111學年第1學期

輔仁大學

醫學資訊與創新應用學士學位學程

專題報告

**程式碼**

題目：

利用門診就醫資料分析疾病的共病性

409570236 吳思嫻

409570224 林虹樺

409570298 王殿馨

409570420 程渝剴

指導老師：曾景平

111年12月

## **一、程式原始碼**

(一)資料處理程式碼

匯入資料處理會使用到的套件。

import matplotlib.pyplot as plt

import matplotlib.style

import pandas as pd

import numpy as np

import seaborn as sns

!pip install -U pandasql

matplotlib.style.use('bmh')

讀取原始資料，觀察資料狀況。

Raw\_Data = pd.read\_csv("/content/drive/MyDrive/專題(一) 共病性查詢系統/2014\_basic.txt",encoding='big5')

Raw\_Data

原始資料Raw\_Data範例如下：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| index | ACCOUNTNO | INDATE | CHARTID | AGE | SEX | COSTNAME | ICD91 | ICD9NAME1 |
| 0 | OZ10\*\*\*\*\*\*53 | 1030121 | W22\*\*\*\*\*57 | 50 | F | 耳鼻喉科 | 683 | 急性淋巴腺炎 |

資料處理過程中不需要使用到就醫序號"就醫序號"、"科別"和"診斷名"，因此將這些欄位刪除。

#刪除"就醫序號ACCOUNTNO""科別COSTNAME"和"診斷名ICD9CODENAME"資料

Data\_drop1 = Raw\_Data.drop(columns=(['ACCOUNTNO','COSTNAME','ICD9CODENAME1','ICD9CODENAME2','ICD9CODENAME3','ICD9CODENAME4','ICD9CODENAME5','ICD9CODENAME6']))

Data\_drop1

Data\_drop1範例如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| index | INDATE | CHARTID | AGE | SEX | ICD91 | ICD92 |
| 0 | 1030121 | W22\*\*\*\*\*57 | 50 | F | 683 | NaN |

觀察資料後可以發現資料中存在一些無意義的資訊，將這些資訊刪除後才能進行資料處理。

#顯示無意義資料

Meaningless\_Data = Data\_drop1[Data\_drop1.AGE == "AGE"]

Meaningless\_Data

#刪除無意義資料

Data\_drop2 = Data\_drop1.drop(Data\_drop1[(Data\_drop1.AGE == "AGE")].index)

Data\_drop2 = Data\_drop2.reset\_index(drop = True) #重設index

Meaningless\_Data = Data\_drop2[Data\_drop2.AGE == "AGE"]

Meaningless\_Data

#顯示刪除無意義資料結果

Data\_drop2

不同時間可能會使用不同版本的疾病分類系統，較舊的資料會使用ICD9，較新的資料可能會使用ICD10。將民國年轉為西元年後更方便溝通和資料整理。目前使用的資料集都是用ICD9，不會有轉換問題，此處程式碼為處理方法示意。

#民國年轉西元年

INDATE1 = Data\_drop2['INDATE'].copy()

INDATE1

#將每筆資料的前四碼加上1911，變成西元年輸出(結果資料類型為int)

INDATE2 = INDATE1.copy()

INDATE2 = pd.to\_numeric(INDATE2)

INDATE2 = INDATE2 + 19110000

INDATE2 = pd.to\_numeric(INDATE2)

INDATE2

#插入"-"以符合MySQL的DATE格式

INDATE3 = INDATE2.copy()

INDATE3 = INDATE3.astype("str")

INDATE3 = INDATE3.to\_frame()

INDATE3['INDATE'] = INDATE3['INDATE'].str[:4] + '-' + INDATE3['INDATE'].str[4:6] + '-' + INDATE3['INDATE'].str[6:]

INDATE3

INDATE = INDATE3.copy()

INDATE

#將民國年日期轉爲符合DATE格式的西元年日期

Data\_convert1 = Data\_drop2.copy()

Data\_convert1['INDATE'] = INDATE

Data\_convert1

Data\_convert1範例如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| index | INDATE | CHARTID | AGE | SEX | ICD91 | ICD92 |
| 0 | 2014-12-01 | W22\*\*\*\*\*57 | 50 | F | 683 | NaN |

將年齡資料形態轉為整數

#將AGE的資料形態轉爲int

Data\_convert2 = Data\_convert1.copy()

Data\_convert2['AGE'] = pd.to\_numeric(Data\_convert2['AGE'])

Data\_convert2.dtypes

原始資料是將一次門診的所有診斷記錄在同一筆資料中，而共病性分析需要將一個患者得到的所有疾病診斷兩兩組合，因此需要將單次門診的所有診斷分割成對應相同患者的多筆資料。

#分割ICD碼資料

ICD\_1 = Data\_convert2[['INDATE','CHARTID','AGE','SEX','ICD9CODE1']]

ICD\_2 = Data\_convert2[['INDATE','CHARTID','AGE','SEX','ICD9CODE2']]

ICD\_3 = Data\_convert2[['INDATE','CHARTID','AGE','SEX','ICD9CODE3']]

ICD\_4 = Data\_convert2[['INDATE','CHARTID','AGE','SEX','ICD9CODE4']]

ICD\_5 = Data\_convert2[['INDATE','CHARTID','AGE','SEX','ICD9CODE5']]

ICD\_6 = Data\_convert2[['INDATE','CHARTID','AGE','SEX','ICD9CODE6']]

ICD\_1.columns = ['INDATE','CHARTID','AGE','SEX','ICD']

ICD\_2.columns = ['INDATE','CHARTID','AGE','SEX','ICD']

ICD\_3.columns = ['INDATE','CHARTID','AGE','SEX','ICD']

ICD\_4.columns = ['INDATE','CHARTID','AGE','SEX','ICD']

ICD\_5.columns = ['INDATE','CHARTID','AGE','SEX','ICD']

ICD\_6.columns = ['INDATE','CHARTID','AGE','SEX','ICD']

ICD\_split1 = pd.concat([ICD\_1,ICD\_2,ICD\_3,ICD\_4,ICD\_5,ICD\_6],ignore\_index = True)

ICD\_split1

ICD\_split1\_test範例如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| index | INDATE | CHARTID | AGE | SEX | ICD |
| 0 | 2014-12-01 | W22\*\*\*\*\*57 | 50 | F | 683 |
| 1 | 2014-12-01 | W22\*\*\*\*\*57 | 50 | F | NaN |

門診資料中不滿六個ICD9診斷碼的部分會記錄為NaN，這些空資料對資料分析沒有用處，因此刪除。本系統分析僅考慮ICD9主碼，也就是診斷碼的前三碼。

#刪除空的ICD資料

ICD\_split2 = ICD\_split1.dropna()

ICD\_split2

#顯示某病患刪除空ICD資料結果

ICD\_split2\_test = ICD\_split2[ICD\_split2.CHARTID == "W290019237"]

ICD\_split2\_test

#取ICD主碼(前3碼)

ICD\_split3 = ICD\_split2.copy()

ICD\_split3['ICD']=[x[:3] for x in ICD\_split3['ICD']]

ICD\_split3

#顯示某病患取ICD主碼結果

ICD\_split3\_test = ICD\_split3[ICD\_split3.CHARTID == "W290019237"]

ICD\_split3\_test

第17、18和19類(17:癥候及不明情況 18:受傷及中毒 19:外傷)的診斷和疾病共病性研究不相關或相關性不明確，因此這些資料也不納入分析資料集。診斷碼以E和V開頭第19類先進行刪除，然後將診斷碼的資料類型轉為整數後刪除780~999的第17和18類

#刪除第19類ICD(E、V)

ICD\_clear1 = ICD\_split3.copy()

ICD\_clear1 = ICD\_clear1[~ICD\_clear1.ICD.str.contains("E")]

ICD\_clear1 = ICD\_clear1[~ICD\_clear1.ICD.str.contains("V")]

ICD\_clear1

ICD\_clear1['ICD'] = pd.to\_numeric(ICD\_clear1['ICD'])

ICD\_clear1.dtypes

#刪除第17、18類ICD(780 ~ 999)

ICD\_clear2 = ICD\_clear1.copy()

ICD\_clear2 = ICD\_clear2.drop(ICD\_clear2[(ICD\_clear2.ICD >= 780)].index)

ICD\_clear2 = ICD\_clear2.reset\_index(drop = True)

ICD\_clear2

ICD\_clear2\_test = ICD\_clear2[ICD\_clear2.CHARTID == "W225384583"]

ICD\_clear2\_test

#將ICD轉回String並補0以符合ICD格式

ICD\_clear3 = ICD\_clear2.copy()

ICD\_clear3['ICD'] = ICD\_clear3['ICD'].astype(str)

ICD\_clear3['ICD'] = ICD\_clear3['ICD'].str.zfill(3)

ICD\_clear3\_test = ICD\_clear3[ICD\_clear3.CHARTID == "W225384583"]

ICD\_clear3\_test

接下來會對病患資料以年齡進行分群(全年齡:0~100、兒童:0~12、青少年:13~24、壯年:25~44、中年:45~64、老年:65~100)，大於100歲的資料中有許多錯誤或難以解釋的資料(例如同一病人出現兩筆性別不同的資料)，且大於100歲的資料佔比極低(佔總資料集0.032%)，因此不考慮年齡100歲以上的診斷。

#年齡錯誤：年齡大於100

Data\_wrong\_Age = ICD\_clear3[(ICD\_clear3.AGE > 100)]

Data\_wrong\_Age

#刪除年齡大於100的資料

ICD\_clear4 = ICD\_clear3.copy()

ICD\_clear4 = ICD\_clear4.drop(ICD\_clear4[(ICD\_clear4.AGE > 100)].index)

ICD\_clear4

#資料分群：年齡

def Age\_Group(age):

if 0 <= age <= 12:

return "Children"

if 12 < age <= 24:

return "Teenager"

elif 24 < age <= 44:

return "Youth"

elif 44 < age <= 64:

return "Middle"

elif age > 64:

return "Elder"

ICD\_split4 = ICD\_clear4.copy()

ICD\_split4['AGE\_GROUP'] = ICD\_split4['AGE'].apply(Age\_Group)

ICD\_split4

ICD\_split4範例如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| index | INDATE | CHARTID | AGE | SEX | ICD91 | AGE\_GROUP |
| 0 | 2014-12-01 | W22\*\*\*\*\*57 | 50 | F | 683 | Middle |

接下來會對各年齡層進行性別分群(不分性別、男性和女性)，這邊以全年齡和兒童資料集為例。

#全年齡性別分群

ICD\_all1 = ICD\_split4.copy()

ICD\_all1

#刪除重複資料(CHARTID相同且ICD相同)

ICD\_all2 = ICD\_all1.copy()

ICD\_all2 = ICD\_all2.drop\_duplicates(subset=['CHARTID','ICD'], keep='first', inplace=False, ignore\_index=True)

ICD\_all2

#不分性別全年齡資料集

Data\_All = ICD\_all2.copy()

Data\_All

#男性全年齡資料集

Data\_All\_Male = Data\_All[(Data\_All.SEX == "M")]

Data\_All\_Male

#女性全年齡資料集

Data\_All\_Female = Data\_All[(Data\_All.SEX == "F")]

Data\_All\_Female

#取出年齡層為兒童的資料

ICD\_Children1 = ICD\_split4[(ICD\_split4['AGE\_GROUP'] == "Children")]

ICD\_Children1

#刪除重複資料(CHARTID相同且ICD相同)

ICD\_Children2 = ICD\_Children1.copy()

ICD\_Children2 = ICD\_Children2.drop\_duplicates(subset=['CHARTID','ICD'], keep='first', inplace=False, ignore\_index=True)

ICD\_Children2

#不分性別兒童資料集

Data\_Children = ICD\_Children2.copy()

Data\_Children

#男性兒童資料集

Data\_Children\_Male = Data\_Children[(Data\_Children.SEX == "M")]

Data\_Children\_Male

#女性兒童資料集

Data\_Children\_Female = Data\_Children[(Data\_Children.SEX == "F")]

Data\_Children\_Female

程式到此已經完成資料集分群和資料清理，接下來要對資料進行分析。

Heat Map：將單一病人獲得的所有疾病診斷兩兩組合，再統計各種疾病組合的人數，統計完成后將這些人數放入一個橫軸與縱軸都是ICD9主碼的矩陣中，最後對矩陣進行條件上色取得Heat Map，此處以全年齡不分性別資料群為例

#Heat Map矩陣

ICD\_all = Data\_all.copy()

ICD\_all = ICD\_all.drop(columns=(['INDATE','AGE','SEX','AGE\_GROUP']))

ICD\_all.sort\_values(by=['CHARTID', 'ICD'])

ICD\_all

!pip install -U pandasql

mysql = lambda q: sqldf(q, globals())

df1 = ICD\_all.copy()

df1['ICD'] = pd.to\_numeric(df1['ICD'])

#宣告780 x 780二維陣列並給予初始值0

column, row = 780, 780

A = [[0]\*row for \_ in range(column)]

#儲存每個病人的ICD

ticd=[0] \* 100

m = 0

ctr = 0

tflag = df1.loc[m,'CHARTID'] #把第一筆病歷號放入 tflag

lmax = df1.shape[0] # dataframe 的行數

#print("Chart no : " , tflag,"==",lmax)

while m < lmax:

if tflag != df1.loc[m,'CHARTID'] :

print("Chart no : " , tflag, " ctr :",ctr )

tflag = df1.loc[m,'CHARTID']

for i in range(0,ctr):

for j in range(i,ctr):

A[ticd[i]][ticd[j]] = A[ticd[i]][ticd[j]]+1

print(ticd[i],"-",ticd[j],"----",A[ticd[i]][ticd[j]])

ctr = 0

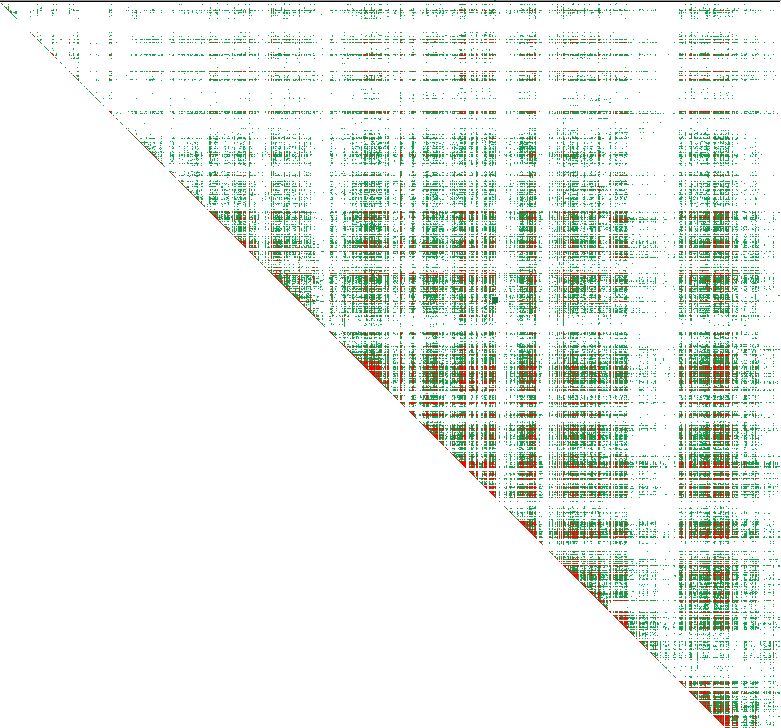
ticd[ctr] = df1.loc[m,'ICD']

#print(m,"=",tflag,"===",ticd[ctr])

ctr = ctr + 1

m = m + 1

矩陣經過上色處理後可以獲得Heat Map如下圖：



疾病人數直方圖和Bubble Map：在處理繪製Heat Map時計算出各疾病組合的人數，矩陣對角線上(ICD1等於ICD2)的數據即表示該資料集中獲得各疾病的人數，以此繪製出疾病人數直方圖。利用這些數據可以看出哪些各年齡或性別人群中，最可能患有哪些疾病，由此可以看出一般情況下一個病人在各年齡患病的趨勢。Bubble Map則可以看出各資料集中的診斷集中在哪個方面(例如兒童的診斷較多集中在呼吸道疾病)。

# 提取Bubble Map所需資料

n = 0

df = pd.DataFrame(columns=['ICD1','ICD2','qty'])

for i in range(780):

for j in range (780):

if A[i][j] != 0:

n1 = i #icd1

n2 = j #icd2

n3 = A[i][j] #qty

df.loc[n] = [n1,n2,n3]

n = n + 1

# 繪製Bubble Map

import pandas as pd

from pandasql import sqldf

mysql = lambda q: sqldf(q, globals())

#df1 = mysql("SELECT n1,n2,n3,round(log10(n3),0) as n4 FROM df where n1 = n2 and n3 > 10000")

df1 = mysql("SELECT ICD1,ICD2,qty FROM df where ICD1 = ICD2 and qty > 1000")

fig,ax=plt.subplots(figsize=(10,10))

ax.scatter(df1['ICD1'], df1['qty'], s=df1["qty"]\*1.2,c=df1["qty"],cmap='Set1',vmin=10,vmax=10000,alpha=0.3)

ax.set(xlim=(0,780),ylim=(200,6000),\

xlabel="ICD-9 code",\

ylabel="Diease Diagnosis",\

title="Patient Diagnosis Disbuition")

for i in range(df1.shape[0]):

ax.text(df1.at[i,'ICD1']-5, df1.at[i,'qty']-5,s=str(df1.at[i,'ICD1']),fontsize=10)

# show the graph

plt.show()

