DSPLexp2

February 15, 2024

Prathamesh Vaidya Roll no:59 Batch - B TE-IT

DSPL-Experiment No: 2

Aim: Data Visualization / Exploratory Data Analysis for the selected data set using Matplotlib and Seaborn

LO's Achieved: LO2.

PO's Achieved: PO1, PO2, PO3, PO4, PO5.

[ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Uploading csv file

[ ]:
```
Data1 = pd.read_csv('insurance.csv')
```

Displaying the head of data

[ ]:
```
Data1.head(5)
```

[ ]:

	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

Displaying statistics about the data

[ ]:
```
Data1.describe()
```

[ ]:

	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

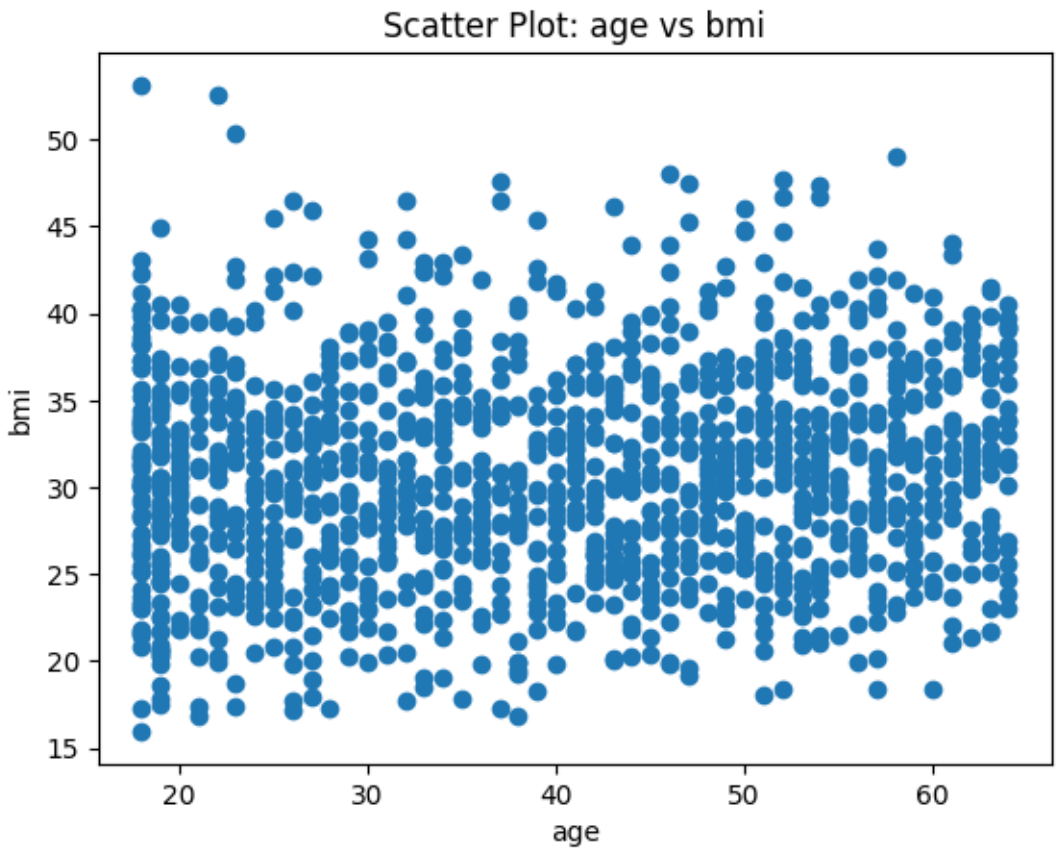
```
25%      27.000000    26.296250     0.000000   4740.287150
50%      39.000000    30.400000     1.000000   9382.033000
75%      51.000000    34.693750     2.000000  16639.912515
max      64.000000    53.130000     5.000000  63770.428010
```

[ ]:
```
Data1.info()
```

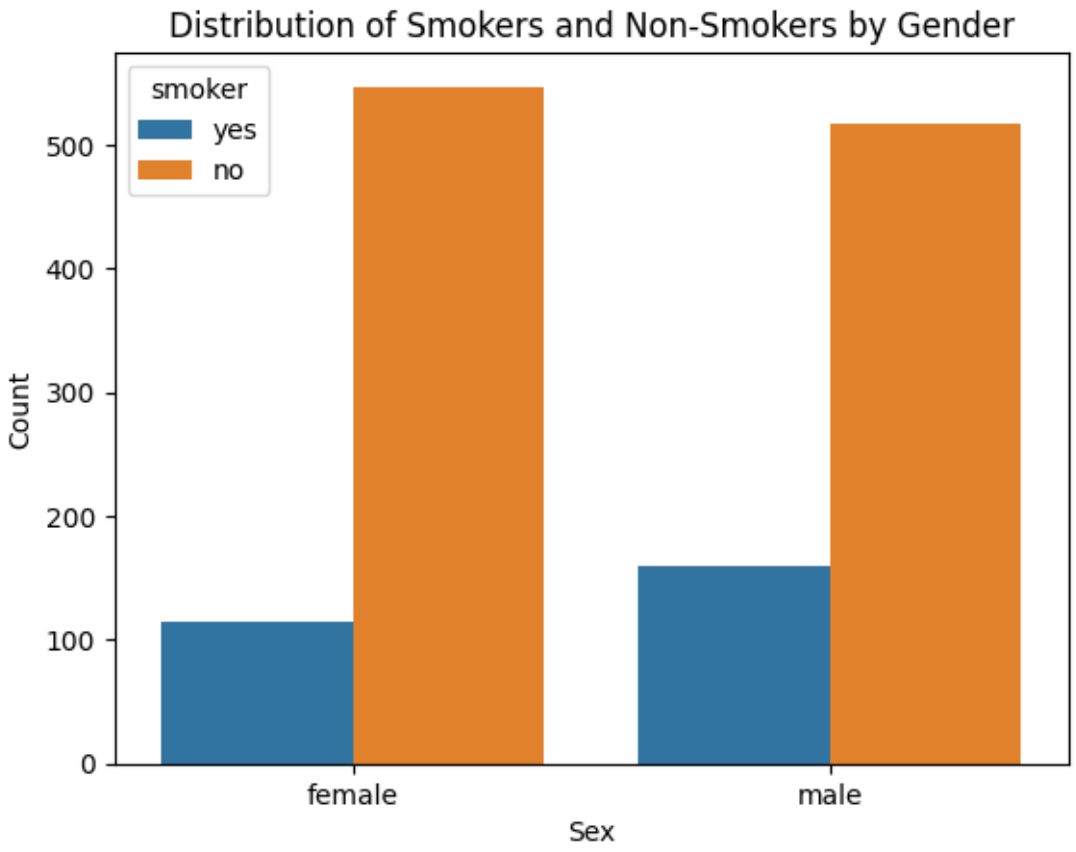
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       1338 non-null   int64
 1   sex       1338 non-null   object
 2   bmi       1338 non-null   float64
 3   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
```

Display:

[ ]:
```
x = Data1['age']
y = Data1['bmi']
plt.scatter(x, y)
plt.xlabel('age')
plt.ylabel('bmi')
plt.title('Scatter Plot: age vs bmi')
plt.show()
```



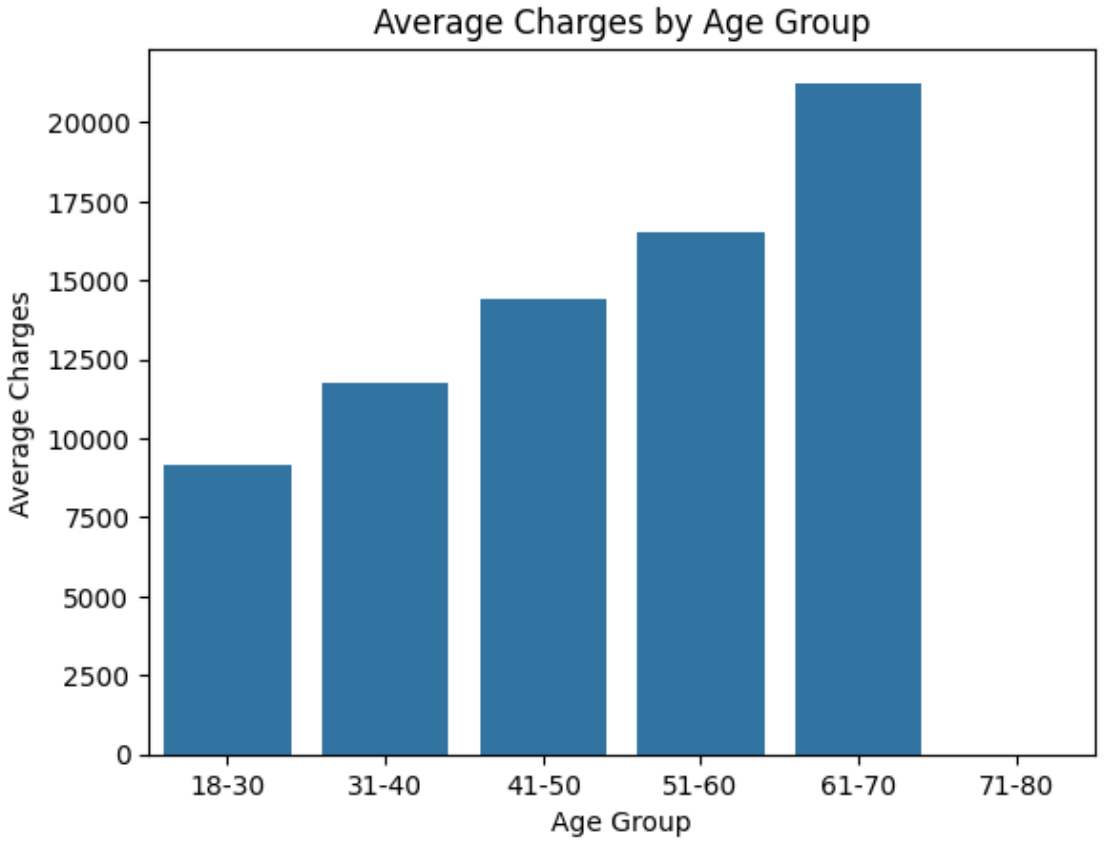
[ ]:
```
sex_smoker_df = Data1[['sex', 'smoker']]

sns.countplot(x='sex', hue='smoker', data=sex_smoker_df)

plt.xlabel('Sex')
plt.ylabel('Count')
plt.title('Distribution of Smokers and Non-Smokers by Gender')
plt.show()
```



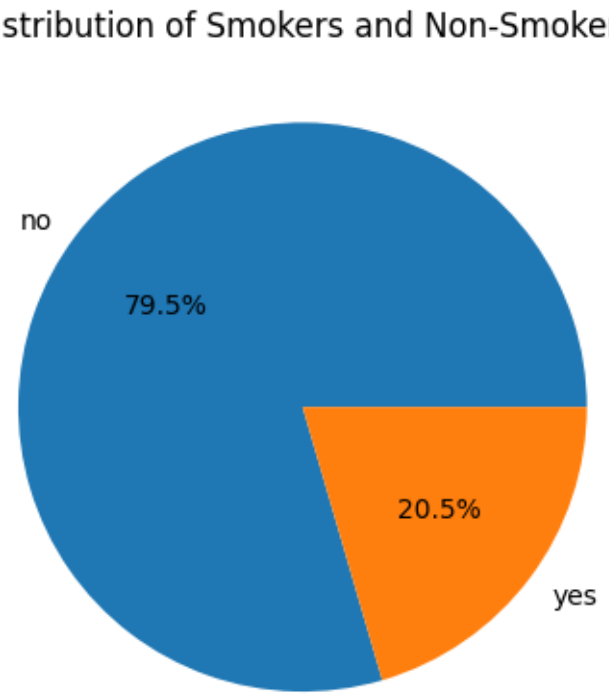
[ ]:
```
age_charges_df = Data1[['age', 'charges']]

# Create age groups (bins)
age_bins = [18, 30, 40, 50, 60, 70, 80]
age_labels = ['18-30', '31-40', '41-50', '51-60', '61-70', '71-80']

# Use .loc to avoid SettingWithCopyWarning
age_charges_df.loc[:, 'age_group'] = pd.cut(age_charges_df['age'],␣
 ↪bins=age_bins, labels=age_labels, right=False)

# Calculate the average charges within each age group
avg_charges_by_age = age_charges_df.groupby('age_group')['charges'].mean().
 ↪reset_index()

# Create a bar plot
sns.barplot(x='age_group', y='charges', data=avg_charges_by_age)

# Customize the plot
plt.xlabel('Age Group')
plt.ylabel('Average Charges')
```

```
plt.title('Average Charges by Age Group')
plt.show()
```

```
<ipython-input-19-79ee4f1a2041>:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  age_charges_df.loc[:, 'age_group'] = pd.cut(age_charges_df['age'],
bins=age_bins, labels=age_labels, right=False)
```



[ ]:
```
smoker_counts = Data1['smoker'].value_counts()

plt.pie(smoker_counts, labels=smoker_counts.index, autopct='%1.1f%%',  )
plt.title('Distribution of Smokers and Non-Smokers')
plt.show()
```



Conclusion: In this experiment, I have studied data visualization techniques and exploratory data analysis methods on an insurance.csv file using Matplotlib and Seaborn libraries of python. It helps in identifying trends and patterns within the attributes of the dataset.

Hence, LO2 is achieved and PO1, PO2, PO3, PO4, PO5 are mapped successfully.