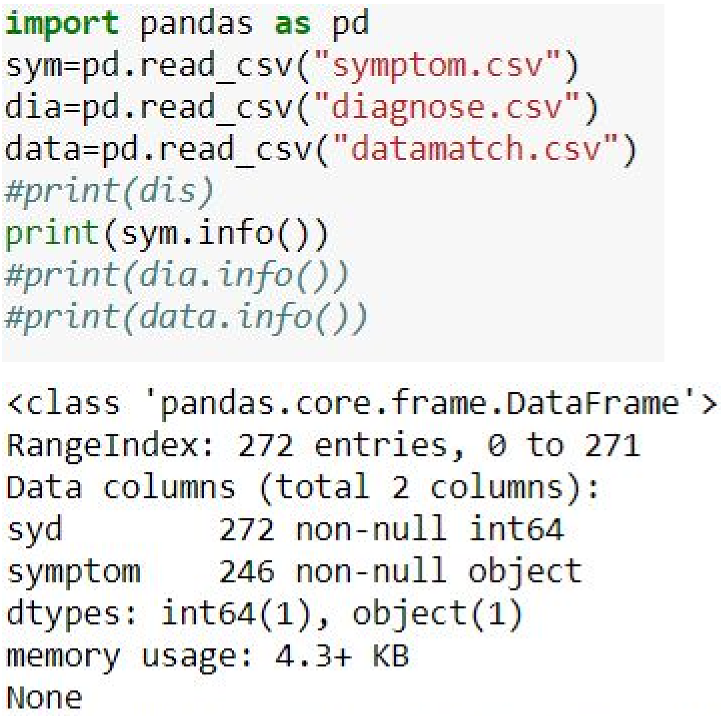
Analysis Of Diseases and Symptoms

Project Objective : To analyse the Symptoms of a Disease and is Diagnostics.
The dataset provided to us helps us to analyse the symptoms of a disease and helps us to diagnose it and tells us how fatal the disease is on a scale of 0 to 3(0 being the least fatal and 3 being most)

Description of Data Set columns :

Three Databases are used in this project :-
Dataset1 (*symptom*) :

- syd:>Symptom ID:
  Gives us the unique Identification number of a specific symptom
  Datatype:int64
  Non-null values:272
- symptom:>Tells us the possible symptoms of all diseases
  Datatype:object
  Non-null values:246



```
import pandas as pd
sym=pd.read_csv("symptom.csv")
dia=pd.read_csv("diagnose.csv")
data=pd.read_csv("datamatch.csv")
#print(dis)
print(sym.info())
#print(dia.info())
#print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 272 entries, 0 to 271
Data columns (total 2 columns):
syd        272 non-null int64
symptom    246 non-null object
dtypes: int64(1), object(1)
memory usage: 4.3+ KB
None
```

Dataset2 (*diagnose*) :

- did:>Disease ID .Gives us the unique Identification number of a specific disease

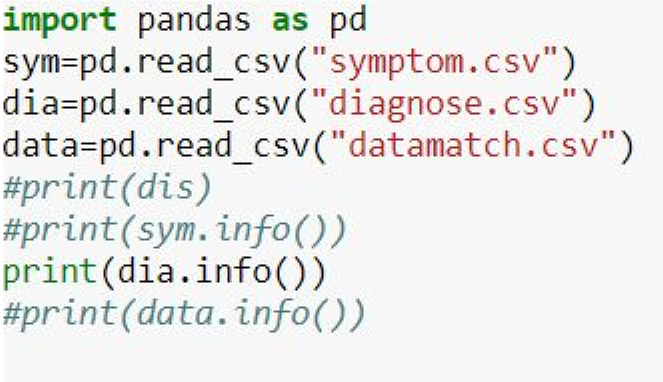
Datatype:int64

Non-null values:1166

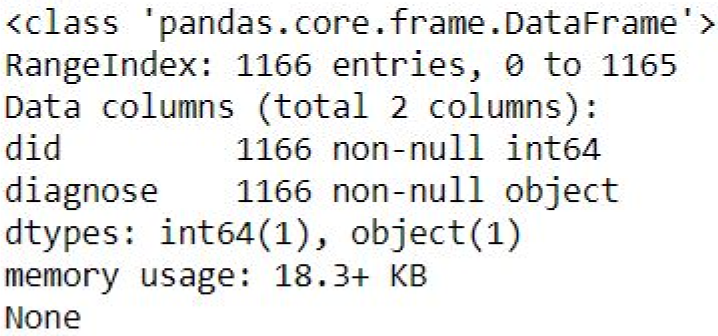
- diagnosis:>Tells us the possible diseases after diagnosing the symptoms.

  Datatype:object

  Non-null values:1166



```python
import pandas as pd
sym=pd.read_csv("symptom.csv")
dia=pd.read_csv("diagnose.csv")
data=pd.read_csv("datamatch.csv")
#print(dis)
#print(sym.info())
print(dia.info())
#print(data.info())
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1166 entries, 0 to 1165
Data columns (total 2 columns):
did         1166 non-null int64
diagnose    1166 non-null object
dtypes: int64(1), object(1)
memory usage: 18.3+ KB
None
```

Dataset3 (*datamatch*) :

- syd:>Symptom ID .Gives us the unique Identification number of a specific symptom

  Datatype:int64

  Non-null values:5568

- did:>Disease ID .Gives us the unique Identification number of a specific disease

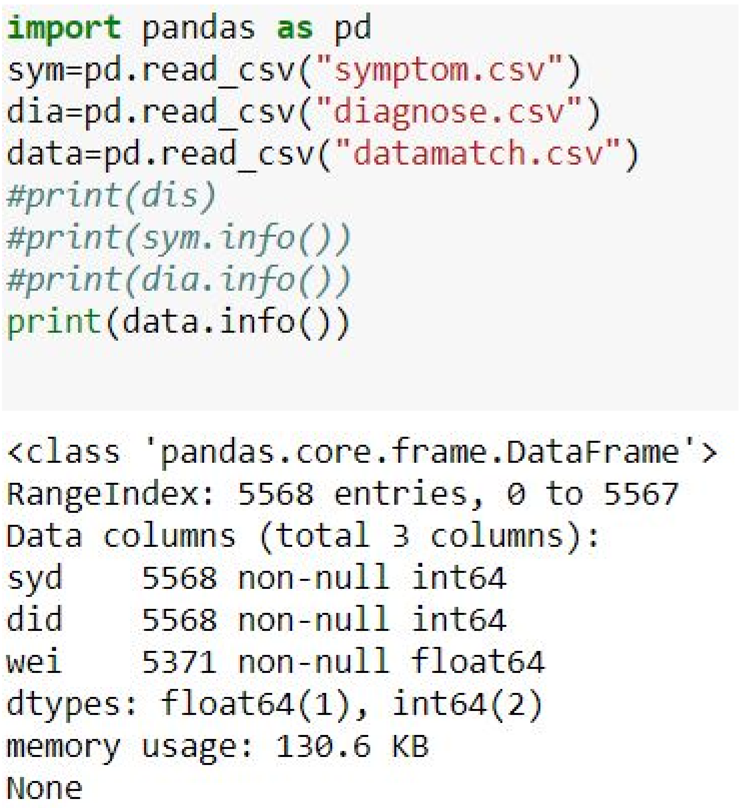
  Datatype:int64

  Non-null values:5568

- wei:>how possible the fatal a disease can be depending upon the lives it claims annually

  Datatype:float64

  Non-null values:5371

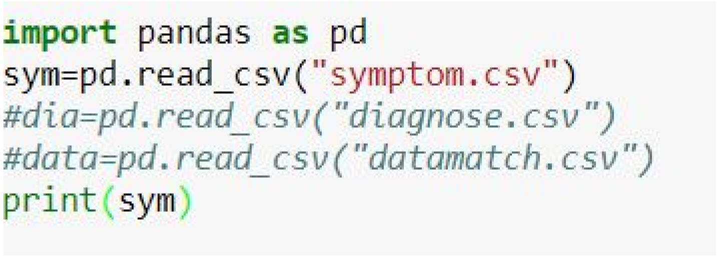


```
import pandas as pd
sym=pd.read_csv("symptom.csv")
dia=pd.read_csv("diagnose.csv")
data=pd.read_csv("datamatch.csv")
#print(dis)
#print(sym.info())
#print(dia.info())
print(data.info())
```

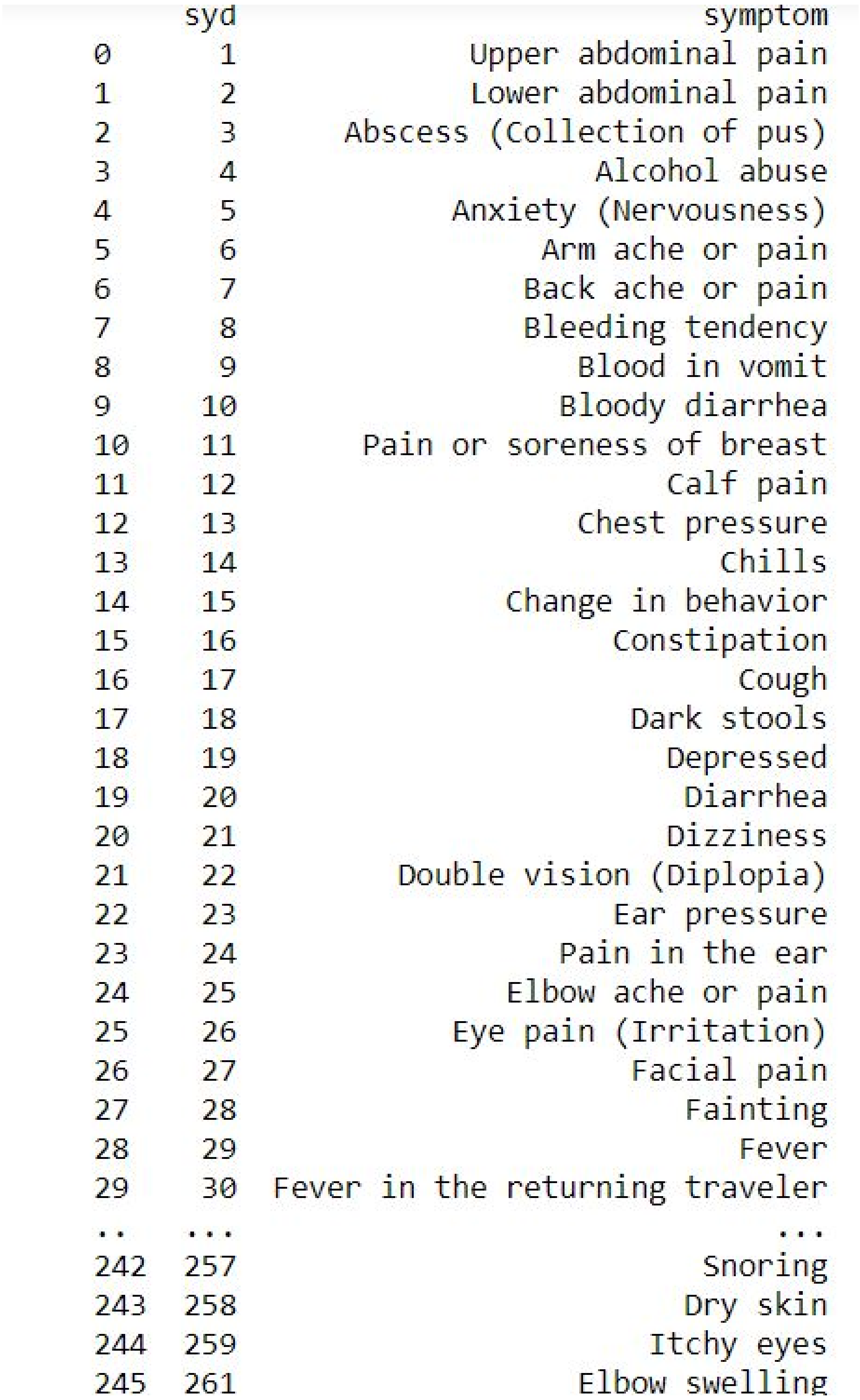
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5568 entries, 0 to 5567
Data columns (total 3 columns):
syd    5568 non-null int64
did    5568 non-null int64
wei    5371 non-null float64
dtypes: float64(1), int64(2)
memory usage: 130.6 KB
None
```

Data Interpretation :-

Dataset1 (*symptom*) :

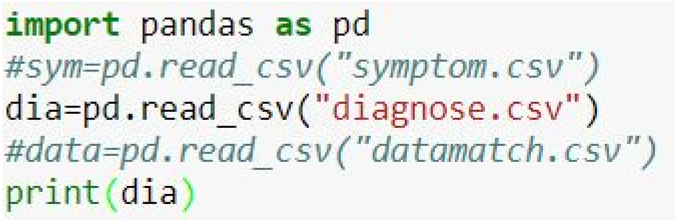


```
import pandas as pd
sym=pd.read_csv("symptom.csv")
#dia=pd.read_csv("diagnose.csv")
#data=pd.read_csv("datamatch.csv")
print(sym)
```

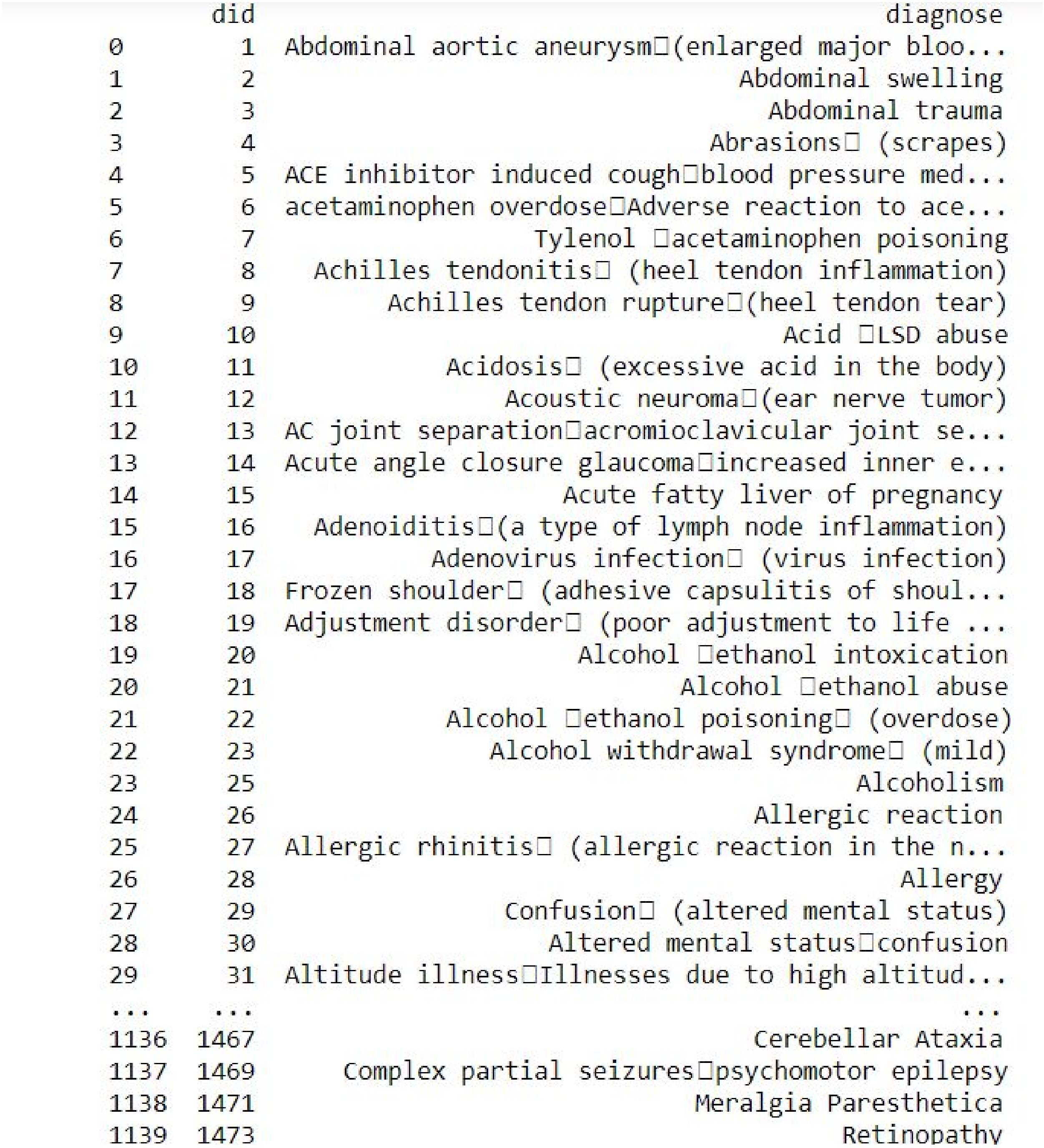


```
     syd                         symptom
0      1            Upper abdominal pain
1      2            Lower abdominal pain
2      3     Abscess (Collection of pus)
3      4                   Alcohol abuse
4      5           Anxiety (Nervousness)
5      6                Arm ache or pain
6      7               Back ache or pain
7      8               Bleeding tendency
8      9                  Blood in vomit
9     10                 Bloody diarrhea
10    11      Pain or soreness of breast
11    12                       Calf pain
12    13                  Chest pressure
13    14                          Chills
14    15              Change in behavior
15    16                    Constipation
16    17                           Cough
17    18                     Dark stools
18    19                       Depressed
19    20                        Diarrhea
20    21                       Dizziness
21    22        Double vision (Diplopia)
22    23                    Ear pressure
23    24                 Pain in the ear
24    25              Elbow ache or pain
25    26           Eye pain (Irritation)
26    27                     Facial pain
27    28                       Fainting
28    29                           Fever
29    30  Fever in the returning traveler
..   ...                             ...
242  257                         Snoring
243  258                        Dry skin
244  259                      Itchy eyes
245  261                  Elbow swelling
```

Dataset2 (*diagnose*) :

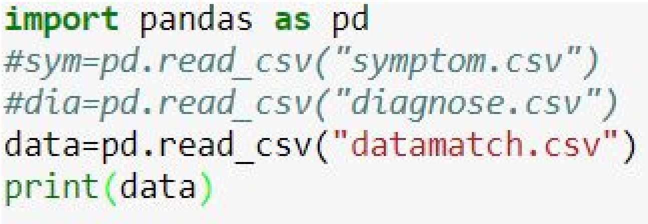


```python
import pandas as pd
#sym=pd.read_csv("symptom.csv")
dia=pd.read_csv("diagnose.csv")
#data=pd.read_csv("datamatch.csv")
print(dia)
```

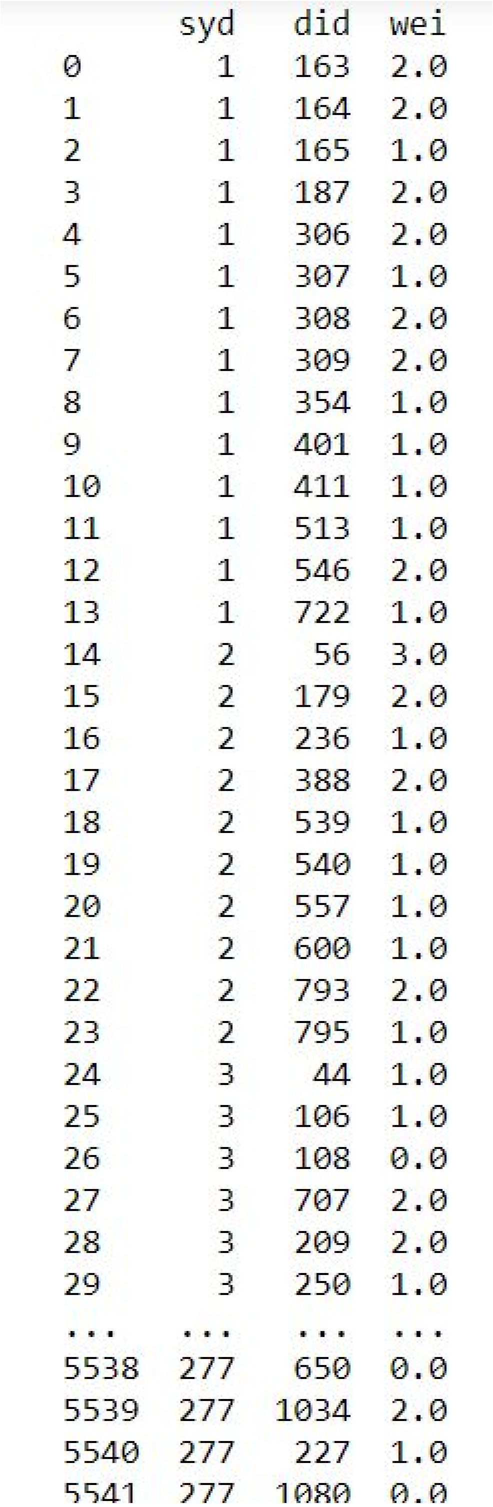


```
       did                                          diagnose
0        1  Abdominal aortic aneurysm□(enlarged major bloo...
1        2                                Abdominal swelling
2        3                                  Abdominal trauma
3        4                              Abrasions□ (scrapes)
4        5  ACE inhibitor induced cough□blood pressure med...
5        6  acetaminophen overdose□Adverse reaction to ace...
6        7                  Tylenol □acetaminophen poisoning
7        8   Achilles tendonitis□ (heel tendon inflammation)
8        9        Achilles tendon rupture□(heel tendon tear)
9       10                                   Acid □LSD abuse
10      11            Acidosis□ (excessive acid in the body)
11      12                Acoustic neuroma□(ear nerve tumor)
12      13  AC joint separation□acromioclavicular joint se...
13      14  Acute angle closure glaucoma□increased inner e...
14      15                    Acute fatty liver of pregnancy
15      16   Adenoiditis□(a type of lymph node inflammation)
16      17           Adenovirus infection□ (virus infection)
17      18  Frozen shoulder□ (adhesive capsulitis of shoul...
18      19  Adjustment disorder□ (poor adjustment to life ...
19      20                     Alcohol □ethanol intoxication
20      21                            Alcohol □ethanol abuse
21      22            Alcohol □ethanol poisoning□ (overdose)
22      23               Alcohol withdrawal syndrome□ (mild)
23      25                                        Alcoholism
24      26                                 Allergic reaction
25      27  Allergic rhinitis□ (allergic reaction in the n...
26      28                                           Allergy
27      29                Confusion□ (altered mental status)
28      30                   Altered mental status□confusion
29      31  Altitude illness□Illnesses due to high altitud...
...    ...                                               ...
1136  1467                                 Cerebellar Ataxia
1137  1469     Complex partial seizures□psychomotor epilepsy
1138  1471                             Meralgia Paresthetica
1139  1473                                       Retinopathy
```

Dataset3 (*datamatch*) :



```
import pandas as pd
#sym=pd.read_csv("symptom.csv")
#dia=pd.read_csv("diagnose.csv")
data=pd.read_csv("datamatch.csv")
print(data)
```



```
       syd   did  wei
0        1   163  2.0
1        1   164  2.0
2        1   165  1.0
3        1   187  2.0
4        1   306  2.0
5        1   307  1.0
6        1   308  2.0
7        1   309  2.0
8        1   354  1.0
9        1   401  1.0
10       1   411  1.0
11       1   513  1.0
12       1   546  2.0
13       1   722  1.0
14       2    56  3.0
15       2   179  2.0
16       2   236  1.0
17       2   388  2.0
18       2   539  1.0
19       2   540  1.0
20       2   557  1.0
21       2   600  1.0
22       2   793  2.0
23       2   795  1.0
24       3    44  1.0
25       3   106  1.0
26       3   108  0.0
27       3   707  2.0
28       3   209  2.0
29       3   250  1.0
...    ...   ...  ...
5538   277   650  0.0
5539   277  1034  2.0
5540   277   227  1.0
5541   277  1080  0.0
```

Analysis performed on the basis of various parameters :

1. Perform analysis on diseases which are most untraceable

2. Perform analysis on the drug related diseases and their fatality

3. Perform analysis to find the easiest Drug Related Disease

4. Perform analysis to find the most curable cancer

5. Perform analyses to find the most fatal cancer

6. Perform analysis to find the top 10 Severe cancer type.

7. Perform analysis to find breast related diseases with their specific symptoms

8. Perform analysis to find the number of diseases whose fatality is 1 or 2
9. Perform analysis on the fact to find possible diseases if the symptoms are delusions and hallucinations.

10. Perform analysis on the fact that diseases which have symptoms of both headache and fever

11. Perform analysis on the fact that Showing, the diseases based on fatality rate

12. Perform analysis on the fact that the number of Bacterial and cervical diseases and their fatality rate
13. Perform analysis on the fact that the symptoms related to eye (irritation) and has a fatality rate equal to 3
14. Perform analysis on the fact that what is the chance of dying from cold symptoms(Graph)
15. Perform analysis on the fact that Diseases with more than twenty five symptoms
16. Perform analysis on the fact that If delusions and hallucinations occur in a patient then it leads to mental illness

17. Perform analysis on the fact that most common symptoms found in the diseases of highest fatality

18. Perform analysis on the fact that fatality rate of diseases which have symptoms of vomiting

19. Perform analysis on the fact that the symptoms related to different types of organ failures and has a fatality rate equal to 3

20. Perform analysis to find the occurrence of symptoms of highest fatality

21. Perform analysis on the fact that diseases with symptoms of upper abdominal pain whose fatality is greater than 2

22. Perform analysis to calculate the Death percentage in top 3 Diseases.

Correlation data on the basis of various parameters(pre-data cleaning) :

1. Perform analysis on diseases which are most untraceable
    - ❖ Diseases vs symptom id: 0.6

2. Perform analysis on the drug related diseases and their fatality
    - ❖ Diseases vs fatality: -0.11

3. Perform analysis to find the easiest Drug Related Diseases
    - ❖ Diseases vs symptom id: 0.6

4. Perform analysis to find the most curable cancer
    - ❖ Diseases vs fatality: -0.11

5. Perform analyses to find the most fatal cancer
    - ❖ Diseases vs fatality: -0.11

6. Perform analysis to find the top 10 Severe cancer type.
    - ❖ Diseases vs fatality: -0.11

7. Perform analysis to find breast related diseases with their specific symptoms
    - ❖ Diseases vs symptom id: 0.6

8. Perform analysis to find the number of diseases whose fatality is 1 or 2
    - ❖ Disease ID vs fatality : -0.19

9. Perform analysis on the fact to find possible diseases if the symptoms are delusions and hallucinations.
    - ❖ Diseases vs symptoms: -0.03

10. Perform analysis on the fact that diseases which have symptoms of both headache and fever
    - ❖ Diseases vs symptoms: -0.03

11. Perform analysis on the fact that Showing, the diseases based on fatality rate
    - ❖ Diseases vs fatality: -0.11

12. Perform analysis on the fact that the number of Bacterial and cervical diseases and their fatality rate

- ❖ Diseases vs fatality: -0.11

13. Perform analysis on the fact that the symptoms related to eye (irritation) and has a fatality rate equal to 3

- ❖ Diseases vs fatality: -0.11

14. Perform analysis on the fact that what is the chance of dying from cold symptoms(Graph)

- ❖ Symptom vs fatality: -0.11

15. Perform analysis on the fact that Diseases with more than twenty five symptoms

- ❖ Diseases vs symptoms: -0.03

16. Perform analysis on the fact that If delusions and hallucinations occur in a patient then it leads to mental illness

- ❖ Diseases vs symptoms: -0.03

17. Perform analysis on the fact that most common symptoms found in the diseases of highest fatality

- ❖ Symptom vs fatality: -0.11

18. Perform analysis on the fact that fatality rate of diseases which have symptoms of vomiting

- ❖ Symptom vs fatality: -0.11

19. Perform analysis on the fact that the symptoms related to different types of organ failures and has a fatality rate equal to 3

- ❖ Disease ID vs fatality : -0.19

20. Perform analysis to find the occurrence of symptoms of highest fatality

- ❖ Symptom vs fatality: -0.11

21. Perform analysis on the fact that diseases with symptoms of upper abdominal pain whose fatality is greater than 2

- ❖ Symptom vs fatality: -0.11

22. Perform analysis to calculate the Death percentage in top 3 Diseases.

- ❖ Diseases vs fatality: -0.11

Correlation data on the basis of various parameters(post-data cleaning) :

1. Perform analysis on diseases which are most untraceable

- ❖ Diseases vs symptom id: 0.42

2. Perform analysis on the drug related diseases and their fatality

- ❖ Diseases vs fatality: -0.10

3. Perform analysis to find the easiest Drug Related Diseases

- ❖ Diseases vs symptom id: 0.42

4. Perform analysis to find the most curable cancer

- ❖ Diseases vs fatality: -0.10

5. Perform analyses to find the most fatal cancer

- ❖ Diseases vs fatality: -0.10

6. Perform analysis to find the top 10 Severe cancer type.

- ❖ Diseases vs fatality: -0.10

7. Perform analysis to find breast related diseases with their specific symptoms

- ❖ Diseases vs symptom id: 0.42

8. Perform analysis to find the number of diseases whose fatality is 1 or 2

- ❖ Disease ID vs fatality : -0.19

9. Perform analysis on the fact to find possible diseases if the symptoms are delusions and hallucinations.

- ❖ Diseases vs symptoms: -0.06

10. Perform analysis on the fact that diseases which have symptoms of both headache and fever

- ❖ Diseases vs symptoms: -0.06

11. Perform analysis on the fact that Showing, the diseases based on fatality rate

- ❖ Diseases vs fatality: -0.10

12 **.**Perform analysis on the fact that the number of Bacterial and cervical diseases and their fatality rate

- ❖ Diseases vs fatality: -0.10

13. Perform analysis on the fact that the symptoms related to eye (irritation) and has a fatality rate equal to 3

- ❖ Diseases vs fatality: -0.10

14. Perform analysis on the fact that what is the chance of dying from cold symptoms(Graph)

- ❖ Symptom vs fatality: -0.12

15. Perform analysis on the fact that Diseases with more than twenty five symptoms

- ❖ Diseases vs symptoms: -0.06

16. Perform analysis on the fact that If delusions and hallucinations occur in a patient then it leads to mental illness

- ❖ Diseases vs symptoms: -0.06

17. Perform analysis on the fact that most common symptoms found in the diseases of highest fatality

- ❖ Symptom vs fatality: -0.12

18. Perform analysis on the fact that fatality rate of diseases which have symptoms of vomiting

- ❖ Symptom vs fatality: -0.12

19. Perform analysis on the fact that the symptoms related to different types of organ failures and has a fatality rate equal to 3

- ❖ Disease ID vs fatality : -0.19

20. Perform analysis to find the occurrence of symptoms of highest fatality

- ❖ Symptom vs fatality: -0.12

21. Perform analysis on the fact that diseases with symptoms of upper abdominal pain whose fatality is greater than 2

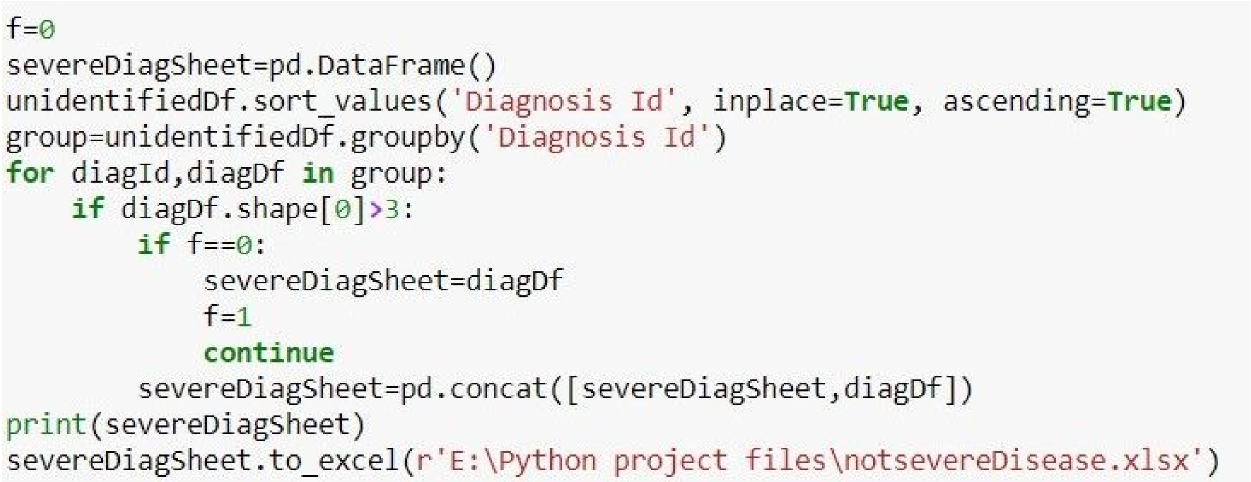
- Symptom vs fatality: -0.12

22. Perform analysis to calculate the Death percentage in top 3 Diseases.

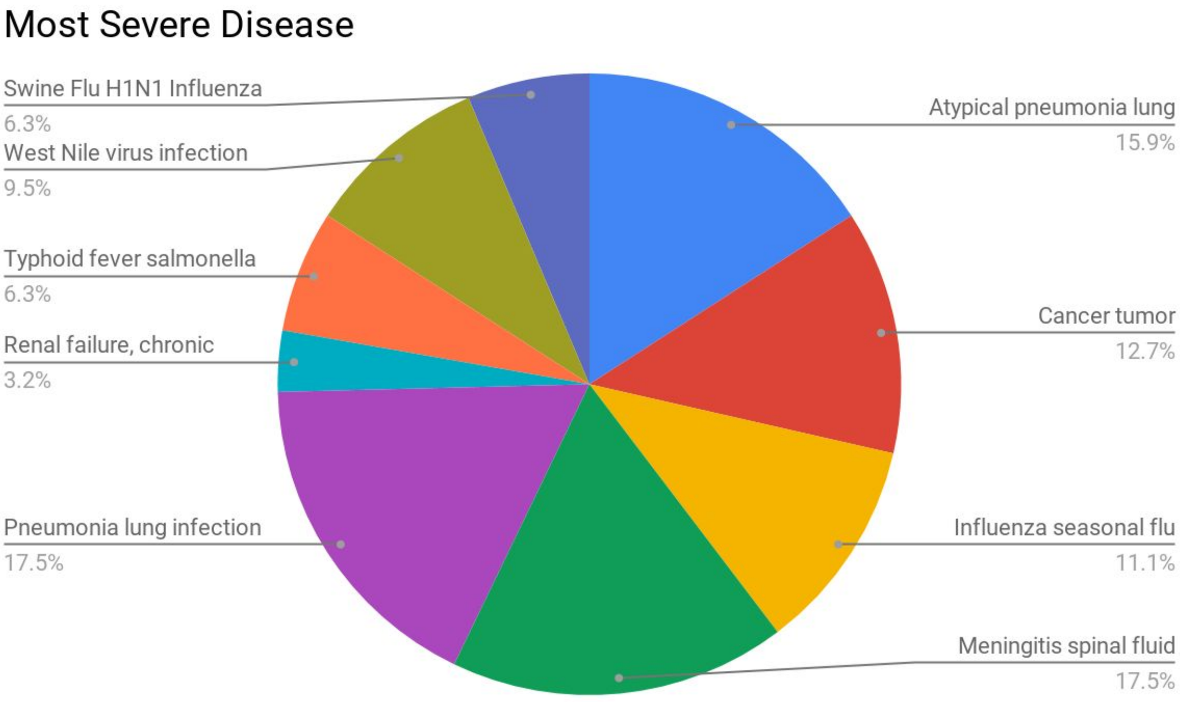
- Diseases vs fatality: -0.10

Analysis performed on the basis of various parameters with code, charts and inferences :

1. Perform analysis on diseases which are most untraceable

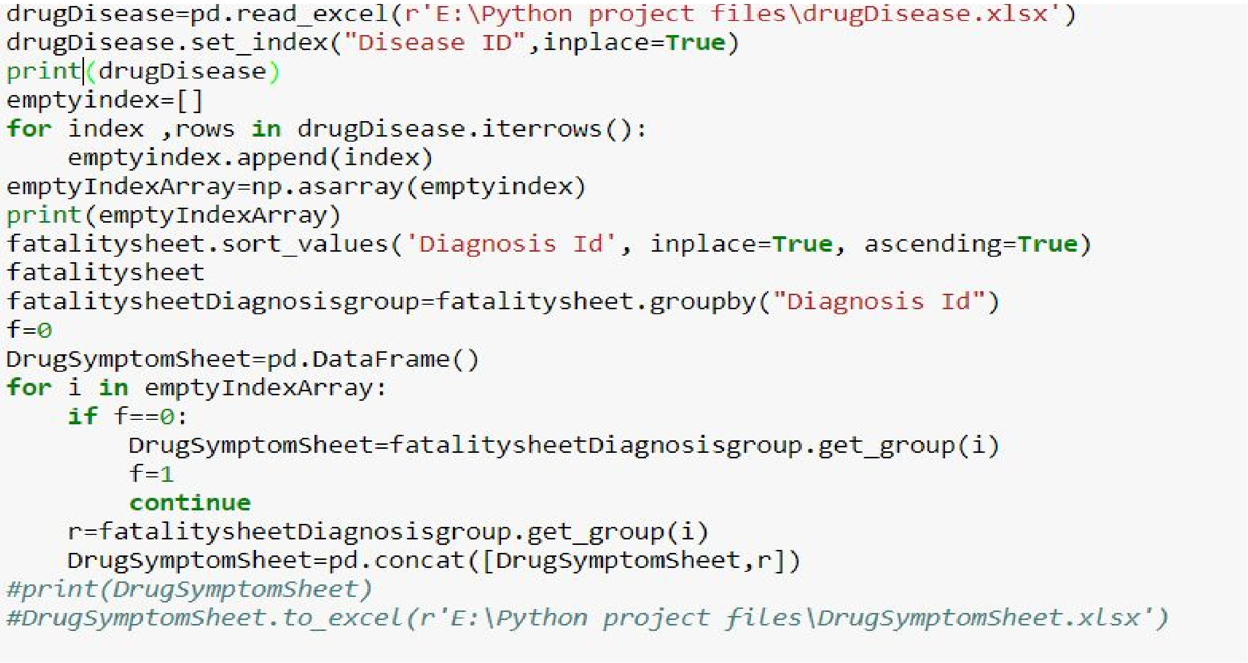


```
f=0
severeDiagSheet=pd.DataFrame()
unidentifiedDf.sort_values('Diagnosis Id', inplace=True, ascending=True)
group=unidentifiedDf.groupby('Diagnosis Id')
for diagId,diagDf in group:
    if diagDf.shape[0]>3:
        if f==0:
            severeDiagSheet=diagDf
            f=1
            continue
        severeDiagSheet=pd.concat([severeDiagSheet,diagDf])
print(severeDiagSheet)
severeDiagSheet.to_excel(r'E:\Python project files\notsevereDisease.xlsx')
```

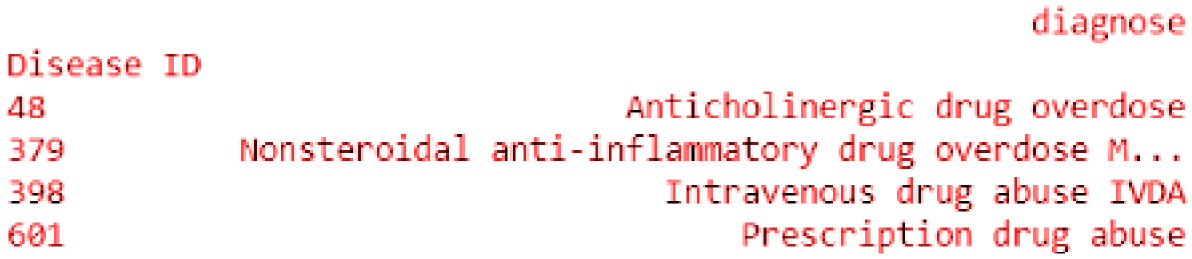


Inference:These are the diseases with most of the symptoms are unknown.

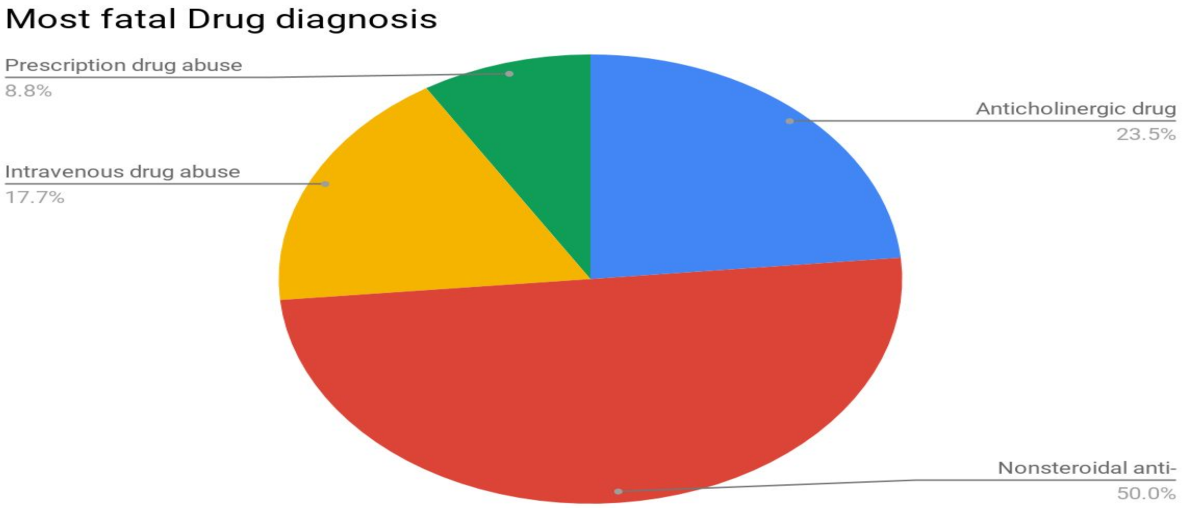
2. Perform analysis on the drug related diseases and their fatality



```
drugDisease=pd.read_excel(r'E:\Python project files\drugDisease.xlsx')
drugDisease.set_index("Disease ID",inplace=True)
print(drugDisease)
emptyindex=[]
for index ,rows in drugDisease.iterrows():
    emptyindex.append(index)
emptyIndexArray=np.asarray(emptyindex)
print(emptyIndexArray)
fatalitysheet.sort_values('Diagnosis Id', inplace=True, ascending=True)
fatalitysheet
fatalitysheetDiagnosisgroup=fatalitysheet.groupby("Diagnosis Id")
f=0
DrugSymptomSheet=pd.DataFrame()
for i in emptyIndexArray:
    if f==0:
        DrugSymptomSheet=fatalitysheetDiagnosisgroup.get_group(i)
        f=1
        continue
    r=fatalitysheetDiagnosisgroup.get_group(i)
    DrugSymptomSheet=pd.concat([DrugSymptomSheet,r])
#print(DrugSymptomSheet)
#DrugSymptomSheet.to_excel(r'E:\Python project files\DrugSymptomSheet.xlsx')
```

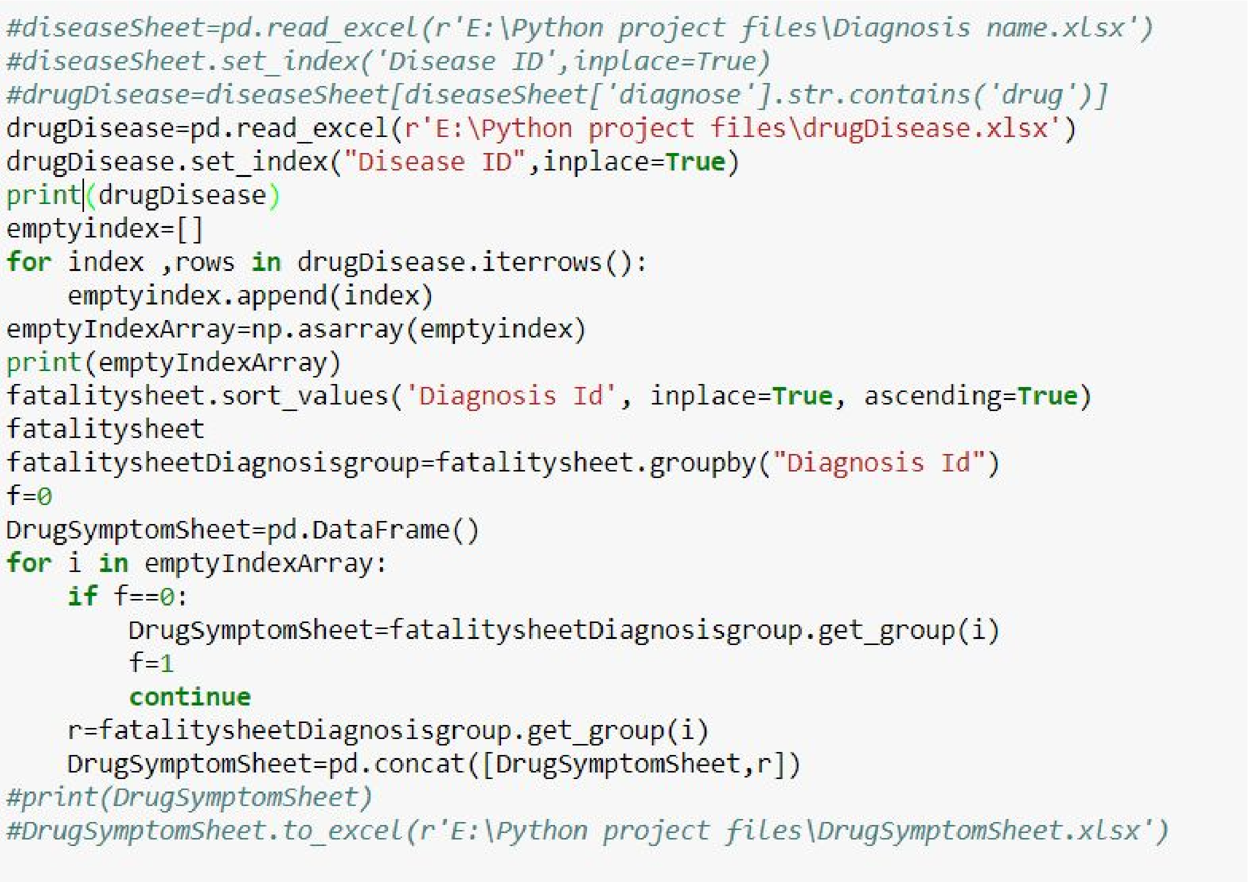


```
                                                diagnose
Disease ID
48                           Anticholinergic drug overdose
379         Nonsteroidal anti-inflammatory drug overdose M...
398                          Intravenous drug abuse IVDA
601                               Prescription drug abuse
```

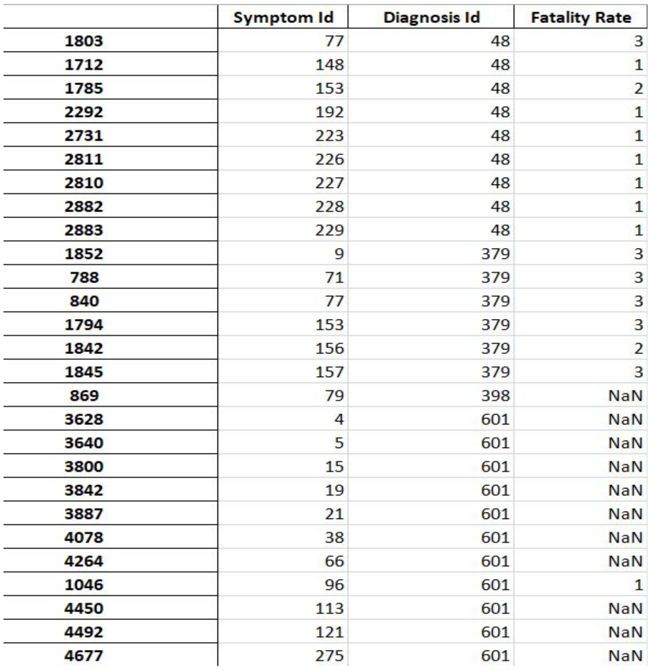


Inference : The most fatal drug diagnosis is **Nonsteroidal anti-inflammatory drug overdose Motrin, Advil** with fatality rate of **2.833**

3.Perform analysis to find the easiest Drug Related Disease



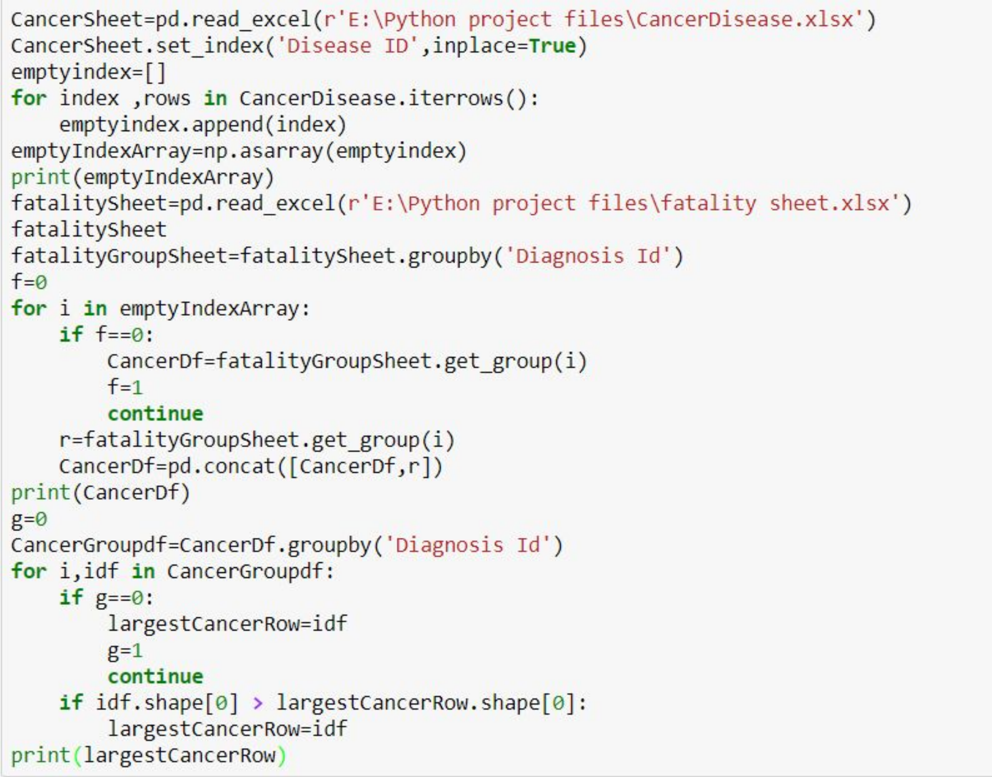
```
#diseaseSheet=pd.read_excel(r'E:\Python project files\Diagnosis name.xlsx')
#diseaseSheet.set_index('Disease ID',inplace=True)
#drugDisease=diseaseSheet[diseaseSheet['diagnose'].str.contains('drug')]
drugDisease=pd.read_excel(r'E:\Python project files\drugDisease.xlsx')
drugDisease.set_index("Disease ID",inplace=True)
print(drugDisease)
emptyindex=[]
for index ,rows in drugDisease.iterrows():
    emptyindex.append(index)
emptyIndexArray=np.asarray(emptyindex)
print(emptyIndexArray)
fatalitysheet.sort_values('Diagnosis Id', inplace=True, ascending=True)
fatalitysheet
fatalitysheetDiagnosisgroup=fatalitysheet.groupby("Diagnosis Id")
f=0
DrugSymptomSheet=pd.DataFrame()
for i in emptyIndexArray:
    if f==0:
        DrugSymptomSheet=fatalitysheetDiagnosisgroup.get_group(i)
        f=1
        continue
    r=fatalitysheetDiagnosisgroup.get_group(i)
    DrugSymptomSheet=pd.concat([DrugSymptomSheet,r])
#print(DrugSymptomSheet)
#DrugSymptomSheet.to_excel(r'E:\Python project files\DrugSymptomSheet.xlsx')
```



	Symptom Id	Diagnosis Id	Fatality Rate
1803	77	48	3
1712	148	48	1
1785	153	48	2
2292	192	48	1
2731	223	48	1
2811	226	48	1
2810	227	48	1
2882	228	48	1
2883	229	48	1
1852	9	379	3
788	71	379	3
840	77	379	3
1794	153	379	3
1842	156	379	2
1845	157	379	3
869	79	398	NaN
3628	4	601	NaN
3640	5	601	NaN
3800	15	601	NaN
3842	19	601	NaN
3887	21	601	NaN
4078	38	601	NaN
4264	66	601	NaN
1046	96	601	1
4450	113	601	NaN
4492	121	601	NaN
4677	275	601	NaN

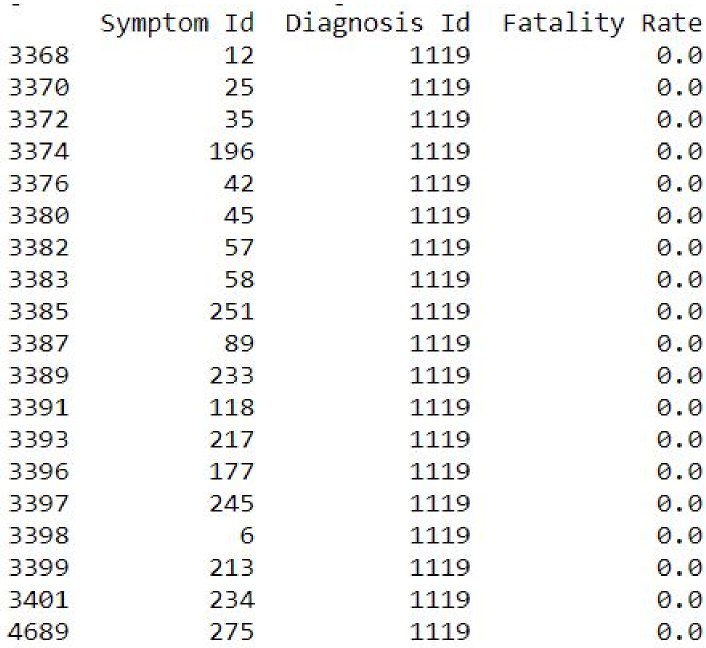
Inference : **Prescription Drug abuse** is the easiest to diagnose with 11 visual symptoms.

4. Perform analysis to find the most curable cancer.



```
CancerSheet=pd.read_excel(r'E:\Python project files\CancerDisease.xlsx')
CancerSheet.set_index('Disease ID',inplace=True)
emptyindex=[]
for index ,rows in CancerDisease.iterrows():
    emptyindex.append(index)
emptyIndexArray=np.asarray(emptyindex)
print(emptyIndexArray)
fatalitySheet=pd.read_excel(r'E:\Python project files\fatality sheet.xlsx')
fatalitySheet
fatalityGroupSheet=fatalitySheet.groupby('Diagnosis Id')
f=0
for i in emptyIndexArray:
    if f==0:
        CancerDf=fatalityGroupSheet.get_group(i)
        f=1
        continue
    r=fatalityGroupSheet.get_group(i)
    CancerDf=pd.concat([CancerDf,r])
print(CancerDf)
g=0
CancerGroupdf=CancerDf.groupby('Diagnosis Id')
for i,idf in CancerGroupdf:
    if g==0:
        largestCancerRow=idf
        g=1
        continue
    if idf.shape[0] > largestCancerRow.shape[0]:
        largestCancerRow=idf
print(largestCancerRow)
```

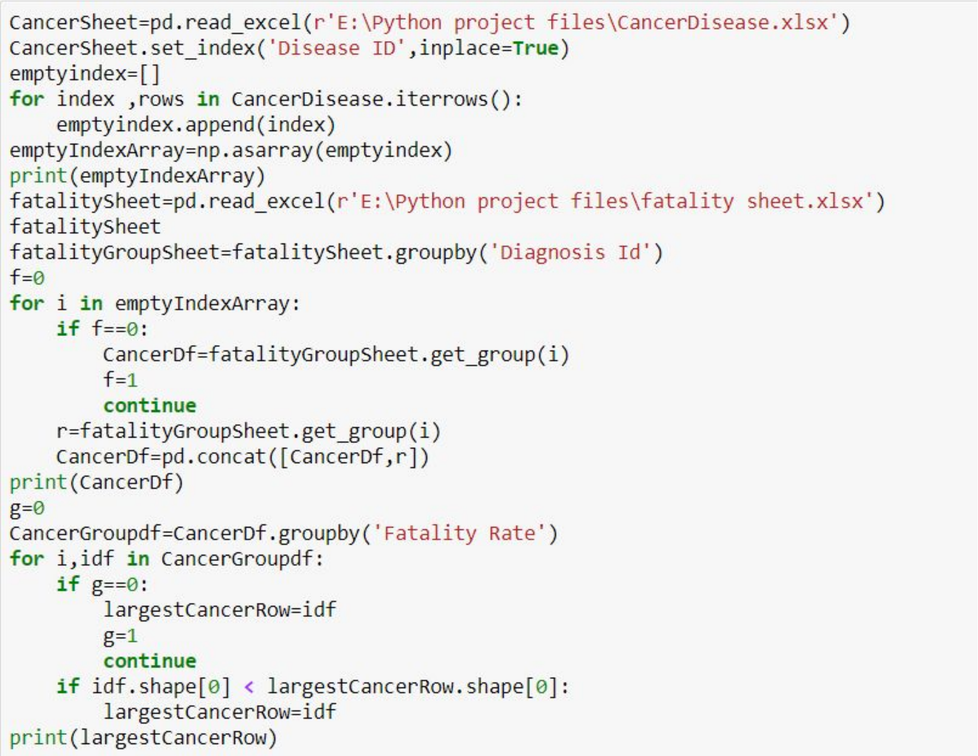
The symptom table of the most curable disease:



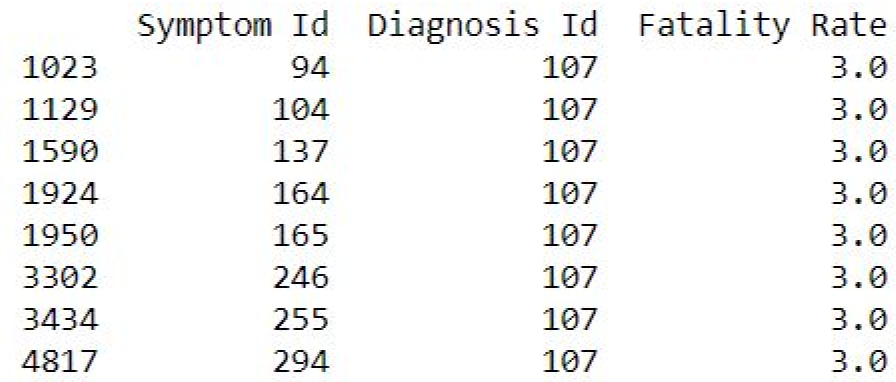
```
      Symptom Id  Diagnosis Id  Fatality Rate
3368          12          1119            0.0
3370          25          1119            0.0
3372          35          1119            0.0
3374         196          1119            0.0
3376          42          1119            0.0
3380          45          1119            0.0
3382          57          1119            0.0
3383          58          1119            0.0
3385         251          1119            0.0
3387          89          1119            0.0
3389         233          1119            0.0
3391         118          1119            0.0
3393         217          1119            0.0
3396         177          1119            0.0
3397         245          1119            0.0
3398           6          1119            0.0
3399         213          1119            0.0
3401         234          1119            0.0
4689         275          1119            0.0
```

Inference : **Sarcoma soft tissue cancer** is the most curable cancer with 19 visual symptoms and with fatality value 0.0

5. Perform analysis to find the most fatal cancer.



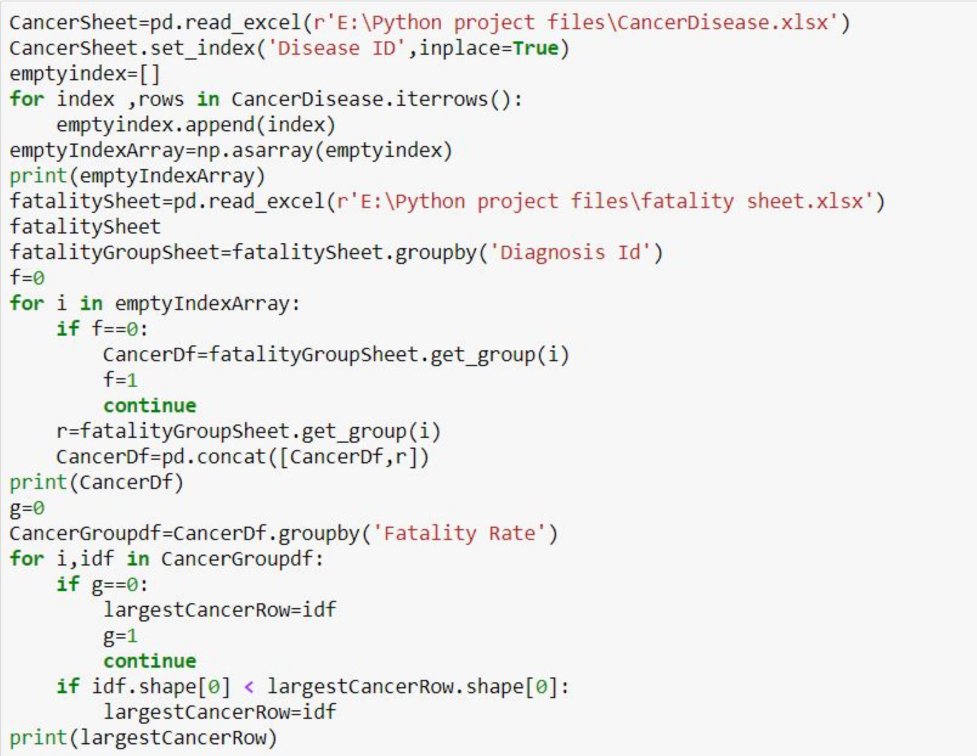
```
CancerSheet=pd.read_excel(r'E:\Python project files\CancerDisease.xlsx')
CancerSheet.set_index('Disease ID',inplace=True)
emptyindex=[]
for index ,rows in CancerDisease.iterrows():
    emptyindex.append(index)
emptyIndexArray=np.asarray(emptyindex)
print(emptyIndexArray)
fatalitySheet=pd.read_excel(r'E:\Python project files\fatality sheet.xlsx')
fatalitySheet
fatalityGroupSheet=fatalitySheet.groupby('Diagnosis Id')
f=0
for i in emptyIndexArray:
    if f==0:
        CancerDf=fatalityGroupSheet.get_group(i)
        f=1
        continue
    r=fatalityGroupSheet.get_group(i)
    CancerDf=pd.concat([CancerDf,r])
print(CancerDf)
g=0
CancerGroupdf=CancerDf.groupby('Fatality Rate')
for i,idf in CancerGroupdf:
    if g==0:
        largestCancerRow=idf
        g=1
        continue
    if idf.shape[0] < largestCancerRow.shape[0]:
        largestCancerRow=idf
print(largestCancerRow)
```



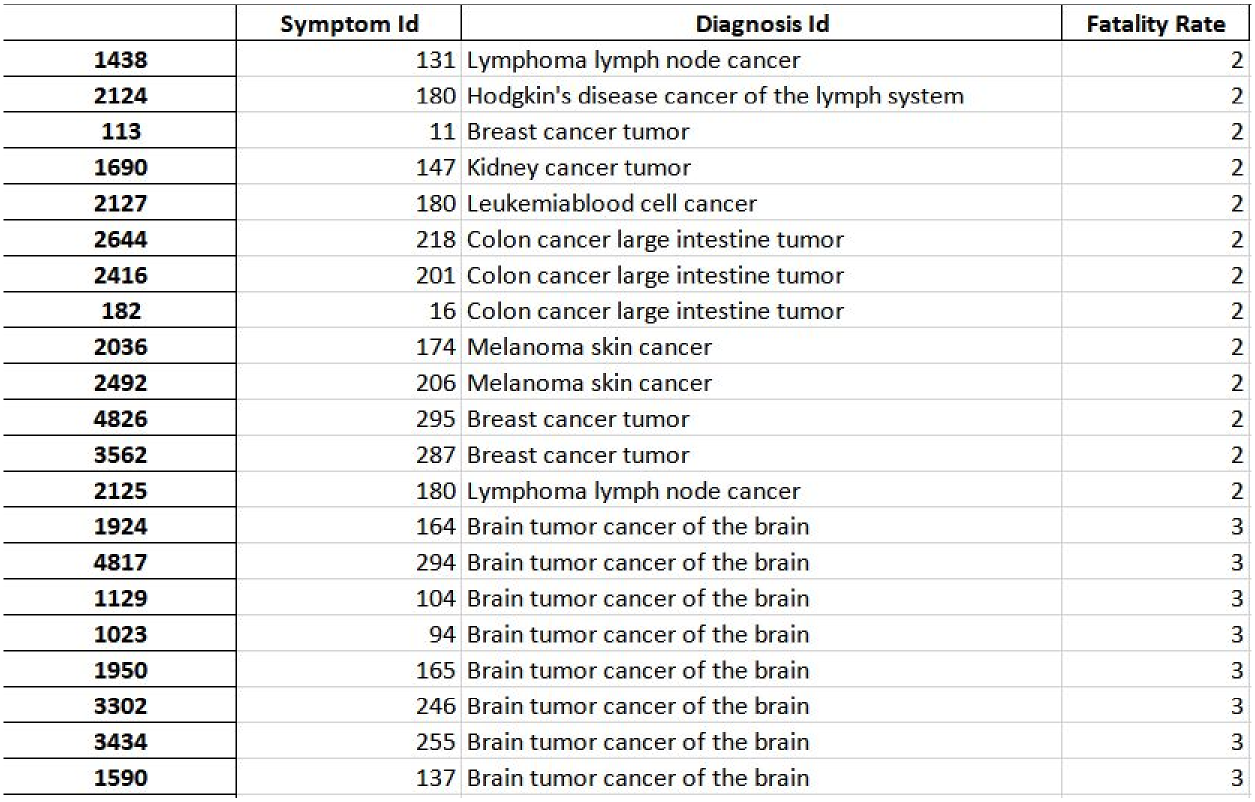
```
      Symptom Id  Diagnosis Id  Fatality Rate
1023          94           107            3.0
1129         104           107            3.0
1590         137           107            3.0
1924         164           107            3.0
1950         165           107            3.0
3302         246           107            3.0
3434         255           107            3.0
4817         294           107            3.0
```

Inference : The most fatal type of cancer is brain tumor or commonly known as cancer of the brain

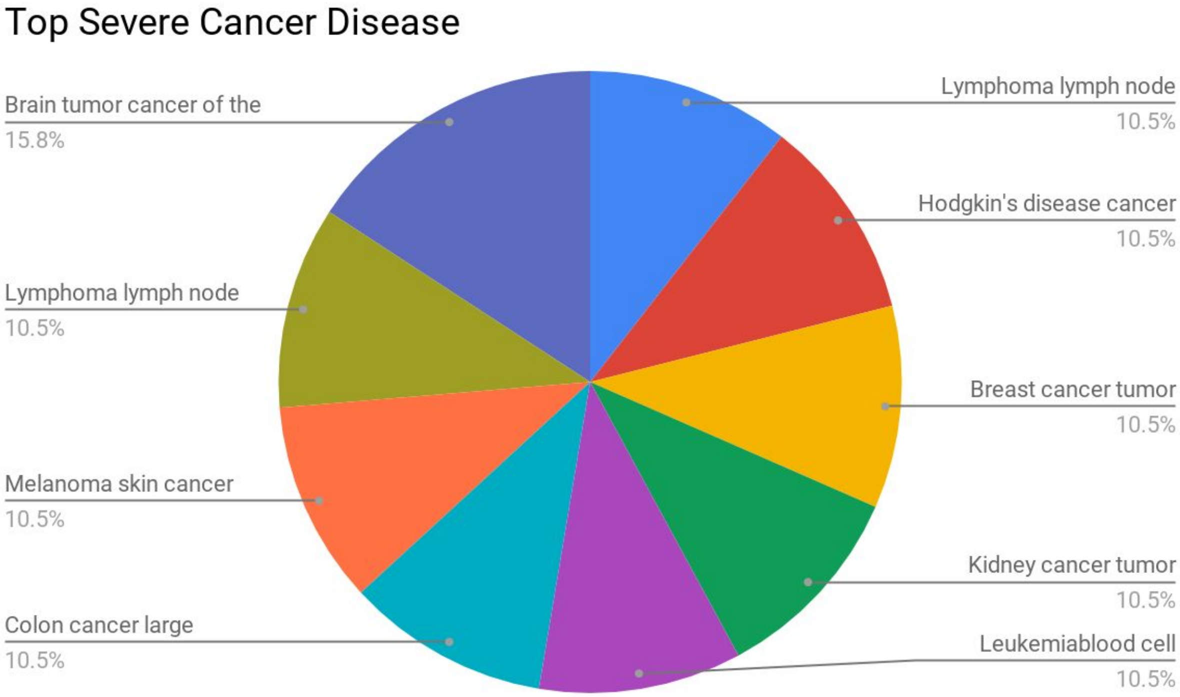
6. Perform analysis to find the top 10 Severe cancer type.



```
CancerSheet=pd.read_excel(r'E:\Python project files\CancerDisease.xlsx')
CancerSheet.set_index('Disease ID',inplace=True)
emptyindex=[]
for index ,rows in CancerDisease.iterrows():
    emptyindex.append(index)
emptyIndexArray=np.asarray(emptyindex)
print(emptyIndexArray)
fatalitySheet=pd.read_excel(r'E:\Python project files\fatality sheet.xlsx')
fatalitySheet
fatalityGroupSheet=fatalitySheet.groupby('Diagnosis Id')
f=0
for i in emptyIndexArray:
    if f==0:
        CancerDf=fatalityGroupSheet.get_group(i)
        f=1
        continue
    r=fatalityGroupSheet.get_group(i)
    CancerDf=pd.concat([CancerDf,r])
print(CancerDf)
g=0
CancerGroupdf=CancerDf.groupby('Fatality Rate')
for i,idf in CancerGroupdf:
    if g==0:
        largestCancerRow=idf
        g=1
        continue
    if idf.shape[0] < largestCancerRow.shape[0]:
        largestCancerRow=idf
print(largestCancerRow)
```

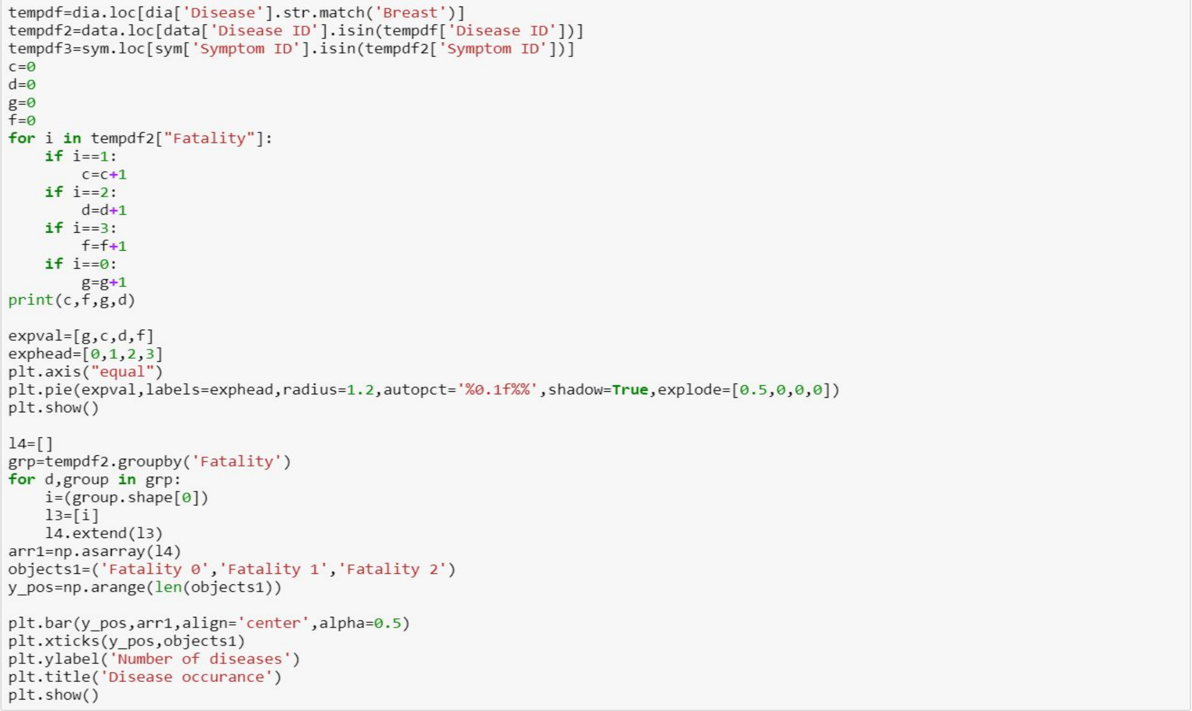


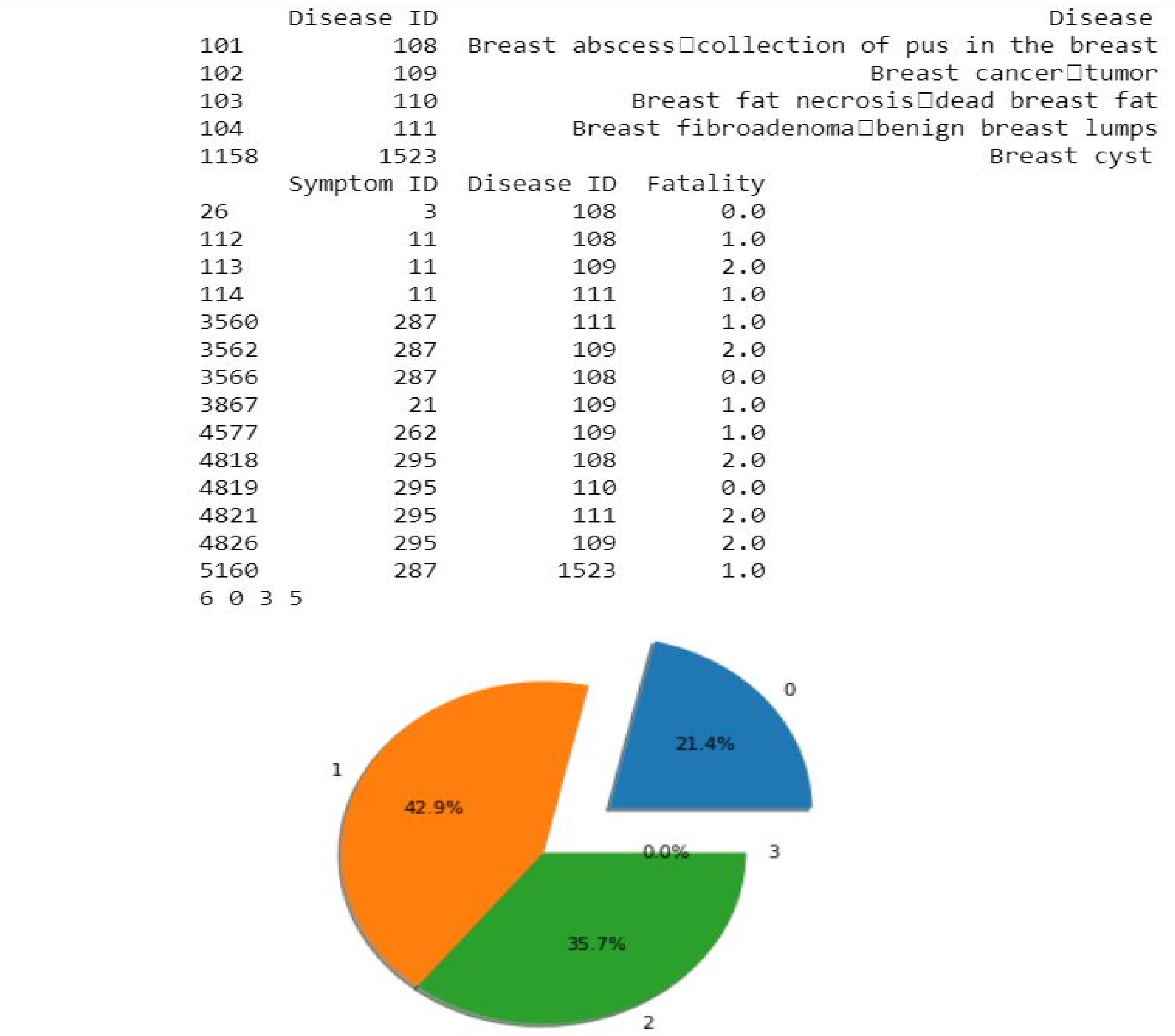
	Symptom Id	Diagnosis Id	Fatality Rate
1438	131	Lymphoma lymph node cancer	2
2124	180	Hodgkin's disease cancer of the lymph system	2
113	11	Breast cancer tumor	2
1690	147	Kidney cancer tumor	2
2127	180	Leukemiablood cell cancer	2
2644	218	Colon cancer large intestine tumor	2
2416	201	Colon cancer large intestine tumor	2
182	16	Colon cancer large intestine tumor	2
2036	174	Melanoma skin cancer	2
2492	206	Melanoma skin cancer	2
4826	295	Breast cancer tumor	2
3562	287	Breast cancer tumor	2
2125	180	Lymphoma lymph node cancer	2
1924	164	Brain tumor cancer of the brain	3
4817	294	Brain tumor cancer of the brain	3
1129	104	Brain tumor cancer of the brain	3
1023	94	Brain tumor cancer of the brain	3
1950	165	Brain tumor cancer of the brain	3
3302	246	Brain tumor cancer of the brain	3
3434	255	Brain tumor cancer of the brain	3
1590	137	Brain tumor cancer of the brain	3



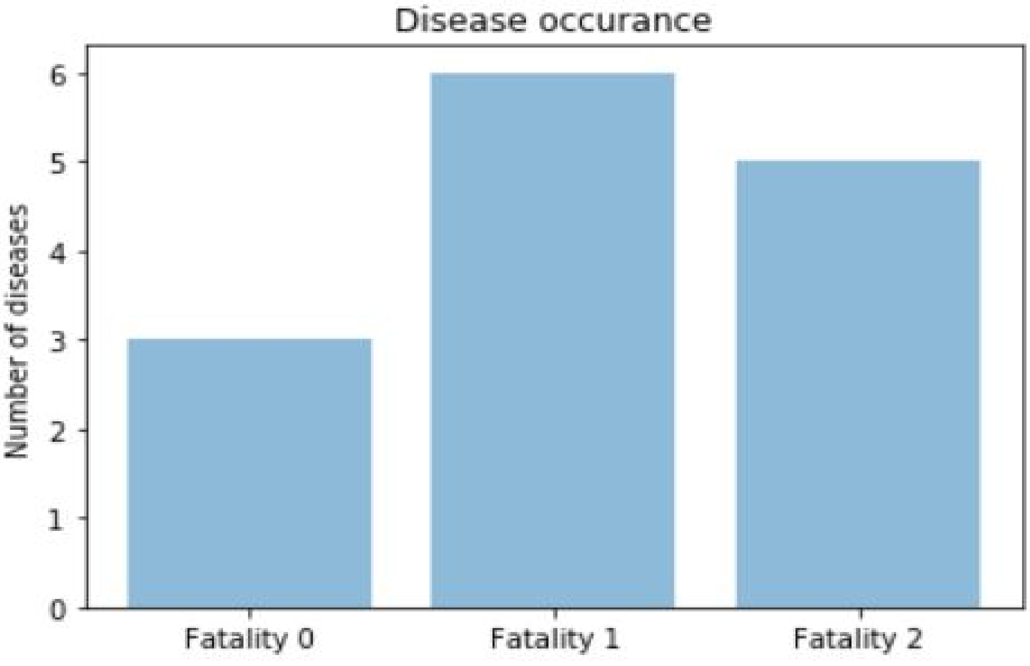
Inference : Most of the cancer whose fatality is 2 are quite untraceable since they have very few symptoms

7. Perform analysis to find breast related diseases with their specific symptoms


```
tempdf=dia.loc[dia['Disease'].str.match('Breast')]
tempdf2=data.loc[data['Disease ID'].isin(tempdf['Disease ID'])]
tempdf3=sym.loc[sym['Symptom ID'].isin(tempdf2['Symptom ID'])]
c=0
d=0
g=0
f=0
for i in tempdf2["Fatality"]:
    if i==1:
        c=c+1
    if i==2:
        d=d+1
    if i==3:
        f=f+1
    if i==0:
        g=g+1
print(c,f,g,d)

expval=[g,c,d,f]
exphead=[0,1,2,3]
plt.axis("equal")
plt.pie(expval,labels=exphead,radius=1.2,autopct='%0.1f%%',shadow=True,explode=[0.5,0,0,0])
plt.show()

l4=[]
grp=tempdf2.groupby('Fatality')
for d,group in grp:
    i=(group.shape[0])
    l3=[i]
    l4.extend(l3)
arr1=np.asarray(l4)
objects1=('Fatality 0','Fatality 1','Fatality 2')
y_pos=np.arange(len(objects1))

plt.bar(y_pos,arr1,align='center',alpha=0.5)
plt.xticks(y_pos,objects1)
plt.ylabel('Number of diseases')
plt.title('Disease occurance')
plt.show()
```

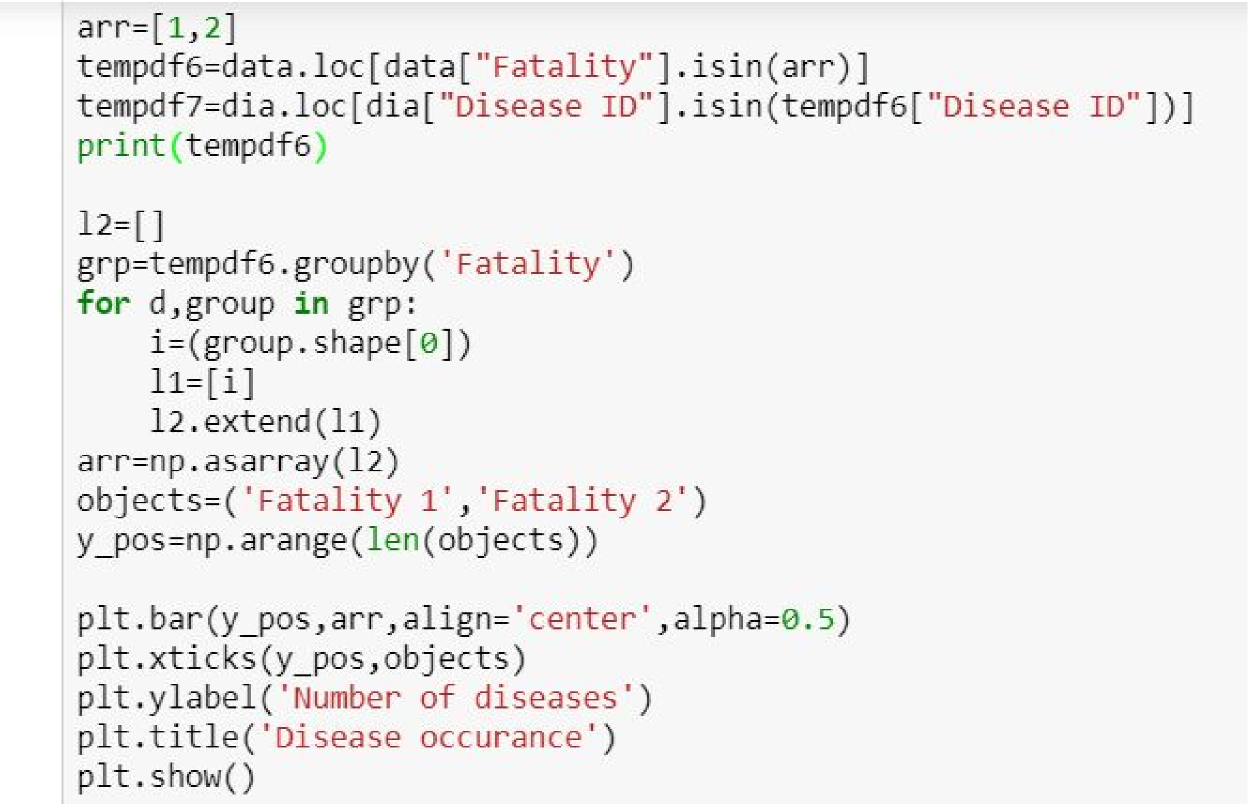


```
      Disease ID                                      Disease
101          108  Breast abscess□collection of pus in the breast
102          109                         Breast cancer□tumor
103          110              Breast fat necrosis□dead breast fat
104          111          Breast fibroadenoma□benign breast lumps
1158        1523                                  Breast cyst
      Symptom ID  Disease ID  Fatality
26             3         108       0.0
112           11         108       1.0
113           11         109       2.0
114           11         111       1.0
3560         287         111       1.0
3562         287         109       2.0
3566         287         108       0.0
3867          21         109       1.0
4577         262         109       1.0
4818         295         108       2.0
4819         295         110       0.0
4821         295         111       2.0
4826         295         109       2.0
5160         287        1523       1.0
6 0 3 5
```

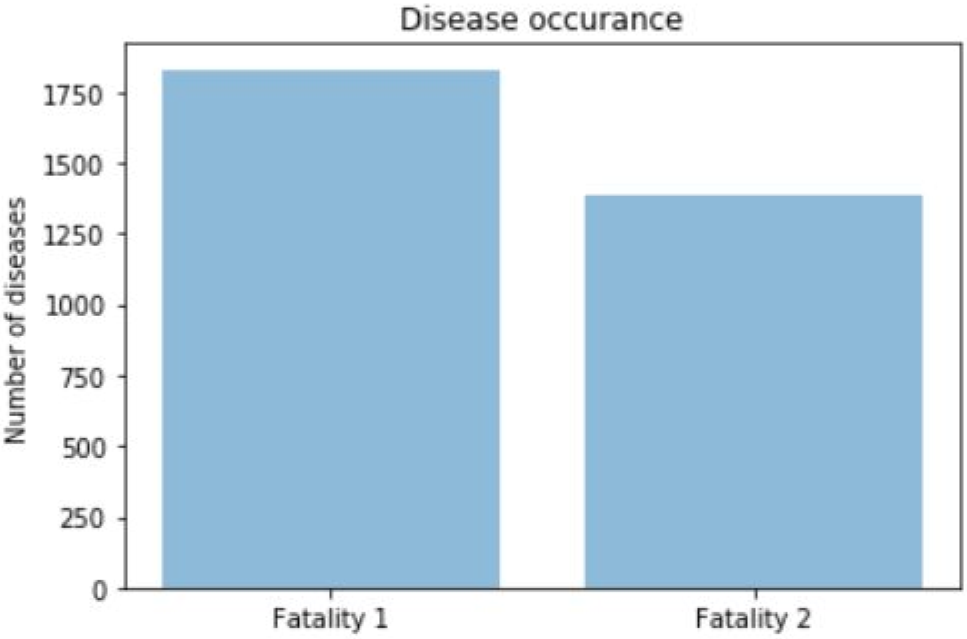


Inference : Most of the breasts related diseases have minimal fatality

8.Perform analysis to find the number of diseases whose fatality is 1 or 2

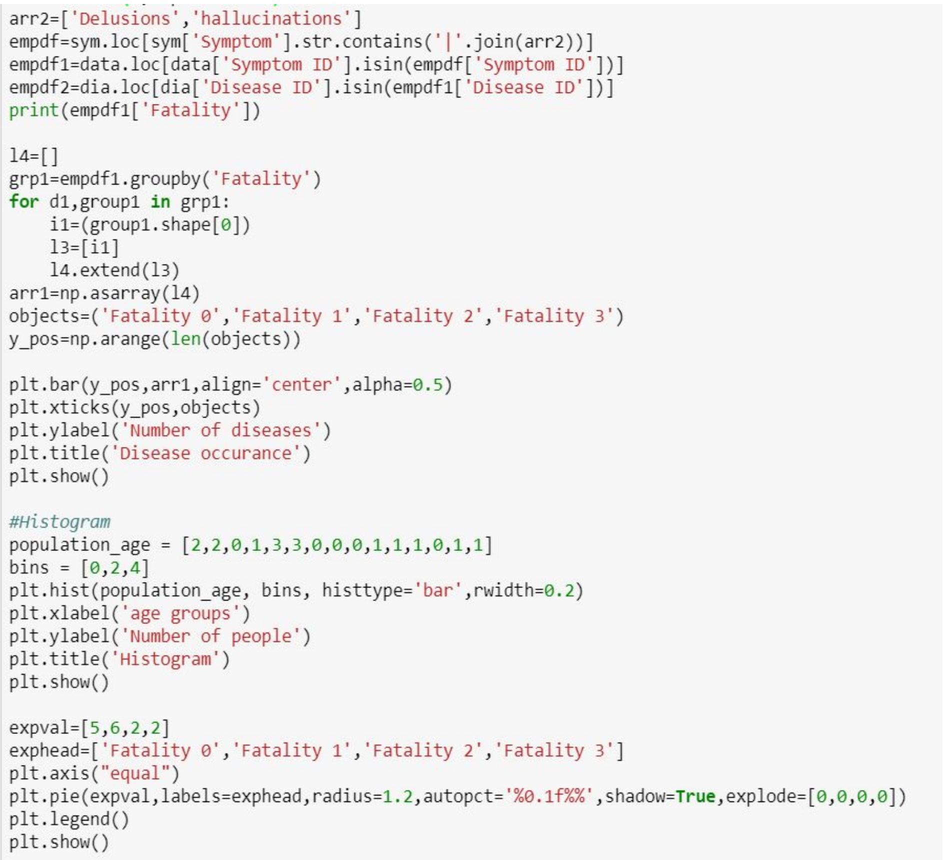


```
arr=[1,2]
tempdf6=data.loc[data["Fatality"].isin(arr)]
tempdf7=dia.loc[dia["Disease ID"].isin(tempdf6["Disease ID"])]
print(tempdf6)

l2=[]
grp=tempdf6.groupby('Fatality')
for d,group in grp:
    i=(group.shape[0])
    l1=[i]
    l2.extend(l1)
arr=np.asarray(l2)
objects=('Fatality 1','Fatality 2')
y_pos=np.arange(len(objects))

plt.bar(y_pos,arr,align='center',alpha=0.5)
plt.xticks(y_pos,objects)
plt.ylabel('Number of diseases')
plt.title('Disease occurance')
plt.show()
```

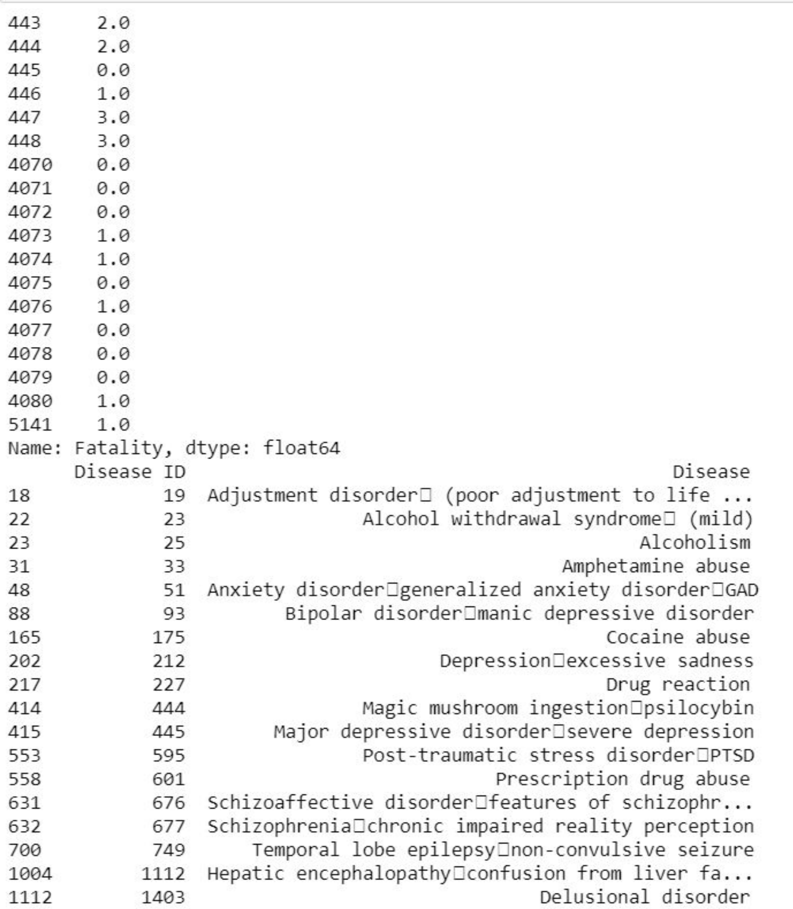


Inference : Most of the diseases have minimal fatality or fatality equal to one

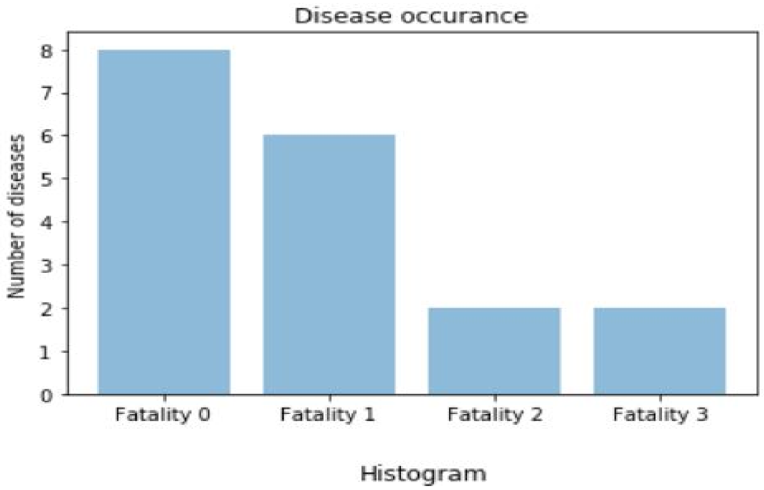
9.Perform analysis on the fact to find possible diseases if the symptoms are delusions and hallucinations.

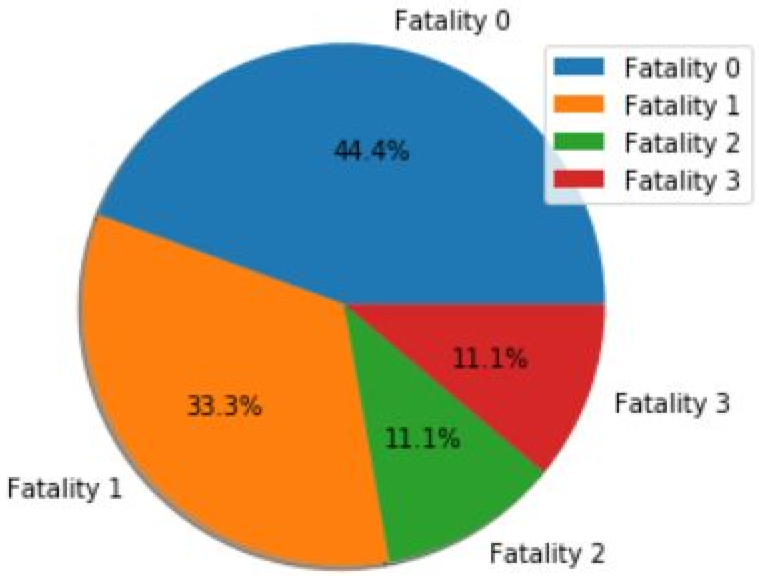


```
arr2=['Delusions','hallucinations']
empdf=sym.loc[sym['Symptom'].str.contains('|'.join(arr2))]
empdf1=data.loc[data['Symptom ID'].isin(empdf['Symptom ID'])]
empdf2=dia.loc[dia['Disease ID'].isin(empdf1['Disease ID'])]
print(empdf1['Fatality'])

l4=[]
grp1=empdf1.groupby('Fatality')
for d1,group1 in grp1:
    i1=(group1.shape[0])
    l3=[i1]
    l4.extend(l3)
arr1=np.asarray(l4)
objects=('Fatality 0','Fatality 1','Fatality 2','Fatality 3')
y_pos=np.arange(len(objects))

plt.bar(y_pos,arr1,align='center',alpha=0.5)
plt.xticks(y_pos,objects)
plt.ylabel('Number of diseases')
plt.title('Disease occurance')
plt.show()

#Histogram
population_age = [2,2,0,1,3,3,0,0,0,1,1,1,0,1,1]
bins = [0,2,4]
plt.hist(population_age, bins, histtype='bar',rwidth=0.2)
plt.xlabel('age groups')
plt.ylabel('Number of people')
plt.title('Histogram')
plt.show()

expval=[5,6,2,2]
exphead=['Fatality 0','Fatality 1','Fatality 2','Fatality 3']
plt.axis("equal")
plt.pie(expval,labels=exphead,radius=1.2,autopct='%0.1f%%',shadow=True,explode=[0,0,0,0])
plt.legend()
plt.show()
```



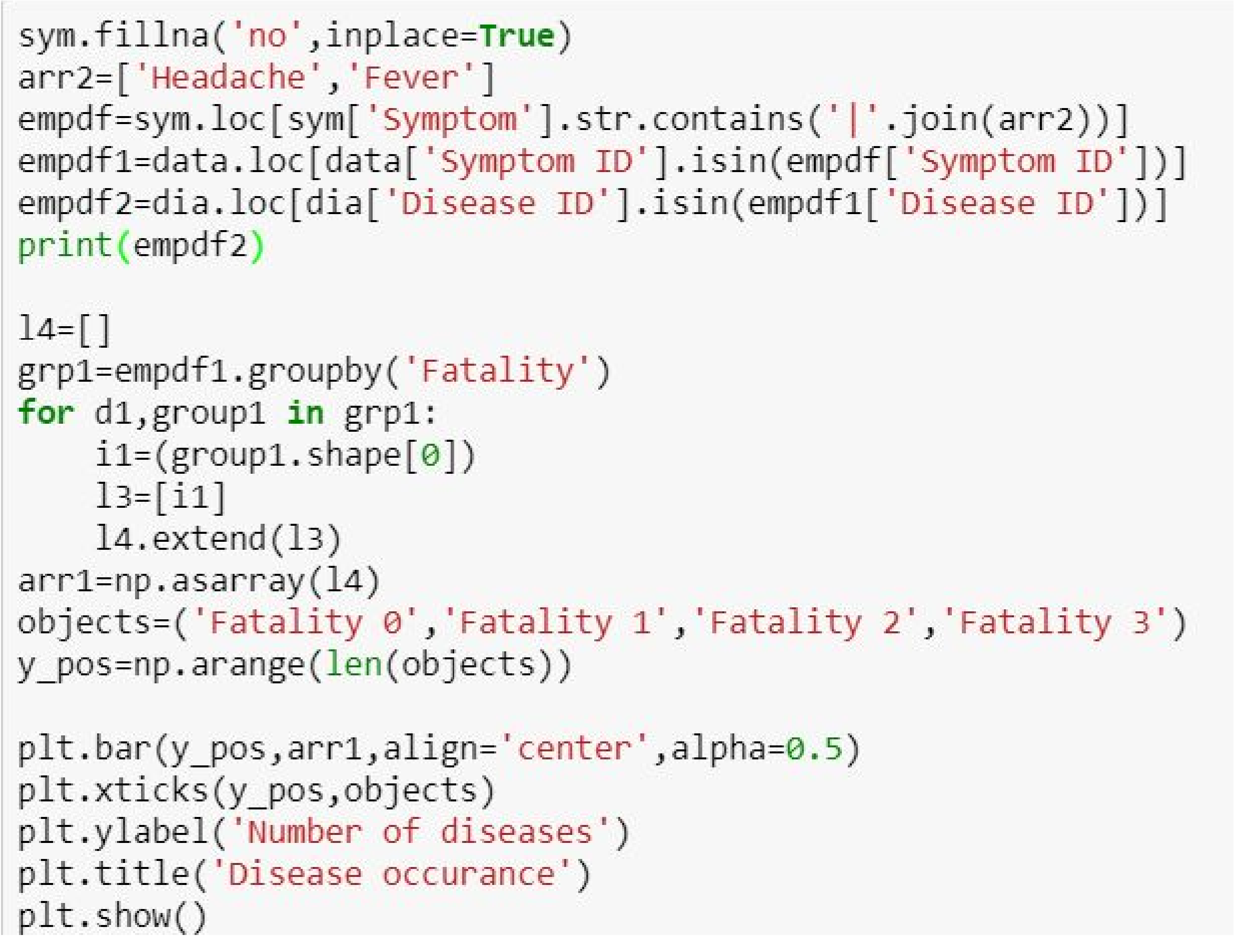
```
443     2.0
444     2.0
445     0.0
446     1.0
447     3.0
448     3.0
4070    0.0
4071    0.0
4072    0.0
4073    1.0
4074    1.0
4075    0.0
4076    1.0
4077    0.0
4078    0.0
4079    0.0
4080    1.0
5141    1.0
Name: Fatality, dtype: float64
      Disease ID                                           Disease
18            19  Adjustment disorder□ (poor adjustment to life ...
22            23              Alcohol withdrawal syndrome□ (mild)
23            25                                        Alcoholism
31            33                                  Amphetamine abuse
48            51  Anxiety disorder□generalized anxiety disorder□GAD
88            93        Bipolar disorder□manic depressive disorder
165          175                                      Cocaine abuse
202          212                   Depression□excessive sadness
217          227                                      Drug reaction
414          444              Magic mushroom ingestion□psilocybin
415          445         Major depressive disorder□severe depression
553          595              Post-traumatic stress disorder□PTSD
558          601                            Prescription drug abuse
631          676  Schizoaffective disorder□features of schizophr...
632          677  Schizophrenia□chronic impaired reality perception
700          749      Temporal lobe epilepsy□non-convulsive seizure
1004        1112  Hepatic encephalopathy□confusion from liver fa...
1112        1403                                 Delusional disorder
```



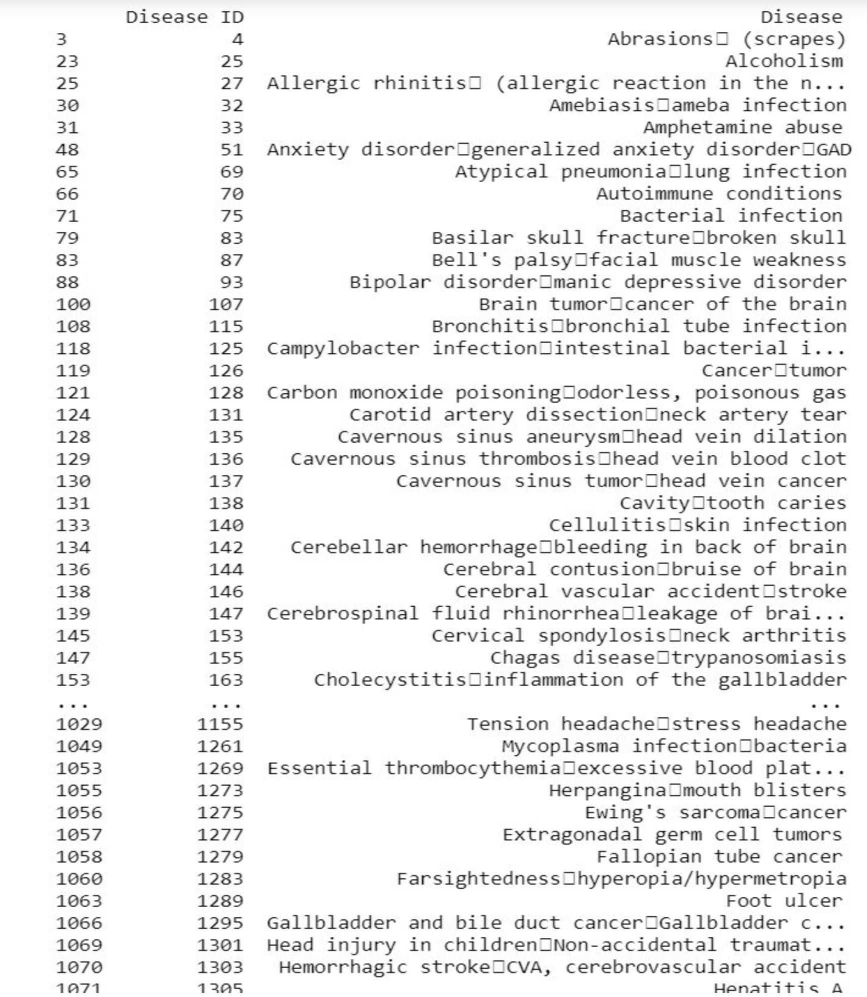


Inference : Most Diseases related delusions and hallucinations are much less fatal. As understood from the above bar graph.

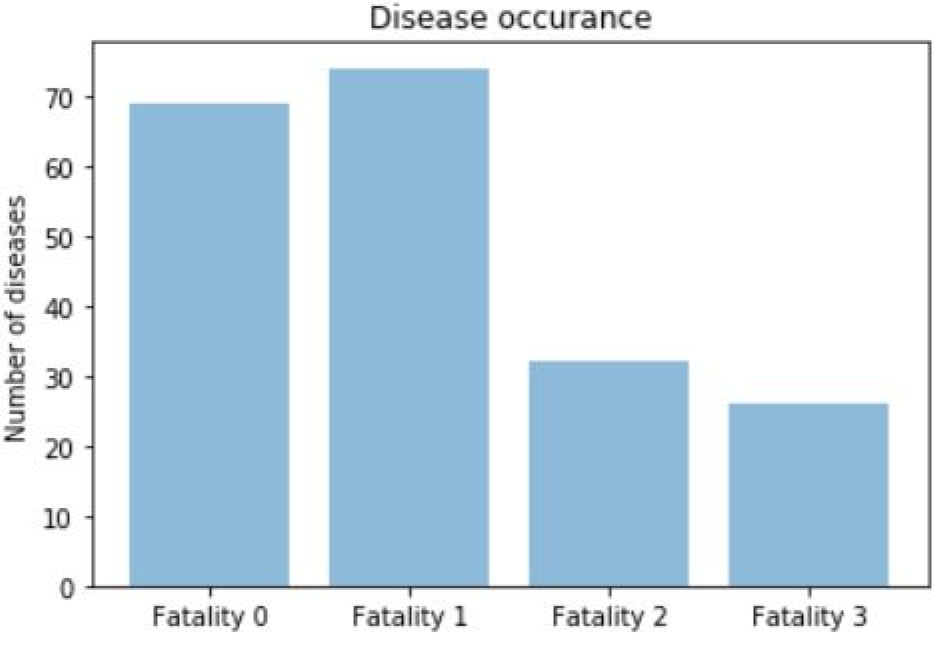
10. Perform analysis on the fact that diseases which have symptoms of both headache and fever



```python
sym.fillna('no',inplace=True)
arr2=['Headache','Fever']
empdf=sym.loc[sym['Symptom'].str.contains('|'.join(arr2))]
empdf1=data.loc[data['Symptom ID'].isin(empdf['Symptom ID'])]
empdf2=dia.loc[dia['Disease ID'].isin(empdf1['Disease ID'])]
print(empdf2)

l4=[]
grp1=empdf1.groupby('Fatality')
for d1,group1 in grp1:
    i1=(group1.shape[0])
    l3=[i1]
    l4.extend(l3)
arr1=np.asarray(l4)
objects=('Fatality 0','Fatality 1','Fatality 2','Fatality 3')
y_pos=np.arange(len(objects))

plt.bar(y_pos,arr1,align='center',alpha=0.5)
plt.xticks(y_pos,objects)
plt.ylabel('Number of diseases')
plt.title('Disease occurance')
plt.show()
```

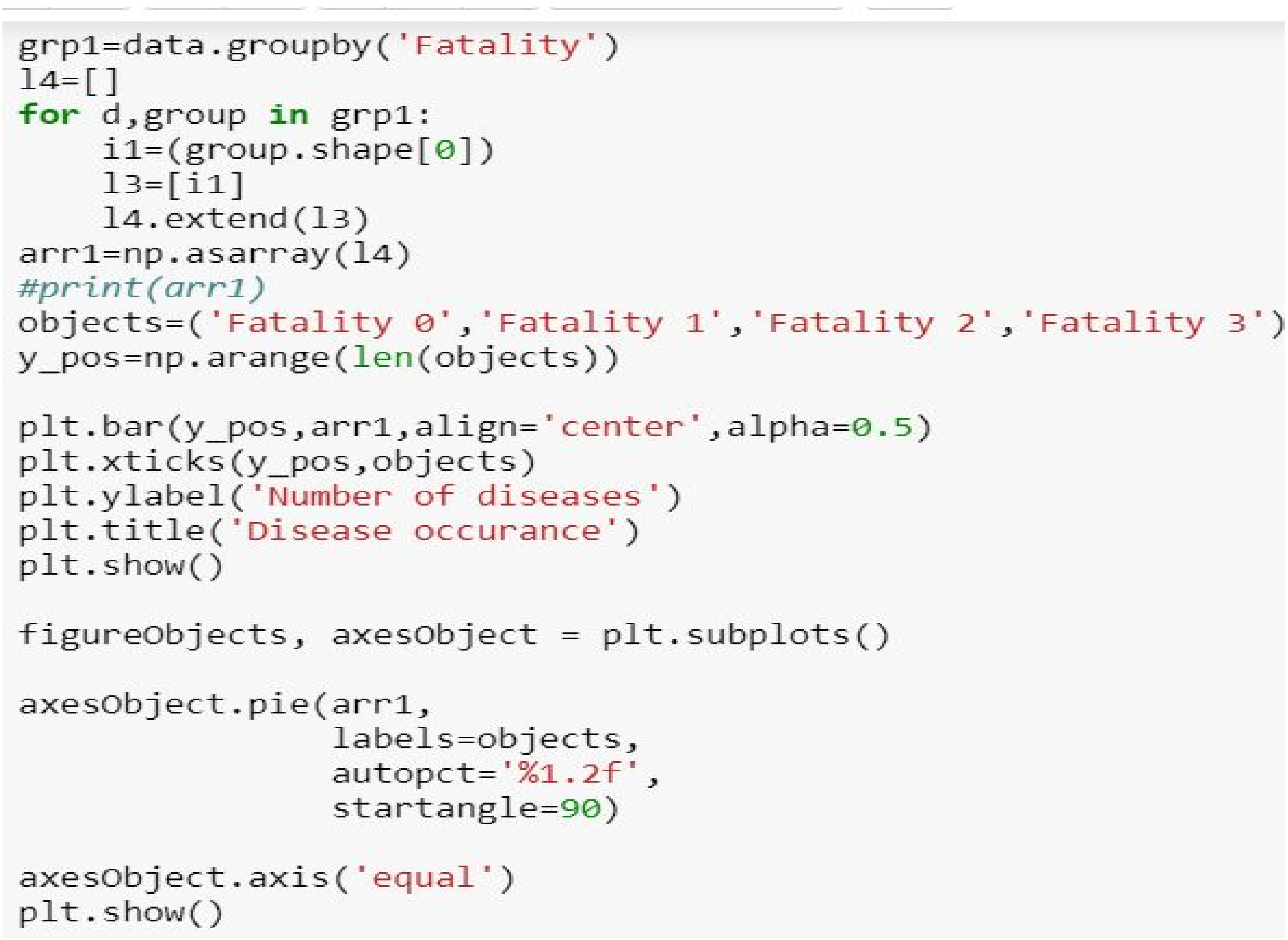


```
      Disease ID                                           Disease
3              4                             Abrasions□ (scrapes)
23            25                                        Alcoholism
25            27  Allergic rhinitis□ (allergic reaction in the n...
30            32                          Amebiasis□ameba infection
31            33                                  Amphetamine abuse
48            51  Anxiety disorder□generalized anxiety disorder□GAD
65            69                Atypical pneumonia□lung infection
66            70                              Autoimmune conditions
71            75                                 Bacterial infection
79            83                  Basilar skull fracture□broken skull
83            87                  Bell's palsy□facial muscle weakness
88            93             Bipolar disorder□manic depressive disorder
100          107                      Brain tumor□cancer of the brain
108          115                 Bronchitis□bronchial tube infection
118          125  Campylobacter infection□intestinal bacterial i...
119          126                                       Cancer□tumor
121          128  Carbon monoxide poisoning□odorless, poisonous gas
124          131             Carotid artery dissection□neck artery tear
128          135            Cavernous sinus aneurysm□head vein dilation
129          136        Cavernous sinus thrombosis□head vein blood clot
130          137               Cavernous sinus tumor□head vein cancer
131          138                                Cavity□tooth caries
133          140                             Cellulitis□skin infection
134          142        Cerebellar hemorrhage□bleeding in back of brain
136          144                 Cerebral contusion□bruise of brain
138          146                 Cerebral vascular accident□stroke
139          147  Cerebrospinal fluid rhinorrhea□leakage of brai...
145          153                  Cervical spondylosis□neck arthritis
147          155                     Chagas disease□trypanosomiasis
153          163      Cholecystitis□inflammation of the gallbladder
...          ...                                                ...
1029        1155                    Tension headache□stress headache
1049        1261                       Mycoplasma infection□bacteria
1053        1269  Essential thrombocythemia□excessive blood plat...
1055        1273                           Herpangina□mouth blisters
1056        1275                              Ewing's sarcoma□cancer
1057        1277                        Extragonadal germ cell tumors
1058        1279                                Fallopian tube cancer
1060        1283                Farsightedness□hyperopia/hypermetropia
1063        1289                                          Foot ulcer
1066        1295  Gallbladder and bile duct cancer□Gallbladder c...
1069        1301  Head injury in children□Non-accidental traumat...
1070        1303   Hemorrhagic stroke□CVA, cerebrovascular accident
1071        1305                                         Hepatitis A
```

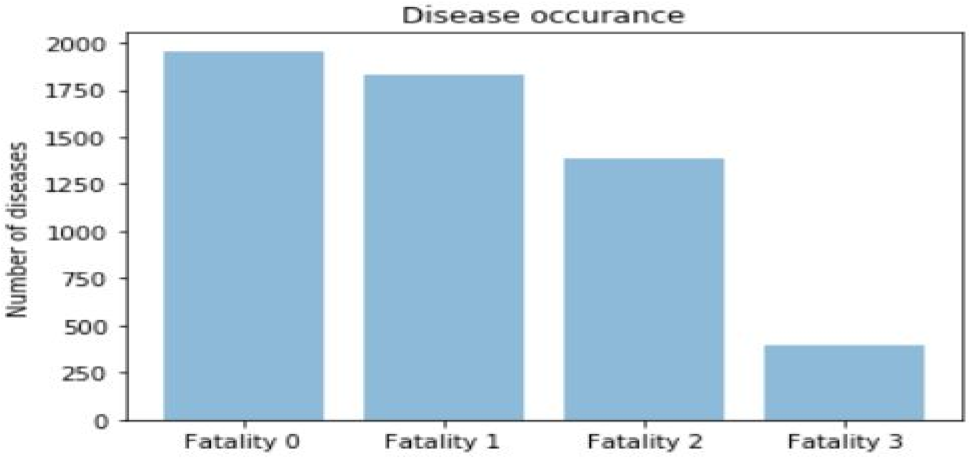


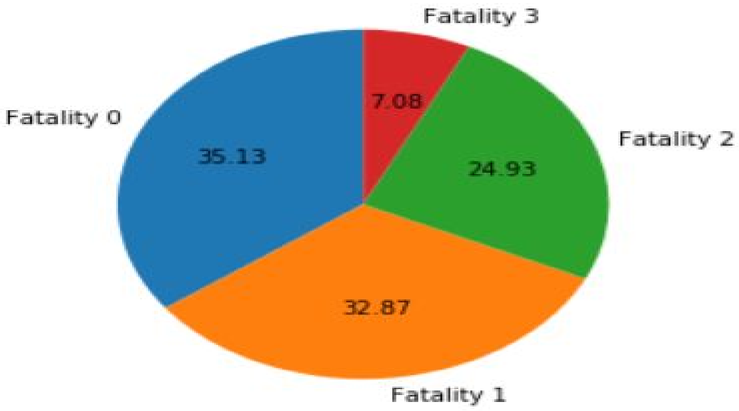
Inference : Most of the diseases with the symptoms of headache and fever have minimal fatality

11. Perform analysis on the fact that Showing, the diseases based on fatality rate



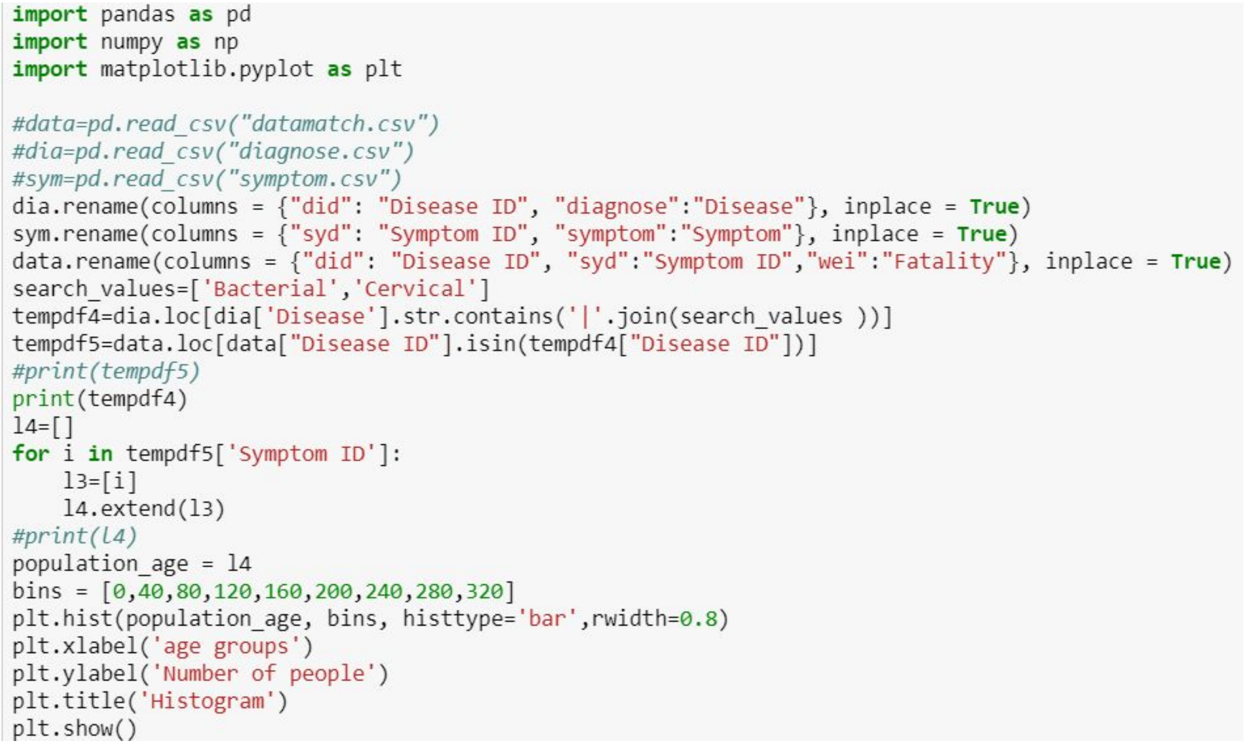
```
grp1=data.groupby('Fatality')
l4=[]
for d,group in grp1:
    i1=(group.shape[0])
    l3=[i1]
    l4.extend(l3)
arr1=np.asarray(l4)
#print(arr1)
objects=('Fatality 0','Fatality 1','Fatality 2','Fatality 3')
y_pos=np.arange(len(objects))

plt.bar(y_pos,arr1,align='center',alpha=0.5)
plt.xticks(y_pos,objects)
plt.ylabel('Number of diseases')
plt.title('Disease occurance')
plt.show()

figureObjects, axesObject = plt.subplots()

axesObject.pie(arr1,
               labels=objects,
               autopct='%1.2f',
               startangle=90)

axesObject.axis('equal')
plt.show()
```



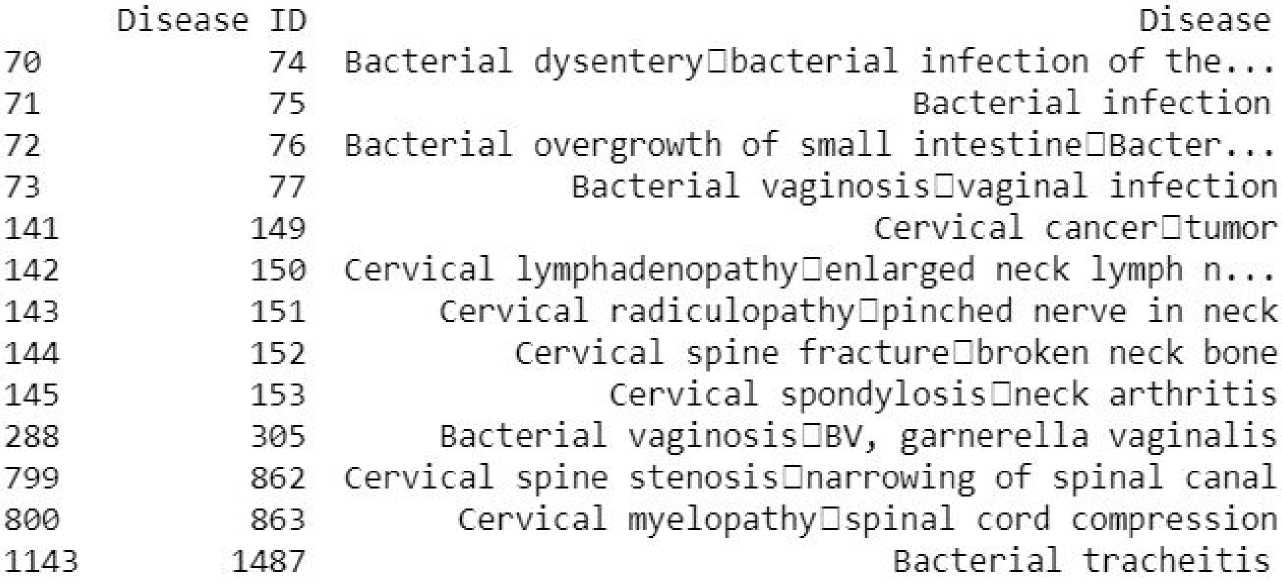


Inference : Most of the diseases are curable and does not cause any sort of major health issues

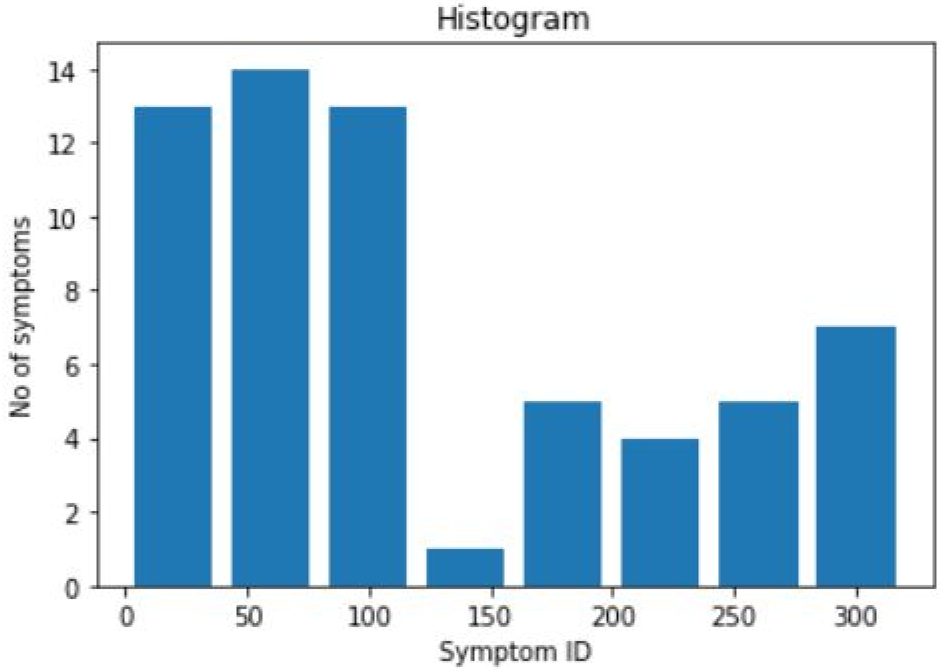
12. Perform analysis on the fact that the number of Bacterial and cervical diseases and their fatality rate using histogram



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

#data=pd.read_csv("datamatch.csv")
#dia=pd.read_csv("diagnose.csv")
#sym=pd.read_csv("symptom.csv")
dia.rename(columns = {"did": "Disease ID", "diagnose":"Disease"}, inplace = True)
sym.rename(columns = {"syd": "Symptom ID", "symptom":"Symptom"}, inplace = True)
data.rename(columns = {"did": "Disease ID", "syd":"Symptom ID","wei":"Fatality"}, inplace = True)
search_values=['Bacterial','Cervical']
tempdf4=dia.loc[dia['Disease'].str.contains('|'.join(search_values ))]
tempdf5=data.loc[data["Disease ID"].isin(tempdf4["Disease ID"])]
#print(tempdf5)
print(tempdf4)
l4=[]
for i in tempdf5['Symptom ID']:
    l3=[i]
    l4.extend(l3)
#print(l4)
population_age = l4
bins = [0,40,80,120,160,200,240,280,320]
plt.hist(population_age, bins, histtype='bar',rwidth=0.8)
plt.xlabel('age groups')
plt.ylabel('Number of people')
plt.title('Histogram')
plt.show()
```

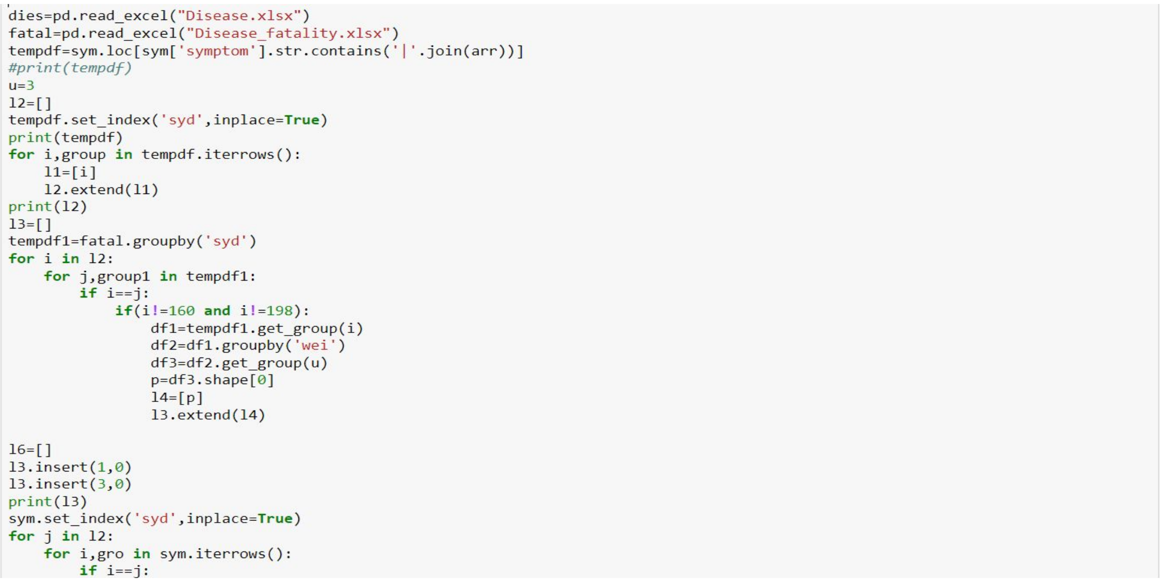


```
      Disease ID                                           Disease
70            74  Bacterial dysentery□bacterial infection of the...
71            75                               Bacterial infection
72            76  Bacterial overgrowth of small intestine□Bacter...
73            77                  Bacterial vaginosis□vaginal infection
141          149                               Cervical cancer□tumor
142          150  Cervical lymphadenopathy□enlarged neck lymph n...
143          151       Cervical radiculopathy□pinched nerve in neck
144          152           Cervical spine fracture□broken neck bone
145          153                Cervical spondylosis□neck arthritis
288          305       Bacterial vaginosis□BV, garnerella vaginalis
799          862  Cervical spine stenosis□narrowing of spinal canal
800          863        Cervical myelopathy□spinal cord compression
1143        1487                               Bacterial tracheitis
```

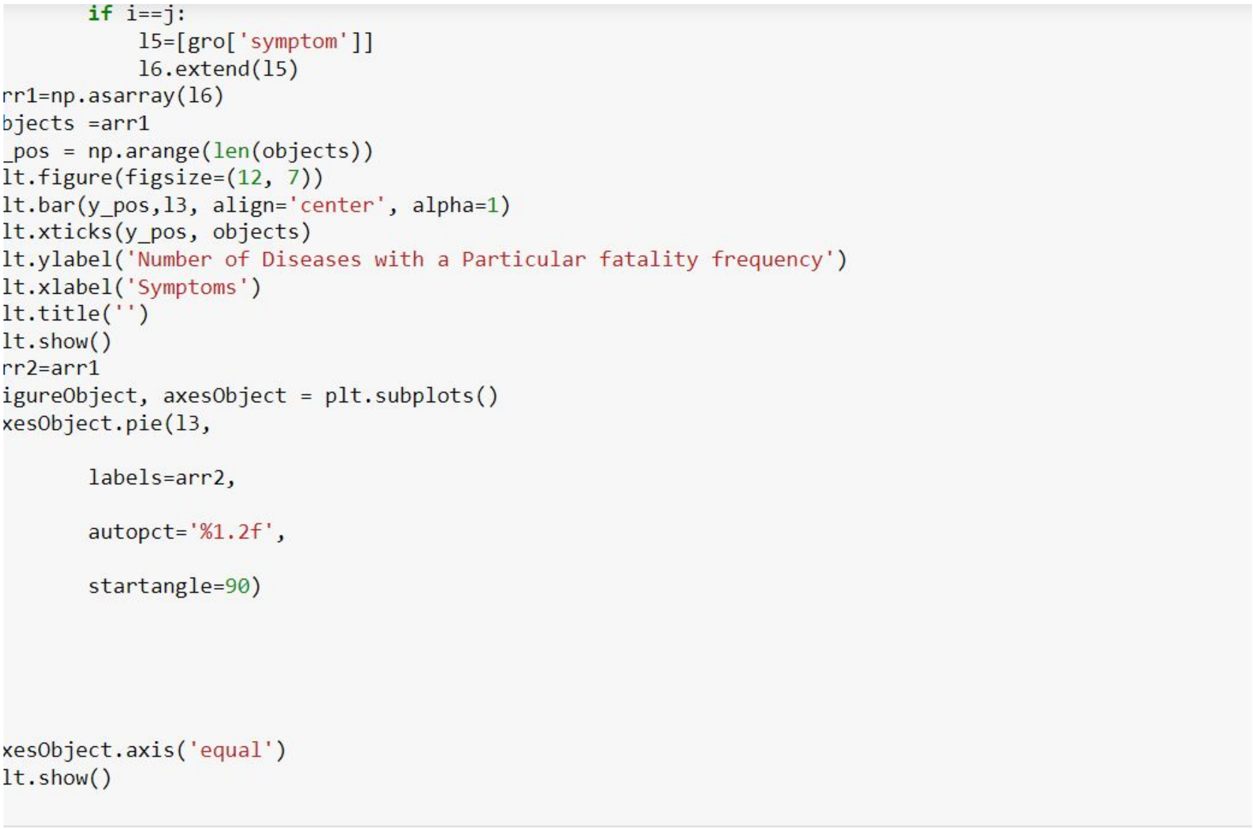


Inference : Neck stiffness or tightness is the most common symptom in bacterial and cervical diseases.

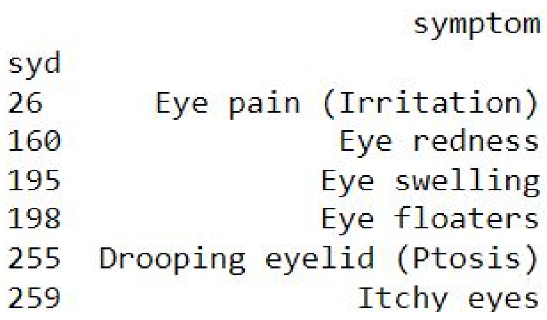
13. Perform analysis on the fact that the symptoms related to eye (irritation) and has a fatality rate equal to 3



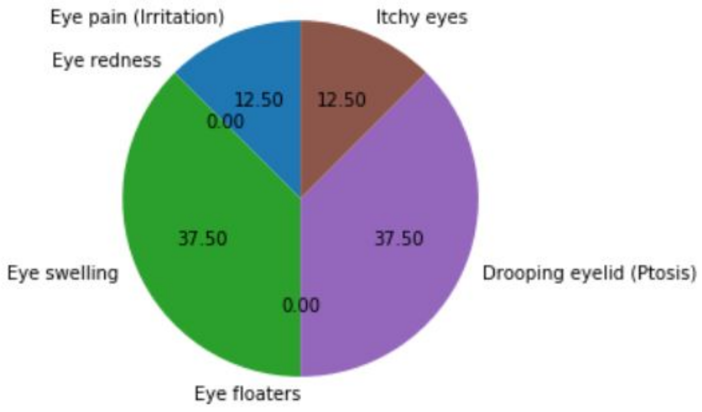
```
dies=pd.read_excel("Disease.xlsx")
fatal=pd.read_excel("Disease_fatality.xlsx")
tempdf=sym.loc[sym['symptom'].str.contains('|'.join(arr))]
#print(tempdf)
u=3
l2=[]
tempdf.set_index('syd',inplace=True)
print(tempdf)
for i,group in tempdf.iterrows():
    l1=[i]
    l2.extend(l1)
print(l2)
l3=[]
tempdf1=fatal.groupby('syd')
for i in l2:
    for j,group1 in tempdf1:
        if i==j:
            if(i!=160 and i!=198):
                df1=tempdf1.get_group(i)
                df2=df1.groupby('wei')
                df3=df2.get_group(u)
                p=df3.shape[0]
                l4=[p]
                l3.extend(l4)

l6=[]
l3.insert(1,0)
l3.insert(3,0)
print(l3)
sym.set_index('syd',inplace=True)
for j in l2:
    for i,gro in sym.iterrows():
        if i==j:
```

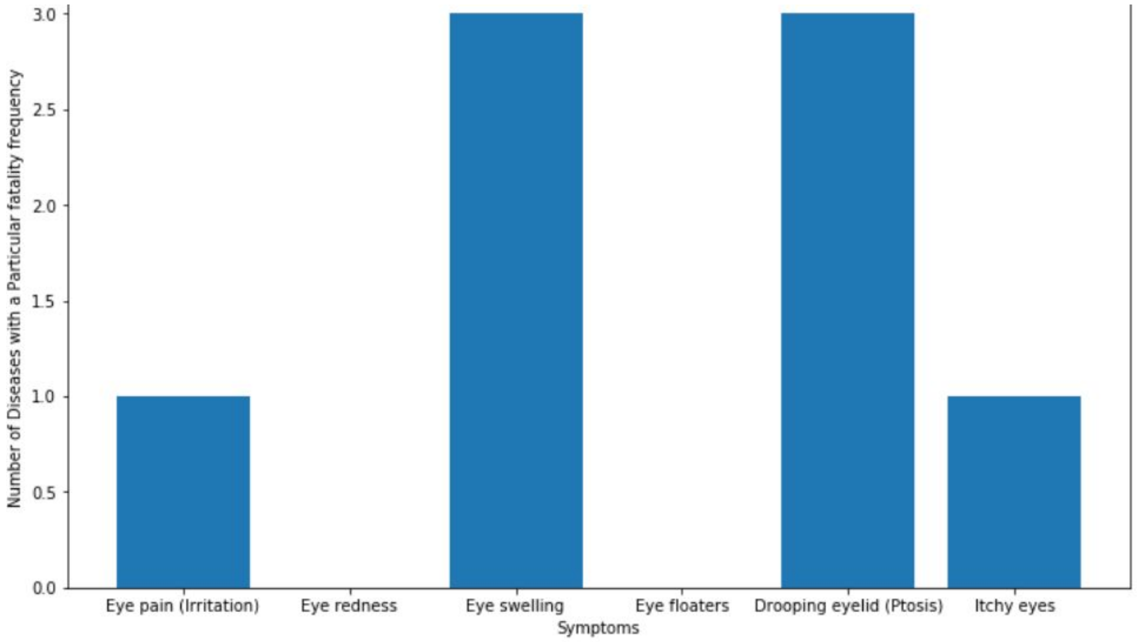


```
        if i==j:
            l5=[gro['symptom']]
            l6.extend(l5)
rr1=np.asarray(l6)
bjects =arr1
_pos = np.arange(len(objects))
lt.figure(figsize=(12, 7))
lt.bar(y_pos,l3, align='center', alpha=1)
lt.xticks(y_pos, objects)
lt.ylabel('Number of Diseases with a Particular fatality frequency')
lt.xlabel('Symptoms')
lt.title('')
lt.show()
rr2=arr1
igureObject, axesObject = plt.subplots()
xesObject.pie(l3,

        labels=arr2,

        autopct='%1.2f',

        startangle=90)




xesObject.axis('equal')
lt.show()
```



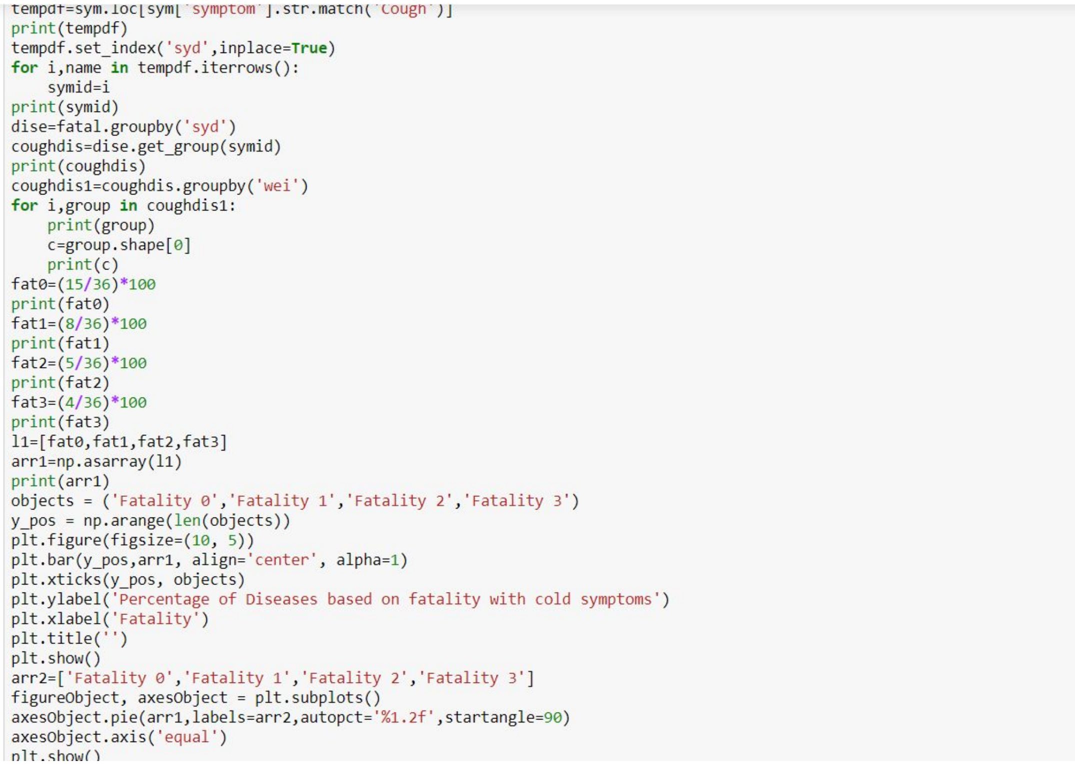
```
                          symptom
syd
26        Eye pain (Irritation)
160                 Eye redness
195                Eye swelling
198                Eye floaters
255   Drooping eyelid (Ptosis)
259                  Itchy eyes
```



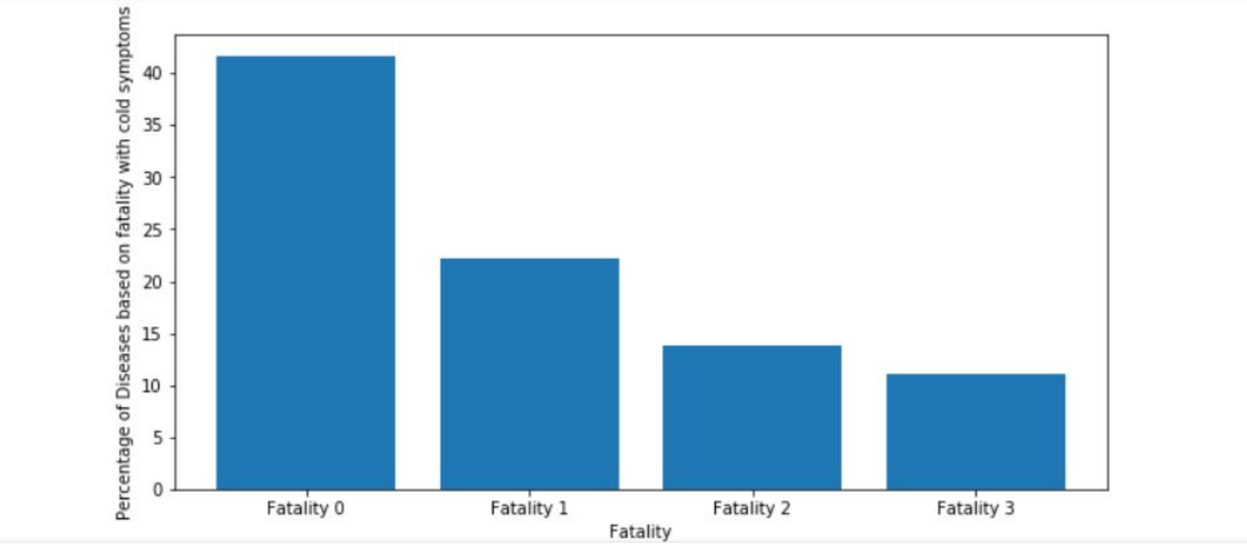


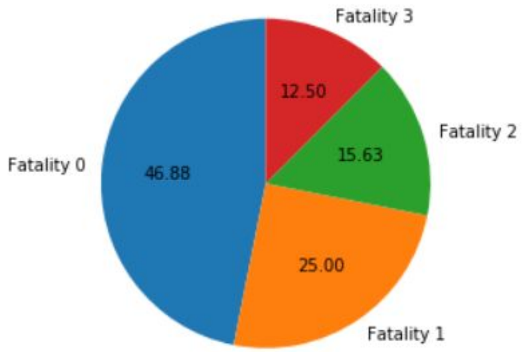
Inference : The most common eye related symptoms are “Eye swelling”,”Drooping eyelids” as they occur in 3 diseases.

14.Perform analysis on the fact that what is the chance of dying from cold symptoms(Graph)



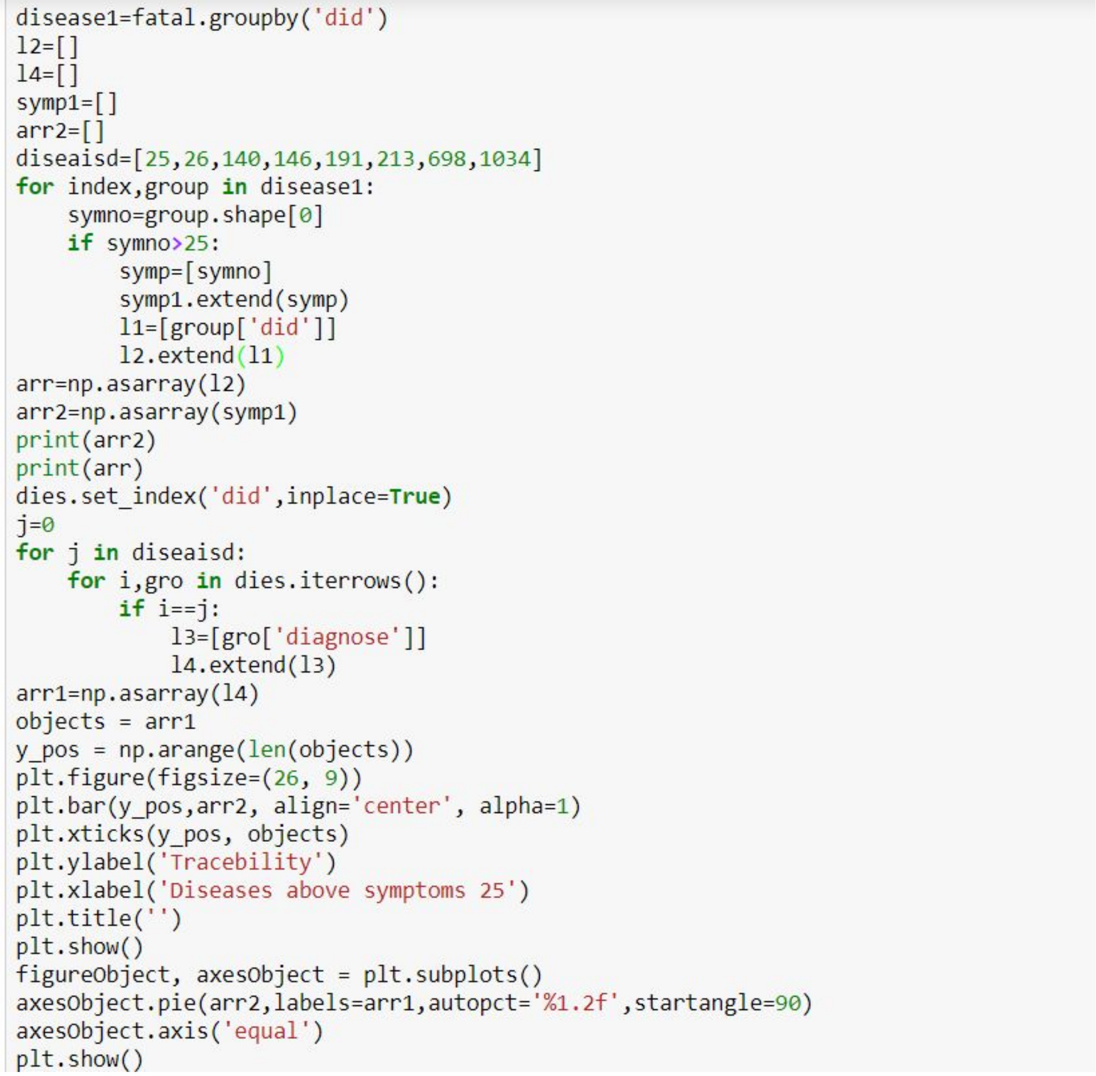
```
tempdf=sym.loc[sym['symptom'].str.match('Cough')]
print(tempdf)
tempdf.set_index('syd',inplace=True)
for i,name in tempdf.iterrows():
    symid=i
print(symid)
dise=fatal.groupby('syd')
coughdis=dise.get_group(symid)
print(coughdis)
coughdis1=coughdis.groupby('wei')
for i,group in coughdis1:
    print(group)
    c=group.shape[0]
    print(c)
fat0=(15/36)*100
print(fat0)
fat1=(8/36)*100
print(fat1)
fat2=(5/36)*100
print(fat2)
fat3=(4/36)*100
print(fat3)
l1=[fat0,fat1,fat2,fat3]
arr1=np.asarray(l1)
print(arr1)
objects = ('Fatality 0','Fatality 1','Fatality 2','Fatality 3')
y_pos = np.arange(len(objects))
plt.figure(figsize=(10, 5))
plt.bar(y_pos,arr1, align='center', alpha=1)
plt.xticks(y_pos, objects)
plt.ylabel('Percentage of Diseases based on fatality with cold symptoms')
plt.xlabel('Fatality')
plt.title('')
plt.show()
arr2=['Fatality 0','Fatality 1','Fatality 2','Fatality 3']
figureObject, axesObject = plt.subplots()
axesObject.pie(arr1,labels=arr2,autopct='%1.2f',startangle=90)
axesObject.axis('equal')
plt.show()
```



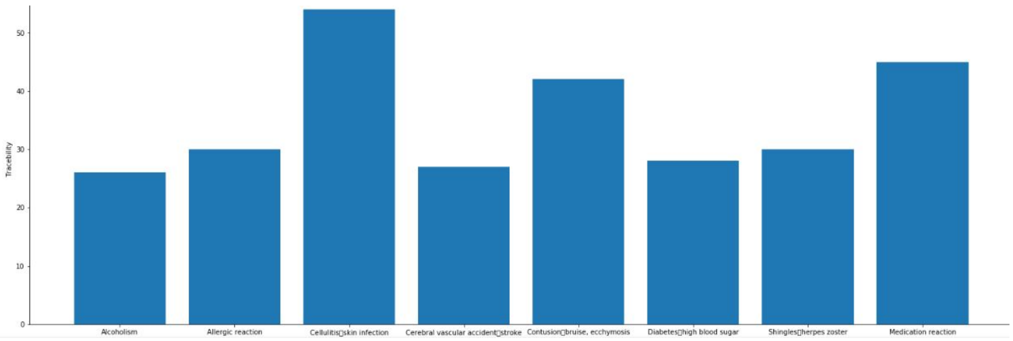


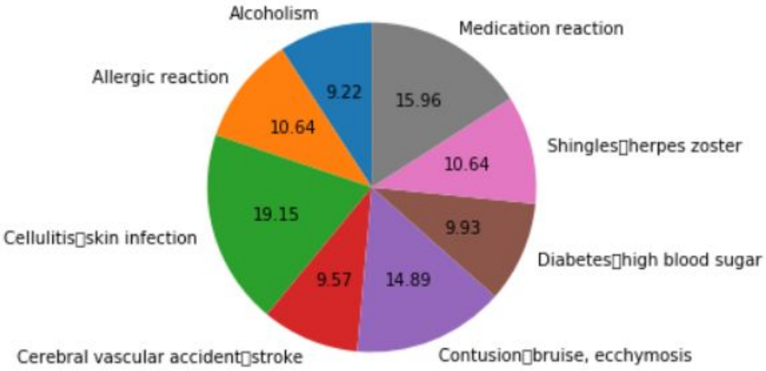
Inference : Cold symptoms do not pose much threat as most of the diseases related to cold symptoms have fatality rate 0.

15.Perform analysis on the fact that Diseases with more than twenty five symptoms



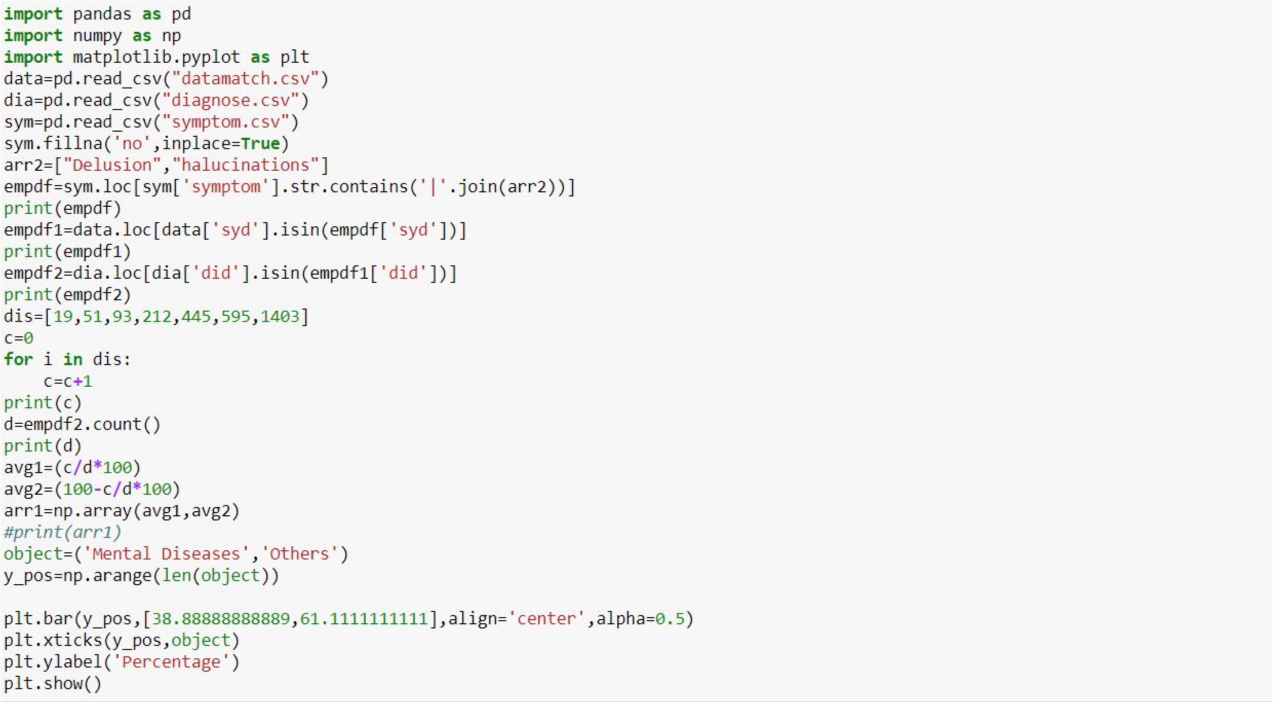
```
disease1=fatal.groupby('did')
l2=[]
l4=[]
symp1=[]
arr2=[]
diseaisd=[25,26,140,146,191,213,698,1034]
for index,group in disease1:
    symno=group.shape[0]
    if symno>25:
        symp=[symno]
        symp1.extend(symp)
        l1=[group['did']]
        l2.extend(l1)
arr=np.asarray(l2)
arr2=np.asarray(symp1)
print(arr2)
print(arr)
dies.set_index('did',inplace=True)
j=0
for j in diseaisd:
    for i,gro in dies.iterrows():
        if i==j:
            l3=[gro['diagnose']]
            l4.extend(l3)
arr1=np.asarray(l4)
objects = arr1
y_pos = np.arange(len(objects))
plt.figure(figsize=(26, 9))
plt.bar(y_pos,arr2, align='center', alpha=1)
plt.xticks(y_pos, objects)
plt.ylabel('Tracebility')
plt.xlabel('Diseases above symptoms 25')
plt.title('')
plt.show()
figureObject, axesObject = plt.subplots()
axesObject.pie(arr2,labels=arr1,autopct='%1.2f',startangle=90)
axesObject.axis('equal')
plt.show()
```



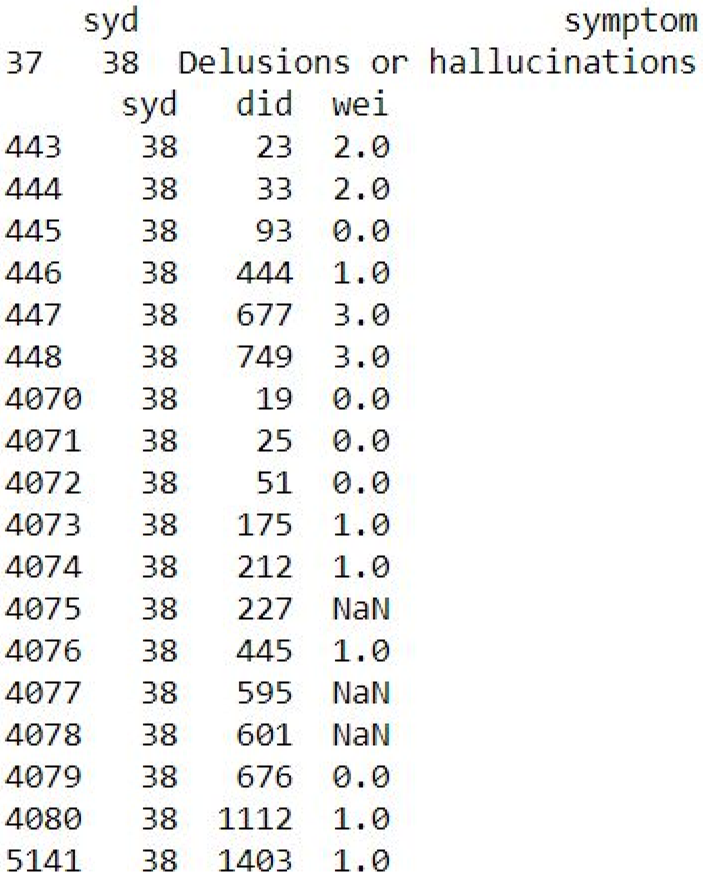


Inference : Diseases with Symptoms Above 25.Among them The easily Traceable Disease is Cellulitis Skin Infection

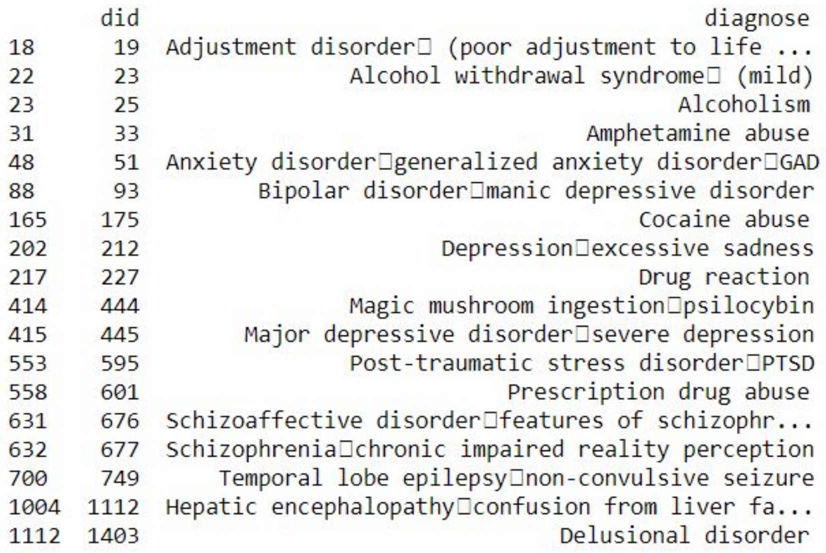
16.Perform analysis on the fact that If delusions and hallucinations occur in a patient then it leads to mental illness.



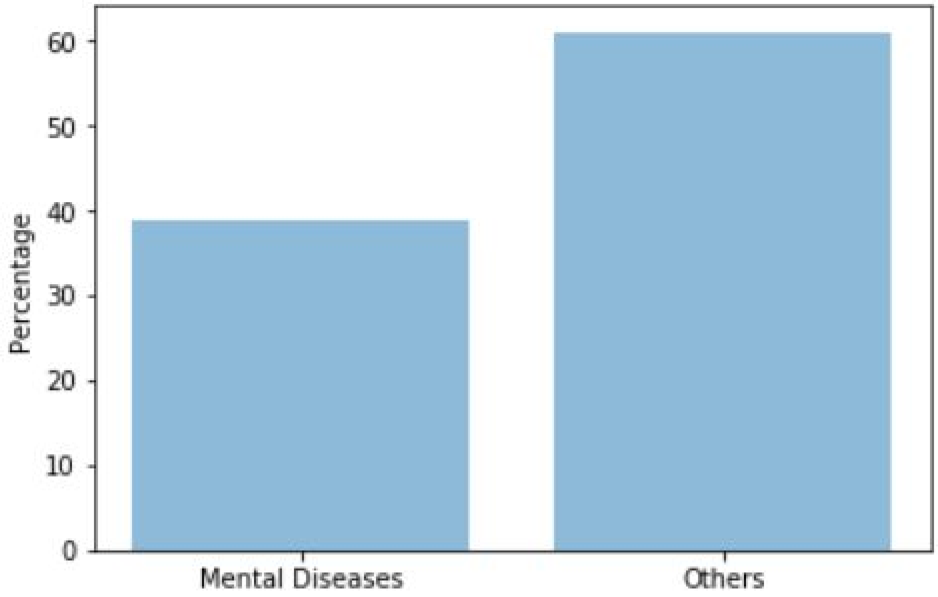
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
data=pd.read_csv("datamatch.csv")
dia=pd.read_csv("diagnose.csv")
sym=pd.read_csv("symptom.csv")
sym.fillna('no',inplace=True)
arr2=["Delusion","halucinations"]
empdf=sym.loc[sym['symptom'].str.contains('|'.join(arr2))]
print(empdf)
empdf1=data.loc[data['syd'].isin(empdf['syd'])]
print(empdf1)
empdf2=dia.loc[dia['did'].isin(empdf1['did'])]
print(empdf2)
dis=[19,51,93,212,445,595,1403]
c=0
for i in dis:
    c=c+1
print(c)
d=empdf2.count()
print(d)
avg1=(c/d*100)
avg2=(100-c/d*100)
arr1=np.array(avg1,avg2)
#print(arr1)
object=('Mental Diseases','Others')
y_pos=np.arange(len(object))

plt.bar(y_pos,[38.88888888889,61.1111111111],align='center',alpha=0.5)
plt.xticks(y_pos,object)
plt.ylabel('Percentage')
plt.show()
```



```
    syd                      symptom
37   38  Delusions or hallucinations
      syd   did  wei
443    38    23  2.0
444    38    33  2.0
445    38    93  0.0
446    38   444  1.0
447    38   677  3.0
448    38   749  3.0
4070   38    19  0.0
4071   38    25  0.0
4072   38    51  0.0
4073   38   175  1.0
4074   38   212  1.0
4075   38   227  NaN
4076   38   445  1.0
4077   38   595  NaN
4078   38   601  NaN
4079   38   676  0.0
4080   38  1112  1.0
5141   38  1403  1.0
```

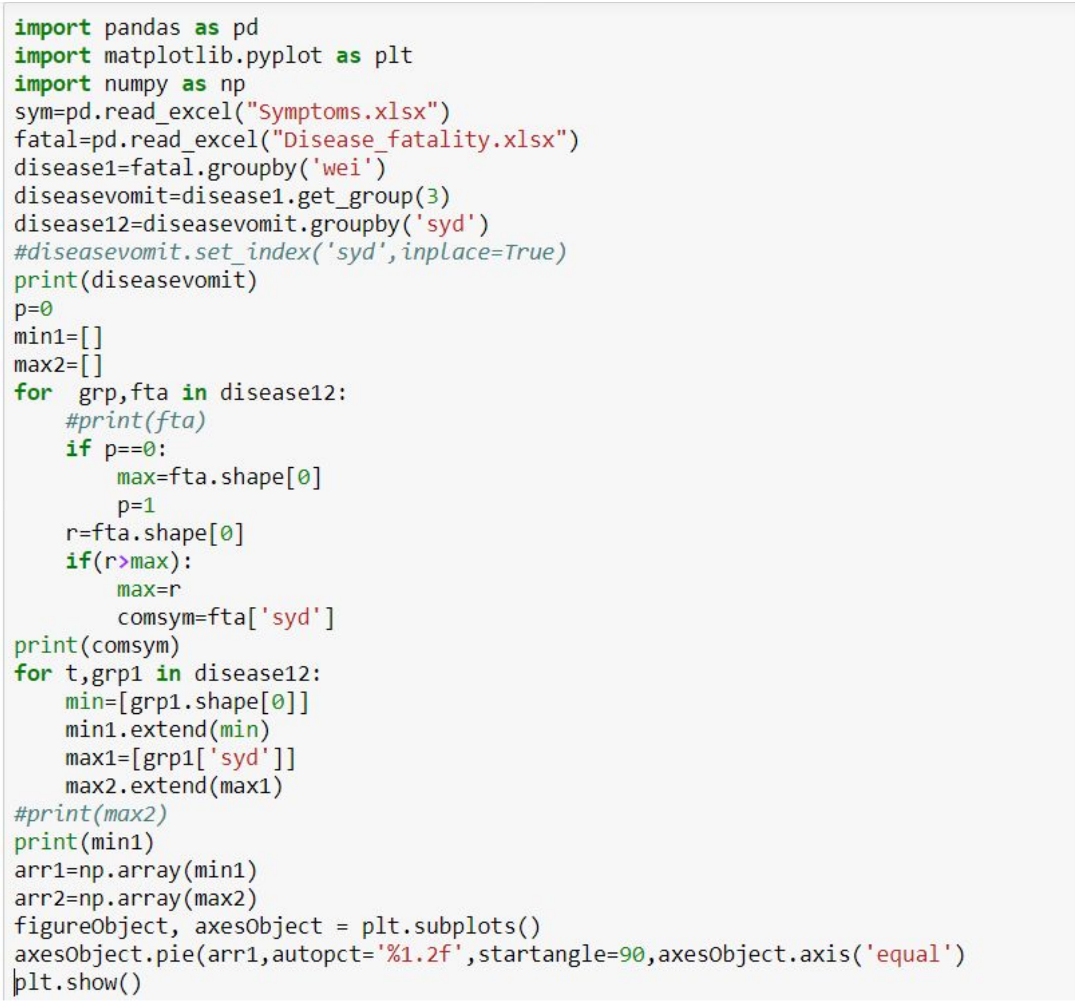


```
       did                                           diagnose
18      19  Adjustment disorder□ (poor adjustment to life ...
22      23               Alcohol withdrawal syndrome□ (mild)
23      25                                        Alcoholism
31      33                                  Amphetamine abuse
48      51  Anxiety disorder□generalized anxiety disorder□GAD
88      93          Bipolar disorder□manic depressive disorder
165    175                                      Cocaine abuse
202    212                       Depression□excessive sadness
217    227                                      Drug reaction
414    444                Magic mushroom ingestion□psilocybin
415    445         Major depressive disorder□severe depression
553    595                Post-traumatic stress disorder□PTSD
558    601                              Prescription drug abuse
631    676  Schizoaffective disorder□features of schizophr...
632    677  Schizophrenia□chronic impaired reality perception
700    749      Temporal lobe epilepsy□non-convulsive seizure
1004  1112  Hepatic encephalopathy□confusion from liver fa...
1112  1403                                  Delusional disorder
```

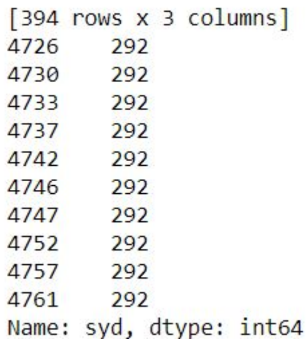


Inference : Delusions and hallucinations are not the main symptoms for mental illness.

17.Perform analysis on the fact that most common symptoms found in the diseases of highest fatality



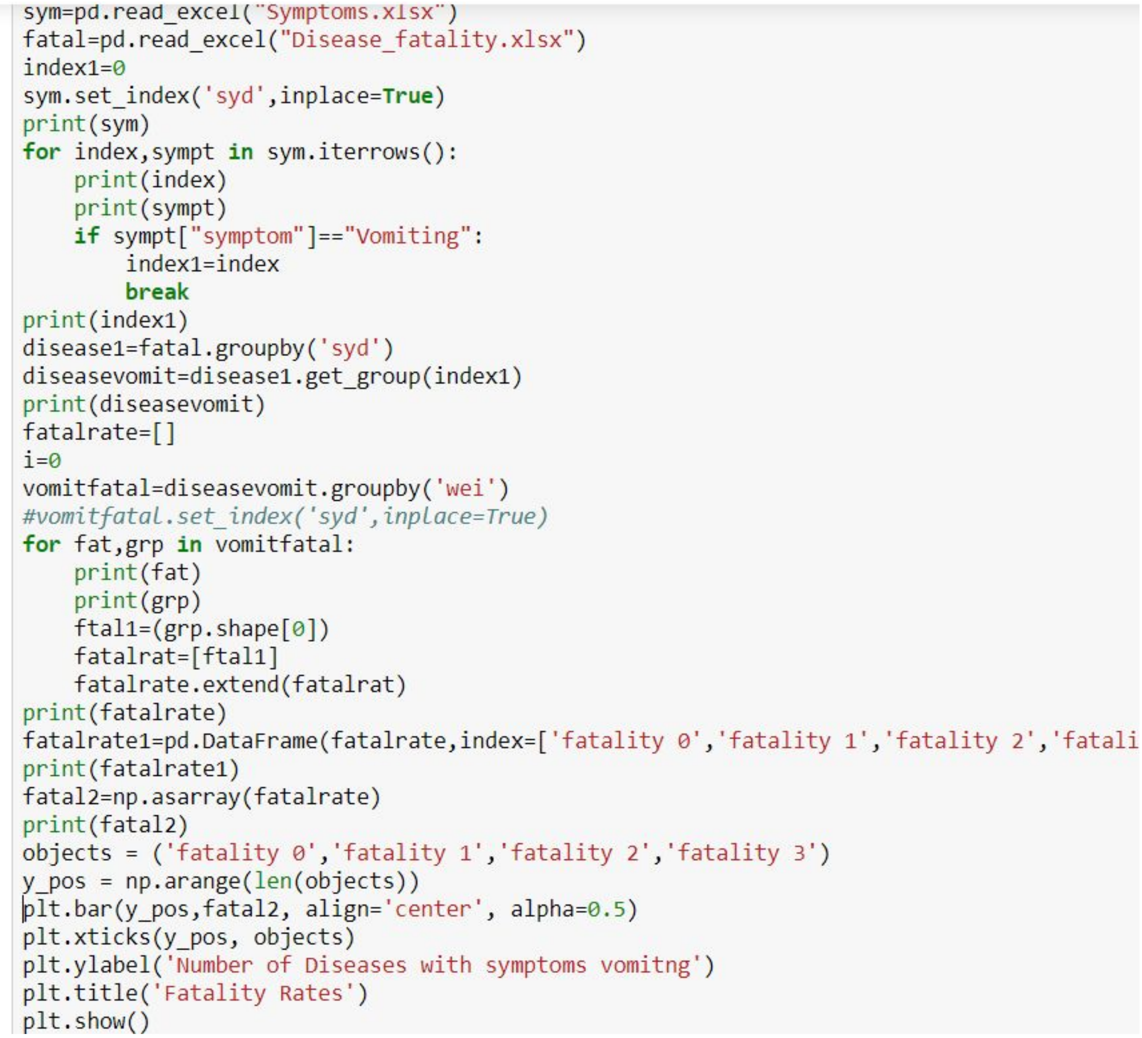
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
sym=pd.read_excel("Symptoms.xlsx")
fatal=pd.read_excel("Disease_fatality.xlsx")
disease1=fatal.groupby('wei')
diseasevomit=disease1.get_group(3)
disease12=diseasevomit.groupby('syd')
#diseasevomit.set_index('syd',inplace=True)
print(diseasevomit)
p=0
min1=[]
max2=[]
for  grp,fta in disease12:
    #print(fta)
    if p==0:
        max=fta.shape[0]
        p=1
    r=fta.shape[0]
    if(r>max):
        max=r
        comsym=fta['syd']
print(comsym)
for t,grp1 in disease12:
    min=[grp1.shape[0]]
    min1.extend(min)
    max1=[grp1['syd']]
    max2.extend(max1)
#print(max2)
print(min1)
arr1=np.array(min1)
arr2=np.array(max2)
figureObject, axesObject = plt.subplots()
axesObject.pie(arr1,autopct='%1.2f',startangle=90,axesObject.axis('equal')
plt.show()
```



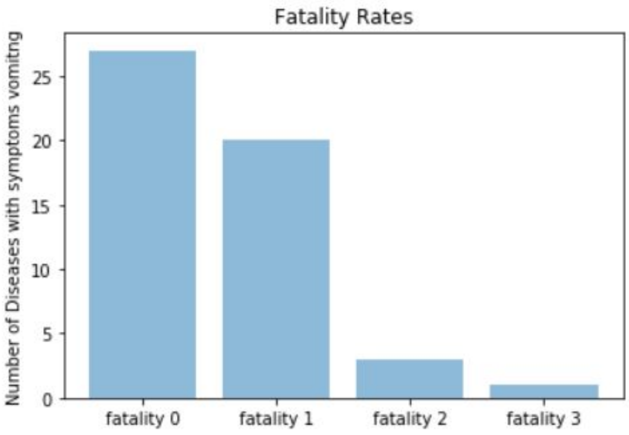
```
[394 rows x 3 columns]
4726    292
4730    292
4733    292
4737    292
4742    292
4746    292
4747    292
4752    292
4757    292
4761    292
Name: syd, dtype: int64
```

Inference : The most common symptom found among diseases which are most fatal is “Confusion”

18.Perform analysis on the fact that fatality rate of diseases which have symptoms of vomiting

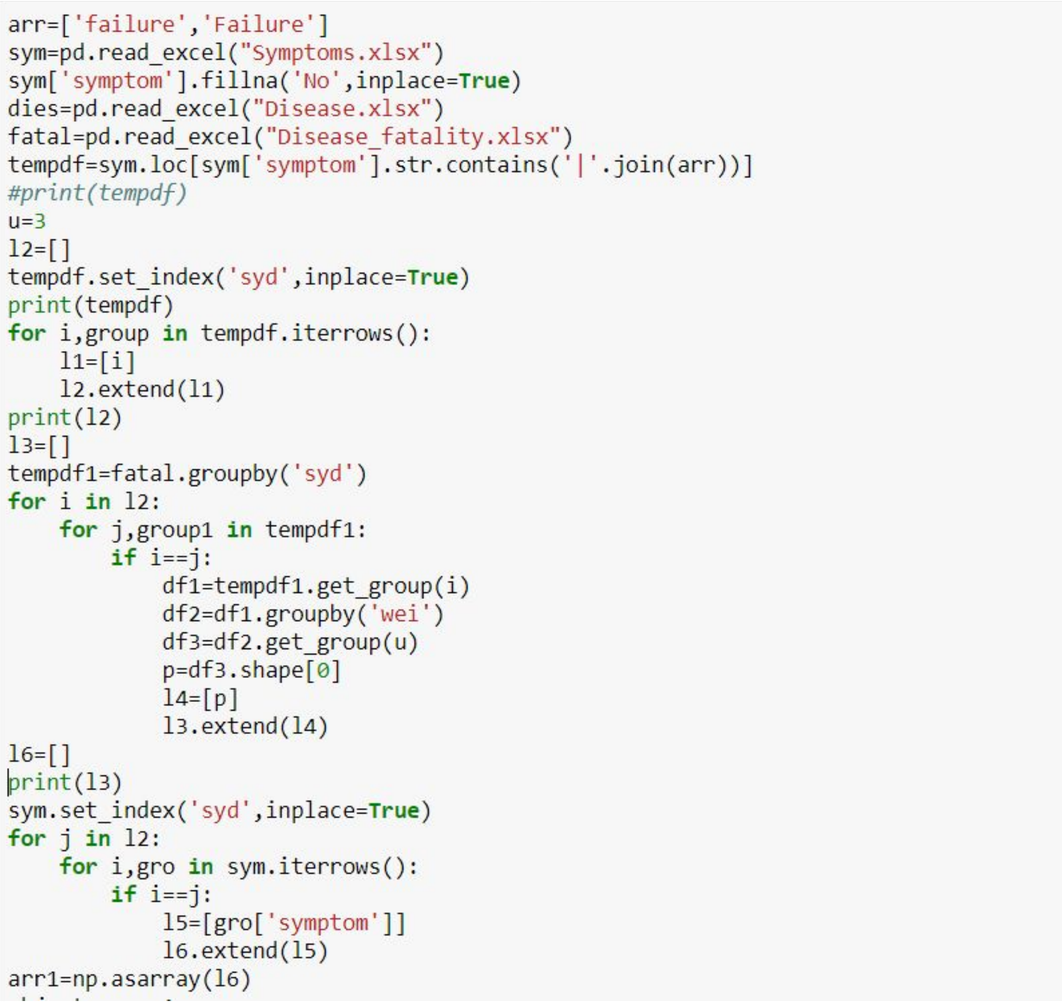


```
sym=pd.read_excel("Symptoms.xlsx")
fatal=pd.read_excel("Disease_fatality.xlsx")
index1=0
sym.set_index('syd',inplace=True)
print(sym)
for index,sympt in sym.iterrows():
    print(index)
    print(sympt)
    if sympt["symptom"]=="Vomiting":
        index1=index
        break
print(index1)
disease1=fatal.groupby('syd')
diseasevomit=disease1.get_group(index1)
print(diseasevomit)
fatalrate=[]
i=0
vomitfatal=diseasevomit.groupby('wei')
#vomitfatal.set_index('syd',inplace=True)
for fat,grp in vomitfatal:
    print(fat)
    print(grp)
    ftal1=(grp.shape[0])
    fatalrat=[ftal1]
    fatalrate.extend(fatalrat)
print(fatalrate)
fatalrate1=pd.DataFrame(fatalrate,index=['fatality 0','fatality 1','fatality 2','fatali
print(fatalrate1)
fatal2=np.asarray(fatalrate)
print(fatal2)
objects = ('fatality 0','fatality 1','fatality 2','fatality 3')
y_pos = np.arange(len(objects))
plt.bar(y_pos,fatal2, align='center', alpha=0.5)
plt.xticks(y_pos, objects)
plt.ylabel('Number of Diseases with symptoms vomitng')
plt.title('Fatality Rates')
plt.show()
```

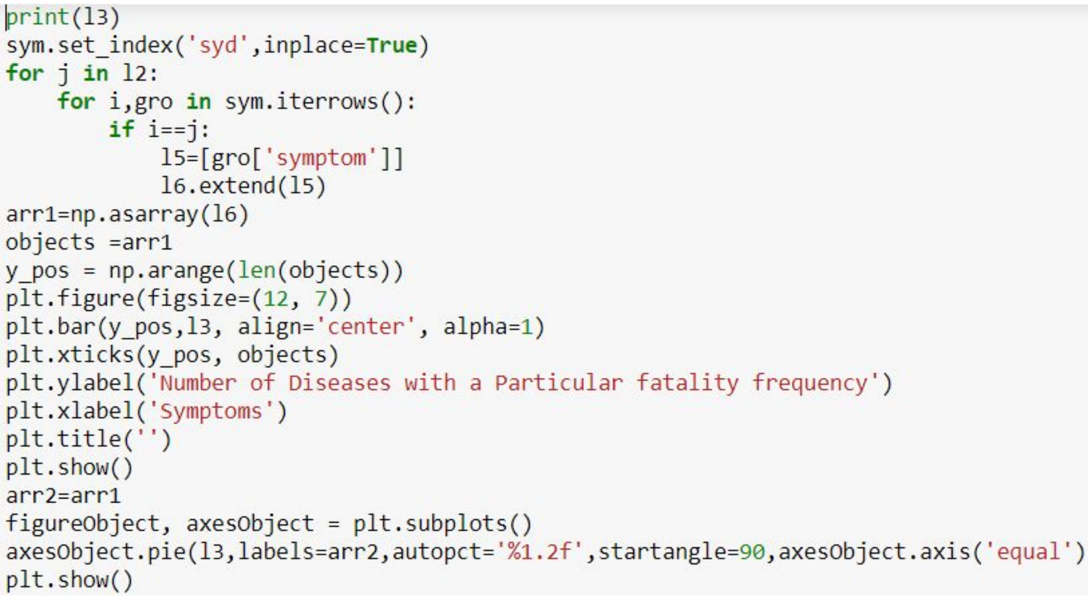


Inference : Vomiting causes the least fatal diseases.

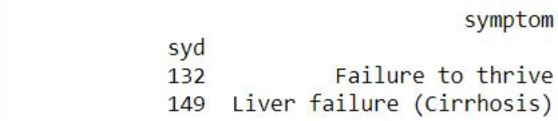
19. Perform analysis on the fact that the symptoms related to different types of organ failures and has a fatality rate equal to 3



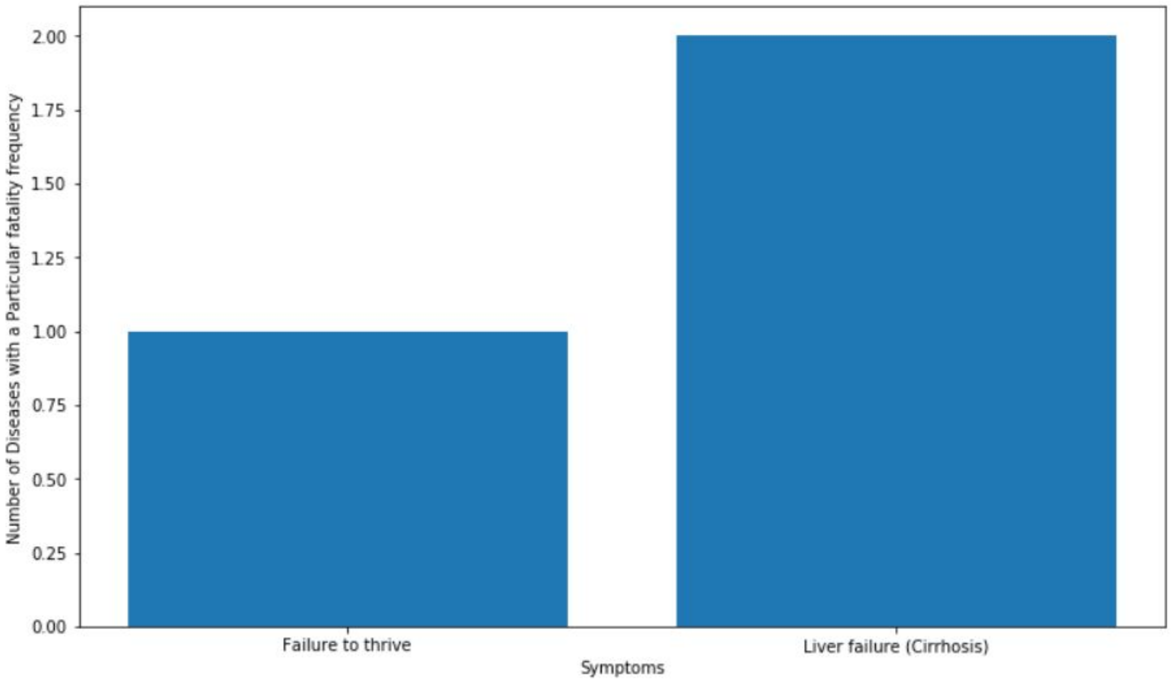
```
arr=['failure','Failure']
sym=pd.read_excel("Symptoms.xlsx")
sym['symptom'].fillna('No',inplace=True)
dies=pd.read_excel("Disease.xlsx")
fatal=pd.read_excel("Disease_fatality.xlsx")
tempdf=sym.loc[sym['symptom'].str.contains('|'.join(arr))]
#print(tempdf)
u=3
l2=[]
tempdf.set_index('syd',inplace=True)
print(tempdf)
for i,group in tempdf.iterrows():
    l1=[i]
    l2.extend(l1)
print(l2)
l3=[]
tempdf1=fatal.groupby('syd')
for i in l2:
    for j,group1 in tempdf1:
        if i==j:
            df1=tempdf1.get_group(i)
            df2=df1.groupby('wei')
            df3=df2.get_group(u)
            p=df3.shape[0]
            l4=[p]
            l3.extend(l4)
l6=[]
print(l3)
sym.set_index('syd',inplace=True)
for j in l2:
    for i,gro in sym.iterrows():
        if i==j:
            l5=[gro['symptom']]
            l6.extend(l5)
arr1=np.asarray(l6)
```

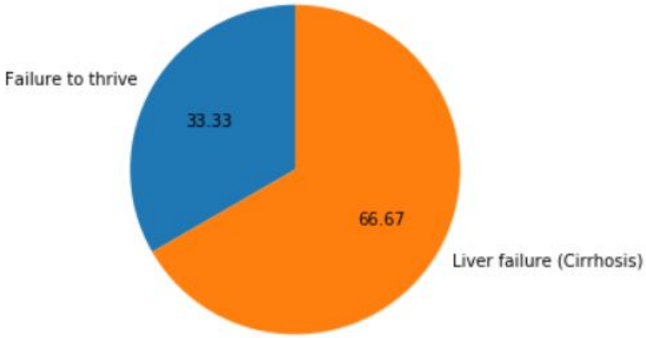


```
print(l3)
sym.set_index('syd',inplace=True)
for j in l2:
    for i,gro in sym.iterrows():
        if i==j:
            l5=[gro['symptom']]
            l6.extend(l5)
arr1=np.asarray(l6)
objects =arr1
y_pos = np.arange(len(objects))
plt.figure(figsize=(12, 7))
plt.bar(y_pos,l3, align='center', alpha=1)
plt.xticks(y_pos, objects)
plt.ylabel('Number of Diseases with a Particular fatality frequency')
plt.xlabel('Symptoms')
plt.title('')
plt.show()
arr2=arr1
figureObject, axesObject = plt.subplots()
axesObject.pie(l3,labels=arr2,autopct='%1.2f',startangle=90,axesObject.axis('equal')
plt.show()
```



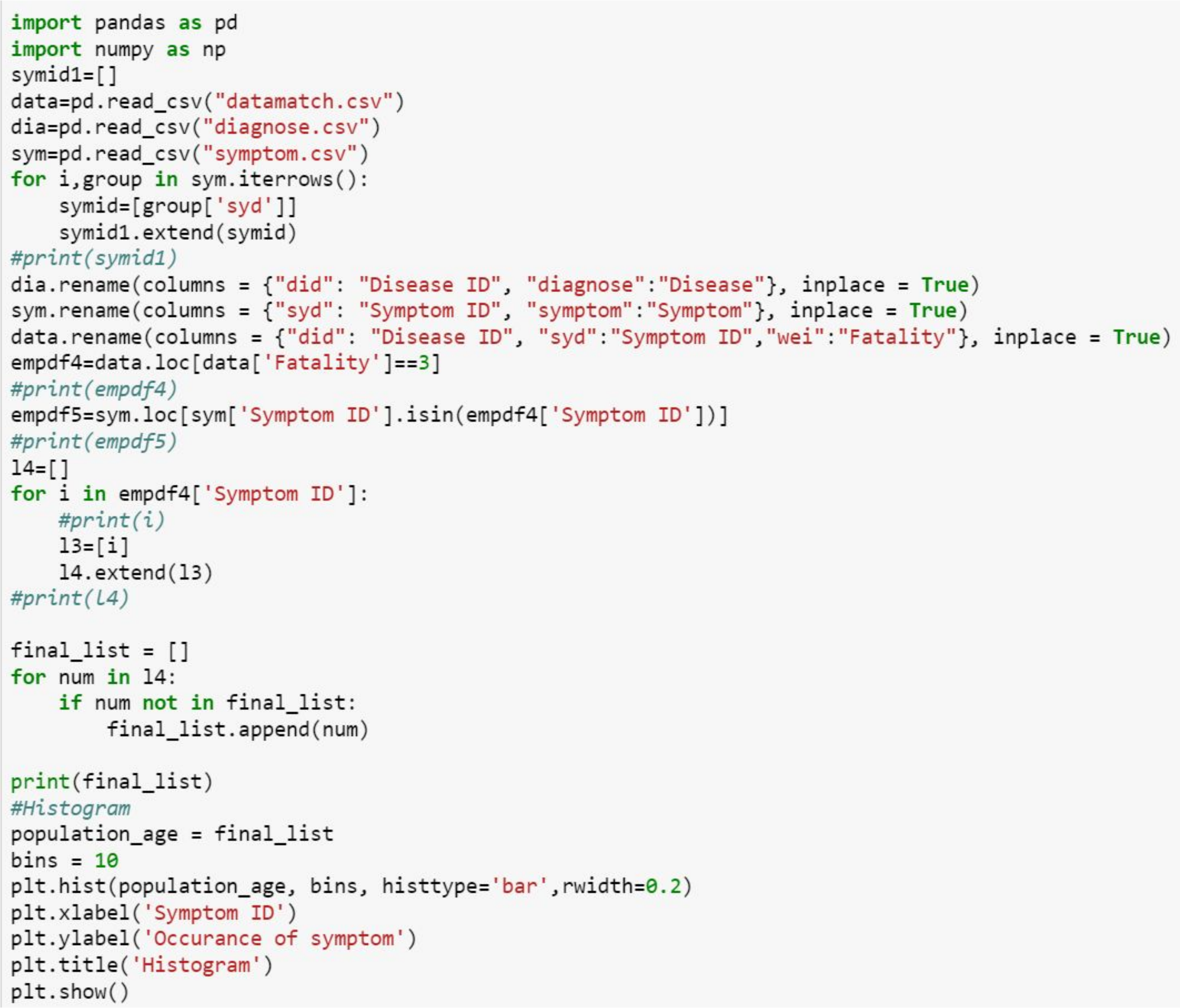
```
                      symptom
syd
132          Failure to thrive
149  Liver failure (Cirrhosis)
```



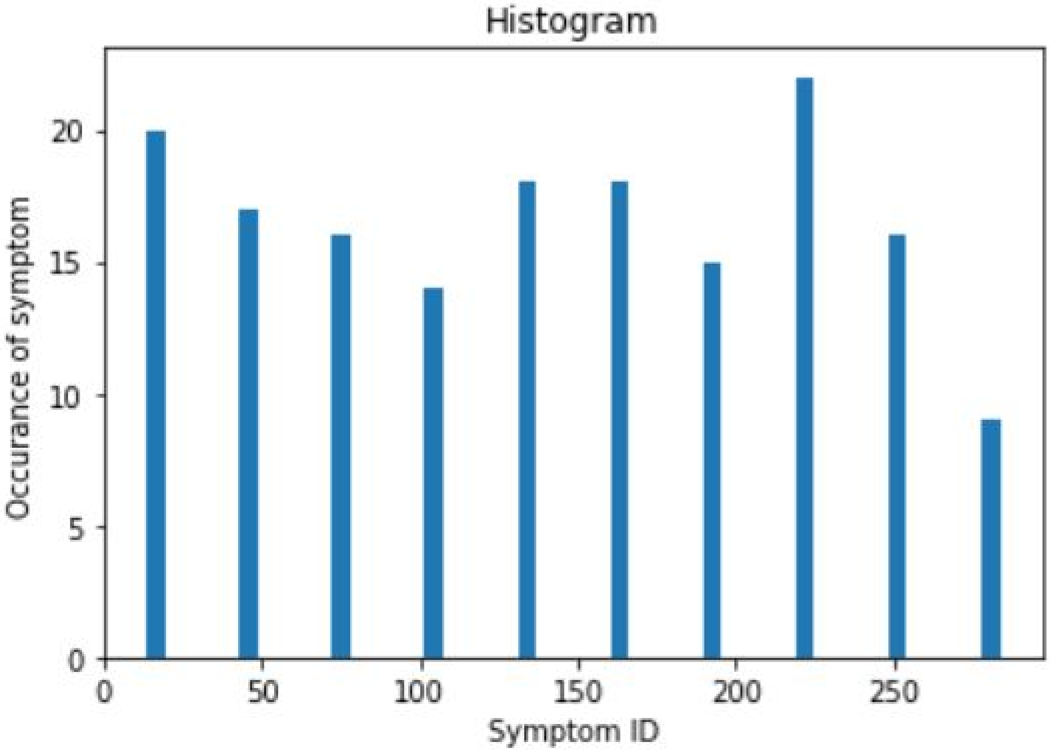


Inference : The most common organ failures are “Liver failure”,”Failure to thrive”. Liver failure being the most common amongst the two.

20. Perform analysis to find the occurrence of symptoms of highest fatality

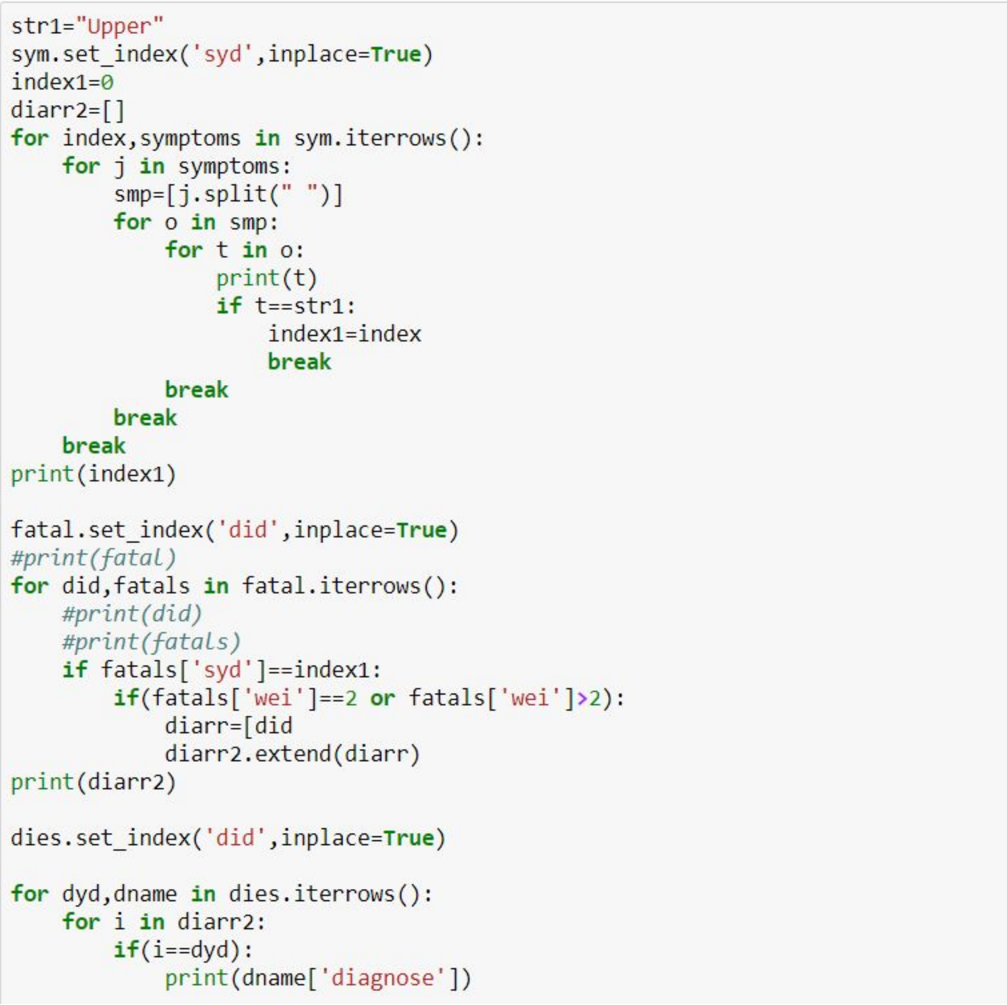


```
import pandas as pd
import numpy as np
symid1=[]
data=pd.read_csv("datamatch.csv")
dia=pd.read_csv("diagnose.csv")
sym=pd.read_csv("symptom.csv")
for i,group in sym.iterrows():
    symid=[group['syd']]
    symid1.extend(symid)
#print(symid1)
dia.rename(columns = {"did": "Disease ID", "diagnose":"Disease"}, inplace = True)
sym.rename(columns = {"syd": "Symptom ID", "symptom":"Symptom"}, inplace = True)
data.rename(columns = {"did": "Disease ID", "syd":"Symptom ID","wei":"Fatality"}, inplace = True)
empdf4=data.loc[data['Fatality']==3]
#print(empdf4)
empdf5=sym.loc[sym['Symptom ID'].isin(empdf4['Symptom ID'])]
#print(empdf5)
l4=[]
for i in empdf4['Symptom ID']:
    #print(i)
    l3=[i]
    l4.extend(l3)
#print(L4)

final_list = []
for num in l4:
    if num not in final_list:
        final_list.append(num)

print(final_list)
#Histogram
population_age = final_list
bins = 10
plt.hist(population_age, bins, histtype='bar',rwidth=0.2)
plt.xlabel('Symptom ID')
plt.ylabel('Occurance of symptom')
plt.title('Histogram')
plt.show()
```



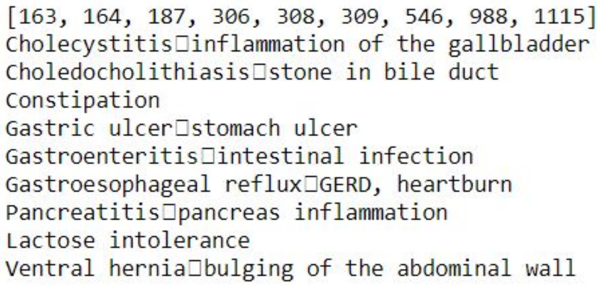
Inference : The most common Symptoms found of fatality 3 are “Confusion”,”Change in behaviour”,”Fainting”,”Fever in the returning traveller”,”Ingestion” and “Headache after trauma”.

21. Perform analysis on the fact that diseases with symptoms of upper abdominal pain whose fatality is greater than or equal to 2



```python
str1="Upper"
sym.set_index('syd',inplace=True)
index1=0
diarr2=[]
for index,symptoms in sym.iterrows():
    for j in symptoms:
        smp=[j.split(" ")]
        for o in smp:
            for t in o:
                print(t)
                if t==str1:
                    index1=index
                    break
            break
        break
    break
print(index1)

fatal.set_index('did',inplace=True)
#print(fatal)
for did,fatals in fatal.iterrows():
    #print(did)
    #print(fatals)
    if fatals['syd']==index1:
        if(fatals['wei']==2 or fatals['wei']>2):
            diarr=[did
            diarr2.extend(diarr)
print(diarr2)

dies.set_index('did',inplace=True)

for dyd,dname in dies.iterrows():
    for i in diarr2:
        if(i==dyd):
            print(dname['diagnose'])
```

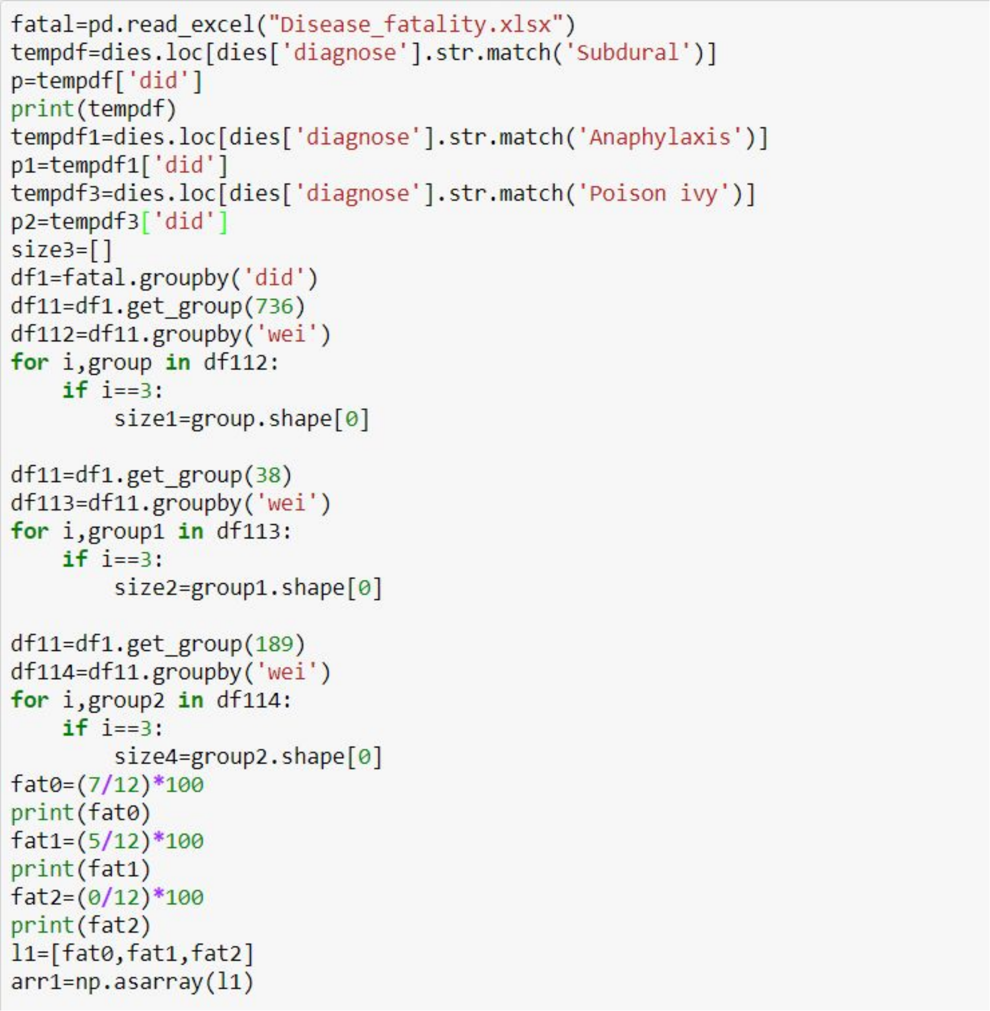
Output:



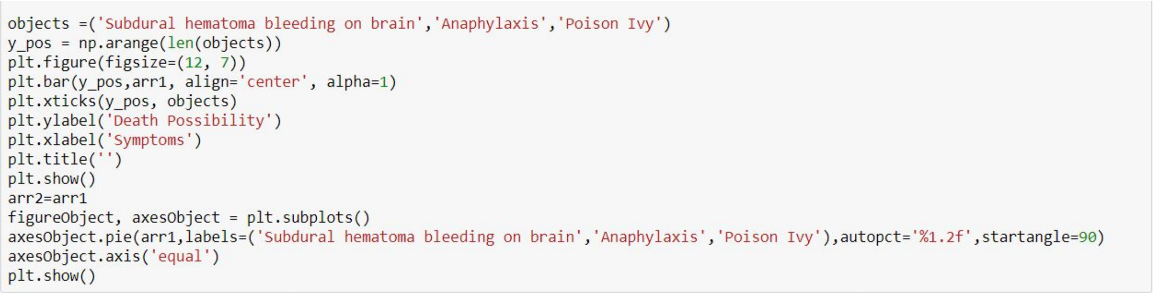
```
[163, 164, 187, 306, 308, 309, 546, 988, 1115]
Cholecystitis□inflammation of the gallbladder
Choledocholithiasis□stone in bile duct
Constipation
Gastric ulcer□stomach ulcer
Gastroenteritis□intestinal infection
Gastroesophageal reflux□GERD, heartburn
Pancreatitis□pancreas inflammation
Lactose intolerance
Ventral hernia□bulging of the abdominal wall
```

Inference: Most of the diseases related to upper abdominal pain are gastric problem

22. Perform analysis to calculate the Death percentage in top 3 Diseases.

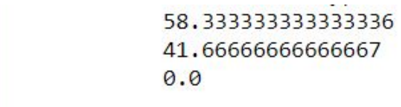


```
fatal=pd.read_excel("Disease_fatality.xlsx")
tempdf=dies.loc[dies['diagnose'].str.match('Subdural')]
p=tempdf['did']
print(tempdf)
tempdf1=dies.loc[dies['diagnose'].str.match('Anaphylaxis')]
p1=tempdf1['did']
tempdf3=dies.loc[dies['diagnose'].str.match('Poison ivy')]
p2=tempdf3['did']
size3=[]
df1=fatal.groupby('did')
df11=df1.get_group(736)
df112=df11.groupby('wei')
for i,group in df112:
    if i==3:
        size1=group.shape[0]

df11=df1.get_group(38)
df113=df11.groupby('wei')
for i,group1 in df113:
    if i==3:
        size2=group1.shape[0]

df11=df1.get_group(189)
df114=df11.groupby('wei')
for i,group2 in df114:
    if i==3:
        size4=group2.shape[0]
fat0=(7/12)*100
print(fat0)
fat1=(5/12)*100
print(fat1)
fat2=(0/12)*100
print(fat2)
l1=[fat0,fat1,fat2]
arr1=np.asarray(l1)
```

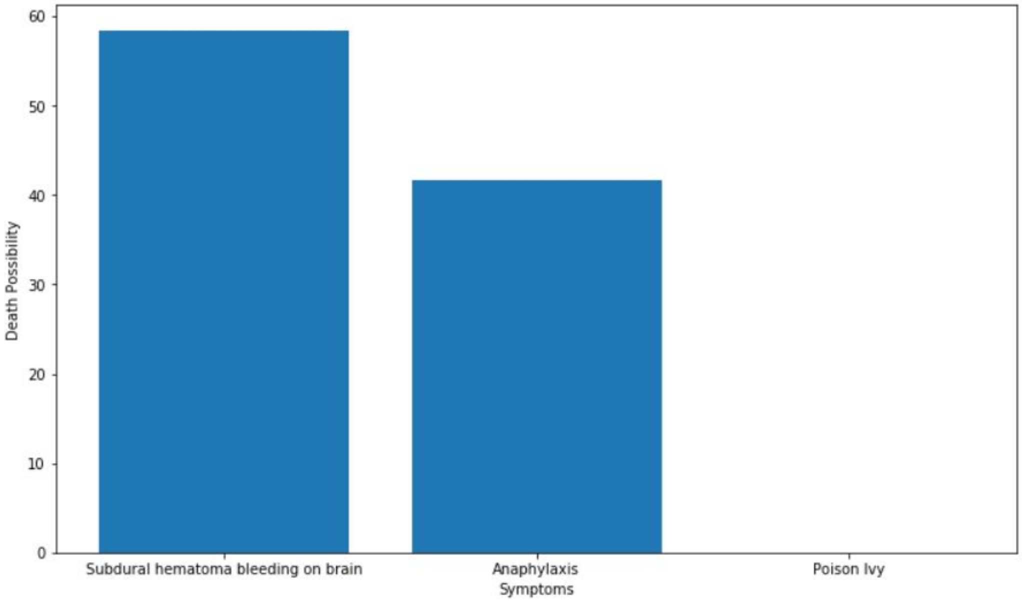


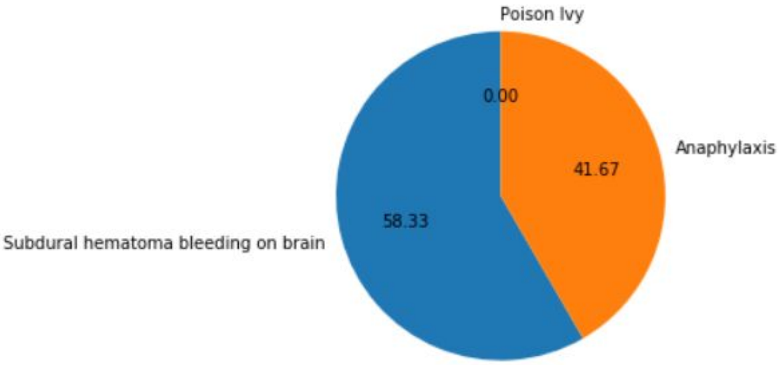
```
objects =('Subdural hematoma bleeding on brain','Anaphylaxis','Poison Ivy')
y_pos = np.arange(len(objects))
plt.figure(figsize=(12, 7))
plt.bar(y_pos,arr1, align='center', alpha=1)
plt.xticks(y_pos, objects)
plt.ylabel('Death Possibility')
plt.xlabel('Symptoms')
plt.title('')
plt.show()
arr2=arr1
figureObject, axesObject = plt.subplots()
axesObject.pie(arr1,labels=('Subdural hematoma bleeding on brain','Anaphylaxis','Poison Ivy'),autopct='%1.2f',startangle=90)
axesObject.axis('equal')
plt.show()
```

Death Percentage:



```
58.333333333333336
41.66666666666667
0.0
```





Inference: The Death percentage of Subdural hematoma bleeding on brain is 58.33

The Death percentage of Anaphylaxis is 41.33

The Death percentage of Poison Ivy is 0.0