

REPORT

The questions of the project are the following:

- What is the balance per group?
- What is the balance per country?
- What is the total profit per client?
- Which date the company had the biggest profit ?
- Which client had the biggest losses?
- Which asset was traded the most?
- Which was the most losing asset?

The project includes two datasets and the exchange rates of 6 currencies to euro. The first dataset contains the client balances and the second the client trades. I imported the two datasets in Python with the Pandas library, then I cleaned the data and finally I visualized the data with the Matplotlib library. I estimated the balance per group and per country, the total profit per client, the date the company had the biggest profit, the client that had the biggest losses and the asset that was traded the most. I answered the first two questions using Excel and the rest with Python. Also, I imported the datasets in SQL and did some data exploration.

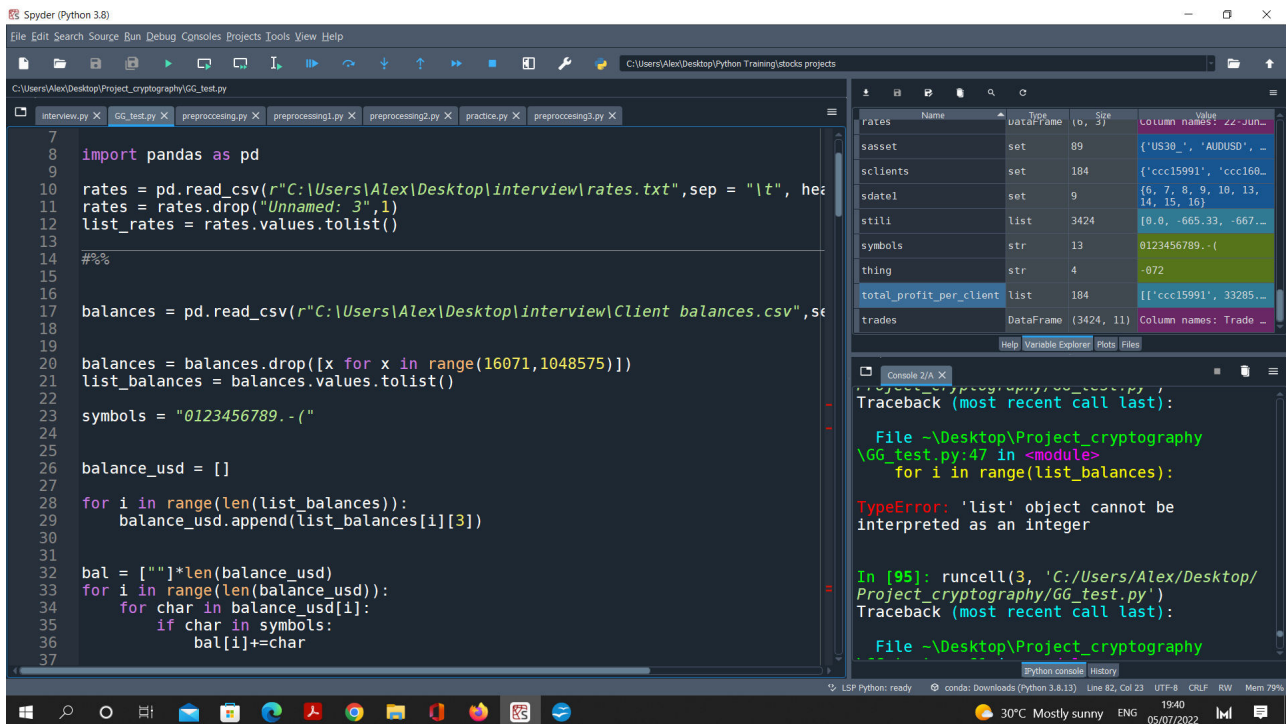
The balance per group is:

Group	Balance (euro)
Gold	47195686,22
Platinum	46024933,81
Regular	46482527,32
VIP	49262740,16

The balance per country is:

Country	Balance (euro)
Canada	20489973,07
Cyprus	22506683,85
Czech Republic	21977089,38
France	21798233,51
Greece	24681544,14
Hungary	24992509,32
Italy	28216183,03
UK	24303662,22

The following is an example of my working environment with some code:



The screenshot shows the Spyder Python IDE interface. The main editor displays a Python script with the following code:

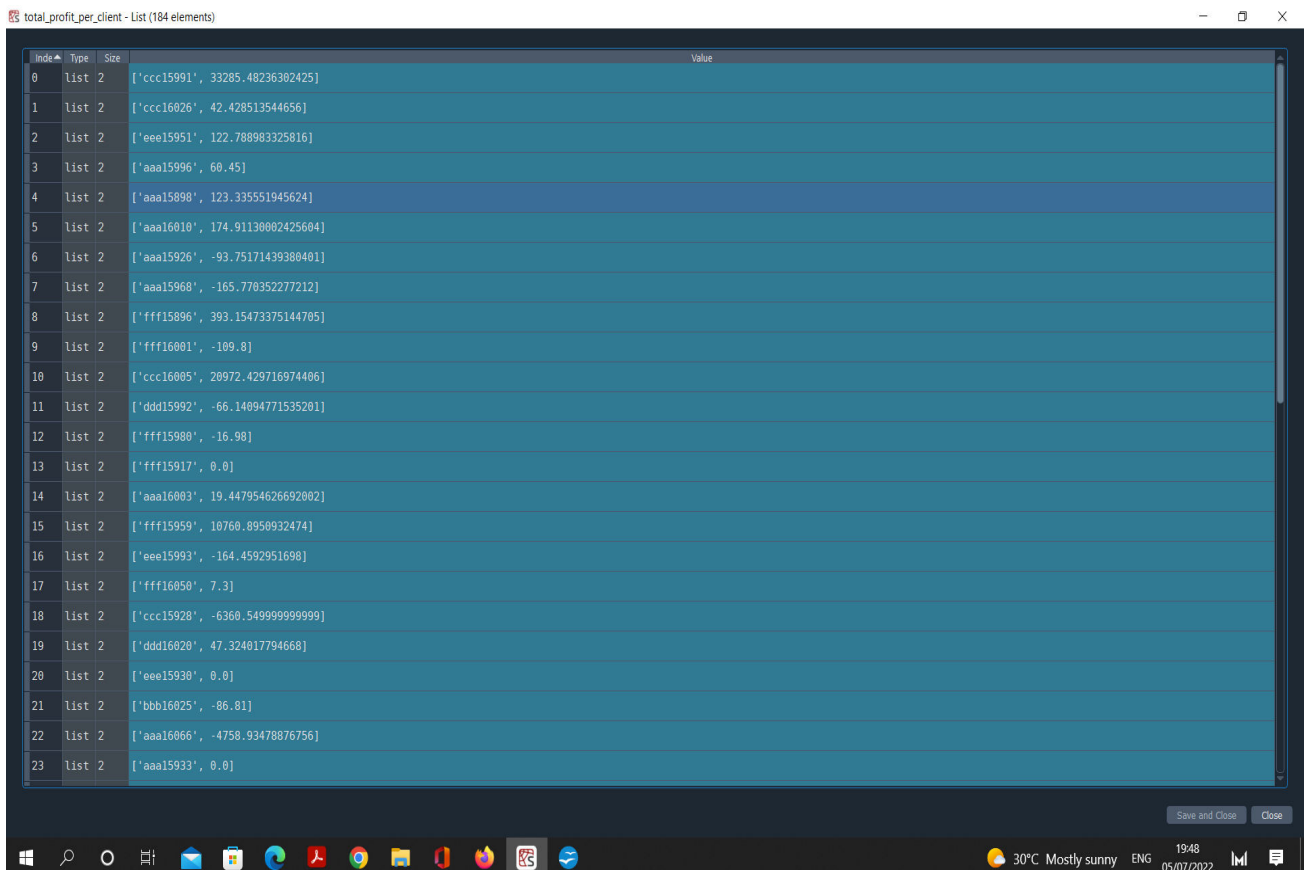
```
7
8 import pandas as pd
9
10 rates = pd.read_csv(r"C:\Users\Alex\Desktop\interview\rates.txt", sep = "\t", header=0)
11 rates = rates.drop("Unnamed: 3", 1)
12 list_rates = rates.values.tolist()
13
14 #%%
15
16 balances = pd.read_csv(r"C:\Users\Alex\Desktop\interview\client_balances.csv", sep = "\t", header=0)
17 balances = balances.drop([x for x in range(16071, 1048575)])
18 list_balances = balances.values.tolist()
19
20 symbols = "0123456789.-("
21
22 balance_usd = []
23
24 for i in range(len(list_balances)):
25     balance_usd.append(list_balances[i][3])
26
27 bal = [""] * len(balance_usd)
28 for i in range(len(balance_usd)):
29     for char in balance_usd[i]:
30         if char in symbols:
31             bal[i] += char
32
33
```

On the right side, the Variable Explorer shows a table of variables:

Name	Type	Id	Size	Column names
rates	DataFrame	10	3	
sasset	set	89		['US30', 'AUDUSD', ...]
sclients	set	184		['ccc15991', 'ccc160...
sdate1	set	9		[6, 7, 8, 9, 10, 13, 14, 15, 16]
stili	list	3424		[0.0, -665.33, -667...
symbols	str	13		0123456789.-('
thing	str	4		-072
total_profit_per_client	list	184		['ccc15991', 33285...
trades	DataFrame	(3424, 11)		Column names: Trade ...

The Console window shows a traceback for a `TypeError: 'list' object cannot be interpreted as an integer` occurring in `GG_test.py` at line 47.

The following is an example with the total profit per client:



The screenshot shows the Variable Explorer window for the variable `total_profit_per_client`, which is a list of 184 elements. The table below represents the data shown in the explorer:

Index	Type	Size	Value
0	list	2	['ccc15991', 33285.48236302425]
1	list	2	['ccc16026', 42.428513544656]
2	list	2	['eee15951', 122.788983325816]
3	list	2	['aaa15996', 60.45]
4	list	2	['aaa15898', 123.335551945624]
5	list	2	['aaa16010', 174.91130802425604]
6	list	2	['aaa15926', -93.75171439380401]
7	list	2	['aaa15968', -165.778352277212]
8	list	2	['fff15896', 393.15473375144705]
9	list	2	['fff16001', -109.8]
10	list	2	['ccc16005', 20972.429716974406]
11	list	2	['ddd15992', -66.14094771535201]
12	list	2	['fff15980', -16.98]
13	list	2	['fff15917', 0.0]
14	list	2	['aaa16003', 19.447954626692802]
15	list	2	['fff15959', 10760.8950932474]
16	list	2	['eee15993', -164.4592951698]
17	list	2	['fff16050', 7.3]
18	list	2	['ccc15928', -6360.549999999999]
19	list	2	['ddd16020', 47.324017794668]
20	list	2	['eee15930', 0.0]
21	list	2	['bbb16025', -86.81]
22	list	2	['aaa16066', -4758.93478876756]
23	list	2	['aaa15933', 0.0]

The following are the dates with the profit for the company:

Dates	Profit
06.06.22	12514,38
07.06.22	10200
08.06.22	23324,43
09.06.22	16560
10.06.22	5301,14
13.06.22	24469,45
14.06.22	17614,06
15.06.22	27512,79
16/06/22	21519.45

The following are the exchange rates of 6 currencies to euro:

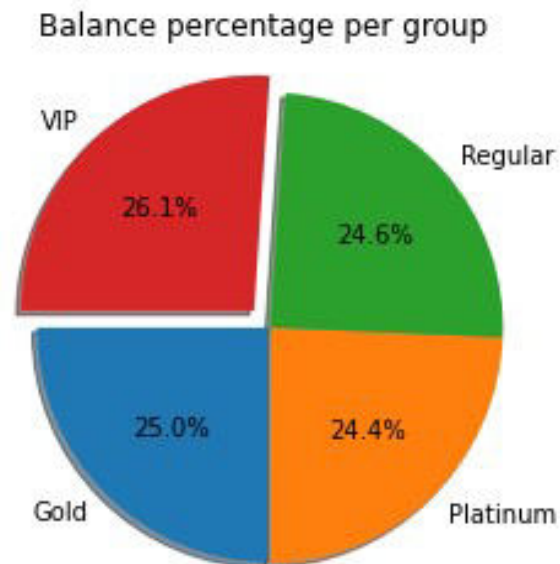
Date	1 EUR	Inv. 1 EUR
USD	1,07	0,9379103358
EUR	1	1
CZK	24,69	0,0405022276
HUF	396,48	0,0025221953
PLN	4,6435	0,215354797
HRK	7,52	0,1329698823

The client that had the biggest losses was ccc15998.

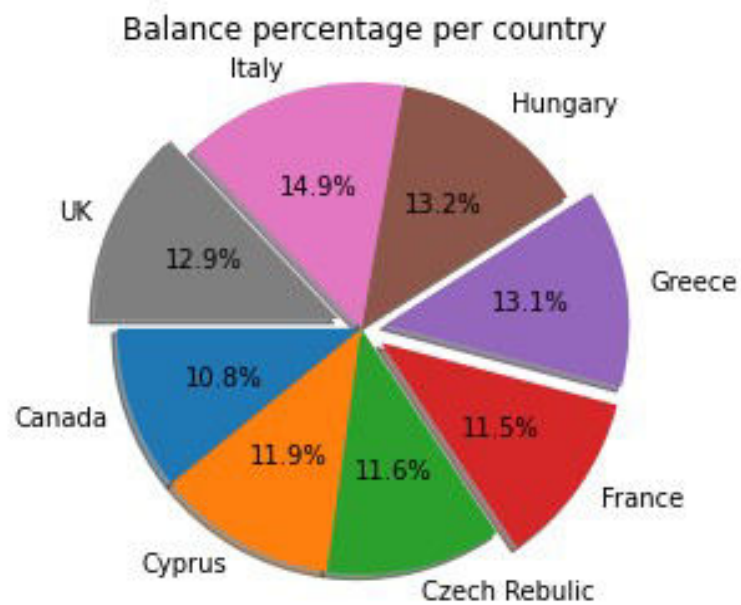
The most traded asset was PBW.

The most losing asset was XAUUSD.

Data visualisation for balance per group:



Data visualisation for balance per country:



Data exploration in SQL:

The screenshot shows the DB Browser for SQLite application. The SQL editor contains the following query:

```
1 SELECT *
2 From clients
```

The results are displayed in a table with the following columns: ClientIdentifier, Date, Asset, and Profit. The data is as follows:

	ClientIdentifier	Date	Asset	Profit
1	fff15896	06/06/2022	WTICrude	0.0
2	eee16056	06/06/2022	WTICrude	-665.33
3	eee16056	06/06/2022	WTICrude	-667.1
4	eee16056	06/06/2022	WTICrude	-1342.59
5	fff15896	06/06/2022	WTICrude	163.54
6	eee16056	06/06/2022	WTICrude	-622.38
7	bbb15906	06/06/2022	US30Cash	0.0
8	bbb15906	06/06/2022	WTICrude	0.0
9	bbb15906	06/06/2022	US30Cash	-40.0
10	bbb15906	06/06/2022	WTICrude	-4.42
11	eee16056	06/06/2022	WTICrude	0.0
12	ggg15918	06/06/2022	WTICrude	275.14
13	fff15973	06/06/2022	USTECHCash	768.73
14	eee16056	06/06/2022	WTICrude	0.0
15	eee16056	06/06/2022	WTICrude	110.83
16	eee16056	06/06/2022	WTICrude	0.0
17	fff16029	06/06/2022	DE30Cash	-2.3
18	eee16035	06/06/2022	XAUUSD	0.0
19	eee16056	06/06/2022	WTICrude	-7276.05
20	eee16056	06/06/2022	WTICrude	60.42

The screenshot shows the DB Browser for SQLite application. The SQL editor contains the following query:

```
1 SELECT *
2 From trading_data
```

The results are displayed in a table with the following columns: Clientname, Group, Country, and BalanceUSD. The data is as follows:

	Clientname	Group	Country	BalanceUSD
1	fff16071	Platinum	CANADA	123,666.00
2	eee16070	Gold	CYPRUS	2,345,677.00
3	ddd16069	Regular	HUNGARY	4,567,444.00
4	ccc16068	VIP	ITALY	8,233,118.74
5	bbb16067	Platinum	GREECE	5,515,549.04
6	aaa16066	Gold	UK	5,101,227.82
7	ggg16065	Regular	Czech Republic	2,602,404.15
8	fff16064	VIP	FRANCE	2,377,691.63
9	eee16063	Platinum	HUNGARY	2,034,422.15
10	ddd16062	Gold	CANADA	1,819,673.04
11	ccc16061	Regular	ITALY	1,736,186.97
12	bbb16060	VIP	CYPRUS	1,677,610.54
13	aaa16059	Platinum	Czech Republic	1,614,794.19
14	ggg16058	Gold	GREECE	1,601,646.14
15	fff16057	Regular	FRANCE	1,599,216.43
16	eee16056	VIP	UK	1,556,583.15
17	ddd16055	Platinum	ITALY	1,497,673.52
18	ccc16054	Gold	HUNGARY	1,443,048.20
19	bbb16053	Regular	CYPRUS	1,309,663.42
20	aaa16052	VIP	CANADA	1,245,340.37