Project_Insurance_Data (1)

February 13, 2024

[ ]:
```
#TLADI LT INSURANCE EDA PROJECT
```

[1]:
```
#importing libraries and loading data
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

# Load the dataset
insurance_data = pd.read_csv('insurance.csv')
```

```
C:\Users\CODEINE\AppData\Local\Temp\ipykernel_7480\2700369507.py:2:
DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next major release of
pandas (pandas 3.0),
(to allow more performant data types, such as the Arrow string type, and better
interoperability with other libraries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466

  import pandas as pd
```

[2]:
```
#checking data head
insurance_data.head()
```

[2]:

	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

[2]:
```
#checking data info
insurance_data.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
```

[3]:
```
# Check the shape of the data
print("Shape of the dataset:", insurance_data.shape)

# Check data types of each column
print(insurance_data.dtypes)
```

```
Shape of the dataset: (1338, 7)
age           int64
sex          object
bmi         float64
children      int64
smoker       object
region       object
charges     float64
dtype: object
```

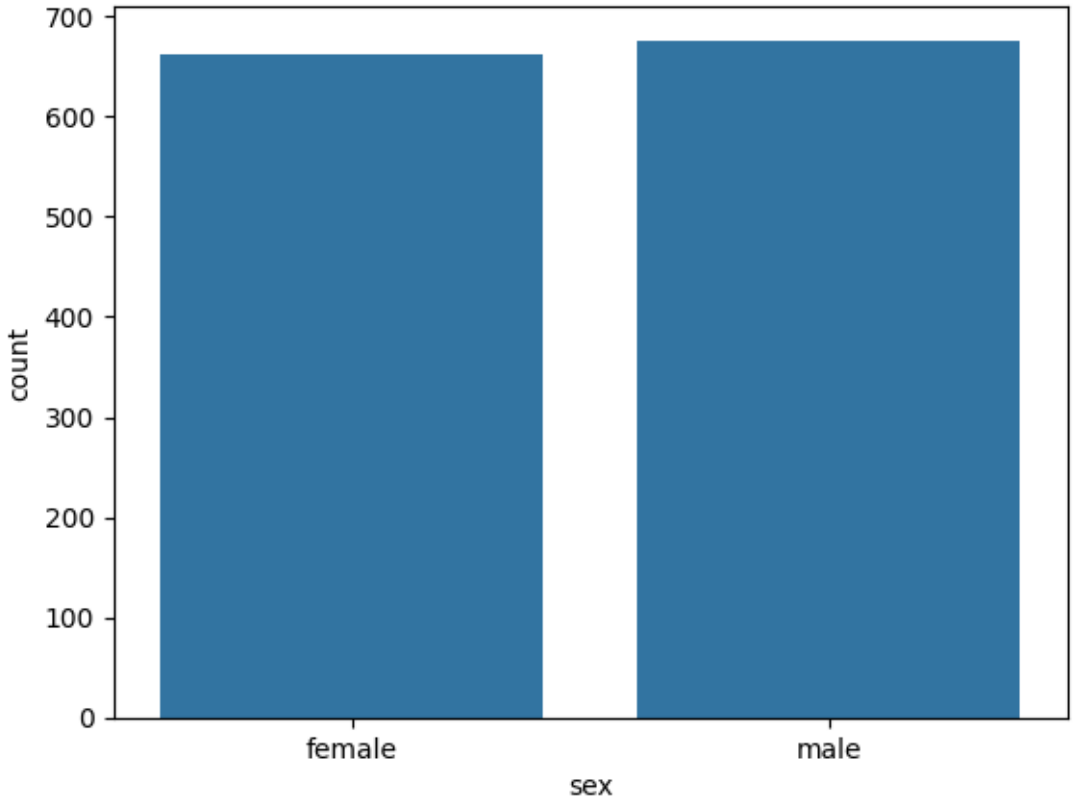
[4]:
```
# Check for missing values
print("Missing values:\n", insurance_data.isnull().sum())
```

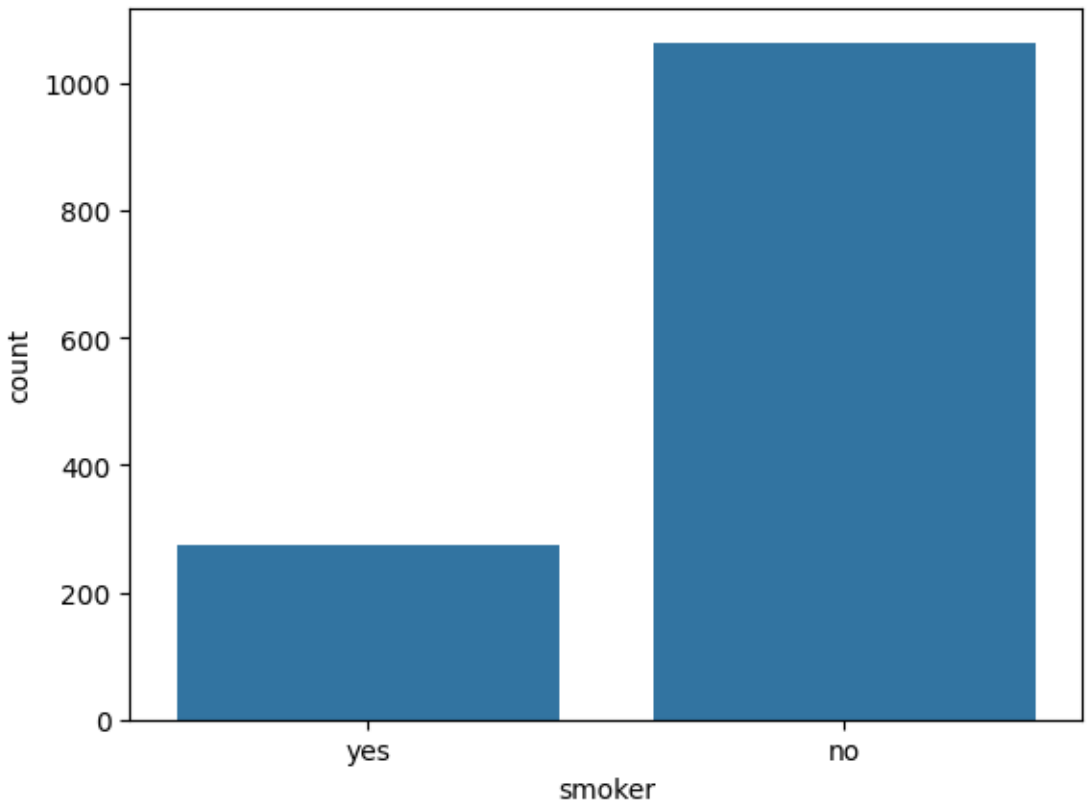
```
Missing values:
 age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

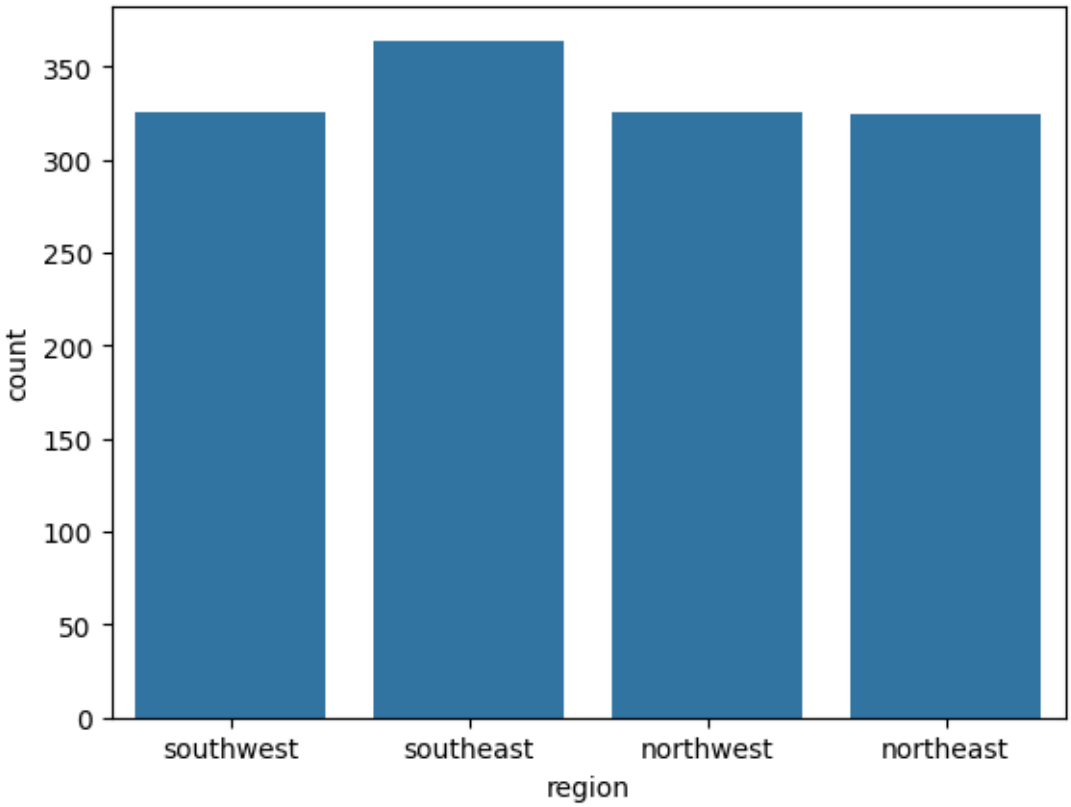
[ ]:
```
#Observations - There are no missing values , no further action is necessary
```

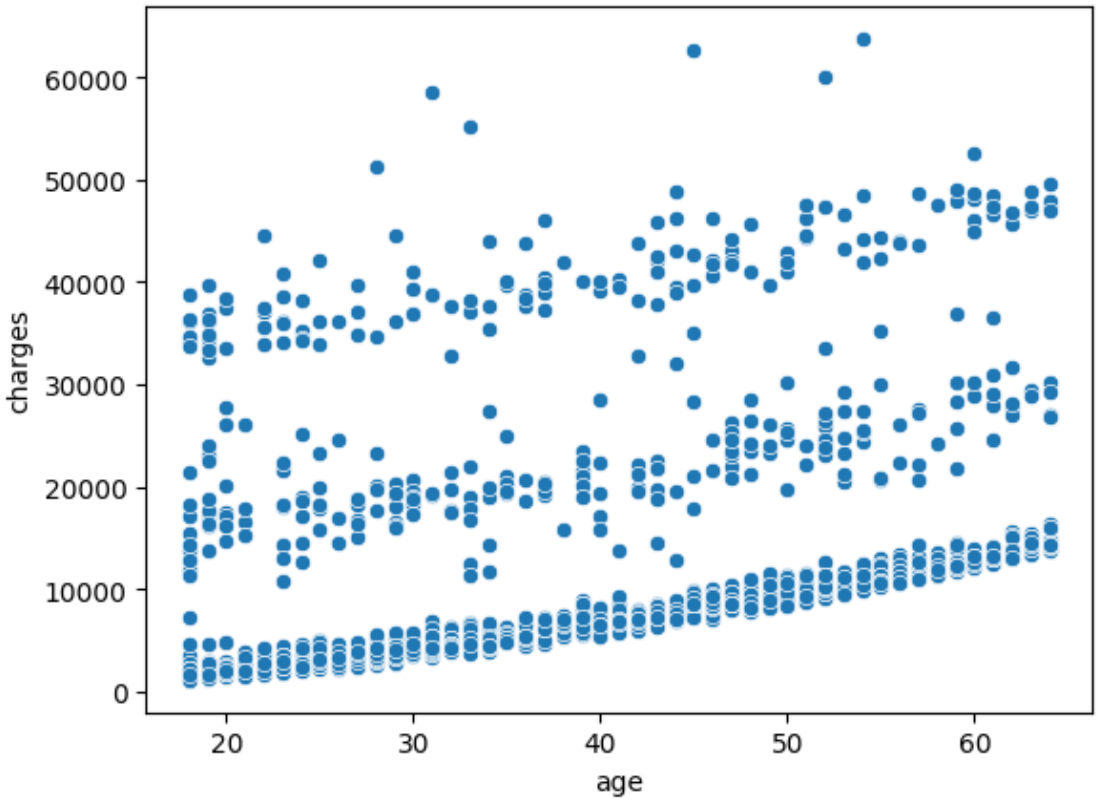
[7]:
```
# Count plot of categorical columns (e.g., sex, smoker, region)
sns.countplot(x='sex', data=insurance_data)
plt.show()
```

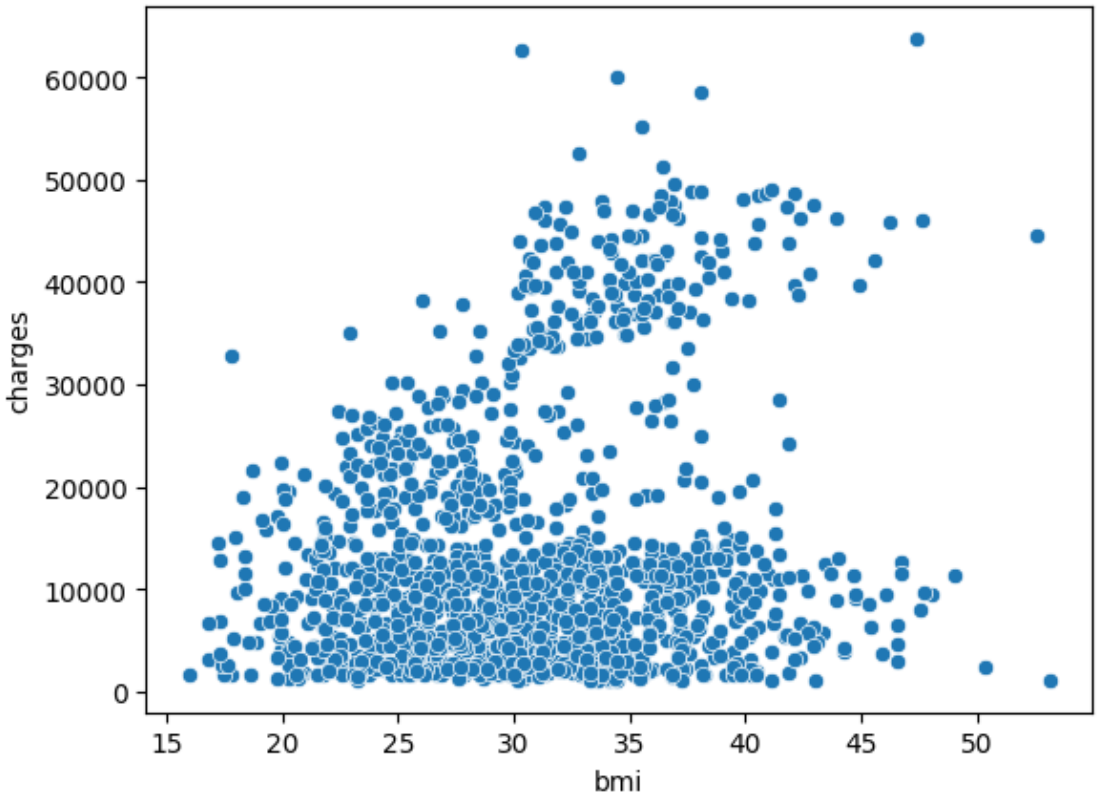
```
sns.countplot(x='smoker', data=insurance_data)
plt.show()

sns.countplot(x='region', data=insurance_data)
plt.show()

# Scatter plot of numerical columns (e.g., age, BMI, children)
sns.scatterplot(x='age', y='charges', data=insurance_data)
plt.show()

sns.scatterplot(x='bmi', y='charges', data=insurance_data)
plt.show()

sns.scatterplot(x='children', y='charges', data=insurance_data)
plt.show()
```

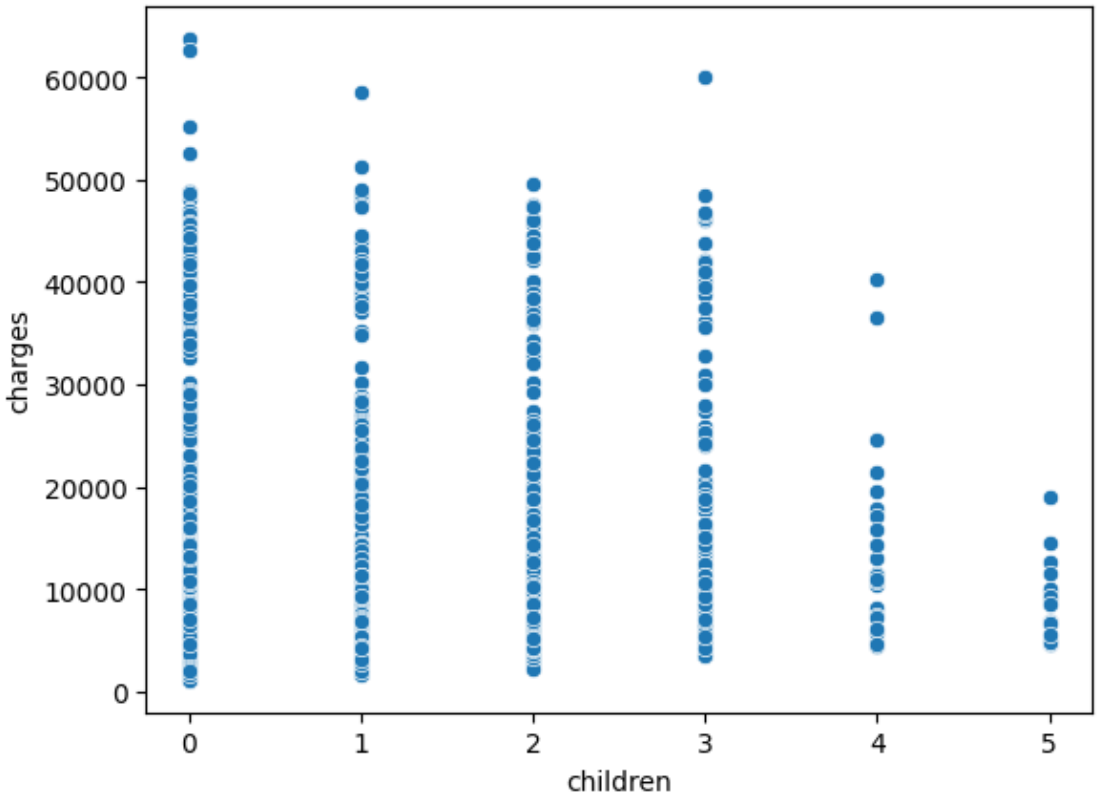



1000
800
600
400
200
0
count
yes
no
smoker


350
300
250
200
150
100
50
0
count
southwest
southeast
northwest
northeast
region


charges
60000
50000
40000
30000
20000
10000
0
20
30
40
50
60
age

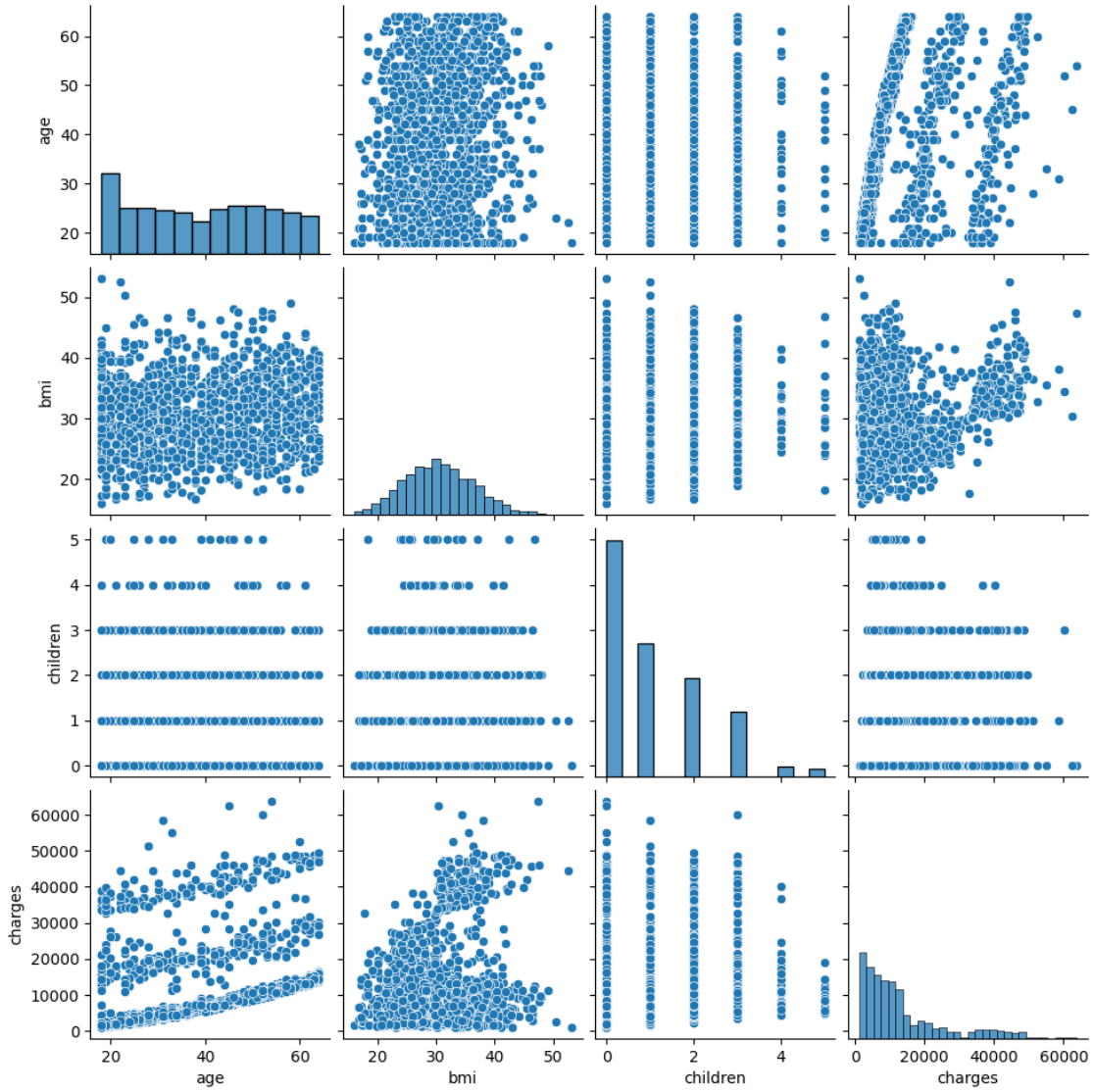

charges
60000
50000
40000
30000
20000
10000
0
15
20
25
30
35
40
45
50
bmi



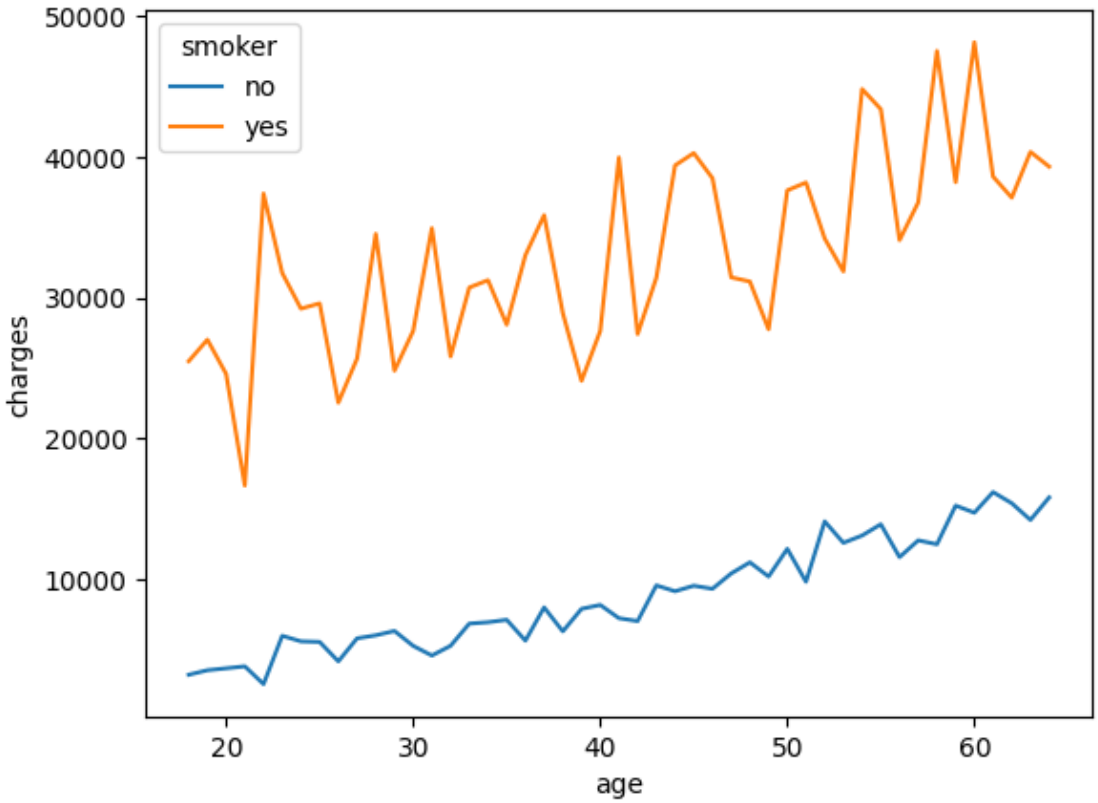
[ ]:
```
#Key Observations: BMI and charges
#The relationship between BMI and charges is not perfectly linear.
#(There are data points scattered throughout the plot, indicating that other␣
 ↪factors besides BMI also influence charges.)
#There is a wider range of charges for people with higher BMI
```

[ ]:
```
#Key Observation :
#Smokers generally have higher charges than non-smokers
#The difference in charges between smokers and non-smokers increases with age.
#The gap between the two groups widens as age increases
#There is a lot of variability in charges within each group -
#(There are points from both groups scattered throughout the plot, -
#indicating that factors other than smoking status also influence charges.
```

[8]:
```
# Pairplot for visualizing feature vs feature
sns.pairplot(insurance_data)
plt.show()
```



[12]:
```
# Group by age and smoker, then calculate mean charges
age_smoker_charges = insurance_data.groupby(['age', 'smoker'])['charges'].
  ↪mean().reset_index()

# Plotting
sns.lineplot(x='age', y='charges', hue='smoker', data=age_smoker_charges)
plt.show()
```



[ ]:
```
#key Observation :chart clearly shows the number of premium charges for smokers
 ↪and non-smokers is increasing as they are aging.
#Also smpokers have heigher premium charges than non smokers
```