DUCROCQ Romain, PGE

Final Project

Python Data Analysis

- Prepared by: **Romain Ducrocq**
- Supervised by: **Dr Hanan Salam**

Msc In Management - PGE - 2A

December 2020

DUCROCQ Romain, PGE

INTRODUCTION

The Home Credit database is composed of 7 .csv files with data about loan applications and applicants. It contains an application dataset, with the main data about the applicants, and 6 additional datasets, with information on specific related topics. The application dataset is split in a training set and a test set. The training set contains the `TARGET` column, with a binary value indicating whether the client has payment difficulties or not, whereas the column is absent from the test set. The intended goal of this dataset is therefore to train a supervised machine learning model on the 7 datasets, and to then apply it on the test set to predict whether a client has payment difficulties or not. However, we are here in the scope of data analysis, and therefore will not consider the test set.

In this project, we will try to figure out relevant information about clients with payment difficulties. As it takes a very long time to load all the datasets, we will focus on 3 of them: `application_train.csv`, `bureau.csv` and `credit_card_balance.csv`. Furthermore, we will use the help of the `HomeCredit_columns_description.csv` annex dataset, which gives us detailed descriptions of the other dataset's columns.

This data analysis is far from being exhaustive, as it is hard to do a multivariable analysis without the help of dimensionality reduction or regression models, but we will follow a logical progression as we go through the data and, at the end, sum up what we have found out.

ANALYSIS

1 - Random data

First, we observe the columns in `application_train.csv`, and load the dataset.

To begin with, we will look at some random columns that could be of interest in the application_train dataset, regarding payment difficulties.

With the `payment_difficulties_percentage_by_column` function, we get the proportion of each variable in a chosen column as a percentage. The function mainly relies on pandas `groupby`, `sort_values` and `apply`.
Here, we apply it to `NAME_CONTRACT_TYPE` and `NAME_EDUCATION_TYPE`:

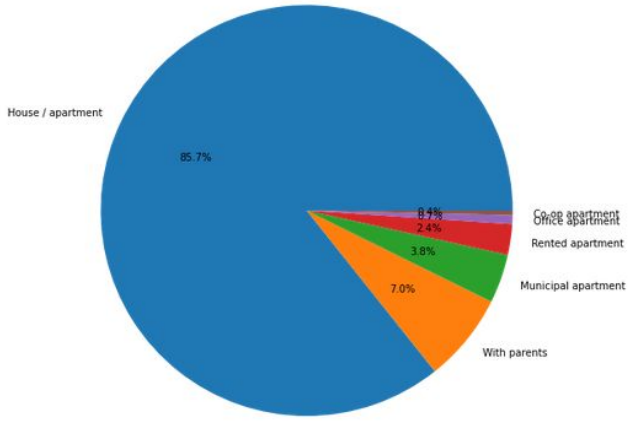
NAME_CONTRACT_TYPE	Percentage
Cash loans	93.54
Revolving loans	6.46

NAME_EDUCATION_TYPE	Percentage
Secondary / secondary special	78.65
Higher education	16.15
Incomplete higher	3.51
Lower secondary	1.68
Academic degree	0.01

We see that an overwhelming majority of payment difficulties are on cash loans, and that the majority of clients with payment difficulties have a secondary education.

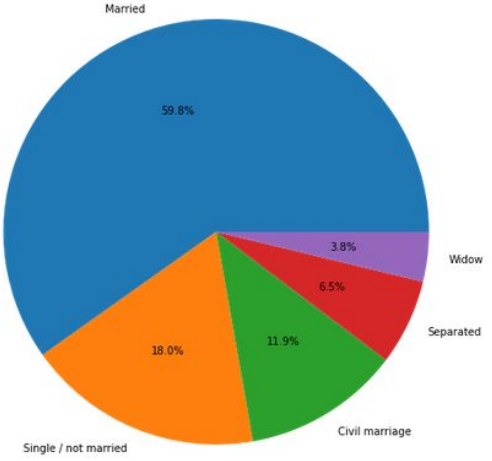
We can also display this information in a more visual way. The function `display_payment_difficulties_percentage_by_column` uses the previous function to display the percentages in a pie chart and a table using matplotlib.
Here, we apply it to `NAME_FAMILY_STATUS` and `NAME_HOUSING_TYPE`:

Payment difficulties by: NAME_HOUSING_TYPE



NAME_HOUSING_TYPE	Percentage
House / apartment	85.69%
With parents	6.99%
Municipal apartment	3.85%
Rented apartment	2.42%
Office apartment	0.69%
Co-op apartment	0.36%

Payment difficulties by: NAME_FAMILY_STATUS



NAME_FAMILY_STATUS	Percentage
Married	59.82%
Single / not married	17.95%
Civil marriage	11.93%
Separated	6.53%
Widow	3.77%

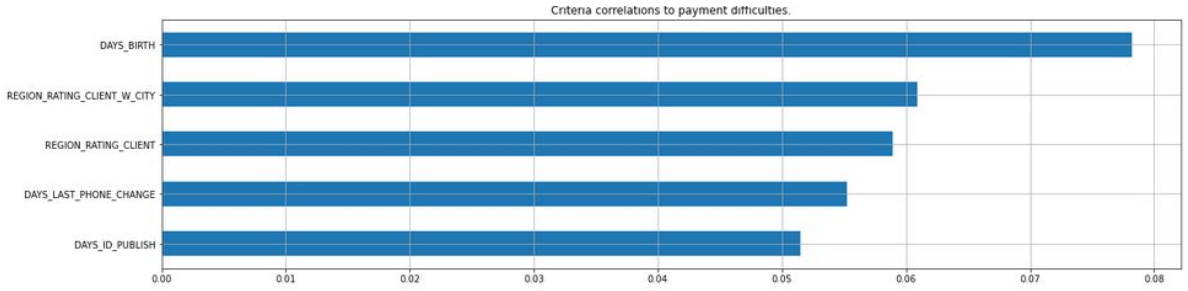
The majority of clients with payment difficulties are either living in a house / apartment and / or are married.

2 - Establishing a profile

Now, we will try to establish the profile of the worst client, i.e. the one with the highest chance to have payment difficulties.

For this, we first look at the correlation between `TARGET` and other columns. We find the 5 most correlated columns, with pandas `corr`, `sort_values` and `head`, and plot them in a matplotlib horizontal bar chart.

```
DAYS_BIRTH                  0.078239
REGION_RATING_CLIENT_W_CITY 0.060893
REGION_RATING_CLIENT        0.058899
DAYS_LAST_PHONE_CHANGE      0.055218
DAYS_ID_PUBLISH             0.051457
```



`DAYS_BIRTH`, `REGION_RATING_CLIENT_W_CITY`, `REGION_RATING_CLIENT`, `DAYS_LAST_PHONE_CHANGE` and `DAYS_ID_PUBLISH` are the five columns that are the most correlated with payment difficulties.

Now, we make a `payment_difficulties_profile_by_column` function to gather the variable with the most occurrences in a chosen column, and apply it to the five columns.

	Profile
DAYS_BIRTH	-11566.0
REGION_RATING_CLIENT_W_CITY	2.0
REGION_RATING_CLIENT	2.0
DAYS_LAST_PHONE_CHANGE	0.0
DAYS_ID_PUBLISH	-4033.0

Thus, the profile being the most likely to have payment difficulties is a 31 years old applicant, living in a region and city with moderate scores, that never changed his phone and changed his IDs 11 years ago for the last time.

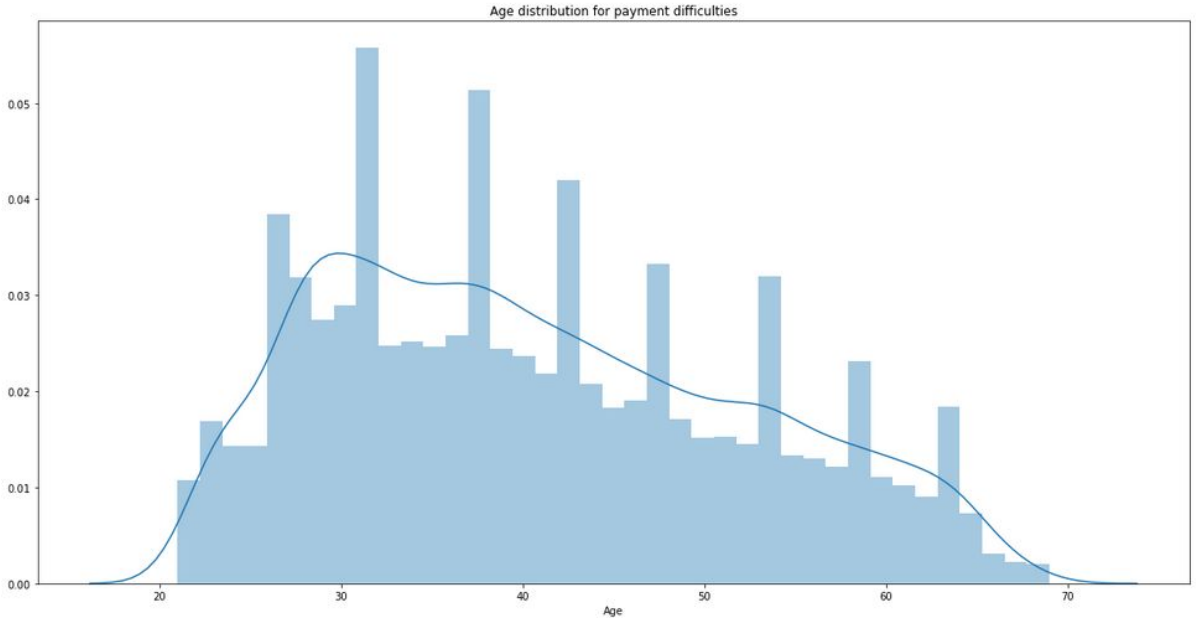
However, this artificially constructed profile does not make much sense as it is way too specific and doesn't take so many factors into account.

Therefore, let's look at the top criterion independently, which is age.

3 - Age and age range distributions

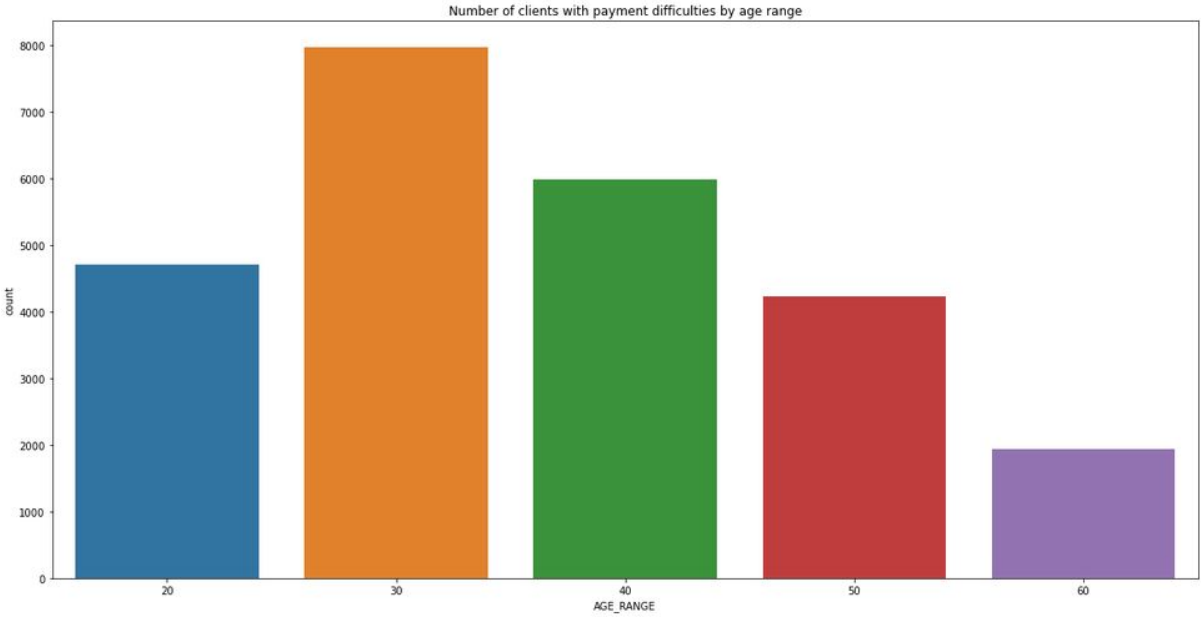
From now on, we extract the payment difficulties data in a new `payment_difficulties` dataframe, in order to not apply the operations to all the rows of the application dataset.

The age in negative days is not handy, so let's convert it in years, by dividing it by 365, multiplying it by -1 and rounding it to 0 decimals, and store it in a new `AGE` column.
We plot the distribution of this new column in a seaborn distplot:



We observe that the clients around 30 years old are indeed the ones with the most payment difficulties.

We can reduce the bins to age ranges, by decades. We divide the ages by 10 and convert them to integers to truncate to decade, and multiply them by 10 again. We store the results in an `AGE_RANGE` column, that we display in a seaborn countplot:



This confirms that the clients in their 30s and 40s are having the most payment difficulties.

4 - Credit type and debt

Now, we will look at another table to find meaningful information to merge with our payment difficulties data and try to reveal interesting patterns.

Thus, we will look at credit types and debts in the bureau dataset.

First, we observe the columns in `bureau.csv`, and load the dataset.

We create a `pivot_table` indexed by `CREDIT_TYPE` with columns that could be interesting to look upon. We chose `CREDIT_ACTIVE`, aggregated by `count`, and `DAYS_CREDIT`, aggregated by `np.mean`, rounded to 0 decimals and multiplied by -1 to get positive values. We then sort the dataframe by descending `CREDIT_ACTIVE`.

CREDIT_TYPE	CREDIT_ACTIVE	DAYS_CREDIT
Consumer credit	1251615	1195.0
Credit card	402195	992.0
Car loan	27690	1329.0
Mortgage	18391	1066.0
Microloan	12413	188.0
Loan for business development	1975	2000.0
Another type of loan	1017	1640.0
Unknown type of loan	555	2213.0
Loan for working capital replenishment	469	653.0
Cash loan (non-earmarked)	56	830.0
Real estate loan	27	782.0
Loan for the purchase of equipment	19	1469.0
Loan for purchase of shares (margin lending)	4	850.0
Interbank credit	1	670.0
Mobile operator loan	1	781.0

Nothing interesting emerges from the `DAYS_CREDIT`, but we see in `CREDIT_ACTIVE` that a massive amount of clients with payment difficulties have a consumer credit.
We will therefore focus only on the clients with payment difficulties and consumer credits.

Therefore, we create a new dataframe `bureau_consumer_credit`, containing the client IDs, their numbers of consumer credits, their total credits and their total debts. At first, we filter the bureau dataset for IDs of clients with a consumer credit, and count the number of occurrences to get the number of consumer credits. This is done with pandas `loc`, `groupby` and `size`. We then left join `AMT_CREDIT_SUM` and `AMT_CREDIT_SUM_DEBT` on `SK_ID_CURR`, with pandas `merge`, `loc`, `groupby` and `sum`. Finally, we sort the values by descending numbers of consumer credit. Here are the 3 first rows of the obtained dataframe:

	SK_ID_CURR	N_consumer_credit	AMT_CREDIT_SUM	AMT_CREDIT_SUM_DEBT
56240	169704	86	7348293.810	878206.500
261932	425396	60	1746000.000	153787.500
157682	295809	57	1819638.675	157747.500

Lastly, we left join this dataframe with the `AGE` and `AGE_RANGE` by payment difficulties on `SK_ID_CURR`, with pandas `merge` and `dropna`. The newly created `payment_difficulties` dataframe now contains age and credit information. Here are the 3 first rows:

	SK_ID_CURR	AGE	AGE_RANGE	N_consumer_credit	AMT_CREDIT_SUM	AMT_CREDIT_SUM_DEBT
0	100002	26.0	20	4.0	724806.00	245781.0
1	100031	51.0	50	5.0	3768151.50	1125000.0
2	100047	48.0	40	3.0	4852134.00	2528203.5

5 - Balance

Now, we want to get information about the balance of the clients to compare all the data.

Thus, we will look at the balance in the credit_card_balance dataset.

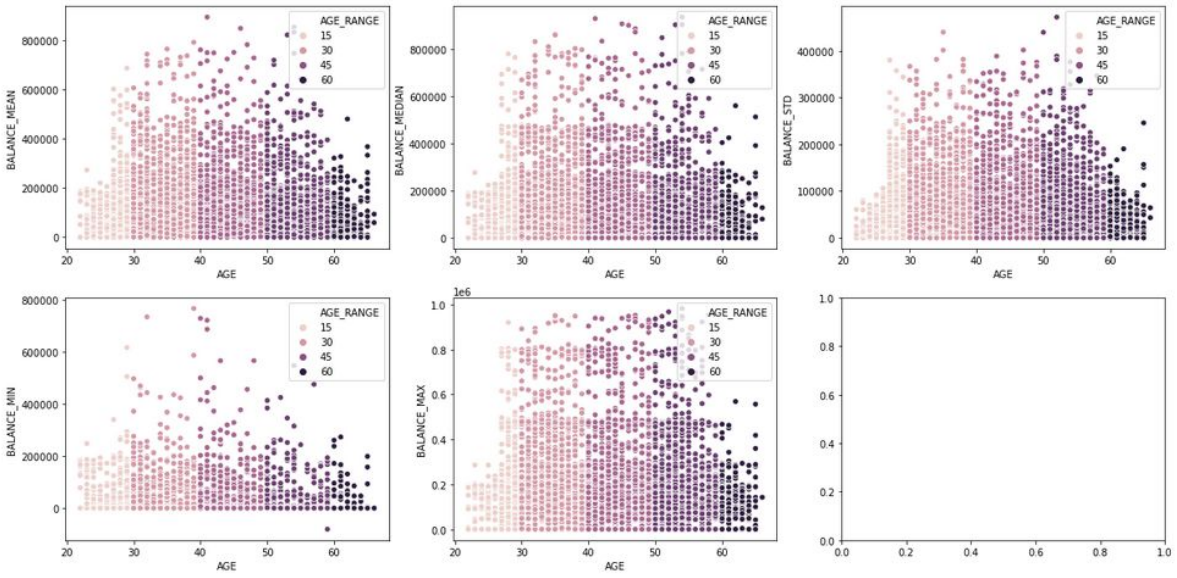
First, we observe the columns in `credit_card_balance.csv`, and load the dataset.

We then create a new `amt_balance` dataframe, with the balance mean, median, max, min and standard deviation for each client. To do this, we create a new column for each information that we left join on `SK_ID_CURR` with pandas `merge` and `groupby`, and a different aggregation function each time, i.e. `mean`, `median`, `min`, `max` and `std`. We then, once again, left join this dataframe with our payment difficulties data on `SK_ID_CURR` with pandas `merge` and `dropna`. Here are the 3 first rows of the obtained dataframe:

	SK_ID_CURR	AGE	AGE_RANGE	N_consumer_credit	AMT_CREDIT_SUM	AMT_CREDIT_SUM_DEBT
15035	381808	52.0	50	18.0	5824937.745	1271371.500
9242	271827	35.0	30	9.0	1136640.375	266116.500
17544	428699	50.0	50	9.0	1460000.430	99418.500

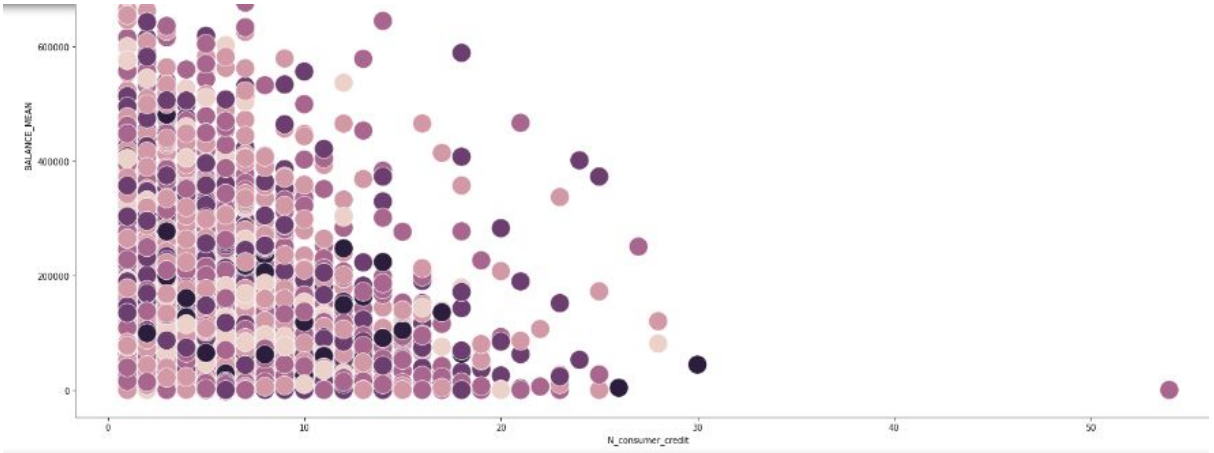
BALANCE_MEAN	BALANCE_MEDIAN	BALANCE_MIN	BALANCE_MAX	BALANCE_STD
407411.763750	23215.7700	0.0	966792.015	471556.935322
456602.053500	540247.4100	0.0	950150.790	439407.924579
464168.529643	521604.9000	0.0	935256.285	439176.095931

Now, we plot the balances mean, median, std, min and max by age, with the age range as hue, into 5 seaborn scatterplot subplots:



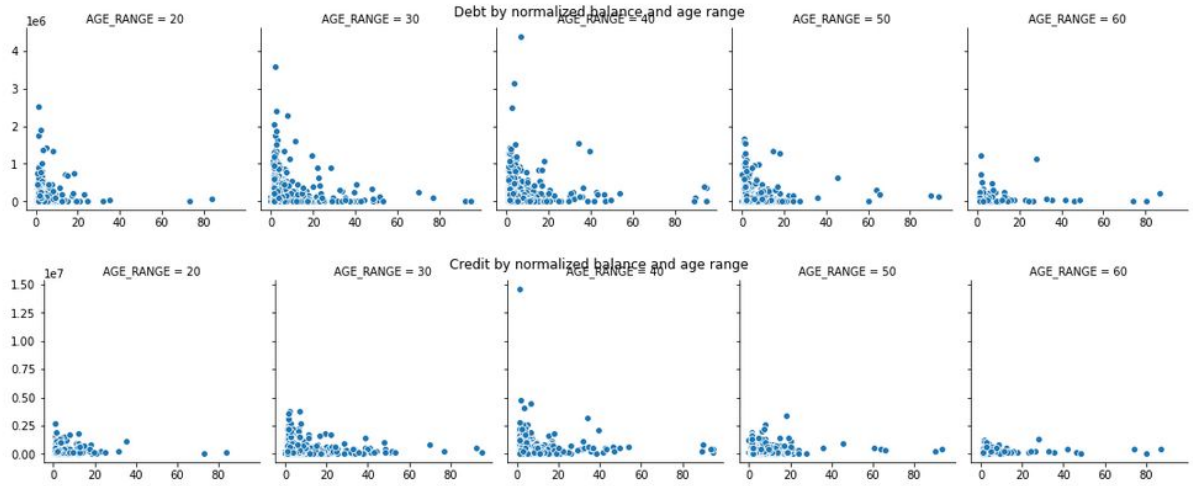
We can clearly see that the clients with payment difficulties under 30 and over 60 have less money in general, and that the range 30 to 60 years old has higher balances. This is not what we could have expected with the age range 30 to 40 being the one with the most payment difficulties. Therefore, the balance alone is not a good indicator of payment difficulties. We could for example assume that clients in their 30s, while having more money than in their 20s, also have much more expenses to deal with.

However, we see a clear relationship between the mean balance and the number of consumer credits, when we plot them in a seaborn relplot:



Here, it appears that the more consumer credits a client has, the lower his mean balance is. However, age doesn't seem to be a strong factor in this case.

Finally, let's take a look at the debt and credit. We display them in two seaborn facetgrids of scatterplots, by age range. For easier visualization, we compare the normalized balance, i.e max balance divided by mean balance, with the unitary debt or credit, i.e. total debt or credit divided by the number of consumer credits.



In the upper row, we see a direct decrease of the balance with the increase of the debt. In the lower row, a similar correlation between balance and credit is seen, but to a much lesser extent, and is thus way weaker. Once again, age doesn't appear as a predominant factor.

6 - Housing and education

The age hasn't told us that much, and we can assume that it implies only correlation with payment difficulties, but not causation.

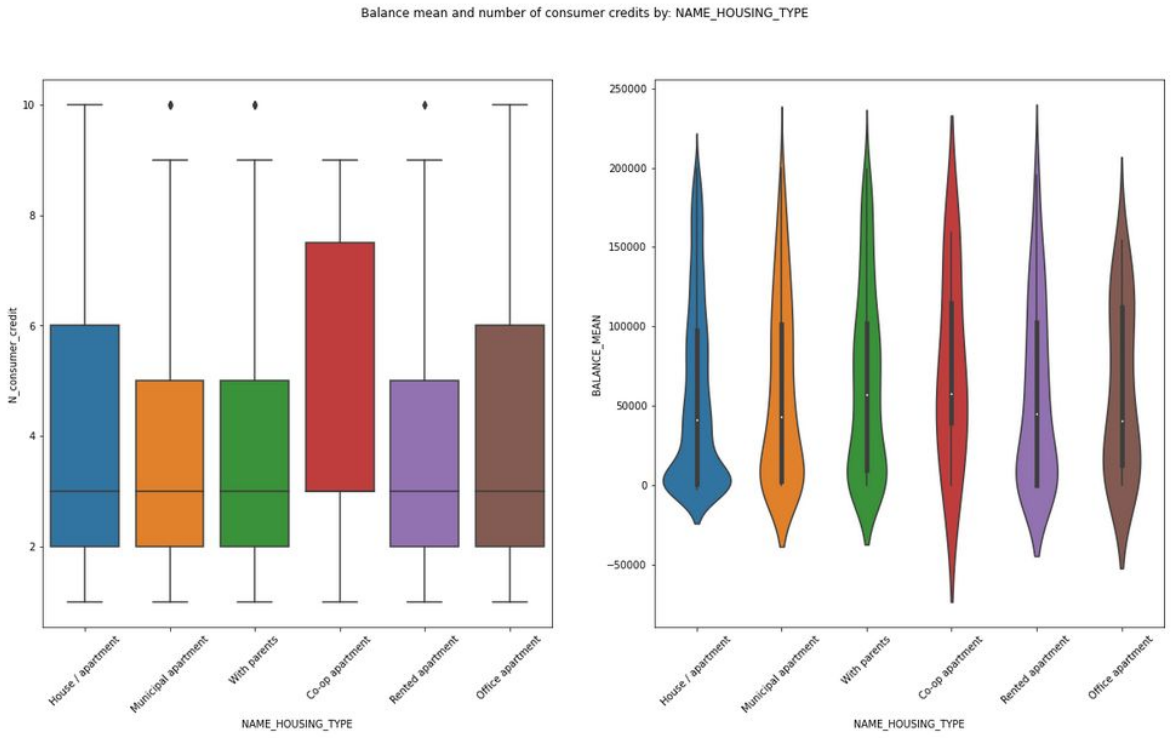
We will look further and add some comparison factors, and dig into other criteria that could be related to the payment difficulties. The education, the housing and the duration since last employment of the clients could be three criteria that could give us valuable information.

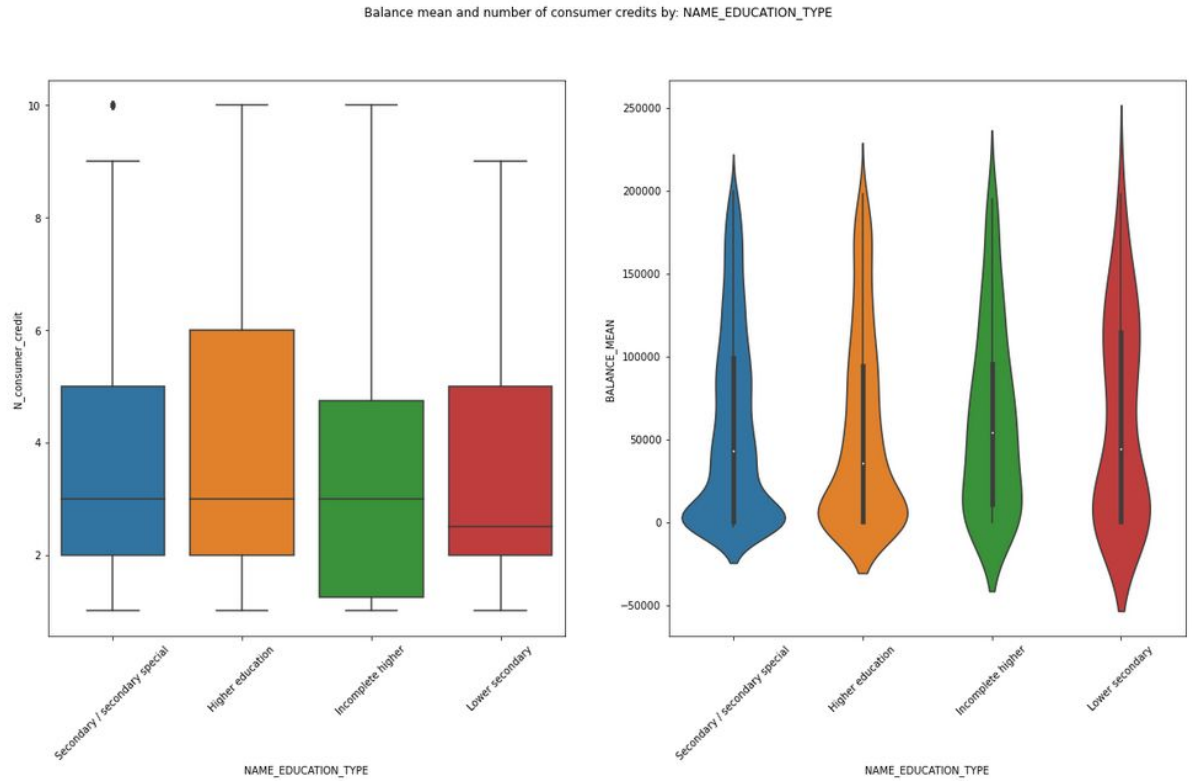
Thus, we left join our payment difficulties dataframe with `NAME_EDUCATION_TYPE`, `NAME_HOUSING_TYPE` and `DAYS_EMPLOYED` from the application dataset on `SK_ID_CURR`, with pandas `merge` and `dropna`. Here are the 3 first rows of the obtained dataframe:

	SK_ID_CURR	AGE	AGE_RANGE	N_consumer_credit	AMT_CREDIT_SUM	AMT_CREDIT_SUM_DEBT	BALANCE_MEAN	BALANCE_MEDIAN
0	100047	48.0	40	3.0	4852134.000	2528203.5	0.000000	0.0000
1	100049	37.0	30	3.0	138767.850	32755.5	48183.296538	10397.1150
2	100181	48.0	40	8.0	2090889.000	0.0	0.000000	0.0000

BALANCE_MIN	BALANCE_MAX	BALANCE_STD	NAME_EDUCATION_TYPE	NAME_HOUSING_TYPE	DAYS_EMPLOYED
0.000	0.000	0.000000	Secondary / secondary special	House / apartment	-1262
0.000	133348.950	54754.919741	Secondary / secondary special	House / apartment	-3597
0.000	0.000	0.000000	Secondary / secondary special	House / apartment	-7676

To easily observe this new data, we create a function `display_n_consumer_credit_by`, that, for a chosen column, displays side by side into 2 subplots a seaborn boxplot with the number of credits and a seaborn violinplot with the balance mean, for each categories in the selected column.
We apply this function to `NAME_HOUSING_TYPE` and `NAME_EDUCATION_TYPE`:

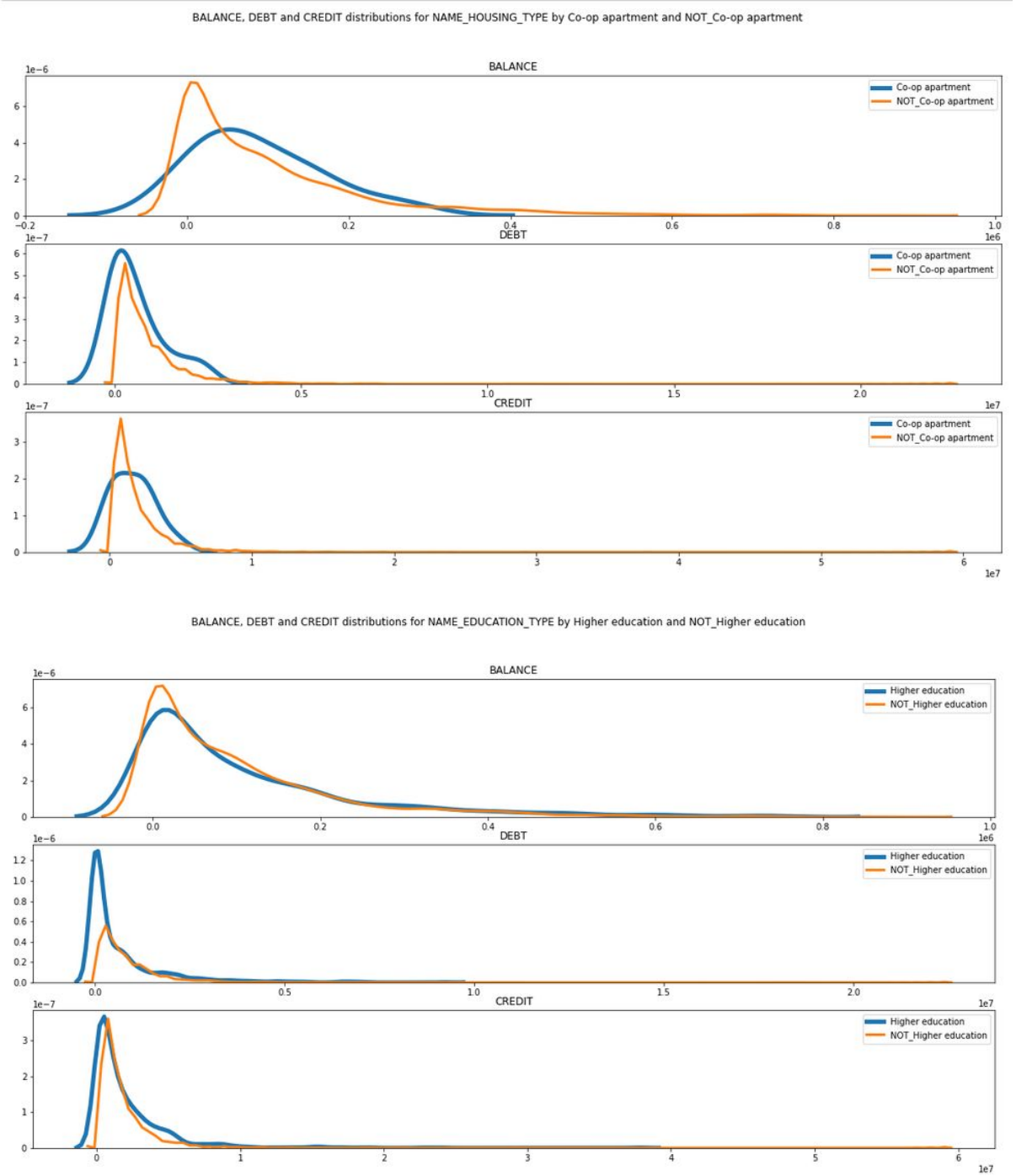




Now, this is interesting. For the housing type, the clients living in co-op apartments are taking much more consumer credits on average, and have a lower mean balance. For the education type, the clients with higher education are also taking much more consumer credits on average.

Thus, we can ask ourselves if these two categories are actually the same clients.

To determine this, we create a `display_comp_categ_notcateg` function, which, for a chosen column and category, displays the inner distributions of the columns for this category and its negation, i.e. for all other categories, by mean balance, total debt and total credit. This is performed with three subplots, each displaying two kdeplots, one for the distribution of the category and one for the distribution of its negation.
When applied to `NAME_HOUSING_TYPE` for `Co-op apartment` and to `NAME_EDUCATION_TYPE` for `Higher education`:



We can see that both have a lower mean balance, but for different reasons. In fact, clients living in co-op apartments have average debt but high credit, while clients with higher education have high debt but average credit.

Thus, we can assume that these are two different types of clients.

We can verify this by getting the number of rows with clients that are both living in a co-op apartment and have a higher education, with pandas conditional `loc` and `len`.

```
Out[42]: 4
```

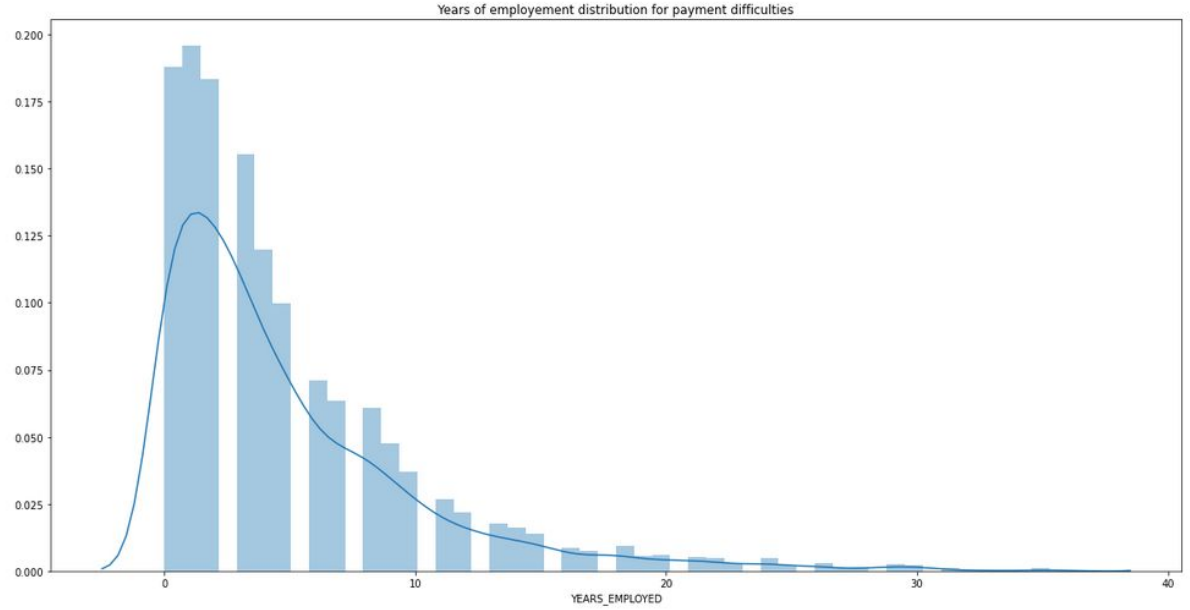
The result is that only 4 clients with payment difficulties are both living in a co-op apartment and have a higher education. They are completely different types of clients.

7 - Duration of employment

Finally, the duration of employment can give us a good hint in relation to payment difficulties.

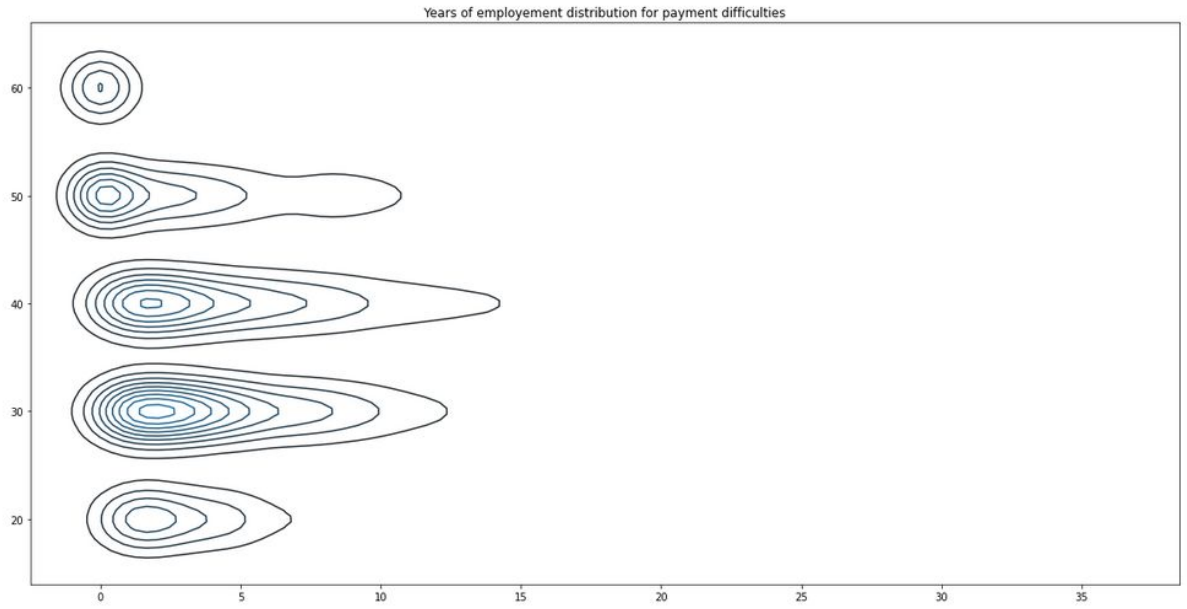
Just like for the age, we transform the negative duration in days to employment durations in years. At first, we see that some incoherent very large positive values, i.e. +958 years of employment, are in the column. We thus filter the `DAYS_EMPLOYED` column to extract only the negative coherent values with a pandas `loc` on negative values. We then divide the remaining values by 365, take their absolute values, round them to 0 decimal, and store them in a `YEARS_EMPLOYED` column.

We can now plot the distribution of the duration of employment in years for the clients with payment difficulties, with a seaborn distplot:



We see that the majority of clients with payment difficulties have been employed for only 0 to 5 years.

Finally, we can also display the density distribution of years of employment by age ranges in a 2 dimensional kdeplot:



Here again, we confirm that the overwhelming majority of clients with payment difficulties are employed for less than 5 years. However, this is true and equally distributed for all age ranges, which could be expected for younger people, but is a significant pattern for older people. This indicates that payment difficulties are tightly linked to employment difficulties and instabilities, as the duration of the last employment does not increase with age.

8 - Saving the dataframe as an SQL table

As far as we have gotten through the analysis, it is, and always will be, incomplete. In fact, we can always find new correlations between data, and discover an infinity of patterns.

Therefore, it is a good idea to save this data for later in an SQL database. To do this, we create a `Payment_difficulties.db` file with `sqlite3.connect`. We then create the table with all its columns in the file by executing a `CREATE TABLE` SQL command with a cursor. Finally, we fill the SQL table with the dataframe, using pandas `to_sql`.

We can apply a simple SQL command on the created file to check the dataframe has been correctly saved, with pandas `read_sql`.
Here, as an example, we select the age, the number of consumer credits, the total debt, the balance mean, the education type, the housing type and the years of employment for all the clients with more than 10 consumer credits and in their 30s, ordered by years of employment, and a limit of 10 results.

```
pd.read_sql(
    "SELECT AGE, N_consumer_credit, AMT_CREDIT_SUM_DEBT, BALANCE_MEAN, \
    NAME_EDUCATION_TYPE, NAME_HOUSING_TYPE, YEARS_EMPLOYED \
    FROM PAYMENT_DIFFICULTIES \
    WHERE N_consumer_credit > 10 AND AGE_RANGE = 30 \
    ORDER BY YEARS_EMPLOYED DESC LIMIT 10",
    conn)
```

	AGE	N_consumer_credit	AMT_CREDIT_SUM_DEBT	BALANCE_MEAN	NAME_EDUCATION_TYPE	NAME_HOUSING_TYPE	YEARS_EMPLOYED
0	39.0	14.0	2308122.000	60224.607947	Secondary / secondary special	House / apartment	20.0
1	38.0	17.0	462555.585	3915.091484	Secondary / secondary special	House / apartment	19.0
2	37.0	14.0	1362077.370	5308.226250	Secondary / secondary special	House / apartment	18.0
3	37.0	16.0	12109.500	0.000000	Higher education	House / apartment	18.0
4	39.0	13.0	35482.500	0.000000	Secondary / secondary special	House / apartment	18.0
5	38.0	18.0	1218870.000	6445.040625	Secondary / secondary special	House / apartment	17.0
6	35.0	16.0	351986.715	0.000000	Secondary / secondary special	House / apartment	17.0
7	37.0	11.0	1987501.635	37846.999432	Secondary / secondary special	House / apartment	17.0
8	38.0	11.0	193149.000	0.000000	Secondary / secondary special	House / apartment	15.0
9	38.0	20.0	1391775.795	0.000000	Secondary / secondary special	House / apartment	15.0

SUMMARY

To sum up this data analysis on clients with payment difficulties:

- The majority have a cash loan, a secondary education, are living in a house or apartment, and / or are married.
- Age is the most correlated factor to payment difficulties.
- A profile of the hypothetical client being the most likely to have payment difficulties has been established.
- The clients in their 30s and 40s are the ones with the most payment difficulties.
- A vast majority of payment difficulties are on consumer credits.
- The balance is not a good indicator for payment difficulties.
- The mean balance decreases with the number of consumer credits.
- There is a strong correlation between high debts and low balances.
- The correlation between credits and balances is less significant.
- Clients living in co-op apartments and clients with higher educations are taking more consumer credits.
- The former have higher credits, whereas the latter have higher debts.
- While having similar payment difficulties, they are not the same clients.
- The majority of clients with payment difficulties are employed for less than 5 years.
- This is true among all age ranges, which implies employment difficulties and instabilities.