eda-lung-cancer

August 14, 2024

1 EDA on Lung Cancer Dataset

Do the EDA on the given dataset: Lung cancer, and extract some useful information from this. Dataset Description: Lung cancer is one of the most prevalent and deadly forms of cancer worldwide, presenting significant challenges in early detection and effective treatment. To aid in the global effort to understand and combat this disease, we are excited to introduce our comprehensive Lung Cancer Dataset.

[2]:
```
#import libraries
import pandas as pd
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# read the dataset
lung_cancer_df = pd.read_csv(r"D:
 ↪\PWskills\assign\solutions\projects-assignment\EDA␣
 ↪assginmets\lungcancer_dataset.csv")

# Display the first few rows to get an overview
lung_cancer_df.head(10)
```

[2]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	\
0	M	65	1	1	1	2	
1	F	55	1	2	2	1	
2	F	78	2	2	1	1	
3	M	60	2	1	1	1	
4	F	80	1	1	2	1	
5	F	58	1	1	1	2	
6	F	70	1	1	1	2	
7	F	74	2	2	1	1	
8	M	77	1	2	1	2	
9	F	67	2	2	2	2	

	CHRONIC_DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL_CONSUMING	COUGHING	\
0	2	1	2	2	2	2	
1	1	2	2	2	1	1	

```
2                1        2        1         2                  1         1
3                2        1        2         1                  1         2
4                1        2        1         2                  1         1
5                2        2        2         1                  2         2
6                2        1        2         2                  2         2
7                1        1        2         1                  1         1
8                1        1        1         1                  2         1
9                1        2        2         1                  2         1

   SHORTNESS_OF_BREATH  SWALLOWING_DIFFICULTY  CHEST_PAIN LUNG_CANCER
0                    2                      2           1          NO
1                    1                      2           2          NO
2                    2                      1           1         YES
3                    1                      2           2         YES
4                    1                      1           2          NO
5                    1                      1           2         YES
6                    2                      2           1         YES
7                    1                      2           1          NO
8                    1                      1           2          NO
9                    2                      1           1          NO
```

[3]:
```
lung_cancer_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 16 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   GENDER                 3000 non-null   object
 1   AGE                    3000 non-null   int64
 2   SMOKING                3000 non-null   int64
 3   YELLOW_FINGERS         3000 non-null   int64
 4   ANXIETY                3000 non-null   int64
 5   PEER_PRESSURE          3000 non-null   int64
 6   CHRONIC_DISEASE        3000 non-null   int64
 7   FATIGUE                3000 non-null   int64
 8   ALLERGY                3000 non-null   int64
 9   WHEEZING               3000 non-null   int64
 10  ALCOHOL_CONSUMING      3000 non-null   int64
 11  COUGHING               3000 non-null   int64
 12  SHORTNESS_OF_BREATH    3000 non-null   int64
 13  SWALLOWING_DIFFICULTY  3000 non-null   int64
 14  CHEST_PAIN             3000 non-null   int64
 15  LUNG_CANCER            3000 non-null   object
dtypes: int64(14), object(2)
memory usage: 375.1+ KB
```

the dataset has two of the columns are object types (categorical variables), rest are all integer types

containing values 1 and 2, where 1 = No and 2 = Yes

[4]:
```
lung_cancer_df.shape
```

[4]:
```
(3000, 16)
```

The data set has 3000 Rows and 16 Columns

[5]:
```
lung_cancer_df.isnull().sum()
```

[5]:
```
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
```

the dataset has no missing values, thats good for us

[6]:
```
# First, we will change the data types of the categorical variables
lung_cancer_df['LUNG_CANCER'] = lung_cancer_df['LUNG_CANCER'].
 ↪factorize(['NO','YES'])[0]
lung_cancer_df['GENDER'] = lung_cancer_df['GENDER'].factorize(['NO','YES'])[0]
# Male = 1 Female = 0
# Yes = 1 No = 0
```

[7]:
```
lung_cancer_df.head()
```

[7]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	\
0	1	65	1	1	1	2	
1	0	55	1	2	2	1	
2	0	78	2	2	1	1	
3	1	60	2	1	1	1	
4	0	80	1	1	2	1	

	CHRONIC_DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL_CONSUMING	COUGHING	\
0	2	1	2	2	2	2	

```
1                1        2        2        2                 1         1
2                1        2        1        2                 1         1
3                2        1        2        1                 1         2
4                1        2        1        2                 1         1

   SHORTNESS_OF_BREATH  SWALLOWING_DIFFICULTY  CHEST_PAIN  LUNG_CANCER
0                    2                      2           1            0
1                    1                      2           2            0
2                    2                      1           1            1
3                    1                      2           2            1
4                    1                      1           2            0
```

[34]:
```
lung_cancer_df.columns.unique
```

[34]:
```
<bound method Index.unique of 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')>
```

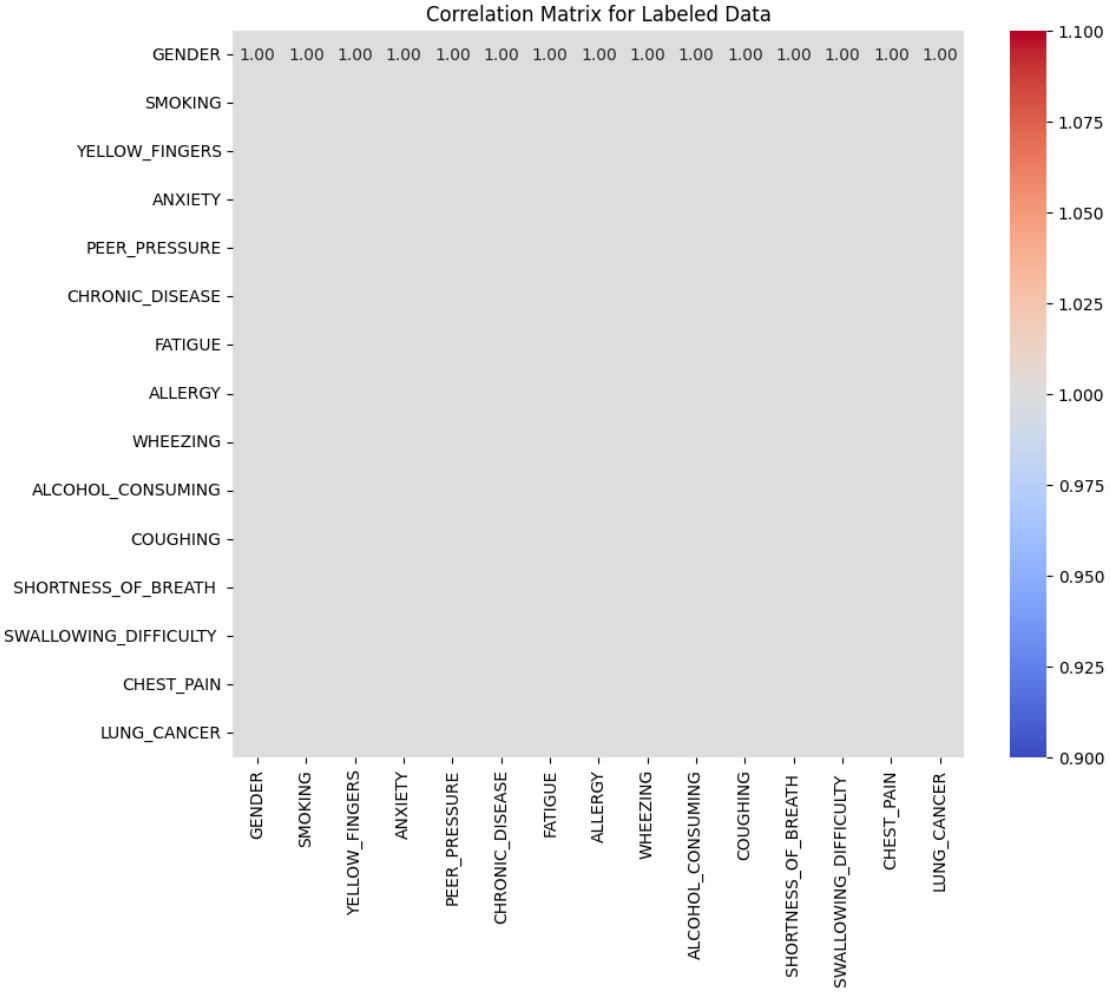
[35]:
```
unique_values = lung_cancer_df.nunique()

# Convert the result to a DataFrame for better readability
unique_values_df = unique_values.reset_index()
unique_values_df.columns = ['Feature', 'Unique Components']

# Display the DataFrame
print(unique_values_df)
```

```
                  Feature  Unique Components
0                  GENDER                  2
1                     AGE                 51
2                  SMOKING                  2
3           YELLOW_FINGERS                  2
4                 ANXIETY                  2
5           PEER_PRESSURE                  2
6          CHRONIC_DISEASE                  2
7                 FATIGUE                  2
8                 ALLERGY                  2
9                WHEEZING                  2
10      ALCOHOL_CONSUMING                  2
11               COUGHING                  2
12    SHORTNESS_OF_BREATH                  2
13  SWALLOWING_DIFFICULTY                  2
14             CHEST_PAIN                  2
15            LUNG_CANCER                  2
```

2 Exploratory Data Analysis

[40]:
```
import seaborn as sns
import matplotlib.pyplot as plt


# Select only numeric columns
numeric_df = lung_cancer_df.select_dtypes(include=['number'])

# Calculate the correlation matrix
correlation_matrix = numeric_df.corr()

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


lung_cancer_df = pd.DataFrame({
    'GENDER': [0,1],
    'SMOKING': [1,2],
    'YELLOW_FINGERS': [1,2],
    'ANXIETY': [1,2],
    'PEER_PRESSURE': [1,2],
    'CHRONIC_DISEASE': [1,2],
    'FATIGUE': [1,2],
    'ALLERGY': [1,2],
    'WHEEZING': [1,2],
    'ALCOHOL_CONSUMING': [1,2],
    'COUGHING': [1,2],
    'SHORTNESS_OF_BREATH ': [1,2],
    'SWALLOWING_DIFFICULTY ': [1,2],
    'CHEST_PAIN': [1,2],
    'LUNG_CANCER': [0,1]
})

# Select only numeric columns including the label
numeric_df = lung_cancer_df.select_dtypes(include=['number'])

# Calculate the correlation matrix
correlation_matrix = numeric_df.corr()

# Create a heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')

# Add title
plt.title('Correlation Matrix for Labeled Data')
```

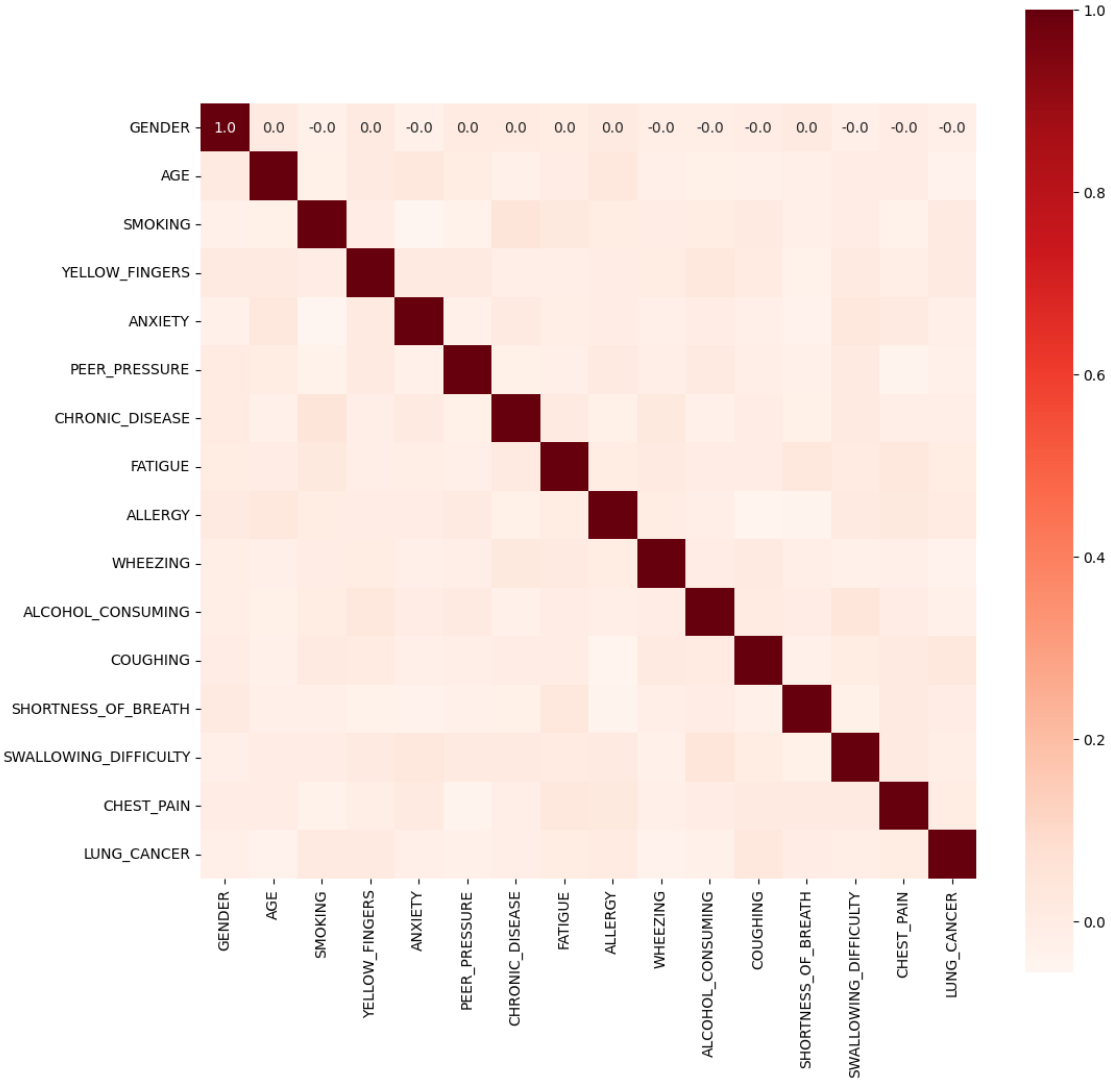
```
# Show the plot
plt.show()
```



As there are no enough numerical data in the dataset, we couldn't use correlation. After using labelencoded numeircals there is no strong corelation.

[8]:
```
# Lets first plot heatmap to check correlation between the variables.

corr = lung_cancer_df.corr()
plt.figure(figsize=(12,12))
sns.heatmap(corr,cbar=True,square=True,fmt='.1f',annot=True,cmap='Reds')
```

[8]: <Axes: >



There aren't many strong correlations among the features, except for a few minor ones:

- More anxiety leads to more difficulty breathing.
- Anxiety and yellow fingers are related, possibly in both directions.
- Shortness of breath causes fatigue.
- Increased anxiety makes swallowing harder.

As there are

However, there isn't a strong correlation between the symptoms and the target variable, possibly due to the data type or survey methods.

[9]:
```
# Now we will rename the 1s and 2s to No and Yes so it is easier to understand
```

```
lung_cancer_df["GENDER"] = lung_cancer_df["GENDER"].replace({1:"Male", 0:
 ↪"Female"})
lung_cancer_df["LUNG_CANCER"] = lung_cancer_df["LUNG_CANCER"].replace({1:"Yes",␣
 ↪0:"No"})
for column in lung_cancer_df.columns:
    lung_cancer_df[column] = lung_cancer_df[column].replace({1: "No", 2: "Yes"})

lung_cancer_df.head()
```

[9]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC_DISEASE	\
0	Male	65	No	No	No	Yes	Yes	
1	Female	55	No	Yes	Yes	No	No	
2	Female	78	Yes	Yes	No	No	No	
3	Male	60	Yes	No	No	No	Yes	
4	Female	80	No	No	Yes	No	No	

	FATIGUE	ALLERGY	WHEEZING	ALCOHOL_CONSUMING	COUGHING	SHORTNESS_OF_BREATH	\
0	No	Yes	Yes	Yes	Yes	Yes	
1	Yes	Yes	Yes	No	No	No	
2	Yes	No	Yes	No	No	Yes	
3	No	Yes	No	No	Yes	No	
4	Yes	No	Yes	No	No	No	

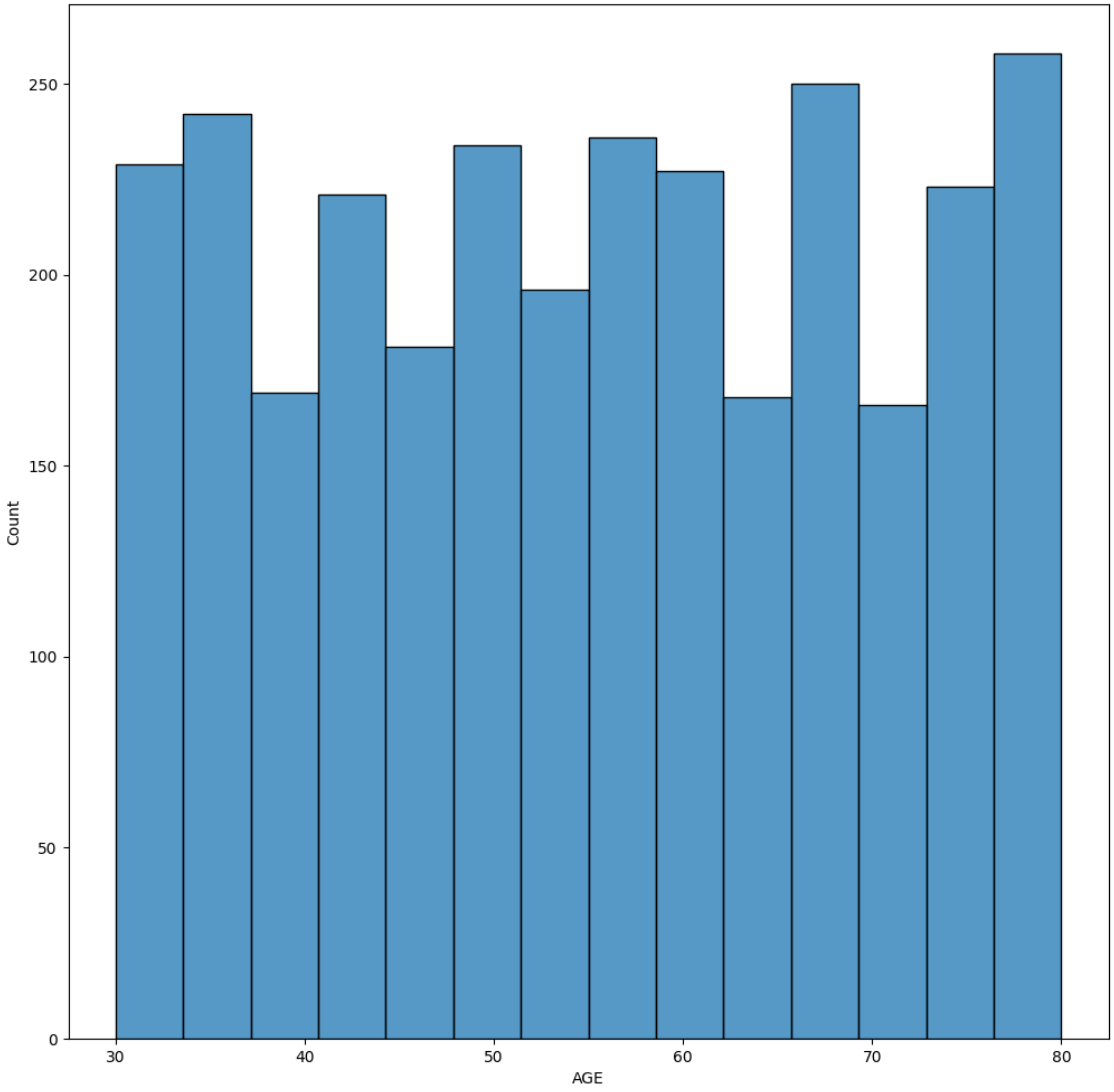
	SWALLOWING_DIFFICULTY	CHEST_PAIN	LUNG_CANCER
0	Yes	No	No
1	Yes	Yes	No
2	No	No	Yes
3	Yes	Yes	Yes
4	No	Yes	No

3 Age Distribution :

[10]:
```
plt.figure(figsize = (12,12))
sns.histplot(lung_cancer_df['AGE'])
```

```
C:\Users\ravit\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n
2kfra8p0\LocalCache\local-packages\Python311\site-
packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is
deprecated and will be removed in a future version. Convert inf values to NaN
before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

[10]: <Axes: xlabel='AGE', ylabel='Count'>

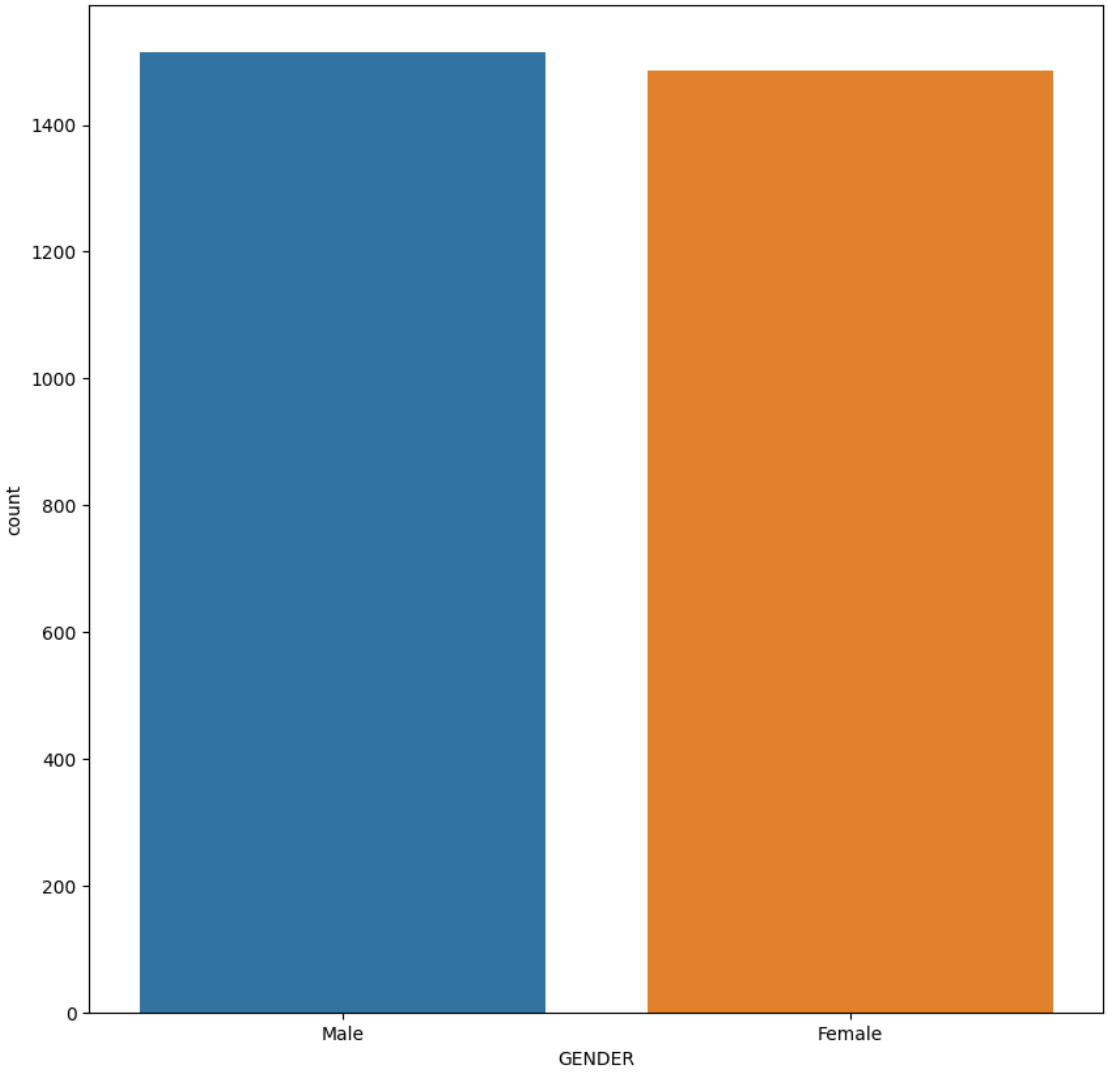


We can see that the data is distributed in discrete of age, we can see that the people of age 30 and 80 has no such significant difference, and other age categories has also no significant difference.

4 Gender Distribution:

[11]:
```
plt.figure(figsize=(10,10))
sns.countplot(x="GENDER", data=lung_cancer_df)
```

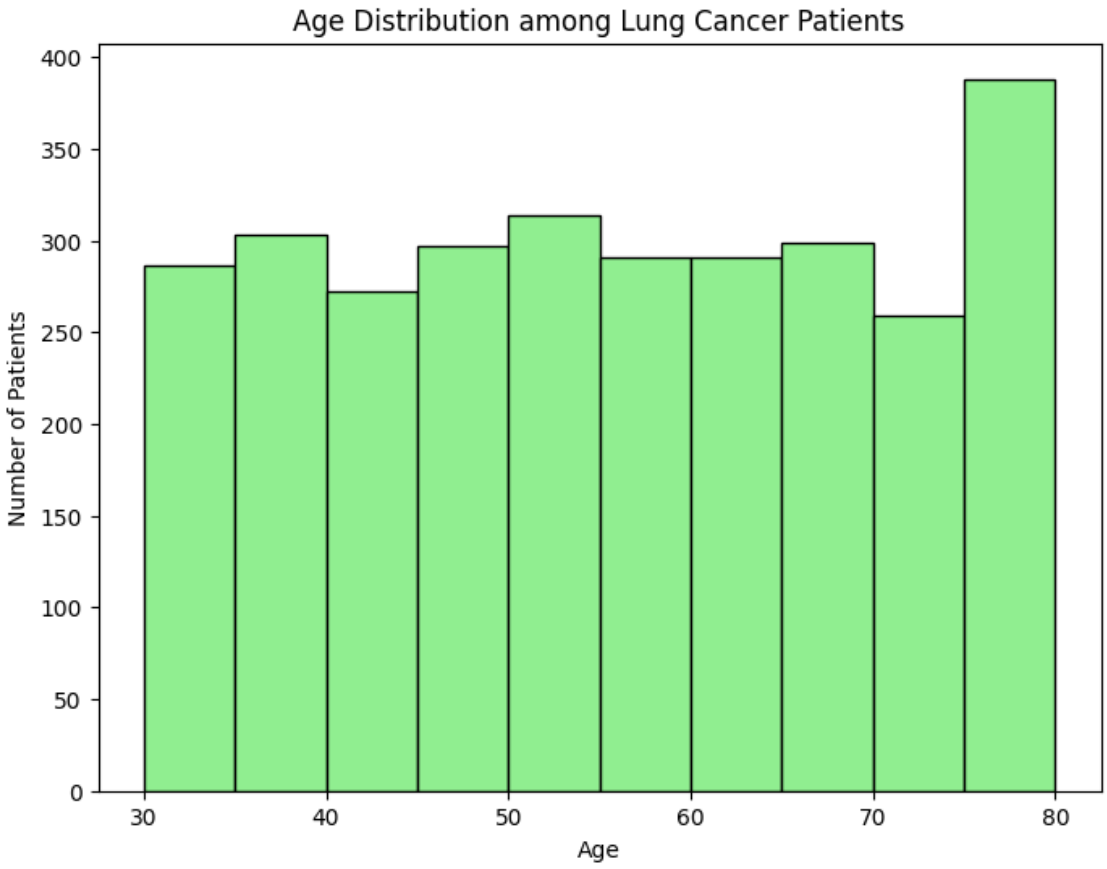
[11]: <Axes: xlabel='GENDER', ylabel='count'>



In the dataset the difference between number of males and females is not too much.

5 Age Distribution with People having Lung Cancer

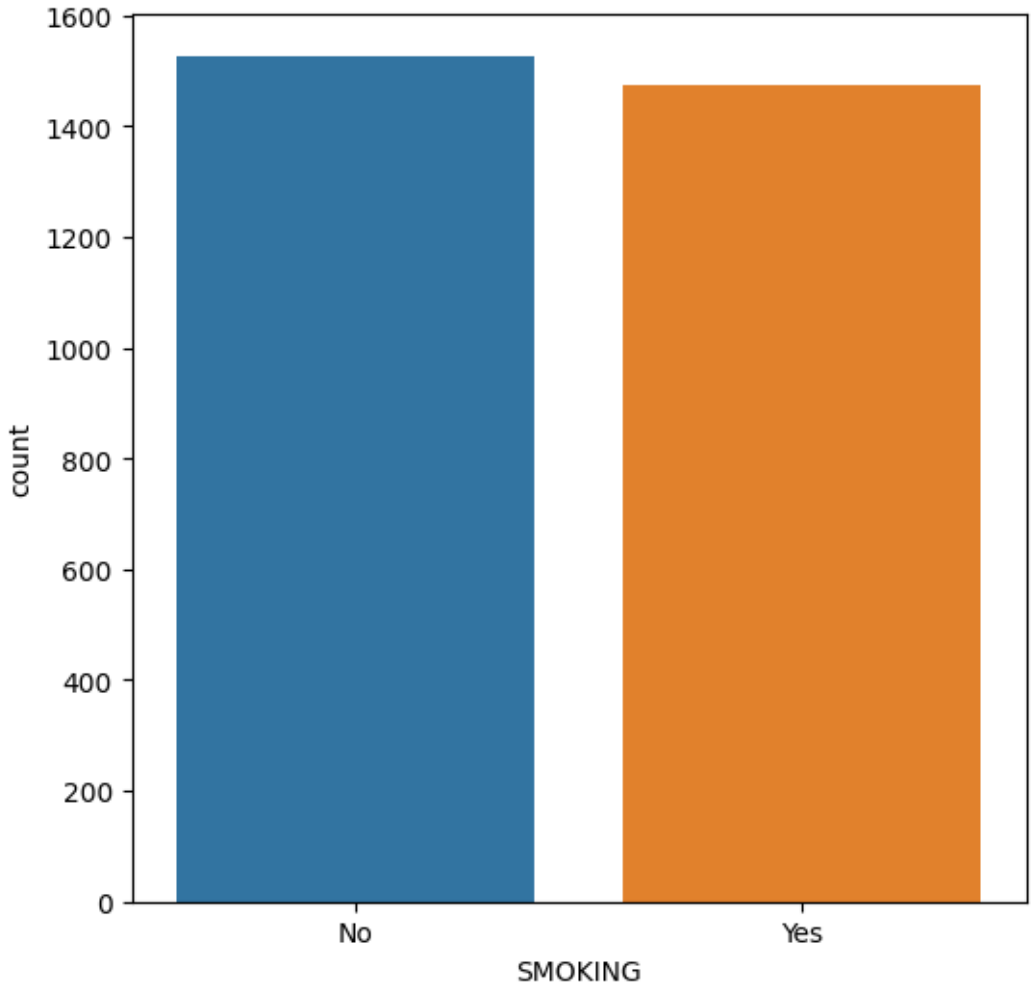
[12]:
```
ages = lung_cancer_df['AGE']

# Create a histogram
plt.figure(figsize=(8, 6))
plt.hist(ages, bins=10, color='lightgreen', edgecolor='black')
plt.xlabel('Age')
plt.ylabel('Number of Patients')
plt.title('Age Distribution among Lung Cancer Patients')
plt.show()
```



6 Smokers count

[13]:
```
plt.figure(figsize=(6,6))
sns.countplot(x="SMOKING", data=lung_cancer_df)
```

[13]:
```
<Axes: xlabel='SMOKING', ylabel='count'>
```

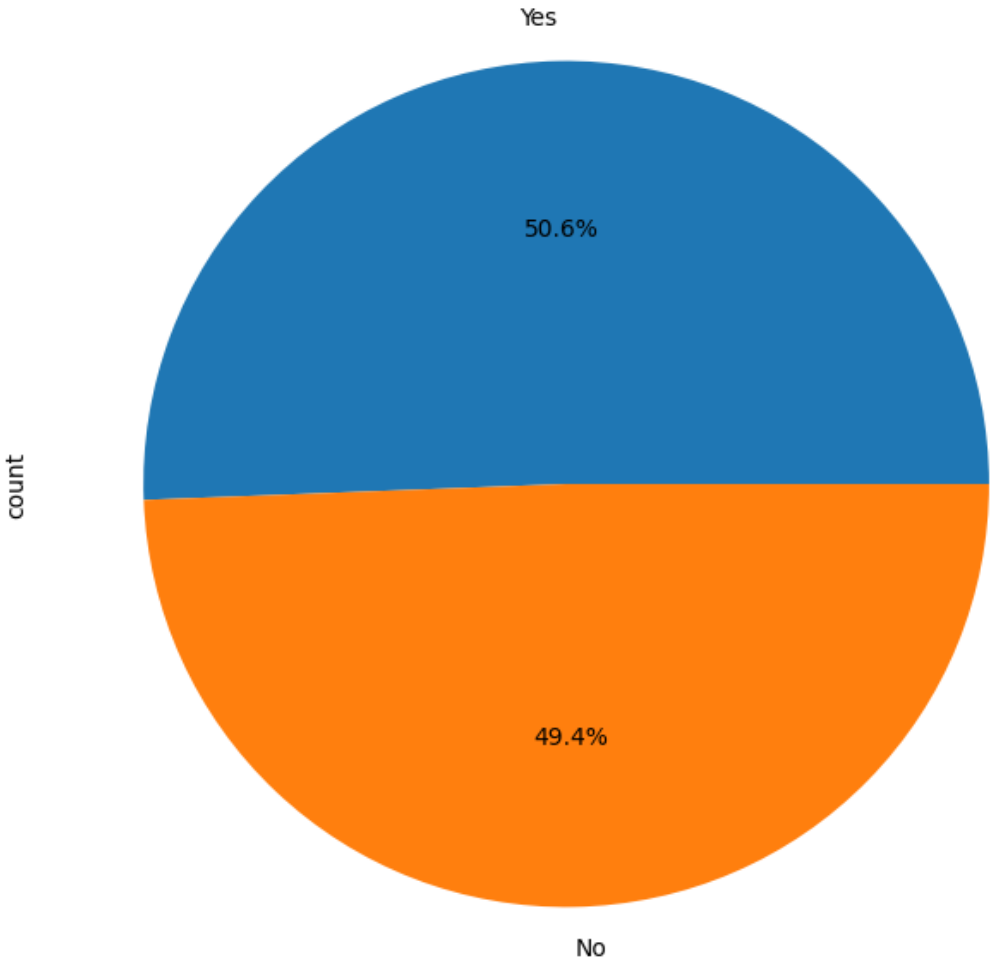


There are more smokers in the dataset than non-smokers

7 Percentage of people with lung cancer in the dataset

[14]:
```
# cancer_data.LUNG_CANCER.value_counts() / len(cancer_data.LUNG_CANCER)

lung_cancer_df.LUNG_CANCER.value_counts().plot(kind='pie',figsize=(8,␣
↪8),autopct='%1.1f%%')
```

[14]:
```
<Axes: ylabel='count'>
```



[15]:
```
lung_cancer_df.LUNG_CANCER.value_counts() *10/ len(lung_cancer_df.
↪LUNG_CANCER)*10
```

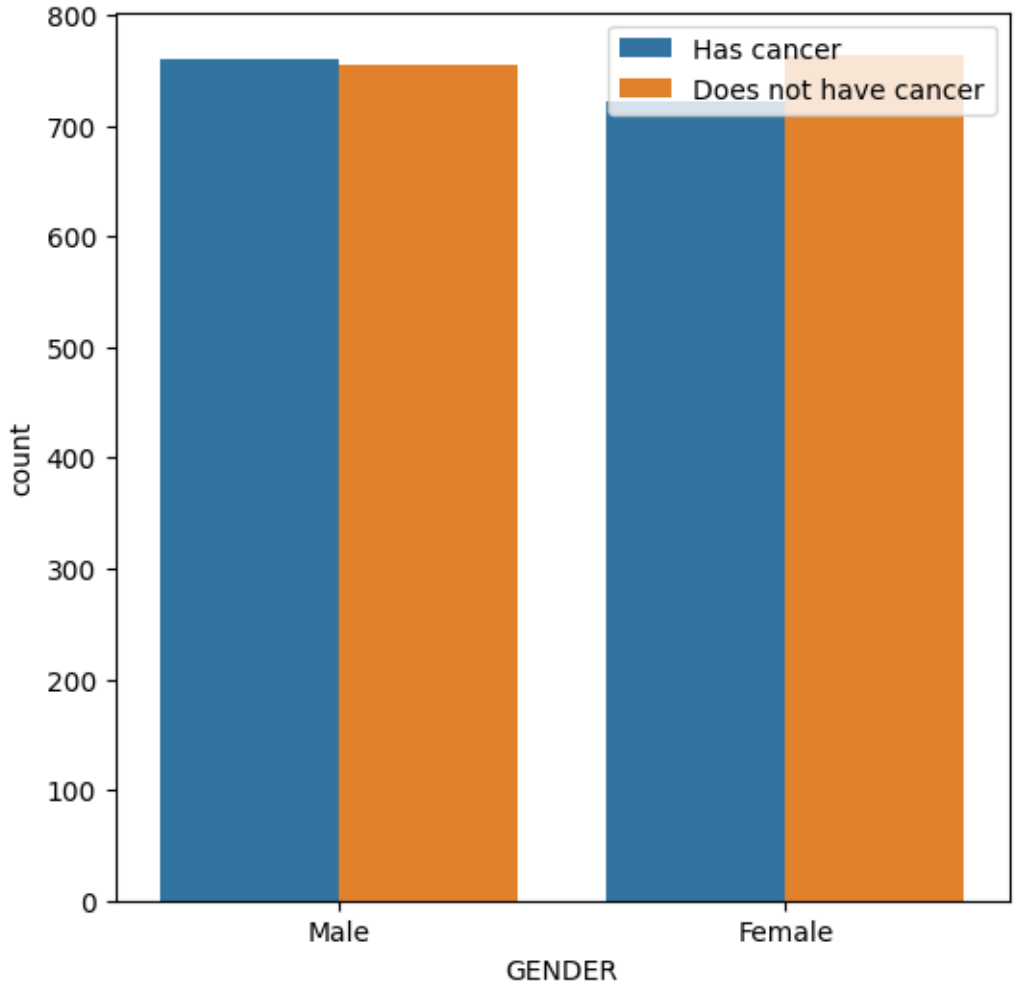
[15]:
```
LUNG_CANCER
Yes    50.6
No     49.4
Name: count, dtype: float64
```

There are 50.6% of Lung Cancer Patiences and 49.4% does not have lung cancer

8 Lung Cancer across Genders

[16]:
```
plt.figure(figsize=(6,6))
sns.countplot(data=lung_cancer_df,x='GENDER',hue='LUNG_CANCER')
plt.legend(["Has cancer", 'Does not have cancer'])
```

[16]: <matplotlib.legend.Legend at 0x230fd9bd910>

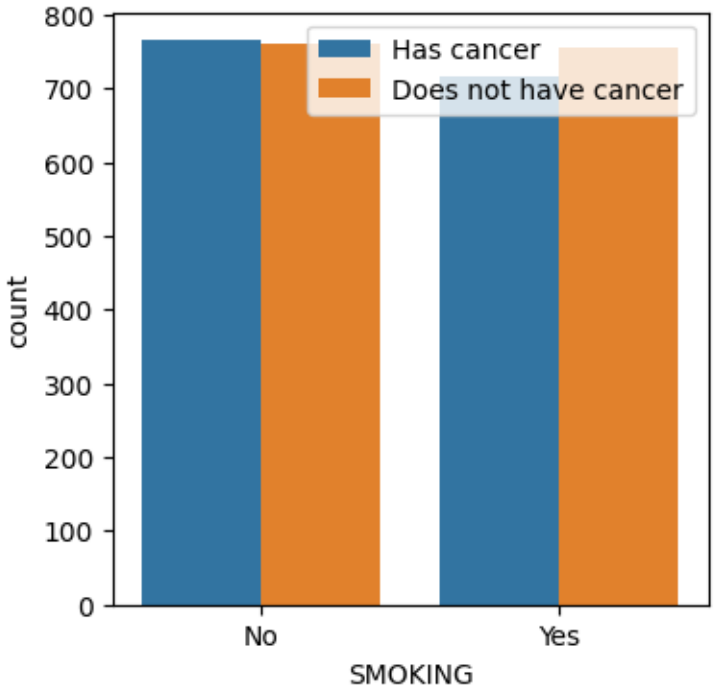


We see that in Male there is no much difference between Cancer and non-Cancer people[Male Cancer count is slighltly higher than non-Cancer]. But, in female we see that there is some amout of difference between cancer and non-cancer people[In female people with cancer are lesser than non-cancer]

9 Smoking and Lung Cancer:

[17]:
```
plt.figure(figsize=(4,4))
sns.countplot(data=lung_cancer_df,x='SMOKING',hue='LUNG_CANCER')
plt.legend(["Has cancer", 'Does not have cancer'])
```

[17]:
```
<matplotlib.legend.Legend at 0x230ff052bd0>
```

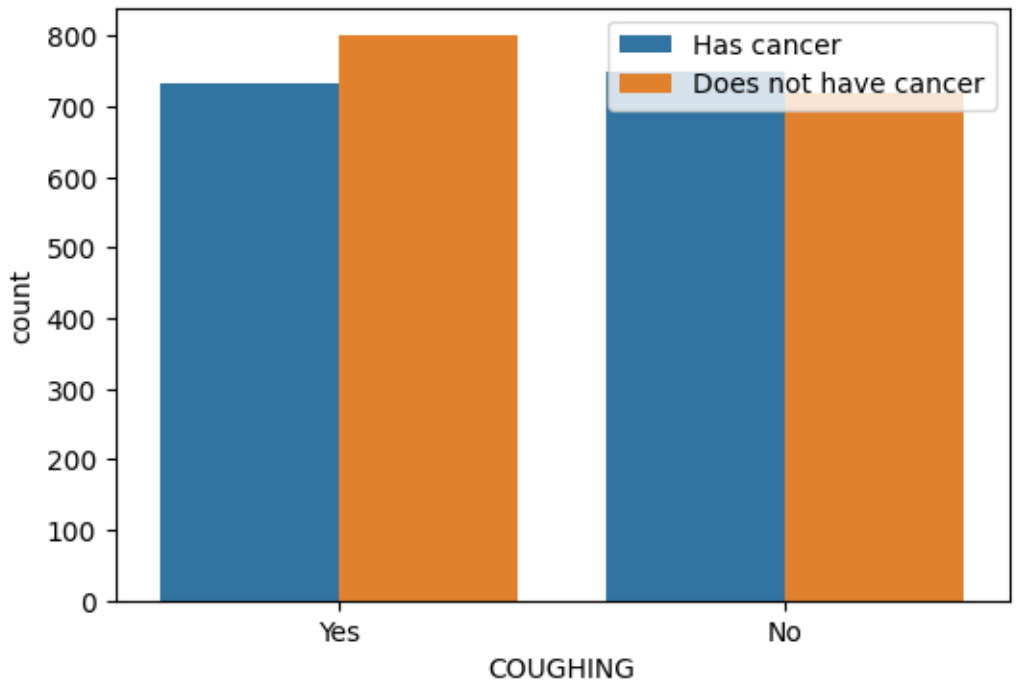


From the above data: People who doesn't smoke : Who has cancer and who doesn't have cancer is almost same. People who smoke: Who has cancer due to smoking is lesser than the people who does not have cancer even after smoking.

10 Coughing and Lung Cancer:

[18]:
```
plt.figure(figsize=(6,4))
sns.countplot(data=lung_cancer_df,x='COUGHING',hue='LUNG_CANCER')
plt.legend(["Has cancer", 'Does not have cancer'])
```

[18]:
```
<matplotlib.legend.Legend at 0x230ff0a8910>
```



The data shows that:

People who cough:Number of people who cough and have cancer is less than the people who cough and does not have cancer.

People who does not cough: People who does not cough and have cancer is slightly higher than the people who does not cough and does not have cancer.This shows that only smoking is not responsible for cancer, other factors are also included may be such has pollution, people working in some chemical/manufacturing industries. Because we have lack of data of peoples demography and profile we cannot conclude.

[19]:
```
lung_cancer_df.head()
```

[19]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC_DISEASE	\
0	Male	65	No	No	No	Yes	Yes	
1	Female	55	No	Yes	Yes	No	No	
2	Female	78	Yes	Yes	No	No	No	
3	Male	60	Yes	No	No	No	Yes	
4	Female	80	No	No	Yes	No	No	

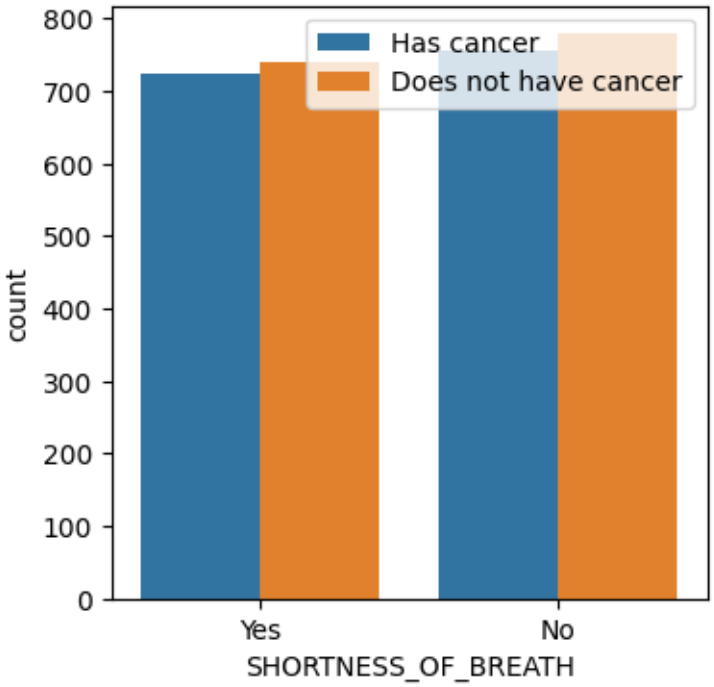
	FATIGUE	ALLERGY	WHEEZING	ALCOHOL_CONSUMING	COUGHING	SHORTNESS_OF_BREATH	\
0	No	Yes	Yes	Yes	Yes	Yes	
1	Yes	Yes	Yes	No	No	No	
2	Yes	No	Yes	No	No	Yes	
3	No	Yes	No	No	Yes	No	
4	Yes	No	Yes	No	No	No	

	SWALLOWING_DIFFICULTY	CHEST_PAIN	LUNG_CANCER
0	Yes	No	No
1	Yes	Yes	No
2	No	No	Yes
3	Yes	Yes	Yes
4	No	Yes	No

11 SHORTNESS OF BREATH with LUNG_CANCER

[20]:
```
plt.figure(figsize=(4,4))
sns.countplot(data=lung_cancer_df,x='SHORTNESS_OF_BREATH',hue='LUNG_CANCER')
plt.legend(["Has cancer", 'Does not have cancer'])
```

[20]: <matplotlib.legend.Legend at 0x230ff132410>



Interestingly, in all above of the plots we can see that it doesn't really matter if the symptoms are showing or not as some people may have cancer yet not show any symptoms. So, regular checkups would be wiser than waiting for symptoms to show up as that may lead it to deteriorate conditions.

[21]:
```
lung_cancer_df.head()
```

[21]:

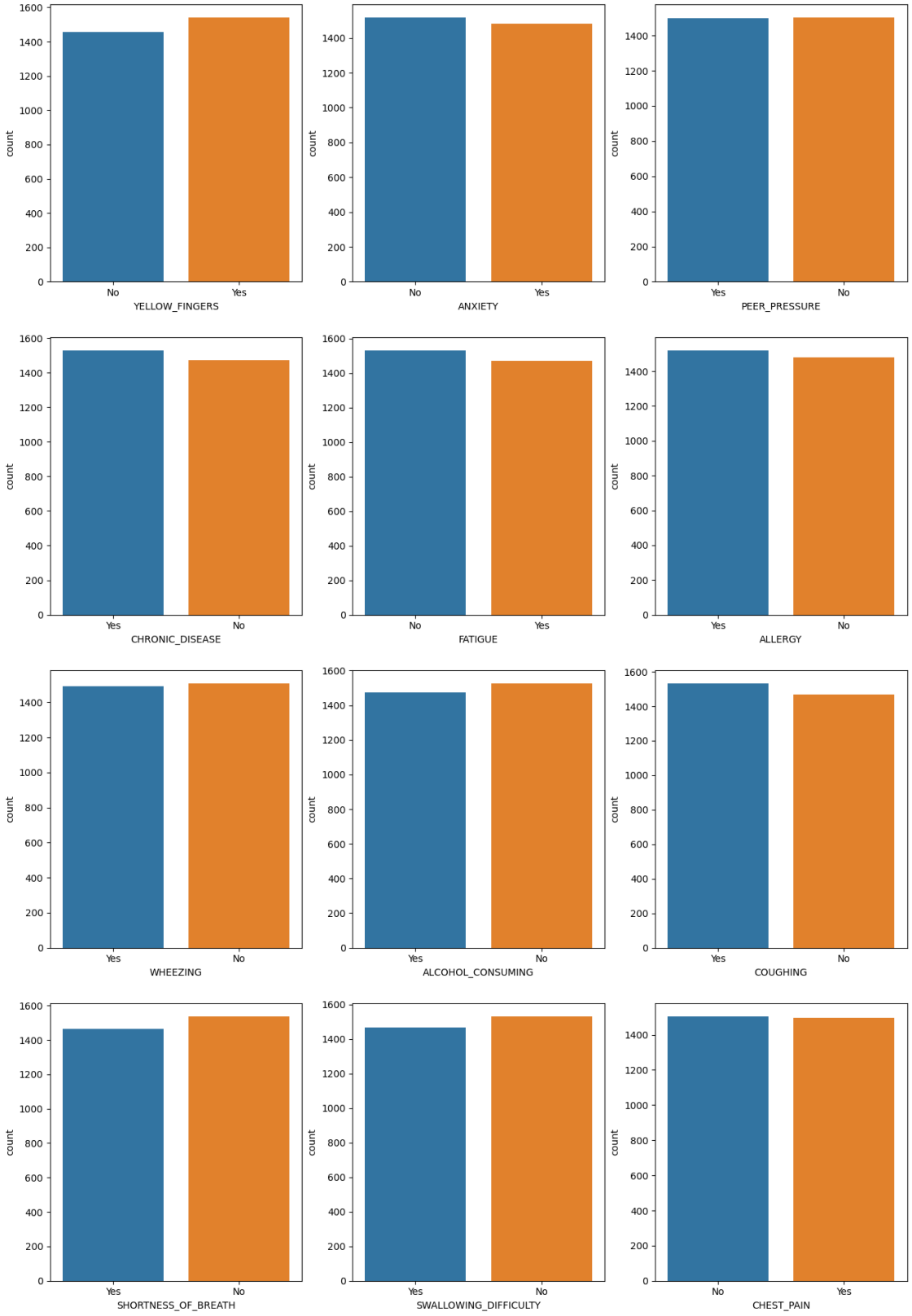
	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC_DISEASE
0	Male	65	No	No	No	Yes	Yes
1	Female	55	No	Yes	Yes	No	No
2	Female	78	Yes	Yes	No	No	No
3	Male	60	Yes	No	No	No	Yes
4	Female	80	No	No	Yes	No	No

	FATIGUE	ALLERGY	WHEEZING	ALCOHOL_CONSUMING	COUGHING	SHORTNESS_OF_BREATH
0	No	Yes	Yes	Yes	Yes	Yes
1	Yes	Yes	Yes	No	No	No
2	Yes	No	Yes	No	No	Yes
3	No	Yes	No	No	Yes	No
4	Yes	No	Yes	No	No	No

	SWALLOWING_DIFFICULTY	CHEST_PAIN	LUNG_CANCER
0	Yes	No	No
1	Yes	Yes	No
2	No	No	Yes
3	Yes	Yes	Yes
4	No	Yes	No

12 Lets plot all symptoms with cancer

[22]:
```
X = ['YELLOW_FINGERS', 'ANXIETY','PEER_PRESSURE', 'CHRONIC_DISEASE', 'FATIGUE',␣
 ↪'ALLERGY', 'WHEEZING','ALCOHOL_CONSUMING', 'COUGHING',␣
 ↪'SHORTNESS_OF_BREATH',  'SWALLOWING_DIFFICULTY', 'CHEST_PAIN']
fig, ax = plt.subplots(nrows = 4, ncols = 3) # 16 subplots
fig.set_size_inches(16,24) # set figure size

for i in range(4):
    for j in range(3):
        sns.countplot(x = lung_cancer_df[X[3 * i + j]] , ax = ax[i][j]) # count␣
 ↪plot
```


count
YELLOW_FINGERS
No
Yes
ANXIETY
No
Yes
PEER_PRESSURE
Yes
No
CHRONIC_DISEASE
Yes
No
FATIGUE
No
Yes
ALLERGY
Yes
No
WHEEZING
Yes
No
ALCOHOL_CONSUMING
Yes
No
COUGHING
Yes
No
SHORTNESS_OF_BREATH
Yes
No
SWALLOWING_DIFFICULTY
Yes
No
CHEST_PAIN
No
Yes

After plotting multiple countplots for the different features, we can see that there is no specific symptom that shows the significant amout of effect that causes cancer.There may be the reason that sometimes the cancer is genitically transmitted, or the data does not have much variance to say that only some specific symptom leads to cancer.

[23]:
```
lung_cancer_df.columns
```

[23]:
```
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')
```

Feature Engineering : runnign chi-square test

- Chi-square: a statistical test used to determine whether there is a significant association or independence between two categorical variables.
- Relationship to Data Chi-Square Values: Quantify the difference between observed and expected frequencies in categorical data. P-Values: Indicate the probability of observing the data if the null hypothesis is true. A low p-value suggests that the observed data is unlikely under the null hypothesis, leading to its rejection.

[26]:
```
import pandas as pd
from scipy.stats import chi2_contingency

# Assuming df_updates is your DataFrame containing the data

# Create a list to store the results
chi2_results_list = []

# List of categorical features
categorical_features = ['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']

def contingency_table(feature):
    p = pd.crosstab(lung_cancer_df['LUNG_CANCER'],lung_cancer_df[feature])
    return(p)
feature = []
pval = []
chi2_result = []

for i in categorical_features:
    feature.append(i)
    result = chi2_contingency(contingency_table(i))
    pval.append(round(float(result[1]),6))
```

```
    if float(result[1]) < 0.05:
        chi2_result.append("Significant")
    else:
        chi2_result.append("Insignificant")
```

[27]:
```
chisquare = pd.DataFrame(data={'PValue':pval,'Result':
 ↪chi2_result},index=feature)
chisquare
```

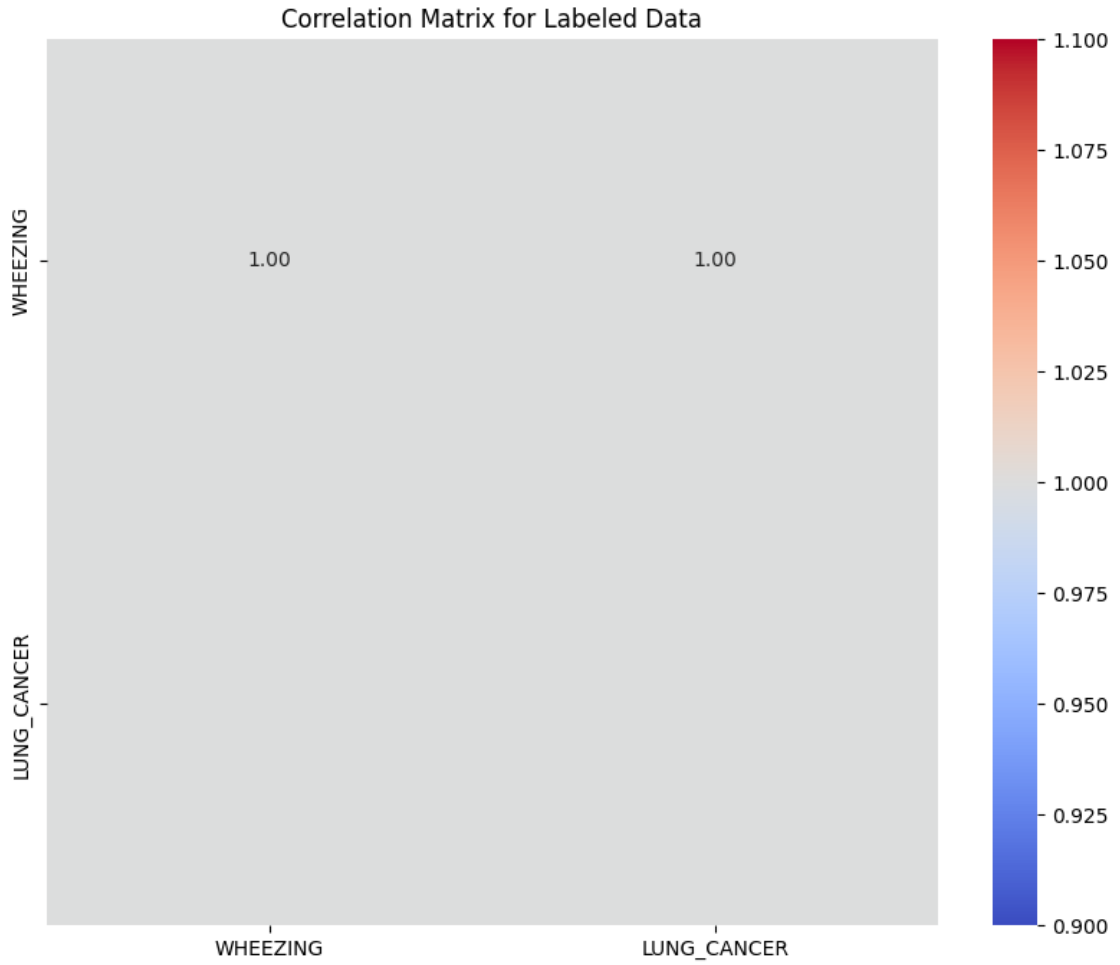
[27]:

	PValue	Result
GENDER	0.397512	Insignificant
AGE	0.431914	Insignificant
SMOKING	0.457905	Insignificant
YELLOW_FINGERS	0.499236	Insignificant
ANXIETY	0.447832	Insignificant
PEER_PRESSURE	0.189076	Insignificant
CHRONIC_DISEASE	0.600243	Insignificant
FATIGUE	0.930959	Insignificant
ALLERGY	0.749028	Insignificant
WHEEZING	0.037708	Significant
ALCOHOL_CONSUMING	0.102650	Insignificant
COUGHING	0.075776	Insignificant
SHORTNESS_OF_BREATH	0.925265	Insignificant
SWALLOWING_DIFFICULTY	0.671044	Insignificant
CHEST_PAIN	0.911374	Insignificant
LUNG_CANCER	0.000000	Significant

From the above method we got two significatn relation of WHEEZING AND LUNGCANCER, Lets plot corelation of these.

[41]:
```
lung_cancer_df = pd.DataFrame({

    'WHEEZING': [1,2],

    'LUNG_CANCER': [0,1]
})

# Select only numeric columns including the label
numeric_df = lung_cancer_df.select_dtypes(include=['number'])

# Calculate the correlation matrix
correlation_matrix = numeric_df.corr()

# Create a heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
```

```
# Add title
plt.title('Correlation Matrix for Labeled Data')

# Show the plot
plt.show()
```



Again we didnt find any co-relation of these variables.

13 Conclusion

Based on the data, we've made inferences such as relationships between symptoms and the likeliness of having lung cancer and relationships with gender, age, etc.

Training a machine learning model with this dataset could help us predict whether or not someone will be diagnosed with lung cancer or not.