**D6**

app\_train[numeric\_columns].columns[app\_train[numeric\_columns].apply(lambda x:len(x.unique())!=2 )]

前面藍色的部分您一定瞭解，就是列出所有篩選過(numeric\_columns)的columns

後面紅色的部分您要先瞭解三個部分

1. 瞭解 Dataframe裡的apply函數:

    apply函數在Dataframe裡非常好用，就是指 把所有的Series都套用這個公式上

    ex: app\_train[numeric\_columns].apply(lambda x:len(x.unique())!=2 )

           把橘色的公式套用在app\_train[numeric\_columns]裡

2. 瞭解 lambda函數:

    lambda其實就是簡易版的def fun(), 他適合用在一次性或是簡短的功能上

    lambda x:len(x.unique())!=2

    先定義一個變數x ，x 會等於len(x.unique()) != 2 ,也就是x唯一值的數量不等於二

    您可以試看看這兩段程式碼，這樣會更清楚我在講什麼

    app\_train['NAME\_CONTRACT\_TYPE'].value\_counts()      
   app\_train['NAME\_CONTRACT\_TYPE'].unique()

3. 瞭解unique()函數:

    unique英文意思就是唯一的，他就是用來刪除掉重複地值，只顯示不重複的值

    有點像是Set(集合) 一樣

dtypes.isin 當中要放的是是 dtype 的型態，而不是字串，所以必須這麼寫會比較好：

```

app\_train.dtypes.isin([np.dtype('int64'), np.dtype('float64')])

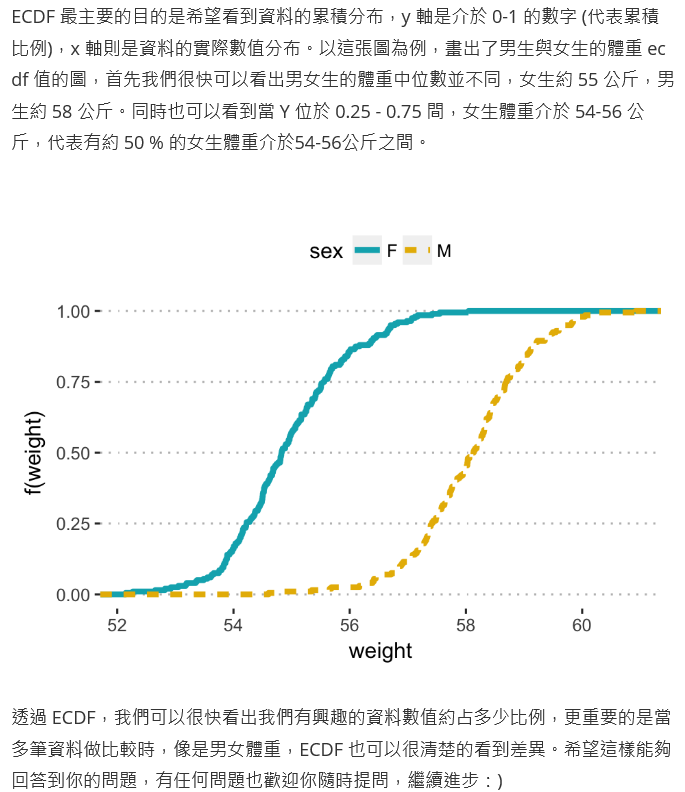
```

也可以改用對 df 直接做 select\_dtypes：

```

app\_train.select\_dtypes(include=['float64', 'int64'])

```



**D7**

quantile\_all = [np.percentile(app\_train[~app\_train['AMT\_ANNUITY'].isnull()]['AMT\_ANNUITY'], q = i) for i in q\_all]

1."~"代表不是的意思，也就是邏輯裡的NOT。

所以app\_train["AMT\_ANNUITY"].isnull()是指說"AMT\_ANNUITY"這個column裡面空的值，而前面多一個"~"就代表不是空的值，也就是我們說的一般值啦!!

2.建議您可以練習拆解看看

step1. app\_train[~app\_train['AMT\_ANNUITY'].isnull()]

step2. app\_train[~app\_train['AMT\_ANNUITY'].isnull()]['AMT\_ANNUITY']

step3. 了解 np.percentile 裡的參數代表什麼意思

step4. 了解 List comprehension

ex: [ 2 \* i for i in range(10) ]

最後就會解出您第二個問題啦!!

這種複雜的寫法，建議從內層慢慢往外看，在看的同時可以搭配印出確認。

這邊補充一下中間這一段： app\_train[~app\_train['AMT\_ANNUITY'].isnull()]，你會發現 ~app\_train['AMT\_ANNUITY'] 會回傳一個 Boolean Series 如下：

```

0 False

1 False

2 True

...

Name: numeric, dtype: bool

```

當我們把上述結果放在 DF 的 index 中，稱為 Boolean filtering/indexing，會將等於 True 的資料留下來。畫句話就是用 True/False 來決定每一筆資料要不要篩選。