# **Exam**

請使用R語言或是Python作答,完成後將答案截圖以及程式碼回傳給人資。

!作答時間為1小時,請優先完成O1再進行O2。

#### Q1

檔案 Data.xlsx 中有兩個sheets,

detail分頁中記錄各個帳號(login)在3/1-3/10的業績,以下為欄位說明:

• group:帳號組別,若帶有Test則為測試組別,不計算該帳號任何業績,請剃除。

• pnl: Profit and Loss, 損益

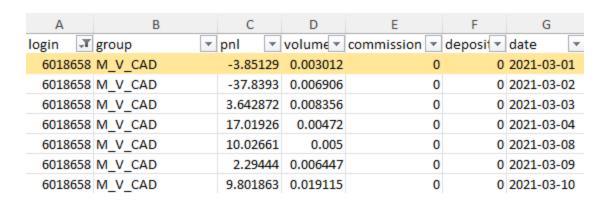
volume:交易量

• commission:手續費

• deposit:入款金額

sales info分頁中記錄各個帳號(login)每天的對應銷售(sales)關係。

舉例來說,帳號6018658在3/1的pnl為-3.85129,



而他的3/1對應銷售為Chris Lizhuang。

Exam 1



其中,銷售的bonus為計數單位,若帳號當日deposit大於3000,則Bonus計+1, 即當日該銷售下有多少帳號入款金額大於3000。

請推導出3/1-3/5各個銷售的業績,並依照pnl欄位進行降序排名。

### ! 輸出格式須包含以下欄位



### Q2

依照給定的DataFrame,寫出一函式能夠尋找出sub最上級的parent與 level,

## 若sub為最上級則返回[本身,0]。

以下圖舉例來說, sub 50 上級為 54、sub 55本身為最上級 (因Parent為0)。

sub	parent
50	54
51	55
52	54
53	56
54	51
55	0
56	0

Exam 2

```
import pandas as pd
from typing import List

# pd.DataFrame
sample_df = pd.DataFrame({
    "sub":[50,51,52,53,54,55,56],
    "parent":[54,55,54,56,51,0,0]
})

def find_master(df: pd.DataFrame, sub: int, level: int) -> List:
    """Return [master, level]. If the sub is the highest, return [sub, 0]"""
```

```
sample_df <- data.frame(
    sub = c(50,51,52,53,54,55,56),
    parent = c(54,55,54,56,51,0,0)
)

find_master <- function(df, sub, level){
    # Return c(master, level). If the sub is the highest, return c(sub, 0)
}</pre>
```

Exam 3