Marco 1 do Projeto: Validação de Hipótese em Base a Risco Relativo

```sql
--analisar o risco de não pagamento entre os mais jovens, o risco entre pessoas com mais empréstimos ativos e o risco entre aqueles que atrasaram pagamentos por mais de 90 dias, considerando a incidência(rr). A incidência ajudará a entender a probabilidade de esses eventos ocorrerem em cada grupo ao longo do tempo, o que pode ser mais relevante para a análise de risco de crédito.
WITH age_groups AS (
  SELECT --quartil da variável
  age,
    NTILE(4) OVER (ORDER BY age) AS quartil_idade,
    default_flag
  FROM
    `risco-relativo.credito.full_join`
)
SELECT --selecao de quartil, valor, total,risco
  quartil_idade,
  AVG(age) AS idade,
  SUM(default_flag) AS total_inadimplentes,
  COUNT(*) AS total_pessoas,
  SUM(default_flag) / COUNT(*) AS incidencia, --(RR)
  (SELECT SUM(default_flag) FROM `risco-relativo.credito.full_join`) / COUNT(*) AS incidencia_total,
  (SUM(default_flag) / COUNT(*)) / ((SELECT SUM(default_flag) FROM `risco-relativo.credito.full_join`) / COUNT(*)) AS risco_relativo
FROM
  age_groups
GROUP BY
  quartil_idade
ORDER BY
  risco_relativo DESC;
--Resultados indicam que, para essa amostra de dados, indicam que os mais jovens têm uma maior probabilidade de não pagar seus compromissos financeiros, enquanto os mais velhos têm uma menor probabilidade de pagamento.

WITH days_groups AS (
  SELECT
    CASE
      WHEN more_90_days_overdue > 1 THEN 'Mais de 90 dias'
      WHEN number_times_delayed_payment_loan_60_89_days > 1 THEN 'Mais de 60 dias'
      ELSE 'Menos de 60 dias'
    END AS faixa_atraso,
    default_flag
  FROM
    `risco-relativo.credito.full_join`
)
SELECT
  faixa_atraso,
  SUM(default_flag) AS total_inadimplentes,
  COUNT(*) AS total_pessoas,
  SUM(default_flag) / COUNT(*) AS incidencia,
  (SELECT SUM(default_flag) FROM `risco-relativo.credito.full_join`) / COUNT(*) AS incidencia_total,
  (SUM(default_flag) / COUNT(*)) / ((SELECT SUM(default_flag) FROM `risco-relativo.credito.full_join`) / COUNT(*)) AS risco_relativo
FROM
  days_groups
GROUP BY
  faixa_atraso
ORDER BY
  risco_relativo DESC;


WITH days90_groups AS (
  SELECT --quartil da variável
  more_90_days_overdue,
    NTILE(4) OVER (ORDER BY more_90_days_overdue) AS quartil_90dias,
    default_flag
  FROM
    `risco-relativo.credito.full_join`
)
SELECT --selecao de quartil, valor, total,risco
  quartil_90dias,
  AVG(more_90_days_overdue) AS dias90,
  SUM(default_flag) AS total_inadimplentes,
  COUNT(*) AS total_pessoas,
  SUM(default_flag) / COUNT(*) AS incidencia, --(RR)
  (SELECT SUM(default_flag) FROM `risco-relativo.credito.full_join`) / COUNT(*) AS incidencia_total,
  (SUM(default_flag) / COUNT(*)) / ((SELECT SUM(default_flag) FROM `risco-relativo.credito.full_join`) / COUNT(*)) AS risco_relativo
FROM
  days90_groups
GROUP BY
  quartil_90dias
ORDER BY
  risco_relativo DESC;
```

```
WITH days60_groups AS (
  SELECT
  number_times_delayed_payment_loan_60_89_days,
    NTILE(4) OVER (ORDER BY number_times_delayed_payment_loan_60_89_days) AS
quartil_60dias,
    default_flag
  FROM
    `risco-relativo.credito.full_join`
)
SELECT
  quartil_60dias,
  AVG(number_times_delayed_payment_loan_60_89_days) AS dias60,
  SUM(default_flag) AS total_inadimplentes,
  COUNT(*) AS total_pessoas,
  SUM(default_flag) / COUNT(*) AS incidencia,
  (SELECT SUM(default_flag) FROM `risco-relativo.credito.full_join`) / COUNT(*) AS
incidencia_total,
  (SUM(default_flag) / COUNT(*)) / ((SELECT SUM(default_flag) FROM
`risco-relativo.credito.full_join`) / COUNT(*)) AS risco_relativo
FROM
  days60_groups
GROUP BY
  quartil_60dias
ORDER BY
  risco_relativo DESC;

WITH days30_groups AS (
  SELECT
  number_times_delayed_payment_loan_30_59_days,
    NTILE(4) OVER (ORDER BY number_times_delayed_payment_loan_60_89_days) AS
quartil_30dias,
    default_flag
  FROM
    `risco-relativo.credito.full_join`
)
SELECT
  quartil_30dias,
  AVG(number_times_delayed_payment_loan_30_59_days) AS dias30,
  SUM(default_flag) AS total_inadimplentes,
  COUNT(*) AS total_pessoas,
  SUM(default_flag) / COUNT(*) AS incidencia,
  (SELECT SUM(default_flag) FROM `risco-relativo.credito.full_join`) / COUNT(*) AS
incidencia_total,
```

```sql
(SUM(default_flag) / COUNT(*)) / ((SELECT SUM(default_flag) FROM
`risco-relativo.credito.full_join`) / COUNT(*)) AS risco_relativo
FROM
  days30_groups
GROUP BY
  quartil_30dias
ORDER BY
  risco_relativo DESC;
--Estes resultados indicam que, para essa amostra de dados, as pessoas com mais de
90 dias de atraso têm um risco relativo maior de não pagamento em comparação com as
outras faixas de atraso. Isso sugere que, ao menos nesta amostra, a hipótese de
maior número de dias de atraso estar associada a um maior risco de não pagamento se
confirma.

WITH active_credit_groups AS (
  SELECT
    using_lines_not_secured_personal_assets,
    NTILE(4) OVER (ORDER BY using_lines_not_secured_personal_assets) AS
quartil_credito,
    default_flag
  FROM
    `risco-relativo.credito.full_join`
)
SELECT
  quartil_credito,
  AVG(using_lines_not_secured_personal_assets) AS uso_credito,
  SUM(default_flag) AS total_inadimplentes,
  COUNT(*) AS total_pessoas,
  SUM(default_flag) / COUNT(*) AS incidencia,
  (SELECT SUM(default_flag) FROM `risco-relativo.credito.full_join`) / COUNT(*) AS
incidencia_total,
  (SUM(default_flag) / COUNT(*)) / ((SELECT SUM(default_flag) FROM
`risco-relativo.credito.full_join`) / COUNT(*)) AS risco_relativo
FROM
  active_credit_groups
GROUP BY
  quartil_credito
ORDER BY
  risco_relativo DESC;

--Esses resultados indicam que, para essa amostra de dados, o risco de
inadimplência aumenta significativamente à medida que o uso do crédito é ativo, com
o quartil mais alto apresentando um risco relativo significativamente maior em
comparação com os outros quartis.
```

```sql
WITH loan_groups AS (
  SELECT
  total_loan,
    NTILE(4) OVER (ORDER BY total_loan) AS quartil_loan,
    default_flag
  FROM
    `risco-relativo.credito.full_join`
)
SELECT
  quartil_loan,
  AVG(total_loan) AS loan,
  SUM(default_flag) AS total_inadimplentes,
  COUNT(*) AS total_pessoas,
  SUM(default_flag) / COUNT(*) AS incidencia,
  (SELECT SUM(default_flag) FROM `risco-relativo.credito.full_join`) / COUNT(*) AS incidencia_total,
  (SUM(default_flag) / COUNT(*)) / ((SELECT SUM(default_flag) FROM `risco-relativo.credito.full_join`) / COUNT(*)) AS risco_relativo
FROM
  loan_groups
GROUP BY
  quartil_loan
ORDER BY
  risco_relativo DESC;
--Esses resultados indicam que, para essa amostra de dados, há uma tendência de aumento no risco de inadimplência à medida que o total de crédito aumenta.

WITH salary_groups AS (
  SELECT
    last_month_salary_median,
    NTILE(4) OVER (ORDER BY last_month_salary_median) AS quartil_salario,
    default_flag
  FROM
    `risco-relativo.credito.full_join`
)
SELECT
  quartil_salario,
  AVG(last_month_salary_median) AS salario,
  SUM(default_flag) AS total_inadimplentes,
  COUNT(*) AS total_pessoas,
  SUM(default_flag) / COUNT(*) AS incidencia,
  (SELECT SUM(default_flag) FROM `risco-relativo.credito.full_join`) / COUNT(*) AS incidencia_total,
(SUM(default_flag) / COUNT(*)) / ((SELECT SUM(default_flag) FROM
`risco-relativo.credito.full_join`) / COUNT(*)) AS risco_relativo
FROM
  salary_groups
GROUP BY
  quartil_salario
ORDER BY
  risco_relativo DESC;
--Esses resultados indicam que, para essa amostra de dados, há uma tendência de
redução no risco de inadimplência à medida que o salário aumenta, com o quartil
mais alto apresentando um risco relativo menor em comparação com os outros quartis.

WITH tipo_credito_groups AS (
  SELECT
  clean_loan_type,
    CASE WHEN clean_loan_type = 'Real Estate' THEN 1 ELSE 0 END AS tipo_credito,
    default_flag
  FROM
    `risco-relativo.credito.full_join`
)
SELECT
  tipo_credito,
  SUM(default_flag) AS total_inadimplentes,
  COUNT(*) AS total_pessoas,
  SUM(default_flag) / COUNT(*) AS incidencia,
  (SELECT SUM(default_flag) FROM `risco-relativo.credito.full_join`) / COUNT(*) AS
incidencia_total,
  (SUM(default_flag) / COUNT(*)) / ((SELECT SUM(default_flag) FROM
`risco-relativo.credito.full_join`) / COUNT(*)) AS risco_relativo
FROM
  tipo_credito_groups
GROUP BY
  tipo_credito
ORDER BY
  risco_relativo DESC;
--Esses resultados sugerem que, para esse conjunto de dados, pessoas com
'tipo_credito' igual a 0 (outher/null) têm uma probabilidade maior de
inadimplência, enquanto aquelas com 'tipo_credito' igual a 1 (real state) têm uma
probabilidade menor.

WITH dependents_groups AS (
  SELECT
  number_dependents_median,
    NTILE(4) OVER (ORDER BY number_dependents_median) AS quartil_dependents,
      default_flag
  FROM
    `risco-relativo.credito.full_join`
)
SELECT
  quartil_dependents,
  AVG(number_dependents_median) AS dependencia,
  SUM(default_flag) AS total_inadimplentes,
  COUNT(*) AS total_pessoas,
  SUM(default_flag) / COUNT(*) AS incidencia,
  (SELECT SUM(default_flag) FROM `risco-relativo.credito.full_join`) / COUNT(*) AS
incidencia_total,
  (SUM(default_flag) / COUNT(*)) / ((SELECT SUM(default_flag) FROM
`risco-relativo.credito.full_join`) / COUNT(*)) AS risco_relativo
FROM
  dependents_groups
GROUP BY
  quartil_dependents
ORDER BY
  risco_relativo DESC;

--Esses resultados indicam que, para essa amostra de dados, não há uma tendência
clara em relação ao risco de inadimplência com base na variável de dependência. Os
riscos relativos são relativamente próximos entre os diferentes.
```

Query Consultas

Identificar Duplicados e Nulos:

```
SELECT *
FROM `credito.default`
WHERE user_id IS NULL OR default_flag IS NULL;
SELECT user_id, COUNT(*)
FROM `credito.default`
GROUP BY user_id
HAVING COUNT(*) > 1; --nao tem

SELECT *
FROM `credito.loans_detail`
WHERE user_id IS NULL OR more_90_days_overdue
 IS NULL OR using_lines_not_secured_personal_assets IS NULL OR
number_times_delayed_payment_loan_30_59_days IS NULL OR debt_ratio IS NULL OR
number_times_delayed_payment_loan_60_89_days IS NULL;
SELECT user_id, COUNT(*)
FROM `risco-relativo.credito.loans_detail`
GROUP BY user_id
HAVING COUNT(*) > 1; --nao tem

SELECT *
FROM `risco-relativo.credito.loans_outstandig`
WHERE user_id IS NULL OR loan_id IS NULL OR loan_type IS NULL;
SELECT user_id, COUNT(*)
FROM `risco-relativo.credito.loans_outstandig`
GROUP BY user_id
HAVING COUNT(*) > 1; -- 35mil duplicados

SELECT *
FROM `risco-relativo.credito.user_info`
WHERE user_id IS NULL OR age IS NULL OR sex
 IS NULL OR last_month_salary
 IS NULL OR number_dependents
 IS NULL;
SELECT user_id, COUNT(*)
FROM `risco-relativo.credito.user_info`
GROUP BY user_id
HAVING COUNT(*) > 1; --7199 null
```

Medidas:

```
--Medidas de tendecia central, dispercao, risco relativo (variavel numerica)
SELECT
  COUNT(*) AS total_records,
  AVG(last_month_salary) AS mean_salary,
  STDDEV(last_month_salary) AS std_dev_salary,
  MIN(last_month_salary) AS min_value_salary,
  MAX(last_month_salary) AS max_value_salary,
  APPROX_QUANTILES(last_month_salary, 2)[OFFSET(1)] AS median_salary,
  STDDEV(last_month_salary) / AVG(last_month_salary) AS risk_relative_salary
FROM
  `risco-relativo.credito.user_info`;

SELECT
  COUNT(*) AS total_records,
  AVG(number_dependents) AS mean_dependents,
  STDDEV(number_dependents) AS std_dev_dependents,
  MIN(number_dependents) AS min_value_dependents,
  MAX(number_dependents) AS max_value_dependents,
  APPROX_QUANTILES(number_dependents, 2)[OFFSET(1)] AS median_dependents,
  STDDEV(number_dependents) / AVG(number_dependents) AS risk_relative_dependents
FROM
  `risco-relativo.credito.user_info`;

SELECT
  COUNT(*) AS total_records,
  AVG(age) AS mean_age,
  STDDEV(age) AS std_dev_age,
  MIN(age) AS min_value_age,
  MAX(age) AS max_value_age,
  APPROX_QUANTILES(age, 2)[OFFSET(1)] AS median_age,
  STDDEV(age) / AVG(age) AS risk_relative_age
FROM
  `risco-relativo.credito.user_info`;

SELECT
  COUNT(*) AS total_records,
  AVG(using_lines_not_secured_personal_assets) AS mean_not_personal,
  STDDEV(using_lines_not_secured_personal_assets) AS std_dev_not_personal,
  MIN(using_lines_not_secured_personal_assets) AS min_value_not_personal,
  MAX(using_lines_not_secured_personal_assets) AS max_value_not_personal,
  APPROX_QUANTILES(using_lines_not_secured_personal_assets, 2)[OFFSET(1)] AS
median_not_personal,
  STDDEV(using_lines_not_secured_personal_assets) /
AVG(using_lines_not_secured_personal_assets) AS risk_relative_not_personal
FROM
  `risco-relativo.credito.loans_detail`;

SELECT
  COUNT(*) AS total_records,
  AVG(debt_ratio) AS mean_ratio,
  STDDEV(debt_ratio) AS std_dev_ratio,
  MIN(debt_ratio) AS min_value_ratio,
  MAX(debt_ratio) AS max_value_ratio,
  APPROX_QUANTILES(debt_ratio, 2)[OFFSET(1)] AS median_ratio,
  STDDEV(debt_ratio) / AVG(debt_ratio) AS risk_relative_ratio
FROM
  `risco-relativo.credito.loans_detail`;

SELECT
  COUNT(*) AS total_records,
  AVG(more_90_days_overdue) AS mean_90days,
  STDDEV(more_90_days_overdue) AS std_dev_90days,
  MIN(more_90_days_overdue) AS min_value_90days,
  MAX(more_90_days_overdue) AS max_value_90days,
  APPROX_QUANTILES(more_90_days_overdue, 2)[OFFSET(1)] AS median_90days,
  STDDEV(more_90_days_overdue) / AVG(more_90_days_overdue) AS risk_relative_90days
FROM
  `risco-relativo.credito.loans_detail`;

SELECT
  COUNT(*) AS total_records,
  AVG(number_times_delayed_payment_loan_30_59_days) AS mean_30days,
  STDDEV(number_times_delayed_payment_loan_30_59_days) AS std_dev_30days,
  MIN(number_times_delayed_payment_loan_30_59_days) AS min_value_30days,
  MAX(number_times_delayed_payment_loan_30_59_days) AS max_value_30days,
  APPROX_QUANTILES(number_times_delayed_payment_loan_30_59_days, 2)[OFFSET(1)] AS
median_30days,
  STDDEV(number_times_delayed_payment_loan_30_59_days) /
AVG(number_times_delayed_payment_loan_30_59_days) AS risk_relative_30days
FROM
  `risco-relativo.credito.loans_detail`;

SELECT
  COUNT(*) AS total_records,
  AVG(number_times_delayed_payment_loan_60_89_days) AS mean_60days,
  STDDEV(number_times_delayed_payment_loan_60_89_days) AS std_dev_60days,
  MIN(number_times_delayed_payment_loan_60_89_days) AS min_value_60days,
  MAX(number_times_delayed_payment_loan_60_89_days) AS max_value_60days,
  APPROX_QUANTILES(number_times_delayed_payment_loan_60_89_days, 2)[OFFSET(1)] AS
median_60days,
  STDDEV(number_times_delayed_payment_loan_60_89_days) /
AVG(number_times_delayed_payment_loan_60_89_days) AS risk_relative_60days
FROM
  `risco-relativo.credito.loans_detail`;

--correlaçao (binaria e numericas)
SELECT
  CORR(more_90_days_overdue, number_times_delayed_payment_loan_30_59_days) AS
correlation_30day_90day,
  CORR(more_90_days_overdue, number_times_delayed_payment_loan_60_89_days) AS
correlation_60day_90day,
  CORR(number_times_delayed_payment_loan_30_59_days,
number_times_delayed_payment_loan_60_89_days) AS correlation_30day_60day,
  CORR(default_flag, more_90_days_overdue) AS correlation_Flag_90day,
  CORR(default_flag, number_times_delayed_payment_loan_60_89_days) AS
correlation_Flag_60day,
  CORR(default_flag, number_times_delayed_payment_loan_30_59_days) AS
correlation_flag_30day,
  CORR(default_flag, last_month_salary) AS correlation_flag_salary,
  CORR(default_flag, number_dependents) AS correlation_flag_dependents,
  CORR(default_flag, age) AS correlation_flag_age,
  CORR(default_flag, using_lines_not_secured_personal_assets) AS
correlation_flag_not_personal,
  CORR(default_flag, debt_ratio) AS correlation_flag_ratio,
FROM
  `risco-relativo.credito.loans_detail` AS ld
JOIN
  `risco-relativo.credito.default` AS d
ON
  ld.user_id = d.user_id
JOIN
  `risco-relativo.credito.user_info` AS ui
ON
  ld.user_id = ui.user_id
```

Identificar outlier:

```
WITH salary_stats AS (
  SELECT
    COUNT(*) AS total_records,
    APPROX_QUANTILES(last_month_salary, 100)[OFFSET(25)] AS quartile_1,
    APPROX_QUANTILES(last_month_salary, 100)[OFFSET(50)] AS median_salary,
    APPROX_QUANTILES(last_month_salary, 100)[OFFSET(75)] AS quartile_3
  FROM
    `risco-relativo.credito.user_info`
),
outliers AS (
  SELECT
    last_month_salary,
    IF(last_month_salary < quartile_1 - (quartile_3 - quartile_1) * 1.5 OR
last_month_salary > quartile_3 + (quartile_3 - quartile_1) * 1.5, 'Outlier', 'Not
Outlier') AS outlier_status
  FROM
    `risco-relativo.credito.user_info`,
    salary_stats
)
SELECT
  outlier_status,
  COUNT(*) AS outlier_count
FROM
  outliers
GROUP BY
  outlier_status;  --quantidade de outlier: 1170

WITH age_stats AS (
  SELECT
    COUNT(*) AS total_records,
    APPROX_QUANTILES(age, 100)[OFFSET(25)] AS quartile_1,
    APPROX_QUANTILES(age, 100)[OFFSET(50)] AS median_age,
    APPROX_QUANTILES(age, 100)[OFFSET(75)] AS quartile_3
  FROM
    `risco-relativo.credito.user_info`
),
outliers AS (
  SELECT
    age,
    IF(age < quartile_1 - (quartile_3 - quartile_1) * 1.5 OR age > quartile_3 +
(quartile_3 - quartile_1) * 1.5, 'Outlier', 'Not Outlier') AS outlier_status
  FROM
```

```
    `risco-relativo.credito.user_info`,
    age_stats
)
SELECT
  outlier_status,
  COUNT(*) AS outlier_count
FROM
  outliers
GROUP BY
  outlier_status; --só total de outlier: 10

WITH dependents_stats AS (
  SELECT
    COUNT(*) AS total_records,
    APPROX_QUANTILES(number_dependents, 100)[OFFSET(25)] AS quartile_1,
    APPROX_QUANTILES(number_dependents, 100)[OFFSET(50)] AS median_dependents,
    APPROX_QUANTILES(number_dependents, 100)[OFFSET(75)] AS quartile_3
  FROM
    `risco-relativo.credito.user_info`
),
outliers AS (
  SELECT
    number_dependents,
    IF(number_dependents < quartile_1 - (quartile_3 - quartile_1) * 1.5 OR
number_dependents > quartile_3 + (quartile_3 - quartile_1) * 1.5, 'Outlier', 'Not
Outlier') AS outlier_status
  FROM
    `risco-relativo.credito.user_info`,
    dependents_stats
)
SELECT
  outlier_status,
  COUNT(*) AS outlier_count
FROM
  outliers
GROUP BY
  outlier_status; --só total de outlier: 3230

WITH not_personal_stats AS (
  SELECT
    COUNT(*) AS total_records,
    APPROX_QUANTILES(using_lines_not_secured_personal_assets, 100)[OFFSET(25)] AS
quartile_1,
    APPROX_QUANTILES(using_lines_not_secured_personal_assets, 100)[OFFSET(50)] AS
median_not_personal,
    APPROX_QUANTILES(using_lines_not_secured_personal_assets, 100)[OFFSET(75)] AS
quartile_3
  FROM
    `risco-relativo.credito.loans_detail`
),
outliers AS (
  SELECT
    using_lines_not_secured_personal_assets,
    IF(using_lines_not_secured_personal_assets < quartile_1 - (quartile_3 -
quartile_1) * 1.5 OR using_lines_not_secured_personal_assets > quartile_3 +
(quartile_3 - quartile_1) * 1.5, 'Outlier', 'Not Outlier') AS outlier_status
  FROM
    `risco-relativo.credito.loans_detail`,
    not_personal_stats
)
SELECT
  outlier_status,
  COUNT(*) AS outlier_count
FROM
  outliers
GROUP BY
  outlier_status; --177

WITH ratio_stats AS (
  SELECT
    COUNT(*) AS total_records,
    APPROX_QUANTILES(debt_ratio, 100)[OFFSET(25)] AS quartile_1,
    APPROX_QUANTILES(debt_ratio, 100)[OFFSET(50)] AS median_ratio,
    APPROX_QUANTILES(debt_ratio, 100)[OFFSET(75)] AS quartile_3
  FROM
    `risco-relativo.credito.loans_detail`
),
outliers AS (
  SELECT
    debt_ratio,
    IF(debt_ratio < quartile_1 - (quartile_3 - quartile_1) * 1.5 OR debt_ratio >
quartile_3 + (quartile_3 - quartile_1) * 1.5, 'Outlier', 'Not Outlier') AS
outlier_status
  FROM
    `risco-relativo.credito.loans_detail`,
    ratio_stats
)
```

```
SELECT
  outlier_status,
  COUNT(*) AS outlier_count
FROM
  outliers
GROUP BY
  outlier_status; --7583

WITH days90_stats AS (
  SELECT
    COUNT(*) AS total_records,
    APPROX_QUANTILES(more_90_days_overdue, 100)[OFFSET(25)] AS quartile_1,
    APPROX_QUANTILES(more_90_days_overdue, 100)[OFFSET(50)] AS median_90days,
    APPROX_QUANTILES(more_90_days_overdue, 100)[OFFSET(75)] AS quartile_3
  FROM
    `risco-relativo.credito.loans_detail`
),
outliers AS (
  SELECT
    more_90_days_overdue,
    IF(more_90_days_overdue < quartile_1 - (quartile_3 - quartile_1) * 1.5 OR
more_90_days_overdue > quartile_3 + (quartile_3 - quartile_1) * 1.5, 'Outlier',
'Not Outlier') AS outlier_status
  FROM
    `risco-relativo.credito.loans_detail`,
    days90_stats
)
SELECT
  outlier_status,
  COUNT(*) AS outlier_count
FROM
  outliers
GROUP BY
  outlier_status; --1946

WITH days30_stats AS (
  SELECT
    COUNT(*) AS total_records,
    APPROX_QUANTILES(number_times_delayed_payment_loan_30_59_days, 100)[OFFSET(25)]
AS quartile_1,
    APPROX_QUANTILES(number_times_delayed_payment_loan_30_59_days, 100)[OFFSET(50)]
AS median_30days,
    APPROX_QUANTILES(number_times_delayed_payment_loan_30_59_days, 100)[OFFSET(75)]
AS quartile_3
```

```
  FROM
    `risco-relativo.credito.loans_detail`
),
outliers AS (
  SELECT
    number_times_delayed_payment_loan_30_59_days,
    IF(number_times_delayed_payment_loan_30_59_days < quartile_1 - (quartile_3 -
quartile_1) * 1.5 OR number_times_delayed_payment_loan_30_59_days > quartile_3 +
(quartile_3 - quartile_1) * 1.5, 'Outlier', 'Not Outlier') AS outlier_status
  FROM
    `risco-relativo.credito.loans_detail`,
    days30_stats
)
SELECT
  outlier_status,
  COUNT(*) AS outlier_count
FROM
  outliers
GROUP BY
  outlier_status;  --5812

WITH days60_stats AS (
  SELECT
    COUNT(*) AS total_records,
    APPROX_QUANTILES(number_times_delayed_payment_loan_60_89_days, 100)[OFFSET(25)]
AS quartile_1,
    APPROX_QUANTILES(number_times_delayed_payment_loan_60_89_days, 100)[OFFSET(50)]
AS median_60days,
    APPROX_QUANTILES(number_times_delayed_payment_loan_60_89_days, 100)[OFFSET(75)]
AS quartile_3
  FROM
    `risco-relativo.credito.loans_detail`
),
outliers AS (
  SELECT
    number_times_delayed_payment_loan_60_89_days,
    IF(number_times_delayed_payment_loan_60_89_days < quartile_1 - (quartile_3 -
quartile_1) * 1.5 OR number_times_delayed_payment_loan_60_89_days > quartile_3 +
(quartile_3 - quartile_1) * 1.5, 'Outlier', 'Not Outlier') AS outlier_status
  FROM
    `risco-relativo.credito.loans_detail`,
    days60_stats
)
SELECT
```

```
  outlier_status,
  COUNT(*) AS outlier_count
FROM
  outliers
GROUP BY
  outlier_status; --1865
```

Decisões:

```
SELECT  --analisar padrões de inadimplência em relação a usuários com dados
faltantes
  u.*,
  d.default_flag
FROM
  `risco-relativo.credito.user_info` AS u
LEFT JOIN
  `risco-relativo.credito.default` AS d
ON
  u.user_id = d.user_id
WHERE
  u.last_month_salary IS NULL AND d.default_flag = 1; --130

SELECT --trocar nulos pela mediana
  IFNULL(last_month_salary, (SELECT APPROX_QUANTILES(last_month_salary,
2)[OFFSET(1)] FROM `risco-relativo.credito.user_info`)) AS
last_month_salary_median,
  IFNULL(number_dependents, (SELECT APPROX_QUANTILES(number_dependents,
2)[OFFSET(1)] FROM `risco-relativo.credito.user_info`)) AS number_dependents_median
FROM
  `risco-relativo.credito.user_info`;
```

```sql
SELECT DISTINCT loan_type -- identificar o valor da variavel categorica
FROM `risco-relativo.credito.loans_outstandig`;

SELECT
  COUNT(loan_id) AS total_loan -- identifica quantos valores de uma variavel tem da
outra (de loan_id)
FROM
  `risco-relativo.credito.loans_outstandig`
WHERE
  user_id = 2; -- por usuario

WITH cleaned_data AS (
  SELECT -- padronozacao de variavel categoricas
    user_id,
    INITCAP(REGEXP_REPLACE(loan_type, r'[^\w\s]', '')) AS clean_loan_type,
    loan_id
  FROM
    `risco-relativo.credito.loans_outstandig`
  WHERE
    loan_type IS NOT NULL
)
SELECT
  user_id,
  IF(clean_loan_type = 'Others', 'Other', clean_loan_type) AS clean_loan_type,
  COUNT(loan_id) AS total_loan --conta os valores da variavel para cada id
FROM
  cleaned_data
GROUP BY
  user_id, clean_loan_type;
```

Juntar tabelas e Identificar dados:

```sql
CREATE OR REPLACE TABLE `risco-relativo.credito.full_join` AS
WITH salary_median AS (
  SELECT --variavel numerica nula trocarda por mediana
    user_id,
    IFNULL(last_month_salary, (SELECT APPROX_QUANTILES(last_month_salary, 2)[OFFSET(1)] FROM `risco-relativo.credito.user_info`)) AS last_month_salary_median,
    IFNULL(number_dependents, (SELECT APPROX_QUANTILES(number_dependents, 2)[OFFSET(1)] FROM `risco-relativo.credito.user_info`)) AS number_dependents_median
  FROM
    `risco-relativo.credito.user_info`
),

cleaned_loans AS (
  SELECT --padronizar variavel categoricas
    user_id,
    INITCAP(REGEXP_REPLACE(loan_type, r'[^\w\s]', '')) AS clean_loan_type,
    loan_id
  FROM
    `risco-relativo.credito.loans_outstandig`
  WHERE
    loan_type IS NOT NULL
),

total_loans AS (
  SELECT
    user_id,
    IF(clean_loan_type = 'Others', 'Other', clean_loan_type) AS clean_loan_type,
    COUNT(loan_id) AS total_loan --agrupar e criar variavel numerica
  FROM
    cleaned_loans
  GROUP BY
    user_id, clean_loan_type
),
--juntar tabelas
merged_data AS (
  SELECT DISTINCT
    u.user_id,
    u.age,
    u.sex,
    sm.last_month_salary_median,
    sm.number_dependents_median,
    tl.total_loan,
    tl.clean_loan_type,
    ld.more_90_days_overdue,
    ld.number_times_delayed_payment_loan_30_59_days,
    ld.number_times_delayed_payment_loan_60_89_days,
    ld.using_lines_not_secured_personal_assets,
    ld.debt_ratio,
    d.default_flag,
    ROW_NUMBER() OVER(PARTITION BY u.user_id ORDER BY u.user_id) AS row_num
  FROM
    `risco-relativo.credito.user_info` AS u
  FULL OUTER JOIN
    total_loans AS tl
  ON
    u.user_id = tl.user_id
  FULL OUTER JOIN
    `risco-relativo.credito.loans_outstandig` AS l
  ON
    u.user_id = l.user_id
 FULL OUTER JOIN
    `risco-relativo.credito.loans_detail` AS ld
  ON
    u.user_id = ld.user_id
  FULL OUTER JOIN
    `risco-relativo.credito.default` AS d
  ON
    u.user_id = d.user_id
  FULL OUTER JOIN
    salary_median AS sm
  ON
    u.user_id = sm.user_id
)

SELECT
  user_id,
  age,
  sex,
  last_month_salary_median,
  number_dependents_median,
  total_loan,
  clean_loan_type,
  more_90_days_overdue,
  number_times_delayed_payment_loan_30_59_days,
  number_times_delayed_payment_loan_60_89_days,
  using_lines_not_secured_personal_assets,
  debt_ratio,
  default_flag
FROM
  merged_data
WHERE
```

```
  row_num = 1;

--identificar dados
SELECT *
FROM `risco-relativo.credito.full_join`
WHERE user_id IS NULL OR age  IS NULL OR sex IS NULL OR last_month_salary_median IS
NULL OR number_dependents_median IS NULL OR total_loan IS NULL
 OR clean_loan_type IS NULL OR more_90_days_overdue IS NULL OR
using_lines_not_secured_personal_assets IS NULL OR
number_times_delayed_payment_loan_30_59_days IS NULL OR debt_ratio IS NULL OR
number_times_delayed_payment_loan_60_89_days IS NULL OR default_flag IS NULL;
--425
SELECT user_id, COUNT(*)
FROM `risco-relativo.credito.full_join`
GROUP BY user_id
HAVING COUNT(*) > 1;

SELECT
  COUNT(*) AS total_records,
  AVG(total_loan) AS mean_loan,
  STDDEV(total_loan) AS std_dev_loan,
  MIN(total_loan) AS min_value_loan,
  MAX(total_loan) AS max_value_loan,
  APPROX_QUANTILES(total_loan, 2)[OFFSET(1)] AS median_loan,
  STDDEV(total_loan) / AVG(total_loan) AS risk_relative_loan
FROM
  `risco-relativo.credito.full_join`;

SELECT
  CORR(default_flag, total_loan) AS correlation_flag_loan
FROM
  `risco-relativo.credito.full_join`;

SELECT
  *,
  default_flag
FROM
  `risco-relativo.credito.full_join`
WHERE
   total_loan IS NULL AND clean_loan_type IS NULL AND default_flag = 0; --364

WITH flag AS (
  SELECT DISTINCT
    user_id,
    total_loan,
    clean_loan_type,
    default_flag
  FROM
  `risco-relativo.credito.full_join`
WHERE
   total_loan IS NULL AND clean_loan_type IS NULL AND default_flag = 1
)
SELECT
  user_id,
  total_loan,
  clean_loan_type,
  default_flag
FROM
  flag
GROUP BY
  user_id, clean_loan_type, total_loan, default_flag;   --61

WITH loans_stats AS (
  SELECT
    COUNT(*) AS total_records,
    APPROX_QUANTILES(total_loan, 100)[OFFSET(25)] AS quartile_1,
    APPROX_QUANTILES(total_loan, 100)[OFFSET(50)] AS median_loan,
    APPROX_QUANTILES(total_loan, 100)[OFFSET(75)] AS quartile_3
  FROM
    `risco-relativo.credito.full_join`
),
outliers AS (
  SELECT
    total_loan,
    IF(total_loan < quartile_1 - (quartile_3 - quartile_1) * 1.5 OR total_loan >
quartile_3 + (quartile_3 - quartile_1) * 1.5, 'Outlier', 'Not Outlier') AS
outlier_status
  FROM
    `risco-relativo.credito.full_join`,
    loans_stats
)
SELECT
  outlier_status,
  COUNT(*) AS outlier_count
FROM
  outliers
GROUP BY
  outlier_status; --3089
```