In [1]:

```
pip install pandas openpyxl numpy matplotlib seaborn scikit-learn
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pandas in c:\users\acer\appdata\local\packages\pythons
oftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-
packages (2.2.3)
Collecting openpyxl
  Using cached openpyxl-3.1.5-py2.py3-none-any.whl.metadata (2.5 kB)
Requirement already satisfied: numpy in c:\users\acer\appdata\local\packages\pythonso
ftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-p
ackages (2.1.1)
Collecting matplotlib
  Using cached matplotlib-3.9.2-cp312-cp312-win_amd64.whl.metadata (11 kB)
Collecting seaborn
  Using cached seaborn-0.13.2-py3-none-any.whl.metadata (5.4 kB)
Collecting scikit-learn
  Using cached scikit_learn-1.5.2-cp312-cp312-win_amd64.whl.metadata (13 kB)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\acer\appdata\local
\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-package
s\python312\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\acer\appdata\local\packages\p
ythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312
\site-packages (from pandas) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in c:\users\acer\appdata\local\packages
\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python3
12\site-packages (from pandas) (2024.2)
Requirement already satisfied: et-xmlfile in c:\users\acer\appdata\local\packages\pyt
honsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\s
ite-packages (from openpyxl) (1.1.0)
Collecting contourpy>=1.0.1 (from matplotlib)
  Using cached contourpy-1.3.0-cp312-cp312-win_amd64.whl.metadata (5.4 kB)
Requirement already satisfied: cycler>=0.10 in c:\users\acer\appdata\local\packages\p
ythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312
\site-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\acer\appdata\local\packa
ges\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\pyth
on312\site-packages (from matplotlib) (4.54.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\acer\appdata\local\packa
ges\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\pyth
on312\site-packages (from matplotlib) (1.4.7)
Requirement already satisfied: packaging>=20.0 in c:\users\acer\appdata\local\package
s\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python
312\site-packages (from matplotlib) (24.1)
Requirement already satisfied: pillow>=8 in c:\users\acer\appdata\local\packages\pyth
onsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\si
te-packages (from matplotlib) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\acer\appdata\local\packag
es\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\pytho
n312\site-packages (from matplotlib) (3.1.4)
Requirement already satisfied: scipy>=1.6.0 in c:\users\acer\appdata\local\packages\p
ythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312
\site-packages (from scikit-learn) (1.14.1)
Requirement already satisfied: joblib>=1.2.0 in c:\users\acer\appdata\local\packages
\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python3
12\site-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\acer\appdata\local\pa
ckages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\p
ython312\site-packages (from scikit-learn) (3.5.0)
Requirement already satisfied: six>=1.5 in c:\users\acer\appdata\local\packages\pytho
nsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\sit
e-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Using cached openpyxl-3.1.5-py2.py3-none-any.whl (250 kB)
Using cached matplotlib-3.9.2-cp312-cp312-win_amd64.whl (7.8 MB)
Using cached seaborn-0.13.2-py3-none-any.whl (294 kB)
Using cached scikit_learn-1.5.2-cp312-cp312-win_amd64.whl (11.0 MB)
Using cached contourpy-1.3.0-cp312-cp312-win_amd64.whl (218 kB)
Installing collected packages: openpyxl, contourpy, scikit-learn, matplotlib, seaborn
```

```
Successfully installed contourpy-1.3.0 matplotlib-3.9.2 openpyxl-3.1.5 scikit-learn-
1.5.2 seaborn-0.13.2
Note: you may need to restart the kernel to use updated packages.
```

In [2]:

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

```
Matplotlib is building the font cache; this may take a moment.
```

In [16]:

```
import pandas as pd
data = pd.read_csv('heart_health.csv')
data.head()
```

Out[16]:

	HeartDiseaseorAttack	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	Diabetes	PhysActivit
0	0	1	1	1	40	1	0	0	
1	0	0	0	0	25	1	0	0	
2	0	1	1	1	28	0	0	0	
3	0	1	0	1	27	0	0	0	
4	0	1	1	1	24	0	0	0	

5 rows × 22 columns

In [17]:

```
data.shape
```

Out[17]:

```
(253680, 22)
```

In [14]:

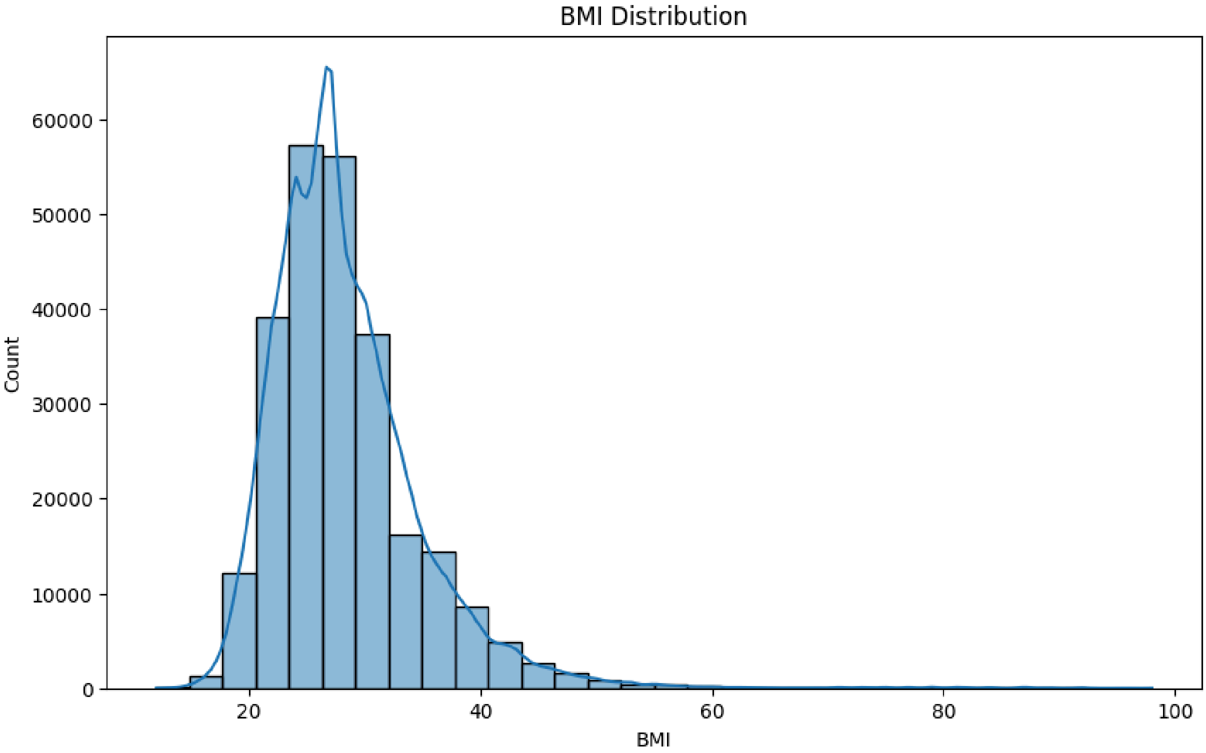
```
categorical_columns = ['HeartDiseaseorAttack', 'HighBP', 'HighChol', 'CholCheck', 'S
                       'Stroke', 'Diabetes', 'PhysActivity', 'Fruits', 'Veggies',
                       'HvyAlcoholConsump', 'AnyHealthcare', 'NoDocbcCost', 'DiffWal
                       'Sex', 'GenHlth', 'Age', 'Education', 'Income']


data[categorical_columns] = data[categorical_columns].astype('category')

print(data.isnull().sum())
```

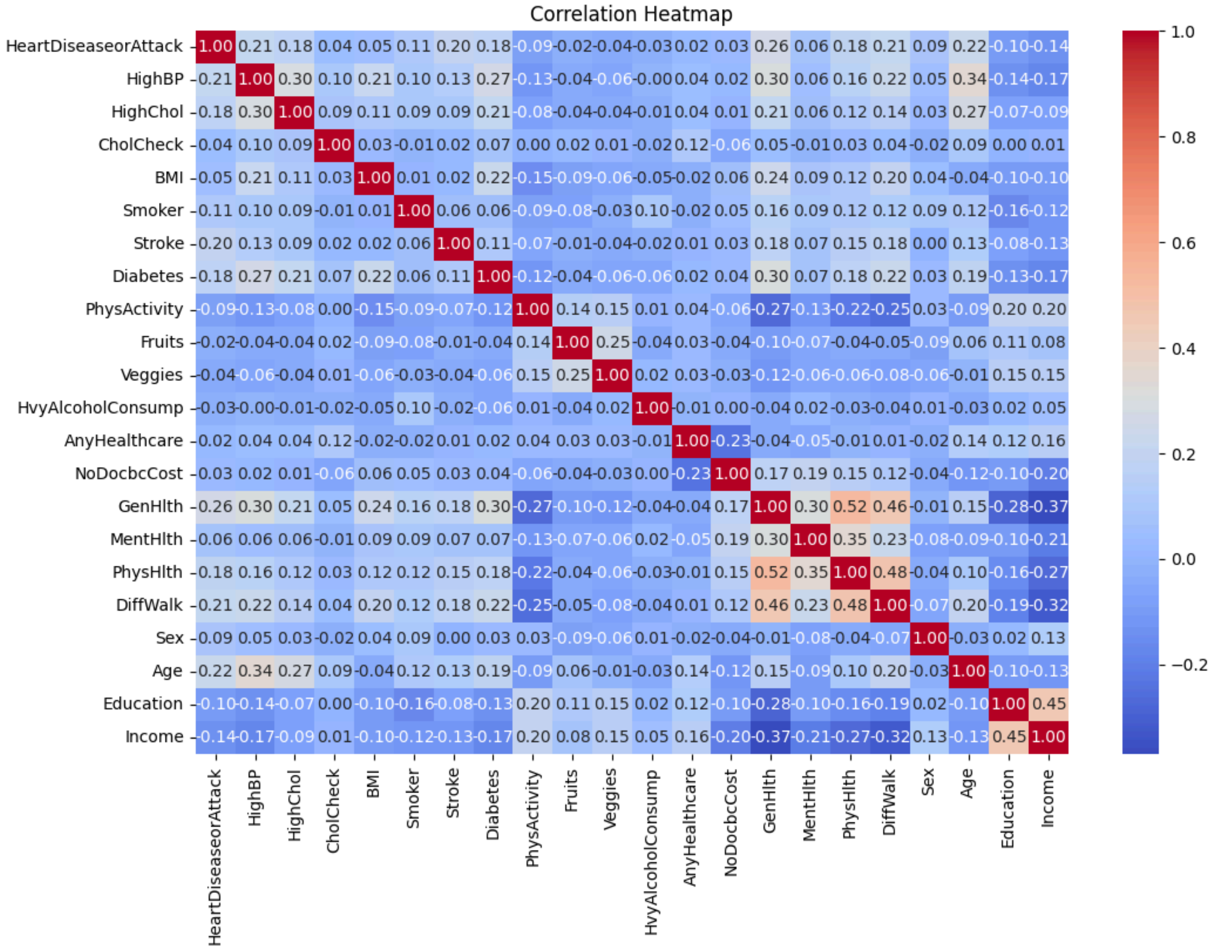
```
HeartDiseaseorAttack    0
HighBP                  0
HighChol                0
CholCheck               0
BMI                     0
Smoker                  0
Stroke                  0
Diabetes                0
PhysActivity            0
Fruits                  0
Veggies                 0
HvyAlcoholConsump       0
AnyHealthcare           0
NoDocbcCost             0
GenHlth                 0
MentHlth                0
PhysHlth                0
```

```
DiffWalk                0
Sex                     0
Age                     0
Education               0
Income                  0
dtype: int64
```

In [18]:

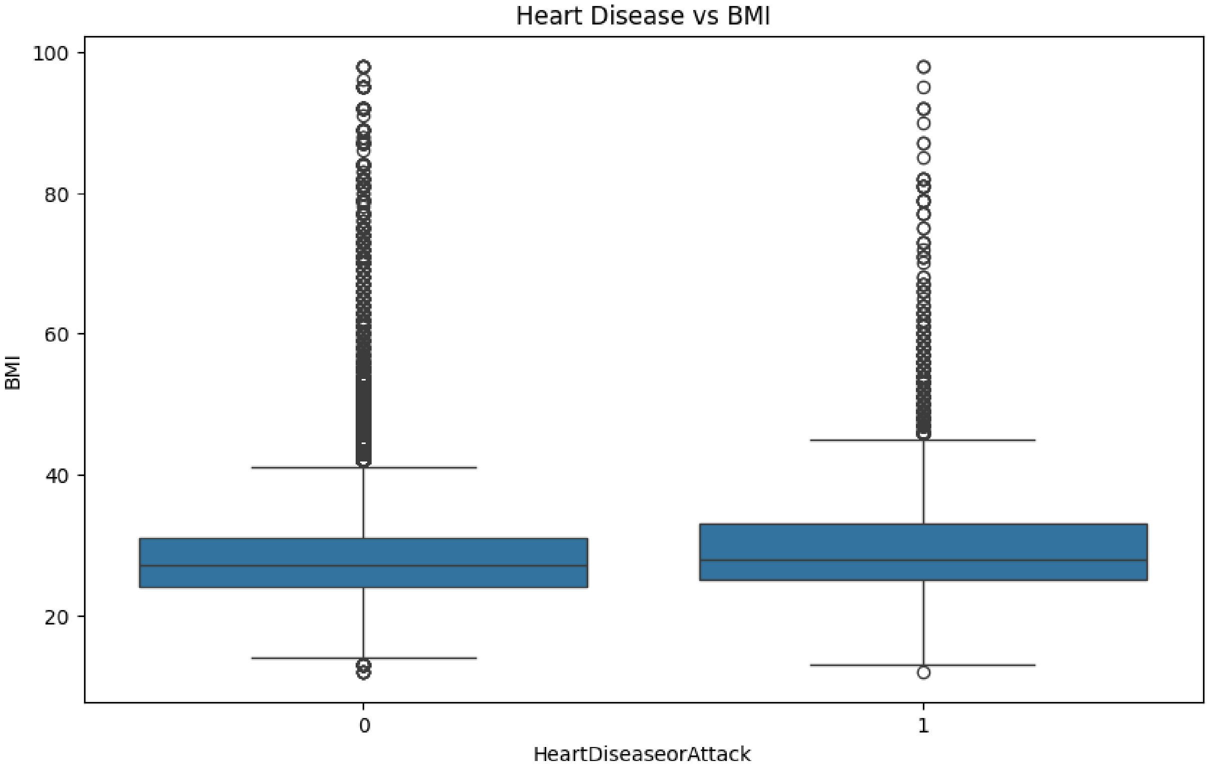
```
data.describe()

plt.figure(figsize=(10, 6))
sns.histplot(data['BMI'], kde=True, bins=30)
plt.title('BMI Distribution')
plt.show()
plt.figure(figsize=(12, 8))
sns.heatmap(data.corr(), annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap')
plt.show()
```

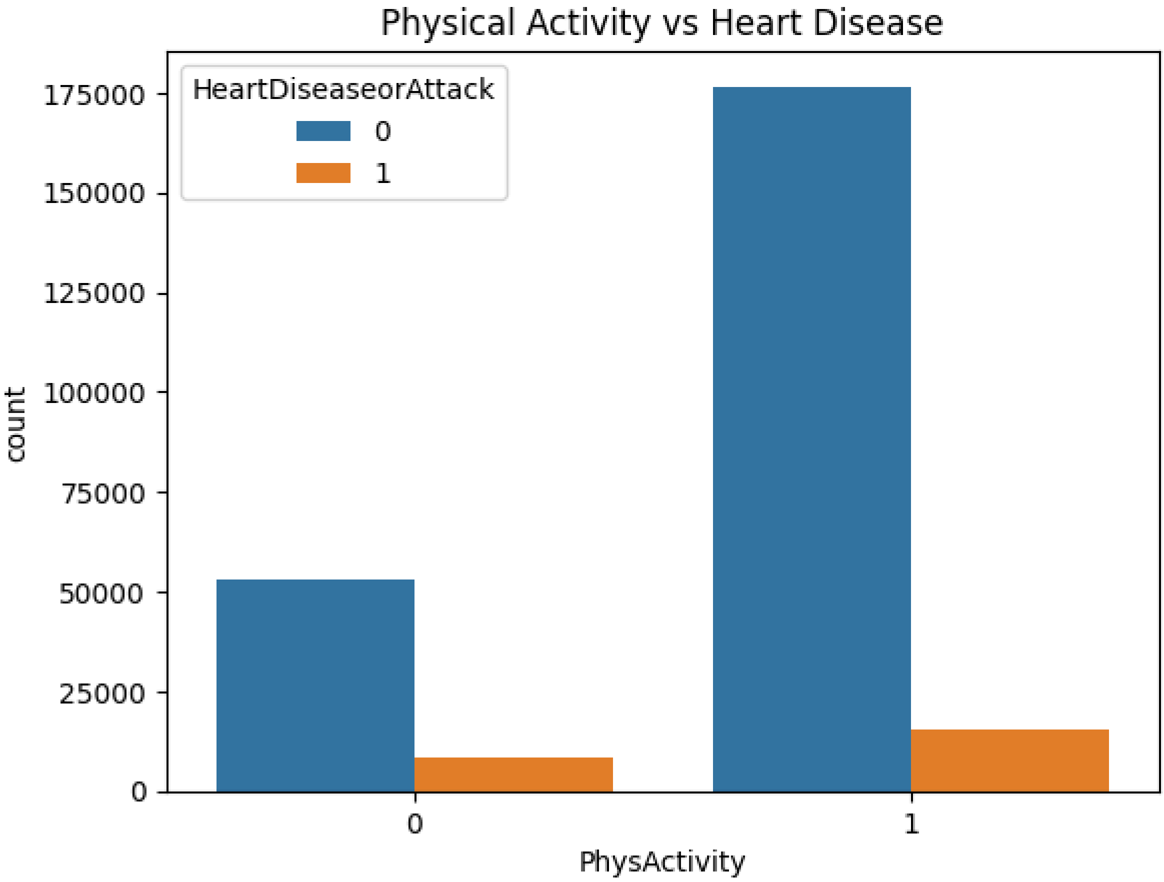




In [19]:

```
plt.figure(figsize=(10, 6))
sns.boxplot(x='HeartDiseaseorAttack', y='BMI', data=data)
plt.title('Heart Disease vs BMI')
plt.show()

sns.countplot(x='PhysActivity', hue='HeartDiseaseorAttack', data=data)
plt.title('Physical Activity vs Heart Disease')
plt.show()
```





In [20]:

```
data['BMI_Category'] = pd.cut(data['BMI'], bins=[0, 18.5, 24.9, 29.9, np.inf],
                              labels=['Underweight', 'Normal weight', 'Overweight',
```

In [21]:

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
```

```
X = data.drop(['HeartDiseaseorAttack'], axis=1)
y = data['HeartDiseaseorAttack']


X = pd.get_dummies(X, drop_first=True)


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_stat


clf = RandomForestClassifier(n_estimators=100, random_state=42)
clf.fit(X_train, y_train)


y_pred = clf.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.9038749605802586
              precision    recall  f1-score   support

           0       0.91      0.99      0.95     45968
           1       0.45      0.11      0.17      4768

    accuracy                           0.90     50736
   macro avg       0.68      0.55      0.56     50736
weighted avg       0.87      0.90      0.88     50736
```

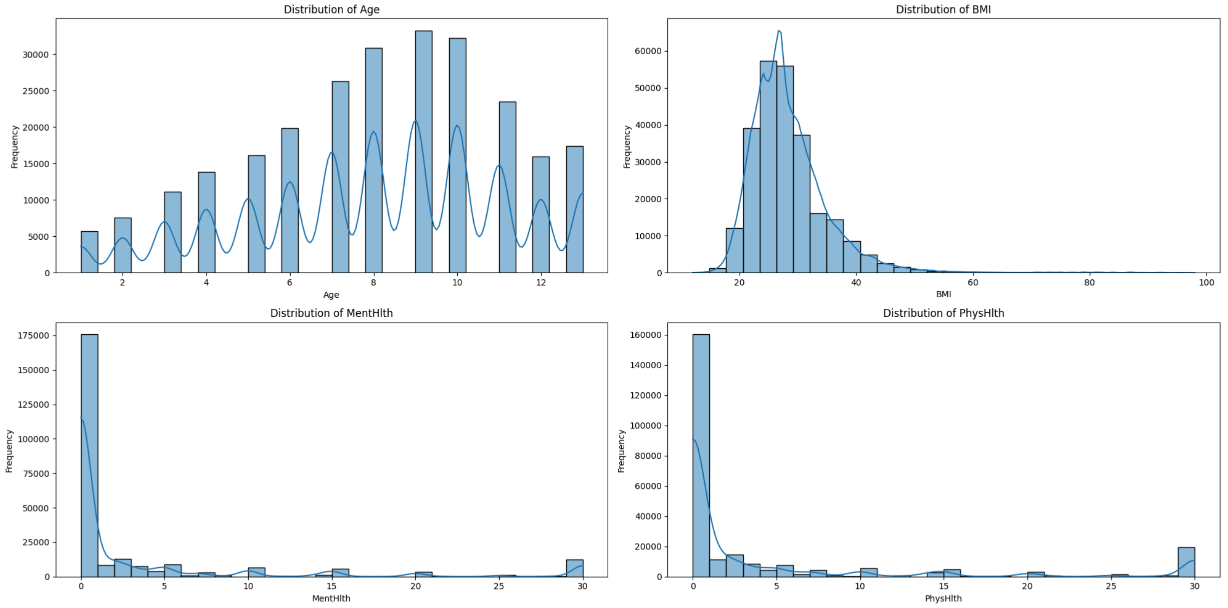
In [25]:

```
# Save cleaned data
data.to_csv('cleaned_heart_disease_data.csv', index=False)
```

UNIVARIATE ANALYSIS

In [29]:

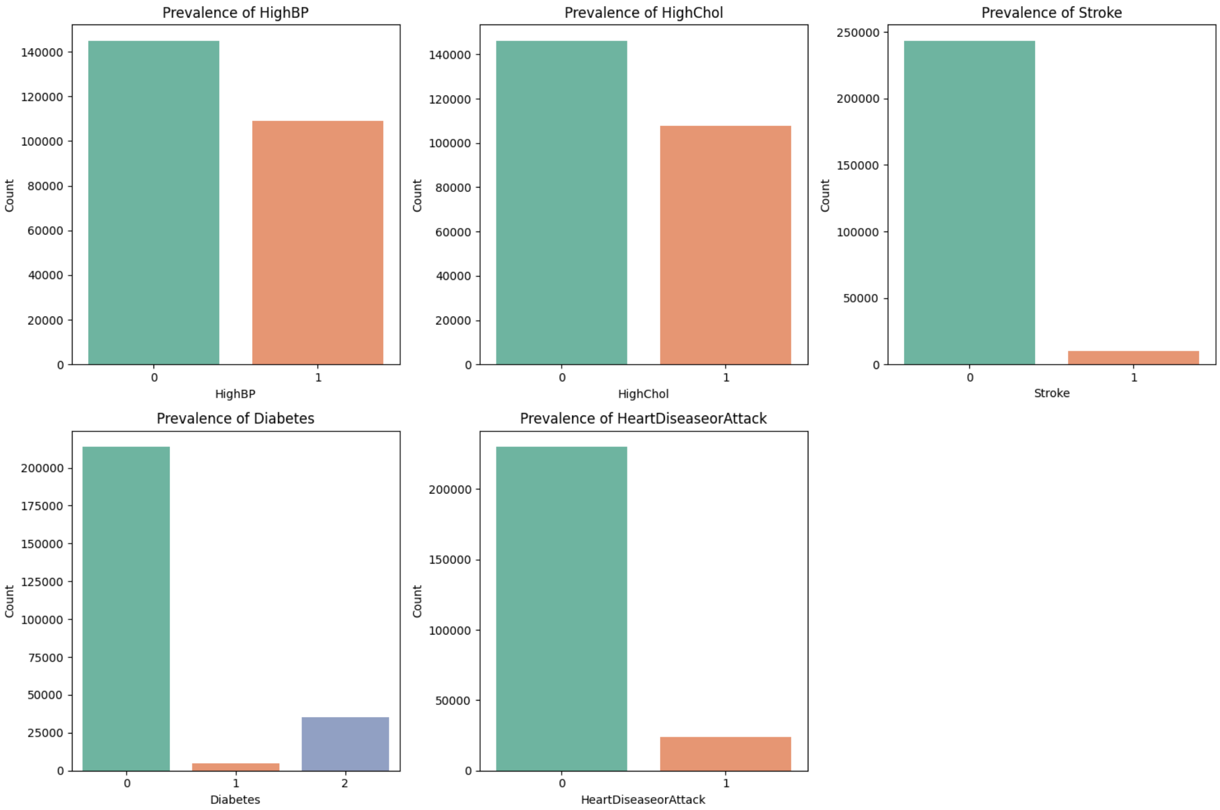
```
import matplotlib.pyplot as plt
import seaborn as sns

variables = ['Age', 'BMI', 'MentHlth', 'PhysHlth']

plt.figure(figsize=(20, 10))
for i, var in enumerate(variables):
    plt.subplot(2, 2, i + 1)
    sns.histplot(data[var], bins=30, kde=True)
    plt.title(f'Distribution of {var}')
    plt.xlabel(var)
    plt.ylabel('Frequency')

plt.tight_layout()
plt.show()
```



In [30]:

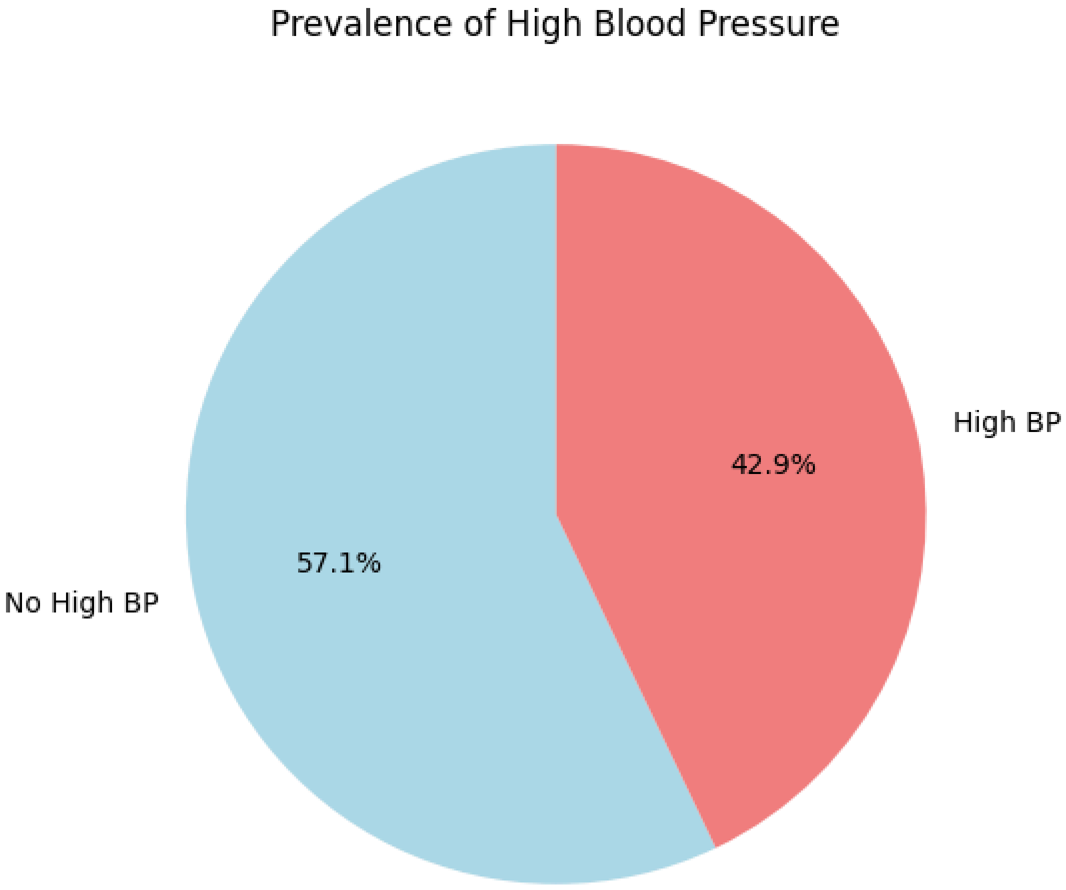
```
conditions = ['HighBP', 'HighChol', 'Stroke', 'Diabetes', 'HeartDiseaseorAttack']

plt.figure(figsize=(15, 10))
for i, cond in enumerate(conditions):
    plt.subplot(2, 3, i + 1)
    sns.countplot(x=data[cond], palette='Set2')
    plt.title(f'Prevalence of {cond}')
    plt.xlabel(cond)
    plt.ylabel('Count')

plt.tight_layout()
plt.show()
```

```
C:\Users\ACER\AppData\Local\Temp\ipykernel_24516\4116980035.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.
0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.countplot(x=data[cond], palette='Set2')
C:\Users\ACER\AppData\Local\Temp\ipykernel_24516\4116980035.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.
0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.countplot(x=data[cond], palette='Set2')
C:\Users\ACER\AppData\Local\Temp\ipykernel_24516\4116980035.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.
0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.countplot(x=data[cond], palette='Set2')
C:\Users\ACER\AppData\Local\Temp\ipykernel_24516\4116980035.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.
0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.countplot(x=data[cond], palette='Set2')
C:\Users\ACER\AppData\Local\Temp\ipykernel_24516\4116980035.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.
0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.countplot(x=data[cond], palette='Set2')
```



In [31]:

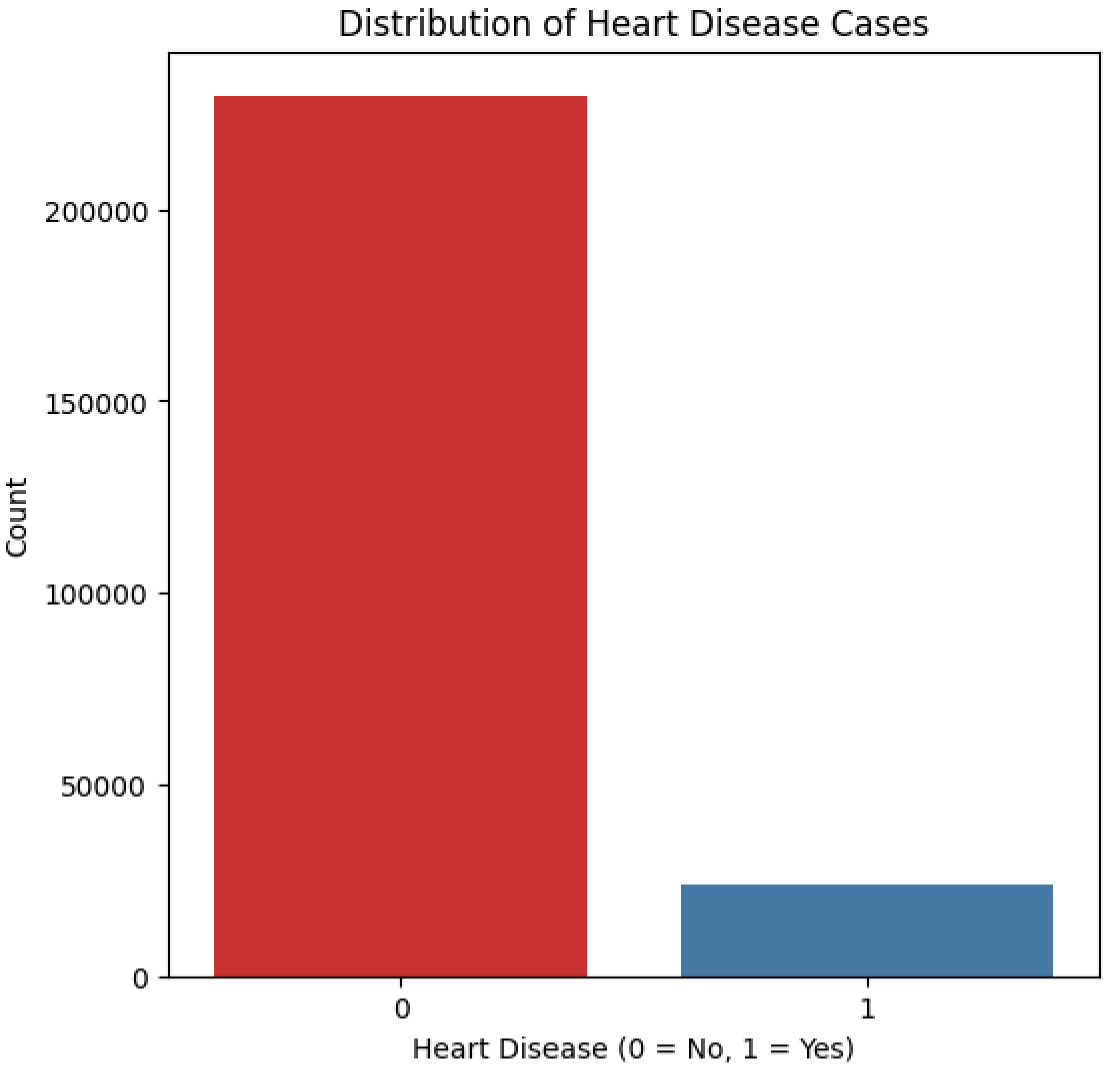
```
high_bp_counts = data['HighBP'].value_counts()
labels = ['No High BP', 'High BP']

plt.figure(figsize=(6, 6))
plt.pie(high_bp_counts, labels=labels, autopct='%1.1f%%', startangle=90, colors=['li
plt.title('Prevalence of High Blood Pressure')
plt.show()
```



In [32]:

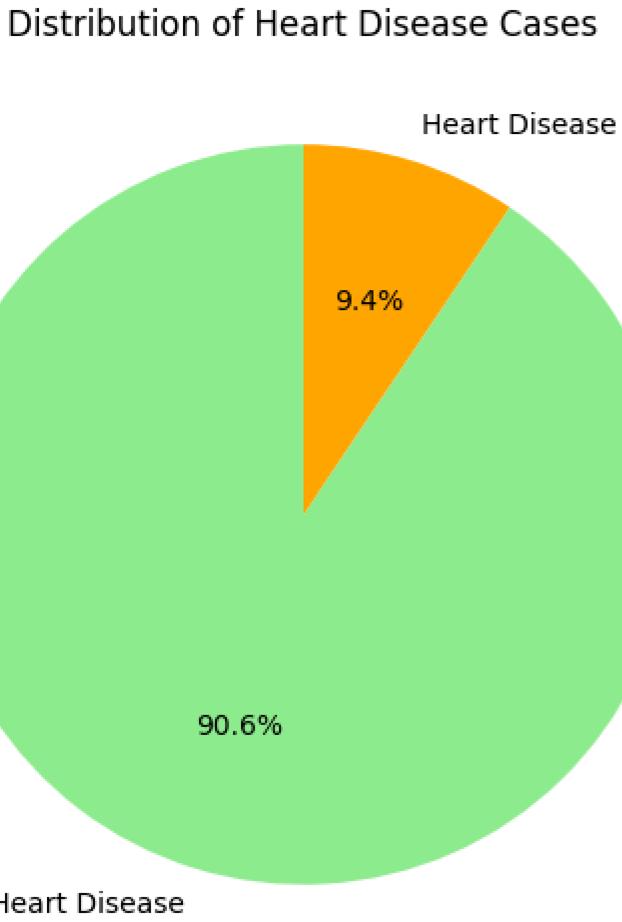
```
plt.figure(figsize=(6, 6))
sns.countplot(x=data['HeartDiseaseorAttack'], palette='Set1')
plt.title('Distribution of Heart Disease Cases')
plt.xlabel('Heart Disease (0 = No, 1 = Yes)')
plt.ylabel('Count')
plt.show()
```

```
C:\Users\ACER\AppData\Local\Temp\ipykernel_24516\3226888871.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.
0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.countplot(x=data['HeartDiseaseorAttack'], palette='Set1')
```



In [33]:

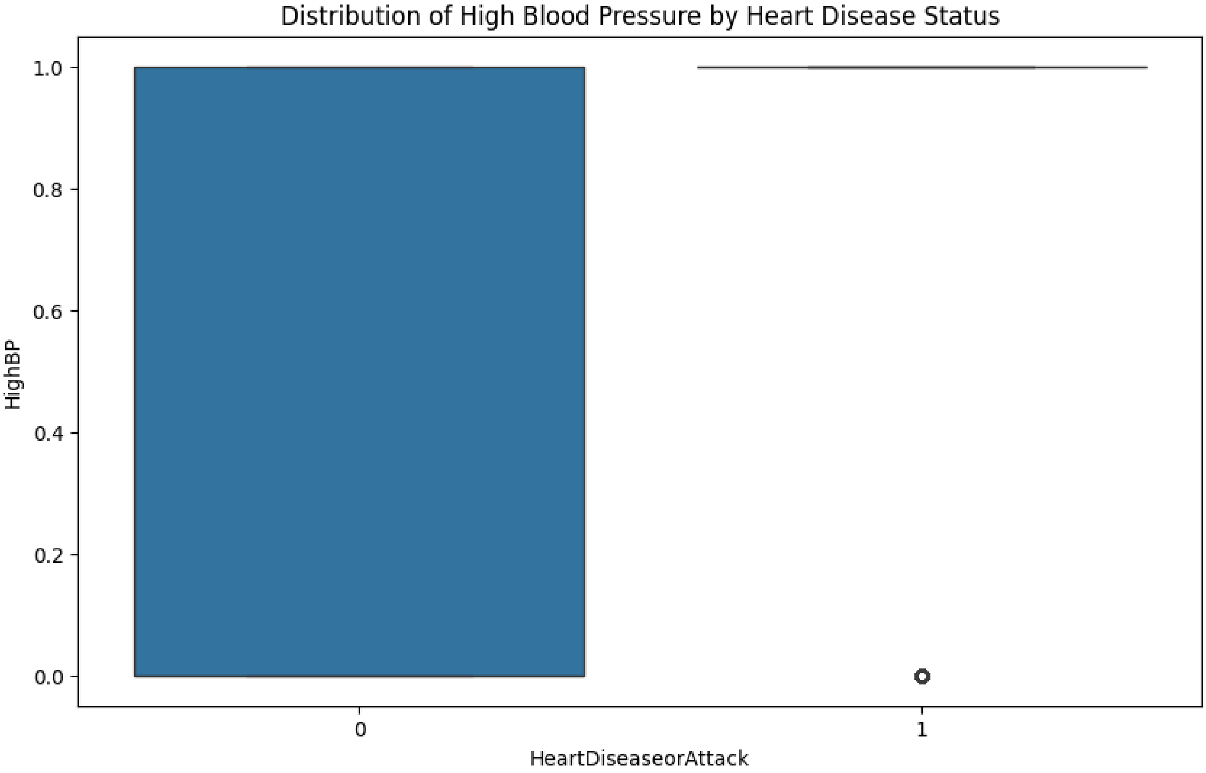
```
heart_disease_counts = data['HeartDiseaseorAttack'].value_counts()
labels = ['No Heart Disease', 'Heart Disease']

plt.figure(figsize=(6, 6))
plt.pie(heart_disease_counts, labels=labels, autopct='%1.1f%%', startangle=90, color
plt.title('Distribution of Heart Disease Cases')
plt.show()
```

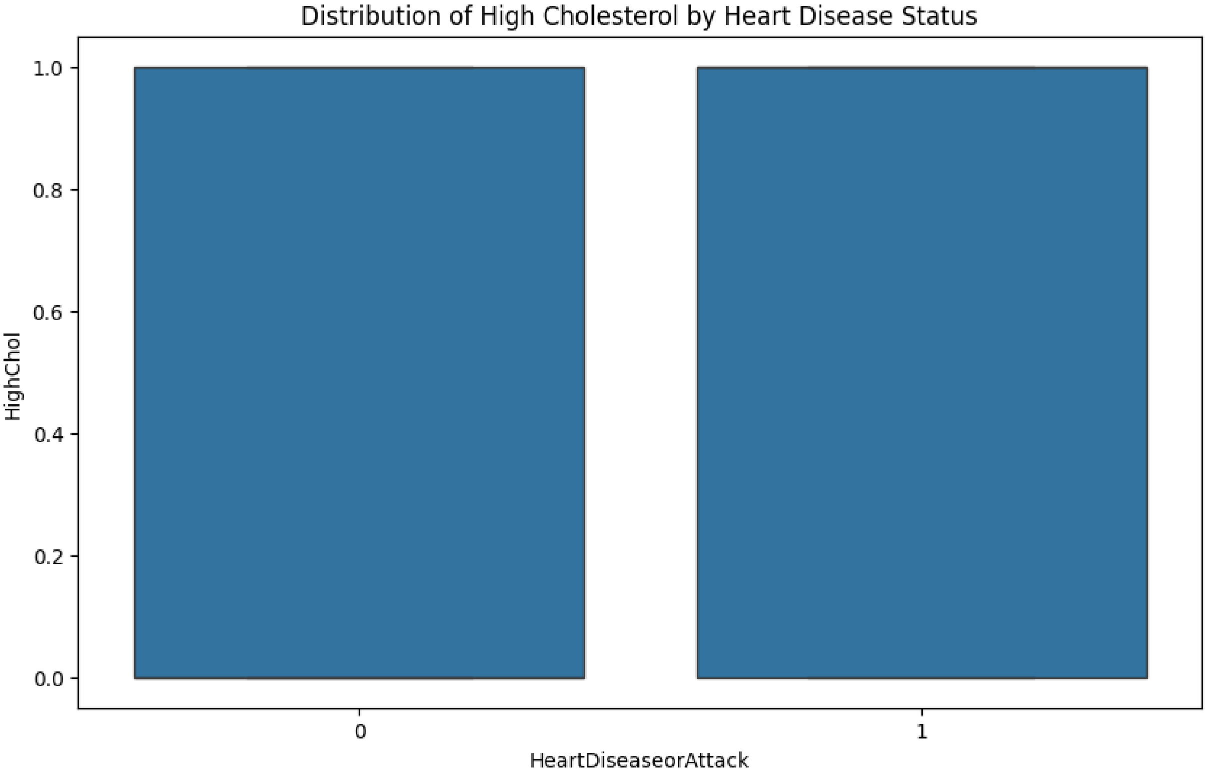


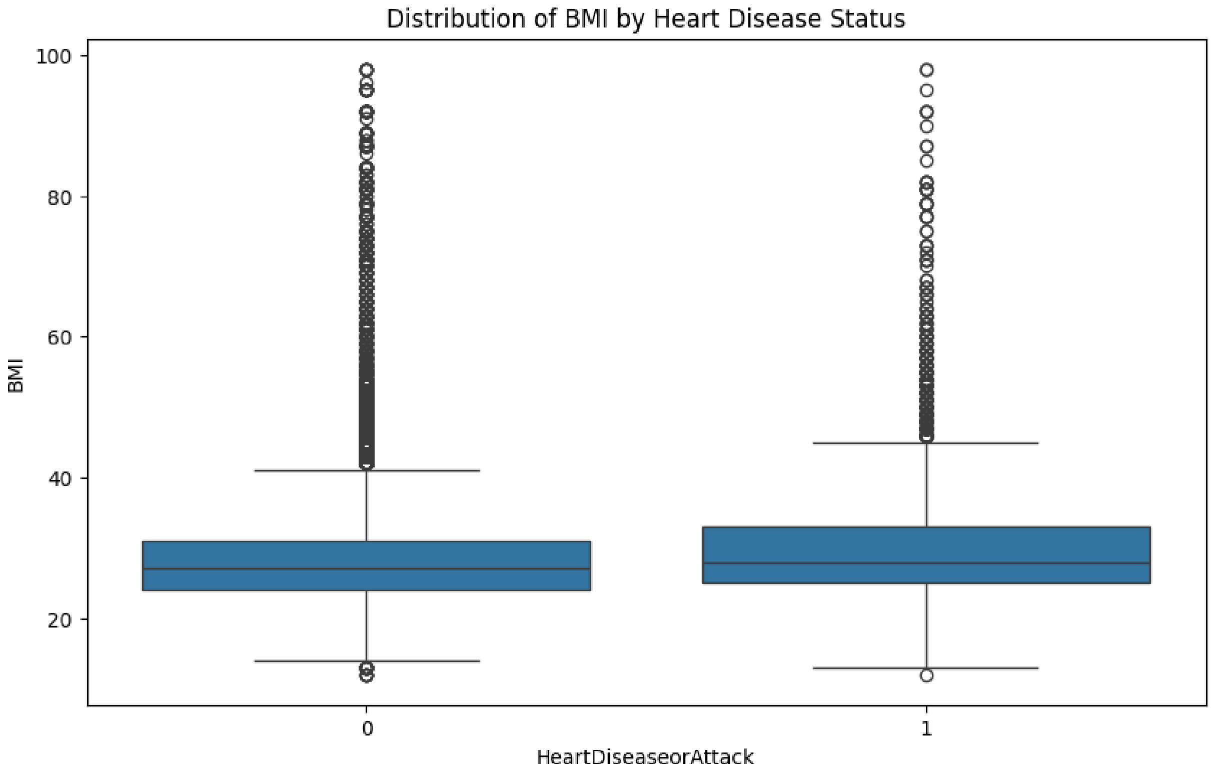
BIVARIATE ANALYSIS

In [40]:

```
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6))
sns.boxplot(x='HeartDiseaseorAttack', y='HighBP', data=data)
plt.title('Distribution of High Blood Pressure by Heart Disease Status')
plt.show()


plt.figure(figsize=(10, 6))
sns.boxplot(x='HeartDiseaseorAttack', y='HighChol',data=data)
plt.title('Distribution of High Cholesterol by Heart Disease Status')
plt.show()


plt.figure(figsize=(10, 6))
sns.boxplot(x='HeartDiseaseorAttack', y='BMI',data=data)
plt.title('Distribution of BMI by Heart Disease Status')
plt.show()
```

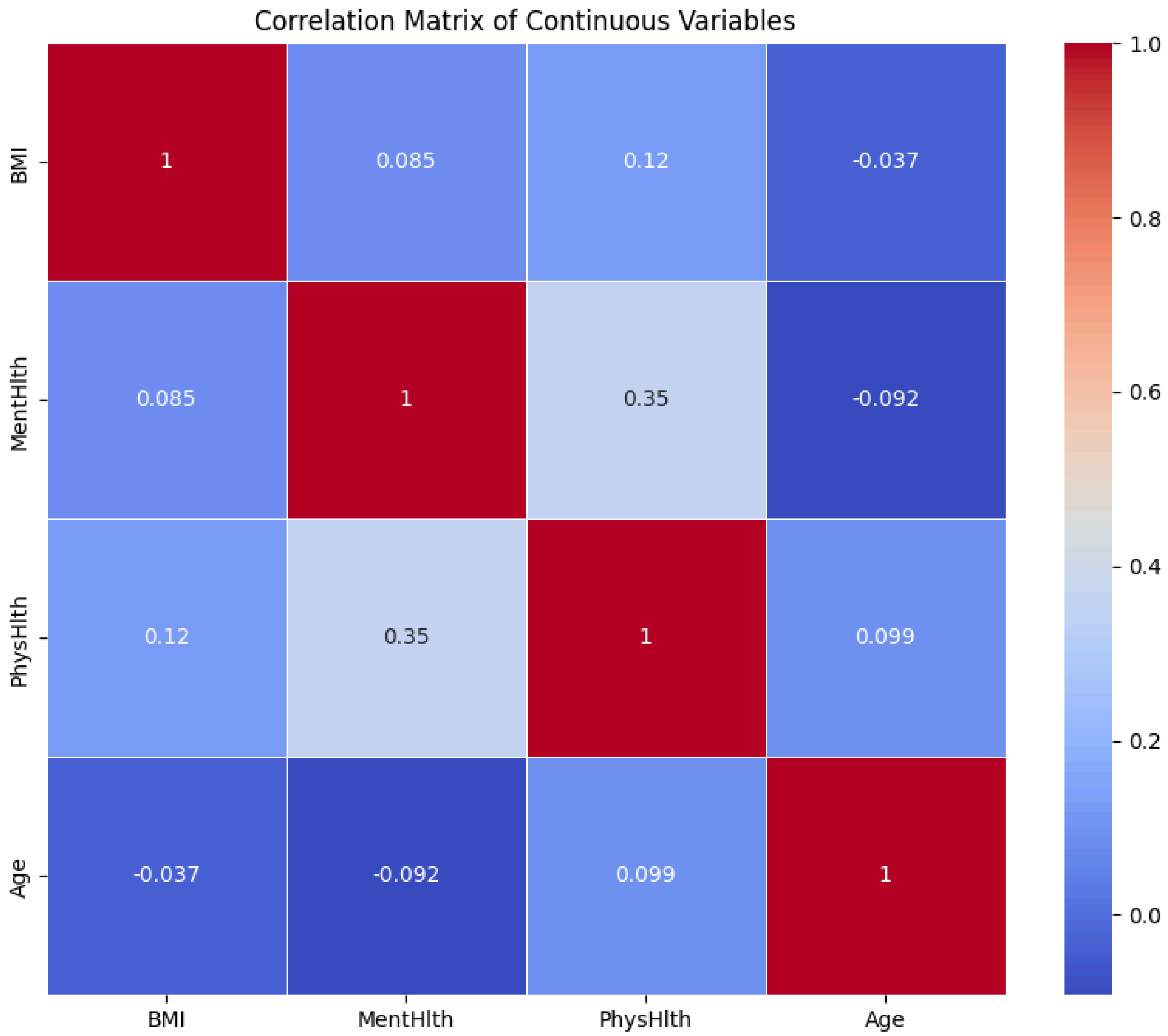

Distribution of High Blood Pressure by Heart Disease Status
HighBP
1.0
0.8
0.6
0.4
0.2
0.0
0
1
HeartDiseaseorAttack


Distribution of High Cholesterol by Heart Disease Status
HighChol
1.0
0.8
0.6
0.4
0.2
0.0
0
1
HeartDiseaseorAttack



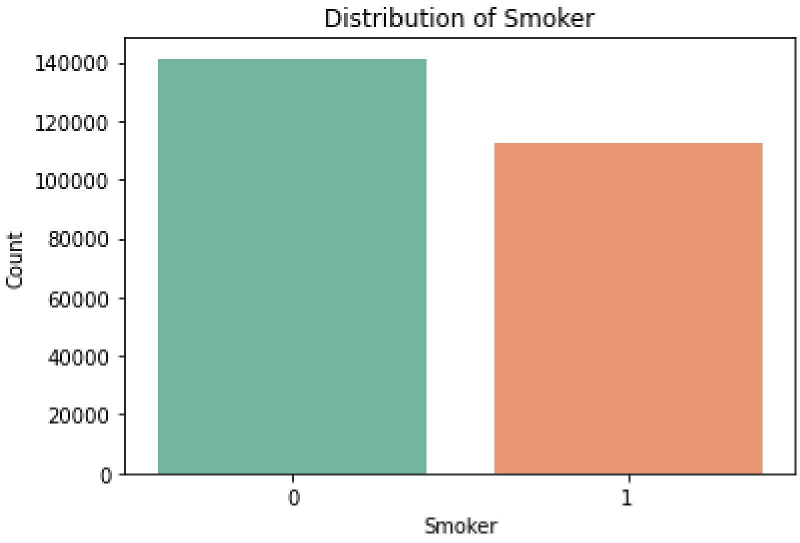
In [42]:

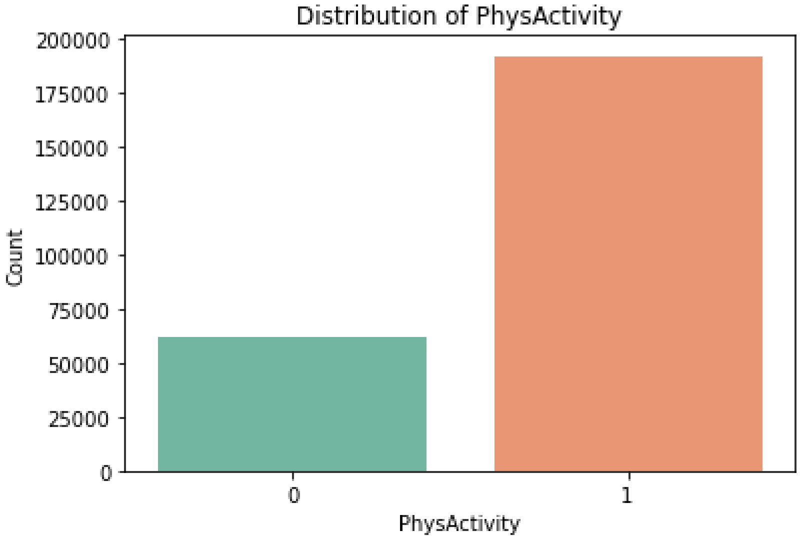
```
corr_matrix = data[['BMI', 'MentHlth', 'PhysHlth', 'Age']].corr()
plt.figure(figsize=(10, 8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Matrix of Continuous Variables')
plt.show()
```

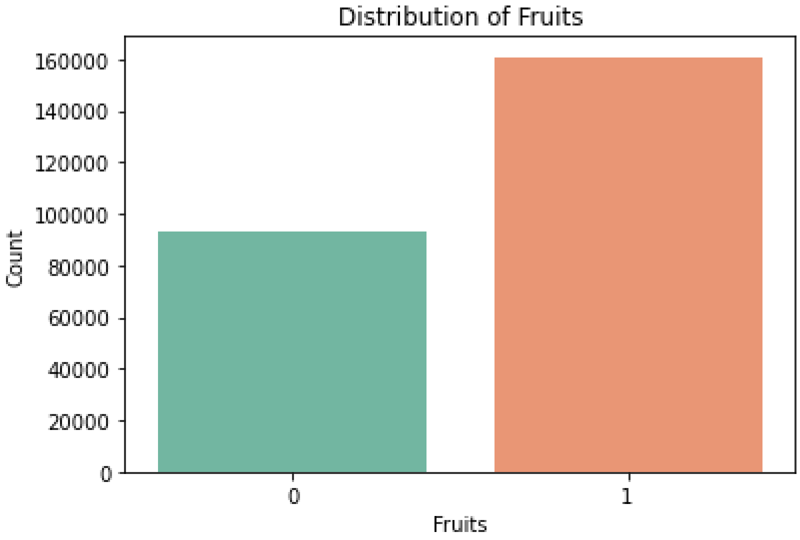


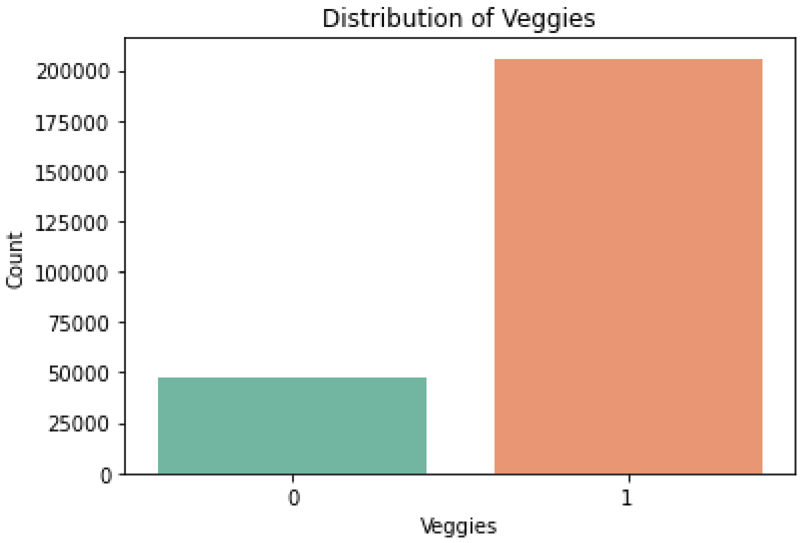
In [14]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
data = pd.read_csv('heart_health.csv')

categorical_vars = ['Smoker', 'PhysActivity', 'Fruits', 'Veggies']


for var in categorical_vars:
    plt.figure(figsize=(6, 4))
    sns.countplot(x=var, data=data, palette="Set2")
    plt.title(f'Distribution of {var}')
    plt.ylabel('Count')
    plt.xlabel(var)
    plt.show()
```


Distribution of Smoker

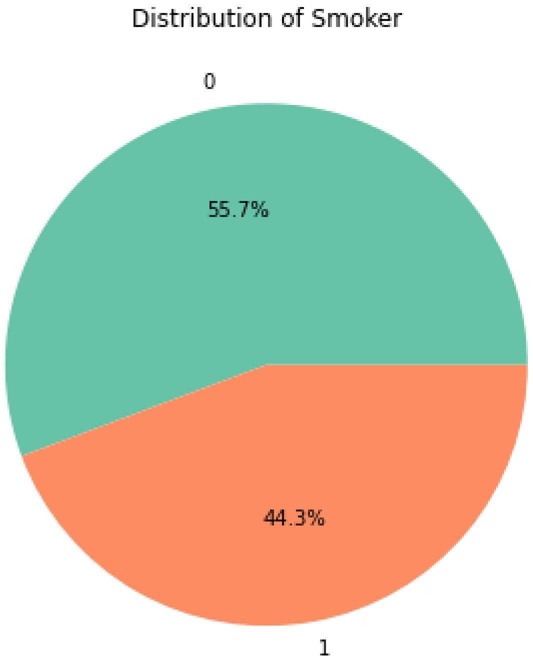

Distribution of PhysActivity

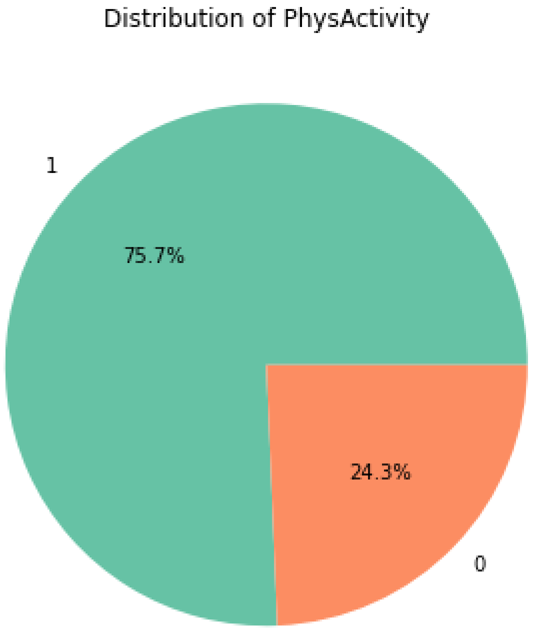

Distribution of Fruits

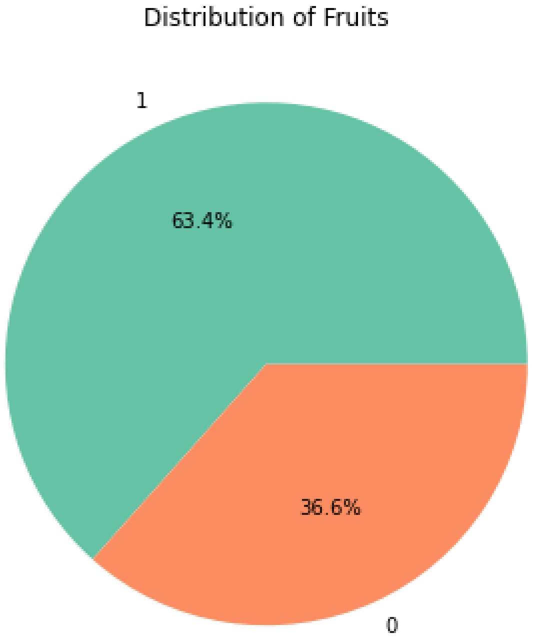


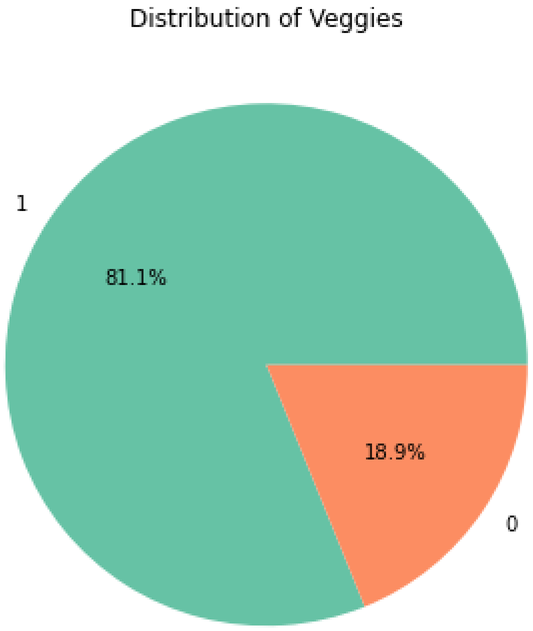
In [16]:

```
for var in categorical_vars:
    plt.figure(figsize=(6, 6))
    data[var].value_counts().plot.pie(autopct='%1.1f%%', colors=sns.color_palette("S
    plt.title(f'Distribution of {var}')
    plt.ylabel('')
    plt.show()
```



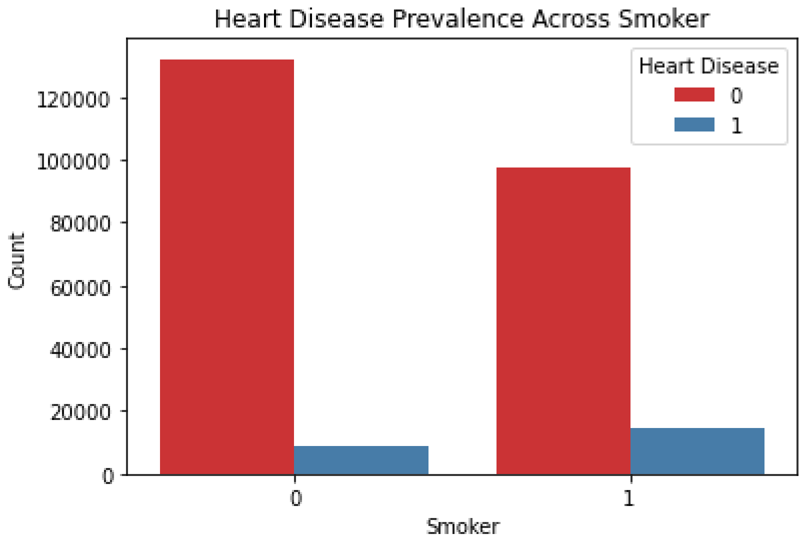

Distribution of PhysActivity
1
75.7%
24.3%
0

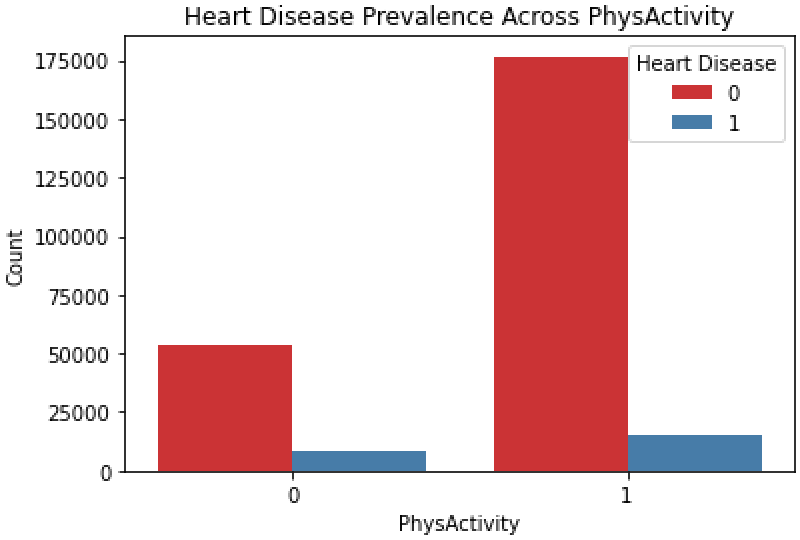

Distribution of Fruits
1
63.4%
36.6%
0

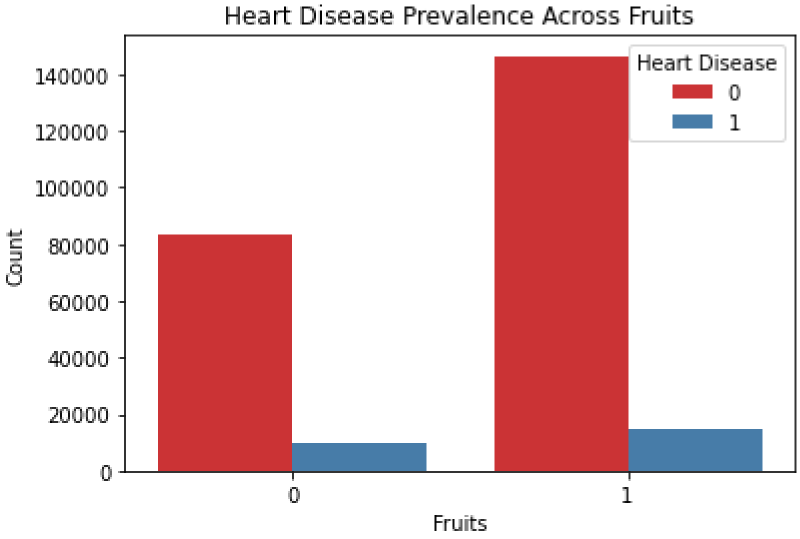


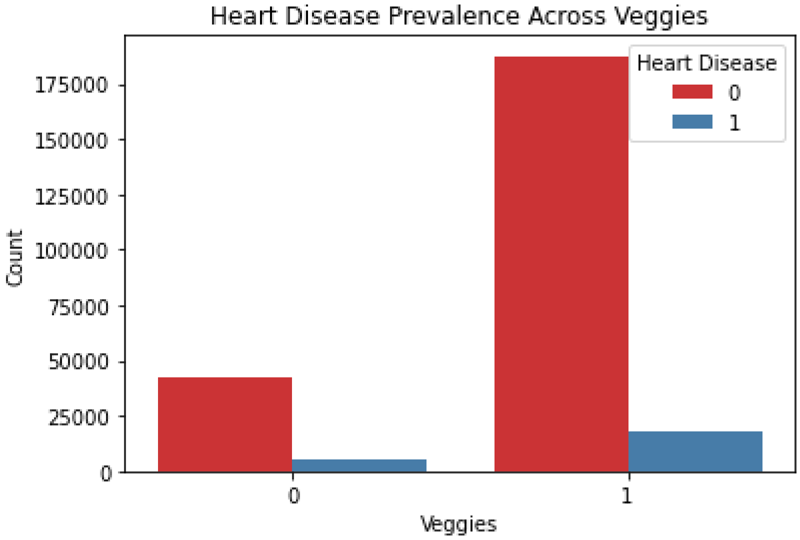
In [20]:

```
for var in categorical_vars:
    plt.figure(figsize=(6, 4))
    sns.countplot(x=var, hue=def stacked_bar_chart(df, var, target):
    crosstab = pd.crosstab(df[var], df[target], normalize='index')
    crosstab.plot(kind='bar', stacked=True, color=['#FF9999', '#66B2FF'], figsize=(6
    plt.title(f'Heart Disease Prevalence Across {var}')
    plt.ylabel('Proportion')
    plt.xlabel(var)
    plt.legend(title=target)
    plt.show()
```





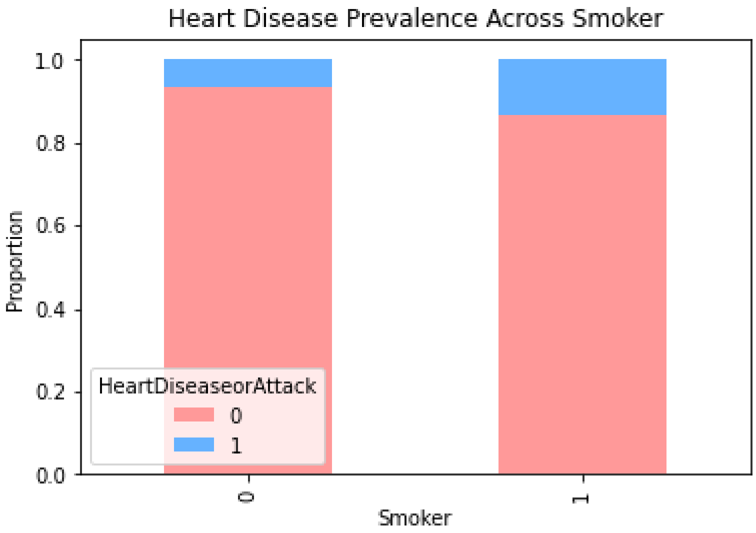


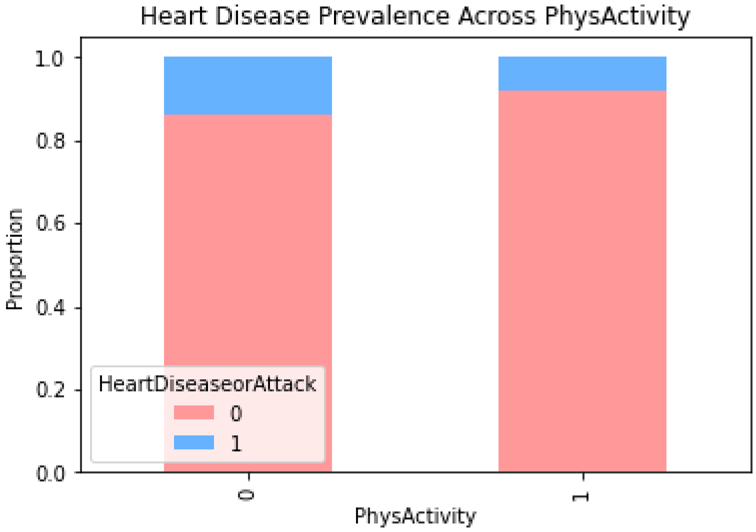


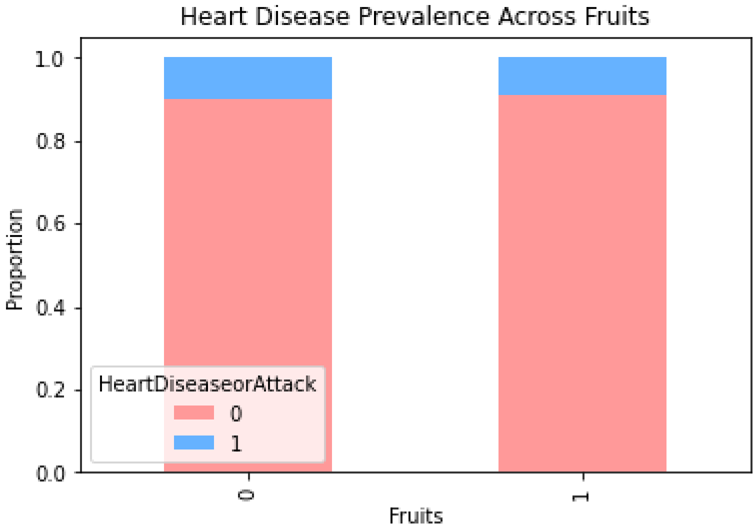
In [28]:

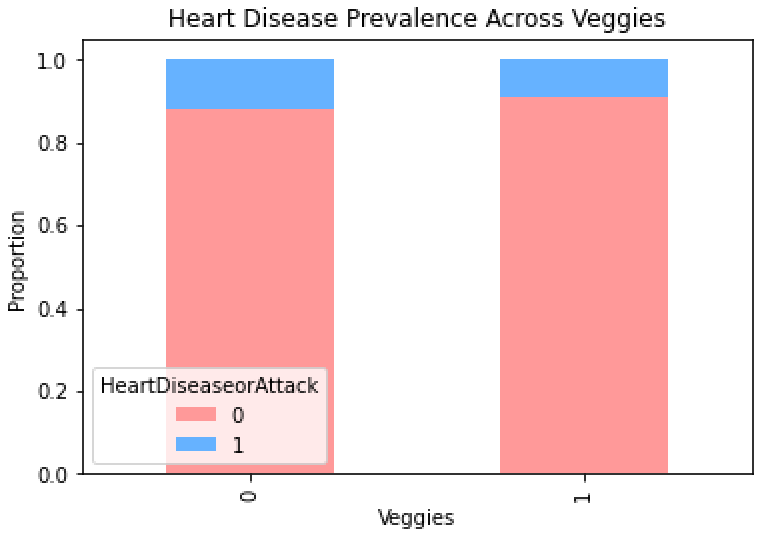
```
def stacked_bar_chart(data, var, target):
    crosstab = pd.crosstab(data[var], data[target], normalize='index')
    crosstab.plot(kind='bar', stacked=True, color=['#FF9999', '#66B2FF'], figsize=(6
    plt.title(f'Heart Disease Prevalence Across {var}')
    plt.ylabel('Proportion')
    plt.xlabel(var)
    plt.legend(title=target)
    plt.show()

# Generate stacked bar charts for categorical variables
```

```
for var in categorical_vars:
 stacked_bar_chart(data, var, 'HeartDiseaseorAttack')
```



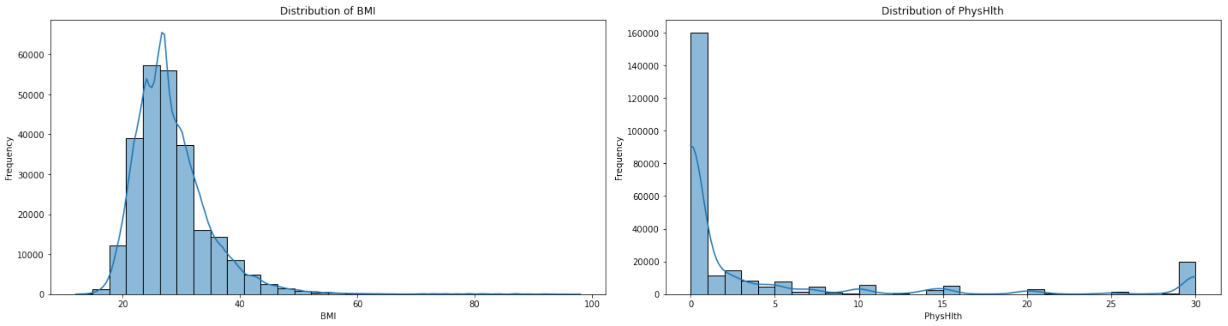






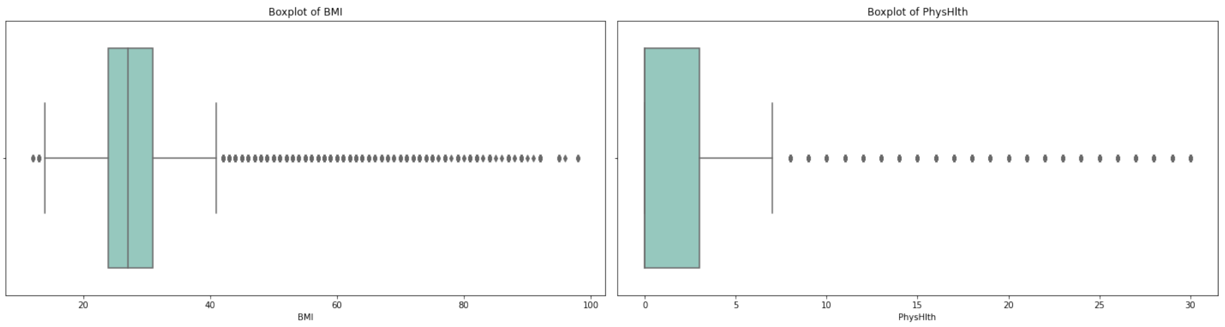
In [31]:

```
continuous_vars = ['BMI', 'PhysHlth']

plt.figure(figsize=(20, 10))
for i, var in enumerate(continuous_vars):
    plt.subplot(2, 2, i + 1)
    sns.histplot(data[var], bins=30, kde=True)
    plt.title(f'Distribution of {var}')
    plt.xlabel(var)
    plt.ylabel('Frequency')

plt.tight_layout()
plt.show()
```



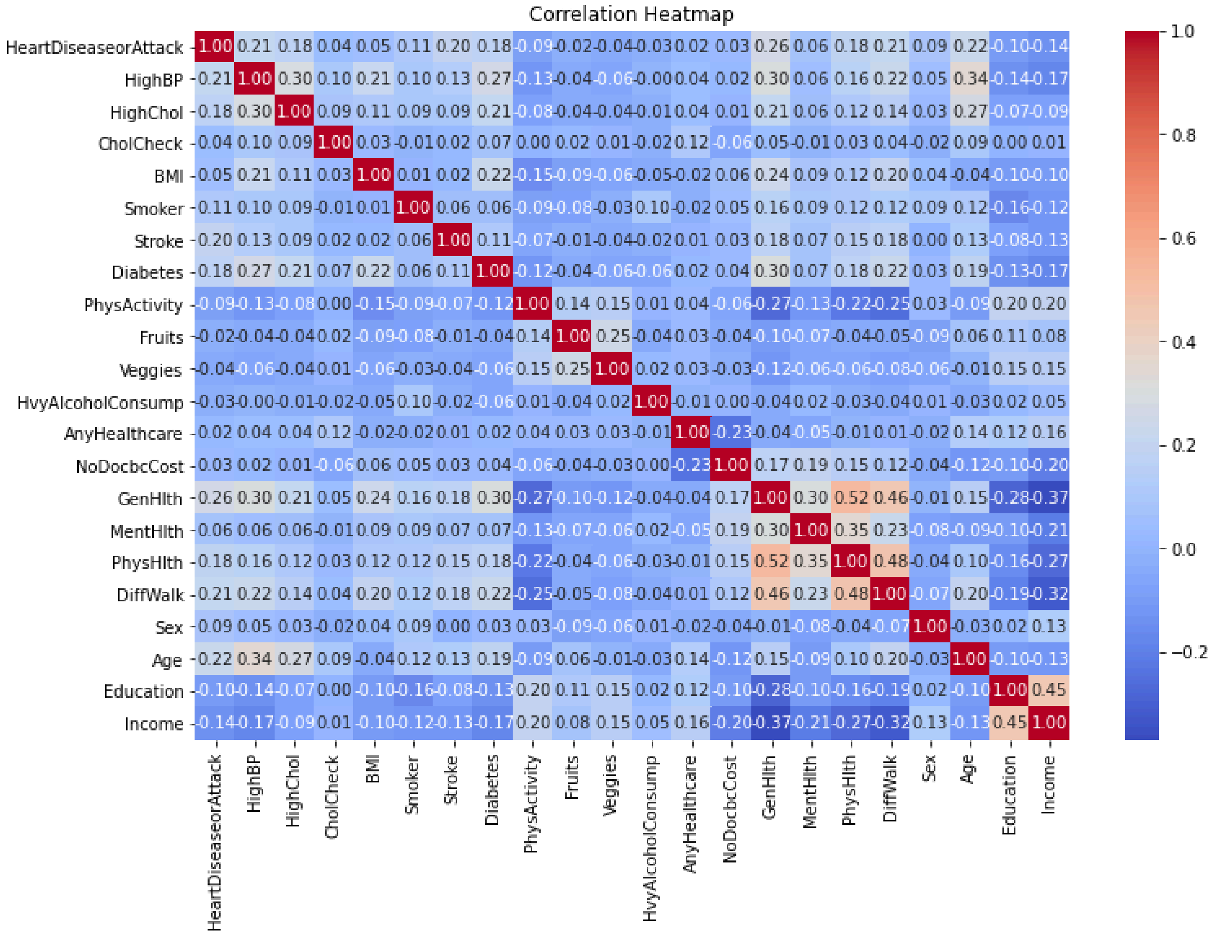
In [32]:

```
plt.figure(figsize=(20, 10))
for i, var in enumerate(continuous_vars):
    plt.subplot(2, 2, i + 1)
    sns.boxplot(x=data[var], palette="Set3")
    plt.title(f'Boxplot of {var}')
    plt.xlabel(var)

plt.tight_layout()
plt.show()
```



In [33]:

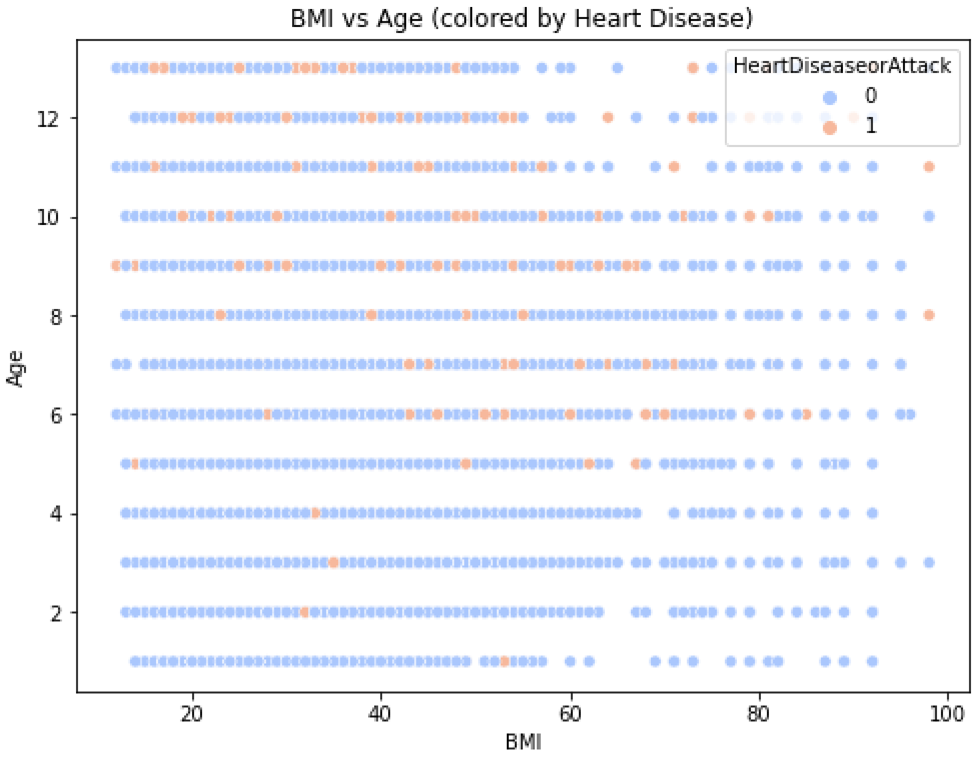
```
correlation_matrix = data.corr()

# Plotting heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap')
plt.show()
```



In [36]:

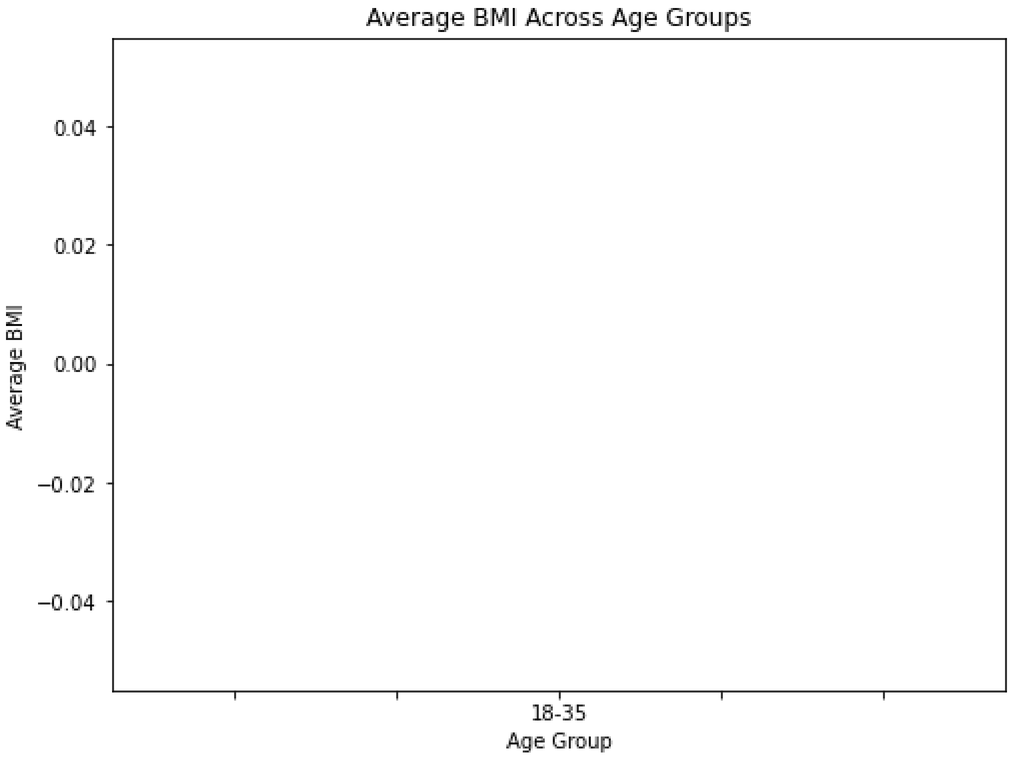
```
plt.figure(figsize=(8, 6))
sns.scatterplot(x='BMI', y='Age', hue='HeartDiseaseorAttack', data=data, palette='co
plt.title('BMI vs Age (colored by Heart Disease)')
plt.xlabel('BMI')
plt.ylabel('Age')
plt.show()
```

```
C:\Users\ACER\anaconda3\lib\site-packages\IPython\core\pylabtools.py:132: UserWarnin
g: Creating legend with loc="best" can be slow with large amounts of data.
  fig.canvas.print_figure(bytes_io, **kw)
```



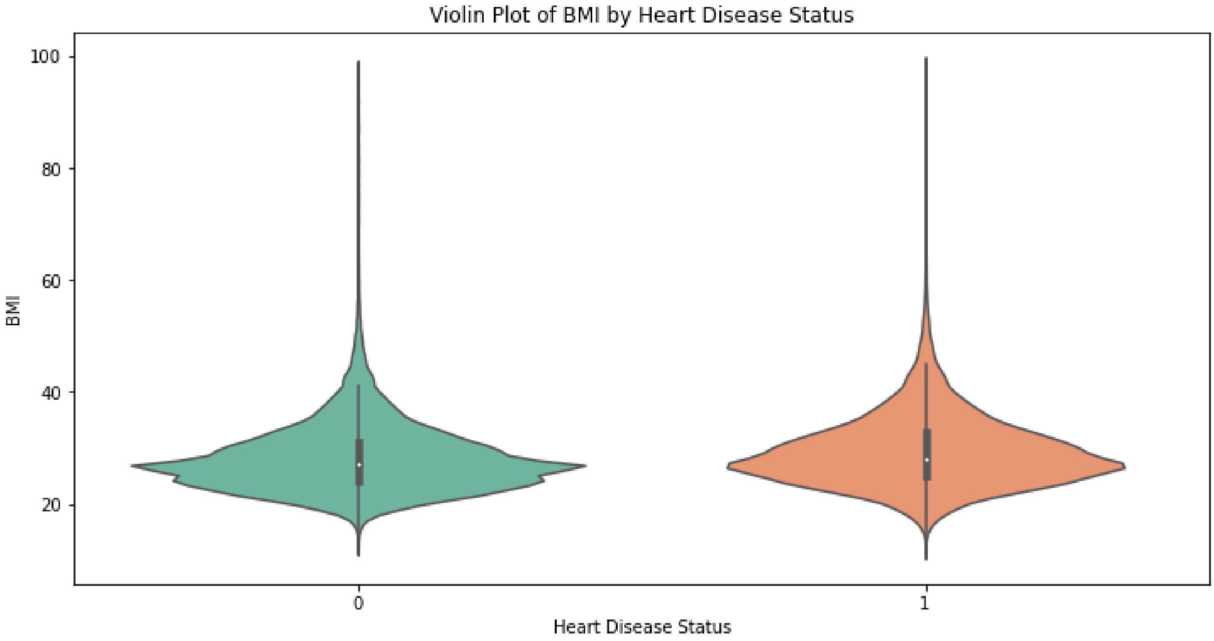
In [38]:

```
age_groups = pd.cut(data['Age'], bins=[18, 35, 50, 65, 80, 100], labels=['18-35', '3
mean_bmi_by_age_group = data.groupby(age_groups)['BMI'].mean()

plt.figure(figsize=(8, 6))
mean_bmi_by_age_group.plot(kind='line', marker='o', color='purple')
plt.title('Average BMI Across Age Groups')
plt.xlabel('Age Group')
plt.ylabel('Average BMI')
plt.show()
```



In [39]:

```
plt.figure(figsize=(12, 6))
sns.violinplot(x='HeartDiseaseorAttack', y='BMI', data=data, palette='Set2')
plt.title('Violin Plot of BMI by Heart Disease Status')
plt.xlabel('Heart Disease Status')
plt.ylabel('BMI')
plt.show()
```



In [ ]:

```
sns.pairplot(data[['Age', 'BMI', 'PhysHlth', 'HeartDiseaseorAttack']], hue='HeartDis
plt.suptitle('Pair Plot of Continuous Variables (colored by Heart Disease)', y=1.02)
plt.show()
```

In [ ]: