FINANCIAL PROGRAMMING

Group Project – Financial Base Table

Project By Arunkkumar Karthikeyan, Siddharth Deshpande, Maria Karakoulian

Problem Statement:

To create a financial base table to analyse the financial status of the customers, and to identify the potential and risky customers and thereby creating correlations between the features (variables) which would help the organization.

Data Exploration:

We have explored the data using the following main functions:

- ***head():*** which in most cases was used to return the first 5 rows of each dataset.
- ***nunique():*** which gave us the number of unique values for specific variables
- ***info():*** which gave us a summary for each variable, containing the number of variables, non-missing values and its type.
- ***shape():*** which gave the number of observations and variables present in each dataset
- ***isna().sum().sum():*** which gave us the total number of missing values in each dataset
- ***describe():*** which shows the statistics such as minimum, maximum, mean, standard deviation of the variables present in each dataset

We have used different customer data that has been provided to us in the form of various tables like loan, account, credit card, daily transactions, demographics, disposition, orders, and client information. There are **5,369 unique clients** and observations with 54 columns in our data mart.
The final base table describes the characteristics of each client (i.e. one observation/row per client).

Libraries:

- ***pandas:*** the library is used to read data files, to convert the variables to date time format, to merge tables and to create dummy variables (one-hot encoding) for categorical variables.
- ***Numpy:*** the library is used to convert the variables to time delta (in months), to round the decimal values, to select certain variables and to replace the null values.
- ***Plotly:*** the library is used to make interactive charts and maps for data analysis and report insights from the analysis.

Data Preparation:

1. Account Table:

- In Account table, we have renamed the values in "frequency" column in English and the 'district_id' to avoid conflict at the merging step.
- We have converted the date column to datetime format and extracted the years, months and days of the date to make it easier further analysis.
- Then we created a variable "LOR" to be able to determine how many months the account exists from the date of account opening (Note: We have considered the current date as 31st Dec 1996).
- At last, we have filtered the observations with account opening date less than or equal to 31st December 1995 as to allow sufficient data for independent variable.

2. Client Table:

- We have transformed the birth number to birth year, birth month and birth day of clients and dropping the birth number.
- We have created Age based on birth year and thereby creating Age group and Age group description for our further analysis.
- We have also extracted the gender as Male and Female based on birth month.

3. Transaction Table:

- We have started our analysis by getting an overview of the data using head(), shape() and info(). Also, we have checked the missing values using isna().sum().sum() and found that there were 2208738 missing values present in the raw dataset.
- We have replaced the missing values in "bank" variable as "Not Applicable" and "account" variable as "Not Available" since these observations are related to partner bank which is irrelevant for our analysis.
- We have replaced the missing values in "operation" variable as "VKLAD" (credit in cash) since the missing fields represents only the credit transactions
- We have replaced the missing values in "k_symbol" variable as "OTHER" and thereby classifying these fields as "Unknown Credit" and "Unknown Withdrawal" based on transaction type (Credit or Withdrawal).
- We have converted the date column to datetime format and thereby extracting the transaction year, month and date to be useful for our further analysis.
- We have created new variables such as trans_type based on type variable, trans_mode based on operation variable, trans_char based on k_symbol variable
- We have found out that there are certain outliers in Amount variable which is below 1st quantile and above 99th quantile. We have not replaced these outliers as we understood that these observations might be useful for our future analysis.
- We have defined class variable and different functions for creating new variables for RFM (Recency, Frequency and Monetary) Analysis which is per account level.
- At last, we have merged the new created variables (27 variables) with the new data frame which has been considered for our final base table.

4. Card Table:

- We have converted the card issued date column to datetime format and thereby extracting the card issued year, month and day and dropping the issued column
- We have also renamed the columns disp_id, card_id and type to avoid conflict at the merging step.
- We have created the target variable "cards issued in 1997" in card table.
- At last, we have filtered the observations where the card issued date is less than or equal to 31st December 1997 (in line with the dependent variable window)

5. Demographic Table:

- We have started our analysis by getting an overview of the data using head(), shape() and info(). Initially, we have renamed the column names to make it more appropriate and explicit.
- We have found out one "?" in variable unemployment_rate_95 and nbr_crimes_95. At first, we have replaced these values as missing (nan) and converted the variable from object type to float. At last, we have replaced the missing values with the median value since the values are skewed towards the median value.
- In addition, we have also created a flag to identify in future which observation has been replaced with the median value.
- Also, we have found out outliers in variables nbr_inhabitants and nbr_crimes_96 which is below 1st quantile and above 99th quantile. We have not replaced these outliers as we understood that these observations might be useful for our future analysis.
- We have created new variable crime_rate_96 (crime rate per 1,000 inhabitants) and thereby segmenting the crime rates as Safe and Dangerous per district.
- At last, we have created a Socio Economic class variable and segmenting the clients as High class, Low class and Middle class.

6. Loan Table:

- We have converted the loan date column to datetime format and thereby extracting the loan issued year, month and day and dropping the date column
- We have segmented the customers basis the loan amount and created new variable as "Category"
- We have created Loan status description and defaulters basis the loan Status variable
- We have also created a flag to identify the customers whether the loan contract has been finished or not and thereby finding out the month remaining to pay the loan and remaining amount to pay. So, the cleaned dataset contains observations per each account (i.e. one observation per account)
- We have created the target variable "loan issued in 1997" in loan table.
- In addition, we have defined class variable to create new variables for Frequency Analysis which is per account level and we have merged the new variables with the original data.
- At last, we have filtered the observations where the loan issued date is less than or equal to 31st December 1997 (in line with the dependent variable window)

7. Order Table:

- In Order table, we have replace the null values in k_symbol as No characterization since there is no appropriate label for these orders.
- Also, we have replaced the values of k_symbol with the characterization of the transaction
- We have created dummy variables (one-hot encoding) for variables bank_to and k_symbol and thereby creating aggregated variables (sum) from dummy variables to denote the sum of orders to each bank and to k_symbol. All these variables represent the observations in account level
- At last, we have merged the new created variables with the new data frame which has been considered for our final base table.

8. Disp Table:

- In disp table, we found out that the number of owners are 4500 and number of disponents are 869
- We have created new variable "count" to describe the number of accounts created per client and thereby merging the results to the original data.
- In addition, we have filtered the observations which are only account Owners (as per the business requirement)

Data Merging :

We have used the cleaned data from the above data pre-processing steps for our merging. Initially, we have started merging our accounts table and disp table to get the observations who are only account owners. On top of that we have merged the client table to get the observations of clients which has matching records with the merged account-disp table. Later, we have merged the loan, card and district tables with the merged table. At last, we have merged the transaction and order table to get the final base table which has one observation per account. There are **2,239 observations and 92 variables** in the final base table which is used for our analysis and getting insights.

```
Entrance [190]: basetable.head()
```

Out[190]:

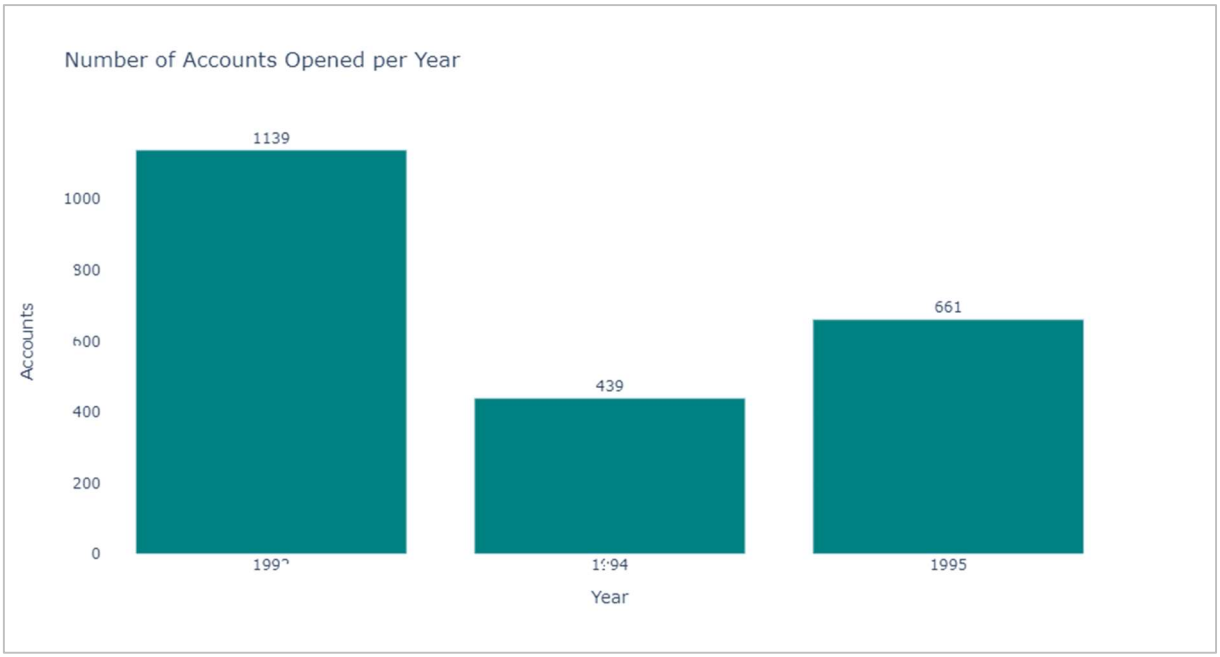
	account_id	bank_district_id	acc_opening_date	LOR in months	client_id	nbr_users	client_district_id	birth_year	birth_day	birth_month	...	order_bank_Is_UV	orc
0	576	55	1993-01-01	47.0	692	2	74	1936	11	1	...	0.0	
1	3818	74	1993-01-01	47.0	4601	2	1	1935	2	4	...	0.0	
2	704	55	1993-01-01	47.0	844	2	22	1945	14	1	...	1.0	
3	2378	16	1993-01-01	47.0	2873	1	16	1975	24	3	...	0.0	
4	2632	24	1993-01-02	47.0	3177	1	24	1938	12	8	...	0.0	

5 rows × 92 columns

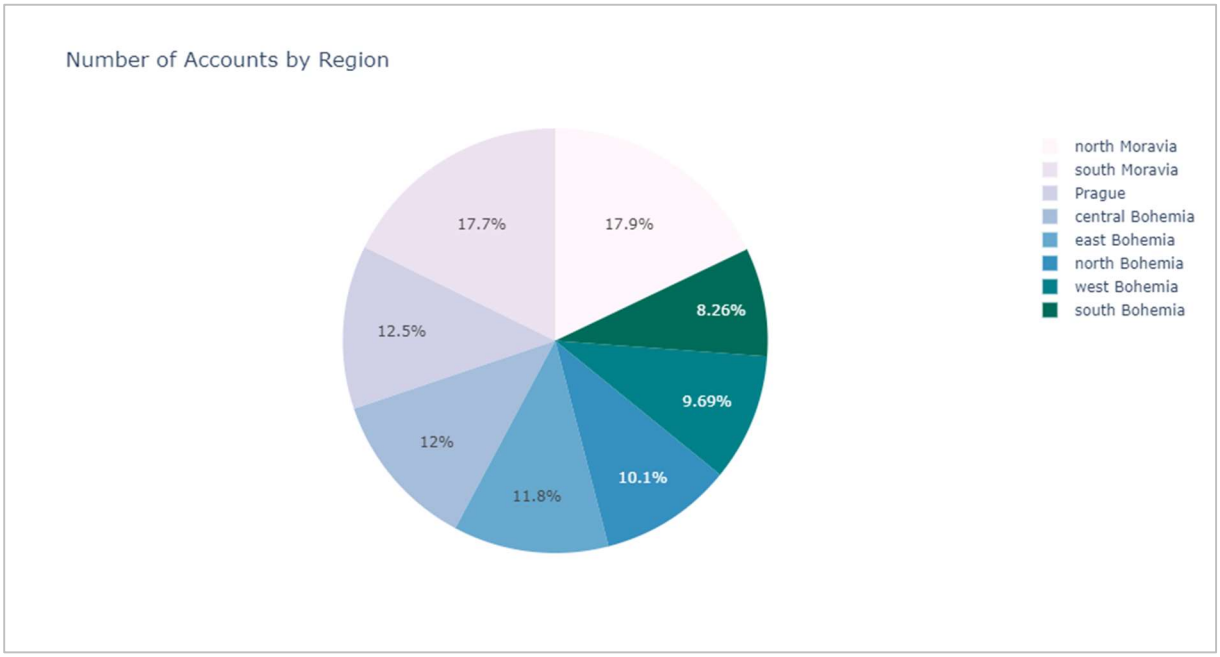
```
Entrance [189]: basetable.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 2239 entries, 0 to 2238
Data columns (total 92 columns):
```

Insights and Data Analysis:

After a sharp decrease in 1994, the number of new accounts opened rebounded to 661 in 1995.



While Prague is the largest city in terms of inhabitants, North Moravia accounted for the largest share of accounts (17.9%), followed by South Moravia (17.7%)

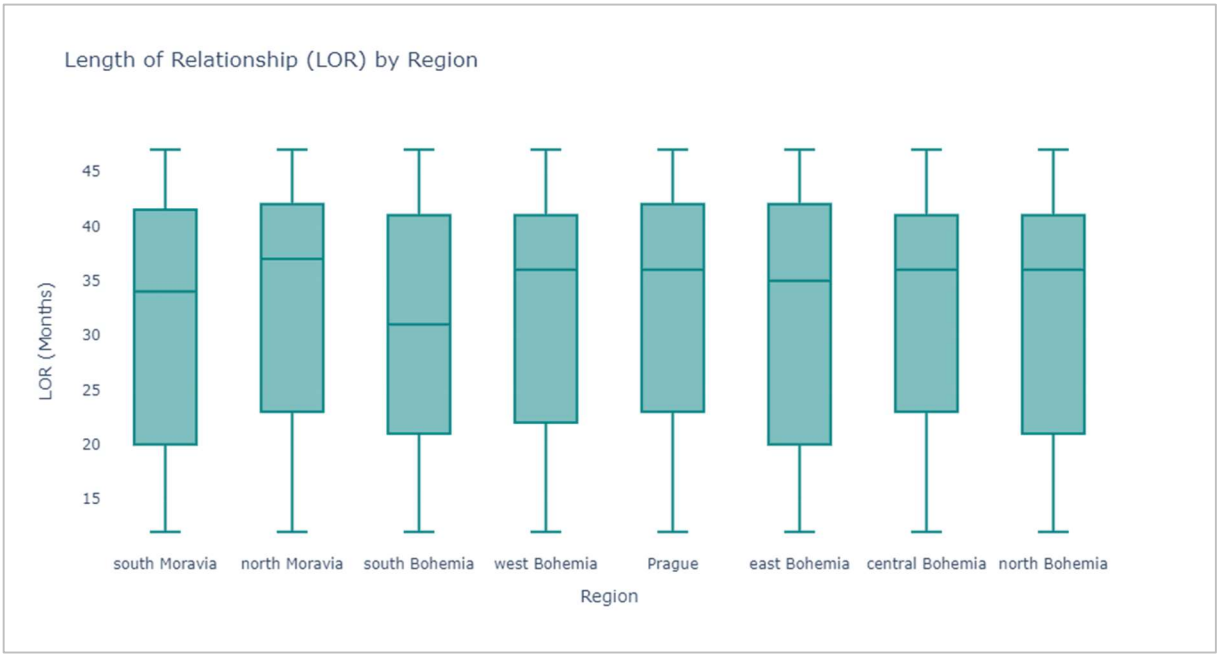


The below table shows the oldest and the newest accounts (first and last two accounts) in each region. Values represent the account ids.

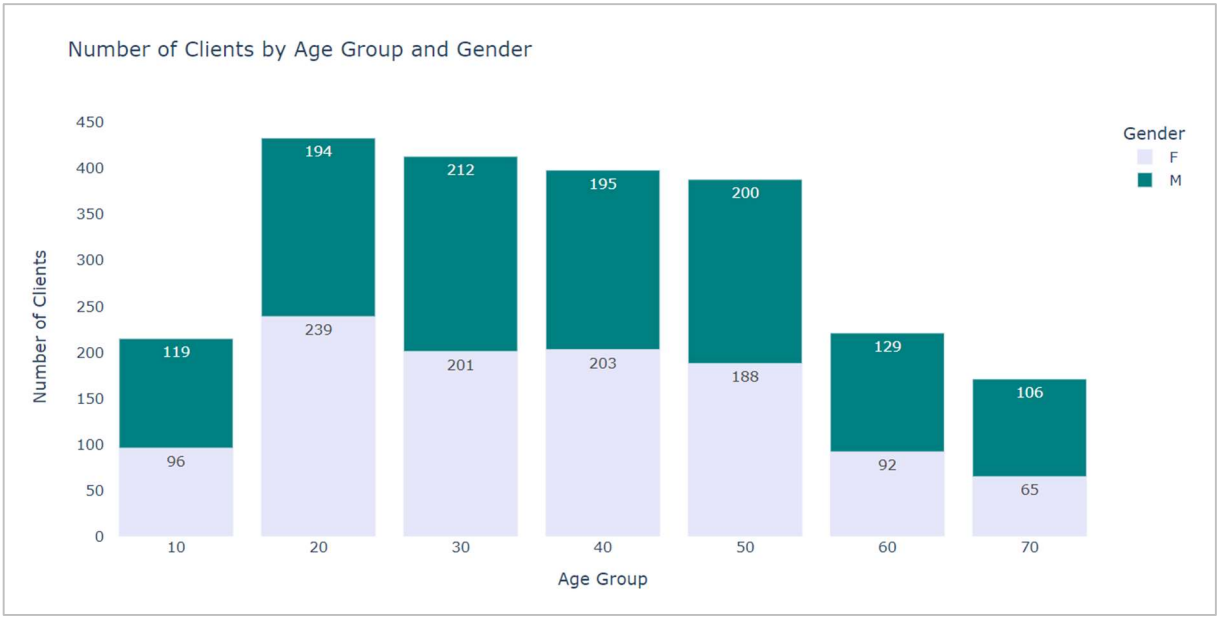
The Oldest and The Newest Accounts

Region	Oldest Accounts	Newest Accounts
Prague	1539	2570
Prague	1637	3655
central Bohemia	485	3476
central Bohemia	2393	8039
east Bohemia	793	1682
east Bohemia	1726	84
north Bohemia	1926	1104
north Bohemia	3510	809
north Moravia	3818	3814
north Moravia	1972	3273
south Bohemia	2378	522
south Bohemia	2357	3559
south Moravia	576	3587
south Moravia	704	2780
west Bohemia	2632	3338
west Bohemia	3871	4439

The below box plot shows the Length of Relationship of clients in each region. We can deduce that South Bohemia has the less senior clients (LOR) when compared to other regions.



The below bar graph represents the number of male and female clients in each age group. We can clearly see that the majority of clients are within the age group of 20-50. We can also notice that clients are almost equally divided in terms of gender.

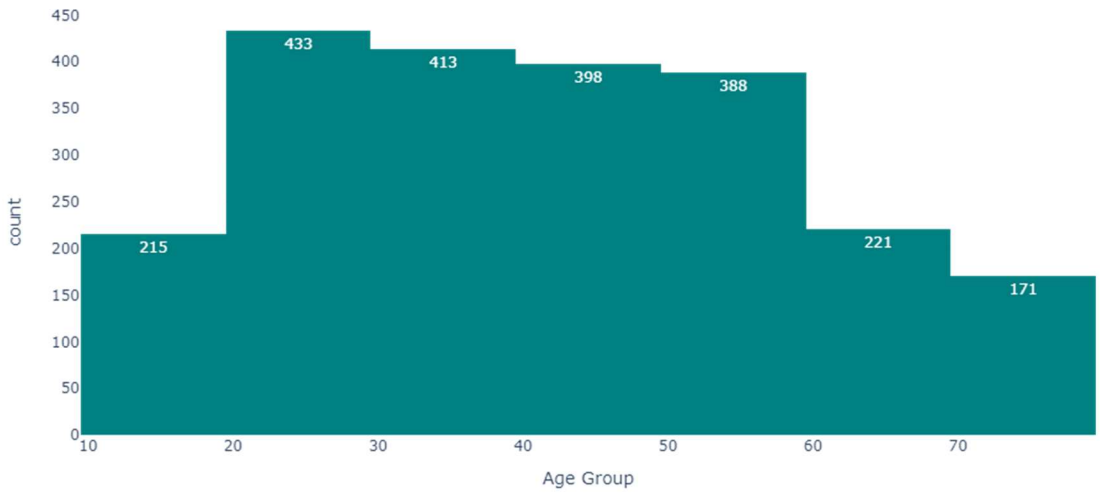


The below table shows the most and least frequent accounts (first and last two accounts) in each region. Values represent the account ids. This information is useful for understanding the frequency of transactions made by the customer in 1996.

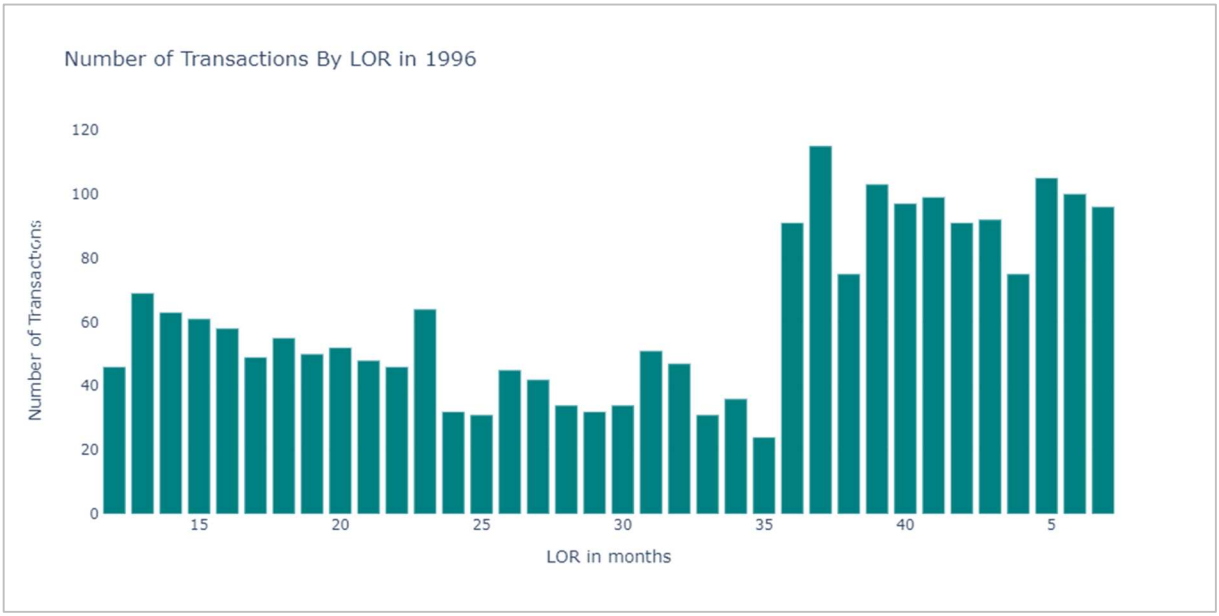
The Most and Least Frequent Accounts in 1996

Region	Most Frequent	Least Frequent
Prague	3643	9265
Prague	2481	8327
central Bohemia	3139	7184
central Bohemia	1359	8261
east Bohemia	459	4079
east Bohemia	838	5952
north Bohemia	1720	7861
north Bohemia	3888	3112
north Moravia	1721	2932
north Moravia	3814	9307
south Bohemia	1091	8405
south Bohemia	3200	5724
south Moravia	2029	7445
south Moravia	799	5422
west Bohemia	2969	767
west Bohemia	138	1766

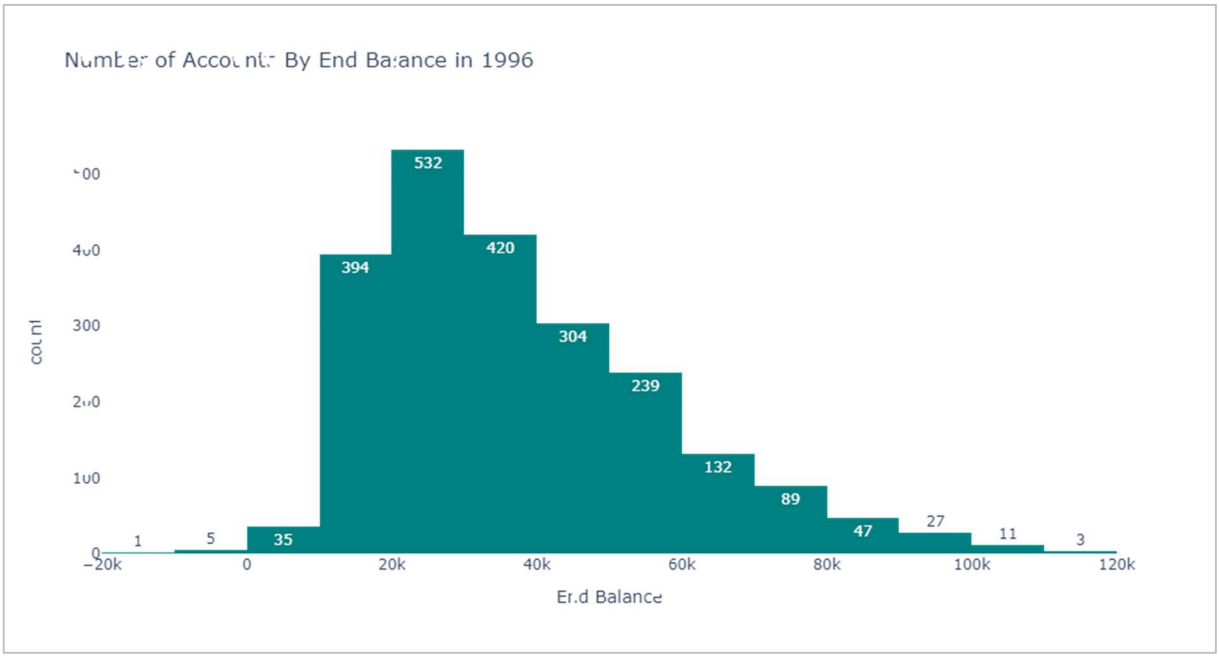
The below histogram represents the number of transactions executed in 1996 by age group. We can deduce that clients within the age group of 20-30 (youths) are the most active.



The below bar graph shows that the number of transactions are higher for the clients who have long been with the bank (i.e. LOR is higher).



The below histogram depicts that the majority of accounts have an end balance ranging between 10-50K.

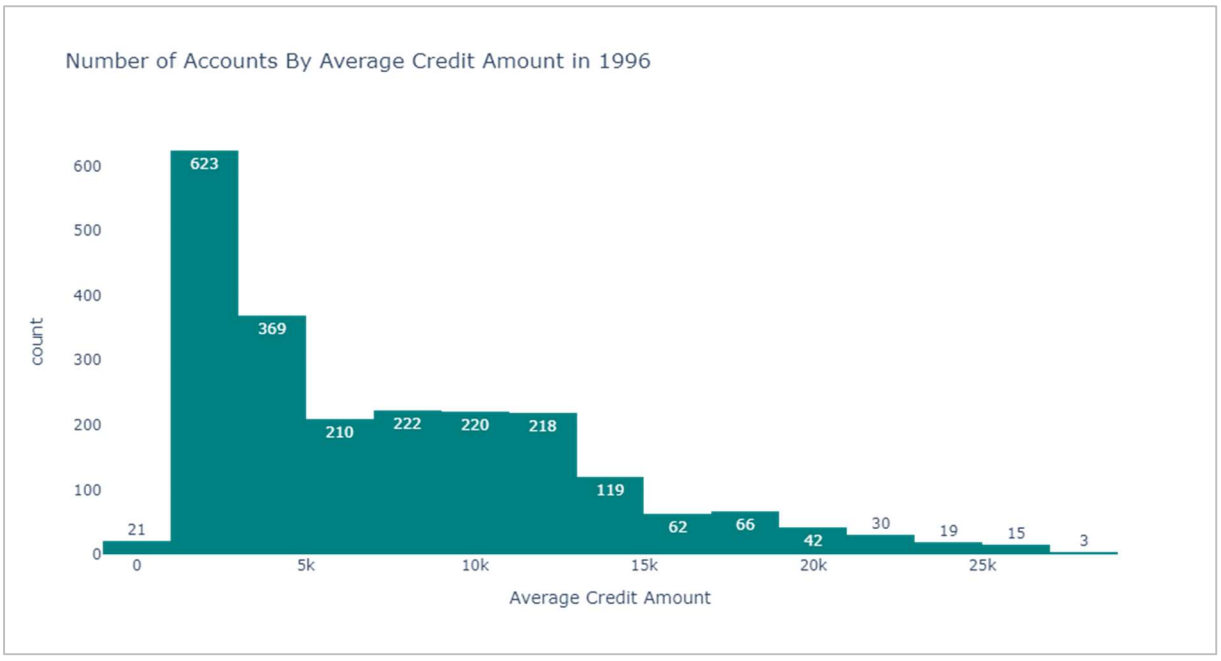


The below table shows the accounts with the largest and smallest end balance (first and last two accounts) in each region. Values represent the account ids. This information is useful to understand the monetary value of each client.

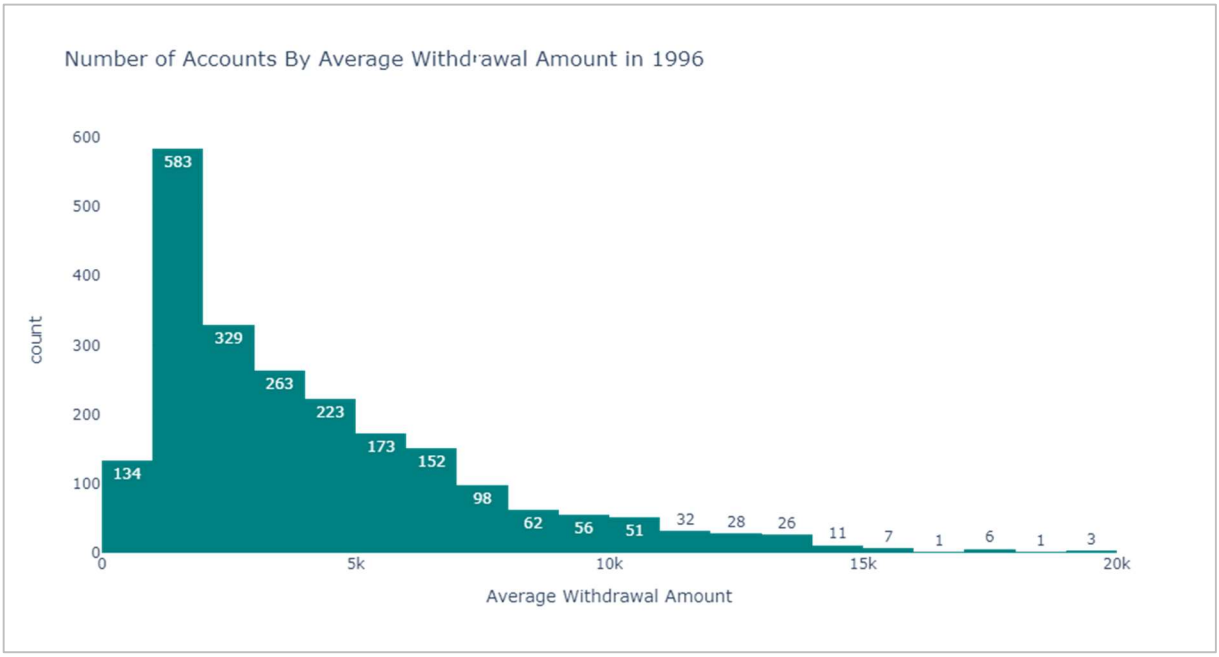
The Accounts with The Largest and The Smallest End Balance in 1996

Region	Largest End Balance	Smallest End Balance
Prague	5125	5125
Prague	2305	2305
central Bohemia	7487	7487
central Bohemia	2859	2859
east Bohemia	3037	3037
east Bohemia	5740	5740
north Bohemia	2464	2464
north Bohemia	1720	1720
north Moravia	6927	6927
north Moravia	2236	2236
south Bohemia	6463	6463
south Bohemia	635	635
south Moravia	6281	6281
south Moravia	7824	7824
west Bohemia	10365	10365
west Bohemia	3407	3407

The below histogram shows that the majority of clients conduct small-sized credit transactions (1-5K).



The below histogram shows that the majority of clients conduct small-sized withdrawal transactions (1-3K).

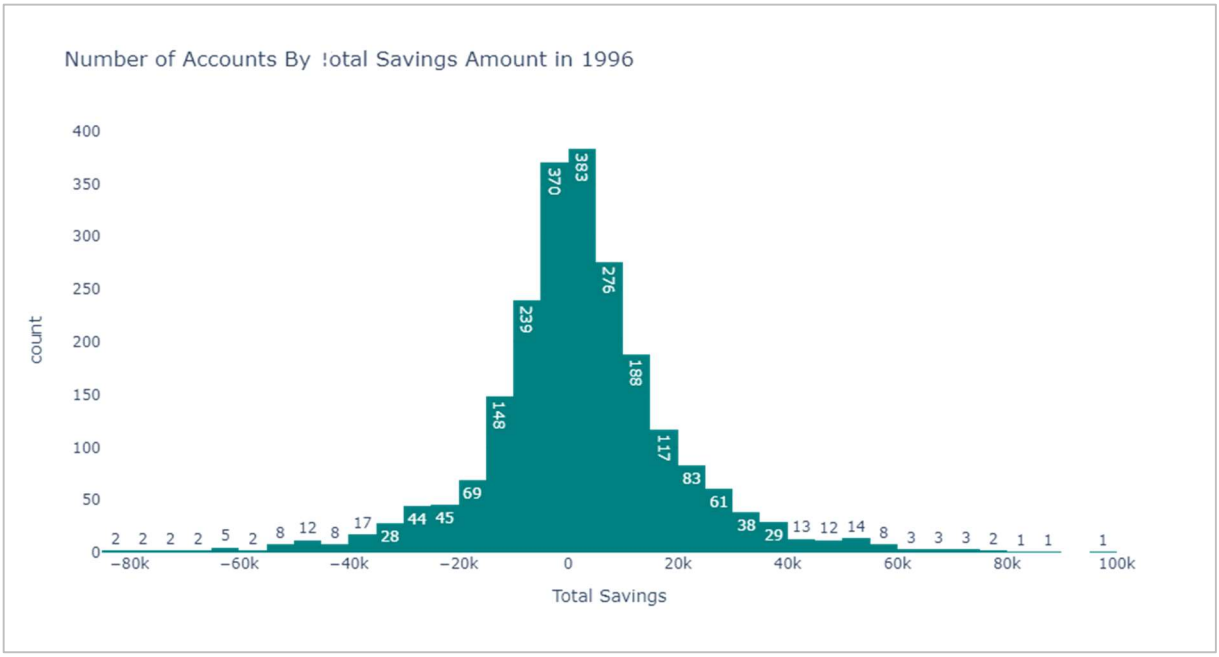


The below table shows the accounts with the largest and smallest end balance (first and last two accounts) in each region. Values represent the account ids. This information is useful to understand the monetary value of each client

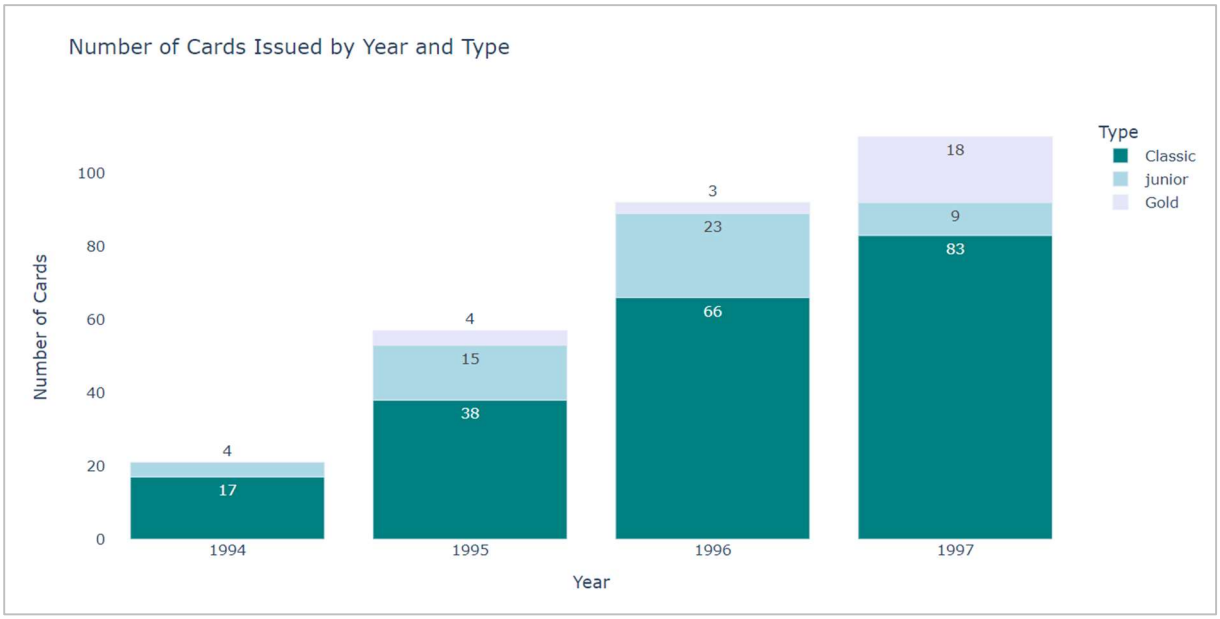
The Accounts with The Largest Average Credit and Withdrawal Transactions in 1996

Largest Average Credit	Largest Average Withdrawal
2170	2894
266	3886
1032	2003
5228	2028
4514	4258
1132	2424
212	624
456	3273
1725	299
2384	1416

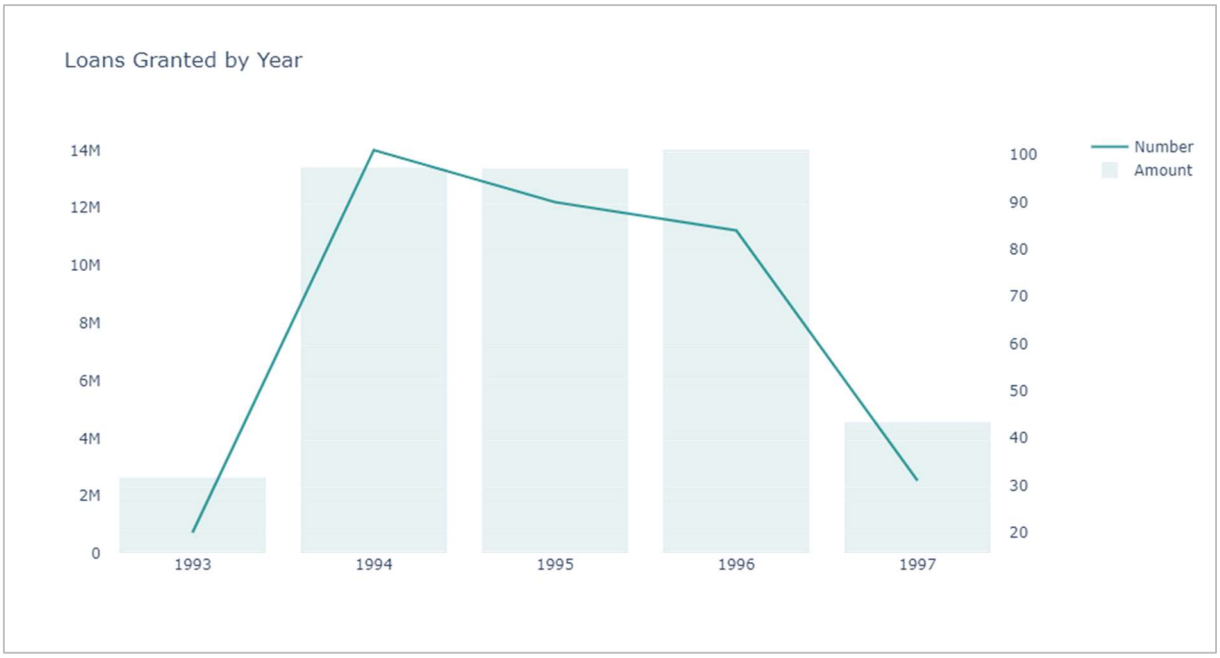
The below histogram indicates that there were equal number of clients who accumulated and lost savings in 1996. Also, most saving gains and losses happened within the range of 0-15K. This is important to target clients who are spending more than earning for a loan.



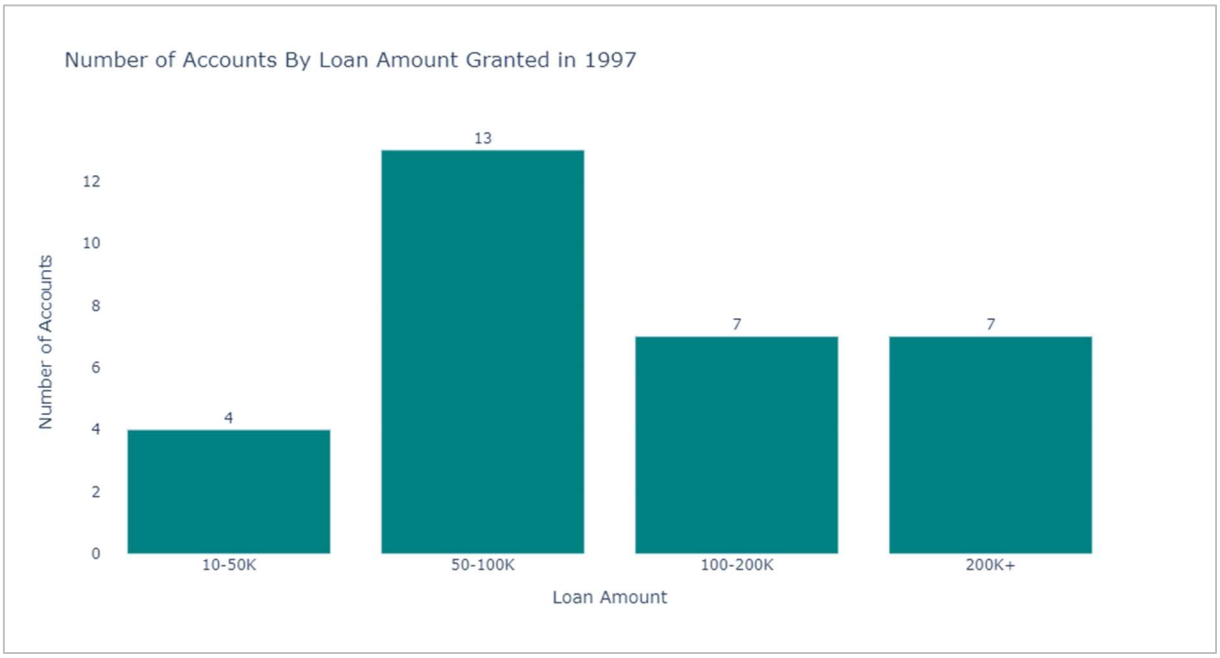
The above bar graph displays the increase in credit cards issued to clients from 1994 to 1997. It also shows that the Classic type is the major issued type of card. Yet it is worth noticing that there has been a jump in gold card issuances in 1997.



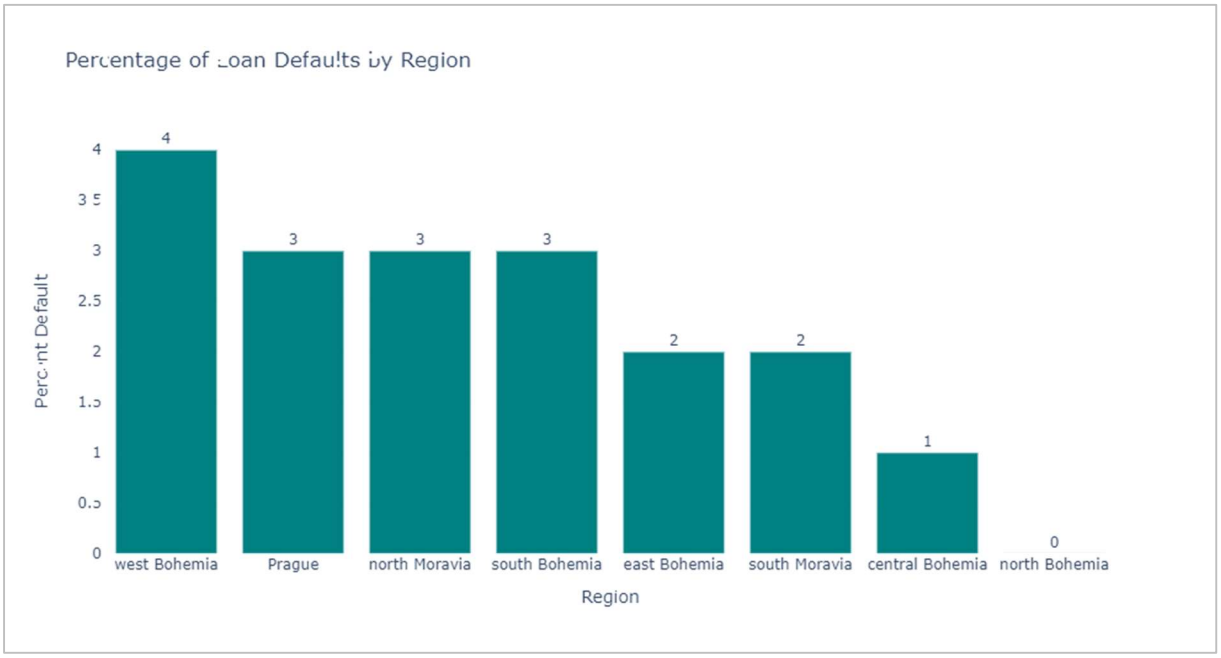
The above graph shows that the number and the amount of loans granted to customers has surged from 1993 to 1994, stayed stagnant from 1994 to 1996 and finally fell in 1997.



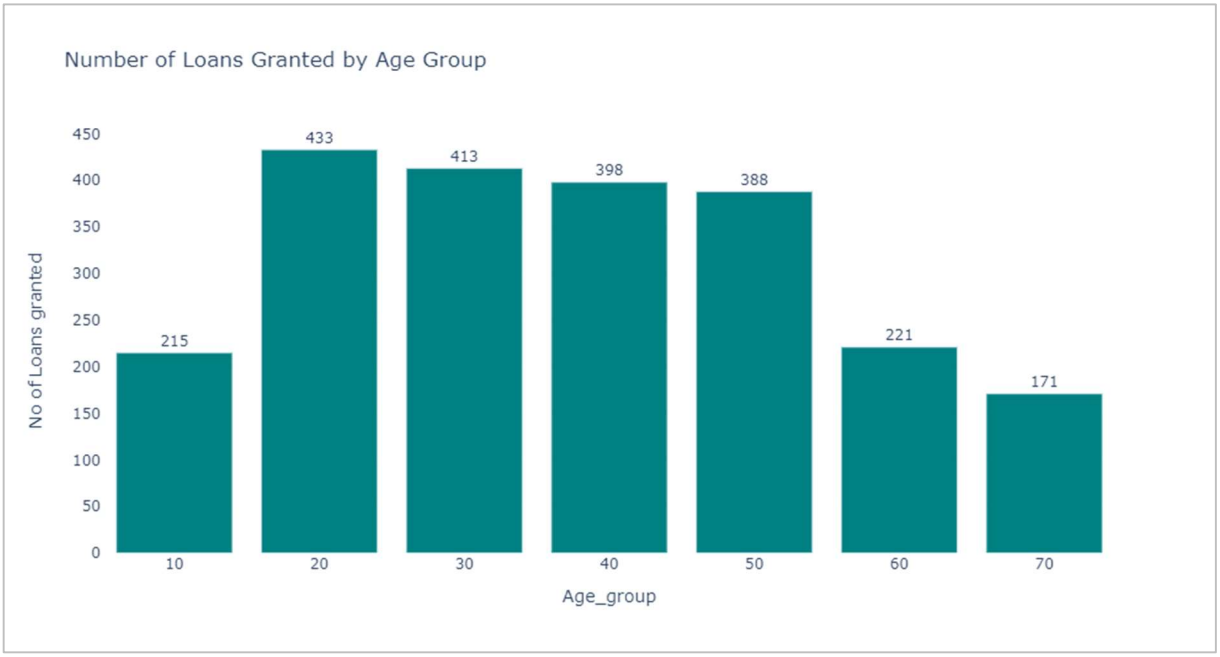
The below bar graph shows that there were high number of accounts which has been granted loan in the range of 50-100k in 1997.



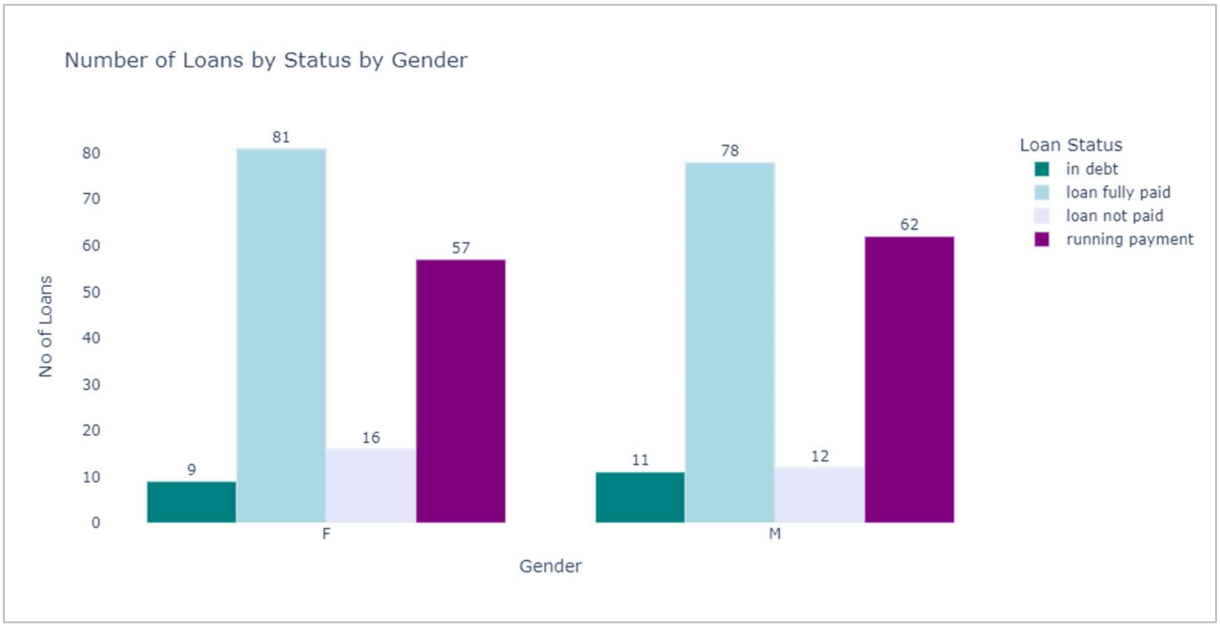
The below bar graph shows the percentage of loan defaults is the highest in west Bohemia while it is 0 is north Bohemia



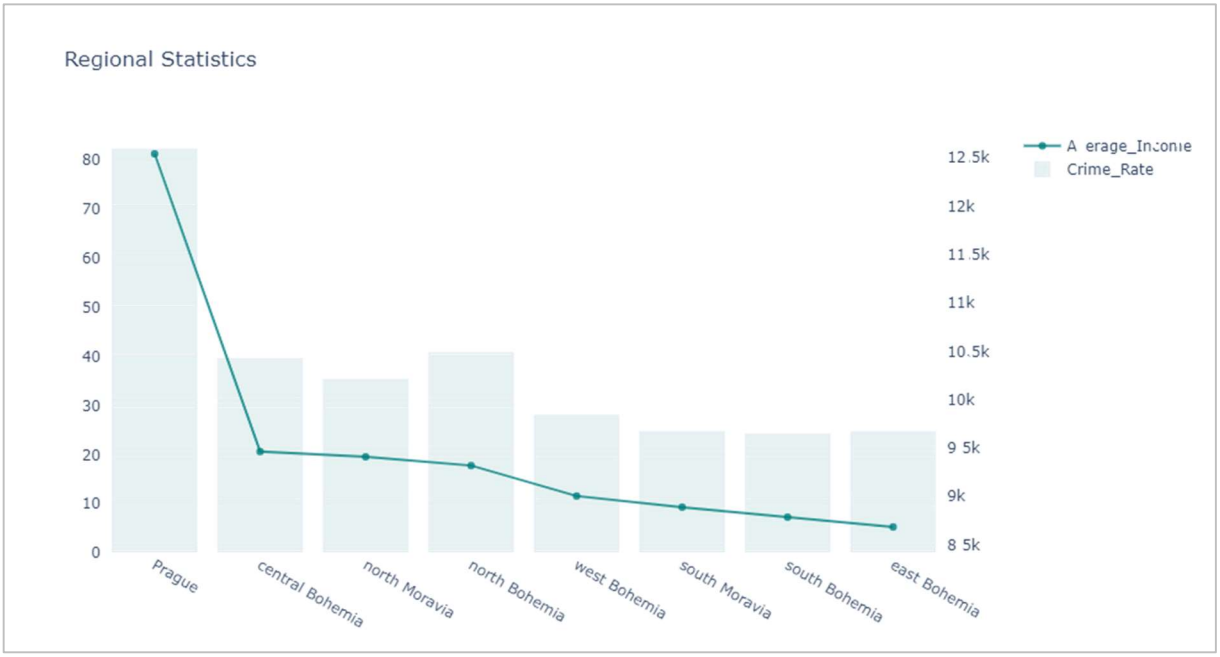
The below bar graph shows that the majority of loans were granted to clients within the age group between 20 to 50 (youths and adults).



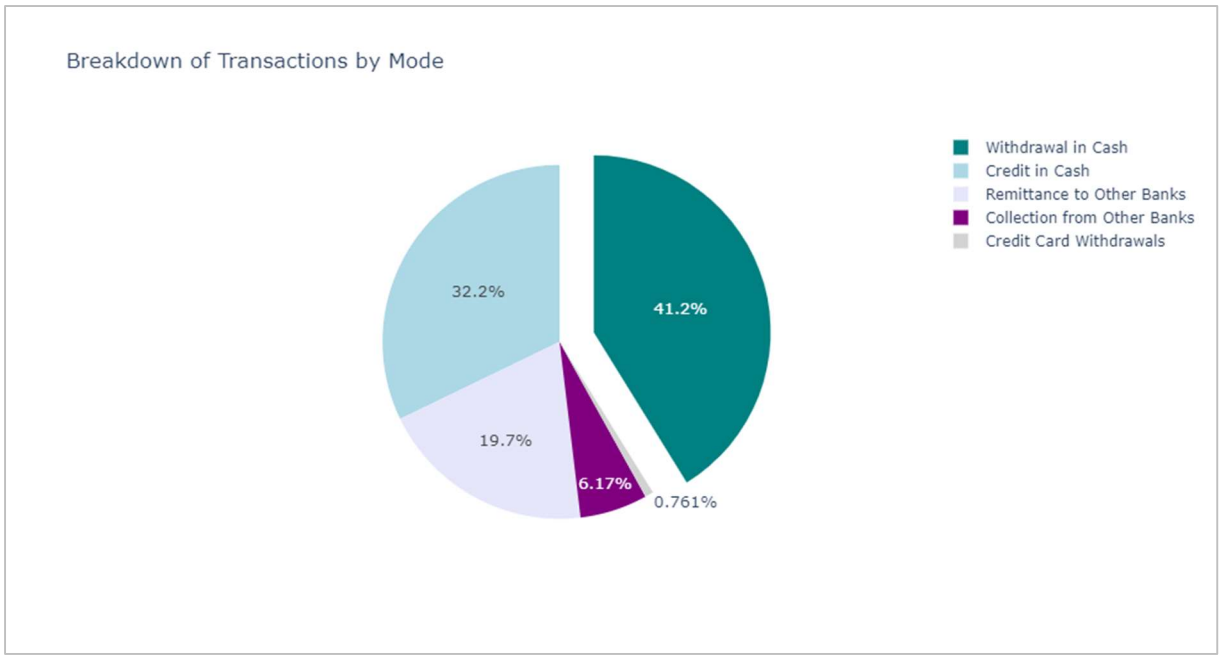
An interesting fact from the below graph is that there are equal number of male and female clients having same figures of loan status. Majority of clients have paid their loans or are in the process of paying.



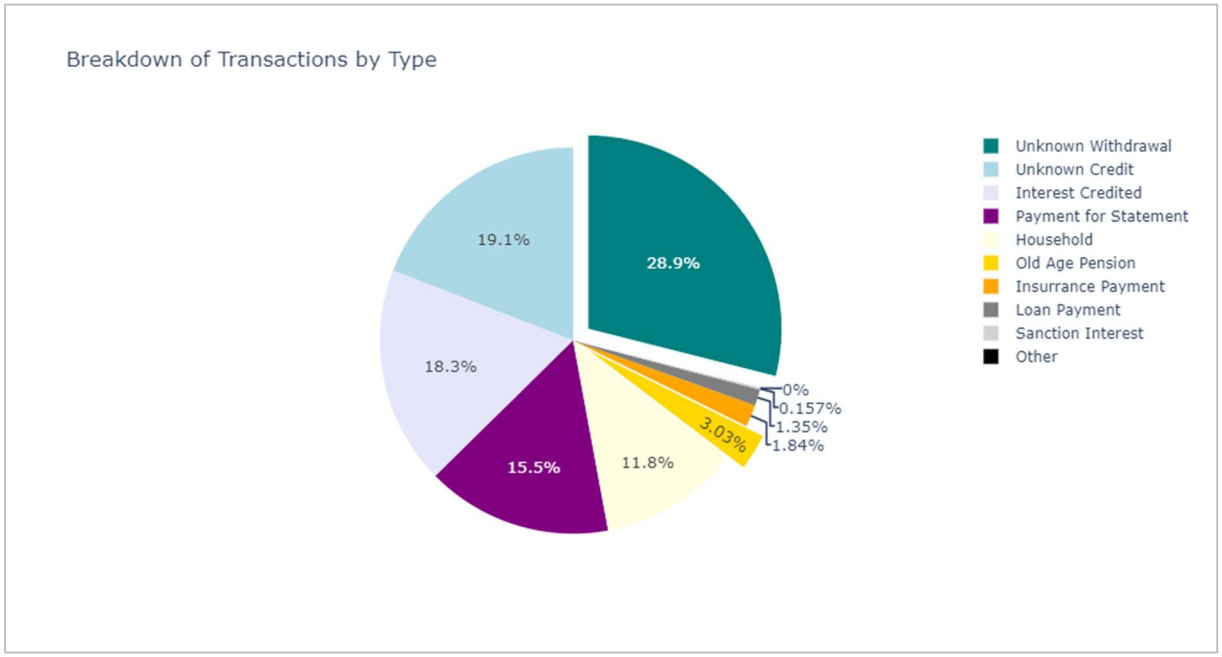
The below bar graph shows that the average income in Prague is significantly higher than other regions. Also, we can observe that there is a positive correlation between the average income and the crime rate across regions.



The below pie chart shows that the majority of transactions has been cash withdrawals and credits.



The above pie chart shows that the majority of transactions were unspecified. Interests accounted for the largest share (18.3%).



Appendix: Variables Description

The base table has 92 variables and below table describes the description and data type of each variable.

Variable Name	Description	From Table	Data Type
account_id	Record identifier for each account	Account	int64
bank_district_id	location of the branch	Account	int64
acc_opening_date	Date the account was opened	Account	datetime64[ns]
LOR (Length of Relationship)	Length of Relationship (Expressed in months since the account was opened)	Account	float64
client_id	Record identifier for each client	Client	int64
Nbr_users	Number of Accounts per Client	Disp	int64
client_district_id	address of the client	Client	int64
birth_year	Birth Year of client. Extracted from birth number of client	Client	int32
birth_day	Birth Day of client. Extracted from birth number of client	Client	int32
birth_month	Birth Month of client. Extracted from birth number of client	Client	int32
gender	Gender of the client, F for female and M for male.	Client	object
Age_group	Age Group of client (bin of 10)	Client	int32
loan_amount	Amount of money the loan is valued	Loan	float64
loan_duration	Time in months of the loan	Loan	float64
loan_monthly_pmts	Amount of money due by month	Loan	float64
loan_year	Year the loan was issued. Extracted from loan date.	Loan	float64
loan _month	Month the loan was issued. Extracted from loan date.	Loan	float64
loan_day	Day the loan was issued. Extracted from loan date.	Loan	float64
loan_granted_in_97_loan	Target variable to describe which clients has granted loan in 1997	Loan	float64
loan_category	Loan Category based on Amount of loan	Loan	object
loan_default	Created a flag to identify the clients with status of Loan Not Paid / client in debt	Loan	float64
loan_status	Status of paying of the loan. A means loan fully paid, B means loan not paid, C means loan payment is still running and D for contract running but client in debt.	Loan	object
Contract_not_finished	Created a flag to identify the clients whether the loan contract has been finished or not	Loan	float64
Contract_finished	Created a flag to identify the clients whether the loan contract has been finished or not	Loan	float64
loan_months_remaining	Remaining months to pay the loan	Loan	float64
loan_amount_remaining	Remaining Loan amount to pay	Loan	float64
loan_recency	Recency of loan issued to client	Loan	timedelta64[ns]
card_type	Type of card ('junior', 'Classic', 'gold')	Card	object
issued_date_card	Date the card was issued to the client	Card	datetime64[ns]
issued_yr_card	Year of card issued. Extracted the year from card date	Card	float64
card_issued_in_97	Target variable to describe which clients has credit cards being issued in 1997	Card	float64
district_id	Record identifier of district	District	int64
district_name	Name of the District	District	object

region	Region of the district	District	object
nbr_inhabitants	Number of inhabitants per district	District	int64
ratio_urban_inh	Ratio of inhabitants	District	float64
average_salary	Average salary per district	District	int64
unemployment_rate_96	Unemployment rate in 1996	District	float64
nbr_entp_1000_inh	no. of entrepreneurs per 1000 inhabitants	District	float64
unemployment_rate_missing	Created a flag to show the observation where missing value has been replaced with numeric value	District	float64
crime_rate_96	Created a flag to show the observation where missing value has been replaced with numeric value	District	float64
SEC	Socio-Economic Class	District	object
High-Class	Created dummy variable to show clients from High Class district	District	uint8
Low-Class	Created dummy variable to show clients from Low Class district	District	uint8
Middle-Class	Created dummy variable to show clients from Middle Class district	District	uint8
last_trans_date	Last transaction date of the account	Trans	datetime64[ns]
recency	Recency of transactions of accounts in 1996	Trans	timedelta64[ns]
freq_total_LTM	Frequency of transactions of accounts in 1996	Trans	int64
freq_credit_LTM	Frequency of credit transactions of accounts in 1996	Trans	float64
freq_withdrawal_LTM	Frequency of withdrawal transactions of accounts in 1996	Trans	float64
end_balance	Final account Balance of each account in 1996	Trans	float64
total_credit_LTM	Total Amount of credit transactions of each accounts in 1996	Trans	float64
total_withdrawal_LTM	Total Amount of withdrawal transactions of each accounts in 1996	Trans	float64
avg_credit_LTM	Average Amount of credit transactions of each accounts in 1996	Trans	float64
avg_withdrawal_LTM	Average Amount of withdrawal transactions of each accounts in 1996	Trans	float64
freq_total_LSM	Frequency of transactions of accounts in last 6 months of 1996	Trans	int64
freq_credit_LSM	Frequency of credit transactions of accounts in last 6 months of 1996	Trans	float64
freq_withdrawal_LSM	Frequency of withdrawal transactions of accounts in last 6 months of 1996	Trans	float64
total_credit_LSM	Total Amount of credit transactions of each accounts in last 6 months of 1996	Trans	float64
total_withdrawl_LSM	Total Amount of withdrawal transactions of each accounts in last 6 months of 1996	Trans	float64
avg_credit_LSM	Average Amount of credit transactions of each accounts in last 6 months of 1996	Trans	float64
avg_withdrawal_LSM	Average Amount of withdrawal transactions of each accounts in last 6 months of 1996	Trans	float64
freq_total_LM	Frequency of transactions of accounts in last month of 1996	Trans	float64
freq_credit_LM	Frequency of credit transactions of accounts in last month of 1996	Trans	float64
freq_withdrawal_LM	Frequency of withdrawal transactions of accounts in last month of 1996	Trans	float64

total_credit_LM	Total Amount of credit transactions of each accounts in last month of 1996	Trans	float64
total_withdrawl_LM	Total Amount of withdrawal transactions of each accounts in last month of 1996	Trans	float64
avg_credit_LM	Average Amount of credit transactions of each accounts in last month of 1996	Trans	float64
avg_withdrawal_LM	Average Amount of withdrawal transactions of each accounts in last month of 1996	Trans	float64
Total_order_amount	Total of amount of order.	Order	float64
No_of_orders	Total number of orders per account	Order	float64
order_bank_Is_AB	Created a flag to identify if the order Bank is AB	Order	float64
order_bank_Is_CD	Created a flag to identify if the order Bank is CD	Order	float64
order_bank_Is_EF	Created a flag to identify if the order Bank is EF	Order	float64
order_bank_Is_GH	Created a flag to identify if the order Bank is GH	Order	float64
order_bank_Is_IJ	Created a flag to identify if the order Bank is IJ	Order	float64
order_bank_Is_KL	Created a flag to identify if the order Bank is KL	Order	float64
order_bank_Is_MN	Created a flag to identify if the order Bank is MN	Order	float64
order_bank_Is_OP	Created a flag to identify if the order Bank is OP	Order	float64
order_bank_Is_QR	Created a flag to identify if the order Bank is QR	Order	float64
order_bank_Is_ST	Created a flag to identify if the order Bank is ST	Order	float64
order_bank_Is_UV	Created a flag to identify if the order Bank is UV	Order	float64
order_bank_Is_WX	Created a flag to identify if the order Bank is WX	Order	float64
order_bank_Is_YZ	Created a flag to identify if the order Bank is YZ	Order	float64
order_k_Is_leasing	Created a flag to identify if the order stands for Leasing	Order	float64
order_k_Is_insurrance	Created a flag to identify if the order stands for insurance payment	Order	float64
order_k_Is_household	Created a flag to identify if the order stands for household payment	Order	float64
order_k_Is_loan	Created a flag to identify if the order stands for loan payment	Order	float64
order_k_Is_NO CHAR	Created a flag to identify if the order stands for no characterization	Order	float64
Year	Year extracted from account opening date of client	Account	object
total_savings	Total Savings of each account	Trans	float64

References

https://numpy.org/doc/stable/reference/arrays.datetime.html

https://stackoverflow.com/questions/33346591/what-is-the-difference-between-size-and-count-in-pandas

https://www.analyticsvidhya.com/blog/2020/02/joins-in-pandas-master-the-different-types-of-joins-in-python/#:~:text=Left%20Join%20in%20Pandas,-Now%2C%20let's%20say&text=Left%20join%2C%20also%20known%20as,columns%20in%20the%20right%20dataframe

https://numpy.org/doc/stable/numpy-user.pdf

https://pandas.pydata.org/docs/reference/api/pandas.get_dummies.html

https://www.altexsoft.com/blog/what-is-data-mart/

https://www.talend.com/resources/what-is-data-mart/

https://matplotlib.org/stable/gallery/color/named_colors.html

https://community.plotly.com/t/text-position-inside-for-label-and-outside-for-value-pie-chart/8952/4

Financial Programming class Sessions Illustrations and Jupyter Notebooks

Predictive and Descriptive Class Session Jupyter Notebooks