SQL Query

Que2. What type of join should we use to join the users table to the activity table?

Ans: Left join (since we want all data from user table)

Que3. What SQL function can we use to fill in NULL values?

Ans: Coalesce function.

Que4. What are the start and end dates of the experiment?

Ans: start date = 2023-01-25 and End date = 2023-02-06

Query: select min(activity.dt)as start\_date,max(activity.dt) as end\_date

from activity;

Que5. How many total users were in the experiment?

Ans: total\_users in experiment=48943

Query: select count(uid)as total\_users

from groups;

Que6. How many users were in the control and treatment groups?

Ans: Control(A):24343 and Treatment(B):24600

Query:

*select g.group, count(uid) as total\_users*

*From groups g*

*group by g.group*

Que7. What was the conversion rate of all users?

Ans: 4.28%

Query:

*select Round (cast(count(distinct uid) as decimal)/cast(count(distinct id) as decimal)\*100,2) as Conversion\_rate*

*from users u left join activity a on u.id = a.uid*

Que8. What is the user conversion rate for the control and treatment groups?

Ans: Control(A): 3.92% and Treatment(B): 4.63%

Query:

*select g.group,*

*Round(cast(count(distinct a.uid) as decimal)/cast(count(distinct u.id) as decimal)\*100,2) as Conversion\_rate*

*from users u*

*left join activity a on u.id = a.uid*

*left join groups g on u.id=g.uid*

*group by g.group*

Que9. What is the average amount spent per user for the control and treatment groups, including users who did not convert?

Ans: Control(A): 3.37 and Treatment(B): 3.39

Query:

*with users spent as(*

*select*

*u.id as userid,*

*sum (COALESCE (a. spent,0)) as total\_spent*

*from activity a*

*right join users u on a.uid = u.id*

*group by u.id*

*)*

*select*

*g.group,*

*round(avg(total\_spent),2) as avg\_spent*

*from users\_spent u left join groups g on u.userid=g.uid*

*group by g.group*

Que10. Why does it matter to include users who did not convert when calculating the average amount spent per user? (this data used to calculate the Hypothesis test in excel spreadsheet)

QUERY:

*SELECT u.id AS user\_id, u.country, u.gender, g.device,g.group,*

*SUM (COALESCE (a.spent, 0)) AS total\_spent\_usd*

*FROM users AS u*

*LEFT JOIN groups AS g ON u.id = g.uid*

*LEFT JOIN activity AS a ON u.id = a.uid*

*GROUP BY u.id, u.country, u.gender, g.device, g.group;*

* This data used for the Tableau visualisation: (4 tables viz., Globox, Globox 2, Globox 3, Globox 4)

1. Globox: (used to visualize the conversion\_rate, avg amount spent, avg amount distribution per user , gender wise, device wise, country wise distribution)

Query:

*Select*

*distinct groups.uid as user\_id, users.country,coalesce(users.gender,'Unknown') as gender, coalesce(groups.device,'Unknown') as device, groups.group sum(coalesce(activity.spent,0)) as total\_spent\_usd, sum(round(coalesce(activity.spent,0) - mod(coalesce(activity.spent,0),10),0)) total\_spent, Round(round(cast(count(distinct activity.uid) as decimal))/round(cast(count(distinct users.id)as decimal),2)\*100,2) as conversion\_rate*

*from groups*

*left join activity on groups.uid=activity.uid*

*left join users on groups.uid=users.id.*

*group by*

*groups.uid, users.country, users.gender, groups.device, groups.group*

Novelty Effect Analysis:

1. Globox 2: (used to visualise Novelty Effect : AVG Amount Spent over time )

Query:

*select*

*distinct groups.uid as user\_id,*

*groups.join\_dt as Join\_date,*

*users.country,*

*coalesce(users.gender,'Unknown') as gender,*

*coalesce(groups.device,'Unknown') as device,*

*groups.group,*

*sum (coalesce (activity. spent,0)) as total\_spent\_usd,*

*sum (round (coalesce (activity. spent,0) - mod (coalesce (activity. spent,0),10),0)) total\_spent,*

*case when sum (coalesce (activity. spent,0)) =0 then 0 else 1 end as Converted*

*from groups*

*left join activity on*

*groups.uid=activity.uid*

*left join users on*

*groups.uid=users.id*

*group by*

*groups.uid,*

*users.country,*

*users.gender,*

*groups.device,*

*groups.group*

1. Globox 3: (used to visualise Novelty Effect: Conversion rate over time and converted users average amount spent over time):

Query:

*select*

*g.join\_dt,g.group,*

*count (distinct u.id) as Total\_Users,*

*count (distinct a.uid) as Paid\_user,*

*SUM(a.spent) as Total\_Spent,*

*CAST (count (distinct a.uid) as Decimal)/CAST(count(distinct u.id) as Decimal) as Conversion\_Rate,*

*CAST(SUM(a.spent) as Decimal)/CAST(count(distinct u.id) as Decimal) as Average\_Amount\_Spent,*

*CAST(SUM(a.spent) as Decimal)/CAST(count(distinct a.uid) as Decimal) as Converted\_Average\_Spent*

*from users u*

*left join activity a on u.id = a.uid*

*join groups g on u.id = g.uid*

*group by*

*g.join\_dt,g.group*

1. Globox 4: (used to visualise Novelty Effect: Number of converted users after joining the experiment):

Query:

*select*

*min(g.join\_dt) as Join\_date,*

*min(a.dt) as date\_converted,*

*case when min(a.dt)-min(g.join\_dt) =0 then 'Same Day'*

*when min(a.dt)-min(g.join\_dt) =1 then '1 day'*

*else concat(min(a.dt)-min(g.join\_dt),' days') end as date\_diff,*

*g.group,*

*g.uid as user\_id,*

*sum(a.spent) as Total\_spent*

*from groups g*

*left join users u on g.uid=u.id*

*join activity a on g.uid=a.uid*

*group by*

*g.uid, g.group*

Confidence Interval

Query: **1. for Average Spent Confidence Interval**

*with cte as(*

*SELECT*

*u.id AS user\_id, u.country, u.gender, g.device,g.group,*

*SUM(COALESCE(a.spent, 0)) AS total\_spent\_usd,*

*CASE WHEN SUM(a.spent) > 0 then 1 else 0 end as converted*

*FROM users AS u*

*LEFT JOIN groups AS g ON u.id = g.uid*

*LEFT JOIN activity AS a ON u.id = a.uid*

*GROUP BY*

*u.id, u.country, u.gender, g.device, g.group*

*),*

*groupA\_cte as(*

*select*

*AVG(a.total\_spent\_usd) as groupA\_avg\_spent,*

*count(a.total\_spent\_usd) as groupA\_total\_spent,*

*sum(converted) as groupA\_total\_converted,*

*ROUND(STDDEV(a.total\_spent\_usd),2) as groupA\_std\_dev\_spent,*

*ROUND(CAST(SQRT(COUNT(a.total\_spent\_usd))AS numeric),2) as groupA\_sqrt\_spent,*

*ROUND(STDDEV(a.total\_spent\_usd),2) / ROUND(CAST(SQRT(COUNT(a.total\_spent\_usd))AS numeric),2) as groupA\_standard\_error*

*from cte a*

*where*

*a.group='A'),*

*groupB\_cte as(*

*select*

*AVG(a.total\_spent\_usd) as groupB\_avg\_spent,*

*count(a.total\_spent\_usd) as groupB\_total\_spent,*

*ROUND(STDDEV(a.total\_spent\_usd),2) as groupB\_std\_dev\_spent,*

*ROUND(CAST(SQRT(COUNT(a.total\_spent\_usd))AS numeric),2) as groupB\_sqrt\_spent,*

*ROUND(ROUND(STDDEV(a.total\_spent\_usd),2) / ROUND(CAST(SQRT(COUNT(a.total\_spent\_usd))AS numeric),2),3) as groupB\_standard\_error*

*from cte a*

*where*

*a.group='B')*

*select*

*ROUND(groupA\_avg\_spent,2) as groupA\_avg\_spent,*

*groupA\_total\_spent as groupA\_total\_users,*

*groupA\_std\_dev\_spent,*

*groupA\_sqrt\_spent,*

*ROUND(groupA\_standard\_error,2) as groupA\_standard\_error ,*

*ROUND(groupB\_avg\_spent,2) as groupA\_avg\_spent,*

*groupB\_total\_spent as groupA\_total\_users,*

*groupB\_std\_dev\_spent,*

*groupB\_sqrt\_spent,*

*ROUND(groupB\_standard\_error,2) as groupB\_standard\_error,*

*ROUND(CAST(SQRT((POWER(groupB\_std\_dev\_spent,2)/groupB\_total\_spent)+(POWER(groupA\_std\_dev\_spent,2)/groupA\_total\_spent))AS numeric),3) AS Standared\_Error*

*,ROUND((groupB\_avg\_spent-groupA\_avg\_spent),3) as "mean differnece"*

*,1.96 as "critical value",*

*ROUND((ROUND((groupB\_avg\_spent-groupA\_avg\_spent),3)) - (1.96\*SQRT((POWER(groupA\_standard\_error,2)+POWER(groupB\_standard\_error,2)))),3) AS lower\_bound,*

*ROUND((ROUND((groupB\_avg\_spent-groupA\_avg\_spent),3)) + (1.96\*SQRT((POWER(groupA\_standard\_error,2)+POWER(groupB\_standard\_error,2)))),3) AS upper\_bound*

*from groupA\_cte,groupB\_cte*

Query: **2. For Conversion Rate Confidence Interval**

*with cte as(*

*SELECT*

*u.id AS user\_id, u.country, u.gender, g.device,g.group,*

*SUM(COALESCE(a.spent, 0)) AS total\_spent\_usd,*

*CASE WHEN SUM(a.spent) > 0 then 1 else 0 end as converted*

*FROM*

*users AS u*

*LEFT JOIN groups AS g ON u.id = g.uid*

*LEFT JOIN activity AS a ON u.id = a.uid*

*GROUP BY*

*u.id, u.country, u.gender, g.device, g.group*

*),*

*groupA\_convt\_cte AS (*

*select*

*count(converted) groupA\_total\_users,*

*sum(converted) as groupA\_total\_converted,*

*cast(sum(converted) as decimal)/cast(count(a.total\_spent\_usd) as decimal)\*100 AS groupA\_converion\_Rate*

*from cte a*

*where*

*a.group='A'),*

*groupB\_convt\_cte AS (*

*select*

*count(converted) groupB\_total\_users,*

*sum(converted) as groupB\_total\_converted,*

*cast(sum(converted) as decimal)/cast(count(a.total\_spent\_usd) as decimal)\*100 AS groupB\_converion\_Rate*

*from cte a*

*where*

*a.group='B'),cte\_se as(*

*select*

*groupA\_total\_users,ROUND(groupA\_converion\_Rate,2) as groupA\_converion\_Rate,*

*groupB\_total\_users,ROUND(groupB\_converion\_Rate,2) as groupB\_converion\_Rate,*

*ROUND(cast((groupA\_total\_converted+groupB\_total\_converted) as decimal)/cast((groupA\_total\_users+groupB\_total\_users) as decimal),4) AS Total\_Converted\_Rate,*

*ROUND((groupB\_converion\_Rate-groupA\_converion\_Rate)/100,4) as conversion\_difference,*

*ROUND(sqrt(cast((groupA\_total\_converted+groupB\_total\_converted) as decimal)/cast((groupA\_total\_users+groupB\_total\_users) as decimal)*

*\*(1-(cast((groupA\_total\_converted+groupB\_total\_converted) as decimal))/cast((groupA\_total\_users+groupB\_total\_users) as decimal))*

*\*((1/cast(groupA\_total\_users as decimal))+(1/cast(groupB\_total\_users as decimal))) ),4) Total\_standard\_error*

*from*

*groupA\_convt\_cte,groupB\_convt\_cte)*

*select*

*\*,*

*ROUND((conversion\_difference-(1.95\*Total\_standard\_error))\*100,2) AS lower\_bound,*

*ROUND((conversion\_difference+(1.95\*Total\_standard\_error))\*100,2) AS upper\_bound*

*from cte\_se*