Lung Cancer Patient Data Analysis

In [2]:

```
# importing import libraries for data analysis
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

1. Read the Data

In [3]:

```
Lung_cancer = pd.read_excel(r"C:\Users\ratho\.ipynb_checkpoints\DATA\Lung Cancer Surve
Lung_cancer

# here we will read the data set
```

Out[3]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	AL
0	M	69	1	2	2	1	1	2	
1	M	74	2	1	1	1	2	2	
2	F	59	1	1	1	2	1	2	
3	M	63	2	2	2	1	1	1	
4	F	63	1	2	1	1	1	1	
...	...	...	...	...	...	...	...	...	
304	F	56	1	1	1	2	2	2	
305	M	70	2	1	1	1	1	2	
306	M	58	2	1	1	1	1	1	
307	M	67	2	1	2	1	1	2	
308	M	62	1	1	1	2	1	2	

309 rows × 16 columns

2. Data Quick Check

In [4]:

```
Lung_cancer.head() # top 5 rows rows
```

Out[4]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLE
0	M	69	1	2	2	1	1	2	
1	M	74	2	1	1	1	2	2	
2	F	59	1	1	1	2	1	2	
3	M	63	2	2	2	1	1	1	
4	F	63	1	2	1	1	1	1	

In [5]:

```
Lung_cancer.tail()  #bottom 5 rows
```

Out[5]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	AL
304	F	56	1	1	1	2	2	2	
305	M	70	2	1	1	1	1	2	
306	M	58	2	1	1	1	1	1	
307	M	67	2	1	2	1	1	2	
308	M	62	1	1	1	2	1	2	

In [6]:

```
Lung_cancer.shape  # checking the shape
```

Out[6]:

```
(309, 16)
```

In [7]:

```
Lung_cancer.info()   # summary of data set
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 309 entries, 0 to 308
Data columns (total 16 columns):
 #   Column                 Non-Null Count  Dtype 
---  ------                 --------------  ----- 
 0   GENDER                 309 non-null    object
 1   AGE                    309 non-null    int64 
 2   SMOKING                309 non-null    int64 
 3   YELLOW_FINGERS         309 non-null    int64 
 4   ANXIETY                309 non-null    int64 
 5   PEER_PRESSURE          309 non-null    int64 
 6   CHRONIC DISEASE        309 non-null    int64 
 7   FATIGUE                309 non-null    int64 
 8   ALLERGY                309 non-null    int64 
 9   WHEEZING               309 non-null    int64 
 10  ALCOHOL CONSUMING      309 non-null    int64 
 11  COUGHING               309 non-null    int64 
 12  SHORTNESS OF BREATH    309 non-null    int64 
 13  SWALLOWING DIFFICULTY  309 non-null    int64 
 14  CHEST PAIN             309 non-null    int64 
 15  LUNG_CANCER            309 non-null    object
dtypes: int64(14), object(2)
memory usage: 38.8+ KB
```

In [8]:

```
Lung_cancer.isnull()   # is ther any null values or not
```

Out[8]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	A
0	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	
304	False	False	False	False	False	False	False	False	
305	False	False	False	False	False	False	False	False	
306	False	False	False	False	False	False	False	False	
307	False	False	False	False	False	False	False	False	
308	False	False	False	False	False	False	False	False	

309 rows × 16 columns

In [9]:

```
Lung_cancer.isnull().sum()  # total null values
```

Out[9]:

```
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
```

3. Seperating Catgorical and numerical data

In [10]:

```
cat = Lung_cancer.select_dtypes(include = 'object').columns
# grouping columns having object datatypes in cat variable

num = Lung_cancer.select_dtypes(exclude ='object').columns
# grouping columns having numercal datatypes in num variable

cat,num
```

Out[10]:

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

4. categorical column analysis

In [11]:

```
cat # check cat data
```

Out[11]:

```
Index(['GENDER', 'LUNG_CANCER'], dtype='object')
```

In [12]:

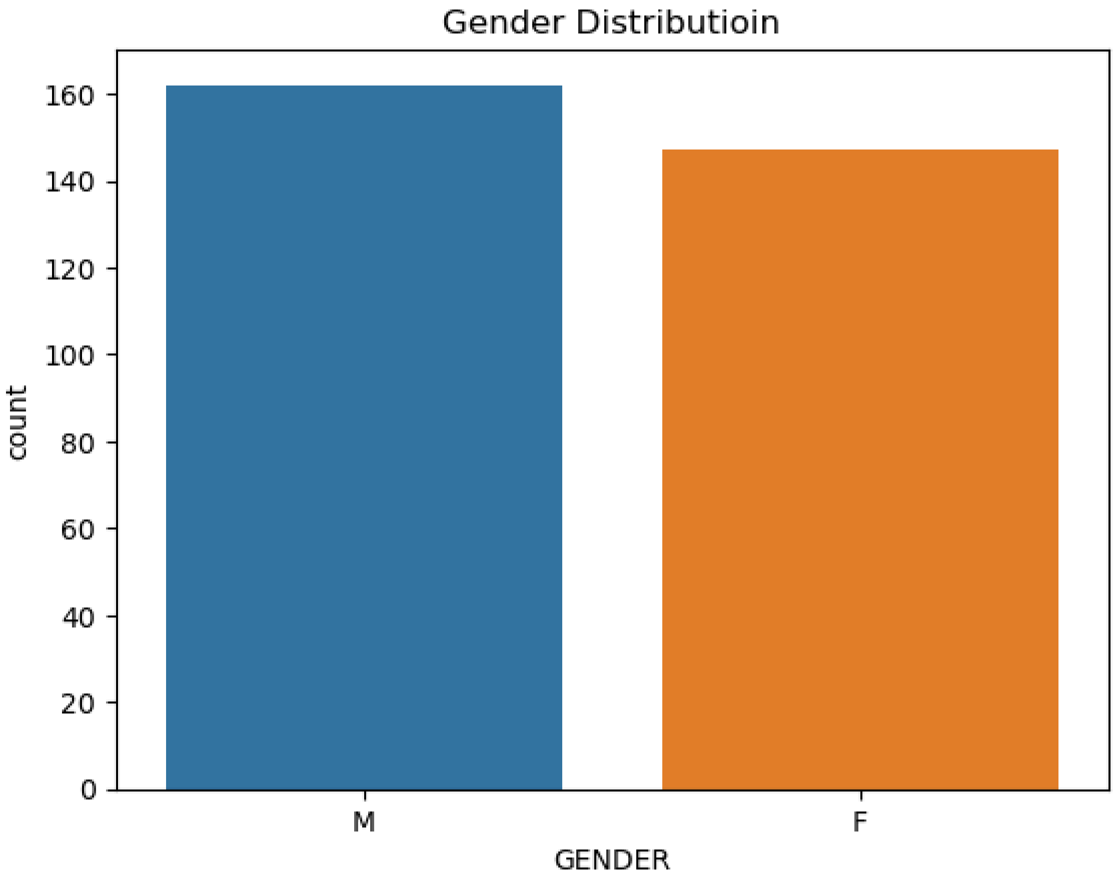
```
# frequency distribution of Gender column

gender_counts = Lung_cancer['GENDER'].value_counts()
print('Gender Frequency:\n', gender_counts)

# ploting the graph

sns.countplot(x = 'GENDER', data = Lung_cancer)
plt.title('Gender Distributioin')
plt.show()
```

```
Gender Frequency:
 M    162
F    147
Name: GENDER, dtype: int64
```



Observations:

1. Male (M) count is higher: The number of males in the dataset slightly exceeds the number of females.

2. Balanced distribution: Despite the difference, the gender distribution appears relatively balanced, with a small gap between the two groups.

Insights:

1. Gender inclusion: The dataset includes a fairly even representation of males and females, which is good for conducting gender-specific analyses.

2. Potential bias: If gender plays a significant role in lung cancer outcomes or characteristics, the slight imbalance should be taken into account during modeling or statistical testing.

3. Further analysis: Investigate if gender correlates with any specific patterns, such as the prevalence of lung cancer, severity, or treatment outcomes.

In [13]:

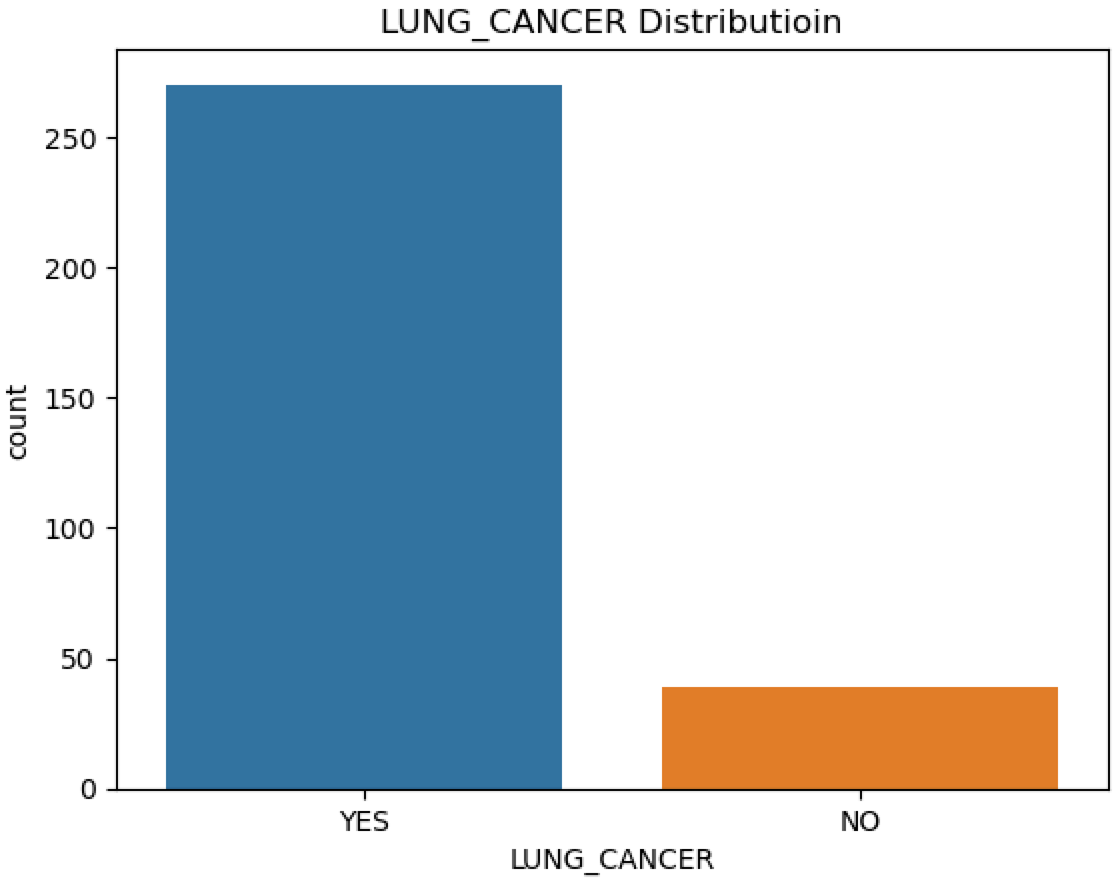
```python
# frequency distribution of Gender column

gender_counts = Lung_cancer['LUNG_CANCER'].value_counts()
print('LUNG_CANCER Frequency:\n', gender_counts)

# ploting the graph

sns.countplot(x = 'LUNG_CANCER', data = Lung_cancer)
plt.title('LUNG_CANCER Distributioin')
plt.show()
```

```
LUNG_CANCER Frequency:
 YES    270
NO      39
Name: LUNG_CANCER, dtype: int64
```



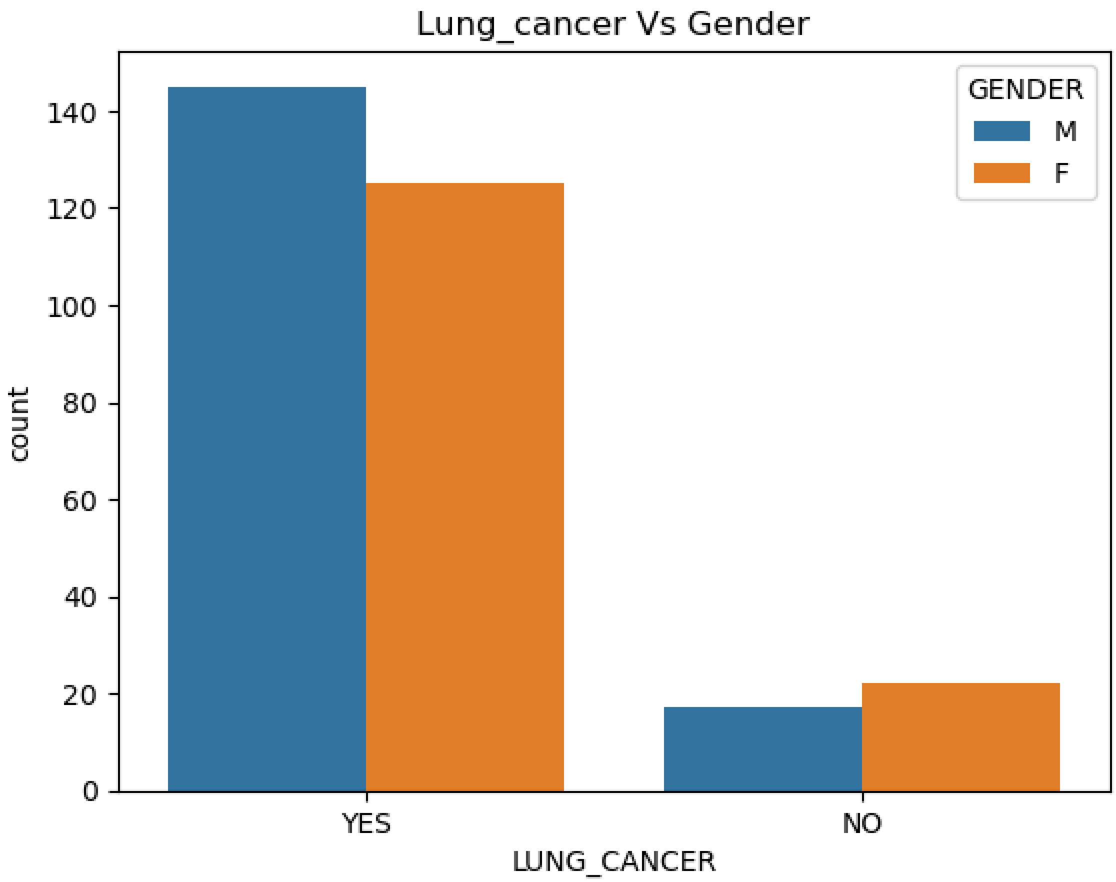
Observation:

The chart shows an imbalance in lung cancer distribution, with most cases in the `YES` category (over 250) and significantly fewer in the `NO` category (under 50). This suggests a strong skew towards positive cases.

In [14]:

```
# Gender Vs , LUNG_cancer freq

sns.countplot(x = 'LUNG_CANCER', hue = 'GENDER', data = Lung_cancer)
plt.title('Lung_cancer Vs Gender')
plt.show()
```



Observations:

Lung Cancer Diagnosis (YES): Males have the highest count of diagnoses. Females also have a high count but are slightly lower than males.

No Lung Cancer Diagnosis (NO): Both males and females have significantly lower counts compared to the YES category. Females slightly outnumber males in the NO category.

General Observation: The total number of individuals diagnosed with lung cancer (YES) is much greater than those not diagnosed (NO).

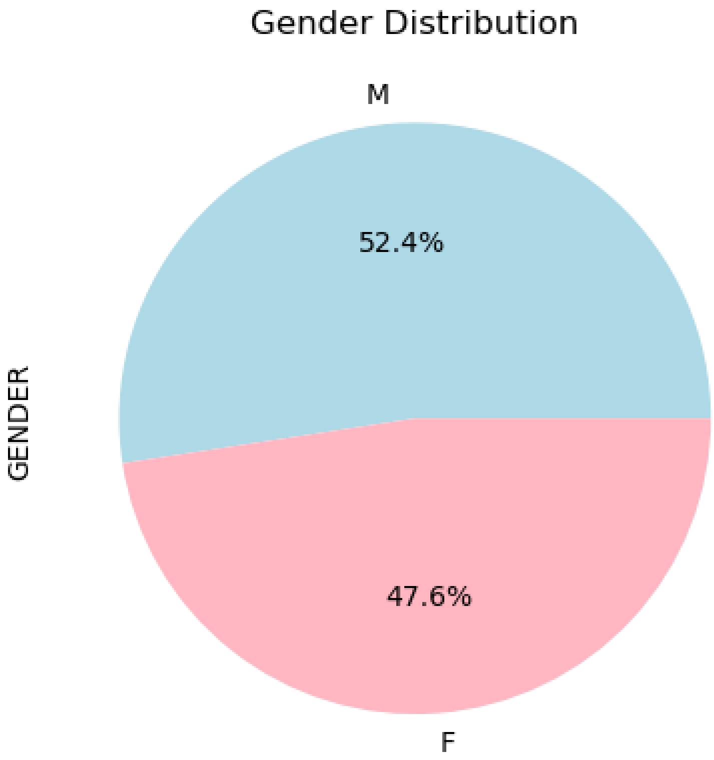
Insights:

- Males are more likely to be diagnosed with lung cancer compared to females, suggesting possible gender-specific risk factors.
- A majority of the population in the chart is diagnosed with lung cancer, which may indicate high exposure to risk factors (e.g., smoking, air pollution, etc.).
- Preventative measures, awareness campaigns, and gender-specific interventions should be prioritized, especially targeting male populations.

In [15]:

```
# pie chart for gender distribution

Lung_cancer['GENDER'].value_counts().plot.pie(autopct='%1.1f%%',colors = ['lightblue',
plt.title('Gender Distribution')
plt.show()
```



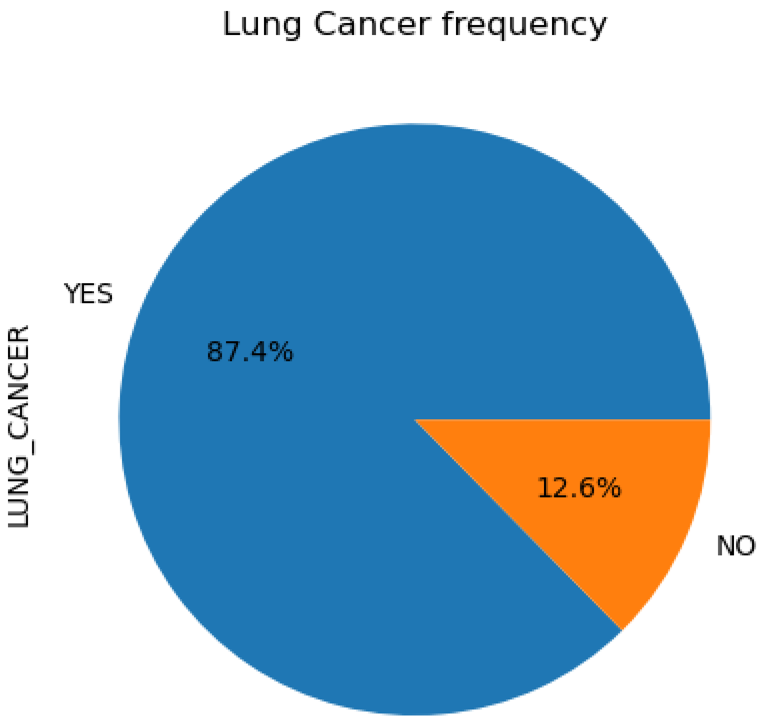
Observations:

- Males (52.4%) slightly outnumber females (47.6%).
- Gender distribution is nearly balanced.

Insights:

- The balanced gender split ensures fair comparison in lung cancer analysis.
- Slight male majority aligns with their higher lung cancer diagnosis rate.

```
In [16]: # pie chart for Lung_cancer freq
Lung_cancer['LUNG_CANCER'].value_counts().plot.pie(autopct = '%1.1f%%')
plt.title('Lung Cancer frequency')
plt.show()
```



Observations:

- **87.4%** of the population is diagnosed with lung cancer (`YES`).
- Only **12.6%** of the population is not diagnosed (`NO`).

Insights:

- Lung cancer diagnosis is highly prevalent in this dataset.
- Indicates a potential high-risk population or environmental/lifestyle factors at play.

In [17]:

```
Cancer_Gender = pd.crosstab(Lung_cancer['LUNG_CANCER'],Lung_cancer['GENDER'])
print('\n Cancer Vs Gender:\n',Cancer_Gender)
```

```
 Cancer Vs Gender:
 GENDER          F    M
LUNG_CANCER
NO             22   17
YES           125  145
```

5. numerical column analysis

In [18]:

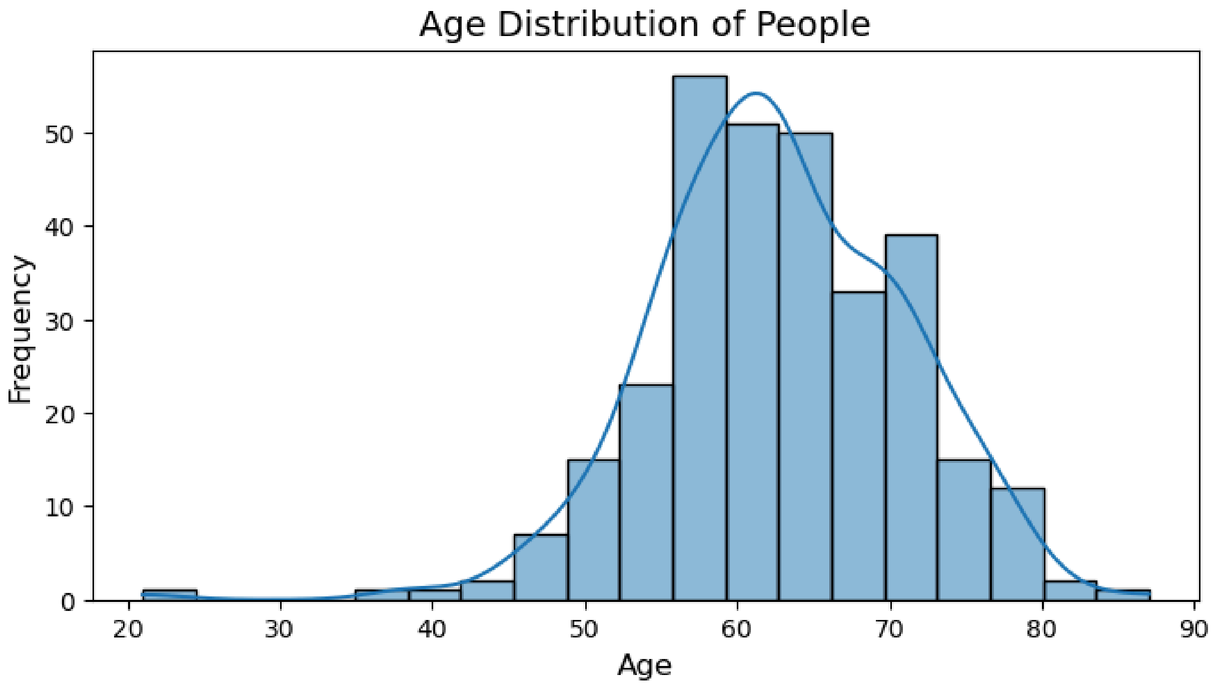
```
num
```

Out[18]:

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

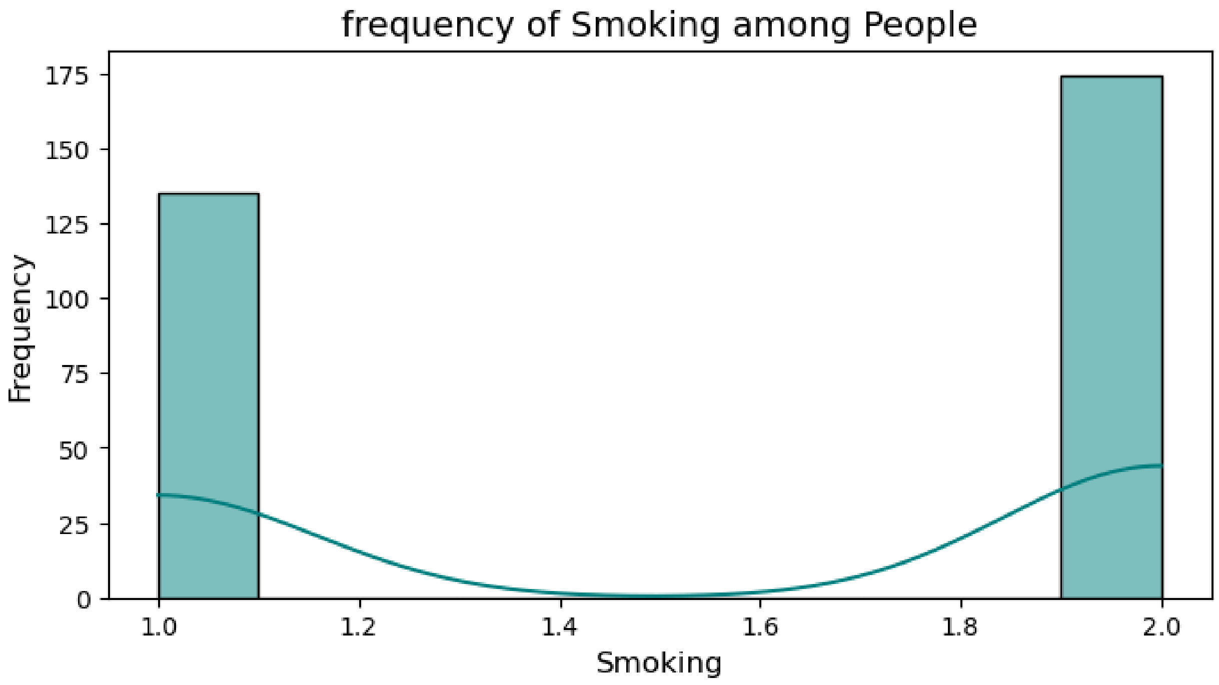
In [19]:

```
plt.figure(figsize= (8,4))
sns.histplot(Lung_cancer['AGE'],kde = True)
plt.title('Age Distribution of People',fontsize = 14)
plt.xlabel('Age',fontsize = 12)
plt.ylabel('Frequency',fontsize = 12)
plt.show()
```



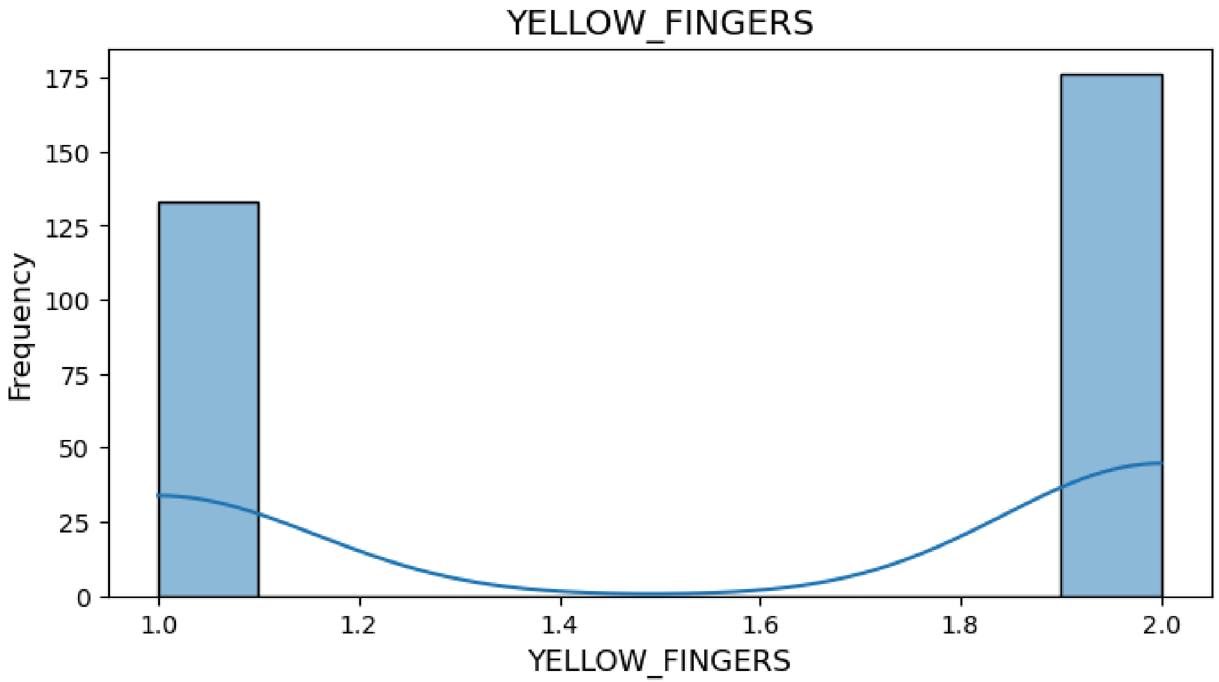
In [17]:

```
plt.figure(figsize= (8,4))
sns.histplot(Lung_cancer['SMOKING'],kde = True, bins = 10, color = 'teal')
plt.title('frequency of Smoking among People',fontsize = 14)
plt.xlabel('Smoking',fontsize = 12)
plt.ylabel('Frequency',fontsize = 12)
plt.show()
```



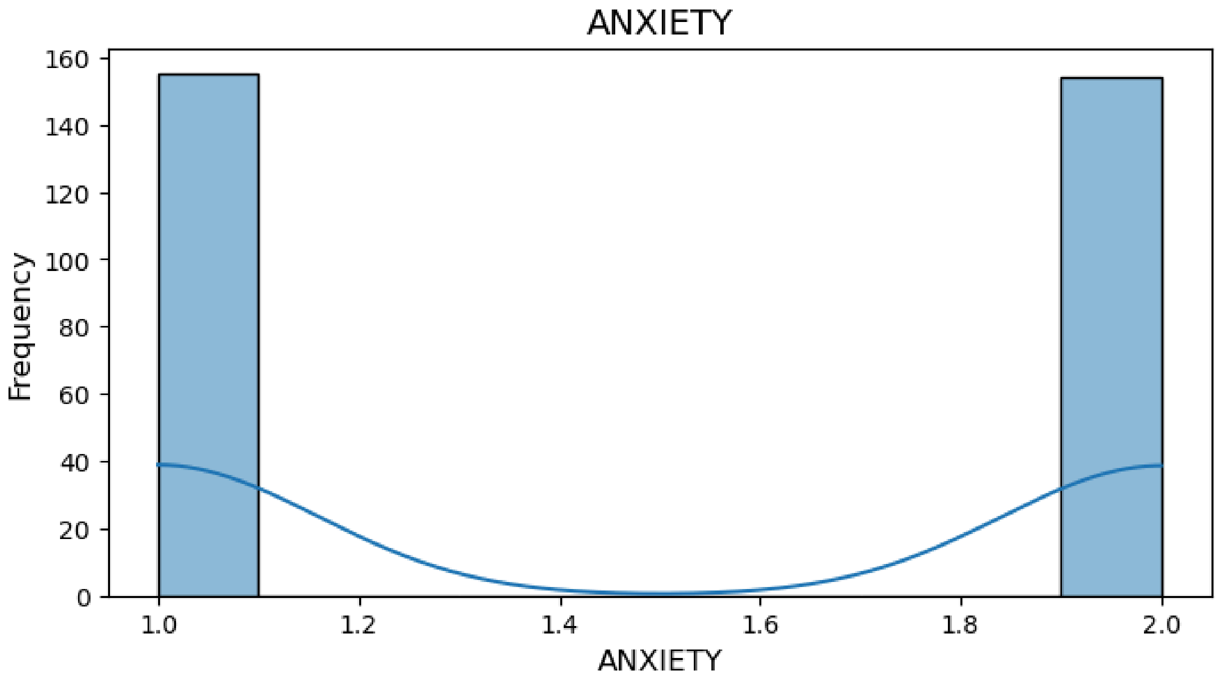
In [15]:

```
plt.figure(figsize= (8,4))
sns.histplot(Lung_cancer['YELLOW_FINGERS'],kde = True)
plt.title('YELLOW_FINGERS',fontsize = 14)
plt.xlabel('YELLOW_FINGERS',fontsize = 12)
plt.ylabel('Frequency',fontsize = 12)
plt.show()
```



In [22]:

```
plt.figure(figsize= (8,4))
sns.histplot(Lung_cancer['ANXIETY'],kde = True)
plt.title('ANXIETY',fontsize = 14)
plt.xlabel('ANXIETY',fontsize = 12)
plt.ylabel('Frequency',fontsize = 12)
plt.show()
```

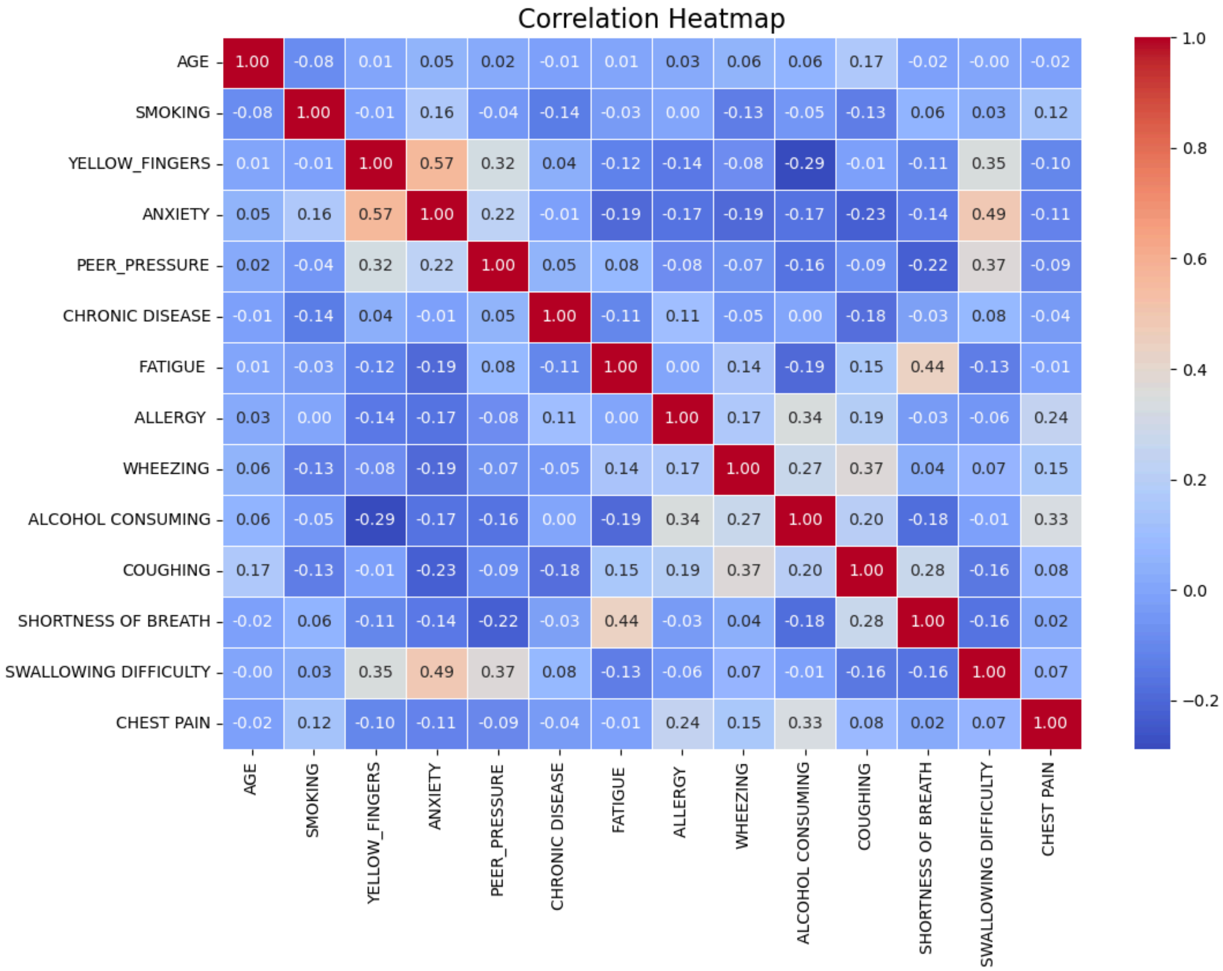


6. Corelation using heat map

In [23]:

```
correlation_matrix = Lung_cancer.corr()


plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)
plt.title('Correlation Heatmap', fontsize=16)
plt.show()
```



Insights:

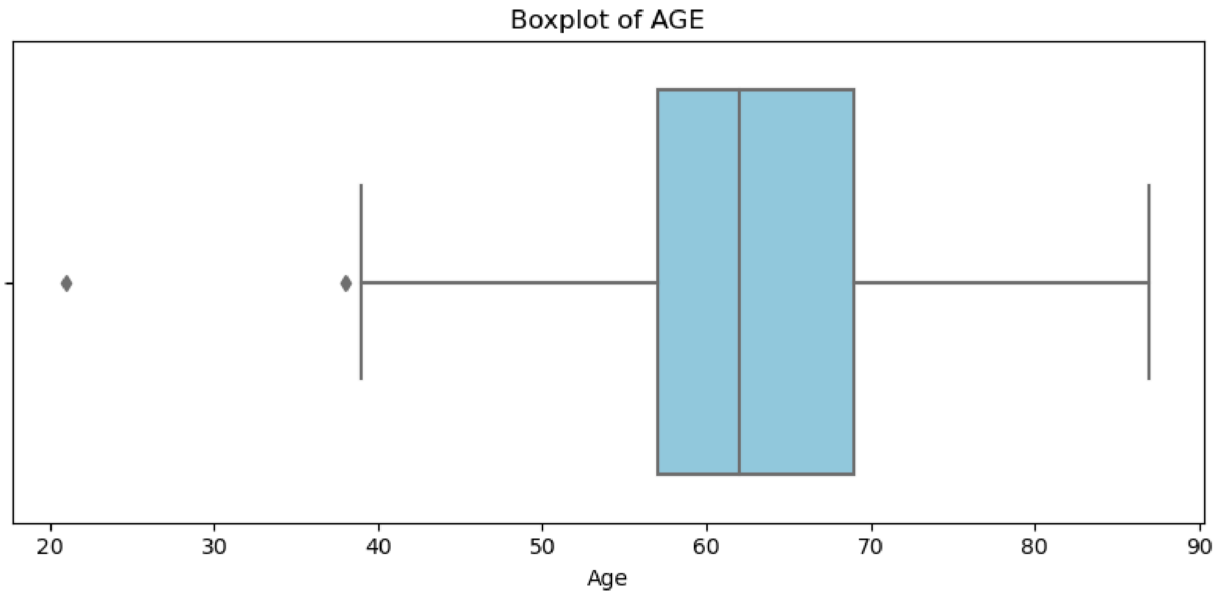
- The heatmap reveals complex relationships between various factors related to lung health and lifestyle.
- Smoking appears to have a strong influence on multiple factors, including anxiety, peer pressure, and yellow fingers.
- Anxiety and alcohol consumption seem to be linked to certain respiratory symptoms.
- Chronic disease might be protective against some respiratory symptoms.

7. Checking outliers

In [24]:

```
# Create boxplots for selected numerical column

plt.figure(figsize=(8, 4))
sns.boxplot(x=Lung_cancer['AGE'], color='skyblue')
plt.title(f'Boxplot of AGE', fontsize=12)
plt.xlabel('Age', fontsize=10)
plt.tight_layout()
plt.show()
```



The boxplot of Age shows a right-skewed distribution with a median age of 60 years. The IQR is 10 years, and there are two outliers representing younger individuals. This indicates a dataset primarily consisting of middle-aged individuals with a significant presence of younger individuals.

In [25]:

```
import pandas as pd

# assuming num is athr list of numerica column names in the dataframe



# loop through all numerical columns and remove outliers using IQR

# calculate Q1(25 th percentile) and Q3 (75th percentile)
Q1 = Lung_cancer['AGE'].quantile(0.25)
Q3 = Lung_cancer['AGE'].quantile(0.75)

# Calculate IQR (interquartile range)
IQR = Q3 - Q1

#Define outer bounds
lower_bound = Q1 -1.5 * IQR
upper_bound = Q3 +1.5 * IQR

# Remove rows where the column value is an outlier
```

```
Lung_cancer = Lung_cancer[(Lung_cancer['AGE'] >= lower_bound) & (Lung_cancer['AGE'] <=

#verify the data aafter removing thr outliers
Lung_cancer.head()
```

Out[25]:

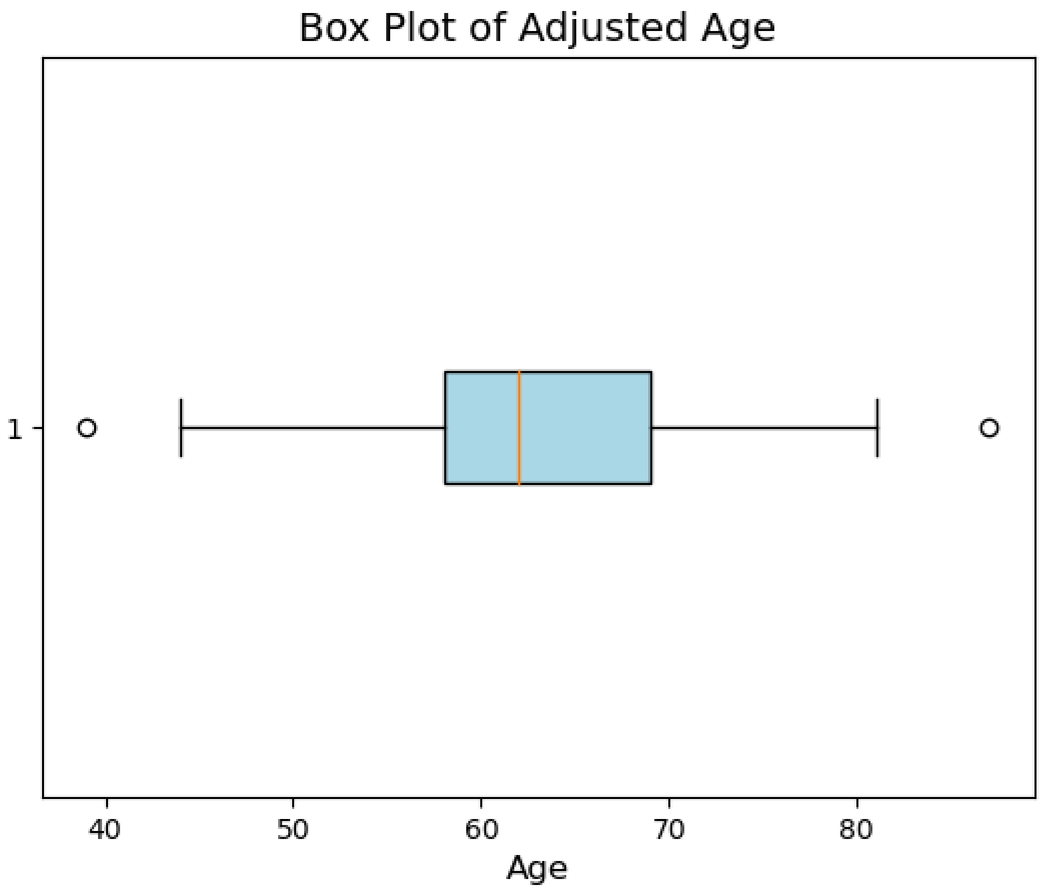
	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLE
0	M	69	1	2	2	1	1	2	
1	M	74	2	1	1	1	2	2	
2	F	59	1	1	1	2	1	2	
3	M	63	2	2	2	1	1	1	
4	F	63	1	2	1	1	1	1	

In [26]:

```
# Create a horizontal box plot for the 'p_wage_1' column
plt.boxplot(Lung_cancer['AGE'], vert=False, patch_artist=True,
            boxprops=dict(facecolor='lightblue'))

# Add title and labels for better clarity
plt.title('Box Plot of Adjusted Age', fontsize=14)
plt.xlabel('Age', fontsize=12)

# Display the box plot
plt.show()
```



Convert cateorical to numerical

In [28]:
```
cat
```

Out[28]:
```
Index(['GENDER', 'LUNG_CANCER'], dtype='object')
```

In [29]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
for column in cat:
    Lung_cancer[column] = le.fit_transform(Lung_cancer[column])
(Lung_cancer)
```

Out[29]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	AL
0	1	69	1	2	2	1	1	2	
1	1	74	2	1	1	1	2	2	
2	0	59	1	1	1	2	1	2	
3	1	63	2	2	2	1	1	1	
4	0	63	1	2	1	1	1	1	
...	...	...	...	...	...	...	...	...	
304	0	56	1	1	1	2	2	2	
305	1	70	2	1	1	1	1	2	
306	1	58	2	1	1	1	1	1	
307	1	67	2	1	2	1	1	2	
308	1	62	1	1	1	2	1	2	

307 rows × 16 columns

Scaling the Data

In [31]:
```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
Lung_cancer_scaled = pd.DataFrame(scaler.fit_transform(Lung_cancer), columns=Lung_canc
```

In [32]:
```
Lung_cancer_scaled
```

Out[32]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE
0	1.0	0.625000	0.0	1.0	1.0	0.0	0.0	1.0
1	1.0	0.729167	1.0	0.0	0.0	0.0	1.0	1.0
2	0.0	0.416667	0.0	0.0	0.0	1.0	0.0	1.0
3	1.0	0.500000	1.0	1.0	1.0	0.0	0.0	0.0
4	0.0	0.500000	0.0	1.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...
302	0.0	0.354167	0.0	0.0	0.0	1.0	1.0	1.0
303	1.0	0.645833	1.0	0.0	0.0	0.0	0.0	1.0
304	1.0	0.395833	1.0	0.0	0.0	0.0	0.0	0.0
305	1.0	0.583333	1.0	0.0	1.0	0.0	0.0	1.0
306	1.0	0.479167	0.0	0.0	0.0	1.0	0.0	1.0

307 rows × 16 columns