI')
plt.ylabel('Charges')
plt.legend(title ='Sex')
plt.show()
```




In [41]:
```
plt.figure(figsize=(6,6))
sns.scatterplot(x='children', y='charges', data= df, hue ='sex')
plt.title('Children Vs Charges')
plt.xlabel ('Children')
plt.ylabel('Charges')
plt.show()
```



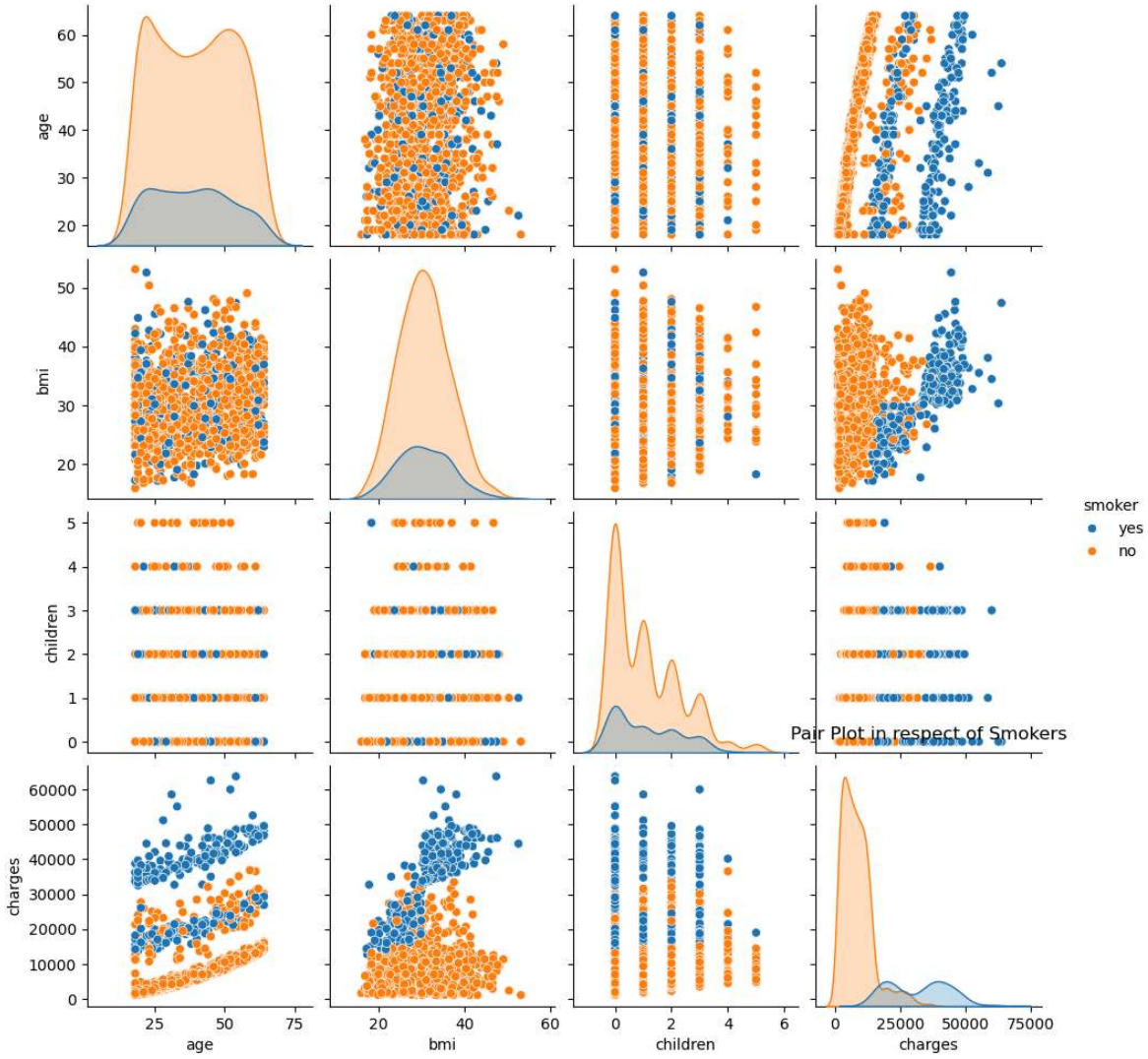

# OBSERVATION :-

Scatter Plots : All three Scatter Plots Show the Co-relation among the follwoing : 1. Age Vs Charges 2. BMI Vs Charges, based on sex 3. Children Vs Charges, based on sex

# 5. Perform data visualization using plots of feature vs feature

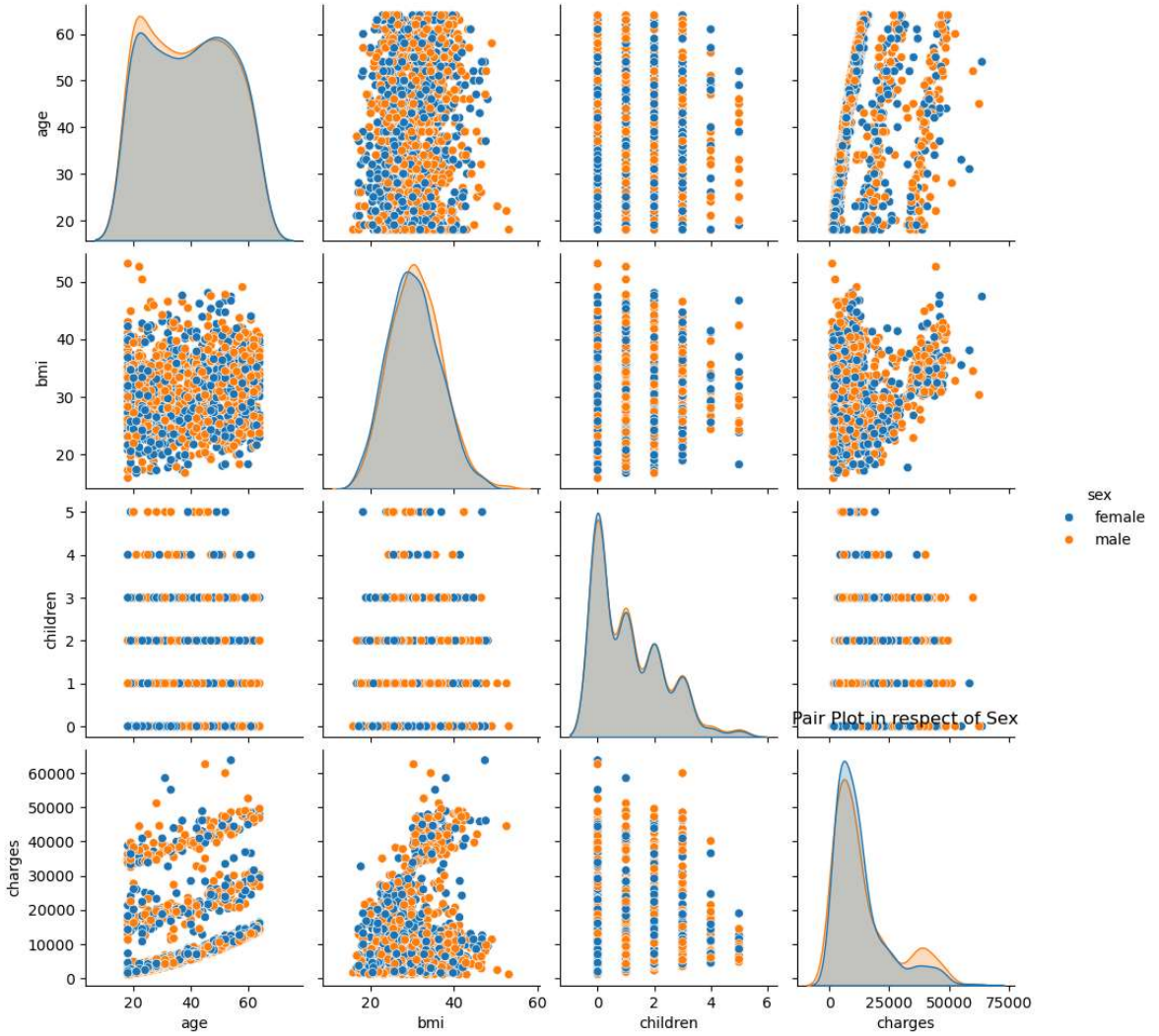
In [43]:
```
plt.figure(figsize=(6,6))
sns.pairplot(data =df, hue ='smoker')
plt.title('Pair Plot in respect of Smokers')
plt.show()
```

```
<Figure size 600x600 with 0 Axes>
```



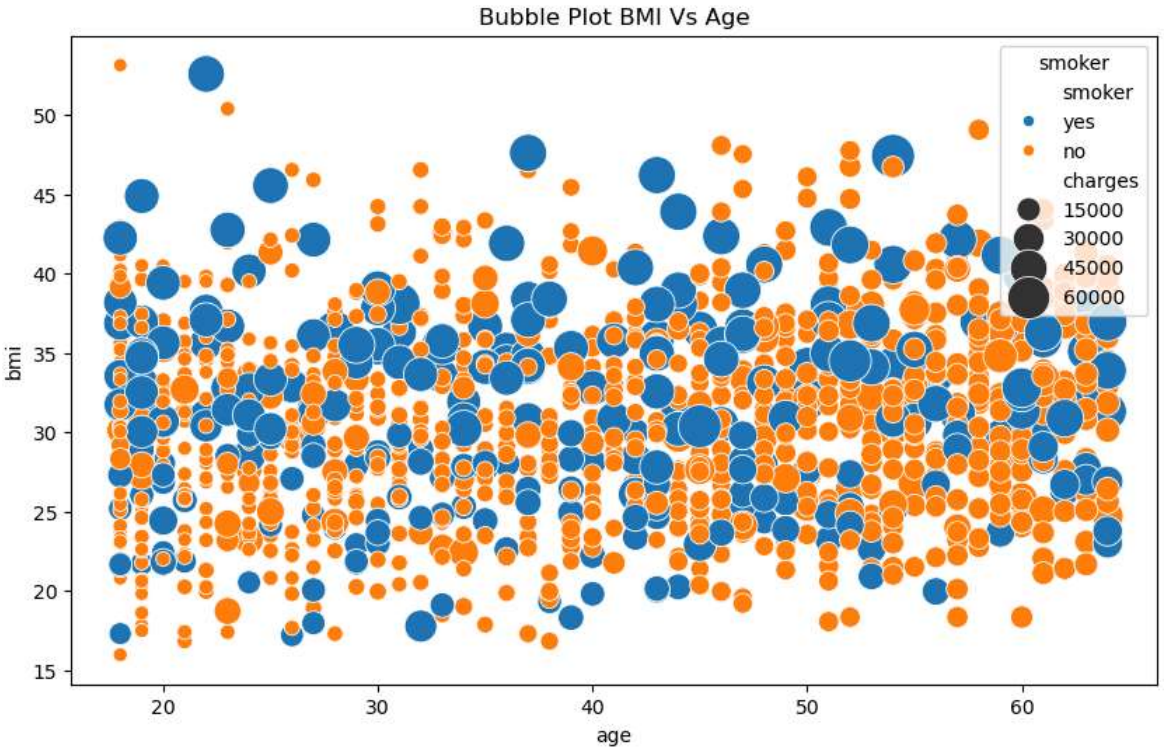

In [44]:
```
plt.figure(figsize=(6,6))
sns.pairplot(data =df, hue ='sex')
plt.title('Pair Plot in respect of Sex')
plt.show()
```

```
<Figure size 600x600 with 0 Axes>
```

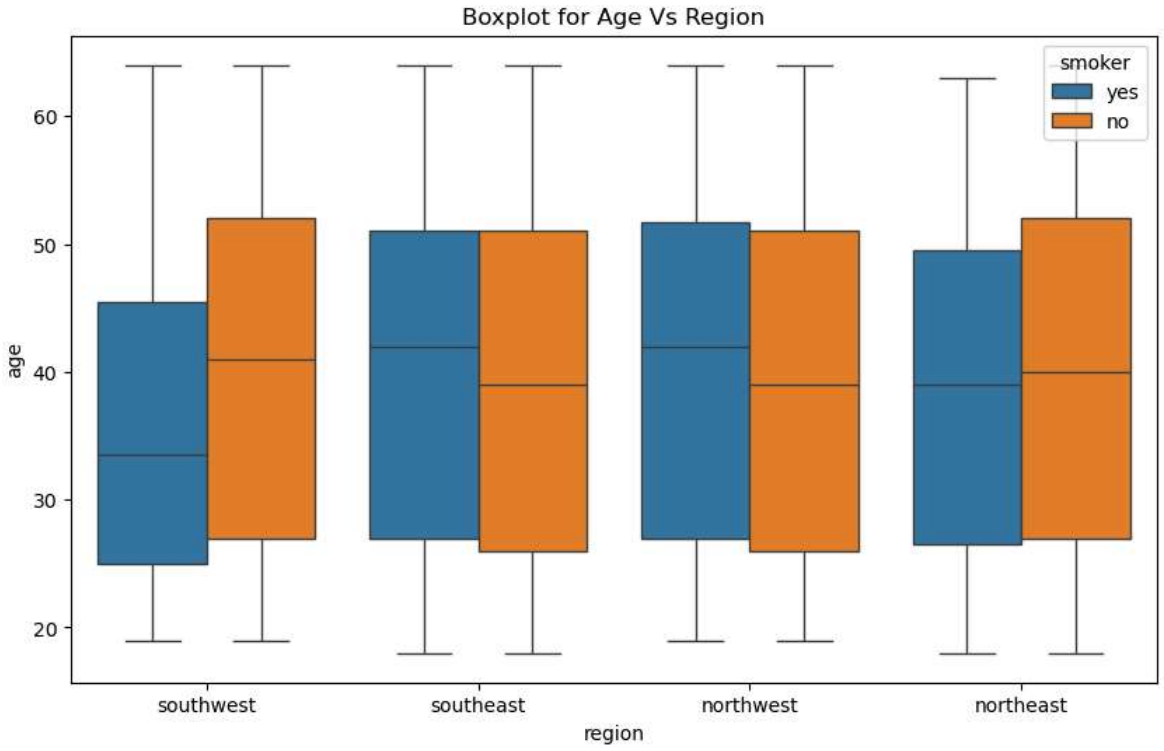



In [103…

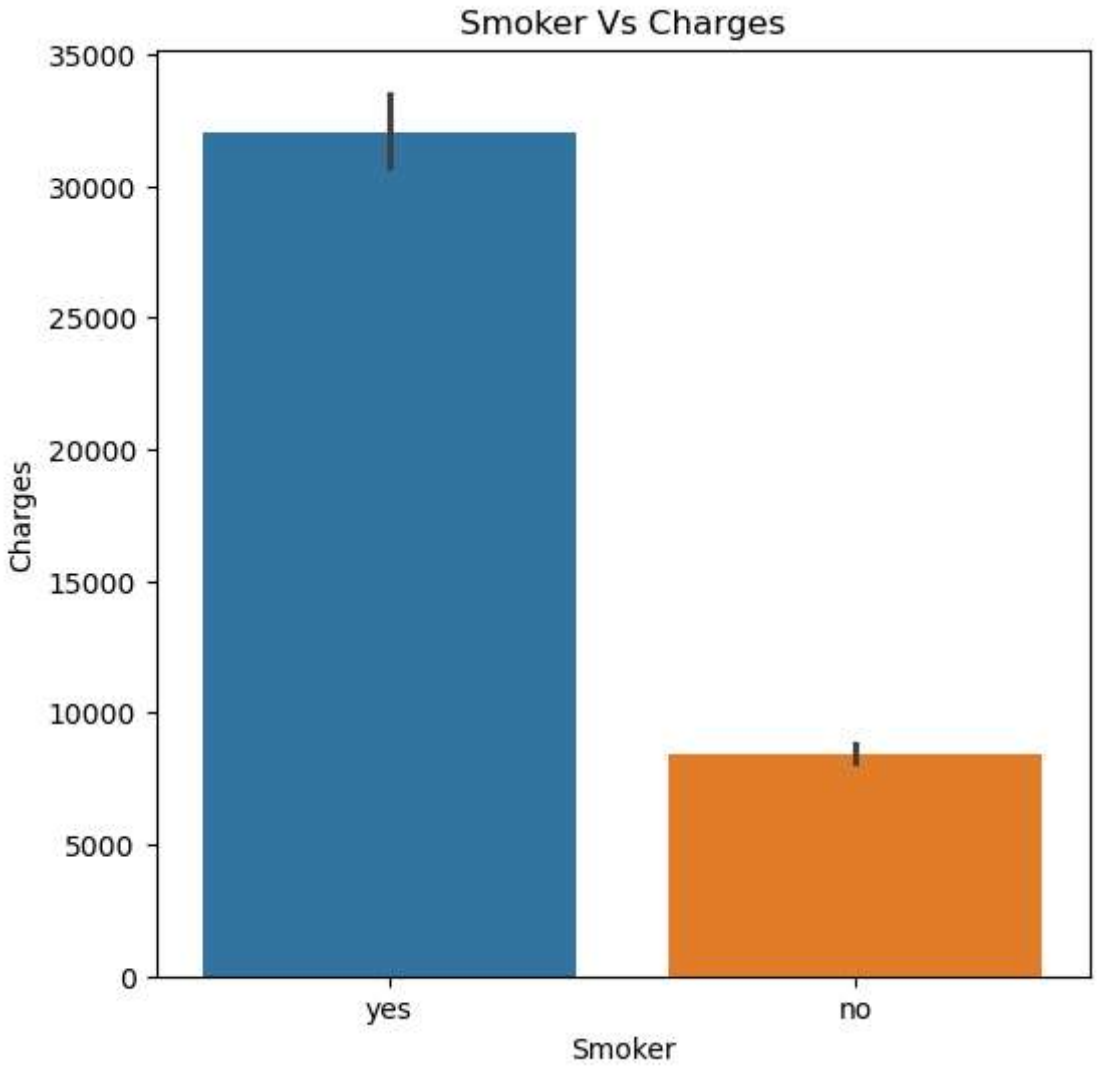
```
plt.figure(figsize=(10,6))
sns.scatterplot(data =df ,x='age', y='bmi', size='charges', sizes =(50,500), hue
plt.title('Bubble Plot BMI Vs Age')
plt.legend(title ='smoker', loc='upper right')
plt.show()
```




In [98]:
```
plt.figure(figsize=(10,6))
sns.boxplot(data =df, x='region', y='age', hue ='smoker')
plt.legend(title ='smoker', loc='upper right')
plt.title('Boxplot for Age Vs Region')
plt.show()
```

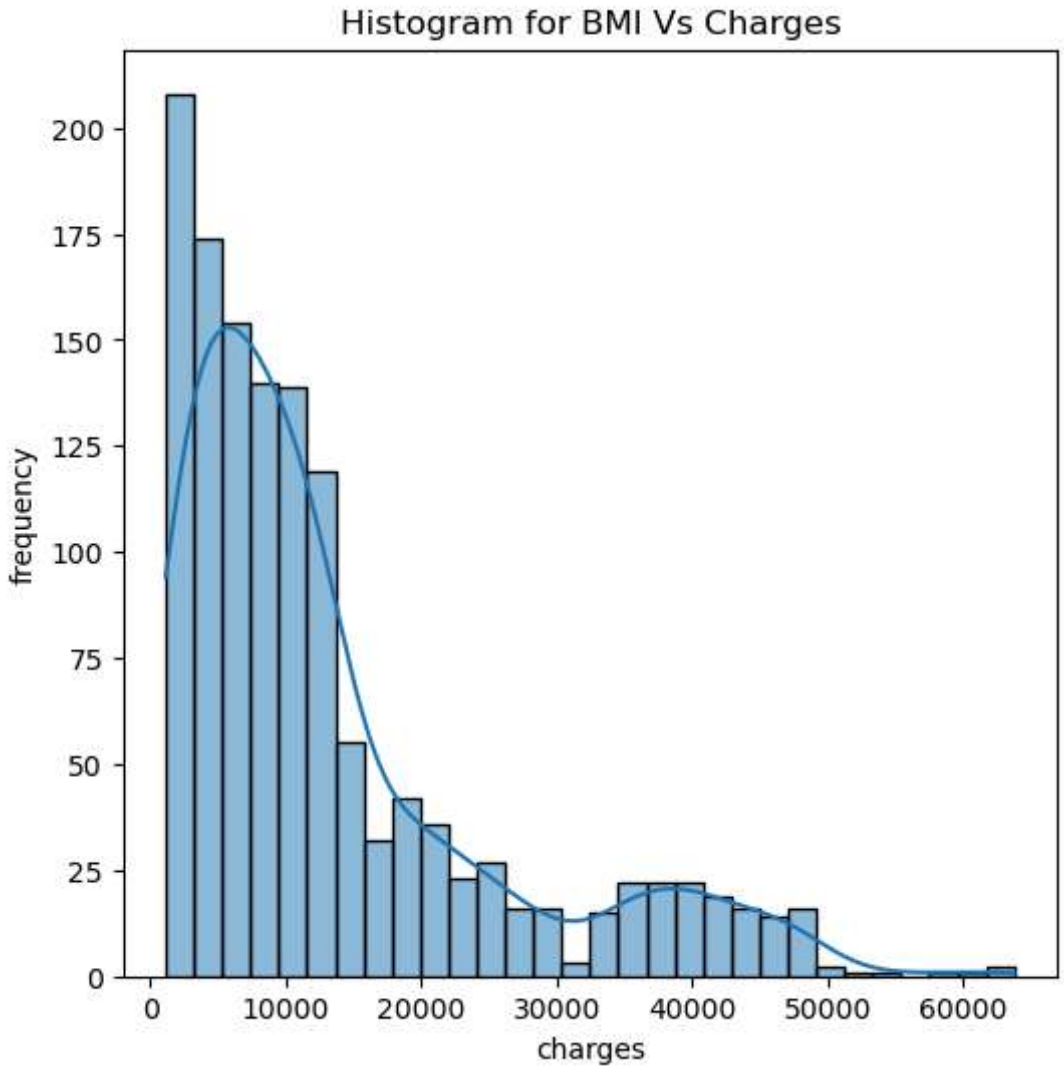



In [172…
```
plt.figure(figsize=(6,6))
sns.barplot(data =df, x= 'smoker', y='charges', hue='smoker')
plt.title('Smoker Vs Charges')
plt.xlabel('Smoker')
plt.ylabel('Charges')
plt.show()
```



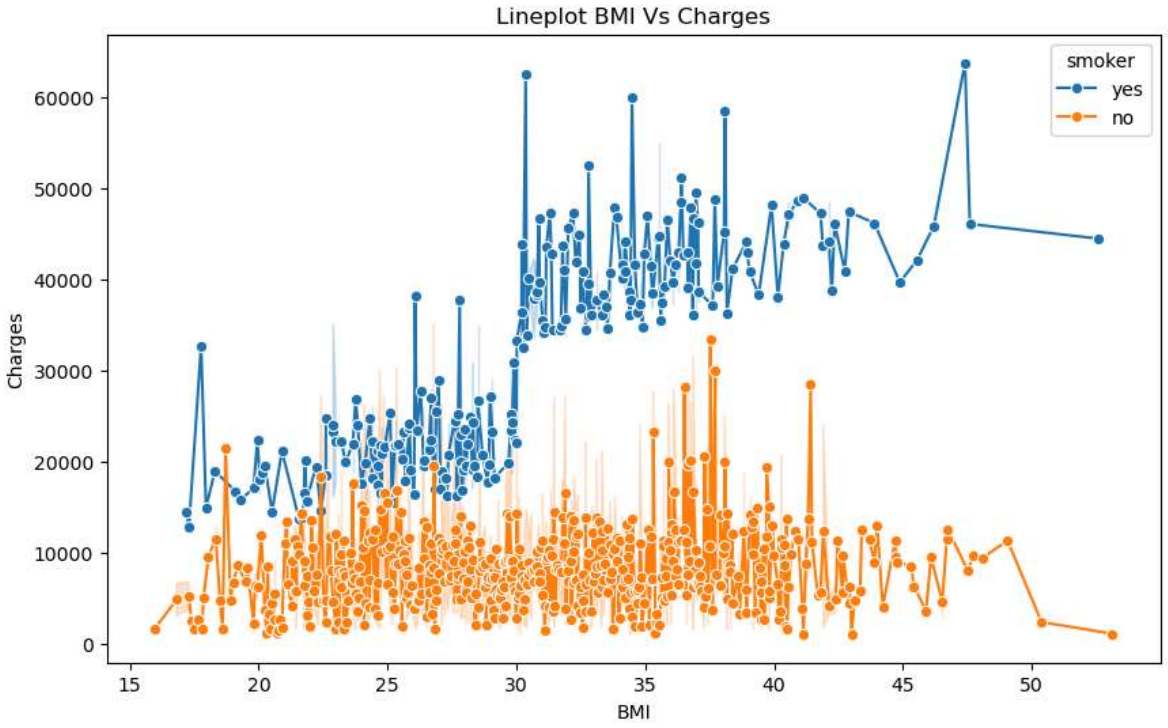

In [206…

```
plt.figure(figsize=(6,6))
sns.histplot(df['charges'], kde =True)
plt.xlabel('charges')
plt.ylabel('frequency')
plt.title('Histogram for BMI Vs Charges')
plt.show()
```




In [100…

```
plt.figure(figsize=(10,6))
sns.lineplot(data =df, x='bmi', y ='charges', hue ='smoker', marker ='o')
plt.title('Lineplot BMI Vs Charges')
plt.xlabel('BMI')
plt.ylabel('Charges')
plt.show()
```

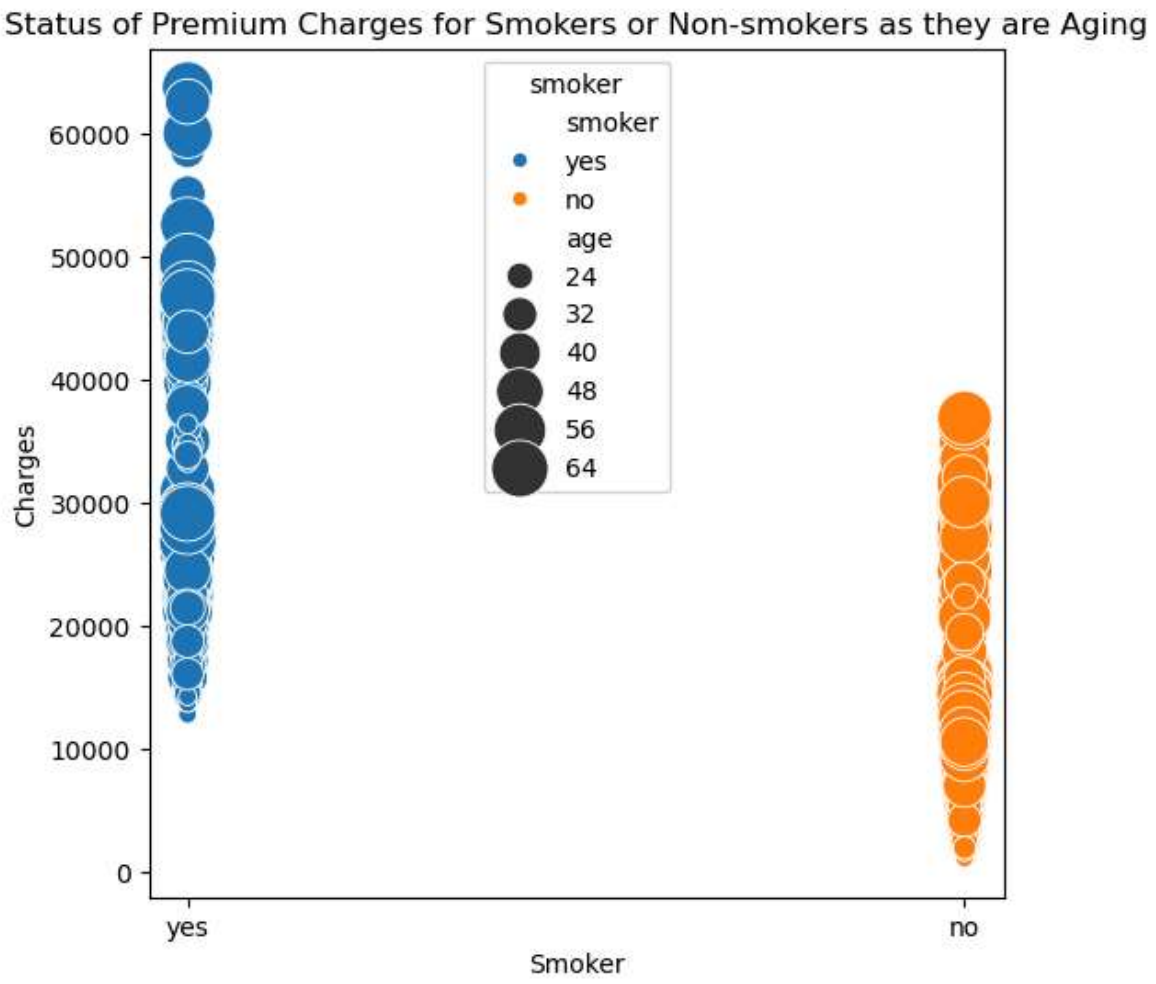



# OBSERVATION

Different Visualization for Different Purpose :- 1. Pair Plot in respect of Smokers - Showing the Co-relation among all age, charges, bim & children in respect of Smokers. 2. Pair Plot in respect of Sex - Showing the Co-relation among all age, charges, bim & children in respect of Sex. 3. Bubble Plot of Age vs. BMI with Charges as Size - Showing the relation between Age Vs. BMI where the size of the Bubble is getting large as the premium charge is going high. 5. Boxplot for Age Vs Region in respect of males & females 6. Smoker Vs Charges - Showing Premium Charges for Smokers are more higher than Non-smoker in respect of Sex 7. Histogram for BMI Vs Charges 8. Lineplot BMI Vs Charges

# 6. Check if the number of premium charges for smokers or non-smokers is increasing as they are aging

In [58]:
```
plt.figure(figsize=(6,6))
sns.scatterplot(data =df , x='smoker', y ='charges',size ='age', sizes=(50,500),
plt.xlabel('Smoker')
plt.ylabel('Charges')
plt.title('Status of Premium Charges for Smokers or Non-smokers as they are Agin
plt.legend(title ='smoker', loc ='upper center')
plt.show()
```




# OBSERVATION

Status of Premium Charges for Smokers or Non-smokers as they are Aging: In overall conclusion the last Bubble plot is showing that smokers that have policy are paying more Premium charges compared to the Non-smokers. Size of Bubble is getting small & large as the age of individual is older than premium charge is getting high & vice -versa. whereas color blue is for the Smokers that have policy while the orange color is for Non-smokers having policy.

In [ ]: