Great Plants XP: “Sign-Up”

Last updated: December 14, 2023, Author: Vriddhi Misra

Summary *(Written at the end of the experiment)*

- A new signup banner will be displayed prominently on the landing page for users to easily sign up for our services.
- We ran an AB test of 1 week, between 2023-10-23 and 2023-11-05, on ~1200 users.
- The results show a significant positive impact on:
    - Conversion rate of sign-ups, with an increase of 12.39 percentage points
- The results show a significant impact on decreasing:
    - The number of days it takes for a user to sign up from when they visited the landing page by 0.58 days
    - The number of orders made by users without signing up by 0.14.
- Other desired results that were not too statistically significant:
    - The average number of sessions before a user signs up decreased by 0.20
    - The average number of orders made by signed up decrease increases by 0.055
- Given that there are no other negative effects on secondary and guardrail metrics, the new feature has a great positive impact overall.
  We recommend rolling out the feature.

Description of the feature and experiment

[This section was written before running the experiment]

Product and Design teams collaborated to introduce a new feature aimed at increasing sign-ups and lead conversions on the Great Plants platform. The experiment evaluates the impact of an additional signup incentive for users during their interaction with the platform.

Description of the Feature:
A new signup banner will be displayed prominently on the landing page for users in the Treatment group. This banner will highlight an easy way for users to sign up during their visit, and motivate them to use our services more often in the long term.
The incentive banner will be strategically placed to capture the user's attention without disrupting the overall user experience. It will create a sense of urgency and value, encouraging users to sign up through the landing page.

The experiment is a classic AB experiment:

- For two weeks a certain number of users (defined below in the planning section) will be randomly split 50/50 into 2 groups: Control and Treatment.
- *Control:* Users in this group will experience the regular, existing platform interface without the additional signup incentive banner.

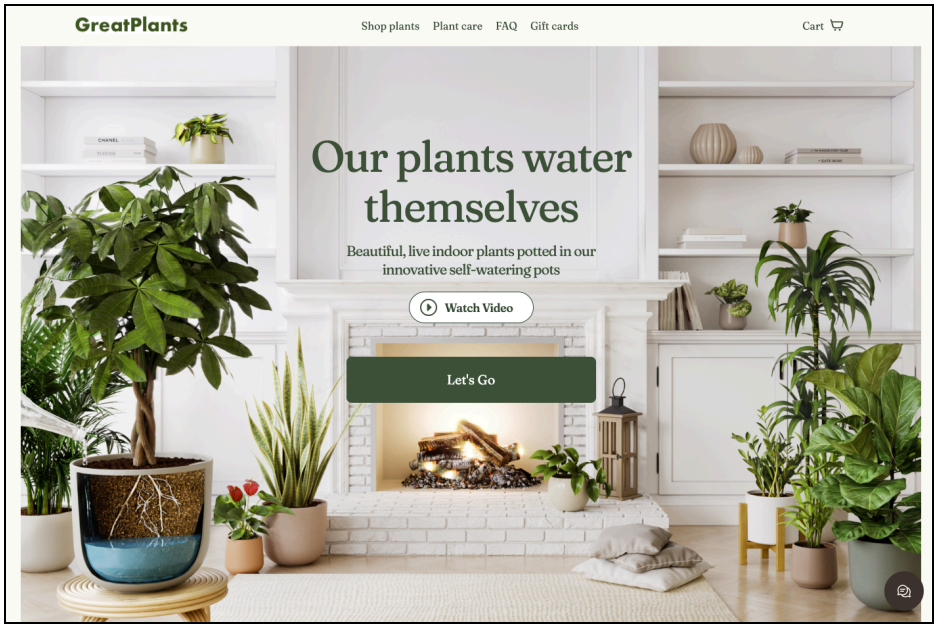

Fig 2. Control Group

- *Treatment:* Users in this group will see the new signup incentive banner during their interactions on the platform.

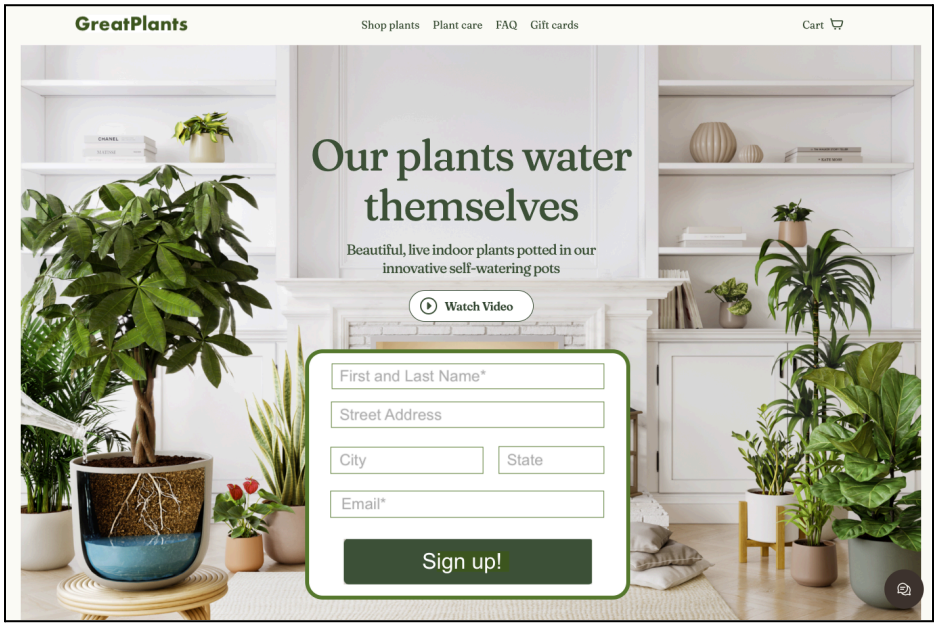

Fig 3. Treatment Group

Then, we will analyze the data and determine that either:

- The new feature has a significant impact on key performance metrics and should be rolled out

- The new feature performed similarly to the old feature, in terms of key performance metrics and should be not rolled out
- The new feature does not show a significant improvement. In this case, we will not roll out the new feature

Experiment planning

We wrote this section on October 22nd, 2023, right before running the experiment

Primary Objective:
Measure the effectiveness of a new landing page in increasing lead sign-ups.

Data Collection:
Utilize historical data from the two weeks preceding the experiment
(October 9th, 2023 to October 22nd, 2023)
Tables: dim_lead, fact_landing, fact_order.

Experimental Groups:
Control Group (Old Page): experiment_groups with Group as 'control'.
Treatment Group (New Page): experiment_groups with Group as 'treatment'.

Randomization:
Randomization will occur at the *lead level,* assigning leads randomly to control or treatment groups during their initial landing page session. The unit of randomization is across leads, not individual sessions. Returning leads will maintain their group assignment based on their IP address to ensure consistency. Metrics for analysis will be at the lead level also to avoid confusion for returning customers experiencing different landing pages during multiple sessions.

Success Metrics

Sample Size Estimation: Determine the required sample size for statistical power, using power analysis.

Hypotheses:
Null Hypothesis: No significant difference in sign-up rates between old and new landing pages.
Alternative Hypothesis: New landing page increases sign-up rates.

Impact Estimation (for Planning):
Assume an impact of at least 22% of the baseline on the primary metric.

Total leads in the 2 weeks before the experiment: 1201

With the *Control Group: 598 Leads, Treatment Group: 603 Leads*
(see Query 1 and Query 2 in the Appendix).

Primary Metric: Conversion Rate (Sign-up completion / Total leads).

Metric	Type	Priority of metric	Default value	Desired effect
Conversion rate of signups	(Sign-up completion /Total leads)	primary	45.42%	Go Up
Average no. of days between landing and signup	Sample Mean	Secondary	2.12 days	Go Down
Average number of sessions before signing up	Sample mean	Secondary	1.746	Go Down
Average number of orders by signed-up users	Sample mean	Guardrail	1.24	Go Up
Average number of orders without signing up	Sample mean	Guardrail	1.22	Go Down
Percentage Orders Per Session	Ratio Metric (avg number of orders/ average number of sessions)	Guardrail	45.43%	No effect

Power Analysis

We want to run the experiment for 2 weeks, between 2023-10-23 and 2023-11-05 (both included). We have thousands of users on our platform but won't need to run the experiment on all of them. A subset will suffice.

We will base power analysis on the primary metric, looking at data from 2 weeks right before the experiment, that is the weeks between October 9th, 2023 to October 22nd, 2023.

As seen in the table above, on average, in the week between 2023-10-09 and 2023-10-22, an active lead (that is, a user with one distinct IP Address) signs up with an average of 45.42%. Let's fix the alpha (probability of Type I error) to be 5%. Actually, given that we are going to run a test on 6 metrics, we will use the Bonferroni correction to be conservative. Therefore our alpha value is 0.05/6 = 0.0083. Let's fix the beta (probability of Type II error) to be 20%.

In the context of a Bernoulli variable, variance is **p*(1-p).**
And even based on the historical data that we analyzed, we saw that the standard deviation is **49.8% or 0.498.**

In the plot below:

- **Delta** (on the y-axis) represents the detectable difference that we will observe between the conversion rate of sign-ups for leads in Control and Treatment.
- **n** (on the x-axis) represents the number of users that we need in each group to detect the corresponding significant delta, after fixing alpha and beta. It was computed using the following formula:

$$n = 2\left(\hat{\sigma}\,\frac{z_{\alpha/2} + z_{\beta}}{\delta}\right)^2$$

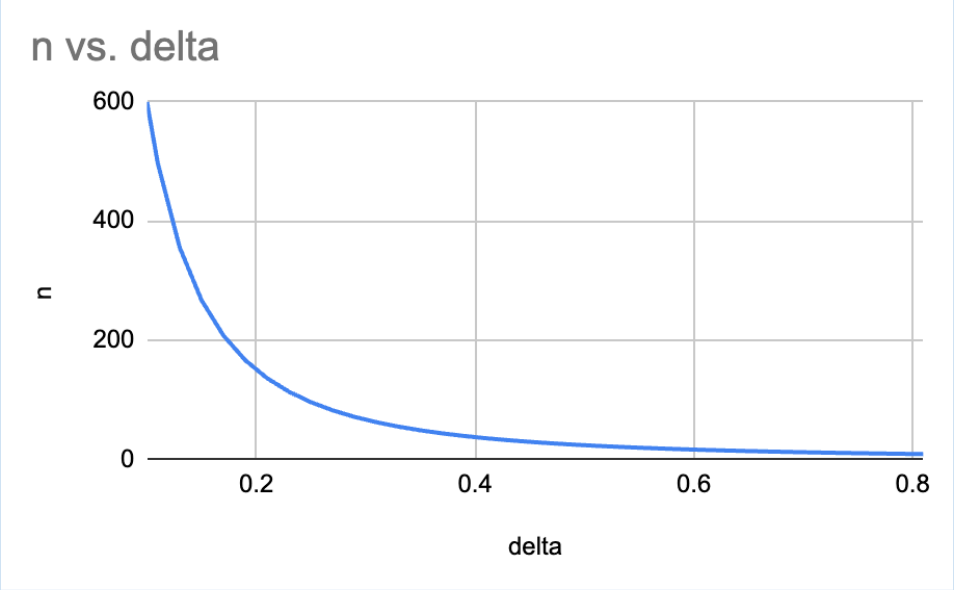

Fig 3. N vs Delta: For our Experiment

After discussing with the Product team, we expect the new feature to have an impact of at least **delta = ~0.1** on the primary metric. The corresponding desired sample size is 599 (in each group). Rounding up the numbers (to be conservative, since there is always some spillage), we believe we can run an experiment on **1200 users** (50% in treatment, 50% in control). If the difference in the primary metric will be at least 0.1 there is a high probability (80%) that our experiment will detect it. See calculations in this spreadsheet.

Data Logging

Our Experimentation platform will automatically store the allocation of users in Control and Treatment in the table **experiment_groups**, which contains the following columns:

- *experiment_name:* the unique identifier of an experiment
- *unit_id:* the identifier of the unit of randomization that was used in the experiment
- *grouped_at:* timestamp of when a unit was assigned to a group in an experiment
- *group:* the name of the group that a unit was assigned to

We won't have to worry about logging additional data since
standard business tables will log data for the behavior of the users. In particular:

- **dim_lead** contains one row per lead that ever started a session on a landing page, with the following columns:
    - lead_id: the lead identifier
    - lead_source: can be one of direct/search/ads /social
    - first_session_at: timestamp of the first session of a lead
    - signup_at: timestamp of the signup completion. If NULL, the lead never signed up.
    - lead_info: encrypted information provided by the lead during signup (name, address, email, etc). If NULL, the lead never signed up.

- **fact_landing** with one row per session that started on a landing page, with the following columns:
    - lead_id: the lead identifier
    - session_id: the session identifier
    - session_at: timestamp of the session
    - signup_at: if signup was completed during the session, the timestamp of the signup is recorded here.
    - order_id: if an order was placed during the session, the order_id is recorded here.

- **fact_order** with one row per order, which could be generated from a session that started on a landing page or from a different flow, with the following columns:
    - lead_id: the lead or client identifier (even after a lead signs up, we still identify them with their lead_id)
    - order_id: the order identifier
    - order_at: timestamp of the order
    - tot_price: total price of the order, before taxes
    - taxes: sales tax applied to the order
    - tot_cost: total cost paid by the customer, including taxes

Rollout

We wrote this section on October 23rd, 2023, at the beginning of the Experiment.

- On 2023-10-23 we officially rolled out the experiment to almost 1200 new users, who have visited the GreatPlants website for the first time. The platform automatically logged the corresponding data.

Name of the experiment	landing_2023
Unit of Randomization	uni_id (~1200)
Sources	Direct, Search, Ads, Social
XP start timestamp	2023-10-23 00:00:00 UTC
XP end timestamp	2023-11-05 23:59:59 UTC
Number of variants	2

Analysis of results

We wrote this section on November 6th, 2023, a few days after the experiment was completed.

The experiment ran between 2023-10-23 and 2023-11-05 (included).
We saw:

- 598 active users in the Control group
- 603 active users in the Treatment group

We don't see 600 users in each group because of some minor spillage (data that was not logged correctly). The 50/50 randomization has worked well since we see almost exactly 50% in Control and 50% in Treatment.

We now have all the data to run our analysis. In particular, we will consider all the metrics in the "Success metric" section above, and we will compare the values obtained in Control and Treatment, with statistical tests. Given that we will perform 6 tests, we will use the Bonferroni correction to declare a result statistically significant. That is, instead of setting a threshold of 0.05 for our p-values, the threshold is 0.05/6 = 0.0083.

Queries in the appendix return the data that we used in this spreadsheet to do our calculations, summarized in the table below:

Metric	Control group	Treatment group	Difference (T-C)	p-value	Effect
Conversion rate of signups	45.65%	58.04%	12.39pp	0.000015	Desired Increase

Average no. of days between landing and signup	1.985348	1.397143	-0.588205	0.000577	Desired decrease
Average number of sessions before signing up	1.7436	1.54	-0.2036	0.01	Desired decrease but not statistically significant
Average number of orders by signed-up users	1.457143	1.512262	0.055119	0.363595	Desired increase but not statistically significant
Average number of orders without signing up	1.335443	1.196262	-0.139181	0.0017	Desired decrease, is statistically significant
Percentage Orders Per Session	47.09%	56.86%	9.7733pp	<0.0001	No effect expected

Conclusion

The new feature has a desired positive impact on:

- The conversion rate of Sign ups shows an increase of 12.39 percentage points and this desired effect is also statistically significant.
- The average number of days it was taking customers to sign up has decreased by about 0.58 days and this desired effect is also statistically significant.
- The Percentage of average number of orders per average number of sessions per user has also increased by 9.77 percentage points.
- Moreover, the average number of orders without signing up has gone down by 0.14 which is a significant change.
- The average number of orders by signed-up users and the average number of sessions before signing up were secondary and guardrail metrics, but they show a desired impact that is not statistically significant, hence we can disregard them.

We recommend rolling out the feature to all users.

APPENDIX

Spreadsheets for this Experiment:

- Historical Data Analysis
- Power Analysis for Conversion Rate
- Metric values for Great Plants

Here are the queries that we used to retrieve the data for the various sections.

For Success Metrics table

Query 1

```
-- Get distinct lead id for sample size calculation
SELECT DISTINCT
 lead_id,
 CASE WHEN MAX(signup_at) IS NOT NULL THEN 1 ELSE 0 END AS signup_flag
FROM
 `fabriziopublic.greatplants.fact_landing`
WHERE
 1=1
 AND session_at >= TIMESTAMP("2023-10-09 00:00:00 UTC")
 AND session_at < TIMESTAMP("2023-10-22 23:59:00 UTC")
GROUP BY
 lead_id



WITH SignupCounts AS (
 SELECT
   eg.group,
   COUNT(DISTINCT dl.lead_id) AS total_leads,
   COUNT(DISTINCT CASE WHEN fl.signup_at IS NOT NULL THEN dl.lead_id END) AS
signup_leads
 FROM
   fabriziopublic.greatplants.experiment_groups eg
 LEFT JOIN
   fabriziopublic.greatplants.dim_lead dl
 ON
```

```
    `
      eg.unit_id = dl.lead_id
    LEFT JOIN
      fabriziopublic.greatplants.fact_landing fl
    ON
      dl.lead_id = fl.lead_id
      AND fl.session_at BETWEEN TIMESTAMP("2023-10-23") AND TIMESTAMP("2023-11-05") --
Assuming the experiment duration
    WHERE
      eg.experiment_name = "landing_2023"
    GROUP BY
      eg.group
)

SELECT
  `group`,
  total_leads,
  signup_leads,
  signup_leads / total_leads AS conversion_rate
FROM
  SignupCounts;
```

For data exploration before the experiment:

```
Data before experiment:
SELECT COUNT(DISTINCT lead_id)
FROM `fabriziopublic.greatplants.fact_landing`
WHERE 1=1
AND session_at >= CAST("2023-10-09 00:00:00 UTC" as TIMESTAMP)
-- Assuming a 2-week period before the experiment
AND session_at < CAST("2023-10-22 23:59:00 UTC" as TIMESTAMP)


-- Calculate default conversion rate before the experiment
WITH SignupData AS (
  SELECT
    COUNT(DISTINCT lead_id) AS total_leads,
    COUNT(DISTINCT CASE WHEN signup_at IS NOT NULL THEN lead_id END) AS converted_leads
  FROM
```

```
    fabriziopublic.greatplants.dim_lead
  WHERE
    first_session_at >= '2023-10-09' -- Assuming a 2-week period before the experiment
    AND first_session_at < '2023-10-23'
)

SELECT
  total_leads,
  converted_leads,
  IFNULL(converted_leads / total_leads, 0)*100 AS conversion_rate
FROM
  SignupData;

-- Explore the number of sign-ups and sources
SELECT
  lead_source,
  COUNT(DISTINCT lead_id) AS total_leads,
  COUNT(DISTINCT CASE WHEN signup_at IS NOT NULL THEN lead_id END) AS total_signups
FROM
  fabriziopublic.greatplants.dim_lead
GROUP BY
  lead_source;

-- Explore average time between landing and signup
WITH SignedUpUsers AS (
  SELECT
    dl.lead_id,
    MIN(fl.session_at) AS landing_time,
    dl.signup_at
  FROM
    fabriziopublic.greatplants.dim_lead dl
  JOIN
    fabriziopublic.greatplants.fact_landing fl ON dl.lead_id = fl.lead_id
  WHERE
    dl.signup_at IS NOT NULL
    AND fl.session_at >= TIMESTAMP("2023-10-09 00:00:00 UTC")
```

```
    AND fl.session_at < TIMESTAMP("2023-10-22 23:59:59 UTC")
  GROUP BY
    dl.lead_id, dl.signup_at
)

SELECT
  AVG(DATE_DIFF(signup_at, landing_time, DAY)) AS avg_days_between_landing_and_signup
FROM
  SignedUpUsers;

-- Explore shopping behavior
SELECT
  lead_id,
  COUNT(DISTINCT order_id) AS total_orders,
  SUM(tot_price) AS total_sales
FROM
  fabriziopublic.greatplants.fact_order
GROUP BY
  Lead_id;

-- Explore revenue from orders per lead
WITH UserOrders AS (
  SELECT
    dl.lead_id,
    fo.order_id,
    fo.tot_cost
  FROM
    fabriziopublic.greatplants.dim_lead dl
  JOIN
    fabriziopublic.greatplants.fact_order fo ON dl.lead_id = fo.lead_id
  WHERE
    fo.order_at >= TIMESTAMP("2023-10-09 00:00:00 UTC")
    AND fo.order_at < TIMESTAMP("2023-10-22 23:59:59 UTC")
)

SELECT
```

```
 AVG(tot_cost) AS avg_revenue_per_user
FROM
 UserOrders;

-- Average orders per signed up user:
SELECT
 COUNT(DISTINCT fo.order_id) / COUNT(DISTINCT fl.lead_id) AS avg_orders_per_user
FROM
 fabriziopublic.greatplants.fact_landing fl
JOIN
 fabriziopublic.greatplants.fact_order fo ON fl.lead_id = fo.lead_id
WHERE
 fl.session_at >= TIMESTAMP("2023-10-09 00:00:00 UTC")
 AND fl.session_at < TIMESTAMP("2023-10-22 23:59:59 UTC")
 AND fl.signup_at IS NOT NULL;


-- Average orders per non signed-up user:
WITH SignedUpUsers AS (
 SELECT
   fo.lead_id,
   COUNT(DISTINCT fo.order_id) AS num_orders
 FROM
   fabriziopublic.greatplants.dim_lead dl
 JOIN
   fabriziopublic.greatplants.fact_order fo ON dl.lead_id = fo.lead_id
 WHERE
   1=1
   AND dl.signup_at IS NULL
   AND fo.order_at >= TIMESTAMP("2023-10-09 00:00:00 UTC")
   AND fo.order_at < TIMESTAMP("2023-10-22 23:59:59 UTC")
 GROUP BY
   fo.lead_id
)

SELECT
```

```
 AVG(num_orders) AS avg_orders,
 STDDEV(num_orders) AS std_dev_orders
FROM
 SignedUpUsers;
```

Query 2

For data exploration before the experiment: this query provides the percentage of order returns, expressed as (returns per user)/(orders per user), where we need to consider all active users.

```
WITH OrderCount AS (
 SELECT
   fl.lead_id,
   COUNT(DISTINCT fo.order_id) AS order_count,
   COUNT(DISTINCT fl.session_id) AS session_count
 FROM
   fabriziopublic.greatplants.fact_landing fl
 LEFT JOIN
   fabriziopublic.greatplants.fact_order fo
 ON
   fl.lead_id = fo.lead_id
 WHERE
   fl.session_at >= TIMESTAMP("2023-10-09 00:00:00 UTC")
   AND fl.session_at < TIMESTAMP("2023-10-22 23:59:59 UTC")
 GROUP BY
   fl.lead_id
)

SELECT
 AVG(order_count) / AVG(session_count) * 100 AS percentage_orders_per_session
FROM
 OrderCount;
```

Query 3

This query returns the count of active clients in the 2 groups of the experiment.

```
-- Count how many users have been assigned into each XP group
```

```
SELECT xp.group, COUNT(*)
FROM fabriziopublic.greatplants.experiment_groups xp
WHERE 1=1
AND experiment_name = "landing_2023"
GROUP BY 1
```

Query 4

This query returns the primary metric and secondary metrics for the 2 groups during the experiment.

```
-extract control and treatment data for sign ups and computed in excel
WITH group_data AS (
 SELECT
   xp.unit_id AS id,
   xp.group AS group_name,
   dl.signup_at IS NOT NULL AS signup_flag
 FROM
   fabriziopublic.greatplants.experiment_groups xp
 LEFT JOIN
   fabriziopublic.greatplants.dim_lead dl ON xp.unit_id = dl.lead_id
     AND dl.first_session_at >= TIMESTAMP('2023-10-23 00:00:00 UTC')
     AND dl.first_session_at <= TIMESTAMP('2023-11-05 23:59:59 UTC')
 WHERE
   xp.experiment_name = 'landing_2023'
)

SELECT
 id,
 signup_flag,
 group_name
FROM
 Group_data;

-- Calculate average number of days between landing and signup
WITH experiment_data AS (
   SELECT
```

```
        `
        eg.group,
        fl.lead_id,
        MIN(fl.session_at) AS landing_time,
        dl.signup_at
    FROM
        fabriziopublic.greatplants.experiment_groups eg
    JOIN
        fabriziopublic.greatplants.fact_landing fl
    ON
        eg.unit_id = fl.lead_id
    LEFT JOIN
        fabriziopublic.greatplants.dim_lead dl
    ON
        fl.lead_id = dl.lead_id
        AND dl.signup_at IS NOT NULL
    WHERE
        eg.experiment_name = "landing_2023"
        AND fl.session_at BETWEEN TIMESTAMP("2023-10-23 00:00:00 UTC") AND
TIMESTAMP("2023-11-05 23:59:59 UTC")
    GROUP BY
        eg.group, fl.lead_id, dl.signup_at
)

SELECT
    `group`,
    AVG(DATE_DIFF(signup_at, landing_time, DAY)) AS
avg_days_between_landing_and_signup,
    STDDEV_POP(DATE_DIFF(signup_at, landing_time, DAY)) AS
sd_days_between_landing_and_signup
FROM
    experiment_data
GROUP BY
    `group`;

-- Calculate average and standard deviation for the number of orders by signed-up
users
```

```
WITH ExperimentData AS (
   SELECT
       eg.group,
       fo.lead_id,
       COUNT(DISTINCT fo.order_id) AS num_orders
   FROM
       fabriziopublic.greatplants.experiment_groups eg
   JOIN
       fabriziopublic.greatplants.fact_order fo
   ON
       eg.unit_id = fo.lead_id
   WHERE
       eg.experiment_name = 'landing_2023'
       --AND eg.group IN ('control', 'treatment')
   GROUP BY
       eg.group, fo.lead_id
)

SELECT
   `group`,
   AVG(num_orders) AS avg_num_orders,
   STDDEV_SAMP(num_orders) AS sd_num_orders
FROM
   ExperimentData
GROUP BY
   `group`;


-- Average orders per non-signed-up user for the landing page experiment
WITH ExperimentGroups AS (
 SELECT
   unit_id,
   `group`
 FROM
   fabriziopublic.greatplants.experiment_groups
 WHERE
```

```
    1 = 1
    AND experiment_name = "landing_2023"
)

, NonSignedUpUsers AS (
 SELECT
    dl.lead_id,
    COUNT(DISTINCT fo.order_id) AS num_orders,
    eg.group
 FROM
    fabriziopublic.greatplants.dim_lead dl
 JOIN
    fabriziopublic.greatplants.fact_order fo ON dl.lead_id = fo.lead_id
 LEFT JOIN
    ExperimentGroups eg ON dl.lead_id = eg.unit_id
 WHERE
    1 = 1
    AND dl.signup_at IS NULL
    AND fo.order_at >= TIMESTAMP("2023-10-23 00:00:00 UTC")
    AND fo.order_at < TIMESTAMP("2023-11-05 23:59:59 UTC")
 GROUP BY
    dl.lead_id, eg.group
)

SELECT
 nsg.group,
 AVG(nsg.num_orders) AS avg_orders,
 STDDEV(nsg.num_orders) AS std_dev_orders
FROM
 NonSignedUpUsers nsg
GROUP BY
 nsg.group;
```

Query 5

This query returns the metric “Percentage of orders by sessins” (guardrail) for the 2 groups during the experiment.

```
-- Calculate average and standard deviation for orders and sessions separately for
Control and Treatment groups
WITH OrderCount AS (
 SELECT
   xp.group,
   fl.lead_id,
   COUNT(DISTINCT fo.order_id) AS order_count,
   COUNT(DISTINCT fl.session_id) AS session_count
 FROM
   fabriziopublic.greatplants.experiment_groups xp
 LEFT JOIN
   fabriziopublic.greatplants.fact_landing fl ON xp.unit_id = fl.lead_id
 LEFT JOIN
   fabriziopublic.greatplants.fact_order fo ON fl.lead_id = fo.lead_id
 WHERE
   xp.experiment_name = "landing_2023"
   AND fl.session_at >= TIMESTAMP("2023-10-23 00:00:00 UTC")
   AND fl.session_at < TIMESTAMP("2023-11-06 00:00:00 UTC")
 GROUP BY
   xp.group, fl.lead_id
)

SELECT
 oc.group,
 AVG(oc.order_count) AS avg_orders,
 STDDEV(oc.order_count) AS sd_orders,
 AVG(oc.session_count) AS avg_sessions,
 STDDEV(oc.session_count) AS sd_sessions,
 COVAR_SAMP(oc.order_count, oc.session_count) AS cov_orders_sessions,
 AVG(oc.order_count) / AVG(oc.session_count) * 100 AS perc_of_orders_by_session
FROM
 OrderCount oc
GROUP BY
 oc.group;
```