SucharithaPuppala_TermProject

August 10, 2022

0.1 WEEK 10

0.2 TERM PROJECT

0.2.1 Author: Sucharitha Puppala

0.2.2 Date : 08-08-2022

0.2.3 Analysis of the Lung Cancer data set.

[1]:
```
# Importing the necessary libraries.

from __future__ import print_function, division
import thinkstats2
import thinkplot
import sys
import numpy as np
import pandas
import random
```

Step 1 : Reading the data set.

[2]:
```
# Function to read the data file i.e. the Survey of the Lung Cancer data file␣
↪which in CSV format.
def LungcancerData():
    df = pandas.read_csv('survey lung cancer.csv')
    return df
```

Step 2 : Getting the data information.

[3]:
```
LCdata = LungcancerData()
LCdata.head()
```

[3]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	\
0	M	69	1	2	2	1	
1	M	74	2	1	1	1	
2	F	59	1	1	1	2	
3	M	63	2	2	2	1	
4	F	63	1	2	1	1	

	CHRONIC DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL CONSUMING	COUGHING	\

```
0                1         2         1         2                2         2
1                2         2         2         1                1         1
2                1         2         1         2                1         2
3                1         1         1         1                2         1
4                1         1         1         2                1         2

   SHORTNESS OF BREATH  SWALLOWING DIFFICULTY  CHEST PAIN LUNG_CANCER
0                    2                      2           2         YES
1                    2                      2           2         YES
2                    2                      1           2          NO
3                    1                      2           2          NO
4                    2                      1           1          NO
```

Step 3 : Include a histogram of each of the 5 variables – in your summary and analysis, identify any outliers and explain the reasoning for them being outliers and how you believe they should be handled (Chapter 2). From the above data set I would like to select the below list of vriables for plotting the histograms:

1. AGE
2. SMOKING
3. CHOUGHING
4. ALCOHOL CONSUMING
5. CHEST PAIN

[4]:
```
# Getting the info of the data set.
LCdata.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
```

[5]:
```
# Checking for null values.
LCdata.isna().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
```

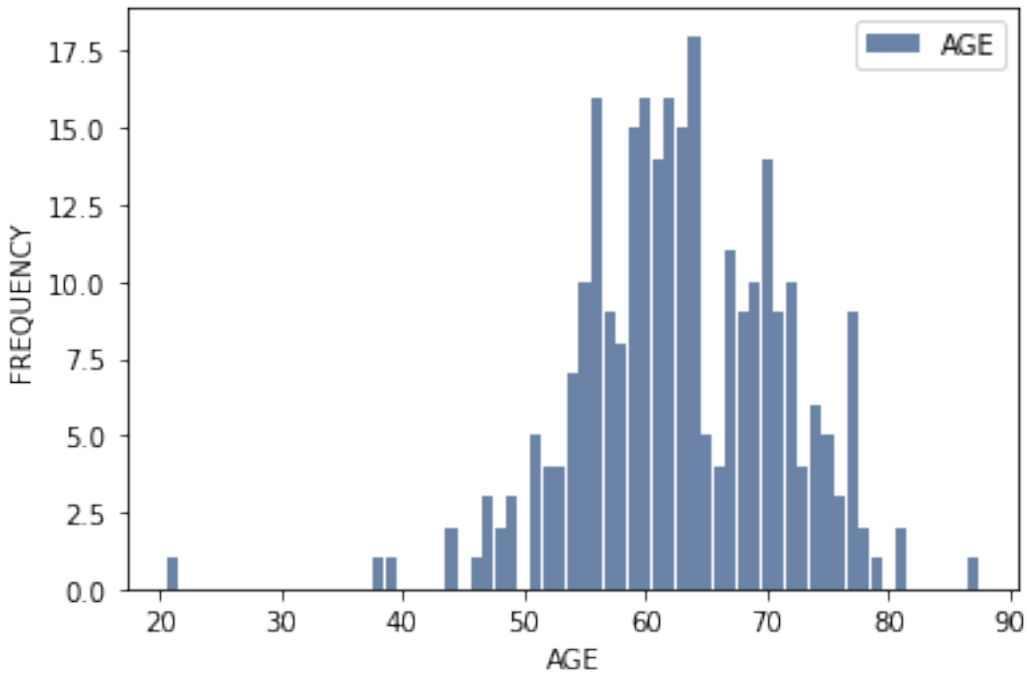
[6]:
```
# Checking for duplicates.
duplicate = LCdata[LCdata.duplicated()].shape[0]
print(f" We have {duplicate} duplicate entries out of {LCdata.shape[0]} entries␣
 ↪in the LUNG CANCER dataset.")
```

```
 We have 33 duplicate entries out of 309 entries in the LUNG CANCER dataset.
```

[7]:
```
# Dropping the duplicates.
LCdata.drop_duplicates(keep='first',inplace=True)
print(f"\n The duplicate records are removed and we have {LCdata.shape[0]}␣
 ↪entries in the LUNG CANCER dataset.")
```

```
 The duplicate records are removed and we have 276 entries in the LUNG CANCER
dataset.
```

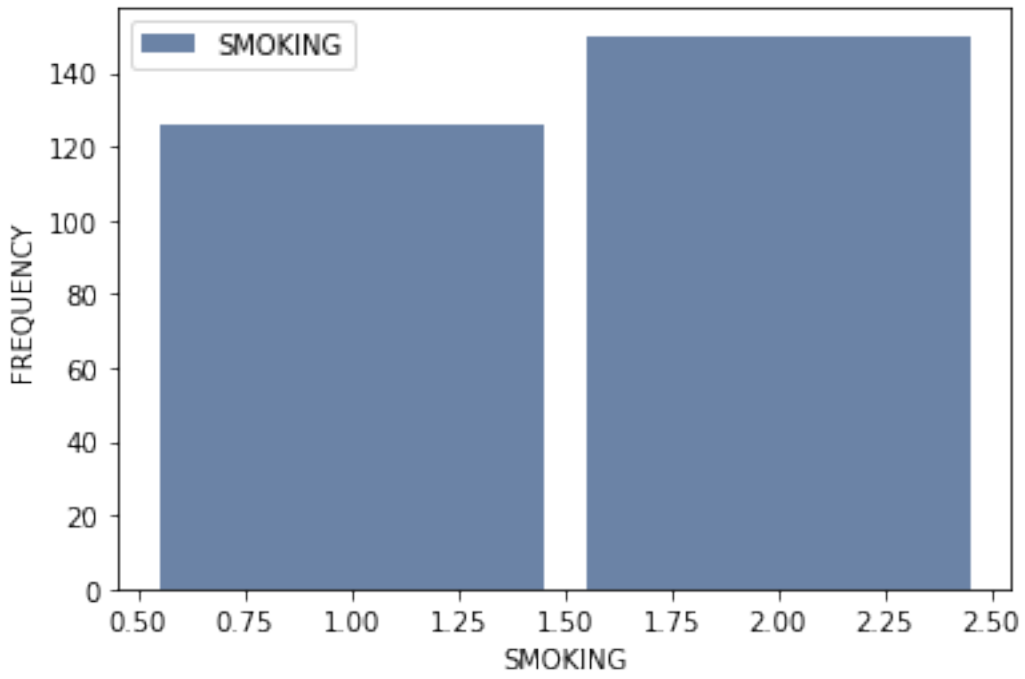
[8]:
```
# 1. Plotting the histogram for 'AGE' variable from the LUNG CANCER dataset.
hist = thinkstats2.Hist(LCdata.AGE, label='AGE')
thinkplot.Hist(hist)
thinkplot.Show(xlabel='AGE', ylabel='FREQUENCY')
```




```
<Figure size 576x432 with 0 Axes>
```

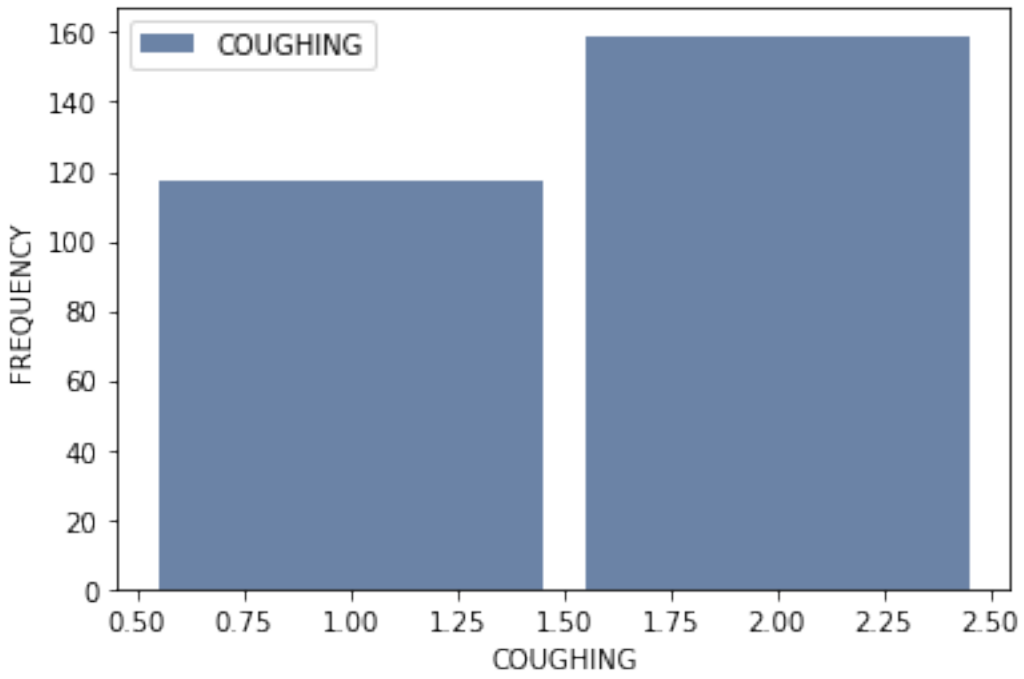
Checking for outliers: From the above histogram we see that the age group between 25 to 39 have no cases of having Lung Cancer.Between the age group of 50 to 79 we can see more number of persons having lung cancer.

[9]:
```
# 2. Plotting the histogram for 'SMOKING' variable from the LUNG CANCER dataset.
hist = thinkstats2.Hist(LCdata.SMOKING, label='SMOKING')
thinkplot.Hist(hist)
thinkplot.Show(xlabel='SMOKING', ylabel='FREQUENCY')
```



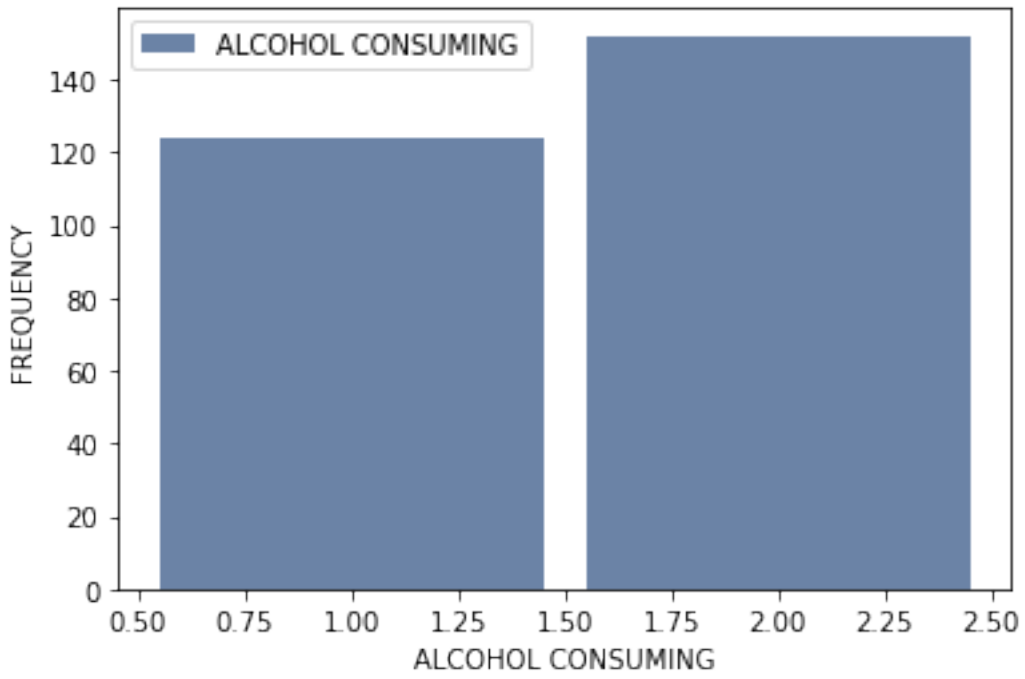

```
<Figure size 576x432 with 0 Axes>
```

[10]:
```
# 3. Plotting the histogram for 'COUGHING' variable from the LUNG CANCER␣
 ↪dataset.
#thinkplot.figure(figsize=(20,10))
hist = thinkstats2.Hist(LCdata.COUGHING, label='COUGHING')
thinkplot.Hist(hist)

thinkplot.Show(xlabel='COUGHING', ylabel='FREQUENCY')
```



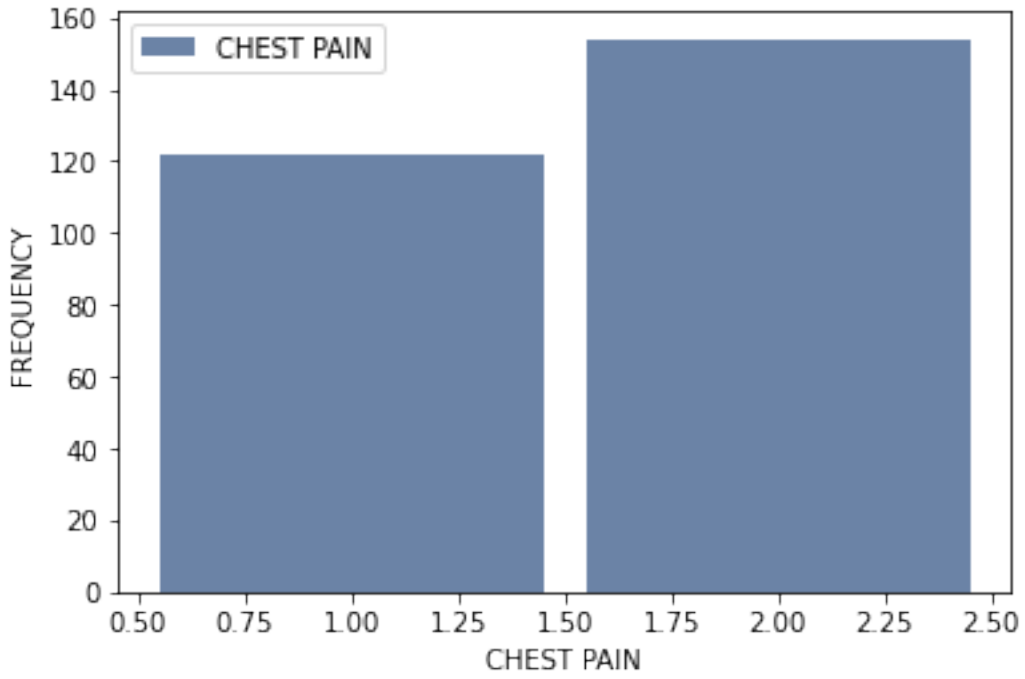

```
<Figure size 576x432 with 0 Axes>
```

[11]:
```
#4. Plotting the histogram for 'ALCOHOL CONSUMING' variable from the LUNG␣
 ↪CANCER dataset.
hist = thinkstats2.Hist(LCdata['ALCOHOL CONSUMING'], label='ALCOHOL CONSUMING')
thinkplot.Hist(hist)
thinkplot.Show(xlabel='ALCOHOL CONSUMING', ylabel='FREQUENCY')
```




```
<Figure size 576x432 with 0 Axes>
```

[12]:
```
#5. Plotting the histogram for 'CHEST PAIN' variable from the LUNG CANCER␣
 ↪dataset.
hist = thinkstats2.Hist(LCdata['CHEST PAIN'], label='CHEST PAIN')
thinkplot.Hist(hist)
thinkplot.Show(xlabel='CHEST PAIN', ylabel='FREQUENCY')
```




```
<Figure size 576x432 with 0 Axes>
```

Step 4: Include the other descriptive characteristics about the variables: Mean, Mode, Spread, and Tails (Chapter 2).

1. AGE
2. SMOKING
3. CHOUGHING
4. ALCOHOL CONSUMING
5. CHEST PAIN

[13]:
```
#1. Descriptive characteristics of the variable 'AGE' is as below:
AGE_mean = LCdata.AGE.mean()
AGE_var  = LCdata.AGE.var()
AGE_std  = LCdata.AGE.std()
print(f"1.Mean of 'AGE' variable from the LUNG CANCER dataset : {AGE_mean}")
print(f"2.Variance of 'AGE' variable from the LUNG CANCER dataset: {AGE_var}")
print(f"3.Standard deviation of 'AGE' variable from the LUNG CANCER dataset:␣
 ↪{AGE_std}")
```

```
1.Mean of 'AGE' variable from the LUNG CANCER dataset : 62.90942028985507
2.Variance of 'AGE' variable from the LUNG CANCER dataset: 70.21358366271411
3.Standard deviation of 'AGE' variable from the LUNG CANCER dataset:
8.37935460896089
```

[14]:
```
#2. Descriptive characteristics of the variable 'SMOKING' is as below:
SMOKING_mean = LCdata.SMOKING.mean()
SMOKING_var  = LCdata.SMOKING.var()
SMOKING_std  = LCdata.SMOKING.std()
print(f"1.Mean of 'SMOKING' variable from the LUNG CANCER dataset :␣
 ↪{SMOKING_mean}")
print(f"2.Variance of 'SMOKING' variable from the LUNG CANCER dataset:␣
 ↪{SMOKING_var}")
print(f"3.Standard deviation of 'SMOKING' variable from the LUNG CANCER dataset:
 ↪ {SMOKING_std}")
```

```
1.Mean of 'SMOKING' variable from the LUNG CANCER dataset : 1.5434782608695652
2.Variance of 'SMOKING' variable from the LUNG CANCER dataset:
0.2490118577075088
3.Standard deviation of 'SMOKING' variable from the LUNG CANCER dataset:
0.49901087934784427
```

[15]:
```
#3.Descriptive characteristics of the variable 'COUGHING' is as below:
COUGHING_mean = LCdata.COUGHING.mean()
COUGHING_var  = LCdata.COUGHING.var()
COUGHING_std  = LCdata.COUGHING.std()
print(f"1.Mean of 'COUGHING' variable from the LUNG CANCER dataset :␣
 ↪{COUGHING_mean}")
print(f"2.Variance of 'COUGHING' variable from the LUNG CANCER dataset:␣
 ↪{COUGHING_var}")
print(f"3.Standard deviation of 'COUGHING' variable from the LUNG CANCER␣
 ↪dataset: {COUGHING_std}")
```

```
1.Mean of 'COUGHING' variable from the LUNG CANCER dataset : 1.576086956521739
2.Variance of 'COUGHING' variable from the LUNG CANCER dataset:
0.24509881422924815
3.Standard deviation of 'COUGHING' variable from the LUNG CANCER dataset:
0.49507455421304797
```

[16]:
```
#4.Descriptive characteristics of the variable 'ALCOHOL CONSUMING' is as below:
ALCOHOL_mean = LCdata['ALCOHOL CONSUMING'].mean()
ALCOHOL_var  = LCdata['ALCOHOL CONSUMING'].var()
ALCOHOL_std  = LCdata['ALCOHOL CONSUMING'].std()
print(f"1.Mean of 'ALCOHOL CONSUMING' variable from the LUNG CANCER dataset :␣
 ↪{ALCOHOL_mean}")
print(f"2.Variance of 'ALCOHOL CONSUMING' variable from the LUNG CANCER dataset:
 ↪ {ALCOHOL_var}")
print(f"3.Standard deviation of 'ALCOHOL CONSUMING' variable from the LUNG␣
 ↪CANCER dataset: {ALCOHOL_std}")
```

```
1.Mean of 'ALCOHOL CONSUMING' variable from the LUNG CANCER dataset :
1.5507246376811594
2.Variance of 'ALCOHOL CONSUMING' variable from the LUNG CANCER dataset:
```

```
0.24832674571805027
3.Standard deviation of 'ALCOHOL CONSUMING' variable from the LUNG CANCER
dataset: 0.4983239365292924
```

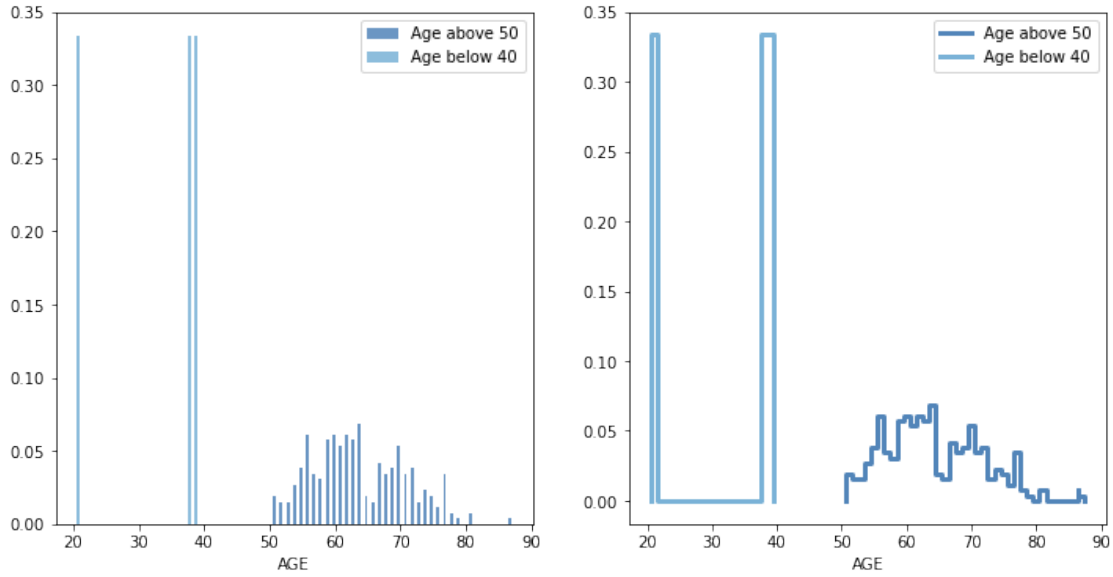
[17]:
```
# 5. Descriptive characteristics of the variable 'CHEST PAIN' is as below:
CP_mean = LCdata['CHEST PAIN'] .mean()
CP_var  = LCdata['CHEST PAIN'].var()
CP_std  = LCdata['CHEST PAIN'].std()
print(f"1.Mean of 'CHEST PAIN' variable from the LUNG CANCER dataset :␣
 ↪{CP_mean}")
print(f"2.Variance of 'CHEST PAIN' variable from the LUNG CANCER dataset:␣
 ↪{CP_var}")
print(f"3.Standard deviation of 'CHEST PAIN' variable from the LUNG CANCER␣
 ↪dataset: {CP_std}")
```

```
1.Mean of 'CHEST PAIN' variable from the LUNG CANCER dataset :
1.5579710144927537
2.Variance of 'CHEST PAIN' variable from the LUNG CANCER dataset:
0.24753623188405766
3.Standard deviation of 'CHEST PAIN' variable from the LUNG CANCER dataset:
0.49753013163431387
```

Step 5: Using pg. 29 of your text as an example, compare two scenarios in your data using a PMF. Reminder, this isn't comparing two variables against each other – it is the same variable, but a different scenario. Almost like a filter. The example in the book is first babies compared to all other babies, it is still the same variable, but breaking the data out based on criteria we are exploring (Chapter 3).

[18]:
```
# Selecting the width
width = 0.45

# For age more at risk of having lung cancer is selcted by the age group␣
 ↪greater than 50 years.
agemore = LCdata[LCdata.AGE > 50]
# For age less at risk of having lung cancer is selected by the age group less␣
 ↪than 40 years.
ageless = LCdata[LCdata.AGE < 40]

# Plotting the PMF's individually for ideal and nonideal cases.
agemore_pmf = thinkstats2.Pmf(agemore.AGE, label='Age above 50')
ageless_pmf = thinkstats2.Pmf(ageless.AGE, label='Age below 40')

thinkplot.PrePlot(2, cols=2)
thinkplot.Hist(agemore_pmf, align='right',width = width)
thinkplot.Hist(ageless_pmf, align='right',width= width)
thinkplot.Config(xlabel = 'AGE', ylable = "PMF")

thinkplot.PrePlot(2)
```

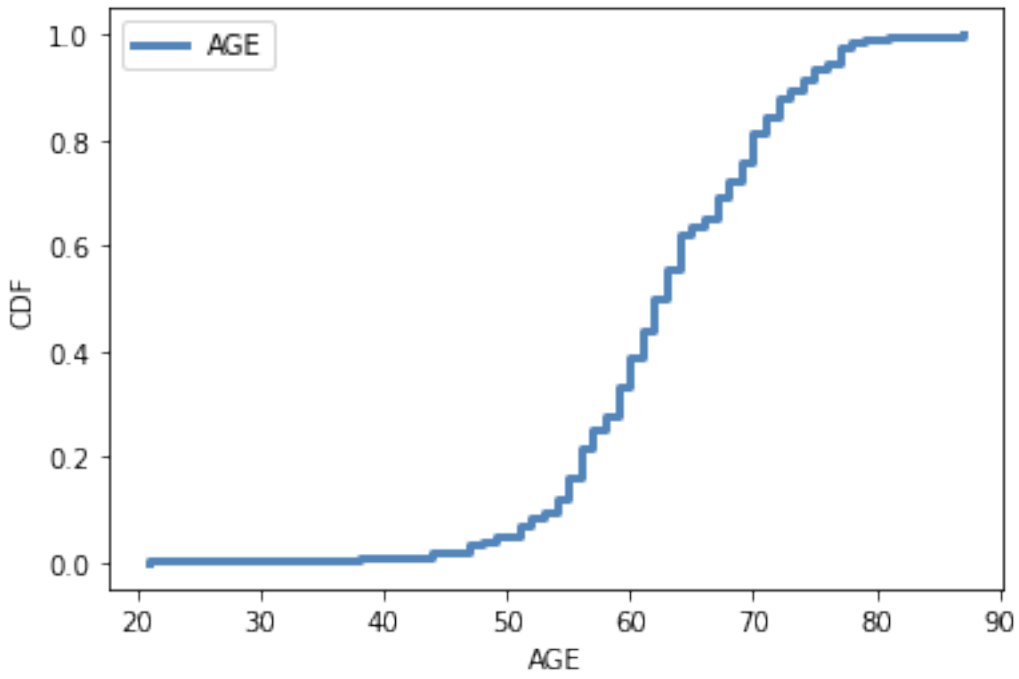
```
thinkplot.SubPlot(2)
thinkplot.Pmfs([agemore_pmf,ageless_pmf])
thinkplot.Show(xlabel='AGE')
```




```
<Figure size 576x432 with 0 Axes>
```

Step 6 : Create 1 CDF with one of your variables, using page 41-44 as your guide, what does this tell you about your variable and how does it address the question you are trying to answer (Chapter 4).

[19]:
```
agecdf = thinkstats2.Cdf(LCdata.AGE, label='AGE')
thinkplot.PrePlot(2)
thinkplot.Cdf(agecdf)
thinkplot.Show(xlabel='AGE', ylabel='CDF')
```



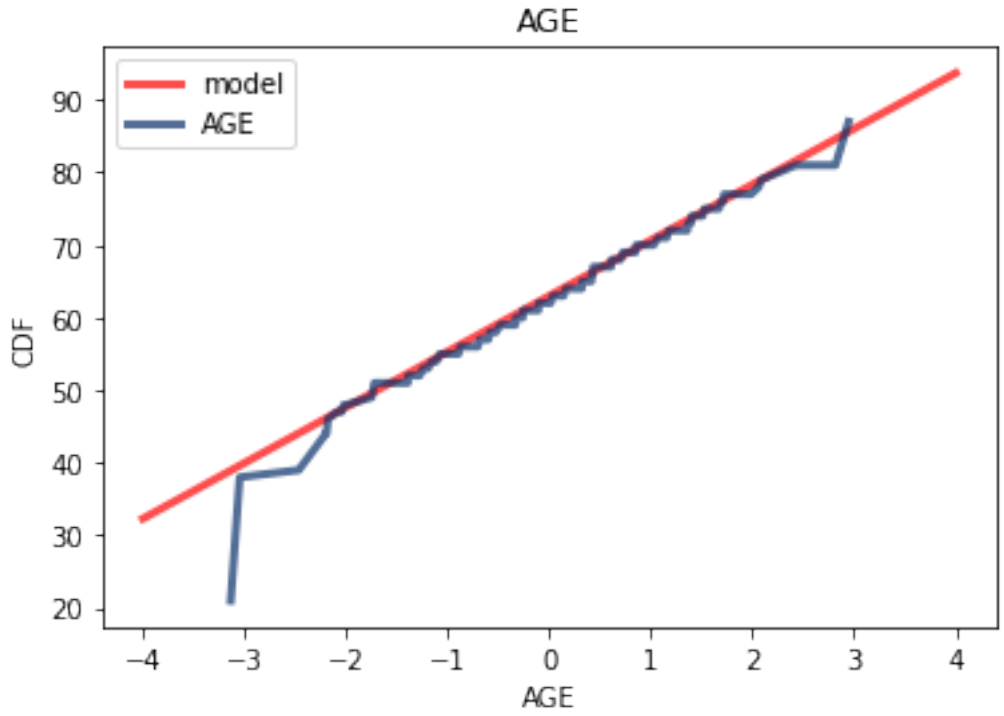

```
<Figure size 576x432 with 0 Axes>
```

1. CDF(Cumulative Distribution Factor): This is a function that maps from a value to its percentile rank.
2. Here the above CDF is plotted for AGE variable.
3. From the above CDF plot the common values appear as steep or vertical sections of the CDF plot.
4. Here from the above plot the mode at age 65 years is apparent.
5. The CDF range is flat for ages (20 to 40) and (80 to 90).

Step 7: Plot 1 analytical distribution and provide your analysis on how it applies to the dataset you have chosen (Chapter 5).

[20]:
```
# Function for normal Probability plot
def MakeNormalPlot(age):
    mean, var = thinkstats2.TrimmedMeanVar(age, p=0.01)
    std = np.sqrt(var)

    xs = [-4,4]
    xs, ys = thinkstats2.FitLine(xs, mean, std)
    thinkplot.Plot(xs, ys, color='red', label='model')

    xs, ys = thinkstats2.NormalProbability(age)
    thinkplot.Plot(xs, ys, label='AGE')
```

[21]:
```
age = LCdata.AGE.dropna()
MakeNormalPlot(age)
thinkplot.Config(title='AGE', xlabel='AGE',
                 ylabel='CDF', loc='upper left')
```



1. From the above plot we can see that both the curves match the model approximately near the mean and deviate in the tails.
2. The people aged below are at less risk in getting lung cancer and the people aged above 50 to 75 are at high risk in getting lung cancer.
3. When we select the maximum possible age group we can remove the people aged below 40 , hence reducing the discrepancy in the lower tail of the distribution.
4. With above plot we can suggest that the normal model describes the distribution well within a few standard deviations from the mean , but not in the tails.
5. For practical purposes if this model best fits or not depends on the purpose.

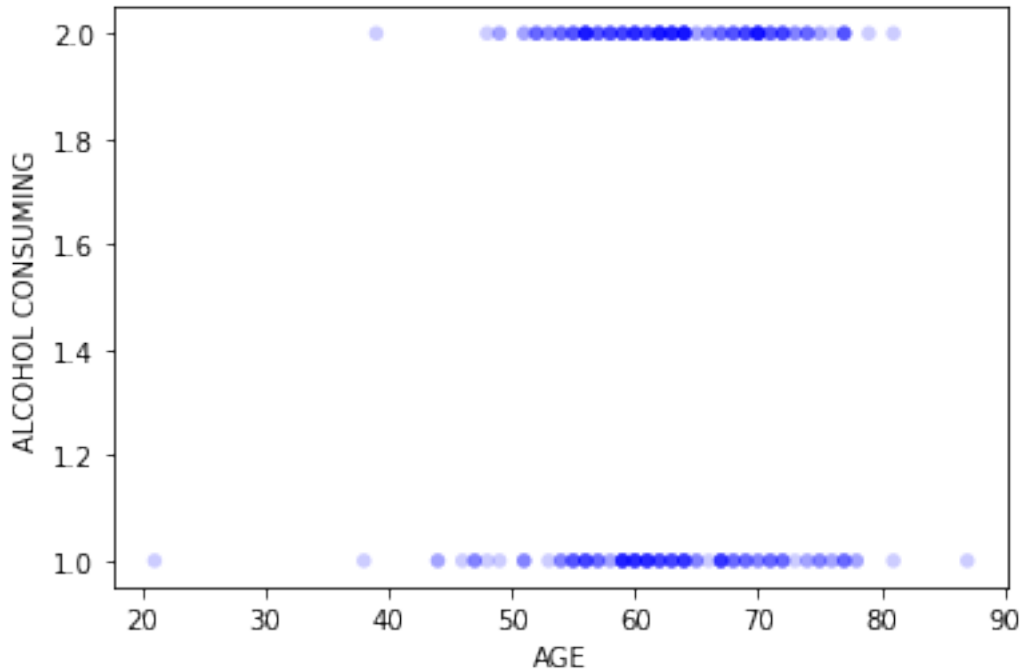
Step 8: Create two scatter plots comparing two variables and provide your analysis on correlation and causation. Remember, covariance, Pearson's correlation, and Non-Linear Relationships should also be considered during your analysis (Chapter 7).

[22]:
```
#Function for computing Covariance.

def Cov(xs, ys, meanx=None, meany=None):
    xs = np.asarray(xs)
    ys = np.asarray(ys)
```

```
    if meanx is None:
        meanx = np.mean(xs)
    if meany is None:
        meany = np.mean(ys)

    cov = np.dot(xs-meanx, ys-meany) / len(xs)
    return cov
```

[23]:
```
# Function for computing Pearson's Correlation.
def Corr(xs, ys):
    xs = np.asarray(xs)
    ys = np.asarray(ys)

    meanx, varx = thinkstats2.MeanVar(xs)
    meany, vary = thinkstats2.MeanVar(ys)

    corr = Cov(xs, ys, meanx, meany) / np.sqrt(varx * vary)
    return corr
```

[24]:
```
# 1. Scatter plot for AGE vs ALCOHOL CONSUMING
alcohol = LCdata['ALCOHOL CONSUMING']
age = LCdata.AGE
thinkplot.Scatter(age,alcohol)
thinkplot.Show(xlabel = 'AGE', ylabel = 'ALCOHOL CONSUMING')
```



```
<Figure size 576x432 with 0 Axes>
```

From the above scatter plot it is clear that there is a linear relation between the variables 'AGE' and 'ALCOHOL CONSUMING'.

[25]:
```
covariance = Cov(age,alcohol)
print(f" Covariance of AGE and ALCOHOL CONSUMING variables from the dataset :␣
 ↪{covariance}")
```

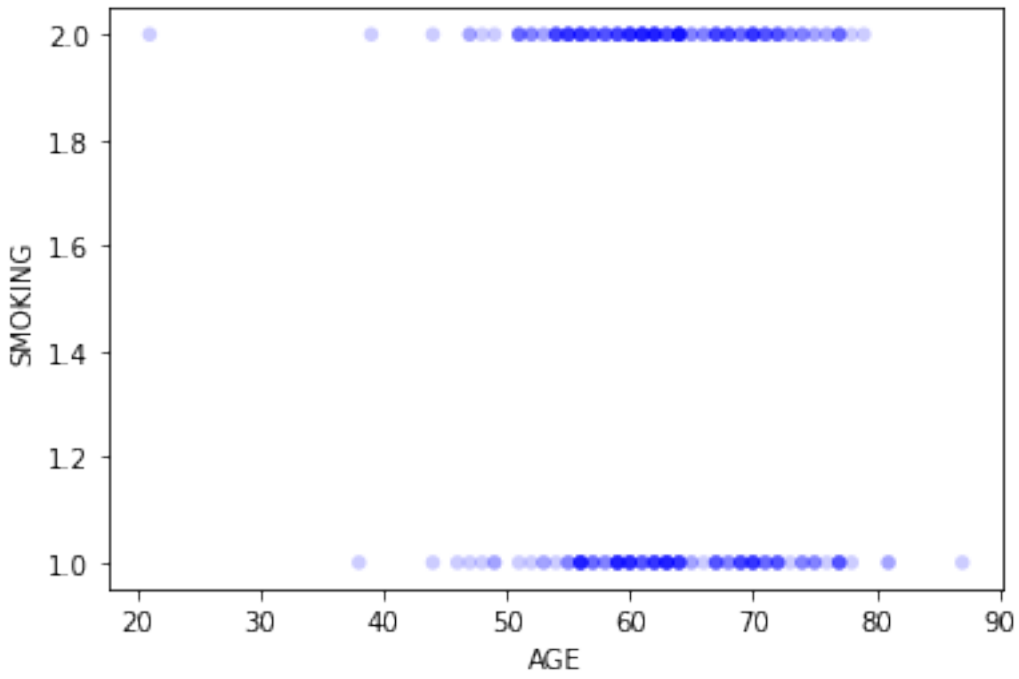
```
 Covariance of AGE and ALCOHOL CONSUMING variables from the dataset :
0.21655114471749642
```

[26]:
```
pcorr = Corr(age, alcohol)
print(f" Pearson's correlation coefficient for AGE and ALCOHOL CONSUMING␣
 ↪variables from the dataset: {pcorr}")
```

```
 Pearson's correlation coefficient for AGE and ALCOHOL CONSUMING variables from
the dataset: 0.05204925930094622
```

1. Covariance measures the tendency of the two variables to vary together.
2. Covariance is maximized if the two vectors are identical, 0 if they are orthogonal, and negative if they point in opposite directions.
3. Solution : From the above result we see that the covariance of the two variables 'AGE' and 'ALCOHOL CONSUMING' are identical, i.e. age group 50 to 80 consume alcohol.
4. Pearson's Correlation is always between -1 and +1(including both).
   a. If Pearson's Correlation is positive , we say that the correlation is positive, i.e. when one variable is high , the other tends to be high.
   b. If Pearson's Correlation is negative, we say that the correlation is negative , i.e. when one variable is high , the other is low.
5. Solution : From the above result it is clear that the correlation between the AGE and ALCOHOL CONSUMING variables is positive.

[27]:
```
#2. Scatter plot for AGE vs SMOKING
age = LCdata.AGE
smoke = LCdata.SMOKING
thinkplot.Scatter(age , smoke)
thinkplot.Show(xlabel = 'AGE', ylabel = 'SMOKING')
```




```
<Figure size 576x432 with 0 Axes>
```

[28]:
```
covariance = Cov(age,smoke)
print(f" Covariance of AGE and SMOKING variables from the dataset :␣
 ↪{covariance}")
```

```
 Covariance of AGE and SMOKING variables from the dataset : -0.3058443604284815
```

[29]:
```
pcorr = Corr(age, smoke)
print(f" Pearson's correlation coefficient for AGE and SMOKING variables from␣
 ↪the dataset: {pcorr}")
```

```
 Pearson's correlation coefficient for AGE and SMOKING variables from the
dataset: -0.07341017865904706
```

1. Covariance measures the tendency of the two variables to vary together.
2. Covariance is maximized if the two vectors are identical, 0 if they are orthogonal, and negative if they point in opposite directions.
3. Solution : From the above result we see that the covariance of the two variables 'AGE' and 'SMOKING' is negative , i.e.they point in opposite direction,i.e age groups 20 SMOKE and may not have LUNG CANCER and the age gropus above 90 may not smoke but have LUNG CANCER .
4. Pearson's Correlation is always between -1 and +1(including both).
   a. If Pearson's Correlation is positive , we say that the correlation is positive, i.e. when one variable is high , the other tends to be high.
   b. If Pearson's Correlation is negative, we say that the correlation is negative , i.e. when one variable is high , the other is low.

5. Solution : From the above result it is clear that the correlation between the 'AGE' and 'SMOKING' variables is negative.

Step 9 :Conduct a test on your hypothesis using one of the methods covered in Chapter 9.

[30]:
```
class HypothesisTest(object):

    def __init__(self, data):
        self.data = data
        self.MakeModel()
        self.actual = self.TestStatistic(data)

    def PValue(self, iters=1000):
        self.test_stats = [self.TestStatistic(self.RunModel())
                           for _ in range(iters)]

        count = sum(1 for x in self.test_stats if x >= self.actual)
        return count / iters

    def TestStatistic(self, data):
        raise UnimplementedMethodException()

    def MakeModel(self):
        pass

    def RunModel(self):
        raise UnimplementedMethodException()
```

[31]:
```
class CorrelationPermute(HypothesisTest):

    def TestStatistic(self, data):
        xs, ys = data
        test_stat = abs(thinkstats2.Corr(xs, ys))
        return test_stat

    def RunModel(self):
        xs, ys = self.data
        xs = np.random.permutation(xs)
        return xs, ys
```

[36]:
```
testdata = LCdata.SMOKING, LCdata.AGE
ht = CorrelationPermute(testdata)
pvalue = ht.PValue()
pvalue
```

[36]: 0.246

Step 10 : For this project, conduct a regression analysis on either one dependent and one explanatory variable, or multiple explanatory variables (Chapter 10 & 11).

[33]:
```
import statsmodels.formula.api as smf
# As the LUNG_CANCER variable has non-numeric values, mapping them with numeric␣
 ↪values.
LCdata['LUNG_CANCER'] = LCdata['LUNG_CANCER'].map({'NO': 1, 'YES': 2})
# regression analysis.
formula = 'LUNG_CANCER ~ SMOKING'
model = smf.ols(formula, data=LCdata)
results = model.fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:            LUNG_CANCER   R-squared:                       0.001
Model:                            OLS   Adj. R-squared:                 -0.002
Method:                 Least Squares   F-statistic:                    0.3337
Date:                Wed, 10 Aug 2022   Prob (F-statistic):              0.564
Time:                        12:42:32   Log-Likelihood:                -97.389
No. Observations:                 276   AIC:                             198.8
Df Residuals:                     274   BIC:                             206.0
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      1.8251      0.068     26.944      0.000       1.692       1.958
SMOKING        0.0241      0.042      0.578      0.564      -0.058       0.106
==============================================================================
Omnibus:                      113.363   Durbin-Watson:                   1.882
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              269.891
Skew:                          -2.099   Prob(JB):                     2.48e-59
Kurtosis:                       5.417   Cond. No.                         7.15
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

[35]:
```
# Regression analysis for multiple variables
ALCOHOL = LCdata['ALCOHOL CONSUMING']
formula = 'LUNG_CANCER ~ AGE + SMOKING + ALCOHOL'
model = smf.ols(formula, data=LCdata)
results = model.fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
```

```
Dep. Variable:            LUNG_CANCER   R-squared:                       0.098
Model:                            OLS   Adj. R-squared:                  0.088
Method:                 Least Squares   F-statistic:                     9.878
Date:                Wed, 10 Aug 2022   Prob (F-statistic):           3.33e-06
Time:                        12:52:20   Log-Likelihood:                -83.286
No. Observations:                 276   AIC:                             174.6
Df Residuals:                     272   BIC:                             189.1
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      1.2400      0.177      7.012      0.000       0.892       1.588
AGE            0.0039      0.002      1.649      0.100      -0.001       0.009
SMOKING        0.0396      0.040      0.991      0.322      -0.039       0.118
ALCOHOL        0.2026      0.040      5.067      0.000       0.124       0.281
==============================================================================
Omnibus:                       91.855   Durbin-Watson:                   1.980
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              185.731
Skew:                          -1.768   Prob(JB):                     4.67e-41
Kurtosis:                       4.910   Cond. No.                         570.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

[ ]: