1)allotted concept: Cancer prediction using naive bayes

```
#For ignoring warning
import warnings
warnings.filterwarnings("ignore")
df=pd.read_csv('/content/cancer patient data sets.csv')
```

```
df
```

	index	Age	Gender	Air Pollution	Alcohol use	Dust Allergy	OccuPational Hazards	Genetic Risk	chroni Lu Diseas
0	0	33	1	2	4	5	4	3	
1	1	17	1	3	1	5	3	4	
2	2	35	1	4	5	6	5	5	
3	3	37	1	7	7	7	7	6	
4	4	46	1	6	8	7	7	7	
...	...	...	...	...	...	...	...	...	
995	995	44	1	6	7	7	7	7	
996	996	37	2	6	8	7	7	7	
997	997	25	2	4	5	6	5	5	
998	998	18	2	6	8	7	7	7	
999	999	47	1	6	5	6	5	5	

1000 rows × 24 columns

```
import pandas as pd
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
df = pd.read_csv('/content/cancer patient data sets.csv')
X = df.drop('Chest Pain', axis=1) # Features
y = df['Chest Pain'] # Target variable
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
gnb = GaussianNB()
gnb.fit(X_train, y_train)
y_pred = gnb.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print('Accuracy:', accuracy)
```

```
Accuracy: 0.87
```

2)a) Perform Regression analysis for a given data set using Seaborn Visualisation with Pandas and Matplotlib.(Individual Data Set)- regression Plot, Multiple plot, scatter plot, correlation coeeficient and all Categorical plots.

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

```
#For ignoring warning
import warnings
warnings.filterwarnings("ignore")
df=pd.read_csv('/content/cancer patient data sets.csv')
df
```

	index	Age	Gender	Air Pollution	Alcohol use	Dust Allergy	OccuPational Hazards	Genetic Risk	chronic Lung Disease	Balanced Diet	...	Coughing of Blood	Fatigue	Weight Loss	Shortness of Breath	Wheezing
0	0	33	1	2	4	5	4	3	2	2	...	4	3	4	2	2
1	1	17	1	3	1	5	3	4	2	2	...	3	1	3	7	8
2	2	35	1	4	5	6	5	5	4	6	...	8	8	7	9	2
3	3	37	1	7	7	7	7	6	7	7	...	8	4	2	3	1
4	4	46	1	6	8	7	7	7	6	7	...	9	3	2	4	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	995	44	1	6	7	7	7	7	6	7	...	7	5	3	2	7
996	996	37	2	6	8	7	7	7	6	7	...	7	9	6	5	7
997	997	25	2	4	5	6	5	5	4	6	...	8	8	7	9	2
998	998	18	2	6	8	7	7	7	6	7	...	9	3	2	4	1

```
df.shape
```

```
(1000, 24)
```

```
#Checking for Duplicates
df.duplicated().sum()
```

```
0
```

```
#Removing Duplicates
df=df.drop_duplicates()
```

```
#Checking for null values
df.isnull().sum()
```

```
index                     0
Age                       0
Gender                    0
Air Pollution             0
Alcohol use               0
Dust Allergy              0
OccuPational Hazards      0
Genetic Risk              0
chronic Lung Disease      0
Balanced Diet             0
Obesity                   0
Smoking                   0
Passive Smoker            0
Chest Pain                0
Coughing of Blood         0
Fatigue                   0
Weight Loss               0
Shortness of Breath       0
Wheezing                  0
Swallowing Difficulty     0
Clubbing of Finger Nails  0
Frequent Cold             0
Dry Cough                 0
Snoring                   0
dtype: int64
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000 entries, 0 to 999
Data columns (total 24 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   index                     1000 non-null   int64
 1   Age                       1000 non-null   int64
 2   Gender                    1000 non-null   int64
 3   Air Pollution             1000 non-null   int64
 4   Alcohol use               1000 non-null   int64
 5   Dust Allergy              1000 non-null   int64
 6   OccuPational Hazards      1000 non-null   int64
 7   Genetic Risk              1000 non-null   int64
 8   chronic Lung Disease      1000 non-null   int64
 9   Balanced Diet             1000 non-null   int64
 10  Obesity                   1000 non-null   int64
 11  Smoking                   1000 non-null   int64
 12  Passive Smoker            1000 non-null   int64
 13  Chest Pain                1000 non-null   int64
```

```
 14  Coughing of Blood         1000 non-null   int64
 15  Fatigue                   1000 non-null   int64
 16  Weight Loss               1000 non-null   int64
 17  Shortness of Breath       1000 non-null   int64
 18  Wheezing                  1000 non-null   int64
 19  Swallowing Difficulty     1000 non-null   int64
 20  Clubbing of Finger Nails  1000 non-null   int64
 21  Frequent Cold             1000 non-null   int64
 22  Dry Cough                 1000 non-null   int64
 23  Snoring                   1000 non-null   int64
dtypes: int64(24)
memory usage: 195.3 KB
```

```
df.describe()
```

	index	Age	Gender	Air Pollution	Alcohol use	Dust Allergy	OccuPational Hazards	Genetic Risk	chronic Lung Disease	Balanced Diet	...	Coughin of Bloo
count	1000.000000	1000.000000	1000.000000	1000.0000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	...	1000.00000
mean	499.500000	37.174000	1.402000	3.8400	4.563000	5.165000	4.840000	4.580000	4.380000	4.491000	...	4.85900
std	288.819436	12.005493	0.490547	2.0304	2.620477	1.980833	2.107805	2.126999	1.848518	2.135528	...	2.42796
min	0.000000	14.000000	1.000000	1.0000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	...	1.00000
25%	249.750000	27.750000	1.000000	2.0000	2.000000	4.000000	3.000000	2.000000	3.000000	2.000000	...	3.00000
50%	499.500000	36.000000	1.000000	3.0000	5.000000	6.000000	5.000000	5.000000	4.000000	4.000000	...	4.00000
75%	749.250000	45.000000	2.000000	6.0000	7.000000	7.000000	7.000000	7.000000	6.000000	7.000000	...	7.00000
max	999.000000	73.000000	2.000000	8.0000	8.000000	8.000000	8.000000	7.000000	7.000000	7.000000	...	9.00000

8 rows × 24 columns

```
from sklearn import preprocessing
le=preprocessing.LabelEncoder()
df['Gender']=le.fit_transform(df['Gender'])
df['Dust Allergy']=le.fit_transform(df['Dust Allergy'])
df['Genetic Risk']=le.fit_transform(df['Genetic Risk'])
```

```
df
```

	index	Age	Gender	Air Pollution	Alcohol use	Dust Allergy	OccuPational Hazards	Genetic Risk	chronic Lung Disease	Balanced Diet	...	Coughing of Blood	Fatigue	Weight Loss	Shortness of Breath	Wheezing
0	0	33	0	2	4	4	4	2	2	2	...	4	3	4	2	2
1	1	17	0	3	1	4	3	3	2	2	...	3	1	3	7	8
2	2	35	0	4	5	5	5	4	4	6	...	8	8	7	9	2
3	3	37	0	7	7	6	7	5	7	7	...	8	4	2	3	1
4	4	46	0	6	8	6	7	6	6	7	...	9	3	2	4	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	995	44	0	6	7	6	7	6	6	7	...	7	5	3	2	7
996	996	37	1	6	8	6	7	6	6	7	...	7	9	6	5	7
997	997	25	1	4	5	5	5	4	4	6	...	8	8	7	9	2
998	998	18	1	6	8	6	7	6	6	7	...	9	3	2	4	1
999	999	47	0	6	5	5	5	4	4	6	...	8	8	7	9	2

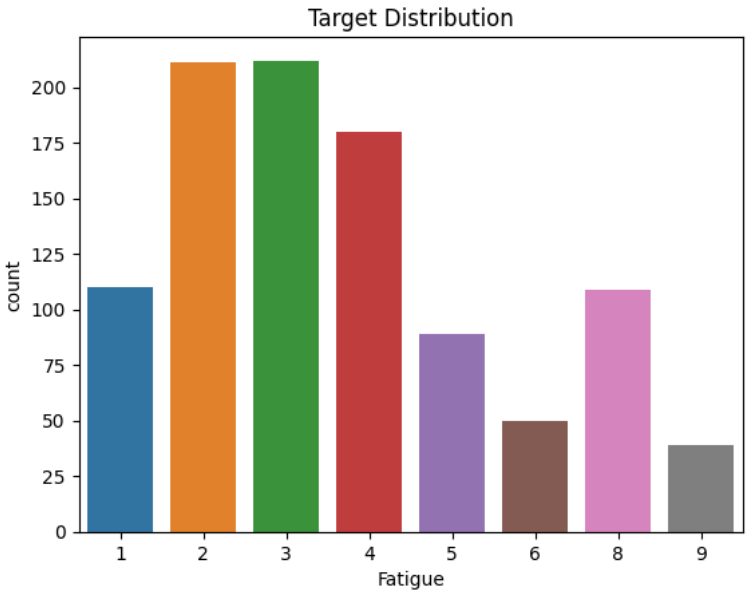
1000 rows × 24 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000 entries, 0 to 999
Data columns (total 24 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
```

```
 0   index                     1000 non-null   int64
 1   Age                       1000 non-null   int64
 2   Gender                    1000 non-null   int64
 3   Air Pollution             1000 non-null   int64
 4   Alcohol use               1000 non-null   int64
 5   Dust Allergy              1000 non-null   int64
 6   OccuPational Hazards      1000 non-null   int64
 7   Genetic Risk              1000 non-null   int64
 8   chronic Lung Disease      1000 non-null   int64
 9   Balanced Diet             1000 non-null   int64
 10  Obesity                   1000 non-null   int64
 11  Smoking                   1000 non-null   int64
 12  Passive Smoker            1000 non-null   int64
 13  Chest Pain                1000 non-null   int64
 14  Coughing of Blood         1000 non-null   int64
 15  Fatigue                   1000 non-null   int64
 16  Weight Loss               1000 non-null   int64
 17  Shortness of Breath       1000 non-null   int64
 18  Wheezing                  1000 non-null   int64
 19  Swallowing Difficulty     1000 non-null   int64
 20  Clubbing of Finger Nails  1000 non-null   int64
 21  Frequent Cold             1000 non-null   int64
 22  Dry Cough                 1000 non-null   int64
 23  Snoring                   1000 non-null   int64
dtypes: int64(24)
memory usage: 195.3 KB
```

```
#Let's check the distributaion of Target variable.
sns.countplot(x='Fatigue', data=df,)
plt.title('Target Distribution');
```



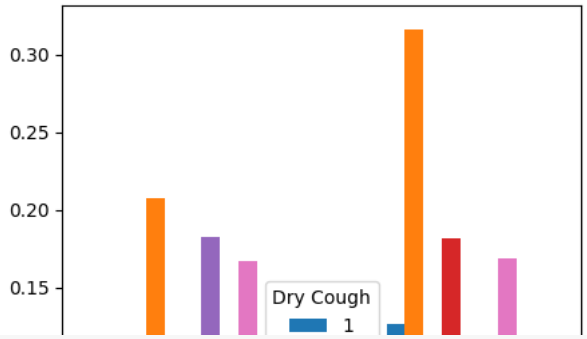

```
df['Dry Cough'].value_counts()
```

```
2    251
7    168
4    141
5    131
1    119
3    101
6     89
Name: Dry Cough, dtype: int64
```

```
# function for plotting
def plot(col, df=df):
    return df.groupby(col)['Dry Cough'].value_counts(normalize=True).unstack().plot(kind='bar', figsize=(5,5))
```

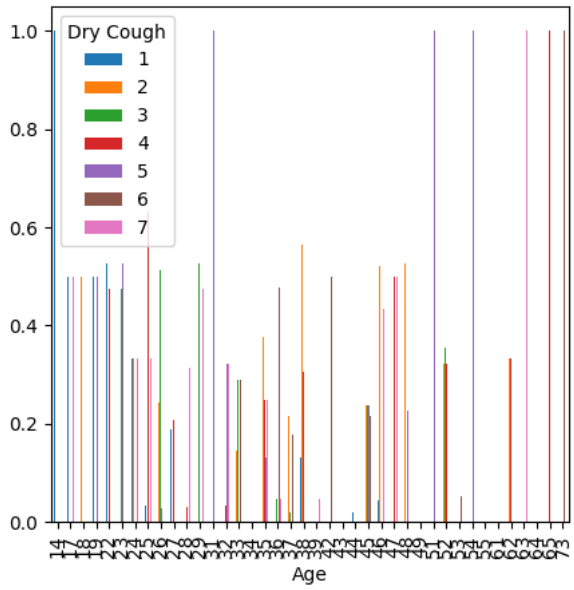
```
plot('Gender')
```

```
<Axes: xlabel='Gender'>
```



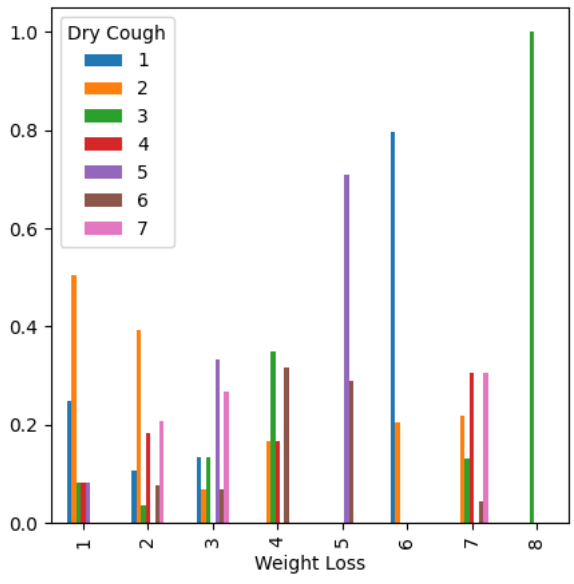

```
plot('Age')
```

```
<Axes: xlabel='Age'>
```



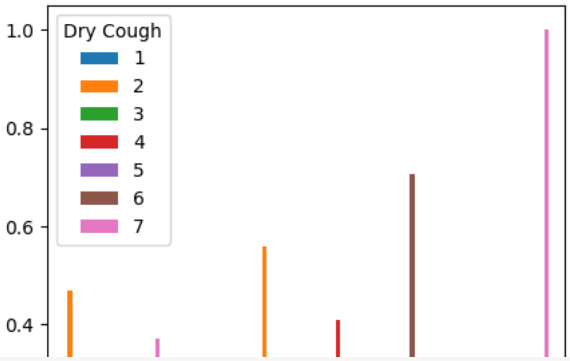

```
plot('Weight Loss')
```

```
<Axes: xlabel='Weight Loss'>
```




```
plot('Wheezing')
```

<Axes: xlabel='Wheezing'>




```
df_new=df.drop(columns=['Gender','Age', 'Weight Loss', 'Wheezing'])
df_new
```

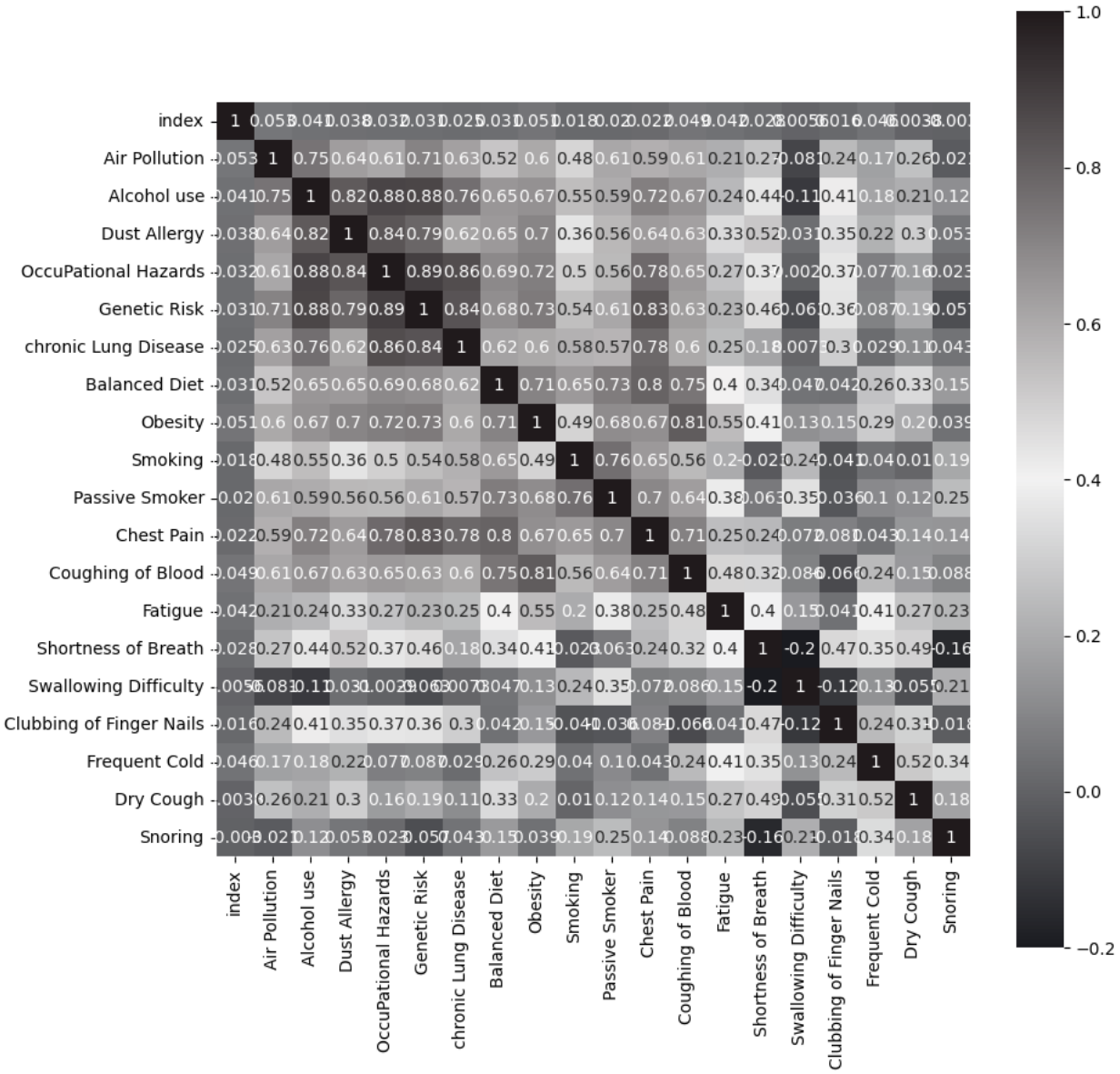
	index	Air Pollution	Alcohol use	Dust Allergy	OccuPational Hazards	Genetic Risk	chronic Lung Disease	Balanced Diet	Obesity	Smoking	Passive Smoker	Chest Pain	Coughing of Blood	Fatigue	Shortness of Breath	Su Di
0	0	2	4	4	4	2	2	2	4	3	2	2	4	3	2	
1	1	3	1	4	3	3	2	2	2	2	4	2	3	1	7	
2	2	4	5	5	5	4	4	6	7	2	3	4	8	8	9	
3	3	7	7	6	7	5	7	7	7	7	7	7	8	4	3	
4	4	6	8	6	7	6	6	7	7	8	7	7	9	3	4	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	995	6	7	6	7	6	6	7	7	7	8	7	7	5	2	
996	996	6	8	6	7	6	6	7	7	7	8	7	7	9	5	
997	997	4	5	5	5	4	4	6	7	2	3	4	8	8	9	
998	998	6	8	6	7	6	6	7	7	8	7	7	9	3	4	
999	999	6	5	5	5	4	4	6	7	2	3	4	8	8	9	

1000 rows × 20 columns

```
#Finding Correlation
cn=df_new.corr()
cn
```

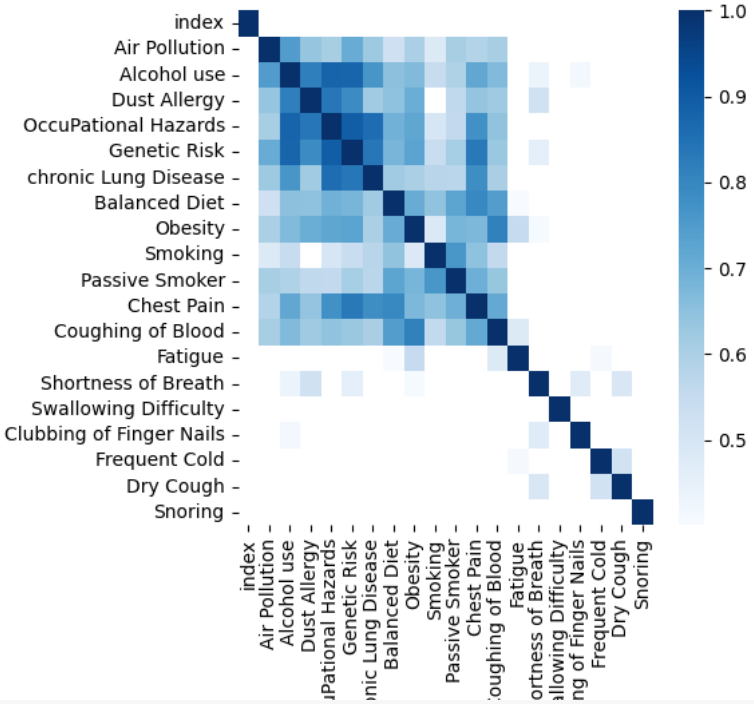
	index	Air Pollution	Alcohol use	Dust Allergy	OccuPational Hazards	Genetic Risk	chronic Lung Disease	Balanced Diet	Obesity	Smoking	Passive Smoker	Chest Pain	Coughing of Blood
index	1.000000	0.053307	0.041374	0.037960	0.032355	0.030725	0.025177	0.030743	0.050584	0.018407	0.019517	0.022210	0.049401
Air Pollution	0.053307	1.000000	0.747293	0.637503	0.608924	0.705276	0.626701	0.524873	0.601468	0.481902	0.606764	0.585734	0.607829
Alcohol use	0.041374	0.747293	1.000000	0.818644	0.878786	0.877210	0.763576	0.653352	0.669312	0.547035	0.592576	0.717242	0.667612
Dust Allergy	0.037960	0.637503	0.818644	1.000000	0.835860	0.787904	0.619556	0.647197	0.700676	0.358691	0.560002	0.639983	0.625291
OccuPational Hazards	0.032355	0.608924	0.878786	0.835860	1.000000	0.893049	0.858284	0.691509	0.722191	0.497693	0.555311	0.775619	0.645947
Genetic Risk	0.030725	0.705276	0.877210	0.787904	0.893049	1.000000	0.836231	0.679905	0.729826	0.543259	0.609071	0.831751	0.632236
chronic Lung Disease	0.025177	0.626701	0.763576	0.619556	0.858284	0.836231	1.000000	0.622632	0.601754	0.578585	0.572698	0.782646	0.602987
Balanced													

```
#Correlation
cmap=sns.diverging_palette(260,-10,s=10, l=10, n=6,
as_cmap=True)
plt.subplots(figsize=(10,10))
sns.heatmap(cn,cmap=cmap,annot=True, square=True)
plt.show()
```



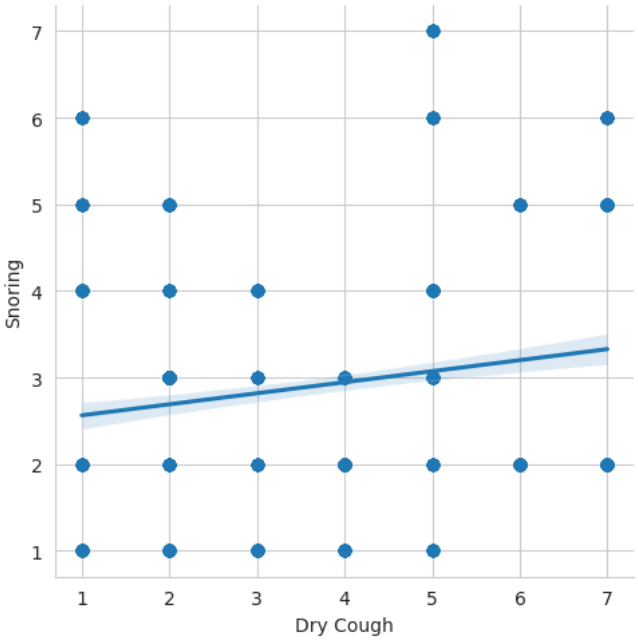

```
kot = cn[cn>=.40]
plt.figure(figsize=(5,5))
sns.heatmap(kot, cmap="Blues")
```

```
<Axes: >
```

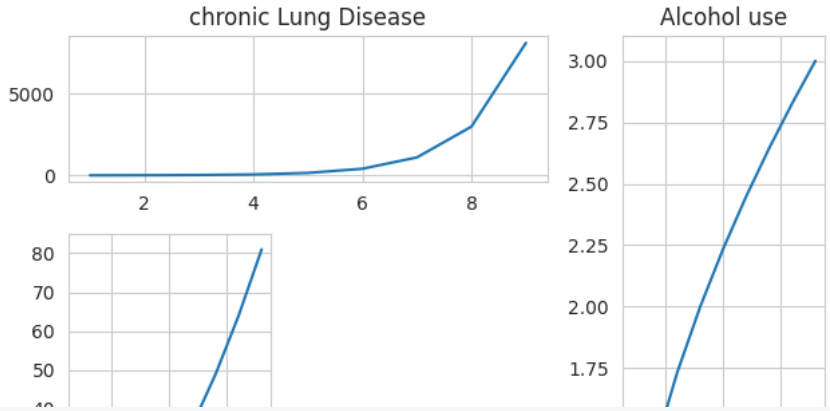



```
sns.set_style('whitegrid')
sns.lmplot(x ='Dry Cough', y ='Snoring', data =df)
```

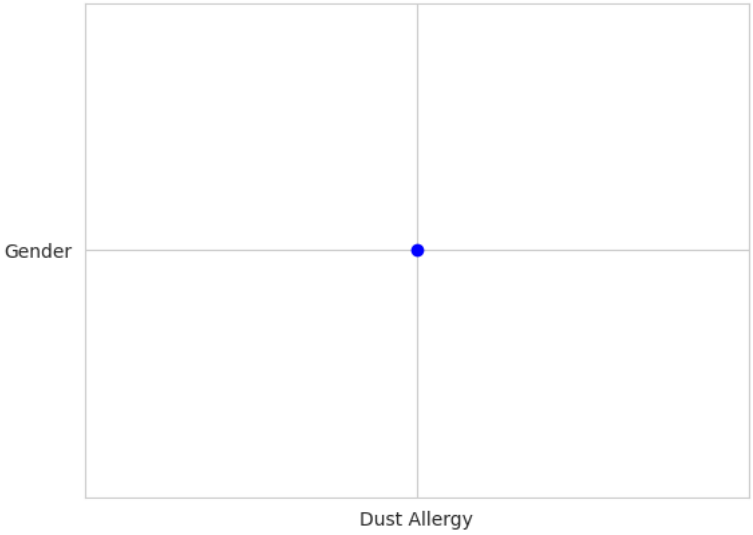
```
<seaborn.axisgrid.FacetGrid at 0x7fee251dcf70>
```




```
import math
plot1 = plt.subplot2grid((3, 3), (0, 0), colspan=2)
plot2 = plt.subplot2grid((3, 3), (0, 2), rowspan=3, colspan=2)
plot3 = plt.subplot2grid((3, 3), (1, 0), rowspan=2)
x = np.arange(1, 10)
plot2.plot(x, x**0.5)
plot2.set_title('Alcohol use')
plot1.plot(x, np.exp(x))
plot1.set_title('chronic Lung Disease')
plot3.plot(x, x*x)
plt.tight_layout()
plt.show()
```




```
 x =['Dust Allergy']
y =['Gender']
plt.scatter(x, y, c ="blue")
plt.show()
```



3)b) Perform t test for the sample you have analysed

```
import numpy as np
from scipy import stats

# Example data
group1 = np.array([1, 2, 3, 4, 5])
group2 = np.array([6, 7, 8, 9, 10])

# Calculate t statistic and p-value
t, p = stats.ttest_ind(group1, group2)

# Print results
print('t statistic:', t)
print('p-value:', p)
```

```
t statistic: -5.0
p-value: 0.001052825793366539
```

✓ 0s completed at 10:50 PM