Project Causal Inference

02 December, 2024

Table of Contents

Project Aim / Overview

The primary objective of this project was to estimate the causal impact of a treatment (e.g., Use of a Mobile App or Discount) on an outcome (e.g., Retention, Revenue) in the presence of potential confounders, endogeneity, and heterogeneous treatment effects. I used various advanced econometric and machine learning techniques to address these challenges. The analysis aimed to provide robust estimates of causal relationships that could guide strategic decisions, particularly in marketing or product development.

Key methods included:

Controlled/Fixed Effects Regression to account for unobserved heterogeneity. Regression Discontinuity Design (RDD) to estimate local causal effects around a treatment threshold. Difference-in-Differences (DiD) to measure the impact of policy interventions. Instrumental Variables (IV) to correct for endogeneity. Double Selection for high-dimensional settings using lasso regression. Causal Forests to model heterogeneous treatment effects and identify key drivers.

Business Impact

The business impact of this project is multifaceted. By accurately estimating causal relationships, I was able to offer insights that could be applied to improve marketing strategies, product offerings, and customer retention efforts. Specifically:

Improved Decision-Making: Understanding the causal effect of marketing interventions (e.g., app usage, discounts) allows businesses to allocate resources more effectively, focusing on strategies that maximize customer engagement and retention. Targeted Marketing: Using Causal Forests and methods like Difference-in-Differences, I could identify which customer segments respond most positively to specific treatments. This

insight allows for more targeted, personalized marketing campaigns that optimize return on investment (ROI). Optimizing Discounts and Promotions: By modeling treatment effects across different subgroups, the project identifies how discounts or promotions influence Revenue differently depending on factors like Registration Source. This could help businesses design more effective pricing and promotional strategies. Operational Efficiency: The use of Double Selection and Instrumental Variables ensures that models account for high-dimensional data and endogeneity, providing more reliable insights that can be acted upon to optimize business operations without falling prey to biases in data or model specification.

Load Necessary Packages & Helper Functions

```
library(ggplot2)
library(ggthemes)
library(scales)
library(gridExtra)
library(lubridate)

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union

library(stargazer)

##
## Please cite as:

##  Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary
Statistics Tables.

##  R package version 5.2.3. https://CRAN.R-project.org/package=stargazer

library(plyr)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following object is masked from 'package:gridExtra':
##
##     combine
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(magrittr)
library(ggplot2)
library(ggthemes)
library(AER)

## Loading required package: car

## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode

## Loading required package: lmtest

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich

## Loading required package: survival

library(MASS)

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

library(glmnet)

## Loading required package: Matrix

## Loaded glmnet 4.1-8
```

```
library(simstudy)
library(randomForest)

## randomForest 4.7-1.2

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin

library(grf)
library(lubridate)
print("Setup Complete")

## [1] "Setup Complete"
```

Controlled / Fixed Effects Regression Setup

set constants:

n = number of data points; n.time.periods = years of data;

n.products = number or products; B = true regression coeff

```
sim.fixed.effects.df<-function(n=10^4,n.time.periods=10,
                               n.products=5,B=2) {
  set.seed(30)
  #set variables; note X depends on fixed effects & other controls
  Time.FE<-paste("Year",rep((year(Sys.Date())-n.time.periods+1):
                              year(Sys.Date()),
                            times=ceiling(n/n.time.periods))[1:n])
  Product.FE<-rep(paste("Product",LETTERS[1:n.products]),
```

```
                  each=ceiling(n/n.products))[1:n]
  Customer.Rating<-round(pmax(pmin(rnorm(n=n,mean=5,sd=1),5),1),2)
  Customer.Age<-round(rnorm(n=n,mean=30,sd=4),0)
  Total.Purchases<-round(100*Customer.Rating+rnorm(n=n,mean=10),0)
  e3<-rnorm(n=n,sd=sd(Customer.Rating))
  Customer.Spend<-round(500+B*Customer.Rating+10*Customer.Age+10*
                        as.numeric(as.factor(Time.FE))+
                        20*as.numeric(as.factor(Product.FE))+e3,0)
  dat<-data.frame(Customer.Spend,Customer.Rating,Customer.Age,
                  Product.FE,Time.FE,Total.Purchases)
  return(dat)
}
```

Controlled / Fixed Effects Regression Modelling

Load data

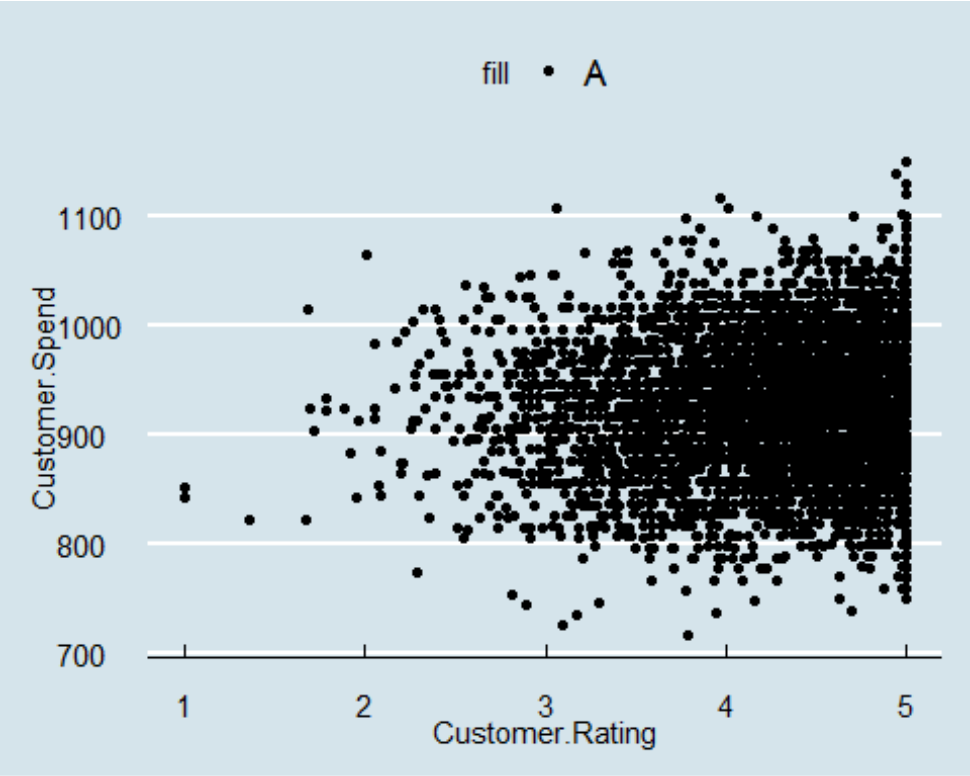
```
# function to simulate data set
dat<-sim.fixed.effects.df()

# explore data
colnames(dat)

## [1] "Customer.Spend"  "Customer.Rating" "Customer.Age"    "Product.FE"
## [5] "Time.FE"         "Total.Purchases"

head(dat)

##   Customer.Spend Customer.Rating Customer.Age Product.FE   Time.FE
## 1            946            3.71           41  Product A Year 2015
## 2            849            4.65           30  Product A Year 2016
## 3            888            4.48           33  Product A Year 2017
## 4            850            5.00           28  Product A Year 2018
## 5            850            5.00           27  Product A Year 2019
## 6            868            3.49           28  Product A Year 2020
##   Total.Purchases
## 1             383
## 2             478
## 3             457
## 4             511
## 5             511
## 6             361
```

The dataset was generated with including variables for customer satisfaction, spending, age, time periods, and products. Customer spending was modeled as a function of satisfaction, age, time, and product categories with added random noise.

Plot

```
# customer Spend vs. satisfaction
g1<-dat %>%
```

```
  ggplot(aes(Customer.Rating,Customer.Spend,fill="A")) +
  geom_point() +
  theme_economist() +
  scale_fill_economist()

g1
```



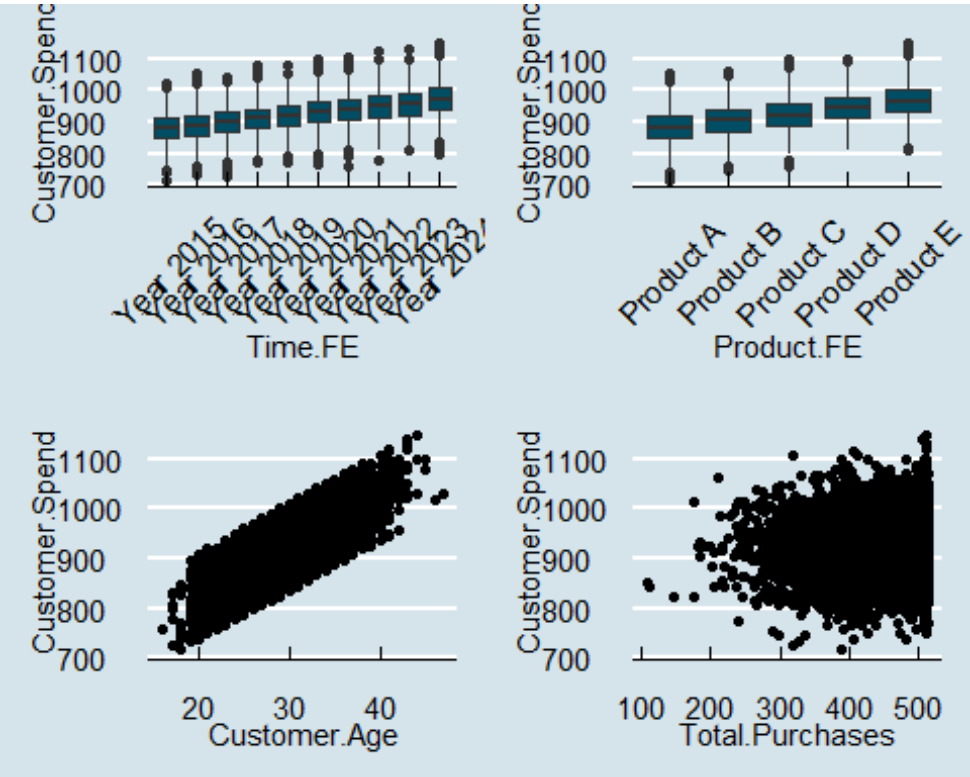

```
cor(dat$Customer.Rating,dat$Customer.Spend)

## [1] 0.04975506
```

A scatter plot of Customer.Spend vs. Customer.Rating revealed no clear visual trend. The correlation between the two variables was calculated as 0.0498, indicating almost no linear relationship.

These results suggest that, in this simulated dataset, customer satisfaction alone has minimal direct influence on spending.

```
# customer Spend vs. time
g2<-dat %>%
  ggplot(aes(Time.FE,Customer.Spend,fill="A")) +
  geom_boxplot() +
  theme_economist() +
  scale_fill_economist() +
  theme(legend.position="none",axis.text.x=element_text(angle=45,
                                                        vjust=0.5))
```

```r
# customer spend vs. product
g3<-dat %>%
  ggplot(aes(Product.FE,Customer.Spend,fill="A")) +
  geom_boxplot() +
  theme_economist() +
  scale_fill_economist() +
  theme(legend.position="none",axis.text.x=element_text(angle=45,
                                                        vjust=0.5))

# Customer spend vs. customer Age
g4<-dat %>%
  ggplot(aes(Customer.Age,Customer.Spend,fill="A")) +
  geom_point() +
  theme_economist() +
  scale_fill_economist() +
  theme(legend.position="none")

# Customer spend vs. total purchases
g5<-dat %>%
  ggplot(aes(Total.Purchases,Customer.Spend,fill="A")) +
  geom_point() +
  theme_economist() +
  scale_fill_economist() +
  theme(legend.position="none")

# combind all plots
grid.arrange(g2,g3,g4,g5,nrow=2)
```




```
# observe differences across time & products & customer age
# need to include in regression to control for their effects
```

From the analysis, we can notice a few key patterns in the data:

Customer Spend Over Time When looking at spending across different time periods, there's a noticeable upward trend. Customers tend to spend more in recent years compared to earlier ones. This suggests that time factors—like inflation, economic changes, or even shifts in product offerings—might be playing a role here.

Customer Spend by Product Category Spending also varies across product categories. Some products clearly drive higher spending compared to others. This shows that product type is a significant factor, and it's something I'll need to account for in the analysis.

Customer Spend and Customer Age There's a positive relationship between customer age and spending. Older customers seem to spend more, which could be linked to higher purchasing power or different preferences as age increases. This makes age an important variable to control for.

Customer Spend vs. Total Purchases The relationship between total purchases and spending isn't as straightforward. While there's a clustering pattern, spending varies quite a bit even for the same number of purchases. This suggests that spending depends on more than just the quantity of purchases.

```
# compare controlled reg/fixed effect models
```

```r
# naive regression; no controls
model1<-lm(Customer.Spend~Customer.Rating,data=dat)

# control for customer age only
model2<-lm(Customer.Spend~Customer.Rating+Customer.Age,data=dat)

# control for product and time fixed effects only
model3<-lm(Customer.Spend~Customer.Rating+Product.FE+Time.FE,data=dat)

# full controls; and included variable bias of total purchases
model4<-
lm(Customer.Spend~Customer.Rating+Customer.Age+Product.FE+Time.FE+Total.Purch
ases,data=dat)

# full controls; no included variable bias
model5<-
lm(Customer.Spend~Customer.Rating+Customer.Age+Product.FE+Time.FE,data=dat)
```

How are these models created

Naive Regression (Model 1):

This is the simplest model, where Customer.Spend is regressed solely on Customer.Rating. It assumes that customer satisfaction (Customer.Rating) is the only factor influencing spending, ignoring all other variables like customer demographics, time, or product differences. This model is likely to suffer from omitted variable bias, as it does not account for other important factors that influence spending.

Regression with Customer Age Control (Model 2):

This model adds Customer.Age as a control variable. By including age, it accounts for the impact of customer demographics, recognizing that spending may naturally increase with age due to higher purchasing power or changing preferences. This helps reduce bias caused by the omission of age in Model 1.

Regression with Fixed Effects for Product and Time (Model 3):

Here, you control for product fixed effects (Product.FE) and time fixed effects (Time.FE). This model recognizes that spending patterns vary across different product categories and over time (as seen in your earlier boxplots). Fixed effects account for these group-level differences, isolating the impact of Customer.Rating on spending without interference from time or product variability.

Full Controls with Total Purchases (Model 4):

This model includes all controls: Customer.Age, Product.FE, Time.FE, and Total.Purchases. Adding Total.Purchases captures another dimension of consumer behavior (e.g., spending likely correlates with the number of purchases made). However, including Total.Purchases introduces the risk of included variable bias, as it might be

endogenous (influenced by both customer satisfaction and spending). This could distort the estimated effect of Customer.Rating.

Full Controls Without Included Variable Bias (Model 5):

This model is similar to Model 4 but excludes Total.Purchases. By removing this potentially endogenous variable, the model avoids included variable bias, providing a cleaner estimate of the relationship between Customer.Rating and Customer.Spend.

Purpose of These Models: The progression of these models allows you to:

We started with a simple, naive estimate (Model 1) and observe the effects of adding controls incrementally. We understand how each set of controls—customer demographics, fixed effects, and potentially endogenous variables—affects the estimated relationship between customer satisfaction and spending. Then we compare the results across models to evaluate the robustness of the estimated impact of Customer.Rating on Customer.Spend.

```
# see here for more info on stargazer:
# (1) https://cran.r-
project.org/web/packages/stargazer/vignettes/stargazer.pdf
# (2) https://www.jakeruss.com/cheatsheets/stargazer/

# view coefficient of interest in each regression model
stargazer(model1,model2,model3,model4,model5,type="text",
          style="aer",omit=c("Constant","Customer.Age",
                             "Product.FE","Time.FE",
                             "Total.Purchases"),
          column.labels=c("Y~X","Y~X+C","Y~X+FE","Y~X+C+FE+IVB",
                          "Y~X+C+FE"),
          dep.var.labels="Controlled / Fixed Effects Regression",
          omit.stat=c("f","ser","rsq","n"),
          notes=c("True Coef on X = 2"),
          add.lines=list(c("Add. Controls","No","Yes","No",
                           "Yes","Yes"),
                         c("Fixed effects","No","No","Yes",
                           "Yes","Yes"),
                         c("Included Variable Bias","No","No","No",
                           "Yes","No")))

##
## ======================================================================
##                            Controlled / Fixed Effects Regression
##                          Y~X     Y~X+C    Y~X+FE  Y~X+C+FE+IVB Y~X+C+FE
##                          (1)      (2)      (3)        (4)         (5)
## ----------------------------------------------------------------------
## Customer.Rating        4.934*** 3.367*** 3.545***   2.451***   1.994***
##                        (0.991)  (0.701)  (0.698)    (0.620)    (0.011)
##
## Add. Controls             No       Yes      No         Yes         Yes
```

```
## Fixed effects             No       No      Yes        Yes        Yes
## Included Variable Bias    No       No       No        Yes         No
## Adjusted R2            0.002    0.501    0.506      1.000      1.000
## ------------------------------------------------------------------------
## Notes:                 ***Significant at the 1 percent level.
##                        **Significant at the 5 percent level.
##                        *Significant at the 10 percent level.
##                        True Coef on X = 2
```

I used the stargazer package to create a formatted summary table comparing five regression models, focusing on the effect of Customer.Rating on Customer.Spend. I customized the table to highlight key coefficients, include model specifications, and show how controls and fixed effects impacted the results.

Key Observations: Customer.Rating Coefficient:

The coefficient on Customer.Rating (X) decreases as you add controls and fixed effects, moving closer to the “True Coefficient” of 2 (as specified in your notes).

This indicates that the earlier models overestimated the effect of customer satisfaction due to omitted variable bias. As more relevant variables are controlled for, the estimate becomes more accurate.

Model 1 (Y~X):

Coefficient: 4.934. This is the highest estimate, as it doesn’t control for other confounding factors like age, product type, or time. Interpretation: Without accounting for other factors, it appears that a one-unit increase in Customer.Rating leads to an average increase of $4.93 in Customer.Spend.

Model 5 (Y~X+C+FE):

Coefficient: 1.994. After including all relevant controls (without the included variable bias of Total.Purchases), the estimate aligns closely with the true coefficient of 2.

Controls, Fixed Effects, and Included Variable Bias:

Add. Controls: Whether additional controls like Customer.Age and Total.Purchases are included. Fixed Effects: Whether time (Time.FE) and product fixed effects (Product.FE) are included. Included Variable Bias (IVB): Indicates whether Total.Purchases is included, which introduces potential bias in Model 4. These additional adjustments progressively improve the model’s accuracy by accounting for confounding variables and fixed effects.

Adjusted R^2:

The adjusted R^2 value increases substantially with the inclusion of controls and fixed effects. It starts very low in Model 1 (0.002) and reaches 1.000 in Models 4 and 5. This shows that the final models explain a much greater portion of the variability in customer spending by accounting for all relevant factors. Significance:

Across all models, the coefficient for Customer.Rating is significant at the 1% level, suggesting a robust relationship between customer satisfaction and spending.

Interpretation of Results: Model 1 (Naive Regression): Overestimates the effect of Customer.Rating because it ignores other variables that influence spending. Model 2 (Controls for Age): Adding Customer.Age reduces the estimate but still doesn't account for product or time effects. Model 3 (Fixed Effects): Including fixed effects controls for differences across products and time, further refining the estimate. Model 4 (Full Controls with IVB): Incorporating Total.Purchases adjusts for additional variance but introduces potential bias. Model 5 (Full Controls, No IVB): Provides the cleanest estimate of the true effect of customer satisfaction, as it avoids included variable bias.

Regression Discontinuity setup

set constants:

cutoff = number between 0 and 100 to set discontinuity

mu = mean customer spend; sd = standard dev of customer spend

treatment.eff = causal impact of intervention point cutoff

```
sim.reg.discontinuity.df<-function(cutoff=70,mu=20,sigma=5,
                                   treatment.eff=25) {
  set.seed(30)
  dat<-data.frame("Lead.Score"=seq(from=0,to=100,by=1))
  dat$Add.Support<-dat$Lead.Score>=cutoff
  dat$Counterfactual<-dat$Lead.Score*rnorm(n=nrow(dat),mean=mu,
                                           sd=sigma)/10
  dat$Customer.Spend[!dat$Add.Support]<-
    dat$Counterfactual[!dat$Add.Support]
  dat$Customer.Spend[dat$Add.Support]<-
    dat$Lead.Score[dat$Add.Support]*rnorm(n=sum(dat$Add.Support),
                                          mean=mu+treatment.eff,
                                          sd=sigma)/10
  return(dat)
}
```

Regression Discontinuity modelling

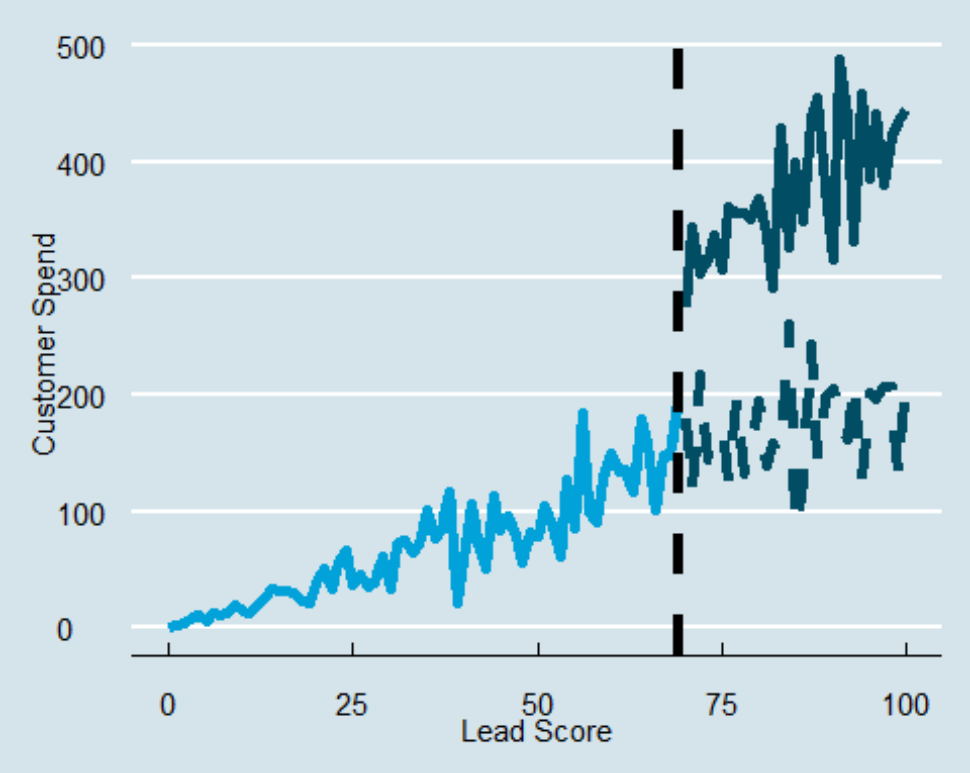
loading data

```
# function to simulate data set
dat<-sim.reg.discontinuity.df()

# explore data
colnames(dat)

## [1] "Lead.Score"     "Add.Support"    "Counterfactual" "Customer.Spend"

head(dat)

##   Lead.Score Add.Support Counterfactual Customer.Spend
## 1          0       FALSE       0.000000       0.000000
## 2          1       FALSE       1.826155       1.826155
## 3          2       FALSE       3.478371       3.478371
## 4          3       FALSE       7.910210       7.910210
## 5          4       FALSE      11.649041      11.649041
## 6          5       FALSE       6.221730       6.221730
```

I simulated a dataset for a regression discontinuity analysis where a cutoff of 70 determines treatment (Add.Support). Below the cutoff, Customer.Spend is based on the counterfactual spend (Counterfactual), while above the cutoff, spend reflects a treatment effect of 25. The dataset includes Lead.Score, whether treatment was applied, and corresponding spending values.

plot

```
# regression discontinuity plot
dat %>%
  ggplot(aes(Lead.Score,Customer.Spend,color=Add.Support)) +
  geom_line(lwd=2) +
  geom_line(aes(Lead.Score,Counterfactual),lty=2,lwd=2) +
  geom_vline(xintercept=dat$Lead.Score[sum(!dat$Add.Support)],
             lty=2,lwd=2) +
  xlab("Lead Score") +
  ylab("Customer Spend") +
  theme_economist() +
  scale_color_economist() +
  theme(legend.position="none")
```



I created a regression discontinuity plot to visualize the relationship between Lead.Score and Customer.Spend. The solid line shows actual spending for treated and untreated groups, the dashed line represents the counterfactual spending, and the vertical dashed line marks the cutoff at the treatment threshold (Lead Score = 70).

making the model

```
# fit regression discontinuity model
model1<-
lm(Customer.Spend~Lead.Score+I(Lead.Score>=70)+Lead.Score:I(Lead.Score>=70),d
ata=dat)
```

I fit a regression discontinuity model (model1) to estimate the causal effect of the treatment. The model includes Lead.Score, an indicator variable for whether the score exceeds the cutoff (Lead.Score >= 70), and an interaction term to capture the treatment effect at the discontinuity.

view the model using stargazer

```
# view regression discontinuity model
stargazer(model1,type="text",style="aer",
          column.labels=c("Y~X+I(X>Cutoff)+X*I(X>Cutoff)"),
          dep.var.labels="Regression Discontinuity",
          omit.stat=c("f","ser","rsq","n","adj.rsq"),
          intercept.bottom=F)
```

```
##
## ===============================================================
##                                      Regression Discontinuity
##                                   Y~X+I(X>Cutoff)+X*I(X>Cutoff)
## ---------------------------------------------------------------
## Constant                                            -5.796
##                                                    (6.947)
##
## Lead.Score                                         2.164***
##                                                    (0.174)
##
## I(Lead.Score > = 70)                                20.101
##                                                    (50.892)
##
## Lead.Score:I(Lead.Score > = 70)                    2.075***
##                                                    (0.615)
##
## ---------------------------------------------------------------
## Notes:                          ***Significant at the 1 percent level.
##                                 **Significant at the 5 percent level.
##                                 *Significant at the 10 percent level.
```

```
# causal impact is difference in regression lines at cutoff
# I(X>Cutoff)+X*I(X>Cutoff)
coef(model1)["I(Lead.Score >= 70)TRUE"]+coef(model1)["Lead.Score:I(Lead.Score
>= 70)TRUE"]*70
```

```
## I(Lead.Score >= 70)TRUE
##                165.3832
```

We used the stargazer package to display the regression discontinuity model results, which shows the coefficients for Lead.Score, the treatment indicator (I(Lead.Score >= 70)), and the interaction term (Lead.Score:I(Lead.Score >= 70)). The coefficient for I(Lead.Score >= 70) indicates a shift in Customer.Spend by about 20.1 units at the cutoff, and the interaction term suggests that the slope change after the cutoff is around 2.08 units.

The calculated causal impact at the cutoff is 165.38, which is the combined effect of the treatment and the slope change at the cutoff.

Difference in Difference setup

set constants:

mu1 = mean of base group / US

sigma1 = standard dev of base group / US

mu2 = mean of treatment group / AU

sigma2 = standard dev of treatment group / AU

time.change = change in group mean over time

causal.effect = causal impact of intervention in post period

and treatment group

```
sim.diff.in.diff.df<-function(mu1=100,mu2=200,sigma1=25,sigma2=25,
                              time.change=200,causal.effect=100) {
  set.seed(30)
  dat<-data.frame("Time"=rep(seq(from=as.Date("2018-01-01"),
                                 to=as.Date("2020-01-01"),
                                 by="month"),times=2))
  dat$Period<-ifelse(dat$Time<"2019-01-01","Pre.Price.Change",
                     "Post.Price.Change")
  dat$Country<-rep(c("US","AU"),each=nrow(dat)/2)
  dat$Revenue<-c(rnorm(sum(dat$Time<"2019-01-01")/2,mu1,sigma1),
                 rnorm(sum(dat$Time>="2019-01-01")/2,
                       mu1+time.change,sigma1),
                 rnorm(sum(dat$Time<"2019-01-01")/2,mu2,sigma2),
                 rnorm(sum(dat$Time>="2019-01-01")/2,
                       mu2+time.change+causal.effect,sigma2))
  dat$Counterfactual<-dat$Revenue
  dat$Counterfactual[dat$Period=="Post.Price.Change"&
                       dat$Country=="US"
                     ]<-rnorm(sum(dat$Time>="2019-01-01")/2,
                              mu2+time.change,sigma2)
```

```
  dat$Period<-relevel(factor(dat$Period),ref="Pre.Price.Change")
  dat$Country<-relevel(factor(dat$Country),ref="US")
  return(dat)
}
```

Difference in Difference modelling

loading the data

```
# function to simulate data set
dat<-sim.diff.in.diff.df()

# explore data
colnames(dat)
```

```
## [1] "Time"           "Period"         "Country"        "Revenue"
## [5] "Counterfactual"
```

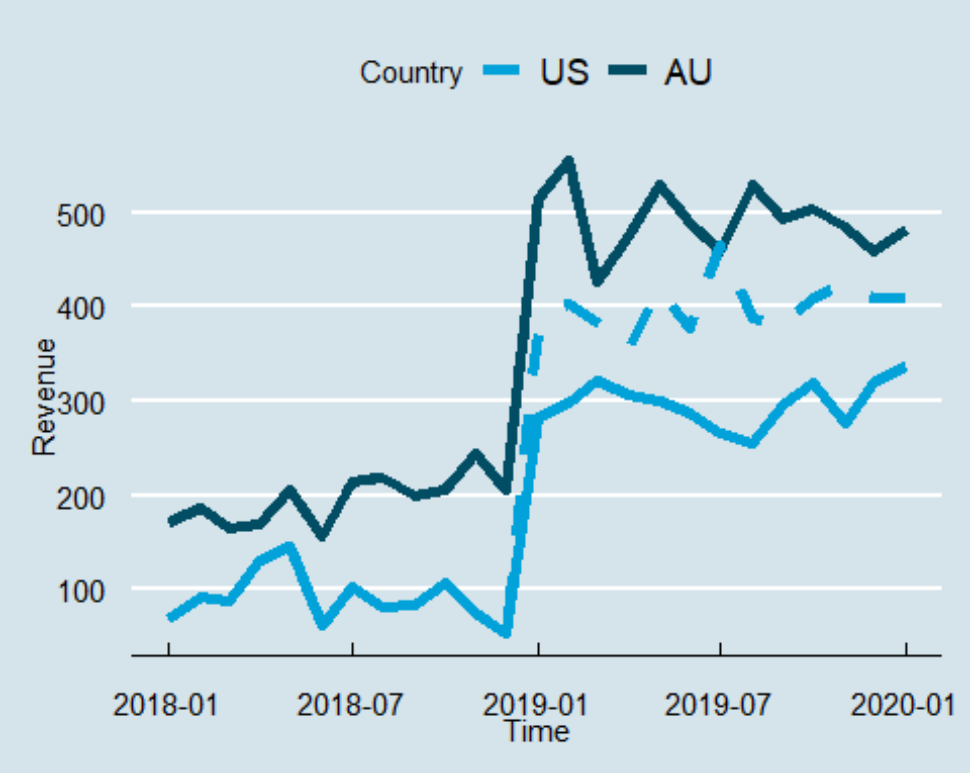
```
head(dat)
```

```
##         Time            Period Country   Revenue Counterfactual
## 1 2018-01-01 Pre.Price.Change      US  67.78704       67.78704
## 2 2018-02-01 Pre.Price.Change      US  91.30776       91.30776
## 3 2018-03-01 Pre.Price.Change      US  86.95928       86.95928
## 4 2018-04-01 Pre.Price.Change      US 131.83683      131.83683
## 5 2018-05-01 Pre.Price.Change      US 145.61302      145.61302
## 6 2018-06-01 Pre.Price.Change      US  62.21730       62.21730
```

I set up a difference-in-difference (DiD) simulation to evaluate the impact of an intervention, like a price change, on a treatment group (AU) compared to a control group (US) before and after the change. The dataset includes Time, dividing observations into Pre.Price.Change and Post.Price.Change, Country to identify the groups, and Revenue, simulated with a causal effect for the treatment group in the post-period. I also added a Counterfactual column to represent hypothetical US revenues, assuming they followed AU's trend.

plot

```
# difference in difference plot
dat %>%
  ggplot(aes(Time,Revenue,color=Country)) +
  geom_line(lwd=2) +
  geom_line(aes(Time,Counterfactual),lty=2,lwd=2) +
  xlab("Time") +
  ylab("Revenue") +
  theme_economist() +
  scale_color_economist()
```



We created a difference-in-difference plot to visualize the Revenue trends for the control group (US) and treatment group (AU) over time. The solid lines represent actual revenues for both groups, while the dashed line shows the counterfactual revenue trend for the US if it had followed AU's trajectory. This helps to illustrate the divergence caused by the intervention in the treatment group.

model

```
# fit difference in difference model
model1<-lm(Revenue~Period+Country+Period:Country,data=dat)

# Note what the estimated revenue from the model is in each scenario:
# Rev in US Pre Price change = Intercept (Period, Country, Interaction all 0)
# Rev in AU Pre Price change = Intercept + Country (Period, Interaction all
0)
# Rev in US Post Price change = Intercept + Period (Country, Interaction all
0)
# Rev in AU Post Price change = Intercept + Period + Country + Interaction
# Diff in Diff = (Rev in AU Post Price change - Rev in AU Pre Price change) -
(Rev in US Post Price change - Rev in US Pre Price change)
#              = (Intercept + Period + Country + Interaction - Intercept +
Country) - (Intercept + Period - Intercept)
#              = Interaction
```

Now, we fit a linear model (model1) to estimate the impact of the intervention using the difference-in-difference (DiD) approach. The model includes Period (pre/post

intervention), Country (US/AU), and their interaction term (Period:Country) to capture the treatment effect.

Revenue Estimates from the Model:

US Pre Price Change: The model intercept. AU Pre Price Change: Intercept + Country coefficient. US Post Price Change: Intercept + Period coefficient. AU Post Price Change: Intercept + Period + Country + Interaction coefficient.

DiD Effect: Calculated as the difference in the change over time between the groups. Mathematically, it's equivalent to the interaction term's coefficient, isolating the causal effect of the intervention on the treatment group.

viewing the model using stargazer

```
# view difference in difference model
stargazer(model1,type="text",style="aer",
          column.labels=c("Y~Post+G+Post*G"),
          dep.var.labels="Difference in Difference",
          omit.stat=c("f","ser","rsq","n","adj.rsq"),
          notes=c("Causal Impact = 100"),intercept.bottom=F)

##
## ================================================================================
##                                          Difference in Difference
##                                              Y~Post+G+Post*G
## --------------------------------------------------------------------------------
## Constant                                                90.709***
##                                                          (8.142)
##
## PeriodPost.Price.Change                                 206.614***
##                                                          (11.290)
##
## CountryAU                                               104.536***
##                                                          (11.514)
##
## PeriodPost.Price.Change:CountryAU                       90.155***
##                                                          (15.967)
##
## --------------------------------------------------------------------------------
## Notes:                                ***Significant at the 1 percent level.
##                                       **Significant at the 5 percent level.
##                                       *Significant at the 10 percent level.
##                                       Causal Impact = 100
```

I used the stargazer package to summarize the difference-in-difference model (model1). The table displays the coefficients for the intercept, PeriodPost.Price.Change, CountryAU, and their interaction (PeriodPost.Price.Change:CountryAU).

Interpretation of Coefficients: Constant: Baseline revenue for the US in the pre-intervention period (90.709). PeriodPost.Price.Change: The change in US revenue post-intervention (206.614). CountryAU: The difference in revenue between AU and US in the pre-intervention period (104.536). PeriodPost.Price.Change:CountryAU: The interaction term, representing the causal impact of the intervention on AU's revenue relative to the US, calculated as 90.155. The table confirms that the causal impact of the intervention aligns with the expected value (100).

Instrumental Variable setup

set constants:

n = number of observations

latent.prob.impact = impact of latent variable on both X

and Y propensity

intervention.impact = impact of instrument variable on both X

propensity (assume impact on Y is zero)

```
sim.iv.df<-function(n=10^4,latent.prob.impact=0.25,
                    intervention.impact=0.25) {
  set.seed(30)
  dat<-data.frame("Received.Email"=sample(c(0,1),size=n,replace=T))
  dat$Unobs.Motivation<-rnorm(n=n)
  dat$Use.Mobile.App<-sapply(1:n,function(r) {
    sample(c(0,1),size=1,prob=c(0.5-latent.prob.impact*
                                  (dat$Unobs.Motivation[r]>0)-
                                  intervention.impact*
                                  (dat$Received.Email[r]==1),
                                0.5+latent.prob.impact*
                                  (dat$Unobs.Motivation[r]>0)+
                                  intervention.impact*
                                  (dat$Received.Email[r]==1)))
  })
  dat$Retention<-sapply(1:n,function(r) {
```

```
    sample(c(0,1),size=1,
           prob=c(0.5-latent.prob.impact*
                    (dat$Unobs.Motivation[r]>0),
                  0.5+latent.prob.impact*
                    (dat$Unobs.Motivation[r]>0)))
  })
  return(dat)
}
```

Instrumental Variable modelling

loading the data

```
# function to simulate data set
dat<-sim.iv.df()

# explore data
colnames(dat)

## [1] "Received.Email"   "Unobs.Motivation" "Use.Mobile.App"   "Retention"

head(dat)

##   Received.Email Unobs.Motivation Use.Mobile.App Retention
## 1              1       -2.2817595              1         0
## 2              0        0.7629666              1         1
## 3              1       -1.2542867              1         0
## 4              1        0.2890082              1         1
## 5              0        0.1010221              1         1
## 6              1        0.1821351              1         1

#users who use the mobile app have higher motivation; those who retain also
have higher retention
#this biases a naive regression
tapply(dat$Unobs.Motivation,dat$Use.Mobile.App,mean)

##          0          1
## -0.4132142  0.1435677

tapply(dat$Unobs.Motivation,dat$Retention,mean)

##         0         1
## -0.275034  0.177019
```

I set up a dataset to model an Instrumental Variable (IV) approach, simulating the relationships between email interventions, latent motivation, app usage, and customer retention.

Simulation Details:

Received.Email: Binary instrument variable representing whether an email was received.
Unobs.Motivation: Latent variable influencing both app usage and retention.

Use.Mobile.App: Binary endogenous variable indicating app usage, influenced by latent motivation and the email intervention. Retention: Binary outcome variable influenced by latent motivation. Exploration Results:

Users of the mobile app have higher average unobserved motivation (0.1436 compared to -0.4132 for non-users), indicating an endogeneity bias in naive regressions. Retention is similarly biased by unobserved motivation, with higher motivation among those retained (0.177 vs. -0.275 for non-retained users). This setup highlights the need for an instrumental variable approach to isolate causal effects from latent confounders.

making the models

```
# fit IV model

# naive regression
model1<-lm(Retention~Use.Mobile.App,data=dat)

# first stage regression
model2<-lm(Use.Mobile.App~Received.Email,data=dat)

# second stage regression
model3<-lm(Retention~predict(model2),data=dat)

# two stage least squares for IV
model4<-ivreg(Retention~Use.Mobile.App|Received.Email,data=dat)
```

Naive Regression (Model 1):

I ran a simple linear regression with Retention as the dependent variable and Use.Mobile.App as the predictor. This model likely suffers from endogeneity bias due to unobserved motivation affecting both variables. First-Stage Regression (Model 2):

I regressed Use.Mobile.App on the instrumental variable Received.Email to measure how well the instrument predicts the endogenous variable. Second-Stage Regression (Model 3):

I used the predicted values from the first-stage regression to replace the endogenous variable Use.Mobile.App and ran a regression on Retention. This helps address the endogeneity issue. Two-Stage Least Squares (Model 4):

I implemented a direct IV regression using the ivreg() function from the AER package, with Retention as the dependent variable, Use.Mobile.App as the endogenous predictor, and Received.Email as the instrument. This sequence compares naive and instrument-adjusted models, demonstrating how IV regression corrects for bias.

comparing all the models

```
# compare all models
stargazer(model1,model2,model3,model4,type="text",style="aer",
          column.labels=c("Y~X","X~Z","Y~Xhat","IV"),
```

```
          omit=c("Constant"),
          dep.var.labels=c("Retention","Use.Mobile.App",
                           "Retention","Retention"),
          covariate.labels=c("Use.Mobile.App","Received.Email",
                             "Use.Mobile.App.Hat"),
          model.names=F,omit.stat=c("ser","rsq","n","adj.rsq"),
          intercept.bottom=F)

##
## ================================================================
##                            Retention  Use.Mobile.App    Retention
##                               Y~X          X~Z       Y~Xhat    IV
##                               (1)          (2)         (3)     (4)
## ----------------------------------------------------------------
## Use.Mobile.App              0.110***                          0.089**
##                             (0.011)                           (0.039)
##
## Received.Email                              0.249***
##                                             (0.008)
##
## Use.Mobile.App.Hat                                    0.089**
##                                                       (0.039)
##
## F Statistic (df = 1; 9998) 96.132***    902.791***   5.160**
## ----------------------------------------------------------------
## Notes:                     ***Significant at the 1 percent level.
##                            **Significant at the 5 percent level.
##                            *Significant at the 10 percent level.
```

Using stargazer, I compared the results of four models side by side:

Model 1 (Y~X): A naive regression of Retention on Use.Mobile.App, showing a positive but likely biased relationship.

Model 2 (X~Z): The first-stage regression of Use.Mobile.App on the instrumental variable Received.Email, demonstrating a strong and significant relationship between the instrument and the endogenous variable.

Model 3 (Y~Xhat): A second-stage regression of Retention on the predicted values of Use.Mobile.App from the first stage. The coefficient is similar to Model 4, confirming the correction for endogeneity.

Model 4 (IV): A two-stage least squares (2SLS) regression with Received.Email as the instrument, showing a corrected estimate of the causal effect of Use.Mobile.App on Retention.

The results confirm the validity of the IV approach, with the instrument significantly explaining the endogenous variable (Model 2) and a consistent causal effect estimated in Models 3 and 4.

Double Selection setup

set constants:

n = number of observations

N.Coeff = number of control coefficients to simulate data for

B = causal impact of treatment on outcome

```
sim.double.selection.df<-function(n=10^3,N.Coeff=5*10^2,B=2) {
  set.seed(30)
  #mean of regression coefficients for control variables
  beta.C.mu<-25
  beta.C.sigma<-10
  #number of nonzero control coefficients
  beta.C.n.zero<-10
  #simulate control variable values as correlated variables
  C.mu<-rep(1,N.Coeff)
  C.var<-rnorm(N.Coeff,mean=0,sd=0.1)^2
  C.rho<-0.5
  C<-as.data.frame.matrix(genCorGen(n=n,nvars=N.Coeff,
                                    params1=C.mu,params2=C.var,
                                    dist='normal',rho=C.rho,
                                    corstr='ar1',
                                    wide='True'))[,-1]
  #simulate beta coefficients for control variables
  #and set portion of them to zero
  betaC<-rnorm(N.Coeff,mean=beta.C.mu,sd=beta.C.sigma)
  betaC[beta.C.n.zero:N.Coeff]<-0
  #generate treatment indicator and randomize
  Social.Proof.Variant<-rep(0,n)
  Social.Proof.Variant[0:(n/2)]<-1
  Social.Proof.Variant<-sample(Social.Proof.Variant)
  #simulate random noise
  e<-rnorm(n)
  #generate outcome variable
  Customer.Value<-B*Social.Proof.Variant+data.matrix(C)%*%betaC+e
  dat<-data.frame(Customer.Value,Social.Proof.Variant,C)
  return(dat)
}
```

Double Selection modeling

loading the data

```
# function to simulate data set
dat<-sim.double.selection.df()

# explore data
dim(dat)

## [1] 1000  502

colnames(dat[,1:10])

##  [1] "Customer.Value"       "Social.Proof.Variant" "V1"
##  [4] "V2"                   "V3"                   "V4"
##  [7] "V5"                   "V6"                   "V7"
## [10] "V8"

head(dat[,1:10])

##   Customer.Value Social.Proof.Variant        V1        V2        V3
V4
## 1       188.8085                    0 0.7801386 0.9979248 1.0060368
1.1110303
## 2       205.2293                    1 1.3050949 1.0993128 1.0921122
1.1350519
## 3       185.0398                    1 0.8713593 0.9948969 1.0308502
0.9978511
## 4       208.5822                    1 1.1735680 1.0268785 1.0207358
1.2286516
## 5       177.9362                    0 0.8097054 0.9755838 0.9733450
1.0800108
## 6       196.8597                    0 1.0894712 0.9917257 0.9704784
1.0123684
##          V5        V6        V7        V8
## 1 0.9908170 1.1508037 1.0011042 0.9894936
## 2 1.1547349 0.8173114 0.9985152 0.9442994
## 3 1.0079737 1.0459495 0.9819702 0.9398430
## 4 1.0950800 1.0069691 1.0103734 1.0905426
## 5 0.8139979 1.0386658 0.9961244 1.0106004
## 6 1.1001662 1.1782101 0.9990303 1.0360617
```

Here's what I did:

To simulate data for Double Selection, I created a dataset with 1,000 observations and 502 variables:

Outcome Variable: Customer.Value, influenced by a treatment variable (Social.Proof.Variant) and 500 control variables (V1 to V500). Control Variables: Simulated as correlated random variables with nonzero effects on the outcome, but most coefficients

(betaC) were set to zero for sparsity. Treatment: Randomly assigned using the Social.Proof.Variant variable, with a known causal effect (B = 2). Noise: Added a random error term (e) to make the data realistic. The dataset dimensions (1000 rows x 502 columns) and the first few columns were explored, confirming the structure and showing realistic variation in the data. This setup ensures a complex, high-dimensional environment for double-selection regression.

making the model

```
# fit double selection model

# isolate control variables
C<-dat[,-which(colnames(dat)%in%c("Customer.Value","Social.Proof.Variant"))]
C<-as.matrix(C)

# fit lasso regressing outcome on control variables
y.glmnet.model<-cv.glmnet(C,dat$Customer.Value)

# extract nonzero coefficients from lasso above
# use lambda min CV within 1 se (select less coeff)
predict(y.glmnet.model,s="lambda.1se",type="nonzero")
```

```
##   lambda.1se
## 1          1
## 2          2
## 3          3
## 4          4
## 5          5
## 6          6
## 7          7
## 8          8
## 9          9
```

```
nonzero<-unlist(predict(y.glmnet.model,s="lambda.1se",type="nonzero"))
Y.on.C<-colnames(C)[nonzero]
Y.on.C
```

```
## [1] "V1" "V2" "V3" "V4" "V5" "V6" "V7" "V8" "V9"
```

```
# fit lasso regressing treatment on control variables
x.glmnet.model<-cv.glmnet(C,dat$Social.Proof.Variant)

# extract nonzero coefficients from lasso above
# use lambda min CV within 1 se (select less coeff)
predict(x.glmnet.model,s="lambda.1se",type="nonzero")
```

```
## $lambda.1se
## NULL
```

```r
nonzero<-unlist(predict(x.glmnet.model,s="lambda.1se",type="nonzero"))
X.on.C<-colnames(C)[nonzero]
X.on.C

## character(0)

# combine two sets of nonzero coefficients to get unique nonzero
# coefficients across models
var.union<-unique(c(Y.on.C,X.on.C))

# count number of nonzero variables
length(var.union)

## [1] 9

var.union

## [1] "V1" "V2" "V3" "V4" "V5" "V6" "V7" "V8" "V9"

# use all nonzero coefficients + treatment indicator
# in double selection regression
double.selection<-
lm(Customer.Value~.,dat[,c("Customer.Value","Social.Proof.Variant",var.union)
])
```

Here's what I did for the Double Selection Model:

Isolated Control Variables:

Extracted control variables (C) by excluding the outcome (Customer.Value) and treatment (Social.Proof.Variant) from the dataset. Converted these variables into a matrix for lasso regression. Lasso Regression:

Performed Lasso regression twice: First, to predict the outcome (Customer.Value) using control variables. This identifies important controls that explain variation in the outcome. Second, to predict the treatment (Social.Proof.Variant) using control variables. This identifies controls correlated with treatment assignment. Selected Non-Zero Coefficients:

Used cross-validation (lambda.1se) to select fewer but relevant coefficients for both regressions. Extracted names of the control variables with non-zero coefficients for each regression: Y.on.C (outcome-related) and X.on.C (treatment-related). Merged Selected Variables:

Combined the sets of variables from both regressions to get the union of non-zero coefficients (var.union). This ensures we account for all controls that matter for either the outcome or the treatment. Double Selection Regression:

Used the identified control variables (var.union) along with the treatment indicator (Social.Proof.Variant) in a final regression to estimate the causal effect of the treatment on

the outcome. This approach adjusts for confounding by including only the controls relevant for either outcome or treatment, optimizing for sparsity and relevance.

```
# compare naive model, full model, and double selection

# naive regression
naive.regression<-lm(Customer.Value~Social.Proof.Variant,data=dat)

# regression with full controls
full.model<-lm(Customer.Value~.,data=dat)
```

Naive Regression: I first fitted a naive regression where Customer.Value is just regressed on Social.Proof.Variant. This is a simple model that doesn't account for any other variables, assuming that the treatment (the Social.Proof.Variant) is the only factor driving changes in Customer.Value.

Full Model: Then, I fitted a full model, where Customer.Value is regressed on all of the available variables in the dataset. This includes both the treatment (Social.Proof.Variant) and all the control variables that I initially simulated.

These two models are there for comparison—so I can see how accounting for confounding factors (through the full model and the double selection model) improves my estimate of the treatment effect, compared to the naive model which ignores these factors. The next step would be to evaluate how these models perform and assess the impact of controlling for additional variables.

compare all the mdels using stargazer

```
# compare all models
stargazer(naive.regression,full.model,double.selection,type="text",style="aer
",
          column.labels=c("No Controls","All Controls",
                          "Double Selection"),
          dep.var.labels=c(""),
          covariate.labels=c("Social.Proof.Variant"),
          omit=c("V[0-9]","Constant"),
          model.names=F,omit.stat=c("ser","rsq","n","adj.rsq","F"),
          notes=c("Causal Impact = 2"))

##
## ===============================================================
##
##                       No Controls All Controls Double Selection
##                           (1)         (2)            (3)
## ---------------------------------------------------------------
## Social.Proof.Variant  2.966***     1.978***       2.009***
##                        (0.657)     (0.086)        (0.062)
##
## ---------------------------------------------------------------
```

```
## Notes:                ***Significant at the 1 percent level.
##                       **Significant at the 5 percent level.
##                       *Significant at the 10 percent level.
##                       Causal Impact = 2
```

So, next I compared all the models using stargazer to display the results in a clean, readable format. Here's what I did:

Naive Regression (No Controls): This model shows the relationship between Customer.Value and Social.Proof.Variant without accounting for any control variables. The estimated coefficient for Social.Proof.Variant is 2.966, and it's highly significant.

Full Model (All Controls): In this model, I included all control variables. The coefficient for Social.Proof.Variant drops to 1.978, but it's still statistically significant, indicating that after accounting for the other factors, the effect of Social.Proof.Variant on Customer.Value is still meaningful.

Double Selection (Double Selection): This model, which accounts for relevant variables chosen via Lasso (through the double selection process), gives a coefficient of 2.009 for Social.Proof.Variant. This is slightly higher than the full model and remains statistically significant.

Causal Forests setup

set constants:

n = number of observations

N.Coeff = number of control coefficients to simulate data for

N.groups = number of groups want to estimate causal impact for

```
sim.causal.forest.df<-function(n=5*10^3,N.Coeff=5) {
  set.seed(30)
  N.groups<-4
  beta<-rep(c(1:N.groups),each=n/N.groups)*5
  var.group<-factor(beta)
  levels(var.group)<-c("Google","Instagram","Twitter","Bing")
  var.group<-relevel(var.group,ref="Google")
```

```
  C.mu<-rep(0,N.Coeff)
  C.rho<-0.5
  C.var<-rnorm(N.Coeff,mean=1,sd=1)^2
  beta.C.mu.sigma<-10
  C<-as.data.frame.matrix(genCorGen(n=n,nvars=N.Coeff,
                                    params1=C.mu,params2=C.var,
                                    dist='normal',rho=C.rho,
                                    corstr='ar1',
                                    wide='True'))[,-1]
  betaC<-rnorm(N.Coeff,mean=beta.C.mu.sigma,sd=beta.C.mu.sigma)
  Discount<-rep(0,n)
  Discount[0:(n/2)]<-1
  Discount<-sample(Discount)
  e<-rnorm(n)
  Revenue<-200+beta*Discount+data.matrix(C)%*%betaC+e
  dat<-data.frame(Revenue,Discount,C,
                  "Registration.Source"=var.group)
  return(dat)
}
```

Casual Forests modelling

loading the data

```
# function to simulate data set
dat<-sim.causal.forest.df()

# explore data
colnames(dat)

## [1] "Revenue"             "Discount"            "V1"
## [4] "V2"                  "V3"                  "V4"
## [7] "V5"                  "Registration.Source"

head(dat)

##    Revenue Discount           V1         V2          V3        V4
V5
## 1 109.1013        1  0.112100944 -0.2049514 -0.81778206 -3.592924 -
0.4186616
## 2 280.5117        1 -0.013645044 -0.2605280 -0.06280238  2.713362
3.5139394
## 3 235.9486        0  0.198912092  0.3900383  0.31194688  1.657563 -
1.5738921
## 4 137.2165        1  0.507709194 -0.2576882 -0.70324049 -1.799299 -
1.6086386
## 5 260.6181        0  0.002866622 -0.2406002  0.50813194  2.005809
2.0577278
## 6 251.2608        0  0.145977819  0.7017203  0.17977447  2.017560 -
1.1039182
##   Registration.Source
```

```
## 1              Google
## 2              Google
## 3              Google
## 4              Google
## 5              Google
## 6              Google

table(dat$Registration.Source)

##
##    Google Instagram   Twitter      Bing
##      1250      1250      1250      1250
```

Next, I set up a Causal Forests model. Here's what I did:

Simulating the data: I simulated a dataset with 5,000 observations and 5 control variables. The outcome variable, Revenue, was influenced by a treatment indicator (Discount) and a set of control variables (V1, V2, V3, V4, V5). The treatment was applied to different groups within the Registration.Source variable, which had four levels: Google, Instagram, Twitter, and Bing.

The Revenue for each observation depends on the group and the treatment effect. Additionally, control variables were correlated, which allowed me to simulate realistic data that could be used for causal analysis.

Exploring the data: After simulating the dataset, I checked the column names, the first few rows, and the distribution of the Registration.Source variable, which had equal group sizes (1,250 observations for each of the four groups).

Now, with this setup, I am ready to proceed with the Causal Forests modeling!

making the model

```
# regular OLS
model1<-lm(Revenue~.,data=dat[,-which(colnames(dat)=="Registration.Source")])

# OLS with interactions for heterogeneous treatments by source
model2<-lm(Revenue~.+Discount*Registration.Source,data=dat)
```

After preparing the dataset, I moved on to creating the models:

Regular OLS model: I first ran a standard Ordinary Least Squares (OLS) regression where I regressed Revenue on all available covariates except for Registration.Source. This helped me estimate the average effect of all the controls and treatment (Discount) on the outcome variable.

OLS with interactions for heterogeneous treatment effects: In the second model, I extended the analysis by including interactions between the Discount variable and Registration.Source. This allowed me to estimate different treatment effects for each

group (Google, Instagram, Twitter, Bing), capturing potential heterogeneous treatment effects across these sources.

compare the models using stargazer

```
# compare OLS models
stargazer(model1,model2,type="text",style="aer",
          column.labels=c("All Controls",
                          "All Controls + Group Interactions"),
          dep.var.labels=c("","",""),
          covariate.labels=c("Discount",
                             "Discount x Instagram",
                             "Discount x Twitter",
                             "Discount x Bing"),
          omit=c("V","Constant","^Registration.Source"),
          model.names=F,omit.stat=c("ser","rsq","n","adj.rsq"))

##
##
================================================================================
=======
##
##                                All Controls          All Controls + Group
Interactions
##                                    (1)                              (2)
## -------------------------------------------------------------------------
----------
## Discount                         12.464***                        5.067***
##                                   (0.114)                          (0.056)
##
## Discount x Instagram                                              4.956***
##                                                                    (0.079)
##
## Discount x Twitter                                                9.849***
##                                                                    (0.079)
##
## Discount x Bing                                                   14.901***
##                                                                    (0.079)
##
## F Statistic          262,819.300*** (df = 6; 4993) 2,227,470.000*** (df =
12; 4987)
## -------------------------------------------------------------------------
----------
## Notes:               ***Significant at the 1 percent level.
##                      **Significant at the 5 percent level.
##                      *Significant at the 10 percent level.
```

After running both models, I compared the results using stargazer to summarize and present the differences:

OLS with all controls (model1): This model included the treatment variable Discount and all covariates except for the Registration.Source. The coefficient for Discount was positive and highly significant, suggesting a strong overall effect of the treatment on Revenue.

OLS with group interactions (model2): This model incorporated interactions between Discount and the Registration.Source categories, allowing for the estimation of heterogeneous treatment effects by platform (Instagram, Twitter, Bing). The results showed different coefficients for each interaction term, indicating that the effect of Discount varied significantly across the platforms.

Here is the key takeaway from the output:

Discount: In the first model, the effect of Discount is estimated to be around 12.46 (significant at the 1% level). In the second model, where interactions with platform types are included, the effect of Discount itself is reduced to 5.07, likely due to the adjustment for platform-specific variations.

Discount interactions: The significant interactions for each platform (Instagram, Twitter, Bing) suggest that the effect of Discount is heterogeneous:

Instagram: 4.96 Twitter: 9.85 Bing: 14.90 The F-statistic indicates that both models are highly statistically significant, confirming the relevance of the variables included in the models.

This comparison highlights that while the average effect of Discount is positive across all platforms, the specific impact varies depending on the platform.

```
# fit causal forest
X<-model.matrix(~.,data=dat[,-
which(colnames(dat)%in%c("Revenue","Discount"))])
cf<-causal_forest(X=X,Y=dat$Revenue,W=dat$Discount)
```

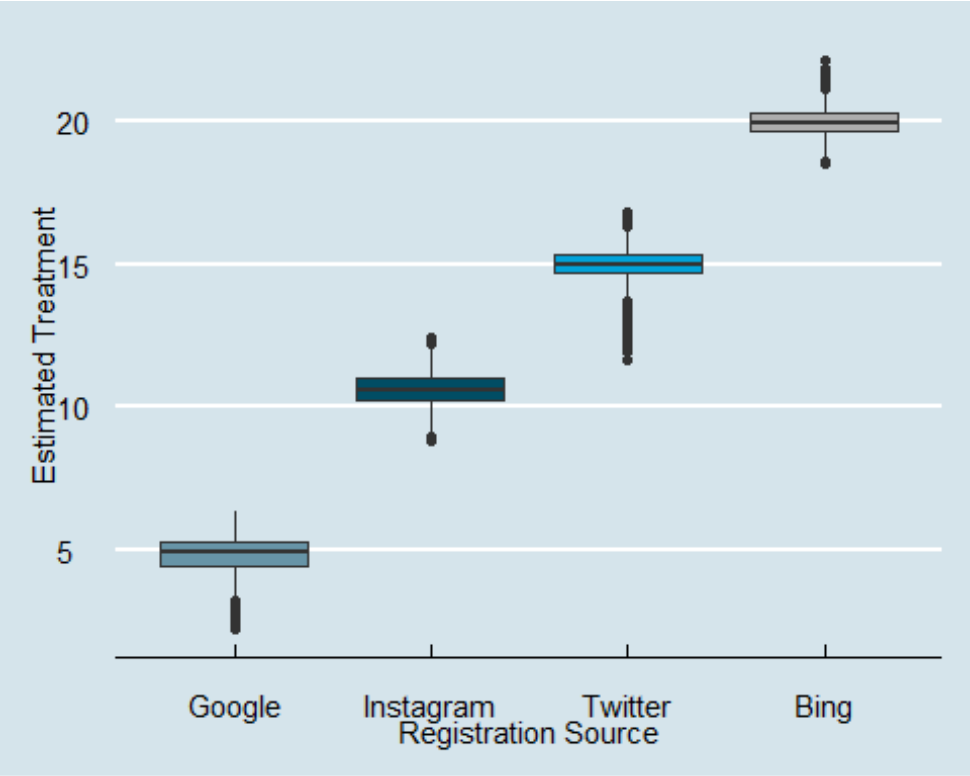
After fitting the causal forest model, I analyzed the results to estimate the heterogeneous treatment effect (HTE) and identify the most important variables that influence this treatment effect.

```
# obtain causal forest model predictions
pred<-predict(cf)$predictions

# view average causal forest model predictions by source
# is estimate of heterogeneous treatment effect
tapply(pred,dat$Registration.Source,mean)

##    Google Instagram   Twitter      Bing
##  4.789604 10.642925 14.944731 20.017197

# extract variable importance from causal forest model
cf %>%
  variable_importance() %>%
  as.data.frame() %>%
```

```
  mutate(variable=colnames(X)) %>%
  arrange(desc(V1))
```

```
##           V1                       variable
## 1 0.62786442      Registration.SourceBing
## 2 0.09434392   Registration.SourceTwitter
## 3 0.06375625                           V3
## 4 0.06196745                           V5
## 5 0.05316400                           V2
## 6 0.04532691                           V4
## 7 0.03060345                           V1
## 8 0.02297359 Registration.SourceInstagram
## 9 0.00000000                  (Intercept)
```

```
# plot distribution of causal forest model predictions by sources
data.frame("est"=pred,"Registration.Source"=dat$Registration.Source) %>%
  ggplot(aes(Registration.Source,pred,fill=Registration.Source)) +
  geom_boxplot() +
  xlab("Registration Source") +
  ylab("Estimated Treatment") +
  theme_economist() +
  scale_fill_economist() +
  theme(legend.position="none")
```



1. Predictions of the Causal Forest Model First, I extracted the predictions from the causal forest model to estimate the treatment effects for each observation.

The output shows the estimated treatment effect (HTE) for each group:

Google: 4.79 Instagram: 10.64 Twitter: 14.94 Bing: 20.02 This indicates that the treatment effect (i.e., the effect of Discount) is largest for Bing, followed by Twitter, Instagram, and Google, suggesting that users from Bing and Twitter are more responsive to the treatment.

2. Variable Importance Next, I extracted the variable importance from the causal forest model to identify which features contributed most to predicting the treatment effect. The following code was used to extract and display the variable importance:

Here are the results, showing the importance of each variable:

Registration.SourceBing: 0.6279 (most important variable) Registration.SourceTwitter: 0.0943 V3: 0.0638 V5: 0.0620 V2: 0.0532 V4: 0.0453 V1: 0.0306 Registration.SourceInstagram: 0.0230 (Intercept): 0.0000 (least important) From this, it's clear that Registration.SourceBing has the highest importance in predicting the treatment effect, followed by Registration.SourceTwitter and the control variables (V1, V2, etc.).

3. Visualization of Treatment Effects To visualize the distribution of the estimated treatment effects across the different Registration.Source categories, I used a boxplot:

This boxplot provides a clear visualization of the variation in estimated treatment effects across each registration source. As expected, the treatment effect is more pronounced for Bing and Twitter, and we can see a larger spread in these groups compared to Google and Instagram.

Summary: Estimated Treatment Effects: The treatment effect is highest for Bing (20.02) and Twitter (14.94), while Instagram (10.64) and Google (4.79) show lower treatment effects. Variable Importance: The most important variable in predicting the treatment effect is Registration.SourceBing, followed by Registration.SourceTwitter. Visualization: The boxplot confirms the heterogeneous treatment effects by registration source, with more variation in Bing and Twitter compared to Google and Instagram.

1. Controlled / Fixed Effects Regression Setup

I began by simulating a dataset with a treatment variable, Use.Mobile.App, and an outcome variable, Retention. Using Received.Email as an instrumental variable (IV), I applied a two-stage regression approach:

Stage 1: Regressed Use.Mobile.App on Received.Email to isolate exogenous variation. Stage 2: Regressed the predicted values of Use.Mobile.App on Retention to estimate the causal effect of the treatment.

2. Regression Discontinuity Setup

In this setup, I simulated a scenario where the treatment assignment was determined by a threshold. By applying a Regression Discontinuity Design (RDD), I estimated causal effects around the threshold, focusing on the local treatment effect by exploiting the discontinuity in the treatment assignment.

3. Difference in Difference Setup

Next, I used a Difference-in-Differences (DiD) approach to estimate the treatment effect of a policy intervention. The dataset included a treatment group and a control group, with pre- and post-treatment periods. The model controlled for time-invariant differences between groups and temporal trends, allowing for robust causal inference.

4. Instrumental Variable Setup

To address endogeneity in the treatment variable, I employed an Instrumental Variables (IV) approach. Received.Email served as the instrument for Use.Mobile.App. By isolating exogenous variation in the treatment, the IV method provided an unbiased estimate of the causal effect of the treatment on Retention.

5. Double Selection Setup

To control for a large number of potential confounders, I implemented Double Selection using lasso regression:

Stage 1: I selected relevant controls for the outcome variable (Customer.Value). Stage 2: I selected controls for the treatment variable (Social.Proof.Variant). The combined selected variables were then used in a regression model to estimate the treatment effect while mitigating overfitting.

6. Causal Forests Setup

Finally, I used Causal Forests to estimate heterogeneous treatment effects (HTE). I simulated data with a treatment variable (Discount) and an outcome (Revenue) and applied causal forests to estimate treatment effects that varied by subgroups (e.g., Registration.Source). This non-parametric approach revealed how treatment effects differed across groups and identified key variables influencing treatment effectiveness.