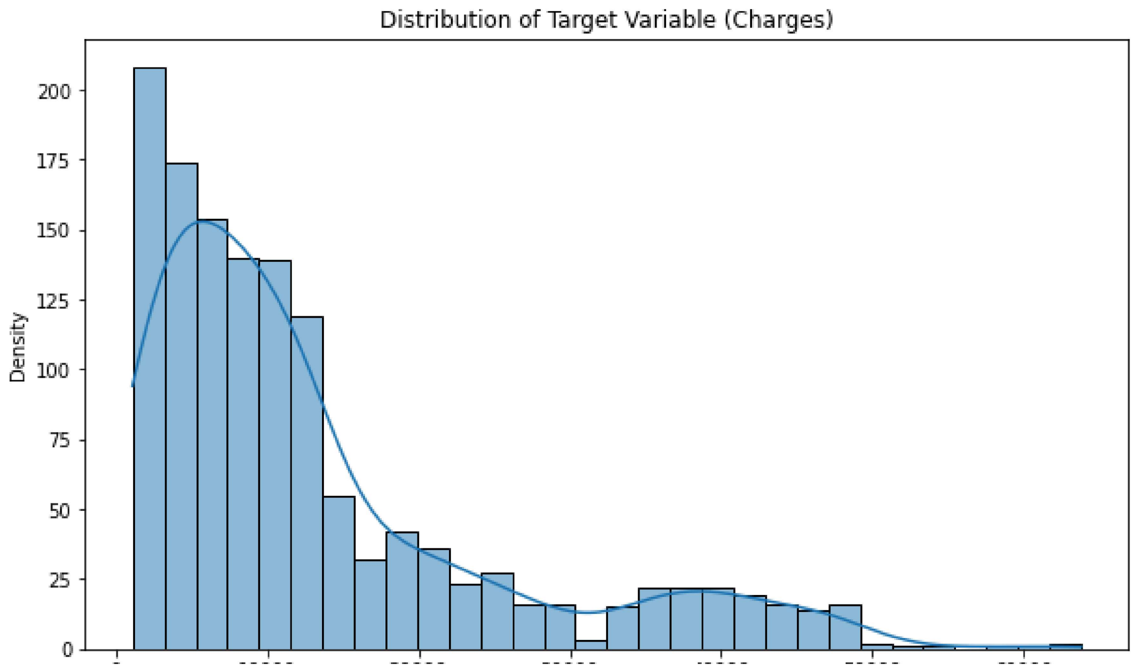
In [6]:

```
pip install pandas matplotlib seaborn
```

```
Requirement already satisfied: pandas in c:\users\91988\anaconda3\lib\site-pa
ckages (1.4.2)
Requirement already satisfied: matplotlib in c:\users\91988\anaconda3\lib\sit
e-packages (3.5.1)
Requirement already satisfied: seaborn in c:\users\91988\anaconda3\lib\site-p
ackages (0.11.2)
Requirement already satisfied: numpy>=1.18.5 in c:\users\91988\anaconda3\lib
\site-packages (from pandas) (1.21.5)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\91988\anaco
nda3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\91988\anaconda3\lib\s
ite-packages (from pandas) (2021.3)
Requirement already satisfied: cycler>=0.10 in c:\users\91988\anaconda3\lib\s
ite-packages (from matplotlib) (0.11.0)
Requirement already satisfied: packaging>=20.0 in c:\users\91988\anaconda3\li
b\site-packages (from matplotlib) (21.3)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\91988\anaconda3
\lib\site-packages (from matplotlib) (1.3.2)
Requirement already satisfied: pillow>=6.2.0 in c:\users\91988\anaconda3\lib
\site-packages (from matplotlib) (9.0.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\91988\anaconda3
\lib\site-packages (from matplotlib) (4.25.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\91988\anaconda3\l
ib\site-packages (from matplotlib) (3.0.4)
Requirement already satisfied: scipy>=1.0 in c:\users\91988\anaconda3\lib\sit
e-packages (from seaborn) (1.7.3)
Requirement already satisfied: six>=1.5 in c:\users\91988\anaconda3\lib\site-
packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [9]:

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

data_path = "C:/Users/91988/Downloads/health_dataset.csv"
try:
    health_data = pd.read_csv(data_path)
except Exception as e:
    print(f"Error loading the dataset: {e}")

print("First few rows of the dataset:")
print(health_data.head())

print("\nColumn names in the dataset:")
print(health_data.columns)

target_column = 'charges'


if target_column not in health_data.columns:
    raise KeyError(f"Target column '{target_column}' not found in the dataset.

feature_columns = health_data.columns.difference([target_column])  # All other


print("\nBasic statistics of the dataset:")
print(health_data.describe())

# Check for missing values
print("\nMissing values in each column:")
print(health_data.isnull().sum())

# Visualize missing values
plt.figure(figsize=(10, 5))
sns.heatmap(health_data.isnull(), cbar=False, cmap='viridis')
plt.title('Missing Values Heatmap')
plt.show()

# Data Cleaning
# Fill missing values or drop columns with a high percentage of missing data
# Fill missing numerical values with the mean
for col in feature_columns:
    if health_data[col].isnull().any():
        if health_data[col].dtype == 'float64' or health_data[col].dtype == 'i
            health_data[col].fillna(health_data[col].mean(), inplace=True)
        else:
            health_data[col].fillna(health_data[col].mode()[0], inplace=True)

# Display the cleaned data information
print("\nInformation about the dataset after cleaning:")
print(health_data.info())

# Exploratory Data Analysis (EDA)
# Count plot of the target variable
try:
    plt.figure(figsize=(10, 6))
    sns.histplot(health_data[target_column], bins=30, kde=True)
    plt.title('Distribution of Target Variable (Charges)')
    plt.xlabel('Charges')
    plt.ylabel('Density')
    plt.show()
except KeyError as e:
    print(f"Error: {e}. Please check if the target column name is correct.")

# Pairplot to visualize relationships between numerical features
sns.pairplot(health_data)
plt.title('Pairplot of Health Dataset')
plt.show()

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


try:
    plt.figure(figsize=(10, 6))
    sns.scatterplot(x='age', y='charges', data=health_data)  # <-- Update with
    plt.title('Age vs. Charges')
    plt.xlabel('Age')
    plt.ylabel('Charges')
    plt.show()
except KeyError as e:
    print(f"Error: {e}. Please check if the column names are correct.")
```



```
In [ ]:
```