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

```
data=pd.read_csv('survey lung cancer.csv')
```

```
data
```

	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	2	2	2	YES
1	M	74	2	1	1	1	2	2	2	1	1	1	2	2	2	YES
2	F	59	1	1	1	2	1	2	1	2	1	2	2	1	2	NO
3	M	63	2	2	2	1	1	1	1	1	2	1	1	2	2	NO
4	F	63	1	2	1	1	1	1	1	2	1	2	2	1	1	NO
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
304	F	56	1	1	1	2	2	2	1	1	2	2	2	2	1	YES
305	M	70	2	1	1	1	1	2	2	2	2	2	2	1	2	YES
306	M	58	2	1	1	1	1	1	2	2	2	2	1	1	2	YES
307	M	67	2	1	2	1	1	2	2	1	2	2	2	1	2	YES
308	M	62	1	1	1	2	1	2	2	2	2	1	1	2	1	YES

309 rows × 16 columns

Next steps: Generate code with data | New interactive sheet

```
data.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	2	2	2	YES
1	M	74	2	1	1	1	2	2	2	1	1	1	2	2	2	YES
2	F	59	1	1	1	2	1	2	1	2	1	2	2	1	2	NO
3	M	63	2	2	2	1	1	1	1	1	2	1	1	2	2	NO
4	F	63	1	2	1	1	1	1	1	2	1	2	2	1	1	NO

Next steps: Generate code with data | New interactive sheet

```
data.tail()
```

	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
304	F	56	1	1	1	2	2	2	1	1	2	2	2	2	1	YES
305	M	70	2	1	1	1	1	2	2	2	2	2	2	1	2	YES
306	M	58	2	1	1	1	1	1	2	2	2	2	1	1	2	YES
307	M	67	2	1	2	1	1	2	2	1	2	2	2	1	2	YES
308	M	62	1	1	1	2	1	2	2	2	2	1	1	2	1	YES

```
data.sample()
```

	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
9	M	53	2	2	2	2	2	1	2	1	2	1	1	2	2	YES

```
data.shape
```

```
(309, 16)
```

```
data.info()
```

```
<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
```

```
data.describe()
```

	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL CONSUMING	COUGHING	SHORTNESS OF BREATH	SWALLOWING DIFFICULTY	CHEST PAIN
count	309.000000	309.000000	309.000000	309.000000	309.000000	309.000000	309.000000	309.000000	309.000000	309.000000	309.000000	309.000000	309.000000	309.000000
mean	62.673139	1.563107	1.569579	1.498382	1.501618	1.504854	1.673139	1.556634	1.556634	1.556634	1.579288	1.640777	1.469256	1.556634
std	8.210301	0.496806	0.495938	0.500808	0.500808	0.500787	0.469827	0.497588	0.497588	0.497588	0.494474	0.480551	0.499863	0.497588
min	21.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
25%	57.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
50%	62.000000	2.000000	2.000000	1.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	1.000000	2.000000
75%	69.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000
max	87.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000

```
data.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

```
data.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')
```

```
data.index
```

```
RangeIndex(start=0, stop=309, step=1)
```

```
data.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

```
data.isnull().mean()*100
```

	0
GENDER	0.0
AGE	0.0
SMOKING	0.0
YELLOW_FINGERS	0.0
ANXIETY	0.0
PEER_PRESSURE	0.0
CHRONIC DISEASE	0.0
FATIGUE	0.0
ALLERGY	0.0
WHEEZING	0.0
ALCOHOL CONSUMING	0.0
COUGHING	0.0
SHORTNESS OF BREATH	0.0
SWALLOWING DIFFICULTY	0.0
CHEST PAIN	0.0
LUNG_CANCER	0.0

dtype: float64

```
data.notnull()
```

	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	True	True	True	True	True	True	True	True	True	True	True	True	True	True	True	True
1	True	True	True	True	True	True	True	True	True	True	True	True	True	True	True	True
2	True	True	True	True	True	True	True	True	True	True	True	True	True	True	True	True
3	True	True	True	True	True	True	True	True	True	True	True	True	True	True	True	True
4	True	True	True	True	True	True	True	True	True	True	True	True	True	True	True	True
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
304	True	True	True	True	True	True	True	True	True	True	True	True	True	True	True	True
305	True	True	True	True	True	True	True	True	True	True	True	True	True	True	True	True
306	True	True	True	True	True	True	True	True	True	True	True	True	True	True	True	True
307	True	True	True	True	True	True	True	True	True	True	True	True	True	True	True	True
308	True	True	True	True	True	True	True	True	True	True	True	True	True	True	True	True

309 rows × 16 columns

```
data.dropna()
```

	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	2	2	2	YES
1	M	74	2	1	1	1	2	2	2	1	1	1	2	2	2	YES
2	F	59	1	1	1	2	1	2	1	2	1	2	2	1	2	NO
3	M	63	2	2	2	1	1	1	1	1	2	1	1	2	2	NO
4	F	63	1	2	1	1	1	1	1	2	1	2	2	1	1	NO
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
304	F	56	1	1	1	2	2	2	1	1	2	2	2	2	1	YES
305	M	70	2	1	1	1	1	2	2	2	2	2	2	1	2	YES
306	M	58	2	1	1	1	1	1	2	2	2	2	1	1	2	YES
307	M	67	2	1	2	1	1	2	2	1	2	2	2	1	2	YES
308	M	62	1	1	1	2	1	2	2	2	2	1	1	2	1	YES

309 rows × 16 columns

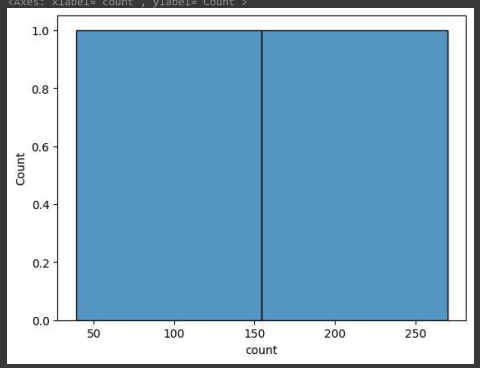
```
data.fillna(12)
```

	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	2	2	2	YES
1	M	74	2	1	1	1	2	2	2	1	1	1	2	2	2	YES
2	F	59	1	1	1	2	1	2	1	2	1	2	2	1	2	NO
3	M	63	2	2	2	1	1	1	1	1	2	1	1	2	2	NO
4	F	63	1	2	1	1	1	1	1	2	1	2	2	1	1	NO
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
304	F	56	1	1	1	2	2	2	1	1	2	2	2	2	1	YES
305	M	70	2	1	1	1	1	2	2	2	2	2	2	1	2	YES
306	M	58	2	1	1	1	1	1	2	2	2	2	1	1	2	YES
307	M	67	2	1	2	1	1	2	2	1	2	2	2	1	2	YES
308	M	62	1	1	1	2	1	2	2	2	2	1	1	2	1	YES

309 rows × 16 columns

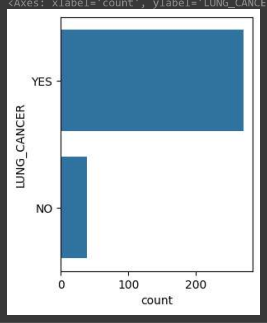
```
sns.histplot(data['LUNG_CANCER'].value_counts())
```

```
<Axes: xlabel='count', ylabel='Count'>
```



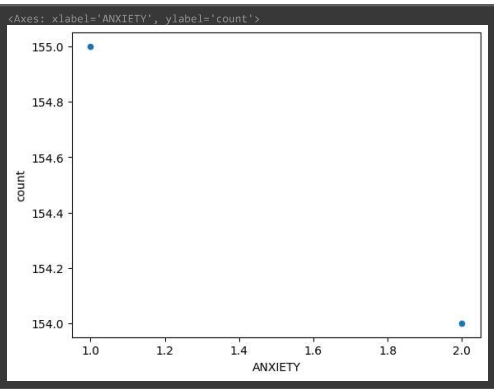

```
plt.figure(figsize=(3,4))
sns.countplot(data['LUNG_CANCER'])
```

```
<Axes: xlabel='count', ylabel='LUNG_CANCER'>
```



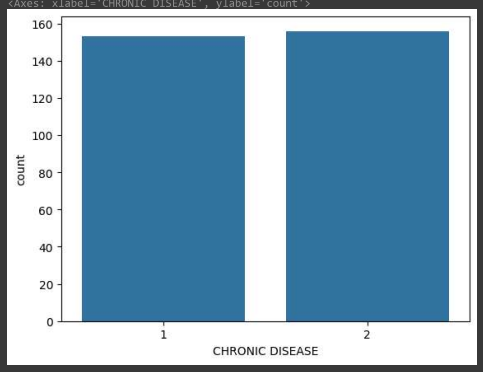

```
sns.scatterplot(data['ANXIETY'].value_counts())
```

```
<Axes: xlabel='ANXIETY', ylabel='count'>
```



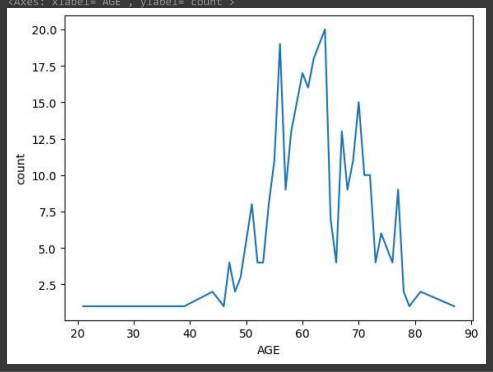

```
sns.barplot(data['CHRONIC DISEASE'].value_counts())
```

```
<Axes: xlabel='CHRONIC DISEASE', ylabel='count'>
```



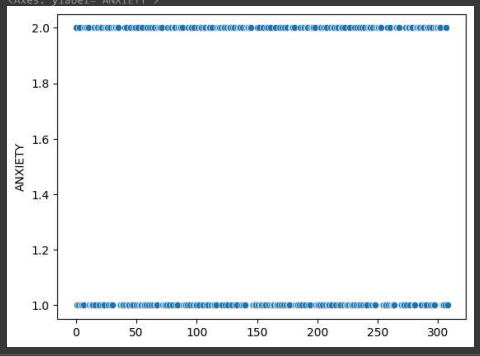

```
sns.lineplot(data['AGE'].value_counts())
```

```
<Axes: xlabel='AGE', ylabel='count'>
```



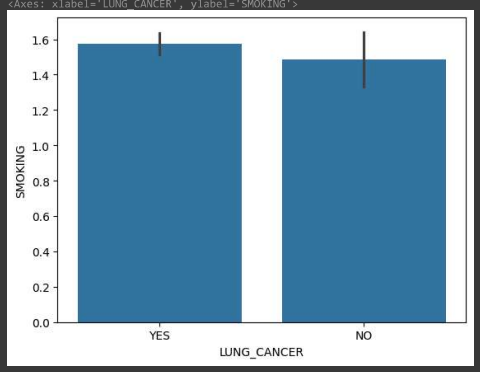

```
sns.scatterplot(data['ANXIETY'])
```

```
<Axes: ylabel='ANXIETY'>
```



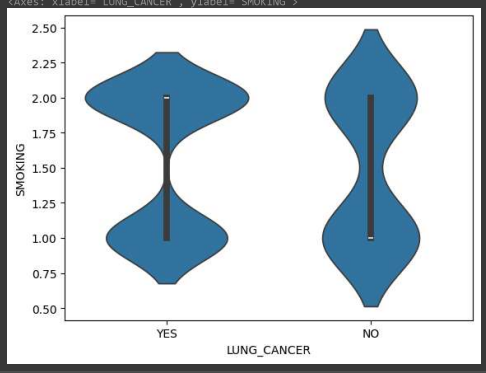

```
sns.barplot(x='LUNG_CANCER',y='SMOKING',data=data)
```

```
<Axes: xlabel='LUNG_CANCER', ylabel='SMOKING'>
```



```
sns.violinplot(x='LUNG_CANCER',y='SMOKING',data=data)
```

```
<Axes: xlabel='LUNG_CANCER', ylabel='SMOKING'>
```




```
data.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
le = LabelEncoder()
```

```
a=['GENDER','LUNG_CANCER']
for i in a:
        data[i]=le.fit_transform(data[i])
```

```
from sklearn.model_selection import train_test_split
```

```
x_train,x_test,y_train,y_test=train_test_split(data.drop('LUNG_CANCER',axis=1),data['LUNG_CANCER'],test_size=0.2,random_state=42)
```

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x_train=sc.fit_transform(x_train)
x_test=sc.transform(x_test)
```

```
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
lr.fit(x_train,y_train)
```

```
▾ LogisticRegression ⓘ ?
LogisticRegression()
```

```
lr.score(x_train,y_train)*100,lr.score(x_test,y_test)*100
```

```
(92.71255060728745, 96.7741935483871)
```

```
y_pred=lr.predict(x_test)
```

```
from sklearn.svm import SVC
sv=SVC()
sv.fit(x_train,y_train)
```

```
▾ SVC ⓘ ?
SVC()
```

```
sv.score(x_train,y_train)*100,sv.score(x_test,y_test)*100
```

```
(94.73684210526315, 96.7741935483871)
```

```
y_pred=sv.predict(x_test)
```

```
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,y_pred)
```

```
cm
```

```
array([[ 1,  1],
       [ 1, 59]])
```

```
from sklearn.metrics import accuracy_score,classification_report
print("Accuracy: " , accuracy_score(y_test,y_pred))
print("Classification Reprot:\n" ,classification_report(y_test,y_pred))
```

```
Accuracy:  0.967741935483871
Classification Reprot:
               precision    recall  f1-score   support

           0       0.50      0.50      0.50         2
           1       0.98      0.98      0.98        60

    accuracy                           0.97        62
   macro avg       0.74      0.74      0.74        62
weighted avg       0.97      0.97      0.97        62
```

KNN MODEL

```
from sklearn.neighbors import KNeighborsClassifier
classification = KNeighborsClassifier(n_neighbors=5,metric='minkowski')
classification.fit(x_train,y_train)
```

```
▾ KNeighborsClassifier ⓘ ?
KNeighborsClassifier()
```

```
y_pred=classification.predict(x_test)
```

```
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,y_pred)
```

```
cm
```

```
array([[ 1,  1],
       [ 3, 57]])
```

```
# Import necessary libraries
import matplotlib.pyplot as plt
import pandas as pd
from sklearn import datasets, neighbors
from sklearn.model_selection import train_test_split
X, y = datasets.make_blobs(n_samples=500, n_features=2, centers=4, cluster_std=1.5, random_state=4)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
knn = neighbors.KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
from mlxtend.plotting import plot_decision_regions
plot_decision_regions(X, y, clf=knn, legend=2)
plt.xlabel('X')
plt.ylabel('Y')
plt.title('KNN with K=5')
plt.savefig('KNN with K=5.jpeg', bbox_inches="tight", dpi=150)
plt.show()
```

```
/usr/local/lib/python3.12/dist-packages/mlxtend/plotting/decision_regions.py:346: UserWarning: You passed a edgecolor/edgecolors ('black') for an unfilled marker ('x').  Matplotlib is ignoring the edgecolor in favo
  ax.scatter(
```