Data Analysis Using Python

(Winter School Project)

January, 2018

Data Analysis on Diseases

Group Members:

Soham Banerjee, RCCIIT, 171170110221
Soumyajyoti Das, RCCIIT, 171170110226
Sourav Sarkar, RCCIIT, 171170110229
Biswadip Dawn, RCCIIT, 171170110271
Rohit Basu, RCCIIT, 171170110323
Subhajit Das, RCCIIT, 171170110354

Table of Contents

Acknowledgement

We take this opportunity to express our profound gratitude and deep regards to our faculty Mr. Kaushik Ghosh for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him from time to time shall carry us a long way in the journey of life on which we are about to embark.

We are obliged to our fellow project team members for the valuable information provided by them in their respective fields. We are grateful for their cooperation during the period of our assignment.

Soham Banerjee
Soumyajyoti Das
Sourav Sarkar
Biswadip Dawn
Rohit Basu
Subhajit Das

Project Objective

The main objective of this project is to interpret and analyse given datasets in the field of disease and healthcare. We would try to gain valuable insights and understand trends in the datasets by using python as a tool for data manipulation and analysis.

Project Scope

In this project we used numpy, pandas and matplotlib for data analysis. We focused on data analysis and interpretation using series and dataframes. The following libraries were imported for the project:

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

Data Description

The entire data is divided into 3 datasets.

1. Symptom Data:
   Contains two columns - ‘did’ and ‘diagnose’, and 272 rows.

2. Diagnosis Data:
   Contains 2 columns - ‘syd’ and ‘symptom’, and 1166 rows.

3. Symptom vs Diagnosis Data:
   Contains 3 columns - ‘did’, ’syd’ and ‘wei’, and 5569 rows.

Data Loading

The data was provided in 3 different .csv files.

For operating and manipulating the data in those files we had to store them in some python data structure.

Hence, the data was loaded into 3 separate dataframes using the pandas read_csv function.

```
df = pd.read_csv("diffsydiw.csv")
dia = pd.read_csv("dia_t.csv")
sym = pd.read_csv("sym_t.csv")
```

Using dataframes we can perform various tasks like grouping, plotting and sorting the data.

Interpreting the Data

There were 3 datasets for the given task.

1. Symptom Dataset:

It contains 2 rows - ‘symptom id’ and ‘symptom’. The ‘symptom id’ is a unique number representing each symptom. The symptom column contains the names of the symptoms. The symptom id consists of 272 numbers within the range of 1 to 306.

2. Diagnosis Dataset:

It contains 2 rows - ‘diagnosis id’ and ‘diagnose’. The ‘diagnosis id’ is a unique number representing each diagnosis. The diagnosis column contains the names of the various diagnoses. The diagnosis id consists of 1166 numbers within the range of 1 to 1537.

3. Symptom vs Diagnosis Dataset:

It contains 3 rows - ‘symptom id’, ‘diagnosis id’ and ‘weight’. It provides a correspondence map for each symptom and diagnosis, along with their respective weights.

A few points that we noted:

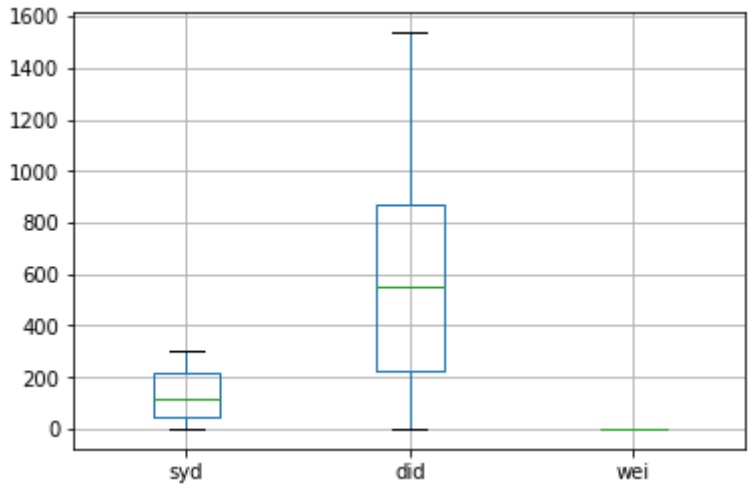
1. A symptom may have different weight based on the corresponding diagnosis.

2. A disease (diagnosis) may have different weight based on the corresponding symptom.

Distribution Analysis

The data provided to us was not quantitative in nature. Hence, distribution analysis of the data did not yield any significant result. No outliers or odd data bits were found.

```
df.boxplot()
```



Data Cleaning and munging

While dealing with the given datasets 5 problems arose.

1. Missing values in the symptom dataset.
2. Missing values in the symptom vs diagnose dataset.
3. Dirty data in the diagnose dataset.
4. Improper column nomenclature in the datasets.
5. Improper nomenclature of the elements of the 'wei' column.

The problems were dealt with in the following ways.

1. The missing values in the 'symptom' column was replaced by the word 'unnamed'. It implies that the symptom was discovered but not named.

The following code was used:

```
sym['symptom'] = sym.apply(lambda x: x['symptom']\
    if type(x['symptom'])==str else 'unnamed',axis=1)
```

The head(40) data before replacement was:

	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
30	31	Fever of unknown origin
31	32	Flank pain
32	33	Frequent urination (Frequency)
33	34	NaN
34	35	Foot pain
35	36	Cranky, crying, fussy, irritable child
36	37	Groin pain
37	38	Delusions or hallucinations
38	39	Hand, finger ache or pain
39	40	Head injury

The head(40) data after replacement operation is:

	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
30	31	Fever of unknown origin
31	32	Flank pain
32	33	Frequent urination (Frequency)
33	34	unnamed
34	35	Foot pain
35	36	Cranky, crying, fussy, irritable child
36	37	Groin pain
37	38	Delusions or hallucinations
38	39	Hand, finger ache or pain
39	40	Head injury

2. The missing values in the ‘wei’ column of the ‘Symptom vs Disease’ dataset fell under two categories.

i. Missing or NaN values.
ii. Zeroes

To overcome the difficulty of dealing with both the cases separately, we filled all the NaN values with 0.

```
df = df.fillna(0)
```

After that we grouped the ‘Symptom vs Disease’ dataset by the ‘did’. Here, ‘did’ stands for ‘disease id’ or ‘diagnosis id’.
Now we would look for the ‘wei’ values for a particular disease.
There can be three possible cases:

i. All the values are 0.
ii. Some of the values are 0 and some non-zero.

iii. All the values are non-zero.

In case of all zeroes(i) we will replace all the zeroes with 2. (We will assume it to be life threatening.)
In case of some zero and some non-zero (ii) values the zeroes are replaced by the ceiling value of the mean of all the non-zero values.
In case (iii) no values will be replaced.

```
def change(x):
    grp = df.groupby('did')
    temp = grp.get_group(x)
    m = temp['wei'].mean()
    if m==0:
        return 2
    else:
        return math.ceil(m)
df['wei'] = df.apply(lambda x : change(x['did']) if x['wei']==0 else x['wei']\
  ,axis=1)
```

head(40) data before replacement:

	syd	did	wei
0	1.0	163.0	2.0
1	1.0	164.0	2.0
2	1.0	165.0	1.0
3	1.0	187.0	2.0
4	1.0	306.0	2.0
5	1.0	307.0	1.0
6	1.0	308.0	2.0
7	1.0	309.0	2.0
8	1.0	354.0	1.0
9	1.0	401.0	1.0
10	1.0	411.0	1.0
11	1.0	513.0	1.0
12	1.0	546.0	2.0
13	1.0	722.0	1.0
14	2.0	56.0	3.0
15	2.0	179.0	2.0
16	2.0	236.0	1.0
17	2.0	388.0	2.0
18	2.0	539.0	1.0
19	2.0	540.0	1.0
20	2.0	557.0	1.0
21	2.0	600.0	1.0
22	2.0	793.0	2.0
23	2.0	795.0	1.0
24	3.0	44.0	1.0
25	3.0	106.0	1.0
26	3.0	108.0	0.0
27	3.0	707.0	2.0
28	3.0	209.0	2.0
29	3.0	250.0	1.0
30	3.0	294.0	0.0
31	3.0	432.0	0.0
32	3.0	439.0	1.0
33	3.0	568.0	3.0
34	3.0	660.0	1.0
35	3.0	720.0	1.0
36	4.0	20.0	1.0
37	4.0	22.0	2.0
38	4.0	23.0	2.0
39	4.0	25.0	1.0

head(40) data after replacement:

	syd	did	wei
0	1.0	163.0	2.0
1	1.0	164.0	2.0
2	1.0	165.0	1.0
3	1.0	187.0	2.0
4	1.0	306.0	2.0
5	1.0	307.0	1.0
6	1.0	308.0	2.0
7	1.0	309.0	2.0
8	1.0	354.0	1.0
9	1.0	401.0	1.0
10	1.0	411.0	1.0
11	1.0	513.0	1.0
12	1.0	546.0	2.0
13	1.0	722.0	1.0
14	2.0	56.0	3.0
15	2.0	179.0	2.0
16	2.0	236.0	1.0
17	2.0	388.0	2.0
18	2.0	539.0	1.0
19	2.0	540.0	1.0
20	2.0	557.0	1.0
21	2.0	600.0	1.0
22	2.0	793.0	2.0
23	2.0	795.0	1.0
24	3.0	44.0	1.0
25	3.0	106.0	1.0
26	3.0	108.0	1.0
27	3.0	707.0	2.0
28	3.0	209.0	2.0
29	3.0	250.0	1.0
30	3.0	294.0	2.0
31	3.0	432.0	2.0
32	3.0	439.0	1.0
33	3.0	568.0	3.0
34	3.0	660.0	1.0
35	3.0	720.0	1.0
36	4.0	20.0	1.0
37	4.0	22.0	2.0
38	4.0	23.0	2.0
39	4.0	25.0	1.0

3. The data elements in the ‘diagnose’ column of the diagnosis dataset contain a dirty bit. It is the male symbol (‘♂’) that appears as an unwanted bit in every instance of the diagnose column.

The data before correction:

	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...
30	32	Amebiasis♂ameba infection
31	33	Amphetamine abuse
32	34	Amphetamine overdose
33	36	Anal fissure♂ (tear)
34	37	Anaphylactoid reactions♂ (pseudo allergic reac...
35	38	Anaphylaxis♂(severe/life threatening allergic ...
36	39	Anemia♂ (low red blood cell count)
37	40	Ankle laceration♂ (cut in skin)
38	41	Ankle swelling
39	42	Ankylosing spondylitis♂ (severe spine arthritis)

```
def repair(i):
    l = i.split('\x0b')
    s = ""
    for j in l:
        s = s+' '+j
    return s
dia['diagnose'] = dia['diagnose'].apply(lambda x: repair(x))
```

The data after correction:

	did	diagnose
0	1	Abdominal aortic aneurysm (enlarged major blo...
1	2	Abdominal swelling
2	3	Abdominal trauma
3	4	Abrasions (scrapes)
4	5	ACE inhibitor induced cough blood pressure me...
5	6	acetaminophen overdose Adverse reaction to ac...
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 s...
13	14	Acute angle closure glaucoma increased inner ...
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 shou...
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 ...
26	28	Allergy
27	29	Confusion (altered mental status)
28	30	Altered mental status confusion
29	31	Altitude illness Illnesses due to high altitu...
30	32	Amebiasis ameba infection
31	33	Amphetamine abuse
32	34	Amphetamine overdose
33	36	Anal fissure (tear)
34	37	Anaphylactoid reactions (pseudo allergic rea...
35	38	Anaphylaxis (severe/life threatening allergic...
36	39	Anemia (low red blood cell count)
37	40	Ankle laceration (cut in skin)
38	41	Ankle swelling
39	42	Ankylosing spondylitis (severe spine arthritis)

4. The column names in the given datasets were mostly cryptic and meaningless. Hence we changed the names like ‘syd’, ‘did’, and ‘wei’ to ‘symptom id’, diagnosis id’, and ‘weight’ respectively.

The data before correction:

	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

```
df.rename(columns={'wei':'weight','syd':'symptom id','did':'diagnosis id'}\
          ,inplace=True)
sym.rename(columns={'syd':'symptom id'},inplace=True)
dia.rename(columns={'did':'diagnosis id'},inplace=True)
```

The data after correction:

	symptom id	diagnose id	weight
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

5. The elements of the ‘weight’ column was given in numbers but each number had its own meaning. Hence it was necessary to rename them according to the type of disease it implied.

The data before cleaning:

	symptom id	diagnose id	weight
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

```
def colu(g):
    if g==1.0:
        s='common'
        return s

    elif g==2.0:
        s='life-threatening'
        return s

    elif g==3.0:
        s='common-paediatrics'
        return s

df['weight']=df['weight'].apply(lambda x:colu(x))
```

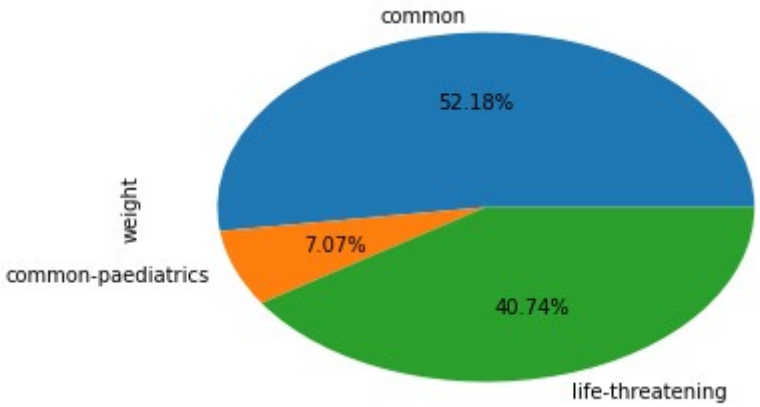
The data after cleaning:

	symptom id	diagnose id	weight
0	1	163	life-threatening
1	1	164	life-threatening
2	1	165	common
3	1	187	life-threatening
4	1	306	life-threatening
5	1	307	common
6	1	308	life-threatening
7	1	309	life-threatening
8	1	354	common
9	1	401	common

Analysing the data based on various parameters

1. Types of diseases (by weight):

```
grp = df.groupby('weight')
temp = grp['weight'].count()
temp.plot(kind = "pie",autopct="%0.2f%%")
```



Inference: Most of the symptom – disease combinations in the dataset are of 'common' type.

2. The most common symptom:

```
grp = df.groupby("symptom id")
top = grp['weight'].count().sort_values(ascending = False)
print(sym[sym['symptom id']==(top == top.max).argmax()])
```

```
In [21]: runfile('C:/Users/SOHAM/Desktop/Python Programs/Winter School Project/
2.py', wdir='C:/Users/SOHAM/Desktop/Python Programs/Winter School Project')
     symptom id     symptom
246         262  Chest pain
```

Inference: Chest pain is the most common symptom within the given data.

3. The disease with the maximum number of symptoms:

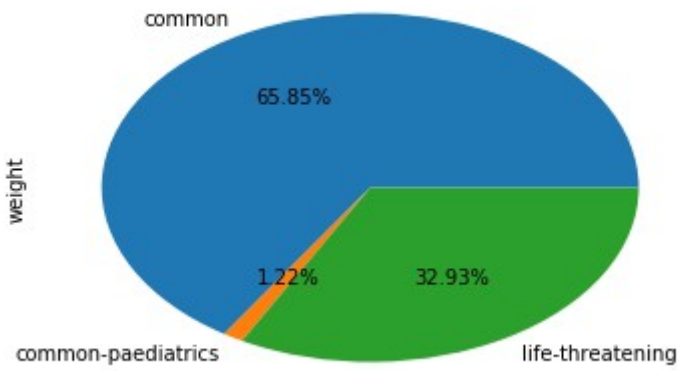
```
grp = df.groupby("diagnosis id")
top = grp['weight'].count().sort_values(ascending = False)
print(dia[dia['diagnosis id']==(top == top.max).argmax()])
```

```
In [25]: runfile('C:/Users/SOHAM/Desktop/Python Programs/Winter School Project/
2.py', wdir='C:/Users/SOHAM/Desktop/Python Programs/Winter School Project')
     diagnosis id                    diagnose
133           140   Cellulitis skin infection
```

Inference: Cellulitis skin infection has the maximum number of symptoms.

4. Chances of chest pain to be life threatening:

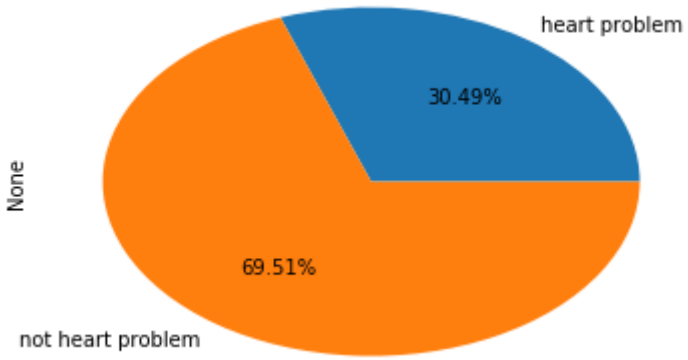
```
grp = df.groupby("symptom id")

toplist = grp['weight'].count().sort_values(ascending = False)
top = (toplist == toplist.max).argmax()
topsym = grp.get_group(top)
grp = topsym.groupby('weight')
toplist = grp['weight'].count()
toplist.plot(kind = 'pie',autopct = '%0.2f%%')
```



Inference: In most cases (approx. 66% times) chest pain is non life threatening.

5. Chances of chest pain to be related to some heart problems:

```
grp = df.groupby("symptom id")
toplist = grp['weight'].count().sort_values(ascending = False)
top = (toplist == toplist.max).argmax()
chestpain = grp.get_group(top)
tempdf = pd.DataFrame()
for i,row in chestpain.iterrows():
    tempdf = tempdf.append(dia[dia['diagnosis id']==row['diagnosis id']])
c = 0
nc = 0
for i,row in tempdf.iterrows():
    res1 = re.search('[Hh]eart',row['diagnose'])
    res2 = re.search('[Cc]ardiac',row['diagnose'])
    if res1 or res2:
        c +=1
    else:
        nc+=1
print(nc)
ls = pd.Series({'heart problem':c,'not heart problem':nc})
ls.plot(kind = 'pie',autopct = "%0.2f%%")
```



Inference: 70% of chest pain cases are not related to heart problems.

6. Symptoms of cellulitis skin infection (the disease with the most number of symptoms):

```
grp = df.groupby("diagnosis id")
toplist = grp['weight'].count().sort_values(ascending = False)
top = (toplist == toplist.max).argmax()
cellulitis = grp.get_group(top)
tempdf = pd.DataFrame()
for i,row in cellulitis.iterrows():
    tempdf = tempdf.append(sym[sym['symptom id']==row['symptom id']])
print(tempdf)
```

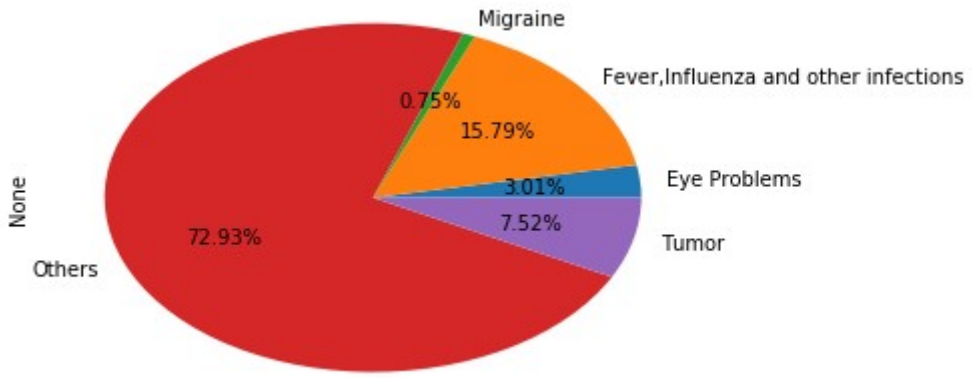
	symptom id	symptom
5	6	Arm ache or pain
10	11	Pain or soreness of breast
11	12	Calf pain
13	14	Chills
24	25	Elbow ache or pain
28	29	Fever
33	34	unnamed
34	35	Foot pain
38	39	Hand, finger ache or pain
44	45	Hip pain
54	55	Knee pain
56	57	Leg ache or pain
57	58	Swelling of both legs
67	68	Neck swelling
99	100	Toe pain
117	118	Wrist pain
80	81	Rash
146	152	Painful rash
178	191	Skin pain
179	192	Hot skin
180	193	Skin swelling
181	194	Lip swelling
183	196	Foot swelling
200	213	Arm swelling
201	214	Calf swelling
203	216	Ear swelling
204	217	Wrist swelling
210	223	unnamed
213	226	Hand redness
214	227	Foot redness
215	228	Arm redness
216	229	Leg redness
220	233	Upper leg pain
221	234	Armpit pain
238	251	Shin pain
41	42	Heel pain
247	263	Skin infection
261	287	Lump or mass of breast
0	1	Upper abdominal pain
6	7	Back ache or pain
22	23	Ear pressure
23	24	Pain in the ear
36	37	Groin pain
51	52	Skin itching
66	67	Neck ache or pain
225	238	Skin sores
246	262	Chest pain
248	264	Stomach and abdominal pain
256	275	Low back ache or pain
255	274	Discharge from ear
245	261	Elbow swelling
258	281	Penis inflammation or swelling
252	271	Pulling at ears
253	272	Skin bumps

Inference: The list given above depicts the symptoms of cellulitis skin infection.

7. Classification of head ache:

```
tempdf = pd.DataFrame()
for i,row in sym.iterrows():
    res = re.search('[Hh]eadache',row['symptom'])
    if res:
        tempdf = tempdf.append(row)
newdf = pd.DataFrame()
for i,row in tempdf.iterrows():
```

```
    newdf = newdf.append(df[df['symptom id']==row['symptom id']])
tempdf = pd.DataFrame()
for i,row in newdf.iterrows():
    tempdf = tempdf.append(dia[dia['diagnosis id']==row['diagnosis id']])
m,t,o,e,f = 0,0,0,0,0
for i,row in tempdf.iterrows():
    res1 = re.search('[Mm]igraine',row['diagnose'])
    res2 = re.search('[Tt]umor',row['diagnose'])
    res3 = re.search('[Ee]ye',row['diagnose'])
    res4 = re.search('[Ff]ever',row['diagnose'])
    res5 = re.search('[Ii]nfluenza',row['diagnose'])
    res6 = re.search('[Ii]nfection',row['diagnose'])
    if res1:
        m+=1
    elif res2:
        t+=1
    elif res3:
        e+=1
    elif res4 or res5 or res6:
        f+=1
    else:
        o+=1
ser = pd.Series({'Migraine': m,'Tumor': t,'Eye Problems':e,\
                 'Fever, Influenza and other infections':f, 'Others': o})
ser.plot(kind = 'pie',autopct = '%0.2f%%')
```



Inference: The pie chart given above shows the various causes of head ache.

8. Most problematic form of cancer (toughest to detect):

```
tempdf = pd.DataFrame()
for i,row in dia.iterrows():
    res = re.search('[Cc]ancer',row['diagnose'])
    if res:
        tempdf = tempdf.append(row)
newdf = pd.DataFrame()
for i,row in tempdf.iterrows():
    newdf = newdf.append(df[df['diagnosis id']==row['diagnosis id']])
```

```
grp = newdf.groupby('diagnosis id')
req = grp['diagnosis id'].count().sort_values()
result = pd.DataFrame()
for i in req.index:
    if(req[i]==req.min()):
        result = result.append(dia[dia['diagnosis id'] == i])
    else:
        break
print(result)
```

	diagnosis id	diagnose
750	802	Vaginal cancer tumor
503	543	Paget disease of the nipple rare breast cancer
1051	1265	Small bowel cancer small intestine
1022	1141	Small cell lung cancer
230	240	Endometrial cancer cancer of the lining of th...
466	504	Nasal cancer tumor

Inference: The types of cancer shown above are more difficult to detect because they have the lowest number of symptoms.

9. Silent Heart Diseases (with the lowest number of symptoms):

```
tempdf = pd.DataFrame()
for i,row in dia.iterrows():
    res1 = re.search('[Hh]eart',row['diagnose'])
    res2 = re.search('[Cc]ardi[(ac)o]',row['diagnose'])
    if res1 or res2:
        tempdf = tempdf.append(row)
newdf = pd.DataFrame()
for i,row in tempdf.iterrows():
    newdf = newdf.append(df[df['diagnosis id']==row['diagnosis id']])
grp = newdf.groupby('diagnosis id')
req = grp['diagnosis id'].count().sort_values()
result = pd.DataFrame()
for i in req.index:
    if(req[i]==req.min()):
        result = result.append(dia[dia['diagnosis id'] == i])
    else:
        break
print(result)
```

	diagnosis id	diagnose
904	982	Cardiac arrest heart stops
184	194	Coronary artery dissection heart artery tear

Inference: The above mentioned heart diseases have the lowest number of symptoms. Hence they are tougher to detect and silently affect a persons health.

10. Symptoms of appendicitis:

```
for i,row in dia.iterrows():
    res = re.search('[Aa]ppendicitis',row['diagnose'])
    if res:
        grp = df.groupby('diagnosis id')
        appendicitis = grp.get_group(row['diagnosis id'])
tempdf = pd.DataFrame()
for i,row in appendicitis.iterrows():
    tempdf = tempdf.append(sym[sym['symptom id']==row['symptom id']])
print(tempdf)
```

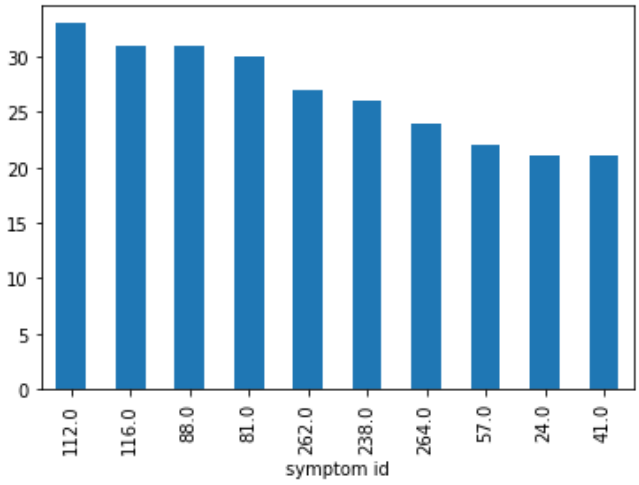
	symptom id	symptom
1	2	Lower abdominal pain
126	128	Inconsolable baby
248	264	Stomach and abdominal pain
0	1	Upper abdominal pain
6	7	Back ache or pain
13	14	Chills
15	16	Constipation
31	32	Flank pain
36	37	Groin pain
53	54	Kidney pain (Flank pain)
112	113	Vomiting

Inference: The list given above shows the symptoms of appendicitis.

11. Most common life threatening symptoms:

```
grp = df.groupby('weight')
tempdf = grp.get_group('life-threatening')
grp = tempdf.groupby('symptom id')
temp = grp['symptom id'].count().sort_values(ascending = False)
tempdf = pd.DataFrame()
for i in temp.index:
    tempdf = tempdf.append(sym[sym['symptom id']==i])
temp.head(10).plot(kind='bar')
print(tempdf.head(10))
```

	symptom id	symptom
111	112	Visual problems
115	116	Tired
87	88	Shortness of breath
80	81	Rash
246	262	Chest pain
225	238	Skin sores
248	264	Stomach and abdominal pain
56	57	Leg ache or pain
23	24	Pain in the ear
40	41	Headache

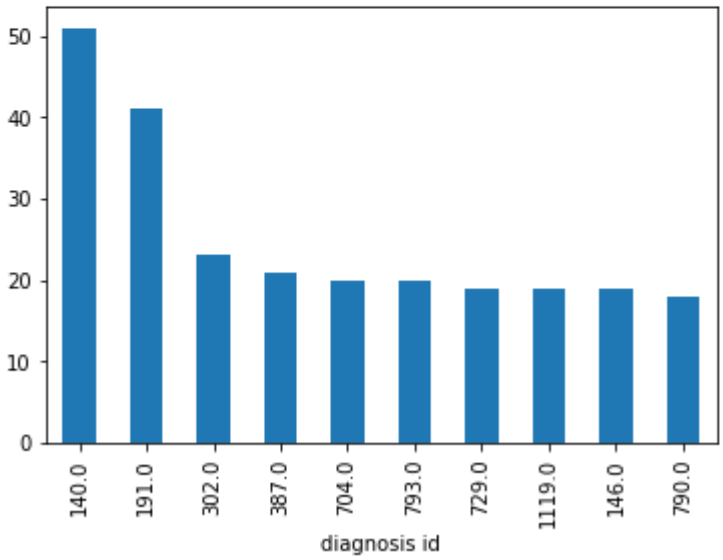


Inference: The list presents the 10 most common symptoms of life threatening diseases.

12. Life threatening diseases with the maximum number of symptoms:

```
grp = df.groupby('weight')
tempdf = grp.get_group('life-threatening')
grp = tempdf.groupby('diagnosis id')
temp = grp['diagnosis id'].count().sort_values(ascending = False)
tempdf = pd.DataFrame()
for i in temp.index:
    tempdf = tempdf.append(dia[dia['diagnosis id']==i])
temp.head(10).plot(kind='bar')
print(tempdf.head(10))
```

	diagnosis id	diagnose
133	140	Cellulitis skin infection
181	191	Contusion bruise, ecchymosis
285	302	Fracture broken bone
361	387	Influenza seasonal flu
658	704	Sinusitis sinus infection
741	793	Bladder infection cystitis, UTI, urinary trac...
681	729	Muscle strain pulled muscle
1010	1119	Sarcoma soft tissue cancer
138	146	Cerebral vascular accident stroke
738	790	Upper respiratory tract infection URI, common...

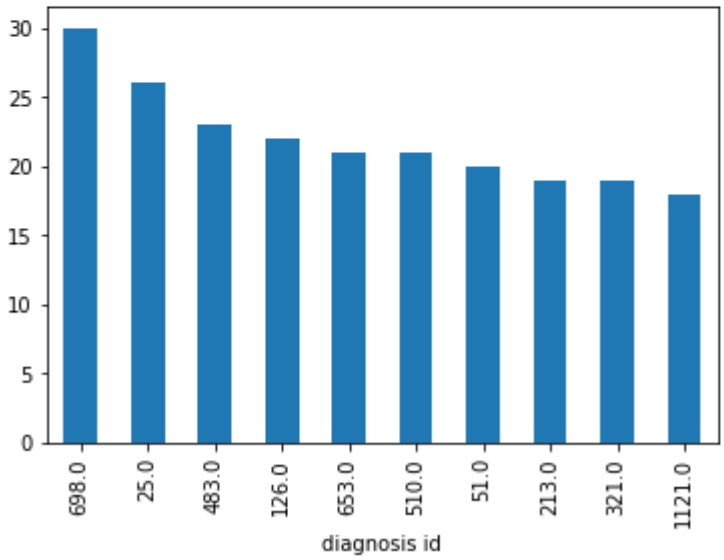


Inference: The list presents the top ten life threatening diseases with the maximum number of symptoms.

13. Common diseases with the maximum number of symptoms:

```
grp = df.groupby('weight')
tempdf = grp.get_group('common')
grp = tempdf.groupby('diagnosis id')
temp = grp['diagnosis id'].count().sort_values(ascending = False)
tempdf = pd.DataFrame()
for i in temp.index:
    tempdf = tempdf.append(dia[dia['diagnosis id']==i])
temp.head(10).plot(kind='bar')
print(tempdf.head(10))
```

	diagnosis id	diagnose
652	698	Shingles herpes zoster
23	25	Alcoholism
448	483	Multiple sclerosis MS
119	126	Cancer tumor
608	653	Renal failure, chronic ongoing kidney failure
472	510	Necrotizing fasciitis life-threatening infection
48	51	Anxiety disorder generalized anxiety disorder...
203	213	Diabetes high blood sugar
302	321	Gout uric acid crystals in the joint causing ...
1011	1121	Bone cancer bone tumor

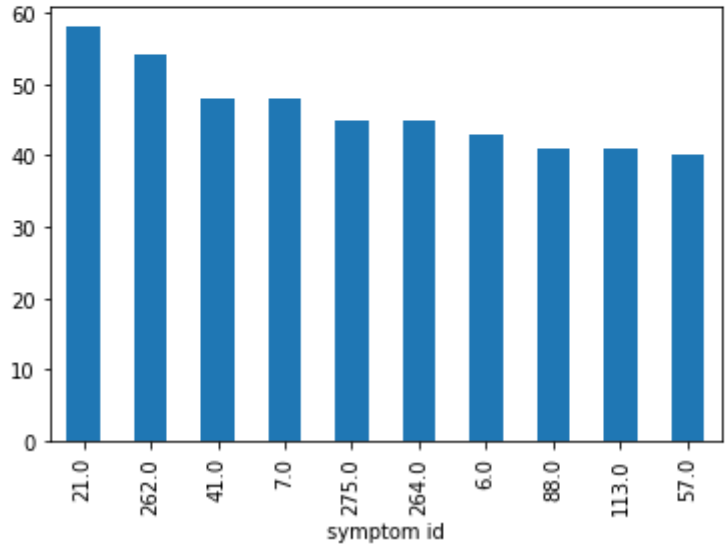


Inference: The list presents the top ten common diseases with the maximum number of symptoms.

14. Most common symptoms of non-lethal diseases:

```
grp = df.groupby('weight')
tempdf = grp.get_group('common')
grp = tempdf.groupby('symptom id')
temp = grp['symptom id'].count().sort_values(ascending = False)
tempdf = pd.DataFrame()
for i in temp.index:
    tempdf = tempdf.append(sym[sym['symptom id']==i])
temp.head(10).plot(kind='bar')
print(tempdf.head(10))
```

	symptom id	symptom
20	21	Dizziness
246	262	Chest pain
40	41	Headache
6	7	Back ache or pain
256	275	Low back ache or pain
248	264	Stomach and abdominal pain
5	6	Arm ache or pain
87	88	Shortness of breath
112	113	Vomiting
56	57	Leg ache or pain

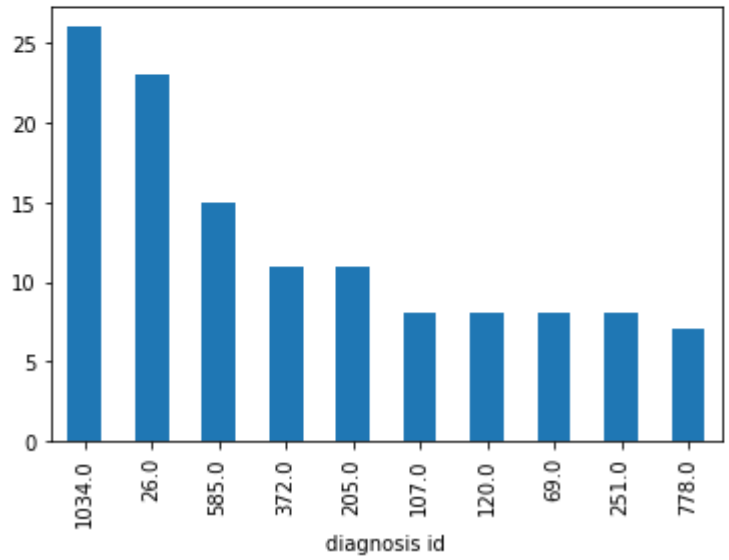


Inference: The list presents the 10 most common symptoms of common diseases.

15. Common-paediatric diseases with the maximum number of symptoms:

```
grp = df.groupby('weight')
tempdf = grp.get_group('common-paediatrics')
grp = tempdf.groupby('diagnosis id')
temp = grp['diagnosis id'].count().sort_values(ascending = False)
tempdf = pd.DataFrame()
for i in temp.index:
    tempdf = tempdf.append(dia[dia['diagnosis id']==i])
temp.head(10).plot(kind='bar')
print(tempdf.head(10))
```

	diagnosis id	diagnose
946	1034	Medication reaction
24	26	Allergic reaction
543	585	Pneumonia lung infection
348	372	Hypoglycemia low blood sugar
195	205	Dehydration
100	107	Brain tumor cancer of the brain
113	120	Burns
65	69	Atypical pneumonia lung infection
240	251	Epidural hematoma bleeding around brain or spine
728	778	Traumatic nerve injury

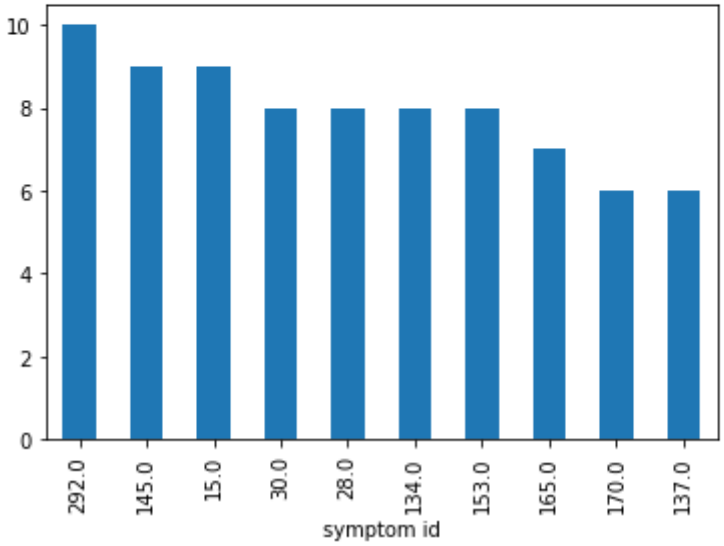


Inference: The list presents the top ten common paediatric diseases with the maximum number of symptoms.

16. Most common symptoms of paediatric diseases:

```
grp = df.groupby('weight')
tempdf = grp.get_group('common-paediatrics')
grp = tempdf.groupby('symptom id')
temp = grp['symptom id'].count().sort_values(ascending = False)
tempdf = pd.DataFrame()
for i in temp.index:
    tempdf = tempdf.append(sym[sym['symptom id']==i])
temp.head(10).plot(kind='bar')
print(tempdf.head(10))
```

	symptom id	symptom
264	292	Confusion
139	145	Headache after trauma
14	15	Change in behavior
29	30	Fever in the returning traveler
27	28	Fainting
132	134	unnamed
147	153	Ingestion
158	165	Bleeding in brain
159	170	Cyanosis (Blue skin coloration)
135	137	unnamed



Inference: The list presents the 10 most common symptoms of common-paediatric diseases.

17. Diagnosis of drug abuse:

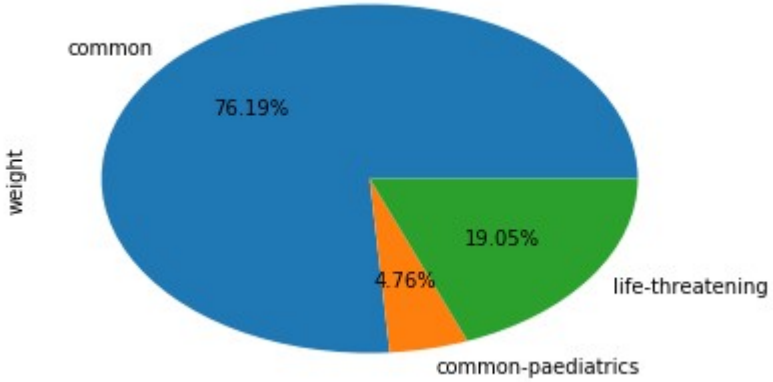
```
temp = pd.DataFrame()
for i,row in sym.iterrows():
    res = re.search('[Dd]rug',row['symptom'])
    if res:
        temp = temp.append(row)
tempdf = pd.DataFrame()
for i in temp['symptom id']:
    tempdf = tempdf.append(df[df['symptom id']==i])
temp = pd.DataFrame()
for i in tempdf['diagnosis id']:
    temp = temp.append(dia[dia['diagnosis id']==i])
print(temp)
```

	diagnosis id	diagnose
9	10	Acid LSD abuse
86	90	Benzodiazepine Valium abuse
165	175	Cocaine abuse
225	235	Ecstasy MDMA abuse
363	389	Inhalants abuse huffing
422	452	Marijuana use
464	502	Narcotic morphine, heroin abuse
558	601	Prescription drug abuse
18	19	Adjustment disorder (poor adjustment to life...
23	25	Alcoholism
31	33	Amphetamine abuse
48	51	Anxiety disorder generalized anxiety disorder...
88	93	Bipolar disorder manic depressive disorder
202	212	Depression excessive sadness
215	225	Drug overuse Prescription Drug Overuse
217	227	Drug reaction
414	444	Magic mushroom ingestion psilocybin
631	676	Schizoaffective disorder features of schizoph...
684	732	Stress
814	878	Attention deficit hyperactivity disorder ADHD
946	1034	Medication reaction

Inference: The list presents the diagnoses of drug abuse.

18. Types of diseases caused by drug abuse:

```
temp = pd.DataFrame()
for i,row in sym.iterrows():
    res = re.search('[Dd]rug',row['symptom'])
    if res:
        temp = temp.append(row)
tempdf = pd.DataFrame()
for i in temp['symptom id']:
    tempdf = tempdf.append(df[df['symptom id']==i])
print(tempdf)
grp = tempdf.groupby('weight')
count = grp['weight'].count()
count.plot(kind = 'pie',autopct = '%0.2f%%')
```



Inference: 19.05% of drug abuse diagnoses are life threatening.

19. Life threatening conditions caused by drug abuse:

```
temp = pd.DataFrame()
for i,row in sym.iterrows():
    res = re.search('[Dd]rug',row['symptom'])
    if res:
        temp = temp.append(row)
tempdf = pd.DataFrame()
for i in temp['symptom id']:
    tempdf = tempdf.append(df[df['symptom id']==i])
grp = tempdf.groupby('weight')
lt = grp.get_group('life-threatening')
temp = pd.DataFrame()
for i in lt['diagnosis id']:
    temp = temp.append(dia[dia['diagnosis id']==i])
print(temp)
```

	diagnosis id	diagnose
9	10	Acid LSD abuse
225	235	Ecstasy MDMA abuse
363	389	Inhalants abuse huffing
422	452	Marijuana use

Inference: The list shows the life threatening conditions caused by drug abuse.

20. Effects of alcohol abuse:

```
temp = pd.DataFrame()
for i,row in sym.iterrows():
    res = re.search('[Aa]lcohol',row['symptom'])
    if res:
```

```
        temp = temp.append(row)
tempdf = pd.DataFrame()
for i in temp['symptom id']:
    tempdf = tempdf.append(df[df['symptom id']==i])
temp = pd.DataFrame()
for i in tempdf['diagnosis id']:
    temp = temp.append(dia[dia['diagnosis id']==i])
print(temp)
```

	diagnosis id	diagnose
19	20	Alcohol ethanol intoxication
21	22	Alcohol ethanol poisoning (overdose)
22	23	Alcohol withdrawal syndrome (mild)
23	25	Alcoholism
196	206	Delirium tremens DTs, severe alcohol withdr...
162	172	Cirrhosis liver failure and scarring
202	212	Depression excessive sadness
289	306	Gastric ulcer stomach ulcer
415	445	Major depressive disorder severe depression
422	452	Marijuana use
464	502	Narcotic morphine, heroin abuse
506	546	Pancreatitis pancreas inflammation
532	574	Phobias irrational fear
558	601	Prescription drug abuse
632	677	Schizophrenia chronic impaired reality percep...

Inference: The list shows the effect of alcohol abuse.

Future Scope of Improvements

The analysis of this disease data can further be improved by implementing machine learning models and using predictive analytics. The above mentioned concepts can be used to predict symptoms or diagnosis for given cases.