

```
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		2	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()

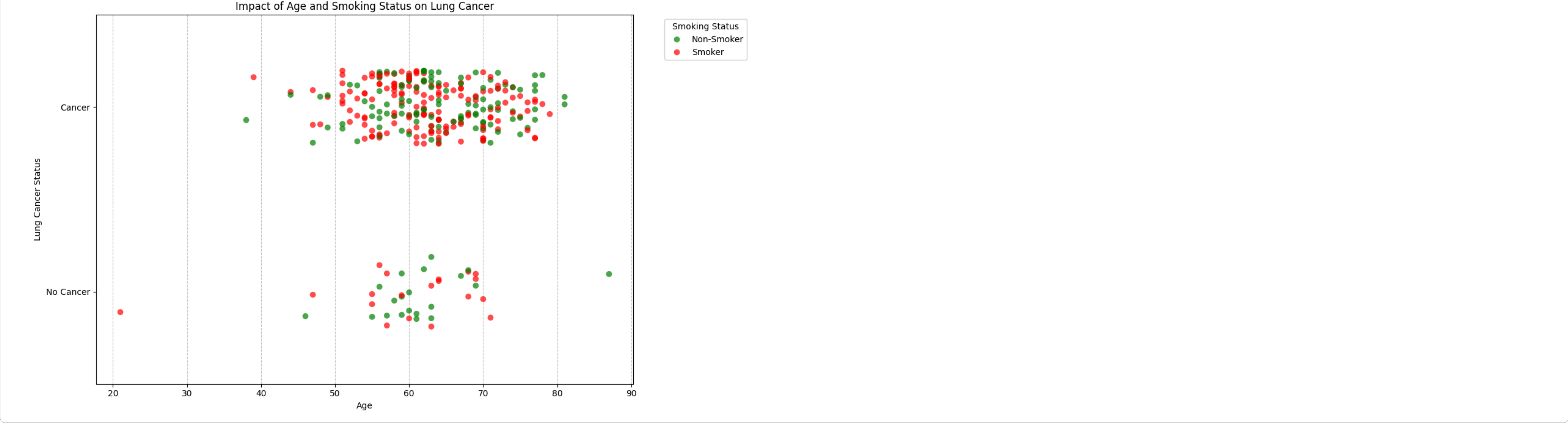
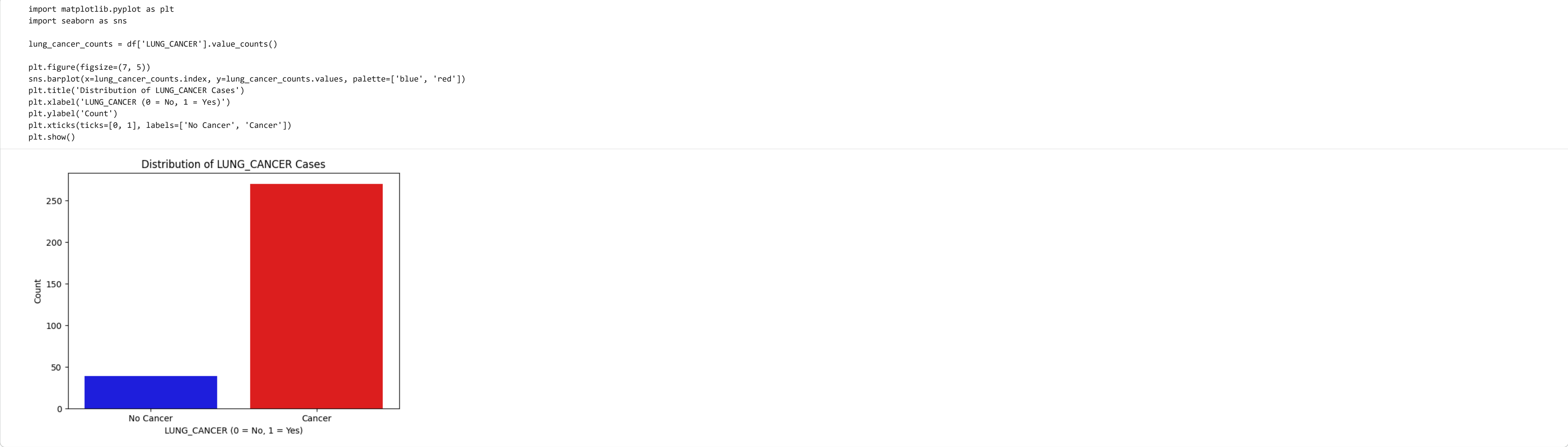
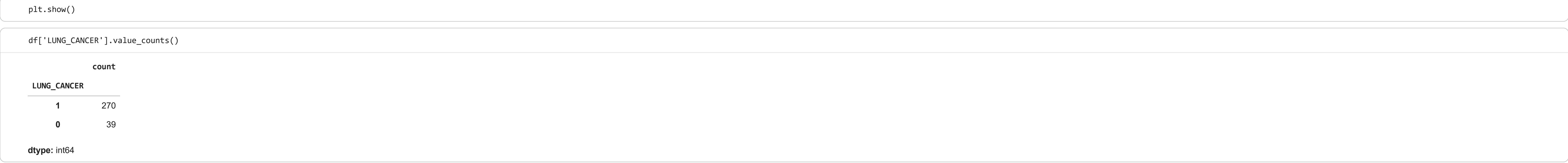
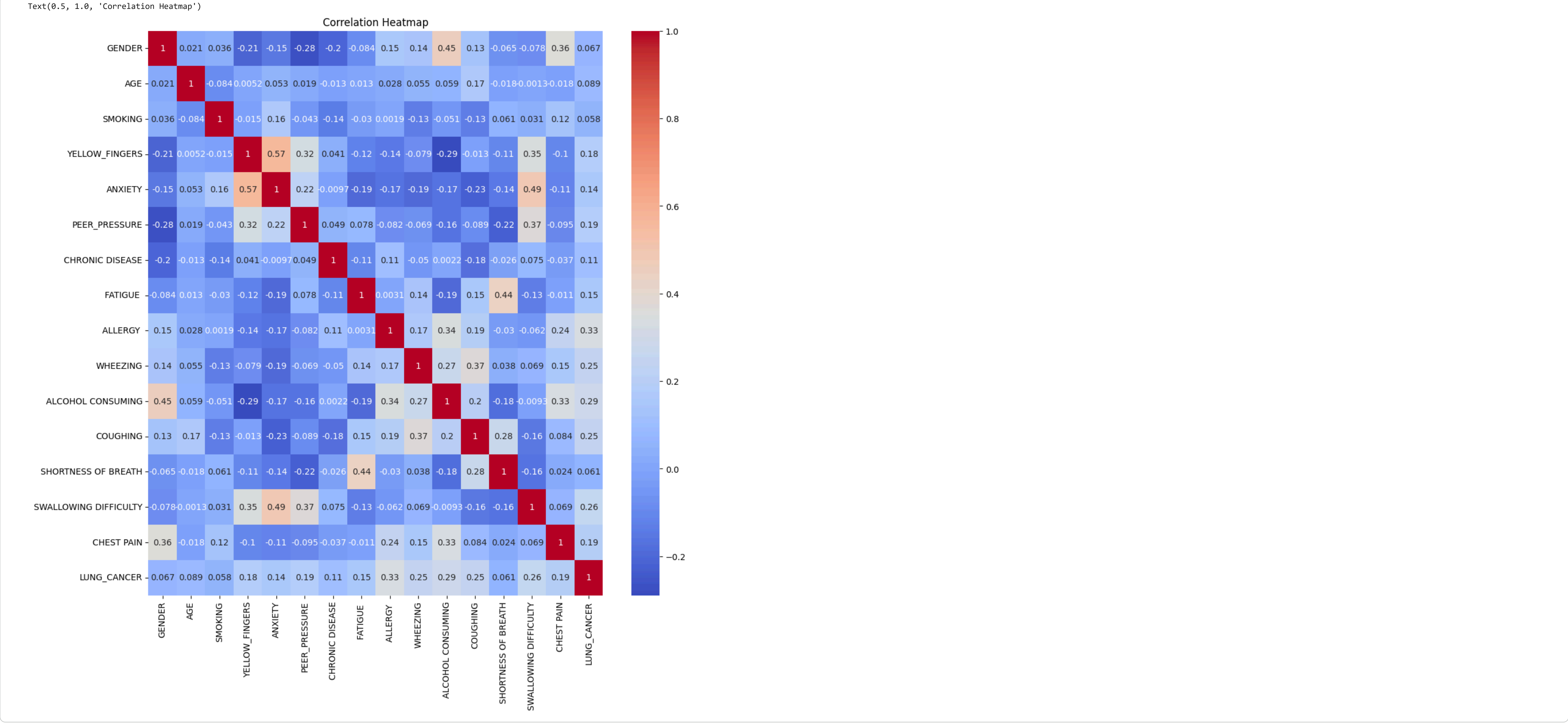
	0
GENDER	0
AGE	0
SMOKING	0
YELLOW_FINGERS	0
ANXIETY	0
PEER_PRESSURE	0
CHRONIC_DISEASE	0
FATIGUE	0
ALLERGY	0
WHEEZING	0
ALCOHOL_CONSUMING	0
COUGHING	0
SHORTNESS OF BREATH	0
SWALLOWING DIFFICULTY	0
CHEST PAIN	0
LUNG_CANCER	0

dtype: int64

plt.figure(figsize=(12, 12))

sns.heatmap(df.corr(), annot=True, cmap='coolwarm')

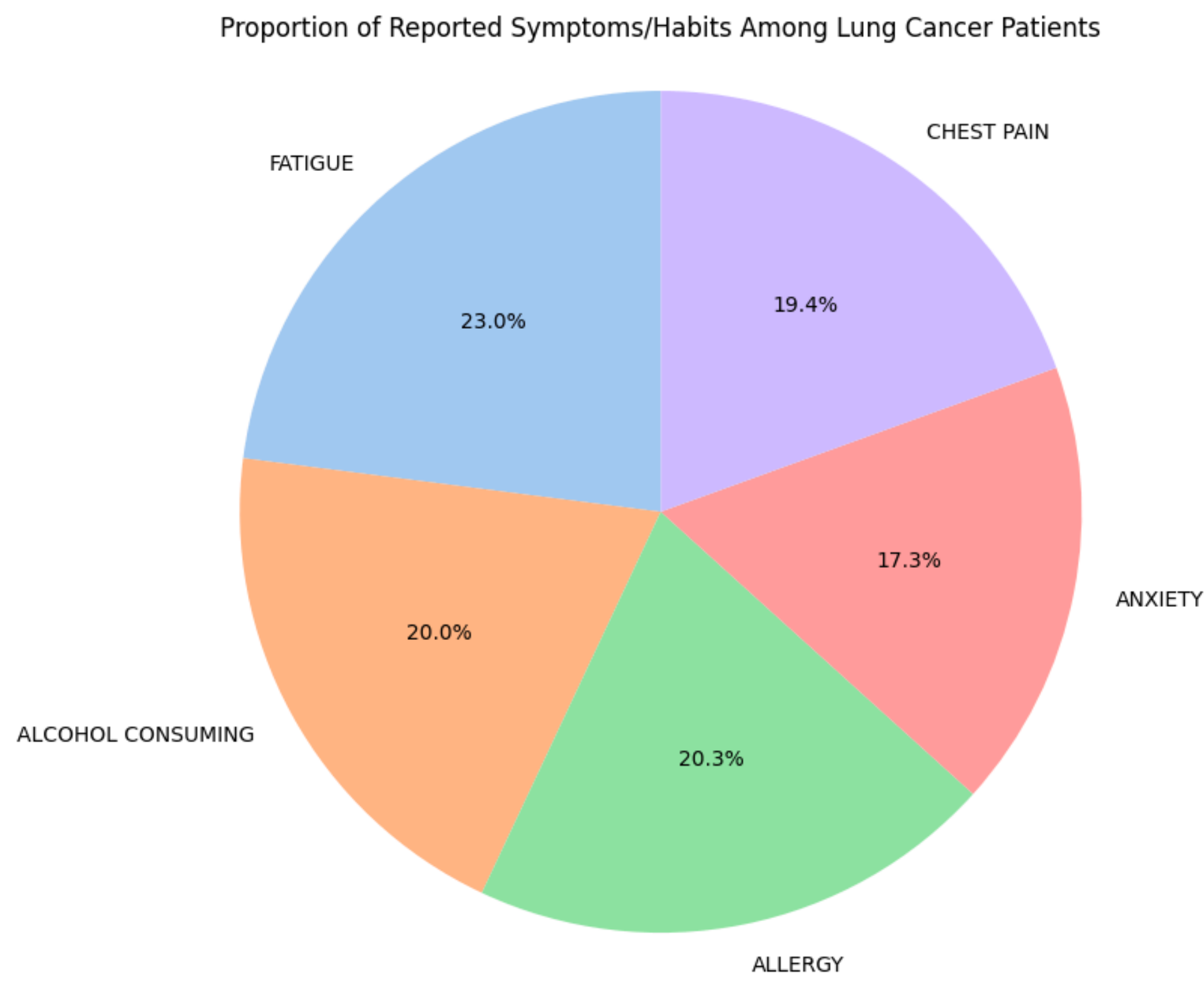
plt.title('Correlation Heatmap')



```
# 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)

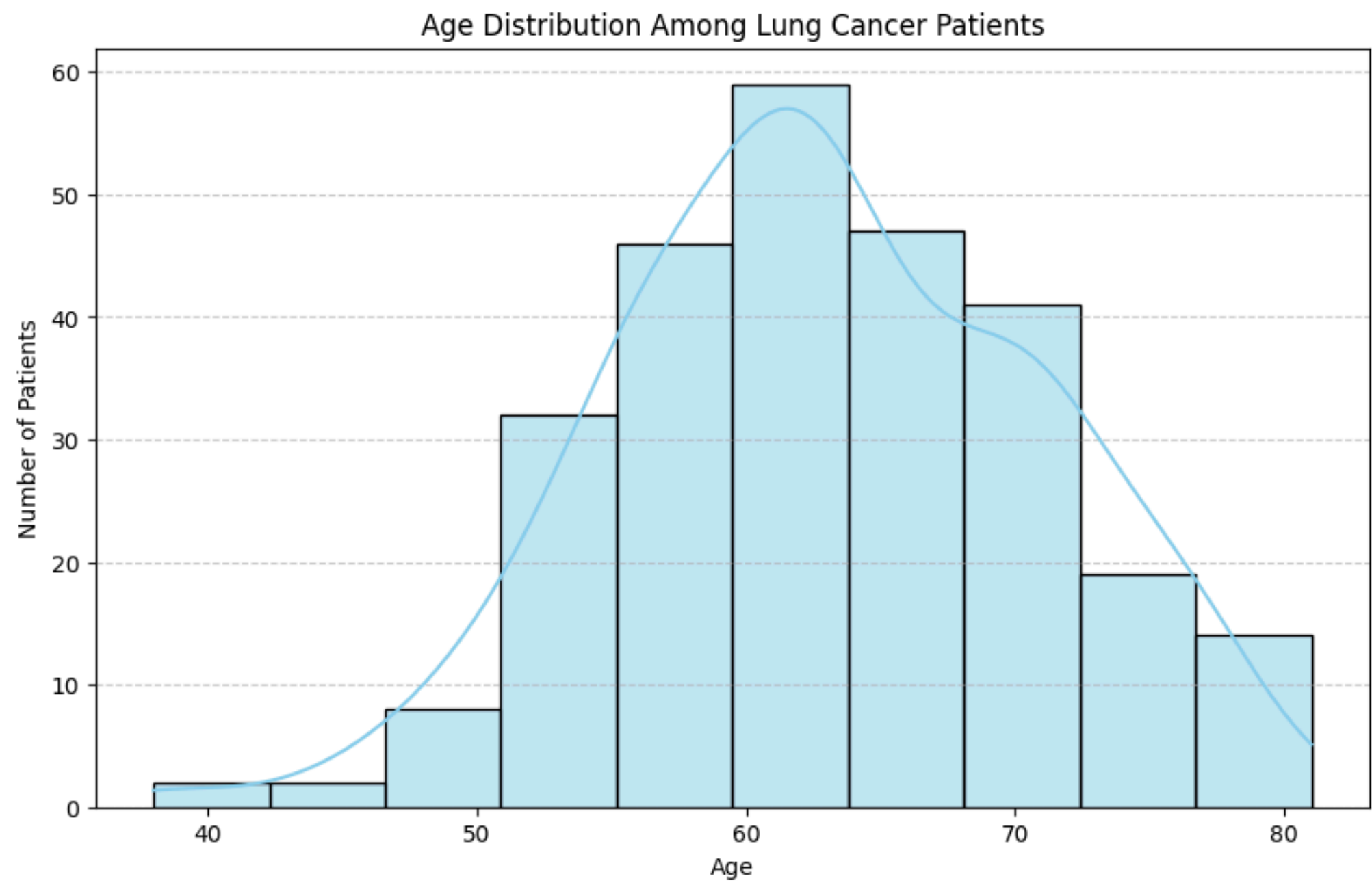
# Create the pie chart
plt.figure(figsize=(10, 8))
plt.pie(symptom_series, labels=symptom_series.index, autopct='%1.1f%%', startangle=90, colors=sns.color_palette('pastel'))
plt.title('Proportion of Reported Symptoms/Habits Among Lung Cancer Patients')
plt.axis('equal') # Equal aspect ratio ensures that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



```
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()
```



```
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()
```

