# Insurance Company Take Home

Nathan Chiu
4.23.25

# PART I (recommended time: 45 minutes)

## Question 1.

**Read all of the questions in this case study, and then review the source tables. You will find a data dictionary in the appendix of this document. Please comment on the integrity of the data.**

Some possible questions for you to discuss:

- What issues are you finding?
- How important are these issues to your ability to analyze the data?
- How do you plan to address them?

To evaluate the integrity of the data, I looked at three criteria:

1. **Completeness**: Are there any missing values?
2. **Consistency:** Are there contradictions in the values and the relationships among each other?
3. **Accuracy:** Are there incorrect or implausible values?

**Note:** I am using BigQuery to write SQL and am naming the three tables as following: ‘data.drivers’, ‘data.users’, and ‘data.policies’.

**What issues are you finding?**

I began by checking for missing values, then ran a series of queries to identify duplicate users, drivers, and policies. I also validated the consistency of user IDs across all three tables to ensure proper joins. Overall, I found missing demographic data in the driver table and some inconsistencies in the policy table such as illogical dates. While the data appears generally accurate across all tables, the boundaries for the currently_insured_length_months field should be clearly defined. For example, it's unclear whether a value of 36 months belongs in the "12–36" or "36–60" category, which could lead to edge case misclassification.

```
-- Check for missing values
SELECT
```

```
  SUM(CASE WHEN education IS NULL OR education = '' OR TRIM(education) = '' OR
education IN ('NULL', 'N/A', 'Unknown', 'None') THEN 1 ELSE 0 END) AS
total_missing_education,
  SUM(CASE WHEN occupation IS NULL OR occupation = '' OR TRIM(occupation) = '' OR
occupation IN ('NULL', 'N/A', 'Unknown', 'None') THEN 1 ELSE 0 END) AS
total_missing_occupation,
  SUM(CASE WHEN marital_status IS NULL OR marital_status = '' OR TRIM(marital_status)
= '' OR marital_status IN ('NULL', 'N/A', 'Unknown', 'None') THEN 1 ELSE 0 END) AS
total_missing_marital_status
FROM `data.drivers`;
```

- The driver table is missing a substantial number of education, occupation, and marital status values with **31.22%, 45.99%, and 31.15%** of them missing respectively

```
-- Check for inconsistencies
SELECT
  COUNT(*) AS total_policies,
  SUM(CASE WHEN start_date > policy_end_date THEN 1 ELSE 0 END) AS invalid_date_range,
  SUM(CASE WHEN cancellation_date IS NOT NULL AND cancellation_date < start_date THEN
1 ELSE 0 END) AS invalid_cancellation_date,
  SUM(CASE WHEN cancellation_date IS NOT NULL AND cancellation_date > policy_end_date
THEN 1 ELSE 0 END) AS cancellation_after_end_date
FROM `data.policies`;
```

- 2553/345,497 (**0.7%**) policies have cancellation dates after the end dates, which doesn't make sense

```
-- Check for duplicate user_ids in user_table (no dupes)
SELECT
  user_id,
  COUNT(*) AS count
FROM `data.users`
GROUP BY user_id
HAVING COUNT(*) > 1
ORDER BY count DESC;

-- Check for users in policy_table not in user_table (referential integrity) - Same #
of users
SELECT
  COUNT(DISTINCT p.user_id) AS policy_user_count,
  COUNT(DISTINCT u.user_id) AS user_count,
```

```
  COUNT(DISTINCT p.user_id) - COUNT(DISTINCT u.user_id) AS difference
FROM `data.policies` p
LEFT JOIN `data.users` u ON p.user_id = u.user_id;
```

- Did not see dupes or discrepancies in the number of users across the datasets

**How important are these issues to your ability to analyze the data?**

These issues impact our analysis in several ways:

- The high percentage of missing demographic data in the driver table (31-46%) limits our ability to analyze how these factors affect retention
- Date inconsistencies in the policy table (0.7% with cancellation dates after end dates) may affect retention calculations
- Ambiguous category boundaries for currently_insured_length could cause misclassification when analyzing retention patterns by insurance history

While these issues won't prevent analysis entirely, they require careful handling to ensure reliable conclusions.

**How do you plan to address them?**

I plan to address these issues with the following approach:

- For missing demographic values: I'll analyze patterns of missingness to determine if they're random or systematic. For retention analysis, I'll create a separate "unknown" category rather than dropping these records, which would significantly reduce our sample size.
- For date inconsistencies: I'll create a clean subset of policies where dates are logically consistent. For calculation of retention metrics, I'll clearly document how these edge cases are handled.
- For category boundary ambiguity: I'll document assumptions about category boundaries (e.g., "36 months belongs to 36-60 category") and maintain consistent application throughout the analysis.
- For all analyses: I'll run sensitivity analyses to determine how different approaches to handling these data quality issues affect our conclusions.

In the given timespan, I won’t realistically have the time to conduct all the above analyses besides the first bullet point, which is the most material. **Edit:** After completing this assignment, marital status, occupation, and education aren’t major factors, but if we were to expand, I would suggest performing the first bullet point.

## Question 2.

Please use SQL to compute the following. Provide both your SQL query along with the answer to each question.

**A. How many users had an active policy on April 30, 2022? A customer has an active policy if they had at least one policy that started on or before April 30, 2022 and has not yet reached its policy_end_date or Cancellation_date.**

```
-- 2A
-- Count distinct users that fit the above criteria
SELECT
  COUNT(DISTINCT user_id) AS active_users
FROM `data.policies`
WHERE
  start_date <= '2022-04-30'
  AND (policy_end_date > '2022-04-30' AND (cancellation_date IS NULL OR
cancellation_date > '2022-04-30')
  );
```

***17,384 users had an active policy on April 30th, 2022.***

**B. What is the median length of a policy in months? Policy length is the policy end date minus the start date.**

```
-- 2B
-- Get policy lengths in months
WITH policy_lengths AS (
  SELECT
    policy_id,
    DATE_DIFF(policy_end_date, start_date, MONTH) AS length_months
  FROM `data.policies`
)

-- Get 50th percentile, which is the median
SELECT
  PERCENTILE_CONT(length_months, 0.5) OVER() AS median_length_months
FROM policy_lengths
LIMIT 1;
```

***The median length of a policy is 6 months.***

**C. Provide a distribution of % of policies by different discrete policy lengths**

```sql
-- 2C
-- Policy length in months and filtering out null dates to avoid including bad data
WITH policy_lengths AS (
  SELECT
    DATE_DIFF(COALESCE(policy_end_date, cancellation_date), start_date, MONTH) AS
policy_length_months
  FROM `data.policies`
  WHERE start_date IS NOT NULL
    AND (policy_end_date IS NOT NULL OR cancellation_date IS NOT NULL)
),

-- Count policies in each discrete policy buckeft
length_distribution AS (
  SELECT
    policy_length_months,
    COUNT(*) AS count
  FROM policy_lengths
  GROUP BY policy_length_months
),

-- Get total (denominator of total percentage)
total_policies AS (
  SELECT SUM(count) AS total FROM length_distribution
)

-- For each discrete month, list the count and percentage of policies
SELECT
  ld.policy_length_months,
  ld.count,
  ROUND(100 * ld.count / t.total, 2) AS percentage
FROM length_distribution ld
CROSS JOIN total_policies t
ORDER BY policy_length_months;
```

***Policy Length Distribution:***

| policy_length_months | count | percentage |
|---|---|---|
| 0 | 9 | 0 |
| 1 | 125 | 0.04 |

| | | |
|---|---|---|
| 2 | 282 | 0.08 |
| 3 | 322 | 0.09 |
| 4 | 329 | 0.1 |
| 5 | 1863 | 0.54 |
| 6 | 304941 | 88.26 |
| 7 | 1873 | 0.54 |
| 8 | 65 | 0.02 |
| 9 | 84 | 0.02 |
| 10 | 110 | 0.03 |
| 11 | 61 | 0.02 |
| 12 | 35377 | 10.24 |
| 13 | 56 | 0.02 |

D. Define the first day of a user's first policy with Insurance Company as the start of week 0. Define the 7-month retention rate (i.e. the 31-week retention rate) as the % of users who started a policy at week 0 that still had an active policy at the end of week 31. What is the 7 month retention rate for all users in this dataset?

**Exclude all users who started their first policy less than 31 weeks ago.**

```
-- 2D
-- Get start dates
WITH first_policies AS (
  SELECT
    user_id,
    MIN(start_date) AS first_policy_start
  FROM `data.policies`
  GROUP BY user_id
),

-- Exclude users who started first policy less than 31 weeks ago
eligible_users AS (
  SELECT
    user_id,
    first_policy_start,
    DATE_ADD(first_policy_start, INTERVAL 31 WEEK) AS week_31_date
  FROM first_policies
  WHERE first_policy_start <= DATE_SUB(CURRENT_DATE(), INTERVAL 31 WEEK)
),
```

```sql
-- Exclude users who started first policy less than 31 weeks ago
retention_flags AS (
  SELECT
    e.user_id,
    MAX(CASE WHEN p.start_date <= e.week_31_date
        AND (p.policy_end_date IS NULL OR p.policy_end_date > e.week_31_date)
        AND (p.cancellation_date IS NULL OR p.cancellation_date > e.week_31_date)
        THEN 1 ELSE 0
      END
    ) AS retained_flag
  FROM eligible_users e
  JOIN `data.policies` p
    ON e.user_id = p.user_id
  GROUP BY e.user_id
)

-- Compute the overall retention rate
SELECT
  ROUND(100.0 * SUM(retained_flag) / COUNT(user_id), 2) AS retention_rate_7_months
FROM retention_flags;
```

***The 7 month retention rate is 40.35%.***

—

## PART II (recommended time: 1 hour)

**Question 3. Create a weekly cohort retention table. Please provide your SQL query and a screenshot of your table.**

Below are specifics on what to build:
● The rows of the table are weekly cohorts based on when users started their first policy. The columns are the number of weeks since starting first policy. The cells of the table are % retention rates.
● The day that a user starts their first policy with Insurance Company is considered week 0.
● We say that a customer has retained if they have at least one active policy on the final day of each week
●Please show 52 rows (i.e. 1 year of weekly cohorts) and 12 columns (12 weeks of retention rates)

```sql
-- Get each user's first policy start date (week 0)
WITH first_policies AS (
  SELECT
    user_id,
    MIN(start_date) AS first_policy_start,
    DATE_TRUNC(MIN(start_date), WEEK(MONDAY)) AS cohort_week_start
  FROM `data.policies`
  GROUP BY user_id
),

-- Create a weekly calendar of up to 12 weeks per user from week 0 (in lieu of not having a dim_date table)
user_weeks AS (
  SELECT
    fp.user_id,
    fp.first_policy_start,
    DATE_TRUNC(fp.first_policy_start, WEEK(MONDAY)) AS cohort_week_start,
    GENERATE_ARRAY(0, 11) AS week_offsets
  FROM first_policies fp
),

-- Unnest to create one row per user per week
unnested_weeks AS (
  SELECT
    uw.user_id,
    uw.cohort_week_start,
    week_offset,
    DATE_ADD(uw.first_policy_start, INTERVAL week_offset WEEK) AS week_end_date
  FROM user_weeks uw, UNNEST(week_offsets) AS week_offset
),

-- Check if user was active at the end of each week
user_activity AS (
  SELECT
    uw.user_id,
    uw.cohort_week_start,
    uw.week_offset,
    MAX(
      CASE
        WHEN p.start_date <= uw.week_end_date
         AND (p.cancellation_date IS NULL OR p.cancellation_date > uw.week_end_date)
```

```
         AND (p.policy_end_date IS NULL OR p.policy_end_date > uw.week_end_date)
        THEN 1 ELSE 0
      END
    ) AS is_active
  FROM unnested_weeks uw
  JOIN `data.policies` p ON uw.user_id = p.user_id
  GROUP BY uw.user_id, uw.cohort_week_start, uw.week_offset
),

-- Calculate cohort sizes (week 0 user counts)
cohort_sizes AS (
  SELECT
    cohort_week_start,
    COUNT(DISTINCT user_id) AS cohort_size
  FROM first_policies
  GROUP BY cohort_week_start
),

-- Final pivot table: retention rate per cohort per week
retention_rates AS (
  SELECT
    ua.cohort_week_start,
    ua.week_offset,
    COUNTIF(ua.is_active = 1) AS retained_users,
    cs.cohort_size,
    ROUND(100 * COUNTIF(ua.is_active = 1) / cs.cohort_size, 2) AS retention_rate
  FROM user_activity ua
  JOIN cohort_sizes cs USING (cohort_week_start)
  GROUP BY ua.cohort_week_start, ua.week_offset, cs.cohort_size
)

-- Pivot weeks as columns (for 12 weeks)
SELECT
  FORMAT_DATE('%Y-%m-%d', cohort_week_start) AS cohort_week_start,
  MAX(IF(week_offset = 0, retention_rate, NULL)) AS week_0,
  MAX(IF(week_offset = 1, retention_rate, NULL)) AS week_1,
  MAX(IF(week_offset = 2, retention_rate, NULL)) AS week_2,
  MAX(IF(week_offset = 3, retention_rate, NULL)) AS week_3,
  MAX(IF(week_offset = 4, retention_rate, NULL)) AS week_4,
  MAX(IF(week_offset = 5, retention_rate, NULL)) AS week_5,
  MAX(IF(week_offset = 6, retention_rate, NULL)) AS week_6,
```

```
  MAX(IF(week_offset = 7, retention_rate, NULL)) AS week_7,
  MAX(IF(week_offset = 8, retention_rate, NULL)) AS week_8,
  MAX(IF(week_offset = 9, retention_rate, NULL)) AS week_9,
  MAX(IF(week_offset = 10, retention_rate, NULL)) AS week_10,
  MAX(IF(week_offset = 11, retention_rate, NULL)) AS week_11
FROM retention_rates
GROUP BY cohort_week_start
ORDER BY cohort_week_start
LIMIT 52;
```

***Weekly Cohort Table***

| cohort_week_start | week_0 | week_1 | week_2 | week_3 | week_4 | week_5 | week_6 | week_7 | week_8 | week_9 | week_10 | week_11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2021-05-03 | 90.91 | 90.91 | 90.91 | 90.91 | 90.91 | 90.91 | 90.91 | 90.91 | 90.91 | 81.82 | 81.82 | 81.82 |
| 2021-05-10 | 92.86 | 92.86 | 92.86 | 89.29 | 89.29 | 85.71 | 85.71 | 82.14 | 78.57 | 78.57 | 82.14 | 82.14 |
| 2021-05-17 | 79.55 | 88.64 | 88.64 | 88.64 | 88.64 | 84.09 | 84.09 | 84.09 | 81.82 | 84.09 | 79.55 | 79.55 |
| 2021-05-24 | 91.67 | 88.89 | 88.89 | 88.89 | 86.11 | 83.33 | 80.56 | 75 | 75 | 72.22 | 72.22 | 75 |
| 2021-05-31 | 90 | 90 | 90 | 90 | 90 | 85 | 80 | 77.5 | 72.5 | 75 | 72.5 | 72.5 |
| 2021-06-07 | 100 | 100 | 100 | 100 | 98.51 | 98.51 | 98.51 | 95.52 | 95.52 | 92.54 | 92.54 | 91.04 |
| 2021-06-14 | 92.06 | 93.65 | 93.65 | 93.65 | 92.06 | 92.06 | 93.65 | 92.06 | 87.3 | 85.71 | 85.71 | 84.13 |
| 2021-06-21 | 91.3 | 91.3 | 92.75 | 92.75 | 92.75 | 89.86 | 88.41 | 88.41 | 86.96 | 85.51 | 85.51 | 84.06 |
| 2021-06-28 | 92.19 | 90.63 | 90.63 | 90.63 | 90.63 | 87.5 | 89.06 | 87.5 | 84.38 | 84.38 | 84.38 | 84.38 |
| 2021-07-05 | 92.54 | 92.54 | 92.54 | 91.79 | 91.79 | 89.55 | 90.3 | 88.06 | 85.82 | 84.33 | 83.58 | 82.09 |
| 2021-07-12 | 93.07 | 93.07 | 93.07 | 93.07 | 94.06 | 93.07 | 93.07 | 89.11 | 87.13 | 87.13 | 86.14 | 84.16 |
| 2021-07-19 | 90.91 | 90.91 | 90.15 | 89.39 | 89.39 | 87.88 | 85.61 | 85.61 | 86.36 | 85.61 | 84.85 | 84.09 |
| 2021- | 94.12 | 94.12 | 93.38 | 93.38 | 92.65 | 91.91 | 91.91 | 90.44 | 90.44 | 89.71 | 89.71 | 90.44 |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 07-26 | | | | | | | | | | | | |
| 2021-08-02 | 96.3 | 95.77 | 94.71 | 94.71 | 94.71 | 93.12 | 91.53 | 89.42 | 86.24 | 85.71 | 84.13 | 83.07 |
| 2021-08-09 | 94.04 | 95.41 | 94.95 | 94.5 | 93.58 | 93.12 | 93.12 | 92.2 | 90.83 | 90.37 | 89.45 | 88.07 |
| 2021-08-16 | 92.15 | 91.62 | 92.15 | 91.62 | 91.1 | 90.05 | 90.58 | 86.39 | 85.86 | 83.77 | 84.29 | 83.77 |
| 2021-08-23 | 91.96 | 90.95 | 90.95 | 90.45 | 89.95 | 87.94 | 87.44 | 85.43 | 84.42 | 83.92 | 82.91 | 82.41 |
| 2021-08-30 | 92.56 | 91.32 | 90.5 | 89.67 | 89.26 | 88.43 | 85.95 | 83.88 | 83.47 | 83.47 | 83.88 | 83.06 |
| 2021-09-06 | 93.33 | 92.59 | 92.59 | 92.96 | 92.96 | 90.37 | 88.89 | 87.78 | 86.67 | 84.81 | 84.81 | 84.07 |
| 2021-09-13 | 95.67 | 95.33 | 95.33 | 94 | 93 | 92 | 92 | 89.33 | 88.67 | 88.33 | 87.67 | 86.33 |
| 2021-09-20 | 93.31 | 95.32 | 94.98 | 93.98 | 92.98 | 92.64 | 91.64 | 88.63 | 86.29 | 85.28 | 84.95 | 83.28 |
| 2021-09-27 | 93.86 | 93.5 | 93.86 | 92.78 | 91.7 | 90.25 | 89.17 | 87.36 | 83.75 | 83.03 | 83.03 | 82.31 |
| 2021-10-04 | 92.05 | 92.88 | 92.6 | 92.33 | 92.05 | 91.51 | 90.41 | 86.3 | 85.48 | 84.66 | 84.11 | 84.11 |
| 2021-10-11 | 94.37 | 93.8 | 94.08 | 93.24 | 92.96 | 91.55 | 91.27 | 87.89 | 86.48 | 85.07 | 83.94 | 82.54 |
| 2021-10-18 | 91.71 | 92.17 | 92.17 | 91.47 | 91.94 | 90.09 | 89.86 | 88.48 | 88.02 | 87.1 | 87.1 | 85.48 |
| 2021-10-25 | 92.35 | 92.35 | 92.1 | 91.11 | 90.86 | 89.14 | 88.4 | 85.19 | 83.95 | 82.22 | 81.98 | 80.49 |
| 2021-11-01 | 91.56 | 92.19 | 92.19 | 91.77 | 90.93 | 89.03 | 88.19 | 86.5 | 86.29 | 85.23 | 85.23 | 83.54 |
| 2021-11-08 | 92.17 | 91.95 | 92.17 | 90.38 | 89.93 | 87.92 | 87.7 | 85.01 | 84.34 | 83.45 | 82.55 | 80.76 |
| 2021-11-15 | 93.16 | 92.97 | 92.97 | 91.99 | 91.6 | 90.23 | 88.48 | 86.13 | 84.38 | 84.18 | 83.79 | 81.64 |
| 2021-11-22 | 92.14 | 92.34 | 91.94 | 91.13 | 90.32 | 88.51 | 87.9 | 85.89 | 85.28 | 84.27 | 84.68 | 84.48 |
| 2021-11-29 | 93.28 | 93.09 | 93.67 | 93.09 | 92.32 | 90.4 | 89.25 | 87.14 | 86.76 | 86.18 | 85.6 | 84.84 |
| 2021-12-06 | 92.07 | 92.62 | 92.44 | 92.62 | 92.25 | 90.41 | 89.48 | 87.82 | 86.35 | 85.42 | 85.24 | 83.21 |
| 2021- | 93.4 | 93.76 | 94.12 | 93.94 | 91.98 | 90.55 | 89.66 | 88.41 | 86.99 | 86.27 | 86.45 | 83.24 |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 12-13 | | | | | | | | | | | | |
| 2021-12-20 | 94 | 94 | 94.53 | 93.65 | 93.3 | 91.53 | 90.83 | 88.18 | 85.71 | 85.36 | 84.13 | 83.25 |
| 2021-12-27 | 92.72 | 93.31 | 93.11 | 92.72 | 92.13 | 91.14 | 91.34 | 89.96 | 87.99 | 87.01 | 87.01 | 85.63 |
| 2022-01-03 | 92.25 | 92.41 | 92.56 | 92.09 | 91.61 | 90.03 | 89.56 | 87.82 | 86.23 | 85.76 | 84.97 | 84.65 |
| 2022-01-10 | 94.1 | 93.61 | 93.61 | 92.46 | 92.46 | 90.82 | 90 | 87.21 | 85.74 | 84.26 | 83.93 | 81.97 |
| 2022-01-17 | 93.9 | 93.32 | 93.32 | 93.18 | 92.74 | 91.15 | 89.84 | 87.66 | 86.21 | 85.49 | 84.91 | 82.44 |
| 2022-01-24 | 92.75 | 93.08 | 92.42 | 91.93 | 91.27 | 88.63 | 88.96 | 87.64 | 86.16 | 84.51 | 84.35 | 81.88 |
| 2022-01-31 | 94.25 | 94.09 | 94.09 | 93.47 | 93.31 | 92.38 | 90.82 | 88.18 | 86.94 | 86.31 | 85.23 | 84.29 |
| 2022-02-07 | 94.98 | 94.12 | 93.97 | 93.69 | 92.68 | 91.25 | 90.39 | 88.52 | 87.09 | 85.37 | 84.65 | 83.79 |
| 2022-02-14 | 93.8 | 93.8 | 93.8 | 93.32 | 92.85 | 91.89 | 90.78 | 89.19 | 89.03 | 87.6 | 86.8 | 85.69 |
| 2022-02-21 | 94.34 | 94.18 | 94.5 | 94.18 | 93.55 | 91.98 | 90.09 | 88.05 | 86.48 | 85.22 | 84.59 | 83.18 |
| 2022-02-28 | 94.47 | 94.72 | 94.6 | 94.72 | 94.22 | 92.46 | 91.58 | 89.57 | 87.56 | 86.43 | 86.06 | 84.05 |
| 2022-03-07 | 95.31 | 95.44 | 95.18 | 94.92 | 94.66 | 93.49 | 92.58 | 90.76 | 89.58 | 88.67 | 88.8 | 87.37 |
| 2022-03-14 | 94.65 | 94.14 | 93.63 | 93.76 | 92.74 | 91.08 | 89.3 | 87.39 | 86.24 | 85.35 | 84.71 | 82.42 |
| 2022-03-21 | 95.13 | 95.13 | 94.2 | 93.68 | 93.02 | 92.36 | 91.17 | 88.01 | 86.56 | 85.51 | 84.45 | 84.19 |
| 2022-03-28 | 95.64 | 95.22 | 95.22 | 94.8 | 93.81 | 92.55 | 91.56 | 89.31 | 87.76 | 86.5 | 85.94 | 83.97 |
| 2022-04-04 | 94.74 | 94.63 | 94.41 | 94.07 | 93.62 | 91.16 | 90.83 | 88.59 | 87.14 | 86.24 | 86.02 | 84.12 |
| 2022-04-11 | 93.95 | 94.32 | 94.2 | 94.32 | 93.95 | 92.22 | 90.12 | 88.27 | 87.28 | 86.17 | 86.17 | 84.94 |
| 2022-04-18 | 95.17 | 94.93 | 94.81 | 94.22 | 93.63 | 92.69 | 91.75 | 88.92 | 87.85 | 86.79 | 86.79 | 85.38 |
| 2022-04-25 | 94.84 | 94.84 | 94.72 | 94.24 | 93.76 | 92.09 | 91.85 | 89.33 | 88.25 | 86.81 | 86.93 | 86.09 |

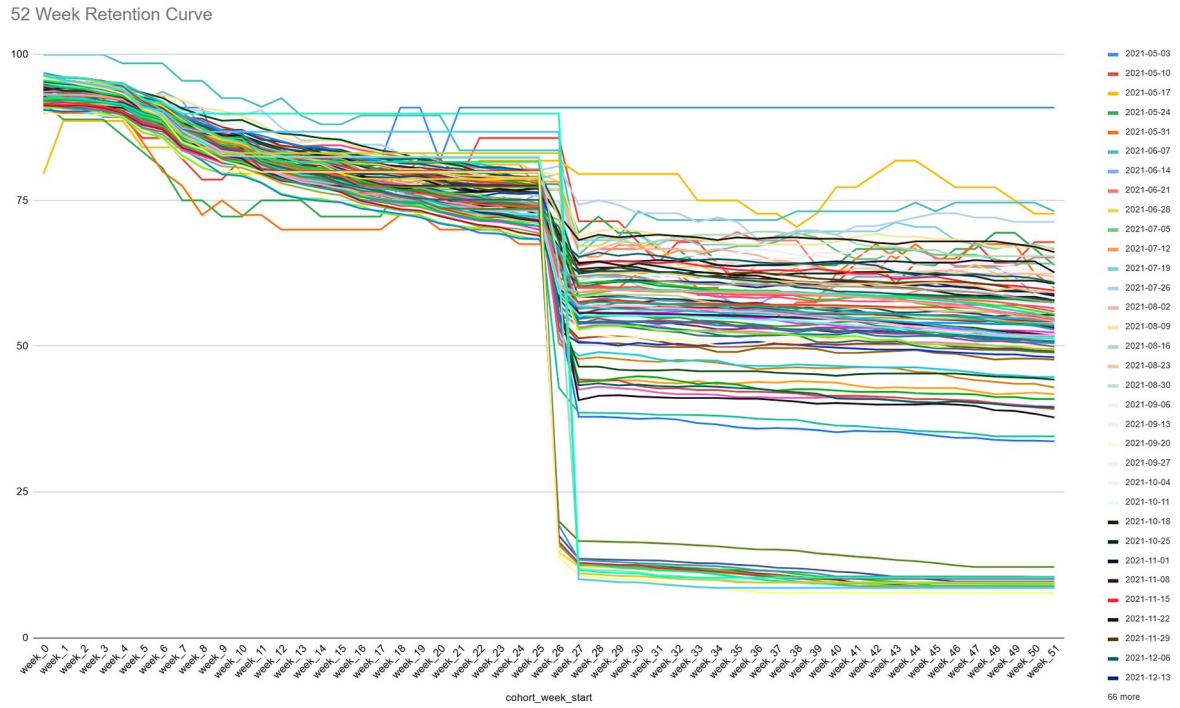
**Question 4. Build out the cohort retention table to have at least 52 columns. Are there any dips in the retention curve that you find concerning? If so, what do you think might be happening from a customer/business perspective?**

```
-- Get each user's first policy start date (week 0)
WITH first_policies AS (
  SELECT
    user_id,
    MIN(start_date) AS first_policy_start,
    DATE_TRUNC(MIN(start_date), WEEK(MONDAY)) AS cohort_week_start
  FROM `data.policies`
  GROUP BY user_id
),

-- Generate 52 weeks from each user's first policy date (dim_date)
user_weeks AS (
  SELECT
    fp.user_id,
    fp.first_policy_start,
    fp.cohort_week_start,
    GENERATE_ARRAY(0, 51) AS week_offsets
  FROM first_policies fp
),

-- Expand user-weeks
unnested_weeks AS (
  SELECT
    uw.user_id,
    uw.cohort_week_start,
    week_offset,
    DATE_ADD(uw.first_policy_start, INTERVAL week_offset WEEK) AS week_end_date
  FROM user_weeks uw, UNNEST(week_offsets) AS week_offset
),

-- Check whether user was active at end of each week
user_activity AS (
  SELECT
    uw.user_id,
    uw.cohort_week_start,
    uw.week_offset,
    MAX(
```

```
      CASE
        WHEN p.start_date <= uw.week_end_date
         AND (p.policy_end_date IS NULL OR p.policy_end_date > uw.week_end_date)
         AND (p.cancellation_date IS NULL OR p.cancellation_date > uw.week_end_date)
        THEN 1 ELSE 0
      END
    ) AS is_active
  FROM unnested_weeks uw
  JOIN `data.policies` p ON uw.user_id = p.user_id
  GROUP BY uw.user_id, uw.cohort_week_start, uw.week_offset
),

-- Cohort sizes
cohort_sizes AS (
  SELECT
    cohort_week_start,
    COUNT(DISTINCT user_id) AS cohort_size
  FROM first_policies
  GROUP BY cohort_week_start
),

-- Retention rates
retention_rates AS (
  SELECT
    ua.cohort_week_start,
    ua.week_offset,
    COUNTIF(ua.is_active = 1) AS retained_users,
    cs.cohort_size,
    ROUND(100 * COUNTIF(ua.is_active = 1) / cs.cohort_size, 2) AS retention_rate
  FROM user_activity ua
  JOIN cohort_sizes cs USING (cohort_week_start)
  GROUP BY ua.cohort_week_start, ua.week_offset, cs.cohort_size
)

-- Pivot retention rates across 52 weeks
SELECT
  FORMAT_DATE('%Y-%m-%d', cohort_week_start) AS cohort_week_start,
  MAX(IF(week_offset = 0,  retention_rate, NULL)) AS week_0,
  MAX(IF(week_offset = 1,  retention_rate, NULL)) AS week_1,
  MAX(IF(week_offset = 2,  retention_rate, NULL)) AS week_2,
  MAX(IF(week_offset = 3,  retention_rate, NULL)) AS week_3,
```

```
MAX(IF(week_offset = 4,  retention_rate, NULL)) AS week_4,
MAX(IF(week_offset = 5,  retention_rate, NULL)) AS week_5,
MAX(IF(week_offset = 6,  retention_rate, NULL)) AS week_6,
MAX(IF(week_offset = 7,  retention_rate, NULL)) AS week_7,
MAX(IF(week_offset = 8,  retention_rate, NULL)) AS week_8,
MAX(IF(week_offset = 9,  retention_rate, NULL)) AS week_9,
MAX(IF(week_offset = 10, retention_rate, NULL)) AS week_10,
MAX(IF(week_offset = 11, retention_rate, NULL)) AS week_11,
MAX(IF(week_offset = 12, retention_rate, NULL)) AS week_12,
MAX(IF(week_offset = 13, retention_rate, NULL)) AS week_13,
MAX(IF(week_offset = 14, retention_rate, NULL)) AS week_14,
MAX(IF(week_offset = 15, retention_rate, NULL)) AS week_15,
MAX(IF(week_offset = 16, retention_rate, NULL)) AS week_16,
MAX(IF(week_offset = 17, retention_rate, NULL)) AS week_17,
MAX(IF(week_offset = 18, retention_rate, NULL)) AS week_18,
MAX(IF(week_offset = 19, retention_rate, NULL)) AS week_19,
MAX(IF(week_offset = 20, retention_rate, NULL)) AS week_20,
MAX(IF(week_offset = 21, retention_rate, NULL)) AS week_21,
MAX(IF(week_offset = 22, retention_rate, NULL)) AS week_22,
MAX(IF(week_offset = 23, retention_rate, NULL)) AS week_23,
MAX(IF(week_offset = 24, retention_rate, NULL)) AS week_24,
MAX(IF(week_offset = 25, retention_rate, NULL)) AS week_25,
MAX(IF(week_offset = 26, retention_rate, NULL)) AS week_26,
MAX(IF(week_offset = 27, retention_rate, NULL)) AS week_27,
MAX(IF(week_offset = 28, retention_rate, NULL)) AS week_28,
MAX(IF(week_offset = 29, retention_rate, NULL)) AS week_29,
MAX(IF(week_offset = 30, retention_rate, NULL)) AS week_30,
MAX(IF(week_offset = 31, retention_rate, NULL)) AS week_31,
MAX(IF(week_offset = 32, retention_rate, NULL)) AS week_32,
MAX(IF(week_offset = 33, retention_rate, NULL)) AS week_33,
MAX(IF(week_offset = 34, retention_rate, NULL)) AS week_34,
MAX(IF(week_offset = 35, retention_rate, NULL)) AS week_35,
MAX(IF(week_offset = 36, retention_rate, NULL)) AS week_36,
MAX(IF(week_offset = 37, retention_rate, NULL)) AS week_37,
MAX(IF(week_offset = 38, retention_rate, NULL)) AS week_38,
MAX(IF(week_offset = 39, retention_rate, NULL)) AS week_39,
MAX(IF(week_offset = 40, retention_rate, NULL)) AS week_40,
MAX(IF(week_offset = 41, retention_rate, NULL)) AS week_41,
MAX(IF(week_offset = 42, retention_rate, NULL)) AS week_42,
MAX(IF(week_offset = 43, retention_rate, NULL)) AS week_43,
MAX(IF(week_offset = 44, retention_rate, NULL)) AS week_44,
```

```
  MAX(IF(week_offset = 45, retention_rate, NULL)) AS week_45,
  MAX(IF(week_offset = 46, retention_rate, NULL)) AS week_46,
  MAX(IF(week_offset = 47, retention_rate, NULL)) AS week_47,
  MAX(IF(week_offset = 48, retention_rate, NULL)) AS week_48,
  MAX(IF(week_offset = 49, retention_rate, NULL)) AS week_49,
  MAX(IF(week_offset = 50, retention_rate, NULL)) AS week_50,
  MAX(IF(week_offset = 51, retention_rate, NULL)) AS week_51
FROM retention_rates
GROUP BY cohort_week_start
ORDER BY cohort_week_start;
```




For Q1 2023 (and to a lesser extent, Q1 2022), there is a significant drop in retention (~ -80 pp). This looks like seasonality since this pattern occurs across multiple cohorts, where the cohort start date is in Q1. From a business perspective, this is probably due to people looking at their subscriptions at the beginning and end of each year and cancelling.

—

## PART III (recommended time: 1.5 hours)

**Question 5. What are the top 3 factors that meaningfully impact 7 month retention (retention at week 31)?**

● You may use any or all of the data provided to you.
● Please clearly explain your results and methodology. Include all SQL queries you used to in your answer to this question, as well as any relevant data visualizations.
● You may use any statistical or exploratory data analysis techniques, as long as you clearly explain your methodology and why you chose it.

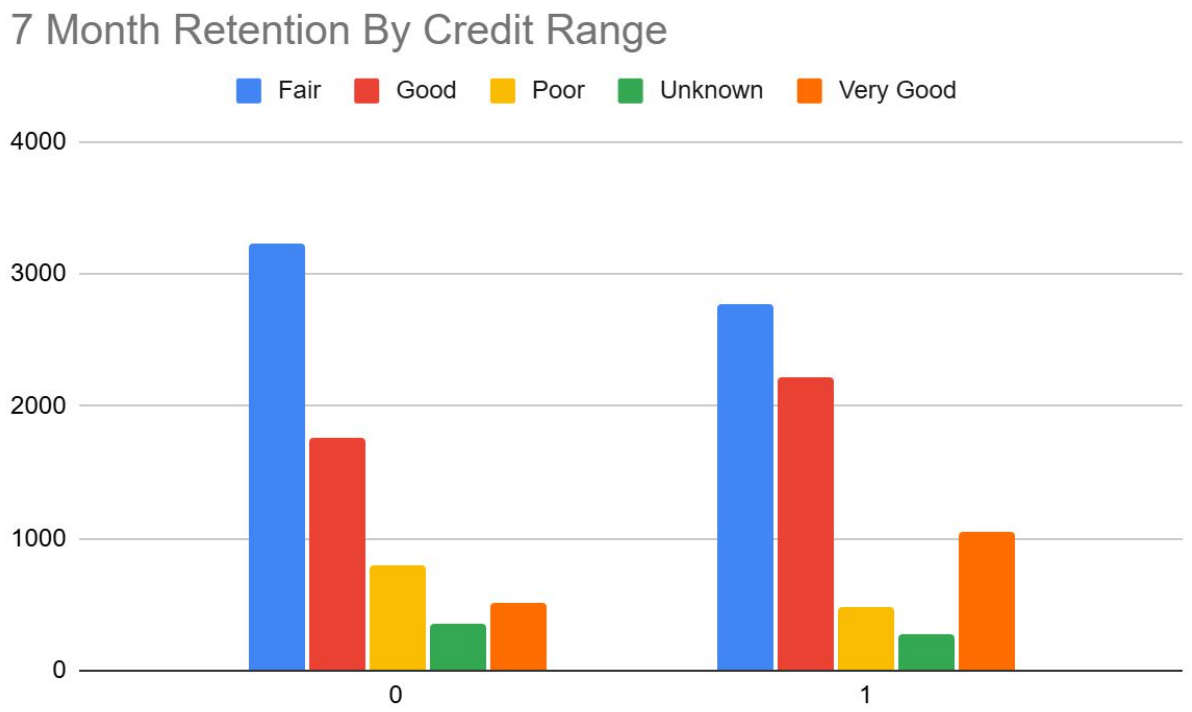
**Question 6. For each factor you identified as impactful to retention in Question 5, please provide your business reasoning for why it matters to retention. It may help to use the framing "Customers who [have x characteristic] are more likely to retain because..."**

```sql
-- Build data for retention analysis
-- Get start date
WITH first_policies AS (
  SELECT
    user_id,
    MIN(start_date) AS first_policy_start
  FROM `data.policies`
  GROUP BY user_id
),

-- Filter to just users with active policy at week 31
eligible_users AS (
  SELECT
    user_id,
    first_policy_start,
    DATE_ADD(first_policy_start, INTERVAL 31 WEEK) AS week_31_date
  FROM first_policies
  WHERE first_policy_start <= DATE_SUB(CURRENT_DATE(), INTERVAL 31 WEEK)
),

-- Label users as retained or not retained (0 or 1)
retention_flags AS (
  SELECT
    e.user_id,
    MAX(
      CASE
        WHEN p.start_date <= e.week_31_date
         AND (p.policy_end_date IS NULL OR p.policy_end_date > e.week_31_date)
         AND (p.cancellation_date IS NULL OR p.cancellation_date > e.week_31_date)
        THEN 1 ELSE 0
      END
    ) AS retained_flag
```

```
  FROM eligible_users e
  JOIN `data.policies` p ON e.user_id = p.user_id
  GROUP BY e.user_id
)

-- Pull features to analyze retention
SELECT
  r.user_id,
  r.retained_flag,
  u.age_group,
  u.credit_range,
  u.has_claims,
  u.has_violations,
  u.currently_insured,
  d.marital_status,
  d.education,
  d.occupation
FROM retention_flags r
LEFT JOIN `data.users` u ON r.user_id = u.user_id
LEFT JOIN `data.drivers` d ON r.user_id = d.user_id;
```

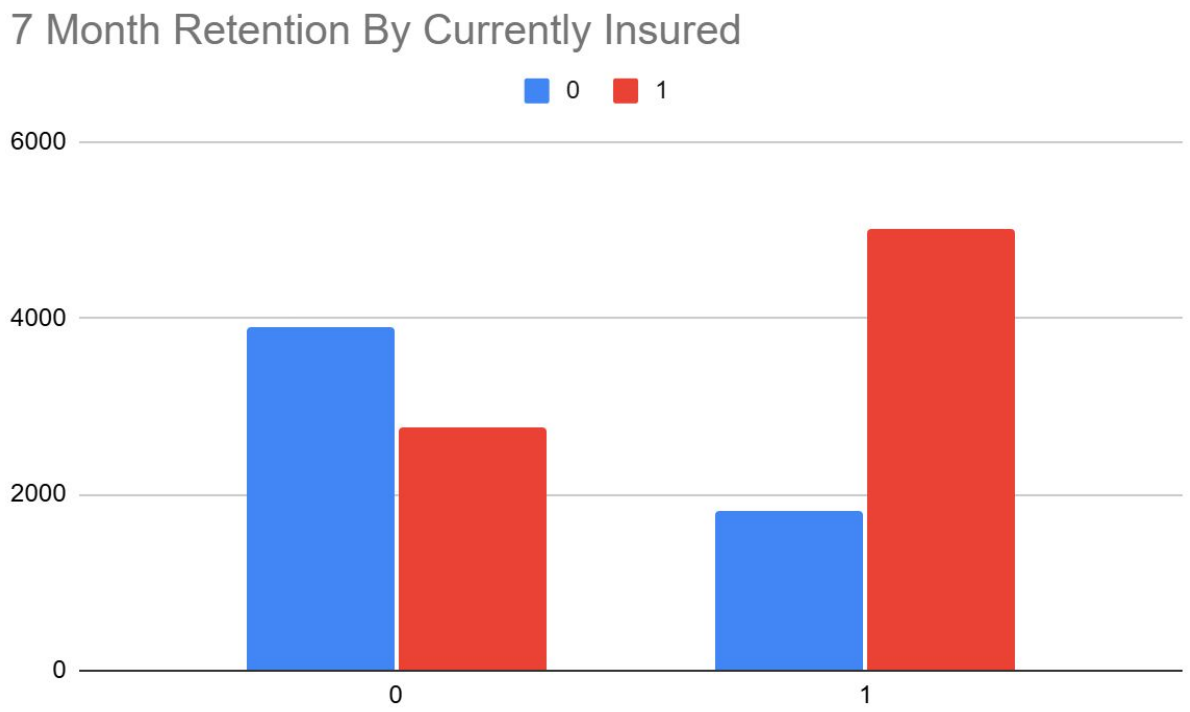
### Top 3 Factors

### 1. Credit Range



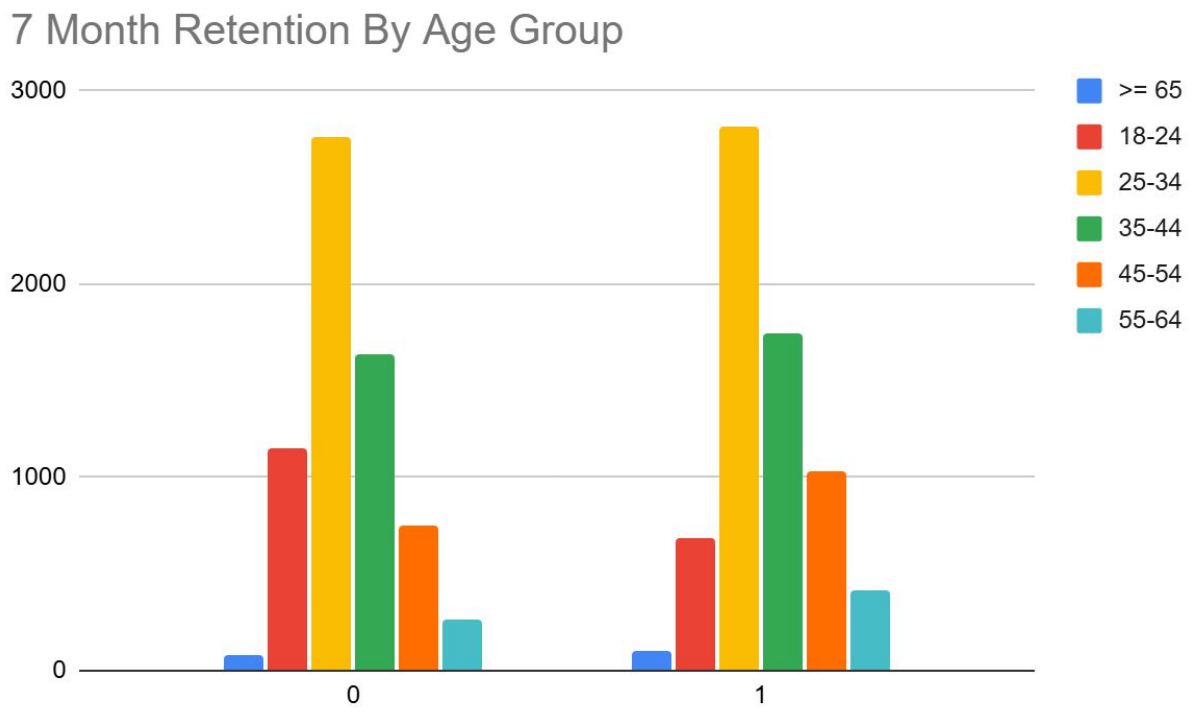

- Customers with higher credit ranges had meaningfully higher retention (Good and Very Good credit). After running a chi-square test, the p-value is way smaller than .05, meaning that **there is a statistically significant relationship between credit range and retention rate.**
    - Very good credit customers can be more financially savvy and value insurance more and are more likely to 1) be able to afford premiums 2) value and prioritize insurance
        - In other words, customers with higher credit are more likely to retain because they are generally more financially stable, value consistency, and may qualify for better long-term policy rates
    - Retention lift: ~20% between lowest and highest credit tiers (Very Good: 67%, Fair: 46%)

**2. Currently Insured at Signup**




- Customers who are currently insured had meaningfully higher retention. After running a chi-square test, the p-value is way smaller than .05, meaning that **there is a statistically significant relationship between current coverage and retention rate.**
    - Customers who had prior insurance showed higher 7-month retention (65% vs. 32%)
    - These users likely understand the importance of continuous coverage and are less likely to drop policies

**3. Age Group**




- Older age groups (especially 45+) had higher retention vs. 18-24, which had the lowest retention rate. After running a chi-square test, the p-value is way smaller than .05, meaning that **there is a statistically significant relationship between age group and retention rate.**
    - Possibly more stable in financial habits and less price-sensitive
    - Customers in older age groups are more likely to retain because they often have more stable driving records, greater life stability, and may prioritize safety and service over premium price

**Seasonality Note**: Retention rates drop drastically in weeks 26-28 for cohorts with start dates the last week of the year or Q1 of the year. This is most likely due to people reviewing their subscriptions and insurance policies this time of the year and cancelling. **Would recommend seasonal promotions and cancellation flow pop-ups to incentive user retention.**

**Question 7. Based on the answers you've provided so far:**

**A. Please recommend 1 new analysis to conduct. What additional data would you like to see in order to better understand the drivers of customer retention?**

**New analysis:** A/B test retention outcomes when adjusting the following criteria on all available user attributes (credit, claims, age, etc.). Additional useful data (to conduct aforementioned tests on) could include:

- Premium size
- Payment/billing method
- Contact frequency with customer service (# of service tickets)
- Policy changes within the first 3 months

**Other analyses:** After doing this, may be worth completing the data for marital status, occupation, and other demographic fields to identify other product offerings.

**B. Please recommend 1 product feature idea to test out that might improve retention in our customer base. Why do you think it will have an impact?**

I recommend experimenting and launching a mid-policy engagement campaign personalized check-ins or policy reviews around week 25). This would help preempt cancellations, especially for Q1 cohorts, by reminding users of value, offering re-quotes, or suggesting bundle savings. This could improve retention where seasonal dips occur too. In other words, I am suggesting a customer lifecycle marketing campaign to ensure engagement at the critical inflection point in retaining and churning a customer.