Project_3

May 2, 2025

[26]:
```
#Hlanhla Hlungwane
#02 May 2025
#Data Analysis for insurance
#Python

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

#Importing and reading the Insurance CSV file
df = pd.read_csv("insurance.csv")
```

[27]:
```
df.head()
```

[27]:

	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

[58]:
```
#Shape of the data set
#There are 1338 Rows and 7 columns
df.shape
```

[58]:
```
(1338, 7)
```

[59]:
```
#Checking the data types of each column in the data frame
df.dtypes
```

[59]:
```
age           int64
sex          object
bmi         float64
children      int64
smoker       object
region       object
```

```
charges     float64
dtype: object
```

[60]:
```
#Missing data
#Replacing all the empty spaces with NaN
df.replace(" ", np.nan, inplace = True)
missing_data = df.isnull()
```

[61]:
```
#Counting the sum of missing data in each column
#There is no missing data
missing_counts = missing_data.sum()
print(missing_counts)
```

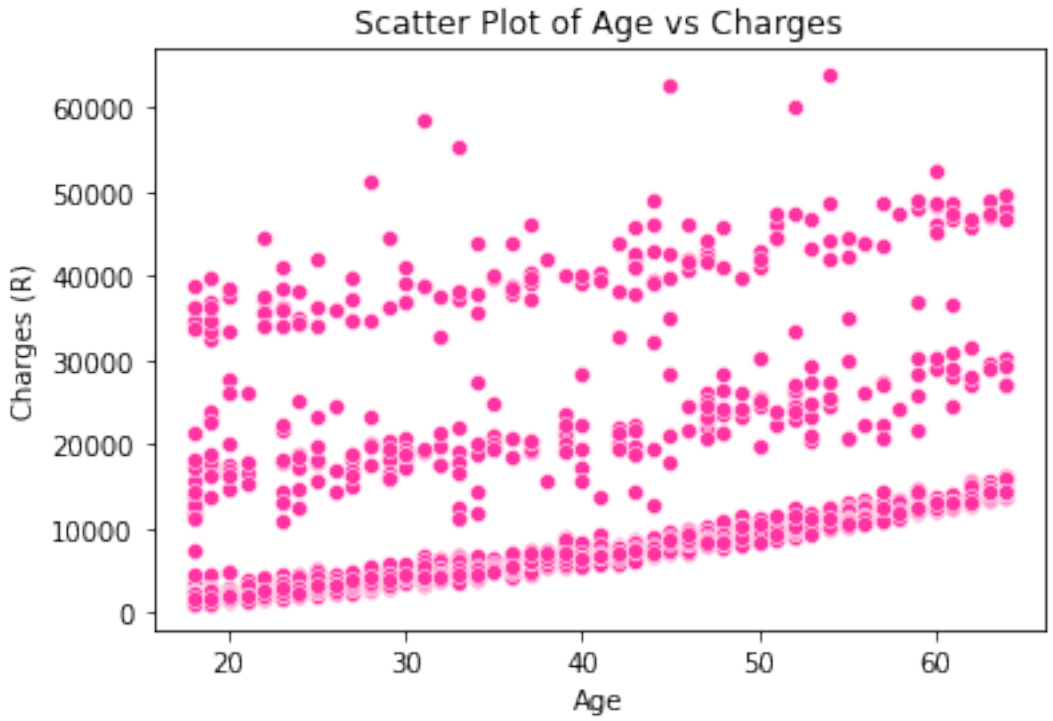
```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

[23]:
```
for column in missing_data.columns.values.tolist():          #To count the␣
 ↪number of missing values in each column
    print(column)
    print(missing_data[column].value_counts())               #False indicates␣
 ↪the number of values that are not null
    print("")                                                #True indicates the␣
 ↪number of missing values
```

```
age
False    1338
Name: age, dtype: int64

sex
False    1338
Name: sex, dtype: int64

bmi
False    1338
Name: bmi, dtype: int64

children
False    1338
Name: children, dtype: int64

smoker
```

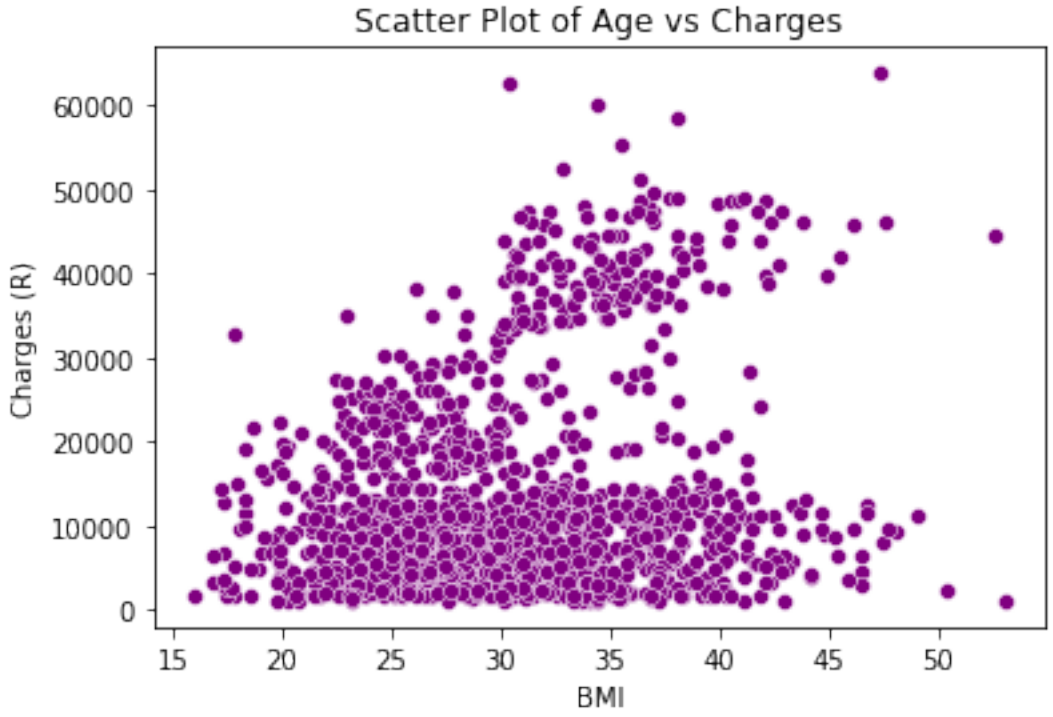
```
False    1338
Name: smoker, dtype: int64

region
False    1338
Name: region, dtype: int64

charges
False    1338
Name: charges, dtype: int64
```

[65]:
```
# Scatter plot of 'age' vs 'charges'
sns.scatterplot(x='age', y='charges', data=df, color='#FF33A1')
plt.xlabel('Age')
plt.ylabel('Charges (R)')
plt.title('Scatter Plot of Age vs Charges')
plt.show()
#Based on the scatter plot, the charges increase with Age
```

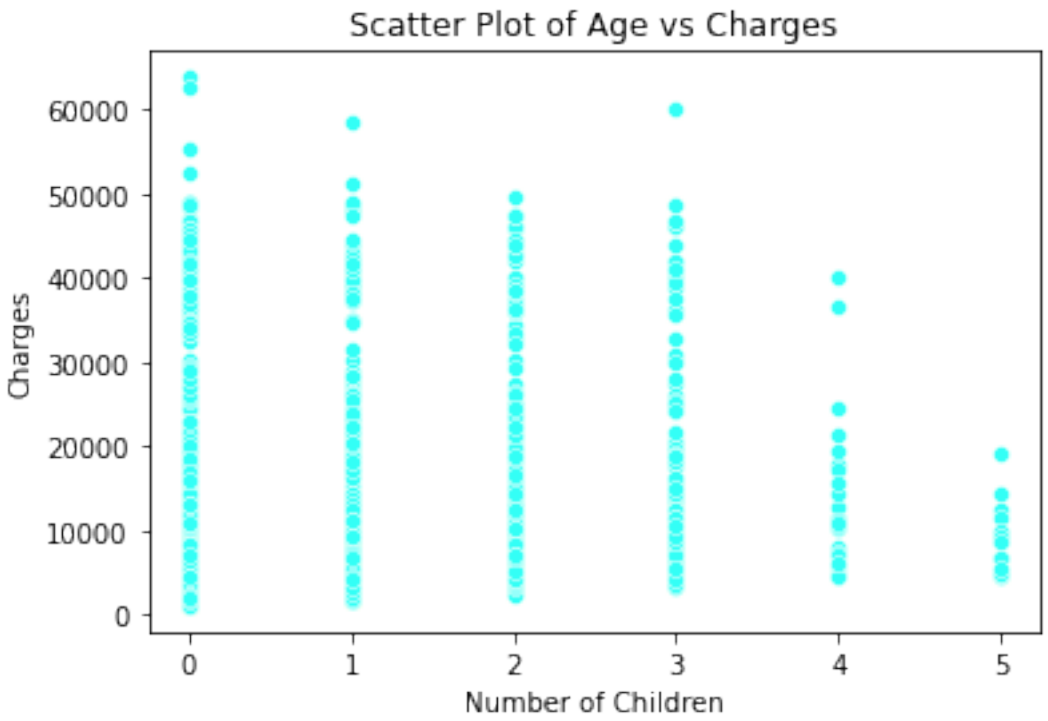


[67]:
```
# Scatter plot of 'age' vs 'charges'
sns.scatterplot(x='bmi', y='charges', data=df, color='purple')
plt.xlabel('BMI')
plt.ylabel('Charges (R)')
```

```
plt.title('Scatter Plot of Age vs Charges')
plt.show()
#Based on the scatter plot below, the charges increase with increasing BMI
```



[40]:
```
# Scatter plot of 'age' vs 'charges'
sns.scatterplot(x='children', y='charges', data=df, color='#33FFF6')
plt.xlabel('Number of Children')
plt.ylabel('Charges(R)')
plt.title('Scatter Plot of Age vs Charges')
plt.show()
#Based on the scatter plot below, the charges decrease with increasing number␣
 ↪of kids
```

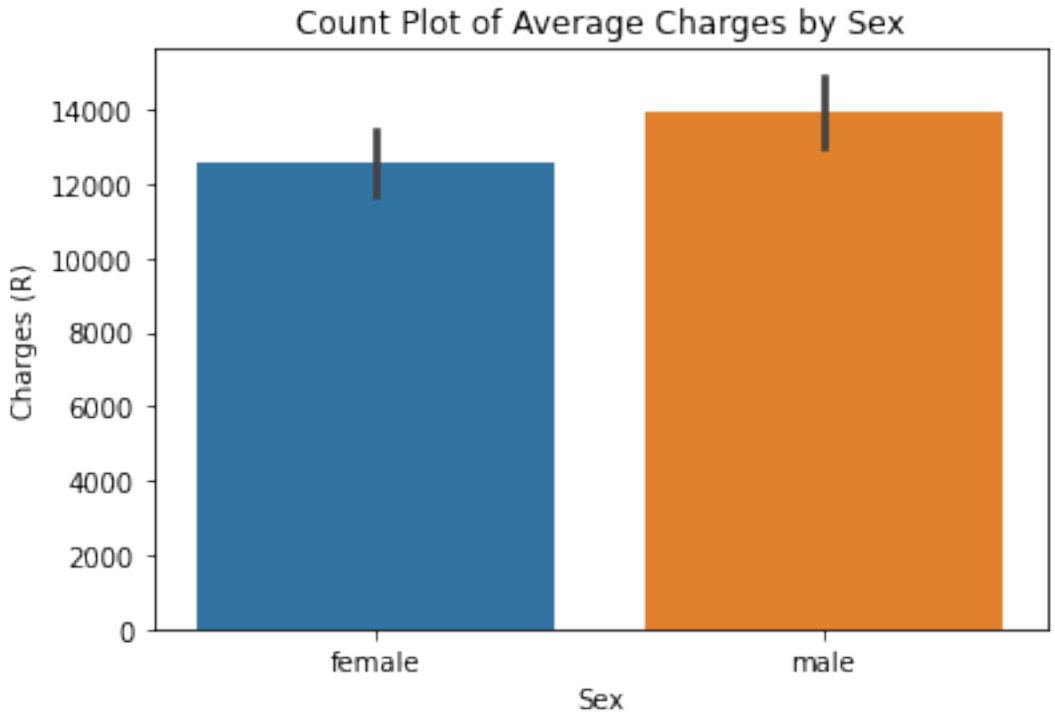


[68]:
```
# Create dummy variables for 'sex', 'smoker', and 'region'to change them from␣
 ↪categorical to numerical
df_dummies = pd.get_dummies(df, columns=['sex', 'smoker', 'region'],␣
 ↪drop_first=True)
print(df_dummies.head())
```

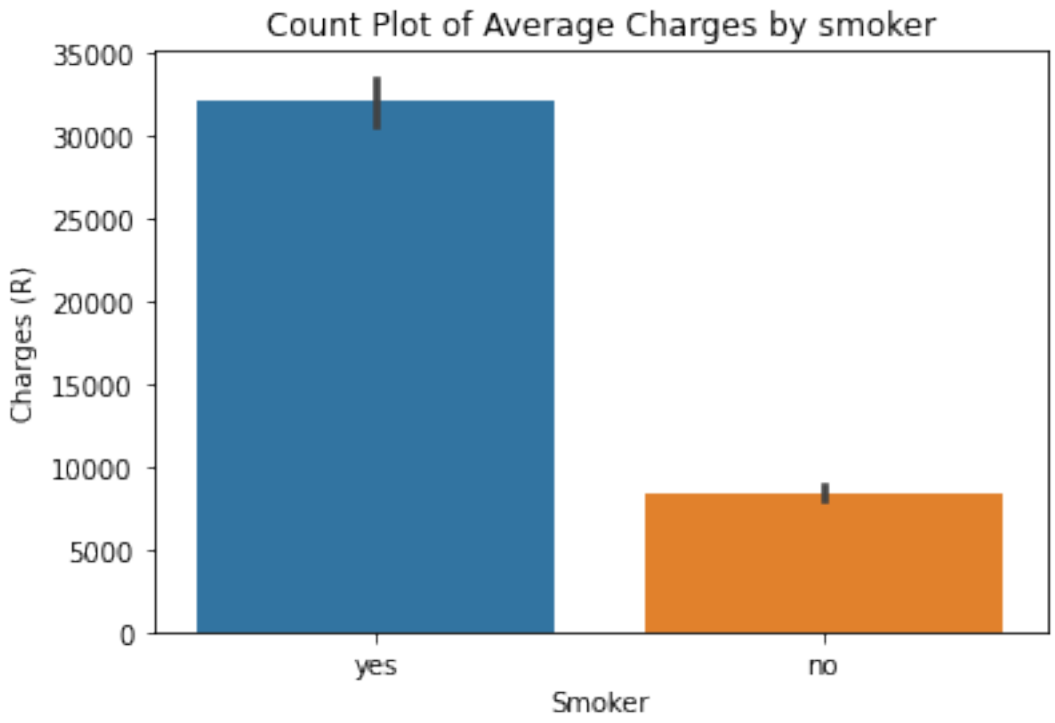
```
   age     bmi  children      charges  sex_male  smoker_yes  region_northwest  \
0   19  27.900         0  16884.92400         0           1                 0
1   18  33.770         1   1725.55230         1           0                 0
2   28  33.000         3   4449.46200         1           0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

   region_southeast  region_southwest
0                 0                 1
1                 1                 0
2                 1                 0
3                 0                 0
4                 0                 0
```

[71]:
```
#Chart of Age vs Charges
sns.barplot(x='sex', y='charges', data=df)
plt.title('Count Plot of Average Charges by Sex')
```

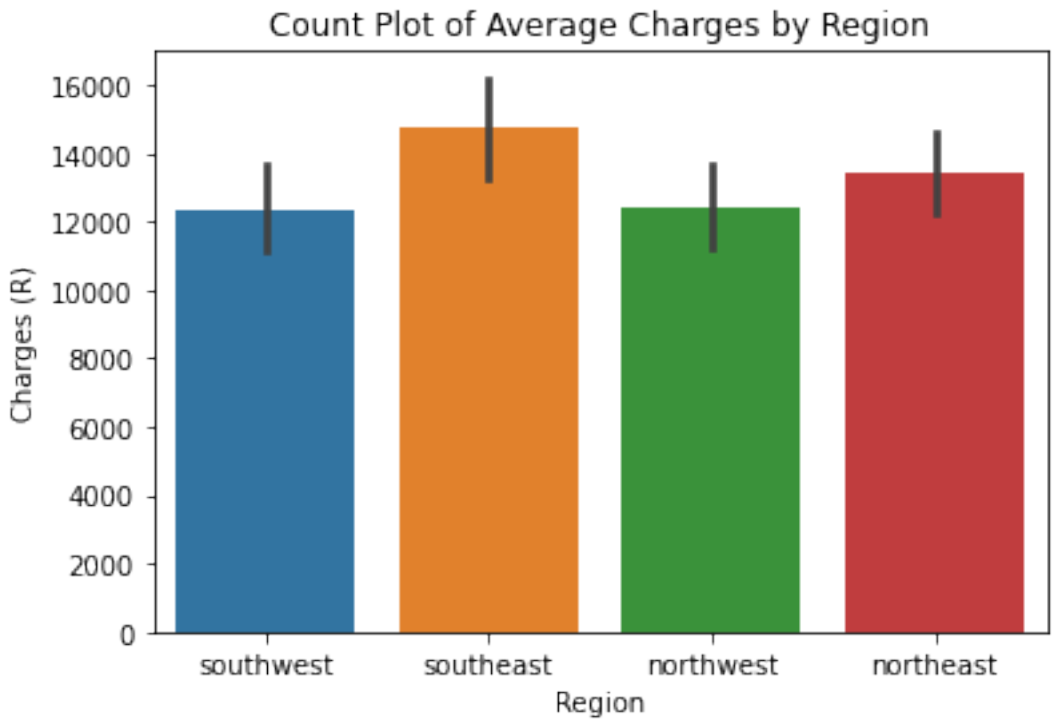
```
plt.xlabel("Sex")
plt.ylabel("Charges (R)")
plt.show()
#Males pay more chages than females
```



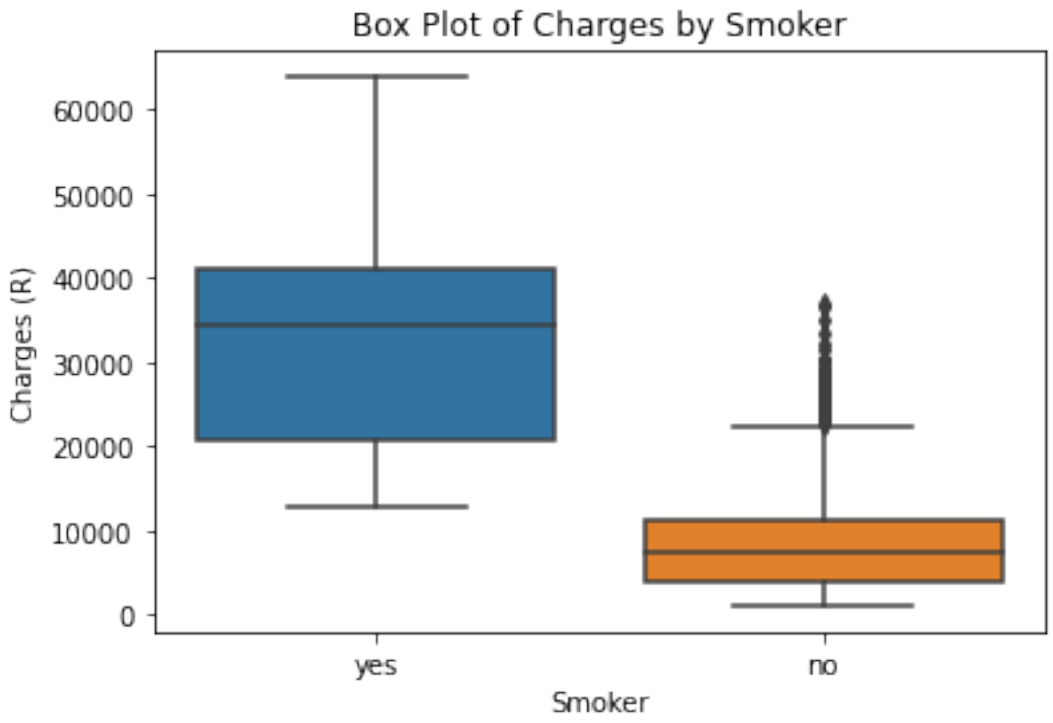
[72]:
```
#Chart of smoker vs charges
sns.barplot(x='smoker', y='charges', data=df)
plt.title('Count Plot of Average Charges by smoker')
plt.xlabel('Smoker')
plt.ylabel('Charges (R)')
plt.show()
#Smokers pay more than non-smokers
```



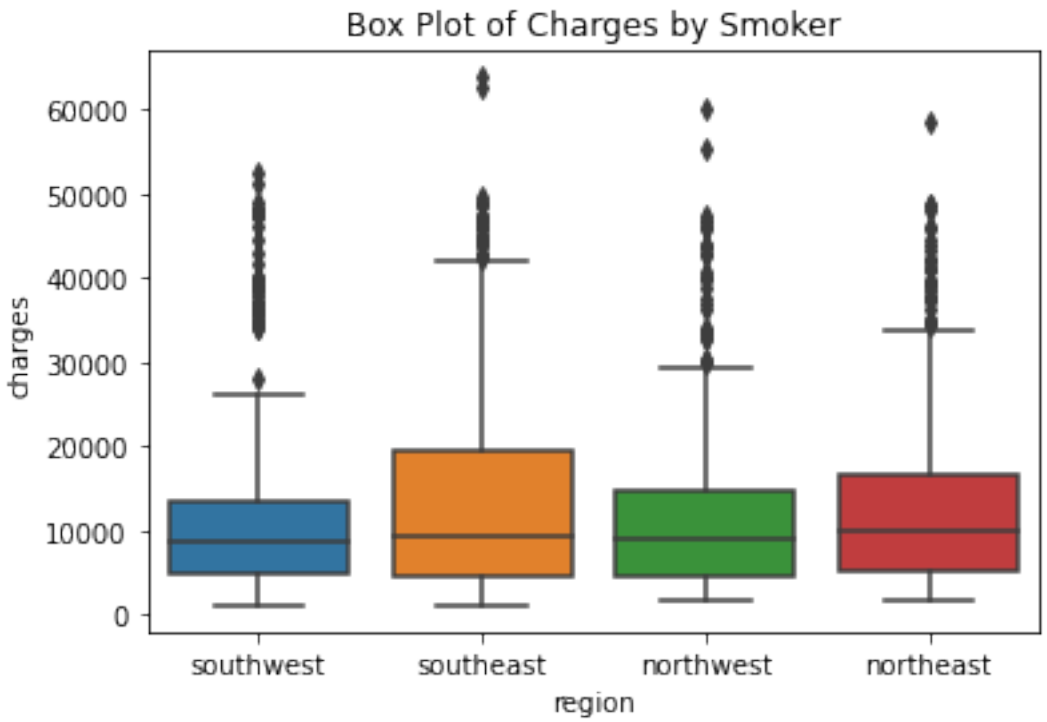
[73]:
```
#Chart of region vs charges
sns.barplot(x='region', y='charges', data=df)
plt.title('Count Plot of Average Charges by Region')
plt.xlabel("Region")
plt.ylabel("Charges (R)")
plt.show()
#People from the south east pay more charges
```



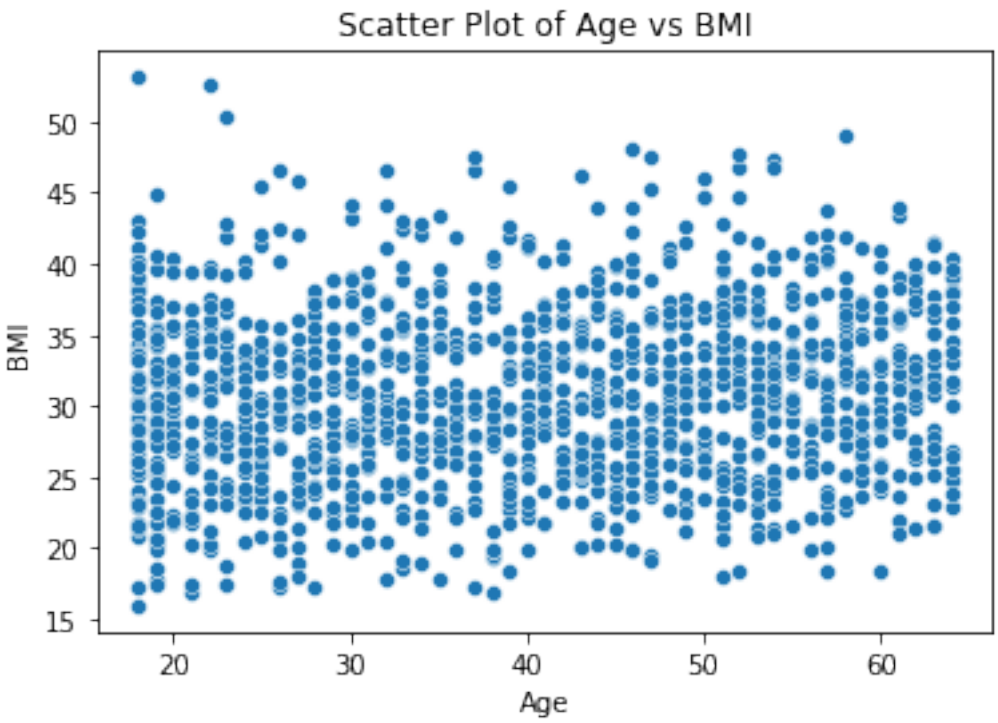
[74]:
```
#Box plot of charges by smoker
sns.boxplot(x='smoker', y='charges', data=df)
plt.title('Box Plot of Charges by Smoker')
plt.xlabel('Smoker')
plt.ylabel('Charges (R)')
plt.show()
#Charges are higher for smokers
```



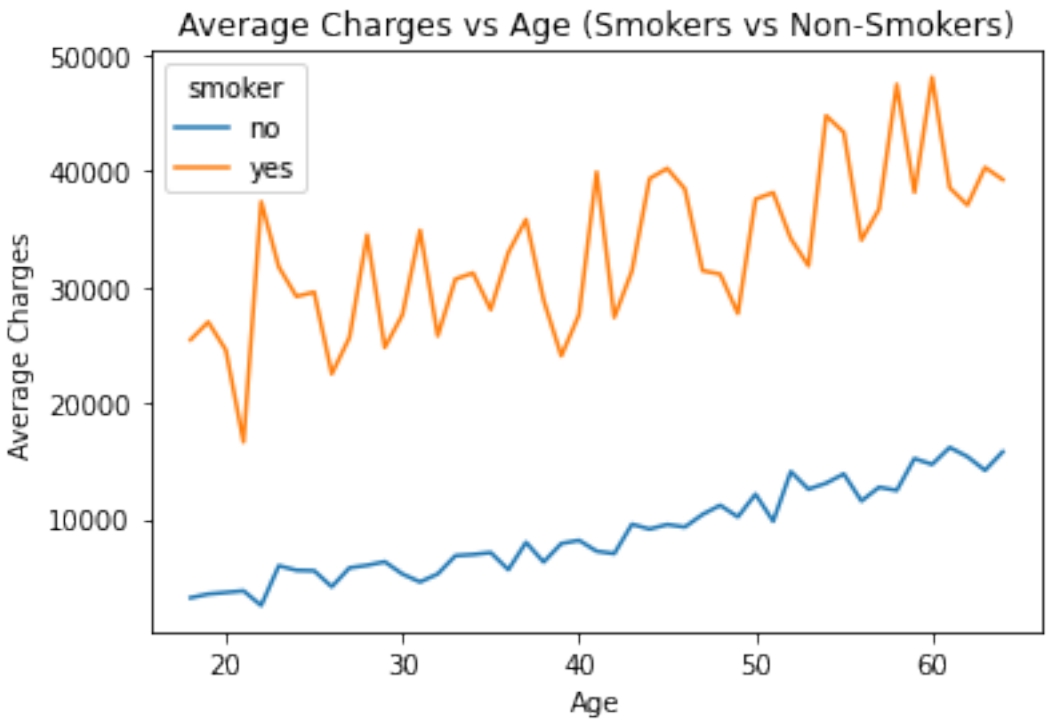
[48]:
```
#Box plot of charges by region
sns.boxplot(x='region', y='charges', data=df)
plt.title('Box Plot of Charges by Smoker')
plt.show()
```



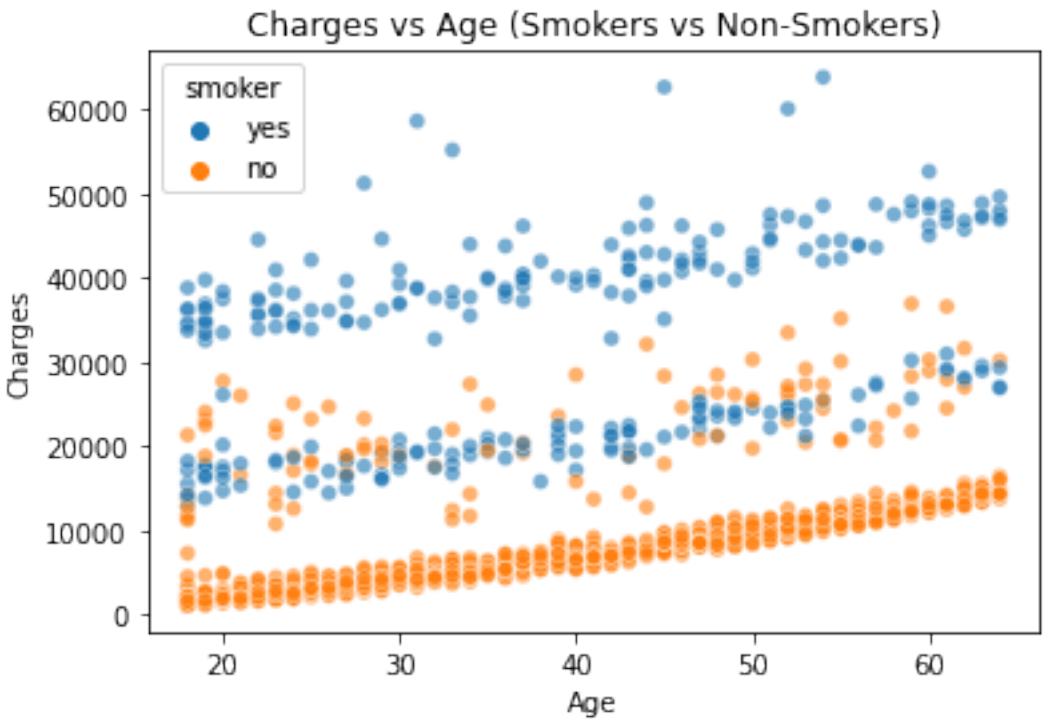
[75]:
```
#Box Plot of Age vs BMI
sns.scatterplot(x='age', y='bmi', data=df)
plt.title('Scatter Plot of Age vs BMI')
plt.xlabel('Age')
plt.ylabel('BMI')
plt.show()
#There is a unifrom relationship between age and BMI
```



[76]:
```
# Group data by 'age' and 'smoker', then calculate the mean charges
grouped_data = df.groupby(['age', 'smoker'])['charges'].mean().reset_index()

# Line plot to visualize the trend
sns.lineplot(x='age', y='charges', hue='smoker', data=grouped_data)
plt.title('Average Charges vs Age (Smokers vs Non-Smokers)')
plt.xlabel('Age')
plt.ylabel('Average Charges')
plt.show()
#Age is directly proportional to Charges
#Smokers always pay more than non-smokers
#Younger non-smokers pay the least charges
#Older smokers pay the most charges
```



[77]:
```
# Scatter plot to visualize individual data points
sns.scatterplot(x='age', y='charges', hue='smoker', data=df, alpha=0.6)
plt.title('Charges vs Age (Smokers vs Non-Smokers)')
plt.xlabel('Age')
plt.ylabel('Charges')
plt.show()
#Age is directly proportional to Charges
#Smokers always pay more than non-smokers
#Younger non-smokers pay the least charges
#Older smokers pay the most charges
```


Charges vs Age (Smokers vs Non-Smokers)
smoker
yes
no
Charges
Age
0
10000
20000
30000
40000
50000
60000
20
30
40
50
60