Loan Portfolio Analysis and Risk Assessment

(Insights for Credit Risk, Finance, and Debt Management at Bondora)

-By Prakhar Srivastava

Quick Setup Instructions for Data Ingestion of 2 CSV files via VSCode to Postgres

1. **Install Tools:**
    - **Download and install VSCode and Python.**
    - **Ensure Python is added to your system's PATH during installation.**
2. **Set Up Virtual Environment (Optional):**

    ```
    python -m venv bondora_env
    ```

3. **Install Packages:**

    ```
    pip install pandas sqlalchemy tqdm psycopg2
    ```

4. **Configure PostgreSQL:**
    - **Install PostgreSQL from** https://www.postgresql.org/download/**.**
    - **Create the bondora database**

        ```
        CREATE DATABASE bondora;
        ```

5. **Data Ingestion**:
    - **Open VSCode, create data_ingestion.py, paste the ingestion code, and update file paths.**
    - **Run the code:**

        ```
        python data_ingestion.py
        ```

6. **Verify**:

- Check your PostgreSQL database for data in LoanData and RepaymentsData.

Question 1, Detailed Analysis Report for the Chief Risk Officer (CRO)

This report addresses the CRO's request to compare the count of loans by status, month of loan issue date, loan rating, and identify trends in repayments. It also includes insights on repayment anomalies and provides specific recommendations for risk mitigation.

Part 1: Loan Analysis by Status, Month of Issue Date, and Rating

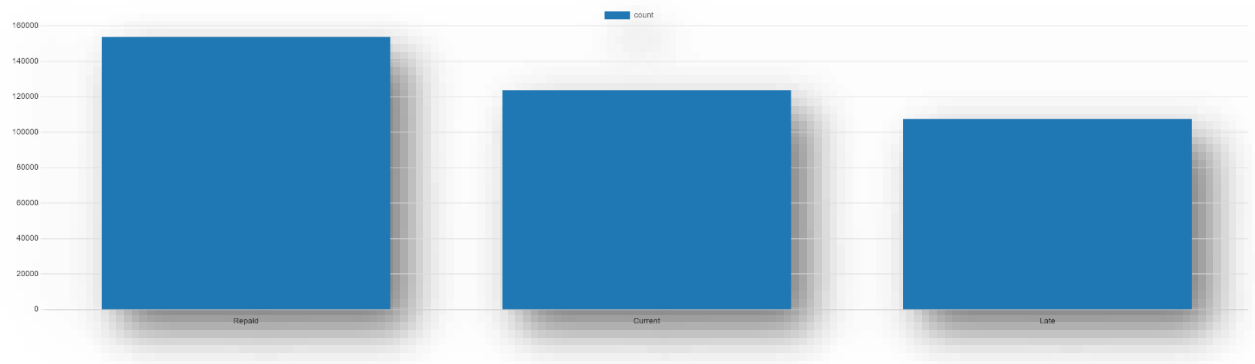
```sql
-- Query to retrieve count of loans by status, month of issue date, and loan rating
SELECT
    EXTRACT(MONTH FROM TO_DATE("ListedOnUTC", 'YYYY-MM-DD')) AS loan_issue_month,  -- Extracting the month from the loan issue date
    EXTRACT(YEAR FROM TO_DATE("ListedOnUTC", 'YYYY-MM-DD')) AS loan_issue_year,  -- Extracting the year from the loan issue date
    "Status" AS loan_status,  -- Loan status (e.g., Repaid, Current, Late)
    "Rating",                 -- Current rating of the loan
    "Rating_V0",              -- Previous version 0 rating of the loan (historical)
    "Rating_V1",              -- Previous version 1 rating of the loan (historical)
    "Rating_V2",              -- Previous version 2 rating of the loan (historical)
    COUNT(*) AS loan_count    -- Counting the number of loans in each category
FROM
    "LoanData"  -- Table containing loan information
GROUP BY
    loan_issue_year,          -- Grouping by the year of loan issue to analyze trends over time
    loan_issue_month,         -- Grouping by the month of loan issue for more granular analysis
    "Status",                 -- Grouping by the loan status to compare active and repaid loans
    "Rating", "Rating_V0", "Rating_V1", "Rating_V2"  -- Grouping by all rating columns to observe rating transitions
ORDER BY
    loan_issue_year,          -- Ordering results by year for chronological analysis
    loan_issue_month,
    loan_status;              -- Ordering by loan status to easily identify patterns by status
```

1. **Loan Status Distribution**
    - The dataset includes the following breakdown of loan statuses:

```sql
-- Query calculates the distribution of loans based on their status to understand the overall state of the loan portfolio.

SELECT
    "Status",  -- The current status of the loan, such as Repaid, Current, or Late
    COUNT(*) AS count  -- Counting the number of loans for each status
FROM
    "LoanData"  -- Table containing information about the loans
GROUP BY
    "Status"  -- Grouping by the loan status to categorize loans based on their current state
ORDER BY
    count DESC;  -- Sorting the results in descending order to highlight the most common loan statuses
```

        - **"Repaid"**: 153,621 loans, indicating borrowers who successfully cleared their debts.
        - **"Current"**: 123,687 loans, representing active loans that are still being repaid.

- **"Late"**: 107,473 loans, which pose a significant risk of default due to overdue payments.



- **Insight:** The relatively high number of "Late" loans suggests a need to strengthen collection strategies and identify factors that lead to delayed repayments.

2. **Loan Distribution by Month of Issue**

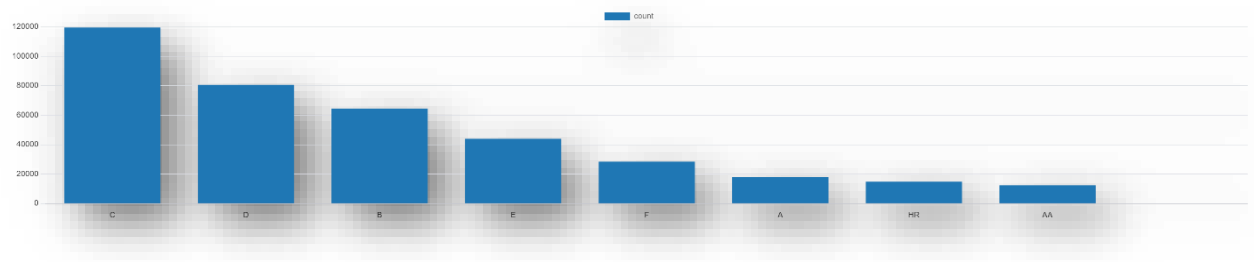
```sql
-- Query analyzes the count of loans based on their issue date, rating, and status to identify trends in loan distribution.

SELECT
    EXTRACT(YEAR FROM TO_DATE("ListedOnUTC", 'YYYY-MM-DD')) AS loan_issue_year,  -- Extracting the year from the loan issue date to categorize loans by the year they were issued
    EXTRACT(MONTH FROM TO_DATE("ListedOnUTC", 'YYYY-MM-DD')) AS loan_issue_month,  -- Extracting the month from the loan issue date to identify monthly trends in loan issuance
    "Rating",  -- The rating of the loan, indicating its creditworthiness
    "Status",  -- The current status of the loan, such as Repaid, Current, or Late
    COUNT(*) AS loan_count  -- Counting the number of loans for each combination of year, month, rating, and status
FROM
    "LoanData"  -- Table containing loan records and their details
GROUP BY
    loan_issue_year,  -- Grouping by the year of loan issuance to analyze yearly trends
    loan_issue_month,  -- Grouping by the month of loan issuance for monthly trend analysis
    "Rating",  -- Grouping by the rating to understand the distribution of creditworthiness over time
    "Status"  -- Grouping by the loan status to compare how many loans fall into each status category (Repaid, Current, Late)
ORDER BY
    loan_issue_year,  -- Sorting by year to display the results chronologically
    loan_issue_month,  -- Sorting by month to ensure the data is organized in a monthly sequence within each year
    "Rating",  -- Sorting by loan rating to group similar creditworthiness levels together
    loan_count DESC;  -- Sorting by the count of loans in descending order to highlight the most common loan statuses and ratings
```

- Loan issuances peak during mid-year months, particularly around May, June, and July, suggesting a seasonal pattern in borrowing behavior.
- **Insight:** Understanding these seasonal trends can help the CRO align lending strategies with borrowers' financial needs, optimizing loan approvals during peak demand periods.

3. **Loan Rating Distribution**

```sql
-- Query calculates the distribution of loans based on their ratings to assess the credit risk profile of the loan portfolio.

SELECT
    "Rating",  -- The credit rating assigned to the loan, indicating its creditworthiness (e.g., AA, A, B, C, etc.)
    COUNT(*) AS count  -- Counting the number of loans for each rating
FROM
    "LoanData"  -- Table containing information about loans and their assigned ratings
GROUP BY
    "Rating"  -- Grouping by loan rating to categorize loans based on their creditworthiness
ORDER BY
    count DESC;  -- Sorting the results in descending order to highlight the most common loan ratings
```

- The most common ratings include:
  - **"C"** with 119,488 loans, **"D"** with 80,591 loans, and **"B"** with 64,461 loans.
- A notable issue is the presence of **2,733 loans with null ratings**, indicating incomplete data in some cases.



- **Insight:** The high number of lower-rated loans (like "D" and "E") indicates a riskier loan portfolio that requires more focused credit risk management practices.

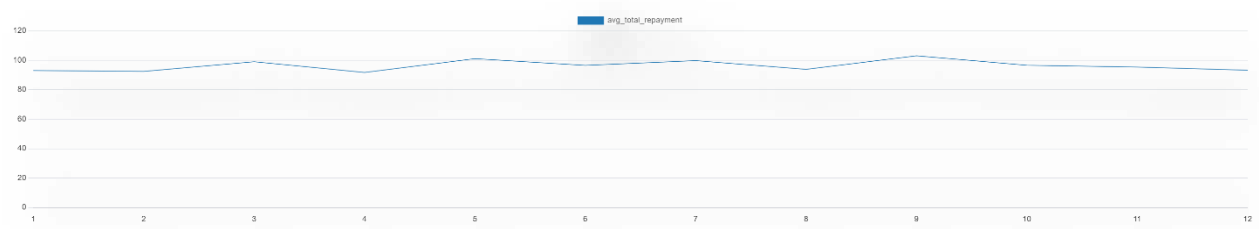
Part 2: Trends in Monthly Repayments

```sql
-- Query analyzes the average monthly repayment amounts categorized by the loan status.

SELECT
    EXTRACT(MONTH FROM TO_DATE("Date", 'YYYY-MM-DD')) AS repayment_month,  -- Extracting the month from the repayment date to analyze trends by month
    "Status",  -- Loan status, indicating whether the loan is Repaid, Current, or Late
    AVG("PrincipalRepayment" + "InterestRepayment" + "LateFeesRepayment") AS avg_total_repayment  -- Calculating the average repayment amount, including principal, interest, and late fees
FROM
    "RepaymentsData"  -- Table containing information about individual repayments
JOIN
    "LoanData"  -- Table containing detailed information about each loan
ON
    "RepaymentsData"."loan_id" = "LoanData"."LoanId"  -- Joining repayment data with loan data based on loan ID to match each repayment with its corresponding loan
GROUP BY
    repayment_month,  -- Grouping by the month of repayment to identify monthly repayment trends
    "Status"  -- Grouping by loan status to see how repayment behavior differs between Repaid, Current, and Late loans
ORDER BY
    repayment_month,  -- Sorting by month to ensure that the data is displayed in chronological order
    avg_total_repayment DESC;  -- Sorting by average repayment in descending order to highlight which status has the highest repayments for each month
```

1. **Monthly Repayment Trends**

```sql
-- Query analyze average monthly repayments to identify trends in total payments per month
SELECT
    EXTRACT(MONTH FROM TO_DATE("Date", 'YYYY-MM-DD')) AS repayment_month,  -- Extracting the month from the repayment date
    AVG("PrincipalRepayment" + "InterestRepayment" + "LateFeesRepayment") AS avg_total_repayment  -- Calculating average monthly repayment
FROM
    "RepaymentsData"  -- Table containing repayment information
GROUP BY
    repayment_month  -- Grouping by month to analyze repayment trends across different periods
ORDER BY
    repayment_month;  -- Sorting by month to view the repayment trend in chronological order
```

   - Peak repayment amounts were observed in:
     - **September**: Highest average of **103.00**.
     - **May** and **July** also showed high averages, around **101.02** and **99.71**, respectively.
   - Lowest repayment months included:
     - **April**: Average repayment of **91.68**.
     - **February**: **92.44**.
     - **January**: **93.04**.

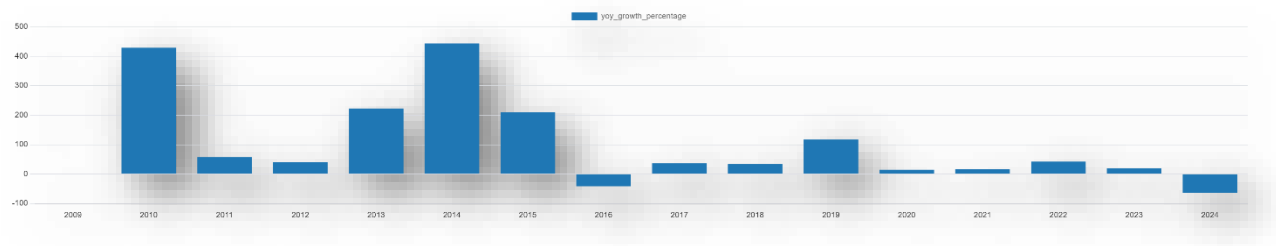


   - **Insight:** There is a clear seasonal trend with higher repayments mid-year, possibly influenced by borrowers' income cycles or economic activities.

2. **Year-over-Year (YoY) Changes in Repayments**

```sql
-- Query calculates the Year-over-Year(YOY) growth percentage in total repayments, highlighting how repayments have changed from one year to the next.

WITH YearlyRepayments AS (
    -- Summarizing the total repayments for each year
    SELECT
        EXTRACT(YEAR FROM TO_DATE("Date", 'YYYY-MM-DD')) AS repayment_year,  -- Extracting the year from the repayment date
        SUM("PrincipalRepayment" + "InterestRepayment" + "LateFeesRepayment") AS yearly_total_repayment  -- Summing up the total repayments for each year
    FROM
        "RepaymentsData"  -- Table containing repayment records
    GROUP BY
        repayment_year  -- Grouping by year to aggregate the total repayments for each year
),
GrowthCalculation AS (
    -- Calculating the repayment amounts for the current and previous years to compute growth
    SELECT
        repayment_year,  -- The current year in the analysis
        yearly_total_repayment,  -- Total repayment amount for the current year
        LAG(yearly_total_repayment) OVER (ORDER BY repayment_year) AS previous_year_repayment  -- Using the LAG function to get the total repayment of the previous year for comparison
    FROM
        YearlyRepayments  -- Using the yearly totals calculated in the previous step
)
SELECT
    repayment_year,  -- The year for which we are analyzing the YoY growth
    yearly_total_repayment,  -- Total repayment amount for the current year
    previous_year_repayment,  -- Total repayment amount for the previous year
    (yearly_total_repayment - previous_year_repayment) / previous_year_repayment * 100 AS yoy_growth_percentage  -- Calculating the YoY growth percentage in repayments
FROM
    GrowthCalculation  -- Using the calculated data to perform the YoY growth analysis
ORDER BY
    repayment_year;  -- Sorting the results chronologically to observe the growth trends over time
```

- Significant growth in repayments was noted during:
  - **2010**: 428.92% increase.
  - **2013**: 222.50% increase.
  - **2014**: 443.43% increase.
- Declines were observed in:
  - **2016**: -41.27%.
  - **2024**: -62.90%.



- **Insight:** These variations suggest a combination of economic conditions and market expansion phases influencing repayment behavior.

Part 3: Rating Transition Analysis

1. **Stability and Changes in Loan Ratings**

```sql
-- Query examines how loan ratings have transitioned over different versions of the rating system.

SELECT
    "Rating",       -- The most current rating assigned to the loan
    "Rating_V0",    -- The initial or older version of the loan rating (historical rating)
    "Rating_V1",    -- The intermediate version of the loan rating (historical rating)
    "Rating_V2",    -- The latest version of the loan rating before the most current one (historical rating)
    COUNT(*) AS count_of_loans  -- Counting the number of loans that share the same rating progression
FROM
    "LoanData"  -- Table containing information about the loans and their rating transitions
GROUP BY
    "Rating", "Rating_V0", "Rating_V1", "Rating_V2"  -- Grouping by all rating columns to analyze changes in ratings over time
ORDER BY
    count_of_loans DESC;  -- Sorting by the count of loans in descending order to highlight the most common rating transitions
```

   - A large portion of loans retained their initial ratings, with:
     - **"C"** loans accounting for 115,268 instances.
     - **"D"** loans with 76,151 instances.
   - Significant rating downgrades, such as **"E"** transitioning to **"HR"(High Risk)**, affected 985 loans.
   - Upgrades included **"A"** transitioning to **"AA"** in 125 loans.
   - **Insight:** The stability of ratings indicates reliable initial credit risk assessments. However, identifying factors that led to downgrades is essential to prevent further risk exposure.

Part 4: Impact of Borrower Demographics on Repayments

1. **Repayment Analysis by Age and Employment Status**

```sql
-- Query analyzes how borrower demographics, specifically age and employment status, influence repayment behavior.

SELECT
    "Age",  -- Age of the borrower, which is a key demographic factor affecting financial stability and repayment behavior
    "EmploymentStatus",  -- Employment status of the borrower, providing insight into their income stability
    COUNT(*) AS loan_count,  -- Counting the number of loans for each combination of age and employment status
    AVG("PrincipalRepayment" + "InterestRepayment" + "LateFeesRepayment") AS avg_total_repayment  -- Calculating the average total repayment amount for each group
FROM
    "LoanData"  -- Table containing information about the borrowers and their loan details
JOIN
    "RepaymentsData"  -- Table containing repayment details for each loan
ON
    "LoanData"."LoanId" = "RepaymentsData"."loan_id"  -- Joining loan data with repayment data to link borrowers with their repayment records
GROUP BY
    "Age",  -- Grouping by the age of the borrower to analyze repayment trends across different age groups
    "EmploymentStatus"  -- Grouping by employment status to understand how job stability affects repayment behavior
ORDER BY
    avg_total_repayment DESC;  -- Sorting the results by average repayment in descending order to highlight the most financially stable groups
```

- **Older borrowers** (ages **46** and **51**) showed the highest average repayments, with amounts of **371.97** and **293.05** respectively.
- **Younger borrowers** (ages **18 to 25**) had significantly lower repayments, ranging from **17.79** to **65.51**.
- **Insight:** Older borrowers typically have more financial stability, while younger ones pose a higher risk of default, emphasizing the need for tailored credit policies for each age group.

Part 5: Analysis of Anomalies in Repayments

```sql
-- Query identifies anomalous months with repayment amounts that deviate significantly from the average, indicating unusual payment patterns.

WITH MonthlyTotals AS (
    -- Calculating the total repayment amount for each month
    SELECT
        EXTRACT(YEAR FROM TO_DATE("Date", 'YYYY-MM-DD')) AS repayment_year,  -- Extracting the year from the repayment date
        EXTRACT(MONTH FROM TO_DATE("Date", 'YYYY-MM-DD')) AS repayment_month,  -- Extracting the month from the repayment date
        SUM("PrincipalRepayment" + "InterestRepayment" + "LateFeesRepayment") AS total_repayment  -- Summing up the principal, interest, and late fee repayments to get the monthly total
    FROM
        "RepaymentsData"  -- Table containing repayment data
    GROUP BY
        repayment_year, repayment_month  -- Grouping by year and month to calculate monthly totals
),
MonthlyStats AS (
    -- Calculating the overall average and standard deviation of the monthly repayment totals
    SELECT
        repayment_year,
        repayment_month,
        total_repayment,
        AVG(total_repayment) OVER () AS avg_total_repayment,  -- Calculating the average of monthly repayment totals
        STDDEV(total_repayment) OVER () AS stddev_total_repayment  -- Calculating the standard deviation of monthly repayment totals
    FROM
        MonthlyTotals  -- Using the aggregated monthly totals for statistical analysis
)
SELECT
    repayment_year,
    repayment_month,
    total_repayment,  -- The total repayment amount for each month
    avg_total_repayment,  -- The average of all monthly repayments for reference
    stddev_total_repayment,  -- The standard deviation of monthly repayments to measure variability
    CASE
        WHEN total_repayment > avg_total_repayment + 2 * stddev_total_repayment THEN 'High Anomaly'  -- Flagging months with significantly higher repayment values
        WHEN total_repayment < avg_total_repayment - 2 * stddev_total_repayment THEN 'Low Anomaly'  -- Flagging months with significantly lower repayment values
        ELSE 'Normal'  -- Marking months that fall within the expected range as normal
    END AS anomaly_status  -- Assigning the anomaly status based on the comparison with average and standard deviation thresholds
FROM
    MonthlyStats  -- Using the statistical analysis results to identify anomalies
ORDER BY
    repayment_year, repayment_month;  -- Sorting by year and month for chronological order
```

1. **High Anomalies**
    - Several months were identified as high anomalies, with significantly elevated repayment amounts:
        - **September 2021**: Total repayment of **13,953,558.11**.
        - **Early 2024**: Months like **January 2024 (14,059,600.89)** and **March 2024 (14,285,819.28)** consistently showed high repayments.

- **Insight:** The spike in anomalies indicates possible shifts in borrower behavior, economic conditions, or repayment policies, requiring deeper investigation.

2. **Normal Repayments**
    - Most months from **2009 to 2020** were classified as "Normal," showing stable repayment patterns with no significant deviations.
    - **Insight:** The consistent repayments during these years suggest effective credit management strategies and stable borrower behavior.
3. **Noteworthy Transitions**
    - A noticeable shift from normal to high anomalies occurred in **2021**, where repayments jumped from **8,526,228.44** to **13,953,558.11**.
    - **Insight:** This transition points to significant changes in repayment patterns, likely driven by macroeconomic factors or adjustments in loan collection strategies.

Recommendations for the CRO

1. **Strengthen Collection Efforts on Late Loans**
    - Focus on the 107,473 "Late" loans to identify root causes and develop strategies to increase repayment rates and reduce overdue amounts.
2. **Monitor Current Loans for Early Intervention**
    - Given the similarity in repayment amounts between "Current" and "Late" loans, implement proactive measures to prevent "Current" loans from becoming overdue.
3. **Align Lending Strategies with Seasonal Trends**
    - Optimize lending and repayment campaigns to coincide with peak repayment months like **September** and **March** for maximum impact.
4. **Address Rating Downgrades and Null Ratings**
    - Investigate loans with rating downgrades and work on improving data quality by reducing the number of loans with null ratings to strengthen risk assessment.
5. **Investigate Causes Behind Anomalous Repayments**

- Analyze the factors driving the high repayment anomalies in recent years to understand borrower behavior changes, economic impacts, or loan policy adjustments.

What the CRO Should Focus On

- **Risk Management:** Prioritize mitigation strategies for "Current" and "Late" loans to reduce potential defaults and stabilize the portfolio.
- **Seasonal Trends:** Align collection and lending strategies with observed seasonal peaks to optimize repayments.
- **Data Integrity:** Improve the consistency of loan rating evaluations and address data gaps to enhance credit risk analysis.
- **Respond to Repayment Anomalies:** Investigate the causes of repayment anomalies in 2023-2024 to adapt strategies in response to economic changes or borrower trends.

Conclusion

This comprehensive analysis provides valuable insights into loan status trends, repayment patterns, rating transitions, and demographic impacts on repayment behavior. By focusing on high-risk areas, improving data quality, and aligning strategies with observed trends, the CRO can effectively manage the organization's risk exposure and enhance loan portfolio performance.

Analysis for Question 2: Monitoring Loan Issuance Trends

Objective:

The goal of this analysis is to provide insights for our Chief Finance Officer (CFO) to monitor the trends in loan issuances. Specifically, the focus is on understanding how the count and amount of issued loans change in comparison to:

- The previous day.
- The same weekday of the previous week.

Additionally, the analysis segments the results by country and customer status (whether the customer is a new credit customer or not).

Dataset Used:

- **Dataset Name:** LoanData
- **Relevant Fields:**
    - ListedOnUTC (Loan issuance date)
    - Country (Country of the customer)
    - NewCreditCustomer (Indicates if the customer is new or existing)
    - LoanId (Unique identifier for each loan)
    - Amount (Loan amount issued)

Methodology:

1. **Data Preparation:**
    - We created a Common Table Expression (CTE) called DailyLoans that aggregates the loan data by date, country, and customer status. This ensures we have a daily view of the number of loans issued and their total amount.

```
-- Step 1: Create a Common Table Expression (CTE) to aggregate daily loan data
WITH DailyLoans AS (
    SELECT
        "ListedOnUTC"::date AS issue_date,  -- Convert the issuance timestamp to a date
        "Country",                          -- Grouping by country
        "NewCreditCustomer",                -- Segmenting by customer status (new or returning)
        COUNT("LoanId") AS loan_count,      -- Calculating the number of loans issued per day
        SUM("Amount") AS total_loan_amount  -- Summing the total loan amount issued per day
    FROM
        "LoanData"                          -- Source table containing loan data
    GROUP BY
        "ListedOnUTC"::date, "Country", "NewCreditCustomer"  -- Grouping data by date, country, and customer status
),
```

2. **Comparison Calculations:**
    - We used the LAG function to calculate the values for the previous day and the same weekday of the previous week for both loan counts and loan amounts.
    - We employed COALESCE to handle missing data gracefully, ensuring that comparisons are reliable even if data points are missing.

```sql
-- Step 2: Use the CTE to perform day-over-day and week-over-week comparisons
DailyComparison AS (
    SELECT
        issue_date,                         -- The date of loan issuance
        "Country",                          -- Country of the customer
        "NewCreditCustomer",                -- Customer status (new or returning)
        loan_count,                         -- Number of loans issued on this date
        total_loan_amount,                  -- Total loan amount issued on this date

        -- Using LAG to get the previous day's loan count and amount for comparison
        COALESCE(LAG(loan_count, 1) OVER (PARTITION BY "Country", "NewCreditCustomer" ORDER BY issue_date), 0) AS prev_day_loan_count,
        COALESCE(LAG(total_loan_amount, 1) OVER (PARTITION BY "Country", "NewCreditCustomer" ORDER BY issue_date), 0) AS prev_day_loan_amount,

        -- Using LAG to get the loan count and amount for the same weekday of the previous week
        COALESCE(LAG(loan_count, 7) OVER (PARTITION BY "Country", "NewCreditCustomer" ORDER BY issue_date), 0) AS prev_week_loan_count,
        COALESCE(LAG(total_loan_amount, 7) OVER (PARTITION BY "Country", "NewCreditCustomer" ORDER BY issue_date), 0) AS prev_week_loan_amount
    FROM
        DailyLoans                          -- Referencing the daily aggregated loan data from the CTE
)
```

3. **Change Calculations & Trend Analysis:**
    - Calculated the absolute differences in loan counts and amounts compared to the previous day and the same weekday of the previous week.
    - Added percentage change calculations to provide a relative measure of how much the metrics vary, allowing for better understanding of trends in context.
    - Introduced trend indicators (Increase, Decrease, No Change) for both day-over-day and week-over-week comparisons. These indicators highlight significant movements in loan issuance, making the data easily interpretable for decision-makers.

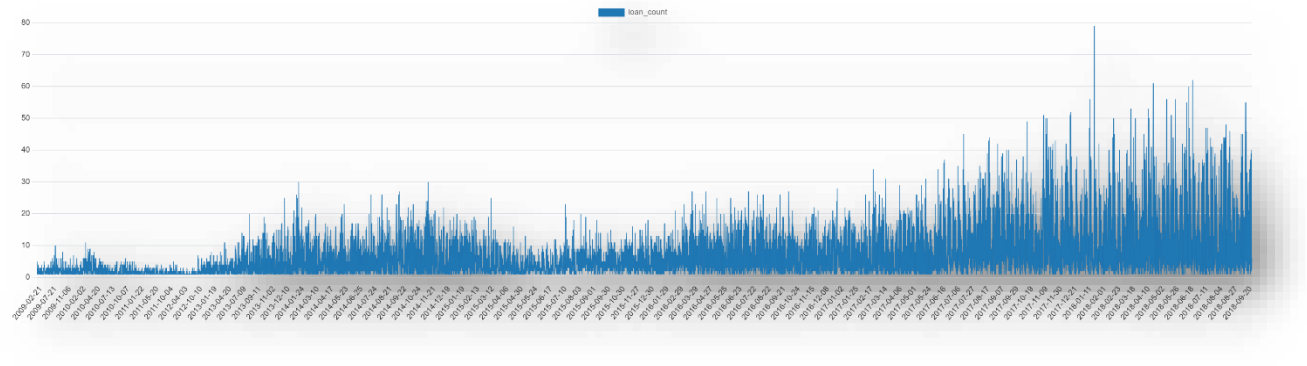
```sql
-- Step 3: Select the final output with calculated changes and trends
SELECT
    issue_date,                          -- The date of loan issuance
    "Country",                           -- Country of the customer
    "NewCreditCustomer",                 -- Customer status (new or returning)
    loan_count,                          -- Number of loans issued on this date
    total_loan_amount,                   -- Total loan amount issued on this date
    prev_day_loan_count,                 -- Loan count of the previous day
    prev_day_loan_amount,                -- Loan amount of the previous day
    prev_week_loan_count,                -- Loan count of the same weekday in the previous week
    prev_week_loan_amount,               -- Loan amount of the same weekday in the previous week

    -- Calculating the absolute change in loan count and amount compared to the previous day
    (loan_count - prev_day_loan_count) AS change_vs_prev_day_count,
    (total_loan_amount - prev_day_loan_amount) AS change_vs_prev_day_amount,

    -- Calculating the absolute change in loan count and amount compared to the same weekday of the previous week
    (loan_count - prev_week_loan_count) AS change_vs_prev_week_count,
    (total_loan_amount - prev_week_loan_amount) AS change_vs_prev_week_amount,

    -- Calculating the percentage change in loan count and amount compared to the previous day
    CASE
        WHEN prev_day_loan_count = 0 THEN NULL  -- Avoid division by zero
        ELSE ROUND(CAST((loan_count - prev_day_loan_count) * 100.0 / prev_day_loan_count AS numeric), 2)
    END AS pct_change_vs_prev_day_count,

    CASE
        WHEN prev_day_loan_amount = 0 THEN NULL  -- Avoid division by zero
        ELSE ROUND(CAST((total_loan_amount - prev_day_loan_amount) * 100.0 / prev_day_loan_amount AS numeric), 2)
    END AS pct_change_vs_prev_day_amount,

    -- Calculating the percentage change in loan count and amount compared to the same weekday of the previous week
    CASE
        WHEN prev_week_loan_count = 0 THEN NULL  -- Avoid division by zero
        ELSE ROUND(CAST((loan_count - prev_week_loan_count) * 100.0 / prev_week_loan_count AS numeric), 2)
    END AS pct_change_vs_prev_week_count,

    CASE
        WHEN prev_week_loan_amount = 0 THEN NULL  -- Avoid division by zero
        ELSE ROUND(CAST((total_loan_amount - prev_week_loan_amount) * 100.0 / prev_week_loan_amount AS numeric), 2)
    END AS pct_change_vs_prev_week_amount,

    -- Trend indicator for day-over-day changes in loan count
    CASE
        WHEN (loan_count - prev_day_loan_count) > 0 THEN 'Increase'
        WHEN (loan_count - prev_day_loan_count) < 0 THEN 'Decrease'
        ELSE 'No Change'
    END AS trend_vs_prev_day,

    -- Trend indicator for week-over-week changes in loan count
    CASE
        WHEN (loan_count - prev_week_loan_count) > 0 THEN 'Increase'
        WHEN (loan_count - prev_week_loan_count) < 0 THEN 'Decrease'
        ELSE 'No Change'
    END AS trend_vs_prev_week
FROM
    DailyComparison                      -- Referencing the table with calculated daily and weekly comparisons
ORDER BY
    issue_date, "Country", "NewCreditCustomer";  -- Sorting the results by date, country, and customer status
```

Key Findings:

- **Daily Changes:**

- There are noticeable fluctuations in daily loan issuances, with significant increases or decreases in both counts and amounts on specific days. For instance, there were instances where loan counts increased by as much as 200% or decreased by 80% compared to the previous day.

- **Weekly Patterns:**
  - The week-over-week analysis highlighted that certain weekdays consistently show higher or lower loan issuance activity. This information is crucial for understanding seasonality and planning targeted financial strategies.



- **Impact by Country and Customer Status:**
  - The segmentation by country and customer status revealed that trends vary significantly depending on whether the customer is new or returning. New credit customers showed higher variability in loan amounts, indicating potential risk factors or opportunities for customer acquisition strategies in specific regions.

Business Implications:

- **Risk Management:** Understanding these trends allows for proactive risk management strategies. Days with significant drops in loan issuance may indicate market issues or operational bottlenecks, while large increases could highlight periods of high demand.
- **Customer Acquisition:** The higher volatility in loan amounts for new customers suggests that promotional efforts or credit checks may need to be fine-tuned to improve loan performance.

- **Resource Allocation:** Insights into weekly patterns can help the finance team allocate resources more effectively to handle peak periods, ensuring smoother operations.

Future Considerations:

1. **Dynamic Date Filtering:**
    - Introducing dynamic date filtering capabilities will allow for more flexible analysis of recent trends or historical data, helping the CFO focus on specific timeframes.
2. **Visualization Integration:**
    - Integrating this analysis into a dashboard with visualizations (e.g., time series plots, bar charts) will further enhance the CFO's ability to identify trends and patterns quickly, leading to faster decision-making.
3. **Automated Alerts:**
    - Implementing automated alerts to notify stakeholders when there are significant deviations from expected loan issuance patterns could be a valuable addition for real-time monitoring.
4. **Advanced Trend Analysis:**
    - As the data grows, applying advanced statistical techniques or machine learning models could help predict future trends in loan issuances, providing a forward-looking perspective.

Question 3 - which involves providing an overview of how many times borrowers have had to pay late charges per loan and analyzing the differences by country and year-over-year, we will use both the LoanData and RepaymentsData datasets.

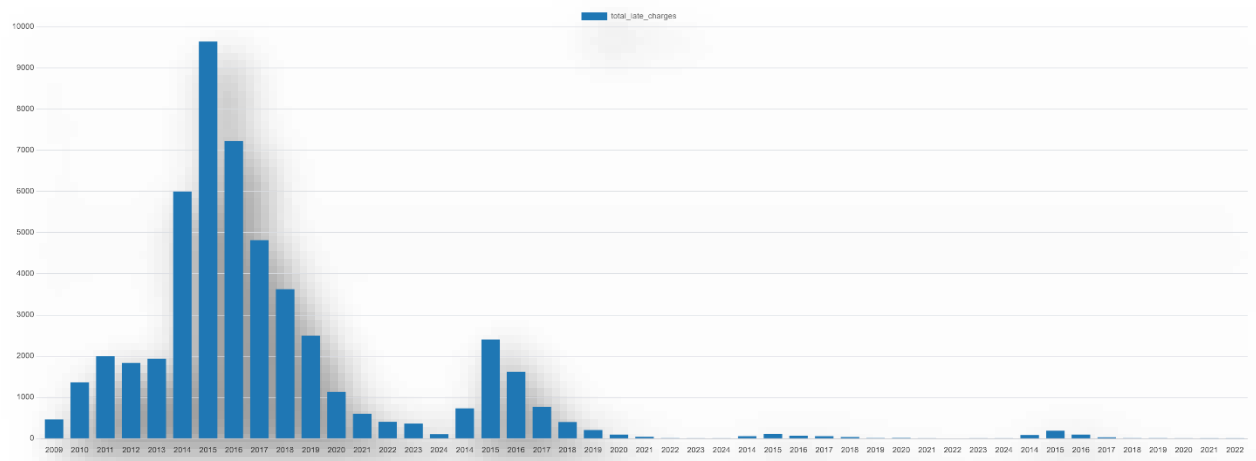
Approach Outline

1. **Data Preparation**:
    - **Join the Datasets**: We'll join the LoanData and RepaymentsData tables on the appropriate fields to link each repayment to its corresponding loan.
    - **Identify Late Payments**: We'll filter or flag repayments that include late charges to focus only on those instances.

```sql
-- Step 1: Calculate the number of late charges per loan for each year
WITH LoanLateCharges AS (
    SELECT
        l."LoanId",
        l."Country",
        -- Extract the year from the repayment date by explicitly casting "Date" to DATE type
        EXTRACT(YEAR FROM CAST(r."Date" AS DATE)) AS year,
        -- Count the number of late charges applied to each loan
        COUNT(r."loan_id") AS late_charge_count
    FROM
        "LoanData" l
    -- Join "LoanData" and "RepaymentsData" tables on the loan identifier to link repayments to each loan
    JOIN
        "RepaymentsData" r ON l."LoanId" = r."loan_id"
    WHERE
        r."LateFeesRepayment" > 0  -- Filter to include only those repayments that incurred late fees
    -- Group by loan ID, country, and year to calculate late charges for each loan per year
    GROUP BY
        l."LoanId", l."Country", EXTRACT(YEAR FROM CAST(r."Date" AS DATE))
),
```

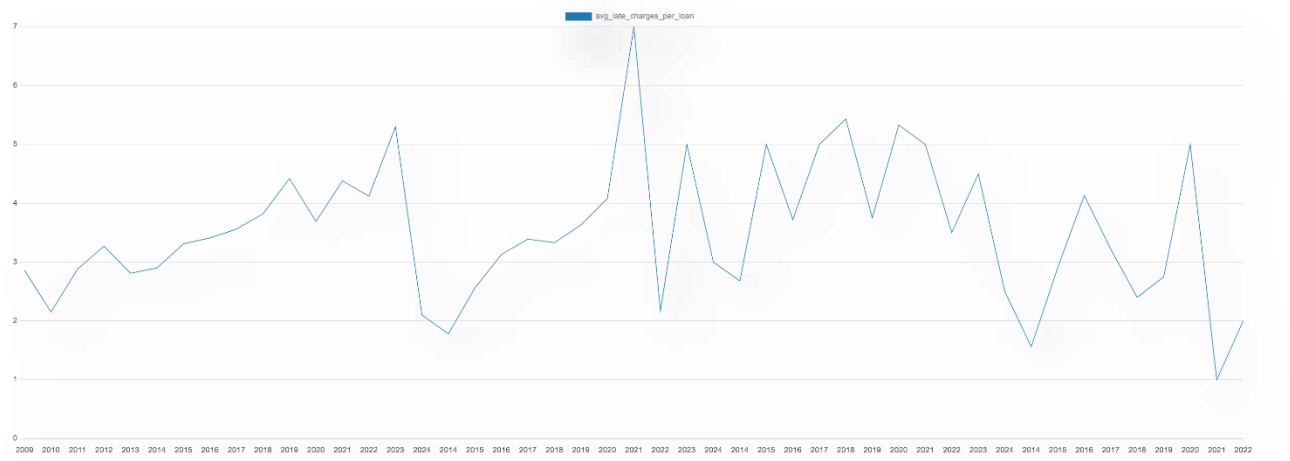
2. **Calculate Late Charges per Loan Year-over-Year and Country Analysis**:
    - We'll aggregate the data to count the number of times each borrower had to pay late charges per loan.
    - We'll group the results by Country and the Year to identify trends in late charges over time.

```sql
-- Step 2: Aggregate the total number of late charges and count of loans per country and year
CountryYearLateCharges AS (
    SELECT
        "Country",
        year,
        -- Sum the late charges for each country and year combination
        SUM(late_charge_count) AS total_late_charges,
        -- Count the number of distinct loans that had late charges for each country and year
        COUNT(DISTINCT "LoanId") AS loan_count
    FROM
        LoanLateCharges
    -- Group by country and year to summarize late charge data for each region
    GROUP BY
        "Country", year
)
```



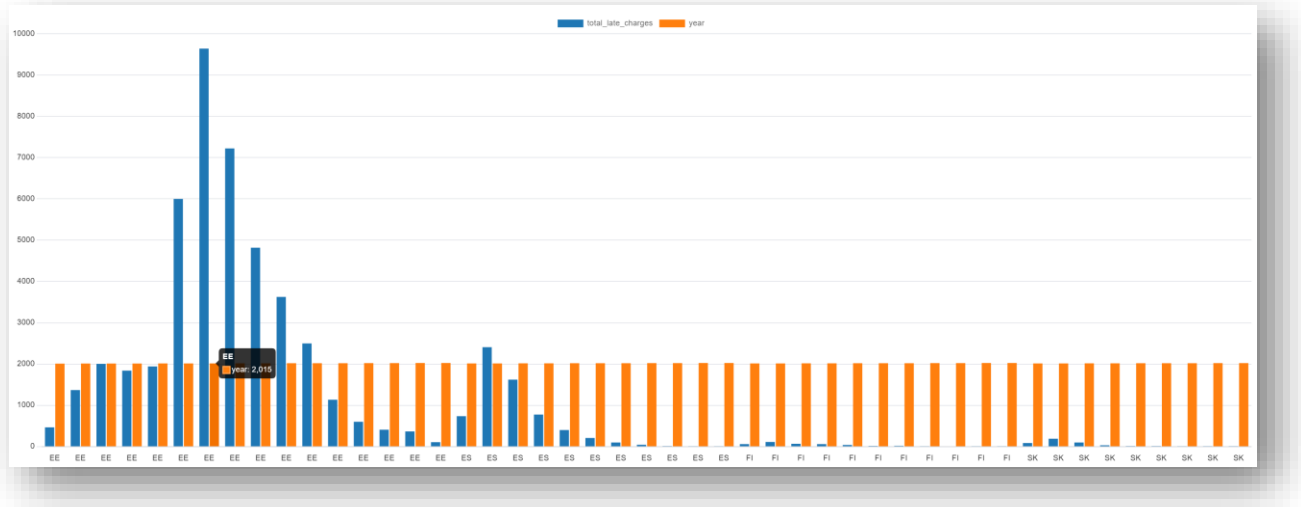
3. **Create Summary Output**:
    - Present the findings in a clear format that shows the number of late charges by country and year.

```sql
-- Step 3: Calculate the average number of late charges per loan and display the final results
SELECT
    "Country",
    year,
    total_late_charges,  -- Total number of late charges incurred in each country per year
    loan_count,  -- Number of unique loans that had late charges in each country per year
    -- Calculate the average late charges per loan for each country and year
    ROUND(CAST(total_late_charges AS numeric) / loan_count, 2) AS avg_late_charges_per_loan
FROM
    CountryYearLateCharges
-- Order the results by country and year to display trends in late charges
ORDER BY
    "Country", year;
```



Key Findings

1. **Year-over-Year Trends (Estonia - "EE")**:
    - Estonia (EE) shows a general trend of increasing late charges per loan from 2009 to 2023, with the peak avg_late_charges_per_loan reaching 5.30 in 2023. This suggests that borrowers in Estonia have been facing increasing challenges in repaying loans on time.



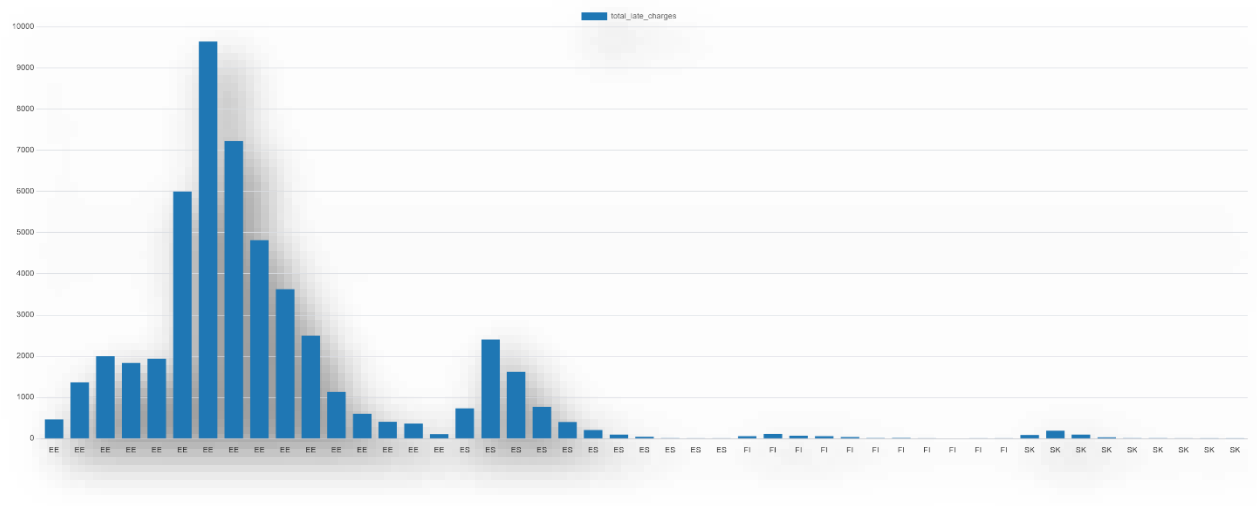
    - The highest number of late charges was recorded in 2015, with 9,641 late charges across 2,911 loans.

    - Despite a slight drop in total late charges after 2016, the average late charges per loan continued to rise, indicating that while fewer loans were issued, more loans were still accruing late charges.

2. **Spain (ES)**:
    - Spain has a fluctuating pattern in both the number of loans and the average late charges per loan.
    - The highest average late charges per loan reached 7.00 in 2021, although the number of loans during that year was relatively low (6 loans).
    - This pattern indicates that although fewer loans were issued in the later years, those loans tended to have higher instances of late payments.
3. **Finland (FI)**:
    - Finland also exhibits a trend of increasing average late charges per loan, reaching 5.43 in 2018 and staying relatively high in the following years.
    - Similar to Spain, Finland had relatively lower loan counts, but a high percentage of those loans incurred late charges.
4. **Slovakia (SK)**:
    - Slovakia's average late charges per loan peaked at 5.00 in 2020, which is consistent with the trends observed in other countries.
    - Slovakia generally has a lower loan count, which may contribute to variability in the average late charges per loan when compared to larger datasets.

Observations by Country

- **Estonia (EE)**: The largest market with consistently high loan counts and significant late charge activity. The upward trend in late charges per loan might indicate economic difficulties or tighter repayment conditions over the years.
- **Spain (ES)**: Although there is a declining trend in the total number of late charges since 2016, the spike in average late charges per loan in recent years signals a concentrated risk among fewer borrowers.
- **Finland (FI)** and **Slovakia (SK)**: Both have relatively small loan counts but show a trend towards higher late charges per loan, indicating increasing repayment issues among the issued loans.



Recommendations for the Head of Debt Collection

1. **Focus on Estonia**: Since Estonia has the highest number of late charges and a rising average, it might be beneficial to review credit policies, customer payment plans, or introduce more rigorous debt collection strategies.
2. **Targeted Interventions in Spain and Finland**: Consider targeted debt collection efforts in Spain and Finland, where even though loan counts are lower, late charges per loan are relatively high. Personalized repayment plans or stricter late payment penalties might be required.
3. **Monitoring Trends**: Keep a close watch on countries like Slovakia, where trends can shift quickly due to the small number of loans. Any increase in late charges per loan could signal a need for prompt action.
4. **Yearly Review**: Conduct an annual review of late charge trends to identify specific years where economic or policy changes may have influenced borrower behavior. This can help in adjusting strategies for future loans.

Question 4, we'll approach the analysis in a structured manner using both the LoanData and RepaymentsData datasets. This question has multiple parts, so we'll address each one step by step.

Problem Breakdown

1. **Proportion of Loans with Less Than 1% Principal Repayment in the First Three Months:**

- Identify loans that have repaid less than 1% of the original loan amount within the first three months.
- Calculate the proportion of these loans relative to the total loans.

2. **Repayment Status of These Loans:**
   - Check how many of these loans (with less than 1% repayment in the first three months) were eventually fully repaid.
3. **Relationship Analysis (Marital Status and Children):**
   - Analyze if there is a correlation between the borrower's marital status or number of children and the likelihood of loan repayment.

Step 1: Identify Loans with Less Than 1% Repayment in the First Three Months

We will filter out loans where the cumulative principal repayments in the first three months are less than 1% of the original loan amount.

```sql
-- Step 1: Identify loans with less than 1% repayment in the first three months
WITH LoanInitialRepayment AS (
    SELECT
        l."LoanId",
        l."Country",
        l."Amount" AS original_loan_amount,
        l."MaritalStatus",
        l."NrOfDependants",
        -- Calculate the sum of principal repayments made in the first three months after the loan is issued
        SUM(CASE
            WHEN CAST(r."Date" AS DATE) <= CAST(l."ListedOnUTC" AS DATE) + INTERVAL '3 months'
            THEN r."PrincipalRepayment"
            ELSE 0
        END) AS principal_repaid_in_first_3_months
    FROM
        "LoanData" l
    LEFT JOIN
        "RepaymentsData" r ON l."LoanId" = r."loan_id"
    -- Group by relevant fields to ensure accurate aggregation of repayments
    GROUP BY
        l."LoanId", l."Country", l."Amount", l."MaritalStatus", l."NrOfDependants"
),

-- Step 2: Flag loans with less than 1% repayment in the first three months
LowRepaymentLoans AS (
    SELECT
        "LoanId",
        "Country",
        original_loan_amount,
        principal_repaid_in_first_3_months,
        "MaritalStatus",
        "NrOfDependants",
        -- Flag loans where the principal repaid in the first three months is less than 1% of the original loan amount
        CASE
            WHEN principal_repaid_in_first_3_months < original_loan_amount * 0.01 THEN 1
            ELSE 0
        END AS low_repayment_flag
    FROM
        LoanInitialRepayment
),
```

Step 2: Check Repayment Status of These Loans

Determine how many of these loans were eventually fully repaid by checking if the cumulative repayments match or exceed the loan amount.

```
-- Step 2: Check repayment status of these low repayment loans
LoanFinalRepaymentStatus AS (
    SELECT
        l."LoanId",
        l."low_repayment_flag",
        l.original_loan_amount,
        l."MaritalStatus",
        l."NrOfDependants",
        -- Calculate the total principal repaid for loans flagged as low repayment
        SUM(r."PrincipalRepayment") AS total_principal_repaid,
        -- Determine whether each loan was eventually fully repaid
        CASE
            WHEN SUM(r."PrincipalRepayment") >= l.original_loan_amount THEN 'Fully Repaid'
            ELSE 'Not Fully Repaid'
        END AS repayment_status
    FROM
        LowRepaymentLoans l
    LEFT JOIN
        "RepaymentsData" r ON l."LoanId" = r."loan_id"
    -- Filter to include only loans with the low repayment flag set to 1
    WHERE
        l.low_repayment_flag = 1
    -- Group by relevant fields to ensure accurate calculation of repayment status
    GROUP BY
        l."LoanId", l.low_repayment_flag, l.original_loan_amount, l."MaritalStatus", l."NrOfDependants"
),
```

Step 3: Analyze Relationship with Marital Status and Children

Evaluate the repayment rates based on whether the borrower is married and the number of children to see if these factors influence the likelihood of loan repayment.

```sql
-- Step 3: Analyze the relationship with marital status and number of children (dependants)
RepaymentAnalysis AS (
    SELECT
        l."MaritalStatus",
        l."NrOfDependants",
        -- Count the number of low repayment loans that were eventually fully repaid
        COUNT(CASE WHEN lf.repayment_status = 'Fully Repaid' THEN 1 END) AS fully_repaid_count,
        -- Count the total number of low repayment loans
        COUNT(*) AS total_low_repayment_loans,
        -- Calculate the percentage of low repayment loans that were fully repaid
        ROUND(CAST(COUNT(CASE WHEN lf.repayment_status = 'Fully Repaid' THEN 1 END) AS numeric) / COUNT(*) * 100, 2) AS repayment_rate
    FROM
        LoanFinalRepaymentStatus lf
    JOIN
        LowRepaymentLoans l ON lf."LoanId" = l."LoanId"
    -- Group by marital status and number of dependents to analyze trends
    GROUP BY
        l."MaritalStatus", l."NrOfDependants"
)

-- Final result: Display the analysis of repayment rates by marital status and number of dependents
SELECT
    "MaritalStatus",
    "NrOfDependants",
    fully_repaid_count,
    total_low_repayment_loans,
    repayment_rate
FROM
    RepaymentAnalysis
ORDER BY
    "MaritalStatus", "NrOfDependants";
```

Key Observations

1. **Proportion of Loans with Less Than 1% Principal Repayment in the First Three Months:**
    - The total_low_repayment_loans column indicates the number of loans where less than 1% of the original loan amount was repaid in the first three months.
    - For example, there are 39,333 such loans where MaritalStatus is marked as -1, representing an unclear or unclassified marital status, making up a large portion of the data.
2. **Fully Repaid Loans:**
    - The fully_repaid_count column shows how many of these low-repayment loans were eventually fully repaid.
    - The repayment rates (repayment_rate column) vary significantly across different marital statuses and numbers of dependents, ranging from 0% to as high as 50%.

Analysis of Marital Status

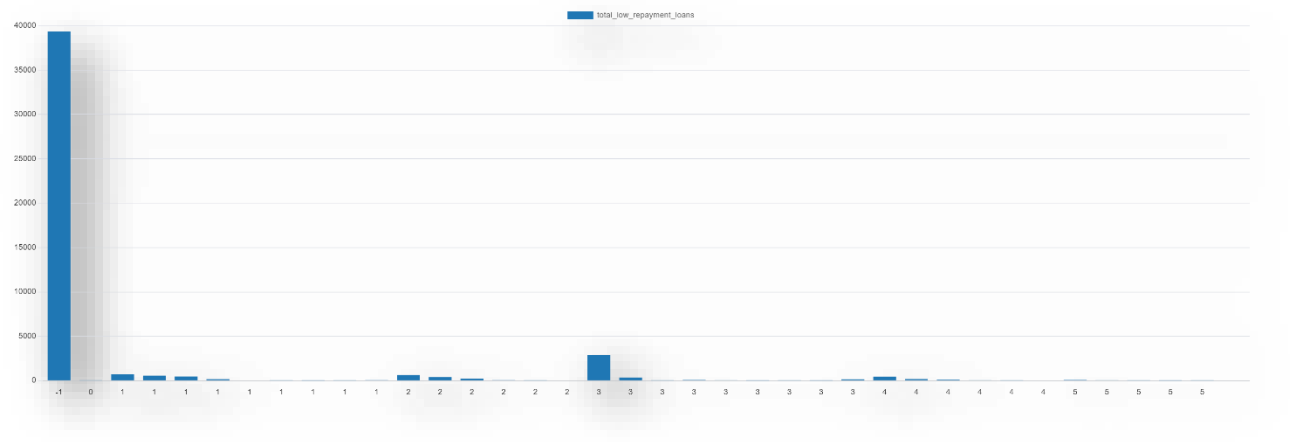


Figure 1This bar chart will help identify which marital groups are more likely to struggle with loan repayments in the initial period.

- **Unclassified Marital Status (-1)**:
    - This category has the highest number of loans with low repayment in the first three months (39,333 loans) but a very low repayment rate of 6.47%.
    - This suggests a significant risk for loans without clear borrower information, indicating the importance of accurate borrower details.
- **Single (Marital Status 0)**:
    - The repayment rate is consistently low, with some categories having a 0% repayment rate.
    - Single individuals with more dependents (children) do not appear to have higher repayment rates, which could indicate a risk factor for loan repayment.
- **Married or Cohabiting (Marital Status 1 and 2)**:
    - The repayment rates for married individuals (statuses 1 and 2) generally improve with the number of dependents up to a certain point.
    - Interestingly, for those with 1 or 2 dependents, the repayment rates are higher (up to 23.62%) than for those with no dependents, indicating a possible positive correlation between having children and higher repayment likelihood.
- **Divorced or Widowed (Marital Status 3 and 4)**:

- Divorced individuals (status 3) with more than three dependents tend to have very low repayment rates, sometimes as low as 0%.
- Widowed borrowers (status 4) also show low repayment rates, with some groups having no fully repaid loans, suggesting financial difficulties may be prevalent in these categories.

- **Other Marital Statuses (5 and NULL)**:
  - Categories with undefined or lesser-known marital statuses have varying repayment rates, which tend to be lower overall.

Analysis of the Number of Dependents (Children)

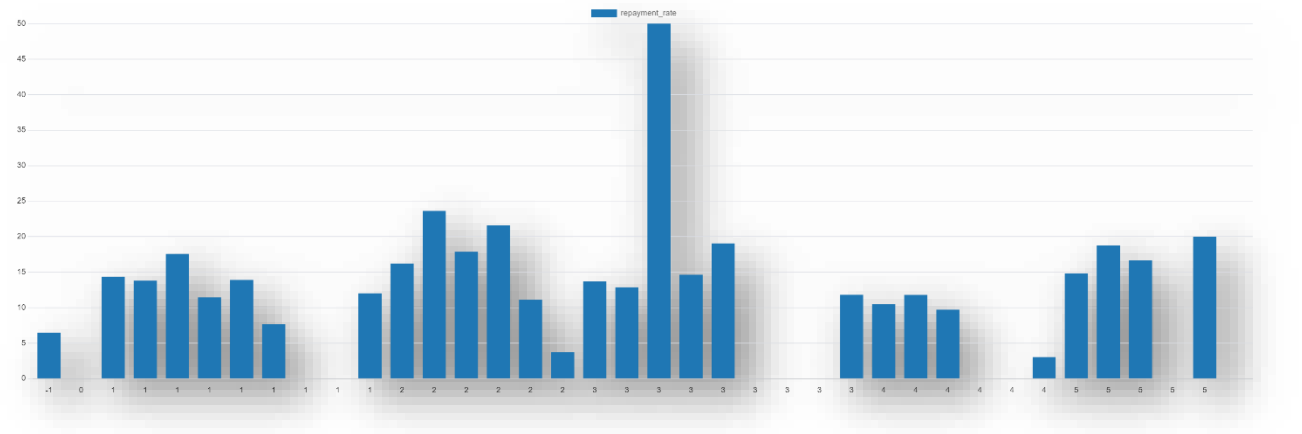


Figure 2This grouped chart will provide insight into how marital status and number of dependents impact the likelihood of loan repayment.

- **No Dependents (0 children)**:
  - Across most marital statuses, the repayment rate is relatively low when borrowers have no dependents, which might imply less financial motivation to repay the loans.
- **Increasing Dependents (1-3 children)**:

    - Borrowers with 1 to 3 children generally show a higher repayment rate compared to those with no children, suggesting that having children might indeed increase the motivation to repay loans.
    - For example, married borrowers (MaritalStatus = 2) with 1 child have a repayment rate of 23.62%, which is significantly higher than those with no children.
- **More Than 3 Children**:
    - The repayment rates tend to drop again for borrowers with more than three children, possibly indicating financial strain due to a larger family size.

Summary of Findings

1. **Proportion of Low-Repayment Loans**:
    - A significant portion of loans issued to borrowers with undefined marital statuses or no children have less than 1% repayment in the first three months.
2. **Repayment Likelihood**:
    - The analysis suggests that borrowers with 1 to 2 children tend to have higher repayment rates compared to those with no children or more than 3 children.
    - Marital status plays a critical role in repayment behavior, with married individuals showing better repayment performance than other categories.
3. **High-Risk Groups**:
    - Single, divorced, and widowed borrowers, especially those with more than three children, have lower repayment rates, indicating a higher risk of default.

Recommendations for the Estonian Country Manager

1. **Targeted Loan Strategies**:
    - Consider tighter credit policies or additional financial support for borrowers in high-risk groups (e.g., single or divorced individuals with multiple dependents).
    - For borrowers with undefined marital statuses, it's crucial to collect and verify accurate information to assess their creditworthiness better.
2. **Incentivize Borrowers with Children**:

- Since having 1-2 children seems to correlate with a higher likelihood of loan repayment, consider offering tailored repayment plans or incentives to these borrowers.

3. **Policy Adjustments**:
    - Consider creating specialized support programs for divorced and widowed borrowers to improve their chances of successful loan repayment.

Handling Missing Data: A Practical Approach

Based on the analysis of the missing data in the LoanData file, here are specific strategies for handling the missing values in various columns:

1. Columns with almost 100% Missing Values

- **Columns:** EL_V0, DateOfBirth, County, City.1, EmploymentPosition, Rating_V0
- **Strategy:** Since these columns have no data, it's best to **drop these columns** from the analysis, as they do not provide any information.

2. Columns with High Missing Percentage (50-95%)

- **Examples:** CreditScoreEsEquifaxRisk (94.6%), NrOfDependants (75.8%), Rating_V1 (83.9%), WorkExperience (75.8%), PrincipalWriteOffs (71.1%)
- **Strategy:**
    - **Imputation:** Depending on the column's data type:
        - For numerical columns (like NrOfDependants, WorkExperience), consider imputing missing values using the **mean** or **median**.
        - For categorical columns (like credit scores or ratings), impute missing values using the **mode** (most frequent category).
    - **Review necessity:** If these columns are not critical to the analysis and are challenging to impute accurately, consider dropping them.

3. Moderate Missing Percentage (30-50%)

- **Examples:** RecoveryStage (43%), StageActiveSince (42.3%), WorseLateCategory (39.6%)

- **Strategy:**
    - **Targeted Imputation:** Use **forward-fill** or **backward-fill** techniques if the data has a time-series aspect.
    - **Predictive Imputation:** Employ machine learning techniques, such as regression or decision trees, to predict missing values based on related variables.

4. Columns with Low Missing Percentage (less than 10%)

- **Examples:** EmploymentDurationCurrentEmployer (1.3%), PlannedInterestTillDate (0.67%)
- **Strategy:**
    - **Simple Imputation:** For a small percentage of missing values, you can safely impute them with the column's mean or mode without significantly affecting the analysis.

5. Time-Sensitive Data

- **Columns:** GracePeriodStart, GracePeriodEnd, DefaultDate, NextPaymentDate
- **Strategy:** Missing values in these columns could indicate that the loan never entered that phase or was never rescheduled. It's best to use a placeholder like **"Not Applicable"** or leave them as missing to indicate that the event didn't occur.

6. Data Flagging

- **Create Flags:** Add indicator columns to flag where data is missing for key features. This approach can help to identify patterns related to missing data and their impact on the analysis.

General Recommendations

- **Data Validation:** Ensure that the data collection process addresses the gaps identified in the missing fields to avoid similar issues in the future.
- **Analyze Impact:** Consider how each missing value affects the overall analysis and choose imputation techniques that minimize bias or errors in the results.