

```
In [ ]: # Prepare By Husssam Almokayed
```

```
In [1]: # INSTALL THE LIBEREARY  
!pip install numpy  
!pip install pandas  
!pip install matplotlib  
!pip install seaborn
```

Requirement already satisfied: numpy in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (2.3.4)

Requirement already satisfied: pandas in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (2.3.3)

Requirement already satisfied: numpy>=1.26.0 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from pandas) (2.3.4)

Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from pandas) (2.9.0.post0)

Requirement already satisfied: pytz>=2020.1 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from pandas) (2025.2)

Requirement already satisfied: tzdata>=2022.7 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from pandas) (2025.2)

Requirement already satisfied: six>=1.5 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)

Requirement already satisfied: matplotlib in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (3.10.7)

Requirement already satisfied: contourpy>=1.0.1 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from matplotlib) (1.3.3)

Requirement already satisfied: cyclor>=0.10 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from matplotlib) (0.12.1)

Requirement already satisfied: fonttools>=4.22.0 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from matplotlib) (4.60.1)

Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from matplotlib) (1.4.9)

Requirement already satisfied: numpy>=1.23 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from matplotlib) (2.3.4)

Requirement already satisfied: packaging>=20.0 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from matplotlib) (25.0)

Requirement already satisfied: pillow>=8 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from matplotlib) (12.0.0)

Requirement already satisfied: pyparsing>=3 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from matplotlib) (3.2.5)

Requirement already satisfied: python-dateutil>=2.7 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from matplotlib) (2.9.0.post0)

Requirement already satisfied: six>=1.5 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.17.0)

Requirement already satisfied: seaborn in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (0.13.2)

Requirement already satisfied: numpy!=1.24.0,>=1.20 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from seaborn) (2.3.4)

Requirement already satisfied: pandas>=1.2 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from seaborn) (2.3.3)

Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from seaborn) (3.10.7)

Requirement already satisfied: contourpy>=1.0.1 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.3.3)

Requirement already satisfied: cycler>=0.10 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (0.12.1)

Requirement already satisfied: fonttools>=4.22.0 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (4.60.1)

Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.4.9)

Requirement already satisfied: packaging>=20.0 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (25.0)

Requirement already satisfied: pillow>=8 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (12.0.0)

Requirement already satisfied: pyparsing>=3 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (3.2.5)

Requirement already satisfied: python-dateutil>=2.7 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (2.9.0.post0)

Requirement already satisfied: pytz>=2020.1 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from pandas>=1.2->seaborn) (2025.2)

Requirement already satisfied: tzdata>=2022.7 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from pandas>=1.2->seaborn) (2025.2)

Requirement already satisfied: six>=1.5 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.17.0)

```
In [2]: import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

```
In [3]: pwd
```

```
Out[3]: 'C:\\Users\\Eng-Hussam\\Downloads\\MAster Ai-Folder'
```

```
In [4]: # CONNECT TO THE DATA SET
df = pd.read_csv('insurance.csv')
```

```
In [5]: # TO PRINT 10 LINEFROM THE DATA SET
df.head(10)
```

Out[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
5	31	female	25.740	0	no	southeast	3756.62160
6	46	female	33.440	1	no	southeast	8240.58960
7	37	female	27.740	3	no	northwest	7281.50560
8	37	male	29.830	2	no	northeast	6406.41070
9	60	female	25.840	0	no	northwest	28923.13692

In [6]: *# to know the data type*
df.dtypes

Out[6]: age int64
sex object
bmi float64
children int64
smoker object
region object
charges float64
dtype: object

In [7]: *# to know the missing value*
df.isnull().sum()

```
Out[7]: age      0
sex      0
bmi      0
children 0
smoker   0
region   0
charges  0
dtype: int64
```

```
In [8]: df.isnull().mean()
```

```
Out[8]: age      0.0
sex      0.0
bmi      0.0
children 0.0
smoker   0.0
region   0.0
charges  0.0
dtype: float64
```

```
In [9]: # Numerical features (fill NaN with mean)
num_cols = df.select_dtypes(include=['number']).columns
df[num_cols] = df[num_cols].fillna(df[num_cols].mean())

# Categorical features (fill NaN with mode)
cat_cols = df.select_dtypes(include=['object']).columns
df[cat_cols] = df[cat_cols].fillna(df[cat_cols].mode().iloc[0])
```

```
In [10]: # check for duplicated data
df.duplicated().sum()
```

```
Out[10]: np.int64(1)
```

```
In [11]: # Remove Duplicated data
df = df.drop_duplicates()
```

```
In [12]: # Recheck Duplicated Data
df.duplicated().sum()
```

```
Out[12]: np.int64(0)
```

```
In [13]: # نسخة من البيانات
data = df.copy()

# باستخدام IQR دالة حساب
def detect_outliers_iqr(column):
    Q1 = column.quantile(0.25)
    Q3 = column.quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    return (column < lower_bound) | (column > upper_bound)

# --- (IQR المناسبة للـ) نحدد الأعمدة العددية فقط ---
numeric_columns = ['age', 'bmi', 'charges']

# outlier flag نضيف أعمدة
for col in numeric_columns:
    data[f'{col}_Outlier'] = detect_outliers_iqr(data[col])

# Outliers إظهار الصفوف التي فيها
outliers_only = data[data.filter(like='_Outlier').any(axis=1)]

outliers_only
```

Out[13]:

	age	sex	bmi	children	smoker	region	charges	age_Outlier	bmi_Outlier	charges_Outlier
14	27	male	42.130	0	yes	southeast	39611.75770	False	False	True
19	30	male	35.300	0	yes	southwest	36837.46700	False	False	True
23	34	female	31.920	1	yes	northeast	37701.87680	False	False	True
29	31	male	36.300	2	yes	southwest	38711.00000	False	False	True
30	22	male	35.600	0	yes	southwest	35585.57600	False	False	True
...	...	...	...	...	...	...	...	...	...	...
1301	62	male	30.875	3	yes	northwest	46718.16325	False	False	True
1303	43	male	27.800	0	yes	southwest	37829.72420	False	False	True
1313	19	female	34.700	2	yes	southwest	36397.57600	False	False	True
1317	18	male	53.130	0	no	southeast	1163.46270	False	True	False
1323	42	female	40.370	2	yes	southeast	43896.37630	False	False	True

145 rows × 10 columns

```
In [14]: outlier_columns = [col for col in data.columns if '_Outlier' in col]
outlier_columns
```

```
Out[14]: ['age_Outlier', 'bmi_Outlier', 'charges_Outlier']
```

```
In [15]: for col in outlier_columns:
          print(col, " → ", data[col].sum(), " Outliers")
          # من هذه الدالة عرفت ان قيمة الاوتلاير انوا منطقية وطبيعية كمعدل كتلة الجسم وقيمة فاتورة العلاج الطبي للتأمين
```

age_Outlier → 0 Outliers
bmi_Outlier → 9 Outliers
charges_Outlier → 139 Outliers

```
In [16]: data[outlier_columns].sum().sort_values(ascending=False)
```

```
Out[16]: charges_Outlier    139
         bmi_Outlier        9
         age_Outlier        0
         dtype: int64
```

```
In [17]: Q1 = df['bmi'].quantile(0.25)
         Q3 = df['bmi'].quantile(0.75)
         IQR = Q3 - Q1

         lower = Q1 - 1.5 * IQR
         upper = Q3 + 1.5 * IQR

         bmi_outliers = df[(df['bmi'] < lower) | (df['bmi'] > upper)]
         bmi_outliers
```

```
Out[17]:
```

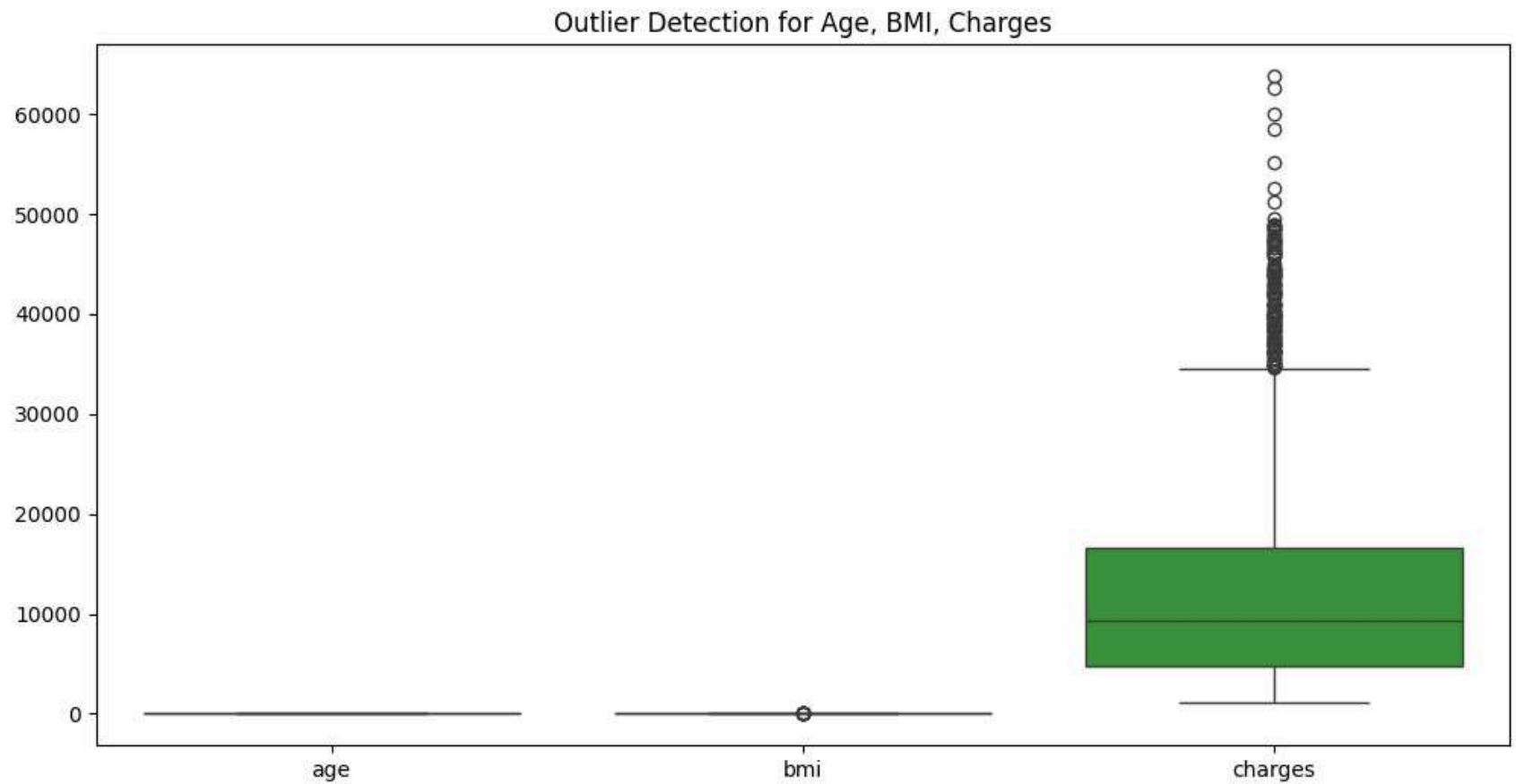
	age	sex	bmi	children	smoker	region	charges
116	58	male	49.06	0	no	southeast	11381.32540
286	46	female	48.07	2	no	northeast	9432.92530
401	47	male	47.52	1	no	southeast	8083.91980
543	54	female	47.41	0	yes	southeast	63770.42801
847	23	male	50.38	1	no	southeast	2438.05520
860	37	female	47.60	2	yes	southwest	46113.51100
1047	22	male	52.58	1	yes	southeast	44501.39820
1088	52	male	47.74	1	no	southeast	9748.91060
1317	18	male	53.13	0	no	southeast	1163.46270

```
In [18]: print("Q1 =", Q1)
         print("Q3 =", Q3)
         print("IQR =", IQR)
         print("Lower Bound =", lower)
         print("Upper Bound =", upper)
```

Q1 = 26.29
Q3 = 34.7
IQR = 8.410000000000004
Lower Bound = 13.674999999999994
Upper Bound = 47.315000000000001

```
In [19]: import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(12, 6))
sns.boxplot(data=df[['age', 'bmi', 'charges']])
plt.title("Outlier Detection for Age, BMI, Charges")
plt.show()
```



In [20]: `pip install scikit-learn`

Requirement already satisfied: scikit-learn in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (1.7.2)
 Requirement already satisfied: numpy>=1.22.0 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from scikit-learn) (2.3.4)
 Requirement already satisfied: scipy>=1.8.0 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from scikit-learn) (1.16.3)
 Requirement already satisfied: joblib>=1.2.0 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from scikit-learn) (1.5.2)
 Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\eng-hussam\anaconda3\envs\code1\lib\site-packages (from scikit-learn) (3.6.0)
 Note: you may need to restart the kernel to use updated packages.

In [21]: *# Encoding Categorical Columns Using LabelEncoder*
This step converts all categorical (object) columns into numerical
values so machine learning models can understand the data.
-----

```
from sklearn.preprocessing import LabelEncoder

# Create the encoder object
label_encoder = LabelEncoder()

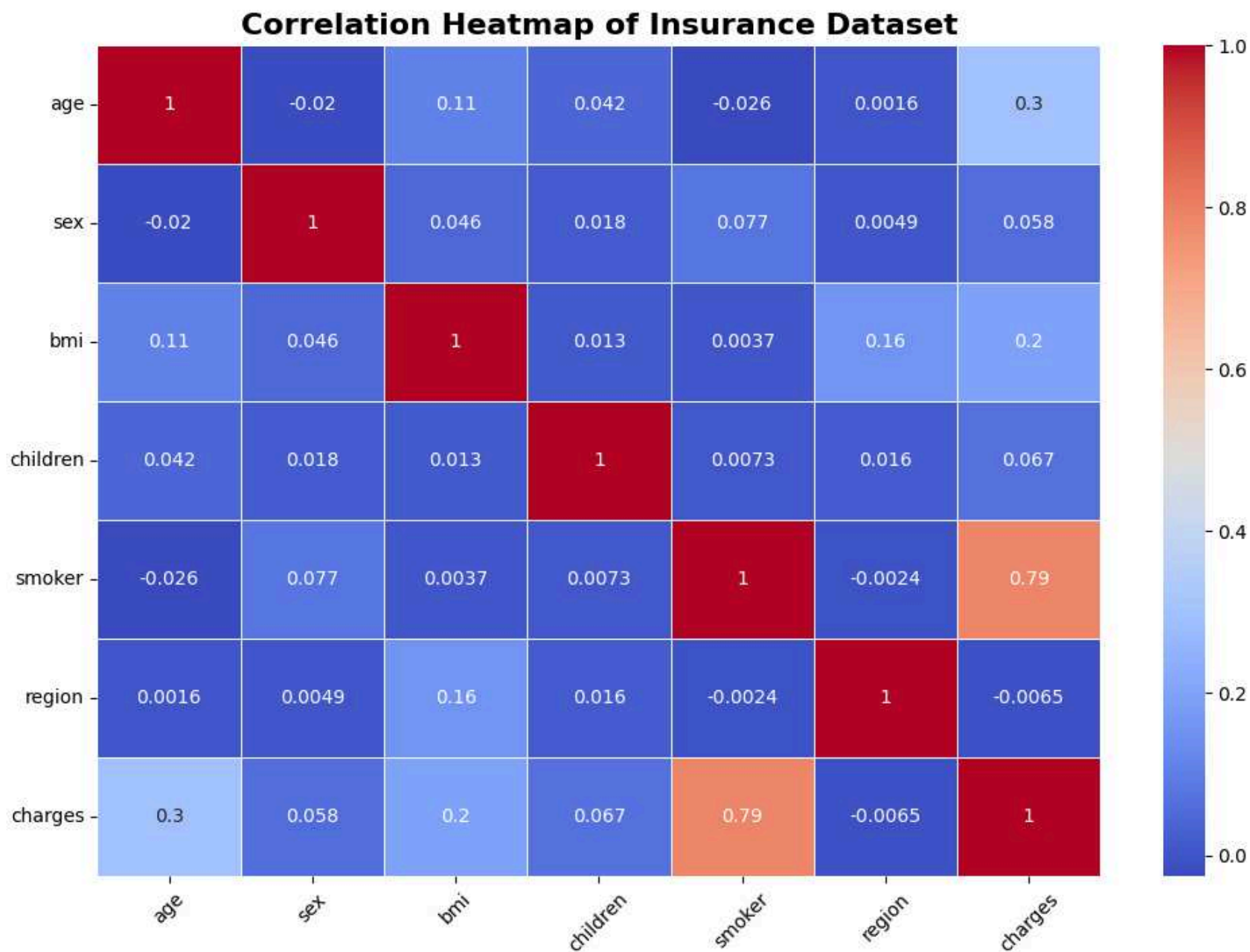
# Loop through all categorical columns and encode them
for column in df.select_dtypes(include=['object']).columns:
    df[column] = label_encoder.fit_transform(df[column])

# Display first 5 rows after encoding
df.head()
```

Out[21]:

	age	sex	bmi	children	smoker	region	charges
0	19	0	27.900	0	1	3	16884.92400
1	18	1	33.770	1	0	2	1725.55230
2	28	1	33.000	3	0	2	4449.46200
3	33	1	22.705	0	0	1	21984.47061
4	32	1	28.880	0	0	1	3866.85520

```
In [22]: # Plot a correlation heatmap to visualize the strength and direction of relationships between numerical features using
plt.figure(figsize=(12, 8))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', linewidths=0.5, linecolor='white')
plt.title('Correlation Heatmap of Insurance Dataset', fontsize=16, fontweight='bold')
plt.xticks(rotation=45)
plt.yticks(rotation=0)
plt.show()
```



```
In [23]: df['smoker'].value_counts()
```

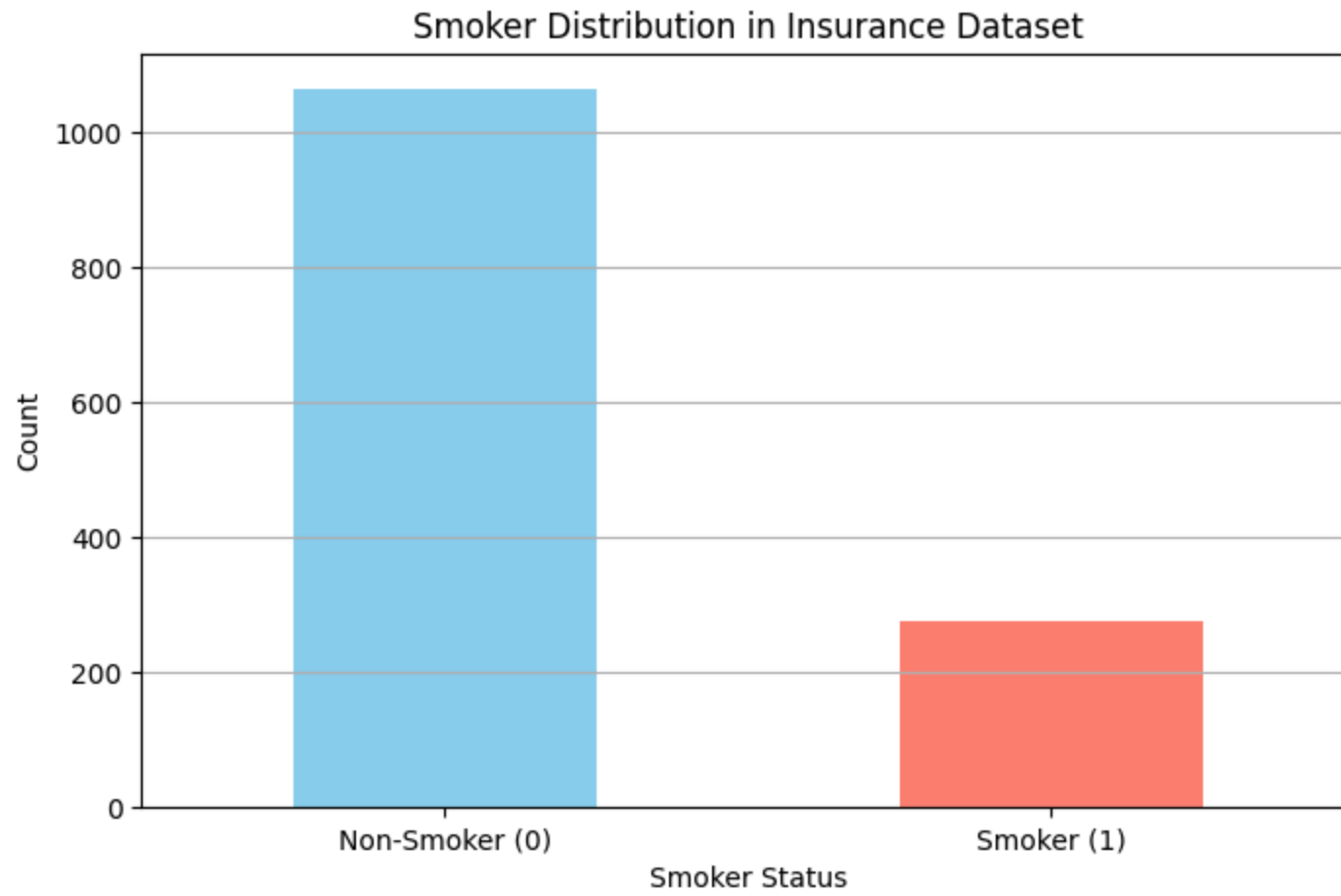
```
Out[23]: smoker
0      1063
1       274
Name: count, dtype: int64
```

```
In [24]: # Create a bar chart
plt.figure(figsize=(8, 5))

df['smoker'].value_counts().plot(kind='bar', color=['skyblue', 'salmon'])

plt.xticks([0, 1], ['Non-Smoker (0)', 'Smoker (1)'], rotation=0)
plt.title('Smoker Distribution in Insurance Dataset')
plt.xlabel('Smoker Status')
plt.ylabel('Count')
plt.grid(axis='y')

plt.show()
```

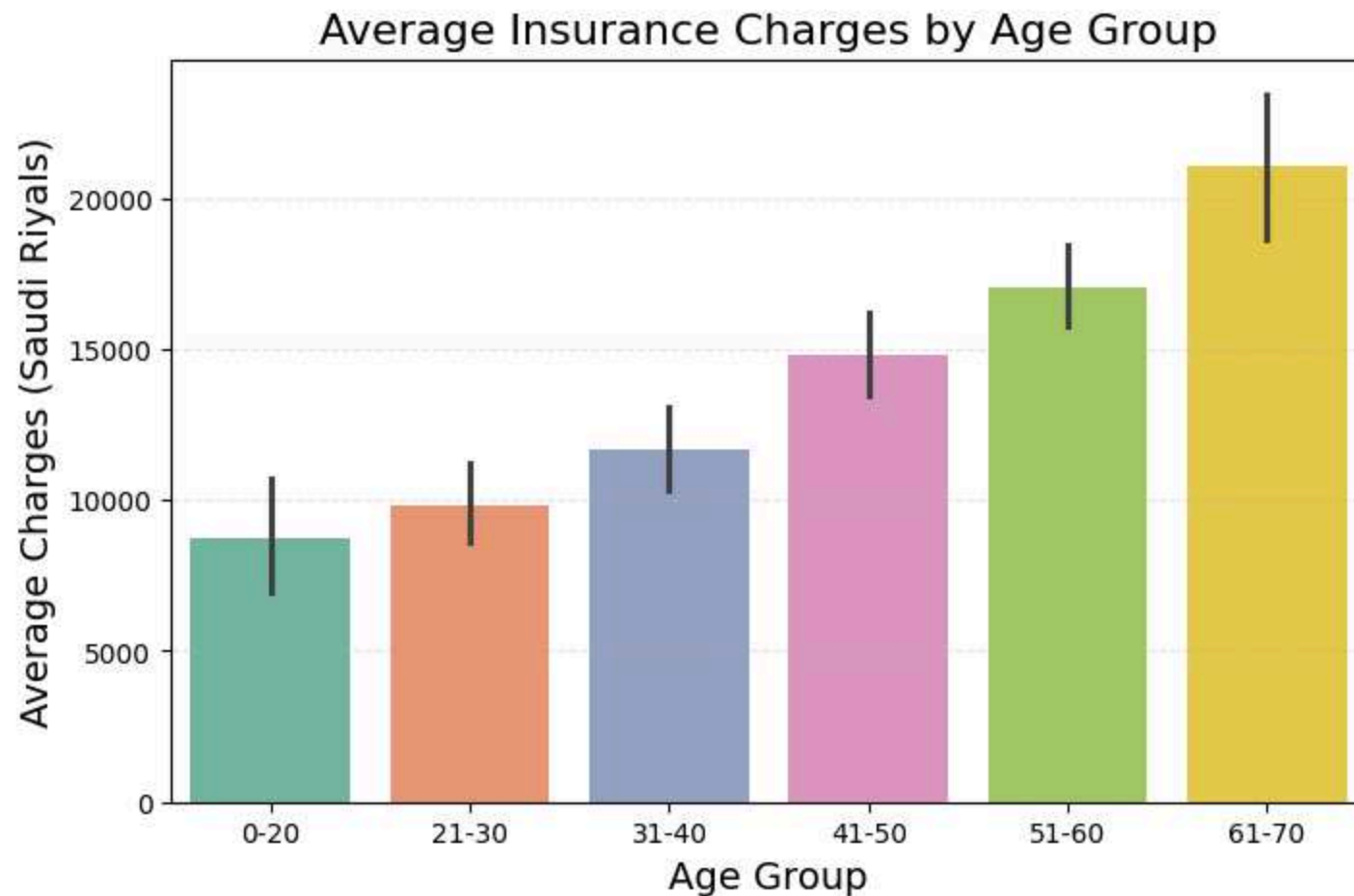


```
In [28]: plt.figure(figsize=(8,5))
sns.barplot(
    x='age_group',
    y='charges',
    hue='age_group',
    data=df,
    estimator='mean',
    palette='Set2',
    legend=False
)

plt.title("Average Insurance Charges by Age Group", fontsize=16)
plt.xlabel("Age Group", fontsize=14)
```

```
plt.ylabel("Average Charges (Saudi Riyals)", fontsize=14)
plt.grid(axis='y', linestyle='--', alpha=0.3)

plt.show()
```



In [25]: `import matplotlib.pyplot as plt`

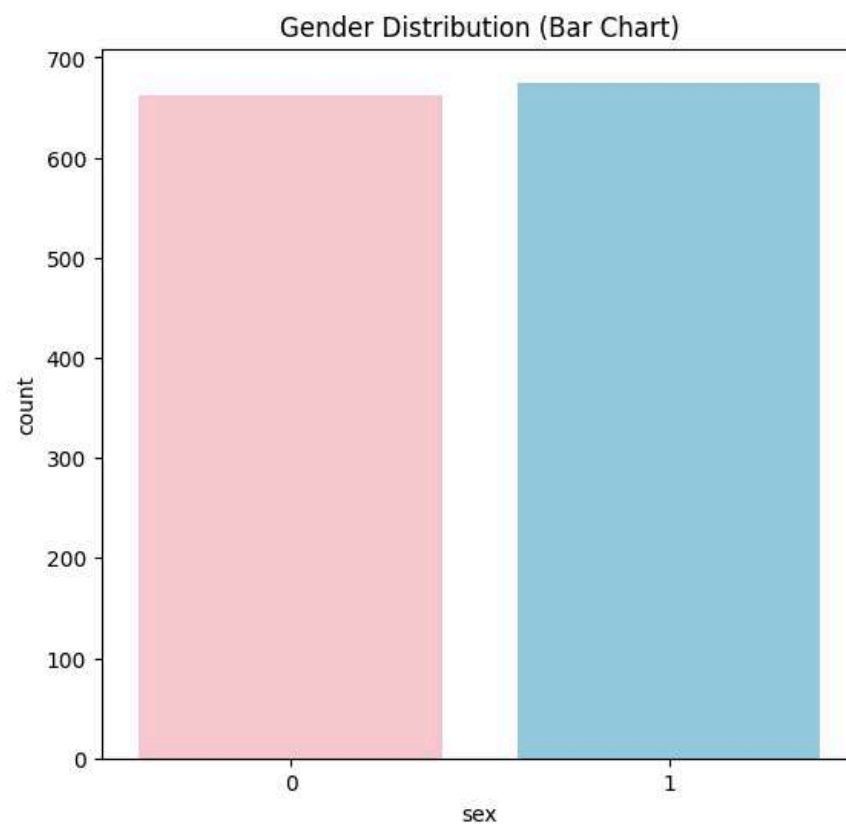
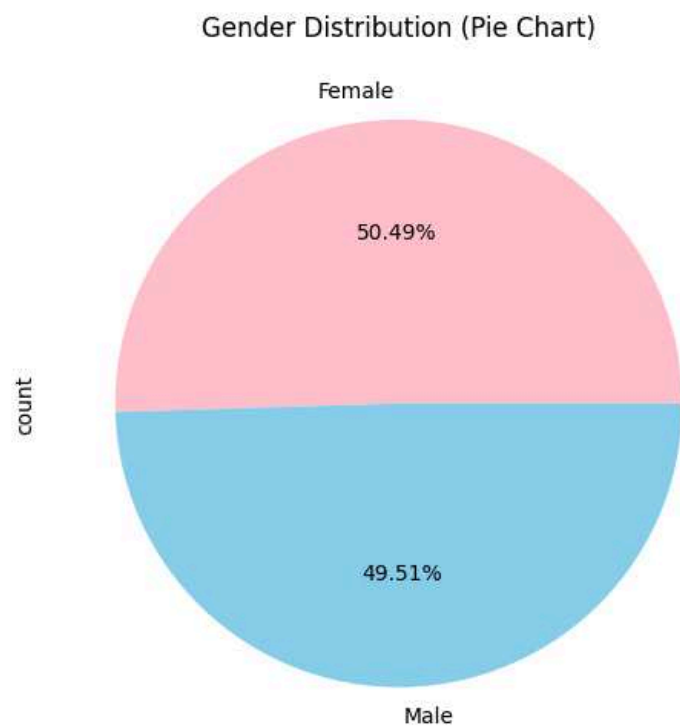
```
# Pie Chart
plt.figure(figsize=(14,6))

plt.subplot(1,2,1)
df['sex'].value_counts().plot.pie(
    autopct='%0.2f%%', colors=['pink','skyblue'], labels=['Female','Male'])
```

```
plt.title("Gender Distribution (Pie Chart)")

# Bar Chart
plt.subplot(1,2,2)
sns.countplot(x='sex', data=df, palette=['pink','skyblue'])
plt.title("Gender Distribution (Bar Chart)")

plt.show()
```



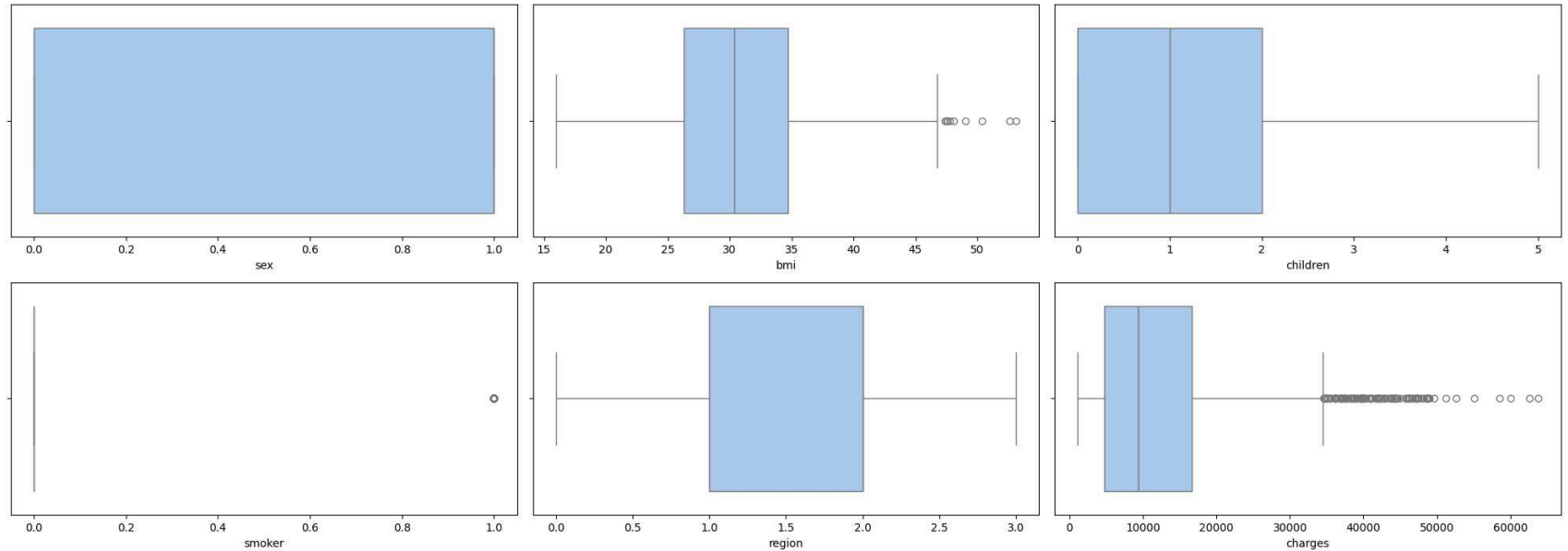
```
In [26]: # Create a large figure to display multiple boxplots
plt.figure(figsize=(20, 35))
# Counter to keep track of subplot positions
plotnumber = 1

for column in df.columns[1:]:
    ax = plt.subplot(10, 3, plotnumber)
```

```

sns.boxplot(data=df, x=column, palette='pastel')
plt.xlabel(column)
plotnumber += 1
# Adjust layout to prevent overlapping of subplots and ensure better spacing.
plt.tight_layout()
#Display all the generated plots
plt.show()

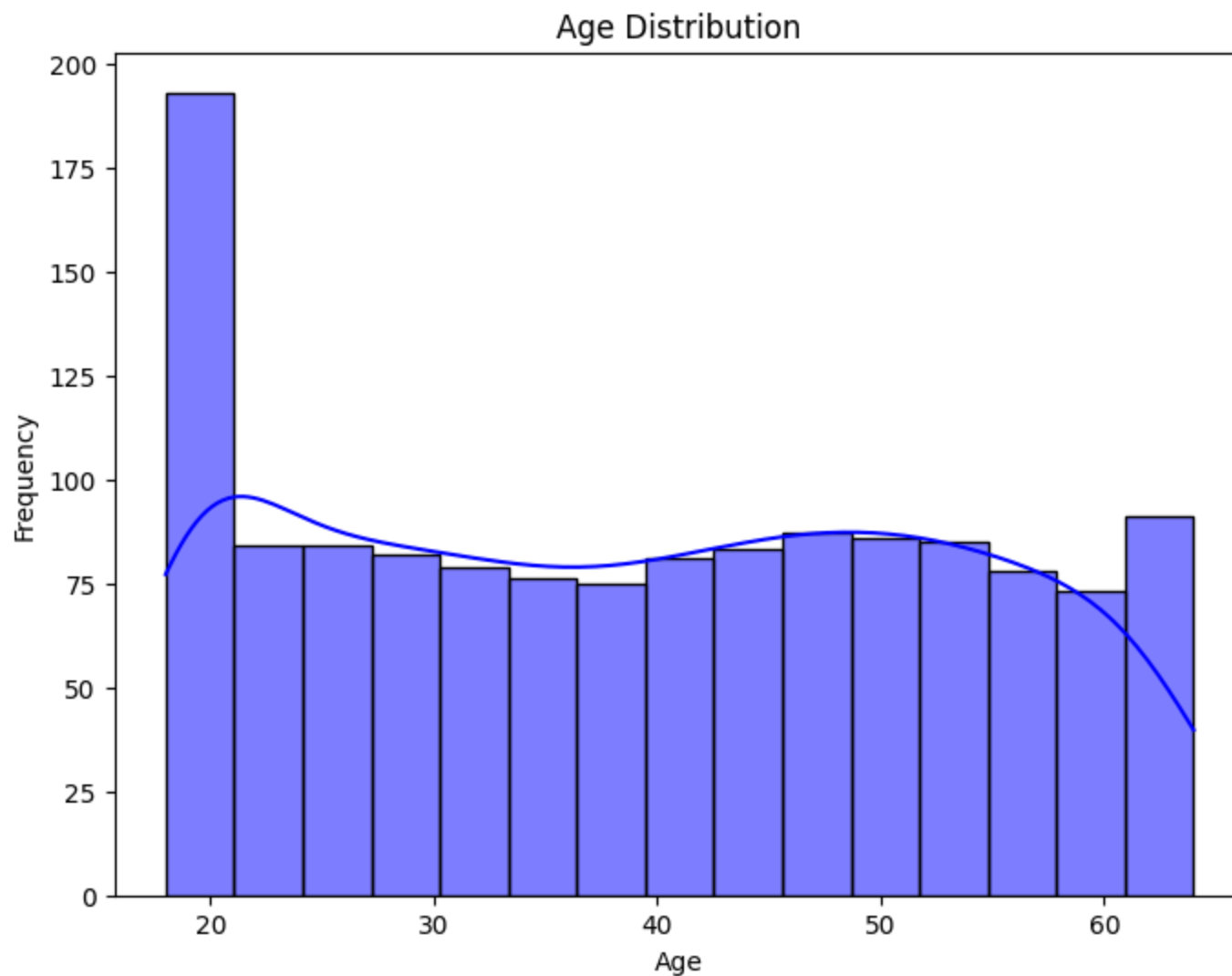
```



```

In [36]: # 1. Distribution of Age
plt.figure(figsize=(8, 6))
sns.histplot(df['age'], bins=15, kde=True, color='blue')
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()

```

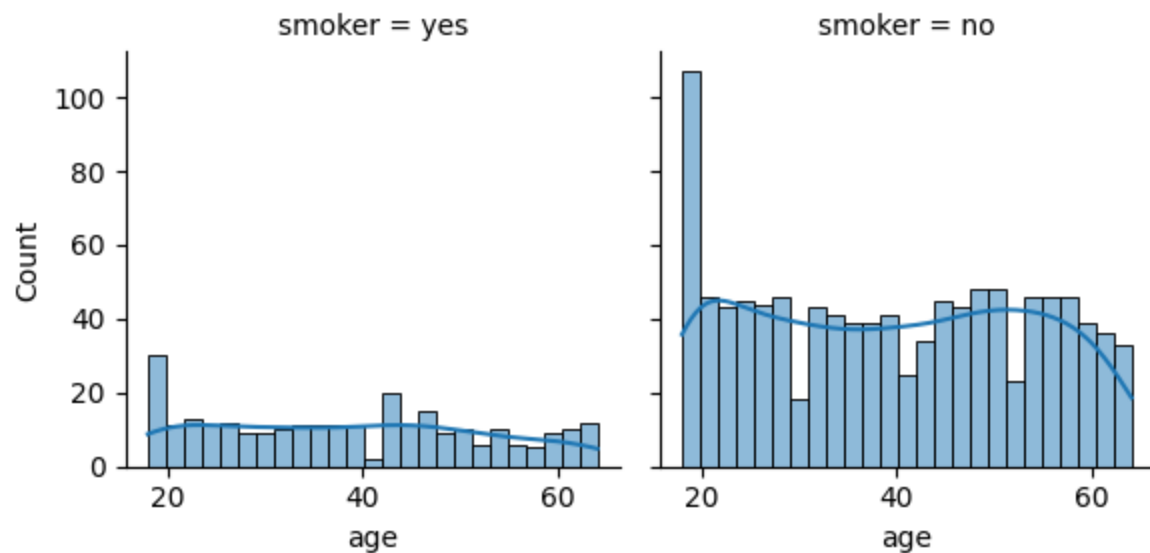


```
In [34]: cat_cols = [col for col in df.columns if df[col].dtype == 'object'] # categorical columns
          num_cols = [col for col in df.columns if df[col].dtype != 'object'] # numerical columns
```

```
In [50]: import seaborn as sns
          import matplotlib.pyplot as plt

          g = sns.FacetGrid(df, col="smoker")
          g.map(sns.histplot, "age", bins=25, kde=True)
```

```
plt.show()
```



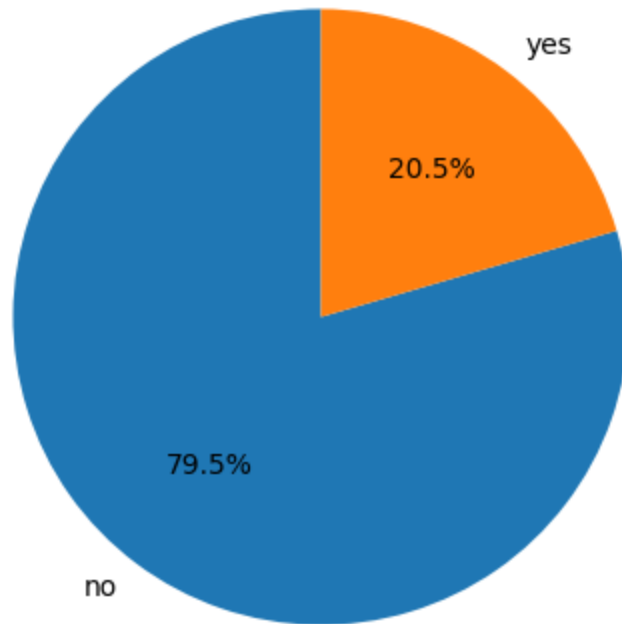
```
In [53]: import matplotlib.pyplot as plt

# حساب عدد القيم في العمود
smoker_counts = df['smoker'].value_counts()

# رسم Pie Chart
plt.figure(figsize=(5, 5))
plt.pie(smoker_counts,
        labels=smoker_counts.index,
        autopct='%1.1f%%',
        startangle=90)

plt.title("Smoker Status Distribution")
plt.show()
```

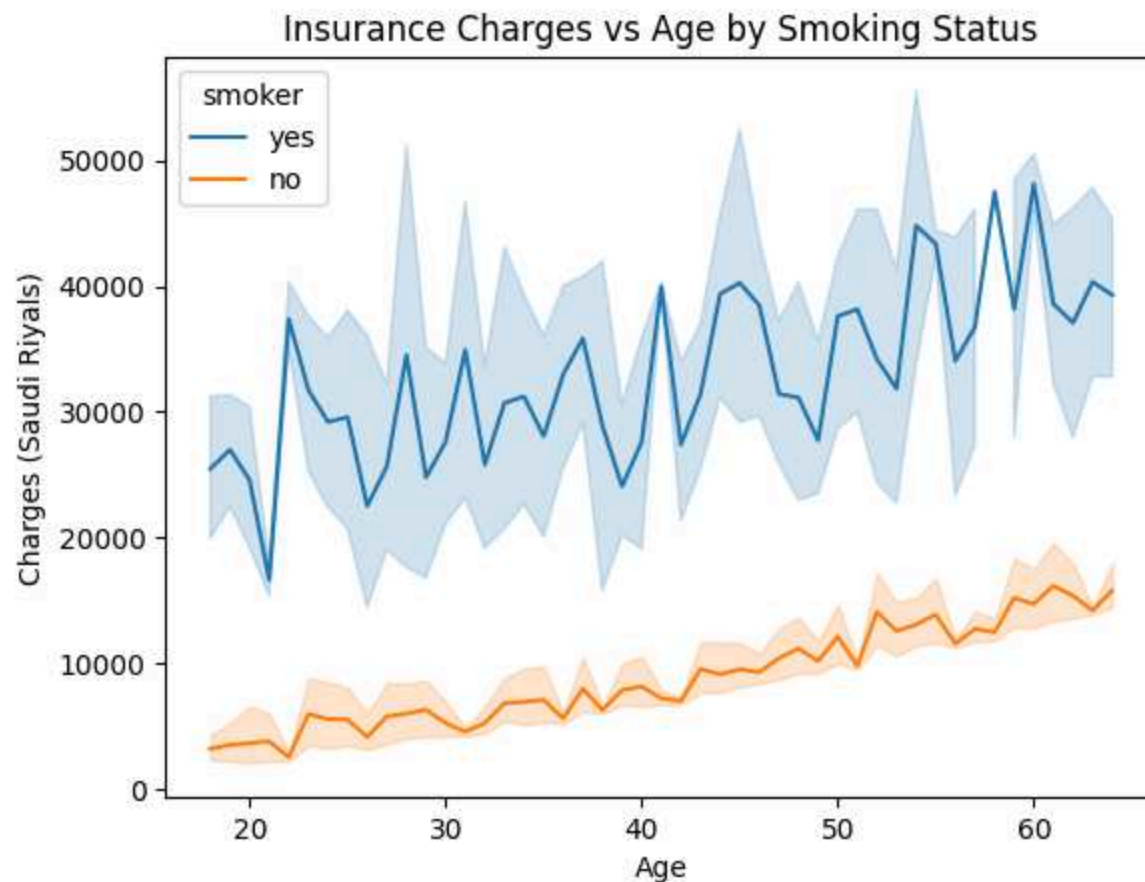
Smoker Status Distribution



```
In [54]: sns.lineplot(x='age', y='charges', hue='smoker', data=df)

plt.title("Insurance Charges vs Age by Smoking Status")
plt.xlabel("Age")
plt.ylabel("Charges (Saudi Riyals)")

plt.show()
```

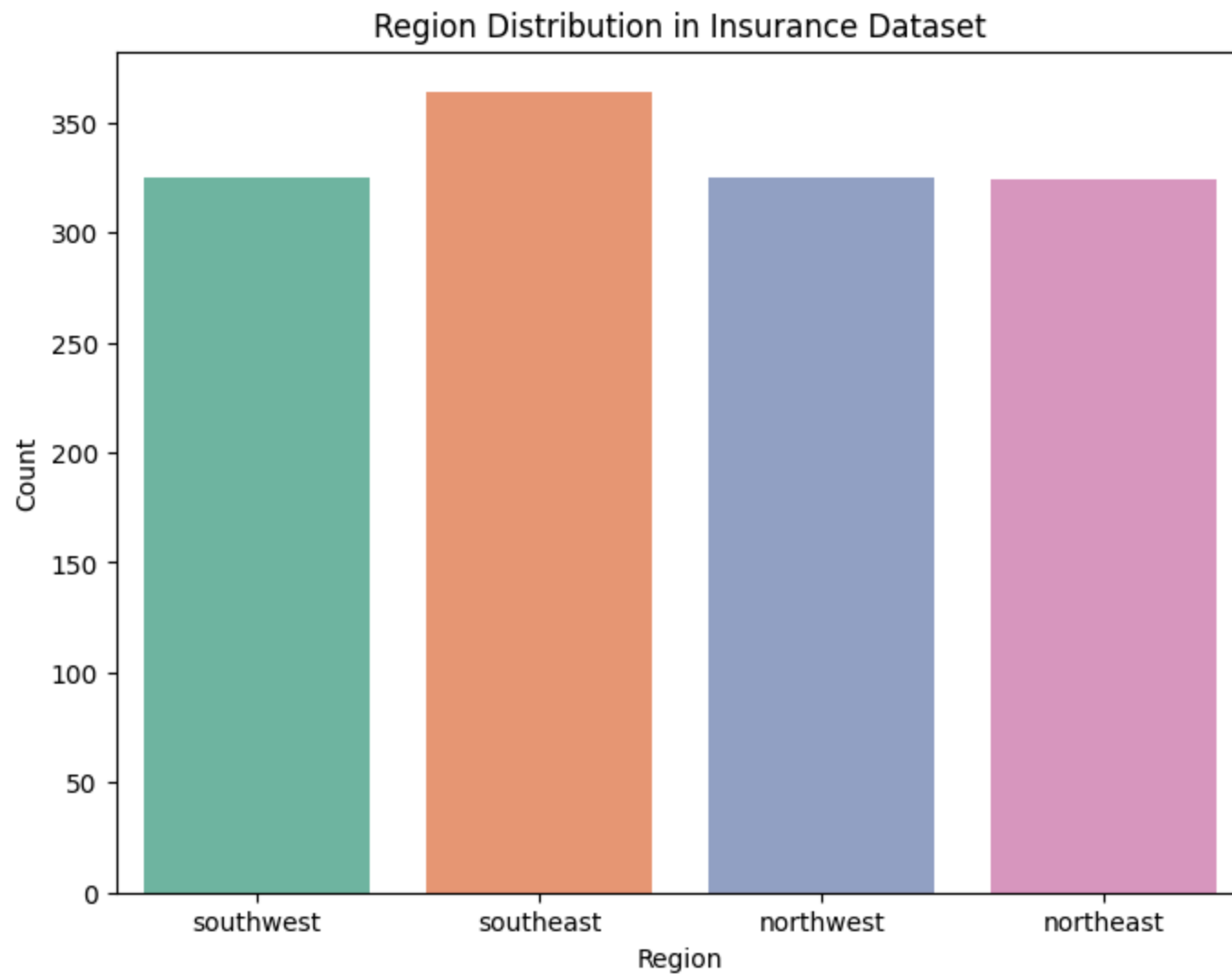


```
In [56]: plt.figure(figsize=(8, 6))

sns.countplot(
    x='region',
    hue='region',
    data=df,
    palette='Set2',
    legend=False
)

plt.title('Region Distribution in Insurance Dataset')
plt.xlabel('Region')
plt.ylabel('Count')

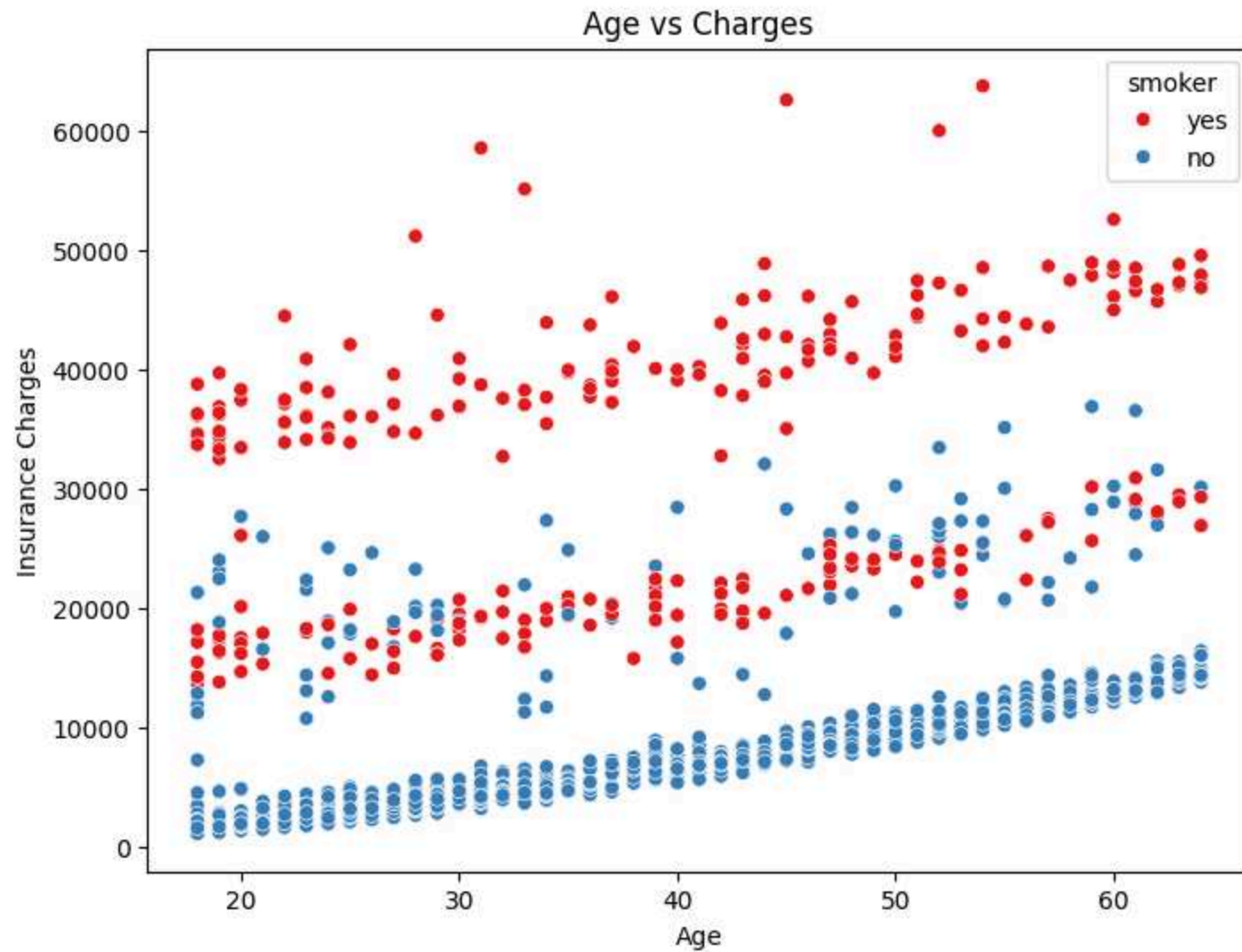
plt.show()
```



```
In [57]: plt.figure(figsize=(8, 6))
sns.scatterplot(
    x='age',
    y='charges',
    data=df,
    hue='smoker',
    palette='Set1')

plt.title('Age vs Charges')
```

```
plt.xlabel('Age')  
plt.ylabel('Insurance Charges')  
plt.show()
```



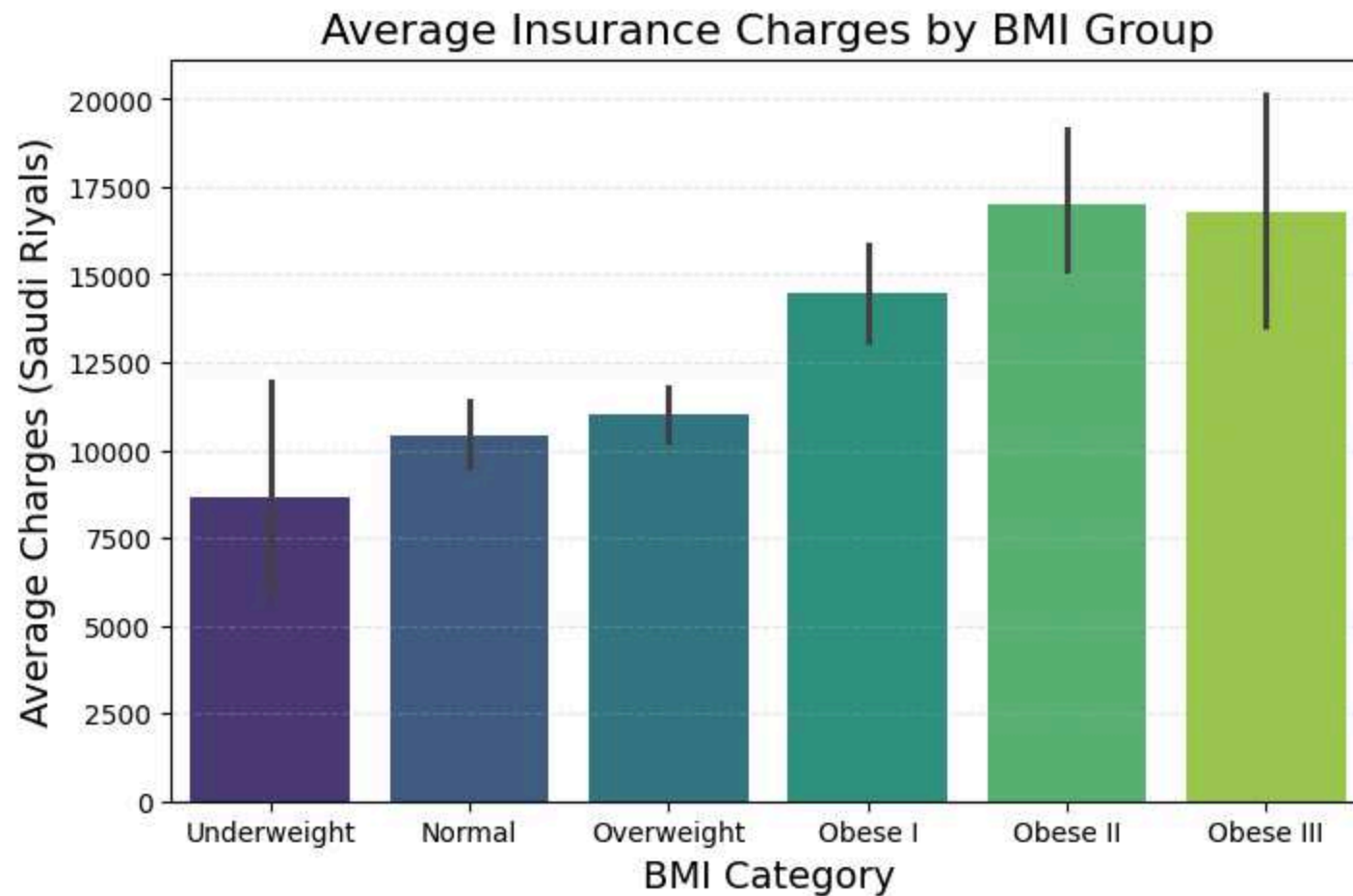
```
In [30]: import seaborn as sns  
import matplotlib.pyplot as plt  
import pandas as pd  
  
# Create BMI groups (bins)
```

```
df['bmi_group'] = pd.cut(df['bmi'],
                        bins=[0, 18.5, 25, 30, 35, 40, 100],
                        labels=['Underweight', 'Normal', 'Overweight', 'Obese I', 'Obese II', 'Obese III'])

# Plot average charges by BMI group
plt.figure(figsize=(8,5))
sns.barplot(
    x='bmi_group',
    y='charges',
    data=df,
    hue='bmi_group',
    estimator='mean',
    palette='viridis',
    legend=False
)

plt.title("Average Insurance Charges by BMI Group", fontsize=16)
plt.xlabel("BMI Category", fontsize=14)
plt.ylabel("Average Charges (Saudi Riyals)", fontsize=14)
plt.grid(axis='y', linestyle='--', alpha=0.3)

plt.show()
```



```
In [68]: # هنا عملت جروب للأعمار
df['bmi_group'] = pd.cut(
    df['bmi'],
    bins=[0, 18.5, 25, 30, 35, 40, 100],
    labels=['Underweight', 'Normal', 'Overweight', 'Obese I', 'Obese II', 'Obese III'])

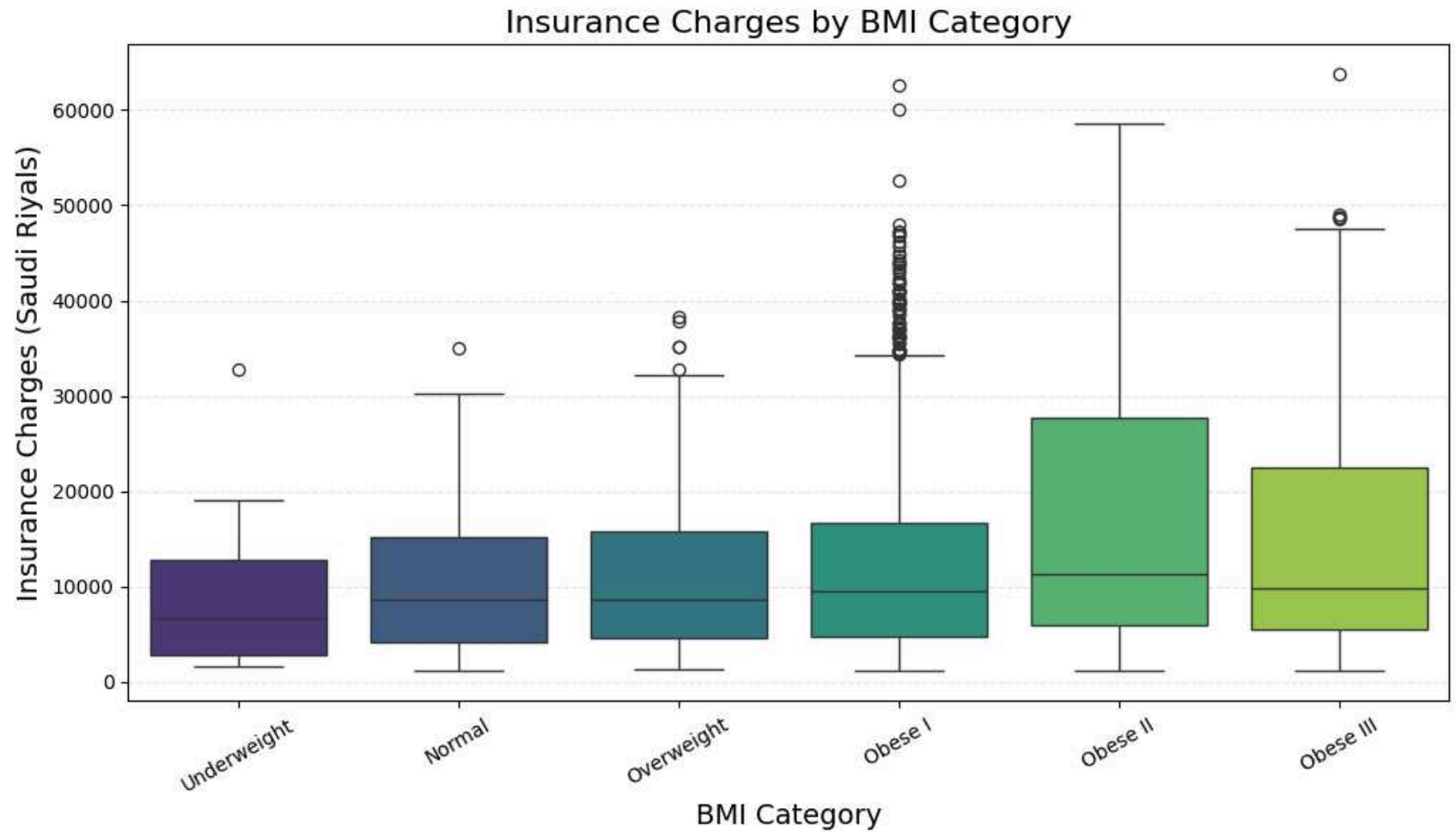
plt.figure(figsize=(12, 6))

sns.boxplot(x='bmi_group', y='charges', data=df, hue='bmi_group', palette='viridis', dodge=False, legend=False)

plt.title('Insurance Charges by BMI Category', fontsize=16)
plt.xlabel('BMI Category', fontsize=14)
plt.ylabel('Insurance Charges (Saudi Riyals)', fontsize=14)
```

```
plt.xticks(rotation=30)
plt.grid(axis='y', linestyle='--', alpha=0.3)

plt.show()
```



In []:

In []: