Loan Data Characterization

Author: Lim Sien Long
Creation Date: 28 Mar 2023

Objective:

This report is to provide insight into the factors that affect loan issuance and performance. It analyses time, population, geographical and transactional information to understand their relationship with loan data.

Approaches

1. Understanding the data
    - Referred to PKDD'99 Discovery Challenge document to understand the meaning of each table and column.
2. Data cleaning
    - Loaded dataset from provided mysqldump file
    - Cleaned all data and initiated constraints (refer to DATA CLEANING APPENDIX and data_wrangling.sql)
3. Formed hypotheses
4. Analysed data to evaluate hypotheses
5. Conclusion and recommended actions

Hypotheses:

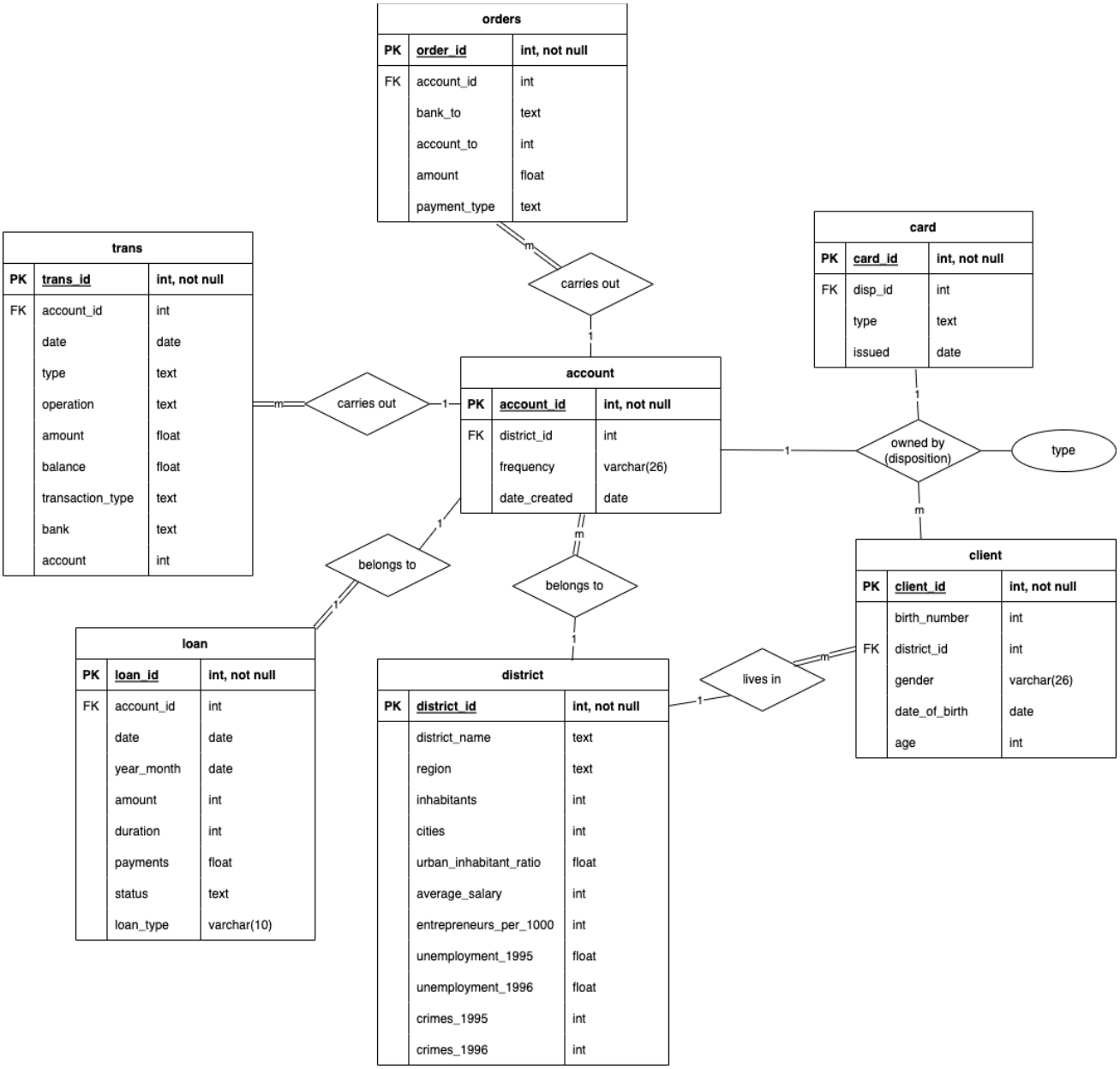
The research will focus on helping the bank's loan manager and officers better plan and target their loan issuance.

1. "Loan issuance follows an upward trend over long periods of time."
2. "Geolocation and population affect loan issuance and loan performance."
3. "Customer transaction behaviour is an indicator for loan performance."

Definitions:

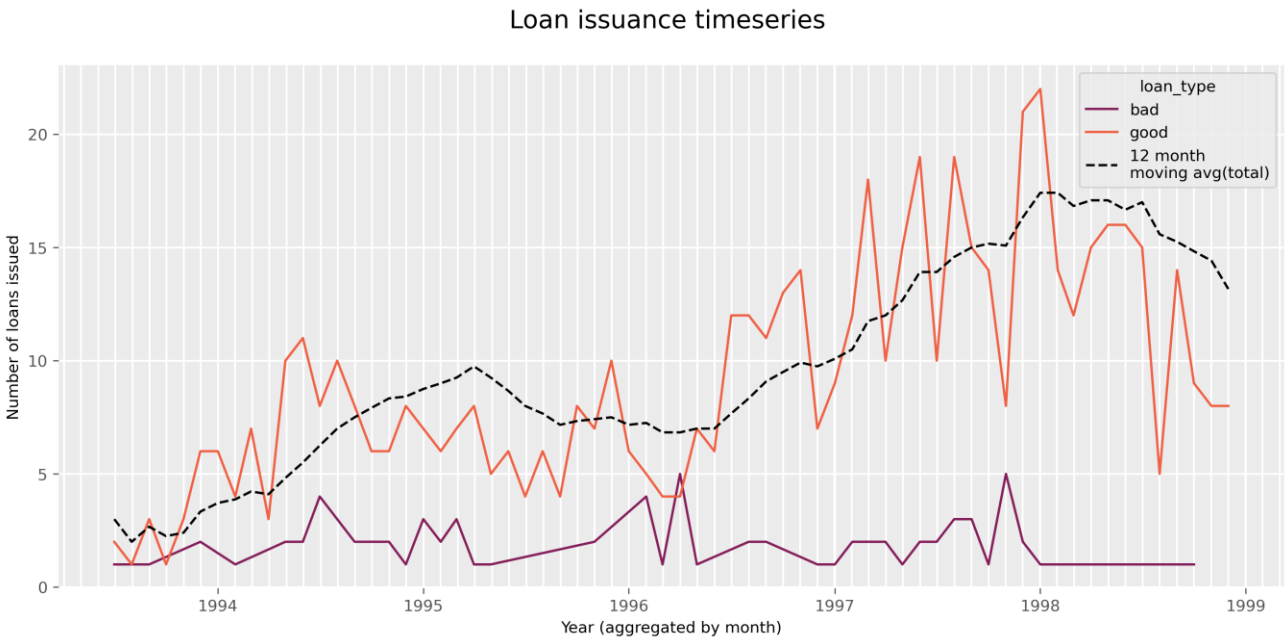
- Loan issuance – number of loans granted
- Loan issuance date – date when the loan was granted
- Loan performance:
    - Good loan
        - loans that have been repaid in full, or likely to be repaid on time with interest
        - for this dataset, we classify loans with status A and C as good loans
    - Bad loan
        - loans that have not been repaid in full, and have missed payments
        - for this dataset, we classify loans with status B and D as bad loans
- Region – Large geographic areas within the Czech Republic, contains one or more districts
- Districts – Smalller geographic area, belong to one region, may contain rural or urban areas or both

Schema

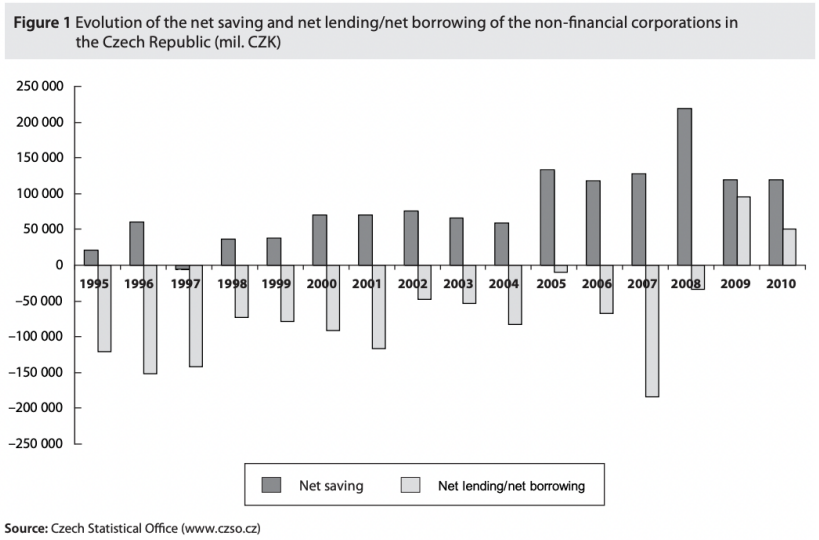


The data consists of financial data from Czech Republic bank accounts starting from years 1993 to 1998 (6 years).

Hypothesis 1 - “Loan issuance follows an upward trend over long periods of time.”



- The 12 month moving average of total loans followed an upward trend over most of the 6 years.
- This agrees with my hypothesis that loan issuance increased over a long timeframe (> 5 years).
- There were two periods with a downward trend Apr 1995 – Apr 1996 and Feb 1998 – Dec 1999.
- A quick search online yielded the following discovery
    - [source] There was a decrease of economic activity caused by the beginning of the economic transformation in the Czech Republic in the period after 1990. After this short (1991–1993) period, economic growth took place with a peak in 1995–1996, immediately followed by an economic crisis with decreasing GDP in 1997–1998.



Additional findings:

- The ratio of good loans to bad loans improved in general, especially between Apr 1996 and Jan 1998. This means that more good loans were issued in comparison to bad loans.

- In general, there were never more than 5 bad loans in a single month.

SQL snippet:

```
WITH rolling_avg_table AS
(SELECT
    `year_month`,
    AVG(COUNT(loan_id)) OVER(ORDER BY `year_month` ROWS BETWEEN 11 PRECEDING AND CURRENT ROW) AS 'moving_avg'
FROM loan
GROUP BY `year_month`
)
SELECT
    l.year_month AS year,
    l.loan_type,
    COUNT(l.loan_id) AS "Loans issued",
    moving_avg
FROM loan l
INNER JOIN rolling_avg_table r ON (l.year_month = r.year_month)
GROUP BY l.year_month, l.loan_type;
```

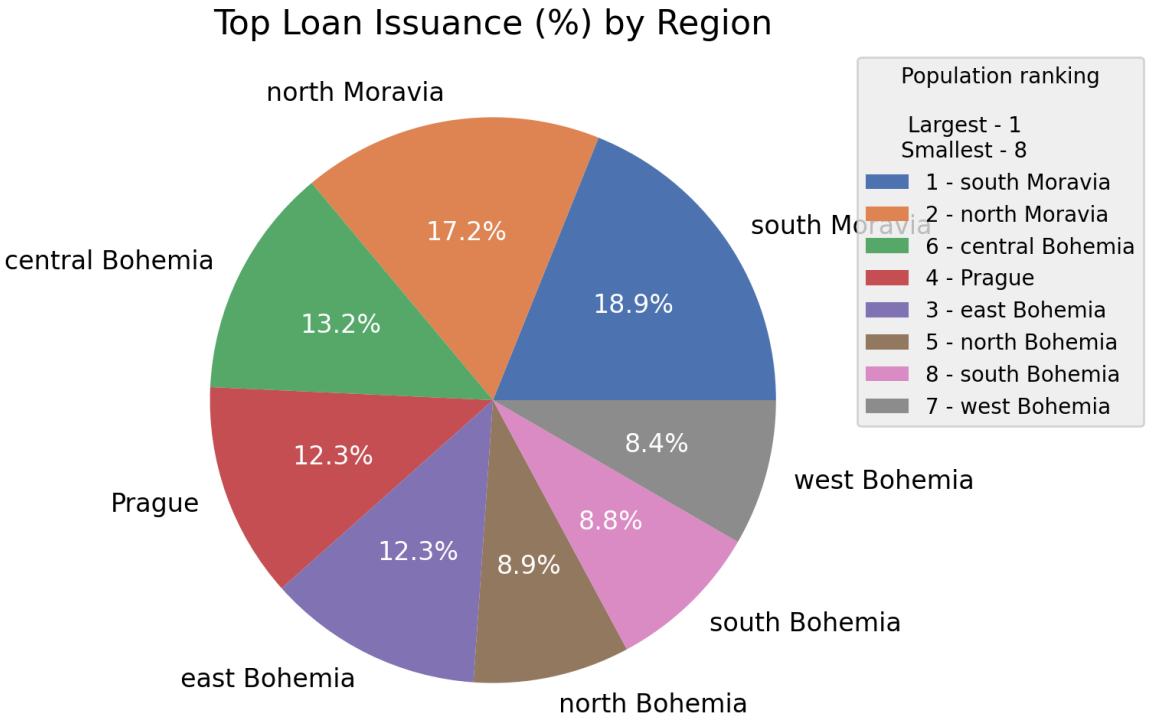
Table preview:

*	year	loan_type	Loans issued	moving_avg
1	1993-07-01	bad	1	3.0000
2	1993-07-01	good	2	3.0000
3	1993-08-01	good	1	2.0000
4	1993-09-01	bad	1	2.6667
5	1993-09-01	good	3	2.6667
6	1993-10-01	good	1	2.2500
7	1993-11-01	good	3	2.4000
8	1993-12-01	bad	2	3.3333
9	1993-12-01	good	6	3.3333
10	1994-01-01	good	6	3.7143

This confirms my hypothesis that “loan issuance follows an upward trend improve over long periods of time”.

Hypothesis 2 - “Geolocation and population affect loan issuance and performance”

Looking at the loan issuances by region, note that each region consist of smaller districts. There are a total of 8 regions, each pie slice shows the % of loans by the region. Compare this with the region’s population ranking (shown in the pie chart legend), 1 being the largest (most populous).



1. The regions with the most loan issuances tend to be highly ranked in population as well.
2. It is important to note that region “Prague” (ranked 4th in population) consists of only one district, this one district alone made up 12.3% of all the loans in the country.

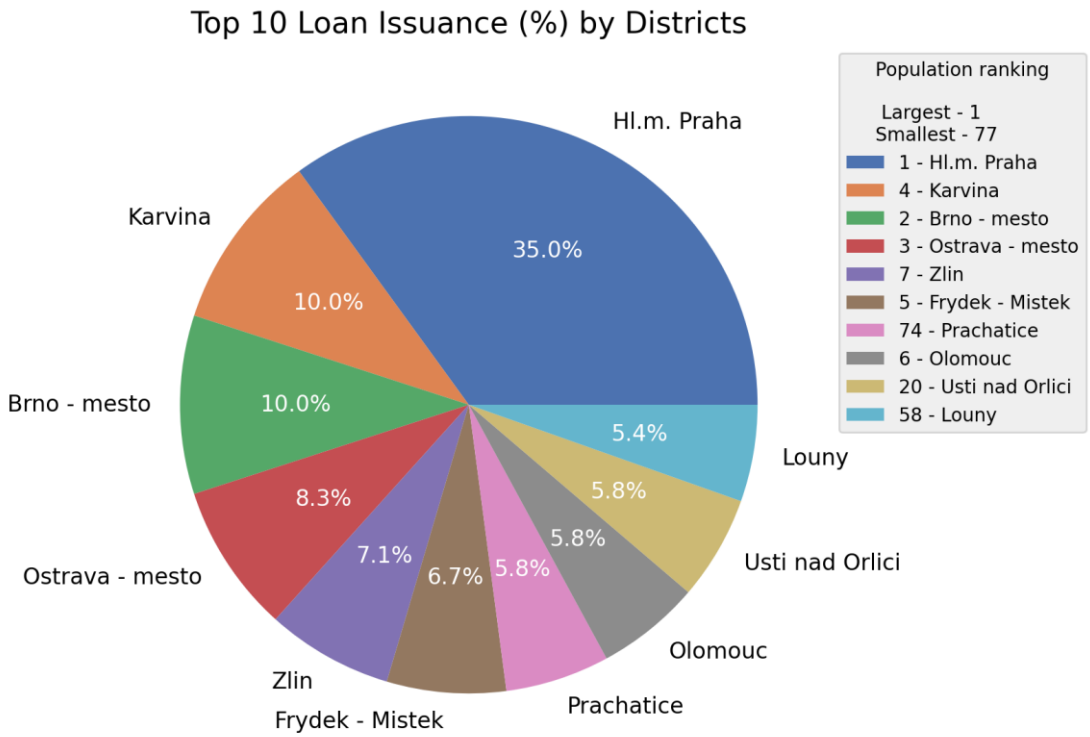
SQL snippet:

```
WITH population AS(
   SELECT
      region,
      DENSE_RANK() OVER(ORDER BY SUM(inhabitants) DESC) population_rank
   FROM district
   GROUP BY region
)
SELECT
   d.region,
   COUNT(loan_id) AS 'loans_per_region',
   population_rank
FROM district d, account a, loan l, population p
WHERE (d.district_id = a.district_id)
AND (l.account_id = a.account_id)
AND (p.region = d.region)
AND (loan_id IS NOT NULL)
GROUP BY region
ORDER BY loans_per_region DESC
```

Table preview:

*	region	loans_per_region	population_rank
1	south Moravia	129	1
2	north Moravia	117	2
3	central Bohemia	90	6
4	Prague	84	4
5	east Bohemia	84	3
6	north Bohemia	61	5
7	south Bohemia	60	8
8	west Bohemia	57	7

Looking at the breakdown of loan issuance by district. As there are too many districts (77) in total. We will focus on exploring the top 10 districts with the most loans issued over the full period of our data.



1. Most of the top 10 districts in terms of loan issuance (number of loans granted) are also the most populated districts (rank 1,2,3,4,5,6,7).
2. The 9 out of the top 10 districts also fall in the top 5 regions (except South Bohemia).
3. As mentioned previously, “Hl.m. Praha” district alone makes up the region Prague.
   It has the greatest number of loans issued. This correlates to its outsized number of inhabitants (11.69% of total population). The next largest district by population is only 3.76%.

Additional table to show % stats [1]

*	district_name	region	percent of total population	percent_total_loans
1	Hl.m. Praha	Prague	11.69	12.32
2	Brno - mesto	south Moravia	3.76	3.52
3	Ostrava - mesto	north Moravia	3.14	2.93
4	Karvina	north Moravia	2.77	3.52
5	Frydek - Mistek	north Moravia	2.22	2.35
6	Olomouc	north Moravia	2.19	2.05
7	Zlin	south Moravia	1.91	2.49
8	Opava	north Moravia	1.77	1.17
9	Ceske Budejovice	south Bohemia	1.72	1.17
10	Plzen - mesto	west Bohemia	1.65	0.88

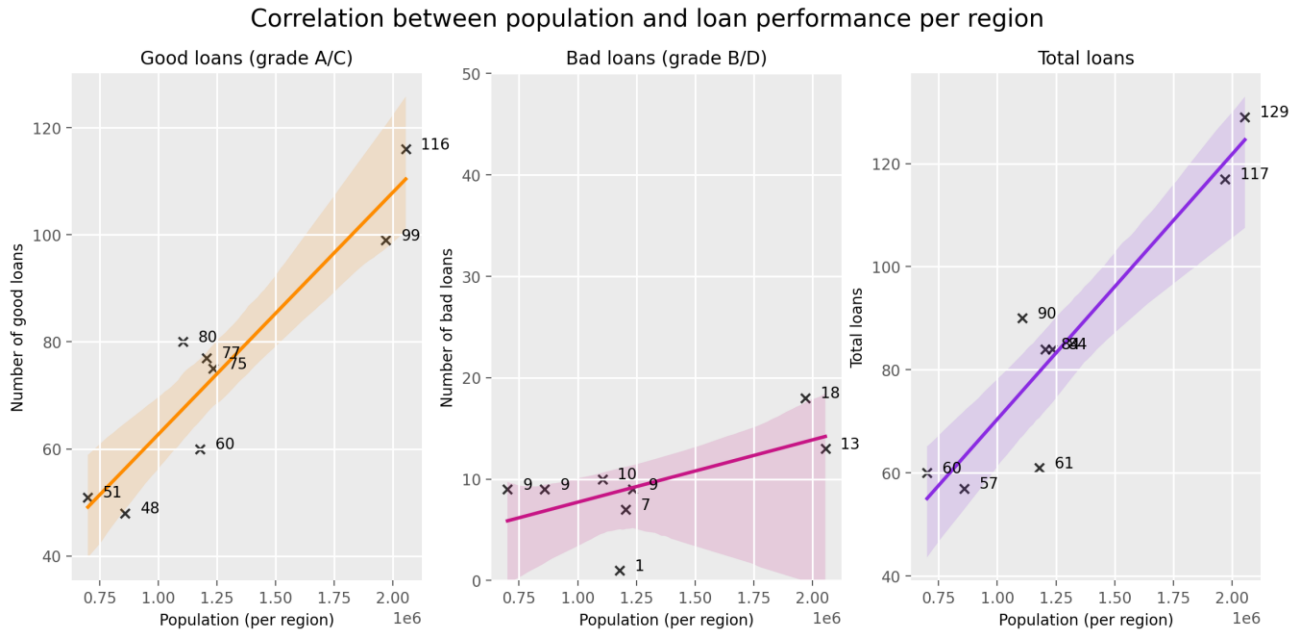
4. Top 10 districts out of 77 districts (12.98% of districts), make up 21.55% of all loans in the country.

SQL snippet:

```
WITH population AS(
   SELECT
      district_id,
      DENSE_RANK() OVER(ORDER BY inhabitants DESC) population_rank
   FROM district)
SELECT
   d.district_name,
   COUNT(loan_id) AS 'loans_per_district',
   population_rank
FROM district d, account a, loan l, population p
WHERE (d.district_id = a.district_id)
AND (l.account_id = a.account_id)
AND (p.district_id = d.district_id)
AND (loan_id IS NOT NULL)
GROUP BY d.district_id
ORDER BY loans_per_district DESC
LIMIT 10;
```

Table preview:

*	district_name	loans_per_district	population_rank
1	Hl.m. Praha	84	1
2	Karvina	24	4
3	Brno - mesto	24	2
4	Ostrava - mesto	20	3
5	Zlin	17	7
6	Frydek - Mistek	16	5
7	Prachatice	14	74
8	Olomouc	14	6
9	Usti nad Orlici	14	20
10	Louny	13	58



Next, we will look at the correlation between population and type of loans, note that

- the data was aggregated by region instead of district, as individual districts have small number of loans (< 5) and which can easily skew the results.
- population correlates better with good loans and total loans
- population has little correlation with bad loans

Thus, population is still more of an indicator for the overall loan issuance and good loans.
We can confirm part of hypothesis 2 - population and geolocation affect loan issuance. However its effect on loan performance is not conclusive.

SQL snippet:

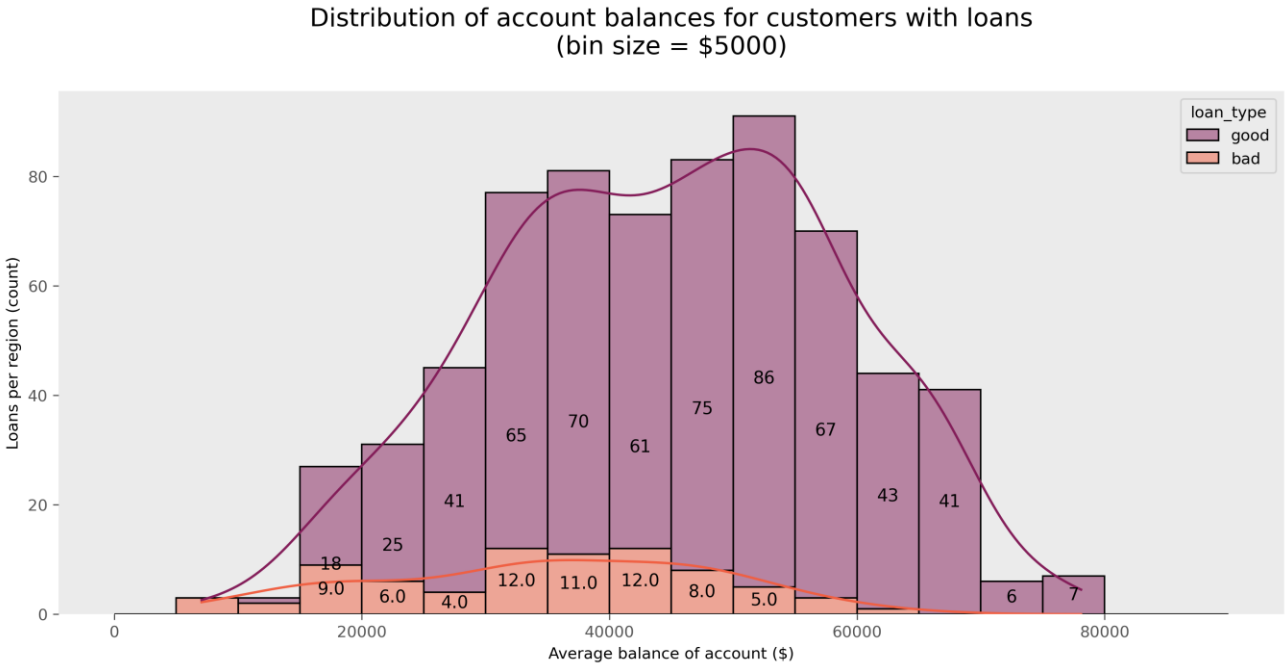
```
WITH bad_loans_table AS(
    SELECT
        region,
        COUNT(loan_id) AS 'bad_loans'
    FROM district d, account a, loan l
    WHERE (d.district_id = a.district_id)
    AND (l.account_id = a.account_id)
    AND (loan_id IS NOT NULL)
    AND (loan_type = "bad")
    GROUP BY region
),
good_loans_table AS(
    SELECT
        region,
        COUNT(loan_id) AS 'good_loans'
    FROM district d, account a, loan l
    WHERE (d.district_id = a.district_id)
    AND (l.account_id = a.account_id)
    AND (loan_id IS NOT NULL)
    AND (loan_type = "good")
    GROUP BY region
)
SELECT
    d.region,
    ROUND(SUM(inhabitants),0) AS population,
```

```
    bad_loans,
    good_loans,
    bad_loans+good_loans AS "total",
    good_loans/(bad_loans+good_loans)*100 AS "% good loans",
    bad_loans/(bad_loans+good_loans)*100 AS "% bad loans"
FROM district d
LEFT JOIN bad_loans_table bl ON (d.region = bl.region)
LEFT JOIN good_loans_table gl ON (d.region = gl.region)
GROUP BY region
ORDER BY population DESC;
```

Table preview:

*	region	population	bad_loans	good_loans	total	% good loans	% bad loans
1	south Moravia	2054989	13	116	129	89.9225	10.0775
2	north Moravia	1970302	18	99	117	84.6154	15.3846
3	east Bohemia	1234781	9	75	84	89.2857	10.7143
4	Prague	1204953	7	77	84	91.6667	8.3333
5	north Bohemia	1178977	1	60	61	98.3607	1.6393
6	central Bohemia	1105234	10	80	90	88.8889	11.1111
7	west Bohemia	859306	9	48	57	84.2105	15.7895
8	south Bohemia	700595	9	51	60	85.0000	15.0000

Hypothesis 3: “Customer transaction behaviour is an indicator for loan performance.”



Looking at accounts with loans and tracking the average balance within the whole time period.

- Accounts with bad loans have a positive skew distribution curve (lower average balances) , higher chance of bad loans when customer average balance < $65,000.
- Accounts with good loans have a more even distribution, with tendency of higher average balances.
- One possible deduction is that customers with low balances may run into cashflow problems and are unable to repay loans on time.

This confirms the hypothesis that customer transaction behaviour (average balance) is an indicator for loan performance. However it should not be used as a sole indicator, consider finding other useful features to do prediction, eg. loan value distribution vs loan performance

Note: average balance here counts the balance whenever a transaction occurs that changes the account balance, it is not aggregated by a fixed time period.

SQL snippet:

```
-- transaction data (average balance for accounts with loans)
SELECT t.account_id, ROUND(AVG(t.balance),2) AS `average_balance`, l.loan_type
FROM trans t, account a, loan l
WHERE (t.account_id = a.account_id)
AND (a.account_id = l.account_id)
AND (l.loan_id IS NOT NULL)
GROUP BY t.account_id
ORDER BY l.loan_type DESC;
```

Binning done in matplotlib for efficiency (can also be done in SQL, but the code will be horrendously long based on my bin size)

Table preview:

*	account_id	average_balance	loan_type
1	2	36313.03	good
2	38	33837.65	good
3	105	31439.76	good
4	176	51850.89	good
5	303	46997.22	good
6	330	40833.15	good
7	349	16349.23	good
8	544	52114.56	good
9	785	43422.77	good
10	813	35847.5	good

Conclusion and recommended actions:

1. Loan issuance has increased over time.
2. Good to bad loan ratio improved over time.
3. Population is a good indicator of loan issuance, but not loan performance.
4. Focus on issuing loans in area with high population, specifically the top 5 regions and top 10 districts.
5. When evaluating loan applications, take into account the customer's average account balance. Customers with lower loan balances (< $65,000) have a higher chance of default.

APPENDIX

DATA CLEANING:

1. To prepare the data for analysis, a few main cleaning steps were done.
2. Renaming columns
    a. Some tables were recreated as new tables, the original table was renamed in case needed for comparison.
    b. For table district, some columns A5-A8 were deemed not useful and hence not included.

```
-- district Table
ALTER TABLE district RENAME TO district_old;

CREATE TABLE district AS
SELECT
   A1 AS district_id,
   A2 AS district_name,
   A3 AS region,
   A4 AS inhabitants,
   A9 AS cities,
   A10 AS urban_inhabitant_ratio,
   A11 AS average_salary,
   A14 AS enterpreneurs_per_1000,
   A12 AS unemployment_1995,
   A13 AS unemployment_1996,
   A15 AS crimes_1995,
   A16 AS crimes_1996
FROM district_old;
```

3. Data transformation
    a. Client table requires data transformation to identify the gender and actual birthdates of clients.
    b. A new column Age was also created from birthdates.

```
-- client Table
```

```
ALTER TABLE client
ADD COLUMN gender VARCHAR(6),
ADD COLUMN date_of_birth TEXT,
ADD COLUMN year VARCHAR(4),
ADD COLUMN month VARCHAR(2),
ADD COLUMN day VARCHAR(2)
ADD COLUMN age INT;

UPDATE client
SET month = CAST(SUBSTRING(CAST(birth_number AS VARCHAR(6)), 3, 2) AS INT),
year = CONCAT('19', SUBSTRING(CAST(birth_number AS VARCHAR(6)), 1, 2)),
day = SUBSTRING(CAST(birth_number AS VARCHAR(6)), 5, 2),
gender = CASE
   WHEN month >= 50 THEN 'F'
   WHEN month < 50 THEN 'M'
   ELSE NULL
END,
month = CASE
   WHEN month >= 50 THEN month-50
   ELSE month
END,
month = LPAD(CAST(month AS VARCHAR(6)), 2, '0'),
date_of_birth = STR_TO_DATE(CONCAT(year, month, day), "%Y%m%d");

ALTER TABLE client
MODIFY COLUMN date_of_birth DATE;

-- Creating age column
UPDATE client
SET age = YEAR("1999-01-01") - YEAR(date_of_birth) -
     (CASE WHEN MONTH("1999-01-01") < MONTH(date_of_birth)
          OR (MONTH("1999-01-01") = MONTH(date_of_birth)
          AND DAY("1999-01-01") < DAY(date_of_birth)) THEN 1 ELSE 0 END);

-- Remove working columns
ALTER TABLE client
DROP COLUMN year,
DROP COLUMN month,
DROP COLUMN day,
DROP birth_number;
```

c. Loan table's loan status were summarized into good and bad loans. A `year_month` column was also created, it for a time series aggregation time by year_month.

```
-- loan Table
ALTER TABLE loan
MODIFY COLUMN `date` DATE,
ADD COLUMN loan_type VARCHAR(10);

-- Classify loans paid in full or wihout missed payments as good loans
-- loans not repaid or with missed payments as bad loans
-- create new year_month column for timeseries
UPDATE loan
SET loan_type = CASE
   WHEN status = 'A' OR status = 'C' THEN 'good'
   WHEN status = 'B' OR status = 'D' THEN 'bad'
   ELSE ''
   END,
SET `year_month` = CONCAT(YEAR(date), '-', MONTH(date), '-', '01');

ALTER TABLE loan
MODIFY COLUMN `year_month` DATE,
```

4. Null values
    a. Transaction table contains many NULL values
    b. After one round of renaming columns and inspecting the data, the following were concluded:
        - 'bank' column NULL values do not need to be filled since all transactions to another bank have bank given
        - all NULL values in 'operation' are interest

- all NULL values in 'type' column are withdrawals because their 'operation' column are all "withdrawals in cash"
- 'operation' column options to only contain (interest / cash / bank)
- 'type' column options to only contain (credit, withdrawal)
- 'transaction_type' column is likely not useful as the data is incomplete and cannot be inferred for filling

```
ALTER TABLE trans RENAME TO trans_old;

CREATE TABLE trans AS
SELECT * FROM trans_old;

ALTER TABLE trans
RENAME COLUMN k_symbol TO transaction_type;

ALTER TABLE trans
MODIFY COLUMN `date` DATE;

UPDATE trans
SET type = CASE
    WHEN type = 'PRIJEM' THEN 'credit'
    WHEN type = 'VYDAJ' THEN 'withdrawal'
    ELSE ''
END,
operation = CASE
    WHEN operation = "VYBER KARTOU" THEN "credit card withdrawal"
    WHEN operation = "VKLAD" THEN "credit in cash"
    WHEN operation = "PREVOD Z UCTU" THEN "collection from another bank"
    WHEN operation = "VYBER" THEN "withdrawal in cash"
    WHEN operation = "PREVOD NA UCET" THEN "remittance to another bank"
    ELSE ''
END,
transaction_type = CASE
    WHEN transaction_type = "POJISTNE" THEN "insurance payment"
    WHEN transaction_type = "SLUZBY" THEN "payment for statement"
    WHEN transaction_type = "UROK" THEN "interest credit"
    WHEN transaction_type = "SANKC. UROK" THEN "sanction interest if negative balance"
    WHEN transaction_type = "SIPO" THEN "household"
    WHEN transaction_type = "DUCHOD" THEN "old-age pension"
    WHEN transaction_type = "UVER" THEN "loan payment"
    ELSE ""
END;

-- Second round
UPDATE trans
SET type = CASE
    WHEN type = '' AND operation = 'withdrawal in cash' THEN 'withdrawal'
    ELSE type
END,
operation = CASE
    WHEN operation = "" AND transaction_type = 'interest credit' THEN 'interest'
    WHEN operation = 'withdrawal in cash' THEN 'cash'
```

```
    WHEN operation = 'credit in cash' THEN 'cash'
    WHEN operation = "collection from another bank" OR operation = "remittance to another bank" THEN 'bank'
    else operation
END;
```

5. Data integrity
    a. Primary and foreign key referencing was done, any conflicts would be surfaced by the RDBMS.

```
/*
KEY REFERENCING
*/
-- PRIMARY KEYS
ALTER TABLE account
ADD PRIMARY KEY (account_id);

ALTER TABLE loan
ADD PRIMARY KEY(loan_id);

ALTER TABLE orders
ADD PRIMARY KEY(order_id);

ALTER TABLE trans
ADD PRIMARY KEY(trans_id);

ALTER TABLE disposition
ADD PRIMARY KEY(disp_id);

ALTER TABLE `card`
ADD PRIMARY KEY(card_id);

ALTER TABLE district
ADD PRIMARY KEY(district_id);

ALTER TABLE client
ADD PRIMARY KEY(client_id);

-- FOREIGN KEYS
ALTER TABLE account
ADD FOREIGN KEY(district_id) REFERENCES district(district_id);

ALTER TABLE loan
ADD FOREIGN KEY(account_id) REFERENCES account(account_id);

ALTER TABLE orders
ADD FOREIGN KEY(account_id) REFERENCES account(account_id);

ALTER TABLE trans
ADD FOREIGN KEY(account_id) REFERENCES account(account_id);

ALTER TABLE disposition
```

```
ADD FOREIGN KEY(client_id) REFERENCES client(client_id),
ADD FOREIGN KEY(account_id) REFERENCES account(account_id);

ALTER TABLE `card`
ADD FOREIGN KEY(disp_id) REFERENCES disposition(disp_id);

ALTER TABLE client
ADD FOREIGN KEY(district_id) REFERENCES district(district_id);
```

b. In this case, the tables were already in 3NF, no conflicts were surface.
c. I also did manual SELECT statements with GROUP BY and DISTINCT to check for any duplicates of keys, just in case.

```
/*
DATA INTEGRITY
*/

-- Check if there are no duplicate of primary key
SELECT COUNT(account_id), COUNT(DISTINCT(account_id))
FROM account;
SELECT COUNT(disp_id), COUNT(DISTINCT(disp_id))
FROM disposition;
SELECT COUNT(order_id), COUNT(DISTINCT(order_id))
FROM `order`;
SELECT COUNT(trans_id), COUNT(DISTINCT(trans_id))
FROM trans;

-- All disp_id are unique to client_id and account_id
SELECT disp_id, COUNT(client_id), COUNT(account_id)
FROM disposition
GROUP BY disp_id
HAVING COUNT(client_id) > 1 OR COUNT(account_id) > 1;

-- All transactions belong to an account
SELECT a.account_id
FROM account a, trans t
WHERE (a.account_id = t.account_id)
AND t.trans_id IS NULL;

-- Every loan belong to exactly 1 account
SELECT a.account_id
FROM account a, loan l
WHERE (a.account_id = l.account_id)
AND l.loan_id IS NULL;
```

ADDITIONAL QUERY SQL CODES

Data 1 – District and region, against population % and loan %

```
-- District and region, against population% and loan%
WITH total_table AS (
   SELECT SUM(inhabitants) as total
   FROM district
),
```

```sql
loans AS (
    SELECT COUNT(*) AS total_loans
    FROM loan
),

loan_district AS (
    SELECT
    d.district_id,
    COUNT(loan_id) AS 'loans_per_district'
    FROM district d, account a, loan l
    WHERE (d.district_id = a.district_id)
    AND (l.account_id = a.account_id)
    AND (loan_id IS NOT NULL)
    GROUP BY d.district_id
)

SELECT
d.district_name,
region,
ROUND((inhabitants/t.total*100),2) AS 'percent of total population',
ROUND((loans_per_district/total_loans*100),2) AS 'percent_total_loans'
FROM district d
INNER JOIN loan_district l ON (d.district_id = l.district_id)
CROSS JOIN total_table t
CROSS JOIN loans
ORDER BY inhabitants/t.total*100 DESC
LIMIT 10;
```

*	district_name	region	percent of total population	percent_total_loans
1	Hl.m. Praha	Prague	11.69	12.32
2	Brno - mesto	south Moravia	3.76	3.52
3	Ostrava - mesto	north Moravia	3.14	2.93
4	Karvina	north Moravia	2.77	3.52
5	Frydek - Mistek	north Moravia	2.22	2.35
6	Olomouc	north Moravia	2.19	2.05
7	Zlin	south Moravia	1.91	2.49
8	Opava	north Moravia	1.77	1.17
9	Ceske Budejovice	south Bohemia	1.72	1.17
10	Plzen - mesto	west Bohemia	1.65	0.88

Additional visuals