Home Work 4

Utkarsh Joshi - 5982808

Introduction

This assignment explores three causal inference methods applied to real-world scenarios. Question 1 uses propensity score matching to estimate the effect of directed search behavior on online retail sales. Question 2 applies the synthetic control method to evaluate the impact of California's cigarette tax on sales. Question 3 implements a regression discontinuity design to measure the effect of ad rank on click-through rates in online auctions. Each question is analyzed using R, with a focus on model interpretation and causal insights.

Assumptions for Each Method:

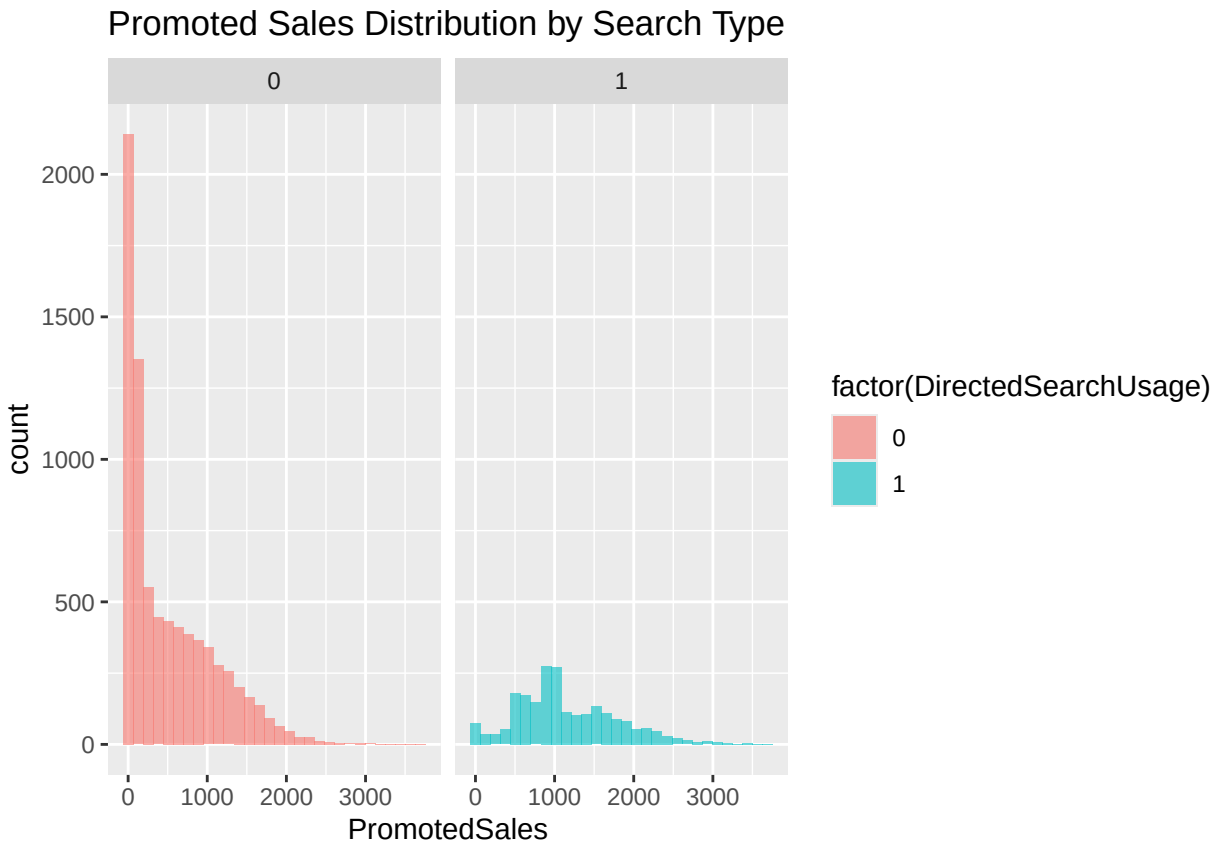
1. Propensity Score Matching (PSM): Assumes unconfoundedness, meaning that all relevant confounders are observed and accounted for through covariates. Also assumes overlap, ensuring that treated and control units are comparable—this is enforced through the caliper threshold.
2. Log-Linear Regression: Assumes a log-linear relationship between the outcome and the treatment variable. The log transformation addresses skewness and improves model fit, assuming constant percentage changes.
3. Synthetic Control: Relies on exogeneity, where the treatment (California's tax policy) is unrelated to unobserved factors that affect cigarette sales. Also assumes a strong pre-treatment fit—i.e., the synthetic control closely mirrors California's trends before the policy, supporting credibility of the counterfactual.
4. Regression Discontinuity Design (RDD): Requires continuity in all other determinants of the outcome around the cutoff. Only the treatment status (top ad rank) should change discontinuously. Also assumes local randomization, meaning near the cutoff, treatment assignment is effectively random, enabling causal interpretation of the jump.

QUESTION 1

General Analysis

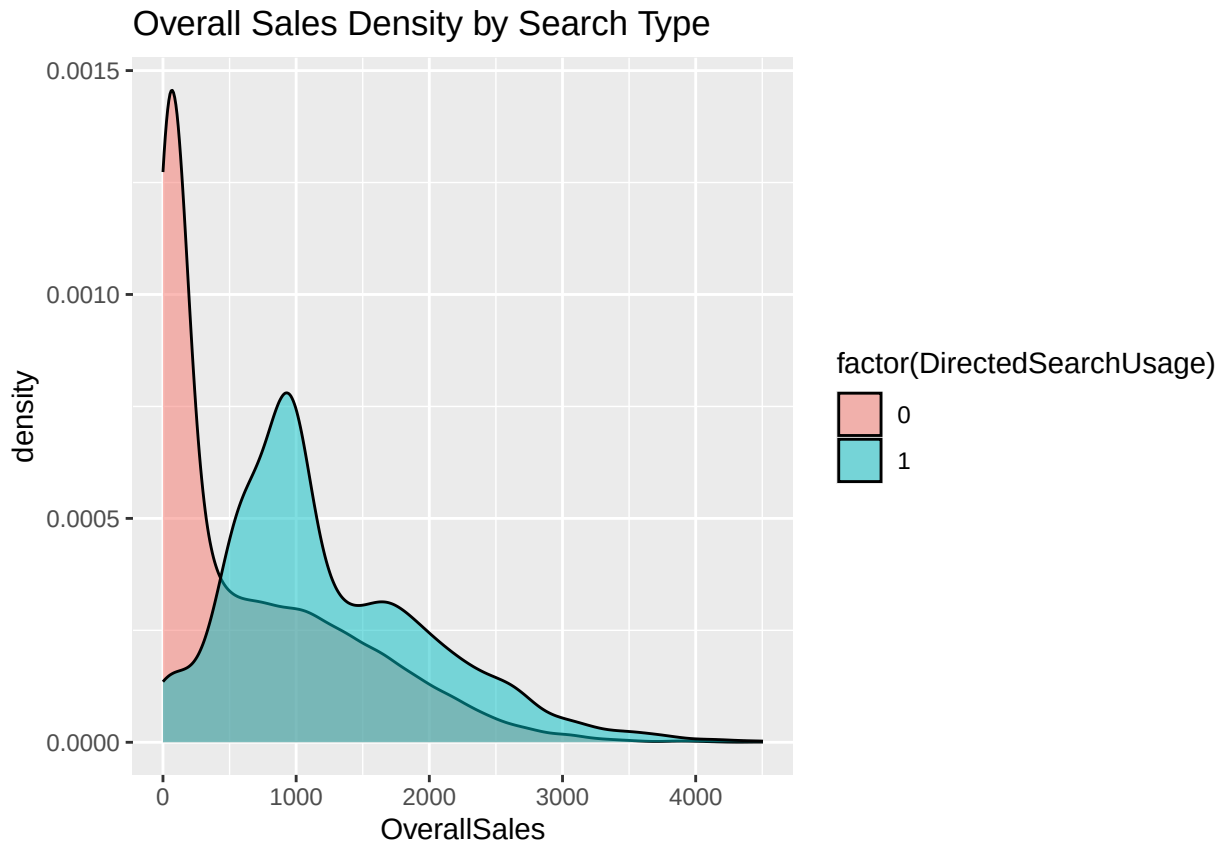
```
setwd("C:/Users/91788/Downloads")

match_df <- read.csv("matching.csv")
```



- Undirected search (0) shows a heavy right-skew with many low or zero promoted sales. - Directed search (1) has higher and more evenly distributed promoted sales.

```
ggplot(match_df, aes(x = OverallSales, fill = factor(DirectedSearchUsage))) +
  geom_density(alpha = 0.5) +
  labs(title = "Overall Sales Density by Search Type")
```

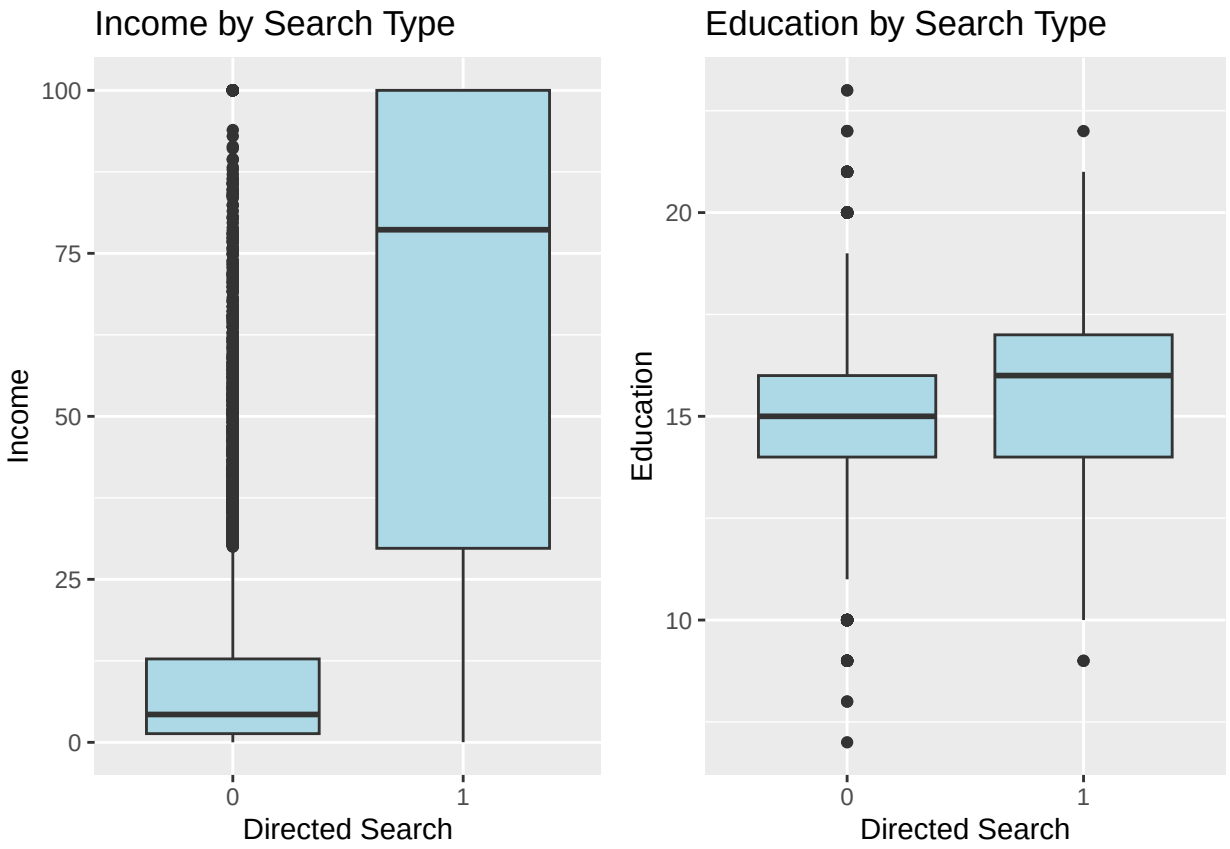


- Undirected search (red) peaks sharply near $0, indicating low overall purchases for most users. - Directed search (blue) has a broader, flatter peak around $1000, with a heavier right tail.

```
p1 <- ggplot(match_df, aes(x = factor(DirectedSearchUsage), y = Income)) +
  geom_boxplot(fill = "lightblue") +
  labs(title = "Income by Search Type", x = "Directed Search", y = "Income")

p2 <- ggplot(match_df, aes(x = factor(DirectedSearchUsage), y = Education)) +
  geom_boxplot(fill = "lightblue") +
  labs(title = "Education by Search Type", x = "Directed Search", y = "Education")

grid.arrange(p1, p2, ncol = 2)
```



- Income: Directed search users (1) have significantly higher income than undirected users (0), with a much wider spread and fewer low-income individuals.
- Education: Directed users also have slightly higher education levels, with a higher median and slightly broader range.

T-Test Analysis: Directed vs. Undirected Search Sessions

```
t.test(PromotedSales ~ DirectedSearchUsage, data = match_df)
```

```
##
##  Welch Two Sample t-test
##
## data:  PromotedSales by DirectedSearchUsage
## t = -43.483, df = 3283.3, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -679.5183 -620.8815
## sample estimates:
## mean in group 0 mean in group 1
##        510.1724       1160.3723
```

```r
t.test(NonpromotedSales ~ DirectedSearchUsage, data = match_df)
```

```
##
##  Welch Two Sample t-test
##
## data:  NonpromotedSales by DirectedSearchUsage
## t = 18.394, df = 4678.4, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  59.17314 73.29179
## sample estimates:
## mean in group 0 mean in group 1
##       152.19720        85.96473
```

```r
t.test(OverallSales ~ DirectedSearchUsage, data = match_df)
```

```
##
##  Welch Two Sample t-test
##
## data:  OverallSales by DirectedSearchUsage
## t = -32.01, df = 3530.1, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -619.7359 -548.1990
## sample estimates:
## mean in group 0 mean in group 1
##        662.3696       1246.3370
```

We conducted Welch two-sample t-tests to compare sales outcomes between Directed (1) and Undirected (0) search sessions across three sales metrics:

- Overall Sales: Directed sessions show significantly higher average sales ($1246 vs. $662), with a tight 95% CI [–619.74, –548.20] and $p < 0.001$.
- Promoted Sales: Directed users spend substantially more on promoted items ($1160 vs. $510), supported by a CI [–679.52, –620.88] and strong significance.
- Non-Promoted Sales: Interestingly, undirected sessions generate more non-promoted sales ($152 vs. $86), suggesting exploratory purchase behavior.

All differences are highly statistically significant ($p < 0.001$), but these estimates may reflect underlying group differences, not causal effects. Hence, we next apply propensity score matching to balance covariates and isolate the true impact of directed search.

Q1(a) Naïve Log-Linear Regression: Effect of Directed Search on Sales Outcomes

To account for skewness and zero values in the sales distributions, we estimate log-linear models for each sales variable by regressing log(Sales + 1) on DirectedSearchUsage. This allows us to interpret coefficients in percentage terms after exponentiation.

Model 1: Overall Sales

```r
model1 <- lm(log(OverallSales + 1) ~ DirectedSearchUsage, data = match_df)
summary(model1)
```

```
##
## Call:
## lm(formula = log(OverallSales + 1) ~ DirectedSearchUsage, data = match_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.7724 -0.7368  0.3776  1.6080  3.2296
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5.17956    0.02463  210.32   <2e-16 ***
## DirectedSearchUsage  1.59287    0.05202   30.62   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.169 on 9998 degrees of freedom
## Multiple R-squared:  0.08573,    Adjusted R-squared:  0.08564
## F-statistic: 937.5 on 1 and 9998 DF,  p-value: < 2.2e-16
```

- Directed search is associated with a 1.593 unit increase in the log of OverallSales + 1, which translates to an approximate 392% increase in overall sales compared to undirected sessions.
- This effect is highly statistically significant ($p < 0.001$).

Model 2: Promoted Sales

```r
model2 <- lm(log(PromotedSales + 1) ~ DirectedSearchUsage, data = match_df)
summary(model2)
```

```
##
## Call:
## lm(formula = log(PromotedSales + 1) ~ DirectedSearchUsage, data = match_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.7309 -0.6833  0.4071  1.5436  3.1575
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          4.97592    0.02392  207.99   <2e-16 ***
## DirectedSearchUsage  1.75498    0.05054   34.73   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.107 on 9998 degrees of freedom
## Multiple R-squared:  0.1076, Adjusted R-squared:  0.1075
## F-statistic:  1206 on 1 and 9998 DF,  p-value: < 2.2e-16
```

- Directed search leads to a 1.755 unit increase in the log of PromotedSales + 1, implying an approximate 478% increase in promoted sales.

- Again, the result is highly statistically significant ($p < 0.001$), suggesting strong intent-driven purchasing behavior.

Model 3: Non-Promoted Sales

```
model3 <- lm(log(NonpromotedSales + 1) ~ DirectedSearchUsage, data = match_df)
summary(model3)
```

```
##
## Call:
## lm(formula = log(NonpromotedSales + 1) ~ DirectedSearchUsage,
##     data = match_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5938 -2.1150  0.1486  2.0775  4.5842
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           3.59385    0.02601  138.17   <2e-16 ***
## DirectedSearchUsage -1.47886    0.05494  -26.92   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.291 on 9998 degrees of freedom
## Multiple R-squared:  0.06757,    Adjusted R-squared:  0.06747
## F-statistic: 724.5 on 1 and 9998 DF,  p-value: < 2.2e-16
```

- The coefficient of –1.479 indicates that directed search reduces non-promoted sales by approximately 77%, highlighting that users engaging in directed search may skip spontaneous or non-promoted purchases.
- The negative effect is statistically significant ($p < 0.001$).

Q1(b) Propensity Score Matching: Estimating Causal Effect of Directed Search

To estimate the causal impact of directed search behavior on sales outcomes, we apply **nearest neighbor matching (1:1, no replacement)** using a logistic regression model. The propensity score is estimated using the following covariates:

- `log(Income)`, `Education`, `NumSessions`, `DaysSinceLastPurchase`, and `HistoricalTotalPurchases`.

We apply a **tight caliper of 0.001** to ensure high-quality matches.

```
match_output <- matchit(DirectedSearchUsage ~ log(Income) + Education + NumSessions +
                          DaysSinceLastPurchase + HistoricalTotalPurchases,
                          data = match_df, method = "nearest",
                          distance = "glm", link = "logit",
                          caliper = 0.001, ratio = 1, replace = FALSE)

summary(match_output)
```

```
##
## Call:
## matchit(formula = DirectedSearchUsage ~ log(Income) + Education +
##     NumSessions + DaysSinceLastPurchase + HistoricalTotalPurchases,
##     data = match_df, method = "nearest", distance = "glm", link = "logit",
##     replace = FALSE, caliper = 0.001, ratio = 1)
##
## Summary of Balance for All Data:
##                           Means Treated Means Control Std. Mean Diff. Var. Ratio
## distance                         0.5906        0.1182          1.7730     2.3001
## log(Income)                      3.8034        1.3472          2.0759     0.5355
## Education                       15.4641       14.8536          0.3022     1.0197
## NumSessions                      5.0156        5.0447         -0.0132     0.9675
## DaysSinceLastPurchase            4.5695        4.4481          0.0206     0.9895
## HistoricalTotalPurchases         4.4289        4.4666         -0.0065     1.0011
##                           eCDF Mean eCDF Max
## distance                     0.4126   0.6764
## log(Income)                  0.4229   0.6622
## Education                    0.0360   0.1273
## NumSessions                  0.0038   0.0110
## DaysSinceLastPurchase        0.0095   0.0239
## HistoricalTotalPurchases     0.0058   0.0183
##
## Summary of Balance for Matched Data:
##                           Means Treated Means Control Std. Mean Diff. Var. Ratio
## distance                         0.3590        0.3590          0.0001     1.0002
## log(Income)                      2.9481        2.9581         -0.0085     1.4760
## Education                       15.3363       15.2817          0.0270     1.1645
## NumSessions                      5.0724        4.9811          0.0414     1.1004
## DaysSinceLastPurchase            4.6886        4.3108          0.0640     1.1082
## HistoricalTotalPurchases         4.5828        4.4735          0.0189     1.5203
##                           eCDF Mean eCDF Max Std. Pair Dist.
## distance                     0.0001   0.0033          0.0003
## log(Income)                  0.0350   0.1091          0.4209
## Education                    0.0120   0.0457          1.1713
## NumSessions                  0.0109   0.0412          1.1170
## DaysSinceLastPurchase        0.0183   0.0401          0.8151
## HistoricalTotalPurchases     0.0132   0.0356          0.8443
##
## Sample Sizes:
##           Control Treated
## All          7759    2241
## Matched       898     898
## Unmatched    6861    1343
## Discarded       0       0
```

```
matched_data <- match.data(match_output)
```

The matched dataset includes 898 treated and 898 control units (total = 1,796). Matching substantially reduces imbalance across covariates, as seen in the standardized mean differences.

Outcome Regression on Matched Sample

Model 1: Overall Sales

```r
matched_model1 <- lm(log(OverallSales + 1) ~ DirectedSearchUsage, data = matched_data)
summary(matched_model1)
```

```
##
## Call:
## lm(formula = log(OverallSales + 1) ~ DirectedSearchUsage, data = matched_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5073 -0.2663  0.3748  1.1493  2.8568
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5.45845    0.06664   81.91   <2e-16 ***
## DirectedSearchUsage  1.04883    0.09424   11.13   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.997 on 1794 degrees of freedom
## Multiple R-squared:  0.06458,    Adjusted R-squared:  0.06406
## F-statistic: 123.9 on 1 and 1794 DF,  p-value: < 2.2e-16
```

```r
100 * (exp(coef(matched_model1)[2]) - 1)
```

```
## DirectedSearchUsage
##            185.4296
```

- After matching, directed search is associated with a 1.049 unit increase in log-transformed overall sales, translating to a 185.4% increase in average sales compared to undirected sessions.
- This effect is statistically significant ($p < 0.001$) and now reflects a more credible causal estimate.

Model 2: Promoted Sales

```r
matched_model2 <- lm(log(PromotedSales + 1) ~ DirectedSearchUsage, data = matched_data)
summary(matched_model2)
```

```
##
## Call:
## lm(formula = log(PromotedSales + 1) ~ DirectedSearchUsage, data = matched_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4624 -0.2117  0.4251  1.0703  2.7322
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5.30571    0.06481   81.86   <2e-16 ***
## DirectedSearchUsage  1.15673    0.09166   12.62   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.942 on 1794 degrees of freedom
## Multiple R-squared:  0.08154,    Adjusted R-squared:  0.08103
## F-statistic: 159.3 on 1 and 1794 DF,  p-value: < 2.2e-16
```

```
100 * (exp(coef(matched_model2)[2]) - 1)
```

```
## DirectedSearchUsage
##            217.9524
```

- Directed search leads to an estimated 218% increase in promoted sales on average (coefficient = 1.157).
- The result is statistically significant ($p < 0.001$), suggesting a strong effect even after adjusting for user-level confounding.

Model 3: Non-Promoted Sales

```
matched_model3 <- lm(log(NonpromotedSales + 1) ~ DirectedSearchUsage, data = matched_data)
summary(matched_model3)
```

```
##
## Call:
## lm(formula = log(NonpromotedSales + 1) ~ DirectedSearchUsage,
##     data = matched_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5090 -2.1333 -0.3656  2.2709  4.4941
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          3.50898    0.08023   43.74   <2e-16 ***
## DirectedSearchUsage -1.37563    0.11346  -12.12   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.404 on 1794 degrees of freedom
## Multiple R-squared:  0.07574,    Adjusted R-squared:  0.07522
## F-statistic:   147 on 1 and 1794 DF,  p-value: < 2.2e-16
```

```
100 * (exp(coef(matched_model3)[2]) - 1)
```

```
## DirectedSearchUsage
##           -74.73199
```

- The coefficient of –1.376 implies that directed search reduces non-promoted sales by approximately 74.7%, a statistically significant effect ($p < 0.001$).
- Even after matching, directed users remain less likely to make spontaneous or unpromoted purchases.

Summary Propensity score matching allows us to isolate the causal effect of directed search, controlling for observed user characteristics. The large and significant differences in sales persist post-matching, strengthening the conclusion that directed search behavior substantially increases promoted and overall purchases, while reducing unplanned purchases.

QUESTION 2

Synthetic Control – Effect of Cigarette Tax in California

The synthetic control method constructs a weighted combination of control states to approximate California's pre-policy sales trend.

To prepare the data for synthetic control, we reshaped cigarette sales into wide format by extracting lagged outcomes for 1975, 1980, and 1988. These serve as key predictors of pre-treatment trends. We also created a numeric state_id to uniquely identify each state and ensure compatibility during model estimation.

```
# Create lagged outcome predictors
lagged_sales <- smoking_df %>%
  filter(year %in% c(1975, 1980, 1988)) %>%
  select(state, year, cigsale) %>%
  pivot_wider(names_from = year, values_from = cigsale, names_prefix = "cigsale_")

# Merge lagged predictors back to full data
smoking_df <- left_join(smoking_df, lagged_sales, by = "state")
smoking_df <- as.data.frame(smoking_df)
```

Step 1: Data Preparation Step 2: Model Setup

```
# Treatment unit
california_id <- unique(smoking_df$state_id[smoking_df$state == "California"])

# Data prep for Synth
synth_prep <- dataprep(
  foo = smoking_df,
  predictors = c("lnincome", "retprice", "age15to24", "beer"),
  predictors.op = "mean",
  special.predictors = list(
    list("cigsale", 1975, "mean"),
    list("cigsale", 1980, "mean"),
    list("cigsale", 1988, "mean")
  ),
  dependent = "cigsale",
  unit.variable = "state_id",
  time.variable = "year",
  treatment.identifier = california_id,
  controls.identifier = setdiff(unique(smoking_df$state_id), california_id),
  time.predictors.prior = 1970:1988,
  time.optimize.ssr = 1970:1988,
  unit.names.variable = "state",
  time.plot = 1970:2000
)
```

```
##
##  Missing data- treated unit; predictor: lnincome ; for period: 1970
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
```

```
##  Missing data- treated unit; predictor: lnincome ; for period: 1971
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1970
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1971
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1972
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1973
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1974
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1975
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1976
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1977
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1978
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1979
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1980
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1981
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1982
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data- treated unit; predictor: beer ; for period: 1983
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data - control unit: 1 ; predictor: lnincome ; for period: 1970
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data - control unit: 1 ; predictor: lnincome ; for period: 1971
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data - control unit: 1 ; predictor: beer ; for period: 1970
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
```

```
##  Missing data - control unit: 1 ; predictor: beer ; for period: 1971
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data - control unit: 1 ; predictor: beer ; for period: 1972
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
##
##  Missing data - control unit: 1 ; predictor: beer ; for period: 1973
##  We ignore (na.rm = TRUE) all missing values for predictors.op.
```

```
# Run Synthetic Control
synth_result <- synth(synth_prep)
```

```
##
## X1, X0, Z1, Z0 all come directly from dataprep object.
##
##
## ****************
##  searching for synthetic control unit
##
##
## ****************
## ****************
## ****************
##
## MSPE (LOSS V): 3.069261
##
## solution.v:
##  0.0010901 0.009765174 0.001041528 0.01119902 0.5811205 0.3235615 0.07222212
##
## solution.w:
##  3.7609e-06 3.0555e-06 0.09383158 0.1106215 4.3302e-06 3.8177e-06 1.3774e-06 1.53224e-05 3.0171e-06 8
```

Step 3: Visualization

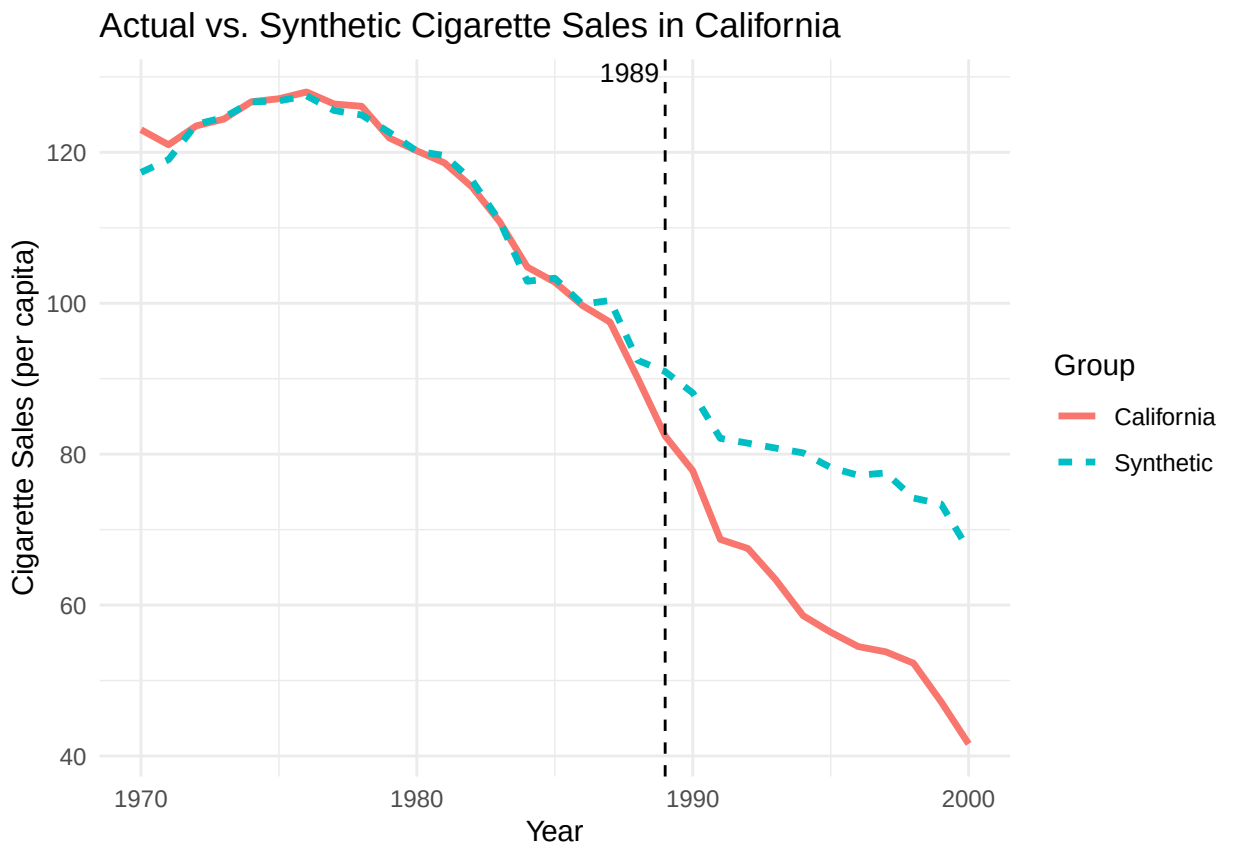
```
# Build plot data
actual_sales <- synth_prep$Y1plot
synthetic_sales <- synth_prep$Y0plot %*% synth_result$solution.w
years <- synth_prep$tag$time.plot

plot_df <- data.frame(
  Year = years,
  California = as.numeric(actual_sales),
  Synthetic = as.numeric(synthetic_sales)
)

plot_long <- pivot_longer(plot_df, cols = c("California", "Synthetic"),
                          names_to = "Group", values_to = "CigaretteSales")

# Plot
ggplot(plot_long, aes(x = Year, y = CigaretteSales, color = Group, linetype = Group)) +
  geom_line(linewidth = 1.2) +
  geom_vline(xintercept = 1989, linetype = "dashed", color = "black") +
  annotate("text", x = 1989, y = max(plot_long$CigaretteSales, na.rm = TRUE),
```

```
          label = "1989", vjust = -0.5, hjust = 1.1, size = 3.5) +
labs(title = "Actual vs. Synthetic Cigarette Sales in California",
     x = "Year", y = "Cigarette Sales (per capita)",
     color = "Group", linetype = "Group") +
theme_minimal()
```



Summary: - We constructed a synthetic control for California using pre-1989 data and key predictors: income, price, youth population, beer consumption, and lagged cigarette sales. - Pre-treatment fit (1970–1988) is strong, indicating a valid synthetic comparison group. - After 1989 (policy intervention), California's cigarette sales sharply diverge downward compared to its synthetic control. - This post-1989 gap represents the causal effect of the tax — suggesting that the policy significantly reduced cigarette consumption in California.

QUESTION 3

Q3: Regression Discontinuity – Impact of Ad Rank on Click-Through Rate (CTR)

We use a sharp **regression discontinuity design (RDD)** to evaluate whether being ranked 1 (vs. 2) in a Google ad auction leads to higher click-through rates (CTR). The running variable is constructed as the **bid difference** between the top two ads in each auction, centered at 0.

```r
# Assign within-auction bid ranks
rd_df <- rd_df %>%
  group_by(auction_id) %>%
  mutate(rank = rank(-bid, ties.method = "first")) %>%
  ungroup()
```

```r
# Filter for top two ads only
rd_filtered <- rd_df %>%
  filter(rank %in% c(1, 2)) %>%
  group_by(auction_id) %>%
  mutate(
    bid_diff = max(bid) - min(bid),
    forcing_var = ifelse(rank == 1, bid_diff, -bid_diff)  # z = forcing variable
  ) %>%
  ungroup()
```

Step 1: Data Preparation and Forcing Variable Step 2: Sharp RDD Estimation using rdrobust

```r
rdd_result <- rdrobust(rd_filtered$ctr, rd_filtered$forcing_var, c = 0)
```

```
## Warning in rdrobust(rd_filtered$ctr, rd_filtered$forcing_var, c = 0): Mass
## points detected in the running variable.
```

```r
summary(rdd_result)
```

```
## Sharp RD estimates using local polynomial regression.
##
## Number of Obs.                20000
## BW type                       mserd
## Kernel                   Triangular
## VCE method                       NN
##
## Number of Obs.                 9837        10163
## Eff. Number of Obs.            7118         7444
## Order est. (p)                    1            1
## Order bias  (q)                   2            2
## BW est. (h)                   0.339        0.339
## BW bias (b)                   0.602        0.602
## rho (h/b)                     0.564        0.564
## Unique Obs.                     457          458
##
## ===============================================================================
##         Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
## ===============================================================================
##   Conventional     0.182     0.031     5.807     0.000     [0.121 , 0.244]
##         Robust         -         -     4.926     0.000     [0.108 , 0.250]
## ===============================================================================
```

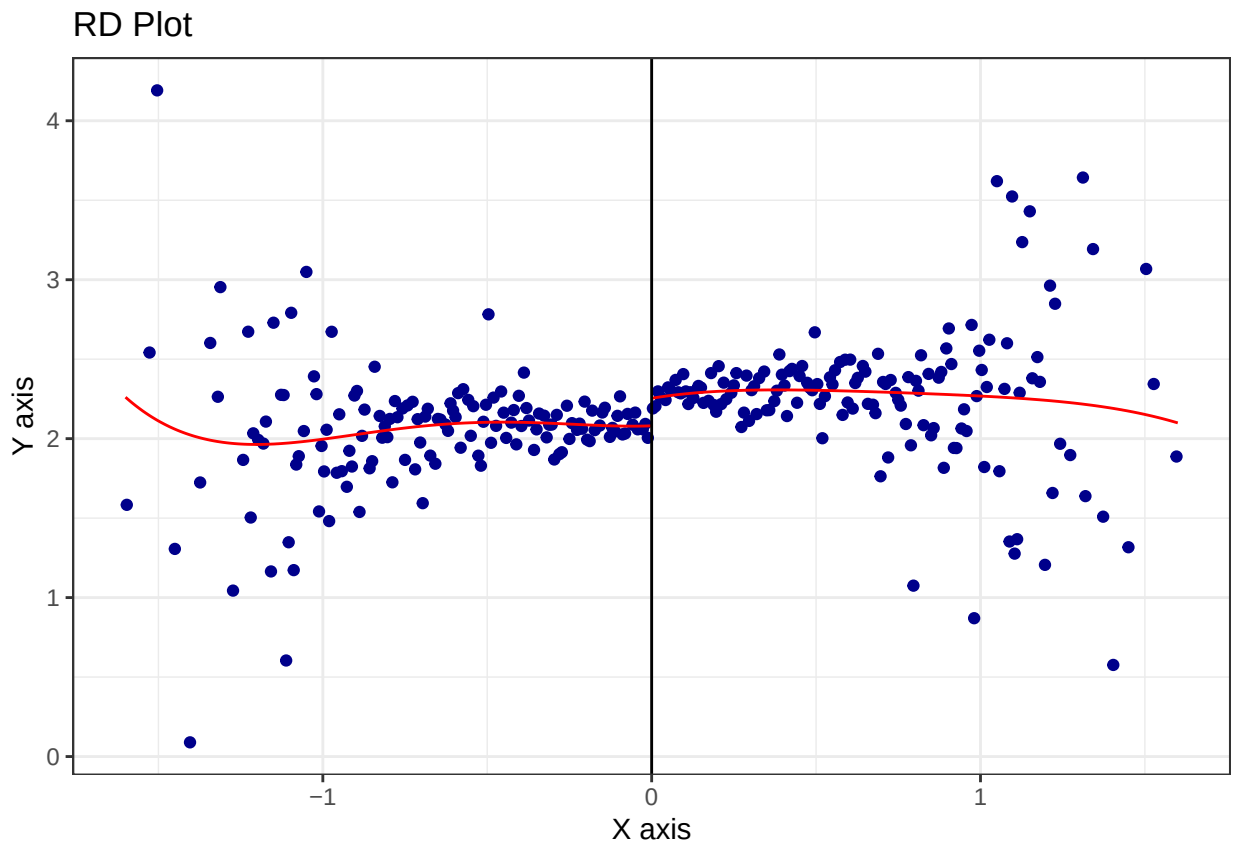
- The local average treatment effect of being ranked 1 (vs. 2) is estimated at 18.2 percentage points.

- This effect is statistically significant ($p < 0.001$), based on both conventional and robust inference.
- The discontinuity at the forcing variable cutoff (0) reflects a causal jump in CTR.

Step 3: RDD Visualization

```
rdplot(rd_filtered$ctr, rd_filtered$forcing_var, c = 0)
```

```
## [1] "Mass points detected in the running variable."
```



- The scatterplot shows a clear jump at the cutoff of $z = 0$.
- Rank 1 ads (positive z) have visibly higher average CTR than rank 2 ads (negative z), supporting the causal claim.

Step 4: Parametric RDD Regression

```
rdd_lm <- lm(ctr ~ forcing_var + ctr + forcing_var * ctr, data = rd_filtered)
```

```
## Warning in model.matrix.default(mt, mf, contrasts): the response appeared on
## the right-hand side and was dropped
```

```
## Warning in model.matrix.default(mt, mf, contrasts): problem with term 2 in
## model.matrix: no columns are assigned
```

```r
summary(rdd_lm)
```

```
##
## Call:
## lm(formula = ctr ~ forcing_var + ctr + forcing_var * ctr, data = rd_filtered)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1065 -0.7598 -0.0029  0.7676  3.3277
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     2.187155   0.007068 309.424  < 2e-16 ***
## forcing_var     0.157649   0.051034   3.089  0.00201 **
## ctr:forcing_var 0.036544   0.021269   1.718  0.08579 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9965 on 19997 degrees of freedom
## Multiple R-squared:  0.006362,   Adjusted R-squared:  0.006263
## F-statistic: 64.02 on 2 and 19997 DF,  p-value: < 2.2e-16
```

- The coefficient on forcing_var is 0.158 (15.8 percentage points), statistically significant ($p = 0.002$).
- The interaction term is not significant, suggesting the slope of CTR does not vary meaningfully across the cutoff.
- This reinforces the view that the jump in CTR at $z = 0$ is due to rank assignment.