In [1]:
```python
# This Python 3 environment comes with many helpful analytics libraries instal
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/d
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will lis

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that ge
# You can also write temporary files to /kaggle/temp/, but they won't be savec
```

```
/kaggle/input/data-set/insurance.csv
```

In [2]:
```python
df = pd.read_csv("/kaggle/input/data-set/insurance.csv")
```

In [3]:
```python
df.head()
```

Out[3]:

	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

In [4]:
```python
df.tail()
```

Out[4]:

	age	sex	bmi	children	smoker	region	charges
1333	50	male	30.97	3	no	northwest	10600.5483
1334	18	female	31.92	0	no	northeast	2205.9808
1335	18	female	36.85	0	no	southeast	1629.8335
1336	21	female	25.80	0	no	southwest	2007.9450
1337	61	female	29.07	0	yes	northwest	29141.3603

In [5]:
```python
df.shape
```

Out[5]:
```
(1338, 7)
```

In [6]:

```
df.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
```

In [7]:

```
df.describe()
```

Out[7]:

	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

In [8]:

```
df.dtypes
```

Out[8]:

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

In [9]:

```
df.isnull().sum()
```

```
Out[9]: age         0
        sex         0
        bmi         0
        children    0
        smoker      0
        region      0
        charges     0
        dtype: int64
```

```
In [10]: df.duplicated().sum()
```

```
Out[10]: 1
```

```
In [11]: df = df.drop_duplicates()

         # Reset index after dropping
         df = df.reset_index(drop=True)
```

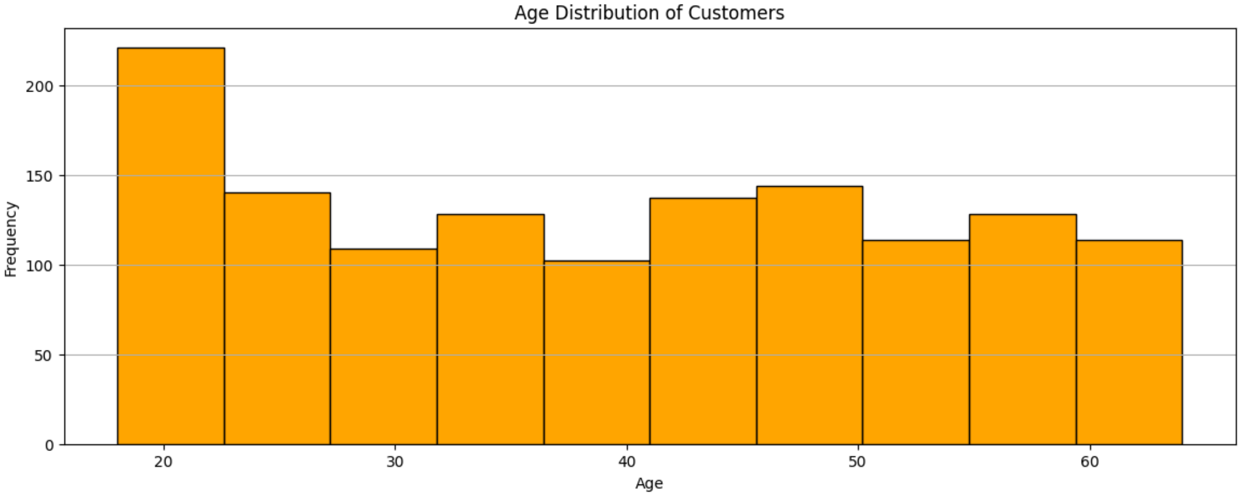
```
In [12]: df.duplicated().sum()
```

```
Out[12]: 0
```

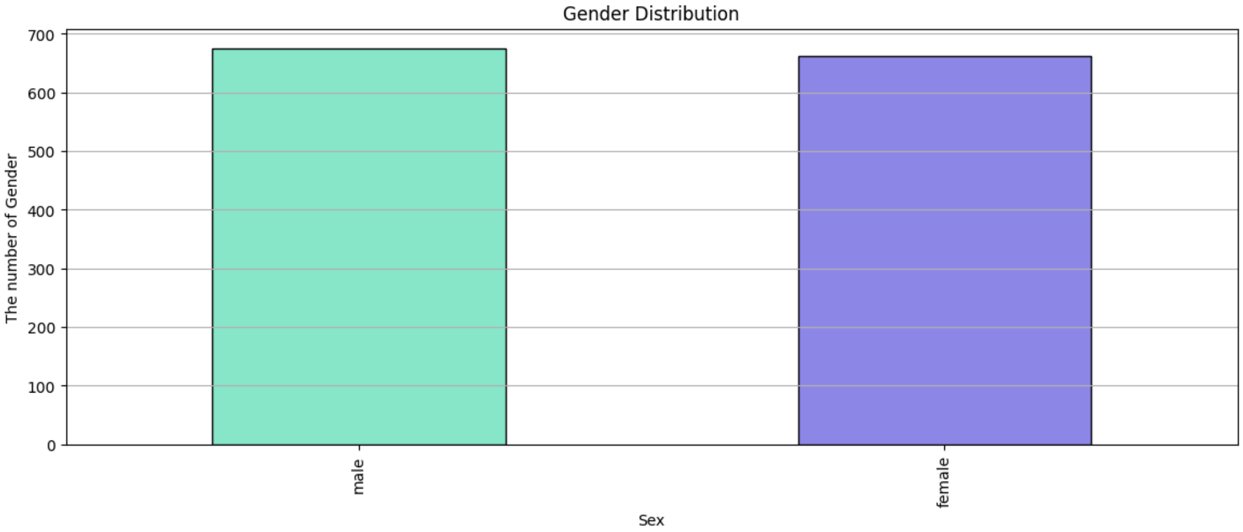
```
In [13]: df.columns
```

```
Out[13]: Index(['age', 'sex', 'bmi', 'children', 'smoker', 'region', 'charges'], dtyp
         e='object')
```

```
In [14]: plt.figure(figsize = (14,5))
         df["age"].plot(kind="hist", bins=10, color="orange", edgecolor="black")
         plt.title("Age Distribution of Customers")
         plt.xlabel("Age")
         plt.grid(axis = "y")
         plt.show()
```



```
In [15]: plt.figure(figsize = (14,5))
         df["sex"].value_counts().plot(kind = "bar", color = ["#8BE8CB","#908BE8"], edg
         plt.ylabel("The number of Gender")
         plt.xlabel("Sex")
         plt.title("Gender Distribution")
         plt.grid(axis = "y")
         plt.show()
```



In [16]:

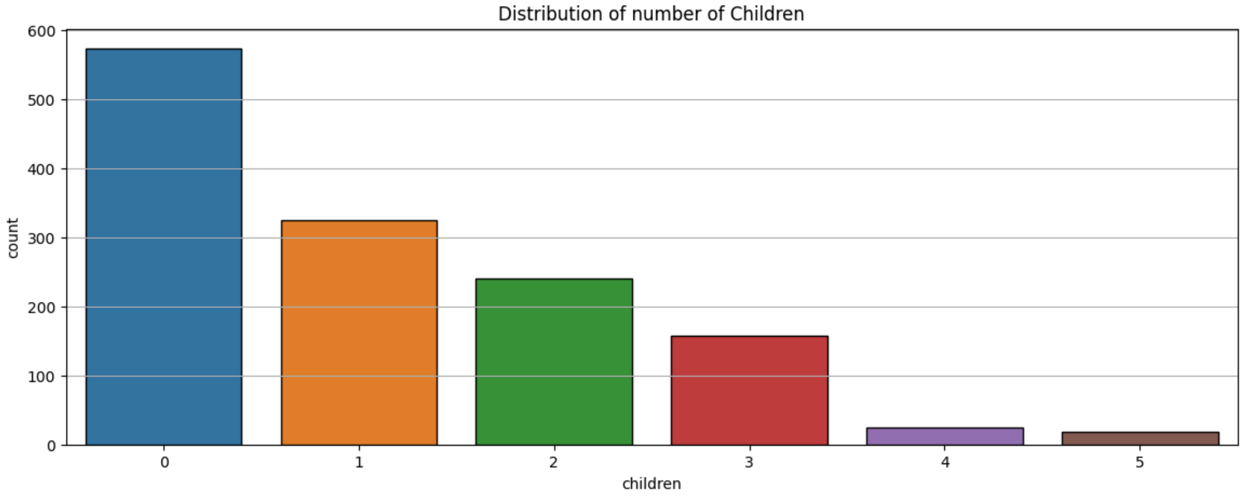
```
df["children"].value_counts()
```

Out[16]:

```
children
0    573
1    324
2    240
3    157
4     25
5     18
Name: count, dtype: int64
```

In [17]:

```
plt.figure(figsize = (14,5))
sns.countplot(x="children", data=df, edgecolor = "black")
plt.title("Distribution of number of Children")
plt.grid(axis = "y")
plt.show()
```



In [18]:

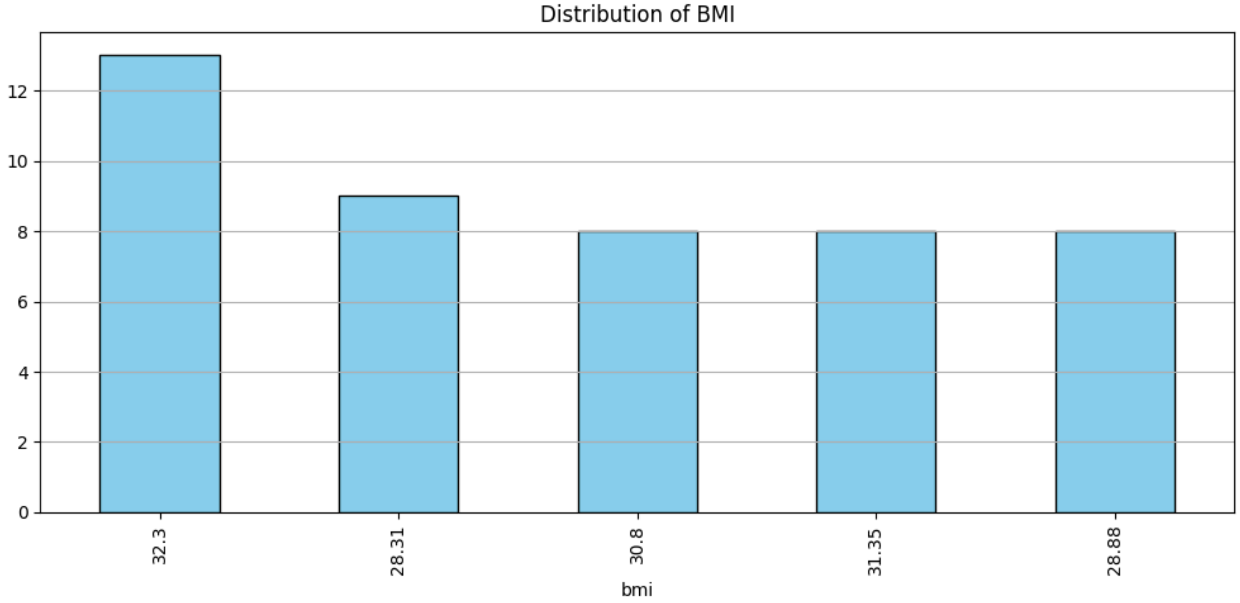
```
df["bmi"].value_counts().head()
```

Out[18]:

```
bmi
32.30    13
28.31     9
30.80     8
31.35     8
28.88     8
Name: count, dtype: int64
```

In [19]:

```
plt.figure(figsize = (10,5))
df["bmi"].value_counts().head().plot(kind="bar", color="skyblue", edgecolor="b
plt.title("Distribution of BMI")
plt.tight_layout()
plt.grid(axis = "y")
plt.show()
```



In [20]:

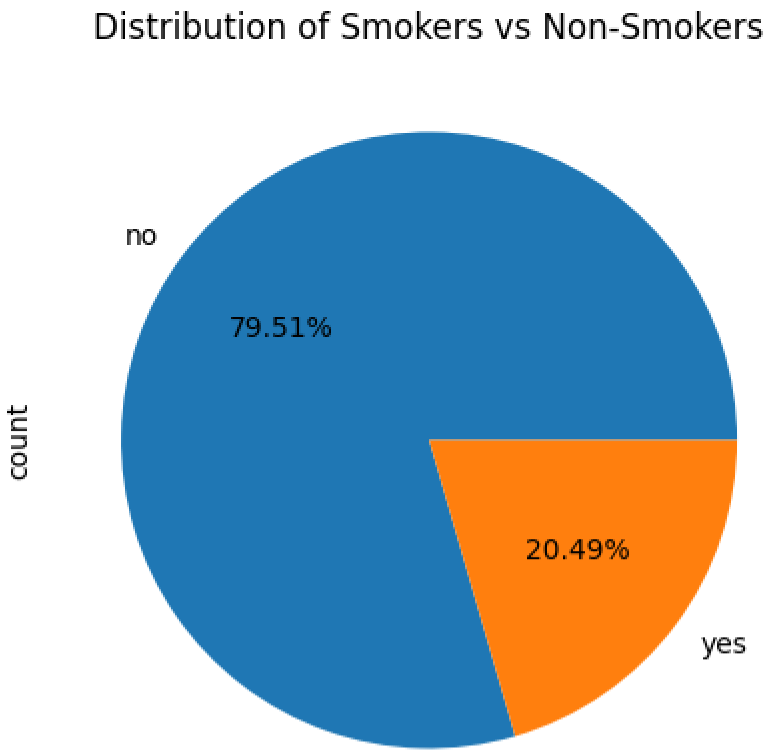
```
df.columns
```

Out[20]:

```
Index(['age', 'sex', 'bmi', 'children', 'smoker', 'region', 'charges'], dtyp
e='object')
```

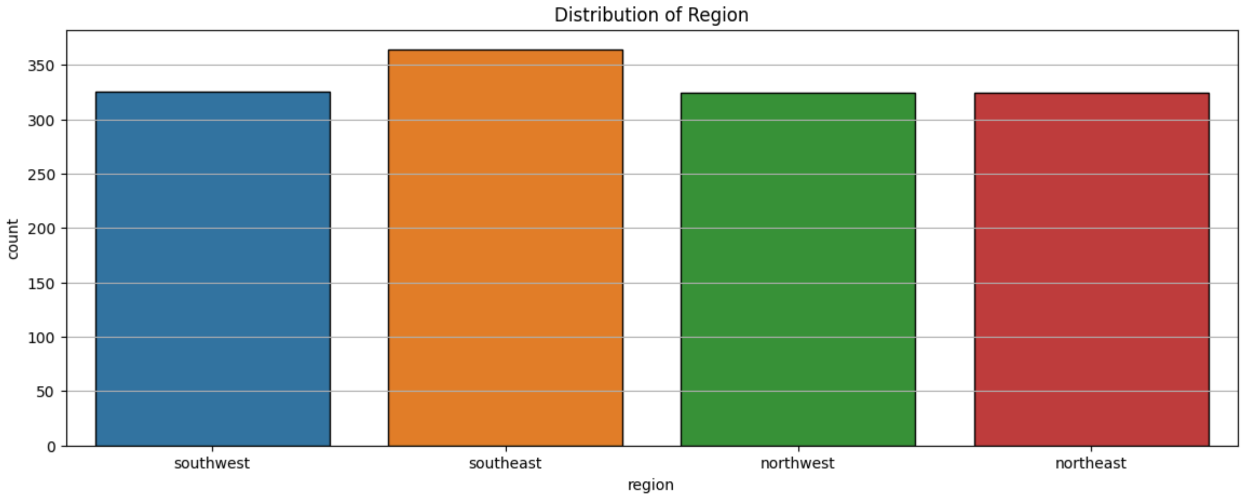
In [21]:

```
plt.figure(figsize=(14,5))
df["smoker"].value_counts().plot(kind="pie", autopct="%1.2f%%")
plt.title("Distribution of Smokers vs Non-Smokers")
plt.show()
```



In [22]:

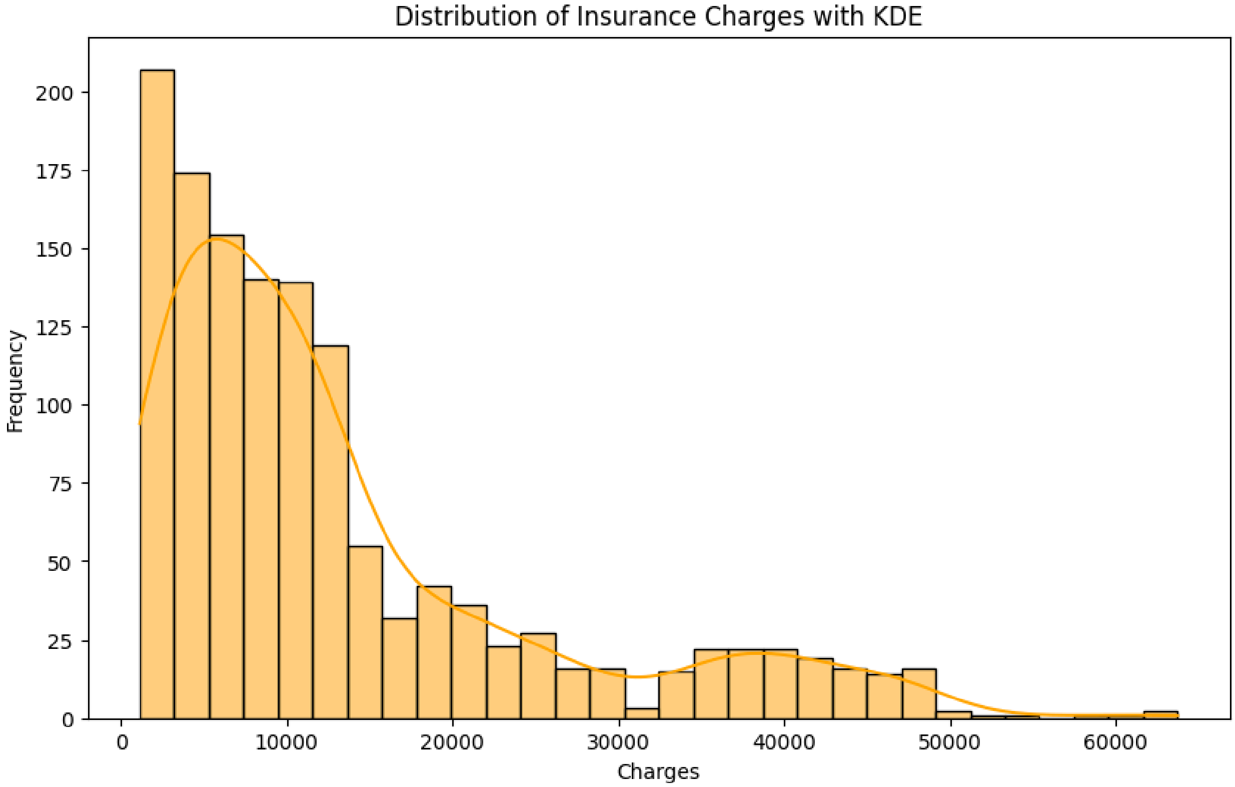
```
plt.figure(figsize = (14,5))
sns.countplot(x="region", data=df, edgecolor = "black")
plt.title("Distribution of Region")
plt.grid(axis = "y")
plt.show()
```



In [23]:

```
plt.figure(figsize=(10,6))
sns.histplot(df["charges"], bins=30, kde=True, color="orange", edgecolor="blac
plt.title("Distribution of Insurance Charges with KDE")
plt.xlabel("Charges")
plt.ylabel("Frequency")
plt.show()
```

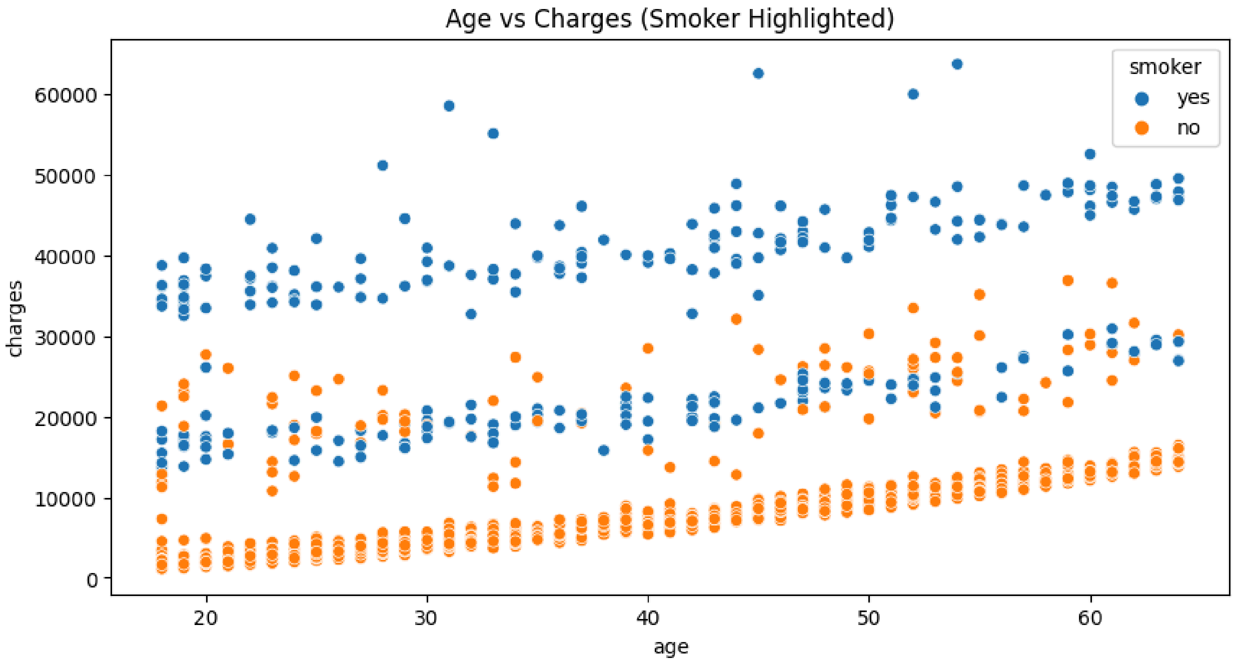
```
/usr/local/lib/python3.11/dist-packages/seaborn/_oldcore.py:1119: FutureWarnin
g: use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



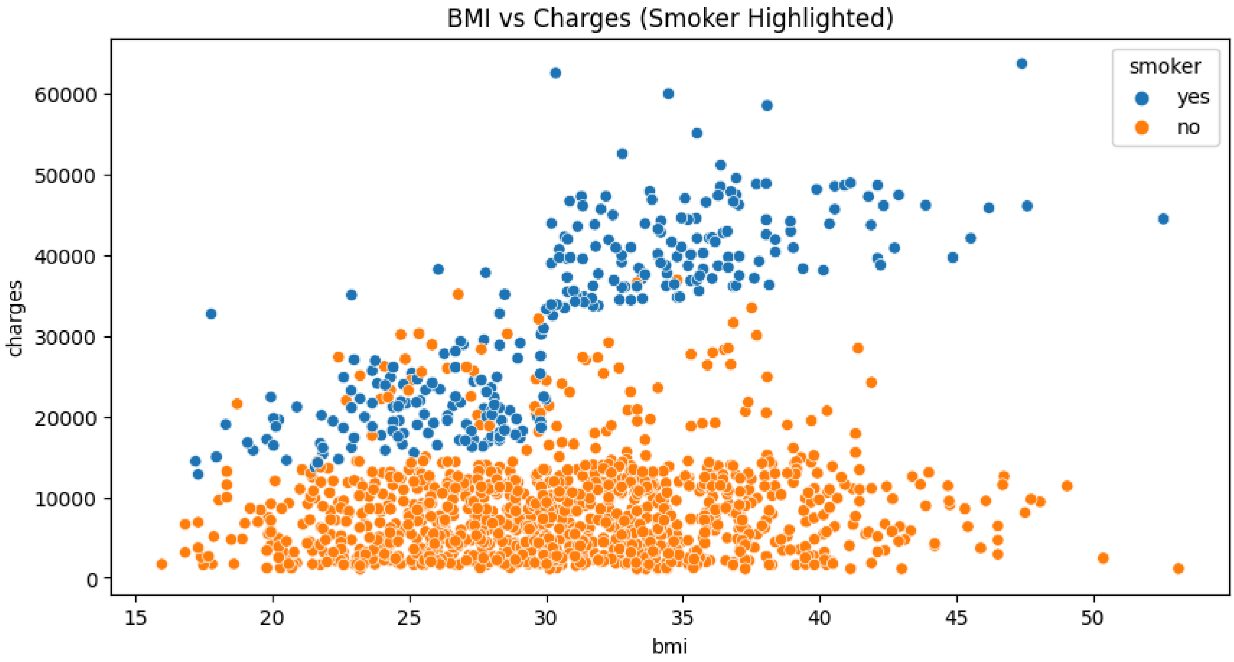
```
In [24]: df.columns
```

```
Out[24]: Index(['age', 'sex', 'bmi', 'children', 'smoker', 'region', 'charges'], dtyp
         e='object')
```

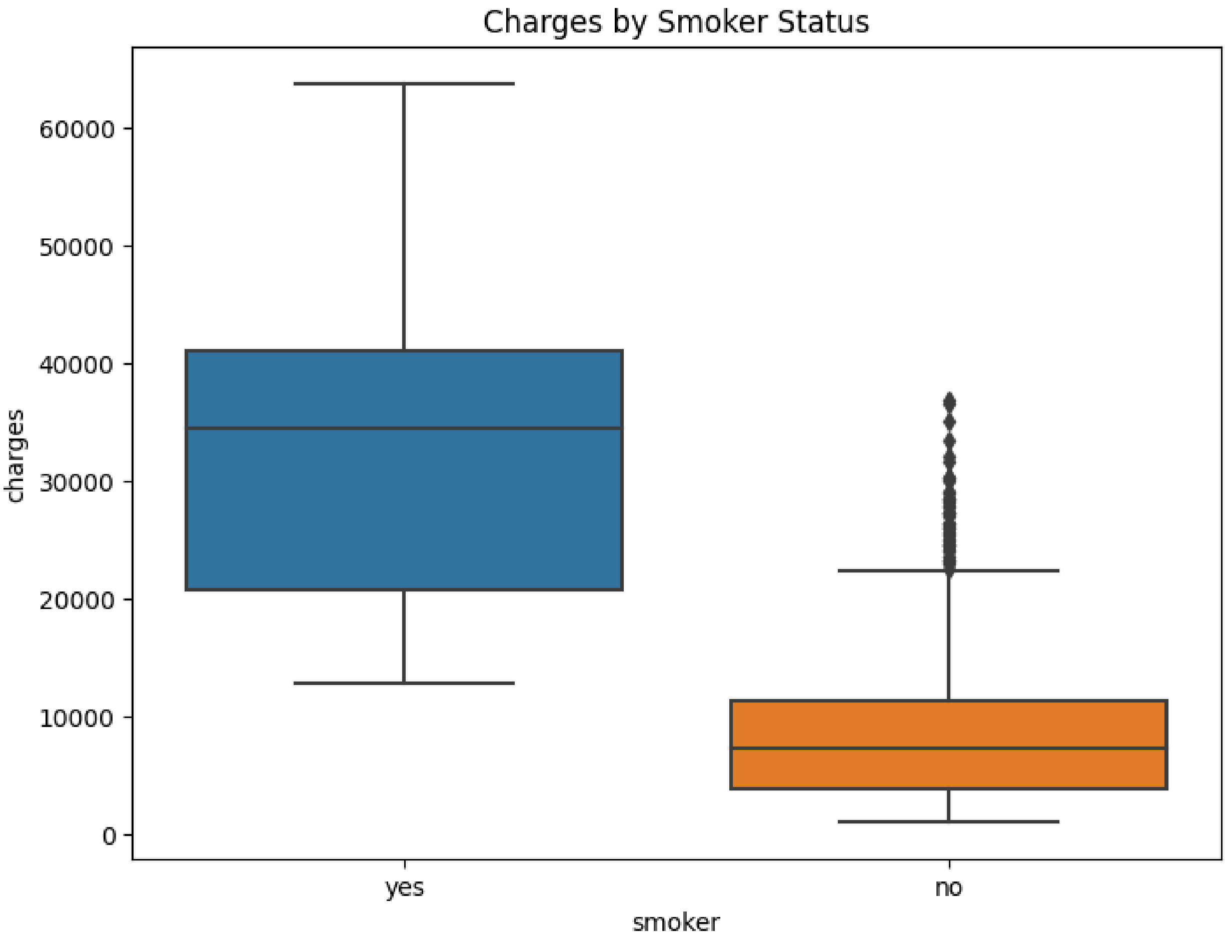
```
In [25]: plt.figure(figsize=(10,5))
         sns.scatterplot(x='age', y='charges', data=df, hue='smoker')
         plt.title("Age vs Charges (Smoker Highlighted)")
         plt.show()
```


Age vs Charges (Smoker Highlighted)

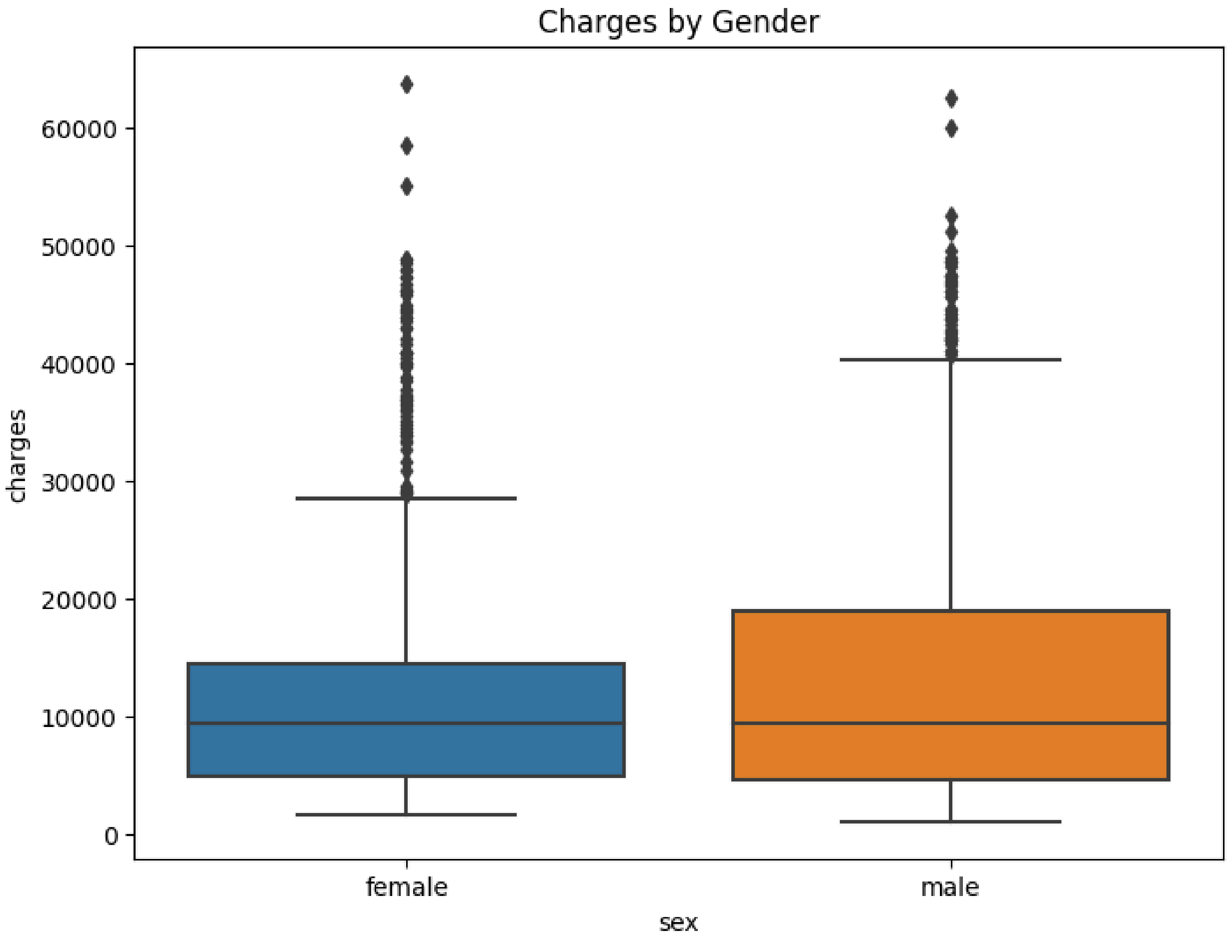
In [26]:
```
plt.figure(figsize=(10,5))
sns.scatterplot(x='bmi', y='charges', data=df, hue='smoker')
plt.title("BMI vs Charges (Smoker Highlighted)")
plt.show()
```


BMI vs Charges (Smoker Highlighted)

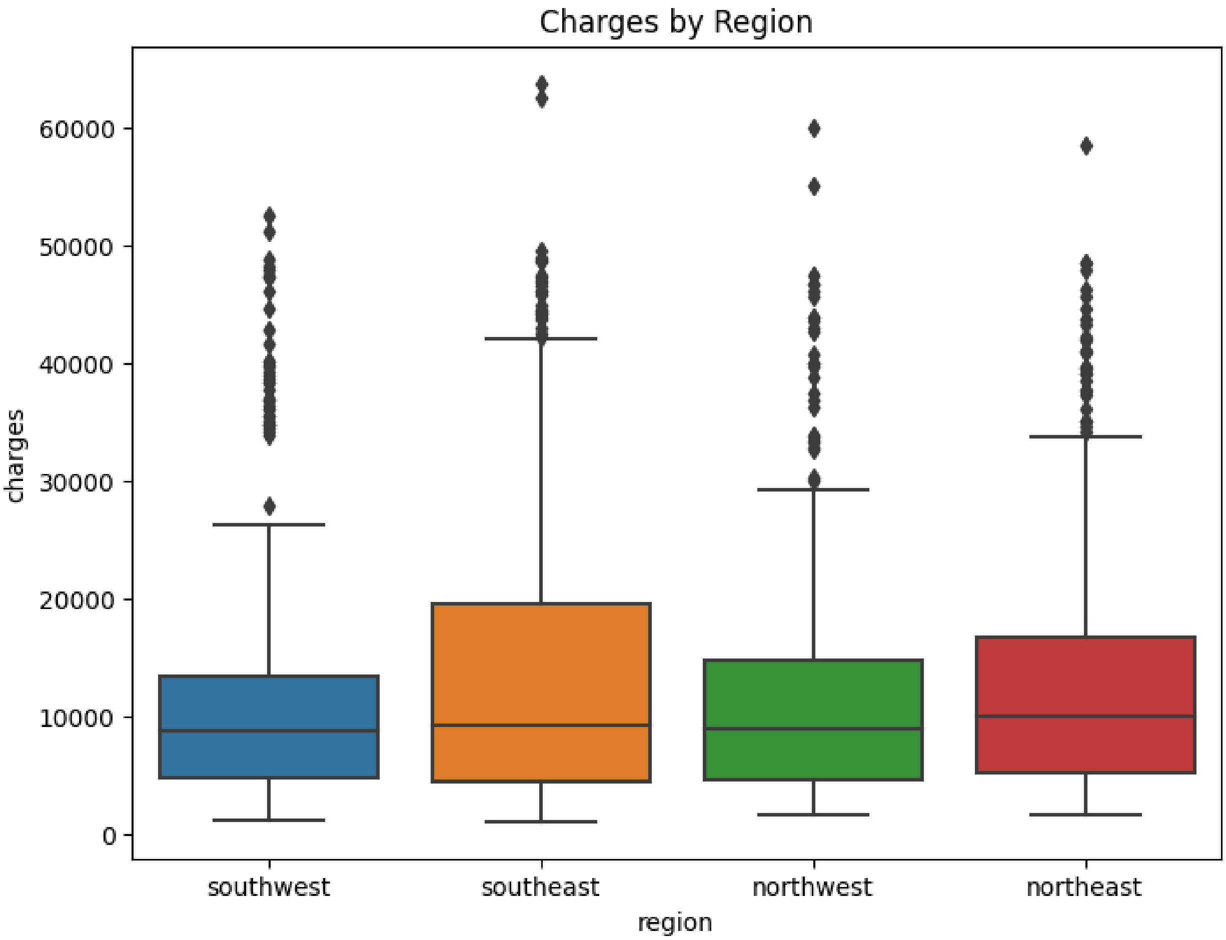
In [27]:
```
plt.figure(figsize=(8,6))
sns.boxplot(x='smoker', y='charges', data=df)
plt.title("Charges by Smoker Status")
plt.show()
```



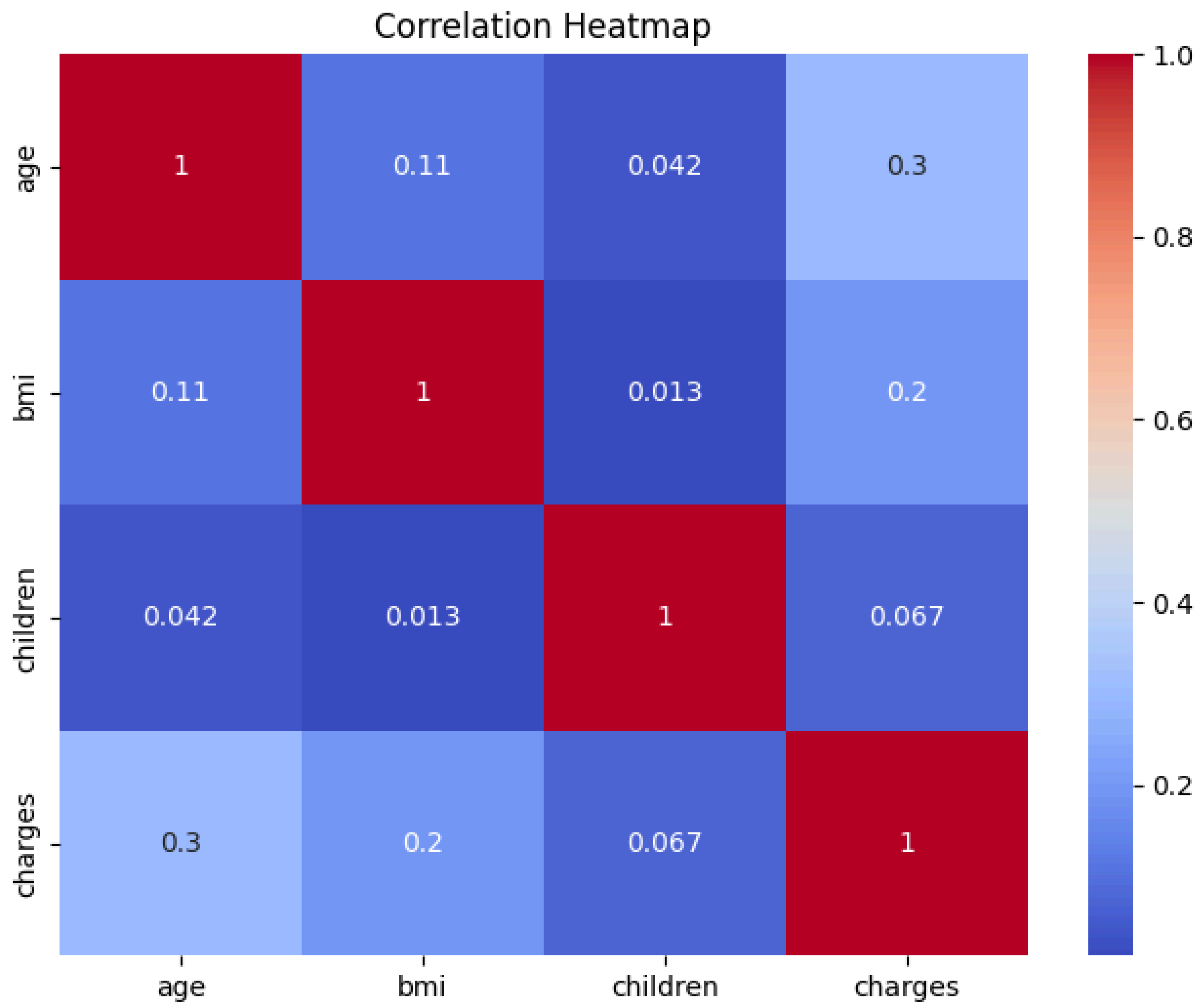
In [28]:
```
plt.figure(figsize=(8,6))
sns.boxplot(x='sex', y='charges', data=df)
plt.title("Charges by Gender")
plt.show()
```



```
In [29]: plt.figure(figsize=(8,6))
         sns.boxplot(x='region', y='charges', data=df)
         plt.title("Charges by Region")
         plt.show()
```



In [30]:
```
plt.figure(figsize=(8,6))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()
```


Correlation Heatmap
age
bmi
children
charges
1
0.11
0.042
0.3
0.11
1
0.013
0.2
0.042
0.013
1
0.067
0.3
0.2
0.067
1
age
bmi
children
charges
1.0
0.8
0.6
0.4
0.2

In [ ]: