Raksha Basiwal

The Advertising Campaign Analysis

1.Business Justification

1. Explain why retargeting customers who initially didn't buy a package makes business sense.

There are several compelling reasons why retargeting customers who initially didn't purchase makes strong business sense:

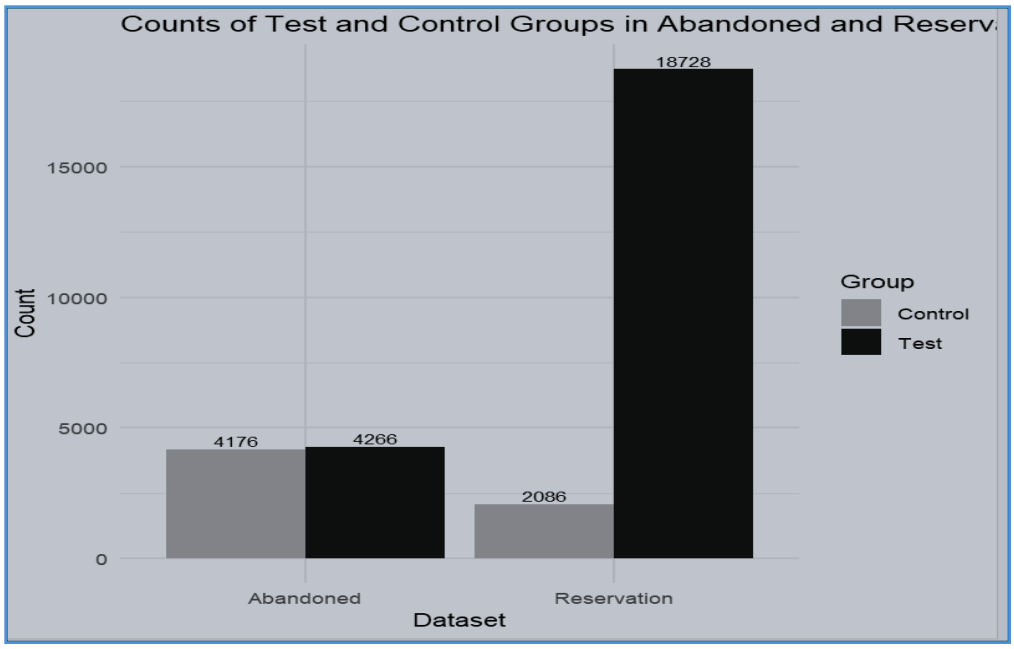
- **Higher Conversion Potential:** Retargeted customers tend to convert at higher rates than brand-new prospects. By visiting the website or engaging with the brand, these customers have already shown interest in the offerings, indicating a stronger purchase intent. Since they are familiar with the brand, retargeting them often yields higher conversion rates.
- **Cost-Effective Strategy:** Attracting new customers generally incurs greater costs than retargeting those who already have a history of engagement with the business. From a budget optimization perspective, dedicating part of the ad spend to retargeting efforts helps maximize ROI by focusing on audiences more likely to convert.
- **Personalized Advertising:** Retargeting enables tailored advertising based on prior interactions. By analyzing customer behavior, businesses can create highly relevant ads that highlight specific products or offer discounts on items customers previously viewed. This personalized approach not only enhances relevance but also boosts engagement and click-through rates, making conversions more likely.
- **Competitive Edge:** In highly competitive markets, retargeting provides an advantage by keeping the brand visible to interested customers who didn't initially convert. By maintaining a presence, the business can secure these customers before they potentially turn to competitors.

2. Analyze the test/control division. Does it seem well-executed

Analyzing the Test/Control Division

- In the *Abandoned* dataset, the split between the test and control groups appears fairly balanced, with only a minor difference of around 90 entries, equating to about a 1% variation. This slight difference is within acceptable limits and suggests that the division was managed effectively.
- On the other hand, in the *Reserved* dataset, there is a substantial imbalance, with the test group containing nearly nine times as many entries as the control group.

Since the *Reserved* dataset will primarily be used for matching purposes, this discrepancy is not a major concern for our analysis.



3. Compute summary statistics for the test variable, segmenting by available State data.

Assuming that summary statistics in this case refers to analyzing the distribution of test and control group totals by State, we'll examine how these groups vary across different States.

```
#segment by available state

if_state <- Abandoned[complete.cases(Abandoned['Address']),]
table(if_state$Test_Control)
```

```
control    test
   1855    1957
> |
```

```
> stats_overview = Abandoned %>%
+   group_by(Address, Test_Control) %>%
+   summarize(Count = n())
`summarise()` has grouped output by 'Address'. You can override using the `.groups` argument.
> print(stats_overview)
# A tibble: 102 × 3
# Groups:   Address [51]
   Address Test_Control Count
   <chr>   <chr>        <int>
 1 AK      control         32
 2 AK      test            29
 3 AL      control         42
 4 AL      test            38
 5 AR      control         46
 6 AR      test            38
 7 AZ      control         44
 8 AZ      test            54
 9 CA      control         37
10 CA      test            48
# ℹ 92 more rows
# ℹ Use `print(n = ...)` to see more rows
>
```

The Abandoned dataset shows a fairly balanced distribution of test and control groups at the state level.

2.Data Alignment

4. From your examination of both files, propose potential data keys to match customers.

I plan to match the files using the following fields: **Contact_Phone**, **Email**, **Incoming_Phone**. Additionally, I will cross-match the Contact_Phone field from the Abandoned dataset with the Incoming_Phone field in the Reserved dataset, and the reverse as well.

5. Detail your procedure to identify customers in:

- *Treatment group who purchased.*
- *Treatment group who didn’t purchase.*
- *Control group who purchased.*
- *Control group who didn’t purchase.*

To identify customers across the specified groups, we start by matching records in the 'Abandoned' and 'Reservation' datasets using several key fields. For each matching condition, a flag is created in the abandoned dataset:

- **email_match**: Matches based on the Email field.

- **incom_match**: Matches based on Incoming_Phone.
- **contact_match**: Matches based on Contact_Phone.
- **incom_contact_match**: Matches where Incoming_Phone from abandoned aligns with Contact_Phone in reservation.
- **contact_incom_match**: Matches where Contact_Phone from abandoned aligns with Incoming_Phone in reservation.

These flags will help us identify customers in both datasets and categorize them into the treatment and control groups based on their purchase behavior.

```
emailmatch = Abandoned$Email[complete.cases(Abandoned$Email)] %in% Reservation$Email[complete.cases(Reservation$Email)]
incom_match = Abandoned$Incoming_Phone[complete.cases(Abandoned$Incoming_Phone)] %in% Reservation$Incoming_Phone[complete.cases(Reservation$Incoming_Phone)]

contactmatch = Abandoned$Contact_Phone[complete.cases(Abandoned$Contact_Phone)] %in% Reservation$Contact_Phone[complete.cases(Reservation$Contact_Phone)]

incom_contact_match = Abandoned$Incoming_Phone[complete.cases(Abandoned$Incoming_Phone)] %in% Reservation$Contact_Phone[complete.cases(Reservation$Contact_Phone)]

contact_incom_match = Abandoned$Contact_Phone[complete.cases(Abandoned$Contact_Phone)] %in% Reservation$Incoming_Phone[complete.cases(Reservation$Incoming_Phone)]
```

```
Abandoned$emailmatch = 0
Abandoned$emailmatch[complete.cases(Abandoned$Email)] = 1 * emailmatch

Abandoned$incom_match = 0
Abandoned$incom_match[complete.cases(Abandoned$Incoming_Phone)] = 1 * incom_match

Abandoned$contactmatch= 0
Abandoned$contactmatch[complete.cases(Abandoned$Contact_Phone)] = 1 * contactmatch

Abandoned$incom_contact_match= 0
Abandoned$incom_contact_match[complete.cases(Abandoned$Incoming_Phone)] = 1 * incom_contact_match

Abandoned$contact_incom_match= 0
Abandoned$contact_incom_match[complete.cases(Abandoned$Contact_Phone)] = 1 * contact_incom_match
```

Next, we establish a logical flag abandoned$pur in the 'Abandoned' dataset to identify if a record matches any record in the 'Reservation' dataset based on one or more of the above matching keys. Using OR (|) logic, this flag will be set to 1 if a match is found based on any condition, and 0 if no match exists. This flag helps us distinguish which customers from the abandoned dataset proceeded to make a reservation.

```
# Logical selection for matching records for those who purchased
Abandoned$pur = 1 * ( Abandoned$emailmatch | Abandoned$incom_match |Abandoned$contactmatch | Abandoned$incom_contact_match | Abandoned$contact_incom_match)
```

· After that, an additional column is created :

```
Abandoned$treat = ifelse(Abandoned$Test_Control == "test", 1, 0)
```

abandoned$treat - a binary column derived from the Test_Control column where "test" is represented as 1 and "control" is represented as 0.

Lastly, I created a table to display the combination of purchase decisions (abandoned$pur) and test/control assignments (abandoned$treat). This table presents the customer counts in each segment, offering a clear view of the distribution across these categories.

```
tab = table(Abandoned$pur, Abandoned$treat)
# Adding row labels for 'Outcome'
rownames(tab) = c("Not Purchased", "Purchased")
# Adding column labels for 'Treatment'
colnames(tab) = c("Control Group", "Treatment Group")
print(tab)
```

```
               Control Group Treatment Group
  Not Purchased          4083            3921
  Purchased                93             345
> |
```

Treatment group who purchased: Represented by the “Purchased” row within the “Treatment Group” column, showing a total of 345.

Treatment group who didn’t purchase: Located in the “Not Purchased” row under the “Treatment Group” column, totaling 3,921.

Control group who purchased: Found in the “Purchased” row of the “Control Group” column, with a count of 93.

Control group who didn’t purchase: Located in the “Not Purchased” row under the “Control Group” column, totaling 4,083.

6. Are there unmatchable records? If yes, provide examples and exclude them from the analysis

Assuming unmatched entries are treated as "not purchased," the solution is as follows:

```
#Filter out unmatched records
unmatchable_Abandoned <- Abandoned[Abandoned$pur == 0, ]
head(unmatchable_Abandoned)

#Dropping unmatching data and selecting matched (purchased) records
Abandoned_match <- Abandoned[Abandoned$pur == 1, ]
```

```
> head(unmatchable_Abandoned)
# A tibble: 6 × 21
  Caller_ID Session First_Name Last_Name Street City  Address Zipcode Email Incoming_Phone Contact_Phone Test_Control emailmatch
  <chr>     <chr>   <chr>      <chr>     <chr>  <chr> <chr>   <chr>   <chr> <chr>          <chr>         <chr>                <dbl>
1 68359340… 2014.0… Bertha     NA        NA     NA    NA      NA      NA    (864)-004-6354 (864)-004-63… test                     0
2 83119994… 2014.0… Kyle       NA        NA     NA    NA      NA      NA    (703)-220-0148 (703)-220-01… control                  0
3 58448995… 2014.0… Paxton     NA        NA     NA    NA      NA      NA    (559)-299-7745 (559)-299-77… control                  0
4 84112006… 2014.0… Thelma     NA        NA     NA    NA      NA      NA    (636)-611-4439 (636)-611-44… test                     0
5 10840705… 2014.0… Lorna      NA        NA     NA    NA      NA      NA    (253)-461-5118 (253)-461-51… control                  0
6 65443966… 2014.0… Leann      NA        NA     NA    NA      NA      NA    (407)-910-9280 (407)-910-92… test                     0
# ℹ 8 more variables: incom_match <dbl>, contactmatch <dbl>, incom_contact_match <dbl>, contact_incom_match <dbl>, pur <dbl>,
#   email <dbl>, state <dbl>, treat <dbl>
>
```

We cannot exclude these records from the analysis, as both "purchased" and "not purchased" entries are essential for a complete evaluation.

7. Provide a cross-tabulation of outcomes for treatment and control groups.

```
#Cross tabulations for all records(purchased and not purchased)
tab = table(Abandoned$pur, Abandoned$treat)
rownames(tab) = c("Not Purchased", "Purchased")
# Add column labels for 'Outcome'
colnames(tab) = c("Control Group", "Treatment Group")
print(tab)
```

8. Replicate the cross-tabulation for five randomly chosen states, detailing your selections.

```
all_states = Abandoned$Address[!is.na(Abandoned$Address)]
set.seed(123)  # Setting a seed for reproducibility
random_states = sample(all_states, 5)

cross_tabulations = list()

for (state in random_states) {
  subset_data = Abandoned[Abandoned$Address == state, ]
  cross_tabulation =  table(subset_data$pur, subset_data$treat)
  rownames( cross_tabulation ) =  c("Not Purchased", "Purchased")
  colnames( cross_tabulation ) = c("Control Group", "Treatment Group")
  cross_tabulations[[state]] = cross_tabulation
}

# Print the cross-tabulations
for (state in random_states) {
  cat("Cross-tabulation for State:", state, "\n")
  print(cross_tabulations[[state]])
  cat("\n")
}
```

```
Cross-tabulation for State: AR

                Control Group Treatment Group
  Not Purchased            45               35
  Purchased                 1                3

Cross-tabulation for State: NH

                Control Group Treatment Group
  Not Purchased            39               26
  Purchased                 5                2

Cross-tabulation for State: TN

                Control Group Treatment Group
  Not Purchased            40               35
  Purchased                 1                5

Cross-tabulation for State: NH

                Control Group Treatment Group
  Not Purchased            39               26
  Purchased                 5                2

Cross-tabulation for State: NE

                Control Group Treatment Group
  Not Purchased            42               30
  Purchased                 3                3

>
```

3. Data Refinement

9. Generate a cleaned dataset with columns: Customer ID — Test Group — Outcome — State Available —Email Available. Each row should correspond to a matched customer from the datasets. (Ensure you attach this cleaned dataset upon submission.

```
# Remove multiple columns
abandon_clean = Abandoned %>%
  select(-(2:17))

#Changing index of columns and their column names
abandon_clean = abandon_clean %>%
  select(1, 5, 2, 4,3:ncol(abandon_clean))
colnames(abandon_clean) = c("Customer_ID", "Test_Group","Outcome","State_Available",
                      "Email_Available")
```

Step 1: Remove unnecessary columns that are not relevant to the analysis.

Step 2: Update the index of the remaining columns and rename them according to the specified template.

Step 3: Export the cleaned dataset for submission, ensuring it meets the required format.

4. Statistical Assessment

*10. Execute a linear regression for the formula: Outcome = α + β * Test Group + error. Share the results*

```
Call:
lm(formula = Outcome ~ Test_Group, data = abandon_clean)

Residuals:
     Min       1Q   Median       3Q      Max
-0.08087 -0.08087 -0.02227 -0.02227  0.97773

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.022270   0.003402   6.545 6.28e-11 ***
Test_Group  0.058602   0.004786  12.244  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2199 on 8440 degrees of freedom
Multiple R-squared:  0.01745,   Adjusted R-squared:  0.01733
F-statistic: 149.9 on 1 and 8440 DF,  p-value: < 2.2e-16
```

Outcome = 0.222 + 0.058 * Test Group + error

Null Hypothesis (H0): There is no effect of Retargeting (Test_Group) on Outcome.

Model Interpretation:

- The intercept for Test_Group is 0.058602. This value represents the difference in the expected outcome between the treatment groups (test and control). Specifically, members of the test group (when Test_Group is 1) have, on average, an outcome value that is 0.058602 higher compared to those in the control group.
- The p-value is highly significant (much smaller than 0.05), suggesting that both coefficients are statistically significant. Therefore, we can reject the null hypothesis and conclude that the retargeting treatment has an effect.

- The Multiple R-squared is 0.01745, which is quite low. This indicates that the linear regression model doesn't explain much of the variability in the outcome. Since the outcome is binary, a linear regression may not be the best model. Logistic regression would be more appropriate, as it models the probability of an event occurring by working with log-odds.

Conclusion: The retargeting campaign has a statistically significant positive effect on the outcome, as individuals in the test group have a higher outcome compared to those in the control group.

11. Justify that this regression is statistically comparable to an ANOVA/t-test.

```
> outcome_2 = aov(Outcome ~ Test_Group, data = abandon_clean)
> summary(out2)
              Df Sum Sq Mean Sq F value Pr(>F)
Test_Group     1    7.2   7.247   149.9 <2e-16 ***
Residuals   8440  408.0   0.048
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For a binary predictor, using linear regression is essentially an alternative method for performing a t-test. In fact, it is mathematically similar to conducting a two-group ANOVA. Both approaches provide different representations of the same statistical tests, focusing on differences in group means.

The coefficient for the Test_Group in the linear regression model reflects the difference in means between the two groups. This is analogous to comparing group means in an ANOVA test, where we evaluate the impact of the treatment (test group) compared to the control group.

The p-value for Test_Group in the regression model is identical to that from an ANOVA test. Both results yield extremely small p-values, indicating that the difference between the two groups is statistically significant.

The F-statistic in the regression model (149.9) is also equivalent to the F-value in ANOVA. This is because, when running a linear regression with a binary independent variable (such as Test_Group), the square of the t-statistic for the binary predictor is equal to the F-statistic that would be obtained from an ANOVA comparing the two groups.

From my analysis, the very small p-value (less than 2.2e-16) for Test_Group in the regression output, along with the large F-statistic in ANOVA, both point to a strong and statistically significant difference in the means between the control and treatment groups.

In conclusion, whether linear regression or ANOVA is used, the results lead to the same conclusion: Participation in the treatment group significantly impacts the outcome compared to the control group. This demonstrates the statistical equivalence between linear regression and ANOVA when analyzing differences in means.

12. Debate the appropriateness of the regression model in making causal claims about the retargeting campaign's efficacy.

- The results of regression analysis alone do not establish causality. A significant regression coefficient indicates an association between the variables, but it doesn't prove that one causes the other.
- While regression can highlight relationships between variables, determining causality is more complex. The difference in outcomes between the treatment and control groups does not necessarily imply that retargeting caused the change. There may be other confounding factors that the model does not account for.
- If key variables that influence the dependent variable are excluded from the analysis, the results may be skewed. This means that even if there is a significant relationship between the independent and dependent variables, the observed effect could be due to these unaccounted factors, rather than a direct causal relationship.

Therefore, while we can conclude that being retargeted increases the likelihood of a positive outcome, we cannot definitively claim that retargeting directly causes the positive result.

13. Integrate State and Email dummies into the regression. Also consider interactions with the treatment group. Compare these results to the previous regression and provide insights

	Dependent variable:			
	Outcome			
	OLS			logistic
	(1)	(2)	(3)	(4)
Test_Group	0.059***	0.058***	0.045***	1.357***
	(0.005)	(0.005)	(0.006)	(0.182)
State_Available		0.017***	0.010	0.475**
		(0.005)	(0.007)	(0.217)
Email_Available		0.036***	0.008	0.282
		(0.007)	(0.011)	(0.294)
Test_Group:State_Available			0.013	-0.153
			(0.010)	(0.247)
Test_Group:Email_Available			0.053***	0.359
			(0.015)	(0.326)
Constant	0.022***	0.011***	0.017***	-4.060***
	(0.003)	(0.004)	(0.005)	(0.161)
Observations	8,442	8,442	8,442	8,442
R2	0.017	0.023	0.025	
Adjusted R2	0.017	0.022	0.024	
Log Likelihood				-1,623.579
Akaike Inf. Crit.				3,259.158
Residual Std. Error	0.220 (df = 8440)	0.219 (df = 8438)	0.219 (df = 8436)	
F Statistic	149.904*** (df = 1; 8440)	65.259*** (df = 3; 8438)	42.654*** (df = 5; 8436)	
Note:				*p<0.1; **p<0.05; ***p<0.01

Linear and Logistic Regression Model Results

Based on the regression models (three linear and one logistic), it's evident that the campaign is having a significant positive impact. Additionally, the inclusion of state and email data boosts the campaign's effectiveness. The interaction terms highlight the value of personalized retargeting efforts.

Comparison of Models

Coefficients Overview:

- **Test_Group**: This represents the influence of being part of the test group (or treatment group).
    - **Model (1)** indicates that membership in the test group increases the outcome by 0.059 units, assuming other variables remain constant.
    - **Model (2) and Model (3)** show a slightly smaller effect, which likely reflects the adjustments made for state and email information.
    - In **Model (2)**, the outcome increases by 0.017 units when state information is included.
    - In **Model (3)**, the effect reduces further to 0.010, suggesting that both state and email details contribute to explaining the outcome.

- **Model (2)** also shows a 0.036 unit increase in the outcome when email information is available.
- However, in **Model (3)**, this effect drops to 0.008, pointing to the diminishing return once both state and email data are considered.

- **Test_Group**

: This interaction term captures how the test group effect changes when email data is available.

- **Model (3)** demonstrates a coefficient of 0.053, indicating a strong positive interaction. This suggests that the combined effect of being in the test group and having email data leads to a greater impact than if each factor acted alone.

Insights from the Retargeting Campaign:

- **Effectiveness of Test Group**: Participation in the retargeting campaign has a consistently positive impact on the outcome, reinforcing the effectiveness of the campaign.
- **State and Email Data**: When state and email data are included in the models, they enhance the effect of the retargeting campaign. This supports the idea that personalized retargeting—tailored to specific states or emails—might be more successful.
- **Significance of Interactions**: The significant positive interaction between the test group and email availability indicates that the retargeting campaign is more successful when email data is used to personalize the approach. This synergy amplifies the impact compared to when the two factors are considered separately.
- **Model Comparisons**: As we incorporate more variables and interaction terms, the effect of the test group becomes somewhat smaller. This suggests that the initial effect observed in **Model (1)** is partially explained by the inclusion of state and email information in the subsequent models.

5. Reflections

14. Reflect on the project:

• Would you modify the experiment design if given a chance.

To enhance the precision of the analysis and gain more valuable insights, I would consider the following design improvements:

- **Test Different Campaign Variations**: Instead of evaluating the overall effectiveness of retargeting, it would be beneficial to experiment with different versions of the

advertising, such as testing alternative messaging, visuals, and calls-to-action, to identify which specific aspects of the campaign are most impactful.

- **Account for External Factors**: It's important to control for external variables, such as geographical events, time zones, or seasonal trends, which can influence user behavior. For example, retargeting efforts may yield better results during peak shopping seasons or holidays in certain regions.
- **Expand Customer Demographic Information**: Understanding the relationship between demographics and campaign outcomes can offer deeper insights. For instance, age might play a significant role in determining how different age groups respond to retargeting. Knowing which segments of the population are more likely to engage could lead to more targeted and effective campaigns.
- **Use Unique Customer IDs for Better Matching**: If both datasets had a unique customer ID that could be used to match records between them, the quality of the matching process would improve. This would provide more accurate and reliable results, ensuring that we can track individual customer behavior and outcomes more precisely.

• Could alternative paths be taken with better-quality data?

In-depth Segmentation: Further segmentation analysis could be conducted by incorporating detailed demographic, behavioral, and psychographic profiles. This would help identify the specific audience segments that respond most positively to retargeting efforts.

Advanced Time-Series Analysis: A more detailed time-series analysis could be implemented if the data were collected with time stamps. This would enable a better understanding of patterns, seasonality, and the immediate versus long-term effects of retargeting.

Utilizing Machine Learning: With a larger dataset, machine learning models could be developed to predict the success of future campaigns or to personalize retargeting strategies for individual users based on their behavior and preferences.

Examining Retention and Attrition: Gathering data on user retention or attrition would offer valuable insights into how retargeting influences long-term consumer loyalty and repeat business.

Building Multi-channel Attribution Models: Incorporating data from multiple channels, such as email, social media, and direct traffic, would allow the creation of a multi-channel attribution model. This would provide a comprehensive view of the overall impact of retargeting across all marketing touchpoints.

• Are there actionable business implications from this analysis?

The analysis provides several actionable insights that can drive improvements in future campaigns. The retargeting efforts have proven effective, so scaling up these campaigns by allocating a larger budget could significantly enhance results. Additionally, continuously refining retargeting criteria based on real-time feedback will help ensure the highest impact.

In terms of data collection, focusing on gathering users' state and email information is crucial, as these factors seem to enhance the effectiveness of the campaign. Expanding the types of data collected can further personalize the retargeting efforts.

Key recommendations for moving forward include:

- **Personalized Campaigns**: Use state and email data to segment the audience and create tailored retargeting messages for each group.
- **Leveraging Interaction Effects**: Take advantage of the positive interaction between test group status and email availability, tailoring approaches based on whether email data is available for users.
- **Integrated Marketing**: Synchronize retargeting efforts with other channels, such as emails, to amplify the effect of each touchpoint.
- **Resource Reallocation**: Based on campaign success, consider redistributing the marketing budget to favor retargeting over less effective strategies.

Finally, regularly revisiting the analysis and updating strategies based on fresh data will be key to maintaining long-term success and improving the efficiency of retargeting campaigns.

15. Self-assessment: Rate your effort (0-100) and anticipated performance. Elaborate if needed, mentioning collaborations.

I would rate my effort as an 85. While the template provided valuable guidance, I went beyond it by investing time to understand the business context, performance-based activities, and customer retargeting strategies. I also explored additional marketing concepts like churn analysis and how to effectively target and convert potential customers. This exercise gave me a solid understanding of marketing campaigns and how I can apply my class learnings to real-world problems. Though the template made some tasks easier, the adjustments I made and the knowledge gained were significant, so I’m not rating myself a full 100.

R Script

```
library(dplyr)

library(stargazer)


#Verify the presence of any missing values.

sum(is.na(Abandoned))

sum(is.na(Reservation))


#Examine the dataset for duplicate entries.

sum(duplicated(Abandoned))

sum(duplicated(Abandoned$Caller_ID))


sum(duplicated(Reservation))

sum(duplicated(Reservation$Caller_ID))


#Analysis of Test and Control Group Distribution

sum(Abandoned$Test_Control == "test" )

sum(Abandoned$Test_Control == "control")

sum(Reservation$Test_Control == "test" )

sum(Reservation$Test_Control == "control")


# Load ggplot2 library

library(ggplot2)
```

```
# Create the data frame with counts
data <- data.frame(
  Group = c("Test", "Control", "Test", "Control"),
  Dataset = c("Abandoned", "Abandoned", "Reservation", "Reservation"),
  Count = c(4266, 4176, 18728, 2086)
)

# Create the bar chart with count labels
ggplot(data, aes(x = Dataset, y = Count, fill = Group)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(aes(label = Count),
            position = position_dodge(width = 0.9),
            vjust = -0.3, # Adjusts the vertical position of the text
            size = 3) + # Size of the text
  labs(title = "Counts of Test and Control Groups in Abandoned and Reservation Datasets",
       x = "Dataset",
       y = "Count") +
  theme_minimal() +
  scale_fill_manual(values = c("Test" = "blue", "Control" = "orange"))



#Segmented by State Availability
if_state <- Abandoned[complete.cases(Abandoned['Address']),]
table(if_state$Test_Control)
```

```
stats_overview = Abandoned %>%
  group_by(Address, Test_Control) %>%
  summarize(Count = n())

print(stats_overview)

Reservation_stats = Reservation %>%
  group_by(Address, Test_Control) %>%
  summarize(Count = n())

print(Reservation_stats)
#The Abandoned dataset shows a relatively balanced distribution overall, as well as at the state level.
#The Reserved dataset displays an uneven distribution both overall and at the state level.


#Perform matching using different keys and create logical vectors to represent each condition.
emailmatch = Abandoned$Email[complete.cases(Abandoned$Email)] %in%
Reservation$Email[complete.cases(Reservation$Email)]
incom_match = Abandoned$Incoming_Phone[complete.cases(Abandoned$Incoming_Phone)] %in%
Reservation$Incoming_Phone[complete.cases(Reservation$Incoming_Phone)]
contactmatch = Abandoned$Contact_Phone[complete.cases(Abandoned$Contact_Phone)] %in%
Reservation$Contact_Phone[complete.cases(Reservation$Contact_Phone)]
incom_contact_match = Abandoned$Incoming_Phone[complete.cases(Abandoned$Incoming_Phone)]
%in% Reservation$Contact_Phone[complete.cases(Reservation$Contact_Phone)]
contact_incom_match = Abandoned$Contact_Phone[complete.cases(Abandoned$Contact_Phone)]
%in% Reservation$Incoming_Phone[complete.cases(Reservation$Incoming_Phone)]
```

```
#Generate flags for matched records


Abandoned$emailmatch = 0

Abandoned$emailmatch[complete.cases(Abandoned$Email)] = 1 * emailmatch


Abandoned$incom_match = 0

Abandoned$incom_match[complete.cases(Abandoned$Incoming_Phone)] = 1 * incom_match


Abandoned$contactmatch= 0

Abandoned$contactmatch[complete.cases(Abandoned$Contact_Phone)] = 1 * contactmatch


Abandoned$incom_contact_match= 0

Abandoned$incom_contact_match[complete.cases(Abandoned$Incoming_Phone)] = 1 *
incom_contact_match


Abandoned$contact_incom_match= 0

Abandoned$contact_incom_match[complete.cases(Abandoned$Contact_Phone)] = 1 *
contact_incom_match



# Logical filter for matched records of customers who purchased

Abandoned$pur = 1 * ( Abandoned$emailmatch | Abandoned$incom_match
|Abandoned$contactmatch | Abandoned$incom_contact_match | Abandoned$contact_incom_match)


# Create additional columns for analyses

Abandoned$email = 1 * complete.cases(Abandoned$Email)

Abandoned$state = 1 * complete.cases(Abandoned$Address)
```

```
Abandoned$treat = ifelse(Abandoned$Test_Control == "test", 1, 0)


tab = table(Abandoned$pur, Abandoned$treat)
# Adding row labels for 'Outcome'
rownames(tab) = c("Not Purchased", "Purchased")
# Adding column labels for 'Treatment'
colnames(tab) = c("Control Group", "Treatment Group")
print(tab)


#Filter out unmatched records
unmatchable_Abandoned <- Abandoned[Abandoned$pur == 0, ]
head(unmatchable_Abandoned)


#Dropping unmatching data and selecting matched (purchased) records
Abandoned_match <- Abandoned[Abandoned$pur == 1, ]


#Cross-tabulate both purchased and non-purchased records
tab = table(Abandoned$pur, Abandoned$treat)
rownames(tab) = c("Not Purchased", "Purchased")
# Add column labels for 'Outcome'
colnames(tab) = c("Control Group", "Treatment Group")
print(tab)



all_states = Abandoned$Address[!is.na(Abandoned$Address)]
set.seed(123)  # Setting a seed for reproducibility
```

```
random_states = sample(all_states, 5)


cross_tabulations = list()


for (state in random_states) {
  subset_data = Abandoned[Abandoned$Address == state, ]
  cross_tabulation =  table(subset_data$pur, subset_data$treat)
  rownames( cross_tabulation ) =  c("Not Purchased", "Purchased")
  colnames( cross_tabulation ) = c("Control Group", "Treatment Group")
  cross_tabulations[[state]] = cross_tabulation
}


# Print the cross-tabulations
for (state in random_states) {
  cat("Cross-tabulation for State:", state, "\n")
  print(cross_tabulations[[state]])
  cat("\n")
}


#Cleaning dataset
# Remove multiple columns
abandon_clean = Abandoned %>%
  select(-(2:17))


#Changing index of columns and their column names
abandon_clean = abandon_clean %>%
```

```
  select(1, 5, 2, 4,3:ncol(abandon_clean))

colnames(abandon_clean) = c("Customer_ID", "Test_Group","Outcome","State_Available",
                            "Email_Available")


#Exporting the clean data set as a csv

write.csv(abandon_clean, file = "abandon_clean.csv", row.names = FALSE)




#Statistical tests

# Run regression analyses


outcome_1 = lm(Outcome ~ Test_Group, data = abandon_clean)

summary(outcome_1)

outcome_2 = aov(Outcome ~ Test_Group, data = abandon_clean)

summary(outcome_2)

outcome_3 = lm(Outcome ~ Test_Group + State_Available + Email_Available , data = abandon_clean)

summary(outcome_3)

outcome_4 = lm(Outcome ~ Test_Group + State_Available + Email_Available +
State_Available*Test_Group + Email_Available*Test_Group, data = abandon_clean)

summary(outcome_4)


#logistic model

logmodel = glm(Outcome ~ Test_Group + State_Available + Email_Available +
State_Available*Test_Group + Email_Available*Test_Group ,family = binomial(link="logit"), data =
abandon_clean)

summary(logmodel)
```

```
# Generate summary table

stargazer(outcome_1,outcome_3,outcome_4,logmodel, type = "text")
```