Lion Studios

Data Science & Analytics

Data Analyst Exercise

|  |  |
| --- | --- |
| Candidate Name | Iuliia Gvarianashvili |
| E-mail | dvortsovajula@gmail.com |
| Submitted On | 24 august 2025 |
| Github Repository | <https://github.com/AppleJul/test-lionstudios> |

**Instructions**

1. There are two sections in this exercise: SQL and Experimental Analysis
2. For the SQL section:
   1. Write your answers in this document in the space provided below each question.
   2. There are multiple possible answers to these questions. Responses will be evaluated on the following dimensions (in order of importance): accuracy, efficiency, readability.
3. For the Experimental Analysis section:
   1. Write your answers to the Pre-Experiment questions in this document in the space provided below each question.
   2. Perform your analysis using the included dataset lionstudios\_data\_analyst\_take\_home.csv in Python. Your analysis should be provided as a Jupyter Notebook uploaded to a public GitHub repository. **Please include a link to the public GitHub repository at the top of this document.**
   3. Write your answers to the Post-Experiment questions in your Jupyter Notebook accompanying the supporting analysis, statistical tests and visualizations.
   4. Responses to the conceptual questions will be evaluated on the following dimensions (in order of importance): appropriateness, practicality, actionability, insightfulness, creativity, comprehensiveness, clarity, concision
   5. Your analysis will be evaluated on the following dimensions (in order of importance): data intuition, storytelling, comprehensiveness, creativity, understanding and appropriateness of statistical tests, programming ability, code readability
4. The in-person technical interview is a combination of discussing past projects and questions related to responses to this take-home exercise.

Notes:  
  
Any additional information you would like to include not related to a specific question should go here.

**SQL**

For the following questions, use the two tables defined below (using postgres, presto, or snowflake vocabulary/grammar):

1. users
   1. For users who have installed multiple games, they have the same user\_id across all games therefore a single user\_id can appear multiple times in the users table
   2. Unique values for os are android and ios
   3. first\_iap\_timestamp is NULL for users who have never made an IAP

|  |  |  |
| --- | --- | --- |
| **Column** | **Datatype** | **Example Data** |
| user\_id | varchar | 1a2bc3-4d5e6f |
| game | varchar | Match 3D |
| os | varchar | android |
| install\_timestamp | timestamp\_ntz | 2021-12-08 13:14:23.000 |
| first\_iap\_timestamp | timestamp\_ntz | 2021-12-08 13:14:23.000 |

1. sessions
   1. user\_id, game and os are the same as *users*
   2. session\_count is a counter that increments every time a user starts a new session
   3. session\_iap\_revenue is NULL when the user did not make an IAP during the session

|  |  |  |
| --- | --- | --- |
| **Column** | **Datatype** | **Example Data** |
| user\_id | varchar | 1a2bc3-4d5e6f |
| game | varchar | Match 3D |
| os | varchar | android |
| session\_count | integer | 3 |
| session\_start\_timestamp | timestamp\_ntz | 2021-12-08 13:14:23.000 |
| session\_end\_timestamp | timestamp\_ntz | 2021-12-08 13:14:23.000 |
| session\_iap\_revenue | float | 9.99 |

**Definitions**

* IAP - in app purchase with real currency
* customer - user who has made an IAP
* active user - user who had a session on a given day

1. **For Puzzle Wars: Heroes sessions between 2021-11-29 and 2021-12-05 (inclusive) what was the daily percentage of active users that became new customers?**

To calculate the daily share of active users who became new customers in the game *Puzzle Wars: Heroes* for the period from 2021-11-29 to 2021-12-05, we followed these steps:

1. Identified all users who were active on each day (i.e., had at least one session that day).
2. Identified all users whose first IAP occurred on the same date as their session.
3. Considered a user a new customer on a given day only if: they had a session on that day, their first IAP occurred on that same day, and both actions happened on the **same platform** (iOS or Android).
4. Calculated the share of new customers out of all active users per day:

- Overall (both platforms combined)

* Separately for Android and iOS, since user behavior and monetization patterns can differ significantly between platforms

Daily results allow us to track trends over time and identify sudden drops. This level of granularity is especially useful for A/B testing, time-series analysis. If needed, we could additionally compute simple daily average across the period or weighted average based on the number of active users per day. But for this analysis, we chose to report daily values for maximum visibility and decision-making precision.

**SQL query (postgres):**

WITH daily\_active\_users AS (

SELECT

DATE(session\_start\_timestamp) AS activity\_date,

user\_id,

os

FROM sessions

WHERE game = 'Puzzle Wars: Heroes'

AND session\_start\_timestamp >= DATE '2021-11-29'

AND session\_start\_timestamp < DATE '2021-12-06'

GROUP BY activity\_date, user\_id, os

),

daily\_first\_time\_purchasers AS (

SELECT

DATE(first\_iap\_timestamp) AS first\_purchase\_date,

user\_id,

os

FROM users

WHERE game = 'Puzzle Wars: Heroes'

AND first\_iap\_timestamp IS NOT NULL

AND first\_iap\_timestamp >= DATE '2021-11-29'

AND first\_iap\_timestamp < DATE '2021-12-06'

GROUP BY first\_purchase\_date, user\_id, os

),

daily\_user\_metrics AS (

SELECT

daily\_active\_users.activity\_date,

daily\_active\_users.os,

COUNT(DISTINCT daily\_active\_users.user\_id) AS active\_users\_count,

COUNT(DISTINCT daily\_first\_time\_purchasers.user\_id) AS new\_purchasers\_count

FROM daily\_active\_users

LEFT JOIN daily\_first\_time\_purchasers

ON daily\_active\_users.user\_id = daily\_first\_time\_purchasers.user\_id

AND daily\_active\_users.activity\_date = daily\_first\_time\_purchasers.first\_purchase\_date

AND daily\_active\_users.os = daily\_first\_time\_purchasers.os

GROUP BY daily\_active\_users.activity\_date, daily\_active\_users.os

)

SELECT

activity\_date,

-- Total share of new customers

(SUM(new\_purchasers\_count) \* 100.0 / NULLIF(SUM(active\_users\_count), 0))

AS share\_of\_new\_purchasers\_all,

-- Share Android

(SUM(new\_purchasers\_count) FILTER (WHERE os = 'android') \* 100.0

/ NULLIF(SUM(active\_users\_count) FILTER (WHERE os = 'android'), 0))

AS share\_of\_new\_purchasers\_android,

-- Share iOS

(SUM(new\_purchasers\_count) FILTER (WHERE os = 'ios') \* 100.0

/ NULLIF(SUM(active\_users\_count) FILTER (WHERE os = 'ios'), 0))

AS share\_of\_new\_purchasers\_ios

FROM daily\_user\_metrics

GROUP BY activity\_date

ORDER BY activity\_date;

1. **For users who installed the game Tile Garden, for what installation weeks was the *average IAP revenue from users who made their first IAP within 24 hours of installation* higher for ios than android (revenue on the day of the first IAP only)?**

To compare iOS and Android monetization performance in the game *Tile Garden*, we analyzed the average revenue generated on the first IAP day for users who made a purchase within 24 hours of installing the game. We followed these steps:

Identified all users who installed the game and completed their first IAP within 24 hours of installation.

For these users, calculated the total IAP revenue from sessions that occurred on the same day as the first IAP.

Grouped the users by install week (starting from Monday) and by platform (iOS vs. Android).

Calculated the average revenue on first IAP day per user for each install week and platform.

Reported only the install weeks where iOS had a higher average revenue than Android.

**SQL query (postgres):**

WITH tile\_garden\_users AS (

SELECT

user\_id,

os,

install\_timestamp,

first\_iap\_timestamp,

DATE\_TRUNC('week', install\_timestamp + INTERVAL '1 day') AS week

FROM users

WHERE game = 'Tile Garden'

AND first\_iap\_timestamp IS NOT NULL

AND first\_iap\_timestamp <= install\_timestamp + INTERVAL '24 hours'

),

revenue\_on\_first\_iap\_day AS (

SELECT

users.user\_id,

users.os,

users.week,

SUM(sessions.session\_iap\_revenue) AS iap\_revenue\_first\_day

FROM tile\_garden\_users AS users

JOIN sessions

ON users.user\_id = sessions.user\_id

AND sessions.game = 'Tile Garden'

AND DATE(sessions.session\_start\_timestamp) = DATE(users.first\_iap\_timestamp)

WHERE sessions.session\_iap\_revenue IS NOT NULL

GROUP BY users.user\_id, users.os, users.week

),

average\_revenue\_by\_week\_platform AS (

SELECT

week,

os,

AVG(iap\_revenue\_first\_day) AS avg\_revenue

FROM revenue\_on\_first\_iap\_day

GROUP BY week, os

)

SELECT

week,

MAX(CASE WHEN os = 'ios' THEN avg\_revenue END) AS ios,

MAX(CASE WHEN os = 'android' THEN avg\_revenue END) AS android

FROM average\_revenue\_by\_week\_platform

GROUP BY week

HAVING MAX(CASE WHEN os = 'ios' THEN avg\_revenue END) > MAX(CASE WHEN os = 'android' THEN avg\_revenue END)

ORDER BY week;

1. **For users who installed Match 3D between 2021-11-21 and 2021-11-27 before installing Merge Life between 2021-11-28 and 2021-12-04, what *percentage of users spent more on IAPs in their first seven days in* Match 3D *than* Merge Life? Exclude users who did not make an IAP in either game.**

To calculate the percentage of users who spent more on IAPs in the first 7 days after installing Match 3D than Merge Life we:

1. Identified users who installed both games - Match 3D installed between 2021-11-21 and 2021-11-27 and Merge Life installed between 2021-11-28 and 2021-12-04. Only users who installed Match 3D *before* Merge Life were included.

2. Calculated IAP revenue separately for each game

3. We used INNER JOINs to ensure that users without revenue in either game were excluded

**SQL query (postgres):**

-- *users who installed Match 3D within the target date range*

WITH users\_installed\_match3d AS (

SELECT

user\_id,

install\_timestamp AS match3d\_install\_date

FROM users

WHERE game = 'Match 3D'

AND install\_timestamp::DATE BETWEEN '2021-11-21' AND '2021-11-27'

),

-- *users who installed Merge Life in the following week*

users\_installed\_mergelife AS (

SELECT

user\_id,

install\_timestamp AS mergelife\_install\_date

FROM users

WHERE game = 'Merge Life'

AND install\_timestamp::DATE BETWEEN '2021-11-28' AND '2021-12-04'

),

-- *only users who installed Match 3D before Merge Life*

users\_with\_both\_games\_installed\_in\_order AS (

SELECT

match3d.user\_id,

match3d.match3d\_install\_date,

mergelife.mergelife\_install\_date

FROM users\_installed\_match3d AS match3d

JOIN users\_installed\_mergelife AS mergelife

ON match3d.user\_id = mergelife.user\_id

WHERE match3d.match3d\_install\_date < mergelife.mergelife\_install\_date

),

-- *IAP revenue in the first 7 days after installing Match 3D*

revenue\_first\_7\_days\_match3d AS (

SELECT

sessions.user\_id,

SUM(sessions.session\_iap\_revenue) AS total\_revenue\_match3d

FROM sessions

JOIN users\_with\_both\_games\_installed\_in\_order AS qualified\_users

ON sessions.user\_id = qualified\_users.user\_id

WHERE sessions.game = 'Match 3D'

AND sessions.session\_start\_timestamp BETWEEN qualified\_users.match3d\_install\_date

AND qualified\_users.match3d\_install\_date + INTERVAL '7 days'

AND sessions.session\_iap\_revenue > 0

GROUP BY sessions.user\_id

),

-- *IAP revenue in the first 7 days after installing Merge Life*

revenue\_first\_7\_days\_mergelife AS (

SELECT

sessions.user\_id,

SUM(sessions.session\_iap\_revenue) AS total\_revenue\_mergelife

FROM sessions

JOIN users\_with\_both\_games\_installed\_in\_order AS qualified\_users

ON sessions.user\_id = qualified\_users.user\_id

WHERE sessions.game = 'Merge Life'

AND sessions.session\_start\_timestamp BETWEEN qualified\_users.mergelife\_install\_date

AND qualified\_users.mergelife\_install\_date + INTERVAL '7 days'

AND sessions.session\_iap\_revenue > 0

GROUP BY sessions.user\_id

),

-- *combine both revenues per user*

combined\_revenue\_per\_user AS (

SELECT

qualified\_users.user\_id,

COALESCE(revenue\_match3d.total\_revenue\_match3d, 0) AS total\_revenue\_match3d,

COALESCE(revenue\_mergelife.total\_revenue\_mergelife, 0) AS total\_revenue\_mergelife

FROM users\_with\_both\_games\_installed\_in\_order AS qualified\_users

INNER JOIN revenue\_first\_7\_days\_match3d AS revenue\_match3d

ON qualified\_users.user\_id = revenue\_match3d.user\_id

INNER JOIN revenue\_first\_7\_days\_mergelife AS revenue\_mergelife

ON qualified\_users.user\_id = revenue\_mergelife.user\_id

)

-- *final result percentage of users who spent more on Match 3D*

SELECT

ROUND(

COUNT(\*) FILTER (

WHERE total\_revenue\_match3d > total\_revenue\_mergelife

) \* 100.0 / COUNT(\*),

2

) AS percentage\_users

FROM combined\_revenue\_per\_user;

**Experimental Analysis**

The Product team for Match 3D would like to run an A/B test to evaluate their new design for the shop button icon. They believe their new icon will increase entrances into the shop by at least 10% and would like to expose 20% of new users to their new design. Right now, there are approximately 5,000 new users a day and about 5% of them click the shop button at least once. Descriptions of each of the columns in the dataset (lionstudios\_product\_analyst\_take\_home.csv)are provided on the last page of this document. Please answer the following questions related to Product’s A/B test proposal.

**Pre-Experiment***Answer here*

1. What questions do you ask the Product team about their experiment?

* Show me how exactly we changed the button. What was the old design and what is the new one?
* Are we changing the button on all devices? Or only for Android/IOS?
* Is there a prototype to look at before starting A/B testing?
* Why did you decide to change this particular button?
* What are the key metrics for this experiment other than entrances into the shop?
* Why are they counting on an increase of exactly 10%?
* Which users do we definitely want to exclude from the test?
* Who will make the final decision on the test results? Who will communicate with?
* When will everything be ready to start testing? changes have been made to the code, the tester has checked that everything works.

1. What experimental design do you recommend to the Product team to investigate their hypothesis?

I recommend a Randomized Controlled Trial (RCT) with the following parameters:

Design:

- 20% of new users included in the experiment (1,000 users/day)

- 50/50 split: Control (current icon) vs. Treatment (new icon)

- Randomization at the user level upon first app launch

Duration: 5–6 weeks to reach 15,600 users per group

Primary Metric: Shop entry rate – percentage of users who clicked the Shop button at least once

Success Criteria: ≥10% relative increase (5% to 5.5%)

Statistical significance: *p* < 0.05, Power: 80%

Risk Mitigation: Staged rollout (start with 5% exposure in the first week)

Daily monitoring dashboard

Automatic stop if key metrics deteriorateм

1. What other hypotheses do you have?

1. The new shop icon could increase the number of purchases per user.

2. Higher engagement with the shop might lead to higher revenue per user.

3. A more visible icon might improve user retention over time.

4. Increased engagement could result in longer average session lengths.

5. The new icon might create a short-term novelty effect but hurt long-term metrics.

1. Do you have any concerns about Product’s proposal?

1. Small Effect Size Detection - Detecting a 10% relative increase (5% → 5.5% absolute) requires very large sample sizes and may result in inconclusive results if the true effect is smaller.

2. Limited Scope - Testing only on new users (20% subset) may not represent the full user base behavior and could miss important impacts on existing players.

3. Sample Size - The proposed test will require 5-6 weeks to achieve statistical significance, which is unusually long for a simple UI change and increases risk of external confounding factors.

4. Metrics - The proposal only considers store entry as a target metric. This does not necessarily reflect the real value of the change.

1. Who do you recommend the Product team target for the experiment?

* new users who just installed the game Match 3D - based on the hypothesis
* segment users by platforms - iOS and Android. There may be a difference in behavior.

1. How long do you recommend running the experiment?

Under the given conditions I recommend running the experiment for approximately **31 days**.

**conditions:**

Baseline conversion rate: 5% (click shop button)

Expected effect: +10%

Daily new users: 5 000

Test exposure: 20% users in test group

Allocation: 50/50

Each group requires 15 600 users

Total sample size: 31 200 users

Duration definition:

Daily test participants:

20% of 5 000 = 1 000 users per day are included in the test

Control group: 500/day

Treatment group: 500/day

Time to reach the desired sample - 15 600 users ÷ 500 users/day = 31 days for each group

**Post-Experiment**

*Answer in your Jupyter Notebook*

1. For all KPIs evaluated: describe your choices in evaluating them
2. What other data about users and their behaviour would you have also liked to have to evaluate this AB test?
3. For all statistical tests performed: describe those statistical tests, their assumptions, and why they are the appropriate choice
4. What was the outcome of the experiment?
5. How confident are you in this result? Why?
6. What are your recommendations for Product as a result of this outcome?

**Dataset: lionstudios\_data\_analyst\_take\_home.csv**

Each row in the dataset represents a user session therefore there can be more than one row per user.

|  |  |
| --- | --- |
| **Column** | **Description** |
| user\_id | Alphanumeric string unique to each user |
| country\_type | US if the user is in the United States, otherwise Non-US |
| platform | android for users on Android devices; ios for users on iOS devices |
| install\_timestamp | The timestamp (UTC) of when the user installed the game |
| session\_timestamp | The timestamp (UTC) at the start of a user’s session |
| icon\_shop\_opens | The number of times during the session the user opened the Cash Shop by pressing the Shop icon |
| inter\_shop\_opens | The number of times during the session the user opened the Cash Shop by clicking on an interstitial (popup) |
| icon\_cash\_purchases | The number of purchases in the Cash Shop during the session the user made after arriving in the shop by pressing the Shop icon |
| inter\_cash\_purchases | The number of purchases in the Cash Shop during the session the user made after arriving in the shop by clicking on an interstitial |
| icon\_revenue\_usd | The total revenue from the Cash Shop earned during the user’s session from purchases made after arriving in the shop by pressing the Shop icon |
| inter\_revenue\_usd | The total revenue from the Cash Shop earned during the user’s session from purchases made after arriving in the shop by clicking on an interstitial |
| session\_length | The length of the user’s session in minutes |
| rvs\_watched | The number of rewarded videos, which are video ads that earn user’s in-game currency and items if they watch them completely, user’s watched during their session |
| levels\_played | The number of levels the user played during their session. This includes all attempts so the number of levels played does not necessarily equal the number of levels gained by the user during the session |