

```
from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive
```

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

df = pd.read_csv('/content/drive/MyDrive/Project 2/survey_lung_cancer.csv')
```

df.head()

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL CONSUMING	COUGHING	SHORTNESS OF BREATH	SWALLOWING DIFFICULTY	CHEST PAIN	LUNG_CANCER
0	M	69	1		2	2	1	1	2	1	2		2		2	YES
1	M	74	2		1	1	1	2	2	2	1		1		2	YES
2	F	59	1		1	1	2	1	2	1	2		1		2	NO
3	M	63	2		2	2	1	1	1	1	2		1		2	NO
4	F	63	1		2	1	1	1	1	2		1	2		1	NO

df.dtypes

	0
GENDER	object
AGE	int64
SMOKING	int64
YELLOW_FINGERS	int64
ANXIETY	int64
PEER_PRESSURE	int64
CHRONIC DISEASE	int64
FATIGUE	int64
ALLERGY	int64
WHEEZING	int64
ALCOHOL CONSUMING	int64
COUGHING	int64
SHORTNESS OF BREATH	int64
SWALLOWING DIFFICULTY	int64
CHEST PAIN	int64
LUNG_CANCER	object

dtype: object

df.columns

```
Index(['GENDER', 'AGE', 'SMOKING', 'YELLOW_FINGERS', 'ANXIETY',
      'PEER_PRESSURE', 'CHRONIC DISEASE', 'FATIGUE ', 'ALLERGY ', 'WHEEZING',
      'ALCOHOL CONSUMING', 'COUGHING', 'SHORTNESS OF BREATH',
      'SWALLOWING DIFFICULTY', 'CHEST PAIN', 'LUNG_CANCER'],
      dtype='object')
```

df.isnull().sum()

	0
GENDER	0
AGE	0
SMOKING	0
YELLOW_FINGERS	0
ANXIETY	0
PEER_PRESSURE	0
CHRONIC DISEASE	0
FATIGUE	0
ALLERGY	0
WHEEZING	0
ALCOHOL CONSUMING	0
COUGHING	0
SHORTNESS OF BREATH	0
SWALLOWING DIFFICULTY	0
CHEST PAIN	0
LUNG_CANCER	0

dtype: int64

```
missing_percentage = df.isnull().mean() * 100
threshold = 30
```

```
columns_to_drop = missing_percentage[missing_percentage >
threshold].index
```

```
df = df.drop(columns=columns_to_drop)
```

```
print(f"Columns dropped: {list(columns_to_drop)}")
print(f"Remaining columns: {df.columns}")
```

```
Columns dropped: []
Remaining columns: Index(['GENDER', 'AGE', 'SMOKING', 'YELLOW_FINGERS', 'ANXIETY',
      'PEER_PRESSURE', 'CHRONIC DISEASE', 'FATIGUE ', 'ALLERGY ', 'WHEEZING',
      'ALCOHOL CONSUMING', 'COUGHING', 'SHORTNESS OF BREATH',
      'SWALLOWING DIFFICULTY', 'CHEST PAIN', 'LUNG_CANCER'],
      dtype='object')
```

```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
for column in df.select_dtypes(include=['object']).columns:
    df[column] = label_encoder.fit_transform(df[column])
df.head()
```

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL CONSUMING	COUGHING	SHORTNESS OF BREATH	SWALLOWING DIFFICULTY	CHEST PAIN	LUNG_CANCER
0	1	69	1		2	2	1	1	2	1	2		2		2	1
1	1	74	2		1	1	1	2	2	2	1		1		2	1
2	0	59	1		1	1	2	1	2	1	2		2		1	0
3	1	63	2		2	2	1	1	1	1	2		1		2	0
4	0	63	1		2	1	1	1	1	2		1	2		1	0

```
for column in df.select_dtypes(include=['float64']).columns:
    mean_value = df[column].mean()
```

```
for column in df.select_dtypes(include=['object']).columns:
    mode_value = df[column].mode()[0]
```

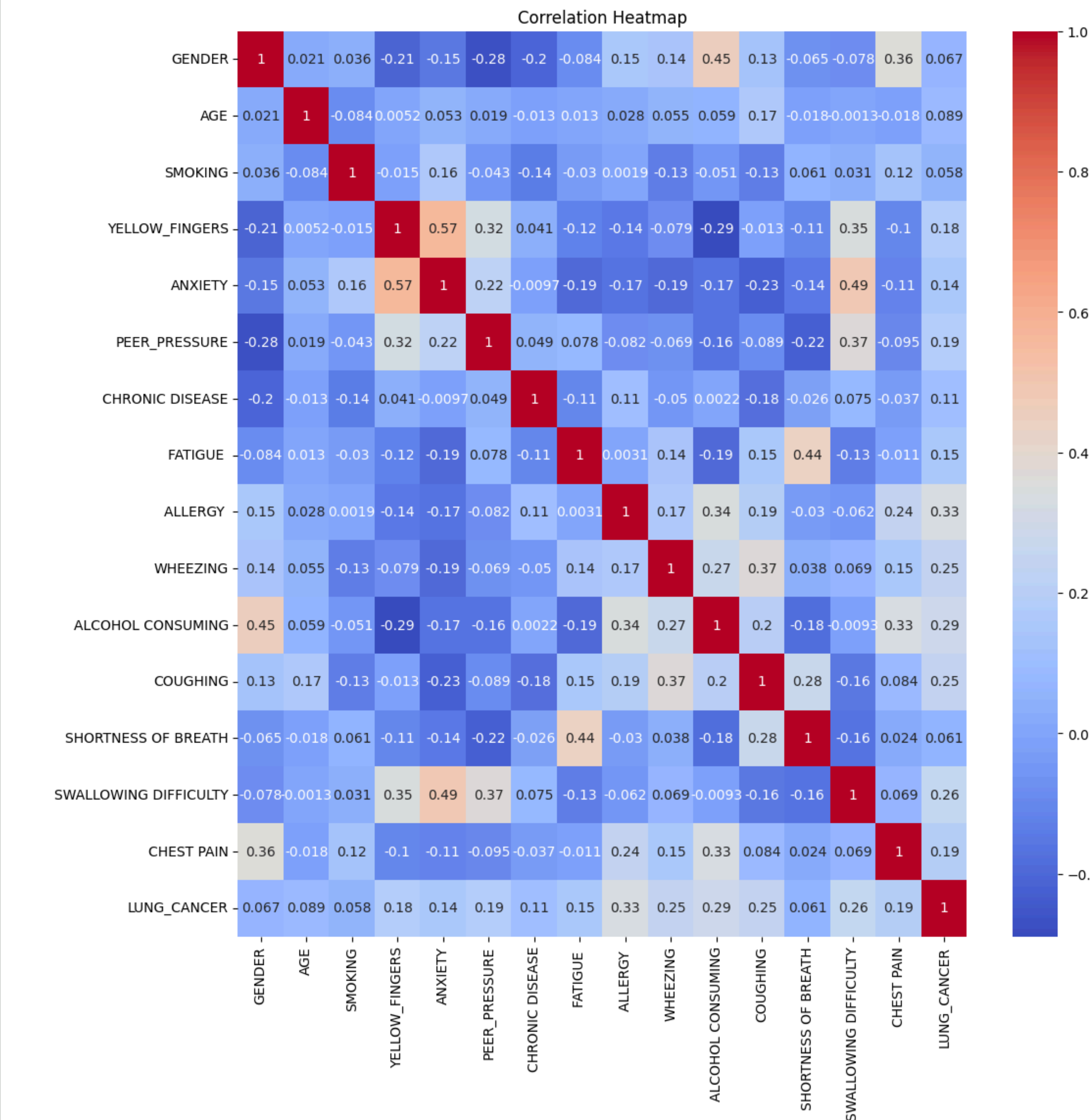
```
for column in df.select_dtypes(include=['object']).columns:
    mode_value = df[column].mode()[0]
df[column] = df[column].fillna(mode_value)
df.isnull().sum()
```

	0
GENDER	0
AGE	0
SMOKING	0
YELLOW_FINGERS	0
ANXIETY	0
PEER_PRESSURE	0
CHRONIC DISEASE	0
FATIGUE	0
ALLERGY	0
WHEEZING	0
ALCOHOL CONSUMING	0
COUGHING	0
SHORTNESS OF BREATH	0
SWALLOWING DIFFICULTY	0
CHEST PAIN	0
LUNG_CANCER	0

dtype: int64

```
plt.figure(figsize=(12, 12))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
```

Text(0.5, 1.0, 'Correlation Heatmap')



```
plt.show()
```

```
df['LUNG_CANCER'].value_counts()
```

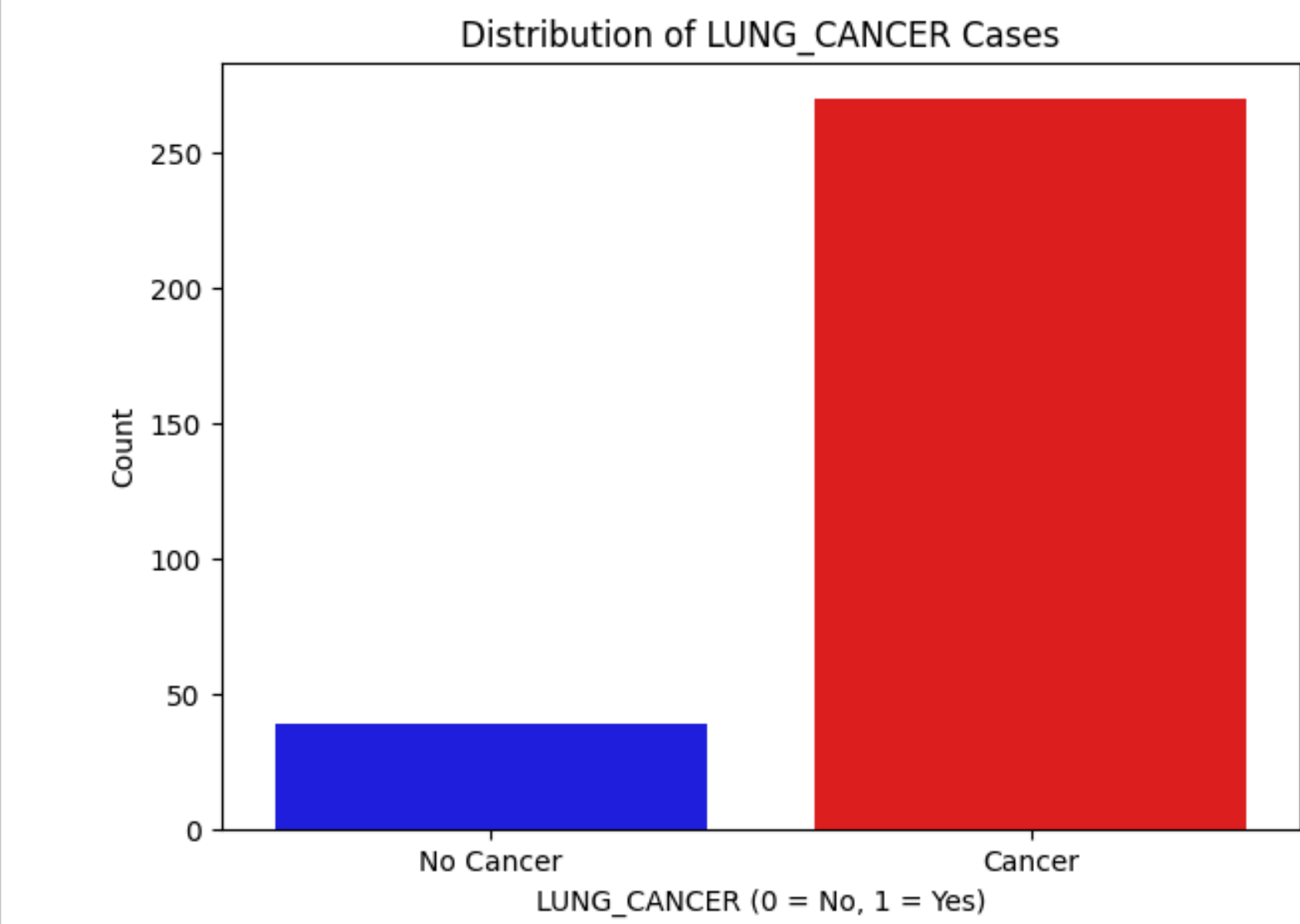
	count
LUNG_CANCER	
1	270
0	39

dtype: int64

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

lung_cancer_counts = df['LUNG_CANCER'].value_counts()

plt.figure(figsize=(7, 5))
sns.barplot(x=lung_cancer_counts.index, y=lung_cancer_counts.values, palette=['blue', 'red'])
plt.title('Distribution of LUNG_CANCER Cases')
plt.xlabel('LUNG_CANCER (0 = No, 1 = Yes)')
plt.ylabel('Count')
plt.xticks(ticks=[0, 1], labels=['No Cancer', 'Cancer'])
plt.show()
```

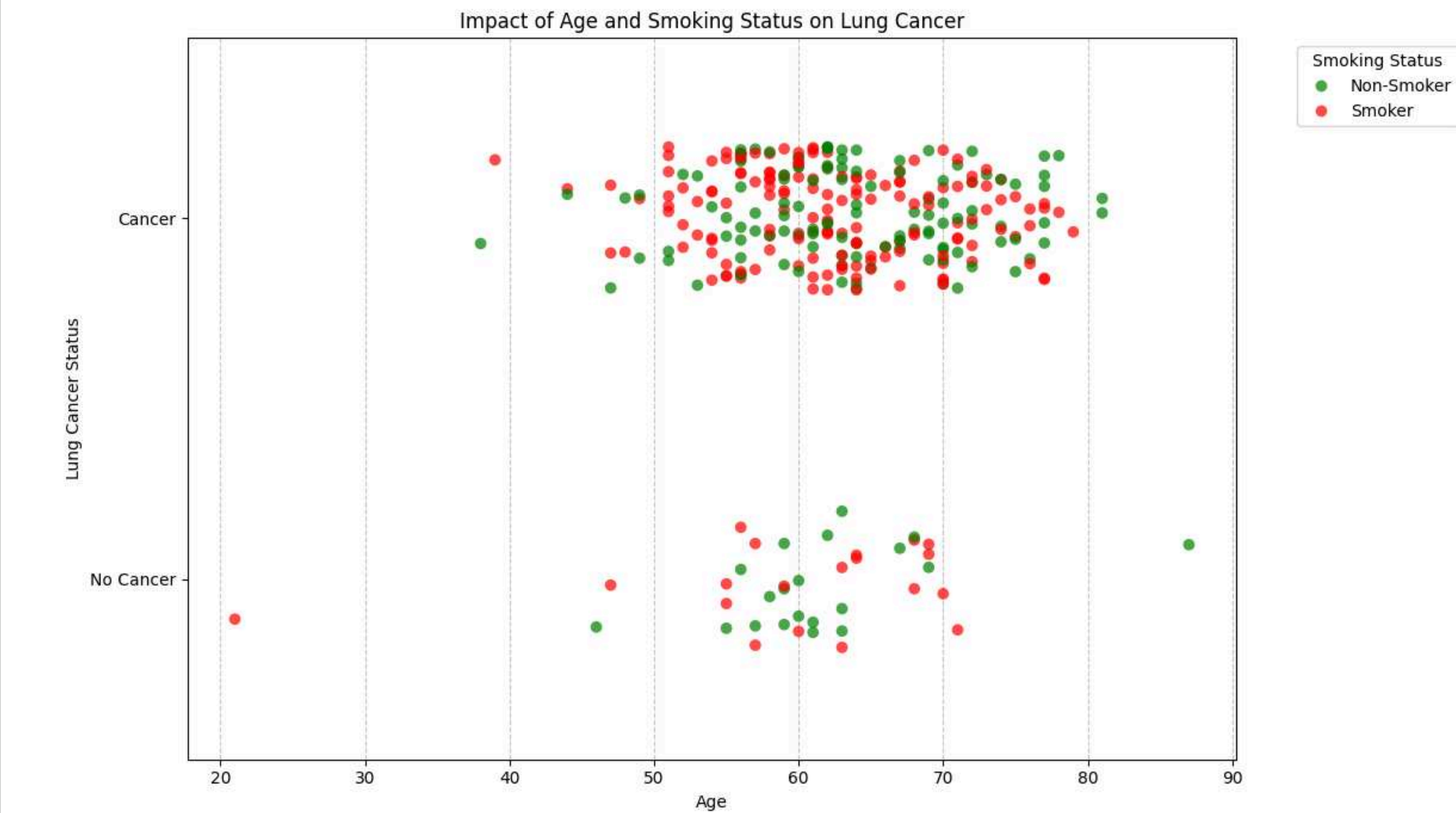


```
import matplotlib.pyplot as plt
import seaborn as sns

# Map numeric values to clear labels for better visualization
plot_df = df.copy()
plot_df['LUNG_CANCER_STATUS'] = plot_df['LUNG_CANCER'].map({0: 'No Cancer', 1: 'Cancer'})
plot_df['SMOKING_STATUS'] = plot_df['SMOKING'].map({1: 'Non-Smoker', 2: 'Smoker'})

plt.figure(figsize=(12, 7))
sns.stripplot(
    data=plot_df,
    x='AGE',
    y='LUNG_CANCER_STATUS',
    hue='SMOKING_STATUS',
    jitter=0.2, # Add jitter to spread out overlapping points
    palette=('Non-Smoker': 'green', 'Smoker': 'red'), # Custom colors
    marker='o', # Marker style
    alpha=0.7, # Transparency
    s=7 # size of points
)

plt.title('Impact of Age and Smoking Status on Lung Cancer')
plt.xlabel('Age')
plt.ylabel('Lung Cancer Status')
plt.grid(axis='x', linestyle='--', alpha=0.7) # Only x-axis grid for better readability
plt.legend(title='Smoking Status', bbox_to_anchor=(1.05, 1), loc='upper left') # Place legend outside
plt.tight_layout() # Adjust layout to prevent labels overlapping
plt.show()
```



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

# Ensure column names are stripped of whitespace for consistent access
df.columns = df.columns.str.strip()

# Filter for lung cancer patients (LUNG_CANCER == 1)
cancer_patients_df = df[df['LUNG_CANCER'] == 1]

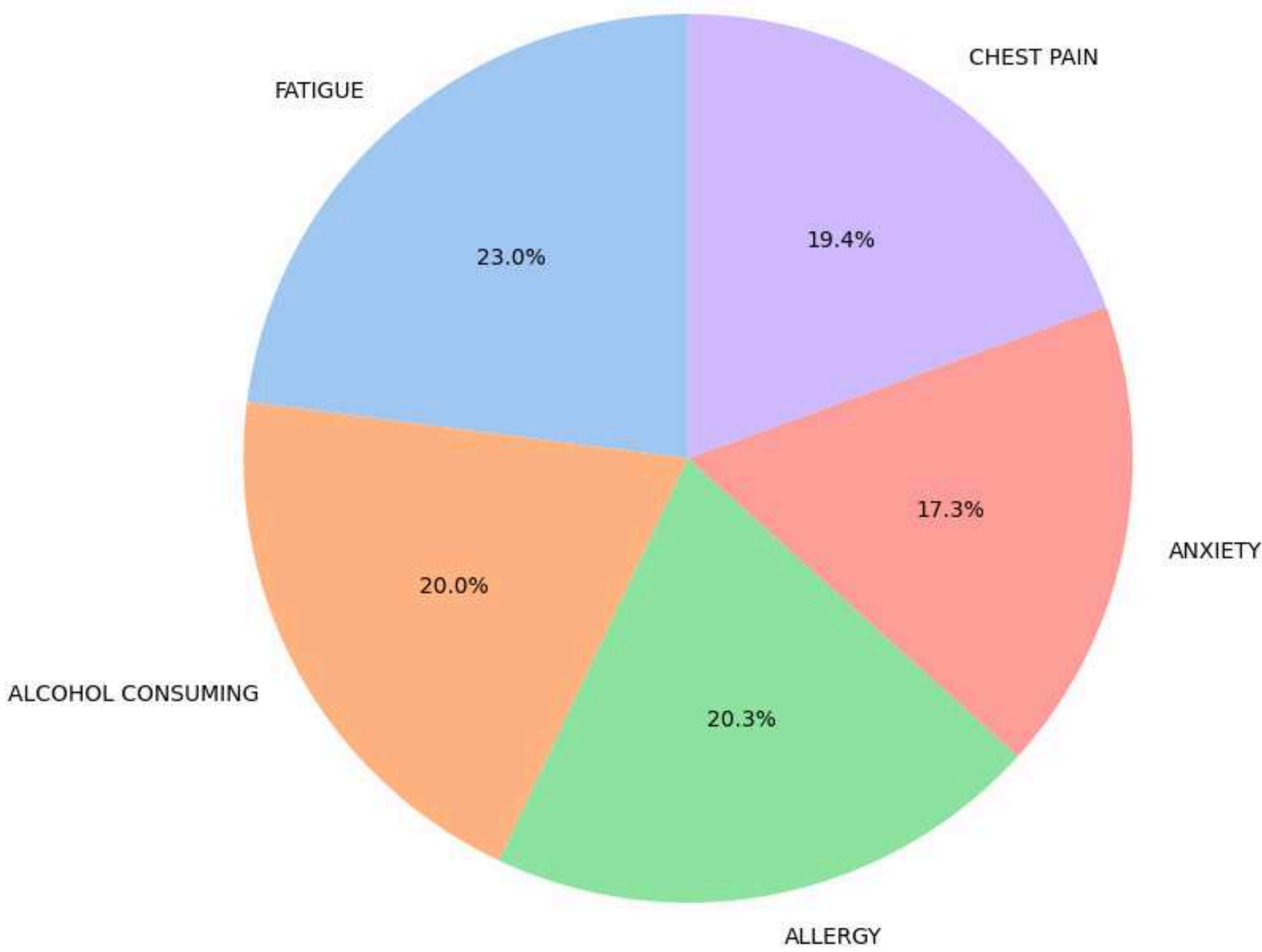
# Define the columns representing the symptoms/habits of interest
symptom_columns = ['FATIGUE', 'ALCOHOL_CONSUMING', 'ALLERGY', 'ANXIETY', 'CHEST_PAIN']
```

```
# Calculate the count of 'Yes' (value 2) for each symptom among cancer patients
symptom_counts = {}
for col in symptom_columns:
    symptom_counts[col] = (cancer_patients_df[col] == 2).sum()

# Convert to a Series for easier plotting
symptom_series = pd.Series(symptom_counts)

# Create the pie chart
plt.figure(figsize=(10, 8))
plt.pie(symptom_series, labels=symptom_series.index, autopct='%1.1f%%', startangle=90, colors=sns.color_palette('pastel'))
plt.title('Proportion of Reported Symptoms/Habits Among Lung Cancer Patients')
plt.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()
```

Proportion of Reported Symptoms/Habits Among Lung Cancer Patients

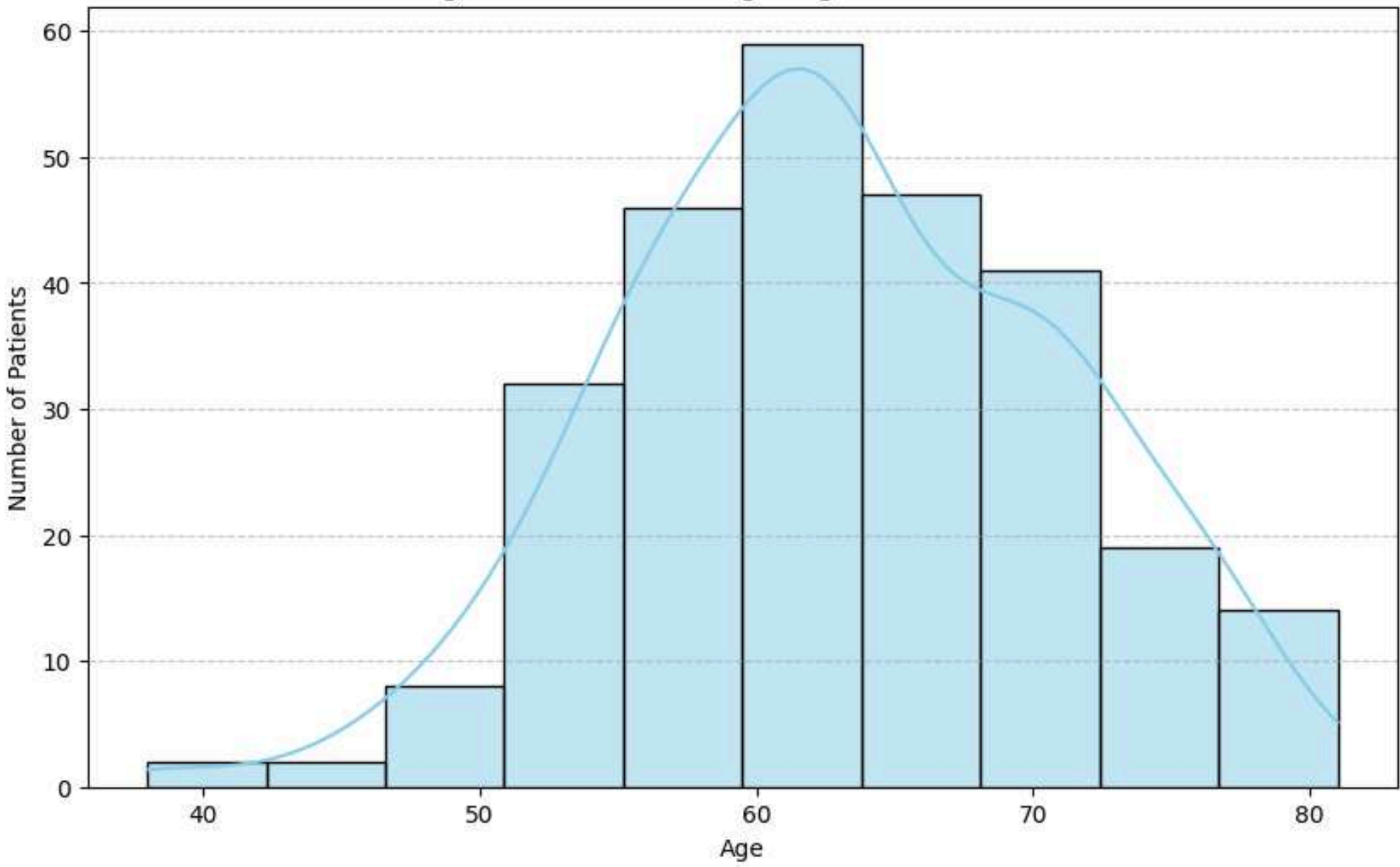


```
import matplotlib.pyplot as plt
import seaborn as sns

# Filter for lung cancer patients
cancer_patients_df = df[df['LUNG_CANCER'] == 1]

plt.figure(figsize=(10, 6))
sns.histplot(cancer_patients_df['AGE'], bins=10, kde=True, color='skyblue')
plt.title('Age Distribution Among Lung Cancer Patients')
plt.xlabel('Age')
plt.ylabel('Number of Patients')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

Age Distribution Among Lung Cancer Patients



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

# Get the value counts for the 'GENDER' column
gender_counts = df['GENDER'].value_counts()

# Map the numeric gender values to descriptive labels
gender_labels = gender_counts.index.map({0: 'Female', 1: 'Male'})

# Create the donut chart
plt.figure(figsize=(8, 8))
plt.pie(gender_counts.values, labels=gender_labels, autopct='%1.1f%%', startangle=90, colors=sns.color_palette('pastel'))

# Draw a circle in the center to make it a donut chart
centre_circle = plt.Circle((0,0), 0.70, fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)

plt.title('Distribution of Gender')
plt.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()
```

Distribution of Gender

