Appendix II - R Code for Enhancing Patient Engagement in Preventative Care: A Field Experiment on the Effectiveness of Personalized and Generic Digital Outreach Strategies

Lynne Wang, Mridul Jain, Jesse Cox, and Tejasvi Kalakota

04/18/2025

```
# Load and preview the first few rows of the data
# This gives a quick view of variables like variant_assignment, unsubscribed, opened, click, and visit
print(head(data))
```

```
##       id variant_assignment   last_seen unsubscribed opened click visit
##    <int>             <char>      <char>        <int>  <int> <int> <int>
## 1:    21            placebo   1/17/2025            1      1     0     0
## 2:    41  treatment_generic a while ago            1      1     0     0
## 3:    44            placebo   9/13/2023            1      1     0     0
## 4:    78  treatment_generic a while ago            1      1     0     0
## 5:   105  treatment_generic a while ago            1      1     0     0
## 6:   190            placebo a while ago            1      1     0     0
```

```
d <- data
```

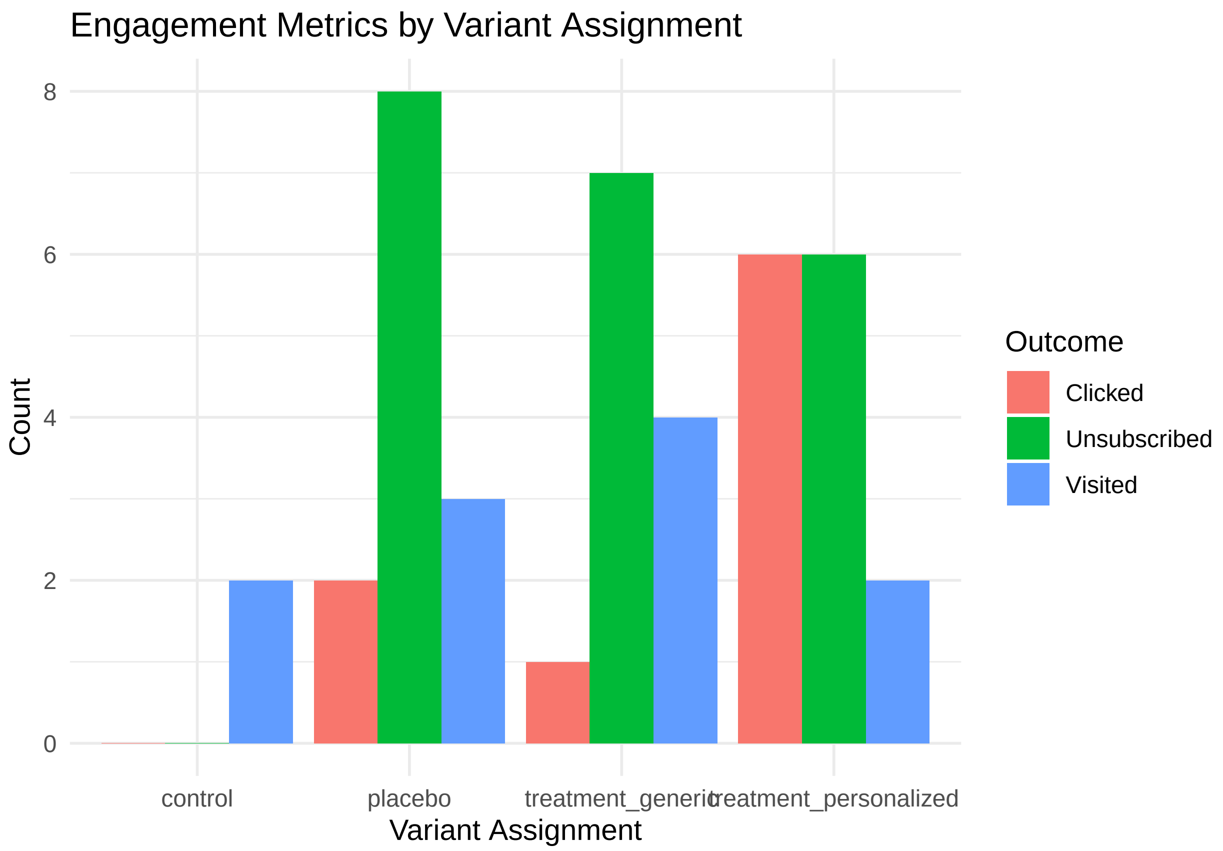
```
# Count the number of patients assigned to each variant group
# Helps confirm balance of randomization across control and treatment groups
data %>%
  count(variant_assignment)
```

```
##        variant_assignment     n
##                    <char> <int>
## 1:                control   108
## 2:                placebo   132
## 3:      treatment_generic   120
## 4: treatment_personalized   109
```

```
# Summarize click behavior by treatment arm
# Outputs total clicks and average click rate per group
d %>%
  group_by(variant_assignment) %>%
  summarise(n = n(), clicks = sum(click, na.rm = TRUE), click_rate = mean(click, na.rm = TRUE))
```

```
## # A tibble: 4 x 4
##   variant_assignment         n clicks click_rate
##   <chr>                  <int>  <int>      <dbl>
## 1 control                  108      0     0
## 2 placebo                  132      2     0.0152
## 3 treatment_generic        120      1     0.00833
## 4 treatment_personalized   109      6     0.0550
```

```
# Summarize key engagement outcomes by treatment group
# Includes counts of clicks, visits, and unsubscribes for plotting
summary_counts <- d %>%
  group_by(variant_assignment) %>%
  summarise(
    Clicked = sum(click),
    Visited = sum(visit),
    Unsubscribed = sum(unsubscribed)
  ) %>%
  pivot_longer(
    cols = c(Clicked, Visited, Unsubscribed),
    names_to = "Outcome",
    values_to = "Count"
  )

# Plot bar chart comparing counts of engagement metrics by variant group
# Visualizes Clicked, Visited, and Unsubscribed across Control, Placebo, and Treatment groups
ggplot(summary_counts, aes(x = variant_assignment, y = Count, fill = Outcome)) +
  geom_bar(stat = "identity", position = position_dodge()) +
  labs(title = "Engagement Metrics by Variant Assignment", x = "Variant Assignment", y = "Count") +
  theme_minimal()
```



```r
# Filter data to include only treatment-related groups (excluding control)
# Used for comparing email open rates across different outreach strategies
data_subset <- data %>%
  filter(variant_assignment %in% c("placebo", "treatment_generic", "treatment_personalized"))

# Estimate linear model to compare open rates across variant_assignment
# Robust standard errors account for heteroskedasticity
model_opened <- lm(opened ~ variant_assignment, data = data_subset)
robust_se <- coeftest(model_opened, vcov = vcovHC(model_opened, type = "HC1"))
print(robust_se)
```

```
##
## t test of coefficients:
##
##                                           Estimate Std. Error  t value Pr(>|t|)
## (Intercept)                              0.9924242  0.0075786 130.9515   <2e-16
## variant_assignmenttreatment_generic      0.0075758  0.0075786   0.9996   0.3182
## variant_assignmenttreatment_personalized 0.0075758  0.0075786   0.9996   0.3182
##
## (Intercept)                              ***
## variant_assignmenttreatment_generic
## variant_assignmenttreatment_personalized
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Intention-to-treat analysis: calculate open rate and sample size per variant
# Used to assess compliance and attrition
itt_open <- data_subset %>%
  group_by(variant_assignment) %>%
  summarise(open_rate = mean(opened == 1), count = n())
print(itt_open)
```

```
## # A tibble: 3 x 3
##   variant_assignment     open_rate count
##   <chr>                      <dbl> <int>
## 1 placebo                    0.992   132
## 2 treatment_generic          1       120
## 3 treatment_personalized     1       109
```

```r
# Filter to just treatment_generic and treatment_personalized groups
# Fit model to compare click rates between personalized and generic emails
data_subset2 <- data %>%
  filter(variant_assignment %in% c("treatment_generic", "treatment_personalized"))

model_click <- lm(click ~ variant_assignment, data = data_subset2)
coeftest(model_click, vcov = vcovHC(model_click, type = "HC1"))
```

```
##
## t test of coefficients:
##
##                                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                              0.0083333  0.0083350  0.9998  0.31847
## variant_assignmenttreatment_personalized 0.0467125  0.0234710  1.9902  0.04777
##
## (Intercept)
## variant_assignmenttreatment_personalized *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Output summary with robust standard errors (HC1)
coeftest(model_click, vcov = vcovHC(model_click, type = "HC1"))
```

```
##
## t test of coefficients:
##
##                                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                              0.0083333  0.0083350  0.9998  0.31847
## variant_assignmenttreatment_personalized 0.0467125  0.0234710  1.9902  0.04777
##
## (Intercept)
## variant_assignmenttreatment_personalized *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Filter to just control, treatment_generic and treatment_personalized groups
# Fit model to compare visit rates between personalized and generic emails

data_subset2 <- data %>%
  filter(variant_assignment %in% c("control", "treatment_generic", "treatment_personalized"))

print(data_subset2)
```

```
##         id     variant_assignment   last_seen unsubscribed opened click visit
##      <int>                 <char>      <char>        <int>  <int> <int> <int>
##   1:    41      treatment_generic a while ago            1      1     0     0
##   2:    78      treatment_generic a while ago            1      1     0     0
##   3:   105      treatment_generic a while ago            1      1     0     0
##   4:   201 treatment_personalized a while ago            1      1     0     0
##   5:   221 treatment_personalized a while ago            1      1     0     0
##  ---
## 333:   463 treatment_personalized a while ago            0      1     0     0
## 334:   464      treatment_generic a while ago            0      1     0     0
## 335:   465                control a while ago            0      0     0     0
## 336:   467      treatment_generic a while ago            0      1     0     0
## 337:   468 treatment_personalized a while ago            0      1     0     0
```

```r
model_click <- lm(visit ~ variant_assignment, data = data_subset2)

# Output summary with robust standard errors (HC1)
coeftest(model_click, vcov = vcovHC(model_click, type = "HC1"))
```

```
##
## t test of coefficients:
##
```

```
##                                              Estimate  Std. Error t value
## (Intercept)                               0.01851852  0.01303089  1.4211
## variant_assignmenttreatment_generic       0.01481481  0.02099368  0.7057
## variant_assignmenttreatment_personalized -0.00016989  0.01834491 -0.0093
##                                          Pr(>|t|)
## (Intercept)                                0.1562
## variant_assignmenttreatment_generic        0.4809
## variant_assignmenttreatment_personalized   0.9926
```

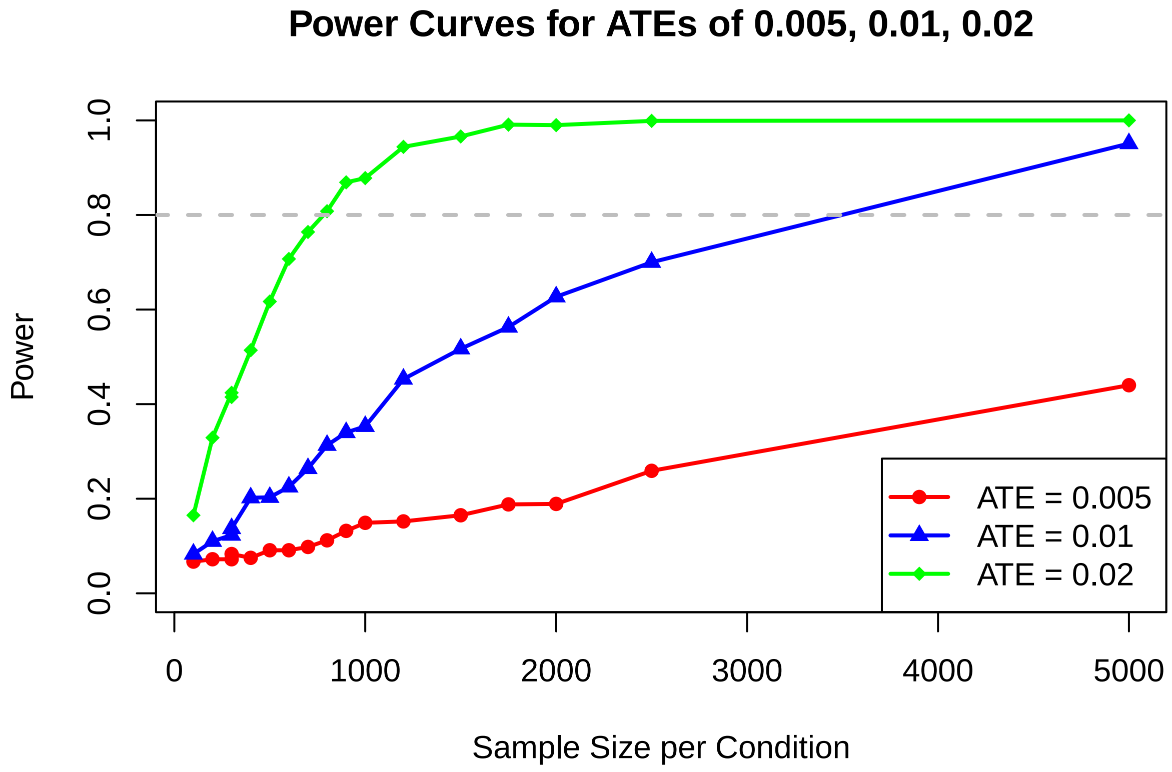
```
# Add covariate for recency of healthcare interaction, and generate interaction term with treatment.
# Enables HTE analysis to see if treatment works differently for patients not recently seen.

today <- as.Date("2025-04-15")

d <- d %>%
  mutate(last_seen = ifelse(last_seen == "a while ago", '04/01/2024', last_seen))

d <- d %>%
  mutate(
    # Step 2: Convert from m/d/Y to Date
    last_seen_clean = mdy(last_seen),

    # Step 3: Compute days since seen
    days_since_seen = as.numeric(Sys.Date() - last_seen_clean),

    # Step 4: Define seen_recently as   365 days
    seen_recently = ifelse(days_since_seen <= 365, 1, 0)
  )
```

```
# Fit a model including an interaction between treatment assignment and not_seen_recently.
# Helps test whether timing of last engagement modifies treatment effects.

# Set 'not seen recently' (0) as the reference level
# Create a new variable: not_seen_recently
d$not_seen_recently <- as.numeric(d$seen_recently == 0)

# Filter out the placebo group
d_filtered <- d %>% filter(variant_assignment %in% c("control", "treatment_generic", "treatment_personal

# Create a new variable: not_seen_recently
d_filtered$not_seen_recently <- as.numeric(d_filtered$seen_recently == 0)

# Fit the model using the new variable
model <- lm(visit ~ variant_assignment * not_seen_recently, data = d_filtered)
robust_se <- vcovHC(model, type = "HC1")
robust_results <- coeftest(model, vcov = robust_se)
print(robust_results)
```

```
##
## t test of coefficients:
##
##                                                              Estimate Std. Error
## (Intercept)                                                  0.083333   0.056926
## variant_assignmenttreatment_generic                        -0.018817   0.072269
## variant_assignmenttreatment_personalized                   -0.011905   0.075182
## not_seen_recently                                          -0.083333   0.056926
## variant_assignmenttreatment_generic:not_seen_recently       0.041289   0.073987
## variant_assignmenttreatment_personalized:not_seen_recently  0.011905   0.075182
##                                                            t value Pr(>|t|)
## (Intercept)                                                 1.4639   0.1442
## variant_assignmenttreatment_generic                        -0.2604   0.7947
## variant_assignmenttreatment_personalized                   -0.1583   0.8743
## not_seen_recently                                          -1.4639   0.1442
## variant_assignmenttreatment_generic:not_seen_recently       0.5581   0.5772
## variant_assignmenttreatment_personalized:not_seen_recently  0.1583   0.8743
```

```
# Simulate statistical power at different sample sizes for three effect sizes: 0.005, 0.01, 0.02.
# Helps visualize sensitivity of the study to detect effects under realistic assumptions.

set.seed(3)

power_test_t <- function(
  mean_control = 0.02,
  mean_treat = 0.03,
  sd_control = sqrt(0.02 * 0.98),
  sd_treat = sqrt(0.02 * 0.98),
  number_per_condition = 40,
  power_loops = 1000,
  verbose = FALSE) {
    p_values <- numeric(power_loops)
    for (i in 1:power_loops) {
        control_group <- rnorm(number_per_condition, mean = mean_control, sd = sd_control)
        treatment_group <- rnorm(number_per_condition, mean = mean_treat, sd = sd_treat)

        test_result <- t.test(control_group, treatment_group, var.equal = FALSE)
        p_values[i] <- test_result$p.value
    }

    power_estimate <- mean(p_values < 0.05)
    return(list('p_values' = p_values, 'power' = power_estimate))
}
```

```
# Generate line plot showing how sample size impacts power for each effect size.
# A horizontal line at 0.8 indicates the standard threshold for acceptable power.

# Sample sizes
samples_per_condition <- c(100, 200, 300, 300, 400, 500, 600, 700, 800, 900, 1000, 1200, 1500, 1750, 200

# Calculate power for treatment effects of +0.005, +0.01, +0.02
power_005 <- numeric(length(samples_per_condition))
power_01 <- numeric(length(samples_per_condition))
power_02 <- numeric(length(samples_per_condition))

for(i in seq_along(samples_per_condition)) {
  power_005[i] <- power_test_t(mean_control = 0.02, mean_treat = 0.025,
                                number_per_condition = samples_per_condition[i])$power
  power_01[i] <- power_test_t(mean_control = 0.02, mean_treat = 0.03,
                               number_per_condition = samples_per_condition[i])$power
  power_02[i] <- power_test_t(mean_control = 0.02, mean_treat = 0.04,
                               number_per_condition = samples_per_condition[i])$power
}

# Assemble power table
power_table <- data.frame(
  "Sample Size" = samples_per_condition,
  "ATE 0.005" = power_005,
  "ATE 0.01" = power_01,
  "ATE 0.02" = power_02
)

# Plot results
plot(samples_per_condition, power_005, type = "o", col = "red", lwd = 2, pch = 16, ylim = c(0, 1),
     xlab = "Sample Size per Condition", ylab = "Power", main = "Power Curves for ATEs of 0.005, 0.01, 
lines(samples_per_condition, power_01, col = "blue", lwd = 2, type = "o", pch = 17)
lines(samples_per_condition, power_02, col = "green", lwd = 2, type = "o", pch = 18)
abline(h = 0.8, col = "gray", lty = 2, lwd = 2)
legend("bottomright", legend = c("ATE = 0.005", "ATE = 0.01", "ATE = 0.02"),
       col = c("red", "blue", "green"), lwd = 2, pch = c(16, 17, 18))
```


Power Curves for ATEs of 0.005, 0.01, 0.02
Power
Sample Size per Condition
ATE = 0.005
ATE = 0.01
ATE = 0.02