In [ ]:

```
!pip install pandas
!pip install matplotlib
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (2.2.2)
Requirement already satisfied: numpy>=1.22.4 in /usr/local/lib/python3.10/dist-packages (
from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-p
ackages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (f
rom pandas) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages
(from pandas) (2024.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from
python-dateutil>=2.8.2->pandas) (1.16.0)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (3.8
.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-package
s (from matplotlib) (1.3.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (f
rom matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packag
es (from matplotlib) (4.54.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packag
es (from matplotlib) (1.4.7)
Requirement already satisfied: numpy<2,>=1.21 in /usr/local/lib/python3.10/dist-packages
(from matplotlib) (1.26.4)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages
(from matplotlib) (24.1)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (
from matplotlib) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-package
s (from matplotlib) (3.2.0)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-pac
kages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from
python-dateutil>=2.7->matplotlib) (1.16.0)
```

In [ ]:

```
from google.colab import files
uploaded = files.upload()
```

Choose File **No file selected**

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-2-21dc3c638f66> in <cell line: 2>()
      1 from google.colab import files
----> 2 uploaded = files.upload()

/usr/local/lib/python3.10/dist-packages/google/colab/files.py in upload(target_dir)
     70   """
     71 
---> 72   uploaded_files = _upload_files(multiple=True)
     73   # Mapping from original filename to filename as saved locally.
     74   local_filenames = dict()

/usr/local/lib/python3.10/dist-packages/google/colab/files.py in _upload_files(multiple)
    162 
    163   # First result is always an indication that the file picker has completed.
--> 164   result = _output.eval_js(
    165       'google.colab._files._uploadFiles("{input_id}", "{output_id}")'.format(
    166           input_id=input_id, output_id=output_id
```

```
/usr/local/lib/python3.10/dist-packages/google/colab/output/_js.py in eval_js(script, ign
ore_result, timeout_sec)
     38   if ignore_result:
     39     return
---> 40   return _message.read_reply_from_input(request_id, timeout_sec)
     41
     42

/usr/local/lib/python3.10/dist-packages/google/colab/_message.py in read_reply_from_input
(message_id, timeout_sec)
     94     reply = _read_next_input_message()
     95     if reply == _NOT_READY or not isinstance(reply, dict):
---> 96       time.sleep(0.025)
     97       continue
     98     if (

KeyboardInterrupt:
```

In [ ]:

```
import pandas as pd
import matplotlib as plt

df1 = pd.read_csv('heart_2022_no_nans.csv')
df1
```

Out[ ]:

	State	Sex	GeneralHealth	PhysicalHealthDays	MentalHealthDays	LastCheckupTime	PhysicalActivities	SleepH
0	Alabama	Female	Very good	4.0	0.0	Within past year (anytime less than 12 months ...	Yes	
1	Alabama	Male	Very good	0.0	0.0	Within past year (anytime less than 12 months ...	Yes	
2	Alabama	Male	Very good	0.0	0.0	Within past year (anytime less than 12 months ...	No	
3	Alabama	Female	Fair	5.0	0.0	Within past year (anytime less than 12 months ...	Yes	
4	Alabama	Female	Good	3.0	15.0	Within past year (anytime less than 12 months ...	Yes	
...	...	...	...	...	...	...	...	
246017	Virgin Islands	Male	Very good	0.0	0.0	Within past 2 years (1 year but less than 2 ye...	Yes	
246018	Virgin Islands	Female	Fair	0.0	7.0	Within past year (anytime less than 12 months ...	Yes	
246019	Virgin Islands	Male	Good	0.0	15.0	Within past year (anytime less than 12 months ...	Yes	
246020	Virgin Islands	Female	Excellent	2.0	2.0	Within past year (anytime less than 12 months ...	Yes	
246021	Virgin Islands	Male	Very good	0.0	0.0	Within past year (anytime less than 12 months ...	No	

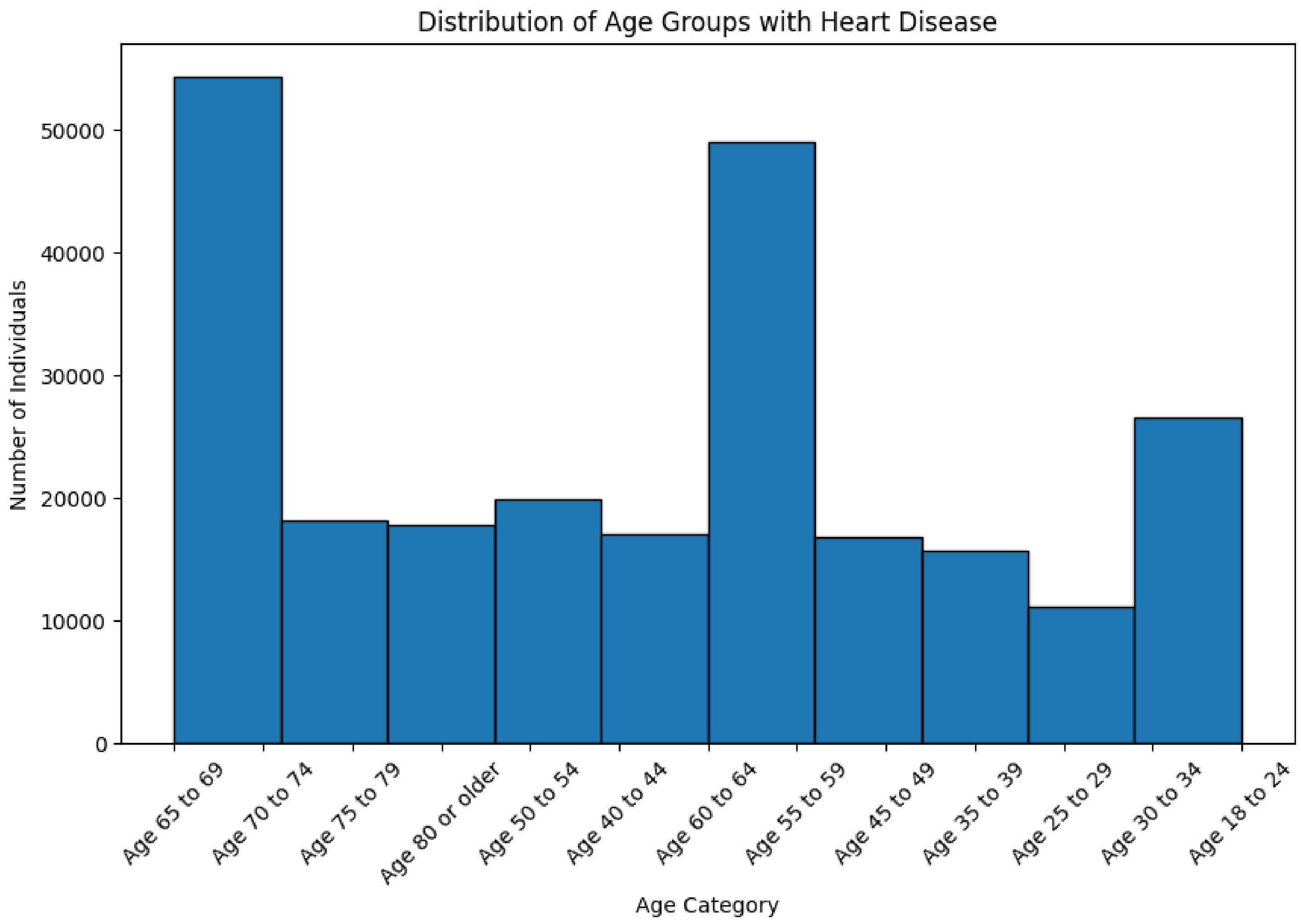
246022 rows × 40 columns

In [50]:

```
import pandas as pd
import matplotlib.pyplot as plt
df1 = pd.read_csv('heart_2022_no_nans.csv')

# Question 1: What is the distribution of Heart Disease across different age groups?
# Visualization: Histogram
plt.figure(figsize=(10, 6))
plt.hist(df1['AgeCategory'], bins=10, edgecolor='black')
plt.title('Distribution of Age Groups with Heart Disease')
plt.xlabel('Age Category')
plt.ylabel('Number of Individuals')
plt.xticks(rotation=45)
plt.show()

# Observation:
# A much taller bar for an older age group compared to a younger one, it could shows more
heart disease cases in the older group
```



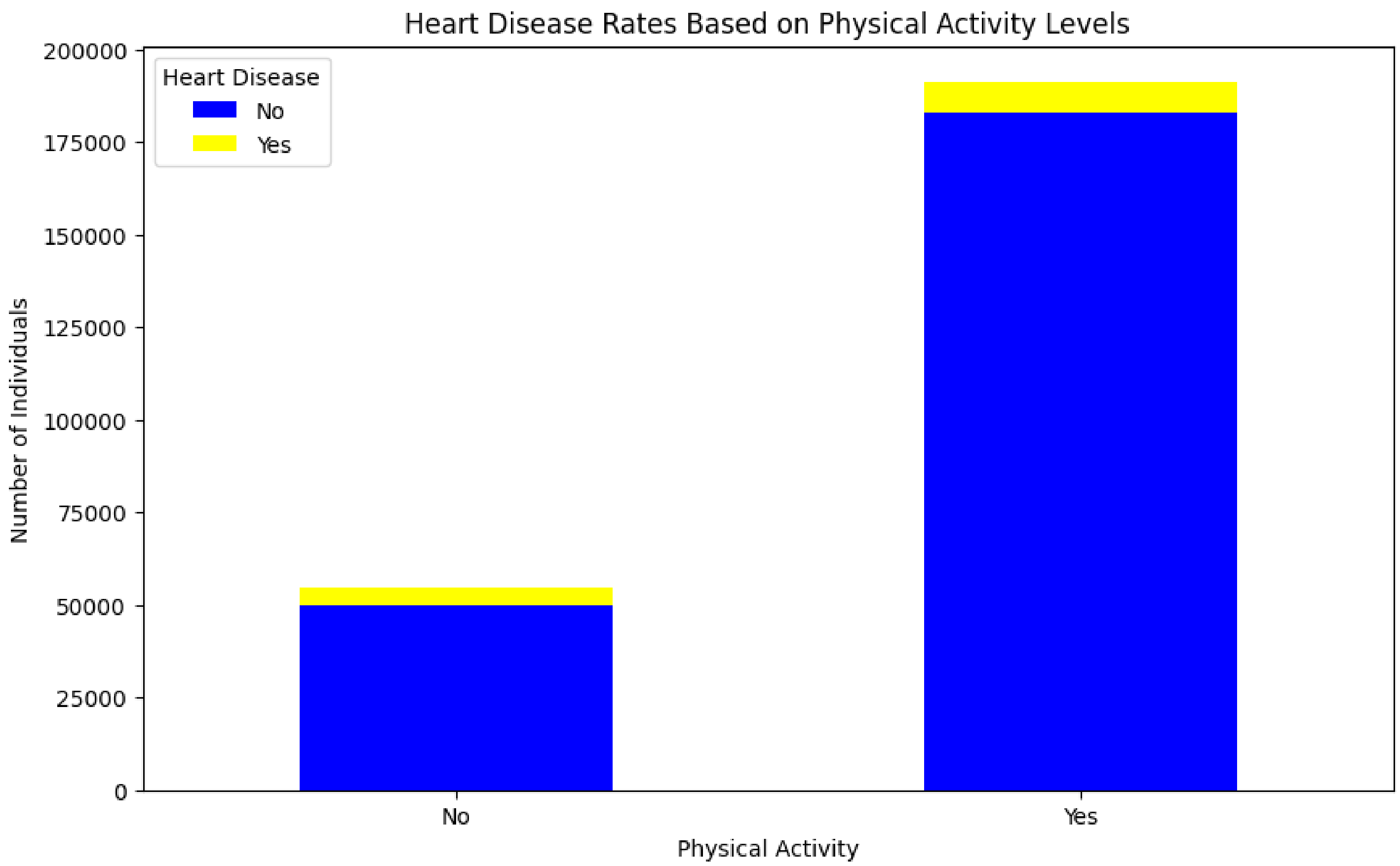
In [51]:

```
import matplotlib.pyplot as plt
import pandas as pd
df1 = pd.read_csv('heart_2022_no_nans.csv')

# Question 2: How common is heart disease among people based on whether or not they’re ph
ysically active?
activity_heart_counts = df1.groupby('PhysicalActivities')['HadHeartAttack'].value_counts
().unstack().fillna(0)

plt.figure(figsize=(10, 6))
activity_heart_counts.plot(kind='bar', stacked=True, ax=plt.gca(), color=['blue', 'yello
w'])
plt.title('Heart Disease Rates Based on Physical Activity Levels')
plt.xlabel('Physical Activity')
plt.ylabel('Number of Individuals')
plt.legend(title='Heart Disease', labels=['No', 'Yes'], loc='best')
plt.xticks(rotation=0)
plt.show()

#Observation: People who are more physically active tend to have fewer heart attacks.
#People who are less physically active seem to have more heart attacks.
```



In [ ]:

```
import pandas as pd
import matplotlib as plt

df2 = pd.read_csv('heart_2022_with_nans.csv')
df2
```

Out[ ]:

	State	Sex	GeneralHealth	PhysicalHealthDays	MentalHealthDays	LastCheckupTime	PhysicalActivities	SleepH
0	Alabama	Female	Very good	0.0	0.0	Within past year (anytime less than 12 months ...	No	
1	Alabama	Female	Excellent	0.0	0.0	NaN	No	
2	Alabama	Female	Very good	2.0	3.0	Within past year (anytime less than 12 months ...	Yes	
3	Alabama	Female	Excellent	0.0	0.0	Within past year (anytime less than 12 months ...	Yes	
4	Alabama	Female	Fair	2.0	0.0	Within past year (anytime less than 12 months ...	Yes	
...	...	...	...	...	...	...	...	
445127	Virgin Islands	Female	Good	0.0	3.0	Within past 2 years (1 year but less than 2 ye...	Yes	
445128	Virgin Islands	Female	Excellent	2.0	2.0	Within past year (anytime less than 12 months ...	Yes	

	State	Sex	GeneralHealth	PhysicalHealthDays	MentalHealthDays	LastCheckupTime	PhysicalActivities	SleepH
445129	Virgin Islands	Female	Poor	30.0	30.0	5 or more years ago	No	
445130	Virgin Islands	Male	Very good	0.0	0.0	Within past year (anytime less than 12 months ...	No	
445131	Virgin Islands	Male	Very good	0.0	1.0	NaN	Yes	

445132 rows × 40 columns

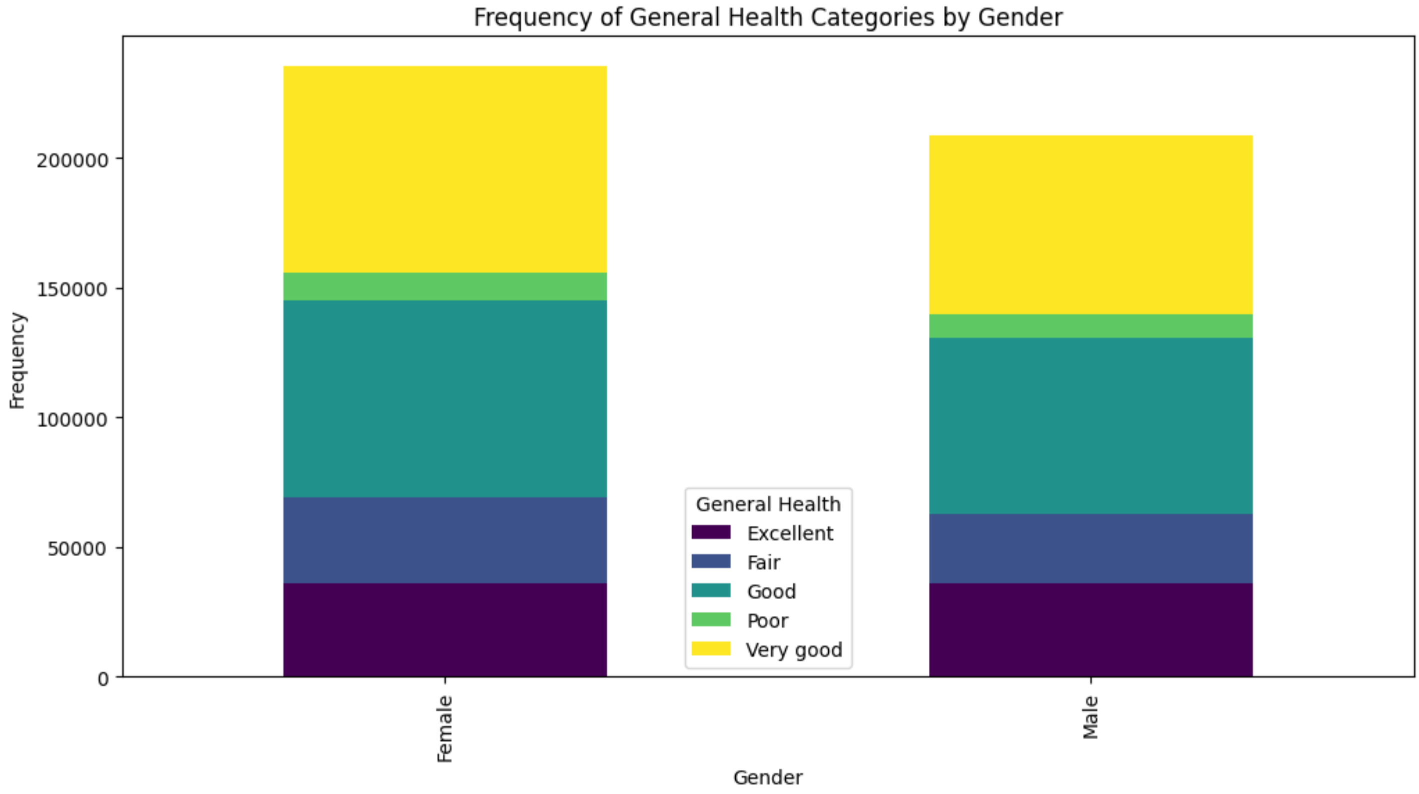
In [ ]:

```
import matplotlib.pyplot as plt
import pandas as pd
df = pd.read_csv('heart_2022_with_nans.csv')

# Question 1: Frequency of General Health Categories by Gender
# Filter data to drop NaNs in 'Gender' and 'GeneralHealth'
health_gender_data = df[['Sex', 'GeneralHealth']].dropna()

# Count occurrences of each GeneralHealth category by Gender
health_gender_counts = health_gender_data.groupby(['Sex', 'GeneralHealth']).size().unsta
ck()

health_gender_counts.plot(kind='bar', stacked=True, figsize=(12, 6), colormap='viridis')
plt.title("Frequency of General Health Categories by Gender")
plt.xlabel("Gender")
plt.ylabel("Frequency")
plt.legend(title="General Health")
plt.show()
#Observation: The bar chart shows that Females tend to have very good health, While Males
does not.
#it also seems that Females tend to have poor health compared to Males
```



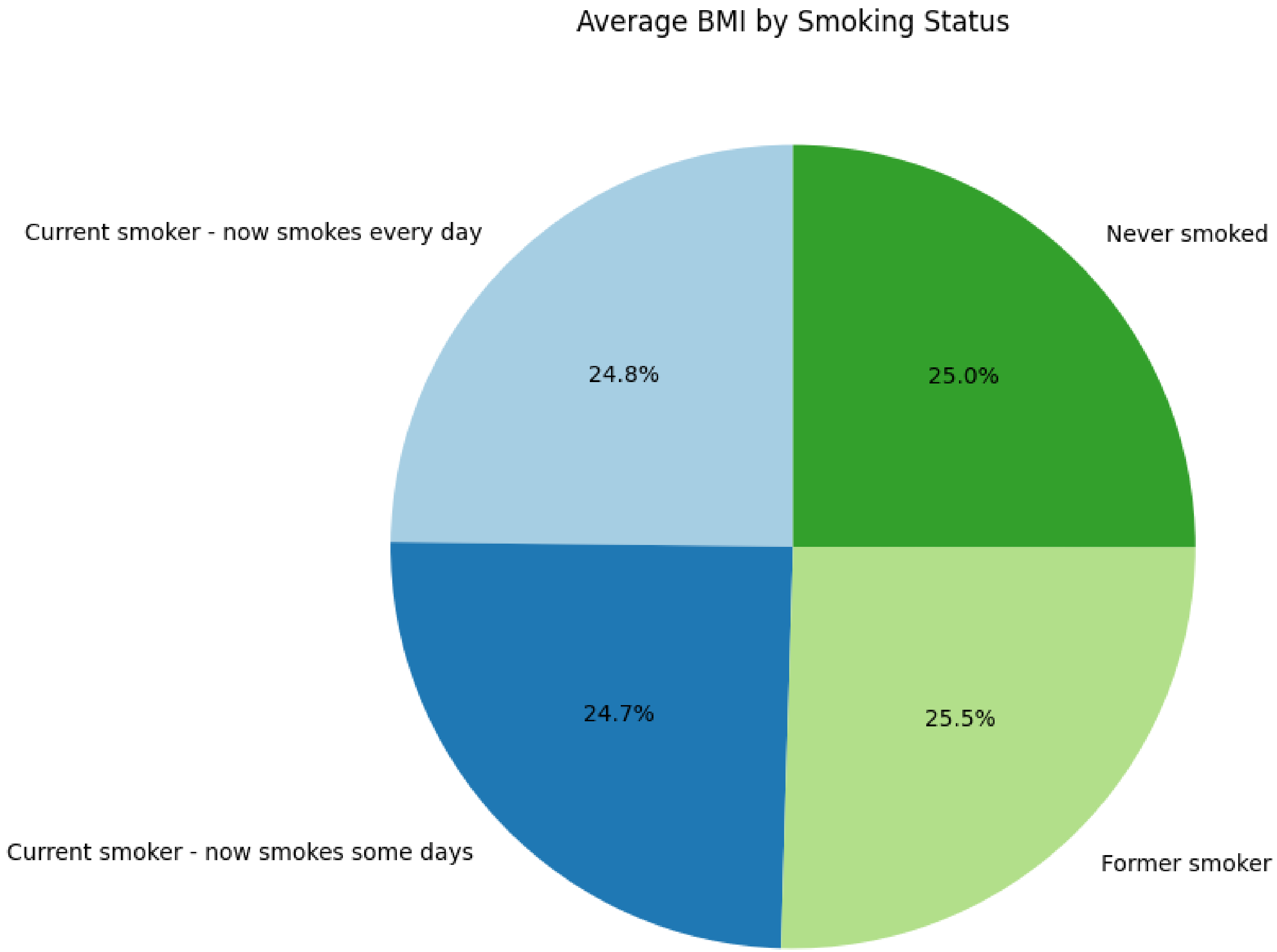
In [ ]:

```
import matplotlib.pyplot as plt
import pandas as pd
df = pd.read_csv('heart_2022_with_nans.csv')
```

```python
# Question 2: Average BMI by Smoking Status
# Filter data to drop NaNs in 'BMI' and 'SmokerStatus'
bmi_smoker_data = df[['SmokerStatus', 'BMI']].dropna()

# Calculate average BMI for each smoking status
avg_bmi_by_smoker_status = bmi_smoker_data.groupby('SmokerStatus')['BMI'].mean()


plt.figure(figsize=(8, 8))
plt.pie(avg_bmi_by_smoker_status, labels=avg_bmi_by_smoker_status.index, autopct='%1.1f%%', startangle=90, colors=plt.cm.Paired.colors)
plt.title("Average BMI by Smoking Status")
plt.show()
#Observation:
#The Pie chart indicates that smoking status doesn’t seem to have a significant effect on BMI, as the averages are quite close across all groups.
#The fact that Former Smokers have a higher BMI might suggest that people tend to gain weight after they quit smoking.
```


Average BMI by Smoking Status
Current smoker - now smokes every day
Never smoked
Current smoker - now smokes some days
Former smoker